THE WANG PROFESSIONAL COMPUTER

Technical Reference Manual

WANG

WANG LABORATORIES, INC.
ONE INDUSTRIAL AVENUE, LOWELL, MA 01851 • TEL: 617/459-5000, TWX 710-343-6769, TELEX 94-7421

To: Purchasers of Wang Professional Computer Open Architecture

From: Wang Laboratories, Inc.

Date: July 29, 1985

Subject: Support Policy for Wang PC Open Architecture (195-4666-9)

The Wang PC Open Architecture package is intended to provide purchasers with sufficient information for creating option boards and applications software for the Wang PC. The Open Architecture package consists of The Wang Professional Computer Technical Reference Manual, 2nd Edition (700-8090A), The Wang Professional Computer Program Development Guide, 2nd Edition, Addendum (700-8018A.01), and the Wang PC BIOS Listing (## to be supplied).

The second edition of the Technical Reference Manual includes new chapters on the system bus, Start and On-board PROMs, and Text/Image/Graphics and Multiport Communication controllers. It also contains technical corrections to the first edition. The addendum to the Program Development Guide contains technical corrections to the second edition.

The Open Architecture package is meant to be the user's source of information. Wang Laboratories, Inc., does not offer a general consulting service. However, users can call the Wang Customer Action Line (CAL) if there appear to be inconsistencies between information in the Open Architecture package and system performance. The CAL toll-free number is 1-800-323-WANG. International customers who do not use 800 numbers can call 1-617-967-3057. Support personnel are available to answer questions from 8:30 a.m. to 5:30 p.m., Eastern time, Monday through Friday.

If you call the Customer Action Line, please make sure you know the version numbers of the system and applications software you are using, as well as the model number of your PC and what option cards it contains. Before you call the Customer Action Line, try the suggestions in The Wang Professional Computer Troubleshooting Guide (715-0140).

WANG LABORATORIES, INC., ONE INDUSTRIAL AVENUE, LOWELL, MA 01851 • TELEPHONE (617) 459-5000, TWX 710-343-6769, TELEX 94-7421

It is Wang Laboratories' position not to service nonWang option boards. The following policies apply to the service of Wang PCs on which nonWang option boards are installed:

- It is the policy of Wang Laboratories to devote its resources exclusively to the support of Wang equipment and accessories. Wang personnel are not trained or authorized to perfrom any installation, repair, or maintenance of nonWang devices.

- When a Wang system or product also contains devices, modifications, alterations, attachments, or features (including microcode) that are not Wang-approved, Wang Customer Services will perform service on the Wang portions only. If Wang personnel, in performing required diagnostic and repair support of Wang equipment, determine that interference of this support is caused by nonWang devices, the customer, at Wang's discretion, will be required to disconnect or remove all such nonWang devices before Maintenance Agreement, Warranty, or Time and Material service is performed.

- If Wang personnel determine that the malfunction reported by the customer is occurring in, or is caused solely by, such nonWang devices, Wang will charge the customer for the entire service call at Wang's standard rates for travel, labor, and parts. These charges will be applied whether or not the customer's equipment is covered under Warranty or a Wang Maintenance Agreement.

Sincerely yours,

Wang Laboratories, Inc.

The Wang Professional Computer Technical Reference Manual

2nd Edition — August 1985

700-8090A

WANG

WANG LABORATORIES, INC., ONE INDUSTRIAL AVENUE, LOWELL, MA 01851 • TEL. (617) 459-5000, TWX 710-343-6769, Telex 94-7421

Disclaimer of Warranties and Limitation of Liabilities

The staff of Wang Laboratories, Inc., has taken due care in preparing this manual; however, nothing contained herein modifies or alters in any way the standard terms and conditions of the Wang purchase, lease, or license agreement by which this software package was acquired, nor increases in any way Wang's liability to the customer. In no event shall Wang Laboratories, Inc., or its subsidiaries be liable for incidental or consequential damages in connection with or arising from the use of the software package, the accompanying manual, or any related materials.

NOTICE:

All Wang Program Products are licensed to customers in accordance with the terms and conditions of the Wang Laboratories, Inc. Standard Program Products License; no ownership of Wang Software is transferred and any use beyond the terms of the aforesaid License, without the written authorization of Wang Laboratories., is prohibited.

PREFACE

The Wang Professional Computer Technical Reference Manual provides technical information about the operation and architecture of the Wang Professional Computer (PC). Using this manual requires a thorough knowledge of computer architecture, including input/output (I/O) interface and controllers. Wang PC users who wish to design their own option boards should use this manual along with The Wang Professional Computer Program Development Guide, 2nd Edition, Addendum (700-8018A.01).

The Technical Reference Manual begins with an overview of the Wang PC system architecture. Subsequent chapters discuss the interrupt system, the keyboard, parallel I/O interface, extended programmable communication interface (EPCI), the various controllers on the system, the system board I/O address assignments, and Start and On-board PROM specifications. The final chapter is a description of the system bus interface. Two appendixes provide explanations of power-on diagnostic error messages and mechanical diagrams for option boards and RF shields.

The Technical Reference Manual is meant to be the user's source of information. Wang Laboratories, Inc., does not offer a general consulting service. However, users can call the Wang Customer Action Line (CAL) if there appear to be inconsistencies between information in the Technical Reference Manual and system performance. The CAL toll-free number is 1-800-323-WANG. International customers who do not use 800 numbers can call 1-617-967-3057. Support personnel are available to answer questions from 8:30 a.m. to 5:30 p.m., Eastern time, Monday through Friday.

If you call the Customer Action Line, please make sure you know the version numbers of the system and applications software you are using, as well as the model number of your PC and what option cards it contains. Before you call the Customer Action Line, try the suggestions in The Wang Professional Computer Troubleshooting Guide (715-0140).

NOTE:
Wang Laboratories cannot support modifications made to the Wang PC operating system.

CONTENTS

CONTENTS (continued)

CHAPTER 6 FLOPPY DISK CONTROLLER

CHAPTER 7 LOW-RESOLUTION VIDEO CONTROLLER

CHAPTER 8 MEDIUM-RESOLUTION VIDEO CONTROLLER

CHAPTER 9 REMOTE TELECOMMUNICATION CONTROLLER

CONTENTS (continued)

CONTENTS (continued)

CONTENTS (continued)

FIGURES

TABLES

TABLES (continued)

CHAPTER 1
SYSTEM ARCHITECTURE

The main processor is an 8086 running at up to 8 MHz; an optional 8-MHz 8087 coprocessor can be installed for high-performance numeric data processing. The 8086 and its 8087 coprocessor communicate with one another across a local interprocessor bus. Both processors communicate with memory and I/O components across the system bus by means of address latches, data transceivers, and an 8288 Bus Controller Chip.

Figure 1-1 shows a processor block diagram. The main processor and optional auxiliary processor interface to system buses by means of local address, data, and control buses as shown.

Figure 1-2 shows a system board block diagram. The buffered address and data buses connect the system board components and provide a system bus interface. Figures 1-1 and 1-2 together form a complete diagram showing all processor circuitry on the system board.

NOTE:
For complete information on system bus operation, refer to Chapter 15, System Bus Interface.

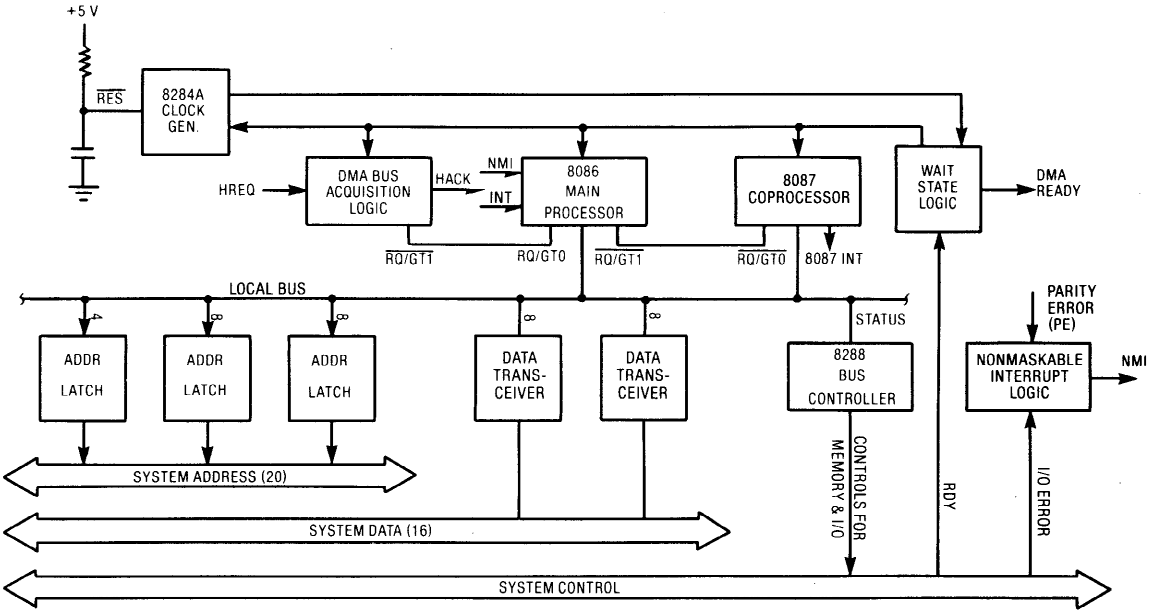


Figure 1-1. Processor Block Diagram

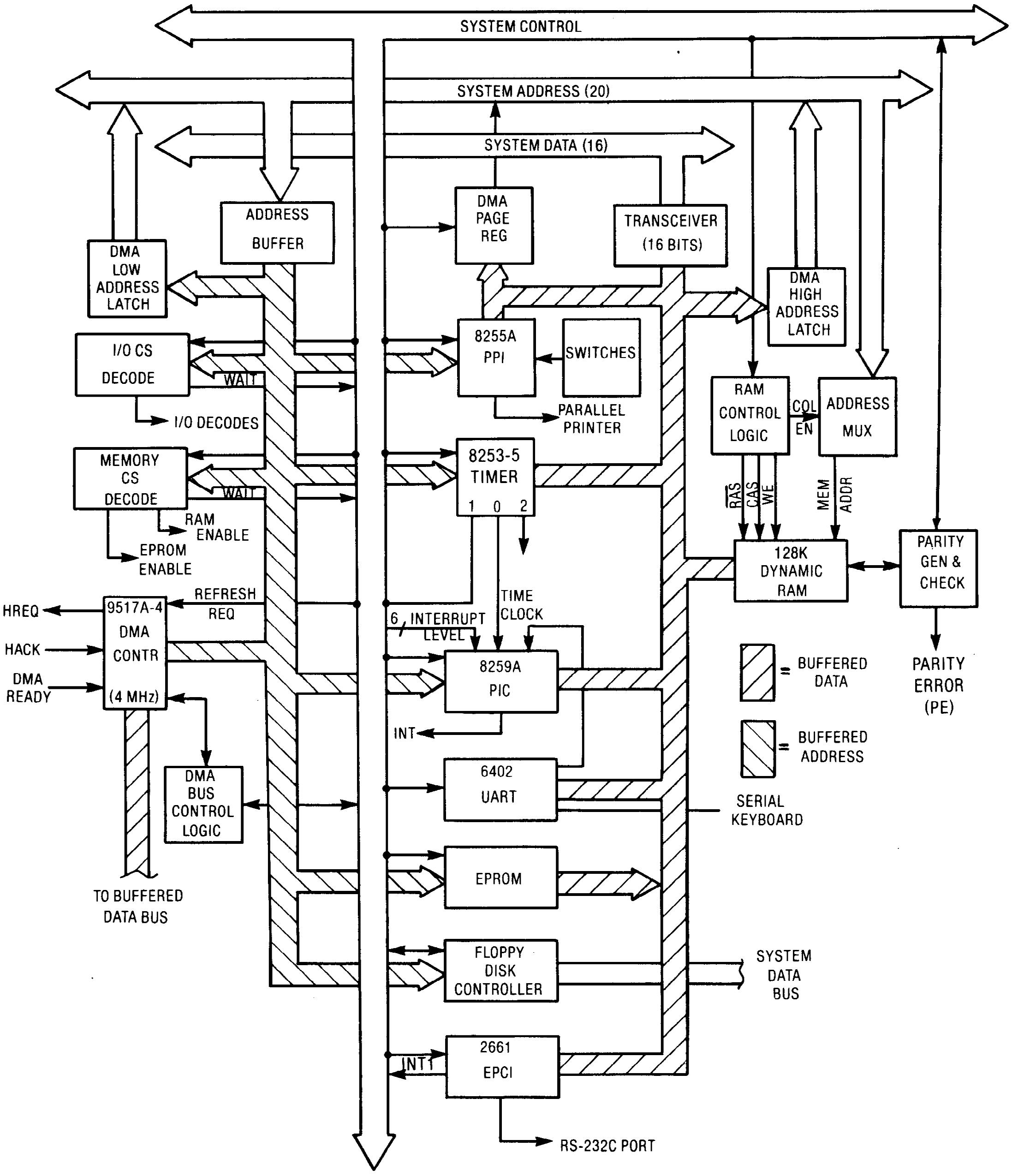


Figure 1-2. System Board Block Diagram

1.1 EXTENDED MEMORY OPTION

An optional extended memory board augments the 128 KB (kilobytes) of standard system memory with 128 KB, 256 KB, 512 KB, or 1 MB (megabytes) of extended dynamic RAM with parity. Extended memory address space is organized into four 128 KB banks. Each bank of extended memory must be initialized for access by means of the Memory Segment Boundary Register (refer to Table 1-1). This register assigns a base address to each of the four extended memory banks. The following are usable base addresses:

20000H	80000H
40000H	A0000H
60000H	C0000H

Two other base addresses, 0 and E0000H, are also possible but must not be used. Each bank of extended memory should have a different base address. To prevent segments from being overmapped, a read of Port 1026 must be done to determine memory size.

As with system board memory, parity is undefined before an extended memory location has been written with data; therefore, a parity error may occur if a program reads an extended memory location that was not first written with data. Extended memory parity errors generate an 8086 nonmaskable interrupt (NMI) request. A program can distinguish extended memory parity errors from system memory parity errors by accessing an I/O port on the extended memory board.

Table 1-1. Extended Memory Board I/O Ports

Port	Description	
1xC0H	Read or Write Extended Memory Segment Boundary Register. Four 4-bit fields determine the base address (3 bits) and on/off status (1 bit) of each 128 KB bank of extended memory. Bit assignments are as follows:	
	2, 1, 0	High-order bits (A19, A18, A17) of base address for first 128 KB bank of extended memory.
	3	Set to 1 when first bank of extended memory is active. If cleared to 0, disables first bank.
	6, 5, 4	High-order bits (A19, A18, A17) of base address for second 128 KB bank of extended memory.
	7	Set to 1 when second bank of extended memory is active. If cleared to 0, disables second bank.
	10, 9, 8	High-order bits (A19, A18, A17) of base address for third 128 KB bank of extended memory.
	11	Set to 1 when third bank of extended memory is active. If cleared to 0, disables third bank.
	14, 13, 12	High-order bits (A19, A18, A17) of base address for fourth 128 KB bank of extended memory.
	15	Set to 1 when fourth bank of extended memory is active. If cleared to 0, disables fourth bank.
1xCEH	Write arbitrary data to clear the parity error flag after a parity error.	
1xFCH	Write arbitrary data to reset the extended memory board. Clears the Extended Memory Segment Boundary Register to 0, thereby disabling all extended memory. Also clears the parity error flipflop and establishes odd parity.	
1xFEH	Read Option ID Code (D0-7 will be 3FH) and interrupt (parity) status. (D8 will be set to 1 if a parity error has occurred since the parity error flipflop was last cleared, otherwise cleared to 0.) Read or write odd/even parity flag. Odd parity is established by setting D9 to 1. This is the default option. Clearing D9 to 0 selects even parity.	

1.2 DIRECT MEMORY ACCESS

The 4-channel DMA controller allocates Channel 0 for dynamic RAM refresh, leaving three channels available for general use. It transfers only byte data, not word data. Low-order bytes from even memory addresses are transferred on the low-order data bus lines (D0-7), and high-order bytes from odd memory addresses go out on the high-order data bus lines (D8-15). Therefore, 8-bit I/O device options that use DMA must include a byte-swapping mechanism to access high-order lines on the data bus.

DMA transfers generally occur in single-byte mode (not block transfer mode), and always use normal (not compressed) timing. Memory-to-memory transfers cannot use a DMA channel and, instead, must be performed by the 8086. However, one I/O option can perform DMA transfers directly to another I/O option, bypassing the system board entirely, provided either the data's source or destination (but not both) is memory mapped. Maximum DMA transfer rate is 300K bytes/second. Maximum processor latency is the duration of the longest locked 8086 instruction: 180 cycles (22.5 microseconds) for a locked DIV instruction. The DMA controller should be programmed for rotating priority to give every channel equal access to the system bus.

DMA Page Registers are 4-bit registers allocated to DMA Channels 1, 2, and 3. Mapped to Output Ports 10C2H, 10C4H, and 10C6H, respectively, they allow direct memory access throughout the 1 MB address space by supplying the four high-order bits of a 20-bit DMA address (A16-19). Before programming the DMA controller to perform a DMA transfer, the 8086 writes the A16-19 page address bits into the DMA Page Register for the DMA channel being used. DMA Page Register content never changes during a DMA transfer and, in particular, the DMA Page Register is not incremented at the end of a page. After accessing the last word on a page, DMA logic then accesses the first word on the same page. To achieve DMA transfers across a page boundary, the DMA controller must be programmed to perform two separate operations by using two different DMA Page Register values.

1.3 PROGRAMMING THE DMA CONTROLLER

Software in the 8086 writes DMA registers (described in Table 1-2) to program each channel of the DMA controller. This establishes the direction of a DMA transfer, the first memory mapped address involved in the transfer, and the number of bytes to be transferred at successively higher or lower memory addresses. The 8086 also programs each I/O device option and assigns no more than one active device to each DMA channel. By convention, the number of the DMA channel assigned to an I/O option board is always four less than its interrupt priority level. For example, all I/O option devices that operate at the interrupt priority level 6 should use DMA Channel 2, and only one such device should be active at a time.

A DMA operation that moves a block of data from an I/O device to a contiguous region of memory is called a DMA process. Once programmed, the DMA controller can complete a DMA process without software intervention. The controller executes a DMA process by performing one or more DMA transfers, each initiated by the I/O device. Although a DMA transfer can involve any number of bytes, single-byte transfers are generally used because they have the smallest impact on the dynamic memory refresh mechanism.

Table 1-2. DMA Controller Ports

Address	Description
10A0H	Current Address Register for DMA Channel 0. Each DMA channel has a 16-bit Current Address Register (accessed as two 8-bit bytes), which holds Bits A0-15 of the address of the next memory location involved in a DMA transfer on that channel. The DMA controller automatically updates the Current Address Register after each DMA transfer. The 8086 accesses this register by first writing to 10B8H, which resets the first/last byte flipflop, if necessary. It then writes the low-order current address byte (A0-7) to 10A0H and concludes by writing the high-order current address byte (A8-15) to 10A0H. Loading a Current Address Register automatically loads the corresponding Base Address Buffer. If autoinitialization is requested, the Current Address Register will be reloaded from the Base Address Buffer at the start of each subsequent DMA process.
10A2H	Word Count Register for DMA Channel 0. Each DMA channel has a 16-bit Word Count Register (accessed as two 8-bit bytes), which holds one less than the number of bytes that remain to be transferred in a DMA process. The DMA controller automatically decrements the Word Count Register after each DMA transfer. The 8086 accesses this register by first writing to 10B8H, which resets the first/last byte flipflop, if necessary. It then writes the low-order word count byte to 10A2H and concludes by writing the high-order word count byte to 10A2H. Loading a Word Count Register automatically loads the corresponding Base Word Count Buffer. If autoinitialization is requested, the Word Count Register will be reloaded from the Base Word Count Buffer at the start of each subsequent DMA process.
10A4H	Current Address Register for DMA Channel 1.
10A6H	Word Count Register for DMA Channel 1.
10A8H	Current Address Register for DMA Channel 2.
10AAH	Word Count Register for DMA Channel 2.
10ACH	Current Address Register for DMA Channel 3.
10AEH	Word Count Register for DMA Channel 3.

Table 1-2. DMA Controller Ports (continued)

<table>
<tr><th>Address</th><th>Description</th></tr>
<tr><td>10B0H</td><td>Write DMA Command Register. This 8-bit register is cleared by /RESET and must be initialized by the 8086 to enable the DMA controller and to establish the prioritization mechanism. Bit assignments are as follows:
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0, 1</td><td>Must be cleared to 0.</td></tr>
<tr><td>2</td><td>DMA controller disabled if set to 1. This bit can be set at any time to suspend all DMA activity without destroying parameters contained in the DMA controller's internal registers.</td></tr>
<tr><td>3</td><td>Must be cleared to 0.</td></tr>
<tr><td>4</td><td>Establishes rotating priority when set to 1. Under rotating priority, each channel is assigned the lowest priority after a DMA transfer has been performed on it, making the next lower channel the highest priority channel. Under fixed priority, selected by clearing this bit to 0, DMA Channel 0 always has highest priority and Channel 3 the lowest.</td></tr>
<tr><td>5</td><td>Must be cleared to 0.</td></tr>
<tr><td>6</td><td>Must be set to 1.</td></tr>
<tr><td>7</td><td>Must be cleared to 0.</td></tr>
</table>
Read DMA Status Register. Indicates which DMA channels are currently performing a DMA process and which channels have pending DMA requests. Bit assignments are as follows:
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0-3</td><td>Set to 1 when the corresponding DMA channel completes a DMA process. When cleared to 0, the channel is performing a DMA process.</td></tr>
<tr><td>4-7</td><td>Set to 1 when the corresponding DMA channel has a pending DMA request. For example, Bit 4 is set when /DREQ0 is active and Bit 7 is set when /DREQ3 is active.</td></tr>
</table>
</td></tr>
</table>

Table 1-2. DMA Controller Ports (continued)

Address	Description
10B2H	Bit Description Write DMA Request Register. Used by diagnostic programs to initiate DMA transfers under software control. Bit assignments are as follows: 1, 0 Select DMA Channel 0 (00), 1 (01), 2 (10), or 3 (11), depending on the binary value. 2 If set to 1 (and the DMA controller is operating in block transfer mode), simulate an active DMA request on the channel designated by Bits 1 and 0. If cleared to 0, cancel the simulated DMA request.
10B4H	Write Individual DMA Mask Register Bit. Bit assignments are as follows: Bit Description 1, 0 Select DMA Channel 0 (00), 1 (01), 2 (10), or 3 (11), depending on the binary value. 2 If set to 1, mask off and disable the DMA channel designated by Bits 1 and 0. If cleared to 0, enable the channel. /RESET automatically disables all four channels.
10B6H	Write DMA Mode Register. Establishes the operating mode and type of DMA transfer performed on a DMA channel. Bit assignments are as follows: Bit Description 1, 0 Select DMA Channel 0 (00), 1 (01), 2 (10), or 3 (11), depending on the binary value. Parameters entered in Bits 2-7 apply only to the designated channel. 3, 2 Bit 2 is set to 1 for I/O write transfers (data originates in memory) and cleared to 0 during read operations. Bit 3 is set to 1 for I/O read transfers (memory is the data destination) and cleared to 0 during writes. At least one of these bits must be cleared to 0. If neither is set to 1, the channel performs normally but does not activate the bus signals that control memory and I/O access.

Table 1-2. DMA Controller Ports (continued)

Address	Description	
10B6H (cont.)	Bit	Description
	4	Set to 1 only if the channel should automatically reinitialize at the end of each DMA process. Autoinitialization repeats a DMA process by starting it over when terminal count is reached or an external /EOP is received.
	5	Set to 1 if the channel should decrement its Current Address Register and transfer data into successively lower memory addresses. If cleared to 0, the channel will increment its Current Address Register.
	7, 6	Bit 6 is set to 1 for single-byte transfers and cleared to 0 during block transfer operations. Bit 7 is set to one-block transfers and cleared to 0 during single-byte operation. At least one of these bits must be cleared to 0. (If neither is set to 1, the channel performs a block transfer only while its DMA request line remains active; it suspends operation when its request line goes inactive. This "Demand mode" allows the I/O device to determine the transfer size and duration. A demand transfer resumes where it was suspended when the channel's DMA request line again goes active.)
10B8H	High- and Low-Order Byte Select. Write arbitrary data to reset DMA register addressing logic so that the next attempt to read or write any DMA Current Address or Word Count Register will access its low-order byte. Subsequent operations that address these 16-bit DMA registers will alternate between high- and low-order bytes.	
10BAH	Write arbitrary data to reset the DMA controller. A software reset performed in this manner has the same effect as the system bus /RESET signal: it clears the Command, Status, and Request Registers, selects the low-order byte as the byte of a 16-bit register that will be accessed next, and sets all four bits of the Mask Register.	
10BEH	Write all Mask Register bits. Bits 0-3 are set to 1 to mask off and disable the corresponding DMA channel.	

Table 1-2. DMA Controller Ports (continued)

Address	Description
10C2H	Write DMA Page Register for Channel 1. The 4-bit DMA page register receives the four high-order bits of a 20-bit DMA current address with A16 in the low-order position (D0) and A19 in the high-order position (D3).
10C4H	Write DMA Page Register for Channel 2.
10C6H	Write DMA Page Register for Channel 3.

1.4 SYSTEM STATUS PORT AND INTERRUPT STATUS PORT

The System Status Port, Input Port 10E0H, supplies six status flags that report system status conditions. Table 1-3 lists the System Status Port bit assignments and describes the status conditions they indicate. The Interrupt Status Port, Input Port 1022H, is another special port that supplies eight interrupt status flags. Table 1-4 lists bit assignments for the Interrupt Status Port and describes the interrupt status conditions.

Table 1-3. System Status Port Signals (Input Port 10E0H)

Bit	Description
0	Memory Parity Flag. Normally set to 1. If cleared to 0, a parity error in system board memory has been detected.
1	I/O Error Flag. Normally set to 1. If cleared to 0, an I/O option board has returned an error indicator by asserting the /I/O ERROR line on the system bus.
2	Unassigned.
3	Floppy Diskette Controller Interrupt Flag. Set to 1 when the floppy Diskette controller completes execution phase of an FDC operation. Cleared to 0 while the FDC is performing an operation. The other type of floppy disk interrupt, which originates at the drive rather than the controller, instead sets Bit 4 or 5 of this port.
4	When set to 1, indicates door disturbed on floppy diskette Drive 1. (This generates an interrupt request.) If cleared to 0, door on floppy diskette Drive 1 has not been opened since CPU last cleared the FDC "drive 1 door disturbed" interrupt request by writing to Output Port 1000H with Bit 2 set to one.
5	When set to 1, indicates door disturbed on floppy diskette Drive 2. If cleared to 0, door on floppy diskette Drive 2 has not been opened since CPU last cleared the FDC "drive 2 door disturbed" interrupt request by writing to Output Port 1000H with Bit 3 set to 1.
6	Door open on floppy diskette Drive 1 when set to 1. Door closed on diskette Drive 1 when cleared to 0.
7	Door open on floppy diskette Drive 2 when set to 1. Door closed on diskette Drive 2 when cleared to 0.

Table 1-4. Interrupt Status Port Signals (Input Port 1022H)

Bit	Signal Name and Description
0	/TIMER 2 INTERRUPT. When cleared to 0, indicates that 8253-5 Timer Channel 2 has reached terminal count.
1	/SERIAL INTERRUPT When cleared to 0, indicates a serial communication interrupt (RxRDY, TxRDY, or /TxEMT/DSCHG) from the 2661 EPCI.
2	/PARALLEL PORT INTERRUPT. When cleared to 0, indicates that the parallel I/O interface is ready to transmit or receive data (ie, logical OR of /DAV, /ACKNLG, and latched BUSY).
3	/DMA INTERRUPT. When cleared to 0, indicates that one of the DMA channels has reached terminal count.
4	KBD INTERRUPT TRANSMIT. When set to 1, indicates that the Keyboard Transmit Buffer Register is empty and available to accept a new byte of output data.
5	KBD INTERRUPT RECEIVE. When set to 1, indicates that there is an input character for the 8086 to read in the Keyboard Receive Buffer Register.
6	FLOPPY DISK INTERRUPT. Set to 1 when the floppy disk has a pending interrupt request, which can be either a floppy disk controller interrupt or a floppy disk drive interrupt. Bits 3-5 of System Status Port 10E0H distinguish between these interrupt sources.
7	8087 INTERRUPT. Set to 1 if the 8087 has a pending interrupt request.

1.5 8253-5 PROGRAMMABLE INTERVAL TIMER

The 8253-5 Programmable Interval Timer chip contains three identical and independent channels, each of which can operate as a realtime clock that generates a periodic interrupt request. Each channel consists of a Control Register, a 16-bit down counter, and two interface signals: a clock input, which decrements the counter, and an output for the channel to assert when its counter reaches 0. The clock input runs at 500 kHz for Channels 0 and 2. Channel 1 receives a 2-MHz clock. Although the 8253-5 timer chip can operate in six different modes, only two are supported: Mode 2, the "rate generator" mode, and Mode 4, the "software triggered strobe" mode.

Timer Channel 0 is used as a realtime clock and must be programmed to operate in Mode 2. Writing to I/O Port 1040H loads the Channel 0 counter with an initial value that designates the number of clock pulses between output pulses (refer to Table 1-5). Channel 0 counts clock pulses, beginning with the next pulse immediately after the counter is loaded. It decrements its counter on the falling edge of each clock pulse. When the counter reaches 0, it generates a Level 0 interrupt request, resets the counter, and immediately begins counting clock pulses for a new timing cycle. You do not need to reload the Channel 0 counter between cycles. If a new value is loaded into the counter, it has no effect until after Channel 0 generates its next interrupt request at the end of the current timing cycle. Since Timer Channel 0 is the only possible source of Level 0 interrupt requests, it has no interrupt status flag. The 8086 clears a Channel 0 timer interrupt by writing arbitrary data to I/O Port 10E0H.

Timer Channel 1 is reserved for timing the interval between dynamic memory refresh bursts. Like Channel 0, Timer Channel 1 operates only in Mode 2. Channel 1 generates DMA requests, not interrupt requests. Its counter must be loaded by writing a count of 60 (3CH) to I/O Port 1042H.

Channel 2 is available for use as a general-purpose timer that generates periodic Level 2 interrupt requests. It operates in either Mode 2 or Mode 4. Mode 4 operation is nearly identical to Mode 2 operation, but lacks the automatic reset offered in Mode 2. Only one interrupt request is generated in Mode 4, and the counter must be reloaded to initiate a new Channel 2 timing cycle. Channel 2 generates interrupt requests at priority Level 2. Bit 0 of the Interrupt Status Port (I/O Port 1022H) signals a Timer Channel 2 interrupt request. The 8086 loads or reads the Channel 2 counter at I/O Port 1044H. It clears an interrupt request from Channel 2 of the timer by reading I/O Port 10E2H.

To begin counting on any timer channel, a program first initializes the channel's Control Register by writing one byte to Output Port 1046H. As shown in Table 1-6, a Control Register byte selects a channel (0-2), a mode of operation (2 or 4), a counter data format (binary or binary coded decimal), and the number of bytes required to specify an initial counter value (1 or 2). Software then loads the counter by writing one or two bytes to the 8-bit counter port dedicated to the channel that was selected: Port 1040H for Channel 0, Port 1042H for Channel 1, or Port 1044H for Channel 2.

Control Register bytes and counter value bytes can be written in any sequence, provided that the proper number of counter value bytes follows each Control Register byte. For example, channels can be initialized one at a time, or all three Control Register bytes can be followed by from three to six counter value bytes for the various channels. A counter value must adhere to the 1- or 2-byte format designated in its Control Register.

If Control Register Bits 5 and 4 both are 0, the timer latches the value of the counter designated by Bits 7 and 6, preserving the count value (at the time that the Control Word was received) until the 8086 reads the count or reprograms the channel; this allows a program to read counter values "on the fly." The timer supplies either one or two bytes at the channel's counter port address, using whichever format was used to initialize the counter. A counter value is unlatched automatically as soon as the 8086 reads it.

Table 1-5. 8253-5 Timer I/O Ports

Port	Description
1022H	Interrupt Status Port. Bit 0 will be cleared to 0 if Timer Channel 2 has a pending interrupt request, otherwise set to 1.
1040H	Write 1- or 2-byte initial counter value for Channel 0.
Read current value of Channel 0 counter.	
1042H	Write 1- or 2-byte initial counter value for Channel 1. Channel 1 counter times dynamic RAM refresh intervals and must have an initial value of 60.
Read current value of Channel 1 counter.	
1044H	Write 1- or 2-byte initial counter value for Channel 2.
Read current value of Channel 2 counter.	
1046H	Write Control Register. Accepts one byte of data as described in Table 1-6.
10E0H	Write arbitrary data to clear Timer Channel 0 interrupt request.
10E2H	Read to clear Timer Channel 2 interrupt request.

Table 1-6. 8253-5 Timer Control Register (Output Port 1046H)

Bit	Description
0	If set to 1, read and write counter values in binary coded decimal format; otherwise, use binary format.
3, 2, 1	Mode of operation. Must be 010 except for Channel 2, which can be either 010 (automatic reset) or 100 (no reset).
5, 4	00 - Counter latch command. 01 - Omit high-order byte of counter value (0 assumed). 10 - Omit low-order byte of counter value (0 assumed). 11 - Read and write both bytes of counter value.
7, 6	00 - Control Word for Channel 0. 01 - Control Word for Channel 1. 10 - Control Word for Channel 2. 11 - Invalid and illegal.

1.6 POWER DISTRIBUTION

Figure 1-3 illustrates power distribution for the Wang PC. The power supply at the center connects to the backplane in the system board enclosure (lower left of figure), the floppy and Winchester disk drives (at right of figure), and the cooling system. The medium-resolution video monitor is powered from its controller board. Other I/O devices meet their own power requirements.

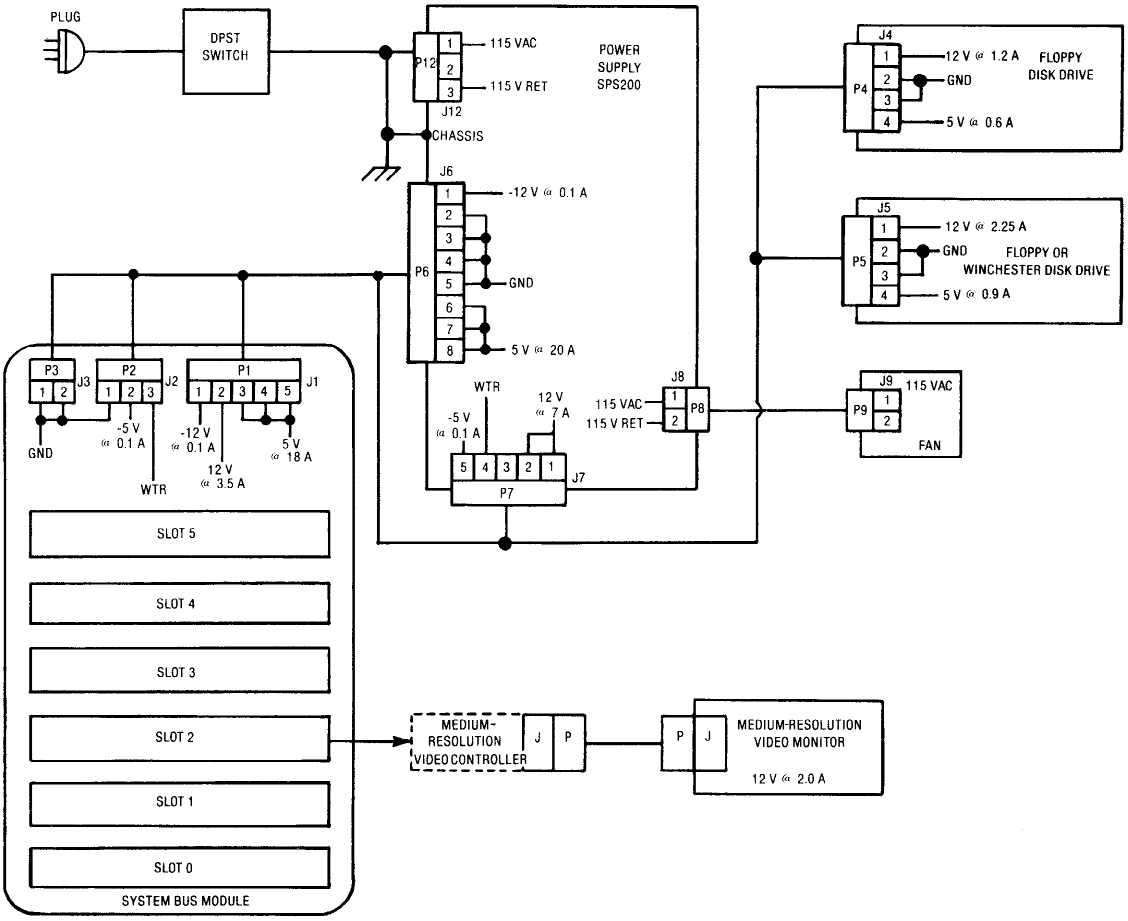


Figure 1-3. Power Distribution Diagram

CHAPTER 2
INTERRUPT SYSTEM

2.1 INTERRUPT REQUESTS

Devices that require processor attention can generate interrupt requests by activating one of the seven interrupt request lines (/IRQ0-6). The interrupt request line number determines the priority of the request, with Level 0 requests having highest priority. Only devices on the system board can generate Level 0 and Level 1 interrupt requests; therefore, /IRQ0 and /IRQ1 are not carried out onto the system bus. I/O option devices installed in expansion slots signal their interrupt requests by means of /IRQ2-6. There are generally several devices capable of activating each interrupt request line. Because interrupts are level-triggered, a device must drive its interrupt request line low and hold it low until the 8086 clears its interrupt request.

An 8259A Programmable Interrupt Controller (PIC) provides eight levels of maskable priority interrupts. Interrupt Level 0, the highest priority interrupt level, allows Channel 0 of the 8253-5 Programmable Interval Timer to generate a periodic interrupt for realtime clocking functions. Level 1 interrupts, also derived from the system board, originate only from the RS-232C serial port, the parallel port, or a time-out on Channel 2 of the 8253-5 timer. I/O option boards can generate Level 2 interrupts; however, Level 2 interrupts can also originate from the keyboard, floppy diskette controller, 8087 coprocessor, and DMA controller.

Every I/O option board that can generate interrupt requests provides a mechanism for accepting an interrupt priority level assignment. (If the option uses DMA, this also determines its DMA channel assignment, which is always four less than its interrupt priority level.) Interrupt channel assignments can change at any time and are generally transparent to any software executing on the I/O option board. Devices that need 8086 attention generate interrupt requests by activating the interrupt request line for the priority level to which they are currently assigned and by holding the interrupt request line low until the 8086 acknowledges the interrupt request.

The PIC monitors all eight interrupt request lines and keeps track of which lines are carrying active interrupt requests. When any interrupt request line is active, the PIC asserts its pending interrupt (/INT) output, which is applied directly to the interrupt request (INTR) input of the 8086. To acknowledge that it is ready to service a pending interrupt, the 8086 returns two Interrupt Acknowledge (/INTA) pulses to the PIC. Then, using either fixed or rotating priority, depending on the mode in which it was programmed to operate, the PIC identifies the active request that has highest priority, builds an interrupt vector from the level of the highest priority request and the current ICW2 value, and sends the interrupt vector across the data bus to the 8086.

An interrupt vector invokes one of eight 8086 interrupt service routines, which must identify the source of the interrupt request by polling or interrogating each device that is capable of requesting an interrupt at the priority level it services. Once it locates the device requesting service, the interrupt handler performs any functions that are needed and causes the device to remove its interrupt request. The handler concludes by writing an OCW2, which signals the end of the interrupt to the PIC.

There are generally several devices capable of activating each interrupt request line. An interrupt service routine for Level 0 or Level 1 interrupts can determine the particular source of an interrupt request by examining the Interrupt Status Port (Input Port 1022H). Interrupt service routines for Level 2 and higher interrupt levels must identify the source of an interrupt request by examining Bit 7 of the highest I/O port address on every I/O option that might have a pending request.

The fully nested interrupt structure allows a high priority interrupt request to generate an interrupt even while a lower priority interrupt is being serviced (provided that the low priority interrupt handler runs with interrupts enabled). In general, at any point in time, the 8086 can be in the process of servicing several active interrupts while several interrupt requests are pending in the PIC.

2.2 INTERRUPTS AND INTERRUPT PRIORITY LEVELS

The following list describes the devices that can generate interrupt requests and the interrupt type and level for each device.

- Nonmaskable interrupts

  - System board RAM parity error — 8086 enables NMIs by writing to I/O Port 10E2H with D0 set to 1. It disables NMIs by writing to Port 10E2H with D0 cleared to 0.

  - Option board I/O error — Option boards assert the I/O ERROR system bus line to request an NMI.

- Level 0 (highest priority) interrupt

  - Realtime clock interrupt — Set by Channel 0 of 8253-5 timer. Cleared by writing arbitrary data to Port 10E0H.

- Level 1 interrupt

  - Software timer (8253-5) interrupt — Set by terminal count signal from Channel 2 of 8253-5 timer. Cleared by reading Port 10E2.

  - 2661 programmable communications interface

    Transmitter ready — Set when Transmit Data Holding Register is ready for 8086 to write new data into it. Cleared when 8086 writes to Transmit Data Holding Register or turns off transmitter.

    Receiver ready — Set when Receive Data Holding Register contains data for 8086 to read. Cleared when 8086 reads Receive Data Holding Register or turns off receiver.

    Transmitter empty or set change — Set after transmitter serializes data and sends last character loaded by 8086 (Transmit Data Holding Register and Transmit Shift Register both empty) or when either /DSR or /DCD changes state. Cleared when 8086 reads EPCI Status Register.

  - Parallel I/O port

    Data available — Set when parallel I/O port has data for 8086 to read. Cleared when 8086 reads a byte of parallel data from Port 10EAH.

    Latched acknowledge — Set by /ACKNLG signal from parallel I/O interface. Cleared when 8086 writes next byte to Output Port 10EAH or when 8086 writes to Output Port 10ECH.

    Not busy — Set when BUSY signal from parallel printer goes inactive. Cleared when 8086 reads Input Port 10ECH.

- Level 2 interrupts

  - Option board interrupt — Set by low level on /IRQ2 from option board. Cleared by 8086.

  - DMA terminal count (Channel 1, 2, or 3) — Set by DMA controller. Cleared by 8086.

  - Keyboard Transmit Buffer Register empty — Set when keyboard Transmit Buffer Register is available to accept new data from 8086. Cleared when 8086 writes to Transmit Buffer Register or by Clear Keyboard Transmit signal from 8086.

  - Keyboard data received — Set when keyboard has data ready for 8086 to read. Cleared when 8086 reads keyboard Receive Buffer Register.

  - Floppy diskette controller chip interrupt — Set when NEC-765 chip completes a floppy diskette operation. Cleared by reading first status byte in result phase of floppy diskette operation.

  - Floppy diskette Drive 1 door disturbed or door disturbed — Set by opening door on floppy diskette Drive 1 or 2. (System Status Drive 2 Port 10E0H determines which door is open.) Cleared by setting Bit 2 (Drive 1) or Bit 3 (Drive 2) of Output Port 1000H.

  - 8087 interrupt — Set by 8087 to indicate that an unmasked exception occurred during numeric instruction execution while 8087 interrupts were enabled. Enabled, disabled, and cleared by 8086 instructions FNCLEX, FNSAVE, and FNINT.

2.3 PROGRAMMING THE INTERRUPT SYSTEM

Initialization Control Words (ICW1, ICW2, and ICW4) establish interrupt system parameters by selecting among various PIC operating modes, only some of which are valid in this implementation. Software in the 8086 designates ICW1 by writing a value of 1FH to Port 1060H. It then sets ICW2 by writing one-fourth of the interrupt vector table base address to Port 1062H. Finally, it writes a value of 0DH to Port 1062H, establishing ICW4.

Operation Control Words (OCW1-3) regulate interrupt processing by enabling or disabling various options or performing other control functions. Written to Port 1062H, OCW1 masks or unmasks any of the eight interrupt priority levels. Setting the OCW1 bit that corresponds to an interrupt level disables subsequent requests at that level. For example, writing a value of 01 into OCW1 disables Level 0 interrupt requests and enables interrupt requests at Levels 1-7. The 8086 can read OCW1 at Port 1062H to determine which interrupt priority levels are enabled. OCW1 shares a port address with ICW2 and ICW4; however, ICW2 and ICW4 always appear as the first and second inputs following an ICW1 and are thus distinguished from OCW1 by context.

OCW2 is written to Port 1060H and serves three functions:

1. Terminates an interrupt request

2. Establishes the highest interrupt priority level

3. Rotates priority levels to implement a round-robin servicing discipline

Bits 3 and 4 of OCW2 must be cleared to 0. (This distinguishes OCW2 from OCW3, which has Bit 3 set, and ICW1, which has Bit 4 set.) Bits 7, 6, and 5 determine the function performed by OCW2 as shown in Table 2-1.

Table 2-1. Format of Operation Control Word 2

Bits	Value	Operation
7, 6, 5	000	Unused.
	001	Terminate highest priority interrupt.
	010	Unused.
	011	Terminate interrupt at priority level designated by Bits 0-2.
	100	Unused.
	101	Terminate highest priority interrupt and rotate priority levels.
	110	Establish priority level designated by Bits 0-2 as lowest priority level. Next higher numbered level will then have highest priority.
	111	Terminate highest priority interrupt and establish priority level designated by Bits 0-2 as lowest priority level. Next higher numbered level will then have highest priority. This function is basically a combination of Functions 1 and 6 (001 and 110).

OCW3 is used to establish Special Mask mode or obtain interrupt status information. Interrupt service routines can use Special Mask mode to alter an interrupt priority scheme and later restore it. OCW3 also makes three different types of status data available.

1. A poll command identifies pending interrupt requests (i.e., device has posted a request but the PIC has not yet posted the corresponding interrupt).

2. An OCW3 function identifies active interrupt requests (i.e., PIC has posted an interrupt but 8086 has not yet acknowledged it).

3. A third function identifies all levels on which an interrupt is being serviced (interrupt acknowledged but interrupt request not yet cleared by an end-of-interrupt command).

Table 2-2 presents the OCW3 format, and Table 2-3 presents the addresses for the interrupt request flags.

Table 2-2. Format of Operation Control Word 3

Bit	Value	Operation
1, 0	00, 01	Unused.
	10	Subsequent reads of Port 1060H return D0-7 set only if there is a pending interrupt request at the corresponding priority level.
	11	Subsequent reads of Port 1060H return D0-7 set only if software is servicing an interrupt at the corresponding priority level.
2		Poll command. If set to 1, next read at I/O Port 1060H returns D7 set if there is a pending interrupt request and, if D7 is set, D0-2 contain the level number of the pending request that has highest priority.
4, 3	01	Bit 4 must be cleared to 0 and Bit 3 must be set to 1.
6, 5	00, 01	No effect on Special Mask mode.
	10	Disable Special Mask mode.
	11	Enable Special Mask mode, allowing an OCW1 to establish a temporary interrupt mask for use only while Special Mask mode remains enabled.
7	0	Must be cleared to 0.

Table 2-3. Interrupt Request Flag Addresses

Level	Type	Interrupt Request Flag
NMI	System Board Parity Error	Port 10E0H Bit 0 = 0
	Option Board I/O Error	Port 10E0H Bit 1 = 0
0	Realtime Clock (Timer Channel 0)	Only possible source
1	Timer Channel 2	Port 1022H Bit 0 = 0
	Serial Communication Interface	Port 1022H Bit 1 = 0
	Parallel I/O Interface	Port 1022H Bit 2 = 0
2	DMA End of Process (Terminal Count)	Port 1022H Bit 3 = 0
	Output to Keyboard	Port 1022H Bit 4 = 1
	Input from Keyboard	Port 1022H Bit 5 = 1
	Floppy Diskette Operation Finished	Port 10E0H Bit 3 = 1
	Floppy Diskette Controller Error	Port 1022H Bit 6 = 1
	Floppy Diskette Door Status	Port 10E0H Bits 4-7
	8087 Co-Processor	Port 1022H Bit 7 = 1
	Option Board Interrupt	Slot offset 10FEH Bit 7 = 1
3-6	Option Board Interrupt	Slot offset 10FEH Bit 7 = 1

CHAPTER 3
KEYBOARD AND SOUND GENERATOR

The detached keyboard contains a dedicated microprocessor that accepts commands from the 8086 while sending both keyboard status data and keystroke data back to the 8086. The 8086 sends command data to the keyboard by writing a sequence of one or more command bytes to I/O Port 10E8H. When the keyboard returns status data or keystroke data, the 8086 receives it by reading bytes from this port. Keyboard data can be sent and received simultaneously across the full-duplex keyboard interface.

3.1 KEYBOARD INTERFACE

After sending or receiving each byte of data, the keyboard interface generates a Level 2 interrupt request. Software examines Bits 4 and 5 of Input Port 1022H, the Interrupt Status Port, to identify the keyboard as the source of an interrupt request and to determine which keyboard function generated the interrupt request. A transmit data interrupt sets Bit 4 of the Interrupt Status Port when the keyboard is free to accept a byte from the 8086. A receive data interrupt sets Bit 5 of this port when the 8086 must accept a byte of data from the keyboard. (Bits 0-3 and 6-7 of the Interrupt Status Port are used with other devices.) The keyboard buffers its output data so that it need not interrupt the 8086 more often than once every 10 milliseconds.

Writing a byte of outbound data at I/O Port 10E8H automatically clears a keyboard transmit interrupt request. Alternately, if there is no more data to send, the interrupt request can be cleared by writing arbitrary data to output Port 10E6H, the keyboard's Clear Transmit Interrupt Port. Reading a byte of inbound data at I/O Port 10E8H automatically clears a keyboard receive interrupt request.

Pressing a key causes the keyboard to send a 7-bit keystroke code to the 8086. Every key has its own unique keystroke code, which is arbitrary and bears no relationship to any of the various character codes. In addition to the keystroke codes, the left and right SHIFT keys also generate different codes, called release codes, when they are released. The release code for any key is identical to its keystroke code, but with high-order Bit 7 set (i.e., keystroke code plus 80H). Special keyboard commands establish release codes for up to five other designated keys or, alternately, for all of the keys on the keyboard.

Any key on the keyboard is a potential repeat key. If the keyboard is programmed to generate a release code for a particular key, the CPU can assume that the key remains down (or pressed) between the time it receives the keystroke code and the time it receives the corresponding release code. Therefore, when a program that is monitoring the keyboard recognizes a pressed key in this way, it can decide whether the key will repeat and at what rate.

Figures 3-1 and 3-2 show keystroke code assignments for the standard keyboard and the extended keyboard, respectively. The extended keyboard is primarily for the European market. The European keyboard adds one vertical row of four new keys at the right side of the main key bank and splits the left SHIFT key to obtain a fifth new position. In addition, the key that generates Code 1F on the standard keyboard generates Code 20 on the extended keyboard.

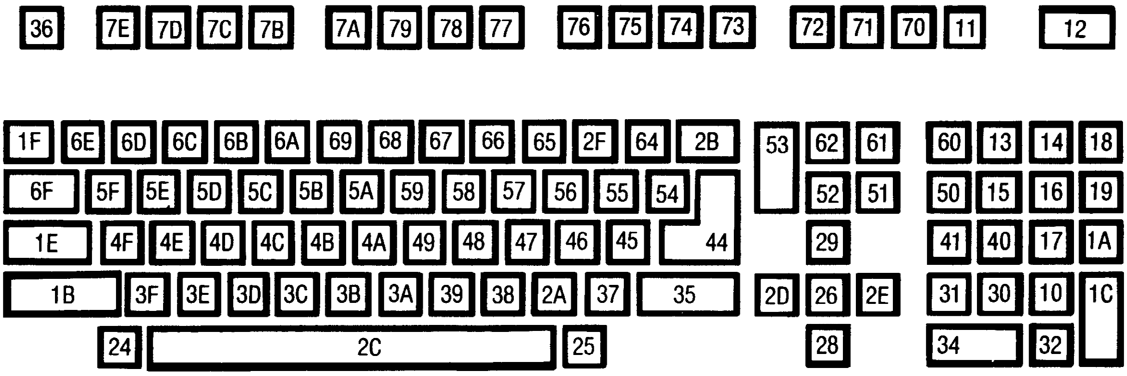

36 7E 7D 7C 7B 7A 79 78 77 76 75 74 73 72 71 70 11 12
1F 6E 6D 6C 6B 6A 69 68 67 66 65 2F 64 2B 53 62 61 60 13 14 18
6F 5F 5E 5D 5C 5B 5A 59 58 57 56 55 54 52 51 50 15 16 19
1E 4F 4E 4D 4C 4B 4A 49 48 47 46 45 44 29 41 40 17 1A
1B 3F 3E 3D 3C 3B 3A 39 38 2A 37 35 2D 26 2E 31 30 10 1C
24 2C 25 28 34 32

Figure 3-1. Standard Keyboard Mapping

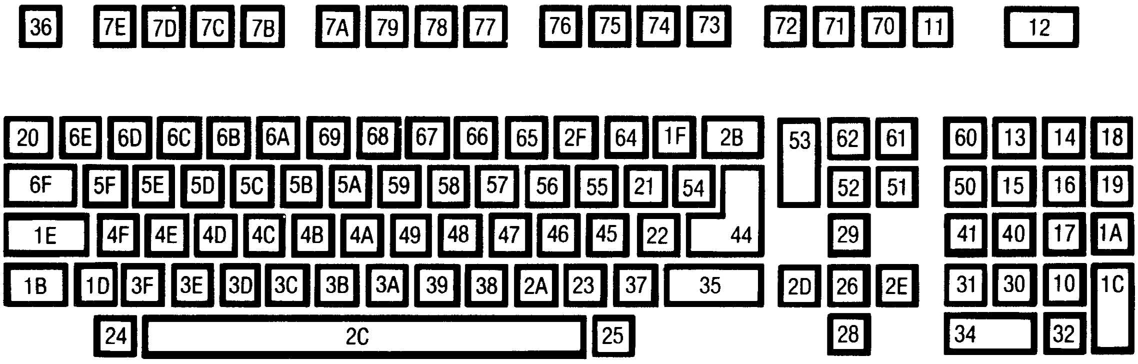

36 7E 7D 7C 7B 7A 79 78 77 76 75 74 73 72 71 70 11 12
20 6E 6D 6C 6B 6A 69 68 67 66 65 2F 64 1F 2B 53 62 61 60 13 14 18
6F 5F 5E 5D 5C 5B 5A 59 58 57 56 55 21 54 52 51 50 15 16 19
1E 4F 4E 4D 4C 4B 4A 49 48 47 46 45 22 44 29 41 40 17 1A
1B 1D 3F 3E 3D 3C 3B 3A 39 38 2A 23 37 35 2D 26 2E 31 30 10 1C
24 2C 25 28 34 32

Figure 3-2. Extended Keyboard Mapping

There are some possible keystroke codes and their corresponding release codes that are not produced by any key. One of these, Code 01, is used as a query response byte, which is always followed by one or more status bytes. When the 8086 makes an inquiry by sending one of the query control byte sequences included in Table 3-1, it next continues accepting keyboard input until it receives the 01 query response byte; it then accepts the appropriate number of status bytes before resuming normal keyboard input.

Table 3-1. Keyboard Control Codes

Code	Description
00	Unassigned.
01-05	Cancel any release codes established earlier and specify from one to five new release codes. Except for the two SHIFT keys (which always produce release codes), keyboard normally generates a keystroke code only when a key is pressed. If programmed to do so, it also generates a different code when the key is released. Release code is identical to keystroke code but with Bit 7 set (i.e., character code plus 80H). For example, the control byte sequence "02, 24H, 1EH" cancels old release codes and establishes release codes for the SHIFT LOCK key plus the CONTROL key. The control byte sequence "04, 29H, 28H, 2EH, 2DH" cancels old release codes and establishes release codes for the North, South, East, and West cursor control keys. Each of the control bytes 01-05 must be followed by the appropriate number of character codes.
06	Establish release codes for every key on the keyboard.
07	Cancel any release codes established earlier without specifying new release codes. The left and right SHIFT keys always generate release codes.
08	Power-on reset.
09	Version level query. Returns a control preamble of 01 followed by the 1-byte PROM code or mask level.
0AH	Clicker Tone character.
0BH	Beep Tone character.
0CH	Invoke Self-test mode. Causes the keyboard to finish sending the byte it is currently transmitting, if any, and then retransmit this byte or the last byte that it sent. If no bytes have been sent, the keyboard transmits a byte of 00.
0DH	Unassigned.

