

Workstation Hardware



Convergent Technologies

Convergent Technologies
Workstation Hardware Manual

Specifications Subject to Change.

Convergent Technologies, Convergent, CTOS, CT-BUS, IWS, EWS, and MWS are trademarks of Convergent Technologies.

Multibus is a trademark of Intel Corporation.

First Edition Revised (April 1981) A-09-00016-01-B

Copyright © 1981 by Convergent Technologies Inc.

Table of Contents

	<u>Page</u>
Preface.....	xiv
 1. System Architecture	
Overview.....	1-1
The mainframe.....	1-2
Memory.....	1-5
Input/output.....	1-8
Multibus interface.....	1-8
 2. Functions and Interfaces (Processor and I/O-Memory Boards)	
Dual bus organization.....	2-1
Local and multibus separation.....	2-4
Local bus interface.....	2-6
Local I/O bus interface.....	2-6
Multibus interface.....	2-7
Multibus master access.....	2-7
Multibus slave access.....	2-8
Priorities.....	2-8
Bus timeout.....	2-9
The 8086 and 8087 processors.....	2-10
The 8086 CPU.....	2-10
The 8087 numeric data processor.....	2-10
Software interface.....	2-13
8086 CPU.....	2-13
8087 NDP.....	2-16
Memory interface.....	2-20
Address space.....	2-20
ROM memory.....	2-20
RAM memory.....	2-23
Multibus/Local-bus memory mapping.....	2-26
Software interface.....	2-27
Parity error register (PER).....	2-27
I/O and control interfaces.....	2-29
I/O peripheral devices and channels.....	2-29
I/O circuits and registers.....	2-30
DMA Control.....	2-30
Interrupt Control.....	2-30
Keyboard Control and LEDs.....	2-31
Interval Timer.....	2-31
I/O-Memory Board LEDs and Audio.....	2-31
Communications Control.....	2-31
I/O Control Register (IOCR).....	2-32
Parity Error Register (PER).....	2-32
Printer Status Register (PSR).....	2-33

Screen Attribute Register (SAR).....	2-33
Floppy Disk Control (FDC).....	2-33
Winchester Disk Control (WDC).....	2-33
Software interface.....	2-34
DMA Transfers.....	2-35
General operating characteristics.....	2-35
Channel uses.....	2-37
Multibus Access.....	2-38
Communications.....	2-38
Disk	2-40
Video Refresh.....	2-40
Bus exchange.....	2-42
Address generation.....	2-43
Communication and Video Addresses.....	2-43
Disk Addresses.....	2-43
Software interface.....	2-47
8237-2 DMA controller.....	2-47
DMA extended address register (EAR).....	2-53
Interrupts.....	2-55
Non-maskable interrupts.....	2-55
Power Failure.....	2-58
Parity Error.....	2-58
Memory Write-protection.....	2-58
Bus Timeout.....	2-59
Maskable interrupts.....	2-59
Communications (Level 1).....	2-59
Internal Timer (Level 3).....	2-59
Printer Keyboard, realtime clock and 8087 Level 4.....	2-59
Disks (Level 7).....	2-60
Software interface.....	2-61
8259A interrupt controller.....	2-61
I/O control register (IOCR).....	2-70
8253 interval timer.....	2-75
Multibus interface.....	2-80
Signal lines.....	2-80
Form factor.....	2-80
Transfers across the interface.....	2-80
Modes of operation.....	2-84
Bus priority resolution.....	2-84
Interrupts.....	2-85
Communications interface.....	2-88
Channels.....	2-88
Transfer control.....	2-88
Switch-selectable clocks.....	2-91
Extended control register.....	2-91
Software interface.....	2-92
Z-80A serial I/O (SIO) controller.....	2-92
SIO extended control register (ECR).....	2-97
SIO interrupt acknowledge register (IAR).....	2-100
Keyboard interface.....	2-102
Keyboard/mainframe communication functions.....	2-102
Data formats.....	2-102

Software interface.....	2-104
8251A communication interface.....	2-104
Printer interface.....	2-109
Software interface.....	2-111
Utilities.....	2-113
Programmable LEDs and audio.....	2-113
Hardwired LEDs.....	2-113
Power	2-115
Power supply.....	2-115
Autorestart function.....	2-115
Reset.....	2-116

3. Mainframe Motherboard

Multibus priority resolution logic.....	3-1
Cluster communications termination.....	3-2
References.....	3-3
Edge-connector pin list.....	3-3
Jumpers.....	3-3

4. Processor Board

Overview.....	4-1
Operation by logic block.....	4-1
The dual bus architecture.....	4-3
Control of the local bus drivers.....	4-4
Processor ready logic.....	4-6
Bus timeouts.....	4-8
The 8086 microprocessor.....	4-8
The 8087.....	4-11
The RAM controller.....	4-11
Memory read cycles.....	4-12
Memory write.....	4-13
Column address strobe.....	4-13
Parity logic.....	4-14
Parity writing.....	4-14
Parity reading and checking.....	4-14
Interrupt control logic.....	4-15
Direct memory access (DMA).....	4-18
Multibus slave READY generation.....	4-21
Non-Multibus READY generation.....	4-22
Multibus interface.....	4-23
Multibus master mode.....	4-23
8289 Multibus handshake.....	4-23
Acquiring multibus.....	4-24
Multibus master mode operation.....	4-26
Command address and data bus control	
during master mode.....	4-27
LOCK access to Multibus.....	4-28
Slave mode.....	4-29
Multibus slave transfer acknowledge logic.....	4-31
Boot PROMs	4-31
AC characteristics.....	4-33

DC characteristics.....	4-42
References	4-44
Edge-connector pin list.....	4-44
Connector J2.....	4-44
Connector J3.....	4-45
Jumpers.....	4-48
Switches.....	4-48
The memory addressing decode PROM.....	4-48
LED indicators.....	4-49
Test points.....	4-49
I/O-Port addresses vs. functions.....	4-49
Device pin functions.....	4-50
8086 HMOS microprocessor.....	4-50
8284 Clock driver.....	4-57
8289 Bus arbiter.....	4-62
8288 Bus controller.....	4-68
8237-2 DMA controller.....	4-72
8259A Interrupt controller.....	4-79

5. I/O Memory Board

Overview.....	5-1
Operation by logic block.....	5-1
The RAM array.....	5-3
Disk control interface (DCI).....	5-5
I/O address decode.....	5-7
The I/O control register (IOCR).....	5-7
Control bits of the IOCR.....	5-9
Status bits of the IOCR.....	5-10
Speaker LESSs and switches.....	5-11
Programmable timer.....	5-11
Keyboard interface.....	5-12
Line printer interface.....	5-12
Real time clocks.....	5-13
Power fail detect.....	5-13
Write protection.....	5-13
Communications.....	5-14
Non-DMA transfers between 8086 and the SIO.....	5-14
DMA transfers between memory and the SIO.....	5-15
Programming the ECR.....	5-15
Multidrop driver control.....	5-16
Multidrop receiver clock detect.....	5-16
Baud rate clock source selection.....	5-16
Channel B level changers.....	5-18
Channel B level changers.....	5-18
AC characteristics.....	5-19
DC characteristics.....	5-30
References.....	5-32
Edge-connector pin list.....	5-32
Connector J4.....	5-32
Connector J5.....	5-33
Jumpers.....	5-36
Switches.....	5-36

Programmable LED and audio.....	5-36
Test points.....	5-37
I/O Port addresses and functions.....	5-37
Device pin functions.....	5-38
8251A USART.....	5-39
8253 programmable timer.....	5-46
Z80A SIO/2.....	5-49

6. RAM Expansion Board

Overview.....	6-1
Operation by logic block.....	6-1
The RAM array.....	6-1
Buffering for ROM Expansion board.....	6-3
AC characteristics.....	6-5
DC characteristics.....	6-6
References.....	6-7
Edge-connector pin list.....	6-7
Connector J1.....	6-7
Jumper.....	6-8

7. ROM Expansion Board

Overview.....	7-1
Operation by logic block.....	7-1
AC characteristics.....	7-4
DC characteristics.....	7-5
References.....	7-6
Edge-connector pin list.....	7-6
Connector J1.....	7-6
Switches.....	7-7

8. Video Control Board

Functional description and software interfaces.....	8-1
Display characteristics.....	8-1
Character map.....	8-2
Video control logic.....	8-2
DMA transfers.....	8-6
Software interface.....	8-8
Screen attribute register.....	8-8
Character map.....	8-11
Font RAM.....	8-14
Operation by logic block.....	8-16
Bus interface.....	8-16
Row buffers.....	8-19
Attribute logic.....	8-21
Video generator.....	8-22
Main control.....	8-25
Microcode word.....	8-29
Controller operation.....	8-32
Timing chain.....	8-34
AC characteristics.....	8-37

DC characteristics.....	8-38
References.....	8-52
Edge-connector pin list.....	8-52
Jumper.....	8-54
Test points.....	8-54
I/O port address.....	8-54

9. Keyboard

Functional description and software interface.....	9-1
Writing to the keyboard.....	9-1
Operation.....	9-6
DC characteristics.....	9-7
References.....	9-8
Connector Pin list.....	9-8
LED indicators.....	9-8
Test points.....	9-8

10. Video Monitor (CRT)

Functional description.....	10-1
Operation by circuit block.....	10-1
Horizontal deflection.....	10-1
Horizontal regulator.....	10-2
Vertical deflection.....	10-2
Bias supply and dynamic focus.....	10-2
Video amplifier.....	10-4
Power.....	10-4
Adjustment procedures.....	10-5
References.....	10-6
Connector pin list.....	10-6
Potentiometers.....	10-6

11. Console Switches

Function.....	11-1
References.....	11-3
Connectors.....	11-3
LED indicator.....	11-3

12. Power Supply

Functional description.....	12-1
AC characteristics.....	12-3
DC characteristics.....	12-4
References.....	12-5
Connectors.....	12-5
Jumper.....	12-5

Glossary	A-1
Appendix A. IEEE 796 (Multibus) Standard.....	B-1
Appendix B. Environmental Characteristics.....	C-1
Appendix C. Workstation Switch and Jumper Settings.....	D-1

List of Figures

<u>Figure</u>	<u>Title</u>	<u>Page</u>
1-1	Workstation System Overview.....	1-3
1-2	Workstation Mainframe.....	1-4
1-3	Memory Address Space.....	1-7
1-4	Memory-Mapping Between the Local Bus and Multibus... 1-9	
2-1	Mainframe Block Diagram.....	2-3
2-2	8086 Address Generation.....	2-11
2-3	8086 Instruction Set.....	2-14
2-4	8086 Instruction Set (Continued).....	2-15
2-5	8087 Instruction Set.....	2-17
2-6	8087 Instruction Set (Continued).....	2-18
2-7	Memory Capacity and Address Space.....	2-21
2-8	ROM Memory Space.....	2-22
2-9	RAM Configurations (kilobytes).....	2-25
2-10	Parity Error Register (PER).....	2-28
2-11	DMA Block Diagram.....	2-36
2-12	DMA Priorities and Characteristics.....	2-39
2-13	Location of Communication and Video Buffers.....	2-41
2-14	Communication and Video Address Generation.....	2-45
2-15	Disk Address Generation.....	2-46
2-16	8237-2 Register Formats.....	2-52
2-17	DMA Extended Address Register (EAR).....	2-54
2-18	Interrupt Priorities and Functions.....	2-56
2-19	Interrupt Block Diagram.....	2-57
2-20	8259A Initialization Sequence.....	2-65
2-21	8259A Initialization Command Words.....	2-66
2-22	8259A Operation Command Words.....	2-67
2-23	8259A Status Registers.....	2-68
2-24	8259A Instruction Set Summary.....	2-69
2-25	I/O Control Register (IOCR).....	2-72
2-26	8253 Control Word.....	2-79
2-27	Multibus Pin Assignments.....	2-82
2-28	Multibus Form Factor.....	2-83
2-29	Maskable Interrupt Expansion on Multibus.....	2-87
2-30	Communication Channel Characteristics.....	2-89
2-31	RS-232 and RS-422 Pinouts.....	2-90
2-32	SIO Write Registers.....	2-95
2-33	SIO Read Registers.....	2-96
2-34	SIO Extended Control Register (ECR).....	2-99
2-35	SIO Interrupt Acknowledge Register (IAC).....	2-101
2-36	8251A Control/Status.....	2-108
2-37	Printer Connector Pinout.....	2-110
2-38	Printer Status Register (PSR).....	2-112
2-39	Programmable LED's and Audio.....	2-114
3-1	Edge-Connector Pin List.....	3-4
3-2	Edge-Connector Pin List.....	3-5
3-3	Edge-Connector Pin List.....	3-6
3-4	Edge-Connector Pin List.....	3-7
3-5	Edge-Connector Pin List.....	3-8

3-6	Edge-Connector Pin List.....	3-9
3-7	Edge-Connector Pin List.....	3-10
3-8	Edge-Connector Pin List.....	3-11
3-9	Edge-Connector Pin List.....	3-12
3-10	Edge-Connector Pin List.....	3-13
3-11	Edge-Connector Pin List.....	3-14
3-12	Edge-Connector Pin List.....	3-15
3-13	Edge-Connector Pin List.....	3-16
3-14	Edge-Connector Pin List.....	3-17
3-15	Mainframe Motherboard Schematic.....	3-18
3-16	Mainframe Motherboard Schematic.....	3-19
3-17	Mainframe Motherboard Schematic.....	3-20
3-18	Mainframe Motherboard Assembly Drawing.....	3-21
3-19	Mainframe Motherboard Assembly Drawing.....	3-22
4-1	Processor Board Block Diagram.....	4-2
4-2	8086 CPU Chip.....	4-56
4-3	8284 Clock Generator.....	4-61
4-4	8289 Bus Arbiter.....	4-67
4-5	8288 Bus Controller.....	4-71
4-6	8237-2 DMA Controller.....	4-78
4-7	8259A Interrupt Controller.....	4-82
4-8	Processor Board Schematics.....	4-83
4-9	Processor Board Assembly Drawing.....	4-85
5-1	I/O-Memory Board Block Diagram.....	5-2
5-2	I/O Address Decode Signals.....	5-8
5-3	Extended Control Register Bits.....	5-17
5-4	8251A USART.....	5-45
5-5	8253 Programmable Timer.....	5-48
5-6	Z80A SIO/2.....	5-55
5-7	I/O-Memory Board Schematic.....	5-56
5-8	I/O-Memory Board Assembly Drawing.....	5-62
6-1	RAM Expansion Board Schematic.....	6-9
6-2	RAM Expansion Board Assembly Drawing.....	6-11
7-1	ROM Expansion Board Schematic.....	7-8
7-2	ROM Expansion Board Assembly Drawing.....	7-9
8-1	Video Control Board Data Paths.....	8-4
8-2	Screen Attribute Register (SAR).....	8-10
8-3	Character Map Format for 132 char/row.....	8-12
8-4	Character Map Format for 80 char/row.....	8-13
8-5	Font RAM Addressing and Data.....	8-15
8-6	Video Board Block Diagram.....	8-17
8-7	Bus Interface.....	8-18
8-8	Row Buffers.....	8-20
8-9	Font RAM Attribute Logic and Video Output.....	8-24
8-10	Block Diagram of Main Control Section.....	8-26
8-11	Video Control - Simplified Flow Chart.....	8-27
8-12	DMA Request Control Lines.....	8-28
8-13	Microcode Word.....	8-30
8-14	Video Timing Parameters.....	8-36
8-15	Video Timing Flow Chart.....	8-55
8-16	Video Control Flow Chart.....	8-56
8-17	Video Control Flow Chart.....	8-57
8-18	Video Board Schematic.....	8-58

8-19	Video Board Assembly Diagram.....	8-64
9-1	List of Keys.....	9-2
9-2	Map of Keyboard.....	9-3
9-3	Data Format for Keyboard Output.....	9-4
9-4	Keyboard Schematic and Assembly Drawing.....	9-9
9-5	Keyboard Assembly Drawing.....	9-10
10-1	CRT-Display Block Diagram.....	10-3
10-2	Video Monitor Schematic.....	10-8
10-3	Video Monitor Assembly Drawing.....	10-10
11-1	Switch and Fuse Locations.....	11-2
11-2	Console Schematic.....	11-4
11-3	Console Assembly Drawing.....	11-5
12-1	Adjustment Potentiometers.....	12-6
12-2	Power Supply Wiring Diagram.....	12-7

GUIDE TO TECHNICAL DOCUMENTATION

This Manual is one of a series that documents the Convergent™ Family of Information Processing Systems. The series includes:

- o Technical Summary
- o Workstation Hardware Manual
- o Peripherals Hardware Manual
- o Central Processing Unit
- o CTOS™ Operating System Manual
- o Executive Manual
- o Editor Manual
- o BASIC Manual
- o FORTRAN Manual
- o COBOL Manual
- o Pascal Manual
- o Assembly Language Manual
- o Debugger Manual
- o Utilities Manual
- o Data Base Management System Manual
- o 3270 Emulator Manual
- o System Programmer's Guide
- o Operator's Guide

This section outlines the contents of these manuals.

The Technical Summary briefly describes the hardware and software of the Convergent Family of Information Processing Systems. It summarizes the other manuals in one volume. It may be helpful to read this overview before reading the other manuals.

The Workstation Hardware Manual describes the mainframe, keyboard, and video display. It specifies system architecture, printed circuit boards (motherboard, processor, I/O-memory, video control, ROM expansion, and RAM expansion), keyboard, video monitor, Multibus interface, communications interfaces, power supply, and environmental characteristics of the workstation.

The Peripherals Hardware Manual describes the disk subsystems. It specifies the disk controller motherboard, controller boards for the floppy disk and the Winchester disks, power supplies, disk drives, and environmental characteristics.

The Central Processing Unit describes the main processor, the 8086. It specifies the machine architecture, instruction set, and programming at the symbolic instruction level.

The CTOS™ Operating System Manual describes the operating system. It specifies services for managing processes, messages, memory, exchanges, tasks, video, disk, keyboard, printer, timer, communications, and files. In particular, it specifies the standard file access methods.

The Executive Manual describes the command interpreter, the program that first interacts with the user when the system is turned on. It specifies commands for managing files and invoking other programs such as the Editor and the programming languages.

The Editor Manual describes the text editor.

The BASIC, FORTRAN, COBOL, Pascal, and Assembly Language Manuals describe the system's programming languages. Each manual specifies both the language itself and also operating instructions for that language. For Pascal, the manual is supplemented by a popular text, Pascal User Manual and Report.

The Debugger Manual describes the Debugger, which is designed for use at the symbolic instruction level. Together with appropriate interlistings, it can be used for debugging FORTRAN, Pascal, and assembly language programs. (BASIC and COBOL, in contrast, are more conveniently debugged using special facilities described in their respective manuals.)

The Utilities Manual describes miscellaneous programs such as the Linker, which links together separately compiled object files, and the Asynchronous Terminal Emulator.

The Data Base Management System Manual describes the data base management system. It specifies (1) the data definition language, which defines the logical structure of data bases and separately defines their physical organization, (2) the host language interfaces for accessing data bases from each of the system's programming languages, and (3) the utilities for creating, loading, unloading, and reorganizing data bases.

The 3270 Emulator Manual describes the 3270 emulator package.

The System Programmer's Guide addresses the needs of the system programmer or system manager for detailed information on operating system structure and system operation. It describes (1) diagnostics, (2) procedures for customizing the operating system, and (3) system utilities normally used only by a system programmer or manager, for example, Initialize Volume, Backup, and Restore.

The Operator's Guide addresses the needs of the average user for operating instructions. It describes the workstation switches and controls, keyboard function, and floppy disk handling.

PREFACE

INTENDED AUDIENCE

The Workstation Hardware Manual deals with the mainframe (processing, memory and I/O-control portions of the system), the keyboard and the video subsystem for a workstation configuration of the system. The Peripherals Hardware Manual deals with the mass storage (floppy disk and Winchester disk) portions of the system.

The manual is intended for OEM engineers who will test or service the system electronics. It is not, however, intended as documentation of production-level testing nor is it designed to support modifications of the hardware. Convergent Technologies does not support modifications to its boards other than those predetermined by design, such as jumper options; the only provision for user additions to the system are through the Multibus* interface slots on the mainframe and through the conventional serial and parallel I/O connectors on the mainframe.

SOFTWARE CONTENT

The system contains several programmable peripheral ICs and registers. All of these are either initialized to default conditions by the standard operating system during power-up/Reset and/or they are supported with preprogrammed routines in the system library that can be bound to user's application programs. In addition, the operating system's interrupt service processes often reinitialize or modify the operation of this circuitry.

The descriptions given under the "Software Interface" sections of certain chapters document the interface between these software events and the hardware. With this exception, the manual concentrates exclusively on hardware.

CONVENTIONS

All numbering of bits in a word, bits on a bus line and I/O-port addresses is done in hexadecimal notation, with decimal equivalents included in parentheses for reference to CT software manuals. For example, the 20 bits in an address are numbered from 0, the least significant bit, to 13 (19 decimal), most significant bit; the signal lines on 16-bit data buses are numbered from 0 to F (0 to 15, decimal).

Signal names use plus (+) and minus (-) suffixes to distinguish active-high from active-low, respectively. For example,

<u>Signal Name</u>	<u>Logical State</u>	<u>Voltage Level</u>
RD-	0 (active)	Low
	1 (inactive)	High
RD+	0 (inactive)	Low
	1 (active)	High

* Multibus is a trademark of Intel Corporation. Throughout this manual the name is used as a synonym for the IEEE 796 Microcomputer System Bus Standard, which is derived from the Multibus.

The convention used here corresponds to the Multibus convention as follows:

<u>CT Convention</u>	<u>Multibus Convention</u>	<u>IEEE 796 Convention</u>
RD-	RD/	RD*
RD+	RD	RD

In all other respects, our convention is the same as the Multibus convention.

REFERENCES

There is a reference section at the end of most chapters in this manual. These sections contain a variety of lists summarizing hardware-related functions. The appendices at the end of the manual contain additional reference material.

Since the system contains several programmable devices whose hardware functions and software interfaces are only briefly summarized in this manual, readers are

likely to want occasional reference to the following manufacturer's literature:

Intel 8086 Family User's Manual
Intel Component Data Catalog
Zilog Z80A-SIO Technical Manual

Several new terms are introduced in the course of this manual. Generally, they are explained when first used but the Glossary at the end of the manual gives explanations of these terms.

1. SYSTEM ARCHITECTURE

OVERVIEW

The workstation system supports interactive single-job, multi-task data processing. It can be operated in either a stand-alone or multi-drop (distributed) environment.

The configuration has many similarities with larger mainframe computers: substantial main memory is available (up to 1 megabyte), hard disk storage is available either directly to the user or through a shared-disk cluster, and rapid processing and I/O are possible.

The system is dissimilar to larger mainframes, however, in that the system operator is usually the principal user; a high-resolution CRT monitor and a keyboard with programmable functions are integral parts of the operator's interface. Moreover, a convenient and widely-used bus structure (Multibus) is provided in the dual-bus architecture for the addition of user-designed hardware.

Figure 1-1 gives an overview of the system architecture. A high-performance 16-bit 8086 microprocessor forms the heart of the system and ultimately controls all activity. The internal bus (called the CT-BUS[™] or "local bus") for the workstation system supports the following resources:

- o video control and high-resolution CRT video display monitor.
- o 8086 central processor, with optional 8087 coprocessor for hardware floating-point operations.

- o ROM or PROM, up to 144 kilobytes.
- o Dynamic RAM, up to 992 kilobytes.
- o Disk control interface, to external hard and floppy mass storage subsystems.
- o DMA controller, for high-speed I/O in transfers involving video, disk and communications, and Multibus.
- o Interfacing for two serial communications channels, one of which can be controlled by the DMA logic.
- o Keyboard interface and keyboard.
- o Centronics-compatible parallel printer interface.
- o A programmable interval timer.
- o Interrupt control for local and Multibus interrupts to the 8086 CPU.
- o A standard IEEE 796 (Multibus) interface for two or four user-supplied boards.

The local bus, which operates asynchronously with and is electrically separate from the Multibus, has its own high-speed protocol for access by the local resources it supports. The memory is the only local-bus resources accessible from the Multibus.

The workstation configuration includes an operator's keyboard and CRT monitor. Other configurations of the system may not include the keyboard or CRT.

THE MAINFRAME

Figure 1-2 illustrates the mainframe for a workstation configuration. It is a set of hardware components that includes a processor (CPU) board, an I/O-memory board, a motherboard (backplane), a video control board, a console (system switches), a power supply and a fan. The mainframe may optionally control a RAM expansion board, a ROM expansion board, and two or four Multibus

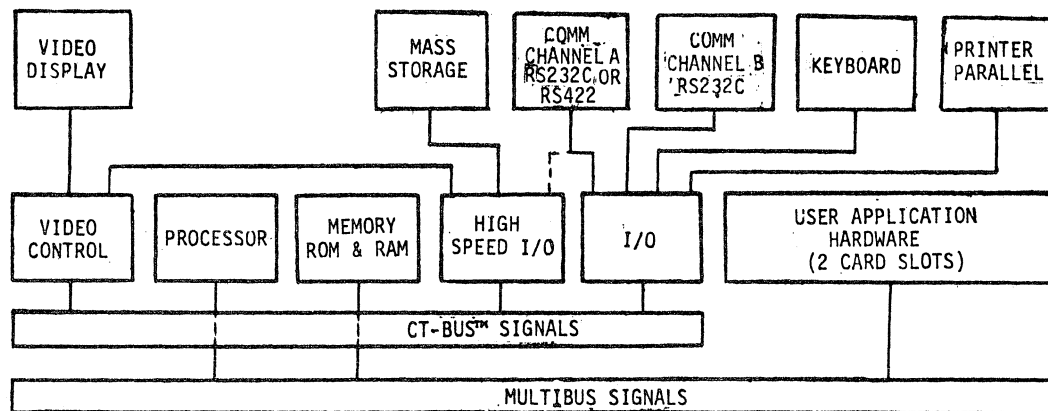
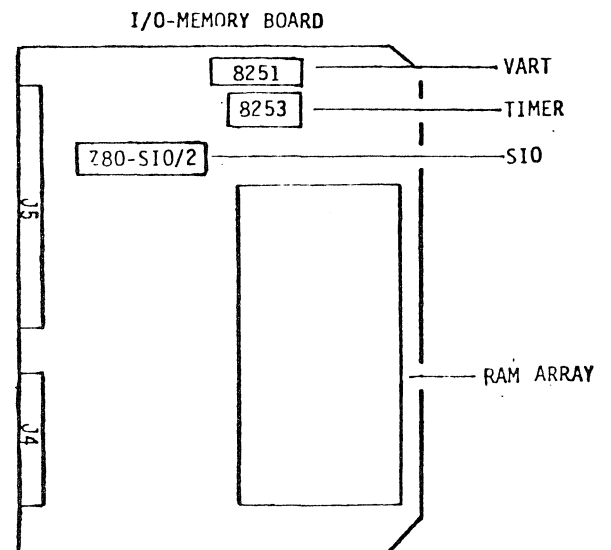
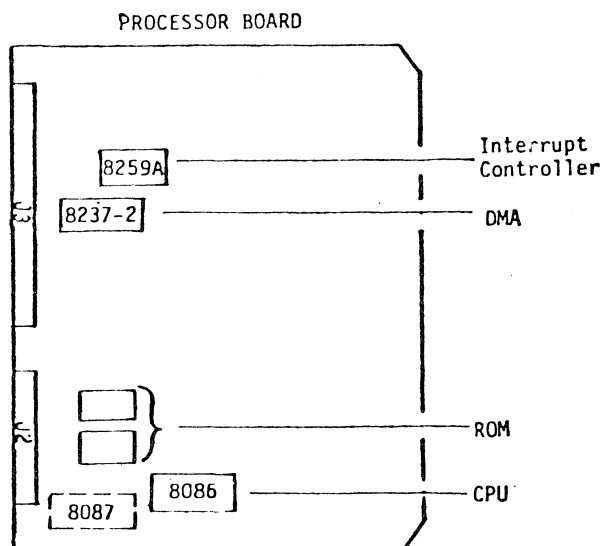
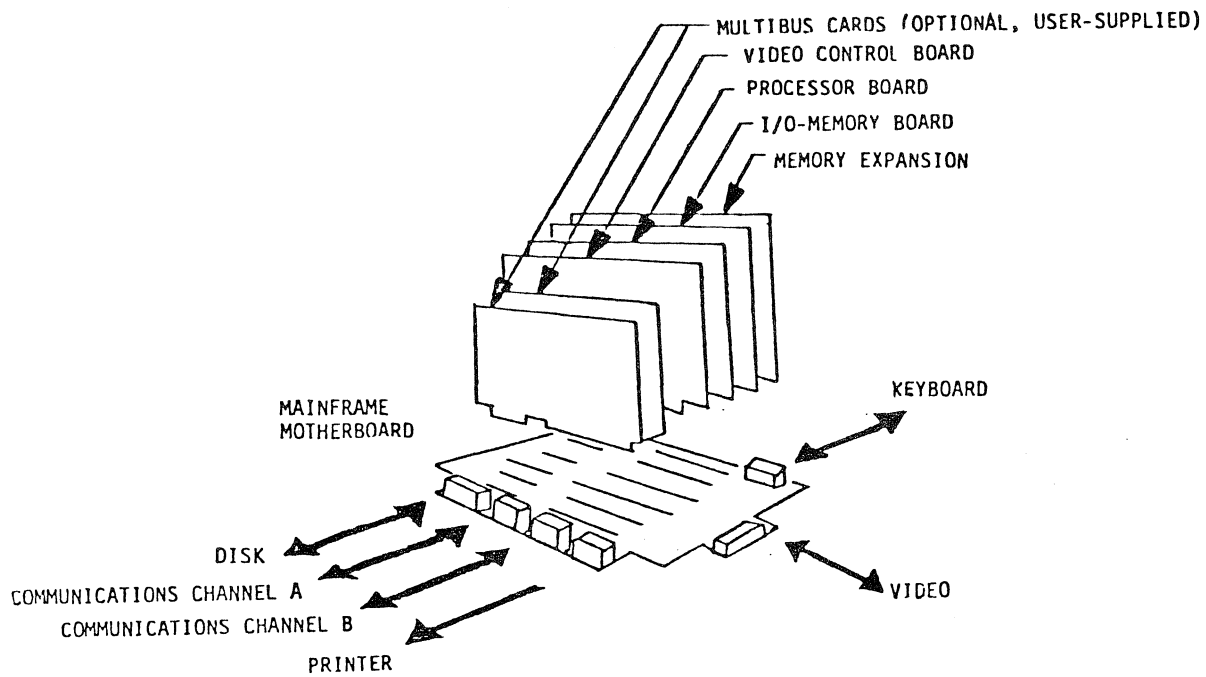


Figure 1-1. Workstation System Overview



Note: Device nomenclature corresponds to Intel device codes with the exception of the SIO which is a Zilog Code. The manufacturer actually used may vary.

Figure 1-2. Workstation Mainframe

boards. If both the RAM and ROM expansion boards are installed, they are stacked onto one mainframe motherboard connector. All boards in the mainframe are contained in a single physical enclosure, although different types of enclosures are used in different mainframe configurations.

The processor board includes the following programmable integrated circuits:

- o 8086 central processing unit (5 MHz, standard; 8 MHz, optional).
- o 8087 numeric data processor (optional) for high-speed arithmetic operations.
- o ROM or PROM, up to 16 kilobytes.
- o A four-channel DMA controller (8237-2 or 9517).
- o An 8259A eight-level interrupt controller for maskable interrupts from the local bus or the Multibus.

The I/O-memory board includes these integrated circuits:

- o Dynamic RAM, up to 512 kilobytes.
- o Z-80 SIO/2 serial I/O controller for dual-channel communication interfacing.
- o 8253 interval timer, for communications baud-rate generation and user application.
- o 8251A UART for asynchronous interfacing to the operator's keyboard.

MEMORY

Figure 1-3 illustrates the system's memory-address space. The maximum 1-megabyte (1024-kilobyte) space is illustrated, although many systems will have only a portion of this space populated with memory chips.

RAM memory occupies the low addresses and can expand to higher addresses in a variety of increments. The lowest space in RAM contains the interrupt table, the operating system, and the I/O buffers.

ROM memory occupies the high addresses and can expand to lower addresses in various increments. The highest space in ROM contains the bootstrap program.

The term "main" ROM or RAM in Figure 1-3 refers to memory chips physically located on one of the two required boards in the mainframe: the processor board for ROM and the I/O-memory board for RAM. The term "expansion" ROM or RAM refers to chips located on one of the optional memory expansion boards.

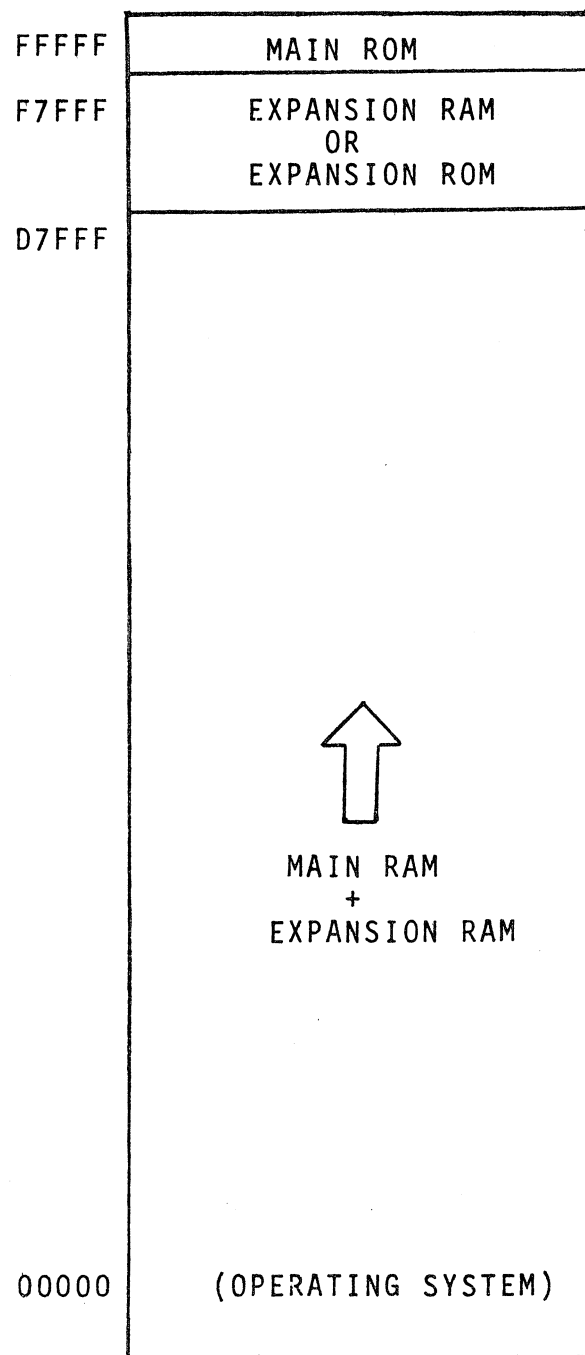


Figure 1-3. Memory Address Space

INPUT/OUTPUT

While the 8086 ultimately controls all activity on the local bus (and, optionally, on the Multibus), much of the I/O associated with the local bus is controlled by the 8237-2 DMA logic.

The following table presents an overview of how data transfers are controlled. The terms "CPU" and "DMA" refer to the 8086 and 8237-2 chips on the local bus.

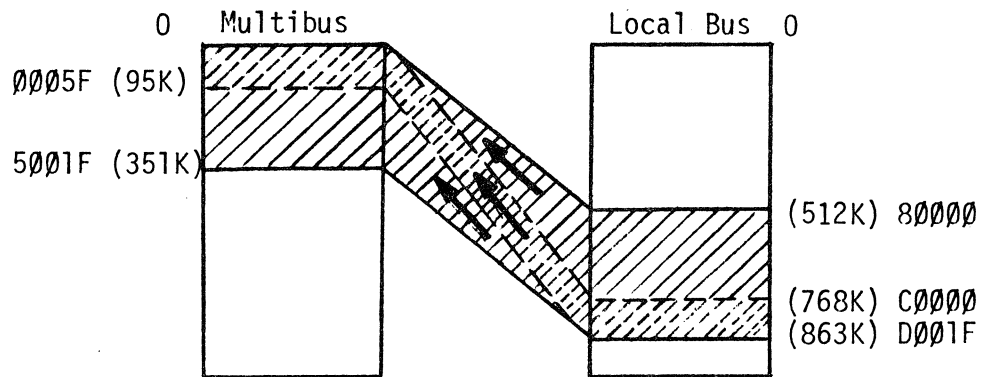
	<u>Data Transfer Method</u>
Video	DMA
Hard Disk	DMA
Floppy Disk	DMA
Communications	DMA or CPU
Keyboard	CPU
Printer	CPU

Of the two communication channels (A and B), only channel A can be configured to use DMA service; this channel can have either an RS-232 or an RS-422 interface.

MULTIBUS INTERFACE

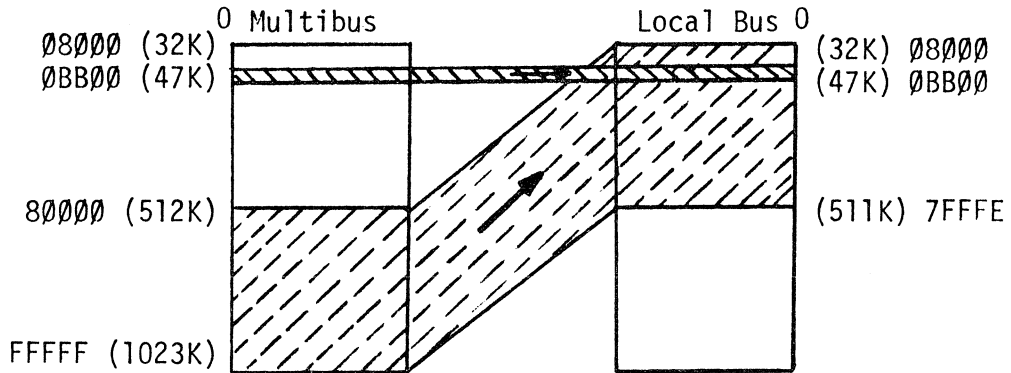
Information in memory can be accessed across the Multibus interface within various memory-address ranges (windows), as depicted in Figure 1-4. The windows are selected by switches on the mainframe boards and they are different, depending on the direction in which the transfer are made: there are three choices for cases in which the local 8086 is controlling the transfer, and four choices for cases in which a Multibus CPU or DMA is controlling the transfer.

The I/O resources supported by the local bus are dedicated, in the sense that they can only be accessed by a local master (i.e., the 8086 CPU or the 8237-2 DMA controller). Multibus devices cannot directly access the local I/O resources; they are limited to accessing local memory. The local 8086 CPU, however, can access Multibus I/O resources.



SWITCH-SELECTABLE CHOICES WHEN LOCAL BUS IS MASTER

- 1 All memory accesses are to local memory.
- 2 768-863K window of local bus maps to 0-95K window on Multibus.
- 3 512-863K window of local bus maps to 0-351K window on Multibus.



SWITCH-SELECTABLE CHOICES WHEN MULTIBUS IS MASTER

- 1 All memory accesses are to Multibus.
- 2 32-47K window of Multibus maps to 32-47K window of local bus.
- 3 512-1023K window of Multibus maps to 0-511K window of local bus.
- 4 Combination of 2 and 3.

Figure 1-4. Memory-Mapping Between the Local Bus and Multibus

2. FUNCTIONS AND INTERFACES (Processor and I/O-Memory Boards)

This chapter describes the functions of the major buses, logic blocks, and hardware interfaces provided on the processor and I/O-memory boards. Software interfaces for the programmable chips and registers on these boards are also described. The video control boards, disk control boards and other components of the system are not covered in this Chapter; their functions and software interfaces are described in other chapters.

DUAL BUS ORGANIZATION

In its most basic configuration, the mainframe for the system consists of a processor board, an I/O-memory board, and a motherboard. The mainframe contains the primary bus structure for the system, a structure which incorporates two sets of buses as illustrated at the top of Figure 2-1:

- o Local Bus: This is the bus that links the entire mainframe. It includes a 20-bit local address bus (LA0 to LA13, hex), a 16-bit local data bus (MEM0 to MEMF, hex) and an 8-bit local I/O bus (IOB0 to IOB7), plus several control lines. The local bus also includes as a subset the processor bus, a collection of 20 time-multiplexed address/data lines (AD0 to AD13, hex) plus several control lines.
- o Multibus: This is the standard IEEE 796 86-pin microcomputer system bus provided for user-supplied boards. It is not used for CT boards, but Multibus board slots are provided on the mainframe motherboard and these slots interface to the local bus.

In Figure 2-1, the mainframe consists of all the blocks except the shaded drawings of peripheral

devices (the CRT, keyboard, printer, disk control and disk); the video control logic is omitted from the mainframe in certain configurations.

Only address and data buses are shown here; control lines are not included in this diagram.

The processor bus subset of the local bus services only the 8086 CPU and the optional 8087 Numeric Data Processor (NDP). Both of these processors have nearly identical pinouts which are tied together by their shared bus. The chips use address/data multiplexing, and it is the function of the local bus interface to demultiplex and latch these lines for convenient use by other devices attached to the local bus and the Multibus.

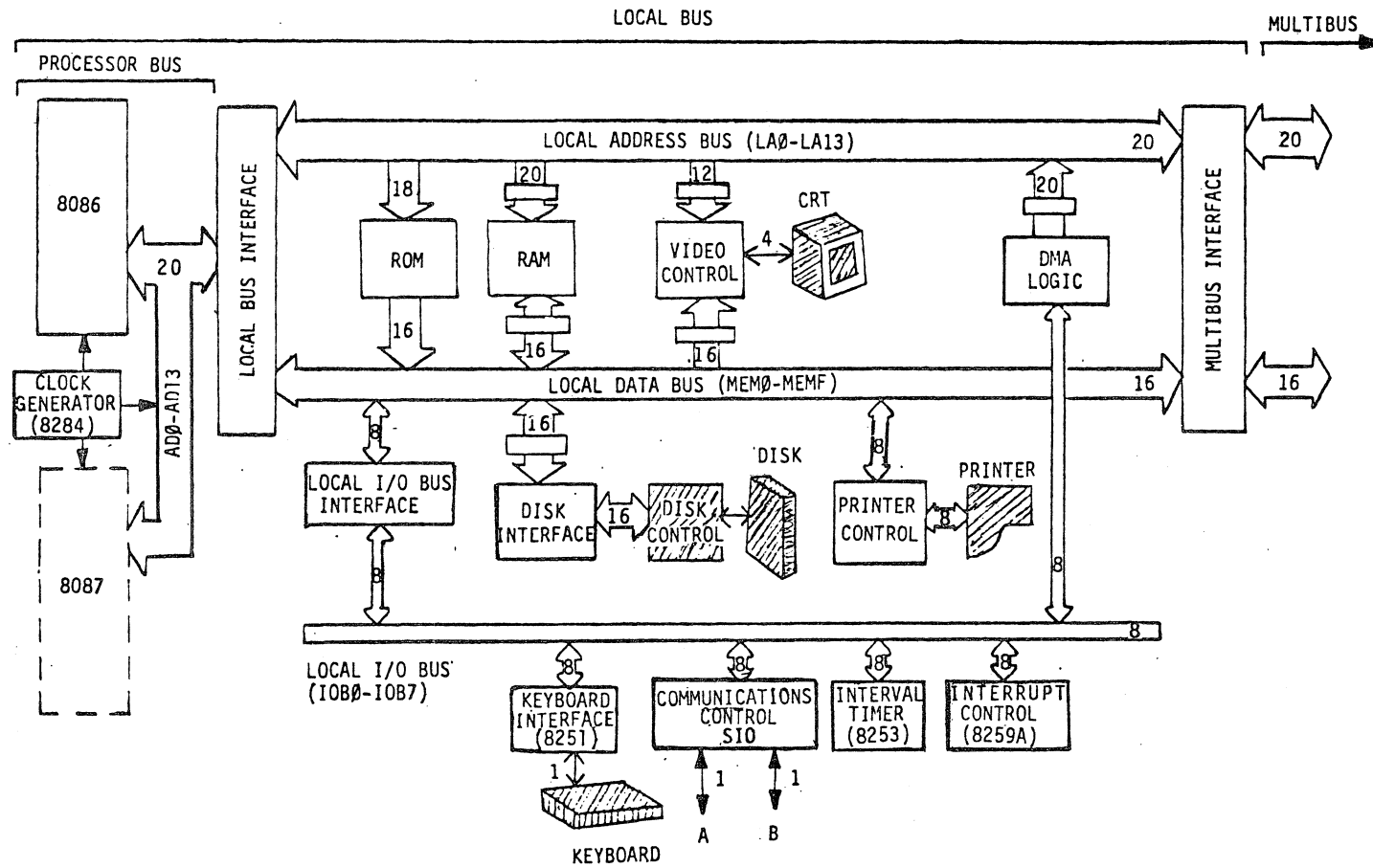
The remainder of the local bus services all of the memory and I/O peripherals provided by CT.

The 20-bit local address bus, LA0-LA13 in Figure 2-1, carries addresses from the local CPU (8086), the local DMA chip (8237-2), or a Multibus master. The addresses reach all of the memory-related resources, including the local ROM, RAM and video control logic. Through the Multibus interface, addresses placed on this bus by the local CPU that are within certain switch-selectable memory ranges (windows) can be directed to the Multibus. Similarly, Multibus masters can address local-bus memory resources within another set of switch-selectable ranges.

The ROM and RAM logic blocks shown in this figure represent both main ROM and RAM (i.e., memory chips installed on the processor or I/O-memory board) as well as expansion memory (i.e., memory chips installed in the ROM or RAM expansion boards). The video control logic is physically contained on its own board which plugs into the mainframe motherboard, and a cable connects the motherboard to the external CRT.

The 16-bit local data bus, MEM0-MEMF in Figure 2-1, connects the demultiplexed processor data bus with the local ROM, RAM, video control, disk control interface, printer control and local I/O bus interface, as well as being buffered through to the Multibus. Note that the disk control

Figure 2-1. Mainframe Block Diagram



interface and the printer interface connect to this bus instead of to the local I/O bus. The disk control board(s), to which the disk control interface logic connects, are located with the disk drives themselves. The printer-interface logic constitutes the entire printer interface (Centronics compatible) provided on the CT system.

The 8-bit local I/O bus, IOB0-IOB7 in Figure 2-1, is buffered off of the local data bus to service the DMA logic, interrupt control, keyboard interface, interval timer and communications interfaces. The keyboard contains its own microprocessor and is attached to the 8251-based keyboard interface logic through a serial cable. The communications control block includes cable connectors on the mainframe enclosure to which standard RS-232 and RS-422 cables can be attached; two separate channels, A and B, are available. The entire local I/O bus is strictly reserved for local resources in the sense that it has no interface with the Multibus and cannot be addressed by a Multibus master. The two serial communications channels on the mainframe provide access to and from external devices.

The processor bus and the rest of the local bus operate in an intimately linked protocol. The local bus and Multibus, by contrast, operate asynchronously.

Word accesses by the CPU may have even or odd addresses on either the local bus or Multibus; the only distinction is that odd-addressed word accesses cause two bus cycles instead of one. For DMA transfers on the local bus, however, only word transfers are made; the LA0 line on the local address bus is always forced to 0. For transfers from the local data bus to the local I/O bus, only the lower byte is seen by the latter bus.

Local and Multibus Separation

The separation of processor bus from local bus is almost a necessity, given the fact that the multiplexed processor lines need to be demultiplexed before they can be used conveniently. The asynchronous separation between local bus and Multibus, however, is not imposed by design restrictions but rather results from a conscious effort to achieve three objectives:

- o Speed: Local bus masters (the 8086, 8087 and 8237-2) can optimize their access to the local bus because they use a bus-arbitration protocol that is more efficient for their well-defined environment than the bus-request/bus-acknowledge protocol used on Multibus.
- o Protection: By controlling the access of Multibus signals to the local bus (a task handled by the local DMA logic) the local bus can protect itself and its memory resources from unauthorized or electrically anomalous interaction with the Multibus.
- o Parallel Processing: Because the local bus is electrically separate from the Multibus, connected only through buffers that can be controlled from the local bus, both buses can operate simultaneously but separately. That is, the local CPU and a Multibus CPU can operate simultaneously as long as they do not require access to resources on the opposite side of the Multibus interface.

The local bus appears as simply another Multibus module to Multibus masters trying to gain access to local resources. In order to gain access to those portions of local memory that are open to them through the switch-selectable memory-space windows, Multibus masters must go through their standard protocol of raising their bus-request (BREQ/) line and awaiting a bus-priority-in (BPRN/) signal through the priority-resolving network.

Local bus masters, on the other hand, have faster and more direct priority-resolving methods for access to their own bus. On one hand, both the 8086 and 8087 watch the same instruction stream being fetched from memory and they execute a mutually exclusive set of instructions; moreover, they have a request/grant handshake that is synchronized by their shared clock. On the other hand, the DMA logic watches status lines from the CPU and is able to anticipate when the bus will become free before the CPU has finished executing an instruction. This results in faster access to local memory than is possible for Multibus masters.

Either the DMA chip or a Multibus master can use the local bus between CPU bus cycles, during non-bus CPU cycles or when the CPU is waiting for permission to use Multibus slave resources.

Local Bus Interface

The logic block in Figure 2-1 that represents the interface between the processor bus and the remainder of the local bus (local bus interface) includes several functions beyond the demultiplexing and latching of the processor's address/data bus. It also includes the bus control logic (organized around an 8288 bus controller chip), the processor address decode logic, and the local I/O address decode logic.

The 8288, which handles local bus control by watching the CPU's status lines, also has an interface to the Multibus through a transceiver. This arrangement puts read/write commands on the Multibus when the local CPU is the Multibus master.

The processor address decode logic determines what logic block (on the processor board, I/O-memory board, or user-installed Multibus boards) a CPU instruction refers to. It thereby determines whether the local bus should connect to the Multibus. This logic also connects to the local I/O address decode logic, to the 8086, the optional 8087 and the 8237-2 DMA chip.

The buffers on the local bus interface are bipolar (TTL-type) in order to achieve high speed and drive. Similarly, the buffers that interface the local address (LA) and local data (MEM) buses to the many devices attached to these buses use bipolar buffering for the same reason. This is in contrast to the devices on the local I/O bus, which have only internal MOS buffering and are much slower.

Local I/O Bus Interface

The local I/O bus interface consists entirely of a TTL transceiver connecting the local data (MEM) bus and the local I/O bus. The devices attached to the local I/O bus are all 8-bit MOS devices with no additional buffering beyond their internal pin buffering, and their drive capacity is limited to the devices on the local I/O bus with

which they interact. Because these devices inherently operate more slowly than those on the local data bus, they require three additional clock cycles (Wait states) beyond the read/write-cycle specifications given in the 8086 data sheet.

Multibus Interface

As mentioned above, the Multibus interface provides standard Multibus slots for user-installed boards. A later chapter defines the standard interface by reference to the related IEEE document.

The Multibus is interfaced electrically to the local bus through bipolar transceivers which are controlled by the local DMA logic. In addition to this buffering, the Multibus interface contains the Multibus control logic (rebuffered from the 8288 in the local bus interface), the Multibus address-decode logic for determining which memory and I/O addresses are meant for it, and the arbitration logic (primarily and 8289) for determining when the local masters can get the Multibus.

The local bus can generate a 10 MHz bus clock for the Multibus or it can receive clocking from the Multibus; this option is switch-selectable.

Interprocessor communication between the local bus and the Multibus is supported by the standard LOCK prefix for 8086 instructions, which can be used to implement test-and-set semaphores or other synchronization primitives in software.

Interaction between the local bus and the Multibus can take place in both directions; i.e., it can occur with the local CPU acting as master (called a "Multibus master access") or with a Multibus board acting as master (called a "Multibus slave access").

Multibus Master Access

When the local CPU is master of the Multibus, hardware switches on the processor board determine which memory-address ranges, if any, refer to memory resources located on the Multibus rather than the local bus. A 96-kilobyte or 352-kilobyte range within the total 1-Megabyte ad-

dress space can be selected for these accesses. For local CPU access to Multibus I/O devices, whenever an I/O address does not pertain to a local bus device, Multibus I/O devices will be accessed.

Multibus Slave Access

Hardware switches on the processor board and software-programmable bits in the local DMA logic control whether a Multibus board can become master of the local bus and, if so, to what areas of memory. The range of local address space accessible to a Multibus master can be set at 16 kilobytes or 512 kilobytes. Only memory accesses (including the video buffer in memory) are permitted.

Memory accesses from a Multibus master may be either byte or word aligned. As in conventional 8086 usage, if word accesses have odd addresses they will take two bus cycles instead of one.

Priorities

Priorities for access to the local bus, from highest to lowest, are as follows:

1. Multibus-master accesses to local memory, in which the local DMA logic arbitrates access but the Multibus master generates addresses.
2. Channel A communications access to and from local memory, in which the local DMA chip generates addresses. (Channel B communications are not serviced by the DMA logic and have a priority of 5, since it uses the CPU for transfer control).
3. Disk accesses (floppy or hard disk) to and from local memory, in which the local DMA chip generates addresses.
4. Video-refresh accesses from local memory, in which the local DMA chip generates addresses.
5. 8086/8087 use of the local bus, in which the 8086 or 8087 generates addresses.

When local bus usage involves memory there are two levels of contention involved: the first level of contention is for the bus itself, as described above. The second level is for the memory. At this second level, memory refresh always has priority over bus masters.

Access by the 8086 and 8087 is given equal priority. These two devices compete with each other for the local bus with a request/grant handshake that is synchronized to their shared clock.

Bus Timeout

The system causes a 6-millisecond bus timeout if a memory location or I/O port does not respond when it is addressed. The timeout will affect accesses of local-bus masters to local-bus slaves, local-bus masters to Multibus slaves, and Multibus masters to local-bus slaves. Timeouts between Multibus masters must be provided independently of the local timeout.

The enabling/disabling of the timeout function interrupt is done with a bit in the I/O Control Register (IOCR); a non-maskable interrupt is caused whenever the memory or I/O device does not respond within the allotted time. The mechanism is designed to protect the local bus's functions against faulty or non-existent slave devices, unplugged cables, faulty programs, or other unforeseen problems.

THE 8086 AND 8087 PROCESSORS

The 8086 CPU

The CPU for the system is an Intel 8086. A 5-MHz version is standard but an 8-MHz version is offered as an option. The chip is operated in its Maximum Mode.

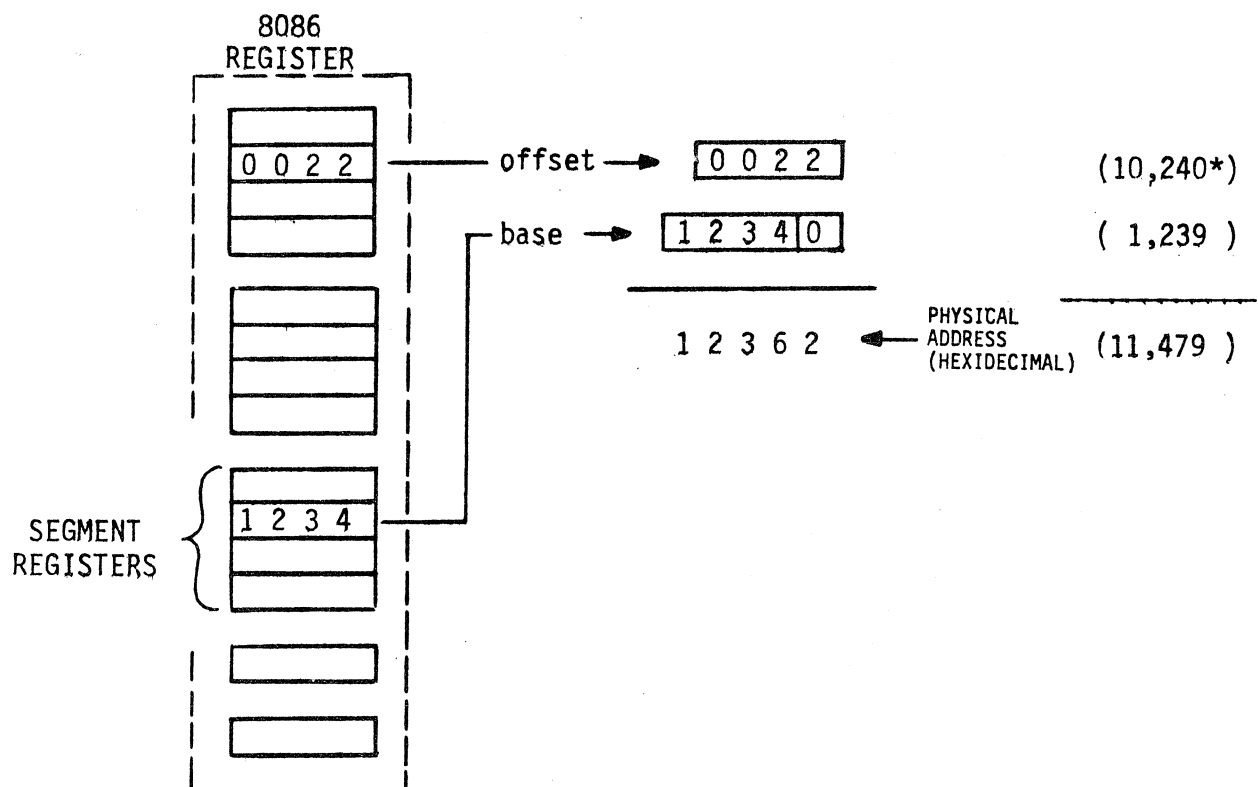
The 8086 achieves high performance in part through pipelined architecture. This architecture combines a 16-bit internal data path with prefetch registers which permit instructions to be fetched from memory and queued for execution during bus cycles that would otherwise go unused.

The 16-bit registers include four registers used to store and update base addresses for memory. Figure 2-2 illustrates the manner in which the contents of two 16-bit registers are manipulated to obtain 20-bit addresses, allowing a 1-megabyte address space. The segment register, one of the four registers referred to above, holds the base address for a given logical segment of memory. The offset for the address is obtained from one of the 8086's general-purpose registers or from memory. Whenever memory is accessed, a physical address is obtained from a logical address by shifting the segment base four bits to the left and adding the offset.

For more background on the 8086, see the Intel 8086 Family User's Manual.

The 8087 Numeric Data Processor

The optional 8087 numeric data processor (NDP) is a coprocessor which can improve system throughput dramatically in some applications. Coprocessing is a special case of multiprocessing, one in which both processors execute from a single instruction stream.



* Effective when shifted right 4, $640 \times 16 = 10,240$.

Figure 2-2. 8086 Address Generation

In the 8086-8087 combination, both processors operate in unison. They monitor the same instruction stream and execute selected instructions from it. For example, while the 8086 deals with memory segmentation, calculating the addresses of operands in memory, the 8087 can perform complex arithmetic and logic operations that would otherwise take the 8086 much longer to compute.

The 8087 can filter out its own instructions as they appear on the bus that it shares with the 8086. The 8086 calculates the memory address for the initial operand, if any, and puts that on the bus. The 8087 latches this address, reads the operand, and begins to execute the required numerical operation, leaving the 8086 free to process non-numeric commands. The 8087 takes control of the bus only when necessary, to load and store operands.

The 8087 can be viewed as an architectural extension of the 8086. It in effect extends the register resources and instruction set of the CPU. Programming the 8086-8087 combination is essentially the same as programming a lone 8086. From the programmer's standpoint the 8087 simply provides 68 new instructions and data types (seven in all) on top of those provided by the 8086.

Software Interface

8086 CPU

The 8086 uses the full instruction set, which is summarized in Figures 2-3 and 2-4.

To fully understand the programming of the 8086, the following publications are recommended:

- o Convergent Technologies Central Processing Unit
- o Convergent Technologies Assembler Manual

Descriptions of the 8086 memory and I/O interfaces are given later in this chapter. The memory address space accessible to the 8086 depends not only on the amount and location of physical memory chips installed in the system but also upon the jumper-selected windows between local-bus and Multibus address spaces.

8086 addressing of I/O ports can be affected by the setting of bit C in the I/O Control Register (IOCR), which is described under the "Interrupts" section of this chapter.

DATA TRANSFER

MOV - Move

Register/memory to/from register	1000100w	mod reg r/m		
Immediate to register/memory	1100011w	mod 000 r/m	data	data if w=1
Immediate to register	1011w	reg	data	data if w=1
Memory to accumulator	1010000w	addr low	addr high	
Accumulator to memory	1010001w	addr low	addr high	
Register/memory to segment register	10001110	mod 0 reg r/m		
Segment register to register/memory	10001100	mod 0 reg r/m		

PUSH - Push

Register/memory	11111111	mod 110 r/m		
Register	01010	reg		
Segment register	000	reg 110		

POP - Pop

Register/memory	10001111	mod 000 r/m		
Register	01011	reg		
Segment register	000	reg 111		

XCMB - Exchange

Register/memory with register	1000011w	mod reg r/m		
Register with accumulator	10010	reg		

IN - Input from

Fixed port	1110010w	port		
Variable port	1110110w			

OUT - Output to

Fixed port	1110011w	port		
Variable port	1110111w			

XLAT - Translate byte to AL

LEA - Load EA to register	11010111			
---------------------------	----------	--	--	--

LDS - Load pointer to DS	10001101	mod reg r/m		
--------------------------	----------	-------------	--	--

LES - Load pointer to ES	11000100	mod reg r/m		
--------------------------	----------	-------------	--	--

LAHF - Load AH with flags	10011111			
---------------------------	----------	--	--	--

SAHF - Store AH into flags	10011110			
----------------------------	----------	--	--	--

PUSHF - Push flags	10011100			
--------------------	----------	--	--	--

POPF - Pop flags	10011101			
------------------	----------	--	--	--

ARITHMETIC

ADD - Add

Reg/memory with register to either	0000000w	mod reg r/m		
Immediate to register/memory	1000000w	mod 000 r/m	data	data if s=w-01
Immediate to accumulator	0000010w	data	data if w=1	

ADC - Add with carry

Reg/memory with register to either	0001000w	mod reg r/m		
Immediate to register/memory	1000000w	mod 010 r/m	data	data if s=w-01
Immediate to accumulator	0001010w	data	data if w=1	

INC - Increment

Register/memory	1111111w	mod 000 r/m		
Register	01000	reg		
AAA - ASCII adjust for add	00110111			
DAA - Decimal adjust for add	00100111			

SUB - Subtract

Reg/memory and register to either	0010100w	mod reg r/m		
Immediate from register/memory	1000000w	mod 101 r/m	data	data if s=w-01
Immediate from accumulator	0010110w	data	data if w=1	

SBB - Subtract with borrow

Reg/memory and register to either	0001100w	mod reg r/m		
Immediate from register/memory	1000000w	mod 011 r/m	data	data if s=w-01
Immediate from accumulator	0001110w	data	data if w=1	

DEC - Decrement

Register/memory	1111111w	mod 001 r/m		
Register	01001	reg		
DEC - Change sign	1110111w	mod 011 r/m		

CMP - Compare

Register/memory and register	0011100w	mod reg r/m		
Immediate with register/memory	1000000w	mod 111 r/m	data	data if s=w-01
Immediate with accumulator	0011110w	data	data if w=1	
AAS - ASCII adjust for subtract	00111111			
DAS - Decimal adjust for subtract	00101111			
MUL - Multiply (unsigned)	1111011w	mod 100 r/m		
IMUL - Integer multiply (signed)	1111011w	mod 101 r/m		
AAM - ASCII adjust for multiply	11010100	00001010		
DIV - Divide (unsigned)	1111011w	mod 110 r/m		
IDIV - Integer divide (signed)	1111011w	mod 111 r/m		
AAD - ASCII adjust for divide	11010101	00001010		
CWD - Convert byte to word	10011000			
CQW - Convert word to double word	10011001			

LOGIC

NOT - Invert	1111011w	mod 010 r/m		
SHL/SAL - Shift logical/arithmetic left	1101000w	mod 100 r/m		
SHR - Shift logical right	1101000w	mod 101 r/m		
SAR - Shift arithmetic right	1101000w	mod 111 r/m		
ROL - Rotate left	1101000w	mod 000 r/m		
ROR - Rotate right	1101000w	mod 001 r/m		
RCL - Rotate through carry flag left	1101000w	mod 010 r/m		
RCR - Rotate through carry flag right	1101000w	mod 011 r/m		

AND - And

Reg/memory and register to either	0010000w	mod reg r/m		
Immediate to register/memory	1000000w	mod 100 r/m	data	data if w=1
Immediate to accumulator	0010010w	data	data if w=1	

TEST - And function to flags, no result

Register/memory and register	1000010w	mod reg r/m		
Immediate data and register/memory	1111011w	mod 000 r/m	data	data if w=1
Immediate data and accumulator	1010100w	data	data if w=1	

OR - Or

Reg/memory and register to either	0000100w	mod reg r/m		
Immediate to register/memory	1000000w	mod 001 r/m	data	data if w=1
Immediate to accumulator	0000110w	data	data if w=1	

XOR - Exclusive or

Reg/memory and register to either	0011000w	mod reg r/m		
Immediate to register/memory	1000000w	mod 110 r/m	data	data if w=1
Immediate to accumulator	0011010w	data	data if w=1	

STRING MANIPULATION

REP - Repeat	1111001z			
MOVSB - Move byte/word	1010010w			
CMPSB - Compare byte/word	1010011w			
SCASB - Scan byte/word	1010111w			
LODSB - Load byte/word to AL/AX	1010110w			
STOSB - Store byte/word from AL/AX	1010101w			

Figure 2-3. 8086 Instruction Set

CONTROL TRANSFER

CALL Call

	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
Direct within segment	1 1 1 0 1 0 0 0	disp low	disp high
Indirect within segment	1 1 1 1 1 1 1 1	mod 0 0 r/m	
Direct intersegment	1 0 0 1 1 0 1 0	offset low	offset high
		seg low	seg high
Indirect intersegment	1 1 1 1 1 1 1 1	mod 0 1 1 r/m	

JMP Unconditional Jump

Direct within segment	1 1 1 0 1 0 0 1	disp low	disp high
Direct within segment-short	1 1 1 0 1 0 1 1	disp	
Indirect within segment	1 1 1 1 1 1 1 1	mod 0 0 r/m	
Direct intersegment	1 1 1 0 1 0 1 0	offset low	offset high
		seg low	seg high
Indirect intersegment	1 1 1 1 1 1 1 1	mod 0 1 1 r/m	

RET Return from CALL

Within segment	1 1 0 0 0 0 1 1		
Within seg adding immmed to SP	1 1 0 0 0 1 0 1	data low	data high
Intersegment	1 1 0 0 1 0 1 1		
Intersegment adding immediate to SP	1 1 0 0 1 0 1 0	data low	data high
JZ/JNZ Jump on equal zero	0 1 1 1 0 1 0 0	disp	
JL/JGE Jump on less-not greater or equal	0 1 1 1 1 1 0 0	disp	
JLE/JBG Jump on less or equal-not greater	0 1 1 1 1 1 1 0	disp	
JG/JAE Jump on below-not above or equal	0 1 1 1 0 0 1 0	disp	
JBE/JNA Jump on below or equal-not above	0 1 1 1 0 1 1 0	disp	
JP/JPE Jump on parity-parity even	0 1 1 1 1 0 1 0	disp	
JO Jump on overflow	0 1 1 1 0 0 0 0	disp	
JS Jump on sign	0 1 1 1 0 0 0 1	disp	
JNE/JNZ Jump on not equal-not zero	0 1 1 1 0 1 0 1	disp	
JNL/JGE Jump on not less-greater or equal	0 1 1 1 1 1 0 1	disp	
JBLE/JB Jump on not less or equal-greater	0 1 1 1 1 1 1 1	disp	

JNB/JAE Jump on not below-above or equal
 JNBE/JA Jump on not below or equal-above
 JNP/JPB Jump on not parity-par odd
 JNO Jump on not overflow
 JNS Jump on not sign
 LOOP Loop CX times
 LOOPZ/LOOPE Loop while zero-equal
 LOOPNZ/LOOPNE Loop while not zero-equal
 JCXZ Jump on CX zero

	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
JNB/JAE	0 1 1 1 0 0 1 1	disp
JNBE/JA	0 1 1 1 0 1 1 1	disp
JNP/JPB	0 1 1 1 1 0 1 1	disp
JNO	0 1 1 1 0 0 0 1	disp
JNS	0 1 1 1 1 0 0 1	disp
LOOP	1 1 1 0 0 0 1 0	disp
LOOPZ/LOOPE	1 1 1 0 0 0 0 1	disp
LOOPNZ/LOOPNE	1 1 1 0 0 0 0 0	disp
JCXZ	1 1 1 0 0 0 1 1	disp

INT Interrupt

Type specified
 Type 3
 INT0 Interrupt on overflow
 IRET Interrupt return

	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
INT	1 1 0 0 1 1 0 1	type
Type 3	1 1 0 0 1 1 0 0	
INT0	1 1 0 0 1 1 1 0	
IRET	1 1 0 0 1 1 1 1	

PROCESSOR CONTROL

CLC Clear carry
 CMC Complement carry
 STC Set carry
 CLD Clear direction
 STD Set direction
 CLI Clear interrupt
 STI Set interrupt
 HLT Halt
 WAIT Wait
 ESC Escape to external device
 LOCK Bus lock prefix

	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
CLC	1 1 1 1 1 0 0 0	
CMC	1 1 1 1 0 1 0 0	
STC	1 1 1 1 1 0 0 1	
CLD	1 1 1 1 1 1 0 0	
STD	1 1 1 1 1 1 0 1	
CLI	1 1 1 1 1 0 1 0	
STI	1 1 1 1 1 0 1 1	
HLT	1 1 1 1 1 1 1 0	
WAIT	1 1 1 1 1 1 1 1	
ESC	1 1 0 1 1 1 1 1	mod 0 1 1 r/m
LOCK	1 1 1 1 1 1 1 1	

Footnotes:

AL - 8-bit accumulator
 AX - 16-bit accumulator
 CX - Count register
 DS - Data segment
 ES - Extra segment
 Above/below refers to unsigned value
 Greater - more positive
 Less - less positive (more negative) signed values
 if s = 1 then "16" reg; if s = 0 then "16-bit" reg
 if w = 1 then word instruction; if w = 0 then byte instruction

if s w = 01 then 16 bits of immediate data form the operand
 if s w = 11 then an immediate data byte is sign extended to form the 16-bit operand
 if v = 0 then "count" = 1; if v = 1 then "count" in (CL)
 x = don't care
 z is used for string primitives for comparison with ZF FLAG

SEGMENT OVERRIDE PREFIX

0 0 1 reg 1 1 0

if mod = 11 then r/m is treated as a REG field
 if mod = 00 then DISP = 0; disp-low and disp-high are absent
 if mod = 01 then DISP = disp-low sign-extended to 16-bits disp-high is absent
 if mod = 10 then DISP = disp-high disp-low
 if r/m = 000 then EA = (BX) - (SI) - DISP
 if r/m = 001 then EA = (BX) - (DI) - DISP
 if r/m = 010 then EA = (BP) - (SI) - DISP
 if r/m = 011 then EA = (BP) - (DI) - DISP
 if r/m = 100 then EA = (SI) - DISP
 if r/m = 101 then EA = (DI) - DISP
 if r/m = 110 then EA = (BP) - DISP
 if r/m = 111 then EA = (BX) - DISP
 DISP follows 2nd byte of instruction (before data if required)

*except if mod = 00 and r/m = 110 then EA = disp-high disp-low

REG is assigned according to the following table

16-Bit (w 1)	8-Bit (w 0)	Segment
000 AX	000 AL	00 ES
001 CX	001 CL	01 CS
010 DX	010 DL	10 SS
011 BX	011 BL	11 DS
100 SP	100 AH	
101 BP	101 CH	
110 SI	110 DH	
111 DI	111 BH	

Instructions which reference the flag register file as a 16 bit object use the symbol FLAGS to represent the file

FLAGS X X X X (OF) (DF) (IF) (TF) (SF) (ZF) X (AF) X (PF) X (CF)

Figure 2-4. 8086 Instruction Set (Continued)

8087 NDP

The optional 8087 performs arithmetic and logical operations on a variety of numeric data types. Users with this option should refer to the Intel literature. A summary of the instruction set is given in Figures 2-5 and 2-6.

7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0

Integer/Real Memory to ST(0)

ESCAPE	MF 1	MOD 0 0 0 R/M	(DISP-LO)	(DISP-HI)
--------	------	---------------	-----------	-----------

Long Integer Memory to ST(0)

ESCAPE	1	1	1	MOD	1	0	1	R/M	(DISP-LO)	(DISP-HI)
--------	---	---	---	-----	---	---	---	-----	-----------	-----------

Temporary Real Memory to ST(0)

ESCAPE	0	1	1	MOD	1	0	1	R/M	(DISP-LO)	(DISP-HI)
--------	---	---	---	-----	---	---	---	-----	-----------	-----------

BCD Memory to ST(0)

ESCAPE	1	1	1	MOD	1	0	0	R/M	(D/SP-LO)	(D/SP-HI)
--------	---	---	---	-----	---	---	---	-----	-----------	-----------

ST(i) to ST(0)

ESCAPE	0	0	1	1	1	0	0	0	ST(i)
--------	---	---	---	---	---	---	---	---	-------

FST = STORE

ST(0) to Integer/Real Memory

ESCAPE	MF	1	MOD	0	1	0	R/M	(DISP-LO)	(DISP-HI)
--------	----	---	-----	---	---	---	-----	-----------	-----------

ST(0) to ST(i)

ESCAPE	1	0	1	1	1	0	1	0	ST(i)
--------	---	---	---	---	---	---	---	---	-------

FSTP = STORE AND POP

ST(0) to Integer/Real Memory

ESCAPE	MF	1	MOD	0	1	1	R/M	(DISP-LO)	(DISP-HI)
--------	----	---	-----	---	---	---	-----	-----------	-----------

ST(0) to Long Integer Memory

ESCAPE	1	1	1	MOD	1	1	1	R/M	(DISP-LO)	(DISP-HI)
--------	---	---	---	-----	---	---	---	-----	-----------	-----------

ST(0) to Temporary Real Memory

ESCAPE	0	1	1	MOD	1	1	1	R/M	(DISP-LO)	(DISP-HI)
--------	---	---	---	-----	---	---	---	-----	-----------	-----------

ST(0) to BCD Memory

ESCAPE	1	1	1	MOD	1	1	0	R/M	(DISP-LO)	(DISP-HI)
--------	---	---	---	-----	---	---	---	-----	-----------	-----------

ST(0) to ST(i)

ESCAPE	1	0	1	1	1	0	1	1	ST(0)
--------	---	---	---	---	---	---	---	---	-------

FXCH = Exchange ST(i) and ST(0)

ESCAPE	0	0	1	1	1	0	0	1	ST(i)
--------	---	---	---	---	---	---	---	---	-------

Comparison

FCOM = Compare

Integer/Real Memory to ST(0)

ESCAPE	MF	0	MOD	0	1	0	R/M	(DISP-LO)	(DISP-HI)
--------	----	---	-----	---	---	---	-----	-----------	-----------

ST(i) to ST(0)

ESCAPE	0	0	0	1	1	0	1	0	ST(i)
--------	---	---	---	---	---	---	---	---	-------

FCOMP = Compare and Pop

Integer/Real Memory to ST(0)

ESCAPE	MF	0	MOD	0	1	1	R/M	(DISP-LO)	(DISP-HI)
--------	----	---	-----	---	---	---	-----	-----------	-----------

ST(i) to ST(0)

ESCAPE	0	0	0	1	1	0	1	1	ST(0)
--------	---	---	---	---	---	---	---	---	-------

**FCOMPP = Compare ST(1) to ST(0)
and Pop Twice**

ESCAPE	1	1	0	1	1	0	1	1	0	0	1
--------	---	---	---	---	---	---	---	---	---	---	---

FTST = Test ST(0)

ESCAPE	0	0	1	1	1	1	0	0	1	0	0
--------	---	---	---	---	---	---	---	---	---	---	---

FXAM = Examine ST(0)

ESCAPE	0	0	1	1	1	1	0	0	1	0	1
--------	---	---	---	---	---	---	---	---	---	---	---

2-17

Arithmetic		7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0
FADD = Addition		
Integer/Real Memory with ST(0)	ESCAPE MF 0 MOD 0 0 0 R/M (DISP-LO) (DISP-HI)	
ST(i) and ST(0)	ESCAPE d P 0 1 1 0 0 0 ST(i)	
FSUB = Subtraction		
Integer/Real Memory with ST(0)	ESCAPE MF 0 MOD 1 0 R R/M (DISP-LO) (DISP-HI)	
ST(i) and ST(0)	ESCAPE d P 0 1 1 1 0 R R/M	
FMUL = Multiplication		
Integer/Real Memory with ST(0)	ESCAPE MF 0 MOD 0 0 1 R/M (DISP-LO) (DISP-HI)	
ST(i) and ST(0)	ESCAPE d P 0 1 1 0 0 1 R/M	
FDIV = Division		
Integer/Real Memory with ST(0)	ESCAPE MF 0 MOD 1 1 R R/M (DISP-LO) (DISP-HI)	
ST(i) and ST(0)	ESCAPE d P 0 1 1 1 1 R R/M	
FSQRT = Square Root of ST(0)		ESCAPE 0 0 1 1 1 1 1 1 0 1 0
FSCALE = Scale ST(0) by ST(1)		ESCAPE 0 0 1 1 1 1 1 1 1 0 1
FPREM = Partial Remainder of ST(0) + ST(1)		ESCAPE 0 0 1 1 1 1 1 1 0 0 0
FRNDINT = Round ST(0) to Integer		ESCAPE 0 0 1 1 1 1 1 1 1 0 0
FXTRACT = Extract Components of ST(0)		ESCAPE 0 0 1 1 1 1 1 0 1 0 0
FABS = Absolute Value of ST(0)		ESCAPE 0 0 1 1 1 1 0 0 0 0 1
FCNS = Change Sign of ST(0)		ESCAPE 0 0 1 1 1 1 0 0 0 0 0
Transcendental		
FPTAN = Partial Tangent of ST(0)		ESCAPE 0 0 1 1 1 1 1 0 0 1 0
FPATAN = Partial Arc tangent of ST(0) + ST(1)		ESCAPE 0 0 1 1 1 1 1 0 0 1 1
F2XM1 = $2^{ST(0)} - 1$		ESCAPE 0 0 1 1 1 1 1 0 0 0 0
FYL2X = $ST(1) \cdot \log_2 [ST(0)]$		ESCAPE 0 0 1 1 1 1 1 0 0 0 1
FYL2XP1 = $ST(1) \cdot \log_2 [ST(0) + 1]$		ESCAPE 0 0 1 1 1 1 1 1 0 0 1
Constants		
FLDZ = LOAD 0.0 into ST(0)		ESCAPE 0 0 1 1 1 1 0 1 1 1 0
FLD1 = LOAD 1.0 into ST(0)		ESCAPE 0 0 1 1 1 1 0 1 0 0 0
FLOPI = LOAD π into ST(0)		ESCAPE 0 0 1 1 1 1 0 1 0 1 1
FLDL2T = LOAD $\log_2 10$ into ST(0)		ESCAPE 0 0 1 1 1 1 0 1 0 0 1
FLDL2E = LOAD $\log_2 e$ into ST(0)		ESCAPE 0 0 1 1 1 1 0 1 0 1 0
FLDLG2 = LOAD $\log_{10} 2$ into ST(0)		ESCAPE 0 0 1 1 1 1 0 1 1 0 0
FLDLN2 = LOAD $\log_e 2$ into ST(0)		ESCAPE 0 0 1 1 1 1 0 1 1 0 1

Figure 2-6. 8087 Instruction Set (Continued)

	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Processor Control																																
FINIT = Initialize NDP	ESCAPE 0 1 1 1 1 1 0 0 0 0 1 1																															
FENI = Enable Interrupts	ESCAPE 0 1 1 1 1 1 0 0 0 0 0 0																															
FDISI = Disable Interrupts	ESCAPE 0 1 1 1 1 1 0 0 0 0 0 1																															
FLDCW = Load Control Word	ESCAPE 0 0 1 MOD 1 0 1 R/M (DISP-LO) (DISP-HI)																															
FSTCW = Store Control Word	ESCAPE 0 0 1 MOD 1 1 1 R/M (DISP-LO) (DISP-HI)																															
FSTSW = Store Status Word	ESCAPE 1 0 1 MOD 1 1 1 R/M (DISP-LO) (DISP-HI)																															
FCLEX = Clear Exceptions	ESCAPE 0 1 1 1 1 1 0 0 0 1 0																															
FSTENV = Store Environment	ESCAPE 0 0 1 MOD 1 1 0 R/M (DISP-LO) (DISP-HI)																															
FLDENV = Load Environment	ESCAPE 0 0 1 MOD 1 0 0 R/M (DISP-LO) (DISP-HI)																															
FSAVE = Save State	ESCAPE 1 0 1 MOD 1 1 0 R/M (DISP-LO) (DISP-HI)																															
FRSTOR = Restore State	ESCAPE 1 0 1 MOD 1 0 0 R/M (DISP-LO) (DISP-HI)																															
FINCSTP = Increment Stack Pointer	ESCAPE 0 0 1 1 1 1 1 0 1 1 1																															
FDECSTP = Decrement Stack Pointer	ESCAPE 0 0 1 1 1 1 1 0 1 1 0																															
FFREE = Free ST(i)	ESCAPE 1 0 1 1 1 0 0 0 ST(i)																															
FNOP = No Operation	ESCAPE 0 0 1 1 1 0 1 0 0 0 0																															
FWAIT = CPU Wait for NDP	1 0 0 1 1 0 1 1																															

FOOTNOTES:

if mod = 00 then DISP = 0*, disp-low and disp-high are absent
 if mod = 01 then DISP = disp-low sign-extended to 16-bits, disp-high is absent
 if mod = 10 then DISP = disp-high; disp-low
 if mod = 11 then r/m is treated as an ST(i) field

if r/m = 000 then EA = (BX) + (SI) + DISP
 if r/m = 001 then EA = (BX) + (DI) + DISP
 if r/m = 010 then EA = (BP) + (SI) + DISP
 if r/m = 011 then EA = (BP) + (DI) + DISP
 if r/m = 100 then EA = (SI) + DISP
 if r/m = 101 then EA = (DI) + DISP
 if r/m = 110 then EA = (BP) + DISP*
 if r/m = 111 then EA = (BX) + DISP

*except if mod = 000 and r/m = 110 then EA = disp-high, disp-low.

MF = Memory Format
 00 — 32-bit Real
 01 — 32-bit Integer
 10 — 64-bit Real
 11 — 16-bit Integer

ST(0) = Current stack top
 ST(i) = ith register below stack top

d = Destination
 0 — Destination is ST(0)
 1 — Destination is ST(i)

P = Pop
 0 — No pop
 1 — Pop ST(0)

R = Reverse
 0 — Destination (op) Source
 1 — Source (op) Destination

For **FSQRT**: $-0 \leq ST(0) \leq +\infty$
 For **FSCALE**: $-2^{15} \leq ST(1) < +2^{15}$ and ST(1) integer
 For **F2XM1**: $0 \leq ST(0) \leq 2^{-1}$
 For **FYL2X**: $0 < ST(0) < \infty$
 $-\infty < ST(1) < +\infty$
 For **FYL2XP1**: $0 < |ST(0)| < (2 - \sqrt{2})/2$
 $-\infty < ST(1) < \infty$
 For **FPTAN**: $0 \leq ST(0) < \pi/4$
 For **FPATAN**: $0 \leq ST(0) < ST(1) < +\infty$

Figure 2-6a. 8087 Instruction Set (Continued)

MEMORY INTERFACE

Address Space

The 1-Megabyte address space, illustrated in Figure 1-3 of the prior chapter, can be populated with memory chips in a variety of ways. ROM occupies the highest addresses and expands downward; RAM occupies the lowest addresses and expands upward.

Main ROM and RAM are both of variable size, depending on the type and number of chips installed. Similarly, expansion ROM and RAM can be of variable size. However, the border line between main and expansion ROM is fixed at hexadecimal address F7FFF (1,015,807 decimal), whereas the border line between main and expansion RAM can be at several alternative address. This allows flexibility in the configuration of main versus expansion RAM without sacrificing the contiguity of the RAM address space.

Figure 2-7 shows the maximum capacities and their associated maximum address-space allocations for ROM and RAM. The two options for expansion memory represent no expansion ROM (option #1) and maximum expansion ROM (option #2).

ROM Memory

The term ROM, as used in this manual, applies to either ROM or PROM. Figure 2-8 gives a detailed illustration of the memory space that can be occupied by ROM.

ROM capacity can be added in increments of 4 kilobytes, up to a total of 16 kilobytes for main ROM (located on the processor board) and a total of 128 kilobytes for expansion ROM (located on the ROM expansion board, although this space can alternatively be occupied by RAM located on the RAM expansion board). Total ROM capacity for the entire system is therefore 144 kilobytes, although the reserved address space (for future use) is 160 kilobytes.

The devices that can be loaded into the available sockets are as follows:

	<u>ROM</u>	<u>RAM</u>
<u>Main Memory:</u>		
Maximum capacity (bytes)	16K	512K
Maximum space allocated (bytes)	32K	512K
<u>Main + Expansion Memory, Option #1</u>		
Maximum capacity (bytes)	16K	992K**
Maximum space allocated (bytes)	32K	992K
<u>Main + Expansion Memory, Option #2</u>		
Maximum capacity (bytes)	160K	864K***
Maximum space allocated (bytes)	160K	864K

* ROM on processor board and RAM on I/O-memory board.

** 1024 minus 32K (for main ROM)

*** 1024 minus 32K (for main ROM) minus 128K (for expansion ROM)

Figure 2-7. Memory Capacity and Address Space

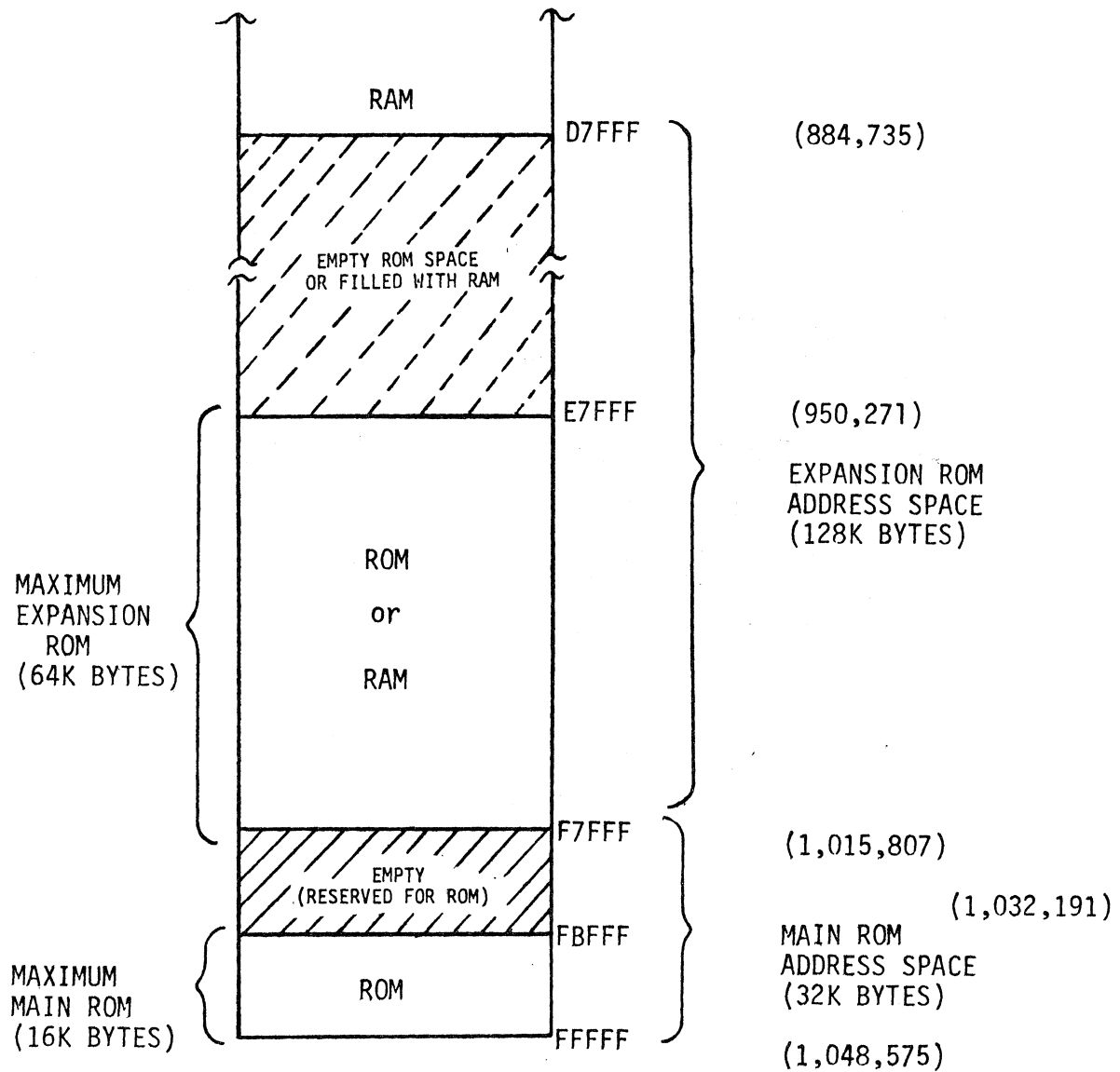


Figure 2-8. ROM Memory Space

<u>Chip Types</u>	<u>Chip Capacity</u>	<u>Board Capacity</u>
Main (processor board):		
2316 ROM or 2716 PROM	2K x 8	4K bytes
2332 ROM or 2732 PROM	4K x 8	8K bytes
2364 ROM or 2764 PROM	8K x 8	16K bytes
Expansion Board:		
2316 ROM or 2716 PROM	2K x 8	32K bytes
2332 ROM or 2732 PROM	4K x 8	64K bytes
2364 ROM or 2764 PROM	8K x 8	128K bytes

All chips on the same board (processor board or expansion board) must be of the same type. The main ROM (processor board) consists of two chips, or one word row. The ROM expansion board consists of eight word rows, two chips per row, so that expansion ROM can be added in one-row increments. Only one ROM and one RAM expansion board can be used in a mainframe.

Access time for main and expansion ROM is 500 nanoseconds. With a 5-MHz CPU clock, 3 Wait states will normally be added to the bus cycle. With an 8-MHz clock, 4 Wait states are normally added.

The bootstrap program that occupies the highest locations of ROM does a basic verification of hardware functions before reading in the operating system. The read is first attempted from floppy drive 0, then from hard-disk drive 0.

RAM Memory

RAM memory is organized in the standard 8086 format of high-byte bank and low-byte bank, thereby allowing either byte or word accesses. Chips are added from the bottom of the address space upward. By 8086 convention, the interrupt pointers occupy the lowest one kilobyte of RAM memory. The operating system is normally loaded above the interrupt pointers.

A total of 512 kilobytes of RAM can be installed on the I/O-memory board. The fully-buffered RAM expansion board can also hold 512 kilobytes, although the total of the two RAM boards cannot exceed 992 kilobytes. The highest 128 kilobytes of permitted RAM space overlaps with the space allocated to the ROM expansion board (see Figure 2-8), hence this space can be used for either RAM

or ROM, but not intermixed. If it is used for RAM, the entire 128-kilobyte space can be used.

RAM can be added in increments of 32 or 128 kilobytes, depending on whether 16K or 64K chips are used. These increments represent word rows. There are a total of 8 word rows possible, 4 on the I/O-memory board and 4 on the RAM expansion board. The types of chips (16K or 64K) can be different on each board. Figure 2-9 shows the possible configuration of chips. Each configuration is associated with a specific RAM-decode PROM and switch setting on the processor board, as described in chapter 4.

RAM is refreshed in a way that does not interfere with non-RAM bus accesses, although it does take precedence over RAM accesses from bus masters. The 8 bits of refresh address are adequate to support virtually all types of 16K and 64K dynamic RAM's. The refresh rate of 78 kHz represents only about 5% of the memory bandwidth, during which RAM is unavailable to the CPU.

A flexible, jumper-controlled scheme of delay-line timing provides support for a wide variety of RAM chips. No Wait states are required when running a 5-MHz clock or when faster RAM's are used with a 8-MHz clock. Timing protection is provided for the T_{ras} and T_{rp} timing specifications of the RAS signal, thereby preventing destructive results from illegal accesses by Multi-bus masters.

A special memory-read optimizer circuit on the processor board generates memory-read commands faster than the 8288 bus controller is able to, thereby allowing the CPU to operate at full speed even with relatively slow RAM chips. The optimizer circuit works in parallel with the 8288, using the status outputs of the 8086.

Single-bit RAM errors can be detected and reported. Even parity is used by default and odd parity can be software selected; the parity logic itself is checked by switching between the two modes for the same data. If a parity error is detected, its address and circumstances (word/byte access, high/low-byte error, last cause of system reset) are stored in the Parity-Error Register (PER) on the processor board.

<u>I/O-Memory Board</u>	<u>Expansion RAM Board</u>	<u>Total</u>
128K	0K	128K
128	32	160
128	64	192
128	96	224
128	128	256
128	256 (64K chips)	384
128	384 (" ")	512
128	512 (" ")	640

Note: The memory configurations listed above are standard. The configurations listed below may not be available and are included to enhance understanding of the hardware. Contact the CT factory for availability of the configurations.

	0	256
256	32	288
256	64	320
256	96	352
256	128	480
256	256 (64K chips)	608
256	384 (" ")	640
256	512 (" ")	768
384 (64K chips)	0	384
384 (" ")	32	416
384 (" ")	64	448
384 (" ")	96	480
384 (" ")	128	512
384 (" ")	256 (64K chips)	640
384 (" ")	384 (" ")	768
384 (" ")	512 (" ")	896
512 (64K chips)	0	512
512 (" ")	32	544
512 (" ")	64	576
512 (" ")	96	608
512 (" ")	128	640
512 (" ")	256 (64K chips)	768
512 (" ")	384 (" ")	896**
512 (" ")	512 (" ")	1024*,**

* 32 Kilobytes of the address space must be reserved for permanent ROM space.

** 160 Kilobytes of the address space must be reserved for ROM space if the optional ROM Expansion Board is used.

Figure 2-9. RAM Configurations (kilobytes).

Typically, the operating system will write the entire contents of memory to disk upon parity-error detection and bootstrap a new version of the operating system.

The bottom 32 kilobytes of RAM, where the operating system typically resides, can be write-protected by setting a bit in the I/O Control Register (IOCR). If a write is attempted into this area, it is prevented and a non-maskable interrupt to the CPU is generated.

Multibus/Local-Bus Memory Mapping

The processor board contains a set of switches which define memory-space windows for local and Multibus masters. The windows determine whether memory resources on the Multibus or on the local bus will be addressed when an address is generated by a master. These windows are illustrated in Figure 1-4 of the prior chapter. Three choices of windows are available for local-bus masters and four choices are available for Multibus masters.

This scheme of windowing is intended to facilitate communication across the Multibus interface with either 8-bit or 16-bit microprocessor boards on the Multibus. It also tends to minimize the amount of physical memory chips needed for high-address communication. A small and large window are available in both directions.

The small window is generally intended for inter-processor communication. On the Multibus side, this window falls within the 64-kilobyte address space of 8-bit boards without interfering with the board's ability to access its own low-end or high-end operating memory. On the local-bus side, there will always be some RAM at the low end of the memory space where communication message buffers are located and Multibus masters will be able to access these locations.

The large window is generally intended for buffering of large I/O devices, such as tape drives. The method used for shifting the window across the Multibus interface simply involves forcing the high-order address bits one way or another.

Software Interface

Parity Error Register (PER)

RAM is checked for single-bit parity errors when enabled for this function through the IOCR. Even parity is used by default, but odd can be selected. If a parity error occurs and the IOCR is enabled for the detect function, a non-maskable interrupt occurs.

The Parity Register, at ports 52 (82 decimal) and 54 (84 decimal), is designed as a status register in the event of such an interrupt. This register has the following bit assignments:

<u>Port</u>	<u>Bit(s)</u>	<u>Function</u>
52 (82)	0-F	Low-order 16 bits of the address at which parity error occurred.
54 (84)	0-3	High-order 4 bits of the address at which parity error occurred.
54 (84)	4	When one, a word access caused the error. When zero, a byte access caused it.
54 (84)	5	When one, a DMA access caused the error.
54 (84)	6	When one, a SLAVE Multibus access caused the error.
54 (84)	D	When one, the most recent system Reset was by power-up. When zero, it was a manual Reset.
54 (84)	E	When one, the low byte of data was in error.
54 (84)	F	When one, the high byte of data was in error.

Figure 2-10 illustrates this register.

I/O AND CONTROL INTERFACES

I/O Peripheral Devices and Channels

As illustrated in Figure 2-1 only two I/O peripheral devices (the keyboard and the dual-communication-channel interface) are attached to the local I/O bus. Because this bus comes off the lower byte of the local data (MEM) bus, all port addresses and I/O data must be on even bytes. Three other I/O devices (video, disk, and printer) are located on the 16-bit TTL-buffered local data (MEM) bus, where speed and drive are higher. Additional I/O peripherals can be added on the Multibus. The video screen is refreshed completely from main memory.

Transfers involving video, disk and one of the two communications channels can use DMA service. All DMA transfers are word transfers. In the case of DMA transfers for video, hard disk and floppy disk, the full word is used since these devices are attached to the 16-bit local data (MEM) bus. In the case of DMA communications on channel A, however, only the low byte of each word is used since the communication path uses the local I/O bus, which sees only the lower 8 bits of the MEM bus.

When the CPU controls transfers, e.g., for the keyboard and printer, it can use byte instructions to peripherals on the local I/O bus and thereby avoid wasting the high byte of each word.

In a standard system, control of the peripheral devices is only done by the operating system, never by a user program or a Multibus master. The hardware control is not adequate to prevent all possible errors, such as Winchester disk and floppy disk simultaneously sending data to memory via DMA transfers (a shared DMA channel is used for both types of drives). For video displays, user programs running locally or on the Multibus can read and write to the video-refresh buffers in local memory, but the operating system handles all video control and status exchanges.

The CPU's I/O to peripheral devices on the Multibus can be controlled by a bit in the I/O Control Register (IOCR), which is described below in the section entitled "Interrupts". The local DMA logic, however, cannot control transfers

across the Multibus interface; DMA of this type can only be done between local memory and a Multibus I/O device using a DMA controller that is located on the Multibus.

I/O Circuits and Registers

There are a number of I/O ports on the various boards of the mainframe and disk subsystems. Nearly all of them are initialized and/or otherwise monitored by the operating system, although in a few cases user software may benefit from interaction with these ports. The descriptions here are followed by the port addresses relating to the functions; all port addresses are even (word-aligned). A reference to further programming details is given under the "Software Interface" section that follows.

DMA Control

The local DMA chip (8237-2 or 9517) on the processor board is used for video, disk and communication transfers, as well as for local-bus arbitration when a Multibus master is seeking access to local memory. Control words are written to the DMA chip for a variety of purposes, including the enabling/disabling (masking) of DMA transfers for particular I/O devices and the enabling/disabling of Multibus access to the local bus. Associated with the DMA chip is the Extended Address Register (EAR), a 3-bit register used in disk accesses to expand the DMA chip's 16-bit word addresses into 19-bit word addresses that cover the full 1 Megabyte address space (the least-significant address line is always held low). The EAR can be initialized by software at the beginning of each DMA access and the hardware will increment it thereafter. Ports 0 through 1C (24) and 68 (104).

Interrupt Control

The 8259A interrupt controller on the processor board, together with the I/O Control Register (IOCR) described below, handle CPU interrupts by the I/O peripherals and other local and Multibus logic. After initialization, the controller can be programmed for various forms of operation, including interrupt masking, end of interrupt,

and interrupt-priority rotation for user-added interrupts. Interrupt status can also be read from the chip. The "Special Fully Nested Mode" on this chip is not supported. Ports 20 (32) and 22 (34).

Keyboard Control and LEDs

An 8251A USART on the I/O-memory board is used primarily to receive communication from the keyboard, which contains its own dedicated firmware-programmed 8048 microprocessor. Transfers from the local bus to the 8251 are used for 8251 initialization after system Reset and for subsequent control bytes that light the 8 programmable LEDs located on the keyboard, set the keyboard in a diagnostics echo mode, or Reset the entire keyboard logic. Ports 38 (56) and 3A (58).

Interval Timer

A three-counter 8253 timer chip on the I/O-memory board is used to generate software-controlled baud rates for the two communication channels. The third counter is available to users and can be programmed to generate a maskable interrupt (level 3) periodically or. Ports 28 (40) through 30 (42).

I/O-Memory Board LEDs and Audio

Six latched LEDs and control for the speaker on the I/O-memory board are available for various functions controlled by users' software. The LEDs are used by the ROM during system boot to indicate various types of failures; thereafter, any application program can set or clear them. The audio signal can be driven in hardware at a standard 625 Hz tone or in software at any other audible frequency. Port 44 (68).

Communications Control

A Z80A-SIO (serial I/O) chip on the I/O-memory board controls the two communication channels. After mode initialization by library subroutine, the status of various communication and interrupt functions can be read from the chip. One of the channels (channel A) can be programmed to operate

either under DMA or CPU control; channel B is always operated by the CPU. Due to the SIO's non-8086 bussing structure, interrupt acknowledges by the CPU are made through a separate logic block called the Interrupt Acknowledge Register (IAR). An Extended Control Register (ECR) is also associated with the SIO; this programmable register adds some transmit/receive control functions that the SIO does not support, such as secondary receive/transmit and control lines used for RS-422 multidrop drivers. Ports 30 (48), 48 (72) through 4E (78), and 60 (96).

I/O Control Register (IOCR)

This 16-bit register on the I/O-memory board serves miscellaneous I/O-related interrupt enabling, status and other control functions. It communicates with the system over the local data (MEM) bus. The entire 16 bits can be read by the CPU and 8 of these bits can be written to perform the following functions:

- o Power-failure interrupt enable and status
- o Parity-error interrupt enable and status
- o Parity sense (odd/even)
- o Local-master/Multibus-slave I/O enable (factory option)
- o Printer interrupt enable and status
- o Bus-timeout interrupt enable and status
- o Realtime (50/60 Hz) clock interrupt enable and status
- o Memory-write interrupt enable and status
- o Programmable-timer interrupt status

In addition to providing these functions, the IOCR also pulls together all non-maskable interrupts and all level-4 maskable interrupts and forwards them to the CPU. Port 56 (86).

Parity Error Register (PER)

A register located on the processor board will latch the 20-bit address at which a RAM parity error occurs. It will also record whether the data was a word or byte, which byte in a word caused the error, and whether the last system Reset was manual or power-up; the latter distinction indicates whether or not the operating system should save the prior contents of memory onto disk. Ports 52 (82) and 54 (84).

Printer Status Register (PSR)

In addition to the 8-bit output data register, a 4-bit input status register is provided on the I/O-memory board for the printer. Depending on the status outputs available from the printer device, the register can indicate when the printer is selected, when its buffer is full, when it is busy printing, and when it is out of paper. Port 50 (80).

Screen Attribute Register (SAR)

The video control board contains a 16-bit control/status register that is used to set four global screen attributes (normal/reverse screen, screen blank, screen half-bright and 80/132 characters per row) and to direct DMA activity for screen refresh and symbol-RAM loading. Port 58 (88).

Floppy Disk Control (FDC)

The floppy disk control board contains an NEC 765 controller chip that is controlled by the CPU and an 8-bit Floppy Control Register (FCR), located on the FDC board. One of the two available ports for this logic is used for NEC 765 commands, transfer status and data transfers. The other port, when read, serves as the NEC 765 main status register; when written to, it serves as the Floppy Disk Control Register. Ports 70 (112) and 72 (114).

Winchester Disk Control (WDC)

The Winchester hard disk control board contains a firmware-programmed 8085 microprocessor and an 8-bit Winchester Control Register (WCR) which are used in a manner analogous to that described above for floppy disk control. Ports 78 (120) and 7A (122).

Software Interface

Software interfaces to the above-mentioned I/O circuit and registers are described in other parts of the Workstation Hardware Manual (WHM) and Peripherals Hardware Manual (PHM) as follows:

<u>I/O Logic</u>	<u>Manual</u>	<u>Chapter</u>	<u>Section</u>
DMA Control	WHM	2	DMA Trans- fers
Interrupt Control	WHM	2	Interrupts
Keyboard Control	WHM	2	Keyboard Interface
Interval Timer	WHM	2	Interrupts
I/O-Man.LEDs/Audio	WHM	2	Utilities
Comm. Control	WHM	2	Comm. Inter- face
IOCR	WHM	2	Interrupts
PER	WHM	2	Memory In- terface
PSR	WHM	2	Printer Interface
SAR	WHM	8	Functions & Interfaces
FDC	PHM	2	Functions & Interfaces
WDC	PHM	4	Functions & Interfaces

DMA TRANSFERS

General Operating Characteristics

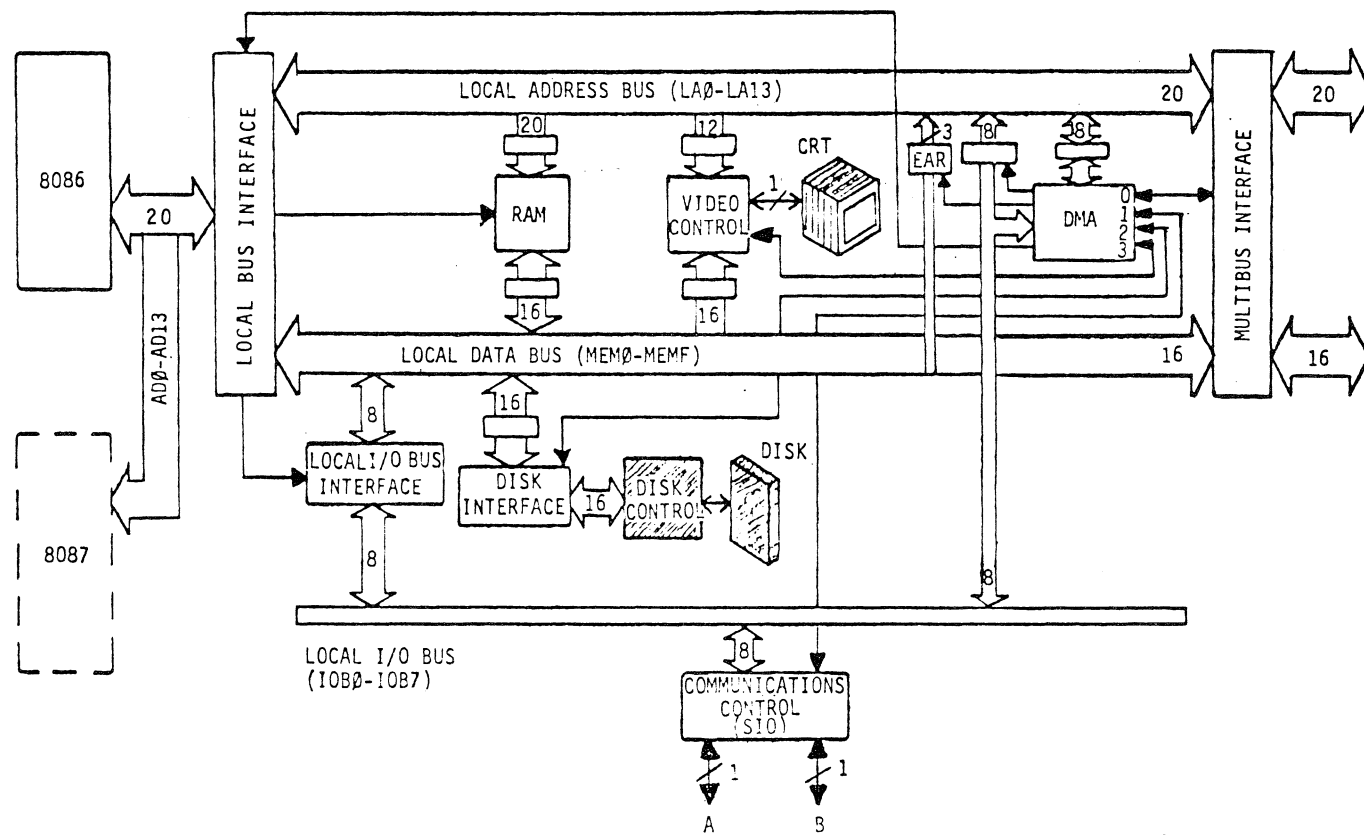
Direct memory access (DMA) transfers are done on the local bus by a four-channel DMA controller chip. The 8237-2 DMA chip is standard but the 9517 chip can alternatively be used. The DMA chip does not service transfers involving the Multibus, except to help arbitrate access by a Multibus master to the local bus memory. When data must be transferred by a local-bus master outward to the Multibus, only the local CPU can be used for this purpose. If DMA across the Multibus interface is desired, it must be handled by an external DMA controller on the Multibus.

Figure 2-11 shows the configuration of data, address-generation and control paths for local DMA logic. The CPU writes commands and reads status through the 8-bit data path connected via the low byte of the MEM bus to the local I/O bus; the commands include CPU-initiated requests for DMA service, starting addresses, transfer block lengths, and enable/disable bits which mask the ability of individual peripheral-device channels to request their own DMA service. The control lines in and out of the DMA chip are used for local-bus request/acknowledge, I/O peripheral request/acknowledge, and ancillary register or buffer control.

