Z80382/Z8L382

Data Communications Processor

User Manual

UM007103-0302

ZiLOG Worldwide Headquarters • 532 Race Street • San Jose, CA 95126-3432
Telephone: 408.558.8500 • Fax: 408.558.8300 • www.ZiLOG.com

Z80382/Z8L382 Data Communications Processor User Manual

This publication is subject to replacement by a later edition. To determine whether a later edition exists, or to request copies of publications, contact

ZiLOG Worldwide Headquarters
532 Race Street
San Jose, CA 95126-3432
Telephone: 408.558.8500
Fax: 408.558.8300
www.ZiLOG.com

Document Disclaimer

Information in this publication concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded. ZiLOG, INC. DOES NOT ASSUME LIABILITY FOR OR PROVIDE A REPRESENTATION OF ACCURACY OF THE INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED IN THIS DOCUMENT. ZiLOG ALSO DOES NOT ASSUME LIABILITY FOR INTELLECTUAL PROPERTY INFRINGEMENT RELATED IN ANY MANNER TO USE OF INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED HEREIN OR OTHERWISE. Devices sold by ZiLOG, Inc. are covered by warranty and limitation of liability provisions appearing in the ZiLOG, Inc. Terms and Conditions of Sale. ZiLOG, Inc. makes no warranty of merchantability or fitness for any purpose. Except with the express written approval of ZiLOG, use of information, devices, or technology as critical components of life support systems is not authorized. No licenses are conveyed, implicitly or otherwise, by this document under any intellectual property rights.

Table of Contents

List of Figures

List of Tables

Preface

ADDITIONAL SOURCES OF INFORMATION

In addition to this manual, you should have access to and be familiar with the following documentation:

- *Advanced 16-Bit Embedded Controllers Databook*, DB983820100
- *Z80382 Evaluation Kit User's Manual*, UM007201-ZMP0700

TRADEMARKS

Z80, Z180, Z380 and Z80382 are trademarks of ZiLOG, Inc.

Architectural Overview

FEATURES

- Embedded Z380™ Processor
  - Maintains Object Code Compatibility with Z80® and Z180™ Microprocessors
  - Enhanced Instruction Set for 16-Bit Operation
  - 16 MB Linear Addressing
  - Two Clock Cycle Instruction Execution Minimum
  - Four On-Chip Register Banks
  - BC/DE/HL/IX/+ Augmented to 32-Bits
  - Clock Divide-by-Two and Multiply-by-Two Options
  - Fully Static CMOS Design with Low-Power Standby Mode
  - 16-Bit Internal Bus
  - Dynamic Bus Sizing (8/16-Bit inter-Operability)
- 16550 MIMIC with I/O Mailbox, DMA Mailbox, and 16 mA Bus Drive
- Three HDLC Synchronous Serial Channels–Serial data rate of up to 10 Mbps
- GCI/SCIT Bus Interface
- Eight Advanced DMA Channels with 24-Bit Addressing
- Plug-and-Play ISA Interface
- PCMCIA Interface
- Two Enhanced ASCIs (UARTs) with 16-Bit Baud Rate Generators (BRG)

- Clocked Serial I/O Channel (CSIO) for Use with Serial Memory
- Two 16-Bit Timers with Flexible Prescalers
- Three Memory Chip Selects with Wait-State Generators
- Watch-Dog Timer (WDT)
- Up to 32 General Purpose I/O Pins
- DC to 20 MHz Operating Frequency @ 5.0V
- DC to 10 MHz Operating Frequency @ 3.3V
- 144-Pin QFP and VQFP Packages

GENERAL DESCRIPTION

The Z80382 (Z382) is designed to address high-end data communication applications such as digital modems (ISDN, GSM, Mobitex & Modacom), xDSL and analog modems (V.34 and beyond). The Z80382 provides a performance upgrade to existing Z80- and Z18x-based designs by utilizing the increased bandwidth of the 380C processor.

Note: In this document the notation *380C* denotes the Z380-compatible CPU core which is embedded in the Z80382.

The 380C microprocessor is a high-performance processor with fast and efficient throughput and increased memory addressing capabilities. The 380C offers a continuing growth path for present Z80- or Z18x-based designs, while maintaining Z80 and Z180 object code compatibility. Its enhancements include added instructions, expanded sixteen megabyte address space and flexible bus interface timing.

In the 380C, the basic addressing modes of the Z80 microprocessor have been augmented to include Stack Pointer Relative loads and stores, 16-bit and 24-bit indexed offsets, and more flexible Indirect Register addressing. Internally, all of the addressing modes allow up to 32-bit linear addressing. However, because the Z382 has only 24 address pins, it can only address 16 MB of memory.

Other additions to the instruction set include:

- A full complement of 16-bit arithmetic and logical operations
- 16-bit I/O operations, multiply and divide
- A complete set of register-to-register loads and exchanges

The 380C register file includes alternate versions of the IX and IY registers. There are four banks of registers in the 380C, along with instructions for switching among them. All of the 16-bit register pairs and index registers in the basic Z80 microprocessor register file are expanded to 32 bits.

The Z80382 includes dynamic bus sizing to allow any mix of 16-and 8-bit memory, and I/O devices in a system. One application for this capability would be to copy code from a low-cost, slow 8-bit ROM to 16-bit RAM, from which the code can be executed at much higher speeds. Memory bus sizes can be configured internally by software to eliminate the necessity for external logic to drive MSIZE.

Some features that have traditionally been handled by external peripherals are incorporated in the Z80382. These on-chip peripherals reduce system chip count and interconnections on the external bus. Figure 1 illustrates these peripherals, and they are summarized in the following paragraphs. Figure 2 illustrates the pin names.

HDLC Synchronous Channels. Three HDLC channels operate at serial data rates of up to 10 Mbps and feature 8-byte receive and transmit FIFOs. These channels can be used for modems, general data communications, and Integrated Services Digital Network (ISDN). The ISDN can be handled separately or through the GCI/SCIT bus interface. HDLG Channels always transfer data through the DMA channels. A TRANSPARENT mode is selectable. Two of the HDLC cells can be pin multiplexed with the ASCIs (UARTs) to provide dynamically switchable (async-sync) DTE interfaces.

DMA Channels. The eight DMA channels provide 24-bit memory addressing and can transfer memory block sizes of up to 64 KBytes (16-bits). These DMA channels can be dynamically assigned to serve the HDLC ports, MIMIC COM port, Host DMA Mailbox, or ASCIs in any mixture. Linked list operation allows all HDLC transmitters and receivers to operate at or above E1 rates simultaneously without loading the bus bandwidth.

16550 MIMIC. Provides connection to a PC ISA bus and emulation of the 16550 UART register set. Improvements include 16 mA output drivers and internal Com port address decode to reduce external PC interface components.

ASCI. Two flexible asynchronous serial channels with baud rate generators, modem control and status.

CSI/O. A clocked serial I/O channel which can be used for serial memory interface.

Timers. Two 16-bit counter/timers with flexible prescalers for wide-range timing applications.

GCI/SCIT Bus Interface. A common interface to ISDN interface devices. Internal signals from this module can be connected to the HDLC channels to provide B-channels and D-channel for ISDN.

Plug-and-Play ISA Interface. Provides auto-configuration in ISA (AT bus) applications.

PCMCIA Interface. Provides connectivity to a PCMCIA bus.

32-Bit General-Purpose I/O. For non-PC add-in applications, four 8-bit ports are provided for general-purpose I/O In ISA or PCMCIA applications, the pins from two of the ports are reallocated to host bus signals and are not available. Pins from the other two ports are selectively multiplexed with on-chip peripheral functions (ASCls, CSI/O PRT). These pins are individually programmable for input/output mode.

I/O Chip Selects. Two I/O chip selects are provided to support I/O access of external peripherals. Each has a programmable base address and provides I/O decode sizes ranging from eight to 512 bytes.

ROM/RAM Chip Selects with Wait-State Generators. Chip select outputs are provided to decode memory addresses and provide memory chip enables. Each chip select has its own wait state generator to allow use of memories with different speeds.

Watch-Dog Timer. A Watch-Dog Timer (WDT) with a wide range of time-constants prevents code runaway and possible resulting system damage. The $\overline{\text{RESET}}$ input can be forced as an output upon the terminal count of the WDT. This allows external peripherals to be reset along with the Z80382.

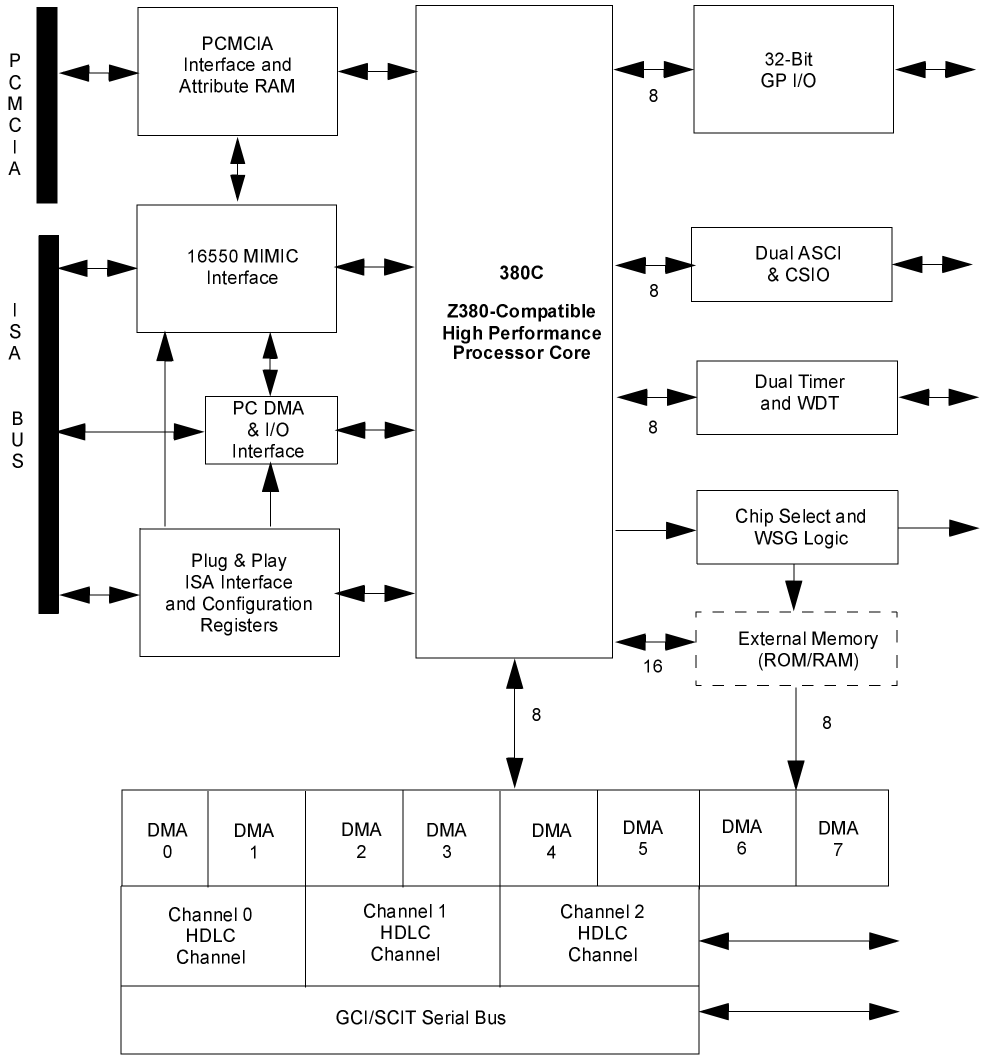


Figure 1. Z80382 Block Diagram

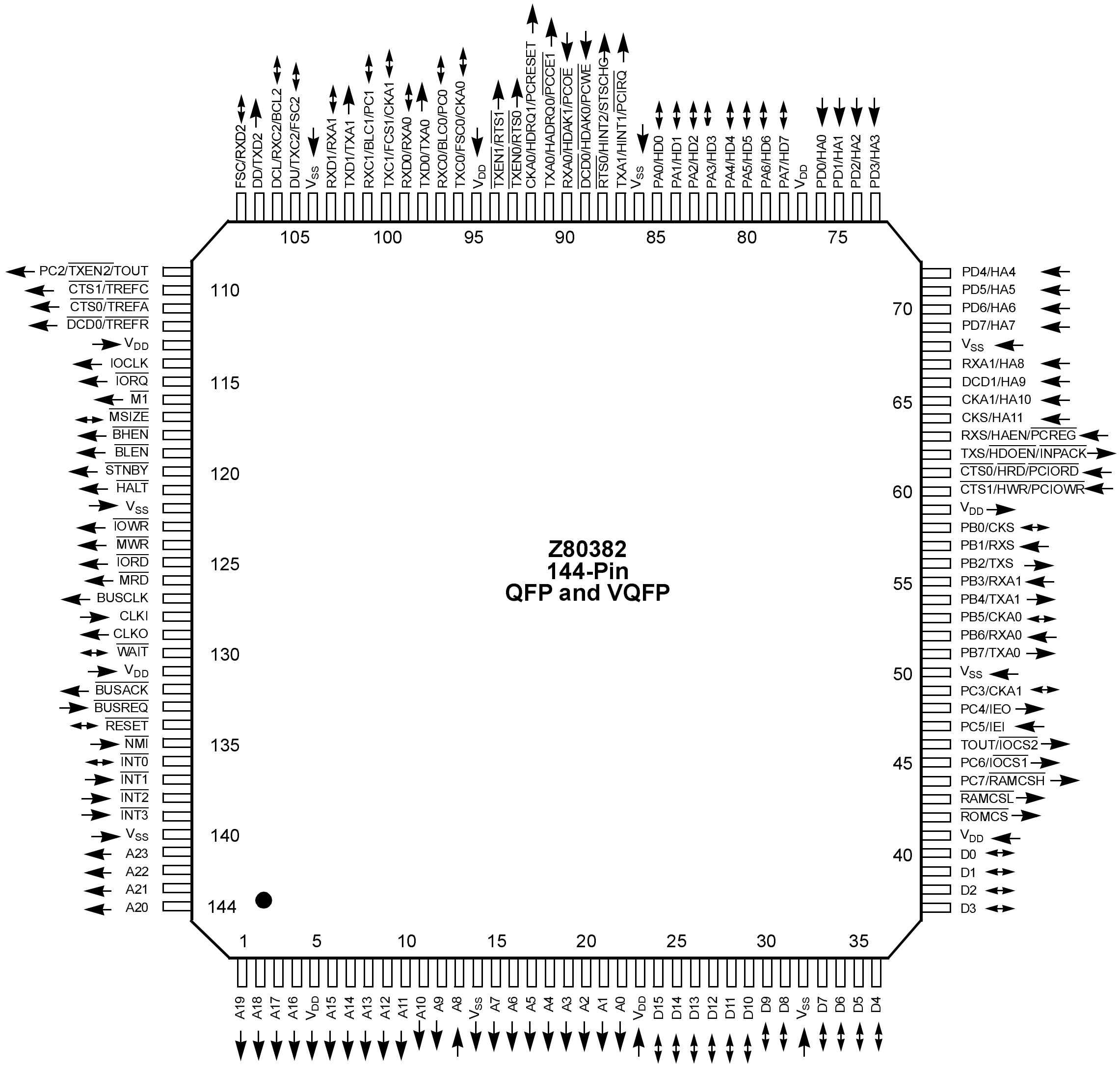


Figure 2. Z80382 144-Pin QFP and VQFP Pin Diagram

PIN DESCRIPTIONS

The following tables describe the input and output signals of the Z80382. Signals are asserted in the High state and negated in the Low state. An overlined signal name indicates that the signal is asserted in the Low state and negated in the High state.

Many pins have multiple functions, and may appear more than once in the pin description tables. In each table, such pins are described using their function in that mode. Likewise, some signals may be output on alternate pins depending on the mode under which the Z80382 is operating. The notation xx/yy in the Pin Number column indicates that the signal may be assigned to pin xx or pin yy.

Table 1. MPU Signals

Pin Name	Pin Number(s)	Description
A23-0	141 - 144, 1 - 4, 6 -13, 15 - 22	**Address Bus** (outputs, active High, tristate): These non-multiplexed address signals provide a linear memory address space of sixteen megabytes. The address signals are also used to access I/O devices.
$\overline{\text{BUSACK}}$	132	**Bus Acknowledge** (output, active Low): This signal, when asserted, indicates that the 380C has accepted an external bus request and has tri-stated its output drivers for the address bus, databus and the bus control signals $\overline{\text{TREFR}}$, $\overline{\text{TREFA}}$, $\overline{\text{TREFC}}$, $\overline{\text{BHEN}}$, $\overline{\text{BLEN}}$ $\overline{\text{MRD}}$, $\overline{\text{MWR}}$, $\overline{\text{IORQ}}$, $\overline{\text{IORD}}$, and $\overline{\text{IOWR}}$. The 380C cannot provide any DRAM refresh transactions while it is in the bus acknowledge state.
$\overline{\text{BHEN}}$	118	**Byte High Enable** (output, active Low, tristate): This signal is asserted at the beginning of a memory or refresh transaction, to request an operation on D15-8. If software initiates a 16-bit memory operation, but $\overline{\text{MSIZE}}$ is asserted indicating a byte-wide memory, only the data on D7-0 is transferred in the current transaction, and another transaction is performed to transfer the other data byte, also on D7-0

Table 1. MPU Signals (Continued)

Pin Name	Pin Number(s)	Description
$\overline{\text{BLEN}}$	119	**Byte Low Enable** (output, active Low, tri-state): This signal is asserted at the beginning of a memory or refresh transaction, to request an operation on D7-0. If software initiates a 16-bit memory transaction, but $\overline{\text{MSIZE}}$ is asserted indicating a byte-wide memory, only the data on D7-0 is transferred in the current transaction, and another transaction is performed to transfer the other data byte, also on D7-0. To align Z80382 documentation and terminology with historical Z80 and industry (for example, Intel) practice, the names of the $\overline{\text{BHEN}}$ and $\overline{\text{BLEN}}$ pins, as well as the D15-8 and D7-0 pins, have been swapped on the Z80382 compared to the Z380. This fact is significant only for those using a Z380 Emulator in a Z80382-based project.
$\overline{\text{BUSREQ}}$	133	**Bus Request** (input, active Low): When this signal is asserted, an external bus master is requesting control of the bus. $\overline{\text{BREQ}}$ has higher priority than all nonmaskable and maskable interrupt requests.
BUSCLK	127	**Bus Clock** (output, active High): This signal is the reference edge for the majority or other signals generated by the 380C. Its frequency may be that of the CLKI pin, or CLKI divided by two or times two.

Table 1. MPU Signals (Continued)

Pin Name	Pin Number(s)	Description
D15-0	24 - 31 33 - 40	**D15-0 Data Bus** (input/output, active High, tristate): This bidirectional 16-bit data bus is used for data transfer between the 380C and memory or I/O devices. In a memory word transfer, the even-addressed (A0 = 0) byte is transferred on D7-0, and the odd-addressed (A0 = 1) byte on D15-8. 8-bit memories must be connected to D7-0, while 8-bit I/O devices must be attached to D15-8 (this difference equalizes electrical loading). See note under $\overline{\text{BLEN}}$ pin description.
$\overline{\text{HALT}}$ $\overline{\text{STNBY}}$	121 120	**Halt, Standby Status** (outputs, active Low): These two outputs indicate the status of the Z80382 as follows: STNBYHALTMode HH: Normal instruction execution HL: Halt instruction LH: Sleep Mode: clock runs but is blocked from most of the chip LL: Standby Mode: oscillator is stopped
$\overline{\text{INT3}}$ $\overline{\text{INT2}}$ $\overline{\text{INT1}}$	139 138 137	**Interrupt Requests** (inputs): Asynchronous maskable interrupt inputs. Can be selected as low- or high-level sensitive, or as falling- or rising-edge triggered.
$\overline{\text{INT0}}$	136	**Interrupt Request** (input/output, active Low, open drain): $\overline{\text{INT0}}$ is logically ORed (positive-logic ANDed) with the interrupt requests from the on-chip MIMIC, DMAs, and HDLC controllers, to create the processor's $\overline{\text{INT0}}$ input.

Table 1. MPU Signals (Continued)

Pin Name	Pin Number(s)	Description
IOCLK	114	**I/O Clock** (output, active High): This signal is a program-controlled divided-down version of BUSCLK. The division factor can be two, four, six or eight with I/O transactions and interrupt-acknowledge transactions occurring relative to IOCLK. IOCLK can be disabled, in which case BUSCLK is the timing reference for I/O transactions. **Note:** The INTACK output of the Z380 has been omitted on the Z80382 for pinning reasons. A similar signal can be easily obtained by low-active-ANDing (positive-logic ORing) the $\overline{\text{M1}}$ and $\overline{\text{IORQ}}$ outputs.
$\overline{\text{IORQ}}$	115	**Input/Output Request** (output, active Low, tristate): This signal is active dung all I/O read and write transactions and interrupt acknowledge transactions.
$\overline{\text{IORD}}$	125	**Input/Output Read Strobe** (output, active Low, tristate): This signal is used to strobe data from the peripherals during I/O read transactions.
$\overline{\text{IOWR}}$	123	**Input/Output Write Strobe** (output, active Low, tri-state): This signal is used to strobe data into the peripherals during I/O write transactions.
$\overline{\text{IOCS1}}$ $\overline{\text{IOCS2}}$	45 46	**I/O Chip Select** (output, active Low): These outputs may be used to access external I/O devices. The base I/O address and range are programmable.

Table 1. MPU Signals (Continued)

Pin Name	Pin Number(s)	Description
$\overline{\text{M1}}$	116	**Machine Cycle One** (output, active Low, tristate): This signal is active during instruction fetch and interrupt acknowledge transactions. The Z80382 does not support RETI decoding by Z80 peripherals (P10), SIO, and CTC. It does support Z80-type interrupt daisy-chaining by devices that include explicit clearing of IUS (for example, SCC).
$\overline{\text{MRD}}$	126	**Memory Read** (output, active Low, tristate): This signal indicates that the addressed memory location places its data on the data bus. $\overline{\text{MRD}}$ is active from the end of T1 until the end of T4 during memory read transactions.
$\overline{\text{MSIZE}}$	117	**Memory Size** (input/open-drain output, active Low): In 16-bit memory operations, this signal indicates whether the addressed memory location is word size (logic High) or byte size (logic Low). In the latter case, the 8-bit memory must be connected to the D7-0 lines, and an additional memory transaction on D7-0 is automatically generated to transfer the other byte of the word. (See the note on pin name swapping after the $\overline{\text{BLEN}}$ pin description.) $\overline{\text{MSIZE}}$ is driven as an open-drain output by the memory decoding modules, when they are enabled in 8-bit mode and the address falls within their range.
$\overline{\text{MWR}}$	124	**Memory Write** (output, active Low, tristate): This signal indicates that the addressed memory location stores the data on the databus, as qualified by $\overline{\text{BHEN}}$ and $\overline{\text{BLEN}}$. $\overline{\text{MWR}}$ is active from the end of T2 until the end of T4 during memory write transactions.

Table 1. MPU Signals (Continued)

Pin Name	Pin Number(s)	Description
$\overline{\text{NMI}}$	135	**Nonmaskable Interrupt** (input, falling edge-triggered): This input has higher priority than the maskable interrupt inputs $\overline{\text{INT3}}$ - $\overline{\text{INT0}}$
$\overline{\text{RESET}}$	134	**Reset** (input/output, active Low, open drain): This input must be active for a minimum of five BUSCLK periods to initialize the Z80382. The effect of $\overline{\text{RESET}}$ is described in detail in the Reset section.
$\overline{\text{ROMCS}}$	42	**ROM Chip Select** (output, active Low): After Reset, the Z80382 drives this output and $\overline{\text{MSIZE}}$ Low for all memory accesses with A23 = 0. Software can program the chip select logic to assert $\overline{\text{ROMCS}}$ for a different range of memory addresses. If ROM is 16 bits wide and composed of two 8-bit devices, connect the Chip Select inputs of both devices to $\overline{\text{ROMCS}}$, and program the hardware not to force $\overline{\text{MSIZE}}$ low in the first two instructions of the ROM code.
$\overline{\text{RAMCSL}}$ $\overline{\text{RAMCSH}}$	43 44	**RAM Chip Select Low, High** (outputs, active Low): After Reset, the Z80382 drives $\overline{\text{RAMCSL}}$ and $\overline{\text{MSIZE}}$ Low for memory cycles with A23-1, and puts the $\overline{\text{RAMCSH}}$ pin under control of its alternate use (a port pin). If RAM is only eight bits wide, connect its Chip Select input to $\overline{\text{RAMCSL}}$. If RAM is sixteen bits wide, connect one of these pins to the chip select of each 8-bit RAM, and reprogram the hardware not to force $\overline{\text{MSIZE}}$ Low, in which case $\overline{\text{RAMCSL}}$ is qualified with $\overline{\text{BLEN}}$ and $\overline{\text{RAMCSH}}$ is qualified with $\overline{\text{BHEN}}$. On the Z80382 these signals have the same timing as address lines, so there is no timing penalty for this qualification.

Table 1. MPU Signals (Continued)

Pin Name	Pin Number(s)	Description
$\overline{\text{TREFA}}$	111	**Timing Reference A** (output, active Low, tristate): This timing reference signal goes Low at the end of T2 and returns High at the end of T4 during a memory read, memory write or refresh transaction. It can be used to control the address multiplexer for a DRAM interface or as the $\overline{\text{RAS}}$ signal at higher processor clock rates.
$\overline{\text{TREFC}}$	110	**Timing Reference C** (output, active Low, tristate): This timing reference signal goes Low at the end of T3 and returns High at the end of T4 during a memory read, memory write or refresh transaction. It can be used as the $\overline{\text{CAS}}$ signal for DRAM accesses.
$\overline{\text{TREFR}}$	112	**Timing Reference R** (output, active Low, tristate): This timing reference signal goes Low at the end of T1 and returns High at the end of T4 during a memory read, memory write or refresh transaction. It can be used as the $\overline{\text{RAS}}$ signal for DRAM accesses.
$\overline{\text{WAIT}}$	130	**Wait** (input, active Low): This input is sampled by either BUSCLK or IOCLK, to insert Wait states into the current bus transaction.

Table 2. UART, Timer and CSI/O Signals

Pin Name	Pin Number(s)	Description
CKA0 CKA1	53/92/96 49/65/100	**Asynchronous Clock 0, 1** (Bidirectional): Clock signals to or from the asynchronous channels (ASCIs).
$\overline{\text{CTS0}}$ $\overline{\text{CTS1}}$	61/111 60/110	**Clear to Send 0,1** (Inputs, active Low): Transmit control signals for the ASCI channels.
$\overline{\text{DCD0}}$ $\overline{\text{DCD1}}$	89/112 66	**Data Carrier Detect 0,1** (Inputs, active Low): Receive control signals for the ASCI channels. $\overline{\text{DCD1}}$ is not available in ISA applications.
$\overline{\text{RTS0}}$ $\overline{\text{RTS1}}$	88/93 94	**Request to Send 0,1** (Outputs, active Low, tristate): Software-controlled output from the ASCI channels.
RXA0 RXA1	52/90/99 55/67/103	**Receive Data 0,1** (Inputs): ASCI Receive data.
TXA0 TXA1	51/91/98 54/87/102	**Transmit Data 0,1** (Outputs): ASCI Transmit data.
CKS	58/64	**Serial Clock** (Bidirectional): The clock for the CSIO channel.
RXS	57/63	**Clocked Serial Receive Data** (Input): Receive data for the CS 10 channel.
TXS	56/62	**Clocked Serial Transmit Data** (Output): Transmit data from the CSIO channel.
TOUT	46/109	**Timer Out** (Output, active High): Pulse output from PRT1.

Table 3. ISA Bus Signals

Pin Name	Pin Number(s)	Description
HD7-0	78 - 85	**Host Data Bus** (Input/Output, tristate): ISA or PCMCIA data bus.
$\overline{\text{HDOEN}}$	62	**Host Data Output Enable** (Output, active Low): This signal goes Low when the Host reads data from the MIMIC, the I/O Mailbox, or the Plug and Play interface, and during Host DMA read cycles.
HA11-0	64 - 67 69 - 76	**Host Address** (Input): Part of the ISA or PCMCIA address bus. The MS bits can be decoded by the built-in address decoder; bits 2-0 determine which MIMIC register the Host accesses. Bits 11-10 are decoded only by the Plug and Play ISA module.
HAEN	63	**Host Address Enable** (Input): HAEN must be low to qualify COM Port decoding, I/O Mailbox decoding, and Plug and Play decoding. To support 16-bit decoding of Host I/O addresses, provide an external decoder for HA15-12 and HAEN all Low and connect its Low-active output to this pin.
$\overline{\text{HWR}}$	60	**Host Write** (Input, active Low): The Host drives this input low to signal the MIMIC that a write operation is taking place. $\overline{\text{HWR}}$
$\overline{\text{HRD}}$	61	**Host Read** (Input, active Low): This input is used by the Host to signal the MIMIC interface that a read operation is taking place.

Table 3. ISA Bus Signals (Continued)

Pin Name	Pin Number(s)	Description
HINT1	87	**Host Interrupt** (Outputs, active High, tristate): One of these outputs is driven High by the Plug and Play module when the MIMIC requests an interrupt from the Host. The unused signal is tristated.
HINT2	88	
$\overline{\text{HDAK0}}$	89	**Host DMA Acknowledge** (Inputs, active Low): These inputs indicate that the Host DMA controller has acknowledged the request and is transferring data.
$\overline{\text{HDAK1}}$	90	
HDRQ0	91	**Host DMA Request** (Outputs, active High, tristate): These outputs request a DMA transfer operation from the Host.
HDRQ1	92	

Table 4. Parallel Ports

Pin Name	Pin Number(s)	Description
PA7-0	78 - 85	**Parallel Ports A, B, C, D** (input/Outputs): These lines can be configured as inputs or outputs on a bit-by-bit basis. In an ISA or PCMCIA application, Ports A and D are not pinned out, the registers for Ports A and D are used by the MIMIC function, and Ports B and C are selectively multiplexed with the on-chip peripherals (ASCIs, CSI/O, PRT). In other applications all four ports are available with minimal multiplexing.
PB7-0	51 - 58	
PC7-0	44 - 45, 47 - 49, 109, 101, 97	
PD7-0	69 - 76	

Table 5. HDLC Serial Channel and GCI/SCIT Signals

Pin Name	Pin Number(s)	Description
TxD0 TxD1 TxD2	98 102 107	**HDLC Transmit** (outputs, tristate): These pins are used to transmit serial data from the HDLC controllers when they are not operating via the GCI/ SCIT interface.
RxD0 RxD1 RxD2	99 103 108	**HDLG Receive** (inputs/outputs, tristate): These pins are used to receive serial data for the HDLC controllers when they are not operating via the GCI/ SCIT interface. In GCI mode, RxD0 and RxD1 can be programmed to be outputs for the received data.
TxC0/FSC0 TxC1/FSC1 TxC2/FSC2	96 100 105	**HDLC Transmit Clock/Frame Sync** (input/ outputs): 1n non-TDM, non-GCI modes, these can be used as external bit clock inputs or can be programmed to output the Tx clock. In non-GCI TDM mode, these pins carry the Frame Sync pulse.
RxC0/BCL0 RxC1/BCL1 RxC2/BCL2	97 101 106	**HDLC Clock/Bit Clock** (inputs/outputs, tristate): Optional external bit clock inputs. These pins can be outputs for the GCI Rx clock in GCI mode (RxC0 and RxC1), or outputs for the DPLL clock in NRZI mode.
$\overline{\text{TxEN0}}$ $\overline{\text{TxEN1}}$ $\overline{\text{TxEN2}}$	93 94 109	**HDLC Transmit Enable** (outputs, active Low): In a non-GCI TDM mode, these outputs indicate when an HDLC Transmitter is enabled and is in its active time slot. In non-GCI, non- TDM mode, these outputs are Low when the Transmitter is enabled. They can be used to enable an external driver on the TxD line.

Table 5. HDLC Serial Channel and GCI/SCIT Signals (Continued)

	Pin	
Pin Name	Number(s)	Description
DU DD	105 107	**GCI/SCIT Data Upstream, Downstream** (input/outputs, open-drain): The two bidirectional data streams of the GCI/SCIT interface.
DCL	106	**GCI/SCIT Clock** (input): Bit clock for the GCI/SCIT interface.
FSC	108	**GCI/SCIT Frame Sync** (input): This pin is used to synchronize the GCI/SCIT serial frames. This pin is driven active by “the upstream device” (ISDN transceiver) at the start of each GCI/SCIT frame.

Table 6. PCMCIA Interface Signals

	Pin	
Pin Name	Number(s)	Description
HA9-0	66 - 67 69 - 76	**PCMCIA Address Bus** (inputs): Provide Host PC addressing of attribute memory, configuration registers, and MIMIC. This bus is decoded by the I/O address decoder.
HD7-0	78 - 85	**PCMCIA Data Bus** (input/outputs): This bus is used for data transfers among the Host PC and the MIMIC, the attribute memory, and the configuration registers.
$\overline{\text{PCIORD}}$	61	**PCMCIA I/O Read** (input, active Low): Used to generate the INPACK signal when an I/O read cycle is within the configured range, and reads from the MIMIC.
$\overline{\text{PCIOWR}}$	60	**PCMCIA I/O Write** (input, active Low): This signal is used to write to the MIMIC.

Table 6. PCMCIA Interface Signals (Continued)

Pin Name	Pin Number(s)	Description
PCWE	89	**PCMCIA Write Enable** (input, active Low): This pin is used to write to the attribute memory or to the configuration register which is addressed via HA9-1. This type of operation is recognized when PCWE, PCCE1, and PCREG are all Low, and either the interface is configured for "I/O and memory" operation, or PCIRQ is High, signifying "ready", when configured as a "memory only" interface. The data applied while PCWE is Low are written to the attribute memory range on the positive edge of the PCWE or Card-enable (PCCE1) signal.
PCOE	90	**PCMCIA Output Enable** (input, active Low): PCOE, PCCE1, and PCREG all Low signify a read from attribute memory or a configuration register as selected by HA9-1.
PCCE1	91	**PCMCIA Chip Enable 1** (input, active Low): PCCE1 Low indicates a read or write access to an even addressed byte in attribute memory, a configuration register, or the MIMIC.
PCREG	63	**PCMCIA Register Select** (input, active Low): PCREG Low indicates a read or write access to the attribute memory range or to the I/O address range.
INPACK	62	**PCMCIA Input Acknowledge** (Output, active Low): UNPACK goes Low while an I/O read access is performed within the configured I/O address range. If the PCMCIA interface is configured such that it reacts independent of the address to all I/O read cycles, then INPACK is activated with PCIORD

Table 6. PCMCIA Interface Signals (Continued)

Pin Name	Pin Number(s)	Description
$\overline{\text{PCIRQ}}$	87	**PCMCIA Interrupt Request** (Output, active Low): After the PCMCIA interface is reset it is in a *memory-only* mode, and this signal is driven Low to signify a Busy state until the 380C writes a register bit to say it is Ready. After the card is then configured by the Host, $\overline{\text{PCIRQ}}$ goes Low to request a Host PC interrupt when the internal INT0 signal is asserted by the MIMIC. $\overline{\text{PCIRQ}}$ is monitored by the PCMCIA Host adapter and, dependent on the configuration, connected to one of the Host interrupts (for example, COM1 or COM2 interrupt). $\overline{\text{PCIRQ}}$ can be programmed to be a pulsed interrupt with a minimal pulse length of one microsecond, or a level-interrupt that is reset when the interrupt is processed by the Host. This choice is made by bit 6 of the Configuration Option Register.
PCRESET	92	**PCMCIA Reset** (input, active High): Setting PCRESET High resets the PCMCIA interlace. The card configuration register is cleared and the PCMCIA interface operates in the *memory-only* mode until it is configured again. The attribute memory has to be initialized by the controller, and the Ready/Busy ($\overline{\text{PCIRQ}}$) signal has to be deactivated.
STSCHG	88	**PCMCIA Status Change** (output): This output is controlled by a bit in the PCMCIA module's 380C Control Register.

Table 7. Miscellaneous Signals

Pin Name	Pin Number(s)	Description
CLKI	128	**Clock/Crystal** (input, active High): An externally generated clock can be input at this pin. Alternatively, a crystal can be connected between CLKI and CLKO. In either case, the frequency at this pin can be used directly as the processor clock (BUSCLK), or divided by two or multiplied by two, under software control.
CLKO	129	**CLKO Crystal** (output, active High): Crystal oscillator connection. This pin must be left open if an externally generated clock is input at the CLKI pin. Feedback on this pin can be disabled by software to save power and noise when an external clock is used.
IEI	47	**Interrupt Enable In** (input, active High): If external devices are connected to $\overline{\text{INT0}}$, and have higher interrupt priority than the on-chip MIMIC, DMAs, and HDLC controllers, this signal must be connected to the IEO output of the lowest-priority among such devices.

Table 7. Miscellaneous Signals (Continued)

Pin Name	Pin Number(s)	Description
IEO	48	**Interrupt Enable Out** (output, active high): If external devices are connected to ($\overline{\text{INT0}}$, and have lower interrupt priority than the on-chip MIMIC, DMAs, and HDLC controllers, this signal must be connected to the IEI input of the highest-priority such device.
V_{DD}	5, 23, 41, 59, 77, 95, 113, 131	**Power Supply:** These eight pins carry power to the device. They must be tied to the same voltage externally.
V_{SS}	14, 32, 50, 68, 86,104, 122, 140	**Ground:** These eight pins are the ground references for the device. They must be tied to the same voltage externally.

380C Central Processing Unit

OVERVIEW

The Central Processing Unit (CPU) core of the Z80382 is the 380C (Z380), which is a binary-compatible extension of the Z80 and Z180 CPU architectures. High throughput rates for the 380C are achieved by a high clock rate, high bus bandwidth and instruction fetch/execute overlap. Communicating to the external world through a 24-bit address bus and an 8- or 16-bit data bus, the 380C is a full 32-bit machine internally, with a 32-bit ALU and 32-bit registers.

The following sections provide a brief overview of the Z380. For additional information, refer to the *Z380 User's Manual,* ZiLOG document DC8297-03, available from your ZiLOG representative or distributor.

Figure 3 illustrates the 380C Core CPU functional block diagram.

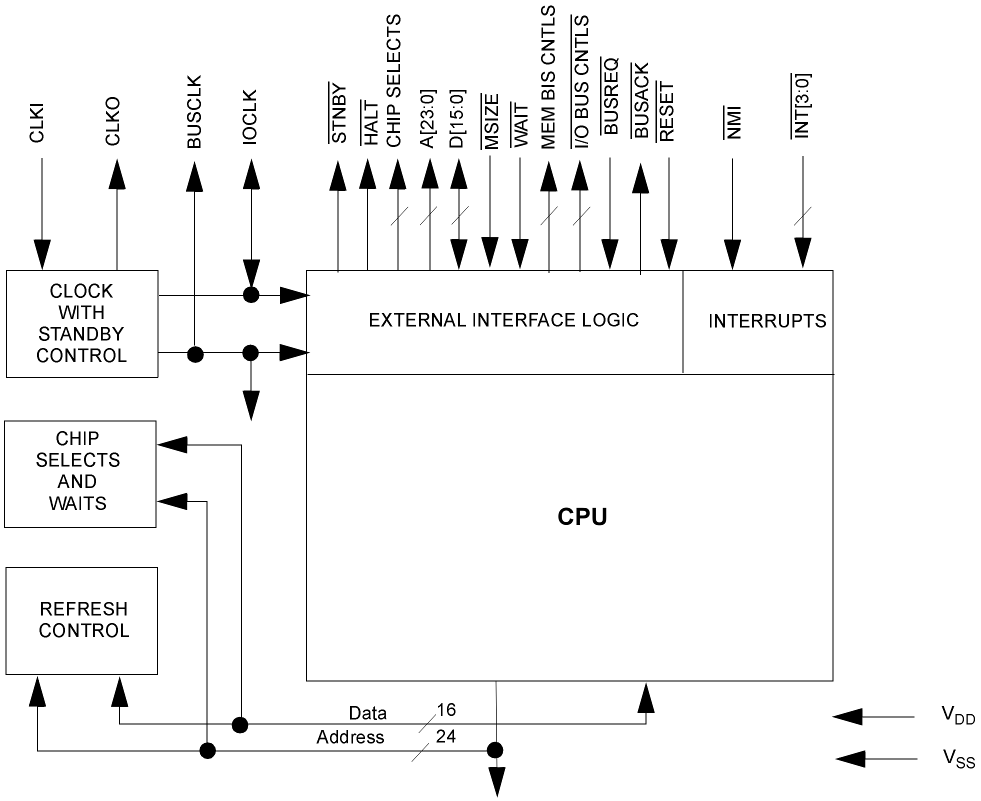


Figure 3. 380C Core CPU Functional Block Diagram

MODES OF OPERATION

The 380C can operate in either NATIVE or EXTENDED mode, as controlled by a bit in the Select Register (SR). In NATIVE mode (the default configuration after Reset), all address manipulations are performed modulo 65536 (16 bits). In this mode the Program Counter (PC) increments across 16 bits. All address manipulation instructions (increment, decrement, add, subtract, indexed addressing, stack relative addressing, and PC relative addressing) operate on 16 bits. The Stack Pointer (SP) increments and decrements across 16 bits. The program counter high-order word is left at all zeros, as are the high-order words of the stack pointer and the I register. NATIVE mode is fully compatible with the Z80 CPU's 64 KByte address space. It is still possible to address memory outside of the 64 KB address space for data storage and retrieval in NATIVE mode, however, as direct addresses, indirect addresses, and the high-order word of the SP, I and the BC, DE, HL, IX and IY registers may be loaded with non-zero values. Executed code and interrupt service routines must reside in the lowest 64 KB of the address space.

In EXTENDED mode, all address manipulation instructions operate on 32 bits, potentially allowing access to a 4 GB address space. In both NATIVE and EXTENDED modes, the Z80382 outputs 24 bits of the address onto the external address bus, limiting the actual usable address space to 16 MB. Only the width of manipulated addresses distinguish NATIVE from EXTENDED mode. The 380C implements instructions to allow switching between NATIVE to EXTENDED modes.

Caution: Switching between NATIVE, EXTENDED, WORD and LONG WORD modes must be done with care, particularly with respect to stacked data and addresses. Appendix A discusses considerations for such mode switching.

In addition to NATIVE and EXTENDED mode, which is specific to memory space addressing, the 380C can operate in either WORD or LONGWORD mode specific to data load and exchange operations. In WORD mode (the reset configuration), all word load and exchange operations manipulate 16-bit quantities. For example, only the low-order words of the source and destination are exchanged in an exchange operation, with the high-order words unaffected.

In LONG WORD mode, all 32 bits of the source and destination are exchanged. The 380C implements two instructions plus decoder directives to allow switching between WORD and LONG WORD modes.

The two instructions perform a global switch, while the decoder directives select a particular mode only for the instruction that they precede.

Note: All word data arithmetic (as opposed to address manipulation arithmetic), rotate, shift and logical operations are always in 16-bit quantities. They are not controlled by either the NATIVE/EXTENDED or WORD/LONG WORD selections. The exceptions to the 16-bit quantities are, of course, those multiply and divide operations with 32-bit products or dividends.

Lastly, all word input/output operations are performed on 16-bit values.

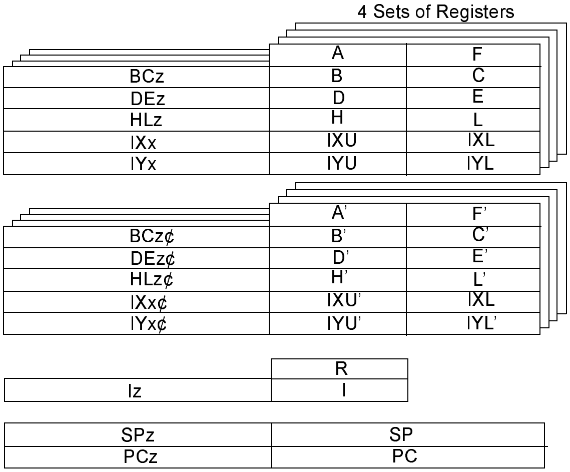
CPU ADDRESS SPACES

The 380C architecture supports several address spaces corresponding to the different types of locations that can be accessed by the CPU. These address spaces are:

- CPU register space
- Memory address space
- I/O address space

CPU Register Space

The CPU register space is depicted in Figure 4 and consists of all of the registers in the CPU register file. These CPU registers are used for data and address manipulation, and are an extension of the Z80 CPU register set, with four sets of this extended Z80 CPU register set present in the 380C. Access to these registers is specified in the instruction, with the active register set selected by bits in the Select Register (SR) in the CPU control register space.



4 Sets of Registers

	A	F
BCz	B	C
DEz	D	E
HLz	H	L
IXx	IXU	IXL
IYx	IYU	IYL

	A’	F’
BCz¢	B’	C’
DEz¢	D’	E’
HLz¢	H’	L’
IXx¢	IXU’	IXL
IYx¢	IYU’	IYL’

	R
Iz	I

SPz	SP
PCz	PC

Figure 4. Register File Organization

Primary and Working Registers

Each working register set is divided into two register files, the primary file and the alternate file. Each file contains an 8-bit Accumulator (A), a Flag register (F), and six general-purpose registers (B, C, D, E, H, and L). Only one file can be active at any given time, although data in the inactive file can still be accessed. Exchange instructions allow the programmer to exchange the active file with the inactive file.

The accumulator (A) is the destination register for 8-bit arithmetic and logical operations. The six general-purpose registers can be paired (BC, DE, and HL), and are extended to 32 bits by the ‘z’ extension to the register to form three 32-bit general-purpose registers. The HL register serves as the 16-bit or 32-bit accumulator for word operations.

CPU Flag Register

The Flag register contains six flags that are set or reset by various CPU operations:

- Carry
- Add/Subtract
- Parity/Overflow
- Half Carry
- Zero
- Sign

Index Registers

The four Index registers in each working register set, IX, IX′, IY and IY′, each hold a 32-bit base address that is used in the Indexed addressing mode. The Index registers can also function as general-purpose registers with the upper and lower bytes of the lower 16 bits being accessed individually.

Interrupt Register

The Interrupt register (I) is used in INTERRUPT modes 2 and 3 for $\overline{\text{INT0}}$ to generate a 32-bit indirect address to an Interrupt Service Routine (ISR). The I register supplies the upper bits of the indirect address and the interrupting peripheral supplies the Lower bits. In the Assigned Vectors mode for $\overline{\text{INT[3:1]}}$, the upper bits of the vector are supplied by the I register, the middle bits come from the Assigned Vector Base register and the least significant bits are a unique value for each interrupting source.

Program Counter

The Program Counter (PC) is used to sequence through instructions in the currently executing program and to generate relative addresses. The PC contains the 32-bit address of the current instruction being fetched from memory. In the Native mode, the PC is effectively only 16 bits long, as carries from bit 15 to bit 16 are inhibited in this mode. In Extended mode, the PC is allowed to increment across all 32 bits.

R Register

The R register can be used as a general-purpose 8-bit read/write register.

Stack Pointer

The Stack Pointer (SP) is used for saving information when an interrupt or trap occurs and for supporting subroutine calls and returns. Stack Pointer relative addressing allows parameter passing using the SP.

Select Register

The contents of the Select Register (SR) determine the CPU operating mode, which register bank is used, the INTERRUPT mode in effect, and other items of this type.

Memory Address Space

The memory address space can be viewed as a linear space of 4GB. The actual usable memory space in the Z80382, however, is 16MB, since only the lower 24 bits of the address are output on the external address bus. The 8-bit byte is the basic addressable element in the 380C memory address space. However, there are other addressable data elements; bits, 2-byte words, byte strings, and 4-byte long words. The size of the data element being addressed depends on the instruction being executed as well as the WORD/LONGWORD mode.

When a word is stored in memory, less significant bytes are stored at lower memory addresses than more significant bytes, as in the Z80 CPU architecture. Also, in 16-bit memory transfers, the less significant byte is present on the D[7:0] lines of the external data bus. “Register Descriptions” on page 61 discusses such byte ordering in greater detail.

I/O Address Space

The 380C CPU architecture distinguishes between memory and I/O spaces and, requires specific I/O instructions. I/O instructions are used to access the Z80382's internal peripherals as well as a number of control registers which deal with functions such as interrupts and traps. I/O instructions are also used to access external peripheral controllers connected to the Z80382's external address, data and control busses.

Note: Like the Z8018x family but unlike the Z380, the Z382 I/O instructions IN0, OUT0, TSTIO, OTIM, OTIMR, OTDM, and OTDMR differ from the older Z80 I/O instructions in providing address bits A8 and above as all zero, not in accessing a separate on-chip I/O space.

DATA TYPES

The Z380 CPU can operate on bits, Binary-Coded Decimal (BCD) digits (4 bits), bytes (8 bits), words (16 bits or 32 bits), byte strings, and word strings. Bits in registers can be set, cleared, and tested. BCD digits, packed two to a byte, can be manipulated with the Decimal Adjust Accumulator instruction (in conjunction with binary addition and subtraction) and the Rotate Digit instructions. Bytes are operated on by 8-bit load, arithmetic, logical, and shift and rotate instructions. Words are operated on in a similar manner by the word load, arithmetic, logical, and shift and rotate instructions. Block move and search operations can manipulate byte strings and word strings up to 64 KB long. Block I/O instructions have identical capabilities.

ADDRESSING MODES

Addressing modes are used by the 380C to calculate the effective address of an operand needed for execution of an instruction. Seven addressing modes are supported by the CPU. Of these seven, one is an addition to the Z80 CPU addressing modes (Stack Pointer Relative) and the remaining six modes are either existing or extensions to the Z80 CPU addressing modes.

Register Addressing

The operand is

- One of the 8-bit registers (A, B, C, D, E, H, L, IXU, IXL, IYU, IYL, A′, B′, C′, D′, E′, H′, L′, IXU′, IXL', IYU′, or IYL′, or
- One of the 16-bit or 32-bit registers (BC, DE, HL, IX, IY, BC′, DE′, HL′, IX′, IY′ or SP) or
- One of the special registers (I or R).

Immediate Addressing

The operand is in the instruction itself and has no effective address. The DDIR IB and DDIR IW decoder directives allow specification of 24-bit and 32-bit immediate operands, respectively.

Indirect Register Addressing

The contents of a register specify the effective address of an operand. The HL register is the primary register used for memory accesses, but BC and

DE can also be used. (For the JP instruction, IX and IY can also be used for indirection.) The BC register is used for I/O space accesses.

Direct Addressing

The effective address of the operand is the location whose address is contained in the instruction. Depending on the instruction, the operand is either in the I/O or memory address space. Sixteen bits of direct address is the norm, but the DDIR IB and DDIR IW decoder directives allow 24-bit and 32-bit direct addresses, respectively.

Indexed Addressing

The effective address of the operand is the location computed by adding the two's-complement signed displacement contained in the instruction and the contents of the IX or IY register. Eight bits of index is the norm, but the DDIR IB and DDIR IW decoder directives allow 16-bit and 24-bit indexes, respectively.

Program Counter Relative Addressing

An 8-, 16- or 24-bit displacement contained in the instruction is added to the Program Counter to generate the effective address. This mode is available only for JUMP and CALL instructions.

Stack Pointer Relative Addressing

The effective address of the operand is the location computed by adding the two's-complement signed displacement contained in the instruction to the contents of the Stack Pointer. Eight bits of index is the norm, but the DDIR IB and DDIR IW decoder directives allow 16- and 24-bit indexes, respectively.

INSTRUCTION SET

The 380C instruction set is an expansion of the Z80 instruction set; the enhancements include support for additional addressing modes and a number of new instructions.

The 380C is op code compatible with the Z80 CPU and Z180 MPU. A Z80/Z180 program can be executed on the 380C without modification.

The instruction set is divided into groups by function; these are listed below. Consult the Z380 User Manual for additional details on the instruction set.

- 8-bit Load/Exchange
- 16/32-bit Load, Exchange, Swap and Push/Pop
- Block Transfers and Search
- 8-bit Arithmetic and Logical Operations
- 16/32-bit Arithmetic Operations
- 8-bit Bit Manipulation, Rotate and Shift
- 16-bit Rotates and Shifts
- Program Control
- I/O Operations
- CPU Control
- Decoder Directives

RESET

The 380C MPU within the Z80382 is placed in a dormant state when the $\overline{\text{RESET}}$ input is asserted. All its operations are terminated, including any interrupt, bus request or bus transaction that may be in progress. The IOCLK output goes Low on the next BUSCLK rising edge and enters into the BUSCLK/8 mode. The address and data buses are tristated, and the bus control signals are driven to their inactive states. The effect of a reset on the 380C and closely-related registers is summarized in Table 8. A reset also affects the on-chip peripherals of the Z80382. These effects are described in later sections of this document.

The $\overline{\text{RESET}}$ input may be asynchronous to BUSCLK, though it is sampled internally on falling edges of BUSCLK. For proper initialization of the 380C, V_{DD} must be within operating specifications and BUSCLK must be stable for more than five cycles with $\overline{\text{RESET}}$ held Low. The $\overline{\text{RESET}}$ input has a built-in Schmitt trigger buffer to facilitate power-on reset generation through an RC network.

Note: If a user system has devices external to the Z80382 that are clocked by IOCLK, these devices may require a $\overline{\text{RESET}}$ pulse width that spans over a number of IOCLK cycles (now at BUSCLK/8) for proper initialization.

The 380C fetches the first instruction 3.5 BUSCLK cycles after $\overline{\text{RESET}}$ is negated, provided such deassertion meets the proper setup and hold times with reference to the falling edge of BUSCLK. Again, with the proper setup and hold times being met, the first rising edge of IOCLK is 11.5 BUSCLK cycles after the deassertion of $\overline{\text{RESET}}$, preceded by a minimum of four BUSCLK cycles where IOCLK is at a Low.

Note: If $\overline{\text{BREQ}}$ is active when $\overline{\text{RESET}}$ is deasserted, the 380C relinquishes the bus instead of fetching its first instruction. IOCLK synchronization still takes place as previously described.

Table 8. Effect of Reset on 380C MPU and Related I/O Registers and Functions

Register or Function	Reset Value (Hex)	Comments
Program Counter	00000000	PCz, PC
Stack Pointer	00000000	SPz, SP
I	000000	Iz, I
R	00	
Select Register	00000000	Register bank 0 selected: AF, Main Bank, IX, IY NATIVE mode WORD mode Maskable interrupts disabled, in mode 0 Bus Request Lock - Off
A, F, BC, DE, HL, IX, IY Registers		Unaffected in all banks
I/O Bus Control Register 0	00	IOCLK = BUSCLK/8
Interrupt Enable Register	01	$\overline{\text{INT0}}$ enabled
Assigned Vector Base Register	00	
Trap and Break Register	00	
Memory Chip Selects and Waits	ROM Chip Select signal is enabled for (`00000000h` - `00007FFFh`) with seven Wait states. RAM Chip Select Low signal is enabled for (`00008000h` - `00FFFFFFh`) with seven T1, three T2 and seven T3 Waits. $\overline{\text{RAMCSH}}$ is inactive. $\overline{\text{MSIZE}}$ is forced for 8-bit accesses in both ranges.	