Table 3-1. Keyboard Control Codes (continued)

Code	Description
0EH	Device type query. Causes the keyboard to return three bytes: a control preamble of 01, a byte containing the low-order configuration switch setting, and a byte containing the high-order configuration switch setting.
0FH	Volume setting query. Causes the keyboard to return two bytes: a control preamble of 01, followed by the current volume control byte.
10H	Turn LED0 on (SHIFT LOCK key).
11H	Turn LED0 off.
12H	Turn LED1 on (LED furthest left on top row).
13H	Turn LED1 off.
14H	Turn LED2 on.
15H	Turn LED2 off.
16H	Turn LED3 on.
17H	Turn LED3 off.
18H	Turn LED4 on.
19H	Turn LED4 off.
1AH	Turn LED5 on.
1BH	Turn LED5 off.
1CH	Turn all LEDs on.
1DH	Turn all LEDs off.
1EH	LED Status query. Causes the keyboard to return two bytes: a control preamble containing 01, followed by a byte in which each of Bits 0-5 is set to 1 only if the corresponding LED is on. Bits 6 and 7 of this byte will be cleared to 0.
1FH	Unassigned.
20H	Accept new volume control byte. The byte following this control byte must have its two high-order Bits 7 and 6 cleared to 0. Bits 5, 4, and 3 specify the new clicker volume setting. Bits 2, 1, and 0 specify the new beeper volume setting. Volume settings range from 0, or fully off, to 7, the loudest possible setting.

Table 3-1. Keyboard Control Codes (continued)

Code	Description
21-30H	Unassigned.
31H	Accept one tone generator control byte. Causes keyboard to load the following byte into the tone generator.
32H	Accept two tone generator control bytes. Causes keyboard to load the following two bytes into the tone generator.
33-FFH	Unassigned.

A 4-conductor cable fitted with a 4-pin DIN connector carries power and interface signals between the keyboard interface and the detached keyboard as shown in Table 3-2. The keyboard interface is designed around a 6402 Universal Asynchronous Receiver Transmitter (UART) operating at 62.5K baud. At power-up or system reset, hardware initializes the UART to transmit and receive a serial protocol that consists of one start bit (cleared to 0), an 8-bit data word (least significant bit first), and two stop bits (set to 1), without parity.

Table 3-2. Keyboard Interface Signals

Connector Pin Number	DIN Plug Pin Number	Signal
1		Negative lead to 8-ohm speaker
2		Positive lead to 8-ohm speaker
3	1	Ground
4	2	Serial data output to keyboard
5	4	5-V power
6	3	Serial data input from keyboard

3.2 COMPLEX SOUND GENERATOR

Sound generator circuitry incorporates an SN76489AN digital complex sound generator chip to provide three programmable tone generators and a noise generator, with programmable attenuation for volume control of all sounds and the ability to produce multiple sounds simultaneously. Tone generator Channels 0, 1, and 2 each contain an independent tone generator and a dedicated attenuator. The noise generator on Channel 3 consists of a noise source with a dedicated attenuator. Attenuators provide volume control by attenuating, or reducing, the loudness of the sound.

The 8086 programs any of the three tone generators on Channels 0, 1, or 2 by loading the 10-bit Period Register for the channel, along with its 4-bit Attenuation Register. Each Period Register receives the value of the period for the output frequency produced on its channel in units of 8 microseconds. This is equivalent to the value obtained by dividing the tone frequency by 125,000. Each Attenuation Register receives an attenuation factor between 0 and 30 dB, inclusive, in units of 2 dB. The maximum attenuation factor actually shuts off the channel so that no sound is produced.

To load a Period Register, the 8086 sends the keyboard a control byte sequence consisting of the control code 32H followed by two Period Register control bytes. The first of these has Bit 7 set to 1, the tone generator channel number in Bits 6 and 5, Bit 4 cleared to 0, and the low-order four bits of the Period Register value in Bits 3-0. The second Period Register control byte has Bits 7 and 6 cleared to 0 and the high-order six bits of the Period Register value in Bits 5-0.

To load an Attenuation Register, the 8086 sends a control byte sequence to the keyboard consisting of Control Code 31H followed by one Attenuation Register control byte. The Attenuation Register control byte has Bit 0 set to 1, the tone generator channel number in Bits 6 and 5, Bit 4 set to 1, and the 4-bit attenuation factor in Bits 3-0.

To load the noise generator's Control Register, the 8086 sends a 2-byte control sequence consisting of Control Code 31H followed by one noise control byte. The noise control byte has binary value 11100 in Bits 7-3, with Bit 2 either set to 1 for Gaussian white noise or cleared to 0 for periodic noise. Bits 1 and 0 of the noise control byte designate the relative noise frequency, as shown in the following chart:

0	0	High-frequency noise
0	1	Intermediate-frequency noise
1	0	Low-frequency noise
1	1	Noise frequency determined by Channel\3 tone generator output

The procedure for loading the noise generator's Attenuation Register is the same as the procedure for loading Attenuation Registers on the other three channels. The 8086 sends a 2-byte control sequence to the keyboard consisting of Control Code 31H followed by one Attenuation Register control byte. The Attenuation Register control byte has Bit 0 set to 1, the tone generator channel number (i.e., 3 or binary 11) in Bits 6 and 5, Bit 4 set to 1, and the 4-bit attenuation factor in Bits 3-0. Attenuation factors are defined as for the three tone generator channels.

CHAPTER 4
PARALLEL I/O INTERFACE

The parallel I/O interface is a general-purpose, bidirectional port that can be used to send or receive 8-bit parallel data with associated status information. It is customarily used to interface a parallel printer, and it carries only outbound data in printer applications. Some parallel printers use a write-only interface and do not return status information. Others do return certain status flags and therefore need a bidirectional interface that handles only character data in the outbound direction but transfers status information in both directions. Table 4-1 describes the various types of status information that can be transferred across the parallel I/O interface.

Table 4-1. Parallel Printer Interface Signals for Input Port 1020H and I/O Port 1024H

Port	Bit	Signal Name and Description
1020H	0	/POWER ON. Generated only by intelligent printers. When cleared to 0, indicates that printer power is applied.
	1	/SMART. Generated by intelligent printer. When cleared to 0, indicates that the printer supports a bidirectional interface. Printers that lack microprocessors either supply 5 V or have no connection, which presents an inactive level (bit set to 1) to indicate a write-only interface.
	2	/DATA AVAILABLE. When cleared to 0, indicates that the transmit buffer on the printer contains a byte of data for the 8086 to read.
	3	SLCT. Set to 1 when the printer is selected.
	4	BUSY Flag (from Pins 11 and 29 of connector). When set to 1, indicates that the printer is busy and cannot receive data.

Table 4-1. Parallel Printer Interface Signals for Input Port 1020H and I/O Port 1024H (continued)

Port	Bit	Signal Name and Description
1020H (cont.)	5	/FAULT. When cleared to 0, indicates that the printer is offline, positioned at the end of a page, or in a similar state that requires operator intervention.
	6	PE. When set to 1, indicates that the printer is out of paper.
	7	ACKNOWLEDGE Flag (latched /ACKNLG from Pins 10 and 28 of connector). When cleared to 0, indicates that the printer has processed the last byte of output and is now ready to accept another data byte. Cleared by any write to I/O Port 10ECH.
1024H	0	/USR0. When cleared to 0, requests automatic line feed after carriage return on Epson printers.
	1	/USR1. When cleared to 0, selects an Epson printer.
	2	/RESET. When cleared to 0, resets the printer to its power-on state. Software sets this bit, waits at least 50 usec, and then clears this bit to initialize the printer.
	3	Unassigned.
	4-7	SW1-4. Four-bit switch settings. Cleared to 0 if switch is closed (or ON), and set to 1 if switch is open (or OFF). These switches are located on the system board about midway between the 96-pin backplane connector and the 26-pin D-type connector for the RS-232C interface. They are positioned with SW1 closest to the 96-pin connector and SW4 closest to the RS-232C connector. System software typically uses these switches to establish the default baud rate for the 2661 EPCI (by loading SW1-4 directly into Bits 0-3 of EPCI Mode Register 2). SW1-4 are not part of the parallel printer interface.

The 8255A Parallel Peripheral Interface chip used to implement the parallel I/O interface also serves two other functions:

1. It contains the Interrupt Status Port (Input Port 1022H).
2. It makes the 4-bit hardware switch setting available to 8086 software.

The 8255A has a control word register that must be initialized by writing 9BH to Output Port 1026H. Interrupt Status Port operation was described in Chapter 2. Instructions for reading the hardware switches are included in this chapter.

Table 4-2 shows the power requirements for the I/O ports.

Table 4-2. Power Requirements and Characteristics

Mode	Requirements and Characteristics
Output	Vo_L = 0.5 V max at 24 mA min. Vo_H = 3.0 V min at 2 mA min. All outputs terminated with a 4.7K-ohm pull-up resistor.
Input	Vo_L = 0.8 V max at 2 mA min. Vo_H = 2.4 V min at 0.5 mA min. All inputs are terminated with 4.7K-ohm pull-up resistors, a 150-ohm resistor connected between the buffer input and the pull-up resistor, and a 180-PF capacitor between the buffer input and ground.
AC Output	Rise and fall times less than 200 nanoseconds. Data set-up time (to /DSTB IN high) of at least 1.5 microseconds. Data hold time (to /DSTB IN low) extends until acknowledge is received or program reads input port. /DSTB IN pulse width of at least 2 microseconds. ACKNOWLEDGE pulse width of at least 100 nanoseconds.
AC Input	Rise and fall times less than 200 nanoseconds. Data set-up time (to /DSTB OUT low) of at least 150 nanoseconds. Data hold time (to /DSTB OUT low) of 0. /DSTB OUT pulse width of at least 500 nanoseconds.

Printers that use a bidirectional interface are usually characterized as intelligent or "smart" terminals. However, other I/O devices might also implement the standard printer protocol and send character data back to the 8086. For example, with proper cabling, two systems can communicate across the parallel I/O interface with each appearing as a printer to the other. Table 4-3 contains the pin assignments for a parallel I/O interface cable.

Table 4-3. Parallel Port Pin Assignments

Pin Number		Signal	I/O
1,19	8	/DSTB IN	O
2,20	18	DATA 1	I/O
3,21	17	DATA 2	I/O
4,22	16	DATA 3	I/O
5,23	15	DATA 4	I/O
6,24	14	DATA 5	I/O
7,25	1	DATA 6	I/O
8,26	2	DATA 7	I/O
9,27	3	DATA 8	I/O
10,28		/AKNLDG	I
11,29	7	BUSY	I
12	9	PE	I
13	10	SLCT	I
14		/USR0/AUTO FEED XT	O
15		/POWER ON / OSCXT	I
18		/SMART / 5 V	I
31,30		/RESET	O
32,33		/FAULT	I
34,16		/DSTB OUT	O
35,17		/DAV	I
36		USR1	O

A program sends output data across the parallel I/O interface by first testing the Busy flag (Bit 4 of Input Port 1020H, which attaches to Pins 11 and 29 of the parallel interface connector). After verifying that the Busy flag is cleared to 0, the program writes a byte of data to I/O Port 10EAH. The OUT 10EAH loads eight data bits into the parallel I/O interface and triggers a strobe pulse, called /DSTB IN, which strobes the data into the printer or other device attached to the interface. (Data input and output strobe signals are named from the printer's point of view.)

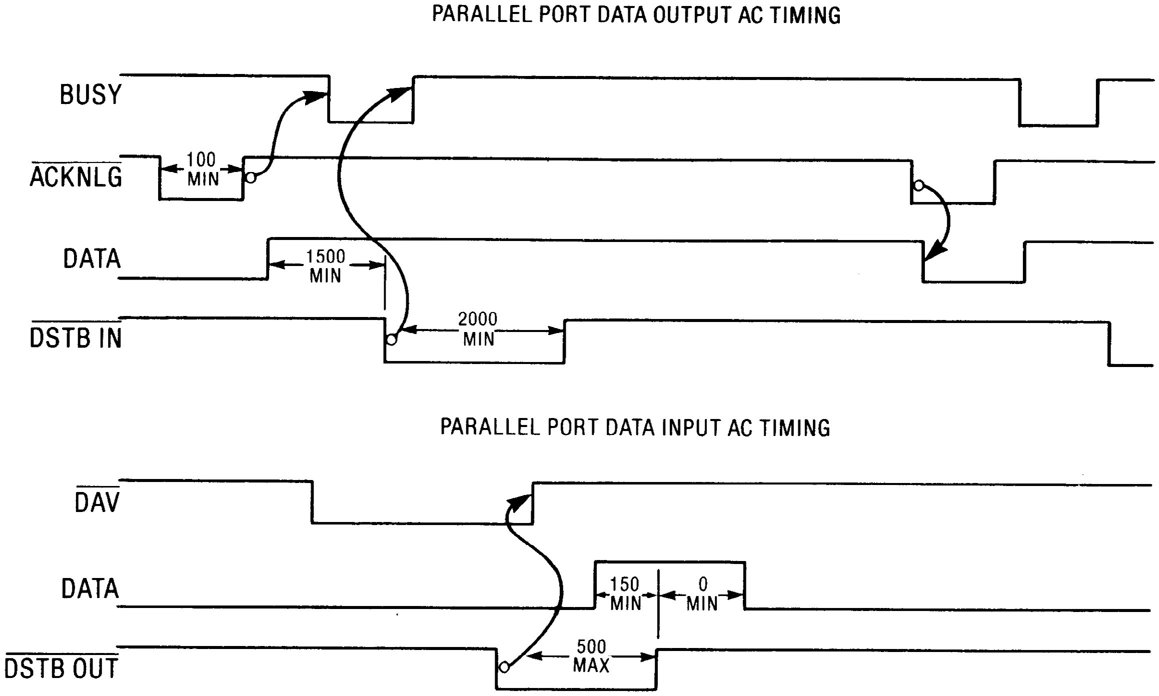
The program then monitors the Acknowledge Flag, Bit 7 of Input Port 1020H. Latched from Pins 10 and 28 of the parallel interface connector, this flag will be set to 1 by any write at Output Port 10EAH. It remains set while the device attached to the parallel I/O interface is busy accepting data; it is cleared to 0 only when the device finishes receiving a byte. Software monitors the Acknowledge Flag to determine when the device is ready to accept the next byte of output. When it finds that the Acknowledge Flag is cleared to 0, the program again tests the Busy Flag and repeats the output sequence. Although the Acknowledge Flag is normally cleared by the device attached to the parallel I/O interface, the 8086 can clear it by writing arbitrary data to Output Port 10ECH.

A program receives input data across the parallel I/O interface by first testing the Data Available Flag (Bit 2 of Input Port 1020H, which attaches to /DAV on Pins 35 and 17 of the connector). This bit will be cleared to 0 only when the device connected to the parallel I/O interface has a byte of input data available for the 8086 to accept. Once Bit 2 of Input Port 1020H becomes 0, reading I/O Port 10EAH triggers a strobe pulse, called /DSTB OUT, which is used to accept the input data byte. Every read access at I/O Port 10EAH injects two wait states to achieve a minimum /DSTB OUT pulse width of 500 ns. The IN 10EAH also sets the Data Available Flag back to 1.

For interrupt-driven transfers across the parallel I/O interface, a Level 1 interrupt signals to the 8086 that the device attached to the interface has a byte of input data (/DAV asserted on connector Pins 35 and 17) or that the device is ready to accept a byte of output data (either latched BUSY to indicate that BUSY is asserted on connector Pins 11 and 29, or latched /ACKNLG to indicate that /ACKNLG is asserted on Pins 10 and 28). Software services the parallel printer interrupt by first examining Input Port 1020H to distinguish input requests from output requests and then reading or writing a byte at Port 10EAH.

An OUT 10EAH automatically clears the latched /ACKNLG signal that causes an output interrupt request. An IN 10EAH generates a /DSTB OUT, which automatically clears the /DAV that causes an input interrupt request. The 8086 can read the BUSY signal that enters via Pins 11 and 29 on the parallel interface connector, but it cannot read the latched BUSY signal that serves as the third source of interrupt requests. If necessary, however, the 8086 can clear the latched BUSY signal by reading I/O Port 10ECH.

Figure 4-1 shows parallel port timing. I/O devices can use the parallel port as a general-purpose interface if the devices meet the input and output AC timing requirements shown in the figure. (All times are in nanoseconds.)



PARALLEL PORT DATA OUTPUT AC TIMING

PARALLEL PORT DATA INPUT AC TIMING

Figure 4-1. Parallel Port Timing Diagram

CHAPTER 5
2661 EXTENDED PROGRAMMABLE COMMUNICATION INTERFACE

The 2661 Extended Programmable Communication Interface (EPCI) is programmed by the 8086 to support RS-232-C asynchronous serial data communication in full- or half-duplex mode. It features 5- to 8-bit characters; 1, 1.5, or 2 stop bits; odd, even, or no parity; overrun, parity, and framing error detection; line-break detection and generation; false start bit detection; automatic Serial Echo mode; and local or remote loopback operation for diagnostics. Unlike many other RS-232-C interfaces, which drive as little as 50 feet of cable, the EPCI and its associated circuits communicate across distances of up to 2000 feet. The EPCI contains a baud rate generator that can be programmed to produce internal transmit and receive clocks. It operates at 16 commonly used rates, ranging from 50 to 19,200 baud.

5.1 EPCI ARCHITECTURE

The EPCI contains a transmitter and a receiver that operate independently. The transmitter accepts parallel data from the 8086, converts it into a serial bit stream, inserts any additional bits required by the programmed communication technique, and generates a composite serial data stream on its output line. The receiver accepts serial data from the RS-232-C interface, converts it to parallel format, strips off bits that implement the communication technique, and sends an assembled data character to the 8086.

A simplified block diagram in Figure 5-1 shows the EPCI transmitter, receiver, and control circuitry. To program the EPCI, the 8086 writes data into the two Mode Registers to establish the baud rate, parity, character format, and related parameters. It also writes data to the Command Register to enable various communication options. The processor monitors the Status Register to determine the detailed state of the communication channel as data is transferred to and from the EPCI.

Both the transmitter and the receiver contain two 8-bit data registers: a holding register that exchanges parallel information with the data bus, and a shift register that exchanges serial data with the RS-232-C interface. Bytes of outbound data enter the Transmit Data Holding Register, where they are then transferred into the Transmit Shift Register and shifted out as a serial bit stream. Bits of inbound data enter the Receive Shift Register, where they are assembled into complete bytes, transferred into the Receive Data Holding Register, and eventually gated onto the data bus.

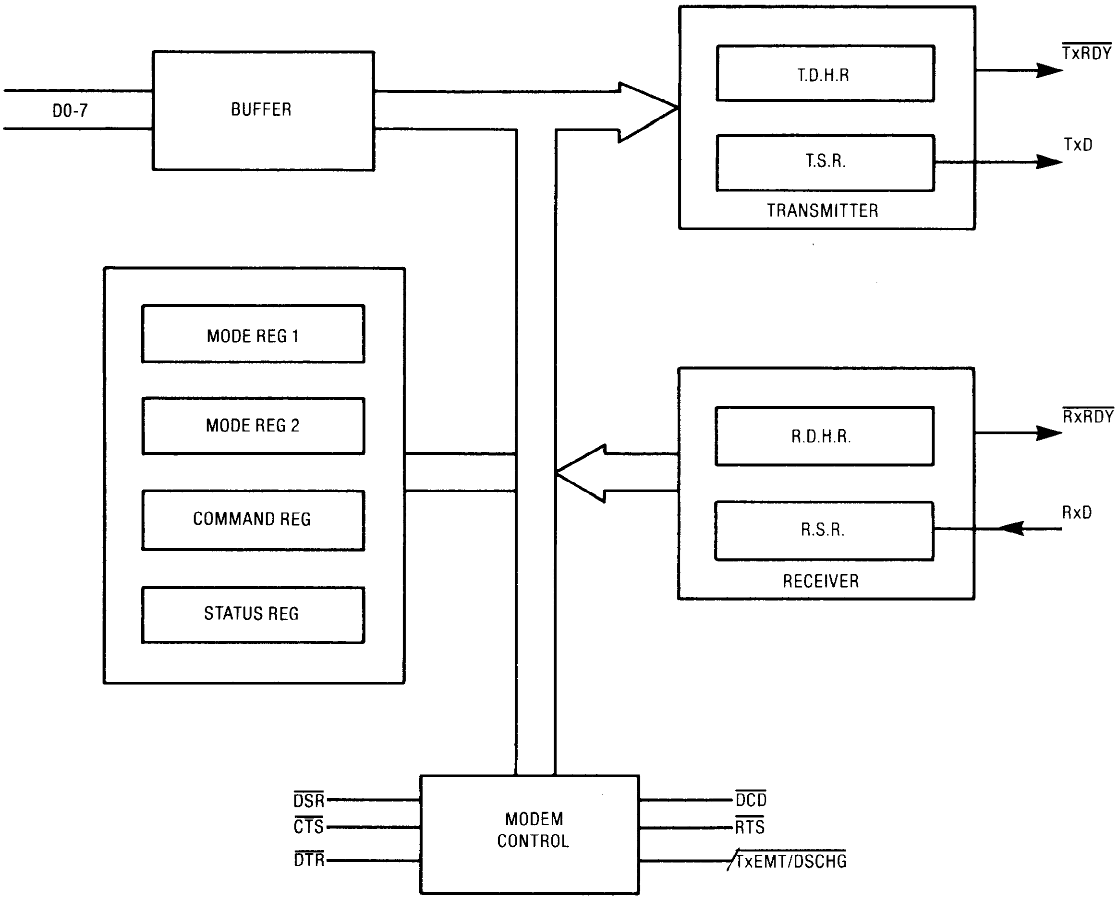


Figure 5-1. EPCI Interface Block

5.2 RECEIVER OPERATION

When its Data Carrier Detect (/DCD, Pin 8) input from the RS-232-C interface is low and the Receive Enable bit of the Command Register (RxEN, bit 2) is high, the receiver begins accepting data by seeking the high-to-low transition of a start bit on its Receive Data (RxD) input line. After detecting a start bit transition, the receiver again samples the RxD line one-half bit time later. If RxD is now high, it continues seeking a start bit; otherwise, it has found the start bit and can now begin to sample the input line, one bit at a time, until it has assembled the proper number of data bits, a parity bit, and one or more stop bits. It then transfers the data to its Receive Data Holding Register, sets the Receive Ready bit (RxRDY) in the Status Register, and asserts its /RxRDY output line, causing a Level 1 interrupt. The 8086 can now read the character at I/O Port 1080H.

The receiver zeros unused high-order bits of the incoming character code and loads the Parity Error, Framing Error, and Overrun Error bits in the Status Register. When it detects that the line was low for the entire character (including stop bits), the receiver signals a line break by transferring one character with Code 0 into the Receive Data Holding Register and setting the Framing Error bit in the Status Register. Following a line break, the RxD input line must return high before the receiver will begin searching for the next start bit.

5.3 TRANSMITTER OPERATION

When its Clear to Send (/CTS, Pin 5) input is low and the Transmit Enable Flag (TxEN, Bit 0) is set in the Command Register, the transmitter indicates to the 8086 that it can begin accepting data by setting the Transmit Ready Bit (TxRDY, Bit 0) in the Status Register and asserting its Transmit Ready output (/TxRDY), which causes a Level 1 interrupt. Then, when the 8086 writes a character into its Transmit Data Holding Register by means of I/O Port 1088H, the transmitter reverses both of these conditions, clearing /TxRDY. As soon as its Transmit Shift Register becomes free, the transmitter transfers the data character out of its Transmit Data Holding Register and again asserts /TxRDY, thereby providing one full character of output buffering.

After automatically sending a start bit, the transmitter sends the proper number of data bits, beginning with the least significant bit, and appends an optional parity bit plus the proper number of stop bits. Then, if a new character is not yet available in the Transmit Data Holding Register, it keeps its TxD output line high, sets the TxEMT/DSCHG bit in its Status Register, and asserts its TxEMT/DSCHG output, which causes a Level 1 interrupt. Transmission resumes when the 8086 loads a new character into the Transmit Data Holding Register. Setting the Force Break Flag (Bit 3 of the Command Register) high forces a continuous low (i.e, a line break) at the transmitter output.

5.4 PROGRAMMING THE EPCI

To establish data communication, an 8086 program first disables any receive or transmit operation that is currently in progress and sets the EPCI operating mode by loading its two Mode Registers and its Command Register. Mode Register 1 must be loaded first; Mode Register 2 need not be loaded if external transmit and receive clocks are used. The 8086 program accesses EPCI internal registers by means of eight I/O ports, only seven of which are used for asynchronous communication. Table 5-1 lists the I/O port address assignments.

Table 5-1. EPCI I/O Port Assignments

Port	Address
Read Receive Holding Register	1080H
Read Status Register	1082H
Read Mode Registers 1 and 2	1084H
Read Command Register	1086H
Write Transmit Holding Register	1088H
Write SYN1, SYN2, and DLE registers (unused)	108AH
Write Mode Registers 1 and 2	108CH
Write Command Register	108EH

5.5 EPCI MODE REGISTERS

When reading (or writing) Mode Register 1 and Mode Register 2, the first access at I/O Port 1084H (or 108CH) references Mode Register 1; the next access at the same address references Mode Register 2. Performing a RESET or reading the Command Register initializes the internal pointers to address Mode Register 1. Tables 5-2 and 5-3 list Mode Register bit assignments.

Table 5-2. EPCI Mode Register 1 (Input Port 1084H, Output Port 108CH)

Bit	Value	Meaning
1, 0	00	Invalid
	01	Asynchronous communication at clock rate
	10	Asynchronous communication at 16 times clock rate
	11	Asynchronous communication at 64 times clock rate
3, 2	00	5-bit character
	01	6-bit character
	10	7-bit character
	11	8-bit character
4		Enable parity generation and checking when set to 1. Disable parity when cleared to 0
5		Even parity when set to 1 and parity is enabled. Odd parity when cleared to 0
7, 6	00	Invalid
	01	One stop bit
	10	One and one-half stop bits
	11	Two stop bits

Table 5-3. EPCI Mode Register 2 (Output Port 1084H)

Bit	Value	Meaning
3, 2, 1, 0	0000	50 baud
	0001	75 baud
	0010	110 baud
	0011	134.5 baud
	0100	150 baud
	0101	300 baud
	0110	600 baud
	0111	1200 baud
	1000	1800 baud
	1001	2000 baud
	1010	2400 baud
	1011	3600 baud
	1100	4800 baud
	1101	7200 baud
	1110	9600 baud
	1111	19,200 baud
4	1	Use internal clock
	0	Use external receiver clock
4	1	Use internal transmitter clock
	0	Use external transmitter clock
7, 6		Unassigned

5.6 EPCI COMMAND REGISTER

Accessed at Input Port 108EH and Output Port 1086H, Command Register Bits 0 and 2 enable and disable the transmitter and receiver, respectively (refer to Table\5-4). Once disabled, the transmitter sends any character in the Transmit Data Holding Register before stopping. Its TxD output then remains in the mark (high) state, while /TxRDY and /TxEMT both go inactive. Disabling the receiver terminates any input operation immediately, even if a character is being assembled.

Command Register Bits 1 and 5 control the Data Terminal Ready (/DTR, Pin 20) and Request to Send (/RTS, Pin 4) outputs, respectively. Data at the output pin is the logical complement of the Command Register bit value. Setting Command Register Bit 3 forces the TxD output low, to its space condition, only after the current character is sent. TxD will later go high for at least one bit time before the next character is sent. Command Register Bit 4 clears all three error flags in the Status Register. This bit resets automatically.

Bits 6 and 7 of the Command Register regulate the EPCI operating mode. Transmitter and receiver operate independently in normal mode, as specified by Mode Register and Status Register contents. In Automatic Echo mode, incoming data is regenerated, clocked, and directed to the TxD line while normal receiver operation continues. The receiver must be enabled (Command Register Bit 2 set), but the transmitter need not be. Communication from the 8086 to the receiver continues normally, but the link from the 8086 and the transmitter is broken. Only the first character of a break is echoed, and the TxD output goes high until the next valid start is detected.

Table 5-4. EPCI Command Register

Bit	Signal	Description
0	TxEN	Transmit Enable. Activates transmitter when set to 1 and disables transmitter when cleared to 0.
1	/DTR	Data Terminal Ready. Forces /DTR output active (low) when set to 1 and inactive when cleared to 0.
2	RxEN	Receive Enable. Activates receiver when set to 1. Deactivates receiver when cleared to 0.
3		Forces a line break when set to 1.
4		When set to 1, resets Status Register error flags for Framing Error (FE, Bit 5), Overrun Error (OE, Bit 4), and Parity Error (PE, Bit 3).
5	/RTS	Request to Send. Forces /RTS output active (low) when set to 1 and inactive when cleared to 0.
7, 6		Operating mode: 00 - Normal operation 01 - Automatic Echo mode 10 - Local loopback test mode 11 - Remote loopback test mode

While in Automatic Echo mode, the EPCI automatically places data assembled by the receiver into the Transmit Data Holding Register. The transmitter then sends this data out on the TxD output line. The transmit clock equals the receive clock, in Automatic Echo mode, and the TxRDY output is always high. The /TxEMT/DSCHG output reflects only the condition of the data set. The TxEN command (Command Register Bit 0) is ignored.

In the first of the two diagnostic modes, the Local Loopback mode, all transmitter output loops back to the receiver input. In this mode, /DTR is connected to /DCD, and /RTS is connected to /CTS. The receive clock equals the transmit clock; the /DTR, /RTS, and TxD outputs all are held high; and the /CTS, /DCD, /DSR, and RxD inputs all are ignored. Command Register Bits 0 (TxEN), 1 (/DTR), and 5 (/RTS) must all be set to 1 in Local Loopback mode. Command Register Bit 2 (RxEN) is ignored.

In the second diagnostic mode, called Remote Loopback mode, data assembled by the receiver is automatically placed in the Transmit Data Holding Register and again sent by the transmitter in the TxD output. The transmit clock equals the receive clock. No data is sent to the 8086, but the three error flags are set in the Status Register. The /RxRDY, /TxRDY, and /TxEMT/DSCHG outputs all are held high. Command Register Bit 1 (TxEN) is ignored, but all other signals operate normally.

5.7 EPCI STATUS REGISTER

Accessed by means of Input Port 1082H, Status Register bits indicate receiver and transmitter state and the condition of the data set (refer to Table 5-5). Status Register Bit 0 (TxRDY) indicates that the transmitter is ready and is valid only when the transmitter is enabled. If cleared to 0, it signifies that the Transmit Data Holding Register contains a character that has not yet been transferred to the Transmit Shift Register. If set to 1, it signifies that the Transmit Data Holding Register is empty and available to accept another character from the 8086. TxRDY is initially set to 1 when the transmitter is enabled (unless the 8086 loaded a character into the Transmit Data Holding Register before enabling the transmitter). It is held set when operating in Automatic Echo mode or in Local Loopback mode.

Status Register Bit 1 (RxRDY) indicates the condition of the Receive Data Holding Register. If set to 1, it signifies that a character was loaded into the Receive Data Holding Register from the Receive Shift Register, and that this character is available for the 8086 to read. This bit is cleared when the 8086 reads the Receive Data Holding Register or when the receiver is disabled. While it remains set to 1, the /RxRDY output is held active (low).

Status Register Bit 3 is set to indicate that the /DSR or /DCD input line has changed state or that the Transmit Shift Register and the Transmit Data Holding Register are both empty. This bit is cleared when the transmitter is first enabled, and it will not be set until the transmitter processes at least one character. The 8086 reads the Status Register to determine which of the two possible conditions activated /TxEMT/DSCHG; this Status Register read in turn clears the TxEMT/DSCHG flag. While it is set, this bit holds the /TxEMT/DSCHG output low. Status Register Bit 3 causes an interrupt when set.

The Overrun Error bit indicates that the 8086 did not read a character from the Receive Data Holding Register before it was overwritten by the next inbound character in the data stream. The Framing Error bit indicates that an incoming character was not framed by the number of stop bits programmed in the Command Register. Disabling the receiver clears all Status Register error bits, as does setting Bit 3 of the Command Register.

Table 5-5. EPCI Status Register (Input Port 1082H)

Bit	Signal	Description
0	TxRDY	Transmit Ready. Indicates that Transmit Data Holding Register is empty when set; this generates a Level 1 interrupt and sets Bit 1 of Parallel Port B. If clear, Transmit Data Holding Register holds next outbound character.
1	RxRDY	Receive Ready. Indicates that Receive Data Holding Register is empty when set; this generates a Level 1 interrupt and sets Bit 1 of Parallel Port B. If clear, Receive Data Holding Register holds next inbound character.
2	TxEMT/ DSCHG	Transmitter Empty or Data Set Change. When set, indicates that either /DSR input or /DCD input has changed state, or that Transmit Data Holding Register and Transmit Shift Register are both empty. Generates a Level 1 interrupt and sets Bit 1 of Parallel Port B.
3		Parity Error.
4		Overrun Error.
5		Framing Error.
6	/DCD	Data Carrier Detect. Set to indicate that /DCD input is active (low).
7	/DSR	Data Set Ready. Set to indicate that /DSR input is active (low).

5.8 RS-232-C INTERFACE

The RS-232-C interface requires one 25-pin, D shell male connector. Use of a short cable (less than about 50 feet) is recommended; however, a longer cable can be used if the resulting load capacitance does not exceed 2500 pounds/foot (measured at the interface point and including the signal terminator). Table\5-6 lists RS-232-C interface signals, their direction as seen by the EPCI, and their pin assignment on the interface cable.

Table 5-6. RS-232-C Interface Signals

Signal	I/O	Pin	Description
	7	Signal ground and common return.	
TxD	O	2	Transmit Data. Serial data output line from transmitter. Value of 1 or mark is high and value of 0 or space is low. Held in mark condition when transmitter is disabled.
RxD	I	3	Receive Data. Serial data input to receiver. Mark is high and space is low.
/RTS	O	4	Request to Send. General-purpose output is the complement of Command Register Bit 5.
/CTS	I	5	Clear to Send. Must be low for transmitter to operate.
/DSR	I	6	Data Set Ready. General-purpose input. Its complement is Status Register Bit 7.
/DTR	O	20	Data Terminal Ready. General-purpose output is the complement of Command Register Bit 1.
/DCD	I	8	Data Carrier Detect. Must be low for receiver to operate. DCD changing state causes an interrupt. Its complement is Status Register Bit 6.

CHAPTER 6
FLOPPY DISK CONTROLLER

The floppy disk controller (FDC) operates one or two 5.25-inch, double-sided, double-density, floppy disk drives recorded at 48 tracks/inch. (For more detailed information, refer to Table 6-6 and Figure 6-1.) The FDC executes 15 different commands that software initiates by transferring multiple bytes of data. Some commands generate results that are returned to the processor by multiple-byte data transfers from the FDC. Because it can involve the transfer of several bytes in either direction, command execution is most conveniently viewed in four phases. The initialization phase begins the command sequence. During initialization, the 8086 accesses I/O ports to select or deselect a drive, turn a drive motor on or off, and perform other command set-up functions. The initialization phase involves only auxiliary circuits, not the uPD765 FDC chip itself.

Next, during the command phase, the 8086 selects a drive and head (i.e., side) and sends the FDC all the information needed to perform a particular operation.

The execution phase follows the command phase. During the execution phase, the FDC executes the function it was instructed to perform.

Following the execution phase, after the function has been performed, the result phase makes auxiliary status information and other housekeeping data available to the 8086. Table\6-1 lists the FDC I/O ports and describes the functions they perform.

6.1 FLOPPY DISK REGISTERS

Software can access three 8-bit registers in the FDC. Before a command is executed, the processor writes command byte sequences to an FDC Command Register at I/O Port 1016H. Command byte sequences differ among the various FDC commands and range in length from one to nine bytes. After a command is completed or terminated abnormally, the processor reads status byte sequences from an FDC Operation Status Register, which is also located at I/O Port 1016H. Most FDC commands return seven result phase bytes that supply information about the last operation performed and the current state of the FDC.

The third FDC register is a Controller Status Register containing eight status flags, only five of which are used in this implementation. Unlike the FDC Operation Status Register, which must be read at the conclusion of most disk operations, the FDC Controller Status Register can be examined at any time by reading I/O Port 1014H. Whereas the Operation Status Register supplies information primarily about the disk drives and the way in which the last FDC command affected them, the Controller Status Register supplies information primarily regarding the FDC interface circuitry.

Table 6-1. Floppy Disk Controller I/O Ports

Port	Description	
1000H	Write with	
	D0	Cleared to 0 to prevent uPD765 FDC chip from receiving an End of Process (/EOP) signal when the 9517A-4 DMA controller reaches terminal count. Write with D0 set to 1 to enable receipt of /EOP.
	D1	Set to 1 to prevent FDC chip from generating a Channel 2 DMA request (/DREQ2 or receiving a DMA acknowledge (/DACK2). This frees DMA Channel 2 for use by another device. Write with D1 cleared to 0 enables Channel 2 DMA operation.
	D2	Set to 1 to clear the "door disturbed" interrupt request from Drive 1.
	D3	Set to 1 to clear the "door disturbed" interrupt request from Drive 2.
1004H	Read or write access deselects Drive Unit 1 and extinguishes its drive select LED.	
1006H	Read or write access selects Drive Unit 1 and illuminates its drive select LED.	
1008H	Read or write access deselects Drive Unit 2 and extinguishes its drive select LED.	
100AH	Read or write access selects Drive Unit 2 and illuminates its drive select LED.	
100CH	Read or write access turns off motor in Drive Unit 1.	
100EH	Read or write access turns on motor in Drive Unit 1.	
1010H	Read or write access turns off motor in Drive Unit 2.	
1012H	Read or write access turns on motor in Drive Unit 2.	

Table 6-1. Floppy Disk Controller I/O Ports (continued)

Port	Description
1014H	Read FDC Controller Status Register.
1016H	Read FDC Operation Status Register. Write FDC Command Register.
1018H	Read or write access here or at 101AH resets the uPD765 FDC chip.
101CH	Read or write access here or at 101EH issues Terminal Count signal (TC) to the uPD765 FDC chip.
10FEH	Read Option ID. Returns Bits 0-6 cleared to 0 (the system board ID code) with Bit 7 set to 1 only if the FDC has a pending interrupt request.

The FDC and the 8086 operate asynchronously. Therefore, during the command phase and the result phase, the processor reads the FDC Main Status Register before transferring each byte of data to the Command Register or from the Operation Status Register. Table 6-2 shows Main Status Register bit assignments. Bits 0 and 1 are set only while their respective drives are performing a seek operation. The FDC sets Bit 4 at the beginning of the command phase, when it receives the first byte of a command byte sequence; and it clears Bit 4 at the end of result phase. When this bit is 0, the FDC can accept the first byte of a new command sequence.

Bits 6 and 7 are the only Main Status Register bits normally examined by software. Bit 6 determines the direction of transfer across the bidirectional data bus. It is set when the 8086 can read the FDC Operation Status Register or transfer input data from the FDC. When cleared, Bit 6 indicates that the 8086 can write the Command Register and transfer output data to the FDC. Bit 7 is set to indicate that the FDC is ready to send or receive data.

Table 6-2. Floppy Disk Controller Status Register (Port 1014H)

Bit	Meaning if Set
0	Drive Unit 1 busy.
1	Drive Unit 2 busy.
2-3	Unused and always 0.
4	FDC busy.
5	Unused and always 0.
6	Direction of Data I/O. If set, 8086 can read FDC's Data Register. If clear, FDC will accept data from 8086.
7	FDC Ready.

6.2 PROCESSOR AND FLOPPY DISK COMMUNICATION

Software uses Main Status Register Bits 6 and 7 for a handshaking protocol based on "ready" and "direction" status. Before sending data to the FDC, a program reads the Main Status Register and waits for input status with ready status (Bit 6 = 0 and Bit 7 = 1). Before receiving data, the program waits for output status with ready status (Bit 6 = 1 and Bit 7 = 1). The FDC sets Bit 7 to 0 for between 2 and 50 us while it accepts each byte of a command byte sequence and sets up a command. It sets Bit 6 to 1 when it has a byte of data available for the processor to read.

During the command and result phases, a program must check the Main Status Register before transferring each of up to nine bytes in either direction; it need not, however, read the Main Status Register during the execution phase. While executing a command, the FDC generates a Channel 2 DMA request (/DREQ2) whenever it needs a byte of data or has a byte of data available. The DMA controller responds with a DMA Acknowledge (/DACK) and read or write signals, depending on the direction of the transfer. When the DMA controller indicates that terminal count was reached (or software accesses I/O Ports 101C-EH to force terminal count), a Level 2 interrupt request signals the end of the execution phase and the beginning of the result phase. Reading the first byte of result phase data automatically resets the FDC interrupt request.

Most FDC commands return seven bytes of data during the result phase: three bytes of auxiliary status information and four bytes of disk address data. Regardless of the amount, which varies among commands, a program must read all of the data generated by an FDC command before it can issue a new command. An FDC command is not finished until the 8086 has read all of the data made available during its result phase.

6.3 COMMAND AND RESULT FORMATS

The 15 FDC operations comprise three command groups with various command and result formats. (Section 6.8 summarizes FDC commands.) Nine of the read, write, and scan operations form one group that uses a standard 9-byte command sequence and a standard 7-byte result format. One specialized disk formatting operation has a unique command sequence, but uses the standard result format. The remaining six special-purpose commands have their own unique command sequence and their own unique result format.

The standard 9-byte command sequence begins with a command code byte that determines the operation to be performed (Bits D0-4) and the recording format (D6). For certain operations that use the standard command format, Bits D5 and D7 of the command code byte indicate whether the command allows a "skip option" and whether the command will access both tracks in a cylinder, respectively. Special sections of this chapter describe both of these options.

The second byte of a standard 9-byte command sequence contains a value of 0 to access Side 0 of the disk or a value of 4 to access Side 1. Bytes 3, 4, and 5 of a standard 9-byte command sequence designate a disk address by supplying the track number, the side, and the sector number, respectively. The "side" value in the fourth byte should be 0 to access Side 0 or 1 to access Side 1. It must agree with the value (0 or 4) that was supplied in the second command byte.

Content of the remaining four bytes depends on the sector size. In order, Bytes 6 through 9 of the standard command sequence should contain the following decimal values:

1, 16, 50, 0	(for 16 sectors/track and 256 bytes/sector)
2, 8, 80, 0	(for 8 sectors/track and 512 bytes/sector)
3, 4, 120, 0	(for 4 sectors/track and 1024 bytes/sector)

All commands return seven bytes of result phase information except the RECALIBRATE, SENSE INTERRUPT STATUS, SPECIFY, SENSE DRIVE STATUS, and SEEK commands. The first three bytes in the standard result sequence are Status Bytes 0, 1, and 2 (described in Table 6-3, Table 6-4, and Table 6-5). The remaining four bytes comprise hardware diagnostic data derived from the command phase input. Although a program must read and accept this information to complete a floppy disk command, it is of limited value and will not be described.

Table 6-3. Floppy Disk Controller Status Byte 0 (Port 1016H)

Bit	Meaning if Set
0-3	Unused and can be arbitrary.
4	Equipment Check. After RECALIBRATE command, indicates that FDC issued 77 step pulses without reaching Track 0. Disk has 80 tracks; therefore, a second RECALIBRATE command will succeed.
5	SEEK End. Set upon successful termination of SEEK command.
6-7	Termination Flag 00 - Normal termination at beginning of result phase 01 - Abnormal termination during command execution 10 - Last command was invalid 11 - Unused and cannot occur

Table 6-4. Floppy Disk Controller Status Byte 1 (Port 1016H)

Bit	Meaning if Set
0	Missing Address Mark. Set by failure to find ID address mark, data address mark, or deleted data address mark. (When set after ID address mark is found, Missing Address Mark in Data Field flag, Bit 0 of Status Byte 2, is also set.)
1	Not Writable. Set when FDC attempts to write disk data while Write-Protect signal (/WPROT) from drive is active.
2	No Data. During READ DATA command, WRITE DELETED DATA command, or one of the scan commands, indicates that FDC cannot find the designated sector. During READ ID command, indicates that FDC cannot read the header ID field. During READ TRACK command, indicates that FDC cannot find starting sector.
3	Unused and always 0.
4	Overrun. Processor failed to supply data or accept data within allowable time period.
5	Data Error. Indicates presence of a CRC error in either the header ID field or the data field. If the error is in the data field, the Data Error in Data Field flag (Bit 5 of Status Byte 2) also will be set.
6	Unused and always 0.
7	End of Cylinder. Could not find designated sector number on the current track.

Table 6-5. Floppy Disk Controller Status Byte 2 (Port 1016H)

Bit	Meaning if Set
0	Missing Address Mark in Data Field. While reading data, FDC failed to find a data address mark or a deleted data address mark.
1	Bad Cylinder. Set if the track number read from a header field is FFH and the command bytes designated a different track number.
2	Scan Not Satisfied. Set when one of the scan commands fails to find a sector that meets the designated scan condition.
3	Scan Equal Hit. Set when one of the three scan commands finds a sector whose data matches the prototype data supplied by the processor.
4	Wrong Cylinder. Set if the track number read from a header field differs from the track number designated in the command bytes.
5	Data Error in Data Field. Set by a CRC error in the 256-byte data field.
6	Control Mark. Set during Read Data command or one of the Scan commands upon detecting a sector that contains a deleted data address mark.
7	Unused and always 0.

6.4 READ AND WRITE ACCESS

To read or write floppy disk data, software first selects the appropriate disk unit and head, and then it executes a SEEK command if necessary to position the head on the correct track. Once the first track involved in a read or write access has been located, software can initiate the transfer by issuing one of the read or write commands. Execution phase follows command phase automatically when the FDC receives a valid command sequence. During execution of a read or write command, the FDC transfers data to or from the disk across DMA Channel 2. A floppy disk command does not explicitly specify the amount of data to be transferred. Instead, software relies on one of four mechanisms for terminating a data transfer:

- End-of-Operation (/EOP) signal from the DMA controller, which indicates that the proper number of bytes has been transferred.

- Terminal Count (/T/C), generated when a program reads or writes either of I/O Ports 101C-EH, ends an operation immediately. /T/C is the easiest way to end a transfer and the only way to terminate an operation in the middle of a sector.

- The index pulse that signals the end of the current track. An unexpected index pulse will cause an abnormal termination.

- Reading the sector designated as the last sector in the seventh byte of the standard command sequence (Sector 16, 8, or 4, using sector sizes of 256, 512, and 1024 bytes, respectively). Reading any higher numbered sector has the same effect. This mechanism for ending a command also causes an abnormal termination and sets the End of Cylinder flag (Bit 7 of Status Byte 1).

Two mechanisms for terminating a command cause normal termination and two cause abnormal termination. The abnormal termination mechanisms return an error code to indicate that the last sector or end of track was encountered unexpectedly.

6.5 FLOPPY DISK CONTROLLER INTERRUPTS

FDC Level 2 interrupt requests occur at the beginning of the result phase for every FDC command except SEEK and RECALIBRATE (which have no result phase), SENSE INTERRUPT STATUS, SENSE DRIVE STATUS, SPECIFY, and an invalid command. Reading the first byte of result phase data automatically clears the Level 2 FDC interrupt request. In the case of a SEEK or RECALIBRATE command, which, in theory, can be overlapped with another command, the interrupt occurs upon completion of command execution. A SENSE INTERRUPT STATUS command must follow each SEEK or RECALIBRATE operation. After a SEEK, the SENSE INTERRUPT STATUS returns the actual track on which the heads are positioned.

The FDC also generates a Level 2 interrupt request when the door on either drive unit is opened. So-called "door-disturbed" interrupts can be distinguished from the FDC interrupt request that terminates execution phase by examining Bits 3, 4, and 5 of System Status Port 10E0H. Bit 3 of this port will be set to 1 only during the interrupt request that terminates execution phase of an FDC command. Bit 4 will be set to 1 only when Drive 1 generates a door-disturbed interrupt request, and Bit 5 will be set only when the door on Drive 2 has been disturbed. Writing output Port 1000H with Bits 2 and 3 set to 1 clears the door-disturb interrupt request from floppy disk Drives 1 and 2, respectively. Either type of floppy disk interrupt also sets Bit 6 of Interrupt Status Port 1022H, and this bit can be examined to distinguish floppy disk interrupt requests from other Level 2 interrupt requests.

6.6 MULTIPLE TRACK COMMANDS

Only a SEEK or RECALIBRATE command can move a disk read/write head from one track to another. Consequently, FDC commands to read or write data usually operate on a byte-by-byte or sector-by-sector basis within a particular track. Command phase data specifies a starting sector within the track on which the head is positioned. The first byte of this sector is the first byte written or read. Successive bytes are transferred, moving from one sector to the beginning of the next, until one of the four conditions described earlier terminates the command.

Certain read and write commands can operate on either a per-track or per-cylinder basis, where a cylinder consists of two corresponding tracks with identical track numbers on opposite sides of a disk. The commands that allow multiple track operation are READ DATA, READ DELETED DATA, WRITE DATA, WRITE DELETED DATA, and all three SCAN commands.

Multiple-track operation is requested by setting high-order Bit 7 of the command code (i.e., adding A0H to the value of the first command byte). Then, instead of terminating at the end of the current track, the transfer continues at the beginning of the first sector on the other side of the disk and terminates at the end of the cylinder. The two tracks in a cylinder can be read or written in either order. /EOP and /T/C will terminate a multiple-track transfer immediately, just as for single-track operations.

6.7 SKIP OPTION

The uPD765 chip supports two types of sectors distinguished by two different address marks. Normal sectors have normal address marks and "deleted data" sectors have unique "deleted data" address marks. If a command that expects normal data encounters deleted data, it terminates abnormally. This also occurs if a command that expects deleted data encounters normal data. The Skip option changes command execution to ignore whichever type of sector the command does not expect. Commands that allow the Skip option are READ DATA, READ DELETED DATA, WRITE DATA, WRITE DELETED DATA, READ TRACK, and all three SCAN commands.

The Skip option is requested by setting Bit 5 of the first command byte (i.e., adding 40H to the command code). Then, instead of returning an error when it encounters a sector containing the wrong type of data, the command will skip over that sector and resume normal execution at the beginning of the next sector.

6.8 FLOPPY DISK CONTROLLER COMMAND SUMMARY

A description of the floppy disk controller codes and the operations they perform follows.

42H -- READ TRACK

Uses standard 9-byte command format. Performs the same function as READ DATA but does not terminate abnormally, even if an error occurs, until the end of the track (or cylinder) is reached. Unlike READ DATA, READ TRACK returns the entire content of the track including gap bytes, address marks, and other formatting information stored on the disk, as well as data from the data field. Helps recover data when a CRC error or other condition prevents normal READ DATA access.

Uses standard 7-byte result format. Sets the No Data flag (Bit 2 of Status Byte 1) and terminates abnormally if the designated starting sector cannot be found. Flags CRC errors in header or data fields by setting Bit 5 of Status Byte 1, but continues normal execution and does not report these errors until result phase. Sets the Missing Address flag (Bit 0 of Status Byte 1) and terminates abnormally if there is no address mark on the track.

03H -- SPECIFY

First command byte contains only the command code 3. Second command byte designates head unload time in Bits 0-3 and step rate time in Bits 4-7. Head unload time must be nonzero and ranges from 32 to 480 milliseconds in increments of 32 milliseconds (1=32 milliseconds, 2=64 milliseconds, ..., F=480 milliseconds). Step rate time is interval between consecutive step pulses. It ranges from 32 to 2 milliseconds in 2-milliseconds decrements and is coded as follows: 0=32 milliseconds, 1=30 milliseconds, 2=28 milliseconds, ..., F=2 milliseconds. Third and final command byte has Bit 0 cleared and designates head load time in Bits 1-7. Head load time must not be 0 and ranges from 4 to 508 milliseconds in 4-milliseconds increments. Recommended value for second command byte is DFH. Recommended value for third command byte is 08. The SPECIFY command does not return any data during result phase.

04 -- SENSE DRIVE STATUS

First command byte contains only Command Code 4. Second (and final) command byte is arbitrary and can be the head select value (0 or 4) used as the second command byte for other commands. Result phase returns one byte formatted as follows:

Bit	Meaning if Set
0-3	Unused and can be arbitrary.
4	Track 0 flag. Set if heads on selected drive are positioned at Track 0.
5	Unused and always set to 1.
6	Write-Protect flag. Set if Write-Protect signal (/WPROT) from selected drive is active.
7	Unused and always 0.

45H -- WRITE DATA

Uses standard 9-byte command format. Reads disk ID fields to locate the starting sector designated in the fifth command byte, then accepts data from DMA Channel 2 and writes it to the disk. Moves from the end of one sector to the beginning of the following sector. Terminates upon reaching the end of the track or cylinder, upon receiving a DMA End-of-Process (/EOP) signal, or when software accesses I/O Port 101C-EH to force terminal count. Pads any unspecified bytes at the end of the last sector with zeros if /EOP or terminal count occurs in the middle of a sector.

Uses standard 7-byte result format. Sets the No Data flag (Bit 2 of Status Byte 1) if the starting sector cannot be found. Sets the Data Error flag (Bit 5 of Status Byte 1) on encountering a header field CRC error. Sets both the Data Error flag and the Data Error in Data Field flag (Bit 5 of Status Byte 2) on encountering a data field CRC error. Any of these errors will terminate the command abnormally.

46H -- READ DATA

Uses standard 9-byte command format. Reads ID fields to locate the starting sector designated in the fifth command byte, then reads data from disk and transfers it across DMA Channel 2. Moves from the end of one sector to the beginning of the following sector. Terminates upon reaching the end of the track or cylinder, upon receiving a DMA End-of-Process (/EOP) signal, or when software accesses I/O Port 101C-EH to force terminal count.

Uses standard 7-byte result format. Sets the No Data flag (Bit 2 of Status Byte 1) if the starting sector cannot be found. Sets the Data Error flag (Bit 5 of Status Byte 1) on encountering a header read error and also sets the Data Error in Data Field flag (Bit 5 of Status Byte 2) if the error was a header CRC error. Any of these errors will terminate the command abnormally.

Upon reading a "deleted data" address mark when the Skip flag (Bit 5 of the first command byte) was 0, set the Control Mark flag (Bit 6 of Status Byte 2) and terminate normally after reading the sector. If the Skip flag was set, skips the "deleted data" sector and reads the next sector instead.

07 -- RECALIBRATE

First command byte contains only Command Code 7. Second (and final) command byte contains 0. Steps heads outward on the selected drive until Track 0 is reached and terminates with the heads on Track 0. The RECALIBRATE command does not have a result phase; instead, it generates a Level 2 interrupt request upon terminating, and software then executes a SENSE INTERRUPT STATUS command to retrieve the RECALIBRATE status. Disk has 80 tracks, but uPD765 FDC chip expects only 77 tracks; therefore, two consecutive RECALIBRATE commands must be issued to ensure reaching Track 0. Any error returned by first RECALIBRATE command can be ignored.

08 -- SENSE INTERRUPT STATUS

Uses a special 1-byte command format consisting of only Command Code 8. Returns two bytes during result phase: Status Byte 0, followed by the location (current track number) of the read/write head for the drive unit currently selected.

49H -- WRITE DELETED DATA

Identical to the WRITE DATA command except that a special "deleted data" address mark replaces the normal address mark at the beginning of each data field.

4AH -- READ ID FIELD

Uses only the first two bytes of the standard 9-byte command format. Returns all data from the first ID field encountered on the current track, using standard 7-byte result format. Sets the Missing Address Mark flag (Bit D0 in Status Byte 1) if there is no address mark on the track. Sets the No Data flag (Bit D2 in Status Byte 2) if no data is found. Either of these conditions terminates the command abnormally.

4CH -- READ DELETED DATA

Identical to the READ DATA command except that the roles of "normal" and "deleted data" sectors are reversed. When the FDC detects a normal data address mark, it reads all data in the sector, sets the Control Mark flag (Bit 6 of Status Byte 2), and terminates the command abnormally. However, if the "skip" bit of the first command word (Bit 5) is set and a normal data address mark is found, it skips this sector and reads the next sector instead.

4DH -- FORMAT TRACK

Uses a special 6-byte command format in which the first command byte contains 4D and the second command contains 0 to select side 0 or 4 to select Side 1. Third, fourth, and fifth command bytes contain the following decimal values:

1, 16, 50	(for 16 sectors/track and 256 bytes/sector)
2, 8, 80	(for 8 sectors/track and 512 bytes/sector)
3, 4, 120	(for 4 sectors/track and 1024 bytes/sector)

Sixth (and final) command byte contains initial value of all data field bytes. Formats an entire track by writing gaps, address marks, ID fields, and data fields in the selected sector size.

During execution phase, software supplies four bytes of data for use in the ID field of every sector: the track number in the first byte, the side number (0 or 1) in the second byte, the sector number in the third byte, and a fourth byte containing the same value supplied in the third command byte (i.e., either 1, 2, or 3, depending on sector size).

Uses the standard 7-byte result format; however, the four bytes of disk address data have no meaning.

0FH -- SEEK

Uses a special 3-byte command format in which first command byte contains F. Second command byte contains 0 to select Side 0 or 4 to select Side 1. Third byte contains target track number. Establishes the direction to move and issues step pulses, at the stepping rate established by the last SPECIFY command, until the head reaches its new track position.

The SEEK command does not have a result phase; instead, it generates a Level 2 interrupt request upon terminating, and software then executes a SENSE INTERRUPT STATUS command, which returns Status Byte 0 and the current track position. A successful SEEK sets the Seek End flag (Bit D5 in Status Byte 0).

51H -- SCAN EQUAL

Uses standard 9-byte command format, except that ninth command byte contains either 1, to scan every sector or 2 to scan alternate sectors. Accesses the disk as though performing a READ DATA command while accepting bytes from DMA Channel 2 as though performing a WRITE DATA command. Compares each byte from disk with the corresponding byte from the DMA controller. If all pairs of bytes match, and the scan condition is therefore met, sets the Scan Hit flag (Bit D3 of Status Byte 2) and terminates normally. Otherwise repeats the scan on the next sector (or the sector following the next sector) in the track. After scanning the last sector on the track without meeting the scan condition, sets the Scan Not Satisfied flag (Bit D2 of Status Byte 2) and terminates normally. Uses standard 7-byte result format in either case.