Memory and I/O read/write commands are generated by the DMA logic and distributed through the local bus interface. The DMA logic then generates addresses for the RAM memory on the local address bus and transfers of data occur on the 16-bit local data (MEM) bus. Transfers always have RAM memory at one end of the path, either at the source or the destination. Memory-to-memory, I/O-to-I/O and ROM accesses are not supported under DMA control.

The DMA controller chip, either 8237-2 or 9517, operates at 4 MHz. The chip has 8-bit data and address paths, although its ability to use both paths together allows them to generate 16 bits of address internally. An additional 3 bits of address are generated by an external register as described below under "Address Generation."

Figure 2-11. DMA Block Diagram



Maximum transfer rate on the local bus with the DMA chip is 2.64 Megabytes/second. Since all transfers are words (the least-significant address bit is always forced to zero), this is equal to 1.32 Mega-transfers/second, the maximum rated speed of the chip. This rate is achieved with a "simultaneous" transfer scheme in which data is read from the source in essentially the same transfer cycle in which it is written to the destination. The transfer cycle for an 8237-2 varies between 3 and 6 clock cycles, depending on the DMA mode, address and I/O peripheral involved in the transfer.

Local DMA transfers uses either the single (word-at-a-time) or the demand transfer mode, depending on the channel. When demand mode is used, hardware logic causes bursts of multiple-word transfers within a single DMA-request cycle for certain peripherals. DMA activity that is controlled by a Multibus DMA controller, however, can only use the single transfer mode, since there is no way to communicate a block length across the Multibus interface. Therefore, all DMA transfers across the Multibus interface are subject to interleaving with other local CPU or DMA transfers on the local bus.

Channel Uses

The DMA chip has four prioritized channels. Each channel has its own request/acknowledge pin, 16-bit starting-address register and 16-bit address-counting register. I/O peripherals use one of the four request/acknowledge signal lines shown in Figure 2-11 to request service.

The CPU maintains software control over these priorities by having the ability to write mask bits to the DMA chip's mask register, thereby enabling or disabling individual I/O devices from requesting DMA service, and by having the ability to change from fixed to rotating priority. One further consideration with respect to DMA priority involves memory refresh: the refresh will always take priority over a DMA transfer for access to RAM.

Figure 2-12 and the sections below outline the nature of activity on the four DMA channels.

Multibus Access

While the local DMA chip does not service transfers across the Multibus interface, it is involved in arbitrating access to the local bus by Multibus masters. When a Multibus master puts an address on the Multibus which corresponds to an established address window into the local bus (see the "Memory Interface" section above), the Multibus interface derives a DMA-request signal from the address. The DMA logic communicates with the local bus interface to obtain bus control, which it then turns over to the Multibus master. Thereafter, all address-generation and control come from that master and addresses generated by the DMA chip are ignored.

Multibus access to the local bus is given highest priority because it is assumed that some Multibus devices, such as realtime devices, will not have their own storage buffering and will require very fast access to local RAM. All local-bus devices, by contrast, have their own storage buffering and can wait longer for local-bus access if necessary.

Communications

DMA communication transfers can occur only through Channel A of the Z80A-SIO serial communications controller chip. Channel B of that chip can only be serviced by the CPU. To obtain DMA service the SIO requests it through its Wait/Ready line (the Ready output activates the request).

Only 16 address bits are available for communications DMA. This means that memory buffers constructed in software must be located in the first 128 kilobytes of memory, since 16 bits address 64K words (128K bytes). Figure 2-13 illustrates this constraint, which also applies to video buffers. Furthermore, the SIO only looks at the lower 8 bits of the local data (MEM) bus via its local I/O bus interface. Therefore, only the low byte of every word transfer is used.

Figure 2-12. DMA Priorities and Characteristics

	<u>DMA CHANNEL (PRIORITY)</u>	<u>SOURCE OF ADDRESS GENERATION</u>	<u>ADDRESS BITS GENERATED</u>	<u>DATA TYPES ADDRESSED</u>	<u>ADDRESS SPACE (BYTES)</u>	<u>DATA TYPES USED</u>	<u>DMA TRANSFER MODE</u>	<u>MAXIMUM BYTES OR WORDS IN TRANSFER</u>
MULTIBUS	0	MULTIBUS	20	BYTE, WORD	≤ 1M	BYTE, WORD	SINGLE	1
COMMUNICATIONS (CHANNEL A ONLY)	1	LOCAL DMA	16	WORD	128K	LOW BYTE	DEMAND	1
DISK: FLOPPY WINCHESTER	2	LOCAL DMA	19	WORD	1M	WORD	DEMAND	1
	2	LOCAL DMA	19	WORD	1M	WORD	DEMAND	8*
VIDEO	3	LOCAL DMA	16	WORD	128K	WORD	DEMAND	3**

* BURSTS OF MORE THAN ONE WORD PER TRANSFER ARE ENFORCED BY HARDWARE EXTERNAL TO THE DMA WHICH MAINTAINS THE PERIPHERALS DREQ LINE ACTIVE EXCEPT WHEN AN END OF PROCESS (EOP) IS RECEIVED.

** SIMILAR TO DISK-TRANSFER BURSTS EXCEPT THAT THE FIRST TRANSFER OF A SCREEN-ROW BLOCK IS FOUR WORDS LONG AND ALL SUBSEQUENT TRANSFERS ARE THREE WORDS LONG.

DMA communication on Channel A is limited to half-duplex operation. While the SIO itself has full-duplex channels, the arrangement with the DMA logic requires a service-request line for each direction and the SIO can generate only one service-request signal for each channel.

DMA transfers for the SIO require Wait states. This is handled by logic which is enabled from the SIO/DMA request/acknowledge handshake and timed by the DMA chip's Write strobe.

Disk

A common DMA channel is used for both floppy and hard disk accesses. The control boards for both types of disk are capable of generating a DMA request and for this reason it is essential that operating-system supervision over these requests on the common signal line be maintained.

The control boards stack bytes into words for the disk control interface and the MEM bus, making bus utilization more efficient.

Unlike communications DMA, disk DMA transfers can access the entire 1-megabyte memory space. The address bits are added at the high end by the software-initialized Extended Address Register (EAR), which is explained below under the section titled "Address Generation."

Video Refresh

Video DMA is similar to communications DMA in that only 16 address bits are used, hence the requirement to store the software screen buffer(s) within the lower 128 kilobytes of memory. Figure 2-13 shows this. The two methods differ, however, in that the video control logic watches the full 16-bit data bus and sends both bytes of data to the CRT.

Transfers of data from memory to the CRT needn't require reinitialization by the CPU for each screen frame. The DMA chip's auto-initialize function is normally used for this, causing automatic restarts after each screen frame is fully transferred.

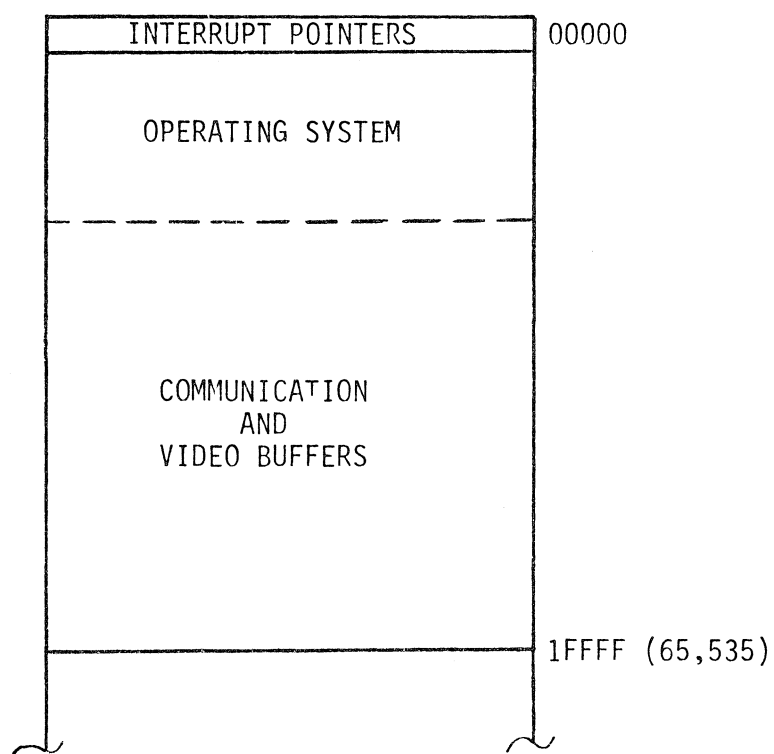


Figure 2-13. Location of Communication and Video Buffers

Transfers from the video board to the processor board are also handled by the DMA logic but they occur only rarely for status checking. After the above four DMA functions are satisfied, priority for the local bus defaults to the CPU.

Bus Exchange

Upon receipt of a service request from a local peripheral, the DMA chip first obtains control of the local bus and then sends the peripheral an acknowledgement which enables the peripheral to transmit or receive its data on the data bus. The peripheral's request line and the DMA chip's acknowledge line are normally held active for a maximum transfer duration of 1 to 8 words, as shown in Figure 2-12. In the case of Winchester disk and video transfers, the maximum duration involves multiple words and is enforced by special hardware logic that interfaces the peripheral's request line to the DMA chip.

The DMA chip obtains control of the local bus by asserting its Hold Request output to the arbitration logic in the local bus interface and awaiting a Hold Acknowledge in return. The arbitration logic uses the CPU's status lines to determine when the 8086 or 8087 is using or preparing to use the bus. At the earliest opportunity, the Hold Acknowledge is sent to the DMA chip, which then takes control of the bus by holding down the CPU's Ready line. This causes the CPU to idle for the duration of the DMA activity.

If a Multibus master is seeking access to the local bus, it first must gain access to the Multibus using the conventional Multibus protocol. Then it places the address on the Multibus. The Multibus address decode logic determines whether a valid local-memory window is being addressed. When this occurs, the same decode logic generates a Multibus service request (SLAVERQ) to the DMA logic. The DMA logic then obtains control of the local bus and returns an acknowledgement to the Multibus interface, which is thereby enabled to put read/write commands on the local bus.

Address Generation

In Multibus accesses to the local bus, addresses are generated by the Multibus master. In the other three types of DMA transfers, addresses are generated by the local DMA logic: communication and video addresses are generated by one method and disk addresses are generated by another method.

Communication and Video Addresses

In these transfers, only 16 address bits are placed on the 20-bit local address (LA) bus by the DMA chip. The low-order byte of the address comes from the DMA-chip's address outputs and the high-order byte is multiplexed onto the local I/O bus (IOB) through the DMA chip's data lines. The high-order 3 bits plus the low-order bit of the local address bus are forced to zero. This means that memory buffers constructed in software must be located in the first 128 kilobytes of memory.

Before loading the DMA chip with a base (starting) address, the actual starting address is shifted right one bit by the operating system. This allows the DMA chip to count words instead of bytes on its 16-bit internal counter and address incrementer, but it means that starting addresses must be word boundaries; i.e., they must be even addresses. This shifting is transparent to the user and is reversed automatically by the DMA logic when it puts addresses onto the local address bus, since the DMA-logic bus lines are connected in hardware to the next higher local address bus line.

Figure 2-14 gives a pictorial overview of how DMA addresses are manipulated in system software and hardware during this process.

Disk Addresses

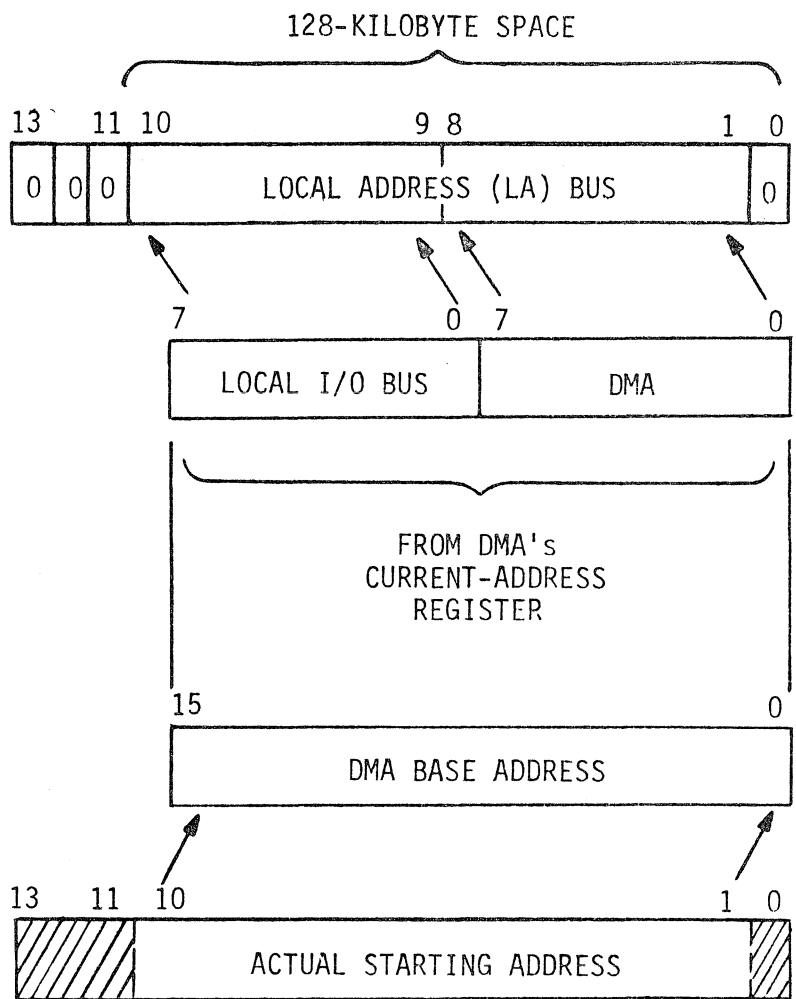
Unlike communications and video DMA, disk DMA transfers can access the entire 1-megabyte address space. The low-order address bit is still zero, allowing the DMA chip to count word transfers in its 16-bit registers instead of byte transfers. But in disk transfers, three additional bits on the high end are supplied by the Extended Address Register (EAR). This gives a

total of 19 address bits which address words, hence the equivalent of 20 bits addressing bytes on word boundaries.

The EAR is initialized by the disk-handler portion of the operating system each time a disk access is started. Once initialized, however, it is incremented in hardware on 64K-word (128-kilo-byte) boundaries, when the capacity of the DMA-chip's counter is reached.

The one-bit address shift in operating-system software is still required prior to loading the DMA registers, just as in communication and video addresses, and it is corrected upon DMA address output in the same hardware manner. This process, like the initializing of the EAR, are generally transparent to the user.

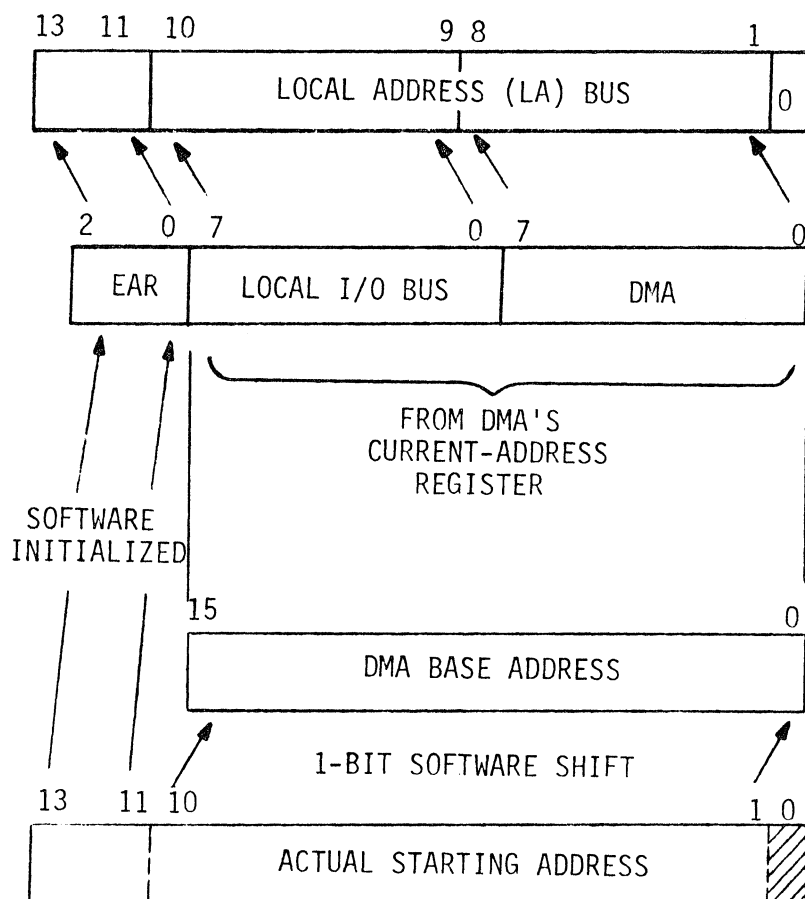
Figure 2-15 illustrates the address-generation technique used for disk accesses. Compare this with the technique illustrated in Figure 2-14 for communication and video accesses.



KEY:  - DON'T CARE

ALL BIT NUMBERS ARE IN HEX.

Figure 2-14. Communication and Video Address Generation



KEY:  - DON'T CARE

ALL BIT NUMBERS ARE IN HEX.

Figure 2-15. Disk Address Generation

Software Interface

8237-2 DMA Controller

The 8237-2 DMA chip is standard with the system, although a 9517 may be used. This section describes only the 8237-2; the 9517 is identical in operation.

The DMA logic is initialized and monitored by the CPU through a set of I/O ports. These ports, which address a set of registers described below, are as follows:

<u>I/O Port</u>	<u>Write Register</u>	<u>Read Register</u>
0	Reserved	Reserved
2	Reserved	Reserved
4	SIO base address	SIO current address
6	SIO base count	SIO current count
8	Disk base address	Disk current address
A (10)	Disk base count	Disk current count
C (12)	Video base address	Video current address
E (14)	Video base count	Video current count
10 (16)	Command	Status
12 (18)	Request	Not used
14 (20)	Set/reset mask	Not used
16 (22)	Mode	Not used
18 (24)	Clear flip-flop	Not used
1A (26)	Master clear	Not used
1C (28)	Mask	Not used
68 (104)	EAR*	EAR*

* See separate section
with this title

When the standard operating system is used, all initialization and control of the DMA logic is done by the operating system. Under these circumstances, you should have no need to write to any of the above ports. The following information will be needed only when a non-standard operating system is used.

Registers: The 8237 has four channels for I/O devices, each with its own request and acknowledge lines. All four are enabled by the standard operating system. Since the Multibus channel is only used for arbitrating access of a Multibus master to the local bus, its channel is not programmed with base addresses and counts as the other three channels are. The register formats are illustrated in Figure 2-16. Their contents are as follows:

Current Address Register: Each channel has a 16-bit Current Address Register. This register holds the value for the address used during DMA transfers. The address is automatically incremented or decremented after each transfer and the intermediate values of the address are stored in the Current Address register during the transfer. This register is written or read by the CPU in successive 8-bit bytes. It may also be reinitialized by an Autoinitialize back to its original value. Autoinitialize takes place only after an End of Process (EOP).

Current Word Count Register: Each channel has a 16-bit Current Word Count register. This register holds the number of transfers to be performed. The word count is decremented after each transfer. The intermediate value of the word count is stored in the register during the transfer. When the value in the register goes to zero, a Terminal Count (TC) will be generated. This register is loaded or read in successive 8-bit bytes by the CPU in the Program Condition. Following the end of a DMA service it may also be reinitialized by an Autoinitialization back to its original value. Autoinitialization can occur only when an EOP occurs.

Base Address and Base Word Count Registers: Each channel has a pair of Base Address and Base Word Count Registers. These 16-bit registers store the original value of their associated current registers. During Autoinitialize these values are used to restore the current registers to their original values. The base registers are written simultaneously with their corresponding current register in 8-bit bytes in the Program Condition by the CPU. These registers cannot be read by the microprocessor.

Command Register: This 8-bit register controls the operation of the 8237. It is programmed by the CPU and cleared by Reset.

Mode Register: Each channel has a 6-bit Mode register associated with it. When the

Mode register associated with it. When the register is being written to by the CPU in the Program Condition, bits 0 and 1 determine which channel Mode register is to be written.

Request Register: This register is not used.

Mask Register: Each channel has associated with it a mask bit which can be set to disable the incoming DREQ. Each mask bit is set when its associated channel produces an EOP if the channel is not programmed for Autoinitialize. Each bit of the 4-bit Mask register may also be set or cleared separately under software control. The entire register is also set by a Reset. This disables all DMA requests until a clear Mask register instruction allows them to occur. The instruction to separately set or clear the mask bits is similar in form to that used with the Request register. All four bits of the Mask register may also be written with a single command.

Status Register: The Status register is available to be read out of the 8237 by the CPU. It contains information about the status of the devices at this point. This information includes which channels have reached a terminal count and which channels have pending DMA requests. Bits 0-3 are set every time a TC is reached by that channel or an external EOP is applied. These bits are cleared upon Reset and on each Status Read. Bits 4-7 are set whenever their corresponding channel is requesting service.

Temporary Register: The Temporary register is not used in this configuration, since memory - to memory transfers are not supported.

Software Commands: These are additional special software commands which can be executed in the Program Condition. These commands are decoded as sets of addresses with the chip select (CS) and I/O write (IOW) lines. The two software commands are:

1. Clear First/Last Flip-Flop: This command is executed prior to writing or

reading new address or word count information to the 8237. This initializes the flip-flop to a known state so that subsequent accesses to register contents by the CPU will address upper and lower bytes in the correct sequence.

2. **Master Clear:** This software instruction has the same effect as the hardware Reset. The Command, Status, Request, Temporary, and internal First/Last Flip-Flop registers are cleared and the Mask register is set. The 8237 will enter the idle cycle.

Idle Mode: When no channel is requesting service, the 8237 will sample the DREQ lines every clock cycle to determine if any channel is requesting a DMA service. The device will also sample the chip select (CS) pin looking for an attempt by the CPU to write or read the internal registers of the 8237. When CS is low and HRQ is low, the 8237 enters the Program Condition in which the CPU can read from or write to the internal registers.

Active Modes: When the 8237 is in the idle mode and a channel requests DMA service, the device will output an HRQ and enter an Active cycle in one of two modes:

Single Transfer Mode: In Single Transfer mode the device is programmed to make one transfer only. The word count will be decremented and the address decremented or incremented following each transfer. When the word count goes to zero, a Terminal Count (TC) will cause an Autoinitialize if the channel has been programmed to do so.

Demand Transfer Mode: In Demand Transfer mode, which is used for all local I/O devices by the standard operating system, the device is programmed to continue making transfers until a TC or external EOP is encountered or until DREQ goes inactive. Thus transfers may continue until the I/O device has exhausted its data capacity. After the I/O device has had a chance to catch up, the DMA service is re-established by means of a DREQ. During the time between services when the CPU is allowed to operate, the intermediate values of address and word

count are stored in the 8237 Current Address and Current Word Count registers. Only an EOP can cause an Autoinitialize at the end of the service.

Transfer Types: Each of the three active transfer modes can perform two different types of transfers. These are Read and Write. Write transfers move data from an I/O device to the memory by activating MEMW and IOR. Read transfers move data from memory to an I/O device by activating MEMR and IOW.

Autoinitialize: By programming a bit in the Mode register, a channel may be set up as an Autoinitialize channel. During Autoinitialize initialization, the original values of the Current Address and Current Word Count registers are automatically restored from the Base Address and Base Word Count registers of that channel following EOP. The base registers are loaded simultaneously with the current registers by the CPU and remain unchanged throughout the DMA service. The mask bit is not set when the channel is in Autoinitialize. Following Autoinitialize the channel is ready to perform another service without CPU intervention.

Priority: The 8237 has two types of priority encoding, fixed and rotating, available as software selectable options. Fixed Priority is used by the CT operating system, with the priority order based upon the descending value of their number. The channel with the lowest priority is 3 followed by 2, 1 and the highest priority channel, 0. After the recognition of any one channel for service, the other channels are prevented from interfering with that service until it is completed.

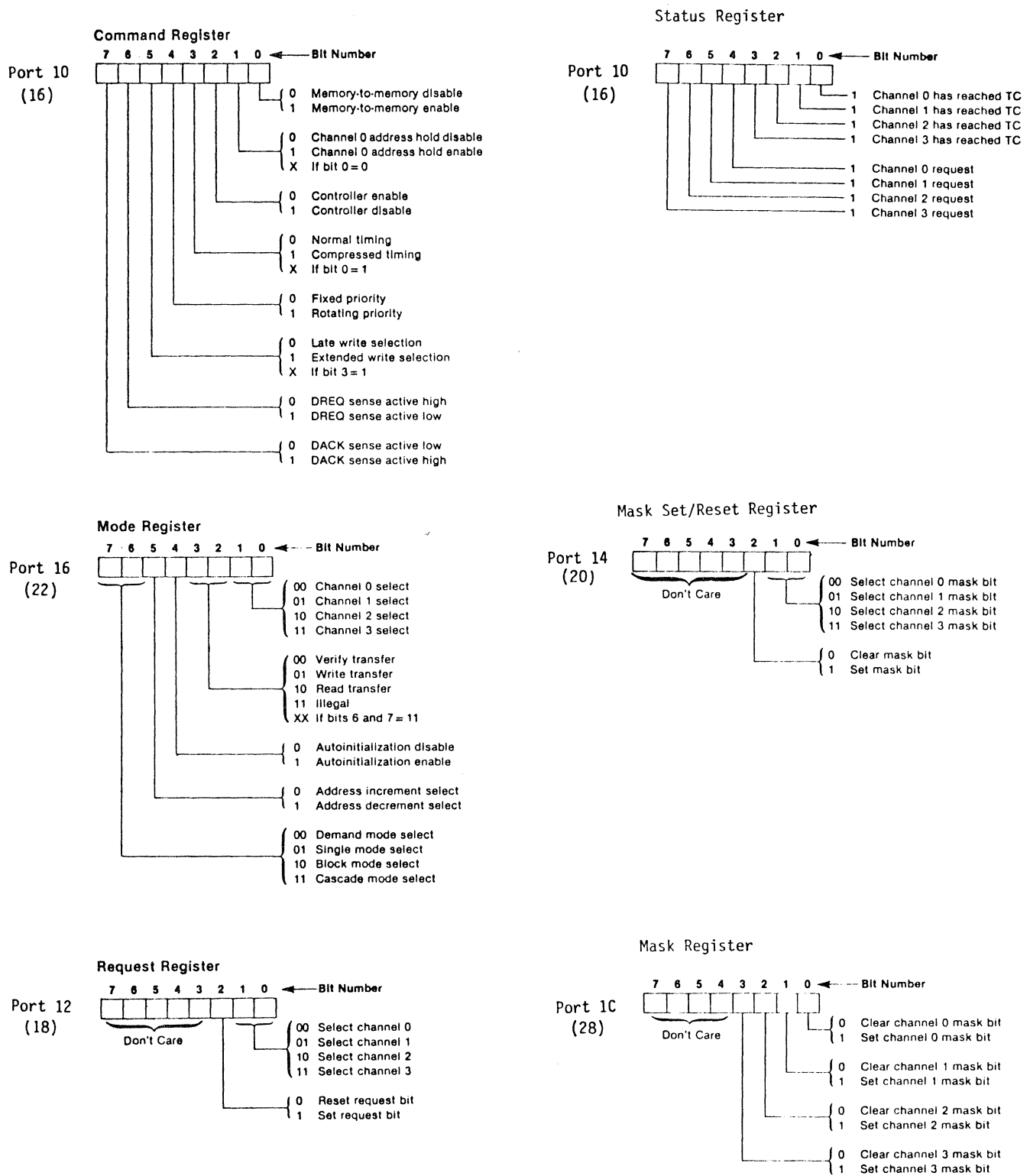


Figure 2-16. 8237-2 Register Formats

DMA Extended Address Register (EAR)

The EAR is a 3-bit register which is written and read from I/O port 68 (104 decimal). It is used only for disk transfers and must be used for all such transfers.

The EAR is loaded by the operating system prior to a disk transfer. Thereafter, it is incremented by hardware on 128-kilobyte boundaries that are counted from the 8237's 16-bit word counter.

A discussion of the DMA address-generation method is given earlier in this chapter. The format for writing to the EAR is given in Figure 2-17, below. The EAR is loaded from the three low-order bits of the local data (MEM) bus.

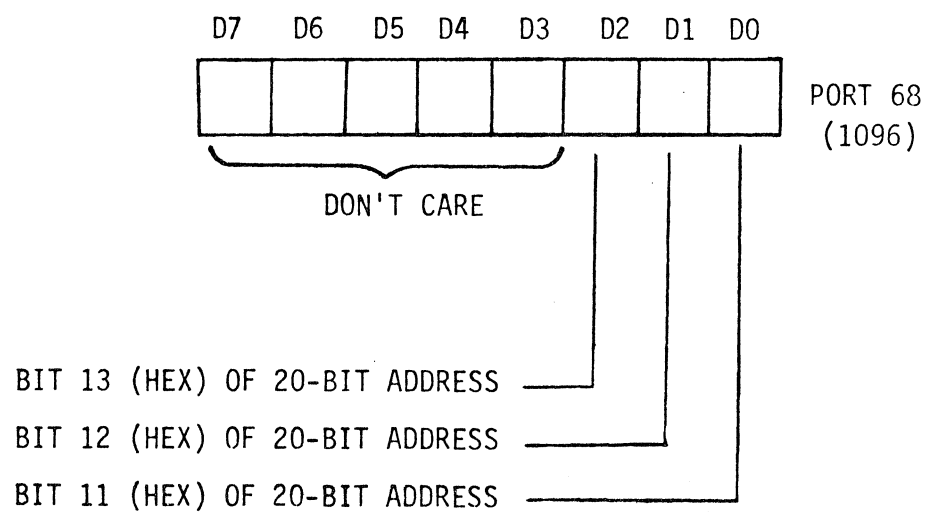


Figure 2-17. DMA Extended Address Register (EAR)

INTERRUPTS

The local bus supports both non-maskable (NMI) and maskable (INTR) interrupts of the 8086. An 8259A interrupt controller supports up to eight priority levels for the maskable interrupts. Of these priority levels, four are reserved for local-bus use and the other four are available for Multibus use. Up to 36 maskable-interrupt priority levels can be supported on Multibus when additional priority-encoding logic (such as additional 8259As) are provided on the Multibus.

While certain features of the way in which interrupt requests are passed to the processor are unique, the manner in which the interrupts are processed is standard for the 8086.

Figure 2-18 lists the types of interrupts supported, along with their usage on the local bus and connections through to the Multibus. Figure 2-19 illustrates the general arrangement of control lines used for passing interrupts to the CPU from the interrupting devices and logic blocks.

Non-Maskable Interrupts

In conventional usage, a "non-maskable interrupt" cannot be masked by software. In the usage implemented on the local bus, however, it can be masked although not through the 8086 itself. This is possible because all inputs to the 8086's NMI pin pass through the I/O Control Register (IOCR), a programmable I/O port that contains an enable/disable mask bit for each interrupting source. In this implementation, the important distinction between non-maskable and maskable interrupts is that the former type have higher priority and are able to get the CPU's attention faster.

Non-maskable interrupts are not part of the Multibus convention. Therefore, these interrupts go only to the local 8086. They come from four sources, each of which has a separate line into the IOCR where the input is latched. Any of the inputs will cause the IOCR to send an NMI to the 8086, which can then read the IOCR status register to determine the cause of the interrupt.

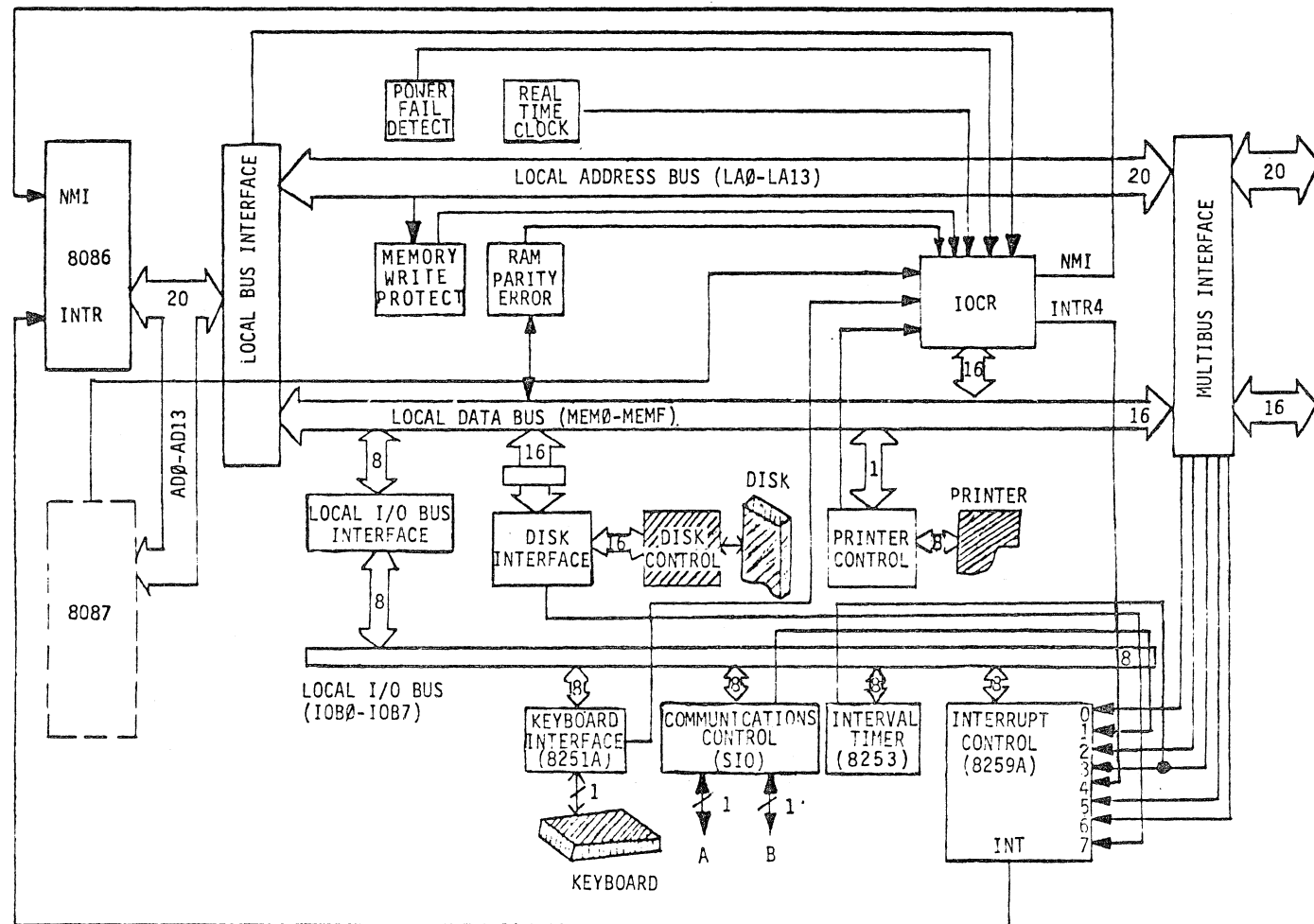
<u>Priority (Highest First)</u>	<u>Local-Bus Usage</u>	<u>Interface To CPU</u>	<u>Local-Bus/Multibus Connection?</u>
Non-Maskable	Power Failure, Parity Error, Write Protect, Bus Timeout.	IOCR	No
Maskable, level 0	Undefined	8259A	Yes
" , level 1	Communications	8259A	No*
" , level 2	Undefined	8259A	Yes
" , level 3	Interval Timer	8259A	Yes**
" , level 4	Printer, Keyboard, Realtime Clock, 8087.	IOCR and 8259A	No*
" , level 5	Undefined	8259A	Yes
" , level 6	Undefined	8259A	Yes
" , level 7	Hard Disk, Floppy Disk.	8259A	No*

* While these interrupt lines are not connected between the local bus and the Multibus, they are connected between all Multibus boards.

** If the local interval timer is not used, this interrupt level can be used by Multibus devices.

Figure 2-18. Interrupt Priorities and Functions

Figure 2-19. Interrupt Block Diagram



NOTE: Shaded elements are not part of mainframe.

All interrupt latches in the IOCR are disabled and cleared upon system Reset. Normally, they are reenabled by the operating system's initialization routine.

Power Failure

Logic on the I/O-memory board can detect AC power transitions. At a line frequency of 60 Hz, for example, the transitions occur at a 120-Hz rate or every 8.33 milliseconds. Capacitors on the power supply are guaranteed to keep the CPU running for at least 12 milliseconds after the AC power is interrupted. Worst-case detection of power failure would then leave 12 minus 8.33, or about 3.67 milliseconds after the failure to take defensive action.

When a missing line transition is detected, a signal is sent from the power-failure detection logic to the IOCR, causing a non-maskable interrupt to the CPU. At the same time, the detection logic disables the disk-control interface so that disks cannot be corrupted beyond any concurrent sector operation.

If the power comes up again before the operator turns the power switch off, the system will automatically Reset and reboot itself. This function is called Auto-Restart.

Parity Error

The parity detection logic on the processor board can detect RAM (not ROM) parity errors and cause the IOCR to assert NMI. When an error occurs, the 20-bit address at which it occurred is stored in the Parity Error Register (PER), from which it can be read by the CPU. The PER is described above, in the "Memory Interface" section.

Memory Write-Protection

Logic on the I/O-memory board can generate an NMI through the IOCR when an attempt is made to write in the lower 32 kilobytes of RAM, where the operating system normally resides. In addition to an interrupt being generated, the write is also inhibited.

Bus Timeout

Logic on the processor board can be enabled to monitor bus cycles. If it finds that none has occurred within 6 milliseconds after a read/write command to memory or an I/O device, it will assert NMI through the IOCR. The timeout will affect all slave accesses on the local bus, whether by local-bus or by Multibus masters. It is designed to protect against non-existent devices and faulty programs.

Maskable Interrupts

The 8259A handles maskable interrupt requests. Figure 2-18 shows four of the 8259A's interrupt inputs (levels 1, 3, 4 and 7) being used by local I/O peripherals. These four are as follows:

Communications (Level 1)

Interrupts from the Z80A-SIO go directly to the level 1 input on the 8259A, where they are then passed to the CPU in their proper priority order, i.e., after any pending level 0 interrupts.

The SIO can generate an interrupt when (a) it is ready to transmit a message, (b) it has received a message, or (c) a noteworthy external or internal status condition has occurred (such as loss of carrier). These functions are independently enabled in software, with Channel A having higher fixed priority than Channel B.

Interval Timer (Level 3): This input to the 8259A is electrically connected to the Multibus, as well as to counter 0 of 8253 interval timer. If the timer is not used, the interrupt level can be used by a Multibus device.

Printer, Keyboard, Realtime Clock, and 8087 (Level 4)

The interrupts from these sources are routed through independent signal lines to the IOCR. They are latched in the IOCR until sent on through a common line to the 8259A's level-4 input. The printer and realtime clock interrupts can be independently enabled or disabled by setting a mask bit in the IOCR.

If an interrupt occurs at level 4, the 8259A will identify it only as that level, not as the specific device that caused the interrupt. It is then up to the CPU to individually poll the possible sources that the 8259A has multiplexed onto that level. The following registers are available for this:

Printer: The printer interface has a status register port 50 (80 decimal), that can be read after an interrupt to determine the cause, in addition to a status bit in the IOCR.

Keyboard: The keyboard interface has an 8251A status register, port 3A (58 decimal).

Realtime Clock: This clock sets a bit (bit 1) in the IOCR, port 56 (186 decimal), at a rate of 50 or 60 Hz, which can be read after an interrupt.

8087: The 8087, which interrupts on errors, has a status register which can be examined after an interrupt by using its own instruction set.

Disks (Level 7)

Both the floppy and Winchester disk controller boards share this common interrupt line through the disk interface to the 8259A. Interrupts on the line normally occur at the completion of a DMA block transfer or other command to the controller. Interrupts may also occur on error or abnormal conditions, such as when a diskette is removed and the drive becomes not ready.

Software Interface

8259A Interrupt Controller

The 8259A has eight maskable interrupt-request inputs, as shown in Figure 2-18. When the standard operating system is used, all initialization and control for priority levels 1, 3, 4 and 7 are done by the operating system. Under these circumstances, users will only have occasion to write command bytes that relate to interrupt levels accessible from the Multibus (i.e., levels 0, 2, 5 and 6).

Commands: The 8259A accepts two types of command words generated by the CPU:

1. Initialization Command Words (ICWs): Before normal operation can begin, each 8259A in the system must be brought to a starting point - by a sequence of 2 to 4 bytes timed by WR pulses.
2. Operating Command Words (OCWs): These are the command words that are sent to the 8259A for various forms of operation, such as:
 - o Interrupt Masking
 - o End of Interrupt
 - o Priority Rotation
 - o Interrupt Status

The OCWs can be written into the 8259A anytime after initialization.

The interrupts at the interrupt-request inputs are handled by two registers in cascade: the Interrupt Request Register (IRR) and the In-Service Register (ISR). The IRR is used to store all the interrupt levels which are requesting service, and the ISR is used to store all the interrupt levels which are being serviced.

An Interrupt Mask Register (IMR) stores the bits which mask the interrupt lines. The IMR operates on the IRR. Masking of a higher-priority input will not affect lower-priority inputs.

The I/O ports used for writing and reading the above commands and status are as follows:

<u>Port</u>	<u>Write</u>	<u>Read</u>
20 (32)	ICW1	IRR
	OCW2	ISR
	OCW3	
22 (34)	OCW1	IMR
	ICW2	
	ICW3	
	ICW4	

Initialization Command Words: Figure 2-20 shows the sequence of initialization and Figure 2-21 shows the command word formats.

ICW1 and ICW2: T7-T3 are inserted as a portion of an interrupt vector in ICW2. The 8259A sets the remaining bits according to the interrupt level. A₁₀-A₅ are ignored and ADI (Address Interval) has no effect.

ICW3: This word is read only when there is more than one 8259A in the system and cascading is used, in which case SNGL=0. It will load the 8-bit slave register. The functions of this register are:

- a. With the 8259A in the master mode, as is always the case for the CT mainframe, (in buffered mode when M/S= 1 in ICW4) a "1" is set for each slave in the system. The master then will enable the corresponding slave to release byte 2 through the cascade lines.
- b. With the 8259A in the slave mode, as may be used in cascaded 8259As on the Multibus, (in buffered mode when M/S = 0 in ICW4) bits 2-0 identify the slave. The slave compares its cascade input with these bits and if they are equal, byte 2 of the call sequence is released by it on the Data Bus.

ICW4: The Special Fully Nested mode is not supported and buffered mode should be used. Bit D0 should be set to 1.

Operation Command Words: After the Initialization Command Words (ICWs) are programmed into the 8259A, the chip is ready to accept interrupt requests at its input lines. However, during the 8259A operation, a selection of algorithms can

command the 8259A to operate in various modes through the Operation Command Words (OCWs).

Figure 2-22 shows the operation command word formats.

OCW1: OCW1 sets and clears the mask bits in the interrupt Mask Register (IMR). M_7-M_0 represent the eight mask bits. $M = 1$ indicates the channel is masked (inhibited), $M = 0$ indicates the channel is enabled.

OCW2: R, SL, EOI - These three bits control the Rotate and End if interrupt modes (see "Modes of Operations", below) and combinations of the two. A chart of these combinations can be found in the Operation Command Word formats.

L_2, L_1, L_0 - These bits determine the interrupt level acted upon when the SEOI bit is active.

OCW3: ESMM - Enable Special Mask Mode (see "Modes of Operation", below). When this bit is set to 1 it enables the SMM bit to set or reset the Special Mask Mode. When ESMM = 0 the SMM bit becomes a "don't care".

SMM - Special Mask Mode (see "Modes of Operation", below). If ESMM = 1 and SMM = 1 the 8259A will enter Special Mask Mode. If ESMM = 1 and SMM = 0 the 8259A will revert to normal mask mode. When ESMM = 0, SMM has no effect.

Interrupt Mask: Each Interrupt Request input can be masked individually by the Interrupt Mask Register (IMR) programmed through OCW1. Each bit in the IMR masks one interrupt channel if it is set (1). Bit 0 masks IR0, Bit 1 masks IR1 and so forth. Masking an IR channel does not affect the other channel operation.

Modes of Operation: The 8259A has several modes of operation. However, the only operating mode used in the CT system is the fully nested "level triggered" mode. The priority scheme allowed within the CT system is "fixed priority".

Reading Status: The input status of several internal registers can be read to update the user information on the system. The following regis-

ters can be read via OCW3 (IRR and ISR) or OCW1 (IMR).

Interrupt Request Register (IRR): 8-bit register which contains the levels requesting an interrupt to be acknowledged. The highest request level is reset from the IRR when an interrupt is acknowledged. (Not affected by IMR).

In-Service Register (ISR): 8-bit register which contains the priority levels that are being serviced. The ISR is updated when an End of Interrupt command is issued.

Interrupt Mask Register (IMR): 8-bit register which contains the interrupt request lines which are masked. The IRR can be read when, prior to the RD pulse, a Read Register Command is issued with OCW3 (RR = 1, RIS = 0). The ISR can be read when, prior to the RD pulse, a Read Register Command is issued with OCW3 (RR = 1, RIS = 1). There is no need to write an OCW3 before every status read operation, as long as the status read corresponds with the previous one; i.e., the 8259A "remembers" whether the IRR or ISR has been previously selected by the OCW3. This is not true when poll is used. For reading the IMR, no OCW3 is needed. The output data bus will contain the IMR whenever RD is active and A0 = (OCW1). Polling overrides status read when P = 1, RR = 1 in OCW3.

Figure 2-23 shows the status-register format and Figure 2-24 summarizes the 8259A instruction set.

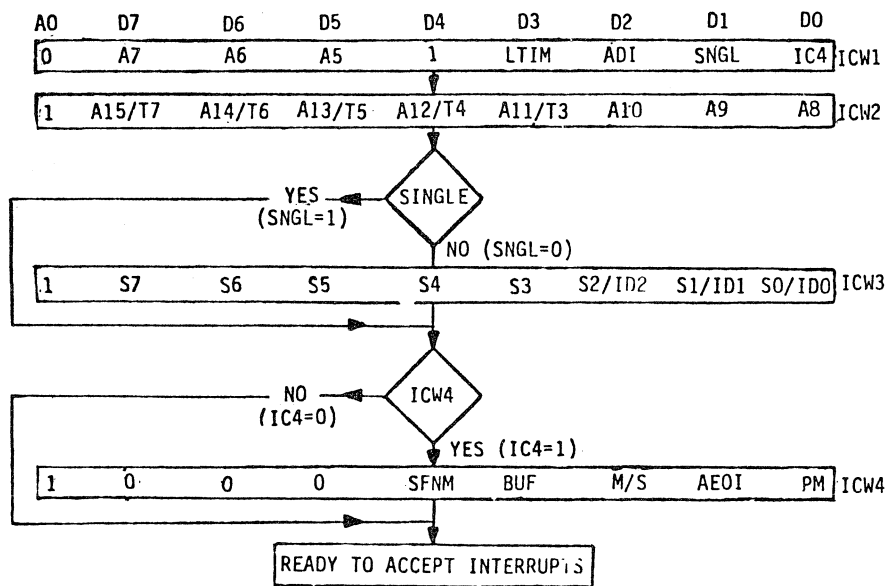
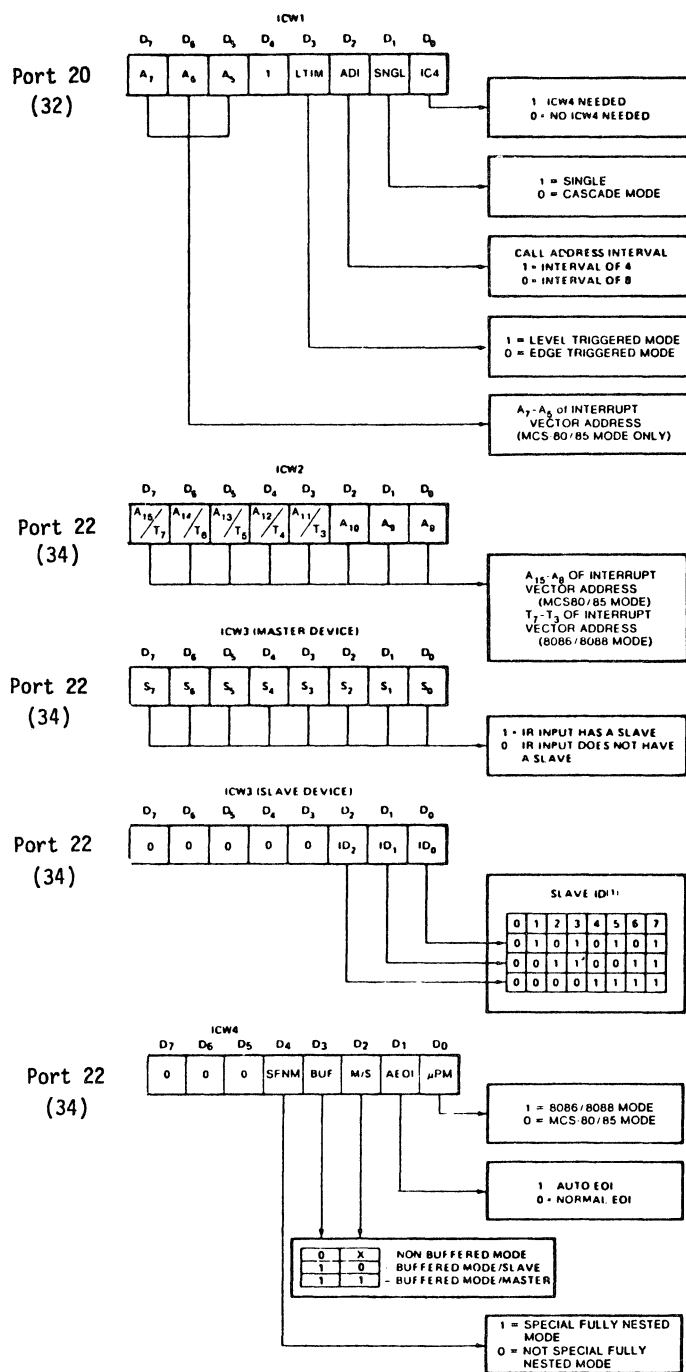


Figure 2-20. 8259A Initialization Sequence



NOTE 1: SLAVE ID IS EQUAL TO THE CORRESPONDING MASTER IR INPUT

Figure 2-21. 8259A Initialization Command Words

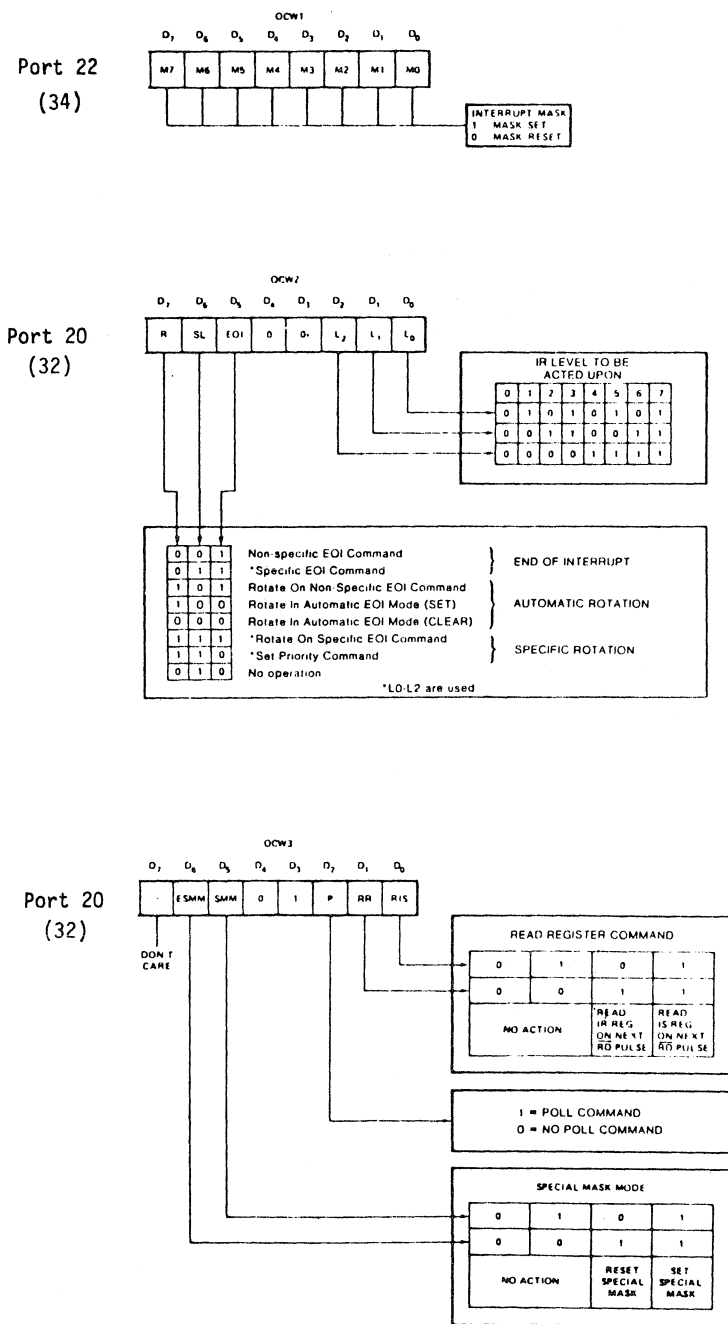


Figure 2-22. 8259A Operation Command Words

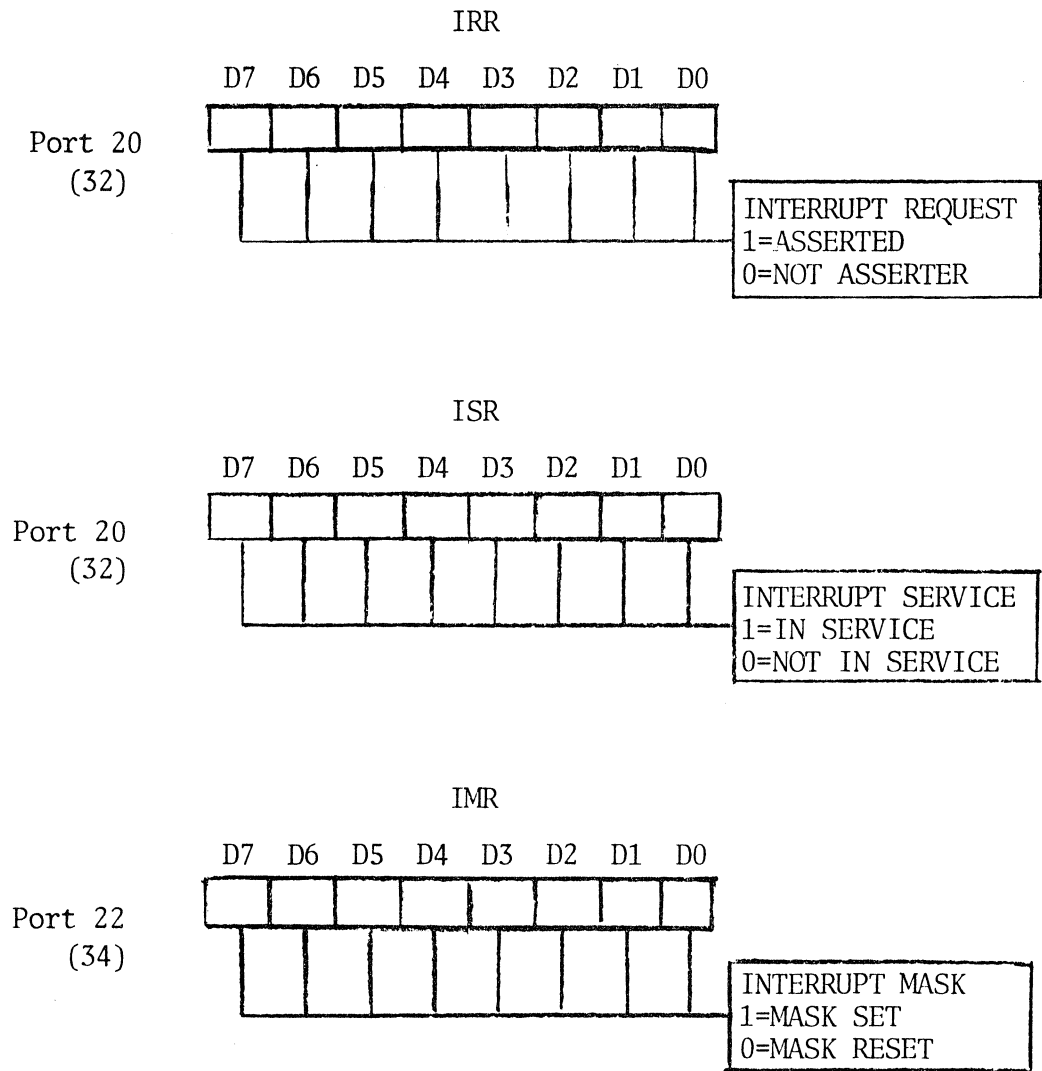


Figure 2-23. 8259A Status Registers

	Inst. #	Mnemonic	D7	D6	D5	D4	D3	D2	D1	D0	Operation Description	
Port 20	1	ICW1 A	A7	A6	A5	1	0	1	1	0	Byte 1 Initialization	Format = 4, single, edge triggered
	2	ICW1 B	A7	A6	A5	1	1	1	1	0		Format = 4, single, level triggered
	3	ICW1 C	A7	A6	A5	1	0	1	0	0		Format = 4, not single, edge triggered
	4	ICW1 D	A7	A6	A5	1	1	1	0	0		Format = 4, not single, level triggered
	5	ICW1 E	A7	A6	0	1	0	0	1	0	No ICW4 Required	Format = 8, single, edge triggered
	6	ICW1 F	A7	A6	0	1	1	0	1	0		Format = 8, single, level triggered
	7	ICW1 G	A7	A6	0	1	0	0	0	0		Format = 8, not single, edge triggered
	8	ICW1 H	A7	A6	0	1	1	0	0	0		Format = 8, not single, level triggered
	9	ICW1 I	A7	A6	A5	1	0	1	1	1	Byte 1 Initialization	Format = 4, single, edge triggered
	10	ICW1 J	A7	A6	A5	1	1	1	1	1		Format = 4, single, level triggered
	11	ICW1 K	A7	A6	A5	1	0	1	0	1		Format = 4, not single, edge triggered
	12	ICW1 L	A7	A6	A5	1	1	1	0	1		Format = 4, not single, level triggered
	13	ICW1 M	A7	A6	0	1	0	0	1	1	ICW4 Required	Format = 8, single, edge triggered
	14	ICW1 N	A7	A6	0	1	1	0	1	1		Format = 8, single, level triggered
	15	ICW1 O	A7	A6	0	1	0	0	0	1		Format = 8, not single, edge triggered
	16	ICW1 P	A7	A6	0	1	1	0	0	1		Format = 8, not single, level triggered
Port 22	17	ICW2	A15	A14	A13	A12	A11	A10	A9	A8	Byte 2 initialization	
	18	ICW3 M	S7	S6	S5	S4	S3	S2	S1	S0	Byte 3 initialization — master	
	19	ICW3 S	0	0	0	0	0	S2	S1	S0	Byte 3 initialization — slave	
	20	ICW4 A	0	0	0	0	0	0	0	0	No action, redundant	
	21	ICW4 B	0	0	0	0	0	0	0	1	Non-buffered mode, no AEOL, 8086/8088	
	22	ICW4 C	0	0	0	0	0	0	1	0	Non-buffered mode, AEOL, MCS-80/85	
	23	ICW4 D	0	0	0	0	0	0	1	1	Non-buffered mode, AEOL, 8086/8088	
	24	ICW4 E	0	0	0	0	0	1	0	0	No action, redundant	
	25	ICW4 F	0	0	0	0	0	1	0	1	Non-buffered mode, no AEOL, 8086/8088	
	26	ICW4 G	0	0	0	0	0	1	1	0	Non-buffered mode, AEOL, MCS-80/85	
	27	ICW4 H	0	0	0	0	0	1	1	1	Non-buffered mode, AEOL, 8086/8088	
	28	ICW4 I	0	0	0	0	1	0	0	0	Buffered mode, slave, no AEOL, MCS-80/85	
	29	ICW4 J	0	0	0	0	1	0	0	1	Buffered mode, slave, no AEOL, 8086/8088	
	30	ICW4 K	0	0	0	0	1	0	1	0	Buffered mode, slave, AEOL, MCS-80/85	
	31	ICW4 L	0	0	0	0	1	0	1	1	Buffered mode, slave, AEOL, 8086/8088	
	32	ICW4 M	0	0	0	0	1	1	0	0	Buffered mode, master, no AEOL, MCS-80/85	
	33	ICW4 N	0	0	0	0	1	1	0	1	Buffered mode, master, no AEOL, 8086/8088	
	34	ICW4 O	0	0	0	0	1	1	1	0	Buffered mode, master, AEOL, MCS-80/85	
	35	ICW4 P	0	0	0	0	1	1	1	1	Buffered mode, master AEOL, 8086, 8088	
	36	ICW4 NA	0	0	0	1	0	0	0	0	Fully nested mode, MCS-80, non buffered, no AEOL	
	37	ICW4 NB	0	0	0	1	0	0	0	1	ICW4 NB through ICW4 ND are identical to ICW4 B through ICW4 D with the addition of Fully Nested Mode	
	38	ICW4 NC	0	0	0	1	0	0	1	0		
	39	ICW4 ND	0	0	0	1	0	0	1	1		
	40	ICW4 NE	0	0	0	1	0	1	0	0		
	41	ICW4 NF	0	0	0	1	0	1	0	1	ICW4 NF through ICW4 NP are identical to ICW4 F through ICW4 P with the addition of Fully Nested Mode	
	42	ICW4 NG	0	0	0	1	0	1	1	0		
	43	ICW4 NH	0	0	0	1	0	1	1	1		
	44	ICW4 NI	0	0	0	1	1	0	0	0		
	45	ICW4 NJ	0	0	0	1	1	0	0	1		
	46	ICW4 NK	0	0	0	1	1	0	1	0		
	47	ICW4 NL	0	0	0	1	1	0	1	1		
	48	ICW4 NM	0	0	0	1	1	1	0	0		
	49	ICW4 NN	0	0	0	1	1	1	0	1		
	50	ICW4 NO	0	0	0	1	1	1	1	0		
Port 20	51	ICW4 NP	0	0	0	1	1	1	1	1		
	52	OCW1	M7	M6	M5	M4	M3	M2	M1	M0	Load mask register, read mask register	
	53	OCW2 E	0	0	1	0	0	0	0	0	Non-specific EOI	
	54	OCW2 SE	0	1	1	0	0	L2	L1	L0	Specific EOI, L0-L2 code of IS FF to be reset	
	55	OCW2 RE	1	0	1	0	0	0	0	0	Rotate on Non-Specific EOI	
	56	OCW2 RSE	1	1	1	0	0	L2	L1	L0	Rotate on Specific EOI L0-L2 code of line	
	57	OCW2 R	1	0	0	0	0	0	0	0	Rotate in Auto EOI (set)	
	58	OCW2 CR	0	0	0	0	0	0	0	0	Rotate in Auto EOI (clear)	
	59	OCW2 RS	1	1	0	0	0	L2	L1	L0	Set Priority Command	
	60	OCW3 P	0	0	0	0	1	1	0	0	Poll mode	
	61	OCW3 RIS	0	0	0	0	1	0	1	1	Read IS register	

Figure 2-24. 8259A Instruction Set Summary

I/O Control Register (IOCR)

The IOCR, port 56 (86 decimal), is a 16-bit register, used to control and report certain mainframe events. It ties into several other logic blocks on the mainframe. Figure 2-25 illustrates the bit pattern.

The high-order byte consists of readable and writable system control bits. Writing a zero into any of these bits disables the logic which generates the corresponding interrupt.

The low-order byte is read-only status flags. Interrupt flags persist until the interrupt condition is removed and the corresponding control bit in the high byte of the IOCR has been set by software. The bit assignments are:

<u>Bit</u>	<u>Type</u>	<u>Meaning</u>
F	Control	Power-Fail Detect Interrupt Enable. When one, enables power fail detect non-maskable interrupt. When zero, clears existing power fail indications.
E	Control	Parity-Error Detect Enable. When one, enables parity error detect and parity error interrupts. When zero clears existing parity error interrupts. When zero clears existing parity error indications and disables detection and interrupts.
D	Control	Odd Parity Enable. When one, the parity logic writes odd parity and checks odd parity. When zero even parity is used.
C	Control	Multibus I/O Enable. Normally this bit has no effect. If factory jumper 2-3 on the CPU board is installed, and IOCR bit C is one, then any I/O instruction with address bit F=0 refers to a Multibus I/O device, and if address bit F=1 or IOCR bit C is 0, the instruction refers to a CT I/O port.

Normally jumper 2-3 is not installed and reference to Multibus I/O devices is done by any I/O instruction with address bit F=1 (IOCR bit C has no effect).

B	Control	Printer Interrupt Enable. When one, enables interrupts from the line printer interface. When zero, clears existing
---	---------	--

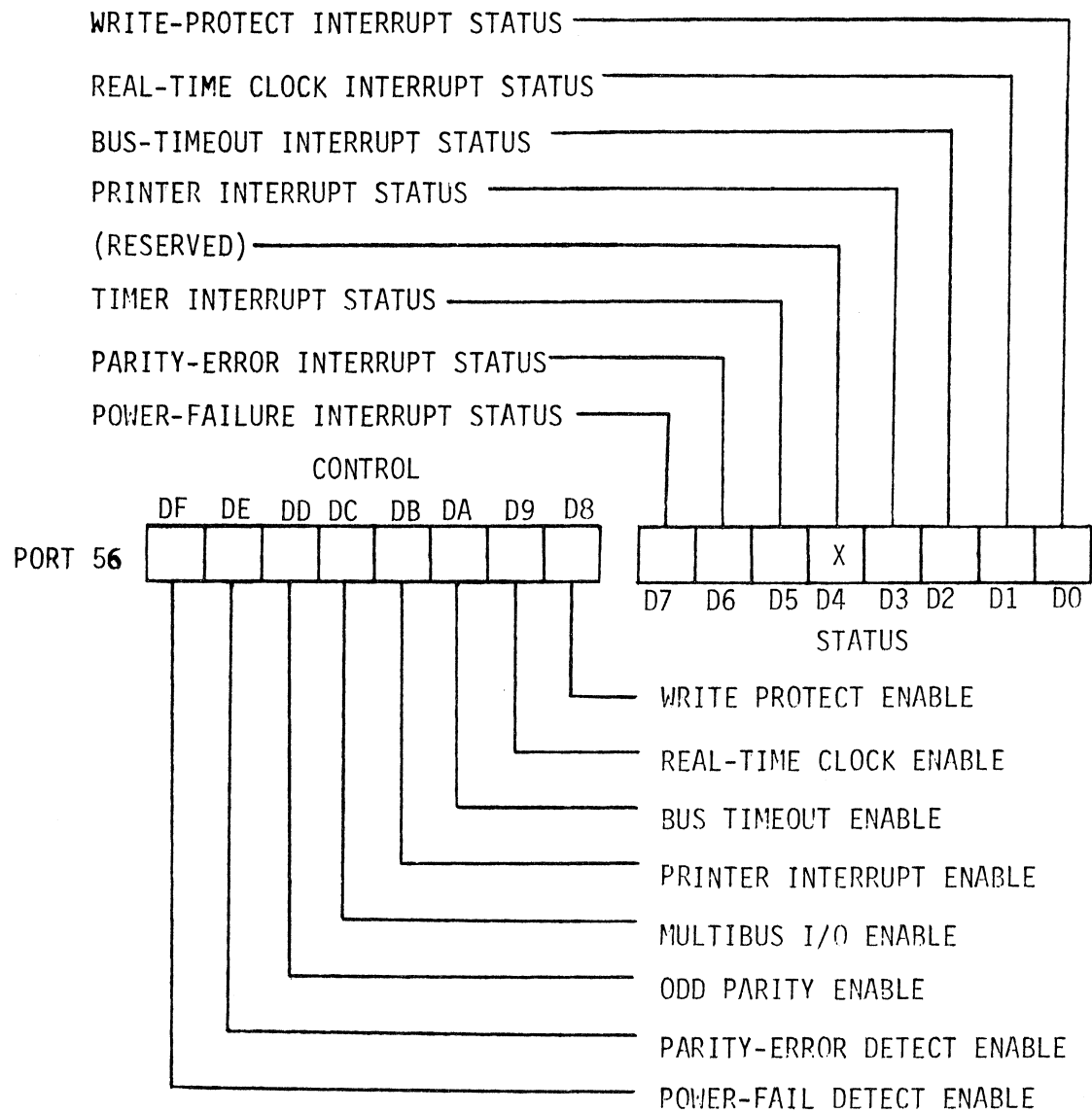


Figure 2-25. I/O Control Register (IOCR)

interrupts and disables generating new interrupts.

- | | | |
|---|---------|--|
| A | Control | Bus Timeout (Non-Existent Device Detect) Enable. When one, a reference to a non-existent I/O register or a non-existent memory location will generate a non-maskable interrupt. When zero, the interrupt will be cleared and the detection will be disabled. |
| 9 | Control | Realtime Clock Enable. When one, realtime-clock interrupts are enabled. When zero, pending realtime-clock interrupts are cleared and new ones are disabled. |
| 8 | Control | Write Protect Enable. When one, the lower 32K bytes of memory cannot be written into. If an attempt is made to write it will cause a non-maskable-interrupt. When zero these functions are disabled and any pending write protect interrupt is cleared. This does not apply to DMA writes. |
| 7 | Status | Power-Fail Interrupts. When one, indicates that a power-fail non-maskable interrupt is pending. |
| 6 | Status | Parity-Error Interrupt. When one, indicates that a parity error non-maskable interrupt is pending. |
| 5 | Status | Timer Interrupt. When one, indicates that the programmable timer (counter 0) is interrupting (level 3 maskable interrupt). |
| 4 | Status | Reserved. |
| 3 | Status | Printer Interrupt. When one, indicates a line printer interrupt is pending (level 4 maskable interrupt). |

- 2 Status Bus Timeout (Non-existent Device) Interrupt. When one, indicates a non-existent I/O device or memory location was referenced and did not respond within 6 milliseconds. This will cause a non-maskable interrupt.
- 1 Status Realtime Clock Interrupt. When one, indicates a realtime clock interrupt is pending (level 4 maskable interrupt).
- 0 Status Write Protect Interrupt. When one, indicates a write protect fault occurred and a non-maskable interrupt was caused by it.

The IOCR control bits are set to zero whenever the system is reset, either manually or during power-up. Normally, the operating system sets one or more of these bits during initialization. When set, the following functions will generate interrupts (NMI = non-maskable interrupt, INTR4 = level 4 maskable interrupt):

<u>Function</u>	<u>Control</u>	<u>Status</u>	<u>Int.Type</u>
Power Failure	F	7	NMI
Parity Error	E	6	NMI
Timer	-	5	INTR3
Printer	B	3	INTR4
Bus Timeout	A	2	NMI
Real-Time Clock	9	1	INTR4
Write Protect	8	0	NMI

8253 Interval Timer

The 8253 timer has three independent channels, designated 0, 1 and 2. Two of these channels (1 and 2) are used for baud-rate generation for the Z80A-SIO communications controller. The third (channel 0) is available to the user and can be programmed to generate a maskable interrupt to the CPU.

The I/O ports assigned to the 8253 are as follows:

<u>Port</u>	<u>Write</u>	<u>Read</u>
28 (40)	Counter 0 divisor	Counter 0 divisor
2A (42)	Counter 1 divisor	Counter 1 divisor
2C (44)	Counter 2 divisor	Counter 2 divisor
2E (46)	Mode (all counters)	Not used

When the standard operating system is used, all initialization and control of counters is done by the operating system.

Note that the Gate input to all counters is permanently enabled (high) and cannot be changed. This limits the use of the channels to certain modes, as described below.

Write Operations: The systems software must program each counter of the 8253 with the mode and hexadecimal count divisor desired. The programmer must write out to the 8253 a MODE control word and the programmed number of count register bytes (1 or 2) prior to actually using the selected counter.

The actual order of the programming is quite flexible. Writing out of the MODE control word can be in any sequence of counter selection, e.g., counter #0 does not have to be first or counter #2 last. Each counter's MODE control word register has a separate address so that its loading is completely sequence independent. (SC0, SC1)

The loading of the Count Register with the actual hexadecimal count divisor, however, must be done in exactly the sequence programmed in the MODE control word (RL0, RL1). This loading of the

counter's count register is still sequence independent like the MODE control word loading, but when a selected count register is to be loaded it must be loaded with the number of bytes programmed in the MODE control word (RL0, RL1). The one or two bytes to be loaded in the count register do not have to follow the associated MODE control word. They can be programmed at any time following the MODE control word loading as long as the correct number of bytes is loaded in order.

All counters are down counters. Thus, the value loaded into the count register will be decremented. Loading all zeroes into a count register will result in the maximum count. In MODE 0 the new count will not restart until the load has been completed. It will accept one of two bytes, depending on how the MODE control words (RL0, RL1) are programmed, then proceed with the restart operation.

Counter 0 has a 19.53KHz clock input.

The baud rate (counters 1 & 2) counters each have a 1.23-MHz clock input. The 16-bit divisor number put into the counter is the number by which 1.23 MHz is divided before going to the SIO. Some common divisors are:

<u>Hex</u> <u>Divisor</u>	<u>SIO Divide</u> <u>Factor (decimal)</u>	<u>Baud</u> <u>Rate</u>
2	1	614,400
4	16	19,200
8	16	9,600
10	16	4,800
20	16	2,400
40	16	1,200
80	16	600
100	16	300
200	16	150
2BA	16	110

Mode Definition: The standard initialization mode for the two communication channels is Mode 3 (Square Wave Rate Generator). When the standard operating system is used, channel 0 of the 8253 can be programmed only in Mode 0, since the operating system initializes the 8259A to expect only level-triggered interrupts and all other modes of the 8253 except mode 0 require an edge-triggered interrupt system.

Mode 0 (Interrupt on Terminal Count): Programming the mode of counter 0 causes any pending interrupt from the counter to be removed. As soon as the counter bytes are loaded, the counter begins to decrement at the 19.53KHz rate. When the counter reaches 0, it will cause a level 4 interrupt.

After the terminal count has been reached, the counter continues to decrement (0000, FFFF, FFFE, ...). Therefore, it is possible to ascertain the software latency time of the interrupt service process by reading the counter at the end of the process.

MODE 3 (Square Wave Rate Generator): With the standard operating system, this mode is used for counters 1 and 2, those that generate baud rate for the SIO channels. It is similar to MODE 2 except that the output will remain high until one half the count has been completed (for even numbers) and go low for the other half of the count. This is accomplished by decrementing the counter by two on the falling edge of each clock pulse. When the counter reaches terminal count, the state of the output is changed and the counter is reloaded with the full count and the whole process is repeated. If the count is odd and the output is high, the first clock pulse (after the count is loaded) decrements the count by 1. Subsequent clock pulses decrement the clock by 2. After timeout, the output goes low and the full count is reloaded. The first clock pulse (following the reload) decrements the counter by 3. Subsequent clock pulses decrement the count by 2 until timeout. Then the whole process is repeated. In this way, if the count is odd, the output will be high for $(N + 1)/2$ counts and low for $(N - 1)/2$ counts.

Reading from the 8253: Writing the control byte 0+X (ie., high nibble must be 0 but low nibble can be anything) to port 2E (46) sets the timer up for a read operation, in which the least and most significant bytes of the timer's count register are successively placed onto the local I/O bus in response to successive reads. To convert this count into the time remaining before the forthcoming interrupt on the TMRINT+ signal line,

multiply by 5.12 microseconds. Note that the count parameter may be in either hex or BCD format depending on whether the last digit of the mode control word used for loading the count valve was 1 or 0, respectively.

It is imperative to complete any read operation which has been started before trying to initiate any further write operations to the 8253. Note that the 8253 timer is not affected by ongoing read operations initiated by control bytes of the form 0+X. Therefore, the 8253 can generate an interrupt while a read operation is underway, and the interrupt cannot be cleared until both bytes have been read out.

The format for writing control information to the 8253 is illustrated in Figure 2-26.

Control Word Format

D7 D6 D5 D4 D3 D2 D1 D0							
Port 2E (47)							
SC1	SC0	RL1	RL0	M2	M1	M0	BCD
SC1 SC0							
0	0	Select Counter 0					
0	1	Select Counter 1					
1	0	Select Counter 2					
1	1	Illegal					
RL1 RL0							
0	0	Counter Latching operation*					
1	0	Read/Load most significant byte only.					
0	1	Read/Load least significant byte only.					
1	1	Read/Load least significant byte first, then most significant byte.					
* See "Reading While Counting". Bits D0 through D3 are don't care if this mode is used.							
M2	M1	M0					
0	0	0 Mode 0					
0	0	1 Mode 1					
X	1	0 Mode 2					
X	1	1 Mode 3					
1	0	0 Mode 4					
1	0	1 Mode 5					
BCD							
		0 Binary Counter 16-bits					
		1 Binary Coded Decimal (BCD) Counter (4 Decades)					

Figure 2-26. 8253 Control Word

MULTIBUS INTERFACE

The electrical interface between the local bus and the Multibus slots is termed the Multibus interface, as shown in Figure 2-1. This interface isolates the CT mainframe from the Multibus. The interface is controlled by various logic blocks, switch settings and jumpers on the CT mainframe. It provides asynchronous separation; the mainframe and Multibus clocks are assumed to have no interaction.

Unlike other implementations of Multibus interfaces, this interface allows simultaneous operation (multiprocessing) of Multibus and local bus masters whenever there are no transfers across the interface.

The IEEE 766 "Multibus" Standard (Appendix A) defines several levels of hardware compliance. The CT version of the Multibus achieves the highest level of compliance with the exception of memory address bits. CT provides 20 bits instead of 24 bits. In terms of the IEEE 796 short hand notation the CT bus achieves a D16 M20 I16 VD23E level of compliance

Signal Lines

The mainframe motherboard contains either two or four slots for Multibus boards, depending on the hardware configuration. These slots are standard Multibus slots with all signals required in the IEEE 796 microcomputer bus standard, as shown in Figure 2-27.

Form Factor

The physical form requirements for user-supplied Multibus boards is shown in Figure 2-28. Maximum height of the Multibus boards, a parameter that is not specified by the IEEE standard, is 6.75 inches from the base of the edge connector.

Transfers Across The Interface

When transfers across the Multibus interface are controlled by the local bus (i.e., the local CPU), either memory or I/O slave resources on the Multibus can be accessed. When a Multibus master

(e.g., a CPU or DMA on the Multibus) controls the transfers, the only slave resources on the local bus that can be accessed is the memory; I/O on the local bus cannot be accessed by a Multibus master.

The memory address space accessible in transfers across the Multibus Interface is illustrated in

Table 4 Pin Assignment of Bus Signals on Multibus Board Connector (P1)

	PIN	(COMPONENT SIDE)		PIN	(CIRCUIT SIDE)	
		MNEMONIC	DESCRIPTION		MNEMONIC	DESCRIPTION
POWER SUPPLIES	1	GND	Signal GND	2	GND	Sig GND
	3	+5V	+5Vdc	4	+5V	+5Vdc
	5	+5V	+5Vdc	6	+5V	+5Vdc
	7	+12V	+12Vdc	8	+12V	+12Vdc
	9		RESERVED	10		RESERVED
	11	GND	Signal GND	12	GND	Signal GND
BUS CONTROLS	13	BCLK/	Bus Clock	14	INIT/	Initialize
	15	BPRN/	Bus Pri. In	16	BPRO/	Bus Pri. Out
	17	BUSY/	Bus Busy	18	BREQ/	Bus Request
	19	MRDC/	Mem Read Cmd	20	MWTC/	Mem Write Cmd
	21	IORC/	I/O Read Cmd	22	IOWC/	I/O Write Cmd
	23	XACK/	XFER Acknowledge	24	INH1/	Inhibit 1 disable RAM
BUS CONTROLS AND ADDRESS	25		Reserved	26	INH2/	Inhibit 2 disable PROM or ROM
	27	BHEN/	Byte High Enable	28	AD10/	Address Bus
	29	CBRO/	Common Bus Request	30	AD11/	
	31	CCLK/	Constant Clk	32	AD12/	
	33	INTA/	Intr Acknowledge	34	AD13/	
INTERRUPTS	35	INT6/	Parallel Interrupt Requests	36	INT7/	Parallel Interrupt Requests
	37	INT4/		38	INT5/	
	39	INT2/		40	INT3/	
	41	INT0/		42	INT1/	
ADDRESS	43	ADRE/	Address Bus	44	ADRF/	Address Bus
	45	ADRC/		46	ADRD/	
	47	ADRA/		48	ADRB	
	49	ADR8/		50	ADR9/	
	51	ADR6/		52	ADR7/	
	53	ADR4/		54	ADR5/	
	55	ADR2/		56	ADR3/	
	57	ADR0/		58	ADR1/	
DATA	59	DATF/	Data Bus	60	DATF/	Data Bus
	61	DATC/		62	DATD/	
	63	DATB/		64	DATB/	
	65	DAT8/		66	DAT9/	
	67	DAT6/		68	DAT7/	
	69	DAT4/		70	DAT5/	
	71	DAT2/		72	DAT3/	
	73	DAT0/		74	DAT1/	
POWER SUPPLIES	75	GND	Signal GND	76	GND	Signal GND
	77		Reserved	78		Reserved
	79	-12V	-12Vdc	80	-12V	-12Vdc
	81	+5V	+5Vdc	82	+5V	+5Vdc
	83	+5V	+5Vdc	84	+5V	+5Vdc
	85	GND	Signal GND	86	GND	Signal GND

Figure 2-27. Multibus Pin Assignments

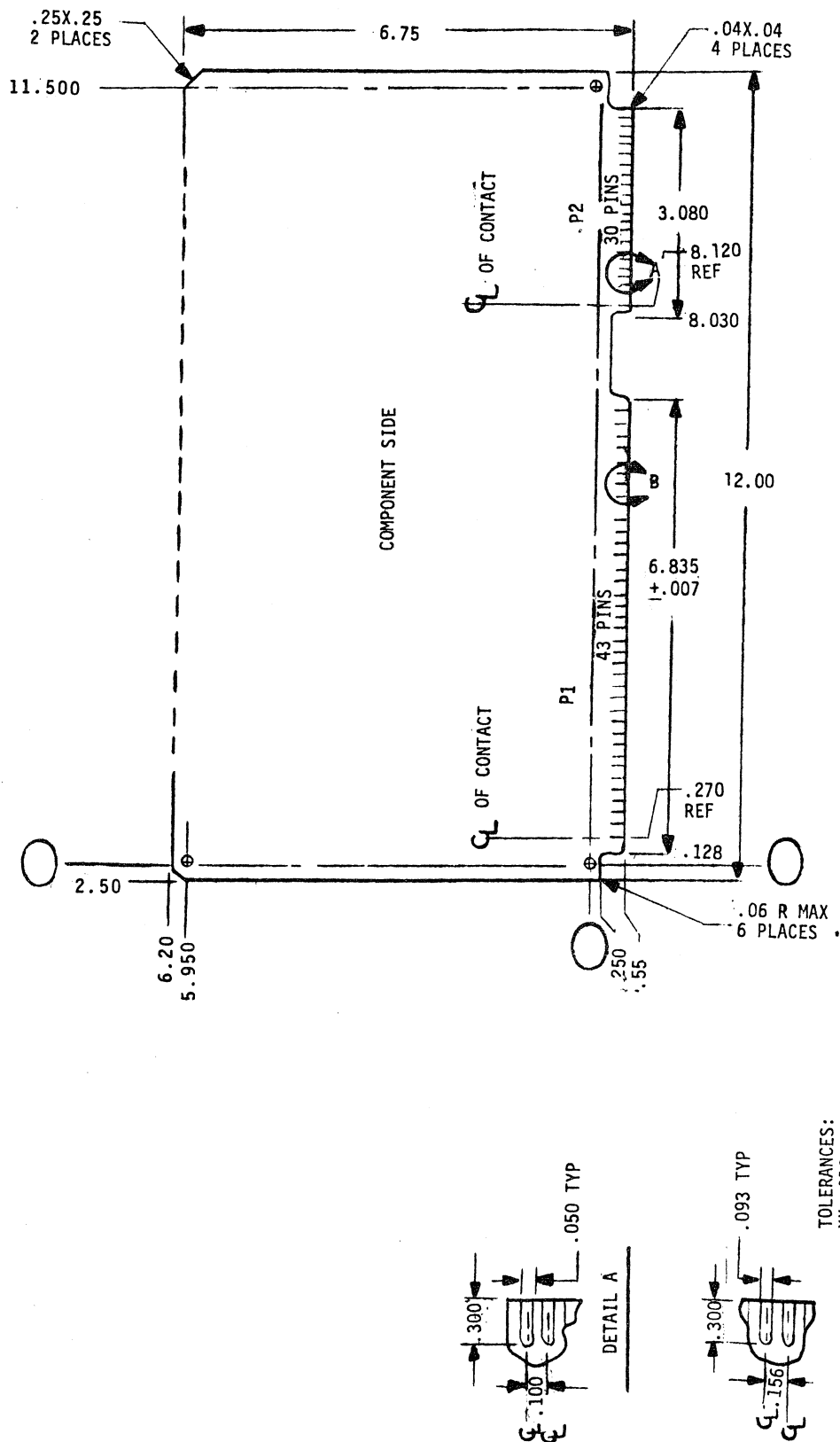


Figure 2-28. Multibus Form Factor

Figure 1-4 and the "Memory Interface" section of this chapter contains a subsection entitled

"Multibus/Local-Bus Memory Mapping" that describes the transfer windows in more detail. In addition, the "DMA transfer" section contains a subsection entitled "Multibus Access" that describes the method by which Multibus masters obtain access to the local memory.

Modes of Operation

The IEEE specifications specifies that two alternative modes of operation (Mode 1 and Mode 2) for Multibus masters be accommodated, although a particular system should operate only in one mode or the other.

The CT system normally operates in the more permissive Mode 2 and it will accommodate any Multibus boards which also operate in this mode.

If a Mode 1 Multibus board is used, however, the entire system will then operate in Mode 1.

In Mode 1, bus masters are limited to single bus transfers per bus connect and system timing is rendered deterministic by compliance with the maximum bus busy period, t_{BYSO} max. In Mode 2, bus masters are not limited in their bus control and compliance with a maximum busy period is not required.

Bus Priority Resolution

A parallel priority resolution scheme is used for arbitrating bus access by Multibus masters. Two sockets on the mainframe motherboard contain a priority encoder chip and a decoder chip to implement this priority resolution scheme.

The parallel arrangement requires that the BREQ/lines of each possible Multibus master (the mainframe plus any user-supplied Multibus boards) be input to the encoder.

The priority of access is fixed as follows, from highest to lowest:

BREQ0	Local Bus
BREQ1	Multibus Slot 1
BREQ2	Multibus Slot 2

Note that BPRO lines must be disabled on Multibus boards using parallel priority.

Serial priority resolution schemes are also supported. These are implemented by removing the parallel-priority chips and enabling BPRO lines on all Multibus boards.

Interrupts

Multibus interrupts adhere to the IEEE standard. However, since ambiguities of interpretation can arise, the general rules with respect to Multibus interrupts can be summarized as follows:

- o Non-maskable interrupts (NMI) from the Multibus to the local bus are not supported.
- o Maskable interrupt requests from the Multibus to the local bus are supported for levels 0, 2, 5 and 6.
- o Maskable interrupt requests between Multibus boards are fully supported at all levels.

The second and third items above imply that maskable interrupt levels 0, 2, 5 and 6 can be serviced either by the local 8086 or by a Multibus master.

The local 8259A can be cascaded with up to three slave 8259As on the Multibus (level 0 cannot be cascaded) to service up to 29 prioritized levels of maskable interrupt within a system. Each slave 8259A can multiplex eight interrupt levels from the Multibus, and there are four available inputs from the Multibus to the local 8259A, giving 25 interrupt levels. The remaining four maskable interrupt levels are for the local functions mentioned above.

Figure 2-29 illustrates a possible configuration. The local 8259A master is shown with three slaves on the Multibus. When an interrupt occurs at a slave 8259A, it is passed on to the CPU through the hardware chain of interrupt requesting. During the first cycle of the interrupt acknowledge, the local 8259A addresses the interrupting slave through the three cascade lines

(CAS0-CAS2) which are attached via the processor bus and local address bus to the Multibus address bus. This tells the interrupting controller that it has precedence and that it should enable the interrupt vector from the interrupting device to be put on the data bus. The Buffer Enable (EN-) line of the local 8259A, which is operated in the Buffered Mode, then enables the proper buffer to allow the 8086 to see the incoming interrupt vector.

This configuration is dealt with in Application Note 59 of the Intel 8086 Family User's Manual. One note of caution: the Special Fully Nested Mode of the 8259A is not supported by CT.

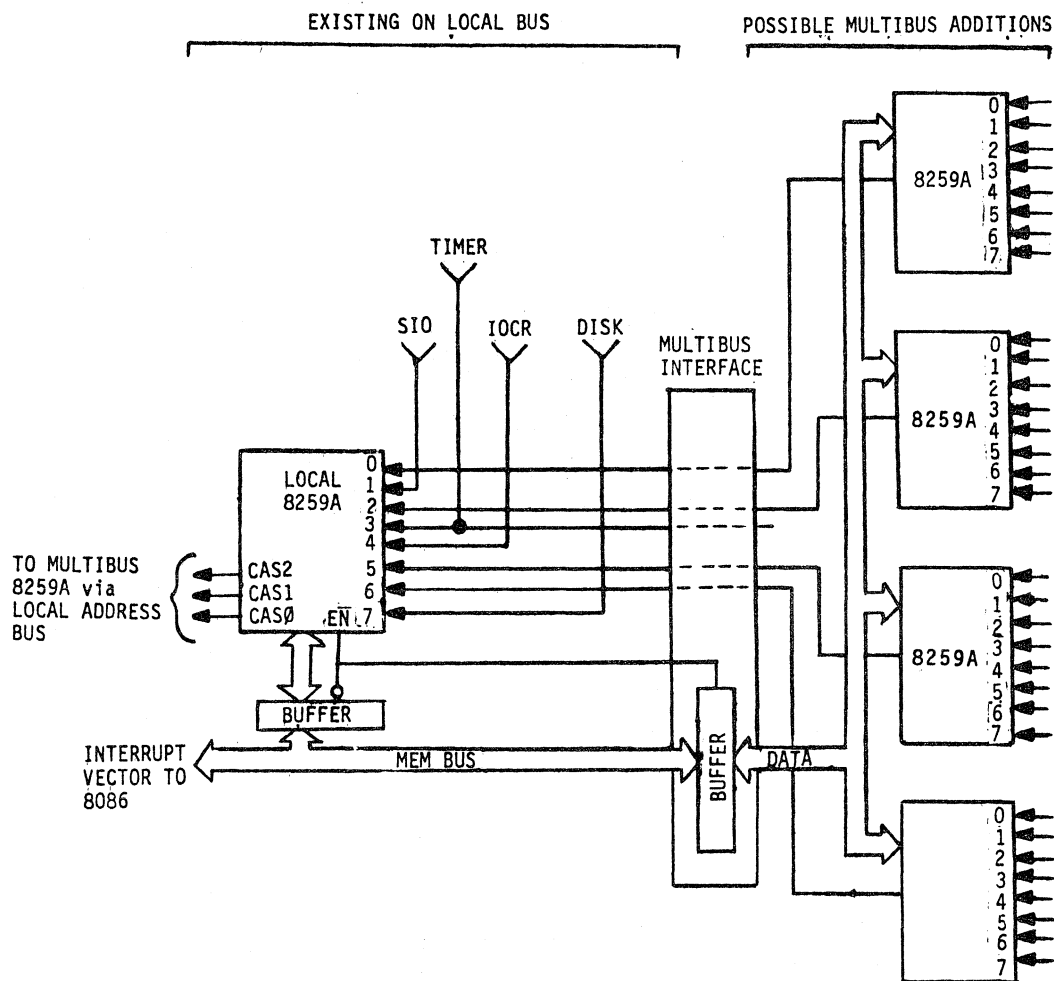


Figure 2-29. Maskable Interrupt Expansion on Multibus

COMMUNICATIONS INTERFACE

Channels

This interface, built around the Z80A-SIO, provides two independent communication channels, channel A and B, either of which can be operated in synchronous or asynchronous protocols. Both channels are brought to standard 25-pin D-type female connectors on the back of the mainframe enclosure. The channels are distinguished as shown in Figure 2-30.

Channel A has the greatest flexibility. It can operate either as an RS-232 or an RS-422 interface and can be controlled by either the CPU (interrupt or polled mode) or the local DMA logic. If DMA transfers are used, the communication channel can operate only in half duplex mode since only one DMA channel is allocated to communication and the SIO has only one Ready line per channel for use as a DMA request signal.

When the RS-422 interface option is selected by jumpers on the I/O-memory board, speeds up to 700 kilobits per second can be used. This balanced-differential interface allows multidrop hardware with the ability to transmit clocks on signals that are separate from the data.

Figure 2-31 illustrates the RS-232 and RS-422 pinouts.

Transfer Control

If the SIO is controlled by the CPU, it typically operates in interrupt-driven mode. In this mode, the SIO interrupts via level-1 maskable interrupts for each receive or transmit character as well as for other modem-control or anomolous conditions. As an alternative to the interrupt-driven mode, the CPU can operate the SIO in a polled mode. In either mode, CPU transfer control allows full-duplex channels.

If Channel A is used in the DMA mode, interrupts to the CPU are programmed in the SIO to occur only for modem-control and anomolous conditions. The SIO's Ready output controls DMA service requests for individual transfers. Since all DMA transfers are word transfers while the

	<u>channel A</u>	<u>channel B</u>
protocols	synchronous or asynchronous	synchronous or asynchronous
interface type	RS-232 or RS-422*	RS-232
transfer control	CPU or DMA	CPU
duplex mode	full or half**	full
max. speed	19.2 kbaud or 615 kbaud***	19.2 kbaud

* Balanced differential
 ** When DMA control is used
 *** When RS-422 is used

Figure 2-30. Communication Channel Characteristics

RS-232-C

<u>Pin</u>	<u>Function</u>
1	Protective Ground
2	Transmit Data
3	Receive Data
4	Request To Send
5	Clear To Send
6	Data Set Ready
7	Signal Ground
8	Carrier Detect
14	Secondary Transmit Data
15	Transmit Clock
16	Secondary Receive Data
17	Receive Clock
20	Data Terminal Ready
22	Ring

RS-422-A

<u>Pin</u>	<u>Function</u>
1	Protective Ground
2	Transmit Data + (from CT)
3	Transmit Clock + (to CT)
4	Transmit Data - (from CT)
5	Transmit Clock - (to CT)
6	Clear To Send + (to CT)
7	Signal Ground
8	Receive Data + (to CT)
14	Request to Send - (from CT)
15	Receive Clock - (to CT)
16	Clear To Send - (to CT)
17	Receive Clock + (to CT)
18	Transmit Clock + (from CT)
20	Request To Send + (from CT)
22	Receive Data - (to CT)
25	Transmit Clock - (from CT)

Figure 2-31. RS-232 and RS-422 Pinouts

communication interface sees only the lower byte of the 16-bit local data bus, all of the high-byte locations in the communications buffer of memory are ignored. Furthermore, when DMA is used for control of Channel A the communications buffer must reside in the lower 128-kilobytes of memory, as illustrated in Figure 2-13.

The "DMA transfers" and "Interrupts" sections of this chapter contain additional information on communications control.

Switch-Selectable Clocks

Switches on the I/O-memory board can be used to select whether internal or external clocking is to be used on the data receive and transmit lines. Either channel can have either the receive or transmit line clocked either internally or externally.

Extended Control Register

A 16-bit register associated with the SIO is used to augment the SIO's set of channel control lines and status information. Among its functions, which enhance both RS-232 and RS-422 communication, are:

- Channel A ring indicator
- Channel A data set ready (DSR)
- Channel A secondary Rx data
- Channel A secondary Tx data
- Channel B ring indicator
- Channel B data set ready (DSR)
- Channel B secondary Rx data
- Channel B secondary Tx data
- RS-422 transmit request
- RS-422 transmitter enabled
- RS-422 carrier detect enabled

Software Interface

Z80A Serial I/O (SIO) Controller

The dual-channel Z80A-SIO/2 controls communication on the two serial communication channels. One of the channels, Channel A, can be operated in a DMA or CPU-driven mode; this channel can be jumper-selected as either an RS-232-C or an RS-422-A interface. Channel B can operate only in a CPU-driven mode as an RS-232-C port.

Due to the SIO's non-8086 bus structure, interrupt acknowledges are sent by the CPU through a separate logic block called the Interrupt Acknowledge Register (IAR). Also, an Extended Control Register (ECR) adds some control functions that are not supported by the SIO. The programming of these registers is described in the two sections that follow this one. This section describes only the SIO itself.

I/O port assignments for the SIO channels and other related registers are as follows:

<u>Port</u>	<u>Write</u>	<u>Read</u>
48 (72)	SIO Ch.A data	SIO Ch.A data
4A (74)	SIO Ch.A command	SIO Ch.A status
4C (76)	SIO Ch.B data	SIO Ch.B data
4E (78)	SIO Ch.B command	SIO Ch. B status
30 (48)	not used	IAR*
60 (96)	ECR* ECR*	

* See separate sections with these titles.

In addition to the above ports, the following ports on the 8253 timer are used to program SIO baud rate divisors, as described in the section relating to the 8253:

<u>Port</u>	<u>Write</u>	<u>Read</u>
2A (42)	Ch.B divisor	Ch.B divisor
2C (44)	Ch.A divisor	Ch.A divisor
2E (46)	8253 Mode	Not used

Initialization Subroutines: The SIO is the only programmable IC in the mainframe that is not initialized automatically by the standard operating system. Instead, several preprogrammed subroutines are provided in the system library for initialization of the SIO to various modes

and speeds. These subroutines can be bound with any application programs.

General Procedures: The application program first issues a series of commands that initialize the basic mode of operation and then other commands that qualify conditions within the selected mode. For example, the asynchronous mode, character length, clock rate, number of stop bits, even or odd parity might be set first; then the interrupt mode; and finally, receiver or transmitter enable. Both channels contain registers, addressed as different I/O ports, that must be programmed prior to operation.

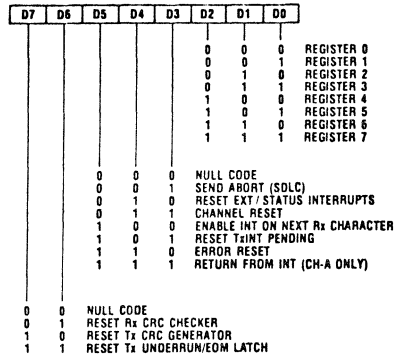
Write Registers: The SIO contains eight write registers for Channel B and seven write registers for Channel A (WRO-WR7) that are programmed separately to configure the functional personality of the channels; WR2 contains the interrupt vector for both channels and is only in the Channel B register set. With the exception of WRO, programming the write registers requires two bytes. The first byte is to WRO and contains three bits (D_0 - D_2) that point to the selected register; the second byte is the actual control word that is written into the register to configure the SIO. WRO is a special case in that all of the basic commands can be written to it with a single byte. Reset (internal or external) initializes the pointer bits D_0 - D_2 to point to WRO. This implies that a channel reset must not be combined with the pointing to any register. The write registers are shown in Figure 2-32.

Read Register: The SIO contains three read registers for Channel B and two read registers for Channel A (RRO-RR2) that can be read to obtain the status information; RR2 contains the internally-modifiable interrupt vector and is only in the Channel B register set. The status information includes error conditions, interrupt vector and standard communications-interface signals. To read the contents of a selected read register other than RRO, the system program must first write the pointer byte to WRO in exactly the same way as a write register operation. Then, by executing a read instruction, the contents of the ad-

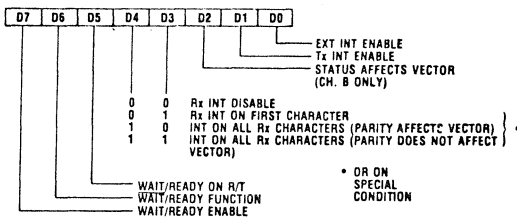
dressed read register can be read by the CPU. The status bits of RRO and RR1 are carefully grouped to simplify status monitoring. For example, when the interrupt vector indicates that a Special Receive Condition interrupt has occurred, all the appropriate error bits can be read from a single register (RR1). The read registers are shown in Figure 2-33.

DMA Operation: To program channel A of the SIO for DMA operation, select the READY function on the SIO's WAIT/READY pin (set bit 6 of WR1) and enable that function (set bit 7 of WR1).

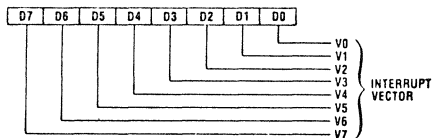
WRITE REGISTER 0



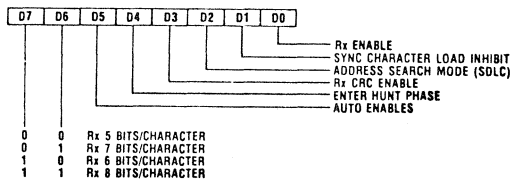
WRITE REGISTER 1



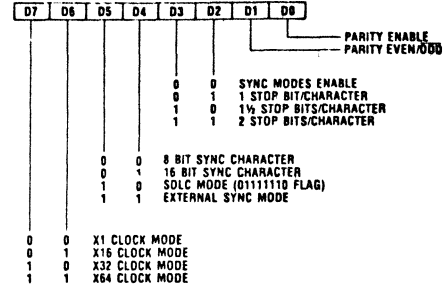
WRITE REGISTER 2 (CHANNEL B ONLY)



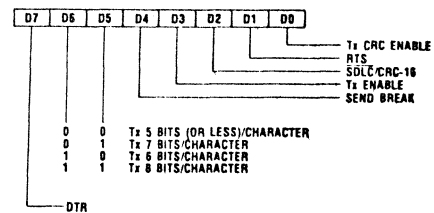
WRITE REGISTER 3



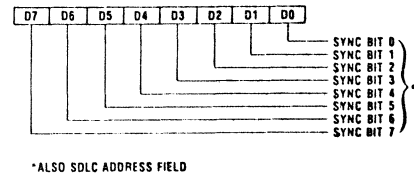
WRITE REGISTER 4



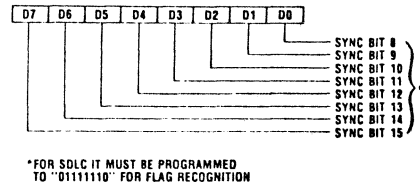
WRITE REGISTER 5



WRITE REGISTER 6



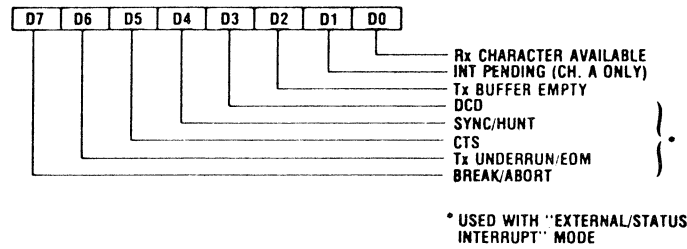
WRITE REGISTER 7



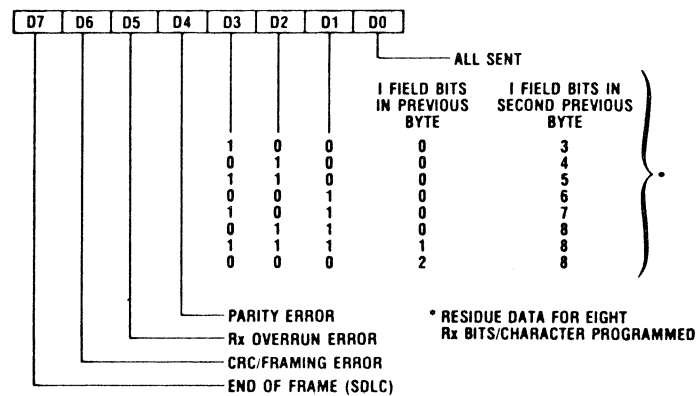
NOTE: A separate set of registers is maintained for channels A and B. Channel A is port 4A and channel B is port 4E.

Figure 2-32. SIO Write Registers

READ REGISTER 0

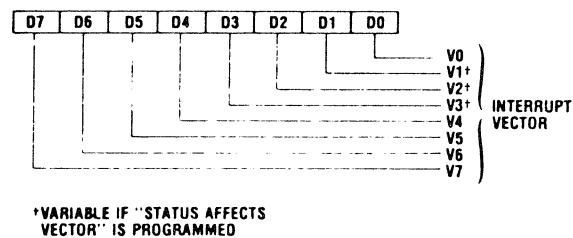


READ REGISTER 1†



† USED WITH SPECIAL RECEIVE CONDITION MODE

READ REGISTER 2



NOTE: A separate set of registers is maintained for channels A and B. Channel A is port 4A and channel B is port 4E.

Figure 2-33. SIO Read Registers

SIO Extended Control Register (ECR)

The ECR is a 16-bit control/status register which is written to and read from port 60 (96). It handles control functions that are not supported by the SIO but which are useful in RS-232 and RS-422 environments. Figure 2-34 illustrates the register.

The 16 bits of the register are allocated as follows:

<u>Bit</u>	<u>Type</u>	<u>Function</u>
F	Status	RS-422 Tx enable. This bit is the same as bit 2.
E	Status	Channel A ring indicator, asserted when 0.
D	Status	Channel B Data Set Ready, asserted when 0.
C	Status	Channel A Secondary Rx data, asserted when 1.
B	--	Not used
A	Status	Channel B ring indicator, asserted when 0.
9	Status	Channel B Data Set Ready, asserted when 0.
8	Status	Channel B Secondary Rx data, asserted when 1.
7	--	Not used
6	--	Not used
5	--	Not used
4	--	Not used
3	Control	Clock detect enable. If I/O memory board switch 3 bit 2 is open and bit 1 is closed, Carrier Detect for channel A comes from receive clock detect circuit. If IOCR bit 3 is one and the Channel A receive clock is active then Carrier Detect

will go active. Setting IOCR
bit 3 to zero forces Carrier
Detect inactive.

- | | | |
|---|---------|---|
| 2 | Control | Tx Enable. When set to one ,
the RS-422 drivers are enabled. |
| 1 | Control | Channel A Secondary Tx data,
asserted when 1. |
| 0 | Control | Channel B Secondary Tx data,
asserted when 1. |

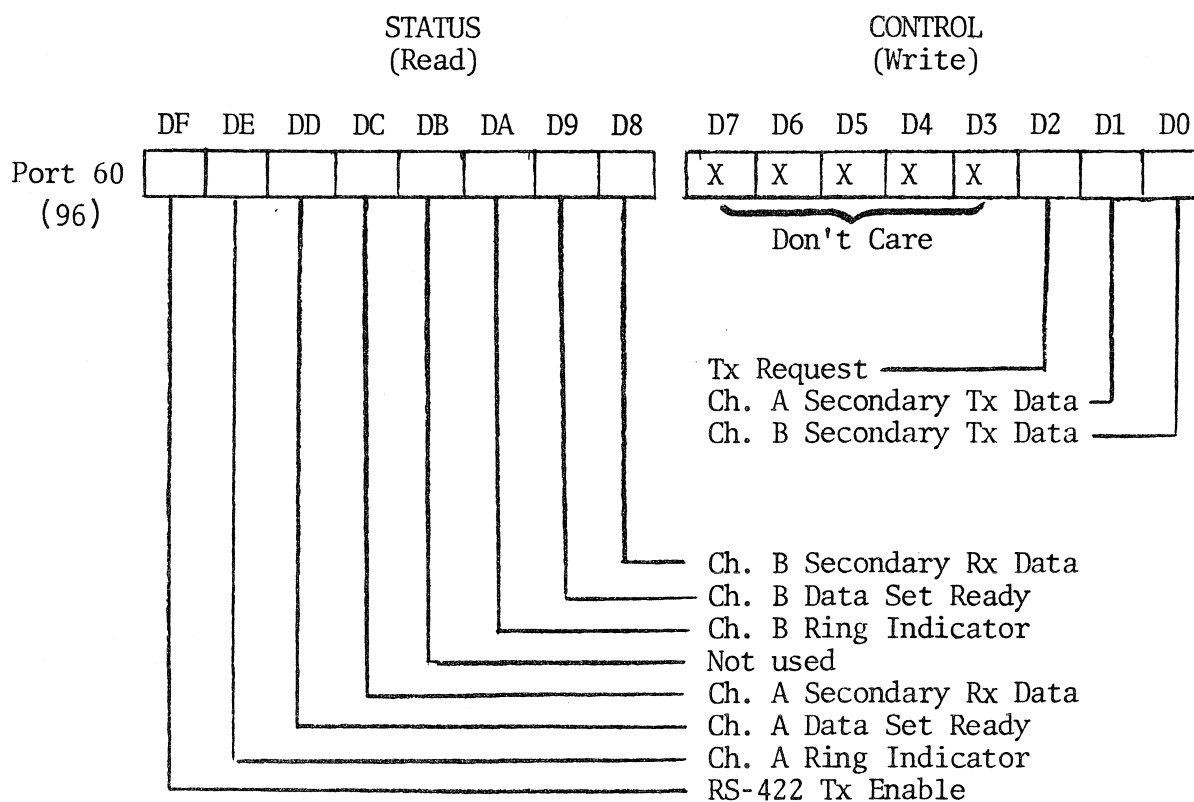


Figure 2-34. SIO Extended Control Register (ECR)

SIO Interrupt Acknowledge Register (IAR)

When the SIO interrupts the CPU, it is acknowledged through a special logic block designed to interface 8086 signals to the SIO's requirements. Within this arrangement, however, the only manner in which the SIO's internal interrupt-pending condition can be cleared is to read the Interrupt Acknowledge Register (IAR), port 30 (48). Figure 2-35 illustrates the register.