Table 8. Effect of Reset on 380C MPU and Related I/O Registers and Functions (Continued) (Continued) (Continued) (Continued)

Register or Function	Reset Value (Hex)	Comments
I/O Waits	External I/O read, write = 7 Waits RETI = 3 waits Interrupt daisy chain = 7 Waits	
DRAM Refresh Controller	Disabled	
Standby mode	Disabled	

LOW-POWER STANDBY MODE

The Z382 provides an optional STANDBY mode to minimize power consumption during system idle time. If this option is enabled, executing the SLEEP instruction stops the Z382's oscillator if it is in use. In any case this option stops clocking internal to the Z382, as well as at the BUSCLK and IOCLK outputs, except to PRT0 if it is enabled. The $\overline{\text{STNBY}}$ and $\overline{\text{HALT}}$ signals go Low, indicating That the Z80382 is entering the STANDBY mode. All Z382 operations are suspended, the bus control signals are driven inactive and the address bus is driven High. STANDBY mode can be exited by asserting any of the ($\overline{\text{RESET}}$, $\overline{\text{NMI}}$, $\overline{\text{INT[3:0]}}$ (if enabled), or optionally, $\overline{\text{BUSREQ}}$ inputs.

If STANDBY mode is not enabled, the Sleep instruction does not stop the Z382's oscillator if it is in use, but blocks clocking from internal modules, except PRT0 if it is enabled. In this case, $\overline{\text{STNBY}}$ (but not $\overline{\text{HALT}}$) goes Low to indicate the Z80382's status.

Timing for Entering STANDBY mode

Figure 5 illustrates the timing for entering STANDBY mode, in an example where IOCLK was programmed to be BUSCLK divided by 2. Note that clocking stops only after IOCLK is Low.

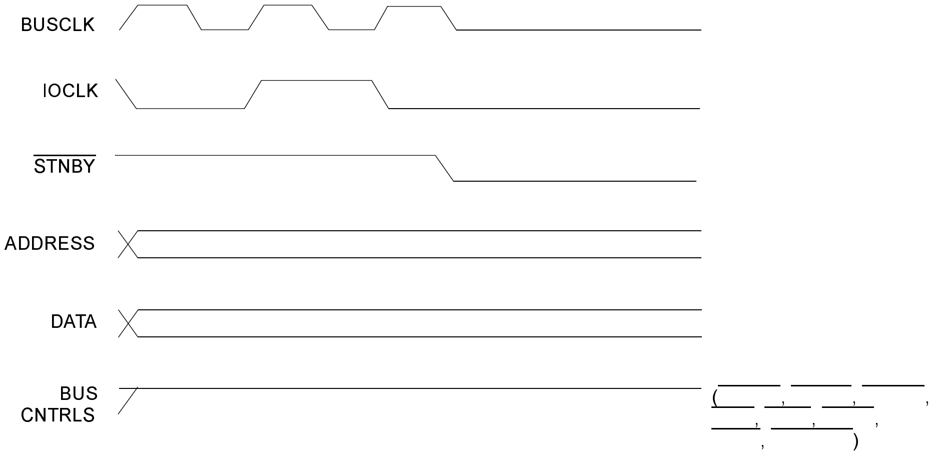


Figure 5. STANDBY Mode Entry Timing

STANDBY Mode Exit With Bus Request

Optionally, if the BRXT bit of the STANDBY mode Control Register (SMCR) was previously set, $\overline{\text{STNBY}}$ goes High when the $\overline{\text{BUSREQ}}$ input is asserted, allowing the external crystal oscillator that drives the Z382's CLK input to restart. A warm-up counter internal to the Z382 proceeds to count, for a duration long enough for the oscillator to stabilize, as selected by the WM bits in the SMCR. When the counter reaches its end-count, clocking resumes within the Z382 and at the BUSCLK and IOCLK outputs.

The Z382 relinquishes the system bus after clocking resumes, with the normal $\overline{\text{BUSREQ}}$, $\overline{\text{BUSACK}}$ handshake procedure. The Z382 regains the system bus when BUSREQ goes inactive, again going through a normal handshake procedure.

Note: Clocking continues, and the Z382 is at the HALT state.

Figure 6 illustrates the STANDBY mode exit with bus request timing.

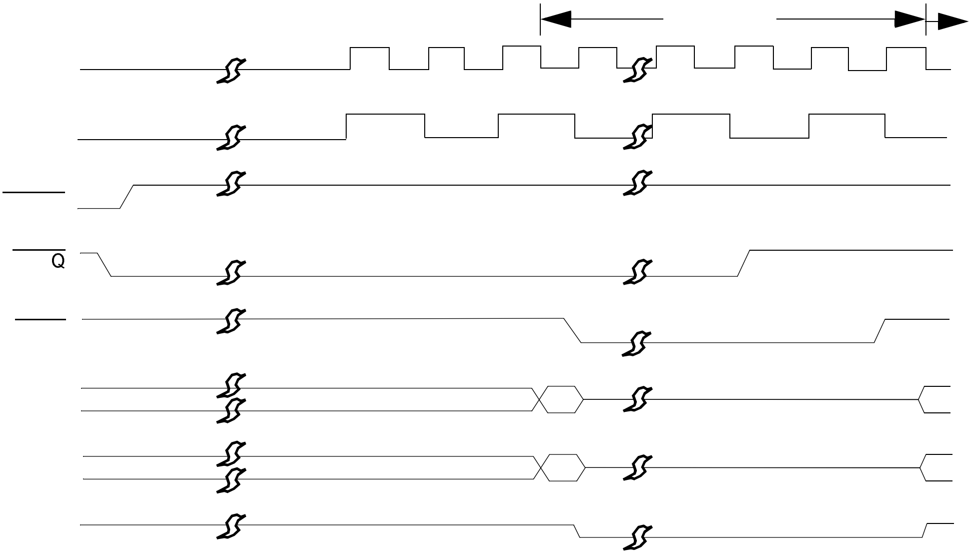


Figure 6. STANDBY Mode Exit with Bus Request Timing

STANDBY mode Exit With Reset

When RESET is asserted, STNBY goes High, allowing the external crystal oscillator that drives the Z382's CLKI input to restart. The RESET pulse provided must be of a duration long enough for oscillator stabilization. The Z382 exits STANDBY mode, and when RESET is deasserted, it goes through the normal reset timing to start instruction execution at address `00000000h`.

Note: Clocking resumes within the Z382 and at the BUSCLK and IOCLK outputs soon after $\overline{\text{RESET}}$ is asserted, when the crystal oscillator is not yet stabilized.

See Figure 7.

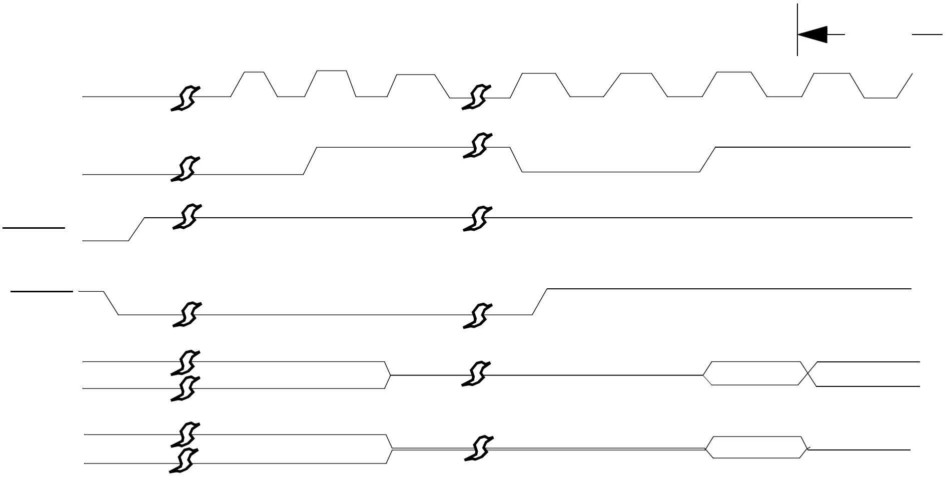


Figure 7. STANDBY Mode Exit with Reset Timing

Standby Mode Exit with External Interrupts

Exit Standby mode by asserting $\overline{\text{NMI}}$. Asserting the maskable interrupt inputs $\overline{\text{INT}}$[3:0] also exits Standby mode, if the global interrupt flag IEF1 was previously exited at logic 1, and for those requests that are individually enabled, as indicated in the Interrupt Enable register.

When excused conditions are met, $\overline{\text{STNBY}}$ goes to logic 1, allowing the external crystal oscillator that drives the Z382's CLK1 input to restart.

The warm-up counter internal to the Z382 counts, for a duration long enough for the oscillator to stabilize, as selected by the WM bits in the Standby Mode control register. When the counter reaches its end-count, clocking resumes within the Z382 and at the BUSCLK and IOCLK outputs. The Z382 performs an interrupt acknowledge procedure appropriate to the interrupt request that initiated the exit from Standby mode. See Figure 8

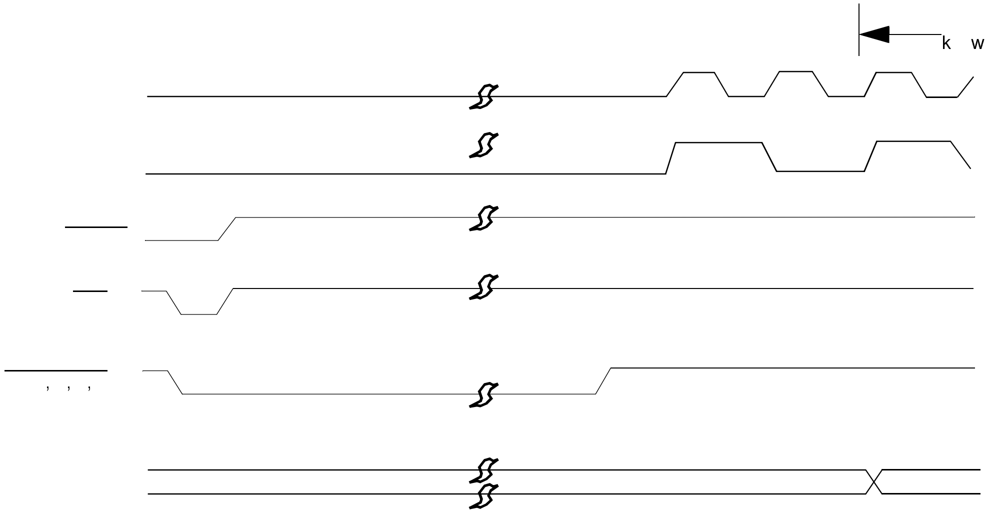


Figure 8. Standby Mode Exit with External Interrupt Timing

STANDBY Mode for On-Chip Crystal Oscillator

The previous discussions have focused on situations where a direct clock is supplied to the Z382's CLKI input. This type of clock may also be created by using an external crystal with the Z382’s oscillator circuit. In the case where the clock is created in this manner, all standby functions described earlier apply. Items worth noting are as follows:

- When STANDBY mode is entered, the feedback path for the on-chip oscillator is disabled, reducing power consumption.
- A user can select a warm-up time, appropriate for the crystal being used, by programming the WM bits in the STANDBY mode Control register.

INTERRUPT LOGIC

The Z382's interrupt structure provides compatibility with the existing Z80 and Z180 with the following exception: the undefined op code trap's occurrence is with respect to the Z380 instruction set, and its response is improved (vs. the Z180) to make trap handling easier. The 380C offers additional features to enhance flexibility in system design.

Of the five external interrupt inputs provided, $\overline{\text{NMI}}$ is a nonmaskable interrupt. The remaining inputs, $\overline{\text{INT[3:0]}}$, are asynchronous maskable interrupt requests.

In an Interrupt Acknowledge transaction, address outputs A[31:4] are driven to logic High. One output amongst A[3:0] is driven Low to indicate the maskable interrupt request being acknowledged. For example, if $\overline{\text{INT0}}$ is being acknowledged, A[3:1] are High and A0 is Low.

Interrupt modes 0 through 3 are supported for maskable interrupt request $\overline{\text{INT0}}$, which can be driven by external and on-chip sources. Modes 0, 1 and 2 have the same schemes as those in the Z80 and Z180. Mode 3 is similar to mode 2, except that 16-bit interrupt vectors are expected from the I/O devices. 8-bit and 16-bit I/O devices can be intermixed in this mode by having external pull-up resistors at the data bus signals D[15:8],for example.

Notes: The INTACK output of the Z380 has been omitted on the Z80382 for pinning reasons. A similar signal can be easily obtained by low-active-ANDing (positive-logic ORing) the $\overline{\text{M1}}$ and $\overline{\text{IORQ}}$ outputs.

> The Z80382 does not support RETI decoding by Z80 peripherals (PIO, S10, and CTC). It does support Z80-type interrupt daisy-chaining by devices that include explicit clearing of IUS (for example, SCC).

The external maskable interrupt requests INT[3:1], as well as the less complex on-chip peripherals (PRTs, ASCls, and CSI/O) are handled in an assigned interrupt vectors mode. INT[3:1] can be used as low- or high-active level-sensitive inputs, or as falling- or rising-edge triggered inputs.

The Z382 can operate in either the NATIVE or EXTENDED mode. In NATIVE mode, PUSHing and POPing of the stack to save and retrieve interrupted PC values in interrupt handling are done as 16-bit values, and the stack pointer rolls over at the 64KB boundary. In EXTENDED mode, the PC PUSHes and POPs are done as 32-bit values, and the stack pointer rolls over at the 4GB memory space boundary. (Because the Z80382 only pins out 24 address lines, this rollover appears to occur at the 16MB boundary externally). The 380C provides an Interrupt Register Extension, whose contents are always output as the address bus signals A[31:16], when fetching the starting addresses of service routines from memory in interrupt modes 2, 3 and the assigned vectors mode. In NATIVE mode, such fetches are automatically processed as 16-bit values and in EXTENDED mode, in 32-bit values. These starting addresses must be even-aligned in memory locations. That is, their least significant bytes must have addresses with A0 = 0.

Interrupt Priority Ranking

The 380C assigns a fixed priority ranking to handle its major categories of interrupt sources, as described in Table 9:

Table 9. Interrupt Priority Ranking

Priority	Interrupt Sources
Highest	Trap (undefined Opcode)
	$\overline{\text{NMI}}$
	$\overline{\text{INT0}}$ (including DMAs, Mimic, and HDLC controllers)
	$\overline{\text{1NT1}}$
	$\overline{\text{INT2}}$
	GCI/SCIT
	PRT0
	PRT1
	CSI/O
	ASCI0
	ASCI1
	Plug and Play ISA or PCMCIA
	I/O Mailbox
Lowest	$\overline{\text{INT3}}$

INT0 Peripherals

Those on-chip peripherals capable of generating their own interrupt vectors, including the Mimic, DMAs, and HDLC controllers, have their interrupt requests logically ORed with the external $\overline{\text{INT0}}$ pin to produce the INT0 signal presented to the 380C processor. These interrupt sources are consecutive in the INT0 daisy-chain, but their relative priority can be programmed in the System Configuration Register. Their priority relative to external INT0 sources is controlled by how the Z80382's IEI and IEO pins are connected.

Trap Interrupt

The 380C generates a trap when an undefined op code is encountered. The trap is enabled immediately after reset, and it is not maskable. This feature can be used to increase software reliability or to implement extended instructions. An undefined op code can be fetched from the instruction stream, or it can be returned as a vector in an interrupt acknowledge transaction in Interrupt mode 0. When a trap occurs, the 380C operates as follows.

1. The TF or TV bit in the Assigned Vectors Base and Trap Register goes active, to indicate the source of the undefined op code.
2. If the undefined op code was fetched from the instruction stream, the starting address of the instruction causing the trap is pushed onto the stack. (The starting address of a decoder directive preceding an instruction encoding is considered the starting address of the instruction.)
3. If the undefined op code was a returned interrupt vector (in interrupt mode 0), the interrupted PC value is pushed onto the stack.
4. IEF1 and IEF2 are cleared.
5. The 380C commences to fetch and execute instructions from address `00000000h`.
6. Instruction execution resumes at address 0, similar to the occurrence of a reset. Testing the TF and TV bits in the Assigned Vectors Base and Trap Register distinguishes the two events. Even if trap handling is not in place, repeated restarts from address 0 is an indicator of possible illegal instructions at system debugging.

Nonmaskable Interrupt

The nonmaskable interrupt input $\overline{\text{NMI}}$ is edge-sensitive, with the Z80382 internally latching the occurrence of its falling edge. When the latched version of $\overline{\text{NMI}}$ is recognized, the following operations are performed.

1. The interrupted PC (Program Counter) value is pushed onto the stack.
2. The state of IEF1 is copied to IEF2, then IEF1 is cleared.
3. The Z382 commences to fetch and execute instructions from address `00000066h`.

Maskable Interrupt $\overline{\text{INT0}}$

Interrupt Mode 0 Response

During the interrupt acknowledge transaction, the external I/O device being acknowledged is expected to output a value onto D[15:8]. The 380C interprets the value as an instruction op code, which is usually one of the single-byte Restart (RST) instructions that pushes the interrupted PC (Program Counter) value onto the stack and resumes execution at a fixed memory location. However, the 380C generates multiple transactions to capture vectors that form a multi-byte instruction. IEF1 and IEF2 are cleared, disabling all further maskable interrupt requests.

Note: Unlike the other interrupt responses, the PC is not automatically PUSHed onto the stack. A trap occurs if an undefined op code is supplied by the I/O device as a vector.

Interrupt Mode 1 Response

An interrupt acknowledge transaction is generated, during which the data bus contents are ignored by the 380C. The interrupted PC value is PUSHed onto the stack. IEF1 and IEF2 are cleared to disable further maskable interrupt requests. Instruction fetching and execution restarts at memory location `00000038h`.

Interrupt Mode 2 Response

During the interrupt acknowledge transaction, the external I/O device being acknowledged is expected to output a vector onto D[15:8]. The interrupted PC value is PUSHed onto the stack and IEF1 and IEF2 are cleared to disable further maskable interrupt requests. The 380C then reads an entry from a table residing in memory and loads it into the PC to resume execution.

In NATIVE mode, as on the Z380, the address of the table entry is composed of the I Extend contents as A[31:16], the I Register contents as A[15:8] and the vector supplied by the I/O device as A[7:0].

In EXTENDED mode, unlike the Z380, the address of the table entry is composed of I Extend as A[31:16], the seven most significant bits of the I register as A[15:9], the vector supplied by the I/O device as A[8-1], and a Low on A0.

The table entry contains the starting address of the interrupt service routine designed for the I/O device being acknowledged. The table, composed of starting addresses for all the INTERRUPT mode 2 service routines, can be referred to as the interrupt mode two vector table. Each table entry must be word-sized if the 380C is in the NATIVE mode and long word-sized if in the EXTENDED mode. In either case it is even-aligned (least significant byte with address A0 = 0).

Interrupt Mode 3 Response

INTERRUPT mode 3 is similar to mode 2 except that a 16-bit vector is expected to be placed on the data bus D[15:0] by the I/O device during the interrupt acknowledge transaction. The interrupted PC is PUSHed onto the stack. IEF1 and IEF2 are cleared to disable further maskable interrupt requests. The starting address of the service routine is fetched and loaded into the PC to resume execution from the memory location with an address composed of the I Extend contents as A[31:16] and the vector supplied by the I/O device as A[15:0]. Again the starting address of the service routine is word-sized if the 380C is in the NATIVE mode and long word-sized if in the EXTENDED mode, in either case even-aligned.

Assigned Interrupt Vectors Mode (INT[3:1], PRTs, CSI/O, ASCIs)

When the 380C recognizes INT[3:1], or a request from an on-chip peripheral that cannot supply an interrupt vector (a PRT, CSI/O or ASCI), it generates an Interrupt Acknowledge transaction which is different from that for $\overline{\text{INT0}}$. This Interrupt Acknowledge transaction has $\overline{\text{IORQ}}$ active for external monitoring purposes, but $\overline{\text{M1}}$, $\overline{\text{IORD}}$, and $\overline{\text{IOWR}}$ inactive so as not to stimulate external devices. The interrupted PC value is PUSHed onto the stack. IEF1 and IEF2 are cleared, disabling further maskable interrupt requests.

The starting address of an interrupt service routine is fetched froth a table entry and loaded into the PC to resume execution. The address of the table entry is composed of the I Extend contents as A[31:16], the AB bits of the Assigned Vectors Base Register as A[15:9] and an assigned interrupt vector specific to the request being recognized as A[8:0]. The assigned vectors are listed in Table 10:

Table 10. Assigned Interrupt Vectors

Interrupt Source	Assigned Interrupt Vector (9 bits)
INT1	000h
INT2	004h
Reserved	008h
GCI/SCIT	00Ch
PRT0	010h
PRT1	014h
CSI/O	018h

Table 10. Assigned Interrupt Vectors (Continued)

Interrupt Source	Assigned Interrupt Vector (9 bits)
ASCI0	01Ch
ASCI1	020h
Plug-and-Play or PCMCIA	024h
I/O Mailbox	028h
INT3	02Ch

The table entries represented by these vectors are spaced at four byte intervals. In NATIVE mode, the lower two bytes of the data in the table are used as the address of the ISR, while in EXTENDED mode, all four bytes are used.

RETI Instruction

The original Z80 family I/O devices (PIO, SIO, CTC) are designed to monitor Return from Interrupt op codes in the instruction stream, signifying the end of the current interrupt service routine. On the Z80382, the $\overline{\text{M1}}$ signal is active during all instruction fetch transactions. Because the 380C may not execute an RETI that it fetches, and because it supports a 16-bit data bus, only half of which is visible to an 8-bit peripheral, the 382 does not support RETI decoding by the PIO, SIO, and CTC.

BYTE ORDERING

As noted in the pin descriptions ("Pin Descriptions" on page 8), the names of the D[15:8] and D[7:0] pins, as well of those of the $\overline{\text{BHEN}}$ and $\overline{\text{BLEN}}$ pins, have been exchanged on the Z80382 as compared to the Z80380. This action brings the Z80382 into line with historical Z80 and industry (such as, Intel) practice, and avoids comments that arose for the Z380, such as *the less significant byte is transferred on D[15:8].*

Table 11 describes all possible memory transactions with respect to the data size requested by software, and memory width implemented by hardware.

Table 11. Z80382 Memory Transaction Characteristics

Software Request			382 Bus Cycle(s)				Response	Data Transferred On	Notes
Dir	**Size**	**A0**	**A23-1**	**A0**	$\overline{\text{BHEN}}$	$\overline{\text{BLEN}}$	$\overline{\text{MSIZE}}$		
R	8	0	AA	L	H	L	X	D [7:0]	
R	8	1	AA	H	L	H	H	D[15:8]	
R	8	1	AA	H	L	H	L	D[7:0]	Note[1]
W	8	0	AA	L	H	L	X	D[7:0]	
W	8	1	AA	H	L	H	H	D[15:8]	Note[2]
W	8	1	AA	H	L	H	L	D[7:0]	Note[2]
R	16	0	AA	L	L	L	H	D[15:0]	
R	16	0	AA	L	L	L	L	D[7:0]	Note[3]
			AA	H	H	L	L	D[7:0]	
W	16	0	AA	L	L	L	H	D[15:0]	
W	16	0	AA	L	L	L	L	D[7:0]	Note[3]
			AA	H	H	L	L	D [7:0]	
R	16	1	AA	H	L	H	H	D[15:8]	Note[4]
			++AA	L	H	L	H	D[7:0]	
R	16	1	AA	H	L	H	L	D[7:0]	Note[4,1]
			++AA	L	H	L	L	D[7:0]	

Table 11. Z80382 Memory Transaction Characteristics (Continued)

Software Request			382 Bus Cycle(s)				Response	Data Transferred On	Notes
Dir	Size	A0	A23-1	A0	$\overline{\text{BHEN}}$	$\overline{\text{BLEN}}$	$\overline{\text{MSIZE}}$		
W	16	1	AA	H	L	H	H	D[15:81	Note[4,2]
			++AA	L	H	L	H	D[7:0]	
W	16	1	AA	H	L	H	L	D[7:0]	Note[4,2]
			++AA	L	H	L	L	D[7:0]	
R	32	0	AA	L	L	L	H	D[15:0]	
			++AA	L	L	L	H	D[15:0]	
R	32	0	AA	L	L	L	L	D[7:0]	Note[3]
			AA	H	H	L	L	D[7:0]	
			++AA	L	L	L	L	D[7:0]	Note[3]
			AA	H	H	L	L	D[7:0]	
W	32	0	AA	L	L	L	H	D[15:0]	
			++AA	L	L	L	H	D[15:0]	
W	32	0	AA	L	L	L	L	D[7:0]	Note[3]
			AA	H	H	L	L	D[7:0]	
			++AA	L	L	L	L	D[7:0]	
			AA	H	H	L	L	D[7:0]	
R	32	1	AA	H	L	H	H	D[15:8]	Note[4]
			++AA	L	L	L	H	D[15:0]	
			++AA	L	H	L	H	D[7:0]	
R	32	1	AA	H	L	H	L	D[7:0]	Note[4,1]
			++AA	L	L	L	L	D[7:0]	Note[3]

Table 11. Z80382 Memory Transaction Characteristics (Continued)

Software Request			382 Bus Cycle(s)				Response	Data Transferred On	Notes
Dir	**Size**	**A0**	**A23-1**	**A0**	**$\overline{\text{BHEN}}$**	**$\overline{\text{BLEN}}$**	**$\overline{\text{MSIZE}}$**		
			AA	H	L	H	L	D[7:0]	Note[1]
			++AA	L	H	L	L	D[7:0]	
W	32	1	AA	H	L	H	H	D[15:8]	Note[4,2]
			++AA	L	L	L	H	D[15:0]	
			++AA	L	H	L	H	D[7:0]	
W	32	1	AA	H	L	H	L	D[7:0]	Note[4,2]
			++AA	L	L	L	L	D[7:0]	Note[3]
			AA	H	L	H	L	D[7:0]	Note[2]
			++AA	L	H	L	L	D[7:0]	Note[5]

NOTES:

1. "++AA" indicates that A[23:1] are incremented from the preceding cycle.
2. The 382 takes the requested odd-addressed byte from D[7:0] rather than D[15:8], because $\overline{\text{MSIZE}}$ Low indicates that it has selected an 8-bit memory.
3. When writing a byte to an odd address, the Z80382 drives it onto D[15:8] and D[7:0], and pays no attention to the $\overline{\text{MSIZE}}$ response
4. The Z80382 tries to transfer 16 bits since the current address is even, but comes back for a subsequent byte cycle because $\overline{\text{MSIZE}}$ Low indicates that it has selected an 8-bit memory.
5. The Z80382 starts the requested multi-byte operation with a byte cycle because the address is odd.

Basic Device Configuration

OVERVIEW

The Z80382 offers a number of on-chip functions which must be configured by software to accommodate the user's system requirements.

The configuration requirements can be broadly typed into two classes. The first class configures the pinout of the Z80382 and the 380C core CPU. Items covered by this class include pin usage, I/O drive characteristics, timing, I/O and memory chip selects, and DRAM refresh and interrupts. The registers used for configuring these items are described in this chapter.

The second class pertains to the on-chip functional units such as ASCIs, DMAs and HDLC controllers, all of which include configuration options. The configuration of these devices is described in the sections of this User Manual which describe those functional units.

CHIP VERSION

The Chip Version ID register is a read-only register which identifies the specific version of the Z380 CPU-based Superintegration device utilized in the Z80382 (called *380C* in this manual).

The current value contained in this register is 00H. This value is subject to change as improved versions of the CPU core are implemented.

MULTIFUNCTION PIN USAGE

These registers relate primarily to how the multifunction I/O device pins are used:

- System Configuration Register
- Pin Multiplexing Register
- Chip Version ID Register (read only)

Certain pins are multiplexed automatically based on register bits in their associated functions:

- In an ISA bus or PCMCIA application (MainMux = 1x), if the Transmit Enable and/or Receive Enable bit(s) in ASCI0 is (are) set, and the ASonHD0 bit isn't set, then pins 51-52 are TXA0 and RXA0, otherwise they are PB7-6. Similarly, if the Transmit Enable and/or Receive Enable bit(s) in ASCII is (are) set, and the ASonHD1 bit is not set, then pins 54-55 are TXA1 and RXA1; otherwise they are PB4-3.
- If bit 6 of Memory Mode Register 2 is 1, then pin 44 is PC7, otherwise it is RAMCSH
- If IOCS1 is enabled, that is the function of pin 45; otherwise it is PC6.
- If IOCS2 is enabled, that is the function of pin 46; otherwise it is TOUT (the pulse output of PRT1).
- If the Tx and Rx Configuration fields of HDLC channel 2 are '000', pins 105-108 act as GCI/SCIT signals; otherwise they act as separate pins for HDLC channel 2.

PROGRAMMABLE LOW-NOISE DRIVERS

To help reduce noise generated by the output switching of the Z382, selected outputs can be placed in a reduced drive configuration. When a pin is placed in low noise mode, its drive is reduced to a third of its normal output drive current. This reduction decreases the slew rate of the driver, which reduces current spikes induced onto the power bussing of the Z382. The Output Drive Control register is used to select this capability.

TIMING AND I/O BUS CONTROL

The Z382's I/O bus can be programmed to run at a slower rate than its memory bus. The Z80382 differs from the Z80380 in that it does not provide heartbeat transactions that emulate a Z80 instruction fetch cycle.

The Z382 is designed to interface with external I/O devices for the Z80 product family by supplying four I/O bus control signals: $\overline{\text{M1}}$, $\overline{\text{IORQ}}$, $\overline{\text{IORD}}$, and $\overline{\text{IOWR}}$. In addition, the Z382 supplies an IOCLK that is a divided down version of its BUSCLK.

The following registers are used to program various timing and I/O-related parameters:

- Clock Control Register
- I/O Waits Register

I/O CHIP SELECTS

Two I/O chip selects, $\overline{\text{IOCS1}}$ and $\overline{\text{IOCS2}}$, are provided to support I/O access to external peripherals. These chip selects are asserted Low when the current 380C address matches the values programmed in the IOCS registers. The number of bits compared can be selected to provide I/O window sizes from 8 to 512 bytes.

These outputs may be asserted during both memory and I/O cycles. The I/O Chip Selects are not asserted during INTACK cycles.

The following registers are used to program the base address and window size of the I/O Chip Selects.

- I/O Chip Select 1 Address High Register
- I/O Chip Select 1 Address Low Register
- I/O Chip Select 2 Address High Register
- I/O Chip Select 2 Address Low Register

RAM AND ROM CHIP SELECTS

Three memory chip select outputs are provided: $\overline{\text{ROMCS}}$, $\overline{\text{RAMCSL}}$, and $\overline{\text{RAMCSH}}$. They support both 8- and 16-bit memories, and are asserted for a selected address range (4KB to 8MB) during both memory and I/O cycles.

Wait state generation can be specified separately for ROM and RAM. Unlike Chip Select and MSIZE signalling, Wait state generation occurs only during memory cycles.

For the selected ROM and/or RAM range, the $\overline{\text{MSIZE}}$ pin can be programmed to be forced Low in an open-drain fashion when the address is in the programmed range, forcing 8-bit accesses in one or both ranges. When $\overline{\text{MSIZE}}$ is forced for 8-bit RAM in this way, $\overline{\text{RAMCSL}}$ is asserted for all cycles in the selected address range, and the $\overline{\text{RAMCSH}}$ pin assumes its alternate use as port pin PC7. When $\overline{\text{MSIZE}}$ is not forced for 8-bit RAM, $\overline{\text{RAMCSL}}$ is qualified by $\overline{\text{BLEN}}$, and $\overline{\text{RAMCSH}}$ acts as a chip select output pin and is qualified by $\overline{\text{BHEN}}$.

If the address ranges specified for RAM and ROM overlap, $\overline{\text{ROMCS}}$ is asserted in the region of overlap, $\overline{\text{RAMCSL}}$ and $\overline{\text{RAMCSH}}$ are not.

The following registers are used to program the base address, window size and other parameters of the RAM and ROM Chip Selects.

- RAM Address High Register
- RAM Address Low Register
- ROM Address High Register
- ROM Address Low Register
- Memory Mode Register 1
- Memory Mode Register 2

Memory Chip Selects and Reset

On power up, ROMCS is active for addresses with A[23:15] = 000H, RAMCSL is active for all other addresses, and MSIZE is forced for 8-bit accesses in both ranges. Such a map is desirable in case Reset is closely followed by a Non-Maskable Interrupt, in which case the return address must be stacked in RAM at addresses such as 00FFFxh (the Stack Pointer resets to 000000h, but does not borrow between A15 - A16 in NATIVE mode).

If ROM is 16 bits wide, ROMCS should be connected to the CE inputs of both devices, and the first instructions of the Reset sequence should include XOR A twice, followed by OUT (MMR2), A. These may be preceded by PUSH AF twice for a debug monitor that attempts to preserve register values across a Reset, but by no other instructions. The OUT (MMR2), A must be followed by either:

- A JR instruction, the first and second bytes of which have the same value, such as 1818, or
- Eight NOP instructions.

Either of these choices allow 16-bit access to become effective. These instructions must in turn be followed by LD A, value and OUT0 (MMR2), A, to set MMR2 to its correct value.

DRAM Refresh

The Z80382 is capable of providing refresh transactions to dynamic memories that have internal refresh address counters. A user can select how often refresh requests should be made to the Z80382's external interface logic, as well as the burst size (number of refresh transactions) for each request iteration. The external interface logic grants these requests by performing refresh transactions with CAS-before-RAS timing on the TREFR, TREFA and TREFC bus control signals. In these transactions, BHEN, BLEN and the RAM chip select signal(s) are driven active to facilitate refreshing all the DRAM modules at the same time. A user can also specify the T1, T2 and T3 Waits to be inserted.

Note: The Z382 cannot provide refresh transactions when it relinquishes the system bus, with its $\overline{\text{BREQ}}$ input active.

In that situation, the number of missed refresh requests are accumulated in a counter, and the missed refresh transactions are performed when the Z382 regains the system bus.

The following registers are used to program the DRAM refresh characteristics.

- Refresh 0 Register
- Refresh 1 Register
- Refresh 2 Register
- Refresh Waits Register

LOW-POWER STANDBY MODE

The Z382 provides an optional STANDBY mode to minimize power consumption during system idle time. If this option is enabled, executing the SLEEP instruction stops the Z382's oscillator if it is in use. In any case this option stops clocking internal to the Z382, as well as at the BUSCLK and IOCLK outputs, except to PRTO if it is enabled. The $\overline{\text{STNBY}}$ and $\overline{\text{HALT}}$ signals go Low, indicating the Z382 is entering the STANDBY mode. All Z382 operations are suspended except for those that do not depend on BUSCLK, such as asynchronous interrupt requests. The bus control signals are driven inactive and the address bus is driven High. STANDBY mode can be exited by asserting any of the $\overline{\text{RESET}}$, $\overline{\text{NMI}}$, $\overline{\text{INT[3:0]}}$ (if enabled), or optionally, $\overline{\text{BREQ}}$ inputs.

If STANDBY mode is not enabled, the SLEEP instruction does not stop the Z382's oscillator if it is in use, but blocks clocking from internal modules, except PRTO if it is enabled. In this case, $\overline{\text{STNBY}}$ (but not $\overline{\text{HALT}}$) goes Low to indicate the Z382's status.

The STANDBY Mode Control register is used to program parameters pertinent to the STANDBY mode:

INTERRUPT CHARACTERISTICS

Interrupt characteristics which require configuration include enabling/disabling specific interrupts, interrupt modes for INT[3:0], and setting the Assigned Interrupts base address vector.

The registers pertaining to these functions are:

- Interrupt Enable Register
- Assigned Vectors Base Register
- Trap and Break Register
- INT[3:1] Control Register

REGISTER DESCRIPTIONS

Register Summary

Table 12 summarizes the registers concerning the basic configuration of the Z80382. This table also notes if the identical register is present in the Z380 device and, if so, its equivalent address:

Table 12. Register Summary

Register Name	380C Address	Z380 Address
Assigned Vectors Base Reg.	0018h	0018h
Trap and Break Register	0019h	0019h
I/O Waits Register	001Eh	000Eh
Refresh Waits Register	001Fh	000Fh
Clock Control Register	0021h	0011h
Refresh Register 0	0023h	0013h
Refresh Register 1	0024h	0014h
Refresh Register 2	0025h	0015h
Standby Mode Control Reg.	0026h	0016h
Interrupt Enable Register	0027h	0017h
Chip Version ID Register	0020h	00FFh
IOCS1 Low Register	002Ah	—
IOCS1 High Register	0028h	—
IOCS2 Low Register	002Ch	—
IOCS2 High Register	002Dh	—
RAM Low Register	002Eh	—
RAM High Register	002Fh	—
ROM Low Register	0030h	—
ROM High Register	0031h	—
Memory Mode Register 1	0032h	—
Memory Mode Register 2	00D3h	—
System Configuration Register	0036h	—

Table 12. Register Summary (Continued)

Register Name	380C Address	Z380 Address
Pin Multiplexing Register	0037h	—
Output Drive Control Register	003Ah	—
INT[3:1] Control Register	003Bh	—

Details on each register are provided in the sections which follow. In the descriptions:

- XXXXh = the register's address in the 380C I/O space, in hexadecimal
- DEF = default value after RESET
- x = indicates that the default value of the bit is indeterminate

Chip Version ID Register (CVIDR)

Chip Version ID Register **0020h**

BIT	7	6	5	4	3	2	1	0
R	Chip Version ID							
W	No Function							
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W	Description
7:0	Chip Version ID	R	Read-only value which provides the version number of the Z380-compatible Superintegration core used in the Z382. The current value is 00H, but is subject to change on future revisions of the product

System Configuration Register (SYSCON)

System Configuration Register **0036h**

BIT	7	6	5	4	3	2	1	0
R W	Mainmux		Refresh	IntIOout	All Page I/O	INT0 PRI		
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W		Description
7:6	Main Multiplexer	R/W		This field controls the functions of pins 60 - 92. PIN 60 - 92 FUNCTION
			00	Pins 60 - 92 can be used for the ASCIs, CSI/O, and ports A and D, except that the full-time outputs among these signals (TXA0, TXA1, RTS0, TxS) are disabled. This is the reset state. Leave it this way, to save power and noise, if your application does not need any of these four outputs.
			01	Pins 60 - 92 are used for the ASCIs, CSI/O, and ports A and D, including the TXA0 TXA1, RTS0, and TxS outputs.
			10	Pins 60 - 92 are connected to the ISA bus of a Host PC.
			11	Pins 60 - 92 are connected to a Host's PCMCIA bus.

Bit(s)	Function	R/W	Description
5	Refresh Outputs	R/W	This field controls pins 110 -112. A 0 selects $\overline{\text{DCD0}}$, $\overline{\text{CTS0}}$, $\overline{\text{CTS1}}$, which is the Reset state. The ASCI auto-enable signals are only taken from these pins in a Host application. A 1 selects TREFA, TREFC, and TREFR. Select this option if DRAM control signals are necessary in the application. If an ISA or PCMCIA application uses DRAM, it cannot use ASCI auto-enable handshaking.
4	Internal I/O Data Out	R/W	If this bit is 1, the D[15:0] pins are driven as outputs from the Z382 during reads from on-chip I/O devices. Leaving this bit 0 reduces power consumption, noise, and EMI/RFI to some extent.
3	All Page I/O Decoding	R/W	A 0 in this bit restricts the I/O address decoding of the Mimic and Parallel Ports to *page 0*: A[15:8] must be zero to access these features. In this case, software must use instructions such as IN0 and OUT0 for Mimic and Port programming. A 1 in this bit makes the address decoding for the Mimic and Ports disregard address lines above A7, as on the Z80182, 187, and 189, so that these devices are replicated in each 256-byte page of I/O space. Select this option if the user's application includes code ported from these devices, that uses IN and OUT instructions.

Bit(s)	Function	R/W	Description	
2:0	Interrupt 0 Priority	R/W		This field controls the relative interrupt priority of the Mimic, HDLC channels and DMA channels on the INT0 daisy chain:
			2:0	INT0 DAISY CHAIN PRIORITY
			000	IEI devices, Mimic, DMA, HDLC, IEO devices (lowest)
			001	IEI devices, Mimic, HDLC, DMA, IEO devices (lowest)
			01X	IEI devices, HDLC, Mimic, DMA, IEO devices (lowest)
			100	IEI devices, DMA, Mimic, HDLC, IEO devices (lowest)
			101	IEI devices, DMA, HDLC, Mimic, IEO devices (lowest)
			11X	IEI devices, HDLC, DMA, Mimic, IEO devices (lowest)

Pin Multiplexing Register (PINMUX)

Pin Multiplexing Register **0037h**

BIT	7	6	5	4	3	2	1	0
R W	ASONHD 1	ASONHD 0	IEI	IEO	CKA1	CKA0	CSI/O	Pin 109
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W	Description
7	ASCI0 on HDLC0 Pins	R/W	If this bit is 1, the pins normally used for HDLC 0 are used for ASCI0 signals instead: TXA0 for TxD0, RXA0 for RxD0, CKA0 for TxC0, PC0 for RxC0, and RTS0 for TXEN0. This bit is active in all applications.
6	ASCI1 on HDLC1 Pins	R/W	If this bit is 1, the pins normally used for HDLC 1 are used for ASCI1 signals instead: TXA1 for TxD1, RXA1 for RxD1, CKA1 for TxC1, PC1 for RxC1, and RTS1 for TXEN1. This bit is active in all applications.
5	IEI Pin Function	R/W	If this bit is 1, pin 47 is IEI; otherwise it is PC5.
4	IEO Pin Function	R/W	If this bit is 1, pin 48 is IEO; otherwise it is PC4.
3	CKA1 Pin Function	R/W	If this bit is 1 in a Host application (MainMux = 1 X), pin 49 is CKA1; otherwise it is PC3.
2	CKA0 Pin Function	R/W	If this bit is 1 in a Host application (MainMux = 1X), pin 53 is CKA0; otherwise it is PB5.

Bit(s)	Function	R/W	Description
1	CSI/O Pins	R/W	If this bit is 1 in a Host application (MainMux = 1X), pins 56-58 are TXS, RXS, and CKS, else they are PB[2:0].
0	Pin 109 Function	R/W	If this bit is 1 when HDLC channel 2's Tx Configuration field is 010, pin 109 carries the TXEN2 function. If this bit is 1 when HDLC channel 2's Tx Configuration field has any other value, pin 109 has the TOUT function. If this bit is 0, as it after a Reset, pin 109 is PC2.

Output Drive Control Register (ODCR)

Output Drive Control Register **003A**

BIT	7	6	5	4	3	2	1	0
R W	Disable Weak Latches PB[7:0]	Reserved	Low Noise Ports	Low Noise ISA/ PCMCIA	Low Noise GCI/ HDLC	Low Noise Control	Low Noise Address	Low Noise Data
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W	Description
7	Disable Weak Latches PB7-0	R/W	Bit 7 can be written as 1 to disable the weak latch function on pins PB[7:0]. This allows these pins to electrically “float” for analog-type applications.
6	Reserved	R/W	Reserved. Program to 0.
5	Low Noise Ports	R/W	This bit controls the drive strength on PB[7:0] and PC[7:0] in all modes, and on PD[7:0] and HA9/CKS which can only be outputs in non-ISA, non-PCMCIA mode.
4	Low Noise ISA/PCMCIA	R/W	This bit controls the drive strength on the majority of the pins used for the ISA and PCMCIA interfaces, including: HDRQ1/PCRESET/CKA0 HINT1/PCIRQ/TXA1 HDRQ0/PCCE1/TXA0 HD[7:0/PA[7:0] HINT2/STSCHG/RTS0 TXS/HDOEN/INPACK
3	Low Noise GCI/HDLC	R/W	This bit controls the drive strength on the pins used for the serial interfaces, including: DD/TxD2 $\overline{\text{TxEN1}}$/$\overline{\text{RTS1}}$ DU/TxC2 TxD0/TXA0 TxEN2/TOUT/PC2 TxC0/CKA0 TxD1/TXA1, TxC1/CKA1 $\overline{\text{TxEN0}}$/$\overline{\text{RTS0}}$

Bit(s)	Function	R/W	Description
2	Low Noise Control	R/W	This bit controls the drive strength on the processor control signals, including: $\overline{BUSACK}$ $\overline{IORQ}$ $\overline{MRD}$ IOCLK $\overline{MWR}$ TREFR $\overline{IORD}$ TREFA $\overline{IOWR}$ TREFC $\overline{HALT}$ $\overline{ROMCS}$ $\overline{STNBY}$ $\overline{RAMCSL}$ $\overline{BLEN}$ $\overline{IOCS2}$/TOUT $\overline{BHEN}$ $\overline{RAMCSH}$/PC7 $\overline{M1}$ $\overline{IOCS1}$/PC6 $\overline{BUSCLK}$
1	Low Noise Address	R/W	This bit controls the drive strength on A[23:0]
0	Low Noise Data	R/W	This bit controls the drive strength on D[15:0].

Clock Control Register (CCR)

Clock Control Register **0021h**

BIT	7	6	5	4	3	2	1	0
R W	Main Clock Rate		Block CLK0	Reserved		I/O Clock Rate		
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W	Description
7:6	Main Clock Rate	R/W	This field controls how BUSCLK is derived from the input clock on CLKI 7:6 BUSCLK = 00 CLKI/2 (Reset State) 01 CLKI 10 CLKI x 2 11 Reserved
5	Block CLKO Output	R/W	This bit resets to 0, in which state the CLKO pin is driven from CLKI, for use with an external crystal. If an externally derived clock (from an external oscillator or other source) is connected to CLKI, software should set this bit to 1 to save power and noise by not driving CLKO.
4:3	Reserved	R/W	Reserved. Program as 0.

Bit(s)	Function	R/W		Description
2:0	I/O Clock Rate	R/W		BUSCLK is divided to produce IOCLK as described below. **Note:** If a clock divide rate of 1 is specified, BUSCLK must be used to connect to I/O devices that require a clock input, since the Z80382 outputs a constant High on IOCLK in this case.
			2:0	IOCLK = BUSCLK DIVIDED BY:
			000	8
			001	2
			010	4
			011	6
			1xx	IOCLK is High. See note above.

I/O Waits Register (IOWR)

I/O WAITS REGISTER **001Eh**

BIT	7	6	5	4	3	2	1	0
R W	I/O Waits			Reserved		Interrupt Daisy Chain Waits		
DEF	1	1	1	0	0	1	1	1

Bit(s)	Function	R/W	Description
7:5	I/O Cycle Wait States	R/W	This binary field defines up to seven wait states to be inserted in external I/O read and write transactions, and at the latter portions of interrupt transactions to capture interrupt vectors. 000 disables I/O waits.
4:3	Reserved	R/W	Reserved. Program as 0.
2:0	Interrupt Daisy Chain Wait	R/W	This binary field defines up to seven wait states to be inserted at the early portions of interrupt acknowledge transactions, to allow the interrupt daisy chain through on-chip and possibly external I/O devices to settle.

STANDBY Mode Control Register (SMCR)

STANDBY Mode Control Register **0026h**

BIT	7	6	5	4	3	2	1	0
R W	Enable STANDBY Mode	BUSREQ Exits STANDBY Mode	Reserved			Warm-up Time		
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W	Description
7	Enable STANDBY Mode Option	R/W	A 1 in this bit enables the Z382 to go into low-power STANDBY mode when the Sleep instruction is executed.
6	BUSREQ Exits STANDBY Mode	R/W	If this bit is 1, STANDBY mode can be exited by asserting $\overline{\text{BUSREQ}}$
5:3	Reserved	R/W	Reserved. Program as 0.
2:0	Warm-up Time Select	R/W	This field determines the approximate running duration of a warm-up counter that provides a delay before the Z382 resumes clocking and operations, from the time an interrupt or bus request (if so enabled) is asserted to exit Standby mode. 2:0 WARM-UP TIME IN BUSCLK CYCLES: 000 No warm-up 001 65,536 010 131,072 100 524,288

I/O Chip Select Registers (IOCS1H, IOCS1L, IOCS2H, IOCS2L)

I/O Chip Select Register High   IOCS1H: 002Bh, IOCS2H: 002Dh

BIT	7	6	5	4	3	2	1	0
R W	I/O Address High							
DEF	0	0	0	0	0	0	0	0

I/O Chip Select Register Low   IOCS1L: 002Ah, IOCS2L: 002Ch

BIT	7	6	5	4	3	2	1	0
R W	I/O Address Low					I/O Size/Enable		
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W	Description
IOCSH 7:0	I/O Address High	R/W	This register holds the upper address bits, used for comparison with address bus lines A[15:8] for generation of $\overline{\text{IOCS1}}$ or $\overline{\text{IOCS2}}$.
IOCSL 7:3	I/O Address Low	R/W	This field holds the lower address bits, used for comparison with address bus lines A[7:3] for generation of $\overline{\text{IOCS1}}$ or $\overline{\text{IOCS2}}$.

<table>
<tr><th>Bit(s)</th><th>Function</th><th>R/W</th><th>Description</th></tr>
<tr><td>IOCSL
2:0</td><td>I/O Size and Enable</td><td>R/W</td><td>These three bits are decoded to determine which bits of the address bus are used in the address comparison. A value of ‘000’ disables the generation of the associated $\overline{\text{IOCSn}}$ signal, and the pin remains in the High state.
<table>
<tr><th colspan="2">I/O SIZE (BYTES)</th><th>ADDRESS LINES COMPARED</th></tr>
<tr><td>000</td><td>—</td><td>IOCSn Disabled</td></tr>
<tr><td>001</td><td>8</td><td>A[15:3]</td></tr>
<tr><td>010</td><td>16</td><td>A[15:4]</td></tr>
<tr><td>011</td><td>32</td><td>A[15:5]</td></tr>
<tr><td>100</td><td>64</td><td>A[15:6]</td></tr>
<tr><td>101</td><td>128</td><td>A115:7]</td></tr>
<tr><td>110</td><td>256</td><td>A[15:8]</td></tr>
<tr><td>111</td><td>512</td><td>A[15:9]</td></tr>
</table></td></tr>
</table>

Memory Mode Register 1 (MMR1)

MEMORY Mode Register 1 **0032h**

BIT	7	6	5	4	3	2	1	0
R W	ROM Chip Select Enable	Reserved	ROM Wait States			RAM T1 Wait States		
DEF	1	1	1	1	1	1	1	1

Bit(s)	Function	R/W	Description
7	ROM Chip Select Enable	R/W	Setting this bit to 1 enables the ROM chip select logic and allows assertion of the $\overline{\text{ROMCS}}$ pin when a match occurs.
6	Reserved	R/W	Reserved. Program as 0.
5:3	ROM Chip Select Wait States	R/W	These three bits control the number of Wait states (0 - 7) inserted into each access that activates $\overline{\text{ROMCS}}$. When programmed as 000, there are no wait states. When programmed as 111, seven Wait states are inserted.
2:0	RAM Chip Select T1 Wait States	R/W	This field controls the number of T1 Wait states (0 -7) that are inserted in transactions that activate $\overline{\text{RAMCSL}}$ and/or $\overline{\text{RAMCSH}}$. When programmed as 000, there are no T1 Wait states. When programmed as 111, seven Wait states are inserted.

Memory Mode Register 2 (MMR2)

MEMORY Mode Register 2 **00D3h**

BIT	7	6	5	4	3	2	1	0
R W	RAM Chip Select Enable	RAM 8/16	ROM 8/16	RAM T2 Wait States		RAM T3 Wait States		
DEF	1	1	1	1	1	1	1	1

Bit(s)	Function	R/W	Description
7	RAM Chip Select Enable	R/W	Setting this bit to 1 enables the RAM chip select logic and allows assertion of the RAM memory chip select outputs when a match occurs.
6	RAM 8-/16-bit Mode	R/W	A 1 in this bit forces $\overline{\text{MSIZE}}$ low, indicating an 8-bit memory, for all accesses in the selected range. When $\overline{\text{MSIZE}}$ is forced for 8-bit RAM in this way, $\overline{\text{RAMCSL}}$ is asserted for all cycles in the selected address range, and the $\overline{\text{RAMCSH}}$ pin assumes its alternate use as port pin PC7. Writing a 0 results in 16-bit accesses. in which $\overline{\text{RAMCSL}}$ and $\overline{\text{RAMCSH}}$ may be asserted.
5	ROM 8-/16-bit Mode	R/W	A 1 in this bit forces $\overline{\text{MSIZE}}$ Low, indicating an 8-bit memory, for all accesses in the selected range. Writing a 0 releases this forcing.
4:3	RAM Chip Select T2 Wait States	R/W	This field controls the number of T2 wait states (0 - 3) inserted in transactions that activate $\overline{\text{RAMCSL}}$ and/or $\overline{\text{RAMCSH}}$.
2:0	RAM Chip Select T3 Wait States	R/W	This field controls the number of T3 wait states (0 - 7) inserted in transactions that States activate $\overline{\text{RAMCSL}}$ and/or $\overline{\text{RAMCSH}}$.

RAM Address Registers (RAMH, RAML)

RAM Chip Select Address Register High 002Fh

BIT	7	6	5	4	3	2	1	0
R W	RAM Address High							
DEF	1	0	0	0	0	0	0	0

RAM CHIP SELECT ADDRESS REGISTER LOW 002Eh

BIT	7	6	5	4	3	2	1	0
R W	RAM Address High				RAM Size			
DEF	0	0	0	0	1	1	1	1

Bit(s)	Function	R/W	Description
RAMH 7:0	RAM Address High	R/W	This register holds the upper address bits, used for comparison with address bus lines A[23:16] for generation of the $\overline{\text{RAMCS}}$ output(s).
RAML 7:4	RAM Address Low	R/W	This field holds the lower address bits, used for comparison with address bus lines A[15:12] for generation of the $\overline{\text{RAMCS}}$ output(s).

