INTELLEC® SERIES II MICROCOMPUTER DEVELOPMENT SYSTEM HARDWARE REFERENCE MANUAL

Manual Order No. 9800556-02 Rev. B

Intel Corporation, 3065 Bowers Avenue, Santa Clara, California 95051

Additional copies of this manual or other Intel literature may be obtained from:

Literature Department
Intel Corporation
3065 Bowers Avenue
Santa Clara, CA 95051

The information in this document is subject to change without notice.

Intel Corporation makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Intel Corporation assumes no responsibility for any errors that may appear in this document. Intel Corporation makes no commitment to update nor to keep current the information contained in this document.

Intel Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in an Intel product. No other circuit patent licenses are implied.

The following are trademarks of Intel Corporation and may be used only to identify Intel products:

BXP	Intellec	Multibus
i	iSBC	Multimodule
ICE	iSBX	PROMPT
iCS	Library Manager	Promware
Insite	MCS	RMX
Intel	Megachassis	UPI
Intelevision	Micromap	μScope

and the combination of ICE, iCS, iSBC, iSBX, MCS, or RMX and a numerical suffix.

A569/382/ 2K DD

PREFACE

This manual describes hardware and software elements of the Intellec Series II Microcomputer Development Systems as necessary to delineate internal processes and data flow. The information is intended to support hardware troubleshooting and to provide a basis for understanding the development system's resources.

Information within the six chapters of this manual is as follows:

- *General Information*—Descriptions of the three basic models (Model 220, 225 and 230) and optional hardware and software features. An overview of the functional organization and a review of the primary intersystem communications path (the Multibus interface).
- *Integrated Processor Board (IPB)*—A technical description of the circuit elements comprising the 8080-based data processing and control center of the Intellec Series II development system.
- *Integrated Processor Card (IPC)*—A technical description of the circuit elements comprising the 8085-based data processing and control center of the Intellec Series II development system.
- *Input/Output Controller (IOC)*—A technical description of the circuit elements that establish an intelligent interface between the IPB or IPC and the integral CRT and diskette drive.
- *Parallel Input-Output (PIO) Subsystem*—A technical description of the circuit elements that establish an intelligent interface between the IPB or IPC and the standard Intellec Series II development system peripherals (paper tape reader/punch, printer and PROM programmer).
- *Keyboard Assembly*—A technical description of the circuit elements that detect and decode keystrokes and keystroke combinations entered from the keyboard.

Information within this manual describes LSI and TTL chips only to the extent necessary to support troubleshooting to the circuit board level. Further details on the chips are provided by chip manufacturer literature including the following Intel manuals:

- *Intel Component Data Catalogue*
- *MCS-80/85 Family User's Manual*, Order Number 121506

Although the information herein is intended to support hardware maintenance, the manual does not contain maintenance procedures or guidelines. Maintenance information is provided in the following manuals:

- *Intellec Series II Microcomputer Development System Schematic Drawings*, Order Number 9800554
- *Intellec Series II Microcomputer Development Systems Models 220, 221, 222 and Models 230, 231, 232 Service Information*, Order Number 9800878
- *Intellec Series II Microcomputer Development Systems Models 225, 226, 227 Service Information*, Order Number 121569

Intellec Series II is compatible with a variety of optional peripheral devices and circuit boards. Technical details on the Intellec Series II development system interface with any such device or board are provided in the associated technical manual. Non-standard devices and boards, which require modification of the Intellec Series II development system hardware or software, void the development system warranty

unless prior written approval is obtained from Intel Corporation. Guidelines for the use of non- standard devices and modifications to meet unique user needs are provided in the following manual:

Intellec Series II Microcomputer Development System Hardware Interface Manual, Order Number 9800555

Other manuals that support use of the Intellec Series II development systems are:

- *ISIS-II User's Guide*, Order Number 9800306
- *Intellec Series II Model 22X/23X Installation Manual*, Order Number 9800559
- *Intellec Series II Microcomputer Development Sytem Monitor Listings*, Order Number 9800605

CONTENTS

TABLES

ILLUSTRATIONS

CHAPTER 1
GENERAL INFORMATION

The Intellec Series II Microcomputer Development Systems are bus-oriented, multiprocessor systems that employ ROM-based as well as diskette-based software. Each development system within the series provides both parallel and serial I/O interfaces to support a variety of external peripherals. Both hardware and software options are available to tailor the Intellec Series II development system to meet the needs of users engaged in the design and implementation of microcomputer-based systems that contain Intel LSI products. These options enhance user-program generation and simplify software debugging and user-hardware troubleshooting.

1.1 Development System Models

There are three basic development system models in the series: the Model 220, the Model 225 and the Model 230. Additionally, there are two variations of each of the basic models according to the operating voltage configuration. For example, a basic Model 220 development system that is configured at the factory for 115 volt operation is designated a Model 220, while a basic Model 220 that is configured for 230 volt operation is designated a Model 221. All models in the series feature an integral video display and an attached keyboard, an integral power supply, a six-slot Multibus-compatible card cage and either one or two flexible disk drives. The development system itself is made up of three microprocessor-based computing elements that are contained on two printed circuit board assemblies. One assembly (either an Integrated Processor Board or an Integrated Processor Card) is inserted into the uppermost slot of the card cage and incorporates the master processor. The other assembly (the Input/Output Controller) is mounted on the inside of the rear panel. This assembly contains the Input/Output Controller (IOC) processor and the Parallel Input/Output (PIO) subsystem processor. Both the Model 220 and the Model 225 include an integral single-density diskette drive (see figure 1-1) and differ only by the circuit board assembly installed in the card cage (the Model 220 uses the 8080-based Integrated Processor Board or "IPB," and the Model 225 uses the 8085-based Integrated Processor Card or "IPC"). The Model 230 (see figure 1-2) is supplied with a separate chassis that contains two double-density diskette drives in lieu of the integral diskette drive common to the Models 220 and 225. The card cage of the Model 230 includes an 8080-based IPB, a 32k RAM board and a two-board double-density diskette controller (to support the two double-density diskette drives).

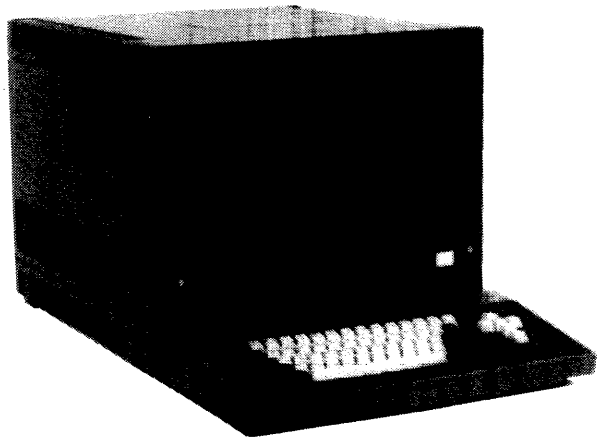


Figure 1-1. Models 220 and 225 556-1

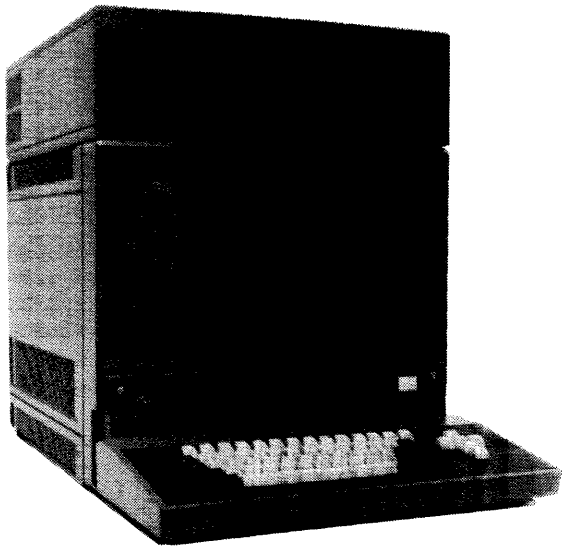


Figure 1-2. Model 230 556-2

All models of the Intellec Series II development systems are supplied with the following software on diskette:

- Intel Systems Implementation Supervisor (ISIS) disk operating system
- 8080/8085 Assembler

Additionally, the following ROM-resident software is supplied:

- Monitor program
- Bootstrap/Diagnostic program

The Models 225 and 230 also include Intel's CRT-Based Text Editor (CREDIT) on diskette.

1.2 Functional Organization

The Intellec Series II Microcomputer Development System is made up of three distinct computing elements: the IPB or IPC, the IOC and the PIO subsystem. Each element includes its own microprocessor and sufficient memory and I/O facilities to perform real-time computing tasks. The programs executed by the three elements limit their respective roles and establish mutual dependence so that the three elements operate together as a single integrated system.

The IPB or IPC is the master processor for the development system. As the master processor, the IPB/IPC alone executes the operating system (Monitor or ISIS), the support programs (assembler, compilers, in-circuit emulator drivers, etc.) and the user's program. In communicating with most I/O devices, the IPB/IPC is not concerned with the timing and format requirements of the device. With the exception of the two serial I/O channels, the IPB/IPC merely directs the IOC or the PIO subsystem to transfer one or more bytes to or from the associated peripheral device.

Control information from the IPB/IPC to the PIO or IOC subsystem is provided in the form of single-byte commands, each of which may or may not be associated with a data byte. Commands without a data byte are used to control operations and functions (i.e., initialization, enabling/disabling interrupts, service request acknowledgements etc.). A data byte associated with a command may be the actual data to or from the device, a status return, or a control parameter required by a device controller prior to a data transfer (e.g., the starting address or byte count of a forthcoming diskette transfer).

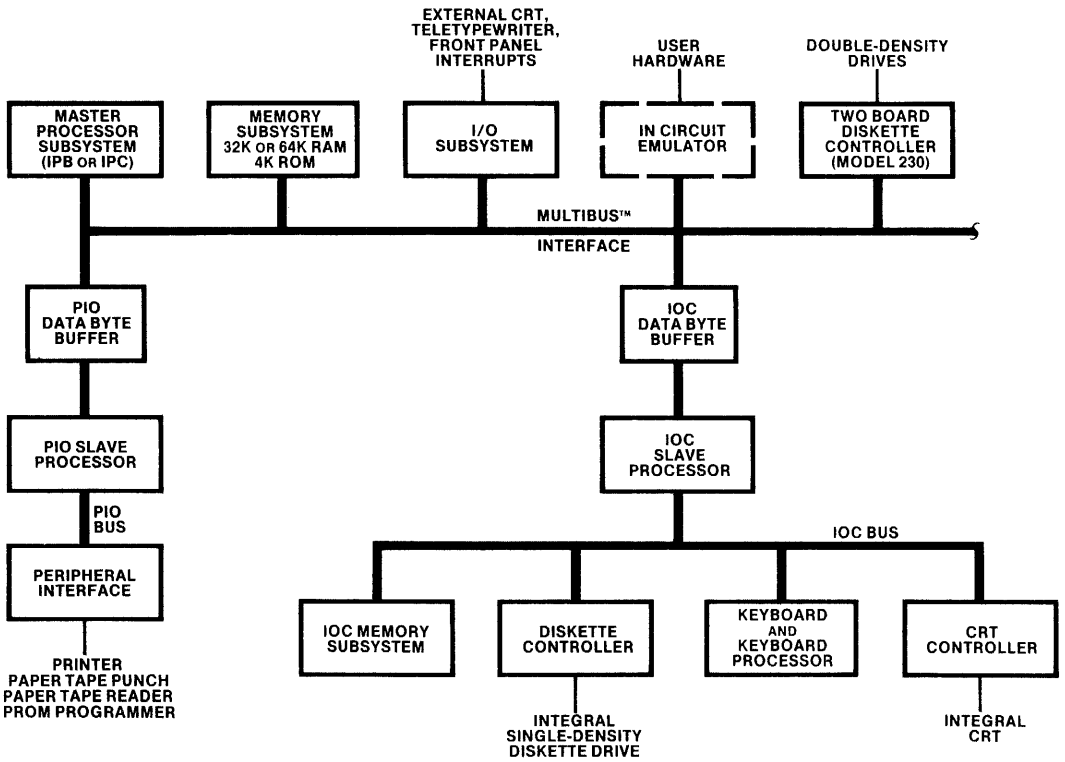


Figure 1-3. Intellec Series II Microcomputer Development System Block Diagram 556-3

The IOC and PIO subsystem are generally inactive except when responding to a command from the IPB/IPC. Any valid command from the IPB/IPC is accepted without reservation (provided any previous command has been completed) and is immediately processed. During command execution, both the IOC and PIO subsystem revise their respective data byte buffer (DBB) flags to indicate acceptance of the command, acceptance of any associated data byte or the completion of command processing. The operating system (Monitor or ISIS) polls the DBB flags to determine the status of command execution by the IOC or PIO subsystem. Note that the only byte returned to the IPB/IPC as a status byte is the DBB flag byte. All other status information from the IOC or the PIO subsystem is returned as a data byte that is recognized as status only by software. The same is true for commands; the only true command is a directive to the IOC or PIO subsystem. Parameters sent to device controllers are output as data bytes and are recognized as parameters only by the IOC or PIO subsystem software.

Few interrupts are implemented by the Intellec Series II development system hardware; interrupts that are implemented are normally not enabled by the system software (Monitor or ISIS). The IPB/IPC does, however, make general use of service requests that appear as bits within status bytes returned from the IOC or PIO subsystem. The service requests are often called interrupts since they inform the master processor that a task initiated by a command has been completed. However, it is important to remember that service requests are returned when the master processor polls the IOC or PIO subsystem to determine its status; true interrupts are initiated by hardware without polling.

It is not necessary for the master processor to continually poll the IOC and PIO to detect the occurrence of most hardware events since these events are generally initiated by commands from the IPB/IPC. In other words, polling after each command is adequate for most devices. However the keyboard of the IOC (or a keyboard associated with either serial I/O channel) is an exception in that key entries are not dependent on IPB/IPC commands. Another exception is any operator-controlled device connected to the PROM programmer connector of the PIO subsystem. The master processor periodically polls both the IOC and the PIO to access status bytes containing service requests.

1.3 Multibus Architecture

As previously stated, either an IPB or IPC assembly is inserted into the top slot of the development system's card cage and, in the case of the Model 230, a two-board diskette drive controller and a 32k RAM board are also inserted into the card cage. The backplane board of the card cage is designed for compatibility with the Intel Multibus architecture. This architecture is represented by a collection of 86 lines that provides a communications path among all bus-compatible boards installed in the card cage. All Intel single board computers and associated memory, peripheral devices and peripheral device controllers adhere to the Multibus architecture.

The Multibus architecture permits any master processor installed in the backplane to assume full control of the bus. In assuming control of the bus, each bus master has access to the resources of any other bus master. In terms of the Intellec Series II development system, another bus master can add resources and features unavailable in the standard system. In terms of another bus master, the development system's resources are available to the bus master and do not have to be duplicated. The available development system resources include all of the IPB/IPC RAM, Monitor ROM, the entire IOC with its subordinate CRT and integral diskette drive, and the entire PIO subsystem with its subordinate paper tape reader/punch, line printer and PROM programmer peripherals. To use the resources of the development system, the bus master must use the memory and I/O port addresses assigned by the Intellec Series II development system as described in this manual.

A few facilities of the development system that are implemented on both the IPB and the IPC are inaccessible to another bus master. On the IPB, these facilities include the bus and system controllers, the control port (port FF), both serial I/O channels, the interval timer and both interrupt controllers. On the IPC, only the bus controller, control port and the systm interrupt controller are inaccessible to another bus master.

Table 1-1 defines the Multibus interface pin assignments, and table 1-2 describes the individual signals.

1.4 Options

Numerous options are available for the Intellec Series II Microcomputer Development Systems to expand a system's resources and functions. Several options are also available to "upgrade" an existing model to a model with increased resources. (For a complete list of available options, refer to the Intel *Systems Data Catalogue*.)

Table 1-1. Multibus Interface Pin Assignments

Board Component Side			Board Circuit Side		
Pin	Mnemonic	Description	Pin	Mnemonic	Description
1	GND	Signal ground	2	GND	Signal ground
3	+5	+5 VDC	4	+5	+5 VDC
5	+5	+5 VDC	6	+5	+5 VDC
7	+12	+12 VDC	8	+12	+12 VDC
9	−5	−5 VDC	10	−5	−5 VDC
11	GND	Signal ground	12	GND	Signal ground
13	BCLK/	Bus Clock	14	INIT/	Initialize
15	BPRN/	Bus Priority In	16	BPRO/*	Bus Priority Out
17	BUSY/	Bus Busy	18	BREQ/	Bus Request
19	MRDC/	Memory Read Command	20	MWTC/	Memory Write Command
21	IORC/	I/O Read Command	22	IOWC/	I/O Write Command
23	XACK/	Transfer Acknowledge	24	INH1/	Inhibit (disable) RAM
25	AACK/	Advanced Acknowledge	26	INH2/	Inhibit (disable) ROM
27	BHEN/	Byte High Enable	28	ADR10/	Address Extension Lines
29	CBRQ/**	Common Bus Request	30	ADR11/	
31	CCLK/	Constant Clock	32	ADR12/	
33	INTA/*	Interrupt Acknowledge	34	ADR13/	
35	INT6/	Interrupt Requests	36	INT7/	Interrupt Requests
37	INT4/		38	INT5/	
39	INT2/		40	INT3/	
41	INT0/		42	INT1/	
43	ADRE/	Address Lines	44	ADRF/	Address Lines
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/	
59	DATE/	Data Lines	60	DATF/	Data Lines
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/	
75	GND	Signal ground	76	GND	Signal ground
77	−10	−10 VDC	78	−10	−10 VDC
79	−12	−12 VDC	80	−12	−12 VDC
81	+5	+5 VDC	82	+5	+5 VDC
83	+5	+5 VDC	84	+5	+5 VDC
85	GND	Signal ground	86	GND	Signal ground

* Not implemented on IPB/IPC
** Only implemented on IPC

Table 1-2. Multibus Interface Signal Descriptions

Signal Mnemonic	Functional Description
AACK/	*Advanced Acknowledge.* A memory acknowledge signal that is generated in advance of the normal transfer acknowledge (XACK/) signal to allow a bus master to shorten its memory access timing.
ADR0/-ADRF/	*Address.* These 16 lines are used to specify the address of the memory location or I/O port to be accesed within a 64k range. ADRF/ is the least significant bit.
ADR10/-ADR13/	*Extended Address.* These four lines are appended to the 16-bit address to increase the addressing range to 1 megabyte. The IPB/IPC monitors these inputs to determine if the address on the bus corresponds to its memory segment (0-64k). ADR10/ is the least significant bit of the extended address.
BCLK/	*Bus Clock.* An asynchronous clock signal used to synchronize bus contention resolving circuits among bus masters. The BLCK/ signal from the IPB/IPC has a minimum period of 100 ns with a 35%-65% (minimum) duty cycle.
BHEN/	*Byte High Enable.* A signal that indicates the presence of a data byte on the DAT8/-DATF/ data lines. This signal is supported only by the memory on the IPC.
BPRN/	*Bus Priority In.* A bus contention resolving signal that indicates to a bus master that no higher-priority bus master is requesting the bus. The IPB/IPC incorporates the bus resolution logic and provides an individual BPRN/ signal to each card-cage slot. BPRN/ is synchronized by BCLK/.
BPRO/	*Bus Priority Out.* A bus contention resolving signal for use with serial (daisy chain) bus priority resolution schemes (BPRO/ is passed to the BPRN/ input of the next lower-priority bus master). BPRO/ is not supported by the IPB/IPC.
BREQ/	*Bus Request.* A bus contention resolving signal for use with parallel-priority resolution schemes. A bus master uses BREQ/ to indicate that it requires the bus. The IPB/IPC monitors the individual BREQ/ inputs and outputs BPRN/ to the highest-priority bus master requesting the bus. BREQ is synchronized by BCLK/.
BUSY/	*Bus Busy.* A common bus contention resolving signal to indicates that the bus is in use. When a bus master gains access the bus, it uses BUSY/ to prevent any other bus master from acquiring the bus. BUSY/ is synchronized by BCLK/.
CBRQ/	*Common Bus Request.* A bus contention resolving signal to allow bus requests by lower-priority bus masters to override bus requests by high-priority bus masters. CBRQ/ is synchronized by BCLK/ and is supported only by the IPC.
CCLK/	*Constant Clock.* An asynchronous clock signal available for general use by modules on the bus. On the IPB/IPC, CCLK/ is synchronous with BCLK/ and has the same period and duty cycle.
DAT0/-DATF/	*Data.* These 16 bidirectional lines transfer data to or from the addressed memory location or I/O port. DAT0/ is the least-significant bit. The IPB/IPC, as a bus master, has a 8-bit wide data path and only uses DAT0/-DAT7/. The IPC memory can use all 16 data lines (DAT0/-DATF/).
INH1/	*Inhibit RAM.* Prevents RAM devices from responding to the memory address on the address lines. INH1/ effectively allows ROM devices to override RAM devices when ROM and RAM are assigned the same address space. INH1/ may also be used to allow memory mapped I/O devices to override RAM.
INH2/	*Inhibit ROM.* Prevents ROM devices from responding to the memory address on the address lines. INH2/ effectively allows start-up software such as ROM-based bootstrap programs to override another ROM device when the two ROMs are assigned the same address space. INH2/ may also be used to allow memory mapped I/O devices to override ROM.
INIT/	*Initialize.* This signal resets the entire system to a known internal state. INIT/ is activated by the front panel RESET switch. When activated by another bus master, INIT/ will not initialize the IPB/IPC.

Table 1-2. Multibus Interface Signal Descriptions (Cont'd.)

Signal Mnemonic	Functional Description
INT0/-INT7/	*Interrupt.* A set of eight, multilevel interrupt request lines accepted by the parallel interrupt resolving logic of the IPB/IPC (INT0/ is the highest-priority interrupt). In response to an interrupt, the IPB/IPC executes an interrupt acknowledge sequence to call the associated user interrupt service routine.
INTA/	*Interrupt Acknowledge.* A signal generated by a master processor in response to an interrupt. Since the IPB/IPC incorporates the interrupt resolution logic, the INTA/ signal is used internally and is not output on the bus.
IORC/	*I/O Read Command.* This signal indicates that the address of an input port has been placed on the address lines and that the port is to place the requested data byte on the data lines.
IOWC/	*I/O Write Command.* This signal indicates that the address of an output port has been placed on the address lines and that the data byte on the data lines is to be accepted by the port.
MRDC/	*Memory Read Command.* Indicates that the address of a memory location has been placed on the address lines and that the requested data byte is to be placed on the data lines.
MWTC/	*Memory Write Command.* Indicates that the address of a memory location has been placed on the address lines and that the data byte to be written into the addressed location has been placed on the data lines.
XACK/	*Transfer Acknowledge.* This signal is the response from memory or an I/O port that indicates that the specified read/write operation has been completed. That is, data has been placed on, or accepted from, the data lines.

1.4.1 Expansion Chassis

A four-slot expansion chassis is available to increase the card cage capacity of any model from six slots to ten slots. The expansion chassis is mounted below the system chassis and includes the required bus interface cables and its own internal power supply.

1.4.2 Universal PROM Programmer

The Intel Universal PROM Programmer (UPP-103) allows a user to program and verify any of the standard Intel PROMs and EPROMs. The PROM programmer interfaces directly to the UPP connector on the rear panel of the system chassis and includes the required interface cable and its own internal power supply. The PROM programmer is supplied with two, user-selected personality modules and a diskette that contains the Universal PROM Mapper (UPM) control program.

1.4.3 In-Circuit Emulators

A number of in-circuit emulators and microcomputer development systems support packages are available to support Intel microcomputer and microprocessor family devices. In-circuit emulators simplify hardware troubleshooting and program debugging in a user's system by emulating the system's microcomputer or microprocessor in real time. In-circuit emulators are available for the MCS-48 family microcomputers and the 8080, 8085 and 8086 family microprocessors.

1.4.4 High-Level Languages

Several high-level languages are available (on diskette) for the Intellec Series II development systems including a PL/M compiler, a FORTRAN compiler, PASCAL compiler and a BASIC interpreter.

1.4.5 IPC Upgrade Kit

The IPC upgrade kit replaces the 8080-based IPB assembly in a Model 220 or Model 230 with an 8085-based IPC assembly as supplied with the Model 225.

1.4.6 Disk Drives

Three disk drive options are available to increase the on-line storage capacity of any development system. A complete double-density diskette drive system as well as a dual double-density drive chassis are available. Both of these options add 1 megabyte of on-line storage to an existing system. The 720 Option consists of a two-board flexible disk drive controller set and a dual drive chassis, while the 730 Option includes only the dual drive chassis. The Option 740 is a complete hard-disk subsystem (two-board hard disk drive controller set and drive) that provides an additional 7.3 megabytes of storage. Table 1-3 defines the drive option applicability for the three development system models.

Table 1-3. Drive Option Applicability

Option Number	Model Number		
	220	225	230
720	X	X	Standard
730	X*	X*	X
740	X	X	X

* Must also have 720 option

1.5 Specifications

Table 1-4 lists the specifications for the three basic models of the Intellec Series II Microcomputer Development Systems. Except where noted, the specifications in table 1-4 are applicable to all models. Table 1-5 lists the system mainframe current requirements and the reserve current available for options in both the system mainframe and expansion chassis.

Table 1-4. Intellec Series II Microcomputer Development System Specifications

IPB Master Processor (Models 220, 221, 230, and 231)	
Microprocessor:	8080A-2 operating at 2.6 MHz.
RAM:	32k (see note).
ROM:	4k (2k in monitor, 2k in boot/diagnostic).
Bus:	Multibus architecture; bus clock rate is 9.8304 MHz.
IPC Master Processor (Models 225 and 226)	
Microprocessor:	8085A-2 operating at 4.0 MHz.
RAM:	64k.
ROM:	4k (2k in monitor, 2k in boot/diagnostic).
Bus:	Multibus architecture; bus clock rate is 9.8304 MHz.
Interrupts:	8-level, maskable, nested priority interrupt network initiated from front panel or user-selected devices.
I/O Interfaces	
Serial:	Two RS232 channels at 110-9600 baud (asynchronous) or 150-56,000 baud (synchronous); programmable baud rates and serial formats. Serial Channel 1 provided with 20 mA current loop.
Parallel:	Interface provided for paper tape punch, paper tape reader, line printer, and UPP-103 Universal PROM Programmer.
Direct Memory Access (DMA):	Standard capability of Multibus architecture; implemented for user-selected DMA devices through optional DMA module. Maximum transfer rate of 2 MHz.
Diskette Subsystem (Models 220, 221, 225 and 226)	
Number of Drives:	One, single density.
Storage Capacity:	250k bytes (formatted).
Transfer Rate:	250k bits/second.
Access Time:	Track-to-track: 10 ms. Head settling time: 10 ms.
Average Random Positioning:	260 ms.
Rotational Speed:	360 rpm.
Average Rotational Latency:	83 ms.
Recording Mode:	FM.

Table 1-4. Intellec Series II Microcomputer Development System Specifications (Cont'd.)

Diskette Subsystem (Models 230 and 231)	
Number of Drives:	Two, double density.
Storage Capacity:	1M bytes (formatted) total.
Transfer Rate:	500k bits/second.
Access Time:	Track-to-track: 10 ms. Head settling time: 10 ms.
Average Random Positioning:	260 ms.
Rotational Speed:	360 rpm.
Average Rotational Latency:	83 ms.
Recording Mode:	M^2FM.
AC Requirements	
Input Voltage:	100/120/220/240 Vac ±10%, 47-63 Hz, single phase.
Input Current:	Model 220 and 225: 5.9A Model 221 and 226: 3.1A Model 230: 5.4A Model 231: 2.7A
Electrical Characteristics:	See table 1-5.
Environmental Characteristics	
Operating Temperature:	32° to 95°F (0° to 35°C).
Relative Humidity:	To 90% without condensation.
Physical Characteristics	
Models 220, 221, 225 and 226 Main Chassis:	Width: 17.37 in. (44.12 cm). Height: 15.81 in. (40.16 cm). Depth: 19.13 in. (48.59 cm). Weight: 86 lb (39 kg).
Models 230 and 231 Main Chassis:	Width: 17.37 in. (44.12 cm). Height: 15.81 in. (40.16 cm). Depth: 19.13 in. (48.59 cm). Weight: 73 lb (33 kg).
Dual Drive Chassis (Models 230 and 231):	Width: 17.6 in. (44.7 cm). Height: 5.7 in. (14.7 cm). Depth: 19.4 in. (49.3 cm). Weight: 43.0 lb (19.5 kg).
Keyboard:	Width: 17.37 in. (44.12 cm). Height: 3.0 in. (7.62 cm). Depth: 9.0 in. (22.86 cm). Weight: 6.0 lb (2.72 kg).

Note: The IPB contains 32k of RAM. The Model 230 and 231 provide 64k of RAM by including a 32k RAM board in card cage.