While this interrupt-pending disable function is the primary reason for having the IAR, the register also contains a copy of the SIO's Read Register 2. Reading the IAR for this information has the advantage of not requiring a prior write, as does reading the SIO's Read Register 2.

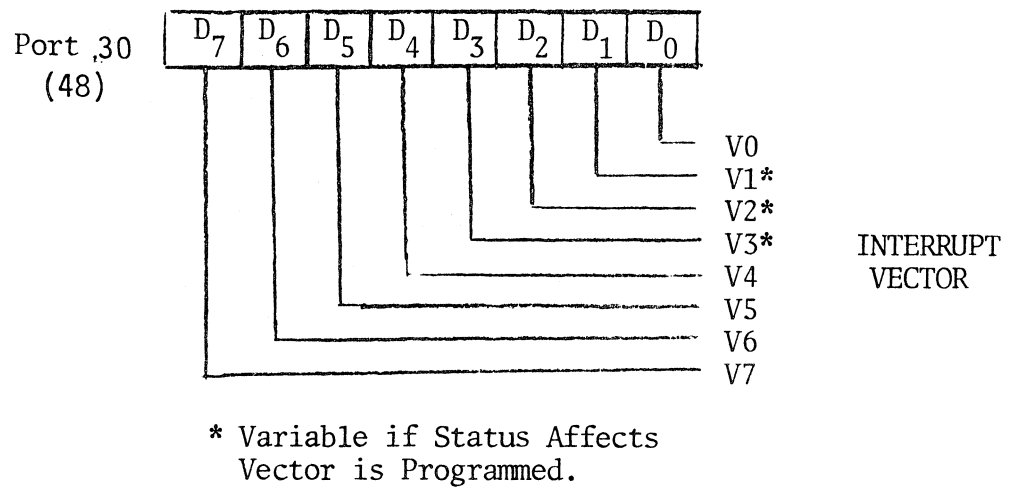


Figure 2-35. SIO Interrupt Acknowledge Register (IAC)

KEYBOARD INTERFACE

The keyboard is housed in a separate enclosure which contains the keys, a firmware-programmed 8048 microcomputer or equivalent with related logic, a set of eight user-programmable LED's, and a cable socket for attachment to the mainframe enclosure.

Keyboard/Mainframe Communication Functions

The 8048 in the keyboard enclosure communicates with the mainframe local bus via the 8251A, which constitutes the central component of the "keyboard interface" on the mainframe block diagram shown in Figure 2-1. The communication path is an asynchronous TTL-level bi-directional serial link that operates at 1220 baud. The path from the keyboard (8048) to the mainframe (8251A) is the one that bears the principal traffic; the opposite direction is used much less frequently for a small number of control and diagnostic commands. Simultaneous bi-directional traffic is supported.

The 8048 continually polls the keys to look for depressions. When it sees a new key depressed it sends its 8-bit code, together with the 8-bit codes of simultaneously-pressed keys, to the 8251A on the mainframe. Upon receiving each byte, the 8251A interrupts the 8086, which calls the service process that puts the byte into the memory buffer reserved for the keyboard.

Four types of commands can be sent to the 8048: a Reset, a request for the checksum of 8048 ROM, an echo command (followed by data to be echoed), and bit patterns for turning the 8 LEDs on or off. When initialized by the operating system to send commands to the 8048, the 8251A interrupts the CPU whenever it is ready to send a byte (i.e., whenever its transmit buffer is empty).

Data Formats

Several keys can be depressed simultaneously. Each key depression contributes a unique 8-bit non-ASCII byte to the outgoing stream of data, so that simultaneous key depressions can be used to obtain a rich variety of commands. Whenever a key changes from being up to being down, or vice

versa, a complete code sequence for all keys which are currently depressed is sent to the 8251A.

Software Interface

8251A Communication Interface

The 8251A is normally used only for asynchronous communication between the keyboard and the I/O-memory board, operating at 1220 baud. It is not intended as a user-programmable device nor can it be programmed for synchronous use.

The ports used for control, status and data are as follows:

<u>Port</u>	<u>Write</u>	<u>Read</u>
38 (56)	Data	Data
3A (58)	Commands	Status

When the standard operating system is used, all initialization and control of the 8251A is done by the operating system. Under these circumstances, users should never have occasion to program this device.

Transmit Functions: The 8251A occasionally functions as a transmitter in order to send commands to the 8048. Commands recognized by the 8048 are as follows:

COMMAND	HEX CODE
Reset	92 (146)
CheckROM	8C (140)
Echo	9E (159)
LED03	AX (160+X)
LED47	BX (176+X)

"Reset" cancels the Echo mode (if enabled) turns off all the LEDs, waits for any bytes in the keyboard buffer to be sent out and then initiates a new scan.

"CheckROM" computes a checksum of the 8048 ROM and sends it to the processor.

"Echo" causes the 8048 to send back the subsequent 256 input bytes verbatim unless one of those bytes is 92 (146) or 80 (128), which are interpreted as Reset or CheckROM commands.

"LED03" and "LED47" write the bit pattern specified by 'X' to LEDs 0-3 and 4-7 respectively. 'X' is the bottom nibble of the command byte.

To transmit data to the keyboard, the operating system sets the Tx enable bit in the 8251A. As soon as the 8251A has finished transmitting a byte it generates a level-four interrupt and the CPU sends either more data for transmission or a new control byte which disables the transmitter and therefore clears the interrupt.

As a manufacturing option, a jumper (G,H,I) is provided to bypass this interrupt; however, the jumper must be in position G-H whenever CT software is used.

Generalized Functions: Once programmed, the 8251A performs its communications functions as follows: The TxRDY output is raised active to signal the CPU that the 8251A is ready to receive a data character from the CPU. This output (TxRDY) is reset automatically when the CPU writes a character into the 8251A. On the other hand, the 8251A receives serial data from the keyboard. Upon receiving an entire character, the RxRDY output is raised active to signal the CPU that the 8251A has a complete character ready for the CPU. RxRDY is reset automatically upon the CPU data read operation. TxRDY and RxRDY cause keyboard level 4 interrupts.

The 8251A cannot begin transmission until the Tx Enable bit is set in the Command Instruction and it has received a Clear To Send input. The Tx output will be held in the marking state upon Reset.

Prior to starting data transmission or reception, the 8251A must be loaded with a set of control bytes.

The control bytes are split into two formats:

1. Mode Instruction
2. Command Instruction

Mode Instructions: The following mode is applicable to the CT system:

Asynchronous Transmission: Whenever a data character is sent by the CPU, the 8251A automatically adds a Start bit (low level) followed by the data bits (least significant bit first), and a Stop bit to each character. The character is then transmitted as a serial data stream on the Tx output. The

serial data is shifted out on the falling edge of TxC- at a rate equal to 1/16 that of the TxC-. When no data characters have been loaded into the 8251A the TxD output remains active (marking).

Asynchronous Receive: The RxD line is normally high. A falling edge on this line triggers the beginning of a START bit. The validity of this START bit is checked by again strobing this bit at its nominal center. If a low is detected again, it is a valid START bit, and the bit counter will start counting. The bit counter thus locates the center of the data bits and the stop bit. Data is sampled on the RxD pin with the rising edge of RxC-. If a low level is detected as the STOP bit, the Framing Error flag will be set. The STOP bit signals the end of a character. This character is then loaded into the parallel I/O buffer of the 8251A. The RxRDY pin is raised to signal the CPU that a character is ready to be fetched. If a previous character has not been fetched by the CPU, the present character replaces it in the I/O buffer, and the OVERRUN Error flag is raised (thus the previous character is lost). All of the error flags can be reset by an Error Reset Instruction. The occurrence of any of these errors will not affect the operation of the 8251A.

Commands: Once the functional definition of the 8251A has been programmed by the Mode Instruction, the device is ready to be used for data communication. The Command Instruction controls and actual operation of the selected format. Functions such as: Enable Transmit/Receive and Error Reset are provided by the Command Instruction.

Once the Mode Instruction has been written into the 8251A, all further control writes (C/D = 1) will load Command Instructions. A Reset Operation (internal or external) will return the 8251A to the Mode Instruction format.

Status Reads: Some of the bits in the Status Read Format have identical meanings to external output pins so that the 8251A can be used in a completely polled environment or in an interrupt driven environment. TxRDY is an exception.

Note that status update can have a maximum delay of 28 clock periods from the actual event affecting the status.

Figure 2-36 illustrates the 8251A control/status byte formats.

Commands To The Keyboard: The commands sent by the CPU to the keyboard via the 8251A are described in Chapter 9.

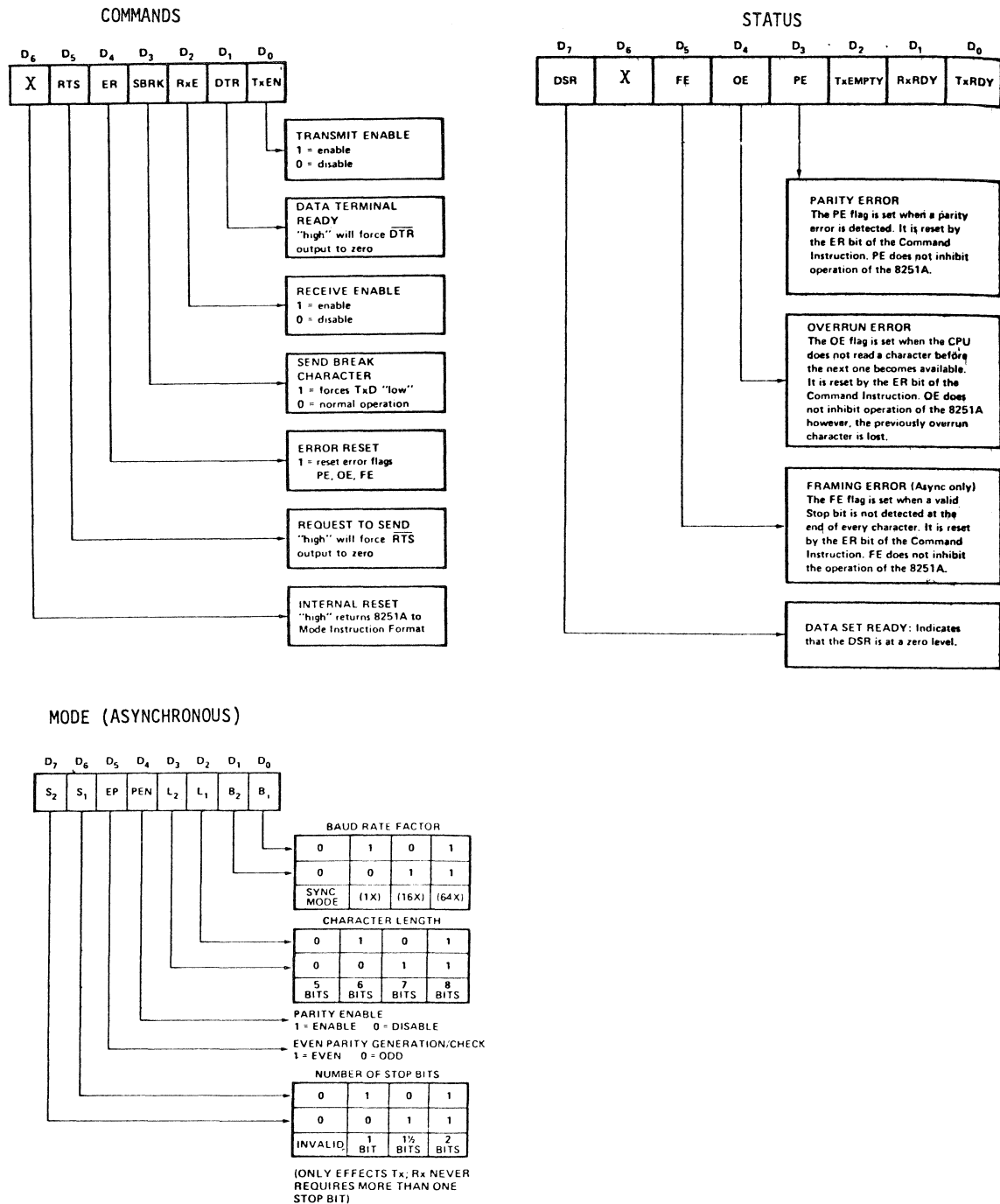


Figure 2-36. 8251A Control/Status

PRINTER INTERFACE

An 8-bit parallel Centronics-type printer interface is provided through a 25-pin D-type female connector on the mainframe enclosure. The interface outputs ASCII data from a memory buffer using the low byte of the MEM bus and I/O port 50 (80). It will accommodate both buffered and non-buffered printers that use the Centronics signaling format. A 4-bit Printer Status Register (PSR) is also available for reading status through port 50 (80).

The pinout for the 25-pin connector, which carries signals for both the data output, status input and control signals, is shown in Figure 2-37.

<u>Pin</u>	<u>Signal</u>	<u>Source</u>
1	Data bit 0	Mainframe
2	Data bit 1	Mainframe
3	Data bit 2	Mainframe
4	Data bit 3	Mainframe
5	Data bit 4	Mainframe
6	Data bit 5	Mainframe
7	Data bit 6	Mainframe
8	Data bit 7	Mainframe
9	Ground	
10	Ground	
11	Ground	
12	Ground	
14	Strobe (DATA STROBE/)	Mainframe
15	Ground	
16	Acknowledge (ACKNLG/)	Printer
17	Printer Busy (BUSY)	Printer
21	No Paper (PE)	Printer
22	Select (SLCT)	Printer
25	Ground	

Figure 2-37. Printer Connector Pinout

Software Interface

Printer Status Register (PSR): Port 50 (80) is used to indicate four status conditions of the printer. The bit assignments for the port are:

<u>Bit(s)</u>	<u>Function</u>
F-4	Not used
3	No paper in printer. Not all printers support this. It is merely a flag and does not generate an interrupt.
2	Buffer busy. A character or other command was sent to the printer and has not been acknowledged by the printer. This is the only printer-status flag that is generated by the mainframe logic rather than the external printer device.
1	Printer select. The printer is online and ready to print. The state of this bit is not affected by any data or control signals which the mainframe sends to the printer.
0	Printer busy. The printer is doing some mechanical activity, such as printing a buffered line or doing a line feed, and will not acknowledge the character just sent for a long time. Worst-case BUSY duration for a Centronics printer is over two seconds.

When printer interrupts are enabled in the IOCR and a character reception is acknowledged by the printer, an interrupt is generated. Typically printer interrupts should only be enabled when a character requiring mechanical motion is sent to the printer. This can be determined by checking bit 0 of the printer status, by checking the character codes sent, or by timing out for bit 2 of the status.

Figure 2-38 illustrates this figure.

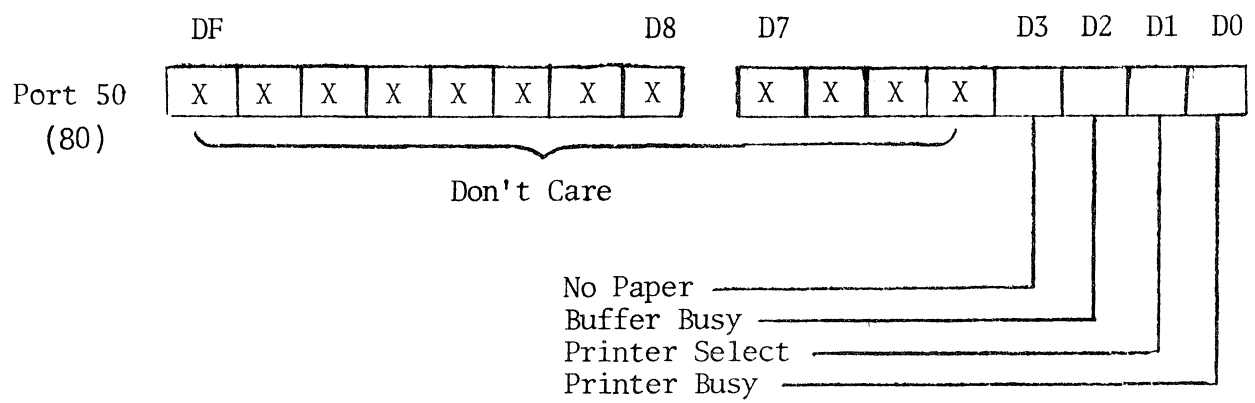


Figure 2-38. Printer Status Register (PSR)

UTILITIES

Programmable LEDs and Audio

Port 44 (68) is used to program the six LEDs at the top of the I/O-memory board, plus the audio signal. The bit assignments for this port are:

<u>Bit</u>	<u>Function</u>
0	When 0, lights LED # CR6
1	When 0, lights LED # CR5
2	When 0, lights LED # CR4
3	When 0, lights LED # CR3
4	When 0, lights LED # CR2
5	When 0, lights LED # CR1

Bits 6 and 7 are used together to operate the speaker as follows:

<u>Bit 6</u>	<u>Bit 7</u>	<u>Function</u>
0	0	No tone
1	0	625-Hz tone
0	1	Speaker in
1	1	Speaker out

Alternating between speaker in and speaker out at various rates can generate any audible frequency.

Figure 2-39 illustrates the control byte format.

Hardwired LEDs

Several non-programmable LEDs are wired to provide visible indication of hardware events to facilitate servicing of the mainframe. Their functions are described in the "Reference" sections of the chapters on each mainframe board.

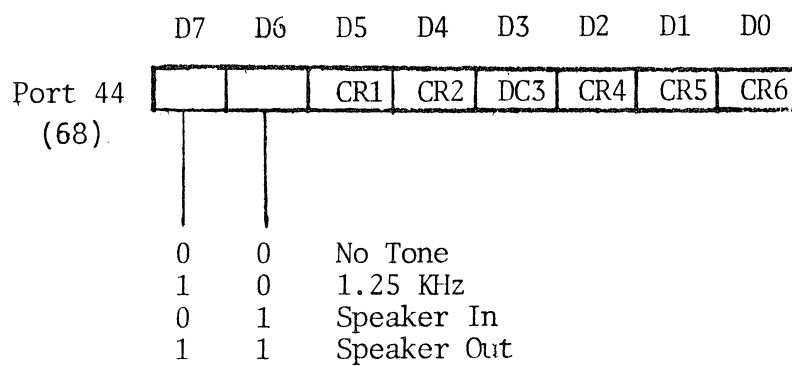


Figure 2-39. Programmable LED's and Audio

POWER

Power Supply

The mainframe power supply services all boards in the mainframe, plus the Multibus, the external keyboard and the CRT monitor. It does not service the disk drives or their controller boards.

Autorestart Function

The system has an automatic Restart/Reset function. This function does two things:

1. When the power switch is turned on a system Reset is automatically activated, thereby causing the CPU to jump to the boot-ROM routine. At the end of this routine, the operating system is loaded either from floppy drive 0 or Winchester drive 0.
2. When the power switch is turned on a bit (the COLD bit) in the Parity Error Register (PER) is set to indicate an automatic, rather than manual, Reset. The distinction can be used by the operating system in deciding whether to write the contents of RAM off to disk: in a manual Reset (COLD bit off), the contents of RAM are considered valid and can be saved for subsequent use; in an automatic (power-up) Reset (COLD bit on), RAM contents are considered invalid.

RESET

A system Reset can happen either automatically during power-up, or manually whenever the operator presses the Reset switch on the mainframe console. The distinction between the two is explained in the prior section entitled "Power."

When either type of Reset occurs, the following logic blocks are reset:

- CPU (8086)
- NDP (8087)
- DMA (8237-2)
- IOCR
- Communications (SIO)
- Multibus Interface
- Keyboard
- Printer timing
- I/O-memory board LEDs
- Video control and CRT blanking

When the IOCR is reset, the interrupt logic for all non-maskable interrupts and level-4 maskable interrupts is cleared and disabled until re-enabled, normally by the operating system.

Multibus resets will always reset the local mainframe.

Programmed I/O is required to reset the disk subsystem unless the disk subsystem undergoes a power-on. The CT system attends to this need.

3. MAINFRAME MOTHERBOARD

The mainframe motherboard has two or four Multibus slots, some Multibus priority resolution logic, terminations for Multibus lines, plus slots for the following boards:

- o processor
- o I/O memory
- o video
- o RAM and/or ROM expansion

It also contains connectors for the peripherals and the power supply. There is only one PC slot in the motherboard for expansion boards. If just a RAM expansion board or just a ROM expansion is used it may be plugged directly into the motherboard. If both are used, the ROM expansion board is plugged into the top of the RAM expansion board.

MULTIBUS PRIORITY RESOLUTION LOGIC

This circuit implements parallel priority resolution among up to three Multibus masters, one of which is this mainframe. Chip U1 determines whether or not at least one of the three Multibus request lines (BREQ0...2-) is active and sends out a binary code which corresponds to the highest priority request line which is now active. Chip U2 is a decoder that asserts the bus priority in (BPRN0...-) line which corresponds to the highest priority request presently active. Prioritization is as follows:

<u>line</u>	<u>priority</u>
BREQ0-	highest
BREQ1-	
BREQ2-	lowest

BREQ0- is assigned to the mainframe while BREQ1- and BREQ2- are assigned to the nearest and farthest Multibus slots (MB1 and MB2, respectively).

CLUSTER COMMUNICATIONS TERMINATION

The motherboard contains four of the six connectors located at the rear panel of the workstation. These support two RS232 interfaces, a parallel printer interface, and the interface to the mass storage subsystem.

One of the two RS232 interfaces may be converted to a RS422 interface for very high speed communications. If the conversion is made in order to implement the CT cluster connection, two additional 9 pin connectors are used in place of one of the RS232 connectors (J7). The 9 pin connectors are mounted on the rear panel and connected to the motherboard through headers J15 and J16. The 9 pin connector system mentioned above is included in all Integrated Workstations but is electrically disconnected. To engage this interface, the following jumper change is required.

The motherboard also contains socket U3. When communication channel A is used for a CT cluster, a jumper plug is inserted in U3. This causes channel A to be configured for half duplex communication, and also connects in the surge protection diodes CR1-CR8.

REFERENCES

Edge-Connector Pin List

The following several pages document the signals that are brought to the various edge connectors of the mainframe motherboard, showing the pin numbers at those connectors.

The connectors, in their order of appearance on the following pages, are:

<u>Number</u>	<u>Board/Cable</u>	<u>Pins</u>
J1	RAM Expansion Board*	80
J2	Processor Board	80
J3	Processor Board	120
J4	I/O-Memory Board	80
J5	I/O-Memory Board	120
J6	Video Control Board	80
J7	Comm. Ch. A Cable	25
J8	Comm. Ch. B Cable	25
J9	Disk Control Cable	50
J10	Keyboard Cable	9
J11	Printer Cable	25
J12	Power Cable	18
J13	Video Control Board	20
J14	Monitor Cable	20
J15	Cluster Comm. Cable	9
J16	Cluster Comm. Cable	9
J17	External Main. Cable	6
J18	Speaker Cable	2
J19	Reset Cable	2
J20	Power Indicator Cable	2
MB1	Multibus Board**	86
MB2	Multibus Board**	86

Notes: * ROM expansion board plugs into top of RAM expansion board or into J1.
 ** Slots for optional user-supplied boards.

Jumpers

<u>Name</u>	<u>Function</u>
A-B	Not normally used
C-D	Not normally used
E-F	Not normally used

Note: These jumpers link Multibus interrupt lines 1, 4, and 7 to three non-Multibus interrupt lines used within the mainframe. The operating system does not support any Multibus utilization of these three lines.

Figure 3-1. Edge-Connector Pin List

SIGNAL	MEM	CPU	CPU	I/O-	I/O-	VIDEO	COMM	COMM	DISK	KYBD.		PRINT-	PWR.	VIDEO	MONI-	DAISY	DAISY		
	EXP			MEM	MEM		CH.	CH.		J	J	ER			TOR	CHAIN	CHAIN		
	J1	J2	J3	J4	J5	J6	J7	J8	J9			J11	J12	J13	J14	J15	J16	MB1	MB2
BCLK-			9															13	13
INIT-			10															14	14
BPRNØ-			11																
BPRN1-																		15	
BPRN2-																			15
BUSY-			13															17	17
BREQØ-			14																
BREQ1-																		18	
BREQ2-																			18
MRDC-			16															19	19
MWTC-			17															20	20
IORC-			18															21	21
IOWC-			19															22	22
XACK-			20															23	23
INH1-			21															24	24
RESERVED			22								2							25	25
INH2-			23															26	26
BHEN-			24															27	27
AD10-			26															28	28
AD11-			28															30	30
AD12-			30															32	32
AD13-			32															34	34
CBRQ-			27															29	29
CCLK-			29															31	31
INTA-			31															33	33
INT7B-			33		34				*									36	36
INT6-			34															35	35
INT5-			36															38	38

▷ SEPARATE LINE
* JUMPER

DWG. NO.

A-43-00003-00

SHT. 2 OF 16

REV. A

Figure 3-2. Edge-Connector Pin List

SIGNAL	MEM EXP	CPU	CPU	I/O- MEM	I/O- MEM	VIDEO	COMM CH. A	COMM CH. B	DISK	KYBD.		PRINT- ER	PWR.	VIDEO	MONI- TOR	DAISY CHAIN	DAISY CHAIN		
	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J11	J12	J13	J14	J15	J16	MB1	MB2
INT4B-			37		37				*									37	37
INT3-			38		87													40	40
INT2-			39															39	39
INT1B-			40		40				*									42	42
INT0-			41															41	41
SIOACK-			42		42														
SIORQ+			43		43														
RDHIPEA-			44		44														
LBHEN+			46																
RDLOPEA-			47		47														
MW-			48		48														
DISKACK-			49		49														
DISKRQ+			50		50														
ODDP+			51		51														
WAIT-			52			44													
IOBCS+			53		53														
IOR-			54		54	46													
DCIRDY+			56		56														
FASTIO+			57		57														
DMACTCS-			58		58														
DMACS-			59		59														
IOB7+			60		60														
IOB6+			61		61														
IOB5+			62		62														
IOB4+			63		63														
IOB3+			64		64														
IOB2+			66		66														
IOB1+			67		67														

DWG. NO.
A-43-00003-00
SHT. 3 OF 16
REV. A

Figure 3-3. Edge-Connector Pin List

SIGNAL	MEM EXP	CPU		I/O- MEM	I/O- MEM	VIDEO	COMM CH. A	COMM CH. B	DISK	KYBD.		PRINT- ER	PWR.	VIDEO	MONI- TOR	DAISY CHAIN	DAISY CHAIN		
	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J17	J11	J12	J13	J14	J15	J16	MB1	MB2
IOB0+			68		68														
DREQ-					69				35										
IOCS-			70		70														
NMI+			71		71														
PICCS-			72		72														
MEMCLK+			73		73	9													
TEST			74								3								
LPTACK-					74							16							
IOW-			76		76	14													
EOP-			77		77	11													
VIDACK-			78			17													
PERINT-			79		79														
MWINH-			80		80														
TCLK-					9		18									6	6		
TCLK+					10		25									7	7		
RXDA-					11		3												
CTSA+					12		5												
RXCA+					13		17									6	6		
TXCA+					14		15									7	7		
DSRA+					16		6												
SRDA+					17		16												
STDA-					18		14												
TXDA-					19		2									8	8		
RTSA+					20		4									9	9		
DTRA+					21		20												
DCDA+					22		8									8	8		
RINGA+					23		22									9	9		
CTSB+					24			5											
(SEE NOTE 1 SHEET 16)																			
DWG. NO. A-43-00003-00 SHT. 4 OF 16 REV. A																			

Figure 3-4. Edge-Connector Pin List

SIGNAL	MEM EXP	CPU	CPU	I/O- MEM	I/O- MEM	VIDEO	COMM CH. A	COMM CH. B	DISK	KYBD.		PRINT- ER	PWR.	VIDEO	MONI- TOR	DAISY CHAIN	DAISY CHAIN		
	J1	J2	J3	J4	J5	J6	J7	J8	J9	J	J							MB1	MB2
										10	17	J11	J12	J13	J14	J15	J16		
RXDB-					26			3											
TXCB+					27			15											
RXCB+					28			17											
DSRB+					29			6											
SRDB-					30			16											
DCDB+					31			8											
RINGB+					32			22											
DTRB+					33			20											
RTSB+					36			4											
STDB-					38			14											
TXDB-					39			2											
KBD1-					41					1									
KBDØ-					46					3									
ENBL-					52				18										
ADRF-			86															44	44
ADRE-			87															43	43
ADRD-			88															46	46
ADRC-			89															45	45
ADRB-			90															48	48
ADRA-			91															47	47
ADR9-			92															50	50
ADR8-			93															49	49
ADR7-			94															52	52
ADR6-			96															51	51
ADR5-			97															54	54
ADR4-			98															53	53
ADR3-			99															56	56
ADR2-			100															55	55

DWG. NO.

A-43-00003-00

SHT. 5 OF 16

REV. A

Figure 3-5. Edge-Connector Pin List

SIGNAL	MEM EXP	CPU	CPU	I/O- MEM	I/O- MEM	VIDEO	COMM CH. A	COMM CH. B	DISK	KYBD.		PRINT ER	PWR.	VIDEO	MONI- TOR	DAISY CHAIN	DAISY CHAIN		
	J1	J2	J3	J4	J5	J6	J7	J8	J9	J	J	J11	J12	J13	J14	J15	J16	MB1	MB2
										10	17								
ADR1-			101															58	58
ADR0-			102															57	57
T0+			81		81														
MB10-			82		82														
RESET-			83		83	13													
PAREN-			84		84														
DATF-			103															60	60
DATE-			104															59	59
DATD-			106															62	62
DATC-			107															61	61
DATB-			108															64	64
DATA-			109															63	63
DAT9-			110															66	66
DAT8-			111															65	65
DAT7-			112															68	68
DAT6-			113															67	67
DAT5-			114															70	70
DAT4-			116															69	69
DAT3-			117															72	72
DAT2-			118															71	71
DAT1-			119															74	74
DAT0-			120															73	73
ACK-					86				2										
CNTRL-					88				20										
SELF-					89				4										
SELW-					90				37										
WRSTB-					91				22										
INTRQ-					92				6										
DWG. NO. A-43-00003-00 SHT. 6 OF 16 REV. A																			

Figure 3-6. Edge-Connector Pin List

SIGNAL	MEM EXP	CPU	CPU	I/O- MEM	I/O- MEM	VIDEO	COMM CH: A	COMM CH: B	DISK	KYBD.		PRINT ER	PWR.	VIDEO	MONI- TOR	DAISY CHAIN	DAISY CHAIN		
	J1	J2	J3	J4	J5	J6	J7	J8	J9	J	J	J11	J12	J13	J14	J15	J16	MB1	MB2
										10	17								
RDSTB-					93				39										
BUS0+					94				24										
BUS1+					99				10										
BUS2+					96				8										
BUS3+					100				43										
BUS4+					97				41										
BUS5+					101				28										
BUS6+					98				26										
BUS7+					102				12										
BUS8+					107				47										
BUS9+					111				33										
BUS A+					106				14										
BUS B+					110				49										
BUS C+					104				30										
BUS D+					109				16										
BUS E+					103				45										
BUS F+					108				32										
LPT0+					117							1							
LPT1+					112							2							
LPT2+					118							3							
LPT3+					113							4							
LPT4+					120							5							
LPT5+					114							6							
LPT6+					119							7							
LPT7+					116							8							
MR-	10	1		1		10													
VIDRQ+		2				16													
NO PAPER+				2								21							
DWG. NO. A-43-00003-00 SHT. 7 OF 10 REV. A																			

Figure 3-7. Edge-Connector Pin List

SIGNAL	MEM EXP	CPU	CPU	I/O- MEM	I/O- MEM	VIDEO	COMM CH: A	COMM CH: B	DISK	KYBD.		PRINT- ER	PWR.	VIDEO	MONI- TOR	DAISY CHAIN	DAISY CHAIN		
	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J17	J11	J12	J13	J14	J15	J16	MB1	MB2
BD2RAM-	9	3																	
LPTBUSY+				3								17							
BD1RAM-		4		4															
COLMPX-	12	6		6															
RARAS-	14	7		7															
PROMEXDC-	11	8																	
BEOP-				8					48										
RFGO-	16	9		9															
RFDONE-	13	10		10															
RAHIWE-	18	11		11															
RALOWE-	17	12		12															
GACCS-				18		12													
78KHZ+		13		13															
SFØ				14									16						
CAS7-	19	14																	
CAS6-	20	16																	
CAS5-	21	17																	
CAS4-	22	18																	
CAS3-		19		19															
CAS2-		20		20															
CAS1-		21		21															
CASØ-		22		22															
LPT SELECT+				16								22							
AUDIO+				17															
HIP+	23	23		23															
LOP+	24	24		24															
MEMF+	26	26		26		26													
MEME+	27	27		27		27													
DWG. NO.										A-43-00003-00				SHT. 8 OF 16			REV. A		

Figure 3-8. Edge-Connector Pin List

SIGNAL	MEM EXP	CPU		I/O- MEM	I/O- MEM	VIDEO	COMM CH: A	COMM CH: B	DISK	KYBD.		PRINT- ER	PWR.	VIDEO	MONI- TOR	DAISY CHAIN	DAISY CHAIN		
	J1	J2	J3	J4	J5	J6	J7	J8	J9	J	J								
	10	17	J11	J12	J13	J14	J15	J16	MB1	MB2									
MEMD+	28	28		28		28													
MEMC+	29	29		29		29													
MEMB+	30	30		30		30													
MEMA+	31	31		31		31													
MEM9+	32	32		32		32													
MEM8+	33	33		33		33													
MEM7+	34	34		34		34													
MEM6+	36	36		36		36													
MEM5+	37	37		37		37													
MEM4+	38	38		38		38													
MEM3+	39	39		39		39													
MEM2+	40	40		40		40													
MEM1+	41	41		41		41													
MEM Ø+	42	42		42		42													
LA13+	43	43		43															
LA12+	44	44		44															
LA11+	46	46		46															
LA10+	47	47		47															
LAF+	48	48		48		48													
LAE+	49	49		49		49													
LAD+	50	50		50		50													
LAC+	51	51		51		51													
LAB+	52	52		52		52													
LAA+	53	53		53		53													
LA9+	54	54		54		54													
LA8+	56	56		56		56													
LA7+	57	57		57		57													
LA6+	58	58		58		58													
DWG. NO.										A-43-00003-00				SHT. 9 OF 16			REV. A		

Figure 3-9.

DWG. NO.	A-43-00003-00	SHT. 10 OF 16	REV. A
----------	---------------	---------------	--------

3-1-3

SIGNAL	MEM EXP	CPU	CPU	I/O-MEM	I/O-MEM	VIDEO	COMM CH. A	COMM CH. B	DISK	KYBD.		PRINT-ER	PWR.	VIDEO	MONI-TOR	DAISY CHAIN	DAISY CHAIN		
	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J17	J11	J12	J13	J14	J15	J16	MB1	MB2
+ 5V ↑	1		1		1	1				6			1					3	3
	2		2		2	2				7			2					4	4
	3		3		3	3							3					5	5
	4		4		4	4							7					6	6
↓	77	77		77		77							10					81	81
	78	78		78		78							18					82	82
	79	79		79		79												83	83
	80	80		80		80												84	84
+ 12V ↑	67	67		67		67					1		5		20			7	7
	68	68		68		68							6					8	8
	69	69		69		69													
+ 12V	70	70		70		70													
- 12V ↑	5		5		5	5					4		8					79	79
	6		6		6	6												80	80
	73			73															
- 12V	74			74															
+32V													12		2				
-32V													14		1				

DWG. NO.
A-43-00003-00
SHT. 11 OF 16
REV. A

[illegible]

Figure 3-12. Edge-Connector Pin List

[illegible]

When configured for cluster control the 9 pin connectors located on the rear panel are connected to the following functions:

P15/P16 Channel A Function

Pin	Signal Name
1	GND
2	GND
3	GND
4	GND
5	GND
6	CLK+
7	CLK-
8	DATA+
9	DATA-
10	---

Figure 3-13. Edge-Connector Pin List

The following reference information describes the inter-connection between motherboard signals and the cluster headers, J15 and J16, when a one to one jumper plug is installed in socket U3.

SIGNAL	I/O- MEM J5	COMM CH. A J7	DAISY CHAIN J15	DAISY CHAIN J16
RXCA+	13	17	6	6
TCLK-	9	18	6	6
TXCA+	14	15	7	7
TCLK+	10	25	7	7
DCDA+	22	8	8	8
TXDA-	19	2	8	8
RINGA+	23	22	9	9
RTSA+	20	4	9	9

Figure 3-14. Edge-Connector Pin List

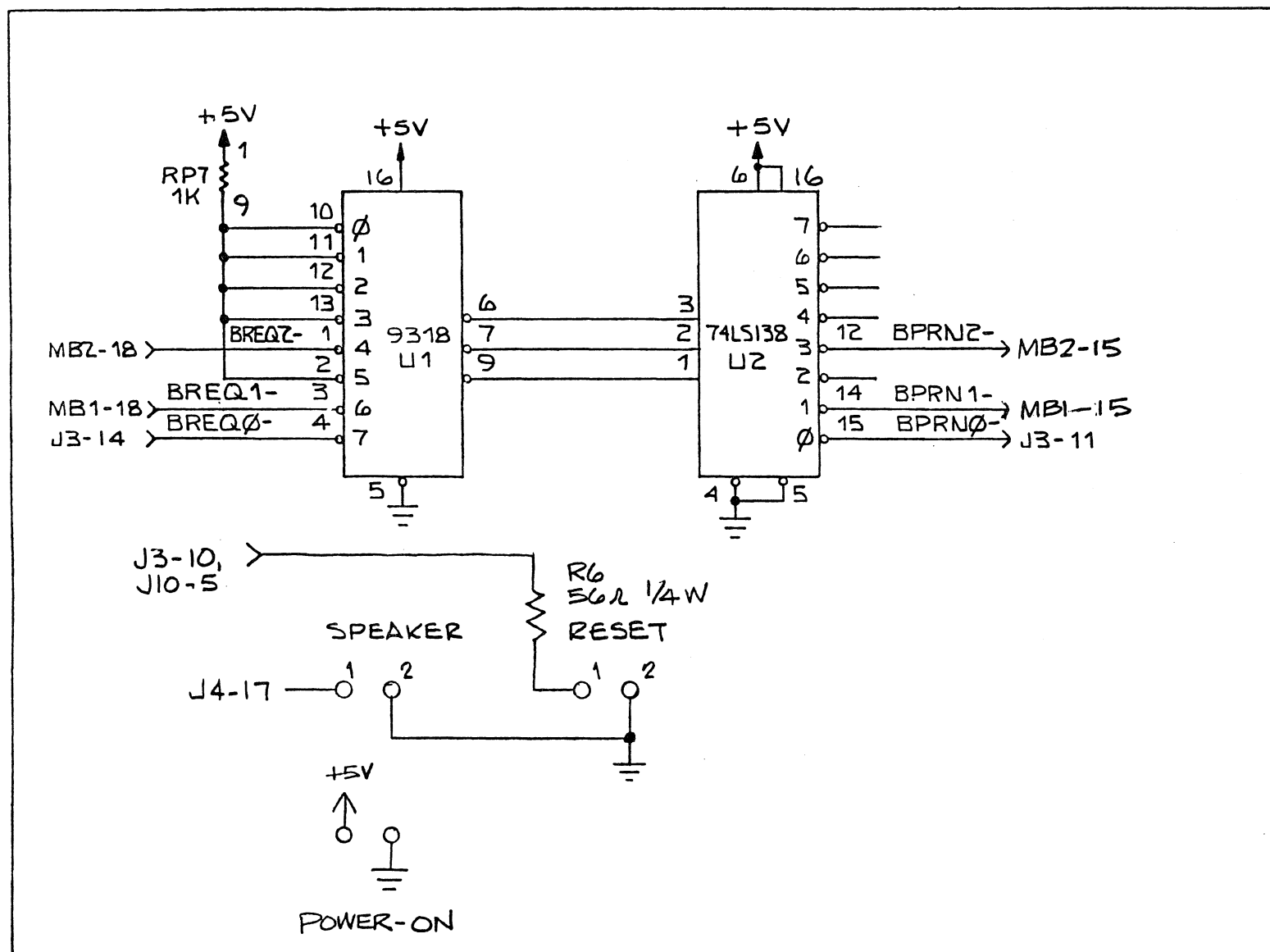


Figure 3-16. Mainframe Motherboard Schematic

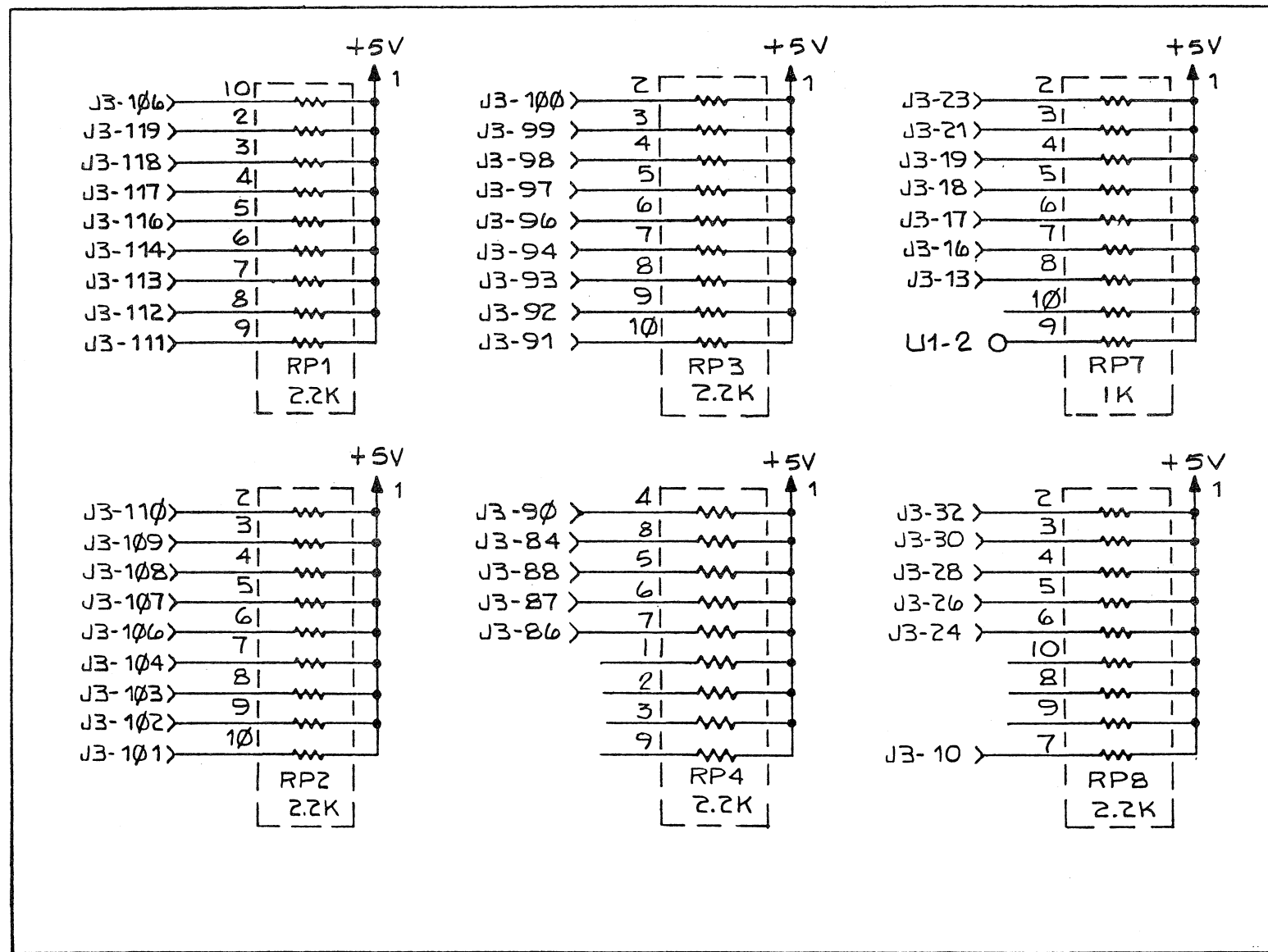
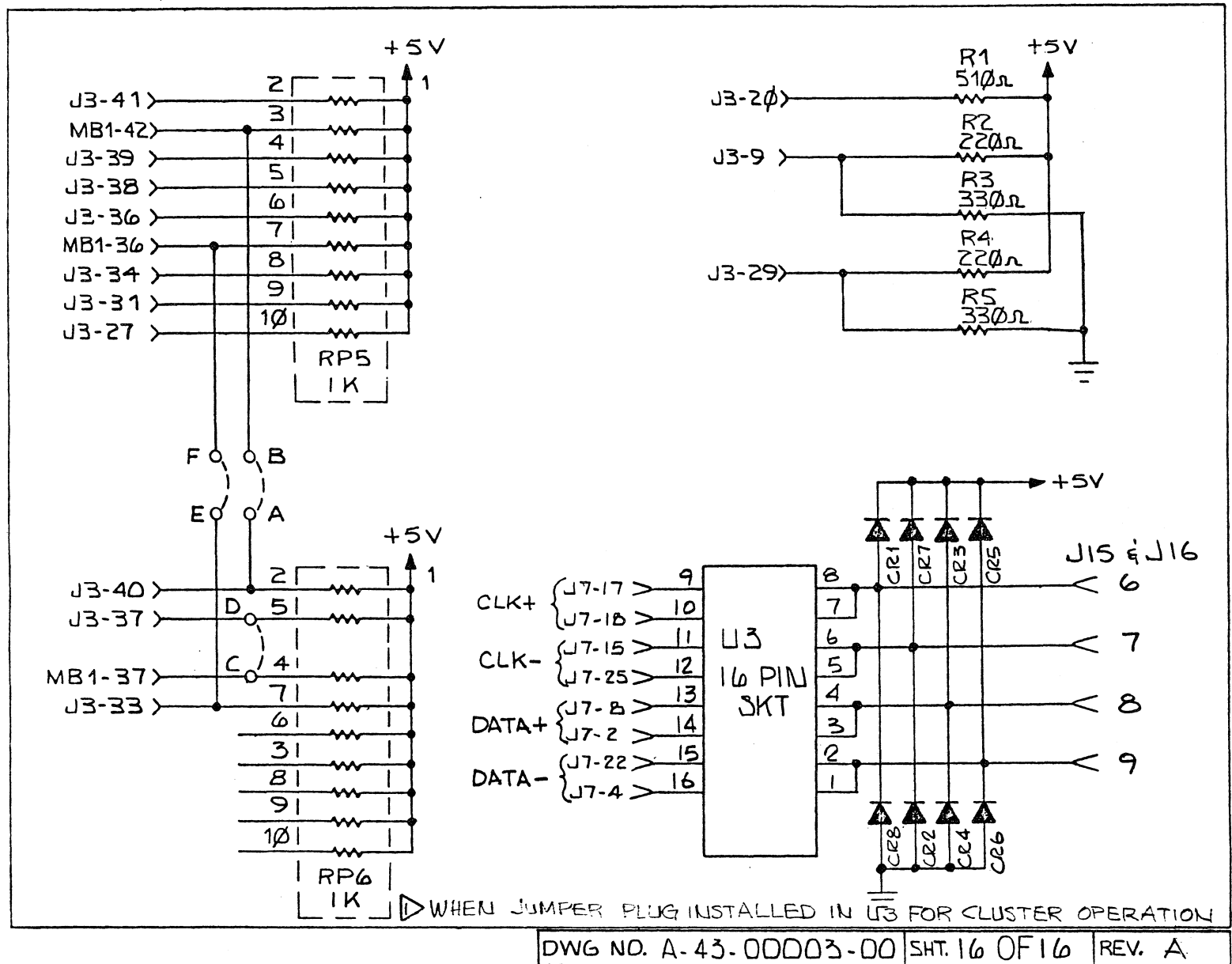
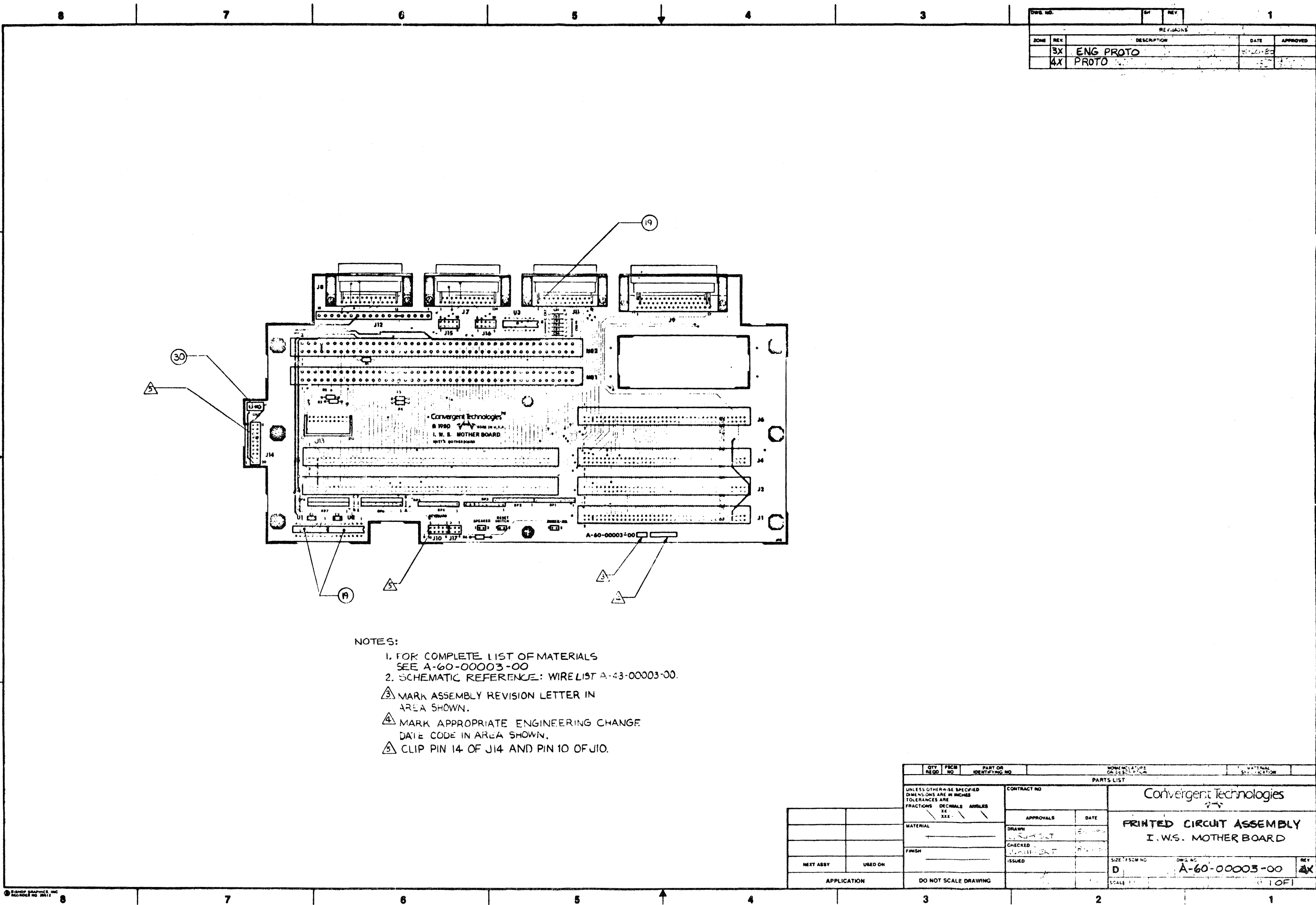


Figure 3-17. Mainframe Motherboard Schematic





- NOTES:
1. FOR COMPLETE LIST OF MATERIALS SEE A-60-00003-00
 2. SCHEMATIC REFERENCE: WIRE LIST A-43-00003-00.
 3. MARK ASSEMBLY REVISION LETTER IN AREA SHOWN.
 4. MARK APPROPRIATE ENGINEERING CHANGE DATE CODE IN AREA SHOWN.
 5. CLIP PIN 14 OF J14 AND PIN 10 OF J10.

DATE NO.		REV	REVISIONS	DATE	APPROVED
ZONE	REV	DESCRIPTION			
3X		ENG PROTO		11-10-80	
4X		PROTO			

QTY REQD	PCB NO	PART OR IDENTIFYING NO	NOMENCLATURE OR DESCRIPTION	DATE	REVISION
PARTS LIST					
Convergent Technologies					
PRINTED CIRCUIT ASSEMBLY					
I.W.S. MOTHER BOARD					
MATERIAL		CONTRACT NO		DATE	
FINISH		APPROVALS		DATE	
NEXT ASSY		DRAWN		DATE	
USED ON		CHECKED		DATE	
APPLICATION		ISSUED		DATE	
DO NOT SCALE DRAWING		SCALE		1/10	
SIZE/PCB NO		DRAWING		REV	
D		A-60-00003-00		4X	

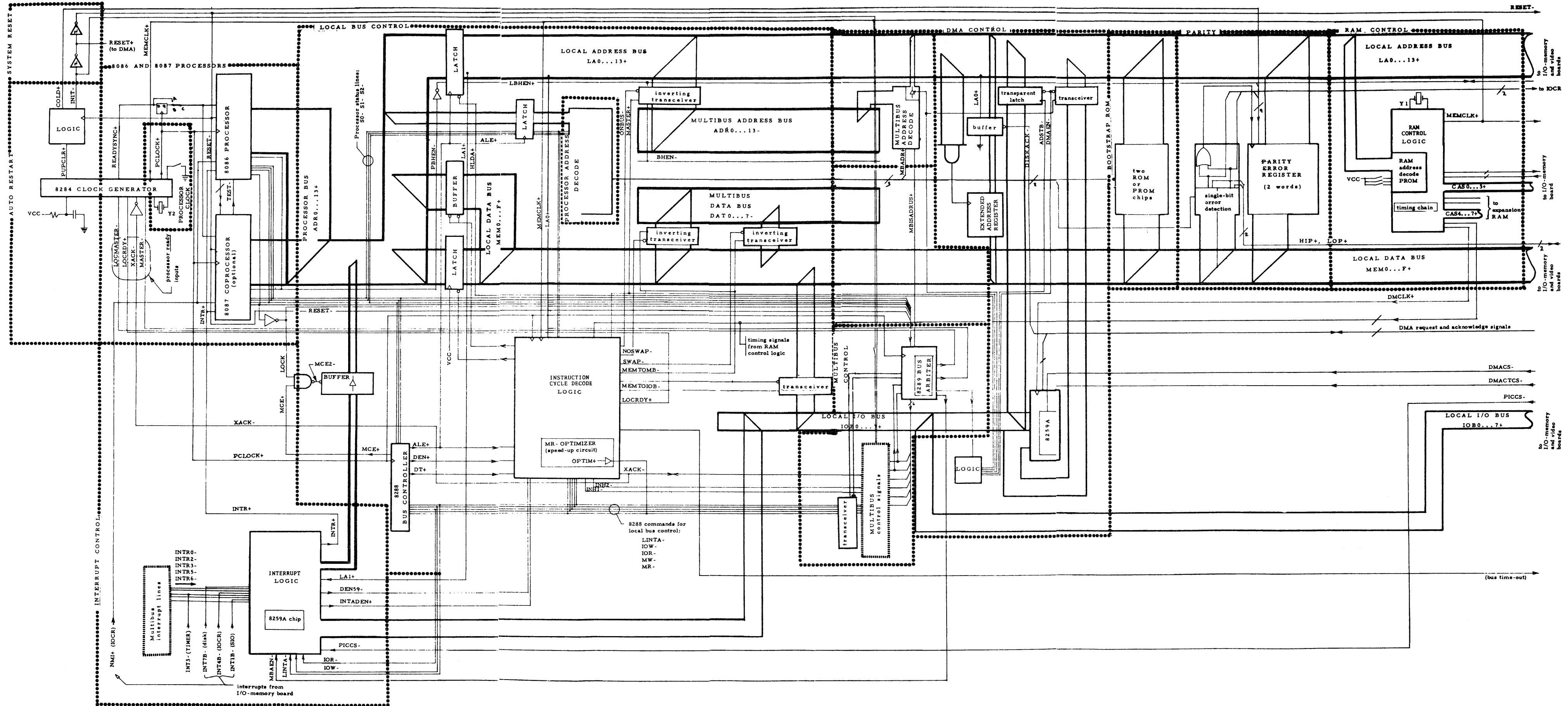
4. PROCESSOR BOARD

OVERVIEW

The processor board is one of two mandatory boards which plug into the mainframe motherboard to implement the minimum-configuration mainframe. Many of the board's functions are built around programmable LSI chips which act as peripherals to the 8086. Optionally, the processor board can be configured with an 8087 numeric data processor (NDP) in addition to the 8086.

OPERATION BY LOGIC BLOCK

Figure 4-1, shows how the major logic blocks are defined and how they relate to one another. The primary reference for logic, however, is the schematics at the end of this chapter. References in the text below contain parenthesized page numbers, and often zone numbers, for these schematics; for example, the reference (p1zA3) means page 1, zone A3 of the schematics.



The board contains the following blocks.

- 1) The dual bus logic
- 2) The 8086 microprocessor
- 3) The 8087 numeric data processor
- 4) Memory control and refresh logic
- 5) RAM parity check and generate logic
- 6) Interrupt control logic
- 7) Direct Memory Access (DMA) control logic
- 8) The Multibus interface
- 9) Bootstrap logic and the bootstrap PROMs

Each of the above blocks is discussed in detail below.

The Dual Bus Architecture

The mainframe uses a dual bus architecture. The CT logic boards reside on the local bus (CT-Bus) and a second bus, Multibus, is available for user add-on logic boards. The local bus can be in three different states as follows:

<u>state</u>	<u>meaning</u>
Idle	The bus is not being used
CPU active	The CPU is using the local bus for an I/O, interrupt acknowledge, or memory cycle. Addresses come from the CPU.
DMA active	The DMA logic is using the local bus for communication, disk, or video DMA transfers. The address comes from the DMA logic. For slave Multibus DMA transfers, the address comes from the Multibus.

In addition, the Multibus to local-bus interface can be in one of three states as follows:

<u>state</u>	<u>meaning</u>
master	The local-bus is master over the Multibus and is driving its command and address signals.
slave	The local-bus is slave to the Multibus and its command and addresses are driven by the Multibus.
Independent	The local-bus and Multibus operate independently.

Control Of The Local Bus Drivers

The local address bus drivers and associated control logic for DMA transfers are described in the DMA section, below.

The 8086 address/data bus connects the 8086 to the 8087 and carries information to the local bus latches and drivers. The AD bus is multiplexed, containing address and data. The signal ALE+ from the 8288 indicates when the AD bus carries address information.

The three latching local address drivers at 11E, 15D, and 15E (p7) latch the contents of AD when ALE+ is active. As long as the DMA state is not active (HLDA+ inactive) these three latches drive the local address bus.

The chips at 14D and 14E (p8) drive data from the AD bus to the MEM (MEMO+ MEMF+) on to the local bus latches and drivers. The AD bus is multiplexed, containing address and data. The signal ALE+ from the 8288 indicates when the AD bus carries address information.

The three latching local address drivers at 11E, 15D, and 15E (p7) latch the contents of AD when ALE+ is active. As long as the DMA state is not active (HLDA+ inactive) these three latches drive the local address bus.

The chips at 14D and 14E (p8) drive data from the AD bus to the MEM (MEMO+ MEMF+) 5E (p7) latch the contents of AD when ALE+ is active. As long as the DMA state is not active (HLDA+ inactive) these three latches drive the local address bus.

The chips at 14D and 14E (p8) drive data from the AD bus to the MEM (MEMO+ MEMF+) ## to the local bus latches and drivers. The AD bus is multiplexed, containing address and data. The signal ALE+ from the 8288 indicates when the AD bus carries address information.

The three latching local address drivers at 11E, 15D, and 15E (p7) latch the contents of AD when ALE+ is active. As long as the DMA state is not active (HLDA+ inactive) these three latches drive the local address bus.

The chips at 14D and 14E (p8) drive data from the AD bus to the MEM (MEMO+ MEMF+) bus when the

signal ADTOMEM- is active. ADTOMEM- goes active at 3B-8 (p5zA5) when HLDA+ is false (no DMA), and the DEN+ and DT+ signals from the 8288 indicate that the 8086 is outputting data.

The chips at 13D and 13E (p8) drive data to the ##D bus from the MEM bus when MEMTOAD- is active. MEMTOAD- is asserted by 6E-8 (p1zA3) when DEN+ is asserted by the 8288, indicating that the 8086 is using AD for a data transfer, and the 8086 status signal S1- is low, indicating that an input command is in progress.

The transceiver at 9F (p8) connects the I/O bus (IOB0+ - IOB7+) and the MEM bus (MEM0+ - MEM7+). It is disabled with ADSTB+ (address strobe) during DMA transfers when the 8237 is using the IOB bus to transfer address information to its high-order address latch at 5G (p7) as described in the DMA section. The direction of the transceiver, 9F (p8), is established by the signal MEMTOIOB+. When active, MEMTOIOB+ causes the MEM bus to drive the IOB bus. MEMTOIOB+ is driven low, causing the IOB bus to drive the MEM bus, by gate 4D-13 (p5zB2), when the 8259A is supplying an interrupt vector on IOB (INTADEN+ active) or when 4C-11 is high. 4C-11 goes high when an I/O read is in progress (IOR- active) and either communications DMA is in progress (SIOACK- active) or the 8086 is addressing a device on the IOB bus (IOBCS+ active).

Processor Ready Logic

Each time the 8086 issues a bus command (IOR-, IOW-, MR-, MW-, or INTA-) it waits for its READY input to go active before completing the command. This facility is used to stretch the commands for devices that are slower than the 8086 minimum access time and for devices that are not ready to transfer.

The 8284 (p1) combines two READY sources, Multibus and local bus. For Multibus Master mode, the LOCMaster- (local master) signal is caused to be inactive at 5C-13 (p4zA4). This causes the 8284 to ignore LOCRDY+ (local bus ready). Since MASTER- is active, pin 16C-4 (p1zD7) of the 8284 is the source for READY generation. XACK- comes from the Multibus directly and is driven by whatever slave is active.

There are two special cases when XACK- may be asserted by the CPU board even though it is the Multibus master: interrupt acknowledge (sometimes) and bus timeout. Gate 4A-8 (p5zB2) asserts XACK- in the two special cases. 4A-8 goes active if MASTER+ is asserted and MASACK+ (master acknowledge) is asserted. MASACK+ is the OR of three cases at 7D-12 (p67zC5). 7D-13 goes active if the cycle timed out, that is, no slave acknowledged in 6 milliseconds. 7D-2 goes active during the first of the pair of INTA- pulses issued during interrupt acknowledge and 7D-1 goes active during the second INTA- pulse if the 8259A on the processor board is supplying the interrupt vector (as opposed to one on Multibus). In all other cases, the addressed Multibus slave responds by driving XACK- active to terminate the cycle.

When the 8086 is not Multibus master (MASTER+ inactive) and no DMA transfer is in progress, 5CD-12 (p4zA4) asserts LOCMaster- (local master). This causes the 8284 to use LOCRDY+ (at 16C-6) (p1) as the source for READY+. LOCRDY+ is generated by gate 5C-8 (p5zB4) and is the OR of several cases. Pins 11 and 10 of 5CV are the RAM acknowledges.

Pin 11 is used if the cycle will have wait states and pin 10 is used for no-wait-state access. The SLOW+/FAST+ flip-flop, 13B-9 (p5zA6), is set to SLOW+ for conditions when wait states are required and to FAST+ when they are not. When FAST+ is active, OCRDY+ is asserted by 7C-8

(p5zA5) as soon as the memory cycle starts (ENMCYC+ active). When SLOW+ is active, LOCRDY+ is asserted by 6C-3 (p5zB6) when PRACK+ (predicted RAM acknowledge) is asserted by the RAM control logic indicating the cycle is about to complete. The SLOW+/FAST+ flip-flop is set to FAST+ whenever the signal IDLE+ goes active. IDLE+ goes active when the 8086 is doing internal cycles and between cycles. The SLOW+/FAST+ flip-flop is set SLOW+ at 13B-10 if PWAIT- goes active. PWAIT- indicates that the 8086 commands have been delayed at the 8288 by DMA or while waiting for Multibus Master status. Also, when a memory cycle starts (MCYC- goes active), if a refresh is pending (RFPNDG-) or the memory control is in the end of a previous cycle (ENDCYC-), gate 6C-11 (p5zB6) goes high, causing SLOW+ to be set.

The non-RAM sources for LOCRDY+ are ORed by gate 6A-8 (p5zB5). The key to the operation of 6A is the shift register at 5A (p6zD6) which counts MEMCLK+ pulses since the beginning of the command. Gate 9C-8 (p6zD7) goes high when any command line is asserted. This allows 5A to shift a 1 (from R3) through its stages, generating the signals 100nS+, 300nS+, 400nS+, and 500nS+ for the corresponding amount of time since the command began.

LOCRDY+ for PROM accesses is generated when PROMCS+ and 500nS+ are active, causing 6A-8 (p5zB5) to go active. This allows at least 500 nanoseconds for PROM access.

The signal IOBCS+ (from the I/O Memory board) indicates that the I/O bus (IOB0+ - IOB7+) is being used for I/O. In that case LOCRDY+ is asserted when 400nS+ goes active, allowing 400 nanoseconds for access to LSI peripheral chips. TMO-, ORed with IOBCS+ at 12B-11 (p5zB6), is a bus timeout signal, used to generate READY if nothing else responds within 6 milliseconds.

FASTIO+ is asserted by the I/O Memory board when a TTL type I/O device, which does not need wait states, is addressed. In this case, after 100 nanoseconds, LOCRDY+ is asserted unless DCIRDY+ is false. DCIRDY+ false indicates that an addressed disk control is not present. It is generated in the disk control interface (DCI) on the I/O Memory board.

Bus Timeouts

The one-shot at 11D-10 (p6) is used to detect bus timeouts. A bus timeout occurs if no command has been started for 6 milliseconds. When that occurs, TMO- (11D-9) is asserted. This causes LOCRDY+ to be asserted as described. If a command is active when the timeout occurs, TO+ will be asserted. TO+ forces MEMRDY+ true in the DMA logic. TO+ is also used on the I/O Memory board to conditionally cause non-maskable interrupts (NMI) depending on the setting of the IOCR (port 56 on the I/O Memory board).

The 8086 Microprocessor

The 8086 and its associated logic are resident on the processor board and are primarily on page one of the schematics. The associated logic consists of instruction decode, reset, and clock generation logic.

Chip 16D (p1) is the 8086 itself. Because pin 33 is strapped low, the chip is in the maximum mode. The maximum mode is used so that the 8288 and 8087 can be used with the 8086.

Inverter 15B-10 (p1zB7) and CR5 are a driver and LED that lights when a Non-Maskable Interrupt is present.

Pin 22 of the 8086 is connected through jumper B-C to flip-flop 15C-9, which is used to synchronize the READY+ line to the 8086. If the 8284 clock generator chip is changed to generate a synchronized READY+, jumper B-C can be removed and jumper A-C installed to use READY+ directly from the 8284 (p1).

The 8284 provides clock generation and reset synchronization logic. Pins 16 and 17 connect to the 15 MHz or optional 24 MHz crystal. This clock is divided by three to generate a 33% duty cycle processor clock (PCLK+). Optionally, for debug, switch 1 bit 4 (bit 4 is connected by pins 4 and 5 of switch 1) can be opened to use the 10 MHz MEMCLK+ as the time base for the processor clock. This results in a 3.3 MHz processor operation that is synchronous with the memory clock. The 8284 also generates and synchronizes the READY+ line for the 8086. READY+ is formed by ORing two sources, Multibus and the local bus. READY+ is generated by the local bus (LOCRDY+) when LOCMaster+ is true and by Multibus

(XACK-) when MASTER- is active. The logic within the 8284 is $READY+ = (LOCMASTER+ * LOC RDY+) + (MASTER+ * XACK+)$. The 8284 also synchronizes READY+ to PCLK+ and the flip-flop 15C-9 provides protection from delays produced by a metastable condition of the 8284 logic. Finally the 8284 also generates power-up clear (PUPCLR+) from RC combination R24 and C14.

Flip-flop 15C-5 (p1zC6) is called the COLD flip-flop. It is set by the power-up RC circuit just described and any other resets will clear it. It can be sensed by software by reading port 54.

PUPCLR+ is buffered by 15B-4 (p1zD6) which drives the Multibus INIT- pin. All other sources for reset drive the INIT- signal low (it is open collector). The INIT- signal is received and buffered by 11B-13 and 11B-10 to generate the system RESET+ and RESET- signals.

Latch 10E (p1zB5) contains the status signals from the 8086 (S0, S1, S2) and the high order address lines (ADLF+ - ADL13+) latched at ALE+ time. The output of the latch is used by the Multibus Master decode logic. This latch is always enabled so that the decode logic can work even if the DMA logic has the local bus.

The 8288 (p1) is used to decode the 8086 status into the command lines for the local bus and for other bus control functions. Primarily it generates the following important signals:

PMR-	Memory Read
MW-	Memory Write
IOR-	I/O Read
IOW-	I/O Write
INTA-	Interrupt Acknowledge

PMR- is inverted (15A-6) and driven by an open-collector driver (15B-2) to make the system memory read signal, MR-. These five signals are the five command lines for the local bus. In addition to these, the 8288 generates ALE+ to indicate when the 8086 multiplexed address/data lines (ADO+ - AD13+) contain status and address information as opposed to data. The DEN+ line from the 8288 indicates that the AD+ bus has data on it. The DT+ signal indicates that the data is going from the 8086 to I/O or memory. The PWAIT- signal going into the 8288 causes it to tri-state the system command lines (described above). PWAIT- is activated when the 8086 does not have

control over the local bus by gate 8B-6. PWAIT- goes active when HLDA is active (indicating DMA activity) or when MBPNDG- is active (indicating that the 8086 is waiting to get on Multibus).

The IDLE+ signal is used to determine when the 8086 is doing internal cycles and not using the local-bus. The S0- and S1- lines are both high when the 8086 is halted or doing internal cycles (passive). 8B-8 (p1zC4) indicates this condition. The pair of flip-flops at 16B are used to delay the status at least one PCLK+ to insure that the last active cycle has completed. The output of the flip-flop delay is the signal IDLE+. IDLE+ is cleared asynchronously at 16B-13 if the status ever goes to one of the active combinations. It is also synchronously cleared at 16B-1 (plus one PCLK+) by the signal PLOCK- from the 8086. This signal indicates that a Lock cycle, a non-idle state, is in progress.

Various cycles are also decoded by the logic of page 1. 8E-3 (p1zC2) is IOCYC-; when active it indicates that an I/O cycle is in progress. 8E-8 is active when a Multibus cycle is requested and during Locked cycles. MBREQ+ at 8E-6 (p1zB2) includes INTA- to combine all reasons for making Multibus Requests. INTA- uses the Multibus to do interrupt acknowledges in case there are slave 8259As on the Multibus. LOCK, 8E9 (p1zB2), acquires the Multibus so that mutually exclusive access can be made to shared memory.

The timing of the MR- signal is critical because it determines the speed required of the RAM chips. Since the 8288 is slow in generating MR-, the MR optimizer circuit (p1zB3) works in parallel with the 8288 to make MR- as early as possible. The MR optimizer output and the 8288 MR- are wire-ORed by open collector inverters 15B-12 and 15B-2 (p1zB2). Flip-flop 8D-5 is set by ALE+ during the second half of 8086 state T1 and the first half of T2. 7D-8 combines this timing signal (T1.5+) with the 8086 status line combination that indicates a memory-read cycle and, at the next falling PCLK+, flip-flop 8D-8 causes the signal OPTIM+ to become active. This causes open-collector driver 15B-12 to assert MR-. At the next falling PCLK+, OPTIM+ will go inactive but the 8288 by then has asserted MR- through driver 15B-2.

The 8087

The 8087 (p 1) connects almost directly in parallel with the 8086. One difference is that the interrupt pin is an output on the 8087. It is asserted when the 8087 wishes to interrupt the 8086. It is tied, through an inverting open-collector driver at 15B-6 (p6zB7), to the level-4 interrupt input of the 8259A interrupt controller. The other difference is that the TEST-input pin of the 8086 (16D-23) corresponds to the BUSY- (16F-23) output of the 8087). They are used to synchronize the two processors. The 8087 uses the other resources of the CPU board in the same way as the 8086.

The RAM Controller

This logic block generates the control signals for the RAM chips and some timing signals for the 8086 and DMA chip.

Crystal Y1 and inverters 15A-8 and 15A-10 and R9, R10, C3, C7, and C8 (p3zD7) constitute a 20 MHz oscillator. This is the source for the memory and DMA clocks. Counters 12A and 13A (p3) are cascaded to give the two memory clocks, MEMCLK- (10 MHz) and 78KHZ+. 78KHZ+ is used primarily to start refresh cycles. MEMCLK- is inverted at 15A-2 to give MEMCLK+ and buffered at 14C-3 (p3zD2) to give BMEMCLK+. BMEMCLK+ is used on the video and I/O boards as a general purpose 10 MHz clock.

Refresh is started when a falling transition of 78KHZ+ causes flip-flop 14B-5 (p3zC6) to set. This can be inhibited for test purposes by grounding test point 4. The refresh request at 13B-5 is then synchronized to the DMA chip clock at 13B-5. This generates RFPNDG- which is used in the DMA logic (to be described later) to cause wait states if the 8086 uses memory during refresh. The complement of RFPNDG- at 14B-5 (p3zB6) is clocked into 14B-12 at the next MEMCLK+ to generate RFTIME- at 14B-8. This is done to allow RFPNDG- to take effect before RFTIME- inhibits new memory accesses at 9C-1 (p3zC5). When RFTIME- is asserted and no memory cycle is in progress the signal at 10B-13 goes high and at the next MEMCLK+ flip-flop 10A-9 (p3zC2) goes active. At the following MEMCLK+, the RFGO+ (refresh going) flip-flop 11A-9 (p3zB2) is set. RFGO- is buffered at 14C-7 to go to the memory arrays on the I/O Memory and RAM Expansion boards. RFGO- causes ANYCYC+ to go active at 7C-

11 (p3zA7). This signal propagates through delay lines 8A and 9A. 40 nanoseconds later (allowing the memory arrays to use RFG0- to switch the RAM address lines to the refresh address) the gate at 7B-13 (p3zA2) causes RFRAS+ to go active. Next, the gate at 7B-1 causes RARAS- to go active. RARAS- goes to the two RAM arrays and becomes the RAS- (Row Address Strobe) for the RAM chips. 220 nanoseconds after RARAS- goes active the signal ENDCYC+ at delay line pin 8A-4 goes active. This in turn causes gate 8B-3 to go high. At the next MEMCLK+ the RFDONE+ flip-flop at 11A-5 (p3zB4) becomes set. RFDONE- is buffered at 14C-5 to go to the RAM arrays to increment the refresh address for the next refresh cycle. RFDONE+ causes the RFG0+ flip-flop to be cleared at the next MEMCLK+. RFDONE- also clears the three refresh flip-flops already described (14B1, 14B13, and 13B-15).

Memory Read Cycles

Memory Reads are started by the activation of MR-, which causes CYCLREQ+ to go active at 7C-3 (p3zC7). When CYCLEREQ+ is active, no refresh is in progress (RFTIME- inactive), the local bus address points to present RAM (PRESELECT+ active), and the logic is not in the ending phase of the previous cycle (ENDCYC- inactive), gate 9C-6 (p3zC5) asserts START-. R20, R21, and C12 guarantee that START- is of sufficient duration to set Schmitt trigger flip-flop 8C-8 (p3zC4), ENMCYC+. ENMCYC+ through gate 9B-3 causes CYCINPROG+ to go active. CYCINPROG+, at 10B-11 (p3zC3) prevents a refresh cycle from starting when a read or write is in progress. At 1B-8 (p3zC2), CYCINPROG+ is stretched by a signal from the delay line to make INPROG+. INPROG+ is used by the DMA logic to determine when its memory cycles are complete. ENMCYC+ is buffered at 2C-10 to make BENMCYC+. This signal is used on the I/O Memory board for communications DMA. ENMCYC+ is also used in the DMA wait state logic and in the READY timing logic. ENMCYC+ also activates RARAS- at 7B-1 (p3zA3) which goes to the RAM arrays to make the RAS signals for the RAM chips. The inverse of ENMCYC+, ENMCYC-, generates ANYCYC+ at 7C-11. This signal propagates through the delay lines 8A and 9A to make the various RAM timing and control signals. 20 nanoseconds after ANYCYC+ the buffer at 145C-9 (p3zA5) activates COLMPX- to switch the RAM chip address lines from row address to column

address. 60 nanoseconds later, CASTIME- at 9A-6 causes the decoder at 11G (p7zC3) to generate the CAS- (Column Address Strobe) for the particular bank of RAMs selected by the address bus. 80 nanoseconds after CASTIME- comes PRACKTIME-. PRACKTIME- is ANDed with ENMCYC to make PRACK (Predicted RAM Acknowledge), which indicates to the READY timing logic that the RAM cycle is about to complete. 240 nanoseconds after ANYCYC+, ENDCYC+ is asserted. It is used to terminate refreshes and normal memory accesses at 11A-2 and 9B-4, respectively. 9B-6 which resets the ENMCYC+ flip-flop, 8C-8 (p3zC4), is activated when ENDCYC+ is active and there is no longer a CYCLEREQ+ from 7C-3. ENDCYC+ is also used at 9B-1 (p3zC3) to prevent a refresh from starting while an old cycle is still in progress. Gate 7B-4 (p3zB1) combines ENDCYC- and a signal (NORF-) that indicates that a refresh cycle is not in progress to generate ENEND+ (Enabled end). ENEND+ is used in DMA wait state timing logic. 100 nanoseconds after ENDCYC+, RACK+ becomes active. It is used to form XACK- (Transfer Acknowledge) in the Multibus interface logic.

Memory Write

Memory Write (MW-) operate in the same way as memory read except for the following differences: MW- can be inhibited at gate 6D-11 (p3zC8) for memory protection (MWINH+ active) or delayed at 2B-5 by WRDLY-. WRDLY- comes from the DMA logic to delay the MW- signal during communications DMA. The write enable signals RALOWE- (RAM array low-bank write enable) and RAHIWE- (high-bank write enable) are generated by gates 10C-6 and 10C-3 (p7zD6), respectively. RALOWE- is active for non-read (write) memory cycles when the low-order address bit (LAO+) is low. RAHIWE- is active for non-read cycles when LAO+ is high (pointing to an odd, or high bank) or during 16-bit accesses, indicated by LBHEN+ active (local bus high enable). Note that the RAM chip write enable pins are only significant when CAS- is active.

Column Address Strobe (CAS) Generation

The 8 CAS- lines for the 8 possible RAM banks are generated by decoder 11G (p7zD3) at CASTIME-, as described above, during ENMCYC- (Enabled memory cycles; that is, non-refresh cycles). The selection of which bank to activate is made by PROM 12G based on the high-order 5 bits of the local

address bus and the settings of three switches, SW4 bits 2, 3, and 4.

Details on the decode PROM and resultant addressing are given below under "References".

Parity Logic

The parity logic is used to constantly check the reliability of the memory. Each byte in the memory array has an extra memory bit (the parity bit) associated with it. Whenever a byte is written, the ninth bit is written such that the total number of 1 bits is even. When the byte is read, if the total number of 1 bits in the 9-bit "byte" is no longer even, an error is flagged. This is the normal case. For testing purposes, a bit in the IOCR (port 56 on the I/O memory board) can be set to change the parity system to odd instead of even. When a parity error is detected, the error address and some status information is saved in a register for software reading and a non-maskable interrupt will be generated if enabled by the IOCR.

Parity Writing, Odd Byte (Even Byte in Brackets)

Because MR- is inactive, 11C-8 (p2zC7) causes CHECKP- to be inactive. When CHECKP- is inactive, 10C-8 (10C-11) forces 13F-1 (13G-4) to 1. The other 8 input bits of the parity generator at 13F (13G) are MEM8+ - MEMF+ (MEMO+ - MEM7+). Thus the odd sum output, 13F-6 (13G-6), is high if MEM8+ - MEMF+ (MEMO+ - MEM7+) have an even number of 1 bits). The parity bit written in this case should be a 0 (to maintain an even number of 1 bits). If the odd parity bit of the IOCR is set, the ODDP+ line will be active. In that case, the output of the parity generator will be complemented by gate 12B-8 (12B-6). Again, because CHECKP- is inactive, the inverting tri-state buffer at 12C-16 (12C-18) inverts the parity information from 12B-8 (12B-6) and drives HIP+ (LOP+), the signal that connects to the data pins of all the odd (high) (even, low) byte parity RAM chips. This bit is written into the parity RAM chips in the same fashion as the 8 data bits are written to the appropriate data chips.

Parity reading and checking, odd byte (even byte)

When MR+ (Memory Read) is active, PRESELECT+ (RAM is present and selected) is active, and PROMCS+

(PROM chip select) is inactive, 11C-8, CHECKP- (Check Parity), goes active. Since it is active, the HIP+ (LOP+) signal carrying the parity bit from the selected RAM parity chip and the 8 data bits being read (on MEM8+ - MEMF+ (MEM0+ - MEM7+)) are summed by parity generator 13F (13G). The signal at 13F-6 (13G-6) is low if the sum is even (normally correct). Again, if the odd parity bit of the IOCR is set (ODDP+ is active), the gate at 12B-8 (12B-6) will invert the sum. At the end of the read, when MR+ goes inactive, the sum bit from 12B-8 (12B-6) will be clocked into parity error flip-flop 13C-9 (13C-5). If the sum bit was 1 (wrong), the parity error flip-flop will be set.

When either parity error flip-flop is set, 9D-6 (p2zB6) causes PERINT+ to go active. When it goes active the registers at 12D, 12E, and 11F (p2) trap the current address on LA0+ - LA13+ (local address bus) and the status of the LBHEN+, HLDA+, and SLAVE+ signals. These bits and the status of the high and low parity error flip-flops can be read through ports 52 and 54.

Interrupt Control Logic

The interrupt control logic is based on the 8259A programmable interrupt controller (p6). It takes 8 input interrupt signals from devices or other 8259As and with masking provides an interrupt signal and interrupt vectors to the 8086.

The 8259A is programmed by passing information between it and the 8086 on the I/O bus, IOB0+ - IOB7+. The I/O Memory board decodes the I/O address of the 8259A and asserts PICCS- (Programmable Interrupt Controller Chip Select). It is used with LA1+ to specify which register within the 8259A is being referenced. The IOR- (I/O read) and IOW- (I/O write) strobes are used to enable the transfer of information during programming.

The eight interrupt signals, INTRO- - INTR7-, after buffering, go into the 8259A. When an active interrupt level is enabled by programming, the 8259A will assert INTR+ to interrupt the 8086. If the interrupt is enabled within the 8086, the 8086 will go into an Interrupt Acknowledge sequence. In Interrupt Acknowledge, two INTA- pulses are issued by the 8086/8288. The first one is used within the 8259A to decide which is the highest priority interrupt. The

second one is used to send the vector to the 8086.

The 8259A on the processor board is called the "master" 8259A. For large interrupt systems, "slave" 8259As can be used in Multibus cards. A "slave" 8259A can drive one of the INTR lines on the Multibus backplane with its INTR pin, and can thereby expand one of INTR lines of the master 8259A into eight individual interrupt lines.

All 8259As must receive all INTA- pulses for their internal priority logic to work. Thus, in case there are slave 8259As, Multibus must always be acquired. If a slave 8259A has generated an interrupt to the master 8259A, the interrupt vector comes from the slave 8259A. The second INTA- must cause the interrupt vector to come from the master 8259A if there is not a slave 8259A on the interrupting level, and from the slave 8259A if there is. When a slave 8259A is to respond, the master 8259A puts the binary number that corresponds to its INTRn- that is causing the interrupt on its cascade lines. This is used to address the proper slave 8259A. The slave 8259A is programmed with its cascade number at initialization and then recognizes its address when the master 8259A puts it on its cascade lines. The cascade number is passed from the master 8259A to slave 8259As on Multibus address lines ADR8- - ADRA-.

The eight interrupt signals, INTR0- - INTR7-, are buffered by the inverting buffer at 3F (p6zB7). Notice that the 8087 numeric data processor can interrupt by asserting INTR4-. After buffering, the eight interrupt signals go into the 8259A at 5F. When that interrupt level is enabled by programming, the 8259A will assert INTR+ to interrupt the 8086. If interrupts are enabled within the 8086, the 8086/8288 (p1) combination will issue the two INTA- pulses.

In order to get consecutive Multibus cycles, the 8086 asserts PLOCK- (processor lock) just prior to the first INTA- and makes it inactive just prior to the second INTA-. This causes the 8289 (p4) to not release the Multibus until the second INTA- is complete.

In addition, the 8288 issues an MCE+ (master cascade enable) just prior to each INTA- pulse. The second MCE+ pulse indicates the time that the cascade lines of the master 8259A are valid.

Gate 6E-3 (p6zC7) ANDs MCE+ and LOCK+ to isolate the second MCE+ pulse, giving MCE2-. The AND of INTA+ and LOCK+ isolate the first INTA- pulse at 6E-11 (p6zC7).

Through logic described in the Multibus Master mode section, INTA- causes the 8289 to acquire control of the Multibus. When it is acquired, signal MBAEN- is asserted by the 8289. This causes 6D-8 (p6zA7) to go active, issuing an interrupt acknowledge strobe to the 8259A at pin 5F-26. Next, the 8259A causes its cascade outputs (CAS0+ - CAS2+) to be asserted with the number corresponding to the slave 8259A, if a slave 8259A is generating the highest priority interrupt.

During MCE2-, driver 3E-7, 5, and 3 (p6zA5) causes the cascade number from the 8259A to be driven onto the 8086 AD bus signals, AD8+, AD9+, and ADA+. Simultaneously, the 8288 generates an ALE+ into the 8086 address latch/driver, 15D (p7zB6), and onto the local address bus (LA8+ - LAA+). The Multibus address driver, 7G, drives the cascade number from the local bus to the Multibus address bits ADR8-, ADR9-, and ADRA-. The slave 8259As get the cascade number from those Multibus lines.

Whenever the 8259A wants to output on the I/O bus (IOB0+ - IOB7+), it asserts its DEN59- output, 5F-16. The processor board uses DEN59- to generate the signal INTADEN+ at 4D-10 (p6zA6) to indicate that the vector is being sent from the master 8259A at 5F. As described in the Multibus Master mode section, DEN59- is used at 4E-11 (p5zB3) to make MBINTA+ (Multibus interrupt acknowledge) and enable the Multibus drivers to carry the vector from the Multibus (and slave 8259A) to the local (MEM) bus. INTADEN+ is used at 4D-13 (p5zB2) to cause the I/O bus (IOB0+ - IOB7+) to drive the MEM bus with the vector when it comes from the master 8259A.

The gate 7D-12 (MASACK+) (p6zC5) generates XACK- (transfer acknowledge) for the INTA pulses. The first INTA- pulse causes 6E-11 and 7D-2 (p6zB5) to go active, issuing an immediate XACK-. The second INTA, if it does not pertain to a slave 8259A, causes INTADEN+ at 4D-10 (p6zA6) to go active. 300 nanoseconds after the INTA starts, the signal 300NS+ goes active, causing 6C-6 and 7D-1 (p6zB5) to go active, allowing for a 300 nanosecond access time before generating XACK-.

Direct Memory Access (DMA)

The DMA logic centers around the 8237 DMA controller chip. The DMA chip has 4 channels to support 4 different requestors; Multibus Slave mode, communications, disk, and video. The channels are assigned as follows:

<u>Channel</u>	<u>Priority</u>	<u>Device</u>
0	highest	Multibus Slave mode
1	2nd highest	Communications
2	3rd highest	Disk
3	lowest	Video

Each channel has a unique request pin and acknowledge pin.

The 8237 is programmed by the 8086 with memory addresses, transfer length and mode information. When the DMA logic is not transferring (HLDA+ inactive), the 8237 can be programmed by the 8086. Information is passed between the 8237 and the 8086 on the I/O bus, IOB0+ - IOB7+. The register that the 8086 is addressing within the 8237 is specified on the bidirectional address lines of the 8237, DMAA0+ - DMAA3+. The DMA chip select signal from the I/O Memory board indicates that the current I/O address on the local bus refers to an 8237 register. When DMACS- is active it causes the IOR- (I/O read) and IOW- (I/O write) signals of the local bus to be driven to the 8237 pins 1 and 2 by asserting the driver enable at 8F-19 (p5zC4).

DMCLK+ is generated from the output of the 20 MHz oscillator by counter 14A (p3zD6). The counter divides by 4, giving a 5 MHz square wave if switch 1 bit 1 (p3zD5) is open. If the switch is closed, the counter divides by 5, giving a 4 MHz DMCLK+.

When a request pin is asserted and that channel is programmed in the 8237 to be enabled, a DMA cycle starts. The 8237 first asserts its hold request (HRQ+) pin at 6F-10 (p5). This causes flip-flop 5B-5 to be set after a one-clock delay (required by the 8237). When 5B-5 is asserted and BUSAVAIL+ (local bus available) is asserted by 4B-8 and the bus is not locked (LOCKED- inactive) then 5C-6 goes active setting the Hold Acknowledge (HLDA+) flip-flop, 8C-3 (p5zA4).

LOCKED- is active during instructions with LOCK prefixes and inhibits DMA starting during "locked" instructions. BUSAVAIL+ is active if the 8086 is waiting for Multibus master status (MBPNDG-) or if the 8086 is not using the bus. When the 8086 is not using the bus, the signal at 10B-10 goes low. This is determined by IDLE+ active, indicating that the 8086 is doing internal cycles.

The RC combination R18, R19, and C11 and the Schmitt trigger inputs of 8C insure that the HLDA+ flip-flop (p5zA4) will not go temporarily active (non-bistable operation).

Once HLDA+ is active, gate 8B-6 (p1zA6) causes 8086 commands to be inhibited by driving the 8288 pin at 9E-6 (p1zA4) high. This holds off the 8086 until the DMA cycle is completed. Furthermore, HLDA+ causes 5C-12 (p5zA4), LOCMaster-, to go inactive so that the 8284 won't give the 8086 a READY signal.

HLDA+ is latched if a parity error occurs by the parity error latch (port 54) at 11F-7 (p2zA6). This is used to determine if the parity error happened during a DMA transfer.

HLDA+ is used at 3A-6 (p4zB5) to inhibit going into Master mode during a DMA transfer.

PROM access is inhibited during DMA transfers by gate 7B-10 (p4zB7).

The open-collector gate at 4A-11 (p5zD4) causes the memory protect circuitry to be overridden by pulling down MWINH+ (Memory Write inhibit) unconditionally.

The gate 7C-6 (p5zA3) inhibits 8086 I/O address decoding by insuring IOCS- is inactive during DMA transfers.

At 3B-9 (p5zA5), HLDA- inhibits the 8086 AD bus from driving the MEM bus during DMA by forcing ADTOMEM- inactive.

The three tri-state latches at 11E, 15D, and 15E (p7) that normally drive the local address bus (LA0+ - LA13+) with the address from the 8086 are disabled during DMA by HLDA+.

Finally as long as the HLDA+ flip-flop is set and the delayed hold (DHOLD-) flip-flop is set at 5B-

5 (p5zB7), the gate at 3G-13 (p5zD6) gives a Hold Acknowledge to the 8237 at 6F-7.

When 6F-7 goes active, the 8237 takes control of the local bus. First, if the high-order eight bits of DMA address are different from those of the preceding cycle, it will drive the new eight address bits on IOB0+ - IOB7+ and pulse its ADSTB+ (address strobe) output (6F-8). This causes the address to be latched into chip 5G (p7zC7). Next it asserts its "address enable" pin 6F-9. If the DMA cycle is not for Multibus Slave mode then DMAEN- (Direct Memory address enable) will go active at 6E-6. DMAEN- also enables the DMA command drivers at 8F-1 (p5zC4) so that the memory read and write and I/O read and write commands from the 8237 drive IOR-, IOW-, MR-, and MW-. If the DMA cycle is for Multibus Slave mode, then the address and commands come from Multibus as already described.

The 8237 is designed to generate only 16 bits of address and the local bus has a 20-bit address space. The following is done for each channel to provide the extra 4 bits.

<u>Channel</u>	<u>Device</u>	<u>LA0+, LA11+ - LA13+</u>
0	Multibus	Driven from the Multibus Interface
1	Communications	All 0, driven by 2E-3 (p7zD7), 5, 7, and 9
2	Disk	LA0+ is 0 and LA11+-LA13+ come from the Extended Address Register (EAR), driven by 2E-12, 14, 16, 18
3	Video	All 0, driven by 2E-3, 5, 7, and 9.

This means that DMA transfer for communications, disk, and video must be on even boundaries, and transfers to and from communications and video must be in the first 128K of RAM.

The Extended Address Register (EAR) is a 3-bit counter provided so that the disk can access any address in the 1-megabyte address space. The EAR, at 2D (p8zC5), is loaded by an active-low strobe coming from 6D-6 when software issues an Output instruction (IOW-) to I/O address 68 (DMACTCS-). The bit correspondence is as follows:

<u>Bit from 8086 register AX</u>	<u>MEM bus bit</u>	<u>EAR bit</u>	<u>Local Address bit</u>
0	0	DMCTO+	LA11+
1	1	DMCT1+	LA12+
2	2	DMCT2+	LA13+

Hardware causes the EAR to increment when a 128K boundary is crossed during disk DMA. QBOUND- is asserted by 98B-8 (p8zB7) if LA1+ - LA10+ are all 1: This enables the EAR to increment when gate 11C-12 (p8zC7) goes high, indicating that the read portion (INCMD+) of a disk DMA (DISKACK+) cycle is completed.

The DISKACK-, VIDACK-, SIOACK-, and SLAVE+ signals indicate that the 8237 is acknowledging disk, video, communications, and Multibus channels, respectively. These are used to enable address buffers as just described and they go to the individual logic blocks for controlling those devices to indicate DMA is in progress. If the count register is expired for the channel being acknowledged, the signal EOP- (End of Process) is asserted at 6F-36.

Multibus Slave READY Generation

For Multibus Slave operation, the remainder of DMA cycle is controlled by the Multibus interface. The Slave operation DMA cycle is terminated (as are all DMA cycles) by the RDY (ready) input to the 8237 (6F-6) going active. RDY is generated at 4C-8 (p5zC6) when MEMRDY+ (Memory ready) and IORDY+ (I/O ready) are both active. For Slave operation, 3G-8 (p5zC7) is active, forcing MEMRDY+ unconditionally asserted. IORDY+ comes from the shift register at 3C-12 (p5zC1). Since it is not a communications DMA cycle (SIOASCK- not asserted), gate 3G-12 puts register 3C in the parallel enable mode. As long as SLAVERQ+ (see the Multibus Slave mode section) is asserted, 1D-11 will continue to assert SLAVE+. This causes the D input to 3C to be low at 1C-10. As soon as SLAVERQ+ terminates, SLAVE+ drops at 1D-11 causing 3C-7 to go high. At the next DMCLK+, IORDY+ goes active causing the DMA RDY signal to go active at 6F-6. This causes the DMA cycle to terminate.

Non-Multibus READY Generation

DMA cycles terminate when the DMA ready signal from gate 4C-8 goes active. 4C-8 ANDs the two conditions for DMA ready (RDY), IORDY+ and MEMRDY+. MEMRDY+ goes inactive when a refresh cycle (RDPNDG-) is pending or in progress and a non-refresh memory cycle (ENMCYC+) is not already going, causing 10A-4 (p5zC7) to preset the MEMRDY- flip-flop. At the end of the DMA memory cycle, ENEND+ will be asserted by the RAM control clearing the MEMRDY- flip-flop by causing 3A-1 to go low. This causes MEMRDY+ to go active. In case the address programmed into the DMA chip did not point to present memory, the timeout (TO+) signal will go active causing gate 3G-10 (p5zC7) to force MEMRDY+ active.

IORDY+ is generated by the shift register at 3C (p5zC1).

The shift register is put in the shift mode by driving 3C-9 to a high state.

This is done (by gate 3G) only during communications DMA acknowledges (SIOACK-) that are doing memory read and I/O write (IOW-) commands. In that case, IORDY+ goes true four DMCLK+ cycles after ENMCYC+ (non-refresh memory cycle) goes active, indicating that the memory cycle is in progress. This is done because the SIO I/O write can't begin until the data is available from memory.

For all other cases, shift register 3C is in the parallel enable mode (3C-9 is low). 1C-1 goes high, causing IORDY+ to go inactive when 1D-8 goes low in a communications DMA cycle (SIOACK-) or when WAIT- is asserted. WAIT- is used by the video control board to cause one DMA wait state for some of its accesses. As soon as the write part of the DMA cycle starts (MW- or IOW- active), 3d-11 goes high, causing 1C-1 to go low. One DMA clock after 1C-1 goes low (causing 3C-7 to go high), IORDY+ is asserted at 3C-12. This causes the DMA Ready pin to be asserted by gate 4C-8 (as long as MEMRDY+ is active). Note that since neither SIOACK- nor WAIT- goes active during disk cycles, 3C-7 never goes low and IORDY+ is asserted throughout the cycle.

Gate 2B-8 causes memory write to be delayed when IORDY- is not active. This is done so that the data can come from the I/O device and get to the

memory before the memory write occurs. 2B-9 inhibits this delay during Multibus slave operation.

The Hold Acknowledge flip-flop (HLDA+) (p5zA4) is reset by by gate 1B when no cycle in progress (INPROG+ inactive) and the Hold Request (HRQ+) signal from the 8237 is no longer active.

Multibus Interface

The Multibus interface works in three entirely separate modes; Independent, Master, and Slave. In the Independent mode, the interface separates Multibus from the local bus (CT bus) and the two operate entirely independently and asynchronously. In the Master mode, the local bus controls Multibus, driving its command and address lines. In the Slave mode, Multibus controls the local bus and Multibus drives the command and address lines of the local bus. The control logic for Master and Slave modes are entirely separate.

Multibus Master Mode

The Multibus Master mode centers around the Multibus arbiter chip, the 8289. It is responsible for acquiring Multibus when the local logic requires it. There are two times when the Multibus interface goes into the Master mode: (1) for explicit accesses by the 8086 to memory or I/O devices not on the local bus, or (2) for interrupt acknowledge. Interrupt Acknowledge goes to the Multibus in case there are slave 8259As in Multibus hardware. Furthermore, for any instruction with a LOCK prefix, the Multibus is acquired, although it does not necessarily go into Master mode. This is done to be sure that if another Multibus master has locked the Multibus to get consecutive cycles in the local RAM, the 8086 instruction with the LOCK prefix won't begin until the other master unlocks Multibus. This is used for passing semaphores between different Multibus masters.

8289/Multibus Handshake

The details of Multibus handshakes can be found in the Multibus specification, IEEE 796. Here is a brief description of each handshake signal:

<u>Signal</u>	<u>Meaning</u>
BREQ-	Bus Request, one per Multibus slot (output). Indicates that a particular Multibus slot wants Multibus Master status.
CBRQ-	Common Bus Request, one per system (bidirectional). Indicates that some Multibus slot that is not Multibus Master wants to be Master.
BPRN-	Bus Priority In, one per Multibus slot (input). Indicates that the Multibus slot has priority and can acquire Master status at the next BCLK-.
BPRO-	Bus Priority Out, one per Multibus slot (output). Indicates that the BPRN-signal of the Multibus slot is active and the slot does not want Multibus Master status.
BUSY-	Bus Busy, one per system (bidirectional). Indicates that some board is Multibus Master and is using the bus.
BCLK-	The lock used for bus priority transitions (output on one Multibus slot, input on all slots).
CCLK-	A general purpose clock (output on one Multibus slot, input on all slots).

The 8289 does the following to get Multibus Master status:

- 1) Asserts CBRQ- and BREQ-, and makes BRPO- inactive, indicating it wants to be Master.
- 2) At the next BCLK-, if BUSY- is inactive and BPRN- is active, the 8289 has acquired Master status. It asserts BUSY- and inactivates CBREQ-.
- 3) When the 8289 is done with the bus, it stops asserting BUSY-, and will no longer force BPRO- inactive. If BPRN- is active, BPRO- will go active.

Acquiring Multibus

The S0 - S2 (16D-26, 27, 28) signals from the 8086 are monitored by the 8289 (p4zB3). When they indicate a bus cycle (Memory, I/O or Interrupt Acknowledge) and pin 3 of the 8289 is ac-

tive, the 8289 will go through a Multibus arbitration handshake to acquire control of Multibus (become Multibus master). When it does, it asserts pin 13, AEN89-.

As just noted, pin 3 of the 8289, the signal ADORLOCK+ (Multibus addressed or Lock prefix), must be active for the 8289 to acquire Multibus. It is caused to go active at 1B-11 (p4zA5) by one of two reasons: (1) SLOCK+ at 1B-12 (synchronized Lock) is indicating that a LOCK prefix or locked Interrupt Acknowledge cycle is in progress, or (2) MBADR- is active, indicating that the present address from the 8086 refers to Multibus memory or I/O. MBADR- is formed at gate 3D-6 (p4zA5) by ANDing the conditions local memory not addressed (LOCMEM-) and local I/O not addressed (LOCIO-).

LOCIO- is generated at gate 4D (p4zA6) by ANDing IOCYC- (I/O cycle active) and the proper address range. The proper address range is determined normally (jumper 1-2 in place) by address bit F (ADLF+) from the 8086 low. That is, addresses 0-7FFF refer to LOCAL I/O devices and addresses 8000-FFFF refer to Multibus I/O devices.

A factory jumper option can be made by removing jumper 1-2 and installing jumper 3-2 (p4zA6). This causes an I/O address to be considered local if the MBIO bit of the IOCR (port 56 bit C on the I/O Memory board) is inactive or if address bit F is high. In other words, with this option, in order to address Multibus for I/O, it is necessary to set the MBIO bit in the IOCR and to address in the range 0-7FFF, and when MBIO is set referring to local I/O devices requires an address in the range 8000-FFFF. For instance, the printer is addressed as port 8050 (hexidecimal) instead of the normal 50. Note that with this option set the software can always refer to local ports as 80xx so that the setting of the MBIO bit becomes inconsequential for referring to local ports.

LOCMEM- (local memory addressed) is generated at gate 9D-8 (p4zA5) by ANDing the latched S2 status line from the 8086 (S2L-), indicating a memory cycle is in progress and the signal MEMWINDO- indicating that the current address from the 8086 refers to local bus memory. MEMWINDO+ is generated by ORing several conditions with gate 10D-8 (p4zA6).

Conditions causing MEMWINDO+ active (local bus access for memory) are:

- 1) Switch 6 bit 3 (pins 3 and 6) closed (all memory accesses are to local bus), or
- 2) ADL13+ low (all memory accesses in the first 512K of the addressing range, 0-7FFFF, refer to local memory), or
- 3) Switch 4 bit 1 closed and ADL12+ low (all memory accesses in the first 768K of the addressing range, 0-BFFFF, refer to local memory), or
- 4) Switch 6 bit 1 closed and LOCPR0M- active (Bootstrap PROM addressed), or
- 5) Switch 6 bit 2 closed and address on ADLF+ - ADL13+ in the top 160K of the address range. This reserves space for the ROM expansion board and the Bootstrap PROM.

When ADORLOCK+ is active and AEN89- is not, MBPNDG- (Multibus Pending) is asserted at 4B-11 (p5zA7) indicating that the 8289 is in the process of acquiring Multibus but has not finished. This causes the 8288 to disable its command lines (at 9E-6) and thereby not assert them until Multibus is acquired.

Multibus Master Mode Operation

The MASTER+ flip-flop indicates that the Multibus interface is in Master mode. In Master mode, the local bus address and command lines drive the Multibus. After Multibus has been acquired, the signal AEN89- is asserted by the 8289. If the 8086 is addressing Multibus, MBADR- will be active, causing MBAEN- (Multibus Address Enable) to go active at 1B-3 (p4zA3). MBAEN- will cause flip-flop 2A-5 (p4zB6) to be set at the next MEMCLK+. If no HLDA- is present (not doing DMA) then at the next MEMCLK+ the MASTER+ flip-flop, 2A-9 (p4zB5), will be set. This pair of flip-flops allows the Multibus address to be driven from the local bus (by MBAEN-) before the board goes into Master mode. The gate at 3A-4 insures that we will not go into Master mode if the DMA has control of the local bus. Note that if a LOCK instruction is used that does not address a Multibus device or memory, MBADR- will not be active and will inhibit MBAEN-. When MBAEN- goes inactive, that is if the 8289 loses control of

the Multibus, the MASTER+ flip-flop will be cleared at 2A-13 (p4zB5). AEN89- is also used in the DMA logic to determine when the 8086 is waiting for the Multibus, and therefore the local bus is free. MBAEN- is used in the interrupt logic to determine that the local bus is the Multibus master and an Interrupt Acknowledge can complete.

Command, Address and Data bus control during Master Mode

MASTER+ causes the command transeiver at 2G (p4zC4) to be enabled by gate 3A-10. The direction of the transfer having already been established by the flip-flop at 2A (p4zB6). This allows the commands on the local bus to be driven onto the Multibus command lines.

MASTER- causes ONBUS- to be asserted by enabling the tri-state inverter at 2F-16 (p4zD6). ONBUS- then enables the Multibus address drivers, 7G, 8G, and 4G (p7). Multibus address lines AD12- and AD13- are driven by the same tri-state inverter as ONBUS- at 2F. Since the input to the tri-state inverter for AD13- is grounded at 2F-2, all Multibus addresses coming from the local bus are in the low 512K of the Multibus address range. Note that AD13- is completely independent of the corresponding local address bus bit LA13-. For AD12- there are two switch selectable choices for Multibus address range. If switch 3 bit 1 is closed then AD12- is driven from LA12+ through the tri-state inverter 2F-12. If switch 3 bit 2 is closed, then all Multibus addresses coming from the local bus are in the low 256K, independent of local address bus bit LA12+.

ONBUS- causes BUSEN+ to be asserted by gate 1C-13 (p8zC6) (the INHIBIT+ signal is described in the Multibus Slave Mode section). This causes the proper data bus drivers to be enabled depending on whether a byte-swap access has to occur. If LA0+ is asserted (an odd byte is being addressed), the Multibus slave addressed will place its data on the low-order 8 bits of the Multibus (DAT0- - DAT7-). The 8086 expects the data to be placed on its high-order 8 bits (AD8+ - ADF+). Therefore, a byte swap must occur. If LA0+ is not asserted then the addressed byte or word must not be swapped. Furthermore, during Interrupt Acknowledge, the byte must not be swapped independent of what is on the local address bus. The signal SWAP- is asserted whenever BUSEN+ is

active and LA0+ is asserted (odd byte addressed) and INTA- is not active. This causes the inverting transceiver at 10F to be enabled which connects the Multibus DAT0- - DAT7- lines to be connected to MEM8+ to MEMF+. Whenever BUSEN+ is active and SWAP- is not, NOSWAP- is asserted at 2B-3 (p8zC4). This enables transceivers 9G and 10G (p8) to connect the Multibus data bus to the MEM bus without swapping.

The direction of the data transfer is established by the signal MEMTOMB+, which, when active, indicates that the MEM bus will drive the Multibus data bus (DAT0- to DATF-). MEMTOMB+ is generated by gate 3B-6 (p5zB2) and is determined by gate 3B-3 for Slave mode (to be described) and 3B-11 for Master mode. 3B-11 is asserted low, causing MEMTOMB+ to be asserted, when MASTER+ is active and MBINTA+ is inactive (not an Interrupt Acknowledge cycle) and INCMD+ is inactive (the command on the local bus is not an input command). INCMD+ is generated by gate 3D-8 (p5zB3) and is the OR of IOR- asserted or MR- asserted. MBINTA+ is asserted by 4E-11 (p5zB3) when INTA+ is active (Interrupt Acknowledge) and the signal DEN59- from the 8259A interrupt controller is inactive, indicating that it is not generating the vector and some slave 8259A on Multibus should.

LOCK access to Multibus

As mentioned under "Multibus Master Mode", for the use of semaphores in multiprocessing it is necessary for the LOCK prefix in 8086 instructions to acquire the Multibus. The signal PLOCK- (processor lock) from 8086 pin 29 is inverted at 7E-2 (p6zC7) to make LOCK+. LOCK+ is latched into 11E-9 (p7zC6) by ALE+ to synchronize to the instruction stream. 11E-9 goes tri-state during DMA (by enabling 11E-1). The signal at 11E-9 is called SLOCKHA+ (synchronized Lock or Hold Acknowledge) and will go active during Hold Acknowledge (DMA), since the driver is in tri-state, or during non-DMA times when a synchronized Lock signal is present. The fact that SLOCKHA+ goes tri-state and possibly float high during DMA is an undesired side effect (hence no pullup) and that condition is gated off at 4E-6 (p4zA4), making the signal SLOCK+ (synchronized Lock). As already described, the signal SLOCK+ causes ADORLOCK+ to go active at 1B-11 (p4zA5) and is one of the conditions that causes Multibus to be acquired. When ADORLOCK+ is active and AEN89- is

not, MBPNDG- (Multibus Pending) is asserted at 4B-11 (p5zA7) indicating that the 8289 is acquiring Multibus but has not done so. This causes the 8288 to disable its command lines (at 9E-6) and thereby not assert any command lines until Multibus is acquired. Once the bus is acquired and AEN89- causes MBPNDG- (Multibus Pending) to go inactive at 4B-11, the signal LOCKED- is asserted by gate 4B-3. LOCKED- is used in the DMA logic (to be described) to determine that the local bus is locked and DMA cycles into local memory should be inhibited.