<table>
<tr><th>Bit(s)</th><th>Function</th><th>R/W</th><th>Description</th></tr>
<tr><td>RAML
3:0</td><td>RAM Size</td><td>R/W</td><td>These four bits are decoded to determine which bits of the address bus are used in the address comparison for generation of the RAMCS output(s).
<table>
<tr><th>3:0</th><th>RAM SIZE (BYTES)</th><th>ADDRESS LINES COMPARED</th></tr>
<tr><td>0000</td><td>8MB</td><td>A[23] only</td></tr>
<tr><td>0001</td><td>4MB</td><td>A[23:22]</td></tr>
<tr><td>0010</td><td>2MB</td><td>A[23:21]</td></tr>
<tr><td>0011</td><td>1 MB</td><td>A[23:20]</td></tr>
<tr><td>0100</td><td>512KB</td><td>A[23:19]</td></tr>
<tr><td>0101</td><td>256KB</td><td>A[23:18]</td></tr>
<tr><td>0110</td><td>128KB</td><td>A[23:17]</td></tr>
<tr><td>0111</td><td>64KB</td><td>A[23:16]</td></tr>
<tr><td>1000</td><td>32KB</td><td>A[23:15]</td></tr>
<tr><td>1001</td><td>16KB</td><td>A[23:14]</td></tr>
<tr><td>1010</td><td>8KB</td><td>A[23:13]</td></tr>
<tr><td>1011</td><td>4KB</td><td>A[23:12]</td></tr>
<tr><td>110x</td><td>—</td><td>Reserved</td></tr>
<tr><td>1110</td><td>—</td><td>Reserved</td></tr>
<tr><td>1111</td><td colspan="2">$\overline{\text{RAMCS}}$ is asserted for all addresses other than those for which $\overline{\text{ROMCS}}$ is asserted.</td></tr>
</table>
</td></tr>
</table>

ROM Address Registers (ROMH, ROML)

ROM Chip Select Address Register High 0031h

BIT	7	6	5	4	3	2	1	0
R W	ROM Address High							
DEF	0	0	0	0	0	0	0	0

ROM Chip Select Address Register Low 0030h

BIT	7	6	5	4	3	2	1	0
R W	ROM Address Low				ROM Size			
DEF	0	0	0	0	1	0	0	0

BIT(s)	FUNCTION	R/W	DESCRIPTION
ROMH 7:0	ROM Address High	R/W	This register holds the upper address bits, used for comparison with address bus lines A[23:16] for generation of the $\overline{\text{ROMCS}}$ output
ROML 7:4	ROM Address Low	R/W	This field holds the lower address bits, used for comparison with address bus lines A[15:12] for generation of the $\overline{\text{ROMCS}}$ output.

BIT(s)	FUNCTION	R/W	DESCRIPTION
ROML 3:0	ROM Size	R/W	These four bits are decoded to determine which bits of the address bus are used in the address comparison for generation of the ROMCS output.

3:0	ROM SIZE (BYTES)	ADDRESS LINES COMPARED
0000	8MB	A[23] only
0001	4MB	A[23:22]
0010	2MB	A[23:21]
0011	1MB	A[23:20]
0100	512KB	A[23:19]
0101	256KB	A[23:181
0110	128KB	A[23:17]
0111	64KB	A[23:16]
1000	32KB	A[23:151
1001	16KB	A[23:14]
1010	8KB	A[23:13]
1011	4KB	A[23:12]
11xx	—	Reserved

Refresh Register 0, 1, 2 (RFSHR0, 1, 2)

Refresh Register 0 0023h

BIT	7	6	5	4	3	2	1	0
R W	Request Interval							
DEF	0	0	0	0	0	0	0	0

Refresh Register 1 0024h

BIT	7	6	5	4	3	2	1	0
R W	Missed Requests Count							
DEF	0	0	0	0	0	0	0	0

Refresh Register 2 0025h

BIT	7	6	5	4	3	2	1	0
R W	Refresh Enable	Reserved	Burst Size					
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function*	R/W	Description
RFSHR0 7:0	Request Interval	R/W	The contents of this register defines the interval between refresh requests to the Z382's external interface logic. A value n specified in this field denotes the request interval to be (4 x n) BUSCLK periods. If [7:0] are programmed as 0s, the request interval is 1,024 BUSCLK periods.
RFSHR1 7:4	Missed Requests Count	R/W	This count increments by 1 when a fresh request is made, to a maximum value of 255. Refresh requests over the maximum value would be lost. When the Z382's external interface logic completes each burst of refresh transactions, the count decrements by 1. A user can read the count status and, if necessary, take corrective actions such as adjusting the burst size. When the refresh function is disabled, this count is held at 0.
RFSHR2 7	Refresh Enable	R/W	A 1 enables the refresh function.
RFSHR2 6	Reserved	R/W	Reserved. Must be 0.
RFSHR2 5:0	Burst Size	R/W	This field defines the number of refresh transactions per refresh request made to the Z382's external interface logic. The burst size ranges from 1 to 64, with the highest size specified with [5:0] equal to 0s.

* Exercise care in defining refresh burst size and request intervals to avoid overburdening the system bus with refresh transactions

Refresh Wait Register (RFWR)

Refresh Waits Register **001Fh**

BIT	7	6	5	4	3	2	1	0
R	T1 Waits			T2 Waits		T3 Waits		
W								
DEF	1	1	1	1	1	1	1	1

Bit(s)	Function[1]	R/W	Description
7:5	T1 Wait States	R/W	This binary field defines up to seven T1 Wait states to be inserted in refresh transactions.
4:3	T2 Wait States	R/W	This binary field defines up to three T2 Wait states to be inserted in refresh transactions.
2:0	T3 Wait States	R/W	This binary field defines up to seven T3 Wait states to be inserted in refresh transactions.

Interrupt Enable Register 0 (IENR)

Interrupt Enable Register 0 **0027h**

BIT	7	6	5	4	3	2	1	0
R	$\overline{\text{INT[3:0]}}$ Pin Status				Enable $\overline{\text{INT[3:0]}}$			
W	No Function							
DEF	X	X	X	X	0	0	0	0

Bit(s)	Function	R/W	Description
7:4	$\overline{\text{INT[3:0]}}$ Pin Status	R	This field provides the real time states of the $\overline{\text{INT[3:0]}}$ pins in bits [7:4], respectively.
3:0[1]	Enable $\overline{\text{INT[3:0]}}$	R/W	These read/write, bits control whether INT3, INT2, INT1, and INT0 (respectively) are enabled. Note that these flags are also affected by enable and disable interrupt instructions (with arguments).

NOTES:

1. On the Z380, bits [3:0] were read-only. They are read/write in the Z382, in addition to being affected by the DI (n) and EI (n) instructions.

Assigned Vectors Base Register (AVBR)

Assigned Vectors Base Register **0018h**

BIT	7	6	5	4	3	2	1	0
R W	Assigned Vectors Base							Reserved
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W	Description
7:1	Assigned Vectors Base Address (AB[15:9])	R/W	The Interrupt Register Extension, Iack, together with AB[15:9], define the base address of the assigned interrupt vectors table in memory space.
0	Reserved	R/W	Reserved. Program as 0.

Trap and Break Register (TBR)

Trap and Break Register **0019h**

BIT	7	6	5	4	3	2	1	0
R W	Reserved						Trap On Instruction Fetch	Trap On Interrupt Vector
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W	Description
7:2	Reserved	R/W	Reserved. Must be 0.
1	Trap on Instruction Fetch (TF)	R/W	TF is set if an undefined op code is fetched in the instruction stream. TF can be reset under program control by writing a 0 to it. However, it cannot be set by writing to it.
0	Trap on Interrupt Vector (TV)	R/W	TV is set if an undefined opcode is returned as a vector in an interrupt acknowledge transaction in mode 0. TV can be reset under program control by writing a 0 to it. However, it cannot be set by writing to it.

INT3-1 Control Register (I31CR)

Interrupt3-1 Control Register **003Bh**

BIT	7	6	5	4	3	2	1	0
R	No Function		INT3 Mode		INT2 Mode		INT1 Mode	
W	Command							
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W	Description
7:6	Command	R/W	When a pin is selected as edge-sensitive and enabled, it begins requesting an interrupt when it detects the specked edge on the pin. When the interrupt is acknowledged, the ISR must write the corresponding non-zero command to this field to clear the interrupt request, before it re-enables interrupts. These bits read as 00. COMMAND 00 No operation 01 Clear INT1 edge 10 Clear INT2 edge 11 Clear INT3 edge
5:4 3:2 1:0	INT3 Mode INT2 Mode INT1 Mode	R/W	These three fields control when and how the Z80382 recognizes an interrupt on the corresponding pin MODE 00 Low Level Sensitive: an interrupt is requested whenever the pin is enabled and it is Low. 01 High Level Sensitive: an interrupt is requested whenever the pin is enabled and it is High. 10 Falling Edge Triggered: when the pin is enabled, an interrupt is requested from the time a falling edge is detected on it, until software writes the corresponding Clear command to the Command field of this register. 11 Rising Edge Triggered: when the pin is enabled, an interrupt is requested from the time a. rising.edge is detected on it, until software writes the corresponding Clear command to the Command field of this register.

Host Interface

OVERVIEW

In addition to the serial data interfaces (see “Serial Communication Channels” on page 243), the Z80382 provides several other facilities for data transfers between the chip and a host system. These are the 16550 MIMIC interface, the Host DMA Mailbox and the Host I/O Mailbox. These are supported by the Plug-and-Play (PnP) and PCMCIA interface modules. The PnP interface provides auto-configuration in ISA (AT bus) applications, while the PCMCIA interface provides this function for PC card environments.

MIMIC INTERFACE

Introduction

The Z80382 includes a 16x50 MIMIC interface that allows it to emulate the operation of a PC UART. The interface allows the Z80382 to be connected directly to an ISA bus or PCMCIA bus without any external circuitry. The MIMIC contains the 16x50 register set and the same interrupt structure. The data path allows parallel transfer of data between the host processor and the internal processor of the Z80382.

Six priority encoded conditions can interrupt the Z80382. When the PC Host reads or writes certain MIMIC registers, an interrupt to the Z80382 is generated. Each interrupt can be individually masked off or all interrupts can be disabled by writing a single bit.

Two 8-bit timers are also available to control the data transfer rate of the MIMIC interface. Their input is tied to a Baud Rate Generator (BRG) in the MIMIC, allowing a wide range of data rates to be emulated. Two additional 8-bit timers are available for programming the FIFO timeout feature (Four Character-Time Emulation) for both the Receiver and the Transmitter FIFOs.

The 16550 MIMIC supports the PC Host interrupt structure via the Plug-and-Play ISA or PCMCIA interface modules. COM Port decoding is also provided by the same modules.

A bit in the Z80382 System Configuration Register controls whether the registers of the 16x50 MIMIC interface are accessible in any page of I/O space, as on the Z8018x family, because only the lowest eight address lines are decoded, or whether A15-8 must be zero to access the registers.

The MIMIC Interface can transfer both Transmit and Receive data under control of the Z80382's DMA channels, minimizing processor overhead and maximizing throughput in high-speed applications.

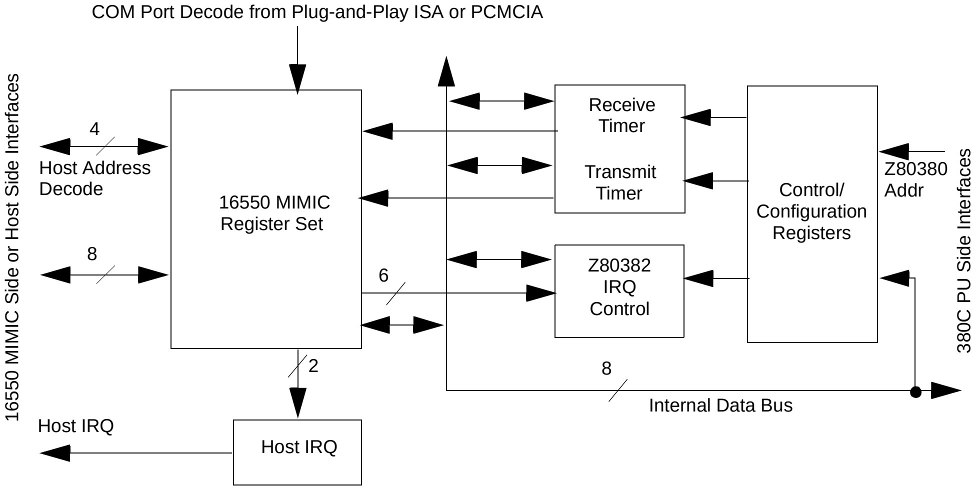


Figure 9. 16550 MIMIC Block Diagram

MIMIC Receiver FIFO

The receiver FIFO is 16 words deep and stores eight data bits and three error bits (Parity error, Framing error and Break detect) for each character received. The data and error bits move together in the FIFO. The error bits become available to the Host side of the interface when that particular location becomes the next address to read. At that time they may either be read by the Host and they may cause an interrupt to the Host interface if so enabled. The error bits are set by the error status of the byte at the top of the FIFO but may only be cleared by reading the Line Status Register (LSR). If successive reads of the receiver FIFO are performed without reading the LSR, the status bits are set if any of the bytes read have the respective error bit set.

The Host processor may be interrupted when 1, 4, 8 or 14 bytes are available in the receiver FIFO. If the FIFO is not empty, but is below the programmed trigger value, a timeout interrupt may be invoked which is triggered if the receiver FIFO is not written by the 380C or read by the Host during an interval determined by the Character Timeout Timer. These interrupts are cleared by the FIFO reaching the trigger point or by resetting the Timeout interval timer by a FIFO 380C write or Host read access.

The Character Timeout Timer is an additional timer with 380C access only which is used to emulate the 16550 four-character timeout delay. It receives the BRG as its input clock. Software is responsible for programming the correct values into the Receiver Timeout Register and the BRG to achieve the correct delay interval for the timeout.

With FIFO mode enabled, the 380C is interrupted when the receiver FIFO is empty. This bit corresponds to a Host read of the receiver buffer in non-FIFO (16450) mode. The interrupt source is cleared when the FIFO becomes non-empty or the 380C reads the IUS/IP register.

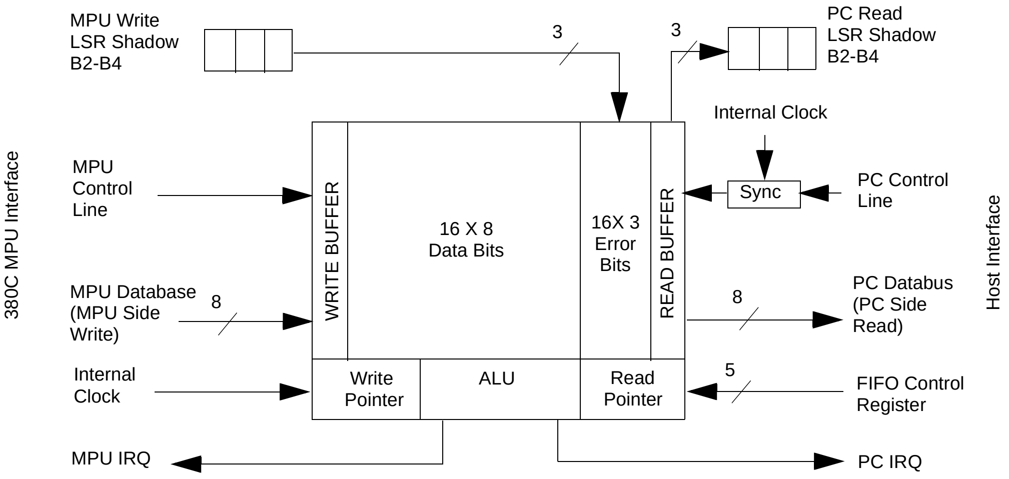


Figure 10. 16550 MIMIC Receiver FIFO Block Diagram

MIMIC Transmitter FIFO

The transmitter FIFO is 16-bytes deep with Host write and 380C read access. In FIFO mode, the Host can be interrupted when the transmitter FIFO becomes empty. The interrupt is cleared when the transmitter FIFO becomes non-empty or the IIR register is read by the Host.

On the 380C interface, the transmit FIFO can be programmed to interrupt the 380C on 1, 4, 8 or 14 bytes of available data. A timeout feature exists, the Transmitter Timeout Timer, which is an additional 8-bit timer with the BRG as the input source. If the transmitter FIFO is non-empty and no Host write or 380C read of the FIFO has taken place within the timer interval, a timeout occurs, causing a corresponding interrupt to the 380C.

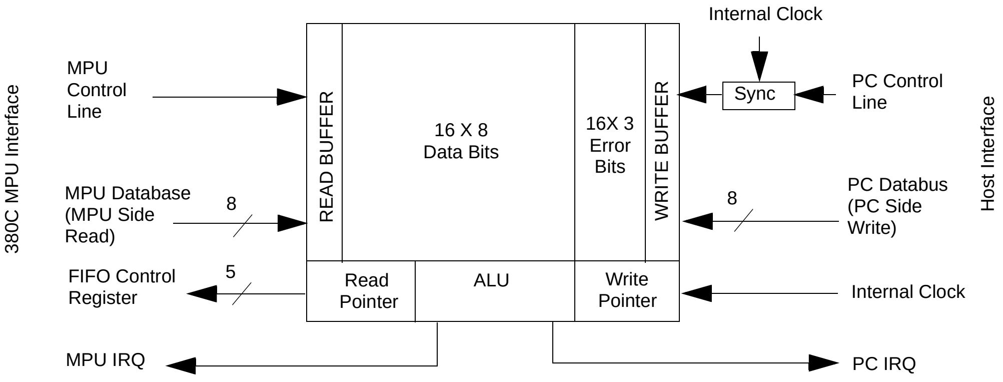


Figure 11. 16550 MIMIC Transmitter FIFO Block Diagram

Z382 MIMIC Synchronization Considerations

Because of the asynchronous nature of the FIFOs on the MIMIC, synchronization is provided to prevent conflicts between simultaneous accesses by the 380C and the Host.

I/O to the FIFO is buffered, the buffers allowing both Host and 380C to access the FIFO asynchronously. Read and Write requests are then synchronized by means of the 380C clock. Incoming signals are buffered in such a way that meta-stable input levels are stabilized to valid 1 or 0 levels. Actual transfers between the buffers and the FIFO memory are timed by the 380C clock. ALU evaluation is performed on a different phase than the transfer to ensure stable pointer values.

Another potential problem is that of simultaneous access of the 380C and Host to any of the various dual-access type registers, that is, registers which are written by one side of the interface and read from the other. This is solved by dual buffering of the various read/write registers. During a read access by either the 380C or Host to a dual-access register, the data in the buffered slave register is not permitted to change. Any write that might take place during this time is stored at the input of the master register. The corresponding status/interrupt is reset appropriately based on the write having followed the read to the register. For example, the IUS/IP bit for the LCR write is not cleared by the 380C read of the LCR if a simultaneous write to the LCR by the Host takes place. Instead, the LSR data changes after the read access and IUS/IP bit 3 remains at logic 1.

Double Transmit Buffering in 16450 Mode

The Z80382 implements double buffering for the transmitter in 16450 mode and sets the TEMT bit in the LSR Register automatically.

If character delay emulation is being used:

1. The Host THRE bit in the LSR Register is set when the THR Register is empty.
2. Host writes to the 16450 THR Register.
3. Whenever the Z80382 TSR buffer is empty and the one character delay timer is in a timeout state, the byte from the THR is transferred to the TSR buffer.
4. Restart character delay timer (timer reloads and counts down) with byte transfer from THR to the TSR buffer.
5. Whenever the TSR buffer is full, the TEMT bit in LSR Register for 380C is reset with no delay.
6. MPU reads TSR buffer.
7. TEMT bit in LSR Register for 380C is set with no delay whenever the TSR buffer is empty.
8. When the TSR buffer is read by the 380C, the THR is empty, and the one character delay timer reaches zero, the TEMT bit in the Host LSR Register is set from 0 to 1.
   The Host THRE bit in the LSR Register is reset whenever the THR

Register is full and set whenever the THR Register is empty. 380C IREQ for the transmit data is triggered whenever the TSR buffer is full and cleared whenever the TSR buffer is empty.

If character delay emulation is not used, the TEMT bit in the LSR is set whenever both the THR and the TSR buffer are both empty. The Host TEMT bit is clear if there is data in either the TSR buffer or THR Register.

Disable this feature when 16550 FIFO mode is enabled.

Transmit and Receive Timers

Because of the speed at which data transfers can take place between the Z80382 and the Host, two timers have been added to alleviate any software problems that a high speed data transfer might cause to existing software. The timers receive their input from the MIMIC BRG clock. This allows the programmer access to a 24-bit timer to slow down the data transfers.

Both timers are single pass and stop on a count of zero. Their purpose is to delay data transfer just as if the 18554 UART had to shift the data in and out serially. If high speed data transfer is not a problem, then data can be read and written as fast as the Host and the 380C can access the devices.

In FIFO operation, the timers can be used to delay the status to the Host interface by the time that would be required to actually shift the characters out or in if an actual UART were present.

Transmitter Timer

An interrupt to the 380C is generated when the Host writes to the Transmitter Holding Register. The 380C then reads the data in the Transmitter Holding Register. If the Transmitter Timer is enabled at the time of this read, the time constant is loaded from the Transmitter Time

Constant Register into the Transmitter Timer and enables the count. After the timer reaches a count of zero, the Transmitter Holding Register Empty bit is set. However, the above is only true when the Host is reading the Transmitter Holding Register Empty bit. To allow the 380C to know that it has already read the byte of data, a mirrored Transmitter Holding Register Empty bit is set immediately following a read from the Transmitter Holding Register. This mirrored bit is always read back to the 380C when it reads the Line Status Register. If the Transmitter Tuner is not enabled when the 380C reads the Transmitter Holding Register, then both Transmitter Holding Register Empty bits are set immediately.

For FIFO operation, the effect is similar in that the status to the Host is delayed, such that a Host interrupt for an empty FIFO does not occur before the time required for each character read from the FIFO by the 380C has elapsed. The effect is that when the delay feature is enabled, the Host does not detect data requests from an empty FIFO any faster than would occur with a true UART. This timer is also used to delay data transfer from the THR Register to the Z80382 TSR buffer in double buffer mode.

Receive Timer

When the 380C writes to the Receive Buffer Register and the Receive Timer is enabled, the Receive Timer is loaded with the Receiver Time Constant, the timer is enabled and counts down to zero. When the timer reaches zero, the Data Ready bit in the Line Status Register is set. As with the Transmit Timer, the Data Ready bit is also mirrored. Immediately upon a write to the Receive Buffer, the mirrored bit is set to let the 380C know that the byte has already been written. If the timer is not enabled, both Data Ready bits are set immediately upon a write to the Receive Buffer.

The FIFO mode of operation is similar in that the status to the Host is delayed by the time required for each character written to the FIFO by the 380. The effect is that when the delay feature is enabled, the Host does not detect a FIFO trigger level or DMA request faster than would occur with a true UART.

Baud Rate Generator

The Baud Rate Generator provides emulation timing for the MIMIC. The BRG output clocks the MIMIC emulation counter, while the BRG is clocked by the BUSCLK output of the 380C. Two 8-bit registers are provided to program the BRG time constant, with the time constant value required for a specific baud rate calculated as follows:

Time constant (decimal) = (BUSCLK/(2*baudrate)) - 2

The desired time constant is programmed into the BRGL and BRGH registers (see below) and the timer is enabled by setting bit 0 of the IOBRG Register. Design is such that on-the-fly modification of the registers does not cause irregular BRG output.

MIMIC Registers

MIMIC Programming Registers

The MIMIC module contains registers that can be accessed by the 380C to control various aspects of MIMIC operation. Table 13 lists these registers:

Table 13. MIMIC Programming Registers

Register	380C I/O Address
I/O and BRG Control Register	00D6h
MIMIC BRG Low Constant Register	00E0h
MIMIC BRG High Constant Register	00E1h
MIMIC Modification Register	00E9h
Rx Timeout Time Constant Register	00EAh
Tx Timeout Time Constant Register	00EBh
FIFO Status and Control Register	00ECh
MIMIC DMA Control Register	00EFh
Transmit Time Constant Register	00FAh
Receive Time Constant Register	00FBh
Interrupt Vector Register	00FCh
Interrupt Enable Register	00FDh
MIMIC IUS/IP Register	00FEh
MIMIC Master Control Register	00FFh

MIMIC Host Interface Registers

In addition to the 380C-accessible registers, the Z80382 contains registers that are accessible to both the 380C and the Host processor. These registers are used to emulate the 16580 UART so that the Host can gain access to them just as if it was emulating a 16550 or 16450. This setup provides compatibility with existing Host communication software. Table 14 lists these registers.

Table 14. MIMIC Host Interface Registers

Register	Host Address[1]	380C I/O Address
FIFO Control Register	02h	00E9h
Receiver Buffer Register	00h[2]	00F0h
Transmitter Holding Reg.	00h[2]	00F0h
Interrupt Enable Register	01h[2]	00F1h
Line Control Register	03h	00F3h
Modem Control Register	04h	00F4h
Line Status Register	05h	00F5h
Modem Status Register	06h	00F6h
Scratch Register	07h	00F7h
Divisor Latch LS Byte	00h[3]	00F8h
Divisor Latch MS Byte	01h[3]	00F9h
Interrupt ID Register	02h	—

NOTES:

1. The host address is relative to the MIMIC base address decoded by the PnP ISA or PC-MCIA modules in the Z80382.
2. DLAB (LCR[7]) = 0.
3. DLAB (LCR[7]) -1.

MIMIC Register Descriptions

The following sections describe the registers associated with the MIMIC Interface module. They are ordered in two groups, per the tables in “MIMIC Programming Registers” on page 101 and “MIMIC Host Interface Registers” on page 102, by ascending 380C I/O address. In the descriptions, the values shown in the line labeled *DEF* are the default values after a reset. X indicates that the reset value is indeterminate.

XXXXh is the register's address in the 380C's or Host's I/O space, as appropriate. In specifying a Host address, *Base* refers to the MIMIC I/O base address assigned during the Plug-and-Play or PCMCIA configuration process.

I/O and BRG Control Register (IOBRG)

I/O and BRG Control Register **380C Address: 00D6h**

BIT	7	6	5	4	3	2	1	0
R W	Reserved			I/O Mailbox Interrupt Enable	$\overline{\text{INT0}}$ Assertion on MIMIC Access	Reserved		BRG Enable
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W	Description
7:5	Reserved		
4	I/O Mailbox Interrupt Enable	R/W	When this bit is a 1, certain accesses by the Host to I/O Mailbox registers causes an interrupt to the 380C via the Assigned Vector facility. See the description in the I/O Mailbox section for additional details.

Bit(s)	Function	R/W	Description
3	$\overline{\text{INT0}}$ Assertion on MIMIC Access	R/W	When this bit is a 1 and $\overline{\text{HALT}}$ is active (power-down), any Host access to the Access MIMIC causes a falling edge on $\overline{\text{INT0}}$. Because this interrupt source has no vector, INT MODE 1 must be used when enabling this feature. This feature is disabled on powerup. $\overline{\text{INT0}}$ assertion is released when $\overline{\text{HALT}}$ is deasserted. The THRE bit, LSR[5], is forced to 0 on the Host side to prevent THR overruns during power-down modes when this feature is enabled. When the MIMIC comes out of power-down, THRE resumes normal functionality.
2:1	Reserved		
0	BRG Enable	R/W	When this bit is set the MIMIC BRG begins counting down to generate a programmed square wave to the MIMIC emulation timers.

Baud Rate Generator Low Register (BRGL)

Baud Rate Generator Low **380C Address: 00E0h**

BIT	7	6	5	4	3	2	1	0
R W	Baud Rate Generator Low							
DEF	1	1	1	1	1	1	1	1

Bit(s)	Function	R/W	Description
7:0	Baud Rate Generator Low	R/W	This field contains the lower byte of the BRG time constant. See “Baud Rate Generator” on page 101.

Baud Rate Generator High (BRGH)

Baud Rate Generator High **380C Address: 00E1h**

BIT	7	6	5	4	3	2	1	0
R W	Baud Rate Generator High							
DEF	1	1	1	1	1	1	1	1

Bit(s)	Function	R/W	Description
7:0	Baud Rate Generator High	R/W	This field contains the upper byte of the BRG time constant. See “Baud Rate Generator” on page 101.

MIMIC Modification Register (MMR)

MIMIC Modification Register **380C Address: 00E9h**

BIT	7	6	5	4	3	2	1	0
R	See FCR							
W	Reserved (Program as 0)						16550/450 Rx Overrun	Reserved (Program as 0)
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W	Description
7:2	Reserved		Program as `000000`
1	Receiver Overrun	W	The 16450 and 16550 allow the last position in their receiver FIFOs to be overwritten by the receiver when the FIFO is full (an Overrun condition). When this bit is `1`, the last position in the Rx FIFO can be overwritten by 380C software. When this bit is a `0`, a write to the RBR has no effect when the Rx FIFO is full.
0	Reserved		Program as `0`

Receiver Timeout Time Constant (RTTC)

Receiver Timeout Time Constant Register **380C Address: 00EAh**

BIT	7	6	5	4	3	2	1	0
R W	Receiver Timeout Time Constant							
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W	Description
7:0	Receiver Timeout Time Constant	R/W	This field contains an 8-bit constant for emulation of the 16550 four-character timeout feature. Software must determine the value to load into this register based on the bit rate and word length specified by the MIMIC interface with the Host. This timer receives its input from the MIMIC BRG Clock and is enabled to count down when the enable bit in the FSCR register is set and the trigger level has not been reached on the receiver FIFO. The counter reloads each time there is a read of or write to the receiver FIFO.

Transmitter Timeout Time Constant (TTTC)

Transmitter Timeout Time Constant Register **380C Address: 00EBh**

BIT	7	6	5	4	3	2	1	0
R W	Transmitter Timeout Time Constant							
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W	Description
7:0	Transmitter Timeout Time Constant	R/W	This field contains an 8-bit constant which is used to determine the interval for the Constant transmitter timeout timer. If allowed to decrement to zero, this timer interrupts the 380C by setting the THR bit in the IUS/IP register. This timer receives its input from the MIMIC BRG Clock and is enabled to count down when the enable bit in the FSCR register is set and the trigger level has not been reached on the transmitter FIFO. The counter reloads each time there is a read of or write to the transmitter FIFO.

FIFO Status and Control Register (FSCR)

FIFO Status And Control Register **380C Address: 00ECh**

BIT	7	6	5	4	3	2	1	0
R W	Transmitter FIFO Trigger Level		Receiver Timeout Enable	Transmit Timeout Enable	Reserved Always 0	Double Buffer Mode	Receive Timeout Mode	Force 16450 Mode
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W	Description
7:6	Transmit FIFO Trigger Level	R/W	This field determines the minimum number of bytes available to read in the transmitter FIFO before an interrupt occurs to the 380.
			b7 b6 Level (bytes) 0 0 1 0 1 4 1 0 8 1 1 14
5	Receiver Timeout Enable	R/W	This bit enables the Z80382 timer that is used to emulate the four-character receiver timeout delay that is specified by the 16550. A Receiver Timeout interrupt to the host occurs under the following conditions: • No read of or write to the receiver FIFO • Data bytes are available but below the Host trigger level (FCR[7:6]) • The receiver timeout timer reaches 0.

Bit(s)	Function	R/W	Description
4	Transmitter Timeout Enable	R/W	This bit enables the Z80382 timer that is used to interrupt the 380C If characters are available in the transmitter FIFO, but are below the trigger level specified in <7:6> above. The timer counts down if this bit is one and the number of bytes is at least one but is below the set transmitter trigger level. The timer time-outs and interrupts the 380C if no read of or write to the Transmitter FIFO takes place within the timer interval.
3	Reserved		Always write and read as 0 by users. Writing a 1 enables a test mode for emulation timers.
2	Double Buffer Mode	R/W	In 16450 mode, setting this bit to a 1 enables TEMT hardware emulation and transmitter double buffering. In 16550 mode this bit must be kept at a 0. Double buffering enables a Transmit Shift Register (TSR) to act as a slave register, while the Host writes to a Transmit Holding Register. The 380C reads from the Transmit Shift Register. This condition allows the Host to write two consecutive bytes into the MIMIC. TEMT emulation: If character delay emulation is being used, the TEMT status bit is set as follows: the TSR is emptied and the associated delay logic has set the (delayed) THRE bit in the LSR. At this time the one character delay timer begins. When this timer reaches zero, the TEMT bit is set if the THR and the TSR output buffer are empty. TEMT is cleared whenever there is data in either the THR or the TSR output buffer.

Bit(s)	Function	R/W	Description
1	Receive Timeout Mode	R/W	Setting this bit enables the RTO timeout to emulate the 16550 device. If this bit is 1, the RTO timer does not start counting the timeout until all characters have been clocked through the receiver character delay emulation logic. This condition prevents an RTO from occurring before a (delayed) receiver trigger level interrupt. When this bit is 0, the RTO measures is timeout from the last read or write to the receiver FIFO.
0	Force 16450 Mode	R/W	If this bit is 1, the MIMIC operates in 16450 mode. Bit 0 in the FCR Register and the MIMIC internal FIFO enable are forced to 0. Bits 7 and 6 in the IIR remain at their last value when this bit is set.

MIMIC DMA Control Register (MDCR)

MIMIC DMA Control Register **380C Address: 00EFh**

BIT	7	6	5	4	3	2	1	0
R W	Transmit DMA Enable	Transmit DMA Channel			Receive DMA Enable	Receive DMA Channel		
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W	Description
7	Transmit DMA Enable	R/W	A `1` enables DMA operation for the MIMIC transmitter.
6:4	Transmit DMA Channel	R/W	Selects the DMA channel for the transmitter when transmit DMA operation is enabled.
3	Receive DMA Enable	R/W	A `1` enables DMA operation for the MIMIC receiver.
2:0	Receive DMA Channel	R/W	Selects the DMA channel for the receiver when receive DMA operation is enabled.

Note: Because the data flowing through the MIMIC has no organization that can be discerned by the hardware, the MIMIC does not use any of the following features of the DMA channels: *with Command*, *Notify at End of Buffer*, or *with Status*.

Transmitter Time Constant Register (TTCH)

Transmitter Time Constant Register **380C Address: 00FAh**

BIT	7	6	5	4	3	2	1	0
R W	Transmitter Time Constant							
DEF	1	1	1	1	1	1	1	1

Bit(s)	Function	R/W	Description
7:0	Transmitter Time Constant	R/W	This field contains the time constant which is used to emulate serial data transmission. See “Transmitter Timer” on page 99.

Receiver Time Constant Register (RTCR)

Receiver Time Constant Register **380C Address: 00FBh**

BIT	7	6	5	4	3	2	1	0
R W	Receiver Time Constant							
DEF	1	1	1	1	1	1	1	1

Bit(s)	Function	R/W	Description
7:0	Receiver Time Constant	R/W	This field contains the time constant which is used to emulate serial data reception. See “Receive Timer” on page 100.

Interrupt Vector Register (IVEC)

Interrupt Vector Register **380C Address: 00FCh**

BIT	7	6	5	4	3	2	1	0
R W	Upper Nibble IVEC				Status op code			0/op code
DEF	0	0	0	0	0	0	0	0

The Interrupt Vector Register contains part or all of the value to return when the 380C processor acknowledges an interrupt from the MIMIC, depending upon the VIS bit in the MIMIC Master Control Register, MMC[0].

When VIS is 0 and/or there is no interrupt pending, the last value written to the register can be read back. If VIS is 1 and an interrupt is pending, the value read is the last value written to the upper nibble plus a code identifying the interrupt that is pending.

If VIS is 1, then the lower 4 bits of the vector change asynchronously depending on the interrupting source.

Bit(s)	Function	R/W	Description
7:4	Upper Nibble IVEC	R/W	These four bits always reads back what was last written into them.
3:1	Status/ op code	R/W	These three bits identify the highest priority pending interrupt if VIS in the MCR register is 1. If VIS bit is 0, then these bits read out what was last written to them.

Value <3:1>	IRQ Source
000	None
001	FCR
010	DLL/DLM
011	LCR
100	MCR
101	RBR
110	TTO
111	THR

0	0/op code	R/W	This bit is always a 0 when the VIS bit is a 1. If the VIS bit is a 0, then this bit reads back what was last written to it.

Interrupt Enable Register (IER)

Interrupt Enable Register **380C Address: 00FDh**

BIT	7	6	5	4	3	2	1	0
R W	Master Interrupt Enable	Enable THR Interrupt	Enable TTO Interrupt	Enable RBR Interrupt	Enable MCR Interrupt	Enable LCR Interrupt	Enable DLL/ DLM Interrupt	Enable FCR Interrupt
DEF	0	0	0	0	0	0	0	0

The IE Register allows each of the MIMIC interrupts to the 380C to be masked off individually or globally.

The priority of interrupts is as follows:

Highest	6. THR IRQ
	5. TTO IRQ
	4. RBR IRQ
	3. MCR IRQ
	2. LCR IRQ
	1. DLL IRQ
	1. DLM IRQ
Lowest	0. FCR IRQ

Bit(s)	Function	R/W	Description
7	Master Interrupt Enable	R/W	If bit 7 is a 0, all interrupts from the 16550 MIMIC are masked off. If this bit is a 1, then interrupts are enabled individually by setting the appropriate bit.
6	Enable THR Interrupt	R/W	If this bit is 1, it enables the Transmitter Holding Register interrupt.
5	Enable TTO Interrupt	R/W	If this bit is 1, it enables the Transmitter Timeout interrupt
4	Enable RBR Interrupt	R/W	If this bit is 1, it enables the Receiver Buffer Register interrupt
3	Enable MCR Interrupt	R/W	If this bit is 1, it enables the Modem Control Register interrupt.
2	Enable LCR Interrupt	R/W	If this bit is a 1, it enables the Line Control Register interrupt.
1	Enable DLL/DLM Interrupt	R/W	If this bit is a 1, it enables the Divisor Latch Least and Most Significant Byte interrupts.
0	Enable FCR Interrupt	R/W	If this bit is a 1, then interrupts are enabled for a Host write to the FIFO Control Register (FCR).

Interrupt Under Service/Interrupt Pending Register (IUSIP)

IUS/IP Register **380C Address: 00FEh**

BIT	7	6	5	4	3	2	1	0
R	Interrupt Under Service	THR Written Interrupt Pending	Trans-mitter Timeout IP	Receive Buffer Read IP	MCR Write Interrupt Pending	LCR Write Interrupt Pending	DLL/DLM Write Int. Pending	FCR Write Or Tx Overrun Ip
W	Reset Highest Priority IUS	No Function. Must be 0.						
DEF	0	0	0	0	0	0	0	0

The IUS/IP Register is used by the 380C to determine what has caused a MIMIC interrupt. This register sets the appropriate bit when an interrupt occurs.

Bit(s)	Function	R/W	Description
7	Interrupt Under Service	R	This bit represents a logical-0R of each individual IUS bit for the internal MIMIC interrupt daisy chain. An IUS bit is set when an interrupt is registered (IP set) and enabled (IE set); the incoming IEI daisy chain is active (chain enabled) and the 380C performs an interrupt acknowledge cycle.
7	Reset Highest Priority IUS	W	Writing a 1 to this bit resets the highest priority IUS bit that is set. Writing a 0 to this bit has no effect.

Bit(s)	Function	R/W	Description
6	Transmitter Holding Register Written Interrupt	R	This bit is set when the Host writes to the Transmitter Holding Register. It is reset when the 380C reads the Transmitter Holding Register. In FIFO mode, this bit is set when the trigger level programmed in FSCR<7:6> is reached (1, 4, 8 or 14 bytes available). If the THR Timer is enabled, this interrupt is delayed by the number of character times programmed as the trigger level. The bit is cleared when the number of data bytes falls below the set trigger level.
5	Transmitter Timeout with Data in FIFO Interrupt	R	This bit is set when the transmitter FIFO has been idle (no read or write) with data in FIFO Interrupt in the FIFO but below the trigger level, and the Transmitter Timeout timer decrements to zero. It is cleared when the FIFO is read or written.
4	Receive Buffer Read Interrupt	R	This bit is set when the Host reads the Receive Buffer Register. It is reset when the 380C writes to the Receive Buffer Register. In FIFO mode, this bit is set when the receiver FIFO has been emptied by the Host This bit and interrupt are cleared when one or more bytes are written into the receiver FIFO by the 380C.
3	Modem Control Register	R	This bit is set when the Host writes to the Modem Control Register. It is reset when Write Interrupt the 380C reads the Modem Control Register.
2	Line Control Register Write Interrupt	R	This bit is set when the Host writes to the Line Control Register. It is reset when the 380C reads the Line Control Register.

Bit(s)	Function	R/W	Description
1	Divisor Latch LS/MS Write Interrupt	R	This bit is set when the Host writes to the Divisor Latch Least Significant or Most Significant byte. It is reset when the 380C reads the LS/MS register(s). To ensure that this bit is cleared, the 380C interrupt service routine should read both registers.
0	FIFO Control Register Write Interrupt	R	This bit is set when the Host writes to the FCR. It is reset when the 380C reads this register.

MIMIC Master Control Register (MMC)

MIMIC Master Control Register **380C Address: 00FFh**

BIT	7	6	5	4	3	2	1	0
R W	Transmit Delay Timer Enable	Receive Delay Timer Enable	Reserved			Reserved		Vector Includes Status
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W	Description
7	Transmit Delay Timer Enable	R/W	If this bit is zero, Transmitter Holding Register Empty (THRE, LSR[5]) is set immediately on a 380C read of the Transmit Holding Register. If set to a one, the transmit delay timer is enabled. When the 380C reads the Transmit Holding Register, the transmit delay timer is loaded automatically from the Transmit Time Constant Register and the timer is enabled to count down to zero. This timer delays setting THRE until the timer times out.
6	Receive Delay Timer Enable	R/W	If this bit is zero, Received Data Available (RDA, LSR[0]) is set immediately on a Enable 380C write to the Receiver Buffer Register. If set to a one, the receive delay timer is enabled. When the 380C writes to the Receiver Buffer Register, it loads the receive delay timer from the Receive Time Constant Register and enables the timer to count down to zero. This timer delays setting RDA until the timer times out.
5:3	Reserved		These bits were used in previous MIMIC interfaces for DMA control. In the Z382 these functions are handled in a separate MIMIC DMA Control Register.

Bit(s)	Function	R/W	Description
2:1	Reserved		These bits were used in previous MIMIC interfaces to control interrupt signalling to the Host. In the Z382 these functions are handled by the Plug-and-Play and PCMCIA interfaces.
0	Vector Includes Status	R/W	This bit is used to select the interrupt response mode of the MIMIC module during a 380C interrupt acknowledge cycle. A 0 in this bit makes the MIMIC return the vector programmed into the IVEC register without change, regardless of its status. A 1 in this bit makes the MIMIC modify its interrupt vector depending on the highest-priority pending interrupt, as described in “Interrupt Vector Register (IVEC)” on page 117. If the 380C is programmed for mode 3 response, zeroes are returned in the more- significant 8 bits of the vector.

FIFO Control Register (FCR)

FIFO Control Register **Host Address: (Base + 02)h**
380C Address: 00E9h

BIT	7	6	5	4	3	2	1	0
R W	Receiver FIFO Trigger Level		FCR Write Interrupt	Tx Overrun Interrupt	Reserved Always 0	Transmit FIFO Reset	Receive FIFO Reset	FIFO Enable
DEF	0	0	0	0	0	0	0	0

Note: On the Z80382 side, this register address is shared with the MIMIC Modification Register (MMR). See “MIMIC Modification Register (MMR)” on page 108.

Bit(s)	Function	R/W[1]	Description
7:6	Receiver FIFO Trigger Level	H:W P:R	This two bit field. determines the minimum number of bytes required in the receiver FIFO before an interrupt to the Host occurs b7 b6 Level (bytes) 0 0 1 0 1 4 1 0 8 1 1 14
5	Reserved	H:W	Write as 0.
	FCR Write Interrupt	P:R	This bit flags an interrupt indicating that the FCR had changed. A read of the FCR from the 380C side clears this bit.
4	Reserved	H:W	Write as 0.
	Tx Overrun Interrupt	P:R	This bit flags an interrupt indicating a Transmitter Overrun. A read of the FOR from the 380C side clears this bit.

Bit(s)	Function	R/W[1]	Description
3	Reserved	H:W P:R	Write as 0.
2	Transmitter FIFO Reset	H:W P:R	Setting this bit to a 1 resets the transmitter FIFO pointers; any data in the FIFO is lost. This bit is self clearing; however, a shadow bit exists that is cleared only when read by the 380C, allowing the 380C to monitor a FIFO reset by the Host.
1	Receiver FIFO Reset	H:W P:R	Setting this bit to a 1 resets the receiver FIFO pointers; any data in the FIFO is lost. This bit is sell clearing; however, a shadow bit exists that is cleared only when read by the 380C, allowing the 380C to monitor a FIFO reset by the Host.
0	FIFO Enable	H:W P:R	The Host writes a 1 to this bit to put the 16550 MIMIC into FIFO mode. This bit must be a 1 when writing to the other bits in this register or they are not programmed. When this bit changes state, any data in the FIFOs and Transmitter Holding and Receiver Buffer Registers is lost and any pending interrupts are cleared.

NOTES:

[1] H indicates Host-side capability, P indicates processor-side (380C) capability.

Receiver Buffer Register (RBR)

Receiver Buffer Register **Host Address: (Base + 00)h, LCR[7] = 0**
380C Address: 00F0h

BIT	7	6	5	4	3	2	1	0
R	Host Reads Data Byte From 380C							
W	380c Writes Data Byte To Host							
DEF	X	X	X	X	X	X	X	X

In FIFO mode operation, these addresses are used to read (Host) and write (380C) the Receiver FIFO.

Bit(s)	Function	R/W[1]	Description
7:0	Received Data	H:R	On the Host side, this address is used to read both received data and the LS Byte of the Divisor Latch. To read received data, the DLAB bit in the Line Control Register (LCR[7]) must be zero.
		P:W	When the 380C has assembled a byte of data to pass to the Host, it places it in the Receiver Buffer Register. If the Received Data Available interrupt is enabled, an interrupt is generated for the Host, and the Data Ready bit is set. If the receiver timer is enabled, the interrupt and setting of the Data Ready bit are delayed until after the timer times out. Also, the shadow bits of the Line Status Register (LSR) are transferred to their respective bits in the LSR or Receive FIFO when the 380C writes to the Receiver Buffer Register (see LSR[4:1]). This allows a simultaneous setting of error bits when the data is written to the Receiver Buffer Register.

NOTES:
1. H indicates Host-side capability, P indicates processor-side (380C) capability.

Transmitter Holding Register (THR)

Transmitter Holding Register **Host Address: (Base + 00)h, LCR[7] - 0**
380C Address 00F0h

BIT	7	6	5	4	3	2	1	0
R	Host Writes Data Byte To 380C							
W	380C Reads Data Byte From Host							
DEF	X	X	X	X	X	X	X	X

In FIFO mode of operation, these addresses are used to read (380C) and write (Host) the Transmitter FIFO.

Bit(s)	Function	R/W[1]	Description
7:0	Transmit Data	H:W	When the Host writes to the Transmitter Holding Register, the Z80382 responds by setting the appropriate bit in the IP register and by generating an interrupt to the 380C if it is enabled. On the Host side, this address is used to write both transmit data and the LS Byte of the Divisor Latch. To write transmit data, the DLAB bit in the Line Control Register (LCR[7]) must be zero.
		P:R	When the 380C reads this register, the Transmitter Holding Register Empty flag (LSR[5]) is set (if the transmit timer is enabled, this bit is set after the timer times out).

NOTES:
1. H indicates Host-side capability, P indicates processor-side (380C) capability.

Interrupt Enable Register (IER)

Interrupt Enable Register **Host Address: (Base + 01)h, LCR[7] = 0**
380C Address: 00F1h

BIT	7	6	5	4	3	2	1	0
R W	Reserved, Always 0			I/O Mailbox Interrupt	Modem Status IRQ Enable	Receiver Line Status IRQ Enable	Transmitter Holding Register Empty IRQ Enable	Received Data Avail-able IRQ Enable
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W^{1,2}	Description
7:5	Reserved		These bits are always read as 0 (Host and 380C
4	I/O Mailbox Interrupt Enable	H:R/W P:R	If this bit is 1, accesses to the I/O Mailbox registers by the 380C causes an interrupt request to the Host See IIR[4] for additional information.
3	Modem Status Interrupt Enable	H:R/W P:R	An interrupt to the Host is generated if bit(s) 0,1, 2 or 3 of the Modem Status Register are set and this bit is a 1.
2	Receiver Line Status Interrupt Enable	H:R/W P:R	An interrupt to the Host is generated N bit(s)1, 2, 3 or 4 of the Line Status Register are set and this bit is a1.

Bit(s)	Function	R/W[1,2]	Description
1	Transmitter Holding Register Empty Interrupt Enable	H:R/W P:R	An interrupt to the Host is generated if THRE (LSR[5]) is set and this bit is a 1.
0	Received Data Available Interrupt Enable	H:R/W P:R	An interrupt to the Host is generated if RDA (LSR[0]) is set, or if a Receiver Timeout occurs, and this bit is a 1.

NOTES:

1. H indicates Host-side capability, P indicates processor-side (380C) capability.
2. On the Host side, this address is used to access both the IER and the MS Byte of the Divisor Latch. To access the IER, the DLAB bit in the Line Control Register (LCR[7]) must be set to a zero.

Line Control Register (LCR)

Line Control Register **Host Address: (Base + 03)h**
380C Address: 00F3h

BIT	7	6	5	4	3	2	1	0
R W	Divisor Latch Access Bit	Set Break	Stick Parity	Even Parity	Parity Enable	Number of Stop Bits	Word Length Select	
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W[1]	Description
7	Divisor Latch Access Bit (DLAB)	H:R/W P:R	This bit allows access to the divisor latch by the Host. If DLAB is set to a 1, host access to the THR, the RBR and the IER is disabled. Host operations to address (Base + 00h) then access the Divisor Latch Least Significant byte, and Host operations to address (Base + 01h) then access the Divisor Latch Most Significant byte. These conditions are transposed if DLAB is set to 0. See “MIMIC Host Interface Registers” on page 102
6	Set Break	H:R/W P:R	This bit has no direct control of the 16550 MIMIC interface. The 380C must emulate the function if it is to be implemented. see the 16550 data sheet for a full description of the normal functions of this bit.
5	Stick Parity	H:R/W P:R	This bit has no direct control of the 16550 MIMIC interface. The 380C must emulate the function if it is to be implemented. see the 16550 data sheet for a full description of the normal functions of this bit.

Bit(s)	Function	R/W[1]	Description
4	Even Parity Select	H:R/W P:R	This bit has no direct control of the 16550 MIMIC interface. The 380C must emulate the function if it is to be implemented. See the 16550 data sheet for a full description of the normal functions of this bit.
3	Parity Enable	H:R/W P:R	This bit has no direct control of the 16550 MIMIC interface. The 380C must emulate the function if it is to be implemented. See the 16550 data sheet for a full description of the normal functions of this bit.
2	Number of Stop Bits	H:R/W P:R	This bit has no direct control of the 16550 MIMIC interface. The 380C must emulate the function if it is to be implemented. See the 16550 data sheet for a full description of the normal functions of this bit.
1:0	Word Length Select	H:R/W P:R	This bit has no direct control of the 16550 MIMIC interface. The 380C must emulate the function if it is to be implemented. see the 16550 data sheet for a full description of the normal functions of this bit.

NOTES:

[1] H indicates Host-side capability, P indicates processor-side 380C capability.

Modem Control Register (MCR)

Modem Control Register **Host Address: (Base + 4)h**
380C Address: 00F4h

BIT	7	6	5	4	3	2	1	0
R W	Reserved			Loop	Out 2	Out 1	RTS	DTR
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W[1]	Description
7:5	Reserved		Reserved for future use, always 0.
4	Loop	H:R/W P:R	When this bit is set to a 1, bits in the Modem Status Register are affected as follows MSR Bit How Affected MSR[6] (RI) = Out 1 MSR[7] (DCD) = Out 2 MSR[5] (DSR) = DTR MSR[4] (CTS) = RTS Emulation of the loopback feature of the 16550 UART must be done by the 380C except for the above.
3	Out 2	H:R/W P:R	This bit has no direct control of the 16550 MIMIC interface. The 380C must emulate the function if it is to be implemented. see the 16550 data sheet for a full description of the normal functions of this bit.
2	Out 1	H:R/W P:R	This bit has no direct control of the 16550 MIMIC interface. The 380C must emulate the function if it is to be implemented. See the 16550 data sheet for a full description of the normal functions of this bit.

Bit(s)	Function	R/W[1]	Description
1	RTS	H:R/W P:R	This bit has no direct control of the 16550 MIMIC interface. The 380C must emulate the function if it is to be implemented. see the 16550 data sheet for a full description of the normal functions of this bit.
0	DTR	H:R/W	This bit has no direct control of the 16550 MIMIC interface. The 380C must emulate the function if it is to be implemented. See the 16550 data sheet for a full description of the normal functions of this bit.

NOTES:

1. H indicates Host-side capability, P indicates processor-side (380C) capability.

Line Status Register (LSR)

Line Status Register **Host Address: (Base + 05)h**
380C Address: 00F5h

BIT	7	6	5	4	3	2	1	0
R W	Error In Receiver FIFO	Trans-mitter Empty	Trans-mitter Holding Register Empty	Break Interrupt (BI)	Framing Error	Parity Error	Overrun Error	Received Data Avail-able
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W[1]	Description
7	Error in Receiver FIFO	H:R P:R	In 16450 mode, this bit is read as 0. In 16550 mode, this bit is set if at least one data byte is available in the FIFO with one of its associated error bits set. This bit clears when there are no more errors (or break detects) in the FIFO.
6	Transmitter Empty	H:R P:R/W	If Double Buffer Mode is enabled, this bit is set automatically by hardware whenever both the THR buffer and the TSR are empty. In other modes, this bit should be set or reset by the 380C.
5	Transmitter Holding Register Empty (THRE)	H:R P:R	This bit is automatically set to 1 when either the THR has been read (emptied) by the 380C (16450 mode) or the Transmitter FIFO is empty (16550 mode). It is set to 0 when either the THR or Transmitter FIFO become non-empty. A shadow bit exists so that the setting of the register bit to a 1 is delayed by the Transmitter Timer, if it is enabled. When reading this bit the 380C does not detect the delay. Both shadow and register bits are cleared when the Host writes to the THR of the Transmitter FIFO.[2]

Bit(s)	Function	R/W[1]	Description
4:2	Break Detect Framing Error Parity Error	H:R P:R/W	These bits are written by the 380C indirectly as follows: The bits are first automatically written to shadow bit locations when the 380C writes to the LSR. When the next character is written by the 380C into the RBR or Receiver FIFO, the data in the shadow bits is copied to the LSR (16450 mode) or the FIFO (16550 mode). Then the shadow bits are automatically cleared. In FIFO mode, the bits become available to the Host when the data byte associated with the bits is next to be read (top of FIFO). The register bits are latched and are cleared only when the Host reads the LSR.
1	Overrun Error	H:R P:R	This bit is set if the 380C writes two bytes to the Receiver Buffer before the Host reads the data in the Buffer (16450 mode) or with a full Receiver FIFO (16550 mode). No data is transferred to the Receiver FIFO under these circumstances. This bit is reset when the Host reads the LSR
0	Received Data Available	H:R P:R	This bit is set to 1 when received data is available either in the Receiver FIFO (16550 mode) or in the Receiver Buffer Register (16450 mode). This bit is set immediately when the 380C writes data to the RBR or FIFO if the Receiver Timer is not enabled but is delayed by the timer interval if the Receiver Timer is enabled A shadow bit exists for 380C read access, so that the 380C does not see the delay (only the Host does). Both bits are cleared to zero immediately upon reading all the data in the Receiver Buffer or the FIFO.

