COPY NO. 220

Technical Manual

VMS 3200 SERIES

SEQUEL

Microdata

Customer Service Training

Technical Manual

VMS 3200 SERIES

SEQUEL

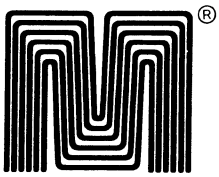
PROPRIETARY INFORMATION

THE INFORMATION CONTAINED HEREIN IS PROPRIETARY TO AND CONSIDERED A TRADE SECRET OF MICRODATA CORPORATION AND SHALL NOT BE REPRODUCED IN WHOLE OR PART WITHOUT THE WRITTEN AUTHORIZATION OF MICRODATA.

© 1982 MICRODATA CORPORATION
ALL RIGHTS RESERVED
TM-TRADEMARK OF MICRODATA CORPORATION
SPECIFICATION SUBJECT TO CHANGE WITHOUT NOTICE
PRINTED IN THE U.S.A.

FOR INTERNAL USE ONLY

82 1158



Microdata Corporation
2361 McGaw Avenue,
Irvine, California 92714
Telephone: 714/540-6730
Customer Service Training

TABLE OF CONTENTS

CONTENTS (Cont'd)

CONTENTS (Cont'd)

CONTENTS (Cont'd)

CONTENTS (Cont'd)

Section TABLE OF CONTENTS (Cont'd) Page

TABLE OF CONTENTS (Cont'd)

TABLE OF CONTENTS (Cont'd)

ILLUSTRATIONS

TABLES

TABLES (Cont'd)

SECTION 1

GENERAL DESCRIPTION

INTRODUCTION

This section includes a general description of the Microdata SEQUEL™ System. The following major subsystems make up the system hardware for the SEQUEL:

- Cabinet with AC power distribution
- DC power supply and blower
- Card cage with:
  - Central processing unit (CPU)
  - Diagnostic and maintenance processor (DMP)
  - Memory
  - Input/output (I/O system
- Disc subsystem
- Magnetic tape subsystem
- Printer subsystem
- Terminal subsystem

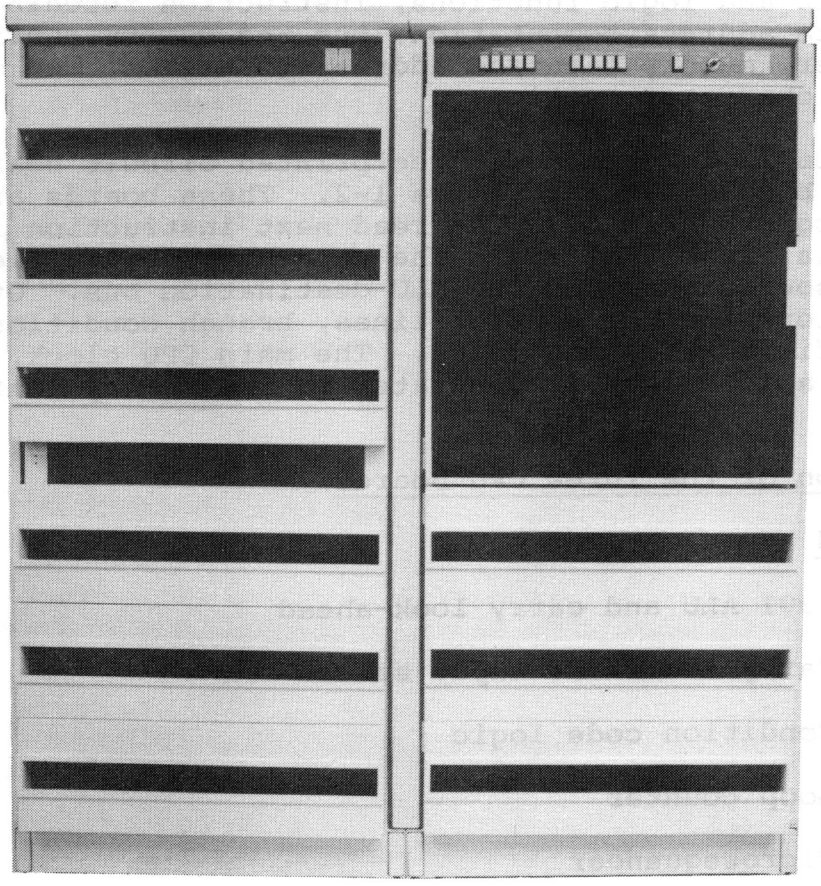


Figure 1-1. The Microdata Two-Bay SEQUEL™ Basic System

Central Processing Unit

The CPU is microprogrammable and is based on a bit slice architecture. It provides the following characteristics:

32 bit data path; CPU, I/O bus, memory

24 bit I/O memory address range (16 megabytes)

600 nanosecond memory cycle time

150 nanosecond microprogram cycle time

1-4 byte memory accesses in one cycle

64 bit main firmware word size

Hardware rotation capability

Character test function

There is a single, bi-directional bus for I/O and memory, which is synchronous to the CPU. Separate hardware exists for the arithmetic and logic functions, instruction fetching and decoding, data address computation, bus and memory control, the microsequencer, memory data and address interface, and data rotation and character testing.

The CPU is implemented on three printed circuit boards as shown in the block diagram, Figure 1-2. These boards are the arithmetic logic unit (ALU), the read next instruction (RNI), and the special function (SF). The three boards are tied together via the ALU source bus, and the ALU destination bus. Other interconnections include control lines, branch conditions, map branch, and firmware command bits. The main CPU clock, which synchronizes all boards, is generated on the memory control board.

Major Function of the Three CPU Boards

1. ALU Board

- 2901 ALU and carry look-ahead
- Carry and shift input select
- Condition code logic
- Loop counter
- Microsequencer

- Firmware command register
- Control clock logic

2. RNI Board

- Instruction fetch and parse picoprocessor
- Program address register
- Instruction registers
- Mapping ROMS (for FW branching)
- Address register files

3. Special Functions Board

- Bus and memory control
- Data Address registers
- Memory read/write data registers
- "X" register
- Rotate/character test

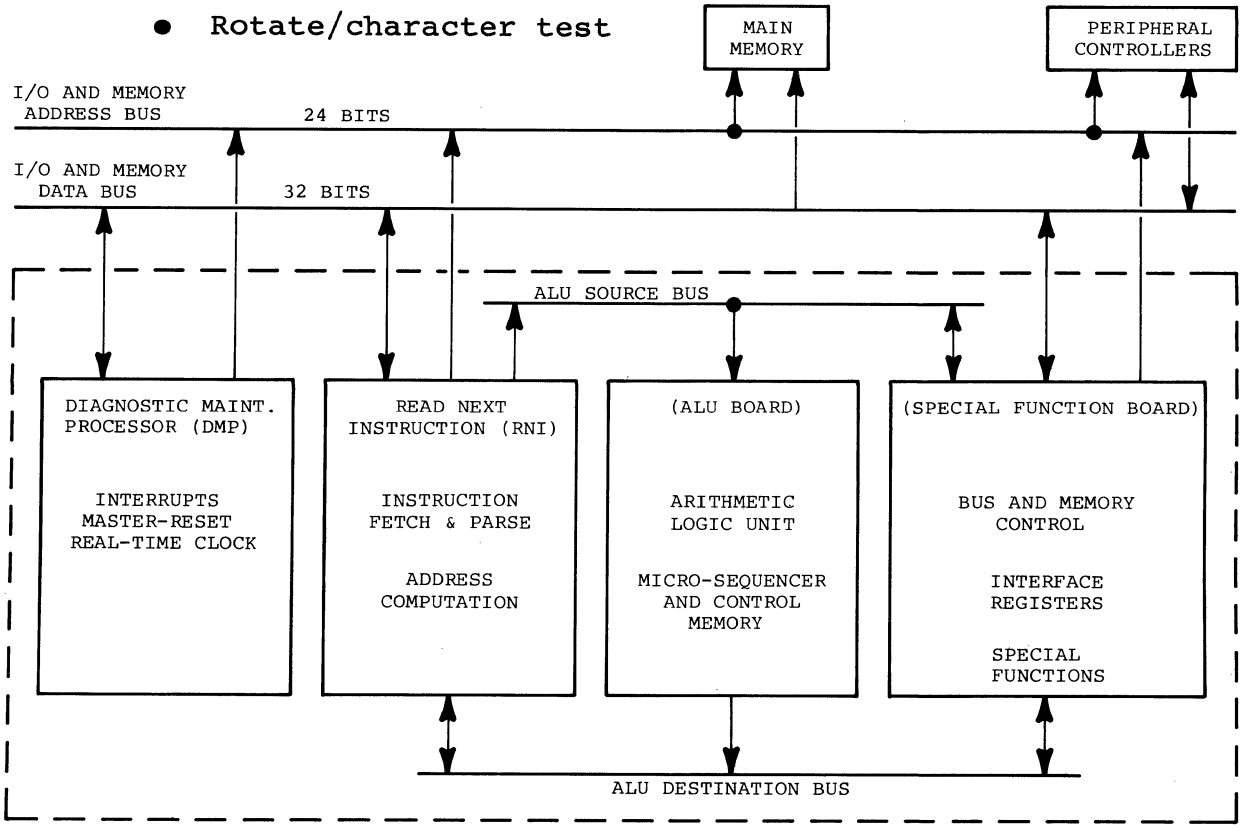


Figure 1-2. CPU Block Diagram

The ALU board also has 16 flags that can be set and tested by the firmware. These flags assist in keeping track of status, modes, etc. There is also an eight bit loop counter that serves a dual purpose. It can be set to any value from 0 to 255 and then be tested by the firmware and automatically be incremented. This is used to facilitate firmware loops with minimum housekeeping time. The second purpose of the loop counter is to provide the firmware with the capability of setting up a programmable direct branching capability.

Memory

The memory is an all semiconductor memory with error detection and correction. It is organized and addressable on a 4-bit /byte basis and provides a single cycle 32 bit access at any byte boundary. It has a 32 bit data width with a 600 nanosecond cycle time. There is a one bit/byte error correction with optional two bit/byte error detection. The memory is byte addressable to the 16 megabyte address range. There are 256 bytes per board (16K X 1 RAM's) with a two megabyte maximum using 256K byte boards.

Input/Output (I/O) System

The I/O system is based on intelligent controllers functioning through a direct memory access (DMA) system. Direct processor/controller communication is reduced to a simple start and halt orders.

The I/O system is also based on single bus architecture with parity checking. It has 16 full DMA channels and up to 256 I/O devices. The interrupt system supports up to three processors and there is an independent I/O processor for disc and communication controllers.

Diagnostic and Maintenance Processor (DMP)

The DMP is a Z80 microprocessor computing system that provides the interface between the operator, the diagnostic terminal and the main CPU. The DMP contains a Z80 microprocessor, 16K x 8 ROM memory, 4K x 8 RAM memory, I/O interface to all peripherals (including the diagnostic terminal), and interface to the main CPU and memory.

The DMP will perform the following functions:

- Execute firmware that resides in the ROM.
- Use the RAM memory for temporary storage and for execution of loadable software.

- Communicate with the CPU and the I/O devices through the I/O bus.
- Transfer data to and from the main memory in the direct memory access mode.
- Provide a programmable timer that will interrupt the main CPU in programmed intervals of 0.5 to 1024 milliseconds.
- Provide a master reset and power fail/restart interrupt, vectored to location '0000' of the Z80 ROM memory.
- Provide a 10Hz (10 times per second) interrupt to the Z80.
- Enable/disable external interrupts.
- Provide the time of day to the main CPU in deciseconds.
- Provide a 24-hour interrupt to the main CPU at midnight of every day.
- Provide communication to the local and remote terminals.
- Provide communication to the SEQUEL control panel.

DC Power System

The power system provides an uninterruptable power supply to be used for the semiconductor memories in the computer system. It uses a battery to maintain the output voltages during short interruptions of the AC power source.

The input is +20 to 40 volts, DC. In the normal power up mode, the input is unregulated, capacitor filtered voltage derived through the AC line through the basic power supply bias transformer. During power outages, the output is supplied by an internal 18-volt battery. The battery will supply the memory standby mode for a minimum of 20 minutes. Transition to and from battery operation is automatic and will not produce any output transients. The battery is a sealed lead acid type.

Output voltages are as follows:

+5.0 VDC at 1.0 to 16.0 amperes
+12.0 VDC at 0.1 to 5.0 amperes
-5.0 VDC at 0 to 0.20 amperes

A minimum current of one ampere is required on the +5 VDC output and 0.1 ampere on the 12V output of the supply to maintain regulation. The supply will not be damaged by a "no load" con-

dition of the +5 VDC output or 12V output. Under no load, the maximum output voltage will not exceed 6 and 14V respectively.

Environmental requirements for the power system are listed in Table 1-1.

TABLE 1-1

POWER SYSTEM ENVIRONMENTAL REQUIREMENTS

Ambient Temperature	
Operating	0°C to 50°C
Storage	-40°C to 70°C
Humidity	
Operating	5% to 90% non-condensing
Storage	5% to 95% non-condensing
Altitude	
Operating	12,000 feet
Storage and Shipping	40,000 feet

The power system unit is designed for 15-minute repair time in a field installation. This will be accomplished by using modular design concepts.

Cabinet and PCB Chassis

The standard cabinet is designed to house a PCB chassis for the CPU, memory and peripheral controllers, a basic DC power supply, two REFLEX© disc drives, and one reel-to-reel tape drive. Systems with more disc or tape drives or an expansion PCB chassis require additional cabinets of the same style. When installed, all cabinets are bolted together, forming an integral multi-bay cabinet.

1. Cabinet Dimensions

Dimensions for the standard cabinet are as follows:

Height, overall	58.50 in.
Width, overall	25.34 in.
Depth	36 in.

Panel Space	47.25 in.
Weight, Empty	350 lb.

2. Castors

Four 3-1/2-inch diameter castors, each independently swiveled 360 degrees, are provided. In addition, four leveling feet are provided to meet the cabinet stability requirements of UL 478.

3. PCB Chassis

The PCB chassis provides mounting for twenty-one 13-inch x 17-inch printed circuit boards. The chassis is mounted at the top of the cabinet with all boards vertical. Forced air enters at the bottom of the chassis and is exhausted at the top of the chassis.

4. Finish

All external cabinet surfaces will be painted with textured polane T paint per Microdata Specification Z20016006, colors to be specified.

5. Magnetic Tape Drive Door

A smoke colored transparent Plexiglas door is provided over the magnetic tape drive to inhibit dust intrusion. A safety interlock will prevent the door from being opened while the drive is in operation.

6. Operator Control Panel

Control switches and status indicators for the CPU and magnetic tape drive are located on a horizontal panel immediately above the tape drive door.

Special features of the SEQUEL are condensed in Table 1-2. This summary highlights the major subsystems that make up the system hardware for the SEQUEL.

TABLE 1-2

SEQUEL FEATURES

- 32 bit super-mini
- 600 nanosecond memory with ECC
- Intelligent I/O system
- Remote and on-line diagnostics
- High efficiency switching-mode power supply

Central Processing Unit (CPU)

- 32 bit data path
- 64 bit micro-instruction with simultaneous ALU control and branching
- 150 nanosecond micro-cycle
- 16 megabyte memory address range
- Calculation of operand addresses by RNI picoprocessor
- REALITY© oriented hardware flags

Memory

- 32 bits wide
- Byte addressable
- 600 nanosecond cycle
- Byte ECC
- 256K byte per board using 16K RAM's (1 megabyte with 64K RAM's)

I/O System

- Single bus architecture
- 6.7 megabyte/second transfer rate
- Supports multiple CPUs
- Channel command protocol
- 16 DMA channels
- 254 I/O devices
- Multiple I/O processors

I/O Processor

- Converts byte oriented devices to 32 bits
- Intelligent DMA multiplexer
  - Maintains memory addresses for discs and terminals
  - Supplies interrupts for multiple CPU's
  - Handles channel command protocol for discs and terminals
- Provides full command chaining in disc controller
- 2901 based microprocessor

Table 2-1. SEQUEL Features (Continued)

Disc Controller ● Relieves software of disc address calculation ● Supports on-line diagnostics ● Improves data recovery procedures ● Supports two REFLEX drives ● Z80 microprocessor based
Asynchronous Communications Line Controller (ACLC) ● Supports on-line diagnostics ● Supports eight modem channels at 9600 baud ● Z80 microprocessor based
Mag Tape and Printer Controller (MTP) ● Supports two tape drives and two line printers ● Provides full command chaining ● Supports on-line diagnostics ● Dual 2901 microprocessors (Two independent 8 bit machines)
Diagnostic and Maintenance Processor (DMP) ● Provides board level fault analysis ● Supports remote and online diagnostics ● Provides simultaneous drive to local and remote diagnostic terminals ● Provides special functions to CPU Power fail/auto restart Time of day clock Process time slice interrupt Interval timer ● Z80 microprocessor based
Power Supply ● 115 Ampere +5 volt capability ● Offline switching technology ● 60 to 70 percent efficiency ● Easy maintenance Slide mounted drawer Field replaceable modules Latched fault indicators

Functional Description

The microprocessor-based CPU is addressed to programmers and analysts responsible for coding the macro-instructions and routines (firmware) that will simulate the assembler language operating system. The firmware is coded in a special purpose, proprietary language, and is embodied in hardware PROMS in the system.

The role of the firmware is to interpret assembler language operating system instructions. Based on these interpretrations, the firmware will invoke the microcode (machine language) macro-routines that will effect the assembler language instructions (see Figure 1-3). The PROM-resident microcode consists of 64-bit instructions that are loaded into the instruction register for execution (refer to Figure 1-4).

The instruction register contains the firmware word that controls system activity for one instruction cycle. The instruction register output, 64 lines, is connected to the circuit elements whose functions are the "decoding" of the bit patterns to activate the desired operations. For each type of operation, the 64-bit instruction is organized into fields. Each field is transmitted to the appropriate decoder.

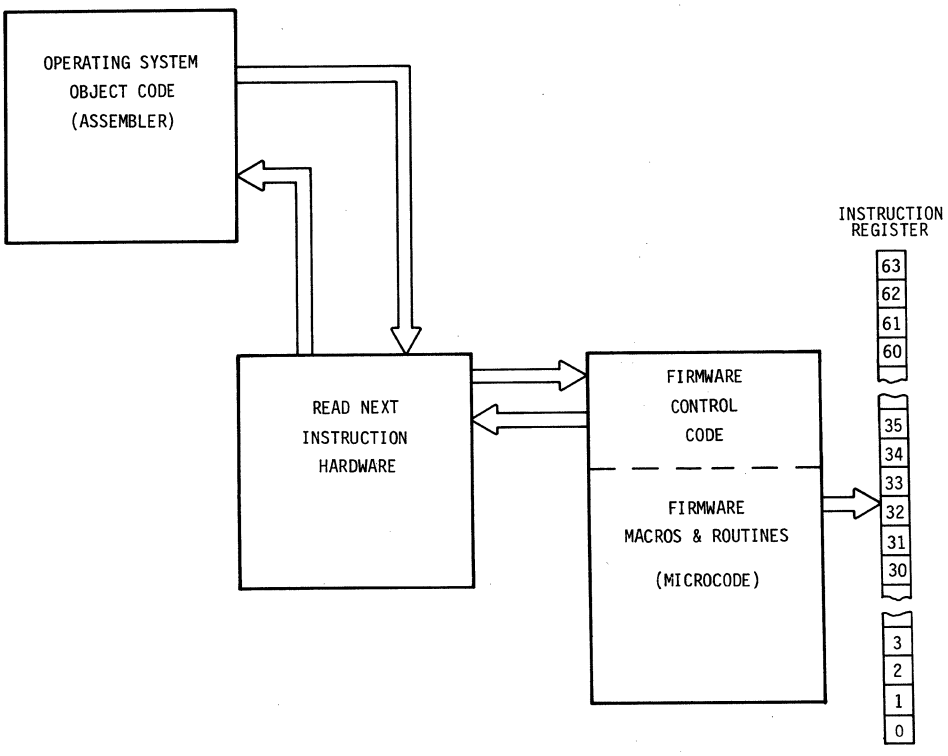


Figure 1-3. Microcode Routines for Assembler Language Instructions

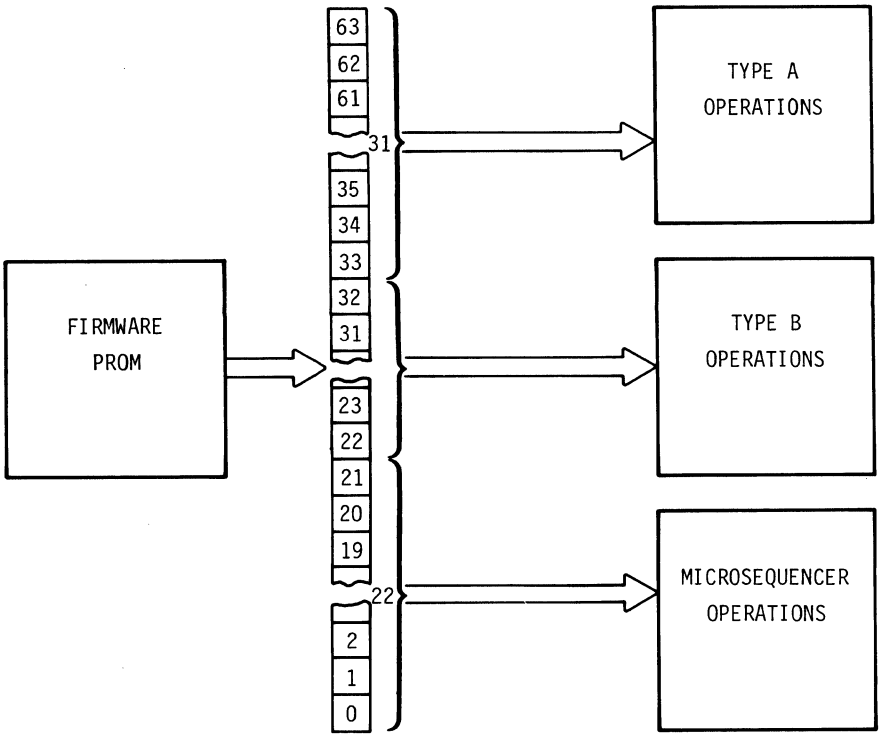


Figure 1-4. 64-Bit Instructions for Instruction Register

General Instructions Format

Each firmware instruction is organized into three basic fields: "A" field, "B" field, and microsequencer field. The "A" field (bits 64-33) consists of 31 bits that are used to implement one of three "A" type instruction. The "B" field (bits 32-22) consists of 11 bits that are used to implement one of seven "B" type instructions.

The microsequencer field (bits 21-0) consists of 22 bits that are used to implement the ordering of the sequence of firmware instructions execution, i.e., program control. This field will be the various branching commands (unconditional, conditional, etc.) that determine the next firmware instruction that will be executed. A summary of each instruction in the three fields is provided in Table 1-3.

TABLE 1-3

BASIC FIRMWARE INSTRUCTION FIELDS

"A" Field (Bits 64-33)	"B" Field (Bits 32-22)	Microsequencer Field (Bits 21-0)	Function
ALU			Instructions direct the arithmetic/ logical operations performed by the ALU on selectable external (to the microprocessor) and/or internal 32-bit words.
Rotate			Instructions direct the rotate operations performed by the rotate circuits on 32-bit words from memory or from the microprocessor.
Addr. Comp			Instructions direct the address computations operations performed by the address computation circuits on address data from the microprocessor and RNI instruction registers.
	Shift		Instructions support an A-type operation calling for the shifting one bit right or one bit left of a 32-bit value being loaded into the register file or G register. The B-type shift instruction controls the input-disposition of the bits that are shifted into the word when the A-commanded shift is executed.
	Memory read/ write		Instructions cause 4 bytes to be received (read) or sent to (write) main memory or a peripheral device.
	Multiply		Instruction provides a multiplication "utility." It is used in conjunction with A-type ALU operations.
	Divide		Instruction provides a division "utility" in a similar manner to the multiply described just previously.
	Character test		Instruction is used in conjunction with A-type ALU operations (logical) for a process that results in a yes/no determination of a "match between a predetermined character(s) and a character string being tested.
	Flags controls		Instructions deal with the setting, resetting, and/or testing of various system flags.
	CPU control		Instructions provide the facility for CPU-RNI communication.
		(Bits 21-0)	These 22 bits are used to implement the ordering of the sequence of firmware instructions execution, i.e., program control. This field will be the various branching commands (unconditional, conditional, etc.,) that determine the next firmware instruction that will be executed.

NAMEPLATE IDENTIFICATION

Figure 1-5 shows the nameplate which is affixed to the back of the left rear of the unit. The information provided includes the model name, top assembly part number, model ID, serial number, and power requirements as to VAC, amperes, and Hz. A listing of the top assembly part numbers is provided in the Appendix.

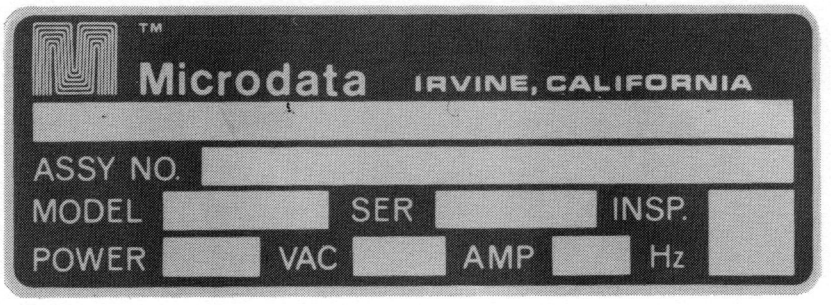


Figure 1-5. SEQUEL Nameplate

SECTION 2

PHYSICAL PLANNING AND INTERFACE REQUIREMENTS

INTRODUCTION

The Microdata SEQUEL Computer System is designed for ease and simplicity of installation. It requires a minimum of space and servicing. All elements of the central system are interconnected by a few easily handled cables. A single power outlet is required for each cabinet.

Installation planning is extremely important. It is not a complex engineering task nor is it a cursory procedure to be treated lightly. Most installations will only require common sense attention to the requirements of the system as a whole, and the relationship of the system to a particular facility and operation.

Proper planning and site preparation before equipment arrives will ensure the speediest installation. A well-planned, prepared installation site will circumvent costly post-delivery system modifications or facility changes, and will avoid extended periods of system inoperation.

SITE PREPARATION

A site layout grid is provided in the appendix to help determine the suitability of a floor plan and to position the SEQUEL system for the most efficient operation and servicing. Table 2-1 is a preinstallation checklist to make certain adequate preparations have been completed. Microdata will assist customers during actual equipment installation and will also provide assistance during installation planning.

Site Selection

Installation sites for SEQUEL systems may vary from compact rooms or corners of other work areas to spacious, air-conditioned computer centers. Sites may be open and readily accessible, or have restricted access for security reasons. The complete system may be centralized in a single room, or data terminals may reach out to extend the system throughout the building, across town or across the country.

Whatever the scope of the system, the major area of concern during the site planning stage is the central computer site. This must house the SEQUEL computer, tape storage units, data printer and one or more data terminals.

TABLE 2-1

SEQUEL PREINSTALLATION CHECKLIST

	Yes	No	Not Applicable
ENVIRONMENTAL CONTROL			
Heating system installed	☐	☐	☐
Air filters installed	☐	☐	☐
Air conditioning system installed	☐	☐	☐
Air filters installed	☐	☐	☐
Humidifier/dehumidifier installed	☐	☐	☐
Carpeting/furniture treated for static electricity	☐	☐	☐
Posted NO SMOKING in vicinity of computer	☐	☐	☐
ELECTRICAL POWER			
Equipment power lines/receptacles installed (separate from all other lines)	☐	☐	☐
Auxiliary power lines/receptacles installed for air conditioner, humidifier, etc.	☐	☐	☐
Lighting fixtures, switches, and controls installed	☐	☐	☐
Equipment power lines free of noise	☐	☐	☐
SITE LAYOUT			
Space allocated for			
SEQUEL system equipment	☐	☐	☐
Furniture	☐	☐	☐
Storage	☐	☐	☐
Equipment, furniture, and storage cables defined	☐	☐	☐
Cable troughs acquired	☐	☐	☐
Routing for data terminal cables defined	☐	☐	☐
Data terminal interface cables routed	☐	☐	☐
FIRE PREVENTION			
Sprinkler system installed	☐	☐	☐
Fire extinguishers:			
Acquired	☐	☐	☐
Positioned/installed	☐	☐	☐
Fire resistant storage cabinet/files acquired	☐	☐	☐
MISCELLANEOUS			
Window drapes/shades installed	☐	☐	☐
Glass tinting completed	☐	☐	☐
Telephone services (Modems)			
Scheduled	☐	☐	☐
Installed	☐	☐	☐

The size of the central computer site depends on four primary criteria:

System size: The room or area must be large enough to accommodate the SEQUEL equipment with certain minimum clearances for operation and maintenance.

Personnel: Additional floor space may be required for personnel activities in the immediate area. For example, operators, programmers or clerks may require desks, tables, or filing cabinets adjacent to the computer system.

Storage: Shelves, filing cabinets, or closets should be provided for auxiliary storage (i.e., operation and technical manuals, printer paper, magnetic tapes, forms, etc.).

Expansion: To protect the long-range economy of a system, provisions should be made to accommodate foreseeable future expansion at the central computer site. Additional electrical power and floor space may be required for equipment added or upgraded at a later time. Providing for future expansion during initial system installation can help avoid costly modifications and disruption of operations later.

Location

The physical location of the central SEQUEL computer site will most likely be determined by two main factors; proximity of the site to the using departments, and availability of suitable space at the facility. The first of these factors often is of secondary importance since the SEQUEL data terminals can be conveniently and economically located just about anywhere they are needed. Generally, it is advisable to locate the offices of programmers, operators, and other personnel who require direct physical access to the central system close to the central site. It is also desirable to locate the central site where all of the peripheral data terminals can be connected directly to the computer without using telephone lines or modems.

Normally, it is possible to select the central computer site from presently available space with a minimum of required facility modifications. Certain costs should be expected for preparing the site for installation of the SEQUEL system. For example, electric wiring modifications may be necessary, an air conditioner may be needed, or room partitions to separate the computer from adjacent work areas may be required. The nature and extent of these preparations will depend on the present conditions of the

site, the specific SEQUEL system configuration, and the requirements for decoration, security, etc.

ENVIRONMENT CONTROLS

All equipment in the SEQUEL system is designed to operate in a controlled atmosphere. Proper evaluation of the site can help avoid additional costs for upgrading environmental deficiencies that may exist in some localized area of a facility.

Environment factors of temperature, humidity and airborne dust in prospective installation sites should be evaluated during the planning stages before the SEQUEL system arrives. If necessary, temperature variations, humidity and airborne dust should be controlled before installation of the SEQUEL system.

Temperature Control

Temperature control is probably the most important environmental factor because the SEQUEL equipment is cooled by surrounding room air. The temperature in office buildings and most other likely installation sites is nearly always controlled to within allowable limits for the SEQUEL equipment. Nevertheless, a number of factors should be considered to determine the adequacy of existing temperature controls.

All electric equipment generates heat. In the SEQUEL system, this heat is discharged into the room and will tend to raise the ambient temperature unless the air conditioning system can handle it. BTU ratings for all SEQUEL equipment are provided on specification sheets in Table 2-2.

TABLE 2-2

SEQUEL EQUIPMENT BTU RATINGS

Single-Bay System	6,000 BTU/Hr
Two-Bay System (Min)	10,639 BTU/Hr
Four-Bay System (Max)	21,278 BTU/Hr
REFLEX II	1,800 BTU/Hr
Printer 300, 600 LPM	1,706 BTU/Hr
MATRIX Printer	1,200 BTU/Hr
PRISM II	365 BTU/Hr
PRISM IV	510 BTU/Hr
5750 Comm. Terminal	550 BTU/Hr
Body Heat Per Person	400 BTU/Hr
Additional Equipment	Calculate 3.41 BTUs per Watt

. Heat will also be contributed by other equipment in the room (i.e., electric typewriters, lights, and auxiliary data processing equipment). For every watt of electrical power consumed by the equipment, approximately 3.4 BTUs of heat will be generated.

Individuals occupying the room will contribute approximately 400 BTUs per person. This can be an important consideration if substantially more people will occupy the area after the computer is installed than occupied it before installation.

A window or glass wall area provides virtually no insulation against energy from direct sunlight. Drapes or shades should be used to protect the SEQUEL equipment from direct sunlight, since this can raise the room temperature excessively without necessarily exceeding the allowable ambient air temperature. If a large glass area cannot be shaded, one of the commercially available glass tinting films that block heat producing infrared rays is recommended.

If there is a question concerning the adequacy of your present temperature control system, a reputable air conditioning specialist should be consulted. The above information may be used to determine the suitability of your present system. Recommended temperature range for the SEQUEL environment is 65°F to 75°F (18°C to 24°C)

> NOTE: Air conditioners generate a large amount of electrical interference on the AC power line of the SEQUEL system unless proper steps are taken to isolate the power disturbances from the power lines serving the computer equipment. If an air conditioning unit is added, advise the air conditioning specialist of the unacceptability of line disturbances.

Humidity Control

High and low extremes of humidity can make paper documents and printer output paper difficult to handle. It can also impair proper operation of SEQUEL equipment, particularly the disc and tape storage units. Low humidity levels can have the most serious effect on system operation and are more frequently encountered than excessively high humidity. Low humidity promotes static electricity buildup in the electronic equipment. Proper equipment grounding minimizes this effect, but will not eliminate it completely.

The rapidly rotating magnetic disc and moving magnetic tape are particularly susceptible to static buildup which can destroy stored data. Static charges are further increased if the system is installed in a carpeted area. Carpeting and equipment ground is covered in the carpeting segment of this section.

High humidity extremes tend to prevent proper flight of the disc magnetic heads and smooth travel of the magnetic tape. Most

heating and air conditioning systems have a drying effect. Most air conditioners are rated for their ability to remove moisture as well as they ability to cool. In extremely humid environments with inadequate humidity controls it may be desirable to install a dehumidifying unit in the computer room. In very dry environments (such as desert areas) a humidifier may be necessary to add moisture to the air. Once again, the effectiveness of the humidit controls can be best evaluated by a qualified air conditioning specialist. The recommended relative humidity range is 40 percent to 60 percent (non condensing).

Dust Control

Airborne dust or dirt particles can cause equipment operatior or maintenance problems. If a film of dust or dirt accumulates on internal surfaces, excessive wear of mechanical parts may occur and electronic components may become shorted (particularly with high humidity). Disc and tape storage units are especially vulnerable to damage from excessive dust.

The disc heads ride on an air bearing that holds the heads about 100 microinches from the disc surface. Practically all dust particles exceed 100 microinches in size. Therefore, every possible effort must be made to maintain a dust-free environment. Heating, air conditioning and ventilating systems should be equipp with adequate air filters, and these filters should be cleaned or replaced at regular intervals to ensure proper temperature control as well as dust filtering.

Static Electricity

Static electricity is not only an annoyance to personnel, but can also cause equipment malfunction. It is important to minimize or eliminate the sources of static generating equipment.

Chairs, seats, or couches with plastic upholstery and rubber wheels should be prohibited in the computer system area. Furnitur of this type builds up and stores a static charge created by the friction of clothing moving around over the plastic. The rubber wheels prevent bleedoff through the floor covering. The discharge of this buildup can cause system malfunction when personnel or the furniture contact the system's framework.

Static problems can be minimized by:

Selecting chairs with antistatic upholstery and metal wheels to avoid static buildup.

Maintaining room above 40 percent relative humidity.

Using carpeting of a type designed to minimize static electricity.

Using chemical sprays to eliminate static on carpeting and furniture.

Smoking Hazards

The smoking of cigarettes, cigars or pipes should be prohibited within the vicinity of the central system. The discs and magnetic tape transports are particularly susceptible to smoke film buildup which can cause corruption of data and equipment damage.

DATA TERMINAL LOCATIONS

The SEQUEL data terminals that provide the user's communication link with the computer system can be located just about anywhere. The terminals are compact and may be mounted on a desk or table top in about the same space as an electric typewriter. Environmental requirements are the same as for other SEQUEL equipment, making the terminals suitable for use in labs, manufacturing areas, etc.

The maximum cable length for direct connection is 52.5 meters (175 feet) using the RS232C interface at 9600 bauds. Longer lengths may be possible with this interface but the integrity of data cannot be guaranteed; therefore, longer lengths are not recommended. The approved communication cable part numbers are provided in Table 2-2.

TABLE 2-3

APPROVED COMMUNICATION CABLE PART NUMBERS

Microdata Cable Assemblies	Lengths
A20032102-001	25 ft.
A20032102-002	50 ft
A20032102-003	75 ft
A20032102-004	100 ft (Kit, connectors not attached)
A20032102-005	200 ft (Kit, connectors not attached)

SECURITY

Internal design and programming of the SEQUEL system may enhance security of data stored within the system. Although SEQUEL data terminals may be located anywhere, accessible to all, data is safeguarded by special user passwords required to obtain information considered to be confidential.

Data access and/or alterations may be controlled through the use of retrieval/update lockout security codes in the file mangement software. These codes may be applied to entire files or to selected portions of the data records.

To protect confidential or sensitive information even further, it may be desirable to restrict physical access to the central system. (It is conceivable that an unauthorized person possessing the knowledge to operate the CPU controls could obtain any information in the system or tamper with the data files.) Maximum security can be achieved by locating the central system behind locked doors. Also, backup programs and data files should be duplicated and stored in a separate location under lock and key. The CPU panel operation can be inhibited by removing its key from the front panel.

ELECTRICAL REQUIREMENTS

Microdata SEQUEL systems sold for use in the United States operate from 208 VAC, 60 Hz single phase power. All elements of the SEQUEL central system (CPU, disc/tape storage units) connect together. SEQUEL data terminals and some peripheral equipment have separate 3-wire power tables which need not be plugged into the same power line as the central system. The central system contains a 30 ampere 208 VAC twist-and-lock 3-prong plug (NEMA L6-30P). This twist-and-lock plug will require installation of a compatible 30 ampere 208 VAC wall socket (NEMA L6-30R). Refer to Figure 2-1.

The data terminal power cables contain a standard 3-prong plug which is compatible with existing wall sockets; however, each wall socket must be an isolated ground receptacle. A conduit may not be used as a ground return.

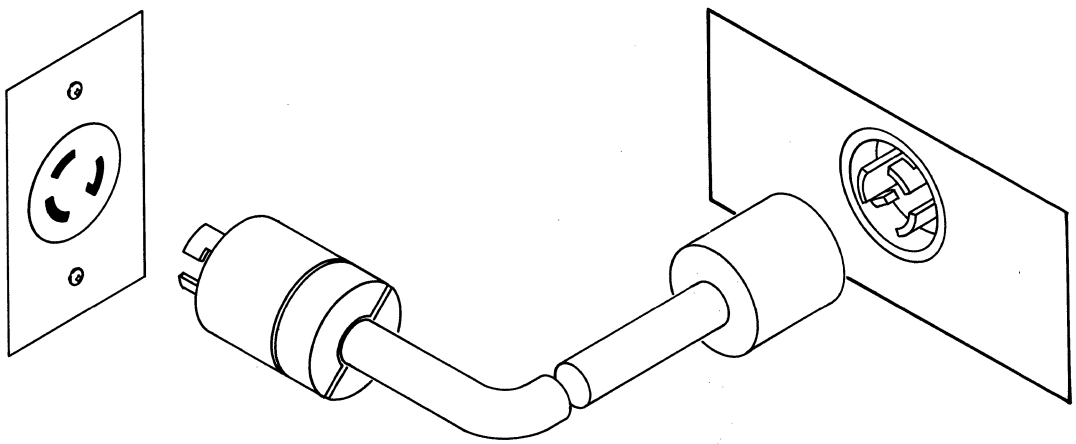


Figure 2-1. Power Plug and Receptacle

AC power at the outlet must be capable of supplying 30 amperes between 187 and 229 VAC, 59.5 and 60.5 Hz. Further, it must not fluctuate more than two percent from no-load to full load conditions and it must be free from line disturbances.