Slave Mode

Slave Mode is entered when the Multibus makes use of RAM on the local bus. It is controlled primarily by the 8237 chip. Although the 8237 is used for arbitration of Slave Mode, addresses and commands come from Multibus rather than the DMA logic. For all other DMA channels the address and command come from the DMA logic during DMA transfers.

A slave request (SLAVERQ+) is generated by gate 1C-4 (p4zC2) when a Memory Read or Memory Write Command (MRDC- or MWTC-) is on Multibus at the same time as an address that refers to the local bus is on Multibus. The Multibus commands are ORed at 1D-3. The Slave addressing conditions are ORed at 1D-6. The gate at 1F-8 (p4zC3) causes recognition of a slave address if the Multibus address is in the range from 32K to 48K-1 and switch 2 bit 4 is open. The other addressing condition at 1D-4 is active if switch 2 bit 1 is closed and the Multibus address is in the upper 512K (AD13- on Multibus is active).

SLAVERQ+ is connected to the DMA chip request pin for channel 0 at 6F-19 (p5zD6). If the DMA chip has been programmed with channel 0 enabled, then the channel 0 acknowledge pin (25) will go active. As long as the acknowledge pin and SLAVERQ+ are active, SLAVE+ is asserted by 1D-11, indicating that the Multibus interface is in Slave mode. Slave Mode affects DMA ready logic at 1C-9 and 3G-8 as described in the DMA section.

SLAVE- enables the tri-state inverter at 2F-19 (p4zD6), causing ONBUS- to go active at 2F-5. ONBUS- causes address transceivers 4G, 7G, and 8G (p7) to be enabled. The direction of the drivers are established by MASTER+ being inactive such that the Multibus address drives the local bus

address. The same tri-state inverter that generates ONBUS- drives LA12+ and LA13+ at outputs 2F-7 and 2F-9 (p5zD6). LA12+ goes directly to the input of the LA12+ driver at 2F-11. The input for the LA13+ driver is pulled high so that for all Slave mode operations, LA12+ is always low (0) and therefore Slave mode memory accesses are always in the first 512K bytes of RAM, regardless of the setting of the corresponding Multibus address bit (AD13-).

BUSEN- is asserted at 1C-13 (p8zC6) when ONBUS- is active and the local bus Slave mode is not being inhibited by the Multibus inhibit lines (INH1- and INH2-). This causes the proper data bus drivers to be enabled depending on whether a byte-swap access has to occur. If LA0+ is asserted (that is, an odd byte is being addressed) then the Multibus slave addressed is required to place its data on the low-order 8 bits of the Multibus (DAT0- - DAT7-). The RAM data for odd bytes is always placed on the high-order part of the bus (MEMB+ - MEMF+). Therefore, a byte swap must occur. If LA0+ is not asserted then the addressed byte or word must not be swapped. The signal SWAP- is asserted whenever BUSEN+ is active and LA0+ is asserted (odd byte addressed) and INTA- is not active (INTA- is never active Slave mode, since the local 8259A cannot be a slave). This causes the inverting transceiver at 10F to be enabled which connects the Multibus DAT0- - DAT7- lines to be connected to MEM8+ to MEMF+. Whenever BUSEN+ is active and SWAP- is not, NOSWAP- is asserted at 2B-3. This enables transceivers 9G and 10G (p8) to connect the Multibus data bus to the MEM bus without swapping.

The direction of the data transfer is established by the signal MEMTOMB+ which, when active, indicates that the MEM bus will drive the Multibus data bus (DAT0- to DATF-). MEMTOMB+ is generated by gate 3B-6 (p5zB2) and is determined by gate 3B-11 for Master mode (as described above) and 3B-3 for Slave mode. 3B-3 goes active (low) causing MEMTOMB+ to go active when SLAVE+ is active and INCMD+ is active (the command on the local bus is an input command). INCMD+ is generated by gate 3D-8 and is the OR of IOR- asserted or MR- asserted. For Slave mode it goes active for memory reads (MR-) only since Slave mode is not used for I/O.

SLAVE+ is an input to the parity error reporting latch (port 54) at 11F-13 (p2zA6), so if a parity error occurs it is possible to detect that it was on a Slave mode memory read.

The flip-flops at 1A (p4zB7) cause at least a 100 nanosecond delay after SLAVE+ goes active before the gate at 3A-10 causes the command transceiver at 2G (p4zC4) to be enabled. The direction of the transceiver is established by the Master mode flip-flop at 2A-5 (p4zB6). This causes the memory read command (MRDC-) and the memory write command (MWTC-) to be driven onto the local bus MR- and MW- lines respectively.

Multibus Slave Transfer Acknowledge (XACK-) Logic

Each command (MRDC-, MWTC-, IORDC-, IOWTC, or INTA-) on the Multibus is completed by the XACK- signal going active. It is the responsibility of the slave device in a Multibus transaction to assert XACK-. When the local bus RAM is the slave, the XACK- must be generated by CT logic. This is done by gate 4A-6 (p5zB3). 4A-6 goes active when SLAVE+ is active, XACKTIME- is active and INHIBIT+ is inactive. INHIBIT+ is the OR of the two signals, INH1- and INH2-, from Multibus, which indicate that RAM slaves should not respond to the current cycle. XACKTIME- is the OR of many conditions, however, for Slave mode operation, only inputs 1, 11, 12, and 13 of 6A (p5zB5) are used. SLAVE+ indicates the Multibus interface is in Slave mode. ENMCYC+ indicates that a RAM cycle is in progress, and XACKWINDOW+ and RACK+ indicate together that the RAM cycle is complete.

Boot PROMs

The NAND gate 12F (p2zB6) decodes local bus Memory Reads in the top 32K of memory and generates LOCPROM- to select the 2 "boot" PROMS. The boot PROMS consist of 2 chips, the high (or odd) byte at 14F and the even (low) byte at 15F. The sockets at 14F and 15F are designed to accomodate several different PROM types as follows:

<u>chip type</u>	<u>organization</u>	<u>number of pins</u>	<u>ROM/EPROM</u>
2316	2K x 8	24	ROM
2716	2K x 8	24	EPROM
2332	4K x 8	24	ROM
2732	4K x 8	24	EPROM
2364	8K x 8	28	ROM
2764	8K x 8	28	EPROM

In order to accomodate both size chips the board has 28-pin sockets. When 24-pin devices are used, they are installed justified toward the connectors J2 and J3.

Note that the schematic is numbered according to the sockets (28 pin numbering). Therefore when 24-pin devices are installed, the chip pin numbers will not match the pin numbers on the schematic. The relationship is as follows:

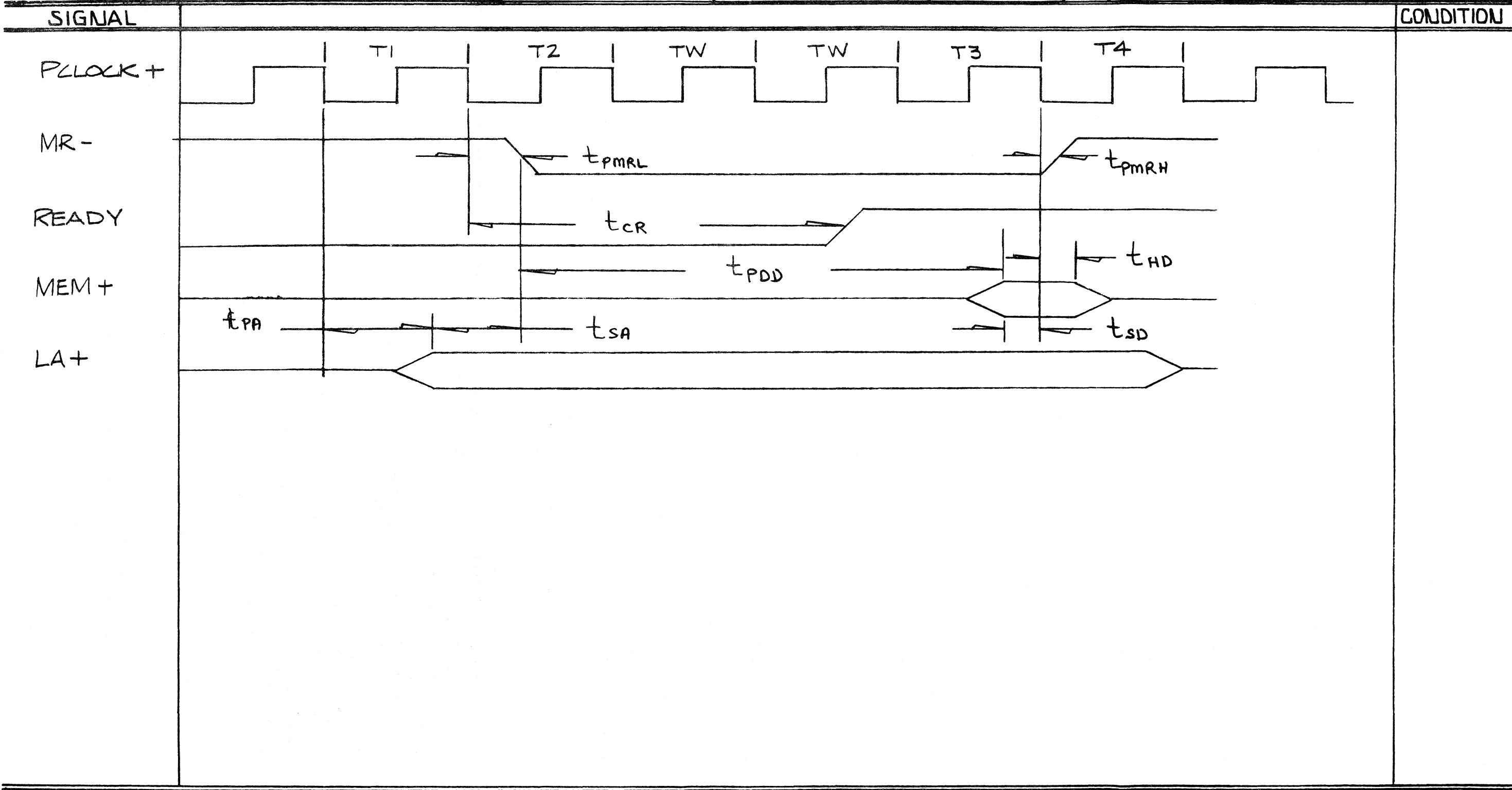
<u>Schematic pin number</u>	<u>24-pin chip pin number</u>	<u>Schematic pin number</u>	<u>24 pin chip pin number</u>
1	not used	28	not used
2	not used	27	not used
3	1	26	24
4	2	25	23
5	3	24	22
6	4	23	21
7	5	22	20
8	6	21	19
9	7	20	18
10	8	19	17
11	9	18	16
12	10	17	15
13	11	16	14
14	12	15	13

On pin 23 of the socket, the 2316 and 2716 require +5 volts and all other chip types require address bit C. This selection is made with bits 1 and 2 of switch 5. Bit 1 is closed to connect the address bit and bit 2 is closed to connect +5.

AC CHARACTERISTICS

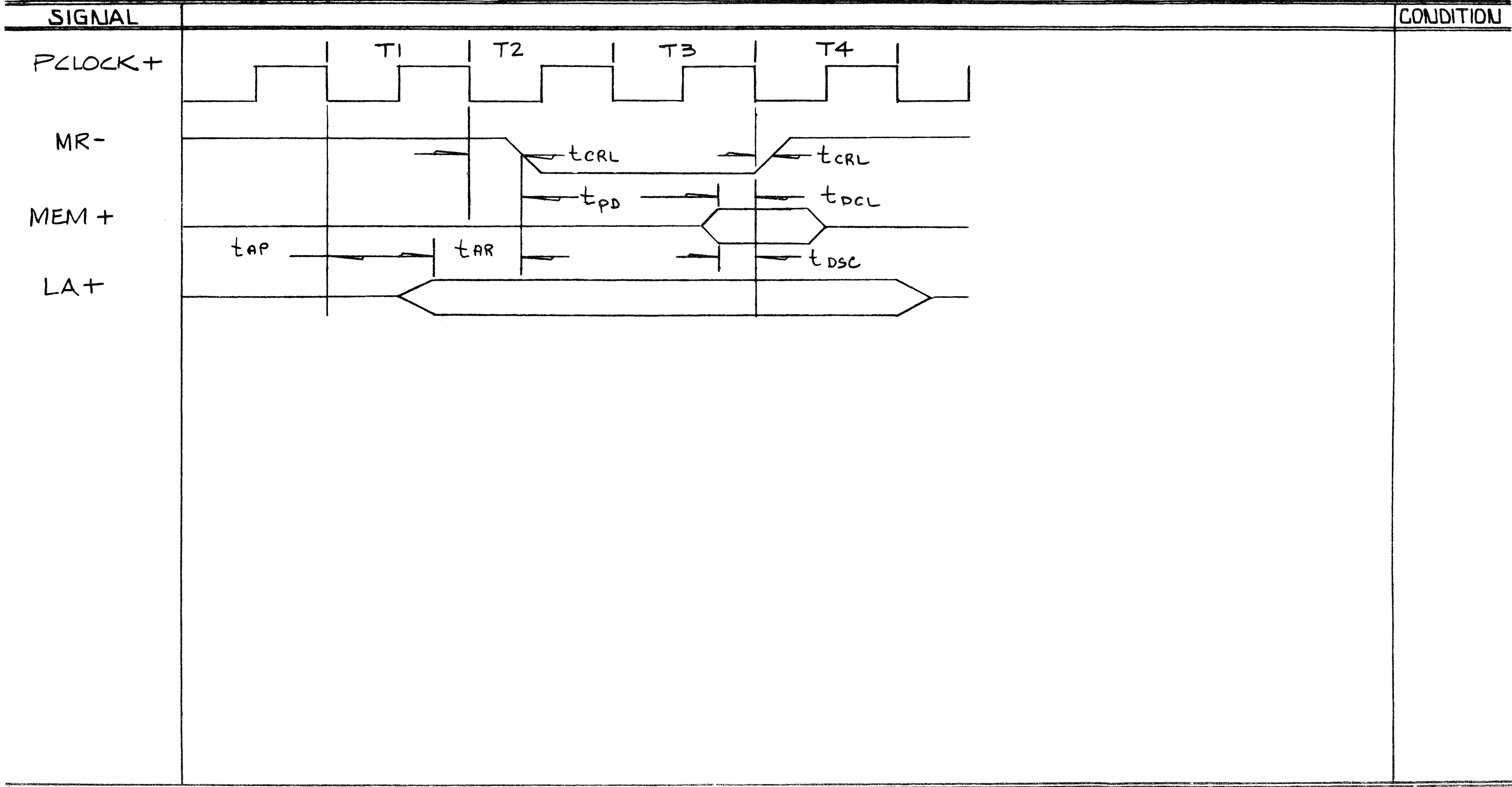


TITLE MR, MEMBUS, TIMING, FROM ACCESS		APPROVALS DRAWN	DATE	DWG NO	
APPLICATION		CHECKED		SHEET	
NEXT ASSY	USED ON	ENG		REV	



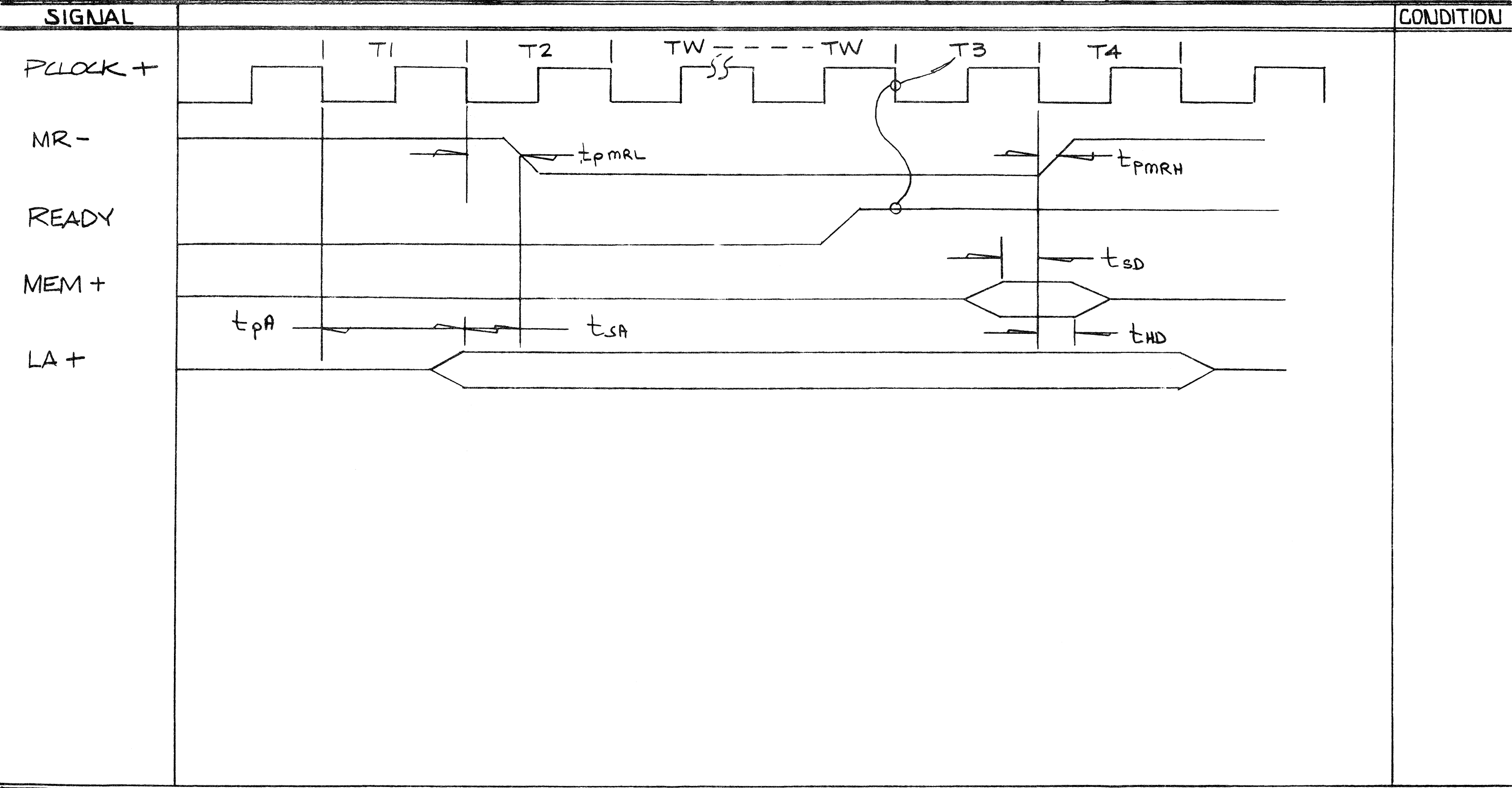


TITLE MR, MEM BUS, TIMING, READY=1		APPROVALS	DATE	DWG NO	
		DRAWN			
APPLICATION		CHECKED		SHEET	REV
NEXT ASSY	USED ON	ENG			



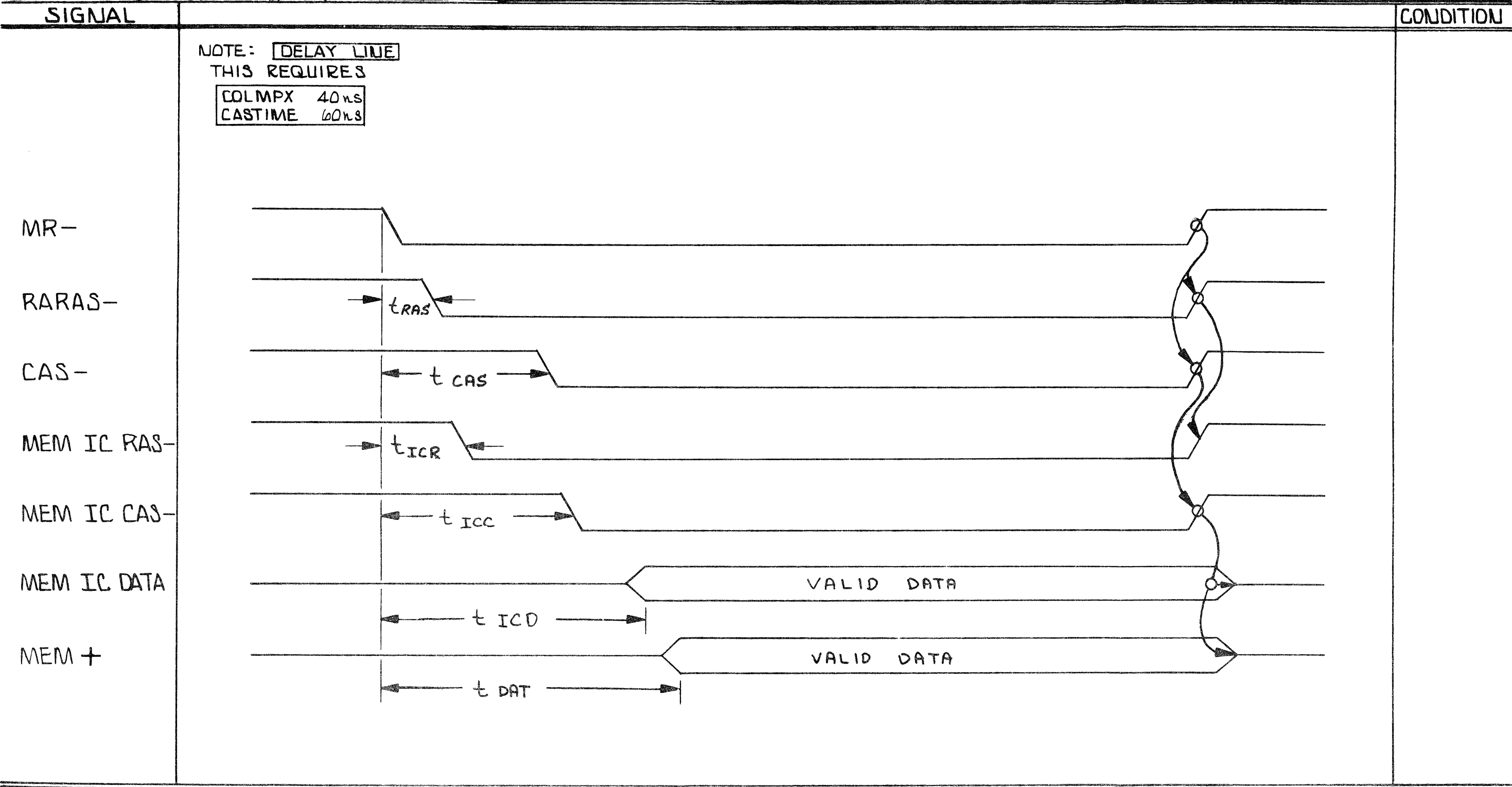


TITLE MR, MEMBUS, TIMING, ASYNCHRONOUS		APPROVALS DRAWN	DATE	DWG NO	
APPLICATION		CHECKED		SHEET	
NEXT ASSY	USED ON	ENG		REV	





TITLE RAM TIMING (BASED ON 5290-4 & 4116-4)		APPROVALS DRAWN SKTANGEN CHECKED	DATE 6-18-80	DWG NO	
APPLICATION				SHEET	
NEXT ASSY	USED ON	ENG		REV	



AC-DC CHARACTERISTICS

[illegible]

AC-DC CHARACTERISTICS

[illegible]

AC-DC CHARACTERISTICS

[illegible]

AC-DC CHARACTERISTICS

[illegible]

RAM TIMING

DC CHARACTERISTICS

AC-DC CHARACTERISTICS

[illegible]

REFERENCES

Edge-Connector Pin List

Connector J2

<u>Pin</u>	<u>Signal</u>
1	MR-
2	VIDRQ+
3	BDZRAM-
4	BD 1 RAM-
5	GND
6	COLMPX-
7	RARAS-
8	PROMEXDC-
9	RFGO-
10	RFDONE-
11	RAHIWE-
12	RALOWE-
13	78KHZ+
14	CAS7-
15	
16	CAS6-
17	CASS-
18	CAS4-
19	CAS3-
20	CAS2-
21	CAS1-
22	CAS0-
23	HIP+
24	LOP+
25	
26	MEMF+
27	MEME+
28	MEMD+
29	MEMC+
30	MEMB+
31	MEMA+
32	MEM9+
33	MEM8+
34	MEM7+
35	
36	MEM6+
37	MEM5+
38	MEM4+
39	MEM3+
40	MEM2+
41	MEM1+
42	MEM0+
43	LA13+
44	LA12+
45	

<u>Pin</u>	<u>Signal</u>
46	LA11+
47	LA10+
48	LAF+
49	LAE+
50	LAD+
51	LAC+
52	LAB+
53	LAA+
54	LA9+
55	
56	LA8+
57	LA7+
58	LA6+
59	LA5+
60	LA4+
61	LA3+
62	LA2+
63	LA1+
64	LA0+
65	
66	
67	+12V
68	+12V
69	+12V
70	+12V
71	
72	
73	
74	
75	
76	
77	+5V
78	+5V
79	+5V
80	+5V

Connector J3

<u>Pin</u>	<u>Signal</u>
1	+5V
2	+5V
3	+5V
4	+5V
5	-12V
6	-12V
7	GND
8	
9	BCLK-
10	INIT-
11	BPRINO-
12	Not Used
13	BUSY-

<u>Pin</u>	<u>Signal</u>
14	BREQ0-
15	
16	MRDC-
17	MWTC-
18	IORC-
19	IOWC-
20	XACK-
21	INH1-
22	RESERVED
23	INH2-
24	BHEN-
25	
26	AD10-
27	CBRQ-
28	AD11-
29	CCLK-
30	AD12-
31	INTA-
32	AD13-
33	INT7B-
34	INT6-
35	
36	INT5-
37	INT4B-
38	INT3-
39	INTZ-
40	INT1B-
41	INT0-
42	SIOACK-
43	SICRQ+
44	RDHIPEA-
45	
46	LBHEN+
47	RDLOPEA-
48	MW-
49	DISKACK-
50	DISKRQ+
51	ODDP+
52	WAIT-
53	IOBCS+
54	IOR-
55	
56	DCIRDY+
57	FASTIO+
58	DMACTICS-
59	DMACS-
60	IOB7+
61	IOB6+
62	IOB5+
63	IOB4+
64	IOB3+
65	
66	IOB2+

<u>Pin</u>	<u>Signal</u>
67	IOB1+
68	IOB0+
69	
70	IOCS-
71	NMI+
72	PICCS-
73	MEMCLK+
74	TEST
75	
76	IOW-
77	EOP-
78	VIDACK-
79	PERINT+
80	MWINH-
81	TO+
82	MBIO-
83	RESET-
84	PAREN+
85	
86	ADRF-
87	ADRE-
88	ADRD-
89	ADRC-
90	ADRB-
91	ADRA-
92	ADR9-
93	ADR8-
94	ADR7-
95	
96	ADR6-
97	ADR5-
98	ADR4-
99	ADR3-
100	ADR2-
101	ADR1-
102	ADRO-
103	DATF-
104	DATE-
105	
106	DATD-
107	DATC-
108	DATB-
109	DATA-
110	DAT9-
111	DAT8-
112	DAT7-
113	DAT6-
114	DAT5-
115	
116	DAT4-
117	DAT3-
118	DAT2-
119	DAT1-
120	DAT0-

Jumpers

<u>Name</u>	<u>Function</u>
A-C	Asynchronous READY
B-C	Synchronous READY

Switches

<u>Switch</u>	<u>Bit</u>	<u>Function</u>
1	1	Close for 4 Megahertz DMA.
1	2	Open for slow memory access (for debugging only).
1	4	Open for synchronous CPU and memory (for debugging only).
2	1	Close for Multibus slave region at greater than 512K.
2	4	Open for 32K to 48K multibus slave region.
3	1	Close for 512K multibus master boundary.
3	2	Close for 768K multibus master boundary.
3	3	Close if mainframe generates Multibus CCLK.
3	4	Close if mainframe generates Multibus BCLK.
4	1	Close if multibus master boundary is 768K.
4	2-4	Memory configuration (See below).
5	1	Close for 2364 and 2764 ROM types.
5	2	Close for 2316 and 2716 ROM types.
6	1	Close for no ROM expansion.
6	2	Open for no ROM expansion.
6	3	Close for no Multibus master memory access.
6	4	Close to always assert Multibus CBRQ.

The Memory Addressing Decode PROM

The CPU board has a 256 x 4 PROM on it that determines what rows of RAMs to enable when, and that tells the memory controller how much memory there is. The PROM determines this information based on the setting of three dipswitches on the board. This table lists the standard set of addressing and RAM configuration choices. Other PROMS can be programmed for other than the standard set.

CPIA

Switch setting SW 4 bit-2 3 4	Configuration number	Memory I/O RAM	Expansion RAM	Total RAM	64K chips on Expansion
c c c	0	128K	OK	128K	no
c o c	1	128K	32K	160K	no
c c o	2	128K	64K	192K	no
c o o	3	128K	96K	224K	no
o c c	4	128K	128K	256K	no
o o c	5	128K	256K	384K	yes
o c o	6	128K	384K	512K	yes
o o o	7	128K	512K	640K	yes

c = closed (on)

o = open

LED Indicators

LED	Meaning
CR1	Bus inactive
CR2	DMA active
CR3	Multibus active
CR5	Non-maskable interrupt
CR6	Interrupt

Test Points

Point	Meaning
TP 1	High during bus timeouts.
TP 2	Low when a DMA cycle is in progress.
TP 3	High a refresh is in progress. Normally oscillates at 12.8 khz.
TP 4	Ground to prevent refresh.
TP 5	High when a non-refresh memory cycle is in progress.

I/O-Port Address vs. Functions

address	logic	write reg.	read reg.
0	DMA	reserved	reserved
2	DMA	reserved	reserved
4	DMA	Comm. address	Comm. address
6	DMA	Comm. count	Comm. count
8	DMA	Disk address (low 16 bits)	Disk address
A	DMA	Disk count	Disk count
C	DMA	Video address	Video address
E	DMA	Video count	Video count
10	DMA	Command	Status
12	DMA	Request	-

14	DMA	Set/Reset Mask	-
16	DMA	Mode	-
18	DMA	Clear flip-flop	-
1A	DMA	Clear	-
1C	DMA	Mask	-
20	8259A	ICW1, OCW2, OCW3	IRR, ISR, INT Level
22	8259A	OCW1, ICW2, ICW3 ICW4	Mask

Device Pin Functions

The following several pages describe the pin functions of the complex ICs on the processor board.

Device Pin Functions: 8086 HMOS MICROPROCESSOR

The Intel 8086 is a new generation, high performance microprocessor implemented in N-channel depletion load, silicon gate technology (HMOS), and packaged in a 40-pin CerDIP package. The processor has attributes of both 8- and 16-bit microprocessors. It addresses memory as a sequence of 8-bit bytes, but has a 16-bit wide physical path to memory for high performance. 8086 features include the following:

- o Direct addressing capability to 1 MByte of memory
- o Assembly language compatible with 8080/8085
- o 14 word, by 16 bit register set with symmetrical operations
- o 24 operand addressing modes
- o Bit, byte, word, and block operations
- o 8- and 16-bit signed and unsigned arithmetic in binary or decimal, including multiply and divide
- o 5 MHz clock rate (8 MHz for 8086-2 (4MHz for 8086-4)
- o Multibus system compatibility

Pin functions tabulated below assume that the 8086 has been configured in its MAXIMUM mode, that is, the MN pin has been strapped to ground.

<u>Signal Name</u>	<u>Pin Name</u>	<u>I/O</u>	<u>Function</u>
--	VCC		+5V supply
--	GND		Ground. Both GND pins must be strapped together.
ADO to ADF+	AD ₀ to AD ₁₅	I/O	These lines constitute the time multiplexed memory/IO address (T ₁) and data (T ₂ , T ₃ , T _W , T ₄) bus. A ₀ is analogous to BHE/ for the lower byte of the data bus, pins D ₇ -D ₀ . It is LOW during T ₁ when a byte is to be transferred on the lower portion of the bus in memory or I/O operations. Eight-bit oriented devices tied to the lower half would normally use A ₀ to condition chip select functions. These lines are active HIGH and float to 3 state OFF during interrupt acknowledge and local bus "bold acknowledge".
Caution: Intel pin names do not use hexadecimal numbers.			
AD10+ to AD13+	A ₁₆ , S ₃ to A ₁₉ , S ₆	0	During T ₁ , these are the four most significant address lines for memory operations. During I/O operations these lines are LOW. During memory and I/O operations, status information is available on these lines during T ₂ , T ₃ , T _W , and T ₄ . The status of the interrupt enable FLAG bit (S ₅) is updated at the beginning of each CLK cycle. A ₁₇ /S ₄ and A ₁₆ /S ₃ indicate which relocation register is presently being used for data accessing. These lines float to 3-state OFF during local bus "hold acknowledge".
BHEN+	BHE/, S ₇	0	During T ₁ the bus high enable signal (BHE/) should be used to enable data onto the most significant half of the data bus, pins D ₁₅ -D ₈ . Eight-bit oriented devices tied to the upper half of the bus would normally use (BHE/) to condition chip select func-

tions. (BHE/) is LOW during T_1 for read, write and interrupt acknowledge cycles when a byte is to be transferred on the high portion of the bus. The S_7 status information is available during T_2 , T_3 , and T_4 . The signal is active LOW, and floats to 3-state OFF in "hold". It is LOW during T_1 for the first interrupt acknowledge cycle.

--	RD/	0	Read strobe indicates that the processor is performing a memory or I/O read cycle, depending on the state of the S_2 pin. This signal is used to read devices which reside on the 8086 local bus. RD/ is active LOW during T_2 , T_3 , and T_4 of any read cycle, and is guaranteed to remain HIGH in T_2 until the 8086 local bus has floated.
			This signal floats to 3-state OFF in "hold acknowledge"
READYSYNC+	READY	I	READY is the acknowledgement from the addressed memory or I/O device that it will complete the data transfer. The RDY signal from memory/IO is synchronized by the 8284 Clock Generator to form READY. This signal is active HIGH.
INTR+	INTR	I	Interrupt request is a level triggered input which is sampled during the last clock cycle of each instruction to determine if the processor should enter into an interrupt acknowledge operation. A subroutine is vectored to via an interrupt vector lookup table located in system memory. It can be internally masked by software resetting the interrupt enable bit. INTR is internally synchronized. This signal is active HIGH.
TEST-	TEST/	I	The TEST/ input is examined by the "Wait" instruction. If the TEST/ input is LOW execution continues, otherwise the processor waits in an "Idle" state. This input is synchronized internally during each clock cycle on the leading edge of CLK.
NMI+	NMI	I	Non-maskable interrupt is an edge triggered input which causes a type 2

interrupt. A subroutine is vectored to via an interrupt vector lookup table located in system memory. NMI is not maskable internally by software. A transition from a LOW to HIGH initiates the interrupt at the end of the current instruction. This input is internally synchronized.

RESET+	RESET	I	RESET causes the processor to immediately terminate its present activity. The signal must be active HIGH for at least four clock cycles. It restarts execution, as described in the Instruction Set description, when RESET returns LOW. RESET is internally synchronized.
PCLOCK+	CLK	I	The clock provides the basic timing for the processor and bus controller. It is asymmetric with a 33% duty cycle to provide optimized internal timing.
S0- S1- S2-	S0 S1/ S2/	O	These status lines are encoded as follows: (0,0,0) Interrupt Acknowledge (0,0,1) Read I/O Port (0,1,0) Write I/O Port (0,1,1) Halt (1,0,0) Code Access (1,0,1) Read Memory (1,1,0) Write Memory (1,1,1) Passive Notation: (0,0,1) means S₀/, S₁/ are LOW and S₂/ is HIGH. Status is active during T₄, T₁, and T₂ and is returned to the passive state (1,1,1) during T₃ or during T_W when READY is HIGH. This status is used by the 8288 Bus Controller to generate all memory and I/O access control signals. Any change by S₂ or S₁, or S₀ during T₄ is used to indicate the beginning of a bus cycle, and the return to the passive state in T₃ or T_W is used to indicate the end of a bus cycle. These signals float to 3-state OFF in "hold acknowledge".
RQ/, RQ/,	GT ₀ / GT ₁ /	I/O	The request/grant pins are used by other processor bus masters to force the processor to release the processor bus at the end of the processor's

current bus cycle. Each pin is bi-directional with RQ/, GT₀/ having higher priority than RQ/, GT₁/, RQ/, GT/. has an internal pull-up resistor so may be left unconnected. The request/grant sequence is as follows:

1. A pulse of 1 CLK wide from another processor bus master indicates a local bus request ("hold") to the 8086 (pulse 1).
2. During the CPU's next T₄ or T₁ a pulse 1 CLK wide from the 8086 to the requesting master (pulse 2), indicates that the 8086 has allowed the local bus to float and that it will enter the "hold acknowledge" state at the next CLK. The CPU's bus interface unit is disconnected logically from the processor bus during "hold acknowledge".
3. A pulse 1 CLK wide from the requesting master indicates to the 8086 (pulse 3) that the "hold" request is about to end and that the 8086 can reclaim the processor bus at the next CLK.

Each master-master exchange of the processor bus is a sequence of 3 pulses. There must be one dead CLK cycle after each bus exchange. Pulses are active LOW.

LOCK- LOCK/ 0

The LOCK/ output indicates that other system bus masters are not to gain control of the system bus while LOCK/ is active LOW. The LOCK/ signal is activated by the "LOCK" prefix instruction and remains active until the completion of the next instruction. This signal is active LOW, and floats to 3-state OFF in "hold acknowledge".

-- QS₁ 0
 QS₀

QS₁ and QS₀ provide status to allow external tracking of the internal 8086 instruction queue.

QS₁ QS₀

0 0 No Operation

0 1 First Byte of Op Code from Queue

1 0 Empty the Queue

1 1 Subsequent Byte from Queue

The queue status is valid during the

CLK cycle after which the queue operation is performed.

--

MN

I

The purpose of the pin is to allow the 8086 to perform in either of two modes: MINIMUM or MAXIMUM. Several pins assume totally different functions when the mode changes. The pin descriptions given above are valid only in MAXIMUM mode.

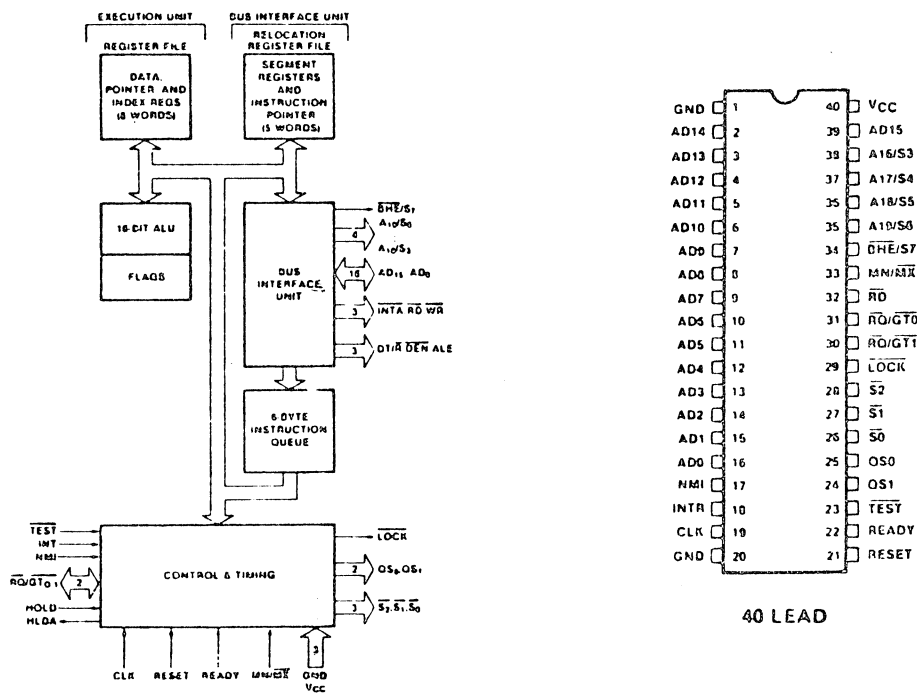


Figure 4-2 8086 CPU Chip

Device Pin Functions: 8284 CLOCK DRIVER

The 8284 is a bipolar clock generator/driver designed to provide clock signals for the 8086 and peripherals. It also contains READY logic for operation with two MULTIBUS systems and provides the processors required READY synchronization and timing. Reset logic with hysteresis and synchronization is also provided.

<u>Signal Name</u>	<u>Pin Name</u>	<u>I/O</u>	<u>Function</u>
--	VCC		+5V supply
--	GND		Ground
MASTER-	AEN1/	I	ADDRESS ENABLE. AEN/ is an active LOW signal. AEN1/ serves to qualify its respective Bus Ready Signal (RDY1 or RDY2). AEN1/ validates RDY1 while AEN2/ validates RDY2. Two AEN signal inputs are useful in system configurations which permit the processor to access two Multi-Master System Busses. In non Multi-Master configurations the AEN/ signal inputs are tied true (LOW).
LOCKMASTER-	AEN2/		
XACK+	RDY1	I	BUS READY (Transfer Complete). RDY is an active HIGH signal which is an indication from a device located on the system data bus that data has been received, or is available. RDY1 is qualified by AEN1/ while RDY2 is qualified by AEN2/.
LOCRDY+	RDY2		
READYSYNC	READY	0	READY. READY is an active HIGH signal which is the synchronized RDY signal input. Since RDY occurs asynchronously with respect to the clock (CLK) it may be necessary for them to be synchronized before being presented to the 8284. READY is

cleared after the guaranteed hold time to the processor has been met.

--	X1	I	CRYSTAL IN. X1 and X2 are the pins to
	X2		which a crystal is attached
	TNK		TNK (TNK) serving as the overtone input. The crystal frequency is 3 times the desired processor clock frequency.
--	F	I	FREQUENCY/CRYSTAL SELECT. F is a strapping option. When strapped LOW, F permits the processor's clock to be generated by the crystal. When F is strapped HIGH, CLK is generated from the EFI input.

MEMCLK+	EFI	I	EXTERNAL FREQUENCY IN. When F is strapped HIGH, CLK is generated from the input frequency appearing on this pin. The input signal is a square wave 3 times the frequency of the desired CLK output.
PCLOCK+	CLK	O	PROCESSOR CLOCK. CLK is the clock output by the processor and all devices which directly connect to the processor's local bus (i.e., the bipolar support chips and other MOS devices). CLK has an output frequency which is 1/3 of the crystal or EFI input frequency and a 1/3 duty cycle. An output HIGH of 4.5 volts ($V_{cc}=5V$) is provided on this pin to drive MOS devices.
--	PCLK	O	PERIPHERAL CLOCK. PCLK is a TTL level peripheral clock signal whose output frequency is 1/2 that of CLK and has a 50% duty cycle.
--	OSC	O	OSCILLATOR OUTPUT. OSC is the TTL level output of the internal oscillator circuitry. Its frequency is equal to that of the crystal.
--	RES/	I	RESET IN. RES/ is an active LOW signal which is used to generate RESET. The 8284 provides a Schmitt trigger input so that an RC connection can be used to establish the power-up reset of proper duration.
PUPCLR+	RESET	O	RESET. Reset is an active HIGH signal which is used to reset the 8086 family processors. Its timing characteristics are determined by RES/.
--	CSYNC	I	CLOCK SYNCHRONIZATION. CSYNC is an active HIGH signal which allows multiple 8284's to be synchronized to provide clocks that are in phase. When CSYNC is HIGH the internal counters are reset. When CSNYC goes LOW the

internal counters are allowed to resume counting. CSYNC needs to be externally synchronized to EFI. When using the internal oscillator CSYNC should be hardwired to ground.

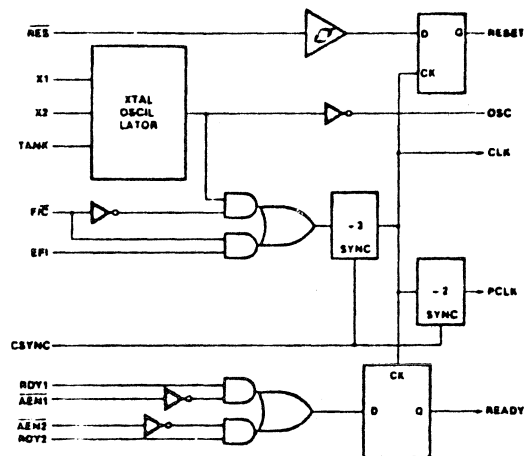
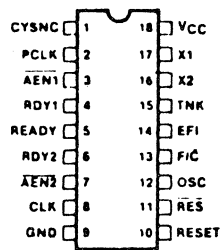


Figure 4-3 8284 Clock Generator

Device Pin Functions: 8289 BUS ARBITER

The 8289 Bus Arbiter, a 20-pin bipolar component for use in medium to large 8086 multimaster/multiprocessing systems, provides bus arbitration as well as bipolar buffering and drive capability.

<u>Signal Name</u>	<u>Pin Name</u>	<u>I/O</u>	<u>Function</u>
--	V _{cc}		+5V supply $\pm$ 10%
--	GND		Ground
--	S0/ S1/ S2/	I	STATUS INPUT PINS: These pins are the status input pins from an 8086 processor. The 8289 decodes these pins to initiate bus request and surrender actions.
PCLOCK+	CLK	I	CLOCK: This is the clock from the 8284 clock chip and serves to establish when bus arbiter actions are initiated.
PLOCK-	LOCK/	I	LOCK: LOCK/ is a processor-generated signal which when activated (low) serves to prevent the arbiter from surrendering the multi-master system bus to any other bus arbiter, regardless of its priority.
--	CRQLCK/	I	COMMON REQUEST LOCK: CRQLCK/ is an active low signal which serves to prevent the arbiter from surrendering the multi-master system bus to any other bus arbiter requesting the bus through the CBRQ/ input pin.
--	RESB	I	RESIDENT BUS is a strapping option to configure the arbiter to operate in systems having both a multi-master system bus and a Resident Bus. When it is strapped high the multi-master system bus is requested or

surrendered as a function of the SYSB input pin. When it is strapped low the SYSB input is ignored.

--	ANYRQST	I	ANYRQST is a strapping option which permits the multi-master system bus to be surrendered to a lower priority arbiter as though it were an arbiter of higher priority (i.e., when a lower priority arbiter requests the use of the multi-master system bus, the bus is surrendered as soon as it is possible). Strapping CBRQ/low and ANYRQST high forces the 8289 arbiter to surrender the multi-master system bus after each transfer cycle. Note that when surrender occurs BREQ is driven false (high).
--	IOB/	I	IO BUS: IOB/ is a strapping option which configures the 8289 Arbiter to operate in systems having both an IO Bus (Peripheral Bus) and a multi-master system bus. The arbiter requests and surrenders the use of the multi-master system bus as a function of the status line, S2/. The multi-master system bus is permitted to be surrendered while the processor is performing IO commands and is requested whenever the processor performs a memory command. Interrupt cycles are assumed as coming from the peripheral bus and are treated as would be an IO command.
--	AEN/	O	ADDRESS ENABLE. AEN/ is the output of the 8289 Arbiter to the processor's address latches, to the 8288 Bus Controller and 8284 Clock Generator. AEN/ serves to instruct the Bus Controller and address latches when to tri-state their output drivers.
--	SYSB	I	SYSTEM BUS, NOT RESIDENT BUS. SYSB is an input signal when the arbiter is configured in the S.R. Mode (RESB is strapped high)

which serves to determine when the multi-master system bus surrendering is permitted. The signal is intended to originate from some form of address mapping circuitry such as a decoder or PROM attached to the resident address bus. Signal transitions and glitches are permitted on this pin from Ø1 of T4 to Ø1 to T2 of the processor cycle. During the period from Ø1 of T2 to Ø1 of T4 only clean transitions are permitted on this pin (no glitches). If a glitch does occur the arbiter may capture or miss it and the multi-master system bus maybe requested or surrendered, depending upon the state of the glitch. The arbiter requests the multi-master system bus in the S.R. Mode when the state of the SYSB pin is high and permits the bus to be surrendered when this pin is low.

CBRQ-	CBRQ/	I/O	COMMON BUS REQUEST. CBRQ/ is an input signal which serves to instruct the arbiter if there are any other arbiters of lower priority requesting the use of the multi-master system bus. The CBRQ/ pins (open-collector output) of all the 8289 Bus Arbiters which are to surrender the multi-master system bus upon request are connected together. The Bus Arbiter running the current transfer cycle will not itself pull the CBRQ/ line low. Any other arbiter connected to the CBRQ/ line can request the multi-master system bus. The arbiter presently running the current transfer cycle drops its BREQ/ signal and surrenders the bus whenever the proper surrender conditions exist. Strapping CBRQ/ low and ANYRQST high allows the multi-master system bus to be surrendered after each transfer cycle. See the pin definition of ANYRQST.
-------	-------	-----	---

RESET-	INIT/	I	INITIALIZE: INIT/ is an active low multi-master system bus input signal which is used to reset all the bus arbiters on the multi-master system bus. After initialization, no arbiters have the use of the multi-master system bus.
BCLK-	BCLK/	I	BUS CLOCK: BCLK/ is the multi-master system bus clock to which all multi-master system bus interface signals are synchronized.
BREQ-	BREQ/	O	BUS REQUEST: BREQ/ is an active low output signal in the parallel Priority Resolving Scheme which the arbiter activates to request the use of the multi-master system bus.
BPRN-	BPRN/	I	BUS PRIORITY IN: BRPN/ is the active low signal returned to the arbiter to instruct it that it may acquire the multi-master system bus on the next falling edge of BCLK. BPRN/ indicates to the arbiter that it is the highest priority requesting arbiter presently on the bus. The loss of BPRN/ instructs the arbiter that it has loss priority to a higher priority arbiter.
BPRO-	BPRO/	O	BUS PRIORITY OUT: BPRO/ is an active low output signal which is used in the serial priority resolving scheme where BPRO/ is daisy chained to BPRN/ of the next lower priority arbiter.
BUSY-	BUSY/	I/O	BUSY/ is an active low open collector multi-master system bus interface signal which is used to instruct all the arbiters on the bus when the multi-master system bus is available. When the multi-master system bus is available the highest requesting arbiter (determined by BPRN/ seizes the bus and pulls BUSY/ low to keep other arbiters off of the bus. When the arbiter is

done with the bus it releases the BUSY/ signal permitting it to go high and thereby allowing another arbiter to acquire the mulit-master system bus.

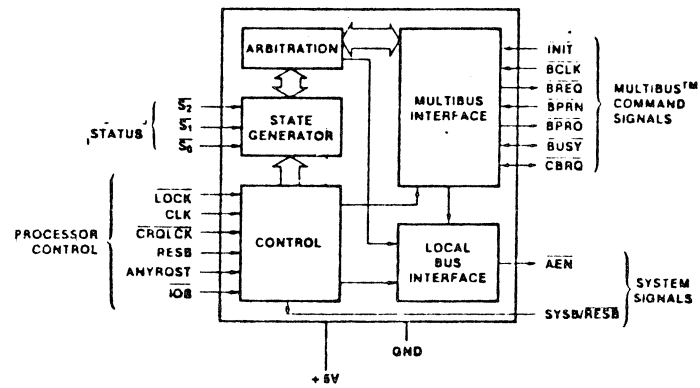
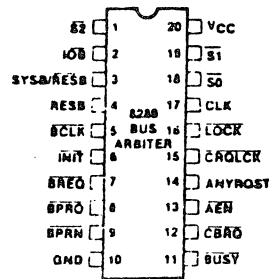


Figure 4-4 8289 Bus Arbiter

Device Pin Functions: 8288 BUS CONTROLLER

The Intel 8288 Bus Controller is a 20-pin bipolar component for use with medium-to-large 8086 systems. The bus controller provides command and control timing generation as well as bipolar bus drive capability while optimizing system performance.

<u>Signal Name</u>	<u>Pin Name</u>	<u>I/O</u>	<u>Function</u>
--	VCC		+5V supply.
--	GND		Ground.
S0-	S0/	I	STATUS INPUT PINS: These pins are the status input pins from the 8086, 8088 or 8089 processors. The 8288 decodes these inputs to generate command and control signals at the appropriate time. When these pins are not in use (passive) they are all HIGH.
S1-	S1/		
S2-	S2/		
PCLOCK+	CLK	I	CLOCK: This is a clock signal from the 8284 clock generator and serves to establish when command and control signals are generated.
ALE+	ALE	O	ADDRESS LATCH ENABLE: This signal serves to strobe an address into the address latches. This signal is active HIGH and latching occurs on the falling (HIGH to LOW) transition. ALE is intended for use with transparent D type latches.
DEN+	DEN	O	DATA ENABLE: This signal serves to enable data transceivers onto either the local or system data bus. This signal is active HIGH.
DT+	DT	O	DATA TRANSMIT/RECEIVE: This signal establishes the direction of data flow through the transceivers. A HIGH on this line indicates Transmit

(write to I/O or memory) and a LOW indicates Receive (Read).

--	AEN/	I	ADDRESS ENABLE: AEN/ enables command outputs of the 8288 Bus Controller at least 105 ns after it becomes active (LOW). AEN/ going inactive immediately 3-states the command output drivers. AEN/ does not affect the I/O command lines if the 8288 is in the I/O Bus mode (IOB tied HIGH).
--	CEN	I	COMMAND ENABLE: When this signal is LOW all 8288 command outputs and the DEN and PDEN/ control outputs are forced to their inactive state. When this signal is HIGH, these same outputs are enabled.
--	IOB	I	INPUT/OUTPUT BUS MODE: When the IOB is strapped HIGH the 8288 functions in the I/O Bus mode. When it is strapped LOW, the 8288 functions in the System Bus mode.
IOW-	IOWC/	O	I/O WRITE COMMAND: This command line instructs an I/O device to read the data on the data bus. This signal is active LOW.
IOR-	IORC/	O	I/O READ COMMAND: This command line instructs an I/O device to drive its data onto the data bus. This signal is active LOW.
MW-	MWTC/	O	MEMORY WRITE COMMAND: This command line instructs the memory to record the data present on the data bus. This signal is active LOW.
MR-	MRDC/	O	MEMORY READ COMMAND: This command line instructs the memory to drive its data onto the data bus. This signal is active LOW.
LINTA-	INTA/	O	INTERRUPT ACKNOWLEDGE: This command line tells an interrupting device that its interrupt has been acknowledged and that it should drive vectoring information onto the data bus. This signal is active LOW.

MCE+	MCE or PDEN/	0	This is a dual function pin. MCE (IOB is tied LOW): Master Cascade Enable occurs during an interrupt sequence and serves to read a Cascade Address from a master PIC (Priority Interrupt Controller) onto the data bus. The MCE signal is active HIGH. PDEN/ (IOB is tied HIGH): Peripheral Data Enable enables the data bus transceiver for the I/O bus during I/O instructions. It performs the same function for the I/O bus that DEN performs for the system bus. PDEN/ is active LOW.
--	AMWC/	0	ADVANCED MEMORY WRITE COMMAND: The AMWC/ issues a memory write command earlier in the machine cycle to give memory devices an early indication of a write instruction. Its timing is the same as a read command signal. AMWC/ is active LOW.
--	AIOWC/	0	ADVANCED I/O WRITE COMMAND: The AIOWC/ issues an I/O Write Command earlier in the machine cycle to give I/O devices an early indication of a write instruction. Its timing is the same as a read command signal. AIOWC/ is active LOW.

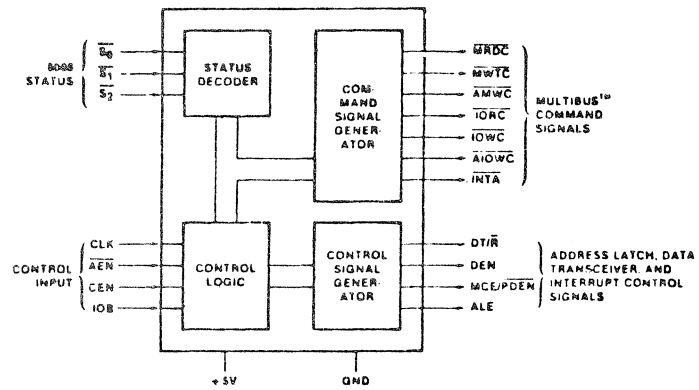
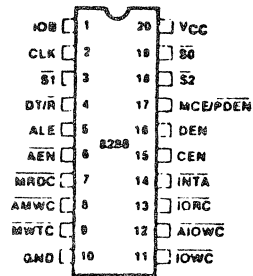


Figure 4-5 8288 Bus Controller

Device Pin Functions: 8237-2 DMA CONTROLLER

The 8237-2 Mulimode Direct Memory Access (DMA) Controller is a peripheral interface circuit for microprocessor systems. It is designed to improve system performance by allowing external devices to directly transfer information to or from the system memory. The 8237-2 offers a wide variety of programmable control features to enhance data throughput and system optimization and to allow dynamic reconfiguration under program control.

The 8237-2 is designed to be used in conjunction with an external 8-bit address register such as the 8282. It contains four independent channels and may be expanded to any number of channels by cascading additional controller chips. The three basic transfer modes allow programmability of types of DMA service by the user. Each channel can be individually programmed to Autoinitiale to its original condition following an End of Process (EOP).

Each channel has a full 16 bit address and word count capability.

The 8237-2 is a 5 MHz selected version of the standard 3 MHz 8237.

<u>Signal Name</u>	<u>Pin Name</u>	<u>I/O</u>	<u>Function</u>
--	VCC		+5 volt supply
--	GND		Ground
DMCLK+	CLK	I	CLOCK controls the internal operations of the 8237 and its rate of data transfers. The input may be driven at up to 3 MHz for the standard 8237 and up to 5 MHz for the 8237-2.
DMAC-	CS/	I	CHIP SELECT is an active low input used to select the 8237 as an I/O device during the idle cycle. This allows CPU communication on the data bus.

RESET+	RESET	I	RESET is an asynchronous active high input which clears the Command, Status, Request and Temporary registers. It also clears the first/last flip/flop and sets the Mask register. Following a Reset the device is in the Idle cycle.
--	READY	I	READY is an input used to extend the memory read and write pulses from the 8237 to accommodate slow memories or I/O peripheral devices.
--	HLDA	I	HOLD ACKNOWLEDGE indicates that control of the system buses has been relinquished.
SLAVERQ+ SIORQ+	DREQ0 DREQ1	I	DMA REQUEST lines are individual asynchronous channel request inputs used by peripheral circuits to obtain DMA service. In Fixed Priority, DREQ0 has the highest priority and DREQ3 has the lowest priority. A request is generated by activating the DREQ line of a channel. DACK will acknowledge the recognition of DREQ signal. Polarity of DREQ is programmable. Reset initializes these lines to active high. DREQ must be maintained until the corresponding DACK goes active.
DISKRQ+	DREQ2		
VIDRQ+	DREQ3		
IOB0+ ...	DB0 ...	I/O	DATA BUS lines are bidirectional three-state signals connected to the system data bus. The outputs are enabled in the Program Condition during the I/O Read to output the contents of an Address register, a Status register, the Temporary register or a Word Count register to the CPU. The outputs are disabled and the inputs are read during an I/O Write cycle when the CPU is programming the 8237 control registers. During DMA cycles the most significant 8 bits of the address are output
IOB7+	DB7		

onto the data bus to be strobed into an external latch by ADSTB.

--	IOR/	I/O	I/O READ is a bidirectional active low three-state line. In the Idle cycle, it is an input control signal used by the CPU to read the control registers. In the Active cycle, it is an output control signal used by the 8237 to access data from a peripheral during a DMA Write transfer.
--	IOW/	I/O	I/O WRITE is a bidirectional active low three-state line. In the Idle cycle, it is an input control signal used by the CPU to load information into the 8237. In the Active cycle, it is an output control signal used by the 8237 to load data to the peripheral during a DMA Read transfer.
EOP-	EOP/	I/O	END OF PROCESS is an active low bidirectional signal. Information concerning the completion of DMA services is available at the bidirectional EOP/ pin. The 8237 allows an external signal to terminate an active DMA service. This is accomplished by pulling the EOP/ input low with an external EOP/ signal. the 8237 also generates a pulse when the terminal count (TC) for any channel is reached. This generates an EOP/ signal which is output through the EOP/ Line. The reception of EOP/, either internal or external, will cause the 8237 to terminate the service, reset the request, and, if Autoinitialize is enabled, to write the base registers to the current registers of that channel. The mask bit and TC bit in the status word will be set for the currently active channel by EOP/ unless the channel is programmed for Autoinitialize. In that case, the mask bit remains clear. EOP/ should be tied high

with a pull-up resistor if it is not used to prevent erroneous end of process inputs.

DMAA0	A0	I/O	A0-A3, the four LEAST SIGNIFICANT ADDRESS LINES, are bidirectional three-state ports. In the Idle cycle they are inputs and are used by the 8237 to address the control register to be loaded or read. In the Active cycle they are outputs and provide the lower 4 bits of the output address.
...	...		
DMAA3	A3		
DMAA4	A4	0	A4-A7, the four MOST SIGNIFICANT ADDRESS LINES, are three-state outputs and provide 4 bits of address. These lines are enabled only during the DMA service.
...	...		
DMAA7	A7		
--	HRQ	0	HOLD REQUEST to the CPU is used to request control of the system bus. If the corresponding mask bit is clear, the presence of any valid DREQ causes the 8237 to issue the HRQ. After HRQ goes active at least one clock cycle (TCY) must occur before HLDA goes active.
SLAVE-	DACK0	0	There is one DMA ACKNOWLEDGE output for each of the four independent DMA channels. This signal notifies the peripheral(s) on a given channel when a DMA cycle has been granted. The sense of these lines is programmable. Reset initializes them to active low.
	DACK1		
DISKACK-	DACK2		
VIDACK-	DACK3		
--	AEN	0	ADDRESS ENABLE: This output enables the 8-bit latch containing the upper 8 address bits onto the system address bus. AEN can also be used to disable other system bus drivers during DMA transfers. AEN is active HIGH.
ADSTB+	ADSTB	0	ADDRESS STROBE facilitates externally latching the upper address byte.

--

MEMR/

0

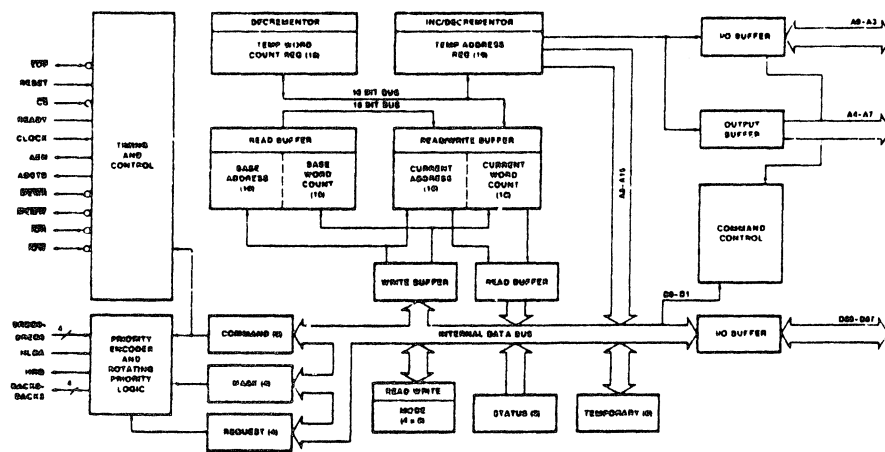
MEMORY READ is an active low three-state output used to access data from the selected memory location during a DMA Read.

--

MEMW/

0

MEMORY WRITE is an active low three-state output used to write data to the selected memory location during a DMA Write.



4 - 78

Device Pin Functions: 8259A INTERRUPT CONTROLLER

The 8259A Programmable Interrupt Controller handles up to eight vectored priority interrupts for the CPU. It is cascadable for up to 64 vectored priority interrupts without additional circuitry. It is packaged in a 28-pin DIP, uses NMOS technology and requires a single +5V supply. Circuitry is static, requiring no clock input.

The 8259A is designed to minimize the software and real time overhead in handling multi-level priority interrupts. It has several modes, permitting optimization for a variety of system requirements.

<u>Signal Name</u>	<u>Pin Name</u>	<u>I/O</u>	<u>Function</u>
--	VCC		+5 volt supply.
--	GND		Ground.
PICCS-	CS/	I	CHIP SELECT: A LOW on this input enables the 8259A. No reading or writing of the chip will occur unless the device is selected.
IOW-	WR/	I	WRITE: A LOW on this input enables the CPU to write control words (ICWs and OCWs) to the 8259A.
IOR-	RD/	I	READ; A LOW on this input enables the 8259A to send the status of the Interrupt Request Register (IRR), In Service Register (ISR), the Interrupt Mask Register (IMR), or the Interrupt level onto the Data Bus.
IOB7+	D7	I/O	DATA: A 3-state, bidirectional 8-bit buffer is used to interface the 8259A to the system Data Bus. Control words and status information are transferred through the Data Bus Buffer.
IOB0+	D0		

CAS2+	CAS2	I/O	COLUMN ADDRESS STROBE: These pins are outputs when the 8259A is used as a master and inputs when the 8259A is used as a slave. As a master, the 8259A sends the ID of the interrupting slave device onto the CAS0-2 lines. The slave thus selected will send its preprogrammed subroutine address onto the Data Bus during the next one or two consecutive INTA/ pulses.
CAS1+	CAS1		
CAS0+	CAS0		
DEN59-	SP/	I/O	SLAVE PROGRAM or ENABLE is a dual function pin. When in the buffered mode this pin is ENABLE, which serves as an output suitable for the control input, of buffer transceivers. When not in the buffered mode, this pin functions as SLAVE PROGRAM an input used to designate a slave (LOW) or a master (HIGH).
--	INT	O	INTERRUPT goes directly to the CPU interrupt input. The V _{OH} level on this line is designed to be compatible with the 8086.
--	IR7	I	The interrupts at the IR input lines are handled by two registers in cascade, the Interrupt Request Register (IRR) and the In-Service (ISR). The IRR is used to store all the interrupt levels which are requesting service, and the ISR is used to store all the interrupt levels which are being serviced.
	...		
	IRO		
--	INTA/	I	INTERRUPT ACKNOWLEDGE pulses will cause the 8259A to release vectoring information onto the data bus. The format of this data depends on the system mode of the 8259A.
LA1+	A ₀	I	This input signal is used in conjunction with WR/ and RD/ signals to write commands into the various command registers, as

well as reading the various
status registers of the chip.
This line can be tied directly to
one of the address lines.

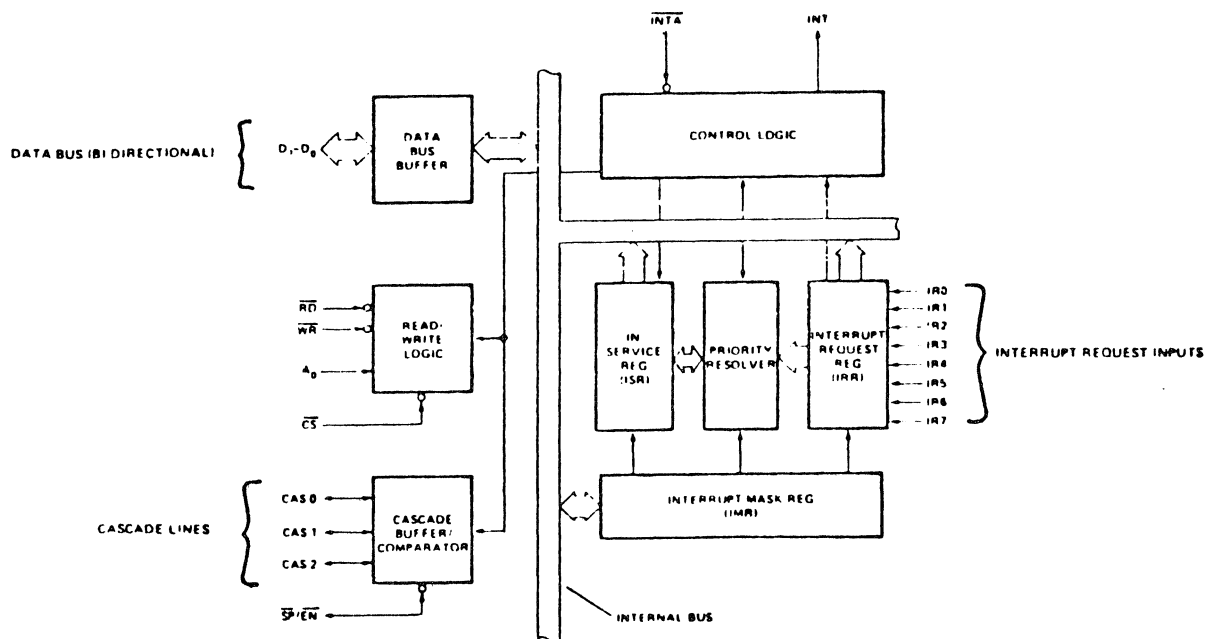
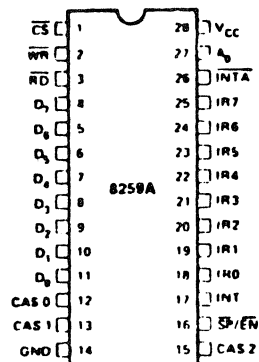
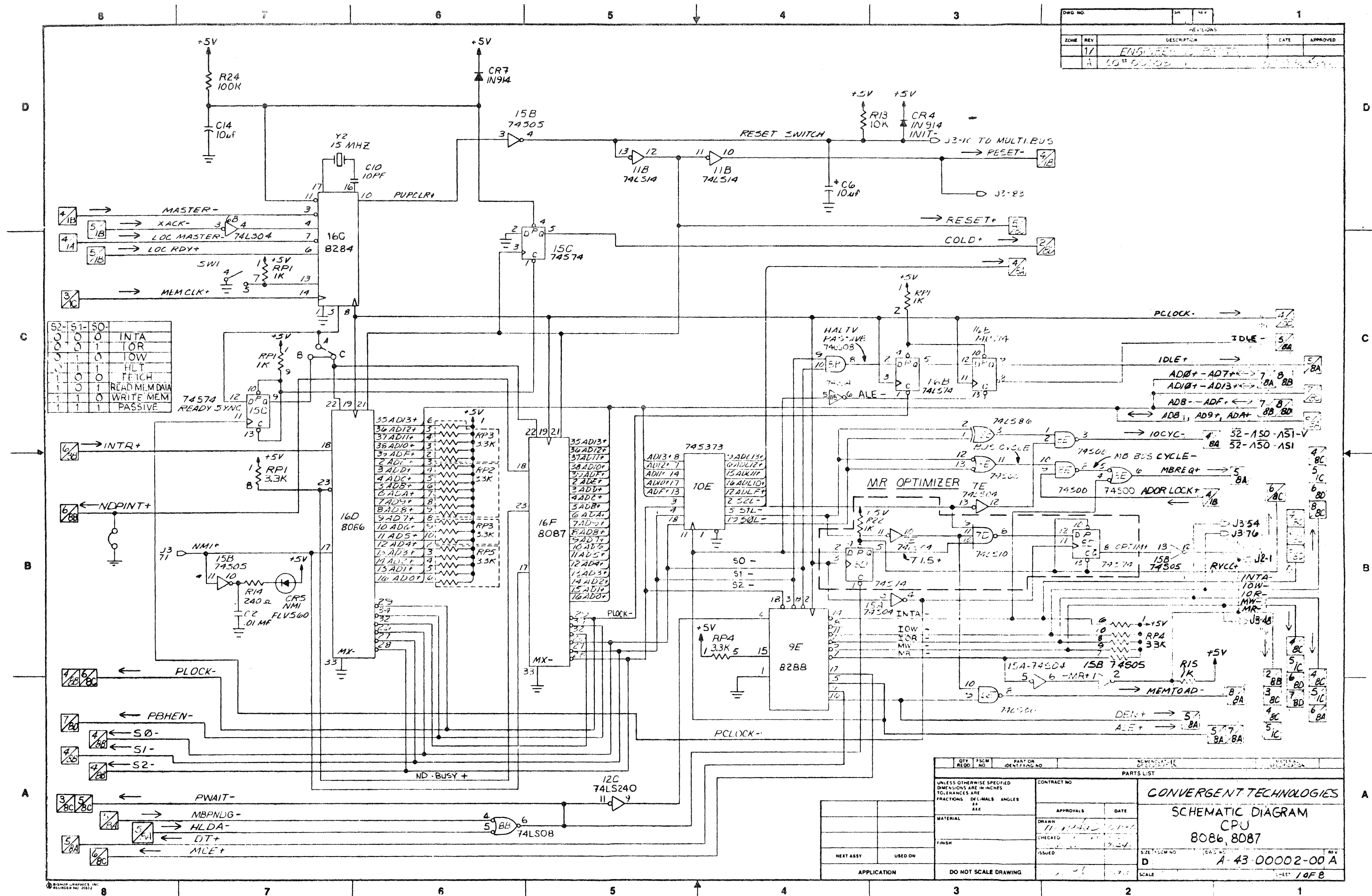
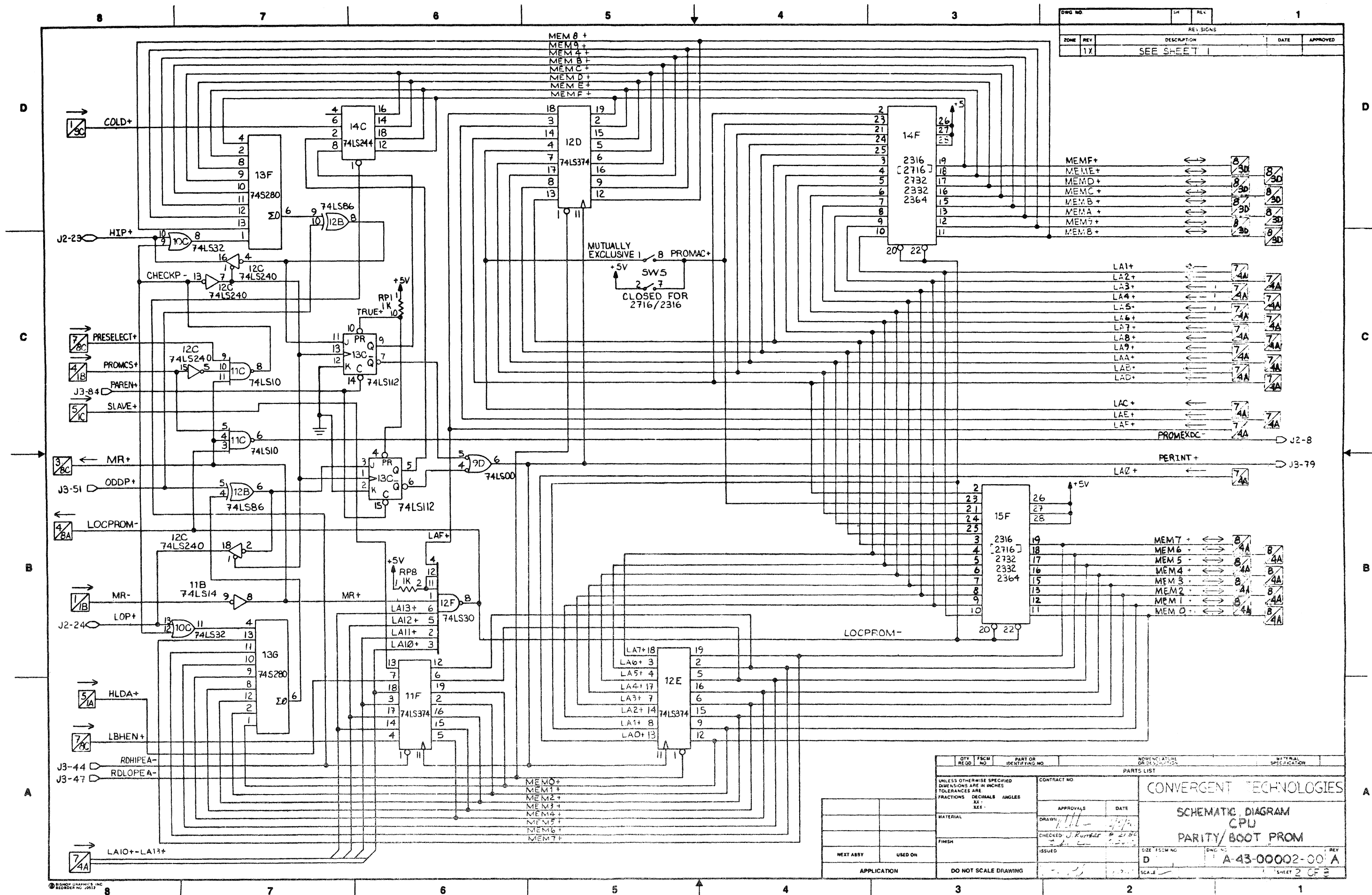
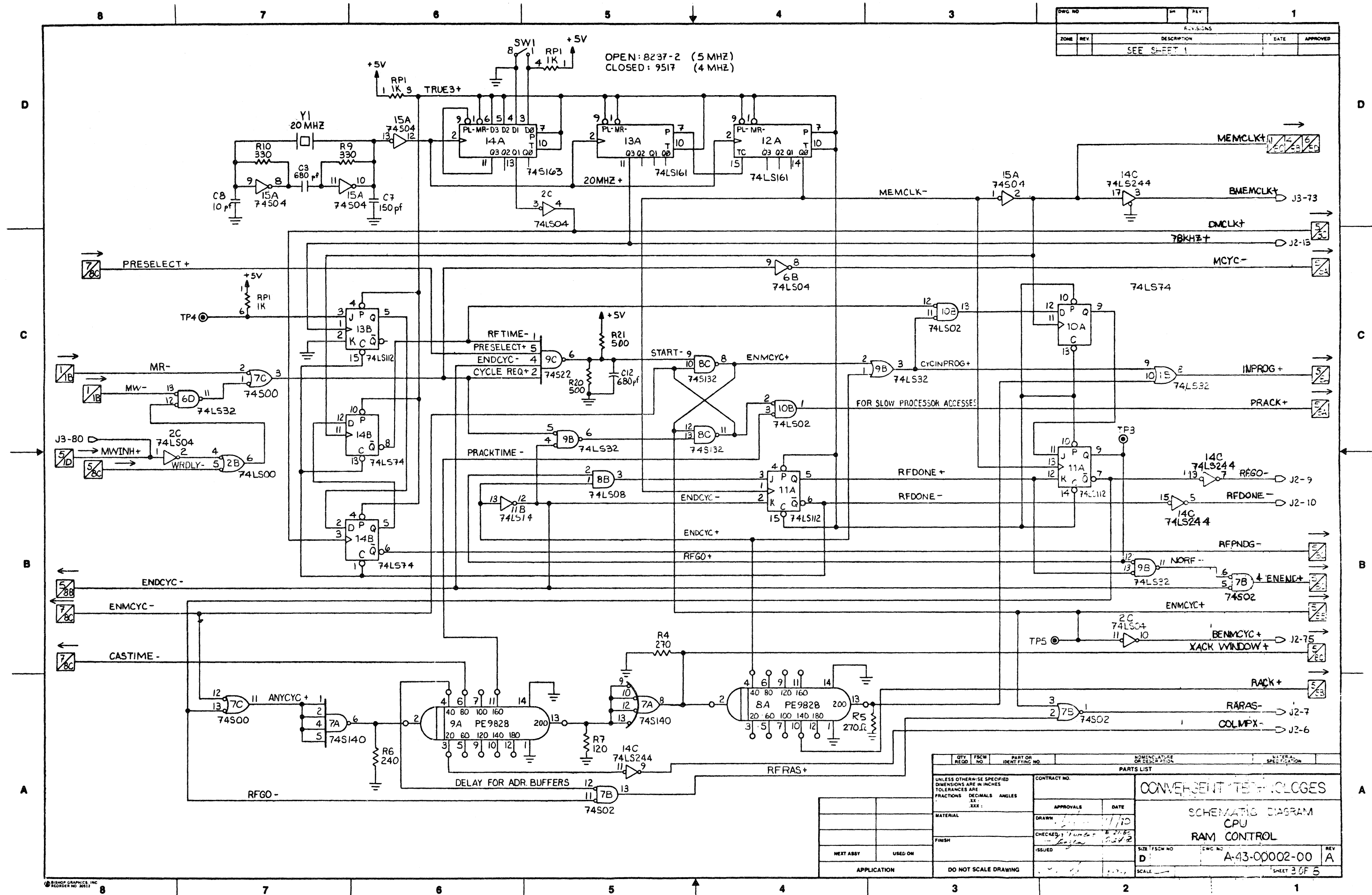
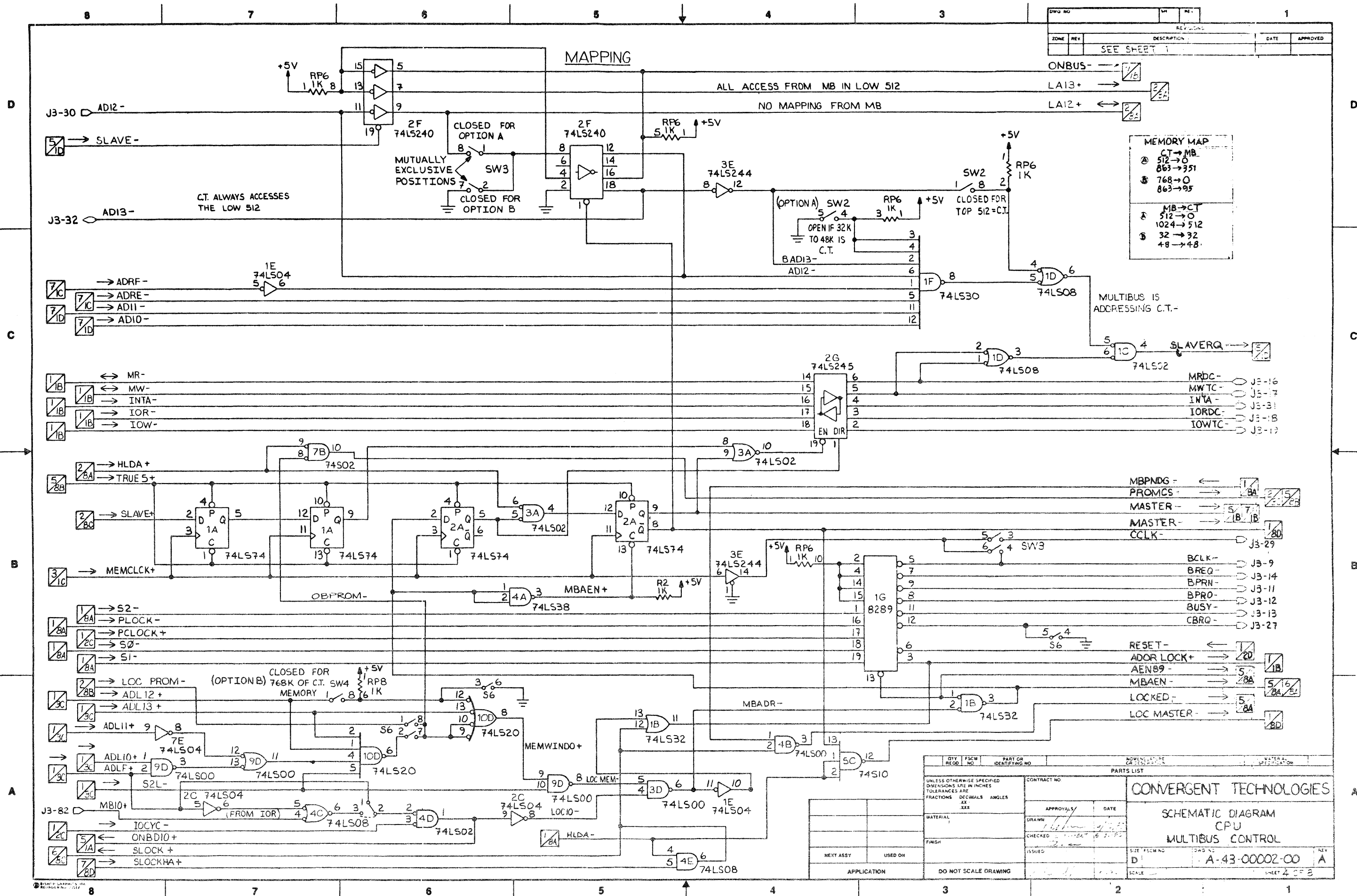


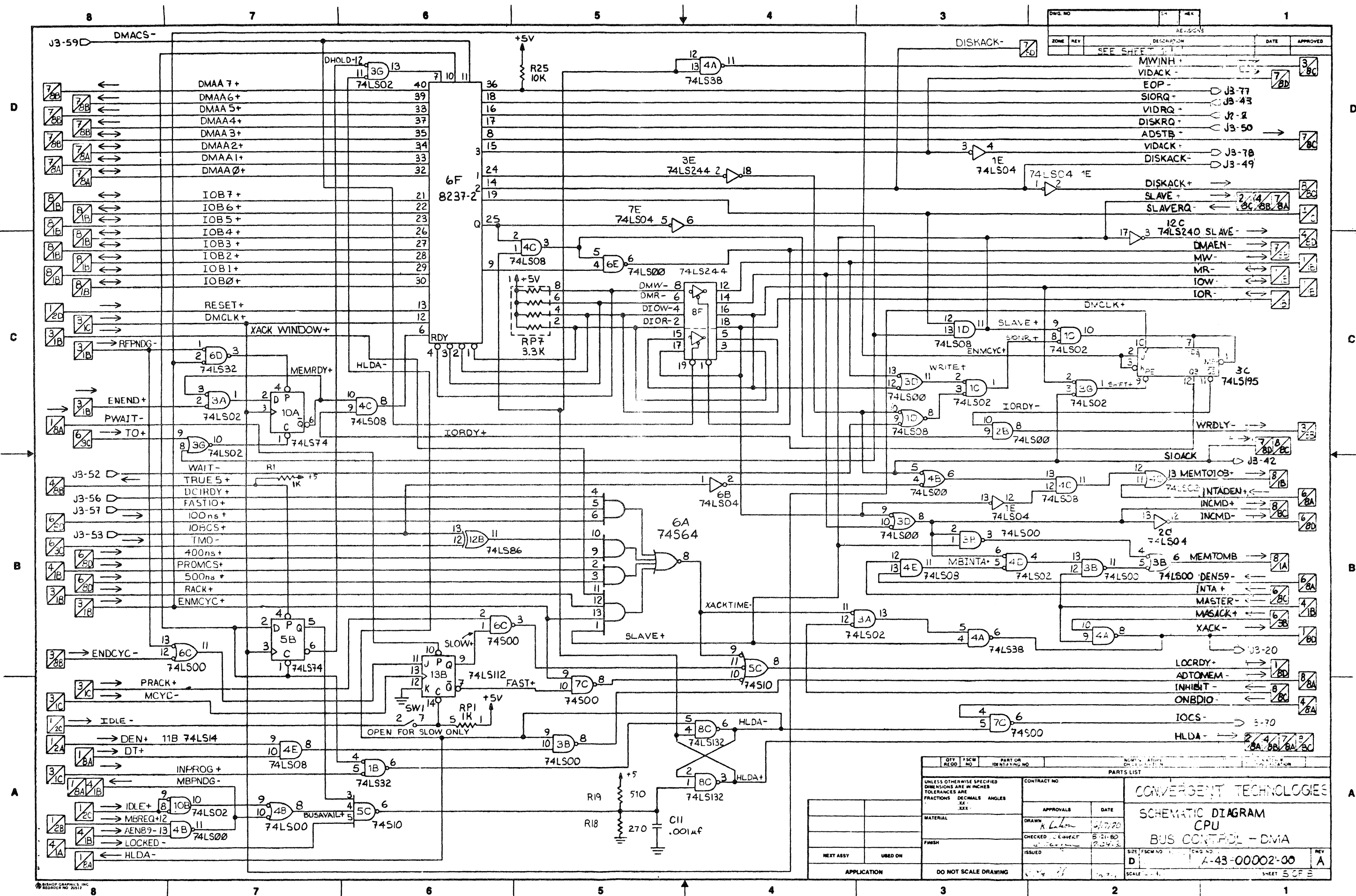
Figure 4-7 8259A Interrupt Controller

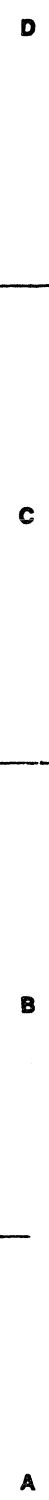


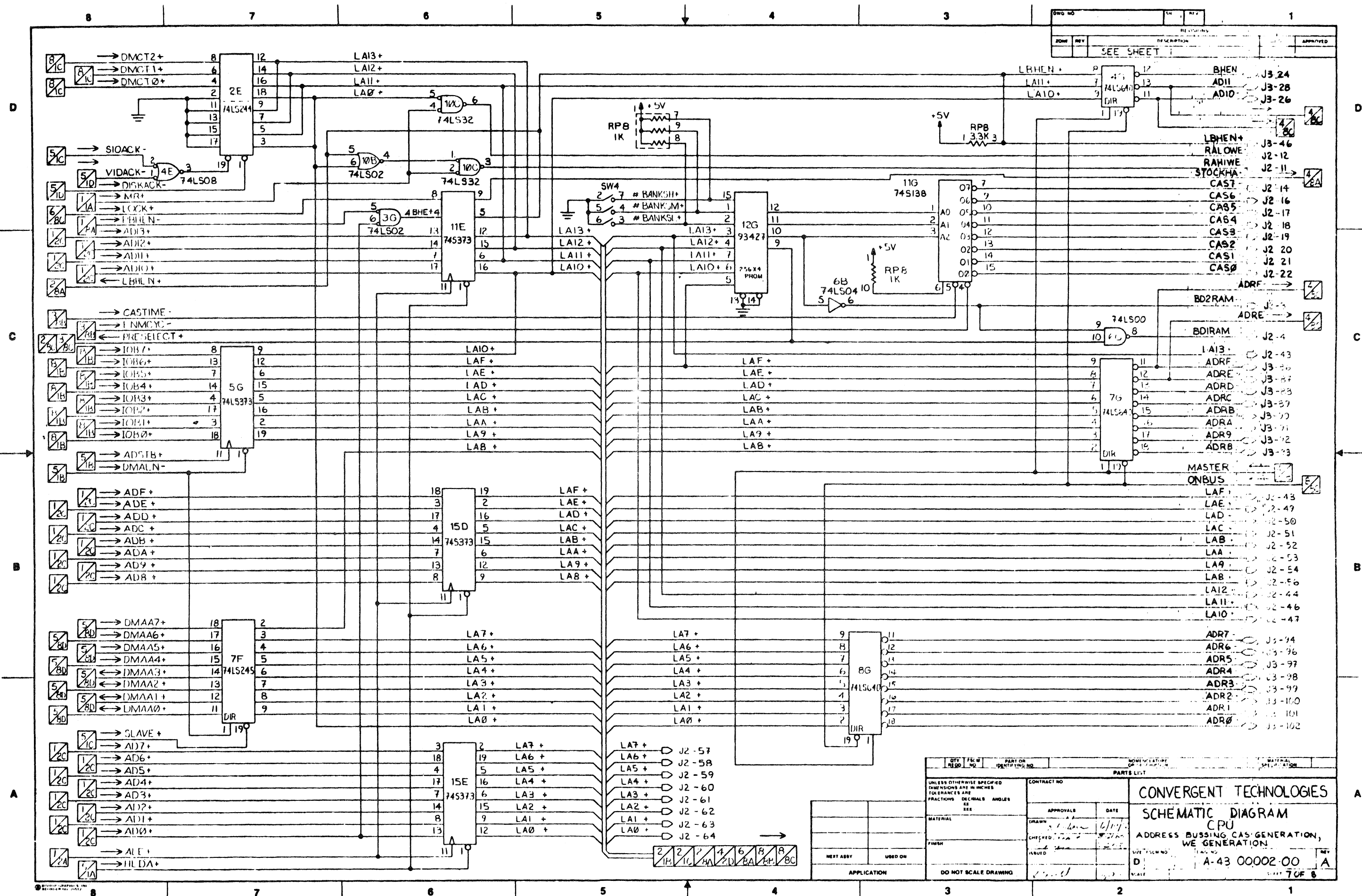


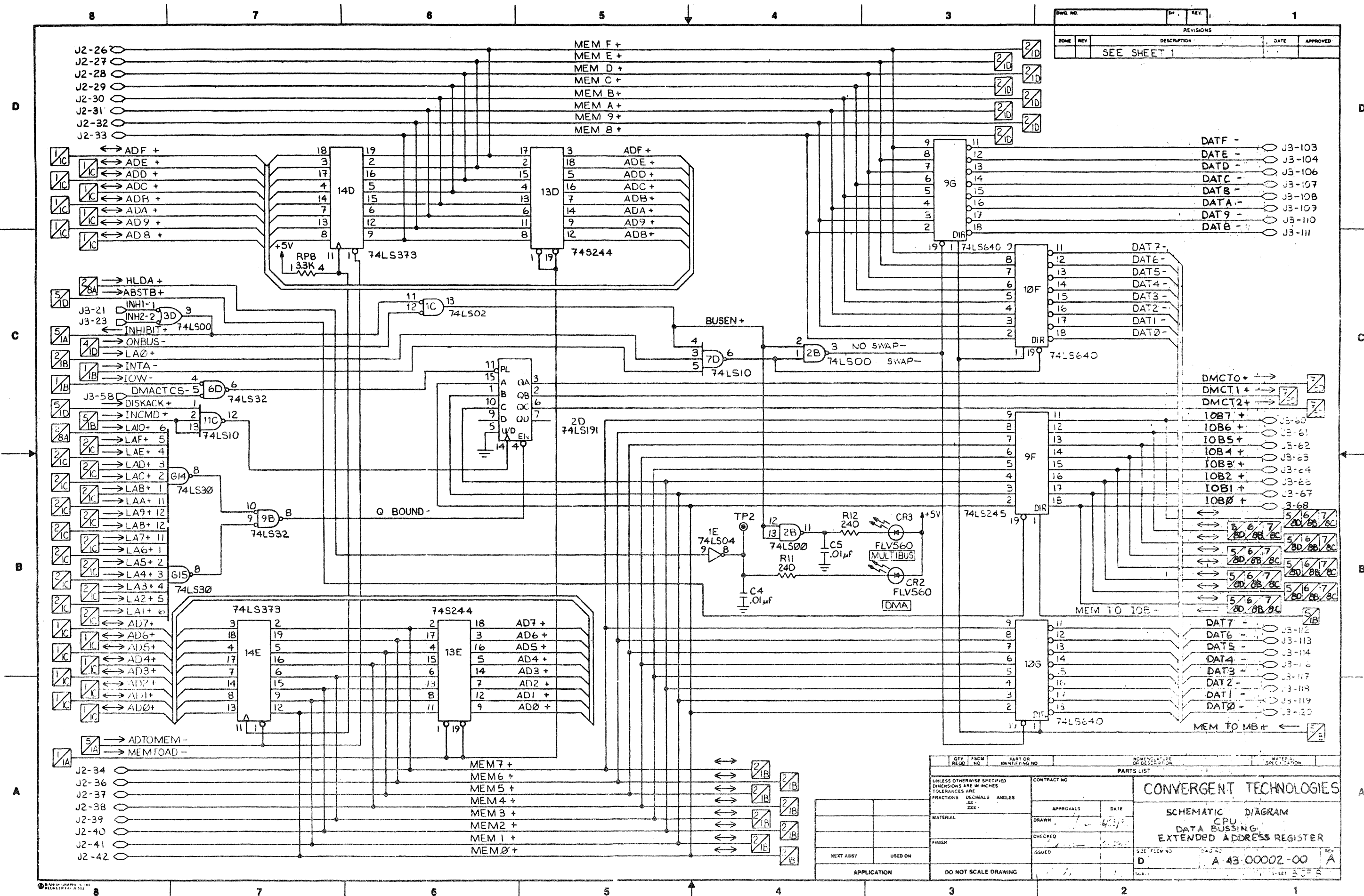












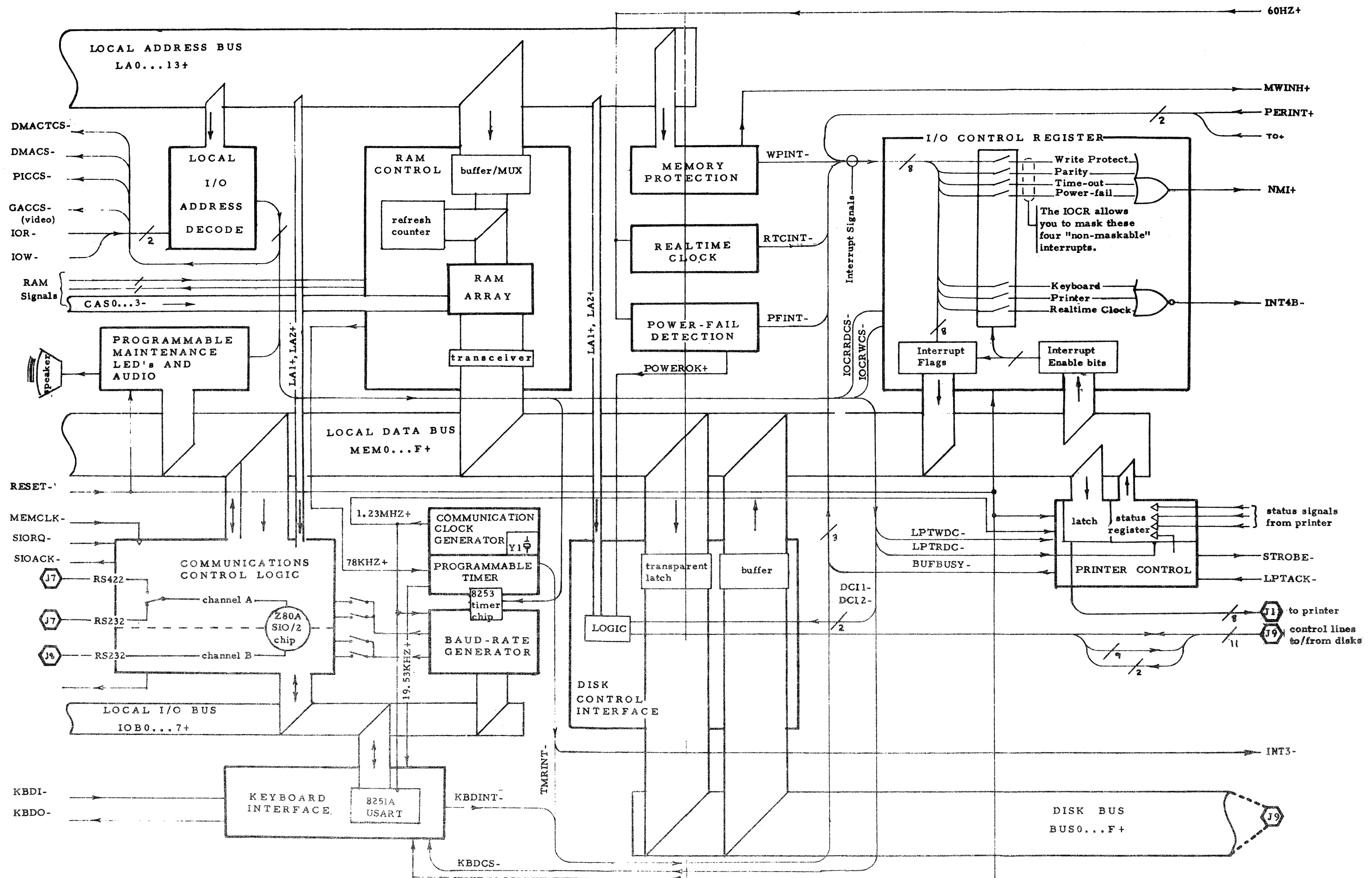
5. I/O-MEMORY BOARD

OVERVIEW

The I/O-memory board is one of two mandatory boards which plug into the mainframe motherboard to implement the minimum-configuration mainframe. It contains main RAM plus control circuitry for all the I/O facilities except for the video logic and the mass storage subsystem.

OPERATION BY LOGIC BLOCK

Figure 5-1, shows how the major logic blocks relate to each other and how they participate in the flow of data on this board. The primary reference, however, is the schematics at the end of this chapter. As in the previous chapter, frequent references are made to schematic page and zone numbers.



The RAM Array

The 7 address lines of the RAM ICs are multiplexed, so that 14 address bits can be represented with 7 pins. The chip is organized as a square matrix of 128 rows x 128 columns. Each row/column combination gives the address of 1 bit, or cell of memory.

The RAS- (row address strobe) signal has two functions: it latches the row address on the 7 address pins and it accesses the specified row of 128 cells. the CAS- (column address strobe) signal also has two functions: it latches the column address, and performs the data operation, read or write, on the one cell specified by the column address of the 128 cells already accessed by RAS-. A write cycle is caused if the WE- pin is asserted at the leading edge of CAS-. The write data at the data input pin of the chip is latched by the leading edge of CAS-. A read cycle is caused if the WE- pin is inactive at the leading edge of CAS-. The read cycle causes the data from the selected cell to be driven on the chip's output pin.

The memory chips used are dynamic RAMs. Data is stored in parasitic capacitance that must be re-charged every 2 milliseconds to prevent leakage from destroying the contents of the memory.

Each time RAS- is asserted the entire 128-cell row corresponding to the state of the address pins is accessed. Whenever the cell is accessed it is re-charged.

In order to insure that all cells of the chip are refreshed (re-charged) every 2 milliseconds there is special refresh circuitry on the processor and I/O Memory boards. This circuitry works by issuing a RAS- every 12.8 microseconds and causing a counter to generate each different row address. CAS- is not issued during refresh, so no data is transferred.

64K RAMs work similarly to the 16K RAMs, except that there are sixteen address bits (one of the power supply pins of the 16K RAM being used as the extra address pin required for the 64K part) and the array is 256 rows of 256 columns.

The RAM array (p1) on the I/O Memory board is organized as 4 banks of 18 chips each. Each bank of 18 chips corresponds to a word. Because the

memory can be accessed a byte at a time, each byte has its own parity bit. So one word is represented as two 9-bit "bytes".

All 72 chips in the array have the same RAS-signal. Bank selection is done by asserting only one of the CAS- signals (CAS0-, CAS1-, CAS2-, CAS3-,). the data-in and data-out pins of each chip are tied together, making a bidirectional data bus (RADO+ - RADF+). Each data bus bit is connected to 4 chips, consisting of the corresponding bit of each word in all 4 banks. The parity bits HIP+ (high-byte parity) and LOP+ (low-byte parity) are connected in the same fashion. The seven address pins (MAD1+ - MAD7+ are connected in parallel to all 72 chips. The signal MAD8+ is an address bit if 64K RAMS are used but is connected to +5 at jumper E-F (p2zB6) for 16K chips. There are two write enable signals, RAHIWE- (high or odd byte write enable) and RALOWE- (low or even byte enable) which cause writing in either the low byte, the high byte, or both. Again, for writing, CAS- selects the particular bank written. Each write enable signal goes to 36 RAM chips.

The RAS- and CAS- signals for read, write and refresh are generated in the memory control of the processor board. When they come on to the I/O Memory board they are called RARAS- (RAM array RAS) (p2zD8) and CAS0- to CAS3- (p1z08), corresponding to banks 0 through 3. At 15H-11 (p2zB5), RARAS- is gated so that only refreshes and accesses to the I/O Memory board (as opposed to the RAM expansion board) generate the signal BD1RAS-. BD1RAS- connects to the RAS- pins of the RAM IC's. RFG- indicates a refresh is in progress and BD1RAM- (from the processor board) indicates the I/O Memory board RAM is being addressed.

There are three sources for the address driven on the RAM chip address pins, the row address (15G (p2zA7)), the column address (14G (p2zB7)), and the refresh row (15F (p2zC7)). The selection of which address driver is enabled is made by the high speed decoder consisting of 14E-6, 8, 11, 13H-11 and 14H-4 and 13 (p2zA7). The RFG0-signal, when active, enables 15F and disables 14G and 15G. The COLMP- (column multiplex) signal causes the row driver (15G) to disable and the column driver (14G) to enable. COLMPX- and RFG0- come from the RAM controller on the processor board. RFDONE- from the processor board

indicates that a refresh cycle has completed and causes the refresh address counters 13F and 14F (p2zC7) to increment.

Data to and from the RAM array is buffered from the MEM bus (MEMO+ - MEMF+) by the transceivers at 10E (p2zD4) and 11E (p2zC4). The direction of the transceivers is established by the signal READ- which is a buffered MR- (memory read) signal. The enable for the high (odd) byte driver is HIEN- and for the low (even) byte is LOEN-. HIEN- is asserted when the I/O Memory board RAM is addressed (BD1RAM- asserted at 13E-3 (p1zA6)) and either a read is in progress (MR-active) or the high-order byte write enable is active (RAHIWE-). LOEN- is asserted when the I/O Memory board RAM is addressed and either a read is in progress or the low-order byte write enable is active (RALOWE-).

Disk Control Interface (DCI)

The disk control interface drives and receives the signals that interface the mainframe to the mass storage subsystem.

During I/O write (IOW-) the latching drivers at 9F and 9H (p3zB4) latch output data. This is done by the BIOW- (buffered IOW-) signal from 6H-3, inverted at 5F-12 (p3zC4). After a delay by R11, C6, and 5F-8, the buffer 6H-16 (p3zB4) makes WRSTB- (write strobe) which goes to the disk controls located in the mass storage subsystem. After another delay by R10, R32, C5, and 5F-6 (p3zC6), the data is driven to the disk controls by 9F and 9H (p3zB4) (BUSO+ - BUSF+). The delays ensure enough data hold and setup time on BUSO+ - BUSF+ for the MOS floppy disk controller.

IOR- is buffered at 6H-12 (p3zB4) to make BIOR- (buffered I/O read). The gate at 10G-6d (p3zB5) enables the BUS to MEM drivers (8F and 8H (p3zC4)) and sends a RDSTB- (read strobe) to the disk controls. 10G-6 is asserted when BIOR- is present and one of the disk controls is addressed (DCI1- or DCI2- active) or disk DMA is active (DISKACK- active). The address decodes, DCI1- and DCI2-, are ORed at 5H-11 (p3zA7) and then they are ORed with DISKACK+ at 4E-13. DISKACK- comes from the DMA logic on the processor board.

The BMR- (buffered memory read) signal at 13H-3 (p3zA3) goes to the disk controls to synchronize their DMA requests. It is a buffered copy of

MR-.

The CNTRL- signal to the disk controls is used to address the control and status registers of the disk controls, rather than the data registers. It is asserted by gate 5H-8 (p3zA6) when DMA is not in progress (DISKACK- inactive) and local address bit LA1+ is high.

The DREQ- (DMA request) signal from the disk controls is buffered at 5F-10 (p3zB5) to make DISKRQ+ which goes to the DMA logic of the CPU board.

The INRQ- (interrupt request) signal from the disk controls causes a level-7 interrupt (INT7-) when gate 3C-4 (p3zB6) is asserted.

The signal ACK- is a bidirectional acknowledge signal between the DCI and the disk controls. During DMA cycles, DISKACK- from the processor board causes gate 5H-3 (p3zA6) to assert ACK-. The disk controls use this as their DMA acknowledge. During non-DMA activity, the disk controls assert ACK- whenever they are addressed (DCI1- or DCI2- active). If a disk control does not assert ACK- when addressed, gate 5H-6 (p3zA6) will cause DCIRDY+ to go false. This causes a bus timeout in the ready logic of the processor board.

The signal EOP- (end of process) from the DMA logic of the processor board indicates that a DMA block is complete. This is buffered at 6H-18 (p3zB4) to generate BEOF- which goes to the disk controls.

The two disk control I/O address decodes, DCI1- and DCI2- are buffered at 6H-5, -7 (p3zB4) to generate SELW- (select winchester) and SELF- (select floppy) for the disk controls.

Inputting from port 74, used for diagnostics, causes the most recent 16 bits of information outputted (IOW- active) to be echoed. This is used to check the BU80+ - BUSF+ drivers and receivers. WRAPDCI+ is asserted at 13H-8 (p4zC3) when port 74 is read (IOR- active, DC12- active, LA2+ active). At 6H-19 (p3zB4), WRAPDC1+ causes the SELF- signal to be disabled to prevent the floppy control from driving BUS0+ - BUSF+ during the input from 74. In addition, 6H-19 causes 6H-9 to be disabled, which in turn causes the DCI drivers 9F and 9H (p3zB4) to be enabled, driving

BUS0+ - BUSF+. Through the normal mechanism of reading the disk control ports, gate 10G-6 (p3zB5) causes the data on BUS0+ - BUSF+ to be driven onto MEM0+ - MEMF+.

When the power fail detect circuit has detected an error, the POWEROK- signal will go inactive. This causes driver IC-9 (p3zC2) to make the ENBL- signal to the disk controls inactive, disabling the disk controls.

