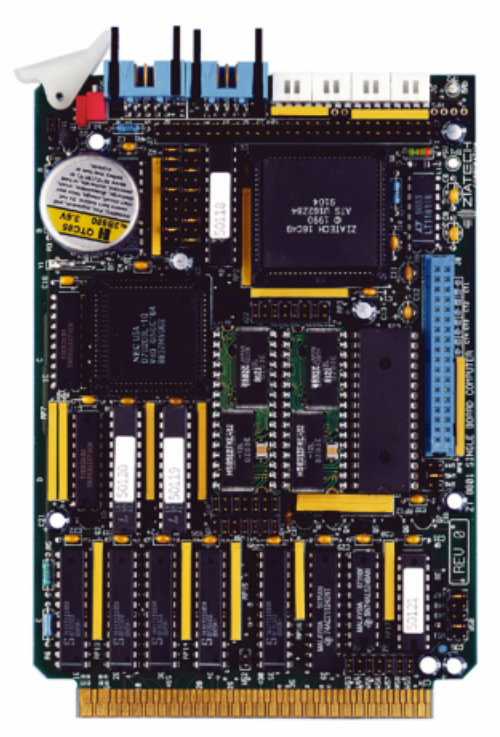
ZT 8801/ZT 88CT01



Single Board V40 Computers

ZIATECH CORPORATION

CONTENTS

WHAT'S IN THIS MANUAL

This manual describes the operation and use of the ZT 8801 and ZT 88CT01 Single Board V40 Computers. The boards are functionally identical; however, the ZT 88CT01 consumes less power and operates over a wider temperature range. Refer to Appendix B, "Specifications," for the particular specifications of each version of the board. Throughout the manual, ZT 8801 refers to both products unless specifically noted otherwise.

The following summarizes the focus of each major section in this manual. Click on any green text to move to that section.

Chapter 1, "Introduction**,"** offers an overview of the ZT 8801. It includes a product definition, a list of product features, a functional block diagram, and a brief description of each block. This chapter is most useful to those who wish to compare the features of the ZT 8801 against the needs of a specific application.

Chapter 2, "Getting Started**,"** summarizes the information you need to begin using your ZT 8801. This includes system requirements, shared memory mapping, local memory mapping, I/O mapping, and frontplane connector descriptions. This may be all the information you need to begin using the ZT 8801.

Chapter 3, "Theory of Operation**,"** presents a detailed description of system level operation. This chapter covers memory, I/O organization, and interrupt structure, and it answers some commonly asked questions about the ZT 8801.

Chapter 4, "Processor Description (V40)**,"** divides the V40 into functional blocks and presents a theory of operation for each. This chapter is most useful to those who are not familiar with the architecture of the V40/8088 series microprocessor.

Chapter 5, "Processor Configuration (V40)**,"** describes the architecture of a V40 software programmable register set used to configure peripherals resident on the V40 for specific applications. These peripherals are described in greater detail in the following chapters.

Chapter 6, "Counter/Timers (V40)**,"** describes the V40 Counter/Timer Control Unit, which includes three 16-bit counter/timers and is functionally equivalent to the 8254 Programmable Interval Timer. This chapter also includes register descriptions.

Chapter 7, "Interrupt Controller (V40)**,"** describes the V40 Interrupt Control Unit, a programmable interface between interrupt generating peripherals and the CPU. This chapter also includes register descriptions.

Chapter 8, "DMA Controller (V40)**,"** describes the V40 Direct Memory Access Control Unit, a programmable peripheral device used to allow temporary masters (other bus masters) access to the on-board system memory. This chapter also includes register descriptions.

Chapter 9, "Serial Communications (V40)**,"** describes the V40 Serial Control Unit, a single serial channel that performs asynchronous serial communication between the V40 and a serial device external to the ZT 8801. This chapter also includes register descriptions and baud rate information.

Chapter 10, "Watchdog Timer**,"** describes the function, configuration, and operation of the ZT 8801 watchdog timer, which is used to monitor ZT 8801 operation and take corrective action if the ZT 8801 fails to function as programmed.

Chapter 11, "SBX Expansion Module**,"** contains a description and installation information for the optional SBX expansion module, which can be used to expand the I/O capabilities of the ZT 8801.

Chapter 12, "Parallel I/O Adapter (16C49)**,"** discusses the 16C49 Parallel Interface Adapter (PIA) in detail. This chapter covers all information necessary to program and interface to the 16C49. It also describes the ZT 2226 24-Channel I/O Mounting Rack, ZT 2223 Industrial I/O Adapter Board, and ZT 2225 Industrial I/O Cable Adapter as they are used with the ZT 8801.

Chapter 13, "Real-Time Clock/Calendar (DS 1202)**,"** describes the function, configuration, and operation of the real-time clock/calendar. The real-time clock provides timekeeping features that are useful in many applications.

Appendix A, "Jumper Configurations**,"** describes the jumper options used to tailor the ZT 8801 to a specific application.

Appendix B, "Specifications**,"** contains electrical, mechanical, and environmental specifications. Frontplane connector pinouts and cable drawings are also included.

Appendix C, "PIA System Setup Considerations**,"** illustrates precautions you should take to prevent the latchup conditions that can occur with CMOS technology.

Appendix D, "Customer Support**,"** offers a product revision history, technical support information, the Ziatech warranty, and instructions for returning the ZT 8801 if service is necessary.

CHAPTER 1. INTRODUCTION

This chapter offers an overview of the ZT 8801 and ZT 88CT01 Single Board V40 Computers. It includes a product description, a list of product features, a functional block diagram, and a brief description of each block.

PRODUCT DESCRIPTION

The ZT 8801 is an 8 MHz, 8-bit, Single Board Computer (SBC) designed for control applications on the STD bus. Ziatech offers DOS (MS-DOS®) and STD ROM™ (Borland Turbo Debugger® environment using Paradigm Systems' DEBUG/RT™) as development platforms on the ZT 8801. The ZT 8801's high level of integration allows for a complete DOS system on one board. On-board PC peripherals are located at the same I/O addresses as on the IBM PC®, which provides a greater degree of software compatibility.

The ZT 88CT01 is the CMOS version of the ZT 8801. It is designed for an extended temperature range and low power applications. All references to the ZT 8801 in this manual also apply to the ZT 88CT01, unless otherwise noted.

The use of the NEC V40 microprocessor, an 8088 compatible processor with a superset of the 8088 instruction set, helps increase performance over that of 8088-based STD CPU boards. The V40 is a CMOS device, which results in lower power consumption. To further reduce power consumption, the ZT 8801 utilizes CMOS technology for most of the peripheral logic. The ZT 88CT01 is composed entirely of TTL-compatible, CMOS devices.

Peripherals include three timer/counters, an interrupt controller, a real-time clock, one RS-232-C serial port (which may be configured to be RS-485), a 48-line digital I/O interface, and three 32-pin memory sockets. One memory socket accepts a Flash or EPROM device of up to 512 Kbytes, and each of the other two memory sockets accepts up to 512 Kbytes of RAM. A bank switching mechanism is employed to permit up to 1 Mbyte of RAM and 512 Kbytes of EPROM all within the 1 Mbyte addressing range of the V40. A general purpose LED indicator and pushbutton reset are also provided.

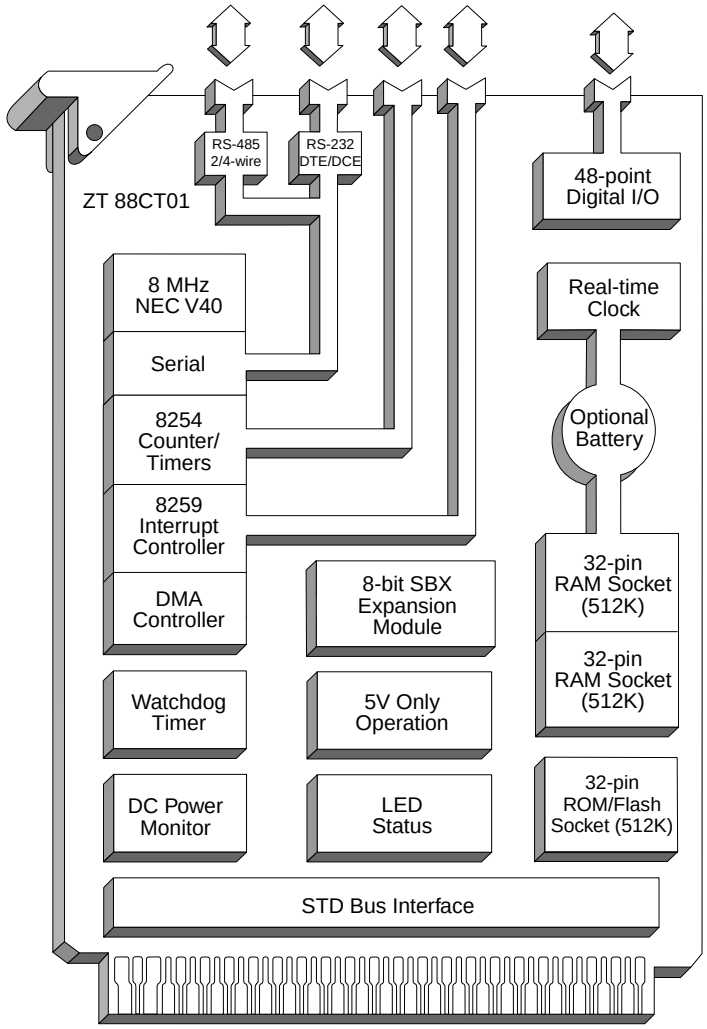
All RAM and the real-time clock may be optionally battery-backed by a 1 Amp-hour lithium battery. DC power failure detection is provided to switch to the battery backup mode during +5 VDC failure.

Ziatech's DOS and STD ROM options provide software support. DOS includes the MS-DOS operating system for the ZT 8801. STD ROM provides software development capabilities when used in conjunction with an IBM PC. The STD ROM option is useful for applications in which the target system does not require an operating system.

FEATURES OF THE ZT 8801

- STD-80 bus compatible
- Optional CMOS version available (ZT 88CT01)
- 8088/8086 code compatible
- Three 32-pin memory sockets, configurable for two RAMs and one EPROM/Flash
- Acceptable RAM sizes are 32 Kbytes through 512 Kbytes
- Acceptable EPROM/Flash sizes are 32 Kbytes through 512 Kbytes
- +5 V-only operation (+12 V required for Flash programming)
- One 8259 compatible interrupt controller (V40)
- Backplane cascaded interrupt support
- Three 8254 compatible counter/timers (V40)
- One RS-232-C serial channel (V40), RS-422/485 configurable
- 48-point Opto 22 compatible digital I/O interface (16C49)
- Eight points of event sense
- SBX expansion module interface
- Latching frontplane connectors
- Real-time clock/calendar (DS 1202)
- Optional battery backup for all RAM and clock
- Watchdog timer
- DC power-fail protection
- Programmable LED indicator
- Pushbutton reset switch
- Optional Ziatech DOS operating system software
- Optional STD ROM development/debug software
- Optional industrial I/O adapter board (ZT 2223), cable adapter (ZT 2225), and 24-channel I/O module mounting rack (ZT 2226)
- Optional cables for serial port, digital I/O port, and Centronics printer interface
- Fully tested while cycling temperature from ambient to 55° C to guarantee reliability (to 80° C for ZT 88CT01)
- Sleep mode provided for on-board oscillators
- Five-year warranty

FUNCTIONAL BLOCKS



STD Bus Interface

The STD bus interface allows the ZT 8801 to interface to other peripherals in the STD bus. Refer to Chapter 3, "Theory of Operation," for details.

V40 (µPD70208) Processor

The NEC V40 is an 8088 compatible microprocessor with a 16-bit internal data bus and an 8-bit external data bus. The V40 executes all code written for the 8088/8086 family of microprocessors and includes a superset of their instruction set. Performance enhancements are provided by way of such architectural features as a dual 16-bit internal data bus, high-speed effective address generation, and additional hidden temporary registers. The added instructions include shift and rotate by immediate value, move string, stack manipulations, and 8080 emulation mode. The 8080 emulation mode enables existing 8-bit 8080 software to run on new 16-bit hardware with few or no software modifications.

Memory and I/O Addressing

The ZT 8801 has three 32-pin JEDEC compatible byte-wide sockets. One of the sockets accepts 128 through 512 Kbyte EPROM/Flash; the other two accept 32 through 512 Kbyte RAMs. The EPROM/Flash socket supports both 5 V and 12 V Flash memory.

Twenty-bit addressing is used for on-board accesses, giving the system 1 Mbyte of direct address space. STD bus accesses have an additional four bits (A20-23) driven low to be compatible with 24-bit addressed memory boards. I/O accesses are accomplished with a 16-bit address, providing 64 Kbytes of I/O space for 16-bit addressed I/O boards or 256 bytes for 8-bit addressed boards.

Wait-State Generator

The ZT 8801 contains a wait-state generator to accommodate I/O and memory boards that need more access time. If enabled, the wait-state generator inserts up to three clock cycles within the normal four clock bus cycle to increase the cycle to seven clocks. This gives memory and I/O boards additional time between address valid time and the end of the bus cycle to complete an access. DOS and STD ROM program one additional wait state (five clock transfer) by default.

Direct Memory Access (DMA)

The ZT 8801 supports external DMA controllers via the BUSRQ* (pin 42) and BUSAK* (pin 41) STD bus control signals. A request for the bus is made to the ZT 8801 via BUSRQ* pin 42, and the ZT 8801 responds with BUSAK* once the microprocessor has signaled its release of the bus. When the DMA transfer is complete, the DMA device releases BUSRQ* and the ZT 8801 then responds by releasing BUSAK*. The ZT 8801 supports DMA for all on-board EPROM and RAM. An example of an external DMA controller is the ZT 8950; it may be used to service an STD I/O device with DMA capability to and from the ZT 8801 memory.

Optional Battery Backup

All RAM and the real-time clock may be selectively battery-backed with a 3.6 V, 1 Amp-hour lithium battery, shipped as a standard option with ZT 8801 DOS systems. When DC power falls below 4.75 V, the battery power switches in and remains until power is again at 4.75 V. At the same time, the DCPDN* STD bus signal (pin 6) is driven active (low) to warn other boards in the system of low DC voltage.

DC Power-Fail Detection

DC power-fail detection senses when DC voltage drops below 4.75 V. This signals the board to switch into battery backup mode, as described in the "Optional Battery Backup" section.

Real-Time Clock/Calendar

The real-time clock/calendar on the ZT 8801 is a Dallas Semiconductor DS 1202. It keeps track of seconds, minutes, hours, days, day of the month, month, and year. The clock automatically corrects for leap years and adjusts for months with fewer than 31 days. The optional battery will back up the real-time clock/calendar.

Watchdog Timer

The onboard watchdog timer is a safety mechanism used in applications guarding against errant software. When enabled, the watchdog timer will timeout and cause a system reset unless the application software is operating correctly and strobes the watchdog timer within a defined period of time.

Serial Communications

The ZT 8801 contains one asynchronous RS-232-C communications channel, which is selectable for RS-422/485. This serial port is not PC (COM) compatible, but is supported by Ziatech's STD Device Driver Package (DDP) and Virtual System Console (VSC) under DOS or Virtual Terminal Interface (VTI) under STD ROM. Transmit Data (TxD) and Receive Data (RxD) are available as DCE and DTE on the frontplane. These signals may optionally be configured as two-wire or four-wire RS-485. All RS-232-C and RS-485 drivers are located on board. The RS-232-C drivers make use of a charge pump for RS-232-C voltage levels, allowing for a +5-V-only system.

Counter/Timers

The ZT 8801 has three independent 16-bit counter/timers, each of which can be used as a timer or event counter. The clock frequency driving each of these timers is a 1.19318 MHz oscillator. An optional off-board source can drive the counter/timers via J2.

The six programmable counter/timer modes are as follows:

1. Interrupt on end of count
2. Frequency divider
3. Square wave generator
4. Software-triggered strobe
5. Retriggerable hardware-triggered strobe
6. Retriggerable one-shot

The output of timer 2 is available at connector J2. The "gate" or enable input to timer 2 is pulled up active by a 100 k ohm resistor and may be controlled by a source on frontplane connector J2.

Interrupts

The programmable interrupt controller (PIC) on the ZT 8801 is equivalent to an Intel 8259A. It has eight interrupt inputs that can be prioritized in software. Its output drives the CPU interrupt input. All PIC interrupt inputs may be jumper selected between various on-board sources and the five frontplane and three backplane sources. Factory default assigns the DOS compatible interrupt selections as shown in the "Interrupt Jumper Selection" illustration.

The interrupt structure follows Version 2.3 and later of the *STD-80 Series Bus Specification and Designer's Guide*, which allows for the RESERVED and CNTRL* signals (STD bus pins 37 and 50, respectively) to be interrupt sources as well as INTRQ* (pin 44). These signals are now referred to as INTRQ1*, INTRQ2*, and INTRQ*, respectively. This interrupt structure provides for more backplane interrupts and may eliminate frontplane cabling for additional interrupts.

The STD-80 bus protocol for PIC cascading is also supported, allowing for 8259A interrupt controller expansion. The PIC may handle up to 57 prioritized interrupts by combining seven off-board sources, each of which may support eight interrupt inputs via a separate "slave" interrupt controller, plus one direct on-board source (IR0 is available only on board).

16C49 Digital I/O Port

The ZT 8801 provides up to 48 lines of digital I/O via the 16C49 event sense peripheral interface. Eight of these lines are used for on-board configuration but may be configured for off-board I/O at the expense of some on-board features. The on-board features controlled by the 16C49 include the LED, memory page switching, RS-485 gating, sleep mode, on-board real-time clock, watchdog strobe, and optional control of a daughter board 5-to-12-V adapter for +12 V Flash devices.

SBX Expansion Module Socket

The SBX expansion module socket is provided to expand the I/O capabilities of the ZT 8801 to meet the needs of the application. The expansion module interface is electrically, mechanically, and functionally compatible with the Intel *iSBX MULTIMODULE™ Standard* (IEEE 959-1988). This makes hundreds of off-the-shelf modules available to the STD bus designer. These include functions such as servo and step motor control, analog-to-digital and digital-to-analog converters, serial and parallel I/O, disk and SCSI controllers, and modem control. Ziatech also offers a prototyping board (ZT eSBX70) for custom expansion module designs.

LED

The LED is controlled with an onboard digital I/O point and is available for application use.

5 V Operation

The ZT 8801 requires only +5 V for operation. For single board applications this can simplify the power supply requirements. Onboard devices, such as the RS-232 transceivers, use charge-pump technology to provide the correct voltage levels required.

Clock Shutdown

The ZT 8801 can shut down the SBX and timer-tick oscillators to conserve power.

CHAPTER 2. GETTING STARTED

This chapter includes all the information you need to properly install the ZT 8801 into an STD bus card cage. You should read this chapter and Chapter 3, "Theory of Operation," before you attempt to use the board. Remember, unless specifically stated otherwise, all references to the ZT 8801 also pertain to the ZT 88CT01.

UNPACKING

Please check the shipping carton for damage. If the shipping carton and contents are damaged, notify the carrier and Ziatech for an insurance settlement. Retain the shipping carton and packing material for inspection by the carrier. Do not return any product to Ziatech without a Return Material Authorization (RMA) number. Appendix D, "Customer Support," explains the procedure you should follow to obtain an RMA number from Ziatech.

WHAT'S IN THE BOX?

The items listed below are included in a standard ZT 8801 order. The list does not include options such as system level software or cabling. Refer to the packing list for a complete list of items shipped. When ordering specific system level software options with the ZT 8801, refer to the software manual for a list of the items that should be included.

- ZT 8801 V40 Single Board Computer or ZT 88CT01 V40 Single Board Computer in anti-static bag

Save the anti-static packing material for use in storing or shipping the ZT 8801.

WARNING!

Like all equipment utilizing CMOS devices, the ZT 8801 must be protected from static discharge. This is especially important for the ZT 88CT01, which contains all CMOS logic and is therefore very sensitive to static discharge. Never remove or install any of the socketed parts except at a static-free workstation.

SYSTEM REQUIREMENTS

The following topics cover physical requirements, power requirements, and environmental requirements.

Physical Requirements

The ZT 8801 is designed to be used in an STD bus system. Therefore, it is physically and electrically compatible with the STD-80 bus standard. The board should normally

be mounted in one slot of an STD bus card cage. Refer to the "Card Dimensions" drawing in Appendix B, "Specifications," for board dimensions.

Power Requirements

Power requirements for the ZT 8801 are +5 VDC at 1 A maximum, 620 mA typical. The ZT 88CT01 requires .8 A maximum and 430 mA typical. ±12 V is not needed for either the ZT 8801 or the ZT 88CT01. The serial drivers use a charge pump mechanism to supply RS-232-C compatible signal levels. +12 V is required if +12 V Flash memory is used or if an expansion module daughter board requires +12 V.

> ***WARNING!***
>
> If you are using an emulator in place of the microprocessor on the ZT 8801, the emulator should be powered down or disconnected before the STD card cage is powered down.

Environmental Requirements

The ambient temperature must be maintained at 0° to 65° Celsius for proper operation and to avoid possible damage to the ZT 8801. (The ZT 88CT01 allows for a lower power requirement and a wider temperature range, as detailed in Appendix B, "Specifications"). Relative humidity should be less than 95% at 40°C, non-condensing.

We recommend that you install the board vertically if you are using it in a convective cooling system that is not equipped with a fan. However, most systems require a fan for proper operation. Horizontal mounting is not recommended unless forced air cooling is provided at a rate of 30 cubic feet per minute passing over the surface of the board.

INSTALLING THE ZT 8801

The fastest way to begin using the ZT 8801 is with the addition of development software available from Ziatech. The STD ROM development system allows you to download application software developed on an IBM PC (or equivalent) through a serial port onto the ZT 8801. The personal computer is used as a development station to create, download, and debug applications written in assembly, C, and other high level languages.

Ziatech DOS for the ZT 8801 is intended for designers who wish to develop an application using a high-level language or for systems requiring a resident operating system (to support disk subsystems, for example). Ziatech's DOS is an MS-DOS operating system that resides on the ZT 8801 and is able to run most PC software.

The ZT 8801's serial port is not PC (COM) compatible, but it is supported by Ziatech's STD Device Driver Package (DDP) and VSC under DOS or VTI under STD ROM.

In addition to high level language support, Ziatech DOS includes an extensive base of easily integrated software, such as support for the following:

- EGA and VGA graphics (ZT 8844 and ZT 8980/8981/8982)
- Floppy disks subsystems (ZT 8950 series)
- Fixed disk subsystems (ZT 8952/8953)
- SCSI-2 interface (ZT 8956)
- RAM and EPROM disks (ZT 8825 and on-board ZT 8801)
- Flash memory in-circuit programming (ZT 8825 and on-board ZT 8801)
- Centronics printer interface (ZT 88CT75)
- IEEE 488 (ZT 8847 and ZT 8848)
- Serial (ZT 8840, ZT 8841, ZT 8932 [quad serial], ZT 88CT75, zSBX 32, and on-board ZT 8801)
- Real-time clock (on-board ZT 8801)
- Digital signal processing (ZT 89CT30)
- Motion control (ZT 8931)
- Parallel processing (ZT 8832)
- PCMCIA (ZT 8921)
- GE FANUC interface (ZT 88CT93)

Configuring the ZT 8801 for STD ROM

The STD ROM development system is available as an option to the ZT 8801 for software development. If STD ROM is ordered along with the ZT 8801, the board is preconfigured and tested at the factory prior to shipment. In STD ROM configuration, all 48 lines of I/O are available for application use. If the ZT 8801 and STD ROM are ordered separately, or if the system has been altered or the ZT 8801 rejumpered and the system does not function properly, refer to the "STD ROM (32K ROM/128K RAM) Configuration" illustration in Appendix A, "Jumper Configurations," and check the configuration requirements given in the following topics.

STD ROM Memory Requirements

- The STD ROM debug monitor is shipped in a 32 Kbyte EPROM. Install this EPROM into the 32-pin socket at location 7D, right justified, with the board oriented component side up, goldfingers to the left (see the "STD ROM (32K ROM/128K RAM) Configuration" illustration in Appendix A, "Jumper Configurations").

- STD ROM requires 2 Kbytes of memory, from address 0h through 7FFh, for program use. The 128 Kbyte static RAM shipped with the STD ROM system should be sufficient for both debug and application program memory. Install this RAM into socket location 5D.

- If more application program RAM memory is required, a larger RAM may be installed in 5D, or an additional RAM may be installed in 6D. The ZT 8801 is designed to support four different memory configurations: one 128 Kbyte RAM; two 128 Kbyte RAMs (256 Kbytes); one 512 Kbyte RAM; or two 512 Kbyte RAMs (1 Mbyte). Smaller RAM sizes, such as 32 Kbytes, are redundantly mapped within each 128 Kbytes. PROM sizes are selectable between 128 and 512 Kbytes. Smaller sizes are redundantly mapped.

STD ROM Cable Requirements

- STD ROM requires a serial link between frontplane connector J6 and the IBM PC or compatible. The ZT 90069 cable shipped with the STD ROM system should be used for this purpose. Plug this cable into connector J6 of the ZT 8801. Plug the 25-pin D-shell connector into COM1 of a PC compatible.

- If your PC has a 9-pin serial cable (for example, IBM AT®), then an adapter cable is required for the 9-pin to 25-pin conversion.

STD ROM Jumper Configuration

The following jumper configuration should be used for STD ROM. Refer to the "STD ROM (32K ROM/128K RAM) Configuration" figure in Appendix A, "Jumper Configurations," for an illustration of this jumper configuration. This configuration is for a 32K ROM (not Flash), 128K RAM system. Note that the 32K ROM is redundantly mapped four times within the top 128K of memory.

INSTALL: W1A, 6, 10, 11, 16, 19, 21, 22, 24, 26, 28, 31, 33, 35, 36, 38, 41, 45, 47, 50, 53, 55, 57, 58, 59

REMOVE: W1B, 2-5, 7-9, 12-15, 17, 18, 20, 23, 25, 27, 29, 30, 32, 34, 37, 39, 40, 42-44, 46, 48, 49, 51, 52, 54, 56

STD ROM Operation

Refer to your STD ROM documentation for software installation and operation procedures.

If the system is not working, check the following:

1. Four ZT 8801 frontplane connectors accept the ZT 90069 serial cable. STD ROM expects to be used with J6.
2. If a PC is used and it has more than one 25-pin male connector, be sure the serial cable is plugged into COM1.
3. Make sure the EPROM and RAM chips are installed in the proper sockets. EPROM should be installed in socket 7D. RAM should be installed in socket 5D.
4. Check pin 1 orientation of the installed EPROM and RAM(s). Pin 1 should be to the left, with the board oriented component side up, goldfingers to the left. It should be right-justified in the socket. This leaves four unused pins on the left side of the socket at 7D.
5. Verify that the jumpers are properly installed, particularly those associated with the socket and memory configurations, that is, W30-W37, W40-W42, W18-W21, and W12-W15.

Configuring the ZT 8801 for DOS

Ziatech's DOS is an optional MS-DOS operating system available for the ZT 8801 processor board. If the ZT 8801 and DOS are ordered together, the board is properly configured and tested as a system prior to shipment. If the ZT 8801 and DOS are ordered separately, or if the ZT 8801 was altered in any way after shipment, you can find instructions to install and boot DOS on the ZT 8801 in the Ziatech *Industrial Computer System Manual*. Refer to the "DOS (128K Flash/256K RAM) Default" figure in Appendix A, "Jumper Configurations," for an illustration of the correct DOS jumper configuration. Note that 40 lines of digital I/O are nominally available in a DOS system.

DOS Memory Requirements

The DOS/BIOS software is shipped in one 128 Kbyte Flash memory for installation onto the ZT 8801 at socket location 7D (see the "DOS (128K Flash/256K RAM) Default" figure in Appendix A, "Jumper Configurations"). Install the Flash only at a static-free workstation. Orient pin 1 properly, to the lower left with the board placed component side up, goldfingers to the left.

Ziatech DOS requires at least 128 Kbytes of static RAM, although the DOS system is shipped with 256 Kbytes. The two 128 Kbyte static RAMs are installed in socket locations 5D and 6D, occupying the memory address range from 0 through 3FFFFh. Again, be sure to orient pin 1 consistently with the Flash's pin 1.

DOS requires battery-backed RAM for system configuration variables; the top 32 Kbytes of a 128, 256, or 512 Kbyte system are designated as the R: drive and also retain system configuration information. For 1 Mbyte systems, the system configuration information is contained in the top 384 Kbytes of memory above the 640 Kbyte system RAM. In two-RAM (256 Kbyte/1 Mbyte) systems, socket 6D must be battery backed by installing W19 and removing W18. In single RAM (128/512 Kbyte) systems, socket 5D must be battery backed by installing W21 and removing W20. A 1 Amp-hour lithium battery is shipped with the DOS system to back up the system information and optionally, all RAM on board. It should be securely placed into the battery socket and held fast with double-sided tape underneath.

Note that when W19 is installed, the whole static RAM is battery backed in socket 6D. W21 will battery back all of the RAM in 5D.

DOS Cable Requirements

If the DOS system is a PC-assisted (PCA) system, a serial link is required between the ZT 8801 and a terminal or host PC. Refer to the Ziatech *Industrial BIOS for CompactPCI® and STD 32® Systems* Manual for cable requirements and software information. The ZT 90069 cable is used to connect J6 to the COM port of the host system. If your PC has a 9-pin serial cable (for example, IBM AT), then an adapter cable is required for the 9-pin to 25-pin conversion.

DOS Jumper Configuration

The ZT 8801 as shipped from Ziatech should be properly configured for a DOS system if the board was ordered with the DOS option. The following is a list of the jumpers assigned at the factory prior to shipment. Refer to the "DOS (128K Flash/256K RAM) Default" figure in Appendix A, :Jumper Configurations," for an illustration of this jumper configuration. This configuration is for a 128K Flash, 2x128K (256K) RAM system. Consult the "DOS (512K ROM/1M RAM) Configuration" figure, also in Appendix A, for a 512K ROM/1 Mbyte RAM system configuration.

INSTALL: W1A, 2, 6, 10-12, 14, 16, 19, 21, 22, 24, 26, 28, 31, 33, 34, 36†, 38, 41, 45†, 47, 50, 53, 55, 57, 58, 59

REMOVE: W1B, 3-5, 7-9, 13, 15, 17, 18, 20, 23, 25, 27, 29, 30, 32, 35, 37†, 39, 40, 42-44, 46†, 48, 49, 51, 52, 54, 56

† Install W37 and W46 and remove W36 and W45 for Flash in-circuit program capability. The +12 V power supply must be regulated within 5% (11.4 V to 12.6 V) for proper Flash memory operation.

Powering Up The DOS System

Be sure the ZT 8801 is seated securely in the card cage and the power switch is off. Plug the card cage into a 120 VAC source. Refer to the list below to find the instructions that are appropriate to your configuration. Instructions are provided for the following:

- PC-Assisted with a host computer (An IBM PC or compatible is used to communicate with the ZT 8801 DOS system.)
- PC-Assisted with a video board ("Stand-Alone Operation") (If the ZT 8801 was purchased with a video board, the system may be operated as a stand-alone system with complete keyboard and video support.)
- Automation Engine (for OEM system designers or high volume users of the ZT 8801 DOS system)

PC-Assisted With A Host Computer:

An IBM PC or compatible is used to communicate with the ZT 8801 DOS system.

1. Connect the DOS system's serial cable from the proper connector (J6) on the ZT 8801 to COM1 on the IBM or compatible PC with the ZT 90069 cable.
2. Install the Host Development Software diskette into drive A of your IBM or compatible PC.
3. Type `A:VSC` and press return. The screen should indicate VSC is installed.
4. Press the ALT + SPACE keys to switch to the DOS system screen. The screen should be clear on the DOS system side.
5. Power on the DOS system.
6. The system configuration appears, followed by the "`ZT P:>`" prompt.

If you need further assistance, refer to your Ziatech system manual.

Stand-Alone Operation:

If the ZT 8801 was purchased with a video board, the system may be operated as a stand-alone system with complete keyboard and video support.

1. Configure the video board for the type of monitor desired:
   a. The ZT 8844 EGA video board is shipped configured for a monochrome monitor. If your monitor is not monochrome, refer to the *ZT 8844 Hardware Manual* for alternate jumper configurations.

b. The ZT 8980 and ZT 8981 video boards are shipped configured for VGA and EGA color monitors, respectively. Refer to your hardware manual for specific jumper configurations.

c. Be sure the video board is installed into the card cage along with the ZT 8801 and the cables to the display and keyboard are attached.

2. Power on the system. You should see the system configuration list and a "`ZT P:>`" prompt on your display. Operation is then identical to that of a PC-compatible type computer.

Automation Engine:

The Automation Engine is available for OEM system designers or high volume users of the ZT 8801 DOS system. The ZT 8801 is shipped with a license for MS-DOS and it is used for systems with completed application software. It is assumed here that the user is familiar with the ZT 8801 DOS system.

MEMORY ADDRESSING

The following eight illustrations show the memory addresses occupied by the ZT 8801 for DOS and STD ROM systems:

512K ROM/1M RAM System Memory Map

512K ROM/512K RAM System Memory Map

512K ROM/256K RAM System Memory Map

512K ROM/128K RAM System Memory Map

128K/256K ROM-1M RAM Memory Map

128K/256K ROM-512K RAM Memory Map

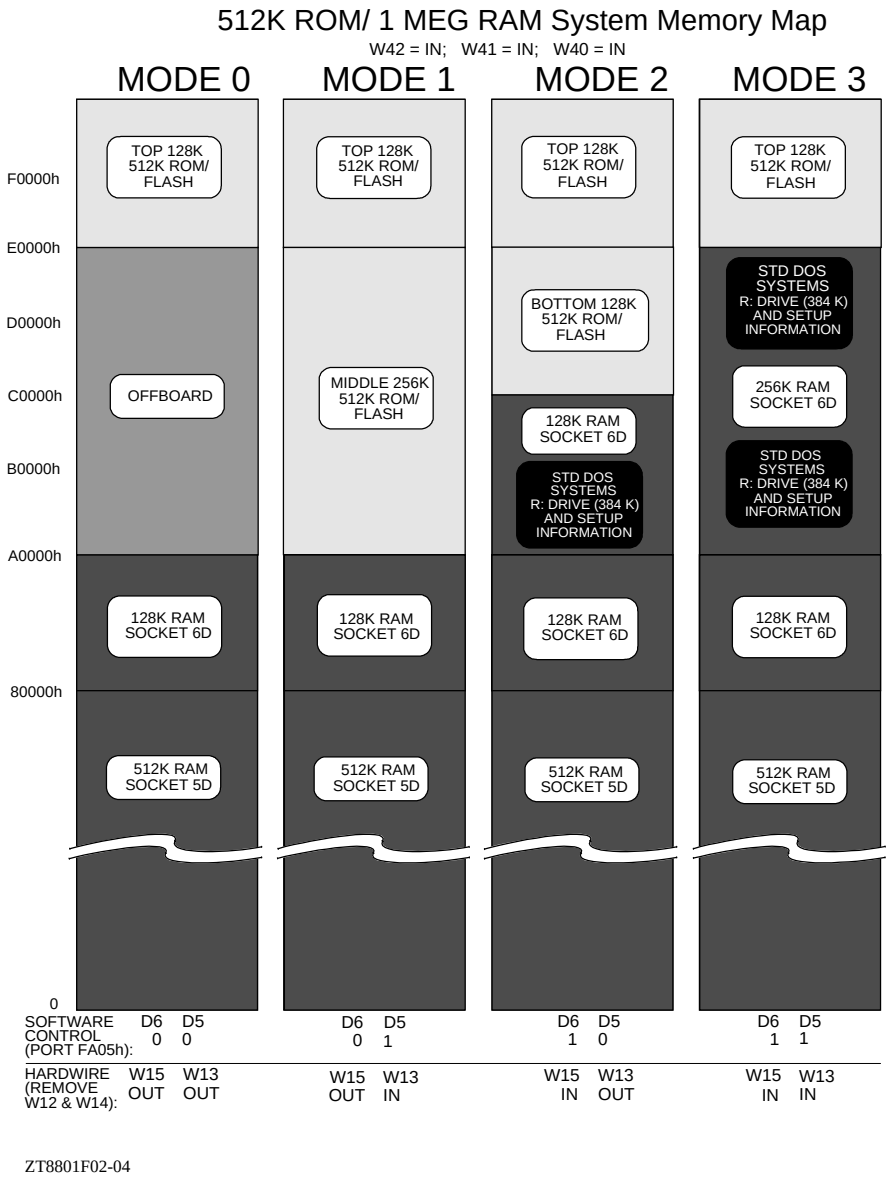
128K/256K ROM-256K RAM Memory Map

128K/256K ROM-128K RAM Memory Map

Access to on-board memory and the backplane is through the full 20-bit memory address, allowing for 1 Mbyte of memory in the system. On-board memory consists of three 32-pin byte-wide sockets. One of these sockets accepts from 32 to 512 Kbyte EPROM/Flash and the other two sockets accept either 128 or 512 Kbyte RAMs. All RAM may be battery backed. The EPROM space is selectable between 128 and 512 Kbytes.

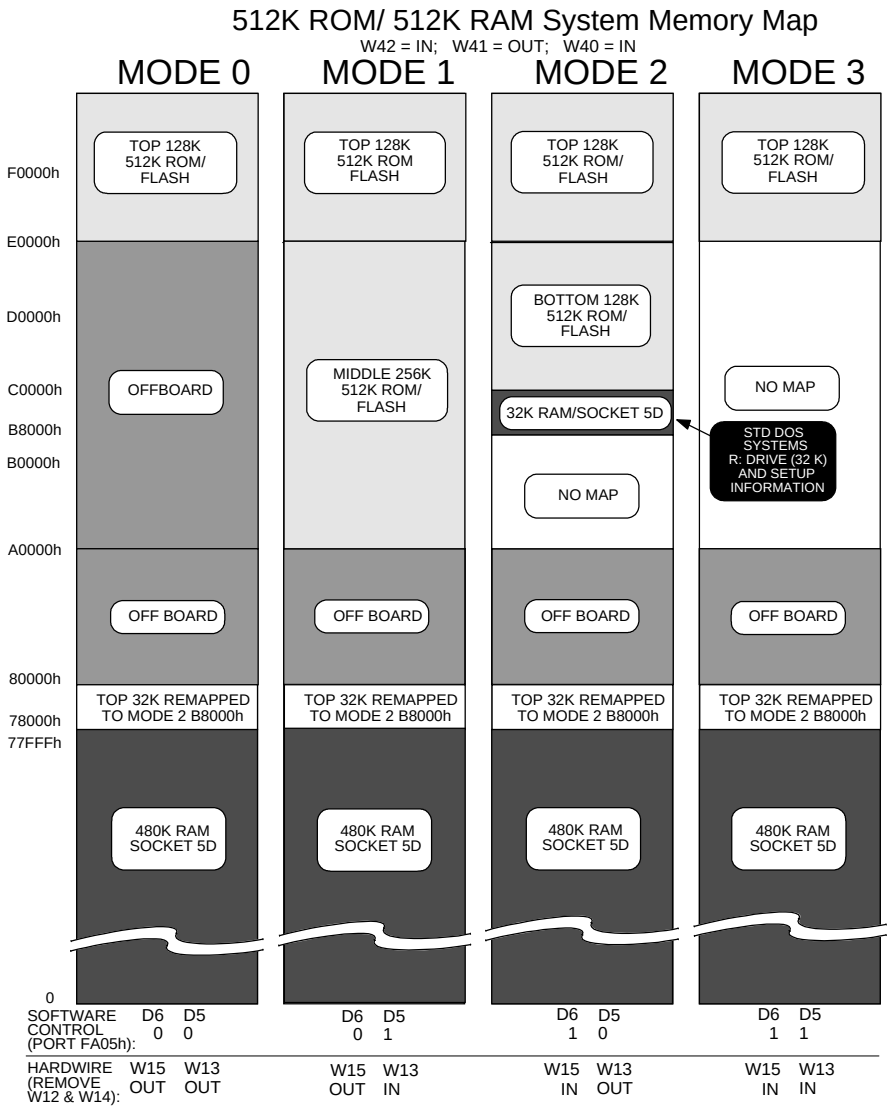
Memory access times require the use of 200 ns parts or faster for both RAM and ROM accesses.

The ZT 8801 has four memory pages that are either software selectable or hardwired. This allows up to 1 Mbyte and 512 Kbyte of EPROM/Flash to be mapped within the 1 Mbyte address space of the V40. Each of these pages is referred to as a MODE. The memory map has also been optimized for one of four RAM memory configurations: 128 Kbytes (one 128 Kbyte part), 256 Kbytes (two 128 Kbyte parts), 512 Kbytes (one 512 Kbyte part), and 1 Mbyte (two 512 Kbyte parts). It is possible to use smaller parts (such as 32 Kbyte RAMs) when configuring the board for 128 Kbyte parts as a method to reduce costs. Using smaller parts causes the part to be redundantly mapped within the 128 Kbyte memory space. If you are using DOS, do not use RAMs smaller than 128 Kbytes.