Electrical specifications for the SEQUEL system are shown in Table 2-4. A separate branch circuit for the SEQUEL CPU should be installed using one of the following methods:

(a) Metal-clad cable (bx), type AC article 334.
(b) Rigid metal conduit article 346.
(c) EMT thin-wall conduit article 348, or
(d) Flexible metal conduit, article 350.

NOTE: The circuit for this equipment must not be run through a conduit with other electrical circuits. It must be a separate and isolated conduit run which passes from the AC receptacles(s) to the circuit breaker.

Power supplied from any U.S. utility company should meet the foregoing requirements. If power at the outlet is deficient, it is likely that the wiring between the outlet and the power lines is inadequate. The problem may be an old or faulty circuit breaker causing an excessive voltage drop. In any case, the power company or a qualified electrician should be consulted to examine the wiring and recommend corrective measures.

TABLE 2-4

SEQUEL SYSTEM ELECTRICAL SPECIFICATIONS

Two-Bay System (Minimum)	208 VAC. Single-phase 60 Hz. Dedicated power line with a 30 ampere breaker required for each bay.
Main CPU Power Receptacle	NEMA L6-30R, Hubbell 2620 or equivalent.
NOTE: ONE RECEPTACLE IS REQUIRED FOR EACH BAY	
Printer, 300 LPM	NEMA 5-15R, Hubbell 5252 or equivalent, dedicated.
Printer, 600 LPM	NEMA 5-20R, Hubbell 5352 or equivalent, dedicated.
PRISM 4 Terminals	NEMA 5-15R, Hubbell 1G-5262 or equivalent (orange).
Power Consumption	Two-Bay (minimum) 3.12 KVA Four-Bay (maximum) 6.24 KVA

Power Failure

In case of power failure, the system will shut down automatically with no damage to system hardware. When power is restored, the system will automatically restart and resume operation. The disc units may require up to three minutes delay after power restoration before they are operational. The printer and magnetic tape units may require manual attention to reload the tape or reposition the paper.

Electrical Noise

Another factor that may require the attention of an electrician is electrical noise and impulses on the AC power lines. Electrical equipment such as air conditioners, copy machines, and typewriters can generate noise transients that may be fed back on the AC power lines. For this reason it is recommended that a separate power line be used for the SEQUEL central power system.

Electrical noise can sometimes be eliminated by repair, replacement, relocation or electrical filtering of the originating device. If the electrical noise is not eliminated as suggested, a suitable power line conditioner may have to be installed in the AC line to the SEQUEL system by a qualified electrician. Microdata can assist in the selection of the line conditioner.

Circuit Breakers

The central system equipment should be on a separate line, protected by a circuit breaker rated at 30 amperes. This breaker will tolerate the short-time startup current surges without tripping.

Power Cable

The SEQUEL equipment should be situated close enough to the power outlet so that the 4.5 meter (15 feet) power cable supplied with the system will suffice. Extension cords are not allowed by the National Electrical Code (NEC). The NEC also forbids flexible cords longer than 4.5 meters on equipment.

Grounding

A good electrical ground is extremely important for reliable equipment operation. Some installations may require direct connection to a grounding stake or other high quality earth ground.

PREPARATION AND SITE LAYOUT

Layout of the SEQUEL system site demands careful planning of the equipment and furniture that are to be moved into place and connected for operation. Once the computer equipment is installed

and interconnected, repositioning is likely to be more involved than simply moving a desk or typewriter.

Locations of equipment and furniture should be based on the operational requirements of the SEQUEL system and the efficiency comfort, and convenience of personnel using the room. Individual pieces of equipment are interconnected by cables of limited length, and it is necessary to maintain certain clearances for access, servicing, and work space. Therefore, you may wish to consider several tentative layouts before selecting the one best suited to your needs.

A site layout worksheet and a page of line drawings representing the individual SEQUEL system elements is Appendix X. Each element is complete with minimum clearances and work space requirements. A scale of 1/4th inch = 1 foot is used for both the worksheet and equipment drawings.

To plan the layout, photocopy both the worksheet and the page of equipment drawings. Measure the computer site and draw it to scale on the worksheet. Cut out the photocopied equipment drawings and position them on the worksheet, pasting them in place when a desired layout is found. Desks, filing cabinets and other furniture can be drawn on the worksheet using the 1/4th inch = 1 foot scale. Repeat the procedure for layout variations and alternate installation sites.

Separate cutout drawings are provided for the SEQUEL data terminals and peripheral equipment. Work areas and cabling/airflow/maintenance clearances (as applicable) are identified on the equipment drawings. Work areas shown are considered to be the minimum required for comfortable, efficient operation of the system. Space permitting, these work areas may be larger than indicated, but under no circumstances should they be smaller or overlap one another.

The CPU cabinet is mounted on rollers and can easily be moved to permit access for maintenance. CPU cabinet access is via the front panel doors, both side doors, and the rear door. At least 24 inches is required behind the CPU cabinet for access and removal of circuit boards, proper air flow and communication cable exits.

Lighting Requirements

Another important consideration during site layout is the location and angular position of the SEQUEL data terminals with respect to light sources. Office desks and all other SEQUEL equipment should receive sufficiently high light levels to permit reading, writing, and operation and servicing of the system. Optimum use of the data terminals may require a somewhat lower light level for easy reading of the CRT display. If the CRT

screen or the operator faces a high intensity light source or unshaded window, the screen will be difficult to read. All terminals should be positioned at right angles to windows and any major light source. Ideally, the area should be somewhat darkened with lighting controls that can be adjusted by the operator.

Cabling

All SEQUEL system elements are supplied with interface cables which connect them to the CPU. The length of the interface cable limits the distance between each piece of equipment and the CPU. For purposes of site layout, standard and maximum cable lengths for all SEQUEL peripheral equipment are summarized below:

Data Terminal: 25, 50, 100, 200

Printer: 20

Power Cable: 8

Power cable for the printer must be a dedicated line of 120V AC, 60 Hz standard ((NEMA 5-15R), 3-prong type.

Cables used in the SEQUEL system are small in size and few in number. This eliminates the need for raised false floors required in many systems. Plastic cable troughs are one way to increase the safety and neatness of an installation. These cable troughs may be purchased inexpensively from any electrical supply store in 6-foot lengths that can be cut to size. Troughs are also available in various widths and heights. An alternative method is overhead cable routing, generally employed to eliminate long cable trough runs when the computer is to be installed in the center of a large floor area. Cables are suspended across or above the ceiling, dropping down to the SEQUEL equipment.

Storage Requirements

Storage requirements vary depending on the SEQUEL system configuration and the individual application. As a minimum, storage space should be provided for the following items:

Printed paper forms (bulk)

Printed reports produced by the system

Spare/off-life magnetic tape reels (if used)

Operator manual

Programmer manuals

Expendable items (printer ribbons, etc.)

Equipment cleaning supplies

Operator and programmer manuals should be kept in a locked, fire-resistant cabinet or container which is immediately available to operation personnel and service technicians. Combustible materials such as cleaning solvents should be stored in fire-resistant containers in accordance with national fire protection association standards. To minimize storage space requirements at the SEQUEL site, bulk items may be stored in a remote location. The environment of the remote area should be controlled within reasonable limits (less than 150° F [68° C] and less than 90 percent relative humidity).

Telephone, Datalines and Modems

It is advisable to have all telephones and telephone data lines installed prior to installation of the SEQUEL system. If a system uses phone lines to transmit data to or from remote data terminals, modems need to be installed. The telephone company regulations require a direct access arrangement (DAA) to be installed by the phone company on each telephone line that connects to a modem. Microdata's representative should be consulted regarding type and speed of modems for SEQUEL.

Carpeting

If your computer site is to be carpeted, it is advisable to select a carpet with antistatic characteristics woven into it. Static charge buildup on nylon carpeting can produce noise transients in the system interface lines which may cause the system to degrade in performance. Wool and nylon carpeting have the worst static characteristics of all types, and should be avoided under all circumstances.

Environmental Control Equipment

Ideally, air conditioning and other environmental control equipment should be located outside the computer site to minimize the acoustical noise level and to reduce the possibility of electrical interference. However, regardless of the physical location of the equipment, it must not be connected to the power lines serving the SEQUEL system. If this is abolutely unavoidable, then proper line filtering measures must be taken. If an air conditioner, humidifier, or dehumidifier are to be installed in the computer site, be certain to allow adequate space for proper operation and servicing of the unit (e.g., filter replacement, etc.).

Fire Prevention

Fire prevention measures should be reviewed and implemented before installation of the SEQUEL system to protect the capital investment and the safety of the employees. Be certain to satisfy the requirements of your insurance carrier and local fire department.

Insurance or local regulations may require an overhead sprinkler system in the SEQUEL site. If so, any costs involved should be determined as early as possible and considered as a factor in site selection. Whether or not a sprinkler system is installed, most regulations require a number of portable CO_2 fire extinguishers to be located in or adjacent to the computer site. Your local fire department can advise you of the number and recommended type of extinguishers required and the best locations for their placement.

Attention should also be given to protecting paper forms, reports, and magnetic tapes from possible destruction in event of fire. A fire resistant safe or file cabinet may be used for this purpose. It is recommended that a duplicate backup copy of important data and files always be kept in a separate, fire resistant facility.

SECTION 3

INSTALLATION AND OPERATION

INTRODUCTION

The purpose of this section is to aid Customer Service personnel during on-site installation of the Microdata SEQUEL System. Before starting the installation make certain that all parts of the system have arrived and that proper tools and diagnostic tapes are available.

UNPACKING

Ensure that the system is near the final location before any external packing material is removed. Check with the customer for the final location. Be certain that the system is also located near its power source. Figure 3-1 shows a typical SEQUEL System in its protective shipment wrapping.

Figure 3-1. SEQUEL Cabinet in Protective Wrapping

The order in which the system is unpacked is of no significance and will vary with system configuration. It is a good idea to begin unpacking with a printer or some other unit that has a large crate. Once this item is unpacked, the crate can be used as a container for the packing of the other units.

> CAUTION:
> Do not cut shipping cartons with a knife or any type of sharp object while unpacking to avoid the possibility of damaging the contents.

Visual Inspection

(a) Examine the shipping crates and cartons for any apparent damage. If damage is observed, note the nature of the damage and follow local procedure for handling damaged shipments.

(b) After unpacking, visually inspect the condition of the system to determine if any damage occured while in transit. Check for scratches or scrapes of painted surfaces and dents in panels. Look for broken shipping straps or damage to packing material.

(c) Check the cabinet leveling legs. Make certain that they are not bent. Remove all doors from the cabinets.

> WARNING
> The cabinet(s) may tip over if the power supply and REFLEX disc drives and/or magnetic tape drives are extended without fully utilizing the SEQUEL cabinet stabilizers (see Figure 3-2).

(d) Roll the first cabinet into its desired location. Carefully peel the protective paper from the magnetic tape unit (MTU) window.

(e) Remove the hold down clamps (shipping brackets) on the REFLEX II disc drive(s) located in the main cabinet. Remove the shipping brackets from the disc drive (s) in the expansion cabinet before rolling the second cabinet into position. Save all hardware at the site.

(f) The cosmetic panel should be removed from the front of the cabinets. Remove and retain the screws holding the radio frequency interference (RFI) screens in place and remove the screens.

(g) There are holdback screws that prevent movement of the power supply. Remove and retain these screws and then check the power supply drawer for free movement. Check

the disc drives to see that they also roll freely on their designated tracks.

> **CAUTION**
>
> Carefully monitor the cables when a power supply, disc drive, or magnetic tape unit is moved or swung back into the cabinet.

Figure 3-2. SEQUEL Cabinets with Extended Stabilizers

(h) Remove and retain the shipping bracket (bar) from the MTU. Cut the plastic tie and remove it from the MTU cables.

(i) Remove the MTU door from the cabinet. (The MTU must swing out fully to allow working space for bolting the cabinets together.)

(j) Check all PCBs to see that they are properly seated in the backplane. Check the switches and seating of all cables (cables on disc controller can go both ways).

(k) Remove the power supply cover and make a careful safety conscious inspection. Refer to Figure 3-3. Check cable plug seating, the wiring connections (including +5 VDC cables. Make certain all capacitor screws are tight. Look for loose nuts and screws and tighten them accordingly.

CAUTION:
Do not short any capacitors under any circumstances with any tools of any kind. Severe injury to personnel could result.

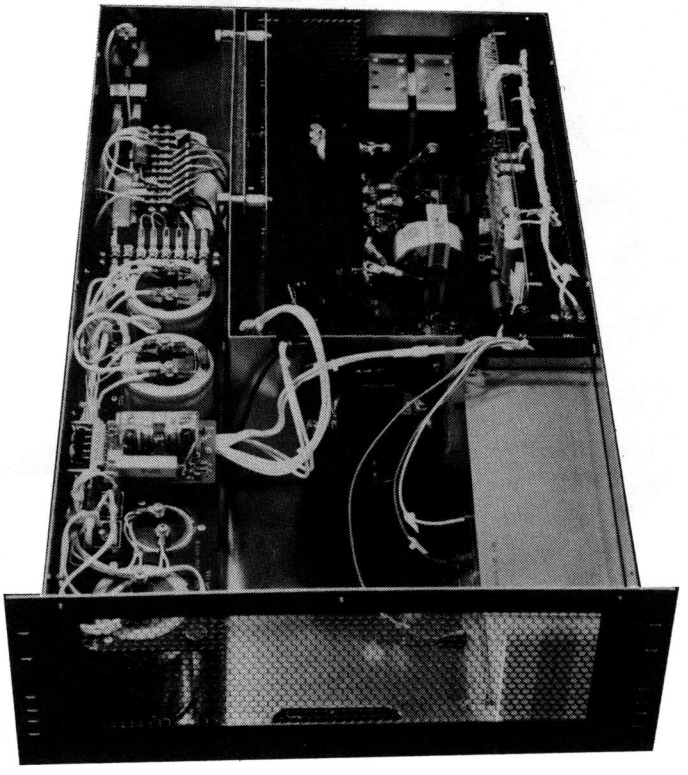


Figure 3-3. Power Supply with Cover Removed

(l) Verify and install memory battery leads to batteries for backup condition.

NOTE:
The bolts are in the separate box for the cables. The box also contains the power cable, external battery cable (power supply to AC distribution), printer cable, DMP to ACLC port 0 cable and sys-gen tape.

SEQUEL CONTROL PANEL

The SEQUEL control panel has a set of switches that are used to select the operational mode of the DMP and the main CPU. This control panel is shown in Figure 3-5. The functions of the switches and the lights are described in the following paragraphs.

Switches

REMT — This is a momentary switch that generates an interrupt to the DMP, allowing the firmware alternately to select the local or the remote terminal.

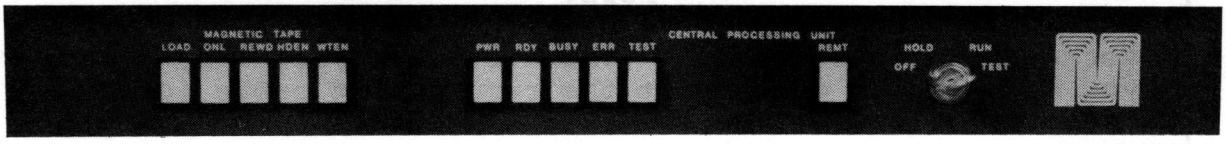


Figure 3-4. SEQUEL Control Panel

Four-Position Rotary Switch

OFF — All system power is off.

HOLD — Power is applied to the memory only. Controller boards can be swapped without destroying memory contents.

RUN — Power is applied to the entire system.

TEST — The DMP can respond to the CTRL-D for diagnostic mode and to the CTRL-F for the firmware debugger.

Lights

PWR — Power is available to the system.

RDY — The hardware system is ready for operation.

BUSY — The SEQUEL bus is not active.

ERR — An error condition has been encountered or the DMP self-test is running.

DIAGNOSTIC MAINTENANCE PROCESSOR

All SEQUEL Systems contain a DMP board. Through the DMP the user can perform various diagnostic tasks, such as using the firmware debugger and executing diagnostic programs. The DMP is also used when booting the system, and for accessing the system when the "normal" system software becomes inoperable. By using a modem, the DMP provides a remote diagnostic capability, allowing the analyst to solve system problems without being on site.

Since the DMP is a new device, it is recommended that the SEQUEL Programmer's Reference Manual be read for details on its operation before the DMP is used.

The DMP can be used from either port 0 or the remote port. If the remote port is to be used, the REMT button on the front panel of the SEQUEL must be pressed. Whatever appears on the terminal from the remote port will also appear on port 0's terminal and vice versa. This is handy when trying to show a user what to do or when the user wishes to show the CE what has been done.

> NOTE:
> The ability to see what is being done on port 0 (and vice versa) is always enabled. Therefore, for security reasons you may wish to disconnect the remote port cable (or disconnect the modem) when it is not needed.

The DMP can be activated either by powering up the system, or by switching the key switch to TEST, and typing CTRL-D on either port 0 or the remote terminal. The DMP will first execute several diagnostic programs. These will take time to execute. In particular, the ACLC SELF TEST may take as long as two minutes or more to run. When the tests are complete, the DMP will display the SYSTEM CONTROL MENU.

If the system was powered down, either via the key switch or because of a power failure, the system will do the following; first, it will power up the CPU and begin running the diagnostic tests. During the course of the tests it will activate the disc units. If the tests complete without an error (remember that the ACLC test takes several minutes), and main memory is still intact (battery backup can retain main memory for at least 20 minutes, depending on the size of main memory and condition of the batteries), and the key switch is in the RUN position, the system will continue processing where it left off. If main memory has been lost, or if the key switch was in the TEST position, the system will display the message:

POWER FAIL, SYSTEM RESTART MAY BE REQUIRED

Pressing <cr> will cause the command menu to be displayed.

DMP Command Menu

The DMP command menu allows the performance of 10 functions:

Reset the system.
Bootload the system.
Enter the firmware debugger.
Stop the CPU.

Restart the CPU.
Toggle the local/remote terminal.
Change the baud rate for port 0.
Execute the CPU firmware diagnostics.
Execute the DMP firmware diagnostics.
Display/alter the contents of main memory.

With the possible exception of option 2, these functions are different from any functions on the REALITY system. It is necessary that the SEQUEL Programmer's Reference Manual be read before using any of these functions, since the consequences of each function may cause the system to lose data. Below is a brief summary of each option.

Option 1: Reset system. This option resets the central CPU and all the controllers. This means that all current I/O operations are destroyed, requiring the performance of a coldstart.

NOTE:
This option is not comparable to an IRI sequence on a REALITY System.

Option 2: Bootload. This option will cause the automatic diagnostics to be executed, and will cause two records to be read from the tape. The system will then display the OPTIONS message, similar to that displayed for REALITY Systems.

Option 3: Enter the firmware debugger. This option tells the CPU to activate the firmware debugger. The firmware debugger is used to debug the monitor, and to debug the system debugger. It may also be entered by typing a CTRL-F from port 0's terminal when the key switch is in the TEST position. To leave the firmware debugger, type an "X" at the debugger prompt.

Option 4: Stop the CPU. This option halts the CPU and disables all interrupts except the DMP handshake interrupt to the CPU.

Option 5: Resume CPU processing. This option tells the CPU to continue processing. Any interrupt which would have been executed prior to Option 4 will now be allowed to take place.

Option 6: Switch control to local/remote terminal. This option selects either the local or remote mode (i.e., which terminal will control the DMP).

Option 7: Change baud rate. This option allows changes in the baud rate for both the local (port 0) and remote terminal.

NOTE:
After the baud rate change is executed, the characters on the terminal will blocks and be garbled. Setting the terminal itself to the correct baud rate will cause the normal characters to be displayed.

Option 8: Run CPU firmware diagnostics. This option executes a diagnostic program which has four options. See the SEQUEL Programmer's Reference Manual for details. After this option is used, a warmstart will be required. In some cases a coldstart may be required.

Option 9: Run DMP firmware diagnostics. This option allows a variety of test programs to be executed. Refer to the Sequel Programmer's Reference Manual for details.

NOTE:
The resident memory test is a destructive test that will require a coldstart when completed.

Option 10: Display/alter memory. This option allows you to display the contents of main memory, and to alter the contents, if desired. Three consecutive <cr>'s will return to the main menu.

Options 1, 4 and 5 are not comparable to an interrupt, reset, interrupt sequence on a REALITY System and should not be used as such. In general, none of the options (with the possible exception of Option 2) should be used without first consulting a Microdata Software Support person.

CABLES

Because of the increased number of ports available on the SEQUEL, a smaller plug and socket for the serial port interface cables has been selected. This connector is a nine-pin conector made of three components: the pin connector, a cable clamp, and a retainer. Terminals ordered specifically for the SEQUEL (part number 7-5420) will have the proper connectors supplied. If a user wishes to utilize existing terminals, the connector may be purchased from a wholesale distributor. The following is a short (i.e., not inclusive) list of suppliers along with the part numbers.

9-Pin Connector	Cable Clamp	Retainer
Cannon DE9P	Cannon DE110963-1	Cannon D20419
TRW Cinch DE9P	AMP 207467-1	TRW Cinch D-20419
Souriau DE9P		Souriau 8630-04
Positronics MDM2000		

The Sequel requires four twisted pairs within the cable (REALITY cables are 2-wire twisted pair). This means that a user cannot simply utilize the terminal cables supplied with a REALITY Series System on the SEQUEL System. The following cable lengths may be ordered from Microdata:

Cable Assemblies	Lengths
A20032102-001	24 feet
A20032102-002	50 feet
A20032102-003	75 feet
A20032102-004	100 feet (connectors not attached)
A20032102-005	200 feet (connectors not attached)

The standard part number for four twisted pair double shielded cable is CS-20032098. In addition to Microdata, the following vendors can supply such a cable in bulk length (this list is not all inclusive) as specification UL2464.

C & M
Brand Rex
Storm
Blake Wire & Cable

Maximum cable lengths for interfacing the terminal with the SEQUEL Systems are as follows:

Baud Rate	Length
1200	1392 ft
2400	636 ft
4800	348 ft
9600	175 ft
19200	87 ft

Data reliability is not quaranteed if longer cables are used. Any such longer cables will at the user's discretion. Line amplifiers (short haul modems) are recommended if greater distances are required.

The terminal interface cable should be wired as follows:

25-Pin Connector	9-Pin Connector
Pin 1	Pin 1
Pin 3	Pin 2
Pin 7	Pin 7
Pin 2	Pin 3
Pin 20	Pin 6

In addition, pins 5, 8, and 9 in the 25-pin connector must be strapped. Pins 5, 8, and 9 in the 9-pin connector are also strapped.

BACKPLANE ZONING AND SLOT ASSIGNMENTS

The backplane has 21 slots and 3 zones where the boards are located. See Figure 3-6. Slot numbers should be counted from left to right as viewed from the back. The zones and slots are as follows:

I/O Zone. (Slots 1 through 8) This area will accept all controllers and the I/O processor.

CPU Zone. (Slots 9 through 12) This area is marked with red plastic rails. The DMP, ALU, RNI and special function boards will be accepted here.

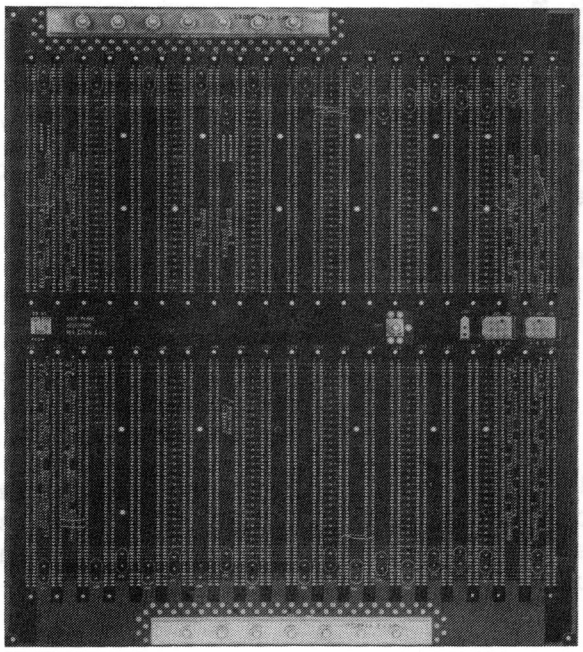


Figure 3-5. SEQUEL Backplane

Memory Zone. (Slots 13 through 21) This zone will accept memory control and memory array boards. Slots 13 and 14 are for the MCB board. Slots 14 through 21 are for the memory array boards.

> NOTE:
> Slot 14 may be used as an alternate slot for the memory control board. Slot 13 may also be used as an alternate slot for the CPU boards, in which case MCB moves to slot 14. This feature may be helpful while troubleshooting a backplane when it is suspected that the connector on the backplane in the CPU zone or MCB slot is defective.

A typical minimum configuration system may be like the following:

Slot Number	Description	Assembly No.
1	MTU/PRTR CTRLR	A20032012
2	ACLC	A20032014
3	ACLC	A20032014
4	ACLC	A20032014
5	ACLC	A20032014
6	IOP	A20032013
7	DISC CTRLR	A20032011
8	(Open slot)	
9	DMP	A20032015
10	ALU	A20032007
11	SPEC. FUNCT.	A20021009
12	RNI	A20032008
13	MEM CTRL BOARD	A20032001
14	MEMORY ARRAY	A20032064
15	MEMORY ARRAY	A20032064
16	" "	"
17	" "	"
18	" "	"
19	" "	"
20	" "	"
21	" "	"

> NOTE:
> If an expansion chassis is used, slot 1 must be reserved for IOP board so that signals from it can be transferred via cable to the expansion backplane.

Installation and Checkout

(a) Install the external battery cable and the DMP-ACLC cable (see Figure 3-6).

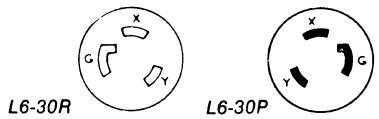
(b) Replace all the doors on the cabinets with the RFI screens in front.

(c) Allow sufficient clearance to any wall so that the back door of the cabinet may swing open without touching the wall. Screw down the jacks so that the cabinets will not roll. Make certain that the tops of the cabinets are the same height.

(d) Connect all cables as shown in the reference drawings.

CD20032100	System Cable Drawing	Single Disc Cabinet
CD20032185	...Expansion	" " "
CD20032220	System Cable Drawing	Dual Disc Cabinet
CD20032222	...Expansion	" " "

(e) Verify that each wall socket is a NEMA-L6-30R and that there is one for each cabinet. Check for proper AC voltage as shown below:



X - G = 120 VAC
Y - G = 120 VAC
X - Y = 208 VAC

NOTE:
The blower assembly has a microswitch that senses proper air flow into the system. In case of restricted air flow or blower fan malfunction, the power supply will not turn on. Care must be taken not to swing the rear doors in or out with a rapid motion since this action may cause the microswitch to malfunction.

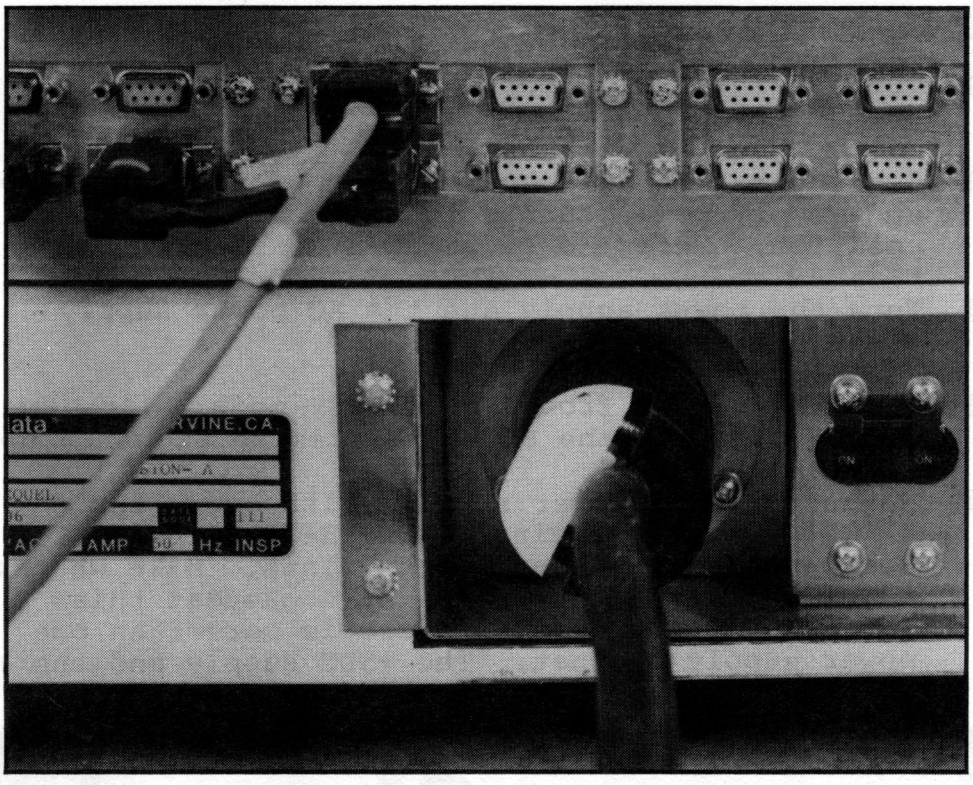


Figure 3-6. SEQUEL I/O Panel

(f) If the user has a modem on the SEQUEL System, the parity, baud rate and PCI setting of the local port 0 must be modified by hardware switches on the DMP board. These settings must match the remote port 1 baud rate.

In most cases the asynchronous modems are operating at 1200 baud. Table 3-1, DMP Baud Rate Select, should be consulted for the proper switch settings. Remove and readjust the DMP board accordingly.

(g) Connect the terminal cable from DMP port 0 to the local PRISM 4. Adjust the local terminal so that it has the same characteristics of parity, 1200 baud, enable parity, odd parity, 7-bit character length, and full duplex mode communication. Figures 3-21, 3-22, and 3-23, PRISM 4 interim communications switch settings, should be consulted.

Whenever the system is powered ON, the local port 0 will default to the 1200 baud rate and be compatible with the remote port for diagnostic purposes.

Powering System

(a) Connect the power supply cord from the wall socket to the AC module.

(b) Set the margin switches on CPU power supply supply and memory power supply to the nomimal (center) position. Refer to Figure 3-8.

(c) Turn the power supply and the AC power supply circuit breakers on.

(d) Turn key switch from OFF to TEST mode, allowing all voltages to the SEQUEL System to sequence up.

(e) Measure the DC power supply voltages on the backplane by following Figure 3-20, SEQUEL Backplane Voltages and Test Points. Make whatever adjustments that are necessary. Repeat this adjustment procedure if there is more than one power supply cabinet. The +5DC supply and the +5V memory DC supply have adjustments.

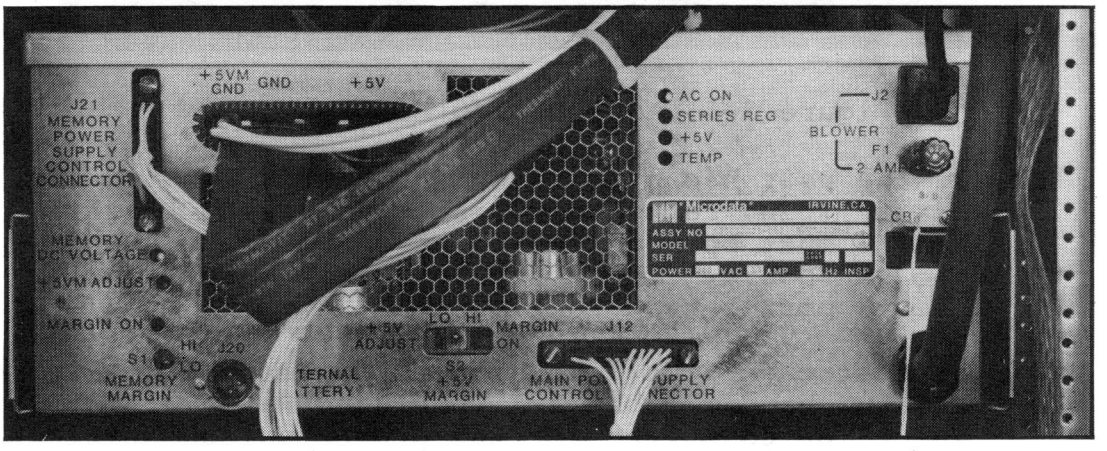


Figure 3-7. SEQUEL Power Supply, Rear View

BOOTLOADING

Bootloading is the procedure for loading all or part of the software to bring the system to a runable state. The bootload procedure loads the software from magnetic tape so a file-save tape should be mounted on tape unit zero before the bootload option is selected. When the bootload option is selected, the DMP runs the self-tests and sends a signal to the CPU that bootload has been requested.

The CPU rewinds the tape and loads the first two records into memory locations 0-X'FF' and X'600-X'7FF'. If the tape is not ready, the CPU waits for the tape to come ready. If either

record is in error, the CPU displays the message

TAPE ERROR, RECORD n

where n is 1 or 2. Main memory program execution is forced to begin at location X'20' in the MPCB, which transfers to the MBOOT program, which displays the options message:

OPTIONS (W, X, A, F, AF, B, D) =

The operator must reply with one of the options presented in the parentheses. The 'D' option is used by customer engineers to format disc platters and verify check-codes. Since it is not a bootload procedure, this discussion does not apply to it.

'W' is the warmstart option, which loads the configurator and parts of the monitor only. All the other options are considered coldstart options.and are described in the SEQUEL reference manuals. They reload some or all of the changeable parts of the system. If erroneous input is entered, the message is repeated. If there are any tape errors from this point on, the system displays the message

TAPE ERROR

on the terminal and "hangs up," which means that the bootload procedure must be started from the beginning.

If there have been no tape errors, the configurator modes are loaded from tape. These modes configure the system by asking the operator a series of questions.

The spooler is assigned to the first line beyond all existing ACLCs, or line 127 if the system has the maximum number of terminals. The system displays the message

n IS THE SPOOLER'S LINE

If something other than "N" or "Y" is typed, the message is repeated.

The system next determines how many lines can be used by the terminal independent process handler (TIPH). The TIPH can have terminal-independent processes for all lines numerically greater than the spooler's up to 127. Hence, if the spooler's line is 127, there are no TIPH lines. If the spooler's line is not 127, the system displays the message

ENTER NUMBER OF TIPH LINES

The maximum number of lines available for use as TIPH lines is 127 minus the spooler's line number. If the number entered in reply to the message is greater than the maximum available, the number of TIPH lines is set to the maximum; otherwise, it is set to the number typed in. A null response is treated as a zero, but otherwise if the response is non-numeric, the message is displayed again.

The system senses how many K-byte memory increments are present and prints the following message on the terminal:

nK OF MEMORY

If there are non-defective disc drives, the number of heads on each usable drive and its device address are displayed in the following message:

DRIVE Dx, or DRIVE Dx, REFLEX III

x is the device address and yy is the number of heads on that disc. For each drive x that the DMP found defective, the system prints

Dx DRIVE DEFECTIVE

The system then asks the question

CONFIGURATION CORRECT? (Y/N)

If "N" is entered, the options message is again displayed, and the system must be reconfigured. If "Y" is typed, the system calculates the addresses of its memory management tables and the beginning FID of available disc space.

If the bootload is a coldstart, the memory management tables are initialized and certain frames are loaded from tape. During a warmstart these data are skipped.

The monitor is reloaded for both warmstart and coldstart. For a warmstart the monitor's test interrupt routine is entered, which has the effect of continuing all processes where they halted. For a coldstart all terminals are set to LOGON.

POWER FAIL/POWER RESTART

When the power fails, the DMP sends a power fail signal to the CPU, which goes into a loop waiting for a power restart. When power is restored, the DMP runs the self-tests. If the rotary switch is set to TEST at power restart, the DMP will display the control menu after the self-tests. If the rotary switch is set to RUN, the DMP's action will depend on whether the memory's power also failed or not. If power to the memory also

failed, the DMP commands the CPU to reset memory parity, and the DMP displays the message:

POWER FAIL--COLD START REQUIRED

If the memory power did not fail, the DMP signals the CPU that power has been restored and the CPU resumes processing from where it left off.

TERMINAL ZERO

In diagnostic mode the DMP controls terminal zero directly, not through the ACLC. The DMP continues to control the terminal while the firmware debugger is running and also during the bootload procedure up until just before the options message is displayed. The options message and all subsequent terminal activity during bootload are handled by the CPU through the ACLC.

If the ACLC was outputting to terminal zero when the DMP takes control of the terminal, the ACLC continues "outputting" even though the terminal is not connected.

SEQUEL SWITCH SETTINGS

> NOTE:
> Switch setting illustrations in the following pages are drawn to represent the switch as it is viewed from the side of the SEQUEL System, with the board installed.

TABLE 3-1

DMP BAUD RATE SELECT

S1 (4 3 2 1)	Baud Rate
0 0 0 0	50
C 0 0 0	75
0 C 0 0	110
C C 0 0	134.5
0 0 C 0	150
C 0 C 0	300
0 C C 0	600
C C C 0	1200
0 0 0 C	1800
C 0 0 C	2000
0 C 0 C	2400
C C 0 C	3600
0 0 C C	4800
C 0 C C	7200
0 C C C	9600
C C C C	19200

NOTE:
C = Closed = ON = Logical 1
0 = Open = OFF = Logical 0

TABLE 3-2

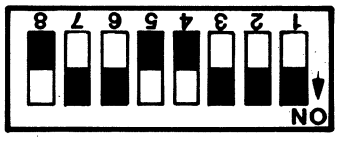
DMP PCI MODE SELECT

S1 (5-7) 7 6 5	Binary Number	Data Bits	Parity Bit	Total Bits
0 0 0	0	7	None	7
0 0 C	1	7	None	8 (8th bit = 1)
0 C 0	2	7	None	8 (8th bit = 0)
0 C C	3	7	Odd	8
C 0 0	4	7	Even	8
C 0 C	5	8	None	8
C C 0	6	8	Odd	9
C C C	7	8	Even	9

S1

OFF = Open

ON = Closed



PCI MODE

BAUD RATE

Note 1: The TERM verb has been expanded to allow the user to set the baud rate and PCI (programmed communications interface) setting for a given terminal. When the baud rate and/or PCI setting is changed, the port number of the affected terminal must be specified (you cannot set the baud rate or PCI from the terminal being changed). The other TERM parameters (same as REALITY) can only be changed on the terminal being affected.

Note 2: The above diagram of S1 depicts the baud rate set to 9600 and the PCI mode set to 7-bit, odd parity for a total of 8 bits. The #8 throw is a "don't care" position; however, for uniformity, leave it in the "open" position.

Note 3: For the DMP board's S1, Open = OFF = logical 0 and Closed = ON = logical 1.

Note 4: The baud rate and PCI settings must correspond with the mode of operation for the CRT terminal connected to the DMP "Local/Remote" port.

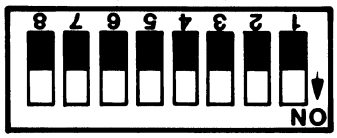
Figure 3-8. DMP Switch Setting

S1

(Location 4K)

OFF = Open

ON = Closed



(This switch is currently not in use)

Figure 3-9. ALU Switch Setting

When S1 is positioned to SB, the MCB is enabled to detect and correct single-bit errors that occur within a single byte of memory data. When S1 is positioned to DB, the MCB will detect double-bit errors.