NOTES:

1. H indicates Host-side capability, P indicates processor-side (380C) capability
2. The THRE bit is forced to 0 on the Host side to prevent THR overruns when $\overline{\text{HALT}}$ is asserted (Power-Down Modes) and bit 3 ($\overline{\text{INT0}}$ assertion on MIMIC access feature) is set. When the MIMIC comes out of power-down ($\overline{\text{HALT}}$ deasserted), the THRE bit resumes normal functionality.

Modem Status Register (MSR)

Modem Status Register **Host Address. (Base + 06)h**
380C Address: 00F6h

BIT	7	6	5	4	3	2	1	0
R W	Data Carrier Detect	Ring Indicator	Data Set Ready	Clear To Send	Delta Data Carrier Detect	Trailing Edge Ring Indicator	Delta Data Set Ready	Delta Clear To Send
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W[1]	Description
7	Data Carrier Detect	H:R P:R/W	See the 16550 data sheet for a full description of the normal functions of this bit. This bit must be written to the desired state by the 380C.
6	Ring Indicator	H:R P:R/W	See the 16550 data sheet for a full description of the normal functions of this bit. This bit must be written to the desired state by the 380C.
5	Data Set Ready	H:R P:R/W	See the 16550 data sheet for a full description of the normal functions of this bit. This bit must be written to the desired state by the 380C.
4	Clear to Send	H:R P:R/W	See the 16550 data sheet for a full description of the normal functions of this bit. This bit must be written to the desired state by the 380C.
3	Delta Data Carrier Detect	H:R P:R/W	This bit is set to 1 whenever the Data Carrier Detect bit (bit 7) changes state, and is reset when the Host reads the Modem Status Register.

Bit(s)	Function	R/W[1]	Description
2	Trailing Edge Ring Indicator	H:R P:R/W	This bit is set to 1 when the Ring Indicator bit (bit 6) changes from 1 to 0 and is reset when the Host reads the Modem Status Register.
1	Delta Data Set Ready	H:R P:R/W	This bit is set to 1 whenever the Data Set Ready bit (bit 5) changes state and is reset when the Host reads the Modem Status Register.
0	Delta Clear To Send	H:R P:R/W	This bit is set to 1 whenever the Clear To Send bit (bit 4) changes state and is reset when the Host reads the Modem Status Register.

NOTES:

1. H indicates Host-side capability, P indicates processor-side (380C) capability.

Scratch Register (SCR)

Scratch Register **Host Address: (Base + 07)h**
380C Address: 00F7h

BIT	7	6	5	4	3	2	1	0
R W	Scratch Register							
DEF	X	X	X	X	X	X	X	X

Bit(s)	Function	R/W[1]	Description
7:0	Scratch Register	H:R/W P:R	This register is used by the Host programmer for temporary data storage. If the Host writes to this register, no interrupt is generated to the 380C.

NOTES:

1. H indicates Host-side capability, P indicates processor-side (380C) capability.

Divisor Latch LSB (DLL)

Divisor Latch LSB **Host Address: (Base + 00)h, LCR[7] =1**
380C Address: 00F8h

BIT	7	6	5	4	3	2	1	0
R W	Divisor Latch LSB							
DEF	X	X	X	X	X	X	X	X

Bit(s)	Function	R/W[1]	Description
7:0	Divisor Latch Least Significant Byte	H:R/W P:R	This register contains the low order byte of the baud rate divisor. A write to this register by the Host generates an interrupt to the 380C. It can then read the baud rate divisor and set up the application appropriately. On the Host side, this address is used to access the receive data and the LS Byte of the Divisor Latch. To access the divisor latch, the DLAB bit in the Line Control Register (LCR[7]) must be a one.

NOTES:

[1] H indicates Host-side capability, P indicates processor-side (380C) capability.

Divisor Latch MSB (DLM)

Divisor Latch MSB **Host Address: (Base + 01)h, LCR[7] = 1**
380C Address: 00F9h

BIT	7	6	5	4	3	2	1	0
R W	Divisor Latch MSB							
DEF	X	X	X	X	X	X	X	X

Bit(s)	Function	R/W[1]	Description
7:0	Divisor Latch Most Significant Byte	H:R/W P:R	This register contains the high order byte of the baud rate divisor. A write to this register by the Host generates an interrupt to the 380C. It can then read the baud rate divisor and set up the application appropriately. On the Host side, this address is used to access the Interrupt Enable Register (IER) and the LS Byte of the Divisor Latch. To access the divisor latch, the DLAB bit in the Line Control Register (LCR[7]) must be a one.

NOTES:

[1] H indicates Host-side capability, P indicates processor-side (380C) capability.

Interrupt Identification Register (IIR)

Interrupt Identification Register **Host Address: (Base + 02)h**

BIT	7	6	5	4	3	2	1	0
R	11 if FIFO Enabled		Reserved	I/O Mailbox Interrupt	Interrupt Identification			Interrupt Pending
W								
DEF	0	0	0	0	0	0	0	0

When the Host accesses the IIR, the contents of the register and all pending interrupts are frozen. New interrupts are recorded, but not acknowledged during the IIR access.

Bit(s)	Function	R/W[1]	Description
7:6	FIFOs Enabled	H:R	These bits read 1 if FIFO mode is enabled on the MIMIC.
5	Reserved		Always reads as 0.
4	I/O Mailbox Interrupt	H:R	This bit is 0 unless bit 4 of the IER is 1 at the time the 380C accesses one of the I/O Mailbox registers. If IER[4] is 1, this bit is set when the 380C reads from the Port A Direction register, the Port D Data register, or the Port D Direction register, or when it writes to the Port A Data register. This bit is cleared when the Host reads the Host I/O Status register (see “Host I/O Status Register (HIOS)” on page 161) and when IER[4] is 0.

<table>
<thead>
<tr><th>Bit(s)</th><th>Function</th><th>R/W[1]</th><th>Description</th></tr>
</thead>
<tbody>
<tr><td>3:1</td><td>Interrupt Identification</td><td>H:R</td><td>This 3-bit field indicates the highest priority pending MIMIC interrupt.
<table>
<thead>
<tr><th>b3</th><th>b2</th><th>b1</th><th>Priority</th><th>Interrupt Source</th><th>Interrupt Reset Control</th></tr>
</thead>
<tbody>
<tr><td>0</td><td>1</td><td>1</td><td>Highest</td><td>Overrun, Parity, Framing Error or Break Detect bits set by 380C.</td><td>Read Line Status Register</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>2nd</td><td>Received Data Trigger Level</td><td>Receiver FIFO drops below trigger level</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>3rd</td><td>Receiver Timeout with data in Receiver FIFO.</td><td>Read Receiver FIFO.</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>4th</td><td>Transmitter Holding Register Empty</td><td>Read the IIR or write to the Transmitter Holding Register</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Lowest</td><td>Modem Status: CTS, DSR, RI or DCD</td><td>Read the Modem Status Register</td></tr>
</tbody>
</table>
</td></tr>
<tr><td>0</td><td>Interrupt Pending</td><td>H:R</td><td>This bit is 0 if an interrupt is pending.</td></tr>
</tbody>
</table>

NOTES:

[1] H indicates Host-side capability, P indicates processor-side (380C) capability.

HOST DMA MAILBOX

Introduction

The Host DMA Mailbox facility provides a path for Host DMA data transfers separate from the MIMIC COM port. Commands and data flow through the COM port, while the DMA path can be used for other purposes. The Host DMA Mailbox feature includes control registers that allow Host DMA data transfer between Host memory and, for example, a modem speaker/microphone CODEC. Transfers are driven by the Host's

DMA on one side; Z80382 DMA channels) or programmed I/O can be used on the other side. Several modes of operation can be programmed:

- Host DMA Write, Z80382 Polled Input
- Host DMA Read, Z80382 Polled Output
- Host DMA Write with Z80382 DMA
- Host DMA Read with Z80382 DMA

On the ISA bus, the Z80382 can use two independent DMA Mailbox facilities. When a facility is enabled in the Plug-and-Play module, that module signals a DMA request by driving HDREQ0 or HDREQ1 High; if a facility is disabled, the corresponding HDREQ pin is 3-stated. A Low on one of the Acknowledge signals, $\overline{\text{HDACK0}}$ or $\overline{\text{HDACK1}}$, more or less simultaneously with $\overline{\text{HWR}}$ or $\overline{\text{HRD}}$ Low when the corresponding HDREQ line is being driven High, indicates a DMA cycle.

In a PCMCIA socket, only one DMA Mailbox can be used, Mailbox 0 in the following descriptions. When an option bit in one of the PCMCIA registers is 1, a DMA request is signalled by setting the $\overline{\text{INPACK}}$ output Low. A DMA cycle is signalled by having the $\overline{\text{PCREG}}$ line High while $\overline{\text{PCIORD}}$ or $\overline{\text{PCIOWR}}$ goes Low.

The following descriptions use the term `assert the DMA Request` to indicate driving HDREQ0 HDREQ1, or INPACK to its active level, and the term `DMA Acknowledge` to indicate the active state on $\overline{\text{HDACK0}}$, $\overline{\text{HDACK1}}$, or $\overline{\text{PCREG}}$

Host DMA Mailbox Operation Overview

The Host DMA Mailbox Control Register (HMC) contains a request bit (HDREQn) for each DMA channel. Write Enable bits are provided to simplify bit manipulations; read-modify-write cycles are not required.

380C software sets an HDREQ bit, using an OUT0 instruction, to request a DMA transfer. If HDREQ is 1, the corresponding DMA Mailbox to assert its DMA Request. If the DMA Mailbox is enabled in the Plug-and-

Play or PCMCIA modules, the associated output pin is also asserted. Z80382 hardware clears the HDREQ bit when the byte transfer is complete. The bit can also be cleared by 380C software. The 380C can read the bit to detect when the transfer is complete.

There are four separate 8-bit data registers in the Host DMA Mailbox facility. HDMAT0 and HDMAT1 can be written by the two possible Host DMA channels, and can be read by a Z80382 DMA Channel or by 380C software. HDMAR0 and HDMAR1 can each be written by a Z80382 DMA channel or by 380C software, and can be read by the two possible Host DMA channels. These four registers comprise the read and write sides of 380C I/O addresses `00D0h` and `00D1h`, per the register descriptions. They have no addresses in Host I/O space, but are implicitly referenced by the DMA Acknowledge line being asserted during a read or write cycle.

The implementation of two sets of DMA Request, Enable, and Acknowledge signals and controls permits the Z80382 to use either of two Host DMA channels without external jumpers.

Host DMA Write, Z80382 Polled Input Operation

Through commands exchanged over the MIMIC COM port, Host software and the 380C agree to configure a host DMA channel (the one connected to the DMA Request and Acknowledge lines of a DMA Mailbox) for a Host DMA write, in which data flows from Host memory to the Z80382. The Host sets up its DMA controller in auto-initialize mode to ensure that data is always available whenever the Z80382 makes a DMA request. The 380C clears the HDREQn register bit so that the DMA Request is negated.

When a Z80382 DMA channel is not used to handle the data, a Host DMA Write proceeds as follows:

1. 380C software writes a 1 to an HDREQn bit, asserting the DMA Request output to the Host.

2. The Host DMA controller begins a memory-read, I/O-write DMA bus cycle at the end of the current Host bus cycle.

3. The Host places valid data on HD[7:0] and asserts $\overline{\text{HWR}}$ or $\overline{\text{PCIOWR}}$ low with DMA Acknowledge asserted.

4. Z80382 hardware negates its DMA Request on the leading edge of DMA Acknowledge logically ANDed (positive logic ORed) with $\overline{\text{HWR}}$ or $\overline{\text{PCIOWR}}$ The HDREQn register bit is not cleared yet.

5. Z80382 hardware latches data into the HDMATn register on the trailing edge of DMA Acknowledge logically ANDed with $\overline{\text{HWR}}$ or $\overline{\text{PCIOWR}}$ On this same edge, hardware clears the HDREQn bit.

6. 380C software polls the HDREQn bit until it is 0, and then reads the data byte from the HDMATn register. Host software should program its DMA controller to stop DMA at the end of valid data or when commanded by the application. When the 380C software sets the HDREQn bit but doesn't see it cleared fairly quickly (because the Host DMA channel isn't active), software can cancel the request by resetting the HDREQn bit.

7. Messages on the COM port confirm that the Host DMA write operation is complete. 380C software clears HDREQn if necessary.

Host DMA Read, Z80382 Polled Output Operation

Through commands exchanged over the MIMIC COM port, Host software and the 380C agree to configure a Host DMA channel (the one connected to the DMA Request and Acknowledge lines of a DMA Mailbox) for a Host DMA Read, in which data flows from the Z80382 to Host memory. The Host sets up its DMA controller in auto-initialize mode to ensure that there is buffer space available whenever the Z80382 makes a DMA request. The 380C clears the HDREQn register bit, ensuring that the DMA Request is negated.

When a Z80382 DMA channel is not used to handle the data, a Host DMA Read proceeds as follows:

1. 380C software writes a data byte to the HDMARn register and then writes a 1 to the HDREQn bit, asserting the DMA Request.
2. The Host DMA controller begins an I/O-read, memory-write DMA bus cycle at the end of the current Host bus cycle.
3. The Host asserts $\overline{\text{HRD}}$ or $\overline{\text{PCIORD}}$ Low with DMA Acknowledge asserted. In response, the Z80382 drives the byte in the HDMARn register onto the HD[7:0] pins, negates the DMA Request, and on an ISA bus drives $\overline{\text{HDOEN}}$ Low. The HDREQn bit is not cleared yet.
4. While Z80382 hardware holds the data on HD[7:0], the Host performs its memory-write half of the DMA cycle. When the Host negates DMA Acknowledge and/or $\overline{\text{HRD}}$ or $\overline{\text{PCIORD}}$ the Z80382 releases HD[7:0], clears the HDREQn bit, and on an ISA bus drives $\overline{\text{HDOEN}}$ high.
5. 380C software polls the HDREQn bit until it is 0, then continues with step 1 for the next byte.

Software can cancel a request if the Host DMA channel does not respond to it in a timely manner. If 380C software fails to detect HDREQn = 0 after setting it, it clears HDREQn.

Host DMA Write with Z80382 DMA Operation

When a Z80382 DMA channel is used to store DMA Mailbox data from the Host into Z80382 memory, the, data transfer process proceeds as follows:

1. 380C software writes a 1 to an HDREQn bit, asserting the DMA Request to the Host, and then writes a 1 to the corresponding DMA Enable bit in the HDCR.
2. The Host DMA controller begins a memory-read, I/O-write DMA bus cycle at the end of the current Host bus cycle.

3. The Host places valid data on HD[7:0] and asserts $\overline{\text{HWR}}$ or $\overline{\text{PCIOWR}}$ low with DMA Acknowledge asserted.
4. Z80382 hardware negates its DMA Request on the leading edge of DMA Acknowledge logically ANDed with $\overline{\text{HWR}}$ or $\overline{\text{PCIOWR}}$ The HDREQn register bit is not cleared yet.
5. Z80382 hardware latches data into the HDMATn register on the trailing edge of DMA Acknowledge logically ANDed with $\overline{\text{HWR}}$ or $\overline{\text{PCIOWR}}$ On this same edge, the hardware clears the HDREQn bit.
6. The clearing of the HDREQn bit causes the DMA Mailbox facility to request a Z80382 DMA transfer from its associated Z80382 DMA channel.
7. When the Z80382 DMA transfer is acknowledged, the DMA Mailbox facility places the data from the HDMATn register on the Z80382's D[7:0] lines, from whence it is written into Z80382 memory.
8. The DMA mailbox facility negates its Z80382 DMA request.
9. When the Z80382 DMA transfer is over, the DMA Mailbox facility asserts DMA Request to the Host, sets the HDREQn bit, and returns to step 2.
10. Messages on the COM port confirm that the Host DMA write operation is complete. 380C software clears HDREQn if necessary.

Host DMA Read with Z80382 DMA Operation

When a Z80382 DMA channel is used to provide DMA Mailbox data from Z80382 memory to the Host, software clears the HDREQn bit as described, and starts the associated Z80382 DMA channel. The data transfer process then proceeds as follows:

1. Software writes a 1 to the enable bit in the HDCR, without first setting HDREQn as in the write process.

2. The Z80382 DMA channel reads data from 284382 memory and places it into the HDMATn register. This action causes the DMA Mailbox facility to set the HDREQn bit to 1, asserting the DMA Request to the Host.
3. The Host DMA controller begins an I/O-read, memory-write DMA bus cycle at the end of the current Host bus cycle.
4. The Host asserts $\overline{\text{HRD}}$ or $\overline{\text{PCIORD}}$ Low with DMA Acknowledge asserted. In response, the Z80382 drives the byte in the HDMARn register onto the HD[7:0] pins, negates the DMA Request, and on an ISA bus drives $\overline{\text{HDOEN}}$ Low. The HDREQn bit is not cleared yet.
5. While Z80382 hardware holds the data on HD[7:0], the Host performs its memory-write half of the DMA cycle. When the Host negates DMA Acknowledge and/or $\overline{\text{HRD}}$ or $\overline{\text{PCIORD}}$ the Z80382 releases HD[7:0], clears the HDREQn bit, and on an ISA bus drives $\overline{\text{HDOEN}}$ High.
6. The clearing of HDREQn causes the DMA channel to read another byte, as in step 2, unless its buffer count has expired.

Host DMA Mailbox Registers

The Host DMA Mailbox module contains a set of registers for programming various aspects of its operation. Table 15 lists these registers.

Table 15. Host DMA Mailbox Registers

Register	Host Address[1]	380C I/O Address
Host DMA Rx Register 0	See note[1]	00D0h
Host DMA Tx Register 0	See note[1]	00D0h
Host DMA Rx Register 1	See note[1]	00D1h
Host DMA Tx Register 1	See note[1]	00D1h
Host DMA Mailbox Control Register	—	00D2h
Host DMA Control Register	—	00E6h

NOTES:

1. On the host side, these registers are implicitly addressed by the DMA acknowledge signals $\overline{\text{HDAK0}}$ and $\overline{\text{HDAK1}}$ for registers 0 and 1 respectively.

Host DMA Mailbox Register Descriptions

The following sections describe the registers associated with the Host DMA Mailbox module. They are ordered by ascending 380C I/O address. In the descriptions, the values described in the line labeled DEF are the default values after a reset. X indicates that the reset value is indeterminate. XXXXh is the register’s address in the 380C I/O space.

Host DMA Receive Register 0 (HDMAR0)

Host DMA Receive Register 0 **Host Address: Implicit**
380C Address: 00D0h

BIT	7	6	5	4	3	2	1	0
R	Host Reads Data From 380C							
W	380C Writes Data To Host							
DEF	X	X	X	X	X	X	X	X

Bit(s)	Function	R/W[1]	Description
7:0	Host Reads Data From 380C	H:R	Contains the data byte from the 380C to the Host. The Host addresses this register implicitly via the DMA acknowledge cycle signaling.
	380C Writes Data To Host	P:W	The 380C writes the data to the Host into this register using either an I/O write cycle or a Z80382 DMA cycle.

NOTES:

1. H indicates Host-side capability, P indicates processor-side (380C) capability.

Host DMA Transmit Register 0 (HDMAT0)

Host DMA Transmit Register 0 HDMAT0 | **Host Address: Implicit**
380C Address: 00D0h

BIT	7	6	5	4	3	2	1	0
R	380C Reads Data From Host							
W	Host Writes Data To 380C							
DEF	0	0	1	1	1	0	0	0

Bit(s)	Function	R/W[1]	Description[2]
7:0	Data: 380C to Host	H:W	The Host writes the data to the 380C into this register using a Host DMA cycle. The Host addresses this register implicitly via the DMA acknowledge cycle signaling.
		P:R	Contains the data byte from the Host to the Z80382. The initial content of this register is preset by Zilog and identifies the type and revision of the chip. A 1 in HMC[6] indicates that this data is available to be read by the 380C. Reading the data resets HMC[6].

NOTES:

1. H indicates Host-side capability, P indicates processor-side (380C) capability.
2. Depending on the revision, the default data may differ from that indicated in the diagram above.

Host DMA Receive Register 1 (HDMAR1)

Host DMA Receive Register 1 **Host Address: Implicit**
380C Address: 00D1h

BIT	7	6	5	4	3	2	1	0
R	Host Reads Data From 380C							
W	380C Writes Data To Host							
DEF	X	X	X	X	X	X	X	X

Bit(s)	Function	R/W[1]	Description
7:0	Data: 380C to Host	H:R	Contains the data byte from the 380C to the Host. The Host addresses this register implicitly via the DMA acknowledge cycle signaling.
		P:W	The 380C writes the data to the Host into this register using either an I/O write cycle or a Z80382 DMA cycle.

NOTES:

1. H indicates Host-side capability, P indicates processor-side (380C) capability.

Host DMA Transmit Register 1 (HDMAT1)

Host DMA Transmit Register 1

Host Address: Implicit
380C Address: 00D1h

BIT	7	6	5	4	3	2	1	0
R	380C Reads Data From Host							
W	Host Writes Data To 380C							
DEF	0	0	1	1	1	0	0	0

Bit(s)	Function	R/W[1]	Description[2]
7:0	Data: 380C to Host	H:W	The Host writes the data to the 380C into this register using a Host DMA cycle. The Host addresses this register implicitly via the DMA acknowledge cycle signaling.
		P:R	Contains the data byte from the Host to the Z80382. The initial content of this register is preset by Zilog and identifies the type and revision of the chip. A 1 in HMC[7] indicates that this data is available to be read by the 380C. Reading the data resets HIVIC[7].

NOTES:

1. H indicates Host-side capability, P indicates processor-side (380C) capability.
2. Depending on the revision, the default data may differ from that indicated in the diagram above.

Host DMA Mailbox Control Register (HMC)

Host DMA Mailbox Control Register (HMC) **380C Address: 00D2h**

BIT	7	6	5	4	3	2	1	0
R	HDMAT1 Data Valid	HDMAT0 Data Valid	HDRE01	HDREQ0	Reserved			
W	Write Enable HDREQ1	Write Enable HDREQ0						
DEF	1	1	0	0	0	0	0	0

Bit(s)	Function	R/W	Description
7	HDMAT1 Data Valid	R	A 1 in bit 7 indicates the availability of device identification data in the HDMAT1 register. When this register is read by the 380C, this bit is reset and the data is no longer available. The identification data is preset by Zilog and identifies the type and revision of the product.
	Write Enable HDREQ1	W	This bit must be a 1 in the data byte associated with the write to this register in order to allow bit 5 of the data byte to be written into HDREQ1. If this bit in the data byte is 0 during the write operation, HDREQ1 is not affected. This bit has no affect on the ability to write into HDREQ0 (bit 4). This bit is not latched and must be set for each HDREQ1 write.

Bit(s)	Function	R/W	Description
6	HDMAT0 Data Valid	R	A 1 in bit 6 indicates the availability of device identification data in the HDMAT0 register. When this register is read by the 380C, this bit is reset and the data is no longer available. The identification data is preset by Zilog and identifies the type and revision of the product.
	Write Enable HDREQ0	W	This bit must be a 1 in the data byte associated with the write to this register in order to allow bit 4 of the data byte to be written into HDREQ0 If this bit in the data byte is 0 during the write operation, HDREQ0 is not affected. This bit has no affect on the ability to write into HDREQ1 (bit 5). This bit is not latched and must be set for each HDREQ0 write.
5	$\overline{\text{HDREQ1}}$	R/W	Writing a 1 to this bit, with bit 7 of the data byte also a 1, sets the bit and makes the DMA Mailbox assert its DMA Request. If the DMA Mailbox is enabled in the Plug-and-Play or PCMCIA modules, that module asserts the corresponding request line to the Host. Z80382 hardware clears this bit when the byte transfer is complete, and on reset. Writing a 1 to bit 7 and a 0 to this bit clears the bit and prevents the DMA Request from being asserted.
4	HDREQ0	R/W	Writing a 1 to this bit, with bit 6 of the data byte also a 1, sets the bit and makes the DMA Mailbox assert its DMA Request. If the DMA Mailbox is enabled in the Plug-and-Play or PCMCIA modules, that module asserts the corresponding request line to the Host. Z80382 hardware clears this bit when the byte transfer is complete, and on reset. Writing a 1 to bit 6 and a 0 to this bit clears the bit and prevents the DMA Request from being asserted.
3:0	Reserved		These bits are reserved and programmed as 0.

Host DMA Control Register (HDCR)

Host DMA Control Register (HDCR) **380C Address: 00E6h**

<table>
<tr><td>BIT</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>R
W</td><td>Host DMA1 DMA Enable</td><td colspan="3">Host DMA1 DMA Channel</td><td>Host DMA0 DMA Enable</td><td colspan="3">HOST DMA0 DMA Channel</td></tr>
<tr><td>DEF</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

Bit(s)	Function	R/W	Description
7	Host DMA1 DMA Enable	R/W	A 1 enables Z80382-side DMA operation for Host DMA transfers using channel 1 registers.
6:4	Host DMA1 DMA Channel	R/W	Selects the Z80382 DMA channel for Host DMA transfers using channel 1 registers.
3	Host DMA0 DMA Enable	R/W	A1 enables Z80382-side DMA operation for Host DMA transfers using channel 0 registers.
2:0	Host DMA0 Channel	R/W	Selects the Z80382 DMA channel for Host DMA transfers using channel 0 registers.

Note: Because the data flowing through the DMA Mailbox has no organization that can be discerned by the hardware, the Host DMA Mailbox does not use any of the following features of the DMA channels: *with Command*, *Notify at End of Buffer*, or *With Status*.

HOST INPUT/OUTPUT MAILBOX

This facility provides an additional data path between the Host and the Z80382. The Plug-and-Play ISA or PCMCIA modules provide qualified read and write strobes to the I/O Mailbox. In ISA mode, the I/O Mailbox resides in I/O space, while in PCMCIA mode it resides in Attribute space.

Four registers are accessible to the Host. One register provides status, three registers provide a data path from the Host to the Z80382, and one of the latter also provides for data transfers from the Z80382 to the Host. Because the MIMIC and PCMCIA pins are multiplexed with Ports A and D on the Z80382, the Port A and D Data and Direction Registers are used as the data registers for the I/O Mailbox as described in Table 16.

Table 16. Port and I/O Mailbox Functions

Port Function	I/O Mailbox Function
D Data	Data: Host → 380
A Direction	Data: Host → 380
A Data	Data: 380 → Host
D Direction	Data: Host → 380

I/O Mailbox Interrupts

On the Z80382, I/O Mailbox activity can cause interrupts on both the Host and 380C sides. Host interrupts are enabled by bit 4 in the IER (see “Interrupt Enable Register (IER)” on page 119) and identified by bit 4 of the IIR (see “Interrupt Identification Register (IIR)” on page 143), neither of which bit positions are used in the 165X0 family. When Host interrupts are enabled, they occur when the 380C reads any of the three mailbox registers that the Host can write, and when the 380 writes the mailbox register that the Host can read. Bit 4 of the IIR, and the interrupt request, are cleared when the Host reads the Host I/O Status register.

380C I/O Mailbox interrupts are enabled by bit 4 of the IOBRG register (see “I/O and BRG Control Register (IOBRG)” on page 104). When this bit is 1, an interrupt request through the Assigned Vector facility is requested when the Host writes any of the 3 mailbox registers that the 380C can read, or when the Host reads the mailbox register that the 380C can write. The interrupt request is cleared when the 380C reads the Host I/O Status register, and whenever IOBRG[4] is 0.

Host I/O Mailbox Registers

The Host I/O Mailbox module contains a set of registers for programming various aspects of its operation. These are:

I/O Mailbox Function	Host ISA	Address[1] PCMCIA	380C I/O Address
Host I/O Status Register	02h	0204h	00D5h
Data: Host → 380C	03h	0206h	00E7h
Data: Host → 380C	00h	0200h	00E8h
Data: Host → 380C	01h	0202h	00EDh
Data: 380C → Host	01h	0202h	00EEh

NOTES:

1. In ISA mode, this Host Address is relative to the I/O Mailbox base address decoded by the PnP module in the Z80382. In PCMCIA mode, the registers reside in the attribute memory space, at even addresses starting at 0200h.

Host I/O Mailbox Register Descriptions

The following sections describe the registers associated with the Host I/O Mailbox module. They are ordered by ascending 380C I/O address. In the descriptions,

- The values indicated in the line labeled DEF are the default values after a reset
- X indicates that the reset value is indeterminate
- `XXXXh` is the register's address in the 380C I/O space

Host I/O Status Register (HIOS)

Host I/O Status Register **Host Address: (Base + 2)h**
380C Address: 00D5h

BIT	7	6	5	4	3	2	1	0
R W	Host I/O Mailbox Enable	Auxiliary Bits			PORT A Data Bits Available	PORT A, Direction Available	PORT D Data Available	PORT D Direction Available
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W[1]	Description
7	Host I/O Mailbox Enable	H:R P:R/W	The 380C must write a 1 to this bit to enable the Host I/O Mailbox and make all registers visible to the Host. When this bit is 0, as after a Reset, Host accesses are disabled.
6:4	Auxiliary Bits	H:R P:R/W	These bits can be written by the 380C and can be read by both the Host and the 380C. The Auxiliary Bits have no specific function.
3	Port A Data Available	H:R P:R/W	If the Host I/O Mailbox is enabled by setting bit 7, this bit is set when the 380C writes to the Port A Data Register. A Host read of the corresponding address clears this bit.

Bit(s)	Function	R/W[1]	Description
2	Port A Direction Available	H:R P:R/W	If the Host I/O Mailbox is enabled by setting bit 7, this bit is set when the Host writes to the address corresponding to the Port A Direction Register. It is cleared when the 380C reads the Port A Direction register.
1	Port D Data Available	H:R P:R/W	If the Host I/O Mailbox is enabled by setting bit 7, this bit is set when the Host writes to the address corresponding to the Port D Data register. It is cleared when the 380C reads the Port D Data register.
0	Port D Direction Available	H:R P:R/W	If the Host I/O Mailbox is enabled by setting bit 7, this bit is set when the Host writes to the address corresponding to the Port D Direction register. It is cleared when the 380C reads the Port D Direction register.

NOTES:

1. H indicates Host-side capability, P indicates processor-side (380C capability.

Host I/O Mailbox Data Transfer Registers

D Direction (DDRD)

PCMCIA Host Address: 0206h
ISA Host Address. (Base + 03)h
380C Address: 00E7h

BIT	7	6	5	4	3	2	1	0
R	380C Reads Data From The Host							
W	Host Writes Data To The 380C							
DEF	X	X	X	X	X	X	X	X

D Data (DRD)

PCMCIA Host Address: 0200h
ISA Host Address: (Base + 00)h
380C Address: 00E8h

BIT	7	6	5	4	3	2	1	0
R	380C Reads Data From The Host							
W	Host Writes Data To The 380C							
DEF	X	X	X	X	X	X	X	X

A Direction (DDRA)

PCMCIA Host Address: 0202h
ISA Host Address: (BASE + 01)h
380C Address: 00EDh

BIT	7	6	5	4	3	2	1	0
R	380C Reads Data From The Host							
W	Host Writes Data To The 380C							
DEF	X	X	X	X	X	X	X	X

A Data (DRA)

PCMCIA Host Address: 0202h
ISA Host Address: (BASE + 01)h
380C Address: 00EEh

BIT	7	6	5	4	3	2	1	0
R	Host Reads Data From The 380C							
W	380C Writes Data To The Host							
DEF	X	X	X	X	X	X	X	X

PCMCIA Interface

Overview

The PCMCIA Interface block integrates all the functions necessary for the operation of I/O interface cards in a socket compliant with PCMCIA Standard 2.0 or 3.0. These are:

- PCMCIA Interface Control
- Attribute Memory
- Configuration Registers
- I/O Interface
- Configurable Address Decoder
- Configurable Interrupt Logic
- Z380 Interface

PCMCIA I/O Interface Control

The I/O interface contains the main functionality of the PCMCIA block. The interface decodes addresses for I/O accesses by the Host according to the PCMCIA standard. The Host writes to the Configuration Option Register an index to select the base address of the desired I/O address range. After configuration, I/O accesses to this address range are recognized, and the MIMIC chip select is asserted when an I/O access is performed to an address in the configured address range. Figure 12 illustrates the PCMCIA Interface block diagram.

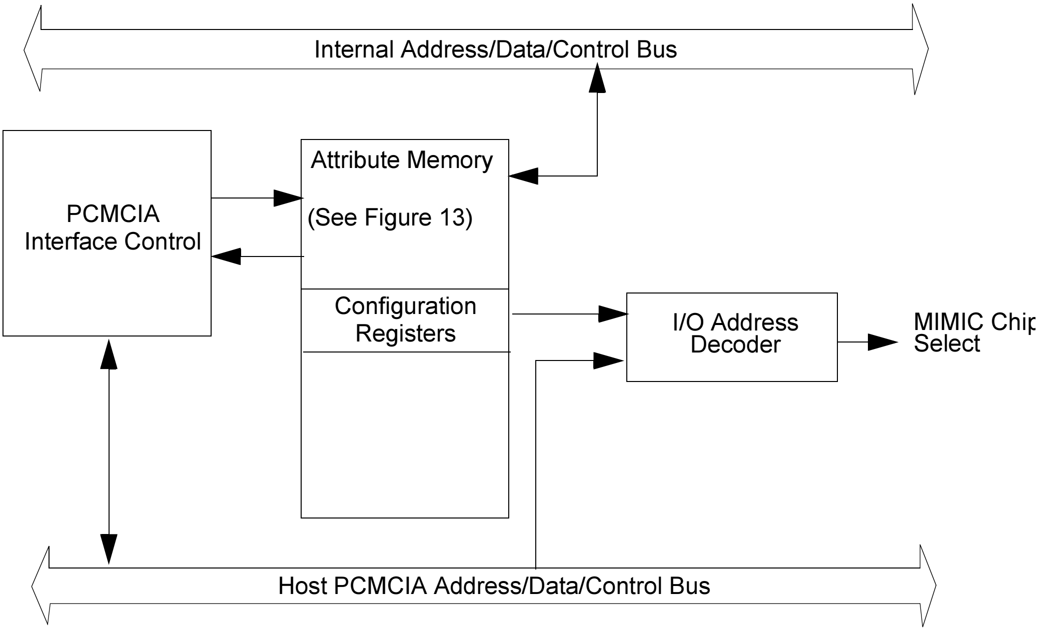


Figure 12. PCMCIA Interface Block Diagram

This section refers to the $\overline{\text{ACK}}$ signal at several points. This signal is an internal signal which is Low when the address applied to HA9-3 is within the configured address range.

I/O Chip Select (MIMICE)

$\overline{\text{MIMICE}}$ is the chip select signal for the 165x0 MIMIC. In case of a valid I/O access (read or write) to the configured address range the logic combination of the input signals $\overline{\text{PCREG}}$ and $\overline{\text{PCCE1}}$ is passed on as $\overline{\text{MIMICE}}$. Otherwise $\overline{\text{MIMICE}}$ remains High:

$$\overline{\text{MIMICE}} = \overline{\text{PCREG}} \texttt{ or } \overline{\text{PCCE1}} \texttt{ or } \overline{\text{ACK}}$$

INPACK Signal

$\overline{\text{INPACK}}$ is driven Low to indicate to the PCMCIA Host that an I/O read access is valid:

$$\overline{\text{INPACK}} = \overline{\text{PCIORD}} \texttt{ or } \overline{\text{PCREG}} \texttt{ or } \overline{\text{PCCE1}} \texttt{ or } \overline{\text{ACK}}$$

Attribute Memory

The attribute memory is the primary mechanism for transfers of configuration data and status between the host system and the PCMCIA card. As illustrated in Figure 13, the attribute memory is segmented into several sections. The Card Information section is 240 bytes of RAM which is loaded by the 380C with information describing the card and its resource requirements, data needed by the Host to configure the card. A portion of the attribute memory allows the host to access the I/O Mailbox registers. Lastly, sections in the attribute memory space are assigned to the Configuration Registers and the Base Address Registers. On the Host side, attribute memory is accessible only on even byte addresses. On the 380C side it can be accessed as bytes or words.

The attribute memory has to be reloaded via the Z380 interface in case of a hardware reset caused by RESET or a soft reset caused by SRESET in the COR (see “Register Descriptions” on page 172).

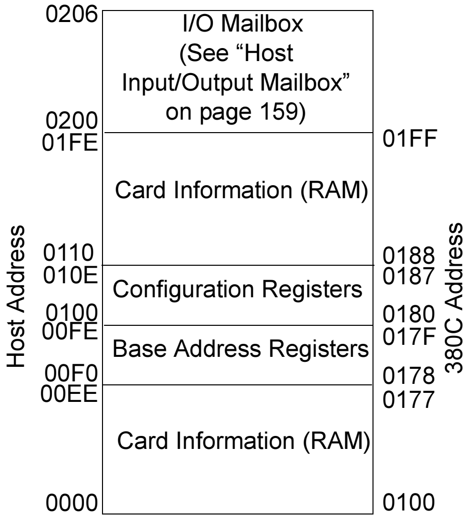


Figure 13. PCMCIA Attribute Memory Organization

Host Read Access to Attribute Memory

The Host applies an address of the attribute memory via the address lines HA9-1 and selects an access to the card with $\overline{\text{PCCE1}}$, $\overline{\text{PCREG}}$, and $\overline{\text{PCOE}}$ all Low.

Function Mode	REG	CE1	OE	WE	HD7-0
STANDBY Mode	X	H	X	X	High-Z
Byte Access 8-bits	L	L	L	H	Even Byte
Byte Access 16-bits	L	L	L	H	Even Byte
Odd-Byte-Only-Access	L	H	L	H	High-Z

Host Attribute Memory Write

In normal operation the attribute memory is write-protected except for the configuration registers. To write to the CIS range within the attribute memory, write protection must be disabled in the Version Number Register. The Host applies the address of the byte of the attribute memory via the address lines HA9-1 and selects an access to the Attribute Memory by driving $\overline{\text{PCCE1}}$, $\overline{\text{PCREG}}$ and $\overline{\text{PCWE}}$ Low.

The state of HD7-0 when $\overline{\text{PCWE}}$ or $\overline{\text{PCCE1}}$ returns to High (whichever occurs first), is stored in the location specified by the address.

Function Mode	REG	CE1	OE	WE	HD7-0
STANDBY Mode	X	H	X	X	XXX
Byte Access 8-bits	L	L	H	L	Even Byte
Byte Access 16-bits	L	L	H	L	Even Byte
Odd-Byte-Only-Access	L	H	H	L	High-Z

Base Address Registers

These seven registers are written by the 380C with the base addresses of 8-byte windows in the host's I/O address space which the host can use to access registers in the MIMIC.

A base address can be any 8-byte boundary within the 10-bit address range. The upper eight bits of the address are stored in the base address register (for example, base address `03F8h` corresponds to `FEH` in the base address register).

If the selected base address register contains `00`, then the MIMIC chip selection ($\overline{\text{MIMICE}}$) is not qualified by HA9-3, but is asserted whenever $\overline{\text{PCREG}}$ and $\overline{\text{PCCE1}}$ are both Low. This mode relies on the Host PC to provide unique address decoding for each PCMCIA slot/socket. Table 17 describes the base address register addresses for the 380C and Host I/O.

Table 17. Base Address Register Addresses

Register	380C I/O Address	Host I/O Address
Base Address Register 0	0178	00F0
Base Address Register 1	0179	00F2
Base Address Register 2	017A	00F4
Base Address Register 3	017B	00F6
Base Address Register 4	017C	00F8
Base Address Register 5	017D	00FA
Base Address Register 6	017E	00FC
Reserved	017F	00FE

Configuration Registers

There are five configuration registers specified by the PCMCIA 3.0 standard and additionally a version number register, two image base address registers, and the seven base address registers described previously. The Host accesses these registers to configure the interface and to retrieve status.

An additional register, the Z380 Control Register (ZCR), is not part of the register set defined in the standard. This register controls the functions of the PCMCIA block via the Z380 controller. Accessible only to the 380C, it controls access to the attribute memory by the 380C and allows the 380C to signal major status changes to the host. Table 18 describes the configuration register addresses.

Table 18. Configuration Register Addresses

Configuration Register	380C U0 Address	Host I/O Address
Z380 Control Register	017F	—
Configuration Option Reg.	0180	0100
Configuration Status Reg.	0181	0102
Pin Replacement Register	0182	0104
Socket Copy Register	0183	0106
Extended Status Register	0184	0108
Image Base Address Register Lower	0185	010A
Image Base Address Register Upper	0186	010C
Version Number Register	0187	010E

Host Configuration Register Read/Write

Host read access to the configuration registers is similar to read access to the attribute memory as described in “Host Attribute Memory Write” on page 168.

Host write access to the configuration registers differs from the write access to the attribute memory described in “Host Attribute Memory Write” on page 168 only in that writing is not affected by write protection.

380C Interface

After a Reset, the attribute memory, the Configuration registers and the Base Address registers can be read and written by the 380C. After the 380C has loaded the Card Information Structure (CIS), it sets the Ready bit in the Z380 Control Register. This action disables further attribute memory access by the 380C, and allows only read operations to the PCMCIA registers. The 380C can always read and write the Z380 Control Register.

The last 48 bytes of attribute memory (`01D0h` - `01FFh`) can conflict with the MIMIC and parallel ports when these modules are configured for *promiscuous* (all-page) I/O address decoding, but this part of attribute memory typically disappears before these modules are actually placed in operation. However, to avoid conflict, these 48 bytes are not accessible if the MIMIC and ports are configured for all-page decoding.

Card Reset and Load Procedure

After a reset the Host has to wait at least 20ms for setup before access to the card is allowed.

To avoid being accessed, the interface drives the signal $\overline{\text{PCIRQ}}$ Low after reset. As every PCMCIA card is not configured after reset, and is seen as a memory-only card, $\overline{\text{PCIRQ}}$ Low acts as a BUSY signal. After the attribute memory and the available base addresses for the address decoder are loaded via the 380C interface, the busy signal must be set inactive by setting the Z-Ready bit in the Z380 Control Register. The attribute memory can now be read by the Host, and the registers can be accessed.

Decoding and Routing Functions

The PCMCIA interface uses the values programmed in the Configuration Registers to decode a MIMIC chip select when the host I/O address signals match the programmed conditions.

Unlike the Plug-and-Play interface, the PCMCIA interface does not perform any routing functions on interrupt and DMA control signals. These are performed at the PCMCIA socket controller on the host side.

Register Descriptions

The following sections describe the registers associated with the PCMCIA interface module. They are ordered by ascending 380C I/O address. In the descriptions,

- The values shown in the line labeled DEF are the default values after a reset
- X indicates that the reset value is indeterminate
- `xxxxh` is the register's address in the 380C I/O space

Z380 Control Register (ZCR)

Z380 Control Register **380C Address: 017Fh**

BIT	7	6	5	4	3	2	1	0
R W	Z-Ready	Status Change	Reserved					
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W	Description
7	Z-Ready	R/W	This bit is 0 after reset, which means that the PCMCIA module is busy and the attribute memory can be accessed by the Z380. After loading the attribute memory, the Z380 should set this bit to 1 to indicate to the PCMCIA Host that the card is now ready for accesses. Once 380C software does this, it can no longer access attribute memory. Setting this bit to a 1 automatically sets the RRDY/BSY and CRDY/BSY bits, PRR[1] and PRR[5], to a 1.
6	Status Change	R/W	This bit is used to signal card status changes like ring indication for host wake-up. Setting this bit causes the REQATTN bit, ESR[4], to be set. Clearing this bit to 0 does not affect REQATTN. If the wake-up function is enabled by the Host via CCR[4], setting this bit causes a corresponding indication on the PCMCIA STSCHG output pin.
5:0	Reserved		Reserved.

Configuration Option Register (COR)

Configuration Option Register **Host Att. Mem. Address: 0100h**
380C Address: 0180h

BIT	7	6	5	4	3	2	1	0
R W	SRESET	LEVLREQ	CONF	DMAEN	Reserved	Base Address		
DEF	0	0	0	0	0	0	0	0

The configuration option register (COR) is used by the host Host to reset the interface. to configure the PCMCIA interface and may be used by the

Bit(s)	Function	R/W[1]	Description
7	SRESET	H:R/W P:R/W	When SRESET is 1 a reset is initiated. This reset differs from the one forced by a HIGH on the RESET input only by not resetting the SRESET bit. When the Host resets this bit to 0, the PCMCIA interface is in the UNCONFIGURED state.[2] The attribute memory must be reloaded by the 380C and the configuration registers must be reprogrammed. This signal is also active during a hardware reset.
6	LEVLREQ	H:R/W P:R/W	When LEVLREQ is 1, the $\overline{\text{PCIRQ}}$ pin operates as a Level Interrupt. For LEVLREQ 0, the interrupt is a pulse. The width of the interrupt pulse is calculated as: twLEVLREQ (in µS) = 48/BUSCLK in MHz
5	CONF	H:R/W P:R/W	When this bit is a 0, only memory access to the PCMCIA interface is permitted. When set to 1, the PCMCIA interface is configured and I/O access is also permitted.

Bit(s)	Function	R/W[1]	Description
4	DMAEN	H:R/W P:R/W	A 1 in this bit enables DMA Mailbox operation via the PCMCIA interface as described in “Host DMA Mailbox” on page 144. When this bit is 1, INPACK is a Low-active DMA request output to the Host, and DMA cycles are acknowledged by $\overline{\text{PCIORD}}$ or $\overline{\text{PCIOWR}}$ going Low with $\overline{\text{PCREG}}$ High. (Host processor I/O cycles always have $\overline{\text{PCREG}}$ low.)
3	Reserved		Reserved as 0.
2:0	Base Address	H:R/W P:R/W	These three bits select a base address register containing the base address of the 8-byte window to be decoded. D2 D1 D0 Base Address 0 0 0 Base Address Register 0 0 0 1 Base Address Register 1 0 1 0 Base Address Register 2 0 1 1 Base Address Register 3 1 0 0 Base Address Register 4 1 0 1 Base Address Register 5 1 1 0 Base Address Register 6 1 1 1 Reserved

NOTES:

1. H indicates Host-side capability, P indicates processor-side (380C) capability.
2. PCMCIA specification 2.1 describes bits 5-0 of this register as an index, and specifies that bits [5:0] = 0 means “not configured.”

Card Configuration and Status Register (CCR)

Card Configuration And Status Register — **Host Att. Mem. Address: 0102h**
380C Address: 0181h

BIT	7	6	5	4	3	2	1	0
R	Changed	SIGCHG	IOIS8	RIEN	AUDIO	PWRDWN	INTR	Reserved
W	0						0	
DEF	0	0	0	0	0	0	0	0

The Card Configuration and Status Register (CCR) contains information about the status of the interface.

Bit(s)	Function	R/W[1]	Description
7	Changed	H:R P:R	This bit reflects the status of the CRDY/BSY bit in the Pin Replacement Register (PRR[5]), or the status of the REQATTN bit in the Extended Status Register (ESR[4]) when the corresponding enable bit (ESR[0]) is set.
6	SIGCHG	H:R/W P:R/W	If this bit is set High, the STSCHG output signal is enabled and reflects the inverted state of the Changed bit.
5	I0IS8	H:R/W P:R/W	This bit has no function. It can be written and read.
4	RIEN	H:R/W P:R/W	If set, and if the card is configured, the STSCHG pin outputs the STSCHG bit in the Z380 Control Register, ZCR[6].
3	Audio	H:R/W P:R	The Host may set this bit to enable audio signalling. If so, 380C software can enable external components to provide audio signalling on the PCMCIA signal SPKR.

Bit(s)	Function	R/W[1]	Description
2	Power Down	H:R/W P:R/W	The Assigned Vector interrupt facility requests an interrupt whenever the Host changes this bit. Reading the CCR clears an interrupt request for PWRDWN, but does not change the bit itself.
1	Interrupt Status	H:R P:R	The content of INTR reflects the status of the MIMIC interrupt signal which is output on the $\overline{\text{PCIRQ}}$ pin.
0	Reserved		Reserved

NOTES:

[1] H indicates Host-side capability, P indicates processor-side (380C) capability.

Pin Replacement Register (PRR)

Pin Replacement Register **Host Att. Mem. Address: 0104h**
380C Address: 0182h

BIT	7	6	5	4	3	2	1	0
R W	0	0	Crdy/Bsy	0	1	1	Rrdy/Bsy	1
DEF	0	0	0	0	1	1	0	1

The pin replacement register is used to provide the status information which is otherwise provided on the $\overline{\text{PCIRQ}}$ pin (RDY/BSY)

Bit(s)	Function	R/W[1]	Description
7:6	Reserved		
5	Change in Ready/Busy	H:R P:R/W	A High indicates that the state of the RRDY/BSY signal has changed from 0 to 1. It is set automatically when the Host sets the Z-READY bit, ZCR[7], to a 1. The 380C can set and clear this bit by using bit 1 as a 'mask' for these operations. To clear CRDY/BSY, write a 0 to this bit while also writing a 1 to bit 1. To set CRDY/BSY, write a 1 to this bit while also writing a 1 to bit 1.
4:2	Reserved		

Bit(s)	Function	R/W[1]	Description
1	Ready/Busy	H:R P:R/W	The PCMCIA Host reads a 0 while the PCMCIA interface block is busy. It is set to a 1 automatically when the 380C has finished loading the attribute memory and sets the Z-READY bit, ZCR[7], to a 1. It can also be directly set and reset by the processor. This bit also acts as a mask for setting and clearing CRDY/BSY See bit 5 above.
0	Reserved		

NOTES:

1. H indicates Host-side capability, P indicates processor-side (380C) capability.

Socket and Copy Register

The Socket and Copy register is implemented for PCMCIA Hosts expecting this optional PCMCIA register in a PCMCIA card. The register has no function in the Z80382.

Socket And Copy Register **Host Att. Mem. Address: 0106h**
380C Address: 0183h

BIT	7	6	5	4	3	2	1	0
R W	Reserved	No Function, Bits Are Readable And Writable						
DEF	0	0	0	0	0	0	0	0

Extended Status Register (ESR)

Extended Status Register **Host Att. Mem. Address: 0108h**
380C Address: 0184h

BIT	7	6	5	4	3	2	1	0
R W	0	0	0	REQATTN	0	0	0	REQATTN Enable
DEF	0	0	0	0	0	0	0	0

The Extended Status register is used to enable and provide status information of external events which are signalled on the STSCHG signal. REQATTN is used in modem applications for signalling an incoming ring signal.

Bit(s)	Function	R/W[1]	Description
7:5	Reserved		
4	Request Attention	H:R/W P:R	A 1 in this bit indicates that the STSCHG bit in the Z382 Control Register has gone from 0 to 1. Writing a 1 to this bit resets it to 0, writing a 0 has no effect.
3:1	Reserved		
0	Request Attention Enable	H:R/W P:R	Setting this bit to 1 enables the setting of the CHANGED bit in the Card Configuration and Status register (CCR[7]) when the REQATTN bit is set. When this bit is 0 this feature is disabled.

NOTES:

[1] H indicates Host-side capability, P indicates processor-side (380C) capability.

Image Base Address Registers (IBRL, IBRU)

The Image Base Address Registers (IBR) deliver a copy of the configured base address. The register can be read out at addresses 10A and 10C Hexadecimal and are read-only.

Image Base Address Register Lower

Host Att. Mem. Address: 010Ah
380C Address: 0185h

BIT	7	6	5	4	3	2	1	0
R	Base Address Lower							
W								
DEF	X	X	X	X	X	0	0	0

Bit(s)	Function	R/W[1]	Description
7:0	Base Address Lower	H:R P:R	Delivers the least-significant eight bits of the configured, 10-bit, I/O base address.

NOTES:

1. H indicates Host-side capability, P indicates processor-side (380C) capability.

Image Base Address Register Upper

Host Att. Mem. Address: 010Ch
380C Address: 0186h

BIT	7	6	5	4	3	2	1	0
R	Base Address Upper							
W								
DEF	0	0	0	0	0	0	X	X

Bit(s)	Function	R/W[1]	Description
7:0	Base Address Upper	H:R P:R	Delivers the most-significant two bits of the configured, 10-bit, I/O base address.

NOTES:

[1] H indicates Host-side capability, P indicates processor-side (380C) capability.

Interface Version Number Register (IVNR)

Interface Version Number Register — **Host Att. Mem Address: 010Eh**
380C Address: 0187h

BIT	7	6	5	4	3	2	1	0
R	DISWP	Version Number						
W								
DEF	0	0	1	1	0	1	1	1

Bit(s)	Function	R/W[1]	Description
7	Disable Write Protection	H:R/W P:R/W	Setting DISWP to a 1 disables Attribute Memory write protection and allows the Host to write data into the CIS portion of Attribute Memory. When this bit is cleared to 0, the CIS portion of the Attribute Memory is read-only from the Host side.
6:0	Version Number	H:R P:R	This field contains the interface version number. The initial version number for the Z80382 is 037h.

NOTES:

1. H indicates Host-side capability, P indicates processor-side (380C) capability.

PLUG-AND-PLAY INTERFACE

Figure 14 illustrates the Plug-and-Play Interface (PnP) block diagram.