5BH -- SCAN LOW OR EQUAL

Identical to SCAN EQUAL, except that the scan condition is "less than or equal to" under unsigned, 1's complement arithmetic.

5DH -- SCAN HIGH OR EQUAL

Identical to SCAN EQUAL, except that the scan condition is "greater than or equal to" under unsigned, 1's complement arithmetic.

6.9 FLOPPY DISK RECORDING FORMAT

Table 6-6 and Figure 6-1 describe and illustrate the floppy disk recording format.

Table 6-6. Floppy Disk Recording Format

Field Name	Size (bytes)	Data Value	Description
GAP 4A	80	4EH	Preamble. Marks beginning of track and allows for tunnel erase recovery time.
SYNC	12	00	Synchronization field for data separator.
IAM	3	C2H	Index Address Mark. Identifies the beginning of a track. Has missing clock transition between third and fourth bits of each byte; therefore, clock value is 14.
	1	FCH	Distinguishes Index Address Mark, which is followed by one byte containing FCH, from ID Address Mark, which is followed by one byte containing FEH, and Data Address Mark, which is followed by one byte containing FBH.
GAP 1	50	4EH	Spacing before first sector on track.
SYNC	12	00	Synchronization field for data separator. Beginning of sector. Fields from SYNC through and including Gap 3 are repeated in each sector.
IDAM	3	A1H	ID Address Mark. Identifies the beginning of the sector ID field. Has missing clock transition between fourth and fifth bits of each byte; therefore, clock value is 0AH.
	1	FEH	Distinguishes ID Address Mark from Index Address Mark and Data Address Mark.
ID	4		ID field contains cylinder or track number (0-79), head or side (0 or 1), sector number (1-16), and sector size code (1, 2, or 3 for 256, 512, and 1024 bytes, respectively). Each of these fields is one byte long.
CRC	2		CRC data for previous eight bytes.
GAP 2	22	4EH	Spacing between ID and data fields.
SYNC	12	00	Synchronization field. Allows each sector's data to be independently synchronized.

Table 6-6. Floppy Disk Recording Format (continued)

Field Name	Size (bytes)	Data Value	Description
DAM	3	A1H	Data Address Mark. Has missing clock transitions between fourth and fifth bits of each byte; therefore, clock value is 0AH.
	1	FBH	Distinguishes Data Address Mark from Index Address Mark and ID Address Mark.
DATA			Arbitrary data.
CRC	2		CRC data for data field and four preceding bytes.
GAP 3	80	4EH	Last Field in Sector. Provides spacing to beginning of following sector, which allows for tunnel erase recovery and system processing time.
GAP 4B		4EH	Postamble between end of last sector and end of track. Allows for variation in drive motor speed.

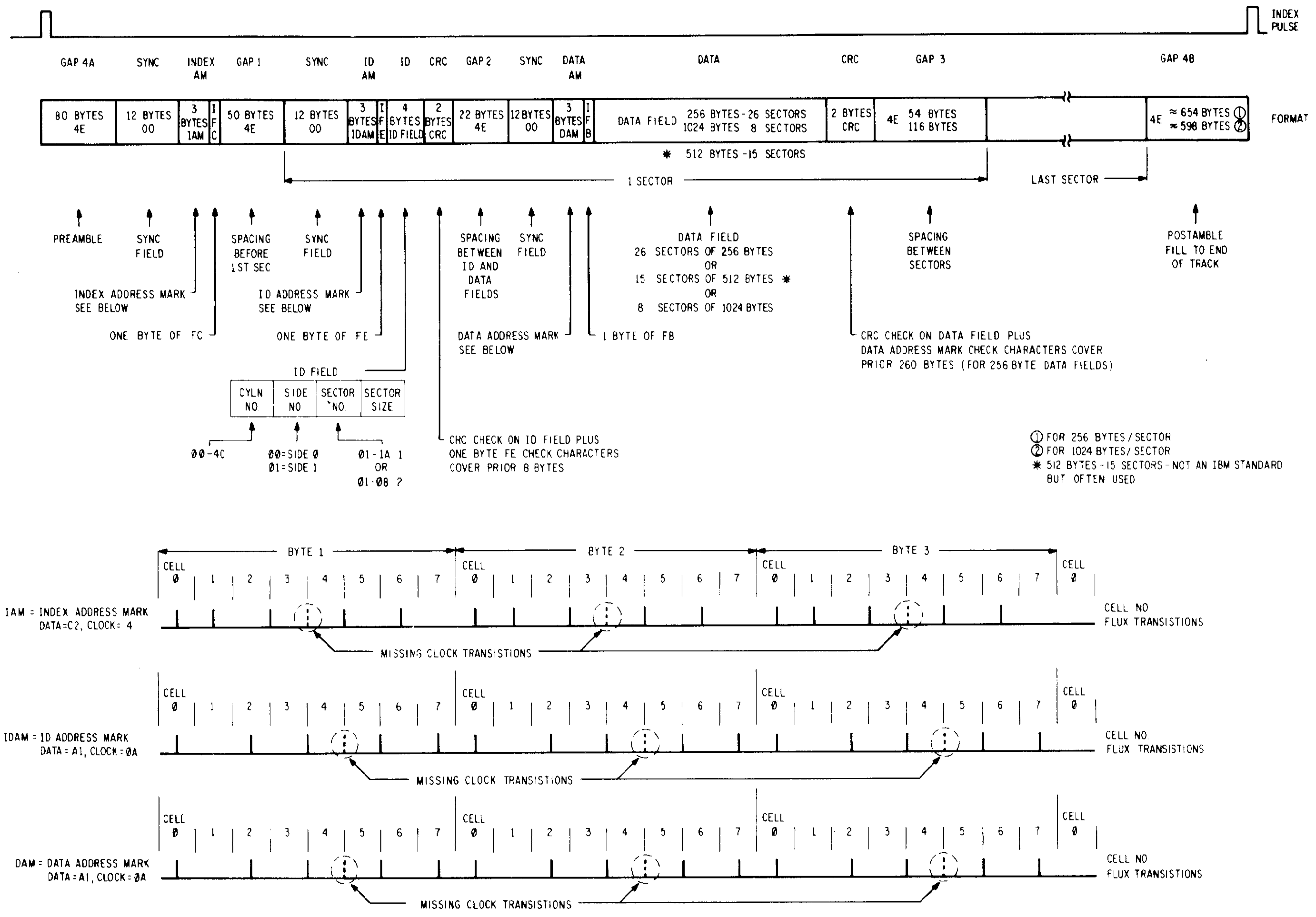


Figure 6-1. Double-Density Recording Format

CHAPTER 7
LOW-RESOLUTION VIDEO CONTROLLER

The Analog low-resolution controller is designed to run either an industry-standard RGB monitor or the Wang PC Color Monitor. The Wang PC low-resolution controller runs either aniIndustry-standard RGB monitor or a broadcast television monitor. Figure 7-1 shows a block diagram of a low-resolution video controller. The MC6845-1 video controller chip is the heart of the video controller, while additional circuitry manages video memory and implements video display options.

All four low-resolution display options use 15.70-kHz horizontal and 60-Hz vertical frequencies. All display 25 lines of text on the screen, but the number of characters in a line can be either 40 or 80, as shown in Table 7-1. Video monitors can display either 40- or 80-column text. A broadcast television monitor must use the 40-column display format. Therefore, broadcast monitors are not suitable for word processing applications.

Table 7-1. Low-Resolution Video Output Options

Output Device	Screen Size	Character Size
Broadcast television for standard low resolution	320 x 225 pixels 40 x 25 characters	8 x 9 pixels (4 bits/pixel)
Video monitor or Wang color monitor	640 x 225 pixels 80 x 25 characters	8 x 9 pixels (2 bits/pixel)

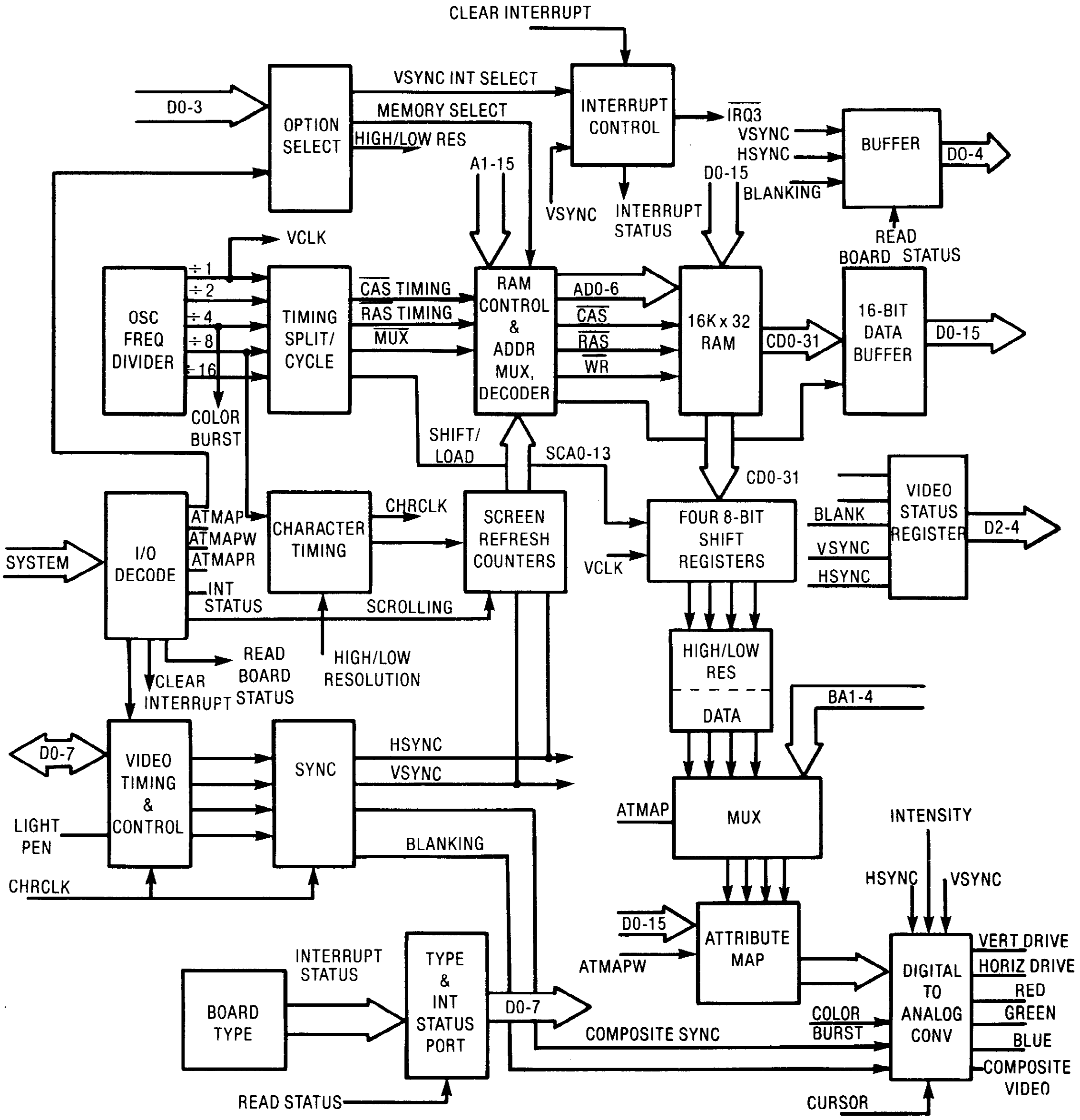


Figure 7-1. Low-Resolution Video Controller Block Diagram

7.1 VIDEO MEMORY ORGANIZATION

Video display screens are divided into 40 or 80 columns. A column is eight pixels wide and runs vertically from the top of the screen to the bottom. A character will generally fit entirely within one column (although this is not a requirement), and certain hardware capabilities, such as horizontal scrolling, operate on a per-column basis. A video monitor has 80 columns that are each eight pixels wide. A broadcast television has 40 columns, each with the same 8-pixel width. Thus, the columns on a broadcast television display are twice as wide as the columns on a video monitor of equal screen size. Broadcast television pixels are also twice as wide as their video monitor counterparts; however, pixels displayed on either device are equal in height. Video monitors can use the 0-column broadcast television format.

The 32K-word, low-resolution video memory space is divided into two planes of bitmap memory that drive an 80-column display, or four memory planes that produce half as many pixels to drive a 40-column display. Video memory must be word addressed; byte addresses cannot be used, and only even addresses are valid. Direct memory access, which handles only byte data, cannot be used to transfer information to or from video memory.

Figures 7-2 and 7-3 show how each byte address maps to a horizontal strip of eight consecutive pixel locations in the video memory space. Figure 7-2 demonstrates video memory addressing for an 80-column display address. For 80-column display, the low-order address bit designates one of two memory planes. The next seven bits select one of 128 columns (only 80 of which are displayed at a time). The eight high-order bits select one of 256 scan lines (only 225 of which are displayed at a time).

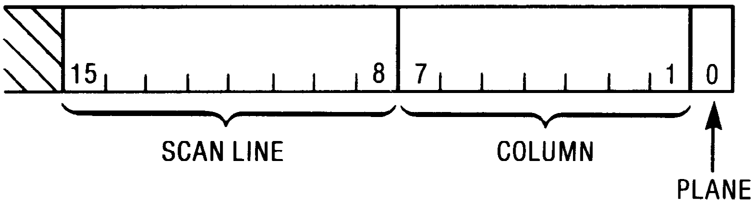


Figure 7-2. 80-Column Display Address

Figure 7-3 demonstrates video memory addressing for a 40-column display address. Twice as many planes of memory hold half as many columns when using 40-column display. Since video memory is word addressed, both planes of 80-column display must be accessed together. In 40-column display, Planes 0 and 1 are accessed together, or Planes 2 and 3 are accessed together.

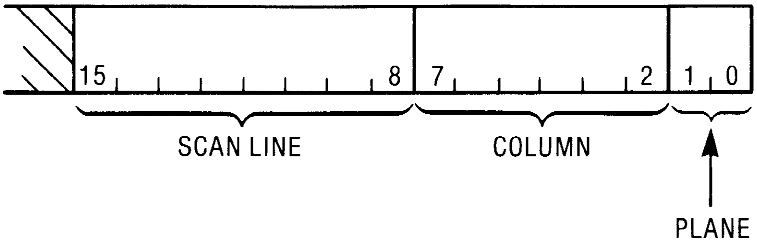


Figure 7-3. 40-Column Display Address

All eight consecutive pixels stored in one byte of video memory occupy the same character column and the same scan line on a display screen. In a sense, each video memory byte address is a set of character column by scan line coordinates. A word of video memory contains the eight pixels for one memory plane in its low-order byte and the eight pixels for another plane in its high-order byte. Both bytes of this word must be accessed together.

Video memory holds 128 columns, each 256 scan lines deep. Because there are only 225 scan lines down a display screen and at most 80 columns across it, only 18,000 words of video memory can be mapped to the screen at one time (in the case of an 80-column display). The remaining 14,768 words of the 32K-word video memory do not map to a visible location and are not displayed.

Figure 7-5 shows video memory organization for an 80-column display. The foreground memory plane contains 128 by 256 bytes at even addresses; the background plane contains bytes at odd addresses. The visible region measures 80 by 225 bytes. Any byte in an even-numbered column can be mapped to the upper left corner of the screen. If an address above 1F60H is mapped to the upper left corner of the screen, the display wraps around as shown.

7.2 HORIZONTAL AND VERTICAL SCROLLING

Software identifies the visible portion of video memory by loading the Scroll Register with the encoded column and scan line coordinates of the byte that is to be displayed in the upper left corner of the screen -- the byte containing the eight pixels in Memory Plane 0 that are to be displayed at Column 0 of Scan Line 0 on the screen (refer to Figure 7-4). The Scroll Register is so named because changing its value scrolls the display window horizontally or vertically within video memory. The Scroll Register accepts only the six high-order bits of the 7-bit column number; therefore, the upper left corner of the display always corresponds to a video memory address that divides evenly by four. This means that even-numbered columns in video memory must occupy even-numbered columns on an 80-column display screen.

Figure 7-4 demonstrates video memory addressing for the Scroll Register. The Scroll Register identifies the scan line to appear at top of display (EBH for Line 0, ECH for Line 1, etc.) and the column to appear at left edge of display (0, 1, etc.). An 80-character display must begin with an even-numbered column; therefore, only six high-order bits of a 7-bit column number are needed. The value of EB00H begins the display at the lowest video memory address (Memory Column 0 of Scan Line 0).

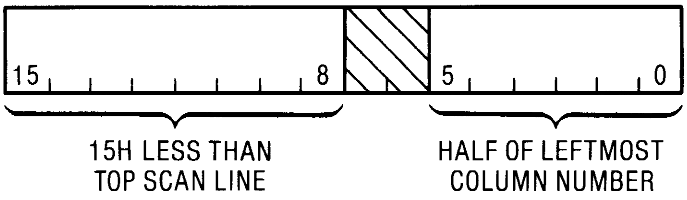


Figure 7-4. Scroll Register

A Scroll Register value of EB00H positions the display window at the lowest video memory address, as shown in Figure 7-5(a). Incrementing the Scroll Register content by one scrolls the display window two character columns to the right (i.e., the displayed text seems to shift two character columns to the left). After 24 such increments, on an 80-column display, the window has scrolled 48 columns to the right and is positioned as shown in Figure 7-5(b). Incrementing the Scroll Register beyond EB18H again shifts the display window two columns to the right, but now it wraps around in video memory so that the rightmost display column contains characters that were originally on the leftmost edge of the display.

If the Scroll Register is again initialized to EB00H, as shown in Figure\7-5(a), increasing the Scroll Register content by 100H now scrolls the display window one scan line downward (i.e., the displayed text seems to shift one scan line upward). After nine such increases, the Scroll Register contains F400H and the window has scrolled downward across one row of characters. Video memory accommodates 256 scan lines, only 225 of which are displayed on the screen; therefore, after 100H has been added to the original Scroll Register content a total of 31 times, the Scroll Register contains 0A00H, and the display window is positioned as shown in Figure 7-5(c). Increasing the Scroll Register beyond 0A00H again shifts the display window downward, but now it wraps around in video memory so that the bottom scan line on the display contains pixels that were originally on the top edge of the display.

Figure 7-5(d) shows the effect of scrolling the display window 24 character columns to the right and 31 scan lines downward, effectively combining both examples just mentioned. This diagram corresponds to a Scroll Register value of 0A18H. Figures 7-5(e) and 7-5(f) show the general case where a display window is positioned at some arbitrary point in video memory, leaving buffer space available to append data either horizontally or vertically.

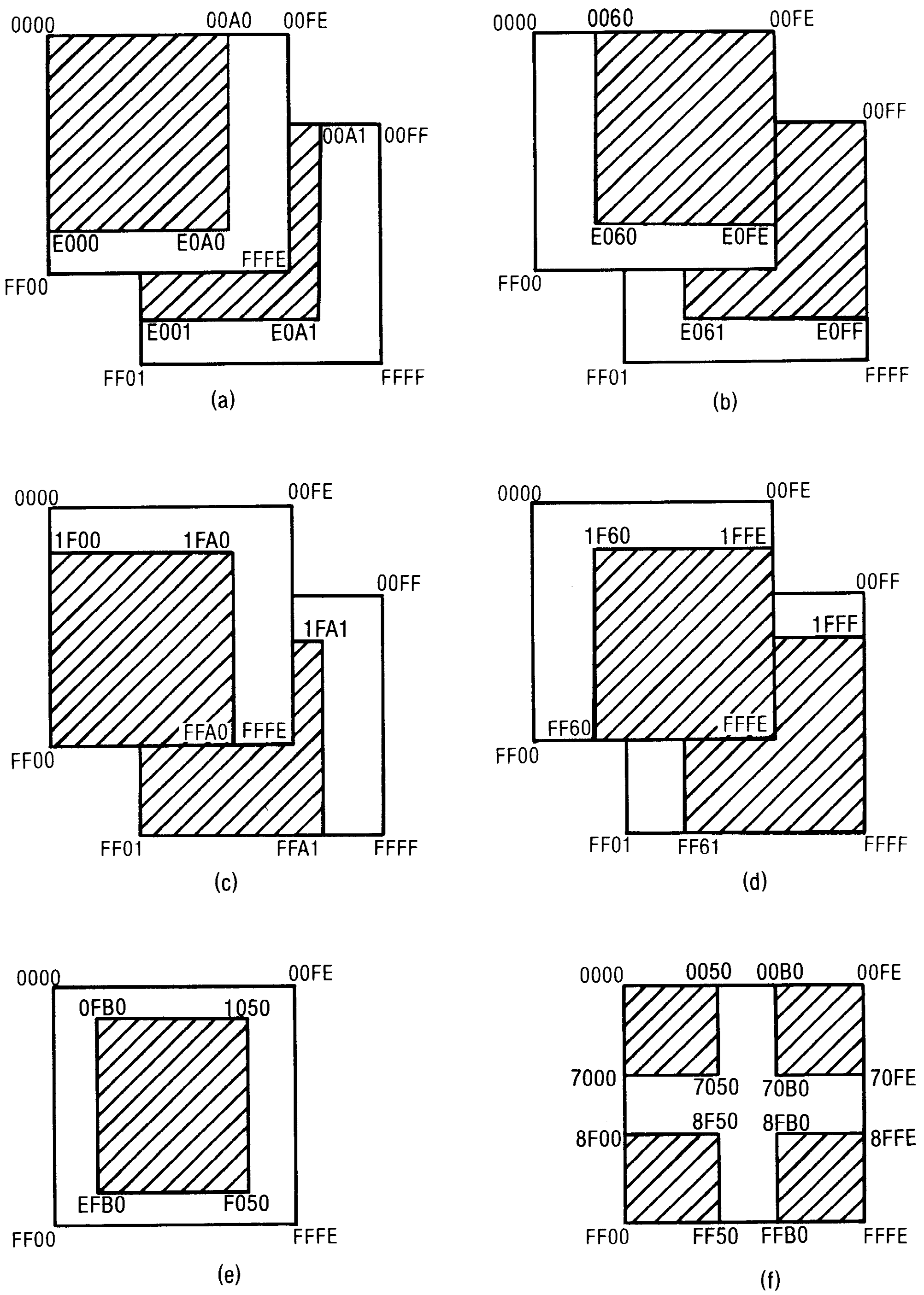


Figure 7-5. Video Memory Organization for 80-Column Display

7.3 VIDEO DISPLAY ATTRIBUTES

A lookup table or "attribute map" changes the two bits that form each 80-column display pixel (or the four bits comprising each 40-column display pixel) into a 4-bit color attribute. To paint a pixel on the screen, hardware first assembles the two or four pixel bits, which will have a value (N) between 0 and 15, inclusive. The circuitry then reads the Nth entry in the lookup table and uses this as the color attribute value.

The 4-bit color attribute in a lookup table entry has Bit 3 set if the blue color gun is on, Bit 7 set if the green color gun is on, Bit 11 set if the red color gun is on, and Bit 15 set to designate high intensity (instead of normal intensity) for all three color guns. These four color code bits implement a pallette of 16 different colors:

0000	Black	8000H	Gray
0008	Blue	8008H	Bright blue
0080H	Green	8080H	Bright green
0088H	Cyan	8088H	Bright cyan
0800H	Red	8800H	Bright red
0808H	Magenta	8808H	Bright magenta
0880H	Brown	8880H	Yellow
0888H	White	8888H	Bright white

On a 40-column display, with four planes of video memory supplying four bits of lookup table input, all 16 of these colors can be displayed simultaneously. On an 80-column display, with only two bits per pixel, only a selection of four different colors can be displayed at a time.

When a monochrome broadcast television or monitor is used for 40-column output, the four bits in an attribute map entry combine to select one of 16 levels of gray. The 16 gray levels span the black-and-white spectrum from complete black to intense white but not in strictly ascending sequence by numerical value. Operating in the 80-column display format, a black-and-white video monitor will display any four of the 16 gray levels that were available in the low-resolution format but with twice as many pixels in the horizontal direction.

A black-and-white video monitor can operate in the broadcast television mode to generate all 16 levels of gray at lower resolution if desired. Program interaction is basically the same in every case: 16 output ports serve to enter lookup table data, and normal memory reference instructions access words of video memory.

Certain RGB monitors will not generate background colors when used with the low-resolution 9222/8222 card. This restriction does not apply to the Analog low-resolution controller.

7.4 PROGRAMMING THE VIDEO CONTROLLER

Software communicates with the low-resolution video controller by means of the I/O ports described in Table 7-2. The table expresses all port addresses as offsets from the slot address at which the board is installed. Video controller I/O ports are eight bits wide except for the Scroll Register, which is a 16-bit port. Low-order address Bit A0 is not decoded; therefore, all video controller ports have even addresses. Bit 0 of I/O Port 1x10H is a video Memory Enable flag that must be set to 1 to activate the Video controller board. This bit allows installation of multiple video controllers, only one of which will be active at any time.

Table 7-3 further describes internal registers in the MC6845 controller chip, accessed by means of Controller Address and Data Register Ports 1x00H and 1x02H, respectively. This table includes required values, in decimal, for the data that must be loaded into certain controller chip registers. All controller chip registers accept 8-bit data unless otherwise indicated.

Table 7-2. Low-Resolution Video Controller I/O Ports

Port	Description
1x00H	Write Controller Address Register. Bits 0-4 address one of the 18 controller registers listed in Table 7-3. A subsequent write to the Controller Data Register, then loads this controller register.
1x02H	Write Controller Data Register. Particular controller register addressed by Controller Address Register receives up to eight bits written to this port.
1x10H	Write Option Register. Specific bits enable or disable particular options: D0 -- Video memory is accessible by 8086 when set to 1. If cleared to 0, 8086 cannot access video memory. Allows use of another video controller while this controller is installed but inactive. D2 -- Display 80 columns when set to 1. Display 40 columns when cleared to 0. D3 -- Vertical Sync (/VSYNC) generates a Level 3 interrupt request at the beginning of the vertical blanking interval when set to 1. The power-up (default) value is 0, or all options disabled.

Table 7-2. Low-Resolution Video Controller I/O Ports (continued)

Port	Description
1x20H	Write Scroll Register (16 bits). Receives the 16-bit, encoded, character and scan line coordinates of the video memory byte to appear as Plane 0 in the upper left corner of the display screen (Column 0 and Scan Line 0). Bits 15 to 8 receive scan line minus 15H. Bits 7 and 6 are not used. Bits 5 to 0 receive column number divided by 2. The Scroll Register must be word-addressed.
1x30H	Read Video Status Register. Bit assignments are as follows: D0-1 -- Unassigned. D2 -- Display is blanked when set to 1. During the vertical blanking interval, a program can change the lookup table without causing flicker. D3 -- Vertical Sync (/VSYNC) is active when set to 1. D4 -- Horizontal Sync (/HSYNC) is active when set to 1. D5-7 -- Unassigned.
1x40-5EH	Write video attribute map entry. Lookup table contains sixteen 4-bit values at I/O Port Addresses 40H, 42H, ..., 5EH. Only the first four of these are active in the 80-column display mode (40H, 42H, 44H, and 46H).
1x70H	Write arbitrary data to clear the pending interrupt flag (Bit 7 of Port 1xFEH).
1xFEH	Read Option Board ID and interrupt status. Returns 10H on D0-6, with D7 high if the board has a pending interrupt request. The only possible interrupt request occurs at the start of a vertical sync interval when Bit 3 at I/O Port 1x10H is set.

Table 7-3. Low-Resolution Video Controller Chip Internal Registers

Register	Contents		
Address	40-Column	80-Column	Register Description
0	38H	71H	Horizontal Total Register. Determines horizontal frequency. Contains one less than sum of displayed character time units plus non-displayed (retrace) character time units.
1	28H	50H	Horizontal Displayed Register. Designates number of characters on a display line.
2	2EH	5CH	Horizontal Sync Position Register. Determines horizontal position of display on screen. Increasing this value shifts the display to the left.
3	74H	78H	Sync Width Register. Bits 0-3 designate HSYNC width in units of character clock pulses. Bits 4-7 designate VSYNC width in units of raster periods.
4	1CH	1CH	Vertical Total Register (7 bits). Contains one less than number of scan lines per vertical refresh cycle.
5	1	1	Vertical Total Adjust Register (5 bits). Establishes the number of extra scan lines (forming a partial character line at the bottom of the screen) needed to complete a vertical refresh cycle.
6	19H	19H	Vertical Displayed Register (7 bits). Designates number of character lines displayed on the screen.
7	1AH	1AH	Vertical Sync Position Register (7 bits). Determines vertical position of display on screen. Increasing this value shifts the display upward.
8	0	0	Interlace and Skew Register (2 bits). Value of 0 requests non-interlaced display without skew.
9	8	8	Maximum Scan Line Address Register (5 bits). One less than number of scan lines in a character cell.

Table 7-3. Low-Resolution Video Controller Chip Internal Registers (continued)

Register	Contents		
Address	40-Column	80-Column	Register Description
AH	67H	67H	Cursor Start Register (7 bits). Regulates cursor format. Bits 0-4 contain the number of the highest scan line in the cursor, counting down from Scan Line 0 at the top of the character cell. Bits 5-6 contain 00 - to display cursor without blinking 01 - to supress cursor 10 - to blink cursor at high frequency 11 - to blink cursor at low frequency
BH	8	8	Cursor End Register (5 bits). Contains the number of the lowest scan line in the cursor, counting down from Scan Line 0 at the top of the character cell.
CH DH	0	0	High and Low Start Address Registers (6 and 8 bits, respectively). Locates first character position that cursor can occupy.
EH FH			High and Low Cursor Registers (6 and 8 bits, respectively). Number of characters preceding cursor on display screen (i.e., 0-998 or 0-1998, depending on resolution). Cursor position does not depend on displayed data; therefore, cursor remains stationary on screen during scrolling.
10-11H			Unused.

7.5 VIDEO MONITOR INTERFACE

Commercially available RGB video monitors use a variety of different cables and connectors, but all require similar interface signals. If the monitor has a SYNC polarity selector, it must be set for positive polarity. At the low-resolution video option board, a 9-pin, D-type connector supplies seven signals required to drive the RGB video monitor. This D-type connector has the following pin assignments:

1	Ground	6	Intensity
2	Ground	7	Unused
3	Red output	8	Horizontal sync
4	Green output	9	Vertical sync
5	Blue output		

CHAPTER 8
MEDIUM-RESOLUTION VIDEO CONTROLLER

The medium-resolution video controller produces Transistor Transistor Logic (TTL) output to drive a monochrome video monitor. Running at 18.824 kHz horizontally and 60 Hz vertically, the controller has 800- by 300-pixel resolution when displaying either text characters or bitmapped dot graphics. The medium-resolution video controller is implemented on two circuit boards: a video character board, which can operate alone, and an optional bitmapped graphics board, which must be paired with a character board.

Figure 8-1 shows a block diagram of the medium-resolution video controller circuitry. The character board (at top of figure) implements most character display functions. The optional graphics board (at bottom of figure) adds bitmapped graphics capability. Ribbon cable connects these boards. Internal address and data buses on an optional board attach to system address and data buses. (Refer to Figure 8-2, which shows attribute logic, for a detailed view of the top right of Figure 8-1.)

Partitioned into a standard matrix containing 25 rows of 80 columns each, the medium-resolution display positions characters or arbitrary symbols, defined by software, within a 10- by 12-pixel character cell. A cursor can be positioned over any character cell. Although cursor size and shape are programmable, the cursor normally occupies all 20 pixels in the lower two rows of a character cell. The cursor can be programmed to blink at either of two repetition rates. Between blink cycles, during the cursor's off time, the overlaid portion of the character appears just as though no cursor were present. The cursor can occupy only one character cell at a time.

A programmable underscore occupies all 10 pixels in the bottom row of a character cell, when designated by software, and a programmable overscore fills all 10 pixels in the top row. Software can assign underscore and overscore attributes to any combination of characters in a display. When adjacent characters are underscored, the underscore portions of their displays form a continuous horizontal line under all of the underscored characters. (This also occurs with adjacent overscored characters.)

Hardware subscript and superscript capabilities modify the normal appearance of a symbol or character. As with underscore and overscore, multiple characters can be subscripted or superscripted in any combination. Subscripting a character rolls the character one scan line downward within its character cell. Superscripting rolls the character cell display two scan lines upward. Wraparound does not occur. Instead, if a character normally has pixels set in the bottom row of its cell, those pixels roll off the bottom of the cell and disappear when the character is subscripted; if a character has pixels defined in the top two rows of its display, those pixels disappear when the character is superscripted.

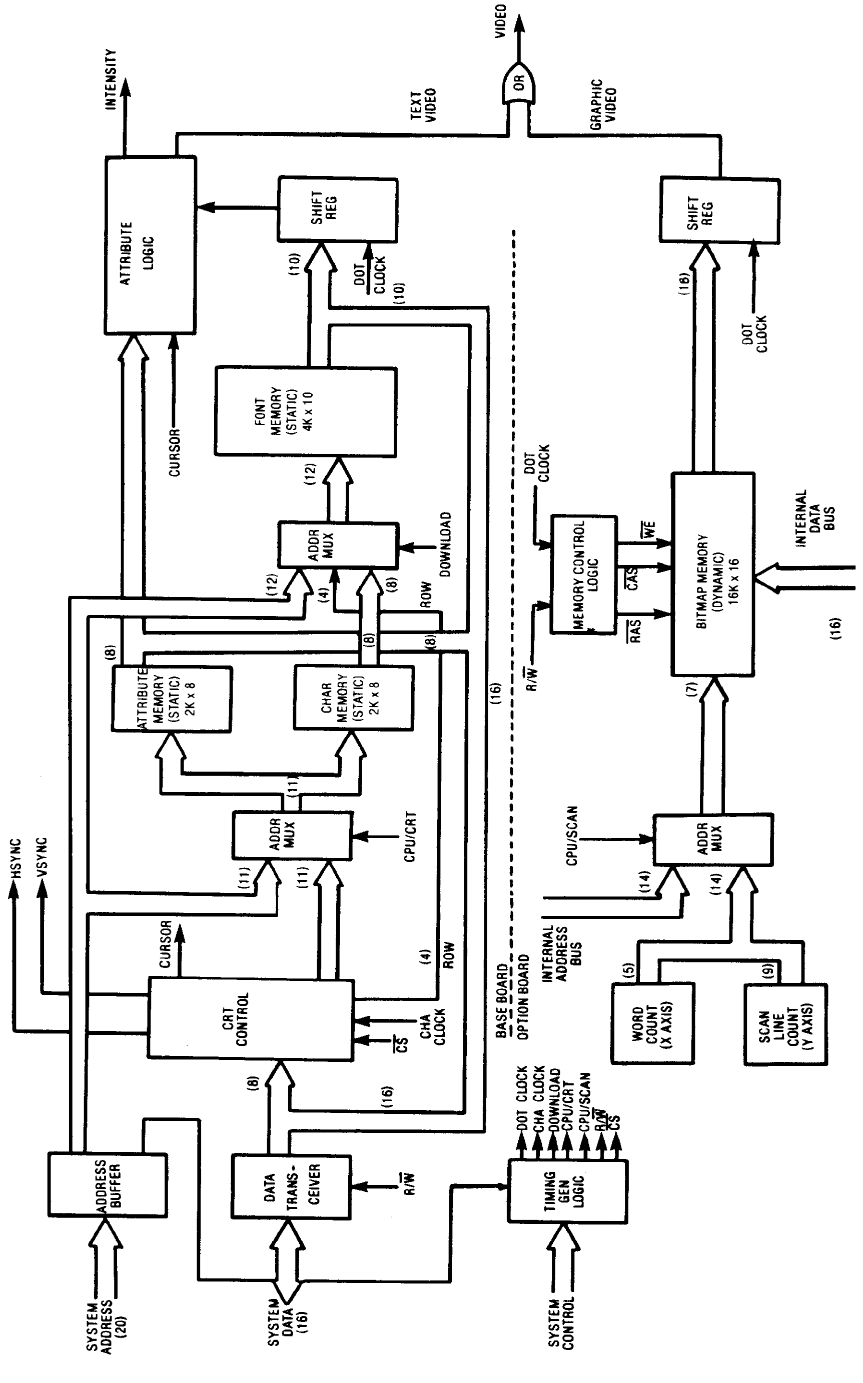


Figure 8-1. Medium-Resolution Video Controller Block Diagram

8.1 FONT TABLE

A font table associates every possible character code (byte) with the arbitrary character cell display pattern that is assigned to the character code. The 4K by 10-bit font table is downloaded from the 8086 to define up to 256 different characters or symbols. It establishes each of the 120 pixels in a 10- by 12-pixel character cell to support true lowercase descenders, accent marks, foreign language fonts, and arbitrary graphic symbols. As with all other memory on the medium-resolution video controller, font table locations must be word-addressed at even memory locations.

The 8086 can update any part of the font table at any time, allowing flicker-free operation without constraints on font table access, and it can read the font table as well as write it. The font table is mapped to Addresses F2000-4000H in the video memory address space. The font table is word-addressed, and it contains 256 16-word entries. Its Nth entry defines the shape of the character with character code N. The word at the lowest address of a font table entry defines the top scan line in a subscripted character display. Successively higher addresses determine successively lower scan lines until the fifteenth word in an entry, which defines the bottom line in a superscripted character display. The sixteenth word at the highest font table entry address is not used.

Figure 8-2 shows the font table address format used to establish or change the symbol assigned to a character code. The figure shows the 800- by 300-pixel bitmap, the frame buffer that holds a full screen of character text, and the font table that defines character shape and appearance. Bitmap memory (at right of figure) contains 328 rows of 50 words each, but the visible portion occupies only the first 300 rows. In the font table (left side of figure), 15 words of 10 bits each define the shape of the character. At any time, only 12 words of 10 pixels each appear in a 10- by 12-pixel character cell. All addresses are hexadecimal word addresses.

Because subscripting shifts a character one scan line downward and superscripting shifts it two scan lines upward, a character actually consists of 15 pixel rows, only 12 of which are visible within its 12-pixel high character cell "window." For example, the character cell display for a subscripted character shows the top 12 rows of that character's complete font table definition. Canceling the subscript produces a normal character display; the character appears to roll upward by one scan line within its character cell window, raising the top row of pixels out of view and shifting in a thirteenth row at the bottom. Superscripting this same character shifts two more rows of pixels upward and out of the character cell window while shifting two new rows -- the fourteenth and fifteenth -- in from the bottom. At any time, only 12 of the 15 pixel rows in a font table entry are visible on the display. The middle nine rows are always visible, but subscripting and superscripting shifts them up and down between the top and the bottom of the character cell window.

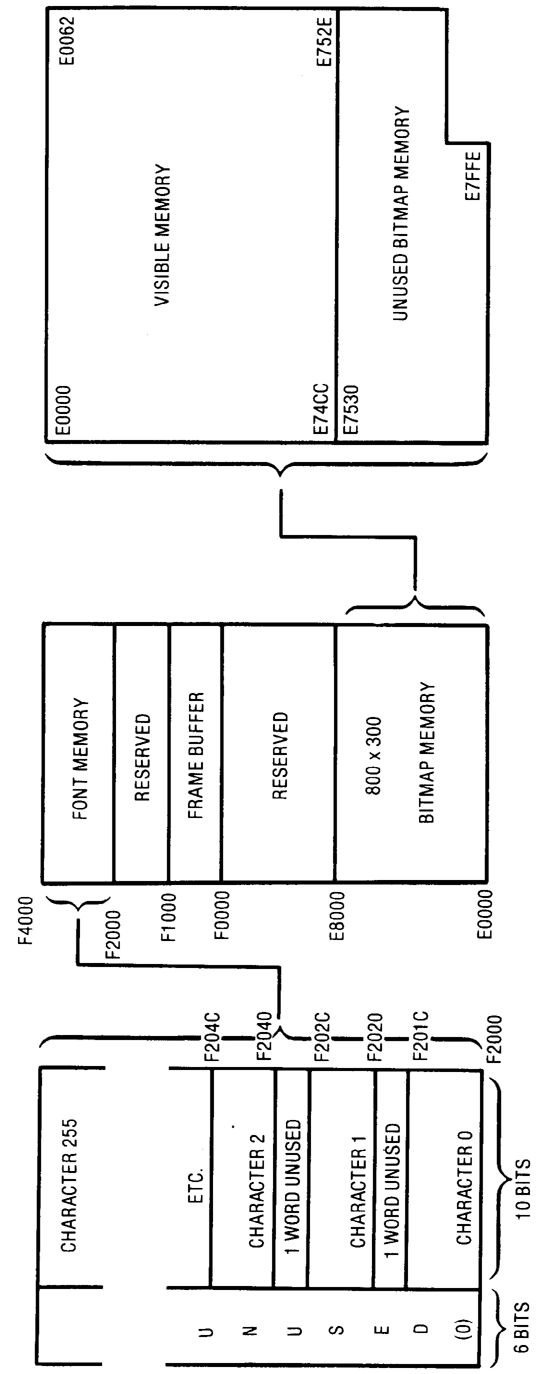


Figure 8-2. Video Memory Organization

Figure 8-3 illustrates character definition. The font table entry defines character shape by supplying a 10- by 15-pixel model that the video controller copies into the 10- by 12-pixel character cell. Character shape is completely arbitrary. Subscript and superscript attributes determine how the font table entry is mapped into the character cell.

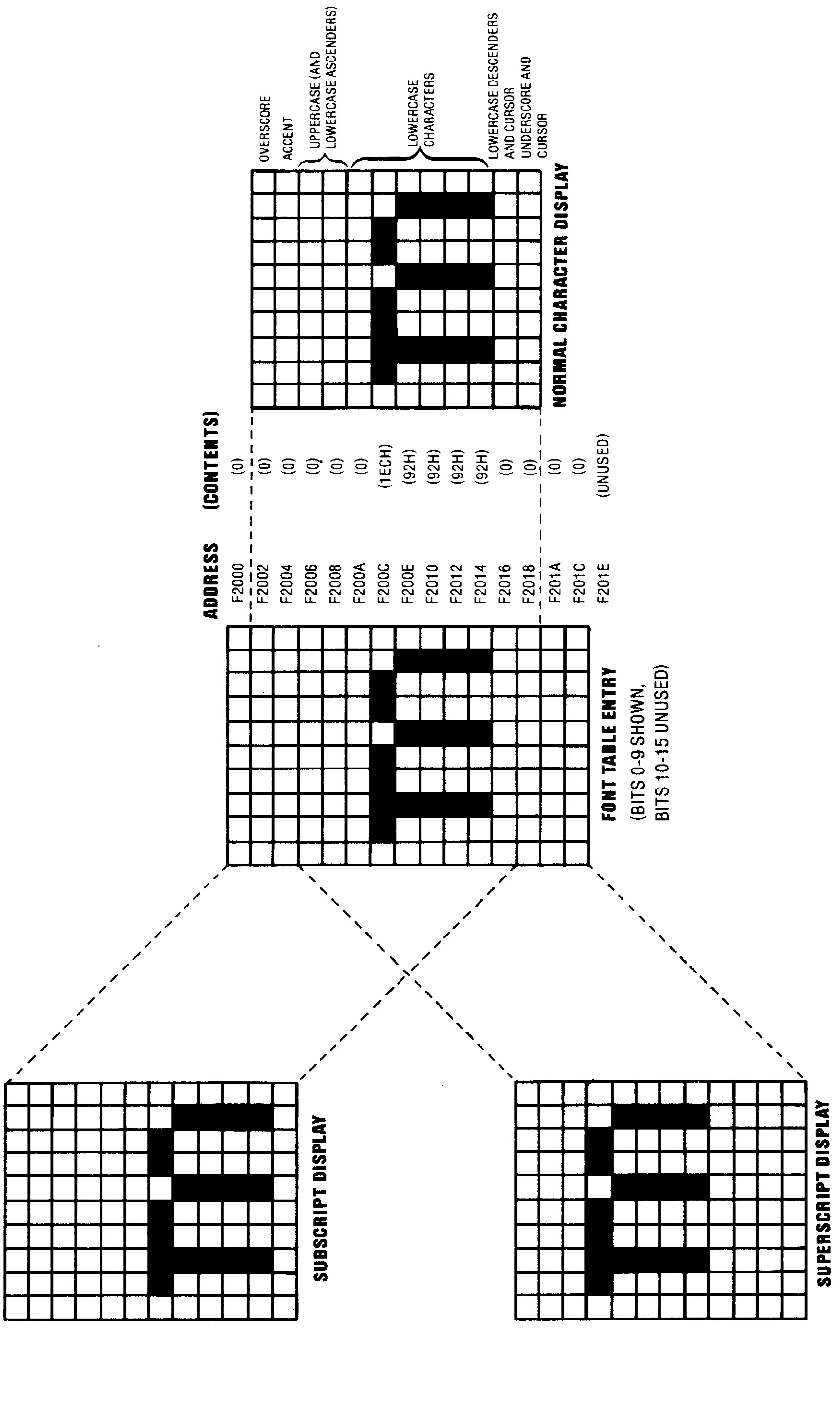
Within its font table entry, a character is defined upside down and in its subscripted position. That is, the first (lowest) row of the 16-row font table entry defines the horizontal strip of pixels that will appear at the top of the character cell when the character is subscripted. The second row of the entry defines the row of pixels that will appear at the top when the character is displayed normally. The fourth row defines the row that will be on top of a superscripted display. The twelfth row will occupy the bottom of a subscripted display. Normal display rolls the thirteenth row of the font table entry up to the bottom of the character cell window, while superscripting shifts in the fourteenth and fifteenth rows. The sixteenth row of each font table entry is not used and cannot be displayed. Pixels defined by high-order bits of each word in a font table entry are displayed to the left of pixels defined by low-order bits.

8.2 FRAME BUFFER

Up to 2000 characters of visible text reside in a 2K-word frame buffer formatted with character codes in the high-order bytes and character attribute bits in the low-order bytes. Frame buffer memory is mapped from F0000H (top left corner of display) to F0FFEH (bottom right corner of display) in the video memory address space. Since each word of the frame buffer corresponds to a particular 10- by 12-pixel character cell on the screen, the 8086 can change any display character simply by writing a new 16-bit code to the appropriate frame buffer address.

To create a character display, the video memory controller scans the first 2000 words in the frame buffer and uses each character code entered into the frame buffer as the address of a font table entry. The font table data determines the shape that will appear by defining the pattern of pixels assigned to the character code. For example, if the word at Frame Buffer Address F0794H contains Character Code 0FH in its high-order byte, the controller paints the pixel pattern defined in the sixteenth font table entry into the tenth character cell on the thirteenth line of the display.

Frame buffer entries must be word addressed at even locations. Frame buffer memory access does not affect the CRT display, allowing flicker-free operation without constraints on frame buffer updating. The 8086 can read the frame buffer to determine what characters are present on the display screen. Characters of text can be interspersed with graphics information on the screen; however, text and graphics are ORed together before being displayed and, therefore, each pixel must be allocated to text or to graphics, but not both.



ADDRESS	(CONTENTS)
F2000	(0)
F2002	(0)
F2004	(0)
F2006	(0)
F2008	(0)
F200A	(0)
F200C	(1ECH)
F200E	(92H)
F2010	(92H)
F2012	(92H)
F2014	(92H)
F2016	(0)
F2018	(0)
F201A	(0)
F201C	(0)
F201E	(UNUSED)

Figure 8-3. Character Definition

Figure 8-4 shows frame buffer and font table addressing. The frame buffer address selects one of 2048 character positions (first 2000 mapped to screen with 0 in top left, 1999 in bottom right). The addressed word within the frame buffer then supplies 8-bit character code plus eight attribute bits for that display screen position. The font table address selects one of 256 characters and one of 16 rows (first 15 used, sixteenth row reserved). The addressed word then supplies a 10-pixel horizontal strip within the character cell.

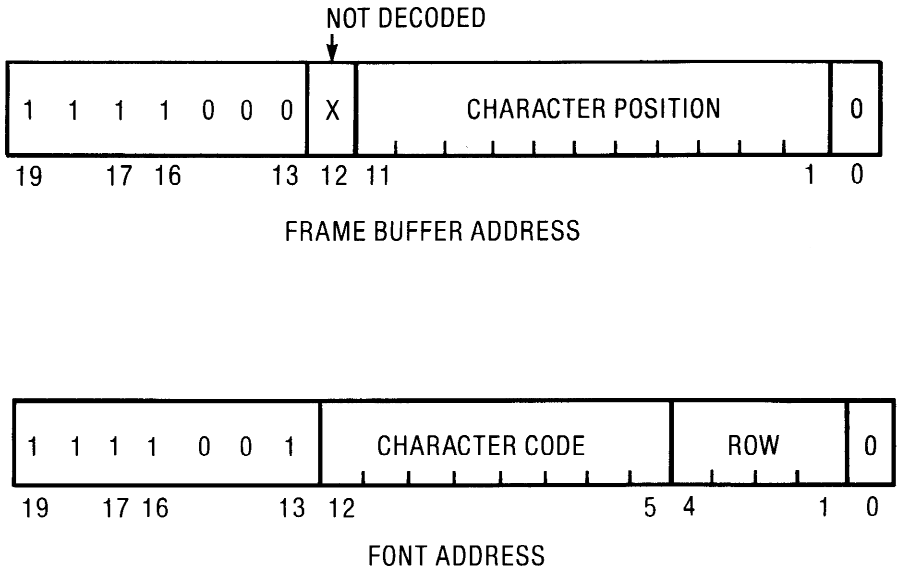


Figure 8-4. Frame Buffer and Font Table Addressing

8.3 BITMAP MEMORY

The 800- by 328-pixel bitmap memory contains a visible region measuring 800 by 300 pixels, which is mapped to the screen, and a nonvisible region measuring 800 by 28 pixels, which is normally not used. Bitmap memory occupies a 16K-word region from E0000H to E7FFFH in the video memory space (refer to Figure 8-2).

The entire 16K-word video memory can be viewed as a matrix containing 328 rows with 50 words in each row. A row of video memory corresponds to a scan line on the display, and only the first 300 rows are mapped to the screen. As shown in Figure 8-2, the visible portion of video memory occupies all 50 words in each of the first 300 rows. Each word in this region of the bitmap contributes a 16-pixel horizontal strip to the display. The remaining 28 rows (578H words) of bitmap memory are not mapped to the screen and make no contribution to the display. Bitmap memory must be word-addressed at even locations. Within a word of bitmap memory, the pixel furthest to the left occupies the high-order bit and the pixel furthest to the right occupies the low-order bit. The 8086 can modify bitmap memory at any time without disrupting the CRT display.

8.4 CHARACTER ATTRIBUTES

Eight character attributes can be selected individually or in combination to modify the normal appearance of a character. Attributes have no effect on graphics. They are selected on a per-character basis by setting specific bits in the low-order byte of the frame buffer entry. (Refer to Section 8.5 for a list of the character attributes and their bit assignments.) Combinations of attributes can be used to represent different character display modes, or fonts, on the screen. For example, using only the bold, underscore, and reverse attributes, a program can define eight different character display modes as follows:

Normal	Reverse
Bold	Reverse bold
Underscore	Reverse underscore
Bold underscore	Reverse bold underscore

Video controller hardware supports all of the 256 possible combinations of attributes; some combinations, however, lead to inconsistencies or conflicts that can detract from their use. If a character has the bold, reverse, and blink attributes, for example, the cursor will not be visible when it is positioned on that character. The cursor generally covers only background pixels in a character cell. In this instance, the cursor and the character's background both appear as high-intensity blinking dots, and one might be indistinguishable from the other.

Figure 8-5 shows how the circuitry integrates font table data, attribute values, and bitmap graphics. Each pixel of a character is ANDed with the blink attribute. The result is then exclusive ORed with reverse attribute. This output is ANDed with the bold attribute to give the pixel's intensity (normal or bold). It is also ORed with the appropriate bit from the bitmap to give the pixel's value (on or off).

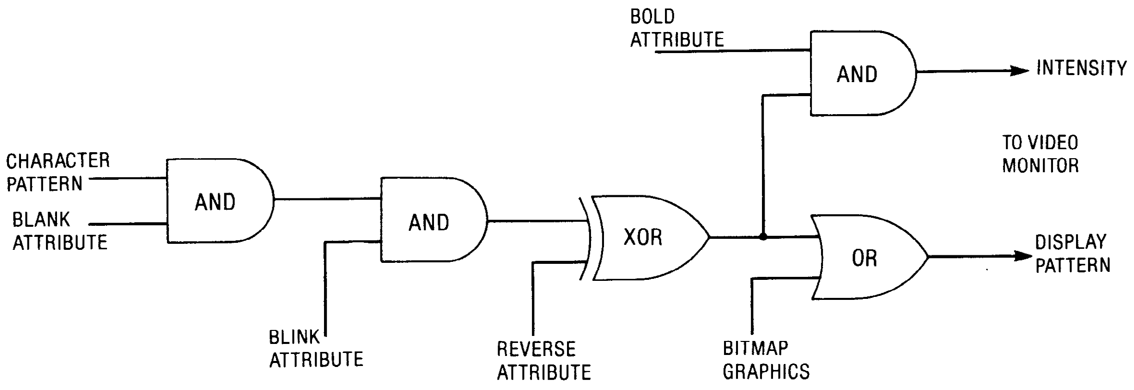


Figure 8-5. Character Attribute Logic

The bold, blank, reverse, and blink attributes modify the way in which a character's font table entry (along with its underscore or overscore attributes) appears on the screen. In contrast, superscript and subscript attributes determine which portion of the font table entry appears. A pixel pattern displayed on the screen is jogged one scan line downward to create a subscripted character, or it is jogged up two lines for a superscript.

The subscript attribute shifts a character one scan line downward within its character cell and the superscript attribute shifts the character two scan lines upward. If both the subscript attribute and the superscript attribute are selected, the character will be shifted one scan line upward. This combination of attributes generally is not used.

8.5 MEDIUM-RESOLUTION CHARACTER ATTRIBUTE BIT ASSIGNMENT

The low-order byte of the 16-bit character code in the frame buffer entry determines the character attribute. A description of the character attributes associated with each bit follows.

BIT 0 -- BLINK
When Bit 0 is set to 1, it causes the character (along with its underscore and overscore attributes) to flash on and off several times a second. The blank attribute supresses blink.

BIT 1 -- REVERSE
Character normally appears light against a dark background. If Bit 1 is set to 1, each pixel is reversed to give a dark character (along with dark underscore and overscore attributes) against a light background.

BIT 2 -- BLANK

Normal character appearance is defined by the character's font table entry. If Bit 2 is set to 1, the character (along with its underscore and overscore attributes) instead appears as a space or blank, just as though none of the pixels are set in its font table entry. Every pixel in a blank character has the background intensity. Therefore, if the reverse attribute is also selected, a blank character will appear as a reverse space, with all of its pixels set.

BIT 3 -- BOLD

When Bit 3 is set to 1, causes the character (along with its underscore and overscore attributes) to appear in high-intensity display mode. If the reverse attribute bit is also selected, it is the background that appears in high intensity. If the blank attribute is also selected, bold will, of course, have no effect.

BIT 4 -- OVERSCORE

When Bit 4 is set to 1, displays every pixel in the top row of the character cell, just as though each of those pixels is set in the character's font table entry. The overscore attribute is unaffected by the subscript and superscript attributes. If the reverse attribute is selected, the overscore will be reversed. If the bold attribute is selected, it will be a high-intensity overscore. If the blank attribute is selected, the overscore also will be blanked. An underscored character can be overscored as well.

BIT 5 -- UNDERSCORE

When Bit 5 is set to 1, displays every pixel in the bottom row of the character cell, just as though each of those pixels was set in the character's font table entry. The underscore attribute is unaffected by the subscript and superscript attributes. If the reverse attribute is selected, the underscore will be reversed. If the bold attribute is selected, it will be a high-intensity underscore. If the blank attribute is selected, the underscore also will be blanked. An overscored character can be underscored as well.

BIT 6 -- SUBSCRIPT

When Bit 6 is set to 1, rolls the character display one scan line downward within its character cell. The bottom row of pixels in a normal character display is shifted out the bottom of the character cell and lost. A new row of pixels defined in the character's font table entry is shifted into the top of the character cell. For a conventional subscript character, this row is usually blank. The subscript attribute determines only how a font table entry is mapped into a character cell and is totally independent of the other attributes.

BIT 7 -- SUPERSCRIPT

When Bit 7 is set to 1, rolls the character display two scan lines upward within its character cell. The top two rows of pixels in a normal character display are shifted upward past the top of the character cell and lost. New rows of pixels defined in the character's font table entry are shifted into the bottom of the character cell. For a conventional superscript character, these two rows are usually blank. The superscript attribute determines only how a font table entry is mapped into a character cell and is totally independent of the other attributes.

8.6 PROGRAMMING THE VIDEO CONTROLLER

Software communicates with the medium-resolution video controller by means of I/O ports described in Table 8-1. All medium-resolution video controller ports are 8-bit ports. Table 8-2 further describes internal registers in the MC6845 controller chip, accessed by means of Address and Controller Register ports at offsets of 0 and 2. This table includes required values for the data that must be loaded into certain controller registers. All controller registers accept 8-bit data unless otherwise indicated.

Table 8-1. Medium-Resolution Video Controller I/O Ports

Port	Description
1x00H	Write Controller Address Register. Bits 0-4 address one of the 18 controller registers listed in Table 8-2. Subsequent write to Controller Data Register then loads this controller register.
1x02H	Write Controller Data Register. Particular controller register addressed by Controller Address Register receives up to eight bits written to this port.
1x10H	Write Option Register. Specific bits enable or disable particular options. D0 -- Video memory accessible by 8086 when set to 1. If cleared to 0, 8086 cannot access video memory. Allows use of another video controller while this controller is installed but inactive. D3 -- When set, vertical sync generates a Level 3 interrupt request at the beginning of the vertical blanking interval. The power-up (default) value is 0 (all options disabled).