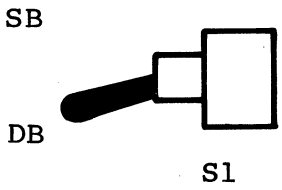
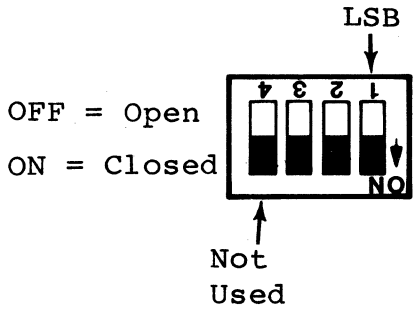


Figure 3-10. MCB Switch Setting



Note 1:	S1 (1-3) is used to select the memory array board number. Board numbers range from 0 through 7, and they must be numbered in logical sequence with no gaps within the range. It should be noted that physical location of the individual MABs is not essential for correct addressing. A typical sequence might be as follows: 2,3,0,1,..."highest" MAB.

S1 3 2 1	Binary	
C C C	0	First Memory Array Address (256K)
C C O	1	Second Memory Array Address
C O C	2	Third Memory Array Address
C O O	3	Fourth Memory Array Address
O C C	4	Fifth Memory Array Address
O C O	5	Sixth Memory Array Address
O O C	6	Seventh Memory Array Address
O O O	7	Eighth Memory Array Address

Figure 3-11. MAB Switch Setting

IOP Switch Configuration Instructions

The IOP (input/output processor), also referred to as the I/O controller, is used to interface the disc controllers and the ACLCs with the SEQUEL CPU and main memory. The first of the following two paragraphs detail the method for selecting the correct device address and mode of operation for the IOP. The second paragraph is a brief discussion of how to select and assign the appropriate bus access and interrupt priority based on the SEQUEL system's priority level structure. For a more thorough discussion on this topic, refer to the SEQUEL Interface Specification (IS20032010) and the IOP Product Specification (PS20032013) documents.

Device Addressing Mode Selection

The IOP may be configured in one of three modes as follows:

(a) As the processor which services the disc controller(s) exclusively in the "DC ONLY" mode;

(b) As the processor which services both disc controller(s) and ACLCs in the "mixed" mode:

(c) Or, as the processor which services the ACLCs exclusively in the "ACLC" mode.

In the first mode, switch S1 is placed in the "DC ONLY" position as marked on the printed circuit board next to the toggle switch. This enables the IOP's firmware to recognize "FC" as the device address for that IOP. (Note: Do not confuse the term "device address" with the term "controller address.")

In the second mode, switch S1 is positioned away from the "DC ONLY" position, thus enabling the IOP's firmware to recognize "FB" as its device address. In this mode the IOP recognizes controller addresses 0, 1, 2, and 3 for the disc controllers, and controller address 4 through 7 for the ACLCs. The range of controller addresses that the IOP will recognize in this mode is from 0 through 7, the maximum allowable range for a system without an expansion chassis.

In the third mode of operation, switch S1 is positioned away from the "DC ONLY" position, thus enabling the IOP's firmware to recognize "FB" as its device address. Additionally, a flat ribbon cable (A20032168-001) is connected between the IOP's J3 connector located on the IOP's foreplane, and connector J3 on expansion backplane. This connection selects the mode in which the IOP will recognize the range of controller addresses from 0 through F as belonging to the ACLCs residing in the expansion chassis. This is the maximum available range of controller addresses for the ACLCs in a system equipped with the optional expansion chassis.

Bus Access and Interrupt Priority Selection

Presently, the IOP may be assigned to either system priority level, depending on which set of controllers it will be servicing. The IOP that is designated for servicing the disc controller(s) in the "DC ONLY" mode shall be assigned to system priority level 2. The IOP that is designated to service the ACLCs in the "ACLC" or missed mode shall be assigned to system priority level 1. These assignments are made on the basis of the controller's memory access times and data transfer rates for each group of controllers. (See Tables 3-4 through 3-8 at the back of this section.) Bus access is controlled by the bus request (BSRQ) and bus address (BSAD) switch settings in the dip switch, S3, located

at 16F on the IOP board. Interrupt priority is controlled by the interrupt request (INRQ) and interrupt address (INAD) switch settings in the dip switch, S2, located at 17F on the IOP board. Refer to Table 3-3 for the proper switch settings.

TABLE 3-3

IOP CONFIGURATIONS

"DC ONLY" mode	Refer to Figure 3-12.
"Mixed" mode (DC + ACLC) .	Refer to Figure 3-13.
"ACLC ONLY" mode	Refer to Figure 3-14.
Note: For the IOP's switches, closed = ON = Logical 0, and Open = OFF = Logical 1.	

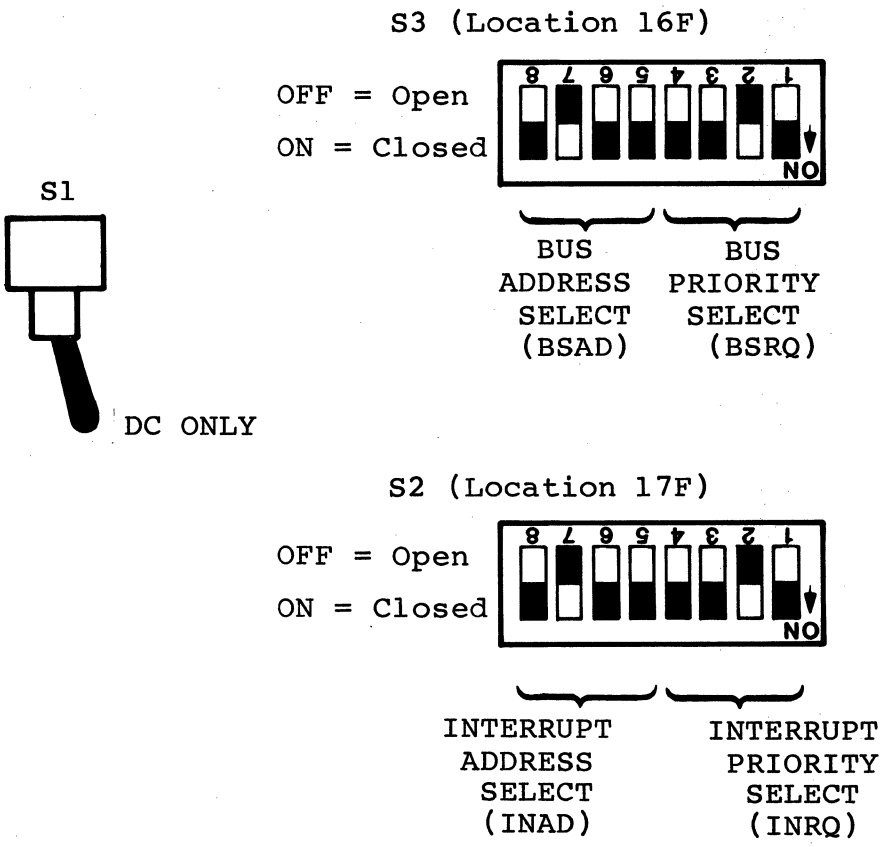


Figure 3-12. IOP Switch Settings, DC ONLY Mode

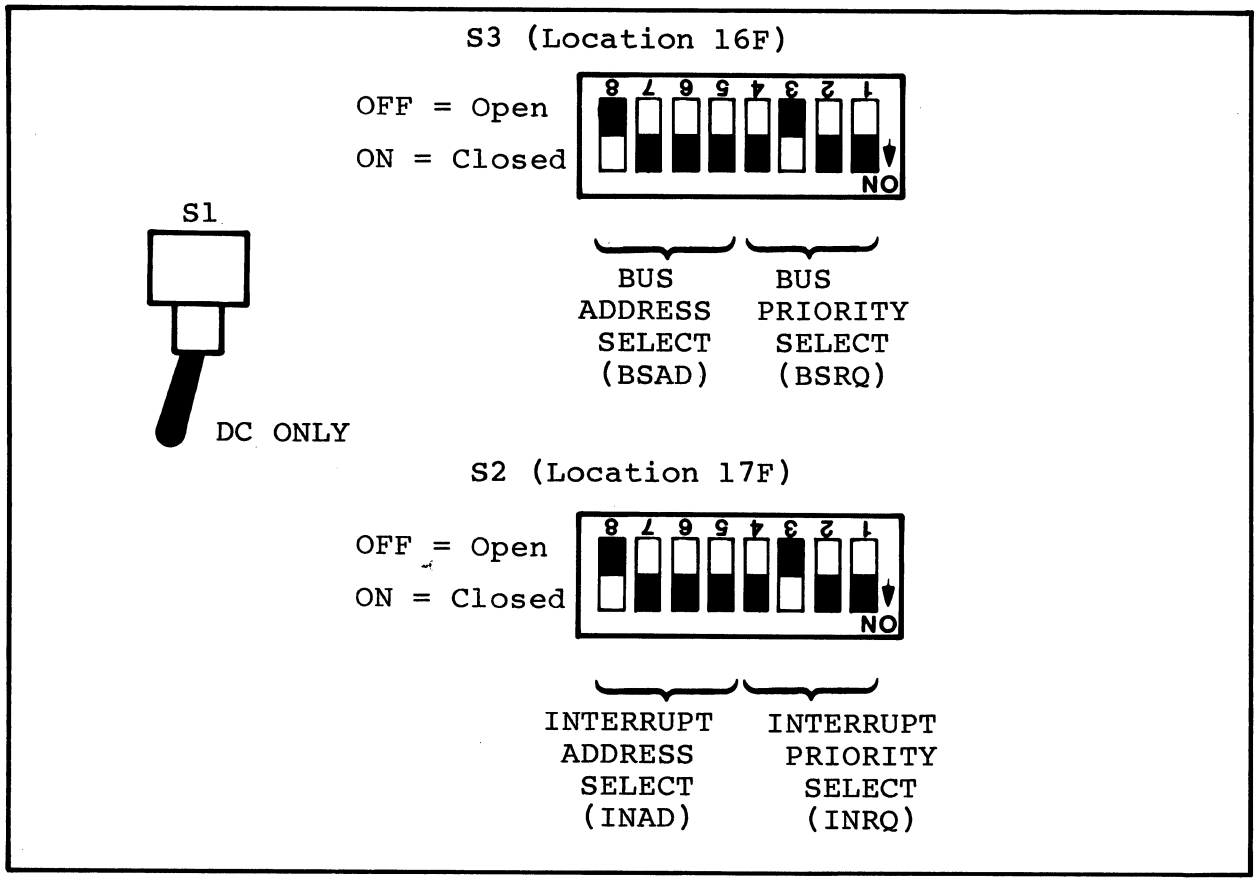


Figure 3-13. IOP Switch Settings, "Mixed DC + ACLC"

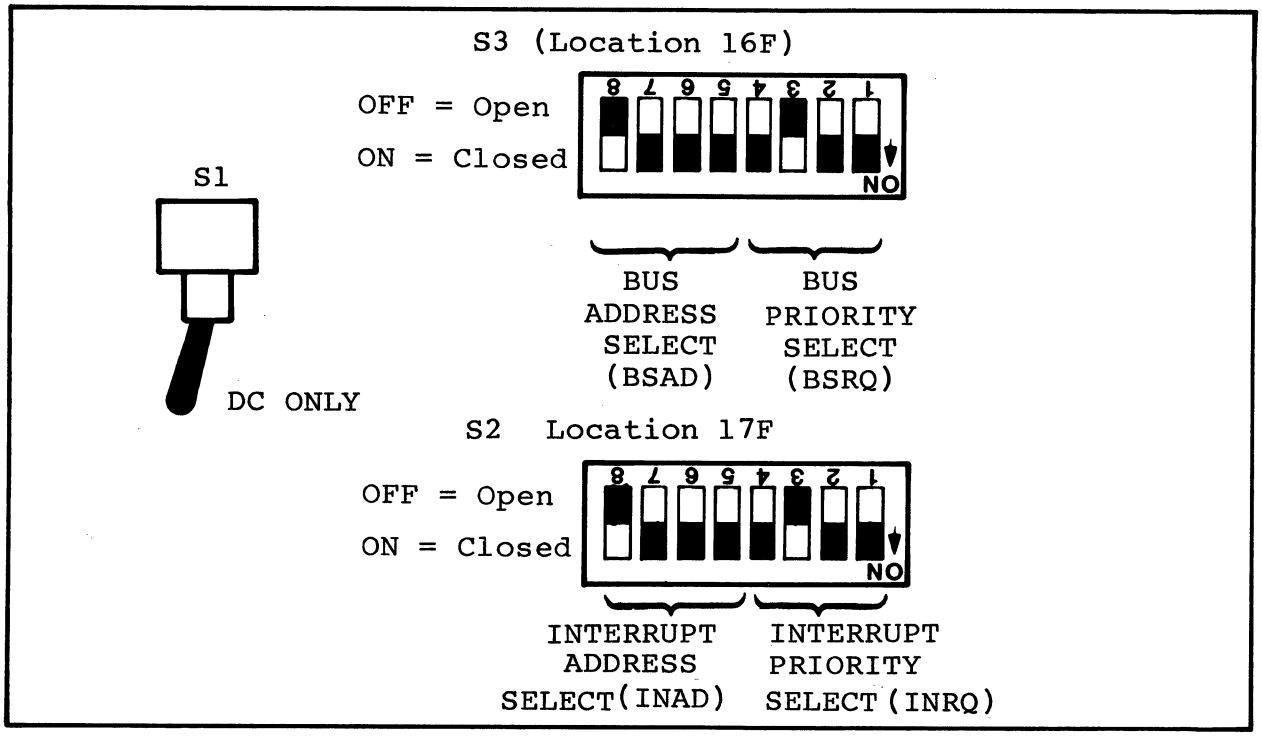
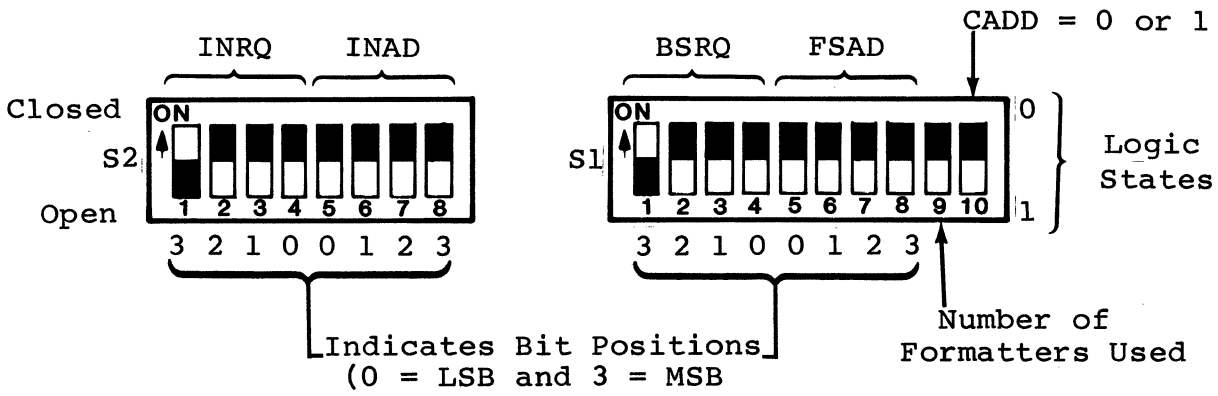


Figure 3-14. IOP Switch Settings, "ACLC ONLY"

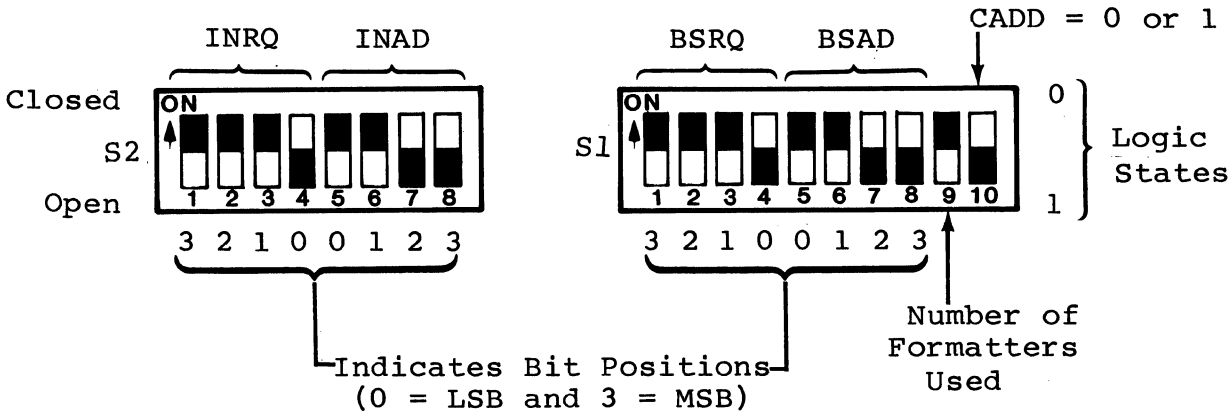


Note 1: The switch settings shown in this figure are for the "first" MT/P board.

Note 2: The controller address (CADD) is 0, selected by S1, throw #10 to ON.

Note 3: S1, throw #9, selects the "number of formatters used" option. Normally the F/F (formatter-to-formatter) mode will be selected, indicating that each MTU being serviced by this controller is equipped with its own formatter board. When set to the OFF position, the alternate D/D (drive-to-drive) mode is selected, indicating that the two MTUs share the same formatter board in a "daisy-chained" manner.

Figure 3-15. MT/P Switch Settings



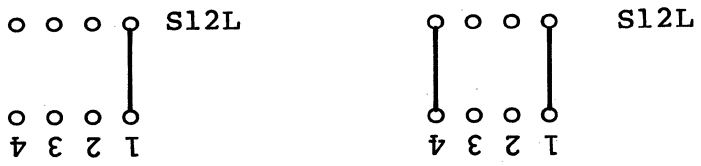
Note 1: The switch settings shown in this figure are for the "second" MT/P board.

Note 2: The controller address (CADD) is 1, selected by S1, throw #10 to OFF.

Note 3: S1, throw #9, selects the "number of formatters used" option. Normally the F/F (formatter-to-formatter) option will be selected, indicating that each MTU being serviced by this controller is equipped with its own formatter board. When set to the OFF position, the alternate D/D (drive-to-drive) option is selected, indicating that the two MTUs share the same formatter board in a "Daisy-chained" manner.

Figure 3-16. Second MT/P Board Switch Settings

Jumpered switch settings are shown in this figure for the "first" and "second" MTU formatter boards. The formatter board is part of the MTU assembly



First MTU Formatter Second MTU Formatter**

** This switch is used for systems that require one MTU printer controller to operate two MTUs, each equipped with a dedicated formatter. This switch setting is used in conjunction with the D/D option. This option is described briefly in Figure 3-12.

o o = Open = OFF

o——o = Closed = ON

Figure 3-17. MTU Formatter Jumpered Switch Settings

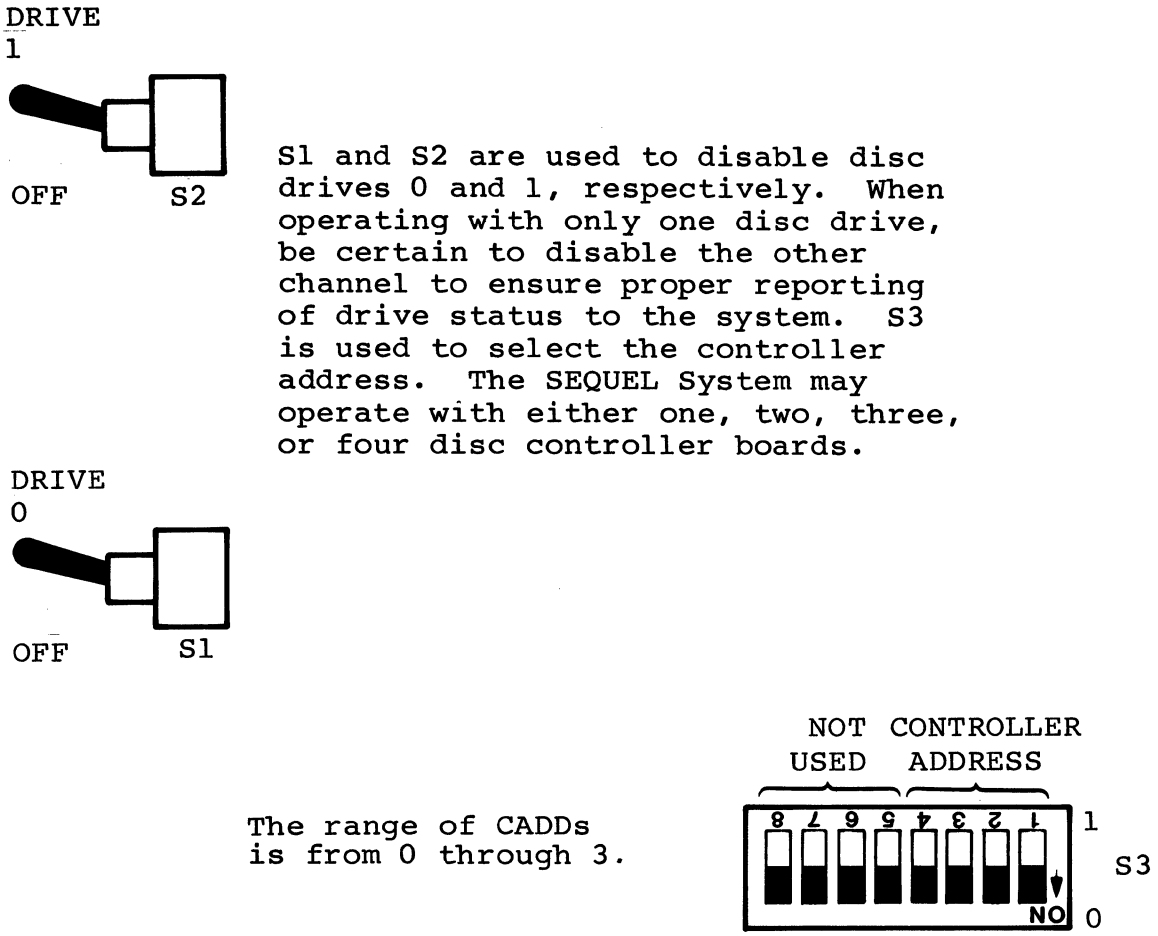
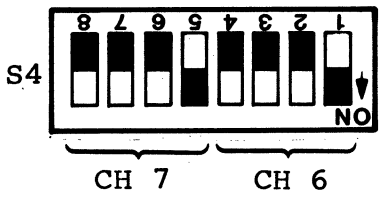
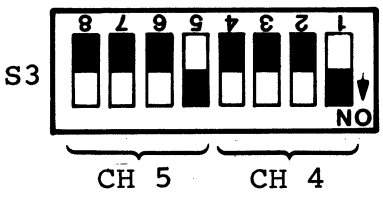


Figure 3-18. Disc Controller Switch Settings



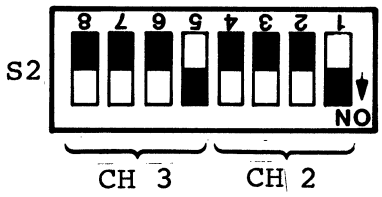
Closed = ON = Logical 0

Open = OFF = Logical 1

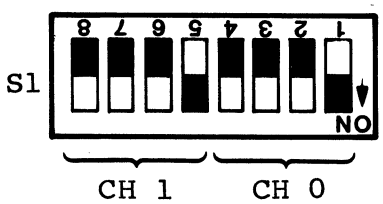


Shown here in S3 is the value in both channels 4 and 5 to 9600 baud. Each channel may operate at a different baud rate. Refer to Table 3-4 for alternate values.

BAUD RATE



The binary count on all controller address switches begins with throw #1. The range of CADDs must not contain any gaps in the logical sequence. Refer back to notes on the IOP for a brief explanation of the CADD range for the ACLCs.



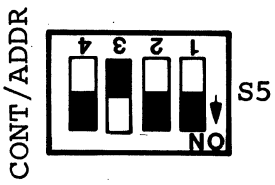
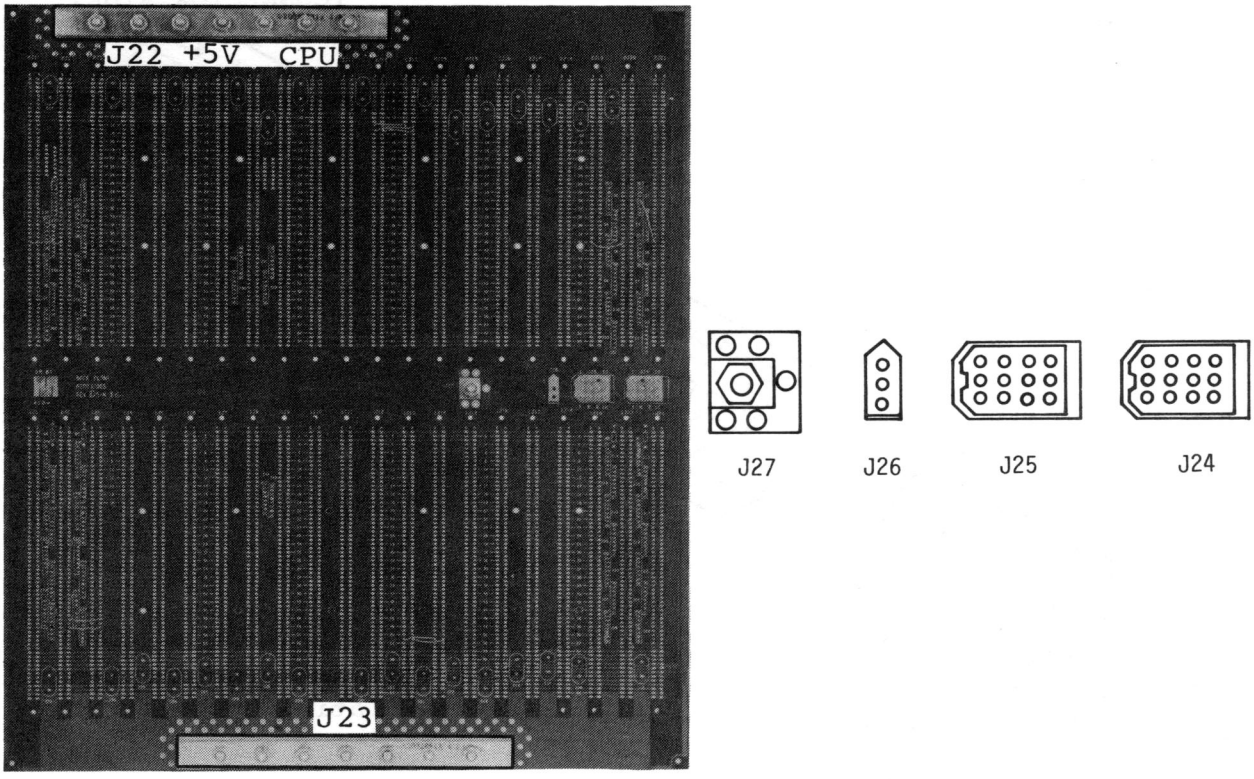


Figure 3-19. ACLC Switch Settings



POWER CONNECTIONS

	J22	J23	J26
*	1 + 5V CPU	1 GND	1 + 5V CPU
	2 + 5V CPU	2 GND	2 GND
	3 + 5V CPU	3 GND	3 + 12V CPU
	4 + 5V CPU	4 GND	J27
	5 + 5V CPU	5 GND	
	6 + 5V CPU	6 GND	** 1 + 5V MEM
	7 + 5V CPU	7 GND	

	J24	J25
**	1 + 12V MEM	1 PFAT/
	2 + 5V MEM	2 + 5V CPU SENSE
*	3 + 12V CPU	3 PFIR/
	4 + 12V MEM	4 PRDY
	5 + 5V MEM	5 COMM SENSE
	6 + 12V CPU	6 CPSS/
**	7 - 5V MEM	* 7 -5V CPU
	8 + 5V MEM	8 COMM SENSE
	9 + 12V CPU	9 MPSS/
*	10 - 12V CPU	10 -5V CPU
	11 + 5V MEM	11 RTCK/
	12 - 12V CPU	12 MPFI/

* CPU VOLTAGES
** MEMORY VOLTAGES

TOLERANCES

VDC	I (MAX)		+/-
+5.0V CPU	115A	3%	0.15V
-5.0V CPU	5A	3%	0.15V
+12.0V CPU	5A	3%	0.36V
-12.0V CPU	2A	3%	0.36V
+5.0V MEM	16A	3%	0.15V
+12.0V MEM	5A	3%	0.36V
-5.0V MEM	0.2A	5%	0.25V

Figure 3-20. SEQUEL Backplane Voltages and Test Points

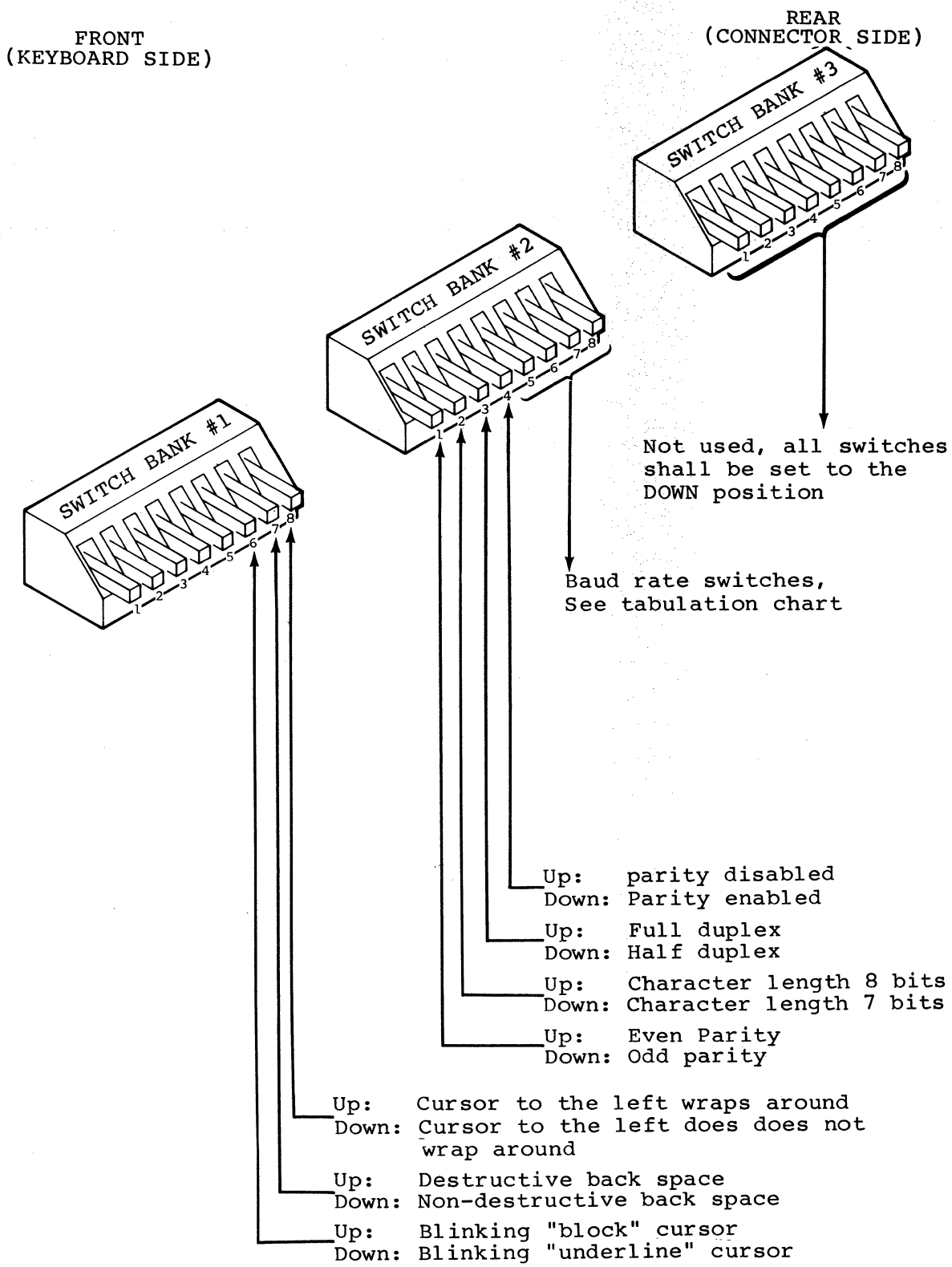
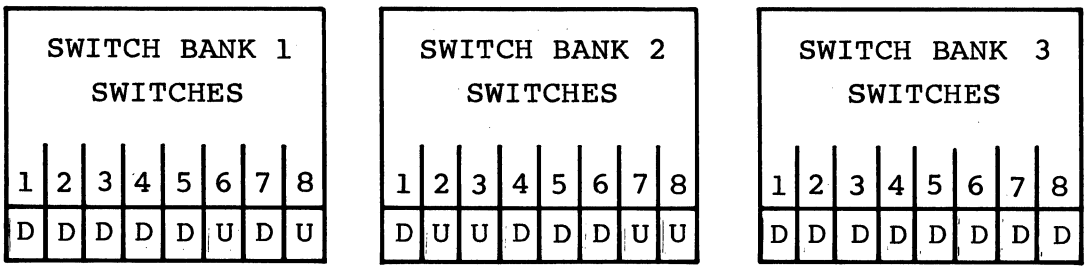


Figure 3-21. PRISM 4A Switch Options

The terminal will be delivered with the following switch settings:



SWITCH BANK 1 SWITCHES							
1	2	3	4	5	6	7	8
D	D	D	D	D	U	D	U

SWITCH BANK 2 SWITCHES							
1	2	3	4	5	6	7	8
D	U	U	D	D	D	U	U

SWITCH BANK 3 SWITCHES							
1	2	3	4	5	6	7	8
D	D	D	D	D	D	D	D

The switch settings above establish the following conditions:

- Blinking "block cursor
- Non-destructive back space
- Cursor to the left wraps around
- Odd parity
- Character length 8 bits
- Full duplex
- Parity disabled
- 9600 baud

Figure 3-22. PRISM 4 Switch Settings

		SWITCH #							
		1	2	3	4	5	6	7	8
PARITY	ODD	DOWN							
	EVEN	UP							
WORD LENGTH	7-BIT		DOWN						
	8-BIT		UP						
DUPLEX	HALF			DOWN					
	FULL			UP					
PARITY USAGE	ENABLE				DOWN				
	INHIBIT				UP				
BAUD RATE	110					DOWN	DOWN	DOWN	DOWN
	150					DOWN	UP	UP	UP
	300					DOWN	DOWN	DOWN	UP
	600					UP	DOWN	DOWN	UP
	1200					DOWN	DOWN	UP	DOWN
	1800					DOWN	UP	UP	DOWN
	2400					DOWN	UP	DOWN	DOWN
	3600					UP	UP	UP	DOWN
	4800					DOWN	UP	DOWN	UP
	9600					DOWN	DOWN	UP	UP
	19200					UP	DOWN	UP	UP

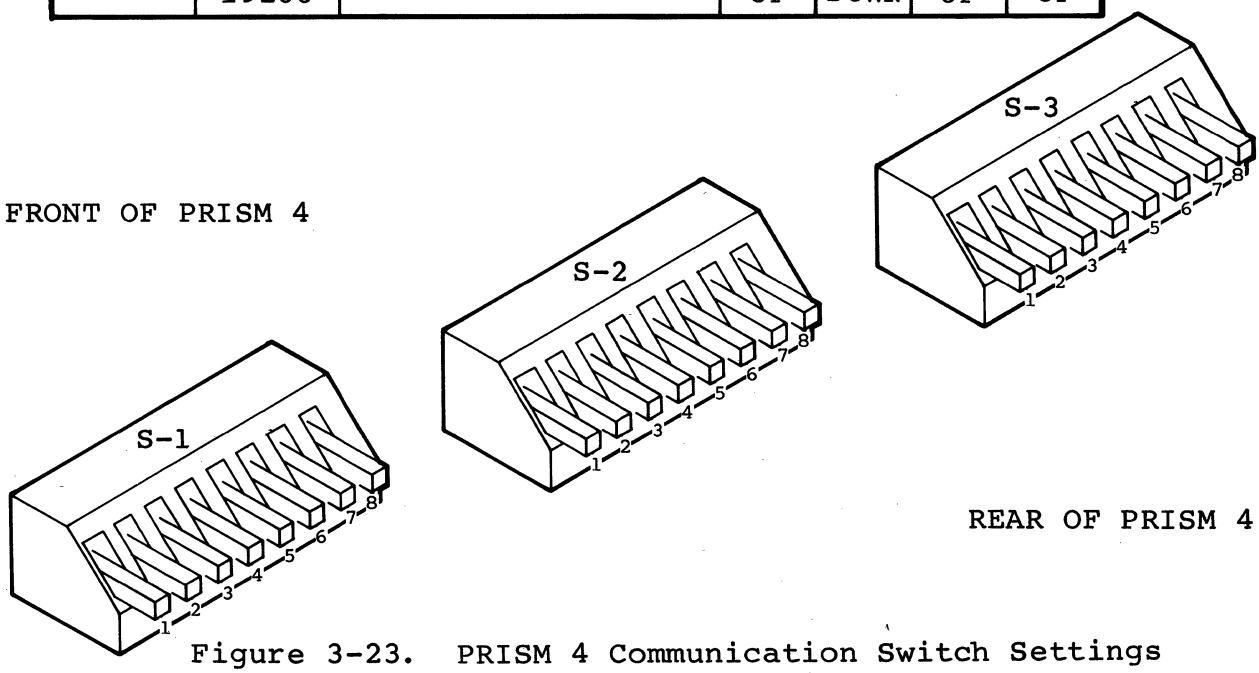
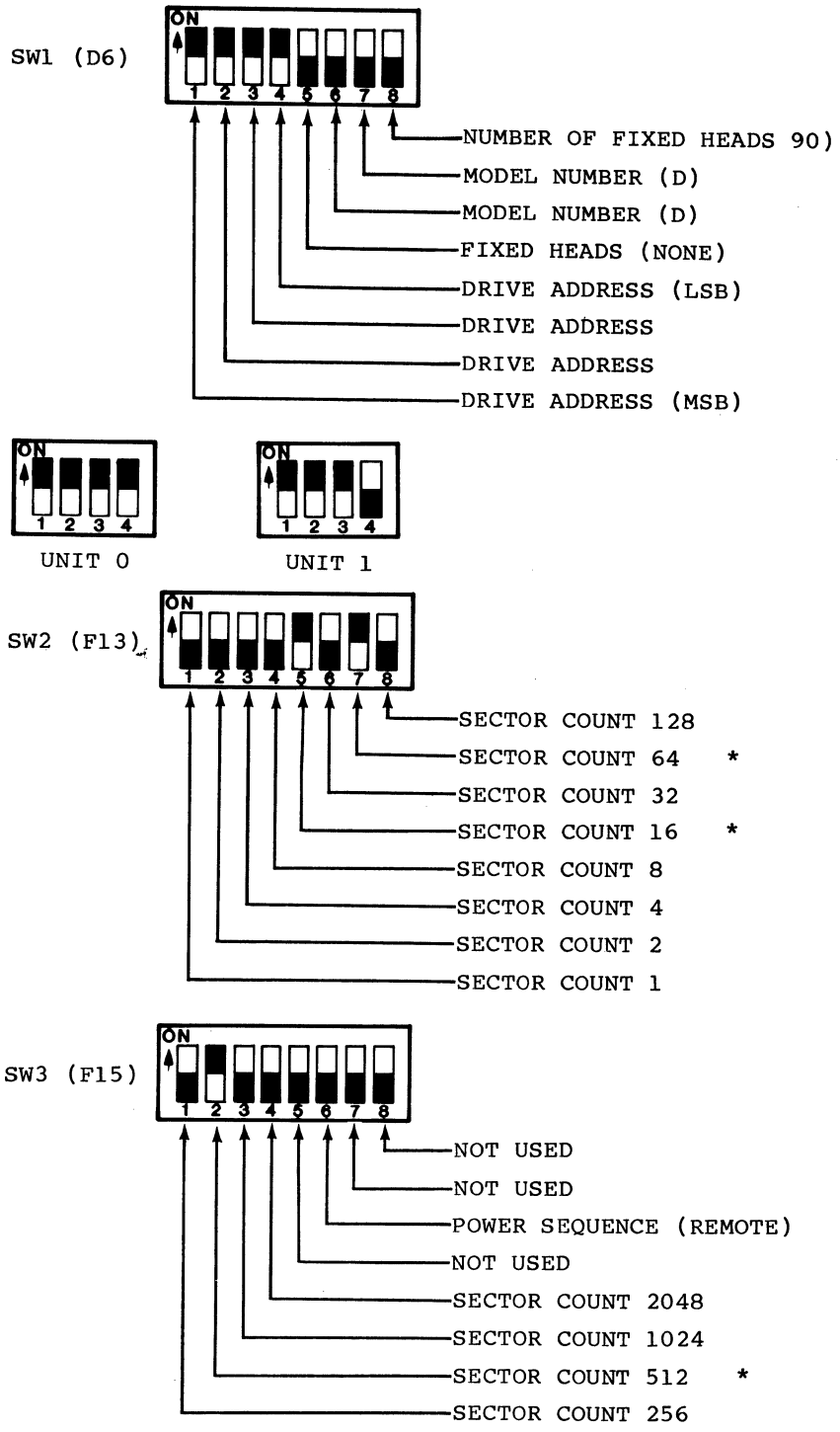


Figure 3-23. PRISM 4 Communication Switch Settings with Baud Rate Options



ON = CLOSED, JUMPERED, LOGICAL 0

OFF = OPEN, NOT JUMPERED, LOGICAL 1

* SEQUEL IS FORMATTED FOR 593 SECTORS.

