GENERAL AUTOMATION

GA-16/110/220 maintenance manual

88A00509A-B

NOTICE

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

General Automation makes no warranty or representation with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. General Automation shall not be liable for errors contained herein.

General Automation assumes no responsibility for the use or reliability of its software on equipment that is not furnished by General Automation.

88A00509A-B

GA-16/110/220
maintenance manual

GENERAL AUTOMATION, INC.
1055 South East Street
Anaheim, California 92803
(714) 778-4800

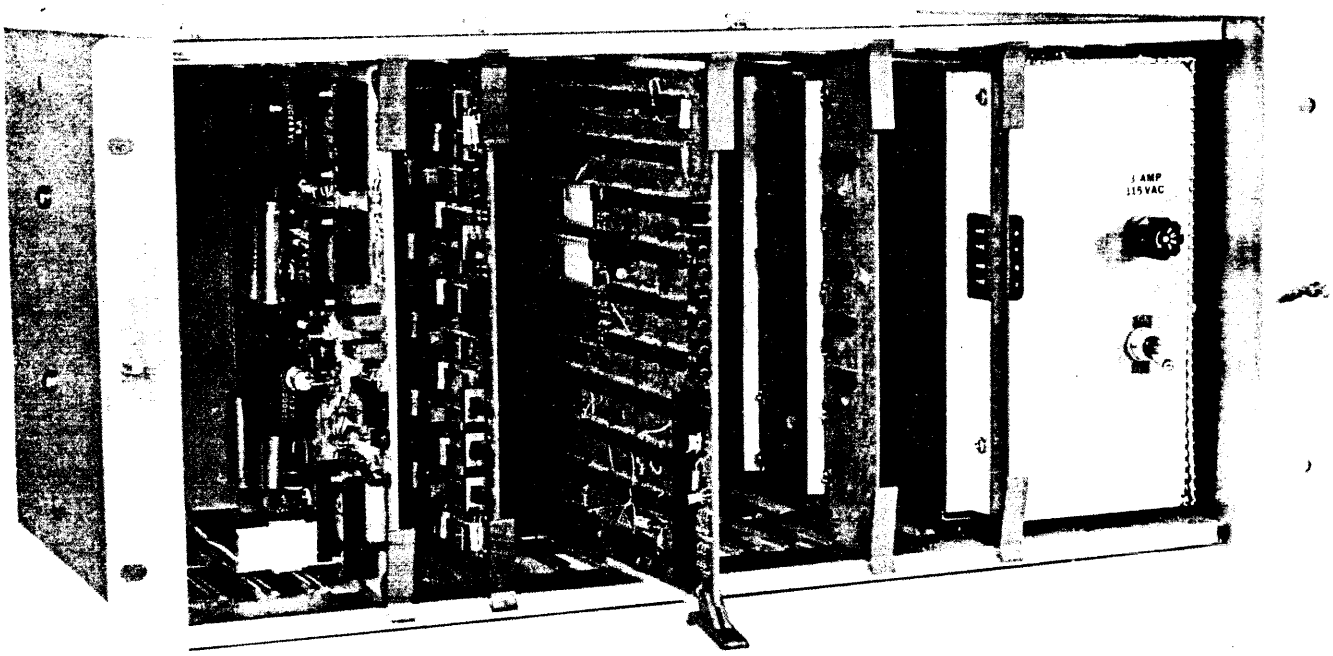
RECORD OF REVISIONS

Title: GA-16/110/220 Maintenance Manual

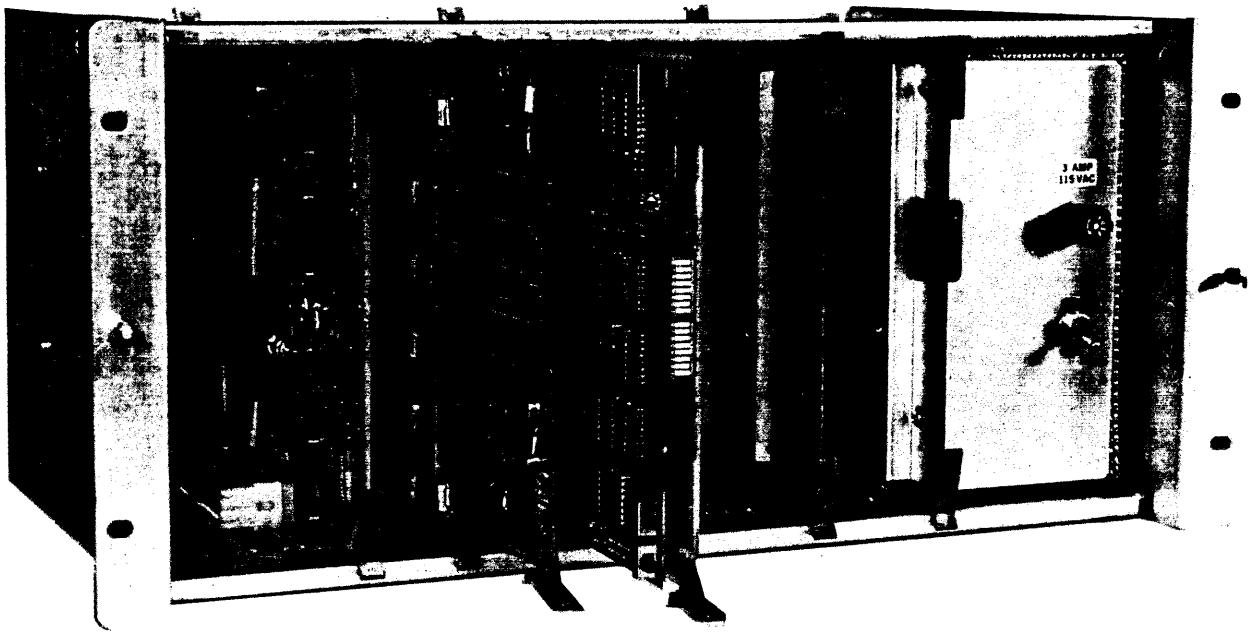
Document Number: 88A00509A-A

DATE	ISSUE
Apr 77	Original Issue
Sep 78	Change A1 - Pages affected: xi, xii Pages added: G-1, G-2
Mar 79	Revision B

FRONTISPIECE



GA-16/110 In Compact Chassis



GA-16/220 In Compact Chassis

FOREWORD

This GA-16/110/220 manual describes the configuration, functional operation, and interfaces of both the GA-16/110 and GA-16/220 computers. Both computers are similar in their basic design, configurations, and programming and are described in one manual.

The purpose of this manual is to supplement the GA-16/110/220 System Reference Manual by supplying detailed information on the CPU hardware. This manual is intended to give a better understanding of the processor logic functions to those individuals with primary responsibility for maintaining the system hardware.

The following documents are recommended for GA-16/110/220 hardware maintenance and supplemental systems information:

Document Number	Title
88A00525A	How to Use Your GA-16/220
88A00508A	GA-16/110/220 System Reference Manual
94A01519A	T&V Reference Manual
94A01531A	Stand-Alone Utilities
90C02422A	Logic Diagram CPU 1
31P02422A	Parts List CPU 1
31D02422A	Assembly Drawing CPU 1
90C02429A	Logic Diagram CPU 2
31P02429A	Parts List CPU 2
31D02429A	Assembly Drawing CPU 2
90C02405A	Logic Diagram - Microconsole
31P02405A	Parts List - Microconsole
31D02405A	Assembly Drawing - Microconsole
90C02410A	Logic Diagram - 8K Memory
31P02410A	Parts List - 8K Memory
31D02410A	Assembly Drawing - 8K Memory
90C02301A	Logic Diagram - Memory Service Module
31D02301A	Assembly Drawing - Memory Service Module
31P02301A	Parts List - Memory Service Module
31D02277A	Assembly Drawing Piggy Back 2K RAM
90C02277A	Piggy Back 2K RAM
31D02371A	Assembly Drawing Memory Parity Protect

Document Number	Title
90C02371A	Memory Parity Protect
51D00073A	Jumbo Power Supply
90C02430A	Battery Back-up Reg.
90C02301A	Logic Diagram - Memory Service Module
31D02301A	Assembly Drawing - Memory Service Module
31P02301A	Parts List - Memory Service Module
90A02469A	Logic Interconnect Diagram - Compact MIB
31D02469A	Assembly Drawing - Compact MIB
31P02469A	Parts List - Compact MIB
90A02409A	Logic Interconnect Diagram - Jumbo MIB
31P02409A	Parts List - Jumbo MIB
31D02409A	Assembly Drawing - Jumbo MIB
01D01386A	Assembly Drawing - AC Power - Battery Back-up

The basic unit for both computers is the GA-16/110 CPU-1 module, which provides a microprogrammed CPU to carry out the GA-16/SPC-16 series instruction set. The GA-16/110 provides controls and indicators, memory bus, programmed I/O bus, vectored priority interrupts, cold start, real time clock interrupt, power fail/auto restart, and operations monitor alarm. An optional piggyback memory module plugs into the GA-16/110 module to provide 2K words of random access memory (RAM) (or combination RAM and programmable read-only-memory (EPROM)). An optional 64-word ROM may be installed on the piggyback memory to provide initial program load (IPL) capability from a peripheral device. A complete GA-16/110 computer (with memory) can, therefore, be provided on a single printed circuit module for insertion into a user's device, such as an "intelligent" terminal.

A GA-16/220 computer consists of the GA-16/110 module plus a second module (CPU-2 module) which adds additional controls including data entry switches that may be sensed by a users program, and a vectored console interrupt button. The 220 module also provides a 1 millisecond real-time clock (RTC), memory mode change under program control, and control instructions for I/O reset and single-step execution. The 220 module enlarges the I/O capabilities to include a built-in serial I/O controller (with vectored interrupt) for TTY or CRT and a direct memory access (DMA) port.

An optional system console interface module (SCI) may be plugged into the 220 module to provide an interactive operator program via TTY or CRT (console ROM). An optional IPL ROM may be installed on the SCI to provide program load capabilities from a variety of peripheral devices. Additional controls on the SCI select and adjust TTY baud rate, select the IPL device, and initiate IPL.

Both the GA-16/110 and GA-16/220 computers may be installed in either a compact or jumbo enclosure which provides printed circuit card slots and connectors for the GA-16/110 CPU and the 220 module. In addition, it also provides slots and connectors for the installation of 4K or 8K dynamic RAM memory modules, a memory parity protect (MPP) module, a memory power supply with backup batteries, and a control module for the battery backup. I/O controller slots are also provided together with rear interface connectors for peripheral devices and provision for the connector of an I/O expansion chassis. The compact enclosure provides a compartment for a main power supply to run the CPU and I/O controllers, while the jumbo enclosure requires an external power supply unit.

Specific differences between the two computers are defined in the text, and the term "GA-16/110/220" is generally used where characteristics are shared.

CAUTION

Several standard GA controllers have had to be updated for compatibility with both the SPC-16 and GA-16 series processors. The controllers affected are listed below:

Model No.	Name
1615-xxxx	CIT-16
3346-3347-xxxx	Disk Controller
1615-xxxx	MHSDC
1615-xxxx	ABTU
1615-0240	APU
1561-xxxx	Asynchronous Communication Controller
1581-xxxx	Asynchronous Communication Controller
1574-xxxx	DMA Asynchronous Communication Controller
1530-xxxx	DMA MUX
1579-xxxx	SDLC
1582-xxxx	Console Controller
1571-xxxx	Paddle Board (Loop Back)

Controllers constructed prior to May 1, 1976, may reflect the proper changes. Any of these controllers, not incorporating the necessary modifications, are not compatible with the GA-16/440, 16/330, 16/220, and 16/110 processors.

Controllers constructed after May 1, 1976, are fully compatible with all GA-16 series and SPC-16 series processors.

Consult your General Automation Field Representative for further details concerning these controllers.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

1 introduction

The GA-16/110 CPU is a complete computer containing CPU, memory, and I/O on a single, 140-pin connector, printed circuit board measuring 7 3/4" x 11" (19.7 cm x 28.0 cm). The GA-16/110 is specifically designed as a "load-and-go" type of processor for dedicated systems, such as, remote controllers and concentrators in large networks.

The I/O-Bus on the GA-16/110 is totally compatible with the SPC-16 and all other GA Series-16 computer systems. All GA-programmed I/O controllers (PIO) operate on the 16/110 systems, however, the 110 has no capability for direct memory access (DMA) controllers. If DMA access is required, the GA-16/110 must be upgraded to a GA-16/220.

The GA-16/110 may be upgraded to a general-purpose GA-16/220 CPU by the addition of one 7 3/4" x 11" module, providing the capabilities of a GA-16/220 as identified in Table 1-1.

Table 1-1. GA-16/110/220 Computer Specifications (Sheet 1 of 5)

Architecture	Microprogrammed 16-bit General-Purpose Computer Parallel Binary Two's Complement Arithmetic Single- and Double-Word Instructions Programmed and Direct Memory Access Input/Output	
Technology	Processor -nMOS LSI and Tri-State (7400, 7400H and Schottky Logic) Circuitry and Unique Four-Layer Printer Circuit Board Packaging	
Memory	Semiconductor RAM, PROM, and EPROM	
Registers	Programmable Registers	16
	Accumulators	6
	Index/Accumulators	6
	Base Register	2
	Subroutine Linkage	2
	Shift Counter Register	1
	Program Counter	1
	Instruction Register	1

Table 1-1. GA-16/110/220 Computer Specifications (Sheet 2 of 5)

Registers (Continued)	Status Indicators	7
	Zero	
	Plus	
	Link	
	Overflow	
	Foreground/Background	
	32K/64K Memory Mode	
	Interrupt System Enable (ISE) Save	
	Shift Count	1
Standard Processor	Real-Time Clock (RTC)	1 ms
	Operations Monitor Alarm (OMA)	150 ms - 300 ms
	Power Fail/Auto Restart (PF/AR)	
	Cold Start	
	64K Direct Addressing	
	Memory Parity Protect (MPP) (Optional)	
	Interrupt Program Timeout (Optional)	
Arithmetic and Logic	Parallel Binary, Two's Complement, Fixed-Point, Bit, Byte, and Word	
	ADD, SUB, COMP, INC, DEC, AND, OR, XOR, SET BIT, RESET BIT, TEST BIT	
	Hardware Multiply and Divide (Unsigned)	
	Signed Multiply/Divide (Optional)	
Instructions (Fourteen Classes)	Memory Reference	
	Memory Referenced Indexed	
	Conditional Jump (Skip) on Eight (8) Conditions	
	Register Operate and Register Operate and Compare	
	Register Operate Literal and Register Operate Literal and Compare	
	Subroutine Return Via Indirect Vector	
	Register Change	
	Shift Left	
	Shift Right	
	Control	
	Input/Output (Addressing to 64 Device Select Codes)	
	Read Console Switches (Additional Input/Output on GA-16/220)	
	Multiply/Divide (Unsigned)	
	Special (Enable Single Step Interrupt, I/O Reset) (Additional Input/Output on GA-16/220)	

Table 1-1. GA-16/110/220 Computer Specifications (Sheet 3 of 5)

Addressing (Eleven Modes)	Direct Direct, Indexed Indirect Indirect, Indexed Program Relative Program Relative, Indirect Base Relative Base Relative, Indexed Base Relative, Indirect Base Relative, Indirect, Indexed Literal
Programmed I/O (PIO)	16-Bit Parallel Transfers Vectored Priority Interrupts 120K Word Transfer Rate Data Transfer To/From Memory To/From Any One of the Sixteen (16) Programmable Registers
Serial I/O (GA-16/220)	Integral Universal Asynchronous Receiver Transmitter (UART) Serial I/O Controller for Console TTY or CRT Data Rates 110 or 9600 Baud (Early Model CPU2 boards) Sixteen Rates, 0 thru 9600 Baud (CPU2 31D02574A)
Direct Memory Access (DMA) I/O (GA-16/220)	16-Bit Parallel Transfers Multi-Channel Operation with Vector Priority Sequencing 900K Word-Per-Second Transfer Rate Interleaved with CPU
Interrupts	Non-Inhibitable (NI) Interrupts Power Fail Auto-Restart Memory Parity, Write Protect, and Program Timeout (Options) TRAP Instruction Single Step/Break (GA-16/220)

Table 1-1. GA-16/110/220 Computer Specifications (Sheet 4 of 5)

Interrupts (Continued)	Internal Inhibitable (IN) Interrupts (GA-16/220) Real-Time Clock Console TTY Console Interrupt External Inhibitable (IN) Interrupts 64 Priority Interrupt Levels
Dimensions	GA-16/110 CPU Module or Additional 220 Module Height 7.250 in. (18.415 cm) Width 0.625 in. (1.588 cm) Width w/Piggyback RAM (or SCI) 1.063 in. (2.72 cm) Depth 11.000 in. (27.940 cm) Compact GA-16/110/220 System with Internal Power Supply or Jumbo GA-16/110/220 System (Requires Separate Power Supply) Height 8.750 in. (18.415 cm) Width 19.000 in. (48.260 cm) Depth (w/o Cables) 21.150 in. (53.721 cm) Depth (w/Cables) 22.250 in. (56.515 cm) Separate Power Supply for Jumbo GA-16/110/220 System Height 10.750 in. (27.305 cm) Width 19.000 in. (48.260 cm) Depth (w/o Cables) 5.750 in. (14.605 cm) Depth (w/Cables) 7.750 in. (19.685 cm)
Temperature	Operable at $0^{\circ}C$ to $50^{\circ}C$
Humidity	Up to 90 Percent Relative (Non-condensing)
Input/output Interface Signals	Low Level - 0.0 to 0.4 volt High Level- 2.0 to 5.0 volts I/O Drivers - 74365 Tri-State TTL Drivers Saturation voltage 0.4 volt Capable of sinking 40 ma to ground
I/O Cable Termination	Twisted Pairlines - 100Ω characteristics impedance Returns grounded at both ends

Table 1-1. GA-16/110/220 Computer Specifications (Sheet 5 of 5)

Power	GA-16/110 Module Only +5V@2.7A, +15V@0.092A, -15V@0.012A Optional Piggyback RAM Memory for GA-16/110/220 w/IPL +5V@1.620A, w/o IPL +5V@1.236A GA-16/220 Module +5V@2.58A, +15V@0.000A, -15V@0.012A 8K by 16 Memory Module +5V@1.5A, -5V@0.004A, +12V@0.400A MPP +5V@2.25A Optional System Console Interface (SCI) w/IPL +5V@1.2A w/o IPL +5V@0.64A GA-16/110 or GA-16/220 Power Supplies for Compact and Jumbo MIB 115 Volts AC, Single-Phase ±10 Percent, 47 to 63 Hz (Internal or External Power Supply) 220 Volts AC, Single-Phase ±10 Percent, 47 to 63 Hz (External Power Supply Only) 230 Watts Maximum (Compact) 500 Watts Maximum (Jumbo)

1.1 COMPONENT IDENTIFICATION

Figures 1-1 through 1-7 are supplied to aid in component module identification and installation.

1.1.1 MEMORY MODULES

The Memory Modules (Figure 1-1) may be installed in J-9, J-10, J-11, and J-12 of the compact chassis (refer to Figures 2-2 and 2-5) and in J-11, J-12, J-13, J-14, J-15, J-16, and J-17 of the Jumbo Chassis (refer to Figures 2-3 and 2-6). The memory may contain one or more 4K, 8K, or 16K boards, or a 32K or 64K board.

1.1.2 GA-CPU-2 MODULE

The CPU-2 (Figure 1-2) module upgrades the GA-16/110 to a GA-16/220. The CPU-2 module is placed in J-8 of the compact chassis (Figure 2-5) or in J-10 of the Jumbo Chassis (Figure 2-6).

1.1.3 GA-CPU-1 MODULE

The CPU-1 Module (Figure 1-3) the nucleus of both the GA-16/110 and GA-16/220 has four possible locations:

1. In the Compact Chassis GA-16/110, the CPU-1 Module is located in J-8 (Figure 2-2).
2. In the Jumbo Chassis GA-16/110, the CPU-1 Module is located in J-10 (Figure 2-5).
3. In the Compact Chassis GA-16/220, the CPU-1 Module is located in J-9 (Figure 2-5).
4. In the Jumbo Chassis GA-16/220, the CPU-1 Module is located in J-11 (Figure 2-6).

1.1.4 CHASSIS ASSEMBLIES

Figure 1-4, 1-5, 1-6 and 1-7 are provided to pictorialy supplement Figure 2-3, 2-4, 2-5 and 2-6. These eight figures should answer any questions concerning actual card placement and chassis configuration. The following points should be noted:

- All cards, when viewed from the front, MUST have the component side to the left. This means that the TTY/CRT Connector and the Memory Service Module (when installed) must have their components facing to the right when viewed from the back of the chassis.
- The Memory Service Module *will not be installed* when a Back-up Power Supply is installed and *must be installed* when no Back-up Power Supply is in a system with plug-in dynamic RAM. It is not mandatory for the 2K piggyback static RAM.
- When a Back-up Power Supply is installed there are two AC line cords to plug into the AC supply line; one cord from the Back-up Power Supply itself, the other from the main systems power supply (internal or external).

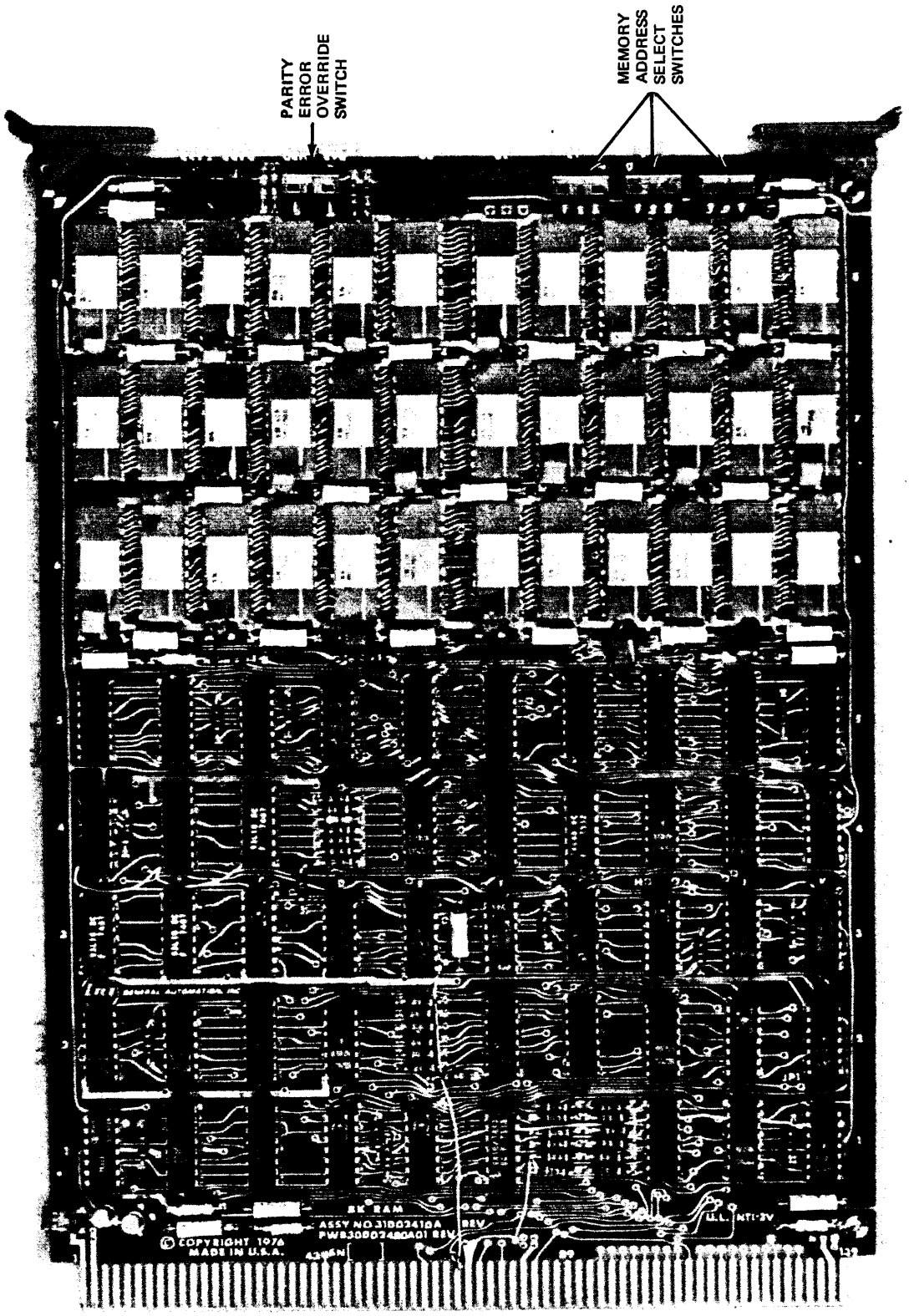


Figure 1-1. 8K x 18 Memory Module

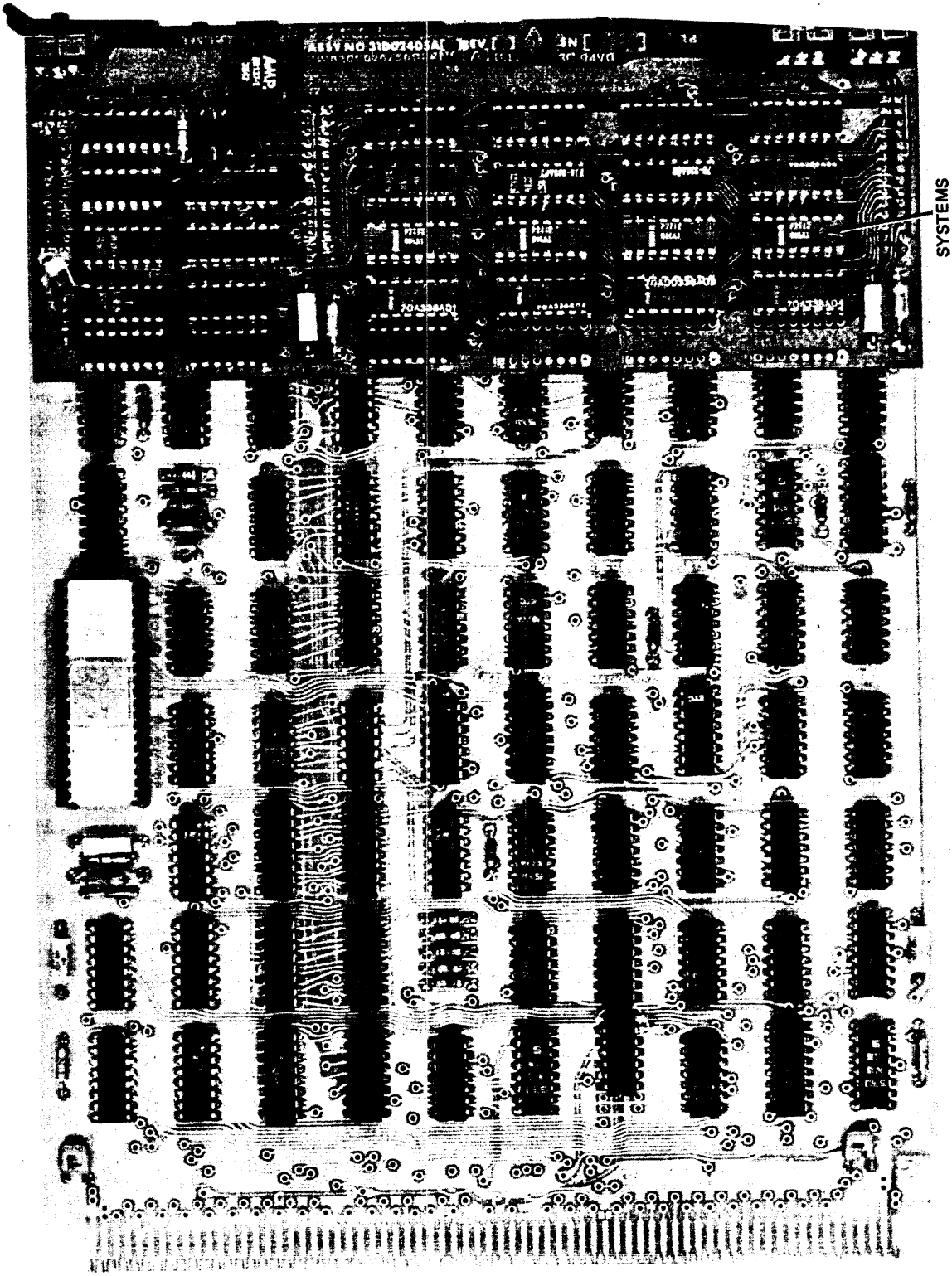


Figure 1-2. GA-CPU-2 Module with SCI

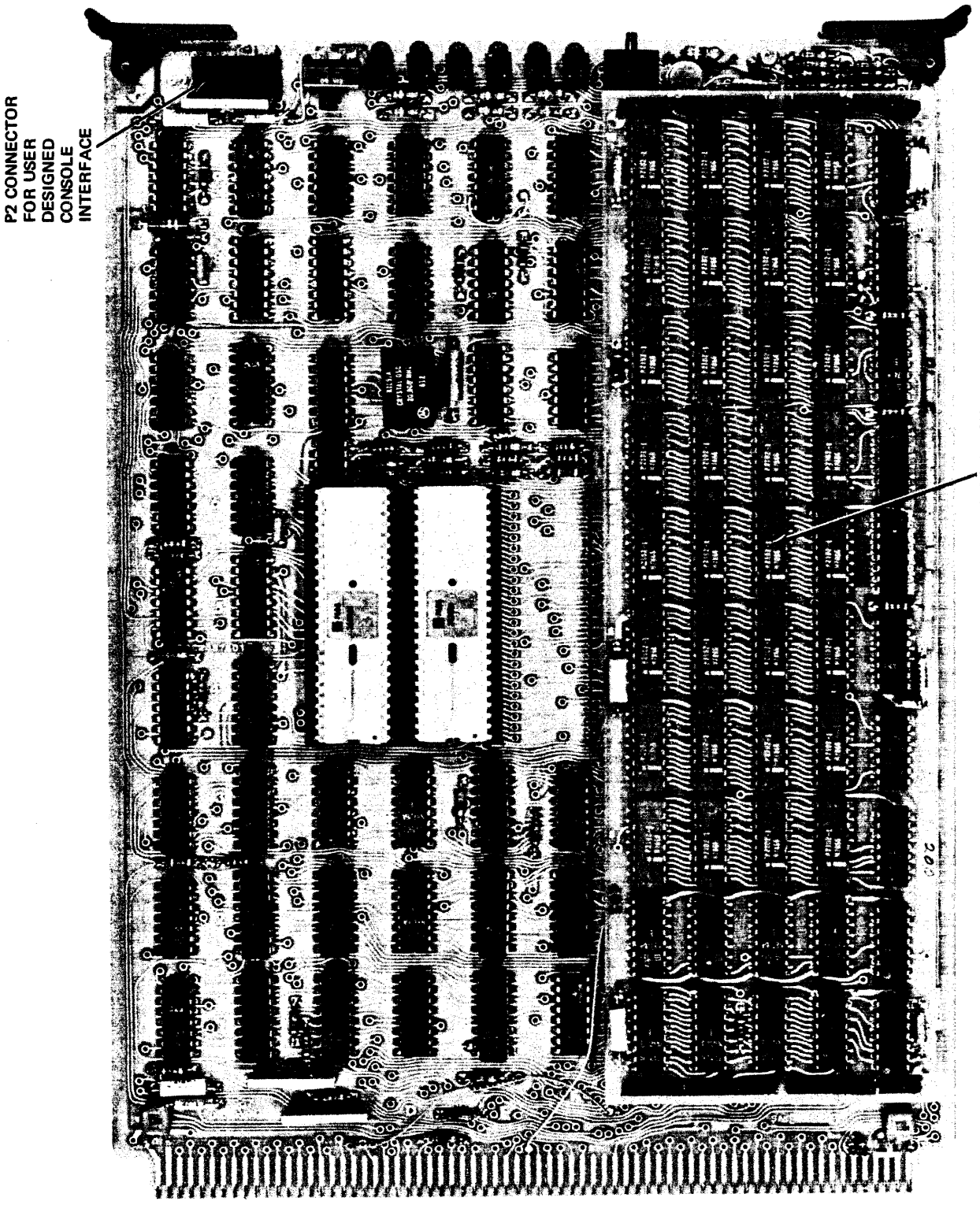


Figure 1-3. GA-CPU-1 Module With 2K Piggyback Memory

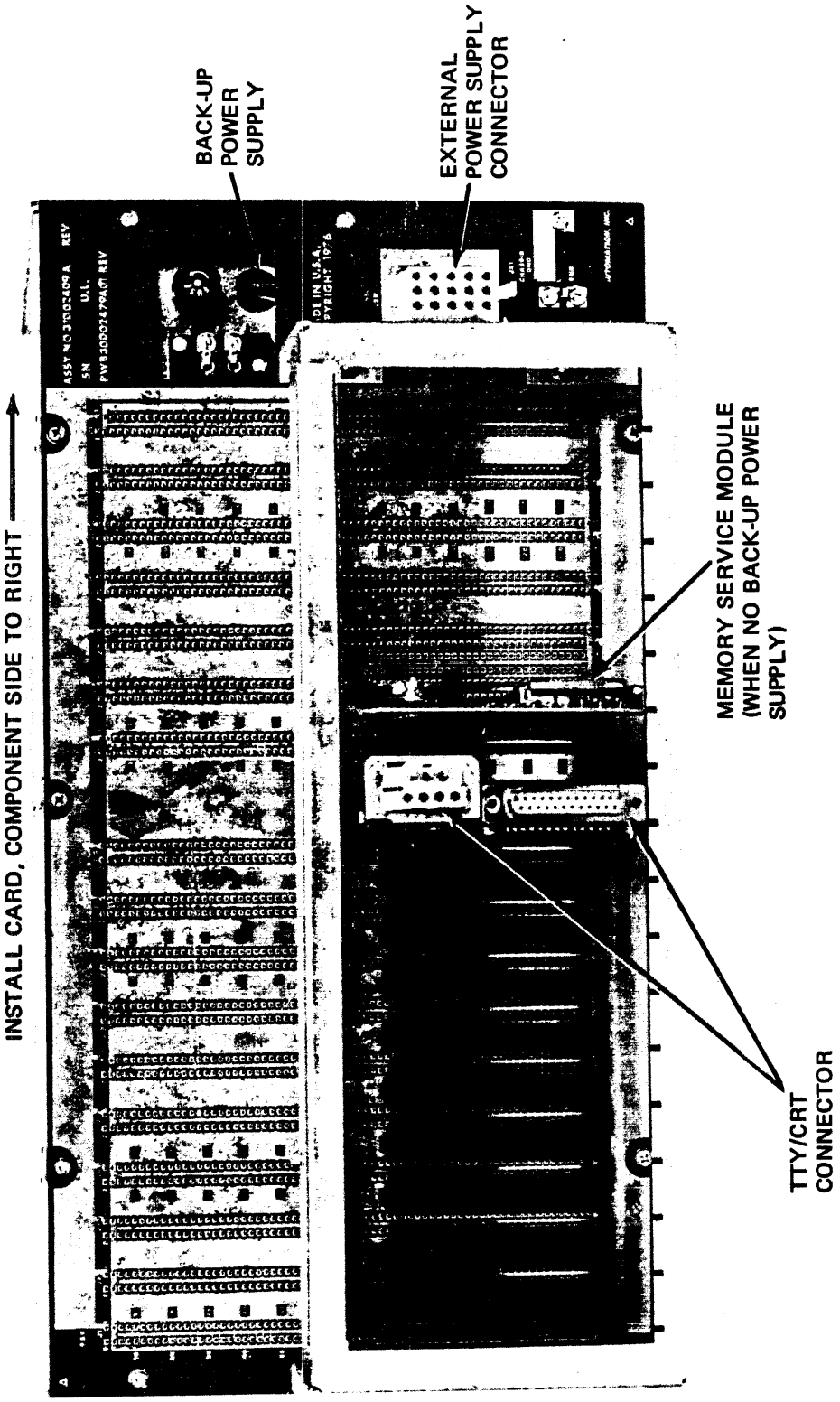


Figure 1-4. Jumbo Chassis (Rear View)

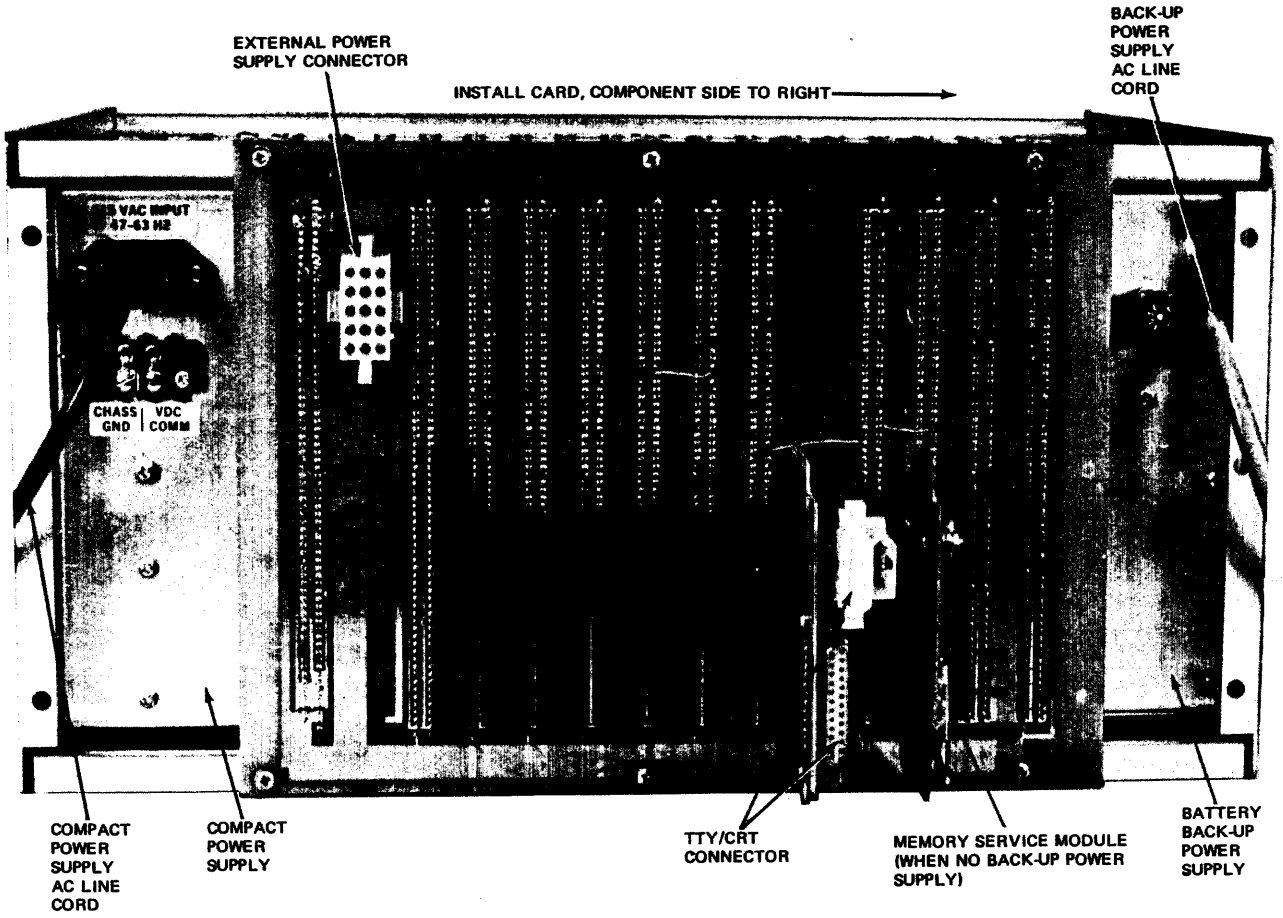


Figure 1-5. Compact Chassis (Rear View)

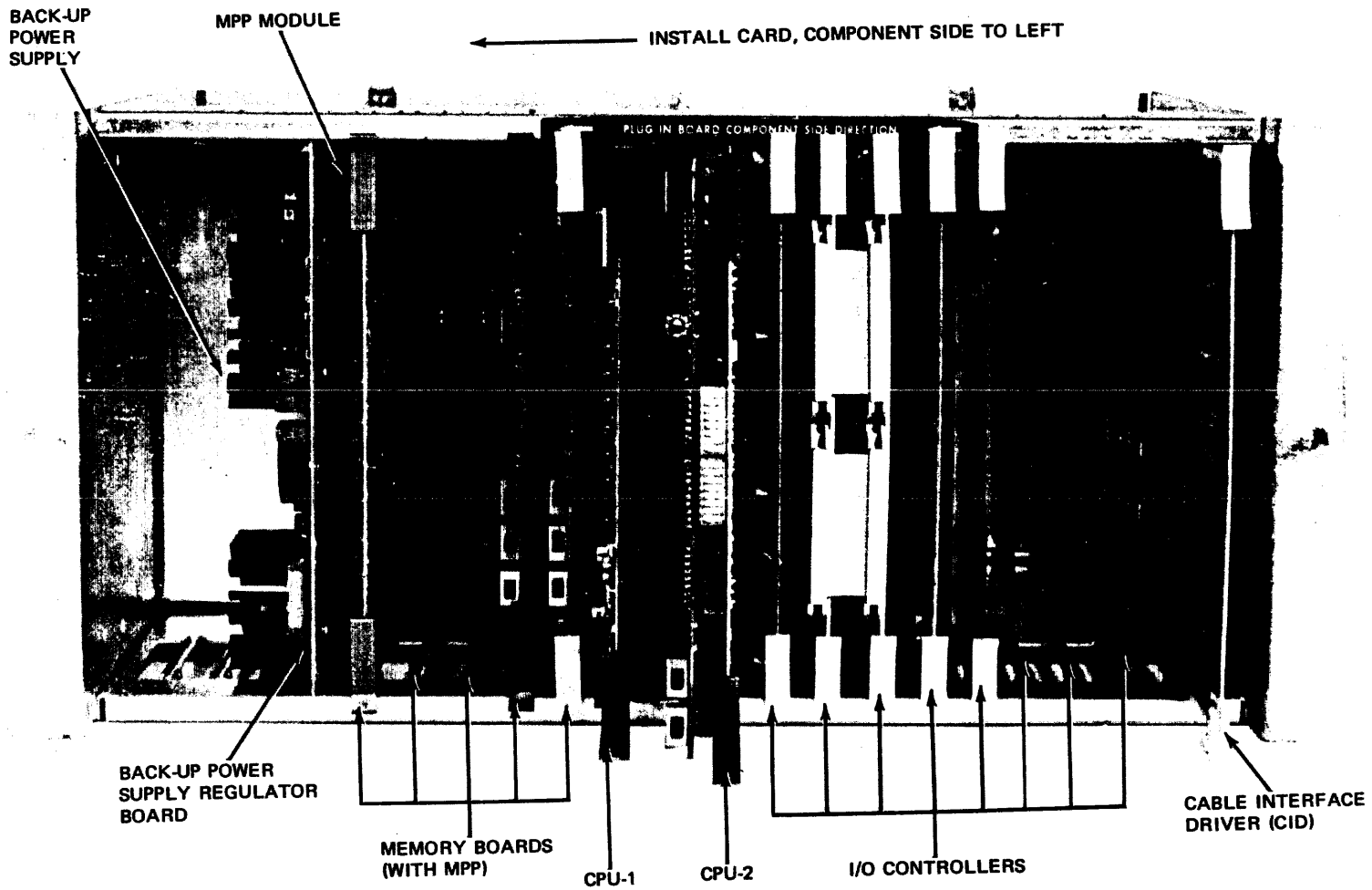


Figure 1-6. GA-16/220 Jumbo Chassis

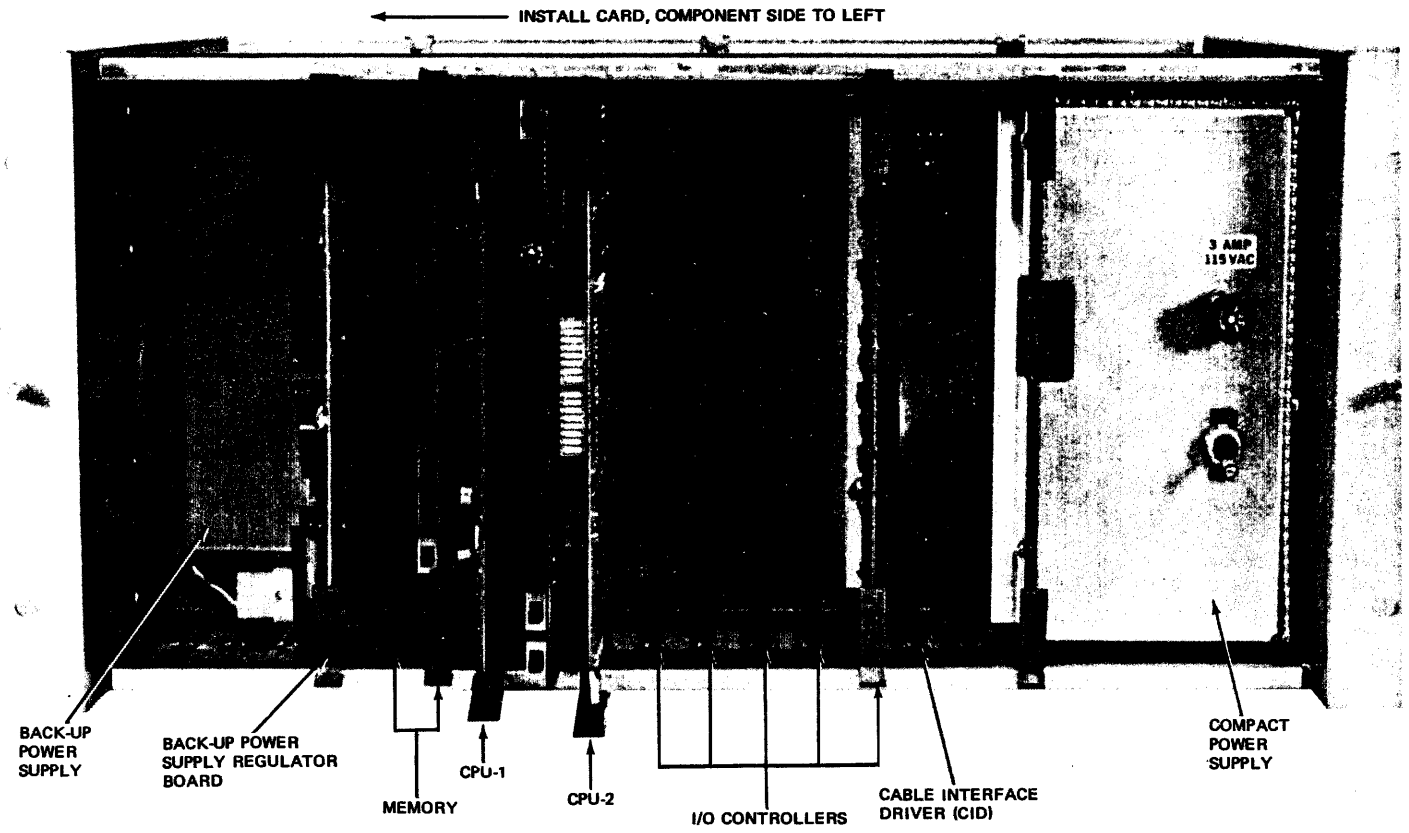


Figure 1-7. GA-16/220 Compact Chassis

GA-16/110/220 functional description 2

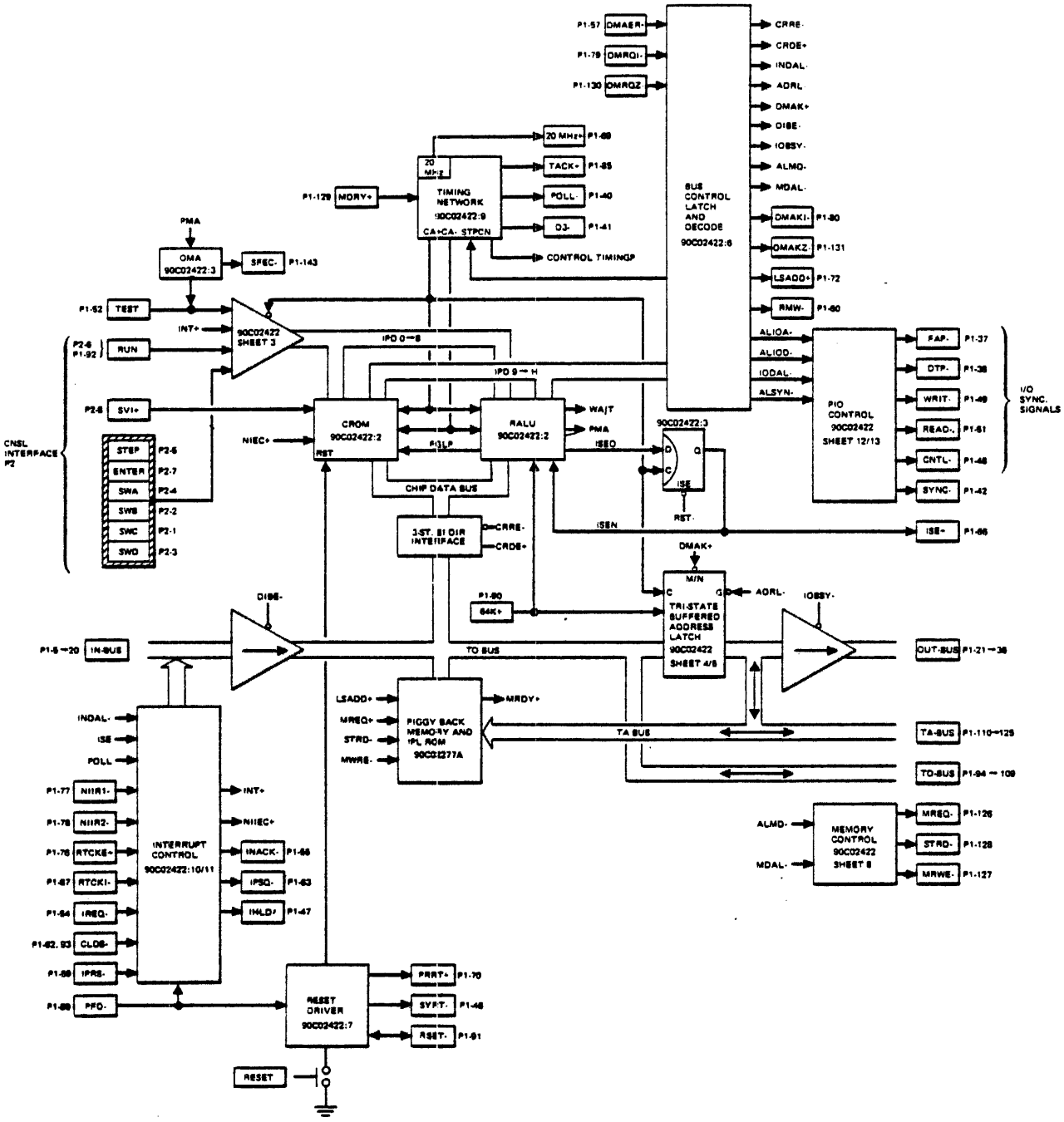
The purpose of this section is to present a basic functional and physical description of both the GA-16/110 and GA-16/220 computer systems.

2.1 GA-16/110 CPU (CPU-1 MODULE)

The GA-16/110 processor (also the heart of the GA-16/220 system) is a high-speed, LSI microprocessor-controlled, CPU on a single board. The nucleus of the central processor is a micro-computer, consisting of two proprietary, N-channel, silicone gate, MOS chips; the Register Arithmetic and Logical Unit (RALU) and the Control Read-Only-Memory (CROM). The RALU, in effect, duplicates the internal organization of the SPC-16 and GA-Series 16 processors. The CROM contains the microcoded instructions and logic necessary to emulate an extended SPC-16 instruction set.

The detailed block diagram of the GA-16/110 CPU module, Figure 2-1, includes key interface signals. The major blocks of the GA-16/110 (CPU-1) module are listed below. (For additional information, refer to the GA-16/110/220 Systems Reference Manual, Section 2.)

- Control Read Only Memory
- Register Arithmetic and Logical Unit
- Initial Program Decode Bus
- Timing Network
- Bus Control Latch and Decode
- Programmed Input/Output Control Network
- Interrupt System Enable Flip Flop
- Tri-State Buffered Address Latch
- Piggyback Memory and IPL ROM
- Tri-State Bi-Directional Interface
- Tri-State Data Bus
- Tri-State Address Bus
- Interrupt Control
- Reset Driver
- In-Bus
- Console Interface
- Run, Test, SVI



509-2-1

Figure 2-1. GA-16/110 CPU Module Block Diagram

2.1.1 CONTROL READ-ONLY-MEMORY (CROM)

The CROM contains the microcode and logic necessary to emulate the expanded SPC-16 instruction set. The microinstructions utilized in the GA-16/110/220 are 34 bits in length and control all signal generation and data transfers within the system.

The functional components of the CROM (reference Figure 2-2) are:

- Program Instruction Register and Control. The "I" register contains the program (user) instruction now being executed. The I register is loaded, via the M-Bus, during instruction fetch (IFETCH) time.

- Programmable Logic Array (PLA). The output of the I register (internal to the CROM) feeds the PLA and selects the correct address of the first instruction of the microprogram (within the CROM) to be executed for proper translation and operation of the current program instruction in I.

- ROM Address Latch. This logic receives the output of the PLA and latches the address of the microinstruction in the CROM.

- ROM. The actual microinstruction read-only-memory which contains all microinstruction sequences (320 words by 34 bits).