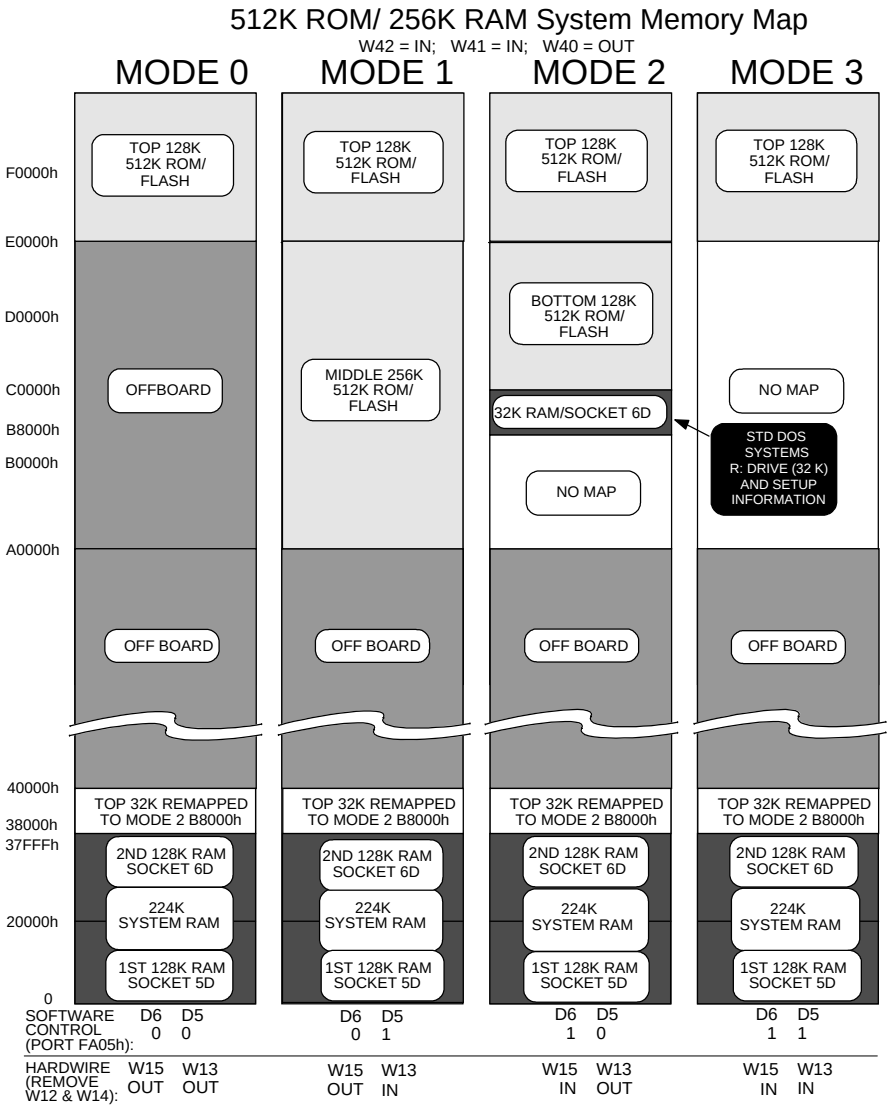
ZT8801F02-04

512K ROM/1M RAM System Memory Map



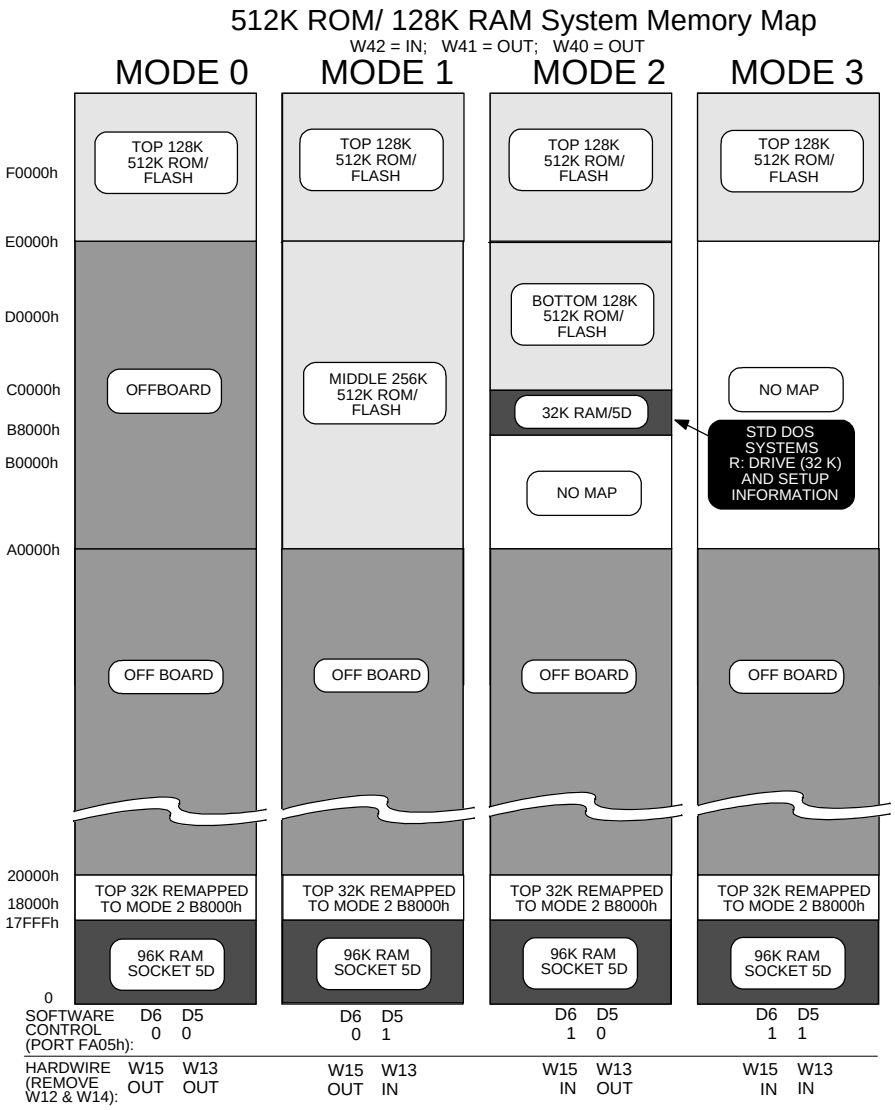
ZT8801F02-05

512K ROM/512K RAM System Memory Map



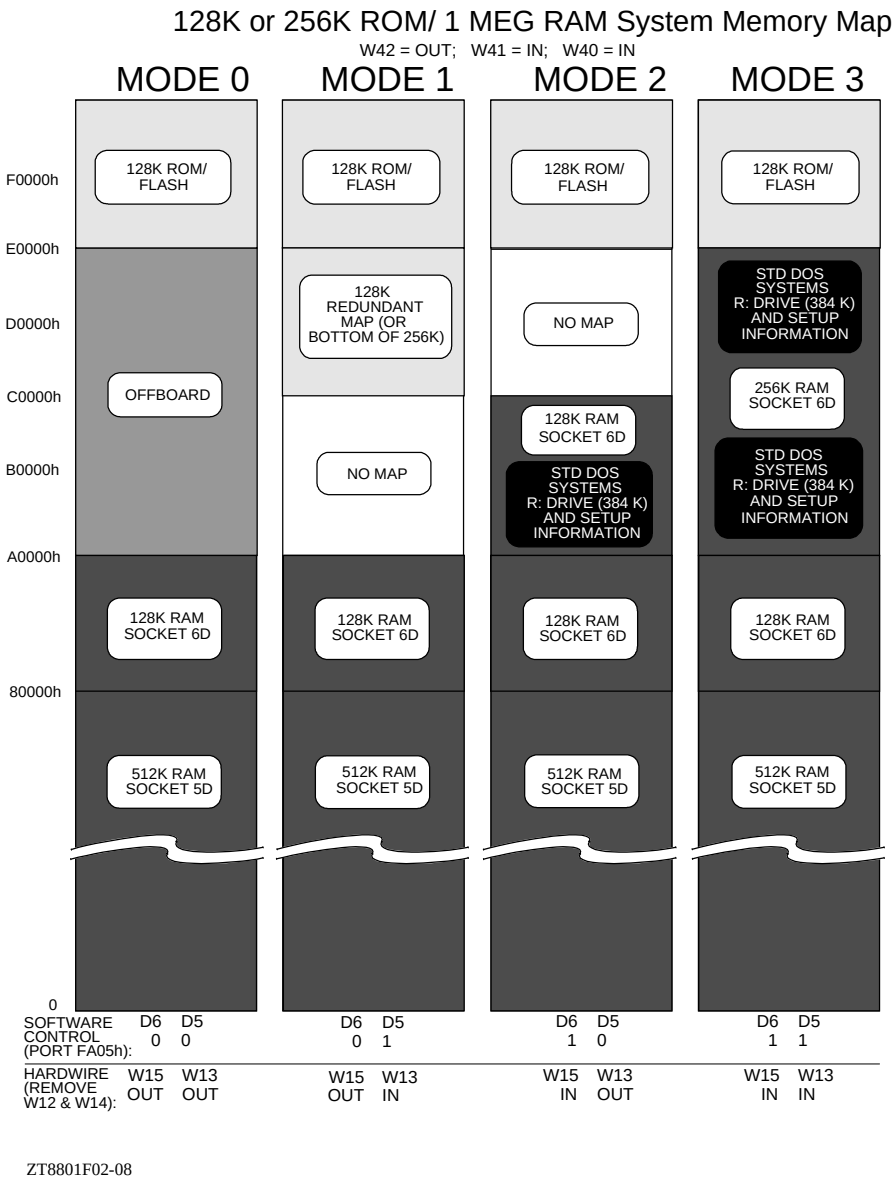
ZT8801F02-06

512K ROM/256K RAM System Memory Map



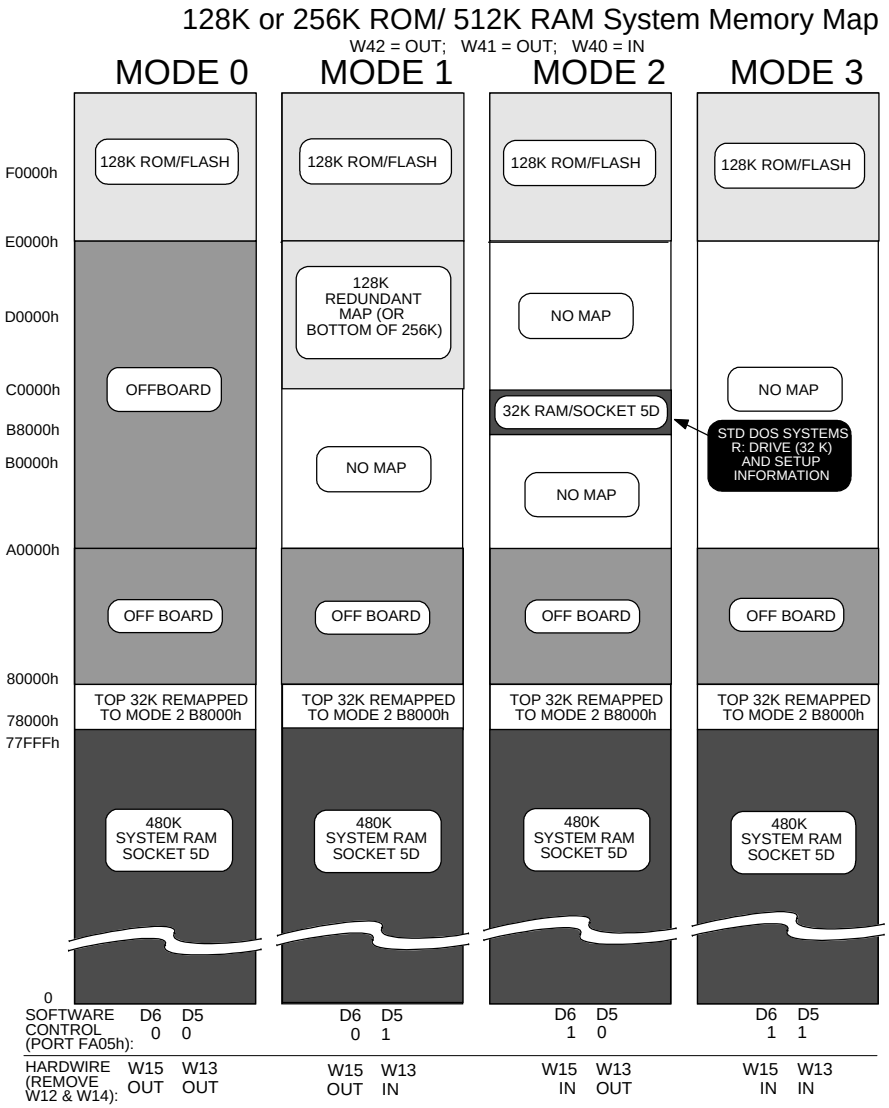
ZT8801F02-07

512K ROM/128K RAM System Memory Map



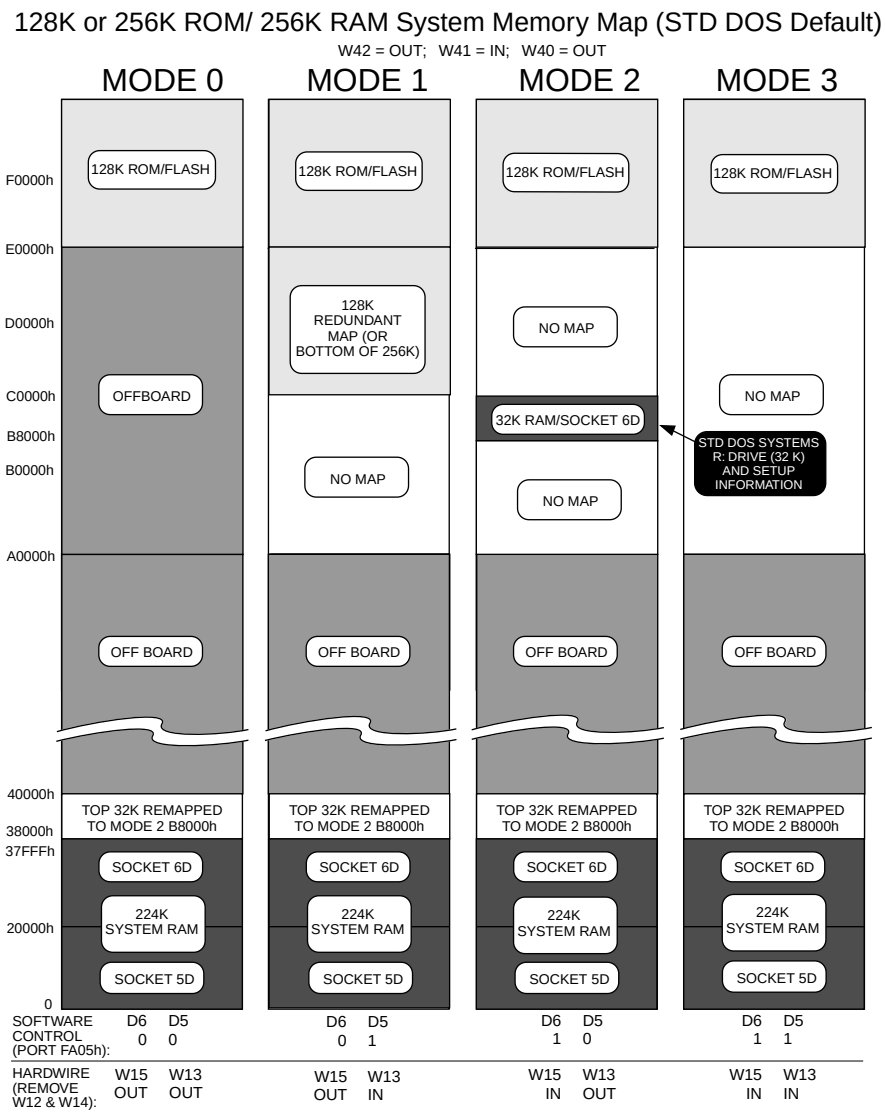
ZT8801F02-08

128K/256K ROM-1M RAM Memory Map

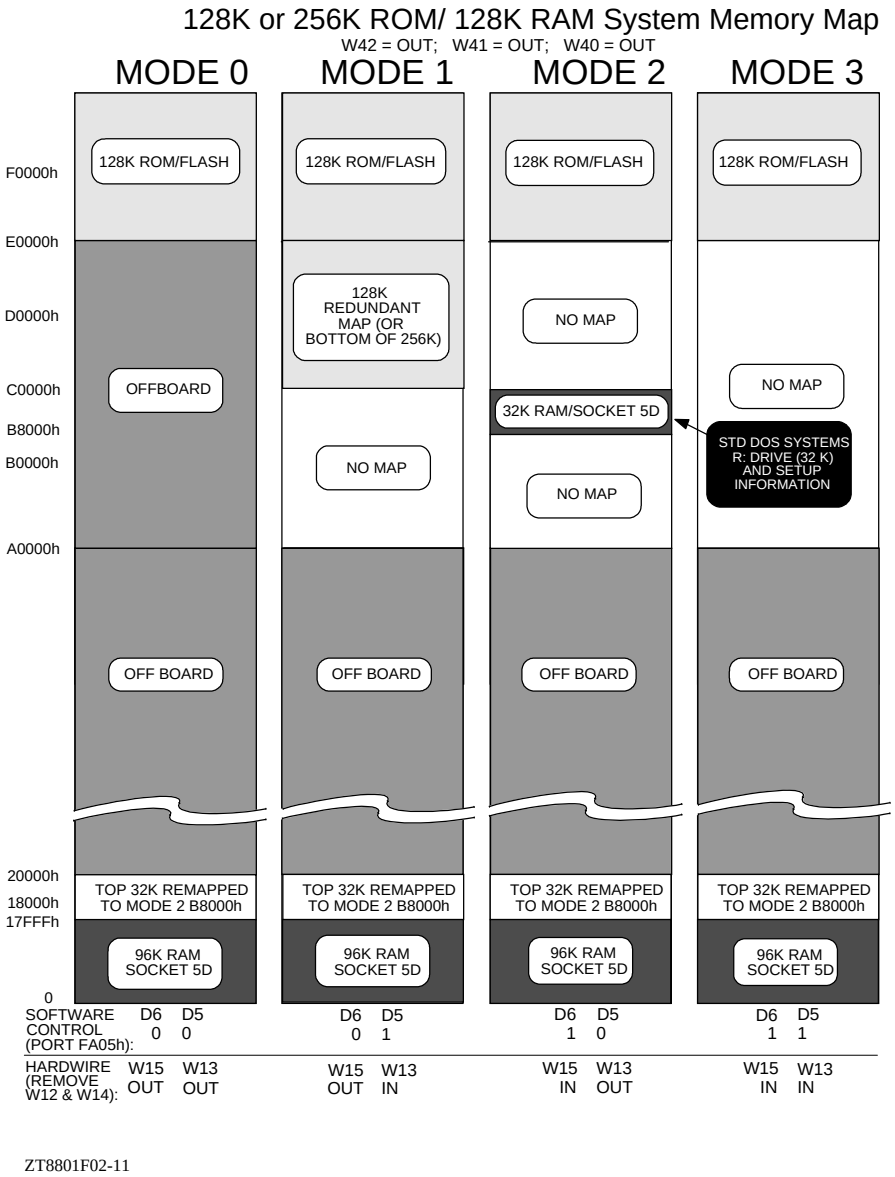


ZT8801F02-09

128K/256K ROM-512K RAM Memory Map



128K/256K ROM-256K RAM Memory Map



128K/256K ROM-128K RAM Memory Map

Memory Modes

The four memory modes (MODE 0 - MODE 3) allow the ZT 8801 to bank switch extra RAM and EPROM/Flash into the 256 Kbyte memory space between A0000h and DFFFFh. Under software control, two parallel port bits are used to select the MODE in which the memory is configured. Bits 5 and 6 of I/O address 0FA05h are used to select the mode. 00b written selects MODE 0; 01b selects MODE 1; 10b selects MODE 2; and 11b selects MODE 3. Other bits are used within this register, which requires that software read the port before writing a new mode to preserve the other bits. We also recommend that you disable interrupts (CLI) before modifying this register. See the "Control Port" section in Chapter 3, "Theory of Operation," for further details.

In all modes, the top 128 Kbytes of EPROM/Flash are always accessible, as is any system or application memory below A0000h.

The memory map figures in this chapter show the memory modes used for the four memory configurations.

MODE 0 In MODE 0, memory accesses to A0000h-DFFFFh go off-board. Note that in DOS systems this memory address range is where video boards and other memory mapped devices are located. MODE 0 is the default mode at power up.

MODE 1 MODE 1 is used when a 512 Kbyte EPROM is installed to reach the middle 256 Kbytes of the EPROM at addresses A0000h-DFFFFh. In this mode, a total of 384 Kbytes of EPROM space exists from A0000h through FFFFFh. Below A0000h, system RAM is available up to the amount of RAM installed.

MODE 2 MODE 2 is used when a 512 Kbyte EPROM/Flash is installed to access the bottom 128 Kbytes of the part, which is accessible between C0000h through DFFFFh. In MODE 2, the top 128 Kbytes of the second 512 Kbyte RAM in a 1 Mbyte system are accessible at addresses A0000h through BFFFFh. Note that in MODE 2 the top 128 Kbytes of the 512 Kbyte EPROM are still accessible between E0000h through FFFFFh.

MODE 3 MODE 3 is used when two 512 Kbyte RAMs are installed in order to access the middle 256 Kbytes of the second RAM. This memory is accessible between A0000h through DFFFFh. MODE 2 is used to access the top 128 Kbytes of the second RAM.

Mode Hardwiring

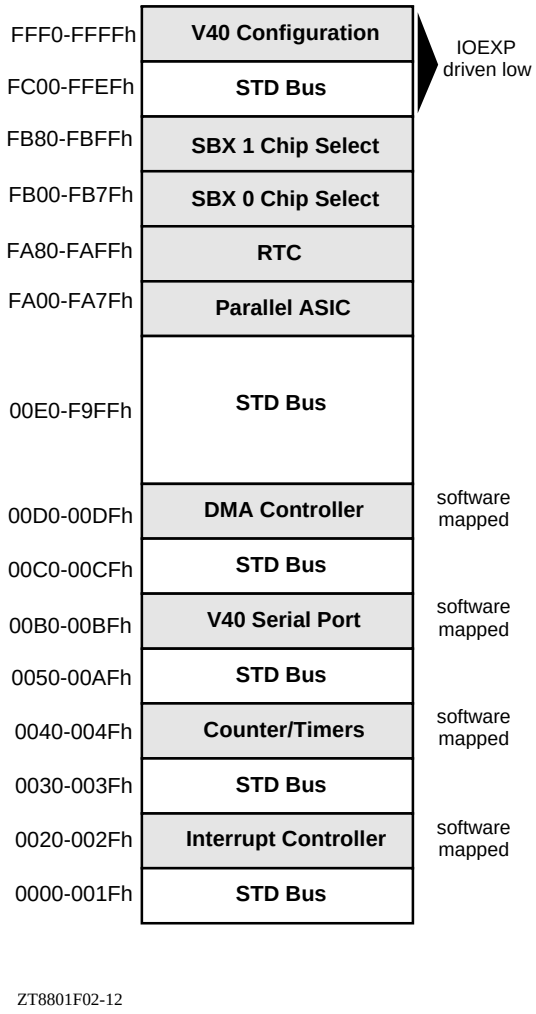
As an alternative to using bits 5 and 6 of I/O address 0FA05h to select the memory mode, the ZT 8801 can be hardwired into one of the modes via jumpers W12-W15. This frees the two parallel port bits for other uses. To hardwire the board into a particular mode, remove W12 and W14. W13 and W15 are then used to select the mode. Removing W13 and W15 forces the board in MODE 0. Installing only W13 forces the board in MODE 1. Installing only W15 forces the board in MODE 2. Installing both W13 and W15 forces the board to always be in MODE 3. DOS users must allow software control over the mode bits and cannot hardwire to a particular mode. The default for STD ROM users is to remove W12-15, forcing the board to always be in MODE 0.

I/O ADDRESSING

The "I/O Map" illustration shows the I/O addresses occupied by the ZT 8801 for DOS and STD ROM systems.

I/O accesses are made via the full 16-bit I/O address, allowing for 64 Kbytes of I/O addresses. Eight-bit addressed I/O boards are also compatible with the ZT 8801, provided they decode the IOEXP signal on the backplane low when generating board

select. IOEXP is driven low only when I/O addresses FC00h through FFFFh are accessed. Eight-bit addressed boards that decode IOEXP low are then accessed only within the FC00h-FFFFh range and are not redundantly mapped throughout the I/O map. For example, an 8-bit board that decodes an address of 80h, as well as decodes IOEXP low, is accessible from the ZT 8801 at address FC80h, but not at 80h, which keeps the I/O map below FC00h free for other 16-bit decoders.



I/O Map

FRONTPLANE CONNECTORS

The following table lists the assignments of frontplane connectors J1-J8. Refer to "Connectors" in Appendix B for an illustration showing connector locations and for information on pin assignments and cabling.

Frontplane Connector Assignments

Connector	Function
J1	Interrupts
J2	Counter/Timers
J3	Serial Channel Transmit
J4	RS-485 Multidrop
J5	RS-232-C DTE
J6	RS-232-C DCE
J7	Parallel I/O
J8	SBX Expansion Module Interface

JUMPER OPTIONS

The ZT 8801 includes many jumper options to tailor the operation of the board to the requirements of specific applications. Refer to Appendix A, "Jumper Configurations," for a full description of configurable jumpers.

CHAPTER 3. THEORY OF OPERATION

This chapter begins by answering commonly asked questions about the ZT 8801 and ZT 88CT01. It then describes the following system level issues:

- Processor performance compared to the IBM PC XT®
- STD bus compatibility
- STD bus memory and I/O transfers
- Direct Memory Access (DMA) support and benefits
- I/O operation, including masking and event sense
- Serial communications using RS-232-C or RS-422/485
- Expanding the ZT 8801 interrupt structure
- Processor reset
- Hardwiring vs. software control through the control port
- Methods of battery backup with DC power-fail detection
- Battery life
- Status indicator access (LED)
- CMOS version available for extended temperature and low power

COMMONLY ASKED QUESTIONS

1. **What are the differences between the ZT 8801 and ZT 8901 STD 32 processor boards?**

   The ZT 8901 is designed with the NEC V53 processor running at 16 MHz. The ZT 8801 is designed with an 8 MHz V40 processor. The ZT 8901 can perform 16-bit transfers on the STD 32 bus, while the ZT 8801 has only 8-bit transfer capability. Both boards have 48 points of Opto 22 compatible I/O. The connector for this I/O is in the same physical location on both boards to allow for modular system integration.

   The ZT 8901 has two DOS COM channels that the ZT 8801 does not have. The ZT 8801 has one 8251 style serial channel that is configurable as RS-232-C, RS-485, or RS-422.

2. **What are the differences between the ZT 8801 and ZT 8802?**

   The ZT 8802 is a superset of the ZT 8801, adding two DOS compatible COM ports for serial interfacing. The ZT 8801 has RS-485 support for the V40 serial port; the

ZT 8802 does not. The ZT 8802 uses surface mount technology, but the mechanical positions of the digital I/O connector and the SBX connector are identical.

3. **Is the ZT 8801 compatible with the mechanical specification of the STD bus?**

   Yes. The ZT 8801 uses no extensions or porches to increase the board size.

4. **Is the V40 pin-compatible with the 80188?**

   The V40 and 80188 are not pin-compatible. This means an 80188 cannot be plugged into the V40 socket. This is unlike the V20 and V30, which are interchangeable with the 8088 and 8086, respectively.

5. **What are the hardware differences between the V40 and the 80188?**

   One of the most notable differences is that the V40 is fabricated with a CMOS process. CMOS technology provides an increase in both temperature range and noise immunity with a reduction in power consumption. The CMOS V40 has a maximum power dissipation of less than ½ W and a standby dissipation of less than 1/10 W. The 80188 has a maximum power dissipation of 3 W; that is, too hot to touch.

   Other differences are as follows: the V40 includes an asynchronous serial port; the V40 interrupt controller and counter/timers have the same architecture as the interrupt controller and counter/timers used in the IBM Personal Computer; and the data transfer rate for the V40 DMA controller is twice as fast.

6. **Is the V40 software compatible with the 8088 and 80188 microprocessors?**

   Yes. The V40 instruction set is 100% object code compatible with the 80188 instruction set. This means a program written for the 8088 or 80188 will execute on the V40. Application software using the peripherals internal to the 80188 requires some modification.

7. **What are the V40 instructions not in the 8088 or 80188 and how are they used?**

   The V40 instruction set is a superset of the 8088 and 80188. This means the V40 includes all of the instructions found in these microprocessors plus a few more. The added instructions are outlined below.

   The following instructions are useful in testing and setting status bits for I/O operations and in bit manipulation for graphics applications.

INS	Insert bit field
EXT	Extract bit field
TEST1	Test bit
SET1	Set bit
CLR1	Clear bit
NOT1	Complement bit

The instructions shown below are useful for manipulating binary numbers in a decimal format.

ADD4S	BCD string addition
SUB4S	BCD string subtraction
CMP4S	BCD string comparison
ROL4	Rotate BCD digit left
ROR4	Rotate BCD digit right

The string I/O instructions shown below can be combined with the repeat prefixes for high speed data transfers between I/O and memory.

INM	String input
OUTM	String output

Other instructions not found in the 80188 instruction set are listed below.

REPC	Repeat while carry set
REPNC	Repeat while carry cleared
FPO2	Floating point operation 2
BRKEM	Break for emulation mode
RETEM	Return from emulation mode

RELATIVE MICROPROCESSOR PERFORMANCE

The "Norton Utilities System Information Version 6.0" was used to measure the ZT 8801 processor performance relative to that of the IBM PC. The test compares several processing tasks; test results are presented in the "Processor Speed Comparison" table below. These benchmarks assume one (1) I/O wait state and zero (0) on-board memory wait states.

Processor Speed Comparison

Test	ZT 8801 Average Test Score Relative to 4.77 MHz PC/XT
Processor Speed Benchmark	1.9
Disk Speed (ZT 8952)	4.5
Overall Performance Index	2.7

STD BUS COMPATIBILITY

The ZT 8801 is fully compatible with Revision 2.3 of the *STD-80 Series Bus Specification and Designer's Guide* (Ziatech document number ZT MSTD80). This revision of the bus specification includes definition of two new backplane interrupt request signals, INTRQ1* and INTRQ2*, which replace the signals RESERVED and CNTRL*, respectively. Prior to Revision 2.3, only one backplane interrupt request INTRQ* existed, requiring frontplane cabling or sharing of the interrupt if more than one backplane interrupt was in the system. DOS uses INTRQ1* for the keyboard interrupt request from the ZT 8980 Super Video Graphics Adapter (VGA) card. Refer to the "Interrupts" section of this chapter for details on these interrupts.

STD INTERFACE

The ZT 8801 performs 8-bit I/O and memory transfers to the STD bus whenever an off-board address is decoded for either cycle type. The ZT 8801 performs a read or write transfer in four STD bus clock cycles, in accordance with the *STD-80 Series Bus Specification and Designer's Guide*, Revision 2.3. STD bus peripherals that insert wait states are accommodated by programming the V40 for one internal wait state. This allows the off-board peripheral to generate additional wait states should they be needed.

The following topics discuss STD bus memory transfers, STD bus I/O transfers, IOEXP, and MEMEX (BHE*).

STD Bus Memory Transfers

The ZT 8801 accesses off-board memory whenever the address generated is not decoded as an on-board memory location. The V40 drives only 20 bits of address on memory transfers; the ZT 8801 drives an additional four address lines (A20-A23) for a total of 24 bits of address. A20-A23 are driven to logical 0 for off-board accesses.

STD Bus I/O Transfers

The ZT 8801 accesses off-board I/O whenever the address generated is not decoded as an on-board I/O location. The ZT 8801 supplies a full 16-bit I/O address for off-board transfers.

IOEXP

The STD-80 bus specification never clearly defined IOEXP usage. The STD 32 bus specification defines that IOEXP be driven low in the I/O address space of FC00h-FFFFh by the current bus master. With this mechanism, boards that decode only 8 bits or 10 bits of address and that can use IOEXP in their board select logic can be logically mapped in this range (FC00h-FFFFh). IOEXP is used as an extra address bit in this

case and keeps the 8- or 10-bit address board from being redundantly mapped in the lower I/O space.

Even though the ZT 8801 is not an STD 32 processor, this definition for IOEXP is used to ease system integration.

MEMEX (BHE*)

The MEMEX signal is passively pulled high by the ZT 8801 or it may be optionally tied to ground via jumper W52. STD 32 peripheral boards interpret this signal as Byte High Enable (BHE*) during 16-bit transfers. When this signal is low, STD 32 peripherals that dynamically support 16-bit transfers use this signal to gate the upper data bits of the transfer onto D8-D15. For this reason, we recommend that W52 not be installed when interfacing to STD 32 peripherals. In this mode, BHE* is always high.

DIRECT MEMORY ACCESS (DMA)

The ZT 8801 supports Direct Memory Access (DMA) transfers between local memory and STD bus system memory or I/O under the supervision of an STD bus DMA controller, such as the ZT 8950. The following discussion covers the advantages and operation of an STD bus DMA controller with respect to the CPU.

The following topics discuss the advantages of DMA, and DMA operation.

Advantages of DMA

The use of DMA can greatly increase performance in a system where block transfers of memory or I/O data are often performed. For example, if a disk subsystem is present on the STD bus, large data transfers from the disk controller to system memory and vice versa frequently take place. The use of DMA can allow these transfers without requiring CPU time, thus improving system throughput.

The CPU normally performs data transfer. In a memory to I/O transfer, the CPU must first fetch the memory read instruction from program memory, read the data from memory, temporarily store the data, fetch the I/O instruction from program memory, and then write the data to I/O. This process becomes very time-consuming when moving large blocks of data, as is required for disk I/O.

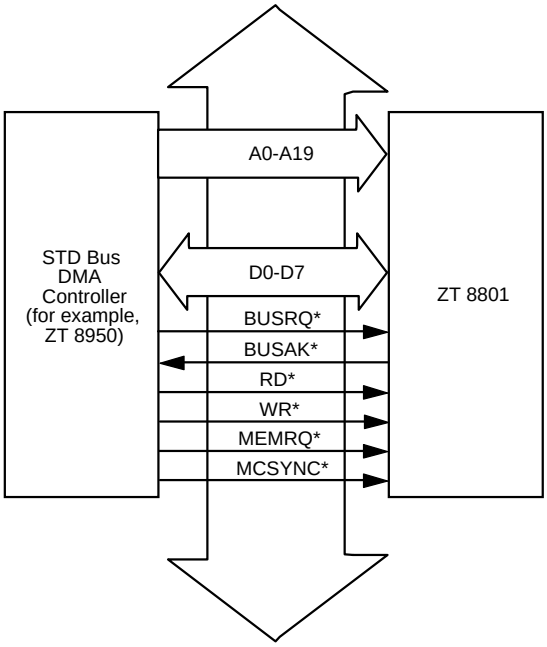
The DMA controller speeds up this operation because the controller contains source and destination address counters that eliminate the need to fetch program instructions; data transfers are performed sequentially under the supervision of the controller. A single cycle transfer enables the data to move from the source onto the bus and to the destination without temporarily storing it.

DMA Operation

The "DMA with STD Bus Controller" illustration shows the interface between the ZT 8801 and an STD bus DMA controller, such as the ZT 8950. The signals shown are required for proper operation of devices on the STD bus during DMA cycles.

The DMA cycle is initiated when the controller asserts the bus request signal BUSRQ* on the STD bus. The ZT 8801 responds to this request by waiting for the CPU to finish the current instruction, then acknowledges the release of the bus by asserting the bus acknowledge signal BUSAK*. Simultaneously, the ZT 8801 turns the address buffers inward to allow access to on-board memory by the DMA controller. The controller is then free to transfer data between an STD bus board and the ZT 8801 or between two STD bus boards. Note that in the case of the ZT 8950, the I/O device may be a separate board. The ZT 8950 has two frontplane connectors for external DMA devices in addition to the on-board (ZT 8950) floppy disk controller.

The STD bus DMA controller must meet the timings for read and write cycles as defined by the STD 80 bus specification.



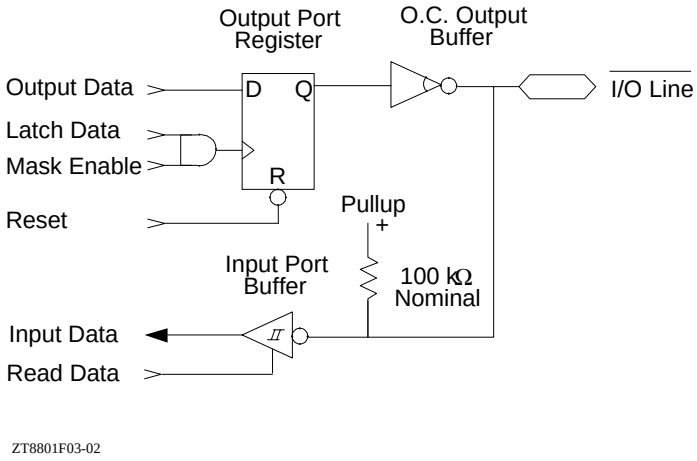
ZT8801F03-01

DMA With STD Bus Controller

I/O POINTS

The 48 input/output points on the ZT 8801 are implemented in one Ziatech 16C49 Parallel Interface Adapter (PIA). The PIA has 48 I/O points in six groups (ports) of eight I/O points each. Each I/O line is illustrated in the "Typical I/O Circuit".

Each typical I/O circuit consists of an output register, open collector output buffer, pullup, and input buffer. Input and output operation are described in the following topics. Applications requiring pullups at the I/O interface should provide appropriate external termination. The internal pullup provided by the 16C49 should not be used for this purpose.



Typical I/O Circuit

Input/Output Operation

Each 16C49 I/O circuit is capable of outputting data, outputting data with readback, and inputting data, as described in the following topics.

Outputting Data

Outputting data is done by writing the output data to a given port, thereby causing the latch data signal to capture the output data into the output port register. The register output is buffered by an inverting open collector output buffer before driving the output signal. An integral pullup resistor ensures that a valid high can be measured when the output is not sinking current while in the de-asserted (off) state. **Note:** Each output port has a mask enable bit that can prevent inadvertent writes. The mask enable bits are controlled from the mask port and are unmasked to allow writes after power up or reset.

Outputting Data With Readback

Outputting data with readback can be accomplished by outputting data as described in the "Outputting Data" section and then reading the input data, thereby causing the read data signal to enable the input port buffer.

Inputting Data

Inputting data is accomplished by reading the input data from a given port, thereby causing the read data signal to input data from the inverting input port buffer. An integral pullup resistor ensures that a valid input is read if the input signal is not connected.

When inputting data, the associated circuit must not be used as an output. This allows the output buffer to be inactive, thereby not contending with the input signal. The output circuits are inactive when a 0 is output, or after reset or power up.

When using a port configured with some output and some input circuits, care must be taken to ensure that any circuits used as inputs are always written with a 0.

Masking Operation

A separate mask port is provided to prevent inadvertent writes to the individual I/O ports. Not required for normal use, this optional port can help ensure system integrity if the system software were to accidentally do I/O port writes. Power up or reset leaves the mask port enabled for I/O writes.

The "Mask Port" table illustrates the data bit and mask port relationships.

Mask Port

Port	D7	D6	D5	D4	D3	D2	D1	D0
5A07h (write)	E7-4	E3-0	Port 5	Port 4	Port 3	Port 2	Port 1	Port 0
5A07h (read)	Int	0	Port 5	Port 4	Port 3	Port 2	Port 1	Port 0

The upper two bits in the mask register are used in conjunction with the event sense port. When writing, the upper two bits of the mask register select the polarity sensed by event sense inputs E0-E7, which are connected to Port 0 (J7 pins 1-8 for E0-E7, respectively). Bits 7 and 6 determine E7-4 and E3-0, respectively. Writing a 0 (power-up default) senses negative events (edges), while writing a 1 senses positive events.

When reading the mask register, the most significant bit (D7) returns the interrupt signal status on the interrupt output pin of the 16C49. A logical 1 means interrupt is asserted.

Event Sense Operation

Eight event sense inputs on the 16C49 PIA are connected in parallel to I/O points I/O 0-7. The event sense circuit is shown in the "Typical Event Sense Circuit" illustration.

The event clear register has eight data bits corresponding to each of the eight event sense flip-flops. To enable operation, set the appropriate bit to a logical 1 (enable). Depending on the polarity selected via the mask register, a positive or negative transition on the I/O point clocks the event sense flip-flop. The event sense flip-flop status can be read via the event status register.

All eight event sense flip-flop outputs are ORed together to generate the hardware interrupt signal. Refer to Chapter 12, "Parallel I/O Adapter (16C49)," for additional information on programming for event sense operation.

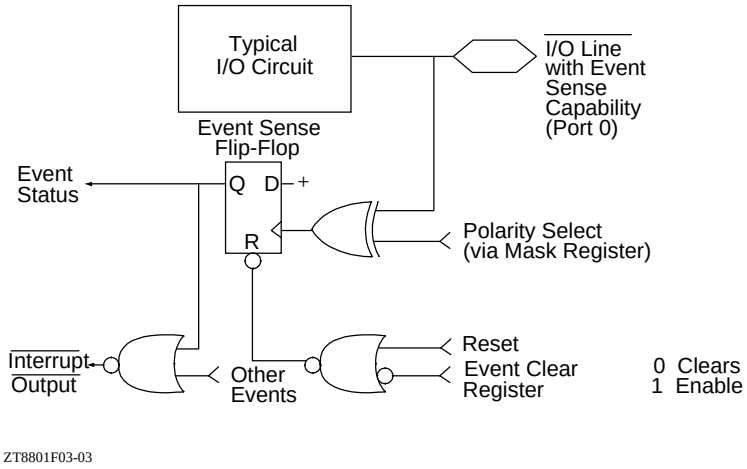
The "Event Sense Port" table illustrates the data bit and event sense relationships.

Event Sense Port

Port	D7	D6	D5	D4	D3	D2	D1	D0
FA06h	E7	E6	E5	E4	E3	E2	E1	E0

When reading the event sense port, each bit set to a logical 1 indicates an event has occurred on that input.

When writing the event sense port, each data bit written with a logical 0 clears its corresponding event sense flip-flop. Each data bit of the event sense port must be written with a 1 to re-enable the corresponding event sense input after it is cleared or after power up or reset.



Typical Event Sense Circuit

Electrical Specifications

The I/O line electrical specifications are listed in "I/O Line Electrical Specifications."

I/O Line Electrical Specifications

Parameter	Specification
Output Sink Current (Iol)	12 mA min.
Output Low Voltage (Vol at Iol)	.4 V max.
Internal Pullup (nominal)	100 kΩ typ.

Reset Operation

Each I/O circuit on the PIA is automatically reset by the reset circuit on the ZT 8801. The precision nonglitching reset circuit on the ZT 8801 prevents the output circuits from glitching on power up or power down. The reset circuit is active when the supply voltage is within 0-4.75 V.

After reset, the mask register is initialized to enable (unmasked) writing to the I/O registers and the event sense status register is cleared. All outputs are disabled (high) until written.

Power Supply Lines

When interfacing to an Opto 22 or equivalent I/O rack, the I/O rack requires +5 V to operate. The I/O rack typically has a two-position terminal block to which power can be supplied, or power can be supplied through the cable connected to the ZT 8801 via the optional

ZT 2225 Industrial I/O Cable Adapter. Refer to the "ZT 2225 Industrial I/O Cable Adapter" section in Chapter 12, "Parallel I/O Adapter (16C49)," for details on interfacing to I/O racks with the ZT 2225 and associated cables.

The +5 V power is available on pins 54-56 of J7. This power is protected by a 1 A miniature fuse. The fuse location on the ZT 8801 is designated F1. Replacement fuses can be purchased from Littelfuse, part #255-001.

Connectors

The ZT 8801 has one header designated J7 for 48 I/O points. Each I/O line on the ZT 8801 is connected to this ribbon cable compatible header. +5 V and ground are available at this interface for powering Opto 22 interfaces. The ZT 2225 Industrial I/O Cable Adapter can be used to make a transition from this 56-pin header to an Opto 22 compatible interface. Refer to

the "ZT 2225 Industrial I/O Cable Adapter" section in Chapter 12, "Parallel I/O Adapter (16C49)," for additional details, and to the J7 pinout in Appendix B, "Specifications," for J7 pin assignments.

Ziatech also offers the ZT 2226 24-Channel I/O Mounting Rack, which can be connected directly to the ZT 8801. It holds up to 24 I/O modules and can be panel mounted using its integral mounting hardware. A second ZT 2226 or other 24-channel I/O module mounting rack can be daisy-chained to the first ZT 2226 to provide a total of 48 I/O module positions.

In situations where it is impossible or undesirable to use the ZT 2226, Ziatech recommends the ZT 2223 Industrial I/O Adapter Board. The ZT 2223 provides many of the same features found on the ZT 2225, but allows the ZT 8801 to be cabled directly to non-Ziatech racks.

Combining a ZT 8801 with either I/O device can reduce the size of a control system and free up valuable enclosure space. See the "I/O Module Mounting Racks" section in Chapter 12, "Parallel I/O Adapter (16C49)," for details.

SERIAL COMMUNICATIONS

The ZT 8801 has one asynchronous RS-232-C serial port, which is also configurable for RS-422/485 use. This serial port, the V40 serial port, supports transmit data (TxD) and receive data (RxD) only. It is not PC (COM) compatible, but is supported by Ziatech's STD Device Driver Package (DDP) and VSC under DOS or VTI under STD ROM. Applications requiring COM port compatibility must use an external COM port such as is available on the ZT 88CT75. J5 supplies Data Terminal Equipment (DTE) compatible signals when used with the ZT 90069 cable assembly. J6 supplies Data Communication Equipment (DCE) configuration. This requires no jumper changes when switching between configurations. The ZT 90069 cable assembly may be used to interface between either of these connectors and an external serial channel. This cable assembly interfaces the 3-pin serial interface to a standard 25-pin D-shell style connector.

Serial Standards

Four Electronic Industries Association (EIA) standards are used for most serial communications. The major differences between the four standards are shown in the "Serial Communication Standards" table. The RS-232-C standard is designed for low data transfer rates between a single transmitter and a single receiver over short distances. The RS-423-A standard increases the number of receivers and the transmission length but because of a single-ended implementation, still has low data transfer rates. The RS-422-A standard implements a differential transmission medium to support all the features of RS-423-A and a much higher transfer rate and is supported by the ZT 8801. The RS-485 standard, also supported by the ZT 8801, is similar to RS-422-A with the added ability to handle multiple drivers and many more receivers.

Serial Communication Standards

Parameter	RS-232-C	RS-423-A	RS-422-A	RS-485
Operation	Single-ended	Single-ended	Differential	Differential
Number Of Drivers/Receivers	1/1	1/10	1/10	32/32
Maximum Cable Length	50	4000	4000	4000
Maximum Data Rate (bps)†	20K	100K	10M	10M

† The ZT 8801 maximum data rate is limited to 19.2 Kbaud by the UART.

RS-232-C Operation (Default)

The V40 serial port can be configured as either RS-232-C or RS-422/485. RS-232-C compatible interfacing is available at J5 and J6. Note that dual operation of RS-232-C and RS-485/422 is not possible. DTE configuration (when used with the ZT 90069 cable) is available at J5; DCE is available at J6.

RS-485/422 Operation

RS-422/485 compatible drivers and receivers are available to allow the serial port to communicate as an RS-422 or RS-485 device. J3 is a 3-pin connector that is connected to a differential driver, which may be optionally gated with the control port at FA05h, bit 3. See the "Control Port" section in this chapter. Connector J4 is optionally configured either as a differential transceiver (driver and receiver) or as a receiver only. It is also optionally gated with the control port bit 3. These combinations allow RS-485 two-wire operation (multidrop), RS-485 four-wire operation, and RS-422 operation (dedicated driver/receiver).

A terminated twisted pair should be used to protect the integrity of the RS-485 signals. A typical twisted pair has a characteristic impedance in the range of 100 to 130 ohms, adequate for data transfer rates of up to 10 MHz. The RS-485 inputs on the ZT 8801 can be terminated by installing a 100 ohm resistor SIP at location RP4 (default). J3 and J4 are balanced when the SIP pin 1 is placed in the pin 1 location. J4 only may be balanced by placing the SIP pin 1 in location 3 of the socket. Extra locations are provided so that SIP will be fully seated in this configuration.

RS-485: 2-Wire Operation

For 2-wire operation, use J4. Remove W10 to disable the RS-232-C receiver. Install W9. The multidrop feature is performed under software control of the RS-485 driver via bit 3 of the control port (FA05h). Writing a 1 to this bit (preserve the other bits by reading first and ORing this bit in) enables the driver. Clearing this bit (by ANDing out to preserve other bits) disables the driver, allowing another device to drive the bus. The receiver may also be controlled via bit 3 to avoid receiving data sent while the ZT 8801 is driving the RS-485 link. See the “RS-485 2-Wire Operation” topic.

RS-485: 4-Wire Operation

For RS-485 4-wire operation, use J3 as the differential driver and J4 as the differential receiver. You can control the driver through bit 3 of the control register if you install W5 and remove W6. Writing a 1 to this bit enables the driver. The driver may be always enabled by removing W5 and W6. Remove W9 for 4-wire operation.

RS-422 Operation

Use J3 as the differential driver or J4 as the differential receiver to configure the serial port as RS-422. J3 may be hardwired with the driver always enabled by removing W5

and W6. Remove W9. Configure J4 as an RS-422 receiver by installing W8 and removing W7.

INTERRUPTS

The ZT 8801 supports both maskable and non-maskable interrupts. This section discusses system level issues

related to these interrupts. Refer to Chapter 7, "Interrupt Controller (V40)," for more information on the operation and programming of the maskable interrupt controller.

Interrupt Request Assignments

The 8259A Programmable Interrupt Controller (PIC) on board the ZT 8801 has seven interrupt input requests plus one internal V40 interrupt (TIMER0), each with a number

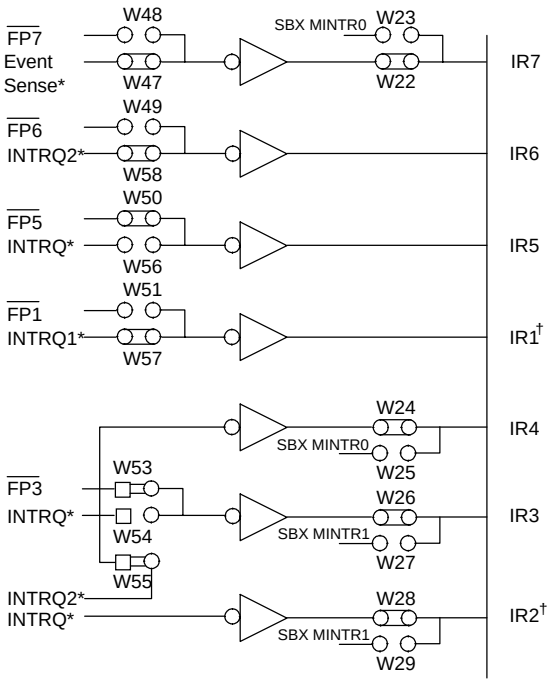
of possibilities for an interrupt source. The "PIC Interrupt Input Requests" figure shows these possibilities, which are assigned via jumper selections W22-W29, W47-W51, and W53-W55. This figure indicates the factory default assignments.