Figure 3-24. REFLEX II Logic Interface Board Switches

TABLE 3-4

ACLC BAUD RATE SELECT

S1 (8 4	S2 7 3	S3 6 2	S4 5 1)	Baud Rate
C	C	C	C	50
C	C	C	0	75
C	C	0	C	110
C	C	0	0	134.5
C	0	C	C	150
C	0	C	0	300
C	0	0	C	600
C	0	0	0	1200
0	C	C	C	1800
0	C	C	0	2000
0	C	0	C	2400
0	C	0	0	3600
0	0	C	C	4800
0	0	C	0	7200
0	0	0	C	9600
0	0	0	0	19200

C = Closed = ON = Logical 1

0 = Open = OFF = Logical 0

TABLE 3-5

BUS PRIORITY ASSIGNMENT

Level	BUS REQUEST	BUS ADDRESS	I/O Device
Highest	BSRQ3	0000	MT/P Addresses E0, E1, E4, E5
	BSRQ3	0001	
	BSRQ3	0010	
	BSRQ3	0011	
	BSRQ2	0100	IOP Address FC
	BSRQ2	0101	
	BSRQ2	0110	
	BSRQ2	0111	
	BSRQ1	1000	IOP Address FB
	BSRQ1	1001	
	BSRQ1	1010	
	BSRQ1	1011	
	BSRQ0	1100	MT/P Addresses
	BSRQ0	1101	
	BSRQ0	1110	
Lowest	BSRQ0	1111	DMP (Hard wired)

TABLE 3-6

INTERRUPT PRIORITY ASSIGNMENT

Level	INT. REQUEST	INT ADDRESS	I/O Device
Highest	INRQ3	0000	MT/P Addresses E0, E1, E4, E5
	INRQ3	0001	
	INRQ3	0010	
	INRQ3	0011	
	INRQ2	0100	IOP Address FC
	INRQ2	0101	
	INRQ2	0110	
	INRQ2	0111	
	INRQ1	1000	IOP Address FB
	INRQ1	1001	
	INRQ1	1010	
	INRQ1	1011	
	INRQ0	1100	MT/P Address E2, E3, E6, E7
	INRQ0	1101	
	INRQ0	1110	
Lowest	INRQ0	1111	

TABLE 3-7

IOP SWITCH SETTINGS

BUS REQUEST	BUS ADDRESS	SWITCHES	INT. REQUEST	INT. ADDRESS	SWITCHES
		S3 (16F) 87654321			S2 (17F) 87654321
BSRQ3	0000	CCCCCCCO	INRQ3	0000	CCCCCCCO
BSRQ3	0001	CCCOCCCO	INRQ3	0001	CCCOCCCO
BSRQ3	0010	CCOCCCCO	INRQ3	0010	CCOCCCCO
BSRQ3	0011	CCOOCCCO	INRQ3	0011	CCOOCCCO
BSRQ2	0100	COCCCCOC	INRQ2	0100	COCCCCOC
BSRQ2	0101	COCOCCOC	INRQ2	0101	COCOCCOC
BSRQ2	0110	COOCCCOC	INRQ2	0110	COOCCCOC
BSRQ2	0111	COOOCCOC	INRQ2	0111	COOOCCOC
BSRQ1	1000	OCCCCOCC	INRQ1	1000	OCCCCOCC
BSRQ1	1001	OCCOCOCC	INRQ1	1001	OCCOCOCC
BSRQ1	1010	OCOCCOCC	INRQ1	1010	OCOCCOCC
BSRQ1	1011	OCOOCOCC	INRQ1	1011	OCOOCOCC
BSRQ0	1100	OOCCOCCC	INRQ0	1100	OOCCOCCC
BSRQ0	1101	OOCOOCCC	INRQ0	1101	OOCOOCCC
BSRQ0	1110	OOOCOCCC	INRQ0	1110	OOOCOCCC
BSRQ0	1111	OOOOOCCC	INRQ0	1111	OOOOOCCC

Note: C = Close = ON = Zero

O = Open = OFF = One

TABLE 3-8

MT/P SWITCH SETTINGS

BUS REQUEST	BUS ADDRESS	SWITCHES	INT. REQUEST	INT. ADDRESS	SWITCHES
		S1 87654321			S2 87654321
BSRQ3	0000	CCCCCCCO	INRQ3	0000	CCCCCCCO
BSRQ3	0001	CCCOCCCO	INRQ3	0001	CCCOCCCO
BSRQ3	0010	CCOCCCCO	INRQ3	0010	CCOCCCCO
BSRQ3	0011	CCOOCCCO	INRQ3	0011	CCOOCCCO
BSRQ2	0100	COCCCCOC	INRQ2	0100	COCCCCOC
BSRQ2	0101	COCOCCOC	INRQ2	0101	COCOCCOC
BSRQ2	0110	COOCCCOC	INRQ2	0110	COOCCCOC
BSRQ2	0111	COOOCCOC	INRQ2	0111	COOCCCOC
BSRQ1	1000	OCCCCOCC	INRQ1	1000	OCCCCOCC
BSRQ1	1001	OCCOCOCC	INRQ1	1001	OCCOCOCC
BSRQ1	1010	OCOCCOCC	INRQ1	1010	OCOCCOCC
BSRQ1	1011	OCOOCOCC	INRQ1	1011	OCOOCOCC
BSRQ0	1100	OOCCOCCC	INRQ0	1100	OOCCOCCC
BSRQ0	1101	OOCOOCCC	INRQ0	1101	OOCOOCCC
BSRQ0	1110	OOOCOCCC	INRQ0	1110	OOOCOCCC
BSRQ0	1111	OOOOOCCC	INRQ0	1111	OOOOOCCC

Note: C = Close = ON = Zero

O = Open = OFF = One

SECTION 4

THEORY OF OPERATION

INTRODUCTION

The SEQUEL CPU is a microprogrammable processor designed to execute the REALITY assembly language instruction set. The design criteria for the computer called for a functional operation that would be transparent to existing REALITY users. To achieve the "through-put" and "bench-mark" requirements, the following enhancement features have been incorporated in the hardware:

(a) Wider data paths
(b) High degree of parallelism
(c) Reduced memory accesses
(d) Special functions
(e) Powerful architecture at four levels: Bus, CPU, ALU, microsequencer
(f) Elimination of clock penalties of jumps and skips

The application of these features allows an ongoing function in the foreground and preparation for the subsequent command in the background to be performed. The "high degree of parallel hardware" provides for the pipelining of instruction and data-paths.

GENERAL CENTRAL PROCESSING UNIT CHARACTERISTICS

General CPU characteristics are listed below. These have been implemented in order to significantly increase processing speed while retaining operating flexibility down to the byte level.

(a) 32 bit data path; CPU, I/O bus, memory
(b) 24 bit I/O and memory address bus
(c) 600 nanosecond memory cycle time
(d) 150 nanosecond microprogram cycle time
(e) 1-4 byte memory accesses in one cycle
(f) 64 bit main firmware word size
(g) Hardware rotation capability
(h) Character test function

CPU HARDWARE

The CPU is implemented on three printed circuit boards. Refer to Figure 4-1, the SEQUEL CPU block diagram, for an overview of the computer. The diagnostic maintenance processor (DMP) is included in this drawing since many of the normal CPU functions are done on the DMP board. These four printed circuit boards

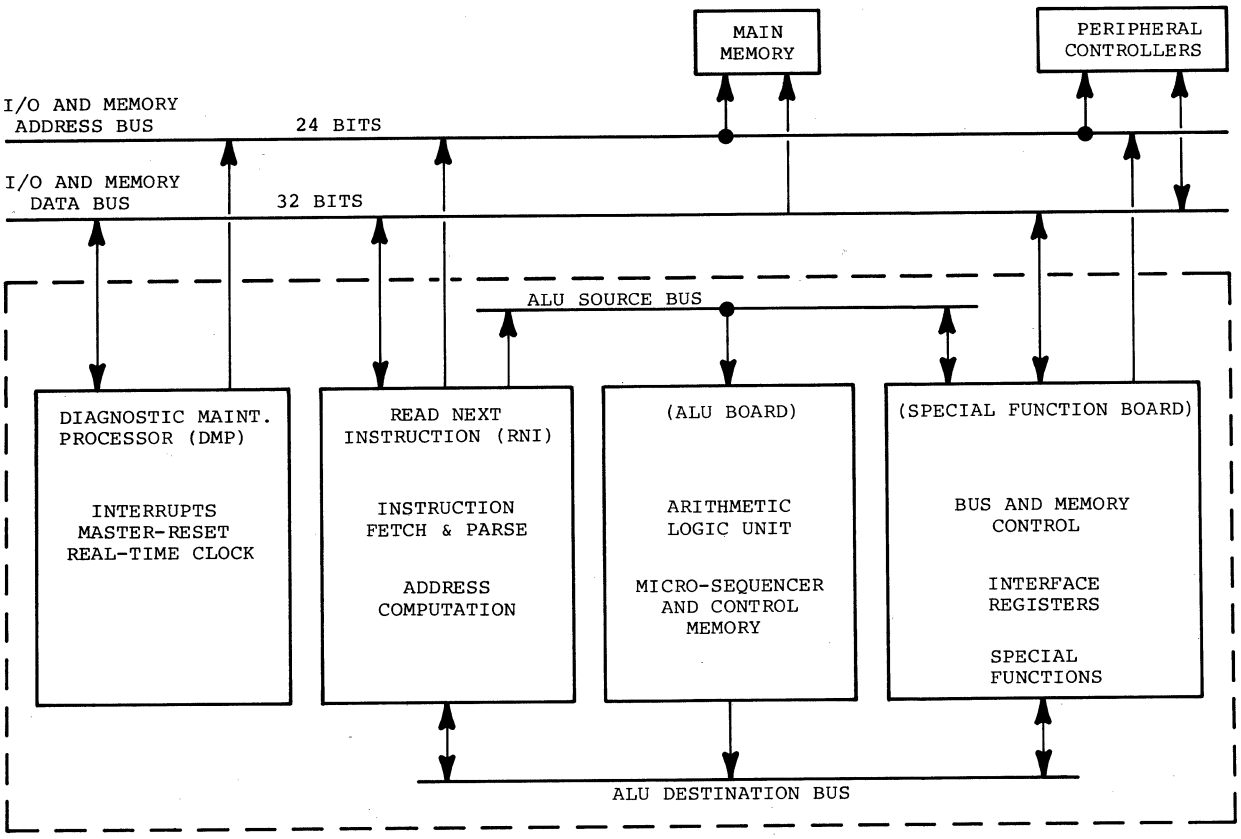


Figure 4-1. SEQUEL CPU Block Diagram

have a specific zone in the SEQUEL chassis. The four slots provided are board interchangeable except that the three CPU boards must be in three successive positions to accommodate the foreplane connections.

The ALU source and destination bus are unique to the CPU hardware. The hardware is implemented in tri-state logic with all control originating from the firmware word.

The 64 bit firmware word is located on the ALU board with the output being latched into the command register. All CPU functions begin with the start of the firmware data bits that reference the time when the data is available from the command register. All CPU firmware operations are completed within one clock cycle. Refer to Figure 4-2 for the basic operational structure.

MAJOR FUNCTIONS OF THE THREE CPU BOARDS

The major functions of the three CPU boards are summarized as follows:

Arithmetic Logic Unit Board

- 2901 ALU and carry look-ahead
- Carry and shift input select
- Condition code logic
- Loop counter
- Microsequencer
- Firmware command register
- Control clock logic

Read Next Instruction Board

- Instruction fetch and parse picoprocessor
- Program address register
- Instruction registers
- Mapping ROMS (for FW branching)
- Address register files

Special Functions Board

- Bus and memory control
- Data address registers
- Memory read/write data registers
- "X" register
- Rotate/character test

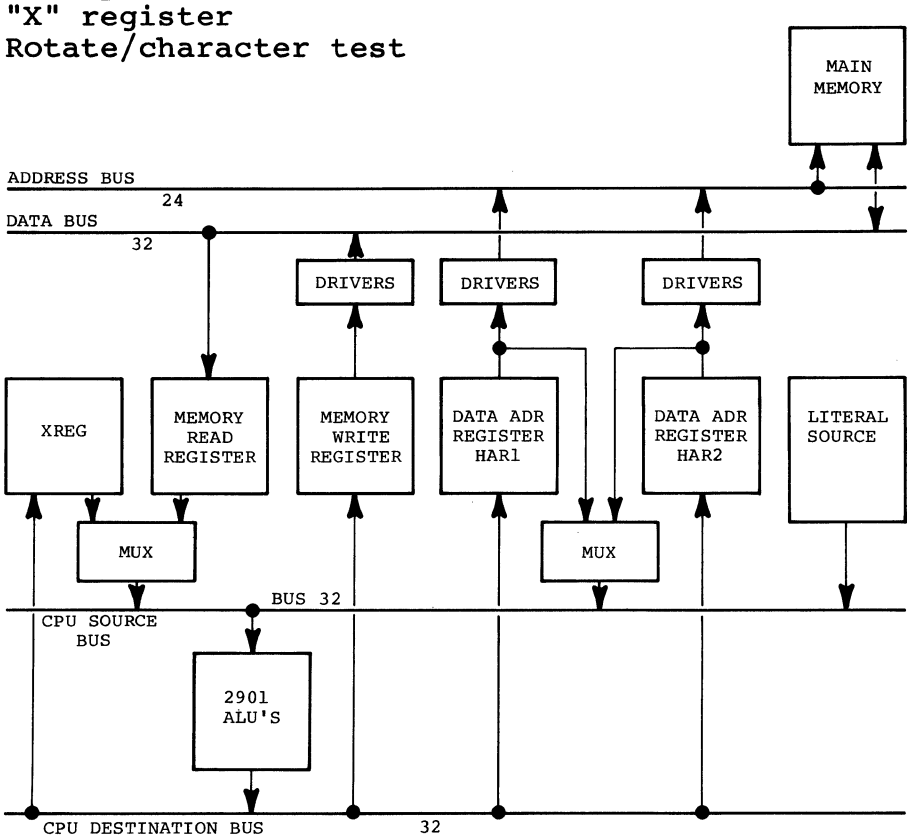


Figure 4-2. CPU Basic Operational Structure

ARITHMETIC LOGIC UNIT BOARD

The nucleus of the CPU is the ALU board. This PCB contains the 32 bit ALU implemented with 2901 bit slice microprocessors and 2902 carry look-ahead chips. The ALU detailed flow diagram is shown in Figure 4-3. Full capability of the 2901 is provided by incorporating all of the 2901 control terms directly in the main firmware word. Complete use of the input and output bus is provided and supported by the firmware word. This allows the CPU to do arithmetic and logical operations on registers external to the 2901, through the 2901, in one firmware clock.

Incorporated with 2901 are shift input selection logic, carry-in logic, condition code logic and multiply/divide logic. The ALU board also has 16 flags that can be set and tested by the firmware. These flags can assist in keeping track of status, operation modes and other related functions.

There is also an eight bit loop counter that serves a dual purpose. The counter can be set to values from 0-255, tested by the firmware, and then be automatically incremented. This is used to facilitate firmware loops with minimum updating required. The second purpose of the loop counter is to provide firmware with the capability of setting up a programmable direct branching capability.

The other main function of the ALU board is the microsequencer. The microsequencer determines which command is executed on each clock by generating a 13 bit address word. This address selects a 64 bit firmware word from the command ROM which is loaded into the command register. The breakdown of the 64 bit firmware word is covered in detail in Section 3.

READ NEXT INSTRUCTION BOARD

The RNI board has two major functions:

(a) Instruction fetching and decoding
(b) Data address computation

Instruction Fetching and Decoding

The RNI hardware fetches REALITY instructions and parses them into the various elements used by the ALU/microsequencer. The RNI block diagram is shown in Figure 4-3A. The RNI hardware has its own picoprocessor. It operates somewhat independent of the ALU/microsequencer.

The RNI hardware is started at a main memory location by loading its program address register. It then fetches and parses the first instruction, and puts the results into a holding register called the second rank instruction register. The hardware

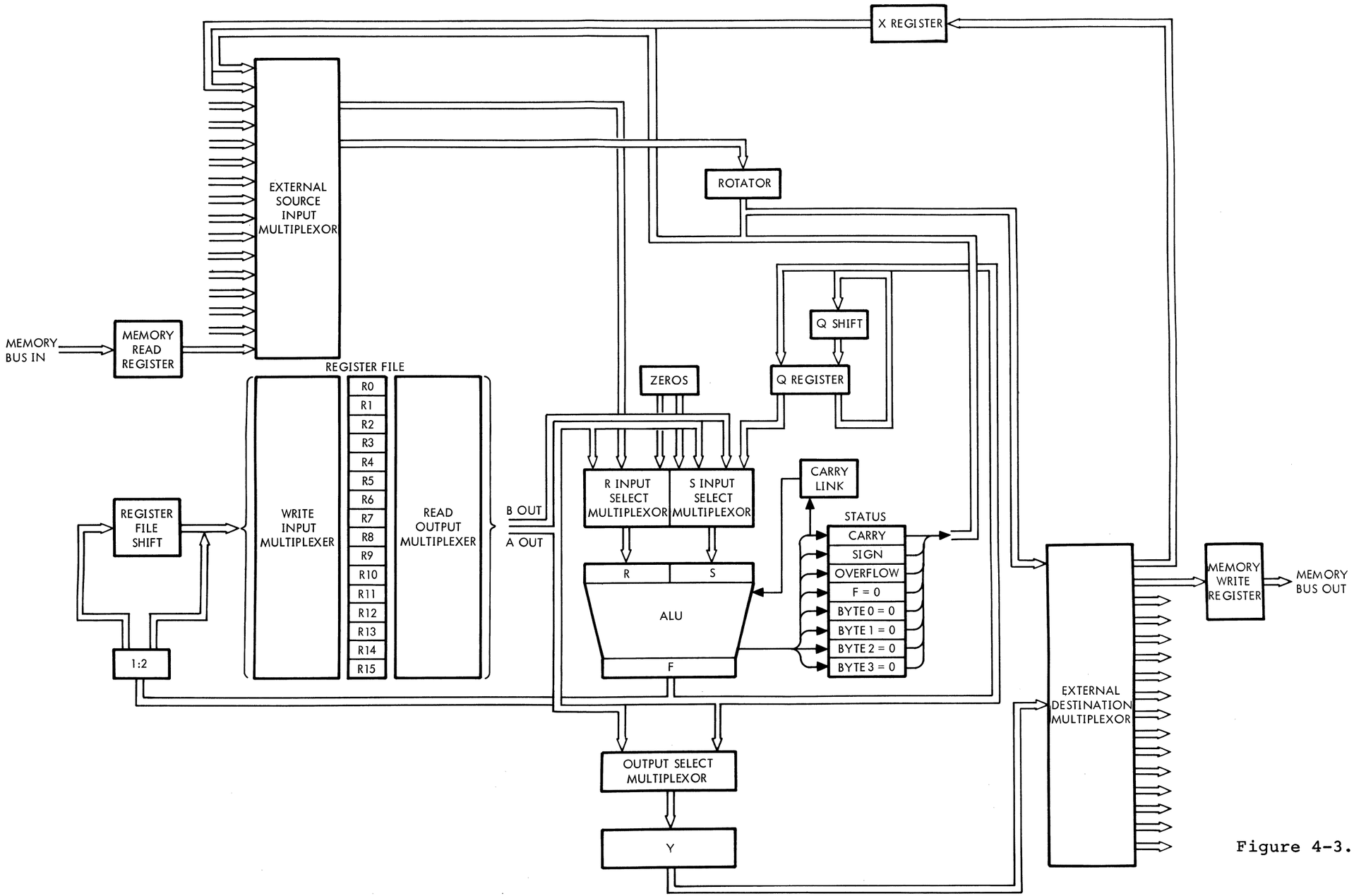


Figure 4-3. ALU Flow Diagram

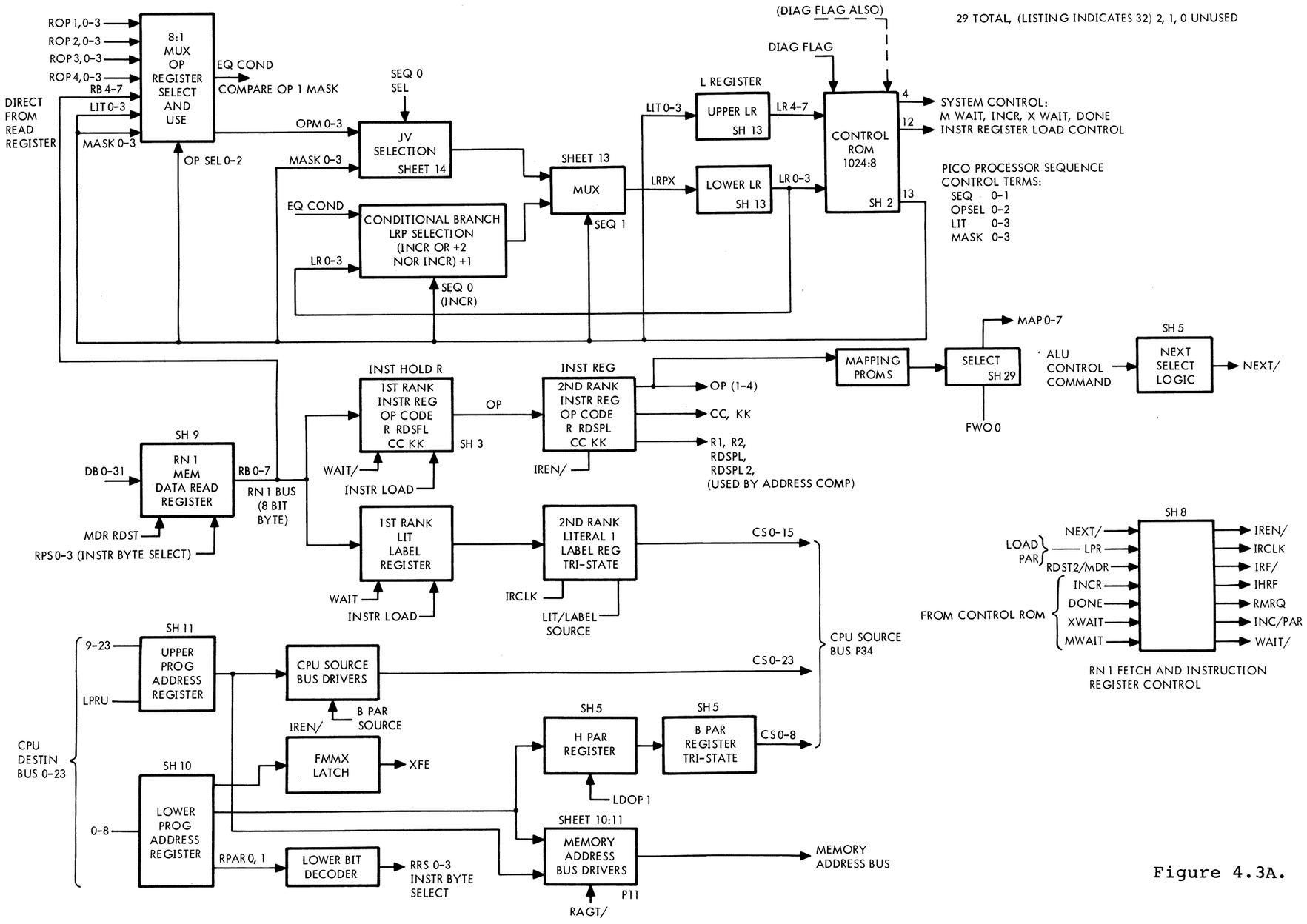


Figure 4.3A. RNI Block Diagram

then fetches the next instruction and starts parsing it while the first one is being executed by the ALU microsequencer. The RNI operation continues until it is full. It then waits until it receives a next instruction command or until its program counter is loaded with a new value.

RNI Address Computation

Addresses for REALITY data are either defined in the REALITY instruction, or are implied by the instruction. The address is generated by combining an address from a file, a displacement, and a constant. Special hardware is provided to perform an address computation in one firmware word. The source of data, and the address computation operation, are determined by the firmware directly or by the register designation and word length specified in the REALITY instruction. The result of the address computation can be placed in one of two hardware address registers.

SPECIAL FUNCTIONS BOARD

The special functions board contains the following functions:

(a) Memory and bus control
(b) Memory data address registers
(c) "X" register
(d) Data rotate logic
(e) Character test logic

Memory and Bus Control

The memory and bus control generates all timing signals and determines priorities for all memory activity. This includes refresh, I/O, RNI and CPU cycles. It also decodes the firmware commands for CPU memory cycle requests.

Memory Registers

There are two 24 bit address registers, (HAR1 and HAR2). Two are needed to support high speed dual operand and string move operations. These registers are loaded from the the ALU or ADC via the CPU destination bus.

There is a 32 bit memory read register that is loaded from the I/O and memory data bus during a CPU read cycle. This register is an input to the CPU source bus. There is a 32 bit memory write register that is loaded via the CPU destination bus. This register is outputted to the I/O and memory data bus during a CPU write cycle.

"X" Register

The 32 bit "X" register is used as a working register for ALU outputs, as a source and destination for rotate operations, and as a shift register for multiply and divide operations.

Rotate Logic

Since data in memory is not necessarily aligned on double word boundaries, data must be rotated after it is fetched. For the same reasons, data must be rotated when it is written back into memory. Special hardware is provided to rotate any multiple of 8 bits on one firmware instruction.

The rotate hardware has as its input the CPU source bus, and its output is the CPU destination bus. Because of this, any one of the source registers can be rotated, and any destination register can receive the result. The ones normally used for rotate sources are the "X" register and read memory register. The number of bytes of the rotation is determined by the firmware directly, or by the memory address and word length. A sign or zero extend function can also be performed by the rotate operation.

Character Test

When strings of data in memory are scanned or moved, the data is tested for values called delimiters. There are four fixed delimiter values and three programmable values. Up to seven delimiters may be selected for a string operation. These are specified in a mask byte, which is part of the REALITY string instruction.

Special hardware is provided to simultaneously test four characters for any combination of the seven delimiters during one memory cycle time of 600 nanoseconds. The four fixed delimiters are tested on one firmware clock directly from the CPU source bus. The three variable delimiters are tested using the ALU, so one firmware clock is required for each delimiter. The results of the test are provided to the ALU/microsequencer as branch conditions.

SEQUEL BYTE HANDLING CAPABILITY

REALITY instructions vary in length from 1-6 bytes, and data lengths vary from 1-6 bytes. Strings can be any length. Data and instructions also can start at any byte boundary. Because of these features, the SEQUEL CPU is highly efficient at dealing with bytes even though it is a full 32 bit machine. These features are incorporated to minimize the housekeeping normally associated with byte handling. Refer to Figure 4-4 for the detailed block diagram of the basic hardware architecture. These features are as follows:

(a) Memory interface
(b) Data position alignment
(c) Individual byte condition testing
(d) RNI and ADC byte handling capability

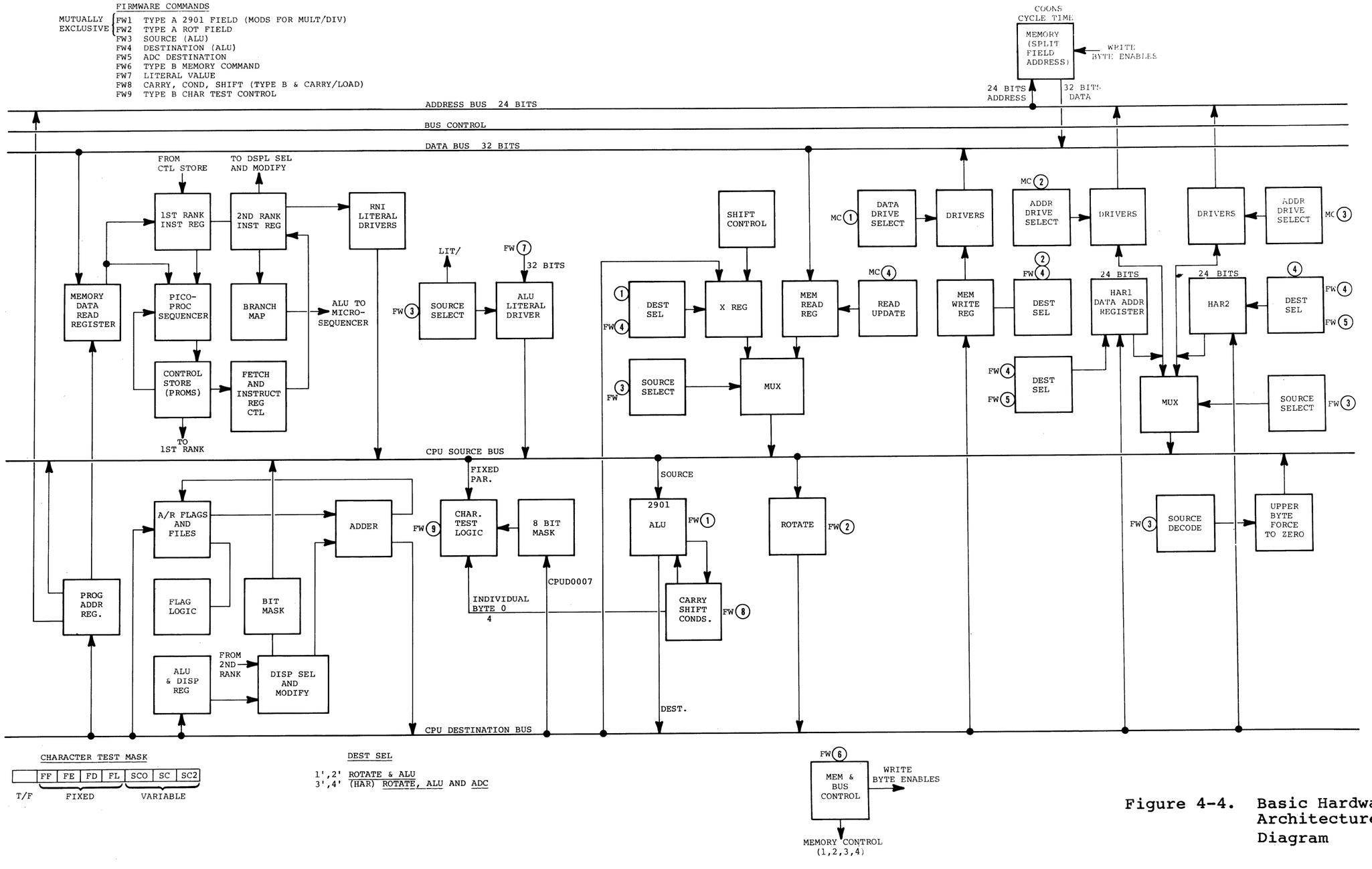


Figure 4-4. Basic Hardware Architecture Block Diagram

Memory Interface

The memory interface has split field addressing, individual byte error correction and detection, and variable length (1,2,3,4 byte) write capability. This means that any contiguous string of bytes from 1-4 can be accessed in one memory cycle even if the string occupies two adjacent 32 bit double words in memory. The memory address points to the most significant byte of the string.

Data Position Alignment

CPU data can be quickly adjusted on byte boundaries to align the ALU and memory for both reads and writes. Sign extension or zero extension can also be done simultaneously with the alignment (called rotation) when the data length is 1,2,3,4 bytes.

Individual Byte Condition Testing

Simultaneous zero condition testing and fixed parameter comparison testing are provided for each byte in the CPU. This makes it possible to do character testing four bytes at a time. Variable length overflow condition is also provided.

RNI and ADC Byte Handling Capability

Both the RNI and ADC functions have features to handle byte level instructions and data, such as byte level parsing, byte displacement handling and byte length sensing.

REALITY INSTRUCTION EXAMPLE

In order to illustrate the functions of the ALU, RNI and ADC, the following example is included. The example consists of the processing by the CPU of one REALITY software instruction. Namely, the subtract and branch instruction (Opcode F, BDZ). This instruction is 6 bytes in length as shown in Figure 4-5. The instruction causes the CPU to do the following functions:

Two data values are fetched from memory

B value is subtracted from A

The result is stored in location A

The result is tested for a conditional branch. If the condition is met, the program branches to the branch location specified in bytes 5 and 6. If the condition is not, the program advances to the next location.

Data locations are specified in the instruction by Rx, and DSPLx. Rx is a 4 bit pointer to one of 16 registers that contains an address of a memory frame. DSPL is a displacement within the

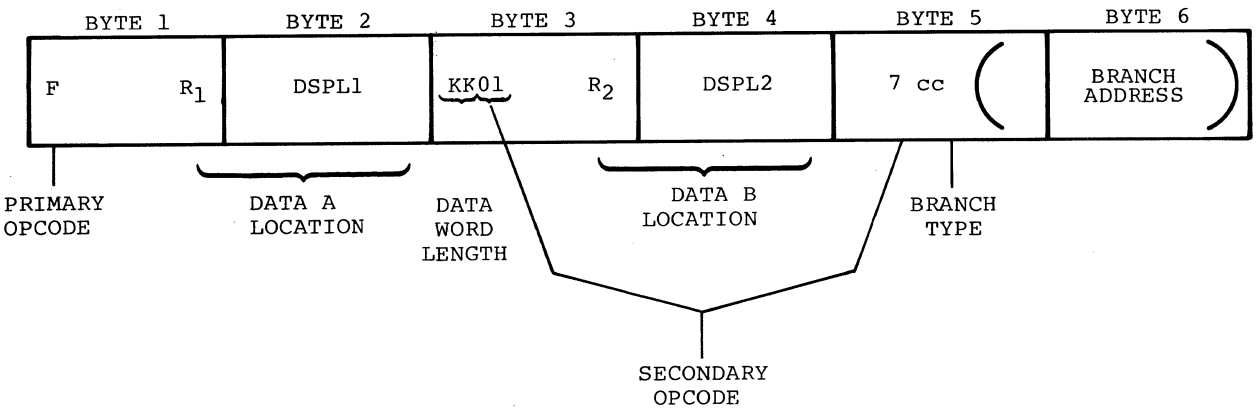


Figure 4-5. REALITY Opcode F Subtract and Branch

frame. The data address is generated by the address computation which combines the value in the register designated by the "R" with the displacement, and adds a constant, if required. Since there are two data locations in this instruction, two separate address computations are required. The data word length is specified by KK. It can be 1,2,4 or 6 bytes in length. The same word length applies to both variables. In the example the word length is 4. There are four branch types which are specified by CC. The branch address is 10 bits in length.

The instruction is identified to the RNI hardware and to the CPU by the primary and secondary OP codes:

Primary: "F" (Byte 1)
Secondary: 01 (binary) (Byte 3)
Secondary: "7" (Byte 5)

This opcode information is used by RNI for parsing, and by the CPU for map branching. Since the secondary OP code value exceeds 4 bits, it is necessary to do a primary map branch, followed by a secondary map branch.

The RNI parses the instruction into its hardware instruction registers as shown in Table 4-1. Assuming that this instruction (BDZ) is the first instruction to be executed by the CPU, the sequence of steps is described in a simplified form as follows:

(a) CPU loads the RNI program address register with software address, and waits for RNI to fetch and parse.

(b) RNI initiates an instruction fetch memory and waits for the memory cycle to complete.

(c) RNI parses the first four bytes of the instruction in sequence.

(d) RNI initiates a second instruction fetch memory cycle and waits for the memory cycle to complete.

TABLE 4-1

RNI PARSING TABLE FOR BDZ (OPCODE F)

Instruction Byte Number	Item	RNI Register	Bit Length	Use by the CPU
1	Opcode F	OP 1	4	Primary Map Branch
	R1	R1	4	Address Computation
2	DSP1	DSP1	8	Address Computation
3	KK	KK	2	Address Computation
				CPU Branch Condition
				CPU Data Rotate
	KK01	OP 4	4	Primary Map Branch
	R2	R2	4	Address Computation
4	DSP2	DSP2	8	Address Computation
5	7	OP 2	4	Secondary Map Branch
	CC	CC	2	CPU Branch Condition
5 / 6	Branch Address	Label	10	CPU Data Source to Determine Software Branch Address

(e) RNI parses the last two bytes of the instruction.

(f) RNI loads the second rank instruction register and sets the instruction ready status to the CPU. At this point RNI begins to parse the next instruction in a background mode.

(g) The CPU does a primary and secondary map branch to get to the firmware routine for processing the BCZ instruction.

(h) The CPU does an address computation and memory read of the first data value.

(i) The CPU then does testing for attach and frame error, and address computation for the second value. It then initiates a memory read for the second data value.

(j) The CPU then rotates the first data value and inputs it into the ALU.

(k) The CPU rotates the second data value and subtracts it from the first data value, which is in the ALU. At the same time it starts a memory write cycle to store the result. The value to be stored is rotated on the next firmware cycle so that it is written by the memory cycle.

(l) The CPU updates the condition flags.

(m) The CPU then tests for a branch according to CC (10 in this case). If the condition is met, the CPU inputs the branch value from the RNI label register, and loads it into the RNI program address register. The cycle then starts over again. If the condition is not met, the CPU issues a next command to RNI. By this time RNI is most likely through parsing the next instruction, so it immediately loads the second rank instruction register with the next instruction and the CPU can start on it immediately.

BUS AND MEMORY CONTROL FUNCTIONS

The bus control (BC) will provide the necessary timing information transfers and a selection process to enable synchronous communication between the ALU and RNI sections of the CPU, memory unit, and I/O control units. The ALU, RNI and memory unit will have direct lines to affect their protocol, whereas the I/O control units will use the I/O interface bus protocol.

The implementation of the bus controller provides the following capabilities:

(a) A prioritized selection process by which a unit gains access to the I/O control units needing interrupt service.