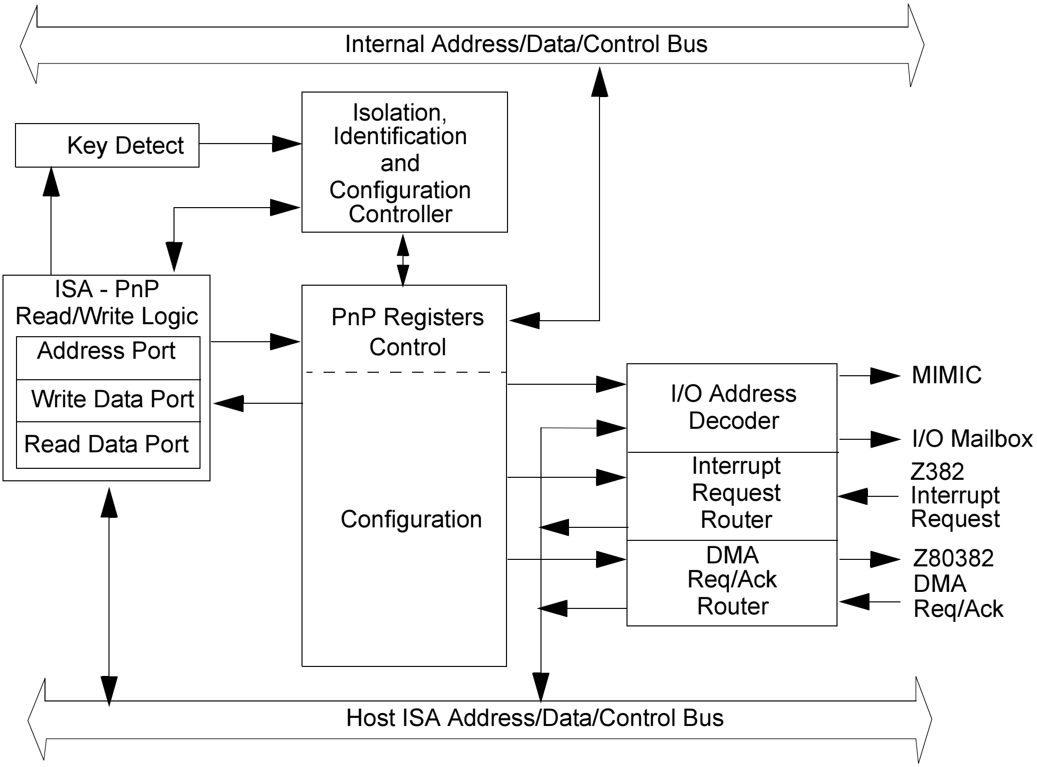


Figure 14. Plug-and-Play Interface Block Diagram

Plug-and-Play Interface Overview

This module, with support from appropriate Z382-based firmware, complies with version 1.0a of the Microsoft™/Intel™ "Plug-and-Play ISA" specification.

The Z80382's PnP module provides for I/O address decoding, interrupt channel selection and DMA channel selection. Pin limitations constrain the internal address decoding for I/O addresses to 12 bits, HA11 - HA0. Since 16-bit decoding is preferred for full Plug-and-Play compliance, an additional input, HAEN, is provided which must be Low for a valid address decode. This permits external decoding of HA15-12 as well as the AEN signal.

ISA Ports

The PnP interface implements three 8-bit ports on the ISA bus:

- The **Address** port is a write-only port at the fixed address 0279h.
- The **Write Data** port is a write-only port at the fixed address 0A79h.
- The **Read Data** port is a read-only port at a programmable address among 0203h, 0207h, 02013h, ..., 03FFh.

The Host may write to the Address port for three reasons:

- As part of sending an Initiation Key to all the PnP cards in the system.
- To select a register on one or all PnP cards as the destination of a subsequent write to the Write Data port.
- To select a register on one card, or the Isolation facility on multiple cards, as the source of data in a subsequent read from the Read Data port.

Basic Host-Side Operation

The Host can always write, in parallel, to the Address ports of all the PnP interfaces in the system. The space that the Host can access by writing to the Address port, and then writing to the Write Data port or reading from the Read Data port, is in large part also accessible to the 380C processor. Its 256 locations are sparsely populated with hardware registers.

After reset, and on command from host software, including in normal operation, the PnP interface is said to be in Wait For Key state. In this state, none of the PnP locations are accessible to the Host on the ISA bus. The Host must first write a specified sequence of 32 values, called an Initiation Key, to the Address port before it can access any of the registers of the PnP interface. When the Initiation key is detected by the PnP interface hardware, the state of the PnP interface is forced to the SLEEP state.

Note: At this point in the process, all PnP cards in the system received the Initiation Key and are in the Sleep state. An isolation mechanism is required for the host to address each card independently. To accomplish this, the host sends a WAKE CSN = 0 command which places all previously unconfigured cards into the ISOLATION state. In this state, an isolation protocol is used to read a unique serial identifier on each card to isolate one Plug-and-Play card at a time.

Each PnP card manufactured must have a non-zero 64-bit identity value that is divided into a 16-bit vendor ID, a 16-bit product ID including revision, and a 32-bit serial number. 380C firmware has complete control of this value; no mechanism for storing or determining it is included in the PnP interface hardware.

During the Isolation protocol, the host selects the PnP card with the 64-bit value having the most low-order 1s, among those in the system. Cards which lose the isolation automatically return to the SLEEP state. The timing requirements of the Isolation protocol are quite slow compared to the speed of the 380C processor, and the 64-bit ID and an associated 8-bit checksum are sequenced to the PnP interface by the 380C on a polled or interrupt-driven basis.

After isolating a single card, host software assigns the isolated card a Card Select Number (CSN), starting with `01h` and ascending for subsequent cards. Assigning a CSN eliminates the card from future repetitions of the protocol. Assigning the CSN also places the card in the

CONFIGURATION state. Then, or later, host software reads the characteristics of the card, called the Resource Data, in a handshake manner with 380C firmware and issues another WAKE CSN = 0 command which places the card back into SLEEP state. At the same time, the unconfigured cards return from SLEEP state to ISOLATION state.

Host software repeats this process until it determines that it has detected all of the PnP cards in the system. Then it allocates resources including memory and I/O space addresses, interrupt levels, and DMA channels. It then puts one card at a time into configuration state by issuing a WAKE CSN = nn command (where `nn` is the card's previously assigned CSN) and writes its allocations to Configuration registers in the PnP register space.

Finally, host software places all the PnP interfaces in the system back in WAIT FOR KEY state, in which they perform address decoding and interface the interrupt and DMA requests and acknowledgments, but have no effect on other system operations. If the host software determines that the system needs reconfiguring, it sends another Initiation Key. In this case, however, it can address a specific card using the previously assigned CSN.

Z382-Side Operation

After Reset

After a Reset, 380C ISA firmware initializes the Configuration registers (`0130h`, `0160h` - `0163h`, `0170h`, `0174h` and `0175h`) to their default values. Initialization of these registers also occurs if host software writes a 1 to bit 0 of the Configuration Control register, which event is recorded in the PnP Master register.

380C PnP Interrupts

The PnP interface requests a 380C processor interrupt via the Assigned Vector facility when the 1E and I P bits are both 1. The Interrupt Service Routine (ISR) reads the PnP Master register, then writes the value back to the Master register to clear IP and any of the event bits (Software Reset, Wake Match, and Isolation Read) that may be 1. Then the ISR examines the event bits and the State field to decide how to proceed. It may also read the status, CSN, or Activate registers, or the RAM.

Isolation Process

If 380C processor. software detects (by a PnP interrupt or by polling) that the State has made the transition from `01` to `10` because the Host has written a zero to the Wake register, that is, because it is beginning an iteration of the Isolation protocol, it resets its pointer into its ID number and checksum. It then writes the first bit of this value to the Isolation Bit in the PnP Master register. If the Isolation Bit is 1, the interface responds actively to a subsequent pair of reads from the Isolation register. Otherwise, R checks the data on HD1-0 in those reads, and goes back to the SLEEP state if another PnP interface responded actively.

If 380C processor software detects that the Host has read the Isolation register and the State is `10`, indicating that it still participates in the Isolation process, and its 72-bit ID and checksum value have not been completely sent, the processor gains access to the next bit of the value, and writes it to the Isolation Bit in the PnP Master register.

Note: The PnP specification defines the minimum time between pairs of reads from the Isolation register as 250 microseconds.

If 380C processor software detects that it has lost the Isolation process because the State is `01`, it remembers this condition internally. The next PnP interrupt occurs when the Host next writes a `0` to the Wake register to start a new Isolation process.

If Z382 processor software detects that it has won an Isolation process because the Host has written a non-zero value to its CSN register, it gains access to the first byte of its Resource data (which follows its Isolation data), writes it to the Resource Data register, and enters the Resource Process as described in "Resource Process" on page 190.

Resource Process

If 380C processor software detects (by a PnP interrupt or by polling) that the State has changed from `01` to `11` because the Host has written a non-zero value to its wake register that matches its CSN register, it resets its pointer to the first byte of its Isolation data, gains access to that byte, and writes it to its Resource Data register.

If 380C processor software detects that the Host has read a byte from the Resource Data register, and it was not the last byte of the Resource data, it gains access to the next byte and writes it to the Resource Data register.

Resource Data Structure

In the terminology used by the Plug-and-Play ISA specification to describe Resource Data, the Resource Data provided by the 380C processor to the Host includes the following:

- Plug-and-Play Version number
- ID string
- Logical Device ID `165x0 UART plus DMA Mailbox plus I/O Mailbox`
- Compatible Device ID `165x0 UART`
- IRQ Format `mask = the two levels to which the HINT pins are connected`, `type = high, edge`
- DMA Format 0: `mask = the level to which HDRQ0 and HDACK0 are connected`

- DMA Format 1: `mask = the level to which HDRQ1 and HDACK1 are connected`
- I/O Port Descriptor: (I/O Mailbox), `10 bit decode`, `Base = 004 to 3FC`, `Alignment = 4`, `Range Length = 4`
- I/O Port Descriptor: `10 bit decode` (MIMIC), `Base = 008 to 3F8`, `Alignment = 8`, `Range length = 8`
- End

In practice, one may offer the host software several alternatives (called by the PnP specification *dependent functions*):

- With and without the I/O Mailbox
- With 0, 1, or 2 DMA Mailboxes
- Preferring the traditional addresses for COM ports for the MIMIC

Configuration Process

The following Configuration registers are implemented in the Z80382 to provide for the resources required by the host to interface to the host-accessible functions within the chip.

- I/O Mailbox I/O Address
- MIMIC I/O Address
- Interrupt Request Level - can be selected to be output on either of the two available interrupt output lines. A unique Z80382 feature allows these two pins to be configured to be any two of the ISA-bus interrupt lines.
- DMA Channel 0, DMA Channel 1 - A unique Z80382 feature allows the two DMA pin pairs to be configured to be any two of the seven ISA-bus DMA channels.

Host writes to the Configuration registers are effective immediately, in hardware, so there is no urgent need for the 380C processor to translate them into other register values. But the 380C processor can use the interrupt that occurs when the Host terminates Configuration state to examine what the Host has done to the Configuration registers, and operate accordingly in the future.

PnP Interface Registers

The PnP Interface module register set consists of three types of registers which are used for programming various aspects of PnP operation:

- Host interface registers which are used to access the control and configuration registers. These registers are accessible only by the Host.
- Plug-and-Play standard control and configuration registers, as described in the PnP Specification, which are used by the Host to isolate, identify and configure the card. Some of these registers are also accessible by the 380C processor. Descriptions of these registers are included at the end of this section. Please consult the PnP Specification for additional information.
- One special control register accessible only by the 380C.

Table 19 summarizes the registers in the PnP Interface module.

Table 19. PNP Interface Module Registers Addresses

Register	Host Address[1]	380C I/O Address
Address Port	0279h	—
Write Data Port	0A79h	—
Read Data Port	Note[2]	—
Read Data Address Reg.	@00	—
Isolation Register	@01	—
Configuration Control Reg.	@02	—
Wake Register	@03	—
Resource Data Register	@04	0104h
Status Register	@05	0105h
Card Select Number Register	@06	0106h
Logical Device Number Reg.	@07	—
Activate Register	@30	0130h
I/O Range Check Register	@31	—
I/O Base Address 0 High Register I/O Mailbox)	@60	0160h
I/O Base Address 0 Low Register I/O Mailbox)	@61	0161h
I/O Base Address 1 High C Register (MIMIC)	@62	0162h
I/O Base Address 1 Low Register (MIMIC)	@63	0163h
Interrupt Request Level 0 Register	@70	0170h
DMA Channel 0 Register	@74	0174h

Table 19. PNP Interface Module Registers Addresses

Register	Host Address[1]	380C I/O Address
DMA Channel 1 Register	@75	0175h
PnP Master Register	—	0102h

NOTES:
1. Addresses preceded by an 'CST' symbol indicate the hexadecimal offset written into the Address Port by the Host to access the register.
2. This address is programmed by the host in the range 0203h -03FFh via register 000.

PnP Register Descriptions

The following sections describe the registers associated with the PnP Interface module. They are ordered as listed in the table above. In the descriptions

- The values shown in the line labeled 'DEF' are the default values after a reset
- X indicates that the reset value is indeterminate
- HHHHh is the register's address in the 380C I/O space or in the host address space, as appropriate
- @HH is the address value written into the Address Port by the host to access an internal register

PnP Address Port

Address Port **Host Address: 0279h**

BIT	7	6	5	4	3	2	1	0
R	No Function							
W	Pnp Interface Register Address							
DEF	X	X	X	X	X	X	X	X

Bit(s)	Function	R/W[1]	Description
7:0	PnP Interface Register Address	H:W	A PnP register (see “PnP Interface Registers” on page 192) is accessed by first writing its address to this port, followed by reading data from the Read Data Port or writing of data to the Write Data Port.

NOTES:

[1] H indicates Host-side capability, P indicates processor-side (380C) capability.

PnP Write Data Port (PNPWDP)

Write Data Port **Host Address: 0A79h**

BIT	7	6	5	4	3	2	1	0
R	No Function							
W	Write Data For Addressed Pnp Register							
DEF	X	X	X	X	X	X	X	X

Bit(s)	Function	R/W[1]	Description
7:0	PnP Register Write Data	H:W	This port is used to write data to the PnP registers (see "PnP Interface Registers" on page 192). The destination of the data is the register whose address was last written into the Address Port.

NOTES:

[1] H indicates Host-side capability, P indicates processor-side (380C) capability.

PnP Read Data Port (PNPRDP)

Read Data Port **Host Address: Relocatable 0203h - 03FFh**

BIT	7	6	5	4	3	2	1	0
R	Read Data From Addressed Pnp Register							
W	No Function							
DEF	X	X	X	X	X	X	X	X

Bit(s)	Function	R/W[1]	Description
7:0	PnP Register Read Data	H:R	This port is used to read data from the PnP registers (see “PnP Interface Registers” on page 192). The source of the data is the register whose address was last written into the Address Port. The Host address of this port is relocatable in the range `0203-03FFh` The host configures this address during the card configuration process by writing a value into PnP register 00. Bits [7:0] in this register become bits [9:2] of the Read Data Port Host address. To address this port, address bits [1:0] must be ones and address bits [11:10] must be zeroes.

NOTES:

[1] H indicates Host-side capability, P indicates processor-side (380C) capability.

PnP Read Data Address Register (PNPRDA)

Read Data Address Register **Host Address @00**

BIT	7	6	5	4	3	2	1	0
R	No Function							
W	Read Data Port Address Bits [9:2]							
DEF	X	X	X	X	X	X	X	X

Bit(s)	Function	R/W[1]	Description
7:0	Read Data Port Address Bits [9:2]	H:W	Host software can write this location, after sending the Initiation key and writing a 0 to the Wake register, to program the address of the Read Data port. Writing this location affects all of the previously unconfigured (CSN = 0) PnP interfaces in the system. Host software performs this action to avoid fixed (non-PnP) I/O cards in the system. The eight bits in this register are matched against HA[9:2]; the Read Data port is always decoded with HA[1:0] as `11` (High) and HA[11:10] as `00` (Low).

NOTES:

1. H indicates Host-side capability, P indicates processor-side (380C) capability.

PnP Isolation Register (PNPIR)

Isolation Register Host Address: @01

BIT	7	6	5	4	3	2	1	0
R	Serial Isolation Data							
W	No Function							
DEF	X	X	X	X	X	X	X	X

Bit(s)	Function	R/W[1]	Description
7:0	Serial Isolation Data [7:0]	H:R	Host software reads this location twice for each bit of the 64-bit ID value and 8-bit checksum, in each iteration of the Isolation protocol. Cards in Isolation state, having a 1 in the current bit, respond to such read accesses by driving 55H and then 0AAH on the data bus, while cards having a 0 do not. If a card in Isolation state, with a 0 as its current bit, sees another card respond with 55 and AA, it returns to Sleep state, dropping out of the current iteration of the protocol. Values other than 55, AA, and ether FF or 00 depending on the platform, indicate that the Read Data address conflicts with a fixed (non-PNP) I/O card, and needs to be reprogrammed.

NOTES:

1. H indicates Host-side capability, P indicates processor-side (380C) capability.

PnP Configuration Control Register (PNPCC)

Configuration Control Register **Host Address: @02**

BIT	7	6	5	4	3	2	1	0
R	Reserved					No Function		
W						Reset Logical Devices	Wait for Key	Reset CSN
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W[1]	Description
7:3	Reserved	H:W	Reserved. Program as zeroes.
2[2]	Reset CSN Command	H:W	Reset Card Select Number to zero.
1[2]	Wait for Key Command	H:W	Return to Wait for Key state.
0[2]	Reset Logical Devices	H:W	Reset logical devices (MIMIC and I/O Mailbox in Z80382) and initialize Configuration Command registers (PnP RAM in Z80382).

NOTES:

1. H indicates Host-side capability, P indicates processor-side (380C) capability.
2. Bits [2:0] are command bits. There is no need for software to clear these bits.

PnP Wake Register (PNPWR)

Wake Register **Host Address: @03**

BIT	7	6	5	4	3	2	1	0
R	No Function							
W	Wake CSN							
DEF	X	X	X	X	X	X	X	X

Bit(s)	Function	R/W[1]	Description
7:0	Wake CSN[2]	H:W	When Host software writes to this write-only location, it selects any and all PnP cards whose CSN register matches the value written, for subsequent write or read operations on other registers. Writing a 00 to this location notifies all cards whose CSN register contain 00 (as after a Reset), if any, that an iteration of the Isolation protocol is beginning, and that they should participate in it. Writing a non-zero value selects the card having a matching CSN for subsequent reading and writing, and deselects other cards. From the Host's view, writing this location also resets the pointer used in reading Resource Data.

NOTES:

1. H indicates Host-side capability, P indicates processor-side (380C) capability.
2. There actually is no Wake register in the Z80382 PnP interface: the value on HD7-0 is compared to the value in the CSN register, and to zero, to decide how to change the State.

Resource Data Register (PNPRD)

Resource Data Register **Host Address: @04**
380C Address: 0104h

BIT	7	6	5	4	3	2	1	0
R W	PNP Resource Data							
DEF	X	X	X	X	X	X	X	X

Bit(s)	Function	R/W[1]	Description
7:0	PNP Resource Data	H:R P:W	After host software has completed the isolation protocol for a card and assigned it a CSN, it can read a description of the card from this read-only location, by successively reading it after each time it detects bit 0 of the Status register set to 1.

NOTES:

1. H indicates Host-side capability, P indicates processor-side (380C) capability.

PnP Status Register (PNPSR)

Status Register **Host Address: @05**
380C Address: 0105h

BIT	7	6	5	4	3	2	1	0
R	Reserved							RSRC Data Ready
W								Reserved
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W[1]	Description
7:1	Reserved	H:R P:R	Read as zeroes.
0	PNP Resource Data Ready	H:R P:R	A 1 indicates that the data in the Resource Data register is valid and can be read by the Host. This bit is cleared by Reset, when host software writes to the Wake register, or reads from the Resource Data register; it is set when the 380C processor writes to the Resource Data register.

NOTES:

[1] H indicates Host-side capability, P indicates processor-side (380C) capability.

PnP Card Select Number Register (PNPCSN)

Card Select Number Register **Host Address: @06**
380C Address: 0106h

BIT	7	6	5	4	3	2	1	0
R	Card Select Number							
W								
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W[1]	Description
7:0	Card Select Number	H:R/W P:R	Host software should write a non-zero value to this read/write register when it has successfully completed an iteration of the Isolation protocol, that is, one that indicates that a single board has been selected without conflict on the Read Address. This register is cleared to 00 by a reset, and if host software writes a 1 to bit 2 of the Configuration Control register.

NOTES:

1. H indicates Host-side capability, P indicates processor-side (380C) capability.

PnP Logical Device Number Register (PNPLDN)

Logical Device Number Register **Host Address: @07**

BIT	7	6	5	4	3	2	1	0
R	Logical Device Number							
W								
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W[1]	Description
7:0	Logical Device Number	H:R	Writes to this location are ignored by the Z80382 PnP interface; the interface always provides 00 on a read from this location. This value identifies the Z80382 as a single logical device.

NOTES:

[1] H indicates Host-side capability, P indicates processor-side (380C) capability.

PnP Activate Register (PNPACT)

Activate Register **Host Address: @30**
380C Address: 0130h

BIT	7	6	5	4	3	2	1	0
R W	Reserved							Activate
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W[1]	Description
7:1	Reserved		Reserved, program as zeroes.
0	Activate	H:R/W P:R/W	This bit, or bit 1 of the I/O Range Check register, @31, must be 1 to qualify the address decode of the MIMIC interface and I/O Mailbox. Bit 0 resets to 0, and is forced to 0 if software writes a 1 to bit 0 of the Configuration Control register. Bit 0 is Read Only to the 382 processor in Configuration State.

NOTES:

[1] H indicates Host-side capability, P indicates processor-side (380C) capability.

PnP I/O Range Check Register (PNPRC)

I/O Range Check Register **Host Address: @31**

BIT	7	6	5	4	3	2	1	0
R W	Reserved						IORC Enable	IORC Data
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W[1]	Description
7:2	Reserved		Reserved. Program as zeroes.
1	I/O Range Check Enable	H:R/W	If Activate (@30[0]) is a zero, a 1 in this bit enables the I/O range check function, a 0 disables it. If Activate is a one, the function is disabled regardless of the state of this bit.[2]
0	I/O Range Check Data	H:R/W	When the I/O range check function is enabled, Host reads from the MIMIC and I/O Mailbox return 55h if this bit is a 1, or 0AAh if this bit is a 0.[3]

NOTES:

1. H indicates Host-side capability, P indicates processor-side (380C) capability.
2. Bit 1 is forced to 0 if software writes a 1 to bit 0 of the Configuration Control register.
3. Bit 0 is forced to 0 if software writes a 1 to bit 0 of the Configuration Control register.

PnP I/O Base Address 0 High Register - I/O Mailbox (IOMBXAH)

I/O Base Address 0 High Register **Host Address: @60**
380C Address: 0160h

BIT	7	6	5	4	3	2	1	0
R W	Reserved				I/O Mailbox Base [11:8]			
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W[1,2]	Description
7:4	Reserved		Reserved. Write as zeroes.
3:0	I/O Mailbox Base Address [11:8]	H:R/W P:R/W	Bits 3-0 of this location are matched against HA11-8 for the I/O Mailbox address decode.

NOTES:

1. H indicates Host-side capability, P indicates processor-side (380C) capability.
2. This register is read-only to the 380C in Configuration state.

PnP I/O Base Address 0 Low Register - I/O Mailbox (IOMBXAL)

I/O Base Address 0 Low Register

Host Address: @61
380C Address: 0161h

BIT	7	6	5	4	3	2	1	0
R W	I/O Mailbox Base [7:2]							Reserved
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W[1,2]	Description
7:2	I/O Mailbox Base Address [7:2]	H:R/W P:R/W	Bits[7:2] of this location are matched against HA[7:2] for the I/O Mailbox address decode.[3]
1:0	Reserved		Reserved. Program as zeroes. Since these bits are not included in the decode for the I/O Mailbox chip select, the decode occupies four contiguous bytes starting at the specified base address.

NOTES:

1. H indicates Host-side capability, P indicates processor-side (380C) capability.
2. This register is read-only to the 380C in Configuration state.
3. This PnP interface does 12-bit address decoding. This can be extended to 16 bits by externally decoding HAI 5-12 and AEN all low, and connecting the low-active decode to the $\overline{\text{HAEN}}$ pin.

PnP I/O Base Address 1 High Register - MIMIC (MIMICAH)

I/O Base Address 1 High Register **Host Address: @62**
380C Address: 0162h

BIT	7	6	5	4	3	2	1	0
R W	Reserved				MIMIC Base [11:8]			
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W[1,2]	Description
7:4	Reserved		Reserved. Write as zeroes.
3:0	MIMIC Base Address [11:8]	H:R/W P:R/W	Bits 3-0 of this location are matched against HA11-8 for the MIMIC address decode.

NOTES:

1. H indicates Host-side capability, P indicates processor-side (380C) capability.
2. This register is read-only to the 380C in Configuration state.

I/O Base Address 0 Low Register - MIMIC (MIMICAL)

I/O Base Address 0 Low Register

Host Address: @63
380C Address: 0163h

BIT	7	6	5	4	3	2	1	0
R W	MIMIC Base [7:3]					Reserved		
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W[1,2]	Description
7:3	MIMIC Base Address [7:2]	H:R/W P:R/W	Bits[7:2] of this location are matched against HA[7:3] for the MIMIC address decode.[3]
2:0	Reserved		Reserved. Program as zeroes. Since these bits are not included in the decode for the I/O Mailbox chip select, the decode occupies eight contiguous bytes starting at the specified base address.[4]

NOTES:

1. H indicates Host-side capability, P indicates processor-side (380C) capability.
2. This register is read-only to the 380C in Configuration state.
3. This PnP interface does 12-bit address decoding. This can be extended to 16 bits by externally decoding HA15-12 and AEN all low, and connecting the low-active decode to the $\overline{\text{HAEN}}$ pin.
4. The I/O Mailbox address is “I/O Range 0” and the MIMIC is “I/O range 1” because the MIMIC may want to be described in Resource Data as a “dependent function” with the traditional COM port addresses preferred.

PnP Interrupt Request Level 0 Register (PNPIRQ)

Interrupt Request Level 0 Register **Host Address: @70**
380C Address: 0170h

BIT	7	6	5	4	3	2	1	0
R W	Host: '0000’ 380C: IRQ Level Shadow				IRQ Level			
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W[1,2]	Description
7:4	IRQ Shadow	H:R/ P:R/W	These bits should be written with the same data as [3:0] by 380C firmware at Reset or Software Reset time. The contents should be the value of the ISA interrupt request level to which HINT1 or HINT2 is connected. After the host configures this register, these bits are compared by the hardware to bits [3:0] to determine whether to drive HINT1 (if equal), or HINT2 (if nonzero and not equal), when the MIMIC asserts its MINT output. When the host configures this register, the hardware prevents bits [7:4] from being overwritten. These bits are read as 0000 when this register is read by the host.[3]
3:0	IRQ Level	H:R/W P:R/W	The Host can write bits 3-0 of this location to select the value of the IRA level between the two that the Resource Data from the 380C2 offered it, or zero if neither level could be assigned to this card.

NOTES:

1. H indicates Host-side capability, P indicates processor-side (380C) capability.
2. This register is read-only to the 380C in Configuration state.
3. If used in a non-PnP environment, [7:4] matches [3:0], and MINT from the MIMIC is routed to the HINT1 output.

PnP DMA Channel 0 Register (PNPDMA0)

DMA Channel 0 Register **Host Address: @74**
380C Address: 0174h

BIT	7	6	5	4	3	2	1	0
R W	Reserved					DMA Channel 0		
DEF	0	0	0	0	0	1	0	0

Bit(s)	Function	R/W[1,2]	Description
7:3	Reserved		Reserved. Program as zeroes.
2:0	DMA Channel 0 (I/O DMA Mailbox 0)	H:R/W P:R/W	This field is programmed by the Host with the DMA Channel number that is offered to it in the Resource Data. This number is the number of the channel to which the HDRQ0 and $\overline{\text{HDAK0}}$ pins are connected. The Host programs a value of `100` if the channel could not be assigned, or if the Resource Data indicates that use of this DMA channel is not required. In operation, the hardware routes the internal DMA request and acknowledge signals from/to I/O DMA Mailbox 0 to the HDRQ0 and $\overline{\text{HDAK0}}$ pins if the value in this field is not `100`, and leaves HDRQ0 3-stated if the value in this field is `100`.

NOTES:

1. H indicates Host-side capability, P indicates processor-side (380C) capability.
2. This register is read-only to the 380C in Configuration state.

PnP DMA Channel 1 Register (PNPDMA1)

DMA Channel 1 Register **Host Address: @75**
380C Address: 0175h

BIT	7	6	5	4	3	2	1	0
R W	Reserved					DMA Channel 1		
DEF	0	0	0	0	0	1	0	0

Bit(s)	Function	R/W[1,2]	Description
7:3	Reserved		Reserved. Program as zeroes.
2:0	DMA Channel 0 (I/O DMA Mailbox 1)	H:R/W P:R/W	This field is programmed by the Host with the DMA Channel number that is offered to it in the Resource Data. This number is the number of the channel to which the $\overline{\text{HDRQ1}}$ and $\overline{\text{HDAK1}}$ pins are connected. The Host programs a value of `100` if the channel could not be assigned, or if the Resource Data indicates that use of this DMA channel is not required. In operation, the hardware routes the internal DMA request and acknowledge signals from/to I/O DMA Mailbox 1 to the $\overline{\text{HDRQ1}}$ and $\overline{\text{HDAK1}}$ pins if the value in this field is not `100`, and leaves HDR01 3-stated if the value in this field is `100`.

NOTES:
1. H indicates Host-side capability, P indicates processor-side (380C) capability.
2. This register is read-only to the 380C in Configuration state.

PNP Master Register (PNPMR)

PnP Master Register **380c Address: 0102h**

BIT	7	6	5	4	3	2	1	0
R W	Interrupt Enable	Isolation Bit	Software Reset	Wake Match	Isolation Read	Interrupt Pending	PNP State	
DEF	0	0	1	0	0	0	0	0

Bit(s)	Function	R/W[1]	Description
7	PNP Interrupt Enable	P:R/W	If this bit is 1, the PnP interface requests a 380C processor interrupt via the Assigned Vector facility if Interrupt Pending, bit 1, is a `1`.
6	Isolation Write	P:W	Software must write the current bit from this card's 72-bit ID/checksum value to this bit during the Isolation process, in response to either the Wake Match or Isolation Read condition with the State field `10`.
5	Software Reset	P:R/W	This bit is set to `1` by Reset or if host software writes a `1` to bit 0 of the Configuration Control register. It is cleared when 380C software writes a `1` to this bit.
4	Wake Match	P:R/W	This bit is set to 1 when host software writes a value to the Wake register that matches the value in the CSN register. it is cleared by Reset, and when 380C software writes a `1` to this bit.
3	Isolation Read	P:R/W	This bit is set to `1` when the state is `10`, at the end of the second read of the Isolation register for the current bit. It is cleared to `0` by reset, and when 380C software writes a `1` to this bit.

Bit(s)	Function	R/W[1]	Description
2	Interrupt Pending	P:R/W	This bit is set if Host software: • Writes to the Configuration Control register, or • Writes a value to the Wake register, that is equal to the value in the CSN register, or • Writes a value to the Wake register that doesn't match the contents of the CSN, if the state had been 1x, or • Reads the Isolation register the second time for a bit, and the State is `10`, or • Writes the CSN register (with the State `1x`), or • Reads the Resource Data register (with the State `11`). The interface requests an interrupt whenever the IP and IE bits are both 1. IP is cleared by reset, and when software writes a 1 to this bit.

Bit(s)	Function	R/W[1]	Description	
1:0	PNP State	P:R		This read-only field defines the current state of the PnP interface.
			Value	State
			00	Walt for Key
			01	Sleep
			10	Isolation
			11	Configuration
				The state is affected by reset and by operations by the Host as follows:
			Any to 00	Assertion of the Reset pin, or host software writes a `1` to bit 1 of the Configuration Control register.
			00 to 01	Detection of an Initiation key.
			01 to 10	When the Host writes `0` to the Wake register, and the CSN register is zero.
			10 to 01	When the Isolation Bit is `0`, and Host software reads the Isolation register twice for a bit, and pins HD7-0 are `01` on the first read, and `10` on the second, indicating another card has a `1` in the current bit.
			10 to 11	When the Host writes the CSN register of the winner of an Isolation process.
			01 to 11	When the Host writes a non-zero value to the Wake register that matches the value in the CSN register.
			1x to 01	When the Host writes a value to the Wake register that does not match the CSN register.

NOTES:

[1] H indicates Host-side capability, P indicates processor-side (380C) capability.

DMA Channels

OVERVIEW

The DMA channels of the Z80382x build on ZiLOG's experience with the 216032 IUSC. Their operation combines features of the IUSC's Array and Linked List modes.

DMA operation can be optionally selected for the Mimic, the ASCIs and the Host DMA Mailbox facility. DMA operation is required to use any of the HDLC channels. Because these peripherals all operate with 8-bit data only, particularly the HDLC channels which the DMA channels are primarily intended to serve, the DMA channels also perform only 8-bit data transfers when operating with data buffers. However, because fetching a new list entry is an overhead operation that can compromise maximum data rates, list accesses use 16-bit transfers. 24 bits of address are output on the A[23:0] lines during each DMA cycle.

DMA Channel/Device Interface

The interface between the DMA channel and its client device includes six lines, as describe in Table 20:

Table 20. DMA Channel/Client Device Lines

Data Request	Device to DMA
Terminate	Device to DMA
Type Fetch	DMA to device
Data Acknowledge	DMA to device
End of Buffer	DMA to device
Store Status	DMA to device

All of these lines are bussed, and are driven by the DMA channel and client device that are currently selected by the DMA scanner.

DMA LISTS AND ASSOCIATED REGISTERS

The DMA channels operate in conjunction with DMA List structures stored in external memory. Each DMA channel contains a pointer into such a list structure, entries in which contain the addresses and lengths of data buffers. List entries always begin at an 8-byte boundary, that is, at an address having its least significant three bits `000`. List structures are placed sequentially in memory, that is, the beginning of the next list structure is normally at an address eight bytes higher than the previous one. A `Transfer in List` entry (see “DMA Channel Operation” on page 222) may be used to link disjoint parts of a list.

The general format of a list entry includes eight bytes, as depicted in Figure 15:

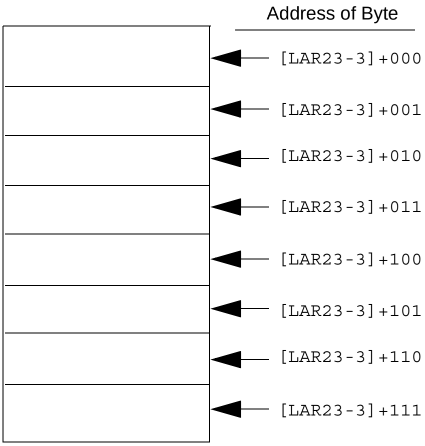


Figure 15. General Format of a DMA List Entry

A DMA list entry contains three defined fields:

- Type/Status–this field defines the type of entry
- Address–this field is the address of an associated data buffer or the address of the next list entry
- Length–this field provides the length of the associated data buffer

Three registers in each DMA channel store data concerning the current operation of the channel:

List Address Register (LAR)

This register is a three-byte register whose 21 most significant (MS) bits contain the base address of the current list entry. A DMA channel begins operation when the host writes the most significant byte of this register. The DMA controller updates this register as it processes new list entries. The three least significant (LS) bits of the LAR are ignored on writing, and always read back as `100` (pointing at the current Status byte/Type in the list).

Buffer Address Register (BAR)

The DMA controller loads the initial value of the current buffer address into the 3-byte BAR from the address field of the current DMA list. At the end of each DMA data transfer, the DMA channel increments the BAR by one.

Buffer Length Register (BLR)

The DMA controller loads the initial buffer length into this register from the buffer length field of the current DMA list entry. At the end of each data transfer, the DMA channel decrements the BLR by one. When it finishes transferring data to or from a buffer, a DMA channel stores the ending BLR value back in the buffer length field of the buffer's list entry.

DMA CHANNEL OPERATION

A DMA channel operates when software loads an address into its List Address Register (LAR). Writing the MS byte of this register sets the channel's Run bit, which prompts a bus access request from the 380C processor. When the processor grants bus access, the DMA channel proceeds to fetch the first list entry from memory, beginning at the address in the LAR.

Table 21 identifies various kinds of list entries.

Table 21. Status Byte/Type List Entries

Status Byte	Type
00	End of List
01	Transfer in List
02	Ready Buffer, no Command, no End of Buffer notification
03	Ready Buffer, no Command, notify device at End of Buffer
04	Buffer in Progress
05	Completed Buffer (no Status)
40–7F	Ready Buffer, with Command, no End of Buffer notification
80–BF	Ready Buffer, with Command, notify device at End of Buffer
C0–FF	Completed Buffer (with Status)

On fetching any Type/Status value except Transfer in List or Ready Buffer, the DMA channel clears its Run bit and requests an interrupt if its List Interrupt Enable bit is 1. Checking of the Status byte/Type helps prevent disorderly operation as well as buffer-ring wraparound.

On fetching a Transfer in List entry, the DMA channel fetches the Address portion of the entry, loads it into its LAR, and fetches another list entry from that address. Buffer rings and linked lists are constructed using this mechanism.

If software requires knowledge of when a certain amount of data has been sent or received, such as an Address field in a received HDLC frame, it sets up a buffer of that length with its own list entry. The DMA channel provides an interrupt at the end of the buffer.

When a DMA channel fetches a Status byte/Type from memory, it asserts the Type Fetch signal to its client device. The client device is prompted to capture the Command if bits D[7:6] of the Status byte/Type are `01` or `10`.

For example, an HDLC transmitter uses the three least significant bits of such a Status byte/Type to indicate how many bits to send from the last byte of the frame. The HDLC receiver uses no Command bits, so that Ready Buffer codes, with and without Command, are equivalent for HDLC reception.

Upon fetching any Ready Buffer entry, the DMA channel fetches the Address and Buffer Length fields, loads them into its Buffer Address and Length Registers respectively, and then rewrites the Status byte/Type to the `Buffer in Progress` code. The DMA channel transfers data into or out of the buffer, under control of the Data Request line from its client device. If there is no request at this point, as is typically the case when software starts a Receive channel, the DMA channel relinquishes bus control and goes Idle until the device asserts Data Request and/or Terminate. For Status byte/Types indicating `Notify device at end of buffer`, the DMA channel asserts its client's End of Buffer line as the last byte of the buffer is transferred.

After a DMA channel has been started and fetched its first list entry, it is inactive unless its client device asserts Data Request and/or Terminate. When the client devices does so, the DMA channel requests bus access. When access is granted, or when it is continuing operation after fetching a list entry, the DMA channel proceeds as detailed in the following sections.

If the device is asserting Data Request, with or without Terminate:

1. The DMA channel asserts Data Acknowledge to the device.
2. If its BLR indicates the buffer is ending, and the Status byte/Type for this buffer said Notify Device, the DMA channel also asserts the End of Buffer signal.
3. At the same time, the DMA channel places the address in its BAR on the address bus, and sets the control signals for a memory read or write per the I/O bit in its DMA Control/Status Register (DCSR).
4. Depending on the data direction, Data Acknowledge makes the device either provide data on the data bus, or capture data from the data bus.

How (and whether) a client device uses End of Buffer is device-dependent. The HDLG transmitter passes this indication through its Tx FIFO, and terminates the Tx frame after sending the data with which the DMA channel asserted End of Buffer. Because of this facility, the only time that an underrun can occur at the HDLG transmitter is when the DMA does not provide data fast enough, INSIDE a frame.

The HDLC receiver does nothing with End of Buffer, so `Ready Buffer` codes with or without EOB are equivalent for HDLG receiving.

At the end of each data transfer, the DMA channel increments the BAR by one and decrements the BLR by one.

If the device signalled Data Request but not Terminate, and the Buffer Length Register has not been counted down to zero, and the Burst bit in the channel's DCSR is set, the DMA channel checks Data Request again. If Burst is `0`, and/or if the device negates Data Request, the channel returns the bus to the processor or another DMA channel. Otherwise, the channel performs another data transfer.

If the device signalled Data Request, but not Terminate, and the Buffer Length Register has been counted down to zero, the DMA channel proceeds as follows:

1. It puts the address of the Status byte/Type (from the LAR) on the address bus, and writes the code for Completed Buffer (no Status) into that byte.
2. If the DMA channel's Buffer IE field indicates Interrupt For All Buffers, or Interrupt for Notify Buffers and a Notify Buffer is selected, it sets its IF bit to request an interrupt.
3. It increments the LAR to the address following this list entry, and goes back to fetch a new list entry from that address, as described previously.

Terminate Signal

The HDLC receiver asserts this signal for an End of Frame, Abort, or Overrun condition (in TRANSPARENT mode it asserts this signal only for an Overrun condition). The HDLC transmitter asserts this signal for an Underrun condition. After the DMA channel transfers a byte if the device signals Data Request and Terminate, or if the device signals Terminate without Data Request, the DMA channel proceeds as follows:

1. It places the address of the Length field on the address bus, and writes the current (16-bit) value in its BLR to memory at that address and the next higher address. This value enables software to ascertain how much data was actually written into, or read out of, this buffer.
2. It puts the address of the Status byte/Type on the address bus, sets the control signals for a memory write, drives D[7:6] both high (1) to indicate Completed Buffer (with Status), and asserts the Store Status signal to the device.
3. In response to Store Status, the device can place up to 6 bits of status on D[5:0]. For the HDLC receiver, this status includes Overrun, End

of Frame, Abort, CRC Error, and the residual bit count. For the HDLC Transmitter, only Underrun prompts a Terminate indication, so status bits [5:0] are unimportant.

1. After the Status byte/Type has been written, the DMA channel advances the LAR over this list entry, that is, to the address of the next entry.
2. If the DMA channel's Buffer IE field indicates anything other than No Buffer Interrupts, it requests an interrupt.
3. The DMA channel then goes back to fetch another list entry from the address in the LAR, as described above.

Note: If the device encounters an error from which operation cannot continue without processor attention, then after signalling Terminate and storing a status byte as described above, the device refrains from asserting Data Request until software has responded. The HDLC transmitter performs this for Underrun.

Adding a Buffer at the End of a List

Software can perform this action while a DMA channel is running, without using the Clear and Set Run facilities in the DMA Control/Status register (DCSR). If there is room for another list entry after the current end-of-list entry, software places the End of List code in the Status byte/Type of the following entry, places the address and length of the new buffer in the current end-of-list entry, and finally substitutes the Ready Buffer code for the End of List code in what was, until this operation, the end-of-list entry.

If there is no room for another entry after the current End of List entry, software arranges a Ready Buffer entry followed by an End of List entry elsewhere in memory, then places the address of the new Ready Buffer entry in the Address field of the current End of List entry, and finally substitutes the Transfer in List code for the End of List code in what was, until this operation, the End of List entry.

After adding a buffer to a list by either method, software reads the DCSR and checks the Run bit, to determine if the channel requires a restart. It is not necessary to check Run before adding the entry, because the channel could reach the end of the list between such a check and when software overwrites the End of List code.)

DMA Interrupts Overview

There is an IP bit for each DMA Channel, but only one IUS bit for all of them. If the interrupt routine for a DMA Channel enables nested interrupts from other devices, it re-enables interrupts near its start and then clears the IP bit. Setting the IUS bit prevents another DMA interrupt. An ISR that disallows nested interrupts clears IP and IUS near its start, in the same register write.

After clearing IP, the ISR reads status and its channel's List Address Register, and scans the DMA list to determine what has occurred. At this point the ISR can, optionally, read the DMA Vector register to determine if any other DMA Channel has an interrupt pending.

Finally, if the ISR allowed nested interrupts, it clears IUS to allow future DMA interrupts. Disabling interrupts again before clearing IUS, prevents stack filling (tail recursion) if interrupt traffic is heavy.

DMA REGISTERS

Centralized Registers

Two registers provide overall control and status of the DMA subsystem:

- DMA Control Register (DMACR)
- DMA Vector Register (DMAVR)

Per-Channel Registers

There are 8 DMA channels in the Z80382. Each channel includes the following registers:

- List Address Register (LAR, 21 bits)
- Buffer Address Register (BAR, 24 bits)
- Buffer Length Register (BLR, 16 bits)
- DMA Control/Status Register (DCSR, 8 bits)

The LAR and DCSR are read/write registers; software tracks the progress of a DMA by monitoring its LAR. BARs and BLRs are accessible only by using special modes selected in the centralized DMA Control Register. The channel stores ending BLR values in the list.

Register Descriptions

Details on each of the above registers are provided in the sections which follow. In the descriptions:

- `HHHHh` = register's address in the 380C I/O space, in hexadecimal
- DEF = default value after RESET
- X = indicates that the default value of the bit is indeterminate

DMA Control Register (DMACR)

DMA Control Register (DMACR) 003Eh

BIT	7	6	5	4	3	2	1	0
R W	Scanend		Reserved				Register Select	
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W		Description
7:4	ScanEnd	R/W		This field controls when bus control is returned to the 380C processor after a DMA channel has operated. For all choices except 00, the DMA facility scans other DMA channels as to whether they are ready to operate. Higher binary values in this field favor DMA operation over host processor operation.
			00	Control is returned to the Host processor after each DMA channel operates.
			01	Control is relinquished as soon as a DMA channel that is not ready to operate (including a channel having is Run bit 0) has been found, or until all the channels have been scanned once if they were all ready.
			10	Control is relinquished after all the DMA channels have been scanned once, and serviced if they are ready to operate.
			11	Control relinquished only after all DMA channels have been scanned as having nothing to do, in succession.

Bit(s)	Function	R/W	Description
5:2	Reserved		Reserved
1:0	Register Select	R/W	This field provides modes whereby the Buffer Address and Buffer Length registers can be read. 00 Addresses 0040h - 0042h, 0044h - 0046h,... , 005Ch - 005Eh access List Address Registers 01 Addresses 0040h - 0042h, 0044h - 0046h, , 005Ch - 005Eh access Buffer Address Registers 10 Addresses 0040h - 0041h, 0044h - 0045h, ... , 005Ch - 005Dh access Buffer Length Registers 11 As 00, except that writing LAR[23:16] doesn't start the DMA channel

DMA Vector Register (DMAVR)

DMA Vector Register (DMAVR) **003Fh**

BIT	7	6	5	4	3	2	1	0
R	Vector Base [7:4]				Device			No IP
W					No Function			
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W	Description
7:4	Vector Base 7:4	R/W	This field must be written with the upper nibble of the base interrupt vector for the DMA channels.
3:1	Device	R	When this register is read with a register read cycle, if the No IP bit is 0, this field identifies the highest priority DMA Channel that has its IP bit set. The DMA channel with the highest priority is DMA0 (000), the lowest priority is DMA7 (111).
0	No IP	R	When this register is read with a register read cycle, it is 0 if any of the DMA channels has is IP bit set, or 1 if none of them have IP set.
7:0	Interrupt Vector[1]	—	This vector is automatically output on the D[7:0] lines in response to an interrupt acknowledge cycle if the highest priority interrupting device is a DMA channel. The contents of the vector are: — The upper nibble of the base interrupt vector on D[7:4]. — A field identifying the highest priority DMA channel that has its IP bit set on D(3:1] (see “Device” below). — A zero on D[0].

NOTES:

1. The DMA channels have no provision for 380C Mode 0 interrupts, nor for “status does not affect vector.”

DMA List Address Registers (DMALAR)

List Address Register High (LARnH, n = 0,1, ... 7) 0042h, 0046h, ... , 005Eh

BIT	7	6	5	4	3	2	1	0
R	List Address [23:16] Note: DMACR[1:0] must be 00 or 11 to access this register							
W								
DEF	0	0	0	0	0	0	0	0

List Address Register Middle (LARnM, n = 0,1, ... , 7) 0041h, 0045h, ... ,005Dh

BIT	7	6	5	4	3	2	1	0
R	List Address [15:8] Note: DMACR[1:0] must be 00 or 11 to access this register							
W								
DEF	0	0	0	0	0	0	0	0

List Address Register Low (LARnL, n = 0,1, ... , 7) 0040h, 0044h, ... ,005Ch

BIT	7	6	5	4	3	2	1	0
R	List Address 7:3] Note: DMACR[1:0] must be 00 or 11 to access this register					100		
W						XX		
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W	Description
LAH[7:0] LAM[7:0] LAL[7:0]	List Address [23:0][1]	R/W	The List Address Register is a three-byte register whose 21 most significant bits contain the address of the current DMA list entry. The DMA channel begins operation when the host writes the most significant byte of this register if the current content of DMACR[1:0] is 00. The DMA controller updates this register as it processes new list entries. The three LS bits of the LAR are ignored on writing, and always read back as 100, (pointing at the current Status byte/Type in the list).

NOTES:

1. These registers are accessible by the 380C only when the value of DMACR[1:0] is 00 or 11.

DMA Buffer Address Registers (DMABAR)

Buffer Address Register High (BARnH, n = 0,1, ... , 7) **0042h, 0046h, ... , 005Eh**

BIT	7	6	5	4	3	2	1	0
R	Buffer Address [23:16] Note: DMACR[1:0] must be 01 to access this register							
W								
DEF	0	0	0	0	0	0	0	0

Buffer Address Register Middle (BARnM, n = 0,1, ... , 7) **0041h,0045h, ... , 005Dh**

BIT	7	6	5	4	3	2	1	0
R	Buffer AddresS [15:6] Note: DMACR[1:0] must be 01 to access this register							
W								
DEF	0	0	0	0	0	0	0	0

Buffer Address Register Low (BARnL, n = 0,1, ... , 7) **0040h, 0044h, ... , 005Ch**

BIT	7	6	5	4	3	2	1	0
R	Buffer Address [7:0] Note: DMACR[1:0] must be 01 to access this register							
W								
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W	Description
BAU[7:0] BAM[7:0] BAL[7:0]	Buffer Address [23:0][1]	R	The DMA controller loads the initial value of the current buffer address into this register from the address field of the current DMA list. At the end of each DMA data transfer, the DMA channel increments the BAR by one.

NOTES:

1. These registers are accessible by the 380C only when the value of DMACR[1:0] is 01.

DMA Buffer Length Registers (DMABLR)

Buffer Length Register High (BLRnH, n = 0, 1, ... , 7) 0041h, 0045h, ... , 005Dh

BIT	7	6	5	4	3	2	1	0
R	Buffer Length [15:8] Note: DMACR[1:0] must be 10 to access this register							
W								
DEF	0	0	0	0	0	0	0	0

Buffer Length Register Low (BLRnL, n = 0, 1, ... , 7) 0040h, 0044h, ... , 005Ch

BIT	7	6	5	4	3	2	1	0
R	Buffer Length [7:0] Note: DMACR[1:0] must be 10 to access this register							
W								
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W	Description
BLU[7:0] BLL[7:0]	Buffer Lengths [15:0][1]	R	The DMA controller loads the initial value of the current buffer length into this register from the buffer length field of the current DMA list. At the end of each character data transfer from the HDLC receiver to the Rx FIFO, or from the Tx FIFO to the HDLC transmitter, the DMA channel decrements the BLR by one. The DMA channel stores the ending BLR value back in the buffer length field of the current DMA list.

NOTES:

1. These registers are accessible by the 380C only when the value of DMACR[1:0] is 10.

DMA Control/Status Register (DMACSR)

DMA Control/status Register (DCSR0,1, ..., 7) 0043h, 0047h, ..., 005Fh

BIT	7	6	5	4	3	2	1	0
R	I/O Direction	Burst Mode Enable	List IE	Buffer IE		Run	IUS	IP
W						DMA Command		
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W		Description
7	I/O Direction	R/W		If this bit is 0, the DMA channel reads data from memory buffers for its client device. If this bit is 1, the DMA channel writes data from its client device into memory buffers.
6	Burst Mode Enable	R/W		A 1 in this bit causes the channel to examine Data Request from its client device immediately after each transfer. If the device does not negate Data Request immediately, the channel keeps control of the bus and transfers more data. A 0 in this bit causes the DMA channel to perform one data transfer, then wait at least seven clock cycles (more for ScanEnd modes less than 11) before examining the Data Request line from its client device again. This choice must be used with client devices that cannot meet the BURST mode timing requirements for negating Data Request. The internal HDLC transmitters and receivers can use BURST mode.
5	List IE	R/W		A 1 in this bit causes this DMA Channel to set its IP bit, requesting an interrupt, when it comes to the end of a list of buffers, or when it encounters a list entry that contains an unexpected value in its Status byte/Type.
4:3	Buffer Interrupt Enable	R/W	 00 01 10 11	This field controls when and whether this DMA Channel sets its IP bit, to request an interrupt, when it comes to the end of a DMA buffer Never Only for With Status buffers For “with Status” and/or “notify at End” buffers For all buffers

Bit(s)	Function	R/W	Description
2	Run	R	This read-only bit enables the DMA channel to operate. It is set when software writes the MS byte of this channel's LAR, and is also set if software writes a Set Run command to this register. Run is cleared by the channel in certain circumstances as described in the Operation section, and is also cleared if software writes a Clear Run command to this register (but see the description of Clear Run below).
1	Interrupt Under Service	R	There is only one Interrupt Under Service bit for all the DMA channels, but it can be read and cleared in any of their Control/Status registers. This read-only bit is set when an interrupt acknowledge cycle returns a vector for a DMA channel. It is cleared by Reset, or when software writes a Clear IUS or Clear IP and IUS command to any DMA channel's DCSR.
0	Interrupt Pending	R	This read-only bit is set when one of the events enabled by bits 5-3 occurs, or when software writes a Set IP command to this register. This bit is cleared by Reset, or when software writes a Clear IP or Clear IP and IUS command to this register. When one or more DMA channels' IP bit(s) is (are) set, and the IUS bit for the DMA channels is cleared, and the INT0 daisy-chain indicates that an Interrupt Service Routine for a higher-priority device is not in progress, the DMA subsystem requests a Z80382 interrupt on INT0.

Bit(s)	Function	R/W	Description
2:0	DMA Command	W	Software can set or clear the IP or Run bit for this channel, or clear the global IUS bit that applies to all the DMA channels, by writing one of the following codes to this write-only field:
			000 No command.
			001 Clear IP. Use this command near the start of a DMA Interrupt Service Routine (ISR) that enables “nested” interrupts from higher-priority devices. Clearing IP before reading status helps ensure that no interrupt events are lost.
			010 Clear IUS. Use this command near the end of a DMA ISR that enables *nested* interrupts from higher-priority devices.
			011 Clear IP and IUS. Use this command near the start of a DMA ISR that does not enable *nested* interrupts.
			100 Reserved.
			101 Set IP Use this command to trigger an interrupt from this DMA channel.
			110 Clear Run. Writing this command immediately prevents further operation by this DMA channel, but does not affect the Run bit in this register until after the next time software reads this register (see Notes[1,2,3]).

Bit(s)	Function	R/W		Description
2:0	DMA Command (Cont.)	W	111	Set Run. This sets the Run bit in this register and enables the DMA channel to continue operation from the state it was in when software issued a preceding Clear Run command. Loading an address into the (MS byte of the) LAR also sets the Run bit, but initializes the state of the DMA channel to fetch from the list pointed to by the address. Do not use this command to start a DMA channel after Reset.