I/O Address Decode

The I/O address decode logic generates the signals shown in Figure 5-2. 12E-8 (p4zB4) ORs several of the decodes to generate FASTIO+. FASTIO+ is used on the processor board to indicate that the addressed I/O device does not require wait states. For all other I/O device addresses, IOBCS+ goes active at 13G-6 (p4zB3). This does two things on the processor board. It causes wait states for the I/O instructions that refer to those devices, and for input instructions it causes the data to come from the I/O bus (IOB0+ - IOB7+) instead of directly from the MEM bus.

The I/O Control Register (IOCR)

The IOCR and its associated logic control the several reasons for generating level-4 interrupts and non-maskable interrupts. The IOCR is also used to control parity, memory write protect, and Multibus I/O addressing. The IOCR consists of drivers 11F and 11H, latch 10H, and flip-flops 9G-9, 9G-5, 8G-9, 7G-9 and 6F-9 (p5). The high-order 8 bits of IOCR, corresponding to MEM8+ - MEMF+ are control bits, and the low-order 8 bits corresponding to MEM0+ - MEM7+ are status bits. The control bits are loaded into register 10H from MEM8+ - MEMF+ when there is an output (IOW-) to the IOCR I/O address as described in the I/O address decode section. The register can be read back when ICTLRDC- goes active (IOR- and IOCR address) causing the driver at 11H to drive the IOCR contents onto MEM8+ - MEMF+. Simultaneously ICTLRDC- causes the IOCR status bits to be driven onto MEM0+ - MEM7+ through 11F.

<u>Signal name</u>	<u>Address</u>	<u>Strobe</u>	<u>Source</u>	<u>Device Addressed</u>
PCICS	38-3F	none	12G-7 (p4zB5)	8251A USART
COMMIA-	30-37	none	12G-9 (p4zB5)	SIO Interrupt Acknowledge
TIMERCS-	28-2F	none	12G-10 (p4zB5)	8253 Programmable Timer
PICCS-	20-27	none	12G-11 (p4zB5)	8259A Interrupt Controller
DCI1	78-7F	none	12F-7 (p4zA5)	Hard disk control
DCI2-	70-77	none	12F-9 (p4zA5)	Floppy disk control
DMACTCS-	68-6F	none	12F-10 (p4zA5)	DMA Extended Address Register
COMMF-	60-67	none	12F-11 (p4zA5)	Comm Extended Control Register
GACCS-	58-5F	none	12F-12 (p4zA5)	Video Global Attribute Register
COMMS-	48-4F	none	12F-14 (p4zA5)	SIO communication chip
PPICS-	40-47	none	12F-15 (p4zA5)	The LEDs/Speaker/and switches
LPTRDC	50-51	IOR-	11G-4 (p4zB3)	Line printer status
RDLOPEA-	52-53	IOR-	11G-5 (p4zB3)	Low order parity error address
RDHIPEA-	54-55	IOR-	11G-6 (p4zB3)	High order parity error address
ICTLRDC-	56-57	IOR-	11G-7 (p4zB3)	IOCR reading
DMACS-	00-1F	none	6E-8 (p4zB3)	8237 DMA controller
LPTWDC-	50-51	IOW-	11G-12 (p5zB7)	Line printer output
ICTLWDC-	56-57	IOW-	11G-9 (p5zB7)	IOCR writing
WRAPDCI+	74-77	IOR-	13H-8 (p4zC3)	DCI wraparound (diagnostics)

Figure 5-2. I/O Address Decode Signals

Control Bits Of The IOCR (read/write at port 56)

- Bit F: PFEN+ - Power failure interrupt enable. This bit when 1 causes PFEN+ to be asserted. When PFEN+ is high, 6F-13 (p5zA6) goes high, enabling flop-flop 6F-9. If the power fail one-shot times out, it will clock 6F-11 setting 6F-9 and causing PFINT- (6F-8) to go active. PFINT- then causes 8E-8 (NMI+) to be asserted, making a non-maskable interrupt. The power fail interrupt is removed by setting this bit to zero, thereby clearing 6F-9.
- Bit E: PAREN+ - Parity interrupt enable. This bit, when 1, enables the parity checking circuitry on the processor board (see its description). If the parity checking circuitry detects an error, it will assert PERINT+. PERINT+ will then cause a non-maskable interrupt through inverter 7E-12 and gate 8E-8. PERINT+ is cleared by setting this bit (and PAREN+) to 0. Note that parity generation is always enabled.
- Bit D: ODDP+ - Odd parity generate and check. This inverts the parity generated and checked by the processor board from the normal, even parity.
- Bit C: MBIO+ - Multibus I/O enable. Normally this does nothing. If a special jumper configuration is used, it enables Multibus I/O addressing. See the processor board Multibus Master mode description.
- Bit B: PRINTEN+ - Printer interrupt enable. This bit, when 1 allows the printer interrupt flip-flop LPTINT+ (9G-9) to be set when the BUFBUSY+ flip-flop in the printer interface is cleared by an acknowledge from the line printer. Setting LPTINT+ causes gate 8E-6 to assert INT4B-, making a level 4 interrupt. The LPTINT+ flip-flop is cleared by setting this bit to 0.
- Bit A: TIMEN+ - Bus timeout interrupt enable. This bit when 1 allows the TIMINT+ flip-flop (g-5) to set if the

bus timeout logic of the processor board detects a timeout. Setting TIMINT+ causes 8E-8 to generate a non-maskable interrupt (NMI+). The timeout interrupt is cleared by setting this bit to zero.

Bit 9: RTCEN+ - Real time clock interrupt enable. This bit, when 1, allows the RTCINT+ flip-flop (8G-9) to set at the next RTC+ strobe (see RTC description). Setting RTCINT+ causes gate 8E-6 to generate a level 4 interrupt (INT4B-). Setting this bit to 0 clears RTCINT+, clearing the Real Time Clock interrupt.

Bit 8: WPEN+ - Write protect enable. This bit enables the memory write protection logic (see its description). If a memory write protection violation occurs, flip-flop 76-9, WPINT+ will be set. This causes a non-maskable interrupt (NMI+ at 8E-8). When this bit is zero it disables the write protection circuitry and clears the WPINT+ flip-flop.

Status Bits Of The IOCR (read only at port 56)

Bit 7: PFINT+ (p5zA4) - Power fail Interrupt. This bit indicates the status of the parity error detect line from the processor board, see bit F description.

Bit 6: PERINT+ (p5zD8) - Parity error interrupt. This bit indicates the status of the parity error detect line from the processor board, see bit E description.

Bit 5: TMRINT+ (p5zC3) - Programmable timer interrupt. This bit, when one, indicates that the programmable timer (8253 timer 0) is causing a level 4 interrupt.

Bit 4: MBIO+ (p5zB2) - Multibus I/O enable. This bit just reflects the setting of bit C.

Bit 3: LPTINT+ (p5zD4) - Line printer interrupt. This bit indicates the

status of the line printer interrupt flip-flop 9G-9. See bit B description.

- Bit 2: TIMINT+ (p5zD4) - Bus timeout interrupt. This bit indicates the status of the Bus timeout interrupt flip-flop 9G-5, see bit A description.
- Bit 1: RTCINT+ (p5zD4) - Real time clock interrupt. This bit indicates the status of the Real time clock interrupt flip-flop 8G-9. See bit 9 description.
- Bit 0: WPINT+ (p5zD4) - Write protection interrupt. This bit indicates the status of the write protect interrupt flip-flop 7G-9. See bit 8 description.

Speaker, LEDs and Switches

The register at 7F (p4zB7) is loaded with the contents of MEM0+ - MEM7+ when an output (IOW-active) is addressed to it (PPICS- active). The bits in the register corresponding to MEM0+ - MEM5+, when 0 cause LEDs CR6 - CR1 to light. Bits 6 and 7 of the register affect the speaker as follows:

Bit 7	Bit 6	Effect
0	0	Speaker not energized
0	1	Speaker oscillates at 625 Hertz
1	0	Speaker not energized
1	1	Speaker energized

The speaker is controlled by gates 10G-8 and 15H-6 (p4zA6). Q1 is used to amplify the output of 15H-6. The driver at 1A (p4zD2) is enabled when an input is done (IOR- active) and it is addressed (PPICS- asserted). It causes the setting of the 8 bit dip switch at 2A (p4zC7) to be read.

Programmable Timer

The chip at 2D (p4zC5) is an 8253 programmable timer chip. It is programmed, via the I/O bus (IOB0+ - IOB7+) when its chip select pin is active (2D-21) and IOR- or IOW- is active. The local address bits LA1+ and LA2+ select the internal register to be read (IOR-) or written (IOW-). Counter 1 and 2 of the 8253 are used as programmable baud rate timers to divide the

1.23MHZ+ output of the communications oscillator and make C1CLK+ (counter one clock) for comm channel B and C2CLK+ (counter two clock) for comm channel A. The other counter, counter 0, is used to generate interrupts on interrupt level 3 at a programmable rate. The input clock frequency for counter 0 is a 19.53 KHz+ clock. It is generated by dividing the 78KHz+ clock (from the processor board) by 4. The output of counter 0 also goes to the IOCR (see its description) to be read by software.

The communications oscillator consists of 2.4576 MHz crystal Y1, R22, C7, C19, C20, and gates 3A-8, 6D-12, and 6E-2 (p4zD4). It is divided in 2 by flip-flop 7G-5 to make the input clock for the baud rate generator counters in the 8253 (2D-18 and 2D-25).

Keyboard Interface

The 8251A USART (Universal Synchronous Asynchronous Transmitter) at 1D (p4zD5) is used for communications with the keyboard microprocessor. It is programmed using the I/O bus (IOB0 - IOB7+) when its chip select is active at 1D-11 and IOR- or IOW- is active. Local address bit LA1+ is used to select among the registers within the chip. The baud rate source for both transmitting to and receiving from the keyboard is the 19.53KHz+ clock from counter 3D (p4zC4). The gate 4E-1 asserts KBDINT- (keyboard interrupt) when the 8251A is ready to transmit (1D-15 high) or has received a character (1D-14 high). The jumper G-H can be cut and H-I installed to prevent transmit interrupts. KBDINT- is used in the IOCR logic (see its description) to cause level 4 interrupts. Serial data to the keyboard is driven by 1C-7 and the data from the keyboard is received at 1C-15 (p4zD4).

Line Printer Interface

As described in the address decode section, the signal LPTWDC- goes active when an output (IOW-) is done to the line printer I/O address. LPTWDC- causes the line printer data to be loaded into the latch at 10F (p5zD5) from the MEM bus (MEM0+ - MEM7+). This drives the data lines to the printer. LPTWDC- also clears the counter at 5E (p5zA6), starting the "STROBE- sequence". At least 800 nanoseconds after LPTWDC- ends, 5E-13 goes active to assert the STROBE- signal to the

printer, causing it to accept the data from latch 10F. STROBE- then stays active for 1.6 microseconds before returning to the inactive state. The inverter 6E-6 holds it inactive until the next LPTWDC-. Besides strobing the printer, STROBE- causes the BUFBUSY+ (buffer busy) flip-flop to set at 8G-5 (p5zA6). This indicates that data has been sent to the printer but has not been acknowledged. The line printer acknowledges receipt of the character by asserting LPTACK-. LPTACK- clears the BUFBUSY+ flip-flop at 8G-1. LPTRDC- goes active when an input (IOR-) is done from the line printer I/O address as described in the I/O address decode section. LPTRDC- causes status information to be driven on MEM bus bits MEM0+ - MEM3+. The status includes BUFBUSY+, and three signals from the line printer, NOPAPER+, LPTSELECT+, and LPTBUSY+. NOPAPER+, although not supported on many printers, indicates that there is no paper in the printer. LPTSELECT+ indicates that the printer is online, and LPTBUSY+ when asserted means that the printer is doing something mechanical that will take several milliseconds. The complement of BUFBUSY+ is used in the IOCR logic (see its description) to generate level-4 interrupts.

Real Time Clocks

60HZ+ is a signal from the power supply that oscillates at the line frequency. C1, C2, R6, and gate 7E-8 (p5zC5) filter 60HZ+ to make RTC+ (real time clock). RTC+ goes to the IOCR logic (see its description) at 8G-11 to generate level 4 interrupts and is also used in the power fail detect logic.

Power Fail Detect

The one-shot at 6G-9 (p5zA6) is triggered on each positive or negative transition of RTC+, 120 Hz or 100 Hz depending on line frequency. If the one-shot times out it signals the IOCR logic (see its description) at 6F-11 (p5zA6) to cause a non-maskable interrupt. Also the signal POWERLOK+ at 6G-10 goes to the DCI to disable the disk controls.

Write Protection

The memory write protection logic allows the first 32K bytes of RAM memory to be read-only. This is used to protect the operating system from being accidentally written on. If bit 8 of the

IOCR is one, and local address bits LAF+ - LA13+ are 0 (indicating an address in the first 32K of RAM) the signal MWINH+ (memory write exhibit) will go active at 12H-2 (p5zB4). The processor board pulls down MWINH+ (it is an open collector signal) during DMA (see the processor board DMA description), allowing DMA writes into the first 32K. The MWINH+ signal when active prevents the memory control on the processor board from responding to memory write (MW-) commands. In addition to this, if MWINH+ is active and a memory write is attempted (MW- goes active), flip-flop 7G-9 (WPINT+) (p5zD4) in the IOCR will be set causing a non-maskable interrupt.

Communications

The communications logic is based on the Z80A-SIO/2. The "SIO" provides synchronous or asynchronous serial communications channels, channel A and channel B. In addition to the provision of the SIO, there is a register in the comm logic (the ECR, port 60) which is used to augment the SIO. Channel A can operate in DMA or programmed data transfer mode, channel B operates in programmed data transfer mode only.

The details of the synchronous and asynchronous protocols and signals is discussed in the Software Interface sections of Chapter 2. The RS232 and RS422 standards and especially the technical manual for the Z80A-SIO may be consulted for information about the protocols.

The flip-flop 2C-5 and 2C-9 and gates 4D-11 and 4D-3 (p6zB6) generate a square 3.33 Megahertz clock by dividing MEMCLK+ from the processor board by 3. This is used as a general purpose clock by the SIO (4F-20) (p6zC6).

The SIO transfers its data and control information to memory and CPU via the I/O bus (IOB0+ - IOB7+).

Non-DMA Transfers Between The 8086 And The SIO

When the SIO is not doing DMA, SIOACK- (p6zC8) will be inactive. This allows local address bits LA1+ and LA2+ and chip select COMMS- to go directly to the SIO pins 33, 34, and 35 respectively (p6zD6). Pin 4F-35 selects the SIO to communicate over IOB and pins 33 and 34, corresponding to LA1+ and LA2+ select the register and channel to transfer within the

chip. Registers are read and written within the IOR- and IOW- strobes. If either IOR- or IOW- is active gate 4C-8 will cause the IORQ- (I/O request) pin 4F-36 to go active. 4A-10 is always active when not doing DMA (SIOACK- inactive) passing IOW- through to 4C-9.

A special case occurs when a program wishes to issue an M1- pulse (Z80A interrupt acknowledge) to the SIO. In that case, although the program does an input from port 30 causing the chip select COMMIA- (p6zC8) to go active, the gate 4D-8 (p6zC7) gates off the IOR- pulse.

When the SIO wishes to interrupt the 8086, it does so by asserting its INT- pin 4F-5 (p6zC6), asserting INT1B- and causing a level 1 interrupt.

DMA Transfers Between Memory And The SIO

The SIO requests a DMA cycle by pulling down its wait/ready pin 4F-10. This causes inverter 3E-10 to make a DMA request (SIORQ+) to the DMA logic of the processor board for the SIO DMA channel. When the processor board asserts SIOACK- (SIO DMA Acknowledge) an SIO DMA cycle starts. Gates 4C-3, 6, and 11 (p6zD6) cause the chip select and addressing pins of the SIO to address the channel A data register. The DMA logic then pulses IOR- to read data from the SIO onto IOB0+ - IOB7+, or pulses IOW- to write data from the bus into the SIO. The gate 4A-11 causes the IOW- to be delayed until the memory read cycle is going (ENMYCYC- from the memory control on the CPU asserted) so that the data will be on IOB when the SIO latches it.

Programming The Extended Control Register (ECR)

The Extended Control Register (ECR), 5G and 4H (p6zA6) is read and written at port 60 (96 decimal). The low order 4 bits, corresponding to MEM0+ - MEM3+, are read/write control bits. The high order 8 bits are read only status bits.

The low order 4 bits are set when an output (buffered I/O write, BIOR-, active) is done to port 60 (96 decimal) (COMMF- active) causing gate 4E-4 and inverter 1C-3 to load register 4H (p6zA6). These bits are readable by inputting (BIOR- active) from port 60 (96 decimal) (COMMF- active) causing gate 4D-6 to enable driver 7H at pin 1 (p6zA5). At the same time the signal from 4D-6 causes the high order 8 bits of MEM (MEM8+ -

MEMF+) to be driven with the status bits from the interface by driver 5G (p6zB6). The Extended Control Register bits are illustrated in Figure 5-3.

Multidrop Driver Control

Bit 2 of the ECR is used to control RS422 drivers in a multidrop environment, such as the CT cluster. When it is 1, the XMIT+ signal goes active. The setting of XMIT+ causes the RS422 drivers to be enabled at 1G-4. The XMIT+ signal can also be read through bit F.

Multidrop Receive Clock Detect

For channel A, the dip switches S3 bits 1 and 2 (p6zB5) determine the source for Carrier Detect. If switch 2 is closed Carrier Detect comes from the RS232 receiver at 3F-8 (p6zB2).

If switch 2 is open and switch 1 is closed it comes from the receiver clock detect circuit. The receive clock detect circuit is enabled when bit 3 of the ECR is 1. This allows the one-shot at 6G-7 (p6zB5) to trigger as long as a receive clock (RXCA-) is active. The output of the one-shot goes through switch 3 to become Data Carrier Detect for channel A (DCDA-). This feature is used in multidrop software to rapidly determine that the line is unused.

Baud Rate Clock Source Selection

There are four different communication clocks used, a receive clock and transmit clock for each channel A and channel B. Each one of these clocks can be generated on-board, by the 8253 (see the programmable timer description), or externally. The choice is made with 8 bit dipswitch S1 (p6zD5) as follows:

<u>Switch 1 bit # closed</u>	<u>Causes</u>
1 (pins 1 and 16)	Channel A external transmit clock
2 (pins 2 and 15)	Channel A external receive clock
3 (pins 3 and 14)	Channel A internal transmit clock
4 (pins 4 and 13)	Channel A internal receive clock
5 (pins 5 and 12)	Channel B external transmit clock
6 (pins 6 and 11)	Channel B external receive clock
7 (pins 7 and 10)	Channel B internal transmit clock
8 (pins 8 and 9)	Channel B internal receive clock

<u>Bit</u>	<u>Input/Output</u>	<u>Signal</u>	<u>Location</u>	<u>Meaning</u>
0	Input/Output	STDB+	4H-12	Secondary transmit data channel B
1	Input/Output	STDA+	4H-15	Secondary transmit data channel A
2	Input/Output		4H-16	RS422 driver enable
3	Input/Output		4H-19	Clock detect enable
8	Input	SRDB+	5G-13	Secondary Receive Data channel B
9	Input	DSRB-	5G-15	Data Set Ready channel B
A	Input	RINGB-	5G-17	Ring indicator channel B
B	not used			
C	Input	SRDA+	5G-8	Secondary Receive Data channel A
D	Input	DSRA-	5G-6	Data Set Ready channel A
E	Input	RINGA-	5G-11	Ring indicator channel A
F	Input	XMIT+	5G-2	RS422 driver enabled

Figure 5-3. The Extended Control Register Bits

See the section about the 8253 concerning the generation of internal baud rate clocks.

Channel B Level Changers (Electrical interface)

Channel B has an RS232 interface only. The receivers at 1H and 2H (p6zC4) convert RS232 signals to TTL signals. RS232 uses a negative voltage to represent a 1 and a positive voltage to represent a 0. The driver at 3H (p6zD5) converts TTL levels to RS232 levels.

Channel A Level Changers

Channel A has an RS232 or an RS422 interface. When jumper plugs JP1 and JP3 (p6zB3) are installed, the RS422 drivers (1G) and the receivers (1F) are used. When jumper plugs JP2 and JP4 (p6zB2) are installed, the RS232 drivers (3G) and the RS232 receivers (2E and 3F) are used.

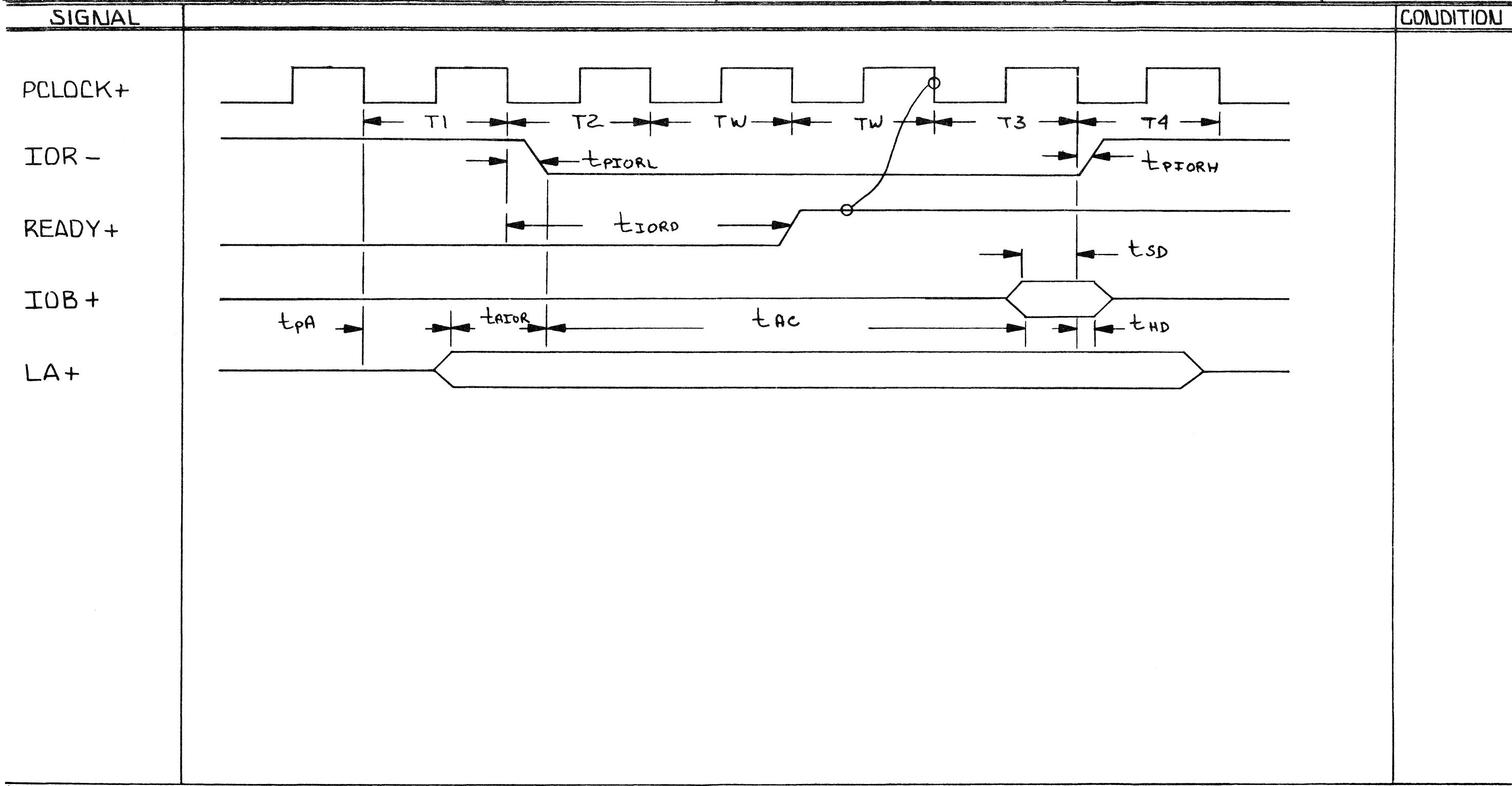
AC CHARACTERISTICS



TITLE PROCESSOR TIMING
IOR, I/O BUS TIMING

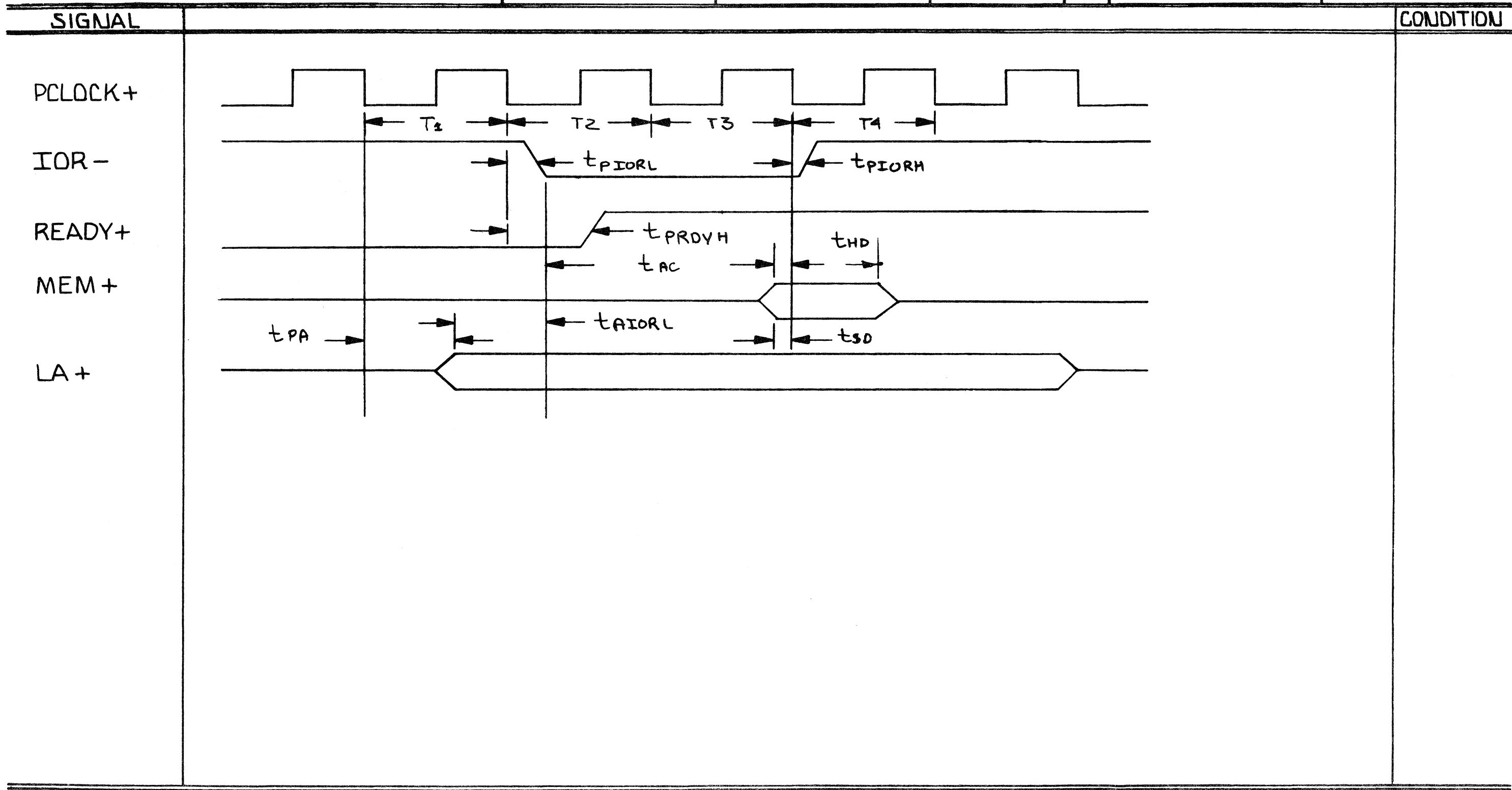
APPROVALS	DATE	DWG NO
DRAWN SKTANGEN	10-20-80	
CHECKED		

APPLICATION		SHEET	REV
NEXT ASSY	USED ON		
ENG			



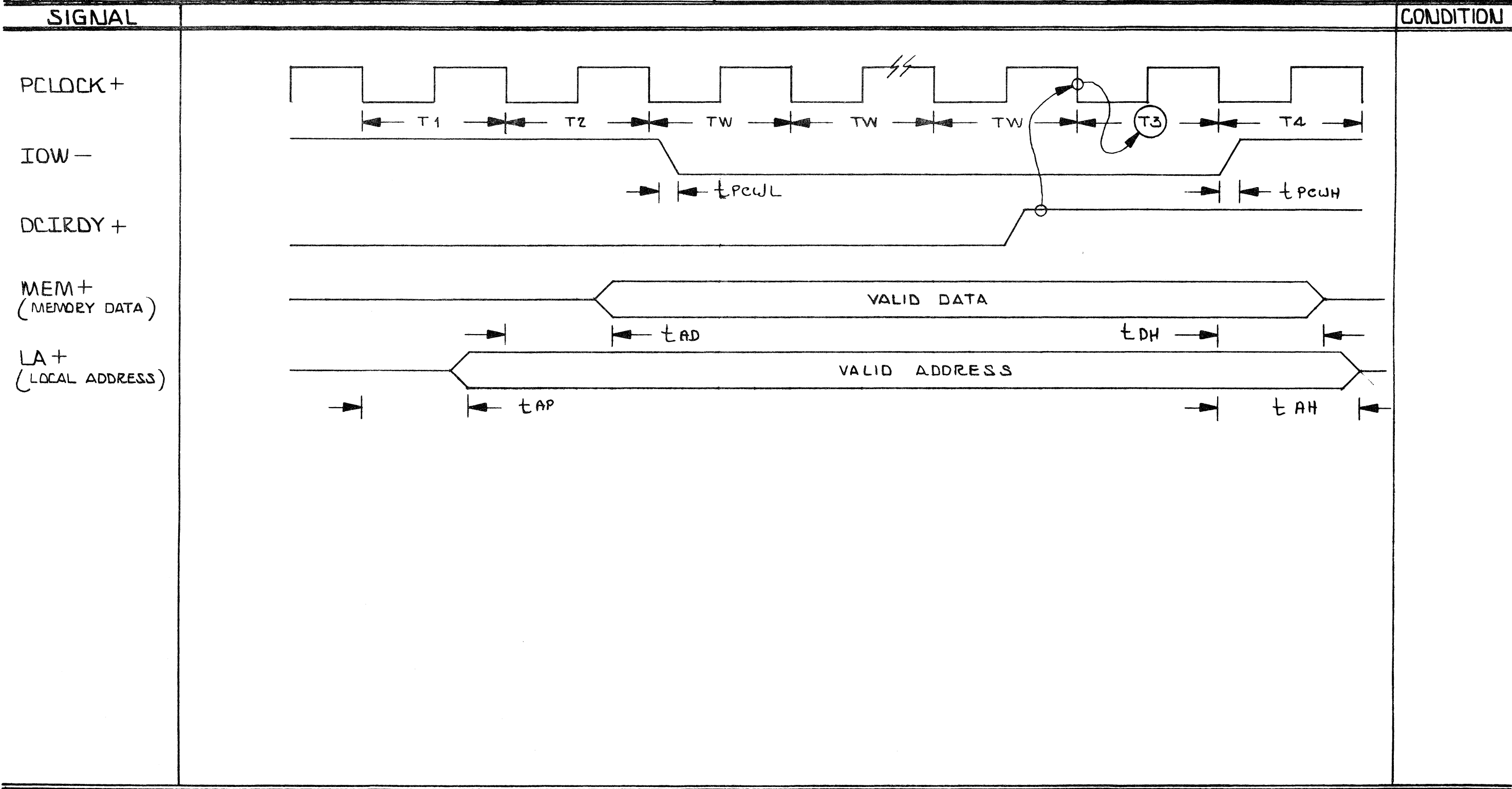


TITLE PROCESSOR TIMING IOR, MEM BUS TIMING (DCIRDY=1)		APPROVALS	DATE	DWG NO	
		DRAWN SLTANGEN	6/20/80		
		CHECKED			
APPLICATION				SHEET	REV
NEXT ASSY	USED ON	ENG			



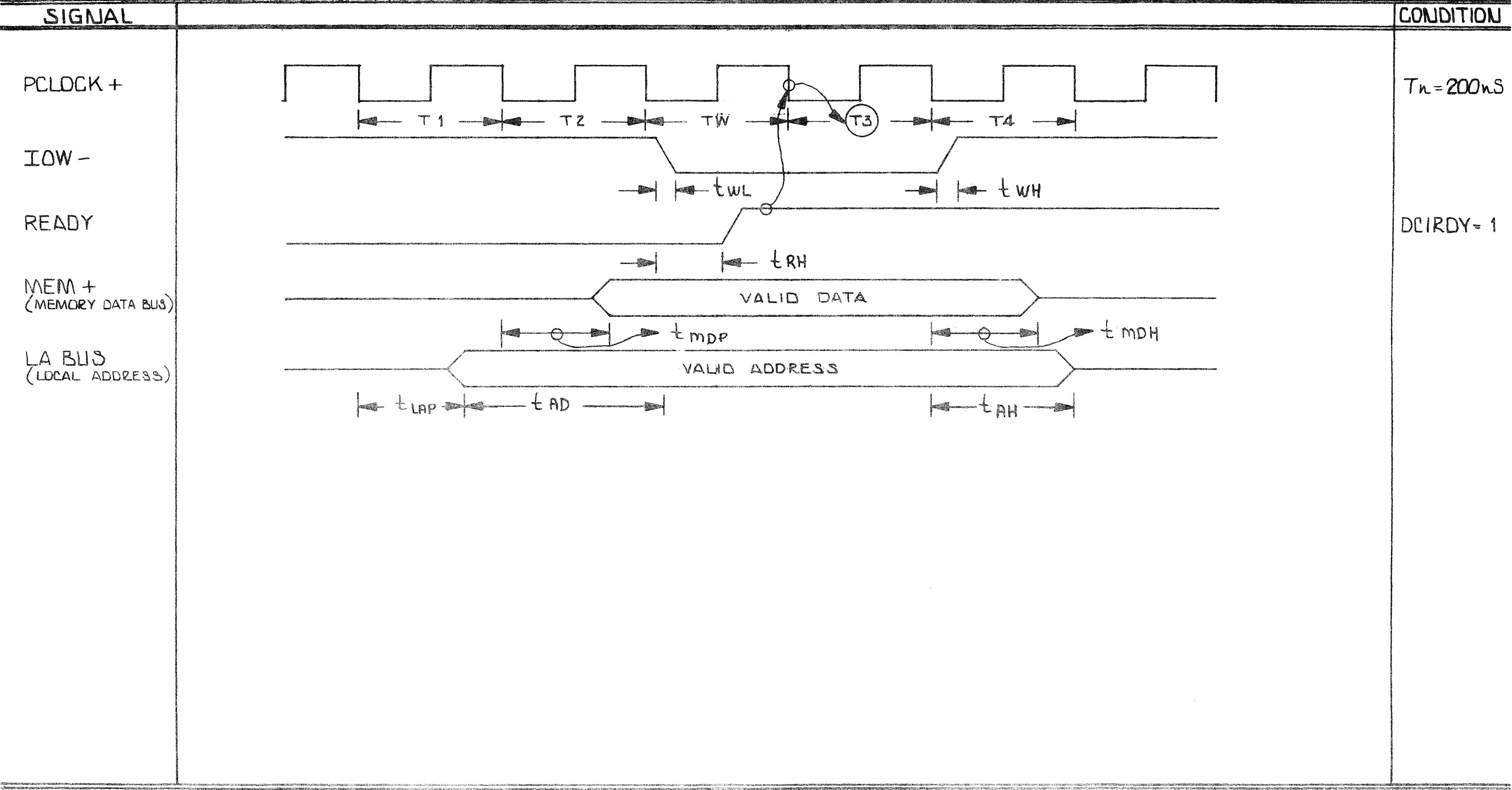


TITLE PROCESSOR TIMING IOW, MEMBUS (ASYNCHRONOUS)		APPROVALS	DATE	DWG NO	
		DRAWN JKTANGEN	10-18-80		
		CHECKED			
APPLICATION				SHEET	REV
NEXT ASSY	USED ON	ENG			



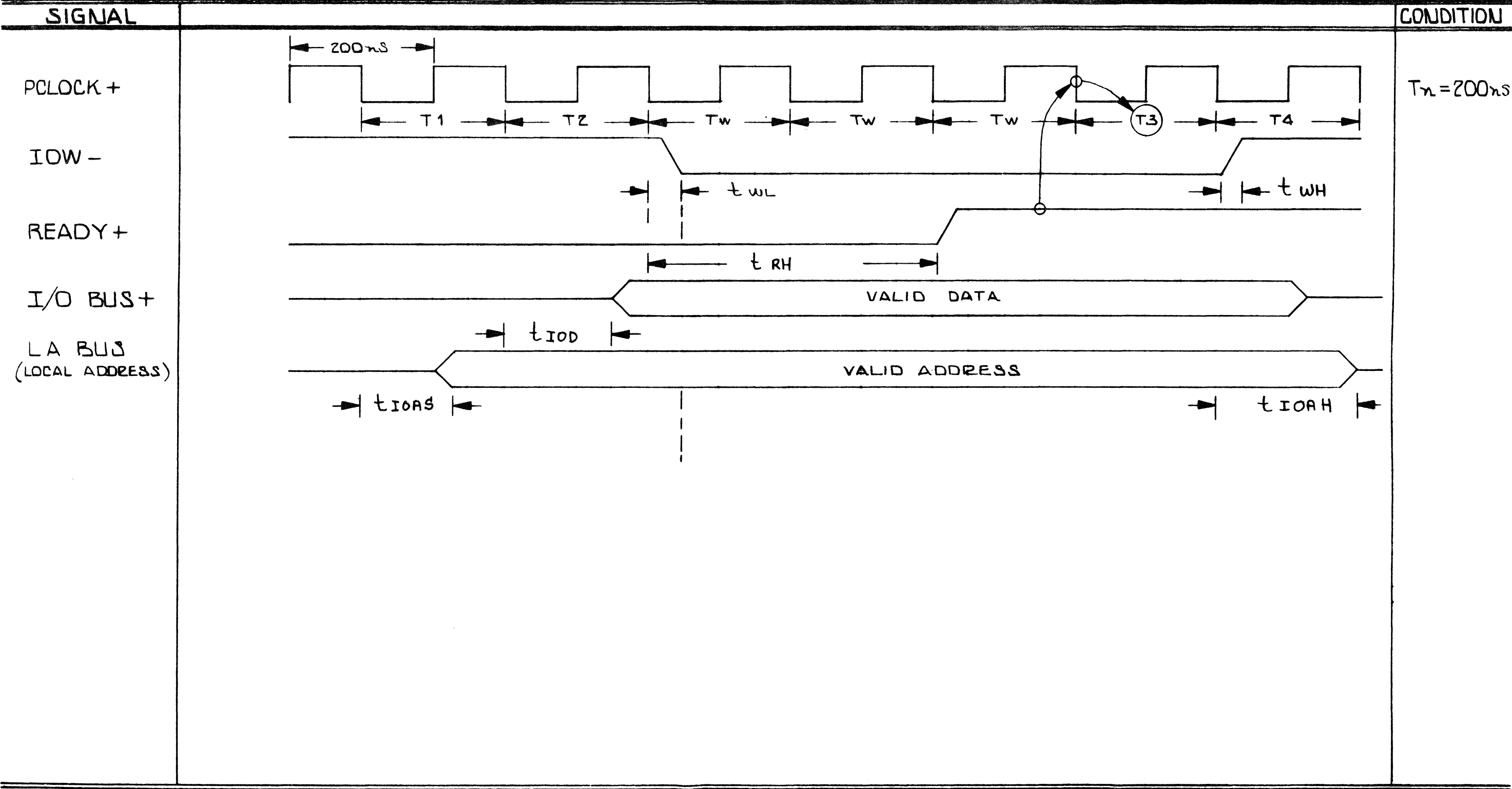


TITLE PROCESSOR TIMING IOW, MEMBUS TIMING		APPROVALS	DATE	DWG NO	
		DRAWN SKTANGEN	6-18-80		
APPLICATION		CHECKED		SHEET	REV
NEXT ASSY	USED ON	ENG			





TITLE PROCESSOR TIMING IOW, I/O BUS TIMING		APPROVALS	DATE	DWG NO	
		DRAWN SKTANGEN	10-17-80		
APPLICATION		CHECKED		SHEET	
NEXT ASSY	USED ON	ENG		REV	



AC-DC CHARACTERISTICS.

[illegible]

AC-DC CHARACTERISTICS

[illegible]

AC-DC CHARACTERISTICS

[illegible]

AC-DC CHARACTERISTICS

[illegible]

IOW, MEM BUS

AC-DC CHARACTERISTICS

[illegible]

IOW, IO BUS' TIMING

DC CHARACTERISTICS

AC-DC CHARACTERISTICS

[illegible]

REFERENCES

Edge-Connector Pin List

Connector J4

<u>Pin</u>	<u>Signal</u>
1	MR-
2	NO PAPER+
3	LPTBUSY+
4	BD1RAM-
5	GND
6	COLMPX-
7	RARAS-
8	BEOP-
9	RFGO-
10	RFDONE-
11	RAHIWE-
12	RALOWE-
13	78KHZ+
14	60HZ+
15	
16	LPT SELECT+
17	AUDIO+
19	CAS3-
20	CAS2-
21	CAS1-
22	CAS0
23	HIP+
24	LOP+
25	
26	MEMF+
27	MEME+
28	MEMD+
29	MEMC+
30	MEMB+
31	MEMA+
32	MEM9+
33	MEM8+
34	MEM7+
35	
36	MEM6+
37	MEM5+
38	MEM4+
39	MEM3+
40	MEM2+
41	MEM1+
42	MEM0+
43	LA13+
44	LA12+
45	
46	LA11+

47	LA10+
48	LAF+
49	LAE+
50	LAD+
51	LAC+
52	LAB+
53	LAA+
54	LA9+
55	
56	LA8+
57	LA7+
58	LA6+
59	LA5+
60	LA4+
61	LA3+
62	LA2+
63	LA1+
64	LA0+
65	
66	
67	+12V
68	+12V
69	+12V
70	+12V
71	
72	
73	-12V
74	-12V
75	
76	BMR-
77	+5V
78	+5V
79	+5V
80	+5V

Connector J5

±

<u>Pin</u>	<u>Signal</u>
1	+5V
2	+5V
3	+5V
4	+5V
5	-12V
6	-12V
7	GND
8	
9	TCLK-
10	TCLK+
11	RXDA-
12	CTSA+
13	RXCA+
14	TXCA+
15	

16	DSRA+
17	SRDA-
18	STDA-
19	TXDA-
20	RTSA+
21	DTRA+
22	DCDA+
23	RINGA+
24	CTSB+
25	
26	RXDB-
27	TXCB+
28	RXCB+
29	DSRB+
30	SRDB-
31	DCDB+
32	RINGB+
33	DTRB+
34	INT7B-
35	
36	RTSB+
37	INT4B-
38	STDB-
39	TXDB-
40	INT1B-
41	KBD1-
42	SIOACK-
43	SIORQ+
44	RDHIPEA-
45	
46	KBDO-
47	RDLOPEA-
48	MW-
49	DISKACK-
50	DISKRQ+
51	ODDP+
52	ENBL-
53	IOBCS+
54	IOR-
55	
56	DCIRDY+
57	FASTIO+
58	DMACTCS-
59	DMACS-
60	IOB7+
61	IOB6+
62	IOB5+
63	IOB4+
64	IOB3+
65	
66	IOB2+
67	IOB1+
68	IOB0+
69	DREQ-

70	IOCS-
71	NMI+
72	PICCS-
73	MEMCLK+
74	LPTACK-
75	
76	IOW-
77	EOP-
78	STROBE-
79	PERINT+
80	MWINH+
81	TO+
82	MBIO-
83	RESET-
84	PAREN+
85	
86	ACK-
88	CNTRL-
89	SELF-
90	SELW-
91	WRSTB-
92	INRQ-
93	RDSTB-
94	BUS0+
95	
96	BUS2+
97	BUS4+
98	BUS6+
99	BUS1+
100	BUS3+
101	BUS5+
102	BUS7+
103	BUSE+
104	BUSC+
105	
106	BUSA+
107	BUS8+
108	BUSF+
109	BUSD+
110	BUSB+
111	BUS9+
112	LPT1+
113	LPT3+
114	LPT5+
115	
116	LPT7+
117	LPT0+
118	LPT2+
119	LPT6+
120	LPT4+

Jumpers

<u>Name</u>	<u>Function</u>
D-E	64K memory chips
E-F	16K memory chips
G-I	8251A Tx-buffer-empty inter- rupt enabled
G-H	8251A Tx-buffer-empty inter- rupt disabled
JP1 (DIP 1-8)	Channel A is RS-422
JP3 (DIP 9-16)	" " "
JP2 (DIP 1-8)	Channel A is RS-232
JP4 (DIP 9-16)	" " "

Switches

<u>Switch</u>	<u>1 bit #</u>	<u>Option when closed</u>
1	Channel A	external transmit clock
2	Channel A	external receive clock
3	Channel A	internal receive clock
4	Channel A	internal transmit clock
5	Channel B	external transmit clock
6	Channel B	external receive clock
7	Channel B	internal transmit clock
8	Channel B	internal receive clock

Programmable LED and Audio Outputs (I/O Port 44)

<u>Bit</u>	<u>Function</u>
0	When 0 lights LED # CR6
1	When 0 lights LED # CR5
2	When 0 lights LED # CR4
3	When 0 lights LED # CR3
4	When 0 lights LED # CR2
5	When 0 lights LED # CR1
<u>6 & 7</u>	Bits 6 and 7 are used to- gether to operate the speaker:
0 0	No tone
1 0	1.24 Kiloherztz tone
0 1	Speaker-in
1 1	Speaker-out

Alternating between speaker-in and speaker-out at the proper rate can be used to generate any frequency.

Test Points

<u>Point</u>	<u>Meaning</u>
1	-5 volt supply
2	Comm channel A clock
3	Comm channel B clock
4	programmable timer clock
5	1.23 Megahertz clock
6	Realtime clock input
7	Power fail timer (High when power file detected)
8	Comm channel A transmit clock
9	Comm channel A transmit clock timeout (low when timed out)
10	SIO internal clock

I/O-Port Addresses vs. Functions

<u>Port</u>	<u>Logic</u>	<u>Write Register</u>	<u>Read Register</u>
28	Timer	Programmable timer	Programmable timer
2A	Timer	Comm channel B	Comm channel B
		baud rate generator	baud rate generator
2C	Timer	Comm channel A	Comm channel A
		baud rate generator	baud rate generator
2E	Timer	Mode	--
30	Comm	--	Interrupt Acknowledge Register
38	Keyboard	Write Data	Read Data
3A	Keyboard	Command	Status
44	LEDs, Audio	Bit 0 to 5 Bits 6 and 7, speaker	Switches on I/O Memory board
48	Comm	Channel A data	Channel A data
4A	Comm	Channel A command	Channel A

4C	Comm	Channel B data	status Channel B data
4E	Comm	Channel B command	Channel B Command
50	Printer	Data	Status
52	Parity	--	Low address bits
54	Parity	--	High address bits, status
56	IOCR	Control	Control, status
58	Video	Global Attribute Register	Global Attri- bute Register
60	Comm	Extended Control Register	Extended Control Register
68	DMA	Extended Address Register	Extended Address Register
70	Floppy	Data/Commands	Data/Status
72	Floppy	Control	Status
78	Hard disk	Data/Commands	Data/Status
7A	Hard disk	Control	Status

Device Pin Functions

The following several pages describe the pin functions of the complex ICs on the I/O-memory board.

Device Pin Functions: 8251A USART

The 8251A is the enhanced version of the industry standard 8251 Universal Synchronous/Asynchronous Receiver/Transmitter (USART). The 8251A is used as a peripheral device and is programmed by the CPU to operate using virtually any serial data transmission technique presently in use (including IBM "bi-sync"). The USART accepts data characters from the CPU in parallel format and then converts them into a continuous serial data stream for transmission. Simultaneously, it can receive serial data streams and convert them into parallel data characters for the CPU. The USART will signal the CPU whenever it can accept a new character for transmission or whenever it has received a character for the CPU. The CPU can read the complete status of the USART at any time. These include data transmission errors and control signals such as SYNDET, TxEMPTY. The chip is constructed using N-channel silicon gate technology.

<u>Signal Name</u>	<u>Pin Name</u>	<u>I/O</u>	<u>Function</u>
--	VCC		+5V supply.
--	GND		Ground.
IOB0 ... IOB7	D ₀ ... D ₇	I/O	A 3-state, bidirectional, 8-bit buffer interfaces the 8251A to the system Data Bus. Data is transmitted or received by the buffer upon execution of INput or OUTput instructions of the CPU. Control words, Command words and Status information are also transferred through the Data Bus Buffer. The command status and data in, and data out are separate 8-bit registers to provide double buffering. The Data Bus Buffer accepts inputs from the system Control bus and generates control signals for overall device operation. It contains the Control Word Register and Command Word Register that store the various control formats for the device functional definition.

1.23MHZ+	CLK	I	CLOCK: the CLK input is used to generate internal device timing. No external inputs or outputs are referenced to CLK but the frequency of CLK must be greater than 30 times the Receiver or Transmitter data bit rates.
--	RESET	I	A HIGH on this input forces the 8251A into an IDLE mode. The device will remain at IDLE until a new set of control words is written into the 8251A to program its functional definition. Minimum RESET pulse width is 6 t _{CY} (clock must be running).
BIOW-	WR/	I	WRITE: A LOW on this input informs the 8251A that the CPU is writing data or control words to the 8251A.
IOR-	RD/	I	READ: A LOW on this input informs the 8251A that the CPU is reading data or status information from the 8251A.
LA1+	C	I	CONTROL/DATA: This input, in conjunction with the WR/ and RD/ inputs, informs the 8251A that the word on the Data Bus is either a data character, control word or status information. 1 = CONTROL/STATUS 0 = DATA
	CS/	I	CHIP SELECT: A LOW on this input selects the 8251A. No reading or writing will occur unless the device is selected. When CS/ is high, the Data Bus in the float state and RD/ and WR/ will have no effect on the chip.
GND	DSR/	I	DATA SET READY: The DSR/ input signal is a general purpose, 1-bit inverting input port. Its condition can be tested by the CPU using a Status Read operation. The DSR/ input is normally used to test Modem conditions such as Data Set Ready.
--	DTR/	O	DATA TERMINAL READY: The DTR/ output signal is a general purpose, 1-bit inverting output port. It can be set "low" by programming the appropriate bit in the Command Instruction word. The DTR/ output signal is normally used for Modem control such as Data Terminal Ready or Rate Select.

--	RTS/	O	REQUEST TO SEND: The RTS/ output signal is a general purpose, 1-bit inverting output port. It can be set LOW by programming the appropriate bit in the Command Instruction word. The RTS/ output signal is normally used for Modem control such as Request to Send.
GND	CTS/	I	CLEAR TO SEND: A LOW on this input enables the 8251A to transmit serial data if the Tx Enable bit in the Command byte is set to a "one." If either a Tx Enable off or CTS off condition occurs while the Tx is in operation, the Tx will transmit all the data in the USART, written prior to Tx Disable command before shutting down. On the 8251A/ S2657/ if CTS off or Tx Enable off condition occurs before the last character written appears in the serial bit stream, that character will be transmitted again upon CTS on or Tx Enable on condition.
--	TxRDY	O	TRANSMITTER READY: This output signals the CPU that the transmitter is ready to accept a data character. The TxRDY output pin can be used as an interrupt to the system, since it is masked by Tx Disabled, or, for Polled operation, the CPU can check TxRDY using a Status Read operation. TxRDY is automatically reset by the leading edge of WR/ when a data character is loaded from the CPU.
Note that when using the Polled operation, the TxRDY status bit is not masked by Tx Enabled, but will only indicate the Empty/Full Status of the Tx Data Input Register.			
--	TxD	O	TRANSMITTER DATA: The Transmitter Buffer accepts parallel data from the Data Bus Buffer, converts it to a serial bit stream, inserts the appropriate characters or bits (based on the communication technique) and outputs a composite serial stream of data on the TxD output pin on the falling edge of TxC/. The transmitter will begin transmission upon being enabled

if CTS/ = 0. The TxD line will be held in the marking state immediately upon a master Reset or when neither Tx Enable nor CTS/ is active or when the transmitter is empty.

-- TxEMPTY 0

When the 8251A has no characters to transmit, the TxEMPTY output will go "high." It resets automatically upon receiving a character from the CPU if the transmitter is enabled. TxEMPTY can be used to indicate the end of a transmission mode, so that the CPU "knows" when to "turn the line around" in the half-duplexed operational mode.

In SYNChronous mode, a "high" on this output indicates that a character has not been loaded and the SYNC character or characters are about to be or are being transmitted automatically as "fillers." TxEMPTY does not go low when the SYNC characters are being shifted out.

19.53KHZ+ TxC/ I

TRANSMITTER CLOCK controls the rate at which the character is to be transmitted. In the Synchronous transmission mode, the Baud Rate (1x) is equal to the TxC/ frequency. In Asynchronous transmission mode the baudrate is a fraction of the actual TxC frequency. A portion of the mode instruction selects this factor; it can be 1, 1/16 or 1/64 the TxC/. The falling edge of TxC/ shifts the serial data out of the 8251A.

19.53KHZ+ RxC/ I

RECEIVER CLOCK controls the rate at which the character is to be received. In Synchronous Mode, the Baud Rate (1x) is equal to the actual frequency of RxC/. In Asynchronous Mode, the Baud Rate is a fraction of the actual RxC/ frequency. A portion of the mode instruction selects this factor; 1, 1/16 or 1/64 the RxC/. Data is sampled into the 8251A on the rising edge of RxC/.

NOTE: In most communications systems, the 8251A will be handling both the transmission and reception operations of a single link. Consequently, the

Receive and Transmit Baud Rates will be the same. Both TxC/ and RxC/ will require identical frequencies for this operation and can be tied together and connected to a single frequency source (Baud Rate Generator) to simplify the interface.

-- RxRDY 0 RECEIVER READY: This output indicates that the 8251A contains a character that is ready to be input to the CPU. RxRDY can be connected to the interrupt structure of the CPU or, for Polled operation, the CPU can check the condition of RxRDY using a Status Read operation.

Rx Enable off both masks and holds RxRDY in the Reset Condition. For Asynchronous mode, to set RxRDY, the Receiver must be Enabled to sense a Start Bit and complete character must be assembled and transferred to the Data Output Register. For Synchronous mode, to set RxRDY, the Receiver must be enabled and a character must finish assembly and be transferred to the Data Output Register.

Failure to read the received character from the Rx Data Output Register prior to the assembly of the next Rx Data character will set overrun condition error and the previous character will be written over and lost. If the Rx Data is being read by the CPU when the internal transfer is occurring, overrun error will be set and the old character will be lost.

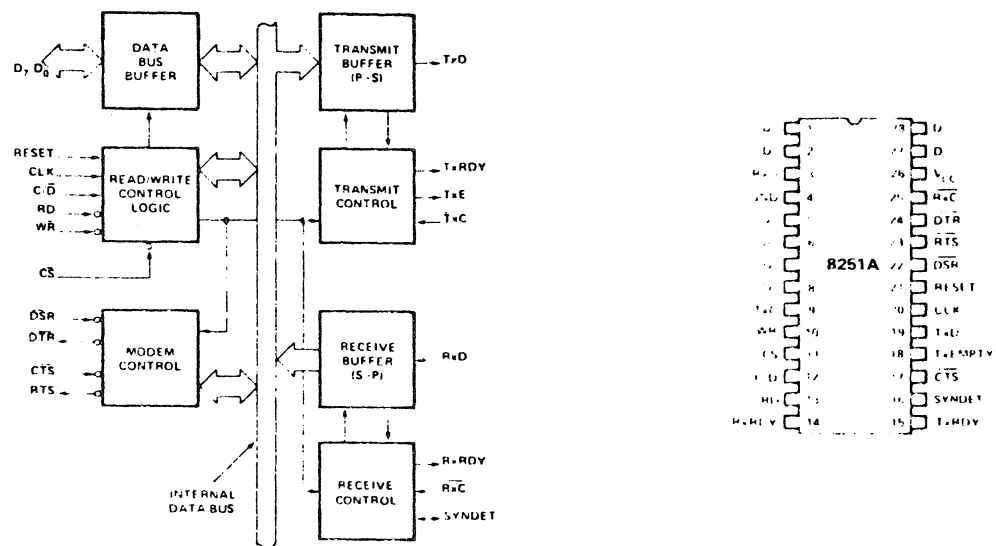
-- RxD I RECEIVE DATA: The Receiver accepts serial data, converts this serial input to parallel format, checks for bits or characters that are unique to the communication technique and sends an assembled character to the CPU. Serial data is input to RxD pin, and is clocked in on the rising edge of RxC/.

-- SYNDET/ I/O BD SYNC DETECT/BREAK DETECT: This pin is used as either input or output, programmable through the Control Word. It is reset to output mode low upon

RESET. When used as an output (internal Sync model), the SYNDET pin will go HIGH to indicate that the 8251A has located the SYNC character in the Receive mode. If the 8251A is programmed to use double SYNC characters (bisync), then SYNDET will go HIGH in the middle of the last bit of the second SYNC character. SYNDET is automatically reset upon a Status Read operation. When used as an input (external SYNC detect mode), a positive going signal will cause the 8251A to start assembling data characters on the rising edge of the next RxC/. Once in SYNC, the HIGH input signal can be removed. When External SYNC Detect is programmed, the Internal SYNC Detect is disabled.

In the asynchronous mode, this pin functions as follows:

This output will go high whenever the receiver remains low through two consecutive stop bit sequences (including the start bits, data bits, and parity bits). Break Detect may also be read as a Status bit. It is reset only upon a master chip Reset or Rx Data returning to a "one" state.



Device Pin Functions: 8253 PROGRAMMABLE TIMER

The 8253 programmable interval timer/counter functions as a general purpose, multi-timing element that can be treated as an array of I/O ports in the system software. It is organized as 3 independent 16-bit counters, each with a count rate of up to 2 MHz. All modes of operation are software programmable.

The 8253 solves one of the most common problems in any microcomputer system, the generation of accurate time delays under software control. Instead of setting up timing loops in systems software, the programmer configures the 8253 to match his requirements, initializes one of the counters of the 8253 with the desired quantity, then upon command the 8253 will count out the delay and interrupt the CPU when it has completed its tasks. It is easy to see that the software overhead is minimal and that multiple delays can easily be maintained by assignment of priority levels.

Other counter/timer functions that are non-delay in nature but also common to most microcomputers can be implemented with the 8253.

- o Programmable Rate Generator
- o Event Counter
- o Binary Rate Multiplier
- o Realtime Clock
- o Digital One-Shot
- o Complex Motor Controller

<u>Signal Name</u>	<u>Pin Name</u>	<u>I/O</u>	<u>Function</u>
--	VCC		+5V supply.
--	GND		Ground.
IOB0 ... IOB7	D0 ... D7	I/O	DATA BUS: A 3-state, bi-directional, 8-bit buffer interfaces the 8253 to the system data bus. Data is transmitted or received by the buffer upon execution of INput or OUTput CPU instructions. The Data Bus Buffer has three basic functions. 1. Programming the MODES of the 8253. 2. Loading the count registers. 3. Reading the count values.
LA1+	A0	I	A0, A1: These inputs are normally

LA2+	A1		connected to the address bus. Their function is to select one of the three counters to be operated on and to address the control word register for mode selection.
TMR1CS-	CS/	I	CHIP SELECT enables the 8253. No reading or writing will occur unless the device is selected. The CS/ input has no effect upon the actual operation of the counters.
IOR-	RD/	I	READ informs the 8253 that the CPU is inputting data in the form of a counters value.
BIOW-	WR/	I	WRITE informs the 8253 that the CPU is outputting data in the form of mode information or loading counters.
19.53KHZ+	CLK0	I	COUNTER CLOCK INPUTS provide the time base for the three independent interval counters.
1.23MHZ+	CLK1		
1.23MHZ+	CLK2		
--	GATE0	I	COUNTER GATE INPUTS can be used to enable, trigger, or disable counters.
--	GATE1		
--	GATE2		
TMRINT+	OUT0	O	COUNTER OUTPUTS: In mode 0, these lines go HIGH when the programmed interval has elapsed. In mode 3, they toggle at the rate which has been programmed.
C1CLK+	OUT1		
C2CLK+	OUT2		

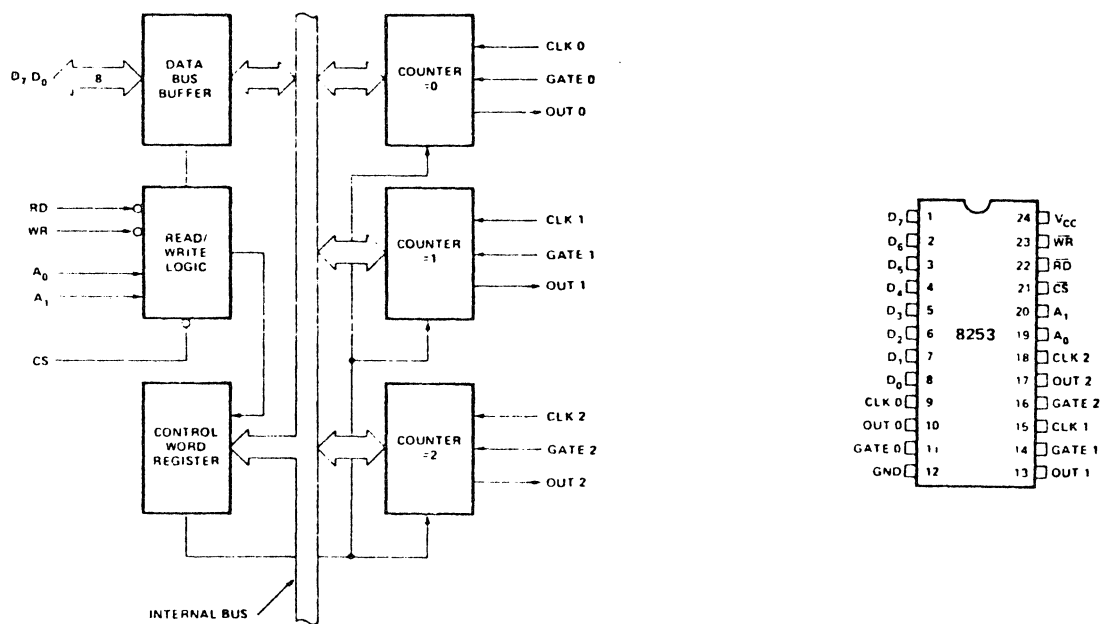


Figure 5-5. 8253 Programmable Timer

Device Pin Functions: Z80A SIO/2 COMMUNICATIONS CONTROLLER

The Z-80 SIO Serial Input/Output Controller is a dual-channel data communication interface. Its basic functions as a serial-to-parallel, parallel-to-serial converter/controller can be programmed by a CPU for a broad range of serial communication applications. The device supports all common asynchronous and synchronous protocols, byte- or bit-oriented, and performs all of the functions traditionally done by UARTs, USARTs and synchronous communication controllers combined, plus additional functions traditionally performed by the CPU. Some of its features are as follows:

- o Two independent full-duplex channels, with separate control and status lines for modems or other devices.
- o Data rates of 0 to 500K bits/second in the x1 clock mode with a 2.5 MHz clock (Z-80 SIO), or 0 to 800K bits/second with a 4.0 MHz clock (Z-80A SIO).
- o Everything necessary for complete messages in 5, 6, 7 or 8 bits/character. Includes variable stop bits and several clock-rate multipliers; break generation and detection; parity; overrun and framing error detection.
- o Synchronous protocols: everything necessary for complete bit- or byte-oriented messages in 5, 6, 7 or 8 bits/character, including IBM Bisync, SDLC, HDLC, CCITT-X.25 and others. Automatic CRC generation/checking, sync character and zero insertion/deletion, abort generation/detection and flag insertion.
- o Receiver data registers quadruply buffered, transmitter registers doubly buffered.

<u>Signal Name</u>	<u>Pin Name</u>	<u>I/O</u>	<u>Function</u>
--	+5V		+5 Volt Supply.

--	GND		GROUND.
--	B	I	CHANNEL A OR B SELECT (input, High selects Channel B). This input defines which channel is accessed during a data transfer between the CPU and the SIO. Address bit A ₀ from the CPU is often used for the selection function.
--	C	I	CONTROL OR DATA SELECT (input, High selects Control). This input defines the type of information transfer performed between the CPU and the SIO. A High at this input during a CPU write to the SIO causes the information on the data bus to be interpreted as a command for the channel selected by B. A Low at C means that the information on the data bus is data. Address bit A ₁ is often used for this function.
--	CE/	I	CHIP ENABLE (Input, active Low). A Low level at this input enables the SIO to accept command or data input from the CPU during a write cycle, or to transmit data to the CPU during a ready cycle.
3.33MHZ+	CLK	I	SYSTEM CLOCK (input). The SIO uses the standard Z-80 System Clock to synchronize internal signals. This is a single-phase clock.
CTSA- CTSB-	CTSA/ CTSB/	I	CLEAR TO SEND (inputs, active Low). When programmed as Auto Enables, a Low on these inputs enables the respective transmitter. If not programmed as Auto Enables, these inputs may be programmed as general-purpose inputs. both inputs are Schmitt-trigger buffered to accommodate slow-risetime signals. The SIO detects pulses on these inputs and interrupts the CPU on both logic level transitions. The Schmitt-trigger buffering does not guarantee a specified noise-level margin.
IOB0 ... IOB7	D ₀ ... D ₇	I/O	SYSTEM DATA BUS (bidirectional, 3-state). The system data bus transfers data and commands between the CPU and

the Z-80 SIO. D_0 is the least significant bit.

DCDA- DCDB-	DCDA/ DCDB/	I	DATA CARRIER DETECT (inputs, active Low). These pins function as receiver enables if the SIO is programmed for Auto Enables; otherwise they may be used as general-purpose input pins. Both pins are Schmitt-trigger buffered to accommodate slow-risetime signals. The SIO detects pulses on these pins and interrupts the CPU on both logic level transitions. Schmitt-trigger buffering does not guarantee a specific noise-level margin.
DTRA- DTRB-	DTRA/ DTRB/	O	DATA TERMINAL READY (outputs, active Low). These outputs follow the state programmed into Z-80 SIO. They can also be programmed as general-purpose outputs.
--	IEI	I	INTERRUPTS ENABLE IN (input, active High). This signal is used with IEO to form a priority daisy chain when there is more than one interrupt-driven device. A High on this line indicates that no other device of higher priority is being serviced by a CPU interrupt service routine.
--	IEO	O	INTERRUPT ENABLE OUT (output, active High). IEO is High only if IEI is High and the CPU is not servicing an interrupt from this SIO. Thus, this signal blocks lower priority devices from interrupting while a higher priority device is being serviced by its CPU interrupt service routine.
--	INT/	O	INTERRUPT REQUEST (output, open drain, active Low). When the SIO is requesting an interrupt, it pulls INT/ Low.
--	IORQ/	I	INPUT/OUTPUT REQUEST (output, open drain, active Low). IORQ/ is used in conjunction with B, C, CE/ and RD/ to transfer commands and data between the CPU and the SIO. When CE/, and RD/ are all active, the channel selected by B/A transfers data to the CPU (a read operation). When Ce/ and IORQ/

are active, but RD/ is inactive, the channel selected by B is written to by the CPU with either data or control information as specified by C. As mentioned previously, if IORQ/ and M1/ are active simultaneously, the CPU is acknowledging an interrupt and the SIO automatically places its interrupt vector on the CPU data bus if it is the highest priority device requesting an interrupt.

--	M1/	I	MACHINE CYCLE (input from Z-80 CPU, active Low). M1/ is active and RD/ is also active, the Z-80 CPU is fetching an instruction from memory; when M1/ is active while IORQ/ is active, the SIO accepts R1/ and IORQ/ as an interrupt acknowledge if the SIO is the highest priority device that has interrupted the Z-80 CPU.
RXCA- RXCD-	RxCA/ RxCB/	I	RECEIVER CLOCKS (inputs). Receive data is sampled on the rising edge of RxC/. The Receive Clocks may be 1, 16, 32 or 64 times the data rate in asynchronous modes. These clocks may be driven by the Z-80 CTC Counter Timer Circuit for programmable baud rate generation. Both inputs are Schmitt-trigger buffered (no noise level margin is specified).
--	RD/	I	HEAD CYCLE STATUS (input from CPU, active Low). If RD/ is active, a memory or I/O read operation is in progress. RD/ is used with B, CE/ and IORQ/ to transfer data from the SIO to the CPU.
RXDA+ RXDB+	RxDA RxDB	I I	RECEIVE DATA (inputs, active High). Serial data at TTL levels.
	RESET/	I	Reset (input, active Low). A Low RESET/ disables both receivers and transmitters, forces TxDA and TxDB marking, forces the modem controls High and disables all interrupts. The control registers must be rewritten after the SIO is reset and before data is transmitted or received.
RTSA-	RTSA/	O	REQUEST TO SEND (outputs, active

RTSB-	RTSB/		Low). When the RTS bit in Write Register 5 (Figure 14) is set, the RTS/ output goes Low. When the RTS bit is reset in the Asynchronous mode, the output goes High after the transmitter is empty. In Synchronous modes, the RTS/ pin strictly follows the state of the RTS bit. Both pins can be used as general-purpose outputs.
-------	-------	--	---

SYNCA- SYNCB-	SYNCA/ SYNCB/	I/O	SYNCHRONIZATION (inputs/outputs, active Low). These pins can act either as inputs or outputs. In the asynchronous receive mode, they are inputs similar to CTS/ and DCD/. In this mode, the transitions on these lines affect the state of the Sync/Hunt status bits in Read Register 0 (Figure 13), but have no other function. In the External Sync mode, these lines also act as inputs. When external synchronization is achieved, SYNC/ must be driven Low on the second rising edge of RxC/ after that rising edge of RxC/ on which the last bit of the sync character was received. In other words, after the sync pattern is detected, the external logic must wait for two full Receive Clock cycles to activate the SYNC/ input. Once SYNC/ is forced Low, it should be kept Low until the CPU informs the external synchronization detect logic that synchronization has been lost or a new message is about to start. Character assembly begins on the rising edge of RxC/ that immediately precedes the falling edge of SYNC/ in the External Sync mode.
------------------	------------------	-----	---

In the internal synchronization mode (Monosync and Bisync), these pins act as outputs that are active during the part of the receive clock cycle in which sync characters are recognized. The sync condition is not latched, so these outputs are active each time a sync pattern is recognized, regardless of character boundaries.

In the Z-80 SIO/2 bonding option, SYNCB/ is omitted.

TXCA- TXCD-	TxCA/ TxCB/	I	TRANSMITTER CLOCKS (inputs). TxD changes from the falling edge of TxC. In asynchronous modes, the Transmitter Clocks may be 1, 16, 32 or 64 times the data rate; however, the clock multiplier for the transmitter and the receiver must be the same. The Transmit Clock inputs are Schmitt-trigger buffered for relaxed rise- and fall-time requirements (no noise level margin is specified). Transmitter Clocks may be driven by the Z-80 CTC Counter Timer Circuit for programmable baud rate generation.
			In the Z-80 SIO/0 bonding option. TxCB/ is bonded together with TxCB/.
TXDA+ TXDB+	TxDA TxDB	O	TRANSMIT DATA (outputs, active High). Serial data at TTL levels.
-- --	WRDYA/ WRDYB/	O	WAIT/READY A, WAIT/READY B (outputs, open drain, when programmed for Wait function; driven High and Low when programmed for Ready function). These dual-purpose outputs may be programmed as Ready lines for a DMA controller or as Wait lines that synchronize the CPU to the SIO data rate. The reset state is open drain.

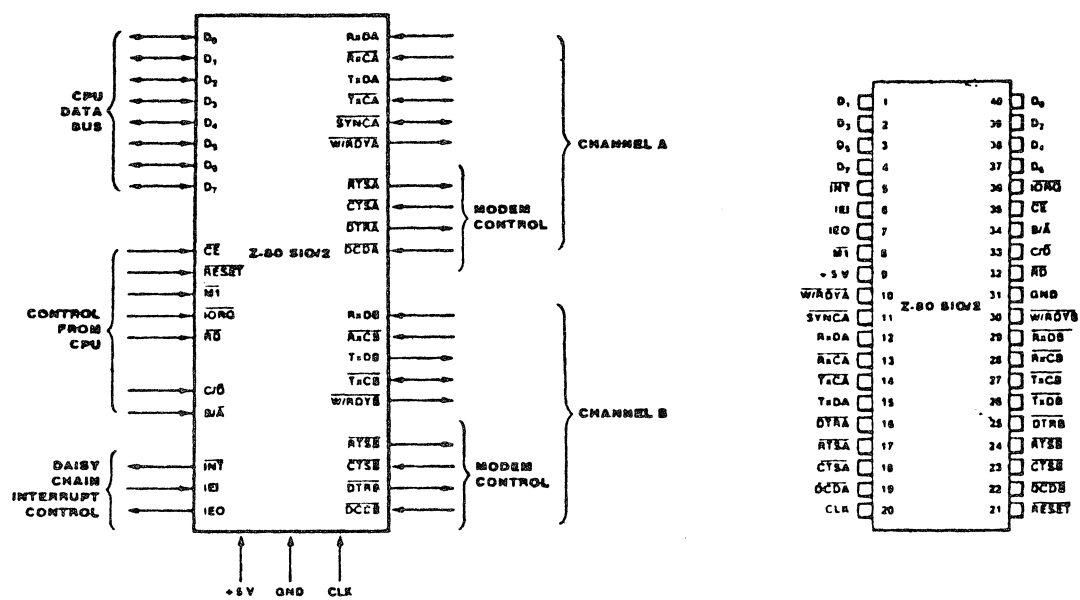
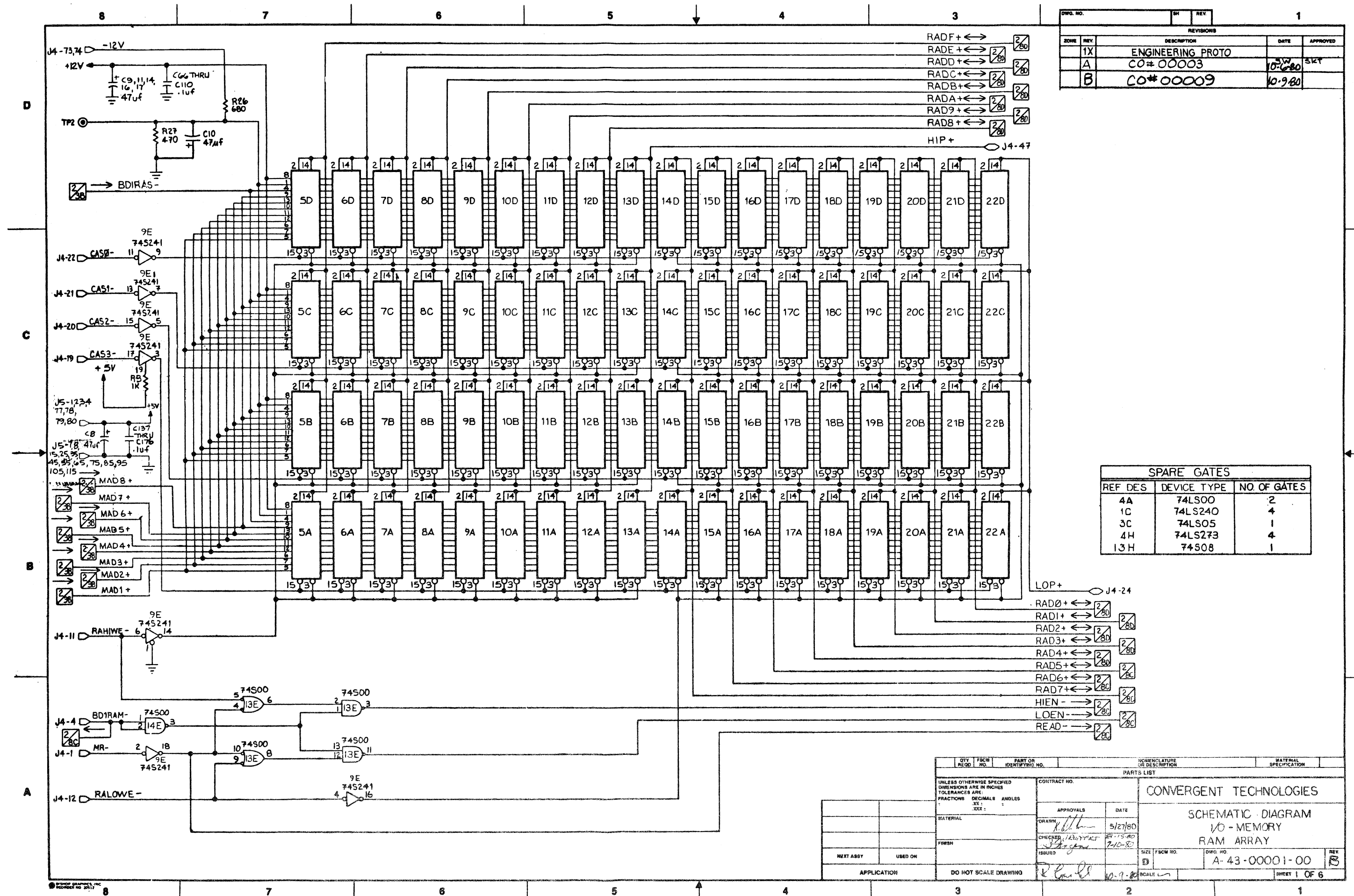
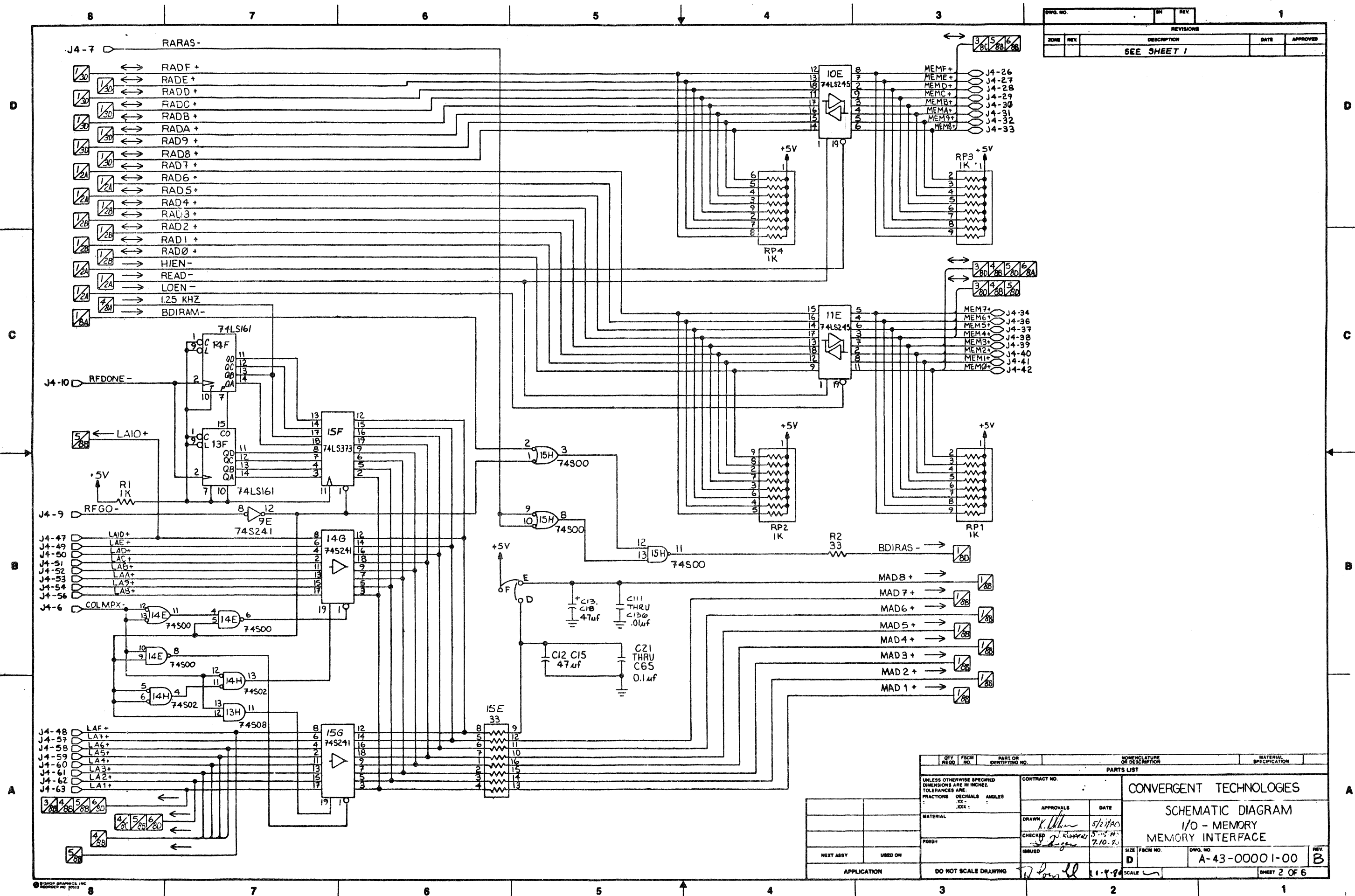
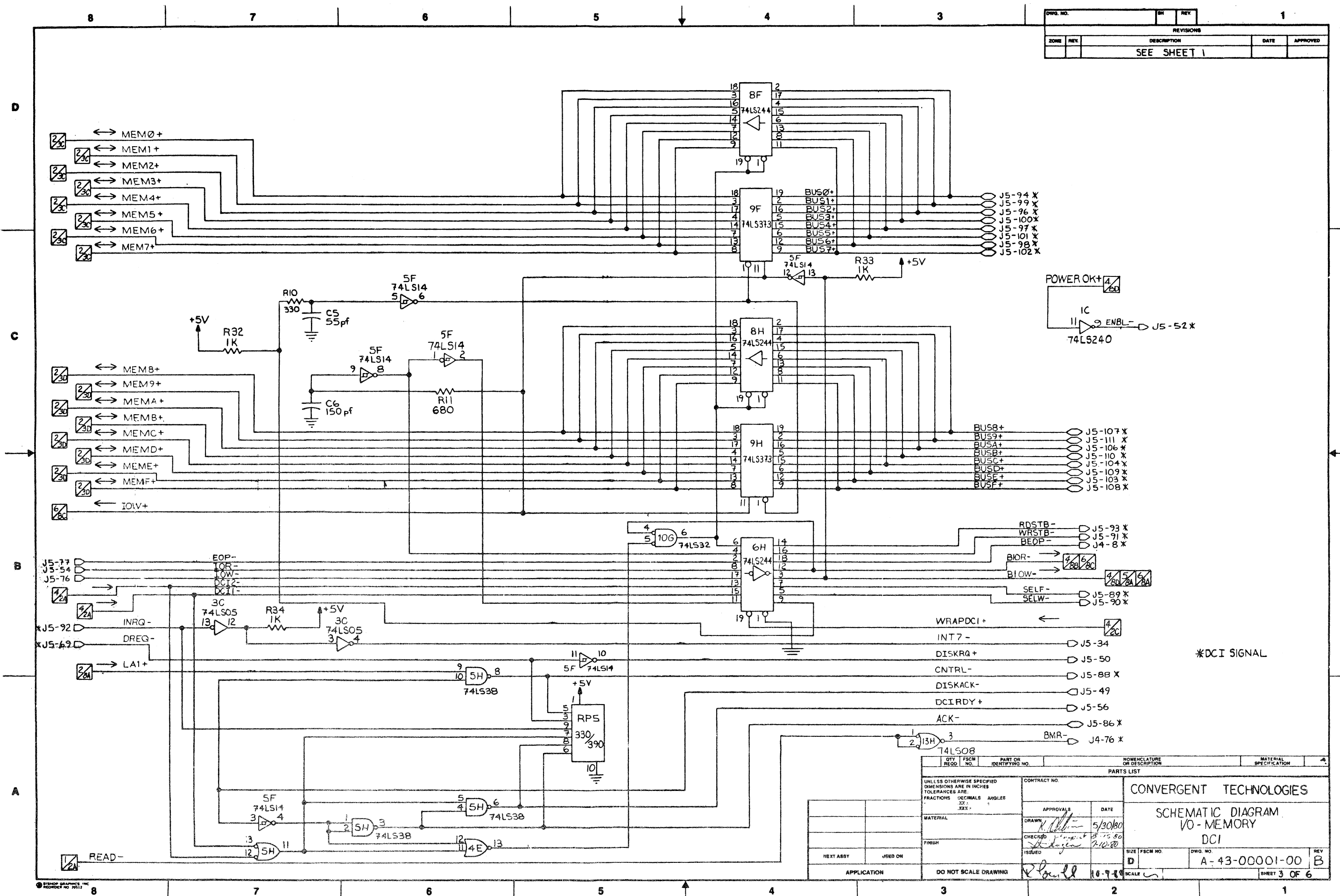
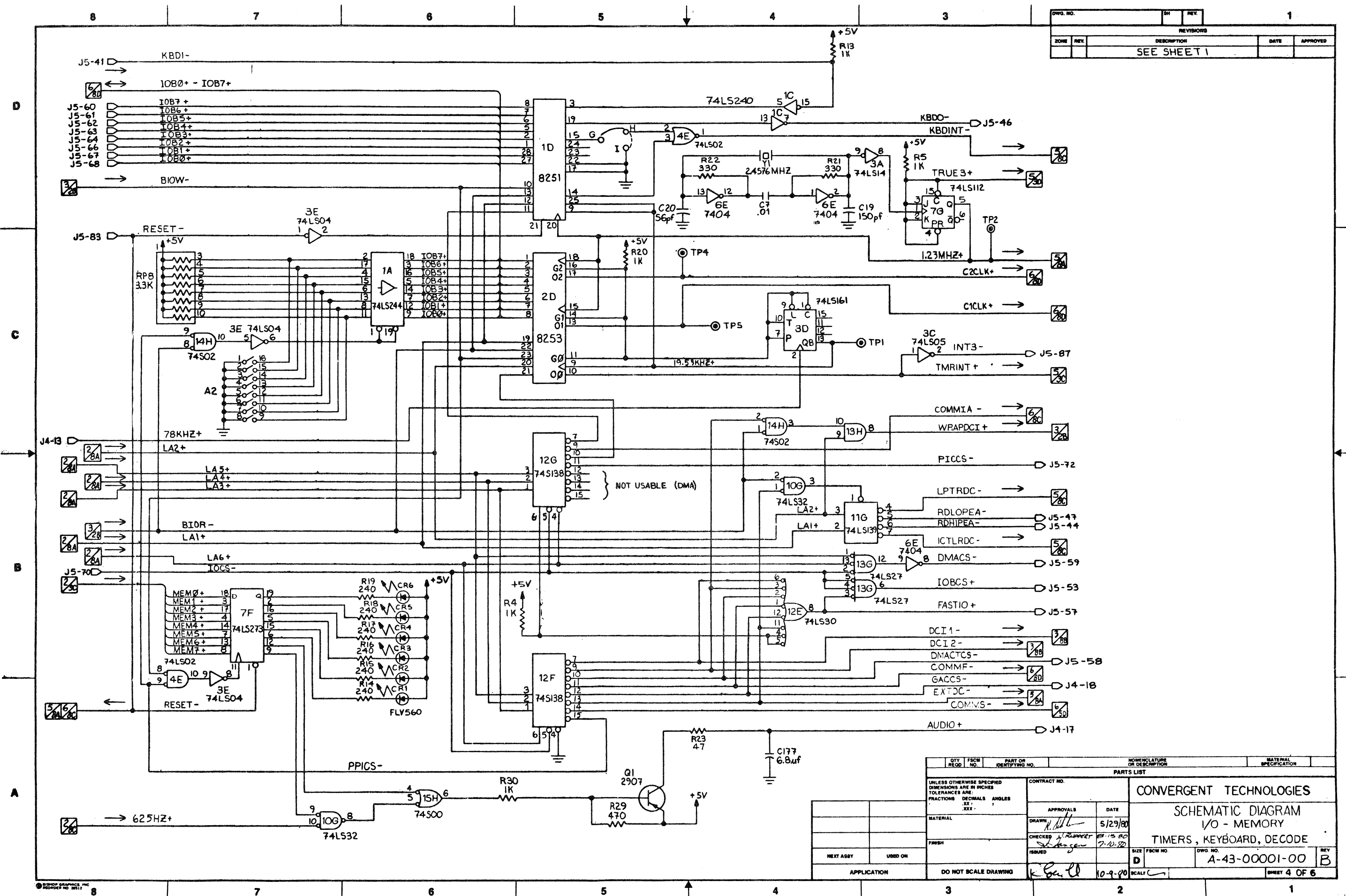


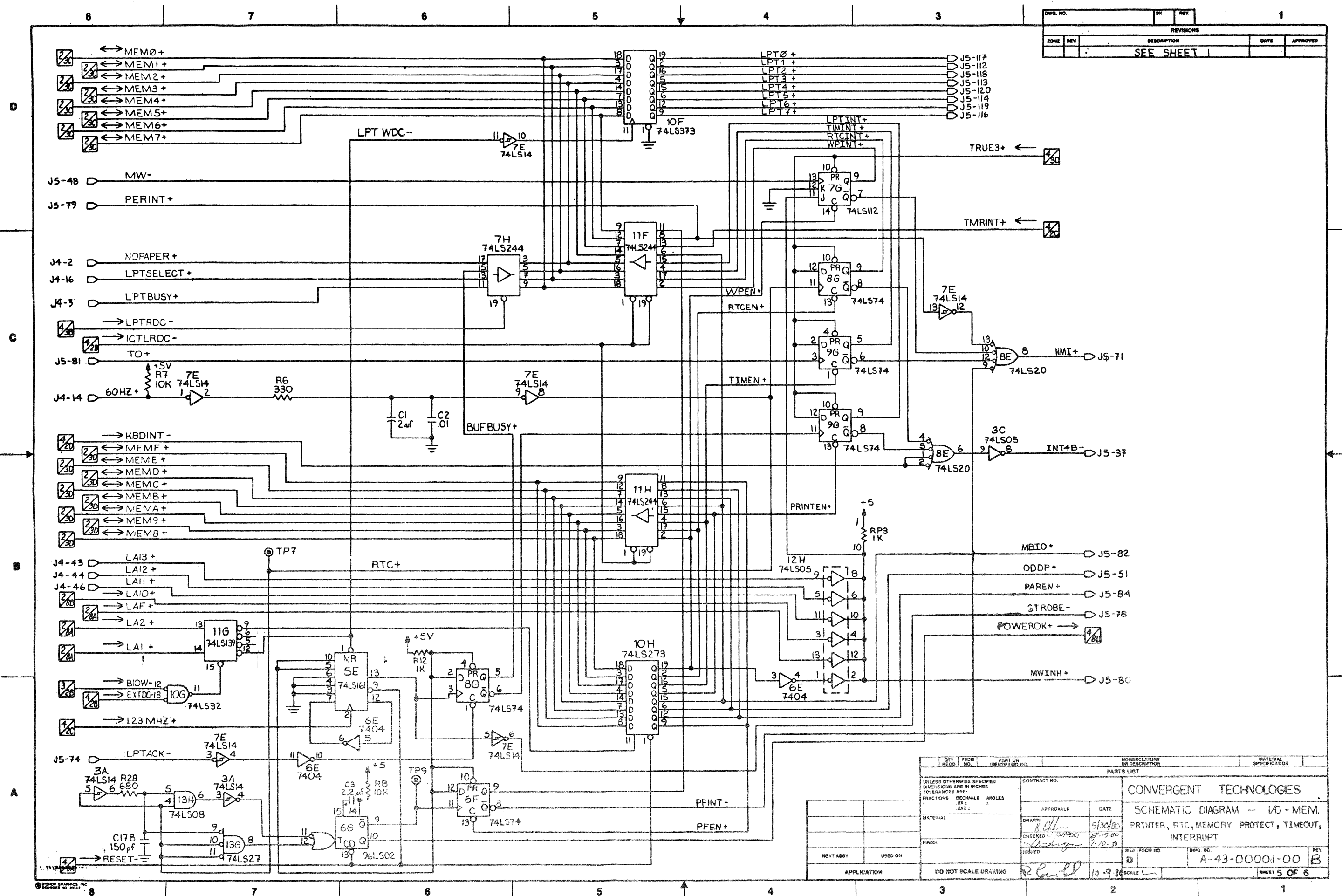
Figure 5-6. Z80A SIO/2

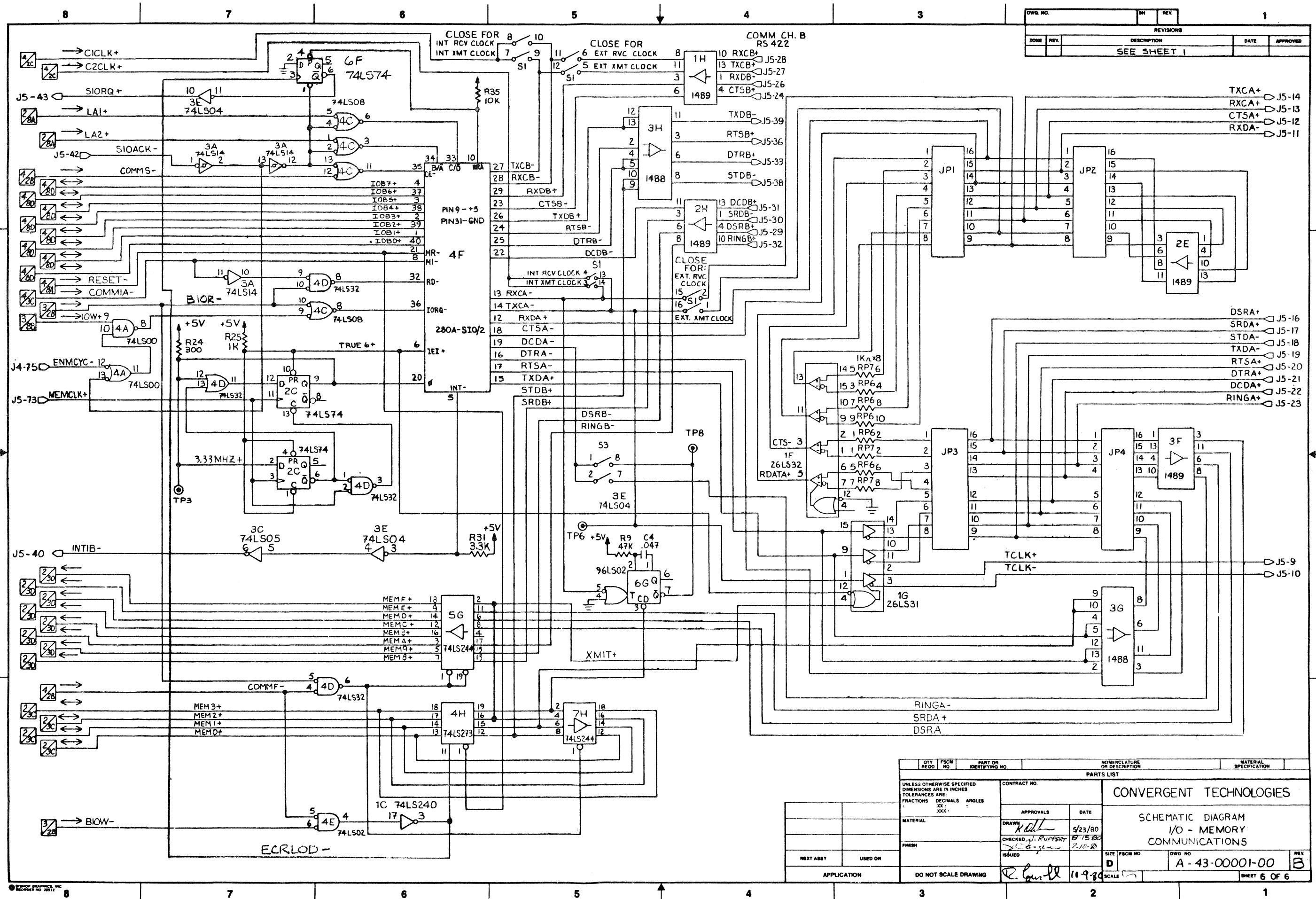












6. RAM EXPANSION BOARD

OVERVIEW

The RAM Expansion board contains the second set of banks of RAM of a CT system. The board may contain 1, 2, 3, or 4 banks of 18 single-bit RAM ICs. For 16K bit devices this is equivalent to 32K, 64K, 96K and 128K respectively. For 64K bit devices this is equivalent to 128K, 256K, 384K, and 512K respectively.

OPERATION BY LOGIC BLOCK

The RAM Array

The 7 address lines of the chips are multiplexed so that 14 address bits can be represented with 7 pins. The chip is organized as a square matrix of 128 rows x 128 columns. Each row/column combination gives the address of 1 bit, or cell of memory.

The RAS- (row address strobe) signal has two functions: it latches the row address on the 7 address pins and it accesses the specified row of 128 cells. The CAS- (column address strobe) signal also has two functions: it latches the column address, and performs the data operation, read or write, on the one cell specified by the column address of the 128 cells already accessed by RAS-. A write cycle is caused if the WE- pin is asserted at the leading edge of CAS-. The write data at the data input pin of the chip is latched by the leading edge of CAS-. A read cycle is caused if the WE- pin is inactive at the leading edge of CAS-. The read cycle causes the data from the selected cell to be driven on the chip's data output pin.

The memory chips used are dynamic RAMs. Data is stored in parasitic capacitance that needs to be re-charged every 2 milliseconds to prevent leakage from destroying the contents of the memory.

Each time RAS- is asserted the entire 128-cell row corresponding to the state of the address pins is accessed. Whenever the cell is accessed it is re-charged.

In order to insure that all cells of the chip are refreshed (re-charged) every 2 milliseconds there is special refresh circuitry on the processor and RAM Expansion boards. This circuitry works by issuing a RAS- every 12.8 microseconds and causing a counter to generate each different row address. No CAS- is issued during refresh, so no data is transferring.

64K RAMs work similarly to the 16K RAMs, except that there are sixteen address bits (one of the power supply pins of the 16K RAM being used as the extra address pin required for the 64K parts) and the array is 256 rows of 256 columns can be accessed a byte at a time, each byte has its own parity bit. So one word is represented as two 9 bit "bytes".

All 72 chips in the array have the same RAS-signal. Bank selection is done by asserting only one of the CAS4- - CAS7- signals. Therefore, each CAS- signal goes to 18 chips and the RAS-signal goes to 72 chips. The data-in and data-out pins of the chips and the RAS- signal goes to 72 chips. The data-in and data-out pins of the chips are tied together making a bidirectional data bus (RAD0+ - RADF+) (p2zD2). Each data bus bit goes to 4 chips, consisting of the corresponding bit of each word in all 4 banks. The parity bits HIP+ (high-byte parity) and LOP+ (low-byte parity) are connected in the same fashion. The seven address pins RAA0+ - RAA6+ (p2zp8) are connected in parallel to all 72 chips. The signal RA5A7+ is an address bit if 64K RAMs are used but is connected to +5 at jumper A-C for 16K chips. There are two write enable signals, RAHIWE- (high or odd byte write enable) and RALOWE- (low or even byte enable) which cause writing in either the low byte, the high byte, or both. Again, for writing, CAS-selects the particular bank written. Each write enable signal goes to 36 RAM chips.

The RAS- and CAS- signals for read, write and refresh are generated in the memory control of the processor board. When they come on to RAM Expansion board they are called RAS- (Row Address Strobe) (p2zA7) and CAS4- -CAS7- (p2zB8), corresponding to banks 4 through 7. At 10F-11 (p2zA6), RAS- is gated so that only refreshes and accesses to the RAM Expansion board (as opposed to the I/O Memory board) generate the signal BRAS-. BRAS- connects to the RAS- pins of the RAM IC's. RFG0- indicates a refresh is in progress and BD2RAM- (from the processor board) indicates the RAM expansion board memory is being addressed.

There are three sources for the address driven on the RAM chip address pins, the row address (6K (p1zB6)), the column address (5K (p1zC6)), and the refresh row (3K (p1zD6)). The selection of which address driver is enabled is made by the high speed decoder consisting of 10J-3, 6, 8, and 11, and 10H-1 and 4 (p1zC6). The RFG0- signal, when active enables 3K and disables 5K and 6K. The COLMPX- (column multiplex) signal causes the row driver (6K) to disable and the column driver (5K) to enable. COLMPX- and RFG0- come from the RAM controller on the processor board. RFDONE- from the processor board indicates that a refresh cycle has completed and causes the refresh address counters 1K and 2K (p1zD6) to increment.

Data to and from the RAM array is buffered from the MEM bus (MEM0+ - MEMF+) by the transceivers at 9F and 9H (p1zC3). The direction of the transceivers is established by the signal READ- which is a buffered MR- (memory read) signal. The enable for the high (odd) byte driver is HIEN- and for the low (even) byte is LOEN-. HIEN- is asserted when the RAM Expansion RAM is addressed (BD2RAM- asserted at 10E-9 (p2zA9)) and either a read is in progress (MR- active) or the high-order byte write signal is active (RAHIWE-). LOEN- is asserted when the RAM expansion memory is addressed and either a read is in progress or the low-order byte write enable is active (RALOWE-).

Buffering For ROM Expansion Board

When a RAM Expansion board and a ROM Expansion board are both used on a motherboard, the ROM Expansion board plugs into the P1 connector of the RAM Expansion board. The address lines (LA1+ - LA10+) from the motherboard connector, J1, are buffered by chips 9A and 10A (p1zB7) to become

BLA1+ - BLA10+ (buffered local address) and go to P1. The data lines from the ROM expansion board (BMEM0+ - BMEF+) are buffered by 9C and 10C (p1zB2) and driven onto the MEM bus (MEM0+ - MEMF+) at the J1 connector. Buffers 9C and 10C are enabled by the ROM expansion board decode signal from the processor board (PROMEXDC-).

AC CHARACTERISTICS

Refer to AC Characteristics in Chapter 4
(Processor Board)

DC CHARACTERISTICS

Refer to DC Characteristics in Chapter 4
(Processor Board)

REFERENCES

Edge-Connector Pin List

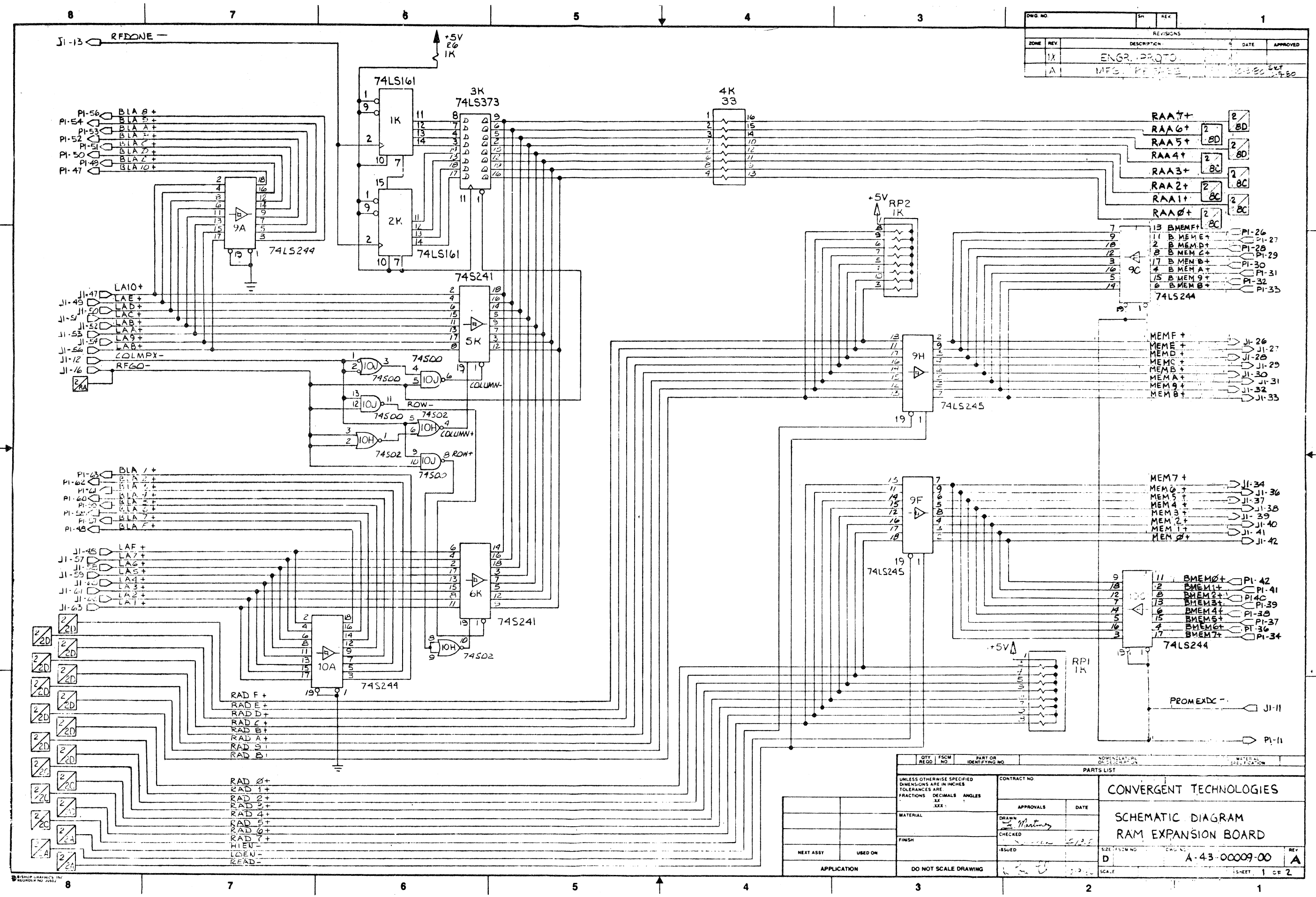
Connector J1

<u>Pin</u>	<u>Signal</u>
1	+5V
2	+5V
3	+5V
4	+5V
5	-12V
6	-12V
7	GND
8	
9	BD2RAM-
10	MR-
11	PROMEXDC-
12	COLMPX-
13	RFDONE-
14	RARAS-
15	
16	RFGO-
17	RALOWE-
18	RAHIWE-
19	CAS7-
20	CAS6-
21	CAS5-
22	CAS4-
23	HIP+
24	LOP+
25	
26	MEMF+
27	MEME+
28	MEMD+
29	MEMC+
30	MEMB+
31	MEMA+
32	MEM9+
33	MEM8+
34	MEM7+
35	
36	MEM6+
37	MEM5+
38	MEM4+
39	MEM3+
40	MEM2+
41	MEM1+
42	MEM0+
43	LA13+
44	LA12+
45	

<u>Pin</u>	<u>Signal</u>
46	LA11+
47	LA10+
48	LAF+
49	LAE+
50	LAD+
51	LAC+
52	LAB+
53	LAA+
54	LA9+
55	
56	LA8+
57	LA7+
58	LA6+
59	LA5+
60	LA4+
61	LA3+
62	LA2+
63	LA1+
64	LA0+
65	
66	
67	+12V
68	+12V
69	+12V
70	+12V
71	
72	
73	-12V
74	-12V
75	
76	
77	+5V
78	+5V
79	+5V
80	+5V

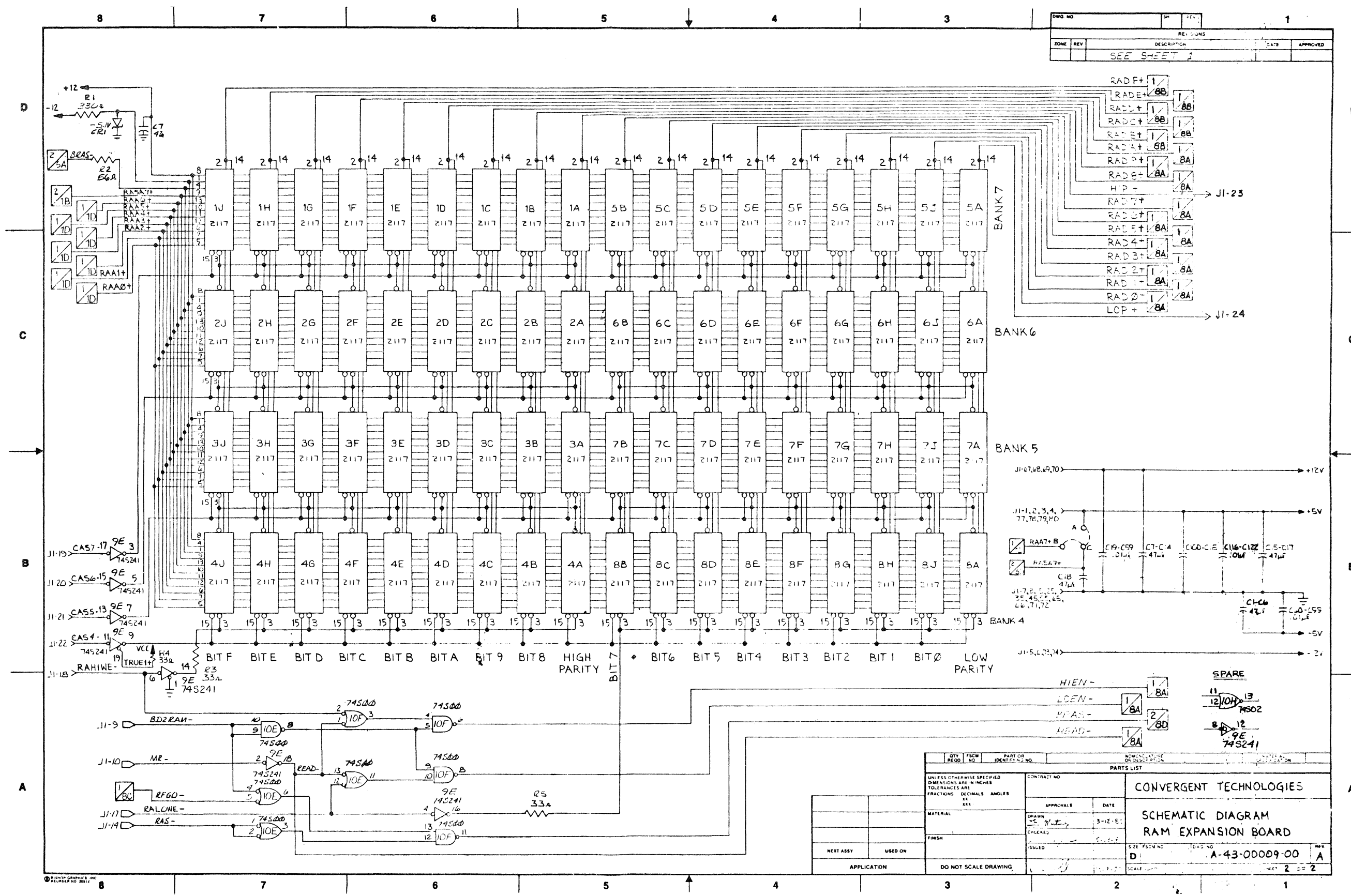
Jumpers

<u>Name</u>	<u>Function</u>
A-C	16K RAM
B-C	64K RAM

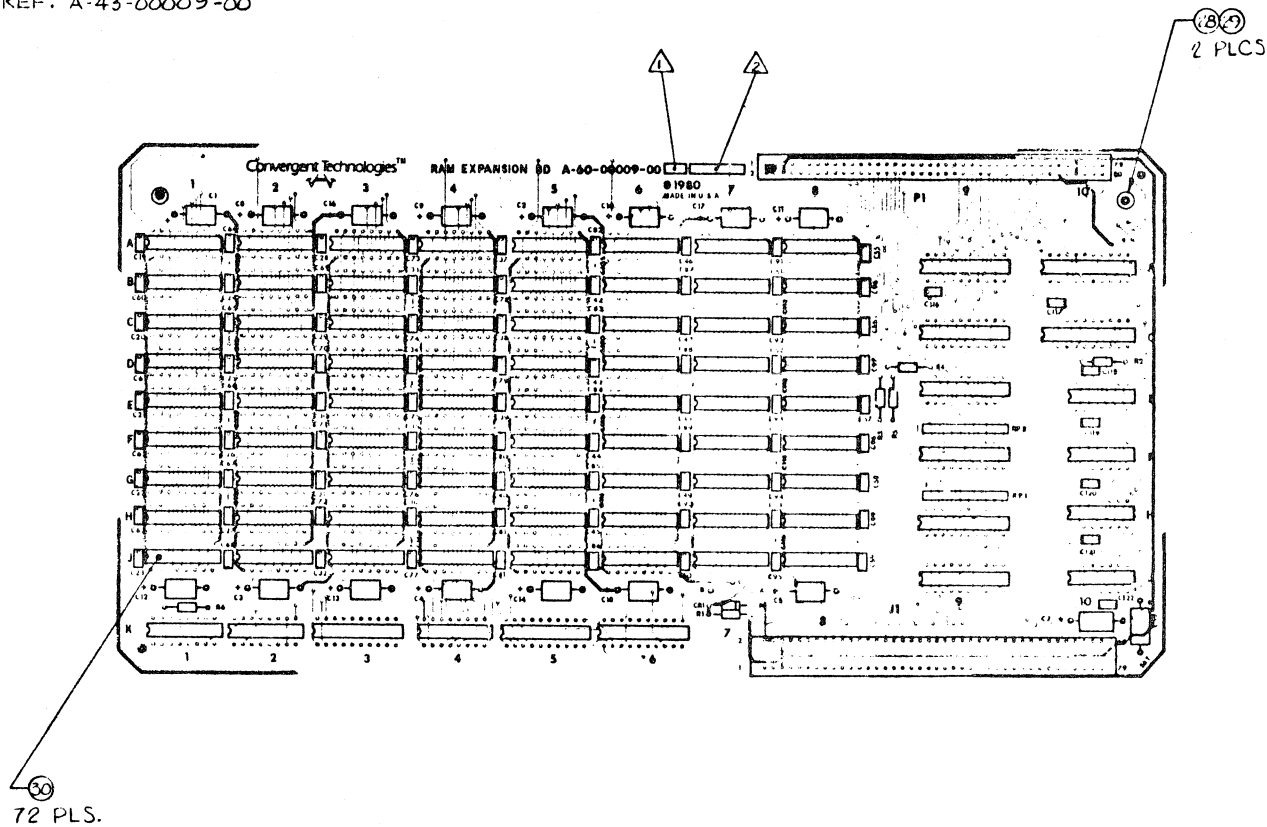


DWG. NO.		REV.	DESCRIPTION	DATE	APPROVED
1X			ENGR. PROTO.		
A			MFG. REF. DES.	10-8-80	SET

QTY REQD.		FSCM NO.	PART OR IDENTIFYING NO.	NOMENCLATURE OR DESCRIPTION		WATERAL SPECIFICATION	
PARTS LIST							
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES TOLERANCES ARE FRACTIONS DECIMALS ANGLES XX .XXX °				CONTRACT NO.			
MATERIAL				APPROVALS		DATE	
				DRAWN <i>E. Montoya</i>			
				CHECKED			
				ISSUED			
FINISH				SCHEMATIC DIAGRAM RAM EXPANSION BOARD			
DO NOT SCALE DRAWING				SIZE: FSCM NO.		DRAWN	
				D		A-43-00009-00	
				SCALE		REV A	
						15-SHEET 1 OF 2	



- NOTES:  MARK ASSEMBLY REVISION LETTER IN AREA SHOWN.
-  MARK APPROPRIATE ENGINEERING CHANGE DATE CODE IN AREA SHOWN.
3. FOR A COMPLETE LIST OF MATERIALS SEE A-60-00009-00
4. SCHEMATIC REF: A-43-00009-00



QTY REQD	FSM NO	PART OR IDENTIFYING NO	CONTRACT NO	DATE
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES TOLERANCES ARE FRACTIONS DECIMALS ANGLES AS SHOWN			PARTS LIST	
MATERIAL			APPROVALS	DATE
FINISH			CHECKED	
ISSUED				
APPLICATION			DO NOT SCALE DRAWING	

CONTRACT NO	DATE	SIZE	REV
Convergent Technologies		D	A
PRINTED CIRCUIT ASSEMBLY RAM EXPANSION BOARD		A-60-00009-00	1 OF 1

7. ROM EXPANSION BOARD

OVERVIEW

The ROM expansion board provides additional ROM and PROM capability to a CT system. It provides 16 chip sockets for an additional up to 128K bytes of read-only memory capability.