(b) The necessary control lines to affect a synchronous transfer of information on the I/O interface bus.

The memory write control (MWC) provides the necessary control and data transfer hardware for the CPU to read and write to memory. It includes the following commands:

(a) CPU memory command decoding
(b) Hardware address register selection
(c) Data word length selection
(d) Hardware address registers
(e) Bus control interface

BUS CONTROL FUNCTIONAL DESCRIPTION

The BC consists of three functionally separate units:

Bus request resolution unit
Interrupt control unit
Data transfer unit

The bus request resolution unit affects the selection process by which an I/O control unit gains access to the I/O interface bus.

The interrupt control unit enables a selection process so that one of 16 I/O control units raising an interrupt can be recognized. The data transfer unit resolves system priority contention for the bus and generates the necessary control signals to affect a data transfer (or memory access). System priority involves the following type accesses:

Memory refresh
I/O transfers
RNI accesses
CPU accesses

BUS CONTROL OPERATIONAL DESCRIPTION

This section describes the detailed signal sequence to permit an orderly flow of information between all units in the system. Refer to Figure 4-6 for an overview of the bus and memory control.

BUS REQUEST RESOLUTION UNIT

The bus request resolution unit provides the necessary hardware to accommodate a mechanism that selects one from many I/O control units (16 maximum) needed access to the I/O interface bus for a data transfer. See Figure 4-7 for the block of the bus request resolution unit.

To initiate a bus request, I/O control units will activate their assigned bus request line. The BC, in turn, will encode these lines via a priority encoder and assert the results onto the bus address lines BSAD2 and BSAD3. In addition, the BC will begin a binary count sequence (starting at count zero) seen on the bus address lines BSAD0 and BSAD1 for I/O control units to decode.

All I/O control units in question compare the bus address lines against their strapped priority level. The highest priority unit sees a match and in turn turn will activate the bus sync line (BSYN), informing the BC that priority contention has been resolved. Upon receiving BSYN, the BC will hold the bus address lines at this priority level until the BSYN is deactivated.

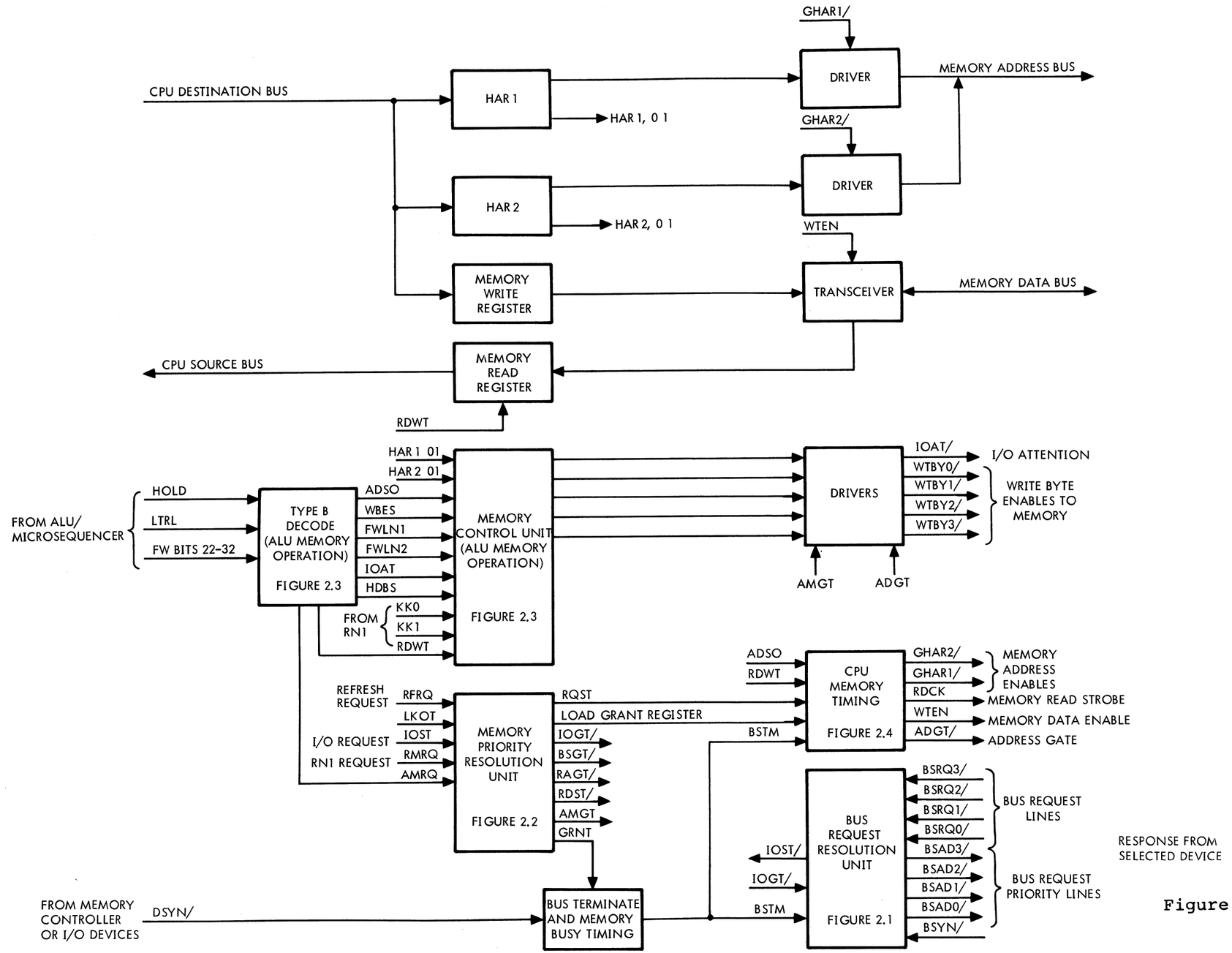


Figure 4-6. Bus and Memory Control Overview

Burst Mode

1. I/O Control Units

An I/O control unit requiring more than one data transfer (maximum of four) may do so by keeping the BYSN signal active. The BC samples the BSYN at the end of a data transfer. If active, the BSGT signal will be deactivated. This sequence may be interrupted by the memory unit, thus requiring a refresh cycle.

> NOTE: The BC will not prevent a burst mode of more than four data transfers. Constraints on unit design will be imposed to prevent more than four data transfers.

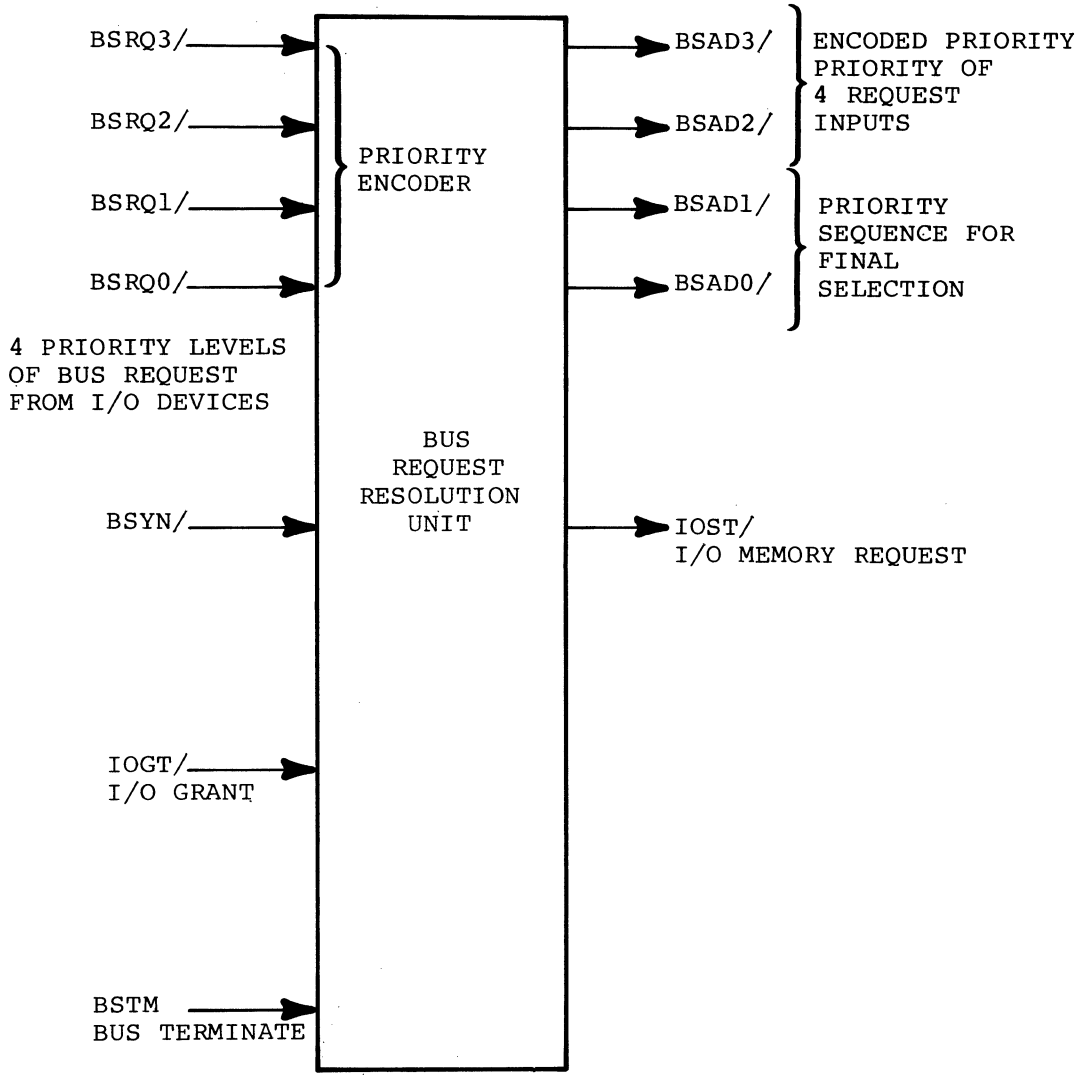


Figure 4-7. Bus and Memory Control Block Diagram

2. ALU

The ALU issues data transfer commands via the type-B firmware word. In the event that more than one transfer is required (such as read/modify write), a hold bus command is issued (via firmware). This command will lock out the I/O control units until reset by firmware.

Interrupt Control Unit

The interrupt control unit will provide the necessary hardware to resolve more than one interrupt (16 maximum) occurring at any given time. The scheme utilized for resolving priority contention is the same as the sequence described for gaining access to the I/O interface bus.

DATA TRANSFER UNIT

I/O interface bus priority contention for the I/O control units are resolved by the bus request resolution unit, yet the BC must provide a mechanism for resolving bus contention between the ALU, RNI, memory unit (for refresh) and the I/O control unit in question. The data transfer unit realizes an implementation for determining which unit will be granted I/O interface bus access.

Once a unit has been granted I/O interface bus access, the data transfer cycle must be complete before a unit of higher priority is granted I/O interface bus access. The same applies when a unit begins a burst mode cycle except that the memory unit may steal a data transfer cycle for a refresh. Figure 4-8 depicts the memory priority resolution unit.

The priority structure for the units in question is as follows:

Highest: Memory unit refresh
I/O control unit
RNI

Lowest: ALU

A data transfer (or memory cycle) on the I/O interface bus is initiated once priority has been resolved. The highest priority unit in question will be issued a grant signal; for instance, a BSGT will be issued to the I/C control units or a RFGT will be issued to the memory control unit.

The BC will generate the necessary control signals to provide for data transfer to the RNI and ALU as specified in the I/O interface product specification. Refer to Figure 4-9 for the bus and memory control block diagram elements.

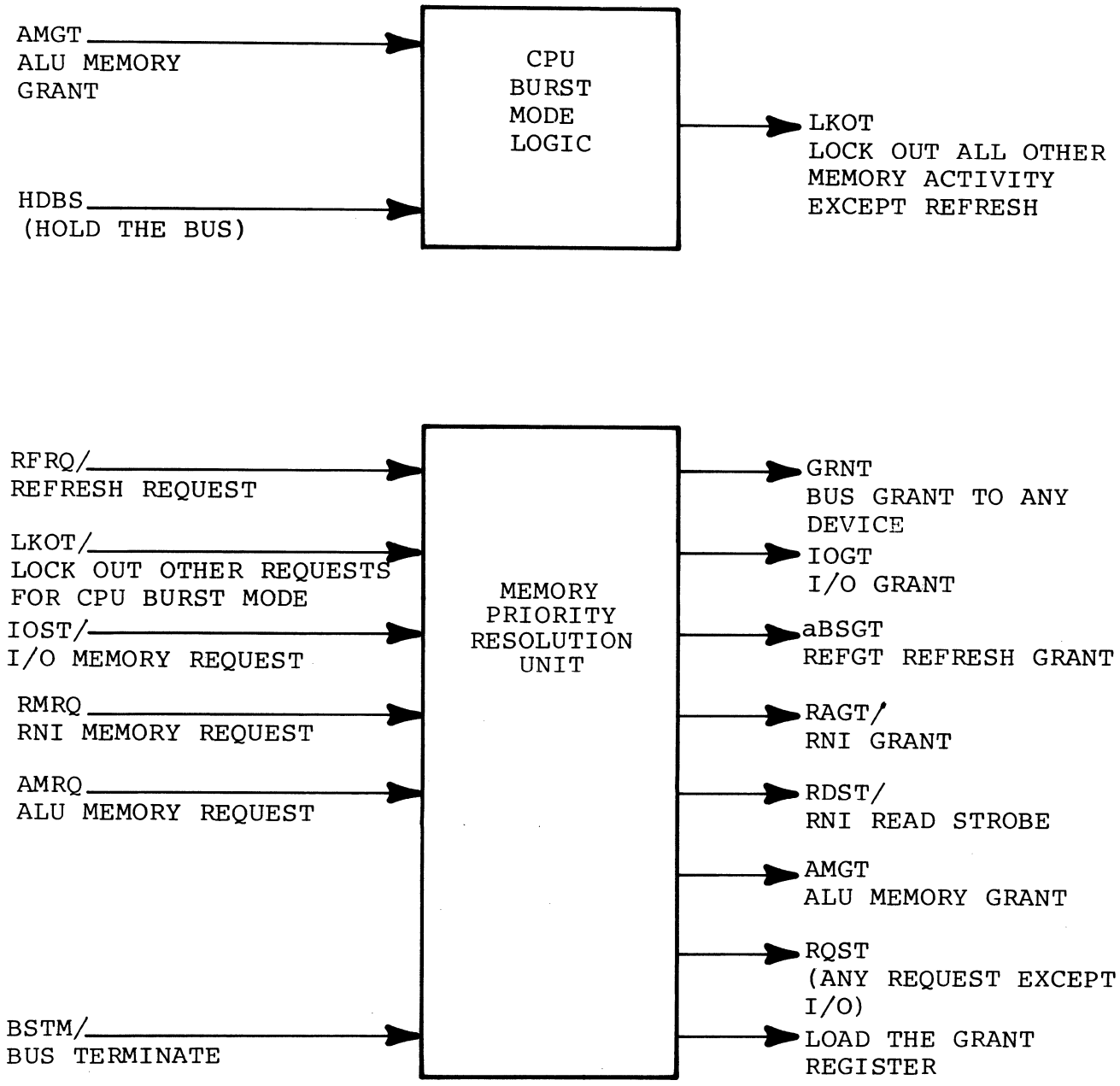


Figure 4-8. Bus and Memory Control Block Diagram Elements

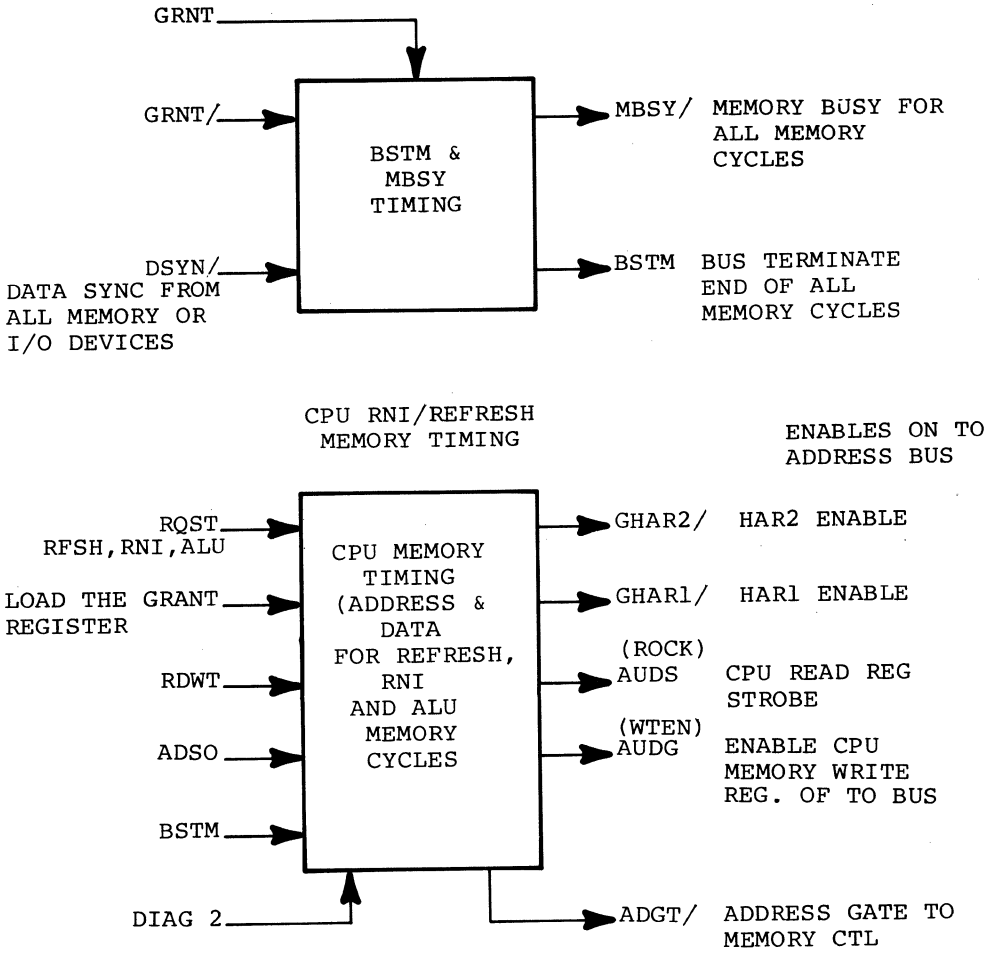


Figure 4-9. Bus and Memory Control Block Diagram Elements

A data transfer from the ALU is initiated via firmware (see Figure 4-10 for the memory control unit section). The ALU has two memory address registers, HAR1, and HAR2. Selection of these registers is also done by firmware. Thus, when the BC receives a data transfer request from the ALU, it will assert the selected HAR(X) register as a memory address source.

In addition, the BC will assert data onto the I/O interface bus data lines from the memory write register or load data from the data lines into the memory read register as commanded by firmware. To facilitate an ALU burst mode capability, firmware will issue a hold bus signal (HDBS). When received, the BC will allow only the ALU or the memory unit to gain access to the I/O interface bus. Figure 4-11 is a diagram of the CPU and memory timing control.

A data transfer from the RNI is initiated by the RNI memory request (RMRQ) signal. The BC will acknowledge by issuing a

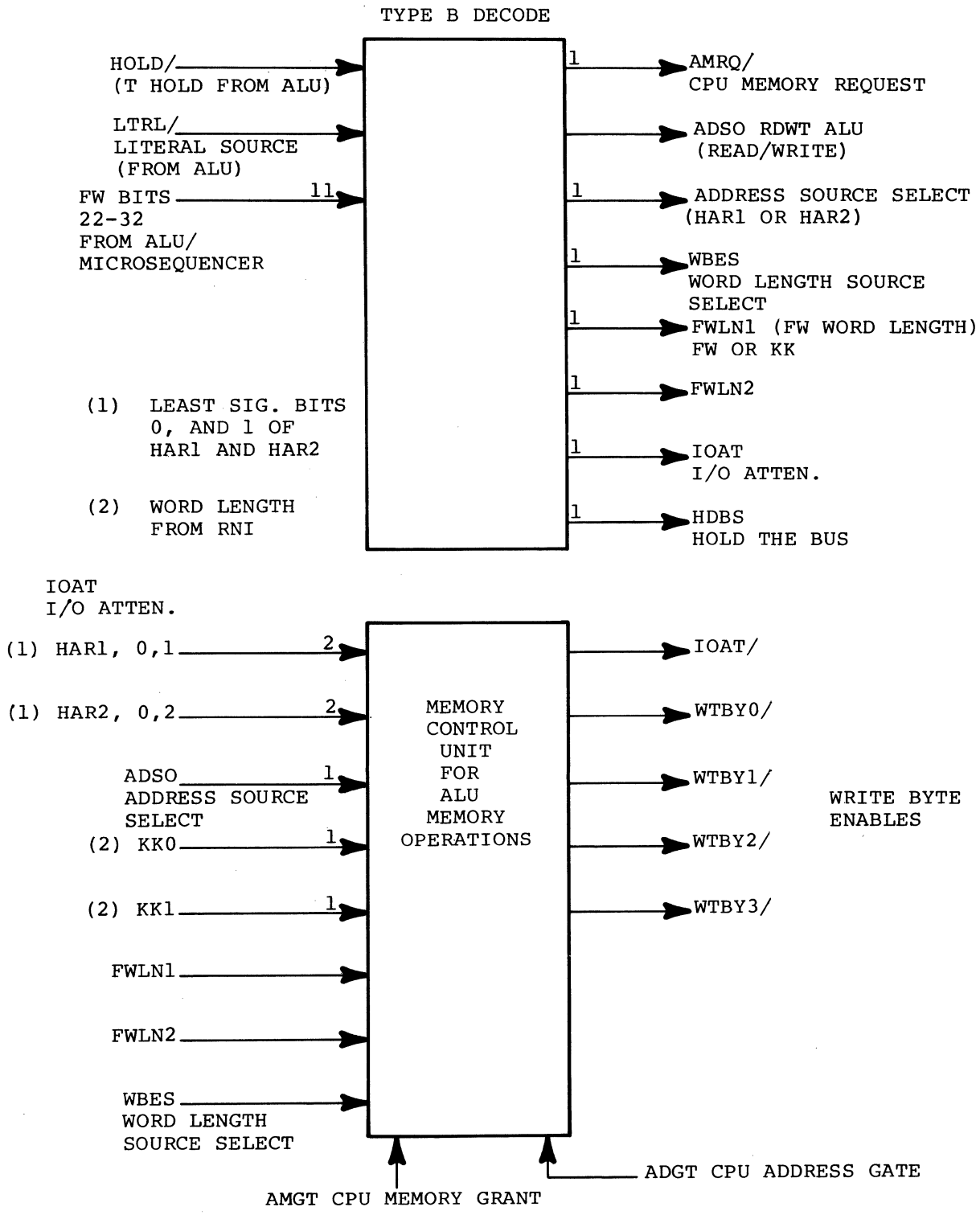


Figure 4-10. Bus and Memory Control Block Diagram Elements

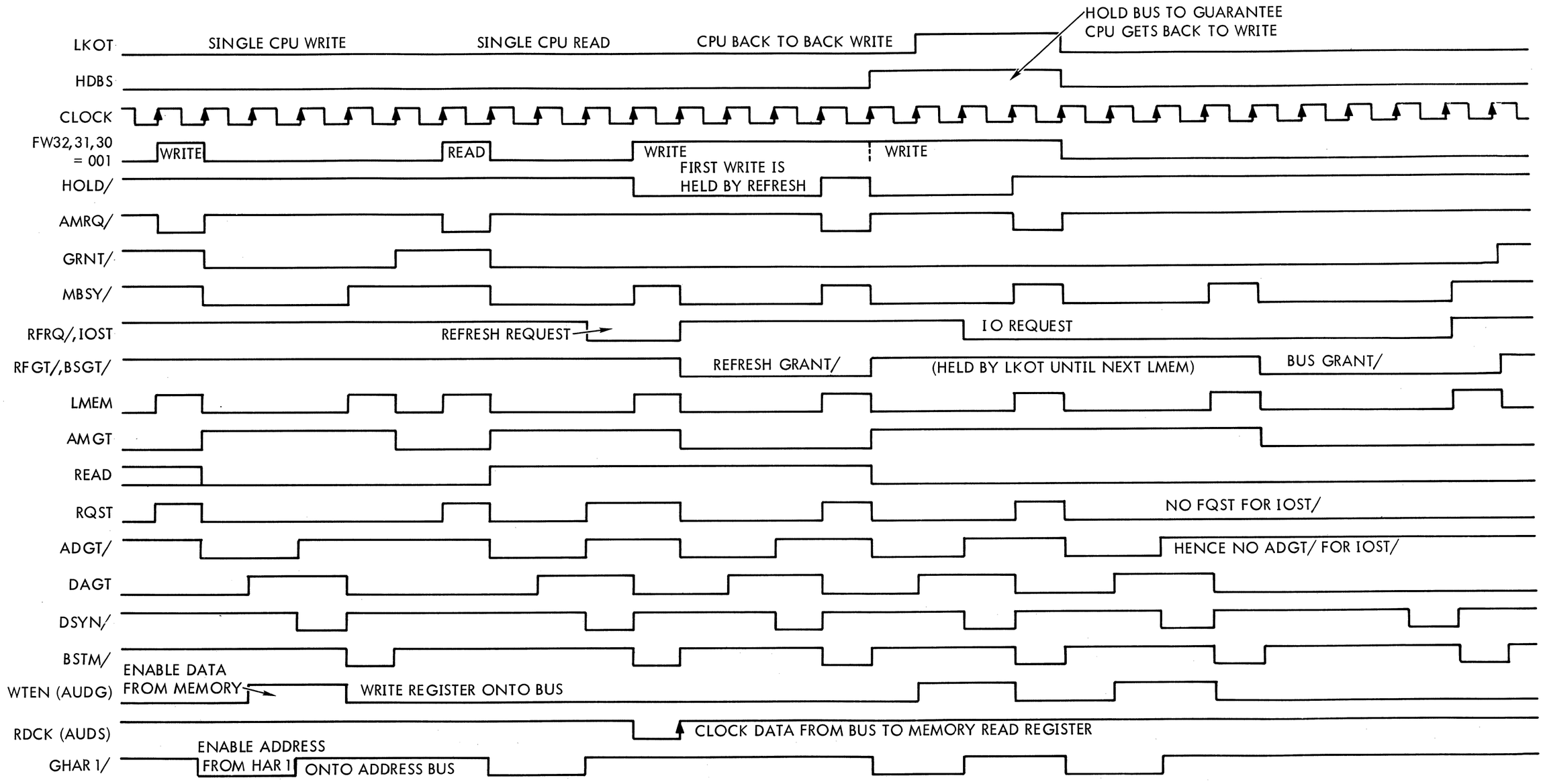


Figure 4-11. CPU and Memory Timing Control Diagram

(RAGT) signal. The RNI will use this signal to assert a memory address and reset the RMRQ signal. The BC will also issue a RNI data strobe (RDST) signal that will be used by the RNI to read data off the memory data bus.

The address gate (ADGT) signal is always asserted for transfers on the I/O interface bus by the BC. The memory unit always monitors this signal so that it may take appropriate action. When a direct transfer to an I/O control is required, the I/O attention (IOAT) is asserted. When the memory unit sees the IOAT signal true, it ignores the data transfer.

> NOTE: The data transfer is the same to an I/O control unit as to the memory unit.

The I/O control unit will strobe data off the data lines when WTBYO is true or place data on the data lines when WTBYO is false. In either case, the I/O control unit will issue a data sync (DSYN) to mark the end of the transfer.

MEMORY CONTROL UNIT

The memory control unit will generate the necessary control signals decoded from the type-B firmware word to affect a memory access cycle.

Upon deciding the type-B select bits as a memory request, the memory control unit will inform the BC to begin contention for the I/O interface bus. Once contention has been resolved, the BC will acknowledge the memory control unit, thus beginning a memory access cycle.

The source for memory address is found in the type-B firmware word (address source). The BC will route the address line drive enable signal to the HAR(X) according to the address source firmware bits. In addition, the memory control unit generates the write byte commands to the memory unit as instructed by firmware.

CPU GATED CLOCKS

System clock is gated in critical areas of the CPU. This assures that the CPU will not access system memory until priority is granted. Refer to Figure 4-12, start-up and T-hold timing diagram, for an illustration of the gated clock control.

CPU FIRMWARE WORD

The firmware that controls the central processor unit operation is physically located on the ALU board. The memory size is 64 bits wide by 8192 bits deep. Addressing for the firmware proms is provided by the microsequencer operation discussed in detail later in this section. The output of the FW proms is

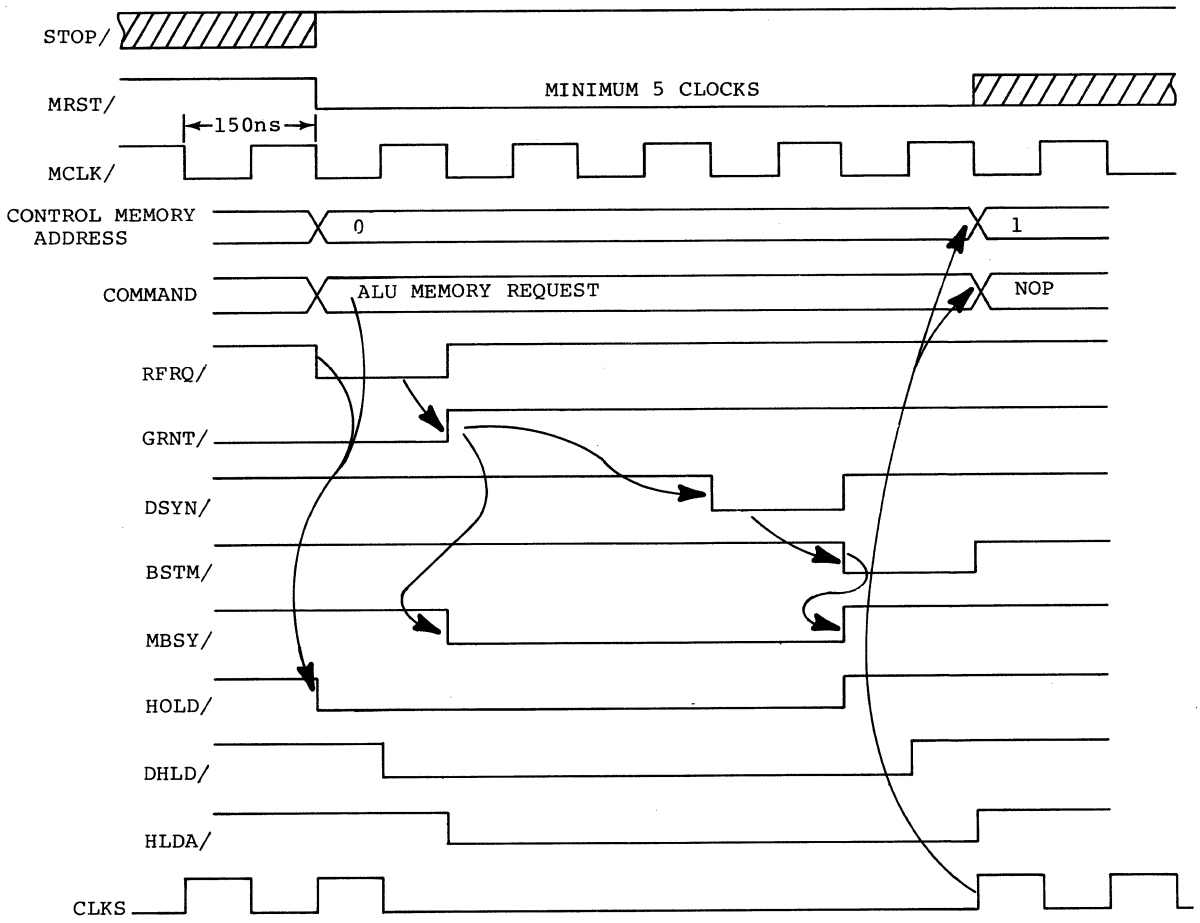


Figure 4-12. Start-Up and T-Hold Timing Diagram

latched into the command memory. A simplified method of relating to accessing the FW memory is that the word is "fetched" on one clock cycle and executed on the next.

GENERAL INSTRUCTION FORMAT

Each firmware instruction is organized into three basic fields: A field; B field; and microsequencer field (refer to Figure 4-13).

A Field (Bits 64-33)

The "A field" consists of 31 bits that are used to implement one of the three "A type" instructions:

ALU	These instructions direct the arithmetic/logical operations performed by the ALU on selectable external (to the microprocessor) and/or internal 32-bit words.
Rotate	These instructions direct the rotate operations performed by the rotate circuits on 32-bit words from memory or from the microprocessor.

Addr. Comp. These instructions direct the address computation operations performed by the address computation circuits on address data from the microprocessor and RNI instruction registers.

> NOTE: There is an exceptional case when an ALU instruction carries an embedded literal. In this instance all 64 bits of the instruction register are used for this single "A type" operation.

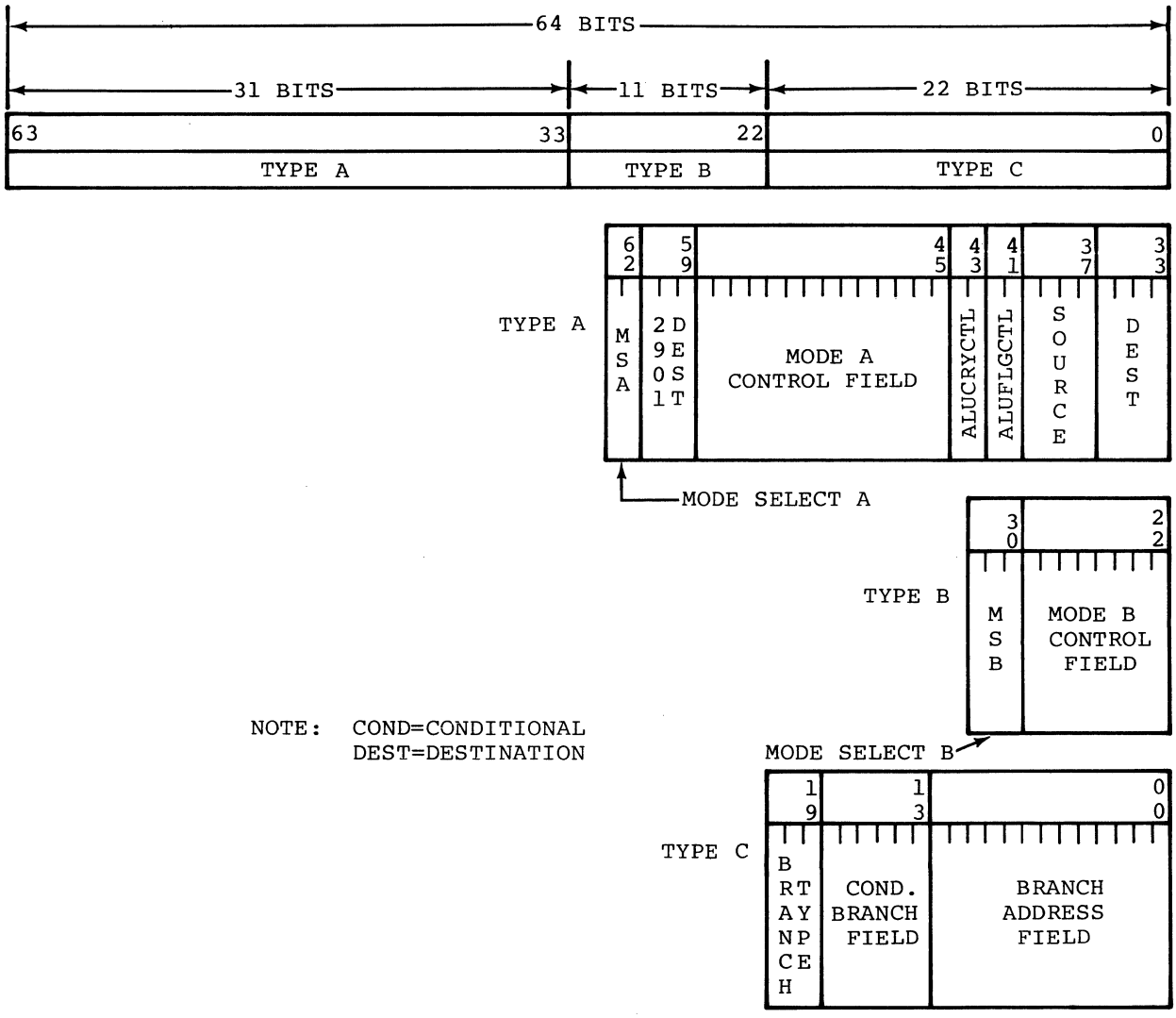


Figure 4-13. The CPU Firmware Word

B Field (Bits 32-22)

The "B field" consists of 11 bits that are used to implement one of seven "B type" instructions:

Shift	These instructions support an "A type" operation calling for the shifting one bit right or one bit left of a 32-bit value being loaded into the register file or Q register. The B-type shift instruction controls the input-disposition of the bits that are shifted into the word when the A-commanded shift is executed.
Memory Read/Write	These instructions cause four bytes to be received from (read) or sent to (write) main memory or a peripheral device.
Multiply	This instruction provides a multiplication "utility." It is used in conjunction with A-type ALU operations.
Divide	This instruction provides a division "utility" in a similar manner to the multiply described just previously.
Character Test	This instruction is used in conjunction with A-type ALU operations (logical) for a process that results in a yes/no determination of a "match" between a predetermined character(s) and a character string being tested.
Flags Control	These instructions deal with the setting, resetting, and/or testing of various system flags.
CPU Control	These instructions provide the facility for CPU-RNI communication.

Microsequencer Field (Bits 21-0)

The microsequencer field consists of 22 bits that are used to implement the ordering of the sequence of firmware instructions execution, i.e., program control. This field will be the various branching commands (unconditional, conditional, etc.) that determine the next firmware instruction that will be executed.

Simultaneous Use of A, B, and C Fields

In many cases, A, B, and C fields may be used simultaneously. There are some exceptions to the free use of simultaneous fields.

Some exceptions are due to conflict, and some just do not make sense. The main restriction occurs when the "A type" ALU operation has a literal as a source. In this case, the lower 32 bits are used as the literal. In this case, no "B type" is executed, and the "C type" is replaced with an automatic microsequencer advance.

Simultaneous uses of the fields are shown in the multiply/ divide examples provided in the "B-Type Operations" paragraphs of this section. Refer to the latter part of this section for a list of SEQUEL firmware restrictions, which provide a breakdown of compatible type A, B and C fields.

FIRMWARE TABLES

Specific use of the individual firmware bits is provided in Tables 4-2 through Tables 4-31. In most cases for the "A-type" fields, the bits are function dependent. Should any doubt arise as to the bits assignments and the specific "A-type" field, refer to Figure 4-14, which presents the composite firmware word.

A-TYPE OPERATIONS

There are three A-type operations: ALU, rotate and address computation. Refer to Figure 4-14 for an overview of the three A-type functions. The individual bit assignments are defined with this illustration.

ALU OPERATIONS

The ALU is a high-speed, "number-crunching" device using a 2901 bit slice that performs one of eight commandable arithmetic/ logical operations on two 32-bit inputs (referred to as the "R" and "S" inputs). It yields a single 32-bit result (referred to as the "F" output), plus operation status bits. The ALU is depicted in Figure 4-15.