NOTES:

1. The delay noted for the Clear Run command is provided so that software can read this register after writing a Clear Run command to it, and determine whether the DMA channel cleared Run (which requires further software attention) or whether software cleared Run by writing the Clear Run command (in which case it can restart the channel with a Set Run command).
2. Software prevents any interrupts for which the ISR could read the CCSR between when it writes the Clear Run command, and when it subsequently reads the CCSR to check Run.
3. Software can use Clear Run and Set Run to stop and then restart a channel, without any effect other than risking underrun or overrun at the device. This facility is provided in case software wants to restructure the list while the channel is operating. Note that software doesn’t need to use Clear and Set Run to add another buffer at the end of a list. “Adding a Buffer at the End of a List” on page 226 describes how to do this.

Serial Communication Channels

OVERVIEW

The Z80382 provides several facilities for serial communication with external devices. These are:

- Two Asynchronous Serial Communications Interfaces (ASCI) or UARTs. These UARTs are similar to the ASCIs incorporated in the Z1 8x series of embedded processors from ZiLOG and are capable of asynchronous communications at rates up to 512 Kbps using a 16x clock.
- Three High-level Data Link Control (HDLC) channels. Capable of full-duplex operation at serial rates up to 10 Mbps in HDLC or transparent modes, these channels operate in conjunction with the Z80382's DMA controllers to provide reception and transmission of packetized data with minimal processor intervention.
- A Clocked Serial Input/0utput (CSI/O) channel for interprocessor communications or to interface with external serial devices such as EEPROMs.

ASYNCHRONOUS SERIAL COMMUNICATIONS INTERFACE

The Z80382 provides two independently programmable ASCIs (UARTs), each including a flexible Baud Rate Generator (BRG). Key features of the ASCIs include:

- Full-duplex operation
- Programmable data format
  - 7- or 8- data bits with optional ninth bit for multiprocessor communication

  - One or two stop bits
  - Odd, even or no parity
- Programmable baud rate generator
- Divide by one, divide by 16 and divide by 64 modes
- Baud rates up to 512 Kbits/sec with 16x clock
- Up to three modem control signals per channel, depending on operating mode of the Z80382
- Programmable interrupt conditions
- Four level data/status FIFOs for the receivers
- Receive parity, framing and overrun error detection
- Optional operation with on-chip DMA controllers

Functional Description

Figure 16 illustrates the major functional blocks within the ASCI.

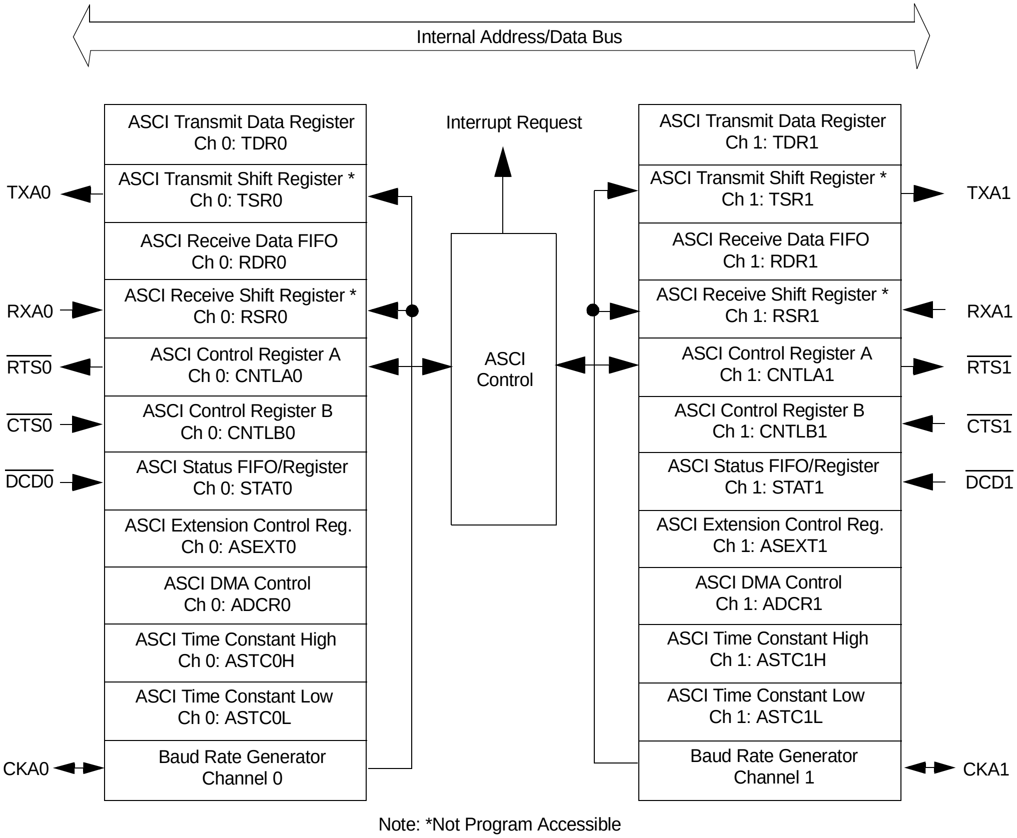

Internal Address/Data Bus
ASCI Transmit Data Register Ch 0: TDR0
ASCI Transmit Shift Register * Ch 0: TSR0
ASCI Receive Data FIFO Ch 0: RDR0
ASCI Receive Shift Register * Ch 0: RSR0
ASCI Control Register A Ch 0: CNTLA0
ASCI Control Register B Ch 0: CNTLB0
ASCI Status FIFO/Register Ch 0: STAT0
ASCI Extension Control Reg. Ch 0: ASEXT0
ASCI DMA Control Ch 0: ADCR0
ASCI Time Constant High Ch 0: ASTC0H
ASCI Time Constant Low Ch 0: ASTC0L
Baud Rate Generator Channel 0
TXA0
RXA0
RTS0
CTS0
DCD0
CKA0
Interrupt Request
ASCI Control
ASCI Transmit Data Register Ch 1: TDR1
ASCI Transmit Shift Register * Ch 1: TSR1
ASCI Receive Data FIFO Ch 1: RDR1
ASCI Receive Shift Register * Ch 1: RSR1
ASCI Control Register A Ch 1: CNTLA1
ASCI Control Register B Ch 1: CNTLB1
ASCI Status FIFO/Register Ch 1: STAT1
ASCI Extension Control Reg. Ch 1: ASEXT1
ASCI DMA Control Ch 1: ADCR1
ASCI Time Constant High Ch 1: ASTC1H
ASCI Time Constant Low Ch 1: ASTC1L
Baud Rate Generator Channel 1
TXA1
RXA1
RTS1
CTS1
DCD1
CKA1
Note: *Not Program Accessible

Figure 16. ASCI Block Diagram

Transmit Data Register

The Transmit Data Register (TDR) is a two-byte FIFO which holds characters to be serialized by the Transmit Shift Register (TSR). Data is transferred from the TDR to the TSR immediately if the TSR is inactive or as soon as the TSR has completed serializing the previous data character. Additional data to be transmitted can be written into the TDR when the TDR Empty status bit, STAT[1], indicates that space is available in the TDR FIFO. The topmost byte of the TDR is read accessible. Reading from the TDR does not affect the ASCI data transmit operation currently in progress.

Transmit Shift Register

When the ASCI Transmit Shift Register receives data from the ASCI Transmit Data Register, the data is shifted out to the TXA pin. When transmission is completed, the next byte (if available) is automatically loaded from the TDR into the TSR and the next transmission starts. If no data is available for transmission, the TXA pin maintains at a continuous High level. This register is not program-accessible.

Receive Shift Register

When the RE bit is set in the CNTLA register, the RXA pin is monitored for a low. One-half bit time after a Low is sensed at RXA, the ASCI samples RXA again. If RXA has returned to High, the ASCI ignores the previous Low and searches for a new one. If RXA is still Low, the ASCI considers this a Start bit and proceeds to clock in the data based upon the internal baud rate generator or the external clock at the CKA pin. The number of data bits, parity, multiprocessor and stop bits are selected by the MOD2, MOD1, MOD0 and MP bits in the CNTLA and CNTLB registers.

After the data is received, one or more subsequent bits (MP, parity and one stop bit) are checked for errors. Data and any errors are clocked into the receive data and status FIFOs during the stop bit if there is an empty position available. Interrupts, the Receive Data Register Full Flag, and DMA requests also go active during this time. If there is no space in the FIFO at the time that the RSR attempts to transfer the received data into it, an overrun error occurs.

Receive Data FIFO

When a complete incoming data byte is assembled in the RSR, it is automatically transferred to the FIFO, which reduces the incidence of overrun errors. The top (oldest) character in the FIFO (if any) can be read using the Receive Data Register (RDR).

The next incoming data byte can shift into the RSR while the FIFO is full, providing an additional level of buffering. However, an overrun occurs if the receive FIFO is still full when the receiver completes assembly of that character and is ready to transfer it to the FIFO. If this condition occurs, the overrun error bit associated with the previous byte in the FIFO is set. The latest data byte is not transferred from the shift register to the FIFO in this case, and is lost. When an overrun occurs, the receiver does not place any further data in the FIFO until the last good byte received has come to the top of the FIFO and sets the Overrun latch, and software then clears the Overrun latch by writing a 0 to the EFR bit (see below). Assembly of bytes continues in the shift register, but this data is ignored until the byte with the overrun error reaches the top of the FIFO and the status is cleared.

When a break occurs (defined as a framing error with the data equal to all zeros), the all-zero byte with its associated error bits is transferred to the FIFO if it is not full. If the FIFO is full, an overrun occurs, but the break, framing error and data are not transferred to the FIFO. Anytime a break is detected, the receiver receives no more data until the RXA pin returns to a High state.

If the channel is set in MULTIPROCESSOR mode and the MPE bit of the CNTLA register is 1, breaks, errors and data is ignored unless the MP bit in the received character is a one. The two conditions listed previously could cause the missing of a break condition if the FIFO is full and the break occurs or if the MP bit in the transmission is not a 1 with the above specified conditions.

Parity and Framing Errors do not affect subsequent receiver operation.

Status FIFO Register

This FIFO contains Parity Error, Framing Error, Rx Overrun, and Break status bits associated with each character in the receive data FIFO. The status of the oldest character of any) can be read from the ASCI status register, which also provides several other, non-FIFO status conditions.

The outputs of the error FIFO go to the *set* inputs of software-accessible error latches in the status register. Writing a 0 to the EFR bit in CNTLA is the only method of clearing these latches. When an error bit reaches the top of the FIFO, it sets an error latch. If the FIFO has more data and the software reads the next byte out of the FIFO, the error latch remains set and stays so until the software writes a 0 to the EFR bit. The error bits are cumulative, so if additional errors are in the FIFO they may set other error latches as they reach the top.

Baud Rate Generator

The Baud Rate Generator (BRG) features two modes. The first is the same as that used in most previous ZiLOG processors, such as the Z80180, and provides a dual set of fixed clock divide ratios. In the second mode, the BRG is configured as a sixteen-bit down counter that divides the processor clock by the value in a software accessible sixteen-bit time constant register. This condition allows virtually any frequency to be created by selecting the appropriate main processor clock frequency. The BRG can also be disabled in favor of an external clock on the CKA pin.

The Receiver and Transmitter subsequently divides the output of the Baud Rate Generator (or the signal from the CKA pin) by 1, 16 or 64 under the control of the DR bit in the CNTLB register and the X1 bit in the ASCI Extension Control Register.

Table 22 lists the formulas used to compute the channel's baud rate:

Table 22. Baud Rate Conditions

Condition	Baud Rate
If SS[2:0] = 111	f_{CKA}/Clock Mode
Else, if BRG Mode = 1	f_{PHI}(2*(TC+2)*Clock Mode)
Else, if BRG mode = 0	f_{PHI}/((10+20*PS)*DIV*Clock Mode)

Where:

- BRG mode is bit 3 of the ASEXT register
- PS is bit 5 of the CNTLB register
- f_{CKA} is the frequency of the clock at the CKA pin
- f_{PHI} is the Z80382 BUSCLK frequency
- TC is the 16-bit value in the ASCI Time Constant registers
- DIV is the divider specified by SS[2:0] in the CNTLB register
- Clock mode (see Table 23) depends on the CCD bit, ASEXT[4] and the DR bit, CNTLB[3]:

Table 23. Clock Mode Parameters

CCD	DR	Clock Mode
0	0	16
0	1	64
1	0	1
1	1	Reserved, Do Not Use

The Time Constant registers value (TC) for a given baud rate is computed as follows:

$$TC = (f_{PHI} (2 * \text{baud rate} * \text{Clock Mode})) - 2$$

DCD and CTS Auto-Enable Modes

The Z382 ASCIs feature optional modes that allow external control of receive and transmit operations. DCD Auto-Enable mode is invoked by writing a 0 to the channel's DCD Disable bit, ASEXT[6]. In this mode, the receiver is enabled only when the channel's $\overline{\text{DCD}}$ input is asserted (Low) and its RE bit its CNTLA is 1. If the $\overline{\text{DCD}}$ input is negated (High) while in AUTO-ENABLE mode, the following conditions apply:

- Any receive operation currently in progress is aborted and further receipt of data from the RXA input is stopped.
- Data in the receive FIFO is lost and the Receiver Data Ready (RDRF) status bit, STAT[7], is held at 0.
- The PE, FE and OVRN receive error flags (STAT[6:4]) are held at 0.

RDRF, OVRN, PE and FE does not resume normal operation until $\overline{\text{DCD}}$ is asserted (Low) and the status register is read. The first read of STAT after $\overline{\text{DCD}}$ goes Low, while enabling normal operation, still indicates that the $\overline{\text{DCD}}$ input is High (STAT[2]). Read STAT twice to ensure that STAT[2] provides the correct status for the input. See Figure 17.

If DCD Disable is 1, the auto-enable mode is disabled and the $\overline{\text{DCD}}$ pin has no effect on receiver operation, but its state can be monitored via CNTLA[2].

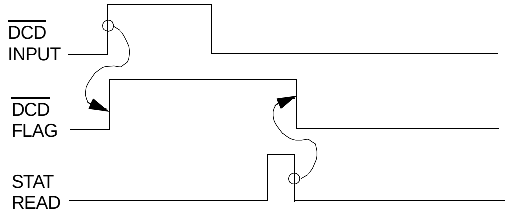


Figure 17. $\overline{\text{DCD}}$ Status Timing Diagram

Clearing CTS Disable, ASEXT[5], to 0 activates the CTS AUTO-ENABLE mode. In this mode, the transmitter is enabled only when the channel's $\overline{\text{CTS}}$ input is asserted (Low) and its TE bit, CNTLA[5], is 1. If $\overline{\text{CTS}}$ is negated (High) while in AUTO-ENABLE mode, transmission of the character currently being shifted out of the TSR is completed. Additionally, any data currently in the TDR is also transmitted. The transmitter is then disabled. However, the previous contents of the transmitter data register and the TDRE flag, STAT[1], are not affected.

If CTS Disable is 1, the AUTO-ENABLE mode is disabled and the $\overline{\text{CTS}}$ pin has no effect on transmitter operation, but its state can be monitored via CNTLB[5].

Reset, and STANDBY Modes

During Reset and in STANDBY mode an ASCI is forced to the following conditions:

- FIFO Empty
- All Error Bits Cleared (including those in the FIFO)
- Receive Enable Cleared (CNTLA[6] = 0)
- Transmit Enable Cleared (CNTLA[5] = 0)

ASCI Registers

Each ASCI contains the following I/O mapped user-accessible registers, as described in Table 24:

Table 24. ASCI Register Addresses

	380C I/O Address	
Register	**ASCI0**	**ASCI1**
Control Register A	0000h	0001h
Control Register B	0002h	0003h
Status Register	0004h	0005
Transmit Data Register	0006h	0007h
Receive Data Register	0008h	0009h
Extension Control Reg.	0012h	0013h
Time Constant Low	001Ah	001Ch
Time Constant High	001Bh	001Dh
DMA Control Register	0038h	0039h

Register Descriptions

Details on each of the above registers are provided in the sections which follow. In the descriptions:

- HHHHh = registers address in the 380C I/O space, in hexadecimal
- DEF = default value after RESET
- X = indicates that the default value of the bit is indeterminate

ASCI Control Register A (CNTLA0, CNTLA1)

ASCI Control Register A **CNTLA0 0000h, CNTLA1, 0001h**

BIT	7	6	5	4	3	2	1	0
R	Multi-Processor Enable	Receiver Enable	Transmitter Enable	Request to Send	Multi-Processor Bit Receive	Mode Select		
W					Error Flag Reset			
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W	Description
7	Multiprocessor Enable	R/W	The ASCI has a multiprocessor communication. mode that utilizes an extra data bit for selective communication when a number of processors share a common serial bus. Multiprocessor data format is selected when the multiprocessor mode bit (MP, CNTLB[6]) in the corresponding CNTLB register is set to 1. If multiprocessor mode is not selected (MP = 0), multiprocessor enable (MPE) has no effect. If multiprocessor mode is selected, MPE enables or disables the “wake-up” feature as follows. If MPE is 1, only received bytes in which the multiprocessor bit (MPB) equals 1 are treated as valid data characters and loaded into the receiver FIFO with corresponding error flags in the status FIFO. Bytes with MPB = 0 are ignored by the ASCI. If MPE is 0, all bytes are received by the ASCI, regardless of the state of the MPB data bit.
6	Receiver Enable	R/W	When set to 1, the ASCI receiver is enabled. When reset to 0, the receiver is disabled and any receive operation in progress is aborted. However, the previous contents of the receiver data and status FIFOs are not affected. If the channel is programmed for DCD Autoenable via ASEXT[6], the $\overline{\text{DCD}}$ input must be Low in order for the receiver to be enabled.
5	Transmitter Enable	R/W	When set to 1, the ASCI transmitter is enabled. When reset to 0, the transmitter is disabled and any transmit operation in progress is aborted. However, the previous contents of the transmitter data register and the TDRE flag are not affected. If the channel is programmed for CTS Autoenable via ASEXT[5], the $\overline{\text{CTS}}$ input must be Low in order for the transmitter to be enabled.
4	Request to Send	R/W	Clearing RTS to 0 asserts the corresponding $\overline{\text{RTS}}$ output pin (Low) while setting RTS to 1 deasserts that pin (High).

Bit(s)	Function	R/W	Description
3	Multiprocessor Bit Receive	R	When multiprocessor mode is enabled (CNTLB[6] = 1), this bit, when read, contains the value of the MPB bit for the data byte currently available at the Receive Data Register (the top of the receiver FIFO).
	Error Flag Reset Command	W	When a zero is written to this bit, the error flags (OVRN, FE, and PE in STAT and BRK in ASEXT) are cleared to zeros. Writing a 1 to this bit has no effect and is not required.
2:0	ASCI Data Format Mode 2, 1,	R/W	These bits program the ASCI data format. Bit — Function — Bit = 0 — Bit = 1 2 — MOD2: Number of Data Bits — 7 — 8 1 — MOD1: Parity Enable — No Parity — With Parity 0 — MOD0 Number of Stop Bits — 1 — 2 The following serial data formats can be selected: | Start bit | 7 or 8 data bits | Optional Parity bit | 1 or 2 stop bits | If MOD1 = 1, a parity bit is appended to the data bits in the transmitted data and the bit is checked on received data PEO, CNTLB[4], selects even or odd parity.

ASCI Control Register B (CNTLB0, CNTLB1)

ASCI Control Register B **CNTLB0: 0002h, CNTLB1: 0003h**

BIT	7	6	5	4	3	2	1	0
R	Multi-Processor Bit Transmit (MPBT)	Multi-Processor Mode (MP)	Clear To Send	Even/Odd	Clock Divide Ratio	Clock Source and Speed		
W			Prescale					
DEF	X	0	0	1	0	1	1	1

Bit(s)	Function	R/W	Description
7	Multiprocessor Bit Transmit (MPBT)	R/W	When multiprocessor communication format is selected (bit 6 = 1), Multiprocessor Bit Transmit (MPBT) is used to specify the multiprocessor data bit for transmission. If MPBT = 1, then a 1 is transmitted. in the bit position. If MPBT = 0, a 0 is transmitted.
6	Multiprocessor Mode (MP)	R/W	When Multiprocessor Mode (MP) is set to 1, the serial data format is configured for multiprocessor mode, adding a bit position whose value is specified in MPBT immediately after the specked number of data bits, and preceding the specified number of stop bits. Note that multiprocessor format has no provision for parity. The serial data format while in MP mode is: | Start bit | 7 or 8 data bits | Multiprocessor Bit | 1 or 2 stop bits | If MP = 0, the data format is based on MOD[2:0] in CNTLA and may include parity.

Bit(s)	Function	R/W	Description
5	Clear to Send	R	When read, this bit reflects the state of the $\overline{CTS}$ pin in a real-time, positive-logic fashion (High = 1, Low = 0).
	Prescale	W	Writing to this bit sets the BRG prescaler. If bits [2:0] in this register are not 111 (external clock), and the BRG mode bit in ASEXT is 0 (use preset BRG), writing a 0 sets the BRG prescaler to divide by 10, while writing a 1 sets it to divide by 30.
4	Parity Even/Odd		Party Even/Odd (PEO) controls the parity bit transmitted on the serial output and the parity check on the serial input. If PEO is 0, even parity is transmitted and checked. If PEO is 1, odd parity is transmitted and checked.
3	Divide Ratio		If the X1 bit in the ASEXT register is 0, the Divide Ratio (DR) bit specifies the divider used to obtain the baud rate from the data sampling clock. If DR is 0, divide-by-16 is used, while if DR is 1, divide-by-64 is used.
2:0	Source/Speed Select (SS[2:0])		If these bits are 111, the CKA pin is used as a clock input, and is divided by 1, 16, or 64 depending on the DR bit and the X1 bit in the ASEXT register. If these bits are not 111 and the BRG mode bit in ASEXT is 0, then these bits specify a power-of-two divider for the PHI clock as follows. SS2 SS1 SS0 Divider (DIV) 0 0 0 ÷1 0 0 1 ÷ 2 0 1 0 ÷ 4 0 1 1 ÷ 8 1 0 0 ÷ 16 1 0 1 ÷ 32 1 1 0 ÷ 64 1 1 1 External Clock

ASCI Time Constant Registers (ASTC0H, 0L, 1H, 1L).

ASCI Time Constant High Registers — ASTC0H: 001Bh ASTC1H: 001Dh

BIT	7	6	5	4	3	2	1	0
R W	ASCI Time Constant High							
DEF	0	0	0	0	0	0	0	0

ASCI Time Constant Low Registers — ASTC0L: 001Ah, ASTC1L: 001Ch

BIT	7	6	5	4	3	2	1	0
R W	ASCI Time Constant Low							
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W	Description
ASTCnH [7:0]	ASCI Time Constant High[1]	R/W	Holds the eight most significant bits of the 16-bit time constant used for baud rate generation when SS[2:0] in CNTLB are not `111` and BRG mode in ASEXT = 1. See “Baud Rate Generator” on page 248, for additional information
ASTCnL [7:0]	ASCI Time Constant Low	R/W	Holds the eight least significant bits of the 16-bit time constant used for baud rate generation when SS[2:0] in CNTLB are not `111` and BRG mode in ASEXT = 1. See “Baud Rate Generator” on page 248, for additional information

NOTES:

[1] The Time Constant register is also used for generation of fixed baud rates when BRG mode = 0. Writing into this register while operating in that mode may corrupt the received and/or transmitted data.

ASCI Extension Control Register (ASEXT0, ASEXT1)

ASCI Extension Control Register **ASEXT0, 0012h, ASEXT1: 0013h**

BIT	7	6	5	4	3	2	1	0
R	RXA State	DCD Disable	CTS Disable	CKA Clock Divider	BRG Mode	RX Interrupt On Start	Break Detect	Send Break
W	NO Function					Clear RX Start IRQ	No Function	
DEF	X	0	0	0	0	0	0	0

Bit(s)	Function	R/W	Description
7	RXA State	R	Provides the real time state of RXA, the channel's receive data input pin.
6	DCD Disable	R/W	Clearing this bit to 0 activates the DCD AUTO-ENABLE mode. In this mode, the receiver is enabled only when the channel's $\overline{\text{DCD}}$ input is asserted (Low) and is RE bit in CNTLA is 1. If the $\overline{\text{DCD}}$ input is negated (High) while in Auto- Enable mode, any receive operation currently in progress is aborted. See “DCD and CTS Auto-Enable Modes” on page 250 for additional information. If this bit is 1, the auto-enable mode is disabled and the $\overline{\text{DCD}}$ pin has no function, but its state can be monitored via STAT[2].

Bit(s)	Function	R/W	Description
5	CTS Disable	R/W	Clearing this bit to 0 activates the CTS AUTO-ENABLE mode. In this mode, the transmitter is enabled only when the channel's $\overline{\text{CTS}}$ input is asserted (Low) and its TE bit in CNTLA is 1. If $\overline{\text{CTS}}$ is negated (High) while in AUTO-ENABLE mode, the transmitter is disabled and any transmit operation in progress is aborted. However, the previous contents of the transmitter data register and the TDRE flag are not affected. If this bit is 1, the AUTO-ENABLE mode is disabled and the $\overline{\text{CTS}}$ pin has no function, but its state can be monitored via CNTLB.
4	CKA Clock Divider	R/W	When SS[2:0] in CNTLB are `111` the channel's clocking is obtained from its CKA input pin and this bit, in combination with DR in CNTLB, controls the divide ratio for that clock. If this bit is 1, the CKA input is used without division. If this bit is 0, the clock is divided by 16 or by 64, as programmed by DR. The following table summarizes the options: CCD DR ASCI Clock 0 0 CKA/16 0 1 CKA/64 1 0 1x CKA 1 1 Reserved
3	BRG Mode	R/W	When this bit is 1 and SS[2:0] in CNTLB are not `111` the channel's programmable clock divider is used as the clock source. If the bit is cleared to `0` and SS[2:0] are not `111`, the channel's clock is as given by SS[2:0] and PS in CNTLB. See “Baud Rate Generator” on page 248 for additional information.

Bit(s)	Function	R/W	Description
2	Rx Interrupt on Start	R	If software sets this bit to 1, a receive interrupt is requested (in a combinatorial fashion) when a start bit is detected on RXA. Such a receive interrupt is always followed by the setting of RDRF in the middle of the stop bit. After such an interrupt, software must write a zero to this bit to clear the interrupt request. One function of this feature is to wake the part from Sleep mode when a character arrives, so that the ASCI receives clocking with which to process the character. Another function is to ensure that the associated interrupt service routine is activated in time to sense the setting of RDRF in the status register, and to start a PRT for baud rate measurement at that time.
	Clear Rx Start Interrupt	W	Writing 0 to this bit clears the Rx Start interrupt request. Writing a 1 to this bit has no effect.
1	Break Detect	R	This status bit is `1` when a Break is detected, defined as a framing error with the data bits all equal to zero. The all-zero byte with is associated error bits are transferred to the FIFO if it is not full. If the FIFO is full, an overrun is generated, but the break, framing error and data are not transferred to the FIFO. Any time a break is detected, the receiver does not receive any more data until the RXA pin returns to a High state.
0	Send Break	R/W	Setting this bit to `1` forces the channel's transmitter data output pin, TXA, to a Low for as long as it remains set. Before starting the Break, the character currently being transmitted from the TSR is completely transmitted and the TDR must become empty (the break does not start until the last character loaded into the TDR is completely transmitted). If a character is loaded into the TDR while a Break is being generated, that character is held until the Break is terminated and it is transmitted.

ASCI Status Register (STAT0, STAT1)

ASCI Status Register **STAT0: 0004h, STAT1 0005h**

<table>
<tr><td>BIT</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>R</td><td>RX Data Register Full</td><td>Overrun Error</td><td>Parity Error</td><td>Framing Error</td><td rowspan="2">Receiver Interrupt Enable</td><td>Data Carrier Detect</td><td>TX Data Register Empty</td><td rowspan="2">Trans-mitter Interrupt Enable</td></tr>
<tr><td>W</td><td colspan="4">No Function</td><td colspan="2">No Function</td></tr>
<tr><td>DEF</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

Bit(s)	Function	R/W	Description
7	Receive Data Register Full	R	RDRF is 1 when the receiver transfers a character from the RSR into an empty Rx FIFO. If a framing or parity error occurs, RDRF is still set and the receive data (which generated the error) is still loaded into the FIFO. When there is more than one character in the FIFO, and software or a DMA channel reads a character, RDRF remains 1. RDRF is cleared to 0 when the FIFO becomes empty after reading the RDR, in STANDBY mode, and by a Reset
6	Overrun Error	R	An overrun occurs if the receive FIFO is full when the receiver completes assembly of a character and is ready to transfer it to the FIFO. If this condition occurs, the overrun error bit associated with the previous byte in the FIFO is set. In this case, the latest data byte is not transferred from the shift register to the FIFO and is lost. Once an overrun occurs, assembly of bytes continues in the shift register, but the receiver does not place any further data in the FIFO until the last good byte received (the byte with the associated overrun error bit set) comes to the top of the FIFO and sets the Overrun latch. Software clears the Overrun latch by writing a 0 to the EFR bit, CNTLA[3].

Bit(s)	Function	R/W	Description
5	Parity Error	R	A parity error is detected when parity generation and checking is enabled by the MOD1 bit, CNTLA[2], and a character has been assembled in which the parity does not match that specified by the PEO bit, CNTLB[4]. PE is FIFOed and the error bit is not actually set until the associated data becomes available for reading in the RDR. Once set, the bit remains set until it is cleared by writing a 0 to the EFR bit, CNTLA[3].
4	Framing Error	R	A framing error is detected when the stop bit of a character is sampled as a zero (space). Like PE, FE is FIFOed and the error bit is not actually set until the associated data becomes available for reading in the RDR. Once set, the bit remains set until it is cleared by writing a 0 to the EFR bit, CNTLA[3].
3	Receiver Interrupt Enable	R/W	Setting RIE to `1` asserts the channel's ASCI interrupt request when Start Detect, ASEXT[2], is set or if any of the following bits in STAT are true: RDRF (bit 7), OE (bit 6), PE (bit 5), or FE (bit 4). However, if the Rx DMA Enable bit in the channel's ASCI DMA Control Register is 1, the ASCI does not request an interrupt because the RDRF is true.
2	Data Carrier Detect	R	This bit reflects the state of the $\overline{\text{DCD}}$ pin in a real-time, positive-logic fashion (High = 1, Low = 0).
1	Transmit Data Register Empty	R	TDRE = 1 indicates that the Transmit Data register FIFO has at least one empty position and that the next data byte to be transmitted can be written into the TDR. If, after writing a byte to the TDR the FIFO is full, TDRE is cleared to `0`. It stays at a `0` until the ASCI transfers a byte from the TDR to the TSR and then TDRE is again set to `1`.
0	Transmit Interrupt Enable	R/W	Setting TIE to `1` asserts the channel's ASCI interrupt request when the channel's TDRE (bit 2) is true.

ASCI Receive Data Register (RDR0, RDR1)

ASCI Receive Data Register **RDR0: 0008h, RDR1: 0009h**

BIT	7	6	5	4	3	2	1	0
R	Receive Data							
W	No Function							
DEF	X	X	X	X	X	X	X	X

Bit(s)	Function	R/W	Description
[7:0]	Receive Data	R	The RSR is the top of the receiver FIFO. If RDRF, STAT[7], is true, there is a valid character in the RDR ready to be read by the 380C or ready to be transferred by the RxDMA process. The FIFOed status for the character in the RDR is available in STAT[6:4]. If character-by-character error status is required, the following procedure should be followed: 1. Read STAT to obtain the error status for the character in the RDR. 2. If any error bits are set, write a 0 to the EFR bit, CNTLA[3], to clear them. 3. Read this register. A POP instruction is performed on both the data and error FIFOs when the character is read from the RDR.

ASCI Transmit Data Register (TDR0, TDR1)

ASCI Transmit Data Register **TDR0: 0006h, TDR1: 0007h**

BIT	7	6	5	4	3	2	1	0
R	No Function							
W	Transmit Data							
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W	Description
[7:0]	Transmit Data	W	The TDR holds the next character to be transmitted. The ASCI moves the contents of the TDR to its Transmit Shift Register if it is idle or when it has completed serialization of the previous character. It simultaneously sets the TDRE flag, STAT[1], and generates an interrupt If so enabled by TIE in STAT, to indicate that the TDR is ready to accept another character.

ASCI DMA Control Register (ADCR0, ADCR1)

ASCI DMA Control Register (ADCR0, 0038h, ADCR1: 0039h

BIT	7	6	5	4	3	2	1	0
R W	TxDMA Enable	TxDMA Channel			RxDMA Enable	RxDMA Channel		
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W	Description
7	TxDMA Enable		If this bit is 1 DMA operation is enabled for the channel's transmitter, using the DMA channel identified in bits 6:4 of this register. This DMA process loads characters into the TDR as TDRE indicates that it has become available for data. Because the data flowing through the ASCIs has no organization that can be discerned by the hardware, the ASCIs do not use any of the following features of the DMA channels: *with Command*, *Notify at End of Buffer*, or *with Status*.
6:4	TxDMA Channel		Identifies the DMA channel to be used for the transmit DMA process when bit 7 is 1.

Bit(s)	Function	R/W	Description
3	RxDMA Enable		If this bit is 1 DMA operation is enabled for the channel's receiver, using the DMA channel identified in bits 2:0 of this register. This DMA process reads characters from the RDR as RDRF indicates that data is available. Since the data flowing through the ASCIs has no organization that can be discerned by the hardware, the ASCIs do not use any of the following features of the DMA channels: *with Command*, *Notify at End of Buffer*, or *with Status*. The RxDMA request is disabled when any of the error flags (PE, FE or OVRN) is set, so that software can know with which character a problem is associated. Also, if this bit is 1, the ASCI does not request a Receive interrupt when is RDRF flag is 1. Rx DMA Enable does not affect interrupt requests because of other causes.
2:0	RxDMA Channel		Identifies the DMA channel to be used for the receiver DMA process when bit 3 is 1 and the error bit is not actually set until the associated data becomes available for reading in the RDR. When set, the bit remains set until it is cleared by writing a 1 to the EFR bit, CNTLA[3].

HDLC SERIAL CHANNELS

The Z80382 features three high speed serial channels, each comprised of a transmitter and a receiver, which can operate in HDLC or transparent (unframed) modes. All data transfers to and from the HDLC channels are carried out by the DMA channels. Each HDLC transmitter and receiver must have an assigned DMA channel to perform its function. Facilities for interrupt-driven or polled transfer of HDLC data are not provided. Eight-character FIFOs on both the transmit and receive side reduce the possibility of overrun and underrun conditions to a minimum, at data rates up to 10 Mbps.

Each HDLC channel has five associated I/O pins. Software selects whether each channel's I/O is on device pins or on the internal TDM highway (the GCI/SCIT bus in the Z80382). If device pins are used, each channel's I/O are configured as either a classic synchronous serial interface, or as the interface to an external time-division multiplexed TDM) highway. In the latter case, a programmable Time Slot Assigner selects the time slot during which the channel is active, for each transmitter and receiver.

The differences in pin use for the two modes are described in Table 25.

Table 25. Pin Use for TDM and Full-Time Operation

Pin	TDM Operation[1,2]	Full-Time Operation[2]
TxD	TxD — A (bused) line onto which the channel's transmitter places data in its programmed time slot. This pin is tristated outside the channel's time slot.	TxD — Driven full time by the channel's transmitter when it is enabled.
RxD	RxD — A (bused) line from which the channel's receiver takes data in its programmed time slot, using the clock supplied on the channel's RxC/BCL pin.	RxD — Sampled in every bit time by the channel's receiver, using the clock supplied on the channel's RxC/BCL pin, or the clock supplied by the internal PLL.
RxC/BCL	BCL — Common 1x bit clock for the channel's receiver and transmitter.	RxC — Clock for the channel's receiver and, optionally, for the channel's transmitter. When operating from the internal DPLL, the receiver clock can be programmed to be output on this pin.
TxC/FSC	FSC — Frame Sync, synchronous to BCL. The channel's transmitter and receiver measure their time slots independently from the rising edge of this signal. The duration of FSC can be one or more BCL cycles.	TxC — This pin can be programmed as an output for the channel's transmitter clock, if that transmitter is programmed to use clocking from its associated baud rate generator. It can also be used as an input for an external clock for the channel's transmitter and, optionally, for its receiver.

Table 25. Pin Use for TDM and Full-Time Operation (Continued)

Pin	TDM Operation[1,2]	Full-Time Operation[2]
$\overline{\text{TxEN}}$	TxEN — An active Low output from the channel's transmitter indicating its time slot, that is, is placing data on TxD. An optional enable for an external driver.	TxEN — An active low output which is asserted when it whenever the channel's transmitter is enabled.

NOTES:

1. Special modes allow the receivers and transmitters to interface to the common GCI/SCIT (IOM-2) TDM bus. See TM R[5:3] and RMR[5:3] mode '000' and mode '001' descriptions later in this document.
2. In TDM modes, the transmitter changes the TxD output data on rising edges of the transmitter clock, while in other modes it does so on falling edges of the transmitter clock. In any mode, the receiver can be programmed to sample the input data from RxD on the rising or falling edges of the receiver clock.

Clock, Data, and Sync Timing

In TDM modes, a rising edge on Frame Sync signals the start of each TDM frame. Each frame must set up to, and hold from, a falling clock edge on BCL. In TDM modes, TxD is switched from rising clock edges, while in non-TDM modes the frame is switched from falling clock edges. In any mode, the clock edge on which the receiver samples RxD is programmable in the Receiver Interrupt Register.

Figure 18 illustrates the HDLC channel block diagram.

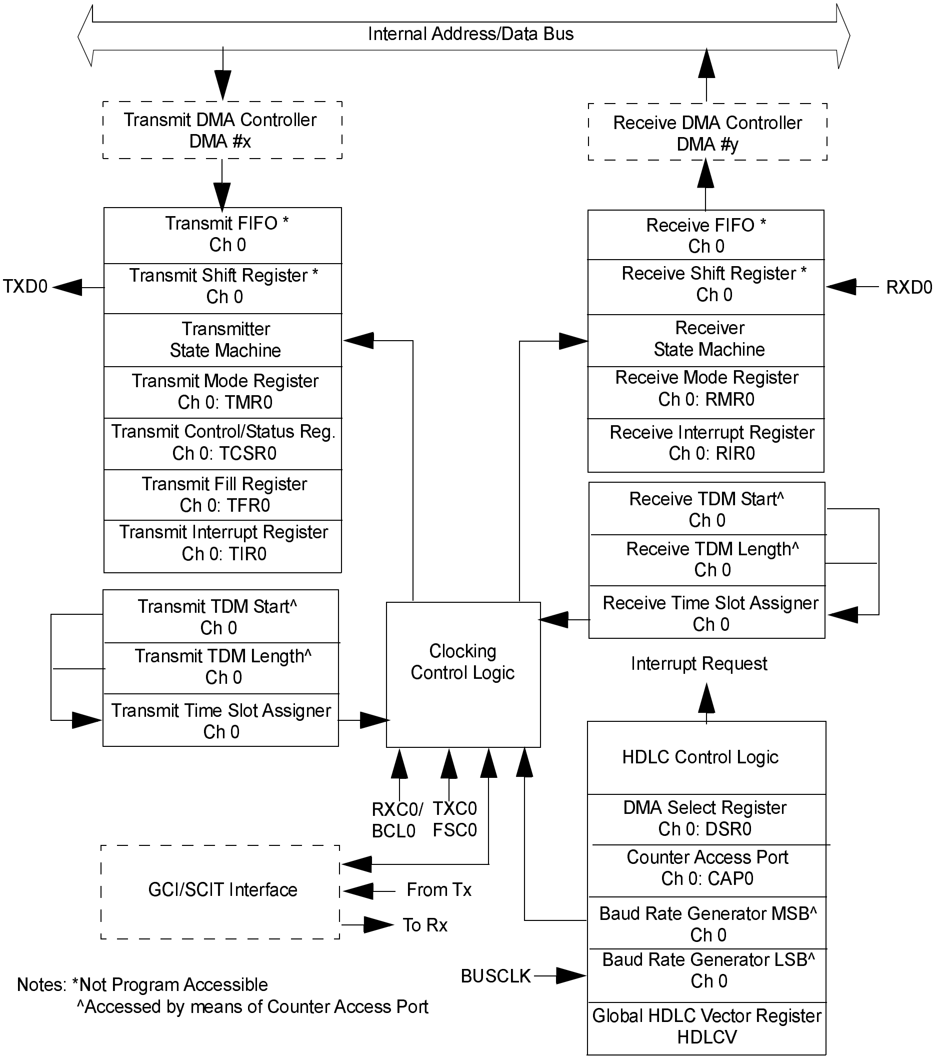

Internal Address/Data Bus
Transmit DMA Controller DMA #x
Receive DMA Controller DMA #y
Transmit FIFO * Ch 0
Transmit Shift Register * Ch 0
TXD0
Transmitter State Machine
Transmit Mode Register Ch 0: TMR0
Transmit Control/Status Reg. Ch 0: TCSR0
Transmit Fill Register Ch 0: TFR0
Transmit Interrupt Register Ch 0: TIR0
Receive FIFO * Ch 0
Receive Shift Register * Ch 0
RXD0
Receiver State Machine
Receive Mode Register Ch 0: RMR0
Receive Interrupt Register Ch 0: RIR0
Receive TDM Start^ Ch 0
Receive TDM Length^ Ch 0
Receive Time Slot Assigner Ch 0
Transmit TDM Start^ Ch 0
Transmit TDM Length^ Ch 0
Transmit Time Slot Assigner Ch 0
Clocking Control Logic
Interrupt Request
HDLC Control Logic
DMA Select Register Ch 0: DSR0
Counter Access Port Ch 0: CAP0
Baud Rate Generator MSB^ Ch 0
Baud Rate Generator LSB^ Ch 0
Global HDLC Vector Register HDLCV
RXC0/ BCL0
TXC0 FSC0
GCI/SCIT Interface
From Tx
To Rx
BUSCLK
Notes: *Not Program Accessible
^Accessed by means of Counter Access Port

Figure 18. HDLC Channel Block Diagram (One of Eight Channels)

Interface with the GCI/SCIT TDM Module

The interface between an HDLC channel and the GCI/SCIT module includes:

TxD A bused line onto which HDLC transmitters place data in their time slots, as directed by software programming.

RxD A bused line from which HDLC receivers take data in their time slots, as directed by software programming.

BCL A common bit clock for HDLC transmitters and receivers. Transmitters change data on TxD on falling edges of BCL, and receivers sample data from RxD on rising edges of BCL.

FSC Frame Sync, synchronous to BCL. Transmitters and Receivers measure their time slots independently from the rising edge of this signal. The duration of FSC can be one or more BCL cycles.

$\overline{\text{TxEN}}$ An active Low output from each Transmitter to the GCI/SCIT module, indicating its time slot, that is, when it is placing data on TxD.

TDM Processing

When the Transmit (Receive) TDM Length register is non-zero, the transmitter (receiver) activates its time slot assigner to clock Tx (Rx) data only within its time slot. If a TDM Start register is non-zero, then after each pulse on FSC, the time slot assigner blocks clocking for the number of bits specified by the TDM Start register. Then, or immediately at Frame Sync if the start value is zero, it enables clocking for the number of bits specified by the TDM Length register. Thereafter, it again blocks clocking until the next Frame Sync pulse. Table 26 lists the Start and Length values for the subchannels of the GCI/SCIT (IOM2) frame.

Table 26. Channel Start and Length Values

Channel	Start	Length
B1 (64K bps)	0	8
B1 (56K)	0	7
B2 (64K)	8	8
B2 (56K)	8	7
D	24	2
IC1	32	8
IC2	40	8

DMA Lists and Transmitter Operation

Note: Please refer to the description of status byte/types in Section 5.3 in conjunction with this topic.

In HDLC mode, a frame to be transmitted can be contained in one or more DMA buffers. The DMA list entry for the last (or only) buffer of a frame must have its status byte/type coded as `Ready Buffer, Notify at End of Buffer`. This condition causes the transmitter to send the CRC (if enabled) and a closing Flag after the last byte of the buffer. Buffers that do not include the end of a frame should have their status byte/types coded as `Ready Buffer, no End of Buffer Notification`.

Two control fields for the transmitter do not reside in processor-accessible register bits, but can be controlled separately for each frame in status byte/types in the DMA list:

- How many bits the transmitter sends from the last byte of the frame, and
- Whether the transmitter sends its accumulated CRC at the end of the frame.

Either of these items can be changed automatically from one frame to the next if the status byte/type for the frame is coded as `Ready Buffer, with Command` and the control bits of that byte are set as described in Table 27:

Table 27. /Status Byte/Type Coding

Bits)	Function
[5]:	Transmit CRC. In HDLC modes, this bit controls whether the transmitter sends its accumulated CRC at the end of each frame. It is written as 1, except in an *end-to-end CRC* application, in which the data buffers) include a CRC that was received from another station. (The HDLC receiver always includes CRCs in memory buffers.) The internal control latch that is loaded from this bit resets to 1.
[2:0]	In HDLC modes, this field controls how many bits are sent from the last byte of the last buffer for the current frame, as detailed below. (Reset value = 111).

Bits [2:0] Value	LSB First (RIR[5] = 0)	MSB First (R[R[5] = 1)
000	Bit [0]	Bit [7]
001	Bits [1:0]	Bits [7:6]
010	Bits [2:0]	Bits [7:5]
011	Bits [3:0]	Bits [7:4]
100	Bits [4:0]	Bits [7:3]
101	Bits [5:0]	Bits [7:2]
110	Bits [6:0]	Bits [7:1]
111	Bits [7:0]	Bits [7:0]

If a frame spans more than one DMA buffer, this status byte/type can be in the DMA list entry for any or all of the buffers for the frame. The last status byte/type for a frame that is coded `with Command` is the one that determines what happens at the end of the frame and thereafter. If these parameters are the same for every frame, subsequent buffers/frames need not be coded `with Command`. if the Reset values *Append CRC* and *8 Bits in Last Byte* are appropriate for all frames, no status byte/type needs to be coded *with Command*.

In HDLC mode or in TRANSPARENT mode with the Underrun Wait bit set to `1`, Completed Buffer codes in status byte/types in Transmitter DMA lists are stored as `with Status` if the transmitter encountered an Underrun while sending the data in the buffer. In all other cases, status byte/types in Transmitter DMA lists are stored as *no status*.

DMA Lists and Receiver Operation

Note: Please refer to the description of status byte/types in “DMA Channel Operation” on page 222 in conjunction with this topic.

HDLC receivers do not use the Command nor End of Buffer notification features of the DMA channels. All `Ready Buffer` codes in status byte/types in Receiver DMA lists are equivalent.

A received frame can be contained in one DMA buffer, or can span two or more buffers. The end of a frame always makes the receiver terminate its current DMA buffer and store frame status in its status byte/type.

When a buffer is filled with receive data, without the last character of the frame being stored in that buffer, that buffer's status byte/type is stored as `Completed Buffer (no Status)`. Buffers that include the last character of a frame, and buffers that could not be completed because the receiver encountered an Overrun condition, are stored as `Completed Buffer (with Status)`. The least significant six bits of such a status byte/type are encoded as described in Table 28, to indicate the status of the buffer.

Table 28. Type/Status Coding for Six Least Significant Bits

Bits)	Function
[5]:	If this bit is 1, the receiver encountered an Overrun condition while filling this buffer.
[4:3]	Indicate the following conditions: 00: Should only occur with an Overrun condition 01: Frame ended with Abort 10: Frame ended with Closing Flag, CRC correct[1] 11: Frame ended with Closing Flag, CRC incorrect[1]
[2:0]	Indicate which bits are valid in the last byte of the last buffer for the current frame, as detailed below.

NOTES:

[1] If frames don't include a CRC, software should consider the 10 and 11 values equivalent. Given random data and 16-bit CRCs, 10 occurs about once in 65K frames in this case.

Bits [2:0] Value	LSB First (RIR[5] = 0)	MSB First (RIR[5] = 1)
000	Bit [7]	Bit [0]
001	Bits [7:6]	Bits [1:0]
010	Bits [7:5]	Bits [2:0]
011	Bits [7:4]	Bits [3:0]
100	Bits [7:3]	Bits [4:0]
101	Bits [7:2]	Bits [5:0]
110	Bits [7:1]	Bits [6:0]
111	Bits [7:0]	Bits [7:0]

Note: In HDLC mode, an Overrun condition automatically forces the receiver into Hunt mode. After the DMA channel has emptied the Rx FIFO and the receiver has seen at least one Flag, these two modules start receiving the next frame into the next Rx DMA buffer.

In Transparent mode, an Overrun makes the receiver terminate the buffer after the DMA channel has read the data preceding the overrun out of the Rx FIFO. It then starts receiving data into the next buffer, maintaining the same byte synchronization.

When a buffer is filled with receive data, without the last character of the frame being stored in that buffer, that buffer's status byte/type is stored as `Completed Buffer (no Status)`. Buffers that include the last character of a frame, and buffers that could not be completed because the Receiver. encountered an Overrun condition, are stored as `Completed Buffer (with Status).`

Passing Frames from Receiver to Transmitter

The residual bit count provided in bits [2:0] of `Completed Buffer (with Status)` codes in status byte/types of Rx DMA lists, expresses the length of the frame in the same way as do bits [2:0] of Ready Buffer *(with Command)* codes in Tx DMA lists. However:

- If in LSB FIRST mode, the Receiver stores received bits in the MS, bits of the last byte of such a frame, while the Transmitter takes bits from the LS bits of the last byte.
- If in MSB FIRST mode, the Receiver stores received bits in the LS bits of the last byte of such a frame, while the Transmitter takes bits from the MS bits of the last byte.

Because of this condition, in a `CRC pass-through` or `no CRC` application, software shifts the fast received character of a frame that is not a multiple of eight bits long before it sends the frame back out. Such software proceeds as follows:

1. Masks bits [2:0] of the *Completed Buffer, with Status* byte
2. 1f these bits are `111`, does nothing.
3. Otherwise, one's complements these three bits (or subtract them from seven).
4. Right-shifts (left-shift if in MSB FIRST mode) the last character to be sent in the frame by the number of bits given by the result of 3 above.

When a received frame includes a CRC, but it is to be re-computed for subsequent transmission, it is not necessary for software to shift the last byte in the last receive buffer of that frame. Instead, software decreases the effective frame length for the transmitter, from that reported in reception, by two bytes to discard a 16-bit CRC or by four bytes to discard a 32-byte CRC.

In all cases software sets the three LS bits of (at least one of) the status byte/type(s) for the Tx frame to the value reported in the status byte/type for the (last buffer of the) received frame. In protocols in which frames must be a multiple of eight bits in length, this value must always be `111`.

HDLC Interrupts

There is an Interrupt Pending (IP) bit for each HDLC Transmitter and Receiver, but only one Interrupt Under Service (IUS) bit for all of them. If the interrupt routine for a transmitter or receiver wants to enable nested interrupts from other devices, it should re-enable interrupts near its start and then clear the IP bit — the IUS bit being set prevents another HDLC interrupt. An ISR that disallows nested interrupts can clear IP and IUS near its start, in the same register write.

After clearing IP, the lSR reads status and its DMA channel's List Address Register, and scans the DMA list to see what has happened. At this point the ISR can, optionally, read the HDLC Vector register to detect if any other HDLC Transmitter or Receiver has an interrupt pending.

Finally, if the ISR allowed nested interrupts, it clears IUS to allow future HDLC interrupts. Disabling interrupts again with a DI instruction, before clearing IUS, prevents stack filling (*tail recursion*) if interrupt traffic is heavy.

Most of the interrupt requirements for HDLC reception can be handled by enabling Status interrupts in the DMA channel associated with each receiver. The only receiver interrupt condition that is not handled by this means is the Idle condition. Idle interrupts are controlled by the Receiver Interrupt register.

Baud Rate Generator and DPLL

If an HDLC channel's transmitter clock is taken from its Baud Rate Generator (BRG), and/or its receiver clock is taken from its DPLL, then the channel's BRG operates. A BRG counts down from the 16-bit value programmed into its BRG Time Constant LS and MS registers, using the chip's BUSCLK. Each time the value is zero, the BRG toggles its output to the DPLL, and one clock later it reloads the value from the TC registers. dividing BUSCLK by (2 x (TC+1)).

If an HDLC channel's receiver clocking is taken from its DPLL, software programs the channel's Time Constant registers with a 16-bit value that corresponds to sixteen times the nominal data rate. Conceptually, when the DPLL detects a change on the raw received data (before NRZI decoding), it clears a counter that is incremented at 16x the nominal bit rate. Half a bit time thereafter, it provides an active edge on its Rx clock output. In the absence of further data transitions it provides the Rx clock as the BRG output divided by 16.

Note: This technique is reliable only when the receive data stream includes regular transitions to re-sync the clock to the data. One way to ensure such transitions in HDLC Mode is by using NRZI encoding and decoding.

When the receive clock is selected from the DPLL, the BRG automatically invokes another (free-running) 4-bit counter to provide the transmitter clock if the BRG is selected as the source of transmitter docking.

The TC value to be written to the Counter Access Port can be computed from the BUSCLK frequency, FBUS, and desired data frequency, F_{SER}:

when not using the DPLL:TC = $(F_{BUS}/(2 \text{ x } F_{SER}))$ -1
$F_{SER} = F_{BUS}/(2 \text{ x } (TC +1))$

when using the DPLL: TC = $(F_{BUS}/(32 \text{ x } F_{SER})$ -1
$F_{SER} = F_{BUSC}/32 \text{ x } (TC + 1))$

Note: If the /2 or X2 option is selected in the Clock Control Register, F_{BUS} is half or twice the crystal or oscillator frequency, respectively.

If the formula above yields a value between -1 and about 0.02 (more likely when using the DPLL), the serial rate is too fast to be generated from BUSCLK. If the formula yields a value that is not an integer, compute the error as follows:

% Error = $100 \text{ x } (1 - (F_{SER} \text{ x N x } (TC+1)/_{FBUS}))$,
where N = 2 or 32 as above.

A larger error than ± 2% typically means that the desired serial rate can not be generated accurately enough from the bus clock. Traditional serial rates can be accurately generated from a 29.4912 MHz crystal or oscillator in x1 mode, or a 14.7456 MHz crystal or oscillator in x2 mode.

HDLC Pin Usage

As described previously, there are five signal pins associated with each HDLC channel. The usage of these pins is controlled primarily by the values of the transmitter and receiver mode and configuration fields in the Transmit and Receive Mode Registers. Table 29 and Table 30 provide additional information on how each channel's pins are used under various operating conditions.