Table 8-1. Medium-Resolution Video Controller I/O Ports (continued)

Port	Description
1xFEH	Read Option Board ID. Character board returns the ID code 11H on D0-6 if it is not attached to a graphics board, or the ID code 15H on D0-6 if it is cabled up to a graphics board. In either case, it returns D7 set to 1 only during the vertical blanking interval. Read access to this port also clears a pending interrupt request.

Table 8-2. Medium-Resolution Controller Chip Internal Registers

Address	Content	Description
0	65H	Horizontal Total Register. Determines horizontal frequency. Contains one less than sum of displayed character time units plus non-displayed (retrace) character time units.
1	50H	Horizontal Displayed Register. Designates number of characters on a display line.
2	57H	Horizontal Sync Position Register. Determines horizontal position of display on screen. Increasing this value shifts the display to the left.
3	3FH	Sync Width Register. Bits 0-3 designate HSYNC width in units of character clock pulses. Bits 4-7 designate VSYNC width in units of raster periods.
4	19H	Vertical Total Register (7 bits). Contains one less than number of scan lines per vertical refresh cycle.
5	4	Vertical Total Adjust Register (5 bits). Establishes the number of extra scan lines (forming a partial character line at the bottom of the screen) needed to complete a vertical refresh cycle.
6	19H	Vertical Displayed Register (7 bits). Designates the number of character lines displayed on the screen.
7	19H	Vertical Sync Position Register (7 bits). Determines vertical sync position on screen. Increasing this value shifts the display upward.
8	0	Interlace and Skew Register (2 bits). Value of 0 requests non-interlaced display without skew.
9	BH	Maximum Scan Line Address Register (5 bits). One less than number of scan lines in a character cell.

Table 8-2. Medium-Resolution Controller Chip
Internal Registers (continued)

Address	Content	Description
AH	6AH	Cursor Start Register (7 bits). Regulates cursor format. Bits 0-4 contain the number of the highest scan line in the cursor, counting down from Scan Line 0 at the top of the character cell. Bits 5 and 6 contain 00 to display cursor without blinking 01 to suppress cursor 10 to blink cursor at high frequency 11 to blink cursor at low frequency
BH	BH	Cursor End Register (5 bits). Contains the number of the lowest scan line in the cursor, counting down from Scan Line 0 at the top of the character cell.
CH DH	0	High and Low Start Address Registers (6 and 8 bits, respectively). Determines first refresh address after vertical blanking interval.
EH FH		High and Low Cursor Registers (6 and 8 bits, respectively). Number of characters preceding cursor on display (0-1999).
10H-11H		Unused.

CHAPTER 9
REMOTE TELECOMMUNICATION CONTROLLER

Two remote telecommunication controller (RTC) option boards form the basis of a family of microprocessor-based telecommunication controllers. One option board, the 8232, implements the RS-232C physical link interface for local data communication. Another board, the 8252, provides the X.21 interface to a circuit switched, synchronous, public data network or a packet-switched network. Both RTC options offer high-speed synchronous data communication (at rates of 19.2K baud and higher) plus all of the functionality needed by higher-level network protocols. Together, these RTC options support a hierarchy of data communication protocols embracing the full spectrum of data communication standards.

9.1 8086 INTERFACE TO RTC

The 8086 initializes an RTC by executing an OUT instruction that causes an RTC software reset. Then, during subsequent operations, the 8086 and the Z80A microprocessor in the RTC can communicate in three ways: by exchanging status information, interrupt requests, or data. The 8086 can read a 16-bit register containing RTC status flags; the RTC, in turn, can read 8086 status from an 8-bit status register. Similarly, the 8086 can access an output port that causes a Z80A interrupt request on the RTC board, and the RTC board's Z80A can interrupt the 8086.

Data transfers are the most common means of communication between the 8086 and the RTC. Two registers (called the Inbound and Outbound Data Registers) form a bidirectional data path for either programmed data transfers or DMA transfers. The 8086 has one DMA channel available to receive data from or transmit data to the RTC. In contrast, the RTC has two DMA channels, one for receiving data from the 8086 and one for sending data to the 8086. I/O ports allow the 8086 and the Z80A to execute programmed data transfers along the same data paths used by the DMA interface.

To send outbound data, the 8086 writes the Outbound Data Register, which the Z80A can read; to receive inbound data, the 8086 reads the Inbound Data Register, which the Z80A can write. The Inbound and Outbound Data Registers are named from a system viewpoint. Inbound data, for example, includes data that comes in from the communication link and moves inward through the RTC toward the main system memory. In contrast, the Z80A in the RTC, which looks back towards the 8086 interface, always reads the Outbound Data register and writes the Inbound Data Register.

Both the Inbound Data Register and the Outbound Data Register are 16-bit registers that accept and receive only 8-bit data. For programmed data transfers, these registers are treated exactly as though they were 8-bit I/O ports. When the 8086 writes either byte of the Outbound Data Register, whichever byte it writes is the next byte made available for the Z80A to read. When the Z80A writes a byte to the Inbound Data Register, whichever byte it writes is replicated in the other byte, and the 8086 can read either one. DMA transfers capitalize on this design to multiplex high- and low-order bytes from D0-15 onto the 8-bit RTC board.

In response to a read at offset 10FEH (its highest I/O port address), the 8232 board returns its option ID code of 1CH on D0-6 and the 8252 board returns its 1E option ID code. The option ID code returned on D0-6 from the port offset 10FEH is the only difference between these two RTC option boards that is enforced by hardware. When gating their 7-bit option ID codes onto D0-6, both boards also drive D7 high only if they have a pending 8086 interrupt request. (Refer to Section 9.2 for a listing of all RTC I/O ports that the 8086 can access.) The RTC decodes only the low-order four bits of an I/O port address offset.

9.2 REMOTE TELECOMMUNICATION CONTROLLER I/O PORTS

A description of the Remote Telecommunication Controller ports that the 8086 can access follows.

Port 1xF0H

Read RTC Status Register. The 8086 can read either the low- or the high-order byte of this 16-bit port, or it can read the entire 16 bits of status. Only Bits 10 and 11 of the RTC Status Register are set by hardware. Bit assignments are as follows:

D4-7 -- Set when Z80A software running on the RTC board executes an OUT 52.

D10 -- Outbound Data Register empty. Set when Z80A (or DMA) reads either byte of the Outbound Data Register, Port 1xF8H. Remaining byte may or may not contain data. Cleared when 8086 (or DMA) writes either byte of Outbound Data Register. Generates a DMA request when D2 at the output port offset F2H is set.

D11 -- Inbound Data Register full. Set to indicate that the Inbound Data Register, Port 1xF4H, contains a single byte of data, which will be replicated in the high-order byte of this port. Cleared when the 8086 (or DMA) reads the Inbound Data Register. Generates a DMA request when D3 at the output port offset F2H is set.

D0-3 -- Unassigned.

D8-9 -- Not used.

D12-15 -- Not used.

Write 8086 Status Register. Loads D0-7 into the 8086 Status Register on the RTC.

Port 1xF2H

Write to enable Outbound Data Register and DMA channels. Normally, Bits D0 and D1 are both set during RTC operation. When cleared, they disable the programmed output (i.e., OUT 1xF8H instruction) data path hardware for diagnostics, but they have no effect on DMA transfers. Also, only one of Bits D2 and D3 is normally set at a time. Clearing both D2 and D3 disables DMA transfers between the RTC and system memory. The default after a reset is D0-3 cleared. Bit assignments are as follows:

D0 -- Set to initialize low-order byte of Outbound Data Register. Cleared to disable low-order byte of Outbound Data Register. When cleared, the output port offset F8H is disabled. Has no effect on DMA.

D1 -- Set to initialize high-order byte of Outbound Data Register. Cleared to disable high-order byte of Outbound Data Register. When cleared, the output port offset F8H is disabled. Has no effect on DMA.

D2 -- Outbound DMA enable. When set, the RTC automatically generates a DMA request each time D10 of the RTC Status Register is set, indicating that either half of the Outbound Data Register is empty and able to accept a byte of data. Software sets this bit when the system DMA controller is programmed to make outbound data transfers (i.e., DMA read cycle).

D3 -- Inbound DMA enable. When set, the RTC automatically generates a DMA request each time D11 of the RTC Status Register is set, indicating that the Inbound Data Register contains a byte of data. Software sets this bit when the system DMA controller is programmed to make inbound data transfers (i.e., DMA write cycle).

D4-7 -- Unassigned.

Port 1xF4H

Read Inbound Data Register and clear Bit 11 of RTC Status Register (Port 1xF0H). The 16-bit Inbound Data Register holds only one byte of data, with Bits 0-7 always equal to Bits 8-15. When the Z80A writes either byte of the Inbound Data Register, that byte is automatically copied into the other byte. This allows the 8086 to obtain both high- and low-order bytes by reading the appropriate portion of the register.

Write interrupt priority level and DMA channel assignment. This information must be provided before 8086 interrupts are enabled. It can be changed at any time. Only one of Bits D1, D2, and D3 is normally set at a time. Bit assignments are as follows:

D1 -- Assigns DMA Channel 1 and Interrupt Level 5, if set. (D2 and D3 should be cleared.)

D2 -- Assigns DMA Channel 2 and Interrupt Level 6, if set. (D1 and D3 should be cleared.)

D3 -- Assigns DMA Channel 3 and Interrupt Level 7, if set. (D1 and D2 should be cleared.)

D0 -- Unassigned.

D4-7 -- Unassigned.

Port 1xF6H

Write arbitrary data to generate a Z80A interrupt request by means of Channel 0 of the secondary CTC.

Port 1xF8H

Write Outbound Data Register. Like the Inbound Data Register, the 16-bit Outbound Data Register holds only one byte of data at a time. The 8086 writes either the high-order byte or the low-order byte of this register, but not both, depending upon whether the outgoing data originated in a high- or low-order byte of memory. Data normally alternates between high- and low-order bytes.

Port 1xFAH

Write arbitrary data to acknowledge 8086 interrupt request from RTC. Causes RTC to remove its interrupt request.

Port 1xFCH

Write arbitrary data to reset the RTC.

Port 1xFEH

Read Option ID code (D0-6 will be 1CH for the 8252 board or 1EH for the 8232 board) and interrupt status (D7 set if the RTC has a pending 8086 interrupt request).

On the 8252 (X.21) option only, write with

D4 set to 1 to generate Level 2 interrupts
D5 set to 1 to generate Level 3 interrupts
D6 set to 1 to generate Level 4 interrupts

Only one of Bits 4-6 is normally set at a time, and Bits 1-3 of Port 1xF4H are normally all cleared to 0 when any of these bits is set. (If one of Bits 1-3 in Port 1xF4H is set to 1, indicating that DMA is being used, the DMA channel determines the interrupt priority level).

9.3 RTC ARCHITECTURE

Outbound parallel data flows from the system bus through the Outbound Data Register to RTC memory and from there to the Serial Input/Output (SIO) chip, which converts the outbound data to serial format and sends it to the RS-232C or X.21 interface connector. On the inbound data path, serial data from the communication interface travels first to the SIO chip, which converts it to parallel format and then through RTC memory on its path to the Inbound Data Register.

Figure 9-1 shows the RTC architecture. The outbound and inbound data registers at the left of the figure interface the RTC with system bus. The SIO chip, shown as two separate channels at the right, interfaces the RTC with the communication link. Four DMA channels connect RTC memory with the 8086 interface and the SIO chip.

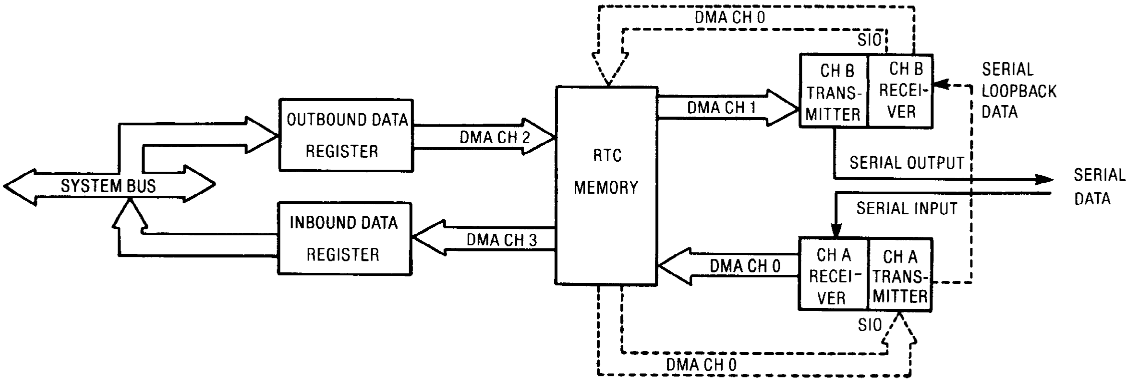


Figure 9-1. RTC Architecture

The SIO chip contains two independent full-duplex channels. Channel A receives input from the RS-232C or X.21 communication interface and transmits loopback output to SIO Channel B. Channel B, in turn, transmits output to the appropriate communication interface and receives loopback input from Channel A. Broken lines in Figure 9-1 show the loopback data path for diagnostic loopback operation, which moves counterclockwise around the right side of the figure. The normal data path, which moves clockwise, is drawn with solid lines.

A 9517-1 DMA controller performs fast, bidirectional, 1-byte data transfers between the Inbound and Outbound Data Registers, RTC memory, and the SIO chip. When receiving data, a DMA write cycle carries inbound data from SIO receiver Channel A to RTC memory across DMA Channel 0. When transmitting, SIO transmit Channel B obtains outbound data from RTC memory by means of a DMA read cycle on DMA Channel 1. Simultaneous transfers in both directions support full-duplex communication. During loopback operation, DMA Channel 1 carries data to SIO Channel A instead of Channel B, and DMA Channel 0 returns data from Channel B instead of Channel A.

9.4 RTC MEMORY

RTC memory consists of 4K bytes of EPROM for bootstrap loading and diagnostic programs and 60K bytes of RAM. EPROM occupies the lowest 4K bytes of address space. RAM, which begins at memory address 1000, is organized as 9-bit memory with eight data bits and a parity bit. RAM parity errors trigger a Z80A nonmaskable interrupt. All I/O hardware on the RTC boards uses I/O-mapped port addressing. Table 9-1 lists the RTC port addresses.

9.5 DIRECT MEMORY ACCESS

Four high-speed DMA data paths reduce processor overhead when transferring data to and from RTC memory. The Channel 0 path carries inbound data from the SIO chip to RTC memory. A buffer full signal from the low- or high-order byte of either SIO receiver channel initiates a Channel 0 DMA request. On DMA Channel 1, a companion data path carries outbound data from RTC memory to the SIO chip. As with Channel 0 DMA requests, a buffer full signal from either SIO transmitter initiates a Channel 1 DMA request.

On DMA Channel 2, a third data path connects the Outbound Data Register to RTC memory. Hardware automatically generates a Channel 2 DMA request when either byte of data is received from the system bus and ready to be transferred from the Outbound Data Register to RTC memory. On DMA Channel 3, a companion path connects RTC memory to the Inbound Data Register. Again, hardware automatically generates a Channel 3 DMA request when either byte of inbound data is gated onto the system bus and new data can be accepted from RTC memory.

Table 9-1. RTC Z80A I/O Port Addresses

Port	Description
00-03	SIO chip registers.
10H-1FH	DMA controller chip registers.
20H-23H	Primary CTC registers.
30H	Write a data character to clear that character's recognition table entry (enabling its character recognition interrupt).
31H	Write a data character to set that character's recognition table entry (disabling its character recognition interrupt).
40H	Read Option Switches. Loads configuration option switch settings into the accumulator with Switch 1 as the least significant bit and Switch 8 as the most significant bit. Write with D0 -- 0 to select external clock (synchronous operation). 1 for internal clock (asynchronous operation). (Refer to the description of Clock mode flipflop in Section 9-6 or 9-8.) D1 -- 0 to send serial data in NRZ mode. 1 to select NRZI mode. D4 -- 0 to set /SRTS to mark. 1 to set /SRTS to space. D7 -- Set to generate an 8086 interrupt request. (The default is 0 in every case.)
44H	Read 8086 Status Register. Write arbitrary data to reset RTC.
48H	Write arbitrary data to signal "DTE Ready" (C=OFF and T=0) on the 8252 (X.21) option board only. The power-up default condition is "DTE Uncontrolled Not Ready" (C=OFF and T=1).

Table 9-1. RTC Z80A I/O Port Addresses (continued)

Port	Description
4CH	Read ACU Status Register on 8232 (RS-232C) option only. Bit assignments are as follows: D0 -- Abandon Call and Retry D1 -- Present Next Digit D2 -- Power Indicator. Power on when set D3 -- Call Origination Status D4 -- Data Line Occupied D5-7 -- Unassigned Write ACU Command Register on 8232 (RS-323C) option only. Bit assignments are as follows: D0-3 -- Digit Bits 0-3 D4 -- Digit Present D5 -- Digit Request Flag D6-7 -- Unassigned Write with Bit 7 set to 1 to enable break detection on the 8252 (X.21) board only. When break detection is enabled, circuitry on the 8252 board gates the received data and clock to the SIO Channel B receiver, which is normally not used except for loopback testing. SIO Channel B then detects the continuous low condition, lasting at least 16 bit times, that signals a break.
50H	Read Outbound Data Register.
51H	Write low-order byte of RTC Status Register. Assigned bits in the high-order byte are set by hardware.
52H	Write D4-7 into Bits 12-15 of the RTC Status Register.

Table 9-1. RTC Z80A I/O Port Addresses (continued)

Port	Description
54H	Enable Inbound Data Register. Bits D4 and D5 normally are both set during RTC operation. When cleared, they disable the programmed output (i.e., OUT 56H instruction) data path hardware for diagnostics, but they have no effect on DMA transfers. Bit assignments are as follows: D4 -- Set to initialize low-order byte of Inbound Data Register. Cleared to disable low-order byte of Inbound Data Register. When cleared, Output Port 56H is disabled for low-order byte output. Has no effect on DMA. D5 -- Set to initialize high-order byte of Inbound Data Register. Cleared to disable high-order byte of Inbound Data Register. When cleared, Output Port 56H is disabled for high-order byte output. Has no effect on DMA. D0-3 -- Unassigned. D6-7 -- Unassigned.
56H	Write Inbound Data Register. OUT 54H with D4 and D5 set should precede first OUT 56H.
60H-63H	Secondary CTC Registers
70H	Read to toggle LED indicator. Read returns byte of 0 and turns LED on (if it is off) or off (if it is on). The initial setting is on. Write arbitrary data to enable bad (i.e., odd) parity generation.
71H	Write arbitrary data to enable good (i.e., even) parity generation. In the absence of a write to Port Offset 70H, good parity is the default setting.

9.6 PRIMARY AND SECONDARY COUNTER/TIMER CHIPS

The primary CTC is used to generate Z80A interrupt requests when any of the four DMA channels completes a DMA transfer operation or when the character recognition logic encounters a special character. It also provides a programmable baud rate clock and performs general-purpose counting or timing functions. The secondary CTC generates Z80A interrupt requests for the Inbound and Outbound Data Registers and the programmed interrupt request port. Thus, primary CTC events originate on the RTC board and secondary CTC events originate in the 8086. References to the CTC imply the primary CTC unless the secondary CTC is specified.

CTC Channel 0 is programmed to operate in counter mode with an initial value of 1; it is used only to generate character recognition interrupt requests. Output from the character recognition lookup table is gated by DMA control logic and applied to the counter input of CTC Channel 0 during DMA write cycles. An active input indicates that the last character received on SIO Channel A was recognized as a special character. This sets the Channel 0 count to 0, and triggers the character recognition interrupt request.

CTC Channel 1 is the programmable baud rate generator. Operating as a timer, it produces the SIO Channel A receiver clock and the SIO Channel B transmitter clock, but only when the clock mode flipflop is cleared to select internal clock operation with an asynchronous modem. Optionally, CTC Channel 1 can also generate an internal baud rate clock for loopback operation, an internal clock for CRC character generation, or a baud rate clock for synchronous or asynchronous null modem applications. When operating as a counter, CTC Channel 1 tallies RS-232C transmitter clock pulses and generates a clock for checking flag characters under the SDLC or HDLC protocols.

CTC Channel 2 receives /EOP signals from the DMA controller and generates DMA End-of-Process interrupt requests. Like CTC Channel 0, which performs a similar function to request character recognition interrupts, CTC Channel 2 operates in counter mode with an initial count value of 1.

CTC Channel 3 is available for use as a general-purpose timer. In addition, Channels 2 and 3 are connected in series to implement an extended timer with Channel 3 as the high-order stage. The extended timer can be used for relatively lengthy timeouts or other special functions when DMA transfers are not required.

All four primary CTC input Channels must be programmed to accept active low-trigger inputs when operating in their counter mode. Secondary CTC input channels must be programmed to accept active high-trigger inputs when operating in their counter mode on Channels 1 and 2. Channel 0 of the secondary CTC must be programmed to accept active low trigger inputs.

9.7 DMA END-OF-PROCESS INTERRUPTS

Programmed to operate in counter mode with an initial count value of 1, CTC Channel 2 receives /EOP (end-of-process) signals from the DMA controller chip and generates a Z80A interrupt request whenever /EOP indicates that a DMA transfer operation is finished. To service the /EOP interrupt, software examines the DMA controller's internal Status Register and identifies the DMA channel (or combination of channels) that has reached terminal count.

An /EOP latch circuit prevents the CTC from missing a second /EOP pulse that might occur after the CTC generates an interrupt request but before it is reinitialized to process the next /EOP pulse. When an interrupt service routine executes an IN 18 to read the DMA controller's Status Register, special circuitry on the RTC board recognizes the IN 18 instruction, clears the old /EOP indication, and prepares to latch a new /EOP signal. After the interrupt service routine finishes processing one /EOP interrupt, the /EOP latch circuit triggers the CTC to request another interrupt in response to the second /EOP signal.

9.8 CLOCK MODE AND INTERNAL VERSUS EXTERNAL TIMING

When power is first applied, the clock mode flipflop is set by default to establish synchronous operation at a rate determined by external transmit and receive clocks in the modem. In this mode of operation, the Receiver Clock (RC) and Transmitter Clock (TC) lines from the communication interface connect to the SIO Channel A transmitter clock (TxCA) and the SIO Channel B receiver clock (RxCB), respectively. RC originates on Pin 17 of the RS-232C connector or on Pin 6 of the X.21 connector. TC comes from Pin 15 of the RS-232C connector or Pin 13 of the X.21 connector. An OUT 40 instruction with D0 cleared to 0 sets the clock mode flipflop to establish this default condition.

On the RTC-232C board, an OUT 40 instruction with D0 set to 1 clears the clock mode flipflop to establish asynchronous operation at a programmed baud rate determined by the clock output from CTC Channel 1. Receiver and transmitter clock lines from the communication interface are disconnected and the CTC Channel 1 clock instead supplies both TxCA and RxCB. (The CTC Channel 1 clock also drives Pins 11 and 19 of the RS-232C connector to supply a clock for use by the null modem and the RxCA and TxCB clocks used during loopback operation. However, these Channel 1 clock paths from the CTC are always enabled and do not require an OUT 40 instruction.)

9.9 SECONDARY REQUEST TO SEND

Executing the OUT 40 instruction with D4 cleared to 0 sets the Secondary Request to Send signal (/SRTS on Pins 11 and 19 of the RS-232C connector) to a mark condition. If D4 is set instead to 1, /SRTS is set to a space condition. This allows for secondary channel break transmission when a half-duplex modem is attached to the RTC-232C board. If an external modem needs an active Secondary Request to Send signal on Pin 19 (other than a secondary channel break transmission), a jumper isolates Pin 19 for exclusive use as a normal /SRTS line. Because they configure the RS-232C interface, Bits D0 and D4 are not used by the OUT 40 instruction on 8252 boards.

9.10 SPECIAL CHARACTER RECOGNITION

A 256-bit by 1-bit static RAM contains a lookup table that determines whether any character received by SIO channel A is one of the special characters that can cause an interrupt request. The lookup table RAM takes its address from the 8-bit data bus and its data from the low-order address line (A0). Therefore, when initializing the character recognition table, accumulator data identifies the lookup table entry, while the Z80A executes either an OUT 30 or an OUT 31. An OUT 30 clears the entry addressed by the accumulator contents (and thereby requests an interrupt whenever the corresponding character is recognized). An OUT 31 sets that lookup table entry (and suppresses interrupt requests when the corresponding character is recognized).

Each inbound character received on SIO Channel A is applied to the character recognition RAM by means of the data bus. The corresponding 1-bit lookup table entry sets or clears a flipflop. DMA circuits clock the flipflop during a DMA write cycle to trigger the counter inputs of CTC Channel 0 if the lookup table entry was 0, indicating a special character that should trigger an interrupt. CTC Channel 0 is programmed in counter mode with an initial value of 1 and is used only to generate the actual interrupt request.

9.11 Z80A INTERRUPTS

Nonmaskable interrupts indicate parity errors, which have the highest priority except for the bus request (/BUSRQ) signal from the DMA Controller. After NMIs, maskable Interrupt Mode 2 and vectored interrupts have the highest priority. The SIO chip and the CTC are the only sources of vectored interrupt requests. Connected in a daisy chain, they provide an interrupt nesting mechanism that automatically selects the highest priority device when both request service simultaneously. An RETI instruction ends an interrupt service routine by reinitializing the daisy chain for proper handling of nested priority interrupts. The possible RTC vectored Z80A interrupts, in order of priority, are as follows:

1. SIO Channel A Receiver (highest priority).

2. SIO Channel A Transmitter (used only for loopback operation).

3. SIO Channel A Status (not used under BISYNC). A Ring Indicator signal provides external/status signal.

4. SIO Channel B Receiver (used only for loopback operation).

5. SIO Channel B Transmitter.

6. SIO Channel B Status (not used under BISYNC).

7. Primary CTC Channel 0 (character recognition interrupt or general-purpose timer).

8. Primary CTC Channel 1 (SIO receiver and transmitter clocks or counter for checking SDLC and HDLC flag characters).

9. Primary CTC Channel 2 (end of process for all four DMA channels).

10. Primary CTC Channel 3 (software timer).

11. Secondary CTC Channel 0 (Z80A interrupt request from 8086).

12. Secondary CTC Channel 1(Z80A interrupt when 8086 writes Outbound Data Register).

13. Secondary CTC Channel 2 (Z80A interrupt when 8086 reads Inbound Data Register).

14. Secondary CTC Channel 3 (unassigned, lowest priority).

9.12 AUTOMATIC CALLING UNIT

On the 8232 (RS-232C) option board, an automatic calling unit (ACU) completely automates data transmission by means of the telephone network. Four binary signal lines (D0-3 or NB1; 2; 4; and 8) to the ACU carry a defined character set of 16 codes. Codes 0 through 9 correspond to the digits 0 through 9. Codes 10 through 13 correspond to asterisk (*), pound sign (#), End of Number (EON), and the separation control character (SEP), respectively. Codes 14 and 15 are not defined. The RTC sends EON after sending the last digit of the number it is dialing. In response to EON, the ACU immediately transfers the communication channel to the data set without waiting for an answer signal from the called data set. SEP signals a pause between successive digits. For example, in response to SEP, the ACU may again wait for a dial tone before turning ON circuit PND. (Table 9-2 provides a description of the ACU interface.)

Table 9-2. Automatic Calling Unit Interface

Name	Pin	I/O	Description
NB1	14	O	Digit Bit 0.
NB2	15	O	Digit Bit 1.
NB4	16	O	Digit Bit 2.
NB8	17	O	Digit Bit 3.
DPR	2	O	Digit Present. Indicates to the ACU that it can read the 4-bit code on NB1, 2, 4, and 8, which must not change state while DPR is active.
CRQ	4	O	Call Request. Asks the ACU to place a call. Must remain active to hold the line open (i.e., remain off-hook). ACU aborts the call if CRQ goes inactive before COS or DSC goes active.
ACR	3	I	Abandon Call and Retry. When active, determines that a prescribed time has elapsed between successive events in the calling procedure and thereby indicates to the RTC when the call should be abandoned. Action is required by the RTC to abandon the call.
PND	5	I	Present Next Digit. When active, indicates that the ACU is ready to accept the next 4-bit digit from the RTC. Otherwise, when inactive, tells the 8086 to disable DPR and set the NB1, 2, 4 and 8 for the next digit.
PWI	6	I	Power Indication. Active to indicate that the ACU is powered on.
COS	13	I	Call Origination Status. Designated as DSS in an earlier RS-366 specification, indicates whether connection to a slave has been established. It can also indicate the status of the automatic call origination procedures. COS does not, however, indicate the operational status or preparedness of the associated data set.
DLO	22	I	Data Line Occupied. When active, indicates that the commmunication channel is in use (i.e., controlled by equipment other than the ACU originating the call). Falling edge does not occur until all interchange circuits return to their idle condition.

CHAPTER 10
WINCHESTER DISK CONTROLLER

The Winchester Disk Controller (WDC) is an intelligent, Z80A-based, I/O option board that operates one 5.25-inch, Winchester-technology, random access disk drive. The Winchester drive stores up to 10 MB of formatted data by using both sides of its one or two permanently mounted platters. Under the direction of the WDC, the drive's moveable read/write head assembly accesses 304 logical tracks on each of the recording media surfaces. There are sixteen 512-byte sectors per track on each surface, for a total of 64 sectors per cylinder on a dual-platter drive or 32 sectors per cylinder on a single-platter drive. Disk platters and recording heads are housed within a hermetically sealed enclosure with a filtered air recirculating system that protects both the media and the heads from temperature variation and environmental contamination.

10.1 WINCHESTER DISK PROGRAMMING

Except for their faster speed and larger storage capacity, 5.25-inch Winchester disk drives have much in common with 5.25-inch floppy disk storage units. WDC capitalizes on the many similarities between these two types of disk drives by implementing a set of commands whose format, function, and operation resemble those of the floppy disk controller. However, although it is loosely modeled on its floppy disk counterpart, the WDC is, in general, more straightforward and easier to use than the floppy disk controller.

As with the floppy disk, WDC command execution is best seen in four phases. An optional initialization phase begins the WDC command sequence. During the initialization phase, the 8086 accesses I/O ports to reset the WDC, abort any previous operation that might still be in progress, establish or change the WDC's interrupt priority level and DMA channel assignment, and perform other command set-up functions. Once the WDC has been initialized, a sequence of WDC commands can be executed without repeating the initialization procedure; however, a program has the option of reinitializing the WDC at any time. Table 10-1 lists the WDC I/O port assignments, and Table 10-2 describes the contents of the Main Status Register.

Table 10-1. Winchester Disk Controller I/O Ports

Port	Description
1x00H	Read WDC Main Status Register (refer to Table 10-2). Write arbitrary data to abort current operation and initialize for a new command. Terminates the command in progress immediately, without returning operation status bytes.
1x02H	Read Operation Status Register. When the WDC finishes performing most commands, the 8086 must read a sequence of eight status bytes from this port before initiating the next Winchester disk operation. WDC Main Status Register must indicate ready status before the 8086 can read this port. Write Command Register. To initiate a Winchester disk operation, the 8086 writes a sequence of eight command bytes to this port. WDC Main Status Register must indicate ready status before the 8086 can write to this port.
1x04H	Read to clear WDC interrupt request. Returns arbitrary data.
1xFCH	Write arbitrary data to reset the WDC. Executes the power-on diagnostic and sets the Controller Fault Flag (Bit 1 in the WDC Main Status Register) if the diagnostic fails.
1xFEH	Read Option ID code (D0-6 will be 01) and interrupt status (D7 set if the WDC has a pending 8086 interrupt request). Write interrupt priority level and DMA channel assignment. This information must be provided before 8086 interrupts are enabled. It can be changed at any time. Only one of Bits D1, D2, and D3 is normally set at a time. Bit assignments are as follows: D1 -- Assigns DMA Channel 1 and Interrupt Level 5 if set. (D2 and D3 should be cleared.) D2 -- Assigns DMA Channel 2 and Interrupt Level 6 if set. (D1 and D3 should be cleared.) D3 -- Assigns DMA Channel 3 and Interrupt Level 7 if set. (D1 and D2 should be cleared.) D0 -- Unassigned. D4-7 -- Unassigned.

Table 10-2. Winchester Disk Controller Main Status Register
(Input Port Offset 1x00H)

Port	Description
0	Ready/Busy Flag. Set to 1 when WDC is busy. WDC accepts Command Register input by means of the I/O port offset 1x01H only when this bit is cleared to 0, which indicates ready status.
1	Controller Fault Flag. Set to 1 to indicate that the power-on diagnostic recognized a WDC error condition.
2	Read Status Flag. Set to 1 when the WDC is ready to return the first byte of a status byte sequence. Cleared to 0 when the 8086 reads the eighth and final byte of a status byte sequence. While set, indicates that the 8086 must accept additional status bytes (or abort the current operation by writing to I/O Port 1X00H) before it can issue the next WDC command.
3	Drive Size Flag. Cleared to 0 if drive is a 5MB unit, otherwise set to 1.
4-7	Number of command bytes received -- or status bytes sent -- since end of last operation. WDC increments this 4-bit field from 0 to 7 as the 8086 reads each byte of status, and clears it after the 8086 reads the eighth status byte. It again increments this field from 0 as the 8086 writes each command byte, and again clears it after the 8086 writes eighth command byte. High-order Bit 7 will be set momentarily while the eighth byte is being transferred in either direction.

10.2 COMMAND PHASE

During the command phase, after any required initialization has been performed, the 8086 loads a sequence of eight command bytes into the WDC Command Register (Output Port 1x02H). The 8-byte command sequence supplies all information that the WDC needs to execute the command. It establishes the function code; a disk address consisting of the next sector, cylinder, and head to access (if any); and the number of sectors involved in the command (if applicable). Certain WDC commands accept other specialized types of information in their 8-byte command sequence. However, regardless of the function being performed, the 8086 always transfers exactly eight bytes into the WDC Command Register, even though some functions do not need all of the command phase information. These eight bytes generally designate the following conditions:

1. Function code
2. Second byte always cleared to 0
3. Low-order byte of starting cylinder number
4. High-order byte of starting cylinder number
5. Starting surface (i.e., read/write head) number (0-3)
6. Starting sector number (0-31)
7. Low-order byte of number of sectors to transfer
8. High-order byte of number of sectors to transfer

10.3 EXECUTION PHASE

To read or write disk data, an 8086 program first initializes a DMA channel and then issues the appropriate WDC command. The execution phase begins when the 8086 writes the eight command bytes to the WDC Command Register. During the execution phase, the WDC issues DMA requests and transfers single bytes of data across the DMA channel as needed. Internal WDC memory buffers up to three sectors (1536 bytes) of input or output data; therefore, the DMA data rate is not critical when reading or writing the Winchester disk. A WDC interrupt signals the end of the execution phase and the beginning of the result phase once the required number of sectors has been transferred.

During a read, write, or other sector-oriented operation, the current disk address automatically increments, increasing from Sector 0 to Sector 0FH, at the end of every 512-byte block. After Sector 0FH on one surface has been accessed, the surface (or head) number automatically increases by one and Sector 0 on the next surface is accessed next. After the WDC accesses Sector 0FH on the highest surface (either Surface 1 on a single-platter drive or Surface 3 on a dual-platter drive), the cylinder number automatically increases by one and Sector 0 of Surface 0 on the next cylinder is accessed next. Unlike sector and surface numbers, the cylinder address does not wrap around to 0 after the highest numbered cylinder has been accessed. Therefore, a single WDC command can read or write any contiguous region of disk data, including the entire disk, but it cannot read or write beyond the highest numbered cylinder.

10.4 RESULT PHASE

If the command is successful, a WDC interrupt request signals the end of the execution phase and the beginning of the result phase. The result phase makes eight bytes of status information and other housekeeping data available to the 8086 through the WDC Operation Status Register (Input Port 1x00H). In response to a WDC interrupt, the 8086 must either abort the command or read all eight bytes of result phase data. The eight bytes of result phase data designate the following conditions:

1. Function code for the command just performed
2. Operation Status Code (refer to Table 10-3)
3. Low-order byte of next (or last) cylinder number
4. High-order byte of next (or last) cylinder
5. Next (or last) surface (i.e., head) number (0-4)
6. Next (or last) sector number (0-15)
7. Number of sectors transferred (low-order byte)
8. Number of sectors transferred (high-order byte)

The disk address contained in the next (or last) sector, surface, and cylinder fields is the address of the last sector transferred sucessfully (when an error occurs) or, in the absence of an error, the address of the sector following the last sector transferred. As mentioned earlier, sector, surface, and cylinder numbers automatically wrap around from Sector 0FH to Sector 0 of the next surface, from the highest numbered surface to Surface 0 of the next cylinder, and from one cylinder to the next. However, after Sector 0FH on the highest numbered surface of Cylinder 96H -- the last sector on the disk -- has been transferred successfully, an illegal cylinder number will appear as the next cylinder number in the third status byte.

Reading I/O Port 1x04H returns arbitrary data and automatically clears the 8086 interrupt request that signaled the beginning of the result phase. The end of the result phase -- and the end of a WDC command -- occurs when the 8086 reads the eighth and final byte of Operation Status Register information. As with the eight bytes of command phase input, the result phase always returns exactly eight bytes of output even though most commands do not use all eight status bytes. A list of the Winchester Disk Controller commands and the functions they perform follows.

CODE C0H -- READ

Begins reading the Winchester disk at the starting cylinder, surface, and sector designated in the third through sixth bytes of the command byte sequence. Transfers the number of sectors designated by the seventh and eighth command bytes. After the last sector on a surface, continues with the first sector on the next surface. After the last surface on a cylinder, continues with the first surface on the next cylinder.

In the absence of errors, returns with the second byte of the status byte sequence set to 80H or 83H; the third through sixth status bytes set to address the sector following the last sector transferred (or Sector 0 of Surface 0 on illegal Cylinder 97H, if Sector 0FH on Surface 3 of Cylinder 96H was accessed); and the seventh and eighth status bytes set to the number of sectors transferred.

If an error occurs, the second status byte contains the error code, the third through sixth status bytes are set to address the last sector transferred successfully, and the remaining two status bytes indicate the number of sectors successfully transferred. Possible error codes are 03 for an uncorrectable ECC error, 04 for failure to locate a particular sector, or 09 for a drive fault.

READ Command Sequence	READ Status Sequence
1. C0H	1. C0H
2. 0	2. 80H or 83H (or error code)
3. First cylinder to read (low-order)	3. Next (or last) cylinder
4. First cylinder high-order	4. Next cylinder high-order
5. Starting surface (head) to read	5. Next (or last) surface
6. Starting sector to read	6. Next (or last) sector
7. Number of sectors to read (low-order)	7. Sectors read successfully
8. Number of sectors to read (high-order)	8. Sectors read (high-order)

CODE C1H -- WRITE

Same as READ, except for the direction of transfer and the types of errors that are possible. If an error occurs during a WRITE transfer, the second status byte contains one of the following error codes: 02 if the disk is write-protected, 04 for failure to locate a particular sector, or 09 for a drive fault.

WRITE Command Sequence	WRITE Status Sequence
1. C1H	1. C1H
2. 0	2. 80H (or error code)
3. First cylinder to write	3. Next (or last) cylinder
4. First cylinder high-order	4. Next cylinder high-order
5. Starting surface (head) to write	5. Next (or last) surface
6. Starting sector to write	6. Next (or last) sector
7. Number of sectors to write (low-order)	7. Sectors written successfully
8. Number of sectors to write (high-order)	8. Sectors written (high-order)

C2H -- VERIFY

Same as READ, except that data from disk is transferred only as far as the WDC internal data buffer. Error checking and correction are performed, but no DMA requests are made and no data passes across the system bus. VERIFY establishes the integrity of a region on the disk without transferring data across the system bus and into system memory. Possible errors are identical to those for the READ command.

VERIFY Command Sequence	VERIFY Status Sequence
1. C2H	1. C2H
2. 0	2. 80H or 83H (or error code)
3. First cylinder to read (low-order)	3. Next (or last) cylinder
4. First cylinder high-order	4. Next cylinder high-order
5. Starting surface (head) to read	5. Next (or last) surface
6. Starting sector to read	6. Next (or last) sector
7. Number of sectors to read (low-order)	7. Sectors read successfully
8. Number of sectors to read (high-order)	8. Sectors read (high-order)

C3H -- REFORMAT TRACK

Reformats the track designated by the third, fourth, and fifth command bytes. The sixth, seventh, and eighth command bytes are not used and should be cleared to 0. A track is the portion of a cylinder located entirely on one surface, under one head. A track contains all 16 of the sectors comprising one surface in a cylinder. After reformatting the designated track and writing a prescribed data pattern, the REFORMAT TRACK command reads the track to establish that it was formatted correctly and that the data was stored and retrieved correctly. (Table 10-5 explains the recording format.)

If the newly formatted track can be located but its data cannot be read, the track is flagged as a bad track. This is not considered to be an error, since the formatting operation succeeded even though the test data generated an ECC error. REFORMAT TRACK returns with the second status byte set to 80H, if successful; to 02 if the disk is write-protected; to 04 on failure to locate the designated track (either before or after reformatting it); or to 09 for a drive fault.

REFORMAT TRACK Command Sequence	REFORMAT TRACK Status Sequence
1. C3H	1. C3H
2. 0	2. 80H (or error code)
3. Cylinder to reformat (low-order)	3. 0
4. Cylinder high-order	4. 0
5. Surface containing track (i.e., head)	5. 0
6. 0	6. 0
7. 0	7. 0
8. 0	8. 0

C9H -- WRITE AND VERIFY

Same as WRITE, except that after transferring each sector onto the disk, reads back the sector, performs error checking, and compares the recovered data with the original data. Returns the same status information as a WRITE command if successful. On an error, returns the same status information as a WRITE command but recognizes two additional error conditions: an ECC error when reading data back from the disk (second status byte set to 03), or a verification error if the recovered data does not match the original data (second status byte set to 05).

WRITE AND VERIFY Command Sequence	WRITE AND VERIFY Status Sequence
1. C9H	1. C9H
2. 0	2. 80H or 83H (or error code)
3. First cylinder to write	3. Next (or last) cylinder
4. First cylinder high-order	4. Next cylinder high-order
5. Starting surface (head) to write	5. Next (or last) surface
6. Starting sector to write	6. Next (or last) sector
7. Number of sectors to write (low-order)	7. Sectors written successfully
8. Sectors to write (high-order)	8. Sectors written (high-order)

F0H -- RUN DIAGNOSTIC

Performs the power-up diagnostic and sets Bit 01 of the WDC Main Status Register (Input Port 1x00H) accordingly. RUN DIAGNOSTIC ignores the second through eighth bytes of the standard command sequence. It does not have a result phase; therefore, RUN DIAGNOSTIC does not generate an 8086 interrupt request upon completion, and it does not return the standard eight bytes of status information.

F1H -- RESET RETRY COUNTER

Clears the WDC internal retry counter to 0. RESET RETRY COUNTER ignores the second through eighth bytes of the standard command sequence. It is always successful, which means that it always returns F1H and 80H in the first two bytes of its status sequence, and it concludes with the third through eighth status bytes cleared to 0.

F2H -- READ RETRY COUNTER

Returns the current value of the WDC internal retry counter in the seventh and eighth status bytes, with the seventh status byte containing the low-order byte of the retry count and the eighth status byte containing the high-order byte of the retry count. The retry counter tallies the number of times that the WDC had to repeat a disk access to locate the designated sector. READ RETRY COUNTER ignores the second through eighth bytes of the standard command sequence. It is always successful, which means that it returns F2H and 80H in the first two bytes of its status sequence, and it concludes with the third through sixth status bytes cleared to 0.

F3H -- FORMAT DISK

Third through sixth command bytes designate a calendar date with the (binary) month in the third byte, the day of the month in the fourth byte, and the year in the fifth and sixth bytes. The entire disk is formatted and the date is written onto the disk. FORMAT DISK returns the second status byte set to 80H if successful, 02 if the disk is write-protected, 04 if it cannot read a newly formatted sector, and 09 on a drive fault. (Table 10-5 explains the Winchester disk recording format.)

FORMAT DISK Command Sequence	FORMAT DISK Status Sequence
1. F3H	1. F3H
2. 0	2. 80H (or error code)
3. Month	3. 0
4. Day	4. 0
5. Year (first of two bytes)	5. 0
6. Year (second byte)	6. 0
7. 0	7. 0
8. 0	8. 0

F4H -- READ ERROR DATA

Returns the content of the 512-byte buffer most recently loaded from the disk. After an uncorrectable error, this allows the 8086 to access data that is known to contain errors for the purpose of salvaging as much information as is possible. READ ERROR DATA uses only the first byte of the standard command byte sequence; the remaining seven bytes should be cleared to 0. READ ERROR DATA is always successful, which means that it always returns 80H in the second byte of its status sequence. Remaining status bytes are cleared to 0.

F5H -- READ RESERVED CYLINDER

Same as READ DATA except that only the first command byte is used. Remaining bytes in the standard 8-byte command sequence should be cleared to 0. Reads the two (or more) sectors comprising the reserved cylinder and, if successful, returns the third through sixth status bytes cleared to 0, 02 in the seventh status byte, and 0 in the eighth status byte. Errors are reported as they are for the READ DATA command.

F6H -- WRITE ENABLE

Allows execution of commands that can alter disk data. These include WRITE (ClH), WRITE AND VERIFY (C9H), REFORMAT TRACK (C3H), FORMAT (F3H), and WRITE PROTECT (F7H). WRITE ENABLE accepts an arbitrary 6-byte password in its third through eighth command bytes. To perform a WRITE ENABLE on a write-protected disk, the WDC compares this password against the password that was last used to write-protect the disk and, if the passwords match, clears the write-protect flag to write-enable the disk. WRITE ENABLE returns the 02 error code if the passwords do not match and the 04 error code if it cannot read the disk. If the disk is not write-protected, it always succeeds.

WRITE ENABLE Command Sequence	WRITE ENABLE Status Sequence
1. F6H	1. F6H
2. 0	2. 80H (or error code)
3. First byte of password	3. 0
4. Second byte of password	4. 0
5. Third byte of password	5. 0
6. Fourth byte of password	6. 0
7. Fifth byte of password	7. 0
8. Sixth byte of password	8. 0

F7H -- WRITE PROTECT

Prevents execution of commands that can alter disk data. These include WRITE (ClH), WRITE AND VERIFY (C9H), REFORMAT TRACK (C3H), and FORMAT (F3H), as well as the WRITE PROTECT command itself. WRITE PROTECT accepts an arbitrary 6-byte password in the third through eighth bytes of its command byte sequence. It returns the 02 error code if the disk is already write-protected or the 04 error code if it cannot read the disk.

WRITE PROTECT Command Sequence	WRITE PROTECT Status Sequence
1. F7H	1. F7H
2. 0	2. 80H (or error code)
3. First byte of password	3. 0
4. Second byte of password	4. 0
5. Third byte of password	5. 0
6. Fourth byte of password	6. 0
7. Fifth byte of password	7. 0
8. Sixth byte of password	8. 0

F8H -- READ ECC ERROR COUNTER

Returns the current value of the WDC's internal ECC error counter in the seventh and eighth status bytes, with the seventh status byte containing the low-order byte of the ECC error count and the eighth status byte containing the high-order byte of the error count. The ECC error counter tallies the number of times that the ECC circuitry recognized and corrected a correctable error. READ ECC ERROR COUNTER ignores the second through eighth bytes of the standard command sequence. It is always successful, and it concludes with the third through sixth status bytes cleared to 0.

F9H -- RESET ECC ERROR COUNTER

Clears the WDC internal ECC error counter to 0. RESET ECC ERROR COUNTER ignores the second through eighth bytes of the standard command sequence. It is always successful, and it concludes with the third through sixth status bytes cleared to 0.

FAH -- DOWNLOAD

Transfers 1792 bytes of data, assumed to be Z80A program code, into WDC memory locations 1000-16FFH. Z80A execution continues with the downloaded instruction that was loaded into location 1000H. DOWNLOAD ignores the second through eighth command bytes. It is always successful, and it concludes with the third through eighth status bytes cleared to 0.

FBH -- UPLOAD

Same as DOWNLOAD (FBH), except that the direction of transfer is from the WDC to system memory, and Z80A execution is not affected.

FCH --READ DISK SIZE

Returns the highest cylinder number in the third and fourth status byte with the number of surfaces in the fifth status byte.

10.5 WDC STATUS INFORMATION

Except for the EXECUTE DIAGNOSTIC (F0H) command, every WDC command returns eight bytes of result data, even though some commands do not use all eight of the status bytes. (EXECUTE DIAGNOSTIC is an exception because it does not have a result phase and it does not generate a result phase interrupt request.) The first status byte always contains the operation code for the command that was performed. The second status byte always contains 80H if the command was performed successfully, or an error code if an error prevented successful execution. Table 10-3 lists the error codes that appear in the second status byte. Table 10-4 shows which status bytes contain significant data for each WDC command. In this table, an X indicates that the status byte contains meaningful, variable information, and any other value indicates that the status byte is always set to that value.

Table 10-3. WDC Operation Status Code Assignments

Value	Meaning
80H	Successful command execution. No errors.
83H	Successful execution after performing error correction during a READ (C0H), VERIFY (C2H), or WRITE AND VERIFY (C9H) command.
02	Disk was write-protected during WRITE (C1H), REFORMAT TRACK (C3H), WRITE AND VERIFY (C9H), FORMAT (F3H), or WRITE PROTECT (F7H) command. Indicates wrong password during WRITE ENABLE (F6H) command.
03	Checksum error during READ (C0H), VERIFY (C2H), WRITE AND VERIFY (C9H), or READ RESERVED CYLINDER (F5H) command.
04	Cannot find designated sector during READ (C0H), WRITE (C1H), VERIFY (C2H), WRITE AND VERIFY (C9H), WRITE PROTECT (F7H), WRITE ENABLE (F6H), or READ RESERVED CYLINDER (F5H) command.
05	Data read back from disk did not match data that was written during WRITE AND VERIFY (C9H) command.
08	Programming error. Can indicate an illegal or unimplemented WDC command code, illegal command byte sequence, or other improper command request.
09	Drive fault during READ (C0H), WRITE (C1H), VERIFY (C2H), WRITE PROTECT (F7H), WRITE ENABLE (F6H), REFORMAT TRACK (C3H), WRITE AND VERIFY (C9H), FORMAT (F3H), or READ RESERVED CYLINDER (F5H) command.

Table 10-4. Status Bytes Set by WDC Commands

WDC Command	Code	Status Byte 1	Status Byte 2	Status Byte 3	Status Byte 4	Status Byte 5	Status Byte 6	Status Byte 7	Status Byte 8
READ	(C0H)	C0H	X	X	X	X	X	X	X
WRITE	(C1H)	C1H	X	X	X	X	X	X	X
VERIFY	(C2H)	C2H	X	X	X	X	X	X	X
FORMAT TRACK	(C3H)	C3H	X	X	X	X	0	0	0
WRITE AND VERIFY	(C9H)	C9H	X	X	X	X	X	X	X
RESET RETRY COUNT	(F1H)	F1H	80H	0	0	0	0	0	0
READ RETRY COUNT	(F2H)	F2H	80H	0	0	0	0	X	X
FORMAT	(F3H)	F3H	X	0	0	0	0	0	0
READ ERROR DATA	(F4H)	F4H	80H	0	0	0	0	0	0
READ RESERVED CYLINDER	(F5H)	F5H	X	0	0	0	0	02	0
WRITE ENABLE	(F6H)	F6H	X	0	0	0	0	0	0
WRITE PROTECT	(F7H)	F7H	X	0	0	0	0	0	0
READ ECC ERROR COUNT	(F8H)	F8H	80H	0	0	0	0	X	X
RESET ECC ERROR COUNT	(F9H)	F9H	80H	0	0	0	0	0	0
DOWNLOAD	(FAH)	FAH	80H	0	0	0	0	0	0
UPLOAD	(FBH)	FBH	80H	0	0	0	0	0	0
READ DISK SIZE	(FCH)	FCH	80H	X	X	X	0	0	0

Table 10-5. Winchester Disk Recording Format

Field Name	Size (bytes)	Data Value	Description
GAP 1	15	4EH	First field of every cylinder. Provides a buffer zone to allow for rotational speed variations.
SYNC 1	14	00	Phase Locked Oscillator (PLO) Synchronization Field. Allows the PLO to lock onto the data frequency.
HDR AM	1	AlH	Header Address Mark. Unique combination of clock and data bits that informs the WDC that header ID information follows. The AlH data pattern has a missing clock transition between its fourth and fifth bits.
	1	FEH	Distinguishes header address mark from data address mark, which instead contains FBH.
ID	4		Header ID field contains cylinder number, head, sector number (0-31), and a checksum, in that order. Each of these fields is one byte long.
GAP 2	3	4EH	Spacing between ID and data fields.
SYNC 2	14	00	Phase Locked Oscillator Synchronization Field. Allows the PLO to lock onto the data frequency.
DATA AM	1	AlH	Data Address Mark. Unique combination of clock and data bits that informs the WDC that the data field follows. The AlH data pattern has a missing clock transition between its fourth and fifth bits.
	1	FBH	Distinguishes data address mark from header address mark, which instead contains FEH.
DATA	256		Arbitrary data. Most significant byte of a word is stored first, followed by low-order byte.
CRC	3		ECC checksum. WDC hardware implements an ECC scheme that detects all burst errors of up to 23 bits and corrects all 8-bit burst errors.
GAP 3	3	4EH	Last field in sector. Provides spacing to beginning of following sector.

CHAPTER 11
MULTIPORT COMMUNICATIONS CONTROLLER

The Multiport Communications Controller (MCC) is mapped into the I/O space of the Wang PC and operates in an interrupt or polled environment only. It does not support DMA transfers. The MCC board supports three RS-232-C ports labeled 1, 2, and 3. Ports 1 and 2 support both asynchronous and synchronous communications. These two ports are controlled by a Z80A-SIO/2 chip. Port 3 supports asynchronous communications only. It is controlled by a Z80A-DART chip.

The MCC board is available in two models: PC-PM042 and PC-PM043. Port 3 of model PC-PM043 supports a secondary channel for split-speed applications.

11.1 MCC BOARD TO SYSTEM BOARD INTERFACE

The MCC board conforms to the standard bus interface specifications for all PC option boards (refer to Chapter 15). The data path on the MCC board Model PC-PM042 is only eight bits wide. Thus, the CPU can perform only single-byte transfers to and from the MCC board. The data path on Model PC-PM043 is 12 bits wide. This allows the status register to be read as a word, with the CPU ignoring D15-D12. All other I/O ports on Model PC-PM043 are accessed as bytes. The option identification for this board is 1FH.

11.2 INTERRUPTS

Interrupts generated by the MCC board can come from four sources. If Ring Indicator on Port 1 or 2 is active and either source is enabled, an interrupt will be generated. The Ring Indicator condition can be checked by reading the status register and interrogating the appropriate bit. The other sources are the Z80A-SIO/2 and Z80A-DART. These chips generate an interrupt if one of the following conditions is enabled: a receive character is available, the transmit buffer is empty, or a programmed special status change has occurred. The Ring Indicator signal on Port 3 ties directly into the Z80A-DART chip and can be programmed as one of the special status conditions that cause an interrupt.

The Z80A-SIO/2 and Z80A-DART are daisy-chained to form a priority structure for the interrupts they generate. Software must issue an OUT to the appropriate SIO or DART registers to clear any interrupts, since the system does not generate an RETI sequence to the MCC board.

Software must program the interrupt request level that the MCC board uses when generating interrupts to the CPU. The MCC board can be programmed to permit interrupts on Levels 2, 3, and 4. The default level on power-up is 2.

11.3 COMMUNICATIONS FEATURES

The three ports use RS-232C level signals via a standard 25-position D-sub connector on each port. The transmit signal from each of the ports has been conditioned so that each port is capable of driving up to 2000 feet of cable.

Ports 1 and 2 support both asynchronous and synchronous communications, including bisynchronous and HDLC/SDLC protocols. In synchronous operation, Ports 1 and 2 can operate at baud rates from 1200 bps to 19200 bps, using an on-board COM8116 dual-baud-rate generator. When configured to use an internal clock, Ports 1 and 2 can also use the external clock as the baud rate generator for synchronous operation. In asynchronous mode, Ports 1 and 2 can operate at rates from 50 bps to 19.2 kbps.

Port 3 supports asynchronous communications only. It can operate at baud rates from 50 bps to 19.2 kbps. The primary and secondary channels on Model PC-PM043 can operate at separate speeds. The secondary channel of Port 3 on Model PC-PM043 is sometimes referred to as Port 4. Because it is not actually a separate port, this document refers to it as the secondary channel of Port 3 or as Port 3, Channel B.

11.4 BAUD RATE GENERATION

Each channel (or port) has its own external baud rate generator (BRG). The output of each baud rate generator is 16 times the actual baud rate. Because asynchronous communications uses this times-16 clock directly, the BRG frequency is used directly by each channel when operating asynchronously. However, synchronous communications use a times-1 clock and therefore must either divide the actual BRG output frequency by 16 or set up the BRG to output a frequency 1/16th the frequency it would output in asynchronous mode. The MCC board uses the latter method. The system configures each BRG by writing to the appropriate I/O port for each channel. Refer to Section 11.5, MCC Board I/O Ports, for the addresses for these I/O ports.

Table 11-1 shows the relationships between the data written and the baud rate.