Each 2901 block represents a four bit bipolar microprocessor slice consisting of a 16-word by 4-bit two port RAM, a high speed ALU, and the associated shifting, decoding and multiplexing circuitry. There are nine bits provided for the micro instruction word. They are used to select the ALU source operands, the ALU function, and the ALU destination register. In this application, the ALU is cascaded with seven additional arithmetic operations. A carry look-ahead capability is provided by the 2902 integrated circuits. This allows the ALU to complete the typical add or subtract function within a 150 nanosecond clock period.

An illustration of the 2901 microprocessor is provided in Figure 4-16. Detailed application information for the 2901 device is provided in Figure 4-17. These figures should provide the

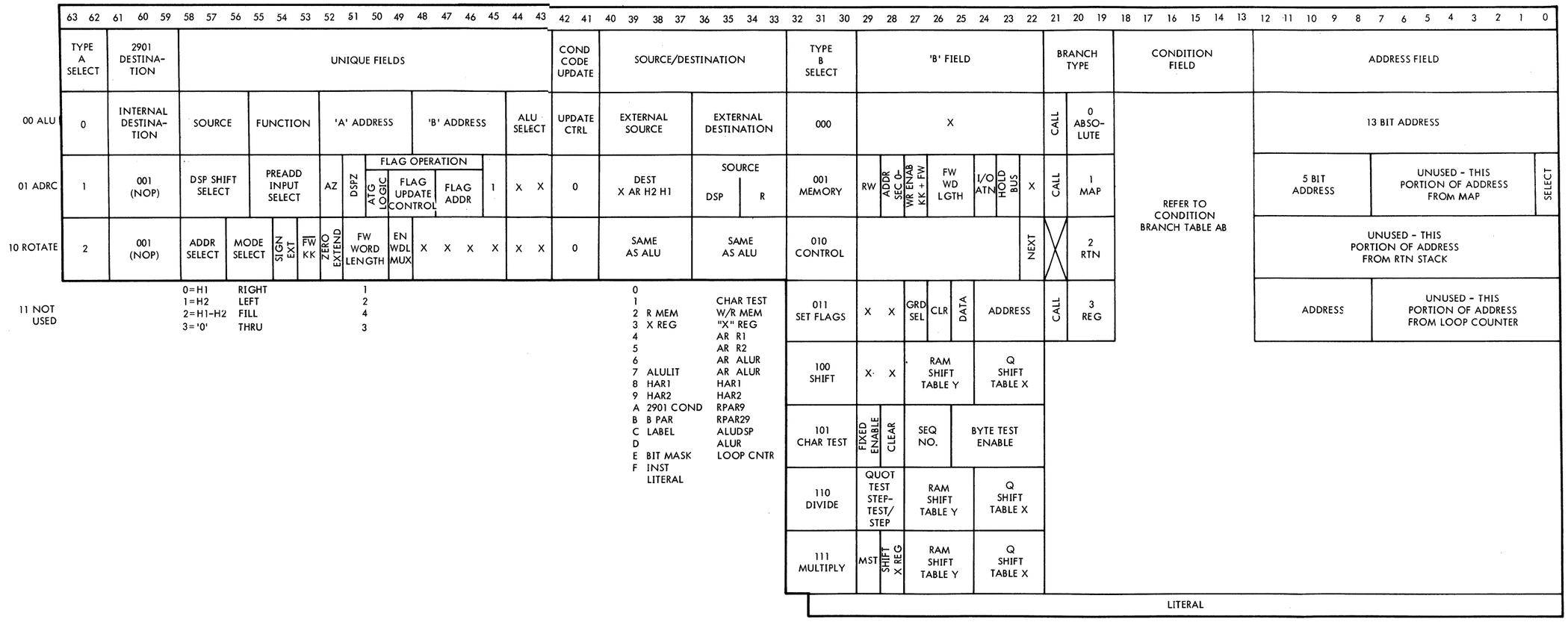
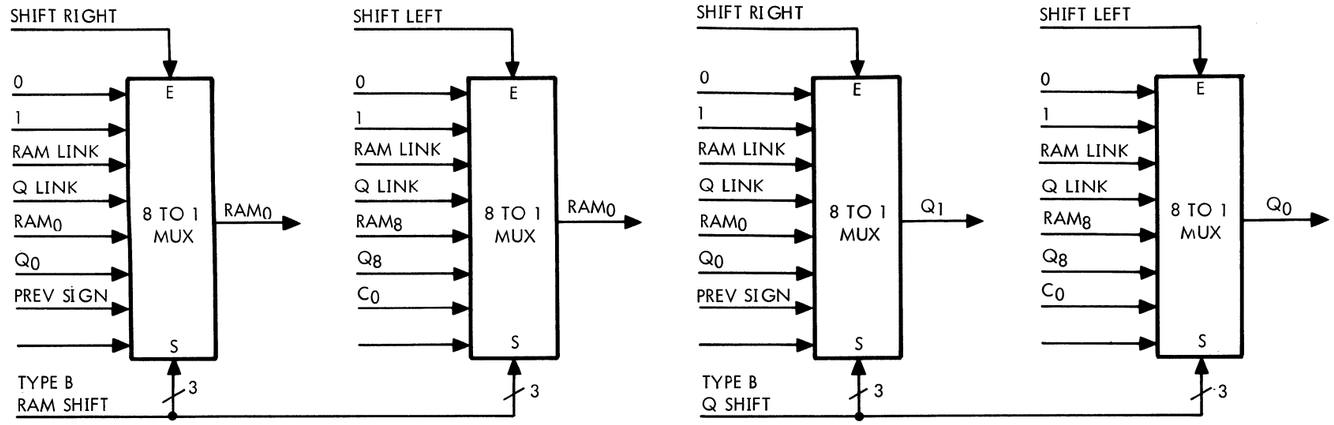
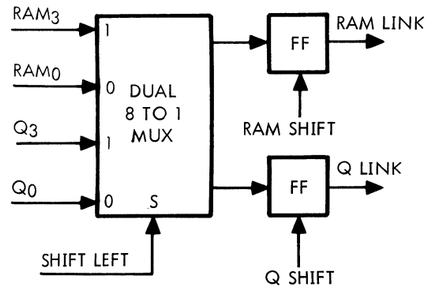


Figure 4-14. Composite Firmware Word





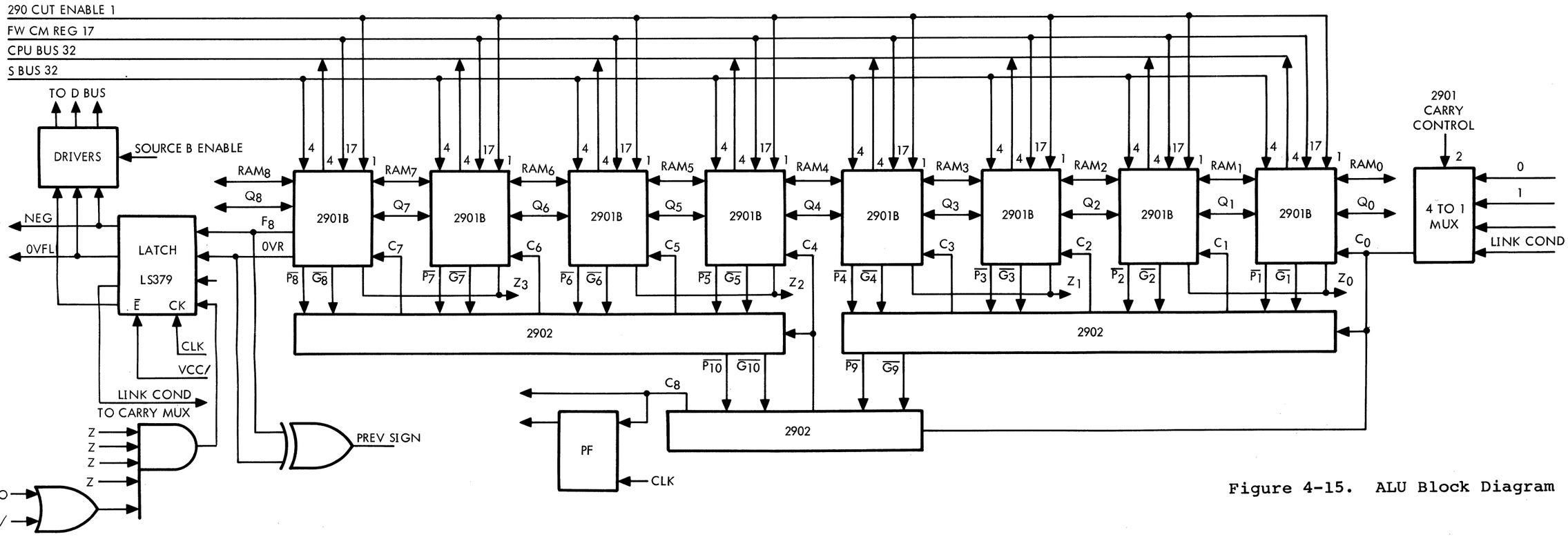


Figure 4-15. ALU Block Diagram

information required to better understand the 2901 as a separate entity. Other information on the 2901 and presented in this document include the following:

Figure 4-18. ALU Destination Control
Figure 4-19. Source Operand and ALU Function Matrix
Figure 4-20. ALU Source Operand Control
Figure 4-21. ALU Function Control

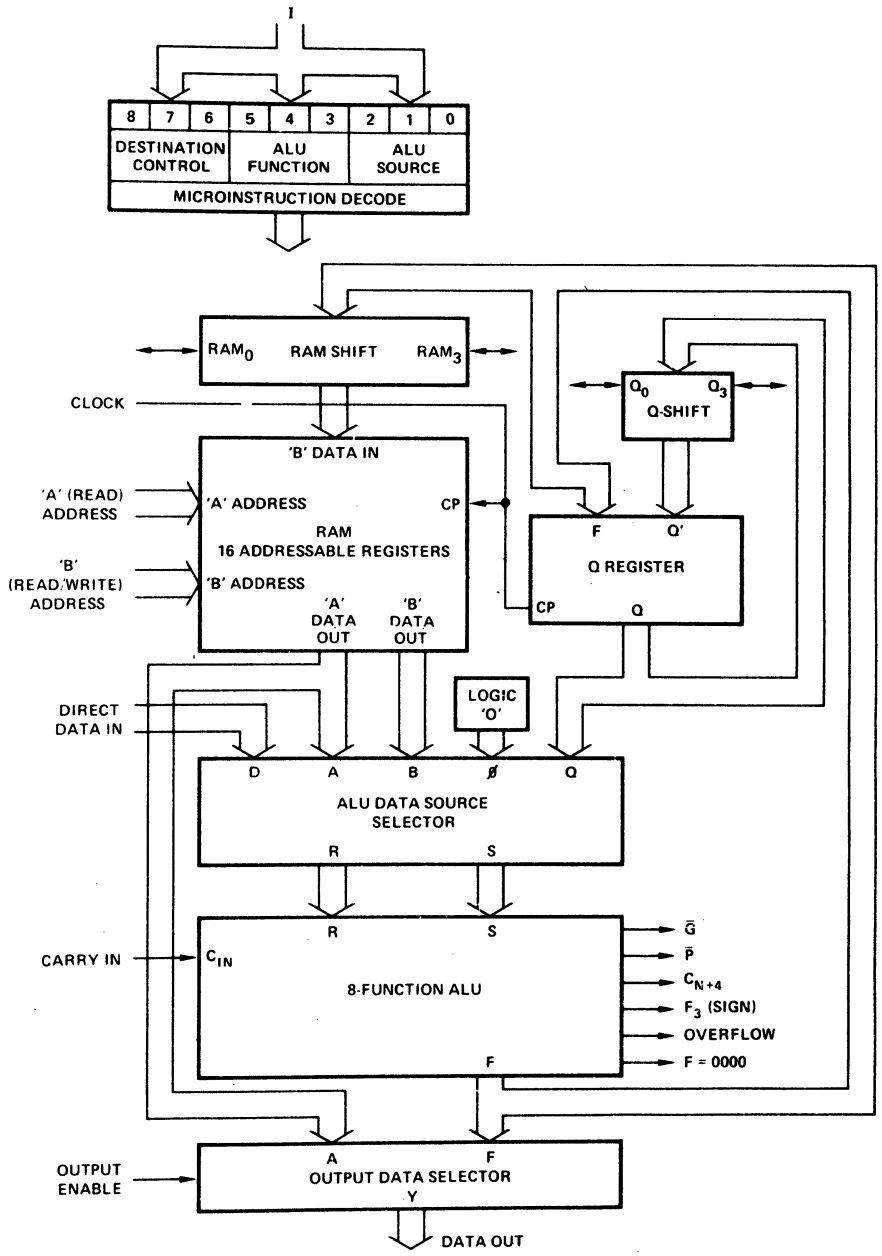
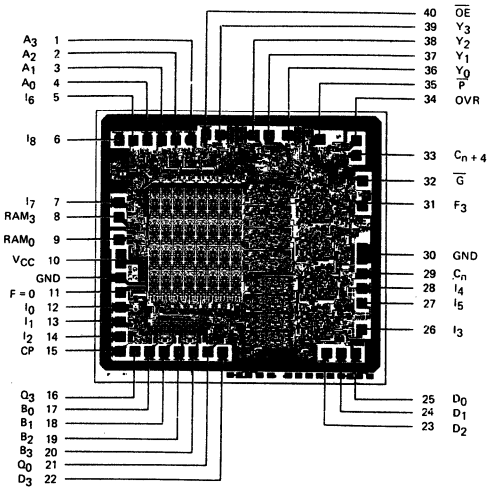


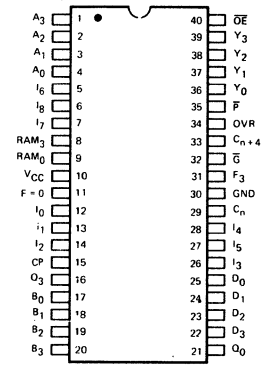
Figure 4-16. Microprocessor Slice Block Diagram

METALLIZATION AND PAD LAYOUT



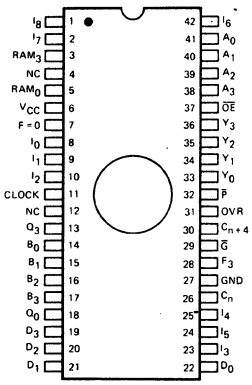
CONNECTION DIAGRAM
Top View

DIP



MPR-006

FLAT PACK CONNECTIONS
Top View



MPR-007

Note: Pin 1 is marked for orientation.

PIN DEFINITIONS

A_{0-3} The four address inputs to the register stack used to select one register whose contents are displayed through the A-port.

B_{0-3} The four address inputs to the register stack used to select one register whose contents are displayed through the B-port and into which new data can be written when the clock goes LOW.

I_{0-8} The nine instruction control lines. Used to determine what data sources will be applied to the ALU (I_{012}), what function the ALU will perform (I_{345}), and what data is to be deposited in the Q-register or the register stack (I_{678}).

Q_3 RAM_3 A shift line at the MSB of the Q register (Q_3) and the register stack (RAM_3). Electrically these lines are three-state outputs connected to TTL inputs internal to the device. When the destination code on I_{678} indicates an up shift (octal 6 or 7) the three-state outputs are enabled and the MSB of the Q register is available on the Q_3 pin and the MSB of the ALU output is available on the RAM_3 pin. Otherwise, the three-state outputs are OFF (high-impedance) and the pins are electrically LS-TTL inputs. When the destination code calls for a down shift, the pins are used as the data inputs to the MSB of the Q register (octal 4) and RAM (octal 4 or 5).

Q_0 RAM_0 Shift lines like Q_3 and RAM_3, but at the LSB of the Q-register and RAM. These pins are tied to the Q_3 and RAM_3 pins of the adjacent device to transfer data between devices for up and down shifts of the Q register and ALU data.

D_{0-3} Direct data inputs. A four-bit data field which may be selected as one of the ALU data sources for entering data into the device. D_0 is the LSB.

Y_{0-3} The four data outputs. These are three-state output lines. When enabled, they display either the four outputs of the ALU or the data on the A-port of the register stack, as determined by the destination code I_{678}.

$\overline{OE}$ Output Enable. When $\overline{OE}$ is HIGH, the Y outputs are OFF; when $\overline{OE}$ is LOW, the Y outputs are active (HIGH or LOW).

$\overline{G}$, $\overline{P}$ The carry generate and propagate outputs of the internal ALU. These signals are used with the Am2902 for carry-lookahead.

OVR Overflow. This pin is logically the Exclusive-OR of the carry-in and carry-out of the MSB of the ALU. At the most significant end of the word, this pin indicates that the result of an arithmetic two's complement operation has overflowed into the sign-bit.

F = 0 This is an open collector output which goes HIGH (OFF) if the data on the four ALU outputs F_{0-3} are all LOW. In positive logic, it indicates the result of an ALU operation is zero.

F_3 The most significant ALU output bit.

C_n The carry-in to the internal ALU.

C_{n+4} The carry-out of the internal ALU.

CP The clock input. The Q register and register stack outputs change on the clock LOW-to-HIGH transition. The clock LOW time is internally the write enable to the 16 x 4 RAM which compromises the "master" latches of the register stack. While the clock is LOW, the "slave" latches on the RAM outputs are closed, storing the data previously on the RAM outputs. This allows synchronous master-slave operation of the register stack.

Figure 4-17. Microprocessor Detail

Mnemonic	MICRO CODE				RAM FUNCTION		Q-REG. FUNCTION		Y OUTPUT	RAM SHIFTER		Q SHIFTER	
	I_8	I_7	I_6	Octal Code	Shift	Load	Shift	Load		RAM_0	RAM_3	Q_0	Q_3
QREG	L	L	L	0	X	NONE	NONE	F → Q	F	X	X	X	X
NOP	L	L	H	1	X	NONE	X	NONE	F	X	X	X	X
RAMA	L	H	L	2	NONE	F → B	X	NONE	A	X	X	X	X
RAMF	L	H	H	3	NONE	F → B	X	NONE	F	X	X	X	X
RAMQD	H	L	L	4	DOWN	F/2 → B	DOWN	Q/2 → Q	F	F_0	IN_3	Q_0	IN_3
RAMD	H	L	H	5	DOWN	F/2 → B	X	NONE	F	F_0	IN_3	Q_0	X
RAMQU	H	H	L	6	UP	2F → B	UP	2Q → Q	F	IN_0	F_3	IN_0	Q_3
RAMU	H	H	H	7	UP	2F → B	X	NONE	F	IN_0	F_3	X	Q_3

X = Don't care. Electrically, the shift pin is a TTL input internally connected to a three-state output which is in the high-impedance state
B = Register Addressed by B inputs.
UP is toward MSB, DOWN is toward LSB.

Figure 4-18. ALU Destination Control, FW61, FW60, FW59

OCTAL I_{210}	OCTAL I_{543} ALU Function \ ALU Source	A, Q	A, B	0, Q	0, B	0, A	D, A	D, Q	D, 0
0	C_n = L, R Plus S	A+Q	A+B	Q	B	A	D+A	D+Q	D
	C_n = H	A+Q+1	A+B+1	Q+1	B+1	A+1	D+A+1	D+Q+1	D+1
1	C_n = L, S Minus R	Q−A−1	B−A−1	Q−1	B−1	A−1	A−D−1	Q−D−1	−D−1
	C_n = H	Q−A	B−A	Q	B	A	A−D	Q−D	−D
2	C_n = L, R Minus S	A−Q−1	A−B−1	−Q−1	−B−1	−A−1	D−A−1	D−Q−1	D−1
	C_n = H	A−Q	A−B	−Q	−B	−A	D−A	D−Q	D
3	R OR S	A ∨ Q	A ∨ B	Q	B	A	D ∨ A	D ∨ Q	D
4	R AND S	A ∧ Q	A ∧ B	0	0	0	D ∧ A	D ∧ Q	0
5	$\overline{R}$ AND S	$\overline{A}$ ∧ Q	A ∧ B	Q	B	A	$\overline{D}$ ∧ A	$\overline{D}$ ∧ Q	0
6	R EX-OR S	A ∀ Q	A ∀ B	Q	B	A	D ∀ A	D ∀ Q	D
7	R EX-NOR S	$\overline{A \forall Q}$	$\overline{A \forall B}$	$\overline{Q}$	$\overline{B}$	$\overline{A}$	$\overline{D \forall A}$	$\overline{D \forall Q}$	$\overline{D}$

+ Plus, − Minus, ∨ = OR, ∧ = AND, ∀ = EX-OR

Figure 4-19. Source Operand and ALU Function Matrix, FW58, FW57, FW56

	MICRO CODE				ALU SOURCE OPERANDS	
Mnemonic	I_2	I_1	I_0	Octal Code	R	S
AQ	L	L	L	0	A	Q
AB	L	L	H	1	A	B
ZQ	L	H	L	2	0	Q
ZB	L	H	H	3	0	B
ZA	H	L	L	4	0	A
DA	H	L	H	5	D	A
DQ	H	H	L	6	D	Q
DZ	H	H	H	7	D	0

Figure 4-20. ALU Source Operand Control, FW58, FW57, FW56

	MICRO CODE				ALU	SYMBOL
Mnemonic	I_5	I_4	I_3	Octal Code	Function	
ADD	L	L	L	0	R Plus S	R + S
SUBR	L	L	H	1	S Minus R	S − R
SUBS	L	H	L	2	R Minus S	R − S
OR	L	H	H	3	R OR S	$R \vee S$
AND	H	L	L	4	R AND S	$R \wedge S$
NOTRS	H	L	H	5	$\overline{R}$ AND S	$\overline{R} \wedge S$
EXOR	H	H	L	6	R EX-OR S	$R \forall S$
EXNOR	H	H	H	7	R EX-NOR S	$\overline{R \forall S}$

Figure 4-21. ALU Function Control, FW55, FW54, FW53

TABLE 4-2

EXTERNAL DESTINATION CONTROL

Bits					No. of Bits Used
36	35	34	33	External Destination	(Right Justified)
0	0	0	0	No Destination	32
0	0	0	1	Character Test Mask	8
0	0	1	0	Memory Write Register	32
0	0	1	1	"X" Register	32
0	1	0	0	Address Register File (R1)	24
0	1	0	1	Address Register File (R2)	24
0	1	1	0	Address Register File (ALUR)	24
0	1	1	1	Address Register File (ALUR)	24
1	0	0	0	Hardware Address Register 1	24
1	0	0	1	Hardware Address Register 2	24
1	0	1	0	Program Address Register (9)	9
1	0	1	1	Program Address Register (24)	24
1	1	0	0	ALU Displacement	8
1	1	0	1	ALU R (RC)	4
1	1	1	0	Loop Counter	8
1	1	1	1		

TABLE 4-3

EXTERNAL SOURCE SELECTION

Bits				Data Bus Contents	No. of Bits
40	39	38	37		
0	0	0	0	Spare	
0	0	0	0	Spare	
0	0	1	0	Memory Read Register	32
0	0	1	1	"X" Register	32
0	1	0	0	Spare	
0	1	0	1	Spare	
0	1	1	0	Spare	
0	1	1	1	ALU Literal	32
1	0	0	0	Hardware Address Register 1	24
1	0	0	1	Hardware Address Register 2	24
1	0	1	0	2901 Condition	3
1	0	1	1	Backup Program Counter (24)	24
1	1	0	0	Instruction Label	16
1	1	0	1	Spare	
1	1	1	0	Displacement Bit Mask	8
1	1	1	1	Instruction Literal	8

TABLE 4-4

ALU CONDITION CODES CONTROL

Bits		Condition Code Update
42	41	
0	0	No action
0	1	Update condition codes without linked 0 Test.
1	0	No action
1	1	Update condition codes with linked 0 Test.

TABLE 4-5

CARRY CONTROL

Bits		ALU Carry in
44	43	
0	0	Force carry bit to zero (Link and status)
0	1	Force carry bit to 1 (Link and status)
1	0	Link previous carry bit to ALU carry in
1	1	Link previous carry bit to ALU carry in

TABLE 4-6

ALU OPERATION CODES

Bits			ALU Function
55	54	53 * (INST 3)	
0	0	0	Add S Input to R Input
0	0	1	Subtract R Input from S Input
0	1	0	Subtract S Input from R Input
0	1	1	Logically OR S Input with R Input
1	0	0	Logically AND S Input with R Input
1	0	1	Logically AND S Input with Complement of R Input
1	1	0	Logically Exclusive OR S Input with R Input
1	1	1	Logically Exclusive NOR S Input with R Input

* INST3 = FW53 + Mul. FW29. XR00. $\overline{\text{EXBT}}$ + DIV. QUOT/

TABLE 4-7

ALU INPUT CONTROL

Bits			Input Selection	
58	57* (INST 1)	56	R Input	S input
0	0	0	A-addressed Register File Register	Q Register
0	0	1	A-addressed Register File Register	B-addressed Register File Register
0	1	0	Zeros	Q Register
0	1	1	Zeros	B-addressed Register File Register
1	0	0	Zeros	A-addressed Register File Register
1	0	1	Data Bus	A-addressed Register File Register
1	1	0	Data Bus	Q Register
1	1	1	Data Bus	Zeros

*INST = FW57 + MUL.FW29. $\overline{\text{X R00 + EXBT}}$

TABLE 4-8

ALU OUTPUT CONTROL

INTERNAL DESTINATION

Bits			Output Selection
61	60	59	
0	0	0	F output to Q Register and CPU Bus via Y.
0	0	1	F Output to CPU Bus via Y.
0	1	0	F output to B-addressed Register File register. A-addressed Register File register to CPU Bus via Y.
0	1	1	F output to B-addressed Register File register. F output to CPU Bus via Y.
1	0	0	F output to B-addressed Register File register via shift (shift right 1 bit). Q Register output to Q Shift (shift right 1 bit) and back to Q Register. F output to CPU Bus via Y.
1	0	1	F output to B-addressed Register File register via shift (shift right 1 bit). F output to CPU Bus via Y.
1	1	0	F output to B-addressed Register File Register via shift (shift right 1 bit). Q Register output to Q shift (shift left 1 bit) and back to Q Register. F output to CPU Bus via Y.
1	1	1	F output to B-addressed Register File Register via shift (shift Left 1 bit). F output to CPU Bus via Y.

TABLE 4-9

MEMORY ADDRESS REFERENCE (ROTATE CONTROL)

Bits		Referenced Address Source
58	57	
0	0	Hardware Address Register 1
0	1	Hardware Address Register 2
1	0	Difference of HAR1 and 2
1	1	0

TABLE 4-10

FW WORD LENGTH (ROTATE CONTROL)

Bits		Qualified Word Length
51	50	
0	0	1 Byte
0	1	2 Bytes
1	0	4 Bytes
1	1	3 Bytes

TABLE 4-11

ROTATE CONTROL

Bits		Rotation Command
56	55	
0	0	Rotate Right
0	1	Rotate Left
1	0	Fill
1	1	Straight Through (No Rotation)

TABLE 4-12

BASE ADDRESS SELECTION

BITS		
34	33	AR File Address Source
0	0	R1
0	1	R2
1	0	ALU R
1	1	ALU R

TABLE 4-13

DISPLACEMENT SELECTION

Bits		
36	35	Displacement Factor Source
0	0	DSP 1
0	1	DSP 2
1	0	ALU DSP
1	1	(Constant) 4

TABLE 4-14

DESTINATION CONTROL

Bits			
39	38	37	Memory Address Destinations
0	0	0	No Destination (Bit Bucket)
0	0	1	Hardware Address Register 1
0	1	0	Hardware Address Register 2
0	1	1	Hardware Address Register and Hardware Address Register 2
1	0	0	AR File
1	0	1	AR File and Hardware Address Register 1
1	1	0	AR File and Hardware Address Register 2
1	1	1	AR File and Hardware Address Register 1 and Hardware Address Register 2

TABLE 4-15

A/R FILE FLAG BIT SELECTED FOR UPDATE

Bits 47	46	Active Flag
0	0	0 (Core)
0	1	1 (Linked)
1	0	2 (Write Required)
1	1	Unassigned

TABLE 4-16

A/R FILE FLAGS CONTROL

Bits 49	48	Flag Control
0	0	No-op (Return flags unaltered)
0	1	Clear (Reset all flags to zero)
1	0	Set (Set specified flag to 1)
1	1	Reset (Reset specified flag to zero)

TABLE 4-17

PRE-ADDER CONTROL (ADC)

Bits		
54	53	Pre-adder Function
0	0	+0 (Std. Length Operand)
0	1	+2 (6-Byte Operand)
1	0	+4 (Character String Scan-Fwd.)
1	1	-4 (Character String Scan-Rev.)

TABLE 4-18

SHIFT CONTROL (ADC)

Bits		
57	56	Shift Command
0	0	No Shift (Operand Length = 1 byte)
0	1	R DSP
1	0	3 Bits Right (Operand Length = 1 bit)
1	1	1 Bit Left (Operand Length = 2, 4, or 6 Bytes)

TABLE 4-19

WORD LENGTH INDICATOR

MEMORY COMMAND

Bits		Operand Length
26	25	
0	0	1 byte
0	1	2 bytes
1	0	4 bytes
1	1	3 bytes

TABLE 4-20

TEST BYTE ENABLES

CHARACTER TEST

Bits				Memory Data Bytes to be Tested			
25	24	23	22	Byte 3	Byte 2	Byte 1	Byte 0
0	0	0	0	X	X	X	X
0	0	0	1	X	X	X	
0	0	1	0	X	X		X
0	0	1	1	X	X		
0	1	0	0	X		X	X
0	1	0	1	X		X	
0	1	1	0	X			X
0	1	1	1	X			
1	0	0	0		X	X	X
1	0	0	1		X	X	
1	0	1	0		X		X
1	0	1	1		X		
1	1	0	0			X	X
1	1	0	1			X	
1	1	1	0				X
1	1	1	1				

TABLE 4-21

VARIABLE TEST

CHARACTER TEST

Bits		Variable to be Used in Searching for a Match
27	26	
0	0	None
0	1	SC0
1	0	SC1
1	1	SC2

TABLE 4-22

X REGISTER CONTROL (MULTIPLY)

Bits		X Register Control
29	28	
0	0	Clear (i.e., reset to 0). Extension Bit Latch (only).
0	1	Clear (i.e., reset to 0). Extension Bit Latch (only).
1	0	Reference Bits 55-53 (A-Type ALU command) and 32-30 to modify ALU command as appropriate (Multiply Step).
1	1	Reference Bits 55-53 (A-Type ALU command) and 32-30 to modify ALU command as appropriate (Multiply step) and shift X Register contents 1 bit to the right filling the Extension Bit Latch with the LSB.

TABLE 4-23

DIVIDE COMMANDS

Bits 29 28	Command Name	Command Function
0 0	Divide Quotient	This operation is used to insert a bit into the quotient register. The accompanying ALU operation is an ADD of the quotient (working register and zero).
0 1	Divide Test	The hardware flags, divided sign, and remainder sign are updated by the result of the ALU operation which is specified as an OR of the most significant dividend (working) register and zero.
1 0	Divide Step	The ALU instruction is modified as determined by the sign bit of the XREG (divisor sign) and the remainder sign. The specified ALU operation is an ADD with either linked carry or 0 carry. The operands are either: (1) A dividend (working) register and a divisor (working) register; OR (2) A quotient (working) register and zero.
1 1	Divide Test Step	Identical to a DIVIDE STEP, but additionally, the sign of the result is used to update the remainder sign kept by the hardware for later divide step use.

TABLE 4-24

FLAGS CONTROL

Bits			Flag Register 1 Flag Selected (Bit 27 = 0)		Flag Register 2 Flag S (Bit 27 = 1)
24	23	22			
0	0	0	Flag 0	FLG 0	Flag 4
0	0	1	Flag 1	FLG 1	Flag 5
0	1	0	Flag 2	FLG 2	Flag 6
0	1	1	Flag 3	FLG 3	Flag 7
1	0	0	Diagnostic	DIAG 1	Flag 8
1	0	1	Up/Down Scan	UDSN	Flag 9
1	1	0	ADDRESS 0	ADR 0	Flag 10
1	1	1	Diagnostic	DIAG 2	Flag 11

TABLE 4-25

REGISTER SHIFT CONTROLS

Q Register Right Shift (LSB shifted out, MSB inserted) Input to the MSB	Bits 24	Bits 23
ZERO	0	0
ONE	0	0
The bit shifted out on the previous shift (left or right) of a Register File input (i.e. linked shift)	0	1
The bit shifted out in the previous shift (left or right) of a Q Register input (i.e. linked shift)	0	1
The bit shifted out (The LSB) in this right shift of the Register File input (i.e, cycle-shift with the Register File input)	1	0
The bit shifted out (The LSB) on this right shift of this Q Register input (i.e, end-around cycle shift)	1	0
Selected carry in bit	1	1
Unassigned	1	1

TABLE 4-25

Q REGISTER SHIFT CONTROLS (CON'T)

Q Register Left Shift (MSB shifted out, LSB inserted) Input to the LSB	Bits 24	23	22
ZERO	0	0	0
ONE	0	0	1
The bit shifted out on the previous shift (left or right) of a Register File input (i.e., linked shift)	0	1	0
The bit shifted out on the previous shift (left or right) of a Q Register input, (i.e., linked shift)	0	1	1
The bit shifted out (The MSB) on this left shift of the Register File input (i.e., cyclic shift with the Register File input)	1	0	0
The bit shifted out (The MSB) on this left shift of this Q Register input (i.e., end-around cyclic shift)	1	0	1
Selected carry in bit	1	1	0
Unassigned	1	1	1

TABLE 4-26

REGISTER FILE SHIFT CONTROLS

Bits 27	26	25	Q Register File Right Shift (LSB shifted out, MSB inserted) Input to the MSB
0	0	0	Zero
0	0	1	One
0	1	0	The bit shifted out on the previous shift (left or right) of a Register File input (i.e., linked shift)
0	1	1	The bit shifted out on the previous shift (left or right) of a Q Register input. (i.e., linked shift)
1	0	0	The bit shifted out (The LSB) on this right shift of this Register File input (i.e., end-around cyclic shift)
1	0	1	The bit shifted out (The LSB) on this right shift of the Q Register input (i.e., cyclic shift with the Q Register input)
1	1	0	The sign bit (from the Status Register) reflecting the sign of the output of the previous ALU operation.
1	1	1	Unassigned

TABLE 4-26

REGISTER FILE SHIFT CONTROLS (CON'T)

Bits			Q Register File Left Shift (MSB Shifted out, LSB vacated) Input to the LSB
27	26	25	
0	0	0	Zero
0	0	1	One
0	1	0	The bit shifted out on the previous shift (left or right) of a Register File input (i.e., linked sift)
0	1	1	The bit shifted out on the previous shift (left or right) of a Q Register input, (i.e., linked shift)
1	0	0	The bit shifted out (The MSB) on this left shift of this Register File input (i.e., end-around cyclic shift)
1	0	1	The bit shifted out (The MSB) on this left shift of the Q Register input (i.e., cyclic shift with the Q Register input)
1	1	0	The sign bit (from the Status Register) reflecting the sign of the output of the previous ALU operation.
1	1	1	Unassigned

TABLE 4-27

BRANCH ADDRESS SOURCE SELECTION

Bits 20 19	Branch Address Selected	
0 0	Firmware Branch Address	FWxx
0 1	Mapped Jump	MAPx
1 0	Return Address	STKx
1 1	Table Jump	LCTx

TABLE 4-28

MICROSEQUENCER COMMAND TYPE

Bits 21 20 19	Microsequencer Command Type
0 0 0	Branch
0 0 1	Map
0 1 0	Return (with stack "Pop")
0 1 1	Register
1 0 0	Call (with stack "push")
1 0 1	Unassigned
1 1 0	Unassigned
1 1 1	Unassigned

NOTE: An unconditional "Next" address operation occurs whenever the microsequencer field is used for a microprogram literal as an ALU input ("0000" in Bits 40-37) of an A-Type ALU operation.

TABLE 4-29

TABLE BRANCH OPERATIONS HAR1

Channel No	Mnemonic	Definition		Board
0 0	RFLG0/	A/R Flags (If		ADC
0 1	RFLG1	ADDR not to HAR_1)	MUX_0 ON	ADC
0 2	RFLG2	for ADDR Comp		ADC
0 3	RFG3		ADC Board	ADC
0 4	XFLG0			ADC
0 5	XFLG1	A/R Flags (If		ADC
0 6	XFLG2	ADDR is to		ADC
0 7	XFLG3	HAR_1) for ADDR Comp		ADC
1 0	DIRF/	RNI Instr. Reg Empty	Branch on Empty	ADC
1 1	ACTV/	Delayed RNI Not Active		ADC
1 2	XFE	Execution Frame Error	MUX_1 on ADC	ADC
1 3	FERR0	Frame Error (ADDR not to HAR1)	Board	ADC
1 4	FERR1	Frame Error (ADDR to HAR1)		ADC
1 5	CC1	Instruction Condition Code		ADC
1 6	CC0	Instruction Condition Code		ADC
1 7	OPA0/	Special OP A Branch	March 24, 1980	
2 0	CCDM	Composite Condition Met		SF
2 1	DLAD0/	Delta Address Bit0	MUX_2 ON	SF
2 2	DLAD1/	Delta Address Bit1	Special FUN	SF
2 3	MRDB	Memory Read Block		SF
2 4	0	Constant (Force A + 1)	Board	SF
2 5	1	Constant (Force Branch)		SF
2 6	K1	Instruction Word Length		SF
2 7	K0	Instruction Word Length		SF
3 0	FLG8			ALU
3 1	FLG9	Programmable Flags 8-11	MUX_3 ON	ALU
3 2	FLG10			ALU
3 3	FLG11			ALU
3 4	RQST	DMP Request CPU Diag.	ALU Board	ALU
3 5	SSW1	Switches		ALU
3 6	SSW2			ALU
3 7	SSW3			ALU
4 0	NEGA	ALU Result Is Negative		ALU
4 1	ZERO	ALU Result is Zero		ALU
4 2	KK1-KK0	Word Length = 6		ALU
4 3	LCFF	Loop CTR = FF		ALU
4 4	SPARE			
4 5	SPARE			
4 6	SPARE			
4 7	SPARE		Reserved for	
5 0	NEGA/		True & False	ALU
5 1	ZERO		Conditions	ALU
5 2	KK1-KK0/			ALU
5 3	LCFF/		MUX on	ALU
5 4	SPARE			
5 5	SPARE		ALU Board	
5 6	SPARE			
5 7	SPARE			
6 0	FLG0			ALU
6 1	FLG1			ALU
6 2	FLG2		MUX on	ALU
6 3	FLG3	Programmable Flags		ALU
6 4	FLG4	0-7	ALU	ALU
6 5	FLG5			ALU
6 6	FLG6		Board	ALU
6 7	FLG7			ALU
7 0	CRCT	Correction Needed for Division		ALU
7 1	OVFL	ALU Result is Over-flowed	MUX on	ALU
7 2	QLNK	Latched Q-Shift Out	ALU	ALU
7 3	RLNK	Latched R-Shift Out	Board	ALU
7 4	INTR	Interrupt=(EINT + IINT + FLG0 + FLG1)		
7 5	EINT	External INTR		ALU
7 6	IINT	Internal INTR		ALU
7 7	SPARE			ALU

TABLE 4-30

ADDRESS COMPUTATION

A/R FLAG MODIFICATION TABLE

ACEN/ (Not Address Comp)	FW 50 Reg or Log	FW FW 48 49 Flag Mod Type	FW FW 47 46 Flag Bit Addr	NFLG 3	NFLG 2	NFLG 1	NF 0
Not 1 ADC	X	X X	X X	LFLG3	LFLG2	LFLG1	LFL
0 (ADC)	0 (REG)	X X	X X	RFLG3	RFLG2	RFLG1	RFL
		0 1 CLR	X X	0	0	0	0
	1 LOGIC						
		0 1 CLR.	X X	0	0	0	0
		1 0 SET	0 0	LFLG3	LFLG2	LFLG1	1
			0 1	LFLG3	LFLG2	1	LFL
			1 0	LFLG3	1	LFLG1	LFL
			1 1	1	LFLG2	LFLG1	LFL
		1 1 RESET	0 0	LFLG3	LFLG2	LFLG1	0
			0 1	LFLG3	LFLG2	LFLG1	LFL
			1 1	0	LFLG2	LFLG1	LFL

TABLE 4-31

NUMBER OF BYTES ROTATE TRUTH

HAR2 - HAR1 Note 1	HAR1, HAR2,0 ADDR	KK00,KK01 or FW50, FW51 WD Length	No. of Rotates (Right or Left	Sel F B N
11	00	00 1 byte	3 (11)	
	00	01 2 bytes	2 (10)	
		10 4 bytes	0 (00)	
		11 3 bytes	1 (01)	
10	01	00 1 byte	2 (10)	
	01	01 2 bytes	1 (01)	
		10 4 bytes	3 (11)	
		11 3 bytes	0 (00)	
01	10	00 1 byte	1 (01)	
	10	01 2 bytes	0 (00)	
		10 4 bytes	2 (10)	
		11 3 bytes	3 (11)	
00	11	00 1 byte	0 (00)	
	11	01 2 bytes	3 (11)	
		10 4 bytes	1 (01)	
		11 3 bytes	2 (10)	

NOTES:

(1) When HAR2 - HAR1 is selected, the complement of the subtraction is used as the effective point of reference and the word length must be specified as one byte.