Table 29. HDLC Pin Usage by Rx/Tx Configuration[1]

	Transmitter set to 'n' uses its own pins as follows:[2]					Receiver set to 'n' uses its own pins as follows:[2]				
Mode (n)	**TxD**	**TxC/ FSC**	**TxEN**	**RxD**	**RxC/ BCL**	**TxD**	**TXC/ FSC**	**TxEN**	**RxD**	**RxC/ BCL**
000[3]	—	—	TxEN	—	—	—	—	—	—	—
001[4]	TxD	TxC	TxEN	—	—	—	—	—	RxD	RxC
010	TxD	FSC[5]	TxEN	—	TxC	—	FSC[5]	—	RxD	RxC
011	TxD	TxC	TxEN	—	—	—		—	RxD	RxC
100	TxD	—[6]	TxEN	—	—	—	—	—	RxD	—[7]
101	TxD	TxC[6]	TxEN	—	—	—	—	—	RxD	RxC[7]
110[8]	—	—	—	—	—	—	—	—	—	—
111[8]	—	—	—	—	—	—	—	—	—	—

NOTES:

1. HDLC pins for channels 0 and 1 are shared with the ASCIs, and pin usage may be affected by programming of the Pin Multiplexing register.
2. Underline indicates used as input, non-underline indicates used as output, — indicates unused unless used by the other half of the channel.
3. For HDLC channel 2, if the Transmitter Configuration is set to 000, the Receiver Configuration field must also be 000.
4. Not permitted for HDLC channel 2. Pins are required for GCI/SCIT I/O.
5. Used only if unit's TDM length field is non-zero.
6. Tx clocking comes from channel's BRG.
7. Rx clocking comes from channel's DPLL
8. Mode not used for transmitter, loopback mode for receiver.

Table 30. HDLC Channel Configurations and Signal Pins

Channel Configuration Conditions[1]					HDLC Pin or Facility Where Listed Signal Is Input or Output[2]								
Channel n	Tx Config.[3]4 (n)	Rx Config. [n]	Tx TDM Length [n]	Rx TDM Length [n]	Frame Sync [n]	Tx Data Out [n]	Tx Clock In [n]	Local Tx Data Out [n]	Local Tx Clk Out (n)	Rx Data In [n]	Rx Clk In [n]	Local Rx Data Out [n]	Local Rx Clk Out [n]
0 - 2	000	000	> 0	> 0	FSC/RxD[2]	Note[4]	Note[5]	—	—	Note[6]	Note[5]	—	—
0 -1[7]	000	001	> 0	> 0	FSC/RxD[2]	Note[4]	Note[5]	—	—	Note[6]	Note[5]	RxD[n]	RxC/BCL[n]
0 -1[7]	001	000	> 0	> 0	FSC/RxD[2]	Note[4]	Note[5]	TxD[n]	TxC/FSC[n]	Note[6]	Note[5]	—	—
0 -1[7]	001	001	> 0	> 0	FSC/RxD[2]	Note[4]	Note[5]	TxD[n]	TxC/FSC[n]	Note[6]	Note[5]	RxD[n]	RxC/BCL [n]
0 - 2	010	010	> 0	> 0	TxC/FSC[n]	TxD[n]	RxC/BCL[n]	—	—	RxD[n]	RxC/BCL[n]	—	—
0 - 2	010	010	0	0	—	TxD[n]	RxC/BCL[n]	—	—	RxD[n]	RxC/BCL[n]	—	—
0 - 2	010	011	0	0	—	TxD[n]	RxC/BCL[n]	—	—	RxD[n]	TxC/FSC[n]	—	—
0-2	010	100	0	0	—	TxD[n]	RxC/BCL[n]	—	—	RED[n]	DPLL [n]	—	—
0 - 2	011	010	0	0	—	TxD[n]	TxC/FSC[n]	—	—	RxD[n]	RxC/BCL[n]	—	—
0 - 2	011	011	0	0	—	TxD[n]	TxC/FSC[n]	—	—	RxD[n]	TxC/FSC[n]	—	—
0-2	011	100	0	0	—	TxD(n)	TxC/FSC[n]	—	—	RxD[n]	DPLL [n]	—	—
0 - 2	011	101	0	0	—	TxD[n]	TxC/FSC[n]	—	—	RxD[n]	DPLL [n]	—	RxC/BCL [n]
0 - 2	100	010	0	0	—	TxD[n]	BRG[n]	—	—	RxD[n]	RxC/BCL[n]	—	—
0 - 2	100	011	0	0	—	TxD[n]	BRG[n]	—	—	RxD[n]	TxC/FSC[n]	—	—
0-2	100	100	0	0	—	TxD[n]	BRG[n]	—	—	RxD[n]	DPLL [n]	—	—

Table 30. HDLC Channel Configurations and Signal Pins (Continued)

Channel Configuration Conditions[1]					HDLC Pin or Facility Where Listed Signal Is Input or Output[2]								
Channel n	Tx Config.[3] 4 (n]	Rx Config. [n]	Tx TDM Length [n]	Rx TDM Length [n]	Frame Sync [n]	Tx Data Out [n]	Tx Clock In [n]	Local Tx Data Out [n]	Local Tx Clk Out (n)	Rx Data In [n]	Rx Clk In [n]	Local Rx Data Out [n]	Local Rx Clk Out [n]
0-2	100	101	0	0	—	TxD[n]	BRG[n]	—	—	RxD[n]	DPLL [n]	—	RxC/ BCL[n]
0-2	101	010	0	0	—	TxD[n]	BRG[n]	—	TxC/ FSC[n]	RxD[n]	RxC/ BCL[n]	—	—
0-2	101	100	0	0	—	TxD[n]	BRG[n]	—	TxC/ FSC[n]	RxD[n]	DPLL [n]	—	—
0-2	101	101	0	0	—	TxD[n]	BRG[n]	—	TxC/ FSC[n]	RxD[n]	DPLL [n]	—	RxC/ BCL[n]
0 - 2	010	110	> 0	> 0	TxC/ FSC[n]	TxD[n]	RxC/ BCL[n]	—	—	TxD[n]	RxC/ BCL[n]	—	—
0 -2	01x/10x	110	0	0	—	Note[8]	Note[8]	Note[8]	Note[8]	TxD[n]	Tx Clock In [n]	—	—
0-2	01x/10x	111	0	0	—	Note[8]	Note[8]	Note[8]	Note[8]	TxD[n]	DPLL [n]	—	—

NOTES:

1. Although other combinations of configuration conditions are possible, they are not useful and should not be used.
2. The term 'local' refers to the channel's outputs used primarily for test or monitoring purposes.
3. Transmitter configurations '11x' are not permitted.
4. DU/TxC[2]/FSC[2] except in subframe 1 (IC1 and IC2 fields) with GCIR[7] = 1, in which case output is on DD/TxD[2].
5. Tx and Rx clock inputs are from the DCL/RxC[2]/BCL[2] pin. This clock is divided by two before being used by the Tx and Rx.
6. DD/TxD[2] except in subframe 1 (IC1 and IC2 fields) with GCIR[7] = 1, in which case Input is from DU/TxC[2]/FSC[2].
7. '001' not permitted for transmitter or receiver of channel 2.
8. See specific transmitter configuration above for source/destination of these signals.

HDLC Register Descriptions

The HDLC module includes the following I/O-mapped registers, listed in Table 31, that can be read and written by the 380C processor:

Table 31. HDLC I/O-Mapped Registers

Register Name	Type
HDLC Vector Register	Global
Transmit Mode Register	Per Channel
Transmit Control/Status Reg.	Per Channel
Transmit Interrupt Register	Per Channel
Transmit Fill Register	Per Channel
Receive Mode Register	Per Channel
Receive Interrupt Register	Per Channel
DMA Select Register	Per Channel
Counter Access Port	Per Channel

Note: The Counter Access Port can be used to access the LS and MS bytes of the Baud Rate Generator's Time Constant, and the transmit and receive TDM Start and TDM Length registers.

Details on each register are provided in the sections which follow. In the descriptions:

- `HHHHh` = register's address in the 380C I/O space, in hexadecimal
- DEF = default value after RESET
- x = indicates that the default value of the bit is indeterminate

Transmit Mode Register TMR0, 1, 2

Transmit Mode Register **Ch. 0: 0060h, Ch. 1: 0068h, Ch. 2: 0070h**

BIT	7	6	5	4	3	2	1	0
R W	Transmitter Mode		Transmitter Configuration			DMA Request Half/One	Underrun Wait	Underrun Action
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W		Description
7:6	Transmitter Mode	R/W		This field selects the main operating mode of the transmitter. In TDM mode, if software changes this field from 00 to non-zero in the middle of the programmed time slot, transmission does not begin until the next frame.
			00	Transmitter disabled. The TxD output is held at a High level and the transmitter is reset. The transmitter FIFO is not affected, and can be filled prior to enabling the transmitter.
			01	TRANSPARENT Mode: The transmitter does not provide any HDLC framing and formatting functions. It sends the eight bits of each byte, in the order specified by RIR[5], on the TxD output.
			10	HDLC: no encoding
			11	HDLC: NRZI encoding. Each 0-bit out of the transmitter makes the NRZI encoder invert the data from the previous bit; 1-bits do not change the data.

Bit(s)	Function	R/W		Description
5:3	Transmitter Configuration[1]	R/W	000	In this mode, Tx clocking is taken from the internal TDM module GCI/SCIT and Tx data is presented to that module. If the Transmit TDM Length field (see "Counter Access Port (CAP0, 1, 2)" on page 306) is zero, as after Reset, there is no Tx clocking. For HDLC channel 2, if this field is 000, the Receiver Configuration field must also be 000.
			001	The same as 000, except that Tx Data is driven out on the TxD pin for this.channel, and a Tx bit clock is driven onto the TxC pin, for debugging/monitoring. Do not program 001 for HDLC channel 2.
			010	Transmit data is driven onto the TxD pin for this channel (only within the time slot if the Transmit TDM Length field is non-zero, full time if the field is set to zero), and transmit clocking comes from the RxC/BCL pin for this channel. If the Transmit TDM Length field is non-zero, the TxC/FSC pin must carry Frame Sync pulses, otherwise TxC/FSC is nether driven nor used. This is the setting for separately-pinned TDM operation for this channel.
			011	Transmit data is driven onto the TxD pin for this channel, and transmit clocking comes from the TXC/FSC pin for this channel. Do not select this option when the Transmit TDM Length field is non-zero, because this pin must be Frame Sync for TDM operation.
			100	Transmit data is driven onto the TxD pin for this channel, and transmit clocking comes from the baud rate generator (BRG) associated with this Transmitter. The TxC/FSC pin is not driven or used. The BRG for this channel operates when this choice is selected.

Bit(s)	Function	R/W		Description
5:3	Transmitter Configuration[2] (Cont.)	R/W	101	Same as 100, except that the BRG clock is driven onto the TxC pin.
			11x	Reserved. Do not program this value.
2	DMA Request Half/One	R/W		If this bit is `1`, the transmitter requests a DMA transfer when there are four empty character locations in the transmit FIFO. If this bit is zero, the transmitter requests a DMA transfer whenever there are any empty locations in the transmit FIFO. When a transmitter requests a DMA transfer, it maintains that request until the transmit FIFO is full, or until the DMA channel asserts the `End of Buffer` signal that marks the last character of an HDLC frame. In the latter case, the transmitter does not assert its DMA request again until the transmit FIFO is empty.

Bit(s)	Function	R/W	Description
1	Underrun Wait	R/W	In HDLC modes, setting this bit to 1 makes the transmitter delay starting to send a frame, until either – The DMA channel has filled the entry transmit FIFO, or – It has written the last byte of a frame to the transmit FIFO. This choice provides maximum protection against transmitter underrun conditions, and should be selected when the bit rate of the channel (or the aggregate bit rate of all channels) is high enough that underruns could occur. When this bit is 0, the transmitter starts sending a frame as soon as the DMA channel provides the first byte of the frame. This choice provides slightly higher throughput, and should be selected when the data rate is such that transmitter underruns cannot occur. In Transparent mode, if Underrun Wait is 0, the transmitter resumes sending data, after an underrun, at the next byte boundary after the DMA channel provides more data. In Transparent mode and in HDLC modes, if Underrun Wait is 1, the transmitter waits after an underrun for software to write a Clear Underrun command to the Transmitter Interrupt register, before taking more data from the transmit FIFO.

Bit(s)	Function	R/W	Description
0	Underrun Action	R/W	In HDLC modes, setting this bit to 1 makes the transmitter send an Abort if an underrun condition occurs. This should be selected when the Link Layer protocol requires properly-formatted frames. Setting this bit to 0 makes the Transmitter send a CRC (if Transmitter CRCs are enabled) and a closing Flag in case of a transmitter underrun. This should be selected if HDLC framing is used only to convey a higher-layer data stream, independent of frame boundaries: In Transparent mode, a 0 in Underrun Action makes the transmitter repeat the last character in case of an underrun. If Underrun Action is 1, the transmitter sends the character in the Transmit Fill register in case of an underrun.

NOTES:

1. For channels 0 and 1, if the channel's pins are assigned to the corresponding ASCI, the HDLC channel can be used on the GCI/SCIT bus if the Transmitter Configuration is '000'. Otherwise, select '010 to '101' for this field, to correspond to the clocking selected in the ASCI and to control whether the ASCI's CKA clock is driven onto the pin.
2. For channels 0 and 1, if the channel's pins are assigned to the corresponding ASCI, the HDLC channel can be used on the GCI/SCIT bus if the Transmitter Configuration is '000'. Otherwise, select '010 to '101' for this field, to correspond to the clocking selected in the ASCI and to control whether the ASCI's CKA clock is driven onto the pin.

Transmit Control/Status Register (TCSR0, 1, 2)

Transmit Control/status Register **Ch. 0: 0062h, Ch. 1: 006Ah, Ch. 2: 0072h**

BIT	7	6	5	4	3	2	1	0
R	Minimum Preframe		Idle Select		TX CRC32	Transmitter State		
W						No Function		
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W		Description	
7:6	Minimum Preframe	R/W		In HDLC modes, this field controls the minimum number of bits sent between frames, and the minimum number of bits sent after the transmitter is enabled, before the first-data character of a frame is sent. The last two choices can be useful when sending to a remote station that can't handle back-to-back frames, for *congestion management*, or to sync up a remote (D)PLL	
				Before First Frame	Between Frames
			00	Opening Flag	Single combined Flag
			01	Opening Flag	Separate closing and opening Flags
			10	16 Idle bits, Flag	Flag, 16 Idle bits, Flag
			11	32 Idle bits, Flag	Flag, 32 Idle bits, Flag

Bit(s)	Function	R/W		Description
5:4	Idle Select	R/W		In HDLC modes, this field selects what the transmitter sends between frames. The “Mark” choice is not subject to NRZI encoding; If NRZI encoding is in use, the first bit of the Idle condition might be presented to the remote receiver as a zero, but this generally does not cause problems. The contents of the Transmit Fill register are not subjected to bit-stuffing, but are encoded if NRZI encoding is selected.
			00	Mark (the Reset state)
			01	The contents of the Fill register
			10	Disjoint Flags. (0111111001111110...)
			11	Shared-zero Flags (011111101111110...)
3	Transmit CRC32	R/W		If this bit is 1 in HDLC mode, and CRC generation is enabled, the transmitter uses the CRG32 polynomial, which provides better error detection on unreliable media. If this bit is 0 in HDLC mode with CRC generation enabled, the transmitter uses the 16-bit CRG-CCITT polynomial.

Bit(s)	Function	R/W		Description
2:0	Transmitter State	R		This read-only field provides the current real-time state of the transmitter. Except when the BRG is used for transmitter clocking this field can change asynchronously to BUSCLK. Software should read the register twice and check for the same value to assure its integrity. Although underrun is reported in the TIR rather than in this field, it is a major part of the state of the transmitter. In all cases except Transparent mode with the Underrun Wait bit = 0, software must release the transmitter from an underrun before it changes state and sends more data. This is accomplished by writing a command to the TIR, or by disabling and then re-enabling the transmitter.
			000	Sending opening Flag, or Transparent Mode
			001	Sending data
			010	Sending CRC
			011	Sending closing Flag
			100	Sending first Idle or Abort
			101	Sending second Idle
			110	Sending third or subsequent Idle
			111	Sending Abort due to underrun not assert its DMA request again until the transmit FIFO is empty.

Transmit Interrupt Register (TIR0, 1, 2)

Transmit Interrupt Register **Ch. 0: 0061h, Ch. 1: 0069h, Ch. 2: 0071h**

<table>
<tr><td>BIT</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>R</td><td rowspan="2">Underrun IE</td><td rowspan="2">Two Idle Sent IE</td><td rowspan="2">One Idle Sent IE</td><td rowspan="2">EOF Sent IE</td><td>Reserved</td><td>Underrun</td><td>HDLC IUS</td><td>TX IP</td></tr>
<tr><td>W</td><td colspan="4">Transmitter Commad</td></tr>
<tr><td>DEF</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

Bit(s)	Function	R/W	Description
7	Underrun Interrupt Enable	R/W	A 1 in this bit makes this transmitter set its IP bit, and requests an interrupt, if an underrun condition occurs.
6	Two Idles Sent Interrupt Enable	R/W	A 1 in this bit makes the transmitter set its IP bit,-and requests an interrupt, each time it has sent two Idle Flags or fifteen Mark bits. This interrupt can occur even if the transmitter is disabled.
5	One Idle Sent Interrupt Enable	R/W	A 1 in this bit makes the transmitter set its IP bit, and requests an interrupt, each time it has sent one Idle Flag or seven Mark bits. This interrupt can occur even if the transmitter is disabled.
4	End Of Frame Sent Interrupt Enable	R/W	A 1 in this bit makes the transmitter set its IP bit, and requests an interrupt, when it has sent the closing Flag of a frame. Such an interrupt can be more useful than an `End of Buffer` interrupt from the associated DMA channel, in that this interrupt indicates that the frame has been completely sent.
3	Reserved		Reserved. Must be 0.

Bit(s)	Function	R/W	Description
2	Underrun	R	This bit is set if the transmitter encounters an underrun condition, and is latched until software writes one of the two `Clear Underrun` commands to this register (bits [2:0]). For Transparent mode with the Underrun Wait bit equal to 0, this bit provides a latched record of the underrun.
1	HDLC Interrupt Under Service	R	There is only one Interrupt Under Service bit for all the HDLC transmitters and receivers, but it can be read and cleared in any of their respective Interrupt registers. This bit is set when an interrupt acknowledge cycle returns a vector for an HDLC transmitter or receiver. It is cleared by Reset or when software writes a `Clear IUS` or `Clear IP and IUS` command to any HDLC Transmitter or Receiver Interrupt Register.
0	Transmitter Interrupt Pending	R	This bit is set when one of the conditions enabled in bits [7:4] occurs, or when software writes a Set IP command to this register. This-bit is cleared by Reset or when software writes a `Clear IP` or `Clear IP and IUS` command to this register.

Bit(s)	Function	R/W	Description	
2:0	Transmitter Command	W		Software can set or clear certain register bits for this transmitter in one case, and clear the transmitter FIFO, by writing one of the following codes to this write-only field.
			000	No operation
			001	Clear IP. Use this command near the start of an HDLC Transmit Interrupt Service Routine (ISR) that enables *nested* interrupts from higher-priority devices. Clearing IP before reading status helps ensure that no interrupt events are lost.
			010	Clear IUS. Use this command near the end of an ISR that enables *nested* interrupts from higher-priority devices.
			011	Clear IP and IUS. Use this command near the start of an ISR that does not enable *nested* interrupts.
			100	Reserved. Do not write this value.
			101	Set IP. This command causes an interrupt from this transmitter.
			110	Clear Underrun. This command clears the Underrun bit in this register in any mode. In HDLC modes, and in Transparent mode with the Underrun Wait bit equal to 1, it also releases the transmitter for further transmission.
			111	Clear Underrun and Tx FIFO. This is similar to the preceding command, except that it also empties the transmitter FIFO of data. Underrun Wait bit equal to 1, it also releases the transmitter for further transmission.

Transmit Fill Register (TFR0, 1, 2)

Transmit Fill Register **Ch. 0: 0063h, Ch. 1: 006Bh, Ch. 2: 0073h**

BIT	7	6	5	4	3	2	1	0
R W	Transmit Fill Character							
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W	Description
7:0	Transmit Fill Character	R/W	The character stored in the Transmit Fill register can be sent between frames in HDLC mode, or in case of an underrun in TRANSPARENT mode. When transmitted, this character is not subject to bit-stuffing, but is encoded it operating in NRZI mode.

Receive Mode Register (RMR0, 1, 2)

Receive Mode Register **Ch. 0: 0064h, Ch. 1: 006Ch, Ch. 2: 0074h**

BIT	7	6	5	4	3	2	1	0
R	Receive Mode		Receive Configuration			DMA Request Half/One	RX CRC32[1]	Single Inter-frame Flag[1]
W							Transparent Start[2]	
DEF	0	0	0	0	0	0	0	0

NOTES:

1. Applies in HDLC Modes
2. Applies in Transparent Mode

Bit(s)	Function	R/W		Description
7:6	Receiver Mode	R/W		This field selects the main operating mode of the receiver. In a TDM mode, if software changes this field from 00 to non-zero in the middle of the programmed time slot, reception does not start until the next frame.
			00	Receiver Disabled: While this value is in effect, the receiver is reset, but the receive FIFO is not affected. When this value is changed from 00 to 10 or 11, the receiver starts by searching for an opening Flag (Hunt mode).
			01	Transparent Mode: the receiver does not detect or remove any HDLC framing and formatting. It assembles each eight bits received into a byte, in the order specified in RIR[5]. When this mode is first set, the receiver searches for the start condition selected by the Transparent Start field.
			10	HDLC, no decoding.
			11	HDLC, NRZI decoding: Each bit that is different from its predecessor is presented to the receiver as a 0, while each bit that is the same is presented as a 1.

Bit(s)	Function	R/W	Description	
5:3	Receiver Configuration	R/W		This field selects the I/O configuration of the Receiver. See Table 29 and Table 30 for pin usage information.
			000	In this mode, Receive data and clocking come from the internal TDM module (GCI/SCIT). If the Receive TDM Length field is zero, as after Reset, there is no Receiver clocking. For HDLC channel 2, if this field is 000, the Tx Configuration field must also be 000.
			001	The same as 000, except that receive data and clocking are driven onto the RxD and RxC pins for this channel. Do not program this value for HDLC channel 2.
			010	Receive data comes from the RxD pin, and receive clocking comes from this channel's RxC/BCL pin. If the Receive TDM Length field is non-zero, this channel's TxC/FSC pin must carry Frame Sync pulses, otherwise TxC/FSC is neither driven nor used. This is the setting for separately-pinned TDM operating mode for this channel.
			011	Receive data comes from the RxD pin; receive clocking comes from the TxC/FSC pin. Do not select this option with any TDM mode.

Bit(s)	Function	R/W	Description
			100 Receive data comes from the RxD pin; receive clocking comes from the DPLL associated with this receiver. When this choice is selected, the DPLL operates.
			101 As 100, except that the DPLL clock is driven onto the RxC pin.
			110 Internal loopback. Receive data comes from the associated transmitter, as does clocking.
			111 Internal loopback with DPLL. Receive data comes from the associated transmitter, receive clocking comes from the DPLL. This condition allows testing with the DPLL in LOOPBACK mode.
2	DMA Request Half/One	R/W	If this bit is 1, the receiver requests DMA transfers when there are four characters in the receiver FIFO, or when a character flagged as the last one of a frame has been written into the receiver FIFO. If this bit is zero, the receiver requests DMA transfers whenever there is (are) any character(s) in the receiver FIFO. In either case, the request is maintained until the receiver FIFO is empty.
1	Receive CRC32	R/W	If this bit is 1 in an HDLC mode, the receiver uses the CRC32 polynomial in CRC (HDLC modes only) checking, which is reflected in bit 3 of "Completed Buffer (with Status)" values in DMA status byte/types. CRC32 provides better error detection on unreliable media such as wireless. If this bit is 0 in HDLC mode, the receiver uses the 16-bit CRC- CCITT polynomial.

Bit(s)	Function	R/W	Description
0	Single Interframe Flag	R/W	This bit is 1 in an HDLC mode, the receiver accepts a single Flag as a valid (HDLC modes only) boundary between successive frames. If this bit is 0 in an HDLC mode, after detecting a closing Flag, the receiver automatically enters “Hunt mode” for a subsequent opening Flag, ignoring whatever data may lie between the two Flags. Note that the receiver always enters Hunt mode when it is first enabled, and after detecting an Aborted frame. Software can also force the receiver into Hunt mode at any time, by writing a command to the Receive Interrupt Register.
1:0	Transparent Start (Transparent mode only)	R/W	This field determines when the receiver starts assembling data when the Receive Mode field is switched from 00 to 01.
			00 Immediately in non-TDM modes, at the start of the next time slot in TDM modes.
			01 At the next transition on RxD (next transition in the time slot in TDM modes).
			10 At the next Space/0 on RxD (next zero in a time slot in TDM modes).
			11 At the next Mark/1 on RxD (next one in a time slot in TDM modes).

Receive Interrupt Register (RIR0, 1, 2)

Receive Interrupt Register **Ch. 0: 0065h, Ch. 1: 006Dh, Ch. 2: 0075h**

BIT	7	6	5	4	3	2	1	0
R	Idle IE	Rx Clock Polarity	MSBIT First	Reserved		Idle	HDLC Lus	Rx IP
W						Receiver Command		
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W	Description
7	Idle Interrupt Enable	R/W	A 1 in this bit causes this Receiver to set its IP bit, and requests an interrupt, when the Idle bit goes from 0 to 1.
6	Receive Clock Polarity	R/W	A 0 in this bit causes the Receiver to sample RxD on the falling edge of the selected receiver clock; a 1 selects the rising edge.
5	Most Significant Bit First	R/W	If this bit is a 1, the channel's transmitter sends the MS bit of each character first, and the channel's receiver places the earliest-received bit in the MS bit of each character. If this bit is a 0, Tx and Rx are *LS bit first*. The former is *telecomm bit order* while the latter is *datacomm bit order*. If this bit is a 1 (0), partial Tx characters must be MS (LS) bit justified, while partial Rx characters are stored in the LS (MS) bits of the last character of a frame.
4:3	Reserved		Reserved. Must be 0.

Bit(s)	Function	R/W	Description
2:0	Receiver Command	W	Software can set or clear various register bits, or change the state of the receiver, by writing a code to this write-only field: 000 No operation 001 Clear IP. Use this command near the start of an HDLC Interrupt Service Routine (ISR) that enables *nested* interrupts from higher-priority devices. Clearing IP before reading status helps ensure that no interrupt events are lost. 010 Clear IUS. Use this command near the end of an ISR that enables nested interrupts from higher-priority devices. 011 Clear IP and IUS. Use this command near the start of an ISR that does not enable nested interrupts. 100 Reserved. Do not write this value. 101 Set IP. This command causes an interrupt from this receiver. 110 Enter HUNT Mode; This command makes the receiver immediately begin searching for an opening Flag, regardless of its previous state. The receiver automatically enters HUNT mode when it is first enabled, when a frame ends with an Abort, and when it receives a closing Flag if the Single Interframe Flag bit is 0. 111 Enter Hunt/Clear Rx FIFO; This is similar to the previous command, except that it also empties the Rx FIFO of any previously received data.
2	Idle	R	This bit is 1 whenever the last 15 bits received by this receiver were all ones. It is a real-time bit, without any latching.

Bit(s)	Function	R/W	Description
1	HDLC Interrupt Under Service	R	There is only one Interrupt Under Service bit for all the HDLC transmitters and receivers, but it can be read and cleared in any of their Interrupt registers. This bit is set when an interrupt acknowledge cycle returns a vector for an HDLC transmitter or receiver. It is cleared by Reset, or when software writes a `Clear IUS` or `Clear IP and IUS` command to any HDLC Interrupt register.
0	Receiver Interrupt Pending	R	This bit is set when Idle Interrupt Enable is 1 and the Idle bit goes from 0 to 1, or when software writes a `Set  IP` command to this register. This bit is cleared by Reset, or when software writes a `Clear IP` or `Clear IP and IUS` command to this register. transmitter or receiver. It is cleared by Reset, or when software writes a `Clear IUS` or `Clear IP and IUS` command to any HDLC Interrupt register.

Counter Access Port (CAP0, 1, 2)

Counter Access Port **Ch. 0: 0066h, Ch. 1: 006Eh, Ch. 2: 0076h**

BIT	7	6	5	4	3	2	1	0
R W	Counter Access Port (BRG Time Constant Low, BRG Time Constant High) (Transmit TDM Start, Transmit TDM Length) (Receive TDM Start, Receive TDM Length)							
DEF	0	0	0	0	0	0	0	0

This register allows the processor to write and read the starting values for various counters in the HDLC channel as follows:

- After software writes the Transmit Mode Register with a Tx Configuration value of `1xx`, the next access to the CAP reads or writes the LS byte of the Baud Rate Generator Time Constant, and the next access reads or writes the MS byte of the time Constant.
- After software writes the Receive Mode Register, with a Tx Configuration value of `0xx`, or after Reset, the next access to the CAP reads or writes the Transmit TDM Start value, and the next access reads or writes the Transmit TDM Length value.
- After software writes the Transmit Mode Register the next access to the CAP reads or writes the Receive TDM Start value, and the next access reads or writes the Receive TDM Length value.

All registers accessed via this port default to a value of `00H` after a Reset.

Bit(s)	Function	R/W	Description
7:0	BRG Time Constant Low	R/W	The least significant eight bits of the BRG time constant.
7:0	BRG Time Constant High	R/W	The most significant eight bits of the BRG time constant.
7:0	Transmit TDM Start	R/W	The number of BCL bit times to wait after a rising edge on FSC before starting to output transmitter data.
7:0	Transmit TDM Length	R/W	The number of bits of transmitter data to output during the transmitter's time slot.
7:0	Receive TDM Start	R/W	The number of BCL bit times to wait after a rising edge on FSC before starting to sample the received data.
7:0	Receive TDM Length	R/W	The number of bits of received data to sample during the receivers time slot. sample the received data.

DMA Select Register (DSR0, 1, 2)

DMA Select Register **Ch. 0: 0067h, Ch. 1: 006Fh, Ch. 2: 0077h**

BIT	7	6	5	4	3	2	1	0
R W	TxDMA Enable	TxDMA Channel			RxDMA Enable	RxDMA Channel		
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W	Description
7	TxDMA Enable	R/W	If this bit is set to 1 DMA operation is enabled for the channel's transmitter, using the DMA channel identified in bits [6:4] of this register. This process requests the DMA controller to load characters into the transmit FIFO whenever it is empty to the extent defined by the transmitter's DMA Request Half/One bit, TMR[2].
6:4	TxDMA Channel	R/W	Identifies the DMA channel to be used for the transmit DMA process when bit [7] is set to 1.
3	RxDMA Enable	R/W	If this bit is set to 1, DMA operation is enabled for the channel's receiver, using the DMA channel identified in bits [2:0] of this register. This process requests the DMA controller to empty characters from the receive FIFO whenever it is filled to the extent defined by the receivers DMA Request Half/One bit, RMR[2].
2:0	RxDMA Channel	R/W	Identifies the DMA channel to be used for the receive DMA process when bit [3] is set to 1 to the extent defined by the receivers DMA Request Half/One bit, RMR[2].

Global HDLC Vector Register (HDLCV)

Global HDLC Vector Register **003Dh**

BIT	7	6	5	4	3	2	1	0
R	Vector Base				Device			No IP
W					Reserved			
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W	Description
7:4	Vector Base	R/W	The base interrupt vector for the HDLC channels.
3:0	Reserved	W	Reserved. Write as zeroes.
3:1	Device	R	When this register is read with a normal read cycle, If the No IP bit is 0, this field identifies the highest priority HDLC device that has is IP bit set: \000 Transmitter 0 (Highest priority) 001 Receiver 0 010 Transmitter 1 011 Receiver 1 100 Transmitter 2 101 Receiver 2 (Lowest priority) 11x These values do not occur.

Bit(s)	Function	R/W	Description
0	No IP	R	When this register is read with a normal read cycle, it is 0 if any of the HDLC Transmitters or Receivers has its IP bit set, or 1 if none of them have IP set. The HDLC channels have no provision for 380 Mode 0 interrupts, nor for *status does not affect vector*.
7:0	Interrupt Vector	—	This vector is automatically output on the D[7:0] output pins in response to an interrupt acknowledge cycle, if any one of the HDLC devices has its IP bit set and is the highest priority interrupting device currently requesting an interrupt. The contents of the vector are as follows: D[7:4] Base interrupt vector bits D[3:1] A field which identifies the highest priority HDLC device that has is IP bit set, as per 'Device' above. D[0] Zero

GCI/SCIT INTERFACE

Figure 19 illustrates the frame structure of the GCI/SCIT interface.

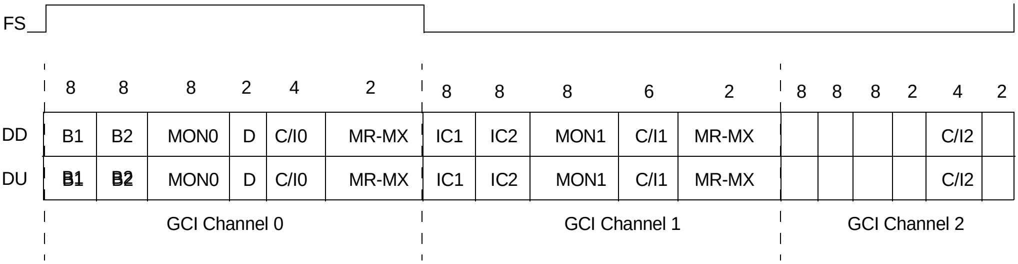


Figure 19. GCI/SCIT Frame Structure (TERMINAL Mode)

Overview

The GCI/SCIT bus (also commonly called the IOM-2™ bus)[1] provides a standardized full-duplex communications facility containing user data, control/programming and status channels. The standard defines two modes: the LINECARD mode and the TERMINAL mode. Both modes utilize the same basic frame arid clocking structure but differ in the number and usage of individual channels. The Z80382 GCI/SCIT interface supports the TERMINAL mode only.

The various channels are time-multiplexed over a four-wire serial interface:

- Data is clocked by a data clock, DCL, that operates at twice the data rate.

[1] IOM-2 is a trademark of Siemens AG.

- Frames are delimited by an 8KHz frame synchronization clock, FSC.
- Data is carried over data upstream and data downstream signals, DU and DD.

TERMINAL Mode Frame Structure

The TERMINAL mode frame is divided into three sub-frames called channels 0, 1, and 2, each containing 32 bits. This 12-byte frame is repeated at a rate of 8KHz, which gives an aggregate data rate of 768Kbits/second.

The first sub-frame is dedicated to controlling the layer 1 transceiver (Monitor and C/I channels) and passing user data (B and D channels) to the Layer 1 transceiver (or between Layer 1 transceivers). The second and third subframes are used for communicating between a controlling device and devices other than the Layer 1 transceiver (Monitor and C/I channels, or between two user data processing devices (IC channels). The C/I channel of the third sub-frame is used for TIC bus applications. The TIC bus is described below.

B Channels

B1 and B2 are the first two 8-bit time slots after the frame sync pulse. Each B channel provides 64Kbps of user data to/from the network.

Monitor Channels

There are two programming channels, Monitor 0 and Monitor 1. Each channel consists of eight bits of data and two associated handshake bits, MR and MX, that control data flow.

D Channel

The 16Kbps D channel (2 bits per frame) provides a connection between the Layer two and Layer one components.

Command/Indicate Channels

Three command/indicate channels, C/I0, C/I1 and C/I2 are provided. Each sub-frame uses one. (C/I2 is the same as TIC, see below.) These channels provide real-time status between devices connected via the (GCI/SCIT). bus.

Intercommunication Channels

Two intercommunication subchannels are provided in GCI channel 1. These provide 64Kbps data paths between user devices.

TIC Bus

This channel is the same as C/I2 and is used to control D channel access with some (GCI/SCIT) devices. It allows multiple Layer 2 devices to individually gain access to the D and C/I channels located in the first subframe.

Data Signals

The data signals on the GCI/SCIT bus are called Data Up stream (DU) and Data Downstream (DD). While each of these is a bus that can be sensed as well as driven in an open-drain (open-collector) fashion by the Z80382 and other devices, GCI practice defines certain fields on each line to flow in certain directions.

The Z80382 always receives from DD and (when enabled) drives DU in the B2, MON0, D, C/IO, MR0 and MX0 fields. Which line is driven and which is received can be selected by software for the IC1, IC2, MON1, and C/I1 fields and the MX1 and MR1 bits.

Monitor Channel Operation

The monitor channels are full-duplex and operate on a pseudo-asynchronous basis, that is, data transfers take place synchronized to frame sync but the flow is controlled by a handshake procedure using the MX and MR bits. The handshake procedure (flow of events) is depicted in Figure 20.

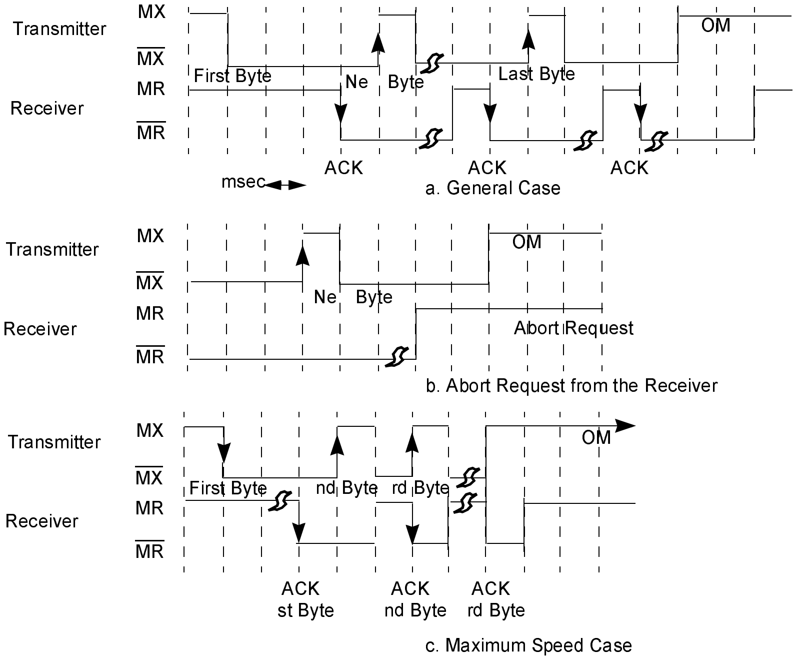


Figure 20. Monitor Handshake Timing

Idle

The MX and MR pair being held inactive (high) for two or more frames constitutes the channel being idle in that direction. The data received in the monitor channel is invalid and should be `11111111`.

Start of Transmission

The first byte of data is placed on the bus and MX is activated (low). MX remains active, and the data is repeated until an inactive-to-active transition of MR is received, indicating that the data has been captured by the receiver.

Subsequent Transmissions

The second and subsequent bytes are placed on the bus after the inactive to active transmission of MR. At the time that the second byte is transmitted, MX is returned inactive for one frame time only; the data is valid in the same frame. In the following frame, MX returns active again and the same byte is transmitted. Data is repeated in subsequent frames and MX remains active until acknowledgment is detected (MR transition from inactive to active).

Maximum Speed Case

The transmitter is capable of minimizing the delay between bytes to achieve higher data throughput than is provided by the general case described above. The first and second bytes are transmitted normally, However, starting with the third byte, the transmitter deactivates MX and transmits new data one frame time after MR is deactivated. In this way, the transmitter anticipates that MR is reactivated, which it accomplishes one frame time after it is deactivated, unless an abort is signalled by the receiver.

End of Message

The transmitter sends an EOM, normally after the last byte of data has been transmitted, by not reactivating MX after deactivating it in response to MR going inactive.

Reception

At the time the receiver sees the first byte, indicated by the inactive-to-active transition of MX, MR is by definition inactive. In response to the activation of MX, the data is read off the bus and MR is activated. MR remains active until the next byte is received or an end of message is detected. Subsequent data is received from the bus on each falling edge of MX, and a monitor channel receive data available interrupt is generated. Note that the data may actually be valid at the time that MX went inactive, one frame time prior to going active. MR is deactivated after the data is read and reactivated one frame time later. The transmitter detects MR going inactive and anticipates its reactivation one frame later. The reception of data is terminated by the reception of an end of message indication.

Abort

The abort is a signal from the receiver to the transmitter indicating that the data has been missed. It is not an abort in the classical sense, which is an indication that the current message should be ignored. The receiver indicates an abort by holding MR inactive for two or more frames in response to MX going inactive.

Flow Control

The receiver can hold off the transmitter by keeping MR active until the receiver is ready for the next byte. The transmitter does not start the next transmission cycle until MR goes inactive.

Monitor Channel Handling

Before transmitting data on a monitor channel, the processor should look at the Monitor 0 or 1 active status bit in GCI Status Register 2 to verify that the channel is inactive. The processor can then write the data to the Monitor Transmit Data Register. This procedure enables the GCI hardware to proceed with the transmission of this data according to the monitor channel protocol. On receiving an acknowledge from the receiver, the transmit data request bit in GCI Status Register 1 is set, indicating that the monitor channel is ready to transmit another byte of data. When the last byte has been acknowledged by the receiver, the processor can set the EOM request bit in the GCI Control Register and the monitor channel then sends an `end of message` signal.

On receiving the monitor data, the receiver writes this data to the monitor receive register and sets the appropriate status bit. This procedure generates a monitor receive data available interrupt, instructing the processor to read this data.

Succeeding bytes of data are received in accordance with the monitor channel protocol, and the processor is informed by means of the monitor receive data available interrupt. The processor forces the receiver to ask for an abort by setting the abort request bit. The receiver asks for an abort in transmission by sending an inactive MR for two consecutive frames. The abort transmission is indicated in the status bit by the transmitter.

C/I Channel Operation

Data on C/I0 and C/I1 is transmitted continuously in each frame until new data is to be sent. A change in C/I channel data is considered valid if the same value has been received in two consecutive frames.

Bus Activation and Deactivation

Deactivation, Upstream to Downstream

The upstream (clock master) unit can initiate deactivation by issuing a series of software handshakes via the C/I0 channel. Having done so, the upstream unit waits for a deactivation indication from all downstream (clock slave) units. Once this is received, a deactivation confirmation is issued, followed by stopping the clocks (forcing them low) and placing the data pin in a high impedance state. After the clocks are stopped, the input pin is monitored for the presence of a timing request from the downstream unit (the pin being pulled low).

Deactivation Request, Downstream to Upstream

Deactivation is normally initiated by the upstream device as described above. When the downstream device re carves the deactivation request over the C/I0 channel, it must respond by sending the deactivation indication.

Activation Request, Downstream to Upstream

The downstream device can request that the clocks be stated by pulling its data output line low. Once the clocks are started, the downstream unit requests activation by sending an activation request over the C/I0 channel.

Activation, Upstream to Downstream

The upstream unit activates the bus by starting the clocks and following the C/I0 channel-based activation procedure.

B1, B2, D, IC1, IC2 Channel Data

For these fields, Rx data and the bit clock are supplied to the HDLC cells, and Tx data is taken from the HDLC cells. Each HDLC transmitter and receiver includes a Time Slot Assigner which can be programmed for any of the subchannels shown above.

GCI/SCIT Registers

The GCI/SCIT module includes the following I/O-mapped registers that can be read and written by the 380C processor:

- GCI Control Register
- GCI Status Register 1
- GCI Status Register 2
- GCI Interrupt Enable Register
- Monitor 0 Transmit/Receive Data Register
- Monitor 1 Transmit/Receive Data Register
- C/I0-C/I2 Transmit/Receive Data Register
- C/I1 Transmit/Receive Data Register

Details on each register are provided in the sections which follow. In the descriptions:

- `HHHHh` = registers address in the 380C I/O space, in hexadecimal
- DEF = default value after RESET
- x = indicates that the default value of the bit is indeterminate

GCI Control Register (GCICR)

GCI CONTROL REGISTER **00C0h**

BIT	7	6	5	4	3	2	1	0
R W	Monitor 1 And C/I1 Direction	Clock Activa-tion Request To Master	Monitor 1 EOM Request	Monitor 1 Abort Request	Monitor 1 Enable	Monitor 0 EOM Request	Monitor 0 Abort Request	Monitor 0 Enable
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W	Description
7	Monitor 1 and C/I1 Direction	R/W	When this bit is 0, the (GCI/SCIT) interface sends on DU and receives on DD in the C/I1 and MON1 fields and the MX1 and MR1 bits. When this bit is 1, the interface sends on DD and receives on DU in these fields and bits.
6	Clock Activation Request to Master	R/W	When this bit is set to 1, the interface initiates the clock activation procedure.
5	Monitor 1 EOM Request.	R/W	When this bit is set to 1, the transmitter sends an EOM on Monitor channel 1 if the Monitor 1 Transmit Data Register is empty.
4	Monitor 1 Abort Request	R/W	When this bit is set to 1, the transmitter sends an abort on Monitor channel 1.
3	Monitor 1 Enable	R/W	Setting this bit to 1 enables Monitor channel 1.

Bit(s)	Function	R/W	Description
2	Monitor 0 EOM Request	R/W	When this bit is set to 1, the transmitter sends an EOM on Monitor channel 0 if the Monitor 0 Transmit Data Register is empty.
1	Monitor 0 Abort Request	R/W	When this bit is set to 1, the transmitter sends an abort on Monitor channel 0.
0	Monitor 0 Enable	R/W	Setting this bit to 1 enables Monitor channel 0.

Monitor 0 Transmit/Receive Data Register (MON0)

Monitor 0 Transmit/Receive Data Register **00C4h**

Bit	7	6	5	4	3	2	1	0
R	Monitor 0 Receive Data							
W	Monitor 0 Transmit Data							
Def	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W	Description
7:0	Monitor 0 Transmit Data	W	Data written into this register, when bit 1 of GCI Status Register 1 is 1, is transmitted on Monitor channel 0 following the monitor protocol.
7:0	Monitor 0 Receive Data	R	Data can be read from this register when bit 0 of GCI Status Register 1 is a 1, for example, in response to a Monitor 0 receive data available interrupt. Reading this register clears GCISR1[0]. There is no danger of overrun on the monitor channels because the hardware can not complete the handshake for further data until this register has been read.

Monitor 1 Transmit/Receive Data Register (MON1)

Monitor 1 Transmit/Receive Data Register **00C5h**

BIT	7	6	5	4	3	2	1	0
R	Monitor 1 Receive Data							
W	Monitor 1 Transmit Data							
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W	Description
7:0	Monitor 1 Transmit Data	W	Data written into this register, when bit 5 of GCI Status Register 1 is 1, is transmitted on Monitor channel 1 following the monitor protocol.
7:0	Monitor 1 Receive Data	R	Data can be read from this register when bit 4 of GCI Status Register 1 is a 1, for example, in response to a Monitor 1 receive data available interrupt. Reading this register clears GCISR1[4]. There is no danger of “overrun” on the monitor channels because the hardware can not complete the handshake for further data until this register has been read.

C/I0/2 Transmit/Receive Data Register (CI02)

C/I0/2 Transmit/Receive Data Register **00C6h**

BIT	7	6	5	4	3	2	1	0
R	CI/O-C/I2 Receive Data							
W	CI/O-C/I2 Transmit Data							
DEF	0	0	0	0	1	1	1	1

Bit(s)	Function	R/W	Description
7:0	C/I0/2 Transmit Data	W	When the 380C writes to this register, the value of bit 7 determines whether the LS bits are sent on DU in the C/I0 or C/I2 time slot. If bit 7 is a 0, bits [3:0] are transmitted continuously on DU in the C/I0 channel, except that four ones mean the device does no driving in the C/I0 field. If bit 7 is 1 and bit 3 is 0, the hardware sets the DU C/I2 Status field to is *Waiting* value. Next, it checks the first bit of the DU C/I2 field, which is called BAC or Bus Access Control. If BAC is low (0), the hardware waits for BAC to become asserted in a future frame. Then, or if BAC is high (1) immediately, the hardware attempts to send bits [2:0] as a *TAD* value in the next three bit times. If in any bit of these three the hardware is given a 1 to send but senses a 0 on DU, it discontinues sending any further bits and waits to try again, whenever BAC is 1 in a future frame. If it sends all three bits without encountering such a conflict, the hardware: 1. Sets the DU CI2 status field in GCISR2 to *In Control,* 2. Requests an interrupt if the C/I2 interrupt enable bit is 1, and 3. Sends a 0 on DU in the BAC bit of subsequent frames while the DU CI2 Status remains *In Control*.. Software writes ones to both bits 7 and 3 of this register to change the DU CI2 Status to *Busy, not Waiting*.
7:0	C/I0/2 Receive Data	R	Whenever data on the DD C/I0 channel changes to the same value for two consecutive frames; the hardware captures it in bits [3:0] of this register. Bit 7 of this register captures the first (*stop/go*) bit of the DD C/I2 field, using the same rules.

C/I1 Transmit/Receive Data Register (CI1)

C/I1 Transmit/Receive Data Register **00C7h**

BIT	7	6	5	4	3	2	1	0
R	C/I1 Receive Data							
W	C/I1 Transmit Data							
DEF	0	0	1	1	1	1	1	1

Bit(s)	Function	R/W	Description
7:0	C/I1 Transmit Data	W	Bit 7 controls the direction of the IC1 field, and bit 6 controls the direction of the IC2 field. In each case a 0 makes the hardware receive from DD and send on DU, while a 1 means receive from DU and send on DD. Data written to bits [5:0] of this register are transmitted continuously in the C/I1 channel by the hardware, except that six ones mean the device does no driving in the C/I1 field.
5:0	C/I1 Receive Data	R	When data on the C/I1 channel changes to the same value for two consecutive frames, it is captured in the LS six bits of this register.

GCI Status Register 1 (GCISR1)

GCI Status Register 1 **00C1h**

BIT	7	6	5	4	3	2	1	0
R	Monitor 1 Abort Received	Monitor 1 EOM Received	Monitor 1 Tx Data Request	Monitor 1 Rx Data Available	Monitor 0 Abort Received	Monitor 0 EOM Received	Monitor 0 Tx Data Request	Monitor 0 Rx Data Available
W	NO FUNCTION							
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W	Description
7	Monitor 1 Abort Received	R	This bit is set when an abort is detected on Monitor channel 1. It is cleared when this register is read.
6	Monitor 1 EOM Received	R	This bit is set when an EOM is received on Monitor channel 1. It is cleared when this register is read.
5	Monitor 1 Transmit Data Request	R	This bit is 1 if Monitor channel 1 is enabled, and any data previously written to the Monitor 1 Tx Data Register has been sent and acknowledged under the Monitor protocol, and any EOM previously requested in the GCI control register has been sent. It is cleared on writing to the Monitor 1 Transmit Data Register, on writing a Monitor 1 EOM request to the GCICR, or if the Monitor 1 channel is disabled.
4	Monitor 1 Receive Data Available	R	This bit is set when Monitor 1 Receive Data Register is written to by the hardware. The bit is cleared when the Monitor 1 Receive Data Register is read.

Bit(s)	Function	R/W	Description
3	Monitor 0 Abort Received	R	This bit is set when an abort is detected on Monitor channel 0. It is cleared when this register is read.
2	Monitor 0 EOM Received	R	This bit is set when an EOM is received on Monitor channel 0. It is cleared when this register is read.
1	Monitor 0 Transmit Data Request	R	This bit is 1 if Monitor channel 0 is enabled, and any data previously written to the Monitor 0 Tx Data Register has been sent and acknowledged under the Monitor protocol, and any EOM previously requested in the GCI control register has been sent. It is cleared on writing to the Monitor 0 Transmit Data Register, on writing a Monitor 0 EOM request to the GCICR, or if the Monitor 0 channel is disabled.
0	Monitor 0 Receive Data Available	R	This bit is set when Monitor 0 Receive Data Register is written to by the hardware. The bit is cleared when the Monitor 0 Receive Data Register is read.

GCI Status Register 2 (GCISR2)

GCI STATUS REGISTER 2 **00C2h**

BIT	7	6	5	4	3	2	1	0
R	Frame Sync	DU C/I2 Status		Monitor 1 Active	Monitor 0 Active	C/I1 Data Received	C/I0 Data Received	Clock Activa-tion
W	No Function							Clear Status Bit
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W	Description
7	Frame Sync	R	This bit is set when a frame sync pulse occurs, and is cleared when this register is read. It can be enabled for interrupt via the GCI Interrupt Enable register.
6:5	DU C/I2 Status	R	This field indicates the C/I2 status on DU: 00 Idle: the last C/I2 field was 'all ones' and this unit is not in control in C/I2, nor waiting to take control. 01 Busy/not wafting: the last C/I2 field was not all ones, and this unit is not in control in C/I2, nor waiting to take control. 10 Waiting: a command to take control in C/I2 has been written to the C/I0/2 Transmit Register, and control has not yet been achieved. 11 In Control: this device has control in C/I2.
4	Monitor 1 Active	R	This bit is set by the hardware when Monitor channel 1 is active and is cleared when the channel is inactive.

Bit(s)	Function	R/W	Description
3	Monitor 0 Active	R	This bit is set by the hardware when Monitor channel 0 is active and is cleared when the channel is inactive.
2	C/I1 Data Received	R	This bit is set when C/I1 receive data has changed to a new value for two consecutive frames. It is cleared when the C/I1 Receive Data register is read
1	C/I0 Data Received	R	This bit is set when C/I0 receive data has changed to a new value for two consecutive frames. It is cleared when the C/I1 receive Data Register is read.
0	Clock Activation	R	This bit is set when clock activation has been detected by the hardware. The 0-to-1 transition of this bit can be enabled for interrupt via the GCI Interrupt Enable register, in which case reading this register clears the interrupt request, but not this status bit.
0	Clear Clock Activation Status	W	Software writes a 0 to this bit after it has received the (last repetition of the) clock deactivation message from the master on the C/I0 channel. Writing a 0 clears the read-side bit and makes the hardware search for clock reactivation. Writing a 1 to this bit has no effect.

GCI Interrupt Enable Register (GCIIE)

GCI Interrupt Enable Register (GCIIE) **GCIIE: 00C3h**

BIT	7	6	5	4	3	2	1	0
R W	C/I2 Interrupt Enable	C/I1 Receive Data Interrupt Enable	C/I0 Receive Data Interrupt Enable	Clock Act/ Frame Sync Intr Enable	Monitor 1 Tx Interrupt Enable	Monitor 1 Rx Interrupt Enable	Monitor 0 Tx Interrupt Enable	Monitor 0 Rx Enable
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W	Description
7	C/I2 Interrupt Enable[1]	R/W	A 1 enables an interrupt when the status changes from Waiting to In Control, or when the first bit of DD C/I2 (the stop/go bit) changes. The former interrupt is cleared by reading Status Register 2, the latter by reading the C/I0/2 Transmit/Receive Data register.
6	C/I1 Receive Data Interrupt Enable	R/W	A 1 enables the C/I1 Receive Data Interrupt.
5	C/I0 Receive Data Interrupt Enable	R/W	A 1 enables the C/I0 Receive Data Interrupt.
4	Clock Activation /Frame Sync Interrupt Enable	R/W	A 1 enables a clock activation interrupt when clocks are deactivated, and a Frame Sync interrupt when clocks are activated. Either of these interrupts are cleared by reading GCI Status Register 2.