Table 11-1. Relations of Data to Baud Rates

Data Written				Asynchronous Model Baud Rates For	Synchronous Mode Baud Rates (Internal Clock)
D3	D2	D1	D0	Ports 1, 2, and 3	for Ports 1 and 2 Only
0	0	0	0	50	
0	0	0	1	75	1200
0	0	1	0	110	
0	0	1	1	134.5	
0	1	0	0	150	2400
0	1	0	1	300	4800
0	1	1	0	600	9600
0	1	1	1	1200	19.20k
1	0	0	0	1800	
1	0	0	1	2000	
1	0	1	0	2400	
1	0	1	1	3600	
1	1	0	0	4800	
1	1	0	1	7200	
1	1	1	0	9600	
1	1	1	1	19200	

11.5 MCC BOARD I/O PORTS

The MCC board decodes 16 I/O port addresses used by the system CPU to access the various control and status registers, as well as the registers within the two communication chips. Table 11-2 lists the I/O address and the device accessed. An S in the System Address column represents the number of the slot in which the MCC board is located. A description of each of the port's functions follows the table.

Table 11-2. I/O Port Addresses

System I/O Address	I/O Port Designation	Port Identification	I/O Operation
1S00H	I/O Port 0	Port 1 Data Registers	Read/Write
1S02H	I/O Port 1	Port 2 Data Registers	Read/Write
1S04H	I/O Port 2	Port 1 Control Registers	Write
1S06H	I/O Port 3	Port 2 Control Registers	Write
1S08H	I/O Port 4	Port 3 Channel A Data Registers	Read/Write
1S0AH	I/O Port 5	Port 3 Channel B Data Registers (for PC-PM043)	Read/Write
1S0CH	I/O Port 6	Port 3 Channel A Control Registers	Write
1S0EH	I/O Port 7	Port 3 Channel B Control Registers (for PC-PM043)	Write
1S10H	I/O Port 8	Board Status Register	Read
1S12H	I/O Port 9	Port 1 Baud Rate Register	Write
1S14H	I/O Port A	Port 2 Baud Rate Register	Write
1S16H	I/O Port B	Ports 1 and 2 Function Register	Write
1S18H	I/O Port C	Port 3 Channel A Baud Rate Register	Write
1S1AH	I/O Port D	Port 3 Channel B Baud Rate Register (for PC-PM043)	Write
1SFCH	I/O Port E	Software Reset for MCC Board	Write
1SFEH	I/O Port F	Option Identification	Read

Port 0 (Port 1 Data Registers): An IN to Port 0 generates a data transfer from the SIO Channel A Receive Data Buffer to the CPU on Bus Data Bits D7-D0. An OUT to Port 0 generates a data transfer from the CPU to the SIO Channel A Transmit Data Buffer on Bus Data Bits D7-D0.

Port 1 (Port 2 Data Registers): An IN to Port 1 generates a data transfer from the SIO Channel B Receive Data Buffer to the CPU on Bus Data Bits D7-D0. An OUT to Port 1 generates a data transfer to the SIO Channel B Transmit Data Buffer from the CPU on Bus Data Bits D7-D0.

Port 2 (Port 1 Control Registers): An OUT to Port 2 generates a control byte to the register specified by Write Register 0 in Channel A of the SIO.

Port 3 (Port 2 Control Registers): An OUT to Port 3 generates a control byte to the register specified by Write Register 0 in Channel B of the SIO.

Port 4 (Port 3 Channel A Data Registers): An IN to Port 4 generates a data transfer from the DART Channel A Receive Data Buffer to the CPU on Bus Data Bits D7-D0. An OUT to Port 4 generates a data transfer to the DART Channel A Transmit Data Buffer from the CPU on Bus Data Bits D7-D0.

Port 5 (Port 3 Channel B Data Registers): An IN to Port 5 generates a data transfer from the DART Channel B Receive Data Buffer to the CPU on Bus Data Bits D7-D0. An OUT to Port 5 generates a data transfer to the DART Channel B Transmit Data Buffer from the CPU on Bus Data Bits D7-D0.

Port 6 (Port 3 Channel A Control Registers): An OUT to Port 6 generates a control byte to the register specified by Write Register 0 in Channel A of the DART.

Port 7 (Port 3 Channel B Control Registers): An OUT to Port 7 generates a control byte to the register specified by Write Register 0 in Channel B of the DART.

Port 8 (Status Register): An IN to Port 8 reads the status buffer to Bus Data Bits D7-D0 on Model PC-PM042 and D11-D0 on Model PC-PM043. Thus, Model PC-PM043 requires a word read of Port 8 to receive all the status information. An OUT to Port 8 is an invalid operation. Table 11-3 defines the Status Register bits.

Table 11-3. Status Register Bits

Bit Value	Meaning
D0=1	Port 1 (SIO Channel A) /Wait-Ready output active.
D0=0	Port 1 (SIO Channel A) /Wait-Ready output inactive.
D1=1	Port 2 (SIO Channel B) /Wait-Ready output active.
D1=0	Port 2 (SIO Channel B) /Wait-Ready output inactive.
D2=1	Port 1 (SIO Channel A) Data Set Ready (/DSR) active.
D2=0	Port 1 (SIO Channel A) Data Set Ready (/DSR) inactive.
D3=1	Port 2 (SIO Channel B) Data Set Ready (/DSR) active.
D3=0	Port 2 (SIO Channel B) Data Set Ready (/DSR) inactive.
D4=1	RI signal of Port 1 (SIO Channel A) went active and was enabled.
D4=0	RI signal of Port 1 (SIO Channel A) inactive or disabled.
D5=1	RI signal of Port 2 (SIO Channel B) went active and was enabled.
D5=0	RI signal of Port 2 (SIO Channel B) inactive or disabled.
D6=1	Port 3 (DART Channel A) /Wait-Ready output active.
D6=0	Port 3 (DART Channel A) /Wait-Ready output inactive.
D7=1	Port 3 (DART Channel A) Data Set Ready (/DSR) active.
D7=0	Port 3 (DART Channel A) Data Set Ready (/DSR) inactive.
D8=1	Port 3 (DART Channel B) /Wait-Ready output active (PC-PM043).
D8=0	Port 3 (DART Channel B) /Wait-Ready output inactive (PC-PM043).
D9 to D11=0	Indicates Model PC-PM043 rather than Model PC-PM042.

Port 9 (Port 1 Baud Rate Register): An OUT to Port 9 configures the baud rate register for SIO Channel A. Only Data Bits D3-D0 are used (refer to Table 11-1, Relations of Data to Baud Rates).

Port A (Port 2 Baud Rate Register): An OUT to Port 9 configures the baud rate register for SIO Channel B. Only Data Bits D3-D0 are used (refer to Table 11-1, Relations of Data to Baud Rates).

Port B (Ports 1 and 2 Function Register): An OUT to Port B writes Data Bits D7-D0 to the function register. Table 11-4 defines the Function Register bits.

Table 11-4. Function Register Bits

Bit Value	Meaning
D0=0	Port 1 receive and transmit clocks derived from on-board BRG.
D0=1	Port 1 receive and transmit clocks derived externally from RS-232C Pins 15 and 17, respectively.
D1=0	NRZ data format selected for Port 1.
D1=1	NRZI data format selected for Port 1.
D2=0	Disable Port 1 ring indicator interrupts.
D2=1	Enable Port 1 ring indicator interrupts.
D3=0	Port 2 receive and transmit clocks derived from on-board BRG.
D3=1	Port 2 receive and transmit clocks derived externally from RS-232C Pins 15 and 17, respectively.
D4=0	NRZ data format selected for Port 2.
D4=1	NRZI data format selected for Port 2.
D5=0	Disable Port 2 Ring Indicator interrupts.
D5=1	Enable Port 2 Ring Indicator interrupts.

D6-D7	Interrupt Request Level to System CPU Select:
	D6=0 D7=0 Interrupt Request Level 2
	D6=1 D7=0 Interrupt Request Level 3
	D6=0 D7=1 Interrupt Request Level 4
	D6=1 D7=1 Invalid Selection

Port C (Port 3 Channel A Baud Rate Register): An OUT to Port C configures the baud rate register for DART Channel A. Only Data Bits D3-D0 are used (refer to Table 11-1, Relations of Data to Baud Rates).

Port D (Port 3 Channel B Baud Rate Register): An OUT to Port C configures the baud rate register for DART Channel B (PC-PM043 only). Only Data Bits D3-D0 are used (refer to Table 11-1, Relations of Data to Baud Rates).

Port E (Software Reset): An OUT to Port E resets the board.

Port F (Option Identification): Reading this port returns a 1FH if there is no interrupt pending from this board and a 9FH if the MCC board has issued an interrupt request.

11.6 SIGNALS USED BY THE MCC BOARD

This section lists the signals used by the RS-232-C connectors of the MCC board. Table 11-5 lists the signals used on the RS-232-C connectors of Ports 1 and 2. Table 11-6 lists the signals used on the RS-232-C connectors of Port 3 on Model PC-PM042. Table 11-7 lists the signals used on the RS-232-C connectors of Port 3 on Model PC-PM043.

Table 11.5. Signals Used on RS-232-C Connectors of Ports 1 and 2

Pin Number	Circuit	Description
1	AA	Protective Ground
2	BA	Transmit Data
3	BB	Receive Data
4	CA	Request To Send
5	CB	Clear To Send
6	CC	Data Set Ready
7	AB	Signal Ground
8	CF	Data Carrier Detect
11		Output of BRG -- used for Wang functions only
15	DB	Transmit Clock
17	DD	Receive Clock
20	CD	Data Terminal Ready
22	CE	Ring Indicator

Table 11-6. Signals Used on RS-232-C Connectors of Port 3 (Model PC-PM042)

Pin Number	Circuit	Description
1	AA	Protective Ground
2	BA	Transmit Data
3	BB	Receive Data
4	CA	Request To Send
5	CB	Clear To Send
6	CC	Data Set Ready
7	AB	Signal Ground
8	CF	Data Carrier Detect
20	CD	Data Terminal Ready
22	CE	Ring Indicator

Table 11-7. Signals Used on RS-232-C Connectors of Port 3 (Model PC-PM043)

Pin Number	Circuit	Description
1	AA	Protective Ground
2	BA	Transmit Data
3	BB	Receive Data
4	CA	Request To Send
5	CB	Clear To Send
6	CC	Data Set Ready
7	AB	Signal Ground
8	CF	Data Carrier Detect
12	SCF	Secondary Data Carrier Detect
13	SCB	Secondary Clear To Send
14	SBA	Secondary Transmit Data
16	SBB	Secondary Receive Data
19	SCA	Secondary Request To Send
20	CD	Data Terminal Ready
22	CE	Ring Indicator .

11.7 DIAGNOSTIC LOOPBACK CONNECTOR

The MCC board uses a special loopback connector for diagnostics. Figure 11-1 is a diagram of the connector showing all pin connections.

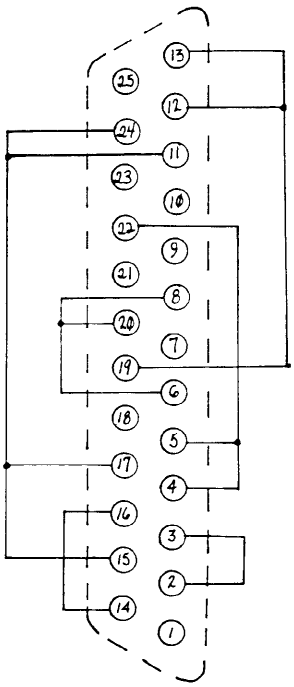


Figure 11-1. Diagnostic Loopback Connector

CHAPTER 12
TEXT/IMAGE/GRAPHICS CONTROLLER

The Wang PC Text/Image/Graphics (TIG) Controller board controls a 12-inch monochrome monitor. Text and bit-map video can be superimposed or displayed independently. Implementation uses two NEC uPD7220 Graphic Display Controllers (GDCs) -- one for text and one for graphics. All memory is accessed through the GDCs, which arbitrate memory accesses and perform memory refreshes. None of the memory is contained in the Wang PC memory address space. Bulk data transfers are by Direct Memory Access (DMA) via GDCs. Command and status communication with the GDCs is achieved through registers located in the I/O address space. Many features and configuration options are also available to the programmer through internal GDC parameter registers.

To program the TIG board, the developer should consult the following documents:

NEC 1982 Catalogue. NEC Electronics U.S.A., Inc., 1982.

NEC uPC 7220/GDC Design Manual. Version 3. NEC Electronics U.S.A., Inc., 1982

Intel 82720/GDC Applications Manual. Intel Corporation, 1983.

12.1 FEATURES AND SPECIFICATIONS

The following items are important features of the TIG board:

- 128 KB (1 million bits) bit-map memory, which is adequate for a full page at 100 pixels/inch resolution.

- 8 KB of character memory capable of holding 4K characters plus display attributes.

- Two complete loadable and readable fonts, each with a full complement of 256 characters.

- Programmable character height up to 32 lines high.

- Two text areas and two image areas, each of which can be scrolled independently.
- Programmable truth-tables for combining text and image. Three software-definable display attribute bits plus one programmable bit to control blinking attributes. Intensify capability is included.
- Hardware subscript, superscript, and underline capability.
- Programmable memory organization for both text and image memory, allowing both portrait and landscape format.
- Bulk transfer of data between main memory and rectangular areas of either text or image memory by DMA.
- Graphics capabilities including drawing of lines, arcs, circles, and rectangles. Tessellation of rectangles with a definable 8x8 pattern.
- The ability to emulate the display capability of the Wang PC medium-resolution (character) controller.

Table 12-1 lists the basic specifications of the TIG board.

Table 12-1. Basic TIG Board Specifications

Parameter	Value
Vertical Rate	80 fields per second interlaced (effectively 40 frames per second)
Horizontal Rate	25.40 KHz (39.37 us)
Resolution	800 horizontal x 600 vertical
Total Number of Pixels	480,000
Character Display	80 columns x 25 or 33 rows (others are possible)
Character Word	10 bits
Graphics Word	16 bits
Pixel Clock	26.416 MHz (37.856 ns)
Character Clock (CCLK)	5.283 MHz (189.3 ns)
Graphics Clock (GCLK)	3.30 MHz (302.8 ns)
Character Memory	8 KB (4 2K X 8 static RAM) 4 KB ASCII characters and 4 KB attributes
Font Memory	16K x 10, 512 matrices 10 bytes wide by (up to) 32 bytes high (10 16K x 1 static RAM)
Bit-map Memory	128 KB (16 64K x 1 dynamic RAM) full 8.5 x 11 page, one bit plane

12.2 SYSTEM BLOCK DIAGRAMS

Figures 12-1 and 12-2 are block diagrams of the TIG Controller circuitry. Figure 12-1 shows the graphics, clock, and system interface portion. Figure 12-2 shows the character portion. The definitions of the hardware signals are given in Table 12-2.

Table 12-2. Hardware Signals

Signal Name	Definition
PCLK	Pixel clock
CCLK	Character clock
GCLK	Graphics clock
CHLD	Character word load pulse
GHLD	Graphics word load pulse
CALE	Character frame/font memory address latch pulse
/CDBIN	Character data bus in (into CGDC)
/GDBIN	Graphics data bus in (into GGDC)
/COE	Character frame memory output enable

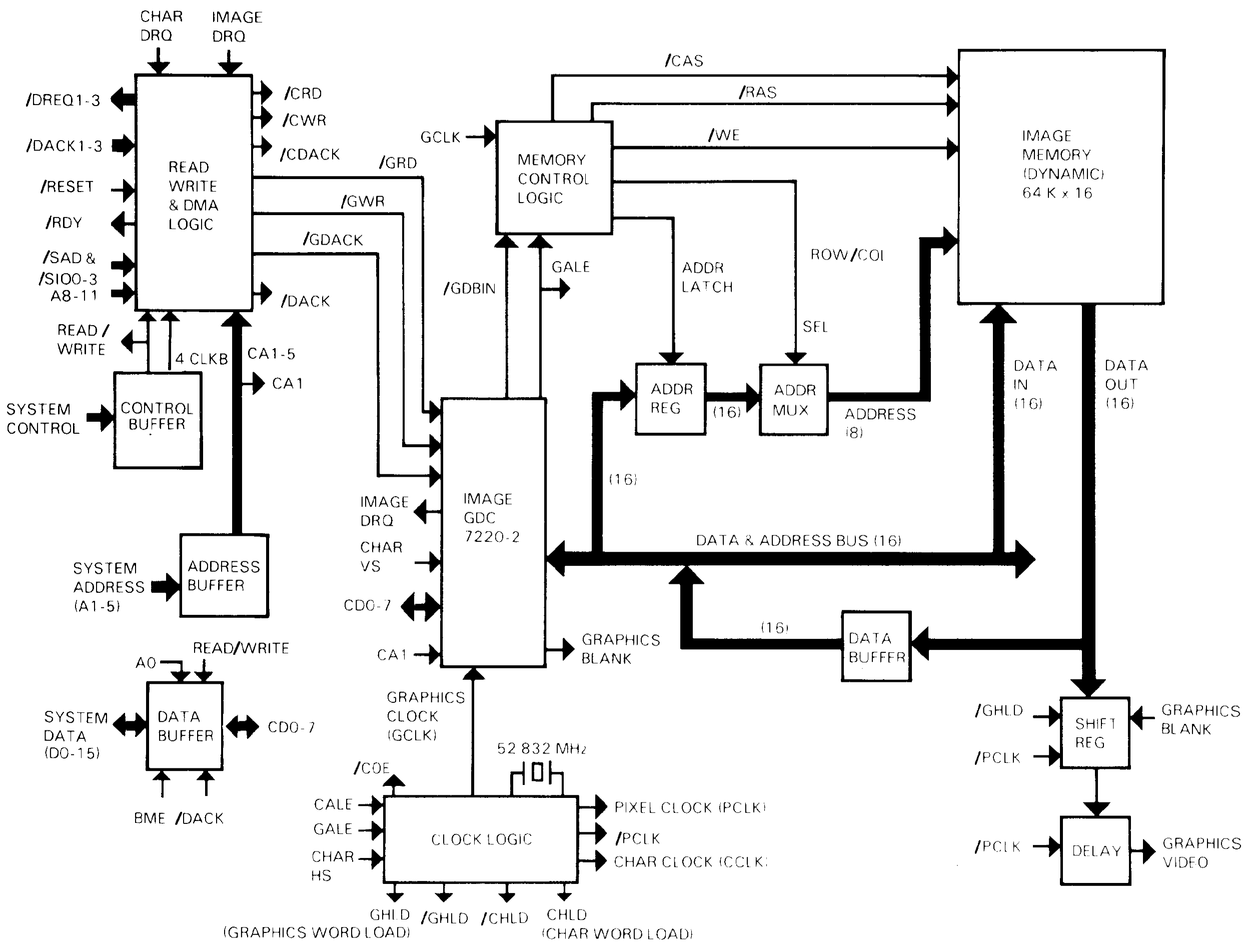


Figure 12-1. Block Diagram: Graphics, Clock, and System Interface.

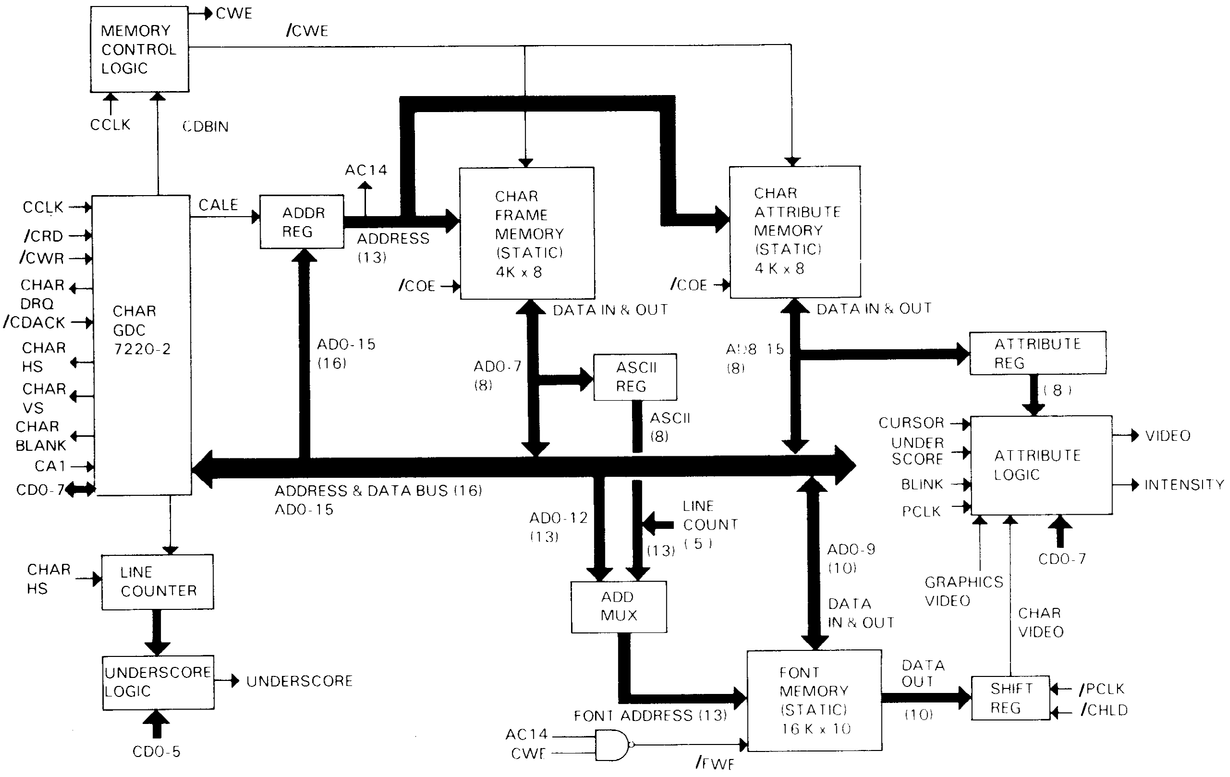


Figure 12-2. Block Diagram: Character Portion

12.3 DEVICE REGISTERS

The TIG Controller contains 28 directly accessible device registers. There are four registers for communicating with each GDC, a 16-register lookup table for display attributes, and two registers for parameters. In addition, each of the two NEC uPD7220 GDC chips contains internal registers that are accessed through the command, status, and parameter ports of the GDC. All registers are a byte wide.

The two refresh memories -- a bit-map for the image and a frame buffer for text -- are not directly available on the processor bus. Small transfers to or from the refresh memories can be performed through the command and parameter ports of the GDC. Bulk data transfers usually take place via DMA to main memory, then through the corresponding GDC device to either the bit-map or the frame buffer. Font tables are also loaded through the character GDC by program transfer or by DMA.

The base address of the controller registers depends upon in which slot the TIG board is installed in the PC chassis. The system must interrogate each slot at start-up to determine what it contains. Note that the I/O address space is distinct from the memory address space. Table 12-3 gives the base address, read/write mode, and function of the TIG board registers. The following sections contain more detailed information about specific registers.

Table 12-3. TIG Board Registers

Address (n = slot#)	Mode	Description
1n00 thru 1n1E	WRITE	Attribute Table
1n20	READ	Character GDC: Status Register
	WRITE	Character GDC: Parameter into FIFO
1n22	READ	Character GDC: Read Data from FIFO
	WRITE	Character GDC: Command into FIFO
1n24	READ	Image GDC: Status Register
	WRITE	Image GDC: Parameter into FIFO
1n26	READ	Image GDC: Read Data from FIFO
	WRITE	Image GDC: Command into FIFO
1n28	WRITE	Underline position register
1n2A	WRITE	DMA channel select and GDC select
1nFC	WRITE	Software board reset
1nFE	READ	Device ID code return.

All registers appear at even (i.e., low-order) addresses. The address lines are not fully decoded. References to addresses in the range 1n00 through 1nFF, other than to those defined, are likely to have undefined effects.

12.3.1 Attribute Table

The attribute table is made of byte-wide registers that are loaded by the CPU with truth tables describing how character and image video are to be combined and displayed. The 8-bit register allows a maximum of 256 ways to display characters and images. The 16-byte attribute table allows any 16 attributes to be available at a given time. The blink feature automatically alternates between using the odd and even bytes of a register pair.

Each character in the frame buffer has eight attribute bits (AT7 to AT0) associated with it. Four of the attribute bits (AT5 to AT2) are used to select one of the 16 registers for that character. Attribute Bits AT5, AT4, and AT3 cause a static selection, but Bit AT2 is ANDed with the character blink signal. Therefore, if AT2 is set, the attribute register alternates between the even and odd numbered register selected by AT5 to AT3. As a result, the generated display blinks accordingly.

Bits 0 through 3 of the byte in the attribute table indicate when a normal intensity pixel should be displayed for each possible combination of image and text. Bits 4 through 7 indicate when a dimmer-than-normal pixel should be displayed. When a normal and a dim pixel are both indicated, a bright pixel results.

If Bits 1, 2 and 3 are set and Bit 0 is clear (refer to Figure 12-3), the logical OR of the image and text in normal intensity is displayed. Similarly, if Bits 4, 5 and 6 are clear and Bit 7 is set, a dimmer-than-normal dot will be displayed when both image and text are present. Together, the register contents just described would display a normal intensity dot where either image or text appears and a high intensity dot where both image and text appear.

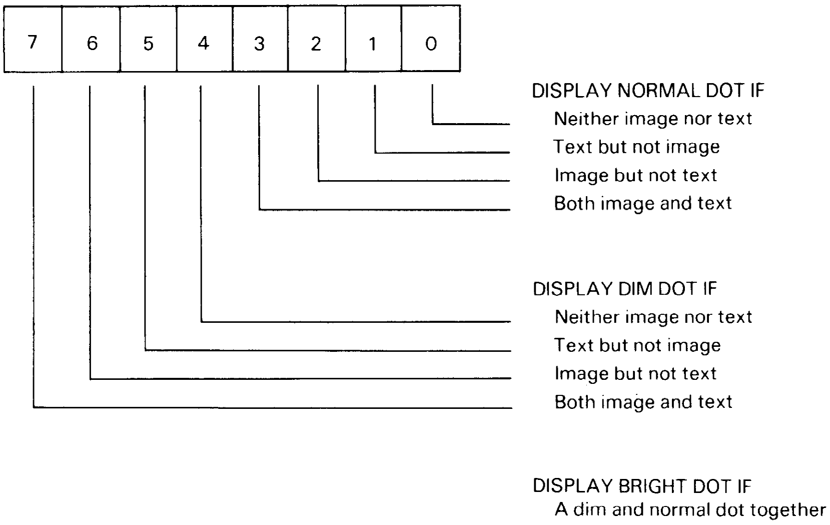


Figure 12-3. Attribute Byte Example

Table 12-4 shows some of the possible video effects and the attribute table value that obtains them.

Table 12-4. Video Attribute Examples

Value	Result
00H	Completely blank
0AH	White text on dark background
AAH	Bright text on dark background
05H	Black text on white background (reverse video)
55H	Black text on bright white background
5AH	White text on dim background
FAH	Bright text on dim background
50H	Black text on dim background
03H	Normal black image on white background
C3H	Dim image on white background
30H	Black image on dim white background
0CH	White image on black background (negative)
01H	Black text and black image on white background
1AH	White text over black image on dim white background

The default table values that permit emulation of the PC medium-resolution display attributes are shown in Table 12-5. Images are completely disabled. The character attribute bits are as follows:

AT5:	Bold (intensify)
AT4:	Blank
AT3:	Reverse Video
AT2:	Blink

Table 12-5. Medium-resolution Display Emulation

Attribute Table I/O Address	Attribute Bits (AT#)				Register Numbers	Default Table Entry (for Wang PC Emulation)
	5	4	3	2		
1n00	0	0	0	0	0	0A
1n02	0	0	0	1	1	00
1n04	0	0	1	0	2	05
1n06	0	0	1	1	3	0F
1n08	0	1	0	0	4	00
1n0A	0	1	0	1	5	00
1n0C	0	1	1	0	6	0F
1n0E	0	1	1	1	7	0F
1n10	1	0	0	0	8	AA
1n12	1	0	0	1	9	00
1n14	1	0	1	0	10	55
1n16	1	0	1	1	11	FF
1n18	1	1	0	0	12	00
1n1A	1	1	0	1	13	00
1n1C	1	1	1	0	14	FF
1n1E	1	1	1	1	15	FF

12.3.2 Underline Position Register

The Underline Position Register contains five bits that program the position of the character underline. The register must be set to correspond to the desired position of the underline based on the specific character font and character height being used. The underline is one scan line high and the position range is from 0 to 31. Figure 12-4 illustrates the Underline Position Register.

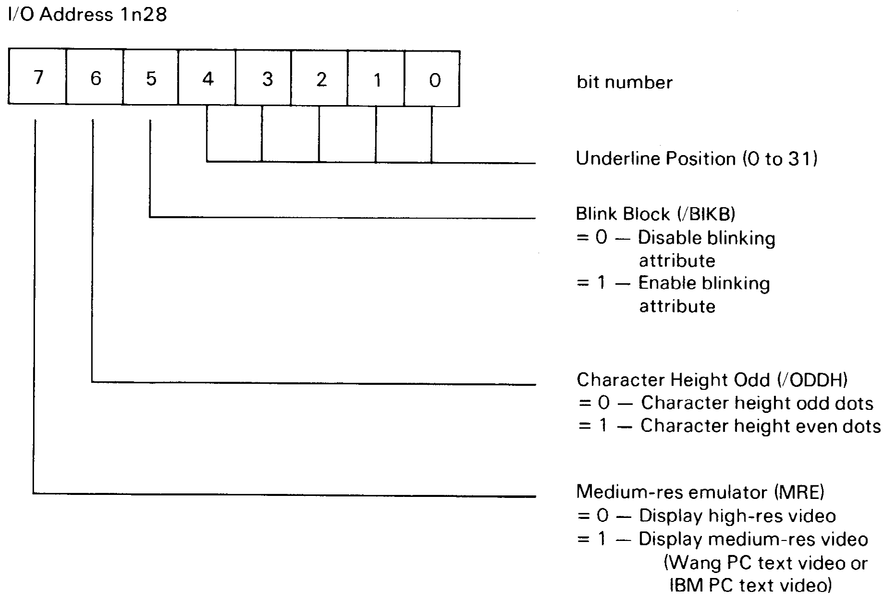

I/O Address 1n28
7
6
5
4
3
2
1
0
bit number
Underline Position (0 to 31)
Blink Block (/BIKB)
= 0 — Disable blinking attribute
= 1 — Enable blinking attribute
Character Height Odd (/ODDH)
= 0 — Character height odd dots
= 1 — Character height even dots
Medium-res emulator (MRE)
= 0 — Display high-res video
= 1 — Display medium-res video (Wang PC text video or IBM PC text video)

Figure 12-4. The Underline Position Register

When enabled, the underline becomes a part of the character and is operated on by the attribute truth tables in the same way that the character dots are; i.e., it is blanked, reversed, intensified and blinked along with the character. However, an underlined character is not shifted up or shifted down when superscript or subscript is enabled.

Bit 5 of the Underline Position Register contains a bit to disable the blinking attribute feature. With Bit 5 reset to 0, attribute Bit 0 is not intermittent and behaves exactly like the other three attribute bits. This allows 16 distinct (non-blinking) character states rather than eight pairs that can alternate.

If the character height is programmed to be an odd number, Bit 6 must be reset to 0. If the character height is even, Bit 6 must be set to 1.

If Bit 7 is set to 0, high-resolution video is sent to the monitor. If Bit 7 is set to 1, medium-resolution video is sent to the monitor, provided the Wang PC 8691 board is also installed. Both hardware and software resets clear all eight bits to 0. Therefore, after the reset, the monitor displays high-resolution video regardless of whether the 8691 board is installed.

12.3.3 DMA Select Register

The DMA Select Register performs two functions: it selects among the three DMA channels available on the PC; it also selects which of the two GDCs is able to perform DMA transfers. Figure 12-5 illustrates the DMA Select Register.

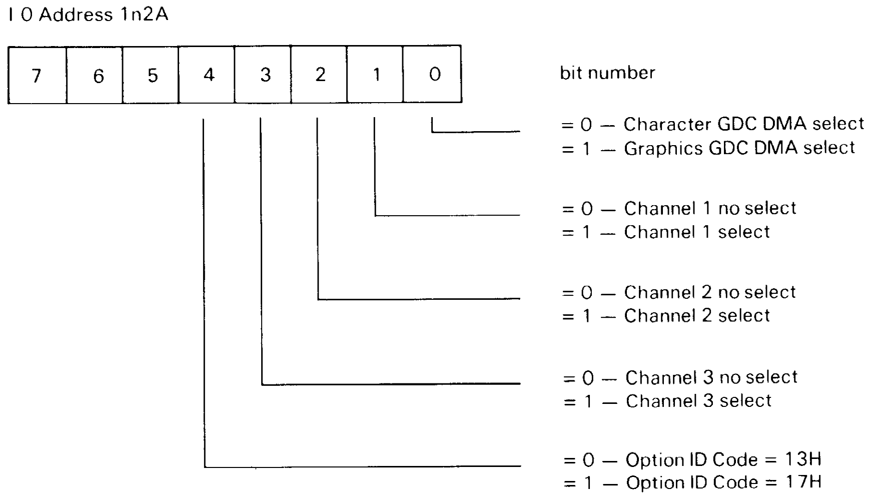

I O Address 1n2A
7
6
5
4
3
2
1
0
bit number
= 0 — Character GDC DMA select
= 1 — Graphics GDC DMA select
= 0 — Channel 1 no select
= 1 — Channel 1 select
= 0 — Channel 2 no select
= 1 — Channel 2 select
= 0 — Channel 3 no select
= 1 — Channel 3 select
= 0 — Option ID Code = 13H
= 1 — Option ID Code = 17H

Figure 12-5. The DMA Select Register

Bit 0 selects the character or image GDC for eligibility to make DMA transfers. If the bit is clear (0), the character GDC is enabled. If the bit is set (1), the image GDC is enabled. The bit may not be changed while the channel is being used.

Bits 1, 2, and 3 of this register select DMA Channels 1, 2, and 3, respectively. One of these bits must be set before either GDC can make a DMA request. No more than one of the channel select bits should be set at any time. Care must be taken so that the channel selected does not conflict with any other device that might also request DMA using the same channel.

Setting Bit 4 to 1 alters the ID code to 17H. After an ID code read or a hardware or software reset, the ID code changes back to 13H.

There are many additional registers internal to the uPD7220 controllers. For further information, refer to the NEC GCD documents. Refer to Section 12.5 for specific assumptions and defaults for initializing and using the uPD7220s in the TIG board.

12.3.4 Software Board-Reset Register

An I/O write command to Port Address 1nFC with arbitrary data resets the I/O registers on the boards, with the following effects:

- The underline position register is reset to 0
- The DMA select register is reset to 0
- All three DMA request lines are deactivated
- The ID code is reset to 13H

12.3.5 Device ID Code Return Register

An I/O read command to Port Address 1nFE returns with the board ID code. The board ID code is set to 13H if 1n2AH Bit 4 is 0. The code is 17H if 1n2AH Bit 4 is 1.

12.4 MEMORY ORGANIZATION

This section describes memory organization for the TIG board. Section 12.4.1 describes frame buffer memory. Section 12.4.2 describes font memory. Section 12.4.3 describes graphics memory.

12.4.1 Frame Buffer Memory

The frame buffer is sufficient to contain 4096 16-bit characters. The frame buffer is addressed through the character GDC at Frame Buffer (word) Addresses 0000H through 0FFFH and again at 1000H through 1FFFH. This method of addressing allows the screen display to begin before the end of the frame buffer at 0FFFH and to wrap to the beginning of the frame buffer as addressed at 1000H. Thus, vertical scrolling capability is provided in excess of the capacity of the frame buffer itself.

Each character is represented by 16 bits. The low-order byte is the character code. All 256 possible characters are available. The high-order byte contains the attributes for that character. Certain of the attributes are fixed in their meaning while others are programmable through the attribute look-up tables (refer to Section 12.3.1).

Figure 12-6 illustrates the full 16-bit character word.

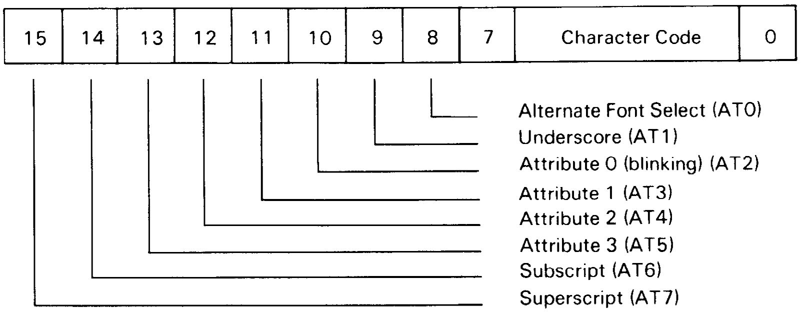


Figure 12-6. 16-bit Character Word

The organization of the buffer is programmable through uPD7220 to be any height or width larger than the screen display. The screen display can be scrolled horizontally and vertically to view any portion of the character memory.

12.4.2 Font Memory

The font memory contains the dot matrix character representations for two full 256-character fonts. One bit (AT0) of the character attribute byte operates as a font select bit. The character matrices are downloaded from the CPU. In medium-resolution controller emulation, the second font is used to produce characters with an overbar. At other times, the second font can be used for other characters and special symbols of any kind.

The character matrices are 10 bits wide by 32 bits high. However, since the height of the characters is programmable, the full height of the matrix is not necessarily used. In particular, for the standard 25-line display, a character height of 24 displayable bits is used. The standard medium-resolution 10 x 12 font can be used by either duplicating each line in the matrix or leaving alternate lines blank.

The subscript and superscript attributes allow the position of the character font matrix, as displayed on the monitor's character row, to be shifted down by two or up by four pixels. When the subscript bit is set to 1 and the superscript bit to 0, the matrix is displayed from its Row 0. The result is a subscripted character, two pixels lower than a normally positioned one. When both bits are 0, the matrix is displayed from its Row 2. The result is a normally positioned character, two pixels higher than a subscripted one. When the subscript bit is set to 0 and the superscript bit to 1, the matrix is displayed starting from its Row 4. The result is a superscripted character, two pixels higher than a normally positioned one. When both bits are 1, the matrix is displayed from its Row 6. The result is a doubly superscripted character, two pixels higher than a superscripted one.

The shifted rows wrap around within the matrix when the characters are programmed to the full height of the 32-row matrix. Therefore, the offset must be accommodated when the matrix is loaded.

The font matrices appear in the address space of the character GDC beginning at Word Address 4000H and continuing through 7FFFH. Each character uses 0020H addresses, one per row of the matrix. The pixels for each row appear in the ten least significant bits of the word, Bits 9 through 0. Bit 9 is the furthest left bit of the character. The first byte of the character matrix (the top row of a normal character) is displayed from Relative Address 2. If the character requires rows beyond its last address, the hardware wraps around to use Addresses 0000H and 0001H.

12.4.3 Graphics Memory

The graphics memory consists of 64K of 16-bit words. It contains over one million pixels. The pixels are mapped to an 800-wide by 600-high display on the CRT. The memory is addressed through the graphics GDC at Addresses 0000H through FFFFH. One GDC memory address corresponds to one 16-bit word.

Pixels of a word are displayed on the CRT in bit ordinal sequence, that is, from the least significant bit (Bit 0) displayed on the left through the most significant bit (Bit 15) displayed on the right.

12.5 PROGRAMMING THE TIG BOARD

This section provides information on programming the GCDs on the TIG board.

12.5.1 DMA Operations

Bulk data transfers to and from the character frame buffer and the image bit map are best accomplished by DMA transfers. Before a transfer is begun, the parameters must be properly set up on both the 8086 and GDC.

The 8086 performs a linear transfer. The 8086 operation requires a setup of a starting address and a byte count for the DMA channel that is used. The GDC performs an area transfer and generally requires a starting address and two dimensions.

The controller board does not generate an interrupt. If an interrupt mode is desired, the interrupt capability of the DMA controller in the Wang PC must be used. The DMA controller must be properly enabled and, when the interrupt occurs, the DMA channels must each be examined to determine which channel caused the interrupt.

CAUTION:
The byte count between the GDC and the DMA controller must be properly coordinated. If the GDC reaches the end of the transfer before the DMA controller, the GCD ceases to make requests. Therefore, the DMA controller never reaches the terminal count, and an interrupt never occurs. The whole process could hang. (The opposite condition, where the DMA byte count runs out early, is not as dangerous and can even be useful sometimes.)

The character and image GDCs share the DMA facility of the board. They cannot perform DMA transfers simultaneously. However, a DMA transfer can be prepared, i.e., the parameters can be loaded into the GDC, even if the DMA hardware is being used by the other GDC. The transfer can be initiated once the other GDC is done.

DMA transfers can be performed either fast (Flash mode) or slow (Flashless mode). In slow mode, the DMA transfers are restricted to times when the display is retracing. Memory accesses at that time do not disturb the appearance of the display. This is the recommended mode for relatively small transfers.

In fast mode, the DMAs are not restricted and take place approximately 25 times faster than in the slow mode. However, the memory accesses conflict with the screen refresh and disturb the appearance of the display. Nevertheless, this is the recommended mode for large, e.g., full-screen, transfers. To avoid bad-looking displays, the screen should be blanked, a fast transfer performed, and then the screen unblanked.

The two GDCs operate at different clock rates. The clock rate also affects the speed at which the GCDs can accept commands and data, including DMA data. Table 12-6 shows the approximate transfer rates.

Table 12-6. Estimated Maximum DMA Transfer Rates

Condition	Character GDC	Image GDC
Maximum DMA rate, us/byte	2.75	2.75
Max DMA bytes/field - fast	3,400	3,150
Max DMA bytes/field - slow	150	140
Max DMA bytes/sec - fast	270,000	250,000
Max DMA bytes/sec - slow	12,000	11,000
Full screen DMA - fast	15 msec	240 msec
Full screen DMS - slow	330 msec	5.5 sec

Since the DMA channel number is programmable, in some applications one DMA channel can be allocated to the character GDC and another to the image GDC. However, it is not clear what benefit this would provide.

I/O data transfers can be programmed instead of using DMA. Programmed transfers may be done on one GDC simultaneously with DMA transfers on the other. For small amounts of data, programmed transfers are more efficient than DMA transfers. Certain applications, such as character-at-a-time I/O to the text memory, might do programmed I/O all the time and skip DMA entirely.

Due to GDC chip power-up restrictions, undesired DMA requests are occasionally issued by the chip. In order to clear these unexpected requests and to perform proper DMA operations, software commands should be sent out in the following sequence (the order of Steps 3 and 4 is interchangeable):

1. Reset both character GDC and graphics GDC.

2. Issue I/O board reset command: OUT &H1nFC,0.

3. Set up DMA Select Register to desired GDC and desired channel number: OUT &H1n2A, &HXXXX.

4. Set up the DMA controller on the CPU board.

5. Issue the GDC DMA command.

12.5.2 Programming the Video Sync Generator

The software must set up the line length so that the number of pixels per line is a multiple of both 16 and 10. The number 1040 is the closest usable value for the CRT. A further restriction is that the number of active display words per line must be a multiple of 2.

Both GDCs must be programmed in Interlaced mode. The total number of lines per frame is 635 divided into two fields of 317.5 each. The total number of vertical active lines is 300.5 per field or 601 per frame. The number of vertical blanked lines is 17 per field divided between the vertical front porch, vertical synchronization, and vertical back porch. Both image and text GDCs should have the same parameters.

Table 12-7 shows one set of horizontal and vertical timing parameters.

Table 12-7. Example of Horizontal and Vertical Timing Parameters

		Image GDC	Text GDC
AW	Active words (800 dots)	50	80
HFP	Horizontal front porch	5	8
HS	Horizontal synchronization	5	8
HBP	Horizontal back porch	5	8
	Total words per line	65	104
VFP	Vertical front porch	3	3
VS	Vertical synchronization	3	3
VBP	Vertical back porch	11	11
AL	Active lines per field	300	300
	Total lines per field	317	317

To make the display an integral number of text lines, the total number of active lines can be reduced. The difference should be added to the vertical front or back porch to keep the same total number of lines.

One of the GDCs must always be designated as the master for generating timing and the other must be the slave. An initialization procedure that allows synchronization is performed by the software. The two GDCs actively synchronize only during this procedure. Afterward they run independently and remain in synchronization only if they are programmed with the correct timing parameters.

The software may designate either GDC as the master. The faster character GDC should be the master under normal conditions. Programming both GDCs as master, i.e., not making one of them a slave, is meaningless and potentially damaging to the hardware. However, either GDC can be run alone (i.e., as the master) and the other GDC left idle.

The programming procedure for synchronizing the two GDCs requires close attention. Refer to the Intel and NEC GCD documents for more information.

12.5.3 Cursor Programming

The cursor on the character GDC is programmable for ON/OFF, size, position, and blink rate. When on, the cursor is fixed at maximum intensity. Typically, double underline cursors are used. To conform with the medium-resolution controller, a four-line cursor should be used from Line 20 through Line 23.

The cursor is not affected by any of the character attributes. However, combinations of attributes may obscure the cursor. In the presence of blinking characters, the cursor blinks differently than the character to help make it visible. Refer to the Intel and NEC GCD documents for more information.

12.5.4 Other GDC Information

The following list contains additional information about the GDCs:

- Both GDCs should be operated in dynamic RAM Refresh mode.
- The image GDC should be operated in Graphics mode only.
- The character GDC should be operated in Mixed mode, but no attempt should be made to use an image area. In Mixed mode, it is simple to trick the GDC into using a full 16-bit word so that a full complement of character attributes will be available.
- The display zoom factors work only in the vertical dimension.

12.6 TIMING DIAGRAMS

This section presents the clock and horizontal timing diagrams for the TIG board. Figure 12-7 shows the clock diagram. Figure 12-8 shows the horizontal and vertical timing diagrams.

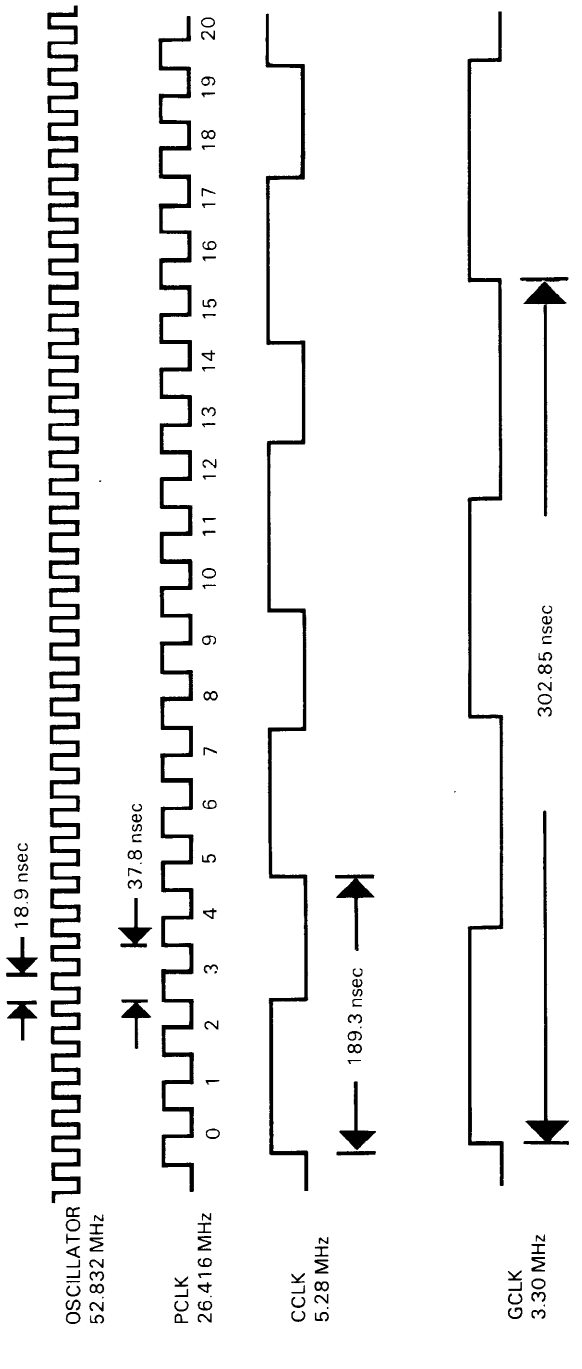


Figure 12-7. TIG Board Clock Timing Diagram

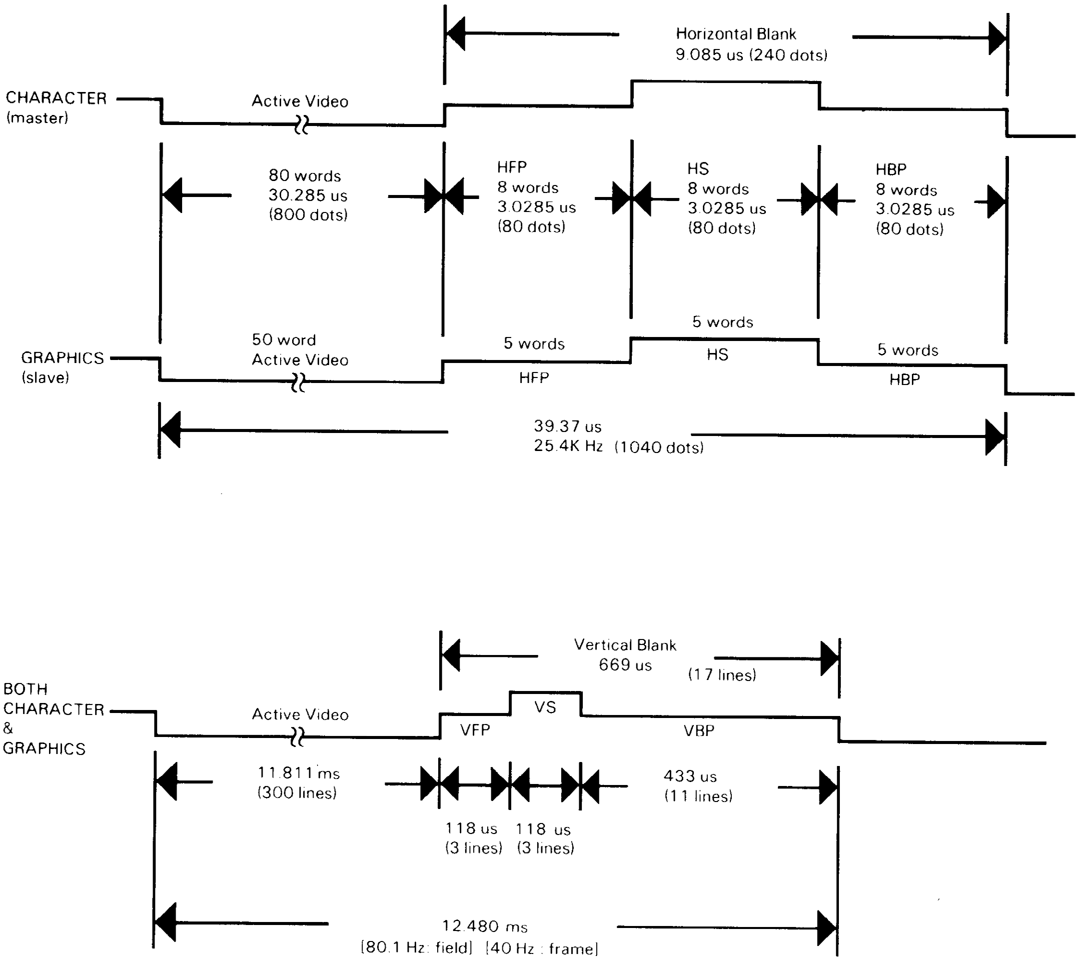


Figure 12-8. TIG Board Horizontal and Vertical Timing Diagrams

12.7 PROGRAMMABLE ARRAY LOGIC CHIPS

The TIG board uses two programmable array logic chips (PAL). One of these chips, L26, is used to decode GDC read and write addresses. The other chip, L16, is used to decode various I/O register addresses. This section presents the specifications for each chip.

12.7.1 The PAL for GDC Read and Write Address Decoding

The specifications for the read/write address decoding chip are as follows:

CHIP LOCATION:	PCB TIG L26
TYPE OF PAL:	PAL10L8
WL# (BLANK PAL)	377-0463
WL# (PROG. PAL)	377-3072

Input Signals:

PIN # 1	EQL	PIN # 5	/AIOW	PIN # 9	CA2
PIN # 2	/DACK	PIN # 6	/CRST	PIN # 10	GND
PIN # 3	CA5	PIN # 7	GDCS	PIN # 11	A0
PIN # 4	CA3	PIN # 8	/IOR		

Output Signal Equations:

```
PIN # 19     AOQ    = /(EQL * /A0

                         + DACK * /A0)

PIN # 18     /RF    = /(EQL * /CA5 * AIOW)

PIN # 17    /DRST   = /(DACK

                         + CRST)

PIN # 16    /RWES   = /(EQL * CA5 * CA3)

PIN # 15     /CWR   = /(DACK * AIOW * /GDCS

                         + EQL * CA5 * /CA3 * AIOW * /CA2)

PIN # 14     /CRD   = /(DACK * /AIOW * /GDCS * IOR

                         + EQL * CA5 * /CA3 * /AIOW * IOR * /CA2)

PIN # 13     /GWR   = /(DACK * AIOW * GDCS

                         + EQL * CA5 * /CA3 * AIOW * CA2)
```

```
    PIN # 12     /GRD  = /(DACK * /AIOW * GDCS * IOR

                            + EQL * CA5 * /CA3 * /AIOW * IOR * CA2)
```

12.7.2 The PAL for I/O Register Address Decoding

The specifications for the I/O register address decoding chip are as follows:

CHIP LOCATION:	PCB TIG L16
TYPE OF PAL:	PAL10L8
WL# (BLANK PAL)	377-0463
WL# (PROG. PAL)	377-3073

Input Signals:

PIN # 1	/AIOW	PIN # 5	CA1	PIN # 9	/BHE
PIN # 2	/RWES	PIN # 6	/RESET	PIN # 10	GND
PIN # 3	/IOR	PIN # 7	DACK	PIN # 11	NC
PIN # 4	CA2	PIN # 8	GDCS		

Output Signal Equations:

```
    PIN # 19      /UWE  = /(AIOW * RWES * /CA2 * /CA1)

    PIN # 18      /DWE  = /(AIOW * RWES * /CA2 * CA1)

    PIN # 17     /IDCD  = /(RWES * IOR * CA2 * CA1)

    PIN # 16     /CRST  = /(AIOW * RWES * CA2 * /CA1
                            + RESET)

    PIN # 15    READY   = /(AIOW * DACK

                            + IOR * DACK)

    PIN # 14    /CDACK  = /(DACK * /GDCS)

    PIN # 13    /GDACK  = /(DACK * GDCS)

    PIN # 12      /DHE  = /(DACK * BHE)
```

CHAPTER 13
SYSTEM BOARD I/O PORT ASSIGNMENTS

Because the floppy disk controller, keyboard interface, communication interface, and parallel port interface all reside on the system board, they all have I/O port assignments in the Slot 0 port address range (1000-10FEH). Table 13-1 lists I/O port partitions within this range.