- Tri-state drivers. The Schottky (tri-state) bus drivers output, at the proper time, the microinstruction, to the rest of the CPU logic.

- IPDx Bus. The Initial Program Decode Bus is used to carry the output of the ROM (microinstruction) to the RALU and remainder of the system logic.

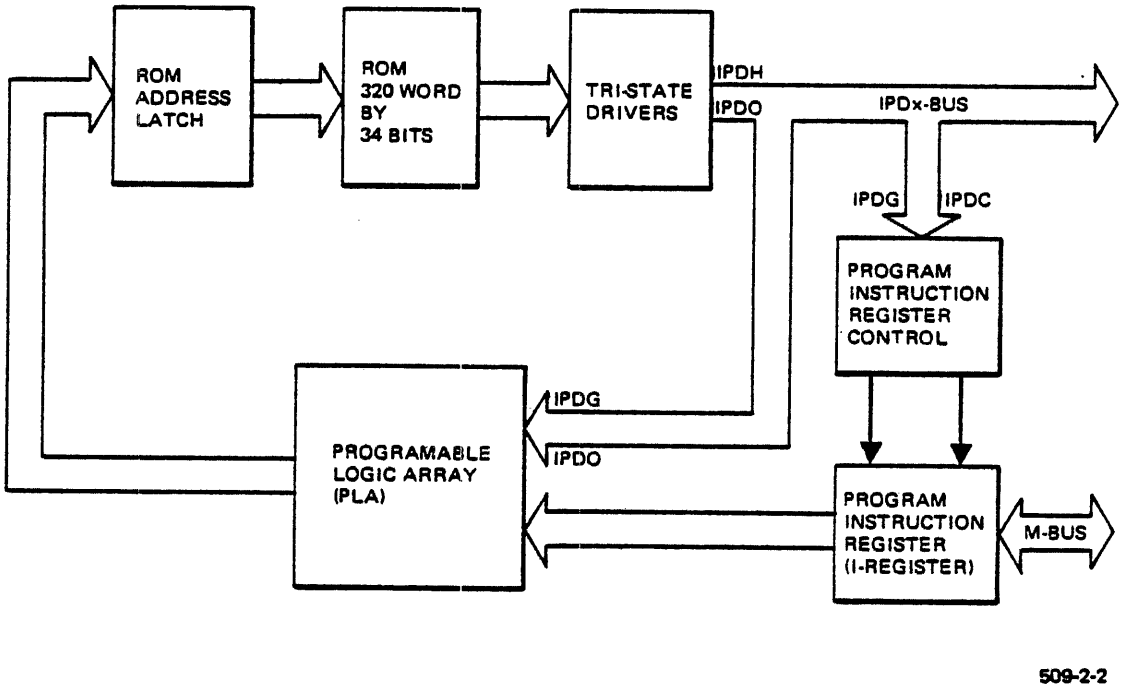


Figure 2-2. CROM Block Diagram

2.1.2 REGISTER ARITHMETIC AND LOGIC UNIT (RALU)

The RALU (reference Figure 2-3) is the computational nucleus of the GA-16/110 and GA-16/220. The RALU performs all arithmetic and logical operations available to the system, as well as the majority of the general-purpose and working registers. The main functional components within the RALU are:

- Microinstruction I-Latch. The register for the microinstruction currently being translated from the CROM. The microinstruction is gated, via the IPDx Bus, from the CROM.

- Function Control Logic. This logic decodes the control functions of the microinstruction to select such things as, source register (RS), destination register (RD) adder inputs/outputs, etc.

- The general-purpose registers are P, W, S, and A through E.

- The Adder is the logic for computation of arithmetic and logical operations.

- Adder gating. The E-Latch, D-Latch and Spread circuitry are used in controlling the selection and conditioning of data to be input to the adder circuitry. (The Spread circuitry is used for manipulation of the data from the W register during address calculations.)

- The M-Bus is the bi-directional memory bus for transferring data between the CROM/RALU and the rest of the processor.

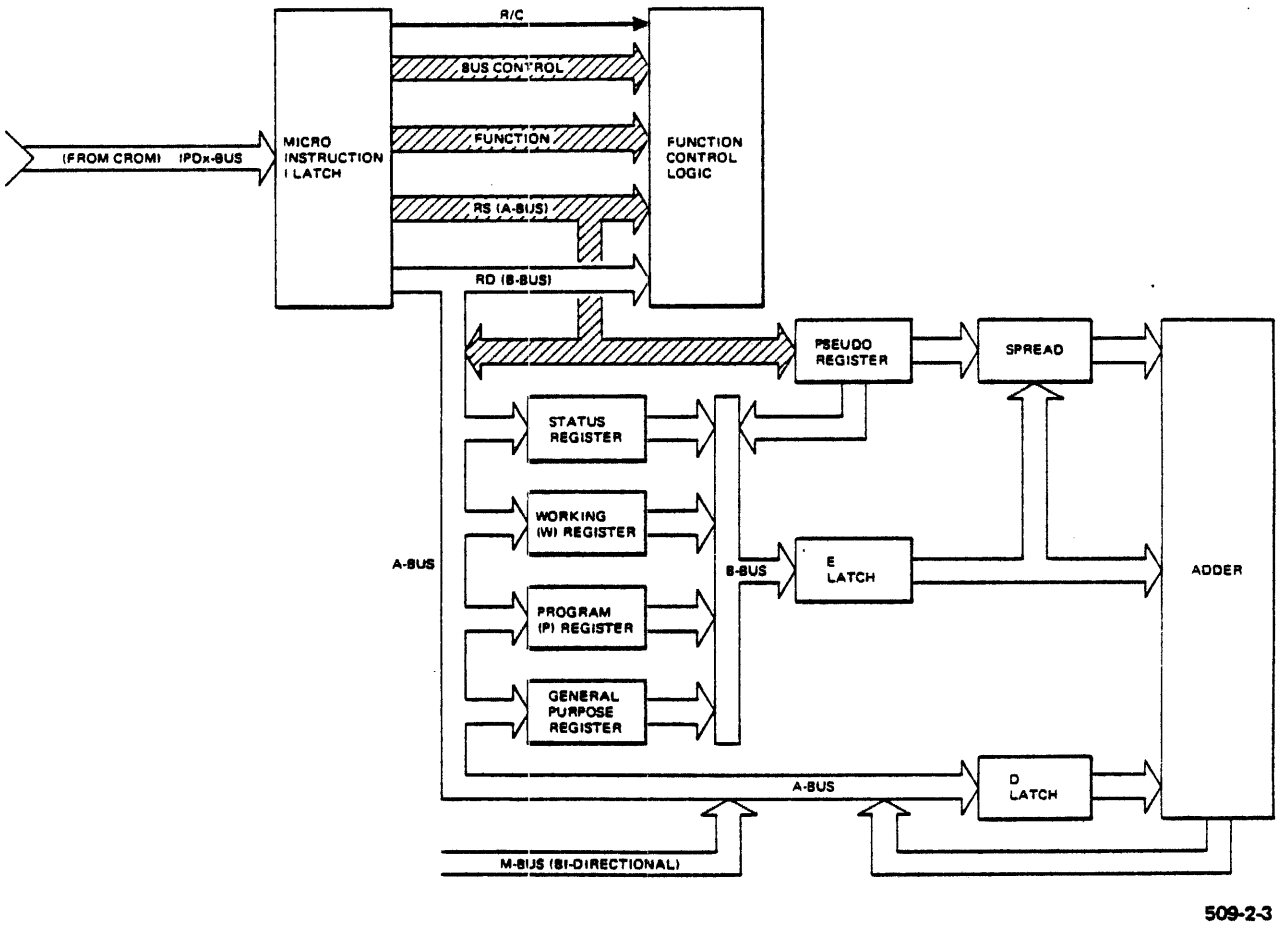


Figure 2-3. RALU Block Diagram

2.1.3 INITIAL PROGRAM DECODE BUS

The IPDX-Bus is the primary path for control information between the CROM and the RALU. This bus also serves as the path for transmitting control signals to other logic within the CPU. (These signals maintain control of the gating of data over the CPU bus network.)

2.1.4 TIMING NETWORK

All processor timing is derived from a 20MHz crystal-controlled oscillator, the heart of the timing network. The timing network outputs the timing pulses CA+ (the processor 'A' clock) and CA- (the processor 'B' clock) to provide timing for the CROM and RALU chipset. The 20MHz+ timing signal is output, via P1 pin 69, to the CPU-2 module (when installed) for GA-16/220 internal timing. TACK+ is the buffered processor 'A' clock that is used in the microcycle timing. TACK+ has a frequency equal to one cycle per processor microcycle. POLL- and D3- are the I/O interface synchronization timing signals.

2.1.5 BUS CONTROL LATCH AND DECODE

The BUS Control Latch and Decode circuitry provides the logic network for decoding the bus control (B-CNTL) field of the CROM microinstruction. The input/output signals are those signals as required for proper microcode operation and are listed below.

- DMRQ1/DMRQ2 — These signals are requests for memory access from DMA channels 1 and 2.
- DMAK1/DMAK2 — The DMAK signals are the CPU DMA acknowledge signals generated in response to DMRQ1 or DMRQ2.
- CRRE — The CROM/RALU Receive Enable signal. This signal is used to gate data into the CROM and RALU logic chips from sources external to the CROM/RALU chipset.
- CRDE — The CROM/RALU Drive Enable signal. This signal enables the output of the CROM or RALU to be gated to the logic outside of the CROM/RALU chipset.
- INDAL — Interrupt Data to RALU. The INDAL signal is generated by the microcode to gate interrupt data (interrupt vectors, etc.) into the RALU.
- ADRL — Address Load Enable. ADRL selects one of two data sources as the output to the TA-Bus (TAXX) lines. The data source may be either the TD lines or the CROM/RALU Data-Bus lines (M0 through MF).
- DMAK — DMA Acknowledge. DMAK is a general acknowledge signal generated anytime the processor is acknowledging either a DMRQ1 or DMRQ2.
- DIBE — Data In-Bus Enable. When active, this control signal, gates the contents of the Tri-State Data-Bus (TDXX) to the CROM/RALU via the CROM/RALU Data Bus lines (M0 through MF).

- IOBSY — I/O Busy. IOBSY indicates that the current microinstruction is performing an input/output operation. This signal enables the Tri-State Address Lines output (TAXX) to be gated to the Out-Bus lines to address the correct peripheral controller.

- ALIOA — RALU Output to I/O Address. Whenever an I/O operation is requested, ALIOA is generated by the microcode to produce IOBSY (see above).

- ALIOD — RALU to I/O Data. After the microcycle used to address a peripheral device (ALIOA), ALIOD is generated to gate the actual data from the processor to the designated peripheral controller.

- IODAL — I/O Data to RALU. This signal is generated during an I/O input microcycle to gate the data, from the I/O device, into the RALU registers.

- ALSYN — RALU to Sync. The sync output generates a timing strobe to allow synchronization of an oscilliscope with the CPU logic timing.

- ALMD — RALU Output to Memory Data. When the program requires the gating of the output of the RALU to memory, ALMD goes true to enable that output, on the chip data bus (M-Bus), to memory via the tri-state data bus (TD-Bus).

- MDAL — Memory Data to RALU. The microcode forces MDAL true to indicate a transfer of a memory word into the RALU. Memory data is transferred via the TD-Bus and M-Bus.

- LSADD — Load Starting Address. During start-up procedures, such as IPL, LSADD combines with the presence (or absence) of the IPL signal to select the proper starting ROM address for the IPL operation.

- RMW — Read/Modify/Write. RMW indicates that the data being read from memory is to be modified and re-written into the same memory location.

2.1.6 PIO CONTROL

The Programmed input/output control network generates the signals necessary to govern operation of the I/O bus. It is the responsibility of this logic to maintain I/O timing compatability with General Automation peripheral controllers.

The timing and gating pulses output from the PIO control network are as follows.

- FAP — Function and Address Pulse. FAP is utilized as an enable to select (address) a specific controller. FAP identifies the contents of the Out-Bus (bits 5 → 0) as the device select address.
- DTP — Data Transfer Pulse. DTP is used to synchronize the actual transfer of data between the CPU and peripheral. This signal indicates that either data input from a peripheral controller is available for the CPU on the In-Bus lines or data output to a peripheral controller is available on the Out-Bus lines.
- WRIT — The write signal is sent to the peripheral device to indicate that the current operation is a data output from the CPU.

- READ — The read signal is generated by the CPU, in response to a data input request, it indicates that a data input to the CPU operation is in progress.

- CNTL — Control indicates, to a peripheral controller, that bits 8, 9 and 10 of the Out-Bus contain a bit combination that is to be decoded as one of eight possible control commands.

- SYNC — The CPU Sync pulse is generated to allow an oscilliscope to be synchronized with the CPU timing cycles. This signal is available on P1 pin 42.

2.1.7 THE ISE FLIP-FLOP

The Interrupt System Enable Flip-Flop allows a programmable enable/disable of all CPU inhibitable interrupts. The ISE has no effect on non-inhibitable interrupts, such as Power Fail Detect (PFD).

2.1.8 TRI-STATE BUFFERED ADDRESS LATCH

The Tri-State Buffered Address Latch is comprised of a series of type 74173 (typical) logic gates and is mechanized to transfer the output of the CROM to the processor TAXX lines when addressing CPU memory. The address latch will also gate the CROM output onto Out-Bus, if IOBSY- is true, thereby indicating that an I/O operation is currently in progress.

2.1.9 PIGGYBACK MEMORY AND IPL ROM

The Piggyback Memory may consist of either static RAM or EPROM combined with RAM. The static RAM piggyback contains 2K of RAM with an optional 64 word, single device, IPL ROM. The IPL may be either teletype (TTY) or high-speed paper tape reader (HSPTR).

The alternate configuration for the piggyback memory is 3K of EPROM combined with 1K of RAM.

In the 2K piggyback RAM, the IPL ROM is additional memory space and does not replace any of the 2K addresses in RAM. The signals utilized by the piggyback memory are:

- MREQ — Memory Request: The memory request signal notifies the memory that a memory cycle is pending.

- STRD — Stop Memory Read. This signal is used to end a memory read cycle. STRD partially enables the memory write operation when an RALU to memory microcycle (RALU → MD) is active.

- MDRY — Memory Data Ready. The MDRY signal is transmitted from the memory module to acknowledge a memory request (MREQ). The trailing edge indicates valid data on the memory output lines.

- MWRE — Memory Write Enable. At the end of a memory read cycle MWRE is generated to enable the gating of data to the addressed memory location. MWRE is generated only if an RALU → MD microcycle is active.

2.1.10 TRI-STATE BI-DIRECTIONAL INTERFACE

Type N8T26B logic chips comprise the interface between the CROM/RALU chipset and the bi-directional Tri-State Data Bus (TD-Bus).

Two signals control gating through this interface: CRDE+ and CRRE- (reference Section 2.1.5, Bus Control Latch and Decode).

2.1.11 BUS NETWORKS

Four buses are used for communication between the CPU-1 module (16/110) and the other system modules: In-Bus, Out-Bus, TA-Bus and TD-Bus.

- In-Bus — In-Bus contains the 16 data lines for data input to the CPU-1 module from the CPU-2 module (16/220 configuration) and the peripheral controllers. The In-Bus lines are also used as the interrupt control access path. Data is enabled over these lines by the signal DIBE (Data In-Bus Enable).

- Out-Bus — The 16 Out-Bus lines are used as the data path for output from the CPU-1 module to the system I/O controllers. Out-Bus is also available to the CPU-2 module. Out-Bus Gating is enabled during I/O operations only (IOBSY-)true.

- TD-Bus — Tri-State Data Bus. The TD-Bus is the bi-directional communications lines for the transfer of data into and/or out of the CROM/RALU chipset and memory TD-Bus gating consists of tri-state Schottky logic chips.

- TA-Bus-Tri-State Address Bus. The TA-Bus is the transfer path for memory addressing information from the CROM/RALU chipset. Like the TD-Bus, the TA-Bus gating consists of the tri-state Schottky logic chips.

2.1.12 INTERRUPT CONTROL

The Interrupt Control network contains the logic for the recognition, programmable masking and generation of all interrupt signals available within the CPU-1 module and its associated peripheral controllers. The interrupt signals received and generated by this logic are listed below.

- ISE — Interrupt Systems Enable. The output of the ISE flip-flop, when set, enables the recognition of the inhibitable interrupts. This setting of the ISE flip-flop is under program control.

- POLL — The POLL signal is a timing signal used for synchronization of I/O and interrupt signaling.

- NIIR1 — Non-Inhibitable Interrupt Number 1. NIIR1 goes true to indicate that a non-inhibitable interrupt is generated on MPP.

- NIIR2 — Non-Inhibitable Interrupt Number 2. NIIR2 is driven true when a non-inhibitable interrupt is generated on a GA-16/220 system, single step or break interrupt.

- RTCKE — Real-Time Clock Enable. The RTCKE line is the programmable enable input for the Real-Time Clock (RTC).

- RTCKI — Real-Time Clock (RTC). On the GA-16/110, this input is available for a user supplied interrupt clock pulse input. On the GA-16/220 the interrupt clock pulse is derived from the 20 MHz crystal oscillator. The GA-16/220 RTC interrupt pulse occurs each millisecond (1ms).

- IREQ — Interrupt Request. Each peripheral device drives this input line active (low) when requesting interrupt access of the CPU.

- CLDS — Cold Start. System Startup and IPL procedures are dependant upon the condition (high or low) of this line. CLDS combines with other conditions to determine if start-up (or IPL) will be from ROM, RAM or manual intervention (refer to GA-16/110/220 Systems Reference Manual, Section 3.4.1.1, System Start-up).

- PFD — Power Fail Detect. The PFD circuit monitors the AC input line. If the AC input drops out of tolerance (below 105vac) the PFD circuit enables this line.

- IPRS — Interrupt Priority Return Status. A common signal, from the peripheral controllers, to indicate that interrupt priority has been established. (From the highest priority controller requesting an interrupt.)

- NIIEC — Non-Inhibitable Interrupt Mode Control. Any of the non-inhibitable interrupts going active forces NIIEC+ true (high). This forces the microcode to an interrupt microinstruction and disables any IPL or console ROM (16/220) operation.

- INT — Interrupt. The interrupt signal (INT) is generated, when an interrupt request is received by the CPU. INT is produced by a non-inhibitable interrupt or by an inhibitable interrupt with the interrupt system enabled (ISE+).

- IHLD — Interrupt Hold. Once an interrupt has been recognized and granted priority, by the CPU, IHLD is driven true (low) to disable any further interrupts.

- IPSO — Interrupt Priority Status Out. This is a serially transmitted priority signal propagated from each I/O controller to the next lowest priority controller. This signal is received via the Interrupt Priority Status In (IPSI) line.

- INACK — Interrupt Acknowledge. (Same as IACK.) The highest priority interrupting controller uses this timing signal to place its vector address on the In-Bus lines.

- INDAL — Interrupt Data To RALU. When the interrupt data is available to the chipset, INDAL is generated by the microcode to gate that data to the RALU.

2.1.13 RESET DRIVER

The Reset Driver circuitry provides the logic to generate a processor reset (PRRT) to the CPU only (REST line) or a systems reset (SYRT) to the I/O controllers as well. The term RSET- is representative of a bi-directional line, activated via a console pushbutton, that causes a reset of the CPU only.

2.1.14 CONSOLE INTERFACE

The Console Interface is a 16-pin DIP socket on the outward edge of the CPU-1 module. This connector is designed to allow the application of a user-designed console device to monitor and/or drive the following processor lines:

- RUN — The run line is a dual-purpose, bi-directional line. When tied (via P2 pin 6), to an indicator light network, RUN indicates the operational status of the CPU. The run line may also be connected to a console switch, thereby allowing a run command to be forced into the CROM/RALU chipset by driving RUN+ true (high), and forcing step true.
- SVI - Save I. The Save I line, when placed into its' true state (low), forces the CROM to continually execute the instruction in the instruction (I) register. Any change to the I register is blocked.
- STEP — The step line allows for single instruction execution. With the processor in the idle mode (RUN) and this line true (high) one complete instruction execution cycle takes place, if RUN is forced true for one micro/cycle.

  If the system has been placed in the RUN mode (via P2 Pin 6) activating STEP initiates automatic program execution.
- ENTER. Activation of the ENTER line gates the contents of user provided console switches (tied to the In-Bus lines 0 through 15) into the CROM/RALU chipset.
- SWA, SWB, SWC, SWD — The four "general-purpose register switches" are available to enable selection of one of eight general-purpose registers to receive the data set into the user-supplied data switches (gated in by ENTER). The processor must be in the idle condition.

2.1.15 TEST

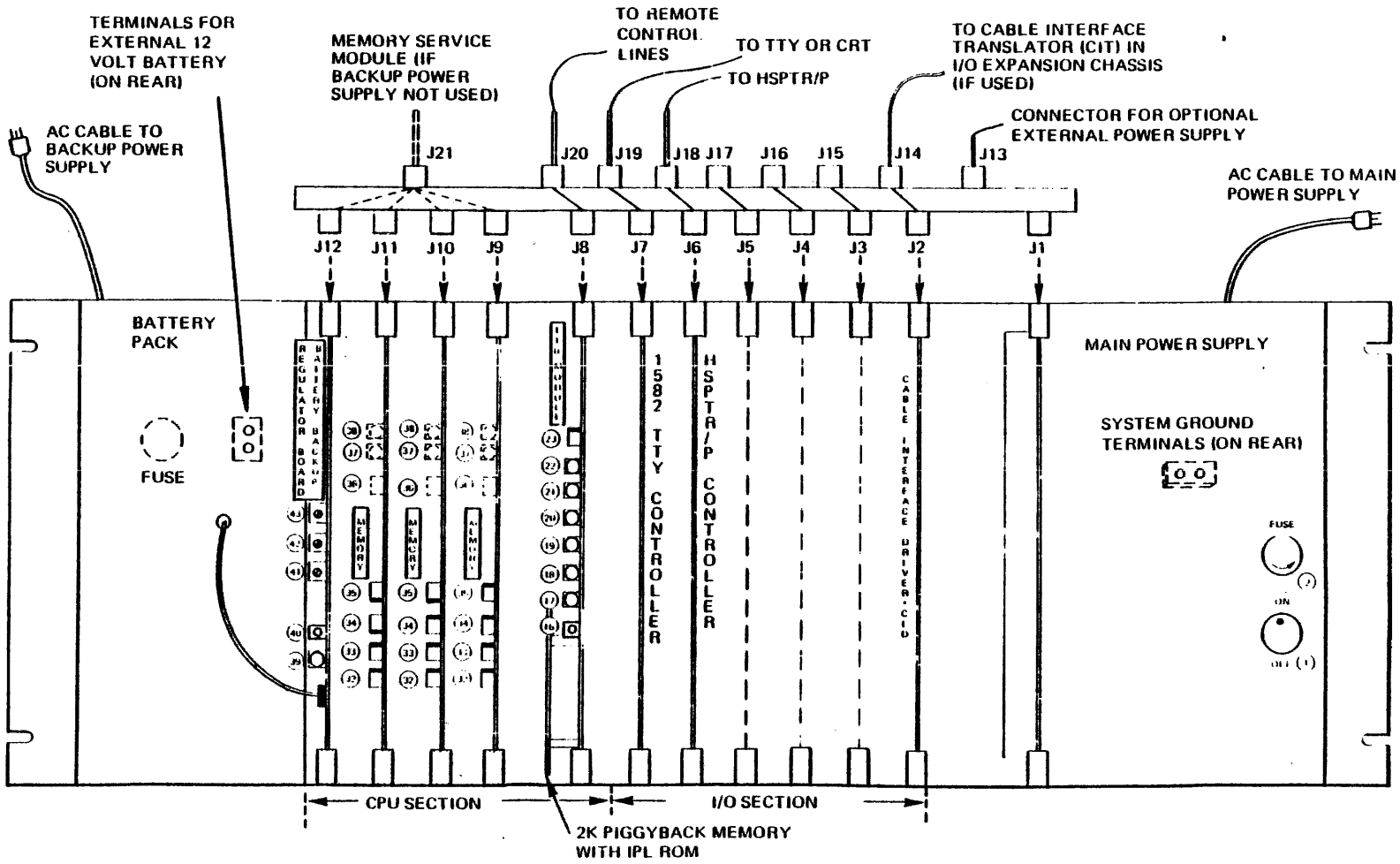
The TEST line returns the indication of the results of a Test instruction to the processor. In actuality, TEST may be active anytime an I/O operation is in progress. However, the processor only samples the TEST line when a TEST instruction is being executed.

2.1.16 OPERATIONS MONITOR ALARM (OMA)

The OMA network provides for an automatic system shutdown if the users program fails to periodically execute a PMA instruction. Once armed (by a PMA instruction) the OMA timer allows 150 ms to 300 ms for the system program to issue another PMA instruction to re-arm the timer. If, within this time frame, the timer is not re-armed, the system switches from the RUN mode to the IDLE mode and forces the systems safe signal (SFEC) false (high). SFEC being high indicates that the system is in an "unsafe" condition.

2.1.17 PHYSICAL CONFIGURATION

In order to aid in component placement, Figures 2-4 and 2-5 are supplied with corresponding Tables 2-1 and 2-2. If additional information is required concerning these items, refer to the GA-16/110/220 Systems Reference Manual, Section 3.



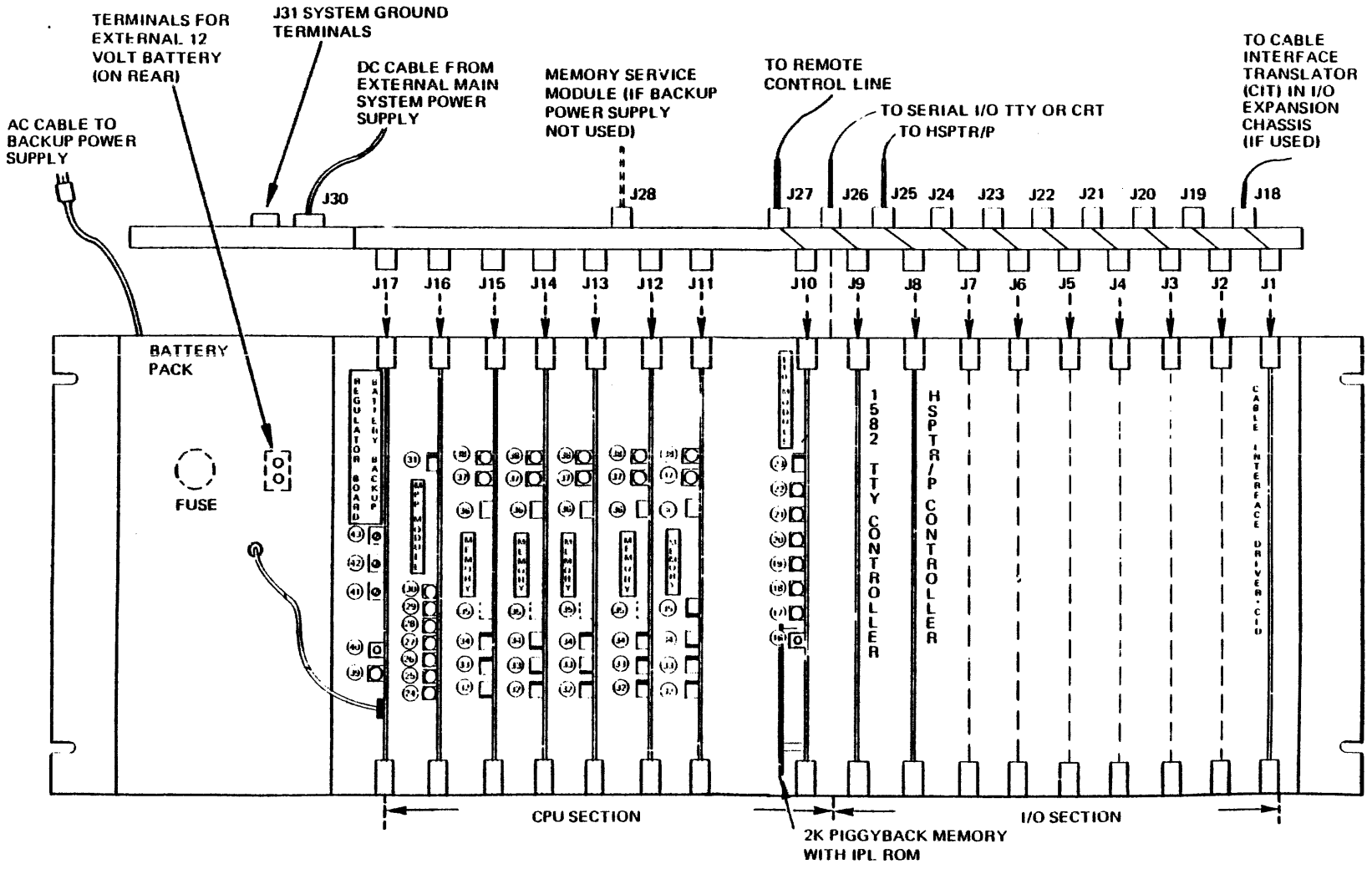
*CIRCLED NUMBERS REFER TO TABLE 2-5.

508-F-1

Figure 2-4. GA-16/110 System in Compact Chassis (No MPP)

Table 2-1. Connector Assignments for Compact Chassis GA-16/110

Front/Module (110)		Rear/Interface (110)	
J1	Main power supply	-	AC input connector and system grounding terminals on power supply.
J2	Cable interface driver (CID)		
J3 J4	I/O controllers or shorting boards.	J13	Connector for optional external power supply cable.
J5 J6 J7	Shorting boards are used in empty slots between controllers or CID to maintain priority chain continuity.	J14	Cable to I/O expansion chassis (6-feet maximum).
		J15 J16 J17 J18 J19	Paddleboards and cables interfacing controllers (in J3 through J7) to peripheral devices.
J8	CPU-1 module with optional piggyback memory and IPL ROM.		
J9 J10 J11	Memory module or Memory Parity Protect module.	J20	Access to cold-start (CLDS-) memory mode (64KM+) and remote control lines (IPLSW-, PFD-, RSET-, RUN-).
J12	J12 also used for battery backup regulator board.	J21	Memory Service Module if backup power is not used.
-	Backup power supply and battery pack (use regulator board in J12).	-	AC cord for backup power supply.
		-	Terminals for external 12-volt battery on rear of battery pack.



CIRCLED NUMBERS REFER TO TABLE 2-5.

508-F-3

Figure 2-5. GA-16/110 System in Jumbo Chassis (With MPP)

Table 2-2. Connector Assignments for Jumbo Chassis GA-16/110

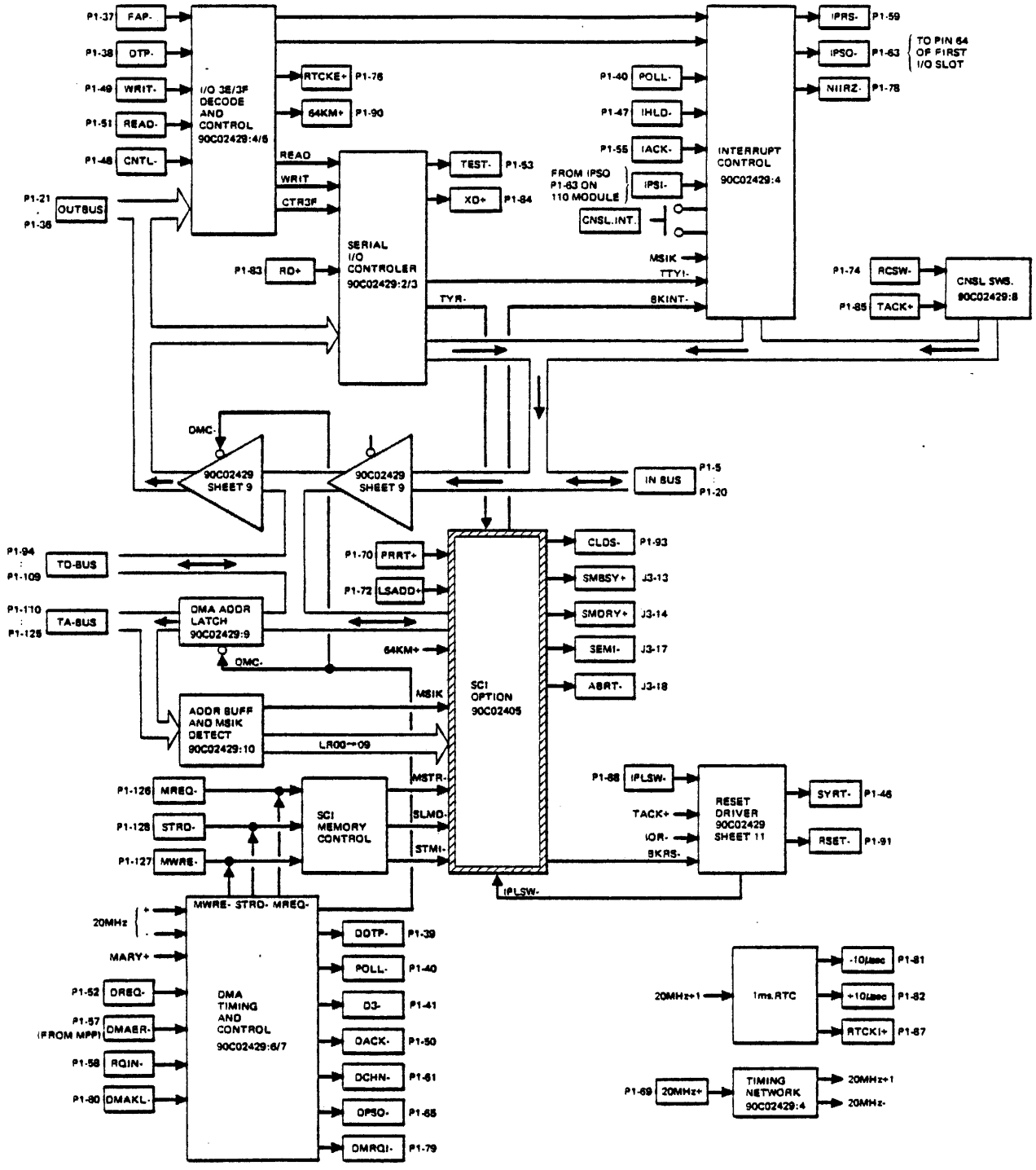
Front/Module (110)		Rear/Interface (110)	
J1	Cable interface driver (CID)	J18	Cable to I/O expansion chassis, (6-feet maximum).
J2	I/O controllers or shorting boards.	J19	Paddleboards and cables interfacing I/O controllers (in J2 thru J10) to peripheral devices.
J3		J20	
J4		J21	
J5	Shorting boards are used in empty slots between controllers or CID to maintain interrupt chain continuity.	J22	
J6		J23	
J7		J24	
J8		J25	
J9		J26	
J10	CPU-1 module with optional piggyback memory and IPL ROM.	J27	Access to cold-start (CLDS-) memory mode (64KM+), and remote control lines (IPLSW-, PFD-, RSET-, RUN-).
J11	Memory modules or memory parity protect module. J17 also used for battery backup regulator board.	J28	Memory Service Module if backup power supply is not used.
J12		J30	Power cable from external power supply.
J13		J31	System grounding terminals.
J14		-	AC cord for backup power supply.
J15		-	Terminals for external 12-volt battery on rear of battery pack.
J16			
J17			
-	Backup power supply battery pack (use regulator board in J17).		

2.2 GA-16/220 (CPU-2 MODULE)

The GA-16/220 processor is structured around the basic GA-16/110 (CPU-1 module) and one additional processor printed circuit board, the CPU-2 module. The CPU-2 module adds the DMA communications port, the Serial I/O controller (CRT/TTY interface), a one millisecond Real Time Clock (RTC) and additional controls and indicators.

The detailed block diagram of the CPU-2 module (Figure 2-6) shows the features added to the GA-16/110 configuration by the addition of the CPU-2 module. The structure of the CPU-2 module is defined below. (For additional information, refer to the GA-16/110/220 Systems Reference Manual, Section 2.)

- 1ms Real-Time Clock
- Interrupt Control Logic
- Console Switches
- DMA Timing and Control
- Systems Console Interface
  - Reset Driver
  - SCI Memory Control
- Address Buffer and MS1K Detect
- DMA Address Latch
- Serial I/O Controller
- I/O 3E/3F Decode and Control
- Timing Network



509-2-4

Figure 2-6. 220 (CPU-2) Module Block Diagram

2.2.1 REAL-TIME CLOCK

The one millisecond Real-Time Clock (RTC) pulse generates an interrupt (RTCKI+) each millisecond only when enabled by program command. If the RTC interrupt is disabled, the RTCKI+ signal is blocked from having any effect. The RTC derives its timing from the 20 MHz oscillator on the CPU-1 module.

2.2.2 INTERRUPT CONTROL LOGIC

The Interrupt Control Logic masks and monitors all interrupt generating conditions. If an interrupt occurs, and is programmatically enabled, the interrupt control logic allows generation of the interrupt via the I/O Bus.

The signals POLL, IHLD, IACK, IPRS, IPSO, and IPSI are I/O network synchronizations and timing pulses and are described in Section 2.1.12, Interrupt Control (GA-16/110 Module description).

The primary GA-16/220 interrupt signals are:

- MSIK — The Most Significant 1K signal indicates that the current memory request is an address in the upper 1K range and, therefore, is accessing the Systems Console Interface (SCI) ROM.
- NIIR2 — See Section 2.1.12.
- CNSL INT — The Console Interrupt condition is produced by the manual depression of the Console Interrupt pushbutton on the CPU-2 module.
- CIEN/TYIEN — The Console Interrupt Enable (CIEN) and Teletype Interrupt Enable (TYIEN) lines signal the translation of the instructions that programmatically enable the interrupt network to detect and propagate the Console Interrupt and TTY Interrupt, respectively, to the CPU-1 module.

2.2.3 CONSOLE SWITCHES

The Console Switches are the 16 miniature, rocker-type, switches mounted on the microconsole. These switches serve as a systems console for manual data input to the GA-16/220 system. Data, input via the console switches, are in binary format and are passed over the In-Bus lines.

Only two signals are utilized by the Console Switch network; TACK+ and RCSW-. TACK+ is the buffered processor 'A' clock (CA+) and is used to mark the correct timing for input of the data contained in the switches. RCSW- is the strobe pulse to enable the gating of the data, from the switches, onto In-Bus.

2.2.4 DMA TIMING AND CONTROL

The Direct Memory Access Timing and Control logic controls all data channel input/output (DCIO), from the Multiple High-Speed Data Channel Controller (MHSDC) over the I/O Bus. The DMA logic also oversees the direct DMA accesses from those controllers incorporating MHSDC-type logic and, therefore, not connected to an MHSDC module for memory access.

All signals, except POLL and D3, being produced or used by the DMA Control are listed below. POLL and D3 are the standard I/O synchronization clock pulses. (Refer to GA-110/220 Systems Reference Manual, Table 6-19, I/O Bus Description).

- DMRQ1 — Direct Memory Request 1. DMRQ1 is forced true whenever a Direct Memory Access Request (DREQ) is received from the data channel.
- DPSO — Data Channel Priority Status Out. Just as the priority of the interrupting controllers is determined by IPSO/IPSI, so the priority of the data channel controllers is maintained by DPSO/DPSI.
- DACK — DMA Acknowledge. Upon recognition and acceptance of a DMA request from a peripheral (or the 220) the CPU-1 generates DACK to acknowledge that interrupt.
- DDTP — DMA Data Transfer Pulse. For standard PIO data outputs, the processor transmits DTP to signal valid data on Out-Bus and available to the controller. In DMA transfers, the same function is accomplished by DDTP.
- DCHN-DMA Chain. The Data Channel Module (or similar logic on a non-MHSDC-DMA controller) produces the DCHN signal to indicate either a "chaining" mode or "initiate" mode in a data channel controller.
- DMC - Direct Memory Cycle. When a DMA request (DREQ) is received by the CPU-2, DMC is generated to indicate that a Direct Memory Cycle is active.
- RQIN - Request Input. RQIN indicates the direction of data flow in a DMA operation. RQIN being true indicates a data input (READ) operation.
- DMAK1 - DMA #1 Acknowledge. The CPU-1 generates DMAK1 to acknowledge a DMA request from the CPU-2 module.
- DREQ —DMA Request. When a data channel controller requires service, DREQ is made low to signal the CPU-2 that a DMA device is seeking access to memory.
- DMAER - DMA Error. With the optional Memory Parity Protect (MPP) option installed, this line indicates a DMA access error (attempt to write into protected memory) by going true (low), or parity error has occurred.

The signals MREQ, STRD, MWRE, and MDRY are memory control signals and are explained in Section 2.1.9.

2.2.5 SYSTEMS CONSOLE INTERFACE (SCI)

The optional Systems Console Interface (SCI) contains a 256-word, random-access-memory (RAM), and a 512-word, read-only-memory (ROM), to provide an interactive TTY/CRT interface utility program. The SCI may also contain a 256-word IPL ROM. Several controls and indicators are also a part of the SCI and are described in the GA-16/110/220 Systems Reference Manual in Sections 1 and 3.

The signals input to, and emanating from, the SCI option are described in the following text:

- 64KM — 64K Memory Mode. When in the 64K mode, this line is true to enable addressing of the upper 32K of memory without translating addresses in the 32K range as console ROM addresses (disables MS1K until 63K addresses).

- TYR — TTY Framing Error. If the Serial I/O controller detects an absence of a valid stop bit after a character, TYR goes true (low) to reset the controller (UART) and produce BKINT or BKRS if enabled.

- BKINT - Break Interrupt. When enabled, by the 2-position, SCI, BREAK switch (S2) this line indicates that the operator has pressed the TTY/CRT BREAK KEY (if installed on the device). This interrupt allows either a system reset or an interrupt depending upon the setting of the BRK MODE SCI switch.

- BKRS — Break Reset. If the BKINT condition is disabled, BKRS is held true (low) and has the same effect as pressing the reset button on the CPU-1 module.

- IPLSW - Initial Program Load Switch. Switch S4 on the microconsole (IPL), when pressed, drives this line true (low). When IPLSW is true, control is passed to the IPL ROM (on the SCI Module) to effect program loading from a specified peripheral device.

- LSADD — Load Start Address. On normal system start-up operation, LSADD combines with the IPL signal to select the proper ROM starting address for IPL operations.

- PRRT - Processor Reset. Either a power fail detect (PFD) or a manual pressing of the Reset Switch (S1), on the CPU-1 module, generates a reset to the SCI via this line.

- CLDS - Cold Start. During certain power-up sequences, the SCI forces this line true to aid in selection of the type of start-up procedure to be implemented.

- SEMI - Semiconductor Memory. When a memory request address translates as being in the upper 1K (MS1K) of memory, SEMI goes true to indicate that the requested address is in the Console ROM. (Combines with ABRT- to form MEMPR- on the CPU-2 module to disable the standard memory at that address.)

- ABRT - Abort. This signal is produced at the same time as SEMI and has the same definition.

- SMDRY - Select Memory Ready. The SMDRY signal indicates a memory read of the most significant 1K of memory (MS1K).

- SMBSY - Select Memory Busy. Memory busy indicates that a memory read operation is in progress. SMBSY and SMDRY combine, on the CPU-2 module to generate Memory Data Ready (MDRY).

- MSTR - Memory Start Read. When a memory request is active, MSTR is generated to partially enable the memory read cycle.

- STRD - Stop Memory Read. At the beginning of a write cycle STRD goes true to end the read cycle. STRD combines with MSTR to produce SLMD.

- SLMD - Select Memory Data. STRD combines with MSTR to produce SLMD. SLMD begins a read cycle in the most significant 1K of memory (MS1K) only (the console ROM).

- STM1 - Start Memory 1. A DMA request for memory access generates STM1. This signal is used in the GA-16/220 to help generate WRTS*.

2.2.6 ADDRESS BUFFER AND MS1K DETECT

The address buffer converts the memory address, currently referencing the SCI memory, from the processor TA-Bus to the SCI LR-Bus lines. The MS1K Detect logic identifies the current memory address as being in the most significant 1K of memory (the SCI ROM).

2.2.7 DMA ADDRESS LATCH

The Direct Memory Access Address Latch captures the address, from the In-Bus, of the memory location being accessed by a DMA controller. When the DMA request is recognized, and granted priority to access memory, the outputs of the address latch become the processor memory address lines (TA-Bus 00 through 15).

2.2.8 SERIAL I/O CONTROLLER

The internal Serial I/O Controller is a Universal Asynchronous Receiver/Transmitter (UART) logic chip. The UART based peripheral controller performs serial/parallel conversion of TTY/CRT input/output data. All communication with the teletype (or CRT) is accomplished through the Serial I/O Controller over the serial I/O bus via either the RS-232 (CRT) or TTY Interface Paddleboard.

Data to be output to the controller arrives from the CPU via the Out-Bus. Data received from the CRT/TTY is sent to the processor via the In-Bus. All other control and data lines are defined in the following text.

- RD - The Receive Data. All serial data from the CRT/TTY arrives via the RD communication line.

- XD - Transmit Data. After the 8-bit, parallel, data has been received, from the CPU, it is serialized and transmitted, via the XD line, to the TTY/CRT.

- TEST - The TEST line is the standard line, monitored by the processor, to indicate the results of an I/O Test instruction.

- TTYI - Teletype Interrupt. If a console - TTY interrupt is generated, indicating that the CRT/TTY is not busy, TTYI forces the TEST line true (low) in response to an I/O Test instruction to Device X'3F', if issued.

- TYR - Console TTY Framing error. Refer to Section 2.2.5, Systems Console Interface.

2.2.9 I/O 3E/3F DECODE AND CONTROL

The I/O 3E/3F Decode and Control network examines I/O instruction issued by the CPU to determine if that instruction is addressing the internal interrupt mask (I/O to X'3E') or the serial I/O controller (I/O to X'3F').

The signals FAP, DTP, WRIT, READ and control (CTRL) are the standard I/O interface timing and synchronization signals. If more information is required on these signals, refer to the GA-16/110/220 System Reference Manual, Section 6, Input/Output Operations.

All other signals concerning the I/O 3E/3F Decode and Control network are as follows:

- RTCKE - Real-Time Clock Enable. An I/O control (CTRL) command to device X'3E' with bit 3 of Out-Bus true (low) causes the Real-Time Clock Interrupt (RTCKI) to be enabled via the RTCKE line.

- TYIEN - TTY Interrupt Enable. An I/O control (CTRL) instruction to device X'3E' with bit 5 of Out-Bus true (low) enables the TTY/CRT to interrupt when not busy. This enable is propagated via the TYIEN line.

- CIEN — Console Interrupt Enable. In order for the console interrupt to be effective a control instruction (CTRL) must be issued to device X'3E' with Out-Bus bit 7 true (low). This drives CIEN true (low) and sets the console TTY (CRT) interrupt mask flip-flop.

- 64KM — 64K Memory Mode. This line is driven true as a result of the 32K-Prog Switch (S1) being placed in PROG position and a program command setting 64K mode. 64K mode is then forced when the processor is executing an RTIV instruction or any interrupt.

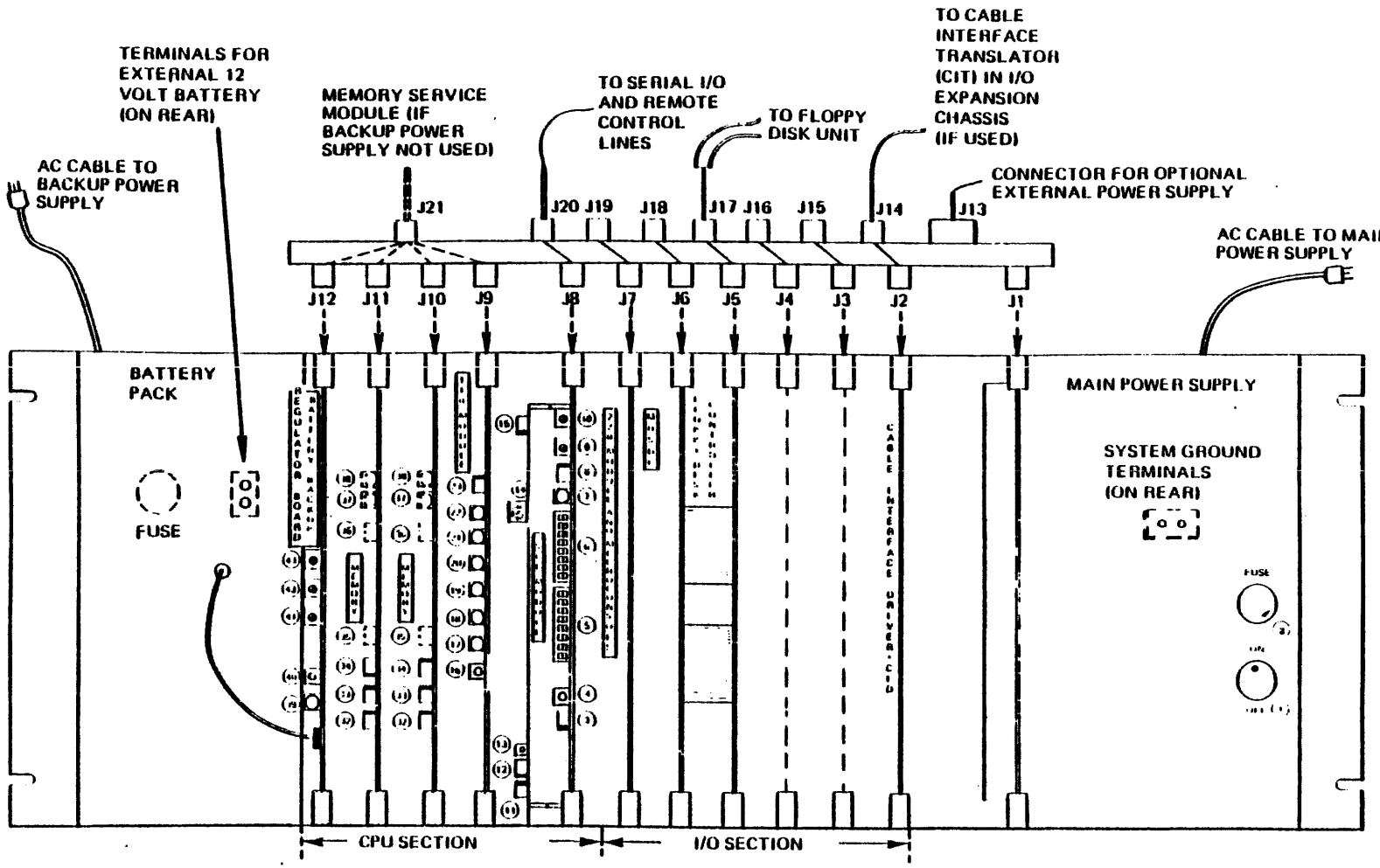
The I/O 3E/3F network also generates certain control signals to the Serial I/O Controller. These signals indicate that the information passing through the controller is: control data (CTR3F) to be maintained within the UART; data to be input to the CPU (READ); or data to be output to the TTY/CRT (WRIT).

2.2.10 TIMING NETWORK

The CPU-2 timing network derives two timing signals 20 MHz+1 and 20 MHz-, from the CPU-1 basic 20 MHz crystal oscillator. These two timing signals produce all timing necessary for the operation of the CPU-2 module.

2.2.11 PHYSICAL CONFIGURATION

In order to aid in component placement, Figures 2-7 and 2-8 are supplied with corresponding Tables 2-3 and 2-4. If additional information is required concerning these items, refer to the GA-16/110/220 Systems Reference Manual, Section 3, GA-16/220 Configuration and Operation.

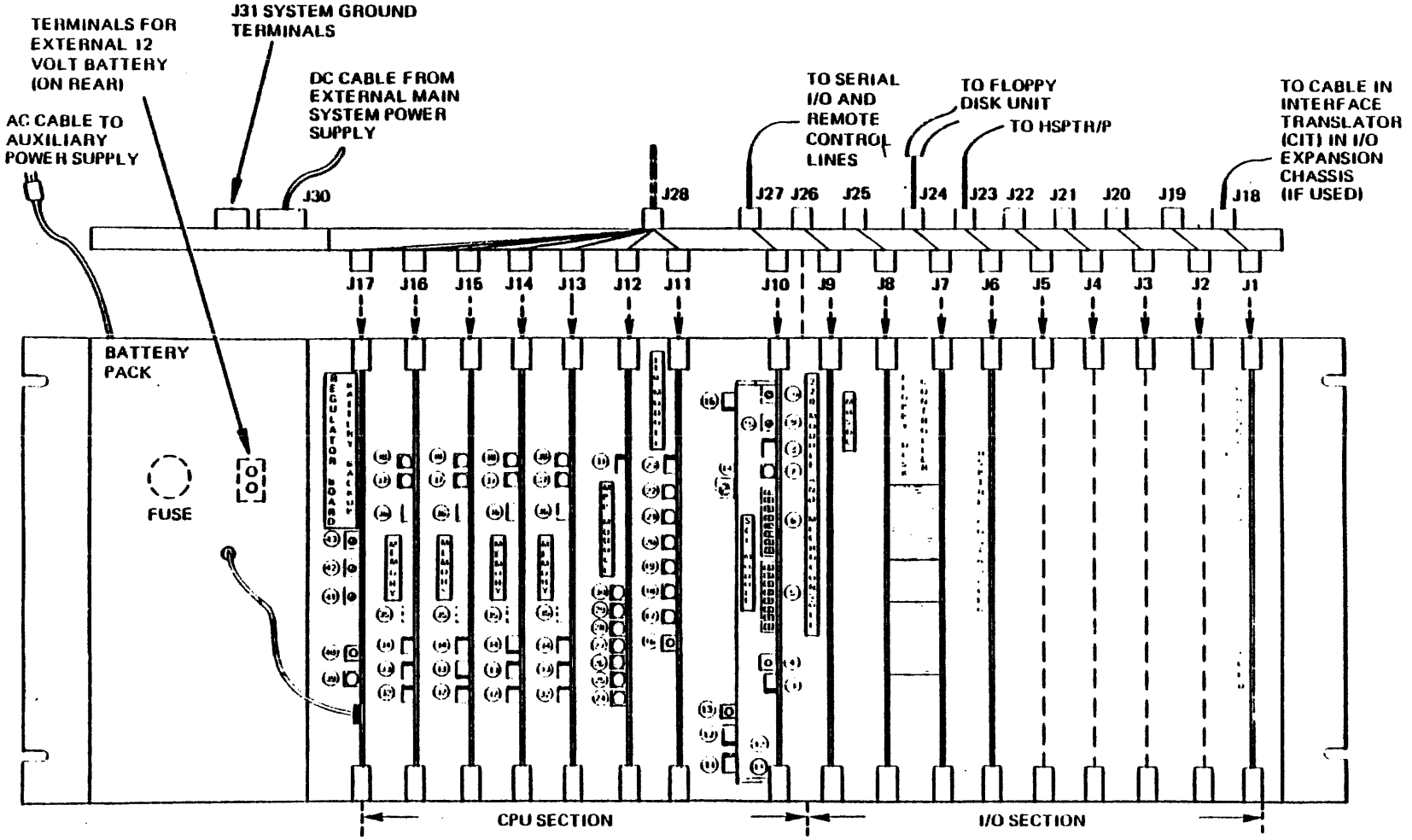


CIRCLED NUMBERS REFER TO TABLE 2-5.