Three of these interrupt assignments are critical for DOS. The counter/timer 0 is used for the Systick Timer and is assigned to interrupt level 0. In addition, if the ZT 8980 Super VGA board is placed in the system, then the keyboard interrupt at level 1 is also critical. Should the ZT 8950 floppy disk controller be installed, then interrupt level 6 is required for proper floppy operation. Ziatech's DOS systems use backplane signals INTRQ1* for keyboard interrupts and INTRQ2* for floppy subsystem interrupts. Do not reassign these interrupt selections in DOS systems.

The "STD Bus Interrupts" table lists the pin assignments for the STD bus interrupts.

STD Bus Interrupts

Interrupt	Pin Number
INTRQ*	P44
INTRQ1*	P37
INTRQ2*	P50



† IR1 and IR2 have other options within the V40. Refer to the OPCN register in Chapter 5 for details.

ZT8801F03-04

PIC Interrupt Input Requests

Polled Interrupts on the STD Bus

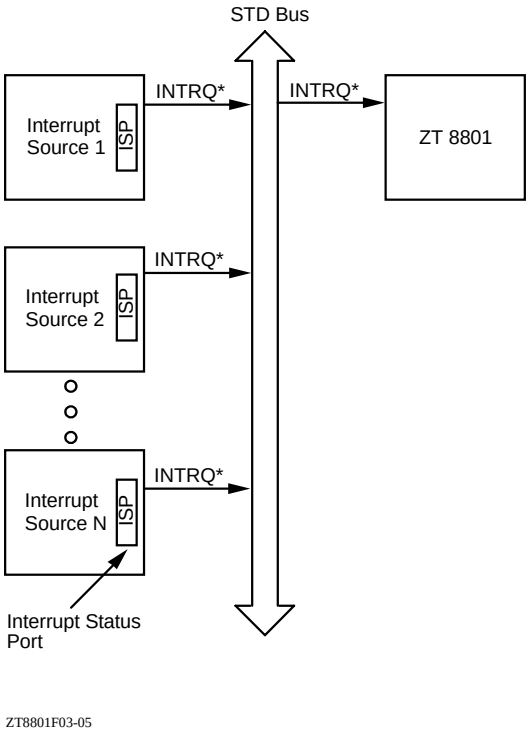
The PIC can be programmed to supply a unique vector for each of these interrupt inputs. This means only one STD bus interrupt per request can be uniquely

defined, as shown in the "Polled Interrupt Structure" illustration.

Since DOS expects the use of INTRQ1* and INTRQ2* for particular I/O devices, this leaves only the one INTRQ* signal for all remaining I/O devices in the system. These devices may share this INTRQ* signal, in which case the application program must poll each possible source to determine which device generated the interrupt. Such a procedure is acceptable for most applications, provided each interrupt source can be polled. A software priority can be established by polling the sources in a specific order.

In general, sharing interrupts requires the use of level-triggered interrupt sources. In this way, if one interrupt exists and another arrives, the level remains active even when the first request has been removed. The level continues to activate the request at the interrupt controller. Conversely, if edge-triggered interrupts are used, the activation of the interrupt request by one device essentially masks the edge created by the second device and the interrupt is not seen by the interrupt controller even though the active logic level remains.

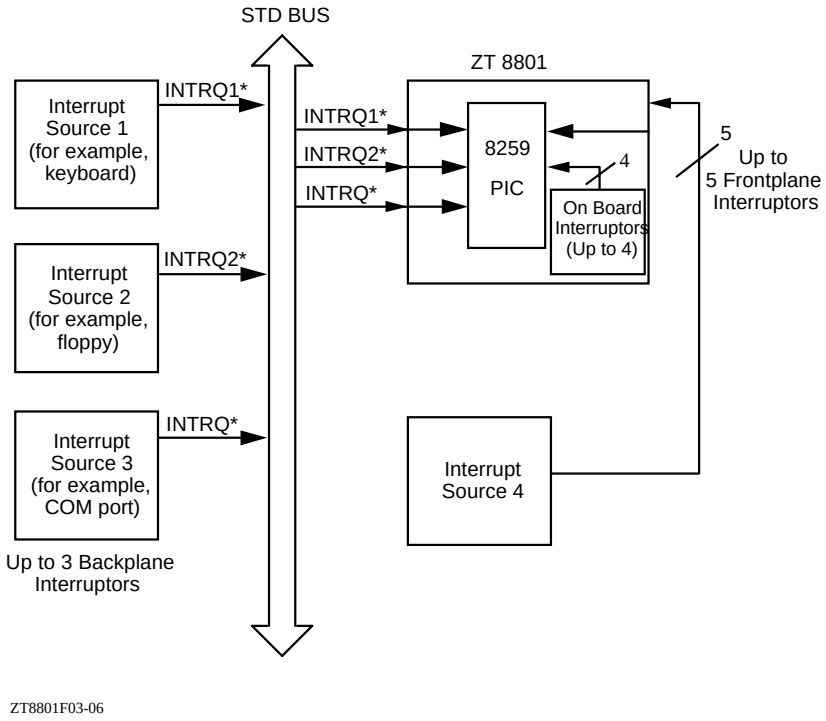
The 8259A PIC can be programmed for either all level-triggered or all edge-triggered interrupts. The use of the counter/timer 0 for the system clock tick requires that the PIC be programmed for edge-triggered interrupts, which is standard for DOS systems. Sharing INTRQ* among two or more devices is not possible under edge-triggered operation.



Polled Interrupt Structure

STD Bus Vectored Interrupts

For more demanding applications, it may be necessary to support each STD bus interrupt source with a unique vector, as illustrated in the "Small Scale Vectored Interrupt Structure" illustration. In this mode, up to eight interrupting devices are automatically provided a vector by the on-board 8259 PIC of the ZT 8801. DOS uses the PIC in this mode. Note that the interrupting devices may be on-board as well as off-board.

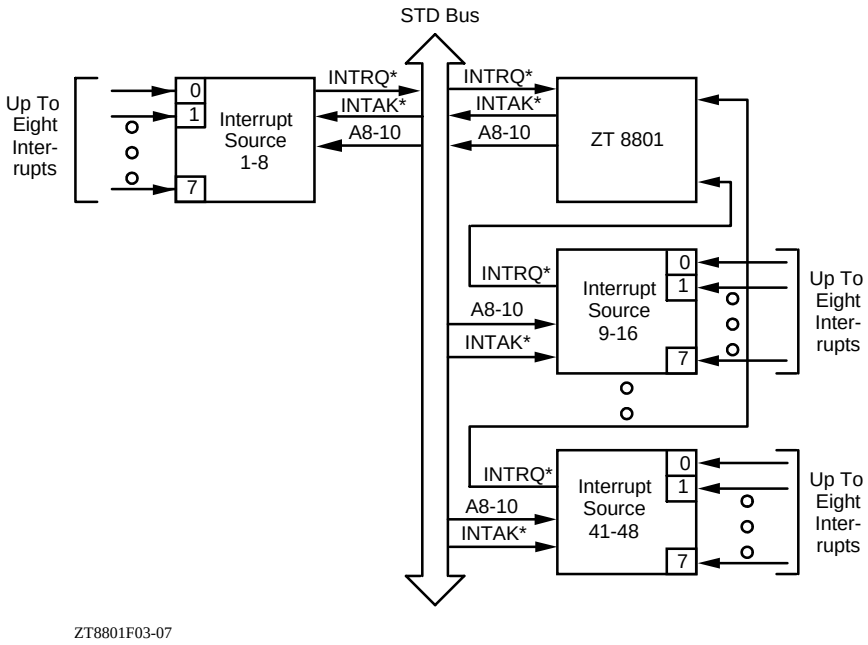


Small Scale Vectored Interrupt Structure

STD Bus Cascaded Interrupts

To allow for a greater number of interrupts, additional interrupt controllers can be added to the STD bus system, permitting each interrupt source to generate a unique vector for its service routine. The ZT 8801 supports the STD-80 implementation of cascaded interrupt controllers; this is useful for demanding applications that have a large number of interrupt sources. The system is illustrated in the "Cascaded Interrupt Structure".

Backplane interrupts INTRQ*, INTRQ1*, and INTRQ2* and any of the frontplane interrupts may be used to cascade interrupt controllers. The interrupt output from the "slave" interrupt controller in the system is tied to the interrupt request input on the ZT 8801 "master" interrupt controller and the PIC programmed accordingly. Then if an interrupt request from a cascaded interrupt controller is to be serviced, the ZT 8801 drives a 3-bit address known as the cascade address onto A8 through A10 of the STD bus during the interrupt acknowledge cycle. This 3-bit address selects one of the cascaded interrupt controllers to provide an interrupt vector for the requesting input. This interrupt scheme supports a maximum of 56 interrupts from external peripherals using IR1-IR7 of the ZT 8801's master PIC. IR0 is not usable for cascade operation and is dedicated to Timer 0. For a complete discussion of cascaded interrupts, refer to Intel Corporation's application note *AP-59*.



Cascaded Interrupt Structure

Non-Maskable Interrupts

In addition to the eight interrupt inputs at the interrupt controller, the ZT 8801 supports one source of an interrupt referred to as a "non-maskable" interrupt. This type of interrupt has higher priority over any of the maskable interrupts from the interrupt controller and is not software maskable. The NMIRQ* signal from the STD bus is buffered and presented to the V40 for service.

RESET

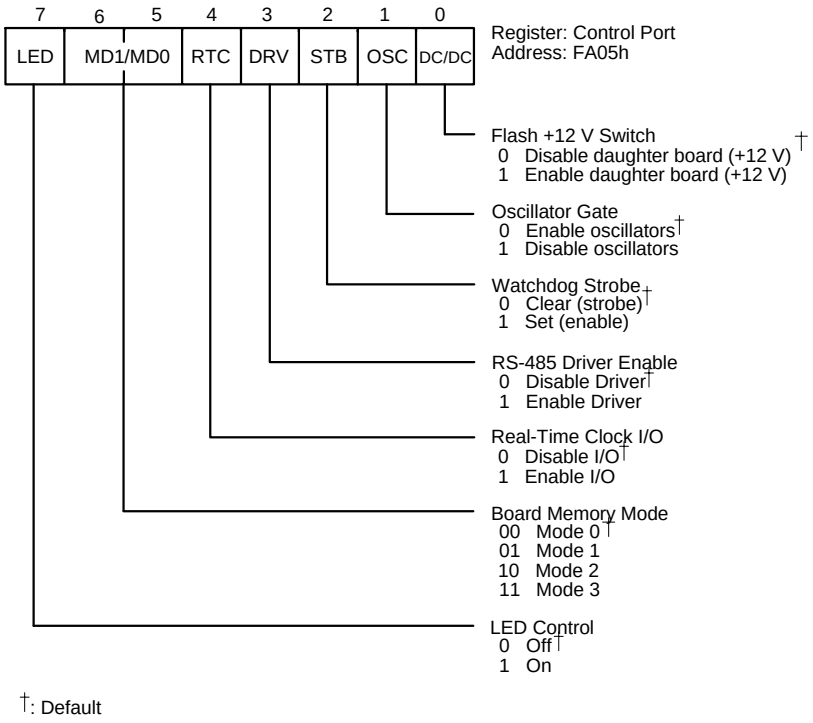
The ZT 8801 is equipped with a System Reset circuit, which asserts the STD bus SYSRESET* signal at any time DC voltage is less than 4.75 V. It also drives the SYSRESET* signal during the time a pushbutton switch drives the PBRESET* STD bus signal to the ZT 8801. The pushbutton switch is first debounced on the ZT 8801 before causing a system reset.

During a reset, the CPU, serial ports, and parallel I/O are reset to initial states (these states are detailed in the descriptions of each of these devices later in this manual). The SBX expansion module is also issued a reset. No other devices are affected by the system reset. The SYSRESET* signal is driven to a logical low for a minimum of 250 ms on power up after the power supply reaches 4.75 V (typically 600 ms, max. 1000 ms). Also on power up, the STD bus DCPDN* signal (pin 5) is monotonically driven to a logical 0 until power is at 4.75 V. If power falls below 4.75 V, DCPDN* is again driven to a logical 0. This signal may be used by external boards protecting their RAM during power fail.

CONTROL PORT

The ZT 8801 utilizes eight bits of the 48-point parallel I/O interface for software configurability. The sixth I/O port (FA05h) is used for the control port. If all 48 points of I/O are needed, then you can trade off features by reconfiguring jumpers and bypassing the software configurability. STD ROM users have the board optimized for maximum I/O. The items controlled by these eight bits are RS-485 driver enable/disable control, memory mode selection, LED, watchdog strobe, real-time clock reset, timer and SBX oscillator disabling, and Flash enable for programming. The "Control Port Bit Map" illustration diagrams the software bit assignments for these features.

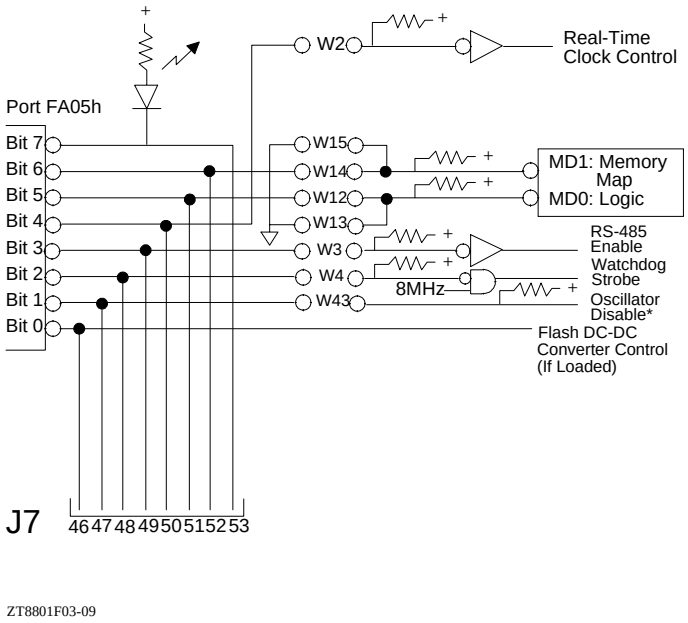
In order to avoid read-modify-write errors, we recommend that you disable interrupts when modifying this register. The default at power up is for all bits to be 0, as shown in the "Control Port Bit Map" illustration.



Control Port Bit Map

Hardwiring Features vs. Software Control

All of the features controlled by the Control Port may be hardwired via jumpers to allow for 48 lines of true off-board I/O. If you are using STD ROM, the board is shipped in this configuration. The "I/O Point Jumper Options" illustration shows the options for these I/O points.



I/O Point Jumper Options

DC POWER-FAIL PROTECTION

The ZT 8801 supports DC power-fail protection. The advantages of this are described in this section.

The factory default setting enables the ZT 8801 to detect 5 VDC and assert a System Reset if power falls below 4.75 VDC. In addition, the RAM and Flash memory chip enables are disabled below 4.75 VDC to prevent erroneous data corruption to these devices during power-down/power-up sequences. If the optional battery is installed, battery voltage switches in to protect those circuits selected by jumpers for battery backup. The real-time clock is always protected if the battery is installed. Jumpers allow additional battery backup of the RAM sockets.

DC power-fail protection provides the advantages of system integrity and battery backup. System integrity is improved because the system is not allowed to operate if DC voltage is less than 4.75 V. System Reset is held active both on power rising and on power falling; therefore, the processor will not try to function when power is invalid and possibly corrupt valid data in battery-backed devices. DC power-fail detection is important for battery backup so that battery power switches in at the proper time as DC fails.

BATTERY

The ZT 8801 contains a socket for an optional 1 Amp-hour, 3.6 V lithium battery (Ziatech part #BAT-00003). The battery protects the real-time clock in case of power failure, and jumpers allow the RAM sockets to also be battery-backed. These jumpers

are W19 installed, W18 removed to battery back the RAM socket at 6D, and jumpers W21 installed, W20 removed to battery back the RAM socket at 5D.

Battery life depends on the current requirement per device powered by the battery. Battery life is calculated for both typical and worst-case situations in the following equations. These equations show how battery life decreases with a greater current load. Note that the battery capacity derates at high temperatures.

1. Real-time clock only

   a. Typical Data Retention Time (25° C):

   Typical Current Drain of Real-Time Clock = 100 nA

   Battery Life (Hours) = Battery Capacity/Current Requirements
   $= 1\ A \cdot h/100 \times 10^{-9}$
   $= 10^{7}$ hours

   which is beyond the 10-year shelf life of the battery (87,000 hours).

   b. Minimum Data Retention Time (65° C):

   Battery Life (Hours) = Battery Capacity/Current
   $= 1\ A \cdot h/1 \times 10^{-6}\ A$
   $= 10^{6}$ hours

   which is limited to the 10-year shelf life of the battery (87,000 hours).

2. Real-time clock, two 128 Kbyte static RAMs (monolithic Sony part assumed)

   a. Typical Data Retention Time (25° C):

   Total Current = RTC + 2 x Sony 581000 - 10L
   = 100 nA + 2 microA = 2.1 microA

   Battery Life = Battery Capacity/Current
   $= 1\ A \cdot h/2.1 \times 10^{-6}\ A$
   = 476,000 hours

   which derates at 25° C to 360,000 hours, still greater than the 10-year shelf life.

   b. Worst Case Retention Time (65° C) with Sony 128K, not low-power:

   Total Current = RTC + 2 x Sony 581000 - 10L
   = 1 microA + 200 microA
   = 201 microA

   Battery Life = Battery Capacity/Current
   $= 1\ A \cdot h/201 \times 10^{-6}$
   = 4,975 hours

   which derates to (4,975)(.87) = 4,328 hours or .5 years

c. Worst Case Retention Time (65° C) with Sony 128K, low-power:

```
Total Current = RTC + 2 x Sony 581000 - 10LL
              = 1 microA + 40 microA
              = 41 microA
Battery Life = Battery Capacity/Current
             = 1 A·h/41 x 10^-6
             = 24,390 hours
which derates to (24,390)(.87) = 21,220 hours or 2.4 years
```

STATUS INDICATOR (LED)

The ZT 8801 has an on-board LED, located near the extractor, for general purpose use. It is turned on by writing a logical 1 to bit 7 of the Control register at I/O address FA05h, and it is turned off by writing a logical 0 to bit 7 of the same address. See the "Control Port" section of this chapter for further detail.

CMOS VERSION OF THE ZT 8801

The ZT 8801 processor board is also available in a CMOS version, the ZT 88CT01. This version provides lower power and extended temperature operation.

The differences between the CMOS version and the non-CMOS version with respect to system level issues are discussed briefly here and in further detail in Appendix B, "Specifications," and Appendix C, "PIA System Setup Considerations."

Added Features of the ZT 88CT01

The ZT 88CT01 extends the operating temperature range to between -40° and +85° Celsius. All on-board logic is CMOS and utilizes an advanced speed TTL compatible CMOS logic family (ACT) to allow the support of both CMOS and non-CMOS STD bus boards. This is an advantage over strictly CMOS logic (HC or AC), which must receive only CMOS logic levels on its inputs and is therefore restricted to operation with CMOS boards only. The ZT 88CT01 does not have this restriction and is named as such to distinguish both CMOS and TTL compatibility.

In addition to these environmental and electrical advantages, the ZT 88CT01 contains an added feature to support very low power applications. Two of the on-board oscillators may be turned off (timer tick and the SBX oscillator) to conserve power. The V40 may also execute the halt instruction to go into a lower power mode.

Clock Shutdown

Power consumption for CMOS logic is directly proportional to the switching speed of the device. The higher the clock frequency, the greater the power dissipation. In order to minimize the power consumption on the ZT 88CT01 board, the Clock Shutdown feature

has been included to allow shutting down the timer tick (1.19318 MHz) and SBX (10 MHz) oscillators. This is dynamically done by writing a 1 to the Control Port at FA05h, bit 1, when W43 is installed. Note that the 16 MHz oscillator that drives the V40 may not be shut down.

Halt with Interrupt Restart

To further decrease power consumption from the Clock Shutdown mode described above, the processor may be halted when processing is not needed and restarted by an interrupt. This interrupt may be from an external source, such as an event requiring service from the processor, or from one of the on-board timers or event sense I/O. If the clock shutdown feature above is not used, the timers continue to run at their full 1.19 MHz frequency and are not affected by a halted processor. Alternatively, the timer clock inputs could count external events and interrupt the processor upon reaching a predetermined count. Any counters not initialized remain idle and do not affect power consumption.

CHAPTER 4. PROCESSOR DESCRIPTION (V40)

The NEC 70208, commonly known as the V40, is a CMOS microprocessor with a 16-bit internal and 8-bit external data bus structure. The V40 instruction set includes all of the instructions of the 8088 and 80188 microprocessors, plus a few more. The added instructions include string I/O, expanded rotate and shift, bit and nibble manipulation, BCD arithmetic, and 8080 emulation mode.

The V40 contains several peripherals frequently used in STD bus applications. These peripherals include a serial controller, interrupt controller, direct memory access (DMA) controller, counter/timers, and a programmable wait-state generator.

This chapter divides the V40 microprocessor into functional blocks and presents an overview of each. More detailed descriptions of the programmable functional blocks are found in subsequent chapters.

ZT 8801 SPECIFICS

The V40 includes a dynamic RAM (DRAM) refresh controller for applications supporting DRAM devices. The ZT 8801 contains only static RAM, so the refresh controller is not needed.

The V40 includes four DMA channels. The ZT 8801 makes use of one of these channels to provide access to on-board memory from external bus masters via the BUSRQ*/BUSAK* sequences. These two STD bus signals are used by external bus masters (such as an off-board DMA controller) to gain control of the STD bus from the ZT 8801. BUSRQ* is driven by the external master to request the bus. The ZT 8801 routes this signal into DMA Channel 0, which if programmed in cascade mode, drives the signal BUSAK* to the external master when the V40 has relinquished control of the on-board memory. The remaining three channels are not supported by the ZT 8801.

FUNCTIONAL BLOCKS

The V40 can be divided into the major functional blocks listed below and shown in the "V40 Block Diagram."

CPU Central Processing Unit

BIU Bus Interface Unit

BAU Bus Arbitration Unit

CGU Clock Generator Unit

VCR V40 Configuration Registers

WCU Wait Control Unit

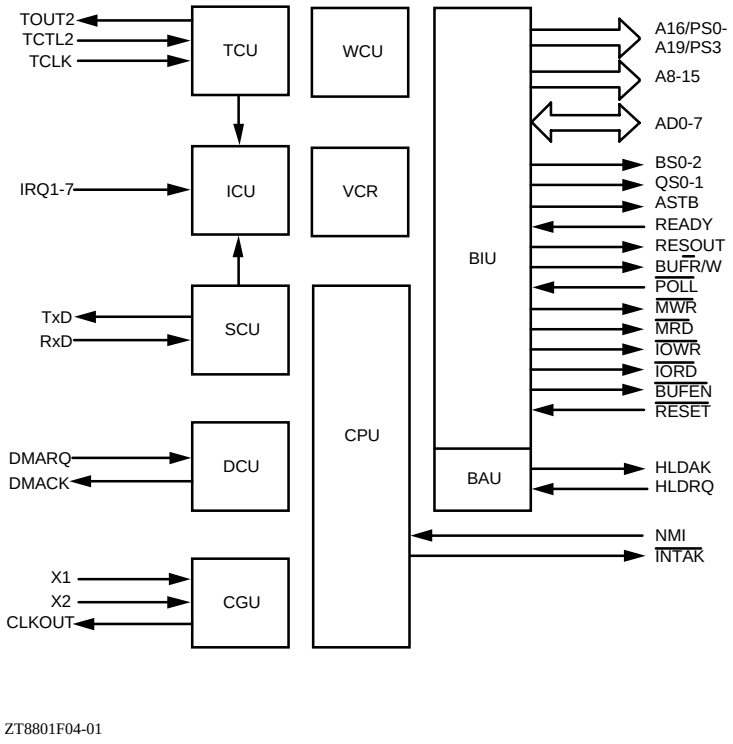
SCU Serial Control Unit

TCU Counter/Timer Control Unit

ICU Interrupt Control Unit

DCU DMA Control Unit

The following topics discuss each functional block in detail.

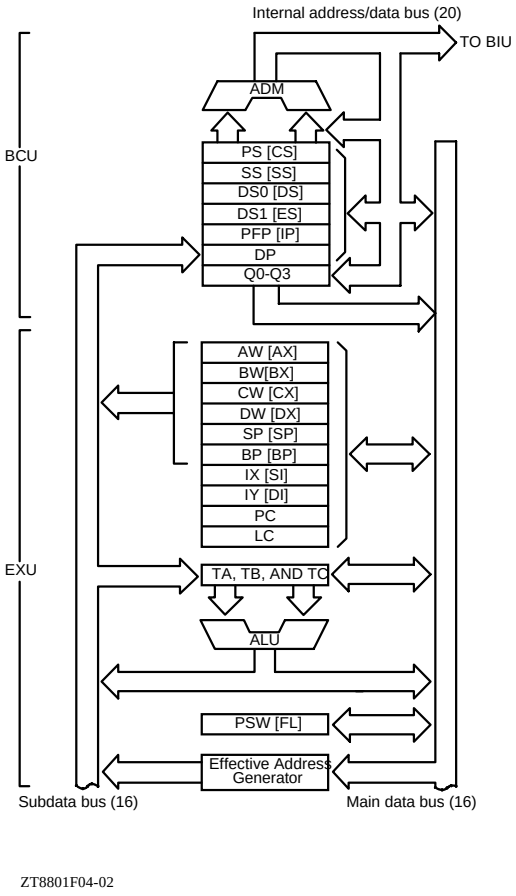


V40 Block Diagram

CPU - Central Processing Unit

The architecture of the CPU functional block is compatible with the 8088. The CPU recognizes all of the instructions found in the 8088 and 80188 microprocessors. The "CPU Block Diagram" illustration shows a block diagram of the CPU divided into two elements, the Bus Control Unit (BCU) and the Execution Unit (EXU). The BCU prefetches instructions and data into a 4-byte instruction queue. The EXU executes the instructions. This pipelined architecture increases the throughput over the typical microprocessor that must wait for an instruction or operand to be fetched before operation is continued.

The following topics discuss the CPU functional blocks, enhanced architecture, and standby mode.



CPU Block Diagram

CPU Functional Blocks

The functional blocks in the "CPU Block Diagram" illustration are described in the topics that follow. The NEC mnemonic is shown for each block, followed by the Intel mnemonic in brackets. For example, the CPU flag register is represented by PSW [FL] because NEC labels it Processor Status Word and Intel labels it Flags.

Segment Registers

PS [CS], SS [SS], DS0 [DS], and DS1 [ES]

The CPU can address up to 1 Mbyte of memory in segments of 64 Kbytes or less. The starting address of a segment is specified in a segment register. The four segment registers are as follows:

PS [CS] Program Segment register

SS [SS] Stack Segment register

DS0 [DS] Data Segment register 0

DS1 [ES] Data Segment register 1

All memory addresses are specified with a segment and offset as shown in the "Segment Registers" table. The segment and offset used depend on the type of instruction being executed.

The program always resides in a program segment pointed to by the PS [CS] register. The PFP [IP] always contains the offset within the program segment. The program stack always resides in the stack segment pointed to by the SS [SS] register. The SP [SP] contains the offset of the top of the stack. Stack variables can be addressed using the BP [BP] register because the default segment register is SS [SS].

Program variables generally reside in the data segment with the segment address in the DS0 [DS] register. The offset of the variable within DS0 [DS] is called the effective address. The EXU calculates the effective address by summing any combination of displacement, base register, and index register. The possible combinations of offset, base, and index provide the programmer with a large variety of addressing modes.

Strings are addressed differently from other variables. The segment register used to point to the source string is DS0 [DS] unless an override is used. The offset for the string source is the IX [SI] register. The segment register for the string destination is always DS1 [ES] and the offset is specified in IY [DI].

Segment Registers

Memory Reference	Default Segment	Alternate Segment	Offset
Instruction Fetch	PS[CS]	NONE	PFP[IP]
Stack Operation	SS[SS]	NONE	SP[SP]
Variable (except following)	DS0[DS]	PS[CS], DS1[ES], SS[SS]	Effective Address
String Source	DS0[DS]	PS[CS], DS1[ES], SS[SS]	IX[SI]
String Destination	DS1[ES]	NONE	IY[DI]
BP[BP] Used As Base Register	SS[SS]	PS[CS], DS0[DS], DS1[ES]	Effective Address
BW[BX] Used As Base Register	DS0[DS]	PS[CS], DS1[ES], SS[SS]	Effective Address

Prefetch Pointer

PFP [IP]

The prefetch pointer PFP [IP] is a 16-bit binary counter that maintains the offset of the next instruction to be fetched into the instruction queue. The BCU fetches a program instruction based on the segment value in the PS [CS] register and the offset in the PFP [IP].

For sequentially addressed instructions, the PFP [IP] is incremented by the number of bytes of the current instruction to point to the next. For program branching, such as intrasegment and intersegment jumps, the PFP [IP] is programmed with a value contained within the jump instruction. The PFP [IP] is not accessible to the programmer.

Data Pointer

DP

This 16-bit register is the destination for the offset address calculated by the effective address generator. The offset address calculation is done by hardware instead of the traditional microcode, saving three to ten clock cycles for every calculation. The DP register is not accessible to the programmer.

Instruction Queue

Q0 - Q3

The instruction queue is a temporary storage location for program instructions and variables that have been fetched by the BCU to be executed by the EXU. The instruction queue consists of four 8-bit registers, Q0 through Q3. These registers allow instruction fetching by the BCU and instruction execution by the EXU to be independent operations. This overlap essentially eliminates the time required to fetch program instructions and data. Q0 through Q3 are not accessible to the programmer.

Address Modifier

ADM

The V40 uses a 20-bit memory address to access any location in the 1 Mbyte addressing range. The 20-bit memory address is the sum of a segment (shifted left four bits) and an offset. The offset is taken from the PFP [IP] if a program instruction is being addressed or from the DP for all other data. The ADM does this addition. If the PFP [IP] was used, the ADM increments it for the next instruction. The ADM is not accessible to the programmer.

General Purpose Registers

AW [AX], BW [BX], CW [CX], and DW [DX]

The CPU has four 16-bit general purpose registers. Each of these registers can be addressed as one 16-bit register or two 8-bit registers. The 16-bit registers are referred to as AW [AX], BW [BX], CW [CX], and DW [DX]. The high order bytes of the 16-bit registers are AH, BH, CH, and DH, while the low order bytes are AL, BL, CL, and DL. The most common use of these registers is to provide a temporary storage location for data. Some instructions do assign specific meanings to the general purpose registers, as shown in the "Implied Use of General Registers" table.

Implied Use of General Registers

Register	Implied Use
AW [AX]:	Word Multiplication/Division Word Input/Output
AL:	Byte Multiplication/Division Byte Input/Output Translation BCD and Decimal/Arithmetic
AH:	Byte Multiplication/Division
BW [BX]:	Translation
CW [CX]:	String Operations
CL:	Variable Shift and Rotate
DW [DX]	Word Multiplication/Division Indirect Input/Output

Pointers and Index Registers

SP [SP], BP [BP], and IX [SI], IY [DI]

The two 16-bit pointer registers are used primarily for stack operations. The Stack Pointer (SP [SP]) is the offset to the top of the stack within the stack segment. This pointer is adjusted automatically each time a stack operation is performed. The Base Pointer (BP [BP]) is an offset to any location within the stack segment. The BP [BP] is useful as a pointer to variables being passed on the stack. Both pointer registers are accessible to the programmer.

The 16-bit index registers are used primarily for string operations. Strings are linear arrays of data that can be organized as words, bytes, nibbles, or even bit values. The index registers specify the offset of the string source (IX [SI]) and destination (IY [DI]) within Data Segment 0 (DS0 [DS]) and Data Segment 1 (DS1 [ES]), respectively. The index registers are adjusted automatically during string transfers. Both IX [SI] and IY [DI] are accessible to the programmer.

Program Counter

PC

The program counter is a 16-bit binary counter that contains the offset address of the next instruction to be executed by the EXU. The PC is automatically incremented each time the EXU reads an instruction from the queue. If the instruction causes a branch in program execution, the PC is programmed with the branch address. At this point the contents of the PC are the same as the PFP [IP]. The difference between the PFP [IP] and the PC is the PFP [IP] contains the offset of the next instruction to be fetched by the BCU and the PC contains the address of the next instruction to be executed by the EXU. The PC is not accessible to the programmer.

Loop Counter

LC

LC is a binary counter used to regulate iterative operations such as string transfers controlled by the repeat prefix and multiple-bit shifts and rotations. The CPU uses hardware for a loop counter as opposed to microcode used by the traditional microprocessor.

Temporary Registers A, B, and C

TA, TB, and TC

These 16-bit registers are used by the Arithmetic and Logic Unit (ALU) during arithmetic and logical instructions such as multiplication, division, and shift and rotate. TA and TB combine for 32-bit temporary storage during multiplication and division. The programmer does not have access to the temporary registers.

Arithmetic and Logic Unit

ALU

The ALU performs arithmetic and logic operations as well as bit manipulation. Arithmetic and logic operations include add, subtract, multiply, divide, increment, decrement, compare, complement, AND, OR, and exclusive OR. Bit manipulation includes shifting, rotating, comparing, setting, clearing, and inverting of individual bits.

Effective Address Generator

EAG

The 16-bit offset address calculated by the EXU for memory operations is called the effective address. The effective address can include a displacement, base, index, or combination of the three, depending on the addressing mode specified in the instruction being executed. Traditional microprocessors calculate this effective address using microcode. The EAG does this calculation in hardware. As an example, the 8088 requires up to 12 clocks to calculate the effective address using microcode. However, the V40 does all effective address calculations in two clocks with the hardware EAG.

The effective address, once calculated by the EAG, is transferred to the DP register where it can be used by the BCU to transfer data between the CPU and memory.

Processor Status Word

PSW [FL]

There are six status flags and four control flags in the 16-bit PSW [FL], as seen in the "Processor Status Word" illustration. Notice that not all 16 bits are defined. Those not

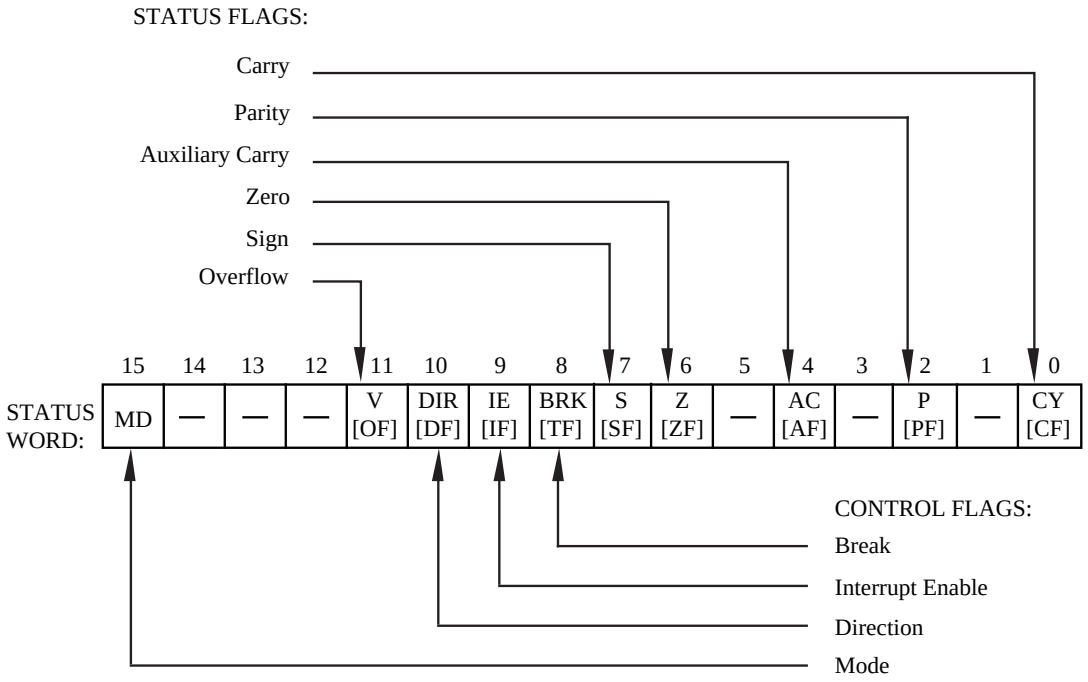
defined are reserved; that is, they may be used in later versions of the processor. Because of this, a program should never rely on a value in any of these reserved bits.

The status flags provide information about the result of arithmetic and logic operations. The status flags are set (logical 1) and reset (logical 0) by the EXU based on the result of an arithmetic or logic operation. These flags can be tested by conditional jump instructions to change the order of program execution.

The control flags are used by the programmer to direct CPU operation. The control flags are set (logical 1) and reset (logical 0) with dedicated instructions. The IE [IF] and BRK [TF] flags are automatically reset when the program enters an interrupt service routine.

The PSW [FL] is automatically preserved on the stack at the start of an interrupt service routine for both hardware- and software-initiated interrupts and at the start of a procedure initiated with the CALL instruction. A return from interrupt instruction, a return from procedure instruction, or a return from emulation instruction restores the contents of the PSW [FL] from the stack.

Different instructions affect the status flags differently. In the descriptions of the status flags in this chapter, reference to bit position is based on the least significant bit being bit 0. The state of a flag is referred to as "set" when a logical 1 is present and "reset" when a logical 0 is present.



Processor Status Word

Status Flags (Processor Status Word)

CY [CF] (Carry Flag)

The carry flag is set if an addition results in a carry out of bit 7 for byte operations or bit 15 for word operations. The CY [CF] flag is also set if a subtraction results in a borrow into bit 7 for byte operation or bit 15 for word operations.

For unsigned byte multiplication, CY [CF] is reset if the most significant byte of the result (register AH) is 0. The same is true of the most significant word (register DW [DX]) for unsigned word multiplication.

For signed multiplication, CY [CF] is reset if the sign bit of the least significant byte (register AL) is extended to the most significant byte (register AH). The same is true for signed word multiplication with the least significant word in register AX and the most significant in register DX.

P [PF] (Parity Flag)

The P [PF] flag is set if the least significant byte of an arithmetic or logical result has an even number of bits set. This flag is useful for checking the parity of ASCII characters.

AC [AF] (Auxiliary Flag)

AC [AF] is set if an addition results in a carry from the four least significant bits of the result. This is true for both byte and word addition. This flag is used by the CPU for BCD arithmetic operations.

Z [ZF] (Zero Flag)

Z [ZF] is set if the result of an arithmetic or logical operation is zero. A common use of this flag is to determine if two numbers are equal. The program subtracts the two values and if Z [ZF] is set, the values are equal.

S [SF] (Sign Flag)

Arithmetic and logic operations set S [SF] equal to the high order bit of the result. This is bit 7 for byte operations and bit 15 for word operations. For signed binary operations, S [SF] is reset for positive results and set for negative results. Programs using unsigned operations usually ignore S [SF] because the high order bit does not reflect the sign of the result.

V [OF] (Overflow Flag)

The V [OF] flag is set if the result of an operation is a positive number that is too large or a negative number that is too small to fit into the destination. The application program can use the overflow flag to determine if the result of two's complement arithmetic operation is out of range.

Control Flags (Processor Status Word)

MD (Mode Flag)

The CPU operates in either native or emulation mode. In native mode, the CPU executes the standard 8086/186 instructions in addition to instructions unique to the V40. In emulation mode, the CPU executes an 8080 based instruction set. The MD flag is used to distinguish between the two modes. MD is programmed using specific instructions to put the CPU in the native mode (MD is set) or emulation mode (MD is reset). Refer to the "Emulation Mode" section in this chapter for more information.

DIR [DF] (Direction Flag)

The CPU supports string operations to manipulate linear arrays of data organized as words, bytes, nibbles, or bits. Index registers are used to point to elements of the array during a string operation. After a string operation is completed, the index registers are incremented or decremented depending on the state of the DIR [DF] flag. If the DIR [DF] flag is set, the index is incremented to point to the next array element. If the DIR [DF] flag is reset, the index is decremented.

IE [IF] (Interrupt Enable Flag)

The IE [IF] flag determines how the CPU responds to maskable external interrupts. If IE [IF] is set, the CPU recognizes maskable external interrupts. The CPU ignores all maskable external interrupts if IE is reset. IE [IF] has no effect on external non-maskable interrupts or internally generated interrupts. IE [IF] is set or reset with dedicated instructions, but will also be reset automatically with a return from interrupt instruction.

BRK [TF] (Break Flag)

Setting the break (or trap) flag puts the CPU into a single-step operation useful for testing program execution. With BRK [TF] set, the CPU automatically generates an internal interrupt after each instruction. The programmer need only develop an interrupt service routine to examine contents of registers, dump memory, or do whatever is necessary for testing.

The BRK [TF] flag is set or reset by transferring the PSW [FL] to the program stack and using memory manipulation instructions to modify it. Once BRK [TF] is modified, it must be transferred back to the PSW [FL] to generate a type 1 interrupt after the execution of each instruction. As part of the interrupt acknowledge, the PSW [FL] is saved on the stack and the BRK [TF] flag is reset. This is done so the processor will not single-step through the interrupt service routine. Once the service routine is completed, the PSW [FL] is restored from the stack automatically, setting the BRK [TF] flag to trap the next instruction.

Enhanced Architecture

The V40 CPU includes several enhancements that provide an increase in performance over the 8088 microprocessor found on many STD bus designs. The most noticeable performance improvements come from additional hardware for the Effective Address Generator, Loop Counters and Shifters, and the use of dual internal data buses.

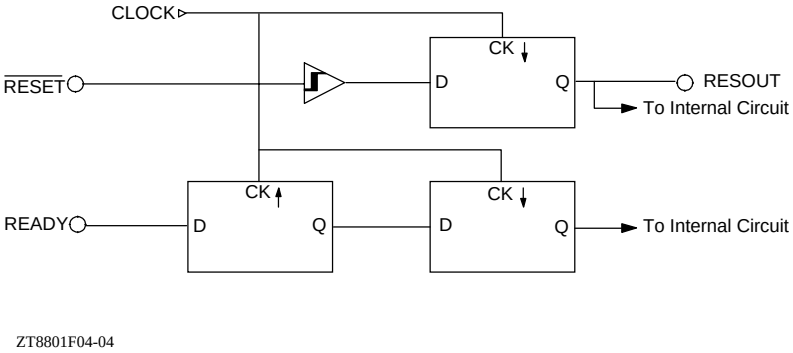
Using a hardware-effective address generator rather than microcode reduces the time to fetch memory operands by as much as 10 clocks per fetch. Using hardware counters and shifters instead of the conventional microcode increases the speed of multiply and divide instructions by as much as four times. Dual internal data buses reduce traffic for instructions with two operands and for effective address calculation. These enhancements add up to a speed increase of as much as 30 percent over the 8088 microprocessor.

Standby Mode

The CPU has a standby mode to reduce power consumption by more than one-tenth during idle periods. Standby mode is automatically entered when the HALT instruction is executed from the native or 8080 emulation mode. This does not affect any of the internal peripherals, such as the counter/timers, interrupt controller, refresh controller, or DMA controller. The CPU automatically exits the standby mode after a reset or an interrupt.

BIU - Bus Interface Unit

The BIU controls the external address, data, and control buses. The BIU also synchronizes the RESET and READY inputs with the clock, as shown in the "RESET and READY Synchronization" illustration. The synchronized RESET signal is used internally. It is provided externally as a signal called RESOUT. The synchronized READY signal is combined with the output of the Wait Control Unit to control the number of wait states inserted during bus operations.



RESET and READY Synchronization

BAU - Bus Arbitration Unit

The V40 includes two internal bus masters and two signals supporting one external bus master. The two internal bus masters are the Central Processing Unit and DMA Control Unit. An external master, such as the 8087 numeric data processor, is supported with the HLDRQ and HLDAK signals. Each of the three bus masters mentioned above needs access to the address, data, and control buses to perform their function. The BAU controls which of these bus masters has access to the buses at any given time. The bus masters are prioritized in the following order:

(1st) DCU - DMA Control Unit

(2nd) HLDRQ - External Bus Master

(3rd) CPU - Central Processing Unit

If one bus master is using the bus and another bus master with higher priority makes a request, the BAU inactivates the current bus master's acknowledge. The BAU grants access to the higher priority bus master after the current bus master removes the request.

The BUSLOCK prefix prevents all bus masters other than the CPU from gaining access to the bus.

CGU - Clock Generator Unit

The CGU halves the frequency of the external oscillator to provide a clock reference with a 50% duty cycle to the CPU. This same signal is available on an external pin called CLKOUT to which all of the V40 timing parameters are referenced.

VCR - V40 Configuration Registers

Twelve programmable registers are used to configure the V40 to meet the needs of varying applications. The V40 configuration registers are located in the top 16 bytes of the 64 Kbytes of I/O address space. The configuration registers define the functions of the programmable pins; the enabling and disabling of the SCU, TCU, ICU, and DCU; the location of the SCU, TCU, ICU, and DCU programmable registers in I/O address space; the wait-state configuration; DRAM refresh; and the counter/timer clock source.

WCU - Wait Control Unit

The WCU provides added flexibility for interfacing to memory and I/O that have varying speed requirements. The V40 includes three internal bus masters that access memory and I/O devices. The number of wait states inserted can be programmed separately for the CPU, RCU, and DCU. The memory space can be divided into three separate areas and the number of wait states defined differently for each. The wait states are programmed through the WCY2, WCY1, and WMB V40 configuration registers.