Table 13-1. System Board I/O Port Assignments

I/O Interface	Low Address	High Address
uPD765 Floppy Disk Controller	1000H	101EH
8255A Programmable Peripheral Interface	1020H	1026H
8253-5 Programmable Interval Timer	1040H	1046H
8259A Programmable Interrupt Controller	1060H	1062H
2661 Extended Programmable Communications Interface	1080H	108EH
9517 DMA Controller	10A0H	10BEH
6402 UART (Keyboard Interface)	10E6H	10E8H
DMA Page Register	10C2H	10C6H
Parallel I/O Interface	10EAH	10ECH
Slot 0 (Floppy Disk) ID Code	10FEH	
NMI Mask, 8087 Interrupt Mask, Clock Interrupt	10E0H	10E4H
Diagnostic LED, Disk Status	10EEH	

The following lists contain the individual port assignments for each partition.

uPD765 Floppy Disk Controller Addresses

Write DMA End-of-Process and acknowledge flipflops	1000H
Read or write to deselect Drive Unit 1	1004H
Read or write to select Drive Unit 1	1006H
Read or write to deselect Drive Unit 2	1008H
Read or write to select Drive Unit 2	100AH
Read or write to turn Unit 1 motor off	100CH
Read or write to turn Unit 1 motor on	100EH
Read or write to turn Unit 2 motor off	1010H
Read or write to turn Unit 2 motor on	1012H
Read Main Status Register	1014H
Data Register	1016H
Read or write to reset FDC chip	1018-AH
Read or write to issue Terminal Count (TC)	101C-EH
Read interrupt request and door disturb status	10FEH

8255A Parallel I/O Interface Addresses

Read eight bits of parallel printer status	1020H
Read interrupt status flags	1022H
Write three printer control lines of read-user switches	1024H
Write 8255A Control Word (D0-7 = 9AH)	1026H

8253-5 Programmable Interval Timer Addresses

Load or read Counter 0	1040H
Load or read Counter 1	1042H
Load or read Counter 2	1044H
Write CTC Mode Word	1046H

8259A Programmable Interrupt Controller Addresses

Interrupt Request Register, In-Service Register, Interrupt Level, ICW1, OCW2-3	1060H
Interrupt Mask Register, ICW2-4, OCW1	1062H

2661 Extended Programmable Communication Interface Addresses

Read Receive Data Holding Register	1080H
Read Status Register	1082H
Read Mode Registers 1 and 2	1084H
Read Command Register	1086H
Write Transmit Data Holding Register	1088H
Write SYN1, SYN2, and DLE Registers (unsupported)	108AH
Write Mode Registers 1 and 2	108CH
Write Command Register	108EH

DMA Controller Addresses

Current Address Register, Channel 0	10A0H
Word Count Register, Channel 0	10A2H
Current Address Register, Channel 1	10A4H
Word Count Register, Channel 1	10A6H
Current Address Register, Channel 2	10A8H
Word Count Register, Channel 2	10AAH
Current Address Register, Channel 3	10ACH
Word Count Register, Channel 3	10AEH
Write Command Register or read Status Register	10B0H
Write Request Register	10B2H
Write individual Mask Register bit	10B4H
Write Mode Register	10B6H
Clear Byte Pointer Flipflop	10B8H
Write arbitrary data to reset DMA controller	10BAH
Write all Mask Register bits	10BEH

DMA Page Register Addresses

Write Channel 1 Page Register	10C2H
Write Channel 2 Page Register	10C4H
Write Channel 3 Page Register	10C6H

Parallel I/O Interface Addresses

Read input data and clear Data Available flag (/DAV)	10EAH
Write output data and set Acknowledge flag (/ACKNLG)	10EAH
Write to clear /ACKNLG flag	10ECH
Read to clear BUSY flag	10ECH

6402 UART (Keyboard Interface) Addresses

Write to clear transmit interrupt	10E6H
Read Receive Buffer Register	10E8H
Write Transmit Buffer Register	10E8H

Slot Identification Code Addresses

Read slot ID (/FID DECODE)	10FEH

NMI Mask, 8087 Interrupt Mask, PCT Interrupt Addresses

Read System Status Port	10E0H
Write to clear timer Channel 0 (realtime clock) interrupt request	10E0H
Write with D0 set to enable NMI or D0 cleared to disable NMI	10E2H
Read to clear timer Channel 2 interrupt request	10E2H
Write with D0=1 to enable (D0=0 to disable) 8087 interrupt	10E4H
Read to light diagnostic LED	10E4H
Read to clear DMA terminal count interrupt request	10E6H
Write arbitrary data to clear parity flag	10EEH
Read to extinguish diagnostic LED	10EEH

CHAPTER 14
START AND ON-BOARD PROMS

This chapter contains information on the Wang PC Start PROM and On-Board PROMs (OBPROM)s. Refer to Appendix B, Power-on Diagnostics, for an explanation of the diagnostic tests included on the Start PROM.

The PROM discussions use the following terms and abbreviations:

Abbreviation	Definition
C/O	Console Output
C/I	Console Input
EA	Effective Address or 20-bit 8086 physical address
OBPROM	On-Board PROM on an option card that carries the code necessary to diagnose, start from, and/or perform C/O to the option card
ONlY	0 = No - 1 = Yes
OiCo	Offset into Code of

The term hierarchy refers to the canonical order of the automatic selection of start devices and C/O devices.

14.1 START PROM OVERVIEW

This section outlines the operation of the Start PROM. For details, refer to Section 14.3, Start-time Detail, and the other sections referred to in the outline.

14.1.1 Power-On Time Operations

At power-on time, the Start PROM performs the following diagnostic tests and operations:

1. Verify system card integrity. A failure of a component, except for serial and parallel ports, is fatal.

2. Verify the integrity of 128 KB of system card memory and clear memory to 00's. Failure is fatal.

3. Inventory all option slots. If a card has an OBPROM, the Start PROM copies data from the OBPROM to RAM (refer to Section 14.17, OBPROMs).

4. Test all available console output, console input, and start devices. Failure is not fatal unless it affects the address and/or data bus or interrupts.

5. Save information from Steps 1, 2, 3, and 4 in mailboxes at Effective Address (EA) 0400H.

6. Select C/O based on the following C/O hierarchy:

    a. Video devices, if any are functional

    b. Serial port, if there are no functional video devices

    c. Parallel port, if the serial port is unusable and a Centronics-type printer is attached

7. Print the System Identifier.

8. Pass control to the start-up code.

14.1.2 Start-up Code Operations

The start-up code performs the following operations, in hierarchical order:

```
1.  List on the C/O all Start and C/O devices that failed the power-on
    diagnostics.

    If the CPU baud rate DIP switch is set to 0000

    then

        If CPU memory is greater than 128 KB, then

        a.  Test extra memory
        b.  Jump to power-on diagnostics

    else:

2.  If the RS-232 port is functional, then

    a.  Turn off DTR and RTS (RS-232 signals) and signal to RS-232 that
        data may now be sent

    b.  Wait 100 milliseconds

    c.  Turn On DTR and RTS (RS-232 signals)

    Enable interrupts.
```

3. Start a 3-second timer running.

4. If there is no usable start source

   then

   a. Print "No Start Source" message
   b. Go to 5.

   else

   a. Select the start source according to the following hierarchy:

      Drive A, if the door is closed

      Drive B, if the door is closed

      Local Communications Card, LIO, or future start devices, if the switch is in the "Remote" position

      The card in the lowest numbered slot, if there are multiple switches in the "Remote" position

      Winchester (the card in lowest numbered slot), if no Local/Remote switches are in the "Remote" position.

   b. Print selected start source.

   c. Load a 4 KB block of data into memory (refer to Section 14.4, Load Procedure Detail).

   If no error ocurred in loading, then go to 6, else continue at 5.

5. In the event of an error, then

   a. Issue an error message to the C/O device.
   b. Wait for C/I command (Int 97H, refer to Section 14.12, Start Error Recovery Routine) to do one of the following:

Retry (go to 3)

Redirect start-up (to a start device other than the one chosen by the hierarchy)

Redirect C/O

Redirect C/I

Reexecute the power-on diagnostics

Access the Manufacturing Diagnostic menu

Execute a quick restart

Access a Help menu

6. Check Bytes 3, 4, 5, and 6 (the first byte is Byte 0) for Wang and verify the checksum of Bytes 0 through 511. If the data is all right, continue with 7, else go to 5.

7. Wait until the 3-second timer has timed out. During this period, the user can do one of the following:

   Truncate the delay

   Redirect Start (start device other than the one chosen by hierarchy)

   Redirect C/O

   Redirect C/I

   Reexecute the power-on diagnostics

   Access the Manufacturing Diagnostic menu

   Execute a quick restart

   Access a Help menu

   Change C/I to asynchronous communications by sending three 01BHs in the RS-232 port

8. Set DS and ES to the Segment of the loaded data, Push DS, Push 0000H, and exectue a FAR RETURN (effectively a FAR JUMP pointer to the first byte of the loaded data).

9. The loaded code may use the Start PROM to perform additional reads, C/O, and/or C/I. If the loaded code experiences an error during an additional read, it may call an Int 97H (i.e., go to 10). If all reads are successful when an operating system takes over, the Start PROM is done.

10. Issue an error message to the C/O device and wait for C/I commands (see RECOVER or Int 97H) to do one of the following:

    Retry (go to 9)

    Redirect Start (start device other than 1 chosen by hierarchy)

    Redirect C/O

    Redirect C/I

    Reexecute the power-on diagnostics

    Access the Manufacturing Diagnostic menu

    Execute a quick restart

    Access a Help menu

14.2 Important Features of the Start-up Process

The following specifications and conditions for the start-up process should be noted:

- Code loaded from the start device is normally located at EA 010000H and the start-up effectively executes a FAR JUMP to 1000:0000H.

- From power-on until C/O is available, the diagnostics update the keyboard LEDs, showing activity and serving as a fatal error indicator.

- Any character from the keyboard or three consecutive 01BHs from the asynchronous port selects the console input device for the start-up. The other source is then ignored.

- If one or more functional video devices exists, C/O is sent to all of them. If no functional video device exists, C/O is sent to the serial port, if that port is functional and Data Set Ready is TRUE. If the serial port is non-functional or DSR is FALSE, C/O is sent to the parallel port, if that port is functional and a Centronics-type printer (e.g., a Wang PC-PM016) is attached. If no C/O device is available, no messages are sent.

- During the diagnostic phase of the start-up, interrupts are disabled and the keystroke buffer does not work for key-ahead. When interrupts are enabled, valid keystrokes are generally singular, and the user should wait for some indication of results before issuing another keystroke.

- Through console input, the user can direct the Wang PC to execute the one of the following following:

  - During a 3-second delay after diagnostics

    Abort the delay and execute the start code immediately

    Start from a specific device other than the one specified by normal priority and, optionally, load into a specific 4 KB block of memory

    Reexecute the power-on diagnostics

    Access the Manufacturing Diagnostic menu

    Execute a quick restart

    Redirect console input/output

    Access the Help menu

  - After an unsuccessful start

    Retry the auto-start hierarchy

    Try another start source

    Reexecute the power-on diagnostics

    Execute a quick restart

    Access the Manufacturing Diagnostic menu

    Redirect console output and/or console input

    Access a Help menu

- Video devices are assumed to have 40 columns. Messages must fit on a 40-column device. The length of a device name can be no longer than 15 characters.

- The valid Wang PC keyboard keys used in the start-up sequences are based on the US Standard Keyboard Layout. All power-on and start-up messages are in English with International Numeric Prefixes.

- To execute a restart the code must execute a FAR JUMP to FFFE:0009 after issuing an OUT 1028H with AL = 0FFH (to disable the upper bank of memory) and, if necessary, resetting Timers 0 and 2 to the power-on defaults.

- Start devices have a hierarchical priority. The first usable device is the start device. An open door on a diskette drive causes the drive to be bypassed. On the Local Communications Option, the Local Interconnect Option, and all future start source option boards, there is a switch for the user to indicate which option board to consider as the start device before the Winchester. If two switches are active, the lowest-numbered slot has priority. The Winchester has the lowest priority because it has no switch.

NOTE:
On a secure Local Communications Option, the Local/Remote switch setting is latched at power-on and cannot be changed without powering off and on again. In Local Communications, the data link is disabled and start-up cannot take place from this option card. In Remote Communications, start-up can take place only from this option card.

- Any option board that can be a start source and/or the only C/O device at start time must carry with it, in an OBPROM, code and data to diagnose, start, and/or perform console output. Code has been included in the Start PROM to copy OBPROM code to RAM and execute it. The high-resolution monitor and Local Communication Option do not have OBPROMS, but the Start PROM treats them as if they do by copying code to RAM.

- A Wang PC can theoretically have 15 option slots.

- Memory Allocation is as described in Table 14-1, which shows the RAM reserved areas, and Table 14-2, which shows the reserved mapping areas.

Table 14-1. RAM Allocation

Address	Usage
00000H through 003FFH	Reserved for interrupt vectors
00400H through 00FFFH	Reserved for mailboxes, variables, and the stack
01000H through 0BFFFH	Reserved for code copied from OBPROMS
0C000H through 0EFFFH	Reserved for Winchester diagnostics
10000H through 13FFFH	Reserved for a copy of the Start PROM for Loop-on-power-on CPU memory diagnostics

Table 14-2. Mapping Areas

Address	Usage
C0000H through CFFFFH	Reserved for Local Communications mapping
E0000H through F3FFFH	Reserved for video memory mapping
F4000H through FBFFFH	Reserved for OBPROM mapping

The Start PROM code memory usage is as follows:

- The diagnostics, start code, fonts, and keyboard translate table occupy 16 KB.
- The System Serial Number starts at EA FFFB0H and is 16 bytes, 8 in each PROM (2732A PROM Address 0FD8H/2764 PROM Address 1FD8H), intermeshed. The serial number in each PROM can be used as a unique ID. Refer to Section 14.21 for the format of the serial number. In unserialized PROMS, this space is filled with "No Serial Number."
- The System Identifier Block, containing the code revision number and other things, starts at EA FFFC0H and is 41 bytes in length. It is coded in ASCII and ends with 0D00H.
- The restart JUMP is located at EA FFFE9H and is five bytes.
- A 2-byte PROM checksum starts at EA FFFEEH. The first byte is the checksum of the even PROM (2732A PROM Address FF7H/2764 PROM Address 1FF7H). The second byte is the checksum of the odd PROM (2732A PROM Address FF7H/2764 PROM Address 1FF7H). The checksum includes all but the last eight bytes of the PROM. In unserialized PROMs, the even checksum is 045H (E) and the odd checksum is 04FH (O).
- The cold start JUMP (Intel Standard) is located at FFFF0H and is five bytes. The remaining 11 bytes in the PROM are reserved by Intel and are set to 00H.

14.3 START-TIME DETAIL

This section provides details of the Start PROM operations outlined in Section\14.1.

The following messages appear after the power-on diagnostics:

```
WANG PROFESSIONAL COMPUTER REV. 2.40
Copyright Wang Laboratories, Inc., 1983
```

If any Slot 00 (CPU) device is non-functional, the following message appears (the maximum number of reasons that can appear is three):

```
*22 [device] - Defective
[## reason]
[## reason]
[## reason]
```

If any Slot 01 to 15 device is non-functional, the following message appears (the maximum number of reasons that can appear is three):

```
*21 [device] In ## - Defective
[## reason]         (Maximum = three reasons or fragments)
[## reason]
[## reason]
```

Refer to Section 14.18, Start-time Messages, for a listing of device names and reasons supplied with the Start PROM.

Next, if the baud rate DIP switch on the CPU is set to 0000b the system displays the following message:

```
91 Looping On Power-Up
```

All system card memory above 128K is tested. The system then reexecutes the power-on diagnostics. The diagnostic loop continues until the switches are set to nonzero.

Otherwise, at this time, the user can key ESCAPE (01BH) three times on a terminal connected to the asynchronous line to become the console input device. Three consecutive 01BHs from the asynchronous port, or any character from the Wang PC keyboard, make that device the default console input device. Inputs from nonconsole-input devices are thrown away, except when that device is the start source.

If no start device is available, the message and prompt for console input appear as follows:

```
***40 No Auto-Start Device
:_
```

At this point, the user has the options listed in Table 14-3.

Table 14-3. User Restart Options

Press	Result
Key D	Redirect start-up
Key I	Redirect console input
Key O	Redirect console output
Key P	Rerun power-on diagnostics
Key M	Access the Manufacturing Diagnostic menu
Key R	Retry start-up - The hierarchy will be reassesed
Key Q	Quick restart
Key Help	Obtain this list

If a start device is available, the message and prompt for console input appear as follows:

```
01 Start From device

_
```

During a 3-second wait, the system loads start code from the selected device and delays the remainder of the 3 seconds. During this delay, the user has the following options:

- Do nothing. Start-up will commence 3 seconds later.
- Key G - Bypass the remainder of the delay and execute the code.
- Key D - Redirect start-up.
- Key I - Redirect console input. The hierarchy is reassesed.
- Key O - Redirect console output - The hierarchy is reassesed.
- Key P - Rerun power-on diagnostics.
- Key M - Access the Manufacturing Diagnostic menu.
- Key Q - Quick restart.
- Key HELP to obtain this list.

When the start-up is from Local Communications, if the File Server is busy for more than 10 seconds, the following message appears:

```
*92 Host Appears Busy. I Will Retry
    Until You Depress A Key.
```

Pressing any key aborts the retry attempts and the start-up operation processes the keystroke as if it had occurred during the 3-second delay.

If the start-up is successful, control passes to the loaded code. If the start-up fails, the message and prompt for console input appear as follows:

```
***41 Start Failed
   ## [device] [reason]
:_
```

For a list of reasons for start-up failure, refer to Section 14.18, Start-time Messages. The user has the options listed in Table 14-3.

If a hardware malfunction occurs during start-up, one of the following messages may appear:

```
***42 Invalid Interrupt
***43 Parity Error
```

The following messages are issued by the disk start-track code if an error occurs during execution of additional reads beyond the 4 KB read of the start-up code:

```
***44 System Files Missing
***45 Disk Read Error
***46 Defective Start Disk
```

The start-up code issues the following messages for disk or nondisk devices if an error occurs during execution of additional reads beyond the 4 KB read of the start-up code:

```
***47 Missing System File
     [- file name]
***48 Read Error
     [- device]
***49 Not Ready Error
     [- device]
```

For a complete description of error messages, refer to Section 14.18, Start-time Messages.

14.4 LOAD PROCEDURE DETAIL

The load and first-read procedures differ for OBPROM and nonOBPROM devices. This section describes these procedures.

14.4.1 Load and First-Read Procedures for NonOBPROM Devices

The following is the load procedure if the selected device does not have an OBPROM (diskette drives, Winchester, asynchronous communications, keyboard).

1. When the set-up code for the selected device is executed, Start Device Specify parameters (refer to Section 14.10) are loaded into appropriate registers, and Int 92H is executed.

2. The Int 92H code copies the passed parameters into the mailbox for the selected device, and the Start From OBPROM flag is set to FALSE.

3. The contents of the device mailbox are copied to the start mailbox.

4. The Int 92H code then returns.

The first-read procedure for nonOBPROM devices is as follows:

1. The registers are loaded with the parameters for a 4K data read. An Int 91H is executed.

2. The Int 91H code calls an appropriate routine (diskette, Winchester, asynchronous communications, or keyboard), which attempts to read the first 4 KB, loads the registers with the results of the operation, and returns.

3. The Int 91H code then returns.

14.4.2 Load and First-Read Procedures for OBPROM Devices

The following is the load procedure if the selected device has an OBPROM.

1. When the OBPROM set-up code is executed, Start Device Specification parameters (refer to Section 4-11) are loaded into appropriate registers, and Int 92H is executed.

2. In the Int 92H code, the contents of the mailbox are copied to the start mailbox, and the Start From OBPROM Flag is set to TRUE.

3. The Int 92H code retrieves a pointer to the copied OBPROM code from the mailbox of the selected device. Int 92H executes a FAR CALL to the Start Device Specify address in the header of the OBPROM code (refer to Section 14.17.2, OBPROM Concept of Operation). This code performs an appropriate Start Device Specify. It then executes a FAR RETURN.

4. The Int 92H code then returns.

The first-read procedure for OBPROM devices is as follows:

1. The registers are loaded with the parameters for a 4 KB data read. An Int 91H is executed.

2. The Int 91H code retrieves a pointer the copied OBPROM code from the mailbox of the selected device. Int 91H then executes a FAR CALL to the read address in the header of the OBPROM code. This code performs a 4 KB read, as specified. It loads the registers with the results of the operation and executes a FAR RETURN.

3. The Int 91H code then returns.

14.5 REDIRECT FUNCTIONS

This section describes the following redirect functions: Redirect Start, Redirect Console Output, and Redirect Console Input.

14.5.1 Redirect Start

The Redirect Start function allows the user to specify the system start device. If this option is selected, the message and prompt for console input appear as follows:

```
14 Redirect Start
:_
```

The command structure is

```
    device [slot] [4 KB Block Address] RETURN
or
    RETURN
```

```
device  A = Floppy A (Slot # irrelevant)
        B = Floppy B (Slot # irrelevant)
        K = Keyboard (Slot # irrelevant)
        S = Asynchronous port (Slot # irrelevant)
        W = Winchester [in slot ##]
        N = New Device [in slot ##] (includes Local Communications,
            LIO, etc.)

slot        01 through 15 (decimal) to indicate the slot, if there
            are duplicate sources.

            00 indicates any slot; the lowest is the default, if
            there are multiple slots.
```

The command is column-oriented, with no spaces. Pressing RETURN with no parameters causes the system to exit without change and reparse the hierarchy.

If the syntax is illegal, the device specified is not in the slot specified, or the memory segment specified is not a 4 KB boundary, etc., one of the message and prompts for console input, appear as follows:

```
*24 Illegal Device
*25 Illegal Slot Number
*26 Illegal Address
*27 Device Not Available
:_
```

At this point, the user must input a valid redirect command or press RETURN to reexecute the hierarchy routine. The HELP key is active with a menu of options.

14.5.2 Redirect Console Output

The Redirect Console Output function allows the user to specify the device(s) to which the system should send console output. After exit, the hierarchy is reparsed.

If this option is selected, the message and prompt for console input appear as follows:

```
15 Redirect Console Out
:_
```

The command structure is

```
    device [device] [device] RETURN
or
    RETURN

    device  V = Video
            S = Serial port
            T = Printer
```

No spaces are allowed in the command. Pressing RETURN without parameters causes an exit with no change. If you specify three devices, you do not need to press RETURN. The HELP key is active with a menu of options.

14.5.3 Redirect Console Input

The Redirect Console Input function allows the user to specify the device from which the system should receive console input. After exit, the hierarchy is reparsed.

If this option is selected, the message and prompt for console input appear as follows:

```
16 Redirect Console In
:_
```

The command structure is

```
    device
or
    RETURN       (exit without change)

    device  K = Keyboard
            S = Serial Port
```

No spaces are allowed in the command. Pressing RETURN without a device causes an exit with no change. The HELP key is active with a menu of options.

14.6 START FROM ASYNCHRONOUS PORT

The Start from Asynchronous Port function is not in the hierarchy and must be invoked from Redirect Start. When this function is invoked, the following prompt is sent to the RS-232 port:

?B?

The Data Source on the other end must respond with

Y

The CPU then starts accepting 4 KB blocks of data in ASCII, high nibble first, low nibble second. The received data is echoed to C/O. The source should wait until each character is echoed before sending another. The baud rate is derived from the CPU DIP switch settings (refer to Section 14.14, Power-on Defaults).

CR is echoed as CR/LF. CR is used for neatness of display only and is not necessary. The termination characters are as follows:

X	Abort	Start-up indicates an error and wait for a retry.
01BH	Execute	Start-up commences after verification of data. Refer to Section 14.20, Checksum Algorithms, on the checksum of first 512 bytes.

14.7 START FROM KEYBOARD

The Start from Keyboard function is not in the hierarchy and must be invoked from Redirect Start. When this function is invoked, the following prompt is sent to C/O:

93 Input Start Data

Executable code is then entered via the keyboard in ASCII, high nibble first, low nibble second. The received data is echoed to C/O.

RETURN is used for neatness of display only and is not necessary. The termination characters are as follows:

X	Abort	Start-up indicates an error and wait for a retry.
01BH	Execute	Start-up commences. Data is not verified.

14.8 INTERRUPT CALLS

All commonly used functions are accessed by interrupt calls. Thus, Start PROM code, loaded code, and OBPROM code can use the same functions. Table 14-4 summarizes the interrupts used. Sections 14.8.1 through 14.8.8 give details concerning the interrupts.

Table 14-4. Start PROM Interrupts

Interrupt	Function
88H	Read Console Input
89H	Test Console Input
8AH	Character Console Output
8BH	String Console Output
8DH	Reserved
8EH	Reserved
8FH	Memory Test (used by diagnostics)
90H	Reserved
91H	Start Device Read
92H	Start Device Specify
93H	Reserved
94H	Reserved
95H	Reserved
96H	Reserved
97H	Start Error Recovery

14.8.1 Console Input Specifications

A console input buffer overrun causes a Beep code to be sent to the console output device. The buffer length is 1 byte.

Int 88H, Read Console Input, checks the keyboard buffer for an input character, returns the character, if any, or the NULL flag, and empties the buffer.

All registers are preserved except AL, which returns any input character; otherwise, AL has 00H.

Int 89H, Test Console Input, checks the keyboard buffer for an input character, returns the character, if any, or the NULL flag, but does not empty the buffer.

All registers are preserved except AL, which returns any input character; otherwise, AL has 00H.

The following list shows the character keys and standard function keys on the Wang PC keyboard:

ABCDEFGHIJKLMNOPQRSTUVWXYZ

0123456789,./;'-+=

Return (0DH), Space (20H), Backspace (08H), Cancel (03H), Execute (1BH), and Help (48H).

Cancel (03H) is intercepted as a quick start.

Table 14-5 lists the codes returned by the special function keys.

Table 14-5. Special Function Key Codes

Key	Code
1	A
2	B
3	D
4	G
5	I
6	K
7	M
8	N
9	O
10	P
11	Q
12	R
13	S
14	T
15	V
16	W

The SHIFT, 2ND, LOCK, and CONTROL keys do not affect the key codes. Acceptance of a keystroke from the Wang PC keyboard is reflected as a click.

14.8.2 Console Output Specifications

Int 8AH, Character Output, sends a character in Al. All registers are preserved; no parameters are returned.

Int 8BH, String Output, sends a string beginning at DS:SI in memory and terminated by a 00H. All registers are preserved; no parameters are returned.

Output to Video Devices

Table 14-6 lists the codes for the keyboard font.

Table 14-6. Keyboard Font Codes

Characters	Codes
!"#$%& ()*+'-./	020H - 02FH
0123456789:; = ?	030H - 03FH
@ABCDEFGHIJKLMNO	040H - 04FH
PQRSTUVWXYZ[\] _	050H - 05FH
abcdefghijklmno	060H - 06FH
pqrstuvwxyz{ | }~ ¢	070H - 07FH

Only Codes 01H, 08H, 0AH, 0CH, 0DH, 0EH, 0FH, 20H through 7FH are valid. All others are ignored.

Video output supports the following functions:

> CR (0DH), LF (0AH), Backspace (08H), Character Emphasis On (0EH), Character Emphasis Off (0FH), Clear Screen (0CH), Home Cursor (01H).

CR is implemented as CR and LF. Character Emphasis mode stays ON until turned OFF. The Current state is stored in the Mailbox of Video device.

Output to the Asynchronous Port

Output to the asynchronous port is the same as output to video devices except that Enhanced-video-on and Enhanced-video-off characters are not output.

Output to Printers

Output to printers is the same as output to video devices, with the following exceptions:

- The Enhanced-video-off character is not output. The character causes variable results, depending on the type of printer.
- Clear Screen causes a form feed in a Centronics-type printer.
- Clear Screen is not sent during quick restart or looping on power-on.

14.9 START DEVICE READ ROUTINE

Int 91H, Start Device Read, passes the following parameters:

- In ES:SI, the starting address in memory.
- In BX, the starting sector number (for disk-type devices).
- In AL, the number of 256-byte blocks to read. If 0, 256 blocks are read. The start in memory plus the product of the number of block times 256 must not cross a 64 KB boundary (this is a DMA hardware restriction). The starting block plus the number of blocks to read must be no greater than the last block in or on the device.
- In DL, the number of retries.

The validity of these parameters are checked against the Start Device Specify parameters only.

All registers are preserved with the following exceptions: DL returns the number of retries executed; if AH is 07FH, SI and DI return message fragments. AH returns one of the following codes:

80H	No error
83H	Corrected Error
01H	Device Not Ready
03H	CRC Error
04H	Format Error
05H	Equipment Malfunction
08H	Programmer Error
09H	Device Dropped Ready
0AH	Time Out
7FH	Generic Error Code (for option boards with an OBPROM). SI and DI point to two halves of an error message, which will be wrapped around the name of the device.

14.10 START DEVICE SPECIFY (DISK PARAMETERS ONLY)

Int 92H, Start Device Specify, passes the following parameters for disk drives:

- In AH, the number of heads per cylinder. The default is 1.
- In AL, the number of sectors per track. The default is 8.
- In DX, Relative Sector 0 on disk. The default is 0.
- In CH, a sector size indicator:

  01H for 256 bytes/sector
  02H for 512 bytes/sector (the default)
  03H for 1024 bytes/sector

- In CL, a gap length indicator. Must be 00H for the Winchester. The values for diskettes are as follows:

  020H for 256 bytes/sector, 16 sectors/track
  02AH for 512 bytes/sector, 8 or 9 sectors/track (the default)
  080H for 1024 bytes/sector, 4 sectors/track.

The following are internally set parameters for the UPD765 diskette controller:

- Head load time (used as step settling time) - 40 ms (14H)
- Step rate - 20 ms (6xH)
- Head unload time (set to maximum) - 480 ms (xFH)

All registers are preserved, and no parameters are returned.

NOTE:
There is no test for validity, so be very careful.

14.11 START DEVICE SPECIFY (OBPROM BOARD)

Int 92H, Start Device Specify, passes parameters for OBPROMS according to the program written for the OBPROM. Use registers when the program is passing only a few parameters. For many parameters, pass a pointer to a data block in ES:BX.

The program must not pass parameters in DS, DI, or BP. These registers are used internally.

All registers are preserved, and no parameters are returned.

NOTE:
There is no test for validity, so be very careful.

14.12 START-UP ERROR RECOVERY ROUTINE

Int 97H, Start Error Recovery, allows users to deal with errors during start-up. The calling routine should have posted an error message to console output.

Start Error Recovery passes the following parameters:

In BL, a bit mask indicating which keystrokes are accepted as valid; 0 indicates do not accept, and 1 indicates accept. The HELP key is also active and gives a menu of options.

Bit	Keystroke	Meaning
0	G	Go. Returns to calling routine. Equivalent To R.
1	R	Retry. Returns to calling routine.
2	D	Redirect Start, C/I, and C/O. Resets the stack, resets the interrupt vector table, and takes control.
3	P	Reexecute power-on diagnostics. Takes control.
4	M	Access Manufacturing Diagnostic menu. Takes control.
5	Q	Execute Quick Restart.
6	Reserved - use 0	
7	Reserved - use 0	

In BH, a bit mask indicating which functions are to be executed.

Bit	Meaning
0	Reserved - use 0
1	Reserved - use 0
2	Reserved - use 0
3	Reserved - use 0
4	Reserved - use 0
5	Reserved - use 0
6	If 1, issue a beep code before accepting a character.
7	If 1, do not issue a CR/LF on entry to the routine.
8	If 1, do not issue a CR/LF after accepting a character.

All registers are preserved if and only if the routine returns to the caller; otherwise they are not preserved. No parameters are returned.

14.13 MEMORY TEST

Int 8FH, Memory Test, clears to 00H and/or tests memory in the 8086 address space with a variable number of patterns. Memory Test passes the parameters in the following list.

In ES:DI, the starting address of the memory to be tested.

In CX, the number of words to test.

In DS:BX, the address of the table of patterns for testing memory. The first word in the table must be 0000H.

In SI, the number of word-size patterns in the table. If SI contains 1, memory is cleared to 00H and is not tested.

Memory Test returns the following parameters:

In AX, XOR of bad data and expected data, if the Carry Flag is set.
Carry Flag -- Set if an error is detected.
Carry Flag -- Cleared if there is no error or memory was only cleared.

All other registers and flags are preserved.

14.14 POWER-ON DEFAULTS

This section lists the power-on defaults used by the Start PROM.

Stack:
SS:SP = FC00:5000H (EA = 01000H)

Timers:
Timer 0 - Set for an interrupt every 10 ms. Interrupt disabled.
Timer 1 - Set for refreshing every 120 us.
Timer 2 - Set for Interrupt 0 every 25 ms. Interrupt enabled.

Interrupt Table:
The Interrupt Table is initialized to all Invalid Interrupt vectors, except for a vector table copied from the Start PROM to RAM starting at Location 00200H and for the NMI interrupt vector.

Interrupts Enabled:
NMI, 1, 2 and 6 (Programmable Interrupt Controller Levels)

Interrupts Disabled:
0, 3, 4, 5 and 7 (Programmable Interrupt Controller Levels)

RS-232 Receive/Transmit Baud Rate:
Automatically set according to the CPU Dip Switch settings as indicated in the following list.

Setting	Baud Rate		
1111	19200		(All Open or Off)
1110	9600		
1101	7200		
1100	4800		
1011	3600		
1010	2400		
1001	2000		
1000	1800		
0111	1200		
0110	600		
0101	300		
0100	150		
0011	134.5		
0010	110		
0001	75		
0000	50	Loop On Power-Up.	(All Closed or On)

14.15 MEMORY ALLOCATION

This section describes Start PROM memory allocation.

Slots 1 through 15:
Fifteen 8-byte mailboxes, 40:00H through 40:77H

Slot 0:
Five 8-byte mailboxes, 40:78H through 40:9FH
A. 40:78H Drive A
B. 40:80H Drive B
C. 40:88H Wang keyboard
D. 40:90H RS-232 port
E. 40:98H Parallel port

Miscellaneous Variables:

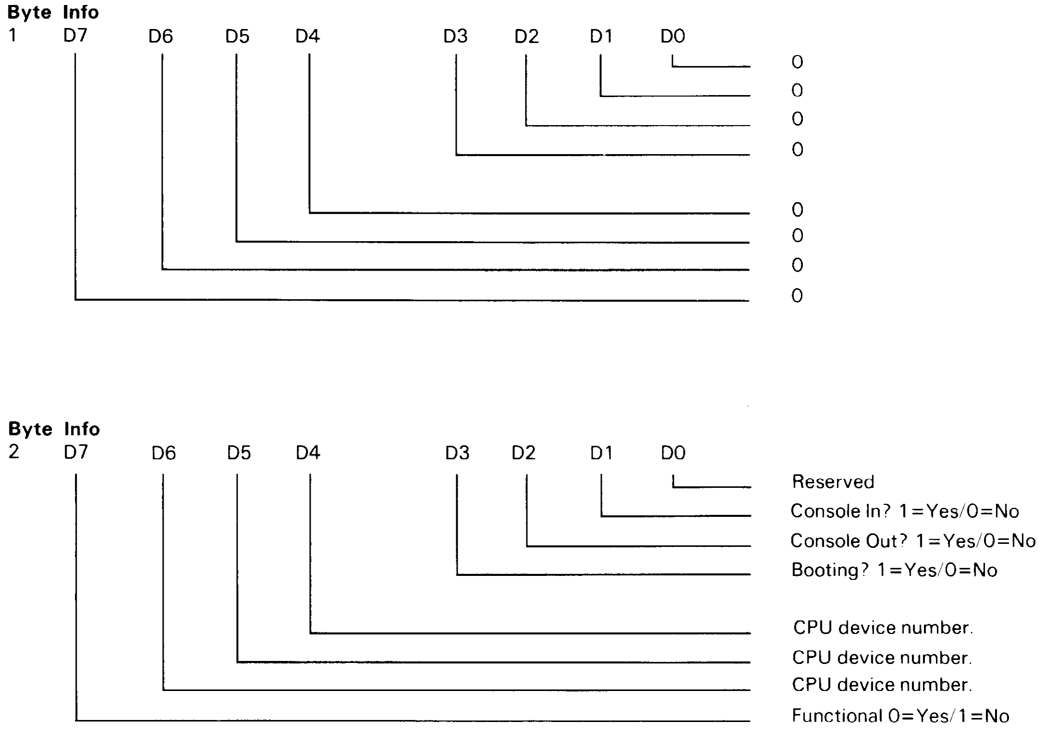
- 40:A0H Reserved. 1 byte. 00H = obsolete or unused.
- 40:A1H Reserved.
- 40:A5H Reserved. Set to FFFFH.
- 40:A7H Start device pointer. One word. Offset, from 40:00H, of the start device mailbox.
- 40:A9H Start mailbox. Eight bytes. A copy of the mailbox of the start device.

14.16 MAILBOX FORMATS

This section gives the formats of the following mailboxes: CPU device, I/O slot, and malfunctioned device.

14.16.1 CPU Device Mailbox Format

Figure 14-1 shows the Slot 00H mailbox format.



f000 0oi0 = Drive 1 (A).
f001 0oi0 = Drive 2 (B).
f010 0oi0 = Wang keyboard.
f011 0oi0 = RS-232 port.
f100 0oi0 = Parallel port.

f = Functional bit
i = Console Input bit

o = Console Output bit
0 = Zero

Figure 14-1. Slot 00H Mailbox Format

Diskette A and B Specifications:

Byte 3 = Number of sectors per track.
Byte 4 = Number of heads per cylinder.
Byte 5 = Relative Sector 0 on disk. Low byte.
Byte 6 = Relative Sector 0 on disk. High byte.
Byte 7 = Sector size. 01H = 256 bytes/sector).
02H = 512 bytes/sector).
03H = 1024 bytes/sector).

Byte 8 = Gap length. 020H = 256 bytes/sector, 16 sectors per track.
02AH = 512 bytes/sector, 8 or 9 sectors per track.
080H = 1024 bytes/sector, 4 sectors per track.

Parallel Port Specifications:

Byte 3 = Reserved. 00H.
Byte 4 = Device connected externally. 00H = Nothing.
01H = Loopback.
02H = Centronics-type printer.
03H = Smart printer (e.g., Wang PC-PM012)
04H - FFH - Reserved.
Byte 5 = Reserved. 00H.
Byte 6 = Reserved. 00H.
Byte 7 = Reserved. 00H.
Byte 8 = Reserved. 00H.

The C/O bit is set in the parallel port mailbox only if a Centronics-type printer is attached.

RS-232 Port Specifications:

Byte 3 = Baud rate (copy of switches)
Byte 4 = Device connected externally. 00 = Nothing.
01 = Loopback.
02 - FF - Undefined.
Byte 5 = Reserved. 00H.
Byte 6 = Reserved. 00H.
Byte 7 = Reserved. 00H.
Byte 8 = Reserved. 00H.

14.16.2 I/O Slot Mailbox Format

Figure 14-2 shows the mailbox format for Slots 01H through 0FH.

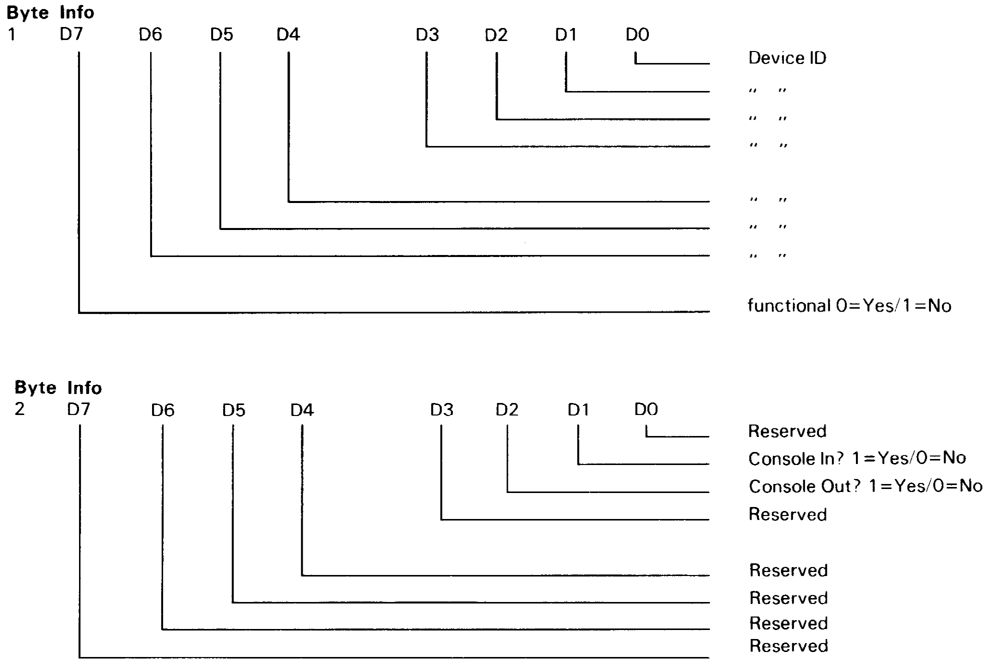


Figure 14-2. Slots 01H Through 0FH Mailbox Format

Winchester Specifications:

Byte 3 = Number of sectors per track.
Byte 4 = Number of heads per cylinder.
Byte 5 = Relative Sector 0 on disk. Low byte.
Byte 6 = Relative Sector 0 on disk. High byte.
Byte 7 = Sector size. 02H = 512 bytes/sector.
Byte 8 = Reserved. 00H.

Low-Resolution Monitor Specifications:

Byte 3 = Current cursor row (0 - 24)
Byte 4 = Current cursor column (0 - 79)
Byte 5 = Bold video flag. Current state. 00H = Normal, 04H = Bold, etc.
Byte 6 = Reserved. 00H.
Byte 7 = Reserved. 00H.
Byte 8 = Reserved. 00H.

Monochrome Monitor Specifications:

Byte 3 = Current cursor row (0 - 24)
Byte 4 = Current cursor column (0 - 79)
Byte 5 = Bold video flag Current state. 00H = Normal, 04H = Bold, etc.
Byte 6 = Font loaded flag. 00H = False, FFH = True.
Byte 7 = Reserved. 00H.
Byte 8 = Reserved. 00H.

Device With OBPROM (refer to Section 14.17):

Byte 3 = Class Code (0=No / 1=Yes)
Byte 4 = Reserved. 00H.
Byte 5 = Offset from CS of copied code (low byte)
Byte 6 = Offset from CS of copied code (high byte)
Byte 7 = Reserved. 00H.
Byte 8 = Reserved. 00H.

14.16.3 Malfunctioned Device Mailbox Format

The specifications for the malfunctioned device mailbox are as follows:

Byte 3 = Low byte of Reason-1 pointer
Byte 4 = High byte of Reason-1 pointer
Byte 5 = Low byte of Reason-2 pointer
Byte 6 = High byte of Reason-2 pointer
Byte 7 = Low byte of Reason-3 pointer
Byte 8 = High byte of Reason-3 pointer

Reason-# is the reason for the failure of the device determined by the power-on diagnostics. There can be up to three reasons or a single reason composed of parts. Refer to Section 14.18.4, Start-up Failure Messages, for details of the messages.

The pointer is an offset from CS. The default is 0000H for no reason (i.e., a null pointer).

14.17 THE OBPROM

The OBPROM scheme allows the creation of future power-on and start-up devices without restricting the design or revising the Start PROM. The scheme is fully implemented only on boards capable of being a start source or of being the only console output at start time.

14.17.1 Controller Design Changes

Implementation of the OBPROM scheme requires two additions to controller designs. The first addition is mandatory for all boards. On all boards, an IN at Port 1XFEH must yield a word in which D0-D6 have the board ID, D7 has the interrupt status, and D15-D8 are a class code (refer to 14.17.3, Class Code Definitions). All controllers, except those in Table 14-7, must return a class code. The default code is 0FFH for a controller that does not implement the second addition to the controller design.

Table 14-7. Controller IDs

Controller	ID
PC System Card	00H
Winchester Controller	01H
Industry-standard Monitor Card	10H
Monochrome Monitor Card	11H
Text/Image/Graphics Controller	13H
Local Communications Option	38H
Memory Expansion Card	3FH
PC CP/M-80 Emulation Card	39H

Any single-board exact replacement for one of the 2-board combinations in Table 14-7 is considered to be part of the list.

The second addition to controller designs applies only to boards capable of being a start source or of being the only console output at start time. These boards must have an OBPROM addressable by the 8086 at EA F4000H (32 KB maximum) after an OUT at Port 1XFAH with D0 = 1. Latching and decoding of Bits D8-D1 to allow a variable mapping location for memory is optional. If a variable mapping location is not implemented, the board must map to EA F4000H. An OUT at Port 1XFAH with D0 = 0, power-on reset, or an OUT at Port 1XFC must remove the PROM from the 8086 memory bus.

The simplest implementation has one PROM and addresses the data at even addresses. For a word read at an even address, the low byte is valid and the high byte unspecified. A word read at an odd address gives unspecified results. The data can be copied into RAM and utilized. The more complex implementation has two PROMS. Code can be addressed by byte or by word.

An OBPROM must contain the following items:

- A checksum of the contents to verify integrity
- The PROM revision level
- A pointer table to the following functions
  - Diagnostic test code
  - Initialization code
  - Start set-up code
  - Specify Start Device code
  - Start Device Read code
  - C/O code

14.17.2 OBPROM Concept of Operation

An OBPROM contains code to diagnose, initialize, start-up from the board, and/or perform console output. At the beginning of the OBPROM, there is a standard header block. Using the length-of-code value in the header, the start code copies the OBPROM data into RAM. The header contains a value for additional buffer space to be cleared beyond the end of code. Thus, code plus data can be longer than the OBPROM itself.

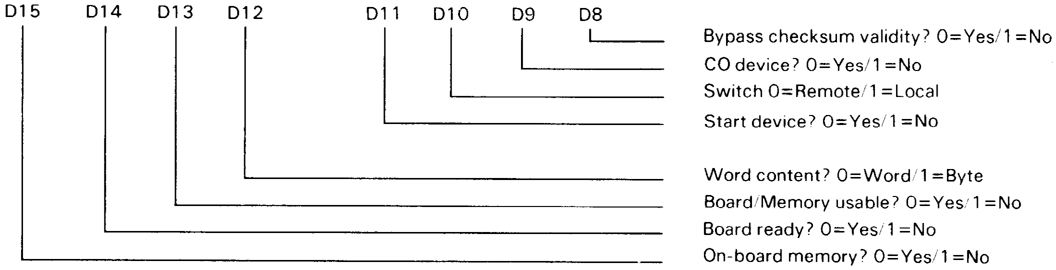
After the data is copied, its checksum is verified and the buffer space is cleared. Runtime information is placed in the header block, and a pointer to the copied code is placed in Parameter 3 of the mailbox for the board's option slot. The diagnostic code from the OBPROM is executed, and the mailbox is updated with the diagnostic results.

All code written for the OBPROM is relative to the start of the PROM. Do not put code and data in the same segment. The Stack Segment Register and Stack Pointer are set by the Start PROM. The stack is from .5 KB to .75 KB. Be sure to restore the stack exactly the way you found it. All registers are saved before any OBPROM code is saved; therefore, you can use them as you wish, always observing the Wang PC Start PROM protocol. Use hardware interrupts cautiously, until you have finished with the Start PROM facilities.

14.17.3 Class Code Definitions

The following class code definitions are for the data as taken from the bus by an IN at 1XFEH. When the data is stored in a mailbox or other data location by the start-up code or power-on diagnostics, the data is inverted such that 1 indicates Yes and 0 indicates No.

Class Code Definitions


D15
D14
D13
D12
D11
D10
D9
D8
Bypass checksum validity? 0=Yes/1=No
CO device? 0=Yes/1=No
Switch 0=Remote/1=Local
Start device? 0=Yes/1=No
Word content? 0=Word/1=Byte
Board/Memory usable? 0=Yes/1=No
Board ready? 0=Yes/1=No
On-board memory? 0=Yes/1=No

D15 On-Board Memory Compatible with Start PROM Scheme. If this bit is 0, the option board contains memory that may be enabled on the bus (at EA F4000H) by issuing an OUT at 1XFAH with Bit D0 set to 1 and Bits D8 through D1 set to F4H. Latching and decoding of Bits D8 through D1 to allow a variable mapping location for memory is optional. If variable mapping is not implemented, the board must map to EA F4000H. Issuing an OUT at 1XFAH with Bit D0 set to 0 disables the memory from the 8086 bus. If this bit is 1, Bits D14 - D8 are ignored.

D14 Board Ready. If this bit is 1, the board is busy resetting, self-diagnosing, or in some way not available for use. While this bit is 1, Bit D13 cannot be considered valid. D13 must be set to its valid state before D14 is set to 0. Even on the simplest dumb board with a PROM, this bit should correctly reflect the state of reset (power-on and 1XFCH) to the board. In any event, if this bit stays at 1 for longer than 60 seconds, the option board will be considered bad.

D13 Board/Memory Usable - If this bit is 1, the option board has been self-diagnosed as bad, or the contents of the on-board memory are invalid. This bit is changeable only if the option board can be self-diagnosed and/or the on-board memory is RAM and data is loaded from an external source under control of the option board, not by request of the Wang PC start-up code. On a dumb board with a PROM, this bit should always be 0. This bit MUST be set to its valid state before D14 is set to 0.

D12 Word Content. If this bit is 1, the option board has data only at even addresses. A word read will yield valid data only in the low byte. If this bit is 0, data exists at both even and odd addresses (e.g., on two EPROMS side-by-side).

D11 Start Device. If this bit is 0, the option board is a valid start source.

D10 Switch. If D11 is 0, D10 is 1 if the Remote/Local switch is in the Local position, and D10 is 0 if the Remote/Local switch is in the Remote Position. If D11 is 1, D10 is always 1.

D9 Console Output Device. If this bit is 0, this option board is a console output device that can be the only console output device in the system.

D8 Bypass Checksum Validation. If this bit is 0, the start-up should bypass the PROM checksum validation algorithm. Use this as a debugging aid only. For manufacturing, this bit must be 1.

14.17.4 Diagnostic Test Code Definition

An OBPROM must contain code to test the option board. The tests must end by clearing memory to all 00H, unless specified to the contrary. Optional time-consuming tests (e.g., memory tests) should be skipped during quick restart by testing the local Quick Restart Flag for 0FFH. Clearing memory to all 00H (the default) or to specified characters is mandatory.

When the test code finishes executing, it should execute a FAR RETURN with the Carry Flag cleared (diagnostic passed) or set (diagnostic failed). If the Carry Flag is set, ERR_MESS_1, ERR_MESS_2, and ERR_MESS_3 must be set as pointers to failure messages or message fragments. The pointers must be offsets from the beginning of the Start PROM to the first byte of the message. If there are less than three messages or message fragments, the unused locations must be set to 0000H. All messages must be prefixed by an international code. Current generic messages offer flexibility and variety so that new message numbers should be unnecessary, though they are allowable. Refer to Section 14.18, Start-time Messages, for the current messages.

14.17.5 Initialization Code Definition

The OBPROM must contain code to initialize the option board to a known state, e.g., Quiet, Non-active, Interrupts Disconnected from bus, DMA Connections Disconnected, etc. Further definition of this state is made by the person designing the start-up console out code. In this state, the option board must not in any way interfere with any other start-up console out device in the Wang PC. The minimum initialization code is an OUT at 1XFCH with AL = 00H. The code must terminate in a FAR RETURN.

14.17.6 Set-up Code Definition

The OBPROM must contain code to set up the option board for the first read. The set-up includes, but is not limited to, activating devices (e.g., turning on the motor in a diskette drive) and setting hardware registers.

The code uses Int 92, Start Device Specify, to set up any variables that can be changed later by the first read. The definition of the Start Device Specify mechanism in Section 14.17.7 must be followed. Even for a null specify operation, Int 92H must be invoked by the set-up code for a proper start-up. The code must terminate in a FAR RETURN.

14.17.7 Specify Code Definition

The OBPROM must contain code to set up variables contained in the copied-code buffer area. These variables can be used by the read code as parameters for each read. For example, the board could be a controller for a diskette drive where bytes/sector, sectors/cylinder, and the number of sides is unknown at start-up time. The system assumes 512 bytes/sector, 8 sectors/cylinder, and single-sided. With these assumptions, the first read of 4 KB is successful. The code from the disk then issues a Start Device Specify command to change the assumptions to reflect the real specifications of the diskette.

In writing this code, the designer sets up registers to hold the values to be specified and executes an INT 92H. Although the specify operation can be null, Int 92H must be executed to set up the start-up process. If the code does not require respecification, the minimum code is a FAR RETURN.

14.17.8 Read Code Definition

The On-Board PROM must contain code to load a file or block of data into the memory location pointed to by ES:SI. For the first system read, this read code is called by an Int 91H, with registers set for Sector 0, 4 KB of data, and three retries. Unused registers should be 0. The initially loaded file or block of data should be reasonably small (4 KB is requested) to allow speed of loading. The first 512 bytes must meet checksum criteria according to the Start PROM specification.

After the read or load is accomplished, successfully or otherwise, this routine must execute a FAR RETURN with registers AH and DL set according to Int\91H specifications. Ending with a FAR RETURN allows operator intervention, redirect, etc. Validation of the loaded data is the responsibility of the read code.

After the FAR RETURN, the Start PROM executes a JMP FAR to the ES:SI address specified in the read. For the data loaded at ES:SI to utilize the built-in utilities of the Start PROM (console output, additional reads, etc.) RAM from EA 400H to EA BFFFH and the interrupt vectors from Int 88H through Int 9FH must not be disturbed.

14.17.9 C/O Character-Handler Definition

The OBPROM must contain code to output the character in AL to the attached output device. This code must terminate in a FAR RETURN.

14.17.10 Data Structure for OBPROMS

The Assembly language data structure for OBPROMs follows. In the structure, a question mark (?) indicates a value supplied at assemble time by the code written for the OBPROM. Zero (0) indicates a value supplied at runtime by the Start PROM. OiCo is short for "offset into the code of".

```
ON_BOARD_PROM       STRUC

LENGTH_OF_CODE      DW      ?       Size Of Code In Bytes -- Rounded Up
                                    to Nearest PARAGRAPH.  Maximum = 8000H
LENGTH_OF_BUFFER    DW      ?       Size Of Buffer In Bytes -- Rounded Up
                                    to Nearest PARAGRAPH.  Maximum = 8000H
REV_EVEN            Db      ?       Rev Number Of Code
REV_ODD             Db      ?       Rev Number Of Code
DIAG_OFFSET         Dw      ?       OiCo DIAGNOSTIC CODE
                    Dw      0       Segment Of Copied Code
INIT_OFFSET         Dw      ?       OiCo INIT CODE
                    Dw      0       Segment Of Copied Code
SPEC_OFFSET         Dw      ?       OiCo SPECIFY CODE
                    Dw      0       Segment Of Copied Code
BOOT_OFFSET         Dw      ?       OiCo BOOT CODE
                    Dw      0       Segment Of Copied Code
READ_OFFSET         Dw      ?       Oico READ CODE
                    Dw      0       Segment Of Copied Code
CON_OUT_OFFSET      Dw      ?       OiCo CONSOLE OUTPUT CODE
                    Dw      0       Segment Of Copied Code
MY_SLOT_NUMBER      Db      0       Slot Number Of This Board Ored With
                                    10H. (11 through 1F)
LOCAL_WARM_BOOT     Db      0       Local Warm-Boot Flag.
DEV_NAME_PTR        Dw      ?       OiCo Device Name String.  May Be
                                    Changed at Runtime.
BOOT_SEG            Dw      0       Segment Number Of START PROM CODE
OFFSET_FROM_BOOT    Dw      0       Offset Of COPIED CODE From START PROM
                                    CODE
MR_FONT_OFFSET      Dw      0       Offset Of Medium-resolution FONT From
                                    START PROM CODE.
LR_FONT_OFFSET      Dw      0       Offset Of Low-resolution FONT From
                                    START PROM CODE.
FONT_HIGHEST        Db      0       Top Character In Either Font.
RESERVED            Db      0       Reserved.

AUX_IDANDFUNCT      Db      0       Auxiliary IDANDFUNCT
AUX_BOOTANDCON      Db      0       Auxiliary BOOTANDCON
ERR_MESS_1          Dw      0       Error Message 1
ERR_MESS_2          Dw      0       Error Message 2
ERR_MESS_3          Dw      0       Error Message 3

ON_BOARD_PROM       ENDS
```

14.18 START-TIME MESSAGES

This section lists the messages displayed at start-time by the power-on and start-up code, or the BIOS. The messages appear in the left-hand column of the page. The right-hand column contains comments that are not part of the messages. In the messages, number signs (#) and items in brackets ([]) are variables whose values are supplied when the system displays the message. Numbers preceded by asterisks appear highlighted on the screen.

14.18.1 Informational Messages

Table 14-8 lists and explains start-time informational messages and their meanings.

Table 14-8. Start-time Informational Messages

Message	Comments
WANG PROFESSIONAL COMPUTER REV. #.##	System identifier, highlighted. REV # is the Start PROM Revision.
Copyright Wang Laboratories, Inc., 1983	Statutory Copyright message.
01 Start From [device]	The system will start from this device. The cursor remains just below the message for a 3-second wait period. The user can redirect, etc., at that time. The cursor then executes a CR/LF.
02 NOW STARTING FROM [device]	Used in Revision 1.00 only.
03 RE-DIRECT START	Used in Revision 1.00 only. Refer to Message 14 for definition and usage.
10 Go	Response displayed after the user presses G during the 3-second delay. Truncates the delay.
11 Retry	Response displayed after the user presses R when a failure has occurred.

Table 14-8. Start-time Informational Messages (continued)

Message	Comments
12 Quick Restart	Response displayed after the user presses Q. The machine will execute a quick restart. Memory will not be diagnosed.
13 Power-Up Diagnostics	Response displayed after the user presses P. Machine will execute a full power-on diagnostic.
14 Re-Direct Start	Response after the user presses D. Allows the user to direct the start-up to the desired device, bypassing the hierarchy.
15 Re-Direct Console Out	Response after the user presses O. Allows the user to direct console output to desired device(s).
16 Re-Direct Console In	Response after the user presses I. Allows the user to direct console input from the desired device.
19 Manufacturing Diagnostic Menu	Response displayed after the user presses M. Activates a menu of tests and sequences for manufacturing use. User may exit only through Restart.
91 Looping On Power-On	The power-on diagnostics are in a loop.
*92 Host Appears Busy. I Will Retry Until You Depress A Key.	This message is printed after 10 seconds if the host appears busy or doesn't respond at all. Retries remain in effect until a key is pressed. At that point, retries abort and processing continues as if the key was pressed during a 3-second delay.

Table 14-8. Start-time Informational Messages (continued)

Message	Comments
93 Input Start Data	Response displayed after a start-up from the keyboard. The user keys in executable hex code, high nibble first, low nibble second. Entering a nonhex digit causes an error. BACKSPACE is illegal. RETURN is legal. EXECUTE terminates entry and executes the code.

14.18.2 Power-on and Redirect Error Messages

Table 14-9 lists and explains the power-on and redirect error messages.

Table 14-9. Power-on and Redirect Error Messages

Message	Comments
*21 [device] In ## Defective	Warning message. The start-up-related device in Slot\## failed the power-on diagnostics. Messages 50 through 69 give the reasons for power-on diagnostic failure.
*22 [device] Defective	Warning message. System card device failed power-on diagnostics. Messages 50 through 69 give the reasons for power-on diagnostic failure.
*23 ILLEGAL RE-DIRECT	Composite of 24 through 27, Revision 1.00 only.
*24 Illegal Device	Illegal device type error.
*25 Illegal Slot Number	The slot number is greater than 15 or is non-numeric.