Figure 2-7. GA-16/220 System in Compact Chassis (No MPP)

Table 2-3. Connector Assignments for Compact Chassis, GA-16/220

Front/Module (220)		Rear/Interface (220)	
J1	Main power supply	-	AC input connector and system grounding terminals on power supply.
J2	Cable interface driver (CID)		
J3	I/O controllers or shorting boards.	J13	Connector for optional external power supply cable.
J4			
J5	Shorting boards are used in	J14	Cable to I/O expansion chassis (6-feet maximum).
J6	empty slots between controllers or CID to maintain interrupt chain continuity.	J15	Paddleboards and cables interfacing controllers (in J3 through J7) to peripheral devices.
J7	MHSDC controller, if used, should go into J7.	J16	
		J17	
J8	CPU-2 module with optional System Console Interface.	J18	
		J19	
J9	CPU-1 module with optional piggyback memory.	J20	Serial I/O paddleboard with connector to TTY or CRT also provides access to cold-start (CLDS-) jumpers and remote control lines (IPLSW-, PFD-, RSET-, RUN-).
J10	Memory modules or		
J11	Memory Parity Protect module.		
J12	J12 also used for battery backup regulator board	J21	Memory Service Module, if auxiliary power supply is not used.
-	Backup power supply battery pack (use regulator board in J12).	-	AC cord for backup power supply.
		-	Terminals for external 12-volt battery on rear of battery pack.



*CIRCLED NUMBERS REFER TO TABLES 2-5.

Figure 2-8. GA-16/220 System in Jumbo Chassis (With MPP)

Table 2-4. Connector Assignments for Jumbo Chassis GA-16/220

Front/Module (220)		Rear/Interface (220)	
J1	Cable interface driver (CID).	J18	Cable to I/O expansion chassis (6-feet maximum).
J2	I/O controllers or shorting boards. Shorting boards are used in empty slots between controllers or CID to maintain interrupt chain continuity. MHSDC controller, if used, should go to J9.	J19	Paddleboards and cables interfacing I/O controller (in J1 through J9) to peripheral devices.
J3		J20	
J4		J21	
J5		J22	
J6		J23	
J7		J24	
J8		J25	
J9		J26	
J10	CPU-2 module with optional System Console Interface.	J27	Serial I/O paddleboard with connector for TTY or CRT. Also provides access to cold-start (CLDS-) jumper, and remote control lines (IPLSW-, PFD-, RSET-, RUN-).
J11	CPU-1 module with optional piggyback memory.		
J12	Memory modules or memory parity protect module. J18 also used for battery backup regulator board.	J28	Memory Service Module, if auxiliary power supply is not used.
J13		J30	Power cable from external power supply.
J14		J31	System grounding terminals.
J15		-	AC cord for backup power supply.
J16		-	Terminals for external 12-volt battery on rear of battery pack.
J17			
-	Backup power supply battery pack (use regulator board in J18).		

Table 2-5. GA-16/110/220 Controls and Indicators (Sheet 1 of 6)

Key	Label	Description and Function
(1) (PS)	ON OFF	The main power switch causes power to be applied to CPU and I/O controllers. Power is also applied to memory modules if memory service module is installed.
(2) (PS)	FUSE	The fuse holder allows replacement of main power supply fuse.
(3) (220)	32K-PGM	The 2-position switch selects memory addressing mode. When in the 32K position program addressing limit is 32K words. When in the PROGRAM position either 32K or 64K mode is selected by a program which sets the memory mode mask word and executes RTNIV (see Section 4.15.2.4 System Reference Manual).
(4) (220)	CNSL INT	Console Interrupt pushbutton, when pressed, causes a program interrupt through vector X'47' if (1) the console interrupt bit in the internal mask word is set (Section 4.15.2.1 Systems Reference Manual), and, (2) the interrupt system is enabled (refer to description of instructions INE and RISE in Section 4.13.2 and 4.10.9 Systems Reference Manual).
(5) (220)	15 8	These miniature binary switches are called the console switches.
(6) (220)	7 0	The console switches may be read by an RCSM or RCSR instruction in a program (Section 4.15.1 Systems Reference Manual). This capability permits writing programs which allow data entry and control via these switches. When a switch is in the position labeled OPEN, the bit is set to a 0.
(7) (220)	DMA ACK	This indicator may blink periodically when direct memory access is occurring. It may illuminate continuously if very high DMA activity is in progress or if there is a malfunction. In a GA-16/220 system DMA activity is normally controlled by the multiple high-speed data channel (MHSDC) controller.
(8) (220)	TTY BAUD*	The 2-position switch changes the baud rate of the built-in serial I/O controller to accommodate either a teletype (Model 33 automatic send receive, ASR, or equivalent) at 110 baud or a CRT terminal at 110 or 9600 baud.
(9) (220)	9600*	This screw adjustment permits fine adjustment of the 9600 baud transmission rate. It is adjusted only when the BAUD switch (8) is in the 9600 position.
(10) (220)	110*	This screw adjustment permits fine adjustment of the 110 baud transmission rate. It is adjusted only when the BAUD switch (8) is in the 110 position.

*These controls not present on CPU2 board 31D02574A; refer to Appendix G.

Table 2-5. GA-16/110/220 Controls and Indicators (Sheet 2 of 6)

Key	Label	Description and Function
(11) (220-SCI)	BKDS	The 2-position switch enables or disables the TTY break capability. When the switch is in the break-disable (BKDS) position, TTY breaks are ignored. When in the other position, TTY break is enabled so when a TTY operator presses the BREAK key on a teletype or CRT (provided unit is equipped with this key) either a processor (CPU) reset or an interrupt occurs depending on the setting of the BKINT switch (12).
(12) (220-SCI)	BKINT	The 2-position switch sets the break mode for TTY or CRT, provided BKDS switch (11) is set so that break mode is enabled. When BKINT switch is set to break-interrupt position (BKINT) a TTY break causes a non-inhibitable interrupt via vector location X'46' (provided CPU is not operating under the SCI Console ROM program or the IPL ROM). Registers and status are saved in locations in the SCI program RAM. When the switch is not in the BKINT position, a TTY break initiates a break reset which causes control to be transferred to the Console ROM and registers and status are not saved. I/O controllers are not reset.
(13) (220-SCI)	IPL	Pushbutton initiates initial program load (IPL). When pressed, control is passed to the IPL ROM installed on the SCI module, and program loading occurs from a peripheral device. Selection of the peripheral device is accomplished by setting the IPL SEL selector switch (14) to the appropriate position for the IPL device installed with the system. This function may be remotely controlled via IPLSW line, or at auto-restart with cold start line at ground.
(14) (220-SCI)	IPL SEL	The IPL selector switch is a 16-position rotary switch. The switch is used to preselect which I/O device will be the source of an IPL when the IPL switch (13) is pressed. (In systems illustrated in Figure 2-7 and 2-8, the device would be floppy disk or teletype.) Load-and-go operation is selectable for all devices, while load-and-stop operation may be selected for teletype, high-speed paper tape reader, or card reader. The switch positions and device selections are shown in Table 3-4.
(15) (220-SCI)	CNSL	The 2-position switch determines both the function of the CPU reset button (16) and an auto-restart operation when power is applied. When in the console position (CNSL) the CPU and I/O are reset and control is transferred to the Console ROM. If in the other position, the status of the cold start line, CLDS, determines the operations as described for RESET pushbutton on the CPU-1 module (16).

Table 2-5. GA-16/110/220 Controls and Indicators (Sheet 3 of 6)

Key	Label	Description and Function
(16) (110)	RESET	Pushbutton resets the CPU and the I/O system, and causes control to be transferred in accordance with the cold start line (CLDS-) and (on a GA-16/220) the CNSL switch (15). On a GA-16/110, the cold start line determines the function of the RESET switch as follows: cold start high: Control is transferred via auto-restart vector X'41'. cold start low: Control is transferred to the IPL ROM. On a GA-16/220, the cold start line determines these functions only if the CNSL switch (15) is not in the CNSL position. In the CNSL position, control is transferred to the console ROM on the SCI. This function may be remotely controlled via the RESET and SYRT lines, together.
(17) (110)	IACK	Interrupt acknowledge indicator blinks when control has passed to a routine via an interrupt vector. It normally blinks so rapidly as to be barely visible. If continuously illuminated, it indicates that the control has not returned from the interrupt processing routine, a "hung-up" condition.
(18) (110)	ISE	Interrupt system enabled indicator is illuminated when the ISE flip-flop is set, and means that the inhibitable interrupt system is enabled. Refer to description of instructions INE and RISE (Section 4 Systems Reference Manual).
(19) (110)	OMA	This indicator (also referred to as the OMA stall indicator) is illuminated when the operators monitor alarm (OMA) has timed out. The OMA must have been initially turned on by a PMA instruction (Section 4.13.7 Systems Reference Manual). When the OMA indicator illuminates, the RUN indicator (21) is extinguished and the CPU is in an idle state. To recover from an OMA, the operator must press the RESET button (16).
(20) (110)	FGND	Foreground indicator is illuminated when the foreground registers are used and extinguished when background registers are used. Refer to instructions BMS and FMS (Sections 4.13.1 and 4.13.2 Systems Reference Manual) for setting foreground or background register usage.
(21) (110)	RUN	The RUN indicator is illuminated when the CPU is in the run mode. If the WAIT indicator (22) is also illuminated, the CPU is in a wait condition as a result of executing a WAIT instruction. If MPP indicators (27) through (30) are illuminated, memory parity error has caused the stall. In order for an MPP stall to occur, the STALL switch (31) must be in the stall position. To recover, the user must RESET the system. If the run indicator is extinguished, the system is in idle.

Table 2-5. GA-16/110/220 Controls and Indicators (Sheet 4 of 6)

Key	Label	Description and Function
(21) (110)	RUN (Cont)	*NOTE: If 18-bit memories are installed without installation of MPP module, a stall condition may occur if the parity override switch (36) is not set. Refer to Section 3.2.7.1, Systems Reference Manual, and description of (36), (37), and (38) in this table.* A remote idle indicator may be implemented via the RUN line, and conversely grounding the RUN line will force the CPU into the idle condition.
(22) (110)	WAIT	Wait indicator illuminates when a WAIT instruction has been executed (Section 4.13.10 Systems Reference Manual).
(23) (110)	-	Unlabeled 2-position battery backup switch faces to the rear of CPU-1 module, and must be set prior to installation. In the backup position (down), +5VB power for the memories (including the piggyback RAM) originates in the backup power supply or from batteries (when AC power is disconnected). When the switch is not in backup position, the power is obtained from the main power supply. A memory service module must also be installed in the rear connector (as shown in Figures 2-2, 2-3, 2-5, 2-6) to obtain power from the main power supply for the 8K/4K RAM memories when the backup power supply is not used.
(24) (MPP)	DPT	The DMA write protect indicator, when illuminated, indicates that an attempt has been made by DMA to write into a memory area (via the high-speed data channel) which has been DMA-protected (refer to Section 5 Systems Reference Manual).
(25) (MPP)	ME	Multiple error indicator illuminates when a second error occurs before software can process a previous error. Other indicators also may be illuminated.
(26) (MPP)	PPT	Program write protector indicator, illuminates when a program has attempted to write data in a memory area which has been program-protected (refer to Section 5 Systems Reference Manual).
(27) (MPP)	LPB	Lower parity bit indicator shows the contents of parity bit for the lower byte of a memory word (illuminated = 1). Content of bit must be compared with the lower byte to determine if an error has occurred; even parity is an error.
(28) (MPP)	UPB	Upper parity bit indicator shows contents of the parity bit for the upper byte of a memory word (illuminated = 1). Content of the bit must be compared with the upper byte to determine if an error has occurred; even parity is an error.

Table 2-5. GA-16/110/220 Controls and Indicators (Sheet 5 of 6)

Key	Label	Description and Function
(29) (MPP)	DPY	DMA parity indicator illuminates when a parity error is detected during a DMA transfer.
(30) (MPP)	PPY	Program parity indicator illuminates when a parity error is detected when a program reads a memory location (data or instruction fetch). *NOTE: Indicators (24) through (30) remain illuminated until MPP status is reset under software control. Refer to Section 5, Systems Reference Manual.*
(31) (MPP)	STALL	The 2-position switch faces to the rear of the MPP module and is preset prior to installation. This switch determines CPU action when a DMA or CPU parity error occurs. When the switch is in STALL position, the memory bus is forced to a busy state, thereby halting the CPU. When not in the STALL position, control is passed to a routine via non-inhibitable interrupt vector X'42'. (All non-parity errors pass control via X'42' regardless of the setting of STALL switch, and a routine must determine type of error.)
(32) 15 (33) 14 (34) 13 (MEM)		Three 2-position switches used on both 8K and 4K memory modules to set the memory to occupy locations starting at one of eight 8K boundaries. Refer to Section 3.2.7 Systems Reference Manual for memory map and corresponding switch settings.
(35) (MEM)	12	The 2-position switch installed only on 4K memories. It selects the upper and lower 4K boundary within the 8K boundary set by switches (32), (33), and (34). Refer to Section 3.2.7, Systems Reference Manual, for memory map and corresponding switch settings.
(36) (MEM)	PAR OVRD	The 2-position switch is installed only on 18-bit memories. This switch is effective only if MPP module is not installed and enables parity override which disables parity error detection when in DOWN position. When in UP position, parity error detection occurs. Action upon a parity error depends on whether or not an MPP module is installed: - When MPP module is not installed and switch is in the UP position, a parity error will cause a CPU wait condition on data fetches or repeated attempt to execute instruction fetches, and the DERR and IERR indicators will identify the error. - When an MPP module is installed, switch may be in either position and parity errors will be identified by the LPB, UPB, DPY, and PPY indicators (27) through (30) on the MPP module. Detection is enabled by PIO to the MPP module (Section 5 Systems Reference Manual). Action taken upon parity error detection then is determined by the setting of the stall switch (30) on the MPP module.

Table 2-5. GA-16/110/220 Controls and Indicators (Sheet 6 of 6)

Key	Label	Description and Function
(37) (MEM)	DERR	The data error indicator illuminates (in 18-bit memories) when a parity error occurs on a data fetch. The data comes out as '0000'. This condition is cleared by a system reset or by a break reset (see description of (16) and (12).)
(38)	IERR	This instruction error indicator (in 18-bit memories) illuminates when a parity error occurs on an instruction fetch. The instructions comes out as '0000' (WAIT). This condition is cleared by a system reset or by a break reset. *NOTE: If the Memory Parity Protect (MPP) option is included in a system, neither the IERR nor DERR indicators will illuminate unless there is a high failure rate. The MPP is managing the memory under program control.*
(39) (BAT)	-	Unlabeled indicator illuminates when power (either from AC line or from batteries) is applied to the memory modules. Light is extinguished when manual cut-off button (38) is pressed (or if batteries are exhausted) provided AC power is disconnected from the auxiliary power supply.
(40) (BAT)	-	Unlabeled pushbutton provides manual cut-off of battery power to memory modules. Pressing this button will have no effect unless AC power is disconnected from auxiliary power supply. If AC power is disconnected, pressing this button will cut off power to memories. Power will not be restored to memories until AC power is reconnected. *NOTE: Application of power to auxiliary power supply is independent of built-in main power supply switch (1), or other means of controlling power to external main power supply.*
(41) (BAT)	BAT CHG ADJ	This control is used to adjust the charge voltage to the battery pack to maintain a 0.5 ampere rate (maximum). Setting for 12-volt, 1.5 ampere battery supplied is 13.6 volts.
(42) (BAT)	5VB	This control is used to adjust the +5-volt regulator.
(43) (BAT)	VDD	This control is used to adjust the +12-volt regulator.

2.3 SYSTEM INTERFACE

Figure 2-9 depicts the general system interface signal configuration, including those signals used in I/O operations. For further I/O interfacing information, reference the GA-16/110/220 System Reference Manual, document number 88A00508A, Section 6.

2.4 POWER SUPPLIES

2.4.1 COMPACT POWER SUPPLY

The plug-in (compact) power supply is a switching, regulator-type providing up to 18 amps of +5VDC. The jumbo power supply provides up to 30 amps.

The compact power supply operates on 115 ±10% VAC. The jumbo power supply contains standard transformer taps for nominal voltages of 100, 115, 200, and 230 VAC sources ± 10%.

2.4.2 BACKUP POWER SUPPLY

The backup power supply performs the dual functions of providing +12VDC and ±5VDC to the memory during normal operation, and during AC input power interruption, to maintain memory contents. The backup power supply consists of two parts:

- Battery pack.
- Regulator board.

2.4.2.1 Backup Power Supply Installation

1. Mount the battery pack on the left inside wall of the chassis. The mounting screws are the two Phillips on the side of the battery pack. They are threaded into the sidewall of the battery pack and support nothing inside the pack. The AC cable is at the top rear of the battery pack.
2. Slide the regulator board part-way in the guides for the left-most card slot (viewed from front). Then connect the DC cable from the battery pack to the 3-prong connector on the lower area of the regulator board (with the AC cord disconnected).

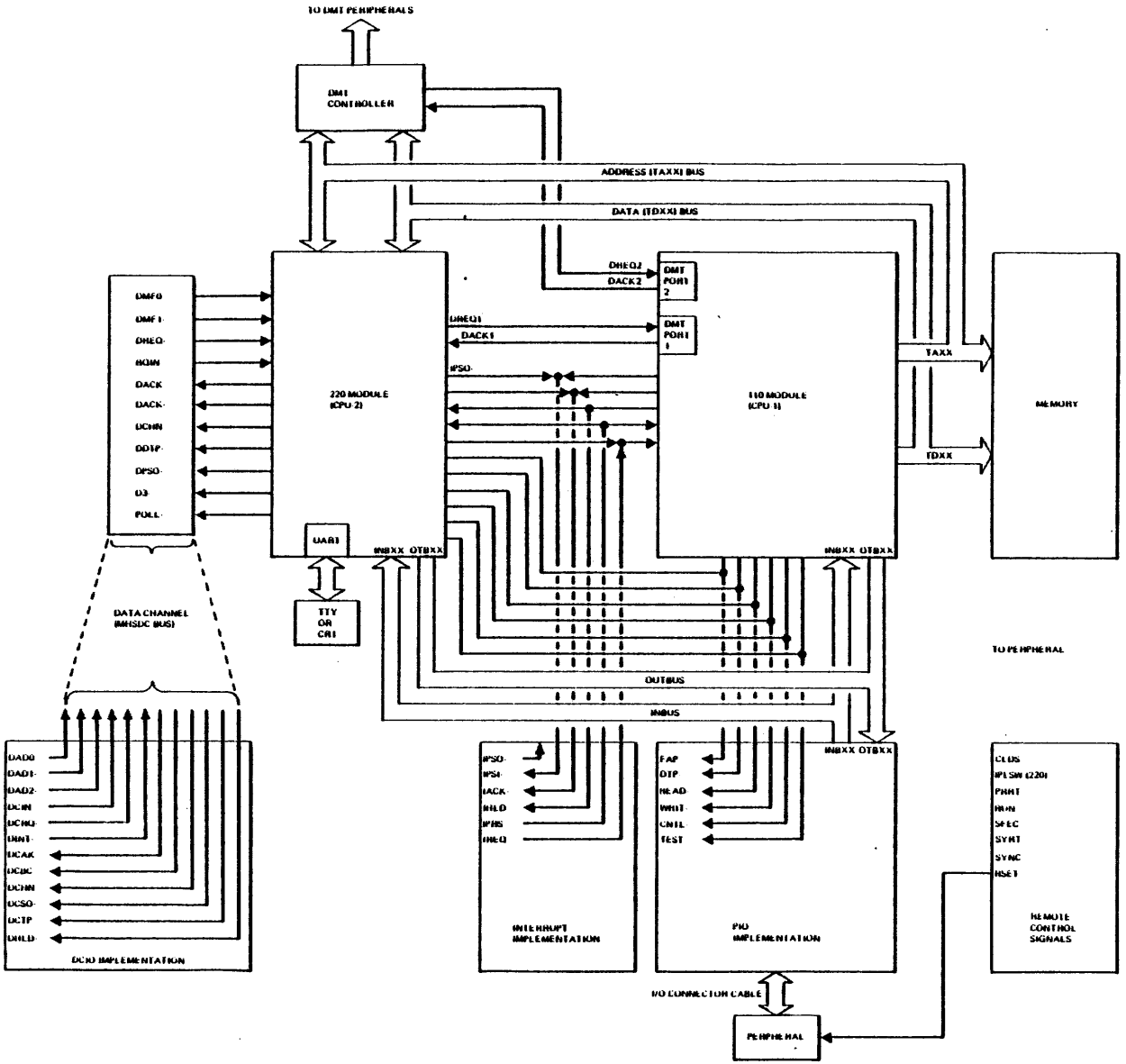
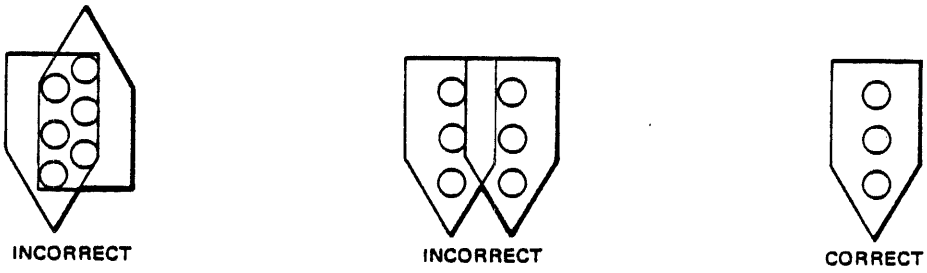


Figure 2-9. GA-16/110/220 Interface

CAUTION

The three pins in the connector are keyed to prevent reversed connection. However, it is possible to offset and misalign the pins and make an improper connection.



2.4.3 MEMORY SERVICE MODULE

If the backup power supply is not installed in the system, the Memory Service Module (MSM) (90C02301A), must be plugged into J21 on the back of the compact chassis. The MSM (when installed) provides +12VDC and -5VDC to the memory modules. When the MSM is installed, S2 (90C02422A Sheet 7) must be closed (UP position) to supply +5VB to the memories (from the system +5V line).

2.4.4 EXTERNAL POWER SUPPLY

An external power supply, such as that used to power a standard GA-1615 I/O cage, may also be used to power the GA-16/220. This external power supply is connected to the processor cabinet via connector J-30, for a Jumbo Chassis, or via connector J-13, for a Compact Chassis.

An external power supply may also be user-supplied via these same connectors.

theory of operation 3

To understand the inner workings of the GA-16/110 or GA-16/220 processor, it is necessary to fully comprehend the operation of the CPU microcode. Section 3 is intended to provide the understanding needed in this area as well as to explain the effects of this microcoding on the processor logic.

3.1 GA-16/110/220 MICROCODE

The GA-16/110/220 system is a microprocessor-based device relying upon the execution of a series of microcode instructions to implement each CPU instruction. These microinstructions are 34 bits in length and reside in the CROM. Each of the available microinstructions follows the format shown in Figure 3-1. For a complete list of the GA-16/110/220 microinstruction set, refer to Appendix B.

G	Br	NADDR	R/C	B-CNTL	FUNCT	Rs	Rd
			I	I I I I I	I I I I	I I I I	I I I I
			P	P P P P P	P P P P	P P P P	P P P P
3 Bits	5 Bits	8 Bits	D	D D D D D	D D D D.	D D D D	D D D D
			H	G F E D C	B A 9 8	7 6 5 4	3 2 1 0

Figure 3-1. 34-Bit Microinstruction

The G-field controls the gating of the foreground status bit, as well as the decode of the interpretation of the destination/source register identifier bits, when included as part of the program instruction in the I register. The Branch field, (Br field) aids in the selection of the next microinstruction from the CROM. This is accomplished by ORing the Br field with the Next Address Field (NADDR field). The Read/Control bit (R/C bit) selects the data for the destination register. If this bit is a logic "1", source data is loaded into the destination register. If this bit is a logic "0", no data is loaded into the destination register. The bus-control field, (B-CNTL field) controls the gating of signals to the ZPOL indicators as well as all data flow within the processor. The function, register source, and register destination fields, (Funct, Rs, Rd fields) combine to specify all register gating internal to the CROM and the RALU as well as the source and destination registers when not specified in the program instruction.

3.1.1 G FIELD

The G field specifies the foreground (FGLP) status bit gating and determines the source and destination registers (when applicable) by decoding the contents of the I register. (Refer to Table 3-1, G-Field Decode.)

Table 3-1. G-Field Decode

Code	Function	
000	No-op	
001	F→IPD3	
010	F→IPD7	
011	F→IPD7 & IPD3	
100		I7, I6, I5 →Rs designator
101	F→IPD3	I7, I6, I5 →Rd designator
110	F→IPD7	I10, I9, I8→ Rd designator
111	F→IPD3 & IPD7	I10, I9, I8 →Rs designator I7, I6, I5→ Rd designator

A G-field equal to X'0' is treated by the hardware as a NO-OP. A G-code of any value other than X'0' or X'4' gates an indication of the status (on/off) of the foreground (FGLP) bit into the most-significant bit position of the specified Rs and/or Rd field, IPD3 or IPD7.

When a G-field of X'4', X'5', X'6' or X'7' exists, gating is enabled to decode Rs and/or Rd directly from the program instruction in the I register bits I7, I6, I5 and/or bits I10, I9, I8. This allows for program selection of source and destination registers.

3.1.2 NEXT ADDRESS AND BRANCH FIELDS (NADDR and Br)

The 8-bit NADDR field is used in conjunction with the 5-bit Br field to generate a 9-bit microinstruction address (Next CROM address). This is accomplished as follows:

NADDR Bits	7	6	5	4	3	2	1	0	*	
Br Code Bits	*	*	*	*	E_{BT}	D_{BT}	C_{BT}	B_{BT}	A_{BT}	OR'ed (NOTE: XBT bits are shown in Table 3-2)
CROM Address Bits	8	7	6	5	4	3	2	1	0	

Next CROM Address Formula

The Branch code is a hex number between '00' and '1F'. The Br code selects a certain combination of bits to generate the ABT through EBT values. In Figure 3-2, all numbers represent that bit position value in the I register. For example: a Br value of X'02' indicates that a memory reference or I/O operation instruction (MR,IPL) is being executed in the register. With this Br code, EBT and DBT both equal zero, CBT will be the result of ANDING I register bit 13 with the complement of I_{12} ($\overline{12}\cdot 13$), BBT will be the value of I_{11} and ABT is the result of OR'ing I_{10} with the compliment of I register bit 13. These resultant values are then plugged into the formula for determining the CROM address bits and are OR'ed with the value in NADDR.

If the microinstruction currently being executed is a Classify Branch (CLSFY, X'1B'), indicating that a new program instruction has just been fetched from memory, a special decode is necessary. A Classify Branch operation is entered each time a new instruction is placed into the I register in order to interpret that (user) instruction and determine the first CROM address in the microroutine unique to this new program instruction. In this CLSFY operation, EBT and DBT are both zeroes, CBT is identified as SC, BBT as SB and ABT as SA. These three special interpretations (SA, SB and SC) are shown at the bottom of Table 3-2, and in Table 3-3.

A Br of X'00' simply forces ABT to a zero (as well as all other branch code bits) and a Br code of X'16' is unused.

Mnemonics such as $\overline{\text{RUN}}$ or $\overline{\text{IOT}}$ are the sampling of that conditions status. $\overline{\text{IOT}}$ for example, would set ABT (in Br code '1E') to a logic '1' of the I/O test line from the peripheral controller being addressed is false.

Table 3-3 is a list of these special branch combinations and include W_1 through W_4 which are used later in the text to indicate conditions of sign extension (spread) or truncation of the W register during address modification operations.

Table 3-2. Branch Codes

All numbers refer to "I" register bit positions

Br	EBT	DBT	CBT	BBT	ABT	Function
00						LSB=0
01				9	10	SKP
02			$\overline{12}\cdot 13$	11	$10 \vee \overline{13}$	MR,IOL
03				11	$10 \vee \bar{9}\cdot\bar{8}$	MRWI,MRWIB
04			3V1	4V3V2	4V0	RCS,RCD,RCO
05			14	9	8	SPO,SP1,IDX
06				11	ZO	SKPB
07				11	P_L	SKPB
08				7	6	IOLA
09					6	
0A				11 6V7	6	IOL
0B				$\bar{7}$V6V4	6V5V4	WMD
0C		4	3	2V1	0	RO,ROL
0D		4	8	REA**	LK	SEX,SFCM,SFCD
0E				11	OF	SKPA
0F				11	LK	SKPB,SFBM,SFBD,SBA, SCS,RCD,RLK
10					*	LSB=1
11					INTMP***	
12			$\overline{14}\cdot 13 \vee 13\cdot\overline{12}$	$15\cdot\overline{14}$	$12\cdot(\overline{14} \vee 6)$	BYE,BYO
13					12	LSI+2
14				$\overline{10}$	$13 \vee 12\cdot\overline{10}$	LSI
15			$\overline{14}$ 13	$\overline{10}$	$\overline{14} \vee 13\cdot\bar{7}\vee\overline{12}$	MRWIE
16						UNUSED
17					DO	RBOP
18					$\overline{RUN}$	
19			SVI	$\overline{RUN}$	$\overline{INT}$	
1A					$\overline{ENA}$†	STBY
1B			SC††	SB††	SA††	CLSFY
1C	$\overline{SWD}$	$\overline{SWC}$	$\overline{SWB}$	$\overline{SWA}$	$\overline{ENTR}$	REG DISPLAY
1D			$\bar{7}\cdot(6\vee\bar{5}\vee 3\cdot 2)$	$\bar{7}\cdot(6\vee\bar{5}[3\vee 1\cdot 0])$	$\bar{7}\cdot(6\vee\bar{5}\cdot 4\vee\bar{5}\cdot\bar{3}\cdot 1)$	CTRL
1E					$\overline{IOT}$†††	INCP
1F					$\overline{STEP}$	WAIT,WAITA

* For Br of '10' - ABT will = A logic '1'

** REA - W register (R) equals the A-Bus (Bits 0→3). This indicates a shift complete.

*** INTMP - Interrupt Mode Prime. NI Interrupt.

† ENA - Enabled at A. Signal, internal to the CROM, used with multiple CROM's to indicate that at the last A clock the CROM was selected.

†† SA = $15\vee\overline{14}\vee 13\cdot 12\vee SPC\cdot\overline{11}\cdot 10$
SB = $15\vee\overline{14}\cdot 13\vee SPC\cdot 11$
SC = $SPC \vee S1\cdot(15\vee\overline{14}\cdot 13\cdot 12)$
SPC = $\overline{15}\cdot\overline{14}\cdot\overline{13}\cdot\overline{12}\cdot$("I" Reg = 0XXX Instruction)
S1 = $4\cdot 3\cdot 2\cdot 1\cdot 6\cdot$("I" Reg - MRWI Instruction XX1F)

††† IOT = I/O Test signal from peripherals. Terms from RALU = ZO, PL, OF, LK REA, DO
External Inputs: INTMP, RUN, SVI, INT, ENTR, SWA, SWB, SWC, SWD, IOT, STEP

Table 3-3. Special Branch Logic Combinations

Dx - Bit 'X' output from 'D' Latch (0 or F)
SA - 15V14V(13·12)V(SPC·$\overline{11}$·10)
SB - 15V($\overline{14}$·13)V(SPC·11)
SC - SPC V (SA·SB·S1)
SPC - $\overline{15}$·$\overline{14}$·$\overline{13}$·$\overline{12}$
S1 - 4·3·2·1·0
SE - SWI V SWP V SW1 - SWA
SF - SWI V SWW V SW2 - SWB
SH - SE V SV V SWS - SWD
SG - SW4 - SWC

SWX - Console Register Select Switches on SPC-16

Spread (<>) and Truncate (11) Functions

MR	W1	-	<W>9	-	Extend Sign (Bit 9)
	W2	-	|W|9	-	Truncate (0 →Bit 15 →10)
MRWI	W3	-	|W|4	-	Truncate (0 →Bit 15→ 5)
SKIP	W4	-	<W>8	-	Extend Sign (Bit 8)

3.1.3 READ/CONTROL FIELD (IPDH)

The R/C field is a "write-into-register" control bit. If this bit is set to a logic "1" (true), the selected destination register (Rd) will be enabled to receive the output of the adder. If R/C is a logic zero (false) the Rd will be disabled. This bit is available for monitoring as the IPDH+ output of the CROM chip Z4E pin 27.

3.1.4 BUS-CONTROL FIELD (IPDG - IPDC)

The B-CNTL field will either enable or disable the ZPOL outputs of the adder to be written into the status register (S) or will generate the signals necessary to effect the gating shown in Table 3-4.

In the most significant bit of the B-CNTL field is a zero (IPDG=0), the outputs of the ZPOL status from the adder will set the corresponding bits in the status register (S) providing its bit in the B-CNTL field is set.

CROM Output term	$I_{P_{D_G}}$	$I_{P_{D_F}}$	$I_{P_{D_E}}$	$I_{P_{D_D}}$	$I_{P_{D_C}}$
B-CNTL Enable/Disable	0	X	X	X	X
	$E_{N_{A_{B_{L_E}}}}$	Z_O	P_L	O_F	L

(X = 1 to enable input to S from adder)

For example, if it is desired to allow the Plus (P_L) status to be gated to S, then IPDG must be a zero and IPDE must be a '1' (B-CNTL = X'04). If IPDE were a zero, the plus indicator input to S would be disabled.

If IPDG were set to a '1', then the B-CNTL field definition would be from Table 3-4. A B-CNTL of X'13' (I/O Data → RALU) would enable the proper logic terms to gate incoming I/O data into designated general-purpose register in the RALU.

Table 3-4. Bus Control

Code	Function	Hex Code
10000	MD → RALU (READ)	10
10001	CONSOLE SW → I (CROM)	11
10010	IVA → RALU	12
10011	I/O DATA → RALU	13
10100	MD → RALU (READ/MODIFY/WRITE)	14
10101	CONSOLE SW → RALU	15
10110	SPEC → RALU*	16
10111	I → RALU	17
11000	RALU → MA	18
11001	RALU → I	19
11010	RALU → I/O ADD.	1A
11011	RALU → I/O DATA	1B
11100	RALU → MD	1C
11101	RALU → CNSL	1D
11110	RALU → SYNC**	1E
11111	I → CNSL***	1F

* SPEC → RALU = IPL in NI mode. IPL address to 'P' and automatic start of a Memory Cycle.

** Used with H.S. MPY DIV or F.P. for SPC-16 compatability.

*** Requires an operators console (not available from GA at this time).

3.1.5 FUNCTION Rs AND Rd (IPDB THROUGH IPD0)

The Func, Rs and Rd fields combine to place the microinstruction into 1 of 16 general instruction groups. These instruction groups may be as simple as a register-to-register transfer operation, where data is taken from the source register and placed into the destination register, or as complex as a byte-oriented instruction where half-words of data are mixed with the contents of a register and the results placed into a specific register.

The classification of a microinstruction begins with a general combination of the function field itself. The first examination will catagorize the instruction group into which the microinstruction falls. Table 3-5 lists these classifications by functions codes.

Table 3-5. Function Field General Classification

Mnemonic	Funct	Rs	Rd	Function
RTR	0	Rs	Rd	Rs → Rd
ADD	1	↓	↓	Rd + Rs → Rd
SUB	2	↓	↓	Rd - Rs → Rd
AND	3	↓	↓	Rd • Rs → Rd
XOR	4	↓	↓	Rd ⊻ Rs → Rd
OR	5	Rs	↓	Rd V Rs → Rd
SHG	6	MOD	↓	Shift group
BYMR	7	MOD	↓	Byte operation
ADWHR	8	WMOD	↓	Rd/2+|W| → W
ADWRW	9	WMOD	↓	Rd + |W| → W
ADWRP	A	WMOD	↓	Rd + |W| → P
MSC	B	MOD	Rd	Miscellaneous group
LLW	C	α		W_U, α → W
LUW	D	α		α, W_L → W
SRL	E	MOD	RMOD	Operations with P,W,S
LGL	F	LIT	MOD	Logical Operations (Literal)

NOTE: 00X = No Op.

where: X ≠ 0 to prevent both busses from loading the same register.

If a microinstruction belongs to one of the first six groups (RTR, ADD, SUB, AND, XOR, OR), one of the "W" register instructions (ADWHR, ADWRW or ADWPP) or either of the microcode byte operations (LLW, LUW), no further classification is necessary. If the FUNCT field decodes as any of the remaining functions (SHG, BYMR, MSC, SRL or LGL), the Rs and/or Rd field must be examined as a modifier to determine the actual operation to be performed.

3.1.5 1 Stand-Alone Function Fields (No Modification)

If a FUNCT/Rs/Rd field were to equal X'104', it will fall into the ADD (Rd+Rs → Rd) group of instructions. In this group, Rd (in this case "B") will be added to the Rs (in this case "A") with the results placed into the Rd (register "B").

However, if the FUNCT/Rs/Rd field were to equal X'BXX' placing the microinstruction in the miscellaneous (MSC) group, a further examination of the Rs field must be made, as it is acting as a modifier and will determine the actual operation performed.

Each of the groups necessitating further interpretation for a definition of the operation to be performed, will be examined.

3.1.5.2 Shift Group

A function field of X'6' places the microinstruction into the Shift group (SHG) of instruction. In actuality only half of the codes within this group perform shift operations; the remainder of the operations are logical/arithmetic operations.

Table 3-6 summarizes the 16 operations performed within this group. The operation performed is a function of the Rs field since it is acting as a modifier.

Table 3-6. Shift Group Operations

Mnemonic	Funct	Rs	Rd	Function	
SHG	6	0	Rd	W → Rd	
Arithmetic		1		Rd + W → Rd	Arithmetic
Operations		2		Rd - W → Rd	and Logical
with results		3		Rd • W → Rd	Operations
to Rd		4		Rd ⊻ W → Rd	
		5		Rd V W → Rd	
		6		Rd + 1 → Rd	
		7		Rd - 1 → Rd	
		8		R_{DF}, R_{DF-1}, → Rd	Right shift extend bit 15
		9		R_{DO}, R_{DF-1}, → Rd	Right shift circular
		A		0, R_{DF-1}, → Rd	Right shift, zero bit 15
		B		1, R_{DF-1}, → Rd	Right shift, set bit 15
		C		R_{DE-0}, R_{DF}, → Rd	Left shift circular
		D		R_{DE-0}, R_{DO}, → Rd	Left shift, extend bit zero
		E		R_{DE-0}, 0 → Rd	Left shift, zero bit zero
		F		R_{DE-0}, 1 → Rd	Left shift, set bit zero

The first eight operations are the result of an instruction, such as an AND instruction being executed in the I register. An AND instruction would generate a FUNCT/Rs/Rd field of X'63X' to effect ANDing of Rd with the value specified in the instructions (now in W register). The result is then gated into Rd (only if R/C = 1). Rd will equate to the decoding of the 'X' in X'63X'.

An Rs modifier of '8' through 'B' specifies a right shift operation. A code of X'68X' (R_{DF}, R_{DF-1} → Rd) is a right shift with bit 15 (R_{DF}) extended. This will place the designated register bits 15 through 1 (R_{DF-1}) into the destination registers bits 14 through 0 and bit 15 (R_{DF}) will be placed back into the destination register bit 15 position, giving the effect of extending bit 15.

The same format that is used in the right shift operations is used in the left shift operations. Left shift operations are commanded by function codes in the range X'6CX' through X'6FX'.

NOTE

It must be remembered that, in order to affect the link indicator in 'S', the B-CNTL field must equal 0XX1.

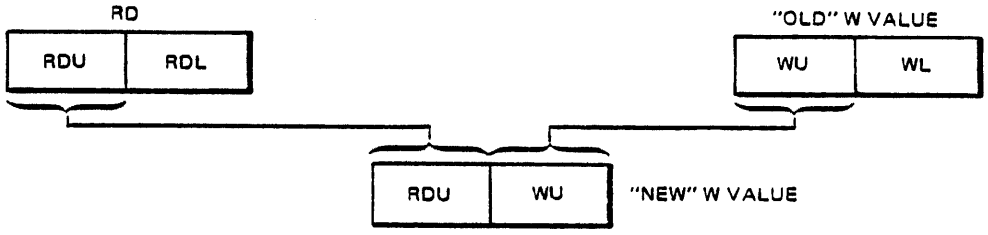
3.1.5.3 Byte Oriented Operations (BYMR)

All byte oriented operations (Function code = X'7XX') manipulate data in memory on a byte or half-word basis.

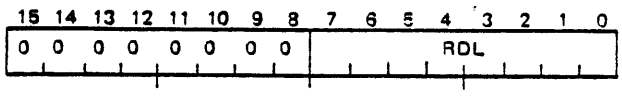
Table 3-7. Byte Oriented Operations

Mnemonic	Funct	Rs	Rd	Function
BYMR	7	0	Rd	RDU,WL → W
		1		RDU,WU → W
		2		RDL,WL → W
		3		RDL,WU → W
		4		RDU,WL → Rd
		5		RDU,WU → Rd
		6		0,RDL → Rd
		7		0, 0 → Rd
		8		WU,RDL → W
		9		WU,RDU → W
		A		WL,RDL → W
		B		WL,RDU → W
		C		WU,RDL → Rd
		D		RDU, 0 → Rd
		E		WL, RDL → Rd
		F		RDL,RDU → Rd

In these byte operations, 16 bits of data will be written into the specified destination register. These 16 bits, however, are manipulated as two equal, 8-bit, blocks of data. For example, if a function field of X'71X' is specified (R_{DU}, W_U → W), the most-significant 8 bits of the designated register (R_{DU}) will be written into the upper 8 bits of the W register. At the same time, the upper 8 bits of W (W_U) will be gated back into W, but they will become the lower 8 bits of the new value in W. Graphically, this would be represented as:



In those cases where a zero is given in the operation field of Table 3-7, that byte will be zeroed.



509-3-1

3.1.5.4 Miscellaneous Group

The MSC group is appropriately named since it consists of a collection of somewhat unrelated functions. (See Table 3-8. Miscellaneous Group Operations).

Table 3-8. Miscellaneous Group Operations

Mnemonic	Funct	Rs	Rd	Function
MSC Miscellaneous	B	0	Rd	W → W
		1		R_d + W → W
		2		R_d - W → W
		3		R_d · W → W
		4		R_d ⊻ W → W
		5		R_d V W → W
		6		R_d → W
		7		W → W Interrupts (to save W & ISE)
		8		R_d ⊻ FFFF → R_d (R_d → R_d register compliment)
		9		M → R_d
		A		S → R_d
		B		P → R_d
		C		R_d → P
		D		R_d → S RTN
		E		R_d → P RTRN S15 → ISE (64K mode) R15 → ISE (32K mode)
		F		R_d → S TRS

This group is used in a multitude of operations from register compliment instructions (X'B8X') to register display, to return from interrupt instructions. (Reference Appendix B Microcode Summary.)

3.1.5.5 SRL Group

Most operations, involving the P, W, and S registers, fall into the SRL group. This being a large group of instructions , requires both the Rs and Rd field to act as modifiers of the FUNCT field (See Table 3-9).

Table 3-9. SRL Group Operations

Mnemonic	Funct	Rs	Rd	Function
SRL Operations with P,W,S	E	0	0	S → W
		1		W + S → W
		2		W − S → W
		3		W • S → W
		4		W ⊻ S → W
		5		W ∨ S → W
		6		W + 1 → W
		7		W − 1 → W
		0	1	P → W
		1		W + P → W
		2		W − P → W
		3		W • P → W
		4		W ⊻ P → W
		5		W ∨ P → W
		6		M → W (M → W & I)
		7		M → W
	E	0	2	S → P
		1		P + S → P
		2		P − S → P
		3		P • S → P
		4		P ⊻ S → P
		5		P ∨ S → P
		6		P + 1 → P
		7		P − 1 → P
		0	3	W → P
		1		P + W → P
		2		P − W → P
		3		P • W → P
		4		P ⊻ W → P
		5		P ∨ W → P
		6		M → P
		7		M → P
		0	4	P → S
		1		S + P → S
		2		S − P → S
		3		S • P → S
		4		S ⊻ P → S
		5		S ∨ P → S
		6		S + 1 → S
		7		S − 1 → S
		0	5	W → S
		1		S + W → S
		2		S − W → S
		3		S • W → S
		4		S ⊻ W → S
		5		S ∨ W → S
		6		M → S
		7		M → S

SRL codes having an even number, Rd with Rs codes of 6 and 7 (e.g., X'E60' W+1→W), are used as increment/decrement functions. SRL codes with an odd number, Rd code and Rs codes of 6 and 7 (e.g., X'E71 M → W), are used in memory transfer instructions.

3.1.5.6 Logical/Literal Group (LGL)

The LGL group of instructions perform logical and arithmetic operations between a specified bit in a literal value and the P, W, and S registers. (See Table 3-10.)

Table 3-10. Logical/Literal Group Operations

Mnemonic	Funct	Rs	Rd	Function
LGL	F	L	0	$W \cdot L \rightarrow W$
Logical			1	$W \vee \overline{L} \rightarrow W$
Operations		Literal	2	$W \cdot \overline{L} \rightarrow W$
(with P, W, S)		Value	3	$W \veebar L \rightarrow W$
		Bit	4	$S \cdot \overline{L} \rightarrow S$
		Positions	5	$S \vee L \rightarrow S$
			6	$S \cdot L \rightarrow S$
			7	$S \veebar L \rightarrow S$
			8	$P \cdot L \rightarrow P$
			9	$P \vee \overline{L} \rightarrow P$
			A	$P \cdot \overline{L} \rightarrow P$
			B	$P \veebar L \rightarrow P$
			C	$L \rightarrow W$
		*	D	$0 \rightarrow S_{L-0}$
		WMOD	E	$\lvert W \rvert \rightarrow \overline{W}$
	↓	↓	F	$\langle W \rangle + P \rightarrow P$

* Originally used to clear "S" lower (Shift Counter)
Current functions:

X'F0D' = 0 → SL (clears SL)
X'F1D' = Reset ISE
X'F2D' = Set ISE
X'F3D' = Wait
X'F4D' = PMA

A function code of X'FX0' through X'FXB' will perform an arithmetic or logical operation between a literal value bit position (specified by Rs) and the corresponding bit position of the assigned register (P,W, or S). For example, if the function code is X'F31', (W V L → W) bit position 3 (the value "L") of a literal value, specified by the instruction in I, is OR'ed with bit position 3 of the value in the W register. The result is stored in W.

A function code of XFXC (L → W) allows loading a specified bit value, from the literal, into the matching bit position in the W register. If the function code is X'FXD' then five possible choices of operation exist: clear S_L (X'F0D'), reset ISE (X'F1D'), set ISE, (X'F2D', a wait command (X'F3D') or pulse monitor alarm (X'F4D').

The last two LGL operations either truncate W (X'FXE') or modify addressing during an MR jump operation (X'FXF').

3.1.5.7 ADWxx Operations

These three operations (ADWHR, ADWRW and ADWRP) are utilized in bit/byte operations and during address modification sequences by working with W.

- ADWHR (Rd/2 + |W| → W) adds one half of the value in a specified register to W and places the results into W. This operation is used during bit/byte operations for selection of upper or lower byte.
- ADWRW (Rd + |W| → W) adds a specified register to the value contained in W and places the result into W. This procedure is for address modification during indexed operations.
- ADWRP (Rd + |W| → P) adds a specified index register to the value in W and places the result into the P register. This code is utilized for address modification during indexed operations.

3.1.5.8 LxW Operations

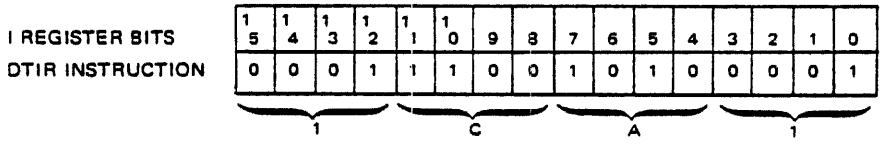
The last two function fields, LLW (X'Cxx') and LUW (X'DXX') allow loading of a byte of microcode (Rs and Rd fields) into either W lower or W upper, respectively. This may allow loading these bytes of microcode into any register by following this operation with a register-to-register (RTR) instruction.

3.2 MICROCODE OPERATIONS ON DTIR

All instruction manipulation and operation, within the GA-16/110/220 is controlled by the microprocessor (CROM/RALU) chipset logic. Resident within the CROM is a complete microprogram unique to each CPU instruction available in the processor repertoire.

Each time a new program instruction is fetched from the CPU memory, the microprocessor executes a Classify Branch microinstruction to examine the newly fetched CPU instruction. The Classify Branch microinstruction translates specific bits within the CROM instruction register (I) to determine the type of instruction that is to be executed. This determination will allow calculation of the next address (Branch Address) within the CROM microcode necessary to locate the correct microinstruction for the next step required in the proper execution of the CPU instruction in the I register.

For example purposes, a DTIR B,X'21' instruction microprogram sequence will be explained in detail. This instruction (machine language code X'1CA1') is placed into the CROM I register in the following format:



509-3-2

The purpose of this instruction is to gate data from peripheral device address X'21' into the CPU B register. The I register is internal to the CROM; the B register is internal to the RALU.

The DTIR microinstruction sequence is:

CROM Address	Mnemonic	Hex Code	Next CROM Address	Operation
X'028'	CLSFY	0 2869 A600	X'036'	W → I/O Add
X'036'	IOL (+2)	0 0064 0000	X'032'	(No-op)
X'032'	IOLA(+1)	0 0A18 0000	X'10F'	(No-op)
X'10F'	DTIR	3 4007 3B90	X'003'	I/O Data → Rd
X'003'	INCP+1	0 6443 8E62	X'021'	P + 1 → P, MA
X'021'	FTCH	0 6C53 0E61	X'028'	MD → W,I

For a complete microcode listing, reference Appendix B.

3.2.1 MICROINSTRUCTION AT X'028'

The first microinstruction in the DTIR instruction sequence (located at CROM address X'028') is the Classify microinstruction. The classify microinstruction begins the interpretation of the newly-fetched CPU instruction (DTIR) that is now resident in the I register.

The Classify microinstruction (X ' 0 2869 A600') decodes as:

	G	Br	NADDR	R/C	B-CNTL	FUNCT	Rs	Rd
Binary	0 0 0	0 1010	0001 1010	0	1 1010	0110	0000	0000
Hex	0	0A	1A	0	1A	6	0	0

The G, Br, and NADDR fields remain internal to the CROM and these signals are not available to the CPU. The G field equal to zero is a No-Op and causes no action during this microinstruction. A Br field equal to X'0A' (IOL) translates bit positions EBT, DBT, and CBT as zeros. Bit position BBT will equal the result of ADDing I register bits 11 and 6, then ORing the result with I_7 (I11·I6V17).

Bit position ABT will equate to I_6. This result (00010) is OR'ed with the contents of the NADDR field to select the CROM address of the next microinstruction on the sequence.

NADDR	0 0 0 1	1 0 1 0
Br Result		0 0 0 1 0
Next CROM ADD	0 0 0 1	1 0 1 1 0 = X'036'

The remaining fields of the classify microinstruction (R/C, B-CNTL, FUNCT, Rs and Rd) generate the IPDH+ through IPD0+ outputs form the CROM/RALU. (Refer to Figure 3-2, CROM/RALU Interface CROM Address X'028'.)

The outputs on the IPDX-Bus represent the binary outputs of the lower 18 bits of microcode at CROM location X'028'. IPD0 through IPD3 are the Rd designators, IPD4 through IPD7 are the Rs designators, IPD8 through IPDB indicates the function code (X'6'), IPDC through IPDG (X'1A') specify the Bus Control and IPDH is the write control (R/C) bit ('0' = no write into destination register).

IPD0 through IPD7 are gated from the CROM internal microcoded ROM to a microinstruction I latch, internal to the RALU, where they select the destination and source register (when applicable) within the RALU register bank. Since the R/C-bit (IPD14), in this microinstruction, is a '0' no write into Rd will occur.

IPD8 through IPDB from the CROM microcoded ROM, along with IPD0 through IPD7 (Rs and Rd) specify the microcode function field. In this example microinstruction, the function field translates to W → Rd (X'600' in bits IPD0 through IPDB). However, since IPDH+ (R/C) is a '0', all data is inhibited from being written into any destination register. This portion of the microinstruction is, therefore, treated as a No-Op.