SCU - Serial Control Unit

The serial control unit is a single asynchronous serial channel used for serial communication between the V40 and a serial device external to the V40. Programming the SCU is similar to programming the 8251A Serial Control Unit for "asynchronous" modes of operation.

TCU - Counter/Timer Control Unit

The Counter/Timer Control Unit includes three 16-bit programmable counter/timers. These counter/timers can be used for SCU baud rate generation, timing loops, timed and periodic interrupts, and external asynchronous event counters. Programming the TCU is similar to programming the 8254 Programmable Interval Timer, with a few restrictions placed on operating modes because of the way the TCU is connected internally to the V40.

ICU - Interrupt Control Unit

Interrupts provide an efficient interface between the V40 CPU and supporting peripheral devices. The ICU supports eight interrupts directly and can be cascaded with other interrupt controllers, such as the 8259A Programmable Interrupt Controller, to support additional interrupting sources. Programming the ICU is similar to programming the 8259A.

DCU - DMA Control Unit

The DCU controls high speed data transfer between I/O and memory devices. The DCU is similar to the 8257 Programmable DMA Controller except the DCU supports the full 1 Mbyte of V40 addressing space. The ZT 8801 supports the DCU in Cascade configuration only.

RESET

Resetting the V40 initializes registers internal to the CPU, VCR, SCU, TCU, ICU, and DCU. The reset states for the CPU registers are given in the "CPU Reset State" table. The reset states for registers internal to the VCR, TCU, ICU, DCU, and SCU are given in their respective chapters, Chapters 5 through 9.

The reset states of the program segment and instruction pointer combine to produce a physical address of FFFF0h. This is the address from which the V40 fetches the first instruction after reset.

CPU Reset State

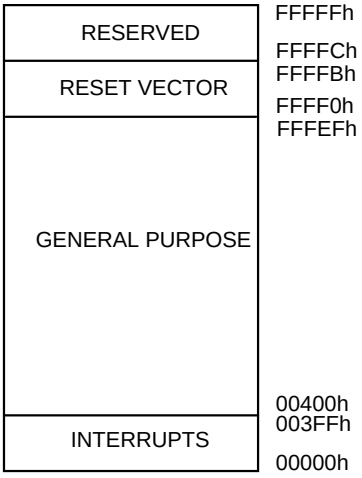
Register	Reset Value
PFP [IP]	0000h
PC	0000h
PS [CS]	FFFFh
SS [SS]	0000h
DS0 [DS]	0000h
DS1 [ES]	0000h
PSW [FL]	F002h
AW [AX], BW [BX]	Undefined
CW [CX], DW [DX]	Undefined
IX [SI], IY [DI]	Undefined
BP [BP], SP [SP]	Undefined
Instruction Queue	Cleared

MEMORY AND I/O ADDRESSING

This section discusses how the V40 communicates with memory and I/O devices. The V40 has a 20-bit address bus and an 8-bit data bus. With 20 bits of address, the V40 can directly access up to 1 Mbyte of memory. The address range is from 0 to FFFFFh, as shown in the "Generic V40 Memory Map" illustration. Address locations 0 to 7Fh are reserved for dedicated interrupts and future enhancements. The address range from 80 to 3FFh completes the interrupt vector table and may be used as needed by the application. The 12 bytes (6 words) from FFFF0 to FFFFBh are the area vectored to by the V40 after a reset. The most common practice is to program this area with an intersegment jump to the start of the application program. The upper four bytes are reserved and must not be programmed.

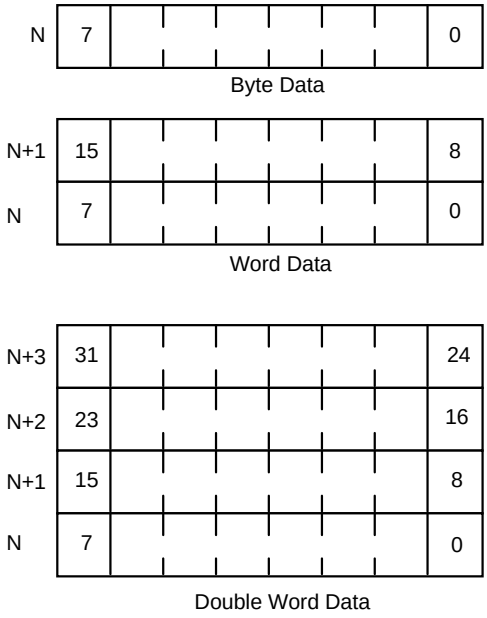
To the programmer, the V40 address space is organized as a contiguous sequence of up to 1 Mbyte. Data can be addressed in units of bytes, words, and double words. Word and double-word values are stored in memory with the most significant byte at the higher address and the least significant at the lower. The "Data Formats" figure illustrates these data formats.

The lower 16 lines of the 20 address lines are also used to address I/O devices. With 16 bits of address, the V40 can directly access up to 64 Kbytes of I/O. The address range is from 0 to FFFFh as shown in the "Generic V40 I/O Map" illustration. Address locations FF00 through FFEFh are reserved for future use. The address range from FFF0 through FFFFh is currently used for the V40 configuration registers. These are registers that define programmable options in the V40, such as wait-state insertion, location of internal peripheral device registers, enabling DRAM refresh, and selecting the period of refresh.



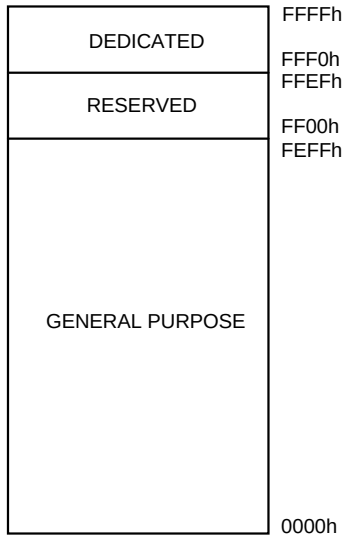
ZT8801F04-05

Generic V40 Memory Map



ZT8801F04-06

Data Formats



Generic V40 I/O Map

INTERRUPTS

The V40 includes a versatile interrupt structure to support both hardware and software initiated interrupts. Hardware interrupts are external inputs to the V40 and can be classified as maskable or non-maskable. Maskable interrupts are routed to the CPU through the ICU. The ICU provides the maskable interrupt inputs with the ability to be level- or edge-triggered, have fixed or rotating priorities, and be individually masked. Non-maskable interrupts are routed directly to the CPU and are not maskable through programming.

Software interrupts are internally generated during program execution, including instructions that can be executed to generate interrupts and error handling for such things as a divide overflow. The "Interrupt Sources" table lists the sources of hardware and software interrupts. The remainder of this chapter explains these interrupts and how the V40 handles them.

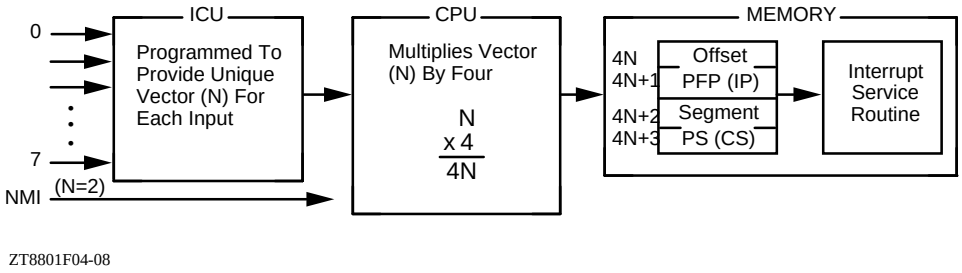
The purpose of an interrupt is to redirect the CPU from its current activity to an interrupt service routine designed to handle the needs of the interrupting source. Every interrupting source is associated with a number that points the CPU to a location in memory that contains the address of the interrupt service routine. This number is called an interrupt vector, and the area in memory where the addresses of the interrupt service routines are stored is called the interrupt vector table. During an interrupt cycle, the CPU multiplies the vector by four to obtain the location of the service routine address in the vector table. The CPU then transfers control to the service routine at the address read from the vector table. This operation is illustrated in the "Interrupt Processing" illustration. It is the programmer's responsibility to load the address of the service routine into the vector table. The address includes a segment and offset value in the format shown.

The vector of an interrupt source must be known before the location of the service routine address can be determined. The interrupt vector table, shown in the "Interrupt Vectors" table, lists the vectors for the sources of interrupts. Fop example, assume you need to write a service routine to handle a non-maskable interrupt (NMI) request. The vector for NMI is 2, as seen in the "Interrupt Vectors" table. The address of the NMI service routine is determined by the application, but that address must be programmed in the vector table at 8h.

Before describing each source of interrupt shown in the vector table, it is useful to summarize the operation of the CPU in response to an interrupt. Interrupts come to the CPU from three sources: the NMI signal external to the V40, the output of the ICU, and from inside the CPU to itself. A vector is supplied in all cases to distinguish between the interrupting sources. The CPU determines the address of the service routine by multiplying the vector times four. Before transferring execution to the service routine, the CPU saves the machine status by pushing the current contents of the PSW [FL] and the return address onto the stack. The CPU then clears the BRK [TF] and IE [IF] flags to prevent subsequent single-step and maskable interrupts, and transfers program execution to the service routine. The service routine is terminated with a "return from interrupt" instruction. This instruction causes the CPU to restore the PSW [FL] from the stack and return execution to the interrupted program. Restoring the PSW [FL] automatically enables single-step and maskable interrupts.

Interrupt Sources

	Interrupt Source	Clocks	Priority
Software	DIVU divide error	45	1
	DIV divide error	45-55	1
	CHKIND breakout error	53-56	1
	BRKV	40	1
	BRK3	38	1
	BRK imm8	38	1
	BRKEM imm8	38	1
	CALLN imm8	38	1
	BRK single step	38	4
Hardware	NMI	38	2
	ICU inputs	49	3



Interrupt Processing

Interrupt Vectors

	Interrupt Source	Vector Number
Software	DIVU divide error	0
	DIV divide error	0
	CHKIND breakout error	5
	BRKV	4
	BRK3	3
	BRK imm8	32-255
	5l BRKEM imm8	32-255
	CALLN imm8	32-255
	BRK single step	1
Hardware	NMI	2
	ICU inputs	32-255

Divide Error

The divide error interrupt is generated by the CPU following execution of a DIV [DIV] or DIVU [IDIV] instruction if the calculated quotient is larger than the specified destination. The interrupt is not maskable and the vector is fixed at 0.

Single-Step

The single-step interrupt is a powerful software debugging tool. The purpose of the single-step interrupt is for software single stepping through a sequence of code. This interrupt is controlled by the BRK [TF] flag in the PSW [FL]. There is no instruction to set the BRK [TF] flag. To set the BRK [TF] flag, the PSW [FL] register must be pushed on the stack, the flag set, and the PSW [FL] popped back off the stack. With the BRK [TF] flag set, a single-step interrupt is generated after each instruction. The CPU responds to the interrupt by pushing the PSW [FL], PS [CS], and PFP [IP] on the stack. The BRK [TF] and IE [IF] are reset to a logical 0 to prevent another single-step or maskable

interrupt. Upon completion of the single-step routine, the CPU restores the PSW [FL], PS [CS], and PSP [IP]. The single-step interrupt is not masked by the IE [IF] bit in the PSW [FL].

Non-Maskable

The V40 has a non-maskable interrupt input called NMI. NMI is rising edge triggered but must remain active for two CPU clocks to guarantee recognition. This interrupt is not maskable and the vector is fixed at 2.

Fixed Vector Instruction

The fixed vector instruction (BRK3 [INT3]) is a special form of the more general variable vector instruction. The difference is the fixed vector is a single byte while the variable vector requires two bytes. The primary use of this instruction is for breakpoint execution during program development. Because it is a single byte, the instruction can be mapped over the smallest possible instruction. This interrupt is not maskable and has a fixed vector of 3.

Overflow

The overflow interrupt is generated if the V [OF] flag is set to a logical 0 and the BRKV [INTO] instruction is executed. This interrupt is useful for trapping overflow errors for mathematical operations. The overflow interrupt is not maskable and the vector is fixed at 4.

Check Index

The purpose of the check index instruction is to test the index of an array against an upper and lower limit. The CHKIND [BOUND] instruction generates a check index interrupt if the index value is less than the lower limit or greater than the upper limit. The vector for the check index instruction is fixed at 5 and is not maskable.

Variable Vector Instruction

Interrupts can be generated using a variable vector interrupt instruction with the format BRK [INT] **xx**, where **xx** is the vector number. Accessing subroutines in this manner means only the subroutine vector is fixed. The location and length of the subroutine can vary without affecting the main program. These interrupts cannot be masked.

Emulation Mode

Two interrupt instructions deal with 8080 emulation mode. The BRKEM instruction is used to transfer V40 operating modes from native to emulation for execution of 8080 based programs. The CALLN instruction is used in emulation mode to call an 8088

procedure. Both instructions have the format and operation of the variable vector instruction. The following topic discusses 8080 emulation in detail.

8080 EMULATION

Designs based on 8080 and 8085 microprocessors have two major limitations: not enough performance and lack of development tools. Upgrading an 8-bit design to a higher performance microprocessor requires time to convert the software. The V40 solves these problems by supporting two modes of operation, emulation and native. When the CPU is in emulation mode, it executes the 8080 instruction set. Emulation mode is used for the existing base of software. In native mode, the CPU executes the 8088 instruction set. All future software development is done in native mode to take advantage of the 8088 instruction set and the large number of development tools designed around it.

The CPU powers up in native mode, which is the normal mode of operation. Two instructions are provided to switch the CPU from native mode to emulation mode and back. Break for Emulation (BRKEM) is the instruction used to switch from native to emulation mode, and Return from Emulation (RETEM) is used to switch back. The effect of these instructions and emulation mode operation is discussed below.

The BRKEM instruction is similar to the BRK [INT] software interrupt. BRKEM includes an 8-bit vector that, when multiplied by four, points to the location in the interrupt vector table that contains the address of the 8080-based routine. During execution of this instruction, the CPU saves the machine status by pushing the current contents of the PSW [FL] and the return address onto the stack. The CPU then clears the mode (MD) flag to a logical 0 and loads the address of the emulation mode routine into the PS [CS] and PFP [IP].

The RETEM instruction is one of four methods to terminate emulation mode. The execution of RETEM is identical in operation to the RETI [IRET] instruction. Upon executing this instruction, the CPU restores the contents of the PSW [FL], PS [CS], and PFP [IP] from the stack, returning program execution to the instruction following BRKEM. The other three methods of exiting emulation mode are a system reset, a hardware interrupt, or the CALLN instruction.

A hardware interrupt suspends the 8080 emulation mode. The CPU pushes the PSW [FL] and return address onto the native mode stack, sets the MD flag to a logical 1, and transfers program execution to the native mode interrupt service routine. When the CPU executes the RETI [IRET] instruction, the PSW [FL] is restored with the MD flag set to a logical 0 and program execution continues in the emulation mode. The CALLN instruction permits the execution of native mode subroutines from emulation mode. The CPU responds to CALLN in the same manner as a hardware interrupt.

The emulation mode cannot be nested. For example, assume the CPU is operating in native mode and executes the BRKEM instruction. The CPU switches to native mode and begins executing emulation code. Next, assume a hardware interrupt (such as the 16450 serial controller) suspends emulation mode and the CPU begins executing the

interrupt service routine in native mode. That interrupt service routine cannot include a BRKEM instruction.

The relationship between the native and emulation mode registers and flags is shown in the "Emulation Mode Registers and Flags"Table. The native mode registers not shown are inaccessible to 8080 programs. They are as follows: AH [AH], PS [CS], SS [SS], DS0 [DS], DS1 [ES], IX [SI], IY [DS], and the upper eight bits of the PSW [FL].

Memory addressing and stack referencing must also be considered. The 8080 addresses a maximum of 64 Kbytes. This block of memory can be located anywhere in the 1 Mbyte address space by programming the PS [CS] word in the interrupt vector table before the BRKEM instruction is executed. All data and stack operations are referenced from the DS0 [DS] register. This register must be initialized before the BRKEM instruction is executed. The values in the PS [CS] and DS0 [DS] registers must be equal for complete compatibility with the 8080 structure.

Emulation mode uses the BP [BP] register for the stack pointer instead of the native mode SP [SP] register in order to reduce the possibility of programming errors in one mode corrupting the stack of the other. This feature is helpful during program development.

Emulation Mode Registers and Flags

	8080	8088
Registers	A	AL
	B	CH
	C	CL
	D	DH
	E	DL
	H	BH
	L	BL
	SP	BP
	PC	PC
Flags	C	CY
	Z	Z
	S	S
	P	P
	AC	AC

CHAPTER 5. PROCESSOR CONFIGURATION (V40)

The V40 is a high integration microprocessor containing a CPU and several peripherals most commonly found in STD bus systems. The V40 also includes a software programmable register set to configure these peripherals to specific applications. This chapter explains the architecture and use of these configuration registers.

VCR - V40 CONFIGURATION REGISTERS

The V40 has 16 configuration registers (4 are reserved), mapped from I/O address FFF0 through FFFFh. The registers are listed in the "V40 Configuration Registers" table below and are discussed in detail in the following topics. All of the registers can be written to with the output instruction and read from with the input instruction. **The value input may be different from the value output, but only in the bits not defined.**

V40 Configuration Registers

I/O Address	Register	Function
FFFFh	Reserved	--
FFFEh	OPCN	V40 multiplexed pin assignment
FFFDh	OPSEL	V40 peripheral enable
FFFCh	OPHA	V40 peripheral I/O address (MSB)
FFFBh	DULA	DCU I/O address (LSB)
FFFAh	IULA	ICU I/O address (LSB)
FFF9h	TULA	TCU I/O address (LSB)
FFF8h	SULA	SCU I/O address (LSB)
FFF7h	Reserved	--
FFF6h	WCY2	DCU wait states
FFF5h	WCY1	CPU memory and I/O wait states
FFF4h	WMB	Memory wait state boundaries
FFF3h	Reserved	--
FFF2h	RFC	Refresh enable & frequency select
FFF1h	Reserved	--
FFF0h	TCKS	Timer/counter clock selection

OPCN - On-Chip Peripheral Connection Register

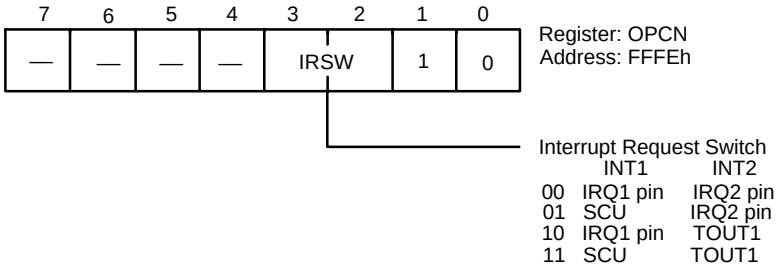
The OPCN register is shown in the "On-Chip Peripheral Connection Register" illustration. This register allows the application to configure the V40 with different capabilities as required.

The two bits of the IRSW field select the interrupt source to be assigned to IRQ1 and IRQ2 of the interrupt controller. The values programmed into bits 2 and 3 depend on the use of the interrupt controller in the application. The pins external to the V40 used for IRQ1 and IRQ2 are connected to the interrupt jumper block.

Bits 0 and 1 are programmed to a 0 and 1 respectively to enable the V40 serial port.

The STD ROM software initializes OPCN to a 06h to configure for the V40 serial port and to use the SCU interrupt on INT1 and IRQ2 on INT2.

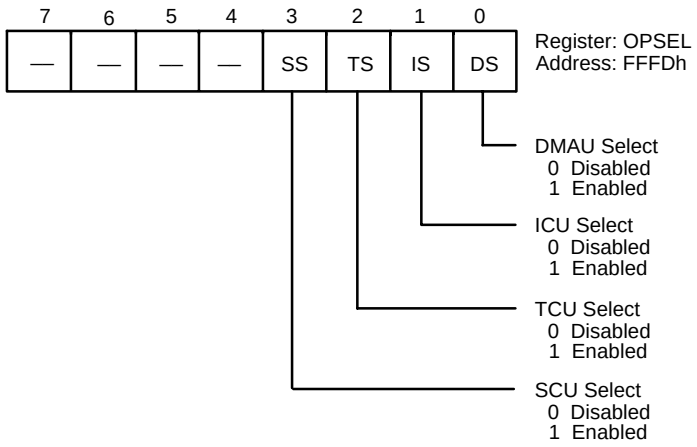
The Ziatech DOS system initializes OPCN to a 02h to configure for the V40 serial port and to use IRQ1 for INT1 and IRQ2 for INT2. Note that DOS systems require the keyboard controller interrupt on IRQ1 and the floppy disk controller interrupt on IRQ2.



On-Chip Peripheral Connection Register

OPSEL - On-Chip Peripheral Selection Register

The V40 integrates several of the most common peripheral devices with a CPU in one package. The peripheral devices include a serial port, interrupt controller, DMA controller, and three counter/timers. The OPSEL register enables or disables these peripheral devices. The format of the OPSEL register is shown in the "On-Chip Peripheral Selection Register" illustration. No restrictions are placed on the use of the OPSEL register. The Ziatech DOS system initializes OPSEL to a 0Fh to enable all of the internal peripherals. The STD ROM software initializes OPSEL to a 0Eh to enable the ICU, the V40 serial port, and the counter/timers for baud rate generation.



On-Chip Peripheral Selection Register

OPHA, DULA, IULA, TULA, and SULA

Five registers determine the I/O base address of the programmable registers used to communicate with the DMA controller, interrupt controller, timer/counters, and serial controller.

OPHA On-Chip Peripheral High Address register (FFFCh)

DULA DMA Unit Low Address register (FFFBh)

IULA Interrupt Unit Low Address register (FFFAh)

TULA Timer/Counter Unit Low Address register (FFF9h)

SULA Serial Unit Low Address register (FFF8h)

The I/O base address is a 16-bit value made up of two 8-bit values. The upper eight bits of the address for the DMA controller, interrupt controller, counter/timers, and serial controller are defined by the OPHA register. The lower eight bits for the DMA channel are programmed in the DULA register. The same holds true for the interrupt controller and IULA register, for the counter/timers and TULA register, and for the serial controller and SULA register.

In operation, OPHA permits the four internal peripheral devices to be mapped to any 256-byte block in the 64 Kbyte I/O address space. The individual registers DULA, IULA, TULA, and SULA are programmed to define the base address of each of these devices anywhere within this block. For example, if the interrupt controller is to be mapped starting at I/O address FF20h, OPHA must be programmed with an FFh and IULA with a 20h.

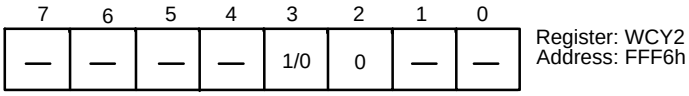
The only restriction placed on the programming of these registers is to be sure the internal V40 peripherals are not mapped in the same address range as other I/O devices local to the ZT 8801. The STD ROM and Ziatech DOS software programs these registers with the values shown in the "STD ROM and Ziatech DOS Address Selection" table.

STD ROM and Ziatech DOS Address Selection

Register	Value	I/O Port Address
OPHA	00	------
DULA	D0	00D0
IULA	20	0020
TULA	40	0040
SULA	B0	00B0

WCY2 - Wait Cycle 2 Register

The V40 includes a programmable wait-state generator to interface to memory and I/O devices that are not fast enough to operate without wait states. The wait-state generator is programmed through the WCY2, WCY1, and WMB configuration registers. The format of the WCY2 register is shown in the "Wait Cycle 2 Register" illustration. The STD ROM software initializes the WCY2 register with a 08h. The Ziatech DOS system programs the WCY2 register with a 00h.

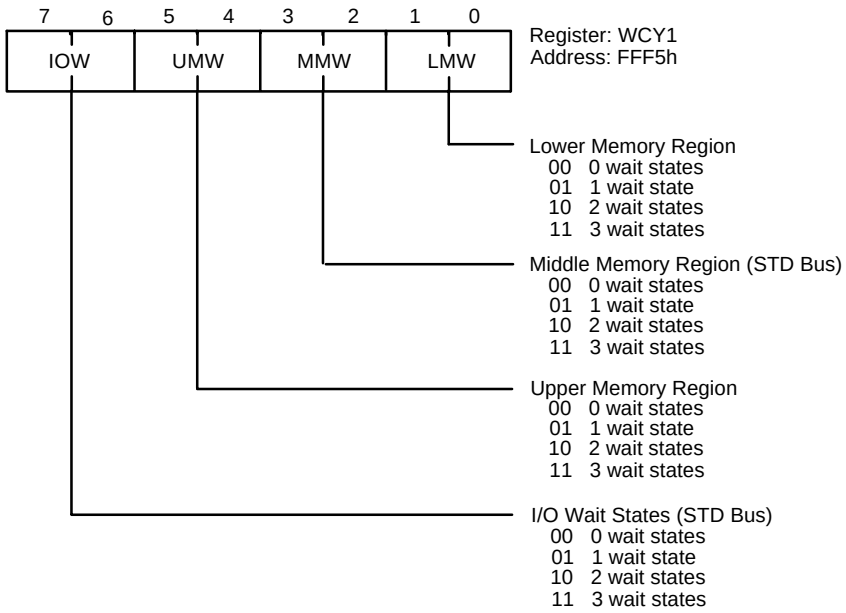


Wait-Cycle 2 Register

WCY1 - Wait Cycle 1 Register

The WCY1 register is divided into four fields as shown in the "Wait Cycle 1 Register" illustration. The first three fields define the number of wait states inserted into three different regions of memory defined by the WMB register.

The ZT 8801 does not require any memory wait states if onboard memory devices with access times less than 200 ns are used. STD bus memory and I/O require one wait state for correct operation. STD ROM and Ziatech DOS software programs the WCY1 register with a 44h. This allows for one wait state for offboard (STD bus) memory and I/O.



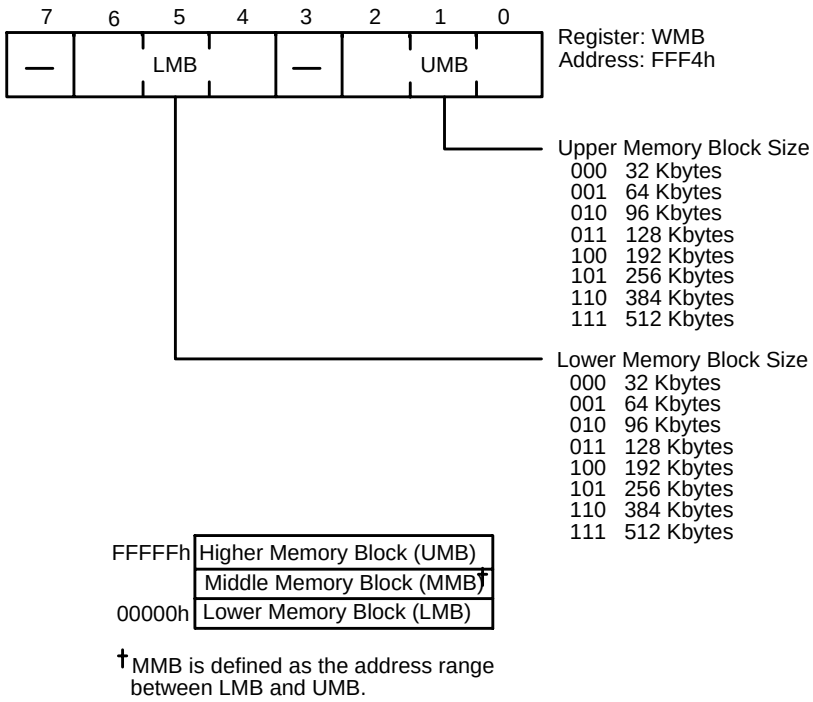
Wait-Cycle 1 Register

WMB - Wait-Cycle Memory Boundary Register

The ZT 8801 does not require any wait states if onboard memory devices with access times less than 200 ns are used. If slower memory devices are used, the WMB register divides the ZT 8801 memory into three regions and the WCY1 register defines the number of wait states inserted into each. As shown in the "Wait-Cycle Memory Boundary Register" illustration, the WMB register is divided into Upper Memory Boundary (UMB) and Lower Memory Boundary (LMB) fields.

The Middle Memory Block is defined between the top of the LMB and the bottom of the UMB. Offboard memory (STD bus) requires one wait state in all cases and is defined by the MMB.

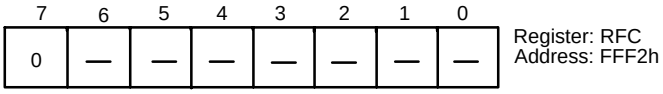
The LMB field defines the lower memory address range starting from zero. This field can be programmed to include the memory devices inserted into the RAM LOW and RAM HIGH sockets. The UMB field defines the upper memory address range ending at FFFFFh. This field can be programmed to include the memory device inserted into the ROM socket. The STD ROM software initializes all bits of the WMB register to 33h. Ziatech DOS configures this register differently for differing amounts of on-board memory. All off-board memory regions are programmed for one wait-state. All on-board regions, including the BIOS, are initialized for zero wait states.



Wait-Cycle Memory Boundary Register

RFC - Refresh Control Register

The ZT 8801 does not use the refresh controller. To prevent the refresh controller from affecting system performance, bit 7 must be programmed with a logical 0, as shown in the "Refresh Control Register" illustration.



Refresh Control Register

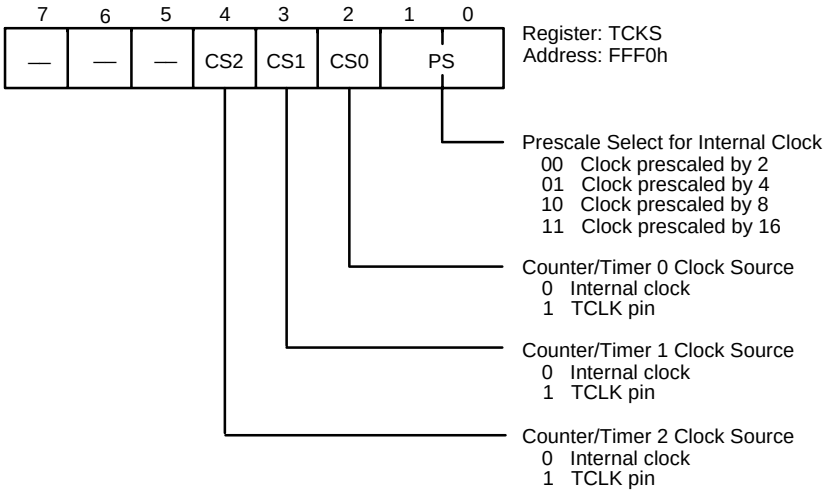
TCKS - Counter/Timer Clock Selection Register

The V40 includes three programmable counter/timers. The TCKS register, shown in the "Counter/Timer Clock Selection Register" illustration, selects the clock source for each counter/timer and a frequency divisor used by all three counter/timers.

The CS0, CS1, and CS2 (Clock Select 0, 1, and 2) bits select the counter/timer clock source to be the reference clock internal to the V40 or the TCLK pin available on an external V40 pin. The V40 clock operates at 8 MHz with a 50% duty cycle. The TCLK signal is available through connector J2. The Prescale (PS) field selects a prescale value that divides the clock frequency of all counter/timers using the V40 internal clock before applying it to the counter/timers.

The STD ROM software programs all bits of TCKS with logical 0s to select the internal clock as the source for all three counter/timers and prescale the clock frequency by 2.

Ziatech DOS configures this register to select the external clock frequency (1.19318 MHz) for all timer/counters and a prescale of 2. The default value for non-VSC DOS systems is 1Ch. If VSC is used, then timer/counter 1 is programmed for the internal clock frequency (16 MHz) for baud rate generation. For VSC DOS systems, the default programmed value for TCKS is 14h.



Counter/Timer Clock Selection Register

RESET

The V40 configuration registers are automatically initialized to a default state when power is applied to the V40 and also during reset from an external source such as a pushbutton reset. The "V40 Configuration Register Defaults After RESET" table shows the default state of the configuration registers before software initialization. The "STD ROM V40 Configuration Register Defaults" table shows the default register values for STD ROM systems, and the "DOS System V40 Configuration Register Defaults" table shows the default register values for DOS systems.

V40 Configuration Register Defaults after RESET

	Default Bit Values [1]							
Registers	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
OPCN	-	-	-	-	0	0	0	0
OPSEL	-	-	-	-	0	0	0	0
OPHA	-	-	-	-	-	-	-	-
DULA	-	-	-	-	-	-	-	-
IULA	-	-	-	-	-	-	-	-
TULA	-	-	-	-	-	-	-	-
SULA	-	-	-	-	-	-	-	-
WCY2	-	-	-	-	1	1	1	1
WCY1	1	1	1	1	1	1	1	1
WMB	-	-	-	-	-	-	-	-
RFC [2]	-	-	-	0	1	0	0	0
TCKS	-	-	-	0	0	0	0	0

[1] Bit positions marked with "-" can default to 1 or 0.

[2] The refresh enable bit of the RFC register is not affected by resets other than power on.

STD ROM V40 Configuration Register Defaults

	Default Bit Values [1]							
Registers	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
OPCN	-	-	-	-	0	1	1	0
OPSEL	-	-	-	-	1	1	1	0
OPHA	0	0	0	0	0	0	0	0
DULA	1	1	0	1	0	0	0	0
IULA	0	0	1	0	0	0	0	0
TULA	0	1	0	0	0	0	0	0
SULA	1	0	1	1	0	0	0	0
WCY2	0	0	0	0	1	0	0	0
WCY1	0	1	0	0	0	1	0	0
WMB	0	0	1	1	0	0	1	1
RFC	0	0	0	0	0	0	0	0
TCKS	0	0	0	0	0	0	0	0

[1] Bit positions marked with "-" can default to 1 or 0.

DOS System V40 Configuration Register Defaults

	Default Bit Values [1]							
Registers	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
OPCN	-	-	-	-	0	0	1	0
OPSEL	-	-	-	-	1	1	1	1
OPHA	0	0	0	0	0	0	0	0
DULA	1	1	0	1	0	0	0	0
IULA	0	0	1	0	0	0	0	0
TULA	0	1	0	0	0	0	0	0
SULA	1	0	1	1	0	0	0	0
WCY2	0	0	0	0	0	0	0	0
WCY1	0	1	0	0	0	1	0	0
WMB [2]	0	1	0	1	0	0	1	1
RFC	0	0	0	0	0	0	0	0
TCKS [3]	0	0	0	1	1	1	0	0

[1] Bit positions marked with "-" can default to 1 or 0.

[2] This value will change depending upon the amount of onboard memory. The value shown is for a 256K system.

[3] This value is for non-VSC systems. VSC systems will be programmed to 14h.

CHAPTER 6. COUNTER/TIMERS (V40)

This chapter describes the Counter/Timer Control Unit (TCU) and provides register descriptions.

The TCU includes three 16-bit programmable counter/timers. Applications for these counter/timers include baud rate generation for the serial controller, timing loops, timed and periodic interrupts, and asynchronous event counters.

The main features of the TCU are:

- Three 16-bit counter/timers
- Six programmable operating modes
- Binary and BCD counting
- Interrupt and polled operation
- Functionally equivalent to 8254

ZT 8801 SPECIFICS

The clock source for each counter/timer is defined in the TCKS V40 configuration register. The choices for the clock source are the V40 internal clock and the TCLK signal. The clock internal to the V40 has a frequency of 8 MHz and a duty cycle of 50%. The TCLK signal is available through connector J2. Optionally, the TCLK signal may be driven by the on-board timer tick oscillator at a rate of 1.19318 MHz, which is consistent with the IBM PC in frequency for the counter/timers. The "J2 Counter/Timer 2 Connector" illustration shows J2 configuration.

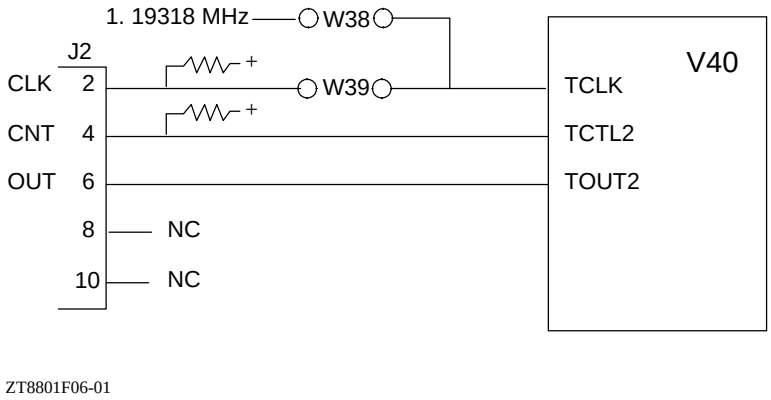
The TCLK signal must meet the following requirements:

- Operating frequency between DC and 8 MHz
- Rise and fall times less than 25 nanoseconds
- Clock low and clock high times greater than 50 nanoseconds

Counter/Timers 0 and 1 have implied uses because of their dedicated output connections to other devices internal to the V40. The output of Counter/Timer 0 is connected to IRQ0 of the interrupt controller. This dedicates Counter/Timer 0 to generating timed or periodic interrupts.

The output of Counter/Timer 1 is connected to the V40 serial port for baud rate generation. The output of Counter/Timer 1 can also be connected to IRQ2 of the interrupt controller if the V40 serial port is not needed. This connection is made with the OPCN V40 configuration register.

Counter/Timer 2 does not have an implied use. Counter/Timer 2 is used in IBM PCs and compatibles for the speaker frequency. The generating speaker is used by DOS through a BIOS function call for reporting keyboard errors and is available for application software. Customers using Counter/Timer 2 must be aware of this, even though there is no speaker on board. A software application may set Counter/Timer 2 for a particular frequency, yet a keyboard error will cause the BIOS to come along later and change it. The output (TOUT2) and control (TCTL2) signals for Counter/Timer 2 are available through connector J2 to be used as required by the application; see the "J2 Counter/Timer 2 Connector" illustration for J2 configuration.



J2 Counter/Timer 2 Connector

PROGRAMMABLE REGISTERS

Four separately addressable registers are provided for communication to the TCU. The TMD (Timer Mode) register specifies the operation of the three counter/timers. The TMD is a write-only register. The other three bi-directional registers are used to write the count to the counter/timers and read back the count and status. These registers are called the Count and Status registers.

The base I/O address of the TCU registers is defined by the OPHA and TULA registers. OPHA is programmed with the high byte and TULA with the low byte of the 16-bit address. Refer to Chapter 5, "Processor Configuration (V40)," for a complete discussion on the V40 configuration registers, including OPHA and TULA. The "TCU Register Addressing" table shows the address of the TCU registers relative to the base address.

The following topics provide descriptions of the TCU's programmable registers.

TCU Register Addressing

Address	Register	Operation
Base + 0	TCT0 Count	Read/Write
Base + 0	TCT0 Status	Read
Base + 1	TCT1 Count	Read/Write
Base + 1	TCT1 Status	Read
Base + 2	TCT2 Count	Read/Write
Base + 2	TCT2 Status	Read
Base + 3	TMD	Write

Timer Mode Register (TMD)

The counter/timers must be initialized with the 8-bit TMD register. The three formats for the TMD register--General Mode, Count Mode, and Multiple Mode--are discussed in the following topics.

The General Mode format is programmed initially to define the operation of the counter/timers. The Count Latch Mode and Multiple Latch Mode are programmed at any time to read the count and status data while the counter/timers are operating.

General Mode

The General Mode format, shown in the "Counter/Timer General Mode Register" illustration, specifies the operating mode of the individual counter/timers. The Select Counter bits specify the Multiple Latch command or which counter/timer will receive the mode. Selecting the Multiple Latch command changes the definition of the TMD register bits to that of the Multiple Latch Mode format. The following bit definitions apply only if the Multiple Latch and Count Latch options are not programmed.

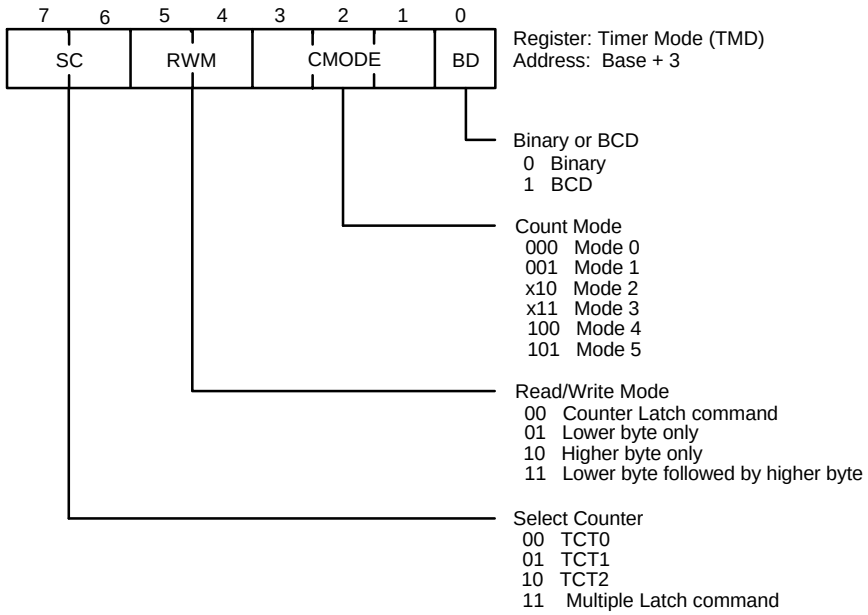
The Read/Write Mode bits specify the Count Latch command or the format of the count transferred between the CPU and the TCU. Selecting the Count Latch command redefines the bits of the TMD register to that of the Count Latch Mode format.

The count transferred to or from the 16-bit counter/timers is one or two 8-bit values depending on the Read/Write Mode bits. If the low byte option is chosen, the 8-bit count transferred to the counter/timer is placed into the low byte of the Down Counter and the high byte is automatically set to 0. The high byte option means the 8-bit count is transferred to the upper byte of the Down Counter with the low byte set to 0. Selecting the two-byte option prepares the counter/timer to receive two bytes, placing the first into the lower byte of the Down Counter and the second into the upper.

A new count can be written into the counter/timers at any time without reprogramming the TMD register. Care must be taken to be consistent with the Read/Write Mode each

time the new count is programmed. As an example, assume that Counter/Timer 0 is programmed with a Read/Write Mode of two bytes. Two bytes must be written to Counter/Timer 0 each time a new count is specified. The same applies for reading Counter/Timer 0; that is, two count bytes must be read at a time.

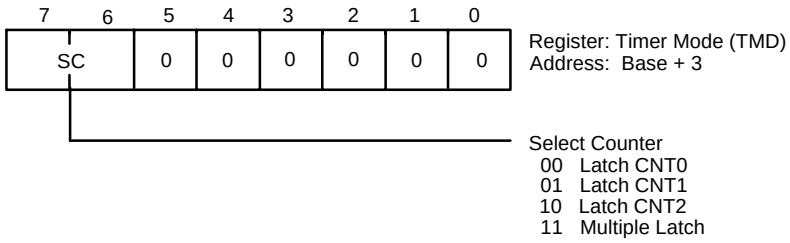
Each counter/timer must be programmed to operate in one of the six possible count modes. The selection of the count mode is based on the needs of the application. The counting operation of each counter/timer is programmed as binary or Binary-Coded-Decimal (BCD). The range of a counter/timer programmed for binary operation is 0 to FFFFh, while BCD operation is 0 to decimal 9999.



Counter/Timer General Mode Register

Count Latch Mode

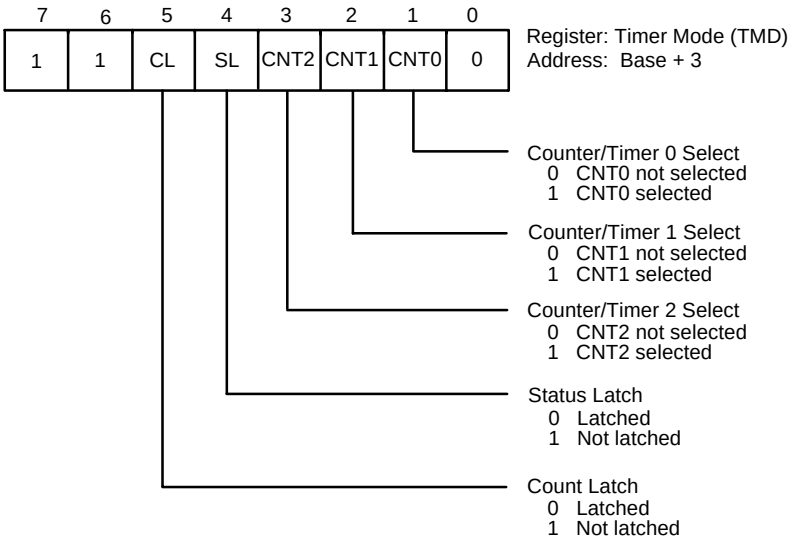
The Count Latch Mode, shown in the "Counter/Timer Count Latch Mode Register" illustration, requires that the first six bits be set to a logical 0. The Select Counter bits specify which counter/timer's data is to be latched.



Counter/Timer Count Latch Mode Register

Multiple Latch Mode

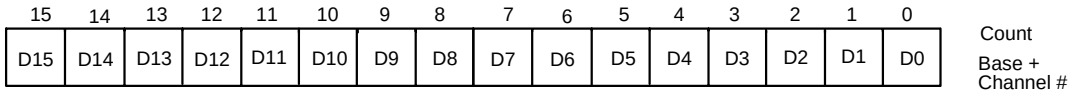
Programming the Select Counter bits of the TMD to logical 1s defines the Multiple Latch command. The format of the Multiple Latch Mode is shown in the "Counter/Timer Multiple Latch Mode Register" illustration. The CNT0, CNT1, and CNT2 bits select which of the counter/timers is latched. The Status Latch and Count Latch bits determine if the status or count, or both, should be latched. The status must be latched to be read. The count can be read without being latched, but it is invalid if it is changing at the time of the read.