OPERATION BY LOGIC BLOCK

The sockets at 1A - 8A and 1B - 8B are designed to accomodate several different PROM types as follows:

<u>chip type</u>	<u>organization</u>	<u>number of pins</u>	<u>ROM/EPROM</u>
2316	2K x 8	24	ROM
2716	2K x 8	24	EPROM
2332	4K x 8	24	ROM
2732	4K x 8	24	EPROM
2364	8K x 8	28	ROM
2764	8K x 8	28	EPROM

All chips on the board must be of the same type.

In order to accomodate both size chips the board has 28-pin sockets. When 24-pin devices are used, they are installed justified toward the connectors (J1).

Note that the schematic is numbered according to the sockets (28 pin numbering). Therefore when 24-pin devices are installed, the chip pin numbers will not match the pin numbers on the schematic. The relationship is as follows:

<u>Schematic pin number</u>	<u>24-pin chip pin number</u>	<u>Schematic pin number</u>	<u>24 pin chip pin number</u>
1	not used	28	not used
2	not used	27	not used

3	1	26	24
4	2	25	23
5	3	24	22
6	4	23	21
7	5	22	20
8	6	21	19
9	7	20	18
10	8	19	17
11	9	18	16
12	10	17	15
13	11	16	14
14	12	15	13

On pin 23 of the socket, the 2316 and 2716 require +5 volts and all other chip types require address bit C. This selection is made with bits 6 and 7 of the dip switch on the board. Bit 6 is closed to connect the address bit and bit 7 is closed to connect +5.

The array is organized as 8 two-chip banks with each two chips representing one 16-bit word. The "A-ROW" chips (1A-8A) are the high order or odd byte chips (corresponding to MEM8+ - MEMF+). The "B-ROW" chips (1B-8B) are the low order or even byte chips (corresponding to MEM0+ - MEM7+).

The address organization of the array depends upon the type of chip used and the corresponding setting of the decode switches. This chart gives that correspondance. The setting of the switches determines what local address lines are used to decode the proper bank.

PROM/ROM chip used	Switch setting					Maximum Memory	Address Lines Decoded		
	1	2	3	4	5		high	medium	low
2316/2716	0	0	1	1	0	32K	LAE+	LAD+	LAC+
2332/2732	1	0	0	0	1	64K	LAE+	LAF+	LAD+
2364/2764	0	1	0	0	1	128K	LAE+	LAF+	LA10+

NOTE: 1 = on
0 = off

Address-range to bank-number correspondence:

2316/2716

<u>Bank number</u>	<u>Address Range</u>
8A AND 8B	F0000-F0FFF
7A AND 7B	F1000-F1FFF
6A AND 6B	F2000-F2FFF
5A AND 5B	F3000-F3FFF
4A AND 4B	F4000-F4FFF
3A AND 3B	F5000-F5FFF
2A AND 2B	F6000-F6FFF
1A AND 1B	F7000-F7FFF

2332/2732

<u>Bank number</u>	<u>Address Range</u>
8A AND 8B	E8000-E9FFF
6A AND 6B	EA000-EBFFF
4A AND 4B	EC000-EDFFF
2A AND 2B	EE000-EFFFF
7A AND 7B	F0000-F1FFF
5A AND 5B	F2000-F3FFF
3A AND 3B	F4000-F5FFF
1A AND 1B	F6000-F7FFF

2364/2764

<u>Bank number</u>	<u>Address Range</u>
8A AND 8B	D8000-DBFFF
4A AND 4B	DC000-DFFFF
6A AND 6B	E0000-E3FFF
2A AND 2B	E4000-E7FFF
7A AND 7B	E8000-EBFFF
3A AND 3B	EC000-EFFFF
5A AND 5B	F0000-F3FFF
1A AND 1B	F4000-F7FFF

The decoder at 10A does the actual bank selection when PROMEXDC- from the CPU address logic is active. PROMEXDC- is buffered at 12B-9.

Chips 11B and 12B are local address bus buffers. Chips 9B and 10B buffer the data from the ROMs onto the MEM (MEM0+ - MEMF+) bus. The data buffers are enabled by the address decode line PROMEXDC- from the processor board.

AC CHARACTERISTICS

Refer to AC Characteristics in Chapter 4
(Processor Board)

DC CHARACTERISTICS

Refer to DC Characteristics in Chapter 4
(Processor Board)

REFERENCES

Edge-Connector Pin List

Connector J1

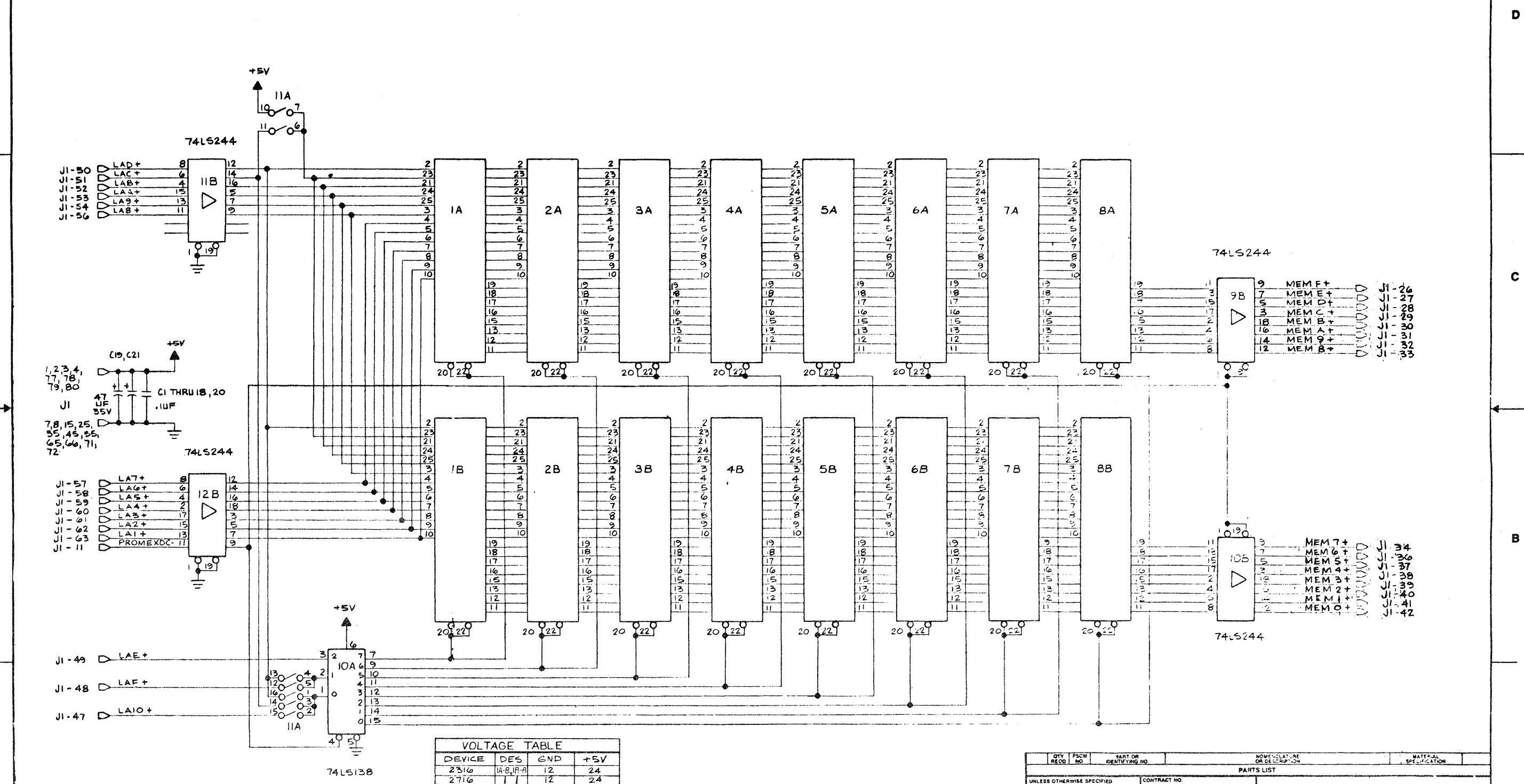
<u>Pin</u>	<u>Signal</u>
1	+5V
2	+5V
3	+5V
4	+5V
5	-12V
6	-12V
7	GND
8	
9	BD2RAM-
10	MR-
11	PROMEXDC-
12	COLMPX-
13	RFDONE-
14	RARAS-
15	
16	RFGO-
17	RALOWE-
18	RAHIWE-
19	CAS7-
20	CAS6-
21	CAS5-
22	CAS4-
23	HIP+
24	LOP+
25	
26	MEMF+
27	MEME+
28	MEMD+
29	MEMC+
30	MEMB+
31	MEMA+
32	MEM9+
33	MEM8+
34	MEM7+
35	
36	MEM6+
37	MEM5+
38	MEM4+
39	MEM3+
40	MEM2+
41	MEM1+
42	MEM0+
43	LA13+
44	LA12+
45	

<u>Pin</u>	<u>Signal</u>
46	LA11+
47	LA10+
48	LAF+
49	LAE+
50	LAD+
51	LAC+
52	LAB+
53	LAA+
54	LA9+
55	
56	LA8+
57	LA7+
58	LA6+
59	LA5+
60	LA4+
61	LA3+
62	LA2+
63	LA1+
64	LA0+
65	
66	
67	+12V
68	+12V
69	+12V
70	+12V
71	
72	
73	-12V
74	-12V
75	
76	
77	+5V
78	+5V
79	+5V
80	+5V

Switches

See text of this chapter.

Dwg. No.		SH	REV	1
REVISIONS				
ZONE	REV.	DESCRIPTION	DATE	APPROVED
RO		ENGR. PRD		
A		MFG RELEASE	10-29-80	



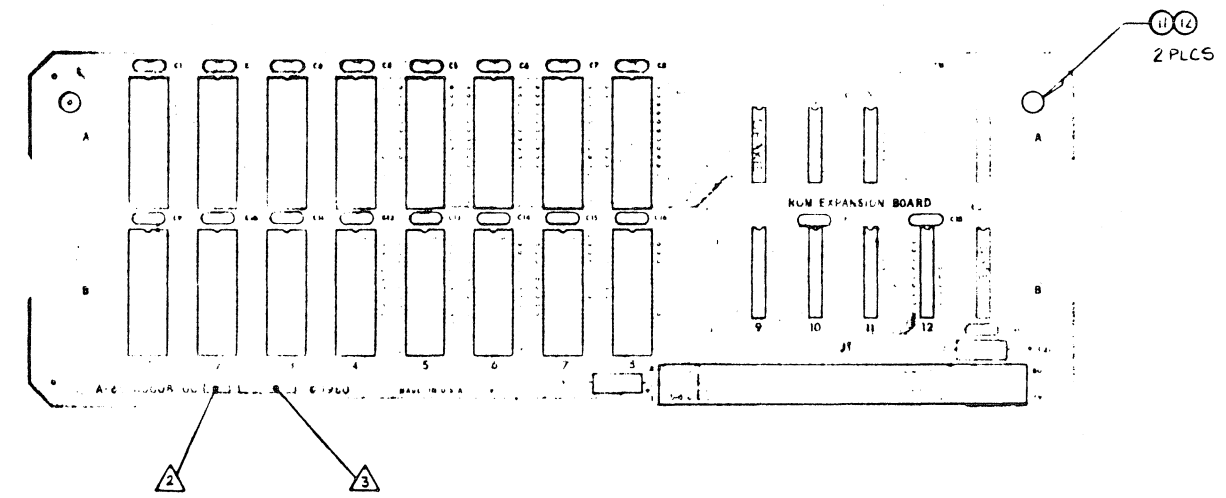
VOLTAGE TABLE			
DEVICE	DES	GND	+5V
2316	1A-8, 1B-8	12	24
2716		12	24
2332		12	24
2732		12	24
2364		14	26-28
2764	1A-8, 1B-8	14	26-28
74LS138	10A	8	16
74LS244	9B-12	10	20

QTY		FSCM		PART OR IDENTIFYING NO		NOMENCLATURE OR DESCRIPTION		MATERIAL SPECIFICATION	
RECD						PARTS LIST			
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES TOLERANCES ARE: FRACTIONS DECIMALS ANGLES 1/16 0.031 30°		CONTRACT NO.		APPROVALS		DATE		CONVERGENT TECHNOLOGIES	
MATERIAL		DRAWN		CHECKED		ISSUED		SCHEMATIC DIAGRAM ROM EXPANSION	
FINISH		NEXT ASSY		USED ON		APPLICATION		DO NOT SCALE DRAWING	
SCALE		SIZE		FSCM NO		DWG NO		REV	
D		A-43-00008-00		A		1 OF 1			

NOTES:

- 1. FOR COMPLETE LIST OF MATERIALS
SEE A-60-00008-00
- 2. MARK WITH IDENTIFICATION LETTER
IN ALL VIEWS
- 3. MARK ALL DIMENSIONS IN INCHES
CHANGING TO MILLIMETERS AS SHOWN
- 4. SCHEMATIC REFERENCE: A-43-00008-00

REV	ENG	DATE	BY
1	A	10-27-72	10-27-72



Convergent Technologies			
PRINTED CIRCUIT ASSEMBLY			
ROM EXPANSION BOARD			
A-60-00008-00			A

8. VIDEO CONTROL BOARD

FUNCTIONAL DESCRIPTION AND SOFTWARE INTERFACES

The video control board is the interface between the local bus and a high-resolution memory-refreshed CRT monitor. Its primary function is to receive symbolic data over the local data (MEM) bus and to translate this data into a video signal according to the font pattern, stored in its on-board RAM.

Display Characteristics

A single custom-designed 15-inch CRT monitor attaches to the video control board via a shielded cable. The 525-line non-interlaced raster scan image on this CRT is driven by digital (ECL) video signals at the following data rates, depending on the number of characters in the display:

34 rows of 132 characters: 51.975 Mbits/sec.
34 rows of 80 characters: 31.185 Mbits/sec.

Character and symbols are contained within a 10 x 15 dot matrix. The typical size of an upper case character is 8 x 10 dots. However, through the use of a look-up table called the Font RAM, the hardware capability of the Video Control Board supports the arbitrary definition of symbols within the 10 x 15 cell.

Character Map

The video display system refreshes data on the screen by a direct DMA transfer from the system memory to the CRT monitor. The area in memory reserved for this purpose is called a character map.

A character map contains:

- o 8-bit character codes for each character to be displayed
- o 4-bit visual attributes for each character (unless character attributes are disabled)
- o Cursor location for each line

Both the size and location of the character map may differ for different application systems. The character map must be located within the first 128K bytes of system memory, above the operating system data area.

Video Control Logic

The video control logic resides on the video control board. Its basic function, among others, is to manage the transfer and translation of data from the character map in mainframe memory to the CRT screen. It does this one character-row (i.e., one row of characters on the CRT screen) at a time.

There are three principal elements in the video logic that handle this function, as shown in Figure 8-1.:

- o Row buffers: Two buffers are used. One buffer receives a character row from the character map in memory while the other buffer reads data out to the Symbol RAM and the Attribute Logic.
- o Font RAM: This RAM on the video control board translates the symbols from the row buffers into an output data stream according to a font description.
- o Attribute logic: This combinatorial logic generates command signals to modify the output of the symbol RAM on a character-by-character basis before it is sent in a video stream to the CRT.

Row Buffers

There are two row buffers. During screen refresh, one row of characters is transferred from the character map in memory to one of the row buffers. Simultaneously, the contents of the other row buffer is read out to the Font RAM where it is translated to a dot pattern carried by the video stream to the CRT.

This process continues through all of the character rows to be displayed on the screen. Then it automatically repeats itself beginning with the top row of characters again. While one row buffer is being written into, the other row buffer is being read out of.

In the DMA transfers from mainframe memory for a given row, all character data is first transferred and then all attribute data (if programmed for use) is transferred. Each of the two row buffers can store 256 x 8 bits of character data and 256 x 4 bits of attribute data. Only 132 bytes are typically used. The first word of data for each character row is reserved for a cursor address, in the event that the cursor is on that row.

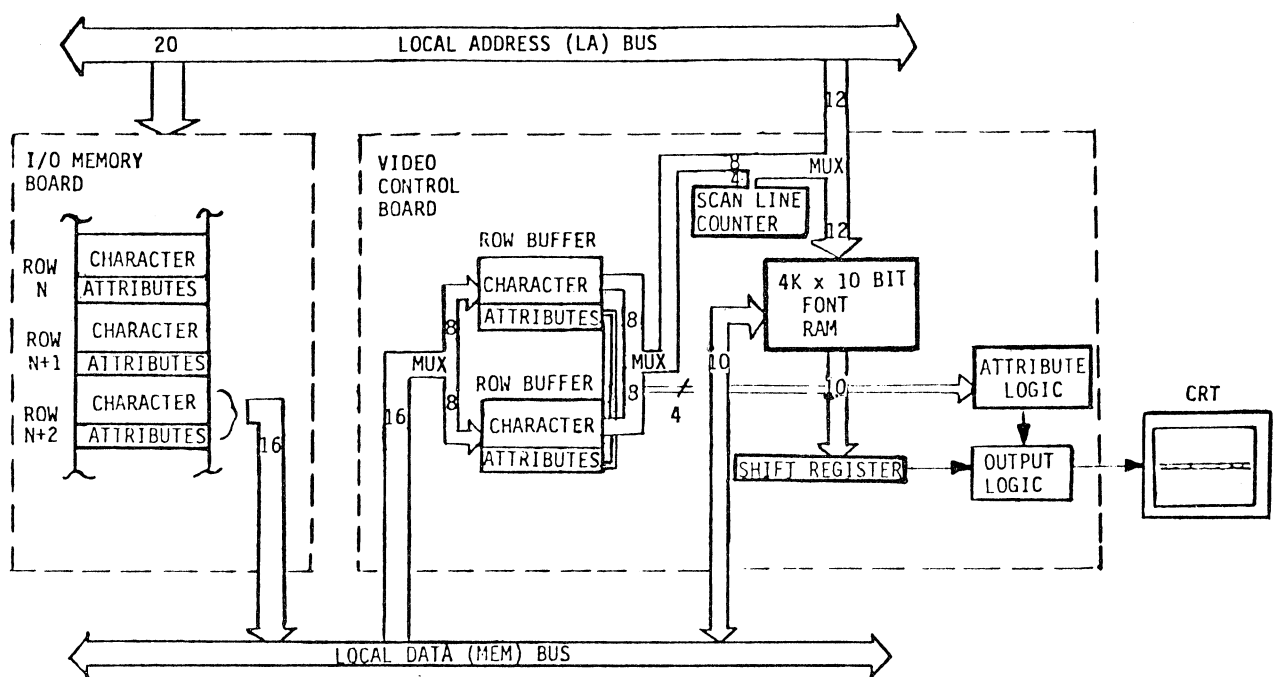


Figure 8-1. Video Control Board Data Paths

The row buffers contain their own addressing logic for use in conjunction with the DMA transfers. This logic is initialized whenever the read/write functions of the two buffers are interchanged.

Font RAM

The Font RAM is a 4K-by-10-bit array in which 256 characters can be represented as 16 10-bit words of font (dot pattern) data. One of these words is not used in the font information, leaving 15 10-bit words to represent the 15-x 10 dot matrix of each character.

The array is accessed for screen refresh and written into with the font data. It can also be accessed for reading of font data for verification purposes.

Refresh Cycles: During video refresh cycles, the character contents of one row buffer are read out to provide the high-order 8 bits of a 12-bit address to the Font RAM, as depicted in Figure 8-1. The low-order 4 bits are provided by a scan-line counter from a timing chain on the video control board.

The scan-line counter counts from 1 to 15, outputting its count on the four low-order address lines. The 15 counts provide the address for the 15 horizontal lines in the character matrix. This process is illustrated in the "Software Interface" section, below.

The output of the Font RAM is a 10-bit-wide parallel data stream that is shifted serially through the output logic to the CRT.

Writing and Reading: The Font RAM is written into from mainframe memory, typically by the operating system during system initialization. Reading of Font RAM data to the mainframe memory can also be done when verification of the array's contents is desired for diagnostic purposes.

The software interface for the Font RAM is described in a separate subsection, below, and is illustrated in Figure 8-5.

Attribute Logic

While character data is being read out of the row buffer to address the Font RAM, the associated attribute data is read out of the same row buffer into the attribute logic. Here, the attributes for each character are compared with screen attributes specified in the programmable Screen Attribute Register (SAR) to determine how to modify the outgoing video stream for a given character.

Attribute modification, if applicable, is done in the output logic to produce characters which are in reverse video, blinking, underline or half-bright.

DMA Transfers

Screen Attribute Register (SAR)

All refresh and Font-RAM transfers are controlled by the DMA logic. The only type of video-related transfer that is not controlled by the DMA logic is the control-status transfer between the CPU and the Screen Attribute Register, port 58.

The Screen Attribute Register controls all video-logic activity, as well as global screen attributes such as blanking (clearing), reverse video, number of characters per row, half-bright and enabling of character attributes. The software interface for the SAR is described in a separate subsection, below, and is illustrated in Figure 8-2.

Transfer Control

Video DMA transfers have the following characteristics:

<u>Function</u>	<u>8237 Mode</u>	<u>Block Length</u>
Font RAM (Read or Write)	Demand	4K words
Refresh	Demand	3 words*

* Except first transfer for each character row, which is 4 words.

DMA transfers to or from Font RAM involve a single, continuous block transfer of 4K words between mainframe memory and the Font RAM.

At the first initialization of the refresh function, the normal mode of operation, the DMA logic begins to continuously transfer the contents of the character map in 3-word blocks; i.e., the DMA takes the bus, transfers 3 words, releases the bus, then takes the bus again some time later for the next 3 words until all characters and attributes for the row are transferred. With the exception of the first transfer in each character row, which contains 4 words, the character map length must be evenly divisible by 3 words, even if padding words are required.

In both cases, the transfer block lengths are enforced by a ROM-based state machine on the video control board. This control logic holds the DMA chip's video-request line active for the appropriate block length, then releases it, then reasserts it for the next block of data.

In refresh transfers the state machine causes the DMA logic to quickly fill up one row buffer in blocks of 3 words, then pause and wait for the other row buffer to be completely translated through the Font RAM. When this is finished, the DMA channel is switched by the state machine to fill up the just-read row buffer. When all character rows on the screen are transferred, the DMA autoinitializes without CPU intervention to the first row and repeats the transfers for the entire screen.

Bus Utilization

Local address (LA) and data (MEM) bus utilization from video-refresh activity typically accounts for 10% to 15% of the total available bus time.

The video requirement can be lessened significantly by eliminating the use of attributes, which then do not need to be transferred from mainframe memory to the row buffers. This is accomplished by programming bit 2 of the Screen Attribute Register.

Software Interface

The operating system and standard object modules permit initializing and controlling all hardware elements and display options of the video interface. These system routines make use of the interfaces explained below.

Screen Attribute Register (SAR)

The 16-bit Screen Attribute Register, port 58, is used to control and monitor video activity. It accepts control information on its lower byte and provides status information on both lower and upper bytes.

The bit assignments for the register, illustrated in Figure 8-2, are as follows:

<u>Bit</u>	<u>Function</u>	<u>Meaning</u>
F-A	STATUS	Not used. Normally 1.
9	STATUS	Screen refresh rate, as selected by a video-board jumper (0=50 Hz, 1=60 Hz).
8	STATUS	Not used. Normally 0.
7	CONTROL/STATUS	Screen half-bright (0=normal, 1=half-bright)
6	CONTROL/STATUS	DMA control (see Figure 8-2).
5	CONTROL/STATUS	DMA control (see Figure 8-2).
4	CONTROL/STATUS	DMA control (see Figure 8-2).
3	CONTROL/STATUS	Selects number of characters per row (0=132, 1=80)
2	CONTROL/STATUS	Attribute enable (0) and disable (1). When attributes are disabled, they are not transferred to the video logic; this improves bus availability for other transfers.

- | | | |
|---|----------------|--|
| 1 | CONTROL/STATUS | Screen reverse video
(0=normal, 1=reverse
video). This affects the
entire screen. |
| 0 | CONTROL/STATUS | Screen blanking (1=dis-
play images, 0=turn
entire screen off and
disable refresh trans-
fers. |

All control functions within the SAR are reset to zero by the system Reset function.

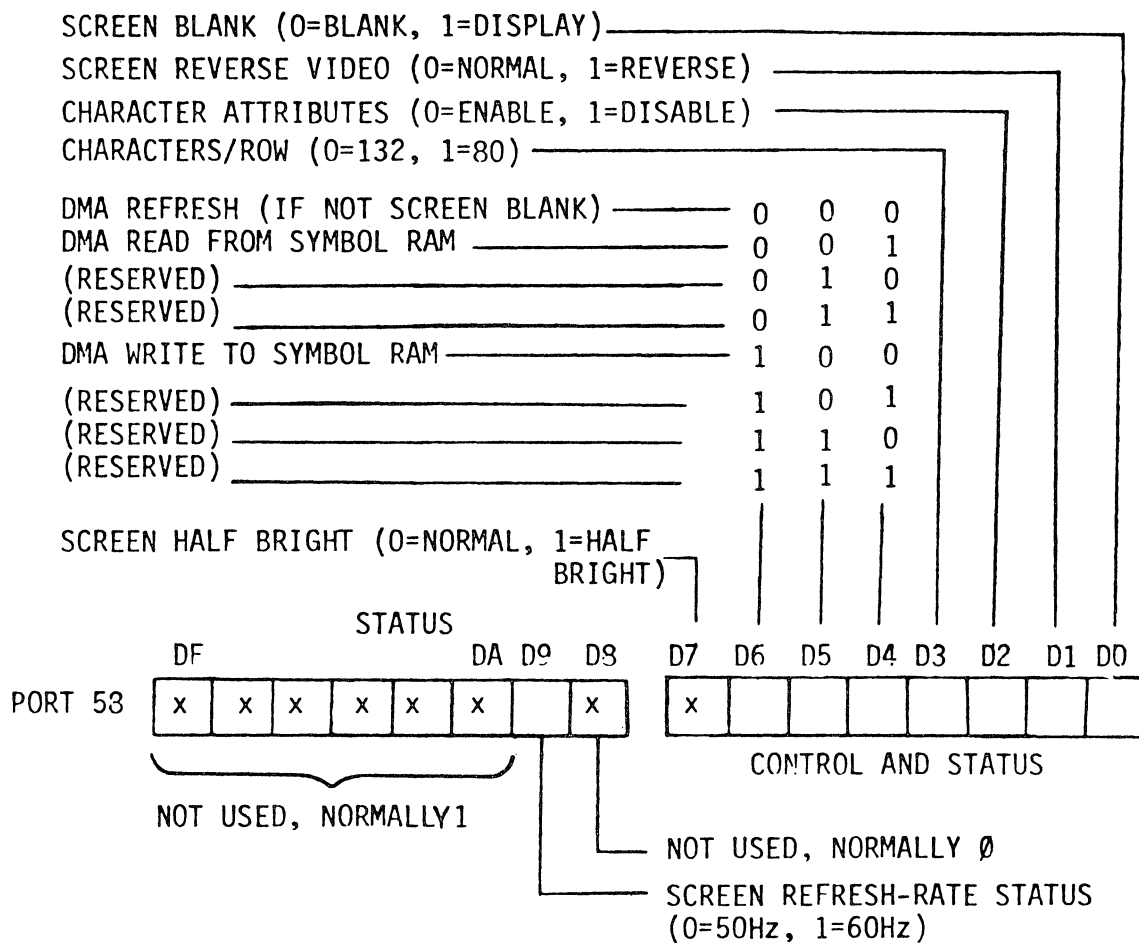


Figure 8-2. Screen Attribute Register (SAR)

Character-Map

The character map includes the 8-bit symbols for all characters to be displayed on the screen. It may also include 4 bits of attribute data for each character. Bit 2 of the Screen Attribute Register specifies whether or not the attributes are present.

The symbols for a given row of 80 to 132 characters are followed immediately by the corresponding attributes. There are 34 rows on the standard CRT screen.

The first word of the character map for each row is reserved for a cursor address, in the event that a cursor is located on that row. If there is no cursor on that row, its address is zero. All DMA transfers except the first one for each row are done in blocks of three words; the first transfer for the row (i.e., the transfer which includes the extra word for the cursor address) is done in a block of four words.

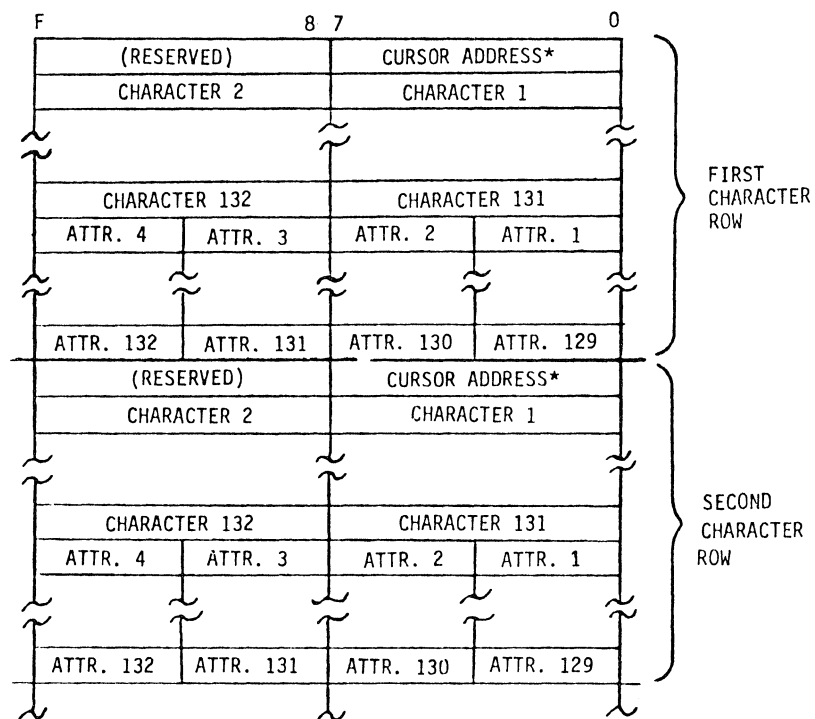
Padding of the character map is necessary if the total number of words to be transferred for each row (character plus attributes), after subtracting the cursor word at the beginning, is not evenly divisible by three. Figure 8-3 illustrates the necessary arrangement of the character map for 132 characters per row when attributes are included. Figure 8-4 shows the format for 80 characters per row. In both figures, the lowest memory address of the map represents the top-most character row on the CRT screen. Character and attribute numbering in these figures corresponds to column numbering on the CRT screen.

Maximum memory space requirements for 34-row character maps, including cursor word and padding for both characters and attributes, are as follows:

<u>characters</u> <u>per row</u>	<u>stored words/row</u>		<u>full-screen</u> <u>memory space</u>
	<u>symbol</u>	<u>attrib</u>	
132	66	33	6800 bytes
80	42	21	4352 bytes

A symbol-address counter on the video board controls all addressing of the row buffers when the row information is being sent from the mainframe memory.

LOW ADDRESS

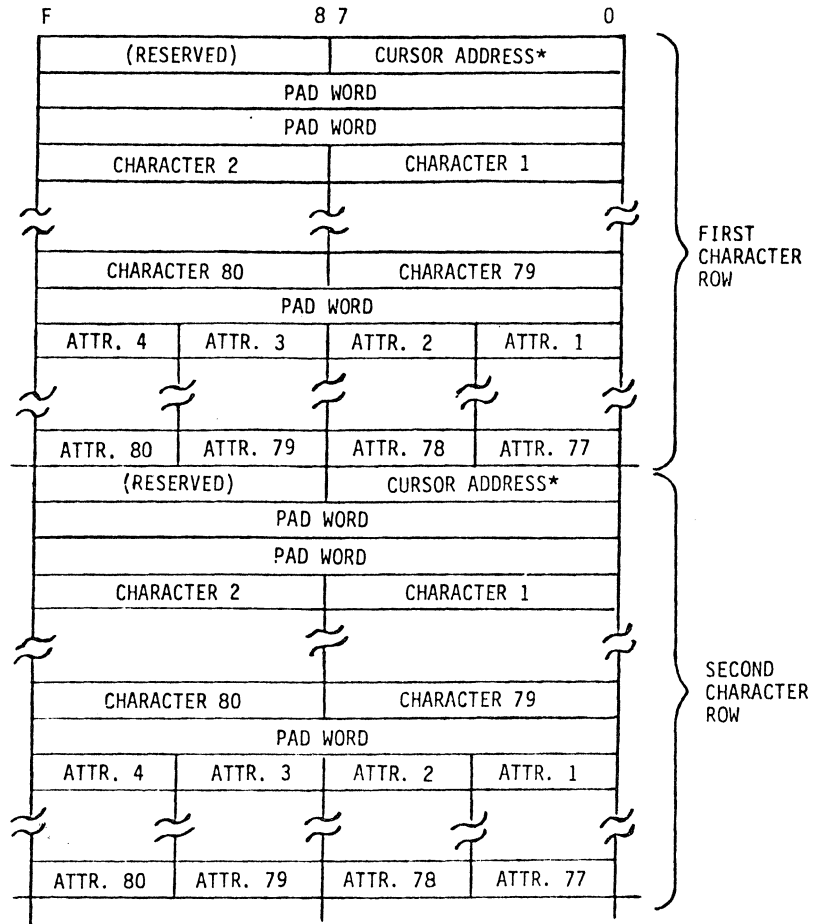


HIGHER ADDRESS

* ENTER ZERO IF THERE IS NO CURSOR ON THE ROW.

Figure 8-3. Character Map Format for 132 Characters/Row

LOW ADDRESS



HIGHER ADDRESS

* ENTER ZERO IF THERE IS NO CURSOR ON THE ROW.

Figure 8-4. Character Map Format for 80 Characters/Row

Font-RAM

The Font RAM translates eight-bit codes into video data according to the data in the Font RAM. The RAM is normally written into during system initialization.

There is space in the Font RAM for 256 symbols. Each symbol has its font data encoded in 16 10-bit words, one of which is not used. This corresponds to the 15 x 10 dot pattern of each character.

The entire block of font data must be word-aligned, starting on a 4K-word (8-kilobyte) boundary. Furthermore, the addresses in mainframe memory must correspond, in their low-order 13 bits (bit 0 is always 0), to the 12-bit addresses required for the Font RAM. Figure 8-5 illustrates the format of this addressing scheme.

The top of Figure 8-5 illustrates the contents of the two mainframe buses during downloading of the character "E". The address on the local address (LA) bus is shifted one bit to the left so as to address only words. This shifting, which is performed transparently by the standard operating system and the hardware, is described in the "DMA Transfers" section of Chapter 2, under the subsection entitled "Address Generation". It is illustrated in more detail in Figure 2-11.

The bottom of Figure 8-5 illustrates the addresses generated by video logic for the Font RAM during a refresh cycle. The scan-line count is held constant until all characters in the row (80 or 132) have been incremented to access the dot pattern for the next scan line over the entire row. The characters in the row buffers are therefore read out 15 separate times during the display of a single row of 80 or 132 characters.

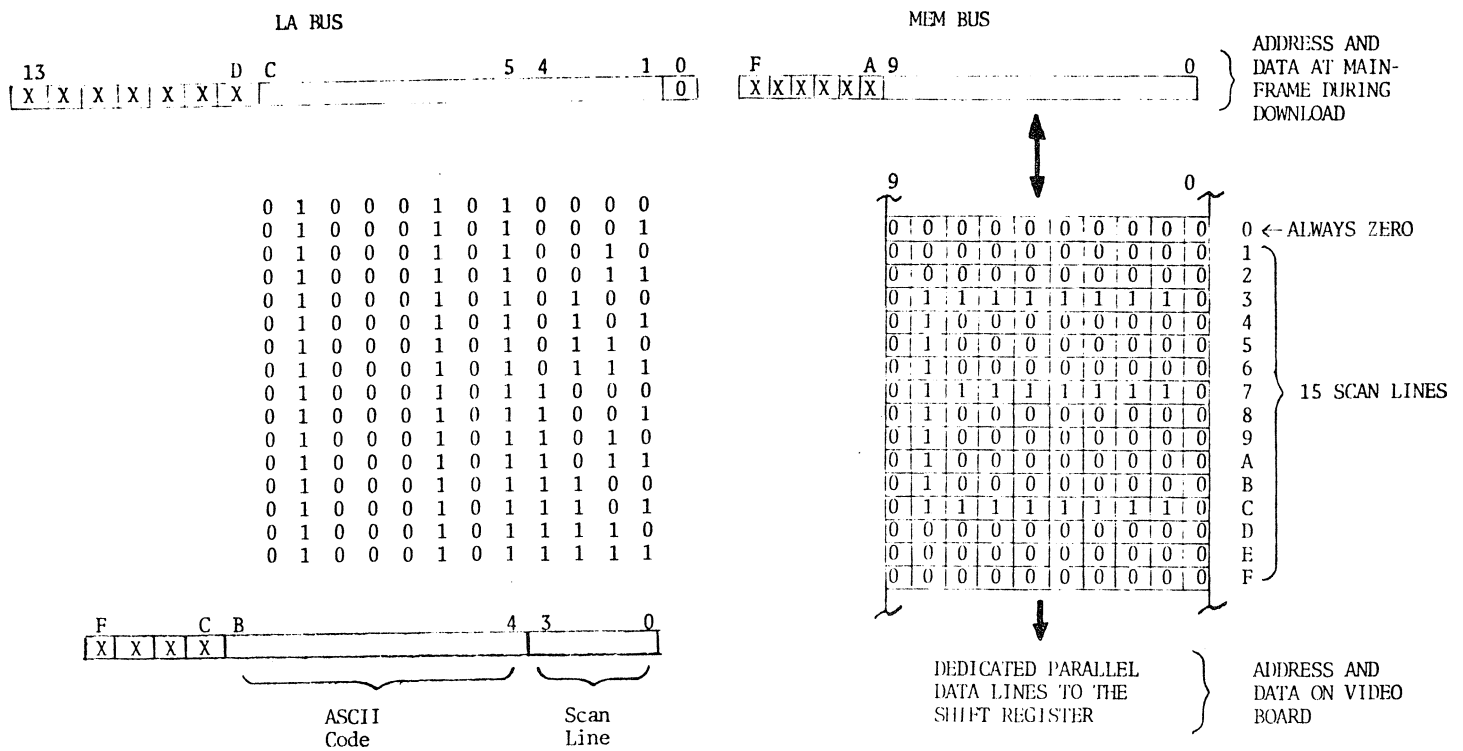


Figure 8-5. Font-RAM Addressing and Data

OPERATION BY LOGIC BLOCK

This section describes the hardware architectures of the video subsystem. A block diagram of the video control board is given in Figure 8-6.

Concisely, the architecture supports local storage of successive rows of character and attribute information. Each row is loaded through a DMA transfer from the system memory. The video board has access to the system's local data (MEM) bus and local address (LA) bus.

The DMA transfers are synchronised to the CRT horizontal and vertical scanning rates.

The character rows are double-buffered to allow for simultaneous screen refresh while loading the buffer for the next row.

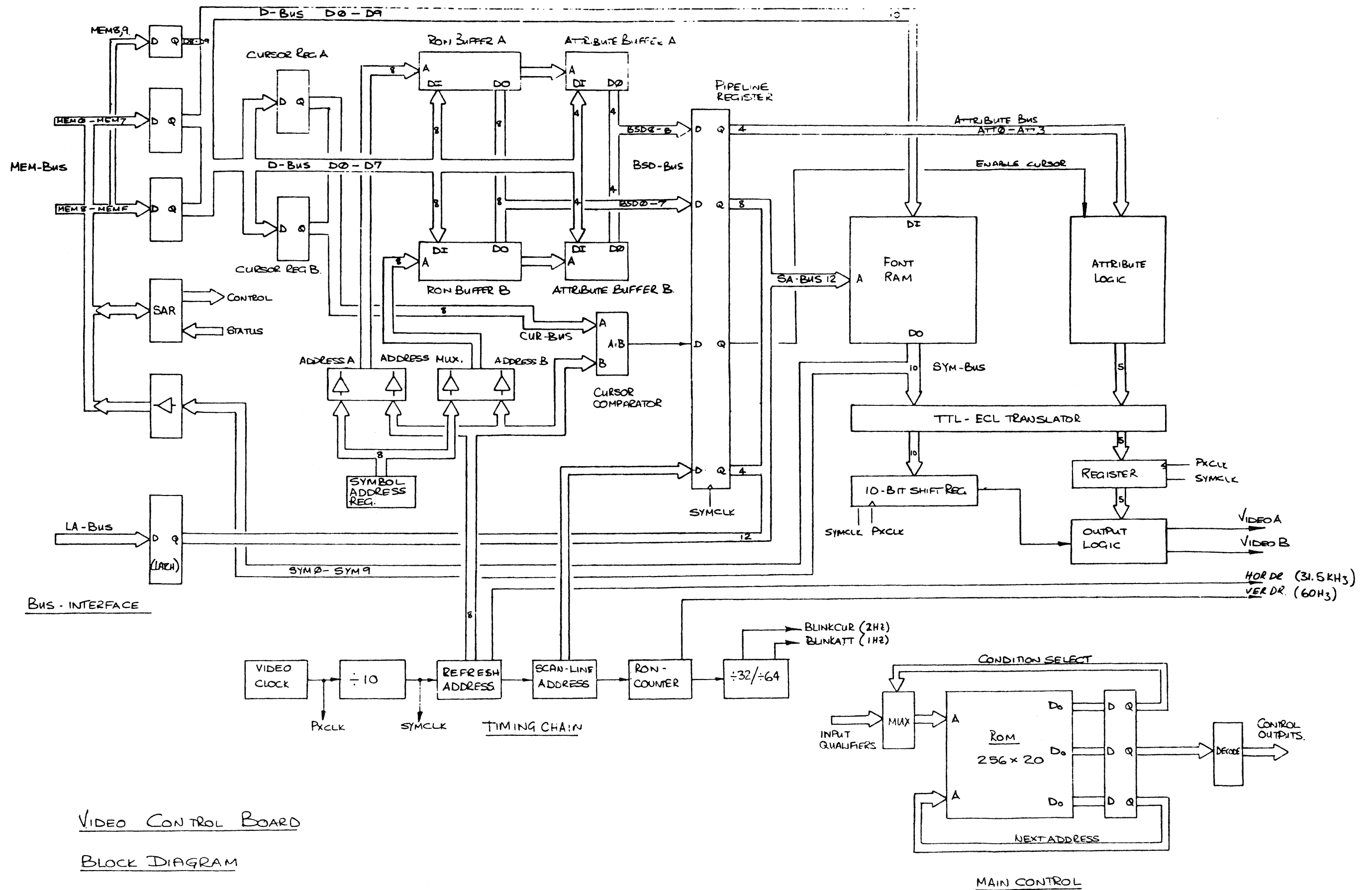
Output from the character row buffers is pipelined to the Font RAM and forms the RAM address.

The RAM output contains the actual character data and is converted to a serial form and combined with attribute control lines to form the video stream for the CRT monitor.

Bus Interface

All data, status and instruction transfers to the video board occur over the local data bus. Under DMA-logic control, data is clocked into a 16-bit holding register whose outputs can be multiplexed onto the lower 8 bits of the D-bus, as shown in Figure 8-7. The D-Bus is used for loading the Font RAM, row buffers and cursor-position holding register. When the Font RAM is read to mainframe memory, data is transferred from the Symbol Bus to the lower 10 bits of the MEM bus. The Screen Attribute Register is loaded and read directly from the MEM bus under programmed I/O control. See the schematics at the end of this chapter.

D-flip-flop 6H, clocked by VERBLANK+, synchronizes the output from bit 0 of the Screen Attribute Register (GBLANK-). Thus, unblanking of the screen begins only at the start of screen refresh cycles.



VIDEO CONTROL BOARD

BLOCK DIAGRAM

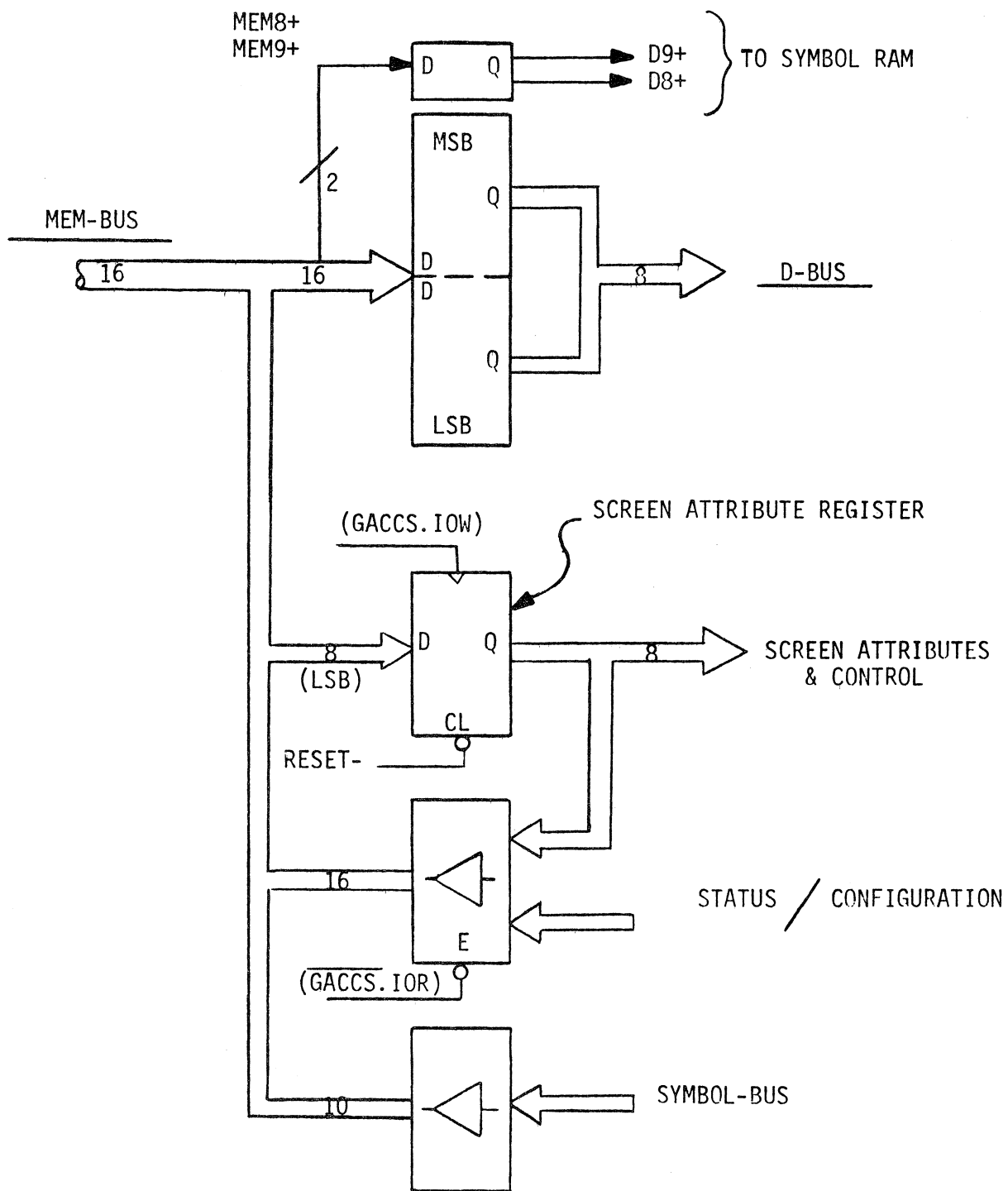


Figure 8-7. Bus Interface

Row Buffers

The two sets of row buffers (character and attribute) are shown in Figure 8-8. A character row buffer contains 256 bytes and an attribute row buffer contains 256 4-bit nibbles. Each character occupies 1 byte and each attribute occupies 1 nibble. During screen refresh, one set of buffers is written into while the other set is read out of. This process continues through all of the character rows to be displayed on the screen. The process repeats itself beginning with the top row of characters again. The signal to switch roles of the row buffers comes from the video timing chain just as horizontal blanking begins at the end of a character row. Flip-flop H6 converts this signal (ROWCLK-) to control lines SELA- and SELB-.

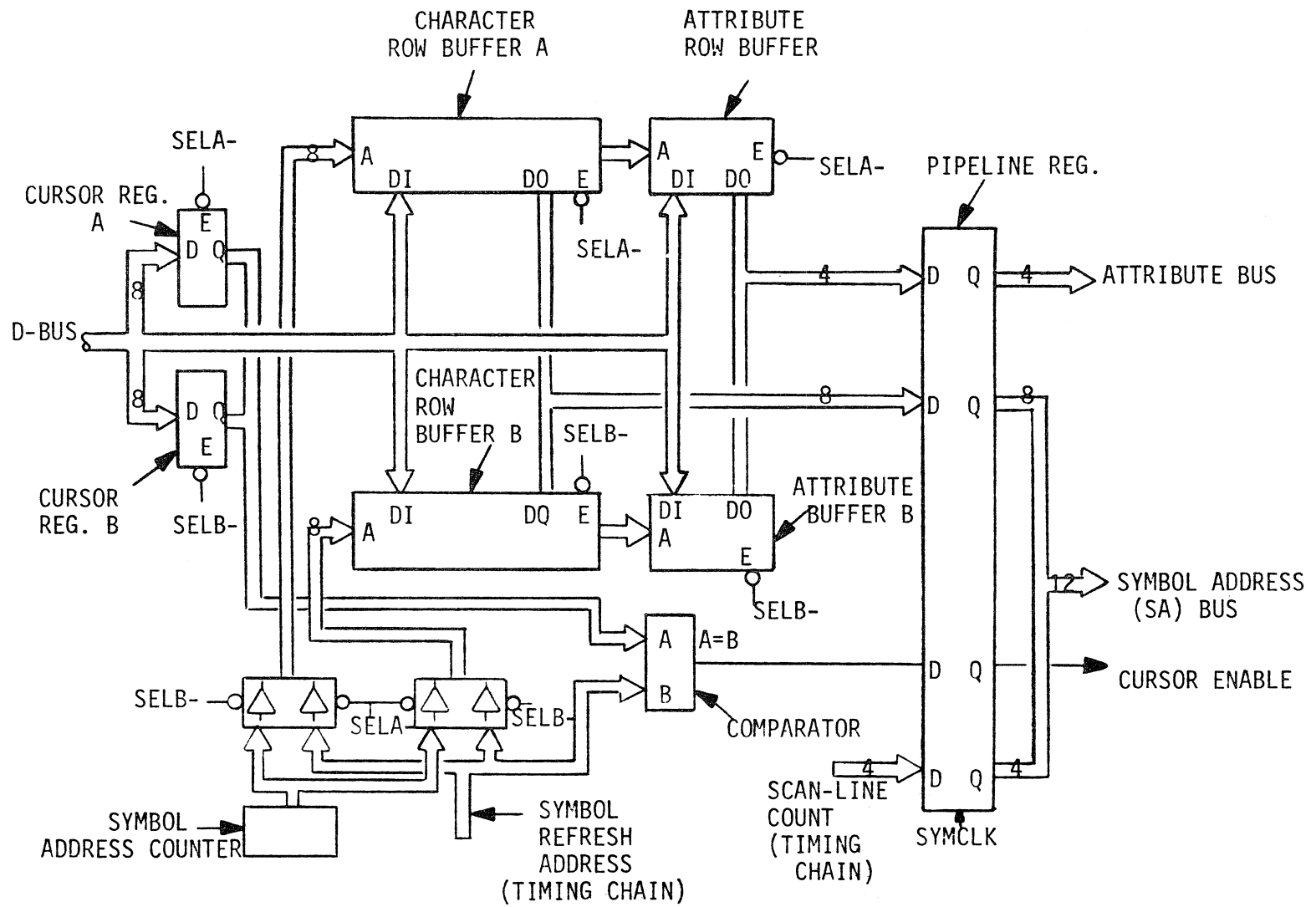
Loading the Row Buffers

Transfers of data into the row buffers proceeds as follows:

- o The state machine (see "Main Control," below) loads the symbol address counters 3C and 4C with a starting address and issues a DMA transfer request.
- o In response to a DMA acknowledgement, the bus control logic splits incoming words into successive bytes and writes them to the D-Bus.
- o The state machine provides write signals which store these character bytes in the character row buffers and increment the symbol address counters. Each three-word burst in a DMA transfer contains six characters.

Character data always precedes attribute data for any given row, and attribute data may be suppressed in order to reduce bus utilization. If attributes have been suppressed (NOATTR - asserted), the transfer is terminated after characters have been loaded. Otherwise, the symbol address counter is again set to its starting address and data transfer continues as before except that words are broken into nibbles and written into the attribute buffer (9C or 9D).

Figure 8-8. Row Buffers



Reading out the Row Buffers

The symbol refresh counter in the video timing chain provides column addresses for reading out the row buffers. The output is a twelve-bit data stream (BSD0...B+) composed of the character and attribute to be displayed during the subsequent period of the symbol clock. In order to allow for RAM access time, data carried by BAS0...B+ is pipelined. Register 11D is used to pipeline BSD8-BSDB+ (the 4 attribute bits) and 11E is used to pipeline BSD0-BSD7, the 8 bit character code. The output of this latter pipeline register becomes the higher-order 8 bits (SA4-SAB) of the symbol address bus. (See Fig. 8-8)

Attribute Logic

The attribute logic is responsible for the generation of control signals used to alter the output video stream. These signals are generated in response to inputs from the attribute bus (from the row-buffers) and also from the Screen Attribute Register. They are used to turn the video stream on, off inverted, or intensified, depending on the logic sense of these signals.

In addition, timing signals are also input to this logic: horizontal and vertical blanking, EOP and both a 2-H₃ and 1-H₃ clock signal for cursor and blinking, respectively.

The output control signals function as follows:

EINTENSIFY+	Used to alter the intensity of the video drive (becomes VIDEO B).
EREVID+	Used to invert the video stream (except for blanking). Thus yielding reverse video.
ECURSOR-	Used to turn on the cursor (exclusive OR'd with video).
E(BLINK + UNDERLINE) + and EBLINK	Both are used in conjunction to turn the video either on (for underline) or on and off (for blink).

When both the cursor and the underline are used, each uses the decoding of scan line 13.

Correspondence between the four character attributes and the attribute bus is as follows:

ATT0-Half bright
ATT1-Underline
ATT2-Reverse video
ATT3-Blink

Cursor control

Cursor registers 14F and 15F provide double-buffering of cursor-position information. While one of these is being used for cursor control within the present row, the other is being loaded with cursor data pertinent to the next row. ENCUR- (Enable Cursor) is active whenever the column bus, COL0...7+, points to the address stored in the cursor register being used for the present row. Generation of COL0...7+ is discussed below in "Video Timing Chain." ECURSOR- (Display Cursor) is asserted whenever all of the following conditions obtain:

ENCUR- is active
BLINKCUR is high
LIN0...3+ points to scan line 13

BLINKCUR is a 1.9Hz square wave generated by the video timing chain.

Video Generator

Figure 8-9 is a block diagram of the video generator.

The Font RAM is the 4K-word by 10-bit lookup table described in "Software Interface" above (see Figure 8-5). During refresh operation, address inputs for the Font RAM come from pipeline register 11E and 11D. These registers combine the line-count bus (LINE0...3+) with part of the buffered-symbol-data bus (BSD0...7+) to form the symbol address bus (SAD0...B+), which addresses the Font RAM. Data outputs of this RAM comprise the Symbol Bus (SYM0...9+), which translated through 6G and 8G into ECL signals. These signals are loaded by ESYMCLK- (synchronous with EPXCLK-) into registers 4G, 5G and 7G, which perform parallel-to-serial conversion of the data stream. ESYMVIDEO+, the serial output, is synchronous with the pixel clock (EPXCLK+).

Gates 2G and 3G transform the video data stream as necessary to implement reverse video and blink attributes as well as cursor display. The attribute holding register (1E and 2E) is designed to assert attribute signals synchronously with the corresponding bits of ESYMVIDEO+ from register 4G pin 2. Propagation delays involving these signals are carefully matched.

There are two outputs from the video generator: EVIDEOA drives the cathode of the CRT, and EVIDEOB controls the level of drive and thus the intensity of the electron beam.

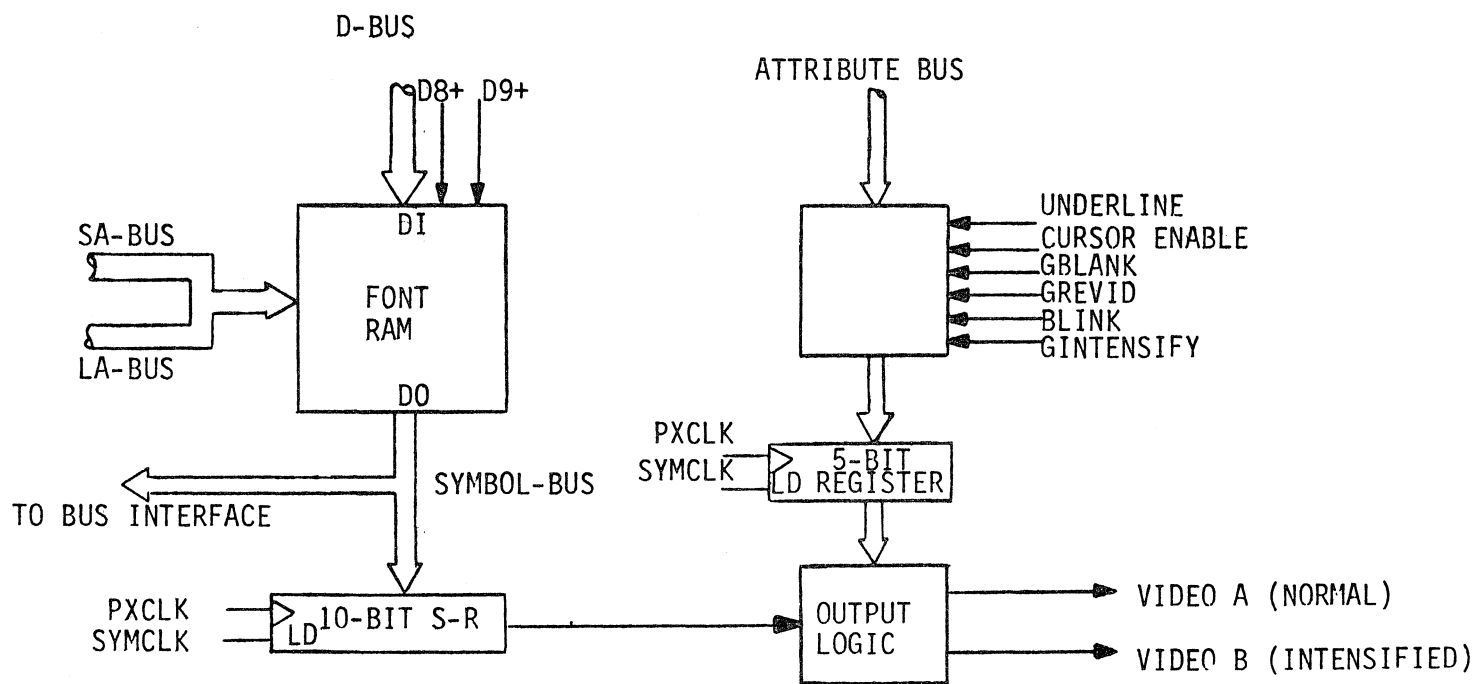


Figure 8-9. Font RAM, Attribute Logic and Video Output

Main Control

The main control logic consists of a microprogrammed state machine, as shown in Figure 8-10. There are 128 states and the output word size is 20 bits.

7 bits are used for next address generation, 4 bits are used for selection of input qualifiers. The remaining 9 bits are used for output controls.

The functions performed by the control logic are listed below:

- (i) Decode WRENO-WREN2 and generate DMA requests. Respond to request/acknowledge handshake and EOP.

Synchronise DMA transfers with horizontal and vertical sync.

- (ii) Generate appropriate read and write strobes for:

Write to character row buffers (AROWWR, BROWWR)

Write to attribute ROW buffers (AATWR, BATTWR)

Write to Font RAM (OSYMWR)

Read Font RAM

Increment and load symbol address counter. (INC+, LOADA)

Figure 8-11 is a simplified flow diagram that shows how the controller synchronizes DMA transfers and generates timing signals for most data transfers involving registers on the video board.

Figure 8-12 lists the states and associated functions of the WRENO-WREN2 control lines that are used to generate DMA requests.

The microprogram is contained in ROM chips A11 through A15. Definitions of the signals which serve as inputs are presented in "Input Qualifiers," below, and a detailed state diagram is appended to this chapter, as Figures 8-17 through 8-19.

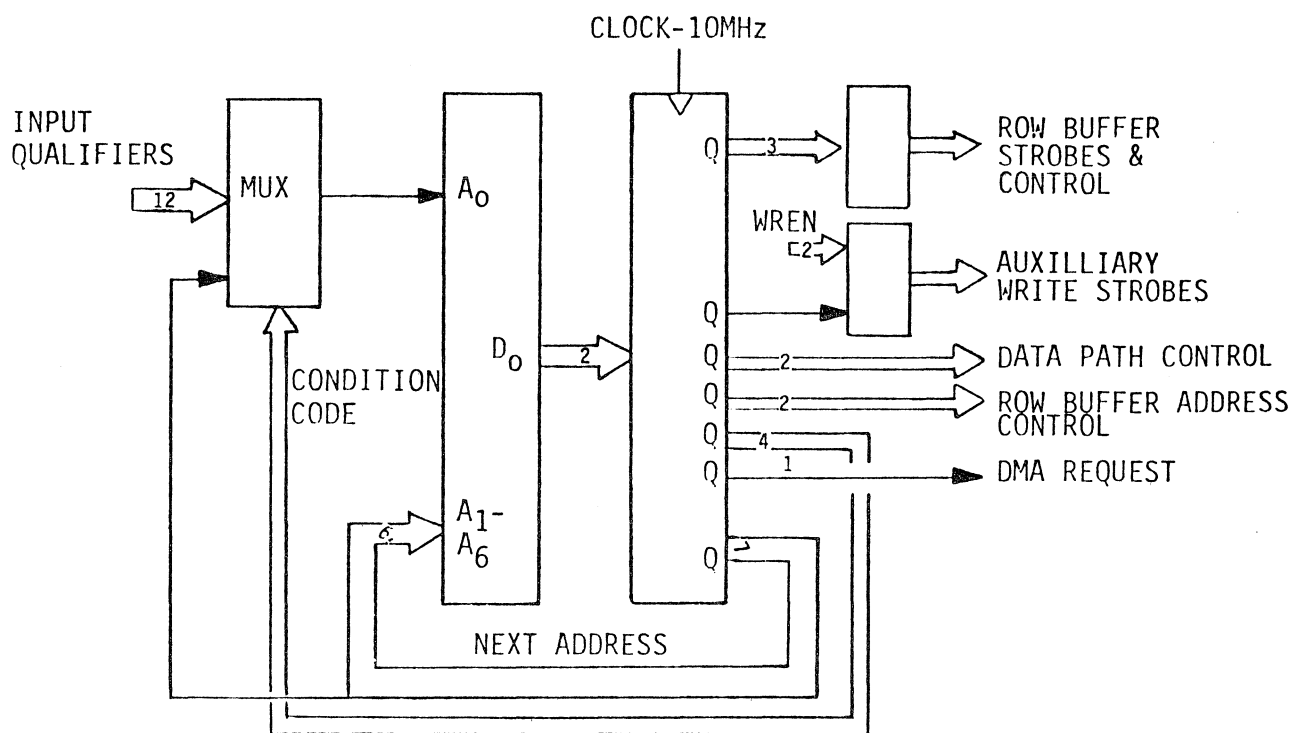


Figure 8-10. Block Diagram of Main Control Section

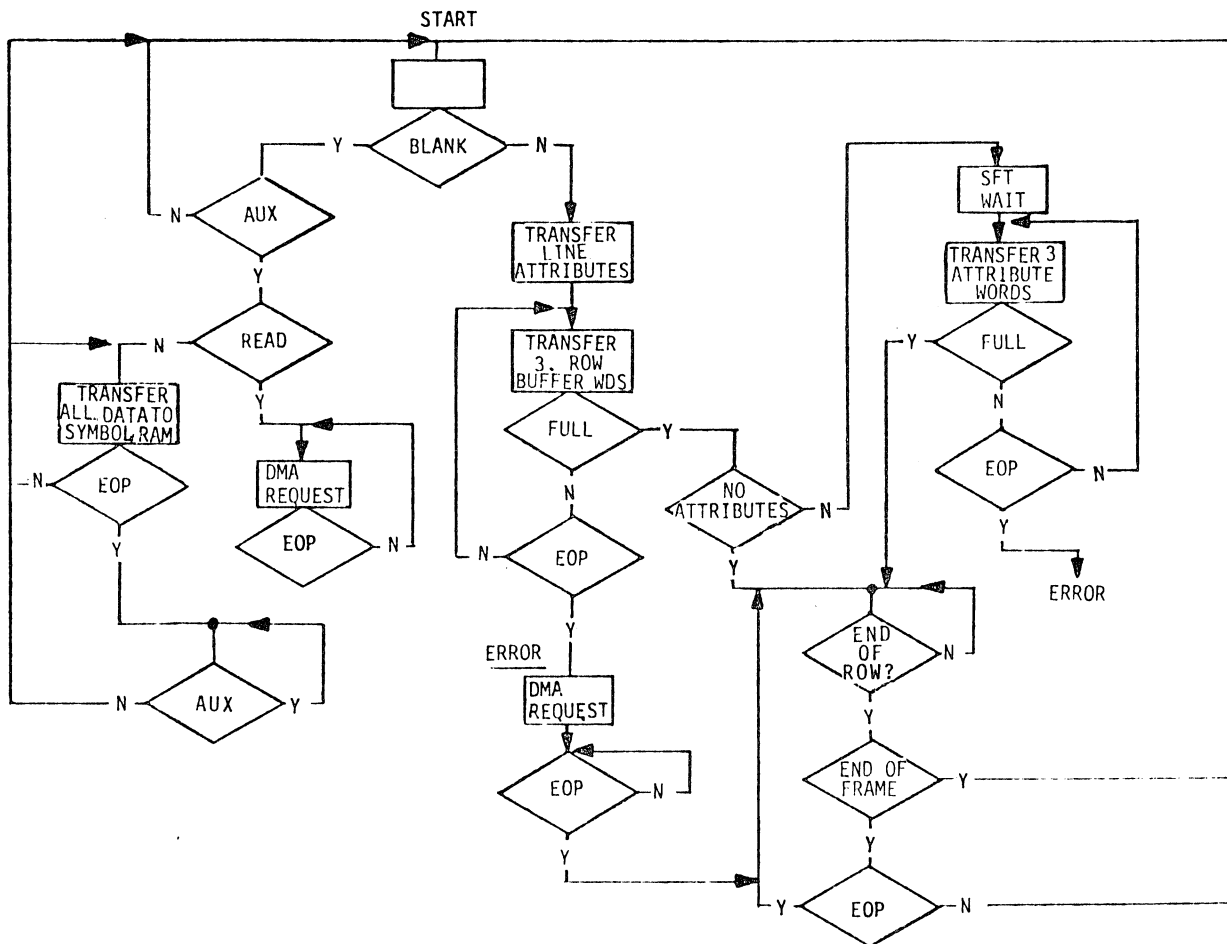


Figure 8-11. Video Control - Simplified Flow Chart

WREN2+	WREN1+	WREN0+	
0	0	0	NORMAL DMA refresh if not GBLANK
0	0	1	OSYMRD DMA read from Font RAM
0	1	0	(Reserved)
0	1	1	(Reserved)
1	0	0	OSYMWR DMA write to Font RAM
1	0	1	(Reserved)
1	1	0	(Reserved)
1	1	1	(Reserved)

Figure 8-12. DMA Request Control Lines

The Microcode Word

Figure 8-13 describes the 20-bit microcode word, which is composed of seven sections:

1. Row Buffer Strobes and Control: The 3 bits decode to generate write strobes for the row and attribute buffers to load the cursor address. In addition, strobes can be generated to set the DMA WAIT flag, to stop DMA transfers and to flag the error recovery state, 'ERROR'.
2. Data Path Control: These 2 bits are decoded to select higher and lower order bytes of the MEM-BUS for row-buffer loading and associated higher and lower order 4-bit nibbles for attribute loading.
3. Auxilliary Write: This bit is used in conjunction with WRITE-enable bits WREN1 and WREN0 to generate write strobes for the Font RAM.
4. Row-Buffer Address Control: The 2 bits are decoded to allow clearing, preloading and incrementing of the row-buffer address counter. This is used during DMA transfers to the row buffer and attribute buffer.
5. Condition Code: The condition code is used in conditional jump instructions and is used to select one of 13 input qualifiers.
6. DMA Request: This line is asserted when the controller wishes to initiate a DMA transfer to or from the video subsystem. The DMA REQUEST line sets the VIDEO REQUEST FLAG. This flag is reset at the end of a DMA transfer (provided DMA REQUEST is not still asserted) or by issuing the STOPDMA command from the state machine.

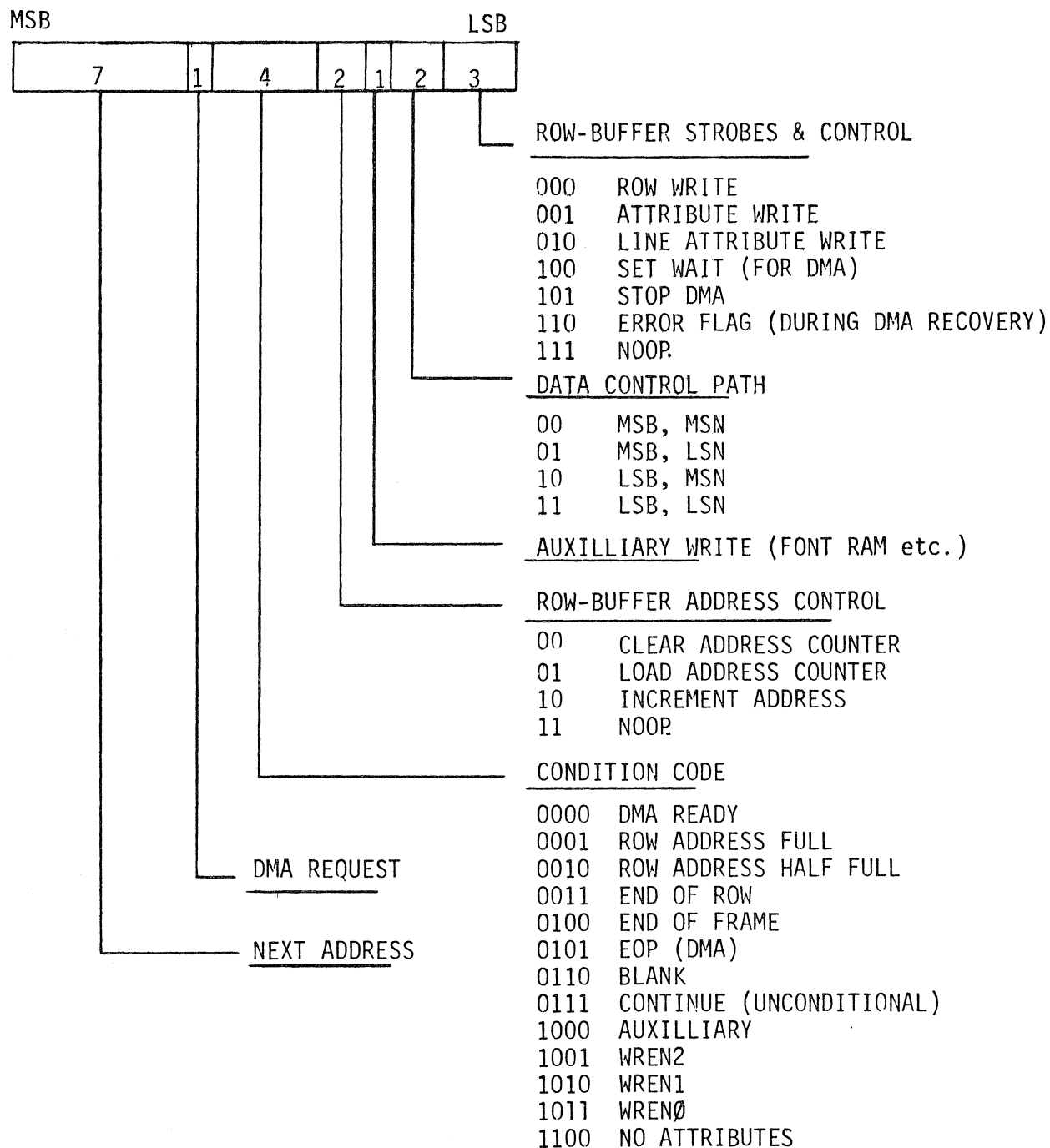


Figure 8-13. Microcode Word

7. **Next Address:** These 7-bits are used to select the next address for the state machine. The least significant bit of the next address is selected according to the state of the input qualifier selected by the value of the condition code.

Input Qualifiers

The twelve input qualifiers which serve as inputs to the controller are described below. Six of these signals come directly from the control word which the CPU writes to the Screen Attribute Register (NOATTR, BLANK, WREN0-WREN2, AUX); two signals come indirectly from the 8237-2 DMA chip on the processor board (EOP, DMARDY); and the other four signals inform the controller when key events take place on the video board (FULL, HFULL, ROWEND, FRAMEND). Because some of the input qualifiers are derived from signals which are not synchronous with MEMCLK+, the main control section D-registers B10, D15 and part of A10 are used for synchronization.

Definitions of the input qualifiers are as follows:

NOATTR- (No Attribute): This signal corresponds to bit 2 of the Screen Attribute Register (See Figure 8-2). Assertion of this signal causes the controllers to omit the DMA transfer of character attributes to the attribute row buffers. The screen attributes are not affected by this signal.

BLANK- (BLANK): This signal corresponds to bit 0 of the Screen Attribute Register. Assertion of the signal causes the controller to stop all DMA transfers. While GBLANK is asserted, the controller is able to write and read the Font RAM but it cannot perform DMA transfers to the character or attribute row buffers.

BEOP- (Buffered End Of Process): This signal, generated from the DMA control, is asserted after the last word has been transferred from the character map for each frame. Following BEOP, the screen is blanked and the controller waits until the end of the frame (FRAMEND) before initiating transfers for the next frame. BEOP should occur after a 3-word burst at the end of a row. If it occurs at any other time, an error is flagged (ERROR FLAG) and the state machine

resynchronizes to the next EOP following FRAMEEND.

WREN0+...2+ (Write Enable): These lines correspond to bits 4 through 6 of the Screen Attribute Register. See Figure 8-12.

AUX- (Auxiliary): An "auxiliary" function, such as reading reading or writing the Font RAM, corresponds to any of WREN0+...2+ being active. The function is inactive when WREN0+...2+ are all zero.

DMARDY+ (DMA Ready): This line indicates that the DMA chip has acknowledged a request for data transfer (issued by the video board) and has transferred a word of information.

FRAMEND+ (Frame End): The low-to-high transition of this video timing signal indicates that the last row has just been displayed. This transition coincides with the start of the vertical blanking time.

ROWEND- (Row End): This timing signal goes high at the end of each character row.

FULL+ (Full): The symbol address counter asserts this signal when DMA transfers have loaded all of the characters in a row into the row buffers.

DATALD- (Data Load): DATALD is effectively the logical AND of VIDACK (from the 8237 DMA chip) and IOW. Thus the trailing edge signifies the completion of a data transfer from system memory to the video control board. DATALD is synchronized in B10 to produce DMARDY (B10-8)

Controller Operation

The following discussion is a description of the state machine programs to load and read the Font RAM and to load the character and attribute row buffer. See Figure 8-11 for a simplified flow chart and Figures 8-16, 8-17 and 8-18 for a state-by-state flow chart.

While GBLANK is asserted, the screen will be blanked and no DMA transfer to the new buffer can take place. If the AUX qualifier is asserted, however, a read or write to the Font RAM can take place (determined by the state of WREN2).

In either case, latches G14 and G13 are enabled in order to allow the symbol address (SA-bus) to

be generated from the local address (LA) bus.

To write to the Font RAM a DMA request is generated (DMAREQ) by the video controller. This request remains asserted during the total transfer of Font data. The logical AND of VIDACK and IOW (from the DMA chip) gives DATALD, which is used to clock data from the MEM-bus to registers F11, F12 (and thus on to the D-bus) and also to latch address data into latches G14 and G13.

DATALD is synchronised to the controller clock to yield DMARDY. Receipt of this signal causes the controller to generate the write strobe, AUXWR (OSYMWR), to the Font RAM. The process repeats until EOP is received from the DMA chip signalling the end of Font data.

At this point, the controller clears the DMA request flag (STOPDMA) and waits for the AUX command to be cleared before returning to the start of the program.

In order to read the Font RAM, DMAREQ is again asserted. The receipt of an acknowledge from the DMA chip causes the address latches G14 and G13 to be made transparent by the command line SYMRD.

The logical AND of SYMRD- and IOR- given SYMEN-, which is used to enable buffers G11, G12 to present data out of the Font RAM (SYM-bus) to the MEM-bus. The DMAREQ remains asserted until EOP is received, signifying that all data has been transferred.

Again, the DMA request flag is cleared and the controller waits for the AUX command to be cleared before returning to the start of the program.

In order to write data to the row buffer, GBLANK is de-asserted. Clocking of D-register H-6 by vertical blank ensures that the assertion of GBLANK occurs only at the beginning of vertical blanking. The symbol address counter is initialized to the row buffer starting address (LOADA) and a DMAREQ is issued. This request is maintained while the cursor data (LATWR) is loaded and also the character-row-buffer data is loaded (ROWWR). In the case of the row-buffer data the request is asserted for 2 words of a 3-word transfer; on the third transfer the request is removed, resulting in the release of the DMA logic from video transfer at the end of the third row buffer data word.

At the end of a 3-word transfer, the controller checks to see if the row buffer is full (FULL). If not, another 3-word transfer is initiated and the process repeats (except for the loading of the cursor) until FULL is asserted, thereby signaling completion of character data transfer for the first row. At this point the controller checks the NOATTR line to determine if attributes are to be transferred. If this is de-asserted (attribute to be transferred) the controller initiates a series of 3-word transfer for attribute data and on each transfer generates the write strobe, ATTWR, for each 4-bit nibble of each word. As in transfers of character row-buffer information, the request is removed at the end of the second transfer, releasing the DMA logic to service other DMA channels.

When the attribute row-buffer is FULL (or if NOATTR is asserted) the controller has completed all data transfers for the row and waits for the end of row qualifier (ROWEND) to make a low-to-high transition.

At this point a check is made of both end of frame (FRAMEND) and EOP. In the former case, the controller returns to the start of the program and begins transfer for the next frame. In the latter case, the presence of EOP causes the screen to be blanked until FRAMEND is asserted, causing the controller to return to the start of the program. If neither is asserted the state machine returns to the start of the program for the next row of data.

A check is made for EOP at the end of each DMA transfer. EOP should always occur at the end of a character row when all character and attribute data has been loaded. If EOP occurs at any other time, this signals that an error has occurred in the transfer and that the controller has lost synchronism with the CRT timing as defined by ROWEND and FRAMEND. Under these circumstances a series of single-word transfers are initiated until a subsequent EOP is detected. At this point, the controller stops all transfers and waits for ROWEND and FRAMEND before jumping to the start of the program and continuing row-buffer transfers.

Timing chain

The timing chain generates the timing signals used to drive the display and to synchronize

video control board operations. Figure 8-6 includes a block diagram of the timing chain and Figure 8-15 summarizes timing parameters.

Crystals Y1 (31.185 MHz) and Y2 (51.975 MHz) control the frequency of a switchable dual oscillator. The input to gate F4 (pin 13) comes from the MODE A/B output of the SAR and enables either the Y1 oscillator or the Y2 oscillator in order to generate the video clock corresponding to the current mode (A or B).

This clock is an ECL signal called EPXCLK+ (gate F4 pin 2). The period of EPXCLK+ corresponds to the time which elapses while the CRT displays a single pixel. ESYMCLK-, the symbol clock, is obtained by dividing EPXCLK+ by ten. Since each character is ten pixels wide, SYMCLK repeats on a character-by-character basis for each scan line. Transistors Q1 and Q2 are used in an emitter-coupled configuration to translate this clock to produce the TTL clock SYMCLK-, which is used in the video generator and drives the rest of the timing chain.

Chips B2 and B4 form the divide-by-N symbol counter whose outputs form the column bus (COL0...7+). A7 is the line counter, which generates LINE0...3+ as the video logic produces the fifteen lines that form a row of characters. The row counter, B7 and B8, divides by 35 or 42 when the raster display rate is 60 or 50 Hz, respectively.

PROM B1 generates horizontal blanking and horizontal drive signals as a function of COL0...7+. PROM B9 generates vertical blanking and vertical drive signals as a function of the row counter outputs, and it also configures the row counter as required for 50 or 60 Hz options. Latch B6 synchronizes blanking and drive signals with respect to the symbol clock (SYMCLK).

PROM B3 sets the preload for the counters B2 and B4, depending on the sense of the MODEA/B line (either 132 or 80 characters per row). Thus, the divide ratio is altered depending on the mode such that HORDR remains constant at 31.5kHz.

Differential line driver H3 transforms the TTL signals VERDR+ and HORDR+ into balanced differential signals which are sent to the video display. These signals, LHORDR and LVERDR, provide horizontal and vertical synchronization, respectively.

<u>Parameter</u>	<u>Mode</u>	
	<u>132</u>	<u>80</u>
Frame refresh rate (Hz)	50/60	50/60
Horizontal scan rate (KHz)	31.500	31.500
Video rate (Mbits/sec)	51.975	31.185
Total scan lines/frame (60Hz)	525	525
Total character rows	35	35
Visible character rows	34	34
Total horizontal dots	1650	990
Total horizontal characters	165	99
Blanking characters 33	19	
Horizontal blanking time (uSec)	6.35	6.09
Vert. blank time (uSec, 60Hz)	476.19	476.19
Vert. blank time (mSec, 50Hz)	3.81	3.81

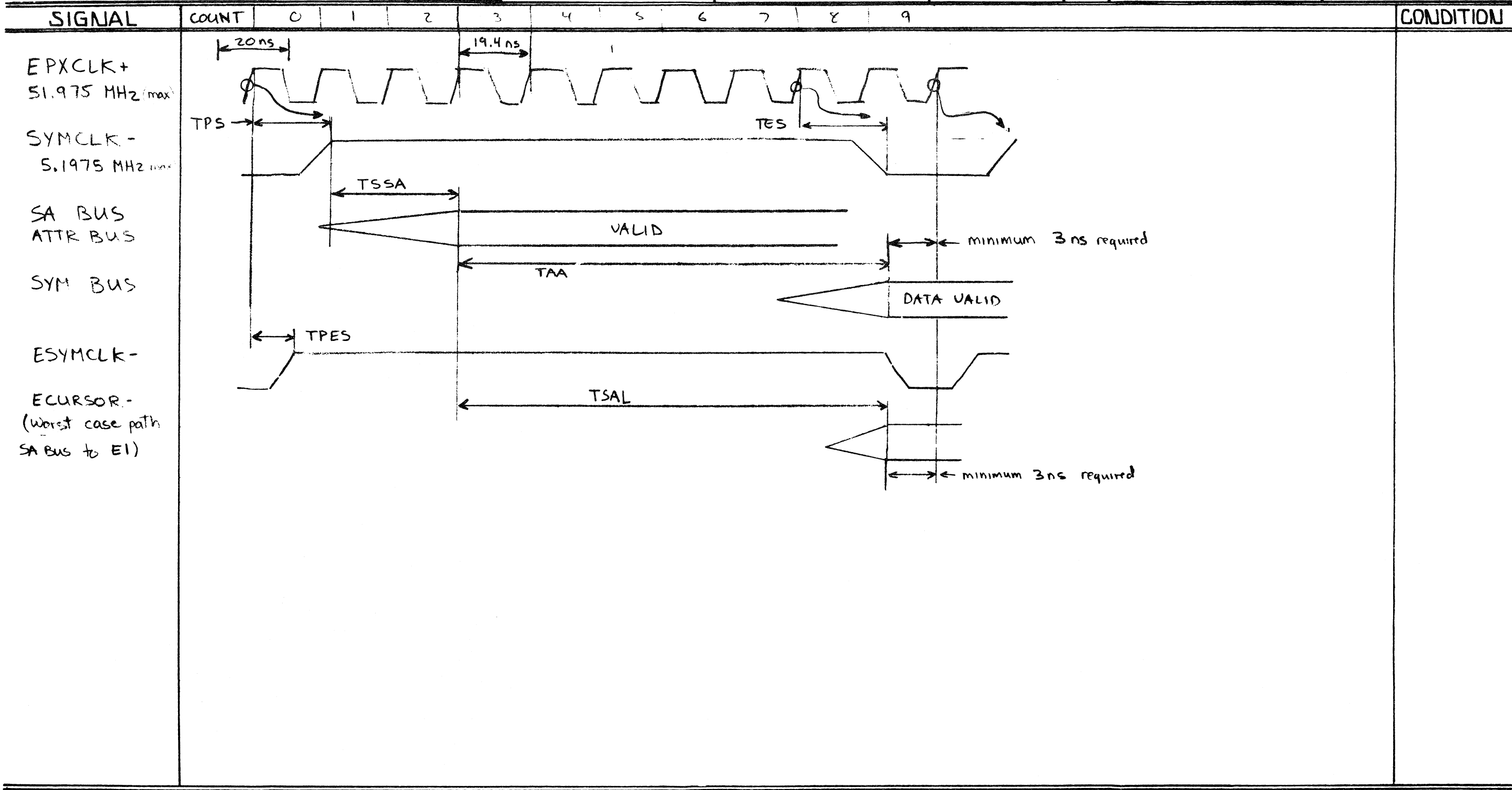
All frequencies are crystal controlled to within .01%.

Figure 8-14. Video Timing Parameters

AC CHARACTERISTICS



TITLE VIDEO-		APPROVALS	DATE	DWG NO	
2nd PIPELINE DELAY E1, E2 DII to LG4 G5 G7		DRAWN			
APPLICATION		CHECKED			
NEXT ASSY	USED ON	ENG		SHEET	REV



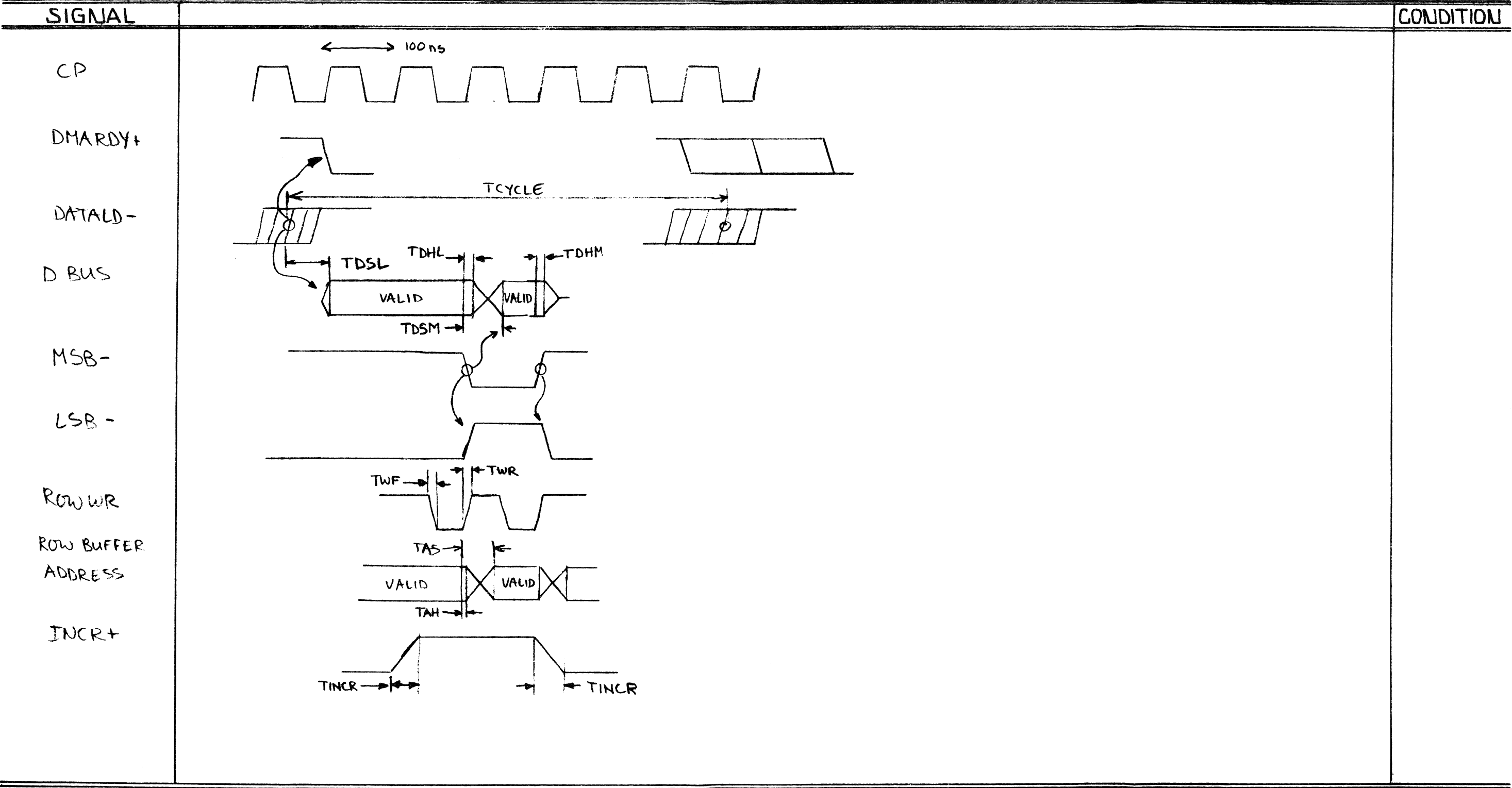


TITLE VIDEO POX DMA HEAD		APPROVALS	DATE	DWG NO	
		DRAWN			
		CHECKED			
APPLICATION				SHEET	
NEXT ASSY	USED ON	ENG		REV	

SIGNAL	CONDITION
DMA CLK 4 or 5 MHz	The diagram shows a sequence of signals over time, aligned with clock cycles S2, S3, SW, S4, and S2. DMA CLK is a periodic square wave. IOR- (at 8237) is an active-low pulse. LA Bus ADDRESS VALID is a pulse. SA Bus ADDRESS VALID is a pulse. SYM Bus DATA VALID is a pulse. MEM Bus DATA VALID is a pulse. MEMW is an active-low pulse. Timing parameters are indicated: TDCL (from IOR- to LA Bus), TDCTW (from IOR- to MEMW), TDCR (from IOR- to SYM Bus), TCY (from IOR- to MEM Bus), TASM (from LA Bus to SA Bus), TADR (from LA Bus to SYM Bus), TBA (from SA Bus to SYM Bus), TAA (from SYM Bus to MEM Bus), TBD (from SYM Bus to MEM Bus), and TDS (from SYM Bus to MEM Bus).
IOR- (at 8237)	
LA Bus	
SA Bus	
SYM Bus	
MEM Bus	
MEMW	

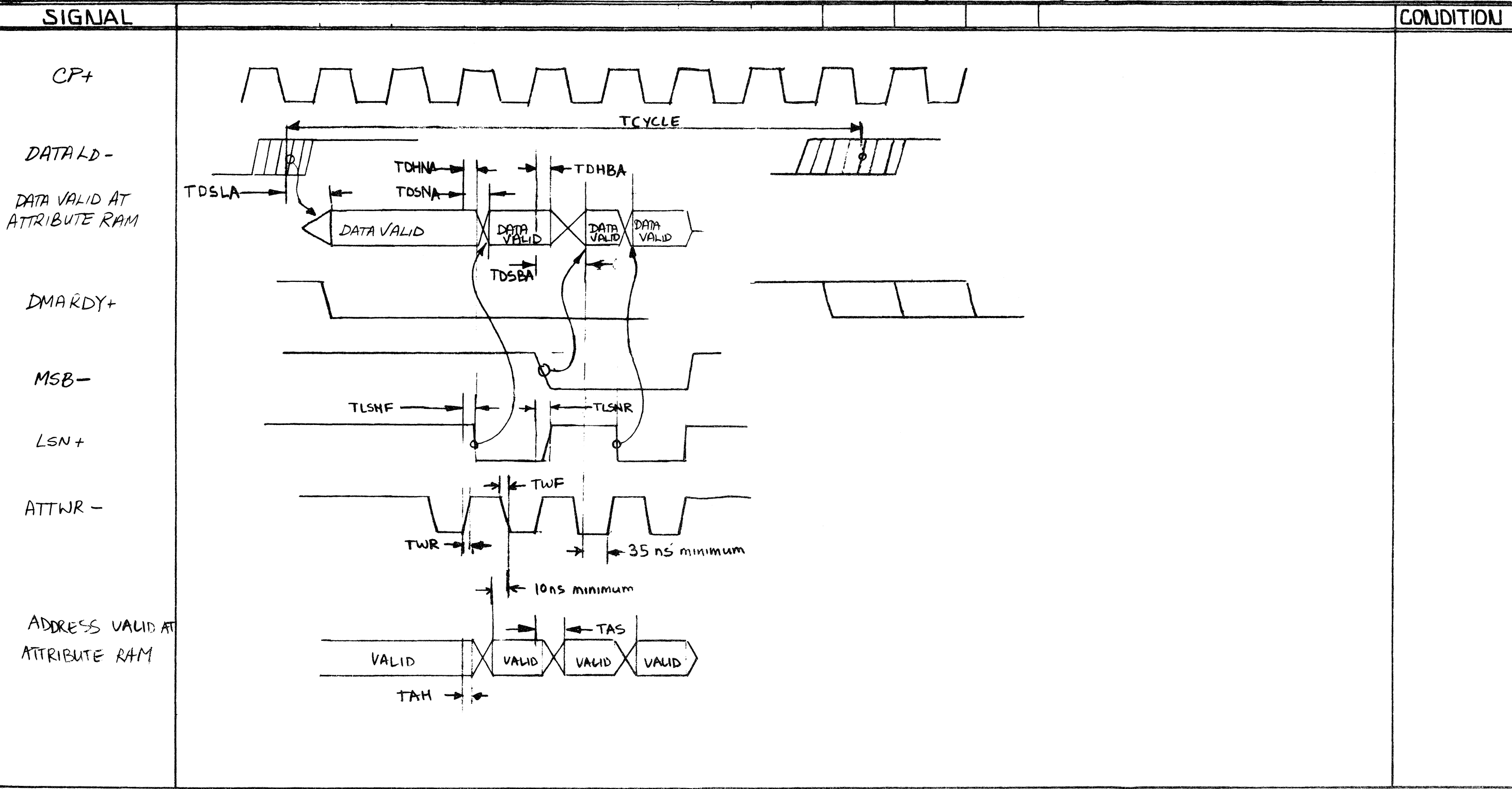


TITLE 11561 -		APPROVALS	DATE	DWG NO	
ROW BUFFER DMA		DRAWN			
APPLICATION		CHECKED		SHEET	
NEXT ASSY	USED ON	ENG		REV	





TITLE VIDEO - ATTRIBUTE DMA -		APPROVALS	DATE	DWG NO	
		DRAWN			
APPLICATION		CHECKED		SHEET	
NEXT ASSY	USED ON	ENG		REV	





TITLE VIDEO
AUX WRITE

APPROVALS
DRAWN
CHECKED

DATE

DWG NO

APPLICATION

SHEET

REV

NEXT ASSY

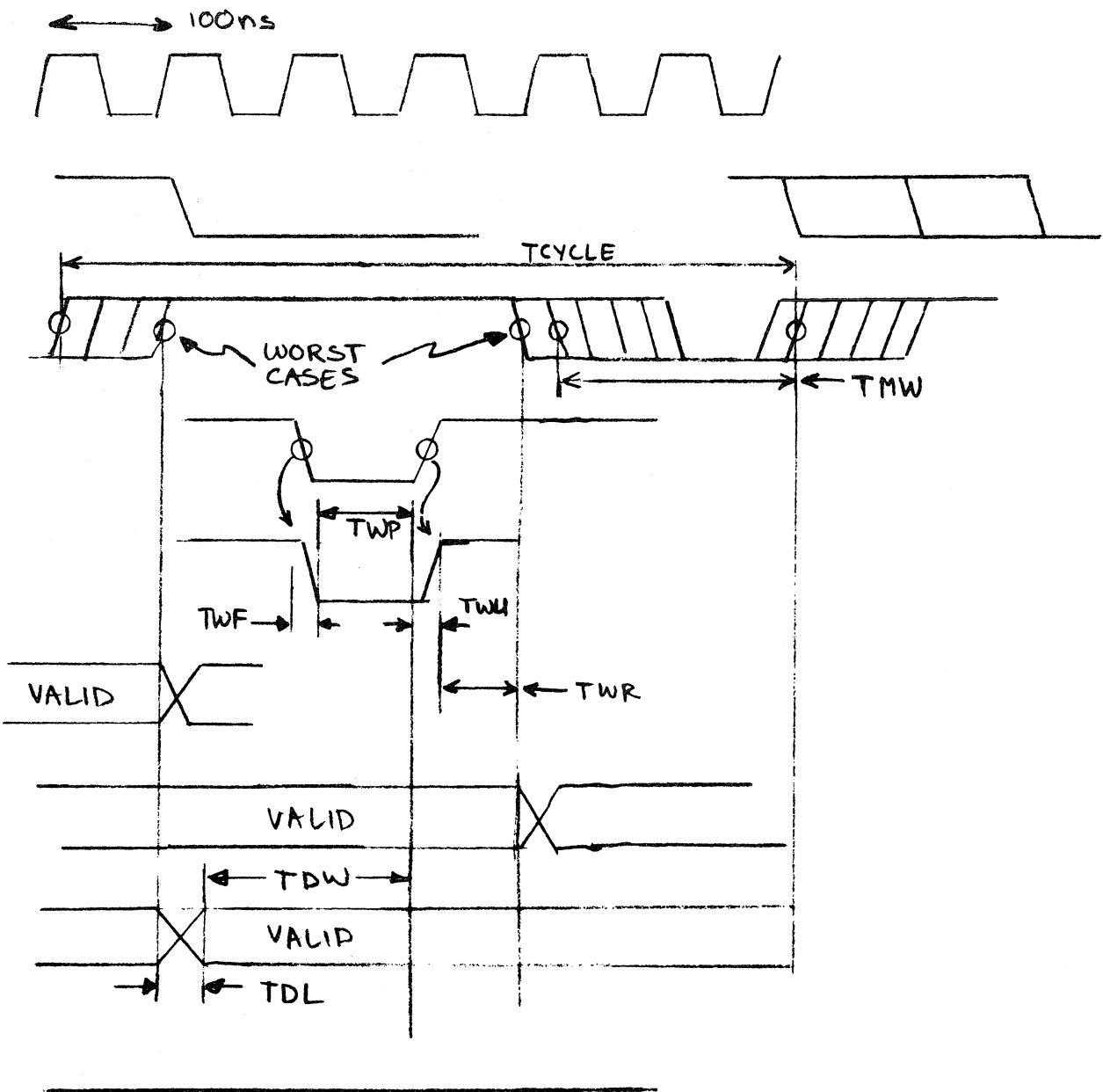
USED ON

ENG

SIGNAL

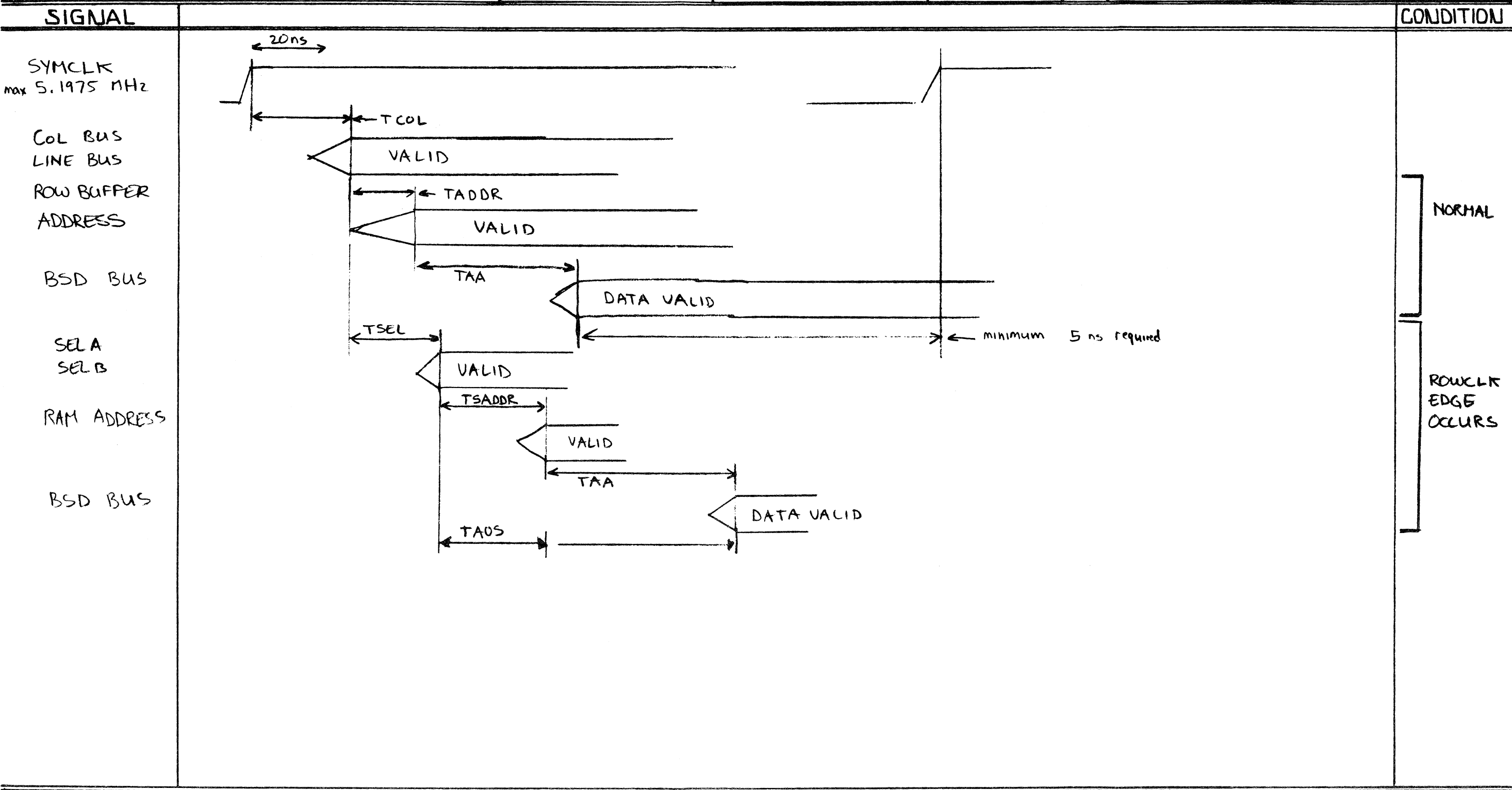
CONDITION

CP+
10 MHz
DMARDY+
DATA LD-
AUX WR-
WE pin
2141-2 RAMs
LA BUS
SA BUS
D BUS
LSB-





TITLE VIDEO		B2,B4 → D11	APPROVALS	DATE	DWG NO
1st PIPELINE DELAY			DRAWN		
APPLICATION			CHECKED		
NEXT ASSY	USED ON		ENG		
				SHEET	REV



AC-DC CHARACTERISTICS 2nd PIPELINE DELAY

[illegible]

AC-DC CHARACTERISTICS

AUX DMA READ

[illegible]

AC-DC CHARACTERISTICS ROW BUFFER DMA

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNIT
TDSL, Setup LSB Data	F12			34	ns
TDHL Hold LSB Data	B14 '504 F12 → Z		24		ns
TD SM	B14 F11			53	ns
TDHM	B14 F11 → Z		18		ns
TWF	B13		6.5	10	ns
TWR	B13		5	8	ns
TAH	C3 C4		13		ns
TAS	C3 C5 C4 D5			45	ns
TINCR	B14 C11 D14			44	ns
TCYCLE	No Set wait, 5MHz DMA		6000		ns
CONSTRAINTS FOR	93422C RAM:				
100 - TD SM	Data Set 35ns before Write Enable ↑	47			ns
200 - TDSL		166			ns
50 - TWF	Write Pulse 30ns wide	40			ns
50 + TWF - TAS	Address Set 10ns before Write enable ↓	11.5			ns
TAH - TWR	Hold Address 5ns after write enable ↑	5			ns
TDHL - TWR	Hold Data 5ns	16			ns
TDHM - TWR	after write enable ↑	10			ns

AC-DC CHARACTERISTICS

ATTRIBUTE DMA

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNIT
TDSL A	Set up Load F12 D10			41	ns
TDSNA	Set up Nibble change B14 D10			32	ns
TDHNA	Hold Nibble Change D10 B14 H7		20		ns
TDSBA	Set up Byte change B14 F11 D10			60	ns
TDHBA	Hold Byte Change B14 D10		14		ns
TCYCLE	Set wait Cycle		800		
TL SNF	Fall B14		11	17	ns
TL SNR	Rise B14		8	15	ns
TWF	Fall B13		6.5	10	ns
TWR	Rise B13		5	8	ns
TAH	Address Hold C3 C4		13		ns
TAS	Address Set C3 C5 C4 D5			45	ns
CONSTRAINTS FOR	93422C RAM:				
100-TDSNA	Data set 35 ns before write enable ↑	59			ns
100-TDSBA		40			ns
50-TWF	write pulse 30ns wide	40			ns
50+TWF-TAS	Address set 10ns before write enable ↓	11.5			ns
TAH-TWR	Hold address 5ns after write enable ↑	5			ns
TDHNA-TWR	Hold data 5ns	12			ns
TDHBA-TWR	after write enable ↑	6			ns

AUX WRITE

[illegible]

AC-DC CHARACTERISTICS 1ST PIPELINE DELAY

[illegible]

DC CHARACTERISTICS

REFERENCES

Edge Connector Pin List

Edge-Connector J6

<u>Pin</u>	<u>Signal</u>
1	+5V
2	+5V
3	+5V
4	+5V
5	-12V
6	-12V
7	GND
8	
9	MEMCLK+
10	MR-
11	EOP-
12	GACCS-
13	RESET-
14	IOW-
15	
16	VIDRQ+
17	VIDACK-
18	
19	
20	
21	
22	
23	
24	
25	
26	MEMF+
27	MEME+
28	MEMD+
29	MEMC+
30	MEMB+
31	MEMA+
32	MEM9+
33	MEM8+
34	MEM7+
35	
36	MEM6+
37	MEM5+
38	MEM4+
39	MEM3+
40	MEM2+
41	MEM1+
42	MEM0+
43	
44	WAIT-
45	
46	IOR-
48	LAF+

<u>Pin</u>	<u>Signal</u>
49	LAE+
50	LAD+
51	LAC+
52	LAB+
53	LAA+
54	LA9+
55	
56	LA8+
57	LA7+
58	LA6+
59	LA5+
60	LA4+
61	LA3+
62	LA2+
63	LA1+
64	LA0+
65	
66	
67	+12V
68	+12V
69	+12V
70	+12V
71	
72	
73	
74	
75	not used
76	not used
77	+5V
78	+5V
79	+5V
80	+5V

Video-Connector P3

<u>Pin</u>	<u>Signal</u>
1	EVIDEOA+
2	GND
3	EVIDEOA-
4	EVIDEOB-
5	GND
6	EVIDEOB+
7	NC
8	NC
9	GND
10	GND
11	LHORDR+
12	LVERDR+
13	LHORDR-
14	LVERDR-
15	GND

<u>Pin</u>	<u>Signal</u>
16	CRT OK
17	
18	GND
19	
20	

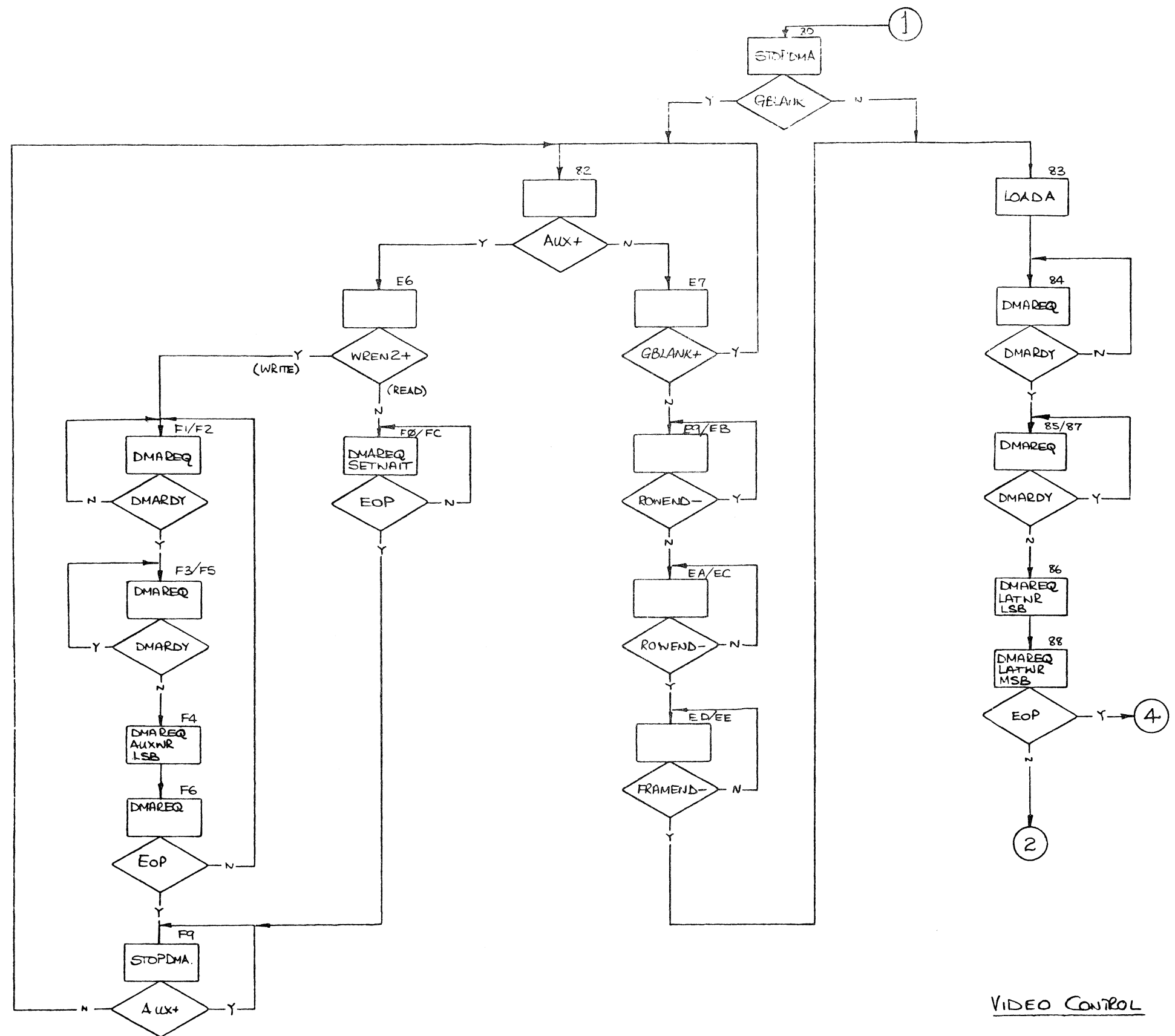
Jumper

<u>Option</u>	<u>D-E</u>
60Hz	open
50Hz	closed

Test points

I/O Port Address

The screen attribute register is port 58.