The B-CNTL segment (IPDG through IPDC) decodes as X'1A' and represents RALU → I/O Add. This B-CNTL feature is used to output the processor instruction (the DTIR) to the M-Bus (M0 through MF). This field also generates the necessary gating and clock terms to output the device address code of the peripheral device being selected. The device select code is gated to the Out-Bus lines.

The gating terms generated by the B-CNTL decode are a result of translations performed by the flip-flop latch arrangement at Z7B and the decode of that output by the decoders at Z8B/Z8C (refer to Figure 3-3, B-CNTL Generation).

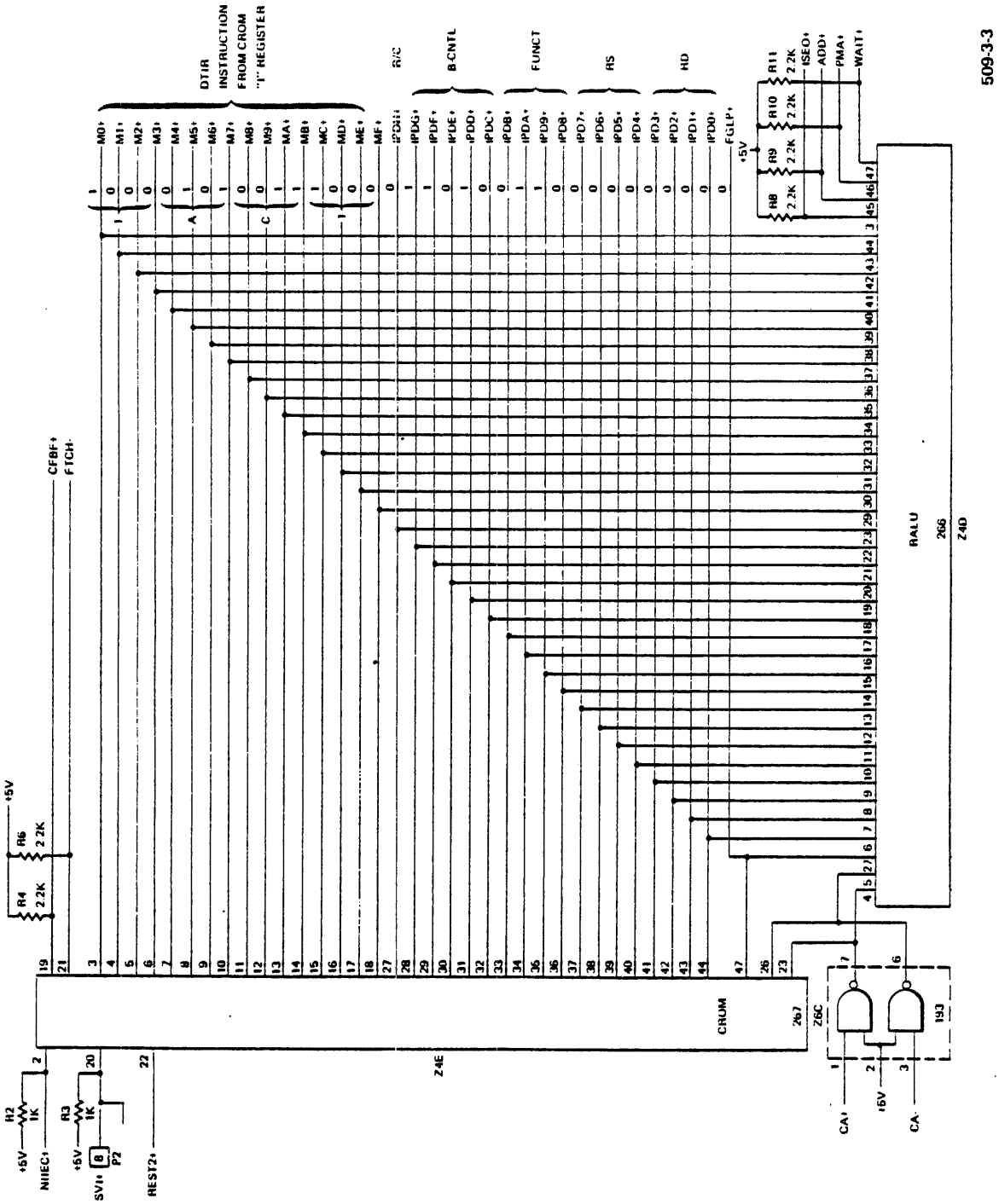


Figure 3-2. CROM/RALU Interface CROM Address X'028'
(90C02422 Sheet 2)

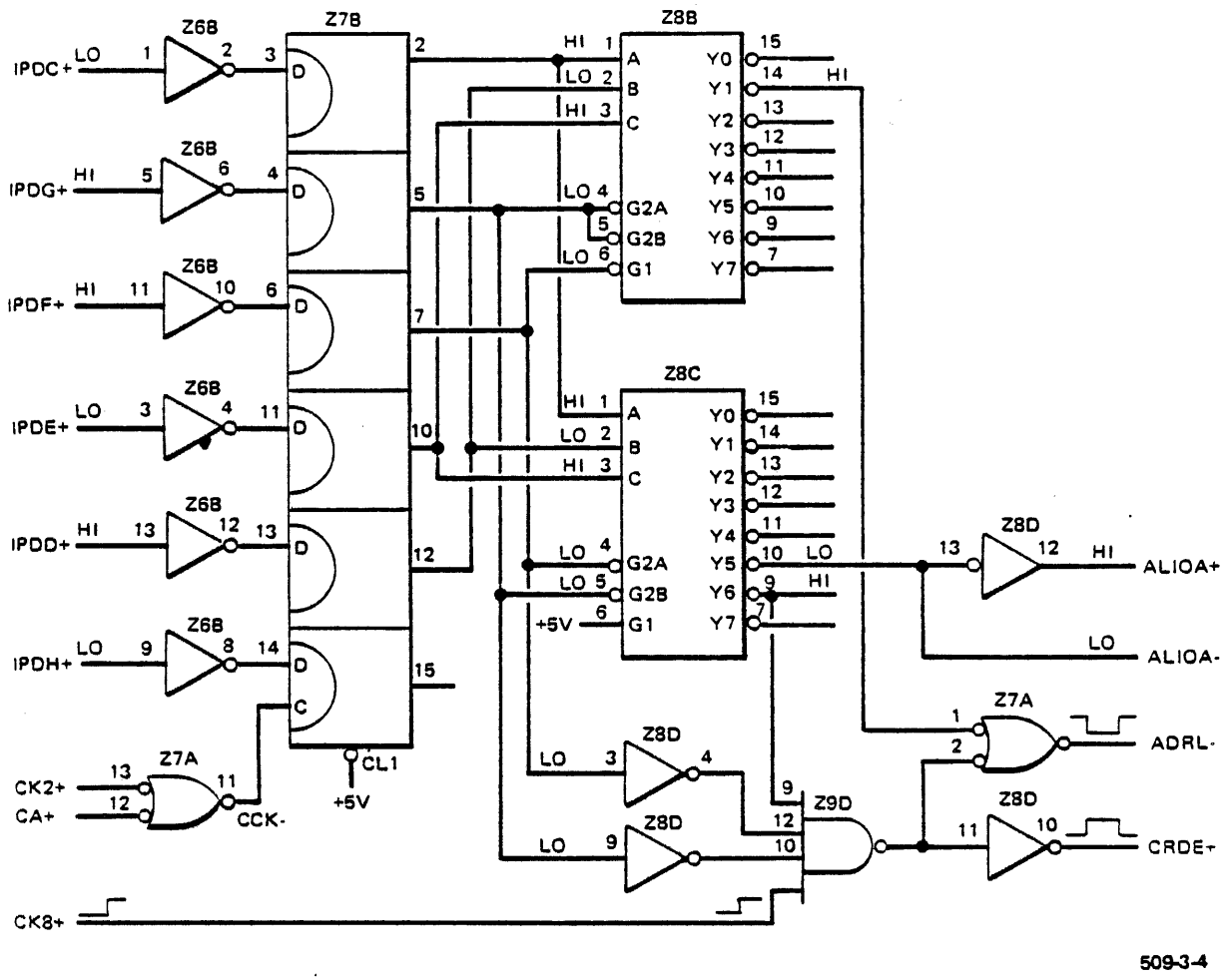


Figure 3-3. B-CNTL Generation CROM Address X'028'
(90C02422 Sheet 6)

During the execution of this first microinstruction, signals Address Load Enable (ADRL-) and CROM/RALU Drive Enable (CRDE+) gate the M-Bus outputs of the CROM I register to the TA-Bus lines (TAXX-). (Refer to Figure 3-4, M-Bus to TA-Bus Gating.)

The TA-Bus contents, representing the actual program instruction, (DTIR, B X'21') X'1CA1', is then gated by the I/O busy signal (IOBSY-) to the Out-Bus at the end of the current microcycle. (See Figure 3-5, TA-Bus to Out-Bus Gating.)

This output allows the selected peripheral device controller to be addressed when CPU1 produces the FAP- to the controllers.

The timing for the microcycle is shown in Figure 3-6, CLSFY Microcycle Timing, CROM Address X'02B'.

3.2.2 MICROINSTRUCTIONS AT X'036' AND X'032'

The second and third microinstructions in the sequence (located at CROM address X'036' and X'032') are, in themselves, NO-OPS. These two microcycles are necessary for timing and allow generation of the signals required in gating data, from the peripheral device, to the controller and from the controller onto the IN-Bus.

The timing and signal interchange, for the two NO-OP instructions, is shown in Figure 3-7. Timing: No-Op Microcycles. IOBSY- remains low throughout the two NO-Op microcycles to enable the data being gated from the addressed peripheral to propagate through the controller to the IN-Bus without danger of interference from another device attempting to use the I/O Bus. Signal FAP- goes true to allow the selected controller to lock onto the bus network. Signal READ- designates the direction of data transfer (into the CPU) thereby, identifying the program instruction as a DTIR.

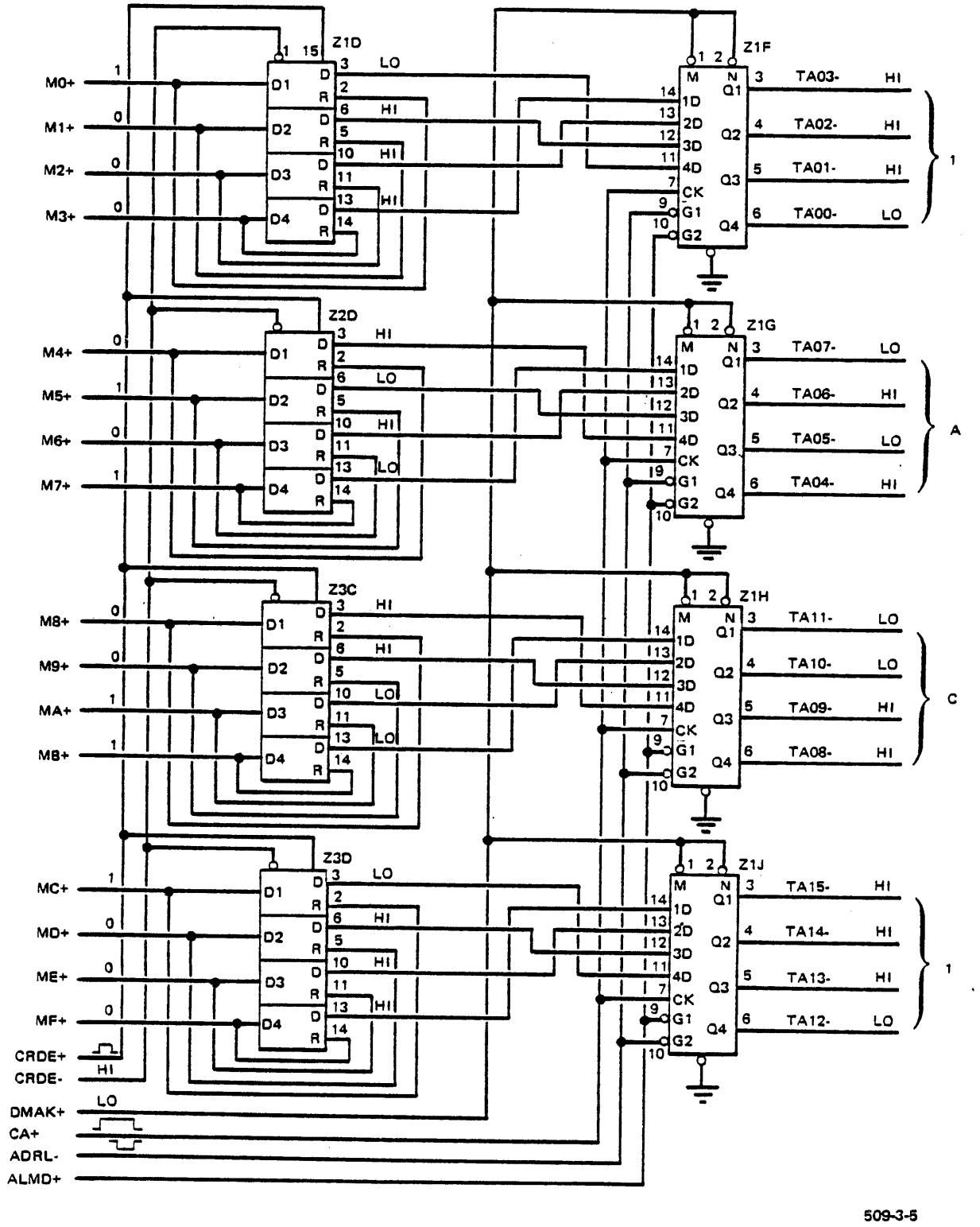


Figure 3-4. M-Bus to TA-Bus Gating
(90C02422 Sheets 4 and 5)

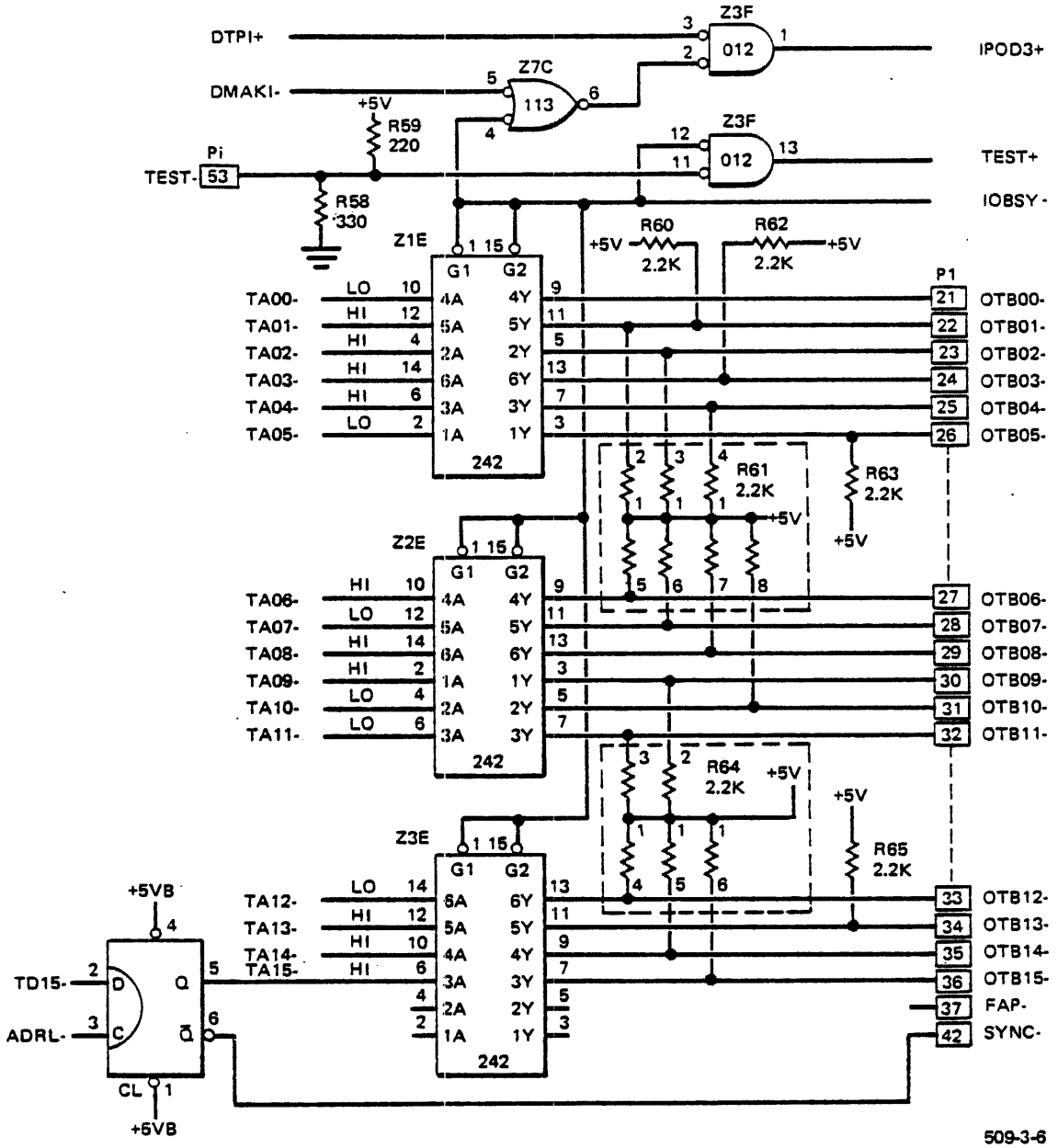


Figure 3-5. TA-Bus to Out-Bus Gating CROM Address X'028'
(90C02422 Sheet 3)

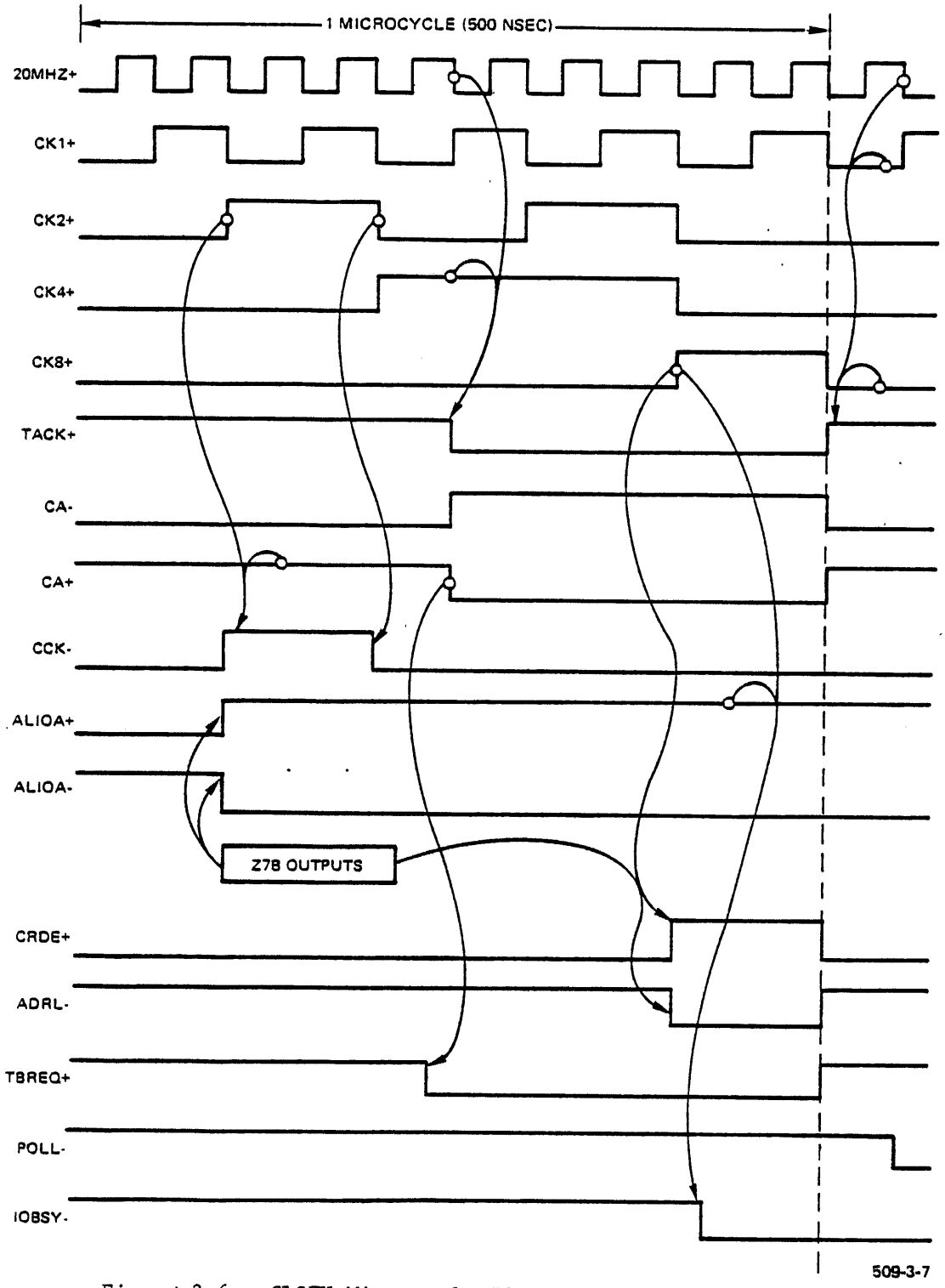


Figure 3-6. CLSFY Microcycle Timing CROM Address X'028'

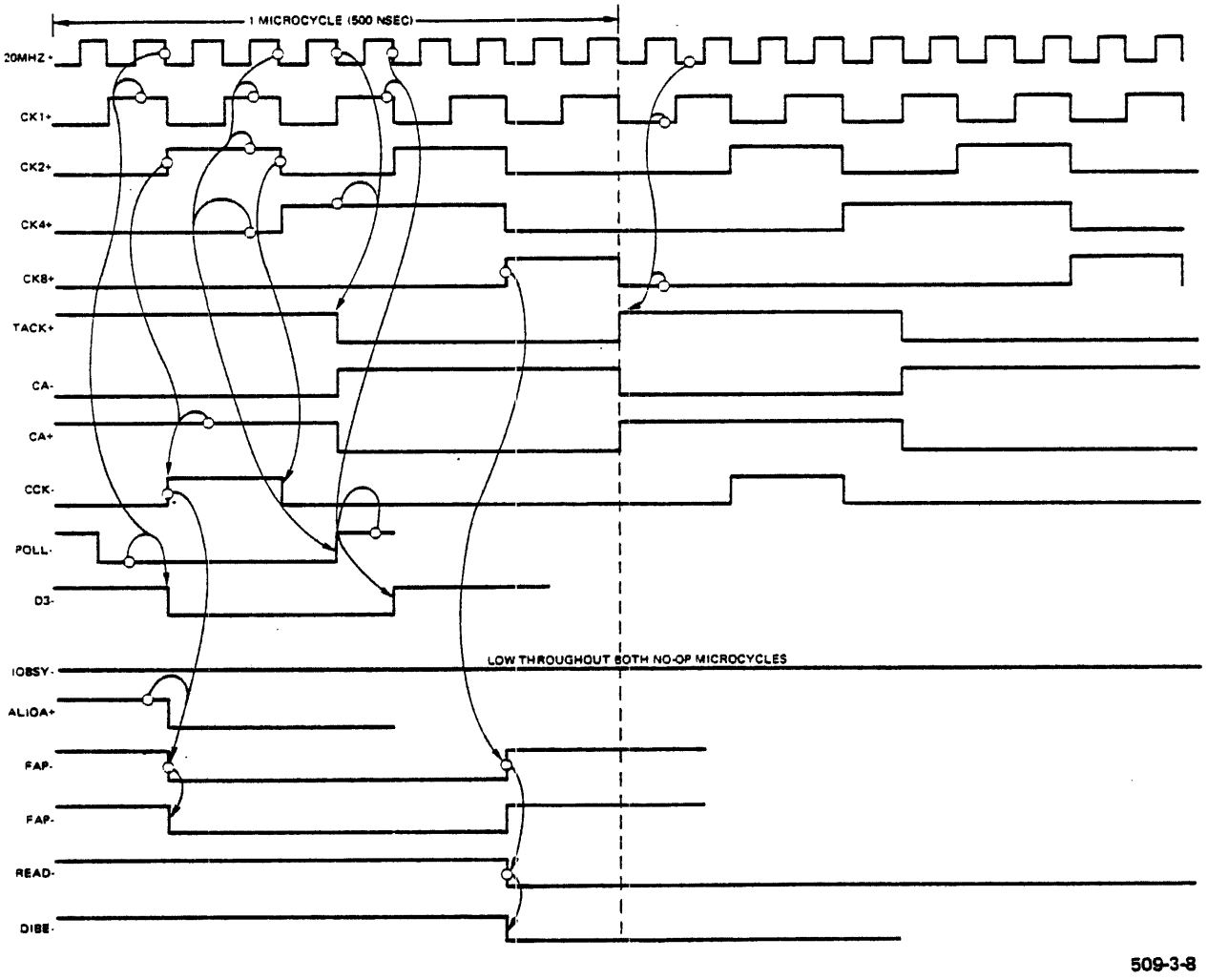


Figure 3-7. No-Op Microcycles

Signal DIBE- is driven true by signal I/O READ and remains true for the duration of the read operation. DIBE enables the IN-Bus lines to the CPU TDXX lines. Reference Figure 3-8 In-Bus to TD-Bus, Typical Gating.

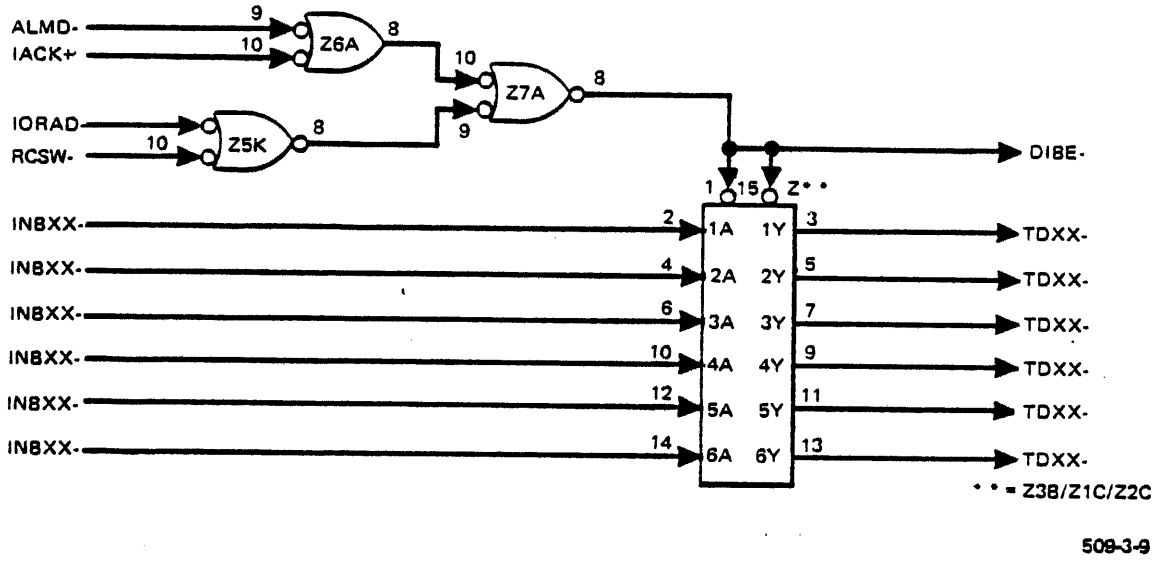


Figure 3-8. In-Bus to TD-Bus, Typical Gating
(90C02422 Sheets 4 and 5)

3.2.3 MICROINSTRUCTION AT X'10F'

The fourth microinstruction in the series, at CROM location X'10F', (X' 3 4007 3B90'), transfers the I/O Data to the destination register and decodes as:

	G	Br	NADDR	R/C	B-CNTL	FUNCT	Rs	Rd
Binary	110	1 0000	0000 0001	1	1 0011	1011	1001	0000
Hex	6	10	01	1	13	B	9	0

The G-field now equal to X'6', causes the Rd portion of the CROM microcode to be replaced with I register bits 10, 9, and 8 (I10, I9, I8 → Rd). This enables selection of the destination register, specified by the program instruction, currently being held in the CROM I register. In this example DTIR I register bits I10, I9, and I8 equal 100. This binary code, when placed in the Rd field, selects the B register (as the destination register) for the incoming I/O data. The G-field translation also forces the foreground bit (F) to the most-significant bit position of the destination register ($F \rightarrow R_{d15}$).

The B_r field (X'10') sets the least-significant bit of the Br code to a logical "1" and forces all other bits of the Br to zero. The Br value OR's with the NADDR value (X"01") to force the next CROM microinstruction address of X'003'.

NADDR	0000	0001
Br result		00001
Next CROM add	0000	00011 = X'003'

The remaining bits of the microinstruction (R/C, B-CNTL, FUNCT, Rs and Rd) generate the IPDH+ through IPD0+ outputs of the CROM/RALU as shown in Figure 3-9, CROM/RALU Interface CROM address X'10F'.

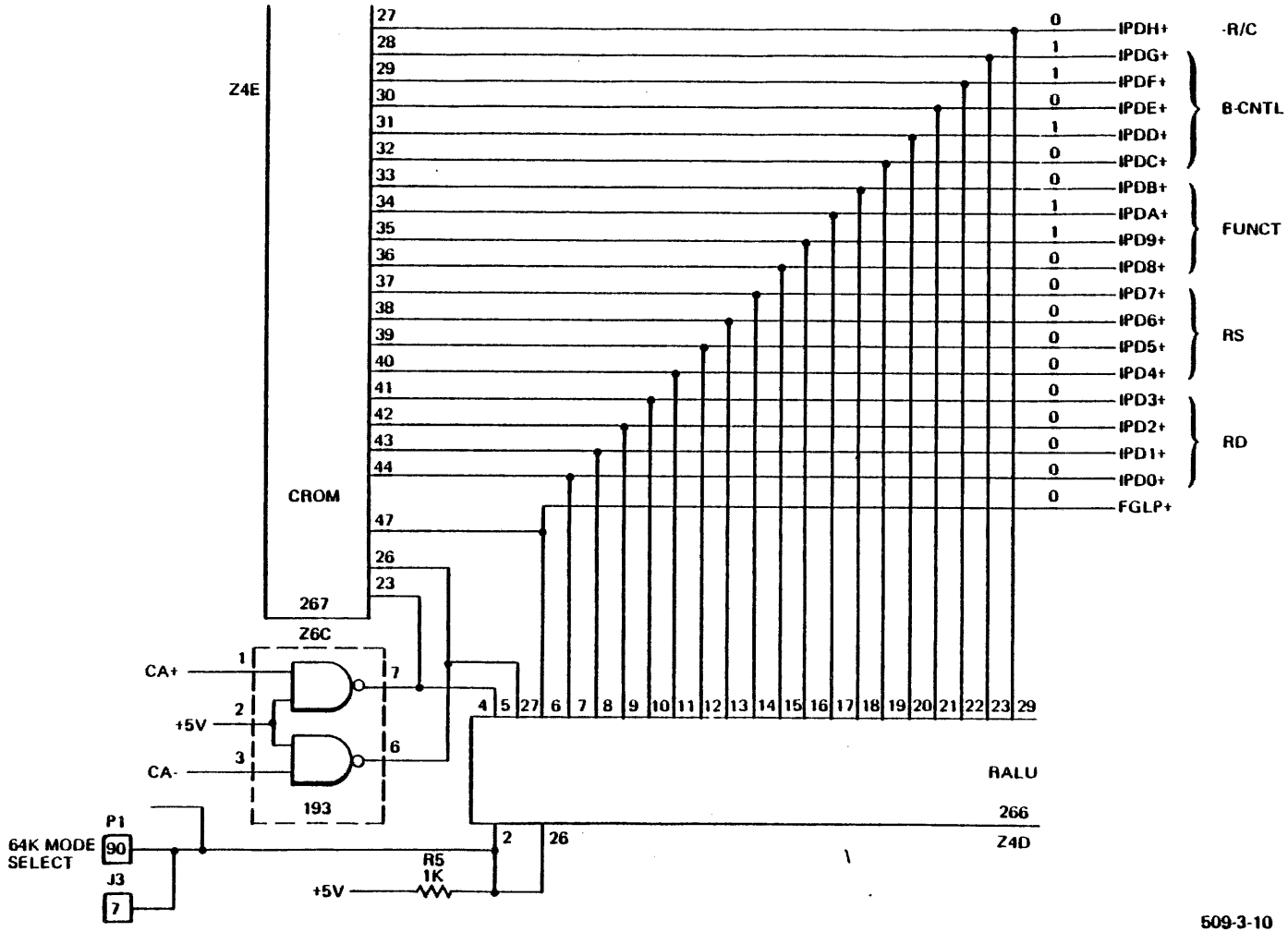


Figure 3-9. CROM/RALU Interface, CROM Address X'10F'
(90C02422 Sheet 2)

Signals IPD0 through IPD7 are gated from the CROM to a microinstruction I latch (internal to the RALU) to select the destination and source registers (when applicable) within the RALU register bank. However, due to the translation of the G field of this microinstruction, bits I10, I9 and I8 instead of IPD0 → IPD2 are used to select the Rd. I10, I9 and I8 equaling 100 select the B register as the destination register. These three bits are gated from the CROM to the RALU via lines MA, M9 and M8, respectively.

The B-CNTL Field (equal to X'13') specifies an I/O Data → RALU operation and allows the gating of the I/O data (from In-Bus) to the RALU register inputs. Since the R/C field is set to a '1', the RALU will be enabled to accept this incoming data and place it into Rd (the B register). The necessary gating signals (IODAL- and CRRE-) are generated as shown in Figure 3-10, B-CNTL generation, CROM Address X'10F'.

The timing for this microcycle is shown in Figure 3-11; DTIP Microcycle timing.

With signals CRRE- true (low) and CRDE+ false (low), the incoming data on In-Bus, will be gated to the RALU destination register. This data enable (over M-Bus lines) is allowed by gating as shown in Figure 3-12, M0 through MF In-Bus gating. Only with both enables (CRDE+ and CRRE-) low, will data be input from the TD-Bus to the M-Bus.

The Function field (equal to X'B') places this microinstruction in the general category of Miscellaneous (MSC) instructions. Therefore, with the Rs field equal to 9 and the Rd field equal to zero, the function M → Rd is identified. (Refer to Table 3-5, Function Field General Classification and Table 3-8, Miscellaneous Group Operation.)

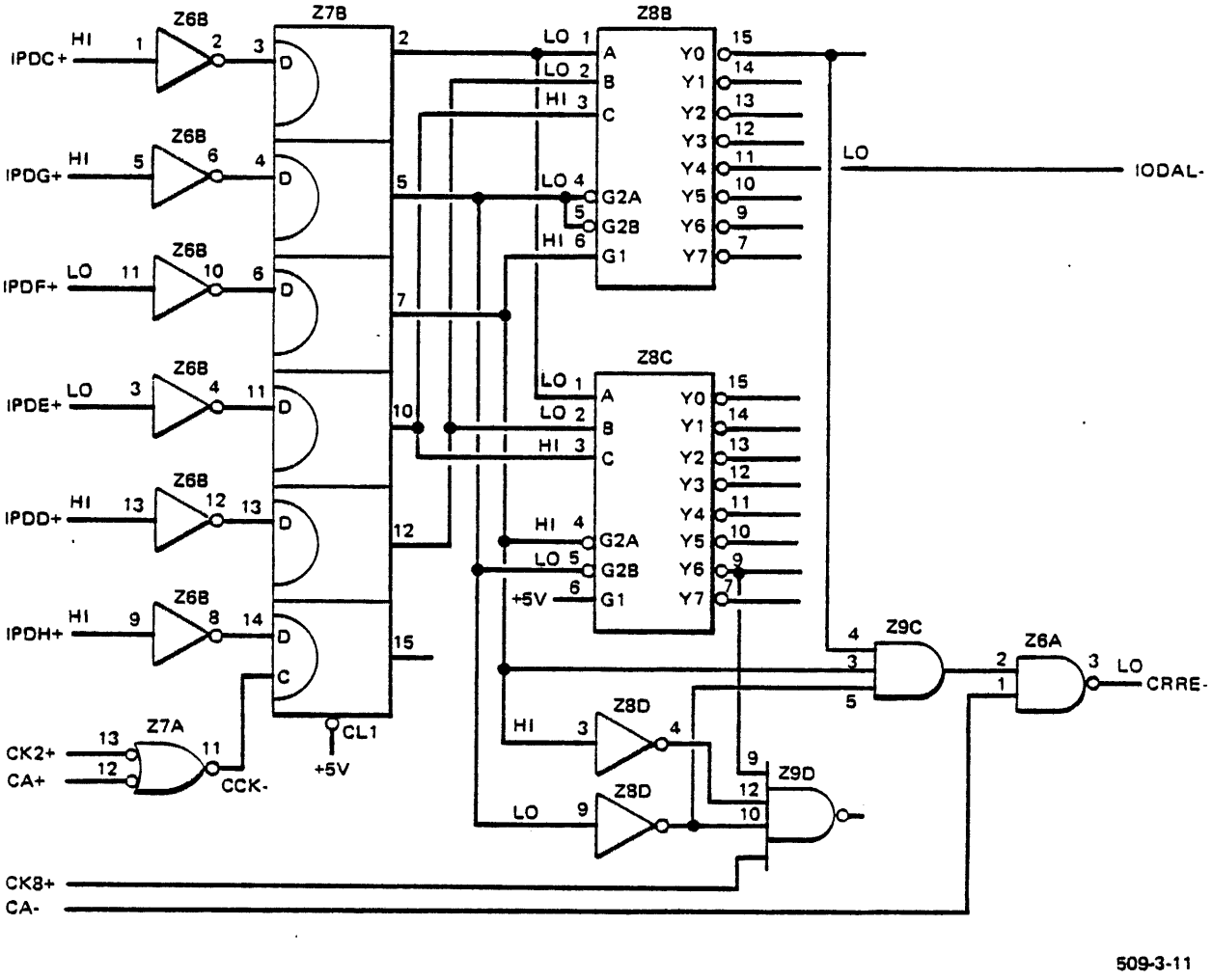


Figure 3-10. B-CNTL Generation, CROM Address X'10F'
(90C02422 Sheet 6)

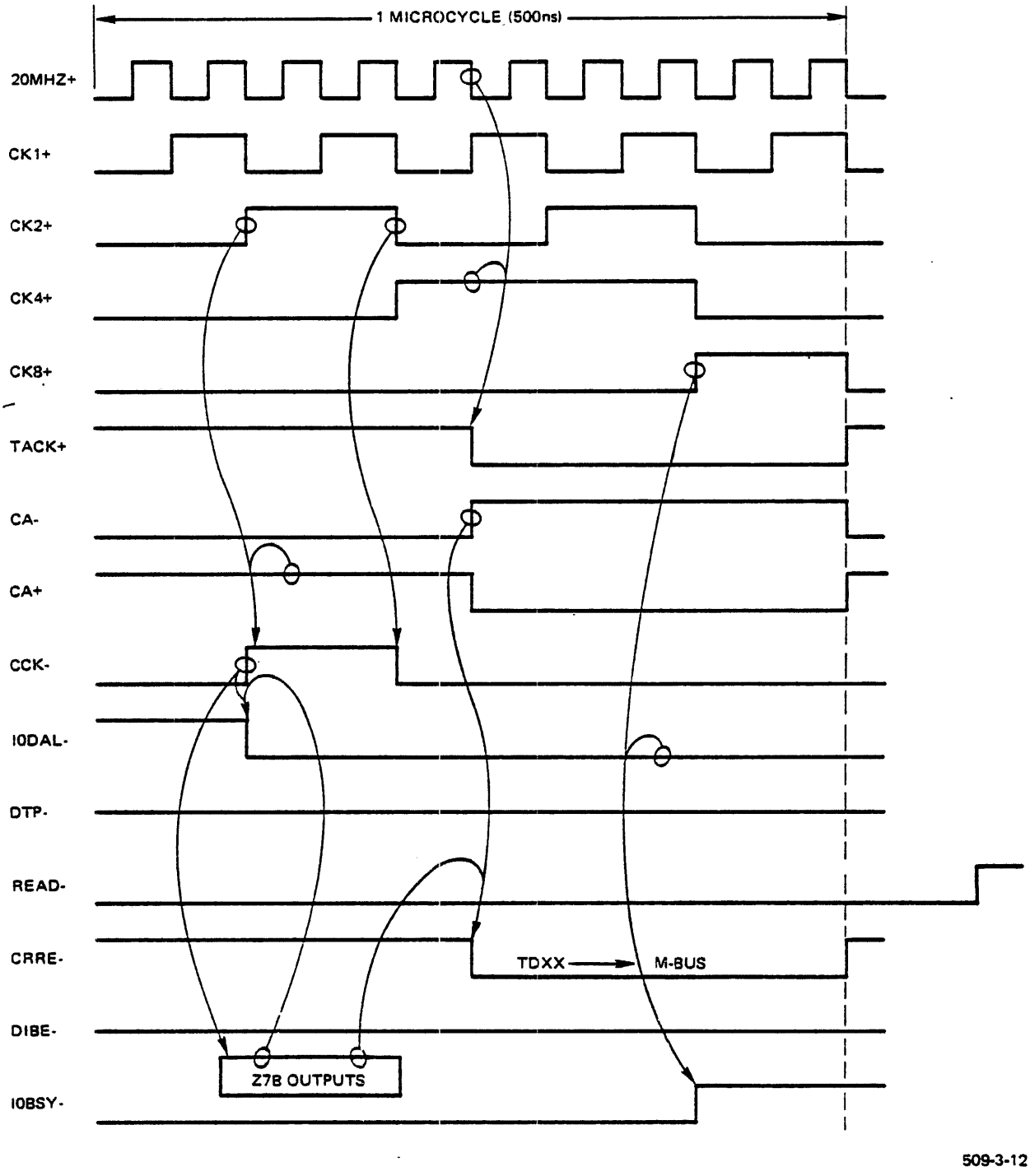


Figure 3-11. DTIR Microcycle Timing

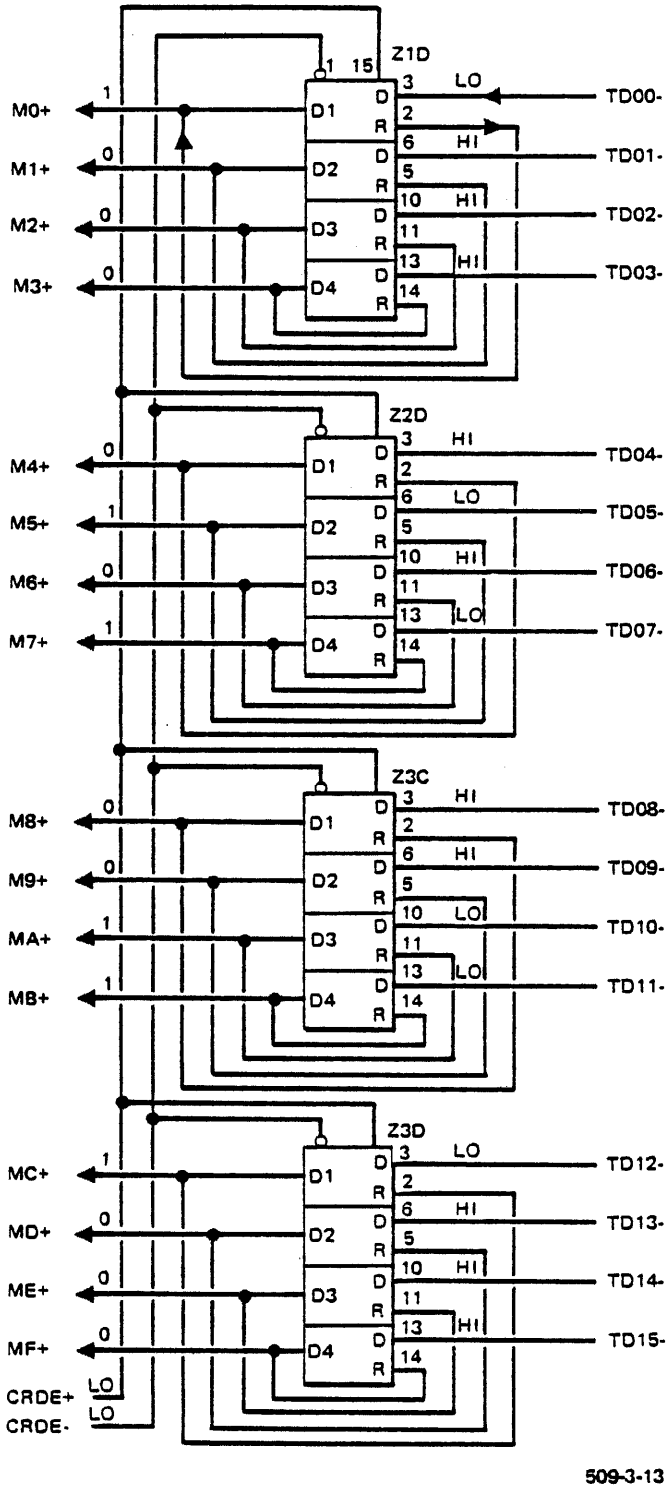


Figure 3-12. M0 through MF In-Bus Gating CROM Address X'10F' (90C02422 Sheets 5 and 6)

3.2.4 MICROINSTRUCTION AT X'003'

The next microinstruction executed is read from CROM address X'003' and translates to X' 0 6443 8E62' (Mnemonic INCP+1). (See Figure 3-13, CROM/RALU Interface, CROM Address X'003').

	G	Br	NADDR	R/C	B-CNTL	FUNCT	Rs	Rd
Binary	000	1 1001	0001 0000	1	1 1000	1110	0110	0010
Hex		19	10	1	18	E	6	2

This microinstruction is the end of the DTIR sequence and the beginning of the instruction fetch (IFTCH) cycle to obtain the next CPU instruction from the CPU memory. The G-field, equal to zero, translates as a NO-OP. The Br-field (X'19') equates CBT to the state of the Save I Switch (SVI), BBT to the condition of the Run Switch (if RUN then BBT = 1), ABT to the interrupt indicator (if INT then ABT = 4) and forces EBT and DBT to zeros. This Br-field with no interrupts, the CPU in the run mode, and no save I indication, OR's with the NADDR field to generate the next CROM address of X'021':

```
NADDR          0001 0 000
Br Result           0 0001
Next CROM Add  0001 0 0001 = X'021'
```

The B-CNTL Field (Reference Table 3-4. Bus Control),(X'18') translates to RALU → MA, to begin a memory cycle. This memory cycle fetches the next program instruction from CPU memory. The next program instruction address, for this example, is decoded as P + 1 due to the Funct Field (X'E') combining with Rs (X'6') and Rd (X'2') which indicates that this microinstruction is in the SRL group performing the "P+1 → P" function. The P register will accept the new value gated to it, since the read/write control (R/C) is a logic '1'.

The B-CNTL field generates the terms ALMA- (RALU → Memory Address), CRDE+, and ADRL- to begin the instruction fetch cycle. (Reference Figure 3-14, B-CNTL Generation, CROM Address X'003'.)

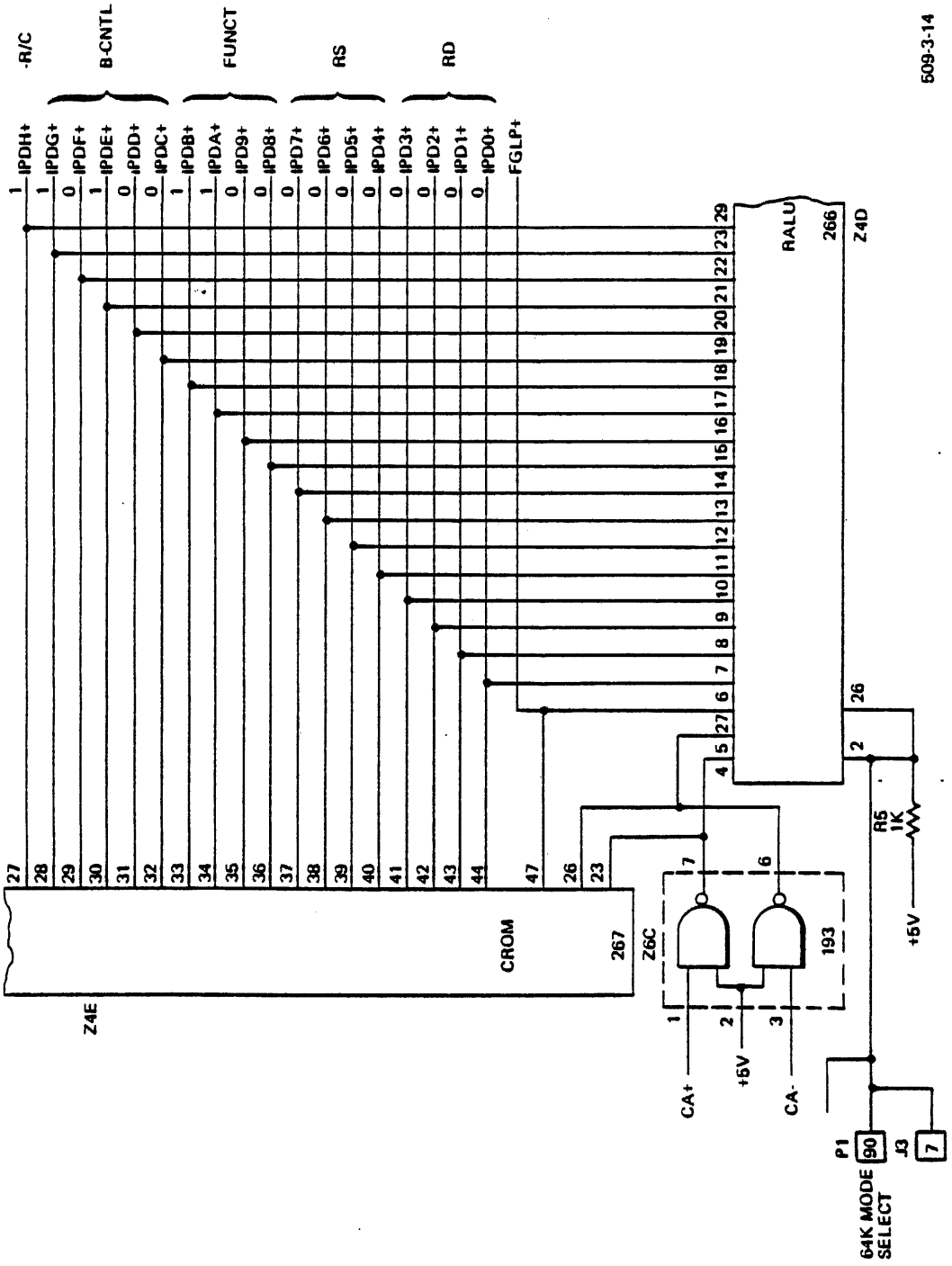


Figure 3-13. CROM/RALU Interface CROM Address X'003'
(90C02422 Sheet 2)

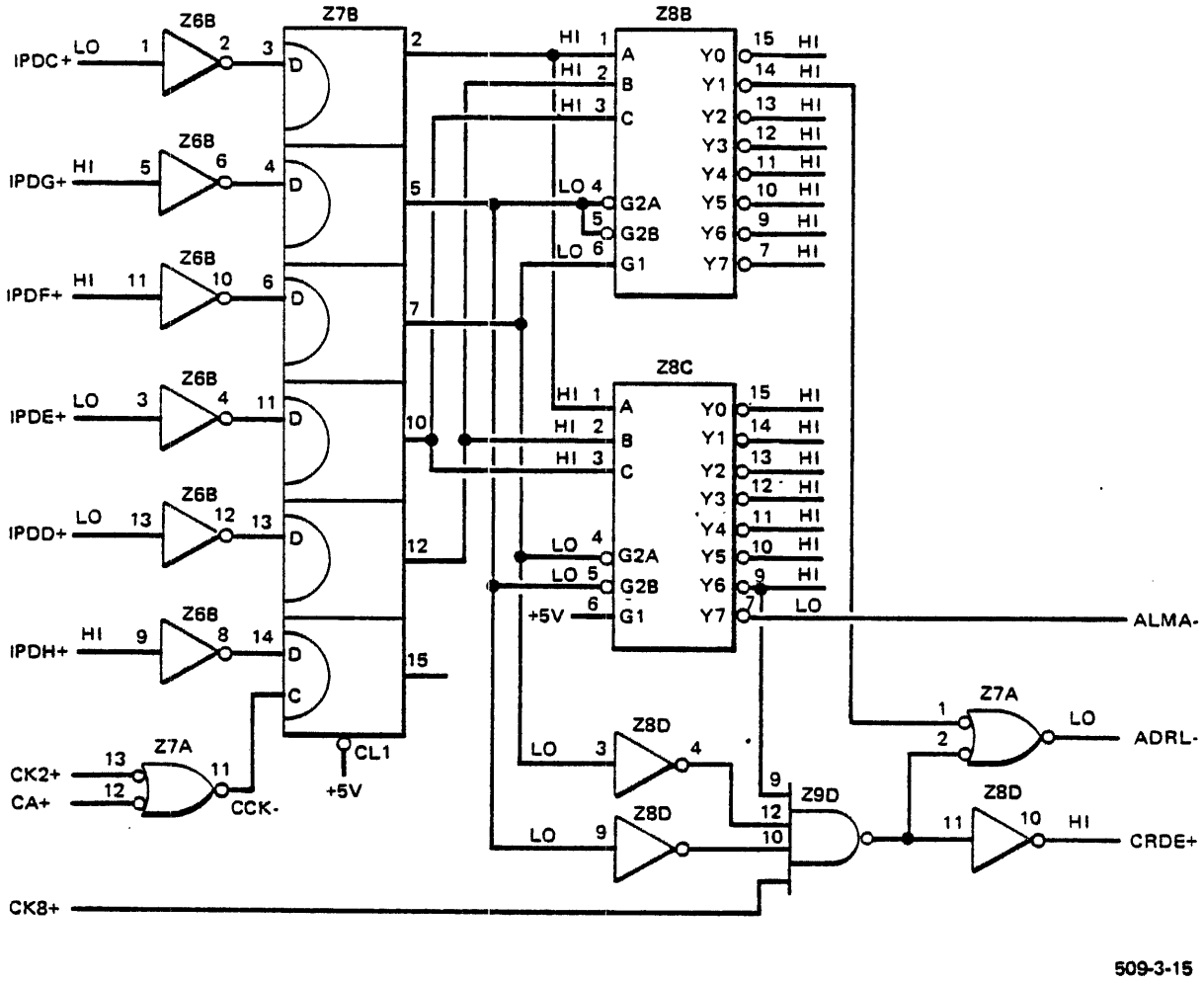


Figure 3-14. B-CNTL Generation, CROM Address X'003'
(90C02422 Sheet 6)

88A00509A-B

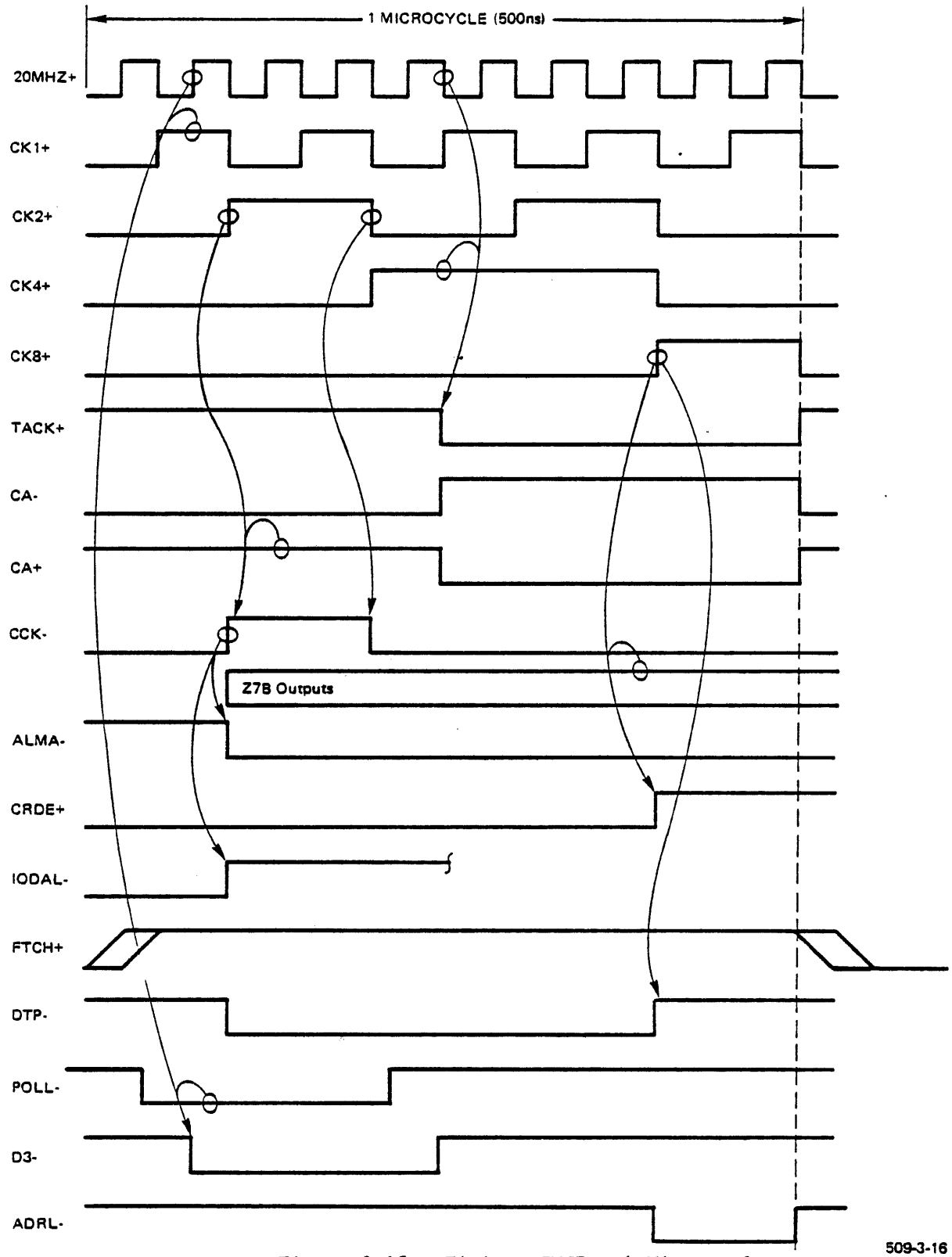


Figure 3-15. Timing, INCP + 1 Microcycle

All timing, for the INCP +1 microcycle, is shown in Figure 3-15. ALMA- goes true to condition a memory request, which will be generated at the beginning of the next microcycle. The term FTCH+ becomes active (from the CROM) to indicate a processor Fetch cycle has been entered. Fetch also is transmitted to the CPU-2 module (90C02429 Sheet 8) to clear I/O Reset (IOR) to the peripheral controllers (SYRT - 90C02429 Sheets 8 and 9) as shown in Figure 3-16, SYRT- Generation.