Counter/Timer Multiple Latch Mode Register

Count Registers

The Count register is illustrated in the "Counter/Timer Count Register" illustration. Unlike the Mode register, there is one Count register for each of the three counter/timers. The Count register transfers count values to and from the Down Counter. The 16-bit register is programmed with a high byte, low byte, or both high and low byte as specified with the Read/Write Mode bits in the Mode register. If the high byte or low byte mode is selected, only one read or write operation is needed for data transfers. Two read or write operations are required for the two-byte mode, with the low byte transferred first, then the high byte.



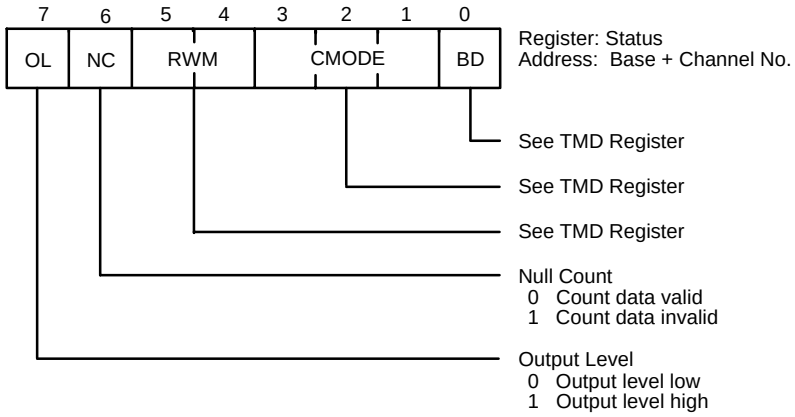
Counter/Timer Count Register

Status Registers

Each counter/timer includes a Status register. The status can be read at any time from the Status register. The "Counter/Timer Status Register" illustration shows the format for the status.

The first six bits of the Status register provide information about the programmed state of the selected counter/timer. This information is in the Status register to prevent the application software from having to save it. The Null Count bit flags when the last count written to the Count register is transferred to the Down Counter. This is designed to prevent the application software from reading the Down Counter before it is updated to the last count written. The Output Level bit contains the current state of the counter/timer output (TOUT).

The Multiple Latch command must be used to read the status. The number of required read operations depends on the Read/Write Mode and the Multiple Latch command. If the Read/Write Mode is high byte or low byte and only the status is latched with the Multiple Latch command, one read operation is all that is needed. Two reads are required if both the status and the count are latched by the Multiple Latch command. The first read is for the status and the second is for the data. If the Read/Write Mode is programmed for two byte transfers and the Multiple Latch command is programmed to latch only the status, one read is all that is required. If both the status and data are latched, three reads are required. The first read is for the status and the next two are for the low and high byte of the count, respectively.



Counter/Timer Status Register

ADDITIONAL INFORMATION

Refer to the NEC *16-Bit V Series Microprocessor* data book for more information on the V40 counter/timers.

CHAPTER 7. INTERRUPT CONTROLLER (V40)

This chapter describes the Interrupt Control Unit (ICU) and provides register descriptions.

The ICU is a programmable interface between interrupt generating peripherals and the CPU. The ICU monitors eight interrupt inputs with programmable priority. When peripherals request service, the ICU interrupts the CPU with a pointer to a service routine for the highest priority device. This type of interrupt management is needed for an efficient interface between the CPU and supporting peripheral devices, such as serial controllers and counter/timers. The major features of the ICU are as follows:

- Eight individually maskable interrupts
- Level-triggered or edge-triggered interrupts
- Fixed and rotating prioritization
- Status available for polled operation
- Functionally equivalent to the 8259

ZT 8801 SPECIFICS

Possible inputs to the interrupt controller are connected as shown in the "Interrupt Controller Inputs" table.

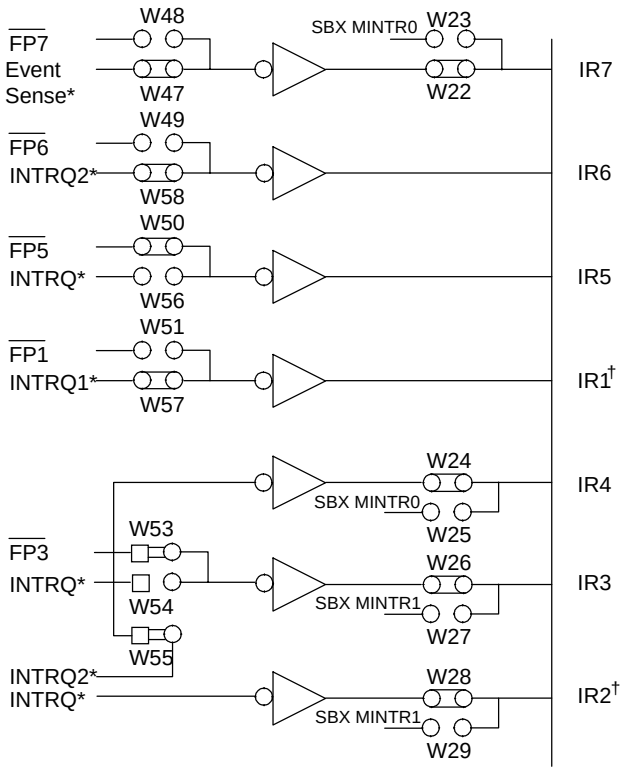
The "Interrupt Jumper Selection" illustration shows the possible interrupt sources for the interrupt input requests IR1-IR7. The IR4 and IR3 levels may be alternately jumpered by using right angle jumpers at W54 and W55, as shown in the "Alternate IR4/IR3 Jumper Selection" illustration.

The following topic describes interrupt controller cascade mode.

Interrupt Controller Inputs

Input	Connection
IRQ0	Counter/Timer 0 output
IRQ1	V40 Serial or FP1* or [INTRQ1*]
IRQ2	Counter/Timer 1 or [INTRQ*] or SBX Expansion Module Interrupt 1
IRQ3	INTRQ* or INTRQ2* or [FP3*] or SBX Expansion Module Interrupt 1
IRQ4	INTRQ* or [INTRQ2*] or SBX Expansion Module Interrupt 0
IRQ5	INTRQ* or [FP5*]
IRQ6	[INTRQ2*] or FP6*
IRQ7	[EVENT SENSE*] or FP7* or SBX Expansion Module Interrupt 0

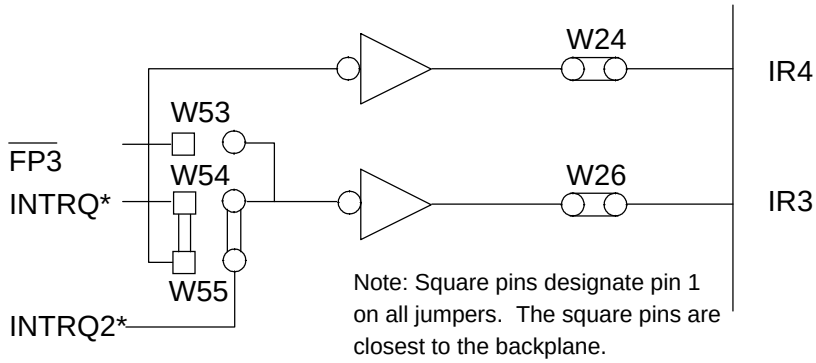
Square brackets [ ] denote default.



† IR1 and IR2 have other options within the V40. Refer to the OPCN register in Chapter 5 for details.

ZT8801F03-04

Interrupt Jumper Selection



ZT8801F07-02

Alternate IR4/IR3 Jumper Selection

Cascade Mode

The interrupt controller includes a cascade mode supported by the ZT 8801. Cascade mode is a scheme by which to expand the number of interrupt request inputs by

connecting up to seven slave interrupt controllers to the inputs of the master interrupt controller. In this scheme, the master interrupt controller must provide a cascade address to the slave controllers during interrupt acknowledge. The cascade address is provided by the ZT 8801 for interrupt levels that are programmed in cascade mode.

When the master interrupt controller is programmed for cascade mode on a particular interrupt level, an interrupt on that level causes the master interrupt controller to initiate a cascade interrupt sequence. The master interrupt controller supplies a 3-bit address (on A8-10 of the backplane) while driving the INTAK* signal on the backplane. The slave interrupt controller responds by driving a vector onto the data lines during the second INTAK* pulse. The host CPU (ZT 8801) then uses this vector to index a table. This table supplies the address in memory of the interrupt service routine. Up to eight interrupts on the slave interrupt controller can then be supported by one master interrupt controller input.

PROGRAMMABLE REGISTERS

The ICU is initialized with Interrupt Initialization Word 1 (IIW1) through Interrupt Initialization Word 4 (IIW4). Once initialized, the operation of the ICU is controlled with the Interrupt Mask Word (IMKW), Interrupt Priority and Finish Word (IPFW), and Interrupt Mode Word (IMDW). There are also three status words that can be read to interrogate the operation of the ICU: the Interrupt Request (IRQ), Interrupt In-Service (IIS), and Interrupt Poll (IPOL). Please note that the "word" reference does not mean the values are 16 bits; all communication to the ICU is done through eight-bit data.

All initialization, operation, and status words are accessed through two I/O addresses as shown in the "ICU Register Addressing" table. (The base address is selected with the OPHA and IULA V40 configuration registers; see the "OPHA, DULA, IULA, TULA, and SULA" section in Chapter 5, "Processor Configuration [V40]," for details.) As might be expected, a specific sequence of read and write operations is needed to pass multiple bytes through a single I/O address. A complete description of all the programmable words and how they are accessed is given in the following topics.

ICU Register Addressing

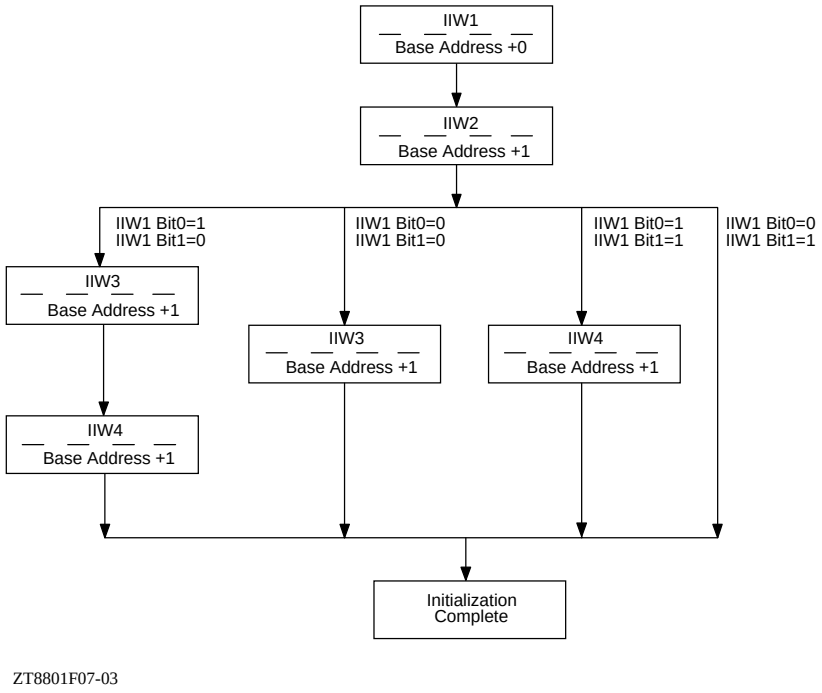
Address	Value	Operation
Base + 0	IRQ, IIS, IPOL	Read
Base + 0	IIW1	Write
Base + 0	IPFW, IMDW	Write
Base + 1	IMKW	Read/Write
Base + 1	IIW2, IIW3, IIW4	Write

Initialization Words (IIW1, IIW2, IIW3, and IIW4)

The ICU must be initialized before it can be used. Initialization consists of writing from two to four bytes called "interrupt initialization words."

The sequence in which these words are programmed is outlined in the "Interrupt Initialization Programming" flowchart. IIW1 and IIW2 must be programmed during any initialization sequence. IIW3 is used to select interrupt levels that are to be used in cascade mode. IIW4 may or may not be programmed as required by the application. The ICU decodes the first two bits of IIW1 to know whether or not to expect IIW4 and IIW3, respectively.

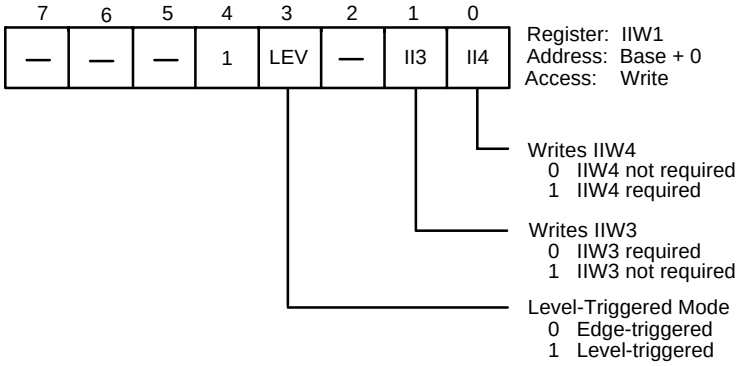
The following topics discuss initialization words IIW1-IIW4.



Interrupt Initialization Programming

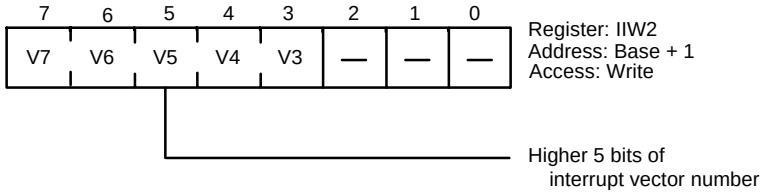
IIW1 and IIW2

Interrupt Initialization Words 1 (IIW1) and 2 (IIW2) are required for ICU initialization. The IIW1 register, shown in the "Interrupt Initialization Word 1" illustration, is divided into three fields labeled II4 (Interrupt Initialization 4), II3 (Interrupt Initialization 3), and LEV (Level). The II4 bit selects whether or not the IIW4 register is to be programmed. If II4 is set to a logical 1, the ICU expects IIW4 to be written as part of the initialization. The II3 bit selects whether or not the IIW3 register is to be programmed. If II3 is set to a logical 0, the ICU expects IIW3 to be written as part of the initialization. The LEV bit of IIW1 selects between level- or edge-triggered interrupt request inputs. Setting LEV to a logical 1 selects level-triggered inputs, while a logical 0 selects edge-triggered.



Interrupt Initialization Word 1

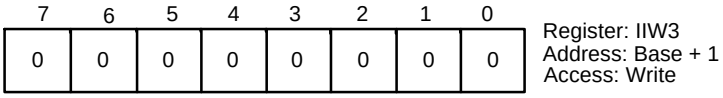
The ICU responds to an interrupt acknowledge by supplying the CPU with an interrupt vector based on which interrupt generated the request and the value programmed into IIW2. The format for IIW2 is shown in the "Interrupt Initialization Word 2" illustration. Bits V3 through V7 define the upper five bits of the vector address.



Interrupt Initialization Word 2

IIW3

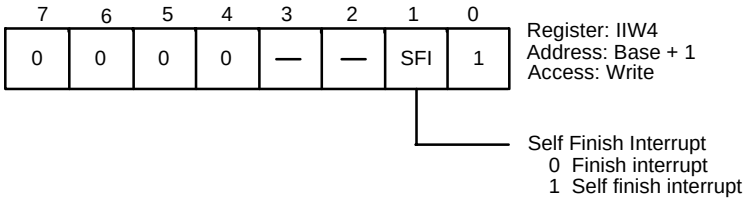
The ZT 8801 does support cascading the interrupt controller inputs to other interrupt controllers. Bits 0 through 7 must be programmed with logical 0s, as shown in the "Interrupt Initialization Word 3" illustration, for the levels that are not to be used in cascade mode. Program a 1 for all levels that are to be used in cascade mode. IIW1 must be programmed with a logical 0 in the II3 bit if IIW3 is used. Common bus practice for cascaded interrupts is to drive the INTRQ* signal from the slave PIC, which is normally jumpered to drive interrupt level 2, on the master PIC.



Interrupt Initialization Word 3

IIW4

The "Interrupt Initialization Word 4" illustration shows the architecture for IIW4. IIW1 must be programmed with a logical 1 in the II4 bit if IIW4 is used. A logical 1 in the SFI bit enables the Self Finish Interrupt and a logical 0 disables it. The interrupt service routine must include an EOI command when the Self Finish Interrupt is disabled.



Interrupt Initialization Word 4

Operation Words (IMKW, IPFW, and IMDW)

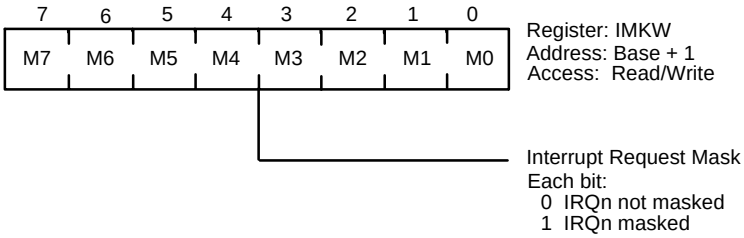
Once initialized, the operation of the ICU is controlled with three 8-bit values called the Interrupt Mask Word (IMKW), Interrupt Priority and Finish Word (IPFW), and the Interrupt Mode Word (IMDW). The Operation Words can be transferred in any sequence to perform such functions as enabling and disabling individual interrupt requests and changing interrupt priorities.

The following topics discuss operation words IMKW, IPFW, and-IMDW.

IMKW

The IMKW masks interrupt request inputs. Interrupts are masked by writing IMKW to the IMK register. IMKW can be read directly from the IMK register to determine the current status of the mask. This eliminates the need for application software to maintain a copy of the mask in program memory.

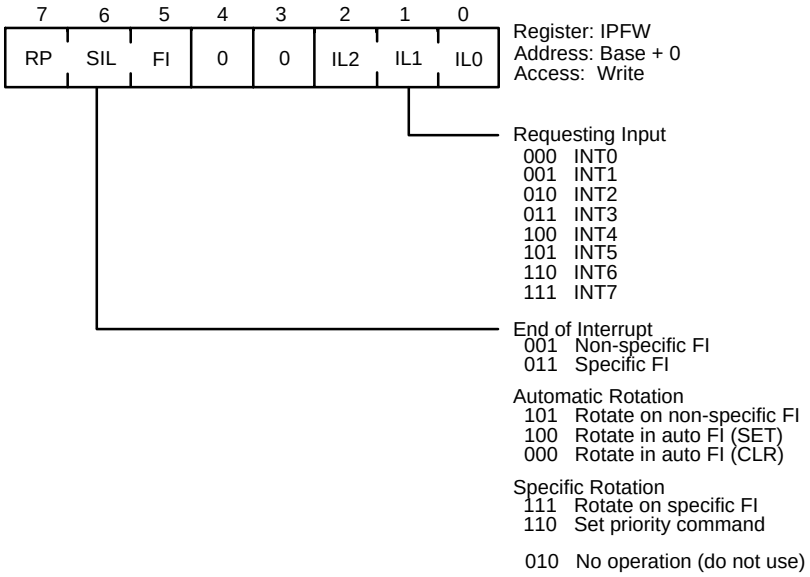
As shown in the "Interrupt Mask Word" illustration, each of the eight bits in IMKW represents an interrupt input. Bit M0 is used to mask IRQ0, M1 is used to mask IRQ1, and so on. Setting a bit in IMKW to a logical 1 prevents the interrupt request for the respective input from being acknowledged by the ICU. The interrupt request is latched in the IRQ register but it never reaches the IIS register.



Interrupt Mask Word

IPFW

IPFW selects fixed or rotating priorities and the method of informing the ICU that an interrupt has been serviced. Operation of the ICU can be changed at any time by writing a new IPFW. Refer to the "Interrupt Priority and Finish Word" illustration when programming the IPFW.



Interrupt Priority and Finish Word

The IL0 through IL2 bits designate an interrupt level. This level is used by certain combinations of the FI, SIL, and RP bits to either reset an interrupt request that has been recognized or set a specific priority.

The ICU uses the IIS register to keep track of which interrupts are being serviced and their relative priorities. The ICU updates the IIS register based on a Finish Interrupt command. There are three methods to generate the finish interrupt command: specific

FI, non-specific FI, and automatic FI. The automatic FI is programmed with IIW4. The specific and non-specific FI are selected with the FI bit. A logical 1 in FI enables the specific or non-specific FI based on the SIL and RP bits.

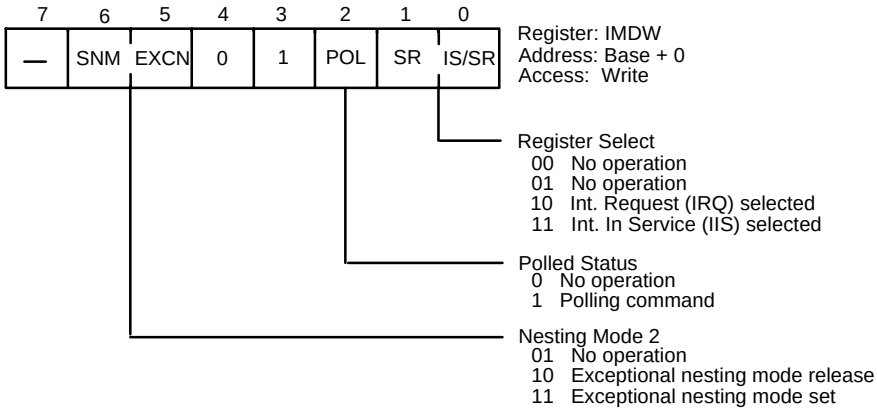
The SIL bit enables bits IL0 through IL2 for selected operations. IL0 through IL2 indicate an interrupt level to be reset during the finish interrupt commands or a new priority for priority rotation commands.

The ICU provides several methods by which to establish priorities for the interrupt request inputs. The RP bit selects the priority rotation options. A logical 1 in the RP bit indicates that rotation in priorities is to take place based on the values of the FI and SIL bits. A logical 0 in the RP bit means no priority rotation will take place.

IMDW

IMDW controls the method of reading status from the ICU and enables a special type of interrupt masking.

The format of the IMDW is shown in the "Interrupt Mode Word" illustration. The first two bits are used to select the IRQ and IIS registers so they can be read by the application software. A logical 1 in bit 0 selects the IIS register and a logical 0 selects the IRQ register. A logical 1 in the SR bit (Select Register, bit 1) enables the reading of the IRQ and IIS registers.



Interrupt Mode Word

The POL bit selects the poll command. The two most commonly used methods to service peripherals in a microprocessor system are polling and interrupts. Although interrupts are the fastest method of servicing peripherals, using the ICU in a polled operation is still faster than polling each peripheral one at a time. Setting the POL bit to a logical 1 enables the reading of the poll status. The POL bit overrides the SR bit if they are both set.

The EXCN and SNM bits can be programmed to enable or disable the exceptional nesting mode of operation. The exceptional nesting mode is armed by setting SNM (Set Nesting Mode) to a logical 1. The EXCN bit can then be used to set or release the

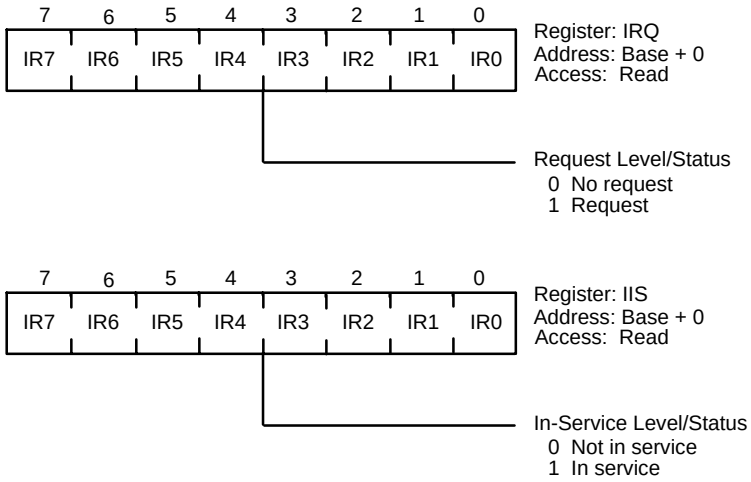
exceptional nesting mode. This operating mode is used to permit interrupts of lower priority than the one currently under service to be recognized.

Status Words (IRQ, IIS, and IPOL)

Three 8-bit status words can be read from the ICU. These are the Interrupt Request (IRQ), Interrupt In-Service (IIS), and Interrupt Poll (IPOL). These words can be read at any time by programming the first three bits of the IMDW. Once the IMDW is programmed to select one of the status words, that word can be read as many times as needed. The IMDW must be programmed with a new value to read another of the status words. The formats of the IRQ, IIS, and IPOL are illustrated in the following topics.

IRQ and IIS

The IRQ and IIS status words, shown in the "Interrupt Status Registers IRQ and IIS" illustration, are taken directly from the Interrupt Request register and Interrupt In-Service register, respectively. The IRQ status word contains all the interrupt levels requesting service. The IIS status word contains all the interrupt levels currently being serviced. Bit 0 of both status words corresponds to IRQ0; bit 1 corresponds to IRQ1, and so on.



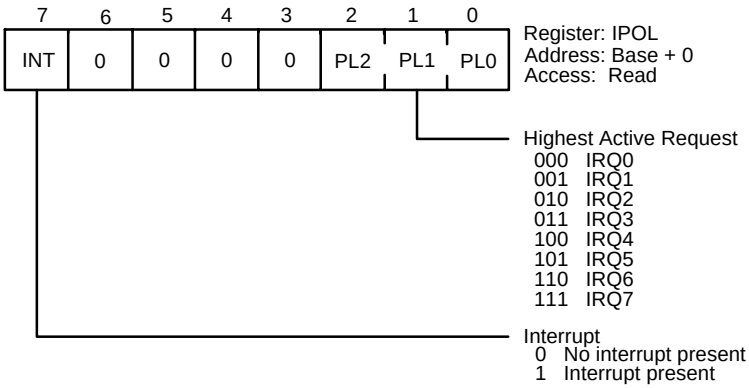
Interrupt Status Registers IRQ and IIS

IPOL

In most applications, polling and interrupts are generally used to service peripherals. The ICU can be used in a polled system to increase the efficiency of servicing peripherals, even though the ICU is designed primarily for interrupt control. The efficiency is increased because the CPU can poll the ICU to determine the status of several peripheral devices at one time.

The "Interrupt Status Register IPOL" illustration shows the IPOL status word. Bits PL0 through PL2 define the highest priority interrupt input requesting service. For example, if all three bits are set to a logical 1, then IRQ7 is the highest priority request.

The INT bit indicates whether there are any interrupt requests. A logical 1 signals an interrupt request and a logical 0 signals no interrupt request. If INT is a logical 0, PL0 through PL2 are all set to logical 1. The typical polling sequence is to set the POL bit in IMDW, read the IPOL register, and test the INT bit. If INT is a logical 1, decode PL0 through PL2 to determine which peripheral to service.



Interrupt Status Register IPOL

ADDITIONAL INFORMATION

Refer to the NEC *16-Bit V Series Microprocessor* data book for more information on the V40 interrupt controller.

CHAPTER 8. DMA CONTROLLER (V40)

This chapter describes the Direct Memory Access Control Unit (DCU) and provides register descriptions.

The DCU is a programmable peripheral device normally used to direct high speed data transfers between memory and I/O. The DCU may also be used in a cascade mode to allow external bus masters access to the system RAM on the ZT 8801. The ZT 8950 Direct Memory Access (DMA) Controller Board is utilized in this fashion.

ZT 8801 SPECIFICS

The ZT 8801 uses one of the four DMA controllers contained in the V40. DMA channel 0 is used in cascade mode to supply the handshake for external STD bus masters. The ZT 8801 supports the BUSRQ* (Bus Request)/BUSAK* (Bus Acknowledge) protocol for transferring ownership of bus resources to an external board. Refer to the following topic for details.

Bus Request/Bus Acknowledge

When channel 0 is programmed in cascade mode, the ZT 8801 supports external bus masters in the STD bus. When a bus master drives the BUSRQ* signal on the backplane, the DCU issues a hold request to the CPU. Upon completion of the current instruction cycle, the CPU releases control to the DCU. The DCU then drives the BUSAK* signal on the backplane, indicating to the STD bus master that it now can perform bus cycles. In this mode, the ZT 8801 allows the bus master access to all on-board memory. When BUSRQ* is released, the ZT 8801 releases BUSAK* and control reverts to the CPU.

The ZT 8950 DMA controller is an example of an STD bus master that uses this protocol. The ZT 8950 also supplies floppy disk capability for DOS systems. Ziatech DOS programs DCU channel 0 for cascade mode to allow the ZT 8950 access to system memory. The DMA controller on the ZT 8950 transfers data between the floppy disk and system memory after gaining control of the bus via the above BUSRQ*/BUSAK* sequence.

PROGRAMMABLE REGISTERS

The DCU occupies 16 consecutive I/O port addresses. Of those 16 addresses, 12 are used to access DCU functions and 4 are reserved. The "DCU Register Addressing" table lists the address of each of the registers relative to a programmable base address. The base address is selected with the OPHA and DULA V40 configuration registers; see the "OPHA, DULA, IULA, TULA, and SULA" section in Chapter 5, "Processor Configuration [V40]," for details.

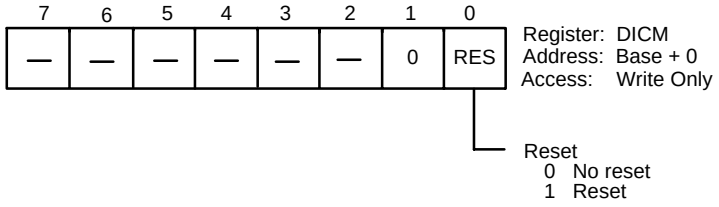
The following topics provide descriptions of the DCU's programmable registers.

DCU Register Addressing

Address	Register	Operation
Base + 0	DICM	Write
Base + 1	DCH	Read/Write
Base + 2	DBC/DCC-low	Read/Write
Base + 3	DBC/DCC-high	Read/Write
Base + 4	DBA/DCA-low	Read/Write
Base + 5	DBA/DCA-middle	Read/Write
Base + 6	DBA/DCA-high	Read/Write
Base + 7	Reserved	--
Base + 8	DDC-low	Read/Write
Base + 9	DDC-high	Read/Write
Base + A	DMD	Read/Write
Base + B	DST	Read
Base + C	Reserved	----
Base + D	Reserved	----
Base + E	Reserved	----
Base + F	DMK	Read/Write

DMA Initialize Command (DICM)

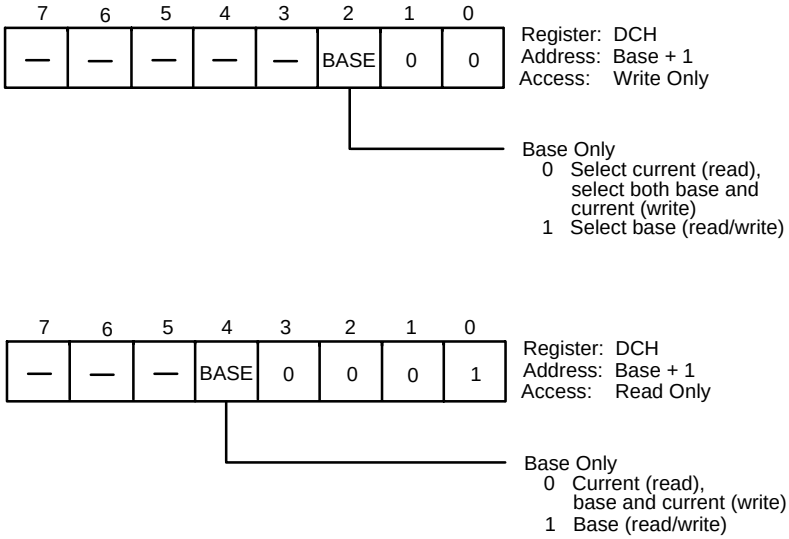
The initialize command, shown in the "DMA Initialize Command Register" illustration, includes one bit that can be set to a logical 1 to reset the DCU. This register must be written to with the byte output instruction.



DMA Initialize Command Register

DMA Channel (DCH)

The DMA Channel register is shown in the "DMA Channel Register" illustration. The DCH register must be accessed with byte output and input instructions. The DCH register has a different format for read and write operations. For the write operation, the BASE bit is used to select between the base and current register groups for both the address and count. Both base and current registers are written or only the current register is read if BASE is first programmed with a logical 0. Programming BASE with a logical 1 selects the base to be read or written to.

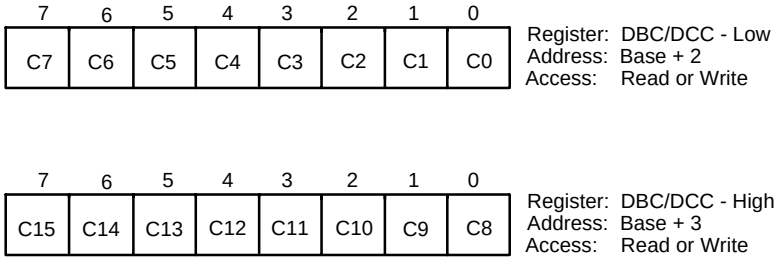


DMA Channel Register

For read operations, the BASE bit set to a logical 0 defines whether the current register is made available for a read operation or whether the base and current registers are written to during a write operation. A logical 1 in the BASE bit means the base register is selected.

DMA Base Count/Current Count (DBC/DCC)

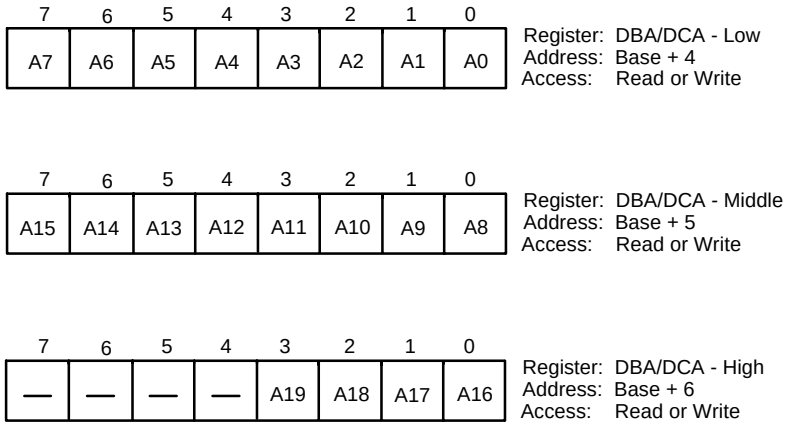
Two DBC/DCC registers make up the 16-bit DMA count, as shown in the "DMA Base And Current Count Registers" illustration. The two DBC/DCC registers can be accessed with byte or word instructions. The function of these registers depends on the BASE bit of the DCH register. If the BASE bit is set to a logical 0, the values written to the Count registers are programmed into both base and current count values. The current count is read from the Count registers if the BASE bit is set to a logical 0. If the BASE bit is set to a logical 1, the values written to the Count registers are programmed into the base value only. The base value is read from the Count registers if the BASE bit is set to a logical 1. In cascade mode, these registers should be programmed to 0.



DMA Base And Current Count Registers

DMA Base Address/Current Address (DBA/DCA)

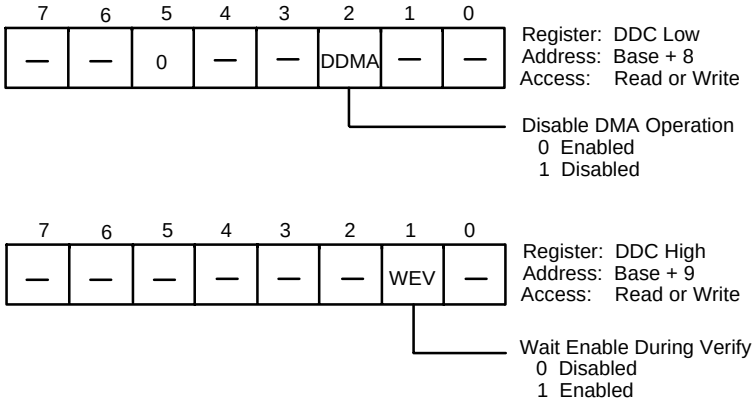
Three DBA/DCA registers specify the 20-bit address. The format of these registers is shown in the "DMA Base And Current Address Registers" illustration. The lower 16 bits of the address can be accessed with byte or word instructions. The upper four bits must be accessed with byte instructions. As is the case with the DBC/DCC registers, the BASE bit of the DCH register defines the operation of the DBA/DCA registers. A logical 0 in the BASE bit specifies that values written to the Address registers are programmed to both base and current values and data read will be current values. If the BASE bit is a logical 1, the values written to the Address registers are programmed into the base value only. The base value is read from the Count registers if the BASE bit is set to a logical 1. In cascade mode, these registers should be initialized to 0.



DMA Base And Current Address Registers

DMA Device Control (DDC)

Two DDC registers select various DCU operating modes. The format for these registers is shown in the "DMA Device Control Registers" illustration. These registers can be accessed with byte or word operations. The DDMA bit can be set to a logical 1 to prevent the DCU from requesting bus access. This should be done when programming any of the DCU registers to prevent incorrect DMA operation.



DMA Device Control Registers

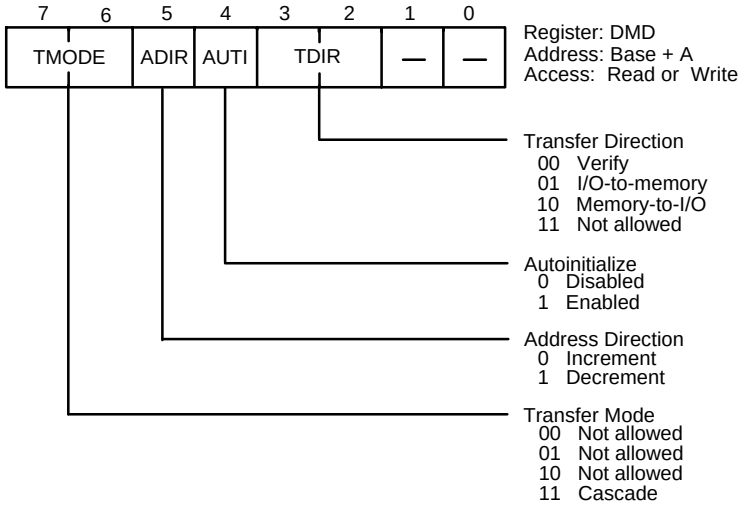
The WEV bit enables or disables wait states to be inserted by the V40 WCU during the verify operation. Programming WEV with a logical 0 disables wait state insertion and programming a logical 1 enables it. The WEV bit has no effect in cascade mode.

DMA Mode (DMD)

The "DMA Mode Register" illustration shows the format of the DMD register. The DMD register can be accessed with byte or word instructions. The TDIR field defines the mode of data transfer. A logical 0 in both bits selects the verify operation. A logical 1 in bit 2 and a logical 0 in bit 3 selects I/O-to-memory transfers. For memory-to-I/O transfers, bit 2 must be programmed with a logical 0 and bit 3 with a logical 1. The ZT 8801 supports only cascade mode operation for channel 0. For cascade mode on channel 0, program this register to C4h.

Autoinitialize is a feature that automatically reloads the DCU current address and count registers from the base address and count registers, respectively. The reload is done when the count register reaches zero. The autoinitialize feature is disabled by programming AUTI with a logical 0 and enabled with a logical 1.

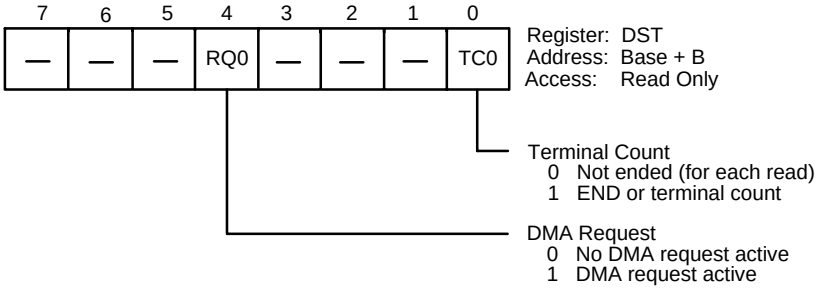
The ADIR bit defines the operation of the DCU address adjuster. If ADIR is programmed with a logical 0, the address adjuster increments the memory address after each data transfer. If ADIR is programmed with a logical 1, the address is decremented. TMODE defines the transfer mode to be demand, single, or block.



DMA Mode Register

DMA Status (DST)

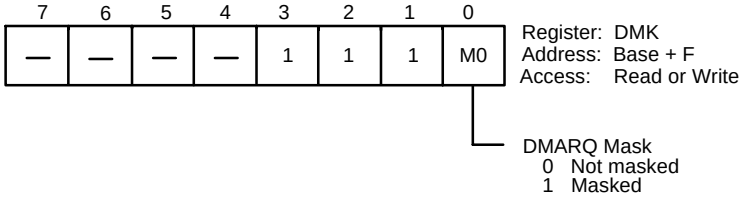
The Status register includes information about the currently programmed state of the DMA channel. The format for DST is shown in the "DMA Status Register" illustration. DST is accessed with the byte read instruction. The TC0 bit indicates when the count register has reached zero and the DMA transfer is completed. A logical 0 in the bit position means the operation has not been terminated and a logical 1 means it has. The RQ0 bit defines the state of the DMA request input. A logical 0 indicates no request active and a logical 1 indicates a request is pending.



DMA Status Register

DMA Mask (DMK)

The DMK register, shown in the "DMA Mask Register" illustration, is used to mask DMA requests made by the DMA channel. The register is accessed with either byte write or read instructions. To mask a DMA channel, the respective bit must be programmed with a logical 1. A logical 0 enables the DMA channel to make requests. Program bit 0 to a logical 0 to enable channel 0 in cascade mode.



DMA Mask Register

ADDITIONAL INFORMATION

Refer to the NEC *16-Bit V Series Microprocessor* data book for more information on the V40 DMA controller.

CHAPTER 9. SERIAL COMMUNICATIONS (V40)

This chapter describes the Serial Control Unit (SCU) and provides register descriptions and baud rate information.

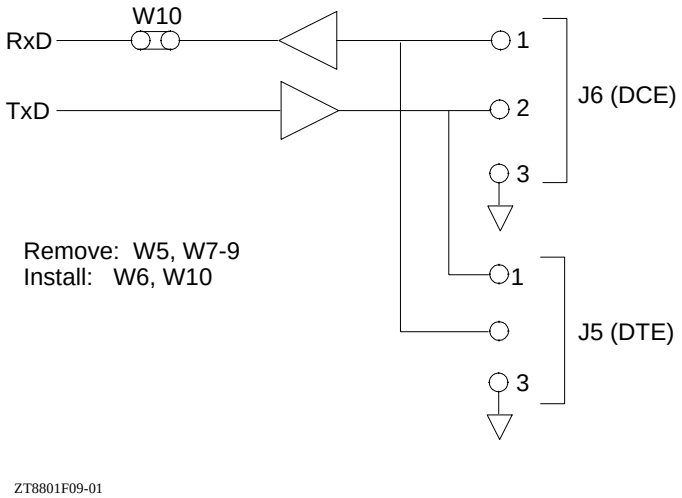
The SCU is a single serial channel that performs asynchronous serial communication between the V40 and a serial device external to the ZT 8801. This serial port is not PC (COM) compatible, but is supported by Ziatech's STD Device Driver Package (DDP) and VSC under DOS or VTI under STD ROM. Applications requiring COM port compatibility must use an external COM port such as is available on the ZT 88CT75.

The major features of the SCU are listed below.

- Full-duplex asynchronous operation
- Clock divisor of 16 or 64
- Baud rates to 19.2 Kbaud
- Programmable character format
- Automatic break detect and handling
- Parity, overrun, and framing detection
- Interrupt and polled operation

RS-232-C OPERATION

The SCU can be implemented as a 3-wire RS-232-C serial port, as shown in the "RS-232-C Configuration" illustration. In RS-232-C mode Transmit Data (TxD), Receive Data (RxD), and Ground (GND) are used. The RS-232-C driver interface allows 5 V-only operation by using a charge pump to create RS-232-C compatible levels.



RS-232-C Configuration

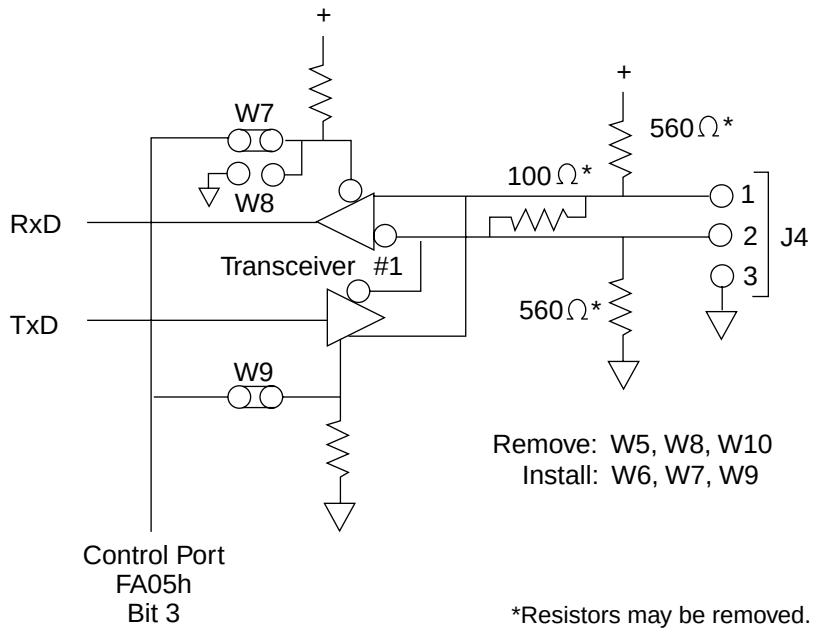
RS-485/422 OPERATION

RS-422/485 compatible drivers and receivers are available to allow the serial port to communicate as an RS-422 or RS-485 device. J3 is a 3-pin connector that is a differential driver which may be optionally gated with the Control Port at FA05h, bit 3. Connector J4 is optionally configured as either a differential transceiver (driver and receiver) or receiver only. It is also optionally gated with the Control Port bit 3. The above combinations allow RS-485 two-wire operation (multidrop), RS-485 four-wire operation, and RS-422 operation (dedicated driver/receiver).