Table 1-5. System Chassis and Optional Expansion Chassis Power Supply Current Capabilities

System Component	Power Supply					
	+5V	+12V	−12V	−10V	+15V	+24V
MODELS 220 and 221						
Power Supply Capacity	30.0A	2.5A	0.3A	1.0A	1.5A	1.7A
IPB	4.0A	0.3A	0.1A	0.01A	—	—
IOC	2.8A	0.1A	—	0.01A	—	—
CRT	—	—	—	—	1.5A	—
Keyboard	0.4A	—	—	—	—	—
Diskette Drive	1.0A	—	—	—	—	1.7A
Total Current Drain	8.2A	0.4A	0.1A	0.02A	1.5A	1.7A
Available for Options	21.8A	2.1A	0.2A	0.98A	0	0
MODELS 225 and 226						
Power Supply Capacity	30.0A	2.5A	0.3A	1.0A	1.5A	1.7A
IPC	4.3A	1.4A	0.2A	0.02A	—	—
IOC	2.8A	0.1A	—	0.01A	—	—
CRT	—	—	—	—	1.5A	—
Keyboard	0.4A	—	—	—	—	—
Diskette Drive	1.0A	—	—	—	—	1.7A
Total Current Drain	8.5A	1.5A	0.2A	0.03A	1.5A	1.7A
Available for Options	21.5A	1.0A	0.1A	0.97A	0	0
MODELS 230 and 231						
Power Supply Capacity	30.0A	2.5A	0.3A	1.0A	1.5A	1.7A
IPB	4.0A	0.3A	0.1A	0.01A	—	—
IOC	2.8A	0.1A	—	0.01A	—	—
CRT	—	—	—	—	1.5A	—
Keyboard	0.4A	—	—	—	—	—
Diskette Controller	5.25A	—	—	0.1A	—	—
iSBC 032 32K RAM Board	2.0A	0.4A	—	0.05A	—	—
Total Current Drain	14.45A	0.8A	0.1A	0.17A	1.5A	0
Available for Options	15.55A	1.7A	0.2A	0.83A	0	1.7A
EXPANSION CHASSIS (OPTIONAL)						
Available for Options	20.0A	2.0A	0.3A	0.8A	N/A	N/A

CHAPTER 2
INTEGRATED PROCESSOR BOARD

The Integrated Processor Board (IPB) is an 8080A-based single board computer system for the Intellec Series II microcomputer development systems. The IPB, in addition to the 8080A microprocessor and associated control logic, includes 32k bytes of random access memory, 4k bytes of read only memory and two serial input/output channels as well as the interfaces to the Multibus, the Input/Output Controller (IOC) and the Parallel Input/Output (PIO) subsystem. Figure 2-1 illustrates the simplified block diagram for the IPB.

IPB communications with the associated system devices are accomplished over the IOC/PIO bus or the Multibus. The IOC/PIO bus is controlled either directly by the IPB microprocessor or indirectly through the IPB by another bus master; the Multibus is controlled either by the IPB microprocessor (when the IPB is the active bus master) or by any other active bus master on the Multibus (the IPB incorporates the required Multibus priority resolution logic). System data transfers are initiated either by hardware interrupts or by the polling of service requests through dedicated I/O ports. As boards or devices are added to the system, additional I/O ports and service requests or interrupts are assigned. The software executed by the IPB provides the ability to expand the system's resources without the necessity of hardware modification. This expansion capability makes the IPB a true general-purpose computing module.

Organization of the Multibus is such that the IPB microprocessor has the lowest priority of the ten bus masters that can be interfaced to the Multibus. The resources of the IPB are accessible to another bus master through the Multibus provided that the bus master uses the I/O ports defined by the IPB. The ROM that contains the bootstrap and diagnostic programs, since it is enabled through a control port accessible only to the IPB, is not directly accessible to any other bus master (access to the ROM-resident monitor program is unrestricted).

Discussions within this chapter describe the functioning of the IPB's 8080A-2 as both the master processor and as the one essential bus master. The functioning of other bus masters is discussed in their corresponding manuals and in the *Intellec Series II Hardware Interface Manual.*

In the remainder of this chapter, the sheet numbers called out in both the text and on the illustrations are references to individual sheets of the IPB schematic diagram included in the *Intellec Series II Schematic Drawings*.

2.1 MASTER PROCESSOR SUBSYSTEM

The IPB master processor subsystem is the computation and control center of the IPB. It also contains logic elements that establish and control the Multibus. Major elements of the subsystem are shown in figure 2-2. The 8080A-2 CPU, the 8224 clock generator, the 8226 data driver, the 8228 system controller and the 8218 bus controller are standard Intel LSI products. The remaining circuit elements are off-the-shelf TTL products.

2.1.1 MASTER PROCESSOR AND CLOCK GENERATOR

The 8080A-2 master processor and its asociated 8224 clock generator (IPB schematic, sheet 3) are employed in a conventional manner except that one wait state is inserted into each memory access machine cycle. To compensate for the wait state, the master processor uses a 2.6 MHz clock, and the average instruction is executed in the same amount of time that would be required if the master processor employed a 2.0 MHz clock without a wait state.

To insert the wait state, the RAM access logic (subsection 2.2.1) provides an acknowledge signal (RAM AACK) to the processor clock generator's ready input. The synchronized RDY output from the clock generator is routed to the 8080's RDY input. The timing of RAM AACK is such that when the 8080 first samples its RDY input (during state T_2), RDY is inactive, and a wait state is inserted into the machine cycle. When the RDY input is again sampled (during T_{WAIT}), it is active, and the 8080 exits the wait state and enters T_3. The other acknowledge signals at the processor clock generator's ready input (XACK/, AACK/, LOC ACK and TO ACK) are used to insert one or more wait states into machine cycles when required by internal I/O transfers or external I/O or memory access operations. LOC ACK (local acknowledge) originates from the IPB's local acknowledge timer and is used with local I/O transfers (transfers between the serial I/O channels, the programmable devices of the IPB, the IOC or the PIO subsystem). TO ACK (timeout acknowledge) is used to allow the 8080 to complete a machine cycle when the expected command acknowledge is not received within approximately 10 milliseconds. One additional signal at the ready input is INTA/ (interrupt ackowledge). This signal provides the required ready indication to the 8085 during interrupt acknowledge machine cycles.

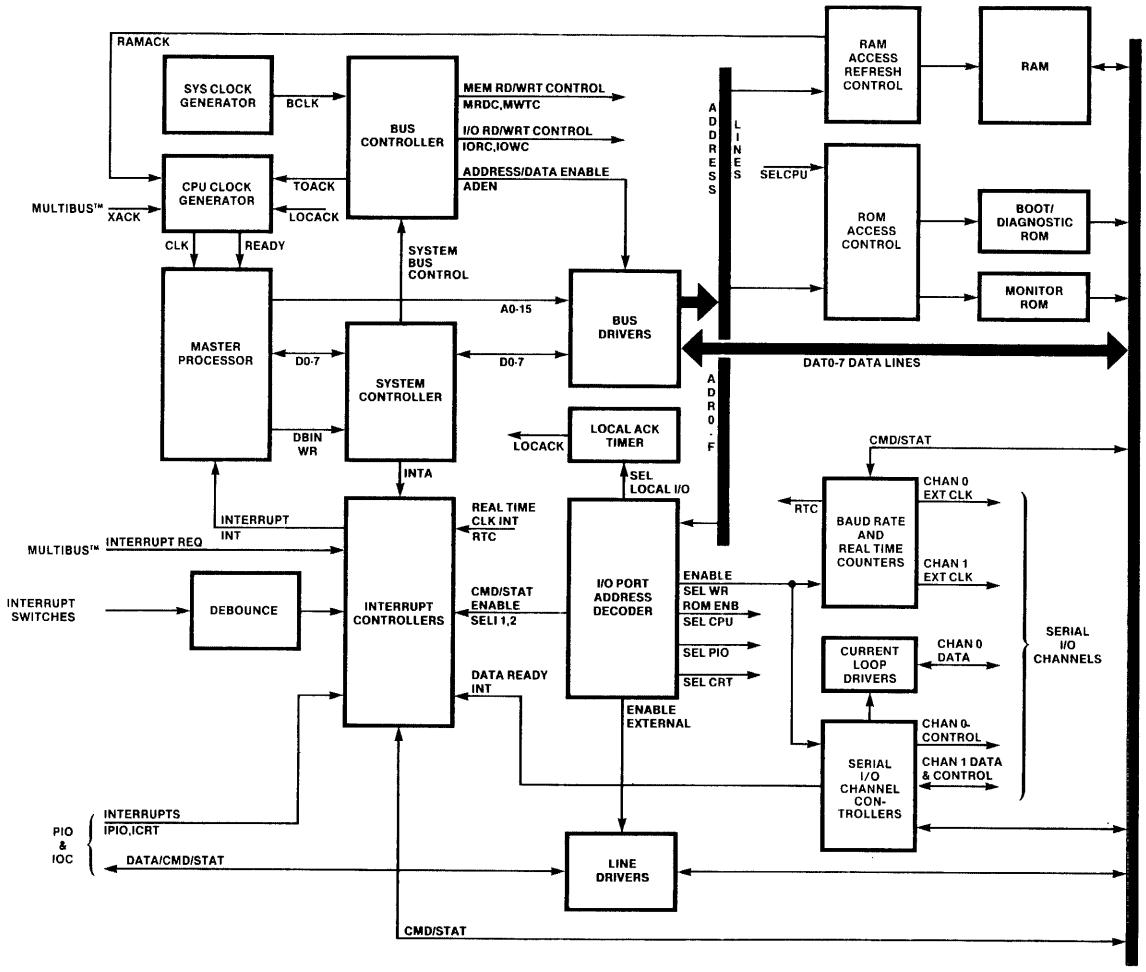
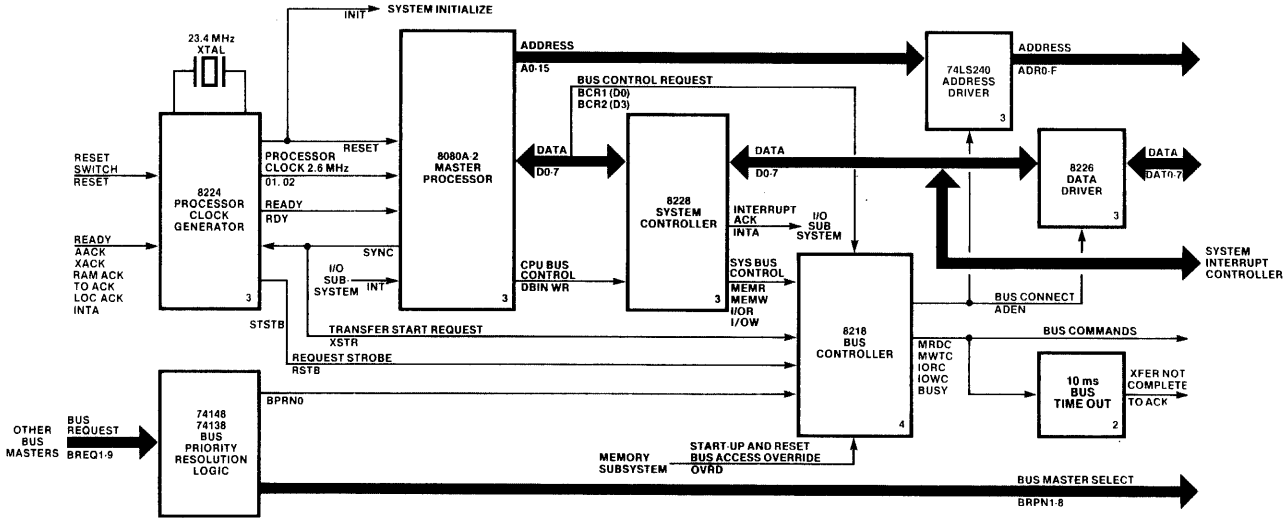


Figure 2-1. IPB Simplified Block Diagram 556-4



NOTE: The number in the lower right-hand corner of each block identifies the sheet number of the IPB schematic.

Figure 2-2. Master Processor Subsystem Block Diagram 556-5

2.1.2 SYSTEM CONTROLLER

The system controller also serves as the bi-directional bus driver for the master processor's data bus (D0-D7). Note that since the system interrupt controller is directly connected to the master processor's data bus, and not the system data bus (DAT0/-DAT7/), no other bus master can access the system interrupt controller. Also, during an I/O read of the system interrupt controller, the 8226 system data bus transceivers are forced into their output (write) mode to disconnect the system data bus from the master processor.

The 8228 system controller (sheet 3), at the beginning of each machine cycle, decodes the status output of the master processor into the individual control signals. The read/write control signals (MEMR/, MEMW/, I/OR/ and IOWR/) are used by the 8218 bus controller to control IPB-initiated internal and external communications. The interrupt acknowledge control signal (INTA/) is used by the 8259 system interrupt controller (to initiate an interrupt vector sequence) and by the clock generator (to enable the master processor's RDY input during interrupt acknowledge machine cycles).

2.1.3 BUS CONTROLLER

The 8218 bus controller (sheet 4) provides the bus acquisition, timing and control functions for IPB-initiated bus cycles.

2.1.3.1 BUS ACQUISITION. When the master processor requires use of the system bus (all 8080 machine cycles except Halt Acknowledge require bus access), the bus controller secures access to the bus and maintains bus access until another bus master requests the bus (the IPB is the lowest-priority bus master) or until the master processor instructs the bus controller to release the bus by executing a Halt instruction.

To initiate a bus acquisition cycle, the bus controller requires an active level on the RSTB/ (request strobe) input and an active level on either (or both) the BCR1 (bus control request 1) or BCR2/ (bus control request 2) input. Referring to sheets 3 and 4 of the schematic, the RSTB/ input is the STSTB/ (status strobe) output from the clock generator, and the BCR1 and BCR2/ inputs are the D0 and D3 outputs, respectively, from the master processor's data bus. Recalling the 8080's machine cycle timing, at the beginning of every cycle, STSTB/ goes active and a "status word" (defining the type of machine cycle to be executed) is output on the master processor's data bus. If another bus master is not requesting the bus, the BPRN/ (bus priority in) input from the bus priority logic on sheet 2 will be active. On the falling-edge of the second or third (machine cycle dependent) bus clock (BCLK/) cycle, if the bus is not in use (indicated by an inactive level on BUSY/) and if a bus request has not been received from another bus master, the bus controller activates BUSY/ and ADEN/ (address and data enable). BUSY/ is a bi-directional bus signal that, when active, indicates to all other bus masters that the bus is in use. ADEN/ is used locally by the IPB to enable the address latches and data transceivers on sheet 3. Note that once the IPB has acquired the bus, the bus controller maintains bus access (holds BUSY/ active) until either another bus master requests the bus or until a Halt instruction is executed. By maintaining bus access, the IPB avoids the bus acquisition overhead when no other bus master is requesting bus access. Table 2-1 defines the bus control request input levels for the various machine cycles, and figure 2-3 illustrates the bus acquisition timing.

Table 2-1. Bus Request Machine Cycles

8080 Machine Cycle	BCR1 (D0)	BCR2/ (D3)	*BCLK/Cycles to Access Bus
Instruction Fetch Memory Read or Write Stack Read or Write I/O Read or Write	0	0	2
Interrupt Acknowledge	1	0	3
Interrupt Acknowledge While Halted	1	1	3

* Assumes bus is available

If the bus is in use (if BUSY/ is low) or if a bus request is pending (if BPRN/ is high), the bus controller delays the generation of BUSY/ and ADEN/ until one BCLK/ cycle after both the bus becomes available and no requests are pending.

2.1.3.2 BUS RELEASE. Once the master processor has acquired the bus, it surrenders the bus only when another bus master requests the bus or when the master processor executes a Halt Acknowledge machine cycle. When another bus master requests the bus, the bus controller completes its current bus control cycle and releases the bus (inactivates BUSY/) on the next bus clock cycle. When a Halt Acknowledge machine cycle is executed, the bus controller releases the bus on the third bus clock cycle.

2.1.3.3 BUS OVERRIDE. The bus controller has an OVRD (override) input that, when active, allows the master processor to maintain bus access between bus cycles by preventing the recognition of a bus request from another bus master. The OVRD input is taken from bit 6 of the ROM enable latch and is set or reset through the IPB's control port. Referring to the schematic, OVRD is automatically activated when the system is initialized to prevent any other bus master from accessing the bus during the master processor's initialization/start-up procedure (following initialization, the Monitor program inactivates OVRD). During system operations, when the master processor must maintain bus access for a critical code section (e.g., when performing a test and set operation), the program would first activate OVRD through the control port.

2.1.3.4 BUS CONTROL CYCLE. The bus controller requires both the receipt of a transfer start request (XSTR) and the acquisition of the bus (BUSY/ and ADEN/ active) before it can begin a bus control cycle. The XSTR input to the bus controller is the SYNC output from the master processor. Since during any machine cycle, SYNC becomes active before STSTB/ goes active (the signal that starts the bus acquisition cycle), if the bus controller does not have access to the bus when a machine cycle begins, the bus control cycle is initiated when the bus is acquired (when BUSY/ and ADEN/ go active). Conversely, if the bus controller already has the bus, the control cycle is initiated by XSTR.

On receipt of the decoded command outputs from the 8228 system controller, the bus controller activates ANYR (any request). Additionally, if a read operation is indicated (IORR/ or MRDR/ active), the bus controller activates RDD. ANYR is externally connected to the bus controller's XCP/ (transfer complete) input, while RDD is used to control the direction of the system data bus transceivers. The bus controller uses an internal delay timer to establish the address set-up and data hold times for the bus cycle. When the bus control cycle is initiated, the delay timer is triggered. Following an approximate 100ns delay (determined by the RC time constant at the DLY ADJ input), the corresponding output command goes active. When the transfer operation is complete, the command output from the 8228 returns to an inactive level. At the bus controller, the loss of the command input causes ANYR and the corresponding command output to return to an inactive level. The negative transition of ANYR, at the XCP/ input, triggers the internal delay. Following the hold-time delay (approximately 100ns), the XCY/ (transfer cycle) output returns to an inactive level (indicating that the bus cycle is complete). Figure 2-4 illustrates the bus control cycle timing.

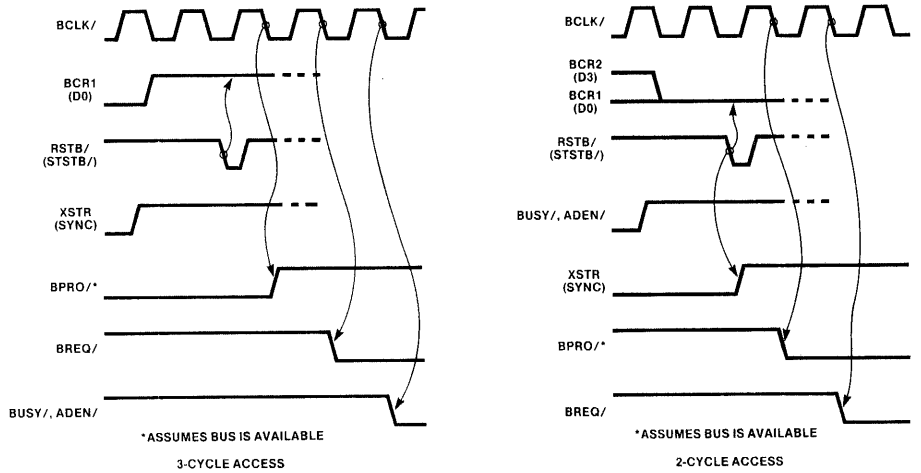


Figure 2-3. Bus Acquisition Timing 556-6

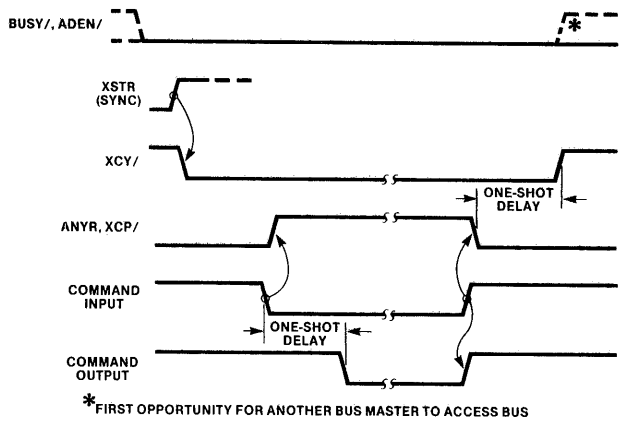


Figure 2-4. Bus Control Cycle Timing 556-7

2.1.4 BUS PRIORITY RESOLUTION LOGIC

Bus priority resolution logic (sheet 2) includes a 74148 eight-to-three priority encoder and a 74138 three-to-eight decoder. The primary inputs to the encoder are bus requests BREQ1/ through BREQ8/. If there is more than one active bus request, the encoder only generates the binary code for the highest priority (highest numbered) request. The decoder uses the binary code from the encoder to activate the appropriate bus priority in (BRPNx/) signal to the requesting bus master.

The highest priority bus master and the lowest priority bus master (the IPB) are excluded from the priority resolution logic. The highest priority bus request (BREQ9/) is used to block all other bus requests and does not require the BRPN/ signal because it assumes control of the bus as soon as any current bus transfer has been completed. The IPB continually requests use of the bus, but is inhibited by BPRN0/ if any other bus master requests the bus.

Routing of the BREQ/ and BPRN/ lines is arranged so that the priority of boards inserted in the Intellec Series II development system is increased toward the bottom of the chassis (including the expansion chassis). The IPB has the lowest priority since it can only be installed in the upper-most slot.

2.2 IPB MEMORY SUBSYSTEM

The memory subsystem of the IPB (figure 2-5) controls up to 64k of random access memory (32k on the IPB and 32k on an optional RAM board) and 4k of read-only memory (ROM). Because the 16-bit addresses output by the master processor can specify a maximum of 64k locations, the two types of memory (ROM and RAM) share the same address space. The technique that permits address space sharing is known as shadowing ROM wherein ROM is preferentially selected unless disabled. Operationally, the use of shadow ROM is unnoticeable because the ROM programs are either called by name or are accessed through use of operator procedures that do not require knowledge of the program locations. From a hardware viewpoint, RAM and ROM are simply two distinct slaves to the master processor.

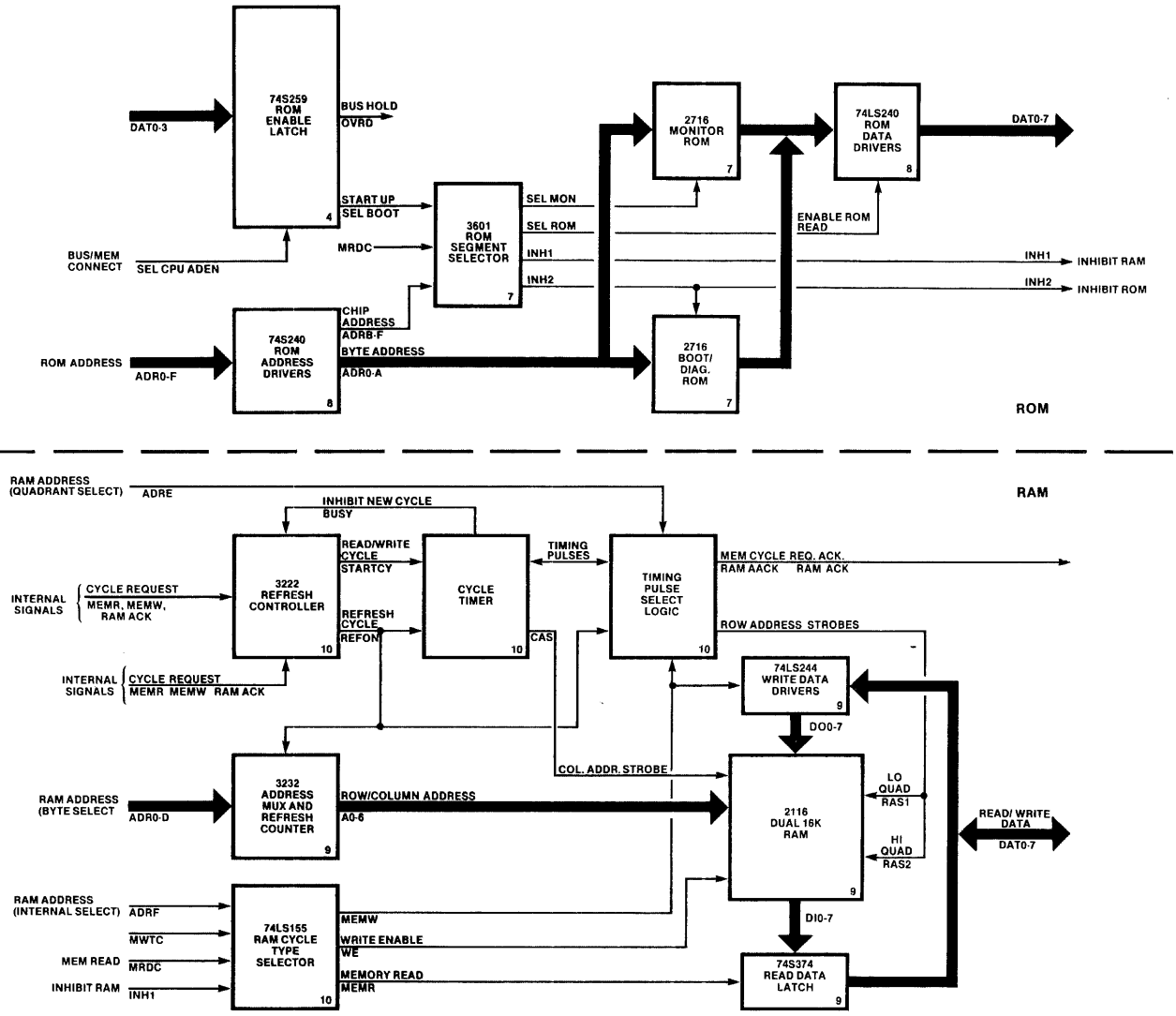
Major circuit elements of the memory subsystem include sixteen 16k bit dynamic RAM chips, two 2k byte ROM chips, a 3222 RAM refresh controller, a 3242 RAM counter, a 3601 high-speed ROM (for memory address decoding), and a variety of TTL gates, flip-flops, drivers, and multiplexers/demultiplexers. All circuit elements are off-the-shelf items, and all but the TTL elements are Intel LSI products.

2.2.1 IPB ROM

Read only memory (ROM) available on the IPB is divided into two segments. One segment is a single 2k ROM chip that contains the Monitor program. The second segment is also a 2k ROM chip and is used to store the bootstrap-loader/diagnostic program.

The Monitor program is accessible to all bus masters. The 2k Monitor ROM segment begins at address F800H. This segment cannot be shadowed, and the last 2k (locations F800H through FFFFH) of the optional 32k RAM board cannot be used.

The boot/diagnostic ROM segment is only accessible to the master processor. The master processor's control port (port FFH) is used to set or reset individual bits within the ROM enable latch in order to move, enable or inhibit the boot/diagnostic ROM segment. Referring to sheet 4, when the master processor is initialized (power on or reset), the ROM enable latch (A73) is cleared (all outputs active). The combination of START UP/ and SEL BOOT/, at the ROM segment selector (A6) on sheet 7, establishes the starting location of the ROM segment at location 0H (the location from which the master processor fetches its first instruction following reset) as well as at location E800H. The boot-strap program immediately jumps into the ROM segment beginning at location E800H and inactivates the START UP/ output from the ROM enable latch (by writing 02H to the control port). At the ROM segment selector on sheet 7, the inactive level of START UP/ disables the ROM segment at location 0H, and the boot ROM's chip select input is active only for addresses ranging from E800H to EFFFH. Disabling the ROM segment at 0H redefines the locations below E800H as RAM and makes this area of memory available for diagnostics.



NOTE: The number in the lower right-hand corner of each block identifies the sheet number of the IPB schematic.

Figure 2-5. IPB Memory Subsystem Block Diagram 556-8

Following execution of the boot/diagnostic program, control is transferred to the Monitor program at F800H. The Monitor program, when it begins execution, inactivates the SEL BOOT/ output from the ROM enable latch (by writing 04H to the control port). At the ROM segment selector, the inactive level of SEL BOOT/ inactivates the boot ROM's chip select input (the boot/diagnostic ROM "disappears" from the address space), and all addresses below F800H are redefined as RAM. Note that provision is included in the Monitor program to reenable the diagnostic portion of the boot/diagnostic ROM segment. Entering the Monitor's Z$ command transfers control to the diagnostic (the Z command subroutine reactivates the SEL BOOT/ output by writing 0CH to the control port).

The ROM enable latch also includes an interrupt enable bit that is set or reset through the master processor's control port. This bit is cleared (interrupts disabled) when the IPB is initialized and is subsequently set (interrupts enabled) when the Monitor is entered. The Monitor's IOC and PIO interface driver routines automatically clear the interrupt enable bit when called (to prevent an interrupt from being serviced during a sequence-dependent section of code) and set the bit on return.

The interrupt enable bit is used to enable or deny the interrupt (INTR) output from the system controller to the master processor. By controlling interrupts in this manner, the system interrupt controller and the master processor's internal interrupt flip-flop are not disturbed when using the Monitor's I/O routines to access system resources. The user program can use the master processor's DI (disable interrupt) and EI (enable interrupt) instructions without regard for Monitor operations.

The input to the 74LS259 ROM enable latch (sheet 4) is a hexadecimal code appearing on data lines DAT0/-DAT3/. The three LSBs of this code select one of four outputs of the ROM enable latch. The MSB of the code either sets or resets the selected output. The four outputs, the function of each output, and the related hexadecimal codes are outlined in table 2-2.

Accessing of enabled ROM segments is accomplished solely by the occurrence of the memory read signal (MRDC). The memory addresses applied through the ROM address drivers are accepted by the ROM segment selector (sheet 7). The ROM segment selector, itself is a ROM that is programmed to decode the five MSBs of the address to select the appropriate 2k ROM chip. The eleven LSBs of the address are applied to the ROM chip to select a specific byte for output to the ROM data drivers (sheet 8).

Outputs of the ROM segment selector include SEL ROM/ to enable the ROM data drivers, Select Monitor to select the Monitor ROM, Select Boot to select the boot/diagnostic ROM, and INH1/ to inhibit RAM operations. When the boot/diagnostic ROM is selected, INH2/ is also generated to allow the boot/diagnostic routine to shadow other possible ROM in the same address space. Both INH1/ and INH2/ are applied to the Multibus.

NOTE

Reassignment of address space for the boot/diagnostic, as discussed in paragraph 2.2.1, is strictly a function of the ROM segment selector that decodes the START UP/ input from the ROM enable latch.

2.2.2 IPB RAM

With normal configurations of Intellec Series II development systems, the IPB accesses either 32k or 64k of RAM with the first 32k located on the IPB board and the second (high address) 32k located on a separate 32k RAM board. Note that the last 2k bytes (addresses F800 through FFFF) of the upper 32k segment cannot be used as they correspond to the Monitor ROM's address space. The internal and external 32k segments of RAM are both Multibus slaves and therefore employ an identical set of data, address and control signals. In fact, the only signal that is unique between the two 32k segments is the most significant bit (ADRF). This one bit determines if the byte to be accessed is from the internal or external segment.