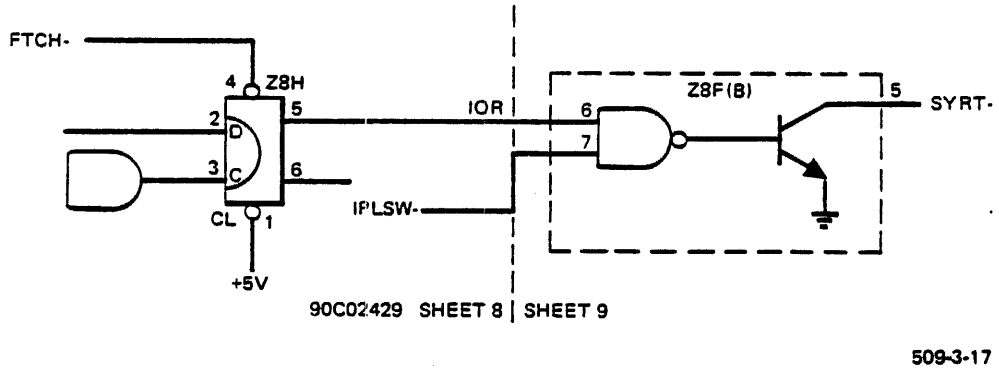


Figure 3-16. SYRT- Generation

Signal DTP- is transferred to the controllers for I/O operations, along with D3- and POLL-. They synchronize interrupt and DMA operations. CRDE- is generated, near the end of the microcycle, and gates the new program address to the memory address (TA) lines. The value of P (P+1) is determined and loaded into the P register entirely within the CROM/RALU chipset.

The P → MA gating is shown in Figure 3-17 (P → Memory Address Gating).

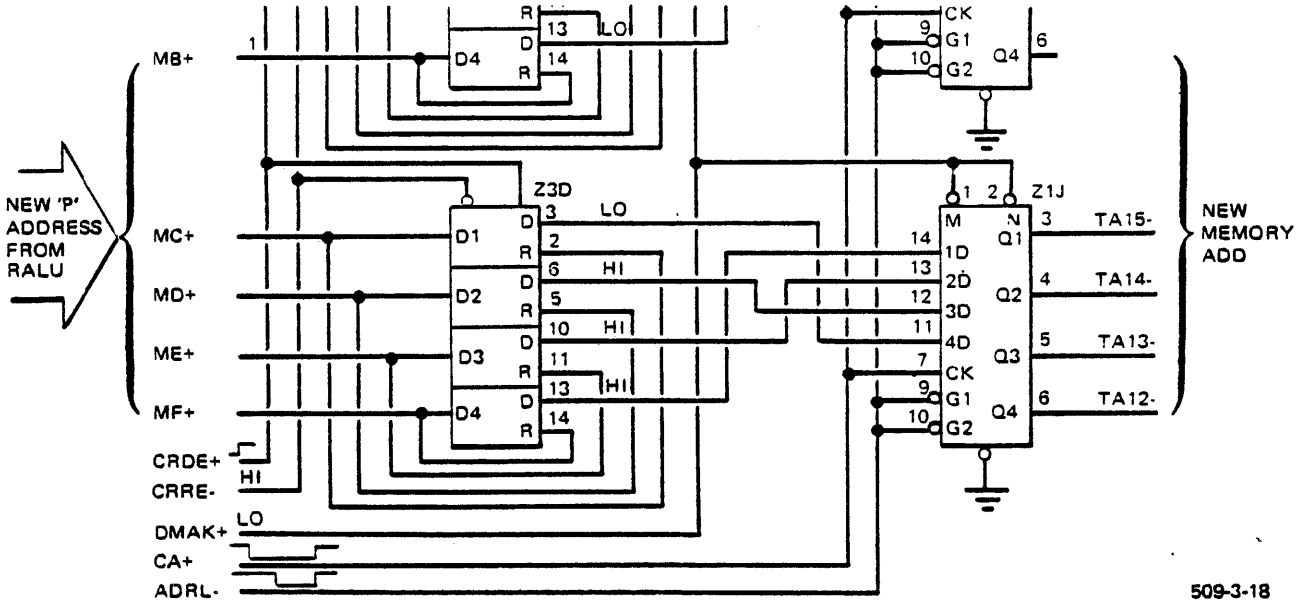


Figure 3-17. P → Memory Address Gating (Partial Representation)
(90C02422 Sheets 5 and 6)

The final instruction of the DTIR microinstruction sequence (at CROM location X'021') is actually the first full microinstruction for the transfer of the next program instruction from CPU memory to the CROM I register. This microinstruction, with the mnemonic "FTCH", is represented by X' 0 6C53 0E61' and decodes as:

	B	Br	NADDR	R/C	B-CNTL	FUNCT	Rs	Rd
Binary	000	1 1011	0001 0100	1 1	0000	1110	0110	0001
Hex	0	1B	14	1	10	E	6	1

As previously described, the G-field, being zero, indicates a No-OP function. The Br-field is translated to indicate a Classify Branch (X'1B'). By the interpretation of the classify branch operation, the next CROM address is dependent upon the output of the instruction register (I) that is being loaded with the new program instruction. This I register output determines (classifies) the type of program instruction to be executed and will direct the CROM to the next microinstruction necessary for the proper sequence of operations. For a further explanation of the calculation of the next CROM address on a Classify Branch operation, refer to Section 3.1.2.

The R/C, B-CNTL, FUNCT, Rs and Rd fields generate the bit/signal outputs of the CROM/RALU as indicated in Figure 3-18. (CROM/RALU Interface, CROM Address X'021'.)

The B-CNTL field (X'10') calls for the transmission of the data on the memory data bus (MD-Bus) to the RALU (MD → RALU) by generation of the terms MDAL, ALMA and CRRE, as shown in Figure 3-19, B-CNTL Generation, CROM Address X'021'.

All timing for the microcycle is shown in Figure 3-20, Timing, FTCH Microcycle.

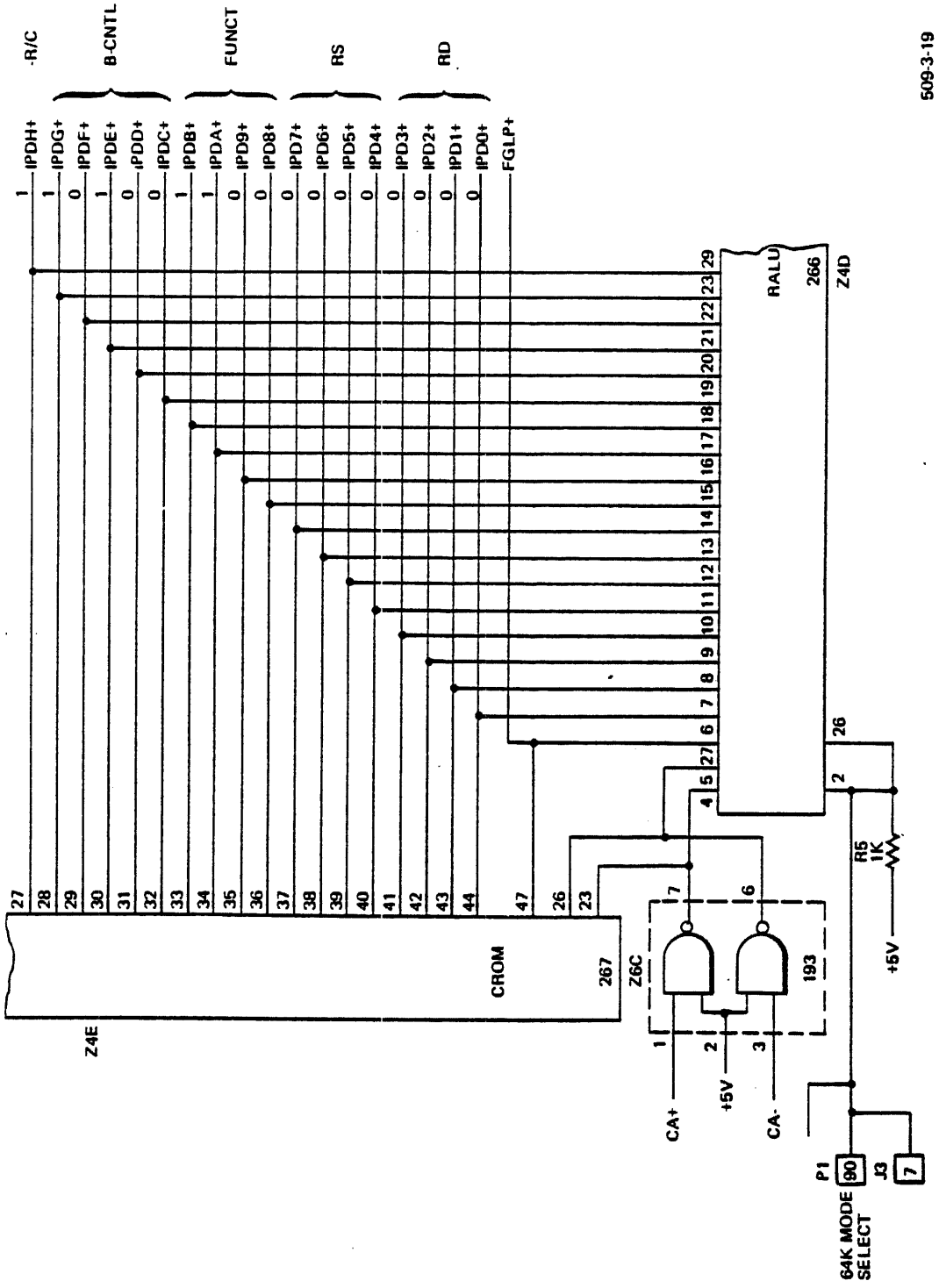


Figure 3-18. CROM/RALU Interface CROM Address X'021'
(90C02422 Sheet 2)

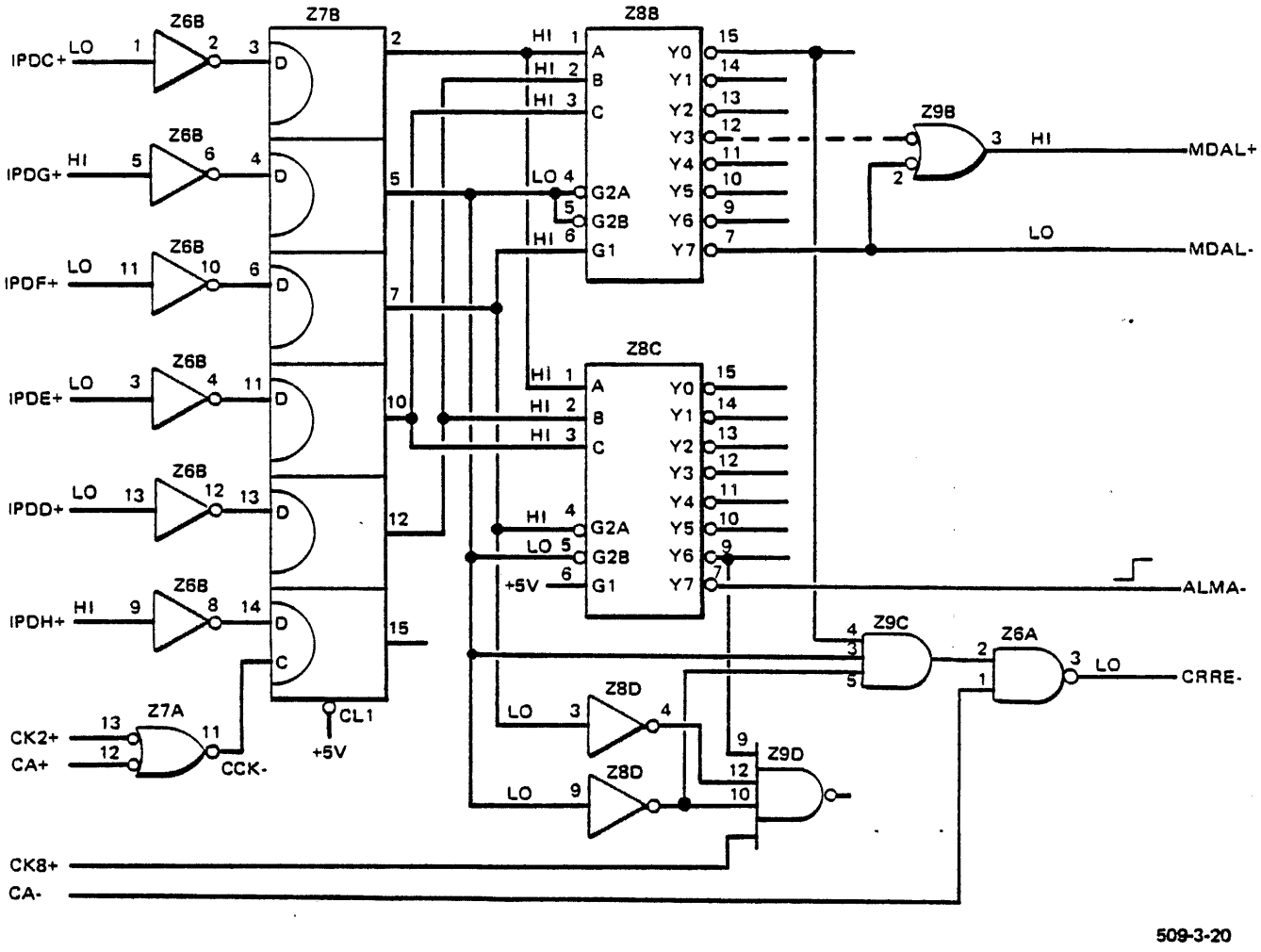


Figure 3-19. B-CNTL Generation, CROM Address X'021'
(90C02422 Sheet 6)

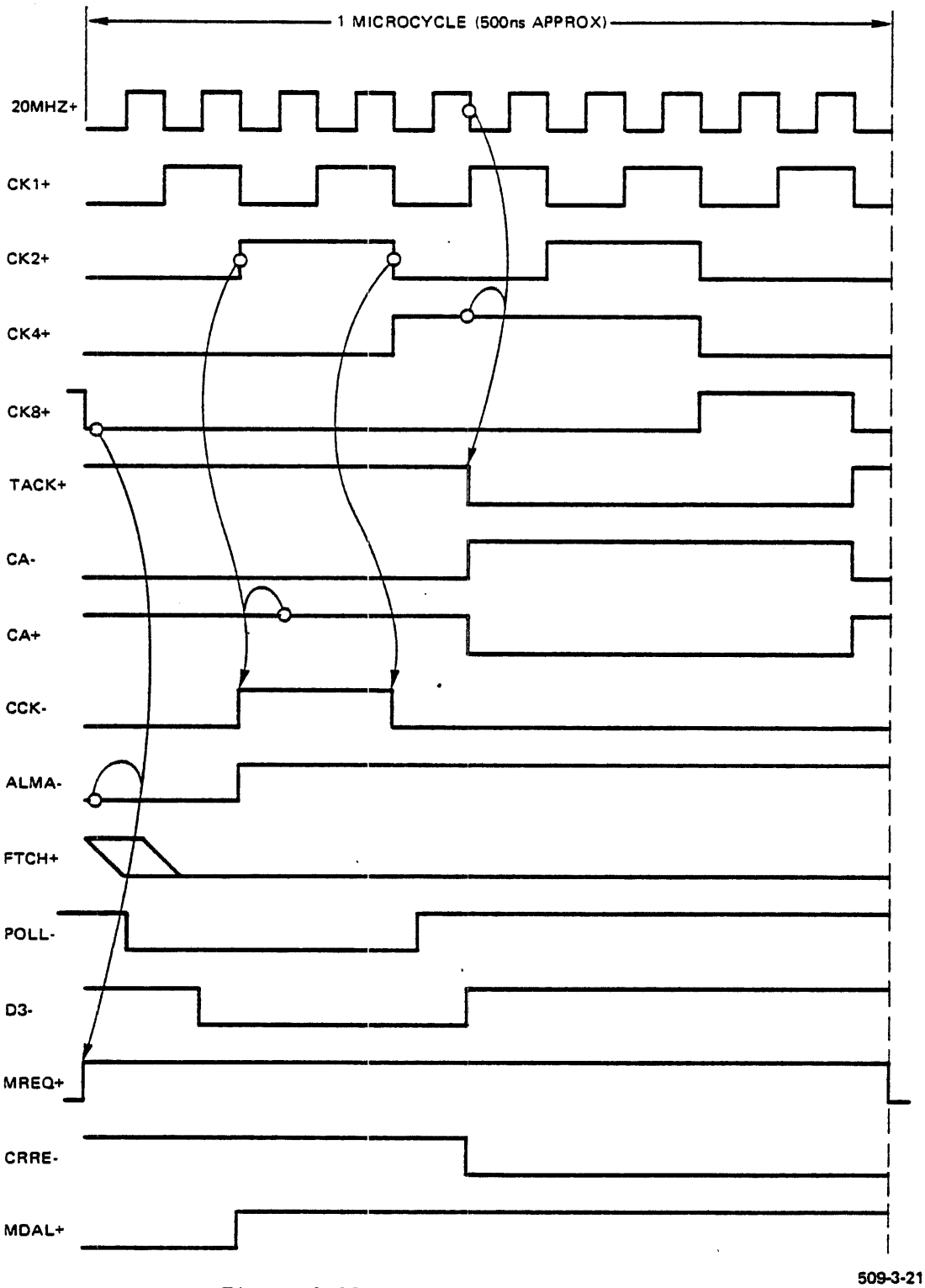


Figure 3-20. Timing, FTCH Microcycle

The memory cycle, to read the next program instruction, is begun by the generation of the memory request signal (MREQ) at the beginning of this microcycle. This signal is generated by ALMA- being true from the last microcycle and the negative going transition of CK8+. The logic for this signal is shown in Figure 3-21, MREQ.

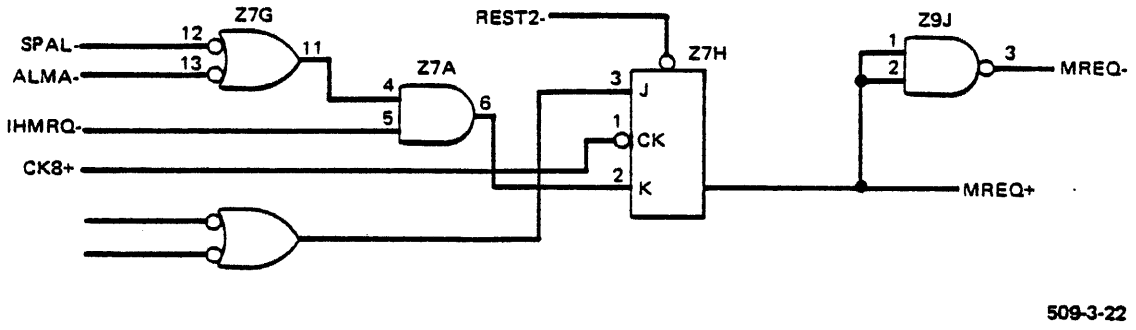


Figure 3-21. MREQ
(90C02422 Sheet 8)

Once the memory request is generated, memory data enters the CPU as MDXX and is transferred via the TD Bus (90C02422 Sheets 4 and 5). Once internal to the processor, the new instruction is gated over the bi-directional TD-Bus to the M-Bus (M0 through MF) and into the selected registers within the CROM/RALU chipset. Reference Figure 3-22 TD to M-Bus Gating.

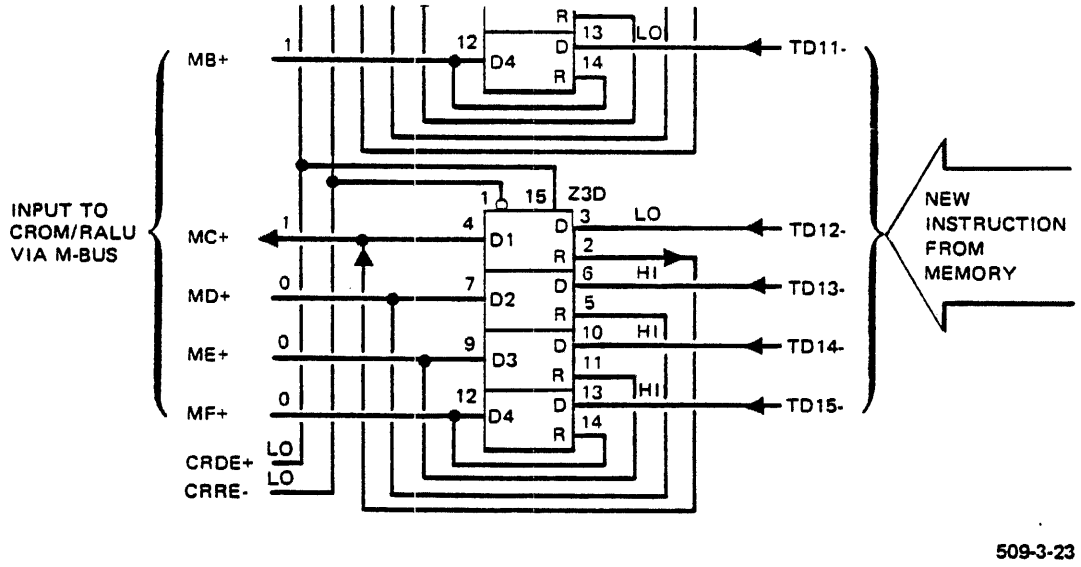
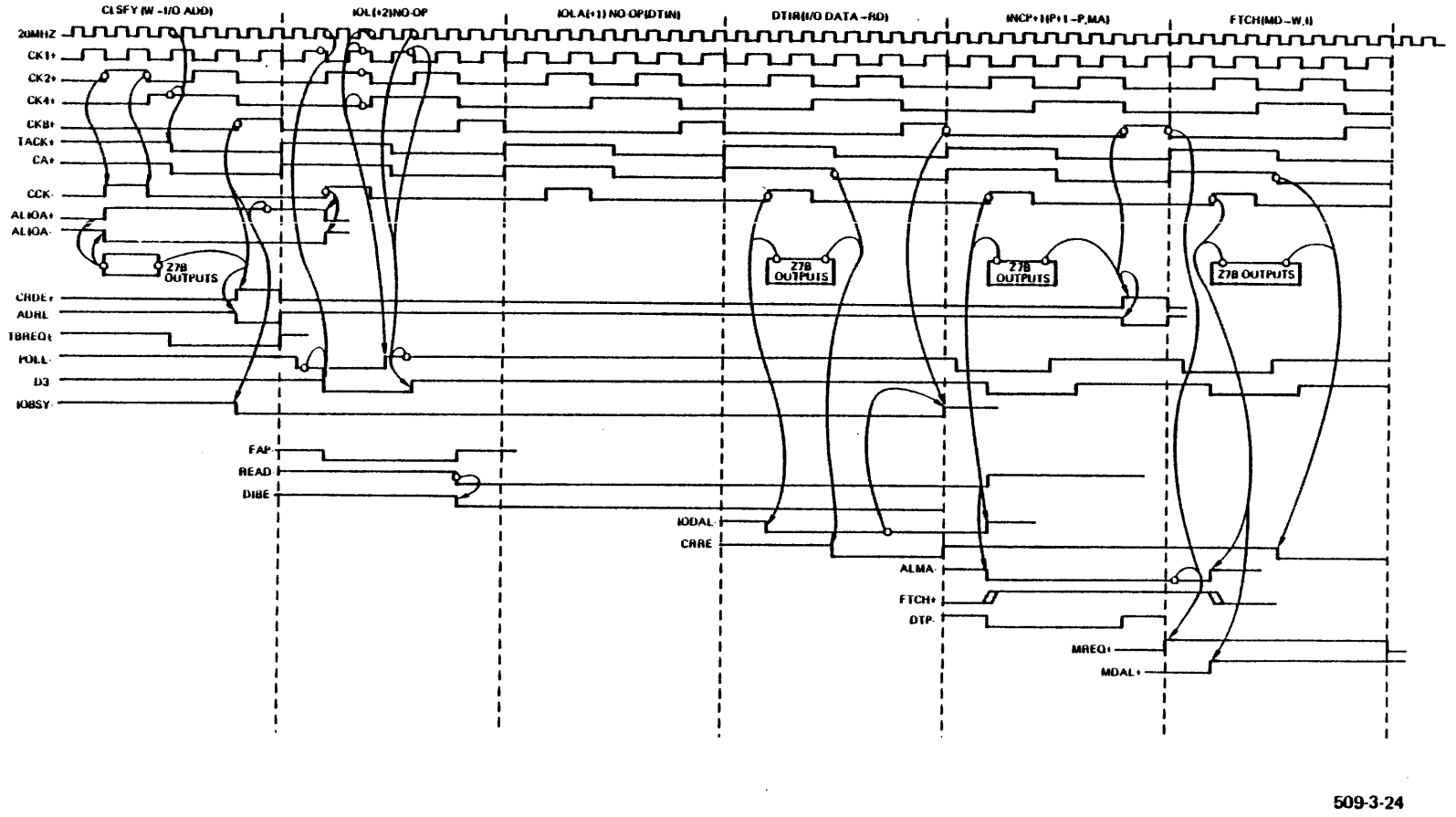


Figure 3-22. TD to M-Bus Gating (Partial Representation)
(90C02422 Sheets 4 and 5)

The Funct, Rs and Rd fields combine as an SRL Class of instruction, (X"E61") to gate the new program instruction into the W register for possible address modification and into the CROM I register for interpretation/classification (M → W,I). All timing and gating to allow this data to be written into W and I is internal to the CROM/RALU Chipset.

The complete timing for the DTIR microinstruction sequence is shown in Figure 3-23, Overall DTIR Microinstruction Sequence Timing.



509-3-24

Figure 3-23. Overall DTIR Microinstruction Sequence Timing

3.3 MEMORIES

The 4K and 8K memory boards, assembly numbers 31D02410 and 31D2585, are available in the following configurations:

-01	4Kx16 Bits
-11	4Kx18 Bits
-21	8Kx16 Bits
-31	8Kx18 Bits

The 16K memory board, assembly number 31D02578A01, is supplied only in a 18-bit configuration.

The 32/64K memory board, assembly number 31D02644A, is available in the following configurations:

-nn	32Kx16 Bits	no parity
-nn	64Kx16 Bits	
-nn	32Kx18 Bits	byte parity
-nn	64Kx18 Bits	
-nn	32Kx22 Bits	includes optional error correction
-nn	64Kx22 Bits	

The basic 64K board is depopulated to provide the 32K memory. The 22-bit memory consists of a 16-bit memory board with an error correction (ECC) module mounted piggyback on the memory board. The ECC module provides 6 bits that increase the basic 16-bit word length to 22 bits. Error correction is obtained for single-bit errors; error detection is obtained for a word with more than one error. (Refer to Section 3.3.5 for the ECC module operation.)

The memory chips being utilized are 28-pin, 16K x 2 bit, dynamic RAMs providing an access time of 200 nanoseconds and a cycle time of 450 nanoseconds. Each memory incorporates a refresh circuit having priority over a read or write cycle. The refresh cycle refreshes a single row address every 32 µsec; therefore, all 64 rows are refreshed every 2 msec.

If parity circuitry is incorporated (18-bit memories), a parity bit is generated for both upper and lower bytes of the word on a write cycle and checked on a read cycle. If a parity error is detected, the Tri-State Data Bus (TD-Bus) is forced to all zeros for that cycle. A Data Error indicator is provided.

3.3.1 READ TIMING LIMITS, ALL MEMORY BOARDS (FIGURE 3-24, READ CYCLE TIMING)

During a read cycle, memory request (MREQ-) must remain low until memory ready (MRDY+) is sensed high, or a time frame of 2 µsec (maximum) has lapsed. Once off, MREQ- must remain off for a minimum of 50 µsec.

The address bus (TAXX-) must be stable for a minimum of 20 nanoseconds before the leading edge of MREQ- and must be maintained stable for at least 175 nanoseconds after MREQ- occurs.

The data bus (TDXX-) data, from memory, will be valid at the leading edge of MRDY+ and will remain valid until MREQ- goes false (high).

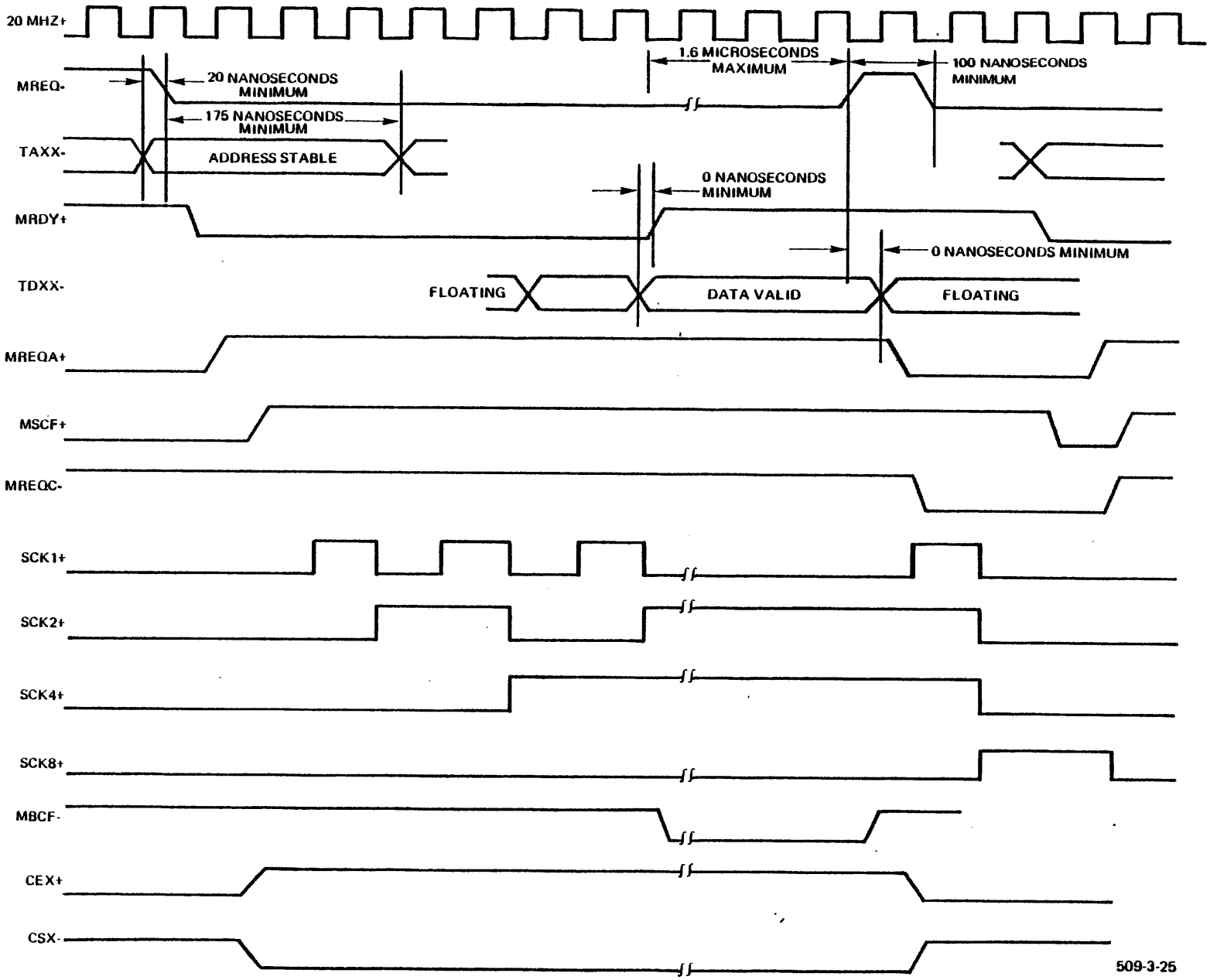


Figure 3-24. Read Cycle Timing

3.3.2 WRITE TIMING LIMITS, ALL MEMORY BOARDS

The write timing limits, not using Stop Read (STRD-), are shown in Figure 3-25 with cycle timing. The tolerances for MREQ- and TAXX- are the same as in Read Timing. Memory write (MRWE-) must go true within 175 nsec (max) after MREQ- goes low. MRWE- must go high concurrently with MREQ-. MRWE- enables the tri-state data bus (TDXX-) to gate into memory.

3.3.3 READ MODIFY WRITE (RMW) TIMING LIMITS, ALL MEMORY BOARDS

Figure 3-26 details the timing necessary for a read modify write (RMW) operation. MREQ- will remain low until MRDY+ goes high (after the write portion of the memory cycle). The TDXX- lines are the same as in the read timing. The term STRD- (stop read) goes low after MDRY+ (read is complete from addressed location) and remains low until MWRE- goes low (the write operation is started). MWRE- will go low a minimum of 100 nsec after STRD- and will return to a false condition with MREQ-.

The data read (TDXX-) must be sampled between the leading edges of MRDY+ and STRD-. The complete cycle must terminate within 200 nanoseconds of MREQ- going true.

3.3.4 REFRESH TIMING

3.3.4.1 4K, 8K, and 16K Boards

The 20 MHz systems clock is divided, by the memory board logic, by 640 to provide a 32 µsec refresh timer. Should the refresh timer become activated during a memory cycle, the refresh cycle will be enabled at the end of the memory cycle (see Figure 3-27, Refresh Timing).

If a memory cycle is requested (MREQ-) during a refresh cycle, the cycle will be delayed until the completion of the refresh cycle. Each refresh cycle refreshes one of the 64 row addresses on both upper and lower 4K.

3.3.4.2 32K and 64K Boards

A 32 µsec refresh timing signal is generated on the 32/64K memory board. This refresh signal performs the functions described above for the 4K, 8K, and 16K memory boards.

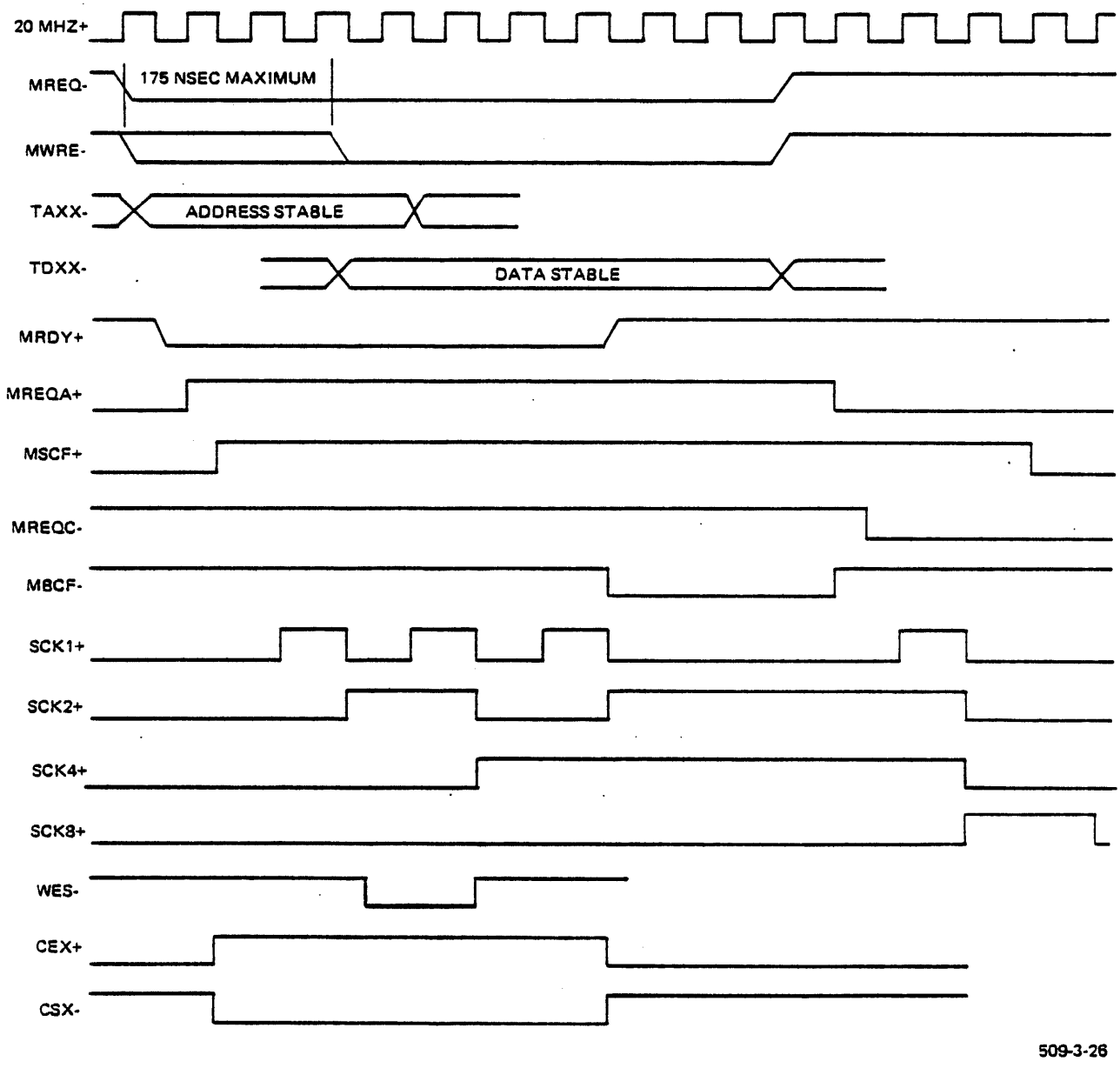


Figure 3-25. Write Cycle Timing

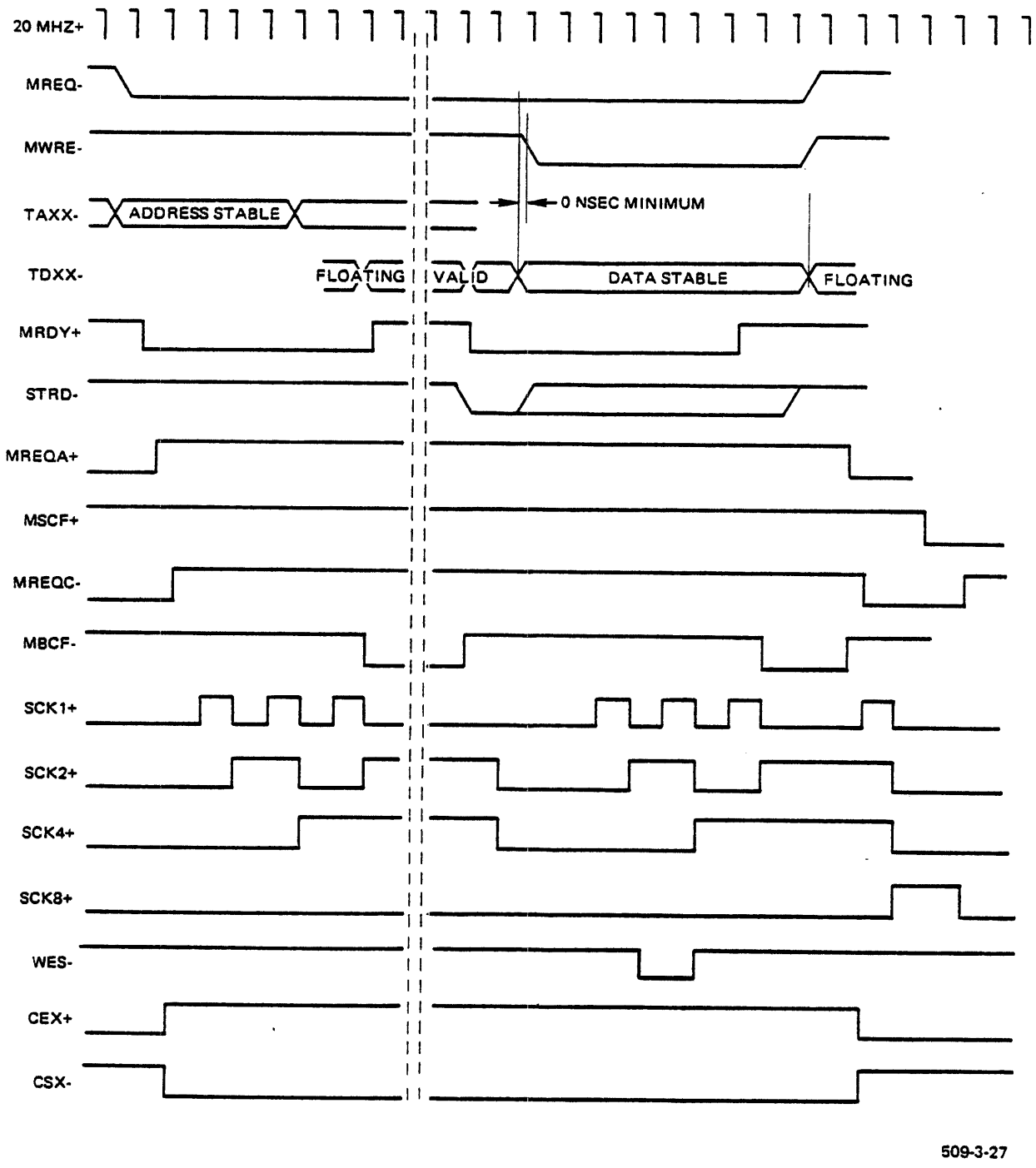


Figure 3-26. Read Modify Write Timing

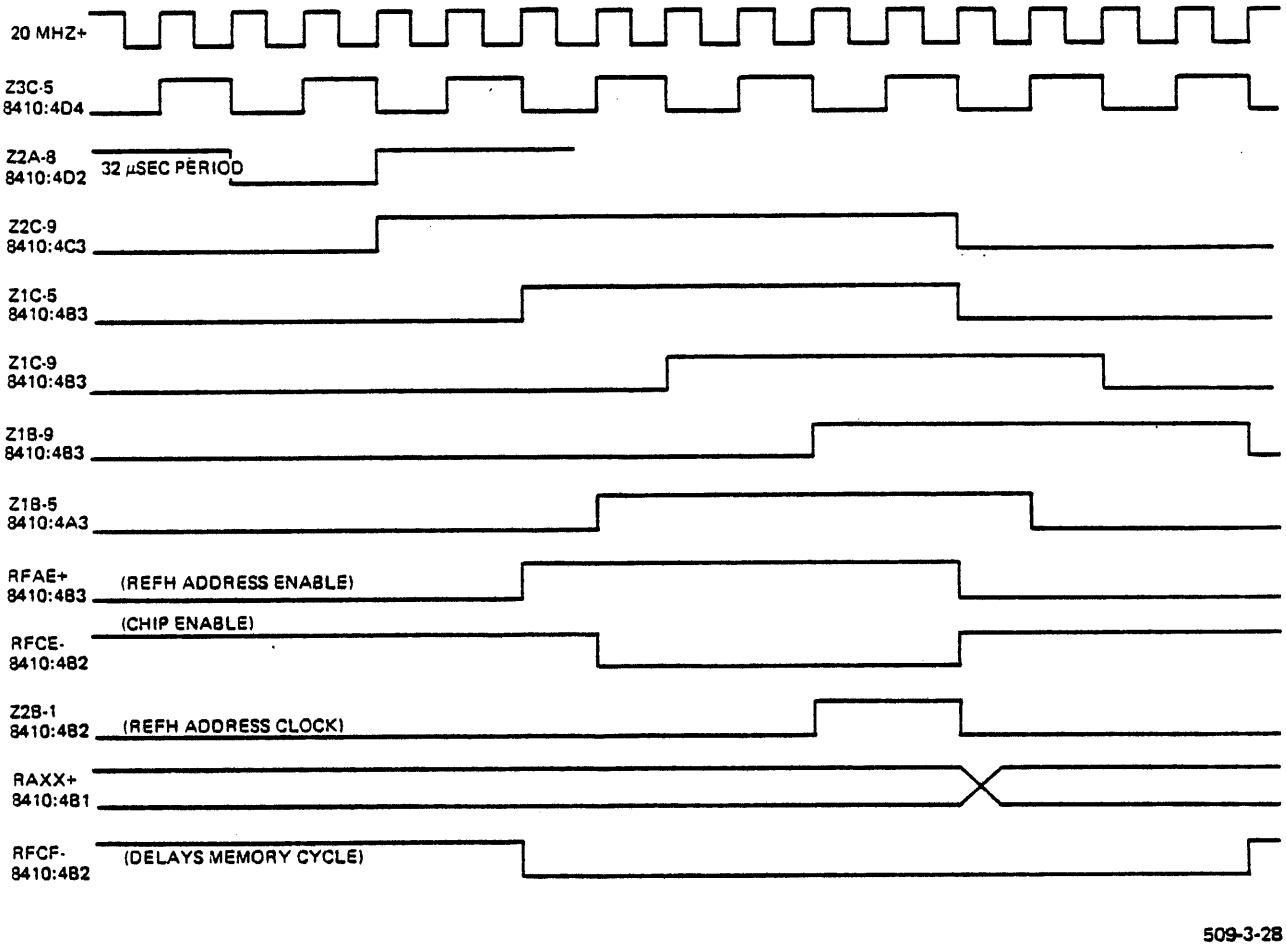


Figure 3-27. Refresh Timing

3.4 REAL-TIME CLOCK (RTCI) INTERRUPT

The GA-16/110 (CPU-1) provides Real-Time Clock interrupt capability utilizing an externally-supplied interrupt signal (clock pulse) and enable. The external RTC interrupt signal may be provided by the user, via the RTCKI+ line (90C02422 Sheet 10), on Pin 87 of connector P1. The enable line for the RTC interrupt is labeled as RTCKI+ (ENABLE) and may be supplied via pin 76 of P-1 (90C02422 Sheet 10).

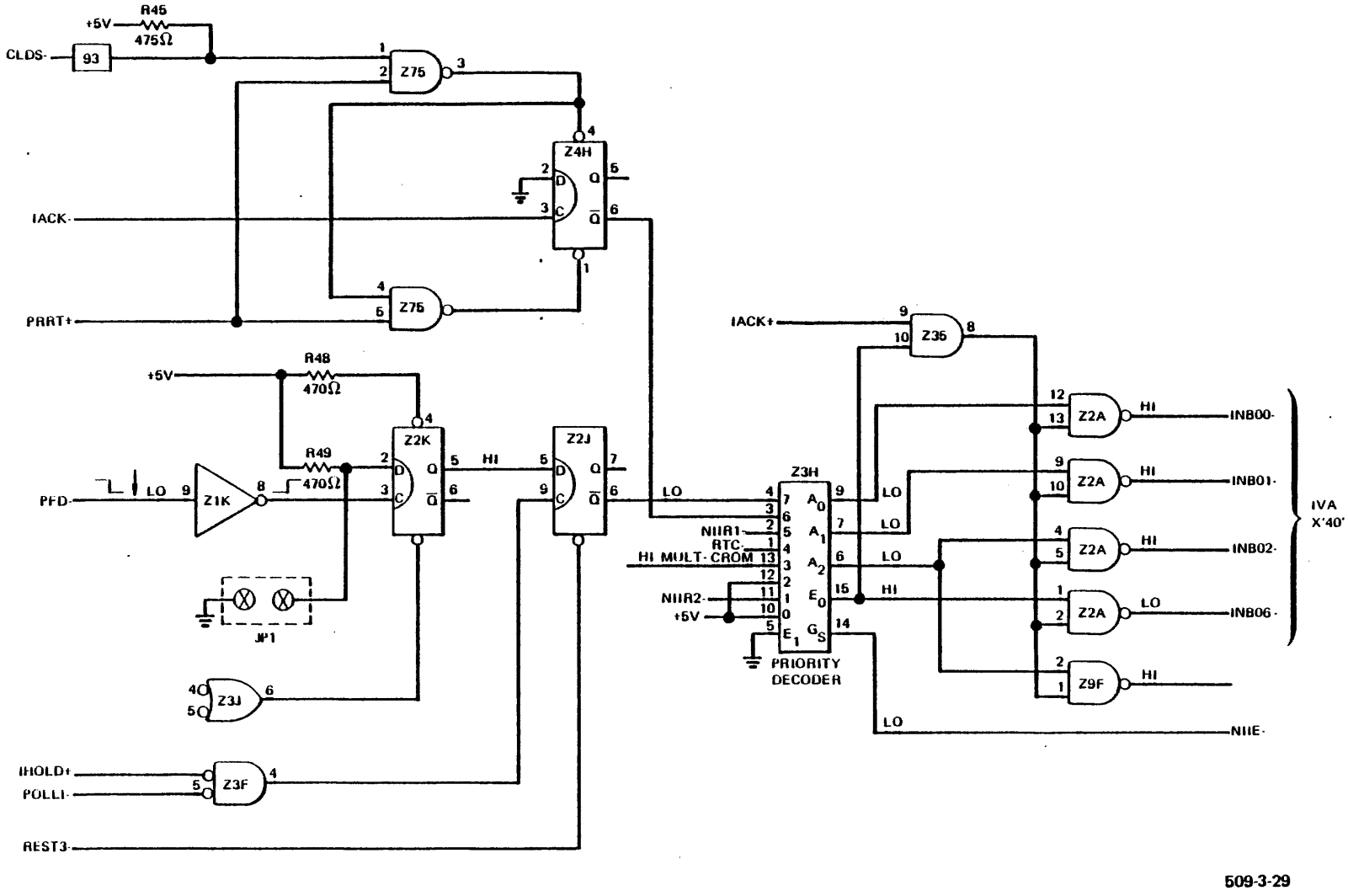
The GA-16/220 (addition of CPU-2 module), provides a hardware-generated, 1ms, RTC interrupt clock pulse (90C02429 Sheet 4). This RTC interrupt clock pulse is derived from the system 20MHz clock oscillator.