Bit(s)	Function	R/W	Description
3	Monitor 1 Tx Interrupt Enable	R/W	A 1 enables the Monitor 1 Transmit Interrupt
2	Monitor 1 Rx Interrupt Enable	R/W	A 1 enables the Monitor 1 Receive Interrupt
1	Monitor 0 Tx Interrupt	R/W	A 1 enables the Monitor 0 Transmit Interrupt
0	Monitor 0 Rx Interrupt	R/W	A 1 enables the Monitor 0 Receive Interrupt

NOTES:

1. Interrupts from the GCI/SCIT module are handled through the Assigned Vector facility. Inmost cases, bits in the status registers are ANDed with the corresponding enable bits in this register, and the results are ORed to make the interrupt request to the Assigned Vector facility.

   The only control function that resembles an overall “GCI/SCIT enable” is whether or not HDLC Channel 2's Tx Configuration and Rx Configuration fields are 000, which is the Reset state. The fact that the GCI Control register is all zero after Reset keeps the interface from doing anything active.

CLOCKED SERIAL I/O (CSI/O)

Overview

The Z80382 includes a synchronous serial I/O port CSI/O which provides half-duplex transmission/reception of fixed 8-bit data with internal or external clocking. Internal clocking can range up to BUSCLK/20 bits/second. The CSI/O is ideal for implementing communications with other processors and peripherals, including serial memories. A block diagram of the CSI/O is illustrated in Figure 21.

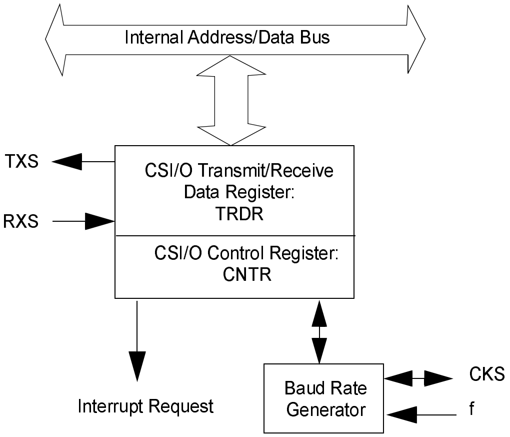


Figure 21. CSI/O Block Diagram

Note: The three pins associated with the CSI/O are multiplexed with other signals and must be configured for CSI/O operation in order to use the CSI/O as described in this section.

CSI/O Operation

The CSI/O can be operated using status polling or interrupt driven algorithms.

Transmit - Polling

1. Poll the TE bit in CNTR until TE = 0.
2. Write the transmit data into TRDR.

3. Set the TE bit in CNTR to 1.
4. Repeat steps 1, 2 and 3 for each transmit data byte.

Transmit - Interrupt Driven

1. Poll the TE bit in CNTR until TE = 0.
2. Write the first transmit data into TRDR.
3. Set the TE and EIE bits in CNTR to 1.
4. When the CSI/O End Flag interrupt occurs, write the next transmit data byte into TRDR.
5. Set the TE bit in CNTR to 1.
6. Repeat steps 4 and 5 for each transmit data byte.
7. After transmitting the last byte, clear the resulting interrupt by performing a ‘fake’ read or write of the TRDR.

Receive - Polling

1. Check the RE bit in CNTR. If it is a 1, a receive operation is already in progress and go to step 3. If it is a 0, and this is not the initial read cycle, go to step 4.
2. Set the RE bit in CNTR to 1.
3. Poll the RE bit in CNTR until RE = 0.
4. Read the receive data from TRDR.
5. Repeat steps 2, 3 and 4 for each receive data byte.

Receive - Interrupt Driven

1. Check the RE bit in CNTR. If it is a 1, a receive operation is already in progress and go to step 3. If it is a 0, and this is not the initial read cycle, go to step 4.
2. Set the RE and EIE bits in CNTR to 1.
3. When the CSI/O End Flag interrupt occurs, read the receive data from TRDR.
4. Set the RE bit in CNTR to 1.
5. Repeat steps 3 and 4 for each receive data byte, except do not execute step 4 after the last byte has been received.

CSI/O Interrupts

The CSI/O interrupt request circuit is depicted in Figure 22.

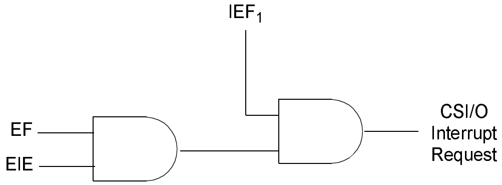


Figure 22. CSI/O Interrupt Generation

CSI/O Registers

The CSI/O module includes the following I/O-mapped registers that can be read and written by the 380C processor:

- CSI/O Control Register
- CSI/O Transmit/Receive Data Register

Details on each register are provided in the sections which follow. In the descriptions:

- `HHHHh` = register's address in the 380C I/O space, in hexadecimal
- DEF = default value after RESET
- x = indicates that the default value of the bit is indeterminate

CSI/O Transmit/Receive Data Register (TRDR)

CSI/O Transmit/Receive Data Register **000Bh**

BIT	7	6	5	4	3	2	1	0
R	CSI/O Receive Data							
W	CSI/O Transmit Data							
DEF	X	X	X	X	X	X	X	X

Bit(s)	Function	R/W	Description
7:0	CSI/O Transmit Data[1]	W	The data written into this register by the 380C is serialized and presented on the TXS pin.
7:0	CSI/O Receive Data[2]	R	Serial data received on the RXS pin is shifted into this register, from whence it can be read by the 380C.

NOTES:

1. TRDR is used for both CSI/O transmission and reception in a half-duplex protocol. The system design must ensure that transmit and receive operations do not occur simultaneously. For example, if CSI/O transmission is attempted while the CSI/O is receiving data, the CSI/O will not work. Also, the TRDR is not buffered. Attempting to perform a CSI/O transmit while the previous transmit data is still being shifted out causes the shift data to be immediately updated, corrupting the transmit operation in progress. Similarly, reading TRDR while a transmit or receive is in progress must be avoided.
2. CSI/O transmit and receive operations in progress are aborted during RESET. However, the contents of TRDR are not changed.

CSI/O Control Register (CNTR)

CSI/O Control Register 000Ah

BIT	7	6	5	4	3	2	1	0
R W	End Flag	End Inter-rupt Enable	Receive Enable	Transmit Enable	Reserved	CSI/O Speed Select		
DEF	0	0	0	0	0	1	1	1

Bit(s)	Function	R/W	Description
7	End Flag	R	EF is set to 1 by the CSI/O to indicate completion of an 8-bit data transmit or receive operation. If the RE (End Interrupt Enable) bit is a 1 when EF is set to 1, a CPU interrupt request is generated. Program access to TRDR should occur only when EF = 1. The CSI/O hardware clears EF to 0 when TRDR is read or written.
6	End Interrupt Enable	R/W	ME is set to 1 to enable a CPU interrupt request when EF is asserted. The interrupt request is inhibited if EIE is 0.
5	Receive Enable	R/W	A CSI/O receive operation is started by setting RE to 1, which then enables the data clock selected by SS2 - 0. In internal clock mode, the data clock is output from the CKS pin. In external clock mode, the clock is input on the CKS pin. In either case, the data clock causes data to be shifted in from the RXS pin. After receiving eight bits of data, the CSI/O automatically clears RE to 0, sets EF to 1, and generates an interrupt if EIE = 1. RE and TE are never both set to 1 at the same time.

Bit(s)	Function	R/W	Description
4	Transmit Enable	R/W	A CSI/O transmit operation is started by setting TE to 1, which enables the data clock selected by SS2 - 0. In internal clock mode, the data clock is output on the CKS pin. In external clock mode, the clock is input onto the CKS pin. In either case, the data clock causes data to be shifted out on the TXS pin. After transmitting eight bits of data, the CSI/O automatically clears TE to 0, sets EF to 1, and generates an interrupt if EI E = 1. RE and TE are never both set to 1 at the same time.
3	Reserved		
2:0	Speed Select	R/W	After RESET, the CKS pin is configured as an external clock input ([2:0] = 111). Changing these values causes CKS to become an output pin and the selected clock to be output when transmit or receive operations are enabled. Source and speed selection are as follows: [2:0] BUSCLK Divider 000 ÷ 20 001 ÷ 40 010 ÷ 80 011 ÷ 160 100 ÷ 320 101 ÷ 640 110 ÷ 1280 111 External Clock (1x)

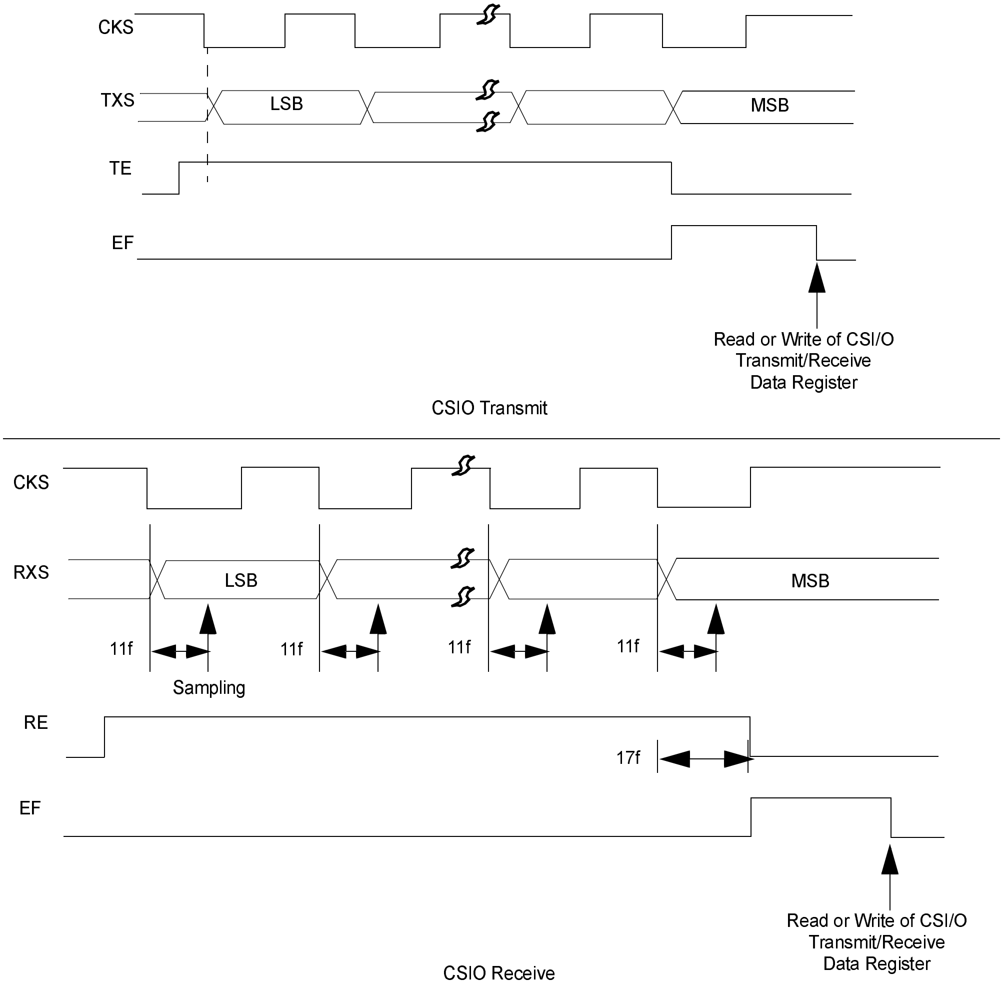
CSI/O Operation Timing Notes

1. Disable the transmitter and receiver (TE and RE = 0) before initializing or changing the baud rate. When changing the baud rate after completion of transmission or reception, a delay of at least one bit time is required before baud rate modification.
2. When RE or TE is cleared to 0 by software, a corresponding receive or transmit operation is immediately terminated. Normally, TE or RE

are automatically cleared to 0 when the corresponding operation is completed.

3. Simultaneous transmission and reception is not possible. TE and RE must not be set to 1 at the same time.

Timing diagrams of CSI/O transmit and receive operations are illustrated in Figure 23.


CKS
TXS
LSB
MSB
TE
EF
Read or Write of CSI/O Transmit/Receive Data Register
CKS
RXS
LSB
MSB
11f
11f
11f
11f
Sampling
RE
17f
EF
Read or Write of CSI/O Transmit/Receive Data Register

CSIO Transmit

CSIO Receive

Figure 23. CSI/O Transmit and Receive Timing Diagram

Counters, Timers and I/O Ports

OVERVIEW

The Z80382 includes two programmable reload timers (PRT) capable of providing a variety of counting and timing functions. Each PRT has an independent prescaler which can divide BUSCLK by powers of two ranging from 1 to 32,768 to provide the clock for a 16-bit down counter that starts at a programmable value. One of the channels can be programmed to set, reset, or toggle the TOUT signal pin.

In addition, a Watch-Dog timer (WDT) is provided to monitor program execution and reset the Z382 in case of faulty operation.

Four parallel bidirectional ports are available for general purpose input and output use, illustrated in Figure 24.

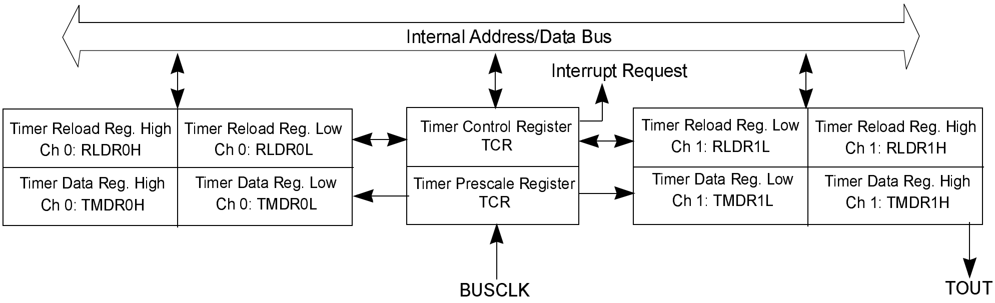


Figure 24. Programmable Reload Timer Block Diagram

The Z80382 contains two separate 16-bit Programmable Reload Timers (PRT). Each PRT channel contains a 16-bit down counter and a 16-bit reload register. The down counter can be directly read and written, and a down counter overflow interrupt can be programmably enabled or disabled. The two channels share a common status/control register and a

Timer Prescale Register which allows the time base for each PRT to be programmed as the Z382 BUSCLK divided by a power of two.

PRT1 can be programmed to set the TOUT pin High or Low or to toggle it when the channel counts down to zero. PRT1 can perform programmable output waveform generation. Figure 25 illustrates the timing of changes at the TOUT output.

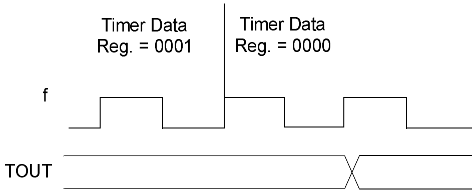


Figure 25. TOUT Timing Diagram

PRT Operation

Each PRT has a 16-bit Timer Data Register (TMDR). accessible as low and high byte registers (TMDRnH, TMDRnL). During RESET, the TMDR is set to `FFFFh`.

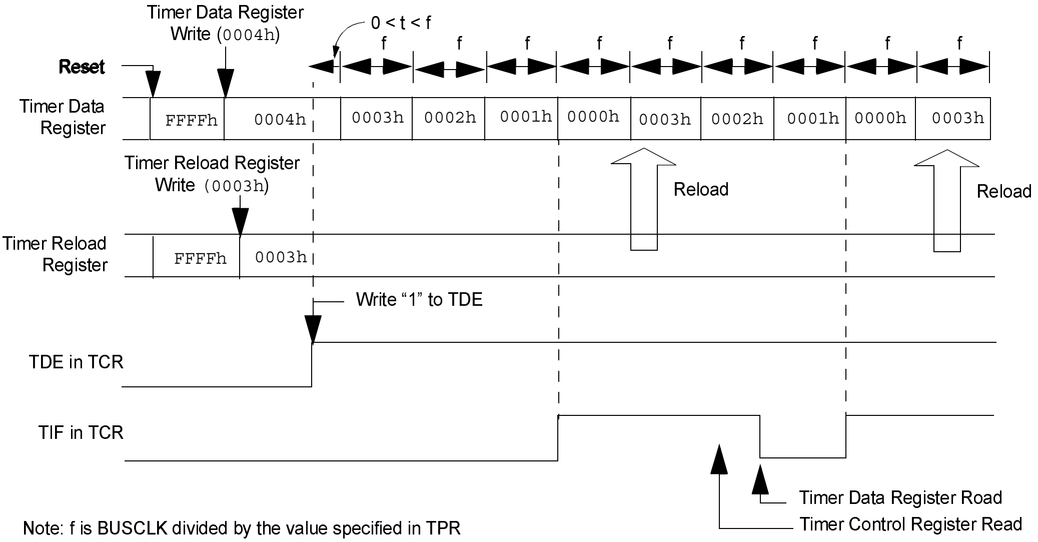
TMDR is decremented once every clock output from the timer prescaler, which divides the BUSCLK signal of the 280382 by a value which is specified for each PRT in the Timer Prescale register, TPR. When TMDR counts down to 0, it is automatically reloaded (at the next clock) with the value contained in its Timer Reload Register (TLDR).

TMDR should be read and written using the following procedures:

- TMDR can be read without stopping the timer, TMDRnL is read first, and then TMDRnH. The lower byte read stores the higher byte value in an internal register. The following higher byte read accesses this internal register. This feature ensures timer data validity by eliminating the potential problem of the upper timer byte updating between the two byte reads. Note the implications of TMDR higher byte internal storage for applications which may read only the upper byte. Specifically, reading TMDR in high byte - low byte order (or high byte only) may result in invalid data. All TMDR read routines should access both the lower and higher bytes, in that order.

- For writing an initial value into the TMDR, the TMDR down counting must be stopped by setting its TDE bit in the Timer Control register (TCR) to 0. Then, the higher and lower bytes or TMDR may be written in any order. Read operations in this state must still be performed in low byte - high byte order to ensure valid data.

Figure 26 illustrates the operation of the PRT.



Note: f is BUSCLK divided by the value specified in TPR

Figure 26. PRT Operation Timing Diagram

PRT Registers

The PRT module includes the I/O-mapped registers that can be read and written by the 380C processor (see Table 32):

Table 32. PRT Registers

Register Name	Type
Timer Prescale Register	Global
Timer Control Register	Global
Timer Data Reg. High, Low	Per Channel
Timer Reload Reg. High, Low	Per Channel

Details on each register are provided in the sections which follow. In the descriptions:

- `HHHHh` = registers address in the 380C I/O space, in hexadecimal
- DEF = default value after RESET
- x = indicates that the default value of the bit is indeterminate

Timer Prescale Register (TPR)

Timer Prescale Register 0011h

BIT	7	6	5	4	3	2	1	0
R W	PRT1 Prescale Select				PRTO Prescale Select			
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W	Description
7:4	PRT1 Prescale Select[1]	R/W	Selects the rate at which PRT1 is clocked. The input clock to PRT1 is BUSCLK/2^{PS}, where PS is the value of this field. This input provides for BUSCLK divisors ranging from 1 to 32,768, as follows:

7:4	PS
0000	1
0001	2
0010	4
0011	8
0100	16
0101	32
0110	64
0111	128

Bit(s)	Function	R/W	Description
7:4 Cont.			7:4 PS 1000 256 1001 512 1010 1,024 1011 2,048 1100 4,096 1101 8,192 1110 16,384 1111 32,768
3:0	PRT0 Prescale Select	R/W	Selects the rate at which PRT0 is clocked. The input clock to PRT0 is BUSCLK/2^{PS}, where PS is the value of this field. This input provides for BUSCLK divisors ranging from 1 to 32,768. See Table 32.

NOTES:

1. In general, software should select the fastest clock that prevents down-counter underflow for the longest time interval that needs to be timed. By use of this prescale facility and the 16-bit down-counter, maximum resolution can be obtained over a wide range of timing requirements.

Timer Control Register (TCR)

Timer Control Register **TCR: 0010h**

BIT	7	6	5	4	3	2	1	0
R W	Timer Interrupt Flag 1	Timer Interrupt Flag 0	Timer Interrupt Enable I	Timer Interrupt Enable 0	Timer1 Output Control		Timer Down Count Enable 1	Timer Down Count Enable 0
DEF	0	0	0	0	0	0	0	0

Bit(s)	Function	R/W	Description
7	Timer Interrupt Flag 1	R	When TMDR1 decrements to 0, TIF1 is set to 1. This generates an interrupt request if enabled by bit 5 (TIE1) = 1. TIF1 is reset to 0 by reading the TCR followed by reading the higher or lower byte of TMDR1.
6	Timer Interrupt Flag 0	R	When TMDR0 decrements to 0, TIF0 is set to 1. This generates an interrupt request if enabled by bit 4 (TIE0) = 1. TIF0 is reset to 0 by reading the TCR followed by reading the higher or lower byte of TMDR0
5	Timer Interrupt Enable 1	R/W	When TIE1 is set to 1, bit 7 (TIF1) = 1 generates a CPU interrupt request. When TIED is reset to 0, the interrupt request is inhibited. The interrupt is connected via the Assigned Vector facility.
4	Timer Interrupt Enable 0	R/W	When TIED is set to 1, bit 6 TIF0 = 1 generates a CPU interrupt request. When TIED is reset to 0, the interrupt request is inhibited. The interrupt is connected via the Assigned Vector facility.

Bit(s)	Function	R/W	Description
3:2	Timer Output Control	R/W	TOC1 (bit 3) and TOC0 (bit 2) control the effect of PRT1 on the TOUT pin (if this signal is programmed as an output in the pin multiplexing logic) when TMDR1 decrements to 0. The possible actions are listed below. See Figure 25 for timing of output changes. 3:2 Effect on TOUT 00 No effect on TOUT 01 Toggle TOUT (H → L, L → H) 10 Set TOUT Low 11 Set TOUT High
1	Timer Downcount Enable 1	R/W	TDE1 enables and disables down counting for PRT1. When TDE1 is cleared to 0, counting is stopped and TMDR1 may be freely read or written. TMDR1 does not decrement until TDE1 is set to 1.
0	Timer Downcount Enable 0	R/W	TDE0 enables and disables down counting for PRTO. When TDE0 is cleared to 0, counting is stopped and TMDR0 may be freely read or written. TMDR0 does not decrement until TDE0 is set to 1.

Timer Data High and Low Registers (TMDR0H, TMDR0L, TMDR1H, TMDR1L)

Timer Data Register High

TMDR0H: 000Dh, TMDR1H: 0015h

BIT	7	6	5	4	3	2	1	0
R W	Timer Data High							
DEF	1	1	1	1	1	1	1	1

Timer Data Register Low

TMDR0L: 000Ch, TMDR1L: 0014h

BIT	7	6	5	4	3	2	1	0
R W	Timer Data Low							
DEF	1	1	1	1	1	1	1	1

Bit(s)	Function	R/W	Description
7:0	Timer Data High	R/W	Contains the MS byte of the current 16-bit PRT value.
7:0	Timer Data Low	R/W	Contains the LS byte of the current 16-bit PRT value.

Timer Reload Registers (TLDR0H, TLDR0L, TLDR1H, TLDR1L)

Timer Reload Register High — TLDR0H: 000Fh, TLDR1H: 0017h

BIT	7	6	5	4	3	2	1	0
R W	Timer Reload High							
DEF	1	1	1	1	1	1	1	1

Timer Reload Register Low — TLDR0L: 000Eh, TLDR1L: 0016h

BIT	7	6	5	4	3	2	1	0
R W	Timer Reload Low							
DEF	1	1	1	1	1	1	1	1

Bit(s)	Function	R/W	Description
7:0	Timer Reload High	R/W	Contains the MS byte of the 16-bit reload value. When a PRT channel's TMDR counts down to 0, its TMDRH is automatically reloaded with the contents of this register.
7:0	Timer Reload Low	R/W	Contains the LS byte of the 16-bit reload value. When a PRT channel's TMDR counts down to 0, its TMDRL is automatically reloaded with the contents of this register. generates a CPU interrupt request. When TIED is reset to 0, the interrupt request is inhibited. The interrupt is connected via the Assigned Vector facility.

WATCH-DOG TIMER

A Watch-Dog timer (WDT) is provided to monitor program execution and reset the Z382 in case of faulty operation.

Overview

A Watch-Dog Timer (WDT) with programmable timeout intervals is available on the Z80382 to limit the impact of software (or possibly hardware) malfunction. The $\overline{\text{RESET}}$ input can be forced as an output upon the terminal count of the WDT, allowing external peripherals to be reset along with the Z80382. Unlike other on-chip functions, the WDT is enabled at Reset and must be disabled by software if its function is not desired. If software does not disable the WDT, it must periodically clear the WDT in order to avoid a hardware reset of the entire chip. A block diagram of the WDT is illustrated in Figure 27.

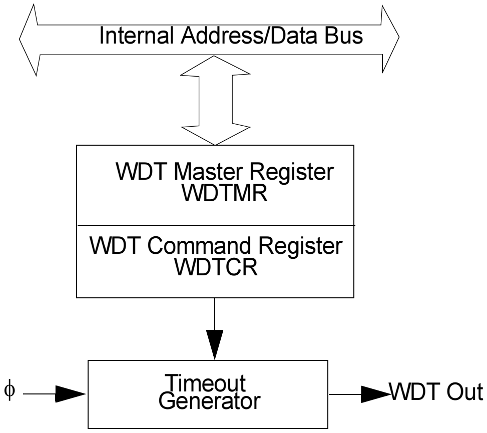


Figure 27. Watch-Dog Timer Block Diagram

WDT Registers

The WDT module includes the following I/O-mapped registers that can be read and written by the 380C processor:

- WDT Master Register
- WDT Command Register

Details on each register are provided in the sections which follow. In the descriptions:

- `HHHHh` = register's address in the 3800 I/O space, in hexadecimal
- DEF = default value after RESET
- x = indicates that the default value of the bit is indeterminate

Watch-Dog Timer Master Register (WDTMR)

Watch-Dog Timer Master Register **002$h**

BIT	7	6	5	4	3	2	1	0
R W	Watch-Dog Timer Enable	WDT Period		Drive/ Reset	Must Be 0011			
DEF	1	1	1	1	0	0	1	1

Bit(s)	Function	R/W	Description
7	Watch-Dog Timer Enable	R/W	If this bit is 1; as it is after a reset, the WDT is enabled. In order to disable the WDT, software must first write this bit as 0 and then write the value B1 H to the WDT Command Register. This two-step disabling procedure helps ensure that runaway software does not accidentally disable the WDT
6:5	WDT Period	R/W	These bits select the time interval that can expire before the software must write a Clear command to the WDT Command Register, before the WDT Timeout Generator outputs a reset pulse. The possible values are: 6-5 Timeout (number of BUSCLKs) 00 65,536 01 262,144 10 1,048,576 11 4,194,304

Bit(s)	Function	R/W	Description
4	Drive/RESET	R/W	If this bit is 1, as after Reset, the $\overline{\text{RESET}}$ pin is pulsed Low when the WDT times out, to reset external logic as well as the internals of the Z80382. If this bit is 0, a WDT timeout resets only the internals the Z80382.
3:0	No Function	R/W	For compatibility with WDTs in other ZiLOG controllers, these bits must be written as `0001`.

WDT Command Register (WDTCR)

WDT Command Register **0029h**

BIT	7	6	5	4	3	2	1	0
R W	Command							
DEF	X	X	X	X	X	X	X	X

Bit(s)	Function	R/W	Description
7:0	WDT Command	R/W	The WDT decodes two values written to the WDTCR:
			6. B1H: Writing this value, when bit 7 of the WDTMR is 0, disables the WDT.
			7. 4EH: Writing this value clears the WDT to zero, delaying the time then it could time out and issue a reset output

PARALLEL PORTS

The Z80382 has four 8-bit bidirectional ports called ports A through D. Each port includes two 8 bit registers: a Direction Register (which allows each bit to be programmable as an input or an output) and a Data Register.

The port outputs are of the 'weak latch' type to prevent floating output levels. The weak latches for Port B can be disabled via ODCR[7] to allow these pins to electrically 'float' for analog-type applications (see “Output Drive Control Register (ODCR)” on page 70).

After power-up, the state of the weak latches is undefined. The weak latches may be used as 'one time pull up/pull down resistors' by first setting the port to an output and then back to an input. The weak latch holds the output value until being overwitten by an external voltage level.

Because of pin multiplexing, ports A and D are available only in non-Host applications, and the registers for Ports A and D are used by the Mimic feature in Host applications. Additional information on the multiplexing of the Port pins is provided in the Device Configuration section of this document, Chapter 3.

Bit 3 in the System Configuration Register controls whether only the lowest eight address lines are decoded, allowing the port data and direction registers to be accessed in any page of I/O space (as on the 18x family, or whether access is limited to a single page (A[15:8] = 0). See “System Configuration Register (SYSCON)” on page 65.

Parallel Port Registers

The Parallel Port module includes the following I/O mapped registers that can be read and written by the 380C processor:

- Data Direction Register A, B, C, D
- Port Data Register A, B, 0, D

Details on each register are provided in the sections which follow. In the descriptions:

- `HHHHh` = register's address in the 3800 I/O space, in hexadecimal
- DEF = default value after RESET
- x = indicates that the default value of the bit is indeterminate

Data Direction Registers (DDRA, B, C, D)

Data Direction Register A, B, C, D — A: 00EDh, B: 00E4h, C: 00DDh, D: 00E7h

BIT	7	6	5	4	3	2	1	0
R W	Data Direction							
DEF	1	1	1	1	1	1	1	1

Bit(s)	Function	R/W	Description
7:0	Data Direction	R/W	The Direction Register determines which pins of the port are inputs and which are outputs. Setting a bit to a 1 programs the corresponding pin (if available) as an input while setting a bit to a 0 programs the pin as an output. In Host applications, the Port A and D Direction Registers are used to buffer data between the Hosts HD7-0 lines and the Z80382 for the Host DMA Mailbox and Host I/O Mailbox functions. See “Host Interface” on page 93.

Port Data Registers (DRA, B, C, D)

PORT DATA REGISTER A, B, C, D **A: 00EEh B: 00E5h, C: 00DEh D: 00E8h**

BIT	7	6	5	4	3	2	1	0
R W	Port Data							
DEF	X	X	X	X	X	X	X	X

Bit(s)	Function	R/W	Description
7:0	Port Data	R/W	When the 380C writes to the Data Register of an available port, the data is stored in this register. Any pins that are identified as outputs in the corresponding Port Direction Register are then driven with the new data. When the 380C reads the Data Register of an available port, the data on the external pins is returned. In Host applications, the Port A and D Data Registers are used for implementation of the "Host I/O Mailbox" feature. See "Host Interface" on page 93.

Register Summary and Index

OVERVIEW

This section provides a summary of all Z80382 registers by functional unit and describes certain differences between the Z80382 and previous Zilog processors, specifically the Z180 and Z380.

Z382 VERSUS Z380 REGISTER MAPS

Both the Z8018x family and the Z80380 assign built-in registers to I/O addresses in the range `0000-0017h`, but do so differently. Many of the built-in peripherals of the Z180 are replicated in the Z382, and these replicated peripherals reside in the conflicting I/O addresses.

The disposition of the various built-in I/O registers in the Z380 is as follows:

- Retained at same I/O address
  - Assigned Vectors Base Register
  - Trap and Break Register
- Moved because of an address conflict with Z180 devices
  - I/O Waits Register
  - Refresh Waits Register
  - Clock Control Register (formerly I/O Bus Control Reg 0)
  - Refresh Registers 0-2
  - STANDBY Mode Control Register
  - Interrupt Enable Register
  - Chip Version ID Register

- Not included in Z80382
  - Lower, Mid-range, and Upper Memory Chip Select Register
  - Lower, Mid-range, and Upper Memory Waits Registers
  - Memory Selects Master Enable Register
  - I/O Bus Control Register 1

Z80382 REGISTER SUMMARY

The following tables list all I/O registers of the Z80382 by major functional unit. A page reference to where in this manual the register is described in detail is also provided

In these tables:

- HHHHh indicates an address in the Z80382 I/O address space, in hexadecimal notation.
- #HHHH indicates an address in the host processor's address space, in hexadecimal notation.
- @HH indicates a host address relative to an I/O base address assigned by the PnP module, in hexadecimal notation.

Table 33. Device Configuration Registers

Register Name	Z382 Address	Z380 Address
"Assigned Vectors Base Register (AVBR)" on page 86	0018h	0018h
"Trap and Break Register (TBR)" on page 87	0019h	0019h
"I/O Waits Register (IOWR)" on page 72	001Eh	00Eh
"Refresh Wait Register (RFWR)" on page 84	001Fh	000Fh
"Clock Control Register (CCR)" on page 70	0021h	0011h
"Refresh Register 0, 1, 2 (RFSHR0, 1, 2)" on page 82	0023h	0013h
"Refresh Register 0, 1, 2 (RFSHR0, 1, 2)" on page 82	0024h	0014h
"Refresh Register 0, 1, 2 (RFSHR0, 1, 2)" on page 82	0025h	0015h
"STANDBY Mode Control Register (SMCR)" on page 73	0026h	x0016h
"Interrupt Enable Register 0 (IENR)" on page 85	0027h	0017h
"Chip Version ID Register (CVIDR)" on page 62	0020h	00FFh
"I/O Chip Select Registers (IOCS1H, IOCS1L, IOCS2H, IOCS2L)" on page 74	002Ah	—
"I/O Chip Select Registers (IOCS1H, IOCS1L, IOCS2H, IOCS2L)" on page 74	002Bh	—
"I/O Chip Select Registers (IOCS1H, IOCS1L, IOCS2H, IOCS2L)" on page 74	002Ch	—
"I/O Chip Select Registers (IOCS1H, IOCS1L, IOCS2H, IOCS2L)" on page 74	002Dh	—
"RAM Address Registers (RAMH, RAML)" on page 78	002Eh	—
"RAM Address Registers (RAMH, RAML)" on page 78	002Fh	—
"ROM Address Registers (ROMH, ROML)" on page 80	0030h	—
"ROM Address Registers (ROMH, ROML)" on page 80	0031h	—

Table 33. Device Configuration Registers (Continued)

Register Name	Z382 Address	Z380 Address
“Memory Mode Register 1 (MMR1)” on page 76	0032h	—
“Memory Mode Register 2 (MMR2)” on page 77	00D3h	—
“System Configuration Register (SYSCON)” on page 63	0036h	—
“Pin Multiplexing Register (PINMUX)” on page 66	0037h	—
“Output Drive Control Register (ODCR)” on page 68	003Ah	—
“INT3-1 Control Register (I31CR)” on page 88	003Bh	—

Table 34. Clocked Serial I/O Registers

Register Name	I/O Address
“CSI/O Control Register (CNTR)” on page 335	000Ah
“CSI/O Transmit/Receive Data Register (TRDR)” on page 334	000Bh

Table 35. Port Registers

Register Name	I/O Address
“Port Data Registers (DRA)” on page 358	00EEh
“Data Direction Registers (DDRA)” on page 357	00EDh
“Port Data Registers (DRB)” on page 358	00E5h
“Data Direction Registers (DDRB)” on page 357	00E4h
“Port Data Registers (DRC)” on page 358	00DEh
“Data Direction Registers (DDRC)” on page 357	00DDh
“Port Data Registers (DRD)” on page 358	00E8h
“Data Direction Registers (DDRD)” on page 357	00E7h

Table 36. ASCI Registers

Register Name	I/O Address
"ASCI Control Register A (CNTLA0)" on page 251	0000h
"ASCI Control Register A (CNTLA1)" on page 251	0001h
"ASCI Control Register B (CNTLB0)" on page 254	0002h
"ASCI Control Register B (CNTLB1)" on page 254	0003h
"ASCI Status Register (STAT0)" on page 260	0004h
"ASCI Status Register (STAT1)" on page 260	0005h
"ASCI Transmit Data Register (TDR0)" on page 263	0006h
"ASCI Transmit Data Register (TDR1)" on page 263	0007h
"ASCI Receive Data Register (RDR0)" on page 262	0008h
"ASCI Receive Data Register (RDR1)" on page 262	0009h
"ASCI Extension Control Register (ASEXT0)" on page 257	0012h
"ASCI Extension Control Register (ASEXT1)" on page 257	0013h
"ASCI Time Constant Registers (ASTC0H)." on page 256	001Ah
"ASCI Time Constant Registers (ASTC0L)." on page 256	001Bh
"ASCI Time Constant Registers (ASTC1H)." on page 256	001Ch
"ASCI Time Constant Registers (ASTC1L)." on page 256	001Dh
"ASCI DMA Control Register (ADCR0)" on page 264	0038h
"ASCI DMA Control Register (ADCR1)" on page 264	0039h

Table 37. Watch-Dog Timer Registers

Register Name	I/O Address
"Watch-Dog Timer Master Register (WDTMR)" on page 353	0028h
"WDT Command Register (WDTCR)" on page 354	0029h

Table 38. DMA Registers

Register Name	I/O Address
"DMA Control Register (DMACR)" on page 227	003Eh
"DMA Vector Register (DMAVR)" on page 229	003Fh
DMA 0 "List Address Register (LAR)" on page 219[1,2,3]	0040h
DMA 0 "List Address Register (LAR)" on page 219[1,2,3]	0041h
DMA 0 "List Address Register (LAR)" on page 219[1,2]	0042h
DMA 0 "DMA Control/Status Register (DMACSR)" on page 235	0043h
DMA 1 "List Address Register (LAR)" on page 219[1,2,3]	0044h
DMA 1 "List Address Register (LAR)" on page 219[1,2,3]	0045h
DMA 1 "List Address Register (LAR)" on page 219[1,2]	0046h
DMA 1 "DMA Control/Status Register (DMACSR)" on page 235	0047h
DMA 2 "List Address Register (LAR)" on page 219[1,2,3]	0048h
DMA 2 "List Address Register (LAR)" on page 219[1,2,3]	0049h
DMA 2 "List Address Register (LAR)" on page 219[1,2]	004Ah
DMA 2 "DMA Control/Status Register (DMACSR)" on page 235	004Bh
DMA 3 "List Address Register (LAR)" on page 219[1,2,3]	004Ch
DMA 3 "List Address Register (LAR)" on page 219[1,2,3]	004Dh

Table 38. DMA Registers (Continued)

Register Name	I/O Address
DMA 3 "List Address Register (LAR)" on page 219[1,2]	004Eh
DMA 3 "DMA Control/Status Register (DMACSR)" on page 235	004Fh
DMA 4 "List Address Register (LAR)" on page 219[1,2,3]	0050h
DMA 4 "List Address Register (LAR)" on page 219[1,2,3]	0051h
DMA 4 "List Address Register (LAR)" on page 219[1,2]	0052h
DMA 4 "DMA Control/Status Register (DMACSR)" on page 235	0053h
DMA 5 "List Address Register (LAR)" on page 219[1,2,3]	0054h
DMA 5 "List Address Register (LAR)" on page 219[1,2,3]	0055h
DMA 5 "List Address Register (LAR)" on page 219[1,2]	0056h
DMA 5 "DMA Control/Status Register (DMACSR)" on page 235	0057h
DMA 6 "List Address Register (LAR)" on page 219[1,2,3]	0058h
DMA 6 "List Address Register (LAR)" on page 219[1,2,3]	00059h
DMA 6 "List Address Register (LAR)" on page 219[1,2]	005Ah
DMA 6 "DMA Control/Status Register (DMACSR)" on page 235	005Bh
DMA 7 "List Address Register (LAR)" on page 219[1,2,3]	005Ch
DMA 7 "List Address Register (LAR)" on page 219[1,2,3]	005Dh
DMA 7 "List Address Register (LAR)" on page 219[1,2,]	005Eh
DMA 7 "DMA Control/Status Register (DMACSR)" on page 235	005Fh

NOTES:

1. These addresses access the List Address registers when DMACR[1:0] = 00 or 11.
2. These addresses access the Buffer Address registers when DMACR[1:0] = 01.
3. These addresses access the Buffer Length registers when DMACR[1:0] = 10.

Table 39. Programmable Reload Timer Registers

Register Name	I/O Address
“Timer Data High and Low Registers (TMDR0L)” on page 349	000Ch
“Timer Data High and Low Registers (TMDR0HL)” on page 349	000Dh
“Timer Reload Registers (TLDR0L)” on page 350	000Eh
“Timer Reload Registers (TLDR0H)” on page 350	000Fh
“Timer Data High and Low Registers (TMDR1L)” on page 349	0014h
“Timer Data High and Low Registers (TMDR1H)” on page 349	0015h
“Timer Reload Registers (TLDR1L)” on page 350	0016h
“Timer Reload Registers (TLDR1H)” on page 350	0017h
“Timer Control Register (TCR)” on page 347	0010h
“Timer Prescale Register (TPR)” on page 345	0011h

Table 40. HDLC Registers

Register Name	I/O Address
“Global HDLC Vector Register HDLCV” on page 307	003Dh
“Transmit Mode Register TMR0,” on page 283	0060h
“Transmit Mode Register TMR0” on page 283	0061h
“Transmit Mode Register TMR0” on page 283	0062h
“Transmit Mode Register TMR0” on page 283	0063h
“Receive Mode Register RMR0” on page 296	0064h
“Receive Mode Register RMR)” on page 296	0065h
“Counter Access Port CAP0” on page 304	0066h

Table 40. HDLC Registers (Continued)

Register Name	I/O Address
"DMA Select Register DSR0" on page 306	0067h
"Transmit Mode Register TMR1" on page 283	0068h
"Transmit Mode Register TMR1" on page 283	0069h
"Transmit Mode Register TMR1" on page 283	006Ah
"Transmit Mode Register TMR1" on page 283	006Bh
"Receive Mode Register RMR1" on page 296	006Ch
"Receive Mode Register RMR1" on page 296	006Dh
"Counter Access Port CAP1" on page 304	006Eh
"DMA Select Register DSR1" on page 306	006Fh
"Transmit Mode Register TMR2" on page 283	0070h
"Transmit Mode Register TMR2" on page 283	0071h
"Transmit Mode Register TMR2" on page 283	0072h
"Transmit Mode Register TMR2" on page 283	0073h
"Receive Mode Register RMR2" on page 296	0074h
"Receive Mode Register RMR2" on page 296	0075h
"Counter Access Port CAP2" on page 304	0076h
"DMA Select Register DSR2" on page 306	0077h

Table 41. GCI/SCIT Registers

Register Name	I/O Address
"GCI Control Register (GCICR)" on page 318	`00C0h`
"GCI Status Register 1 (GCISR1)" on page 325	`00C1h`
"GCI Status Register 2 (GCISR2)" on page 327	`00C2h`
"GCI Interrupt Enable Register (GCIIE)" on page 329	`00C3h`
"Monitor 0 Transmit/Receive Data Register (MON0)" on page 320	`00C4h`
"Monitor 1 Transmit/Receive Data Register (MON1)" on page 321	`00C5h`
"C/I0/2 Transmit/Receive Data Register (CI02)" on page 322	`00C6h`
"C/I1 Transmit/Receive Data Register (CI1)" on page 324	`00C7h`

Table 42. Z80382 Mimic Registers

Register Name	I/O Address	Host Address
"MIMIC Master Control Register (MMC)" on page 122	00FFh	None
"Interrupt Under Service/Interrupt Pending Register (IUSIP)" on page 119	00FEh	None
"Interrupt Enable Register (IER)" on page 117	00FDh	None
"Interrupt Vector Register (IVEC)" on page 115	00FCh	None
"Receiver Time Constant Register (RTCR)" on page 114	00FBh	None
"Transmitter Time Constant Register (TTCH)" on page 113	00FAh	None
"Divisor Latch MSB (DLM)" on page 140	00F9h	@01, DLAB = 1
"Divisor Latch LSB (DLL)" on page 139	00F8h	@00, DLAB = 1
"Scratch Register (SCR)" on page 138	00F7h	@07
"Modem Status Register (MSR)" on page 136	00F6h	@06
"Line Status Register (LSR)" on page 134	00F5h	@05
"Modem Control Register (MCR)" on page 132	00F4h	@04
"Line Control Register (LCR)" on page 130	00F3h	@03
"Interrupt Enable Register (IER)" on page 128	00F1h	@01, DLAB = 0
"Receiver Buffer Register (RBR)" on page 126	00F0h	@00, DLAB = 0 (Read)
"Transmitter Holding Register (THR)" on page 127	00F0h	@00, DLAB = 0 (Write)

Table 42. Z80382 Mimic Registers (Continued)

Register Name	I/O Address	Host Address
“MIMIC DMA Control Register (MDCR)” on page 112	00EFh	None
“FIFO Status and Control Register (FSCR)” on page 109	00ECh	None
“Transmitter Timeout Time Constant (TTTC)” on page 108	00EBh	None
“Receiver Timeout Time Constant (RTTC)” on page 107	00EAh	None
“Interrupt Identification Register (IIR)” on page 141	None	@02 (Read)
“FIFO Control Register (FCR)” on page 124	00E9h	@02 (Write)
“MIMIC Modification Register (MMR)” on page 106	00E9h	None
“Baud Rate Generator Low Register (BRGL)” on page 104	00E1h	None
“Baud Rate Generator High (BRGH)” on page 105	00E0h	None
“I/O and BRG Control Register (IOBRG)” on page 102	00D6h	None

Table 43. Host DMA Mailbox and Host I/O Mailbox Registers

Register Name	I/O Address	Host Address
"Host DMA Control Register (HDCR)" on page 156	`00E6h`	None
"Host DMA Mailbox Control Register (HMC)" on page 154	`00D2h`	None
"Host DMA Transmit Register 1 (HDMAT1)" on page 153	`00D1h` (Read)	See note[1]
"Host DMA Receive Register 1 (HDMAR1)" on page 152	`00D1h` (Write)	See note[1]
"Host DMA Transmit Register 0 (HDMAT0)" on page 151	`00D0h` (Read)	See note[1]
"Host DMA Receive Register 0 (HDMAR0)" on page 150	`00D0h` (Write)	See note[1]
"Host I/O Status Register (HIOS)" on page 159	`00D5h`	@02
"Host I/O Mailbox Data Transfer Registers" on page 161 D Data Direction Register (Data: Host → 380)	`00E7h`	PCMCIA: #0206 ISA: @03
"Host I/O Mailbox Data Transfer Registers" on page 161 D Data Register (Data: Host → 380)	`00E8h`	PCMCIA: #0200 ISA: @00
"Host I/O Mailbox Registers" on page 158 A Data Direction Register (Data: Host → 380)	`00EDh`	PCMCIA: #0202 ISA: @01
"Host I/O Mailbox Registers" on page 158 A Data Register (Data: 380 → Host	`00EEh`	PCMCIA: #0202 ISA: @01

NOTES:

[1] These registers are implicitly addressed by the DMA Acknowledge signals $\overline{\text{HDAK0}}$ and $\overline{\text{HDAK1}}$ for registers 0 and 1 respectively.

Table 44. PCMCIA Memory and Registers

Register Name	I/O Address	Host Address
"Configuration Option Register (COR)" on page 172 Low Attribute Memory	0100h - 0177h	Attr. #0000 - 00EEh (even)
"Configuration Option Register (COR)" on page 172 Base Address Registers 0-6	0178h - 017Eh	Attr. 00F0h - 00FCh (even)
"Z380 Control Register (ZCR)" on page 171	017Fh	None
"Configuration Option Register (COR)" on page 172	0180h	Attr. #0100
"Card Configuration and Status Register (CCR)" on page 174	0181h	Attr. #0102
"Pin Replacement Register (PRR)" on page 176	0182h	Attr. #0104
"Socket and Copy Register" on page 178	0183h	Attr. #0106
"Extended Status Register (ESR)" on page 179	0184h	Attr. #0108
"Image Base Address Registers (IBRL)" on page 180	0185h	Attr. #010A
"Image Base Address Registers IBRU)" on page 180	0186h	Attr. #0100
"Interface Version Number Register (IVNR)" on page 182	0187h	Attr. #010E
"Configuration Option Register (COR)" on page 172 High Attribute Memory	0188h - 01FFh	Attr. #0110 - #01FE (even)

Table 45. ISA Plug-and Play-Registers

Register Name	I/O Address	Host Address[1]
"PnP Address Port" on page 193	None	#02794
"PnP Write Data Port (PNPWDP)" on page 194	None	#0A794
"PnP Read Data Port (PNPRDP)" on page 195	None	#0203 - #03FF
"PnP Read Data Address Register (PNPRDA)" on page 196	None	@00
"PnP Isolation Register (PNPIR)" on page 197	None	@01
"PnP Configuration Control Register (PNPCC)" on page 198	None	@02
"PnP Wake Register (PNPWR)" on page 199	None	@03
"PNP Master Register (PNPMR)" on page 213	0102h	None
"Resource Data Register (PNPRD)" on page 200	0104h	@04
"PnP Status Register (PNPSR)" on page 201r	0105h	@05
"PnP Card Select Number Register (PNPCSN)" on page 202	0106h	@06
"PnP Logical Device Number Register (PNPLDN)" on page 203	None	@07
"PnP Activate Register (PNPACT)" on page 204	0130h	@30
"PnP I/O Range Check Register (PNPRC)" on page 205r	None	@31
"PnP I/O Base Address 0 High Register - I/O Mailbox (IOMBXAH)" on page 206	0160h, 0161h	@060, @61
"PnP I/O Base Address 1 High Register - MIMIC (MIMICAH)" on page 208	0162h, 0163h	@62, @63
"PnP Interrupt Request Level 0 Register (PNPIRQ)" on page 210	0170h	@70
"PnP DMA Channel 0 Register (PNPDMA0)" on page 211 "PnP DMA Channel 1 Register (PNPDMA1)" on page 212	0174h, x0175h	@74, @75

NOTES:

1. '@' in this column indicates the value(s) required to be written by the host into the PnP Address Port to access the register(s).

Z80380/380C Differences

OVERVIEW

This Appendix describes the differences between the stand-alone Z80380 MPU and the 380C MPU core embedded in the Z80382. See also:

- Table 10, “Assigned Interrupt Vectors”, as contrasted to Table 5 in the Z380 Microprocessor Product Specification.
- “Interrupt Mode 2 Response” on page 48, specifically the paragraph describing interrupt vector handling in EXTENDED Mode.
- “Byte Ordering” on page 50.
- “Z382 Versus Z380 Register Maps” on page 359
- Table 33, “Device Configuration Registers”

INSTRUCTION SET DIFFERENCES

RESC–Reset Control Bit

The RESC instruction is now capable of resetting the EXTENDED Mode (XM) bit. The syntax for this instruction is: RESC XM

POP–Pop Control Register

The POP SR instruction sets and resets the XM bit.

Changing the XM Bit

1. NATIVE/EXTENDED mode affects how certain 16 or 32 bit arithmetic instructions function, the size of return addresses for CALL, RST, and RETURN instructions, as well as interrupts and traps, what size addresses are fetched from interrupt tables, and whether PC incrementing and SP incrementing and decrementing carry from bit 15 to bit 16.
2. Interrupts must be DI'ed when XM is changed in either direction.
3. Before interrupts are re-enabled in mode 2 or 3 after XM is changed in either direction, either the I register must be reloaded to point to an interrupt table with the right-sized entries for the new mode, or the table must be restructured at the same address.
4. The intended use of the XM-clearing function is on exit from an operating system, debug monitor, real-time exec, or similar system software, to a task or program running in native mode.
5. Typical operation is for the system software to PUSH the task's SR, disable interrupts if they are still enabled, set EXTENDED mode, and reorganize the interrupt table before El'ing, if interrupts are allowed during its operation. On returning to a NATIVE-mode task or program, if the operating system changed the interrupt table it must disable interrupts, change the table back again, then POP the task's SR to restore the task's XM, LW, and IE bits before returning to the task.
6. A RET instruction following the POP SR fetches its return address consistently with the new state of the XM bit.
7. If the POP SR also enabled interrupts, an immediately following interrupt acknowledge sequence stores its return address consistently with the new state of the XM bit.

8. Return addresses on existing stacks, from CALLS, RSTs, interrupts, or traps, can in general only be returned to in the same mode (native or extended) in which each was pushed.

EX A, L; EX A, A'- Exchange With Accumulator

The EX A, L and EX A, A' instructions, which were non-operational in the Z380, have been fixed.

EFFECT OF RESET ON CPU REGISTERS

In the 380C, Reset clears the high word of SP. No MPU registers other than PC and SP are affected by Reset.

Z380 ERRATA FIXED

All errata documented in the Z380 Customer Procurement Specification, ZiLOG document number DC-4120-07, have been corrected in the 380C, or eliminated by removing the module containing the error (for instance, the Z380 Chip Select and Wait State block).

Crystal Oscillator Operation

OVERVIEW

The crystal oscillator circuit in the Z80382 is designed to operate with fundamental-mode crystals of up to 20MHz. Third-overtone mode crystals must be used above that frequency, and are preferred for any crystal frequency above 15MHz.

Figure 28 illustrates the components required to be connected to the CLKI and CLKO pins for operation as a crystal oscillator. When using a third-overtone mode crystal, the additional LC tank circuit must be added to filter the fundamental frequency component that the crystal generates.

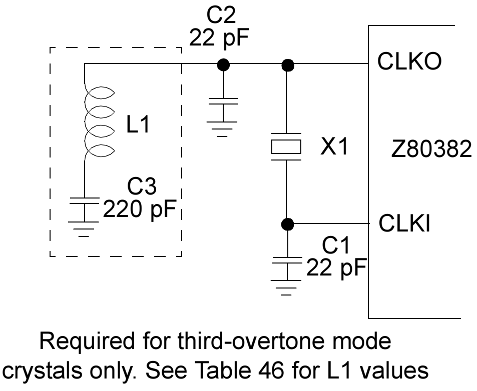

C2
22 pF
L1
C3
220 pF
X1
CLKO
Z80382
CLKI
C1
22 pF
Required for third-overtone mode
crystals only. See Table 46 for L1 values

Figure 28. Crystal Oscillator Connections

Because the tank circuit is a filter, the value of the inductor (L1) needs to be tuned for the frequency used. The sample points in Table 46 enable the designer to interpolate, if necessary, for the actual frequency being used:

Table 46. Frequency and L1 Values

Crystal Frequency (MHz)	L1 (µH)
20	12.0
25	8.2
33	4.7

It may also be desirable to increase the value of C1 to 30 pF to maintain balanced capacitance between the two legs of the oscillator.

Index