The internal 32k RAM (figure 2-5) is implemented as a dual 16k memory to allow use of standard Intel chips such as the 2116 or 2117 16k-by-1 bit dynamic RAM. Eight of these chips are connected in parallel to establish a basic 16k byte bank of RAM, and two such banks comprise the internal 32k RAM. Selection of a 16k bank is accomplished by the second MSB of the address (ADRE). The remaining 14 LSBs of the address are multiplexed through an Intel 3242 address multiplexer and refresh counter in the form of a 7-bit row address and a 7-bit column address. In other words, each of the RAM chips is treated as a 128-by-128 matrix that provides storage for a specific bit position of 16k distinct bytes.

A given byte is accessed upon application of RAS (row address strobe) and CAS (column address strobe) to the eight RAM chips comprising a bank. In compliance with 2116/2117 requirements, RAS is applied before CAS for normal read/write operations, and the specific operation (read or write) is determined by WE (write enable). Refresh is accomplished by application of RAS without CAS. Because RAS affects 128 bits within each RAM chip, a total of 128 bytes are refreshed within each bank during each refresh cycle (one row of each RAM bank). A refresh cycle (figure 2-6) occurs once every 15 microseconds and the entire 32k memory is refreshed within two milliseconds.

Table 2-2. ROM Enable Latch Decoding

Name	Function	Set Code	Reset Code
SEL BOOT/	Boot/diagnostic ROM selection	0CH	04H
START UP/	Boot start address= 0000 (not E800)	Illegal	02H
Override	Inhibit other Multibus users	09H	01H
Interrupt Enable	Disable interrupts	0DH	05H

* Code input levels are active low (low = 1)

Read/write operations are initiated by MRDC/ (memory read command) or MWTC/ (memory write command) if the RAM is enabled (INH1/=1) and if the address specifies internal RAM (ADRF=0). If a read operation is specified, MEMR/ and DATA STROBE enable the output of the read data latch (sheet 9) onto the bus. For write operations, MEMW/ (memory write) and W/ (write enable) are output to the write data drivers and the RAM chips, respectively.

The read/write commands initiate RAM access, but a RAM cycle does not occur until the generation of RAS/ and CAS/. These strobes are generated by actions of the refresh controller, the cycle timer, and the timing pulse select logic (sheet 10). A read or write cycle is initiated by the refresh controller on the occurrence of MEMR/ or MEMW/ if the RAM is not being refreshed (BUSY/=1). When a cycle is initiated, STARTCY/ is output to the cycle timer to initiate the shifting of a "1" state through a series of flip-flops. The specific timing employed for a read or write cycle is determined by the timing pulse select logic. The timing of read and write cycles is shown in figures 2-7 and 2-8.

Refresh cycles are initiated once every 15 microseconds by an R/C time constant of the refresh controller unless a read/write cycle is in progress (BUSY/=0). The refresh controller's REFON (refresh on) output is sent to the timing pulse select logic to alter the RAM timing cycle. REFON is also sent to the address multiplexer and refresh counter to cause this chip to select an internally generated row address instead of the externally supplied RAM address. The internal row address is generated by a 7-bit, modulo-128 binary counter.

Upon receipt of REFON, the timing pulse select logic modifies the timing cycle such that CAS/ is inhibited. In this case, RAS1/ and RAS2/ are both generated to cause refresh of a specified row in both banks of RAM. RAM AACK, RAM ACK and DATA STROBE are not output during refresh cycles. Any request for a read/write cycle during a refresh cycle is not aborted, but is merely delayed for up to 550 nanoseconds. The cumulative delay of user programs due to this method of asynchronous RAM refresh is less than four percent.

2.2.3 EXTERNAL ROM AND RAM

ROM and RAM existing on optional or user-designed circuit boards, and tied directly to the Multibus, may be accessed by the IPB provided that the memory is within the IPB's address range (0 to 64k). All such memory extensions must have mutual compatibility with Multibus timing as implemented within Intellec Series II development systems.

2-3. IPB INPUT-OUTPUT SUBSYSTEM

The I/O subsystem of the IPB (figure 2-9) provides overall management of all Intellec Series II development system communications with the master processor. Such communications include "system" I/O transfers to or from other optional Multibus boards and "local" I/O transfers. The system I/O transfers are accomplished via the Multibus interface.

Local I/O transfers use the Multibus, but are accomplished via I/O ports of the master processor. The local I/O transfers are subdivided into external (PIO and IOC) transfers and internal I/O tranfers. All internal I/O transfers are between the master processor and the on-board programmable resources of the IPB.

The I/O subsystem controls the routing of external I/O transfers to and from the PIO and IOC. In the case of external transfers, the I/O subsystem decodes the I/O port addresses and enables the common data lines of the PIO and IOC. The port addresses also

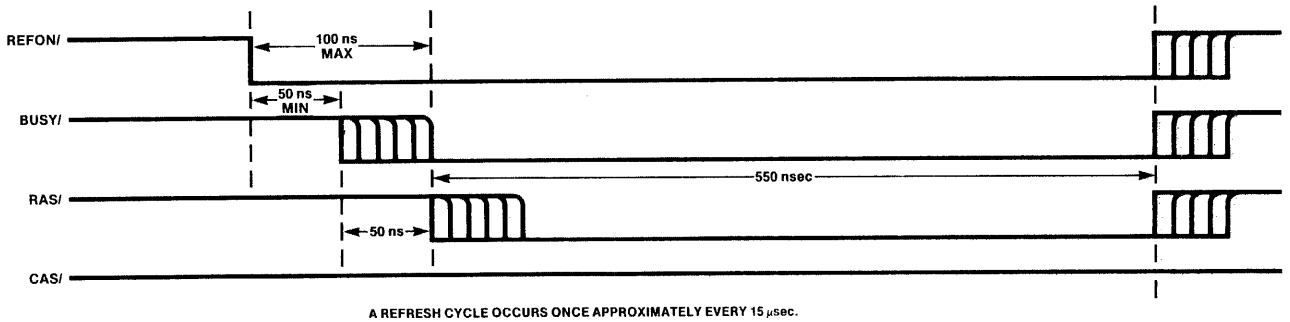
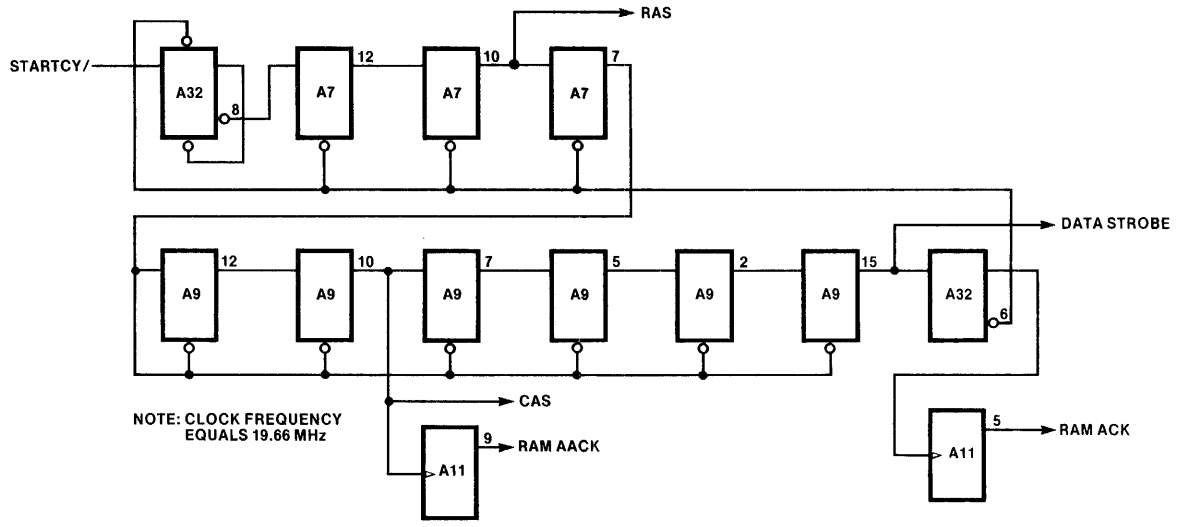


Figure 2-6. RAM Refresh Cycle Timing 556-9

READ SELECTS A INPUTS OF 74S257 TO ESTABLISH A CIRCUIT EQUIVALENT TO THE FOLLOWING:



556-10

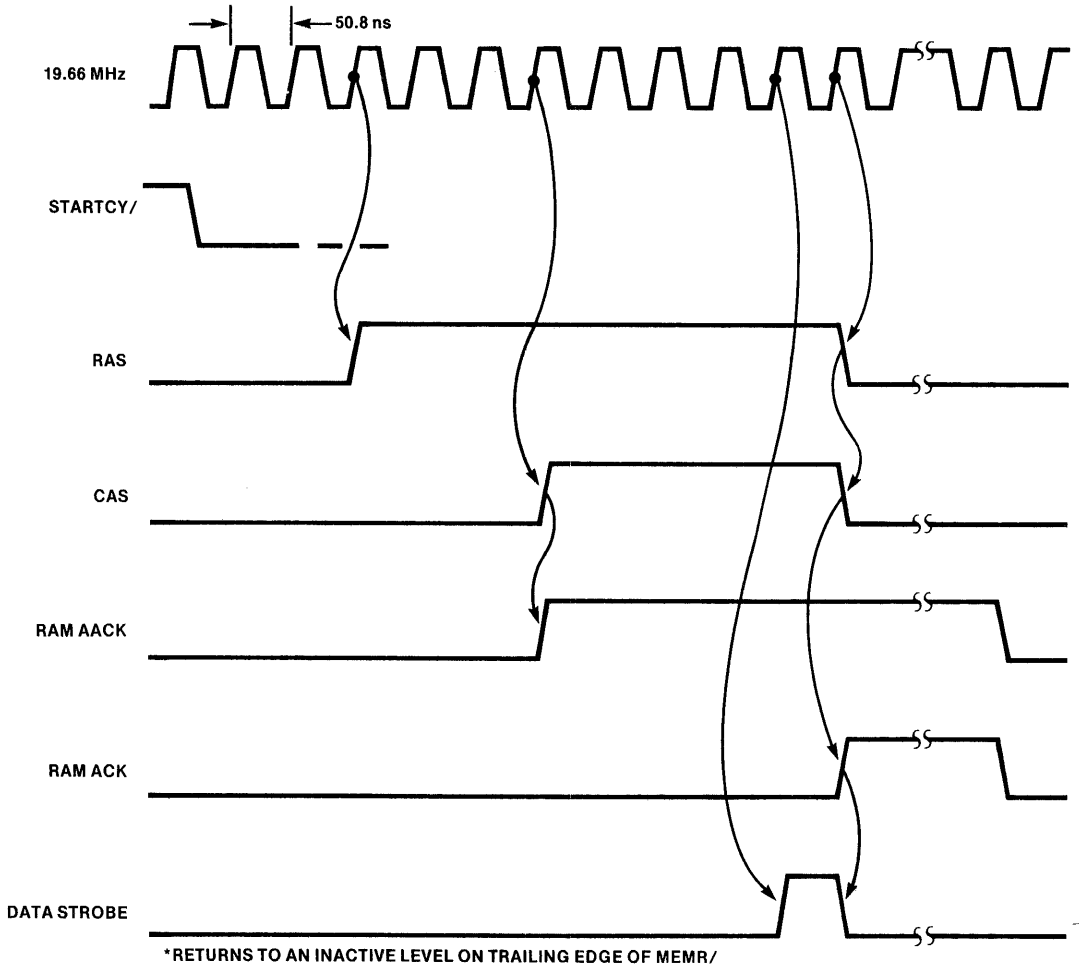


Figure 2-7. RAM Read Cycle Timing

556-11

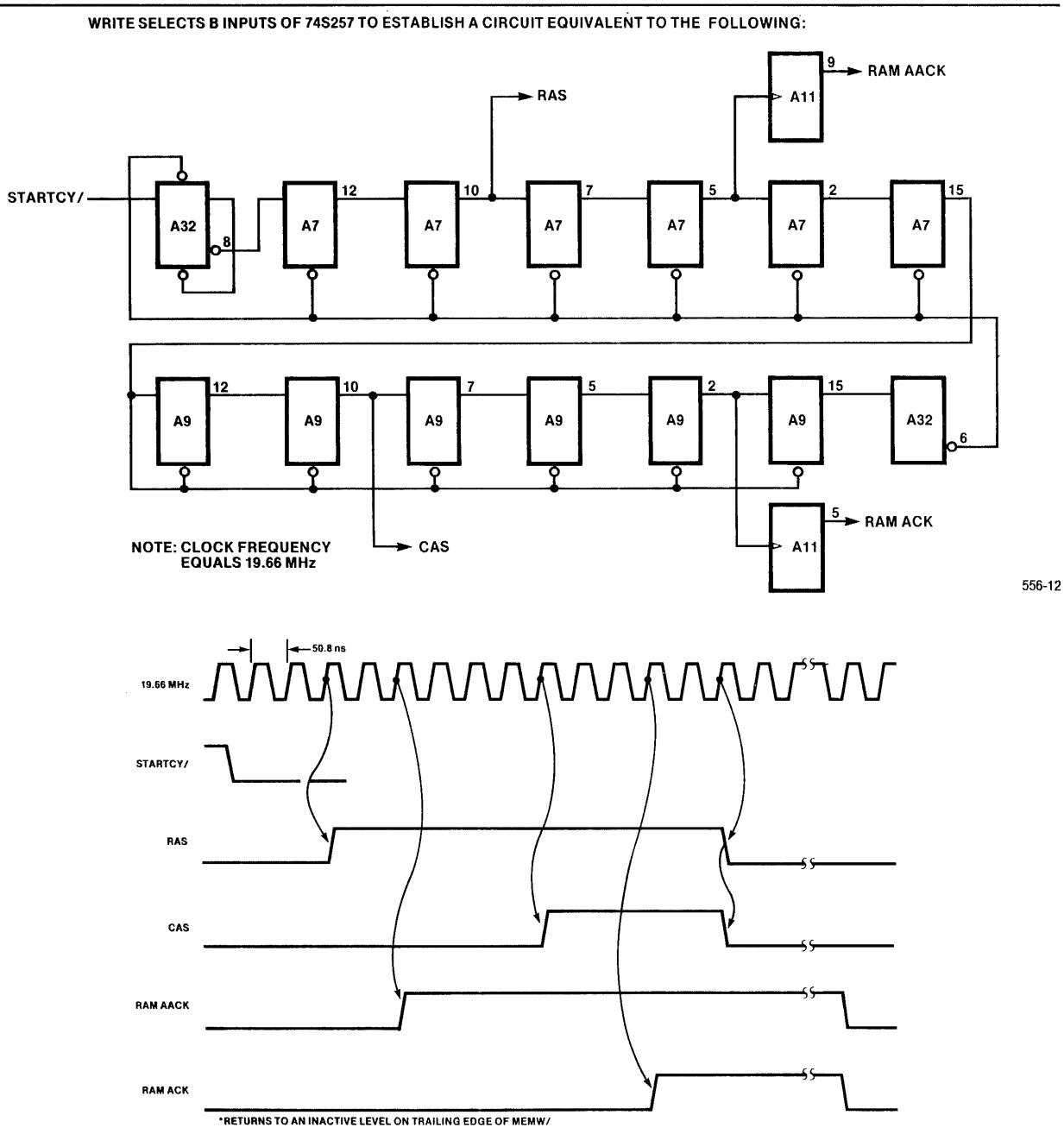
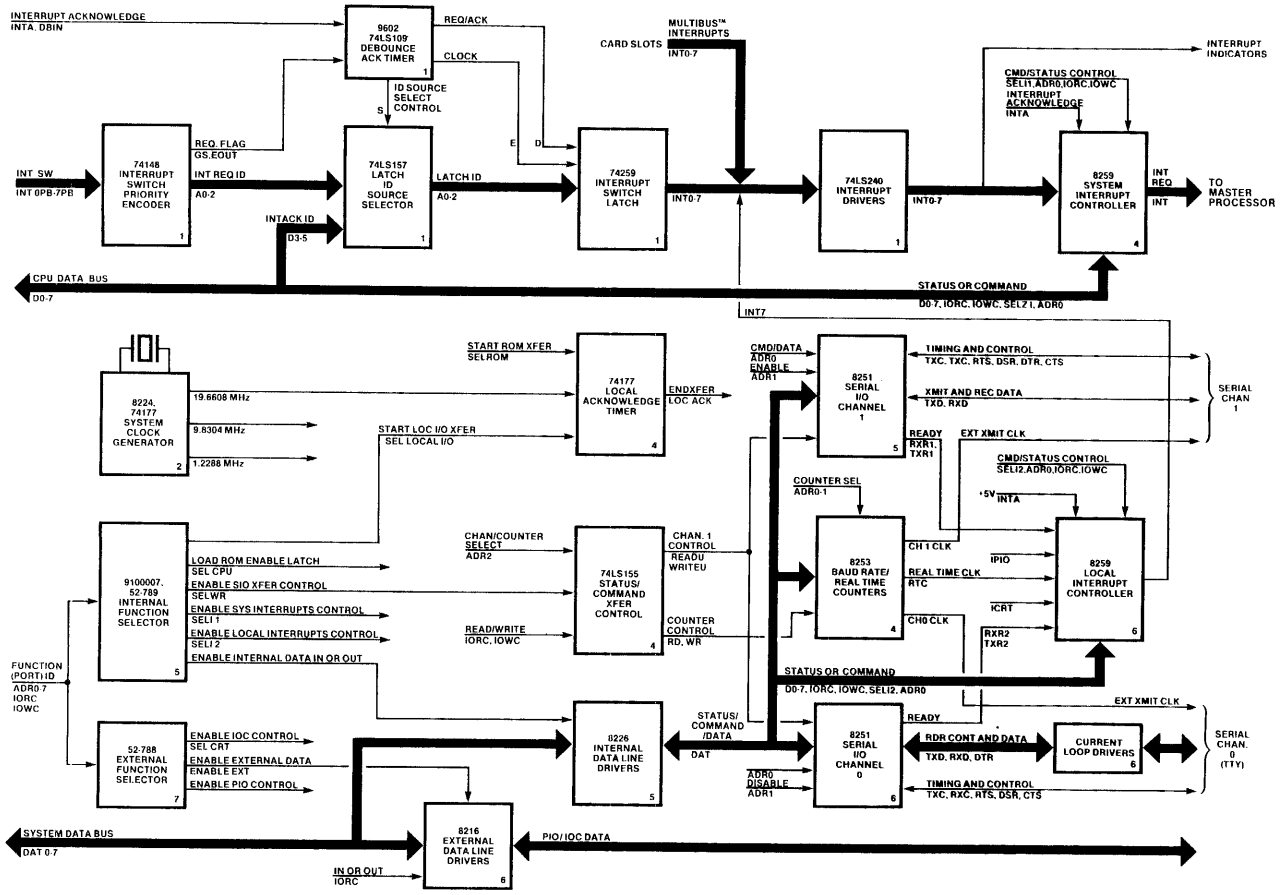


Figure 2-8. RAM Write Cycle Timing

cause the I/O subsystem to generate a select signal for either the PIO or the IOC. The select signals initiate communications between the master processor and the 8041 microprocessor of the PIO or the 8080A-2 microprocessor of the IOC. These external microprocessors are, in turn, responsible for communications with I/O devices (peripherals, integral disk, and CRT/keyboard). The IOC microprocessor is also directly responsible for issuing commands to, and receiving status from, the programmable chips of the IOC.

The internal I/O transfer circuits decode I/O port addresses from the master processor. Each decoded port address causes the generation of a select signal that is used to enable a data transfer between the master processor and the programmable chip. Three types of data transfers occur:

- Output of a command (i.e., operating parameter) to the selected chip.
- Input of a status byte from the selected chip.
- Input or output of a data byte to or from the selected chip.



NOTE: The number in the lower right-hand corner of each block identifies the sheet number of the IPB schematic.

Figure 2-9. IPB I/O Subsystem Block Diagram 556-14

A total of five programmable chips are employed within the I/O subsystem. The functions of each chip are:

- Serial I/O Channel 0 (8251)—A universal synchronous/asynchronous receiver transmitter (USART) that establishes either a 20mA current loop or RS-232 interface between an external serial device and the IPB master processor. Commands permit full duplex asynchronous communications from 110 to 1200 baud for the current loop interface using seven bits plus parity for teletypewriter compatibility. The commands can also establish full duplex synchronous communications from 150 to 56k baud for compatibility with RS-232 or Bi-sync terminals or modems. The serial channel may use internal or external baud rate clocks as established by jumpers. Status bytes returned on command indicate the state of the current transfer and the occurrence of any transmission error. When the system is initialized, the boot program sets the channel for asynchronous operation with two stop bits, an 8-bit character length and a baud rate factor of 16X (the channel 0 external clock signal from the baud rate counter is 1.76 kHz which, divided by 16, provides a 110 baud rate).

- Serial I/O Channel 1 (8251)—Identical to serial I/O channel 0 with the exception that associated circuits are not available to establish the current loops required by teletypewriters. Serial channel 1 is also programmed for asynchronous operation with two stop bits, an 8-bit character length and a baud rate factor of 16X. The channel 1 external clock signal is 38.4 kHz and provides a 2400 baud rate.

- Baud Rate/Real Time Counters (8253)—A programmed timer consisting of three independent counters. Two of these counters establish baud rates for the serial I/O channels. The third counter is used to generate a 1 ms real-time clock signal. Command inputs establish both the

frequency and the mode of operation for the timer/counter. The content of any counter is returned to the master processor on command. The real time clock is not used by basic operations of Intellec Series II development systems; it is reserved for user applications. When the system is initialized, all three counters are programmed for Mode 3 (square wave generator) operation and use a two-byte count register. The count register values and counter functions are as follows:

Counter	Function	Counter Value
0	Channel 0 Clock	698
1	Channel 1 Clock	32
2	1 ms Real Time Clock	1229

- System Interrupt Controller (8259)—A programmable interrupt controller that establishes priorities and masking for eight interrupt request lines (INT0-7). Highest priority is assigned to interrupt INT0. Each line may carry an interrupt request from either a front panel interrupt switch or from a Multibus user. Some Multibus devices such as ICE modules may employ one or more reserved interrupt requests to enable identification by software, but there is no distinction between a given Multibus interrupt request and the equivalent front panel switch. The lowest priority interrupt (INT7) is used by the IPB's local interrupt controller. The system interrupt controller is operated in a fully nested mode such that an interrupt request being serviced may be interrupted by a higher priority interrupt. Status returned on command includes the identity of the interrupt request being serviced, the identities of all interrupts currently nested, and the content of the internal mask register. All interrupts may be selectively enabled or disabled by the master processor program of the eight interrupts, INT0 and INT1 are reserved by the Monitor and ISIS, and INT7 is used by the local interrupt controller.
- Local Interrupt Controller (8259)—The local interrupt controller processes seven interrupt requests: a receive data and a transmit data interrupt request for each serial I/O channel, a PIO interrupt request, an IOC interrupt request, and a real time clock interrupt request. These interrupt requests are fully nested and an INT7 interrupt to the system controller is generated on any local interrupt. Upon the occurrence of INT7, the master processor program causes polling of the local interrupt controller to determine the origin of the interrupt request. A code identifying the highest priority interrupt request being serviced is returned to the master processor via the data lines. In all other respects, the local interrupt controller is functionally identical to the system interrupt controller. Interrupt assignment for the local interrupt controller is as follows:

INT0 Serial I/O channel 0 input data ready
INT1 Serial I/O channel 0 output data ready
INT2 Serial I/O channel 1 input data ready
INT3 Serial I/O channel 1 output data ready
INT4 Real Time Clock
INT5 PIO operation complete
INT6 IOC operation complete
INT7 Not Used

NOTE

The preceding Intel programmable chips and all of the other chips of the I/O subsystem are standard off-the-shelf products that are individually supported by documentation adequate for maintenance purposes. The following text assumes familiarity with the documentation and explains only the usage or interactions between these chips.

2.3.1 INTERRUPT REQUEST CIRCUITS

The interrupt request circuits consist of the previously discussed system and local interrupt controllers plus a number of TTL chips. Most of the TTL chips are used to provide debouncing and latching for the front panel interrupt switches. Details of these circuits are shown on sheet 1. Inputs from the front panel interrupt switches are applied to the 74148 interrupt switch priority encoder. The encoder generates a three-bit binary code for the switch being pressed. If more than one switch is pressed, the binary code generated is that of the highest priority (lower numbered) switch. Any active switch input causes the GS output to be low and the EOUT output to be high. The GS output triggers the first (10 millisecond) pushbutton debounce timer, and the EOUT output enables the second (500 nanosecond) timer to be triggered by the first timer.

The binary output of the encoder is applied through the 74LS157 ID source selector multiplexer to the 74259 interrupt switch latch. The binary code (latch ID) is accepted by the interrupt switch latch on the first $\Phi 1'$ clock following the triggering of the second timer (the second timer sets the debounce flip-flop to enable the $\Phi 1'$ clock at NAND gate A16). When the interrupt latch is enabled, the level present at the latch's D input is stored into the addressed latch. Referring to the schematic, the D input originates from the auto acknowledge timer and, since the timer

is idle, will be at a logic high level. The interrupt switch latch thus stores a single bit that corresponds to the activated interrupt switch and, in effect, reconverts the binary code into a discrete interrupt signal. The interrupt latch outputs are wire OR'ed with the corresponding bus interrupts (INT0/-INT7/) at the inputs to the 74S240 interrupt drivers.

During the second cycle of an interrupt acknowledge sequence (second INTA/), the D3, D4 and D5 data bits of the low-order call address byte from the 8259 system interrupt controller, since the 8259 was initialized with a call address interval of 8, reflect the binary code of the interrupt switch currently being serviced. At this point in time, the auto acknowledge timer will be logically arranged to:

1. Multiplex the D3 through D5 data bits at the B inputs to the ID source multiplexer to the interrupt switch latch (Q output at A17 pin 10 logically high).
2. Set the D input to the interrupt switch latch to a logic low level in order to clear the addressed interrupt switch bit (Q output at A17 pin 9 logically low).
3. Enable the interrupt switch latch (Q output at A17 pin 6 logically high).

The addressed interrupt (interrupt currently being serviced) is cleared from the interrupt latch during the second INTA/ cycle when Φ1′ and DBIN are active.

CAUTION

Attempting to reprogram the 8259 interrupt controller with a call address interval of 4 will cause the interrupt acknowledge logic to address the wrong bit within the interrupt switch latch.

2.3.2 EXTERNAL (PIO/IOC) I/O CONTROL CIRCUITS

The master processor controls the PIO and the IOC in the same way as it would control any peripheral device. Commands are issued by the master processor to accomplish I/O data transfers, to issue new operating parameters, and to cause the return of status bytes. When the task specified by a command is completed, a service request bit is set within a status byte. When accessed, the status byte informs the master processor of compliance with the command. The PIO or IOC then waits until a new command is output by the master processor. Although the PIO and IOC respond in a similar manner, these two functions are completely independent of each other.

The external function selector ROM (A71 on sheet 7) decodes the following I/O port addresses.

F8,F9 PIO
C0,C1 IOC
C2,C3 Reserved for future use by IOC
C4,C5 Reserved for future I/O controllers

Programs that direct PIO or IOC activities use a specific port for a given purpose. For example, port C1 is used for commands and DBB status, while all IOC data is transferred through port C0. (All status bytes other than the DBB status byte and all parameters are considered data.) The type of transfer to occur is indicated by address bit A0 (the command/data signal) and IORC/ (the I/O read command) or IOWC/ (the I/O write command).

Any one of the port addresses listed above causes generation of the ENABLE EXT/ signal which enables the 8216 external data line drivers (sheet 6). The direction of data flow within the drivers is independently determined by the state of I/O read command signal (IORC/).

If the port address is F8 or F9, SEL PIO/ is generated. This signal is applied directly to the chip select (CS) pin of the PIO's 8041 microprocessor. When SEL PIO/ is low, the data bus buffer of the PIO processor is enabled.

If the port address decoded by the external function selector is C0 through C3, SEL CRT/ is generated to enable the IOC's data bus buffer.

NOTE

While the mnemonic SEL CRT/ would seem to imply association with the CRT only, this signal is also active for diskette and keyboard transfers controlled by the IOC.

2.3.3 INTERNAL I/O CONTROL CIRCUITS

The internal I/O control circuits establish communications paths between the master processor and the programmable chips of the IPB I/O subsystem. The process involved consists of the following sequence:

- Select chip by means of an I/O port address.
- Select command-out, status-in, data-out, or data-in transfer mode.
- Enable data lines between the chip and the master processor.
- Start 1.6 microsecond local acknowledge timer to enable the master processor to complete the transfer cycle (ready input active).
- Transfer the command, status, or data byte via the data bus.

The I/O port address is used with either the I/O read or I/O write command to control the transfer of a command, status, or data byte between the programmable chips and a master processor. The IORC/ and IOWC/ signals only establish the direction of data flow; the I/O port addresses are responsible for chip selection and for varying degrees of chip control. In the case of the 8251 serial I/O chips, a separate port is required for the serial data and for the status and commands that monitor and control the data. Two port addresses are, therefore, employed. With 8259 interrupt controller chips, commands and status are transferred over two adjacent ports. In effect, the LSB of the port address (ADR0) augments the command concurrently appearing on the data lines. Augmenting of commands is also done for the 8253 baud rate and real time counters chip in which case the two LSBs of the port address (ADR0 and ADR1) are used to select one of the three counters or a control register within the 8253 chip. The data lines are then free to read or write counter values or to write to the control word registers that establish counter modes.

Decoding of the internal port addresses is performed by the two 3601 ROMs (A69 and A70) on sheet 5. These two chips decode 13 distinct port addresses to produce seven control signals. The port addresses and associated chip select signals for both internal and external I/O control are shown in table 2-3. The following text discusses each of the internal data control chip select signals and defines how they relate to I/O subsystem hardware.