3.5 POWER FAIL DETECT

Both GA-provided power supplies (jumbo and compact) monitor the AC primary voltages. If AC input voltage should drop below 104.5 VAC, the interrupt logic is signaled. The sensing of a loss of power drives the power-fail detect line, PFD-, true (reference Figure 3-28) to initiate a non-inhibitable interrupt through the power-fail vector address (location X'40').

NOTE

When the GA-16/110/220 is powered by a user-supplied power system, the power-fail signal must be supplied to the logic via pin 89 of connector P1 (90C02422A Sheet 7).



88A00509A-B

Figure 3-28. PFD Interrupt Vector Address
(90C02422A Sheet 10)

3.6 ERROR CORRECTION OPTION

The error correction option provides single-bit error correction and multiple-bit error detection for a 32K or 64K memory board. The error correction option consists of a single printed circuit module (31D02647A) which plugs into a 32Kx16-bit or 64Kx16-bit memory board.

To perform error correction and detection, the ECC board generates and stores six check bits as each data word is written into memory. During memory read, a second group of six check bits is generated and exclusive-ORed with the check bits generated during write. If the result is zero, the data word was stored and read correctly. A single-bit error results in a non-zero result with the bit position in error specified by the decode of the result. Multiple-bit errors also generate a non-zero result from the exclusive-OR operation, but the decode does not indicate positions for the erroneous bits.

NOTE

Upon initial power up, all ECC check bits are incorrect. All memory locations must be initialized by a store operation before the error correction and detection is enabled. Failure to initialize the memory will cause false error indications.

3.6.1 OPERATIONAL MODES

Operational modes for the error correction options are established by the program. The error correction option utilizes the following operational modes:

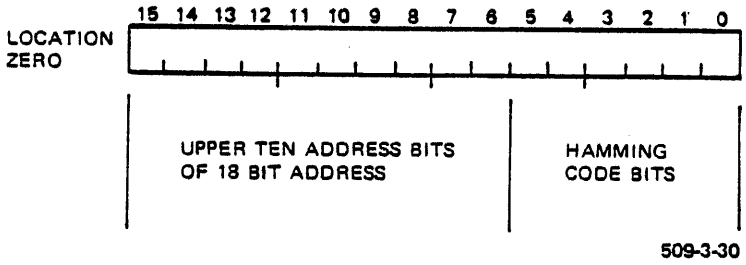
- Error Correction OFF — Disables the error correction feature.
- Normal 1 Mode — The ECC module generates an interrupt on both correctable (single-bit) and non-correctable (multiple-bit) errors.
- Normal 2 Mode — The ECC module generates an interrupt on non-correctable errors only while single-bit errors are corrected.
- Write Data Only Mode — Used for test or diagnostic purposes only. New check bits are generated, but not stored in memory. The original check bits remain.
- Input Status Mode — Read status word from ECC module if the associated memory board has an error (see below).
- Error Correction Memory Mode — This mode is reserved for memory tester usage.

Additional software-related information on the ECC operation is supplied in the GA-16/110/220 System Reference Manual.

3.6.2 INPUT STATUS MODE

The input status mode allows a status word to be generated by the error correction board for each memory module with an error.

A status priority signal from the CPU is propagated through each error correction board. The first board with an error will trap the status priority signal. Executing a read of memory location zero will cause the error correction board to place the following status word on the data bus:



Following the read command, the status priority signal is sent to the next error correction board having an error. This board also blocks propagation of the priority status signal, and responds to a CPU read of location zero. After transmitting the status priority signal, the CPU continues to read until a status word of all zeros has been received, indicating no more errors exist.

NOTE

Normal Mode Two, single bit errors also cause the ECC to generate a status word which can be examined utilizing the Input Status Mode.

3.7 POWER UP/RESTART

During power-up sequencing the system is reset, ISE is off, the CPU is placed in both foreground and 32K modes, and a non-inhibitable interrupt (NIIEC+) is generated. The non-inhibitable interrupt forces one of several courses of action:

1. When the cold start line (CLDS-) is high, the processor generates an interrupt, indirect, through address X'41', the dedicated memory restart vector. (Reference Figure 3-29. Restart Vector Address Generation.) At location X'41', the starting address of the power-up subroutine should be available. For this feature to be properly implemented, a battery back-up power supply must be installed to preserve the contents of the RAM memory while AC power is unavailable.

2. If, on power-up, the cold start line (CLDS-) is low (ground), two possibilities exist:

   a. If the systems console interface (SCI) is installed and the console is active, (CNSL- is true, 90C02405A Sheet 2), a jump is forced to address X'nC00'. Address X'nC00' (7C00 or FC00) is the beginning address of the console ROM and allows interactive operator manual restart of the system by entering the SCI ROM program.

      Access to the starting address of the console ROM is accomplished through the logic depicted in Figure 3-30. Console ROM Address Logic. The term LSADD+ is driven true and the signal IPLF- is false. This is due to the fact that the console is active (CNSL- is true). Since LSADD+ is true and IPLF- is false, the start-up address "read" from Z2D and Z2C (now disabled by IPLF-) is X'nC00' (X'7C00' or X'FC00'). This value is gated over the TD-Bus, through the CPU-2 module and back to the SCI via the TD lines. The value on the "TDxx' lines is the restart address (X'nC00') and accesses the starting location in the console ROM (90C02405A Sheets 4, 5 - Z6B, Z5B, Z4B, Z3B).

   b. If the SCI is installed, but the microconsole is disabled due to the console switch being out of the CNSL position, the signal IPLF- (90C02405A Sheet 2) is driven true. IPLF- being true, in conjunction with LSADD+, "reads" the setting of the IPL 16-position selector switch (S1 in Figure 3-30. Console ROM Address Logic) onto the TD-Bus. This value, which is the IPL ROM address, is routed through the CPU-2 module back to the SCI module, via the TD-Bus lines, as the address of the specific routine, within the 256-word IPL PROM, (90C02405A Sheets 4 and 5). This affects the automatic loading of the program from the selected IPL device.

   c. Small, dedicated, GA-16/110's and GA-16/220's with CLDS- low and without the SCI option, automatically reload the program via the IPL PROM on the memory board (90C02277A Sheet 2).

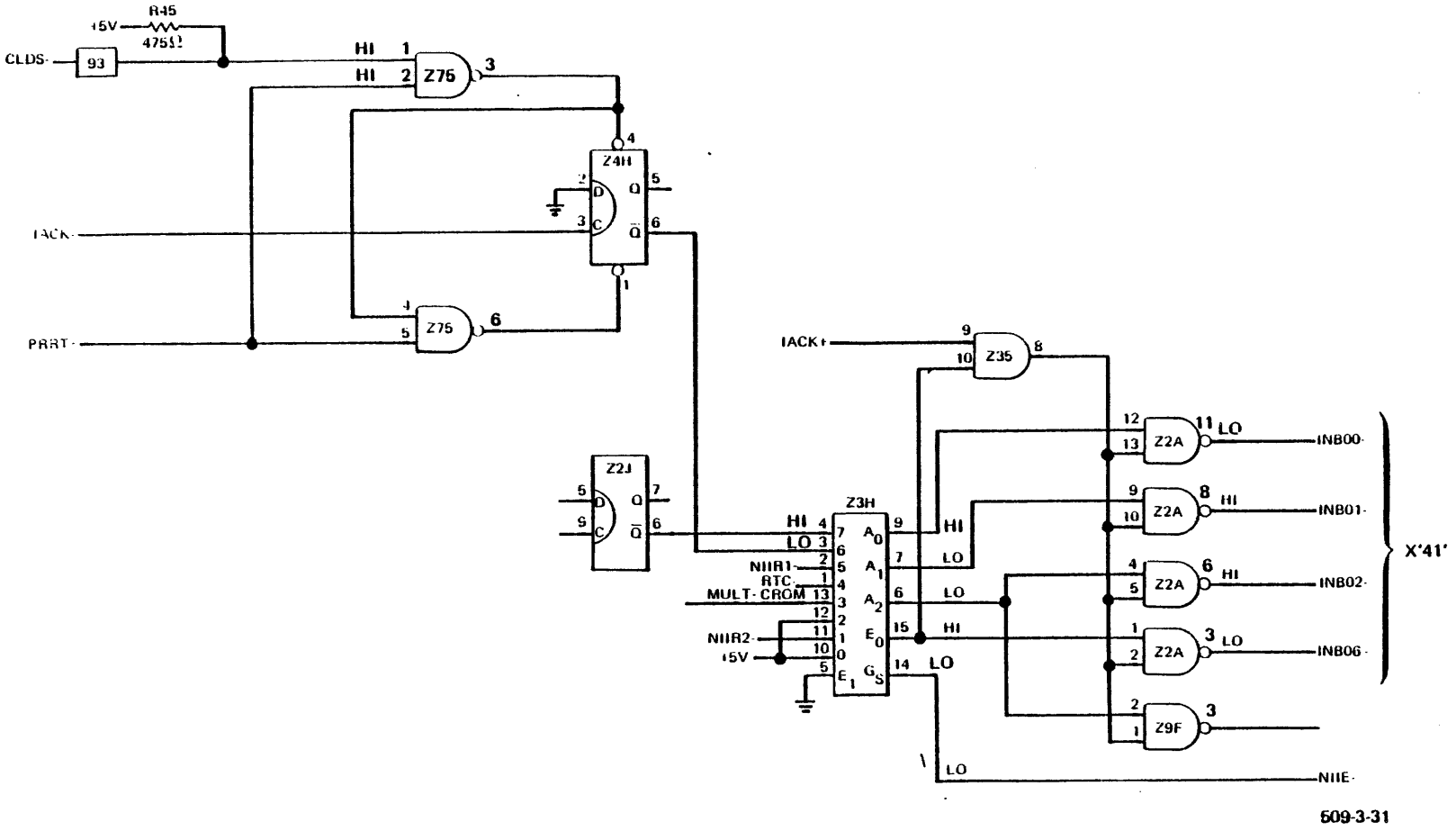


Figure 3-29. Restart Vector Address Generation
(90C02422 Sheet 10)

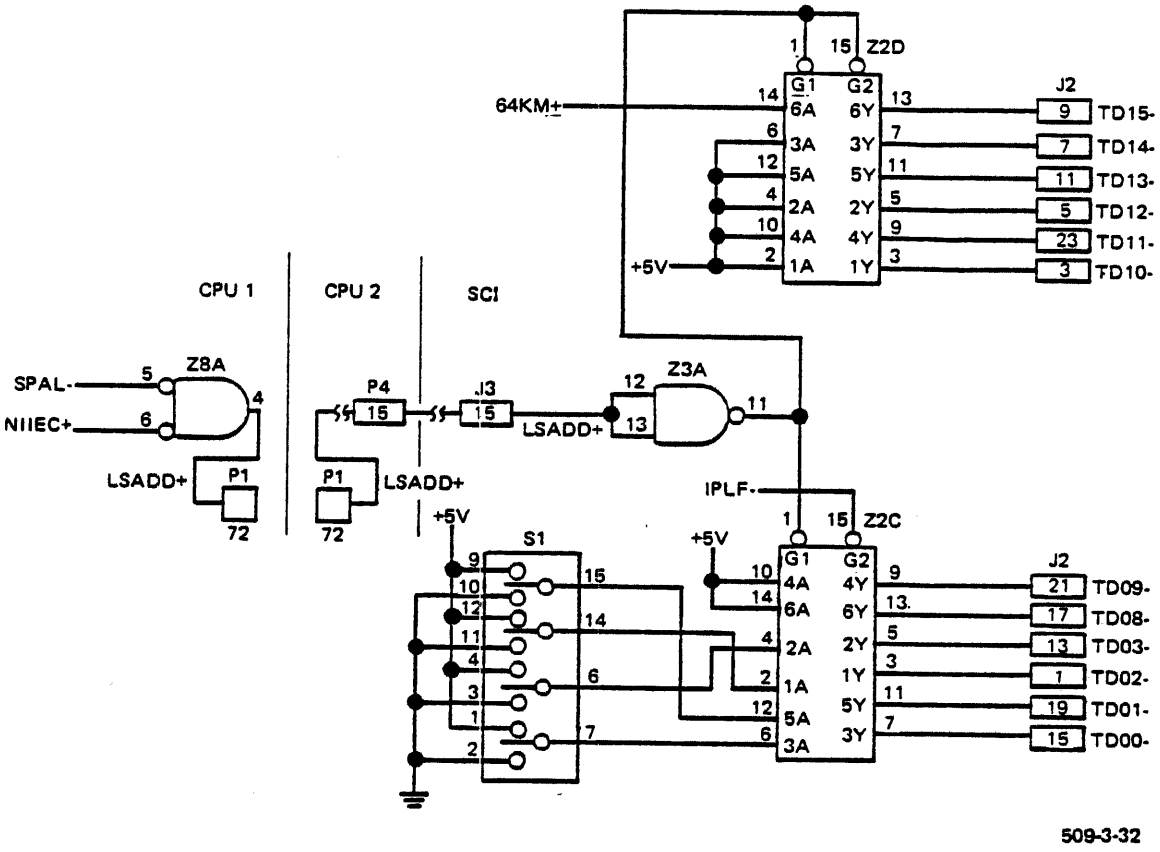


Figure 3-30. Console ROM Address Logic
(90C02422 and 90C02405 Sheet 2)

3.8 USER-DESIGNED CONSOLE INTERFACING

A 16-pin connector plug (P2) has been supplied on the front edge of the CPU-1 module. This connector allows for user-designed, console interface capability. The following signals are available via P2 (reference 90C02422A Sheets 2/3):

SVI- - P2-Pin 8 — This is the dynamic save I control which, when active, allows continuous execution of one instruction by preventing the gating of any new instruction into the CROM I register.

ENTER- - P2-Pin 7 — The ENTER signal (when the CPU is in idle) allows strobing of a user designed manual data entry switches into the selected CPU register. The manual data switches may be tied to the INBUS lines.

SWA,B,C,D- - P2-Pins 4,2,1,3, respectively. These four "register select" switches allow SPC-16 compatability in that they are used for hex address selection of the programmable registers A through E (X '0' through X'7'). The register address codes are the same as those used on the instruction set.

RUN+ - P2-Pin 6 — This line is driven high, to power an indicator, when the CPU is in the run mode. This line may also be tied to a console switch and, thereby, be used as a run command to the CPU.

STEP- - P2-Pin 5 — In the Idle mode: Driving this line true, via a console pushbutton, allows the CPU to fetch an instruction from a location in memory (specified in the P register), execute that instruction and increment the P register. In the run mode: automatic program execution is initiated by activating the STEP+ line.

3.9 SERIAL I/O CONTROLLER

The GA-16/220 provides serial I/O communication to either a CRT or Teletype (Model 33ASR type) via an internal (UART controller) serial I/O controller (reference 90C02429A, Sheet 2).

On early model CPU2 boards, the communications baud rate is selectable, via S3, at either 9600 baud for CRT (Z8A) or 110 baud for teletype (Z9C). Selection of one, or the other, timing pulse controls the transmit and receive clock (TRC/RRC) inputs to the UART (pins 40 and 17 respectively). On later model CPU2 boards, baud rates of 0 though 9600 are selectable (see Appendix G).

The serial I/O controller is assigned device address X'3F' for program selection and interrupts, via IVA X'45', the TTY not busy vector address.

Hardware cabling between the controller and TTY (or CRT) is accomplished via an interface paddleboard. GA assembly 31D01843A or 2417 TTY/OSC, or the RS232 connector paddleboard (GA assembly number 31D02411A01) are used in early production processors. Later processors may include any of four adapters between the serial I/O controller and a peripheral device:

- Model 1622-0021 — Current loop serial I/O adapter without a loop power supply. Converts TTL levels of the UART serial I/O controller to 3-wire, 60V, 20ma current loop signals. Also provides for external monitor and control lines. Uses 31D02486A11 TTY/RS232 paddleboard.

- Model 1622-0022 — Same as 1622-0021, but includes a loop power supply and uses 31C02552A21 paddleboard.

- Model 1622-0023 — RS232 serial I/O adapter. Converts UART TTL levels to EIA RS232 levels for communications with peripheral devices. Uses 31D02486A21 TTY/RS232 paddleboard.

- Model 1622-2234 — Combined RS232 and current loop serial I/O adapter (without loop power supply). Converts UART TTL levels to 3-wire, 60V, 20ma current loop and RS232 signals. Provides both 3-pin current loop connector and 25-pin RS232 connector. Both current loop and RS232 devices receive same data simultaneously. One device at a time can transmit and the message can be displayed on the other. Uses paddleboard 31D02486A01.

3.10 EPROM PROGRAMMER

The erasable, programmable read only memory (EPROM) that may be used in the Piggyback Memory (Section 2.1.9) can be reprogrammed if the GA Model 1622-0051 EPROM Programmer is available. Appendix H shows how the EPROM Programmer and an ultra-violet lamp (for erase) can be used to reprogram EPROM boards or discrete EPROM chips.

GA-16/110/220 instruction summary A

This appendix contains tables which summarize the instructions for the GA-16/110/220. The tables provide a quick reference to all instructions and provide look-up tables of both CAP-16 assembly code and the corresponding binary and hexadecimal instruction format.

Table A-1 lists the basic types of instructions, the general format of the instructions, and provides a key reference number correlating to Table A-2.

Table A-2 is an alphabetical list of instructions, the corresponding hexadecimal representations, and a symbolic representation of the operations which result.

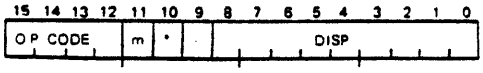
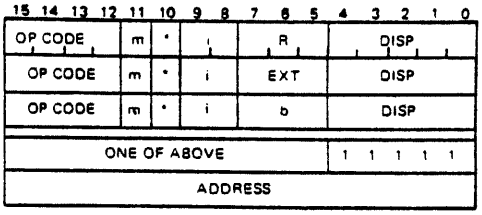
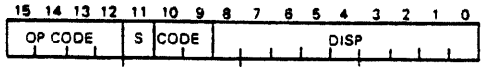
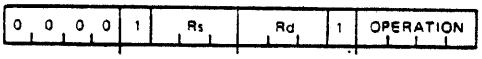
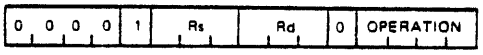
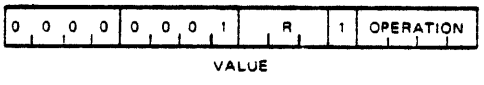
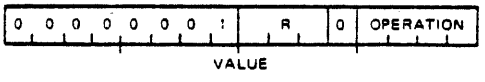
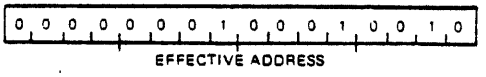
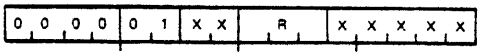
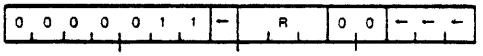
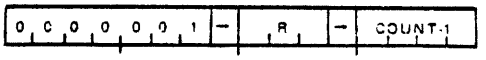
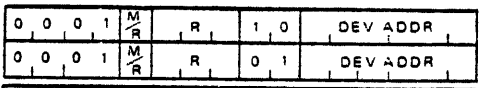
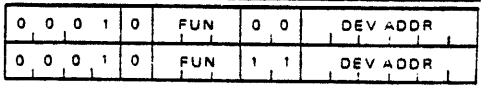
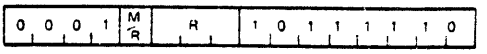
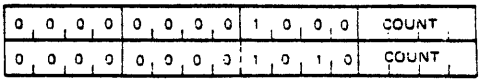
Table A-3 provides additional detail concerning memory reference instructions (Reference 1 in Tables A-1 and A-2.) This table shows addressing modes, range of displacements (or effective addresses), CAP-16 Assembler codes, and corresponding binary and hexadecimal instruction formats.

Table A-4 provides additional detail concerning memory referencing with indexing instructions (references 2, 3, and 4 of Tables A-1 and A-2).

Table A-5 provides detailed instruction formats for standard I/O. This includes the built-in teletype controller, internal mask words, console switch register, and display of data via console data display lights. Refer to Appendix B for the ASCII character set used for teletype.

If further information regarding an instruction is needed reference GA Publication 88A00508A, GA-16/110/220 System Reference Manual, Section 4.

Table A-1. General Instructions

INSTRUCTION TYPE GENERAL FORMAT CAP-16	TABLE A-2 REFERENCE	GENERAL FORMAT
Memory Reference op-code [*]address,[m_1]①	1	
Memory Reference with Indexing op-code R,[*]address[,i][,m_2]① op-code [*]address[,i][,m_2] op-code b,address[,i][,m_2]	2 3 4	
Skip (Extended Displacement) op-code address	5	
Register Operate op-code Rd,Rs	6	
Register Operate Compare op-code Rd,Rs	7	
Register Operate Literal op-code R,value	8	
Register Operate Literal Compare op-code R,value	9	
Subroutine Return Indirect op-code address	10	
Register Change op-code R	11	
Shift Left op-code R	12	
Shift Right op-code R,count	13	
Control op-code	14	
Input/Output op-code R,dev-addr	15	
Control/Test op-code Fun,dev-addr	16	
Read Console Switches op-code R,X'3E'	17	
Multiply/Divide op-code [count]	18	

508-A-1

Notes to Table A-1

① MODE CODE IN CAP-16 STATEMENT

CAP-16	ADDRESSING	INSTRUCTION FORMAT
m_1	0 Program-relative	1 word M=0
m_1	1 Base-relative	1 word m=1
m_2	0 Absolute	1 word m=0 disp≦11110
m_2	1 Base-relative	1 word m=1 disp≦11110
m_2	2 Absolute	2 word m=0; disp=11111
m_2	3 Base-relative	2 word m=1; disp=11111

[*] indicates indirect addressing, causes *field in instruction to be set=1.

Table A-2. Summary of General Instructions

COMMAND	[1]TABLE A-1 REF.	[2]HEX CODE				[3]OPERATION
ADD	6	0	8+Rs	2Rd+1	9	Rs+Rd→Rd;S_Z,S_P,S_O,S_L
ADDC	7	0	8+Rs	2Rd	9	Rs+Rd→DBUS;S_Z,S_P,S_O,S_L
ADDS	11	0	7	2R	8	$R+S_{Shift}→R$;S_Z,S_P,S_O,S_L
ADDV	8	0	1	2R+1	9	R+(P+1)→R;S_Z,S_P,S_O,S_L
ADDVC	9	0	1	2R	9	R+(P+1)→DBUS;S_Z,S_P,S_O,S_L
AND	6	0	8+Rs	2Rd+1	7	Rd∧Rs→Rd;S_Z,S_P
ANDC	7	0	8+Rs	2Rd	7	Rd∧Rs→DBUS;S_Z,S_P
ANDV	8	0	1	2R+1	7	R∧(P+1)→R;S_Z,S_P
ANDVC	9	0	1	2R	7	R∧(P+1)→DBUS;S_Z,S_P
BMS	14	0	4	0	8	0→S_F (Use Background registers)
CMR	2	E	X	2R 2R+1	X[4]	R-(EA)→DBUS;S_Z,S_P,S_O,S_L
CMPL	11	0	7	2R	0	Complement R→R;0→S_L[R initially = 0]|1→S_L[R initially ≠ 0]
CTRL	16	1	fun	dev-addr		Controller carries out function
DECM	3	F	X	4 5	X[4]	(EA)-1→(EA);S_Z,S_P,S_L
DECR	11	0	7	2R	2	R-1→R;S_Z,S_P,S_L
DIV	18	0	0	A	count	BC÷A→C; Remainder→B;S_L
DSPL	11	0	5	2R	4	(No observable function on GA-16/110/220
DTIM	15	1	R	80+dev-addr		Controller→*R
DTIR	15	1	8+R	80+dev-addr		Controller→R
DTOM	15	1	R	40+dev-addr		*R→Controller
DTOR	15	1	8+R	40+dev-addr		R→Controller
EXBY	11	0	6	2R	4	$R_{7-0}↔R_{15-8}$
EXIT	11	0	5	2R	2	$R_{14-0}→P_{14-0}$/R→P
FMS	14	0	4	0	C	1→S_F (Use Foreground registers)
INCM	3	F	X	0 1	X[4]	(EA)+1→(EA);S_Z,S_P,S_L
INCR	11	0	7	2R	E	R+1 R;S_Z,S_P,S_L
INE	14	0	4	0	3	I→ISE

508-A-2

Table A-2. Summary of General Instructions (Cont'd.)

COMMAND	[1]TABLE A-1 REF.	HEX CODE				[3]OPERATION
INH	14	0	4	0	2	0→ISE
JMP	1	7	X	X	X	EA_{14-0}→P_{14-0} / EA_{15-0}→P_{15-0}
JSR	1	6	X	X	X	P+1→E_{14-0};ISE→E_{15};EA→P_{14-0};0→ISE/P+1→E;ISE→S_{ISE};EA→P;0→ISE
LARS	3	F	X	8 9	X[4]	(EA)···(EA+7)→A,X,Y,Z,B,C,D,E;$(EA+8)_{15}$→S_{ISE};$(EA+8)_{8-0}$→S_{8-0}
LDA	1	4	X	X	X	(EA)→A
LDBY	2	8	X	2R 2R+1	X[4]	$(EA)_{15-8}$→R_{7-0}[i even]‖$(EA)_{7-0}$→(EA)→R_{7-0}[i odd]
LDR	2	C	X	2R 2R+1	X[4]	(EA)→R
LDV	3	0	1	2R+1	5	(P+1)→R;S_Z,S_P
LKR	14	0	4	2	0	0→S_L
LKS	14	0	4	3	0	1→S_L
MPY	18	0	0	8	count	A·C→B,C;S_L
OR	6	0	8+Rs	2Rd+1	0	RdVRs→Rd;S_Z,S_P
ORC	7	0	8+Rs	2Rd	0	RdVRs→OBUS;S_Z,S_P
ORV	8	0	1	2R+1	0	RV(P+1)→R;S_Z,S_P
ORVC	9	0	1	2R	0	RV(P+1)→OBUS;S_Z,S_P
PMA	14	0	4	4	0	0→PMA; Reset OMA timer, (must be issued every 200 ms±60ms)
RBIT	4	3	X	2b 2b+1	X[4]	$(\overline{EA})_{b+8}$→S_Z;0→$(EA)_{b+8}$[i even]‖$(\overline{EA})_b$→S_Z;0→$(EA)_b$[i odd]
RCSM	17	1	R	B	E	CSW → *R
RCSR	17	1	8+R	B	E	CSW → R
RCSW	11	0	6	2R+1	0	(Cannot be used on GA-16/110/220; use RCSR instead)
RISE	11	0	5	2R	1	R_{15}→ISE/S_{ISE}→ISE
RLK	11	0	7	2R	1	R+S_L→R;S_Z,S_P,S_0,S_L
RTNIV	10	0	1	1	2	$((EA))_{14-0}$→P_{14-0};$((EA)+1)_{15}$→ISE/((EA))→P;$((EA)+1)_{15}$→ISE
RTR	6	0	8+Rs	2Rd+1	5	Rs→Rd;S_Z,S_P
RTRN	11	0	5	2R	3	R_{14-0}→P_{14-0};R_{15}→ISE / R→P;S_{ISE}→ISE
SARS	3	F	X	C D	X[4]	A,X,Y,Z,B,C,D,E,S→(EA)···(EA+8)
SBIT	4	B	X	2b 2b+1	X[4]	$(\overline{EA})_{b+8}$→S_Z;1→$(EA)_{b+8}$[i even]‖$(\overline{EA})_b$→S_Z;1→$(EA)_b$[i odd]

Table A-2. Summary of General Instructions (Cont'd.)

COMMAND	[1]TABLE A-1 REF.	[2]HEX CODE				[3]OPERATION	
SKM	5	2	6 7	X	X	EA→P [$S_{P=0}$]|P+1→P[$S_{P=1}$]	XX is 2's complement of displacement when 2nd digit is odd; skip reverse (range +255 thru -256)
SKN	5	2	4 5	X	X	EA→P [$S_{Z=0}$]|P+1→P[$S_{Z=1}$]	
SKOF	5	2	0 1	X	X	EA→P [$S_{O=0}$]|P+1→P[$S_{O=1}$];S_O	
SKOT	5	2	8 9	X	X	EA→P [$S_{O=1}$];S_O|P+1→P[$S_{O=0}$]	
SKP	5	2	E F	X	X	EA→P [$S_{P=1}$]|P+1→P[$S_{P=0}$]	
SKR	5	2	2 3	X	X	EA→P [$S_{L=0}$]|P+1→P[$S_{L=1}$]	
SKS	5	2	A B	X	X	EA→P [$S_{L=1}$]|P+1→P[$S_{L=0}$]	
SKZ	5	2	C D	X	X	EA→P [$S_{Z=1}$]|P+1→P[$S_{Z=0}$]	
SLC	12	0	6	2R	3	R_b→R_{b+1}; R_{15}→R_0; R_{15}→S_L;S_Z,S_P,S_L	
SLCL	12	0	7	2R	3	R_b→R_{b+1}; S_L→R_0; R_{15}→S_L; S_Z,S_P,S_L	
SLIO	12	0	7	2R	5	R_{15}→S_L; R_b→R_{b+1}; 1→R_0; S_Z,S_P,S_L	
SLIZ	12	0	7	2R	4	R_{15}→S_L; R_b→R_{b+1}; 0→R_0; S_Z,S_P,S_L	
SRA	13	0	2	2R+1	count-1	R_{15}→R_{15}; R_b→R_{b-1}···;R_0→S_L[count]; S_Z,S_P,S_L	
SRC	13	0	3	2R	count-1	R_0→R_{15}; R_b→R_{b-1}···;R_0→S_L[count]; S_Z,S_P,S_L	
SRCL	13	0	3	2R+1	count-1	S_L→R_{15}; R_0→L; R_b→R_{b-1} [count]; S_Z,S_P,S_L	
SRLC	13	0	2	2R	count-1	0→R_{15};R_b→R_{b-1}···; R_0→S_L[count]; S_7,S_P,S_L,S_{Shift} [shift stops when S_L=1; count→S_{Shift}]	
STA	1	5	X	X	X	A→(EA)	
STBY	2	9	X	2R 2R+1	X[4]	R_{7-0}→$(EA)_{15-8}$[i even]|R_{7-0}→$(EA)_{7-0}$[i odd]	
STR	2	0	X	2R 2R+1	X[4]	R→(EA)	
SUB	6	0	8+Rs	2Rd+1	6	Rd-Rs→Rd; S_Z,S_P,S_O,S_L	
SUBC	7	0	8+Rs	2Rd	6	Rd-Rs→OBUS; S_Z,S_P,S_O,S_L	
SUBV	8	0	1	2R+1	6	R-(P+1)→R; S_Z,S_P,S_O,S_L	
SUBVC	9	0	1	2R	6	R-(P+1)→OBUS: S_Z,S_P,S_O,S_L	
SYNC	14	0	4	8	0	PULSE→SYNC TEST POINT	
TBIT	4	A	X	2h 2b+1	X[4]	$(\overline{EA})_{b+8}$→S_Z[i even]|$(\overline{EA})_b$→S_Z[i odd]	
TEST	16	1	fun	CO+dev-addr		Function True P+2→P; Function False P+1→P	
TRAP	14	0	0	1	X	ISE→$(7D)_{15}$;I_{14-0}→$(7D)_{14-0}$;P→(7C);(44)→P;0→ISE	

Table A-2. Summary of General Instructions (Cont'd.)

COMMAND	TABLE A-1 REF.	HEX CODE				OPERATION
TRS	11	0	5	2R	8	$R_8 \rightarrow S_F$; $R_7 \rightarrow S_Z$; $R_6 \rightarrow S_P$; $R_5 \rightarrow S_0$; $R_4 \rightarrow S_L$; $R_{3-0} \rightarrow S_{Shift}$/ $R_{15} \rightarrow S_{ISE}$; $R_{8-0} \rightarrow S_{8-0}$
TSR	11	0	6	2R	8	$S_{ISE} \rightarrow R_{15}$; $S_{Mode} \rightarrow R_{14}$; $0 \rightarrow R_{13-9}$; $S_F \rightarrow R_8$; $S_Z \rightarrow R_7$; $S_P \rightarrow R_6$; $S_0 \rightarrow R_5$; $S_L \rightarrow R_4$; $S_{Shift} \rightarrow R_{3-0}$
WAIT	14	0	0	0	X	P→P; Press STEP switch to do P+1→P
XEC	11	0	5	2R+1	0	R→I
XOR	6	0	8+Rs	2Rd+1	8	Rd¥Rs→Rd; S_Z,S_P
XORC	7	0	8+Rs	2Rd	8	Rd¥Rs→OBUS; S_Z,S_P
XORV	8	0	1	2R+1	8	R¥(P+1)→R; S_Z,S_P
XORVC	9	0	1	2R	8	R¥(P+1)→OBUS; S_Z,S_P
ZERO	11	0	6	2R	0	0→R
ZLBY	11	0	6	2R	2	$0 \rightarrow R_{15-8}$
ZRBY	11	0	6	2R	1	$0 \rightarrow R_{7-8}$

Notes to Table A-2

①Numbers in Table A-1 reference column identify the general format of instructions included in Table A-2.

②

CAP-16 CONVENTIONS TRANSLATED TO BINARY AND HEX EQUIVALENTS

CAP-16 MNEMONIC	BINARY CODE	R Rd	2R 2Rd	2R+1 2Rd+1	8+R 8+Rs
A	000	0	0	1	8
X	001	1	2	3	9
Y	010	2	4	5	A
Z	011	3	6	7	B
B	100	4	8	9	C
C	101	5	A	B	D
D	110	6	C	D	E
E	111	7	E	F	F

③CONVENTIONS USED IN OPERATION COLUMN

R	register (usage determined by inst.)
Rs	source register
Rd	destination register
EA	effective address
R_b	bit in a register
R_{b+1}	next bit left
R_{b-1}	next bit right
IV	interrupt vector for I/O
DBUS	only a comparison is made, setting appropriate status register bits

/	32K/64K mode alternate operation
|	alternate operation based on a condition for transfer
[ ]	condition for alternate operation
Data	transfers are full word unless bits are specifically indicated; e.g.,
R	always means contents of register
*R	means R points to address in memory (i.e., Indirect Address)

STATUS REGISTER, S

TERM	BIT	DESCRIPTION
S_{ISE}	S_{15}	ISE SAVE STATUS
S_{MODE}	S_{14}	0=32K/1=64K
S_F	S_8	FOREGROUND
S_Z	S_7	ZERO
S_P	S_6	PLUS
S_O	S_5	OVERFLOW
S_L	S_4	LINK
S_{Shift}	S_{3-0}	SHIFT COUNT

LOGIC SYMBOLS

V Or

⊻ Exclusive-OR

∧ And

MATH SYMBOLS

+ Add

- Subtract

· Multiply

: Divide

④Refer to Table A-4 when the Hex Code indicates one of the following:

X 2R / 2R+1 X

X 2b / 2b+1 X

X { } X

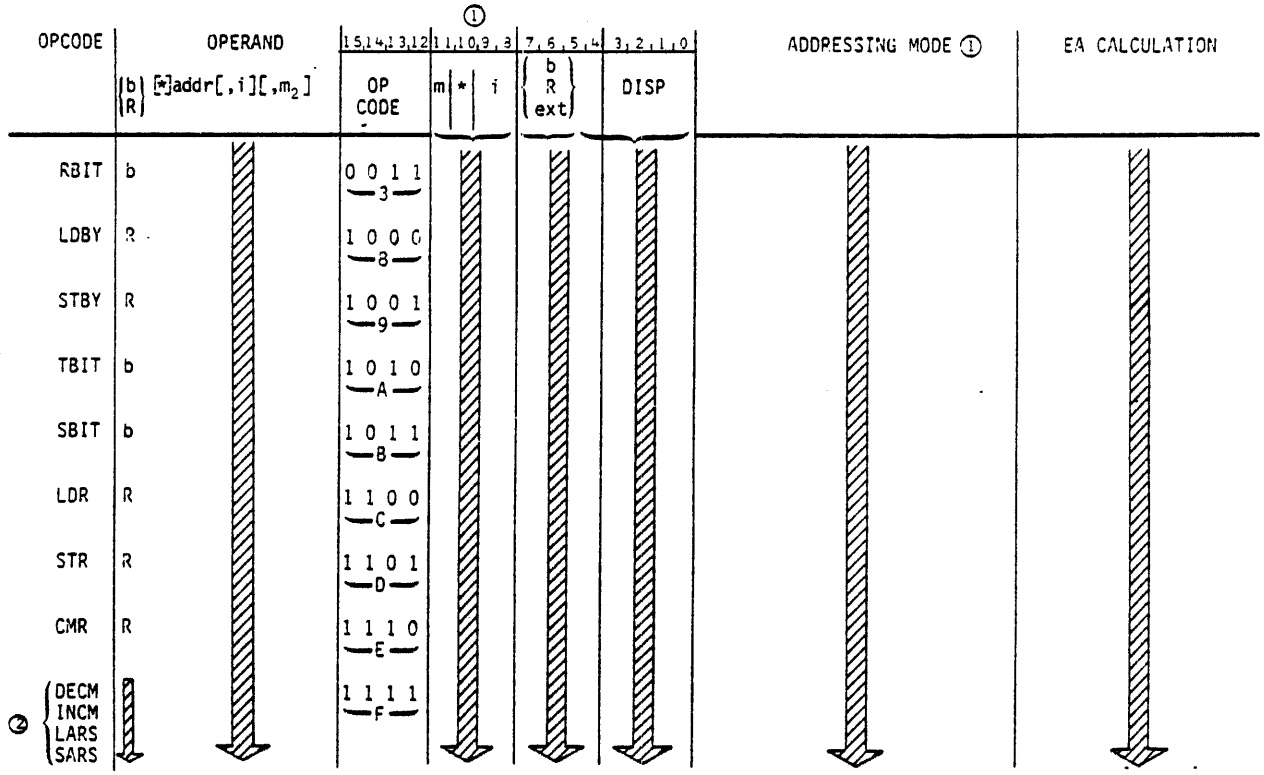
Table A-3. Memory Reference Instructions EA Calculations

CAP-16 CODING FORMAT		BINARY HEX				ADDRESSING MODE	EA CALCULATION	NEXT INSTRUCTION
OPCODE[*]address [,m]		15 14 13 12 CODE	11 m	10 *	9 8 7 6 5 4 3 2 1 0 Displacement			
JMP JSR LDA STA	Refer to Examples Below	0 1 1 1 — 7 0 1 1 0 — 6 0 1 0 0 — 4 0 1 0 1 — 5			Refer to examples below for range and transitions from negative (2s complement) to positive values			
Refer to Codes	-512 : -1 0 : +511				0 0 1 0 — 2 0 0 0 0 — 0 0 0 0 — 0 0 0 1 1 — 3 1 1 1 1 — F 1 1 1 1 — F 0 0 0 0 — 0 0 0 0 — 0 0 0 0 — 0 0 0 0 1 — 1 1 1 1 — F 1 1 1 1 — F	Program Relative	P+1+disp	P+1
	*-512 : *-1 *0 : *511				0 1 1 0 — 6 0 0 0 0 — 0 0 0 0 — 0 0 1 1 1 — 7 1 1 1 1 — F 1 1 1 1 — F 0 1 0 0 — 4 0 0 0 0 — 0 0 0 0 — 0 0 1 0 1 — 5 1 1 1 1 — F 1 1 1 1 — F	Program Relative Indirect	(P+1+disp)	P+1
	0,1 : 1023,1				1 0 0 0 — 8 0 0 0 0 — 0 0 0 0 — 0 1 0 1 1 — B 1 1 1 1 — F 1 1 1 1 — F	Base Relative	D+disp	P+1
	*0,1 : *1023,1				1 1 0 0 — C 0 0 0 0 — 0 0 0 0 — 0 1 1 1 1 — F 1 1 1 1 — F 1 1 1 1 — F	Base Relative Indirect	(D+disp)	P+1

Range of address (disp) may be numbers shown above or any CAP-16 address expression which equals these numbers.

508-A-3

Table A-4. Memory Reference with Indexing, EA Calculations, Ref. 2, 3, 4 in A-1 and A-2



NOTES:

①Addressing mode and indexing are determined by M, *, i bits as follows:

m = 0	Absolute	i = 00	No index
= 1	Base	= 01	X index
* = 0	Direct	= 10	Y index
= 1	Indirect	= 11	Z index

For instructions which reference bits (RBIT, SBIT, TBIT) and instructions referencing bytes (LDBY and STBY) an even index value refers to left byte, and an odd index value refers to right byte. Therefore, EA is divided by 2 for those instructions.

②Refer to sheet 3 and 4 for extension (ext.).

508-A-4

Table A-4. Memory Reference with Indexing, EA Calculations (Cont'd)
Ref. 2, 3, 4 in A-1 and A-2

OPCODE	OPERANDS {b|R} [*]addr[,i][,m_2]	15,14,13,12 OP CODE	11,10,9,8 m * i	7,6,5,4 {b R ext}	3,2,1,0 DISP	ADDRESSING MODE	EA CALCULATION ③
	{addr,, / addr,0,} {0 0 2}		0 0 0 0 —0—			Absolute Direct	addr→EA
	addr,X, {0 2}		0 0 0 1 —1—			Absolute Direct Indexed	addr+X→EA
	addr,Y, {0 2}		0 0 1 0 —2—				addr+Y→EA
	addr,Z, {0 2}		0 0 1 1 —3—				Addr+Z→EA
	{*addr,, / *addr,0,} {0 0 2}		0 1 0 0 —4—			Absolute Indirect	(addr)→EA
	*addr,X, {0 2}		0 1 0 1 —5—			Absolute Indirect Indexed	(addr)+X→EA
	*addr,Y, {0 2}		0 1 1 0 —6—				(addr)+Y→EA
	*addr,Z, {0 2}		0 1 1 1 —7—				(addr)+Z→EA
	{addr,, / addr,0,} {1 1 3}		1 0 0 0 —8—			Base-relative Direct	addr+D→EA
	addr,X, {1 3}		1 0 0 1 —9—			Base-relative Direct, Indexed	addr+D+X→EA
	addr Y, {1 3}		1 0 1 0 —A—				addr+D+Y→EA
	addr,Z, {1 3}		1 0 1 1 —B—				addr+D+Z→EA
	{*addr,, / *addr,0,} {1 1 3}		1 1 0 0 —C—			Base Indirect	(addr+D)→EA
	*addr,X, {1 3}		1 1 0 1 —D—			Base Indirect Indexed	(addr+D)+X→EA
	*addr,Y, {1 3}		1 1 1 0 —E—				(addr+D)+Y→EA
	*addr,Z, {1 3}		1 1 1 1 —F—				(addr+D)+Z→EA

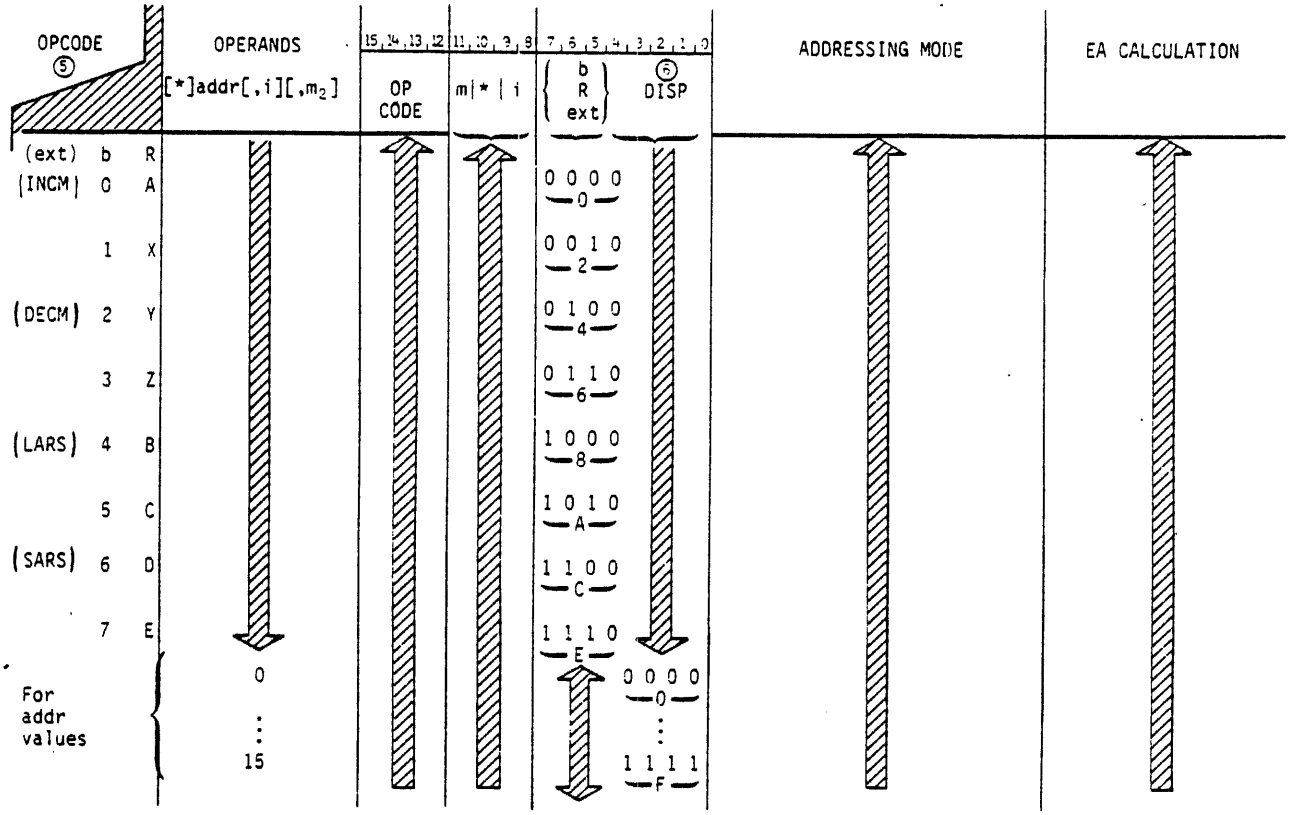
NOTES:

③ EA Calculation is determined by addressing mode. The EA is calculated from the disp field for addr values ≤ 1E. For values > 1E disp field = 1F and disp value in P+1 (2-word instruction).

When m_2 is coded 2 or 3, disp is set to 1F and all displacement value from 0 through 65,536 is in P+1.

If m_2 is not coded, the assembler will provide addressing mode and will determine 1 or 2-word instruction on basis of value of addr.

Table A-4. Memory Reference with Indexing, EA Calculations (Cont'd)
Ref. 2, 3, 4 in A-1 and A-2



NOTES:

⑤ Range of values defining b, R, or INCM, DECM, LARS or SARS instructions and MSB of displacement are in byte 3

⑥ Next instruction is determined by contents of disp field. If disp ≠ 1E, next instruction is at P+1.
If disp = 1F, next instruction is at P+2.
Next instruction is also at P+2 when m_2 is coded 2 or 3.

Table A-4. Memory Reference with Indexing, EA Calculations (Cont'd)
Ref. 2, 3, 4 in A-1 and A-2

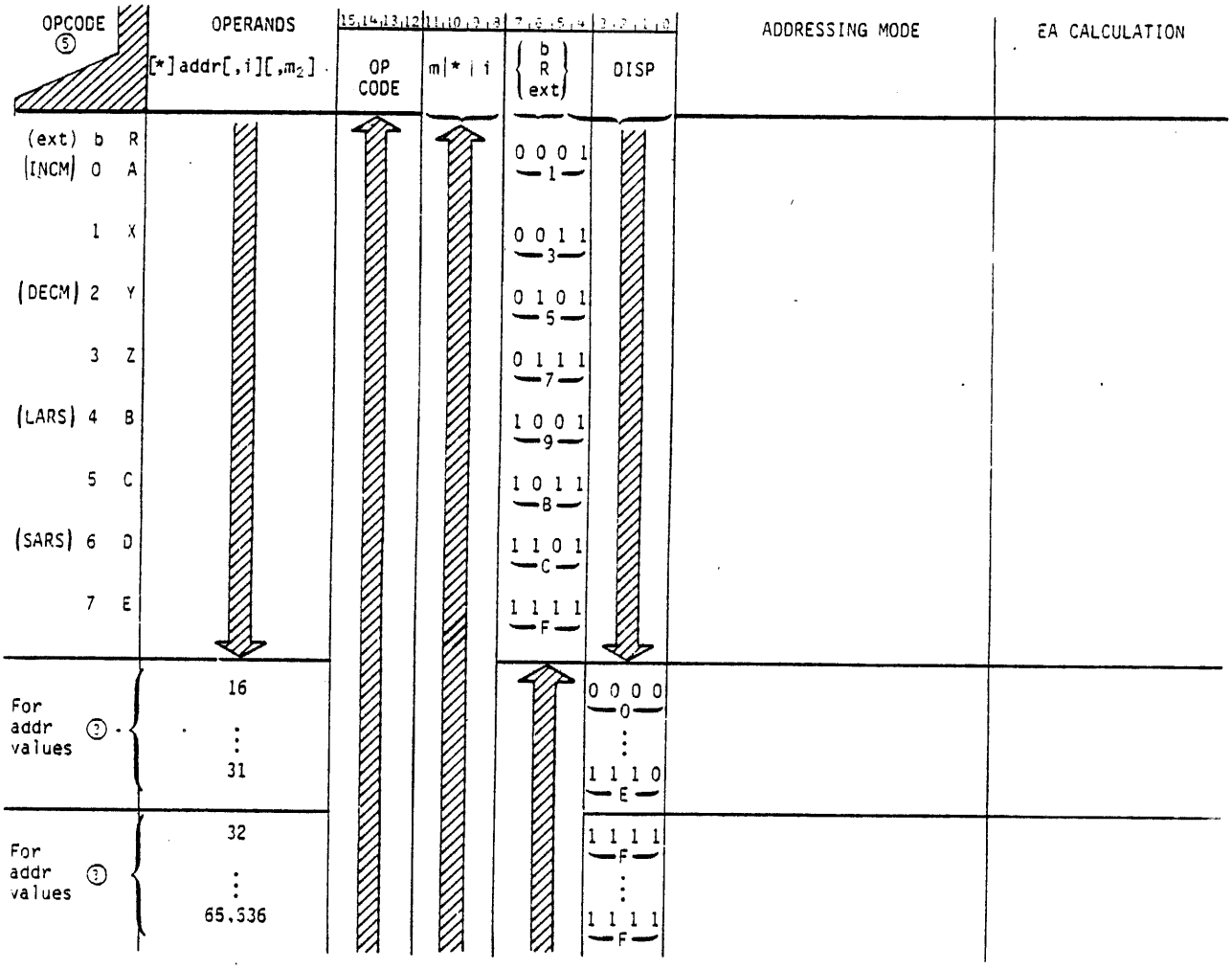


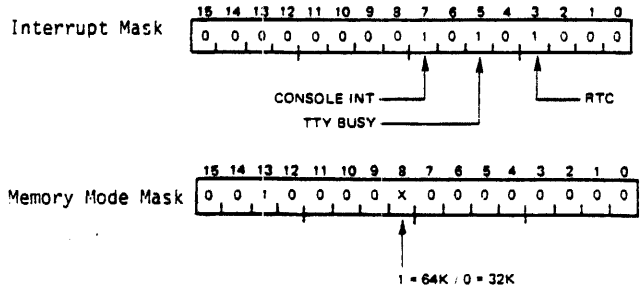
Table A-5. Standard I/O Data & Instructions (Sheet 1 of 2)

TTY DATA OF DEV ADDR 3F INT VECTOR = 45

Reg to TTY	DTOR	1	8+R	7	F
Mem to TTY	DTOM	1	R	7	F
TTY to Reg	DTIR	1	8+R	B	F
TTY to Mem	DTIM	1	R	B	F
TEST for Not-Busy	TEST	1	0	F	F
Set to Transmit	CTRL	1	0	3	F
Set to Rcv	CTRL	1	2	3	F
Set to Rcv/Echo	CTRL	1	4	3	F
Set to Break	CTRL	1	6	3	F

Refer to Appendix B for ASCII data.

MASK DATA FOR DEV ADDR 3E



OUTPUT INSTRUCTIONS TO DEV ADDR 3E

From Register	DTOR	1	8+R	7	E
From Memory	DTOM	1	R	7	E

Power up disables interrupts and sets 32K mode; 32K mode can also be permanently enabled with switch on CPU.