Table 14-9. Power-on and Redirect Error Messages (continued)

Message	Comments
*26 Illegal Address	Address specified conflicts with copied code or is higher than 128K.
*27 Device Not Available	Device specified is not in slot specified or is not functional.

14.18.3 Power-on Diagnostics Defective Device Reasons

Table 14-10 lists and explains the reasons displayed by power-on diagnostics messages.

Table 14-10. Power-on Diagnostics Defective Device Reasons

Message	Comments
50 Or Not Installed	Keyboard presence cannot be verified.
51 Memory Error [- specific]	An error of the [specific] kind while memory was being diagnosed.
52 Read Error [- specific]	Error reading [specific].
53 Write Error [- specific]	Error writing [specific].
54 DMA Error [- specific]	DMA data transfer error.
55 Status Error [- specific]	A device or LSI chip returned an incorrect or invalid status.
56 Loop Back	Loopback connector attached.
57 System Card Failure	Device-related electronics on the system card appear to have failed.
58 Not Ready [- specific]	[specific] did not become ready when expected.

Table 14-10. Power-on Diagnostics Defective Device Reasons (continued)

Message	Comments
59 Init Failed [- specific]	[specific] failed to initialize properly.
60 Missing [specific]	[specific] signal or status is mandatory but did not occur, for example, Missing Vert Sync.
61 Processor Failure [- specific]	[specific] microprocessor failed.
62 Register Failure [- specific]	[specific] register failed.
63 Invalid [specific]	[specific] is mandatory but invalid. For example, invalid data, invalid command, or invalid status.
64 Failed To [specific]	The option failed to execute the mandatory [specific].
69 RAM Space Exhausted	The device may not be defective, but it could not be tested. There was too little room left in RAM to copy PROM code. This should be a rare occurence, since there is 44 KB of RAM for copied PROM code. If this is a start device, it cannot be used for start-up unless it is tested. Rearranging cards may allow diagnosis of this card, but another device will then have insufficient RAM.

14.18.4 Start-up Failure Messages

Table 14-11 lists and explains the start-up failure messages.

Table 14-11. Start-up Failure Messages

Message	Comments
***40 No Auto-Start Device	All auto-start devices are defective or not installed, or the door is open, etc.
***41 Start Failed	Unsuccessful start-up. Messages 70 through 77 give the reasons for start-up failures.
7# [device] [reason]	
70 [device] Read Error	Read failed due to bad data.
71 [device] No Wang Start Track	The disk does not have a valid Wang PC start-up track.
72 [device] Not Ready	The device was not ready. For example, there was no diskette in the drive.
73 [device] Failure	Catch-all phrase if nothing else applies or failure is strange.
74 [device] Format Error	Read failed due to a format error.
75 [device] Can't Find Start File	Start-up file not found. For Local Communications cards, network cards, and other OBPROM options.
76 [device] Invalid Start File	Start-up file has invalid data. Equivalent in meaning to Message 71 for Local Communications cards, network cards, and other OBPROM options.
77 [device] Invalid Response	Invalid response, according to the specified protocol, by [device].

14.18.5 Fatal Error Messages

Table 14-12 lists and explains fatal error messages.

Table 14-12. Fatal Error Messages

Message	Comments
***42 Invalid Interrupt	Invalid interrupt executed. The system must be powered off and on again to reexecute the power-on diagnostics.
***43 Parity Error	Parity Error detected. The system must be powered off and on again to reexecute the power-on diagnostics.

14.18.6 BIOS Start-up Failure Messages

Messages ***44 through ***49 exist primarily for tight code situations. They can be used instead of the more detailed ***41 Start Failed/7# [device] [reason] messages.

The messages in Table 14-13a issued by the disk start-track code if an error occurs during execution of additional reads beyond the 4 KB read of the start-up code.

Table 14-13a. BIOS Start-up Failure Messages

Message	Comments
***44 SYSTEM FILES MISSING	System file(s) missing from the disk.
***45 DISK READ ERROR	Disk Read Error.
***46 DEFECTIVE START DISK	Invalid FAT Table Data.

The messages in Table 14-13b can be issued by the start-track code for nondisk devices if an error occurs during execution of additional reads beyond the 4 KB read of the start-up code.

Table 14-13b. BIOS Start-up Failure Messages (continued)

Message	Comments
***47 Missing System File [- file name]	System file [file name] is missing.
***48 Read Error [- device]	Read error on [device].
***49 Not Ready Error [- device]	Not ready error on [device].

14.18.7 Device Names Displayed by the Start PROM Messages

The following are the device names displayed in Start PROM messages.

Name	Devices
Drive A	Floppy A
Drive B	Floppy B
Winchester	Winchester Disk
Video Card	Any Video Controller
Local Com Option	Local Communications Option
Option Card	Name for Unidentifiable Option Card
Serial Port	Asynchronous or RS-232-C Port
Parallel Port	Parallel Port
Keyboard	Wang Keyboard

14.19 HELP MENUS

This section presents the menus that appear if the user presses the HELP key. Items in parentheses are comments that do not appear on the screen with the menu.

Each menu option has an international numeric prefix. The right two digits of International Prefixes 101 through 116 correspond to the Special Function keys that produce the letter required for that option (refer to Section 14.8.1, Console Input Specifications). Using an SF key aids translation, since an alphabetic key may not be mnemonic in another language.

The following menu appears if the user presses the HELP key during a 3-second delay or after an error:

```
121 Options:

104 G = Go                    (Optional; active only during a 3-second delay.)
112 R = Retry                 (Optional; active only after an error.)
103 D = Redirect Start
109 O = Redirect Console Out
105 I = Redirect Console In
111 Q = Quick Restart
110 P = Power-Up Diagnostics
```

The following menu appears if the user presses the HELP key during redirect console output. Devices are listed only if they exist and are functional.

```
122 Output Device(s):

115 V = Video
113 S = Serial Port
114 T = Printer

150 and/or RETURN to Exit
```

The following menu appears if the user presses the HELP key during redirect console input. Devices are listed only if they exist and are functional.

```
123 Input Device:

106 K = Keyboard
113 S = Serial Port

151 or RETURN to Exit
```

The following menu appears if the user presses the HELP key during redirect start. Devices are listed only if they exist and are functional.

```
124 Start Device:

101 A = Drive A
102 B = Drive B
116 W = Winchester
108 N = New Devices
106 K = Keyboard
113 S = Serial Port

140 plus optional 2-digit slot number
150 and/or RETURN to Exit
```

14.20 CHECKSUM ALGORITHMS

The checksum algorithm for the first 512 bytes of the start track uses an Add With Carry (Adc) of all 512 bytes and produces a zero result. The following program creates a checksum byte and places it at the end of the 512-byte block:

```
CREATE_SIMPLE_CHECK_SUM:
          Cld                            ;Direction = Increment
          Xor  BX                        ;Clear Accumulator and Clr Carry
          Mov  CX, 511                   ;Do All But Last Byte
          Mov  SI, Offset DGROUP:DATA    ;Point To Data

S_CHECK_SUM_LOOP:
          Lodsb                          ;Get Byte
          Adc  BL,AL                     ;Add In Byte
          Loop S_CHECK_SUM_LOOP          ;Do As Needed

          Adc  BL,0                      ;Add In Zero AND the Carry.
          Neg  BL                        ;Negate Result.
          Mov  [SI],BL                   ;Store Checksum Byte
```

The following program verifies the checksum created by the preceding program:

```
VERIFY_SIMPLE_CHECK_SUM:
          Cld                            ;Direction = Increment
          Xor  BX                        ;Clear Accumulator and Clr Carry
          Mov  CX, 512                   ;Do All 512 Bytes
          Mov  SI, Offset DGROUP:DATA    ;Point To Data

V_S_CHECK_SUM_LOOP:
          Lodsb                          ;Get Byte
          Adc  BL,AL                     ;Add In Byte
          Loop V_S_CHECK_SUM_LOOP        ;Do As Needed

          Jnz  $                         ;Hang if result is not zero
```

The checksum algorithm for the PROM data is more complex than the start track checksum algorithm. The PROM data algorithm starts with a nonzero seed, has a variable operation in the calculation, and is order-dependent. The last byte is the calculated checksum.

The following program verifies the checksum in the even PROM (2764) only:

```
VERIFY_PROM_CHECK_SUM:

          Cld                               ;Direction = Inc
          Mov  AX,0FC00                     ;PROM CODE Segment.
          Mov  DS,AX                        ;Set DS
          Mov  CX,01FF8H                    ;8192-8 Words In PROMS
          Mov  SI,0                         ;Offset of 0 from CS.
          Mov  BL,0FFH                      ;Seed

P_CHECK_SUM_LOOP:
          Rol  BL,1                         ;Rotate AL.  Carry <= D7
          Jnc  P_CHECK_SUM_04               ;IF No Carry THEN Skip Xor

          Xor  BL,8                         ;ELSE Xor Checksum with 08H.

P_CHECK_SUM_04:
          Lodsw                             ;Retrieve Word
          Xor  BL,AL                        ;Xor Byte With Checksum

          Loop P_CHECK_SUM_LOOP             ;Repeat As Necessary

          Test BL,BL                        ;Q: Is Checksum ZERO?
          Jnz  $                            ;A: Hang If Not
                                            ;A: Else Continue.
```

To verify the odd PROM (2732A), modify code to Xor BL with AH instead of AL. The same algorithm is used to verify OBPROM data. Instead of every other byte, all bytes are checked in one pass.

14.21 WANG PC SYSTEM SERIAL NUMBER

The format used for the serial number on each PROM creates a unique 8-byte string by using the time and date of creation and several internal Wang codes, the RDB number, the PROM-burning system master number, and the number of the PROM-burning terminal. Time-of-creation is specified to the second. The master number is greater than 0 and less than 64. The final three bits are mux bits set to zeros.

The internal format of the serial number is as follows:

sec	min	hour	day	month	year	RDB number	master number	unit number	mux bits
6b	6b	5b	5b	4b	7b	16b	6b	6b	3b

CHAPTER 15
SYSTEM BUS INTERFACE

15.1 INTRODUCTION

This chapter explains the Wang PC system bus and the hardware requirements for designing I/O options to interface to that bus. The chapter gives specifications and guidelines that must be adhered to for reliable and trouble-free operation.

15.2 SYSTEM OVERVIEW

The Wang PC utilizes an 8086 processor (main CPU) and an optional 8087 processor (high-speed numeric data processor) as coprocessors operating at a clock rate of 8 MHz. A four-channel DMA controller, with three channels for system use, is provided to allow I/O devices to transfer data to or from memory without help from the processor, except for DMA setup and configuration.

The 8086 and 8087 coprocessors communicate with each other over a local interprocessor bus. They communicate with I/O option boards over the system bus. The DMA controller also drives the system bus. The system bus is the set of signals utilized to transfer address, data, status, commands, etc. between I/O option boards and the processor or system memory. The I/O option boards must properly interface with the system bus to perform their functions.

Only three system bus masters are allowed in the Wang PC: the 8086 and 8087 coprocessors and the DMA controller. The 8086 is normally the bus master. It gives the system bus to the DMA controller or the 8087 in response to a request for a bus cycle via one of the 8086 Request/Grant lines. The DMA controller or 8087 uses the system bus for I/O transfers to and from memory. Then it returns control of the system bus to the 8086 with a release pulse on the 8086 Request/Grant line. Optional I/O boards can be addressed by any of the bus masters, depending on the particular design.

System board memory includes 128 KB to 1 MB of dynamic RAM, which can store system programs, application programs, or data. In addition to the dynamic RAM, a standard 16 KB of EPROM occupies the highest 16 KB in the memory-mapped address space. One parity bit is maintained with every byte of system board RAM to ensure data reliability. Parity is undefined until after a memory location has been written with data; therefore, a parity error can occur whenever a program reads a memory location that was not earlier written with data. Parity errors in system board memory generate a nonmaskable interrupt (NMI) request. Bit 0 will be set to 0 in the System Status Port (Input Port 10E0H).

Memory is physically organized into two 8-bit bytes forming 16-bit data words. Addressing is done by byte (lower or upper) or word. Word addressing activates lower and upper bytes simultaneously.

The Wang PC is designed as a "normally ready" system. Normally ready means that both the processor and the DMA controller go through their normal sequence of cycles unless access time constraints require the addressed device to extend the present cycle by one or more wait states. The mechanism that extends bus cycles is the system-ready (RDY) line. At the proper time, the addressed device may pull the RDY line low to lengthen the present bus cycle by inserting wait states.

The system bus provides several distinct interrupt lines by which the I/O option boards direct the processor to perform a predetermined task in an interrupt service routine. Boards can use interrupts to initiate actions, terminate actions, or report certain conditions to the processor at appropriate times.

Other circuits on the system board are the diskette controller, the EPROM, the 3-channel counter/timer, the parallel port (printer interface), RS-232-C serial port, and the serial keyboard port. These circuits are not explained in this chapter. Refer to the appropriate chapters elsewhere in this manual.

15.3 SYSTEM BUS SIGNALS

The system bus provides 16 bidirectional data lines, 20 address lines, timing and control signals, an error interrupt signal, DMA interface signals, interrupt signals, slot identification codes, and a signal for indicating correct power supply operation. Power lines provide ±5-V and ±12-V power. Interrupt request lines implement six levels of priority interrupt and one nonmaskable "error" interrupt. DMA request and acknowledge signals regulate access to the three available DMA channels. Table 15-1 lists the bus signal names, active levels, whether they are input to or output from the system (CPU) board (Wolftrap and SIDO-3 are outputs from the power supply and the motherboard, respectively), and the functions these signals perform or control. Table 15-2 lists the connector pin assignments for system bus signals on the option board connectors.

Table 15-1. I/O Bus Signals

Signal	Active	I/O	Description
ADSTB	High	O	Address Strobe. Generated by the DMA controller at the beginning of a DMA transfer and at the beginning of a 256-byte boundary to latch a new upper address byte (A8-A15). In DMA single transfer mode, ADSTB occurs with each DMA byte transfer.
/AIOWC	Low	O	Advanced I/O Write Command. The processor or DMA controller issues /AIOWC to write data from the data bus (D0-15) to an I/O device.
ALE	High	O	Address Latch Enable. Generated by the 8288 Bus Controller chip. ALE provides a falling edge that can be used to latch a valid address from the system address bus during processor-generated cycles.
/AMWC	Low	O	Advanced Memory Write Command. The processor or DMA controller issues /AMWC to write data from the data bus to memory or a memory-mapped device.
A0	Low	O	Low-Order Address Line. Generated by the processor or the DMA controler. A0 is active (low) to access the low-order byte (D0-D7) of the 16-bit word addressed by Address Lines A1-A19. To access only the high-order byte, A0 is high and the system bus signal, /BHE, is active. Both A0 and /BHE are active during full-word access.
A1-A19	High	O	Address Bus. Generated by the processor or by the DMA controller. A1-A19 identify words of memory, memory-mapped registers, or I/O ports. A1-A19 implement a 1 MB memory address space (512K words). A19 is the most significant address bit.
/BHE	Low	O	Byte High Enable. Generated by the processor or the DMA controler. /BHE is active low to access the high-order byte (D8-D15) of the 16-bit word addressed by Address Lines A1-A19. To access only the low-order byte, /BHE is high and system bus signal A0 is active (low). Both /BHE and A0 are active during full word access.

Table 15-1. I/O Bus Signals (continued)

Signal	Active	I/O	Description
CLK	High	O	System Clock. From the 8284A clock generator chip. The CLK line runs at 8 MHz with a 125 ns period and an approximate 33% duty cycle. Processor-generated bus cycles are specified relative to the CLK signal.
/DACKO-3	Low	O	DMA Acknowledge Lines. /DACKn is the acknowledgement for DMA request signal /DREQn. /DACKO indicates that the current bus cycle is a dynamic RAM refresh cycle.
DEN	High	O	Data Enable. Generated by the 8288 bus controller. DEN provides a level that can be used to enable data to or from the system data bus during a processor-generated bus cycle.
/DREQ1-3	Low	I	DMA Request Lines. Held low by I/O device to request DMA service. It can be released after the corresponding /DACKn is asserted by the DMA controller.
DT//R	High	O	Data Transmit/Receive. Generated by the 8288 bus controller. DT//R indicates the direction of data flow during a bus cycle initiated by the processor. It is high to transmit (write to I/O or memory) and low to receive (read from I/O or memory).
DO-D15	High	I/O	Data Bus. DO is the low-order bit. DO-D7 comprise the low-order byte, which always occupies an even memory address, while the high (odd) byte is D8-D15.
4CLK	High	O	A 4 MHz clock with a 250 ns period and a 50% duty cycle that is the reference for DMA-generated bus cycles.
GND			Ground lines.
HACK	High	O	DMA Hold Acknowledge. Indicates that a DMA cycle is in progress and that the DMA controller is driving the system bus.

Table 15-1. I/O Bus Signals (continued)

Signal	Active	I/O	Description
/I/O ERROR	Low	I	I/O Device Error. Parity (or other error checking) circuits drive /I/O ERROR low to indicate an error and generate a nonmaskable interrupt (NMI) to the processor.
/IORC	Low	O	I/O Read Command. The processor or DMA controller issues /IORC to read data from an I/O device via the system data bus.
/IRQ2-7	Low	I	Interrupt Request Lines. Used to generate a prioritized interrupt by asserting /IRQn and keeping it low until the processor acknowledges the request. /IRQ2 has highest priority of system bus interrupts when fixed priority is used.
MCE	High	O	Master Cascade Enable. Issued by the processor during the beginning of each interrupt acknowledge cycle.
/MRDC	Low	O	Memory Read Command. The processor or DMA controller issues /MRDC to read data from memory or a memory-mapped device via the system data bus.
RDY	High	I	I/O Device Ready. Memory or I/O devices pull the normally high RDY line low to extend the current bus cycle with wait states: CLK cycles if the processor is bus master or 4CLK cycles if DMA is master.
/RESET	Low	O	System Reset Synchronized with the falling edge of CLK. /RESET is used to initialize system logic during the power-on sequence or after a power outage.
/SAD	Low	O	Slot Address Decode. Indicates that system bus lines A13-A15 are cleared to 0, while A12 is set to 1. This provides I/O option boards with partial I/O decoding. Since /SAD will be active even when a memory address in the X1XXXH format is on the bus, it must be further qualified by /IORC or /AIOWC to read or write an I/O port. Inactive during DMA cycles.

Table 15-1. I/O Bus Signals (continued)

Signal	Active	I/O	Description
SIDO-3	High	O[a]	Slot Identification Pins. Hard-wired to +5v pullup resistor or to ground. SID lines present a 4-bit code that uniquely identifies each slot in the system to the option board placed in that slot.
T/C	High	O	Terminal Count. Output from DMA controller chip. T/C indicates that the chip's word count has been exhausted for the current DMA channel in use.
/WTR	Low	O[b]	Wolftrap. From the power supply. When active, /WTR indicates that DC power is applied.
+12V			Twelve-volt power.
-12V			Minus twelve-volt power.
+5V			Five-volt power.
-5V			Minus five-volt power.

[a] SIDO-3 are outputs from the motherboard into the option boards.
[b] /WTR is an output from the power supply into the option boards.

Table 15-2. System Bus Pin Assignments (Option Slots Only)

Pin	Signal	Pin	Signal	Pin	Signal
1	GND	30	A13	60	D11
2	GND	31	A14	61	D12
3	CLK (8 MHz)	32	A15	62	D13
4	/RESET	33	A16	63	D14
5	-5 V	34	A17	64	D15
6	/IRQ2	35	A18	65	/WTR
7	/IRQ3	36	A19	66	HACK
8	/IRQ4	37	/BHE	67	/I/O ERROR
9	/IRQ5	38	GND (Mon Pwr)	68	ADSTB
10	/IR06	39	GND (Mon Pwr)	69	T/C
11	/IRQ7	40	/DACK 0	70	RDY
12	/AIOWC	41	/DACK 1	71	/SAD
13	/IORC	42	/DACK 2	72	MCE
14	/AMWC	43	/DACK 3	73	+12v (Mon Pwr)
15	/MRDC	44	/DREQ1	74	+12v (Mon Pwr)
16	ALE	45	/DREQ2	75	-12v
17	A0	46	/DREQ3	76	+12v
18	A1	47	DEN_	77	4CLK
19	A2	48	DT/$\overline{R}$	78	+5v
20	A3	49	DO	79	+5v
21	A4	50	D1	80	+5v
22	A5	51	D2	81	SIDO
23	A6	52	D3	82	SID1
24	A7	53	D4	83	SID2
25	A8	54	D5	84	SID3
26	A9	55	D6	85	GND
27	A10	56	D7	86	GND
28	A11	57	D8		
29	A12	58	D9		
		59	D10		

15.4 SYSTEM TIMING

This section presents six timing diagrams that define the time relationships of all system bus signals with respect to the related signals required to perform specific operations. All timing is shown at the system board connector (CPU).

15.4.1 Clock Definition and Synchronization

Figure 15-1 shows the relationship between the 8 MHz clock (CLK) and the 4 MHz clock (4CLK). Figure 15-2 shows the relationship between processor-generated T states, the 8 MHz clock, and the RDY line. Figure 15-3 shows the relationships between DMA-generated S states, the 4 MHz clock, and the RDY line. All times are in nanoseconds. Note that CLK is not a 50% duty cycle signal. However, 4CLK is an approximate 50% duty cycle signal.

CAUTION:
Avoid using 4CLK for processor-generated functions or CLK for DMA-generated functions.

Figure 15-2 shows that if a device addressed by the processor (HACK bus signal is low) needs to extend the current cycle by one wait state (125 ns), it must pull the RDY line low a minimum of 44 ns prior to the falling edge of CLK at the end of State T2. The RDY line must be held low until 2 ns prior to this same clock edge. It must be brought back high at least 44 ns minimum before the next falling edge of CLK (at the end of T3). If more than one wait state is needed, the RDY line is not brought high until the needed processor cycle delay is inserted. The RDY line would then be returned to high a minimum of 44 ns prior to the falling edge of CLK, at the beginning of the last wait state (TW). Note that the top line of Figure 15-2, showing processor T- states, shows only one wait state (TW) inserted.

In processor-generated cycles (HACK low), when an option board's address is present, the board may look for the ALE signal or the I/O and memory control signals (/IORC, /AIOWC, /MRDC, /AMWC) to initiate bringing the RDY line low (refer to Figures 15-4 and 15-5). These signals become active during T1 (ALE) or T2 (/IORC, /AIOWC, /MRDC, /AMWC). During bus signals initiated by the processor, the RDY line must be returned to high synchronously with the CLK signal.

Figure 15-3 shows how an option board can insert one wait state of 250 ns (SW) when the DMA controller is addressing it (HACK is high). When the option board is addressed (its designated /DACKn line is low), it can bring RDY low. Alternatively, it can wait until the DMA controller generates /IORC, /AIOWC, /MRDC, or /AMWC and the board's designated /DACKn line is low. It is important that the activation and deactivation of the RDY signal for DMA wait state insertion be synchronous to 4CLK, as illustrated in Figure 15-3. Refer to Figure 15-6 for timing of the signals during DMA-generated bus cycles.

For a one-wait-state DMA cycle extension, the RDY line must be brought low a minimum of 89 ns prior to the falling edge of 4CLK, at the end of DMA State S3. The RDY line must be held low for a minimum of 77 ns after this same edge of 4CLK. It must be returned to high at least 89 ns prior to the next falling edge of 4CLK, which is at the end of SW. If additional wait states are needed, RDY must be kept low until at least 77 ns into the last required wait state but no later than 89 ns before the end of the last wait state.

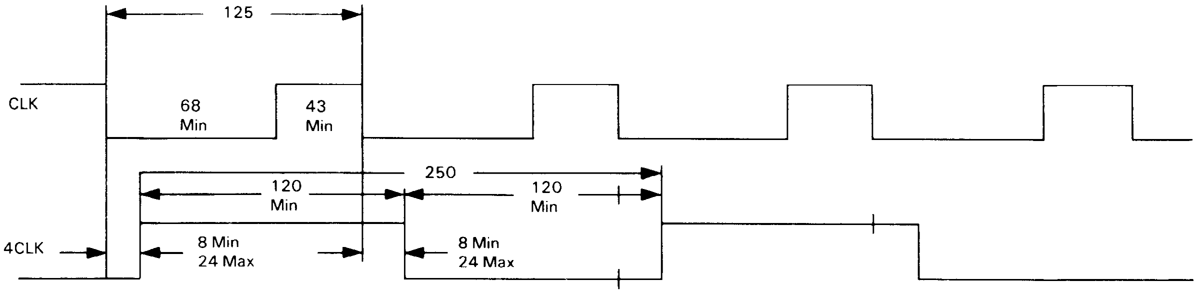


Figure 15-1. Clock Definition

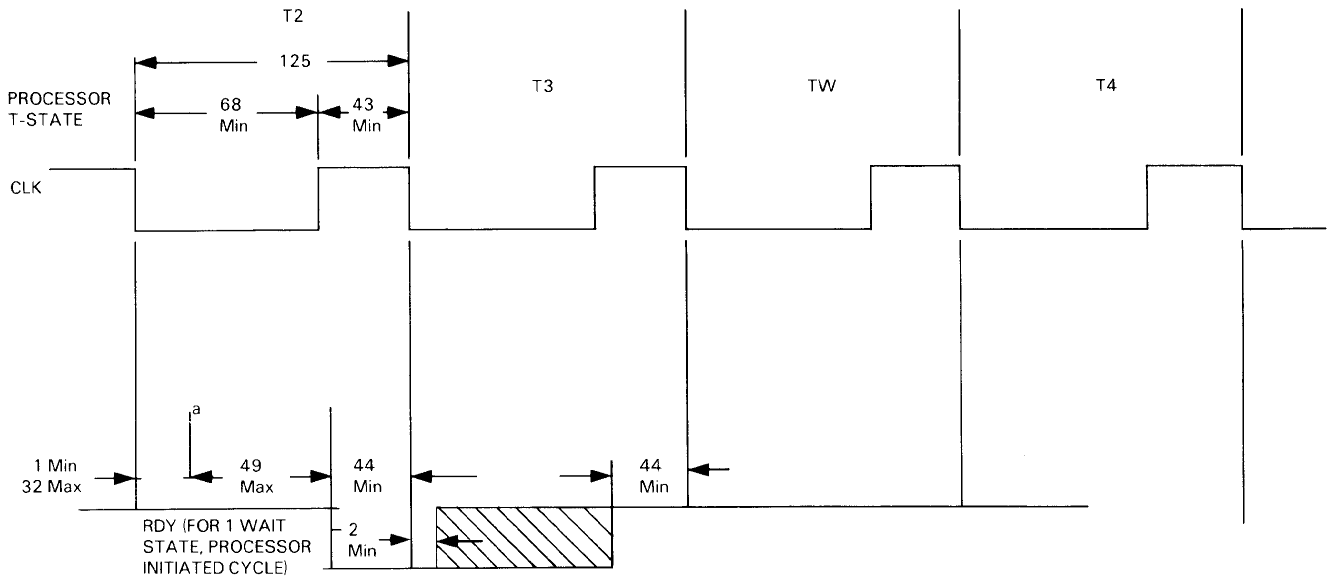


Figure 15-2. Processor, Clock, and RDY Synchronization

[a] During processor-generated bus cycles, I/O and memory control signals (/IORC, /AIOWC, /MRDC, /AMWC, not shown) go active 1 to 32 ns after the start of processor T2 state.

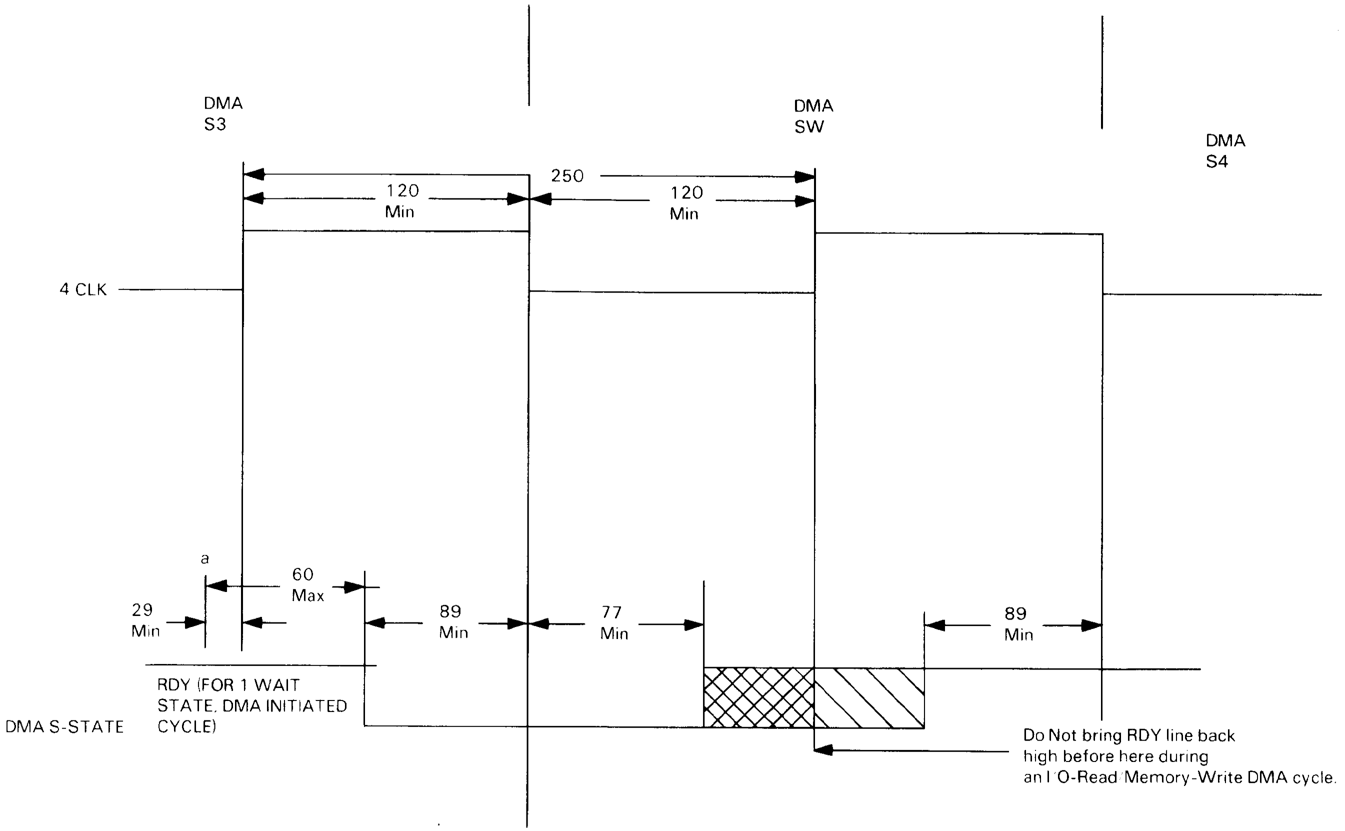


Figure 15-3. DMA, Clock, and RDY Synchronization

[a] During DMA-generated bus cycles, I/O and memory control signals (/IORC, /AIOWC, /MRDC, /AMWC, not shown) go active at least 29 ns before positive 4CLK transition in DMA S3 state.

To insert more than one wait state, the device holds RDY low until at least 2 ns before the beginning of the second from last processor wait state; it releases RDY at least 44 ns before the beginning of the last processor wait state. Once addressed, a device typically pulls RDY low on the trailing edge of ALE (if processor-addressed) or in response to one of the I/O and memory control signals (/IORC, /AIOWC, /MRDC, /AMWC). These signals are used to qualify the address as being valid at the time.

To insert wait states for bus cycle extensions, an option board must know if it is being addressed by the processor (HACK is low) or by the DMA controller (HACK is high). The board then pulls the RDY line low relative to processor or DMA-generated signals, respectively. The option board holds RDY low as long as wait states are required. For processor-generated cycles, the RDY line is returned to high synchronously with the CLK signal (i.e., just after the start of the last 125 ns wait state). For DMA-generated cycles, the RDY line is returned to high synchronously with the 4CLK signal (i.e., just before the start of the last 250 ns wait state). The maximum cycle extension time is 5 us. No I/O option board should ever add more than 40 wait states to a processor-generated bus cycle or 20 wait states to a DMA-generated cycle.

15.4.2 Processor-generated Read Cycle

Figure 15-4 shows a processor-generated read cycle. This can be a memory read cycle (/MRDC low) or an I/O read cycle (/IORC low). One or the other of these two signals will be active to indicate the type of processor read cycle in progress. Figure 15-4 shows minimum and maximum timing parameters for the two types of read cycles. All times are in nanoseconds.

The cycle starts with an ALE (Address Latch Enable) pulse, the trailing (falling) edge of which indicates that a valid address is present on the system address bus. Devices that latch the address should delay slightly (10-15 ns) the trailing edge of ALE as the latching function and use zero set-up time latches. Comparing the Address Bus holding a valid address to the trailing edge of ALE (refer to Figure 15-4) will indicate the reasons for these stipulations.

The DT//R line (Data Transmit/Receive) goes low to indicate that the processor is requesting to receive information. This signal can be used to determine the direction function for transceivers the option board uses to interface with the system data bus for processor-generated cycles. The particular read command type (/MRDC or /IORC) then becomes active (low). The I/O option board can start its read cycle here (with /MRDC or /IORC). When the DEN (Data Enable) signal is high, it can be used to enable onto the system data bus the data requested by the processor from the option board. The /MRDC or /IORC signal could also be used for this purpose.

The processor requires that the data be on the system data bus a minimum of 41 ns prior to the beginning of processor T-State T4. The data must be held on the system data bus for a minimum of 4 ns after the beginning of T4 by the option board. By the end of T4, the data bus must be fully floated by the option board.

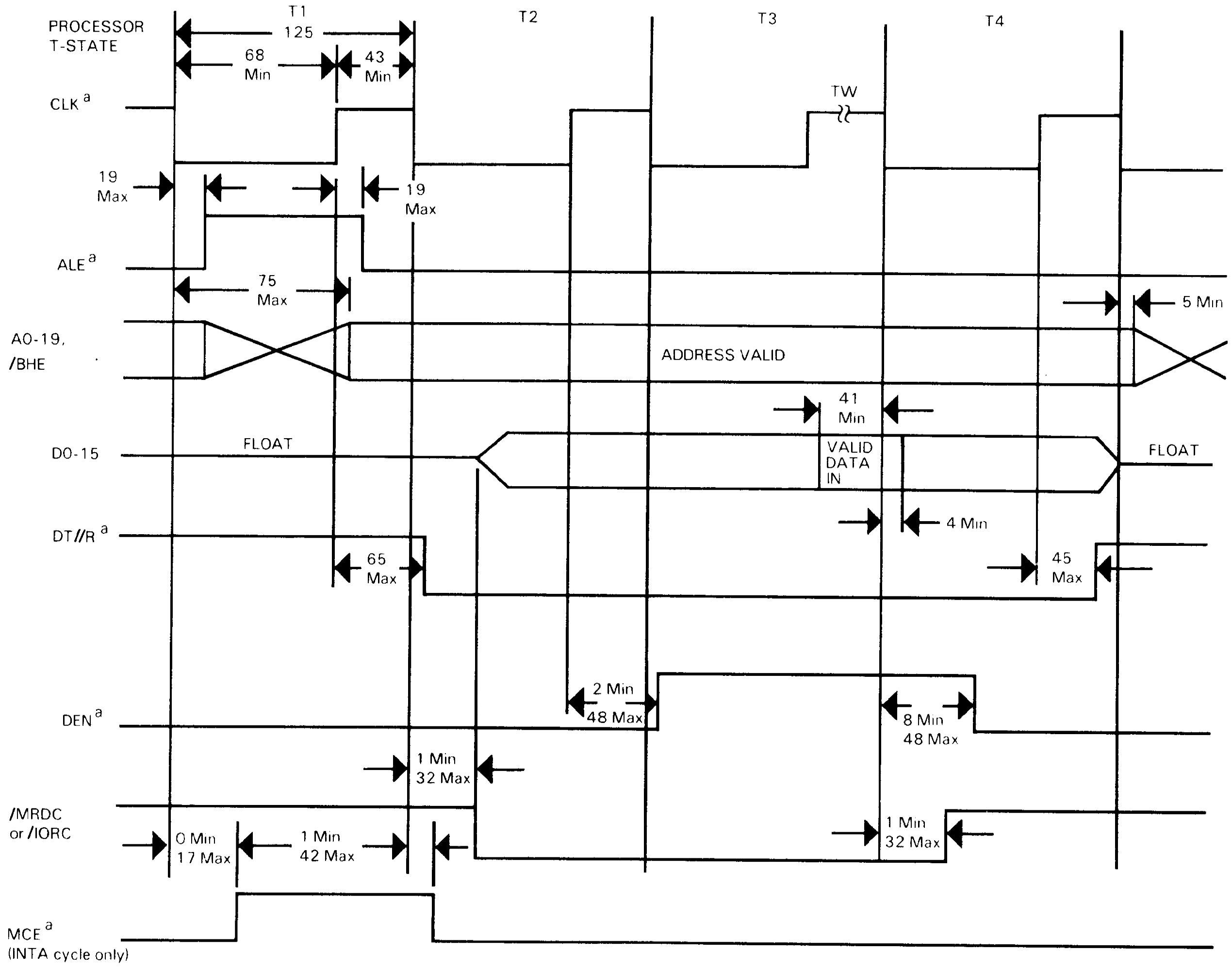


Figure 15-4. Processor-generated Read Cycle

[a] Signal is also also valid for a processor-generated interrupt acknowledge cycle.

NOTE:
An option board should not drive the system data bus during processor-generated read cycles to the board until the board has valid and stable data to deliver to the bus. Unstable data during these cycles can cause multiple high-frequency full transitions between TTL-low and TTL-high states on the bus. The data bus does stabilize by the proper time for the processor, but the high frequency switching causes the system ready line to hang the processor.

15.4.3 Processor-generated Write Cycle

Figure 15-5 shows a processor-generated write cycle. This can be a memory write cycle (/AMWC low) or an I/O write cycle (/AIOWC low). One or the other of these two signals will be active to indicate the type of processor write cycle in progress. The figure shows the minimum and maximum timing parameters for the two types of write cycles. All times are in nanoseconds.

The cycle starts with an ALE (Address Latch Enable) pulse, the trailing (falling) edge of which indicates that a valid address has been placed by the processor on the system address bus. Devices that latch the address should slightly delay (10-15 ns) the trailing edge of ALE as the latching function and use zero set-up time latches. Referring to Figure 15-5 indicates the reasons for these stipulations.

The DT//R line (data transmit/receive) remains high throughout the cycle to indicate that the processor is transmitting data to the addressed device. This signal can be used to determine the direction function for transceivers the option board uses to interface with the system data bus for processor-generated cycles. The particular write command type (/AMWC or /AIOWC) then becomes active (low). The I/O option board can start a write cycle here. However, the data from the processor is not valid on the system data bus until a maximum of 96 ns after the leading edge of /AMWC or /AIOWC. But data is guaranteed valid at the trailing edge of /AMWC or /AIOWC.

Figure 15-5 indicates the period of time that the processor maintains valid data on the system data bus. Data on D0-15 is guaranteed to be valid all through T3 (and TW), until the rising edge of /AMWC or /AIOWC. It remains valid until at least 4 ns after the rising edge of CLK in the T4 state. Notice that write cycle DEN timing is not the same as DEN timing for a read cycle. The DEN signal starts earlier and ends later in the processor write cycle.

The /SAD (Slot Address Decode) signal is not shown in Figure 15-4 or Figure 15-5. It is valid (low) a maximum of 30 ns after the address is valid. It remains valid as long as the address is valid. /SAD is low when the address on the address bus is X1XXX hex (A15-A12 = 0001 binary) and the processor is bus master. HACK is low (the processor is the bus master) if the /SAD signal is asserted. Hardware designers need not further include HACK to qualify /SAD.

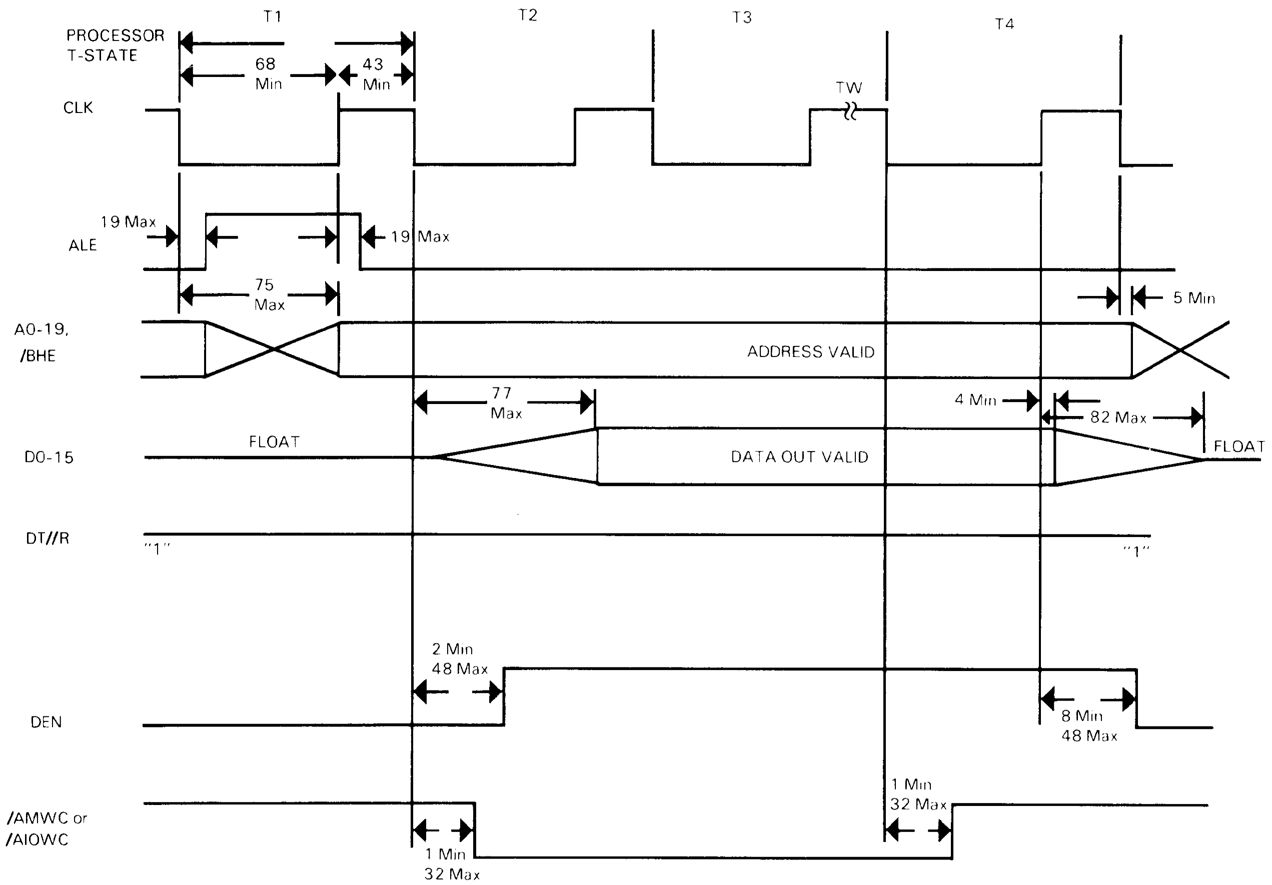


Figure 15-5. Processor-generated Write Cycle

15.4.4 DMA-generated Cycle

Figure 15-6 shows the system bus timing for a DMA-generated cycle, Single-transfer mode only. The figure shows the timing for both memory-read-I/O-write and I/O-read-memory-write cycles. All times are in nanoseconds.

An option board asks the DMA controller to make data transfers for it by pulling its /DREQ line (DMA request) low a minimum of 87 ns prior to the falling edge of 4CLK. This DMA bus cycle can start in less than 1 us from the time the /DREQ is made. Therefore, the option board must not activate its /DREQ line unless it is ready for a DMA bus cycle. The assertion of the /DREQ line must be synchronized to 4CLK.

By convention, the DMA channel assigned to an option board is independent of its interrupt priority level. An input port on the option board typically accepts new interrupt priority levels and DMA channel assignments whenever the 8086 software changes them. Interrupts and DMA capability can be independently enabled or disabled by the 8086 (refer to Section 15.5.2, Use of DMA).

HACK (Hold ACKnowledge) becomes active (high) after the processor has granted the use of the system bus to the DMA controller. However, the option board must wait until its particular /DACK (DMA ACKnowledge) line becomes active (low) to initiate an operation, since the DMA cycle could be for another DMA device. Once the correct /DACK goes low, the option board can release its /DREQ line. It may also keep its /DREQ line active (low) for additional DMA transfers, if it is able to keep up with the transfer rates (refer to Section 15.7, System Bus Utilization).

The DMA controller uses the bus to transfer data between memory (or memory-mapped devices) and I/O devices by bytes, generally returning control of the bus back to the 8086 after each byte has been transferred. In its demand mode, the DMA controller will not give up the system bus until it has reached its programmed count or until the option board releases its /DREQ line, that is, stops demanding DMA service. Single-byte transfer mode is preferred. Unless the proper system timing and bus utilization calculations are made, multiple-byte DMA transfers will interfere with the floppy disk controller or memory refresh circuitry and prevent proper system operation. Take note of the timing requirements described in Section 15.7, System Bus Utilization, before using demand mode.

In Demand mode, to stop DMA before the specified count, the option board must return its /DREQ line to high a minimum of 90 ns prior to the end of S3 of the current DMA cycle. In Demand mode, the S1 state is eliminated (leaving only S2, S3, and S4) until a 256-byte address boundary is crossed.

The DMA controller places a memory address on the address bus. Different parts of the address become valid at different times. For an I/O-write/memory-read DMA cycle, the DMA controller asserts the /MRDC and the /AIOWC lines as shown in Figure 15-6. The memory (or memory-mapped device) must retrieve data from the addressed location and place it on the data bus within 158 ns after the start of the S4 state (as indicated on the fourth timing line from bottom of Figure 15-6). This data must remain valid until at least 103 ns after the rising edge of 4CLK in the S4 state, or 24 ns after the rising edge of /MRDC. The I/O device on the option board should latch this data (write to the I/O device) at the trailing (rising) edge of /AIOWC.

An I/O-mapped device that receives DMA data must begin accepting the data no earlier than 158 ns after the falling edge of 4CLK at the start of the S4 state. It must finish accepting the data no later than 24 ns after the rising edge of /MRDC. A memory-mapped device might use /MRDC to gate data onto the data bus, and an I/O-mapped device might use /AIOWC to latch data from the data bus.

If the DMA-generated bus cycle is a memory write (I/O read) cycle, the DMA controller places a memory address on the system address bus as before but, instead, asserts /AMWC and /IORC, as shown in Figure 15-6. According to the /DACK signal that is active, the I/O device or option board should place data onto the system data bus as indicated on the bottom timing line of Figure 15-6. This data is then written into memory at the address that is on the system address bus. The I/O-mapped device that supplies the data can extend the DMA-generated cycle by injecting wait states if necessary. It must place valid data on the data bus within 26 ns prior to the falling edge of 4CLK at the start of the S4 state, and the data must remain valid for at least 62 ns after the falling edge of 4CLK. The I/O device or option board must fully float the system data bus by 96 ns after the middle of DMA State S1.

All DMA transfers are byte transfers. During a memory read cycle (I/O write), memory or a memory-mapped device places a low-order byte (even byte address) onto D0-7 when A0 is active (low). It places a high-order byte (odd byte address) onto D8-15 when /BHE is active (low). The I/O-mapped data recipient must accept bytes of data from D0-7 or D8-15 as determined by address bus signals A0 and /BHE.

The procedure is reversed during I/O read (memory write) cycles, but with one important difference. Since system board memory uses A0 and /BHE to determine whether D0-7 or D8-15 contains each byte, an I/O-mapped data source can place identical data onto both sets of data bus lines without being concerned about whether the current byte is a high- or a low-order byte.

The T/C (terminal count) signal becomes active when the programmed number of DMA transfers has been completed (at the end of the last transfer cycle for the currently active DMA channel).

During DMA, Address Lines A16-A19 are not guaranteed valid when /MRDC or /AMWC control signals go active (low). A convenient point to consider them valid is the next rising edge of the 4CLK signal following /MRDC or /AMWC active. Also consider all address lines valid at this point during Demand mode DMA transfers.

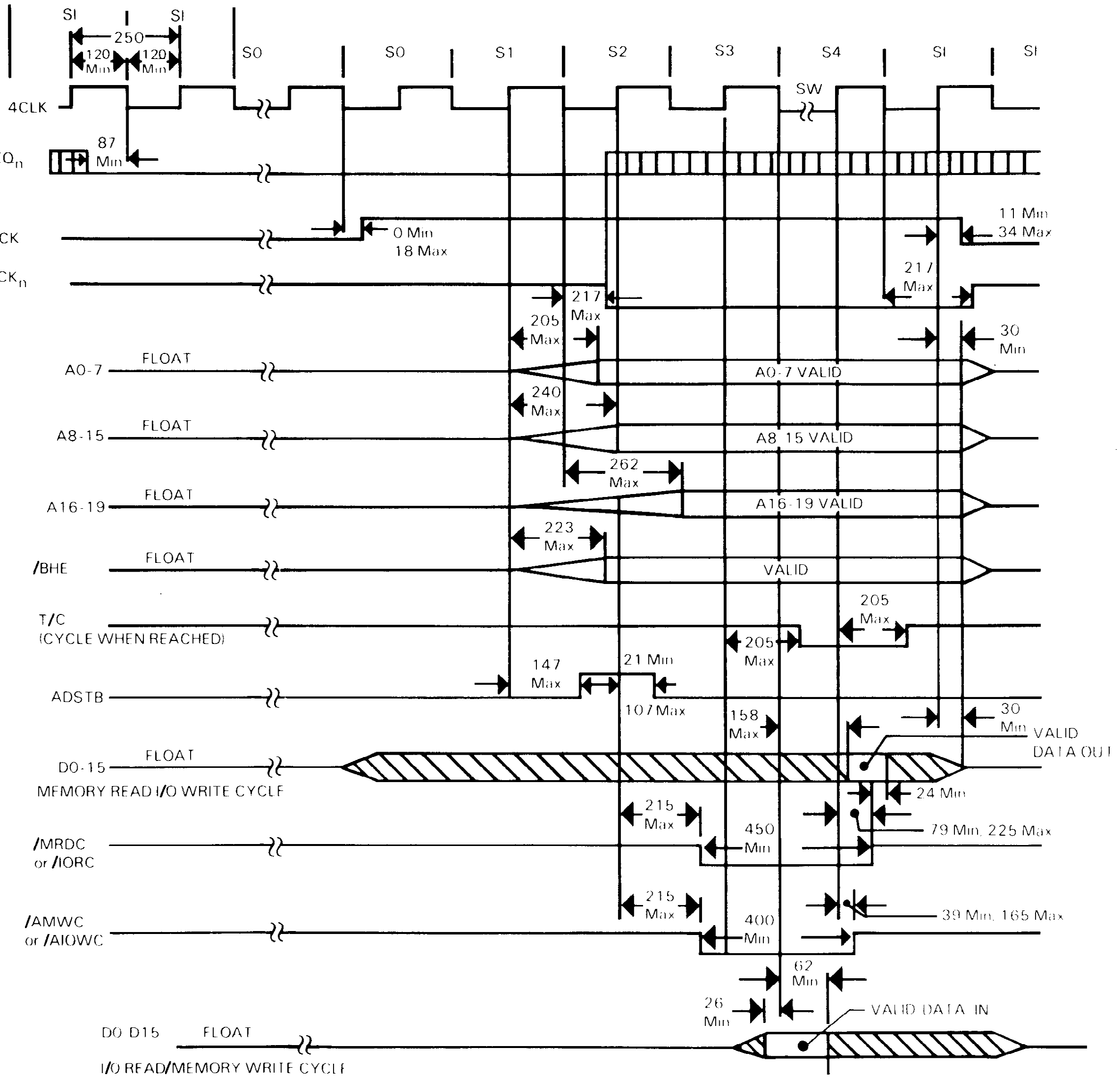


Figure 15-6. DMA-generated Cycle (Single-byte Mode)

[a] /DREQ can be released after /DACK, as shown by the shaded portion of the timing diagram; or /DREQ can be kept active for additional transfers. Refer to Section 15.7, System Bus Utilization, for system bus bandwidth restrictions for the system operating environment using DMA transfers.

Due to timing difficulties in specific implementations, consider using /MRDC as the strobe signal for the I/O write in a DMA memory-read-I/O-write cycle.

15.5 SYSTEM BUS INTERFACE CONSIDERATIONS

This section discusses some of the general requirements for interfacing an option board to the system bus.

15.5.1 Memory and I/O Mapping

The 8086 uses two different types of addressing. Memory reference instructions generate memory-mapped addresses. Input and output instructions generate I/O-mapped addresses that identify I/O ports. When the system address bus contains an address on A1-19, other bus control lines determine whether the address identifies a memory location or an I/O port. HACK must be low in both cases.

The processor addresses memory-mapped functions by means of memory-read (/MRDC) and/or memory-write (/AMWC) cycles. The address the processor places on the system bus during these cycles is a memory address, not the address of an I/O device. Refer to Figure 15-7 for the Wang PC memory map.

The processor addresses I/O-mapped functions by means of I/O-read (/IORC) and I/O-write (/AIOWC) cycles. The address the processor places on the system bus during these cycles is the address of an I/O device, not a memory address. Refer to Figure 15-8 for the Wang PC I/O map.

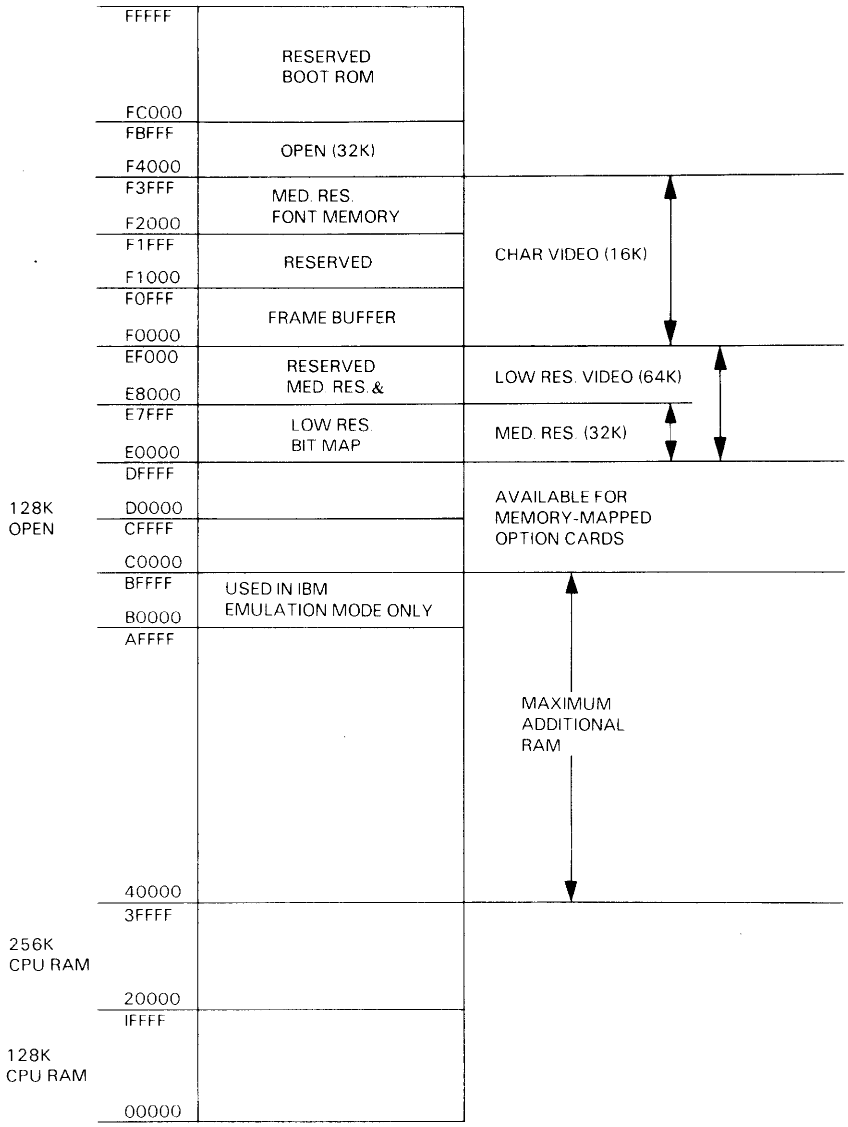


Figure 15-7. Memory-mapped Addressing

	2000	RESERVED DO NOT USE	
1FFF	1F00	SLOT 15	RESERVED REFER TO SECTION 15.5.5
1EFF	1E00	SLOT 14	
1DFF	1D00	SLOT 13	
1CFF	1C00	SLOT 12	
1BFF	1B00	SLOT 11	
1AFF	1A00	SLOT 10	
19FF	1900	SLOT 9	
18FF	1800	SLOT 8	BASE SYSTEM
17FF	1700	SLOT 7	
16FF	1600	SLOT 6	
15FF	1500	SLOT 5	
14FF	1400	SLOT 4	
13FF	1300	SLOT 3	
12FF	1200	SLOT 2	
11FF	1100	SLOT 1	
10FF	1000	SLOT 0	CPU
0FFF	0100	RESERVED DO NOT USE	
00FF	00F8	RESERVED DO NOT USE	
00F7	0000	RESERVED DO NOT USE	

Figure 15-8. I/O-mapped Addressing

NOTE:
Each option board slot has 128 dedicated words of I/O-mapped address space for I/O ports. Therefore, actual I/O port addresses depend on the slot in which the board resides.