One significant factor illustrated in table 2-3 is that the SEL LOCAL I/O/ signal is generated for each of the port addresses used by both the internal and external I/O control circuits. This signal is applied to the 74177 local acknowledge timer (sheet 4). The timer enables master processor program execution 1.6 microseconds after the start of any local I/O transfer.

The enabling of the various ROM segments is provided by the ROM enable latch (section 2.2.1). The ROM enable latch is an output-only port that is controlled by the SEL CPU/ ouput from the internal function selector on sheet 5.

The SELI1/ and SELI2/ outputs of the internal function selector are used to select the 8259 local interrupt controller (sheet 4) and the 8259 system interrupt controller (sheet 6), respectively. Both controllers use IORC/ to enable reading of status, IOWC/ to enable writing of commands, and ADR0 to augment commands appearing on the data lines. One difference between implementation of the two controllers is that the system interrupt controller is connected directly to the CPU's data bus (D0-D7), whereas the local interrupt controller is connected to the system data bus (DAT0-DAT7) via the local data bus transceivers (sheet 5).

The 8226 local data bus transceivers are controlled by two signals from the internal function selector. One of these two unnamed signals selects the transceiver while the other selects the direction (input or output). All programmable chips of the I/O subsystem except the system interrupt controller communicate with the master processor via the local data bus transceivers.

The SELWR/ output of the internal function selector is used with ADR2/, IORC/, and IOWC/ as an input to the 74LS155 status/command transfer control circuit (sheet 4). Outputs from this chip select either the 8253 baud rate and real time counters or the pair of 8251 serial I/O channels. The RD and WR signals select the 8253 and establish the transfer of a command (WR) or a status byte (RD). The type of byte transferred is determined by ADR0/ and ADR1/. The function of the READU/ and WRITEU/ signals is similar to RD and WR. READU/ and WRITEU/ are used by both serial I/O channels.

The serial I/O channel chips (sheets 5 and 6) use ADR1/ to select one or the other of these chips (when ADR1/ is low, channel 0 is selected). ADR0/ is applied to the command/data input of both chips to indicate whether the transfer is serial data to or from the external device or command/status information.

Table 2-3. Port Address/Control Signal List

Port Address	Device Selected	Port Function	Output Signal Names
F0	8253 Timer	Channel 0 Baud Rate	SELWR/ *ENABLE INTERNAL SEL LOCAL IO/
F1		Channel 1 Baud Rate	
F2		Real Time Clock	
F3		Counter Mode Select	
F4	8251 Serial I/O Channel 0	Data	
F5		Command/Status	
F6	8251 Serial I/O Channel 1	Data	
F7		Command/Status	
F8	PIO Subsystem	Data	SEL PIO/ ENABLE EXTERNAL SEL LOCAL IO/
F9		Command/Status	
FA	8259 Local Interrupt Controller	Command/Status	SELI2/ *ENABLE INTERNAL SEL LOCAL IO/
FB		Command/Status	
FC	8259 System Interrupt Controller	Command/Status	SELI1/ SEL LOCAL IO/
FD		Command/Status	
FE	Reserved		
FF	ROM Enable Latch	Access Control	SEL CPU/ SEL LOCAL IO/
C0	Input/Output Controller	Data	SEL CRT/ ENABLE EXTERNAL
C1		Command/Status	
C2		Reserved	
C3		Reserved	
C4	Reserved	N/A	SELDPP ENABLE EXTERNAL SEL LOCAL IO/
C5			

* ENABLE INTERNAL is used to identify two unnamed signals that control the local data bus transceivers.

CHAPTER 3
INTEGRATED PROCESSOR CARD

The Integrated Processor Card (IPC) is an 8085A-based single board computer for the Intellec Series II Microcomputer Development Systems. The IPC itself includes 64k bytes of random access memory, 4k bytes of read only memory that contain the bootstrap, diagnostic and monitor programs, and two serial input/output channels as well as the interfaces to the Multibus, the Input/Output Controller (IOC) and Parallel Input/Output (PIO) subsystem. Figure 3-1 is a simplified block diagram of the IPC. The dashed lines divide the IPC into its three major subsystems.

As a single board computer, the IPC controls all communications with the peripheral devices associated with the IOC and the PIO subsystem and, as a Multibus master, controls all communications with bus slaves interfaced to the Multibus. (Other bus masters use the IPC's bus priority logic to acquire control of the Multibus and to access the IPC's resources.) Data transfers between the IPC and peripherals associated with the IOC/PIO interface, the serial I/O channels or the Multibus are initiated by hardware interrupts or as a result of the IPC's polling of device status through dedicated I/O ports. As new devices are interfaced, additional interrupts and I/O ports are assigned, but the operation of the IPC hardware remains unchanged; the software is expanded to include the new device. Since it is only necessary to expand the software, the IPC represents a true general-purpose computing system; most other bus masters are dedicated to special purpose tasks and rely on the IPC for such functions as bus priority resolution and RAM access and refresh.

Operation of the Multibus is arranged so that the IPC has the lowest bus priority of the ten potential bus masters that can be interfaced to the Multibus. Use of the IPC's resources (i.e., serial I/O channels, CRT, PROM programmer, etc.) by another bus master is restricted only in that the bus master must use the I/O ports defined by the IPC. The ROM that contains the bootstrap and diagnostic programs, since it is enabled through a control port accessible only to the IPC, is not directly accessible to any other bus master (access to the ROM-resident monitor program is unrestricted).

In the ensuing discussions, references will be made to individual sheets of the integrated processor card schematic contained in the *Intellec Series II Schematic Drawings*, Manual Order Number 9800554.

3.1 MASTER PROCESSOR SUBSYSTEM

The master processor subsystem is the computational and control center for the IPC. As shown in figure 3-2, the major components of the master processor system are an Intel 8085A-2 microprocessor, an Intel 8219 bus controller, two Intel 8283 octal latches and an Intel 8287 octal transceiver.

3.1.1 MASTER PROCESSOR

The master processor for the IPC is an 8085A-2 microprocessor that is operated at 4.0 MHz (the microprocessor is shown on sheet 4 of the schematic). The 8085 is used in a conventional manner (refer to the *MCS-80/85™ Family User's Manual*, order number 121506 for an operational description) with the following exceptions: the HOLD, RST and TRAP functions are not used, one wait state is inserted into each memory read (or instruction fetch) cycle, and two wait states are inserted into every memory write cycle.

To establish the wait states, the READY input to the 8085 is controlled by ready flip-flop A69 (shown on sheet 5 of the schematic). At the beginning of any instruction cycle, the flip-flop is held in its reset state (READY inactive) by the absense of any of the OR'ed acknowledge signals (RAM ACK/, local acknowledge, TO ACK, etc.). During a memory read or write cycle, the clear from the ready flip-flop is removed by the RAM advanced acknowledge (RAM AACK/) signal. The flip-flop is subsequently clocked set on the next falling edge of CLK OUT/ (the inverted 8085 clock output signal) provided that the ADEN/ output from the bus controller (A96 on sheet 4) is active. As explained in section 3.1.2, bus controller operation is initiated on the falling edge of ALE during state T1 of a machine cycle, and ADEN/ goes active when the bus controller acquires the bus and remains active until another bus master requests the bus or a Halt instruction is executed. The memory read (MRD/) and memory write (MWT/) signals from the bus controller (which cause the RAM controller on sheet 10 to initiate a RAM access cycle) are displaced by one CPU clock cycle. Referring to figure 3-2, the WR input at the bus controller occurs one-half clock cycle later in state T2 due to the action of the delayed write flip-flop, while the RD input occurs one-half clock cycle earlier on the trailing edge of ALE in state T1 (ALE clocks the advanced read flip-flop set for both opcode fetch and memory read machine cycles). The timing of the RAM AACK/ signal from the RAM controller and

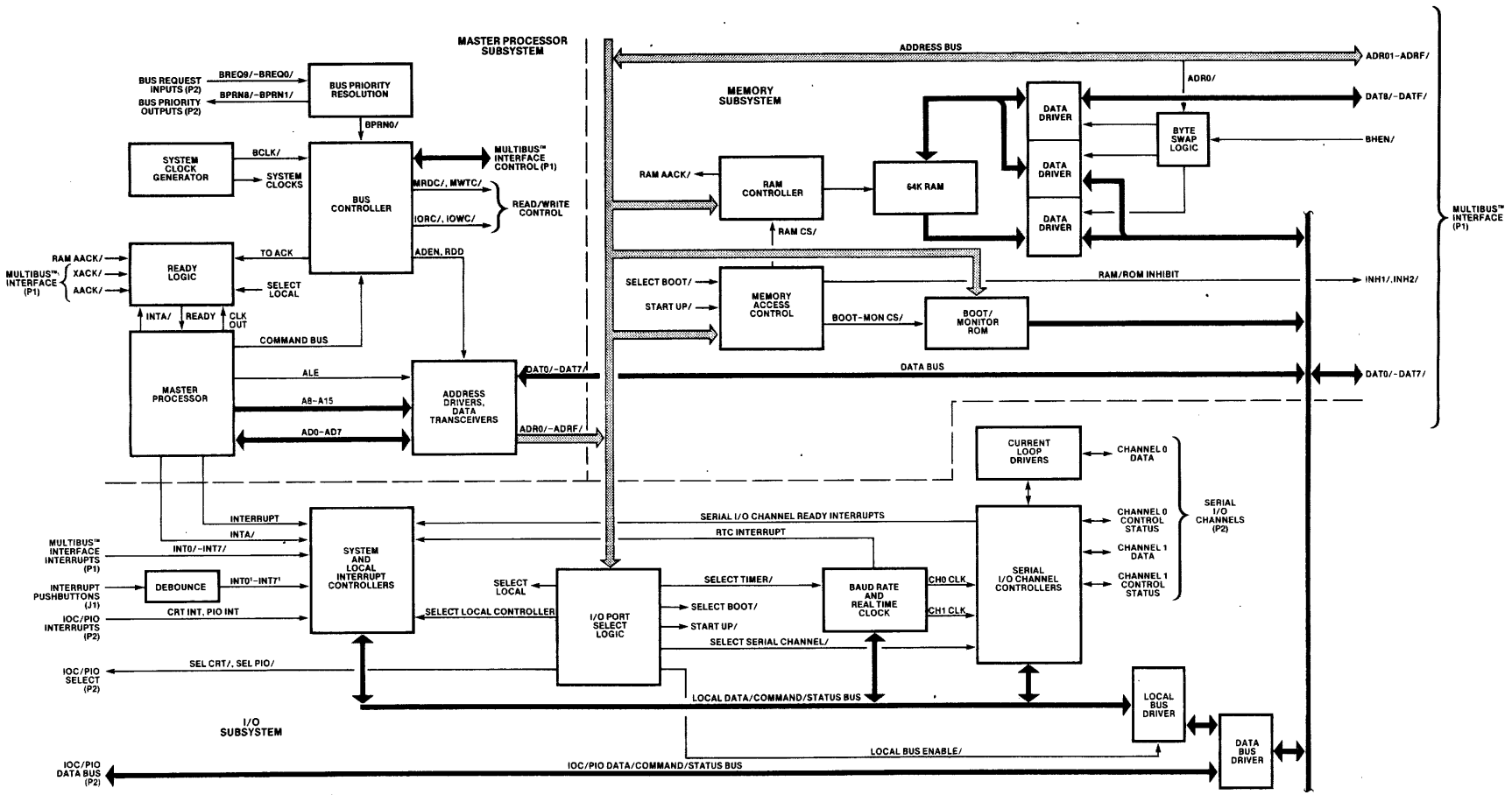


Figure 3-1. IPC Simplified Block Diagram

556-15

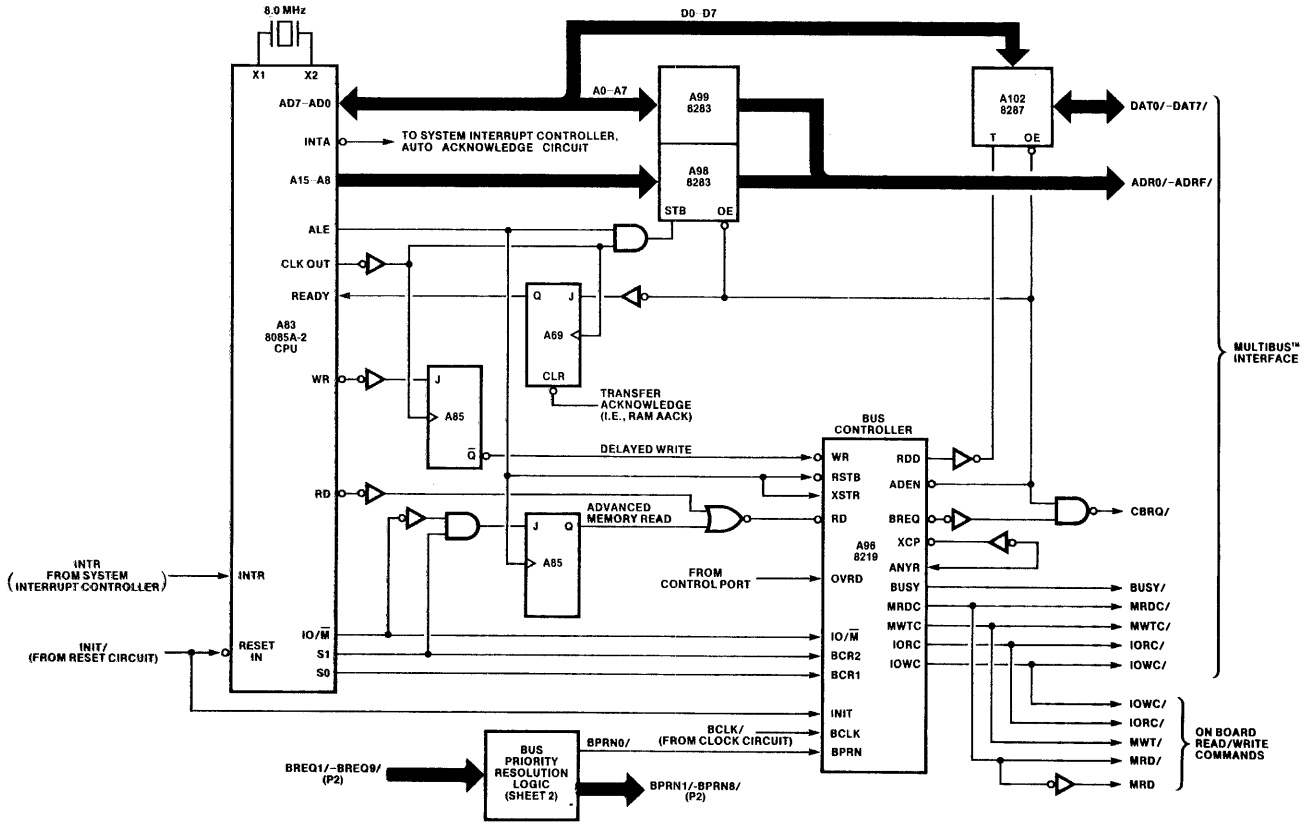


Figure 3-2. Master Processor Subsystem Block Diagram 556-16

the one clock cycle difference between the MRD/ and MWT/ signals causes one wait state to be inserted into each memory read (or opcode fetch) cycle and two wait states to be inserted into each memory write cycle. The other acknowledge signals gated to the clear input of the ready flip-flop (the output from local acknowledge timer, TO ACK and, RAM XACK/) are used to insert wait states into machine cycles when required by internal I/O transfers or external I/O or memory access operations. The output from the local acknowledge timer (binary counter A47 and flip-flop A69) on sheet 5 is enabled for all local I/O transfers (transfers between the serial I/O channels, the programmable devices on the IPC, the IOC and the PIO subsystem) and generates an acknowledge signal approximately 2.0 microseconds following the command. TO ACK (timeout acknowledge) allows the 8085 to complete a machine cycle when the expected command acknowledge is not received within approximately 10 milliseconds. RAM XACK/ is the normal (non-advanced) acknowledge signal from the RAM controller. Since the IPC's ready logic uses RAM AACK/, RAM XACK/ is not required by the IPC. RAM AACK/ and RAM XACK/ do however occur simultaneously when the RAM controller is in the "delayed SACK mode" (when a RAM access cycle is requested during a refresh cycle). The output from the local acknowledge timer, TO ACK or RAM XACK/ generates the Multibus transfer acknowledge signal (XACK/) when another bus master uses the IPC's resources. One additional signal that is gated to the ready flip-flop's clear input is INTA/ (interrupt acknowledge). This signal provides the required ready indication to the 8085 during interrupt acknowledge machine cycles.

3.1.2 BUS CONTROLLER

The 8219 bus controller (sheet 4) provides the bus acquisition, timing and control functions for IPC-initiated bus cycles.

3.1.2.1 BUS ACQUISITION. When the IPC master processor requires use of the system bus (all 8085 machine cycles except Halt require bus access), the bus controller secures access to the bus and maintains bus access until another bus master requests the bus (the IPC is the lowest-priority bus master) or until the master processor instructs the bus controller to release the bus by executing a Halt instruction.

To initiate a bus acquisition cycle, the bus controller requires an active level on either (or both) the BCR1 and BCR2 (bus control) inputs *and* an active level on the RSTB/ (request strobe) input. Referring to sheet 4, the BCR1 and BCR2 inputs are the S0 and S1 status outputs from the 8085, and the RSTB/ input is the buffered ALE output. Recalling the 8085's machine cycle timing, the status outputs (and ALE) go active at the beginning of state T1 of every machine cycle, and ALE returns to an inactive level (generating RSTB/) later in state T1. If another bus master is *not* requesting the bus, the BPRN0/ (bus priority in 0) input from the bus priority logic on sheet 3 will be active. On the third bus clock (BCLK/) cycle, if the bus is not in use (indicated by an inactive level on BUSY/) and if a bus request has not been received from another bus master, the bus controller activates BUSY/ and ADEN/ (address and data enable). BUSY/ is a bidirectional bus signal that, when active, indicates to all other bus masters that the bus is in use. ADEN/ is used locally by the IPC to enable the address latches (A98 and A99), the data transceivers (A102) and the ready flip-flop on sheet 5. Note that once the IPC has acquired the bus, the bus controller maintains bus access (holds BUSY/ active) until either another bus master requests the bus or until a Halt instruction is executed. By maintaining bus access, the IPC avoids the bus acquisition overhead when no other bus master is requesting bus access. If the bus is in use (if BUSY/ is low) or if a bus request is pending (if BPRN0/ is high) when the 8085 requests the bus, the bus controller delays the generation of BUSY/ and ADEN/ until one BCLK/ cycle after both the bus becomes available and no requests are pending.

3.1.2.2 BUS RELEASE. Once the master processor has acquired the bus, it surrenders the bus only when another bus master requests the bus or when the 8085 master processor executes a Halt Acknowledge machine cycle. When another bus master requests the bus, the bus controller completes its current bus control cycle and releases the bus (inactivates BUSY/) on the next bus clock cycle. When a Halt Acknowledge machine cycle is executed, the bus controller releases the bus on the third bus clock cycle.

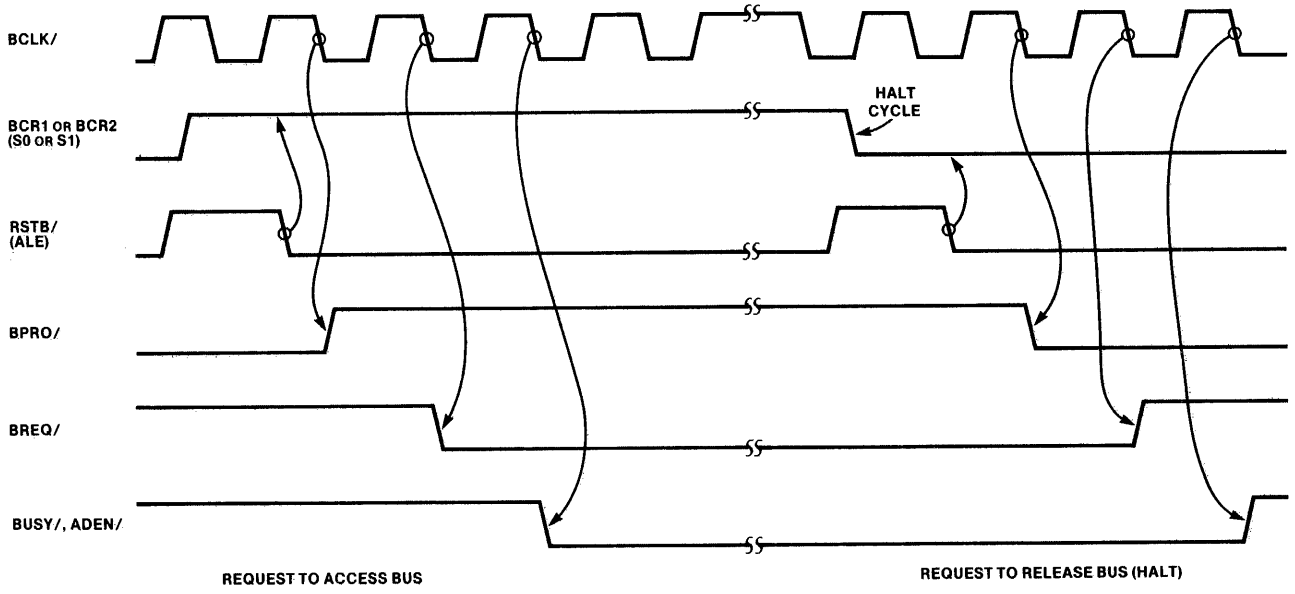


Figure 3-3. Bus Acquisition Timing 556-17

3.1.2.3 BUS OVERRIDE. The bus controller has an OVRD (override) input that, when active, allows the 8085 master processor to maintain bus access between bus cycles by preventing the recognition of a bus request from another bus master. Referring to sheet 6, the OVRD input is taken from bit 6 of the IPC's control port. OVRD is automatically activated when the system is initialized to prevent any other bus master from accessing the bus during the master processor's initialization/start-up procedure (following initialization, the Monitor program inactivates OVRD). During system operations, when the 8085 master processor must maintain bus access for a critical code section (e.g., when performing a test and set operation), the program would first activate OVRD through the control port.

3.1.2.4 BUS CONTROL CYCLE. The bus controller requires both the receipt of a transfer start request (XSTR) and the acquisition of the bus (BUSY/ and ADEN/ active) before it can begin a bus control cycle. The XSTR input to the bus controller is the buffered ALE output from the 8085 (XSTR is active on the leading edge of ALE, while RSTB/ is active on the trailing edge). Consequently, if the bus controller already has access to the bus, the control cycle is initiated by XSTR; if the bus controller does not have access to the bus, the control cycle is initiated when the bus is acquired (when BUSY/ and ADEN/ go active).

For all machine cycles that require bus access, the 8085 uses IO/M and either RD or WR to define the bus controller command to be generated for the control cycle. When either the bus controller's RD or WR input goes active, the bus controller activates ANYR (any request). If a read operation is specified (RD active), the bus controller additionally activates RDD. ANYR is externally connected to the bus controller's XCP (transfer complete) input, while RDD is used to control the direction of the data bus transceivers (A102).

The bus controller uses an internal delay timer to establish the address set up and data hold times for the bus control cycle. When the bus control cycle is initiated, the delay timer is triggered. Following an approximate 100 ns delay (determined by the RC time constant at the bus controller's DLY ADJ input), the corresponding command output goes active. When the bus data transfer cycle is complete, the 8085 inactives its command output (RD or WR). At the bus controller, the loss of the command input

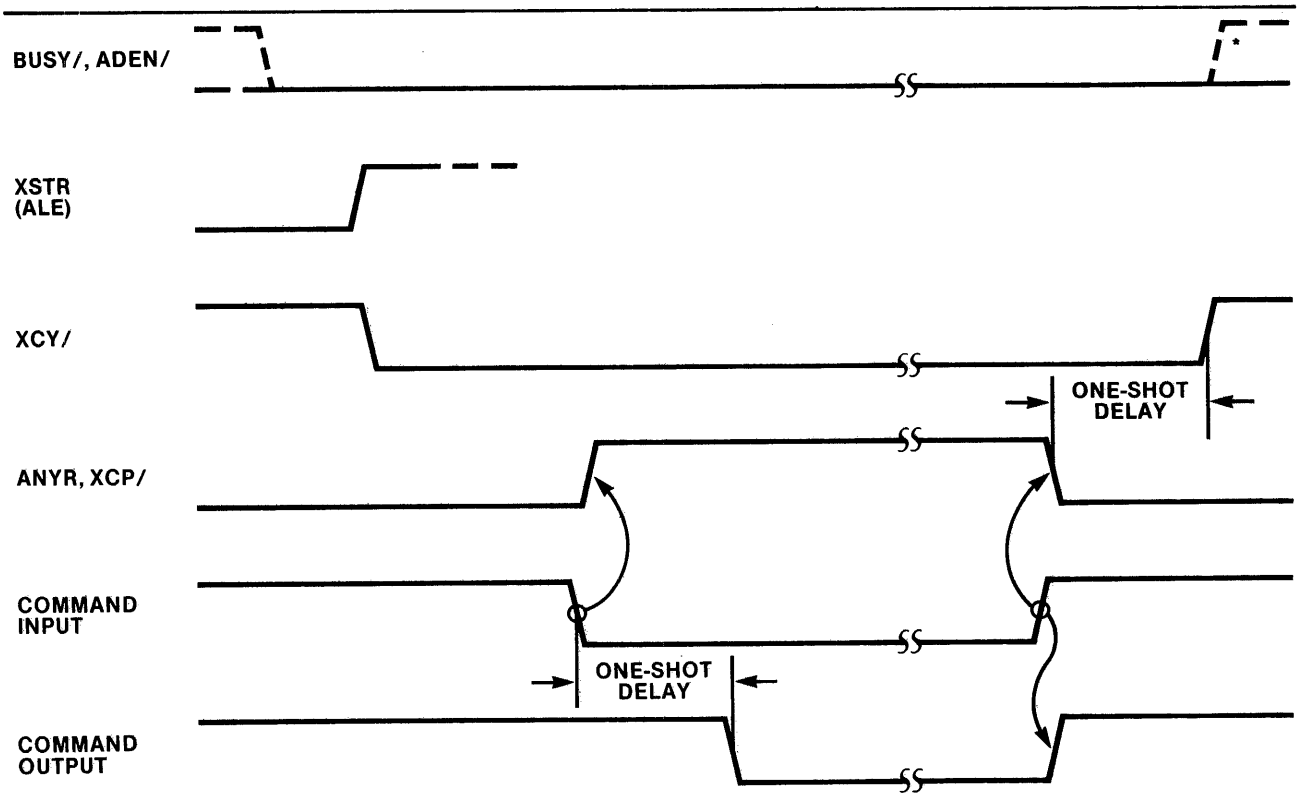


Figure 3-4. Bus Control Cycle Timing 556-18

causes ANYR and the corresponding command output to return to an inactive level. The negative transition of ANYR, at the XCP input, triggers the internal delay timer. During the 100 ns hold time interval, ADEN/ and BUSY/, remain active to hold the data bus transceivers enabled and to maintain bus access. (Another bus master can acquire the bus only after the timeout interval). Figure 3-4 illustrates the bus control cycle timing.

3.1.4 BUS PRIORITY RESOLUTION LOGIC

Bus priority resolution logic (sheet 2) includes a 74148 eight-to-three priority encoder and a 74138 three-to-eight decoder. The primary inputs to the encoder are bus requests BREQ1/ through BREQ8/. If there is more than one active bus request, the encoder only generates the binary code for the highest priority (highest numbered) request. The decoder uses the binary code from the encoder to activate the appropriate bus priority in (BRPNx/) signal to the requesting bus master.

The highest priority bus master and the lowest priority bus master (the IPC) are excluded from the priority resolution logic. The highest priority bus request (BREQ9/) is used to block all other bus requests and does not require the BRPN/ signal because it assumes control of the bus as soon as any current bus transfer cycle has been completed. The IPC continually requests use of the bus, but is inhibited by BPRN0/ if any other bus master is requesting the bus.

Routing of the BREQ/ and BPRN/ lines is arranged so that the priority of boards inserted in the Intellec Series II development system is increased toward the bottom of the chassis (including the expansion chassis). The IPC has the lowest priority since it can only be installed in the upper-most slot.