To set or clear mask bits, output a register with desired bits =1 or 0 (as required) as shown in following example:

MACHINE	CAP-16		FUNCTION
0600	ZERO	A	Clear Register A
011D	ORV	A,X'nnnn'	Load hex equivalent of bit pattern
nnnn			2nd word formatted per desired mask data
187E	DTOR	A,X'3E'	Set the mask

INPUT INSTRUCTIONS TO DEV ADDR 3E TO READ CONSOLE SWITCHES

To Register	RCSR or DTIR	1	8+R	B	E
To Memory	RCSM or DTIM	1	R	B	E

DISPLAY DATA ON CONSOLE -

From Register	DSPL	0	5	2R	4

508-A-5

Table A-5. Standard I/O Data & Instructions (Sheet 2 of 2)

CONTROL INSTRUCTIONS TO DEV ADDR 3E				
Single Step (SSTEP) CTRL	1	1	3	E
I/O Reset (IORST) CTRL	1	2	3	E

Single step control instruction executable in upper 1K of memory mode in use. NI interrupt via location X'46', with P+1 and ISE saved in locations X'7E' and X'7F', will occur after first instruction outside upper 1K of memory is executed.

GA-16/110/220 microcode summary B

The following microcode listing is provided as a quick reference for operations, functions, and the hex coding of each of the microinstructions contained within the standard GA-16/110/220 microprocessor (CROM) chipset.

Mnemonic	Add.	M Code	Next Address Group	Operation
RST	000	0 6 4 0 E 0 F 8 5	006	$SVL_{8\rightarrow S}$
LNI	001	0 0 0 4 9 8 B B 0	024	P → MA
INCP	002	0 4 0 0 6 0 E 6 2	003	P + 1 → P,
		0 6 4 4 3 8 E 6 2	020	P + 1 → P, MA
INT	004	0 0 0 F A 0 B 7 0	07C	ISE, $W_{F\rightarrow 0}$ → W
		0 0 0 6 3 2 E 7 1	030	INT DATA → W
IPL	006	0 4 0 0 9 2 0 0 0	005	INT DATA →
		0 6 4 4 3 6 E 7 3	020	SPEC → P, MA
REGISTER DSPLY	008	0 7 0 0 1 1 0 0 0	008	CNSL → I
		0 6 4 4 9 F 0 0 0	024	I → DSPLY
		0 7 0 0 3 5 E 7 1	008	CNSL → W
		0 6 4 4 9 D 6 0 0	024	W → DSPLY
		0 4 0 0 3 5 E 7 3	001	CNSL → P
		0 6 4 4 9 D 3 B 0	024	P → DSPLY
		0 7 0 0 3 5 E 7 5	008	CNSL → S
		0 6 4 4 9 D B A 0	024	S → DSPLY
	010	0 F 0 0 3 5 B 9 7	008	CNSL → E
		0 E 4 4 9 D B 6 7	024	E → DSPLY
		0 F 0 0 3 5 B 9 6	008	CNSL → D
		0 E 4 4 9 D B 6 6	024	D → DSPLY
		0 F 0 0 3 5 B 9 5	008	CNSL → C
	015	0 E 4 4 9 D B 6 5	024	C → DSPLY
		0 F 0 0 3 5 B 9 4	008	CNSL → B
		0 E 4 4 9 D B 6 4	024	B → DSPLY
		0 F 0 0 3 5 B 9 3	008	CNSL → Z
		0 E 4 4 9 D B 6 3	024	Z → DSPLY
	01A	0 F 0 0 3 5 B 9 2	008	CNSL → Y
		0 E 4 4 9 D B 6 2	024	Y → DSPLY
		0 F 0 0 3 5 B 9 1	008	CNSL → X
		0 E 4 4 9 D B 6 1	024	X → DSPLY
		0 F 0 0 3 5 B 9 0	008	CNSL → A
		0 E 4 4 9 D B 6 0	024	A → DSPLY

Mnemonic	Add	M Code								Next Address Group	Operation
FTCH	020	0	4	4	0	9	2	0	0	020	INT DATA → W
		0	6	C	5	3	0	E	6	028	M-DATA → W,I
		0	4	4	0	9	2	0	0	004	INT DATA → W?
		0	7	0	0	1	0	E	6	008	M-DATA → I
		0	4	4	0	9	2	0	0	004	INT DATA → ?
		0	6	C	5	3	7	E	7	028	I → W
		0	4	4	0	9	2	0	0	004	INT DATA →
	027	0	7	0	0	1	0	0	0	008	M-DATA →
CLSFY	028	0	2	8	6	9	A	6	0	034	W → I/O ADD
		0	0	8	8	2	0	E	6	028	P + 1 → P
		0	0	4	7	A	0	E	6	03C	P + 1 → P
		0	0	C	B	0	0	0	0	058	
		0	1	5	4	8	0	0	0	0A4	
		0	1	5	C	8	0	0	0	0E4	
		0	3	2	2	0	0	0	0	110	
		0	4	0	A	B	8	E	6	055	P + 1 → P, MA
SVP	030	0	E	0	7	6	0	B	B	03A	P → E (DISP)
IOLA	031	3	4	0	0	5	B	B	6	003	R_d → I/O Data
		0	0	A	1	8	0	0	0	10C	
		0	7	8	0	5	B	6	0	003	W → I/O Data
IOL	034	0	4	0	0	5	B	6	0	003	W → I/O Data
		3	2	4	7	1	8	B	6	038	R_d → MA
		0	0	0	6	4	0	0	0	032	
		0	2	0	6	0	0	0	0	031	
DTXM	038	0	4	0	0	5	C	6	0	038	W → M-Data
		0	0	0	6	B	0	E	7	034	M-Data → W
WTD	03A	0	4	0	9	1	8	6	0	049	W → M-ADD
		0	7	C	7	5	D	6	0	03A	W → DSPLY
SKP	03C	0	3	8	9	8	0	0	0	04C	
		0	1	8	A	0	0	0	0	050	
		0	3	C	A	0	0	0	0	050	
		0	1	C	A	0	0	0	0	050	
MR	040	0	6	4	4	3	8	F	9	020	P + W_1 → P,MA
		0	5	0	9	1	8	F	9	048	P + W_1 → P,MA
		0	E	4	4	3	8	A	9	020	D + W_2 → P,MA
		0	D	0	9	3	8	9	9	048	D + W_2 → MA
		0	8	0	8	2	0	B	B	040	P → E (DISK)
		0	C	0	8	2	0	B	B	041	P → E (DISK)
		0	8	0	8	6	0	B	B	042	P → E (DISK)
		0	C	0	8	6	0	B	B	043	P → E (DISK)

Mnemonic	Add	M Code	Next Address Group	Operation
LSI	048	0 0 0 A B 0 E 7 1	054	M-Data → W
		0 0 0 A 7 0 E 7 3	052	M-Data → P
		0 8 0 A 7 0 B 9 0	052	M-Data → A
		0 8 0 A 5 C B 6 0	052	A → M-Data
SKPA	04C	0 0 0 A 2 0 F 5 6	050	$S \cdot L_5 \rightarrow S$
		0 0 0 A 6 0 F 5 6	052	$S \cdot L_5 \rightarrow S$
		0 0 0 A 6 0 F 5 6	052	$S \cdot L_5 \rightarrow S$
		0 0 0 A 2 0 F 5 6	050	$S \cdot L_5 \rightarrow S$
SKPB	050	0 6 4 4 3 8 F 8 F	020	$P+W_4 \rightarrow P,MA$
		0 6 4 4 1 8 B B 0	020	P → MA
		0 6 4 4 1 8 B B 0	020	P → MA
		0 6 4 4 3 8 F 8 F	020	$P+W_4 \rightarrow P,MA$
R1ADO	054	0 4 C 9 5 8 6 0 0	04A	W → MA
MRWI*	055	0 0 C B B 0 E 7 1	05C	M-Data → W
RBOP	056	0 4 8 E 3 4 E 7 1	070	M-Data → W
		0 4 8 F 3 4 E 7 1	078	M-Data → W
MRWI	058	0 1 4 C 2 0 F 4 E	060	$W_3 \rightarrow W$
		0 5 4 D 3 8 F 4 E	068	$W_3 \rightarrow W,MA$
		0 9 4 C 2 0 9 4 6	060	$D+W_3 \rightarrow W$
		0 D 4 D 3 8 9 4 6	068	$D+W_3 \rightarrow W,MA$
MRWIB	05C	0 1 4 C 0 0 0 0 0	060	
		0 5 4 D 1 8 6 0 0	068	W → MA
		0 9 4 C 2 0 B 1 6	060	D+W → W
		0 D 4 D 3 8 B 1 6	06B	D+W → W,MA
IDX	060	0 0 0 A D 8 6 0 0	056	W → MA
		0 D C A F 8 8 F 1	056	X/2+W → W,MA
		0 D C A F 8 8 F 2	056	Y/2+W → W,MA
		0 5 4 D 5 8 6 0 0	06A	W → MA
		0 D 4 D 7 8 9 F 1	06A	X+W → W,MA
		0 D 4 D 7 8 9 F 2	06A	Y+W → W,MA
		0 D 4 D 7 8 9 F 3	06A	Z+W → W,MA
MRWIE	068	0 1 4 C B 0 E 7 1	064	M-Data → W
		0 1 4 C B 0 E 7 1	064	M-Data → W
		2 C 0 0 5 C B 6 0	003	Rd → M-Data
		2 C 0 0 7 0 B 9 0	003	M-Data → Rd
		0 1 4 C B 0 E 7 1	064	M-Data → W
		0 1 4 C 3 0 E 7 1	060	M-Data → W
		0 2 5 0 1 4 E 7 1	068	M-Data → (RMW)
		0 4 8 E 3 4 E 7 1	070	M-Data → W(RMW)

Mnemonic	Add	M Code	Next Address Group	Operation
BYE	070	0 0 0 7 2 D E 6 0	038	W+1 → W,Ind
		0 0 0 7 2 D E 7 0	038	W-1 → W,Ind
		2 C 0 F 2 0 7 5 0	079	$R_{du}, W_u \rightarrow Rd$
		2 C 0 0 5 C 7 2 0	003	$Rdl, W_L \rightarrow$ M-Data
		2 C 0 F 0 F 6 2 0	079	Rd-W → Ind
		2 0 1 0 4 8 F 8 0	082	$W \cdot L_{(8\div)} \rightarrow$ Ind
		2 4 0 F 0 8 F 8 0	079	$W-L_{(8\div)} \rightarrow$ Ind
	077	2 0 1 0 C 8 F 8 0	086	$W \cdot L_{(8\div)} \rightarrow$ Ind
ADDS	078	2 C 0 0 6 F 6 1 0	003	$W_3 \rightarrow W$
RSM	079	2 A 0 0 5 0 0 0 0	003	M-Data →
BYO	07A	2 C 0 F 2 0 7 4 0	079	$R_{du}, W_L \rightarrow Rd$
		2 C 0 0 5 C 7 8 0	003	$Wu, RdI \rightarrow$ M-Data
NMOD	07C	0 0 1 2 5 6 F 1 D	092	Spec → (MA) ISER
BYO	07D	2 0 1 1 4 8 F 0 0	08A	$W \cdot L_{(0+)} \rightarrow$ Ind
		2 4 0 F 0 8 F 0 0	079	$W \cdot L_{(0+)} \rightarrow$ Ind
		2 0 1 1 C 8 F 0 0	08F	$W \cdot L_{(0+)} \rightarrow$ Ind
LSA	080	0 C 1 0 7 0 B 9 0	083	M-Data → A
		0 C 1 0 5 C B 6 0	083	A → M-Data
BITE	082	2 4 0 0 5 C F 8 2	003	$W \cdot L_{(8+)} \rightarrow$ M-Data
WCWX	083	0 2 5 0 B 8 E 6 0	084	W+1 → W,MA
LSX	084	0 C 1 0 F 0 B 9 1	087	M-Data → X
		0 C 1 0 D C B 6 1	087	X → M-Data
BITE	086	2 4 0 0 5 C F 8 1	003	$WVL_{(8+)} \rightarrow$ M-Data
WCWY	087	0 2 5 1 3 8 E 6 0	088	W+1 → W,MA
LSI	088	0 C 1 1 7 0 B 9 2	08B	M-Data → Y
		0 C 1 1 5 C B 6 2	08B	Y → M-Data
BITO	08A	2 4 0 0 5 C F 0 2	003	$W \cdot L_{(0)} \rightarrow$ M-Data
WCWZ	08B	0 2 5 1 B 8 E 6 0	08C	M-Data → W,MA
LDE	08C	0 C 1 1 F 0 B 9 3	08F	M-Data → Z
		0 C 1 1 D C B 6 3	08F	Z → M-Data
BITO	08E	2 4 0 0 5 C F 0 1	003	$WVL_{(0)} \rightarrow$ M-Data
WCWZ	08F	0 2 5 2 6 8 E 6 0	090	W+1 → W,MA

Mnemonic	Add	M Code	Next Address Group	Operation
LSB	090	0 C 1 2 7 0 B 9 4	093	M-Data → B
		0 C 1 2 5 C B 6 4	093	B → M-Data
NMOD2	092	0 0 1 2 D C 6 0 0	096	W → M-Data
WCWC	093	0 2 5 2 B 8 E 6 0	094	W+1 → W,MA
LSC	094	0 C 1 2 F 0 B 9 5	097	M-Data → C
		0 C 1 2 D C B 6 5	097	C → M-Data
NMOD3	096	0 0 1 3 5 6 0 0 0	09A	W+1 → W,MA
WCWD	097	0 2 5 3 3 8 E 6 0	098	W+1 → W,MA
LSD	098	0 C 1 3 7 0 B 9 6	09B	M-Data → D
		0 C 1 2 5 C B 6 6	09B	D → M-Data
MNOD4	09A	0 0 1 3 D C B B 0	09E	P → M-Data
MRWE	09B	0 2 5 3 B 8 E 6 0	09C	W+1 → W,MA
LSE	09C	0 C 1 3 F 0 B 9 7	09F	M-Data → E
		0 C 1 3 D C B 6 7	09F	E → M-Data
NMOD5	09E	0 6 0 7 7 2 E 7 1	03A	Int Data → W
WCWS	09F	0 2 5 4 3 8 E 6 0	0A0	W+1 → W → MA
LSS	0A0	0 4 0 0 7 0 E 7 5	003	M-Data → S
		0 4 0 0 5 C B A 0	003	S → M-Data
STBY	0A2	0 7 D 6 0 0 F 3 D	0B0	WAIT
		0 6 9 4 4 0 0 0 0	0A2	
SPO	0A4	0 2 D 5 0 0 0 0 0	0A8	
		0 0 1 C 3 8 E 6 2	0E0	P+1 → P,MA
		0 0 2 1 C 0 F 0 D	10E	0 → S_L
		0 0 2 1 C 0 F 0 D	10E	0 → S_L
WMD	0A8	0 8 2 4 6 0 7 7 4	122	0 → B
		0 0 1 5 8 0 F 0 D	0AC	0 → S_L
		0 7 B 6 0 0 F 3 B	0B0	WAIT
		0 6 9 4 4 0 C 0 0	0A2	Wu,a → W
SPCD	0AC	0 B D 6 6 1 6 E 5	0B2	$C_{E\rightarrow 0}$, 0 → C, Ind
		0 B D 6 6 1 6 F 5	0B2	$C_{E\rightarrow 0}$, 1 → C, Ind
		0 C 2 4 2 1 6 E 5	121	$C_{E\rightarrow 0}$, 0 → C, Ind
		0 C 0 0 6 1 6 F 5	003	$C_{E\rightarrow 0}$, 1 → C, Ind

Mnemonic	Add	M Code	Next Address Group	Operation
WAIT	0B0	0 7 D 6 8 0 0 0 0	0B4	
		0 6 4 4 1 8 8 B 0	020	P → MA
SFBD	0B2	0 B D 7 2 1 6 E 4	0B8	$B_{E\rightarrow 0}$, 0 → B, Ind
		0 B D 7 2 1 6 F 4	0B8	$B_{E\rightarrow 0}$, 1 → B, Ind
WAITA	0B4	0 7 D 6 8 0 0 0 0	0B4	
		0 6 4 4 3 8 E 6 2	020	P+1 → P,MA
SFBM	0B6	0 C 1 7 6 1 6 A 4	0BB	$0,B_{F+1}$ → B,Ind
		1 8 1 6 E 0 1 0 4	0B6	$B+A \rightarrow B$
SBA	0B8	1 8 1 7 6 1 2 0 4	0BA	B-A → B,Ind
		1 8 1 7 6 1 1 0 4	0BA	B+A → B,Ind
CTSD	0BA	0 3 5 5 A 0 E 6 4	0AC	S+1 → S
CTSM	0BB	0 3 5 7 A 0 E 6 4	0BC	S+1 → S
SFCM	0BC	0 B D 6 E 1 6 A 5	0B6	$0,C_{F\rightarrow 1}$ → C,Ind
		0 B D 6 E 1 6 B 5	0B6	$1,C_{F\rightarrow 1}$ → C,Ind
		0 B E 4 2 1 6 A 5	120	$0,C_{F\rightarrow 1}$ → C,Ind
		0 B E 4 2 1 6 B 5	120	$1,C_{F\rightarrow 1}$ → C,Ind
ROL	0C0	2 C 0 0 4 F 6 2 0	003	R_D-W → Ind
		2 C 0 0 4 C 6 3 0	003	R_D·W → Ind
		0 0 0 A 4 0 0 0 0	052	
		2 C 0 0 4 C 6 0 0	003	W → Ind
		2 C 0 0 4 C 6 4 0	003	$R_D \oplus W$ → Ind
		2 C 0 0 4 F 6 1 0	003	R_D + W → Ind
		0 0 0 A 4 0 0 0 0	052	
		2 C 0 0 4 C 6 5 0	003	R_DVW → Ind
		2 C 0 0 6 F 6 2 0	003	R_D-W → R_D,Ind
	0C9	2 C 0 0 6 C 6 3 0	003	R_D·W → R_D,Ind
	0CA	0 4 1 C 1 8 E 6 0	0E1	W+1 → MA
		2 C 0 0 6 C 6 0 0	003	W → R_D, Ind
		2 C 0 0 6 C 6 4 0	003	$R_D \oplus W \rightarrow R_D$,Ind
		2 C 0 0 6 F 6 1 0	003	R_D+W → R_D,Ind
		0 0 0 A 4 0 0 0 0	052	
		2 C 0 0 6 C 6 5 0	003	R_DVW → R_D,Ind

Mnemonic	Add	M Code	Next Address Group	Operation
SEX	0D0	2 B D C 6 D 6 A 0	0E2	0,$R_{DF\rightarrow 1}\rightarrow R_D$, Ind
		2 B D C 6 D 6 A 0	0E2	0,$R_{DF\rightarrow 1}\rightarrow R_D$, Ind
		2 C 1 C 6 D 6 A 0	0E3	0,$R_{DF\rightarrow 1}\rightarrow R_D$, Ind
		2 C 1 C 6 D 6 A 0	0E3	0,$R_{DF\rightarrow 1}\rightarrow R_D$, Ind
		2 8 1 C 6 D 6 9 0	0E2	R_{D0},$R_{DF\rightarrow 1}\rightarrow R_D$, Ind
		2 8 1 C 6 D 6 9 0	0E2	R_{D0},$R_{DF\rightarrow 1}\rightarrow R_D$, Ind
		2 C 1 C 6 D 6 9 0	0E3	R_{D0},$R_{DF\rightarrow 1}\rightarrow R_D$, Ind
		2 8 1 C 6 D 6 8 0	0E2	R_{DF},$R_{DF\rightarrow 1}\rightarrow R_D$, Ind
		2 8 1 C 6 D 6 8 0	0E2	R_{DF},$R_{DF\rightarrow 1}\rightarrow R_D$, Ind
		2 C 1 C 6 D 6 8 0	0E3	R_{DF},$R_{DF\rightarrow 1}\rightarrow R_D$, Ind
		2 8 1 C 6 D 6 A 0	0E2	0,$R_{DF\rightarrow 1}\rightarrow R_D$, Ind
		2 8 1 C 6 D 6 B 0	0E2	0,$R_{DF\rightarrow 1}\rightarrow R_D$, Ind
		2 C 1 C 6 D 6 A 0	0E3	0,$R_{DF\rightarrow 1}\rightarrow R_D$, Ind
		2 C 1 C 6 D 6 B 0	0E3	0,$R_{DF\rightarrow 1}\rightarrow R_D$, Ind
RLV	0E0	0 3 1 8 3 0 E 7 1	0C0	M-Data → W
IVP1	0E1	0 0 2 1 3 0 E 7 3	108	M-Data → P
SCS	0E2	0 3 5 A 2 0 E 6 4	0D0	S+1 → S
		0 6 4 4 2 8 E 6 2	020	P+1 → P,MA
SP1	0E4	0 7 5 D 0 0 0 0 0	0E8	
		0 1 1 E 0 0 0 0 0	0F0	
		0 1 1 F 0 0 0 0 0	0F8	
		0 1 2 0 0 0 0 0 0	100	
CTRL	0E8	0 4 0 0 5 E 6 0 0	003	W→SYNC
		0 4 0 0 4 0 F 1 D	003	$IS\overline{E}R$
		0 4 0 0 6 0 F 8 6	003	$S\cdot\overline{L}_8$ → S
		0 4 0 0 4 0 F 2 D	003	$IS\overline{E}S$
		0 4 0 0 6 0 F 4 6	003	$S\cdot\overline{L}_4$ → S
		0 4 0 0 6 0 F 8 5	003	SVL_8 → S
		0 4 0 0 4 0 F 4 D	003	PMA
RCS	0F0	0 4 0 0 4 0 0 0 0	003	$R_S\rightarrow R_D$(NO-OP)
		2 C 0 0 4 0 B E 0	003	$R_D\rightarrow P$ (X→ISE)
		2 C 0 0 5 D B 6 0	003	$R_D\rightarrow$DSPLY
		2 E 4 4 9 9 B 6 0	024	$R_D\rightarrow I$
		2 8 0 A 6 0 B C 0	052	$R_D\rightarrow P$
		2 8 0 A 6 0 B E 0	052	$R_D\rightarrow P$(X→ISE)
		2 C 0 0 6 0 B D 0	003	$R_D\rightarrow S$
		0 4 0 0 4 0 0 0 0	003	$R_S\rightarrow R_D$(NO-OP)

Mnemonic	Add	M Code	Next Address Group	Operation
RCD	0F8	2 C 0 0 6 0 7 7 0	003	$0,0 \rightarrow R_D$
		2 C 0 0 6 0 7 D 0	003	$R_{DU}, 0 \rightarrow R_D$
		2 C 0 0 6 0 7 F 0	003	$R_{DL}, R_{DU} \rightarrow R_D$
		2 C 0 0 6 0 B 9 0	003	$CNSL \rightarrow R_D$
		2 C 0 0 6 0 7 6 0	003	$0, R_{DL} \rightarrow R_D$
		2 C 0 0 6 D 6 C 0	003	$R_{DE \rightarrow 0}, R_{DF} \rightarrow R_D$, Ind
		2 C 0 0 6 0 B A 0	003	$S \rightarrow R_D$
		0 4 0 0 4 0 0 0 0	003	
RC0	100	2 C 2 1 0 D 6 7 0	109	$R_{D-1} \rightarrow$ Ind
		0 3 E 1 4 0 0 0 0	10A	
		2 C 0 0 6 D 6 E 0	003	$R_{DE \rightarrow 0}, 0 \rightarrow R_D$, Ind
		2 C 0 0 6 D 6 F 0	003	$R_{DE \rightarrow 0}, 1 \rightarrow R_D$, Ind
		2 C 0 0 6 D 6 7 0	003	$R_{D-1} \rightarrow R_D$, Ind
		0 3 E 0 4 0 0 0 0	102	
		2 C 0 0 6 D 6 6 0	003	$R_{D+1} \rightarrow R_D$, Ind
		0 0 2 1 A 0 E 0 0	10C	$S \rightarrow W$
IVP2	108	0 4 2 4 5 8 6 0 0	123	$W \rightarrow MA$
CMPL	109	2 C 0 0 6 C B 8 0	003	$\overline{R}_D \rightarrow R_D$
RLR	10A	2 C 0 0 4 F B 6 0	003	$R_D \rightarrow$ Ind
		2 C 0 0 6 F 6 6 0	003	$R_{D+1} \rightarrow R_D$, Ind
ADDS1	10C	0 0 0 F 2 0 F 3 E	078	$W+R_D \rightarrow R_D$, Ind
DTIM	10D	0 4 0 6 B 3 E 7 1	035	I/O Data $\rightarrow$ W
SHI	10E	0 3 5 A 2 0 F E 5	0D0	$SVL_E \rightarrow S$
DTIR	10F	3 4 0 0 7 3 B 9 0	003	I/O Data $\rightarrow R_D$
R0	110	3 C 0 0 4 F 2 0 0	003	$R_D - R_S \rightarrow$ Ind
		3 C 0 0 4 C 3 0 0	003	$R_D \cdot R_S \rightarrow$ Ind
		0 4 0 0 4 0 0 0 0	003	
		3 C 0 0 4 C 0 0 0	003	$R_S \rightarrow$ Ind
		3 C 0 0 4 C 4 0 0	003	$R_D \oplus R_S \rightarrow$ Ind
		3 C 0 0 4 F 1 0 0	003	$R_D + R_S \rightarrow$ Ind
		0 4 0 0 4 0 0 0 0	003	
		3 C 0 0 4 C 5 0 0	003	$R_D VR_S \rightarrow$ Ind
		3 C 0 0 6 F 2 0 0	003	$R_D - R_S \rightarrow R_D$, Ind
		3 C 0 0 6 C 3 0 0	003	$R_D \cdot R_S \rightarrow R_D$, Ind
		0 4 0 0 4 0 0 0 0	003	
		3 C 0 0 3 C 0 0 0	003	$R_S \rightarrow R_D$, Ind
	11C	3 C 0 0 6 C 4 0 0	003	$R_D \oplus R_S \rightarrow R_D$, Ind
		3 C 0 0 6 F 1 0 0	003	$R_D + R_S \rightarrow R_D$, Ind
		0 4 0 0 4 0 0 0 0	003	
		3 C 0 0 6 C 5 0 0	003	$R_D VR_S \rightarrow R_D$, Ind

Mnemonic	Add	M Code	Next Address Group	Operation
FIX	120	0 6 4 4 3 8 E 6 2 1 C 0 0 6 0 1 0 4	020 003	P+1 → P, MA R_D → + R_S → R_D
MPY	122	0 0 1 7 8 0 F 0 D	0BC	0 → S_L
IVP3	123	0 0 0 A 7 0 E 6 3	123	M-Data → P (D_F → ISE)

GA-16/110/220 mnemonics list C

A

ABRT	ABORT (Memory Overlay Active)
ADR3E	CNSL SWS/Internal Mask & 64K Mode
ADR3F	CNSL-TTY
ADRL	Address Load Enable
ALCL	ALU to Console (Register Data)
ALIOA	RALU Output to I/O ADDR
ALIOD	RALU Output to I/O Data
ALMA	RALU Output to Memory Address
ALMD	RALU Output to Memory Data
ALSYN	RALU Output to SYNC
ASYNC	SYNC Strobe

B

BCK8	Gate Delayed CK8
BKINT	Break Interrupt
BKRS	Break Reset
BUSY	Memory Busy F/F for DMA(SPC-16)(CPU2)

C

C1,C2, C4, C8 C9	DMA Timing Counter Outputs (Binary)
C8C1	C8 and C1
C8C2	C8 and C2
CA+	Processor 'A' Clock
CA-	Processor 'B' Clock ($\overline{A}$)
CCK	Control Clock (100 + ns. Into CA+)(CK2+)
CFBF	Multiple Crom Feedback
CIEN	CNSL Int. Enable Mask
CK1,CK2, CK8	Processor Timing Counter Outputs (Binary)
CLDS	Cold Start
CNTL	SPC-16 Control to I/O
CRDE	Crom/RALU Drive Enable (To T.D. Bus)
CRRE	Crom/RALU Receive Enable (From T.D. Bus)
CSLI	CNSL Int (Polled)
CTR3F	CNSL-TTY Control Strobe CTRL X '3F'

D

D3	Psuedo D3 (SPC-16)
DA	Data Avail (CNSL-TTY UART)
DACK,DACK1	SPC-16 DMA
DACKR	Dack Reset
DCHN,DDTP	SPC-16 DMA
DDTPR	DDTP Reset
DIBE	Data In-Bus Enable
DI3	Processor Internal D3
DMAEN	DMA #1 Enable
DMAER	DMA Error (From MPP Option)
DMAK	DMA 1 or 2 Ack.
DMAK1	DMA #1 Ack.
DMAK2	DMA #2 Ack.
DMC	DMA Active (SPC-16)
DMCR	DMC Reset
DMRQ1	DMA #1 Req.
DMRQ2	DMA #2 Req.
DPSO,DREQ	SPC-16 DMA
DTP	SPC-16
DTP1	Processor Internal DTP

E

ENTER	CNSL "enter" Sw.

F

FAP	SPC-16
FE	Framing Error (CNSL-TTY UART)
FGLP	Foreground Status
FTCH	Processor I-Fetch Indicator

I

IACK	SPC-16
IHMRQ	Inhibit Memory Req.
IHLD	SPC-16
IHOLD	"IHLD" Internal
INACK	SPC-16 "IACK"
INB00 thru INB15	SPC-16

I

INDAL	Int. Data to RALU
INT	Interrupt Req. to Crom
IOA3E	I/O Addr's X'3E' Decode
IOA3F	I/O Addr's X'3F' Decode
IOBSY	I/O Busy
IODAL	I/O Data to RALU
IOR	I/O Reset (CNTL X'3E' Decode)
IORAD	I/O 'Read' F/F
IPD0 thru IPDA	Crom/RALU M Code Interface Bus
IPDA thru IPDH	Crom/RALU M Code Interface Bus
IPLSW	IPL Sw. Remote
IPOD3	Inhibit Poll/D3
IPRS	SPC-16
IPSI	SPC-16
IPSO	SPC-16
IREQ	SPC-16
ISE	SPC-16
ISEO	ISE Control from RALU

L

LR00 thru LR09	Buffered T.A. Bus (Mem. Addr.) CPU2 to SCI
LSADD	Load Start Addr.

M

MD00 thru MD15	Memory Data Bus
MDAL	Mem. Data to RALU
MDRY	Mem. Data Ready
MEMPR	Mem. Protect
MEMPY	Mem. Parity Error
MREQ	Mem. Req.
MS1K	Most Sig. 1K Mem. Block Decode/ Latch
MSTR	SCI Mem. Start Read
MWRE	Mem. Write Enable
MWRT	DMA Write Select (SPC-16)
MYPR	My Priority (Interrupt)
M0 thru M9	Crom/RALU Data Bus
MA thru MF	Crom/RALU Data Bus

N

NC48	Not C8 or C4
NIIE	N.I. Int. Enabled
NIIEC	N.I. Int. Mode Control
NIIR1	Ext. N.I. Int. #1
NIIR2	Ext. N.I. Int. #2

O

OTB00 thru OTB15	SPC-16 Out Bus

P

PFD	Power Fail Detect
PMA	Pulse Monitor Alarm
POLL	SPC-16
POLLC	Poll Clock ("Poll" and not "IHLD")
POLLI	Internal Poll
PRRT	Processor Reset From -110

R

RCINT	Reset Cnsl. Int.
RCSW	Read Cnsl. Sws.
RD	CNSL-TTY Rcv. Data
READ	SPC-16
RESET 4	Processor Internal Reset
REST, REST2, REST3, REST4	Processor Internal Reset
RI	Cnsl-TTY Rcv. Input (UART)
RMW	Read-Modify-Write
RQIN	SPC-16 DMA
RSET+	Buffered RSET-
RSET-	Bi-directional Processor Reset
RTCKE	Real-Time Ck. Enable
RTCKI	Real-Time Ck. Input
RTCKI (ENABLE)	Real-Time Ck. Enable (RTCKE)
RTIV	Return Indirect Vector
RUN	Processor Run Mode

S

SCRCK	
SEMI	Semiconductor Mem.
SFEC	System Safe Line
SLMD	SCI Read Sel.
SMBRY	SCI BSY/RDY
SMBSY	SCI BSY/RDY
SPAL	Special Input to RALU (Restart)
STEP	Cnsl. Step Switch
STM1	SCI Start
STPCN	Stop Counter
STRD	Stop Mem. Read
SVI	Cnsl. 'SVI' Switch
SWA thru SWD	Encoded Cnsl. Reg. Select Sws.
SWALI	Cnsl. Sw. Data to Crom/RALU
SYNC	SPC-16
SYRT	SPC-16

T

TACK	Buffered Proc. 'A' Clock Ref.
TA00 thru TA15	Mem. Addr. Bus
TBREQ	'B' Clock T.D. Bus Req. (CROM/ RALU)
TD00 thru TD15	Mem. Data Bus
TEST	SPC-16
THRE	XMTR Holding Reg. Empty (CNSL-TTY UART)
TRE	XMTR Reg. Empty (CNSL-TTY UART)
TRO	XMTR Reg. Output-Serial (CNSL- TTY UART)
TTYI	CNSL-TTY Int.
TYIEN	CNSL-TTY Int. Enable Mask
TYR	CNSL-TTY Framing Error to Master Reset (UART)

W

WAIT	Wait Inst. Indicator
WIPRS	Wait for 'IPRS'
WRIT	SPC-16
WRTS*, WRTS+, WRTS-	SCI Write Strobe Edge Select (*220-Ground "WRTS-" Jumper "WRTS*" to "WRTS+" 330-+5 to "WRTS" Jumper "WRTS*" to "WRTS-")

X

XD	XMTR Data (CNSL-TTY)
+5VB	+5 (BATT B.U. Option)
+8V	Crom/RALU VDD Supply
+10 µsec	TTY Paddle Board Chopper Drive
20MH	20MHZ System Master Clock
20MH+1	20MHZ System Master Clock
20MHZ	20MHZ System Master Clock
64KM	64K Mem. Mode

early GA-16/110/220 configurations D

Figures D-1 and D-2 show the configurations of the compact chassis built prior to November 1976. The major changes made in currently shipped units provide additional space between CPU-1 and CPU-2 modules to accommodate a redesigned SCI module. (See Section 3, Figure 3-2.) Tables D-1 and D-2 show connector assignments for early compact chassis in both GA-16/110 and GA-16/220 configurations.

Figures D-3 and D-4 show the configurations of jumbo chassis built prior to October 1976. To provide more space between CPU-1 and CPU-2 modules to accommodate redesigned SCI module an I/O slot has been eliminated in currently shipped units. Table D-3 and D-4 shows connector assignments for early jumbo chassis in both GA-16/110 and GA-16/220 configurations.

Figure D-5 shows the priority interrupt chain wiring for early compact and jumbo chassis.

The circled numbers are keyed to Table 2-5, Section 2, for descriptions of controls and indicators.

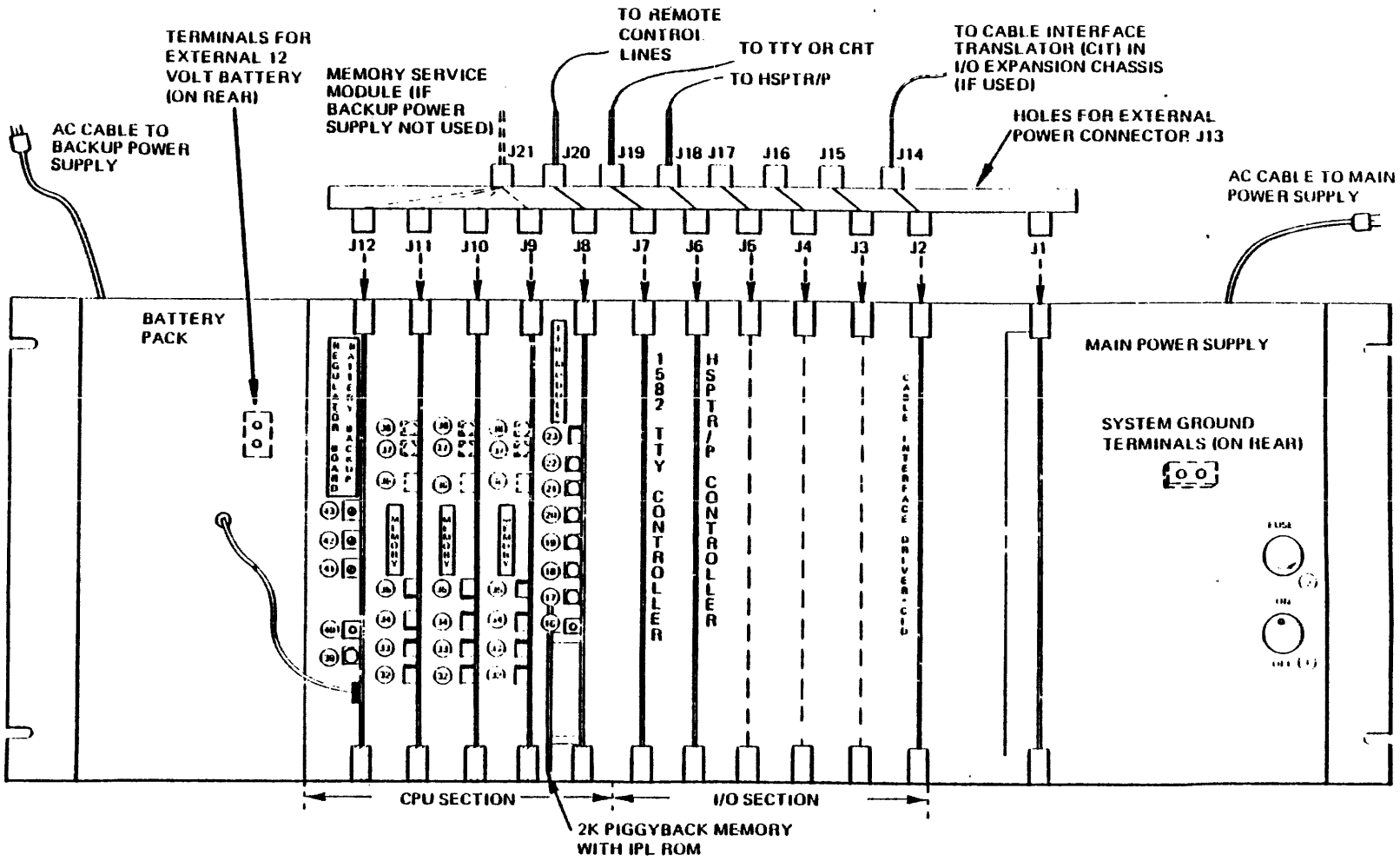


Figure D-1. GA-16/110 System in Compact Chassis (No MPP)

88A00509A-B

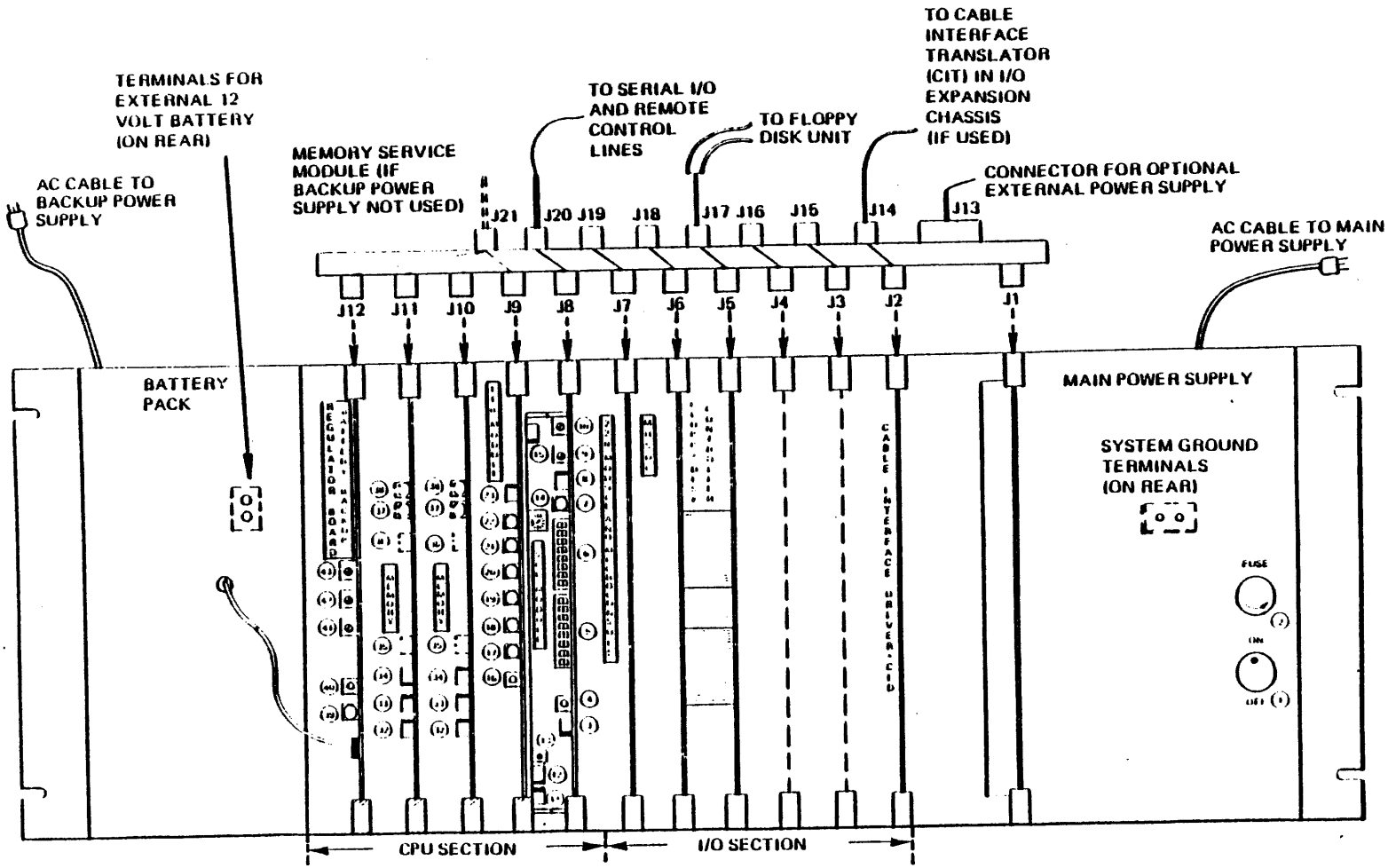


Figure D-2. GA-16/220 System in Compact Chassis (No MPP)

Table D-1. Connector Assignments for Compact Chassis, GA-16/110 Computer System

Front/Module		Rear/Interface	
J1	Main power supply	-	AC input connector and system grounding
J2	Cable Interface Driver (CID)	J13	Holes for connector for optional external power supply cable.
J3	I/O controllers or shorting boards.	J14	Cable I/O expansion chassis (6 ft. maximum).
J4			
J5	Shorting boards are used in empty slots between controllers or CID to maintain interrupt chain continuity.	J15	Paddleboards and cables interfacing controllers (in J3 through J7) to peripheral devices.
J6		J16	
J7		J17	
J8	110 module with optional piggyback memory and IPL ROM.	J18	
J9	4K/8K memory module or DMT controller (no write protect).	J19	
J10	4K/8K memory modules or Memory Parity Protect module.	J20	Access to cold-start (CLDS-) memory mode (64KM+), and remote control lines (IPLSW-, PFD-, RSET-, RUN-) control lines.
J11			
J12	J12 also used for battery backup regulator module.	J21	Memory Service Module if auxiliary power supply is not used.
-	Auxiliary power supply and battery pack (use regulator module in J12).	-	AC cord and terminals for external 12-volt battery on rear of auxiliary power supply.

Table D-2. Connector Assignments for Compact Chassis, GA-16/220 Computer System

Front/Module		Rear/Interface	
J1	Main Power Supply	-	AC input connector and system grounding terminals on power supply.
J2	Cable Interface Driver (CID)	J13	Connector for optional external power supply cable.
J3, J4	I/O controllers or shorting boards.	J14	Cable to I/O expansion chassis (6 ft. maximum).
J5, J6	Shorting boards are used in empty slots between controllers or CID to maintain interrupt chain continuity.	J15, J16, J17, J18, J19	Paddleboards and cables interfacing controllers (in J3 through J7) to peripheral devices.
J7	MHSDC controller, if used, should go into J7.		
J8	220 module with optional System Console Interface		
J9	110 module with optional piggyback memory.	J20	Serial I/O paddleboard with connector to TTY or CRT also provides access to cold-start (CLDS-) jumpers and remote control lines (IPLSW-, PFD-, RSET-, RUN-).
J10, J11	4K/8K memory modules or Memory Parity Protect Module.	J21	Memory Service Module if auxiliary power supply is not used.
J12	J12 also used for battery backup regulator module.	-	AC cord and terminals for external 12-volt battery on rear of auxiliary power supply.
-	Auxiliary power supply and battery pack (use regulator module in J12).		

Table D-3. Connector Assignments for Jumbo Chassis, GA-16/110 Computer System

Front/Module		Rear/Interface	
J1	Cable Interface Driver (CID).	J19	Cable to I/O expansion chassis, 6 ft. maximum.
J2	I/O controllers or shorting boards. Shorting Boards are used in empty slots between controllers or CID to maintain interrupt chain continuity.	J20	Paddleboards and cables interfacing I/O controllers (in J2 through J10) to peripheral devices.
J3		J21	
J4		J22	
J5		J23	
J6		J24	
J7		J25	
J8		J26	
J9		J27	
J10		J28	
J11	110 module with optional piggyback memory and IPL ROM	J29	Access to cold-start (CLDS-), memory mode (64KM+), and remote control lines (IPLSW-, PFD-, RSET-, RUN-).
J12	4K/8K memory module or DMT controller (no write protect).	J30	Memory Service Module if auxiliary power supply is not used.
J13	4K/8K memory modules or memory parity protect module. J18 also used for battery backup regulator module.	J32	Power cable from external power supply.
J14		J33	System grounding terminals.
J15		-	AC cord and terminals for external 12-volt battery on rear of auxiliary power supply.
J16			
J17			
J18			
-	Auxiliary power supply and battery pack (use regulator module in J18).		

Table D-4. Connector Assignments for Jumbo Chassis, GA-16/220 Computer System

Front/Module		Rear/Interface	
J1	Cable Interface Driver (CID).	J19	Cable to I/O expansion chassis (6 ft. maximum).
J2	I/O controllers or shorting boards. Shorting boards are used in empty slots between controllers or CID to maintain interrupt chain continuity.	J20	Paddleboards and cables interfacing I/O controllers (in J2 through J10) to peripheral devices.
J3		J21	
J4		J22	
J5		J23	
J6		J24	
J7		J25	
J8		J26	
J9	MHSDC controller, if used, should go into J10.	J27	
J10		J28	
J11	220 module with optional System Console Interface.	J29	Serial I/O paddleboard with connector for TTY or CRT. Also provides access to cold-start (CLDS-) jumper, and remote control lines (IPLSW-, PFD-, RSET-, RUN-).
J12	110 module with optional piggyback memory.		
J13	4K/8K memory modules or Memory Parity Protect Module. J18 also used for battery backup regulator module.	J30	Memory Service Module if auxiliary power supply is not used.
J14			
J15			
J16		J32	Power cable from external power supply.
J17			
J18		J33	System grounding terminals.
-	Auxiliary power supply and battery pack (use regulator module in J18).	-	AC cord and terminals for external 12-volt battery on rear of auxiliary power supply.