RS-485 2-Wire Operation

J4 is used for 2-wire operation, as shown in the "RS-485 2-Wire Configuration" illustration. Remove W10 to disable the RS-232-C receiver. The multidrop feature is performed under software control of the RS-485 driver via bit 3 of the Control Port (FA05h). Writing a 1 to this bit (preserve the other bits by reading first and ORing this bit in) enables the driver. Clearing this bit (by ANDing out to preserve other bits) disables the driver, allowing another device to drive the bus. The receiver may also be controlled via Control Port bit 3 to avoid receiving data sent.

Note: W8 must be installed and W7 removed so RxD is always driven, preventing the signal from floating to an unknown state. This will allow all data sent to also be received.

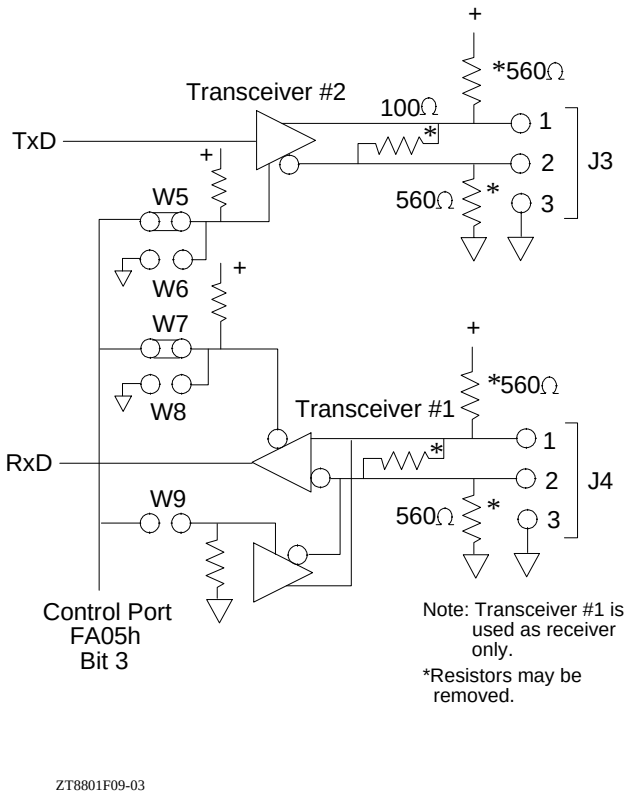


RS-485 2-Wire Configuration

RS-485 4-Wire Operation

RS-485 4-wire operation is accomplished by using J3 as the differential driver and J4 as the differential receiver, as illustrated in the "RS-485/422 4-Wire Configuration"

illustration. The driver may be controlled via bit 3 of the control register if W5 is installed and W6 removed. Writing a 1 to this bit enables the driver. The driver may be always enabled by removing W5 and W6. Remove W9 to disable Transceiver #1's driver. Also remove W10 to disable the RS-232-C receiver.



RS-485/422 4-Wire Configuration

RS-422 Operation

The serial port may be configured as RS-422 by using J3 as the differential driver or J4 as the differential receiver. J3 may be hardwired with the driver always enabled by removing W5 and W6. Remove W9 and W10. Configure J4 as an RS-422 receiver by installing W8 and removing W7.

RS-485/422 TERMINATION RESISTOR CONFIGURATION

The RS-485/422 termination resistors are loaded in sockets along the front edge of the board. These sockets are labeled RP4 and RP5 and are just to the left of J3-J6 when the board is viewed with the goldfingers to the left.

The following topics discuss passive pull-up/pull-down, characteristic impedance, configuring J3/J4 pull-up/pull-down networks, and configuring impedance matching for J3 and J4.

Passive Pull-Up/Pull-Down

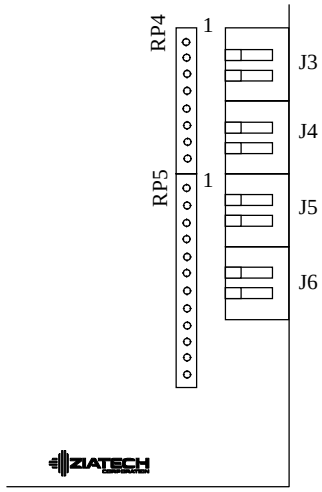
The RS-485/422 interface is nominally a twisted pair differential implementation. The ZT 8801 allows for passive pull-up and pull-down of the differential levels to ensure proper signal levels on undriven signals.

Characteristic Impedance

The characteristic impedance of the twisted pair is 100 ohms, which requires proper transmission line termination to minimize signal reflection. The ZT 8801 allows a 100 ohm resistor to be selectively inserted for both J3 and J4.

Configuring J3, J4 Networks

Both J3 and J4 may have the pull-up/pull-down network installed, as well as the characteristic impedance network, by positioning two resistor packs at locations RP4 and RP5. RP4 is just to the left of J3 and is used to configure impedance matching for both J3 and J4. RP4 consists of eight pins, with pin 1 the pin closest to the board's extractor tab. RP5 is in line with RP4 and is used to configure the pull-up/pull-down network for both J3 and J4. RP5 consists of 12 pins, with pin 1 just below pin 8 of RP4. The "RS-485/422 Termination Resistor Pack Locations" illustration shows the locations of RP4 and RP5.

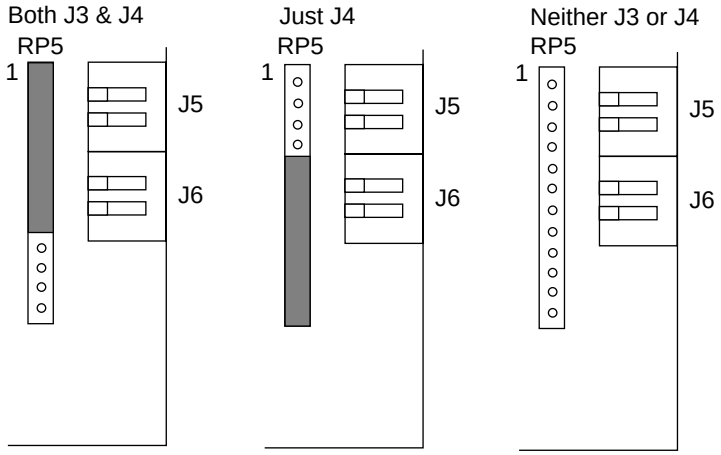


RS-485/422 Termination Resistor Pack Locations

Configuring Pull-Up/Pull-Down Networks for J3 and J4

The default configuration is to pull-up/pull-down both J3 and J4. This is done by loading pin 1 of the 560 ohm resistor pack (an 8-pin resistor pack) at pin 1 of RP5. To terminate only J4, insert the resistor pack justified to the bottom of RP5. This puts pin 1 of the resistor pack at pin 5 of the RP5 socket. To terminate neither J3 nor J4, simply remove

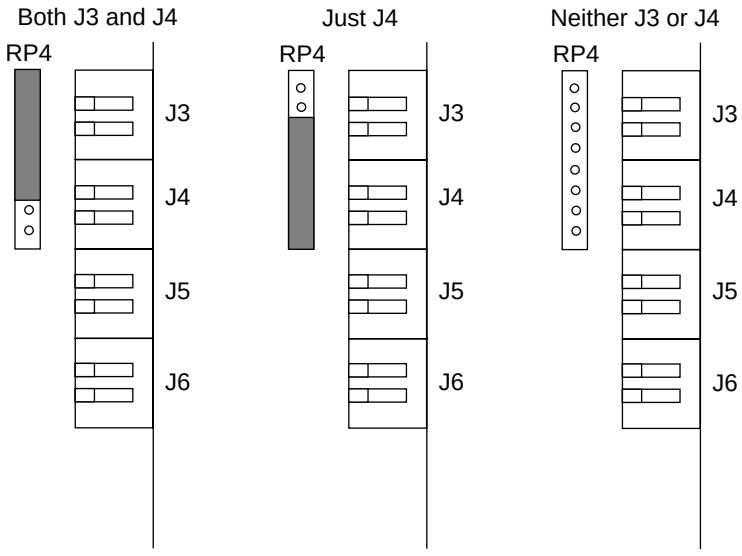
the resistor pack. The figure "RS-485/422 Pull-Up/Pull-Down Configuration" illustrates these three options.



RS-485/422 Pull-Up/Pull-Down Configuration

Configuring Impedance Matching for J3 and J4

The impedance match network is default inserted for both J3 and J4. This is done by loading pin 1 of the 100 ohm resistor pack (a 6-pin resistor pack) at pin 1 of RP4. To match only J4, load pin 1 of the resistor pack at pin 3 of the RP4 location. To match neither J3 nor J4, remove the resistor pack entirely. The figure "RS-485/422 Impedance Matching Configuration" illustrates these three options.



RS-485/422 Impedance Matching Configuration

PROGRAMMABLE REGISTERS

Six registers are used for communication with the SCU. The Serial Transmit Buffer (STB) and Serial Receive Buffer (SRB) store data to be transferred to the serial link and from the serial link, respectively. The Serial Command (SCM) and Serial Mode (SMD) registers define the operating mode. The Serial Interrupt Mask (SIMK) register controls the receive and transmit interrupts. The Serial Status (SST) register provides information on the current state of the transmitter and receiver.

The base I/O address of the SCU registers is defined by the OPHA and SULA registers. OPHA is programmed with the high byte and SULA with the low byte of the 16-bit address; see the "OPHA, DULA, IULA, TULA, and SULA" section in Chapter 5, "Processor Configuration [V40]," for details.

The address of each register, relative to the base address, is shown in the "SCU Register Addressing" table.

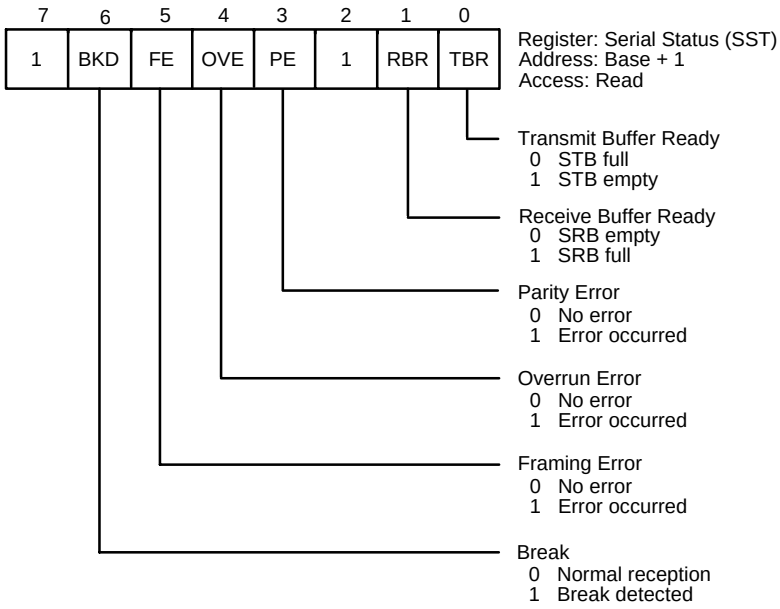
SCU Register Addressing

Address	Register	Operation
Base + 0	Receive Buffer	Read
Base + 0	Transmit Buffer	Write
Base + 1	Status Register	Read
Base + 1	Command Register	Write
Base + 2	Mode Register	Write
Base + 3	Interrupt Mask Register	Read/Write

The following topics provide descriptions of the SCU's programmable registers.

Serial Status Register (SST)

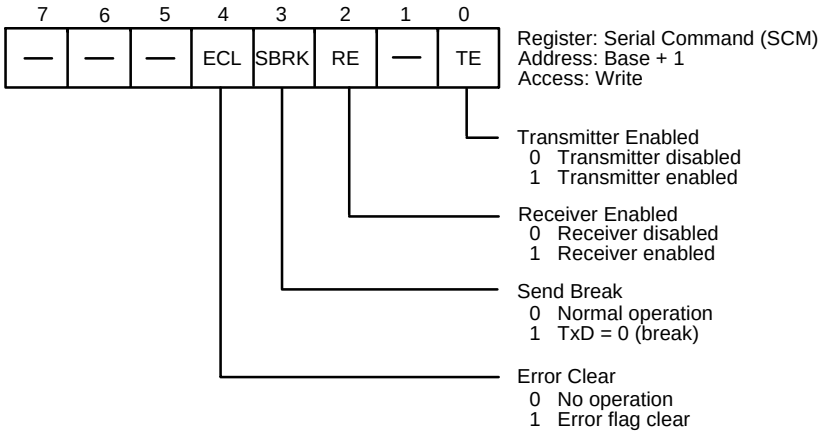
The "Serial Status Register" illustration shows the architecture of the SST register, which can be read at any time.



Serial Status Register

Serial Command Register (SCM)

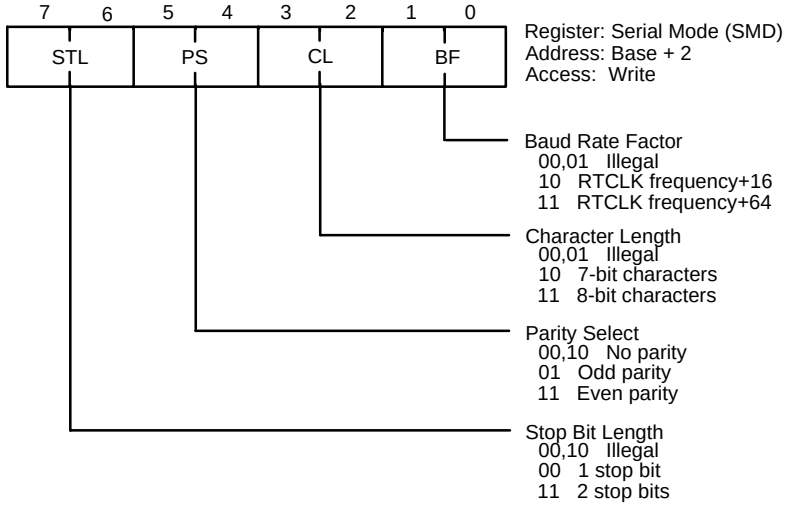
The "Serial Command Register" illustration shows the SCM register bit map. The SCU is configured with the SCM and the SMD registers. The SCM register includes the functions that are most likely to be modified during operation. The SMD register will more than likely be programmed only once for initialization.



Serial Command Register

Serial Mode Register (SMD)

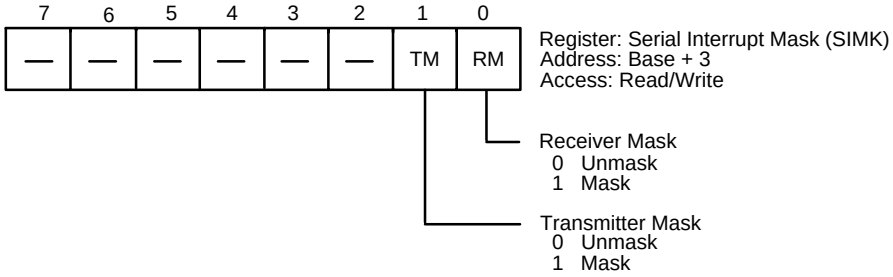
The format for the SMD register is given in the "Serial Mode Register" illusration. The SMD register includes all the functions that are not likely to change after they have been initialized. Bits 1 and 2 combine to define the clock divisor for the baud rate. Programming the baud rate is defined in more detail in the "Baud Rate" section of this chapter.



Serial Mode Register

Serial Interrupt Mask Register (SIMK)

The SCU is capable of interrupting the CPU when a character is received into the Serial Receive Buffer or transmitted out of the Serial Transmit Buffer. The SIMK register includes two programmable bits, as shown in the "Serial Interrupt Mask Register" illustration, to enable the interrupts. Setting the RM bit to a logical 1 prevents the SCU from generating an interrupt when a character is received. A logical 0 in the RM bit enables the interrupt. The TM bit provides the same control for transmitted characters.



Serial Interrupt Mask Register

BAUD RATE

The SCU baud rate is determined by the output of Counter/Timer 1. Counter/Timer 1 must be initialized for a specific mode of operation and programmed with a count that defines the required baud rate. The following discussion explains the initialization and how to calculate the count. Note that 38.4 Kbaud is not supported except in applications that can accept a baud rate not within 4% of the nominal rate (for example, ZT 8801 to ZT 8801).

To use Counter/Timer 1 as a baud rate generator, the TCKS register must specify that Counter/Timer 1 has an internal clock with a divisor determined by the formula below. The TCU Counter/Timer Mode register must also be programmed to configure Counter/Timer 1 for binary operation as a square wave generator (Mode 3). Different baud rates are obtained by programming Counter/Timer 1 with different counts. The relationship between the baud rate and the count is given below:

Count = V40 Clock / (Prescale x Baud Rate x Baud Factor)

This formula includes all the factors that determine the baud rate as a function of the V40 clock. The best way to understand the formula is to trace the clock signal from the V40 clock to the SCU. The V40 clock starts with the following value:

8×10^6 Hz

This frequency is prescaled as defined by the PS bits in the TCKS V40 configuration register. At this point the clock is input to the counter/timers. Counter/Timer 1 further divides the signal by the programmed count before it is input to the SCU. The last division is accomplished internally to the SCU by the baud factor programmed in the BF bits of the Serial Mode register.

As an example, assume the following parameters are specified:

V40 Clock = 8×10^6 Hz

Prescale = divide by two (TCKS PS bits = 00)

Baud Rate = 9600

Baud Factor = divide by 16 (SMD BF bits = 10)

The calculation to determine the count value to be programmed into Counter/Timer 1 is shown below to be 26. Please note that the calculated count and the programmed count differ by 0.16 percent. To guarantee proper operation, the percentage difference must never be greater than four. The "Baud Rate Counts" table lists the count values to be programmed into Counter/Timer 1 to generate the more common baud rates.

Count = $8 \times 10^6 / (2 \times 9600 \times 16) = 26.04$

Baud Rate Counts

BAUD RATE	COUNT DEC/HEX [1] Baud Factor Divided By: 16	64
110	2273/08E1	568/0238
150	1667/0683	417/01A1
300	833/0341	208/00D0
600	417/01A1	104/0068
1200	208/00D0	52/0034
2400	104/0068	26/001A
4800	52/0034	13/000D
9600	26/001A	[2]
19200	13/000D	[2]
38400	[2]	[2]

[1] The count values listed assume the PS bits of the TCKS V40 configuration register are programmed with logical 0s.

[2] This combination exceeds the recommended 4% error allowance.

If the counter/timers are driven with the external TCLK, instead of the V40 clock, the formula for calculating the value to be loaded into Counter/Timer 1 is given below. Note that the frequency of the external clock signal is not dependent on the PS bit of the TCKS register.

Count = External Clock / (Baud Rate x Baud Factor)

ADDITIONAL INFORMATION

Refer to the *NEC 16-Bit V Series Microprocessor* data book for more information on the V40 serial controller.

CHAPTER 10. WATCHDOG TIMER

The primary function of the watchdog timer is to monitor ZT 8801 operation and take corrective action if the ZT 8801 fails to function as programmed. The watchdog timer is a single stage implementation. When enabled, the watchdog timer must be strobed within one of three configurable time periods or a reset is generated to the V40 and system.

The major features of the watchdog timer are as follows.

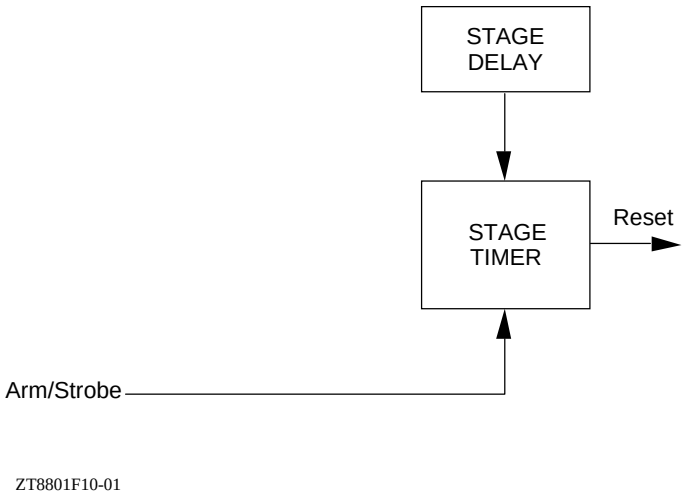
- Enabled and disabled through jumper selection
- Armed and strobed through software control

ZT 8801 SPECIFICS

The watchdog timer must be jumper selected before it is operational. If the watchdog timer is selected, bit 2 of the Control Port (FA05h) becomes dedicated to the watchdog timer and cannot be used for general purpose I/O. It is possible to use an external source to strobe the watchdog by using connector J7, pin 48, as an input and not using the Control Port bit 2 to strobe the watchdog. See the "Programming" section in this chapter for details.

FUNCTIONAL DESCRIPTION

A functional diagram of the watchdog timer is illustrated below. The diagram includes a timer and a delay for the single stage. The functional blocks are described in the following topics.



Watchdog Timer Functional Diagram

Stage Timer

The watchdog timer generates a reset to the local CPU if the watchdog timer is jumper selected (W4 IN) and armed, and a strobe does not occur within the time period defined by the Stage Delay functional block.

Stage Delay

The stage delay has a default range of 400 milliseconds minimum and 2000 milliseconds maximum. The minimum delay time means the watchdog timer must be strobed sooner than 400 ms after the last strobe to prevent a local reset. The maximum delay time means it could take up to 2000 ms before the local reset occurs if the watchdog is not strobed. The stage delay is jumper selectable. The available delay ranges are shown in the "Watchdog Timer Stage Delay Options" table. The lower part of the range is the maximum time software has to strobe the watchdog to guarantee no system reset. The maximum value represents the maximum time the watchdog could take to reset the system.

Watchdog Timer Stage Delay Options

W16	W17	Min.	Max.	Typ.
IN†	OUT	400 ms	2000 ms	1200 ms
OUT	OUT	250 ms	1000 ms	600 ms
OUT	IN	50 ms	250 ms	150 ms
IN	IN	NOT ALLOWED		

†Default. Other options are factory special orders.

OPERATION

In operation, the local CPU is programmed to strobe the watchdog timer at a periodic rate less than the stage time delay. If the local CPU fails to operate as programmed, a local reset is generated.

The following topics discuss reset and programming.

Reset

The watchdog timer is disarmed during and after both a power up and reset condition.

The watchdog timer generates a local reset if allowed to time out. This reset period lasts for up to one second.

Programming

The following topics discuss arming and strobing the watchdog timer with the control port, and strobing the watchdog timer on an external event.

Control Port

The watchdog timer is armed and strobed with bit 2 of the control port at FA05h. Jumper W4 must be in for watchdog operation. The watchdog timer is armed with the following programming sequence:

1. Read the control port at FA05h with interrupts disabled.
2. Logically OR this value with a 4h (set bit 2).
3. Rewrite this new value to port FA05h to arm the watchdog timer. A reset occurs unless the watchdog is strobed before the stage delay time (min.) selected by W16 and W17 expires.

The watchdog timer is strobed with the following programming sequence.

1. Read the control port at FA05h with interrupts disabled.
2. Logically AND this value with an FBh (clear bit 2).
3. Rewrite this new value to port FA05h to strobe the watchdog timer.
4. Logically OR the previous value written in the step above with a 4h (set bit 2).
5. Rewrite this new value to port FA05h to arm the watchdog timer again.

External Strobe

The watchdog timer may be set up to be strobed on an external event. If this is desired, Control Port bit 2 must remain cleared, which is the default after reset.

To use an external source, connector J7, pin 48, is used to connect to the watchdog. W4 must be installed. The external source must drive this pin to 0 V to arm the watchdog. It then must drive this input to 5 V and then 0 V before the minimum time selected by W16 and W17 to strobe the watchdog and avoid reset.

CHAPTER 11. SBX EXPANSION MODULE

The SBX expansion module provides a means to expand the I/O capabilities of the ZT 8801. The expansion module interface is electrically, mechanically, and functionally compatible with the Intel *iSBX MULTIMODULE™ Standard*. This level of compatibility ensures that expansion modules produced by other manufacturers will operate with the ZT 8801. Some of the functions available on expansion modules are as follows (Ziatech product numbers are in parentheses):

- Serial and parallel I/O (zSBX CT32 and zSBX 30)
- Stepper and servo motor controllers
- Analog-to-digital and digital-to-analog converters
- Disk and SCSI controllers
- Modems
- Video controllers
- IEEE 488 controllers (zSBX 20)
- Bar code readers
- Prototyping boards for custom I/O designs (ZT eSBX 70)

FEATURES

The major features of the expansion module interface are listed below.

- Standard interface for expanding I/O capabilities
- Compatible with Intel *iSBX MULTIMODULE Specification*
- Added address lines for custom designs
- Supports both single-wide and double-wide expansion modules

ZT 8801 SPECIFICS

The expansion module interface is supported through connector J8; the pin assignments are given in the "Connectors" section of Appendix B, "Specifications."

The expansion module standard defines three address lines and two chip selects. This provides a total of 16 I/O port addresses. To overcome this limitation, the ZT 8801 expansion module adds four address lines. These address lines are connected in the default configuration and can be removed using cuttable traces; refer to the description of CT1-4 in Appendix A, "Jumper Configurations." The chip select zero signal (MCS0) is

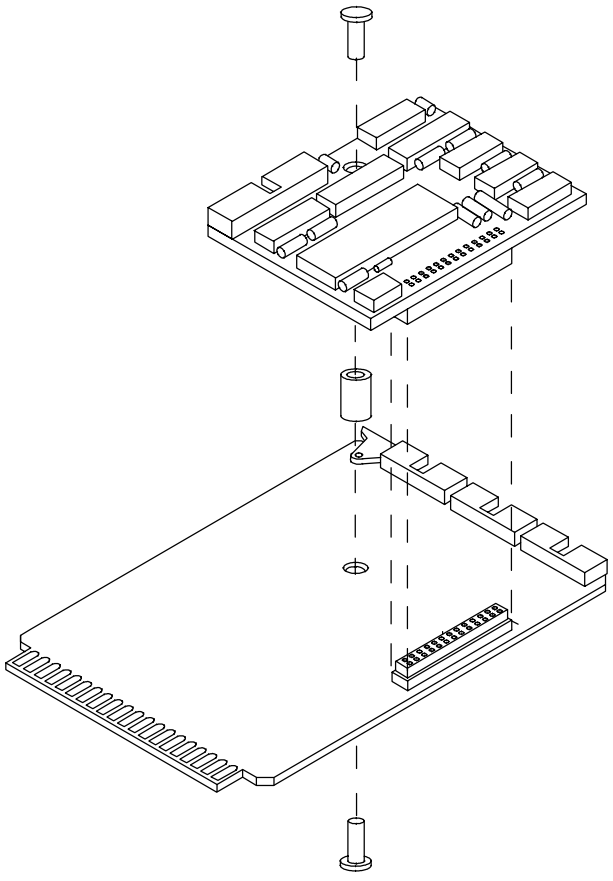
valid over the I/O address range FB00 through FB7Fh, and the chip select one signal (MCS1) is valid over the I/O address range FB80 through FBFFh.

The expansion module supports two interrupt request signals for interrupt driven communications. Interrupt request zero (MINTR0) is connected to either IRQ7 (W23) or IRQ4 (W25) of the interrupt controller, and interrupt request 1 (MINTR1) is connected to either IRQ3 (W27) or IRQ2 (W29) of the interrupt controller. The interrupt controller is explained in detail in Chapter 7, "Interrupt Controller (V40)."

The module present (MPST*) signal is not supported. DMA transfers to the expansion module are not supported.

INSTALLATION

The SBX expansion module is installed on the ZT 8801 as shown in the "SBX Expansion Module Installation" figure. The expansion module is mechanically secured to the ZT 8801 at the J8 connector and with the threaded spacer shipped with the expansion module.

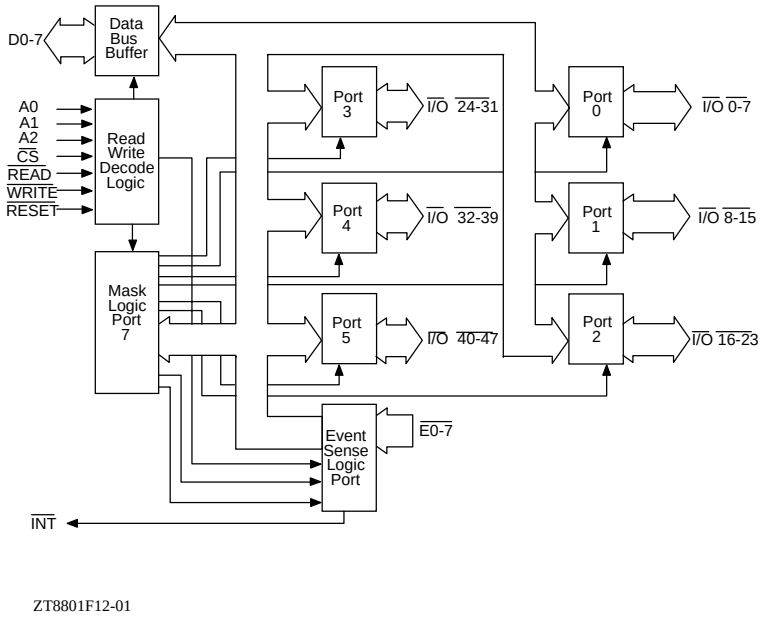


ZT8801F11-01

SBX Expansion Module Installation

CHAPTER 12. PARALLEL I/O ADAPTER (16C49)

With a functional block diagram as the basis for discussion, this chapter summarizes the features of the 16C49 Parallel Interface Adapter (PIA). Topics discussed include the processor interface/connector pinout and the I/O interface. The chapter concludes with a description of methods for interfacing the ZT 8801 to I/O module mounting racks.



16C49 PIA Block Diagram

FEATURES OF THE 16C49

The Ziatech 16C49 is an 84-pin high density Parallel Interface Adapter ASIC device for interfacing to digital I/O and Opto 22 or equivalent industrial I/O racks. The 16C49 includes eight ports: six ports consisting of eight I/O points for a total of 48 I/O points per device, a mask port, and an event sense port. The eight ports are I/O mapped from FA00h to FA07h. Each of the 48 I/O points can be used as an output, output with readback, or input.

To avoid false triggering of external devices, the 16C49 will not glitch outputs during power up or power down. Additional 16C49 features include:

- 48 I/O points (low true)
- Open drain outputs
- 50-300 kΩ internal pullups on I/O points
- Mask register to prevent inadvertent writes to I/O points
- Eight positive or negative event sense inputs with interrupt

- Low power, 1 micron CMOS technology
- -40° to +85° Celsius operation

PROCESSOR INTERFACE/CONNECTOR PINOUT

The figure "16C49 PIA Block Diagram" illustrates how the processor interfaces with the 16C49 PIA and connector J7. A simple non-multiplexed bus interface allows data to be written to or read from one of eight ports within the 16C49. The "16C49 PIA I/O Ports" table illustrates the various I/O addressed ports within the 16C49.

16C49 PIA I/O Ports

Port #	Port Address	I/O Read Register	I/O Write Register	Connector/ Pin #s
0000	FA00h	I/O 0*-7*	I/O 0*-7*	J7/1-8
0001	FA01h	I/O 8*-15*	I/O 8*-15*	J7/10-17
0002	FA02h	I/O 16*-23*	I/O 16*-23*	J7/19-26
0003	FA03h	I/O 24*-31*	I/O 24*-31*	J7/28-35
0004	FA04h	I/O 32*-39*	I/O 32*-39*	J7/37-44
0005	FA05h	I/O 40*-47*	I/O 40*-47*	J7/46-53
0006	FA06h	Event Status	Event Clear	
0007	FA07h	Mask Register	Mask Register	

Notes:

Connector J7 pins 9, 18, 27, 36, and 45 connect to GND. Pins 54, 55, and 56 are connected to +5 V.

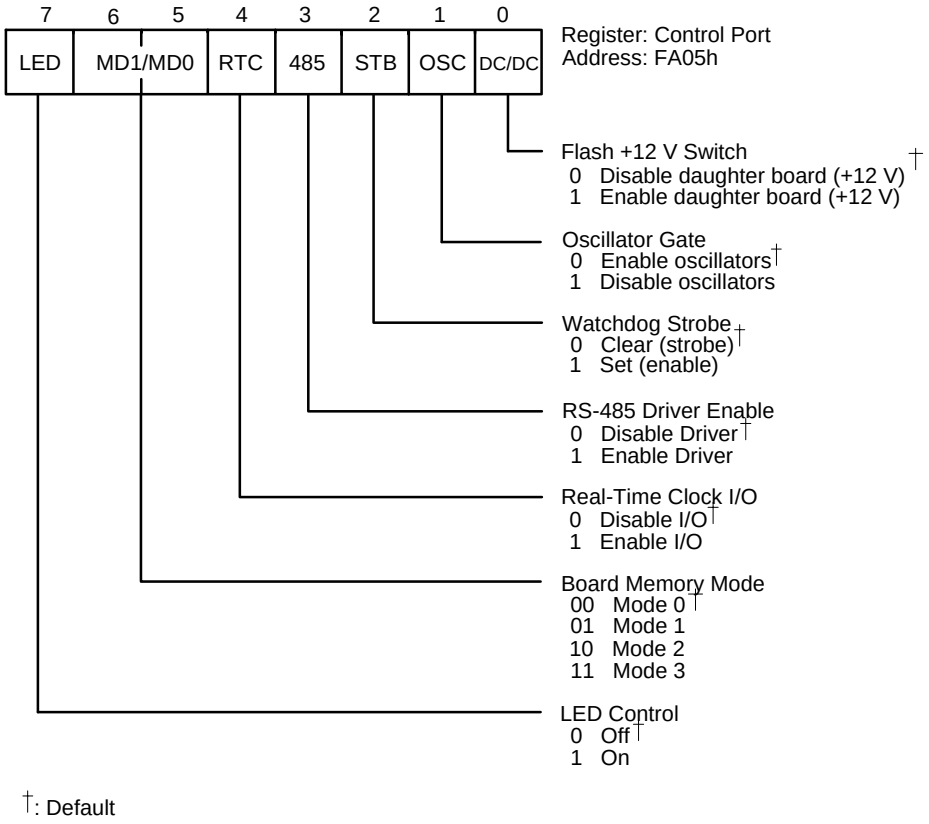
An asterisk indicates an active low signal.

The ports within the 16C49 PIA consist of six I/O ports, an event port, and a mask port. Each port is written to or read from by doing an I/O write or read at the board Base Address of the PIA plus the register address specified in the "16C49 PIA I/O Ports" table. I/O port and mask port operation are covered in the "I/O Port Description And Operation" section of this chapter.

Control Port

The sixth port (at FA05h) is default dedicated for on-board use, but may be configured for off-board I/O at the expense of some on-board features. See the "I/O Point Jumper Options" illustration in Chapter 3, "Theory of Operation," for an illustration of hardwiring versus software control via the control port. The items controlled by these eight bits are RS-485 driver enable/disable control, memory mode selection, LED, watchdog strobe, real-time clock reset, timer and SBX oscillator disabling, and Flash enable for programming. The "Control Port Bit Map" illustration diagrams the bit assignments for these features. Daggers indicate default settings.

We recommend that you disable interrupts when modifying this register to avoid writing stale data.



Control Port Bit Map

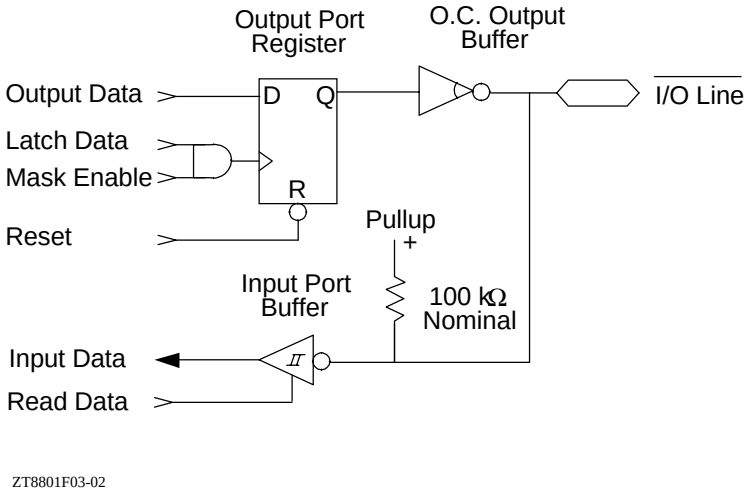
I/O INTERFACE

The following topics discuss I/O circuit operation, I/O circuit specifications, reset operation, I/O port description, I/O module mounting racks, and the Centronics printer interface.

I/O Circuit Operation

Each PIA has 48 I/O points in six ports of eight I/O points each. Each I/O point is illustrated in the "Typical I/O Circuit" figure.

Each typical I/O circuit consists of an output register, open collector output buffer, pullup, and input buffer. Input and output operation are described in the following topics.



Typical I/O Circuit

Outputting Data

Outputting data is accomplished by writing the output data to a given port, thereby causing the latch data signal to capture the output data into the output port register. The register output is buffered by an inverting open collector output buffer before driving the output signal. An integral pullup resistor ensures that a valid high can be measured when the output is not sinking current while in the de-asserted or "off" state.

Note: Each output port has a mask enable bit that can prevent inadvertent writes. The mask enable bits are controlled from the mask port and are unmasked to allow writes after power up or reset.

Outputting Data With Readback

Outputting data with readback can be accomplished by outputting data as described above and then reading the input data, thereby causing the read data signal to enable the input port buffer.

Inputting Data

Inputting data is accomplished by reading the input data from a given port, thereby causing the read data signal to input data from the inverting input port buffer. An integral pullup resistor ensures that a valid input is read if the input signal is not connected.

When inputting data the associated circuit must not be used as an output. This allows the output buffer to be inactive, thereby not contending with the input signal. The output circuits are inactive when a zero is output, after reset, or after power up.

When using a port configured with some output and some input circuits, care must be taken to ensure that any circuits used as inputs are always written with a 0.

I/O Circuit Specifications

The "I/O Point Specifications" table lists the electrical specifications of each I/O point.

I/O Point Specifications

Parameter	Specification
Output Sink Current (Iol)	12 mA min.
Output Low Voltage (Vol at Iol)	.4 V max.
Internal Pullup	50 k ohm - 300 k ohm

Reset Operation

Each I/O circuit on the 16C49 PIA is reset automatically by the reset input. The reset input is connected to a precision nonglitching reset circuit on the ZT 8801 to prevent the output circuits from glitching on power up or power down. The reset circuit is active when the supply voltage is within 0-4.75 V.

After SYSRESET* is asserted, the mask register is initialized to enable (unmasked) writing to the I/O registers.

I/O Port Description and Operation

I/O ports are used to communicate with the 16C49. Six I/O ports are used to output and input data. An event sense port can be used to determine input transitions on I/O points, and a mask register is used to prevent inadvertent writes to the outputs. Each port is described in the following topics.

I/O Port Operation

The six I/O ports implement eight I/O points each. Data bit D0 corresponds to the least significant I/O point in each port. The "I/O Point Write/Read Ports" table lists the port and I/O point assignments for outputs and inputs.

Writing the specified port with the corresponding data bit set (1) causes the I/O output to become active (low). Writing a port with the bit reset (0) causes the I/O output to become inactive (high).

Reading the specified port returns the status of the input point. A high bit (1) indicates the presence of a low true input (0). A low bit (0) indicates a high input (1) is present.

I/O Point Write/Read Ports

Port	D7	D6	D5	D4	D3	D2	D1	D0
0	I/O 7*	I/O 6*	I/O 5*	I/O 4*	I/O 3*	I/O 2*	I/O 1*	I/O 0*
1	I/O 15*	I/O 14*	I/O 13*	I/O 12*	I/O 11*	I/O 10*	I/O 9*	I/O 8*
2	I/O 23*	I/O 22*	I/O 21*	I/O 20*	I/O 19*	I/O 18*	I/O 17*	I/O 16*
3	I/O 31*	I/O 30*	I/O 29*	I/O 28*	I/O 27*	I/O 26*	I/O 25*	I/O 24*
4	I/O 39*	I/O 38*	I/O 37*	I/O 36*	I/O 35*	I/O 34*	I/O 33*	I/O 32*
5	I/O 47*	I/O 46*	I/O 45*	I/O 44*	I/O 43*	I/O 42*	I/O 41*	I/O 40*

* An asterisk indicates an active low signal.

Event Sense Port Operation

The event sense port is used to optionally determine input transitions on I/O points (polarity is selected by the mask port). Event status of the eight event inputs (E0-7) is read and event sense status clearing is done via the event sense port (see the "Event Sense Port" table).

Event Sense Port

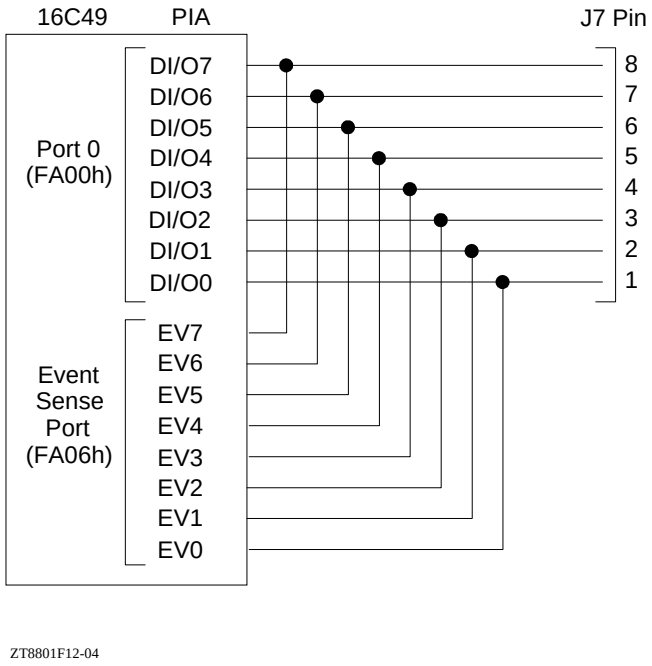
Port	D7	D6	D5	D4	D3	D2	D1	D0
FA06h	E7	E6	E5	E4	E3	E2	E1	E0
J7 pin	8	7	6	5	4	3	2	1

When reading the event sense port, each bit being set to a logical 1 indicates an event on that input has occurred.

When writing the event sense port, each data bit written with a logical 0 clears its corresponding event sense flip-flop. Each data bit of the event sense port must be

written with a 1 to re-enable the corresponding event sense input after it is cleared or after power up or reset.

The event sense inputs are connected to the same signals as Port 0 (FA00h), as shown in the "Event Sense Inputs" illustration.



Event Sense Inputs

Mask Port Operation

The mask register is used to prevent writing the output ports unless enabled. Power-up default has the register unmasked to allow writes to the output ports. Writing the mask register bits D0-5 with a one (1) masks I/O ports 0-5 respectively. Reading the mask port is allowed to determine which ports are enabled.

The "Mask Port" table illustrates the data bit and mask port relationships.

Mask Port

Port	D7	D6	D5	D4	D3	D2	D1	D0
FA07h (write)	E7-4	E3-0	Port 5	Port 4	Port 3	Port 2	Port 1	Port 0
FA07h (read)	Int	0	Port 5	Port 4	Port 3	Port 2	Port 1	Port 0

The upper two bits in the mask register are used in conjunction with the event sense port. When writing, the upper two bits of the mask register select the polarity sensed by event sense inputs E0-E7, which are connected to I/O points 0-7 (active low) on the ZT 8801. Bits 7 and 6 determine E7-4 and E3-0, respectively. Writing a 0 (power-up default) senses negative events (edges), while writing a 1 senses positive events.

When reading the mask register, the most significant bit (D7) returns the interrupt signal status on the interrupt output pin of the 16C49. A logical 1 means the interrupt is asserted.

I/O Module Mounting Racks

The parallel I/O is often used to manage one or two industrial I/O module mounting racks such as those offered by Ziatech (ZT 2226) or Opto 22. This section discusses several methods for interfacing the ZT 8801 to I/O module mounting racks.

Note: Be sure to read Appendix C, "PIA System Setup Considerations," before interfacing I/O module mounting racks to the 16C49 PIA, since the CMOS technology utilized in the PIA can exhibit latchup characteristics under certain conditions. Appendix C illustrates precautions you should take to prevent latchup.

ZT 2226 24-Channel I/O Mounting Rack

The ZT 2226 is a 24-channel I/O module mounting rack that connects directly to the ZT 8801. It holds up to 24 I/O modules and can be panel mounted using its integral mounting hardware. This setup can reduce the size of the control system and free up valuable enclosure space compared to other I/O interface setups.

A 56-pin header on the ZT 2226 provides a direct interface to the parallel port header (J7) on the ZT 8801. The 56-pin header is pin-for-pin compatible, accepts the same cable connector, and has the same electrical specifications as the ZT 8801 header. A typical interface cable is a standard 56-conductor ribbon cable such as the ZT 90089 (40 inches in length). Power for the ZT 2226 can be supplied across the ribbon cable or by a set of terminals that connect to a remote power supply. The 56-pin interface eliminates custom cables, adapter cards, and I/O interface cards.

You can daisy-chain a second ZT 2226 or other 24-channel I/O rack to the first ZT 2226 to provide a total of 48 I/O module positions. To accommodate this, the 48 I/O signals are divided into two sets of 24. One set is connected to the first ZT 2226. The second set is connected to the 50-pin daisy-chain header. The daisy-chain header is connected to a second I/O rack using a 50-conductor ribbon cable such as the ZT 90022 (9.5 inches), ZT 90072 (10 feet), or ZT 90137 (2 feet).

Combining the ZT 8801 and the ZT 2226 creates a compact control system with 24 I/O module positions that can be configured specifically for your application. I/O modules are available from Opto 22 and Grayhill and range in function from DC output types to Quadrature input types. I/O modules are also available with extended operating temperature capabilities. The ZT 2226 provides large 10 AWG terminals to connect field wiring to the I/O modules.