(2) Use HAR = 0 for fill

Status Bits

For each activation of the ALU, eight status bits are produced. These bits qualify the result produced by the operation, and are stored in registers for subsequent evaluation and use. The eight bits are:

Carry A carry bit of "1" is produced by the ALU whenever an arithmetic addition of the two 32-bit inputs produces a 33-bit result.

A carry bit is also produced when a subtraction operation yields a zero result. The carry register is updated when any ALU "add" or "sub" operation is executed.

> NOTE: The following three registers are updated only when specified by an update code in the firmware instruction.

Sign The sign bit is produced whenever there is a negative result.

Overflow The overflow bit is set whenever an arithmetic add of the two 32-bit inputs produces a result outside the range,

$$-2^{31} \text{ to } 2^{31}-1$$

Output (F)=0 The "F=0" bit is set whenever all 32 bits of the result (F) of an arithmetic or logical operation are zero, regardless of the carry bit.

> NOTE: "Linked 0" update, a series of ALU outputs can be tested to determine if they all equal zero. Any one of them not equalling zero causes "F=0" to be reset and maintained reset for the duration of the test.

Byte 0=0 The "byte 0=0" bit is set whenever the least significant byte (bits 7-0) of the result (F) of an arithmetic or logical operation are all zero.

> NOTE: When individual byte=0 condition exists, these are used for character testing only.

Byte 1=0 The "byte 1=0" bit is set whenever the second least significant byte (bits 15-8) of the result (F) of an arithmetic or logical operation are all zero.

Byte 2=0 The "byte 2=0" bit is set whenever the second most significant byte (Bits 23-16) of the result (F) of an arithmetic or logical operation are all zero.

Byte 3=0 The "byte 3=0" bit is set whenever the most significant byte (bits 31-24) of the result (F) of an arithmetic or logical operation are all zero.

ALU Operation Selection

Selecting of an ALU function is controlled by bits FW55, FW54 and FW53 of the firmware word latched into the instruction register. Refer to Table 4-6 for a representation of the ALU operation coded. As listed, this represents the functional conditions of the ALU adder.

To accommodate the multiply and divide functions, these lines are modified. This is covered in detail in the "Multiply/ Divide" paragraphs of this section.

In commanding the ALU, it must be remembered that the carry-in bit enters into the operation. Depending on the type of application of the operation, this bit requires handling. Handling of the carry bit is controlled by bits FW44 and FW43. The bit pattern are:

Bits		Disposition
44	43	
0	0	Force carry-in bit to zero (default)
0	1	Force carry-in bit to 1
1	0	Link previous carry-out bit to ALU carry-in
1	1	Link previous carry-out bit to ALU carry in

Control of the condition codes is handled by bits FW42 and FW41 of the firmware word. The bit patterns are:

Bits		Disposition
42	41	
0	0	No action
0	1	Update condition codes without "linked 0" test
1	0	No action
1	1	Update condition codes with "linked 0" test

ALU Inputs

Inputs to the ALU consist of seven selectable sources, three to the "R" input and four to the "S" input, two of which are shared. Six of these seven inputs are considered to be internal to the microprocessor (refer to Figure 4-3).

Internal ALU Inputs

The ALU register files are the primary source of input data to the ALU. For a detailed discussion, refer to the "Register Files" paragraphs in this section. The internal scratchpad 32-bit by 16 registers provide two inputs to the adder. By selective "A-addressing" the contents of any register (R0-R15) in the register file will be admitted to the "R" input or the "S" input.

An unshared internal input to the ALU is again the register file. In this case, by selective "B-addressing" the contents of any register (R0-R15) in the register file can be admitted to the "S" input. Another single internal input to the ALU "S" input is the "Q" register, whose purpose is to return the ALU output back to an input, shifted or unshifted. The "Q" register will be discussed in more detail later in this section.

External ALU Inputs

The ALU receives "external" inputs via the data bus, which is connected to the "R" input. The data bus provides the data for one of sixteen "multiplexed external inputs. The inputs available through this multiplexing are:

- Memory Read Register
- ALU Status Bits
- Rotation Output
- Literal Data From Instruction Register
- Hardware Address Register 1
- Hardware Address Register 2
- Backup Program Counter (24-bits)
- RNI Label
- RNI Literal

The selection of the data bus as an ALU input must always be thought of in terms of external input multiplexor selection, described in more detail later in this section.

ALU Input Summary

Of the "three plus four" (R and S, respectively) inputs, not all are legal input combinations. By design, only eight of the twelve possible combinations are implemented. The following table summarizes the twelve possible combinations and indicates which have been implemented.

R / S	Reg. File (A. Addr)	Data Bus	Zeros
"Zeroes"	No	Yes	No
Reg. File (A-Addr)	No	Yes	Yes
Reg. File (S-Addr)	Yes	No	Yes
Q-Reg.	Yes	Yes	Yes

ALU Input Selection

Selection inputs to the ALU "R" and "S" inputs is controlled by bits FW48, FW57 and FW56 of the firmware word latched into the instruction register. Each of the eight possible bit combinations cause a 32-bit string to be admitted into the "R" input and another 32-bit string to be admitted into the "S" input. The bit patterns are:

Bits			Input Selection	
58	57	56	"R" Input	"S" Input
0	0	0	A-Addressed Reg. File Register	Q-Register
0	0	1	A-Addressed Reg File Register	B-Addressed Reg. File Register
0	1	0	"Zeros"	Q-Register
0	1	1	"Zeros"	B-Addressed Reg. File Register
1	0	0	"Zeros"	A-Addressed Reg. File Register
1	0	1	Data Bus	A-Addressed Reg. File Register
1	1	0	Data Bus	Q-Register
1	1	1	Data Bus	"Zeros"

In considering the above selections, it must be remembered that a register file selection and a data bus selection must both be further qualified to be meaningful. When the data bus is selected as the "R" input, this selection is indirectly enabling the entry of one of 16 possible "external" inputs to the

ALU. Gating the desired one of 16 external inputs into the data bus is controlled by bits, FW40, FW39, FW38 and FW37 of the firmware word latched into the instruction register. Each of the 16 possible bit combinations causes a 32-bit string to be gated onto the data bus.

> NOTE: When the input is less than 32 bits, it is impressed on the data bus right justified.

Unused upper bytes are forced to zero, except for the condition codes. The external source selection is shown in Table 4-3.

When a register file is selected for the "R" or "S" (or both) inputs, this selection is gating one of the 16 register's contents onto a bus connected to the appropriate input (R or S). The "A-addressed" bus connected to both inputs (R and S) while the "B-addressed" bus is connected only to the "S" input.

> NOTE: The "A-addressed" bus is also connected to an ALU output.

Addressing of the output from the register file is controlled by bits FW42, FW51, FW50 and FW49 (for the A-bus) and bits FW48, FW47, FW46 and FW45 (for the B-bus) of the firmware word latched into the instruction register.

> NOTE: B-addressing is also used to control an ALU output.

Each of the 16 possible bit combinations causes the 32-bit contents of the applicable register to be impressed on the appropriate bus. The bit patterns are:

A-Address				A-Bus	B-Address				B-Bus
				Contents	Bits				Contents
52	51	50	49		48	47	46	45	
0	0	0	0	R0	0	0	0	0	R0
0	0	0	1	R1	0	0	0	1	R1
0	0	1	0	R2	0	0	1	0	R2
0	0	1	1	R3	0	0	1	1	R3
0	1	0	0	R4	0	1	0	0	R4
0	1	0	1	R5	0	1	0	1	R5
0	1	1	0	R6	0	1	1	0	R6
0	1	1	1	R7	0	1	1	1	R7
1	0	0	0	R8	1	0	0	0	R8
1	0	0	1	R9	1	0	0	1	R9
1	0	1	0	R10	1	0	1	0	R10
1	0	1	1	R11	1	0	1	1	R11
1	1	0	0	R12	1	1	0	0	R12
1	1	0	1	R13	1	1	0	1	R13
1	1	1	0	R14	1	1	1	0	R14
1	1	1	1	R15	1	1	1	1	R15

In summary, admitting bit strings to both the "R" and "S" ALU inputs requires consideration of the following factors, most of which are controllable by the firmware word (refer to Figure 4-3).

(a) Admitting one of the three available binary strings to the "R" input and one of the four available binary strings to the "S" input (IR bits FW48-FW56).

(b) When the selected "R" input is the A-address bus (i.e., the contents of a register file register), the desired register (R0-R15) must be specified (IR bits FW52-FW49).

(c) When the selected "R" input is the data bus, the desired external data source must be specified (IR bits FW40-FW37).

(d) When the selected "R" input is "zero," no other consideration is implied.

(e) When the selected "S" input is the A-address bus (i.e., the contents of a register file register), the desired register (R0-R15) must be specified (IR bits FW52-FW49).

(f) When the selected "S" input is the "B" address bus (i.e., the contents of a register file register), the desired register (R0-R15) must be specified (IR bits FW48-FW45). Note that the "B-address" is also used to admit the ALU "F" output into the register file. Thus, using a B-address input precludes using the "B" address to return the ALU result to the register file, unless the result is to be stored in the same register that the "S" input was taken from (i.e., the original contents of the B-addressed register are overlaid at the end of the instruction cycle).

(g) When the selected "S" input is "zeros," no other consideration is implied.

(h) When the selected "S" input is the "Q" register, no other consideration is directly implied. However, a "shifting" consideration may be indirectly implied.

In reviewing the preceding eight points, it becomes evident that careful selection of available options can enhance the efficiency of the firmware. For example, choosing an A-addressed "S" input rather than, for the same register, a B-addressed "S" input, retains the option of using the B-address to return the ALU result to any register file register in the same instruction

cycle. A detailed study of Figure 4-3 will reveal other similar opportunities to exploit the available power of a firmware word. This study should also reveal nonefficient or even illegal operations that are not readily apparent to the firmware programmer.

ALU Output

Output from the ALU is the 32-bit result (plus a possible carry bit) of the operation (arithmetic or logical) performed on the two 32 bit strings from the "R" and "S" inputs. This output can be directed to internal and/or external destinations.

1. Internal ALU Output Destinations

The ALU output (i.e., result) is automatically latched into the "F" register, which is integral to the ALU. The contents of this register are automatically impressed on the "F" bus. The "F" bus is connected to three internal destinations making the ALU result available to all of them simultaneously. These destinations are the register file, the "Q" register, and the external output selection multiplexor (refer to Figure 4-3).

2. External ALU Output Destinations

The ALU "F" output is connected to a 2:1 multiplexor whose single output is the ALU output bus. The second input to this multiplexor is the file register "A" bus. This leg of the "A" bus allows the contents of an A address register file register to be applied to the output bus instead of the ALU "F" output. Thus, it can be seen that when the ALU "F" output is not needed for an external ALU output, an A-addressed register can be directed to an external destination simultaneously with the "F" output being used internally.

The output contents, called "Y" register, are not automatically impressed on the CPU bus, but rather are gated onto the bus by an "ENABLE2" command derived from the ALU operation identifier. The ALU "Y" output is gated onto the CPU bus when the A-type command is not an address computation or rotate operation.

ALU Output Destination Selection

The ALU destination selection process is somewhat more complex than the input selection process. There are two outputs to consider (F and Y), and because of the integral nature of the "Q" register, control of inputting and outputting of the "Q" register is integrated with the ALU output control. (The "Q" register is covered in more detail later.)

ALU output destination selection (AND "Q" register selection) is controlled by bits FW61, Fw60, and FW59 of the firmware word latched into the instruction register. These three bits provide

eight selection choices, but each choice usually affects more than one selection. Refer to Table 4-8 for the ALU output control selections.

In considering the ALU output selections, it should be remembered that a register file selection and a CPU bus selection must both be further qualified to be meaningful. When the register file is selected as a destination, the specific register must be addressed. Addressing of the input to the register file is controlled by bits FW48, FW47, FW46, and FW45 of the firmware word latched into the instruction register. It should also be remembered that the B-address serves a dual function of both input and output addressing. Each of the 16 possible bit combinations causes the 32-bit ALU output from the "F" register to be admitted to the addressed register file register. The bit patterns are:

B-Address Bits				"F" Output To Register
48	47	46	45	
0	0	0	0	R0
0	0	0	0	R1
0	0	1	0	R2
0	0	1	1	R3
0	1	0	0	R4
0	1	0	1	R5
0	1	1	0	R6
0	1	1	1	R7
1	0	0	0	R8
1	0	0	1	R9
1	0	1	0	R10
1	0	1	1	R11
1	1	0	0	R12
1	1	0	1	R13
1	1	1	0	R14
1	1	1	1	R15

When the CPU is selected as a destination, there are two considerations. The first involves a selection of one of the two available outputs to the CPU bus; ALU "F" output or the A-addressed register file. The second involves selecting an "external" destination for the CPU bus. Selecting of one of the available outputs to the CPU bus is done by bits FW61, FW60, and FW59 of the firmware word. However, when the register file is part of the selection, the specific register must be addressed.

Addressing of this output (A-bus) from the register file is controlled by bits FW52, FW51, FW50, and FW49 of the firmware

word latched into the instruction register. It must also be remembered that the A-address serves a dual function of both ALU input and ALU output addressing. Each of the 16 possible bit combinations causes the 32-bit contents of the addressed register to be admitted to the ALU "Y" output register and thus to the CPU bus. The bit patterns are:

A-Address				CPU Bus
Bits				Contents
52	51	50	49	
0	0	0	0	R0
0	0	0	1	R1
0	0	1	0	R2
0	0	1	1	R3
0	1	0	0	R4
0	1	0	1	R5
0	1	1	0	R6
0	1	1	1	R7
1	0	0	0	R8
1	0	0	1	R9
1	0	1	0	R10
1	0	1	1	R11
1	1	0	0	R12
1	1	0	1	R13
1	1	1	0	R14
1	1	1	1	R15

When the CPU bus in involved in output selection (i.e., when there is "external" destination), this indirectly enables the passing of the contents of the CPU bus to the external destinations. Selection of the external destination is controlled by bits FW36, FW35, FW34, and FW33 of the firmware word latched into the instruction register. Each of the possible 16 bit combinations select one of the 16 external destinations. The CPU external destination control is shown in Table 4-2.

Whenever the A/R file is selected as an external destination, there is a restriction that an ALU arithmetic operation is not allowed. All other ALU operations are permissable.

Finally, in reviewing the output destination selections, there are four of them (100-111) that involve shifting of inputs to the register file and/or the "Q" register. Using one of these choices mandates the use of a B-type operation that will handle the bits that are being shifted in/out of the shifted string being placed in the register file and/or "Q" register. The B-type operation dealing with these bits is described later in this document. Note that when both the register file and "Q" register inputs are shifted in the same instruction, they are both shifted in the same direction.

ALU OUTPUT DESTINATION SELECTION SUMMARY

Internally routing the ALU result (internally with respect to the microprocessor) and impressing an output (ALU result or a register file register) on the CPU bus requires consideration of the following factors, most of which are controllable by the firmware word (refer to Figure 4-3).

(a) The ALU result (F register) has three possible destinations: Q register, register file and/or CPU bus. Control of these destinations is a function of system design and is affected in predetermined combinations (FW bits 61-59).

(b) When a selected destination is the register file, the desired register must be "B-addressed" (FW bits 48-45). Note that the B-address is also used to impress the contents of a register onto the B bus (ALU "S" input). If both functions are employed, the "read" to the B bus is affected first, followed by an overlaying of the just-read contents with the ALU "F" output.

(c) When a selected destination is the CPU bus, there is an implied further selection of either the ALU result (F) or the contents of an A-addressed (FW bits 52-49) register file register. Note that the A-address is also used to impress the contents of a register on the A bus (ALU R or S inputs).

    Further note that in conjunction with (b) above, the ALU result (F) can be written into any register in the register file, file (A address). It is also possible to have a third simultaneous action occur by having the contents of the "Q" register read to the Q-shift circuits, shifted, and the shifted result read back into the "Q" register.

(d) When a selected destination is the CPU bus, the desired external destination must be specified (FW bits 36-33).

In reviewing the preceding four points, it becomes evident that careful selection of available options can enhance the efficiency of the firmware. For example, choosing an A-addressed "S" input rather than, for the same register, a B-addressed "S" input, retains the option of using the B-address to return the ALU result to any register file register in the same instruction cycle.

A detailed study of Figure 4-3 will reveal other similar opportunities to exploit the available power of a firmware word. This study should also reveal "nonefficient" or even illegal operations that are not readily apparent to the firmware programmer.

Register File

The register file is a 32-bit by 16-word register, high-speed "scratchpad" memory for the arithmetic logic unit. The input (write) to these registers is the ALU output (F). This output can be written into the register file either "as is" shifted left one bit, or shifted right one bit. The "shift" decision is integral to the ALU output destination selection circuits. However, when a selection is made that involves a shift, this mandates a B-type operation that deals with the bit being shifted into, and the bit being shifted out of, the 32-bit string being shifted. (B-type operations are covered in detail later in this document.

The output of the register file consists of two paths (busses) that are individually (and jointly) addressable. The A-bus connects the register file with (1) the ALU "R" input, (2) the ALU "S" input, and (3) the ALU "Y" output via the 2:1 multiplexing circuits. The "B" bus connects the register file with only the ALU "S" input. Register selection is accomplished by "A" and "B" addressing, "A" for reading a register and "B" for reading and/or writing a register. That is:

- The "A" address reads the contents of a register to the "A" bus.
- The "B" address reads the contents of a register to the "B" bus.
- The "B" address writes the ALU result (F) shifted or unshifted to a register.

Thus, any one of the following operations can be performed with the register file in a single instruction cycle.

(a) Read the contents of one register to the "A" bus.

(b) Read the contents of one register to the "B" bus.

(c) Read the contents of one register to the "A" bus and the "B" bus.

(d) Read the contents of one register to the "A" bus and read the contents of another register to the "B" bus.

(e) Read the contents of one register to the "A" bus and write the ALU result (F), shifted or unshifted, back into another register.

(f) Read the contents of one register to the "A" bus and write the ALU results (F), shifted or unshifted, back into the same register.

(g) Read the controls of one register to the "B" bus and write the ALU result (F), shifted or unshifted, back into the same register.

(h) Read the contents of one register to the "A" bus, read the contents of another register to the "B" bus, and write the ALU result (F), shifted or unshifted, back into the "B-read" register.

(i) Read the contents of one register to the "A" bus and the "B" bus, and write the ALU result (F), shifted or unshifted, back into the "B-read" register.

(j) Write the ALU result (F), shifted or unshifted, into any register.

"Q" Register

The "Q" register is a single 32-bit "utility" register integral to the arithmetic logic unit. The "Q" register can be used for "scratchpad" memory and as an accumulator, although not as efficiently as a register file register. Inputting to and outputting from the "Q" register is controlled in conjunction with ALU output selection, as is use of the "Q" register "shifter." Refer to Figure 4-3.

A selection that involves a shift mandates a B-type operation that deals with the bit being shifted into, and the bit being shifted out of, the 32-bit string being shifted. (B-type operations are covered in detail in other portions of this document.) With its attendant "shifter" and in conjunction with a shifting of the register file input, a 64-bit shift can also be affected.

ROTATE OPERATIONS

Rotate operations involve cycle shifts of zero, one, two or three bytes, either to the right or to the left. Right shifts are used generally to "right-justify" arithmetic values for subsequent processing. Left shifts are generally used to restore previously right-shifted values.

In conjunction with right shifts, there is a "sign extend" function that fills all leading zero positions with the value's sign bit, which is required for the internal handling of mathematical operations. These operations are made necessary by the format in which data values are stored in memory, as explained in the following example. There is also a zero extend capability which is similar to sign extend. For examples of the use of rotate, refer to Figure 4-22.

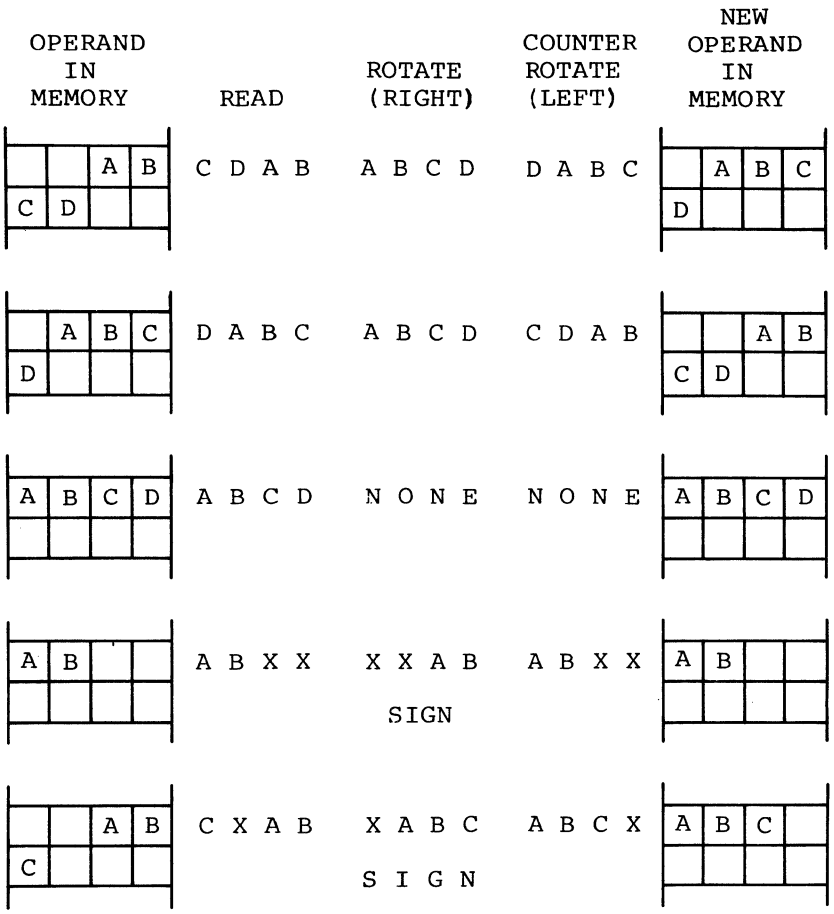


Figure 4-22. Examples of Use of Rotate

REALITY memory is organized into sequential and contiguous 4-byte groups referred to in this discussion as "words," i.e.:

```
Word----------------Increasing Byte Address

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

   4               "    "    "    "

   5               "    "    "    "
Increasing Word
Address
```

Note that the byte significance in a word decreases from left to right while the byte address in memory increases from left to right.

Additionally, the 32-bit strings manipulated by the microprocessor are also organized into 4-byte groups referred to a "words," i.e.:

Byte3 Byte2 Byte1 Byte0

However, the functional "word" dealt with by the microprocessor need not (and generally does not) coincide with the physical "word" of memory. That is to say, a functional "word" of bytes 3 through 0 could be stored in memory in the following way.

```
1
2                Byte3 Byte2
3  Byte1 Byte0
4
```

When the above functional word is retrieved from memory into the memory read register, it is formatted as follows:

Byte3 Byte2 Byte1 Byte0

It is the function of the rotate circuits to adjust the memory format required by the microprocessor, i.e.:

Byte3 Byte2 Byte1 Byte0

This is accomplished for the above example by a 2-byte cyclic right shift. Finally, when the value is to be returned to the same memory location, the rotation circuits must again be employed to counter-rotate (2-byte cyclic left shift) the value back to its memory-store format.

An ancillary function of the rotate circuits is the "fill" capability. This permits any byte of a 4-byte word to be duplicated in the other three bytes. That is, with byte-2 selected as the pattern byte, the word:

0 1 1 0 1 1 1 1 0 0 1 0 1 0 1 0

Byte-3 Byte-2 Byte-1 Byte-0

can be filled to:

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Byte-3 Byte-2 Byte-1 Byte-0

The rotate functions along with the ALU and address computation functions share the same instruction register (refer to Figure 4-23). For the rotate functions, bits FW43 through FW49 are "don't care cases." Bits FW63 and FW62 are "1" and "0," respectively, to identify the instruction as a rotate command.

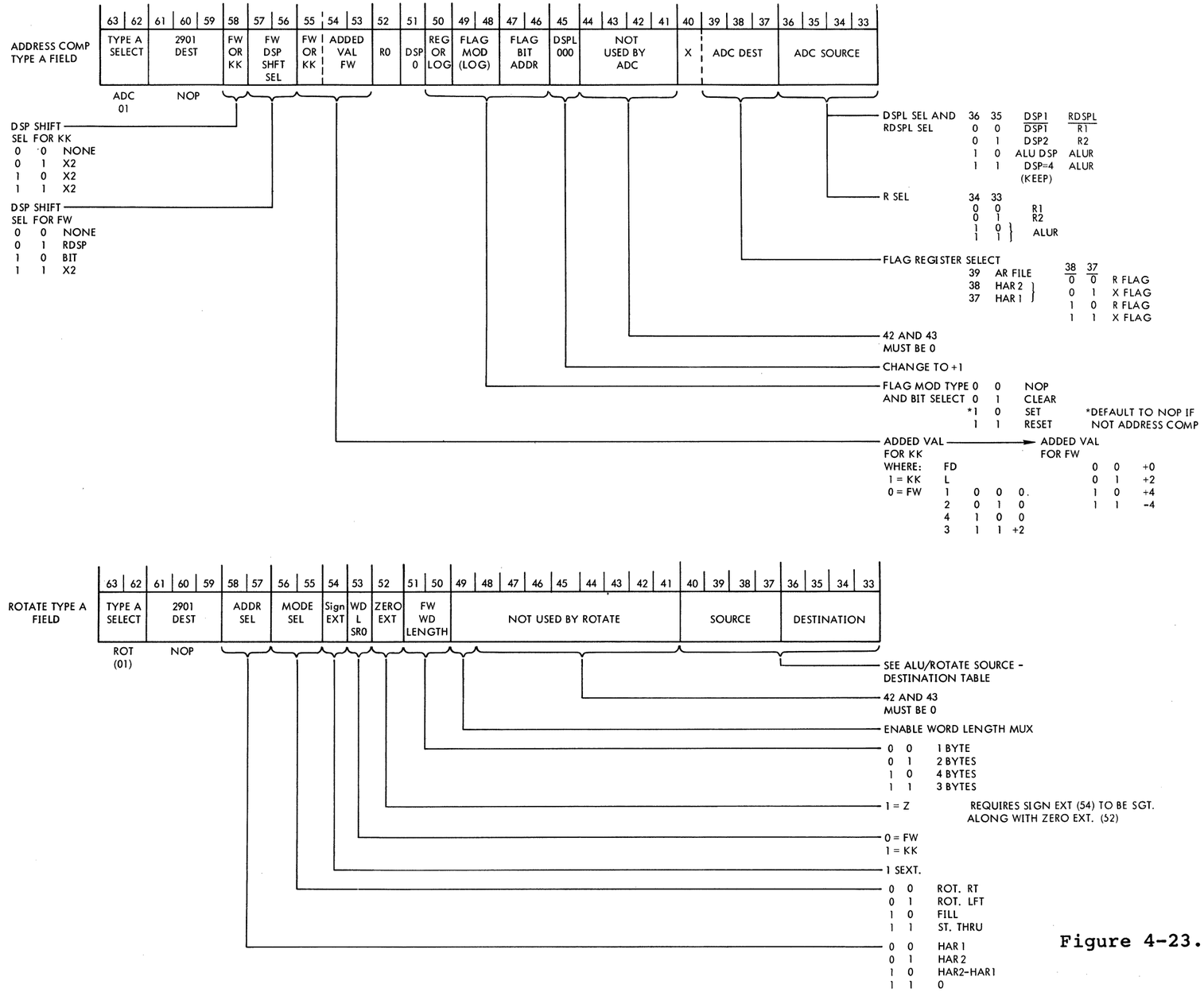


Figure 4-23. Address Comp and Rotate Type A Field Functions

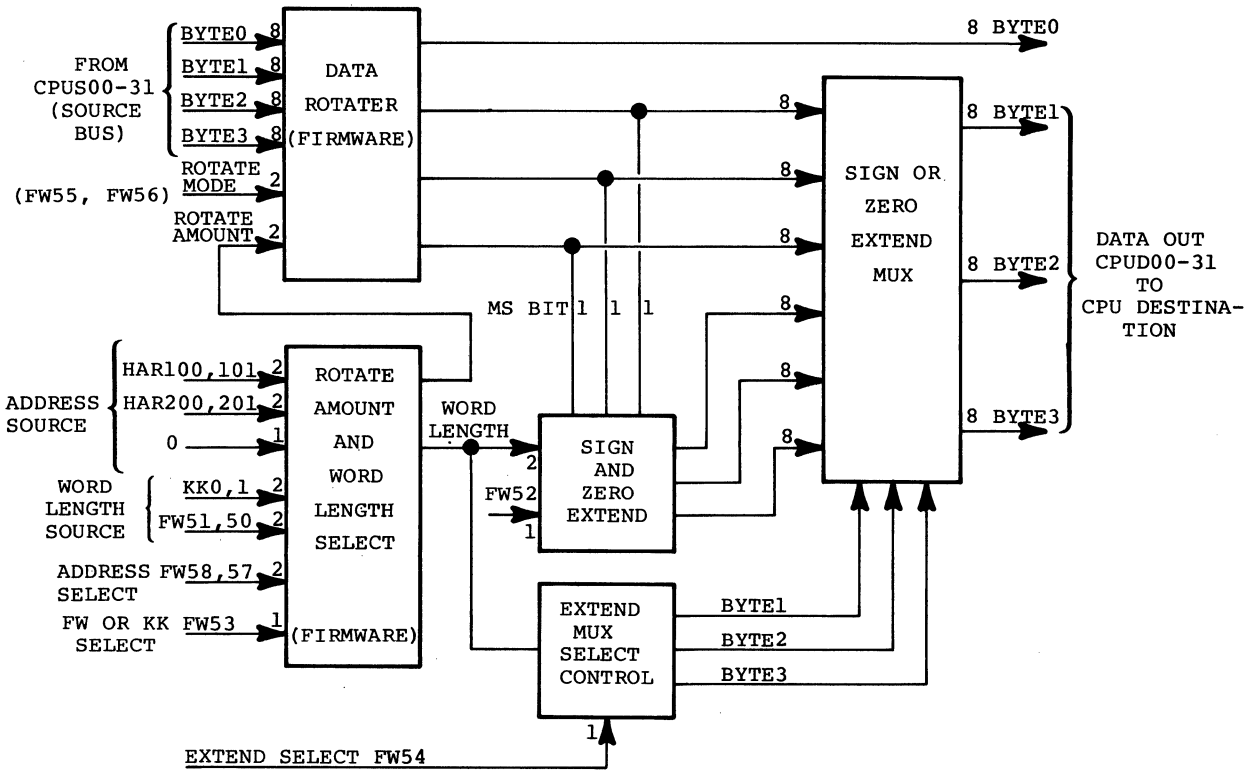


Figure 4-24. Rotate and Sign Extend Diagram

Bits FW61 through FW59 are "0," "0," "1," respectively, to block ALU output activity during the rotate instruction cycle. The remainder of the rotate command is described in the paragraphs that follow. Figure 4-24 is a functional block diagram of the rotate and sign extend operation.

Rotate Input Selection

Selection of the source of the 32-bit, 4-byte string that is to be rotated is controlled by bits FW40, FW39, FW38 and FW37 of the firmware word latched into the instruction register. The same sources as the ALU are possible, but there are only two normally used sources; the "X" register and the memory read register. Usually, the memory read register is the source when the data is coming from memory or a peripheral device (prior to processing), while the "X" register is the source for data (following processing), for data going back to memory or a peripheral device via the memory write register (refer to Figure 4-4).

Note that for either input, the desired data must have been loaded into the source register "X" or memory read during an instruction cycle previous to the rotate instruction cycle.

For either input, the length of the input string must be qualified. This qualification is controlled by bit FW53 of the firmware word latched into the instruction register. This bit is a "pointer" to the qualification data. The bit patterns are:

Bit 53	Qualification
0	Word length is contained in "FW" (Explicitly)
1	Word length is contained in "KK" (Referenced)

"KK" is a word length parameter contained in the source assembly instruction (operations system assembly code) from which the firmware word was derived. "FW" is a word length contained in the firmware word itself in the field consisting of bits FW51 and FW50. These two bits provide the qualification of the four possible words (to be rotated), and lengths. The bit patterns are:

Bit 51	50	Qualified Word Length
0	0	1-Byte
0	1	2-Bytes
1	0	3-Bytes
1	1	4-Bytes

The final piece of information regarding the input string that is needed by the rotate circuits is the location of the most significant byte of the input, i.e., the arrangement of the bytes in the memory read register.

This arrangement is a factor of how the word was stored in memory. This information is derived from the memory address of the most significant byte of the input string. This address (and the implied byte arrangement) is referred to by an "address pointer." This pointer is embodied in bits FW58 and FW57 of the firmware word latched into the instruction register. These two bits reference the register(s) that contain(s) the memory address of the data being brought into the memory read register from memory. The bit patterns are:

Bits		Referenced Address Source
58	57	
0	0	Hardware Address Register 1 (HAR1)
0	1	Hardware Address Register 2 (HAR2)
1	0	HAR1-HAR2 (HAR1 minus HAR2)
1	1	0

In summary, with the following information:

(1) Input data source, usually the "X" register or memory read register,

(2) The length of the word to be rotated, and,

(3) The word byte arrangement.

The rotation circuits can perform the rotation of the desired number of bytes. The direction of rotation (left or right), however, is commanded separately as described in the following paragraph.

Rotate Operation Selection

Rotation is commanded by bits FW56 and FW55 of the firmware word latched into the instruction register. These two bits command the direction of rotation (right on input from memory and left on output to memory) plus the "fill" facility, and a "straight through" (no rotation) when the input/output word is coincident in the proper arrangement. The bit patterns are:

Bits		Rotation Command
56	55	
0	0	Rotate Right
0	1	Rotate Left
1	0	Fill
1	1	Straight Through (No Rotation)

In the case where a right rotation is being performed on a value to align it for ALU processing, the "sign-extend" or "zero-extend" facility can be invoked for ALU processing. The sign bit must be in the leftmost bit position. The sign-extend facility performs the desired transportation of the sign bit. In the same manner, a "zero bit" can be extended instead of the sign bit. Sign bit extension is controlled by FW54 while zero bit extension is controlled by FW52, along with FW54 = 1. The bit patterns are:

Bit	Sign Extend Command	Bit	Zero Extend Command
54		52	
0	No Sign Extend	0	No Zero Extend
1	Invoke Sign Extend	1	Invoke Zero Extend

Note: Both bits must be set to perform the zero extension.

Rotate Output Selection

The output of the rotation circuit is a 32-bit, 4-byte string. This output is placed on the CPU destination bus. Destination selection is controlled by bits FW36, FW35, FW34 and FW33 of the firmware word latched into the instruction register. These are the same as the ALU destinations. Normally the only two destinations used for rotate are the memory write and the "X" register.

Bits		Output Select
34	33	
1	0	Memory Write Register
1	1	"X" Register

Note: These output selections are obtained with bits FW36 and FW35 both zero.

Rotate, Special Case

The rotate logic is used to support a special case requirement in the ALU. The rotate, type A command, is invoked with FW49 to latch two data word length lines. These two lines, LWDLNO and LWDLNI are used on the ALU to determine negative sign bit and overflow operation. This allows the ALU to make these decisions on byte boundaries. Refer to Figure 4-24 for an overview. A type "B" control command with FW23 low is required to clear these latched two lines.

ADDRESS COMPUTATION OPERATIONS

In the REALITY system, the management of the main memory resource centers on the principle of keeping in main memory only the information (instructions and data) that is needed at the time. When it is not needed immediately, the information is stored on disc. Information is then, as required, moved back and forth between disc and main memory in "packets" that consist of 512 bytes referenced by its frame address. Specific data value is also referenced by a displacement within the frame.

When a frame is moved from disc to main memory, it is placed in whatever frame space is available. Once in memory a piece of information in the frame must now be referenced by both its displacement and the location of the frame in the main memory, i.e., its location in the frame with respect to the main memory address of the first byte of the frame. Therefore, it is the function of the address computation circuits to provide an absolute main memory address.

The displacement concept is made up of three components. First, data is always referenced by a base register and, second, a displacement away from the base register.

> NOTE: The base register, whose value is stored in A/RFILE, need not point to the first byte of a memory buffer/frame. It may have a displacement component from the beginning of the frame.

The third component of the displacement has to do with whether a frame is in linked or unlinked format. In unlinked format, all 512 bytes of a frame are available, with the first available byte being called "byte zero." In this case, there is no third displacement component. If a frame is in linked format, only the last 500 bytes are available and the first of the available bytes is called "byte 1." Therefore, there is a displacement of eleven which must be added to the already computed displacement to generate the actual memory address of the data. The address computation circuits do not deal with the third component of displacement because this is handled by the CPU ALU when converting from disc address to main memory address.

The address computation process is used whenever an operand is being fetched from main memory, i.e., being placed in the memory read register for processing in the subsequent instruction cycle.

A firmware address computation instruction causes an operation that uses:

(a) A base memory address at the very beginning of the frame or within the frame in which the operand is located.

(b) The "displacement," or the position in the frame of the beginning bit/byte of the operand with respect to the base memory address.

(c) The length of the operand, and:

(d) The operand logical boundaries (bit, byte, multibyte).

The created instruction then causes the address computation circuits to compute the absolute address of the operand and to load this address into one of the two hardware address registers. This address is then used to fetch the operand from memory to the memory read register, making it available as an ALU input. A simplified functional block diagram of the address computation circuits is shown in Figure 4-25.

In order for an address computation to be valid, the selected location in the address register file must have been put into the attached form by a prior ALU operation. This means that the virtual address, consisting of a frame identification (FID) and a displacement, is converted to a main memory address. During address computation the displacement, added to the attached register, is either derived from the instruction (DSPL1, DSPL2) or provided by the ALU (ALUDSP) and is not the same as the displacement contained in the virtual address.