3.2 MEMORY SUBSYSTEM

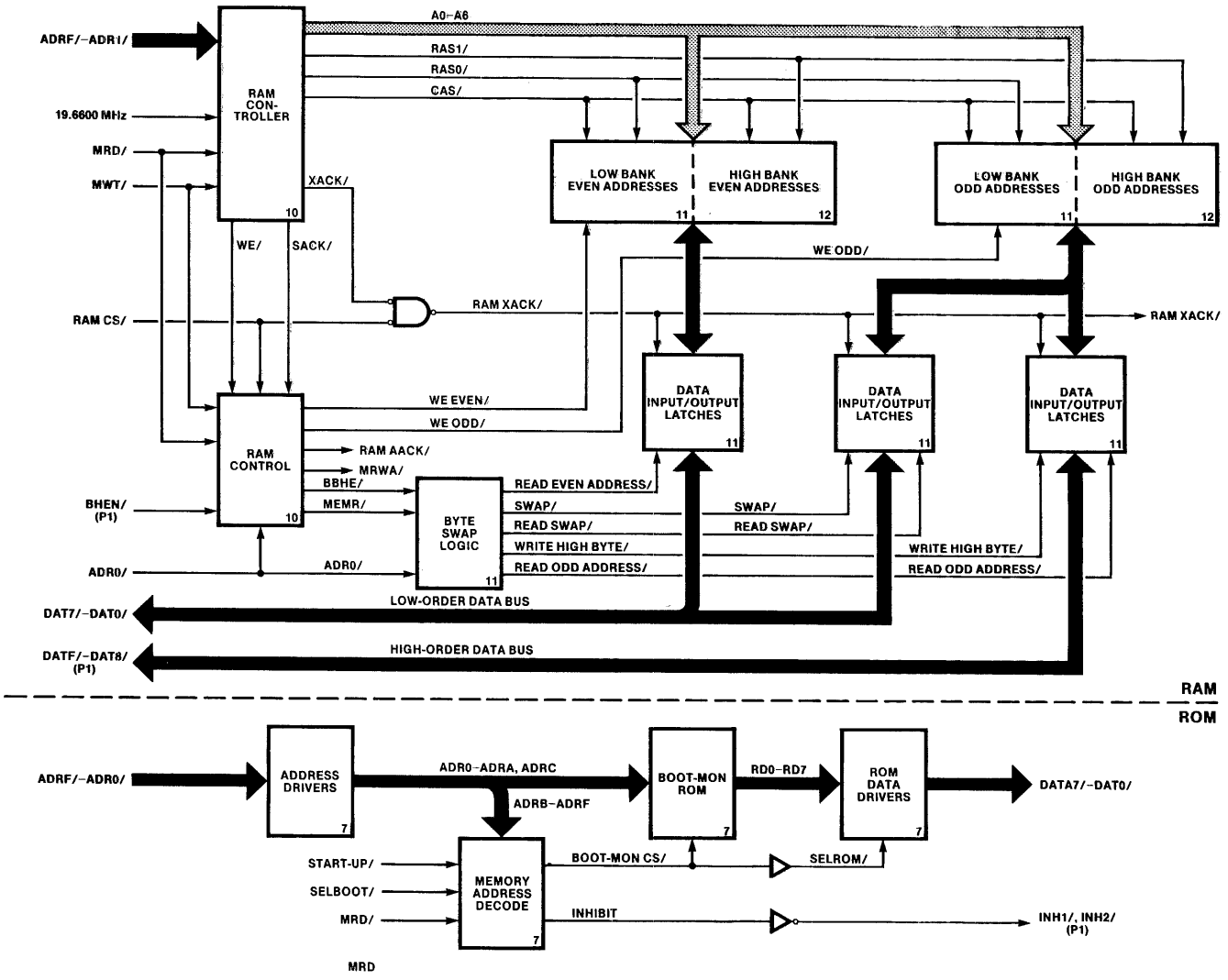
The memory subsystem of the IPC (figure 3-5) controls 64k bytes of random access memory (RAM) and 4k bytes of read-only memory (ROM). Because the 16-bit addresses output by the IPC master processor can specify a maximum of 64k locations, the two types of memory (ROM and RAM) share the same address space. The technique that permits address space sharing is known as shadowing ROM wherein ROM is preferentially selected unless disabled. Operationally, the use of shadow ROM is unnoticeable because the ROM programs are either called by name or are accessed through use of operator procedures that do not require knowledge of the program locations. From a hardware viewpoint, RAM and ROM are simply two distinct slaves to the master processor.

Major circuit elements of the memory subsystem include 32 16k-bit dynamic RAMs, a 4k-byte ROM, an 8202 RAM controller, a 3625A ROM (for memory address decoding), and a variety of TTL drivers and gates. All circuit elements are off-the-shelf components, and all but the TTL elements are Intel LSI products.

3.2.1 IPC ROM

Read-only memory (ROM) consists of two separate program segments contained in a single 4k ROM chip. One segment contains the Monitor program, and the other segment contains the bootstrap-loader/diagnostic program. The Monitor program is accessible to all bus masters and begins at address F800H. This segment cannot be shadowed, and the last 2k (locations F800H through FFFFH) of RAM cannot be used.

The boot/diagnostic ROM segment is only accessible to the IPC master processor. Individual bits within the master processor's control port (port FF) are set or reset in order to move, enable or inhibit the boot/diagnostic ROM segment. Referring to sheet 6, when the master processor is initialized (power on or reset), the control port (A84) is cleared (SEL BOOT/, START UP/ and OVRD active and interrupts disabled). The combination of START UP/ and SEL BOOT/, at the memory address decoder (A68) on sheet 7, establishes the starting location of the boot/diagnostic ROM segment at location 0H (the location from which the 8085 fetches its first instruction following reset) as well as at location E800H. The boot strap program immediately jumps into the ROM segment beginning at location E800H and inactivates the START UP/ output from the control port (by writing 02H to port FF). At the memory address decoder on sheet 7, the inactive level of START UP/ disables the ROM segment beginning at location 0H, and the boot/diagnostic segment is enabled only for addresses ranging from E800H to EFFFH. Disabling the ROM segment at 0H redefines the locations below E800H as RAM and makes this area of memory available for diagnostics. Following execution of the boot/ diagnostic program, control is transferred to the Monitor program at F800H. The Monitor program, when it begins execution, inactivates the SEL BOOT/ output from the control port (by writing 04H to port FF). At the memory address decoder, the inactive level of SEL BOOT/ disables the ROM's chip select input for addresses ranging from E800H through EFFFH (the boot/diagnostic ROM segment "disappears" from the address space), and all addresses below F800H are redefined as RAM. Note that provision is included in the



NOTE: The number in the lower right-hand corner of each block identifies the sheet number of the IPC schematic

Figure 3-5. IPC Memory Subsystem Block Diagram 556-19

Monitor program to reenable the diagnostic portion of the boot/diagnostic ROM segment. Entering the Monitor's Z$ command transfers control to the diagnostic (the Z command subroutine reactivates the SEL BOOT/ output by writing 0CH to port FF).

The control port also includes an interrupt enable bit. This bit is cleared (interrupts disabled) when the IPC is initialized and is subsequently set (interrupts enabled) when the Monitor is entered. The Monitor's IOC and PIO interface driver routines automatically clear the interrupt enable bit when called (to prevent an interrupt from being serviced during a sequence-dependent section of code).

The interrupt enable bit is used to enable or deny the interrupt (INTR) output from the system interrupt controller to the master processor. By controlling interrupts in this manner, the system interrupt controller and the master processor's internal interrupt flip-flop are not disturbed when using the Monitor's I/O routines to access system resources. The user program can use the master processor's DI (disable interrupt) and EI (enable interrupt) instructions without regard for Monitor operations.

The input to the control port on sheet 6 is a hexadecimal code appearing on data lines D0/ through D3/. The three least-significant bits of this code select one of four control outputs; the most-significant bit either sets or resets the selected output. The four outputs, the function of each output, and the related hexadecimal codes are outlined in table 3-1.

Table 3-1. Control Port Decoding

Name	Function	Set Code	Reset Code
SEL BOOT/	Boot/diagnostic ROM selection	0CH	04H
START UP/	Boot start address= 0000 (not E800)	Illegal	02H
Override	Inhibit other Multibus users	09H	01H
Interrupt Enable	Disable interrupts	0DH	05H

* Code input levels are active low (low = 1)

Referring again to the memory address decoder on sheet 7, when the decoder selects the ROM, it also enables ROM data drivers A103 (SEL ROM/ active) to place the addressed data byte on the bus and activates INHIBIT to generate the bus inhibit signals (INH1/ and INH2/) and to disable the on-board RAM (RAM CS/ inactive).

3.3.2 IPC RAM

The 64k-byte IPC RAM is implemented with 32 Intel 16k-by-1 bit dynamic RAMs that are partitioned into four 16k-byte banks and organized as shown in figure 3-5. The two low banks contain the odd and even address locations ranging from 0H to 7FFFH, and the two high banks contain the odd and even locations ranging from 8000H to FFFFH (remember that RAM locations F800H-FFFFH cannot be accessed). Selection of an individual bank is determined by the states of the low- and high-order address bits (ADR0 selects either an odd or even bank and ADRF selects either a low or high bank). The remaining 14 address bits (ADR1-ADRE) are multiplexed by the RAM controller to form a 7-bit row address and a 7-bit column address. In other words, each of the eight RAM chips in a bank forms a 128-by-128 matrix that stores a specific bit position for 16k bytes. The organization and implementation of IPC RAM allow a bus master with a 16-bit wide data bus to access two "aligned" memory locations in a single transfer cycle using both the low-order (DAT0/-DAT7/) and high-order (DAT8/-DATF/) data buses. The 8085 master processor of the IPC, since it has an 8-bit wide data bus, cannot perform 16-bit word transfers and is restricted to using the low-order data bus for all data transfers.

All RAM access cycles, whether initiated by the IPC's master processor or by another bus master, are controlled by an Intel 8202 RAM Controller. The RAM controller is selected for an access cycle by the RAM CS/ signal from the memory address decoder on sheet 7. This signal is active when the address on the bus is within the range of 0H through F7FFH. The read or write access cycle is initiated by MRD/ or MWT/, respectively. The RAM controller, on detecting an active level at either its RD or WR input, starts the access cycle by multiplexing the seven address bits at its AL0 through AL6 inputs to the banks as the 7-bit row address and activating SACK/ (system acknowledge) and RAS0/ or RAS1/ to strobe the row address into either the low banks (RAS0/ active) or high banks (RAS1/ active). If a write operation is specified (WR input active), the RAM controller activates WE/ (write enable). This signal is used by the RAM control logic to generate either WE EVEN/ or WE ODD/ according to the state of the low-order address bit (ADR0). The RAM controller then multiplexes the seven address bits at its AH0 through AH6 inputs to the banks as the 7-bit column address and activates CAS/ (column address strobe) to access the addressed byte. At the conclusion of the CAS/ interval (approximately 250 ns), the RAM controller activates XACK/ (transfer acknoweldge). This signal is used to latch the data byte accessed into the selected input (read) latch. Note that if a RAM access cycle is requested during a refresh cycle, SACK/ is delayed and occurs coincident with XACK/. Figure 3-6 illustrates a basic RAM access cycle.

The RAM controller uses an internal refresh timer to initiate a refresh cycle once every 14.6 microseconds. During a refresh cycle, the RAM controller activates both RAS0/ and RAS1/ (without CAS/) to refresh one complete row (128 bytes) within each bank. An internal modulo-128 row counter is incremented with each refresh cycle so that the entire 64k bytes are refreshed within two milliseconds.

The RAM control logic (see figure 3-5) uses the low-order address bit (ADR0) to determine odd/even bank selection and uses both ADR0 and the BHEN/ (byte high enable) input from the Multibus to determine data byte routing between RAM and the low- and high-order data buses. Table 3-2 outlines the operation of the control logic for both read and write RAM access.

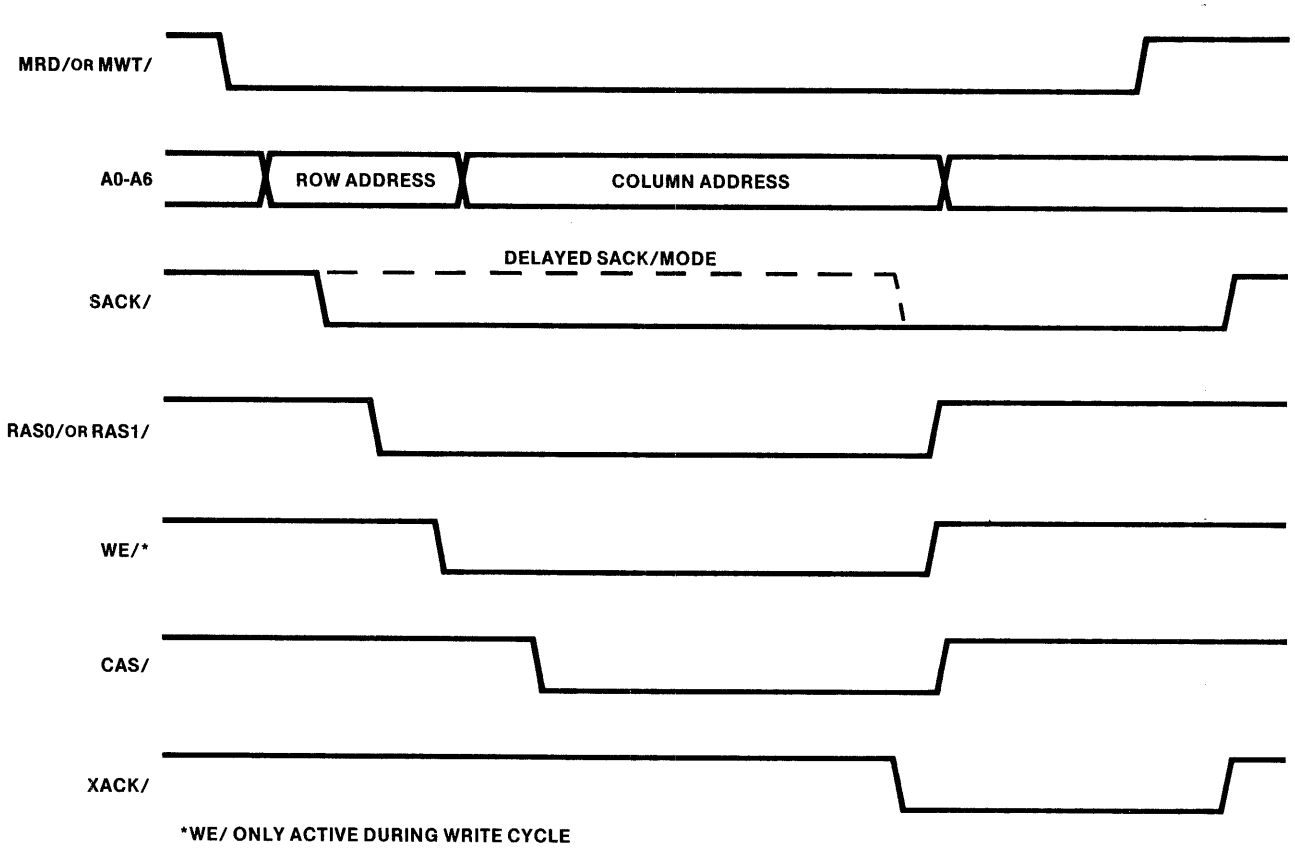


Figure 3-6. RAM Access Cycle

556-20

Table 3-2. RAM Control Logic Operation

Control Signal Levels		Banks Selected*	Data Lines Selected	Control Outputs Active	
ADR0	BHEN/			Read Access	Write Access
Low	High	Even	DAT0/-DAT7/	READ EVEN ADDRESS/	WE EVEN/
High	High	Odd	DAT0/-DAT7/	SWAP/ READ SWAP/	WE ODD/ SWAP/
Low	Low	Both	DAT0/-DATF/	READ EVEN ADDRESS/ READ ODD ADDRESS/ (WRITE HIGH BYTE/)	WE EVEN/ WE ODD/ WRITE HIGH BYTE/
High	Low	Odd	DAT8/-DATF/	READ ODD ADDRESS/	WE ODD/ WRITE HIGH BYTE/

*If ADRF is inactive (addresses 0H-7FFFH), the low bank is selected;
if ADRF is active (addresses 8000H-FFFFH), the high bank is selected.

3.3 IPC INPUT-OUTPUT SUBSYSTEM

The I/O subsystem of the IPC (figure 3-7) provides overall management of all Intellec Series II development system communications with the IPC master processor. Such communications include "system" I/O transfers to or from other optional Multibus boards and "local" I/O transfers. The system I/O transfers are accomplished via the Multibus interface.

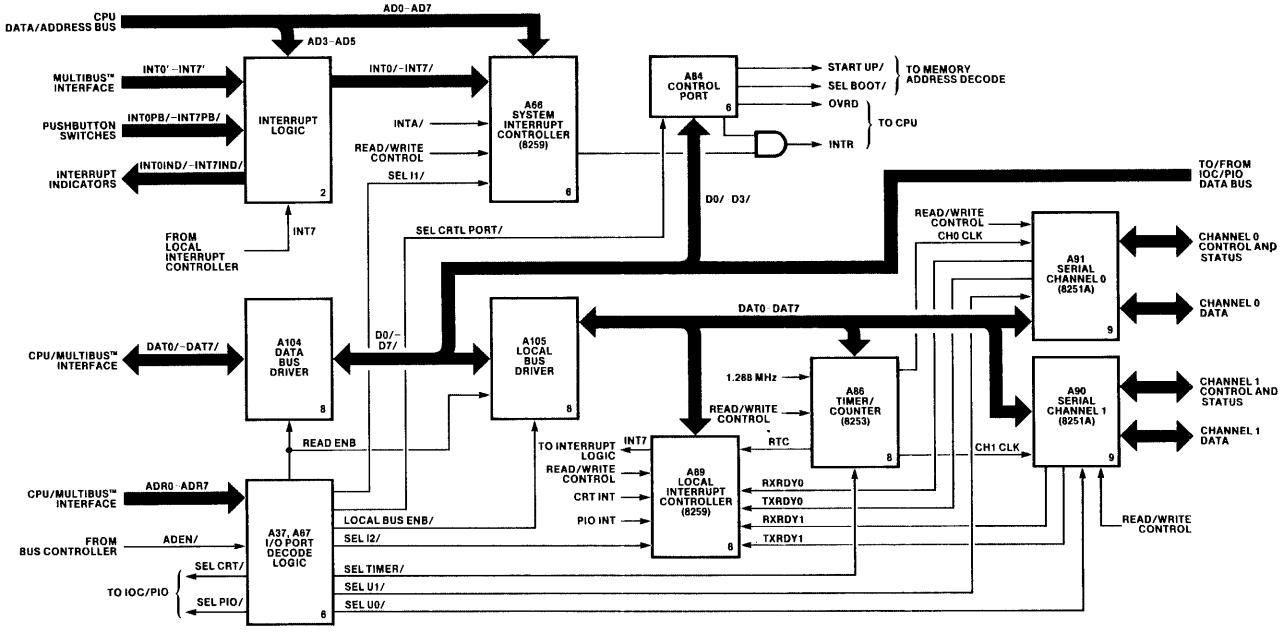
Local I/O transfers use the Multibus, but are accomplished via I/O ports of the master processor. The local I/O transfers are subdivided into external (PIO and IOC) transfers and internal I/O transfers. All internal I/O transfers are between the IPC master processor and the on-board programmable resources of the IPC.

Decoding of the internal and external I/O port addresses is performed by the decode logic on sheet 6. This logic decodes 19 distinct port addresses to produce ten control signals. The port addresses and associated chip select signals for both internal and external I/O control are shown in table 3-3.

One significant factor illustrated in table 3-3 is that the SEL LOCAL signal is generated for each of the port addresses used by both the internal and external I/O control circuits. This signal is applied to the local acknowledge timer (sheet 5). The timer enables master processor program execution 2.0 microseconds after the start of any local I/O transfer or local ROM access. The enabling of the various ROM segments is provided by the memory address decoder (section 3.2.1)

3.3.1 EXTERNAL (PIO/IOC) I/O TRANSFERS

The I/O subsystem of the IPC controls the routing of external I/O transfers to and from the PIO and IOC. In the case of external transfers, the I/O subsystem decodes the I/O port addresses and enables the common data lines of the PIO and IOC. The port addresses also cause the I/O subsystem to generate a select signal for either the PIO or the IOC. The select signals initiate communications between the master processor and the 8041 microprocessor of the PIO or the 8080A-2 microprocessor of the IOC. These external microprocessors are, in turn, responsible for communications with I/O devices (peripherals, integral disk, and CRT/keyboard). The IOC microprocessor is also directly responsible for issuing commands to, and receiving status from, the programmable chips of the IOC. Note that although the PIO and the IOC respond in a similar manner, they are completely independent of one another.



NOTE: The number in the lower right-hand corner of each block identifies the sheet number of the IPC schematic.

Figure 3-7. IPC I/O Subsystem Block Diagram 556-21

Table 3-3. I/O Port Address Decoding

Port Address	Device Selected	Port Function	Output Signal Active
F0	8253 Timer	Channel 0 Baud Rate	SEL TIMER/ LOCAL BUS ENB/ SEL LOCAL
F1		Channel 1 Baud Rate	
F2		Real Time Clock	
F3		Counter Mode Select	
F4	8251 Serial I/O Channel 0	Data	SEL U0/ LOCAL BUS ENB/ SEL LOCAL
F5		Command/Status	
F6	8251 Serial I/O Channel 1	Data	SEL U1 LOCAL BUS ENB/ SEL LOCAL
F7		Command/Status	
F8	PIO Subsystem	Data	SEL PIO/ SEL LOCAL
F9		Command/Status	
FA	8259 Local Interrupt Controller	Command/Status	SEL I2/ LOCAL BUS ENB/ SEL LOCAL
FB		Command/Status	
FC	8259 System Interrupt Controller	Command/Status	Select I1/ SEL LOCAL
FD		Command/Status	
FE	Reserved		
FF	Control Port	Access Control	SEL CRTL PORT/ SEL LOCAL
C0	Input/Output Controller	Data	SEL CRT/ SEL LOCAL
C1		Command/Status	
C2		Reserved	
C3		Reserved	

Programs that direct PIO or IOC activities use a specific port for a given purpose. For example, port C1 is used for IOC commands and DBB status, while all IOC data is transferred through port C0. (All status bytes other than the DBB status byte and all parameters are considered data.) The type of transfer to occur is indicated by the low-order address bit (the command/data signal) and IORC/ (the I/O read command) or IOWC/ (the I/O write command). Addressing port C0 or C1 activates SEL CRT/ to enable the IOC's data byte buffer.

NOTE

While the mnemonic SEL CRT/ would seem to imply association with the CRT only, this signal is also active for diskette and keyboard transfers controlled by the IOC.

If the port address is F8 or F9, SEL PIO/ is generated. This signal is applied directly to the chip select (CS) pin of the PIO's 8041 microprocessor. When SEL PIO/ is active, the data byte buffer of the PIO processor is enabled.

3.3.2 INTERNAL I/O TRANSFERS

The internal I/O control logic establishes communications paths between the master processor and the programmable chips of the IPC I/O subsystem. The process involved consists of the following sequence:

- Select chip by means of an I/O port address.
- Select command-out, status-in, data-out, or data-in transfer mode.
- Enable data lines between the chip and the master processor.
- Start 2.0 microsecond local acknowledge timer to enable the master processor to complete the transfer cycle (ready input active).
- Transfer the command, status, or data byte via the data bus.

The internal I/O transfer circuits decode I/O port addresses from the master processor. Each decoded port address causes the generation of a select signal that is used to enable a data transfer between the master processor and the programmable chip. Three types of data transfers occur:

- Output of a command (i.e., operating parameter) to the selected chip.
- Input of a status byte from the selected chip.
- Input or output of a data byte to or from the selected chip.

A total of five programmable chips are employed within the I/O subsystem. The functions of each chip are:

- Serial I/O Channel 0 (8251)—A universal synchronous/asynchronous receiver transmitter (USART) that establishes either a 20mA current loop or RS-232 interface between an external serial device and the IPC master processor. Commands permit full duplex asynchronous communications from 110 to 1200 baud for the current loop interface using seven bits plus parity for teletypewriter compatibility. The commands can also establish full duplex synchronous communications from 150 to 56k baud for compatibility with RS-232 or Bi-sync terminals or modems. The serial channel may use internal or external baud rate clocks as established by jumpers. Status bytes returned on command indicate the state of the current transfer and the occurrence of any transmission error. When the system is initialized, the boot program sets the channel for asynchronous operation with two stop bits, an 8-bit character length and a baud rate factor of 16X (the channel 0 external clock signal from the baud rate counter is 1.76 kHz which, divided by 16, provides a 110 baud rate).
- Serial I/O Channel 1 (8251)—Identical to serial I/O channel 0 with the exception that associated circuits are not available to establish the current loops required by teletypewriters. Serial channel 1 is also programmed for asynchronous operation with two stop bits, an 8-bit character length and a baud rate factor of 16X. The channel 1 external clock signal is 38.4 kHz and provides a 2400 baud rate.
- Baud Rate/Real Time Counters (8253)—A programmable timer consisting of three independent counters. Two of these counters establish baud rates for the serial I/O channels. The third counter is used to generate a 1 ms real-time clock signal. Command inputs establish both the frequency and the mode of operation for the timer/counter. The content of any counter is returned to the master processor on command. The real time clock is not used by basic operations of Intellec Series II development systems; it is reserved for user applications. When the system is initialized, all three counters are programmed for Mode 3 (square wave generator) operation and use a two-byte count register. The count register values and counter functions are as follows:

Counter	Function	Counter Value
0	Channel 0 Clock	698
1	Channel 1 Clock	32
2	1 ms Real Time Clock	1229

- System Interrupt Controller (8259)—A programmable interrupt controller that establishes priorities and masking for eight interrupt request lines (INT0-INT7). Highest priority is assigned to interrupt INT0. Each line may carry an interrupt request from either a front panel interrupt switch or from a Multibus user. Some Multibus devices such as ICE modules may employ one or more reserved interrupt requests to enable identification by software, but there is no distinction between a given Multibus interrupt request and the equivalent front panel switch. The lowest priority interrupt (INT7) is used by the IPC's local interrupt controller. The system interrupt controller is operated in a fully nested mode such that an interrupt request being serviced may be interrupted by a higher priority interrupt. Status returned on command includes the identity of the interrupt request being serviced, the identities of all interrupts currently nested, and the content of the internal mask register. All interrupts may be selectively enabled or disabled by the master processor program. Of the eight interrupts, INT0 and INT1 are reserved by the Monitor and ISIS, and INT7 is used by the local interrupt controller.

- Local Interrupt Controller (8259)—The local interrupt controller processes seven interrupt requests: a receive data and a transmit data interrupt request for each serial I/O channel, a PIO interrupt request, an IOC interrupt request, and a real time clock interrupt request. These interrupt requests are fully nested and an INT7 interrupt to the system controller is generated on any local interrupt. Upon the occurrence of INT7, the master processor program causes polling of the local interrupt controller to determine the origin of the interrupt request. A code identifying the highest priority interrupt request being serviced is returned to the master processor via the data lines. In all other respects, the local interrupt controller is functionally identical to the system interrupt controller. Interrupt assignment for the local interrupt controller is as follows:

  INT0 Serial I/O channel 0 input data ready
  INT1 Serial I/O channel 0 output data ready
  INT2 Serial I/O channel 1 input data ready
  INT3 Serial I/O channel 1 output data ready
  INT4 Real Time Clock
  INT5 PIO operation complete
  INT6 IOC operation complete
  INT7 Not Used

NOTE

The preceding Intel programmable chips and all of the other chips of the I/O subsystem are standard off-the-shelf products that are individually supported by documentation adequate for maintenance purposes. The following text assumes familiarity with the documentation and explains only the usage or interactions between these chips.

The I/O port address is used with either the I/O read or I/O write command to control the transfer of a command, status, or data byte between the programmable chips and a master processor. The IORC/ and IOWC/ signals only establish the direction of data flow; the I/O port addresses are responsible for chip selection and for varying degrees of chip control. In the case of the 8251 serial I/O chips, a separate port is required for the serial data and for the status and commands that monitor and control the data. Two port addresses are, therefore, employed. With 8259 interrupt controller chips, commands and status are transferred over two adjacent ports. In effect, the LSB of the port address (ADR0) augments the command concurrently appearing on the data lines. Augmenting of commands is also done for the 8253 baud rate and real time counters chip in which case the two LSBs of the port address (ADR0 and ADR1) are used to select one of the three counters or a control register within the 8253 chip. The data lines are then free to read or write counter values or to write to the control word registers that establish counter modes. Note that in addition to activating the appropriate chip select signal, addressing any of the I/O ports associated with internal I/O transfers (except the system interrupt controller) also activates LOCAL BUS ENB/. This signal, on sheet 8, enables the local bus drivers (A105).

The SELI1/ and SELI2/ outputs of the port decode logic are used to select the 8259 local interrupt controller (sheet 8) and the 8259 system interrupt controller (sheet 6), respectively. Both controllers use IORC/ to enable reading of status, IOWC/ to enable writing of commands, and ADR0 to augment commands appearing on the data lines. One difference between implementation of the two controllers is that the system interrupt controller is connected directly to the CPU's data bus (AD0-AD7), whereas the local interrupt controller is connected to the system data bus (DAT0/-DAT7/) via the local data bus drivers. Note that the chip select signal to the system interrupt controller is qualified by ADEN/ and therefore can only be addressed by the IPC's master processor.

3.3.1 INTERRUPT REQUEST CIRCUITS

The interrupt request circuits consist of the previously discussed system and local interrupt controllers plus a number of TTL chips. Most of the TTL chips are used to provide debouncing and latching for the front panel interrupt switches. Details of these circuits are shown on sheet 2. Inputs from the front panel interrupt switches are applied to the 74LS148 interrupt switch priority encoder. The encoder generates a three-bit binary code for the switch being pressed. If more than one switch is pressed, the binary code generated is that of the highest priority (lower numbered) switch. Any active switch input causes the GS output to be low and the EOUT output to be high. The GS output triggers the first (10 millisecond) pushbutton debounce timer, and the EOUT output enables the second (500 nanosecond) timer to be triggered by the first timer.

The binary output of the encoder is applied through the 74LS157 ID source selector multiplexer to the 74LS259 interrupt switch latch. The binary code (latch ID) is accepted by the interrupt switch latch when the second timer is triggered. When the inter-

rupt latch is enabled, the level present at the latch's D input is stored into the addressed latch. Referring to the schematic, the D input originates from the auto acknowledge counter (A6) and, since the counter is idle, will be at a logic high level. The interrupt switch latch thus stores a single bit that corresponds to the activated interrupt switch and, in effect, reconverts the binary code into a discrete interrupt signal. The interrupt latch outputs are wire OR'ed with the corresponding bus interrupts (INT0/- INT7/) at the inputs to the 74LS240 interrupt drivers.

During the second cycle of an interrupt acknowledge sequence (second INTA/), the AD3, AD4 and AD5 data bits of the low-order call address byte from the 8259 system interrupt controller (since the 8259 was initialized with a call address interval of 8) reflect the binary code of the interrupt switch currently being serviced. At this point in time, the auto acknowledge counter will be logically arranged to:

1. Multiplex the AD3 through AD5 data bits at the B inputs to the ID source multiplexer to the interrupt switch latch (1QB output from binary counter A6 logically high).
2. Set the D input to the interrupt switch latch to a logic low level in order to clear the addressed interrupt switch bit (inverted 1QB output from A6 logically high).
3. Enable the interrupt switch latch (2QA and 2QB output from A6 logically high to generate EN/) to clear the adressed interrupt (interrupt currenty being serviced) from the interrupt latch.