ZT 2223 Industrial I/O Adapter Board

In situations where it is impossible or undesirable to use the ZT 2226, Ziatech recommends the ZT 2223 Industrial I/O Adapter Board. The ZT 2223 converts the ZT 8801's 56-pin parallel I/O interface to two 50-pin, I/O mounting rack compatible connectors with alternating signal and ground lines.

The ZT 2223 is intended primarily for connecting to I/O module mounting racks that do not accept Opto 22 Generation 4® (G4) miniature modules. Manufacturers include Grayhill, Potter & Brumfield, Gordos, Crydom, Analog Devices, and others. The ZT 2223 is not designed to mount to racks using a card-edge connector.

The ZT 2223 interfaces with the ZT 8801's parallel I/O connector (J7) via a standard 56-conductor ribbon cable such as the ZT 90089 (40 inches). This interface eliminates custom cables and I/O interface cards. The ZT 2223's 56-pin header is pin-for-pin compatible, accepts the same cable connector, and has the same electrical specifications as the ZT 8801 header.

The ZT 2223 then plugs directly onto the first I/O mounting rack using a downward-facing 50-pin connector and is secured to the mounting plate with standoffs. A second 50-pin header provides a daisy-chain cable interface for a second I/O mounting rack. Use a 50-conductor ribbon cable such as the ZT 90022 (9.5 inches), ZT 90072 (10 feet), or ZT 90137 (2 feet).

Power for the ZT 2223 is jumper selectable from the 56-conductor ribbon cable.

ZT 2225 Industrial I/O Cable Adapter

As a third option, the ZT 2225 Industrial I/O Cable Adapter can be used to interface the ZT 8801 to one or two I/O module mounting racks. The ZT 2225 plugs directly onto the ZT 8801 J7 parallel I/O connector in a piggyback fashion and accepts one or two ZT 90072 cables (50-pin both ends) or ZT 90021 cables (50-pin header at one end, card-edge connector at the other). Each cable interfaces the ZT 2225 directly to an I/O module mounting rack. The ZT 2225 can be mounted externally to the card cage if an empty card cage slot is not available.

The "ZT 8801 J7 Connection to ZT 2225" table shows the relationship between the I/O port address of each ZT 8801 parallel I/O bit and the associated pin on the ZT 2225.

ZT 8801 J7 Connection to ZT 2225

J7 Pin	Signal	Port Address [hex]	J7 Pin	Signal	Port Address [hex]
1	IO00*/E0*	FA00 bit 0	28	IO24*	FA03 bit 0
2	IO01*/E1*	FA00 bit 1	29	IO25*	FA03 bit 1
3	IO02*/E2*	FA00 bit 2	30	IO26*	FA03 bit 2
4	IO03*/E3*	FA00 bit 3	31	IO27*	FA03 bit 3
5	IO04*/E4*	FA00 bit 4	32	IO28*	FA03 bit 4
6	IO05*/E5*	FA00 bit 5	33	IO29*	FA03 bit 5
7	IO06*/E6*	FA00 bit 6	34	IO30*	FA03 bit 6
8	IO07*/E7*	FA00 bit 7	35	IO31*	FA03 bit 7
9	GND	--	36	GND	--
10	IO08*	FA01 bit 0	37	IO32*	FA04 bit 0
11	IO09*	FA01 bit 1	38	IO33*	FA04 bit 1
12	IO10*	FA01 bit 2	39	IO34*	FA04 bit 2
13	IO11*	FA01 bit 3	40	IO35*	FA04 bit 3
14	IO12*	FA01 bit 4	41	IO36*	FA04 bit 4
15	IO13*	FA01 bit 5	42	IO37*	FA04 bit 5
16	IO14*	FA01 bit 6	43	IO38*	FA04 bit 6
17	IO15*	FA01 bit 7	44	IO39*	FA04 bit 7
18	GND	--	45	GND	--
19	IO16*	FA02 bit 0	46	IO40*	FA05 bit 0 (DC/DC)
20	IO17*	FA02 bit 1	47	IO41*	FA05 bit 1 (OSC)
21	IO18*	FA02 bit 2	48	IO42*	FA05 bit 2 (WDSTB)
22	IO19*	FA02 bit 3	49	IO43*	FA05 bit 3 (485)
23	IO20*	FA02 bit 4	50	IO44*	FA05 bit 4 (RTC)
24	IO21*	FA02 bit 5	51	IO45*	FA05 bit 5 (MD0)
25	IO22*	FA02 bit 6	52	IO46*	FA05 bit 6 (MD1)
26	IO23*	FA02 bit 7	53	IO47*	FA05 bit 7 (LED)
27	GND	--	54	+5 V	-- | 1 A fused
			55	+5 V	-- | 1 A fused
			56	+5 V	-- | 1 A fused

Note: An asterisk (*) indicates an active low signal.

As a word of caution, note that the connector type and pin assignments of different I/O module mounting racks vary significantly. Always compare the ZT 2225 pin assignments given in the "ZT 8801 J7 Connection to ZT 2225" table to those of the I/O module mounting rack to ensure compatibility.

Note also that the ZT 90021 and ZT 90072 digital I/O cables mentioned above supply +5 V power on pin 49. **When ribbon cable power is not desired, be sure the I/O rack ribbon cable power jumper is removed** or use a ZT 2226 I/O module mounting rack.

Centronics Printer Interface

Two methods are used to interface the ZT 8801 to a standard IBM printer cable. The first method uses the ZT 90156, an 8" (20.3 cm) printer interface cable with a 56-pin transition connector on one end and a 25-pin female D-shell connector on the other. The "ZT 90156 Printer Interface Cable" illustration shows the ZT 90156 cable and its pin assignments. If you prefer to make a custom cable, we recommend a length of three feet or less.

The second method by which to interface the ZT 8801 to a Centronics printer is the ZT 2225 Industrial I/O Cable Adapter mentioned earlier. It plugs directly onto the ZT 8801 J7 parallel I/O connector and accepts the ZT 90028 cable, which then interfaces J4 of the ZT 2225 to a standard IBM printer cable. Note that the ZT 2225 can be mounted externally to the card cage if an empty card cage slot is not available.

CHAPTER 13. REAL-TIME CLOCK/CALENDAR (DS 1202)

The function of the real-time clock is to preserve the time of day through the host board's power cycle. During the boot sequence, an operating system reads the real-time clock (RTC) and maintains the time of day for system use. In addition to the time of day, the RTC is responsible for keeping the day, date, and year. Some RTCs also keep track of such things as leap years and end of month rollovers for months shorter than 31 days.

In the case of Ziatech DOS, the system time maintained while power is on is independent of the RTC. One of the timer/counters (Timer 0) generates a periodic interrupt every 54.93 milliseconds for the BIOS to update a memory area that is then used by applications for time or date information. Note that under DOS, when the "time" command is typed at the command prompt, the BIOS does not read the RTC, it reads its own real time, which the timer/counter is updating. The RTC is used to provide the initial time only at boot up.

ZT 8801 SPECIFICS

The ZT 8801 uses the Dallas Semiconductor DS 1202 Serial Timekeeper Chip. Communication is performed with the V40 microprocessor via I/O port FA80h, bit 0. Reads and writes to this bit allow the serial data of the RTC to be communicated. Through this interface, command and data bytes are read and written. This port is redundantly mapped in the I/O space from FA80h through FAFFh. In addition, a Control Port bit (FA05h, bit 4) is used to enable and disable communication with the DS 1202. Note that W2 must be installed for the RTC to be interfaced.

FUNCTIONAL DESCRIPTION

The DS 1202 contains a real-time clock and calendar and 24 bytes of static RAM. The real-time clock subsystem provides seconds, minutes, and hours. The calendar provides day, date, month, and year information. The end of the month is automatically adjusted for months with less than 31 days. In addition, leap year compatibility is provided. The clock can be configured for either 12 or 24 hour mode. In 12 hour mode, an AM/PM indicator is provided. Data may be transferred to and from the RTC one byte at a time or in a burst of up to 24 bytes. The DS 1202 is designed to operate on very low power, using a maximum of 1.2 mA in operation and 300 nA maximum in data retention mode (battery backup).

OPERATION

The real-time clock/calendar (DS 1202) is read and written from FA80h bit 0. The RTC must be enabled for I/O by writing a 1 to bit 4 of the Control Port. Be careful not to alter other bits in the Control Port. In addition, the SBX port (I/O FB00h-FBFFh) must not be read or written while the RTC I/O pin is enabled. The following sequence must be followed for correct RTC operation:

1. Clear interrupts.
2. Enable the RTC I/O pin (set FA05h, bit 4).
3. Re-enable interrupts (optional).
4. Perform RTC I/O (via FA80h, bit 0).
5. Clear interrupts (if enabled in step 3).
6. Disable the RTC I/O pin (clear FA05h, bit 4).
7. Re-enable interrupts.

For example, to set bit 4, the following assembly language code would be used:

```
cli             ;           clear
                interrupts
mov dx, 0Fa05h  ; Control Port
in  al, dx      ; read it
or  al, 010h    ; enable RTC I/O
out dx, al      ;
sti
```

To initiate any data transfer to the RTC, an 8-bit command code is written via FA80h bit 0 after RTC I/O is enabled (via the Control Port FA05h, bit 4). This command byte specifies which of the 32 bytes are accessed, whether a read or write cycle is to be performed, and whether a byte or burst transfer will take place. Following the command byte, subsequent cycles then either read or write data through bit 0 of FA80h. There are 24 bytes (192 bits) of RAM, and the clock/calendar is implemented in 8 bytes (64 bits). After data is read or written, it is important that the RTC I/O pin be disabled via the Control Port. The following code accomplishes this:

```
cli             ;           clear
                interrupts
mov dx, 0Fa05h  ; Control Port
in  al, dx      ; read it
and al, 0EFh    ; disable RTC I/O
out dx, al      ;
sti
```

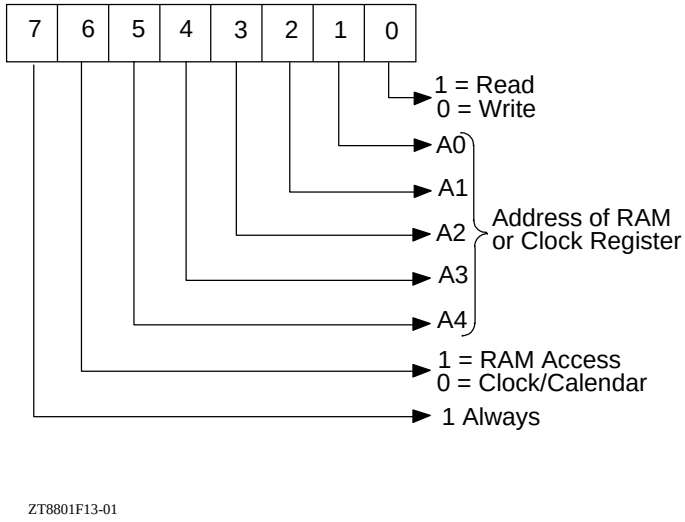
This step must be undertaken before a new location (in byte mode) or a new burst access is undertaken.

COMMAND BYTE

The following topics discuss byte mode, burst mode, data input/output, and the clock/calendar.

Byte Mode

The command byte is shown in the "Command Byte Bit Format (Byte Mode)" illustration. Each transfer is initiated by a command byte. The Most Significant Byte (bit 7) must be a logical 1. If it is a logical 0, further action is terminated. Bit 6 specifies whether the clock/calendar or the general purpose RAM is being accessed. Ziatech operating systems do not use the RAM, so it is available for application use. If this bit is a logical 1, then RAM is to be accessed. If it is a logical 0, then the clock/calendar will be accessed. Bits 1-5 specify which of the 24 RAM locations will be accessed (bytes 0-23). Bytes 24-31 are not provided. Bit 0 specifies a write operation if logical 0 or a read operation if logical 1. The command byte is written starting with bit 0.

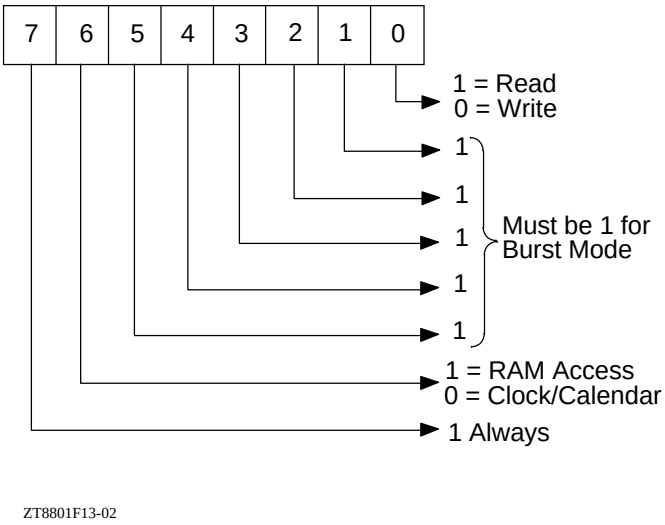


Command Byte Bit Format (Byte Mode)

Byte reads/writes.

Burst Mode

Both the clock/calendar and RAM may be read/written in a software burst by initiating a burst mode command. The burst mode command is the same as a byte command except the address bits (bits 1-5) are all set to logical 1 (see the "Command Byte Bit Format [Burst Mode]" illustration). After the burst mode command has been programmed, all RAM or clock/calendar bytes are read or written as a stream of serial I/O without having to send a command for each location. The 24 bytes of RAM take 192 reads or writes via FA80h to complete the sequence. The eight bytes of clock/calendar data take 64 accesses to FA80h to complete the sequence.



Command Byte Bit Format (Burst Mode)

Burst reads/writes.

Byte Mode Versus Burst Mode

Byte mode is useful when only a portion of the RAM or clock/calendar needs to be accessed, such as the seconds register. For a few bytes only, this mode is more efficient because it does not have the additional overhead of reading and writing the remaining unused bytes as in a burst transfer. A burst transfer is more efficient for applications that utilize most or all of the RAM or clock/calendar. For reference, Ziatech's DOS interfaces to the RTC in byte mode. Depending upon how the interface code is structured, you must decide which mode is more efficient. Remember, we recommend that you disable interrupts during RTC accesses, which may preclude burst mode operation.

Data Input/Output

Following the command byte, the data for the type of command being performed is read or written in either a byte or burst. Again the RTC Data Port (FA80h, bit 0) is used to serially read or write each data bit of each byte. Additional reads or writes beyond that for which the command port is programmed are ignored. Following the data transfer, the RTC I/O pin must be disabled by clearing bit 4 of the Control Port to allow new commands to be written. Data is read or written starting with bit 0 of port FA80h.

An example of code reading from byte 5 of the RAM is given below.

Note that the following code disables interrupts throughout the reading of the RTC. This might not be feasible in some systems due to interrupt latency issues. At a minimum, the Control Port read/write sequence should be protected from interrupts. If interrupts are enabled during the RTC data sequence, be careful during system design to not have interrupt service routines also modify the RTC or read the SBX port.

```
enable_rtc_io:
                  cli             ;            clear
                                  interrupts
                  mov dx, 0Fa05h  ; Control Port
                  in  al, dx      ; read it
                  or  al, 010h    ; enable RTC I/O
                  out dx, al      ;
command_byte:
                  mov     al,             ; command byte CBh
                          11001011b
                  mov     dx, 0FA80h      ; RTC address
                  mov     cx, 8           ; 8 data bits
out_data:
                  out     dx, al          ;
                  ror     al, 1           ; get next bit
                  loop    out_data        ;
                  mov     cx, 8
get_data:
                  in      al, dx
                  ror     ax, 1           ; shift into ah
                  loop    get_data        ; get next bit
                  xchg    al, ah          ;  assume  es  and  di  setup
                                          and
                  stosb                   ; store at ES:[di]
disable_rtc_io:
                  mov dx, 0Fa05h  ; Control Port
                  in  al, dx      ; read it
                  and al, 0EFh    ; disable RTC I/O
                  out dx, al      ;
                  sti
```

Clock/Calendar

The clock/calendar is contained in eight write/read registers. Each byte is accessible individually or as part of a burst read or write. In burst mode, 64 reads or writes are needed to read all eight bytes. The data is in Binary Coded Decimal (BCD) format. The registers are shown in the "Clock/Calendar Registers" illustration.

Command Bytes/Definition

Register	Function	Command Address (HEX)	Write=W Read=R	Range Data (BCD)	Register Definition							
					7	6	5	4	3	2	1	0
0	Seconds	80 81	W R	00-59	Ch	10 Sec			Sec			
1	Minutes	82 83	W R	00-59	0	10 Min			Min			
2	12 Hrs. 24 Hrs.	84 85	W R	01-12 00-23	12\ 24	0 0	AP 10	HR HR	Hour			
3	Date	86 87	W R	01-31	0	0	10 Date		Date			
4	Month	88 89	W R	01-12	0	0	0	10M	Month			
5	Day	8A 8B	W R	01-07	0	0	0	0	Day			
6	Year	8C 8D	W R	00-99	10 Year				Year			
7	Write Protect	8E 8F	W R	00-80	WP	Always Zero						

31	Clock Burst	BE	W
		BF	R
0	RAM 0	C0	W
		C1	R
• • •	• • •	• • •	• • •
23	RAM 23	EE	W
31	RAM Burst	FE	W
		FF	R

ZT8801F13-03

Clock/Calendar Registers

Clock Halt Flag

Bit 7 of the seconds register is used to halt the real-time clock for extreme low-power usage. When this bit is set to 1, the DS 1202 is placed in low-power mode, drawing no more than 100 nA. In ZT 8801 applications, low-power mode is not necessary. The battery used has a shelf life of 10 years, which is equivalent to a usage of 11 microAmp. Assuming neither of the static RAM sockets is being battery backed, the RTC pulls only 1 microAmp maximum, which means the shelf life of the battery is the limiting factor.

AM-PM/12-24 Hour Mode

Bit 7 of the hours register is defined as the 12- or 24-hour mode select bit. When set high, the 12-hour mode is selected. In the 12-hour mode, bit 5 is the AM/PM bit. A logic high indicates PM. In 24-hour mode, bit 5 is the second 10 hour bit (20-23 hours).

Write-Protect Register

Bit 7 of the write-protect register is the write-protect bit. The first seven bits (bits 0-6) are forced to 0 when read. Before any write operation to the clock or RAM, bit 7 must be 0. When high, the write protect bit prevents a write operation to any other register.

Clock/Calendar Burst Mode

The clock/calendar command byte specifies burst mode operation if bits 1-5 are set high. In this mode all eight bytes are read or written starting with bit 0 of register 0. Data is read/written via I/O port FA80h, bit 0.

APPENDIX A. JUMPER CONFIGURATIONS

The ZT 8801 includes jumper options that tailor the board's operation to the requirements of specific applications. Options are selected by installing and removing shorting receptacles (jumpers). In addition, four extra address lines can be added to the expansion module interface using surface mount 0 ohm resistors (cuttable traces).

This appendix provides detailed descriptions of the jumper and cuttable trace options, as well as an illustration of the board showing the locations of all jumpers and cuttable traces.

FACTORY DEFAULT JUMPER CONFIGURATIONS

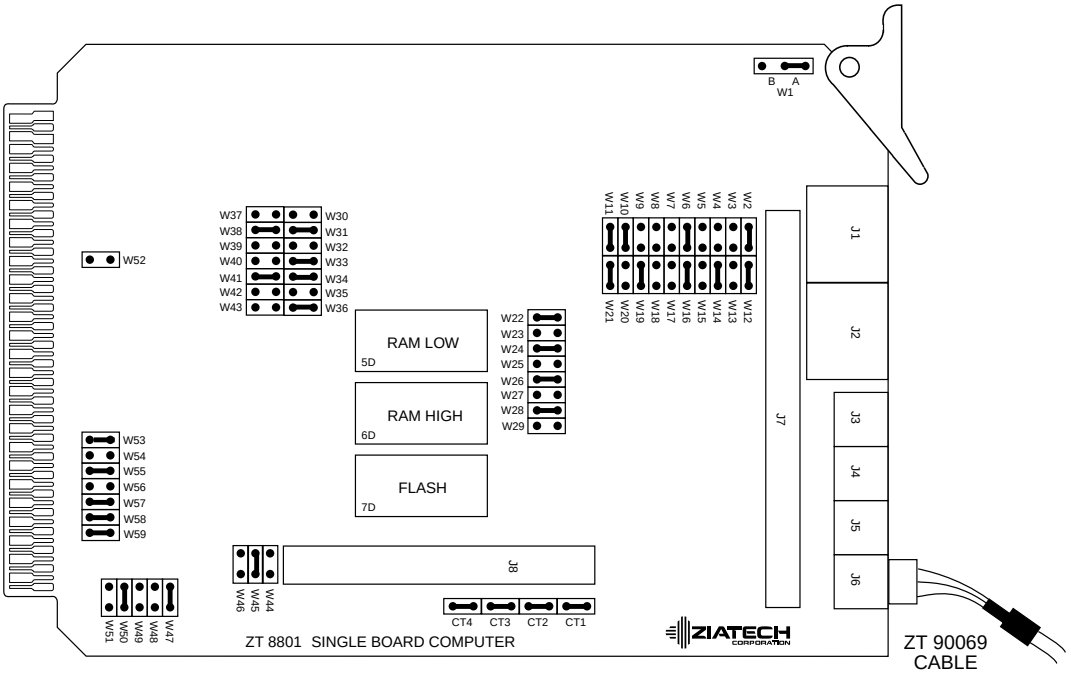
The following figures show default jumper and cuttable trace configurations:

Ziatech DOS: "DOS (128K Flash/256K RAM) Default"

Alternate DOS Configuration: "DOS (512K ROM/1M RAM) Configuration"

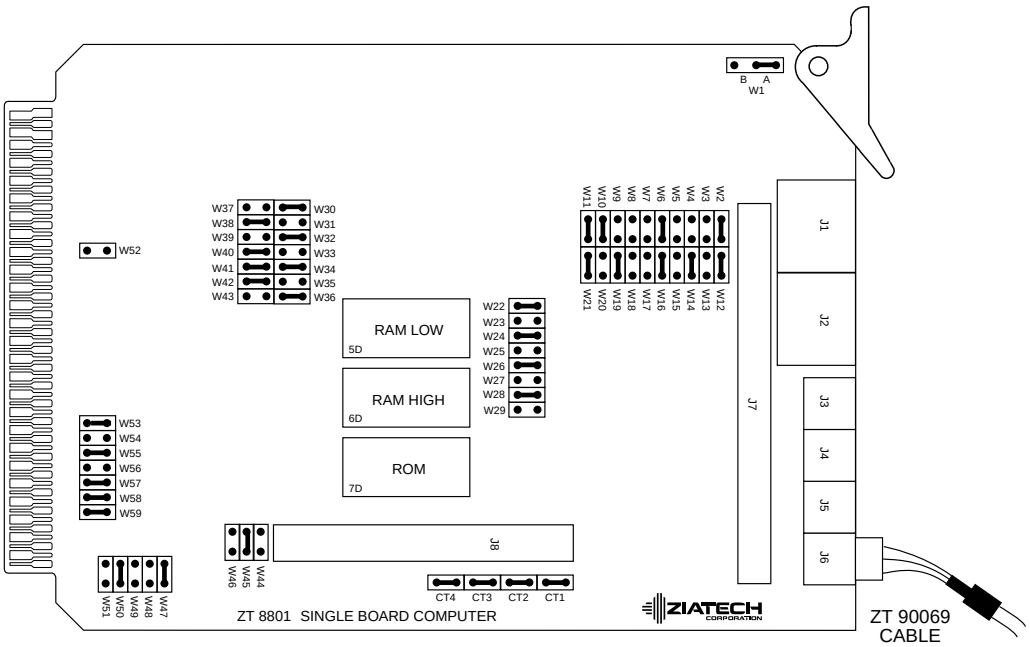
STD ROM: "STD ROM (32K ROM/128K RAM) Configuration"

"Customer Jumper Configuration" illustrates jumper pin locations. You may wish to document your custom jumper configuration using this figure.

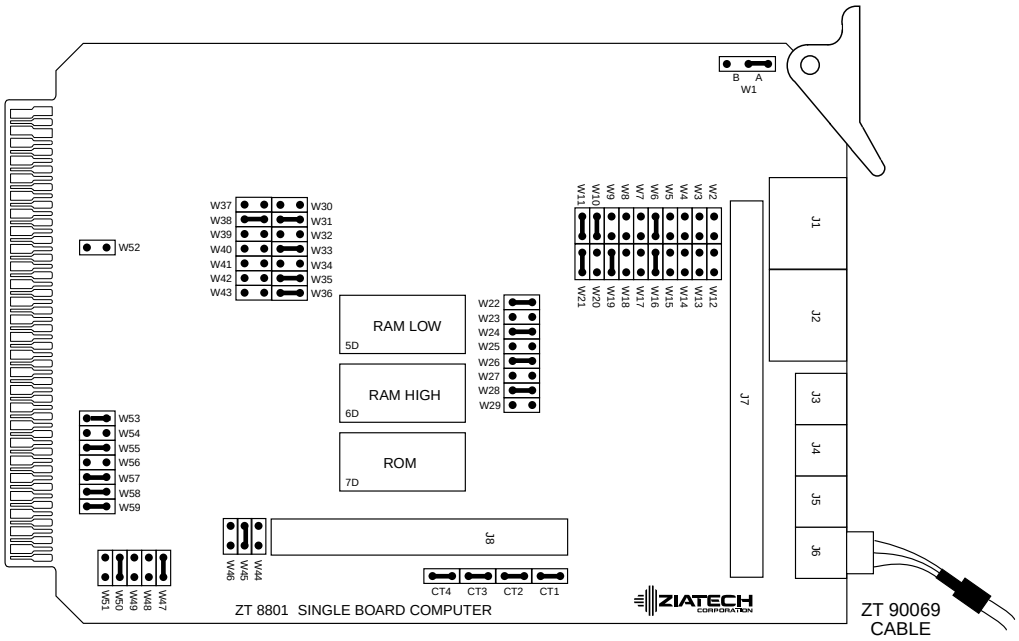


Note: INSTALL W46, W37; REMOVE W45, W36 for Flash in-circuit program capability.

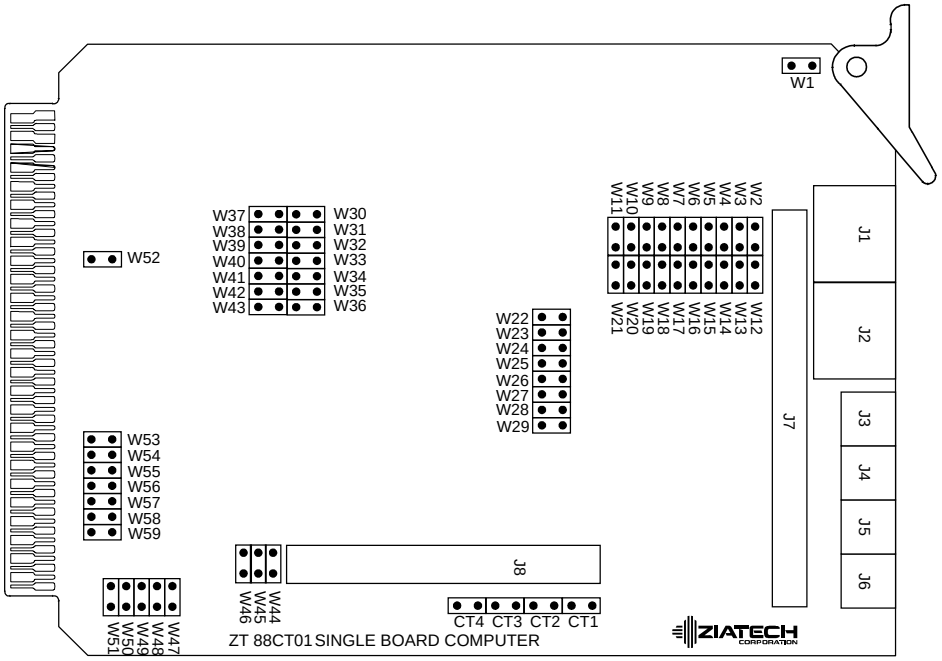
DOS (128K Flash/256K RAM) Default



DOS (512K ROM/1M RAM) Configuration



STD ROM (32K ROM/128K RAM) Configuration



ZT8801FA-04

Customer Jumper Configuration

JUMPER CROSS-REFERENCE TABLE

The table below divides the jumpers into functional groups. Click on any jumper number for a description of the jumper.

Jumper Cross Reference

Function	**Jumper #**
Battery Backup	W1, W18-21
Control Port Config.	W2-4, W43, W12-15
Digital Ground/AUX GND	W59
Interrupts (STD Bus	
and Frontplane)	W47-51, W53-58, W22-29
MEMEX (BHE*)	W52
Memory Configuration:	
28-/32-pin Device Sel.	W34-35
Flash/ROM Config.	W44-46, W36-37
Memory Size	W40-42, W30-33
Reserved (Factory Use)	W11
Serial Port Interface Config.	W5-10
Timer Clock Selection	W38-39
Watchdog Timer Delay	W16-17

JUMPER DESCRIPTIONS

The following topics list the jumpers in the order shown in the "Jumper Cross-Reference" table (alphabetically by function). A dagger (†) indicates a standard default jumper configuration, which is for DOS operation.

Cuttable trace descriptions are at the end of this appendix.

W1,W18-21

Battery Backup Device Selection --- determines if the local RAM devices, RAM LOW and RAM HIGH, are powered by the battery when the system power is turned off. Note that this operation is valid only if the optional battery is present.

The RAM devices should be designed for low power operation (less than 15 µA data retention current). W1A is used to supply battery power to the RAM. With power off, W1B may be installed to erase the battery-backed RAM and real-time clock contents. The RAM HIGH is socket 6D. The RAM LOW is socket 5D. In a DOS system, both sockets should be battery backed as shown. Default installs W1A and removes W1B.

	W18	W19	Battery-Backed Devices
†	OUT	IN	RAM HIGH battery backed
	IN	OUT	RAM HIGH not battery backed

	W20	W21	Battery-Backed Devices
†	OUT	IN	RAM LOW battery backed
	IN	OUT	RAM LOW not battery backed

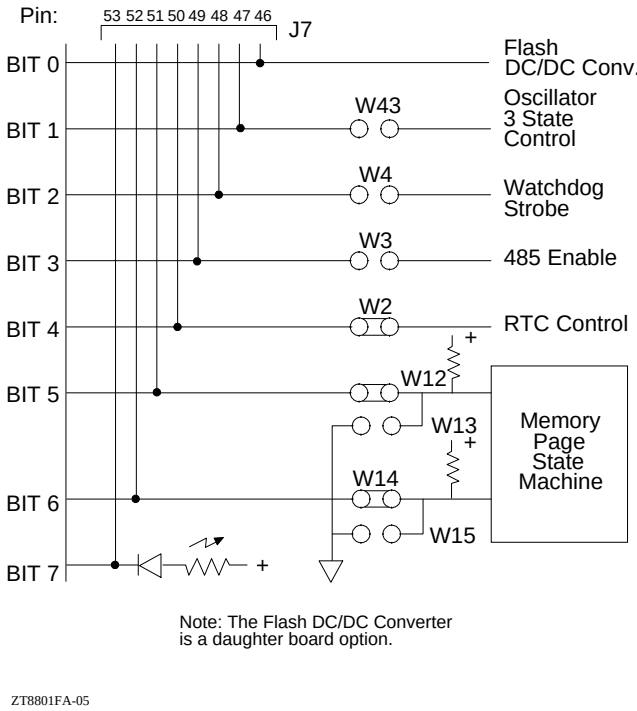
† Factory default (DOS) configuration.

W2-4,W43,W12-15

Control Port Configuration. The Control Port (FA05h) is part of the 16C49 PIA interface. These jumpers allow software control over several features of the ZT 8801, as shown in the "Control Port Configuration" illustration.

Jumper		Function
W2	IN†	Controls real-time clock I/O pin enable. When installed, writing logical 1 to bit 4 of the Control Port enables the I/O pin on D0 of the data bus.
	OUT	Disables RTC I/O.
W3	IN	Allows control of 485 transceivers (see W6-10 description) via bit 3 of the Control Port.
	OUT†	Disables software control of 485 transceivers.
W4	IN	Allows watchdog strobe signal to be driven by bit 2 of the Control Port.
	OUT†	Disables watchdog capability.
W43	IN	Allows Control Port bit 1 to three-state SBX and system timer oscillators.
	OUT†	Enables oscillators.
W12,14	IN†	Allow Control Port bits 5 and 6 (MD0 and MD1 respectively) to drive memory map mode bits. W13 and W15 must be removed.
	OUT	Allow W13 and W15 to be installed.
W13,15	IN	Allow hardwiring of memory map mode bits if W12, W14 removed.
	OUT†	Allow W12 and W14 to be installed.

† Factory default (DOS) configuration; STD ROM systems have W2-4 OUT, W43 OUT, W12-15 OUT.



Control Port Configuration

W59

Digital Ground and AUX GND.

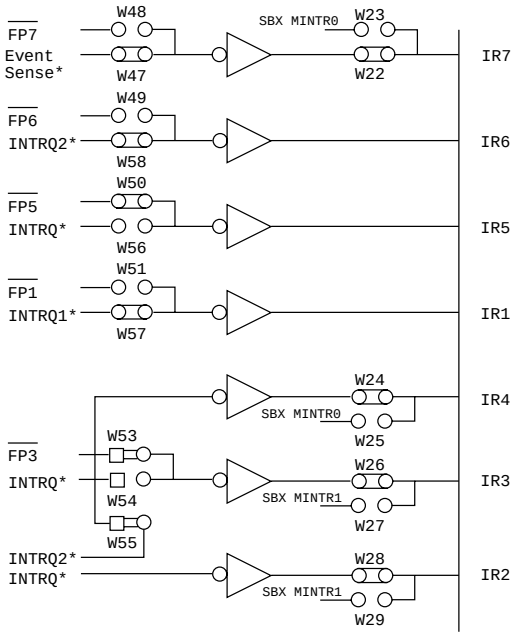
W59	Description
IN †	Electrically connects the AUX ground pins on the backplane to digital ground.
OUT	Allows digital ground and AUX GND to be at different potentials.

W47-51, W53-58, W22-29

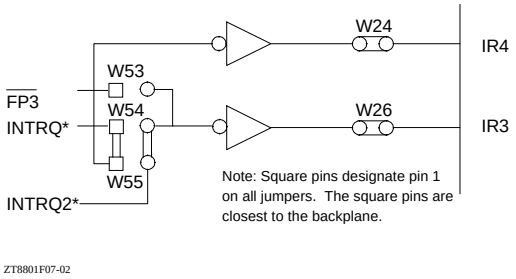
STD Bus and Frontplane Interrupt Selection. These jumpers allow configuration of the interrupt subsystem. Interrupts may be driven by the backplane, frontplane, or expansion module. The defaults are shown in the "Interrupt Jumper Selection" illustration. Note that the IR4 and IR3 levels may be alternately jumpered by using right angle jumpers at W54 and W55; see the "Alternate IR4/IR3 Jumper Selection" illustration.

---

† Factory default (DOS) configuration.



Interrupt Jumper Selection



Alternate IR4/IR3 Jumper Selection

W52

MEMEX (BHE*). This signal is the STD 32 BHE* signal and should not be driven low when using the ZT 8801 with these peripherals.

W52	Description
IN	Forces MEMEX to be driven low (0 V) for all memory boards (some boards need this).
OUT†	MEMEX is pulled up high.

† Factory default (DOS) configuration.

W34,35

Memory Configuration: 28-Pin/32-Pin Device Selection. It is possible to load ROM devices that are smaller than 128K onto the ZT 8801. For cost-sensitive systems, this may be an effective way to reduce costs. These devices are redundantly mapped within the top 128K of memory. STD ROM is shipped in 32K EPROM.

	W35	W34	Device Selection
†	OUT	IN	32-pin ROM/Flash
	IN	OUT	28-pin (for example, 32K ROM)

W44-46, W36-37

Flash Memory/ROM Socket Configuration. These jumpers select the function for pins 1 and 31 at socket 7D. Note that 12 V Flash is not available in 512K memory sizes. Pin 1 of the Flash device can be driven by one of three sources: +5 V (W45), +12 V from the backplane (W46), or an optional +5 V to +12 V DC/DC converter (W44). Install only one of W44-46. W36 and W37 control write or address selection for pin 31 of socket 7D.

	W44	W45	W46	Description
	IN	OUT	OUT	Supply pin 1 of PROM/Flash socket with DC/DC converter output
†	OUT	IN	OUT	Supply pin 1 of PROM/Flash socket with Vcc
	OUT	OUT	IN	Supply pin 1 of PROM/Flash socket with +12 V from backplane

	W36	W37	Description
†	IN	OUT	Supply A18 for 512K PROM socket pin 31
	OUT	IN	Supply write signal to Flash pin 31 (Flash only)

† Factory default (DOS) configuration.

W40-42, W30-33

Memory Size Configuration. These jumpers allow configuration of the RAM and ROM/Flash installed. The RAM HIGH is socket 6D, and the RAM LOW is socket 5D.

	W33	W32	W31	W30	W41	W40	RAM LOW	RAM HIGH
	OUT	IN	OUT	IN	IN	IN	512K	512K
†	IN	OUT	IN	OUT	IN	OUT	128K	128K
	IN	OUT	OUT	IN	OUT	IN	512K	--
	IN	OUT	IN	OUT	OUT	OUT	128K or smaller	--

	W42	ROM/Flash Size
†	OUT	256K or smaller
	IN	512K only

W11

Reserved; factory option only.

W5-10

Serial Port Interface Configuration. These jumpers allow selection of the type of interface for the V40 serial port. Since both RS-485 and RS-232-C receivers can drive the internal RxD signal at the V40, be careful to not enable both at the same time. Disable the RS-232-C receiver by removing W10; disable the RS-485 receiver by removing W7 and W8.

There are two RS-485 transceivers. Transceiver #1 is used for RS-485 2-wire operation using a multidrop configuration. Software control is given by the Control Port at FA05h, bit 3. In RS-485 4-wire operation, transceiver #2 is used as a driver only and transceiver #1 is used as a receiver only. Software control of the driver is given by the Control Port at FA05h, bit 3.

---

† Factory default (DOS) configuration.

The default is for RS-232-C configuration, as shown below.

Note: The following jumper pairs **must not** be installed at the same time:

W5 and W6 must not both be installed.

W7 and W8 must not both be installed.

W8 and W10 must not both be installed.

W7 and W10 must not both be installed.

Removing W7 and W8 (which is default) always disables Transceiver #2's receiver. See the “RS-485 2-Wire Operation” topic in chapter 9 for more information.

The following figures illustrate the various configurations:

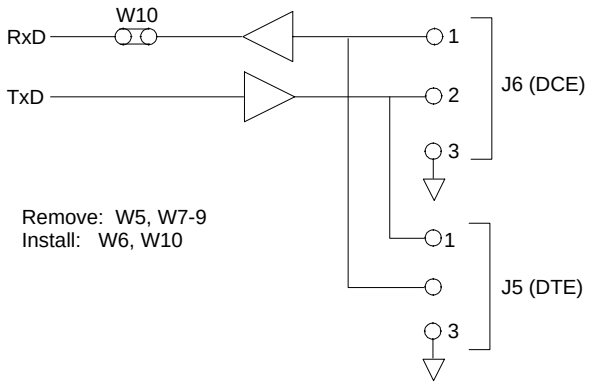
"RS-232-C Configuration"

"RS-485 2-Wire Configuration"

"RS-485 4-Wire Configuration"

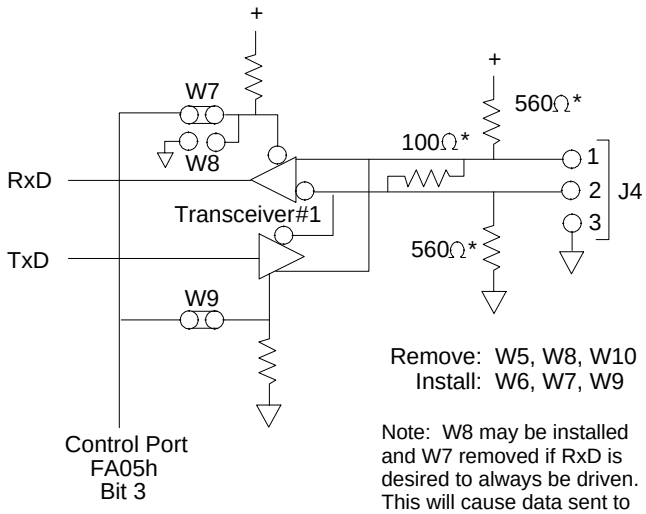
	Jumper	Config.	Description
†	W5	IN	Allows software control of Transceiver #2 driver
	W5	OUT	Allows W6 to be installed
†	W6	IN	Disables Transceiver #2 driver always
	W6	OUT	Allows software control if W5 installed
†	W7	IN	Allows software control of Transceiver #1 receiver
	W7	OUT	Allows W8 to be installed
†	W8	IN	Enables Transceiver #1 driver always
	W8	OUT	Allows software control if W7 installed
†	W9	IN	Allows software control of Transceiver #1 driver
	W9	OUT	Always disables Transceiver #1 driver
†	W10	IN	Allows RS-232-C receiver to drive RxD
	W10	OUT	Allows RS-485 receiver to drive RxD

† Factory default (RS-232-C) configuration.



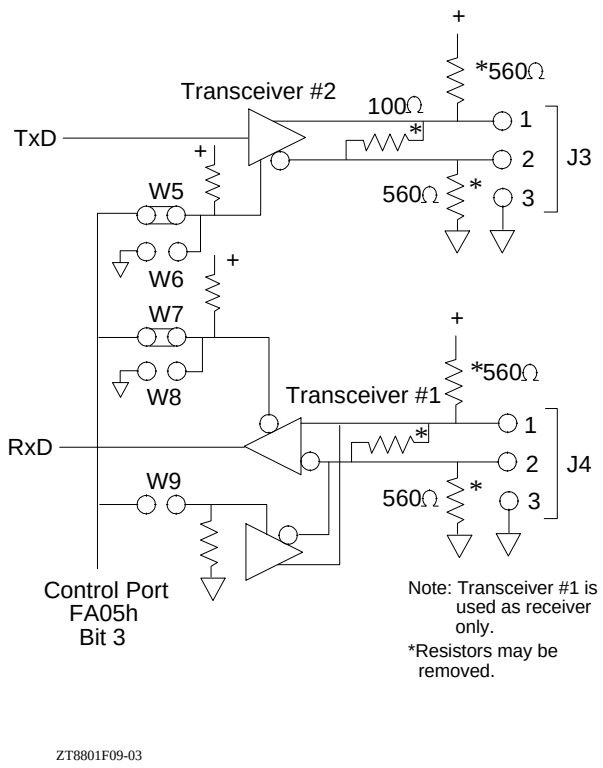
ZT8801F09-01

RS-232-C Configuration



ZT8801F09-02

RS-485 2-Wire Configuration



RS-485 4-Wire Configuration

W38,39

Timer Clock Selection. The V40 counter/timers have two sources for the TCLK input. For DOS systems, this clock is driven by a 1.19318 MHz frequency for compatibility. Alternately, this clock input can be driven from frontplane connector J2, pin 2.

	W39	W38	TCLK Input Source
†	OUT	IN	1.19318 MHz
	IN	OUT	Selected by J2, pin 2.

W16,W17

Watchdog Strobe Interval. These jumpers allow configuration of the watchdog timer interval, which is selectable from one of three periods. This time interval defines the amount of time that an application can wait to strobe the watchdog timer before a system reset is generated. The minimum time of each range is to be used for software

† Factory default (DOS) configuration.

planning. The maximum of each range defines how long the application must wait for a reset if the integrity of the system is lost.

	W16	W17	Description
†	IN	OUT	Watchdog interval 400-2000 ms, 1200 ms typical
	OUT	OUT	Watchdog interval 250-1000 ms, 600 ms typical
	OUT	IN	Watchdog interval 50-250 ms, 150 ms typical
	IN	IN	*Not allowed!* Shorts Vcc to ground

† Factory default (DOS) configuration.

CUTTABLE TRACE DESCRIPTIONS

The ZT 8801 supports the less frequently selected option of SBX module address expansion with the addition or removal of four 0 ohm surface mount resistors, CT1-CT4. Adding or removing cuttable traces requires a solder iron and someone qualified to use it on small surface mount devices.

The following topic provides a detailed description of CT1-CT4. A dagger (†) indicates the default configuration. See the figure “DOS 128K Flash 256K RAM Default” for an illustration showing cuttable trace locations. Factory default installs CT1-CT4.

CT1-4

SBX Module Address Expansion --- increase the number of I/O port addresses available to the SBX expansion module by increasing the number of address lines connected to the expansion module socket. The Intel *iSBX MULTIMODULE Standard* defines only three address lines, A0, A1, and A2; factory default for the ZT 8801 connects A0-A6. The extra four address lines, A3-A6, are connected via cuttable traces CT1-CT4, as shown below.

To remove an address line, cut the corresponding trace. To reconnect an address line, solder a shorting wire across the trace.

Chapter 11, "SBX Expansion Module," provides a detailed description of the expansion module interface, and the description of connector J8 in Appendix B, "Specifications," shows the expansion module pinout.

	CT1	CT2	CT3	CT4	Address Lines Connected
	Out	Out	Out	Out	A0-A2 only
	Out	Out	Out	In	A0-A3 (A3 = J8 pin 30)
	Out	Out	In	In	A0-A4 (A4 = J8 pin 28)
	Out	In	In	In	A0-A5 (A5 = J8 pin 10)
†	In	In	In	In	A0-A6 (A6 = J8 pin 24)

† Factory default configuration.

APPENDIX B. SPECIFICATIONS

This appendix contains the electrical, environmental, and mechanical specifications for the ZT 8801 and ZT 88CT01 Single Board V40 Computers.