VIDEO CONTROL

FLOW CHART

LINE ATTRIBUTE DMA &

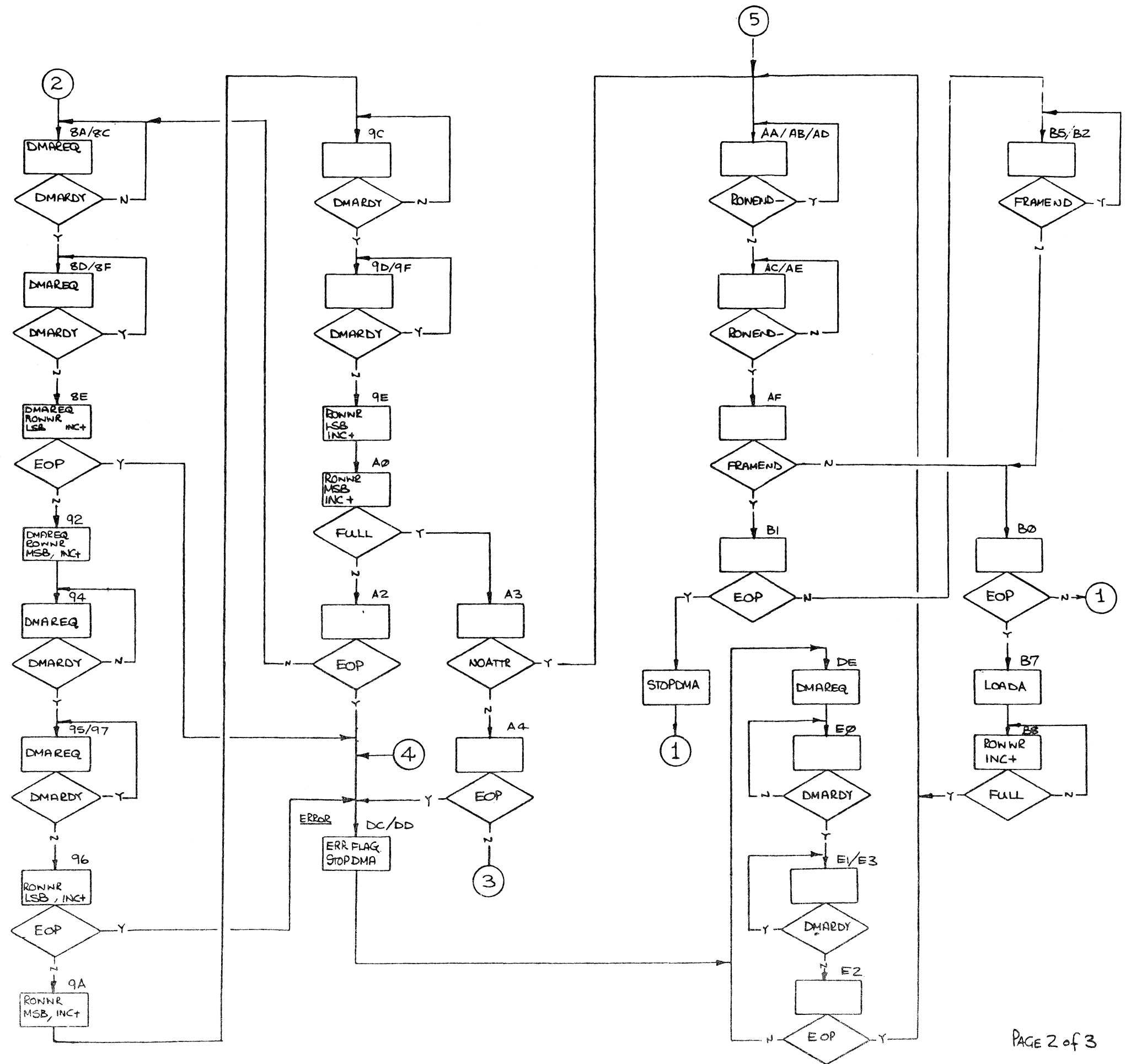
AUXILIARY FUNCTIONS

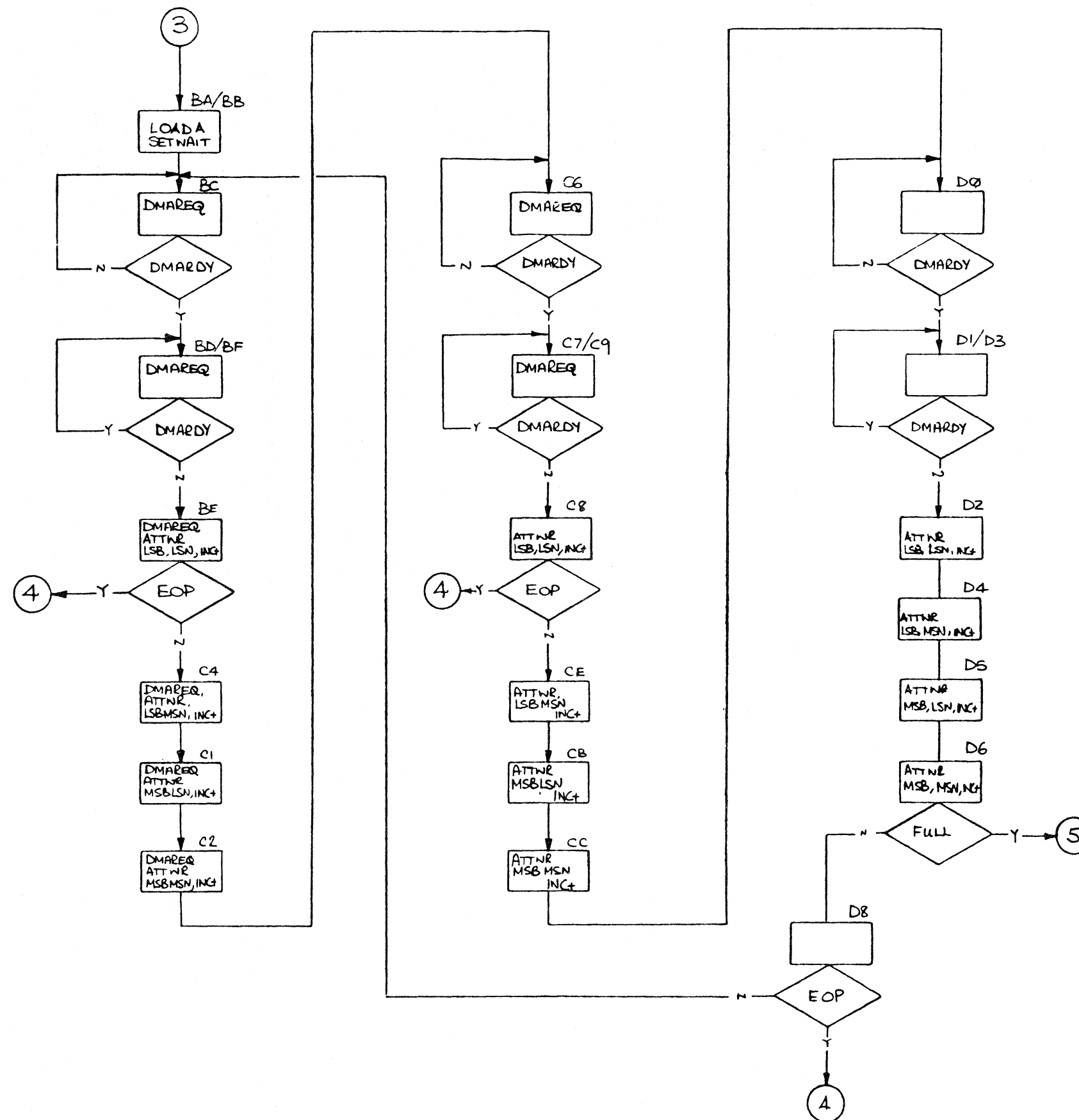
VIDEO CONTROL

FLOW-CHART

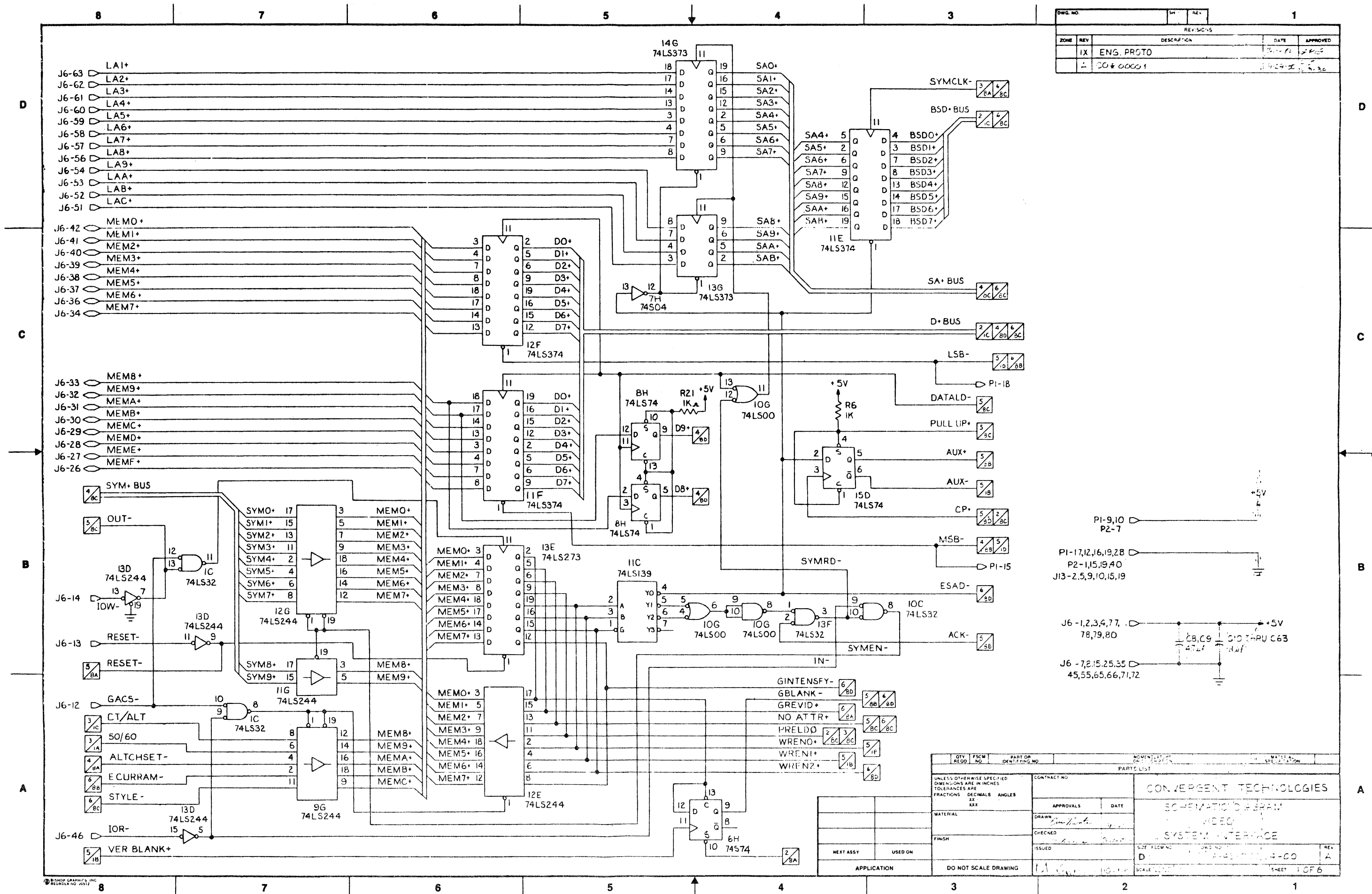
ROW-BUFFER DMA &

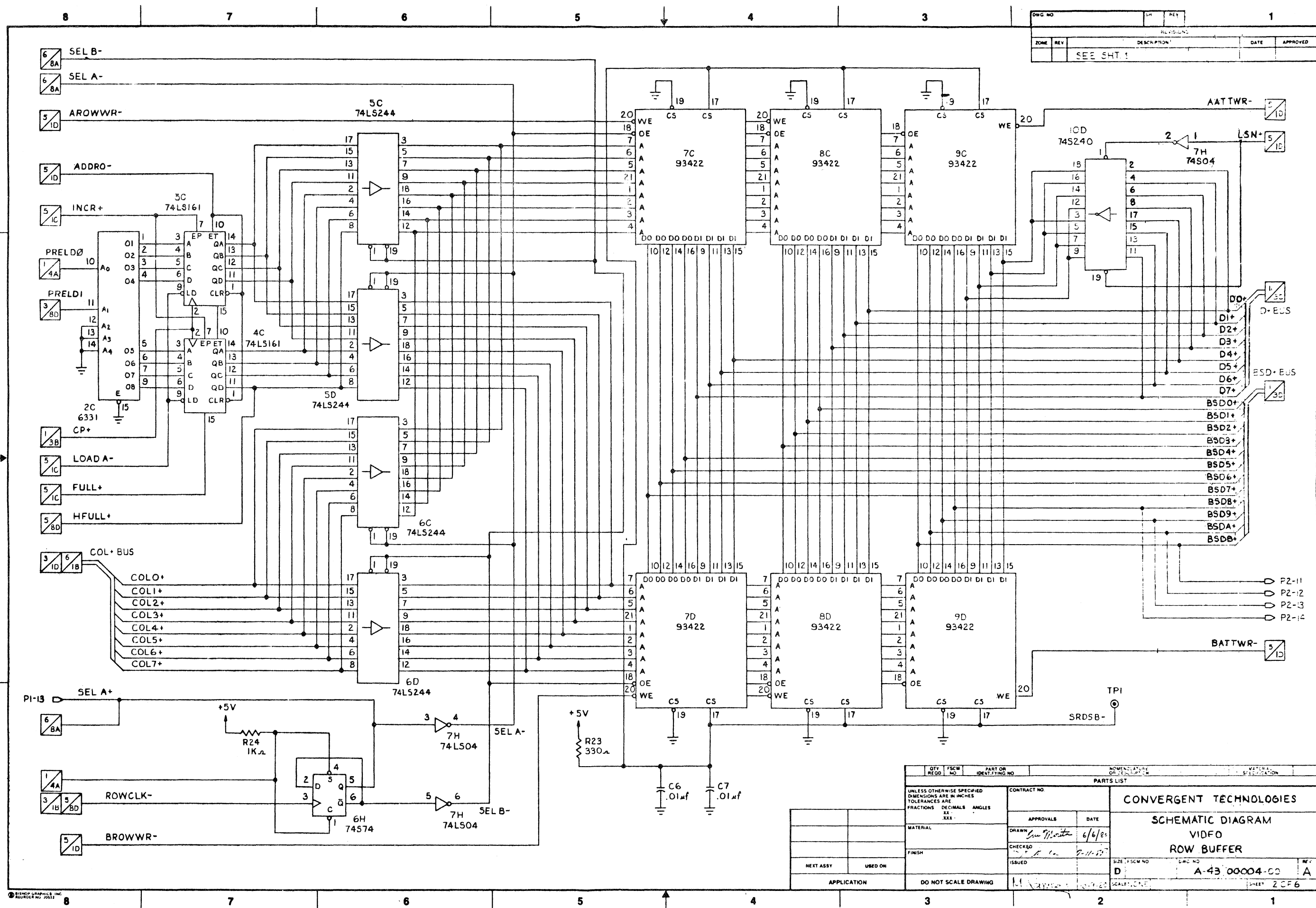
ERROR CONDITIONS





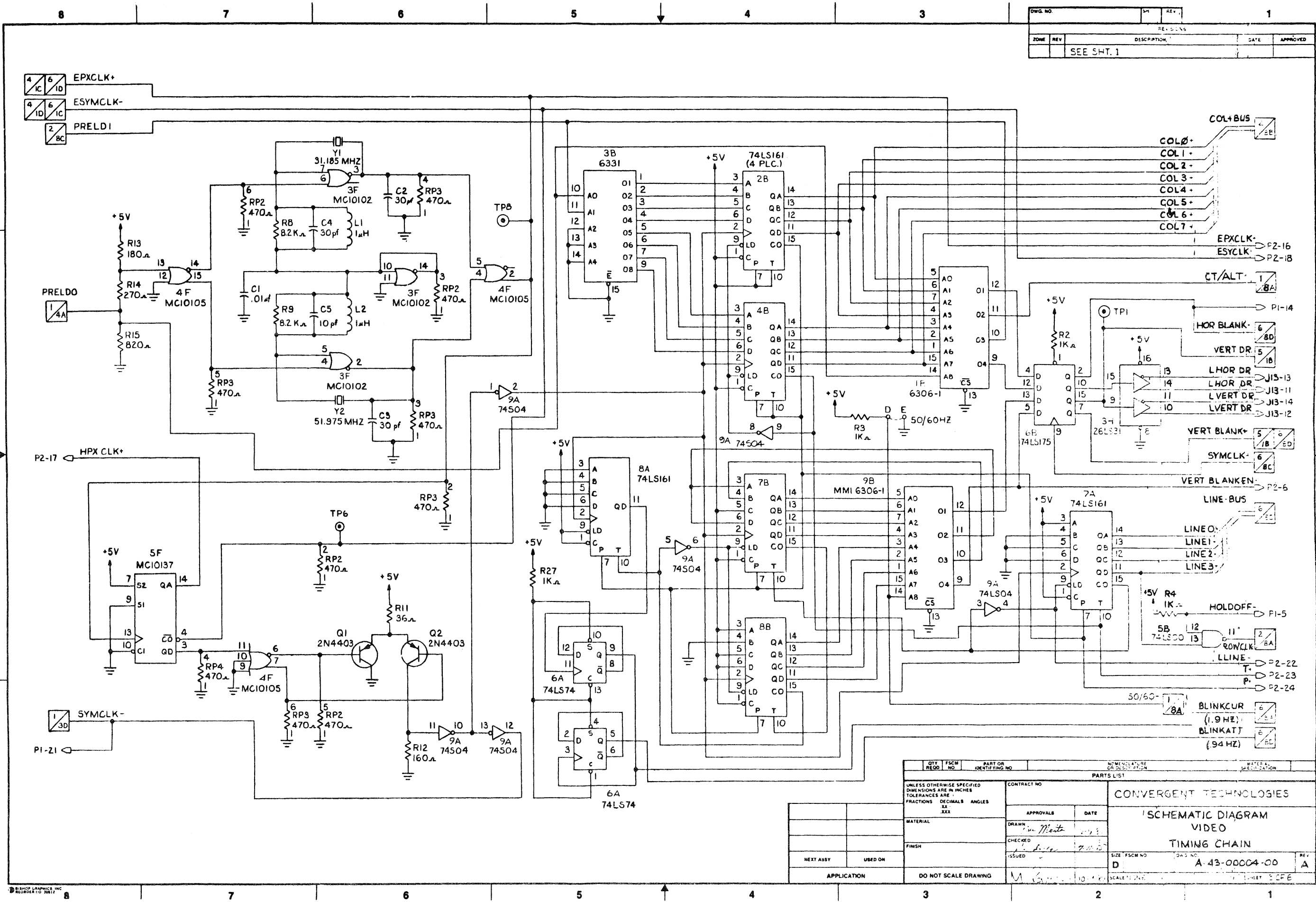
VIDEO CONTROL
FLOW CHART
ATTRIBUTE DMA





DWG NO		REV	DESCRIPTION	DATE	APPROVED
SEE SHT. 1					

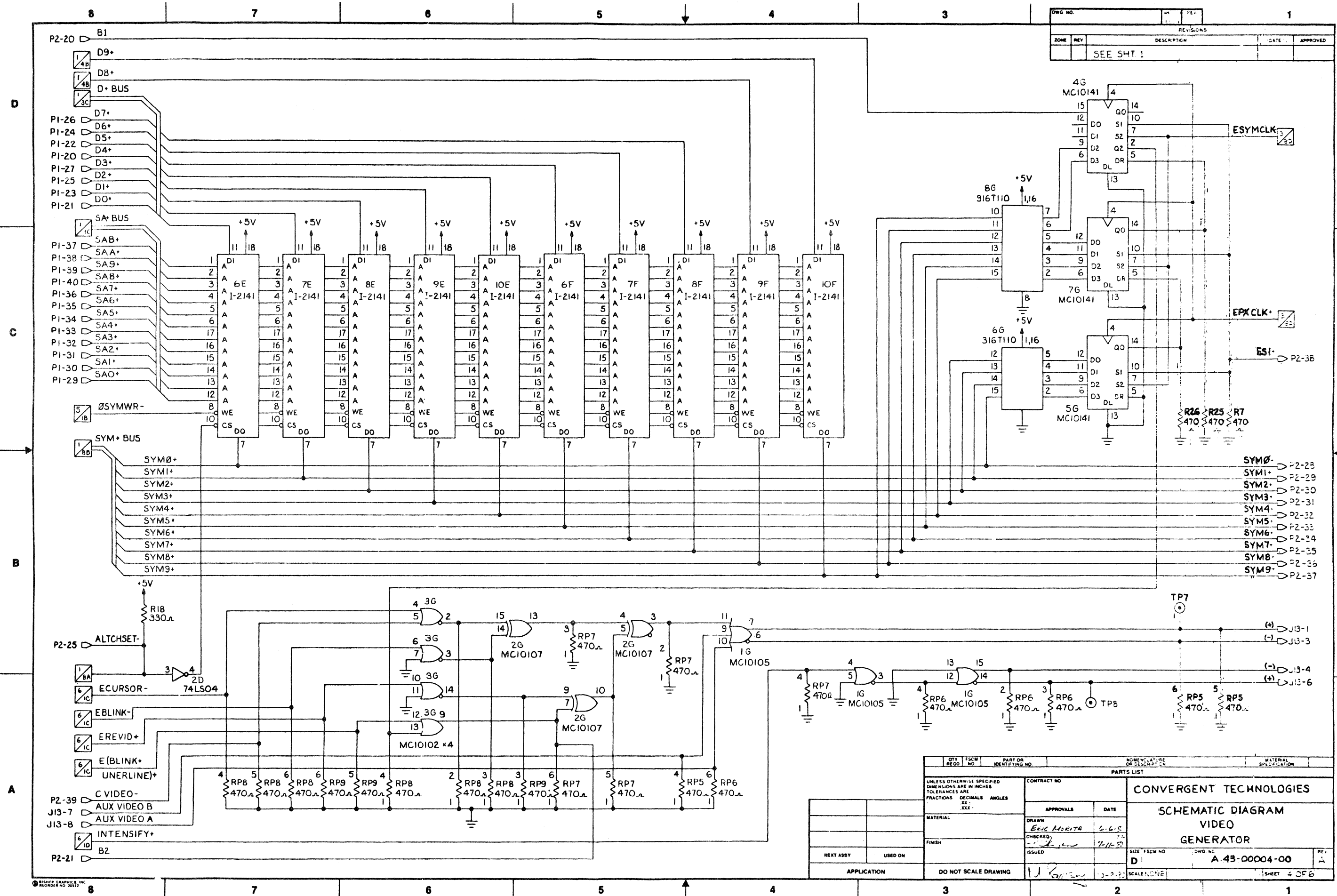
QTY REQD		FSCM NO	PART OR IDENTIFYING NO	NOMENCLATURE OR DESCRIPTION		MATERIAL SPECIFICATION	
PARTS LIST							
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES TOLERANCES ARE FRACTIONS DECIMALS ANGLES XX XXX				CONTRACT NO.			
MATERIAL				APPROVALS	DATE	CONVERGENT TECHNOLOGIES	
FINISH				DRAWN <i>Jim Winters</i>	6/6/82	SCHEMATIC DIAGRAM	
				CHECKED <i>Jim Winters</i>	7-11-82	VIDEO	
				ISSUED		ROW BUFFER	
DO NOT SCALE DRAWING				SCALE: NONE	SIZE: FSCM NO	DWG NO	REV
					D	A-43 00004-00	A
						SHEET	2 OF 6



REV	DATE	APPROVED
1		

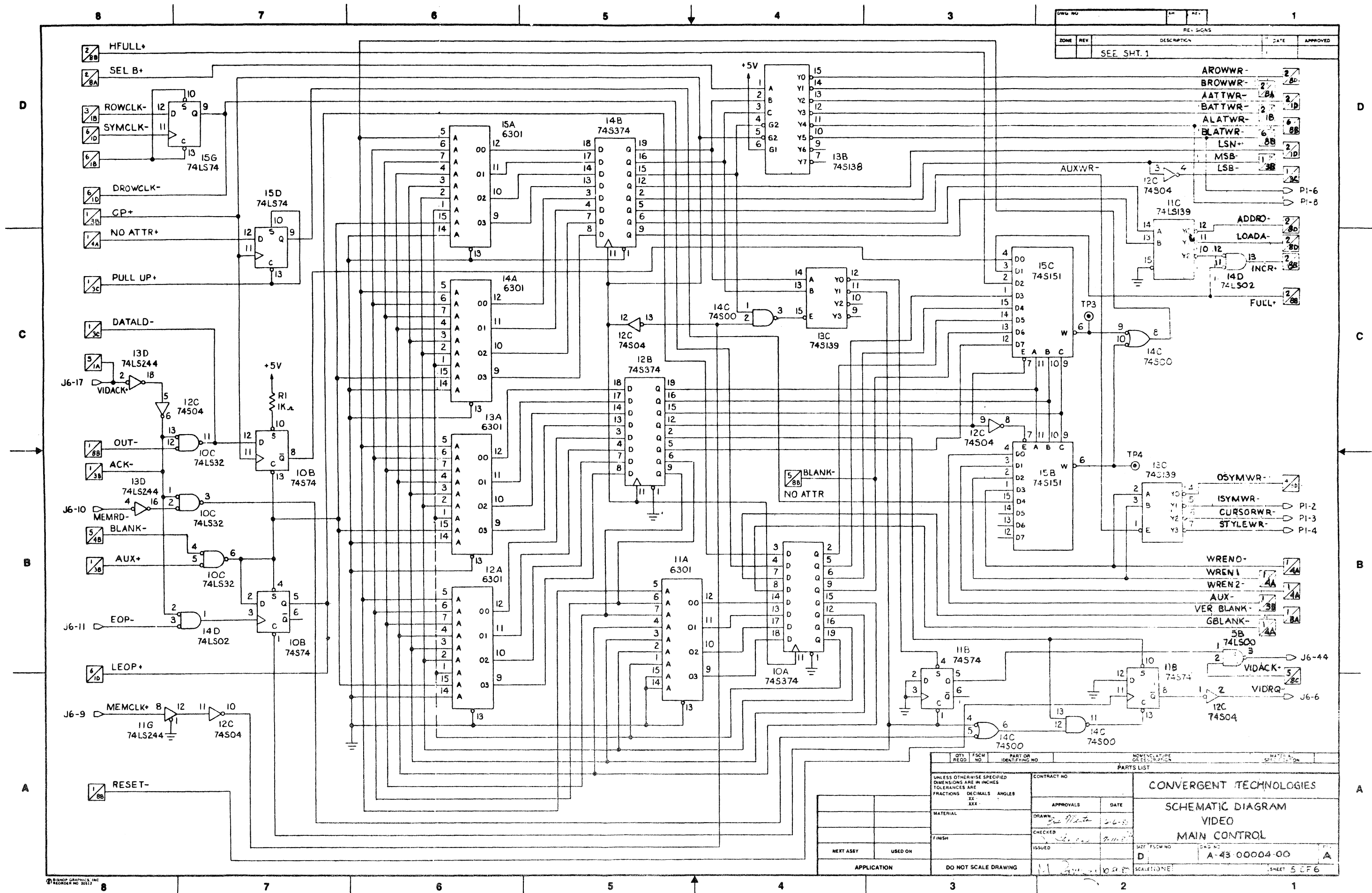
SEE SHT. 1

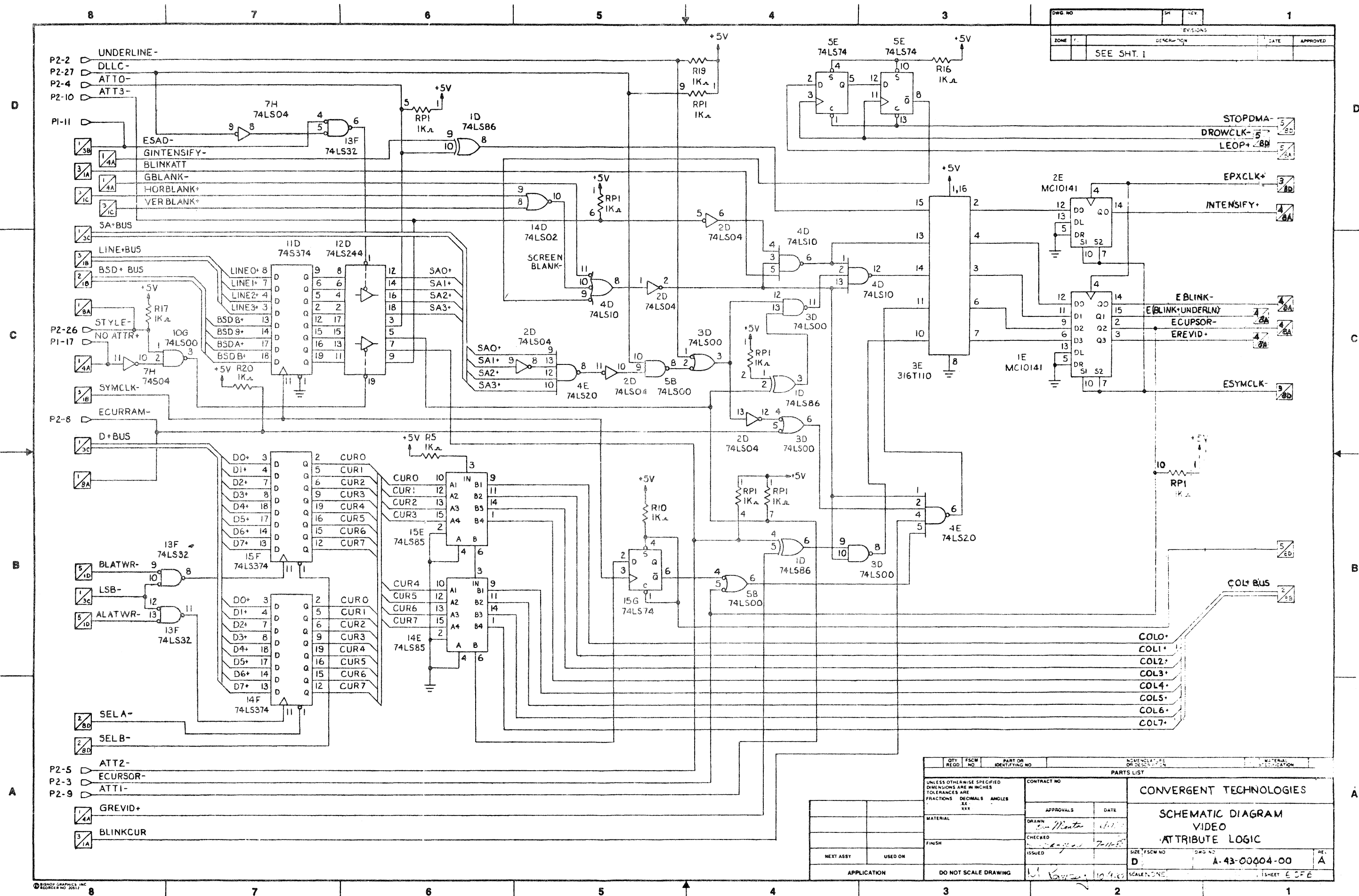
QTY REQD	FSCM NO	PART OF IDENTIFYING NO	NOMENCLATURE OR DESCRIPTION		MATERIAL SPECIFICATION
PARTS LIST					
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES TOLERANCES ARE FRACTIONS DECIMALS ANGLES XX XXX			CONTRACT NO		
MATERIAL			APPROVALS	DATE	CONVERGENT TECHNOLOGIES SCHEMATIC DIAGRAM VIDEO TIMING CHAIN
			DRAWN <i>in Monte</i>	<i>10/1/80</i>	
			CHECKED		
FINISH			ISSUED <i>10/1/80</i>		
DO NOT SCALE DRAWING			SIZE FSCM NO		REV
<i>M. G. 10/1/80</i>			D A-43-00004-00		A
			SCALE 1/2" = 1"		SHEET 1 OF 6

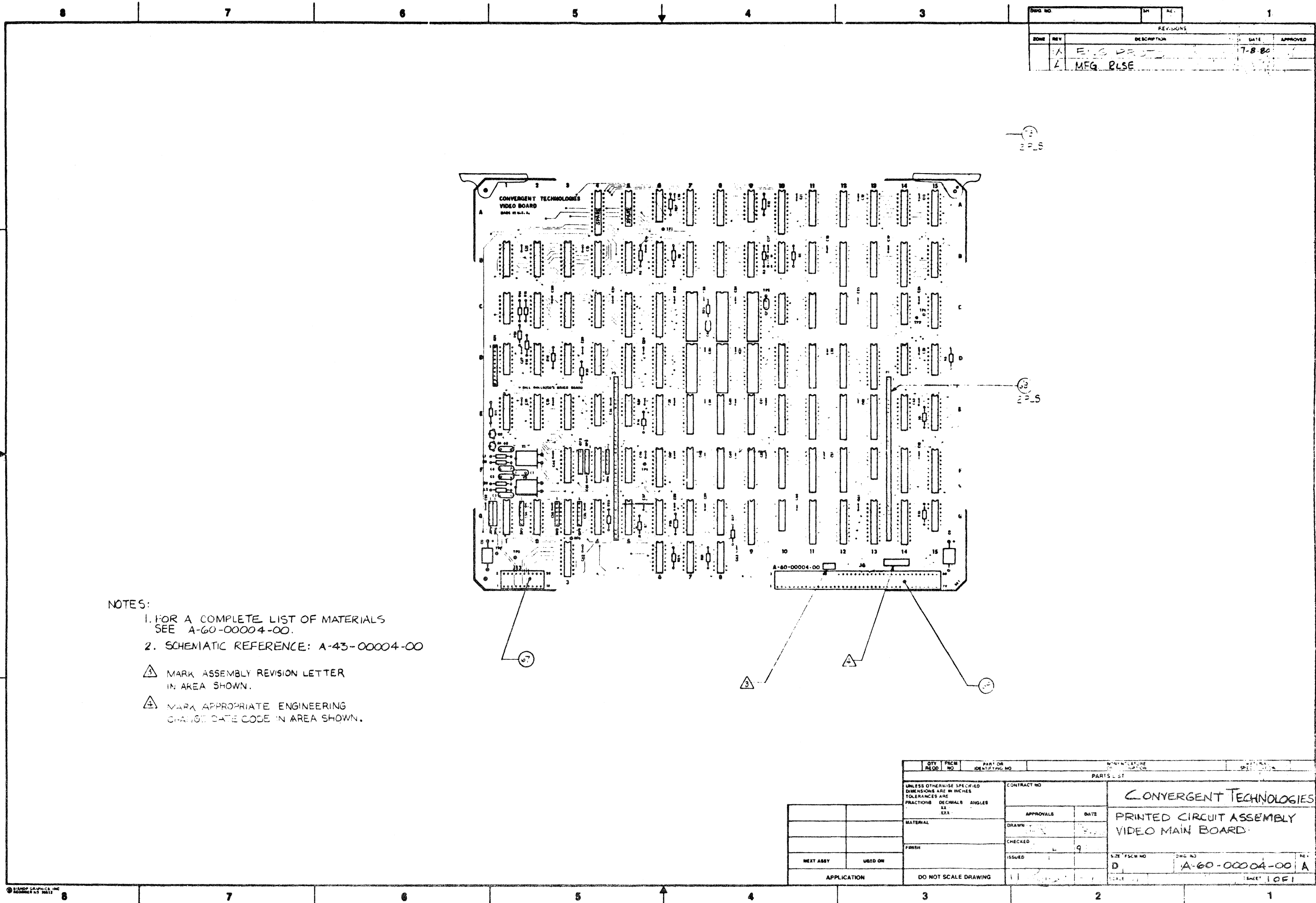


DWG NO		REV	DATE	APPROVED
SEE SHT. 1				

QTY REQD		FSCM NO		PART IDENTIFYING NO		NOMENCLATURE OR DESCRIPTION		MATERIAL SPECIFICATION					
PARTS LIST													
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES TOLERANCES ARE FRACTIONS DECIMALS ANGLES .XX .XXX .XXX				CONTRACT NO		CONVERGENT TECHNOLOGIES SCHEMATIC DIAGRAM VIDEO GENERATOR							
				APPROVALS						DATE			
MATERIAL				DRAWN ERIC MORITA		6-6-8		SIZE FSCM NO D 1		DWG NO A-43-00004-00		REV A	
FINISH				CHECKED [Signature]		7-11-8							
				ISSUED									
DO NOT SCALE DRAWING				11 [Signature]		10-2-8		SCALE NONE		SHEET 4 OF 6			







REV. NO.		REV.	DESCRIPTION	DATE	APPROVED
1	A	ENG. DESIGNED		7-8-80	
2	A	MFG. PLSE			

- NOTES:
1. FOR A COMPLETE LIST OF MATERIALS SEE A-60-00004-00.
 2. SCHEMATIC REFERENCE: A-43-00004-00
- ⚠ MARK ASSEMBLY REVISION LETTER IN AREA SHOWN.
- ⚠ MARK APPROPRIATE ENGINEERING CHANGE DATE CODE IN AREA SHOWN.

QTY REQD	FROM NO.	PART OR IDENTIFYING NO.	DATE REVISION	SHEET NO.
PARTS LIST				
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES TOLERANCES ARE FRACTIONS DECIMALS ANGLES XX.X			CONTRACT NO.	
MATERIAL			APPROVALS	DATE
FINISH			CHECKED	9
NEXT ASSY			ISSUED	
APPLICATION			DO NOT SCALE DRAWING	
CONVERGENT TECHNOLOGIES			PRINTED CIRCUIT ASSEMBLY VIDEO MAIN BOARD	
SCALE			SIZE	SCALE NO.
A-60-00004-00			NO.	A
SHEET 1 OF 1				

9. KEYBOARD

FUNCTIONAL DESCRIPTION AND SOFTWARE INTERFACE

The keyboard is housed in a separate enclosure which contains the keys, a firmware-programmed 8048 microcomputer with related logic and eight LED's. The 8048 scans the keyboard, controls the LED's, and functions as an asynchronous serial port. The mainframe communicates with the keyboard via a TTL level bidirectional serial link.

Figure 9-1, List of Keys, and Figure 9-2, Map of Keyboard, show the correlation between the eight-bit codes sent by the 8048 to the mainframe and the 102 keys physically present in the keyboard.

When any key is depressed or released, the keyboard sends an eight bit byte for each of the keys which are currently depressed. Mainframe software translates these eight bit codes into ASCII characters and special codes for CT-defined functions. Several keys can be depressed simultaneously to make up a single command. The 8048 in the keyboard sets the MSB only when sending the last of the 8-bit codes corresponding to keys currently depressed.

Figure 9-3 illustrates the format in which keyboard output is sent by the 8048 to the 8251A. If no keys are depressed a null code, C0 (192), is sent.

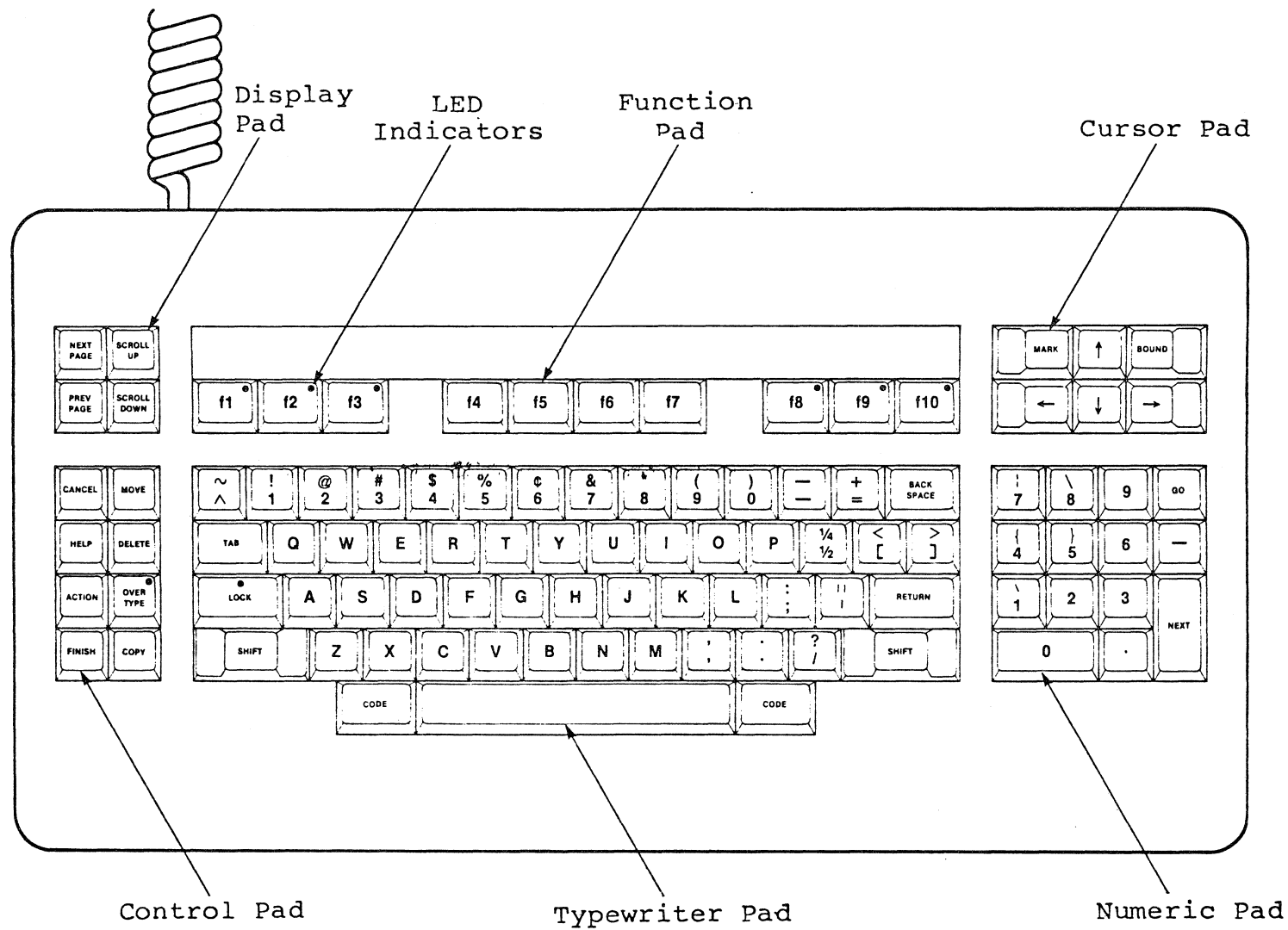
Writing to the Keyboard

The mainframe can send the following commands to the keyboard:

<u>Keyboard Code</u>	<u>Keystation</u>	<u>Keyboard Code</u>	<u>Keystation</u>
00	HELP	27	' (single quote)
01	up arrow	28	unused code
02	MARK	29	unused code
03	BOUND	2A	unused code
04	FINISH	2B	=
05	PREV PAGE	2C	, (comma)
06	1/2	2D	- (hyphen)
07	CANCEL	2F	/
08	BACKSPACE	30...39	0...9
09	TAB	3A	unused code
0A	RETURN	3B	;
0B	down arrow	3C	unused code
0C	NEXT PAGE	3D	unused code
0D	NEXT	3E	unused code
0E	left arrow	3F	unused code
0F	right arrow	40	indicates the last key released; al- ways has high bit on (that is, 0C0h)
10	(SH-L')	41	num 6
11	SCROLL UP	42	num -
12	MOVE	43	ACTION
13	SCROLL DOWN	44	OVERTYPE
14	COPY	45	LOCK
15	F1	46	num 2
16	F2	47	num 3
17	F3	48	left SHIFT
18	F4	49	right SHIFT
19	F5	4A	num 0
1A	F6	4B	num
1B	G0	4C	left CODE
1C	F7	4D	right CODE
1D	F8	4E...5A	unused code
1E	F9	5B	
1F	F10	5C	num 7
20	space	5D	
21	num 9	5E	(carat)
22	(SH-R')	5F	unused code
23	(0 ')	60	num 1
24	(NEXT')	61...7A	a...z
25	unused code	7B	num 4
26	unused code	7C	num 8
		7D	num 5
		7E	unused code
		7F	DELETE

Figure 9-1. List of Keys

Figure 9-2. Map of Keyboard



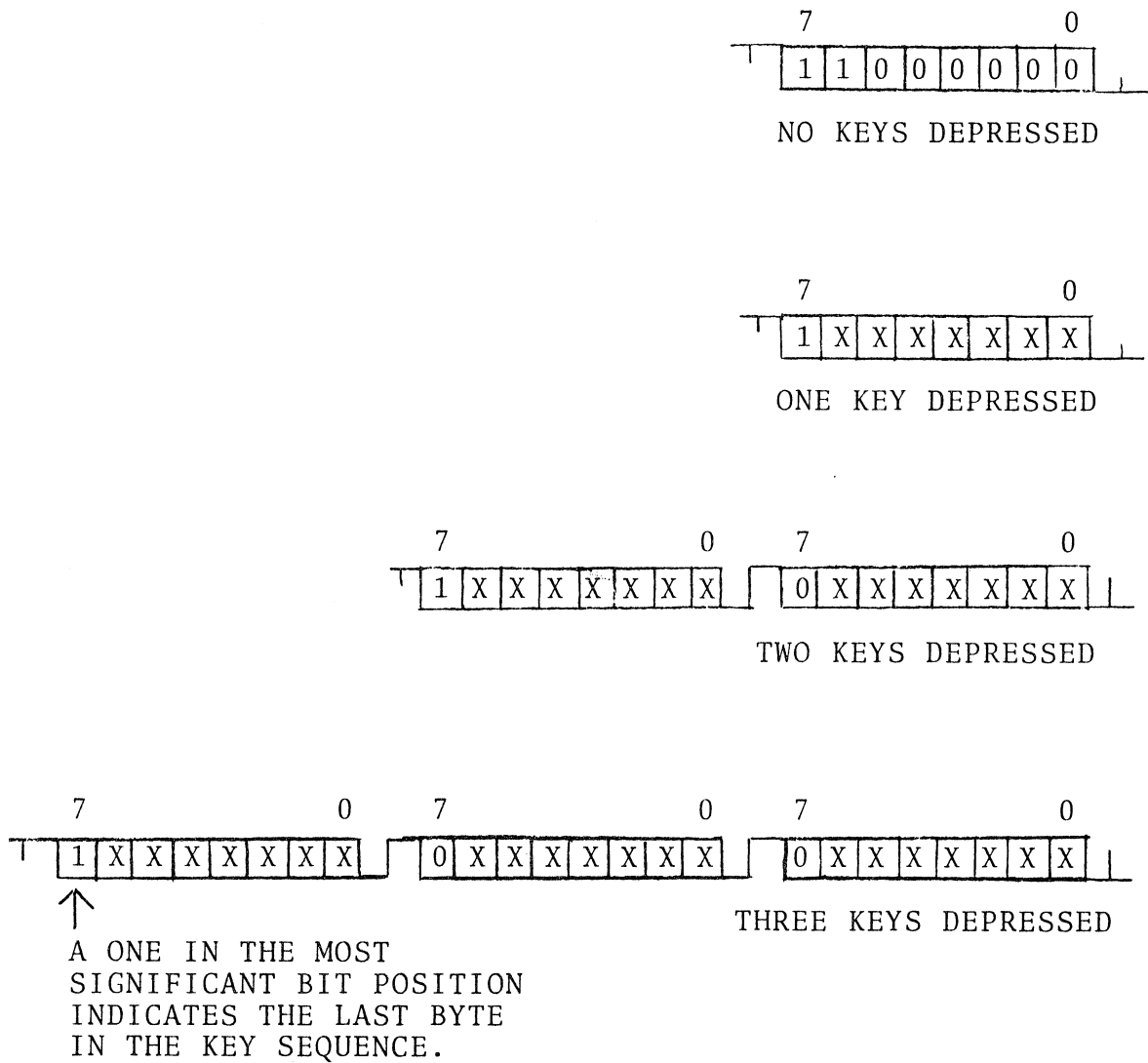


Figure 9-3. Data Format for Keyboard Output

<u>COMMAND</u>	<u>HEX CODE</u>
Reset	92 (146)
CheckROM	8C (140)
Echo	9E (158)
LED03	AX (160+X)
LED47	BX (176+X)

"Reset" cancels the Echo mode (if enabled) turns off all the LEDs, waits for any bytes in the keyboard buffer to be sent out and then initiates a new scan.

"CheckROM" computes a checksum of the 8048 ROM and sends it to the processor.

"Echo" causes the 8048 to send back the subsequent 256 input bytes verbatim unless one of those bytes is 92 (146) or 80 (128), which are interpreted as Reset or CheckROM commands.

"LED03" and "LED47" write the bit pattern specified by 'X' to LEDs 0-3 and 4-7 respectively. 'X' is the bottom nibble of the command byte.

OPERATION

The 8048 scans the keyboard by generating a series of row address (pins 21-24) and monitoring the state of the column lines (pins 27-34). Chip IC6 demultiplexes the row addresses and drives the corresponding row lines in the keyboard array.

Crystal Y36 and the number of program steps involved in the 8048's UART operations determine the baud rate, which is 1221 baud, the same as that of the 8251A USART in the mainframe.

DATA IN is buffered by a Schmitt trigger inverter, whose hysteresis helps ensure good noise immunity. DATA OUT comes from an open-collector TTL gate, which has an external 1K pullup resistor.

DC CHARACTERISTICS

AC-DC CHARACTERISTICS

[illegible]

REFERENCES

Connector Pin List

Assignment of pins in the D-type AMP connector is as follows:

<u>Pin</u>	<u>Function</u>
1	Data out
2, 4, 8, 9	Ground
6, 7	+5 volts
5	Reset
3	Data in

LED Indicators

LED indicators are built into eight of the keys, as follows:

<u>LED</u>	<u>Station</u>	<u>Key</u>
LED0	17	f10
LED1	16	f9
LED2	15	f8
LED3	10	f3
LED4	9	f2
LED5	8	f1
LED6	63	LOCK
LED7	62	OVERTYPE

Test Points

Eight pins of the 8048 microprocessor are brought out to test points:

<u>Point</u>	<u>8048 pin</u>	<u>Function</u>
TP1	19	DB7
TP2	18	DB6
TP3	17	DB5
TP4	16	DB4
TP5	15	DB3
TP6	14	DB2
TP7	13	DB1
TP8	12	DB0

LEFT BLANK INTENTIONALLY

Figure 9-5. Keyboard Assembly Drawing

10. VIDEO MONITOR (CRT)

FUNCTIONAL DESCRIPTION

The video monitor contains a 15" CRT with 110-degree deflection. Maximum picture size is 10.4" x 7.75". Circuitry within the video monitor transforms four digital signals from the video control board into drive signals for the CRT. Deflection is by raster scan at 50- or 60 Hz, non-interlaced.

OPERATION BY CIRCUIT BLOCK

Figure 10-1 shows a block diagram of the video monitor. The following sections describe each of the blocks outlined in the diagram.

Horizontal Deflection

The horizontal deflection circuitry, in conjunction with the horizontal section of the deflection yoke, is responsible for generating the horizontal (or fast) portion of the raster scan. The horizontal scanning rate is 31.5 KHz.

The horizontal drive section is responsible for generating a linear current waveform in the yoke that repeats at the horizontal rate. The peak amplitude of the current is adjusted to cause the CRT electron beam to traverse the full width of the screen (typically 4A peak).

The input signal to the horizontal drive is generated in the horizontal phase locked loop. This circuit is responsible for the phase-synchronization of the input drive signal, LHORDR (from the video control board) with the retrace signal from

the horizontal deflection inductor. Such a synchronization ensures consistent centering of the CRT raster over a wide range of component tolerances and environmental extremes.

Horizontal Regulator

This circuit controls the amplitude of the horizontal deflection signal, which corresponds to the width of the raster scan. A potentiometer adjustment controls this width.

Vertical Deflection

The vertical deflection circuitry, in conjunction with the vertical section of the deflection yoke, is responsible for the vertical (or slow) portion of the raster scan. The vertical deflection rate is 50 or 60 Hz, determined by the frequency of the input drive signal, LVERDR (from the video control board).

Similar to the horizontal deflection, a linear current waveform is generated, repeating at the vertical rate and with peak amplitude sufficient to cause the electron beam to scan the total height of the screen. This current waveform is generated by the vertical deflection amplifier in response to a linear voltage waveform applied to its input from the vertical ramp generator. The ramp generator produces a linear ramp voltage, synchronized to LVERDR and with independently adjustable amplitude (height) and linearity.

Bias Supply and Dynamic Focus

The CRT has three grids which are biased with respect to the cathode by the amounts given below.

<u>Grid Number</u>	<u>Name</u>	<u>voltage</u>
1	bias electrode	-70
2	brightness electrode	300-1,000
4	focus electrode	0-500

A bias supply powered from the horizontal drive circuit provides these voltages. In addition, an external brightness pot is included which alters the voltage on grid number 1.

The diagram illustrates the internal architecture of a video display system, centered around a cathode ray tube (CRT). The main components and their connections are as follows:

- Horizontal Control:**
 - L. HOR. DR.** (Left Horizontal Drive) feeds into the **HORIZ. PLL** (Phase-Locked Loop).
 - The **HORIZ. PLL** output goes to the **HORIZ. DRIVE** block.
 - A **HOLD** input, represented by a potentiometer, also feeds into the **HORIZ. PLL**.
 - The **HORIZ. DRIVE** block provides the horizontal deflection signal to the CRT and also feeds into the **HORIZ. REG.** (Horizontal Register).
- Vertical Control:**
 - VERT. SIZE** and **VERT. LIN.** (Vertical Linearity) inputs, each with a potentiometer, feed into the **RAMP GENERATOR**.
 - The **RAMP GENERATOR** is also controlled by **LVER. DR.** (Vertical Drive) and provides the vertical deflection signal to the CRT.
 - The **RAMP GENERATOR** output also feeds into the **DEFL. AMP** (Deflection Amplifier).
- Power and Biasing:**
 - A **+12V** supply feeds into a **REG.** (Regulator), which provides a **6.3v** **VHEATER** (Heater) voltage.
 - A **+32V** supply feeds into the **H.V. REG.** (High Voltage Regulator).
 - The **H.V. REG.** is controlled by an **ENABLE** signal and provides power to the **H.V. SUPPLY** (High Voltage Supply).
 - The **H.V. SUPPLY** outputs **VA(16Kv)** (16 Kilovolts) to the CRT.
 - The **BIAS. SUPPLY** (Bias Supply) block receives inputs from the **HORIZ. DRIVE** and the **H.V. SUPPLY**. It provides several control voltages: **REMOTE BRIGHTNESS**, **BRIGHTNESS** (via a potentiometer), **CRT FOCUS** (via a potentiometer), and a sum of signals ($V_{a2} + V_{a1} + \bar{V}_{a4}$) to a summing junction.
- Signal Processing:**
 - The **VIDEO CATHODE** signal from the CRT is amplified by the **VIDEO AMP** (Video Amplifier).
 - The **VIDEO AMP** output is split into two channels, **E VIDEO A** and **E VIDEO B**, each passing through a buffer or driver stage.
 - The **DEFL. AMP** output is also fed into the **VIDEO AMP**.
- Dynamic Focus and Parabola:**
 - The **HORIZ. DRIVE** signal is also fed into the **DYNAMIC FOCUS** block.
 - The **VERT. DRIVE** signal is fed into the **VERT. PARABOLA** block.
 - The **DYNAMIC FOCUS** and **VERT. PARABOLA** blocks interact to provide feedback to the **HORIZ. DRIVE** and **VERT. DRIVE** respectively.

The dynamic focus circuit is used to alter the focus voltage as a function of the position of the electron beam on the CRT face. In this way, optimum focus can be maintained resulting in maximum resolution. The dynamic focus waveform is a composite of two parabolic waveforms. One waveform repeats at the horizontal rate (HORIZ.PARABOLA) and the other repeats at the vertical rate (VER.PARABOLA). This waveform, when summed with the D.C. focus electrode bias (adjustable by the C.FOCUS control) yields a waveform that offers maximum focus correction at the corners of the CRT raster.

Video amplifier

EVIDEOA and EVIDEOB, the ECL signals for video display data, are converted to VIDEO, the signal which drives the cathode. The level-shifting amplifier is an emitter-coupled configuration with common-base output stage. The amplitude is 25 volts for normal data and 20 volts for half-bright data (controlled by EVIDEOB).

Power

Anode voltage for the CRT is generated by a 16-kV supply which operates from a 24 volt regulated power source. This power source is derived from +32V, and it is enabled only when horizontal drive is functioning.

Another regulator provides 6.3 volts for the heater and 5 volts for the video amplifier. Power for this second regulator comes from +12V.

ADJUSTMENT PROCEDURE

REFERENCES

Connector Pin List

Video Connector J14

<u>Pin</u>	<u>Signal</u>
1	-32
2	+32
3	GND
4	LHORDR(-)
5	LHORDR(+)
6	GND
7	N.C.
8	GND
9	VIDEOA(-)
10	VIDEOA(+)
11	GND
12	VIDEOB(+)
13	VIDEOB(-)
14	N.C.(PLUG)
15	GND
16	LVERDR(+)
17	LVERDR(-)
18	CRTOK
19	GND
20	+12

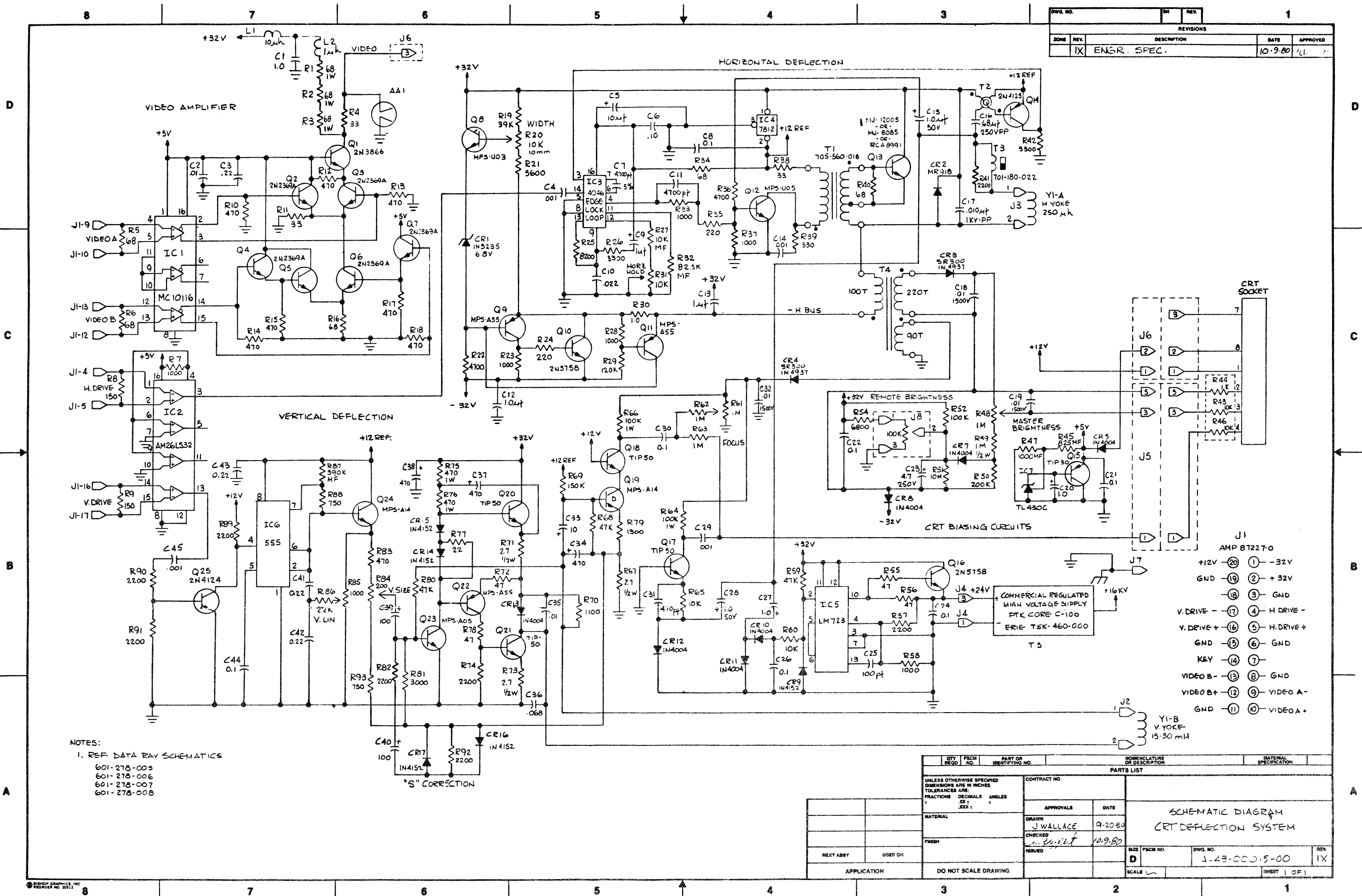
Potentiometers

<u>Name</u>	<u>Part #</u>	<u>Location and Function</u>
BRIGHTNESS		An externally accessible thumbwheel on bottom of display used for brightness control.
VERT.SIZE		Adjusts the height of the raster.
VERT.LIN		Adjusts the vertical linearity, that is, the separation between horizontal lines.
CTR.FOCUS		Adjusts the focus at the center of the screen.
WIDTH		Adjusts the width of the raster.
G2		Adjusts brightness and

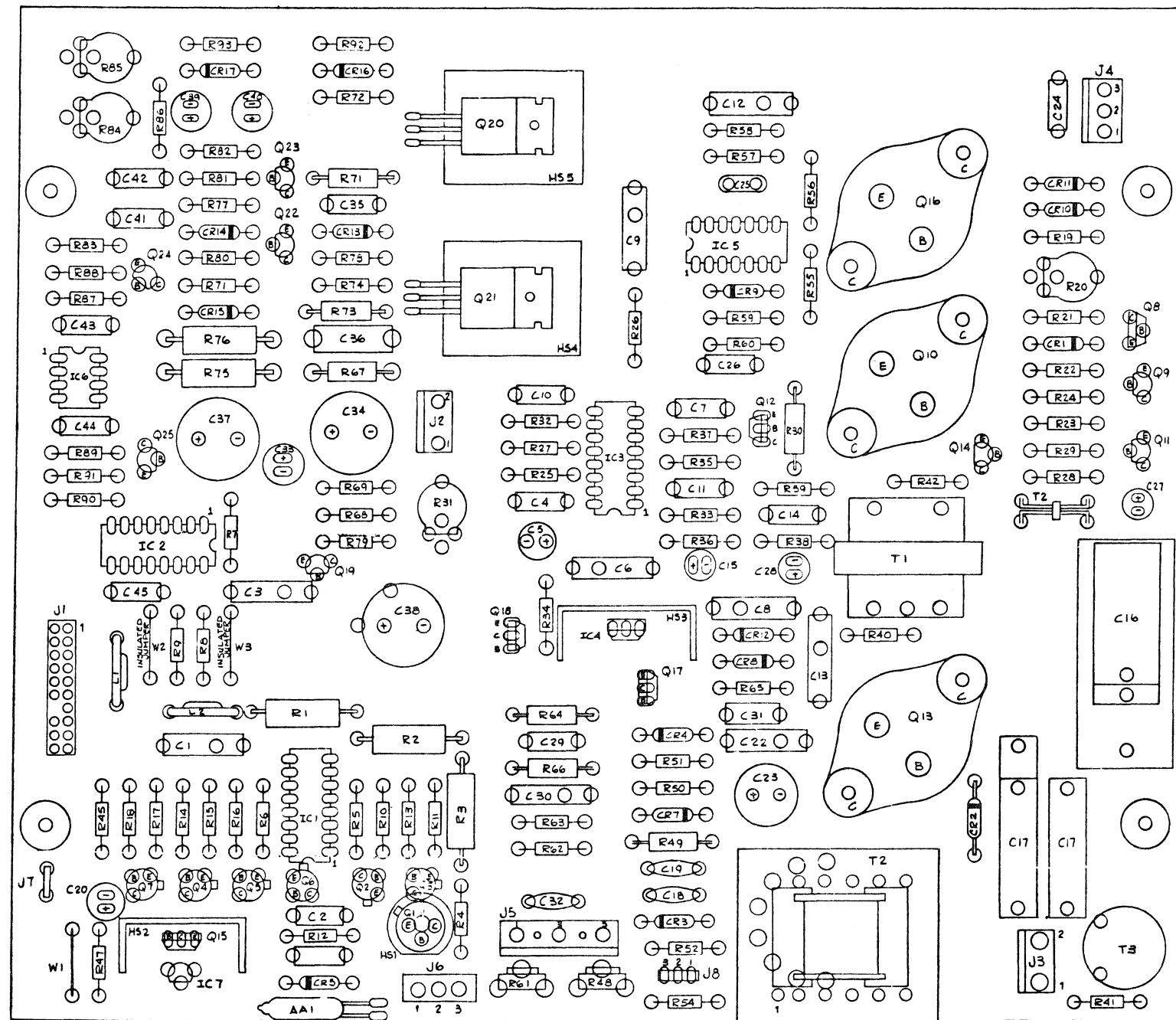
should be set with the "BRIGHTNESS" thumbwheel at maximum to cause background video to be just visible (used to set maximum brightness level).

H.HOLD

Adjusts the center frequency of the horizontal phase-locked loop. Adjust to ensure reliable locking of LHORDR.



DWG. NO.		SH	REV.	1	
REVISIONS					
ZONE	REV.	DESCRIPTION		DATE	APPROVED
	IX	ENG SPEC.		10-9-80	10/10/80



QTY		FROM	PART OR IDENTIFYING NO.	NOMENCLATURE OR DESCRIPTION		MATERIAL SPECIFICATION	
REQD		NO.		PARTS LIST			
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES TOLERANCES ARE: FRACTIONS DECIMALS ANGLES ± .XX ± .XXX ± °				CONTRACT NO.		CONVERGENT TECHNOLOGIES	
MATERIAL				APPROVALS		DATE	
FINISH				DRAWN		10-1-80	
NEXT ASSY				CHECKED		10-1-80	
USED ON				APPROVED			
APPLICATION				DO NOT SCALE DRAWING			
SIZE		FSCN NO.		DWG. NO.		REV.	
D				A 60-00015-00		IX	
SCALE				SHEET		1 OF 1	

11. CONSOLE SWITCHES

FUNCTION

The console switch set includes power-on and reset switches, the power-on LED, and fuses. The primary function of the console is to allow the operator to turn on and manually reset the system.

The only front-panel control is the power switch, which has a built-in LED. The reset switch and the fuses are accessible from the rear panel.

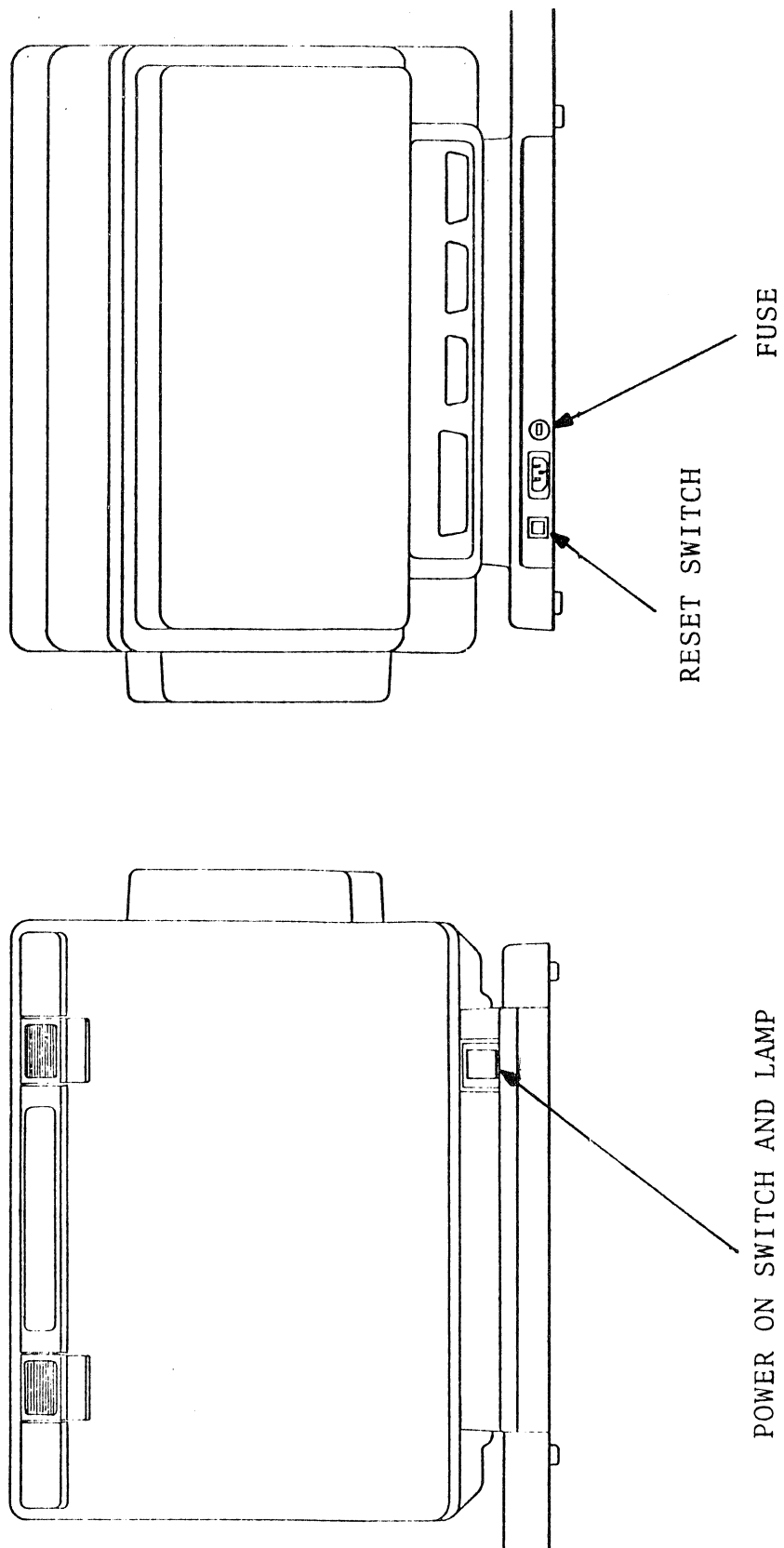


Figure 11-1. Switch and Fuse Locations

REFERENCES

Connectors

Reset switch: terminated with a 2-pin Molex connector that plugs into the motherboard.

Power-on switch: Soldered to the wiring harness

LED Indicator

The power-on LED is mounted within the power-on switch.

Refer to Power Supply Schematic
(Chapter 12)

Figure 11-2. Console Schematic

12. POWER SUPPLY

FUNCTIONAL DESCRIPTION

The workstation power supply is a 20 KHz switching regulator that generates the following voltages:

+5V	±1%
+12V	±1%
-12V	±1%
+32V	-0%
	+10%
-32V	-0%
	+10%

All voltages except $\pm 32V$ are independently adjustable. The total output power available is 190W. The minimum power available to the Multibus section of the workstation is as follows:

+5V	,	6.4A Max.
+12V	,	0.6A Max.
-12V	,	0.5A Max.

Both over-voltage and over-current protection is provided on +5V and $\pm 12V$. In particular, the 5V supply is "crow-barred" to guard against failure due to externally applied voltages. The $\pm 32V$ outputs are independently fused. The power supply is protected against excessive loads with an overall power-limit capability. Further, the supply is protected against excessive heat rise with a thermal switch that turns off primary power if the heat-sink temperature exceeds 80°C. The overall efficiency of the supply exceeds 70%.

The supply also generates the SFO signal, a square wave derived from the power line. This signal serves to generate the real-time clock and the power-fail detect signals in the mainframe.

AC CHARACTERISTICS

DC OUTPUT PARAMETERS											
ITEM	LIMITS	OUTPUT	OVERALL REG	RIPPLE <i>mV</i>	I-MIN ADC	I MAX ADC	TYPICAL LOAD VAR.	DELIV- ERED PWR-W	PROTEC- TION		COMMENTS
			% Linear Post Reg						OC	OV	
1	REQD.	+5VDC	±1 No, FB var.	50	.15	20	-25%	100	*	Y	FB loop closed on this output.
2	REQD	+12	±1 Yes	50	1.5	2.5	+32 -63	30	Y	Y	
3	REQD	-12	±1 Yes	50	0.2	0.5	+20	6	Y	Y	
	DESIGN										
4	REQD	+32	+10 No - 0	250	0.8	1.00	+20 - 0	40	*		Power for integrat- ed CRT Video Monitor.
	DESIGN										
5	REQD	-32	+10 No - 0	250	.4	0.5	+20 - 0	16	*		Regulated in monitor circuit assemblies.
	DESIGN										
REQD								192			

* Bulk current sensing on the primary.

DC CHARACTERISTICS

DC OUTPUT PARAMETERS											
ITEM	LIMITS	OUTPUT	OVERALL REG	RIPPLE <i>mV</i>	I-MIN ADC	I MAX ADC	TYPICAL LOAD VAR.	DELIV- ERED PWR-W	PROTEC- TION		COMMENTS
			% Linear Post Reg						OC	OV	
1	REQD.	+5VDC	±1 No, FB var.	50	.15	20	-25%	100	*	Y	FB loop closed on this output.
2	REQD	+12	±1 Yes	50	1.5	2.5	+32 -63	30	Y	Y	
3	REQD DESIGN	-12	±1 Yes	50	0.2	0.5	+20	6	Y	Y	
4	REQD DESIGN	+32	+10 No - 0	250	0.8	1.00	+20 - 0	40	*		Power for integrat- ed CRT Video Monitor.
5	REQD DESIGN	-32	+10 No - 0	250	.4	0.5	+20 - 0	16	*		Regulated in monitor circuit assemblies.
REQD								192			

* Bulk current sensing on the primary.

REFERENCES

Connectors

J1 - AC input
J2 - fan connector
J3 - DC output

Jumper

There is one jumper for selecting 110 or 220 V operation.

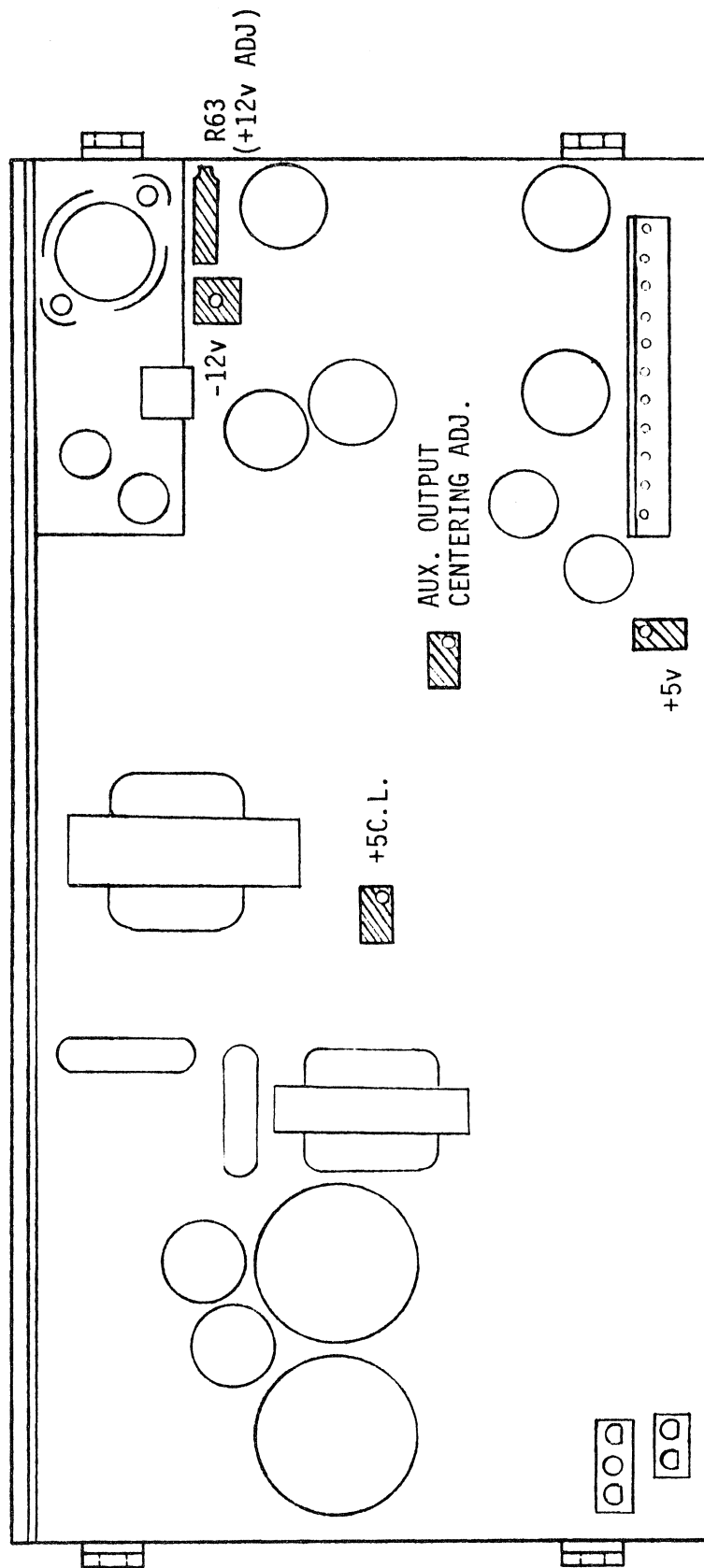
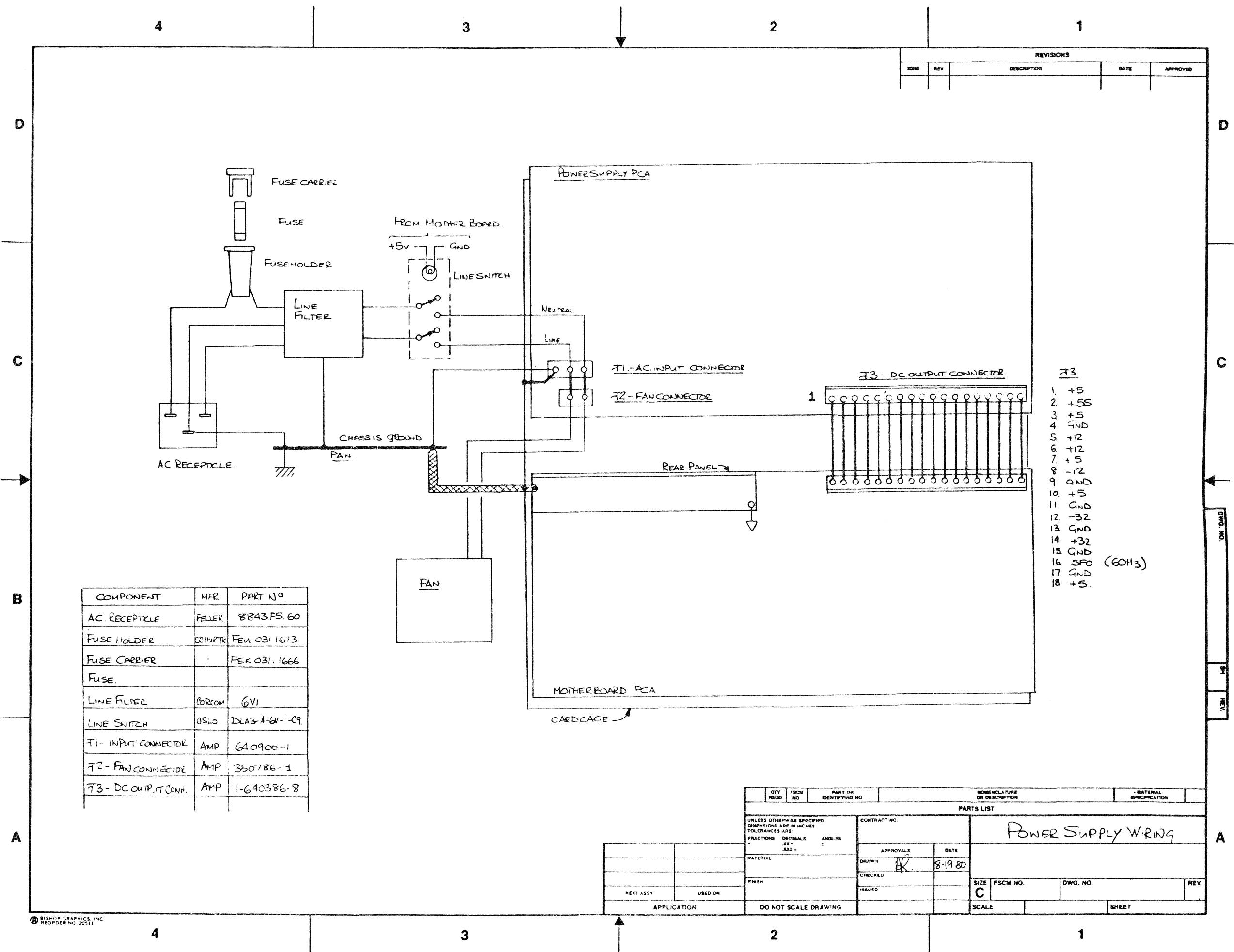


Figure 12-1. Adjustment Potentiometers



COMPONENT	MFR	PART NO.
AC RECEPTACLE	FELLER	8843.PS.60
FUSE HOLDER	SCHURTER	FEU 031.1673
FUSE CARRIER	"	FEK 031.1666
FUSE		
LINE FILTER	CORCOM	GV1
LINE SWITCH	OSLO	DLA3-A-64-1-C9
T1- INPUT CONNECTOR	AMP	640900-1
T2- FAN CONNECTOR	AMP	350786-1
T3- DC OUTPUT CONN.	AMP	1-640386-8

- T3
- 1. +5
 - 2. +5S
 - 3. +5
 - 4. GND
 - 5. +12
 - 6. +12
 - 7. +5
 - 8. -12
 - 9. GND
 - 10. +5
 - 11. GND
 - 12. -32
 - 13. GND
 - 14. +32
 - 15. GND
 - 16. SFO (60H3)
 - 17. GND
 - 18. +5

QTY REQD	FSCM NO.	PART OR IDENTIFYING NO.	NOMENCLATURE OR DESCRIPTION	MATERIAL SPECIFICATION
PARTS LIST				
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES TOLERANCES ARE: FRACTIONS DECIMALS ANGLES 1/16 0.0031 1/16 0.0031 1/16 0.0031				
CONTRACT NO.		APPROVALS		
DATE		DRAWN		
CHECKED		ISSUED		
NEXT ASSY		USED ON		SCALE
APPLICATION		DO NOT SCALE DRAWING		SHEET

Power Supply Wiring

8-19-80

SIZE C FSCM NO. DWG. NO. REV.

GLOSSARY

Auto-restart: An automatic system Reset after power-up. This happens whenever the power switch is turned on. The COLD latch on the processor board records the cause of the last Reset, whether manual via the Reset switch (warm start) or automatic via the Auto-restart function (cold start).

Bit numbering: Bits in an 8-bit byte, 16-bit word or 20-bit address are numbered from right to left, beginning with zero as the least significant bit and counting in hexadecimal. Thus, the bits in a byte are numbered 0 to 7, bits in a word are 0 to F (with bits 8 to F being the high or most significant byte), and bits in a 20-bit address are 0 to 13.

Board: One of several types of boards manufactured by CT. The set of boards includes: processor, I/O-memory, video control, mainframe motherboard, ROM expansion, RAM expansion, disk motherboard, floppy disk control and Winchester disk control. None of the boards has the Multibus form factor, although the motherboard accepts only standard Multibus boards in the user slots.

Buffer: An amplifying signal driver, not necessarily implying storage capability. Storage buffers are properly called "storage buffers."

Bus: A collection of signal lines used by more than one device. In general, three types of buses comprise a bus set (which is also referred to as a bus): the address bus, data bus and control bus. There are two bus sets in the mainframe: the local bus, of which the processor bus is a part, and the Multibus. See Figure 1-1.

Bus master: See "master."

Byte: 8 bits. See also "high byte" and "low byte."

Clear: To write a logical zero or series of zeroes to a register. See also "reset."

Cluster configuration: A hardware configuration in which workstations can be connected to one common Shared Resource Station (SRS).

Communications control: The hardware logic on the I/O-memory board that is comprised principally of the Z80A-SIO and its associated RS-232 and RS-422 interface circuits.

Console: The power switch, power-on LED, and reset switch available to the operator. The console is located on the mainframe enclosure and does not include the keyboard or the CRT monitor.

CPU: The 8086 chip located on the processor board.

CT: Convergent Technologies, the manufacturer of this system.

Disk: Hard (Winchester) disk or floppy disk.

Disk interface: The buffering and cable connector on the I/O-memory board that provides the interface between the local data (MEM) bus and the disk control boards located in Mass Storage Stations.

DMA: Usually, a direct-memory-access transfer utilizing the local address and local data (LA and MEM) buses, or the 8237 DMA chip located on the processor board (also called the local DMA). Direct memory access always involves memory at one (and only one) end of the transfer path. The local DMA cannot control transfers involving Multibus devices (only the local CPU can do this), but Multibus DMAs can control byte-at-a-time or word-at-a-time transfers involving local memory. Multibus-controlled DMA can only involve local memory; it cannot involve I/O devices on the local bus.

Driver: See "buffer."

ECR: See "extended control register."

Even byte: See "low byte."

Expansion ROM or RAM: ROM on the ROM expansion board, or RAM on the RAM expansion board.

Extended control register (ECR): A 16-bit control/status register, port 60, on the I/O-memory board used to augment the communication functions of the SIO chip.

FCR: See "floppy control register."

FDC: Floppy disk control or the board associated with this function.

Floppy control register: An 8-bit writable (not readable) register, port 72, used in floppy disk control.

Font RAM: A RAM array located on the video control board that is used to define character fonts associated with ASCII codes. The screen attribute register (SAR) controls the loading of this RAM.

Global attribute register: See "screen attribute register."

High byte: The most significant byte in a word (bits 8 to F), also called the odd byte since it always has an odd address in memory.

IAR: See "interrupt acknowledge register."

Integrated workstation (IWS): A hardware configuration in which the mainframe, keyboard and CRT monitor are all placed on top of the operator's desk.

Interrupt acknowledge register (IAR): An 8-bit register, port 30, on the I/O-memory board used by the CPU to acknowledge interrupts from the SIO.

Interrupt control: The hardware logic on the processor board that consists principally of the 8259A interrupt controller chip.

I/O circuit: An integrated circuit (chip) used for various I/O functions in the context of the CPU.

I/O control register (IOCR): A 16-bit register, port 56, on the I/O-memory board used for control and status information about power-failure detect, parity detect, Multibus I/O, line-printer interrupts, bus timeouts, realtime clock and memory write protection.

IOB bus: The local I/O bus, an 8-bit bus consisting of signal lines IOB0 to IOB7. This bus has MOS-type buffering.

IOCR: See "I/O control register."

Keyboard interface: The hardware logic on the I/O-memory board that contains, among other things, the 8251A serial I/O chip.

LA bus: See "local address bus."

Local address bus: A 20-bit bus on the processor and I/O-memory boards consisting of signal lines LA0 to LA13 (hex).

Local bus: The set of mainframe buses that serves all hardware resources provided by Convergent Technologies, as distinguished from the empty Multibus slots provided on the mainframe. On one side of the local bus interface the set consists of the LA (local address) bus, the MEM (local data) bus, and the IOB (local I/O) bus. On the other side of the local bus interface it consists of the processor bus, which serves the 8086 and 8087. The dividing line between the local bus and the outside world is the Multibus interface.

Local bus interface: A collection of hardware logic on the processor and I/O-memory boards that includes processor address decode, local bus control (8288 plus other logic), local I/O address decode and buffers. It provides interfacing between the processor and local resources.

Local data bus: A 16-bit bus on the processor and I/O-memory boards consisting of signal lines MEM0 to MEMF, hex.

Local DMA: The 8237 DMA located on the processor board.

Local I/O bus: An 8-bit bus on the processor and I/O boards consisting of signal lines IOB0 to IOB7.

Low byte: The least significant byte in a word (bits 0 to 7), also called the even byte since it always has an even address in memory.

Mainframe: A set of hardware boards consisting of one processor board, one I/O-memory board, and one motherboard, plus one console (power switch, power-on LED and reset switch), one power supply and one fan. In addition, the mainframe may optionally contain one ROM expansion board, one RAM expansion board, and one video control board. All required and optional boards in the

mainframe are contained in a single physical enclosure, although different types of enclosures are used in the various mainframe configurations. All mainframe boards are interconnected via edge connectors mating to the mainframe motherboard. See also, "system."

Mainframe motherboard: A backplane board which serves the primary function of interconnecting all other boards in the mainframe.

Main ROM or RAM: ROM on the processor board or RAM on the I/O-memory board. See also "expansion ROM or RAM."

Mass Storage Station (MSS): An enclosure which contains various combinations of disk drives, the associated control boards, a power supply, and Multibus slots for user expansion. It is connected to a workstation via a cable.

Master: A device or board that can obtain control over the local bus or the Multibus. The set of local-bus device masters consists of the CPU, NDP and DMA; their bus-access priorities are described in the chapter on "Bus Organization." The set of Multibus device and board masters is user-defined. On Multibus, any board that can control the Multibus is regarded as a Multibus master although the board can itself contain several device masters.

MB: See "Multibus."

MEM bus: See "local data bus."

Motherboard: There are motherboards for each enclosure. Try "mainframe motherboard."

Multibus: The IEEE 796 Microcomputer System Bus Standard. It is derived from the Intel standard and the term "Multibus" is an Intel trademark. It contains 20 address lines, 16 data lines and several control and power lines for a total of 86 lines at an edge connector designated P1.

Multibus interface: A collection of hardware logic on the processor board that includes the multibus control logic, Multibus address decode, 8289 bus arbiter chip and buffers. It provides asynchronous electrical separation between the local bus and the Multibus and is under the control of the local 8237 DMA which can prevent Multibus access to the local bus.

Multibus Master Access: A local-bus master access of a slave resource on the Multibus.

Multibus Slave Access: A Multibus master access of a slave resource on the local bus.

NDP: The optional 8087 numeric data processor chip.

NIBBLE: Four bits.

Operator: Any person who operates a system using the keyboard.

OS: Operating system.

Parity error register (PER): A 24-bit register, ports 52 and 54, on the processor board that latches the address at which a RAM parity error occurred and indicates whether the data was a word or byte, which byte in a word caused the error, and whether the last system Reset was manual or power-up.

Peripheral device: An electromechanical unit such as a keyboard, CRT, disk drive or printer. See also "I/O circuit."

PER: See "parity error register."

Port: Any chip or collection of chips addressable by the CPU as an I/O port.

Printer interface: The hardware logic on the I/O-memory board that generates and receives Centronics-compatible parallel signals for an external printer.

Printer status register (PSR): A 4-bit register, port 50, on the I/O-memory board that can indicate when the printer is selected, when its buffer is full, when it is busy printing and when it is out of paper.

Processor bus: The set of signal lines connecting the 8086 CPU and the 8087 with the local bus interface. It includes a multiplexed address/data/status bus with 20 signal lines named AD0 and AD13, plus several control lines. Demultiplexing of the bus takes place in the local bus interface. The processor bus is part of the local bus.

PSR: See "printer status register."

Reset: A system Reset, as defined by the reset pin functions of the various chips connected to the system Reset function; or a software register reset, equivalent to "clear."

ROM: Either ROM or PROM.

RT: Realtime, as in the 50/60 Hz realtime clock.

SAR: See "screen attribute register."

Screen attribute register (SAR): A 16-bit command and status register, port 58, used to determine CRT screen attributes and to control video-related DMA activity.

Set: To write a logical one or series of ones to a register.

SIO: The Z80A Serial I/O chip on the I/O-memory board. It controls all communication through the RS-232 and RS-422 ports. It is the only programmable chip that is not initialized by the operating system; normally, it is initialized by a user's application program.

Shared Resource Station (SRS): An enclosure used in cluster configurations to control access to shared resources. It contains a mainframe with a special version of the operating system.

Slave: A device or set of devices controlled by a master. All devices on the local bus can be slaves to certain local bus masters, but only the local ROM and RAM can be a slave to Multibus masters.

SRS: Shared Resource Station.

Standalone configuration: A hardware configuration in which a single workstation is attached to a Mass Storage Station (MSS) via a 15-foot cable.

System: A configuration with at least one mainframe.

Storage buffer: A buffer with storage, such as a flip-flop, latch or register.

User: An OEM person who may configure hardware on the Multibus.

Video control: The video control board which plugs into the mainframe motherboard.

WCR: See "Winchester control register."

WDC: See "Winchester disk control."

Winchester: A hard-disk drive, either the 8-inch or the 14-inch variety. The drives can have multiple heads.

Winchester control register: An 8-bit writable (not readable) register, port 7A, used in hard disk control.

Winchester disk control: Hard-disk control, or the board that performs this functions.

Word: 16 bits. A word consists of a high byte (odd address) and a low byte (even address). Word boundaries fall on even addresses.

Appendix A
IEEE 796 (Multibus) Standard

**PROPOSED
MICROCOMPUTER SYSTEM BUS STANDARD
(P796 BUS)**

**Submitted by
IEEE Computer Society Subcommittee
Microcomputer System Bus Group**

September, 1980

TABLE OF CONTENTS

	Page
1.	General
1.1	Scope
1.2	Object
1.3	Definitions
1.3.1	General System Terms
1.3.2	Signals and Paths
2.	Functional Description
2.1	796 Bus Elements
2.1.1	Masters
2.1.2	Slaves
2.1.3	796 Bus Signals
2.1.3.1	Control Lines
2.1.3.1.1	Clock Lines
2.1.3.1.2	Command Lines (MWTC*, MRDC*, IOWC*, IORC*)
2.1.3.1.3	Transfer Acknowledge Line (XACK*)
2.1.3.1.4	Initialize (INIT*)
2.1.3.2	Address and Inhibit Lines
2.1.3.2.1	Address Lines (24 Lines)
2.1.3.2.2	Byte High Enable Line (BHEN*)
2.1.3.2.3	Inhibit Lines (INH1* and INH2*)
2.1.3.3	Data Lines (DAT0*-DATF*)
2.1.3.4	Interrupt Lines
2.1.3.4.1	Interrupt Request Lines (INT0*-INT7*)
2.1.3.4.2	Interrupt Acknowledge (INTA*)
2.1.3.5	Bus Exchange Lines
2.1.3.5.1	Bus Request (BREQ*)
2.1.3.5.2	Bus Priority (BPRN* and BPRO*)
2.1.3.5.3	Bus Busy (BUSY*)
2.1.3.5.4	Common Bus Request (CBRQ*)
2.2	Data Transfer Operation
2.2.1	Data Transfer Overview
2.2.2	Signal Descriptions
2.2.2.1	Initialize (INIT*)
2.2.2.2	Constant Clock (CCLK*)
2.2.2.3	Address Lines (ADR0*-ADR13*)
2.2.2.4	Data Lines (DAT0*-DATF*)
2.2.2.5	796 Bus Commands
2.2.2.5.1	Read Operation
2.2.2.5.2	Write Operation
2.2.2.6	Transfer Acknowledge (XACK*)
2.2.2.7	Inhibit (INH1 and INH2*)
2.3	Interrupt Operations
2.3.1	Interrupt Signal Lines
2.3.1.1	Interrupt Request Lines (INT0*-INT7*)
2.3.1.2	Interrupt Acknowledge (INTA*)
2.3.2	Classes of Interrupt Implementation
2.3.2.1	Non-Bus Vectored Interrupts
2.3.2.2	Bus Vectored Interrupts
2.4	796 Bus Exchange
2.4.1	796 Bus Exchange Signals
2.4.1.1	Bus Clock (BCLK*)
2.4.1.2	Bus Busy (BUSY*)
2.4.1.3	Bus Priority In (BPRN*)
2.4.1.4	Bus Priority Out (BPRO*)
2.4.1.5	Bus Request (BREQ*)
2.4.1.6	Common Bus Request (CBRQ*)
2.4.2	Bus Exchange Priority Techniques
2.4.2.1	Serial Priority Technique
2.4.2.2	Parallel Arbitration Technique

TABLE OF CONTENTS (Continued)

3.	Electrical Specifications	3-1
3.1	General Bus Considerations	3-1
3.1.1	Logical and Electrical State Relationships	3-1
3.1.2	Signal Line Characteristics	3-2
3.1.2.1	In-Use Signal Line Requirements	3-2
3.1.2.2	Backplane Signal Trace Characteristics	3-3
3.1.3	Power Supply Specifications	3-3
3.1.4	Temperature and Humidity	3-5
3.2	Timing	3-5
3.2.1	Read Operations (I/O and Memory)	3-7
3.2.2	Write Operations (I/O and Memory)	3-7
3.2.3	Inhibit Operations	3-8
3.2.4	Interrupt Implementations	3-9
3.2.4.1	NBV Interrupts	3-9
3.2.4.2	BV Interrupts	3-9
3.2.5	Bus Control Exchanges	3-10
3.2.5.1	Serial Priority	3-11
3.2.5.2	Parallel Priority	3-12
3.2.6	Miscellaneous Timing	3-12
3.3	Receivers, Drivers and Terminations	3-12
4.	Mechanical Specifications	4-1
4.0	Introduction	4-1
4.1	Backplane Considerations	4-1
4.1.1	Board to Board Relationships	4-1
4.1.2	796 Bus Pin Assignments	4-2
4.2	796 Bus Board Form Factors	4-2
4.2.1	Connector Naming and Pin Numbering Standards	4-2
4.2.2	Standard Outline of Printed Wiring Board	4-2
4.2.3	Bus Connectors	4-3
5.	Levels of Compliance	5-1
5.1	Variable Elements of Capability	5-1
5.1.1	Data Path	5-1
5.1.2	Memory Address Path	5-1
5.1.3	I/O Address Path	5-1
5.1.4	Interrupt Attributes	5-2
5.2	Masters and Slaves	5-2
5.3	Compliance Level Notation	5-3
5.3.1	Data Path	5-3
5.3.2	Memory Address Path	5-3
5.3.3	I/O Address Path	5-3
5.3.4	Interrupt Attributes	5-3
5.3.5	An Example	5-4

LIST OF FIGURES

Figure		Page
1	796 Bus Master and Slave Example	2-3
2	796 Bus Interface Lines	2-9
3	796 Bus Read Operation	2-10
4	796 Bus Write Operation	2-10
5	796 Bus Address Line Usage	2-12
6	796 Bus Data Line Usage	2-13
7	Memory or I/O Read Timing	2-15
8	Memory or I/O Write Timing	2-16
9	Inhibit Timing for Write Operation	2-18
10	Non-Bus Vectored (BV) Interrupt Logic	2-21
11	Bus Vectored (BV) Interrupt Logic	2-23
12	Serial Priority Technique	2-26
13	Parallel Priority Technique	2-26
17	Setup, Hold, and Command Ringing Summary	3-4
18	Line-to-Line Coupling Characteristics	3-4
19	Read AC Timing	3-8
20	Write AC Timing	3-8
21	Inhibit AC Timing	3-9
22	Bus Vectored (BV) Interrupt AC Timing	3-9
23	Bus Exchange AC Timing	3-11
24	Common Bus Request AC Timing	3-11
25	Serial Priority AC Timing	3-12
26	Parallel Priority AC Timing	3-13
27	Constant Clock AC Timing	3-13
28	Command Separation AC Timing	3-13
29	Initialize AC Timing	3-13
31	796 Bus Backplane Card to Cage Separation	4-3
32	Typical 796 Bus Backplane	4-3
33	Connector and Pin Numbering	4-4
34	Standard Printed Wiring Board Outline	4-4

LIST OF TABLES

		Page
Table 1	790 Bus Power Supply Specifications	3-5
Table 2	796 Bus Timing Specifications Summary	3-6
Table 3	Bus Drivers, Receivers, and Terminations	3-14
Table 4	Pin Assignments of Bus Signals on 796 Bus Board Connector (P1) ...	4-5
Table 5	Pin Assignments of Bus Signals on 796 Bus Board Connector (P2) ...	4-6

Bibliography

Intel MultibusTM Specification

IEEE Microcomputer System Bus Standard (796 Bus)

1. General

1.1 Scope. One of the most important elements in a computer system is the bus structure that supplies the interface for all the hardware components. This bus structure contains the necessary signals to allow the various system components to interact with each other. It allows memory and I/O data transfers, direct memory accesses, generation of interrupts, etc. This document provides a detailed description of all the elements and features that make up the IEEE 796 Bus.