Attempting to reprogram the 8259 interrupt controller with a call address interval of 4 will cause the interrupt acknowledge logic to address the wrong bit within the interrupt switch latch.

CHAPTER 4
INPUT-OUTPUT CONTROLLER

The input-output controller (IOC) consists of all hardware elements required to establish IPB/IPC communications with the integral diskette drive and CRT terminal of the Intellec Series II development system chassis. The IOC circuit elements comprise the bulk of the IOC board. The IOC board also contains circuit elements of the PIO as discussed in Chapter 5.

In terms of chip population and overall organization, the IOC (figure 4-1) is similar to the IPB/IPC. Both contain a microprocessor that issues commands to, and receives status from, the programmable chips on their respective boards. However, in terms of the functions performed, the IOC is more closely related to the PIO. This is true because the IOC and PIO both execute ROM-resident device-control programs whereas data to or from the devices is processed solely by the IPB/IPC. The functional similarity of the IOC and PIO allows their use of an IPB/IPC interface that is virtually identical in terms of hardware signal sequences and the command and status formats employed.

The common IPB/IPC interface of the IOC and PIO is known as the data byte buffer (DBB). In the PIO, the DBB is implemented within the 8041 microprocessor chip. In the IOC, the DBB is implemented by means of discrete TTL elements. In both cases, the DBB is a bidirectional buffer that provides storage for one input byte and one output byte. Associated with the DBB are four flags:

- The input buffer full (IBF) flag indicates the presence of a new input byte.
- The output buffer full (OBF) flag indicates the presence of a new output byte.
- The command/data (C/D) flag identifies the byte as either control information (command or status) or device data (data).
- The busy (F0) flag indicates inability to accept a new IPB/IPC command (the current command is still being processed).

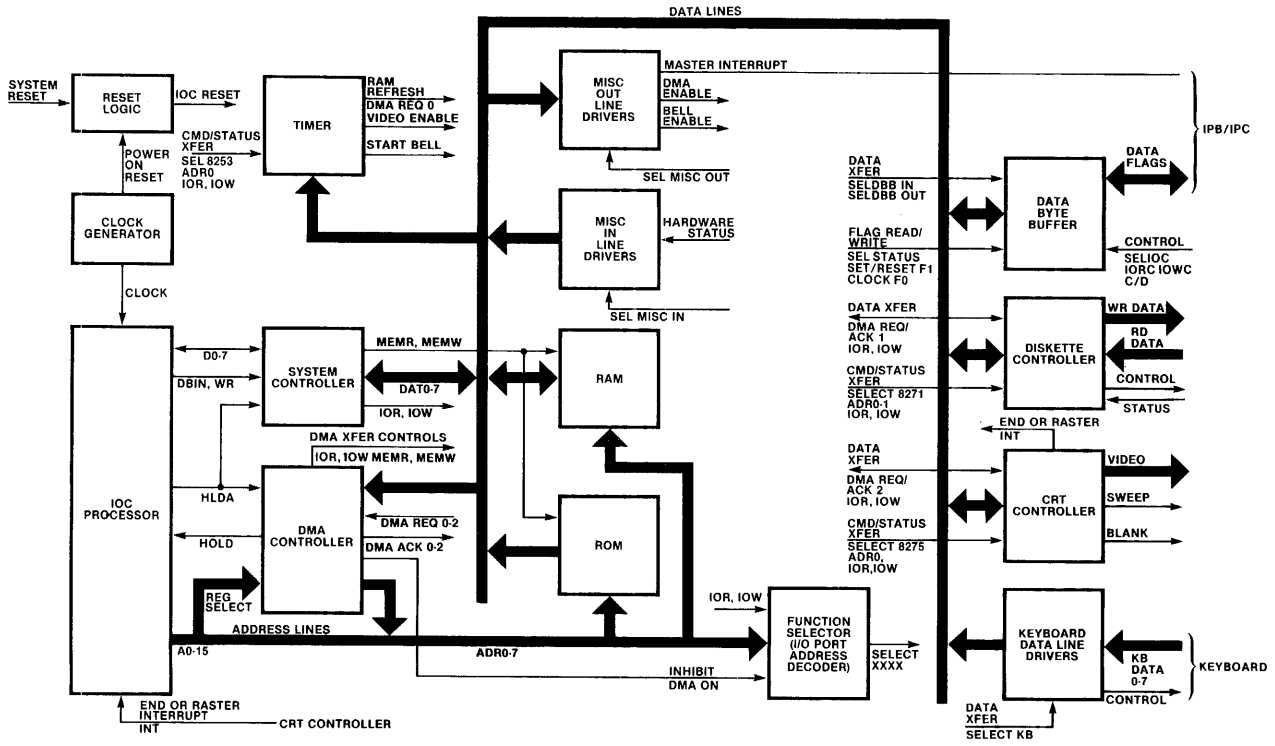


Figure 4-1. IOC Simplified Block Diagram

556-22

The DBB flags are used to alert the IOC when a command or data byte is input by the IPB/IPC and to indicate when a status or data output byte is accepted by the IPB/IPC. The flags are also used by the IPB/IPC to determine the status of IOC operations. The basic operational sequence of the IOC is as follows:

- Issuance of a command by the IPB/IPC.
- Acceptance of the command and associated input data bytes (if any) by the IOC.
- Polling of the DBB flags by the IPB/IPC.
- Acceptance of IOC output data or status bytes (if any) by the IPB/IPC.

All data, command, and status byte transfers between the IOC and IPB/IPC are controlled by the IPB/IPC, and each data or status byte transfer is in response to a specific command; the IOC cannot initiate a transfer to or from the IPB/IPC. However, a set of commands from the IPB/IPC can initiate a relatively complex sequence of events within the IOC. For example, commands from the IPB/IPC can establish the parameters for block transfers from diskette and then control the data byte transfers from the IOC to the IPB/IPC. The IOC uses the block parameters to accomplish DMA block transfers from diskette to the IOC RAM. The data byte transfers are then made from IOC RAM to the IPB/IPC. Writing to diskette is accomplished in the same manner except that the DMA block transfer occurs after all byte transfers from the IPB/IPC to RAM have been completed.

The only other data transfers via DMA are those to the CRT. In this case, DMA block transfers from RAM are used to perform CRT screen refresh. Byte transfers from the IPB/IPC to the CRT display segment of IOC RAM merely modify the data that is to be displayed. The DMA block for the CRT is of fixed size (enough to fill the screen), and blank areas of the screen consist of space characters. For reasons that will be explained later, the DMA block for the CRT is subdivided into two parts of variable size.

In contrast to the IPB/IPC master processor, the IOC processor makes use of only one hardware interrupt. This single interrupt is generated by the 8275 CRT controller chip at the end of the CRT display raster. The interrupt causes the IOC processor to determine the RAM addresses that are to be accessed for the next display raster. The new parameters (starting address and terminal count) for the first part of the display are output to DMA channel 2, while the new parameters for the second part of the display are output to DMA channel 3. DMA channel 2 provides the CRT display data, and when channel 2 reaches its terminal count, the parameters previously output to channel 3 are automatically transferred to channel 2. The lower part of the display, which contains the most recent entries, then appears on the CRT screen. DMA channels 0 (RAM refresh) and 1 (diskette data transfer) do not use hardware interrupts.

Aside from the four DMA channels, all communications with the IOC processor are accomplished via I/O ports. The I/O ports are used for communicating with the IPB/IPC processor via the DBB, for setting and resetting the DBB flags, for entry of data and status from the keyboard processor, for communicating with the programmable chips of the IOC, and for otherwise controlling IOC operations. The programmable chips include an 8257 DMA controller, an 8253 interval timer, an 8271 diskette controller, and an 8275 CRT controller.

Use of IOC I/O ports by the programmable chips is fixed by chip design, but the specific port address used for any chip function is established by IOC design. The I/O port addresses not only select the programmable chip, but also serve as command extensions that inform the programmable chip of the type of data, command, or status transfer that is to occur on the data lines.

In the remainder of this chapter, the sheet numbers called out in both the text and on the illustrations are references to individual sheets of the IOC schematic diagram included in the *Intellec Series II Schematic Drawings*.

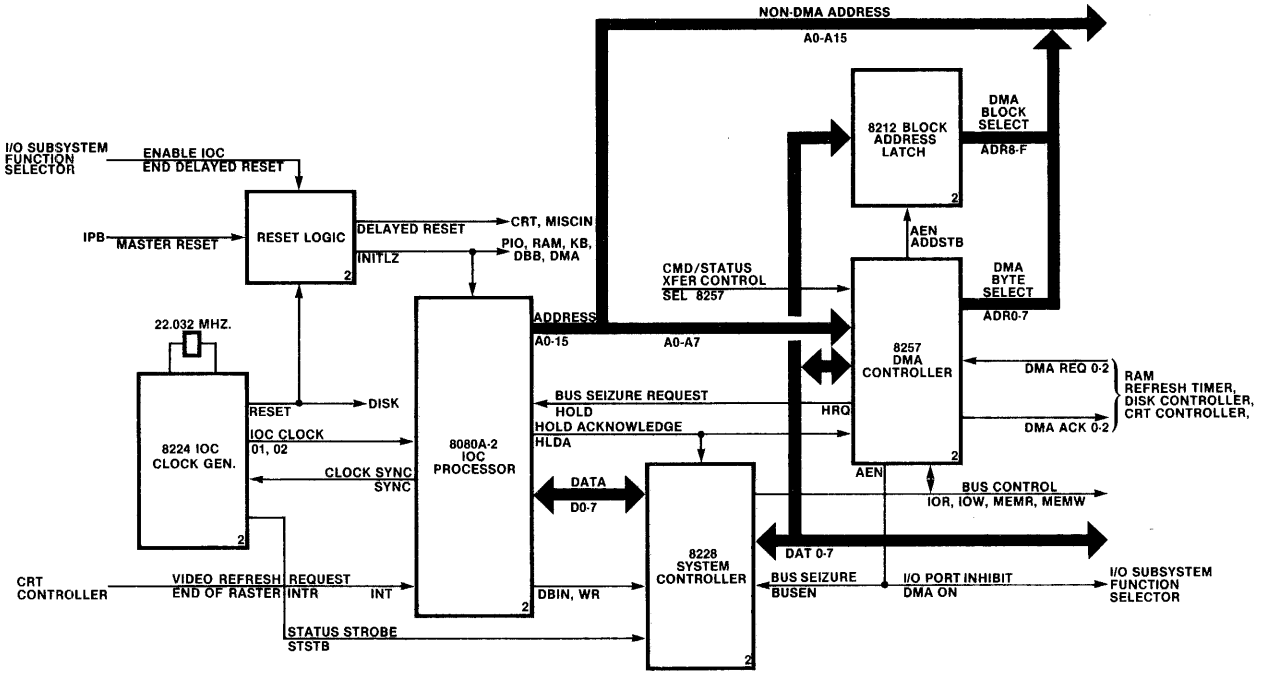
4.1 IOC PROCESSOR SUBSYSTEM

The IOC processor subsystem (figure 4-2) is a conventional arrangement of standard Intel LSI chips plus a few TTL components to provide signal gating. The chips include an 8080A-2 microprocessor, an 8224 clock generator, an 8228 system controller, an 8257 DMA controller, and an 8212 bidirectional 8-bit register that holds the eight most significant bits of the DMA controller's address to IOC RAM.

4.1.1 IOC PROCESSOR AND CLOCK GENERATOR

The 8080A-2 IOC processor and its associated 8224 clock generator are shown in detail on sheet 2. The clock generator employs a 22.032 MHz crystal to produce a 2.448 MHz (408 ns) clock that is used throughout the IOC. The clock generator also employs an external RC circuit to generate a power-on reset signal (RST). See paragraph 4.1.2 for details of the reset logic.

The IOC processor has an external connection between its WAIT and READY pins. With this arrangement, the processor provides a single wait state for every memory or I/O access. No acknowledge signal is required since all memory and



NOTE: The number in the lower right-hand corner of each block identifies the sheet number of the IOC schematic.

Figure 4-2. IOC Processor Subsystem Block Diagram 556-23

I/O elements of the IOC are able to accept (read) or provide (write) the accessed byte without the use of additional wait states.

To accommodate direct memory access (DMA), the IOC processor receives its HOLD signal from the DMA controller (HRQ) and responds by generating the hold acknowledge signal (HLDA). The HRQ signal is externally ANDed with the software-controlled DMA ENABLE signal to allow disabling of DMA during power-on or reset while the programmable chips are being initialized by the IOC processor. At all other times, DMA is enabled.

Use of the HOLD signal to delay processing during DMA cycles is a form of "cycle stealing." DMA is used for RAM refresh and for transfers between RAM and either the CRT controller or the diskette controller. Diskette transfers always occur in response to a command from the IPB/IPC in which case the cycle stealing involved does not delay "normal" processing. CRT and RAM refresh cause a lengthening of all IOC processes. RAM is refreshed at intervals of 15.5 microseconds, and the CRT requires a DMA transfer on the average of one transfer every 8.33 microseconds under worst case conditions. The two refresh operations use 38 percent of the available processing time. Diskette transfers via DMA occur once every 32 microseconds when required by an IPB/IPC command and use an additional seven percent of the processing time. Under worst case conditions, all other processing is allocated 55 percent of the processing time.

4.1.2 RESET LOGIC

The IOC reset logic (sheet 2) provides two types of hardware reset. The first type is generated when the clock generator detects application of power to the IOC board. The second type occurs when the front panel RESET switch of the IPB/IPC board is pressed and thereby causes the generation of the MASTER RESET/ signal. Both types cause the generation of the INITLZ/ signal that is applied to most IOC board hardware elements including the PIO processor. The power-on reset generates two additional reset signals (RESET and RESET/) to reset all hardware elements including the 8275 CRT controller and the 8253 timer (which are not reset by MASTER RESET/).

The reason the 8275 and 8253 chips are not reset by MASTER RESET/ is that this action would cause the CRT raster to collapse (as seen when CRT power is removed). The RESET switch thus removes

character displays from the CRT screen without disruption of the horizontal and vertical drive signals (HORIZ DRIVE and VERTICAL DRIVE/).

4.1.3 DMA CONTROLLER

The 8257 DMA controller is unique in that it is the only IOC chip (other than the IOC processor) that can assume full control of the IOC bus. The address lines of the processor and the data lines of both the processor and the IOC's 8228 system controller are floated while the DMA controller generates the RAM address and the control signals required to accomplish a data byte transfer between RAM and the requesting device. After gaining control of the bus, a single data byte is transferred during four clock cycles after which control of the bus is returned to the processor unless another DMA request is pending.

During normal (non-DMA) operation, the system controller decodes bus commands appearing on the processor's data lines (D0-D7) during state T2 to produce the memory read/write control signals (MEMR/ and MEMW/) and the input-output control read/write signals (IOR/ and IOW/). The bus control signal to be generated is determined during status strobe (STSTB) time. If a read operation is specified, the system controller immediately outputs IOR/ or MEMR/, while if a write operation is specified, the system controller waits for the processor to output WR/ before activating either IOW/ or MEMW/. During this time, the address lines (A0-A15) contain either a memory address (if MEMR/ or MEMW/ is active) or a port address (if IOR/ or IOW/ is active).

During system power-on or reset, all DMA channels are disabled and subsequently initialized via I/O ports F0-F8 of the IOC processor. Initialization consists of establishing the RAM starting address and the terminal count (i.e., block size) for each of the DMA channels that is to access RAM. Following initialization, a mode set command is sent to the DMA controller to establish the common operating modes of all channels and to selectively enable each channel. In the IOC, the operating mode selected uses the auto-load option wherein the starting address and terminal count supplied to DMA channel 3 are automatically transferred to channel 2 when the channel 2 terminal count has been decremented to zero (completion of the DMA block transfer). The auto-load mode is established by 8257 logic and cannot affect, or be duplicated by, DMA channels 0 or 1.

Block parameters are supplied to DMA channels 1, 2, and 3 when the channels are initialized. Channel 0 does not require block parameters since the channel only steals machine cycles to perform RAM refresh cycles. To further clarify use of the four DMA channels, details of DMA controller operation are provided in the following text in relationship to RAM refresh, integral diskette transfers, and CRT transfers.

DMA channel 0 performs a RAM refresh cycle once every 15.5 microseconds as established by the 8253 timer. The timer output, DMA REQ 0, is the highest priority DMA request. The DMA controller generates a hold request (HRQ) to gain access to the IOC bus. When the IOC processor has completed the current instruction cycle, it generates hold acknowledge (HLDA) and essentially disconnects itself from the data lines (D0-D7) and the address lines (A0-A15). The system controller accepts HLDA and floats its output data lines (DAT0-DAT7).

Upon sensing HLDA, the DMA controller generates DMA ACK 0/ followed by MEMR/ to cause the RAM to execute one read cycle refresh. During this refresh cycle, the DMA controller becomes the IOC bus master and performs a normal data transfer cycle (an address is placed on the address lines). However, the RAM, having begun a refresh cycle, ignores the DMA address. The content of the data lines is also ignored since AEN (DMA ON) disables port address decoding and since no other DMA user is enabled by DMA ACK 0/. Once initialized during power-on or reset, DMA channel 0 is not reinitialized.

DMA channel 1 is initialized prior to any block transfer to or from diskette. If the block transfer is from diskette (disk read), initialization and the associated block transfer are started as soon as the appropriate IPB/IPC command is received. If the block transfer is to the diskette (disk write), the block parameters are sent to the DMA controller, but their use awaits completion of the data byte transfers from the IPB/IPC to IOC RAM. (Block parameter commands are always issued prior to the commands that provide byte transfers to or from the IPB/IPC.) Initialization of channel 1 is accomplished by the IOC processor through I/O ports F2, F3, and F8. These ports select the starting address, establish the terminal count, and set modes and enable channels of the DMA controller, respectively. The IOC processor additionally uses port F8 to check DMA controller status (all channels).

NOTE

The IOC only controls diskette transfers to or from the single integral diskette drive.

During initialization, the DMA controller is controlled by the IOC processor in the same manner as any other programmable chip. The four MSBs of port addresses F2, F3, and F8 are decoded by the 8205 function selector of the I/O subsystem to

produce SELECT 8257/. Concurrently, the four LSBs of the port address are directly decoded by the DMA controller. The starting address, terminal count, and mode set parameters are accepted from the data lines as each port is addressed. Two successive byte transfers occur for the 16-bit starting address and terminal count words.

NOTE

The terminal count consists of 14 bits that establish a maximum block size of 16k. The two MSBs of the terminal count word establish the subsequent DMA transfer as being either to or from diskette. Block sizes are multiples of 128 for diskette and 2000 (maximum) for the CRT.

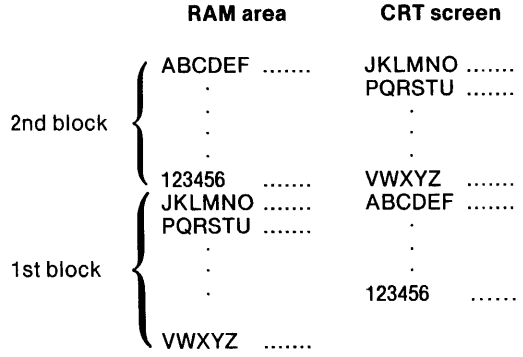
After the DMA controller has been initialized for diskette operations, the 8271 diskette controller is initialized. This generally consists of specifying exactly what area of the diskette is to be accessed. Upon completion of its search for the specified area, the diskette controller issues DMA REQ 1. The DMA controller then generates HRQ and becomes the IOC bus master when the processor responds by issuing HLDA. The IOC processor and the system controller are then effectively disconnected from the data and address lines. The DMA controller then places the eight MSBs of the starting address on the data lines and generates ADDSTB to cause the 8212 to latch this half of the RAM address. Concurrently, the DMA controller activates DMA ACK 1/ to inform the diskette controller that it has been selected for the DMA cycle.

The DMA controller generates either IOR/ and MEMW/ (for diskette read) or IOW/ and MEMR/ (for diskette write) based on the two MSBs of the previously received terminal count word. The RAM then accesses the selected address and a byte transfer occurs between the RAM and the diskette controller.

The DMA controller then increments the DMA channel 1 address and decrements the channel 1 terminal count to prepare for the next DMA REQ 1. When the terminal count reaches zero, the DMA controller status for channel 1 is updated to inform the IOC processor that the complete block has been transferred to or from diskette.

DMA channels 2 and 3 operate together to refresh the CRT display at a rate of 50 or 60 Hertz (depending on the line frequency). Channel 2 is directly responsible for the transfer of display characters from RAM to the CRT controller. Channel 3 is used to store new block parameters (starting address and terminal count) for channel 2. The channel 3 block parameters are automatically loaded into channel 2 when the terminal count for channel 2 decrements to zero. The transfer of parameters from channel 3 to channel 2 constitutes an auto load.

The main advantage of the auto load feature is that scrolling of the CRT screen can be accomplished without rewriting the RAM area containing the CRT data. (Scrolling causes the CRT display to move up with each line feed as a new line is started at the bottom of the screen.) With the auto load feature, a given RAM address remains associated with a specific character on the CRT screen, even as that character moves up the screen as a result of scrolling. Two blocks of RAM addresses are output to the DMA controller. The first of these blocks defines the area of the RAM from the oldest line of text (top line of CRT screen) to the end of the RAM buffer used for CRT display. The second block defines the area of RAM from the beginning of the RAM buffer to the newest line of text (bottom of the CRT screen). The relative position of display data in the RAM buffer is illustrated below:



As lines are added to the text, the beginning address of the first block and the last address of the second block are changed to reposition the display data while allowing fixed program references to the RAM data. When the screen is completely filled, each new line clears and overwrites the oldest line. This operation necessitates recycling of the variabale block addresses back through the beginning of the CRT table.

At power-on or reset, the starting address of the CRT table and a terminal count of 2000 (25 rows of 80 characters) minus 1 are output to DMA channel 2. Concurrently, the entire CRT table is cleared so that the CRT screen displays only the cursor. Each time the CRT beam reaches the lower right-hand corner of the screen, the CRT controller generates an end of raster interrupt (END OF RASTER INTR). This interrupt causes the IOC processor to establish the pair of block parameters based on the position of the newest line of data within the CRT table. The parameters of the first block are output to DMA

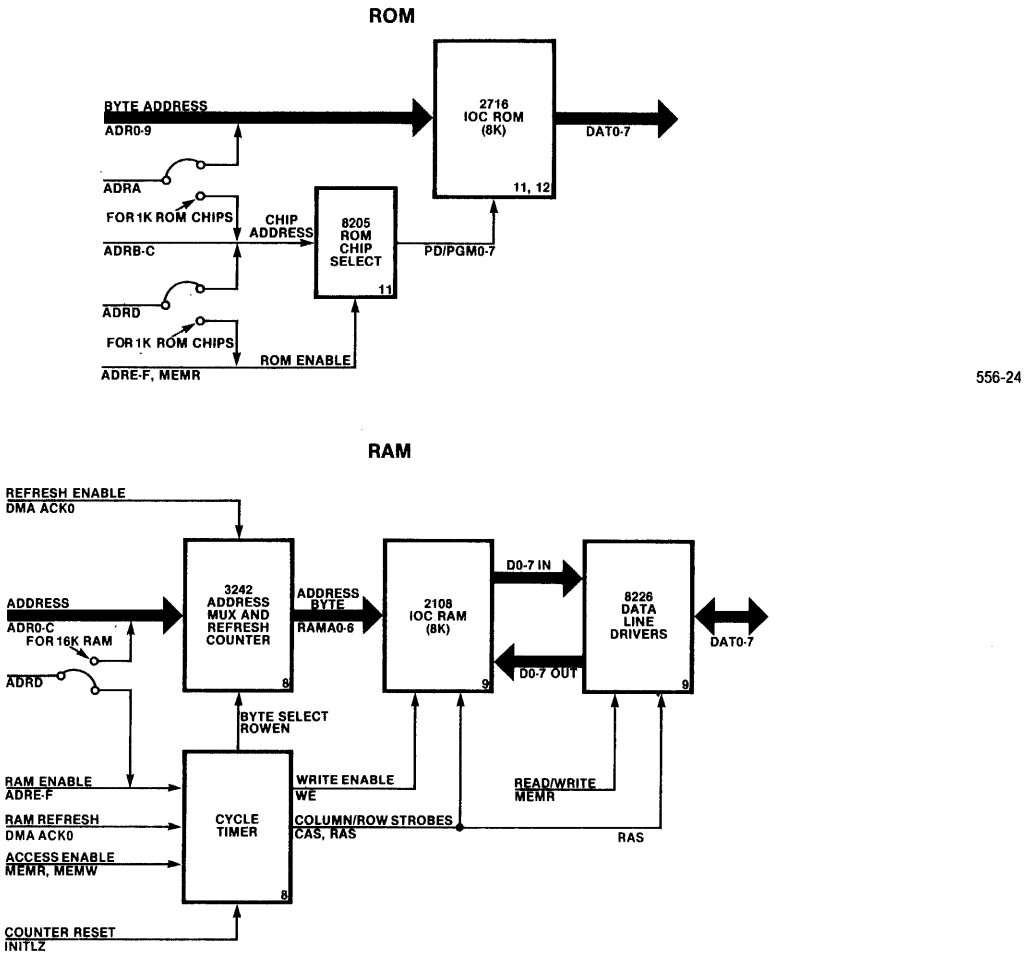
channel 2 and the parameters of the second block are output to channel 3. The DMA controller is then prepared to "paint" the entire CRT screen.

All DMA data transfers to the CRT controller are made via DMA channel 2. DMA channel 3 neither receives DMA requests nor issues DMA acknowledges or hold requests. The signaling sequence of channel 2 is identical to that of channel 1 as previously explained for diskette data transfers. However, whereas diskette transfers are always in response to IOC processor commands, the CRT transfers are cyclic and independent of the timing of keystrokes or programmed outputs to the CRT. Text intended for display is added to the CRT table asynchronously.

4.2 IOC MEMORY SUBSYSTEM

The IOC memory subsystem (figure 4-3) is relatively simple due to the use of non-overlapping addresses for ROM and RAM (shadow ROM is not used). the RAM circuits consist of eight 2108/2109 RAM chips that provide 8k bytes of data storage, two 8226s that provide data line buffering, a 3242 address multiplexer and refresh counter, and a TTL cycle timer that generates RAM control signals. The ROM circuits consist of four preprogrammed 2k byte ROM chips and an 8205 binary decoder that selects the ROM chips based on the MSBs of the applied address.

Both RAM and ROM can each be expanded to 16k bytes by in-plant modification of the IOC board. RAM expansion is accomplished by replacement of the 2108/2109 chips with 2116/2117 chips and



NOTE: The number in the lower right-hand corner of each block identifies the sheet number of the IOC schematic.

Figure 4-3. IOC Memory Subsystem Block Diagram 556-25

concurrent alteration of jumpers that change the function of address bit ADRD from that of RAM enabling to that of byte selection. The ROM circuits are normally jumpered for 2k 2316/2716 ROM chips with expansion accomplished by insertion of four additional 2k ROM chips in currently unused sockets. The arrangement of the ROM circuits is such that 1k ROM chips can be used (ADRA is rejumpered to provide chip selection rather than byte selection). ADRD is also rejumpered to provide ROM decode enabling rather than chip selection. With this arrangement, ADRF is no longer used for ROM decode enabling since all ROM and RAM use the low-order 32k bytes of memory space.

4.2.1 IOC ROM

All of the ROM circuits normally used within the IOC are shown on sheet 11. The additional ROM circuits shown on sheet 12 of the schematic are used only when the four additional ROM chips are installed. With the normal set of four 2k ROM chips, the addresses assigned to ROM are 0000 through 1FFF. In this case, the five MSBs of the address (ADRB through ADRF) are decoded upon the occurrence of MEMR/ to select a ROM chip, and the LSBs (ADR0-ADRA) are applied directly to the ROM chips to address the specified byte.

4.2.2 IOC RAM

Circuits of the IOC RAM have been designed for simplicity by employing the read cycle refresh mode in which the timing of the refresh cycle is identical to that of a normal read cycle. Refresh cycles occur once every 15.5 microseconds as established by the I/O subsystem timer. During each cycle, the 128 RAM locations of one row are refreshed. The 64 cycles required to refresh the entire 8k RAM are completed in less than one millisecond.

Refresh cycles are initiated when the I/O subsystem timer generates DMA REQ 0. This signal, being the highest priority DMA request, causes the DMA controller to initiate a DMA cycle as soon as the current processor instruction cycle is completed. When the DMA controller generates DMA ACK 0/, the IOC RAM begins a refresh cycle. During the refresh cycle, the IOC bus is in an idle state because the DMA controller inhibits both the IOC processor (HOLD) and the I/O ports (DMA ON). Between refresh cycles, the IOC RAM responds to the memory control bus commands (MEMR/ or MEMW/) only if the address is in the range 4000 through 5FFF (ADRE true, and ADRD and ADRF false).

The memory timing and control logic (sheet 8) operates identically during read cycles (MEMR/), write cycles (MEMW/), and refresh cycles (DMA ACK 0/). The timing logic contains a shift register, the outputs of which (1Q-6Q) are all false when the RAM is idle. The control logic contains a start cycle flip-flop that starts the timer when a request for a RAM cycle is received. An end of cycle flip-flop is used to maintain the RAM control lines (RAS/ and CAS/) while the shift register is recycling to its initial state. The end of cycle flip-flop also resets the start cycle flip-flop and, itself, is reset at the end of the command cycle. The shift register, the start cycle flip-flop, and the end of cycle flip-flop use the 22.032 MHz output from the IOC processor subsystem clock pulse generator. RAM chip control signals produced by the memory timing and control logic include the row address strobe (RAS/), the column address strobe (CAS/), and write enable (WE/). The ROWEN signal is used by the 3242 to multiplex either the row or column address to the RAM. The timing of these outputs is shown on figure 4-4.