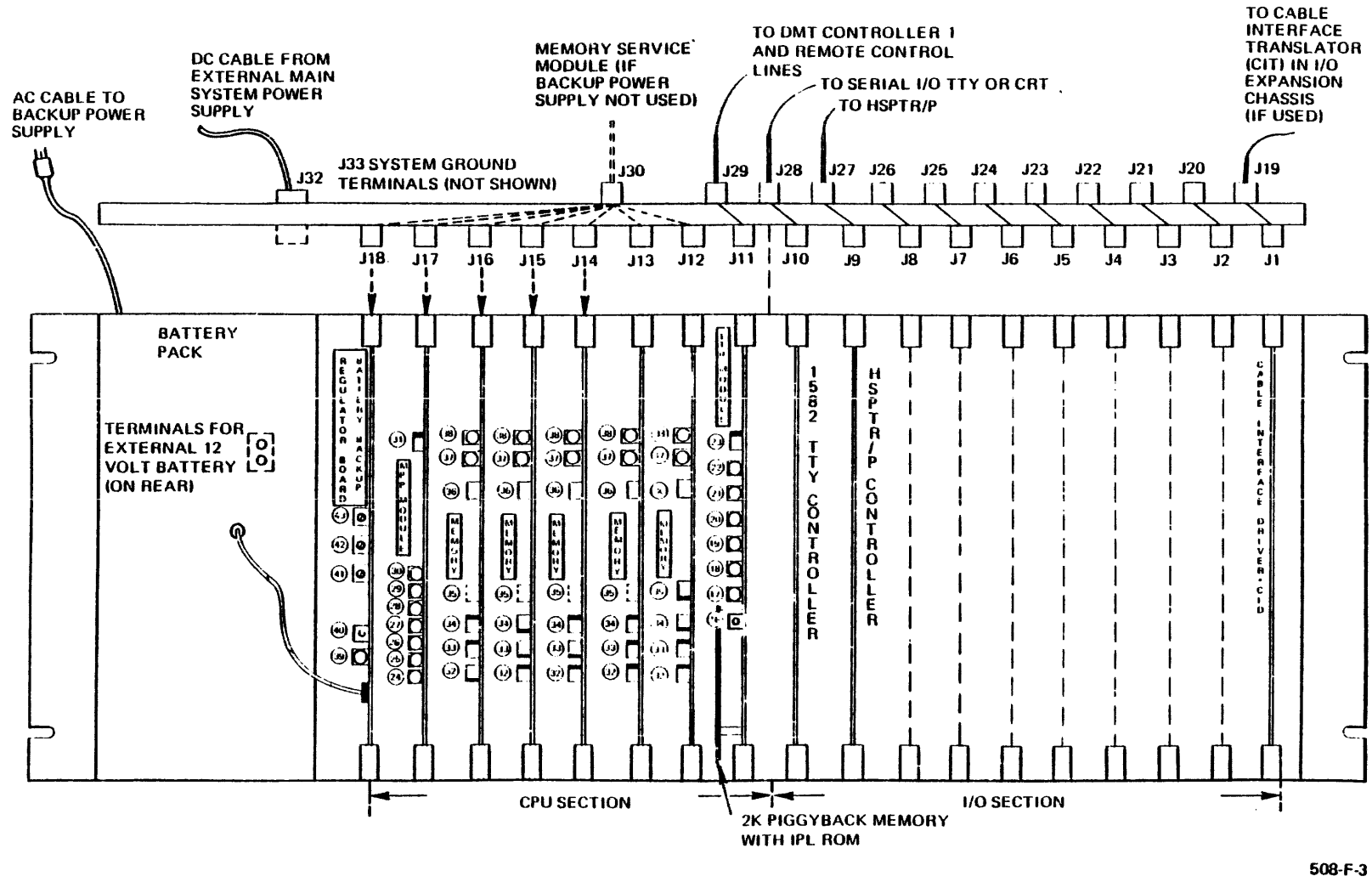


Figure D-3. GA-16/110 System in Jumbo Chassis (with MPP)

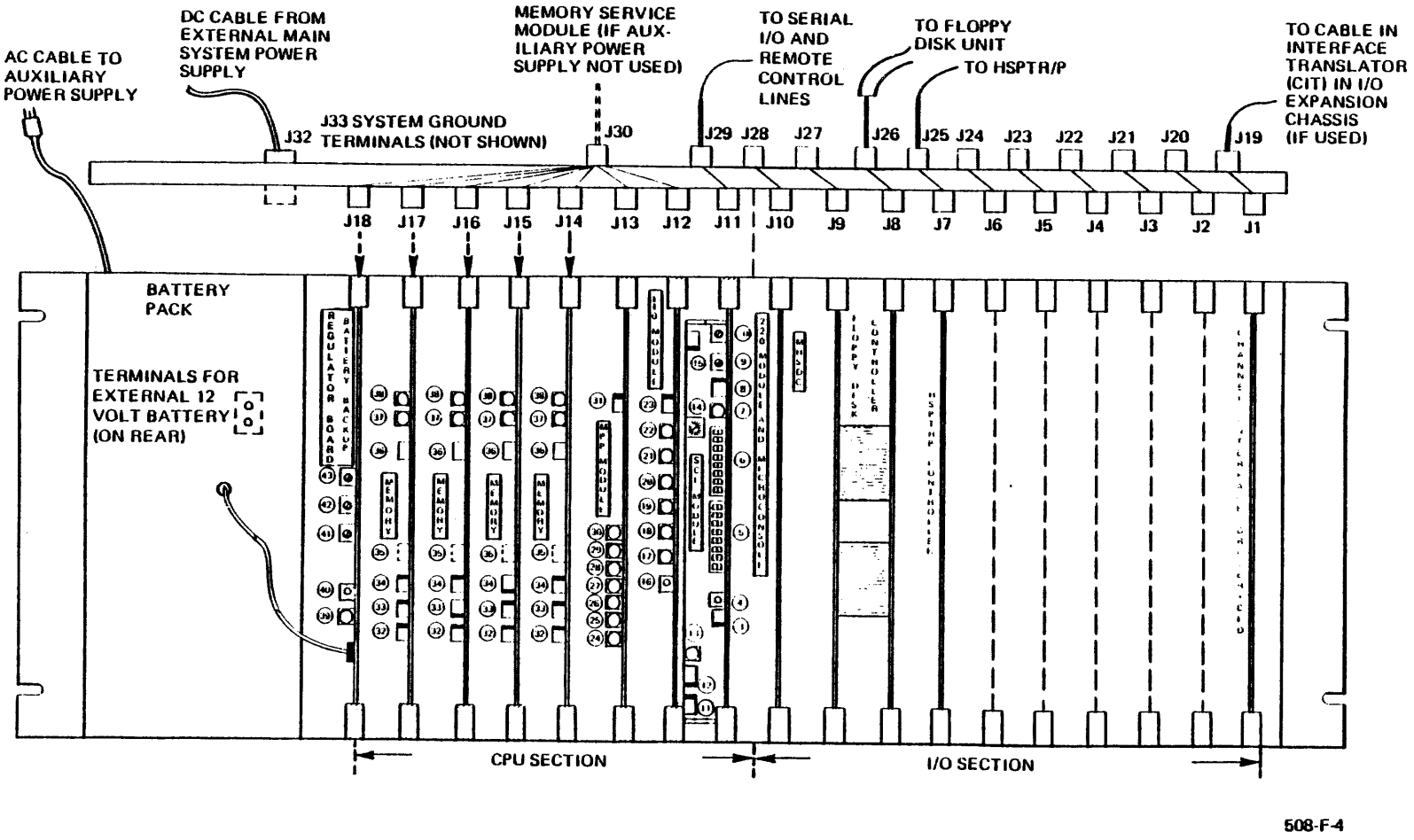


Figure D-4. GA-16/220 System in Jumbo Chassis (with MPP)

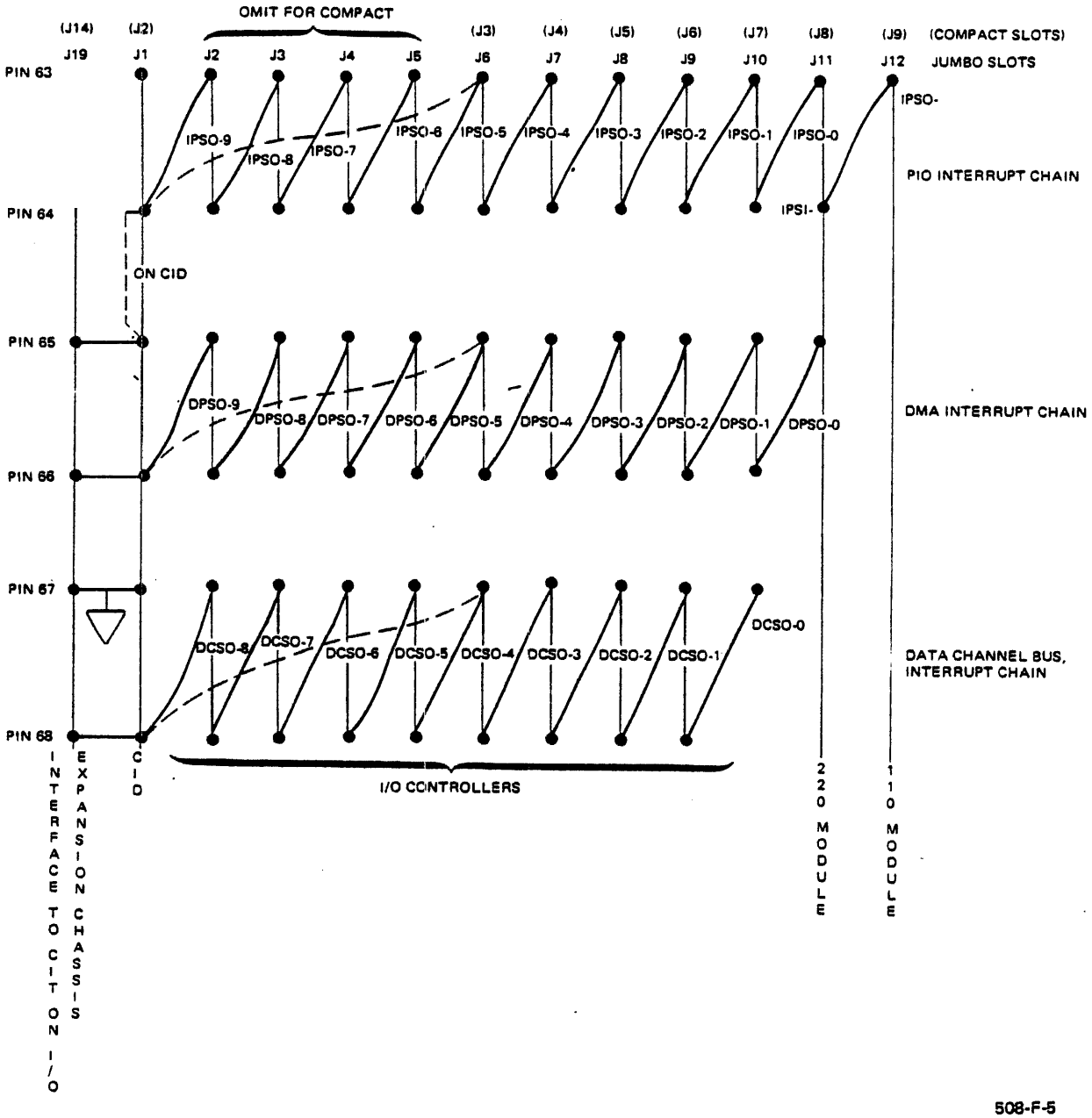


Figure D-5. Typical I/O System Interrupt Chain for Early Compact Jumbo Chassis

GA-16/220 microconsole signal index

E

ABRT-	J3-18	3C1
BKINT-	J1-14	3C4
BKRS-	J1-11	2B4, 3C4
BUSS-		4A1, 5A1
CLDS-	J3-20	2B1
CMD15+		4C2
CMD14+		4C2
CMD13+		4C2
CMD12+		4C2
CMD11+		4B2
CMD10+		4B2
CMD09+		4B2
CMD08+		4B2
CMD03+		5B2
CMD02+		5B2
CMD01+		5B2
CMD00+		5B2
CNSL-		2B4, 3A4
IPLF-		2A3
IPLS-		4A1, 5A1
IPLSW-	J1-22	2B4, 3B4
LR09+	J1-24	3D4, 5A4
LR08+	J1-23	3D4, 4C4, 5A4
LR07+	J1-21	4C4, 4B4, 5B4
LR06+	J1-19	4C4, 4B4, 5B4
LR05+	J1-20	4C4, 4B4, 5B4
LR04+	J1-17	4C4, 4B4, 5B4
LR03+	J1-18	4D4, 4B4, 5B4
LR02+	J1-15	4D4, 4B4, 5B4
LR01+	J1-16	4D4, 4B4, 5B4
LR00+	J1-13	4D4, 4B4, 5B4
LR09-		3D3, 5B4
LSADD+	J3-15	2D4
MD06-	J2-25	5D1
MD05-	J2-27	5C1
MD04-	J2-26	5C1
MD03-	J2-14	5B1
MD02-	J2-2	5B1
MD01-	J2-20	5B1
MD00-	J2-16	5B1

MD15-	J2-10	4D1
MD14-	J2-8	4D1
MD13-	J2-13	4C1
MD12-	J2-6	4C1
MD11-	J2-24	4B1
MD10-	J2-4	4B1
MD09-	J2-22	4B1
MD08-	J2-18	4B1
MD07-	J2-28	4D1
MDOT+		3D1, 4A1, 5A1
MDOT-		3C1, 4A1, 5A1
MSTR-	J1-12	3B4
MS1K+	J1-9	3B4
PRRT+	J3-22	2B4
RASL-		3D1, 4A1, 5A1
SLMD+	J1-7	3D4
SEMI-	J3-17	3C1
SMBSY-	J3-13	3A1
SMDRY+	J3-14	3B1
STM1-	J1-8	3D4
TD15-	J2-9	2D1
TD14-	J2-7	2D1
TD13-	J2-11	2D1
TD12-	J2-5	2D1
TD11-	J2-23	2C1
TD10-	J2-3	2C1
TD09-	J2-21	2C1
TD08-	J2-17	2C1
TD04-	J2-1	2C1
TD03-	J2-13	2C1
TD02-	J2-1	2C1
TD01-	J2-19	2C1
TYR-	J1-5	3B4
WRT-		3A1, 4A1, 5A1
WRTS+	J3-10	3A4
WRTS-	J3-9	3A4
WRTS8	J3-11	3D1
64KM+	J3-16	2D4

test and verify programs F

The enclosed list of applicable Test and Verify programs is supplied to allow a user to order those T&V's required for each individual system configuration.

The model number provided is the basic 5-digit code. To correctly order a T&V, two additional digits must be appended to this 5-digit code in the following format:

XTXXXmf

where:

x = Alpha or Numeric digit
m = Medium of the requested T&V
f = Format of the requested T&V

The Medium of the T&V may be specified as:

L = Listing
C = Card Deck
P = Paper Tape
M = Magnetic Tape

The Format may be any of the following:

S = Source Code
B = Binary Code
P = Program Generation System (PGS)
X = Mixed Source

It must be noted that not all T&V's are available in all Mediums and Formats.

Model No.	Revision No.	Title
4TC02	1	1575/1579 Sync Data Link Comm Controller
4TC03	1	1578 Synch Data Link Comm Controller T&V
4TC05	1	1574-0001 Isochronous Comm Controller T&V
5T000	1	GA-16 System Exerciser T&V
5T010	1	GA-16 Signed MPY/DIV T&V
5T100	1	GA-16 Series TTY T&V
5T110	2	GA-16 Series Memory Parity Protect T&V
5T14A	1	GA-16 Semiconductor Memory T&V
5T18M	2	GA-16 Floating Point Processor T&V (Section 1)
5T200	1	GA-16 Series High-Speed Paper Tape T&V
5T20A	1	GA-16 CalComp 936 (3374) Plotter T&V
5T30A	1	GA-16 H-P 7210 (3375) Plotter T&V
5T35A	1	GA-16 CalComp 3371/3372 Plotter T&V
5T40A	1	GA-16 CalComp 936 (3374) DMA T&V
5T600	1	GA-16 Card Reader T&V
5T610	1	GA-16 Card Punch 3314 T&V
5T700	1	GA-16 Series Line Printer T&V
5T70A	1	GA-16 3345 DMS Disk T&V
5T80A	1	GA-16 CCIF T&V
5T850	1	GA-16 Series 3346/3347 Disk T&V
5T870	1	GA-16 Series 3349 Floppy Disk T&V
5T880	1	GA-16 3341/43 Disk w/Sector Bits In Label T&V
5T890	1	GA-16 3346/3347 Double Density Disk T&V
5T900	1	GA-16 3342 Drum T&V
5T950	1	GA-16 ADDS Console 980 CRT T&V
5TA00	1	GA-16 Series 3331 Mag Tape T&V
5TE00	1	GA-16 801 Auto Calling Unit T&V
5TE05	1	GA-16 Hazeltine 2000 CRT T&V
5TE45	1	GA-16 ADC/DAC T&V
5TE55	1	GA-16 Digital Process I/O T&V
5TE75	1	GA-16 GAARD T&V

redesigned GA-16/110/220 processor boards G

This appendix provides a description of the newly designed GA-16/110 (CPU-1) and GA-16/220 (CPU-2) processor boards. These boards can be identified by the following assembly numbers:

- CPU-1 Board - 31D02573A
- CPU-2 Board - 31D02574A

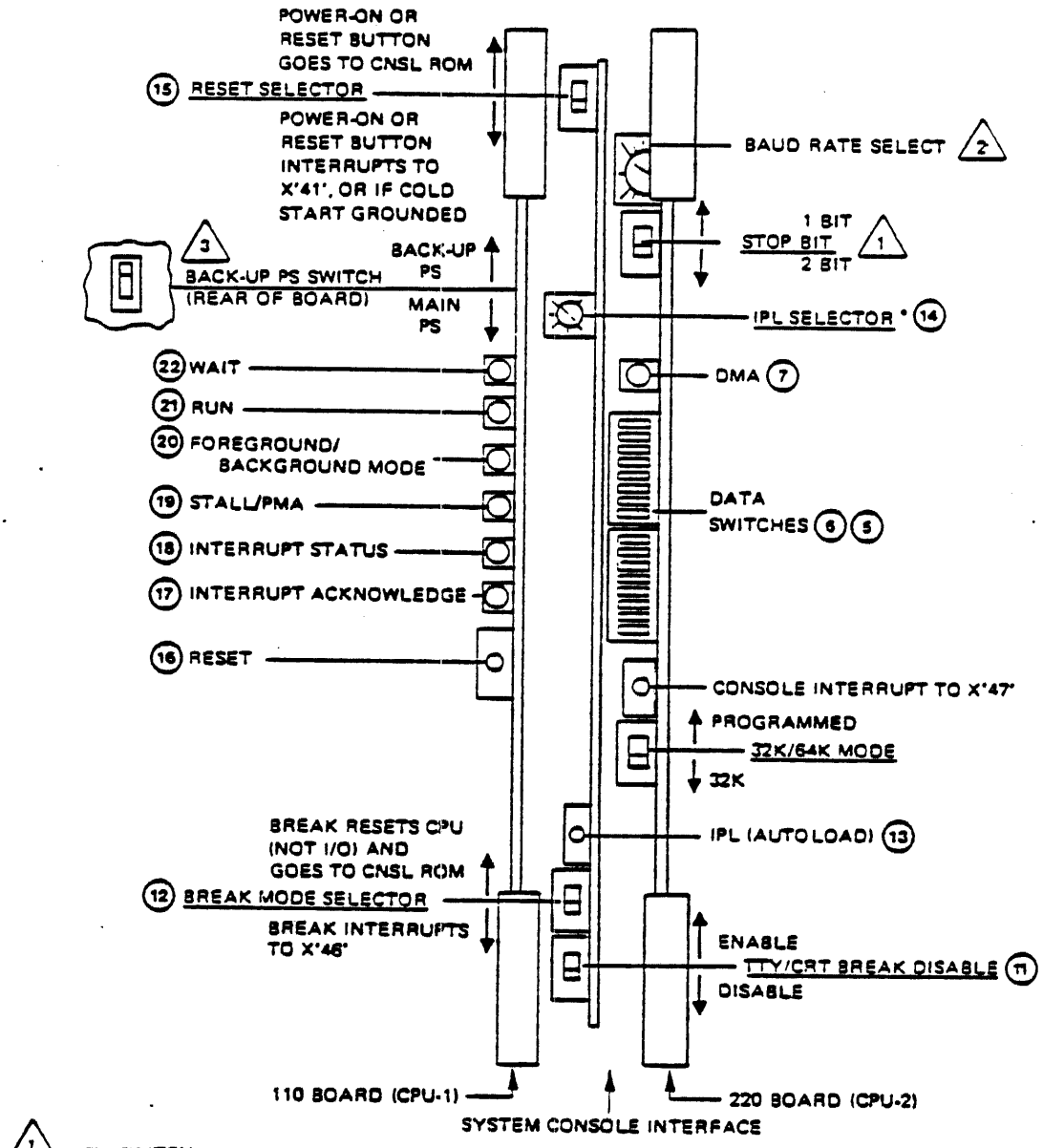
The main change in the CPU-1 board is the relocation of the Battery Back-Up Power Supply Switch which is now located between board coordinates D1 and E1. When this switch is in the up position, +5VB power for the memories originates from the back-up power supply or the batteries (when AC power is disconnected). When the switch is in the down position, memory power is supplied by the main power supply via the memory service module.

NOTE

The memory service module must be used when the switch is in the down position.

The main change in the CPU-2 board is the removal of the two-position 110/9600 baud switch. This switch was replaced by a rotary switch which provides fifteen selectable baud rates. A STOP BIT select switch has also been added which allows a selection of a one- or two-bit "STOP BIT."

Figure G-1 provides a composite view of the CPU-1 and CPU-2 boards containing these switches. The selectable baud rates are also shown in Figure G-1. The remaining switches and indicators have not been changed and can be found in Table 3-3 next to the key numbers (circled).



SWITCH SETTING	BAUD RATE	SWITCH SETTING	BAUD RATE
0	NOT USED	9	1200
1	50	A	1800
2	75	B	2400
3	110 *(TTY)	C	3600
4	134.5	D	4800
5	150	E	7200
6	300	F	9600 (*CRT)
7	600		
8	900		

508-G-1

Figure G-1. Redesigned GA-16/110/220 Processor Boards

EPROM programmer application H

The EPROM Programmer is a single board unit designed to allow data to be transferred from 220 RAM memory to EPROM memory boards utilizing TI TMS 2708JL or equivalent ultraviolet erasable EPROM parts. The Programmer will also program discrete TI TMS 2708JL, GA P/N 70A00369A02 parts. This description of the EPTOM Programmer application is based on GA Specification 82500761A.

To use the EPROM Programmer, the following equipment is required:

a. A 220 computer with serial I/O device (TTY or CRT) and RAM memory capacity to contain data to be transferred to the EPROM.

b. An ultra-violet lamp to erase EROM parts. (Model UVS-54, manufactured by Ultra-Violet Products, Inc., San Gabriel, California, or equivalent.)

H.1 MECHANICAL DESCRIPTION

The Programmer board occupies one memory slot in a standard 110/220 enclosure. The board requires five inches of clearance in front of the enclosure and 1-3/4 inches above it. The board has sockets to accept two EPROM chips, the piggyback EPROM board, or the 16K EPROM board.

Figures H-1, H-2, and H-3 contain installation drawings of the Programmer. Figure H-1 shows the installation for programming discrete EPROMS (70A00369A02), Figure H-2 shows the installation for programming the EPROM piggyback board (32D02532A), and Figure H-3 shows the installation for programming the 4/8/16K EPROM board (31D02566A).

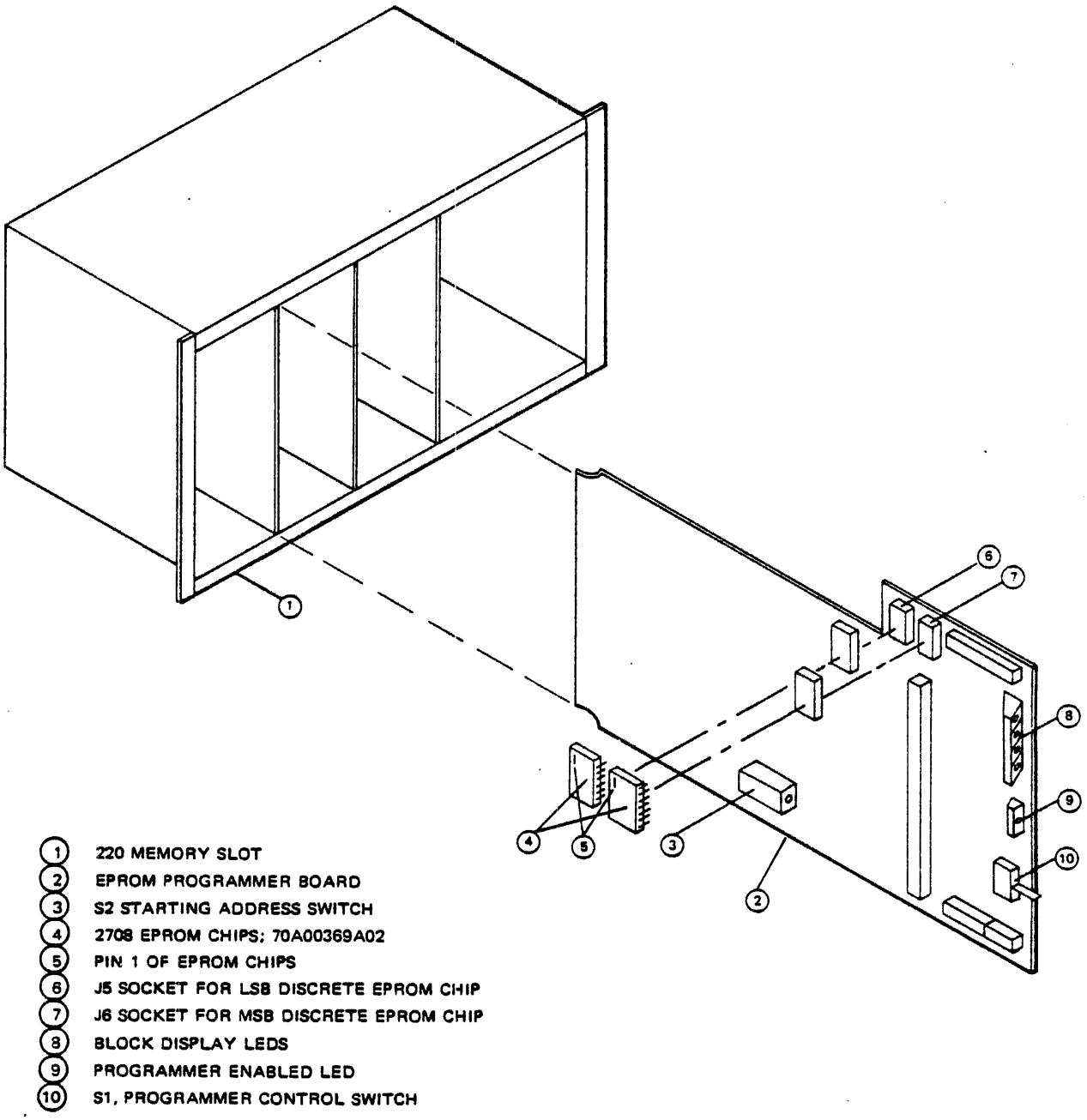


Figure H-1. Programming Discrete EPROMS

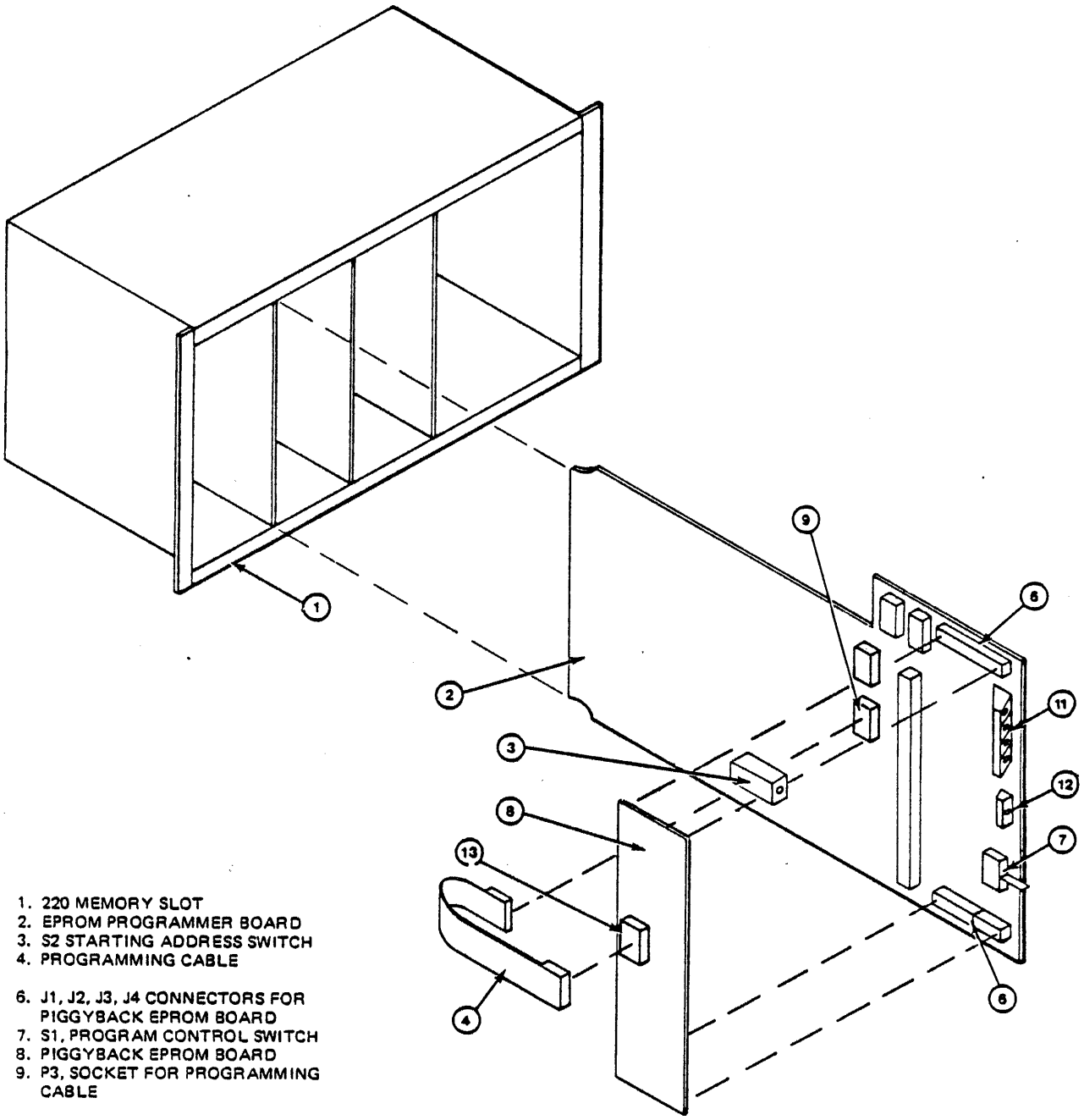
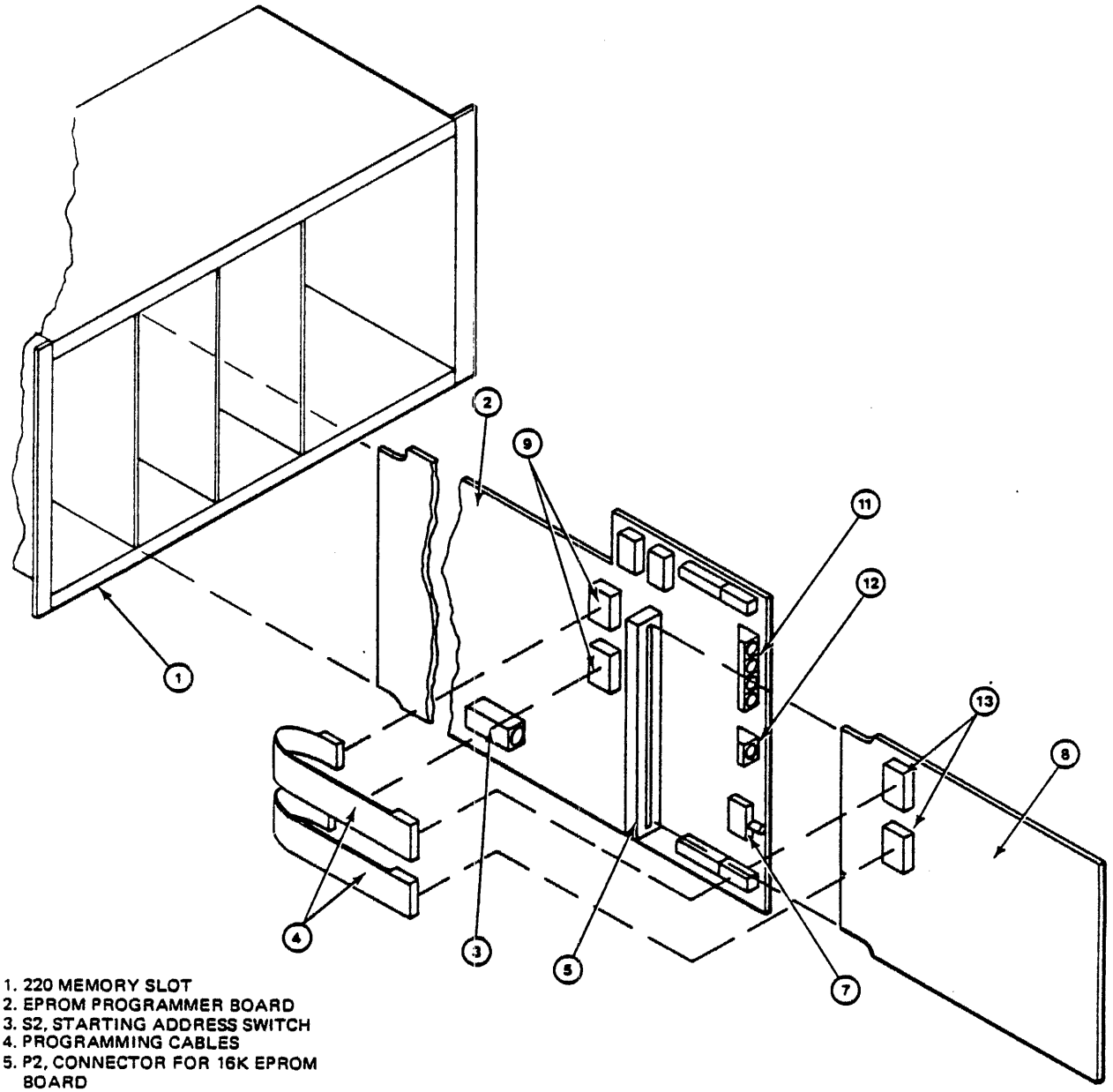


Figure H-2. Programming Piggyback Board 31D92532A



1. 220 MEMORY SLOT
2. EPROM PROGRAMMER BOARD
3. S2, STARTING ADDRESS SWITCH
4. PROGRAMMING CABLES
5. P2, CONNECTOR FOR 16K EPROM BOARD

7. S1, PROGRAM CONTROL SWITCH
8. 16K EPROM BOARD
9. P3, P4 SOCKETS FOR PROGRAMMING CABLES

11. BLOCK DISPLAY LEDS
12. PROGRAMMER ENABLE LED
13. PROGRAMMING SOCKETS ON 16K EPROM BOARD

509-H-3

Figure H-3. Programming 4/8/16K EPROM Board 31D02566A

H.2 ELECTRICAL DESCRIPTION

The Programmer board operates on the standard 110/220 memory bus. It uses 2048 addresses above the switch-selectable starting address. The starting address of the Programmer board can be selected on 4K boundaries with a hexadecimal switch, S2, mounted on the board (refer to Table H-1). The first 512 words in the Programmer are assigned to a Read-Only-Memory that contains a Write/Verify/Control program. The second 512 words are dedicated to a Control Function block. The remaining 1024 addresses are assigned to the 1K block of EPROM to be accessed (see Figure H-4 and Table H-1).

Table H-1 shows the distribution of the Write/Verify/Control program, the Control Functions block, and the 1K block assigned to EPROM data within the Programmer for each of 16 start addresses selected by S2.

H.2.1 CONTROL FUNCTIONS

Up to 16K of EPROM may be attached to the Programmer for programming. One 1K block of EPROM to be programmed is selected (addressed) by a store instruction that places a data word in the start address of the Programmer's Control Functions block. The lower four bits of the data word stored at the address of the Control Functions block will select one of 16 possible 1K blocks of EPROM (see Figure H-4). Table H-2 provides addresses for the 16 1K blocks of EPROM.

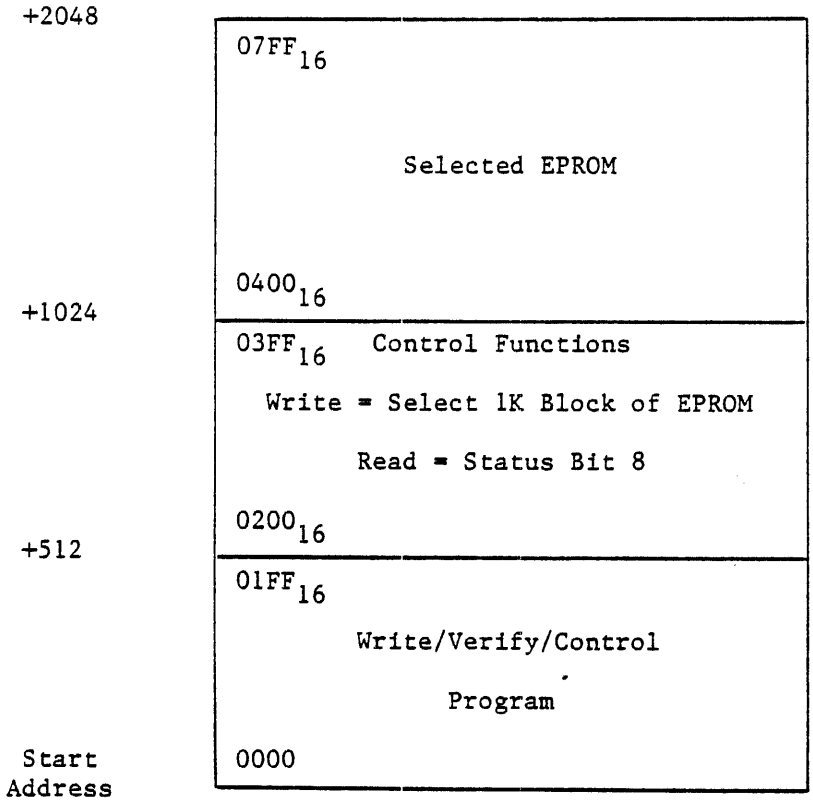
The Programmer has a manual control and indicator to allow the programming function to be enabled or disabled. The Programmer will be disabled by System Reset, Power Up Restart, or by pressing the momentary Press On, Press Off switch. Status is displayed by an indicator provided on the board; On indicates enabled.

H.2.2 PROGRAMMING

The EPROM is programmed by executing store instructions that place the data in the Programmer addresses assigned to the EPROM. This will generate a 25 volt program pulse, drive the CS/WE signal on the selected EPROM to +12 volt, and hold the Write cycle for 1 msec.

H.2.3 ERASE

Before programming, the EPROM is erased by exposing the chip through the transparent lid to high intensity ultraviolet light (wavelength 2537 angstroms). The recommended exposure is ten watt-seconds per square centimeter. This can be obtained by, for instance, 20 to 30 minutes exposure of a filterless Model UVS54 shortwave UV lamp about 2.5 centimeters above the EPROM. After erasure all bits are in the "1" state.



The address boundaries shown in the Programmer map are for an S2 start address selection of 0000. Table H-1 shows address boundaries within the Programmer for each of 16 possible start addresses by S2.

Figure H-4. Programmer Map

Table H-1. Programmer Address Boundaries For S2 Selections

S2 Position	Control Program From	To	Control Functions Block From	To	EPROM From	To
0	0000	01FF	0200	03FF	0400	07FF
1	1000	11FF	1200	13FF	1400	17FF
2	2000	21FF	2200	23FF	2400	27FF
3	3000	31FF	3200	33FF	3400	37FF
4	4000	41FF	4200	43FF	4400	47FF
5	5000	51FF	5200	53FF	5400	57FF
6	6000	61FF	6200	63FF	6400	67FF
7	7000	71FF	7200	73FF	7400	77FF
8	8000	81FF	8200	83FF	8400	87FF
9	9000	91FF	9200	93FF	9400	97FF
A	A000	A1FF	A200	A3FF	A400	A7FF
B	B000	B1FF	B200	BBFF	B400	B7FF
C	C000	C1FF	C200	C3FF	C400	C7FF
D	D000	D1FF	D200	D3FF	D400	D7FF
E	E000	E1FF	E200	E3FF	E400	E7FF
F	F000	F1FF	F200	F3FF	F400	F7FF

All address boundaries are in hex.

Table H-2. EPROM Addressing in 1K Blocks

(EPROM Installed in System)	(EPROM Installed on Programmer)	
EPROM HEX ADDR	BLOCK	HEX ADDR FROM PROGRAMMER
*0000 to 03FF	0	Start ADDR + 400
**0400 to 07FF	1	Start ADDR + 400
**0800 to 0BFF	2	Start ADDR + 400
**0C00 to 0FFF	3	Start ADDR + 400
1000 to 13FF	4	Start ADDR + 400
1400 to 17FF	5	Start ADDR + 400
1800 to 1BFF	6	Start ADDR + 400
1C00 to 1FFF	7	Start ADDR + 400
2000 to 23FF	8	Start ADDR + 400
2400 to 27FF	9	Start ADDR + 400
2800 to 2BFF	A	Start ADDR + 400
2C00 to 2FFF	B	Start ADDR + 400
3000 to 33FF	C	Start ADDR + 400
3400 to 37FF	D	Start ADDR + 400
3800 to 3BFF	E	Start ADDR + 400
3C00 to 3FFF	F	Start ADDR + 400

(X'400' is added to the start address on the Programmer board because the 1K of storage directly following the Programmer start address is dedicated to the Write/Verify/Control program and the Control Functions block.)

* = Discrete EPROM Parts

** = Piggyback EPROM Board

all = 16K EPROM Board

H.3 PROGRAM DESCRIPTION

The Control program included on the Programmer board provides five functions: Program and Verify, Erase Verify, Verify, Test 1 and Test 2. The program is initiated by transferring control from the console to the selected starting address of the Programmer board.

Example 1: From console enter XOOOG (CR) where X is the 0-F hex digit selected on the starting address switch.

This will result in an output to the serial I/O device 'CC?'. The desired control command can now be inputted preceded by an '$'. The program will then ask for a block number, 'BLOCK'. The block number is a hexadecimal digit assigned to each 1K of EPROM to be accessed (refer to Table H-2). If a single block is to be accessed, enter the single hex digit. If multiple blocks are to be accessed, enter the first and last hex digits separated by a slash (see Example 2).

Example 2: Single Block 1 (CR) (for EPROM addresses 400_{16} to $07FF_{16}$)

Multiple Block 1/3 (CR) (for EPROM addresses 0400_{16} to $0FFF_{16}$).

If needed, the program will then ask for a From address ('FROM'). This will be the first address of the block of data to be transferred from the 110/220 RAM memory to EPROM.

H.3.1 PROGRAM AND VERIFY COMMAND

Example 3:

PROGRAM OUTPUT	OPERATOR RESPONSE
CC?	$PROG (CR)
BLOCK	1/3 (CR)
FROM	0400 (CR)

If the programmer is disabled it will display:

ENABLE PROG.	PRESS PROG.
LFCR	ENABLE SW.

If the programmer is enabled it will display:

LFCR	NONE

Example 3 will result in 3K words of data starting at address $(0400)_{16}$ to be transferred to EPROM. After each 1K block has been programmed a word for word comparison is done. If no errors are found, an assurance message is printed, 'Verified Block X', and the next block will be programmed. If an error is found, the error is displayed as in Example 4.

Example 4: ERROR

FROM	BLOCK	ADDR	IS	S/B
0406	1	006	1224	1234

After an error message is printed the program will wait for an operator input. Space will continue to verify the rest of the block. Carriage return will cause a return to the start of the program.

H.3.2 ERASE VERIFY COMMAND

Example 5:

PROGRAM OUTPUT	OPERATOR RESPONSE
CC?	$ERASE (CR)
BLOCK	1/3 (CR)

Example 5 will result in testing 3K of EPROM for erasure (all bits high). If no errors are found, an assurance message will be printed for each block 'Verified Block X'. If an error is found, the error is displayed as in Example 6.

Example 6:

BLOCK	ADDR	IS	S/B
1	006	FFDF	FFFF

After each error message is printed the program will wait for an operator input. Space will continue the erase test. Carriage return will terminate the test and return to the start of the program.

H.3.3 VERIFY COMMAND

Example 7:

PROGRAM OUTPUT	OPERATOR RESPONSE
CC?	$VERIFY (CR)
BLOCK	1/3 (CR)
FROM	400 (CR)

Example 7 will result in 3K words starting at address (0400)$_{16}$ to be compared with 3K words of EPROM. If an error is found, it will be displayed as in Example 4.

H.3.4 TEST COMMAND

Example 8:	PROGRAM OUTPUT	OPERATOR RESPONSE
	CC?	$TEST1 CR -or-
		(TEST 2 CR)
	BLOCK	1/2 CR

Example 8 will result in Blocks 1 and 2 being tested for programmability. The test programs all bits to zero in the first location then scans all other locations for zeros. If none are found, the second location is programmed to all zeros and the unprogrammed part scanned. This continues until all locations are zeroed. If no errors are found, the assurance message is printed after each block. If an error is found, the error message is printed as in Example 4. The From display is the address that was programmed to zeroes and the Block/ADDR display is the address where the error was found. If the above two addresses are the same it indicates that the word could not be zeroed or programmed. Test 1 starts at the lowest numerical address and works up. Test 2 starts at the highest address and works down.

H.3.5 RETURN TO CONSOLE COMMAND

Example 9:	PROGRAM OUTPUT	OPERATOR RESPONSE
	CC?	$CNSL CR

Example 9 will cause program control to be returned to the console program.

H.4 STEP-BY-STEP PROGRAMMING EXAMPLE FOR PIGGYBACK BOARD 31D02532A

A program located between Hex addresses $(0400)_{16}$ and $(0B00)_{16}$ is to be transferred to a 2K piggyback EPROM board:

1. Insure that the EPROM parts are erased by exposing them to the ultra-violet lamp for 45 minutes.

2. Install the Programmer board in any unused memory slot in the 220 computer. Set the Hex starting address switch, S2, to any unused location (for this example, set S2 to 7). See Figure H-2, item 3 and Table H-1. The Programmer's control RDM program will now start at hex location $(7000)_{16}$. Install the piggyback EPROM board on the Programmer and connect the programming cable (see Figure H-2).

3. Apply power to the 220 system. Load and verify the program to be transferred with TTY, PTR, etc. Transfer control from the 220 Console program to the programmer control program by entering "7000G" on the keyboard. The control program will respond with "CC?".

4. Verify that the EPROM parts are erased (all outputs high) by selecting the Erase Verify command as shown in Section 4.2 If any errors are found, erase the EPROMS again as in Step 1, above. After verifying that the EPROMS are erased, enable the Programmer and select the Program and Verify command as shown in Section 4.1. It will take about two minutes to program each 1K block. After two minutes the program will output 'Verified Block 1; (if no errors) and and after two more minutes the program will output 'Verified Block 2' and 'CC?' indicating that the programming operation is complete and no errors were found.

H.5 STEP-BY-STEP PROGRAMMING EXAMPLE FOR DISCRETE EPROMS 70A00369A

A program located between Hex addresses $(0400)_{16}$ and $(07F0)_{16}$ is to be transferred to 1K of EPROM:

1. Insure that the EPROM parts are erased by exposing them to the ultra-violet lamp for 45 minutes.

2. Install the Programmer board in any unused memory slot in the 220 computer. Set the Hex starting address switch, S2, to any unused location (for this Example set S2 to 7). See Figure H-1, item 3 and Table H-1. The control ROM program will now start at Hex location $(7000)_{16}$. Install the EPROM chips on the programmer. Observe that Pin 1 of the chips are up as in Figure H-1.

3. Apply power to the 220 system. Load and verify the program to be transferred with TTY, PTR, etc. Transfer control from the 220 Console program to the programmer control program by entering "7000G" on the keyboard. The control program will respond with "CC?".

4. Verify that the EPROM parts are erased (all outputs high) by selecting the Erase Verify command as shown in Section 4.2. If any errors are found, erase the EPROMS again as in step 1, above. After verifying that the EPROMS are erased, enable the Programmer and select the Program and Verify command as shown in Section 4.1. It will take about two minutes to program each 1K block. After two minutes the program will output 'Verified Block 1' (if no errors) and 'CC?' indicating that the programming operation is complete and no errors were found.

NOTE

Discrete EPROMS use the low-order addresses in block 0.

H.6 STEP-BY-STEP PROGRAMMING EXAMPLE FOR THE 4/8/16K EPROM BOARD (31D02566A)

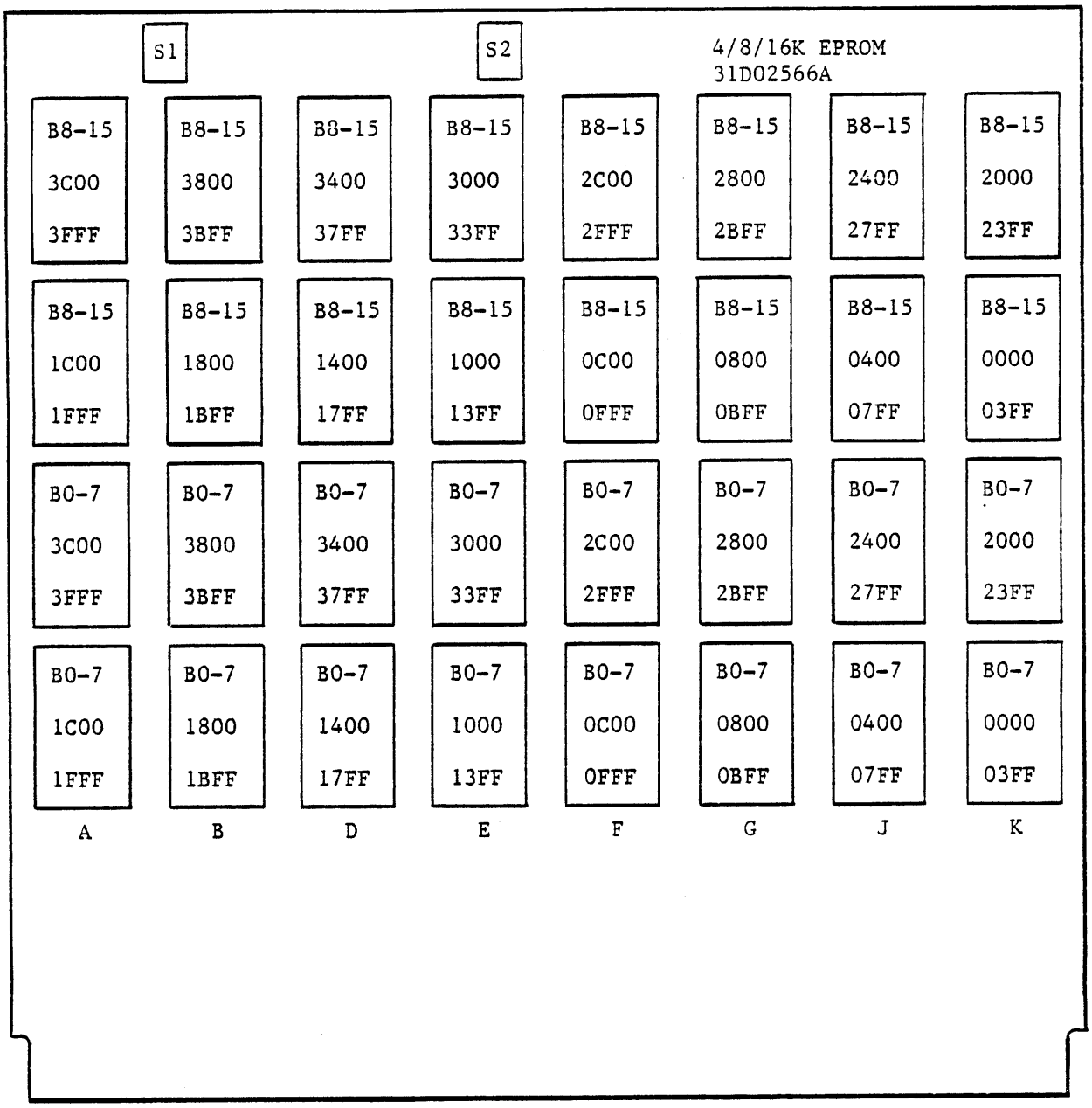
A program located between Hex addresses $(0000)_{16}$ and $(0FFF)_{16}$ is to be transferred to a 4K EPROM board:

1. Insure that the EPROM parts are erased by exposing them to the ultra-violet lamp for 30 minutes.

2. Install the Programmer board in any unused memory slot in the 220 computer. Set the Hex starting address switch, S2, to any unused location (for this Example, set S2 to 7). See Figure H-3, item 3 and Table H-1. The control ROM program will now start at Hex location $(7000)_{16}$. Install the EPROM board on the programmer and connect the programming cable (see Figures H-3, H-5 and H-6).

3. Apply power to the 220 system. Load and verify the program to be transferred with TTY, PTR, etc. Transfer control from the 220 Console program to the programmer control program by entering "7000G" on the keyboard. The control program will respond with "CC?".

4. Verify that the EPROM parts are erased (all outputs high) by selecting the Erase Verify command as shown in Section 4.2. If any errors are found, erase the EPROMS again as shown in step 1, above. After verifying that the EPROMS are erased, enable the Programmer and select the Program and Verify command as shown in Section 4.1. It will take about two minutes to program each 1K block. After two minutes the program will output 'Verified Block 0' (if no errors) and after two more minutes the program will output 'Verified Block 1' and so on until the last block (3) has been programmed and verified. 'CC?' will then be printed indicating that the programming operation is complete and no errors were found.



1. B0-7 refers to bits 0,1,2,3,4,5,6,7

2. 0000
   refers to hex address boundaries
   03FF

509-H-4

Figure H-5. EPROM Chip Location Reference to Bit Position and Address

S2 Block Size Select

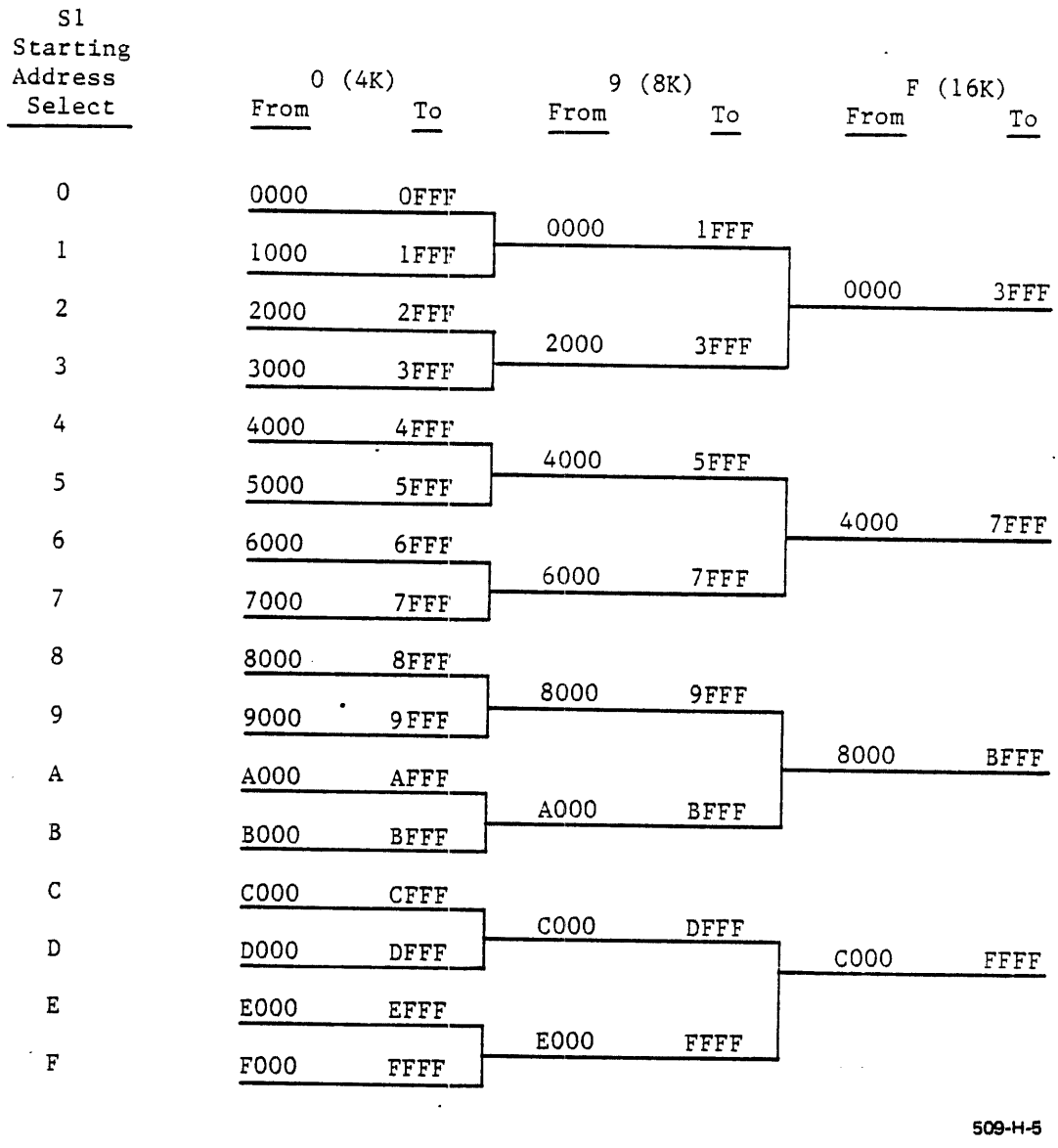


Figure H-6. 4/8/16K EPROM Board Address Switches

COMMENT SHEET

Document No. _____
Title: _____
Page: _____

FROM:

NAME: _____

BUSINESS ADDRESS: _____

_____

_____

Does this publication meet your requirements? Yes ☐ No ☐

If no, please explain. _____

_____

_____

_____

_____

Do you wish a reply? Yes ☐ No ☐

COMMENTS: (Describe any errors, suggested ideas, additions, or deletions etc. Please include page number.)
All comments and suggestions become the property of General Automation, Inc.

_____

_____

_____

_____

_____

_____

_____

_____

No Postage Stamp Necessary If Mailed In the U.S.A.
(See Other Side)
Fold On Dotted Lines And Staple

YOUR COMMENTS, PLEASE . . .

This publication serves as a reference for systems analysts, programmers and operators of General Automation systems. Your answers to the questions on the back of this form help us produce better publications for your use.

FOLD

First Class
Permit No. 423
Anaheim, California

BUSINESS REPLY MAIL NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY

General Automation, Inc.
1055 South East Street
Anaheim, California 92803

Attention: Publications Services

FOLD

CUT ALONG LINE

General Automation, Inc.
1055 South East Street
Anaheim, California 92803

GENERAL AUTOMATION

1055 SOUTH EAST STREET, ANAHEIM, CA 92805 TELEPHONE: (714) 778-4800