Any addressable device can be designed to respond to a memory-mapped address or an I/O port address. In general, however, addresses on I/O option boards are I/O-mapped because I/O ports are identified in terms of the expansion slot in which a circuit board is installed. If I/O option boards contain memory-mapped addresses, those addresses remain the same no matter where the circuit board is installed. They must be reserved for their intended purpose, and special circuitry must disable those addresses when they are not needed.

An example of a memory-mapped device on an I/O option board is the video memory on the various video controller options. Video memory occupies memory-mapped addresses that are reserved exclusively for its use, and each video controller provides a means of disabling its video memory by writing to certain I/O ports on the option card.

Figure 15-9 is a block diagram of the system board I/O decoding logic. The diagram shows how port assignments are established for parallel printer interface, keyboard interface, and various specialized devices that occupy system board space. Printer, keyboard, and RS-232 connectors are shown at the right of the diagram. Remaining port address lines exit the diagram at the bottom.

From a programming standpoint, the major difference between memory-mapped addressing and I/O-mapped addressing is that a large number of memory reference instructions are available for accessing memory-mapped addresses, but only a few I/O instructions serve to access I/O ports. Memory reference instructions can access words or bytes. I/O port addresses generally reference 16-bit words, although this is not a requirement and, in fact, relatively few I/O ports are a full 16 bits wide.

CAUTION:
Memory-mapped devices must qualify their decoded addresses with the HACK signal low when the processor is communicating with them and with their allocated /DACKn signal low when the DMA controller is communicating with them. This prevents a problem from developing as a result of address lines not being valid when an /MRDC or /AMWC is issued during dynamic RAM refresh (/DACK0 active, low).

This Page Intentionally Left Blank

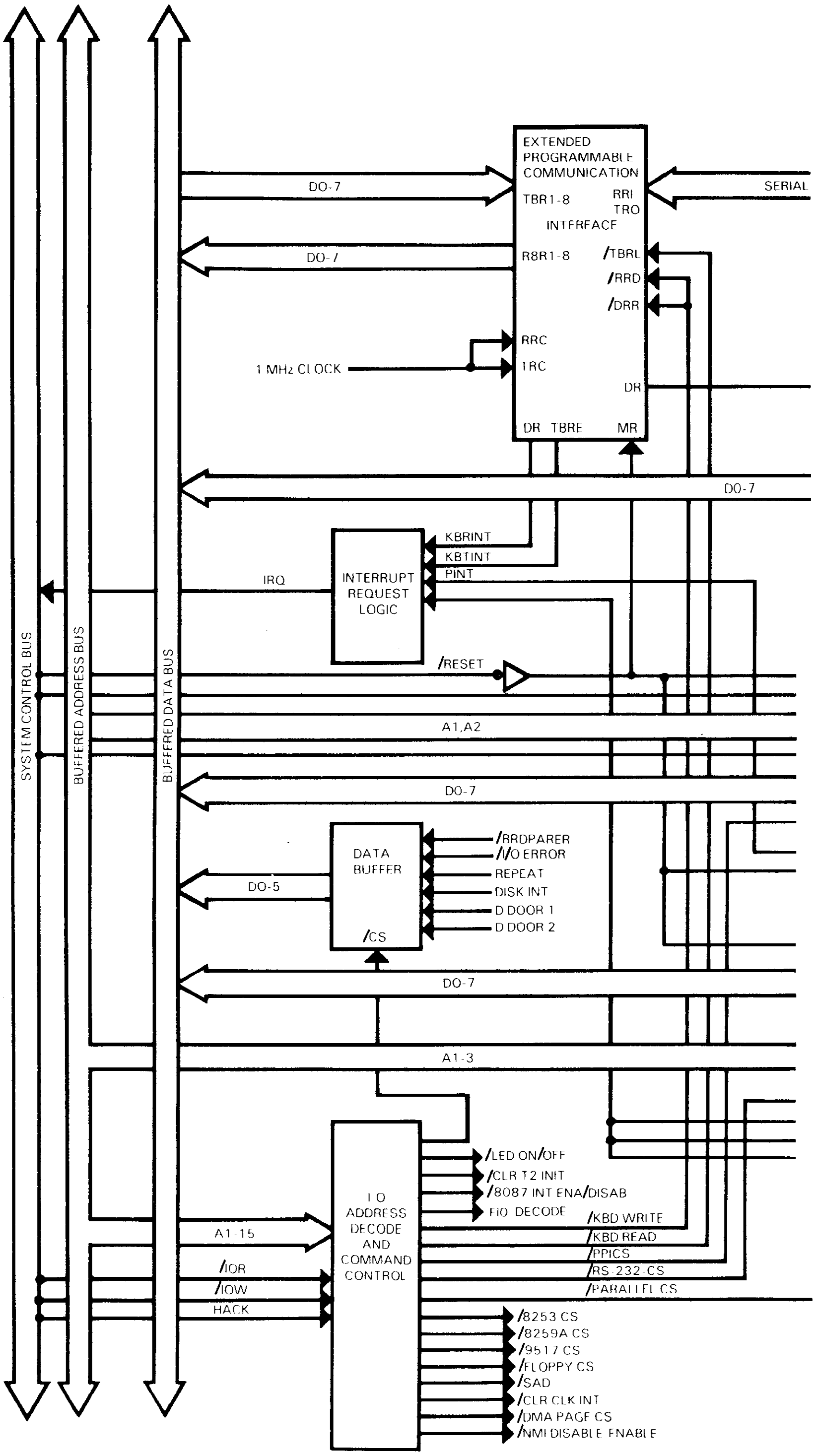


Figure 15-9. System Board I/O Decoding

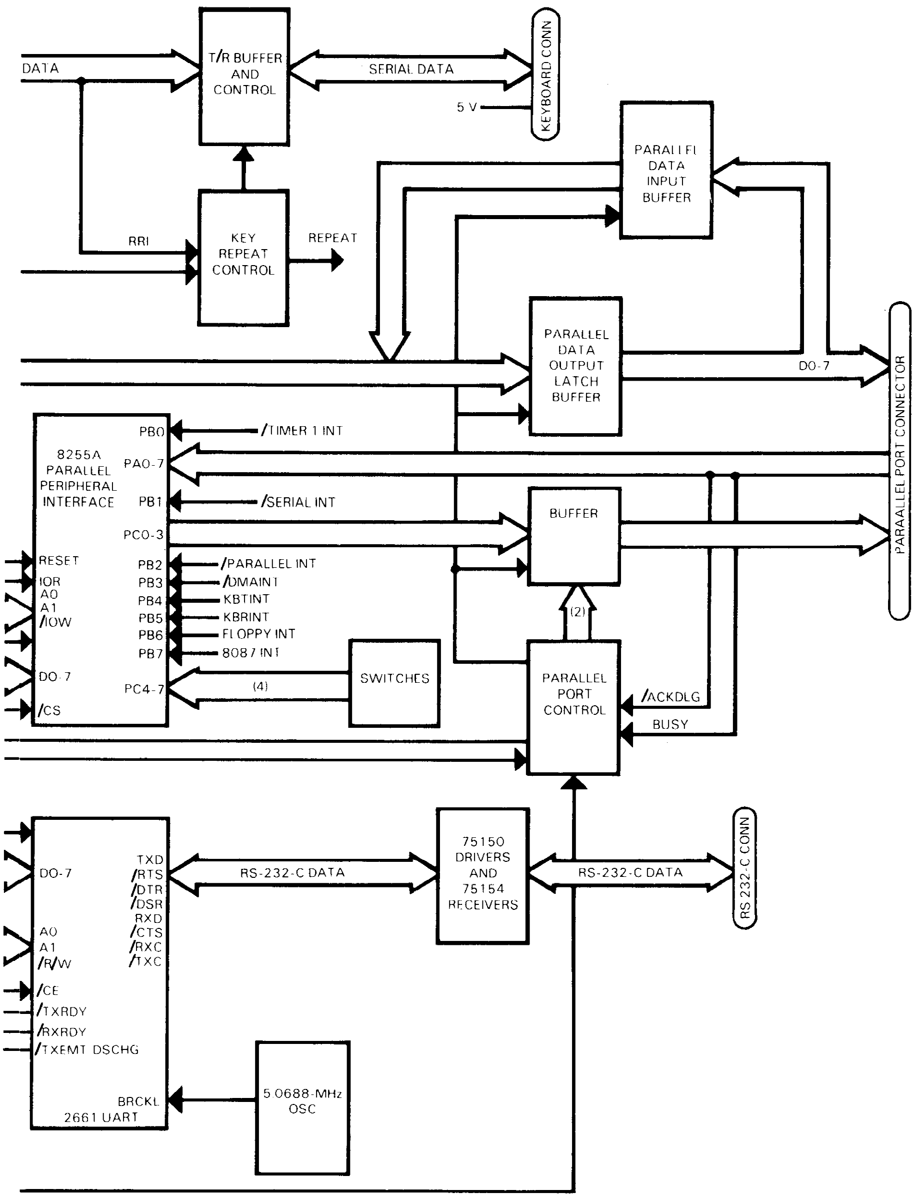


Figure 15-9. System Board I/O Decoding (continued)

15.5.2 Use of DMA

The DMA controller has no connection with mapping. During a DMA-generated cycle (that is, when HACK is high), the address bus always has a memory address for a device or set of devices in the processor's memory map. In a DMA data transfer, the other participating device is an I/O device with respect to the DMA controller. This DMA I/O device has no system address as such. It is activated from the DMA controller by its particular /DACK (DMA ACKnowledge) line. When the I/O device has been activated, DMA provides direct memory access from the device to memory or from memory to the device. Memory-to-memory or I/O-to-I/O transfers cannot use a DMA channel. They must be performed by the 8086.

The 4-channel DMA controller allocates Channel 0 (Demand mode) to refresh dynamic RAM. DMA transfers byte data only. Option boards that use DMA must include a byte-swap mechanism. The option board hardware must place the byte (or accept the byte) on the proper half of the system data bus according to the A0 and /BHE lines.

All DMA data transfers use normal timing of the 9517A DMA controller (not compressed timing) and Extended Write mode. Single Transfer mode is preferred. Demand mode can be used if bus utilization is properly calculated (refer to Section 15.7, System Bus Utilization, and to the discussion of Demand mode in Section 15.4.).

The maximum single-byte DMA transfer rate is 300 KB per second. The maximum processor latency is the longest locked instruction (refer to Section 15.7.1, Processor-DMA Transactions), that is, 22 1/2 us. The DMA controller is programmed for rotating priority.

DMA page registers are 4-bit registers allocated to DMA Channels 1, 2, and 3. These registers allow direct memory access to the entire 8086 memory address space by supplying the four high-order bits of a 20-bit DMA address (A16-19). Before beginning a DMA transfer, software writes the A16-19 page address bits into the page register for the DMA channel being used. After software initializes the DMA page register and programs the DMA controller, the option board can assert its /DREQ line (DMA request). When the board receives a /DACK on its assigned /DACK (DMA ACKnowledge) line, the DMA transfer begins.

There are generally several devices capable of activating each DMA request line. Therefore, open-collector or tri-state devices must be used. If tri-state devices are used, their outputs must never be driven high. The two logic states permitted on DMA request lines are low and open-collector or tri-state.

Hardware does not change the DMA page register contents during a DMA transfer. After accessing the last word on a DMA page, DMA logic accesses the first word on the same page. To perform a DMA transfer across a 64 KB DMA page boundary, software must set up two separate DMA operations, using two different DMA page register values. For this reason, an option board must not use T/C (Terminal Count) to indicate that the DMA transfer is complete. The board must maintain its own count of the number of transfers.

Option boards using both DMA and Interrupts will do so with independently activated functions. The option board should have a mechanism to independently turn the DMA or interrrupt channel on or off. There will be times when no DMA channel will be assigned but interrupts are still permissible and vice versa. Interrupts are assigned to a channel from 3 to 7.

Option boards using DMA or interrupts or both must have latches to which the processor can write to allocate the corresponding DMA and interrupt channels to the option board. Also, the processor must write to a latch on the option board to specify the DMA transfer direction or to inhibit the option board from issuing interrupts or from issuing DMA requests or both.

15.5.3 Use of Interrupts

Option devices can generate processor interrupt requests by activating one of the interrupt request lines. Only devices on the system board can generate Level 0 and Level 1 requests. The /IRQ0 and /IRQ1 lines are not carried out to the system bus. Option boards route their interrupt requests to the interrupt controller by means of System Bus Lines /IRQ2-7.

There are generally several devices capable of activating each interrupt request line. Therefore, open-collector or tri-state devices must be used. If tri-state devices are used, their outputs must never be driven high. The two logic states permitted on interrupt request lines are low and open-collector or tri-state.

Because interrupts are triggered by the signal level, a device must drive its interrupt request line low and hold it low until the 8086 clears the interrupt request. The request can be cleared by an output instruction to a specified I/O-mapped port on the option board or by sending a command to a command register on the option board, if the board has such a register.

The interrupt controller monitors all interrupt request lines and keeps track of which lines are carrying active interrupt requests. When any interrupt request line is active, the interrupt controller asserts its pending interrupt output (INT), which is applied directly to the interrupt request (INTR) input of the 8086. To acknowledge that it is ready to service a pending interrupt, the processor returns two Interrupt Acknowledge (/INTA) pulses to the interrupt controller.

An MCE pulse is generated at the beginning of each Interrupt Acknowledge cycle and is presented to the System Bus as a means for option boards to identify this cycle. The interrupt controller then indentifies the active request that has highest priority, builds an interrupt vector from the level of the highest priority request, and sends the interrupt vector across the data bus to the 8086. The interrupt controller can be programmed to use either fixed or rotating priority.

An interrupt vector invokes an 8086 interrupt service program. The program must identify the source of the interrupt request by interrogating each device that is capable of requesting an interrupt at the priority level it services. Once the interrupt handler locates the device requesting service, the handler performs any functions that are needed and causes the device to remove its interrupt request. The request is cleared by one of the two methods listed previously. The interrupting option board will set Bit 7 in its option identification code byte and bring Data Bus Line D7 high when the ID code is requested (I/O Read) by the processor.

The fully nested interrupt structure allows a high-priority interrupt request to generate an interrupt even while a lower-priority interrupt is being serviced, provided that the low-priority interrupt handler runs with interrupts enabled. In general, at any point in time, the 8086 can be in the process of servicing several active interrupts while several interrupt requests are pending in the interrupt controller. Devices that use both DMA and interrupts should maintain the DMA channel and interrupt number as completely independent functions.

15.5.4 Use of the Ready Line

When a device addressed by the processor (by memory-mapping or I/O-mapping) or activated by the DMA controller (its /DACK is active) cannot respond within the time specified in Figures 15-4 through 15-6, the device can extend the cycle addressing or activating it by pulling the RDY line low. Wait states are inserted with the command lines remaining active until the RDY line is returned to high. Thus, the option device gets this additional time to respond. Refer to Figures 15-2 and 15-3 and to Section 15.4 for the timing and functionality of the RDY line. Note that the RDY line should not be held low longer than 5 us.

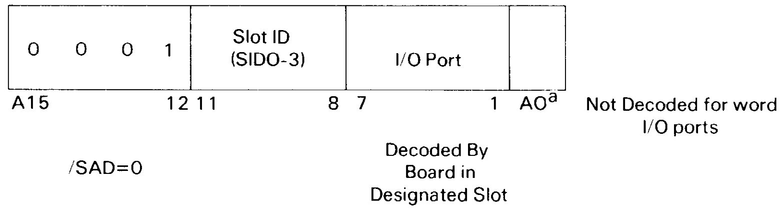
15.5.5 I/O Slot Decoding and Option Codes

No switch settings are needed at power-up to establish a system I/O configuration. Instead, four bus pins present unique slot identification numbers (SID0-3) to the circuit boards that are installed in each of up to 15 option slots. The system board is considered as occupying a sixteenth slot designated as Slot 0. The Wang PC currently uses up to eight option slots.

Option boards determine which expansion slot they occupy by reading the 4-bit slot ID code on System Bus Lines SID0-3. The boards identify themselves to the processor by returning unique 7-bit option identification codes (ID0-6) at designated I/O port addresses.

The slot address decode (/SAD) bus line is active (low) during processor-generated cycles (HACK is low) if the system address bus carries an address of X1XXX Hex (X1000-X1FFF). When /SAD is active, an option board decodes A1-7 and reads or writes the data bus if the following are also true: /IORC or /AIOWC is active, and A8-11 match the unique slot number the option board sees on SID0-3. The lower seven address lines give an option board up to 128 dedicated I/O ports. Each port is up to 16 bits wide, but not all bits must be used. Unassigned bits and unassigned I/O ports return arbitrary data when the 8086 reads them. However, designers should make every effort to avoid unused (i.e., undriven) bits for any supported I/O port.

Port addresses range from 1000H (lowest word port at Slot 0) to 1FFEH (highest word port at Slot 15 (refer to Figure 15-10). Option boards with byte-wide ports may use A0 and /BHE to enable the proper byte for a total of 256 byte-wide I/O ports per option board.


0 0 0 1
Slot ID (SID0-3)
I/O Port
A15
12 11
8 7
1
A0[a]
Not Decoded for word I/O ports
/SAD=0
Decoded By Board in Designated Slot

[a] A0=0=Lower Byte
/BHE=0=Upper Byte
A0=/BHE=0=Both Upper and Lower Bytes

Figure 15-10. I/O Port Addressing

Slot Identification Code Signal Lines SID0-3 are tied to a ground or a 5-volt pullup resistor on the motherboard. Option boards must not draw more than 400 uA per SID line when in the high state.

Each option board gates its unique identification code (ID0-6) onto System Data Bus Lines D0-6 in response to a read at its highest word I/O port address. For example, I/O Port Addresses 1100-11FF belong to the board in Slot 1. This returns its identification code at I/O Port Address 11FE. I/O Port Addresses 1F00-1FFF belong to the board in theoretical Slot 15, which returns its identification code at I/O Port Address 1FFE.

The 7-bit option identification code allows up to 128 different options. Table 15-3 lists the current option identification codes. Codes 60H to 6FH are reserved for boards manufactured by vendors other than Wang. All codes other than 60H to 6FH are reserved for Wang use.

Table 15-3. Option Identification Codes

Circuit Board	ID Code (IDO-6)
PC System Card	00H
Winchester Disk Controller	01H
Reserved	02H
Color Monitor Card	10H
Industry-standard Monitor Card	10H
Monochrome Monitor Card	11H
Wang/IBM Monochrome Emulation	11H
Reserved	12H
Text/Image/Graphics Controller	13H
Reserved	14H
Reserved	15H
Reserved	15H
IBM PC Color Emulation	16H
Reserved	17H
Reserved	1CH
Reserved	1DH
Reserved	1EH
Multiport Communications Option	1FH
Reserved	20H
Reserved	21H
Reserved	22H
Reserved	23H
Reserved	24H
Reserved	25H
Reserved	26H
Reserved	2EH
Reserved	2FH
Local Interconnect Option	30H
Reserved	32H
Local Communications Option	38H
PC CP/M 80 Emulation Card	39H
Reserved	3CH
Reserved	3EH
Memory Expansion Card	3FH
Scanner/Printer Card	40H
Laser Printer Controller	41H
Reserved	42H
Reserved	43H
Reserved	44H
Reserved	45H
Reserved	46H
Reserved	47H
Reserved	70H
Reserved	71H
Reserved	78H

I/O Ports 1000-10FEH are system board I/O ports, although the system board does not occupy a slot and, in fact, there is no Slot 0. An identification code of 7F indicates that a slot is not occupied (D7 - D0 = FF). When gating its 7-bit identification code onto D0-6, an option board brings D7 high if, and only if, it has a pending interrupt request. Software interrogates this interrupt request status flag when polling devices to identify the source of a particular interrupt.

An option board that is the equivalent of two boards can use a technique called "pseudo-slot addressing." The Wang PC recognizes Slots 0 through 15, with Slot 0 reserved for the system card. The largest system currently offered for the Wang PC has eight physical slots (s = 1 to 8). Therefore, Slots 9 through 15 are unused. The first address of a dual board is determined by the physical slot in which it resides (11XXH to 17XXH). The address of the second board is its physical slot plus 8, that is, 1(s+8)XXH (19XXH to 1FXXH). Thus, if a dual board in in Slot 2, its first I/O space is at Address 12XXH, and its second I/O space is at 1AXXH. However, a dual board cannot occupy Slot 8, since Slot 16 is not valid.

Each option board must drive all 16 data lines for the option ID codes. The upper byte (D8-D15) should contain FFH, if the board does not have a start-up PROM. An option board must also drive all 8 lines (byte mode) or 16 lines (word mode) when any of the board's other functions are read. Thus, no data lines should be left floating when a device or register is read.

It is strongly recommended that option boards with 16-bit write registers qualify the decoding of these registers with A0 and /BHE. This gives software greater lattitude in controlling the board's functions.

If software reads nonexistent devices, it should not expect FFH, unless the code can assure sufficient bus idle time before the read is performed. Also, if only a byte exists, a byte read (not a word read) should be performed. These restrictions are needed because, on a system with a heavily loaded 8-slot chassis, it can take up to 500 ns for the system bus to be pulled high by the 4.7k pullup resistors after a previous bus operation.

15.5.6 Other Hardware Functions

On all boards, the processor must be able to write to Port 1XFCh with arbitrary data to cause the option board hardware to experience a software reset identical in all respects to the power-on reset issued by the system at power-on time (refer to Chapter 14, Start and On-board PROMs).

An option board must gate its 7-bit ID code onto Bus Lines D0-D6, its interrupt request bit onto Bus Line D7 and its 8-bit Option Board Class Code onto Bus Lines D8-D15, when the processor reads Port 1XFEh (refer to Section 14.17.3, Class Code Definitions).

For boards capable of being a start-up source or of being the only console output at start-up time, there must be an OBPROM addressable by the processor at Effective Address F4000h (32 KB maximum) after a write to Port 1XFAh with D0 set to 1. Latching and decoding of Bits D8-D1 of Port 1XFAh, to allow a variable mapping location for memory, is optional. If a variable mapping location is not implemented, the board must map to effective address F4000h.

The simplest implementation has one PROM and addresses the data at even addresses. For a word read at an even address, the low byte is valid and the high byte unspecified. A word read at an odd address gives unspecified results. The data can be copied into RAM and utilized. The more complex implementation has two PROMS; code can be addressed by byte or by word.

An option board can have one or more status buffers. A status buffer would be addressed by the processor as an I/O device (I/O-mapped) and would indicate various information related to the status or operational condition of the board. The hardware designer defines the information and configures it into a status buffer.

If the option board can activate the /I/O ERROR bus signal, a bit in a status buffer must be allocated to indicate that this board activated the signal. The architect must assign the board an I/O-mapped address to which the processor can write to shut off the board's /I/O ERROR signal after recognition. Also, a software reset (a write to 1XFCH) should clear the /I/O ERROR signal.

An option board can be designed with one or more I/O-mapped command registers. The processor would write a command (8 or 16 bits) into this register to control various functions on the board.

15.6 HARDWARE AND SIGNAL INTERFACE REQUIREMENTS

Table 15-3 details the drive or loading requirements for an option board to interface to the system bus. In the table, O.C. stands for Open Collector, and T.S. stands for Tri-state.

Table 15-4. Option Board/System Bus Interface

Signal	CPU Input or Output	Option Board Drive Reg. (IOL)	Option Board Load Limit (IIL)	Motherboard Terminator (47 & 470pF Series Combo)
CLK	Out	-	4 mA	Yes
/RESET	Out	-	2 mA	-
/IRQ2-IRQ7	In	8 mA O.C. or T.S.[a]	0	-
/AIOWC	Out	-	2 mA	Yes
/IORC	Out	-	2 mA	Yes
/AMWC	Out	-	2 mA	Yes
/MRDC	Out	-	2 mA	Yes
ALE	Out	-	4 mA	Yes
A0-A15	Out	-	2 mA	-
A16-A19	Out	-	0.8 mA	-
/BHE	Out	-	2 mA	-
/DACK0-DACK3	Out	-	4 mA	-
/DREQ1-DREQ3	In	8mA O.C. or T.S.[a]	0	-
DEN	Out	-	4 mA	Yes
DT//R	Out	-	2 mA	Yes
D0-D15	In/Out	24mA T.S.	2 mA	-
/WTR	Out from Power supply	-	2 mA	-
HACK	Out	-	2 mA	Yes
/I/O ERROR	In	8mA O.C. or T.S.[a]	0	-
T/C	Out	-	2 mA	-
RDY	In	16mA O.C. or T.S.[a]	0	-
/SAD	Out	-	2 mA	-
MCE	Out	-	2 mA	-
ADSTB	Out		4 mA	-
4CLK	Out	-	4 mA	Yes
SID0-SID3	Tied to GND or to single 1K pullup resistor on motherboard	-	400 uA for IIH No limit for IIL	-

[a] If Tri-state devices drive these signals, their outputs must never be permitted to be driven high as other devices may be pulling the same line low at the same time.

When the final artwork of an option board is laid out on a printed circuit board, the maximum stub length of any signal from the option board to the motherboard connector should not exceed 3 inches. If a signal from the motherboard connector is connected to more than one point, all branches of the signal added together should not exceed 5 inches. When a new option board design is initially tested, it must be made to work on an extender card.

When designing option boards, refer to the printed circuit board outline drawing in Appendix A. Each board must have an RF shield at the rear of the unit. Refer to the RF shield drawing in Appendix A.

15.7 System Bus Utilization

This section discusses the types and numbers of devices able to utilize the system bus during the same interval of time. It also discusses the data transfer rates the system bus can support for these various devices.

15.7.1 Processor/DMA Transactions

The system bus has only three possible bus masters: the two coprocessors (8086 and 8087) and a four-channel DMA controller (9517A or 8237A). The coprocessors operate at a basic clock rate of 8 MHz with a 125 ns period, and the DMA controller runs at 4 MHz with a 250 ns period.

The 8086 grants the use of the system bus to the DMA controller and the 8087 coprocessor. The device, DMA or 8087, needing bus mastership issues a request pulse to the 8086. After a request/grant latency time (refer to Table 15-5), the 8086 issues a grant pulse and removes itself from driving the system bus. The new bus master (DMA or 8087) performs its task, then issues a release pulse to the 8086 and removes itself from driving the system bus. The 8086 then reassumes bus mastership.

A bus master must always complete a memory cycle before it relinquishes the bus to the DMA controller. Interfacing the 8086 Request/Grant line (/RQ/GT0) to the DMA bus acquisition logic prevents a DMA request from seizing the bus before the end of a memory cycle.

The DMA controller has higher priority than the 8087. The DMA controller is programmed in Rotating Priority mode for its four channels so that all DMA devices have equal access.

Table 15-5. Request/Grant Latency

Operating Condition	8086 Request/Grant Delay
Normal Instruction Processing-/LOCK inactive	3-6(10[a])clocks
/INTA Cycle Executing-/LOCK active	15 clocks
Locked XCHG Instruction Processing-/LOCK active[b]	24-31(39[a])clocks

[a] The number of clocks in parentheses applies when the instruction being executed references a word operand at an odd address boundary. Each clock is 125 ns for an 8 MHz 8086.

[b] Locked instructions have a request/grant latency equal to the duration of that instruction (the number of clocks times 125 ns per clock).

Channel 0 of the DMA controller provides the refresh address for a /RAS-only dynamic RAM refresh cycle. Every 120 us, Timer Channel 1 initiates a Channel 0 DMA request. In Demand mode, a total of eight DMA Channel 0 transfers are completed to refresh eight row addresses of RAM. Then the DMA controller releases the system bus. This procedure insures that the 128 row addresses of RAM are refreshed each 2 ms. (Two-hundred-and-fifty-six row addresses are refreshed every 4 ms for TMS-4164 RAM chips.) All devices interfacing to the system bus must be designed so that, when they are operating simultaneously with all other possible simultaneous devices, the dynamic RAM refresh sequence will be allowed to occur each 120 us period.

Any analysis of worst-case bus timing must consider the delay in acknowledging a bus request, the time needed to service a DMA request, and the overhead involved in changing bus masters. The maximum delay in acknowledging a bus request (called request/grant latency) depends on the 8086 instruction being executed and can be 3 to 6 clock cycles during normal instruction processing, 10 cycles when a normal instruction accesses a word at an odd memory address, and up to 180 cycles for a DIV instruction with /LOCK active and a nonaligned word operand. In general, a locked instruction has request/grant latency equal to its full execution time, and a nonaligned word operand increases request/grant latency by additional clock cycles. An interrupt acknowledge cycle with /LOCK active adds a further 15 cycles to the request/grant latency.

Table 15-6 lists the various periods that make up the overhead and transfer time associated with DMA transfers. Not including the cycle (1/4 us) for the DMA request, 1/2 us is required to issue the request pulse to the 8086. The time for the request pulse does not affect bus activity. However, it is used in calculations for the time to service a DMA device.

After the request/grant latency period (refer to Table 15-5), another 1/2 us is required to issue the grant pulse from the 8086 and to have the DMA controller ready to drive the system bus. System bus activity cannot occur during this 1/2 us period. However, this period does enter into the system bus bandwidth calculations.

DMA transfers take place on a byte basis and alternate between the lower and upper halves of the data bus. The first byte transferred takes 1 us (4 DMA clock cycles), as does any byte transferred to or from memory at an address that divides evenly by 100H. All other bytes in a multiple-byte transfer (Demand mode) take 3/4 us (three DMA clock cycles), until a 256-byte address boundary is reached. The byte transfer at that boundary will take 1 us, but the following byte transfers will again each take 3/4 us. The 64 KB address boundaries cannot be crossed by the DMA controller without processor intervention to change the DMA page register. If a DMA transfer injects wait states, they extend the time to transfer any byte by 1/4 us per wait state added.

After the DMA controller completes its byte transfers, a 1/4 us period is required to release the system bus back to the 8086. This period also is not useable by the system bus.

Table 15-6. DMA System Bus Acquisition and Utilization Times

Operation	Duration	System Bus Utilization
Issue Request	1/2 us	0 (other devices)
Request/Grant Latency	See Table 1	0 (other devices)
Receive and Process Grant	1/2 us	1/2 us
First DMA Byte Transfer	1 us[a]	1 us
Additional DMA Byte Transfers @	3/4 us[a]	3/4 us (Demand mode)
DMA Transfer at 256 Byte Boundary	1 us[a]	1 us
Release System Bus	1/4 us	1/4 us

[a] Add 1/4 us for each wait state requested by the DMA device.

Apart from the delay in acknowledging a bus request, changing bus masters, performing a DMA transfer, and relinquishing the bus, a certain amount of bus capacity must be reserved for mandatory system functions such as RAM refresh and servicing the floppy disk controller. Out of every 120 us period, refreshing occupies 7.0 us (1/2 + 1 + 7 x 3/4 + 1/4) of the system bus bandwidth and takes 7.5 us to complete from the time of the request.

When the floppy disk drive is operating, it uses single-byte DMA transfers with one wait state. The diskette drive requires 2 us (1/2 + 1 + 1/4 + 1/4) of the system bus bandwidth every 32 us, on average, since it is an unbuffered device. The worst case is 26 us. The diskette controller interface circuitry delays the DMA request of the 765 Floppy Disk Controller (FDC) chip to the DMA controller chip by 1 us before making the request. This prevents the DMA controller from servicing the FDC before the FDC is ready.

The Winchester disk controller also uses single-byte DMA transfers. However, the Winchester controller is a buffered device and, therefore, has no need to prevent erroneous operation by an absolute requirement for the DMA service rate. The Winchester controller can retrieve data at the rate of 1.6 us per byte, but this rate is not supported by single-byte DMA transfer. Therefore, the Winchester controller buffers the data. However, once the sector is located, the Winchester disk controller transfers data to and from disk faster than the DMA controller transfers data across the bus. As a result, additional devices operating simultaneously affect the rate at which Winchester disk data can be transferred from the buffer to the system.

For system level integration of additional devices to the Wang PC system bus, the absolute requirements of the refresh and of the floppy disk controller must be assured. The hardware will support multiple-byte DMA transfers (Demand mode) as well as single-byte transfers. However, the duration of multiple DMA transfers must be carefully calculated and limited to allow required processor functions (parallel port, RS-232-C serial port, video, etc.) to continue, as well as the DMA functions of refresh and floppy disk control. Locked instructions should be avoided (refer to Table 15-5), as well as word operands at odd address boundaries. A locked Divide instruction can inhibit bus grant for 22 1/2 us (180 clocks). Also, nonbuffered devices, such as floppy disks, impose very restrictive operating conditions for the machine.

15.7.2 Examples

This section presents some sample calculations for bus utilization.

Example 1

In the first example, the program being run has one locked XCHG instruction, one /INTA cycle (/LOCK active), and one normal instruction as a group of three consecutive instructions, in any order. The following sequence is for a 26 us interval, the worst-case floppy disk servicing interval. (Although 32 us is the *average* floppy byte transfer interval, 26 us is the worst case -- 129th or 257th bytes of a 512-byte sector.)

Locked XCHG Instruction (word operand at odd address boundary)	4 7/8 us
DMA Channel 0 (Refresh)	7 us
/INTA Cycle (/Lock active)	1 7/8 us
DMA Channel 1	(DMA 1)
Normal Instruction (word operand at odd address boundary)	1 1/4 us
DMA Channel 2 (Service Floppy Disk Controller)	2 us
	17 us + (DMA1)

Since a single-byte DMA transfer with no wait states takes 1 3/4 us, the 26 us worst-case floppy disk servicing interval would not be exceeded. However, if a locked instruction of 97 clocks or longer (locked Divide is 180) can be encountered during floppy operation, no DMA operations may be performed other than Refresh. The requests for bus usage by the DMA controller (1/2 us duration), are issued during non-DMA time (instruction fetch or execution time or bus idle time) and, therefore, are not included in bus utilization calculations.

Example 2

The second example has one INTA cycle (/LOCK active), one DMA service interval (DMA 3), and three consecutive normal instructions, in any order.

/INTA Cycle (/LOCK active)	1 7/8 us
DMA Channel 3	(DMA 3)
Normal Instruction - not locked (word operand at odd address boundary)	1 1/4 us
DMA Channel 0 (Refresh)	7 us
Normal Instruction (6 clocks)	3/4 us
DMA Channel 1	(DMA 1)
Normal Instruction (6 clocks)	3/4 us
DMA Channel 2 (Service Floppy Disk Controller)	2 us
	13 5/8 us + (DMA 3) + (DMA 1)

Since 26 - 13 5/8 = 12 3/8 us, several single-byte transfers can be performed on both Channels 3 and 1. The total bus utilization must be less than the worst case floppy disk service interval of 26 us. The additional normal instruction latency would be 3/4 us. If the system bus is idle at the time of a DMA instruction, the DMA controller still takes 1 1/4 us for DMA latency to issue a new request and to receive and process the bus grant.

Example 3

In the third example, the diskette drive is not operating.

/INTA Cycle (/LOCK active)	1 7/8 us
DMA Channel 1	(DMA 1)
Normal Instruction	3/4 us
DMA Channel 2 (Not floppy disk)	(DMA 2)
Normal Instruction	3/4 us
DMA Channel 3	(DMA 3)
Normal Instruction	3/4 us
DMA Channel 0 (Refresh)	7 us
	11 1/8 us + (DMA 1) + (DMA 2) + (DMA 3)

Since the dynamic RAM refresh interval is 120 us, 108 7/8 us remain to be used by the DMA channels or the processor. If DMA Channel 1 has five wait states, DMA Channel 2 has seven wait states, and both are using single-byte transfers, DMA Channel 1 takes 3 us (1/2 + 1 + 5 x 1/4 + 1/4), and DMA Channel 2 takes 3 1/2 us (1/2 + 1 + 7 x 1/4 + 1/4). Therefore, 102 3/8 us remain for DMA Channel 3.

If DMA Channel 3 uses Demand mode transfers with no wait states, it can transfer 128 consecutive bytes (assuming proper hardware design). That DMA transfer would take 96 3/4 us, or 97 us if a 256-byte address boundary is crossed (96 3/4 = 1/2 + 1 + 127 x 3/4). This leaves 5 3/8 us remaining out of the 120 us refresh interval.

If the three DMA channels in this example were each using single-byte transfers with no wait states, the floppy disk was not operating, and only normal non-locked instructions were encountered, a sequence of a normal instruction and a single-byte transfer would take 2 1/2 us (3/4 + 1 3/4). With the 108 7/8 us remaining from the 120 us refresh interval, a total of 43 of these sequences could be performed. Thus, a total of 43 single-byte transfers could be performed during the 120 us refresh interval.

As the discussion of the previous examples shows, intimate system operating knowledge is required before a calculation involving device transfer rates or system bus bandwidth utilization can be made.

15.8 POWER LIMITS

Tables 15-7 and 15-8 show the maximum power allowed for each I/O option board on the 5-slot and 8-slot chassis, respectively. These power limits include any power delivered through the option board to an external device. The Monochrome Monitor Controller's boards are the exception. They draw 2 amps from the +12v supply through a set of connector contacts on the motherboard connector reserved for that purpose. Only one Monochrome Video Controller board is permitted in the basic system. To install others, the +12 volts for the corresponding monitor must be obtained from an external power supply.

Table 15-7. Maximum I/O Option Board Power

+5 volts	2.5 amps	12.5 watts
-5 volts	20 mA	100 mW
+12 volts	200 mA	2.4 watts
-12 volts	30 mA	50 mW

APPENDIX A
OPTION BOARD AND RF SHIELD MECHANICAL SPECIFICATIONS

This appendix contains the mechanical diagrams for option boards and RF Shields. It also contains specifications for electrodeposited zinc coatings on RF Shields.

A.1 OPTION BOARD MECHANICAL OUTLINE

Figure A-1 shows the option board mechanical outline.

HOLE LEGEND & TOLERANCES

HOLE DIA.	TOLERANCE
0135-125	+003-001
126-250	+004-001
251-500	+005-001

TOL.xx±.010 FRAC.±1/64
xxx±.005 ANG±1°30′ FINISH125

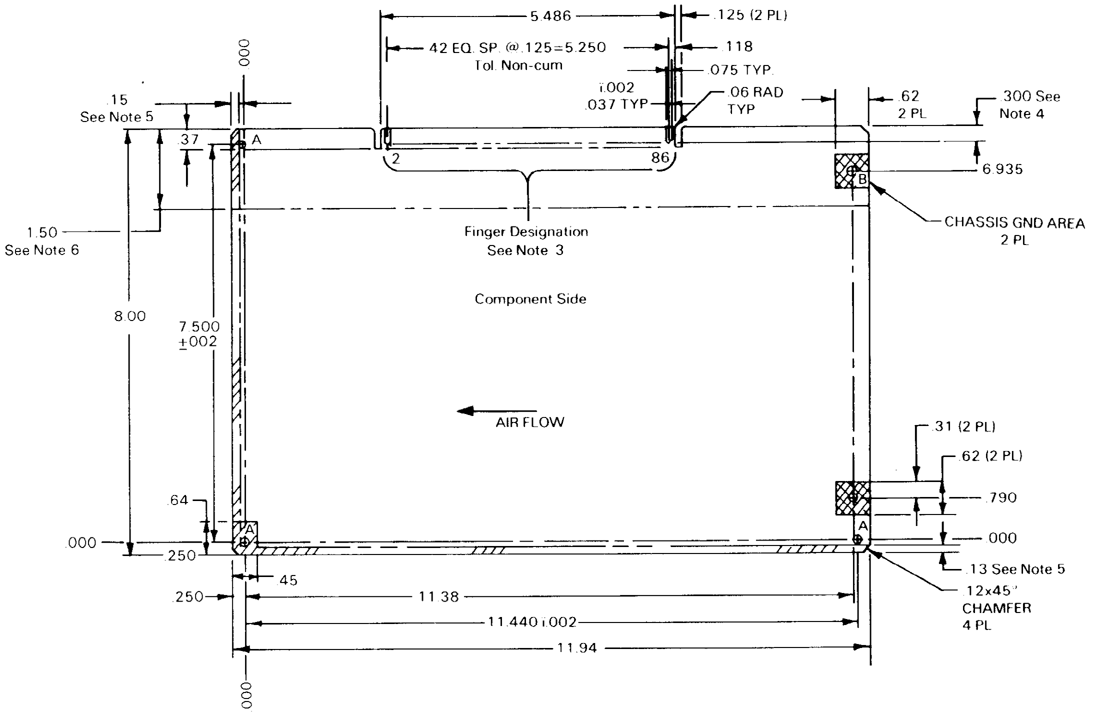


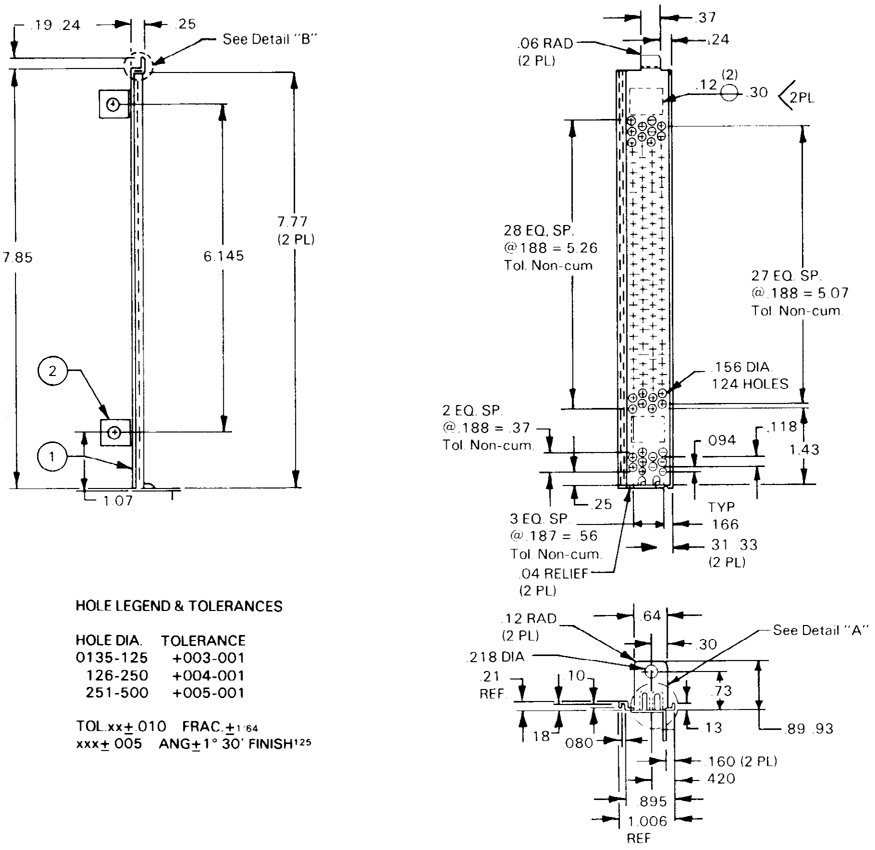
Figure A-1. Option Board Mechanical Outline

Notes for Figure A-1

1. The material is FL-GFN .062 C 1/1 A2A.
2. Holes designated A must be primary drilled and are for automatic insertion tooling.
3. Odd-number designations are for the far side; even-number designations are for the near side. Numbers are not to be silk-screened and are only for the purpose of circuit layout and wiring.
4. Denotes depth of nickel and gold plating on fingers. Nickel or gold plating is required only in finger areas. Both sides must be .0001 nickel with .00003 gold.
5. No circuitry or components in cross-hatched area, both sides, as indicated.
6. Height of components in this area is to be a maximum of .38 inches.
7. Height of components in this area is to be a maximum of .63 inches.
8. The area designated is to be used as chassis ground, component side.

A.2 RF SHIELD SPECIFICATIONS

Figures A-2a and A-2b show the specifications for RF shields.



HOLE LEGEND & TOLERANCES

HOLE DIA.	TOLERANCE
0135-125	+003-001
126-250	+004-001
251-500	+005-001

TOL.xx± .010   FRAC. ±1/64
xxx± 005   ANG±1° 30' FINISH^{125}

Figure A-2a. RF Shield Specifications

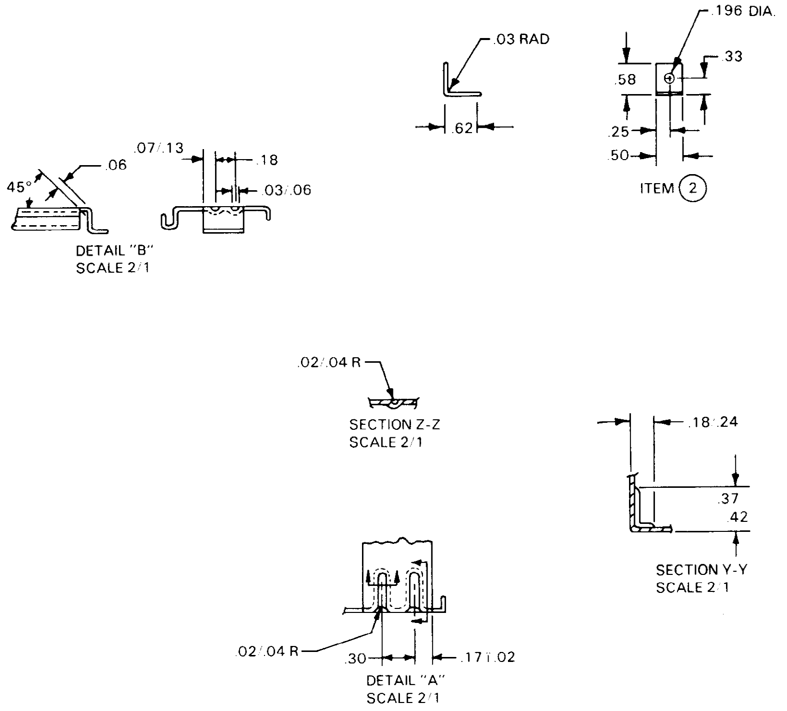


Figure A-2b. RF Shield Specifications (continued)

Notes for Figures A-2a and A-2b

1. Remove burrs and break all sharp edges.

2. The inside bend radii must be a maximum of .015 inches unless otherwise specified.

A.3 ZINC PLATING SPECIFICATIONS

This section covers the requirements for electrodeposited zinc platings. The materials used must produce platings that meet the requirements specified here. Anodes or baths containing mercury must not be used for zinc plating.

A.3.1 General Requirements

The basis metal must be free from visible defects that would be detrimental to the appearance or the protective value of the plating. The basis metal must be subjected to such cleaning, pickling, and plating procedures as are necessary to yield deposits as specified here.

Zinc must be deposited directly on the basis metal without a preliminary plating of another metal.

The plating must not be applied until all machining, welding, forming, and perforating on the article are complete.

The plating must have a maximum surface resistivity of 0.1 ohm/sq.

The plating must be zinc with supplementary chromate treatment. The chromate treatment must be a chemical treatment in an aqueous solution of salts and acids resulting in a continuous smooth, protective film, distinctly colored bright (clear). The articles so treated must be thoroughly rinsed and dried according to part requirements for the particular chemical process used.

The minimum thickness of zinc must be 0.00020 inches (0.005 millimeters) on all visible surfaces that can be touched by a ball 0.75 of an inch in diameter. The thickness of all other visible surfaces must be not less than 0.00015 inch.

Holes and other openings from which the external environment is completely excluded are not subject to a thickness requirement but must show evidence of plating.

A.3.3 Sampling and Testing Requirements

A lot consists of plated articles processed under the same condition, of approximately the same size and shape, and submitted for inspection at one time. A representative sample must be selected from each lot in accordance with MIL-STD-105 Sampling Procedures at Special Inspection Level S-3. The lot must be accepted or rejected in accordance with minimum Acceptable Quality Level (AQL) 2.5 percent.

Each sample selected must be inspected and the plating thickness measured in several places where the plating would be expected to be minimum. If the minimum plating thickness on any article is less than 70 percent of the specified thickness, the lot must be rejected. The minimum thickness measured on each article must be recorded and the average value of the minimum thickness for all the samples computed. If the average minimum thickness is less than 0.00020 inches, the lot must be rejected.

The samples selected must be capable of passing the salt spray test in accordance with the American Society for Testing and Materials (ASTM) Standard B 117. Lots must be accepted or rejected in accordance with AQL 2.5 percent, except that the minimum set tested must be three plated parts or separate specimens. The plating must show neither white corrosion products of zinc nor basis metal corrosion products when test specimens are subjected to 36 hours of continuous exposure to the salt spray. Corrosion products visible to the unaided eye at normal reading distance (approximately 12 inches) shall cause rejection, except for white corrosion products at the edges of the article.

Sample plated pieces shall be selected from each lot in accordance with MIL-STD-105 Sampling Procedures at General Inspection Level 2 for visual examination. The zinc deposit must be smooth, fine grained, adherent, uniform in appearance, and free from blisters, pits, nodules, indications of burning, and other defects. All details and workmanship must conform to the best practice for high-quality plating.

Any piece having one or more defects must be rejected. Lots must be accepted or rejected in accordance with AQL 2.5 percent. The following imperfections will not be cause for rejection:

- Superficial staining that has been demonstrated to result from rinsing
- Small racking marks under 1/8 inch in diameter
- Scratches that are covered by zinc plate finish

APPENDIX B
POWER-ON DIAGNOSTICS

During the power-on diagnostic tests, the keyboard LEDs display the kernel test number as the power-up proceeds through the test. Figure B-1 illustrates the layout of the keyboard LEDs and the bit weight assigned to each one. Numeric values derived from the bit weights are used as test identifiers and error codes (refer to Tables B-2 and B-3). Table B-1 illustrates how to interpret LED codes, with one LED illuminated.

NOTE:
In each table of this appendix, an asterisk indicates the LED is illuminated; a 0 indicates it is extinguished. For ease of reading the asterisk and zero sequences, the tables seperate LED 20 from LED 10.

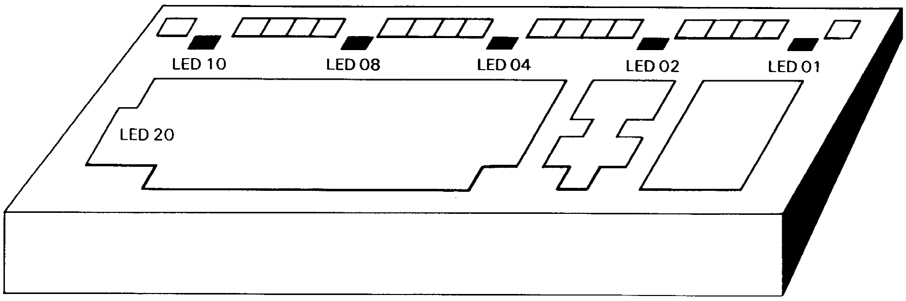


Figure B-1. Keyboard LEDs

Table B-1. Keyboard LED Numeric Values

LEDs	Binary Value	Hex Value
* 00000	10 0000B	20H
0 *0000	01 0000B	10H
0 0*000	00 1000B	08H
0 00*00	00 0100B	04H
0 000*0	00 0010B	02H
0 0000*	00 0001B	01H

If the system card fails a test, the keyboard audio alarm beeps once, and the keyboard LEDs continue to display the number of the test that failed. Table B-2 defines the condition that caused the fatal error.

If Kernel Tests 0DH, 10H, 11H, 12H, 17H or 18H fail, a memory error has been detected. The keyboard LEDs immediately replace the test number with one of the error codes shown in Table B-3.

Table B-2. Power-up Diagnostic Fatal Error Codes

LED CODE	HEX VALUE	ERROR CONDITION
0 000*0	02H	The parity flag (Bit 0) in the System Status register will not clear.
0 00*00	04H	DMA Address Register 0 failed to retain the test pattern 55AAH.
0 00*0*	05H	DMA Address Register 0 failed to retain the test pattern AA55H.
0 00**0	06H	TIMER 0's interrupt flag will not clear.
0 0*000	08H	TIMER 0 failed to generate an interrupt or interrupt not detectable in PIC IRR Bit 0.
0 0*00*	09H	TIMER 0's interrupt at PIC IRR Bit 0 not clearable.
0 0*0*0	0AH	TIMER 2's interrupt flag will not clear.
0 0*0**	0BH	TIMER 2 failed to generate an interrupt or interrupt no detectable in PIC IRR Bit 1.
0 0**00	0CH	TIMER 2's interrupt at PIC IRR Bit 1 not clearable.
0 0**0*	0DH	Refresh test failed between Addresses 0000:0000 and 0000:0200H. Refer to Table B-3 for further information.
0 0***0	0EH	Diagnostic hung during procedure FIRST PASS.
0 *0000	10H	Memory test between Addresses 0000:1000 and 0000:FFFFH (Bank 1). Refer to Table B-3 for further information.
0 *000*	11H	Memory test between Addresses 1000:0000 and 1000:FFFFH (Bank 2). Refer to Table B-3 for further information.
0 *00*0	12H	Memory test between Addresses 0000:0000 and 0000:1000H (Bank 3). Refer to Table B-3 for further information.

Table B-2. Power-up Diagnostic Fatal Error Codes (continued)

LED CODE	HEX VALUE	ERROR CONDITION
0 *00**	13H	Memory parity error detected.
0 *0*00	14H	PIC mask register failed pattern test.
0 *0*0*	15H	Live interrupt failed to occur.
0 *0***	17H	Memory test between Address 2000:0 and end of system card memory. Refer to Table B-3 for further information.
0 **000	18H	Memory parity error during test 17H. Refer to Table B-3 for further information.

The error codes listed in Table B-3 will help locate a possible bad memory chip. If the code is in the range from 20H to 2FH, only one XOR data bit was detected; the related RAM chip is listed in the right-hand column. If the code is in the range 30H to 3FH, more than one XOR bit was detected. The RAM chip listed in the right-hand column then contains the least significant bit found.

Table B-3. Fatal Error Codes

				8221	9521	9521
	HEX		HEX		Lo Bank	Hi Bank
LED code	value	LED code	value	FRU RAM	FRU RAM	FRU RAM
* 00000	20H	* *0000	30H	L130	L41	L43
* 0000*	21H	* *000*	31H	L112	L35	L37
* 000*0	22H	* *00*0	32H	L106	L31	L33
* 000**	23H	* *00**	33H	L101	L25	L27
* 00*00	24H	* *0*00	34H	L87	L19	L21
* 00*0*	25H	* *0*0*	35H	L85	L13	L15
* 00**0	26H	* *0**0	36H	L69	L9	L11
* 00***	27H	* *0***	37H	L57	L3	L5
* 0*000	28H	* **000	38H	L131	L42	L44
* 0*00*	29H	* **00*	39H	L113	L36	L38
* 0*0*0	2AH	* **0*0	3AH	L107	L32	L34
* 0*0**	2BH	* **0**	3BH	L102	L26	L28
* 0**00	2CH	* ***00	3CH	L88	L20	L22
* 0**0*	2DH	* ***0*	3DH	L86	L14	L16
* 0***0	2EH	* ****0	3EH	L70	L10	L12
* 0****	2FH	* *****	3FH	L58	L4	L6
0 *****	1FH	Code for no-XOR-bits-found. Under normal conditions, this code should never occur. If it does occur, replace the power-up PROMs.				

WANG

Customer Comment Form

Publication Number ______ 700-8090A

Title ______ WANG PROFESSIONAL COMPUTER TECHNICAL REFERENCE MANUAL

Help Us Help You . . .

We've worked hard to make this document useful, readable, and technically accurate. Did we succeed? Only you can tell us! Your comments and suggestions will help us improve our technical communications. Please take a few minutes to let us know how you feel.

How did you receive this publication?

□ Support or Sales Rep
□ Wang Supplies Division
□ From another user
□ Enclosed with equipment
□ Don't know
□ Other ______
______
______
______
______
______

How did you use this Publication?

□ Introduction to the subject
□ Classroom text (student)
□ Classroom text (teacher)
□ Self-study text
□ Aid to advanced knowledge
□ Guide to operating instructions
□ As a reference manual
□ Other ______
______

Please rate the quality of this publication in each of the following areas.

	EXCELLENT	GOOD	FAIR	POOR	VERY POOR
Technical Accuracy — Does the system work the way the manual says it does?	□	□	□	□	□
Readability — Is the manual easy to read and understand?	□	□	□	□	□
Clarity — Are the instructions easy to follow?	□	□	□	□	□
Examples — Were they helpful, realistic? Were there enough of them?	□	□	□	□	□
Organization — Was it logical? Was it easy to find what you needed to know?	□	□	□	□	□
Illustrations — Were they clear and useful?	□	□	□	□	□
Physical Attractiveness — What did you think of the printing, binding, etc?	□	□	□	□	□

Were there any terms or concepts that were not defined properly? □ Y □ N If so, what were they? ______

After reading this document do you feel that you will be able to operate the equipment/software? □ Yes □ No
□ Yes, with practice

What errors or faults did you find in the manual? (Please include page numbers) ______
______
______
______

Do you have any other comments or suggestions? ______
______
______
______

Name ______ **Street** ______

Title ______ **City** ______

Dept/Mail Stop ______ **State/Country** ______

Company ______ **Zip Code** ______ **Telephone** ______

Thank you for your help.

WANG

Fold

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 16 LOWELL, MA

POSTAGE WILL BE PAID BY ADDRESSEE

WANG LABORATORIES, INC.
PUBLICATIONS DEVELOPMENT
ONE INDUSTRIAL AVENUE
LOWELL, MASSACHUSETTS 01851

Fold

Cut along dotted line.