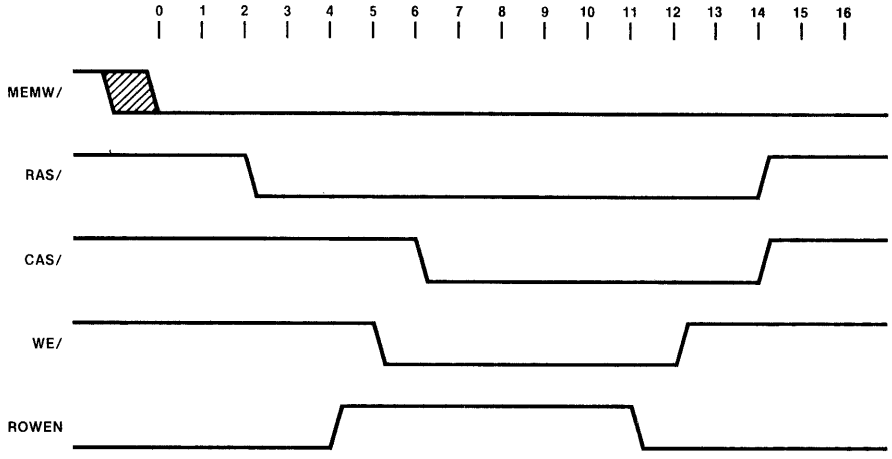


Figure 4-4. IOC RAM Timing 556-26

The 3242 address multiplexer and refresh counter (sheet 8) operates in either the read/write mode or the refresh mode depending on the state of DMA ACK 0/. When DMA ACK 0/ is false (high) the states of the IOC bus address lines are applied to the RAM chips (sheet 9), but are ignored until the memory control logic receives MEMR/ or MEMW/ and subsequently generates RAS/. Since at this time ROWEN is inactive, the high-order address bits (ADR7-ADRD) are multiplexed to select the specified row within RAM. As the cycle continues, ROWEN goes active to multiplex the low-order address bits (ADR0- ADR6) from the address multiplexer. The low-order (column) address is subsequently enabled at the RAM by CAS/, and the RAM location specified by the row and column addresses is accessed.

NOTE

All RAM locations within a row are refreshed when the row is specified during normal read cycles. The refresh cycles merely ensure that all rows are selected during each 2-millisecond period as required by the dynamic RAMs.

When DMA ACK 0/ goes active (low), the 3242 chip uses the value of an internal counter to generate a row address. The memory cycle is started in the normal manner to cause the generation of RAS/ and CAS/. However, only the row address strobe (RAS/) has significance since all locations of the row are refreshed. The refresh counter of the 3242 is a modulo-128 binary counter that is incremented each time DMA ACK 0/ goes inactive.

4.3 IOC INPUT-OUTPUT SUBSYSTEM

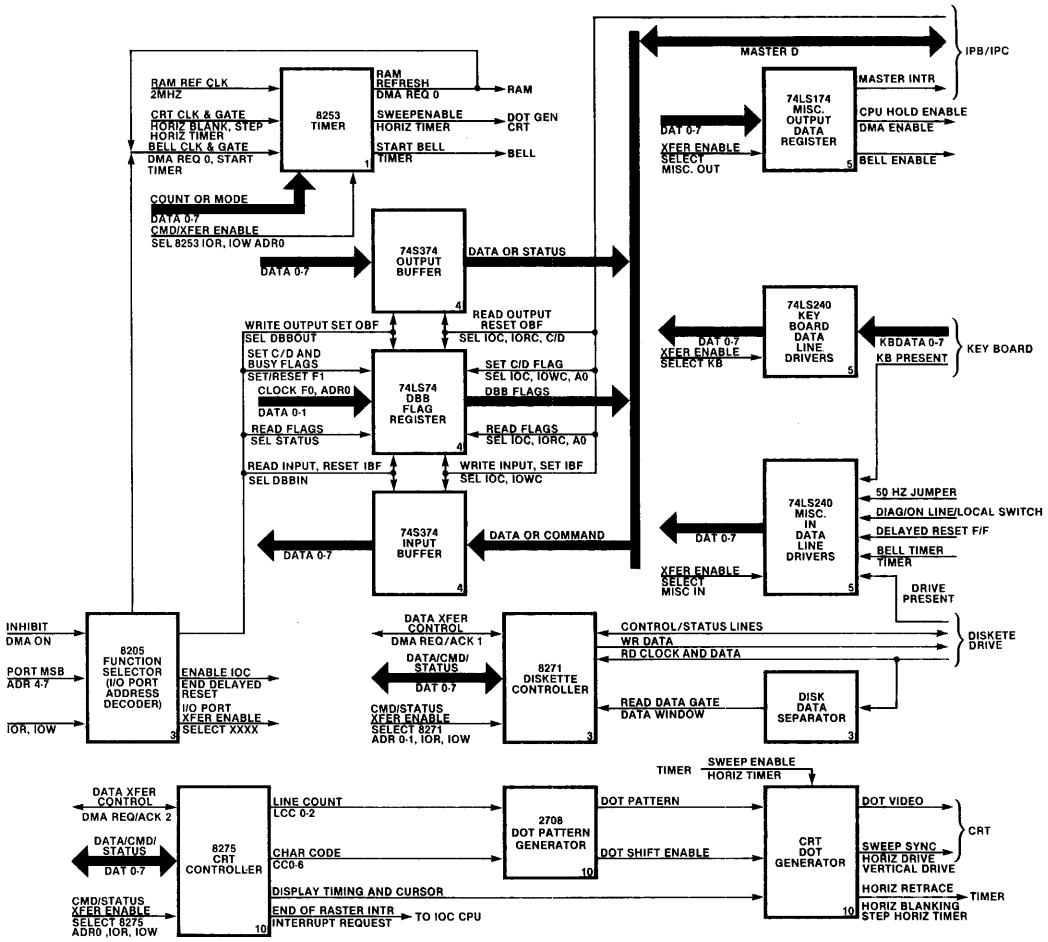
The IOC input-output subsystem (figure 4-5) establishes communications paths between the IOC processor subsystem and circuits external to the IOC board. There are four primary paths. One path is to the IPB/IPC and consists of TTL registers that comprise the bidirectional data byte buffer (DBB). The second path consists of TTL line receivers for inputs from the 8041 keyboard processor. The third path is a bidirectional path to the integral diskette. This path consists of an 8271 programmable diskette controller plus a number of TTL elements. The TTL elements monitor the serial clock and data input from the diskette drive to establish a "data window" for the diskette controller. The fourth path is to the CRT. It consists of an 8275 programmable CRT controller plus a combination of LSI and TTL elements. These elements drive the CRT sweeps to synchronize the raster and generate intensified dots on the horizontal sweeps to produce the display characters.

In addition to the preceding, the input-output subsystem contains support circuits that are not dedicated to a single communications path. These support circuits include the port decode logic (a pair of 8205 binary decoders) that produces chip select signals from the four MSBs of the 8-bit I/O port addresses (the LSBs are decoded, if necessary, at the selected chips). Another support circuit is a three-section 8253 timer that is used for RAM refresh and CRT sweep timing and for keying a console alarm that produces an audible tone under program control. The final circuits of the subsystem consist of a "miscellaneous" output register and miscellaneous input line drivers. These circuits allow IOC software sensing and control of hardware events.

4.3.1 IOC PORT DECODE LOGIC

The IOC port decode logic (sheet 3) decodes the four MSBs of the I/O port address to control the IOC hardware. The port decode outputs and the associated port addresses are listed in table 4-1. In some cases (ports 1x-5x), the port addresses are used to set or reset flip- flops or to start the alarm timer. In these cases, the data lines of the IOC bus are not used, and the LSBs of the port address are used only to differentiate between setting and resetting of the flip-flop (port 2x only). In all other cases, the port address is used to enable reading or writing from a register. With ports 0x, 6x, 8x, 9x, Ax, and Bx, the register is a discrete element of the IOC. With ports Cx through Fx, the register may be one of several within a programmable chip. Port address LSBs are typically used by the programmable chips.

The IOC port decode logic consists of two 8205 binary decoders, both of which are disabled during any DMA transfer (DMA ON). One of the decoders (A60) is enabled only during I/O write operations (IOW/), whereas the other decoder (A59) is enabled during both I/O read (IOR/) and write (IOW/) operations. This arrangement inhibits the use of read commands on I/O ports that are exclusively reserved for data writing or control of hardware functions. Enabling of the binary decoders is controlled by the I/O port address MSB (ADR7). This bit enables the read/write decoder chip when true and the write only decoder chip when false. Thus the low-order I/O port addresses (0x-7x) are reserved for writing and control, whereas the high-order I/O port addresses (8x-Fx) may be used for reading, writing, or control.



NOTE: The number in the lower right-hand corner of each block identifies the sheet number of the IOC schematic.

Figure 4-5. IOC I/O Subsystem Block Diagram 556-27

Table 4-1. Port Address Functions

I/O Port Address	Mnemonic	Function
0x	SELECT DBBOUT/	Write to DBB output buffer, set/reset OBF flag
1x		Not used
2x	CLOCK F0/	Set/reset IOC busy flag
3x	SET F1/	Set Command/Data flag (status to IPB/IPC)
4x	RESET F1/	Reset Command/Data flag (data to IPB/IPC)
5x	START TIMER/	Enable 100 ms timer (bell)
6x	SELECT MISCOUT/	Write to miscellaneous output register
7x		Not Used
8x	SELECT MISCIN/	Read miscellaneous data
9x	SELECT KB/	Read keyboard data
Ax	SELECT STATUS/	Read DBB status (flags)
Bx	SELECT DBBIN/	Read DBB input buffer, reset IBF flag
Cx	SELECT 8271/	Read/write diskette controller
Dx	SELECT 8275/	Read/write CRT controller
Ex	SELECT 8253/	Read/write interval timer
Fx	SELECT 8257/	Read/write DMA controller

4.3.2 MISCELLANEOUS INPUT/OUTPUT CIRCUITS

The miscellaneous input/output circuits (sheet 5) provide for communications between the IOC processor and a variety IOC hardware elements. The miscellaneous output data register is merely a grouping of latching flip-flops that control DMA enabling, bell enabling, and the generation of an interrupt to the IPB/IPC processor. DMA ENABLE is used during startup to disable the DMA controller prior to its initialization. The bell enable signal is used only to enable the audible signal; the bell is energized by the TIMER/ output of the 8253 timer. This arrangement allows use of the bell timer for other purposes in that TIMER/ is also a miscellaneous input.

The miscellaneous input data line drivers are used to return the state of TIMER/ to the IOC processor to allow its use for purposes other than energizing the console bell. One such usage is to provide a timeout for diagnostic programs.

The remaining miscellaneous inputs monitor the status of external devices (KB PRESENT/ and FLOPPY PRESENT/), determine the position of the rear panel DIAG/ON LINE/LOCAL switch, and detect the presence of the 50 Hz jumper (50 HZ JMPR/). All of the miscellaneous inputs, other than TIMER/, are used to redirect IOC program operations.

4.3.3 IOC INTERVAL TIMER

The 8253 interval timer (sheet 1) is used to establish the frequency of RAM refresh cycles and the duration of the console bell tone. The RAM refresh segment of the timer (counter 0) is incremented by the 2 MHz clock and is initialized with a count of 31. This causes the RAM refresh signal to occur once every 15.5 microseconds. The output (DMA REQ 0) is maintained until the DMA controller acknolwedges that a refresh cycle has started (DMA ACK 0/). Counter 0 requires no program intervention after system initialization.

The second segment of the timer (counter 1) is used by the CRT control circuits to establish the width of the HORIZ DRIVE signal to the CRT. This horizontal retrace timing is dependent on the value entered in counter 1 during initialization. When 60 Hz operation is specified (50 Hz jumper omitted), the counter is initialized with a value of 44 to provide a horizontal drive width of approximately 24 microseconds (1.836 MHz STEP HORIZ TIMER clock frequency divided by 44). When 50 Hz operation is specified (50 Hz jumper installed), the counter is initialized with a value of 48 to provide a horizontal drive width of approximately 26.2 microseconds. The reduced drive width for 60 Hz operation partially compensates for the higher vertical sweep frequency of the CRT (the vertical sweep is set to the line frequency). Counter 1 operation is enabled by HORIZ BLANKING (the latched horizontal retrace signal from the CRT controller).

The third counter, counter 2, is used to energize the console bell when the bell is enabled by IOC software. Under normal circumstances, the bell is sounded (i.e., the timer is started) when a character is entered in the 70th character position of a line. The timer is started by START TIMER/ which enables the counter 0 output to the clock input of counter 2. The period of this clock (15.5 microseconds) and the counter value entered during timer initialization provide a bell tone of approximately 90 milliseconds.

4.3.4 DATA BYTE BUFFER

The data byte buffer (DBB) consists of two 8-bit registers (the input buffer and the output buffer), four flag flip-flops (that indicate the status of transfers between the IOC and the IPB/IPC), and two four-bit gated line drivers (that enable coupling of the flag states to either the IOC or IPB/IPC data lines). The DBB completely isolates the data lines of the IOC from the data lines of the IPB/IPC, while allowing either processor to access the flags and thereby determine when a transfer is pending, in process, or completed.

Both processors examine the flags prior to using the DBB. The IOC processor examines the flags to detect the presence of a new command and then, if necessary, polls the flags to detect the presence of data in the input buffer. Conversely, the IPB/IPC examines the flags (if the preceding command required a data or status return) and awaits an indication that the command has been completely processed.

Referring to sheet 4, two of the flags, the output buffer full (OBF) flag and the input buffer full (IBF) flag, directly indicate the presence of valid data in the DBB. The IBF flag is set by the IPB's or IPC's master processor when a command or data byte is clocked into the input buffer (SEL IOC/, IOWC/). This flag is reset by the IOC when the byte is accessed via port address Bx (SEL DBBIN/).

The OBF flag is set by the IOC when a status or data byte is clocked to the output buffer via port address 00 (SEL DBBOUT/). The OBF flag is reset by the IPB/IPC when the byte is accessed (SEL IOC/, IORC/, and A0 all low). When status (i.e., the flag states) is read by the IPB/IPC, control signals SEL IOC/ and IORC/ are low and A0 is high. During a status read, the flags are gated to the IPB/IPC data

lines and both the IBF and the OBF flags are not affected. Flag reading by the IOC is accomplished when SELECT STATUS/ is low (port address Ax).

The third flag, the command/data flag (F1), is set or reset by either processor. For inputs from the IPB/IPC (SEL IOC/, IOWC), this flag is controlled by the state of A0. For outputs, the IOC processor software uses port addresses 3x and 4x to generate SET F1/ and RESET F1/, respectively, to set or reset the C/D flag flip-flop. Note that while the function of A0 from the IPB/IPC remains the same for input or output (that of identifying a byte as either data or command/status), its function as an input is to set the flag and its function as an output is to select the byte source. The byte containing flag status represents the status of the interface between the IPB/IPC and the IOC; all other IOC status is returned within data bytes.

The fourth flag associated with the DBB is the IOC busy flag (F0). This flag is set by the IOC when a command is accepted and is reset by the IOC when the command is fully processed. Port addresses 00 and 01 set and reset the flag, respectively. The IPB/IPC has no control over the busy flag; a command output to the IOC is prohibited when this flag is set.

4.3.5 KEYBOARD INPUT CIRCUITS

The keyboard input circuits (sheet 5) consist of the gated line drivers that control data entry from the 8041 keyboard processor. The enabling of the line drivers occurs when the IOC processor generates I/O port address 90 or 92. The MSBs of the port address are decoded to generate SELECT KB/. This signal enables the line drivers and selects the keyboard processor's DBB for a data read operation (READ RB/). One of the LSBs (ADR1) is sent directly to the keyboard processor where it selects either a character byte (when false) or a status byte (when true). The status returned is identical to the flags discussed in paragraph 4.3.4. However, the DBB is internal to the 8041 processor chip. Further information on the keyboard processor is provided in Chapter 6.

4.3.6 CRT CONTROL CIRCUITS

The CRT control circuits (sheet 10) consist of an 8275 CRT controller chip, a 2708/2308 ROM that is used to provide dot patterns (character font) for displayed characters, and a variety of TTL elements that produce the final CRT video and establish overall timing of the CRT sweeps (and related internal functions of the IOC).

The CRT employed within the Intellec Series II development system chassis contains the analog circuits necessary for the generation of the electron beam and for the sweeping of electron beam in both the horizontal and vertical axes. The sweep circuits are designed for generation of a raster wherein 260 or more horizontal sweeps occur for each vertical sweep. Video displays are produced by external control of beam intensity and by synchronization of the sweep circuits to position the video on the screen. All sweep synchronization, beam intensification, and blanking required to produce the desired display are done external to the CRT module.

The standard method for displaying characters on a CRT raster is to produce dots that intensify those segments of the characters that coincide with specific horizontal sweeps. To display a single horizontal row of characters, several lines of intensified dots are produced. Each character is displayed within a 7 by 8 dot matrix, and a single character is normally 5 dots wide and 7 dots high (see figure 4-6). Additional horizontal sweeps (two for 60 Hz and three for 50 Hz operation) are used to provide vertical separation between the rows of characters and to provide for the display of the cursor underline. The overall display is 80 characters in width and 25 characters high.

Characters to be displayed are input via DMA when the CRT controller generates DMA REQ 2/. The sequence is such that the 80 characters that comprise a row are input and processed as an entity. During the processing of any row, DMA input continues since the CRT controller has storage provision for two complete rows of characters. A row always consists of 80 characters; space characters at the beginning, within or at the end of any row are displayed as blanks. In processing a row, the CRT controller generates line counts (LC0-LC2) and character codes (CC0-CC6). The character codes for each character within a row are output, in sequence, for each of the eight lines of the row.

The 2308/2708 character font ROM uses the line count and character code as an address to access a specific dot pattern. The entire process is synchronized to the appropriate horizontal sweeps. The characters that may be produced include all of the 128 characters of the ASCII set. The number of unique line count/character code combinations is thus 1024 (128 characters times 8 lines).

The output of the character font ROM for each line of the character includes six bits for the dots (the character field) plus two bits that control the insertion of a small (half dot) delay in the display of either three-bit half of the dot pattern as shown in figure 4-6. This selective "displacement" of the dots permits improved character readability.

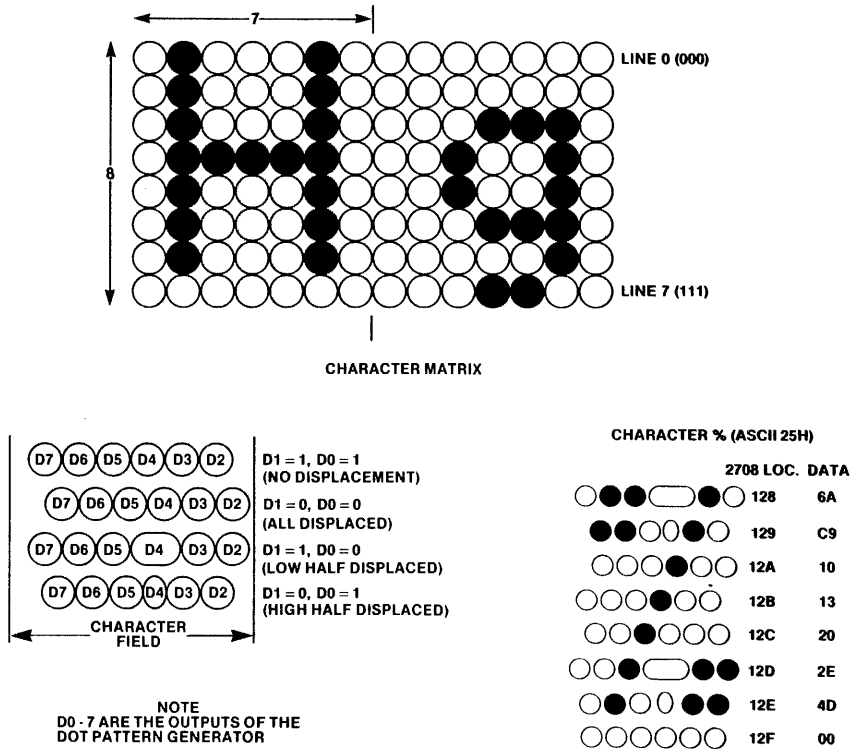


Figure 4-6. Character Display Matrix 556-28

The 6-bit dot pattern and the two displacement bits are output by the character font ROM as a parallel byte. The CRT dot generator serializes the string of dot patterns and inserts the delays specified by the displacement bits. The resulting serial stream of dots is synchronized to the horizontal sweeps and is blanked during retrace time and during the sweeps that separate the rows of characters. Dot pattern timing is shown in figure 4-7.

The first sweep following any row of characters (the ninth line) is used for display of the underline cursor. Generation of the underline is accomplished by a direct connection between the CRT controller and the video driver. The cursor is displayed by momentarily strobing the video driver so that all other signals controlling CRT beam blanking are overridden.

The serial steam of dot patterns is applied to two flip-flops. One of these flip-flops uses the 11.016 MHz clock pulse and supplies normal (not displaced) dots. The second flip-flop uses the complement of 11.016 MHZ clock and supplies displaced dots. These two flip-flops are used to produce all dot patterns; their outputs provide the desired combination of normal and displaced dots to the 74LS151 multiplexer (A5).

The multiplexer control inputs (A, B, and C) select either normal or displaced dots in either normal and reverse video. Input A is controlled by the second LSB of the character font ROM during the first four dots of each character line and by the LSB during the last two dots. With normal (non-reversed) video, normal dots are accepted at multiplexer input D2 when A is false. When A is true, displaced dots are accepted at input D3. When reversed video is specified (B false), complements of the dot patterns are accepted at multiplexer inputs D0 and D1. When control input C is true, (suppress video) the accepted inputs are either D4-D5 (reverse) or D6-D7 (normal), depending on the state of input B. When input C is true, no dots are produced for normal video and all dots are produced for reversed video; no characters are displayed when C is true.

The vertical sweep of the CRT is synchronized to the 50 Hz or 60 Hz line frequency to prevent display beating against ambient lighting. However, the display time of each character, the number of characters per row, and the number of character rows remain unchanged regardless of the vertical sweep rate. Compensation for 50 Hz operation is provided by increasing the number of horizontal sweeps and by extending the horizontal and vertical

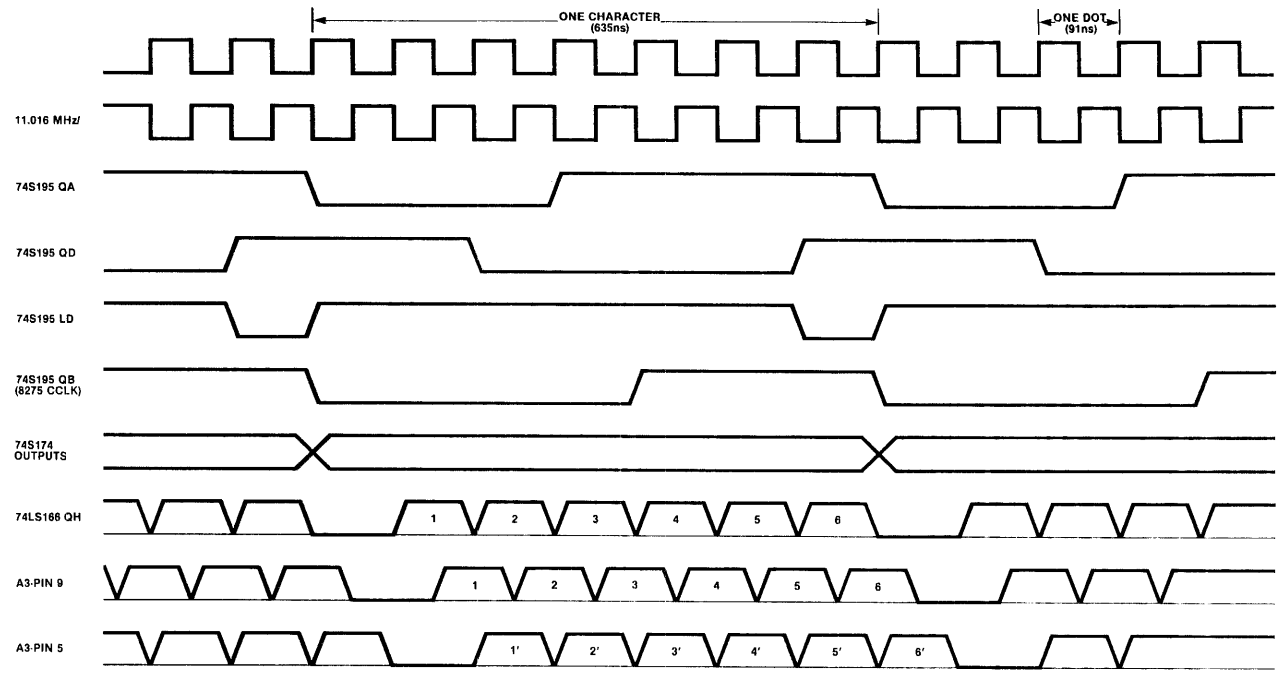


Figure 4-7. Dot Pattern Timing

556-29

retrace intervals. These latter variables are established by values sent to the CRT controller when it is initialized. Associated hardware timing circuits control sweep triggering during the retrace intervals. These hardware timing circuits include the 8253 interval timer and the 74S195 shift register of the CRT dot generator. Both the programmed and the hardware-controlled variables are determined by the presence or absence of the 50 HZ JUMPER (W8). Sweep timing is shown in figure 4-8.

During power-on or reset, the IOC software reads the miscellaneous input byte containing the 50 Hz JUMPER bit. If the jumper is removed (60 Hz operation), the parameters sent by the IOC software to the CRT controller include a vertical retrace row count of 2, a lines-per-row value of 10, and a horizontal retrace count of 20. If the jumper is installed (50 Hz operation), the vertical retrace row count is 3, the lines-per-row value is 11 and the horizontal retrace count is 26. These two combinations of initialization parameters provide the compensation required for 50 and 60 Hz operation.

The horizontal and vertical retrace counts are used by the CRT controller to establish the duration of HRTC (horizontal retrace) and VRTC (vertical retrace) signal intervals. The video suppress signal (VSP) is generated in coincidence with either signal to provide the required blanking during retrace. The two retrace signals are used to enable the hardware timing circuits for the CRT. The hardware timing circuits, in turn, generate HORIZ DRIVE and VERTICAL DRIVE/ to trigger the sweep circuits of the CRT.

The 74S195 shift register of the CRT dot generator is used during both sweep and retrace intervals. The register is connected to operate as a shift counter by virtue of feedback from the MSB (QD) to the serial input (J,K) pins. If there were no other logic associated with the shift register, it would continually produce four zeros followed by four ones (modulo-8). However, the occurrence of all ones (QA and QD true) causes a parallel load. During any horizontal sweep, the shift register is loaded with a binary value of 1100 (D and C inputs locially high, B and A inputs logically low), and the register operates as a modulo-7 counter. In addition to parallel loading the shift register, the load output (A6 pin 3) also loads the 6-bit dot pattern from the character generator ROM (A19) into the serializer (A4) and loads the displacement bits and video control signals into the control latch (A9). The dot pattern is subsequently shifted out of the serializer by the next five clock pulses, and the entire cycle is repeated for the dot pattern of the next character.

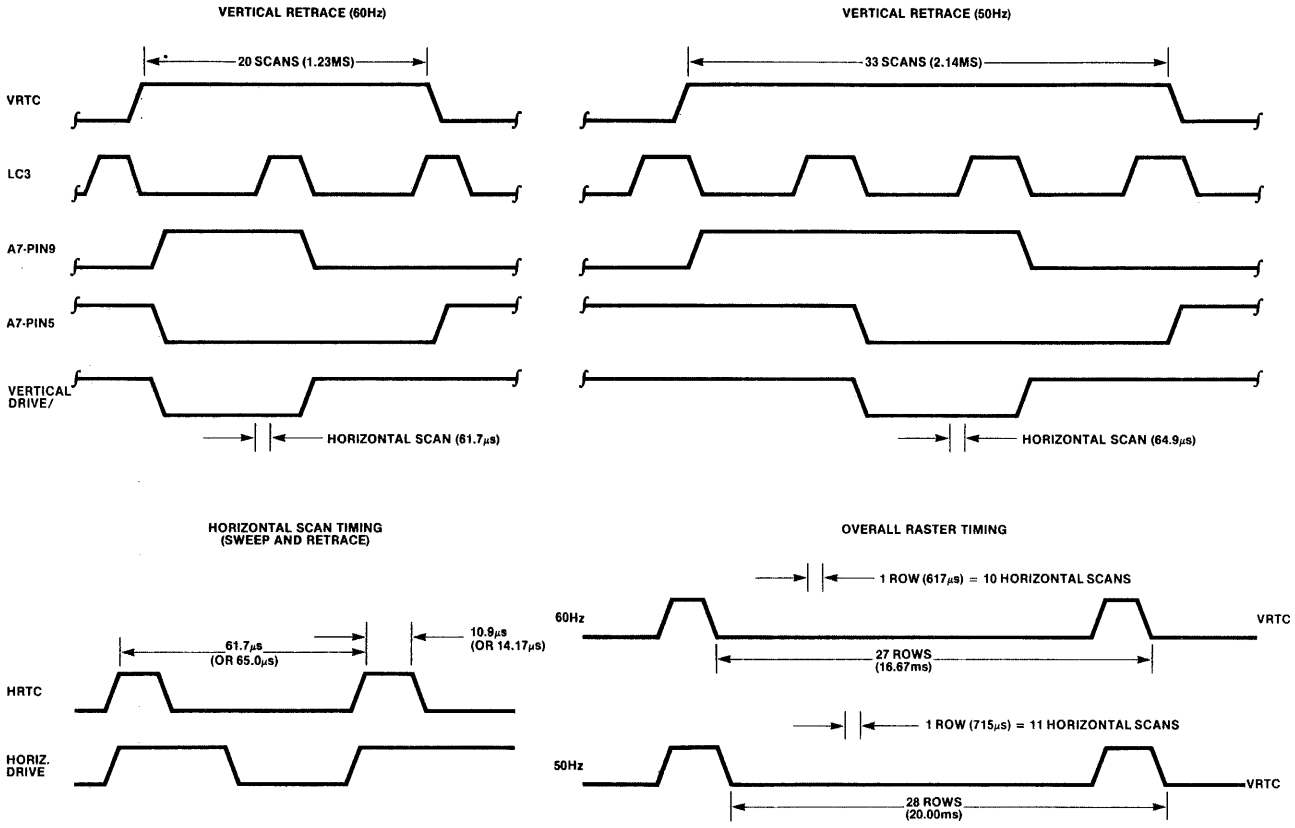


Figure 4-8. CRT Sweep Timing 556-30

The QB output from the shift register is used to selectively gate the displacement bits from the control latch to the dot pattern multiplexer. While the first four dot bits are being clocked to the multiplexer, the QB output is logically low to enable the second LSB of the character font ROM to the multiplexer's shift control (A) input. While the last two dot bits are being clocked, the QB output is logically high to enable the LSB of the character font ROM to the A input. The QB output is also used as the clock input to the CRT controller (A20). The negative transition of QB advances the controller's character counter to cause the next character code to be output to the serializer.