ELECTRICAL AND ENVIRONMENTAL

This section provides tables showing the following:

- Absolute maximum ratings
- DC operating characteristics
- Digital I/O interface characteristics
- Battery backup characteristics
- STD bus loading characteristics

Absolute Maximum Ratings

Supply Voltage, Vcc: 0 to 7 V

Supply Voltage, AUX +V: 0 to 13 V

Supply Voltage, AUX -V: 0 to -13 V

Storage Temperature

ZT 8801: -40°to +85° Celsius

ZT 88CT01 (w/o battery): -50° to +125° Celsius

ZT 88CT01 (with battery): -40°to +100° Celsius

Operating Temperature

ZT 8801: 0° to +65° Celsius

ZT 88CT01: -40°to +85°Celsius

Non-condensing relative humidity: <95% at 40°Celsius

DC Operating Characteristics

Supply Voltage, Vcc: 4.75 to 5.25 V

Supply Voltage, AUX +V (Flash): 11.4 to 12.6 V

Supply Voltage, AUX -V: Not Required

Supply Current, Vcc (without SBX)

ZT 8801: 0.6 A typ, 1.0 A max

ZT 88CT01: 0.4 A typ, 0.8 A max

Supply Current, AUX +V (Flash): 0.03 A max

Supply Current, AUX -V: Not Required

Note: Current requirements measured with 256K Flash memory and 256K of RAM.

Digital I/O Interface Characteristics

V_{OL} (max) at I_{OL} of 12 mA (max): 0.4 V (max)

V_{OH} (min) at I_{OH} of 4 mA (max): 3 V (min)

Battery Backup Characteristics

Supply Voltage, Vcc: 4.49 V max

Retention Time

Real-Time Clock only: 10 years min. (90,000 hrs.)

Real-Time Clock and Local RAM: 12,500 hrs. min., 90,000 hrs. typical

(Assuming RAMs have combined retention current of 80 µA)

STD Bus Loading Characteristics

The unit load is a convenient method for specifying the input and output drive capability of STD bus cards. With this method, one unit load is equal to one LSTTL load as follows:

- Current for single input load: 20 µA
- Current for single output drive: -400 µA

The unit load reflects current requirements at worst case conditions over the recommended supply voltage and ambient temperature ranges. An output drive of 60 unit loads drives 60 STD bus cards having input ratings of one unit load.

The tables "Bus Signal Loading, P Connector" and "Bus Signal Loading, E Connector" show load values for STD-80 and STD 32 connections, respectively.

Bus Signal Loading, P Connector

PIN (CIRCUIT SIDE) / OUTPUT DRIVE / INPUT LOAD / MNEMONIC							PIN (COMPONENT SIDE) / OUTPUT DRIVE / INPUT LOAD / MNEMONIC
+5 VOLTS [A]			2	1			+5 VOLTS [A]
GROUND			4	3			GROUND
DCPDN*	0	60	6	5	-		VBATT
D7/A23 [5]	2	58	8	7	58	2	D3/A19 [1]
D6/A22 [5]	2	58	10	9	58	2	D2/A18 [1]
D5/A21 [5]	2	58	12	11	58	2	D1/A17 [1]
D4/A20 [5]	2	58	14	13	58	2	D0/A16 [1]
A15	1	59	16	15	59	1	A7
A14	1	59	18	17	59	1	A6
A13	1	59	20	19	59	1	A5
A12	1	59	22	21	59	1	A4
A11	1	59	24	23	59	1	A3
A10	1	59	26	25	59	1	A2
A9	1	59	28	27	59	1	A1
A8	1	59	30	29	59	1	A0
RD*	1	59	32	31	59	1	WR*
MEMRQ*	1	59	34	33	60	0	IORQ*
(MEMEX) BHE*	0	60	36	35	60	0	IOEXP
MCSYNC* (ALE*)	1	59	38	37	-	1	INTRQ1*
STATUS 0*	1	59	40	39	59	1	STATUS 1*
BUSRQ*	1	59	42	41	60	0	BUSAK* [2]
INTRQ*	1	-	44	43	60	0	INTAK*
NMIRQ*	1	-	46	45	59	1	WAITRQ*
PBRESET*	1	-	48	47	60	0	SYSRESET* [3]
INTRQ2* (CNTRL*)	1	-	50	49	60	0	CLOCK* [3]
PCI [4]	0	-	52	51	-	0	PCO [4]
AUX GND			54	53			AUX GND
AUX-V			56	55			AUX+V

Notes:
An asterisk (*) indicates a low level active signal.
[1] High order address bits multiplexed over data bus.
[2] BUSAK* is an output in permanent master configuration only.
[3] SYSRESET* and CLOCK* are outputs in permanent master configuration and inputs in temporary master configuration.
[4] PCI connected to PCO.
[5] A20-A23 are always driven to logical 0 on the STD bus for memory cycles.

Bus Signal Loading, E Connector

PIN (CIRCUIT SIDE)					PIN (COMPONENT SIDE)
OUTPUT DRIVE					OUTPUT DRIVE
INPUT LOAD					INPUT LOAD
MNEMONIC					MNEMONIC
LOCK*	0	E2	E1		GND
XA23	0	E4	E3	0	XA19
XA22	0	E6	E5	0	XA18
XA21	0	E8	E7	0	XA17
XA20	0	E10	E9	0	XA16
RSVD	0	E12	E11	0	NOWS*
+5V		E14	E13		+5V
MREQx*	0	E16	E15	0	MAKx*
GND		E18	E17		GND
D31	0	E20	E19	0	D27
D30	0	E22	E21	0	D26
D29	0	E24	E23	0	D25
D28	0	E26	E25	0	D24
GND		E28	E27	0	D23
D15	0	E30	E29	0	D22
D14	0	E32	E31	0	D21
D13	0	E34	E33	0	D20
D12	0	E36	E35		GND
D11	0	E38	E37	0	D19
D10	0	E40	E39	0	D18
D9	0	E42	E41	0	D17
D8	0	E44	E43	0	D16
MASTER16*	0	E46	E45		GND
AENx*	0	E48	E47	0	IRQx
BE3*	0	E50	E49	0	BE1*
BE2*	0	E52	E51	0	BE0*
GND		E54	E53	0	MEM16*
W-R	0	E56	E55	0	M-IO
DMAIOR*	0	E58	E57	0	DMAIOW*
EX8*	0	E60	E59	0	IO16*
START*	0	E62	E61	0	CMD*
EX32*	0	E64	E63	0	EX16*
T-C	0	E66	E65	0	EXRDY
+5V		E68	E67	0	INTRQ3*
DREQx*	0	E70	E69	0	DAKx*
MSBURST*	0	E72	E71	0	SLBURST*
XA31*	0	E74	E73	0	XA27*
XA30*	0	E76	E75	0	XA26*
XA29*	0	E78	E77	0	XA25*
XA28*	0	E80	E79	0	XA24*

Note: An asterisk (*) indicates a low level active signal.

MECHANICAL

This section includes the following mechanical specifications:

- Card dimensions and weight
- Connectors (including connector locations, descriptions, and pinouts)
- Cables showing dimensions for building cables to interface with the ZT 8801

Card Dimensions & Weight

The ZT 8801 meets the *STD-80 Series Bus Specification and Designer's Guide* for all mechanical parameters. In a card cage with 0.625 inch spacing, the ZT 8801 requires one card slot without an SBX expansion module installed and two card slots with an SBX expansion module installed. Mechanical dimensions are shown in the "Card Dimensions" illustration and outlined below.

Board Length: 16.51 ±0.063 cm (6.500 ±0.025 in)

Board Width: 11.43 ±0.038 cm (4.500 ±0.015 in)

Board Thickness: 0.158 ±0.013 cm (0.062 ±0.007 in)

Board Weight: 200 g (7 oz)

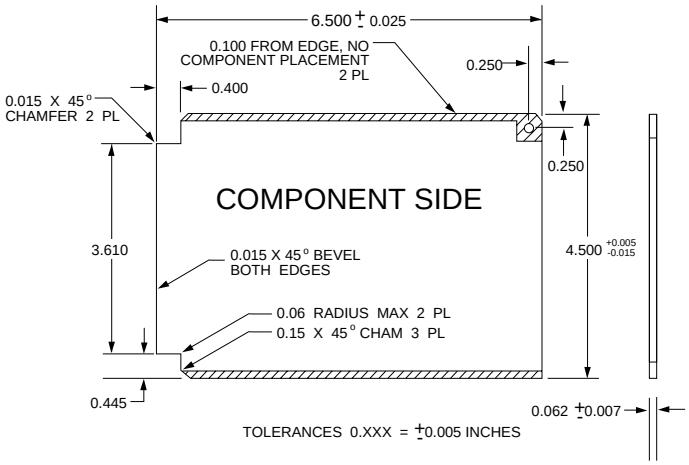
Board Height From Top Surface

Without following items: 9.65 mm (0.38 in) max

Including Battery: 12.12 mm (0.48 in) max

Including SBX Expansion Module: 27.94 mm (1.10 in) max

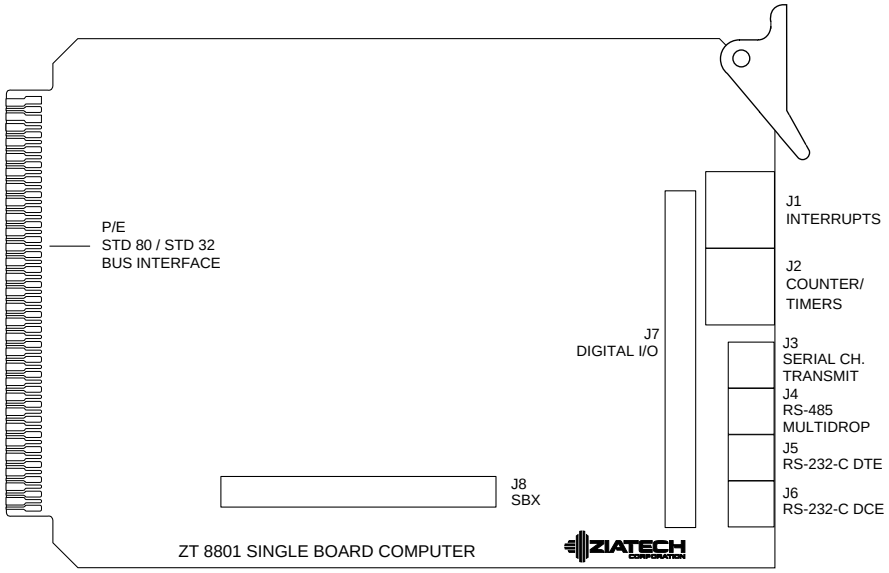
Board Height From Bottom Surface: 1.14 mm (0.045 in) max



Card Dimensions

Connectors

The ZT 8801 includes 10 connectors to interface to the STD bus and application-specific devices. A description of each connector is given in the following topics. A description and pin map for each connector is given below.



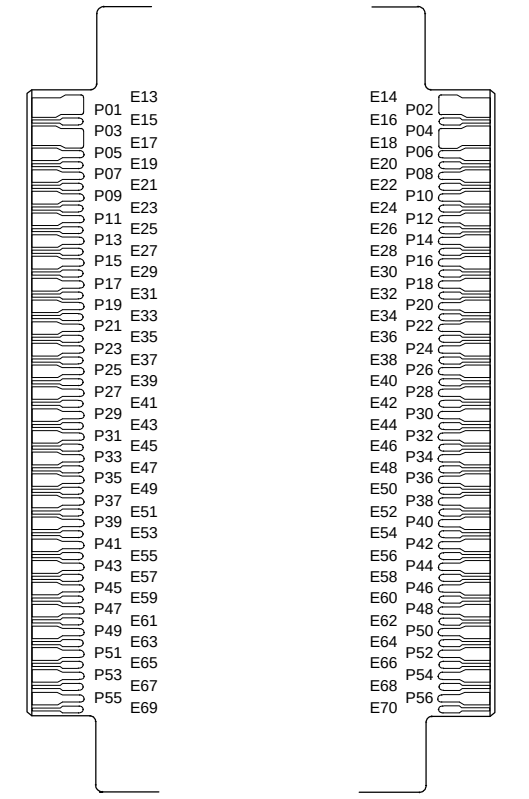
Connector Locations

P: The P connector is the interface between the ZT 8801 and the STD-80 bus. This connector is a 56-pin (dual 28-pin) card-edge connector with fingers on 0.125 inch centers. Mating connectors are a Viking 3VH28/1CND5 or equivalent for a three-level wire wrap, and a Viking 3VH28/1CNK5 or equivalent for the solder tail.

The "P/E Connector Pinout" illustration shows pin assignments for the P connector, and the "STD Bus Loading, P Connector" table shows signal assignments.

E: The E connector extends the P connector to interface the ZT 8801 to the STD 32 bus. This connector combines with the P connector to make a 114-pin (dual 57-pin) card-edge connector with fingers on 0.062 inch centers. Mating connectors are a Viking S3VT68/5DE12 or equivalent for the card extender, or a Viking S3VT68/5DP12 or equivalent for the solder tail.

The "P/E Connector Pinout" illustration shows pin assignments for the E connector, and the "Bus Signal Loading, E Connector" table shows signal assignments.



P/E Connector Pinout

J1: J1 is a latching 10-pin (dual 5-pin) male transition connector with 0.1 inch lead spacing. The interrupt inputs are available through this connector. The mating connector is a T&B Ansley #622-1000 or equivalent.

J1 Interrupt Connector Pinout

Pin	Description	Signal
2	Frontplane Interrupt	FP1*
4	Frontplane Interrupt	FP3*
6	Frontplane Interrupt	FP5*
8	Frontplane Interrupt	FP6*
10	Frontplane Interrupt	FP7*
Odd	Ground	

Note: An asterisk indicates an active low signal.

J2: J2 is a latching 10-pin (dual 5-pin) male transition connector with 0.1 inch lead spacing. The counter/timer and interrupt inputs are available through this connector. The mating connector is a T&B Ansley #622-1000 or equivalent.

J2 Counter/Timer And Interrupt Connector Pinout

Pin	Description
2	Timer/Counter Clock
4	Counter 2 CONTROL*
6	Counter 2 Output
8	No Connection
10	No Connection
Odd	Ground

Note: An asterisk indicates an active low signal.

J3: J3 is a latching 3-pin male low-profile header with 0.1 inch lead spacing. The serial channel transmit is available through this connector, driven by an RS-485 differential driver. The mating connector is a Molex 39-01-0033 or equivalent.

J3 Serial Connector Pinout

Pin	Description
1	Differential Driver (+)
2	Differential Driver (-)
3	Ground

J4: J4 is a latching 3-pin low profile header with 0.1 inch lead spacing. J4 is used for RS-485 multidrop communication and may be jumper selected as an RS-485/422 receiver. The mating connector is a Molex 39-01-0033 or equivalent.

J4 RS-485 Multidrop Connector Pinout

Pin	Description
1	Differential Transceiver (+)
2	Differential Transceiver (-)
3	Ground

J5, J6: J5 and J6 are latching 3-pin low profile headers with 0.1 inch lead spacing. J5 provides RS-232-C DTE communications (when used with the ZT 90069 cable), and J6 provides RS-232-C DCE communications (with ZT 90069).

J5 RS-232-C DTE Connector Pinout

Pin	Description
1	RS-232-C Transmit Data (TxD)
2	RS-232-C Receive Data (RxD)
3	Ground

J6 RS-232-C DCE Connector Pinout

Pin	Description
1	RS-232-C Receive Data (RxD)
2	RS-232-C Transmit Data (TxD)
3	Ground

J7: J7 is a 56-pin (dual 28-pin) vertical male header with 0.1 inch lead spacing. This connector provides 48 digital I/O lines, fused +5 V ±10%, and ground. J7 can interface directly to the 56-pin header on the ZT 2226 I/O module mounting rack via a standard 56-conductor ribbon cable such as the ZT 90089. J7's pin assignments also enable the ZT 2225 Industrial I/O Cable Adapter to connect J7 directly to one or two I/O module mounting racks with 8, 16, or 24 positions. For applications not using this cable, the mating connector is a T&B Ansley #622-5600 or equivalent.

J7 Parallel Port Pinout

Pin	Signal	Port Address [hex]	Pin	Signal	Port Address [hex]
1	IO00*/E0*	FA00 bit 0	28	IO24*	FA03 bit 0
2	IO01*/E1*	FA00 bit 1	29	IO25*	FA03 bit 1
3	IO02*/E2*	FA00 bit 2	30	IO26*	FA03 bit 2
4	IO03*/E3*	FA00 bit 3	31	IO27*	FA03 bit 3
5	IO04*/E4*	FA00 bit 4	32	IO28*	FA03 bit 4
6	IO05*/E5*	FA00 bit 5	33	IO29*	FA03 bit 5
7	IO06*/E6*	FA00 bit 6	34	IO30*	FA03 bit 6
8	IO07*/E7*	FA00 bit 7	35	IO31*	FA03 bit 7
9	GND	--	36	GND	--
10	IO08*	FA01 bit 0	37	IO32*	FA04 bit 0
11	IO09*	FA01 bit 1	38	IO33*	FA04 bit 1
12	IO10*	FA01 bit 2	39	IO34*	FA04 bit 2
13	IO11*	FA01 bit 3	40	IO35*	FA04 bit 3
14	IO12*	FA01 bit 4	41	IO36*	FA04 bit 4
15	IO13*	FA01 bit 5	42	IO37*	FA04 bit 5
16	IO14*	FA01 bit 6	43	IO38*	FA04 bit 6
17	IO15*	FA01 bit 7	44	IO39*	FA04 bit 7
18	GND	--	45	GND	--
19	IO16*	FA02 bit 0	46	IO40*	FA05 bit 0 (DC/DC)
20	IO17*	FA02 bit 1	47	IO41*	FA05 bit 1 (OSC)
21	IO18*	FA02 bit 2	48	IO42*	FA05 bit 2 (WDSTB)
22	IO19*	FA02 bit 3	49	IO43*	FA05 bit 3 (485)
23	IO20*	FA02 bit 4	50	IO44*	FA05 bit 4 (RTC)
24	IO21*	FA02 bit 5	51	IO45*	FA05 bit 5 (MD0)
25	IO22*	FA02 bit 6	52	IO46*	FA05 bit 6 (MD1)
26	IO23*	FA02 bit 7	53	IO47*	FA05 bit 7 (LED)
27	GND	--	54	+5 V	-- | 1 A fused
			55	+5 V	-- | 1 A fused
			56	+5 V	-- | 1 A fused

Note: An asterisk indicates an active low signal

J8: J8 is a latching 36-pin (dual 18-pin) female SBX connector with 0.1 inch lead spacing. This connector includes the power, address, data, and control signals needed to add custom I/O to the ZT 8801. The mating connector for an 8-bit SBX interface is a T&B Ansley #609 BX360.

J8 SBX Expansion Module Connector Pinout

Pin	Signal [1]	Description
1	+12V	+12 V
2	-12V	-12 V
3	GND	Signal Ground
4	+5V	+5 V
5	RESET	Reset
6	MCLK	10 MHz Clock
7	MA2	Address 2
8	MPST*	Module present [3]
9	MA1	Address 1
10	RSVD	Reserved - Address 5 [4]
11	MA0	Address 0
12	MINTR1	Interrupt 1 [5]
13	IOWRT*	I/O Write
14	MINTR0	Interrupt 0 [5]
15	IORD*	I/O Read
16	MWAIT*	Wait Request
17	GND	Ground
18	+5V	+5 V
19	MD7	Data Bit 7
20	MCS1*	Chip Select 1 [2]
21	MD6	Data Bit 6
22	MCS0*	Chip Select 0 [2]
23	MD5	Data Bit 5
24	RSVD	Reserved - Address 6 [4]
25	MD4	Data Bit 4
26	TDMA	Terminate DMA [3]
27	MD3	Data Bit 3
28	OPT1	Option 1 - Address 4 [4]

29	MD2	Data Bit 2
30	OPT0	Option 0- Address 3 [4]
31	MD1	Data Bit 1
32	MDACK*	DMA Acknowledge [3]
33	MD0	Data Bit 0
34	MDRQT	DMA Request [3]
35	GND	Ground
36	+5V	+5 V

Notes:

[1] Signals ending with an asterisk (*) are active low, and signals without an asterisk are active high.

[2] The 8-bit I/O address range for chip select 0 is FB00h through FB7Fh; for chip select 1, FB80h through FBFFh.

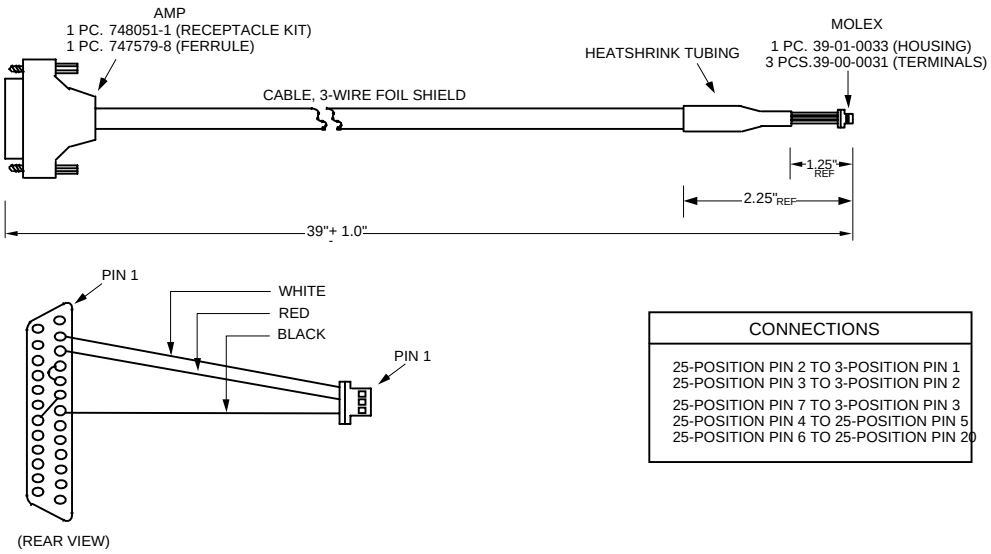
[3] These signals are not supported. The MPST* is not connected and TDMA is connected to ground. MDACK* is passively pulled up.

[4] These signals provide additional address lines to the three supported in the expansion module specification. This feature is supported with CT1 through CT4, as described in the "Cuttable Trace Descriptions" section of Appendix A.

[5] Interrupt 0 and interrupt 1 are routed to the interrupt jumper block.

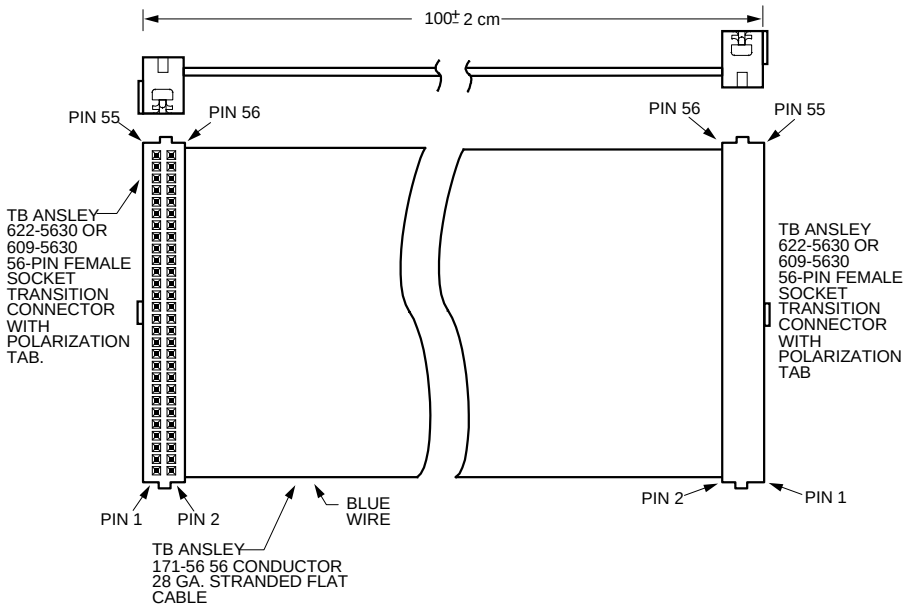
Cables

The following cables are available from Ziatech Corporation. Cable drawings are included here for those who wish to make their own cables:



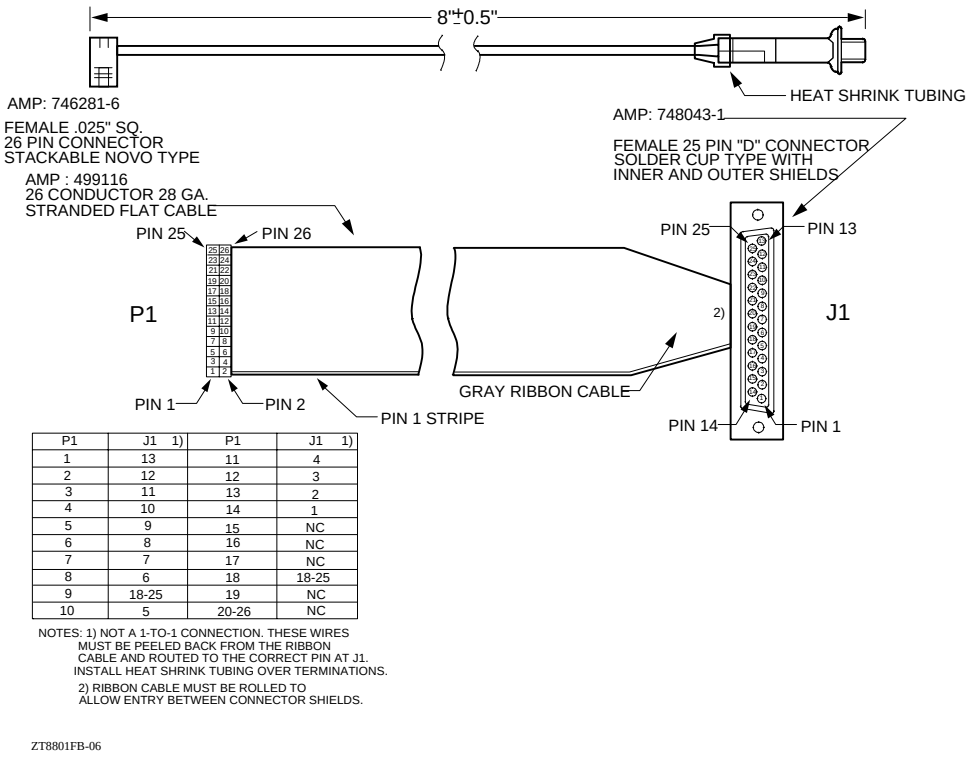
ZT8801FB-04

ZT 90069 Revision B Serial Cable



ZT8801FB-05

ZT 90089 Revision A Digital I/O Cable



P1	J1 1)	P1	J1 1)
1	13	11	4
2	12	12	3
3	11	13	2
4	10	14	1
5	9	15	NC
6	8	16	NC
7	7	17	NC
8	6	18	18-25
9	18-25	19	NC
10	5	20-26	NC

NOTES: 1) NOT A 1-TO-1 CONNECTION. THESE WIRES MUST BE PEELED BACK FROM THE RIBBON CABLE AND ROUTED TO THE CORRECT PIN AT J1. INSTALL HEAT SHRINK TUBING OVER TERMINATIONS.

2) RIBBON CABLE MUST BE ROLLED TO ALLOW ENTRY BETWEEN CONNECTOR SHIELDS.

ZT8801FB-06

ZT 90156 Revision 0 Printer Interface Cable

APPENDIX C. PIA SYSTEM SETUP CONSIDERATIONS

The 16C49 Parallel Interface Adapter (PIA) device used on the ZT 8801 is designed by Ziatech to offer bidirectional I/O signals with or without event sense capability. This device features low power, high speed, wide temperature operation achievable only by utilizing CMOS technology.

Although CMOS technology offers many advantages, you must observe a few cautions when interfacing to any CMOS parts.

CMOS inputs and outputs can exhibit latchup characteristics. These inherent characteristics of any CMOS technology can result in the formation of a Silicon-Controlled Rectifier (SCR) that appears between Vcc and ground when voltages greater than Vcc or less than ground are applied to inputs or outputs. When this happens, Vcc is effectively shorted to ground. The only way to remove the latchup condition is to shut off the power supply. If a large current is allowed to flow through the chip, its operating temperature may increase, it may exhibit intermittent operation, or it may be damaged.

CMOS inputs must be protected from slow rising signals and inductive coupling on their inputs. Failure to do so will allow a potentially large current to flow through the chip, damaging the chip.

The purpose of this appendix is to illustrate precautions you should take to prevent latchup conditions and protect inputs.

PREVENTING SYSTEM LATCHUP

The most common causes of latchup are:

- Input signals applied before the input circuitry is powered, resulting in a signal to power supply sequence mismatch
- Input signals greater than Vcc or less than ground, resulting in a signal level mismatch

Each of these conditions is covered in the following topics.

Power Supply Sequence Mismatch

A common application is to interface to a 24-position ZT 2226, Opto 22, or equivalent I/O module rack. Vcc and ground are provided from the ZT 8801 through connector J7, with Vcc protected by a 1 A fuse. This application is illustrated in Figure 1. In this application, no power supply sequence mismatch exists because the power supplying the input circuitry within the PIA is applied before or at the same time as the power supplying the external signals. Proper system operation will result.

However, if a power source other than that supplying the PIA is used to power the external signals, then a power sequence mismatch could occur, resulting in a latchup condition. An external power source might be required if the external circuitry requires

more than the 1 A supplied by the cable or if a custom interface is being designed (see Figure 2 for an example).

One solution is to switch the external signals' power supply with an output that is controlled by the computer. In this manner, if the computer is off, so is the external power supply. This solution is illustrated in Figure 3.

A simpler solution is to power the relay controlling the external power supply directly from Vcc and ground supplied by the interface cable.

Another solution is to utilize the same switch to control the computer's power supply and the external signals' power supply, as illustrated in Figure 4. This is an acceptable solution for power supply sequence mismatches as long as the computer supply ramps up faster than the external power supply. This ensures the PIA input circuitry is powered before the external signal circuitry.

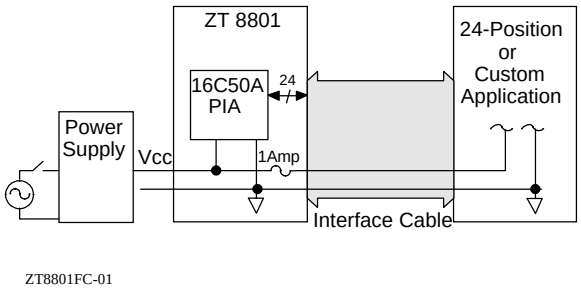


Figure 1. I/O Rack Vcc and Ground Supplied Via Interface Cable.
Correct Power Supply Sequence, Signal Level Matched

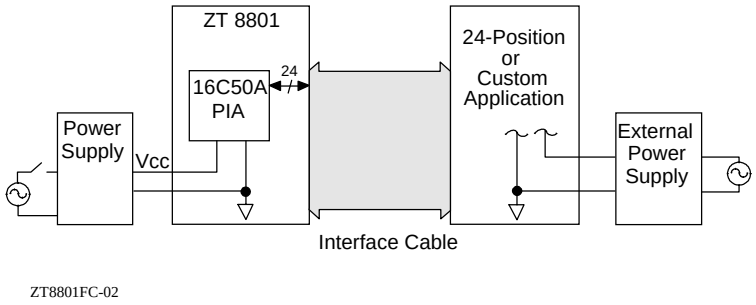


Figure 2. I/O Rack Vcc and Ground Supplied Externally.
Potential Power Supply Sequence Mismatch, Signal Level Mismatch

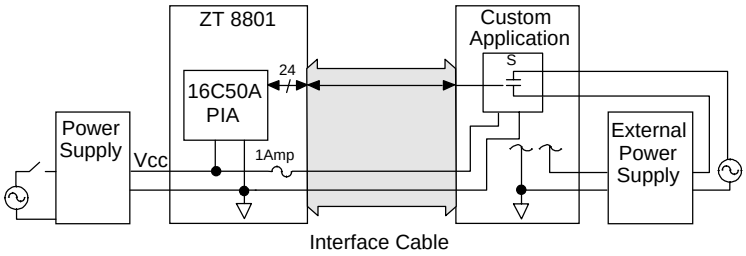


Figure 3. Computer-Switched External Power Supply.
Correct Power Supply Sequence, Potential Signal Level Mismatch

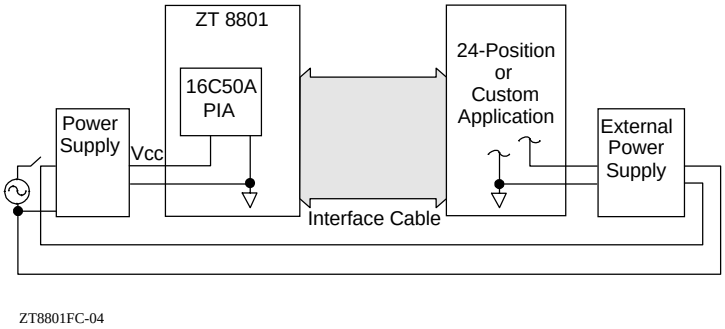


Figure 4. Computer And External Power Supply w/ Common Switch.
Correct Power Supply Sequence, Potential Signal Level Mismatch

Signal Level Mismatch

Power supplying the external signal in Figure 1 is always relative to the PIA input circuitry power because power is provided over the interface cable. Signal level mismatches will not occur and proper system operation will result. However, if separate power supplies are used, there are two predominant causes of signal level mismatches.

The first (assuming no sequencing problems) occurs when the two supplies are not referenced to each other, as illustrated in Figures 2, 3, and 4. This results in signals that may be higher than Vcc or lower than ground, potentially causing SCR latchup. All that is generally needed is to reference one supply to the other, typically by connecting a common ground. The most convenient way of connecting a common ground is to use the interface cable. Figures 5, 6, and 7 illustrate correct ground connections.

The second cause of mismatch occurs when the two power supplies are referenced to each other but the Vcc difference between the two power supplies exceeds .5 V. This results in signals that could be greater than Vcc, causing SCR latchup. This is easily remedied by adjusting the external power supply voltage to be within .5 V of the computer power supply voltage.

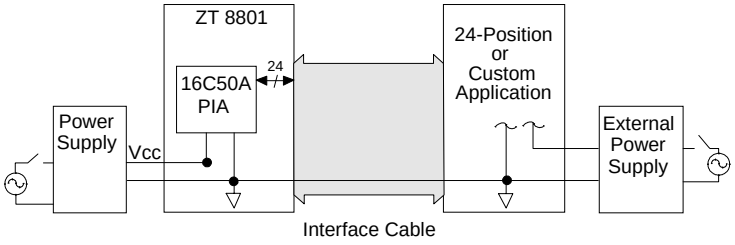


Figure 5. I/O Rack Vcc Supplied Externally, Common Ground. Potential Power Supply Seq. Mismatch, Correct Signal Level Match

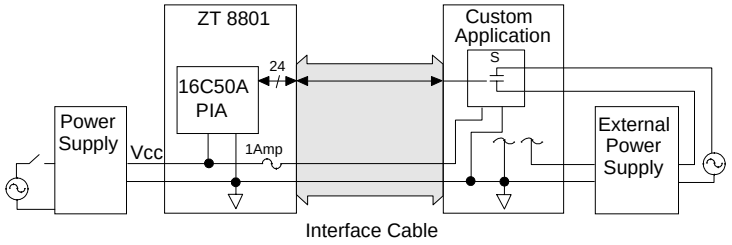


Figure 6. Computer-Switched External PS, Common Ground. Correct Power Supply Sequence, Correct Signal Level Match

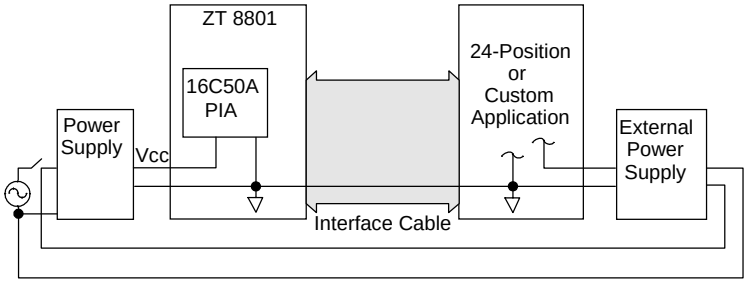


Figure 7. Computer & External PS W/ Common Switch & Ground. Correct Power Supply Sequence, Correct Signal Level Match

PROTECTING CMOS INPUTS

The most common causes of damaged inputs are:

- Slow rise times, resulting in a ground bounce within the chip
- Inductive coupling on I/O lines causing noise to be coupled into the chip, resulting in intermittent operation

Each of these conditions is covered in the following topics.

Rise Times

Slow rise times on a CMOS input can easily cause the transistor to bounce between Vil and Vih. When this oscillation occurs, the operating current goes up, resulting in "ground bounce." Ground bounce can cause internal latchup or can cause other system components to malfunction. A pullup termination resistor is used to increase the rise time.

Input rise times must be kept to less than 50 ns. Given a maximum chip capacitance of 10 pF, a 5 k ohm resistor is the largest that could be used without additional cabling. As cabling is added, the capacitance goes up, resulting in the use of a smaller pullup resistor until the maximum sink current of the output is achieved.

If the 16C49 PIA device is driving the output, its maximum sink current at a Vol of .4 V is 12 mA. This gives a lower limit of 420 ohms for the pullup resistor, allowing a maximum cabling capacitance of 110 pF. Note that while the input feature of the PIA may not be used by your application (PIA used as an output only), the input circuitry remains in parallel; therefore, the output rise time is still a critical parameter that the input still sees. The output rise time must not exceed 50 ns.

Be wary of using low pass filters to remove electrical noise. The resulting capacitance is typically too large to meet the 50 ns rise time requirement.

Typically, optical isolators are used to help remove electrical noise while providing for different grounds. Separate grounds are achieved through the use of an additional power supply for the optocoupler rather than using the computer's power supply. If the computer's power supply powers the optocouplers, electrical isolation is defeated. An example of one such circuit is illustrated in Figure 8. The circuit can be altered to allow for design considerations.

Assuming a Vil of 1 V maximum for the 16C49 PIA, the HP Dual Optocoupler must have a Vol of less than or equal to 1 V over the operating temperature. Using a TTL-compatible optocoupler gives a Vol of .6 V maximum with rise and fall times (50 ns and 10 ns, respectively) that are easily compatible with the PIA, given a 1 k ohm pullup.

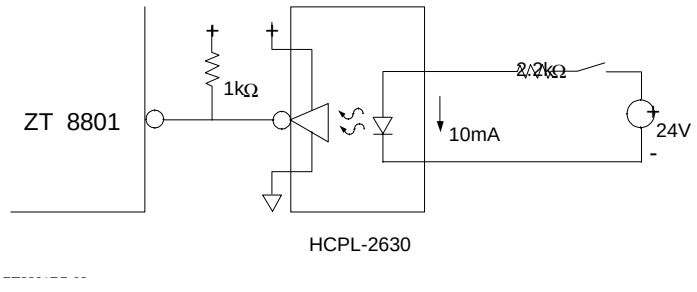


Figure 8. PIA-to-Optocoupler Interface Example

Inductive Coupling

Inductive coupling on I/O lines can cause noise to be coupled into the chip, resulting in intermittent operation. This situation occurs when the PIA I/O signals are routed with other signals within a wire bundle. One way to filter inductively coupled noise, or any noise for that matter, within a system with the same ground (not using optocouplers) is illustrated in Figure 9.

In the above circuit, the Texas Instruments 74S1053 Schottky diode clamps limit a transient to ±1 V above +5 V or below ground. The ferrite bead has a 50 ohm impedance at the frequency of interest. As the diodes begin to clamp and current flows through them, the voltage across the LCA05 5 V bidirectional TransZorbs® increases, causing them to conduct and allowing the majority of energy to flow through them instead of through the diode clamps. The 39 pF capacitor, in conjunction with the ferrite bead, forms an additional low pass filter, and is entirely optional. The 1 k ohm pullup ensures adequate rise time on the signal. The fuse acts as additional insurance against catastrophic events that might destroy the TransZorb and diode clamps.

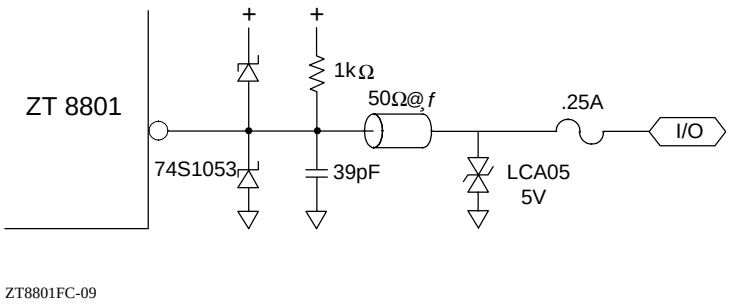


Figure 9. PIA-to-Filter Interface Example

ADDITIONAL INFORMATION

You can find additional design information in the *Advanced CMOS Logic Designer's Handbook,* published by Texas Instruments.

APPENDIX D. CUSTOMER SUPPORT

This appendix offers technical and sales assistance information for this product, and also the necessary information should you need to return a Ziatech product.

TECHNICAL/SALES ASSISTANCE

If you have a technical question, please call Ziatech's Customer Support Service at the number below, or e-mail our technical support team at tech_support@ziatech.com. Ziatech also maintains an FTP site located at ftp://ftp.ziatech.com.

If you have a sales question, please contact your local Ziatech Sales Representative or the Regional Sales Office for your area. Address, telephone and FAX numbers, and additional information is available at Ziatech's website, located at:

http://www.ziatech.com.

Corporate Headquarters

1050 Southwood Drive

San Luis Obispo, CA 93401 USA

Tel (805) 541-0488

FAX (805) 541-5088

RELIABILITY

Ziatech has taken extra care in the design of the ZT 8801 in order to ensure reliability. The product was designed in top-down fashion, using the latest in hardware and software design techniques, so that unwanted side effects and unclean interactions between parts of the system are eliminated. Each ZT 8801 has an identification number. Ziatech maintains a lifetime data base on each board and the components used. Any negative trends in reliability are spotted and Ziatech's suppliers are informed and/or changed.

RETURNING FOR SERVICE

Before returning any of Ziatech's products, you must phone Ziatech at (805) 541-0488 and obtain a Returned Material Authorization (RMA) number. The following information is needed to expedite the shipment of a replacement to you:

1. Your company name and address for invoice
2. Shipping address and phone number
3. Product I.D. number
4. If possible, the name of a technically qualified individual at your company familiar with the mode of failure on the board

If the unit is out of warranty, service is available at a predesignated service charge. Contact Ziatech for pricing and please supply a purchase order number for invoicing the repair.

Pack the board in **anti-static** material and ship in a sturdy cardboard box with enough packing material to adequately cushion it. *Any product returned to Ziatech improperly packed will immediately void the warranty for that particular product!* Mark the RMA number clearly on the outside of the box before returning.

ZIATECH WARRANTY

Ziatech provides a five-year limited warranty to its customers. Ziatech also has an explicit policy regarding the use of Ziatech products in life support systems. These topics are covered in the following sections.

Five-Year Limited Warranty

Products manufactured by Ziatech Corporation are covered from the date of purchase by a five-year warranty against defects in materials, workmanship, and published specifications applicable to the date of manufacture. During the warranty period, Ziatech will repair or replace, solely at its option, defective units provided they are returned at customer expense to an authorized Ziatech repair facility. Products which have been subjected to misuse, abuse, neglect, alteration, or unauthorized repair, determined at the sole discretion of Ziatech, whether by accident or otherwise, are excluded from warranty. The warranty on fans and disk drives is limited to two years and the warranty on flat panel displays is limited to nine months from date of purchase. Other products and accessories not manufactured by Ziatech are limited to the warranty provided by the original manufacturer. Consumable items (fuses, batteries, etc.) and software are not covered by this warranty.

Ziatech Corporation warrants that for a period of ninety (90) days from the date of purchase; the media on which software is furnished will be free of defects in materials and workmanship under normal use; and the software contains the features described in the Ziatech price list. Otherwise, the software is provided “AS IS”. This limited warranty extends only to Customer as the original licensee. Customer's exclusive remedy and Ziatech's entire liability under this limited warranty will be, at Ziatech's option, to repair or replace the software, or refund the license fee paid therefore.

Ziatech may offer, where applicable and available, replacement products; otherwise, repairs requiring components, assemblies, and other purchased materials may be limited by market availability.

Ziatech assumes no liability resulting from changes to government regulations affecting use of materials, equipment, safety, and methods of repair. Ziatech may, at its discretion, offer replacement products.

THE ABOVE WARRANTY IS IN LIEU OF ANY OTHER WARRANTY, WHETHER EXPRESSED, IMPLIED, OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, ANY

WARRANTY FOR FITNESS OF PURPOSE, MERCHANTABILITY, OR FREEDOM FROM INFRINGEMENT OR THE LIKE, AND ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATIONS, OR SAMPLE.

Ziatech neither assumes nor authorizes any person to assume for it any other liability. The liability of Ziatech under this warranty agreement is limited to a refund of the purchase price. In no event shall Ziatech be liable for loss of profits, use, incidental, consequential, or other damage, under this agreement.

Life Support Policy

Ziatech products are not authorized for use as critical components in life support devices or systems without the express written approval of the president of Ziatech Corporation. As used herein:

1. Life support devices or systems are devices or systems which support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.

2. A critical component is any component of a life support device or system whose failure to perform can be expected to cause the failure of the life support device or system, affect its safety, or limit its effectiveness.

TRADEMARKS

Turbo Debugger is a registered trademark of Borland International, Inc.

MULTIMODULE is a trademark of Intel Corporation.

AT, IBM, IBM PC, and PC XT are registered trademarks of International Business Machines, Inc.

MS-DOS is a registered trademark of Microsoft Corporation.

Generation 4 is a registered trademark of Opto 22.

DEBUG/RT is a trademark of Paradigm Systems.

STD ROM is a trademark of Ziatech Corporation.

STD 32 is a registered trademark of Ziatech Corporation.

1050 Southwood Drive
San Luis Obispo, CA 93401 USA
Tel: (805) 541-0488
FAX: (805) 541-5088
E-Mail: tech_support@ziatech.com
Internet: http://www.ziatech.com