The bus supports two independent address spaces: memory and I/O. During memory cycles, the bus allows direct addressability of up to 16 megabytes using 24-bit addressing. During I/O bus cycles, the bus allows addressing of up to 64K I/O ports using 16-bit addressing. Both memory and I/O cycles can support 8-bit or 16-bit data transfers.

The bus structure is built upon the master-slave concept where the master device in the system takes control of the bus and the slave device, upon decoding its address, acts upon the command provided by the master. This handshake (master-slave relationship) between the master and slave devices allows modules of different speeds to be interfaced via the bus. It also allows data rates up to five million transfers per second (bytes or words) to take place across the bus.

Another important feature of the bus is the ability to connect multiple master modules for multiprocessing configurations. The bus provides control signals for connecting multiple masters in either a serial or parallel priority fashion. With either of these two arrangements, more than one master may share bus resources.

This document has been prepared for those users who intend to evaluate or design products that will be compatible with the IEEE 796 system bus structure. To this end, the necessary signal definitions and timing and electrical specifications have been covered in detail.

This standard deals only with the interface characteristics of microcomputer devices: not with design specifications, performance requirements, and safety requirements of modules.

Throughout this standard, the term "system" denotes the byte or word interface system that, in general, includes all the circuits, connectors, and control protocol to effect unambiguous data transfer between devices. The term "device" or "module" denotes any product connected to the interface system that communicates information via the bus, and that conforms to the interface system definition.

1.2 Object. This standard is intended:

- (1) To define a general purpose microcomputer system bus.
- (2) To specify the device-independent electrical and functional interface requirements that a module shall meet in order to interconnect and communicate unambiguously via the system.
- (3) To specify the terminology and definitions related to the system.
- (4) To enable the interconnection of independently manufactured devices into a single functional system.
- (5) To permit products with a wide range of capabilities to be interconnected to the system simultaneously.
- (6) To define a system with a minimum of restrictions on the performance characteristics of devices connected to the system.

1.3 Definitions. The following general definitions apply throughout this standard. More detailed definitions can be found in the appropriate section.

1.3.1 General System Terms

Compatibility. The degree to which devices may be interconnected and used without modification, when designed as defined in Sections 2 and 3 of this standard. Section 5 introduces the notion of levels of compliance and the corresponding notation.

Bus cycle. The process whereby digital signals effect the transfer of data bytes or word across the interface by means of an interlocked sequence of control signals. Interlocked denotes a fixed sequence of events in which one event must occur before the next event can occur.

Interface. A shared boundary between two systems, or between parts of systems, through which information is conveyed.

Interface system. The device-dependent electrical and functional interface elements necessary for communication between devices. Typical elements are: driver and receiver circuits, signal line descriptions, timing and control conventions, and functional logic circuits.

System. A set of interconnected elements which achieve a given objective through the performance of a specified function.

1.3.2 Signals and paths

Bus. A signal line or a set of lines used by an interface system to connect a number of devices, and to transfer information.

Byte. A group of eight adjacent bits operated on as a unit.

Word. Two bytes or sixteen bits operated on as a unit.

High state. The more positive voltage level used to represent one of two logical binary states.

Low state. The more negative voltage level used to represent one of two logical binary states.

Signal. The physical representation of data.

Signal level. The relative magnitude of a signal when compared to an arbitrary reference. Signal levels in this standard are specified in volts.

Signal line. One of a set of signal conductors in an interface system used to transfer messages among interconnected devices.

Signal parameter. That element of an electrical quantity whose values or sequence of values convey information.

2. Functional Description

This section provides an overall understanding of how the 796 Bus functions, and describes the elements that connect to the bus, the signals that provide the interface to the bus, and the different types of operations performed on the bus.

In this section, as well as throughout the specification, a clear and consistent notation for signals has been used. The Memory Write Command (MWTC) will be used to explain this notation. The terms one:zero and true:false can be ambiguous, so their use will be avoided. In their place, we will use the terms electrical High and Low (H and L). A nathan (asterisk) following the signal name (MWTC*) indicates that the signal is active low as shown:

MWTC* = Asserted at 0 volts

The signal (MWTC*) driven by a three state driver will be pulled up to V_{CC} when not asserted. The following is used to further explain the notation used in this specification.

Function	Electrical	Definition Logic	State
MWTC	H	1 True	Active, Asserted
	L	0 False	
MWTC*	L	1 True	Active, Asserted
	H	0 False	

2.1 796 Bus Elements. This subsection describes the elements (masters and slaves) that interface to the bus and the Multibus signal lines that comprise this interface.

2.1.1.1 Masters. A master is any module having the ability to control the bus. The master exercises this control by acquiring the bus through bus exchange logic and then generating command signals, address signals, and memory or I/O addresses. To perform these tasks, the master is equipped with either a central processing unit or logic dedicated to transferring data over to the bus to and from other destinations. Fig. 1 depicts a system that includes a master and two slave models.

The 796 Bus architecture can support more than one master in the same system, but in order to do this, there must be a means for each master to gain control of the bus. This is accomplished through the bus exchange logic (see 2.4).

Masters may operate in one of two modes of operation. However, a particular configuration should have either all mode 1 masters or all mode 2 masters. Modes 1 and 2 are defined as follows:

Mode 1: Masters are limited to single bus transfers per bus connect. System timing is rendered deterministic by compliance with a maximum bus busy period. That period is limited by the parameter *tbys0 max.* (see 3.2.5).

Mode 2: Masters are unlimited in the bus control. They may invoke bus override. Bus timeouts are allowed. Compliance with the maximum busy period is not required.

The last classification is included to allow for a very broad class of operations, giving users maximum flexibility in meeting these applications' needs. The first mode of operation is defined to allow systems designers to predict the overall performance of their systems without concern for uncontrolled timing parameters such as bus timeout.

2.1.2 Slaves. Another type of module that can interface to the bus is the slave. Slave modules decode the address lines and act upon the command signals from the masters. The slaves are not capable of controlling the bus. Some examples of bus slaves are shown in Fig. 1.

2.1.3 796 Bus Signals. Signals transferred over the bus can be grouped into several classes based on the functions they perform. The classes are:

- (1) Control Lines
- (2) Address and Inhibit Lines
- (3) Data Lines

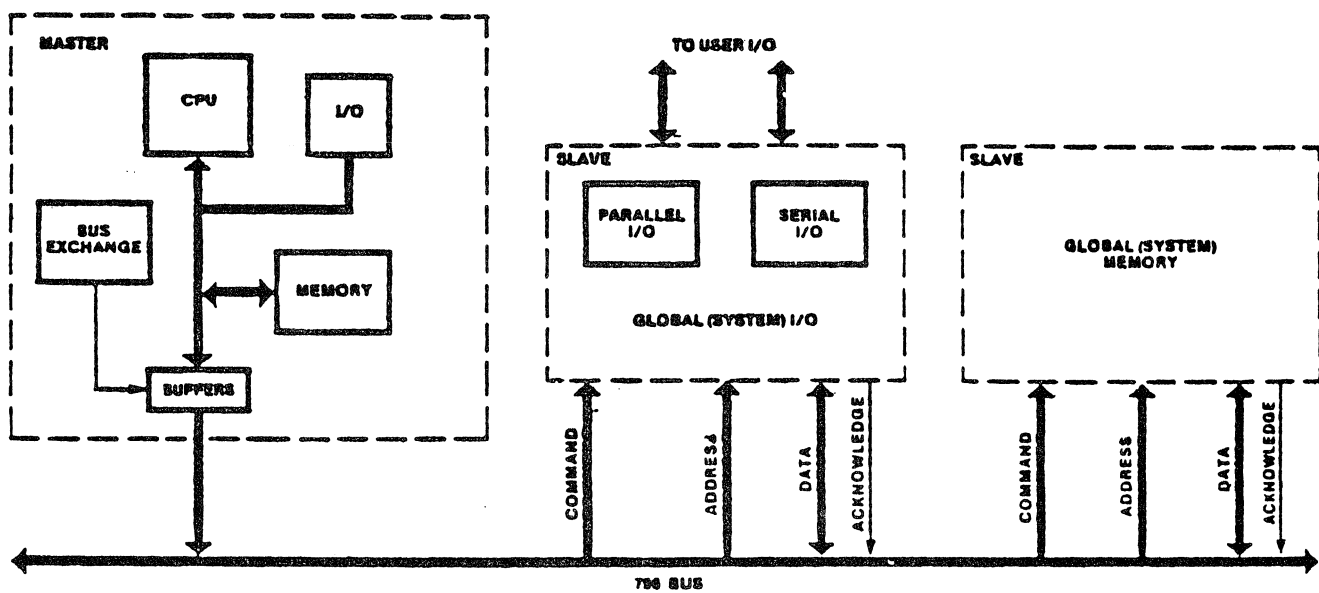


Fig. 1
796 Bus Master and Slave Example.

(4) Interrupt Lines

(5) Bus Exchange Lines

The following subsections explain the different classes of 796 Bus signals.

2.1.3.1 Control Lines. The following signals are classified as control lines:

Class	Function	Signal
Clocks	Constant Clock	CCLK*
	Bus Clock	BCLK*
Commands	Memory Write	MWTC*
	Memory Read	MRDC*
	I/O Write	IOWC*
	I/O Read	IORC*
Acknowledge	Transfer Acknowledge	XACK*
Initialize		INIT*

2.1.3.1.1 Clock Lines.

(1) *Bus Clock (BCLK*)*. A periodic signal used to synchronize the bus contention logic; it may be slowed, stopped, or single stepped. The Bus Clock shall be generated by one and only one source within the system. This means that each standalone bus master must have the capability of generating an acceptable clock that can optionally be connected to, or disconnected from, the bus. In a multimaster system, only one of the masters shall have its clock connected to the bus.

(2) *Constant Clock (CCLK*)*. A periodic signal of constant frequency, which may be used by masters or slaves as a master clock. The Constant Clock shall be generated by one and only one source within the system. This means that each bus master must have the capability of generating an acceptable clock that can optionally be connected to, or disconnected from, the bus. In a multimaster system, only one of the masters shall have its clock connected to the bus.

2.1.3.1.2 Command Lines (MWTC*, MRDC*, IOWC*, IORC*). The command lines are elements of a communication link between the masters and slaves. There are two command lines for memory and two command lines for I/O. An

active command line indicates to the slave that the address lines are carrying a valid address, and that the slave is to perform the specified operation. In a data write cycle, the active command line (MWTC* or IOWC*) additionally indicates that the data is valid on the bus. In a data read cycle, the transition of the command (MRDC* or IORC*) from active to inactive indicates that the master has received the data from the slave.

2.1.3.1.3 Transfer Acknowledge Line (XACK*). This line is used by the slaves to acknowledge commands from the master. XACK* indicates to the master that the requested action is complete, and that data has been placed on, or accepted from, the data lines.

2.1.3.1.4 Initialize (INIT*). The INIT* signal is generated to reset the entire system to a known internal state. This signal is usually generated prior to starting any operations on the system. INIT* may be generated by any or all of the bus masters or by an external source such as a buffered and debounced front panel switch.

2.1.3.2 Address and Inhibit Lines. The address and inhibit lines are used for the following signals:

Function	Signal
Address Lines	ADR0*-ADR17* (0-9, A-F, 10-17 in hexadecimal)
Byte High Enable	BHEN*
Inhibit Lines	INH1* and INH2*

2.1.3.2.1 Address Lines (24 lines). These lines, which specify the address of the referenced memory location or I/O device, allow a maximum of 16 megabytes (16,777,216 bytes) of memory to be accessed. When addressing an I/O device, a maximum of 16 address lines (ADR0*-ADRF*) are used, thus allowing the

addressing of a maximum of 64K devices. An I/O module must also be able to be configured to decode only 8 address lines (ADR0*-ADR7*) and ignore the upper 8 lines (see 2.2.2.3).

2.1.3.2.2 Byte High Enable Line (BHEN*). This byte control line is used to enable the upper byte (bits 8-F) of a 16-word bit word to drive the bus. The signal is used only on systems that incorporate 16-bit memory modules.

2.1.3.2.3 Inhibit Lines (INH1* and INH2*). The inhibit lines can be invoked for any memory read or memory write operation (MRDC* or MWTC*). An inhibit line is asserted by a slave to inhibit another slave's bus activity during a memory read or write operation. The inhibit signal generated by the inhibiting slave is derived from decoding the memory address lines. The inhibiting slave can decode a single address, a block of addresses, or any combination of single and block addresses.

When it detects the specific address during an actual command (MRDC* or MWTC*), the inhibiting slave generates an inhibit signal, which is sensed by the inhibited slave. When so inhibited, this slave module disables its drivers from all data, address, and acknowledge bus lines, although it may actually perform internal operations. (All modules that may be inhibited must have completed internal operations within 1.5 microseconds from the start of the command line. This interval [1.5 microseconds] is also the minimum acknowledge timing for modules issuing inhibits. This guarantees that inhibited modules have enough time to return to their normal state before the current bus command is completed.)

2.1.3.3 Data Lines (DAT0*-DATF*). These 16 bidirectional data lines transmit and receive information to and from a memory location or an I/O port. (DATF* is the most-significant bit and DAT0* is the least-significant bit). In 8-bit systems, only lines DAT0*-DAT7* are valid.

2.1.3.4 Interrupt Lines. The interrupt lines consist of the following signals:

Function	Signal
Interrupt Requests	INT0*-INT7*
Interrupt Acknowledge	INTA*

2.1.3.4.1 Interrupt Request Lines (INT0*-INT7*). Interrupts are requested by activating one of the eight interrupt request lines. INT0* has the highest priority and INT7* has the lowest priority.

2.1.3.4.2 Interrupt Acknowledge (INTA*). In response to an Interrupt Request signal, an Interrupt Acknowledge signal can be generated by a bus master with bus vectored interrupt capability. The Interrupt Acknowledge signal is used to freeze the interrupt status and request the placement of the interrupt vector address on the bus data lines.

2.1.3.5 Bus Exchange Lines. The bus exchange lines are used by the following signals:

Function	Signal
Bus Clock	BCLK*
Bus Request	BREQ*
Bus Priority	BPRN*, BPRO*
Bus Busy	BUSY*
Common Bus Request	CBRQ*

A master gains control of the bus through the manipulation of these signals.

2.1.3.5.1 Bus Request (BREQ*). A signal used by the bus masters in a priority resolution circuit to indicate a request for control of the bus.

2.1.3.5.2 Bus Priority (*BPRN and *BPRO**).** The priority functions allow masters to break deadlocks that occur when more than one master concurrently requests the bus. The Bus Priority In (*BPRN**) signal indicates to a particular master that no higher priority master is requesting use of the bus. The Bus Priority Out (*BPRO**) signal is used in serial (daisy chain) bus priority resolution schemes. In such a scheme, *BPRO** is passed by one master to the *BPRN** input of the master with the next lower bus priority; when active, the *BPRO** signal indicates that the higher priority master does not require control of the bus.

2.1.3.5.3 Bus Busy (*BUSY).** A signal activated by the master in control of the bus to indicate that the bus is in use. This prevents other masters from gaining control of the bus.

2.1.3.5.4 Common Bus Request (*CBRQ).** A signal that maximizes a master's data transfer rate to the bus by sensing the absence of other bus requests. The *CBRQ** signal does this by serving two functions. It indicates to the master controlling the bus whether or not another master needs to gain control of the bus. To the other masters, it is a means of notifying the controlling bus master that it must relinquish control of the bus if a higher priority request is pending.

2.2 Data Transfer Operation. The primary function of the 796 Bus architecture is to provide a path for the transfer of data between modules on the bus. The following subsections describe the different types of data transfers and the means by which they are implemented using the signals previously described. Fig. 2 can be referenced during the following discussion.

The discussion of the data transfer operation of the bus is covered in three parts:

- (1) An overview of the operation
- (2) A detailed description of the signals used in the transfer
- (3) A discussion of the specifics pertaining to the different transfers

It is assumed in this discussion that there is only one master on the bus, and therefore no bus contention exists. (The bus exchange logic is discussed in 2.4).

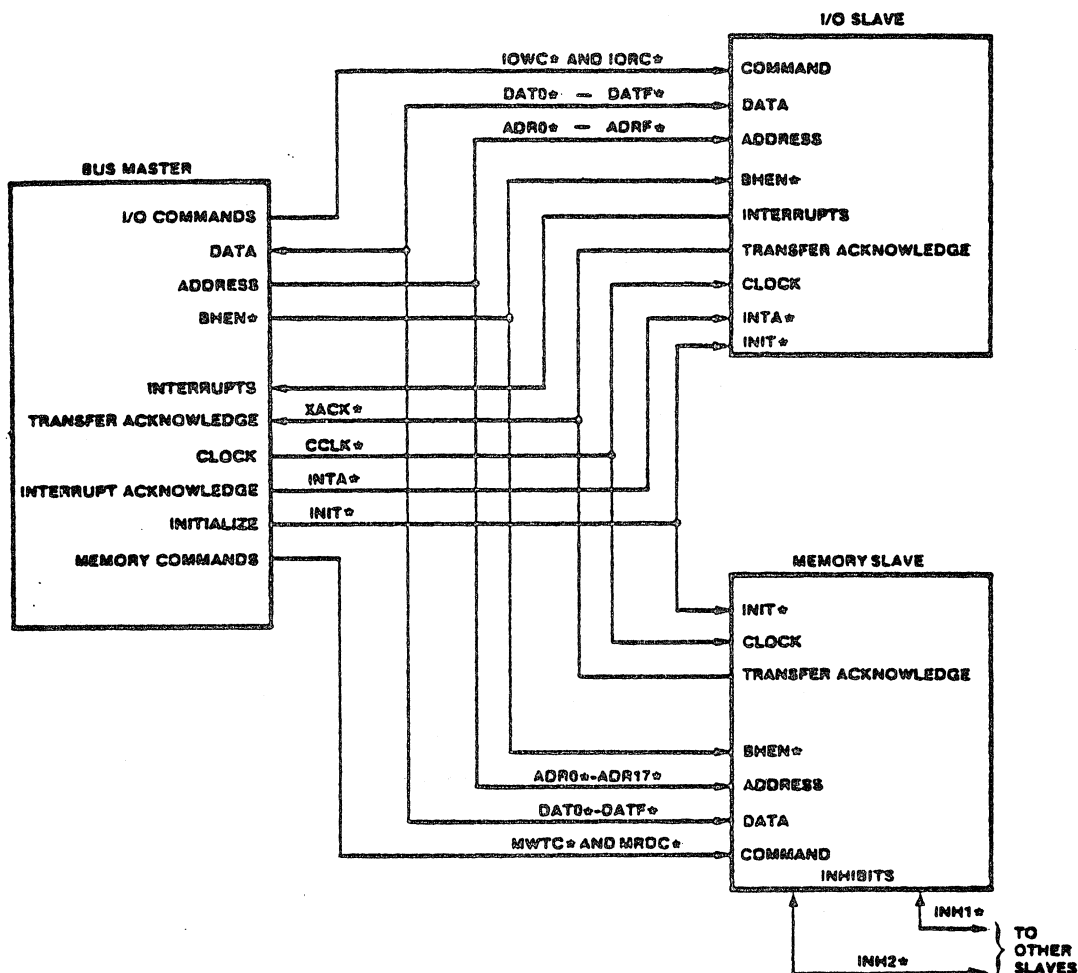


Fig. 2
796 Bus Interface Lines

2.2.1 Data Transfer Overview. A data transfer is accomplished as follows. First the bus master places the memory address or I/O port address on the address lines. (If the operation is a write, the data would also be placed on the data lines at this time.) The bus master then generates a command (I/O read or write, or memory read or write), which activates the appropriate bus slave. The slave accepts the data if it is a write operation, or places the data on the data lines if it is a read operation. A Transfer Acknowledge signal is then sent to the bus master by the bus slave, allowing the bus master to complete its cycle by removing the command from the command line and then clearing the address and data lines. Fig. 3 and Fig. 4 show the basic timing for a read and write data transfer operation.

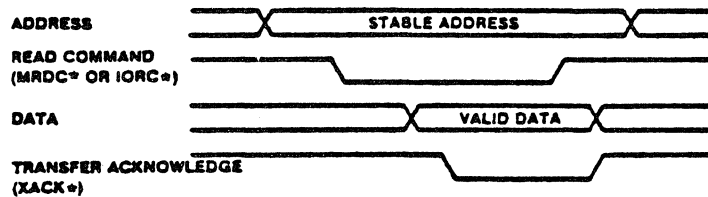


Fig. 3
796 Bus Read Operation

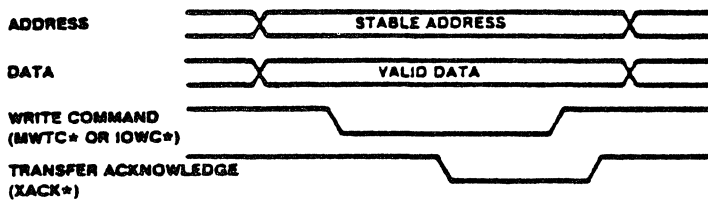


Fig. 4
796 Bus Write Operation

2.2.2 Signal Descriptions. This subsection provides a detailed description of the 796 Bus signals. Included are timing, signal origination, and other information pertaining to the specific function that each signal performs in the data transfer operation.

2.2.2.1 Initialize (INIT*). Prior to any operation of the bus, all system modules should be reset to a known internal state. This can be accomplished by an INIT* signal initiated by one of three sources:

- (1) A power-on clear circuit (RC network), which holds INIT* low until the power supplies reach their specific voltage outputs
- (2) A reset button, which is sometimes provided on the system front panel for operator use. Note that this button must be debounced
- (3) A software command that can be implemented to pull down the INIT* line.

The INIT* line is driven by open-collector gates and requires signal conditioning to meet the electrical specifications of the bus.

2.2.2.2 Constant Clock (CCLK*). The Constant Clock signal, which is driven by only one source, provides a timing source for any or all modules on the bus. CCLK* is a periodic signal with a specified frequency and is driven by a clock driver circuit.

2.2.2.3 Address Lines (ADR0*-ADR17*). The address lines are used to specify the address of the memory location or the I/O device that is being referenced by the command. There are 24 address lines, binary coded, to allow up to 16,777,216 bytes of memory to be referenced. These lines are driven by three-state drivers and are always controlled by the master using the bus.

For I/O bus cycles, master modules have the option of generating 8 bit or 16 bit addresses. Because of this, all I/O slaves must be capable of being configured to decode 8 address bits (ADR0*-ADR7*) and ignore the upper address bits or to decode all 16 bits of address (ADR0*-ADR17*). Note that in a system using 8 bit I/O addresses, the value of the upper 16 bits of address is unknown. A master generating only 8 bit address may set the upper 16 address bits to any arbitrary value.

Refer to Fig. 5 for an example of address line usage.

2.2.2.4 Data Lines (DAT0*-DAT15*). These are 16 bidirectional data lines used to transmit and receive information to and from a memory location or I/O port. The 16 lines are driven by the master on write operations and by the addressed slave (memory or I/O) on read operations. Both 16-bit and 8-bit transfers can be accomplished by using only lines DAT0*-DAT7* (with DAT0* being the least-significant bit).

There are three types of transfers that take place across the bus:

- (1) Transfer of low (even) byte on DAT0*-DAT7*
- (2) Transfer of high (odd) byte on DAT0*-DAT7* (using swap byte function)
- (3) Transfer of a 16-bit word

Fig. 6 shows the data lines, and the contents of these lines for the three types of transfers mentioned.

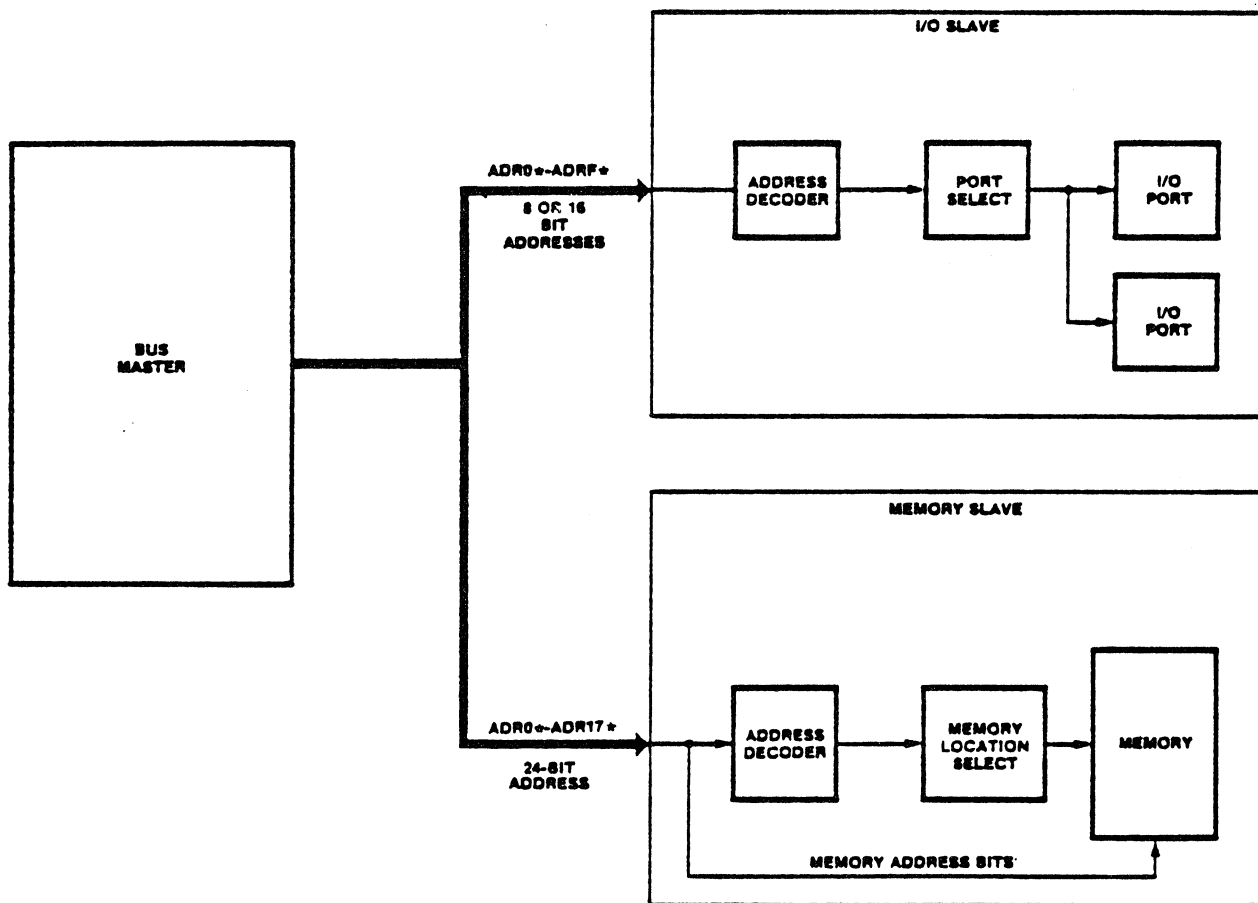


Fig. 5
796 Bus Address Line Usage

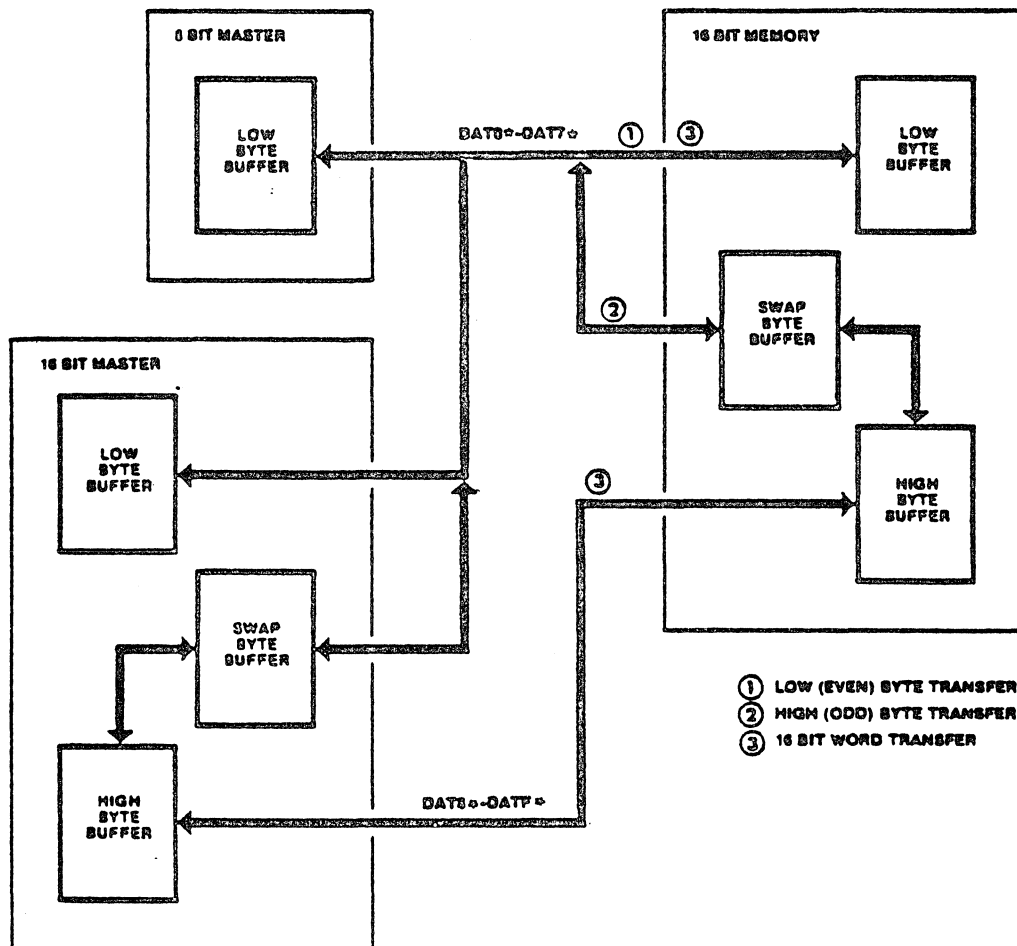


Fig. 6
796 Bus Data Line Usage

Two signals control the data transfers. Byte High Enable (BHEN*) active indicates that the bus is operating in the 16-bit mode, and the address bit 0 (ADRO*) defines an even-byte or odd-byte transfer.

For an even byte transfer, BHEN* and ADRO* are inactive, indicating the transfer of an even byte. The transfer takes place across data lines DAT0*-DAT7*.

For an odd-byte transfer, BHEN* is inactive and ADRO* is active, indicating the transfer of an odd byte. On this type of transfer, the odd (high) byte is transferred through the Swap Byte Buffer to DAT0*-DAT7*. The high (odd) byte is transferred across on DAT8*-DAT15* to make 8-bit and 16-bit systems compatible.

For a 16-bit transfer, BHEN* is active and ADR0* is inactive. On this type of transfer, the low (even) byte is transferred on DAT0*-DAT7* and the high (odd) byte is transferred across the bus on DAT8*-DATF*.

The 796 Bus data lines are always driven by three-state drivers.

2.2.2.5 796 Bus Commands. In this subsection we will discuss the command lines and how they work in conjunction with other lines to accomplish a read or a write operation. There are four command lines:

Function	Line
Memory Read Command	MRDC*
I/O Read Command	IORC*
Memory Write Command	MWTC*
I/O Write Command	IOWC*

The command lines, which are driven by three-state drivers on the bus master, indicate to the slave the action that is being requested.

2.2.2.5.1 Read Operation. The two read commands (MRDC* and IORC*) initiate the same basic type of operation. The only difference is that MRDC* indicates that the memory address is valid on the address lines, whereas IORC* indicates that the I/O port address is valid on the address lines. This address (memory or I/O port) must be valid on the bus 50 nanoseconds prior to the read command being generated. When the read command is generated, the slave module (memory or I/O port) places the data on the data lines and returns a Transfer Acknowledge (XACK*) signal, indicating that the data is on the bus. When the bus master receives the acknowledge, it strobes in the data and removes the command (MRDC* or IORC*) from the bus. The slave address (memory or I/O port) remains valid on the bus a minimum of 50 nanoseconds after the read command is removed. XACK* must be removed from the bus within 65 nanoseconds after the command is removed to allow for the next bus cycle. Fig. 7 shows the timing for the Memory Read or I/O Read command.

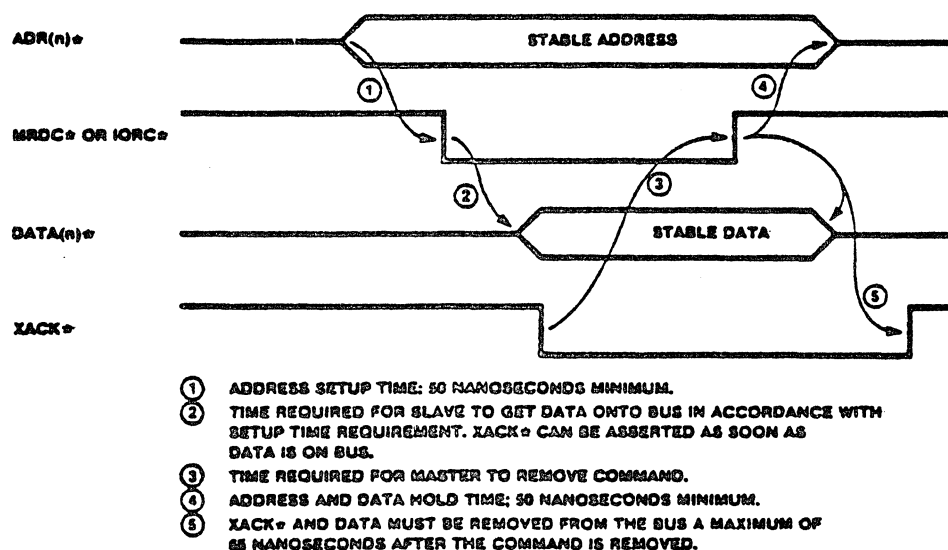


Fig. 7
Memory or I/O Read Timing

2.2.2.5.2 Write Operation. The write commands (MWTC* and IOWC*) initiate the same basic type of operation. MWTC* indicates that the memory address is valid on the address lines, whereas IOWC* indicates that the I/O port address is valid on the address lines. The address (memory or I/O) and data must be valid on the bus 50 nanoseconds prior to the write command being generated. This requirement allows data to be latched on either the leading or trailing edge of the command. When the write command (MWTC* or IOWC*) is asserted, the data on the data lines is stable and can be accepted by the slave. The slave indicates acceptance of the data by returning a Transfer Acknowledge (XACK*), allowing the bus master to remove the command, address, and data from the bus. XACK* must be removed from the bus within 65 nanoseconds to allow for the next bus cycle. Fig. 8 shows the timing for the Memory Write or I/O Write command.

2.2.2.6 Transfer Acknowledge (XACK*). The Transfer Acknowledge (XACK*) signal is the response from the bus slave (memory or I/O) indicating that the commanded read or write operation is complete and that the data has been placed on, or accepted from, the data lines. In effect, this signal (XACK*) allows the bus master to complete the current bus cycle.

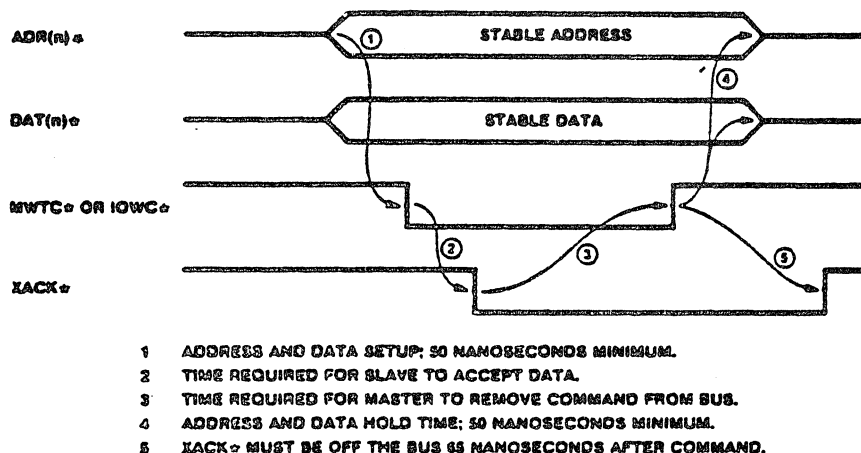


Fig. 8
Memory or I/O Write Timing

If a bus master addresses a non-existent or malfunctioning memory or I/O module, an acknowledge will not be returned to the master. If this should occur, the bus master would normally wait indefinitely for an acknowledge and would therefore never relinquish control of the 796 Bus. To avoid this possibility, a bus timeout function can optionally be implemented on a bus master to terminate a bus cycle after a preset interval, even if no acknowledge has been received. A bus timeout can therefore be defined as any data transfer cycle terminated by the master before the Transfer Acknowledge (XACK*) signal is received. The minimum allowable bus timeout interval is 1.0 millisecond.

2.2.2.7 Inhibit (INH1* and INH2*). The inhibit lines can be invoked for any memory read or memory write operation (MRDC* or MWTC*). An inhibit line is asserted by a slave to inhibit another slave's bus activity during a memory read or write operation. The inhibit signal generated by the inhibiting slave is derived from decoding the memory address lines ($t_{ID} = 100$ nanoseconds maximum). The inhibiting slave can decode a single address, a block of addresses, or any combination of single and block addresses.

When it detects the specific address during the actual command (MRDC* or MWTC*), the inhibiting slave generates an inhibit signal, which is sensed by the inhibited slave. When so inhibited, this slave module disables its drivers from all

data, address, and acknowledge bus lines, although it may actually perform internal operations. (All modules that may be inhibited must have completed internal operations with 1.5 microseconds from the start of the command line. This interval [1.5 microseconds] is also the minimum acknowledge (t_{ACC}) timing for modules issuing inhibits. This guarantees inhibited modules enough time to return to their normal state before the current bus command is completed.)

The slaves involved in the inhibit operation fall into three inhibit classes: top (inhibit) priority, middle priority, and bottom priority. In reference to the above paragraphs, a higher priority slave module would be the inhibiting slave and a lower priority slave would be the inhibited slave. $INH1^*$ is asserted (during the appropriate address) by a middle priority slave (such as a read-only memory module or memory-mapped I/O module) to inhibit the bus activity of a bottom priority slave (such as a read/write RAM module). $INH2^*$ is asserted (at the appropriate address) by a top priority slave (such as an auxiliary or a bootstrap ROM module) to inhibit the bus activity of a middle priority slave. The top priority slave shall also assert $INH1^*$ so that a bottom priority slave will also be inhibited. The inhibit lines shall be asserted low by open collector (or equivalent) drivers. When both a middle and a top priority inhibiting slave are activated, $INH1^*$ will be asserted by drivers on both modules.

The use of the inhibit signals during memory reads ($MRDC^*$) shall not cause any adverse effects within the inhibited slave module. That is, data in the inhibited slave shall not be altered and its status register, if any, shall not be affected.

The use of the inhibit signals during memory writes ($MWTC^*$) shall be allowed, and might or might not affect the data within the inhibited slave. If the data is affected, it shall be only within the one byte (or word) that is being addressed. (No other data within the inhibited slave shall be altered.)

The inhibit signals, when issued, shall be generated with 100 nanoseconds (t_{ID}) after the address is stable. (See Fig. 9.) A command may be generated as early as 50 nanoseconds (t_{AS}) after the address is stable. This timing can cause the inhibit to occur after the command has been received by the inhibited module. To prevent false acknowledges, modules that can be inhibited must not generate an acknowledge until the inhibit signals have had time to become valid (50 nanoseconds after the command).

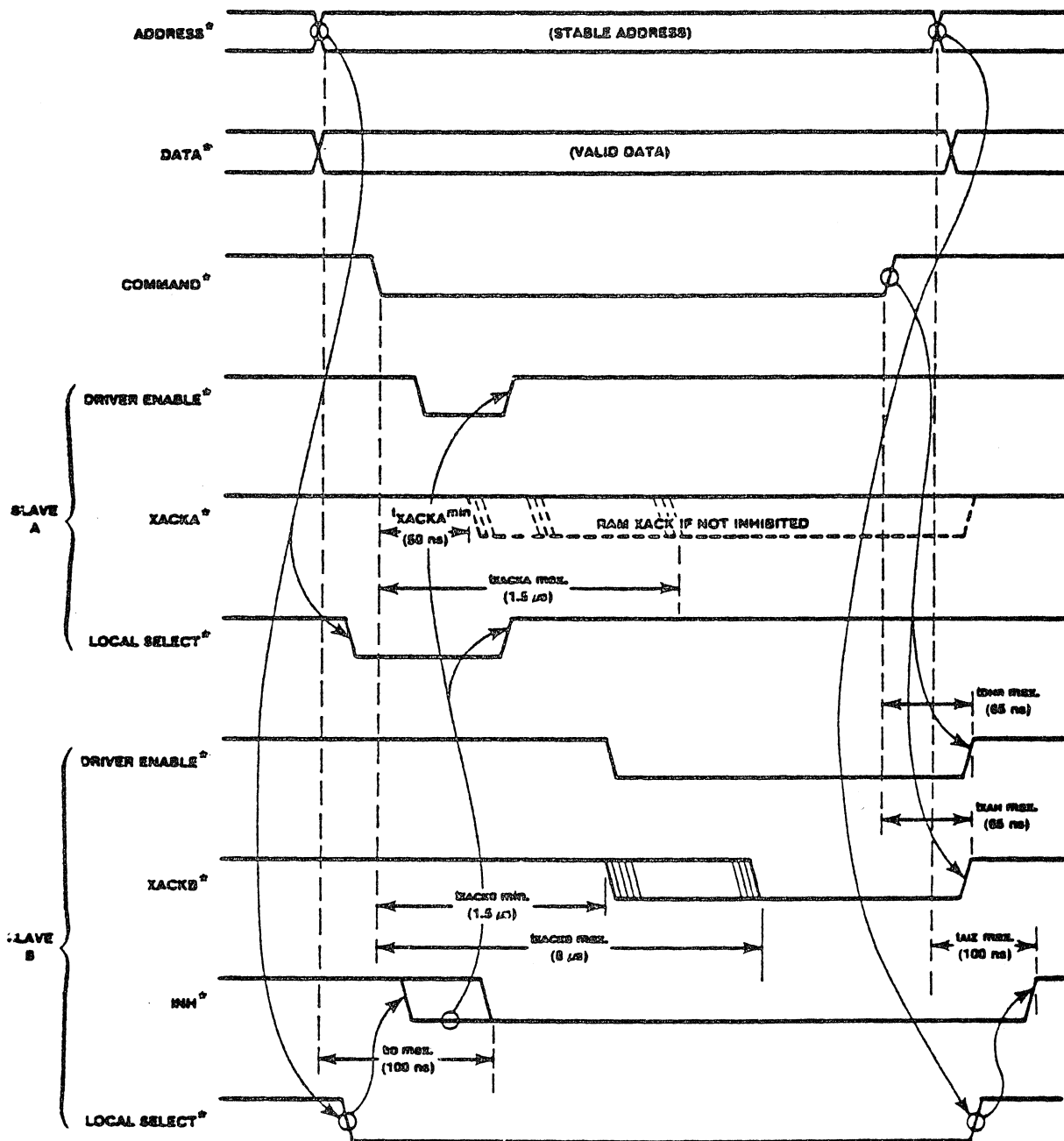


Fig. 9
Inhibit Timing for Write Operation

Fig. 9 shows the timing for an inhibit operation. In this example, both PROM and RAM have the same memory addresses; therefore, the PROM inhibits the RAM.

Although inhibit signals may be generated during IORC*, IOWC*, or INTA* operations, these signals are ignored by other slaves (including the slave that should respond to the INTA*, IORC*, or IOWC*).

2.3 Interrupt Operations. The following subsections explain the 796 Bus signal lines used in the interrupt operation, and the two different types of interrupt implementation. Refer to Section 5.1.4 for information on levels of compliance with respect to interrupt attributes.

2.3.1 Interrupt Signal Lines.

2.3.1.1 Interrupt Request Lines (INT0*-INT7*). A set of interrupt request lines (INT0*-INT7*) is provided on the bus. An interrupt is generated by activating one of the eight interrupt request lines with an open-collector driver. All interrupts are level-triggered, rather than edge-triggered. Requiring no edge to trigger an interrupt allows several sources to be attached to each line. The interrupt request lines are prioritized, with INT0* having the highest priority and INT7* having the lowest priority.

2.3.1.2 Interrupt Acknowledge (INTA*). An interrupt acknowledge line (INTA*), driven by the bus master, requests the transfer of interrupt information on the bus. The specific information timed onto the bus depends on the implementation of the interrupt scheme. In general, the leading edge of INTA* indicates that the address bus is active; the trailing edge indicates that data is present on the data lines.

2.3.2 Classes of Interrupt Implementation. There are two types of interrupt implementation schemes: Non-Bus Vectored (NBV) and Bus Vectored (BV). The two schemes are explained in the following subsections.

2.3.2.1 Non-Bus Vectored Interrupts. Non-Bus Vectored (NBV) interrupts are those interrupts handled on the bus master that do not require the 796 Bus for transfer of the interrupt vector address. The interrupt vector address is generated by the interrupt controller on the master and transferred to the processor over the local bus. The slave modules generating the interrupts can reside on the master module or on other bus modules, in which case they use the 796 Bus interrupt request lines (INT0*-INT7*) to generate their interrupt requests to the bus master. When an interrupt request line is activated, the bus master performs its own interrupt operation and processes the interrupt. Fig. 10 shows an example of NBV interrupt implementation.

2.3.2.2 Bus Vectored Interrupts. Bus Vectored (BV) interrupts are those interrupts that transfer the interrupt vector address over the 796 Bus from the slave to the bus master using the INTA* command signal.

When an interrupt request occurs, the interrupt control logic on the bus master interrupts its processor. The processor on the bus master generates the INTA* command, freezing the state of the interrupt logic for priority resolution. The bus master also locks (retains the bus between bus cycles) the 796 Bus to guarantee itself back-to-back bus cycles. After the first INTA* command, the bus master's interrupt control logic puts an interrupt code on the 796 Bus address lines. The interrupt code is the address of the highest priority active interrupt request line. At this point in the BV interrupt procedure, two different sequences can occur because the 796 Bus can support masters that generate either two or three INTA* commands.

If the bus master generates two INTA* commands, one more INTA* command will be generated. This second INTA* causes the bus slave interrupt control logic to transmit its interrupt vector address on the 796 Bus data lines. The address is used by the bus master to service the interrupt.

If the bus master generates three INTA* commands, two more INTA* commands will be generated. These two INTA* commands allow the bus slave to put its 2-byte interrupt vector address on the 796 Bus data lines (one byte for each INTA*). The interrupt vector address is used by the bus master to service the interrupt.

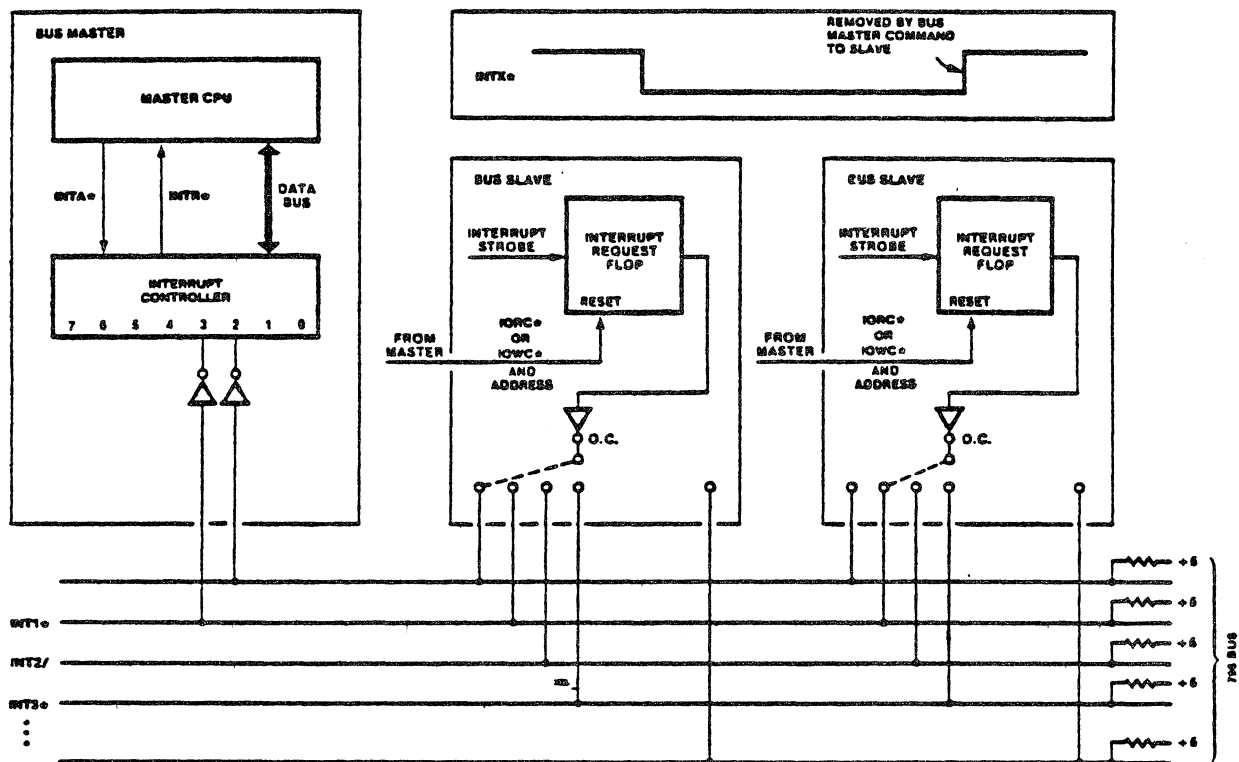


Fig. 10
Non-Bus Vectored (NBV) Interrupt Logic

NOTE

The 796 Bus can support only one type of Bus Vectored interrupt in a given system. However, the 796 Bus can support both Bus Vectored (BV) and Non-Bus Vectored (NBV) interrupts within the same system.

Fig. 11 depicts an example of BV interrupt implementation.

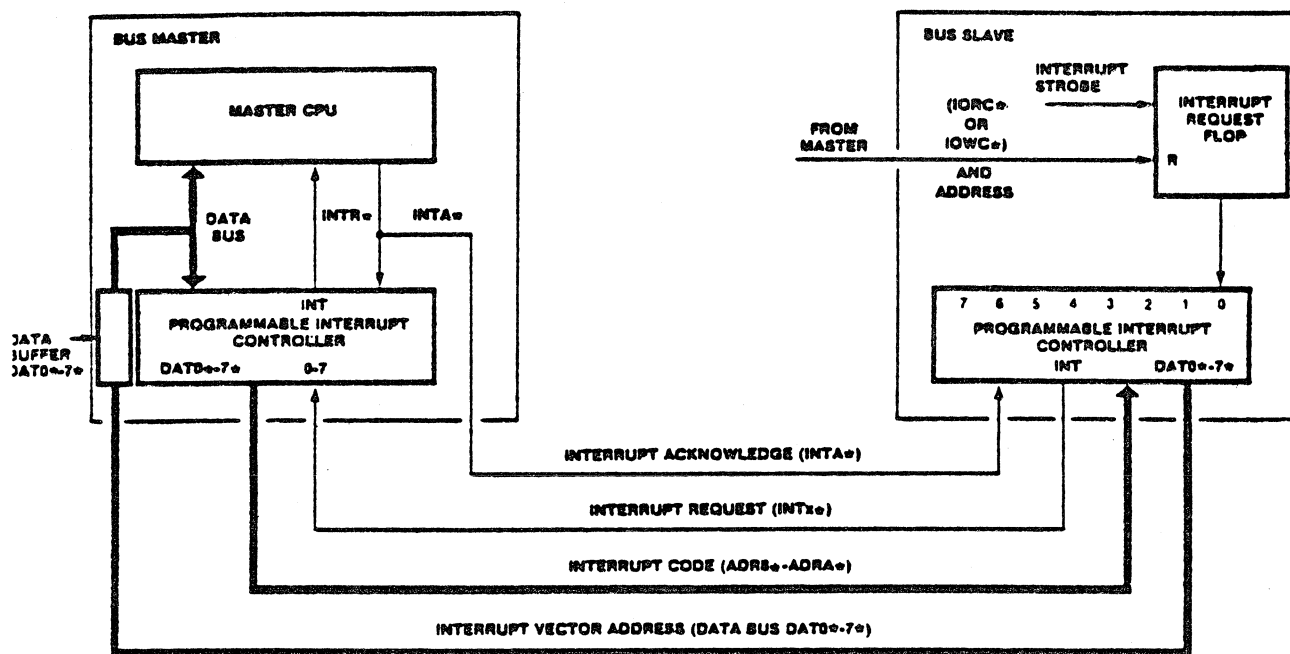
2.4 796 Bus Exchange. The 796 Bus can accommodate several bus masters on the same system, each taking control of the bus as it needs to effect data transfers. The bus masters request bus control through a bus exchange sequence.

The discussion of the 796 Bus exchange will be separated into three parts. The first part explains the signals involved, the second part discusses the bus exchange priority techniques (serial and parallel), and the third part explains the implementation of the exchange logic.

2.4.1 796 Bus Exchange Signals. A set of six signals is used to implement the bus exchange operation. All bus exchange signals are synchronized by BCLK*.

2.4.1.1 Bus Clock (BCLK*). This periodic clock signal is used to synchronize the exchange logic, with synchronization occurring on the trailing (high-to-low) edge of the pulse. BCLK* has a duty cycle of approximately 50 percent, a maximum frequency of 10 MHz, and can be slowed, stepped, or stopped as called for by system design. There is no requirement for synchronization between BCLK* and CCLK*, but they may be derived from the same source. The BCLK* line is driven by a TTL clock driver.

2.4.1.2 Bus Busy (BUSY*). This signal is driven by the master in control of the bus. All other masters monitor BUSY* to determine the state of the bus. This bidirectional signal, which is driven by an open-collector gate, is synchronized by BCLK*.



TIMING

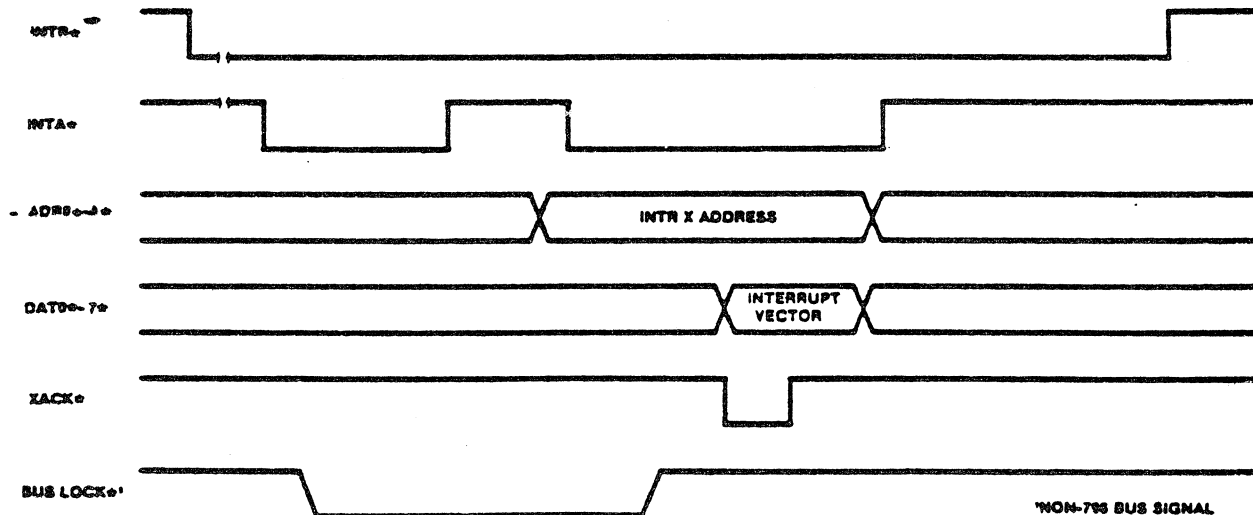


Fig. 11
Bus Vectored (BV) Interrupt Logic

2.4.1.3 Bus Priority In (BPRN*). A non-bused signal that indicates to a master that no master of higher priority is requesting control of the bus. BPRN* is synchronized by BCLK* and driven by TTL gates. In a serial resolution scheme, this is the master's input from the priority chain. In a parallel resolution scheme, this is the master's input from the parallel priority circuit.

2.4.1.4 Bus Priority Out (BPRO*). This non-bused signal, when activated by a bus master, indicates to the bus master of the next lower priority that it may gain control of the bus (i.e., no higher priority requests are pending for control of the bus). This signal is used only in a daisy-chained serial priority resolution scheme and should be connected to the Bus Priority In (BPRN*) input of the next lower priority bus master. BPRO* is driven by TTL gates and is synchronized by BCLK*.

NOTE

Each bus master must allow its BPRO* signal to be disconnected from the BPRO* line on the 796 Bus so that, if desired, a parallel priority resolution scheme can be used. This capability is to allow some bus masters to have their BPRN* inputs driven by a central parallel resolution circuit instead of by the BPRO* of the next higher priority master.

2.4.1.5 Bus Request (BREQ*). The Bus Request (BREQ*) line is used with the parallel priority resolution scheme, and is a request of the master for 796 Bus control. The priorities of the BREQ* from each master are resolved in a parallel priority resolution circuit. The highest priority request enables the BPRN* input of that master, allowing it to gain control of the bus. BREQ* is synchronized by BCLK* and is a TTL output.

2.4.1.6 Common Bus Request (CBRQ*) (Optional). Any master that wants control of the 796 Bus but does not control it, can activate CBRQ* with an open-collector gate. If CBRQ* is high, it indicates to the bus master that no other master is requesting the bus and therefore the present bus master can retain the bus. There are times when this can save the bus exchange overhead for the current master. This is

because quite often when a master is controlling the bus, there are no other masters that are requesting the bus. Without CBRQ*, only BPRN* indicates whether or not another master is requesting the bus and, for BPRN*, only if the other master is of higher priority. Between the master's bus transfer cycles, in order to allow lower priority masters to take the bus if they need it, the master must give up the bus. At the start of the master's next transfer cycle, the bus must be regained. If no other master has the bus, this can take approximately three BCLK* periods. To avoid this overhead of unnecessarily giving up and regaining the bus when no other masters need it, CBRQ* may be used. Any master that wants but does not have the bus, must drive this line low (true). The master that has the bus can, at the end of a transfer cycle, sense CBRQ*. If it is not low, then the bus does not have to be released, thereby eliminating the delay of regaining the bus at the start of the next cycle. (At any time before the master's next cycle, any other master desiring the bus will drive CBRQ* and cause the master to relinquish the bus at that time.)

Masters that use CBRQ* must be able to disable that function such that they can be used with masters that do not generate the CBRQ* signal.

2.4.2 Bus Exchange Priority Techniques. Two bus exchange priority techniques are discussed: a serial technique and a parallel technique. Fig. 12 and Fig. 13 illustrate these two techniques. Note that the parallel and serial schemes are compatible and therefore can be combined and used together on the same bus. The bus exchange implementation discussed in 2.4.3 is the same for both techniques.

2.4.2.1 Serial Priority Technique. Serial priority resolution is accomplished with a daisy-chain technique (see Fig. 12). With such a scheme, the bus priority output (BPRO*) of each master is connected to the bus priority input (BPRN*) of the next lower priority master. The BPRN* of the highest priority master in the serial chain shall either be always active or connected to a central Bus Arbiter as described in 2.4.2.2. The latter connection would be used if a parallel-serial priority structure were used.

Serial priority resolution is accomplished in the following manner. The BPRO* output for a particular master is asserted if and only if its BPRN* input is active and that master is not requesting control of the bus. Thus, if a master requests control of the bus, it shall set its BPRO* high, which in turn disables the BPRN* of all lower

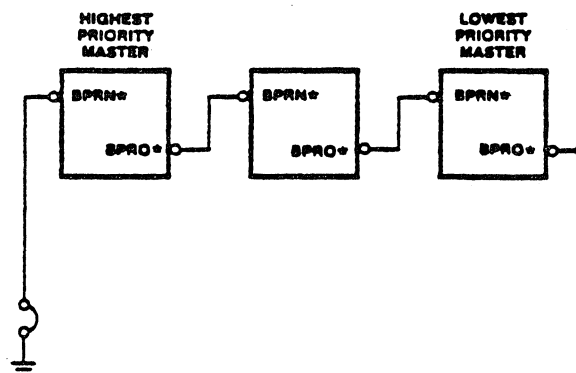


Fig. 12
Serial Priority Technique

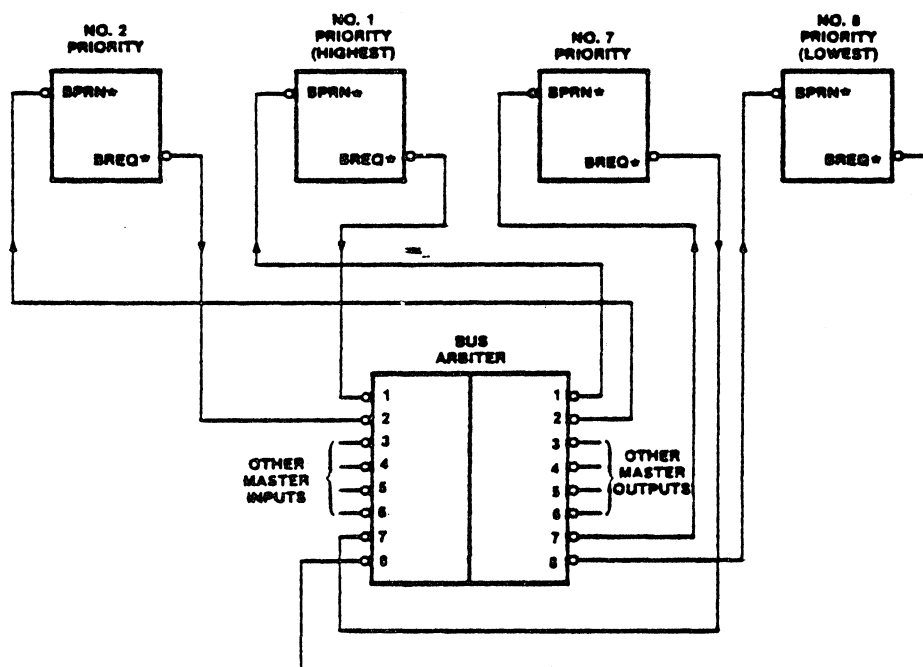


Fig. 13
Parallel Priority Technique

priority masters. The number of masters that can be linked in a serial chain is limited by the fact that the BPRN* signal must propagate through the entire chain within one BCLK* cycle. If the maximum BCLK* of 10 MHz is used, then the number of masters in a serial chain is limited to three.

2.4.2.2 Parallel Arbitration Technique. In the parallel technique, the bus allocation is determined by a Bus Arbiter (see Fig. 13). This may be a priority scheme, which determines the next master by a fixed priority structure or some other mechanism for allocation (e.g., sequential). The BREQ* lines are used by the arbiter to signal the next master on the appropriate BPRN* line. The BPRO* lines are not used in the parallel allocation BPRN* scheme.

3. Electrical Specifications

This section presents the electrical specifications for the 796 Bus as follows:

- (1) General bus considerations of the state relationships, signal line characteristics, and power supplies
- (2) Timing specifications for the bus signals
- (3) Specifications for the signal line drivers and receivers, as well as the electrical termination requirements

When electrical specifications indicate minimum or maximum values for the bus, they must be measurable at any point on the bus.

Note that a particular implemented bus could have any amount of bus propagation delay and ringing (before setup times), as long as all bus parameters (e.g., setup, hold, and other times) are met at all points on the bus. However, to facilitate the design of a compatible set of modules (masters and slaves) that use the bus, the standard maximum bus propagation delay will be specified as $t_{PD} \text{ (max)}$.

3.1 General Bus Considerations.

3.1.1 Logical and Electrical State Relationships. The signal names indicate whether or not the signal lines on the 796 Bus are active high or active low. If the signal name ends with a nathan (" * ", a non-superscript asterisk), then the signal is active low and its logical-electrical state relationship for that signal is:

<u>Logical State</u>	<u>Electrical Signal Level</u>	<u>At Receiver</u>	<u>At Driver</u>
0	H=TTL high state	$5.25V \geq H \geq 2.0V$	$5.25V \geq H \geq 2.4V$
1	L=TTL Low state	$.8V \geq L \geq -.5V$	$.5V \geq L \geq 0V$

If the signal name has no nathan (no " * "), then the signal is active high and its logical-electrical state relationship for that signal is:

<u>Logical State</u>	<u>Electrical Signal Level</u>	<u>At Receiver</u>	<u>At Driver</u>
0	L=TTL low state	$.8V \geq L \geq -.5V$	$.5V \geq L \geq 0V$
1	H=TTL high state	$5.25V \geq H \geq 2.0V$	$5.25V \geq H \geq 2.4V$

These specifications are based on TTL, where the power source is $5V \pm 5\%$, referenced to logic ground (GND).

When specified, current flow into a node has a positive sign and current flow out of a node has a negative sign.

3.1.2 Signal Line Characteristics. The following subsections describe two types of requirements. The first includes the requirements on the signal line that are measured when the signal line is in use. The second type includes those that are measured under special test conditions.

3.1.2.1 In-Use Signal Line Requirements. During normal use, the rise and fall times of the signals depend on which type of driver is used (see 3.3). Typical rise and fall times are:

	<u>Open Collector</u>	<u>Totem Pole</u>	<u>3-State</u>
Rise Time	—	10 ns	10 ns
Fall Time	10 ns	10 ns	10 ns

The maximum signal propagation delay on the bus is $t_{PD}(\max)$. This is measured from the edge of any one board plugged into the 796 Bus to any other board plugged into the 796 Bus:

$$t_{PD}(\max) = 3 \text{ ns}$$

These dynamic signal parameters can be tested by using 74S20 gates as drivers.

NOTE

For all boards plugged into the bus, the setup, hold, and any other times are measured at the edge of the board where it is plugged into the bus. This means that all board-internal delays must be taken into account, while still providing for the setup, hold, and other times.

After Power-Up, the following specifications apply:

- (1) Bus termination required for each signal line (see 3.3).
- (2) Settling time for all command line signals (see 2.2.2.5) after a transition is zero.

On these lines the ringing cannot go beyond the noise immunity levels—i.e., high, minimum; or low, maximum. This also applies to the Transfer Acknowledge and Inhibit lines.

For all address lines (see 2.2.2.3) the signals must be stable (settled) at least 50 nanoseconds before any command line goes active (setup time). This settling requirement means there can be no ringing beyond the noise immunity levels (high, minimum; low, maximum). These requirements also apply to the data lines (see 2.2.2.4) during any write operations.

For all data lines during read operations, the setup time is zero before the Transfer Acknowledge (XACK *) signal goes active; and the hold time is zero after the read-type command goes inactive.

The setup, hold, and command ringing are summarized and graphically presented in Fig. 17.

3.1.2.2 Backplane Signal Trace Characteristics. Requirements for line-to-line coupling characteristics are shown in Fig. 18. The specific test conditions under which the specifications are to be met are also shown.

3.1.3 Power Supply Specifications. Table 1 provides all power supply specifications. All voltages not shown in Table 1 that may be required on a board plugging into the 796 Bus should be derived from one of the standard voltages (+5V, +12V, -12V).

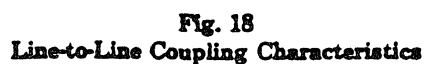
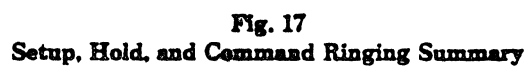


Table 1
796 Bus Power Supply Specifications

Parameter	Standard ¹			
	Ground	+5	+12	-12
Mnemonic	GND	+5V	+12V	-12V
Bus Pins	P1-1,2,11,12, 75,76,85,86	P1-3,4,5,6, 81,82,83, 84	P1-7,8	P1-79,80
Tolerance	Ref.	±1%	±1%	±1%
Combined Line & Load Reg	Ref.	0.1%	0.1%	0.1%
Ripple (Peak to Peak)	Ref.	50 mV	50 mV	50 mV
Transient Response (50% Load Change)		100 μ s	100 μ s	100 μ s

¹Point of measurement is at connection point between motherboard and power supply. At any card edge connector a degradation of 2% maximum (e.g. voltage tolerance $\pm 2\%$) is allowed.

3.1.4 Temperature and Humidity. Bus specifications should be met with temperature and humidity within the following ranges:

Temperature: 0-55 °C (32-150 °F); freemoving air across modules and bus

Relative Humidity: 90% maximum (no condensation)

This represents standard environment for the 796 Bus. It may be desirable to create more (or less) severe environmental restrictions in some applications.

3.2 Timing. This subsection describes all timing specifications on the 796 Bus. It does not present descriptions or functional relationships (which are given in Section 2); however, this section does imply the functionality when relating two signals.

Table 2 summarizes the timing specifications in this section. For detailed descriptions, refer to the specific subsection(s) in the right-hand column.

The timing diagrams shown in this section usually show the minimum or maximum values required for each parameter. However, for clarity, the diagrams in this specification do not usually show both the minimum and maximum parameters. For this reason, the bus timing specification (Table 2) should be referenced for complete information. The timing diagrams show how all of the parameters are defined in relation to the signals involved.

Table 2
796 Bus Timing Specifications Summary

<u>Parameter</u>	<u>Description</u>	<u>Minimum</u>	<u>Maximum</u>	<u>Units</u>	<u>Reference</u>
t_{AH}	Address Hold Time	50		ns	3.2.1, 3.2.2, 3.2.4
t_{AIZ}	Address to Inhibit High Delay	0	100	ns	3.2.3
t_{AS}	Address Setup Time (at slave board)	50		ns	3.2.1, 3.2.2, 3.2.4
t_{BCY}	BCLK * Period	100	∞	ns	3.2.5
t_{BPRNO}	BPRN * to BPRO *	0	30	ns	3.2.5
t_{BPRNS}	BPRN * to BCLK * Setup Time	22		ns	3.2.5
t_{BPRO}	BCLK * to BPRO *	0	40	ns	3.2.5
t_{BREQH}	BCLK * to BREQ * High Delay	0	35	ns	3.2.5
t_{BREQL}	BCLK * to BREQ * Low Delay	0	35	ns	3.2.5
t_{BSYO}	CBRQ * to BUSY * to BCLK *	—	12	μs	3.2.5
t_{BUSY}	BUSY * delay from BCLK *	0	70	ns	3.2.5
t_{BUSYS}	BUSY * to BCLK Setup Time	25		ns	3.2.5
t_{BW}	BCLK * Width	$0.35t_{BCY}$	$0.65t_{BCY}$		3.2.5
t_{CBRO}	BCLK * to CBRQ *	0	60	ns	3.2.5
t_{CBRQS}	CBRQ * to BCLK * Setup Time	35		ns	3.2.5
t_{CCY}	CCLK * Period	100	110	ns	3.2.6
t_{CMD}	Command Pulse Width	100	t_{TOUT}	ns	3.2.1, 3.2.2
t_{CMPH}	Command Hold Time	20		ns	3.2.1, 3.2.2
t_{CPM}	Central Priority Module Resolution Delay (parallel priority)	0	$t_{BCY} - t_{BREQ}$ $- 2t_{PD}$ $- t_{BPRNS}$ $- t_{SKEW}$		3.2.5
t_{CSEP}	Command Separation	100		ns	3.2.4, 3.2.6
t_{CW}	CCLK * Width	$0.35t_{CCY}$	$0.65t_{CCY}$	ns	3.2.6
t_{DHR}	Read Data Hold Time	0	65	ns	3.2.1, 3.2.4
t_{DHW}	Write Data Hold Time	50		ns	3.2.2
t_{DS}	Write Data Setup Time	50		ns	3.2.2

Table 2
796 Bus Timing Specifications Summary (Cont'd.)

Parameter	Description	Minimum	Maximum	Units	Reference
t _{DXL}	Read Data Setup Time to XACK*	0		ns	3.2.1, 3.2.4
t _{IAD}	XACK* Disable from Inhibit (internal parameter on an inhibited slave; used to determine t _{XACKA} min.)	0	100 (arbitrary)	ns	2.3.2
t _{ID}	Inhibit Delay	0	100 (recommend < 100 ns)	ns	3.2.3
t _{INIT}	INIT* Width	5		ms	3.2.6, 3.2.7
t _{INTA}	INTA* Width	250		ns	3.2.4
t _{OUT}	Timeout Delay	1	dc (∞)	ms	—
t _{PD}	Standard Bus Propagation Delay		3	ns	3.1.2, 3.2.5
t _{SKEW}	BCLK* Skew		t _{PD}		3.2.5
t _{XACK}	XACK* Time (for slaves without inhibit function)	0	8	μs	3.2.1, 3.2.2, 3.2.4
t _{XACKA}	XACK* Time of an Inhibited Slave	t _{IAD} + 50 ns	1500	ns	3.2.3
t _{XACKB}	XACK* Time of an Inhibiting Slave	1500	8000	ns	3.2.3
t _{XAH}	XACK* Hold Time	0	65	ns	3.2.1, 3.2.2, 3.2.4
Serial Priority	See 3.2.5				

3.2.1 Read Operations (I/O and Memory). A read operation transfers data from memory or from I/O to the master that is controlling the bus (see 2.2.). The lines involved and timing specifications for a read operation are shown in Fig. 19. See the special inhibit operation in 3.2.3.

3.2.2 Write Operations (I/O and Memory). A write operation transfers data from the master controlling the bus to memory or I/O (see 2.2). Timing for a write operation is shown in Fig. 20. See 3.2.3 for inhibit operations.

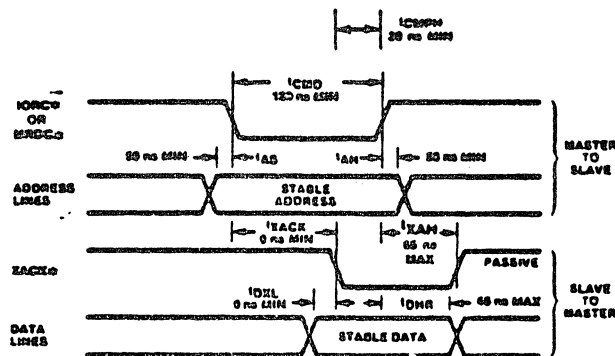


Fig. 19
Read AC Timing

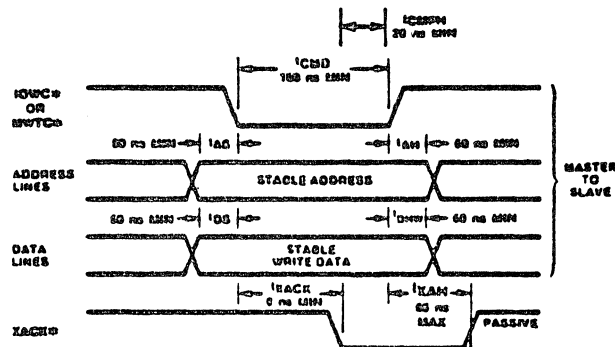


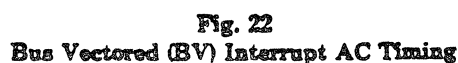
Fig. 20
Write AC Timing

3.2.3 Inhibit Operations. An inhibit operation may accompany any memory read or memory write operation. The main effect is for one slave to inhibit another slave from driving the data lines and from returning (driving) an acknowledge (XACK*). I/O addresses cannot be inhibited. Although inhibit signals may be generated during IORC*, IOWC*, or INTA* operations, these signals are ignored by other slaves (including the slave that should respond to the INTA*, IORC*, or IOWC*). Inhibit timing is as illustrated in Fig. 21. Related subsections are:

Subsection	Number
Functional Descriptions	2.1.3.2.3
Timing Specification Summary	3.2
Read Operations	3.2.1
Write Operations	3.2.2
Interrupt Implementations	3.2.4



3.2.4.2 BV Interrupts. BV interrupts are those interrupts that transfer the interrupt vector address along the 796 Bus from the slave to the bus master in response to the INTA* command signal. BV interrupt timing is shown in Fig. 22.



When an interrupt request occurs, the interrupt control logic on the bus master interrupts its processor. The processor on the bus master generates an INTA* command, which freezes the state of the interrupt logic on the 796 Bus for priority resolution. The bus master also locks the 796 Bus (retains the bus between bus cycles) to guarantee itself back-to-back bus cycles. After the first INTA* command, the bus master's interrupt control logic puts an interrupt code onto the 796 Bus address lines. The interrupt code is the address of the highest priority active interrupt request line. At this point in the BV interrupt procedure, two different sequences could take place. The difference occurs because the 796 Bus can support masters that generate either two or three INTA* commands during the interrupt process.

If the bus master generates two INTA* commands, one more INTA* command will be generated. This second INTA* causes the bus slave interrupt control logic to transmit its interrupt vector address on the 796 Bus data lines. The address is used by the bus master to service the interrupt.

If the bus master generates three INTA* commands, two more INTA* commands will be generated. These two INTA* commands allow the bus slave to put its 2-byte interrupt vector address onto the data lines (one byte for each INTA* command). The interrupt vector address is used by the bus master to service the interrupt.

NOTE

The 796 Bus can support only one type of BV interrupt in a given system. However, it can support both BV and NBV interrupts in the same system.

Subsections related to BV and NBV interrupts are:

Subsection	Number
Functional Descriptions	2.3.2.2
Timing Specification Summary	3.2

3.2.5 Bus Control Exchanges. A bus control exchange takes control of the bus (i.e., the ability to do read, write, and interrupt acknowledge operations) from one master and gives it to another master. See 2.4 for a functional description of this process.

For a master that does not use the bus signal CBRQ* (Common Bus Request), the timing specifications in Fig. 23 apply.

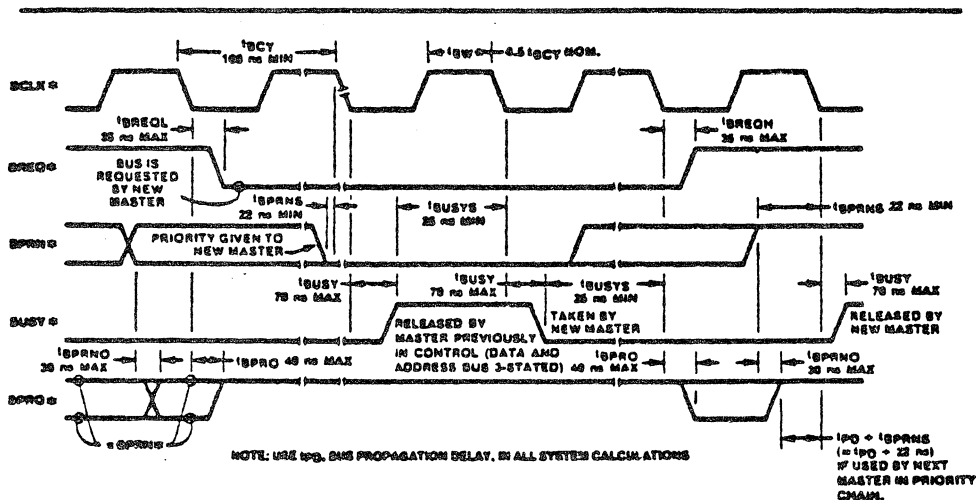


Fig. 23
Bus Exchange AC Timing

For a system using CBRQ* (Common Bus Request), each master must also satisfy the timing requirements illustrated in Fig. 24. Note that before "releasing the bus" (i.e., releasing BUSY*), the hold times, etc., of any ending cycle must still be met as described in the previous subsections of this chapter. Likewise, after "taking the bus" (i.e., driving BUSY* low), it is necessary to satisfy all applicable setup and other timing parameters for a cycle just beginning.

3.2.5.1 *Serial Priority.* For a system that uses a serial priority scheme (i.e., daisy-chain BPRO*s to BPRN*s) (see 2.4), the timing specifications in Fig. 25 apply.

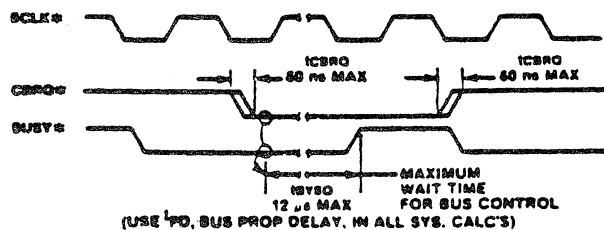


Fig. 24
Common Bus Request AC Timing

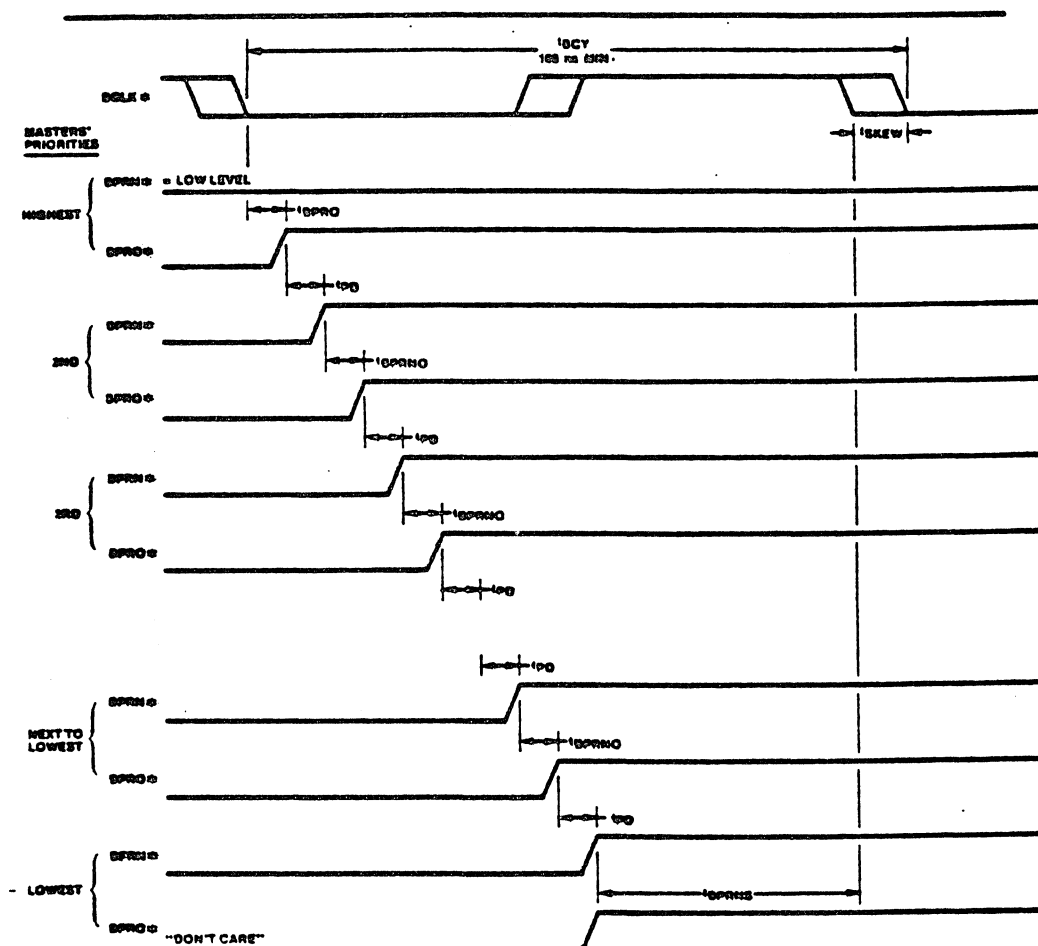


Fig. 25
Serial Priority AC Timing

3.2.5.2 Parallel Priority. For a system that uses a parallel priority scheme (i.e., a central priority resolver) (see 2.4), the following system and CPM (Central Priority Module) timing specifications of Fig. 26 apply.

3.2.6 Miscellaneous Timing. The timing diagrams in Figs. 27, 28, and 29, show the timing of Constant Clock (CCLK^*), Command Separation (t_{CSEP}), and Initialize (t_{INIT}), respectively.

3.3 Receivers, Drivers and Terminations. Non-timing specifications unique to each signal line or to groups of signal lines are presented in Table 3. The requirements for the signal line receivers, drivers, and bus terminations, and the locations of the receiver, driver, and termination for each signal are given.

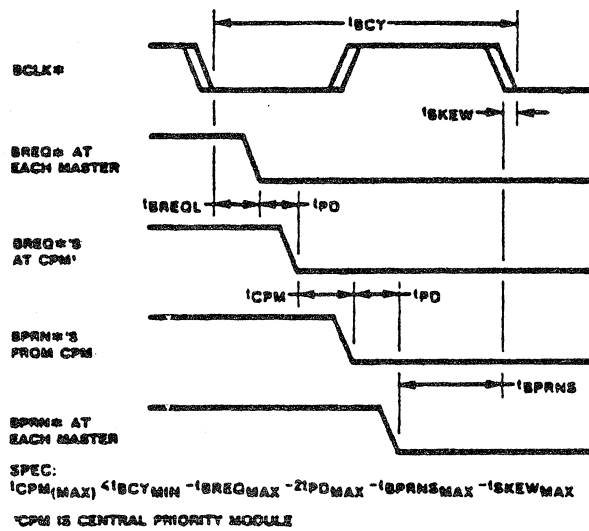


Fig. 26
Parallel Priority AC Timing

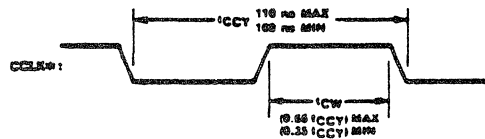


Fig. 27
Constant Clock AC Timing

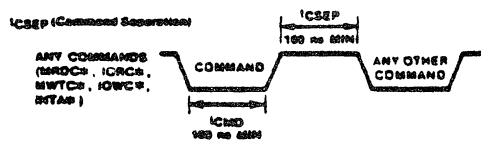


Fig. 28
Command Separation AC Timing

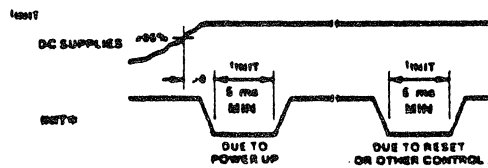


Fig. 29
Initialize AC Timing

Table 3
Bus Drivers, Receivers, and Terminations

Bus Signals	Location	Type	Driver ^{1,3}			C _O Min _{pt}	Location	Receiver ^{2,3}			C _I Max _{pt}	Termination		
			I _{OL} Min _{mA}	I _{OH} Min _{μA}	I _{OH} Max _{μA}			I _{IL} Max _{mA}	I _{IH} Max _{μA}	Location ⁵		Type	R	Units
DAT0* - DATF* - (16 lines)	Masters & Slaves	TRI	16	-2000	—	300	Masters & Slaves	-0.8	125	18	1 place	Pullup	2.2	KΩ
ADR0* - ADR17* , BHEN* (25 lines)	Masters	TRI	16	-2000	—	300	Slaves	-0.8	125	18	1 place	Pullup	2.2	KΩ
MRDC* , MWTC*	Masters	TRI	32	-2000	—	300	Slaves (Memory; Memory- Mapped I/O)	-2	125	18	1 place	Pullup	1	KΩ
IORC* , IOWC*	Masters	TRI	32	-2000	—	300	Slaves (I/O)	-2	125	18	1 place	Pullup	1	KΩ
XACK*	Slaves	TRI	32	-400	—	300	Masters	-2	125	18	1 place	Pullup	510	Ω
INH1* , INH2*	Inhibit- ing Slaves	OC	16	—	-250	300	Inhibited Slaves (RAM, PROM, ROM, Memory- Mapped I/O)	-2	50	18	1 place	Pullup	1	KΩ
BCLK*	1 place (Master usually)	TTL	48	-3000	—	300	Masters	-2	125	18	Mother- board - To +5V To GND	To +5V To GND	220 330	Ω
BREQ*	Each Master	TTL	10	-200	—	60	Central Priority Module	-2	50	18	Central Priority Module (not req)	Pullup	1	KΩ
BPRO*	Each Master	TTL	3.2	-200	—	60	Next Master in Serial Priority Chain at its BPRN*	-1.6	50	18	(not req)			
BPRN*	Parallel: Central Priority Module Serial: Prev Masters BPRO*	TTL	16	-400	—	300	Masters	-4	100	18	(not req)			

Table 3
Bus Drivers, Receivers, and Terminations (Cont'd.)

Bus Signals	Location	Type	Driver ^{1,3}			C _O Min _{pf}	Location	Receiver ^{2,3}		C _I Max _{pf}	Termination			
			I _{OL} Min _{mA}	I _{OH} Min _{mA}	I _{OH} Max _{mA}			I _{IL} Max _{mA}	I _{IH} Max _{mA}		Location ⁴	Type	R	Units
BUSY* CBRQ*	All Masters	OC	20	—	-250	300	All Masters	-2	50	18	1 place	Pullup	1	KΩ
INIT*	Master	OC	32	—	-250	300	All	-2	50	18	1 place	Pullup	2.2	KΩ
CCLK*	1 place	TTL	48	-3000	—	300	Any	-2	125	18	Mother- board	To +5V To GND	220 330	Ω
INTA*	Masters	TRI	32	-2000	—	300	Slaves (Inter- rupting I/O)	-2	125	18	1 place	Pullup	1	KΩ
INT0* - INT7* (8 lines)	Slaves	OC	16	—	-250	300	Masters	-1.6	40	18	1 place	Pullup	1	KΩ

¹Driver Requirements:

I_{OH} = High Output Current Drive
I_{OL} = Low Output Current Drive
C_O = Capacitance Drive Capability
TRI = 3-State Drive
OC = Open Collector Driver
TTL = Totem-pole Driver

²Receiver Requirements:

I_{IH} = High Input Current Load
I_{IL} = Low Input Current Load
C_I = Capacitive Load

³For low and high voltage specifications see 3.1.1.

⁴±5%, ¼W Resistors

⁵All termination resistors specified as "1 place" are typically located on the motherboard.

4. Mechanical Specifications

4.0 Introduction. This section describes all the physical and mechanical specifications that a designer must concern himself with when designing a 796 Bus backplane or when designing printed circuit boards that will plug into the 796 Bus interface.

4.1 Backplane Considerations. This section is a discussion of the things that the designer must consider when designing a 796 Bus backplane.

The maximum length of the backplane connecting modules is 18 inches. Extender boards used within the system will not be supported by the bus unless the overall resulting length of the bus including the extender card is less than the 18-inch maximum.

4.1.1 Board to Board Relationships. The following printed circuit board specifications must be adhered to when designing 796 Bus compatible boards which are to operate in a 0.6-inch board to board spacing backplane.

a. *Board to Board Spacing (L_C)* — center to center of boards when plugged into backplane must be at least 0.6 inches ± 0.02 .

b. *Board Thickness (L_T)* — the typical board thickness is 0.062 ± 0.005 inches.

c. *Component Lead Length (L_L)* — the length of the component leads below the printed circuit board cannot exceed 0.093 inches.

d. *Component Height (L_H)* — the following equation is used to determine the maximum height of the components above the printed circuit board:

$$L_H < L_C - L_T - L_L$$

$$L_H < 0.58" - 0.067" - 0.093"$$

$$L_H < 0.420 \text{ inches (including board warpage)}$$

Electrically conductive components require L_H to be decreased to 0.40 inches.

An example of a typical backplane and the components necessary to implement it are shown in Fig. 32.

This section contains only the mechanical specifications for designing a 796 Bus interface. The designer must also take into consideration the electrical specifications in Section 3.

4.1.2 796 Bus Pin Assignments. Printed circuit boards which are designed to interface to the 796 Bus have two connectors which plug into the backplane. P1 (Primary) and P2 (Auxiliary). Table 4 shows the pin/signal assignments for the P1 connector on the printed circuit boards. Reserved signals on the P1 connector must be bussed as normal signal lines on the backplane. Table 5 shows the pin/signal assignments for the P2 connector on the printed circuit boards. If a backplane is used then the "Reserved and bussed" signals must be bussed as normal signal lines, and the "Reserved but not bussed" signals shall have no connections.

4.2 796 Bus Board Form Factors. Certain 796 Bus characteristics must be taken into consideration when designing printed circuit boards that interface to it. The designer will ensure himself of 796 Bus compatibility if the specifications discussed in this chapter are followed.

4.2.1 Connector Naming and Pin Numbering Standards. The connectors on the printed circuit boards designed for the 796 Bus interface should adhere to the following standards (see Fig. 32).

- a. The connectors on the bus side of the board will be called P1, P2. P1 is the 86 pin main connector, and P2 is the 60 pin auxiliary connector.
- b. Mating connectors on the motherboard (796 Bus) backplane will be called J1, J2, etc.
- c. Pins should be numbered with odd number pins on the component side of the board, and in ascending order when going counterclockwise around the board (as shown in Fig. 33).

4.2.2 Standard Outline of Printed Wiring Boards. Figure 34 shows the standard outline for 796 Bus-compatible boards (Printed Wire Boards and Printed Circuit Boards). The non-bus edge of the board is not restricted. The remainder of the board including connectors P1 and P2 must adhere to the dimensions shown in Fig. 34. Only the basic board's standard vertical height is currently specified.

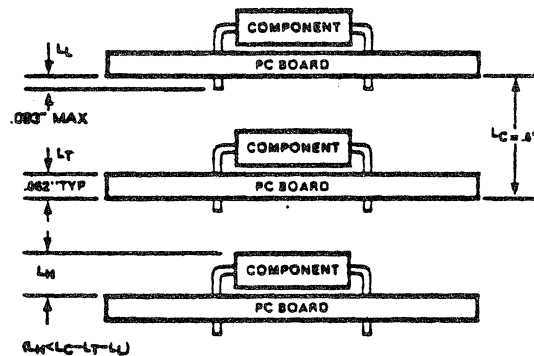


Fig. 31
796 Bus Backplane Card to Card Separation

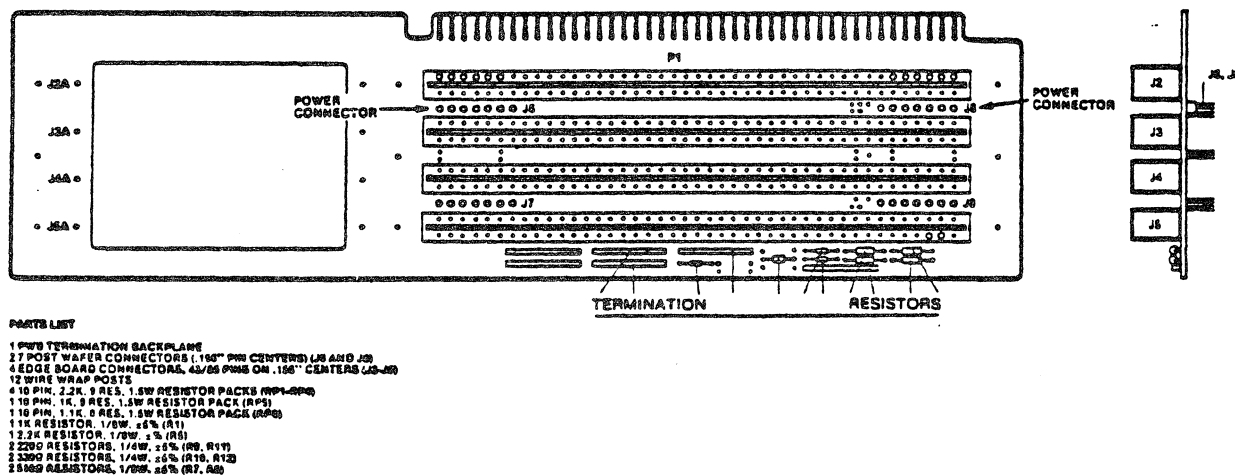


Fig. 32
Typical 796 Bus Backplane

4.2.3 Bus Connectors. The 796 Bus backplane has connectors that mate to the P1 (43/86 pin) board edge connector. The backplane uses 43/86 pin on 0.156" centers connectors.

The P2 connector is a 30/60 pin board edge connector with 0.100" pin centers.

Table 4
Pin Assignment of Bus Signals on 796 Bus Board Connector (P1)

	Pin	(Component Side)		Pin	(Circuit Side)	
		Mnemonic	Description		Mnemonic	Description
Power Supplies	1	GND	Signal GND	2	GND	Signal GND
	3	+5V	+5Vdc	4	+5V	+5Vdc
	5	+5V	+5Vdc	6	+5V	+5Vdc
	7	+12V	+12Vdc	8	+12V	+12Vdc
	9		Reserved, bussed	10		Reserved, bussed
	11	GND	Signal GND	12	GND	Signal GND
Bus Controls	13	BCLK*	Bus Clock	14	INIT*	Initialize
	15	BPRN*	Bus Pri. In	16	BPRO*	Bus Pri. Out
	17	BUSY*	Bus Busy	18	BREQ*	Bus Request
	19	MRDC*	Mem Read Cmd	20	MWTC*	Mem Write Cmd
	21	IORC*	I/O Read Cmd	22	IOWC*	I/O Write Cmd
	23	XACK*	XFER Acknowledge	24	INH1*	Inhibit 1 (disable RAM)
Bus Controls and Address	25		Reserved, bussed	26	INH2*	Inhibit 2 (disable PROM or ROM)
	27	BHEN*	Byte High Enable	28	AD10*	Address Bus
	29	CBRQ*	Common Bus Request	30	AD11*	
	31	COLK*	Constant Clk	32	AD12*	
	33	INTA*	Intr Acknowledge	34	AD13*	
Interrupts	35	INT6*	Parallel Interrupt Requests	36	INT7*	Parallel Interrupt Requests
	37	INT4*		38	INT5*	
	39	INT2*		40	INT3*	
	41	INT0*		42	INT1*	
Address	43	ADRE*	Address Bus	44	ADRF*	Address Bus
	45	ADRC*		46	ADRD*	
	47	ADRA*		48	ADRE*	
	49	ADR8*		50	ADR9*	
	51	ADR6*		52	ADR7*	
	53	ADR4*		54	ADR5*	
	55	ADR2*		56	ADR3*	
	57	ADR0*		58	ADR1*	
Data	59	DATE*	Data Bus	60	DATF*	Data Bus
	61	DATC*		62	DATD*	
	63	DATA*		64	DATB*	
	65	DAT8*		66	DAT9*	
	67	DAT6*		68	DAT7*	
	69	DAT4*		70	DAT5*	
	71	DAT2*		72	DAT3*	
	73	DAT0*		74	DAT1*	
Power Supplies	75	GND	Signal GND	76	GND	Signal GND
	77		Reserved, bussed	78		Reserved, bussed
	79	-12V	-12Vdc	80	-12V	-12Vdc
	81	+5V	+5Vdc	82	+5V	+5Vdc
	83	+5V	+5Vdc	84	+5V	+5Vdc
	85	GND	Signal GND	86	GND	Signal GND

All Reserved pins are reserved for future use and should not be used if upwards compatibility is desired.

Table 5
Pin Assignment of Bus Signals on 796 Bus Board Connector (P2)

	Pin	(Component Side)		Pin	(Circuit Side)	
		Mnemonic	Description		Mnemonic	Description
	1		Reserved, Not Bussed	2		Reserved, Not Bussed
	3		Reserved, Not Bussed	4		Reserved, Not Bussed
	5		Reserved, Not Bussed	6		Reserved, Not Bussed
	7		Reserved, Not Bussed	8		Reserved, Not Bussed
	9		Reserved, Not Bussed	10		Reserved, Not Bussed
	11		Reserved, Not Bussed	12		Reserved, Not Bussed
	13		Reserved, Not Bussed	14		Reserved, Not Bussed
	15		Reserved, Not Bussed	16		Reserved, Not Bussed
	17		Reserved, Not Bussed	18		Reserved, Not Bussed
	19		Reserved, Not Bussed	20		Reserved, Not Bussed
	21		Reserved, Not Bussed	22		Reserved, Not Bussed
	23		Reserved, Not Bussed	24		Reserved, Not Bussed
	25		Reserved, Not Bussed	26		Reserved, Not Bussed
	27		Reserved, Not Bussed	28		Reserved, Not Bussed
	29		Reserved, Not Bussed	30		Reserved, Not Bussed
	31		Reserved, Not Bussed	32		Reserved, Not Bussed
	33		Reserved, Not Bussed	34		Reserved, Not Bussed
	35		Reserved, Not Bussed	36		Reserved, Not Bussed
	37		Reserved, Not Bussed	38		Reserved, Not Bussed
	39		Reserved, Not Bussed	40		Reserved, Not Bussed
	41		Reserved, Bussed	42		Reserved, Bussed
	43		Reserved, Bussed	44		Reserved, Bussed
	45		Reserved, Bussed	46		Reserved, Bussed
	47		Reserved, Bussed	48		Reserved, Bussed
	49		Reserved, Bussed	50		Reserved, Bussed
	51		Reserved, Bussed	52		Reserved, Bussed
	53		Reserved, Bussed	54		Reserved, Bussed
Address	56 58	ADR16* ADR14*	Address Bus	56 58	ADR17* ADR15*	Address Bus
	59		Reserved, Bussed	60		Reserved, Bussed

All Reserved Pins are reserved for future use and should not be used if upwards compatibility is desired.

5. Levels of Compliance

This section presents the concept and notation of levels of compliance with the 796 Bus Standard as follows:

(1) Variable Elements of capability composing the essence of 796 Bus standard compliance.

(2) General discussion of compliance relationship for masters and for slaves.

(3) Notation for describing level of compliance with the 796 Bus Standard.

The notion of levels of compliance is introduced to facilitate the use of 796 Bus products of varying capability manufactured by diverse vendors. It bounds the variability allowed within the 796 Bus specification and provides a succinct and convenient notation for these variables.

5.1 Variable Elements of Capability The 796 Bus is very versatile allowing systems to be constructed with boards of varying capability. The 796 bus allows for variations in data path width, I/O address path width, and interrupt attributes. In addition it is recognized that some vendors products have differing memory address path width.

5.1.1 Data Path The 796 Bus allows for both 8 and 16 bit data path products. The 16 bit data path products use the byte swapping technique described in paragraph 2.2.2.4, thus allowing the 8 and 16 bit products to work together.

5.1.2 Memory Address Path The 796 Bus standard designates a 24 bit address path. In many systems a 16 or 20 bit address path may be sufficient, though not fully 796 Bus compatible.

5.1.3 I/O Address Path The 796 Bus allows for both 8 and 16 bit I/O address paths. The 16 bit path products must also be configurable to act as 8 bit path products.

5.1.4 Interrupt Attributes The 796 Bus (paragraph 2.3) allows for considerable variety in interrupt attributes. A product may support no interrupts, Non Bus Vectored (NBV) interrupts, two cycle bus vectored interrupts, and three cycle bus vectored interrupts. There are two methods of interrupt sensing: the preferred level-triggered; and for historical compatibility only, edge-level-triggered.

Level-Triggered. The active level of the request line indicates an active request. Requiring no edge to trigger an interrupt allows several sources to be attached to a single request line. Sources for level triggered sense inputs should provide a programmatic means to clear the interrupt request.

Edge-Level-Triggered. The transition from the inactive to the active level indicates an active request if and only if the active level is maintained at least until it has been recognized by the master. The requirement for a transition precludes multiple sources on a request line. But, Edge-Level triggering removes the requirement that the source have a programmatic means to clear the interrupt request.

NOTE

Edge-Level-Triggering is described only to allow for historical compatibility.
New designs shall use level-triggered interrupt sensing.

A master may support either or both of the above interrupt sensing methods. It is necessary to configure the system such that the sources of the interrupt requests correspond to the interrupt sensing method of the master. Note that a source which is compatible with Level-Triggering is also compatible with Edge-Level triggering.

5.2 Masters and Slaves When constructing 796 Bus systems it is not necessary that all modules have identical capabilities. One may for instance have a master with an 8/16 bit data path and a slave with an 8 bit data path. The system is completely functional, though the application must restrict itself to 8 bit access to the slave.

The key concept when constructing a 796 Bus system is that of required capability versus supplied capability. Each product will provide some set of capability. A transaction between two such products will be restricted to use that capability which is the intersection of the sets of capability of the two products. In some cases the intersection may be null implying fundamental incompatibility. It is the responsibility of the system designer to assure the viability of this intersection.

5.3 Compliance Level Notation A notation is introduced which allows a vendor to succinctly and accurately specify a product's level of compliance with the 796 Bus standard. For boards which may act as either masters or slaves, the compliance levels must be specified for both cases. Increasing levels of compliance subsume lesser levels for data path width, memory address path width and I/O address path width. Interrupt attributes are listed discreetly as they are independent of one another. The lack of an element (i.e., no I/O address path) specification normally implies no capability for this element. An exception is that Level-Triggered interrupt sensing is to be assumed unless Edge-Level sensing is specified.

5.3.1 Data Path

D8 represents an 8 bit data path

D16 represents an 8/16 bit data path

5.3.2 Memory Address Path

M16 represents a 16 bit memory address path

M20 represents a 20 bit memory address path

M24 represents a 24 bit memory address path

5.3.3 I/O Address Path

I8 represents an 8 bit I/O address path

I16 represents an 8 or 16 bit I/O address path

5.3.4 Interrupt Attributes

V0 represents Non Bus Vectored interrupt requests

V2 represents two cycle bus vectored interrupt requests

V3 represents three cycle bus vectored interrupt requests

E represents Edge-Level triggering only

E+ represents Level or Edge-Level triggering

The interrupt attributes notation can be concatenated to represent multiple capabilities.

5.3.5 *An Example* A versatile combination I/O and memory slave board which supports an 8/16 bit data path, a 20 bit memory address, an 8 or 16 bit I/O address, NBV interrupt requests, two and three cycle bus vectored interrupt requests would be specified as follows:

796 Bus Compliance: Slave D16 M20 I16 VO23

Appendix B
ENVIRONMENTAL CHARACTERISTICS

Appendix C

MANUFACTURING SWITCH AND JUMPER SETTINGS

STANDALONE SYSTEM

- 1) Motherboard jumper plug not installed (non-cluster).
- 2) RAM expansion jumper A-C installed (16K RAM's).
- 3) CPU board jumper TP6 to ground installed (no 8087).
- 4) CPU S1: bit 1 ON - 4 MHz DMA
bit 2 ON - no waits for memory read
bit 3 OFF- not used
bit 4 ON - 5 MHz 8086
- 5) CPU S2: bit 1 OFF- no Multibus to CT window at 512K
bit 2 OFF- not used
bit 3 OFF- not used
bit 4 ON - no Multibus to CT window at 32K
- 6) CPU S3: bit 1 ON - CT to Multibus window starts at 512K
bit 2 OFF- CT to Multibus window starts at 512K
bit 3 ON - CT generates CCLK-
bit 4 ON - CT generates BCLK-
- 7) CPU S4: bit 1 OFF- Up to 512K of CT RAM
bit 2 ON - for 128K RAM, OFF for 256K RAM
bit 3 ON - for 128K or 256K RAM
bit 4 ON - for 128K or 256K RAM
- 8) CPU S5: bit 1 OFF- 2716 PROMS
bit 2 ON - 2716 PROMS
bit 3 OFF- not used
bit 4 OFF- not used
- 9) CPU S6: bit 1 OFF- less than 840K of RAM
bit 2 ON - less than 840K of RAM
bit 3 OFF- Up to 512K of CT RAM
bit 4 ON - CBRQ always asserted
- 10) I/O Memory jumper E-F installed (16K RAMs)
- 11) I/O Memory jumper plugs JP2 and JP4 installed, JP1 and JP3 not installed. (Channel A - RS232)
- 12) I/O Memory S1: bit 1 OFF - Channel A internal clocking
bit 2 OFF - Channel A internal clocking
bit 3 ON - Channel A internal clocking

bit 4 ON - Channel A internal clocking
bit 5 OFF- Channel B internal clocking
bit 6 OFF- Channel B internal clocking
bit 7 ON - Channel B internal clocking
bit 8 ON - Channel B internal clocking

13) I/O Memory S2: all bits ON - port 44=0, not a workstation

14) I/O Memory S3: bit 1 OFF - External Carrier Detect
bit 2 ON - External Carrier Detect
bit 3 OFF - not used
bit 4 OFF - not used

USER'S COMMENT SHEET

Workstation Hardware Manual
EW-100 A-09-00016-01-B

We welcome your comments and suggestions. They help us improve our manuals. Please give specific page and paragraph references whenever possible.

Does this manual provide the information you need? Is it at the right level? What other types of manuals are needed?

Is this manual written clearly? What is unclear?

Is the format of this manual convenient in arrangement, in size?

Is this manual accurate? What is inaccurate?

Name _____ Date _____
Title _____ Phone _____
Company Name/Department _____
Address _____
City _____ State _____ Zip Code _____

☐ Please check here if you'd like a reply.

Thank you.
All comments become the property of Convergent Technologies.

Seal or tape for mailing - do not use staples
fold



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 1309 SANTA CLARA, CA.

POSTAGE WILL BE PAID BY -

Convergent TechnologiesTM

2500 Augustine Drive
Santa Clara, Ca. 95051

ATTN: TECHNICAL PUBLICATIONS

fold

cut along here



Convergent Technologies

2500 Augustine Drive, Santa Clara, CA 95051 • (408) 727- 8830

Printed in U.S.A.