During horizontal retrace, the CRT controller activates HRTC. This signal, in addition to activating the HORIZ DRIVE output to the CRT to start the retrace interval, modifies the operation of the shift register and enables counter 1 (the horizontal retrace timer) of the 8253 interval timer on sheet 1 (HORIZ BLANKING active). The logic low level at the C input to the shift register causes the register to be parallel loaded with a binary value of 1000, and the register operates as a modulo-6 counter. Accordingly, the QB output occurs at a 1.836 MHz rate which, at the clock input to the interval timer (STEP HORIZ TIMER), starts the horizontal retrace timeout. The active HORIZ TIMER/ output lengthens the duration of the HORIZ DRIVE signal to the CRT to the required pulse width. Further details on the 8253 timer may be found in paragraph 4.3.3.

During vertical retrace, the CRT controller activates VRTC. This signal is used to enable the two vertical drive flip-flops (A7). Referring to sheet 10 of the schematic, the VERTICAL DRIVE/ output to the CRT is active when the first flip-flop is set and the second flip-flop is reset. (While VRTC is inactive, the first flip-flop is held reset and the second flip-flop is held set.) Both flip-flops are clocked on the trailing-edge of the high-order line count bit (LC3) from the CRT controller. This bit goes active just before the ninth horizontal scan line (line 8) of any row and returns to an inactive level (clocking the flip-flops) just prior to the conclusion of the tenth scan line (60 Hz operation) or eleventh scan line (50 Hz operation) of the row.

When operating at 60 Hz, the first negative transition of LC3 occurs at the beginning of the vertical retrace interval (at the beginning of the 26th row). This transition clocks the first flip-flop set and, since the 50 Hz jumper is not installed, clocks the second flip-flop reset to generate VERTICAL DRIVE/. Since the VRTC output from the CRT controller remains active for two row counts (the CRT controller is intialized with a vertical retrace count of 2 for 60 Hz operation), vertical retrace occurs during the 26th row, and the 27th row is positioned at the very top of the screen. When operating at 50 Hz, the negative transition of LC3 at the beginning of the 26th row clocks the first flip-flop set, but since the 50 Hz jumper is installed, also clocks the second flip-flop set. Consequently, the VERTICAL DRIVE/ output remains inactive, and the 26th row is positioned at the bottom of the screen. At the beginning of the 27th row, LC3 is again generated. Since the first flip-flop is set, the second flip-flop is clocked reset (generating VERTICAL DRIVE/), and vertical retrace occurs during the 27th row. Since the VRTC output remains active for three row counts (the CRT controller is initialized with a vertical retrace count of 3 for 50 Hz operation), the 28th row of the frame is positioned at the very top of the screen.

During 50 Hz operation, in order to scan an entire frame (28 rows) in precisely 20 milliseconds, an additional compensation circuit is enabled to shorten the horizontal retrace intervals during the 27th row of the frame. Referring again to sheet 10, the NAND gate (A40) at the D preset input to the shift register is active during the horizontal retrace intervals of lines 0 through 7 (LC3 inactive) of the 27th row. Accordingly, the shift register operates as a modulo-5 counter and the 27th row is scanned in 696 nanoseconds instead of the expected 715 nanoseconds.

4.3.7 DISKETTE CONTROL CIRCUITS

The integral diskette control circuits (sheet 3) consist of an 8271 diskette controller plus the TTL elements that comprise the diskette data separator. The diskette controller receives commands via I/O ports of the IOC processor, transfers read/write data via DMA channel 1, and establishes the interface with the integral diskette drive of the Intellec Series II microcomputer development system chassis. The data separator augments the diskette drive interface by converting the double frequency flux pattern (clock and data) from the drive into a series of data windows that are used by the diskette controller to extract binary data from the combined clock/data pattern.

The diskette controller remains in an idle state until it receives a command via an I/O port of the IOC processor. Three I/O port addresses are used: port C0 (for commands and status), port C1 (for parameters and results), and port C2 (to reset the diskette controller).

The MSBs of port address (ADR4-ADR7) are decoded externally by the function selector (A59) to select the 8271. The output from the function selector, combined with IOW/, causes the diskette controller to accept commands or parameters appearing on the data lines (DAT0-DAT7). The LSB of the port address (ADR0) is used to distinguish between commands and parameters. When the 8271 is selected and IOR/ is active, the IOC processor is requesting either a status or result byte, depending on the state of ADR0. The second LSB of the port address (ADR1) is only active when the IOC processor uses port address C2 to reset the diskette controller.

The command byte defines the operation to be performed whereas succeeding parameter bytes supply information required to perform the operation (for a diskette read/write operation, the parameter bytes provide the track and sector addresses).

When a command requires repositioning of the read/write head and/or the transfer of data to or from the diskette, the diskette controller enters the execution phase on receipt of the last parameter byte. When the controller enters the execution phase, it first checks to be sure that the drive is ready and then, if necessary, steps the head to the addressed track and loads the drive head. If a read, write or verify operation is specified, the diskette controller waits for the occurrence of an index pulse (to locate the logical beginning of the track) and then begins reading the individual sector ID fields to locate the addressed sector.

If a read operation is specified, the diskette controller remains in the read mode. After the addressed sector's data address mark is read, the controller begins assembling the serial data bits read from the diskette into 8-bit bytes. Each time a byte is assembled, the diskette controller generates a DMA request (DMA REQ 1) to the DMA controller. The DMA controller, in turn, generates DMA ACK 1/ to the diskette controller and then writes the assembled byte to the addressed location in IOC RAM. When the complete sector (128 bytes) has been transferred, the IPB/IPC processor reads the diskette controller's result register (to determine if the operation was successfully completed) and then reads in the data block from IOC RAM via the DBB.

If a write operation is specified, the diskette controller switches from the read mode to the write mode during the post ID gap and generates a DMA request to the DMA controller to access the first data byte to

be written from IOC RAM. (The IPB/IPC processor must first write the data block to IOC RAM before initiating the diskette write operation.) The diskette controller rewrites the sync bytes and the data address mark on the diskette and, immediately following the data address mark, begins writing the accessed data byte as a serial stream of clock and data pulses. While the first byte is being written, the next byte is accessed from RAM. When the complete data block (1024 bits) has been written, the controller appends its 2-byte CRC character to the end of the data block.

The double frequency (FM) recording technique employed by the diskette is based on a 4μs bit cell. Each bit cell begins with a 250 ns clock pulse (clock pulses provide synchronization during diskette read operations). When a binary one is to be written, a second 250 ns pulse is generated approximately 2 μs after the start of the bit cell. (If a binary zero is to be recorded, the second pulse is omitted.) The 250 ns pulses, the 4 μs bit rate, and the 32 μs DMA transfer rate are all derived from the 4 MHz clock pulse used by the diskette controller.

During a diskette read operation, the diskette controller samples the level of its DATA WINDOW input at the beginning of each 4 μs bit cell to determine if the previous bit cell contained a data pulse (the DATA WINDOW input will be active if the previous bit cell contained a data "1" bit). Referring to sheet 3, the data separation logic consists of a pulse detector (74S175 flip-flop A2) and a window counter (74S163 binary counters A17 and A29). The pulse detector generates a delayed 45 ns pulse whenever a clock or data pulse is present in the incoming pulse stream from the diskette (UNSEPD DATA/). This pulse presets the window counter with a binary value of 1000001. As a result, the QC output from the second binary counter (A29) will be logically high, which will enable the first binary counter and activate the DATA WINDOW input to the diskette controller. The window counter is advanced by the 22.032 MHz clock signal. Consequently, if 63 clock pulses (2.86 μs) occur before the next incoming pulse is detected, the QC output will be clocked to a logic low level. Assuming that the window counter is loaded by a clock pulse (the zero bits in the sector gaps synchronize the data separation logic to the clock pulses), if the previous bit cell contained a zero bit (absence of a data pulse), the window counter would overflow, and the DATA WINDOW input would be inactive when the next (clock) pulse was received. Conversely, if the previous bit cell contained a one bit (the presence of a data pulse), the window counter would be reloaded, and the DATA WINDOW input would be active when the next clock pulse was received.

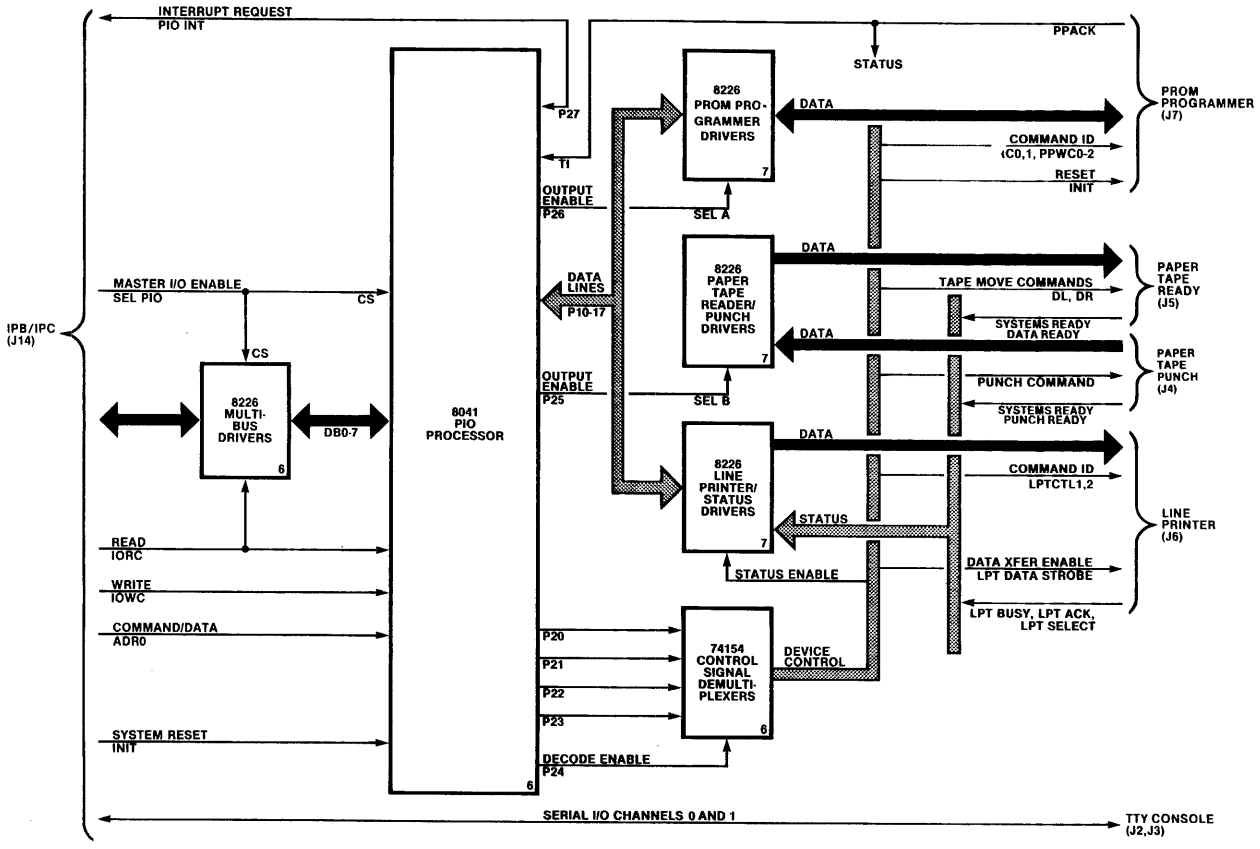
CHAPTER 5
PARALLEL INPUT-OUTPUT SUBSYSTEM

The parallel input-output (PIO) subsystem consists of all hardware elements required to establish IPB/IPC communications with the four standard Intellec Series II development system peripherals. These standard peripherals are a paper tape reader, a paper tape punch, a line printer, and an Intel Universal PROM Programmer.

The PIO subsystem is incorporated on the IOC board; the IOC board provides connectors for all external devices that are normally interfaced with the Intellec Series II development system. The IOC board also contains the connectors for the serial I/O channels described in Chapters 2 and 3.

Circuit elements of the PIO (figure 5-1) include an 8041 microprocessor (the PIO processor), eight 8226 bidirectional line drivers, and a single TTL chip (a 74154 demultiplexer). The PIO processor provides the "intelligence" necessary for the control and monitoring of the four external devices; other elements either provide isolation of the processor or demultiplex its outputs. The role of the PIO processor is to accept general directives from the IPB/IPC master processor and convert these directives into the detailed signal sequences required by the external devices.

In the remainder of this chapter, the sheet numbers called out in both the text and on the illustrations are references to individual sheets of the IOC schematic diagram included in the *Intellec Series II Schematic Drawings*.



NOTE: The number in the lower right-hand corner of each block identifies the sheet number of the IOC schematic.

Figure 5-1. PIO Block Diagram

556-31

5.1 PIO PROCESSOR

The 8041 microprocessor (sheet 6) is one of a family of microprocessors that provides true general-purpose processing while requiring a minimum of external circuits. The 8041 has an internal timer, 1k bytes of ROM and 64 bytes of RAM. The 8041 is often referred to as a universal peripheral interface (UPI) due to its ability to efficiently interface peripheral devices while serving as a slave to a master 8080/8085 microprocessor. Use of the 8041 in the PIO is typical of its application.

Since the 8041 employs internal memory, no pins are reserved for addressing. Instead, there are three byte-parallel I/O channels. The first of these channels (D0-D7) is very similar to the bidirectional bus of the 8080. The PIO processor uses this bus for communications with the master processor of the IPB/IPC. The other two I/O channels of the 8041 are I/O ports consisting of eight lines each. The first I/O port (pins P10-P17) is used by the PIO processor to transfer data to or from the four external devices. The second I/O port (pins P20-P27) is used to provide signals that augment or control the information that is transferred over the first I/O port. The signals of the two ports are collectively known as the PIO bus.

Five of the eight pins of the second I/O port control the 74154 demultiplexer in the generation of 13 control signals. These thirteen demultiplexed control signals, plus three control signals directly from the second I/O port, make a total of 16 control signals generated by the PIO. Of these 16 control signals, three are used within the PIO to control data routing, one is an interrupt request to the IPB/IPC, and the remaining 12 are used directly by the external peripheral devices.

NOTE

The interrupt request (PARALLEL INT) is normally not used by system software, but is available for user applications.

5.2 PIO-IPB/IPC INTERFACE

The PIO-IPB/IPC interface (sheet 6) provides a direct communications path between the IPB/IPC master processor and the PIO processor. All such communications are initiated by the IPB/IPC master processor with the PIO processor operating as a slave.

A distinct protocol is used to accomplish inter-processor transfers. This protocol is based on requirements of the data byte buffer (DBB) within the 8041. The DBB provides storage for data and four flags that are independently accessable to both the PIO and master processor. The flags include IBF (input buffer full), OBF (output buffer full), F0 (8041 busy), and C/D (a flag that identifies PIO input data as a command or a data byte and PIO output data as either a status or a data byte).

A complete description of the DBB and its flags is provided in Chapters 2 and 3. However, to understand PIO operations it is necessary only to be aware that the flags are used by both processors to establish a master/slave relationship. The IPB/IPC master processor reads the flags to determine when the PIO has accepted an input byte, when a data or status byte is being returned, or when a prior PIO operation has been completed. The PIO processor reads the flags to determine when a command or input data byte has been received or when a status or output byte has been accepted by the IPB/IPC. Both processors set and reset the flags as necessary to properly indicate local status.

The IPB/IPC I/O port drivers and the DBB are enabled by SEL PIO/. This signal is derived from the port address. The direction of the transfer is conveyed to the DBB by MASTER RD/ (PIO output) and MASTER WR/ (PIO input), and the type of PIO input byte (command or data) is defined by the state of MASTER A0 (i.e., the LSB of the IPB port addresses used to transfer commands and data). The I/O port drivers use only MASTER RD/ to determine the direction of the transfer. The PIO cannot enable the I/O port drivers and therefore cannot initiate a transfer. The PIO only responds to commands and sets the DBB flags to indicate when the requested byte has been accepted or placed in the DBB output buffer.

The other signals of the PIO-IPB/IPC interface are PARRALLEL INT (PIO interrupt) and INITLZ/ (system reset). The PIO interrupt is derived from the MSB of the PIO bus (P27). System reset (INITLZ/) occurs during power-on and when the RESET switch is pressed. In either instance, the PIO software is reinitialized, and all current operations of the peripheral devices are cancelled.

5.3 PERIPHERAL DEVICE INTERFACE

As previously mentioned, the first port (port 1) of the 8041 is used for bidirectional data transfers to or from the various peripheral devices of the PIO subsystem, and the second port (port 2) is used to select the data path and to control the transfer. As shown on sheet 7, three separate pairs of 8226 bidirectional bus drivers are interfaced to port 1 (P10-P17). One pair of drivers (A78 and A79) is used for two-way exchanges with the PROM programmer. A second

pair (A73 and A74) supplies an output data byte to the paper tape punch or accepts an input data byte from the paper tape reader. The third pair of drivers (A75 and A76) provides an output data byte to the line printer or returns a byte that reflects the status of all of the associated PIO peripherals. The four least-significant bits of port 2 (P20-P23) are decoded by the 74154 decoder (A77) on sheet 6 to generate one of thirteen control signals. The P24 bit from port 2 is used to enable the decoder, and the P25 and P26 bits are used to control the direction of the reader/punch and PROM programmer bus driver pairs, respectively. Note that to ensure that the control signal decode lines (P20-P23) are stable when the decoder is enabled, the 8041 executes a two-instruction sequence. The first instruction defines the control signal and bus driver direction, but does not enable the decoder (P24 remains high). The second instruction then activates P24 without altering the states of the other port 2 bits to generate the required control signal. Table 5-1 defines the control signal decoding.

Whenever the system is initialized or following any input or output operation, the 8041 outputs an all "ones" byte from port 1 to "program" the port for bidirectional operation and initializes the bus driver pairs for output through port 2.

To output a data byte, the 8041 performs the following sequence:

1. The data byte to be output to the peripheral is placed on the port 1 data lines (P10-P17).

2. The first of a series of control bytes is output from port 2. This byte defines the peripheral control signal to be generated (P20-P23) and ensures that the bus driver pairs are set for data output (P25 and P26 high). Note that since the decoder is currently disabled, the STATUS ENABLE/ output will be inactive, and the line printer/status driver pair will be set for output.

3. The next control byte enables the decoder (P24 low) to activate the decoded control signal. Note that although the output data byte is available from all three bus driver pairs, it is only accepted by the peripheral receiving the control signal.

4. A subsequent control byte is output from port 2 to disable the decoder (P24 high) in order to return the selected peripheral control signal to an inactive level.

Table 5-1. Address Decoding for Peripheral Device Control Signals

Port Bits (P23)	(P22)	(P21)	(P20)	Control Signal Active	Control Signal Function
0	0	0	0	PPWC0/	PROM programmer write strobe 0
0	0	0	1	PPWC2/	PROM programmer write strobe 2
0	0	1	0	PPWC1/	PROM programmer write strobe 1
0	0	1	1	PPRC0/	PROM programmer read request 0 (data)
0	1	0	0	PPRC1/	PROM programmer read request 1 (status)
0	1	0	1	INIT/	PROM programmer reset
0	1	1	0	PUNCH COMMAND/	Paper tape punch operation
0	1	1	1	DL/	Paper tape reader left tape advance
1	0	0	0	DR/	Paper tape reader right tape advance
1	0	0	1	LPT DATA STROBE/	Line printer data strobe
1	0	1	0	LPT CTL 1/	Line printer control 1
1	0	1	1	LPT CTL 2/	Line printer control 2
1	1	0	0	STATUS ENABLE/	Device status input enable
1	1	0	1	none	Unused
1	1	1	0	none	Unused
1	1	1	1	none	Unused

The operation of the 8041 during data input is dependent on the source of the data byte. When the input data byte originates from the PROM programmer, the 8041 defines the control signal (PPRC0/ or PPRC1/) and sets the PROM programmer bus driver pair for input (P26 low) with the first control byte and then generates the control signal by enabling the decoder with the second control byte. The 8041 accepts the input byte at port 1 and outputs a subsequent control byte to terminate the control signal. When the input data byte originates from the paper tape reader, the 8041 outputs a control byte to set the reader/punch bus driver pair for input (P25 low) and then reads in the data byte at port 1. (Note that only one control byte is output since the paper tape reader does not require a read control signal.) To read in the peripheral status byte, the 8041 defines the control signal (STATUS ENABLE/) with the first control byte and generates the signal with the second control byte. STATUS ENABLE/ sets the line printer/status bus driver pair for input, and the 8041 accepts the status byte at port 1. A subsequent control byte terminates the control signal.

The peripheral status byte input to the 8041 reflects the general status of all of the associated PIO peripherals. This byte is used by the 8041 to derive the more detailed individual peripheral status bytes that are available to the master processor on command. Table 5-2 defines the individual bits of the peripheral status byte (refer to the *Intellec Series II Hardware Interface Manual* for definitions of the individual status bytes available from the PIO subsystem). Note that two of the status bits (ACKNOWLEDGE/ from the line printer and PPACK/ from the PROM programmer) are asynchronous pulses and therefore cannot be considered valid in the status byte. The PPACK/ (PROM programmer acknowledge) pulse is detected by the 8041's T1 input.

Table 5-2. Device Status Byte Format

Port Pin	Signal Mnemonic	Function
P10	PUNCH READY/	True when paper tape punch is ready to perform punch operation.
P11	SYSTEMS READY/	Not generated by standard paper tape punch. Indicates that power is applied to tape punch.
P12	DATA READY/	True when paper tape reader is ready to advance left or right.
P13	SYSTEMS READY/	Not generated by standard paper tape reader. Indicates that power is applied to tape reader.
P14	BUSY/	True when printer is performing a printer operation.
P15	ACKNOWLEDGE/	Pulse indicating printer busy condition. The 5.0 microsecond pulse of the standard printer cannot be detected by the PIO.
P16	SELECT/	True when printer function switch is set to on-line.
P17	PPACK/	Pulse indicating that PROM programmer is ready to input a data byte. The 100 nanosecond pulse of the standard PROM programmer cannot be detected by the PIO on input port P10-P17, and thus is detected by input T1.

Note that the bidirectional bus driver outputs are inverted.

CHAPTER 6
KEYBOARD ASSEMBLY

The keyboard assembly is unique among subcontractor-supplied components of Intellec Series II development systems in that it contains an Intel-designed LSI component.

6.1 PRINCIPLES OF OPERATION

To enable the detection of keystrokes, the keys are logically organized into a matrix of rows and columns. Since there are 62 individual keys, an 8-by-8 matrix is used. Strobes are sequentially applied to the individual rows, and all columns are continually monitored. When a key is depressed, the associated row strobe is applied to the corresponding column output to the 8041. The known row strobe output combined with the active column input allows the 8041 to identify any depressed key.

Most keystrokes detected are simply converted to the appropriate ASCII character code and forwarded to the IOC on demand. However, three keys (SHIFT, CNTL, and TPWR) are used by the 8041 to modify the code generated for most of the other keys. The SHIFT key determines if the code output is to be for an upper- or lower-case character. The CNTL (control) key, when used in conjunction with a character key, causes the generation of a non-printable control character. Finally, the TPWR (typewriter) key is used to partially override the SHIFT key in that when the TPWR key is deselected, alphabetic character keys always produce an upper-case code. The TPWR key, which is mechanically latched on or off, does not alter the codes of the numerical or any other non-alphabetic keys.

The REPEAT KEY is used in conjunction with any code producing key to cause the code of the latter key to be generated at a rate of 10 times per second. As with the SHIFT, CNTL, and TPWR keys, the REPEAT key produces no ASCII code when used alone. Some additional keys (i.e., the TAB and cursor control keys) are unique in that they produce a code equivalent to the code produced by the CNTL and an alphabetic key (TAB and CNTL-I produce the same code).

The 8041 generates row strobes and detects column pulses to determine when a key is pressed and also determines when more than one key is pressed. If one of two keys pressed is the SHIFT, CNTL or REPEAT key, the 8041 produces the required ASCII code. If any other two keys (except TPWR) are pressed concurrently, the code of each key is generated as the key is pressed. This action is known as "N key rollover" and allows the overlapping of keystrokes without the loss of data.

Character codes for keystrokes are forwarded via the IOC to the IPB/IPC as the master processor issues keyboard read commands. These commands are normally issued at a rate more than adequate to stay abreast of rapid typing. However, if higher-priority processing delays the keyboard read command, up to eight characters may be temporarily stacked by the 8041 keyboard processor. When the stack is full, further keystrokes are lost.

Only two types of commands are issued by the IOC to the keyboard. The first of these commands is the keyboard read data command discussed above. This command occurs when the IOC reads data from I/O port address 90. The second command is the keyboard read status command that occurs when the IOC reads data via I/O port 92. The only distinction between these two port addresses is the second least-significant bit (ADR1). This bit (STATUS/DATA) is applied to the A0 input of the 8041 keyboard processor. The keyboard processor thus supplies either a data byte containing the character code or a status byte containing the 8041's data byte buffer (DBB) flags. Note that the keyboard commands either select the DBB output buffer (to access keyboard characters) or the DBB flag register (to access the DBB flags). This arrangement allows internal operations of the keyboard processor to continue indefinitely without any intervention from the IOC. Furthermore, the only DBB flag of significance to the IOC is the output buffer full (OBF) flag. This flag is set when a keyboard character code is ready for transfer to the IOC. The busy flag (F0), command/data flag, and IBF flag are not used since the keyboard processor receives no data from the IOC and since there is no need to inhibit the issuance of commands by the IOC. The F0 flag is however used to indicate keyboard selection of local operation. (In local operation, the keystrokes are displayed on the CRT, but do not otherwise affect operation of Intellec Series II development system.) To select local operation the SHIFT, CNTL, and L keys are pressed concurrently. To return to on-line operation, the SHIFT, CNTL and O keys are pressed concurrently. Local operation can also be selected by placing the DIAGNOSTIC/LINE/LOCAL switch on the rear panel of the system chassis to the LOCAL position. The keyboard status byte contains the OBF flag in the LSB position (DB0) and the on-line/local (F0) flag in the third LSB position (DB2).

6.2 KEYBOARD ELECTRONICS

The keyboard electronics (refer to sheet 2 of the keyboard specification schematic drawing) consist of the 8041 keyboard processor, a 74154 row- select demultiplexer, two 7404 line drivers, a 3.58 MHz crystal, and 61 switches that form the key matrix and provide an output on coincidence of a row strobe and key activation. The interface between the keyboard processor and the IOC is very simple. All transfers via the data lines (DB0-DB7) are from the keyboard to the IOC through the 7404 line drivers. An output is supplied to the IOC when READ KB/ goes low. The output is an ASCII character code when STATUS/DATA is low or the DBB flags when STATUS/DATA is high. No other control lines are used.

Port 1 of the 8041 keyboard processor provides both a binary coded row address and a row strobe (demultiplexer enable). To ensure that the row address (P12-P15) is stable when the row strobe (P10 and P11) goes active, the row address is first applied to the demultiplexer with one instruction (without a row strobe) and the demultiplexer is then enabled (while maintaining the row address) by a second instruction. When strobed, the decoded row-select output goes active and enables all of the switches within the associated row. When a key is pressed and its row is strobed, the corresponding column output goes active. The eight key matrix column outputs are applied to the port 2 (P20-P27) inputs of the 8041.

REQUEST FOR READER'S COMMENTS

The Microcomputer Division Technical Publications Department attempts to provide documents that meet the needs of all Intel product users. This form lets you participate directly in the documentation process.

Please restrict your comments to the usability, accuracy, readability, organization, and completeness of this document.

1. Please specify by page any errors you found in this manual.

2. Does the document cover the information you expected or required? Please make suggestions for improvement.

3. Is this the right type of document for your needs? Is it at the right level? What other types of documents are needed?

4. Did you have any difficulty understanding descriptions or wording? Where?

5. Please rate this document on a scale of 1 to 10 with 10 being the best rating. _____

NAME _____ DATE _____

TITLE _____

COMPANY NAME/DEPARTMENT _____

ADDRESS _____

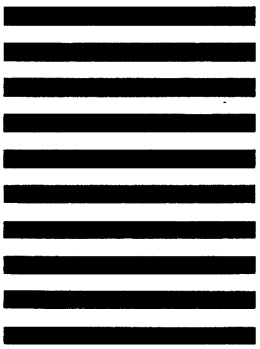
CITY _____ STATE _____ ZIP CODE _____

Please check here if you require a written reply. ☐

WE'D LIKE YOUR COMMENTS ...

This document is one of a series describing Intel products. Your comments on the back of this form will help us produce better manuals. Each reply will be carefully reviewed by the responsible person. All comments and suggestions become the property of Intel Corporation.

NO POSTAGE
NECESSARY
IF MAILED
IN U.S.A.



BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 1040 SANTA CLARA, CA

POSTAGE WILL BE PAID BY ADDRESSEE

Intel Corporation
Attn: Technical Publications M/S 6-2000
3065 Bowers Avenue
Santa Clara, CA 95051

INTEL CORPORATION, 3065 Bowers Avenue, Santa Clara, California 95051 (408) **987-8080**

Printed in U.S.A.