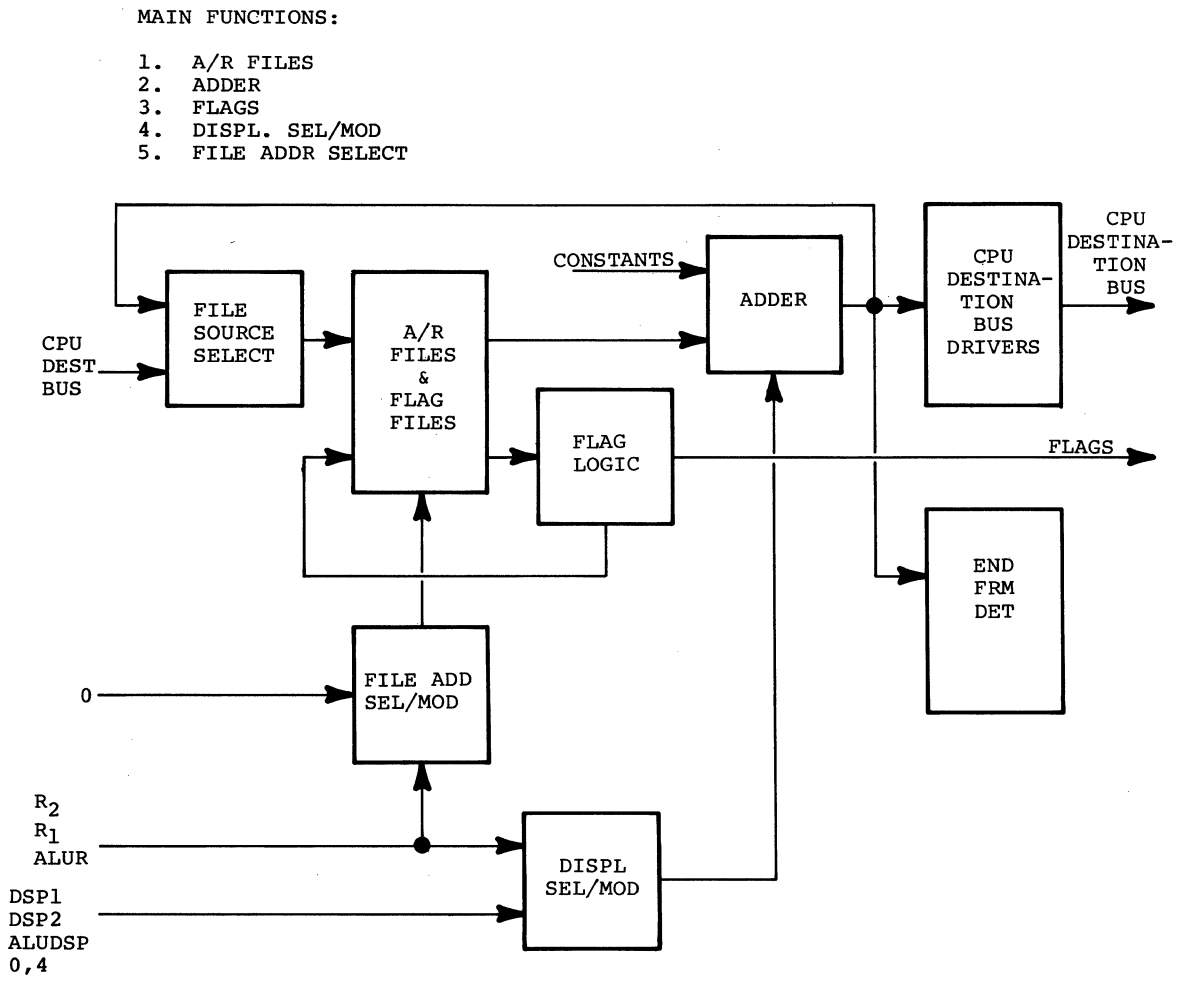


Figure 4-25. Address Computation (Top Level)

The general expression for an address computation contains the following four items:

Computed Address	=	Content of Selected Address Register File Location	+	Displacement Factor	+	Added Constant

Each of these items has many variations, which are controlled by firmware bits, and by KK (from RNI).

In order to facilitate describing these variations, the general expression shown above can be illustrated in its two basic formats:

General Expression

Computed Address	=	Content of Selected Address Register File Location	+	Displacement Factor	+	Added Constant

Basic Format

(a)	ADC Destination	=	C	r(x)	+	DSPLY SHIFT	+ Constant
(b)	ADC Destination	=	C	r(x)	+	RDSPLY 1	+ Constant

The two basic formats are the same except for the displacement factor where RDSPLY is a special case of displacement shift. To see an overview of all firmware command fields used in address computation refer to Figure 4-23.

Address Register File

The address register (AR) file, shown toward the upper center of Figure 4-26, is a 24-bit-wide by 16-register-deep memory for storing base addresses. (A base address is a main memory address.)

The first register in the address register file (AR0) always contains the base address for the current (i.e., now executing) program control block (PCB). The second register in the file (AR 1) always contains the base address of the current program frame. The remaining registers are not dedicated.

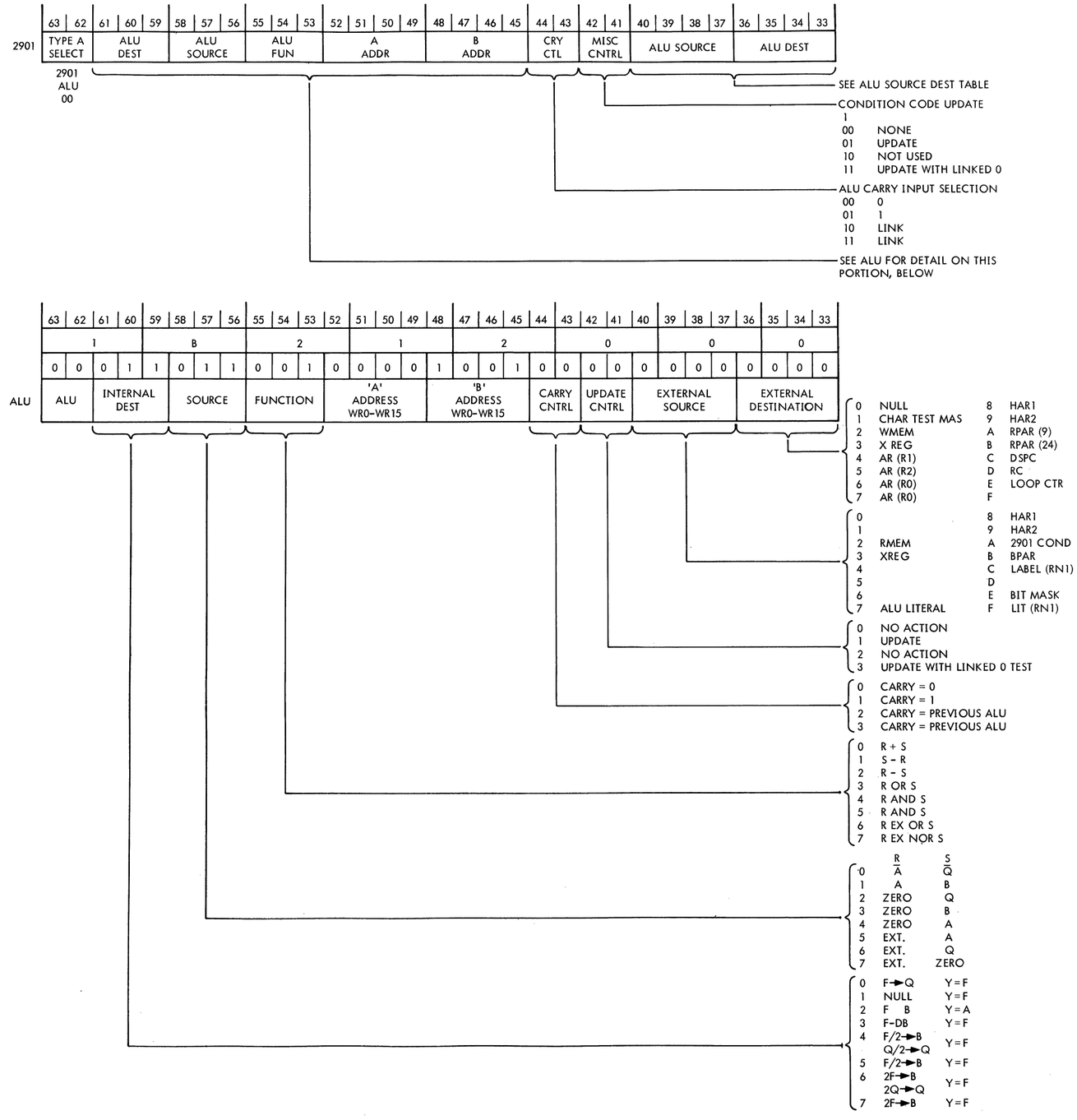


Figure 4-26. Address Register File

The AR file can be loaded by the CPU or from the result of an ADC operation. When it is loaded, bits 0-11 of the input (base addresses) to the AR file is supplied by either the CPU bus (ALU) or from the output of the address computation process. Bits 12-23 are always supplied by the CPU bus. Therefore, bits 12-23 are never updated by an address computation operation.

Base Address Selection (AR File Output)

The selection of one of the 16 base addresses from the AR file is controlled by bits FW52, FW34, and FW33 of the firmware word latched into the instruction register. These three bits select one of three 4-bit registers that contains an address of an AR file register, which in turn contains a 24-bit main memory base address. The bit patterns are:

Address File Source	Firmware Bits		
R Selection (RX)	52	34	33
R_1	0	0	0
R_2	0	0	1
ALUR (RC)	0	1	0
ALUR (RC)	0	1	1
R = 0	1	X	X

R1 and/or R2 are loaded with 4-bit addresses from the RNI circuits. ALUR is loaded with a 4-bit address from the ALU (CPU bus). R1 and R2 are the most often used sources while ALUR is provided as an address source so that instructions that implicitly reference an AR file register can be handled. Whichever source is selected, the 4-bit address is then used to address the output and/or input of the AR file. The 24-bit base address (AR file output) is directed through another register that allows only the lower nine bits of the base address to be combined with the appropriate displacement factor as discussed.

The selected AR file address, on its way to the AR input/output control circuits, is intercepted by a group of circuits that allow this address to be forced to zero, i.e., to access the base address (in AR0) of the current program control block (PCB). The "force-to-zero" is commanded by bit 52 of the firmware word.

Displacement Selection

As previously mentioned, locating an operand in main memory requires both the base address of the frame in which it resides, and its relative position (displacement) in the frame. Obtaining

this displacement is controlled by bits 51, 36, and 35 of the firmware word latched into the instruction register. These bits select one of three 8-bit registers that will contain a frame displacement factor, or force displacement values of 0 or 4.

The displacement selection is controlled by firmware bits 51, 36, and 35 as follows:

	Firmware Bits		
Displacement Selection	51	36	35
DSP 1	0	0	0
DSP 2	0	0	1
ALU DSP	0	1	0
DSPL = 4	0	1	1
DSPL = 0	1	X	X

DSP1 and/or DSP2 are loaded with 8-bit displacement factors from the RNI circuits. ALU DSP is loaded with an 8-bit displacement factor from the ALU (CPU bus). DSP1 and DSP2 are the most often used sources while ALU DSP is provided (in conjunction with ALU R described previously) as a displacement source for instructions impliciting referencing a base address. The output of the displacement source is "preprocessed" by a shifter and/or an adder before being combined with a base address as discussed in "Memory Address Destination Selection."

The displacement can be forced to zero. This "force to zero" is commanded by bit 51 of the firmware word. When this bit is a 1, the output of the selection process (i.e., the 4.1 multiplexing circuits) is inhibited, effectively causing all zeros to be sent to the length processing address (see "Displacement Adder" section).

In conjunction with the displacement selection, the hardware uses the lower three bits of the selected displacement to create the displacement bit mask, which is an "external" input to the ALU (bits 40-37 equal 1110, respectively). This mask is generally used in bit operand operations and consists of 32 bits in which the lower eight are significant. Bits 31-8 are always 1s.

The lower three bits of the selected displacement are used to address one of the eight bits (0-7) of the mask, whose bit is reset to zero while the other seven default to 1s. The bit mask can then be used to isolate the addressed bit in the byte (in an ALU logical operation, for example).

Displacement Shifting

The selected displacement factor is sent to multiplexing circuits in three forms: (1) as is, (2) shifted one bit left, and (3) shifted three bits right. The "as is" form indicates that the operand is one byte in length.

> NOTE: Since the displacement value is embodied in eight bits, a 1-byte operand must be located within the first 256 bytes of the frame. However, this restriction can be circumvented by having the base address point to a byte other than the frame's first byte.

The "one bit left" form indicates that the operand is either two, four, or six bytes in length. Shifting left one bit of the displacement effectively doubles its value, and thus automatically provides a 2-byte, or word displacement into the entire 512-byte, 256-word frame. The "three bits right" form indicates that the operand consists of a single bit.

> NOTE: Since the displacement value is embodied in eight bits, a 1-bit operand must be located within the first 256 bits, i.e., the first 32 bytes of the frame. However, this restriction can be circumvented by having the base address point to a byte other than the frame's first byte. Shifting right three bits of the displacement effectively divides its value by eight.

Shifting is controlled by bits 57 and 56 of the firmware word latched into the instruction register. Bit 58 specifies whether the shift control is to be explicitly determined by bits 57 and 56 of the firmware word (bit 58 = 0) or implicitly controlled by a reference to the KK field of the current software instruction (bit 58-1). The KK field is a 2-bit field that will contain one of the following length indicators.

KK	Operand Length
00	1 Byte
01	Word (or "Tally"), i.e., 2 Bytes
10	Double Word (or "Tally"), i.e., 4 Bytes
11	Triple Word (or "Tally"), i.e., 6 Bytes

When bit 58 = 1 (with the operand length and required shifting) is determined from KK, bits 57 and 56 must both be 0. When bit 58 = 0, operand length (and shifting) is then explicitly determined by bits 57 and 56.

The bit patterns are:

Displacement Shift	RNI Word Length Input KK_1	KK_0	KK or FW Select	Firmware Bits 58	57	56
None	0	0		0	0	0
Left 1 (X2)	0	1		0	0	0
Left 1 (X2)	1	0		0	0	0
Left 1 (X2)	1	1		0	0	0
None	X	X		1	0	0
RDSP 1	X	X		1	0	1
Right 3 (-8)	X	X		1	1	0
Left 1 (X2)	X	X		1	1	1

In the above list, note that bit pattern 01 is associated with R DSP. R DSP is a special case where a displacement into AR file is being furnished for the purpose of accessing the contents of a virtual address register in the program control block frame. For R DSP, the displacement factor takes on the form:

$$RDSPL = 100 + 8 \times R$$

The output of the displacement shift control circuits (signifying both frame displacement and operand length) is then sent to the "pre-adder." For RDSP, the R is determined by firmware bits 36, 35.

	Firmware Bits	
R	36	35
R_1	0	0
R_2	0	1
ALUR (RC)	1	0
ALUR (RC)	1	1

Pre-Adder

The pre-adder adds one of four constants to the supplied displacement factor.

Add Zero (+0): Adds zero to the displacement factor affecting no alterations. This is the function most commonly performed by the pre-adder when the operand is one of the standard length (one bit, or two, three, or four bytes).

Add Two (+2): Adds two (bytes) to the displacement factor. This function is used for a 6-byte operand. It serves to access the four least significant bytes first, to be followed by a subsequent accessing of the remaining two bytes. Remember that the ALU handles 32-bit, or 4-byte words, maximum.)

Add Four (+4): Adds four (bytes) to the displacement factor. This function is used when scanning multibyte character strings. It serves to access successive 4-byte segments of a multibyte character string in the forward (ascending address) direction.

Subtract Four (-4): Subtract four bytes from the displacement factor. This function is used when scanning multibyte character strings. It serves to access successive 4-byte segments of a multibyte character string in the reverse (descending address) direction.

Pre-adder operation selection is controlled by bits 55, 54, and 53 of the firmware word latched into the instruction register. Bit 55 specifies whether the pre-adder control is to be explicitly determined by bits 54 and 53 of the firmware word (bit 55 = 0) or implicitly controlled by a reference to the KK field of the current software instruction (bit 55 = 1). The KK field is a 2-bit field that will contain one of the following length indicators. Firmware bit 45 controls the addition of one (+1) to the adder independent of the constants added by the pre-adder.

A constant can be added for any address computation. The constant is selected by firmware bits 55, 54, 53, 45 and KK.

	Added Constant	Firmware Bits		
Table A		54	53	45
FW55 = 0	0	0	0	0
	1	1	1	1
	1	0	0	1
	2	0	1	1
	3	0	1	1
KK is ignored	4	1	0	0
	5	1	0	1
	-4	1	1	0
	-3	1	1	1

Table B	Added Constant	KK	KK	FW
FW55 = 1		1	0	45
	0	0	0	0
	0	0	1	0
FW 54, 53 must	0	1	0	0
be = 0	+2	1	1	0
	1	0	0	1
	1	0	1	1
	1	1	0	1
	+3	1	1	1

Displacement Adder

The output of the pre-adder is combined with the lower nine bits of the base address from the AR file yielding the lower 9 bits of the memory address. The upper 15 bits are not modified by address computation since an address computation operation spans only one frame in main memory (512 bytes). For hardware design convenience bits 0 - 11 of the address value are written back into the AR file whenever it is selected as a destination. The remaining bits (12-23) are only loaded from the CPU destination bus.

The output of the adder (bits 0-8) and the rest of the address (bits 9-23) go to CPU destination bus drivers to load to HAR1 or HAR2 as destinations of the address computation operation. The main memory address from the displacement adder is also sent to the "frame error" detection circuits. In conjunction with the KK value (a carry bit [when generated] from the displacement adder), these circuits produce a "frame error" flag whenever the main memory address (including its length) is outside of the current frame. As an example, when the main memory address is lower or higher than the 512 main memory locations containing the current frame. Frame error operation is shown in Figure 4-27.

Memory Address Destination Selection

Selecting the destination(s) for the computed main memory address is controlled by bits 39, 38, and 37 of the firmware word latched into the instruction register. The bit patterns are:

Bits 39	38	37	Memory Address Destinations
0	0	0	No destination
0	0	1	Hardware address register 1
0	1	0	Hardware address register 2
0	1	1	Hardware address register 1 and Hardware address register 2
1	0	0	AR file
1	0	0	AR file and hardware address register 1
1	1	0	AR file and hardware address register 2
1	1	1	AR file and hardware address register 1 and hardware address register 2

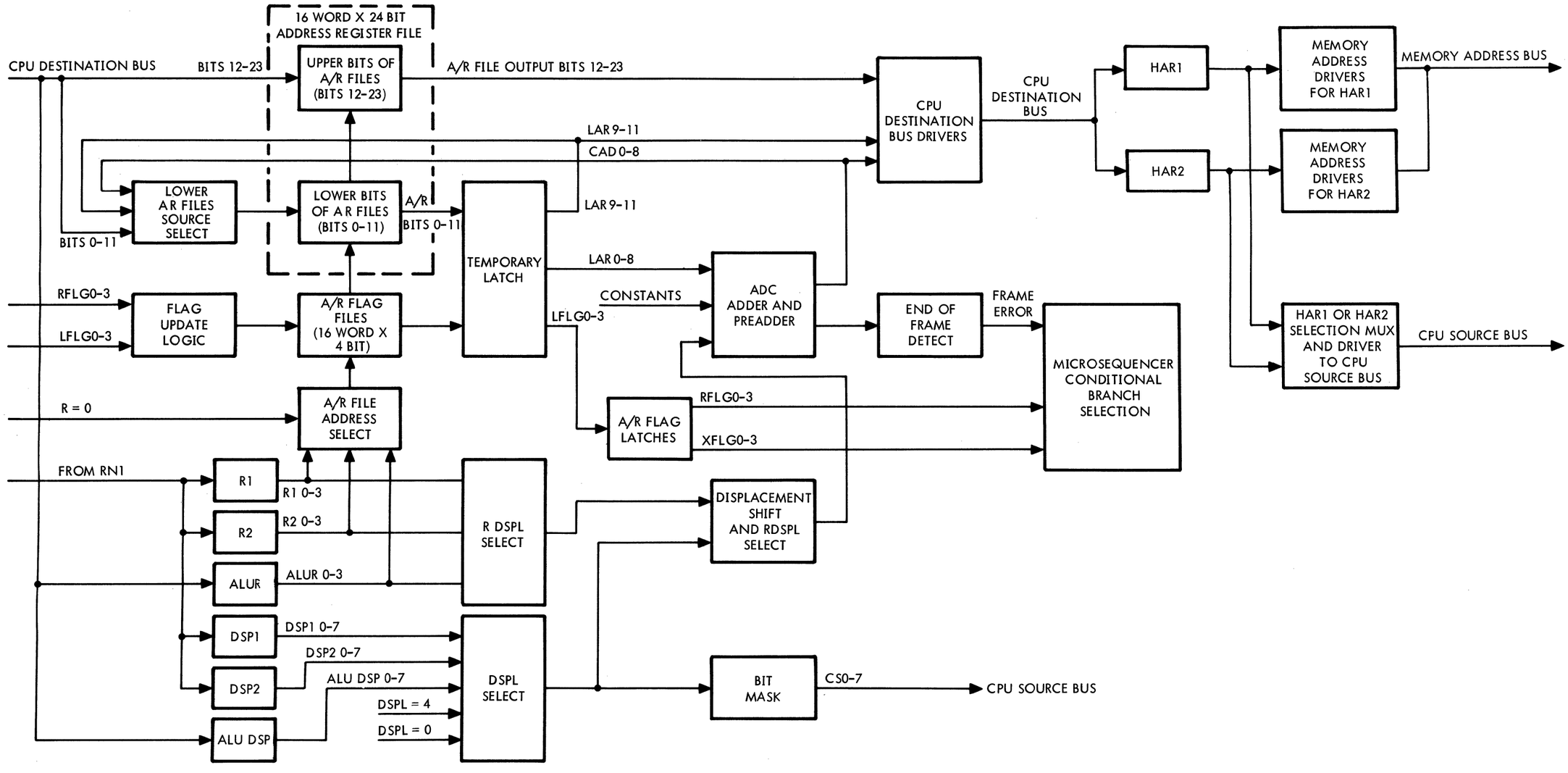
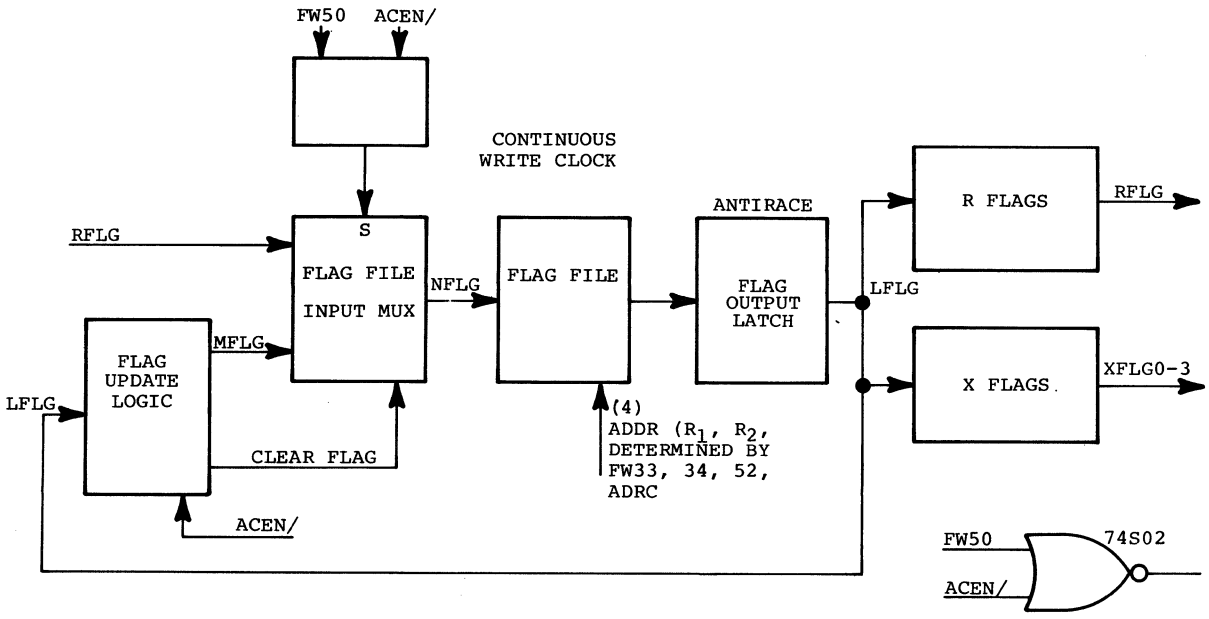


Figure 4-27. Address Computation Block Diagram

When one of the selections is made involving the AR file, it must be remembered that the computed address will be written back into the same AR file register from which the base address was extracted. Only bits 0-11 are written back.

Address Flags

The address register file consists of sixteen 24-bit address registers (refer to Figure 4-28). Associated with each of these registers (i.e., with the addresses contained in them) is a group of four flags. These flags are maintained in a register file consisting of sixteen 4-bit flag registers (see Figure 4-29). Whenever an address computation is commanded involving an address register file register, that register's corresponding flag register contents are automatically read, passed through the flag processing circuits and returned, altered or unaltered, to the same register in the flag register file.



FLAG UPDATE SELECTION

FW 49	FW 48	FW 47 | 46
0	0 NOP	FLAG
0	1 CLR	BIT
1	0 SET	ADDR
1	1 RESET	IF SET OR RESET

FLAG INPUT MUX SELECT
(FW50 = REG OR LOGIC SELECT)

TYPE A FUNCTION	ACEN/	FW50	SELECTED FUNCTION
ADD COMP	0	0	RFLGRG
ADD COMP	0	1	LOGIC
NOT ADD COMP	1	0	LOGIC
NOT ADD COMP	1	1	LOGIC

Figure 4-28. ADC Flag Function

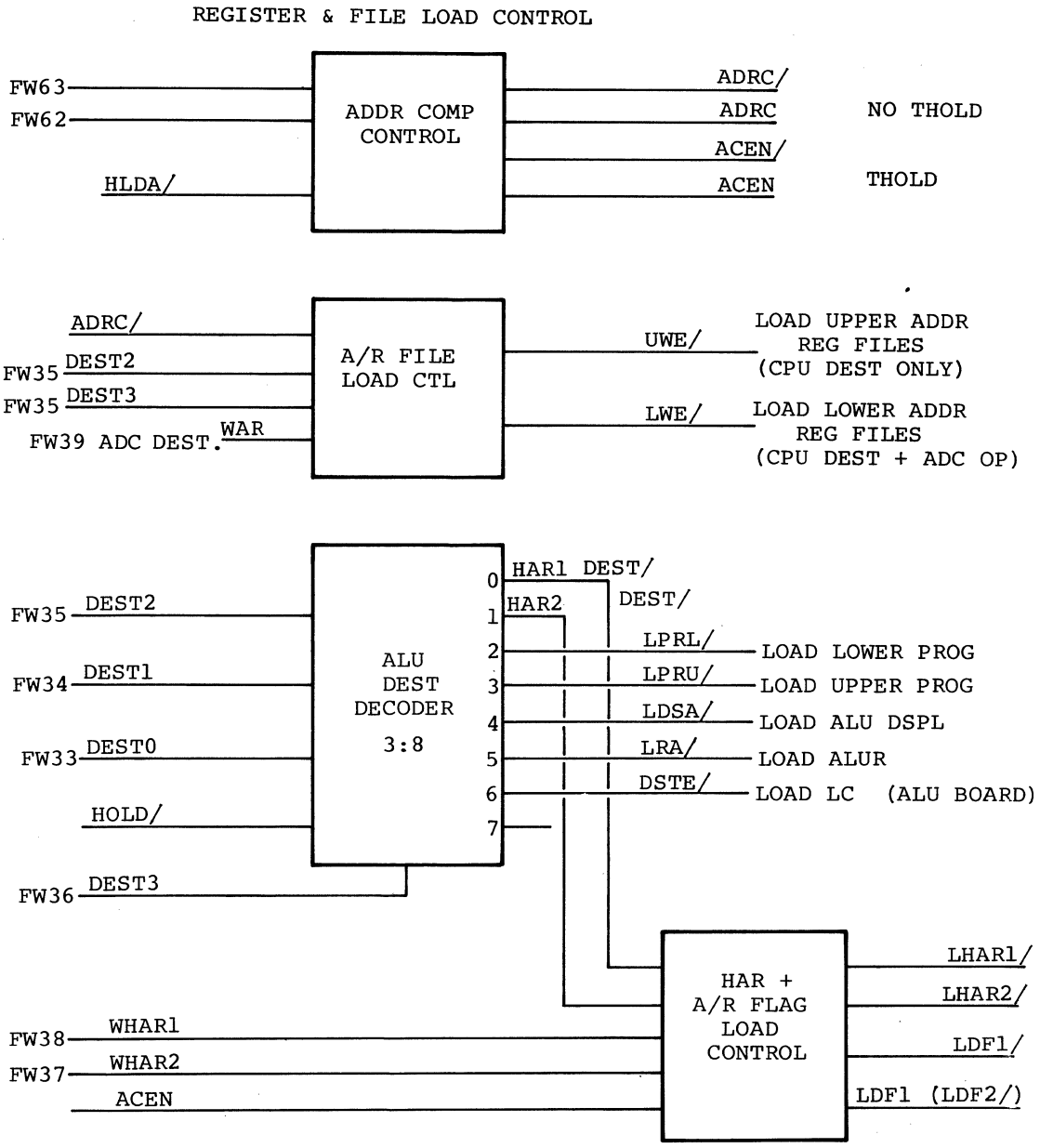


Figure 4-29. Address Comp Block Diagram

In addition, these flags can be accessed (i.e., passed through the flags processing circuits) independent of a "formal" address computation operation by "no-op"-ing the address computation part of the command. While in the flags processing circuits, the address flags can be left unaltered, set/rest individually or as a group (set/clear flags control), and/or forwarded for testing (HAR-1 flags). The flags themselves have the following meanings in the "set" (1) condition.

Flag ID	Denotation
0	Attached: Corresponding AR file register contents (address) are a main memory address.
1	Linked: Corresponding AR file register contents (address) are an address in a "linked" frame.
2	Write required: Corresponding AR file register contents (address) are an address in a frame in main memory that must be written back to disc (i.e., disc file update) before the frame can be released.
3	Currently unassigned.

Flags processing is accomplished as part of an address computation operation. Selection of the appropriate flags register (FL0-FL15) is coupled to the selection of the address register (AR0-AR15). For example, if AR7 is referenced for an address computation operation, then FL7 is implicitly controlled by bits 49, 48, and 47, 46 in conjunction. Bits 49 and 48 in the instruction register direct the setting/resetting of the flags. The bit patterns are:

Bits		Flags control
0	0	No-op (Return flags unaltered)
0	1	Clear (Reset all flags to zero)
1	0	Set (Set specified flag to 1)
1	1	Reset (Reset specified flag to zero)

Bits 47 and 46 in the instruction register are used to specify a particular flag when the set (10) or reset (11) control is indicated in bits 49 and 48.

The bit patterns are:

Bits		Active flag
0	0	0 (Attached)
0	1	1 (Linked)
1	0	2 (Write required)
1	1	Unassigned

Bit 50 in the instruction register controls a 2:1 flag file input multiplexor (refer to Figure 4-30). This multiplexor usually passes a processed flag set from the control circuits back to the flags register file register from which it was accessed (bit 50 = 1). However, it can instead pass (bit 50-1) the contents of the "NOT HAR1 FLAGS" register (RFLGO-3). This register is a holding device to allow a set of flags (accessed in a current instruction cycle) to be read back to the flags register file to a different (than the one from which it was read) register on a subsequent instruction cycle. For the instruction accessing the flag set, the flag register is specified in reference to the address register file selection in the address computation operation part of the instruction.

For the instruction returning the flag set to a different flag register, the address computation operation is "no-op"-ed except for the reference to the address register (bits 34 and 33) whose corresponding flags register is to receive the flag set.

Again, referring to Figure 4-30, when a flags register is accessed, it is sent to two of three available places: the set/clear control circuits and either the HAR1 or NOT HAR1 register is determined by that part of the associated address computations operations dealing with the destination selection. Specifically only NOT HAR1 is loaded when bit 37 of the address computation destination selective bits (39-37) is zero, and only HAR1 is loaded when bit 37 is 1. A summary of all flag update operations is shown in Table 4-30.

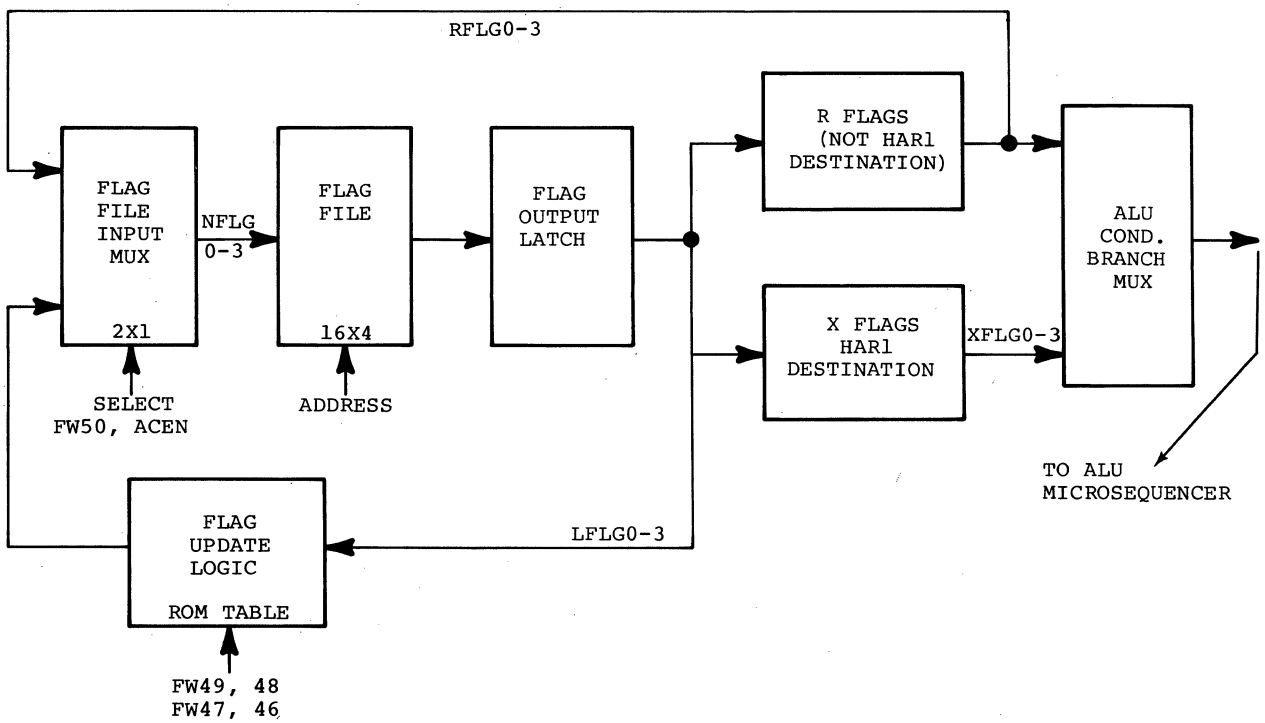


Figure 4-30. ADC Flag Logic

B-TYPE OPERATIONS

There are seven B-type operations: shift, memory read/write, multiply, divide, character test, flags control and CPU control. Refer to Figures 4-31, 4-32, 4-33, and 4-34 for a detailed breakdown of the B-type commands. There are certain restrictions in using B-type with A-type operations, i.e., not all combinations are valid. In addition, in some cases a specific B-shift statement is mandated by the preceding A-type operation, i.e., a B-type shift statement with an A-type ALU statement having an ALU output shift. These restrictions and mandates are covered in more detail as they apply.

B-Type Operation Selection

Selecting one of the seven B-type operations is controlled by bits FW32, FW31 and FW30 of the firmware word latched into the instruction register. The bits patterns are:

FW Bits			Command
32	31	30	
0	0	0	No operation
0	0	1	Memory read/write
0	1	0	CPU control
0	1	1	Flags control
1	0	0	Shift control
1	0	0	Character test
1	1	0	Divide
1	1	1	Multiply

B-Type Shift Operations

Whenever an A-type ALU operation involving the shifting of the register file and/or Q register inputs is commanded, it must be accompanied by the B-type shift operation that handles the bits being shifted in and/or out of the string being shifted. A-type shifting commands are controlled by bits FW61, FW60, and FW59 of an ALU operation, whose bits control the selection of the ALU output destinations. The following lists the A-type ALU output destination selections that involve shifting:

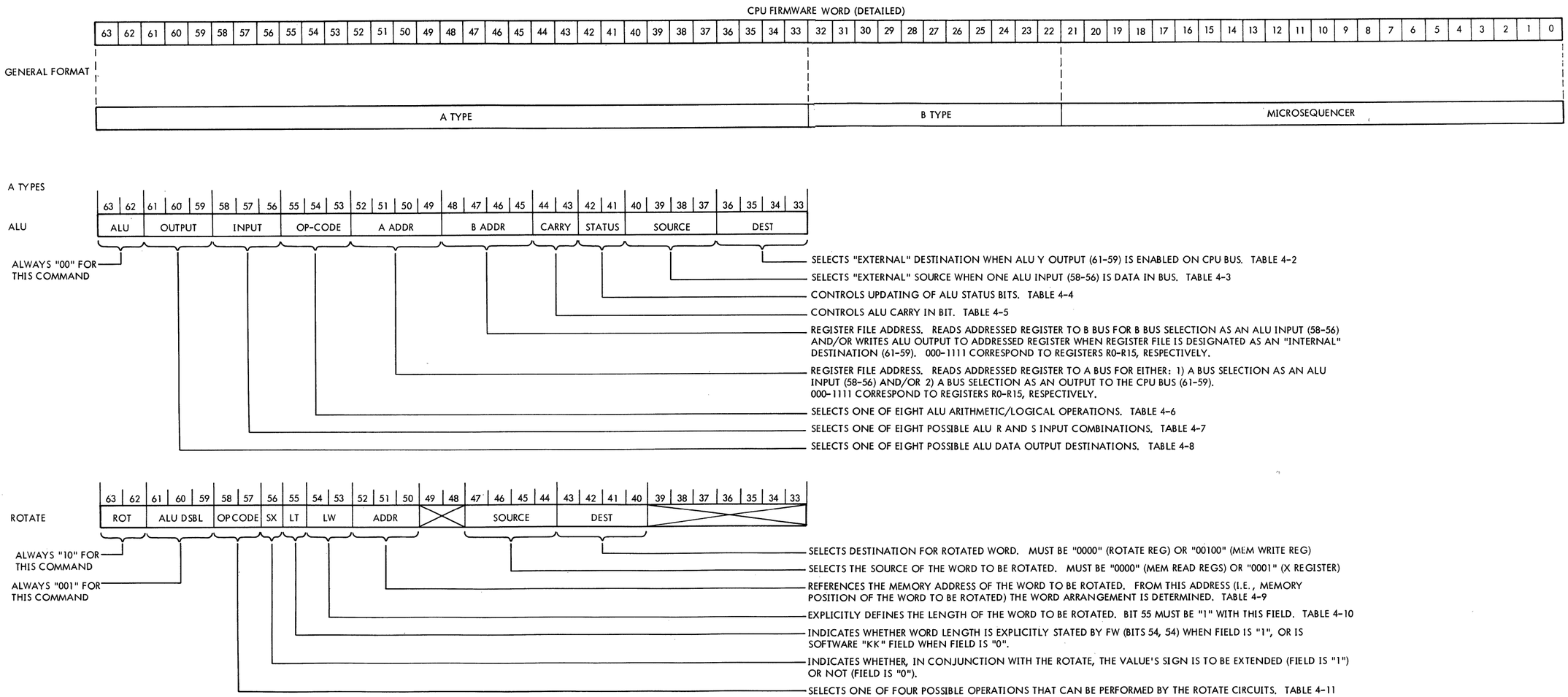
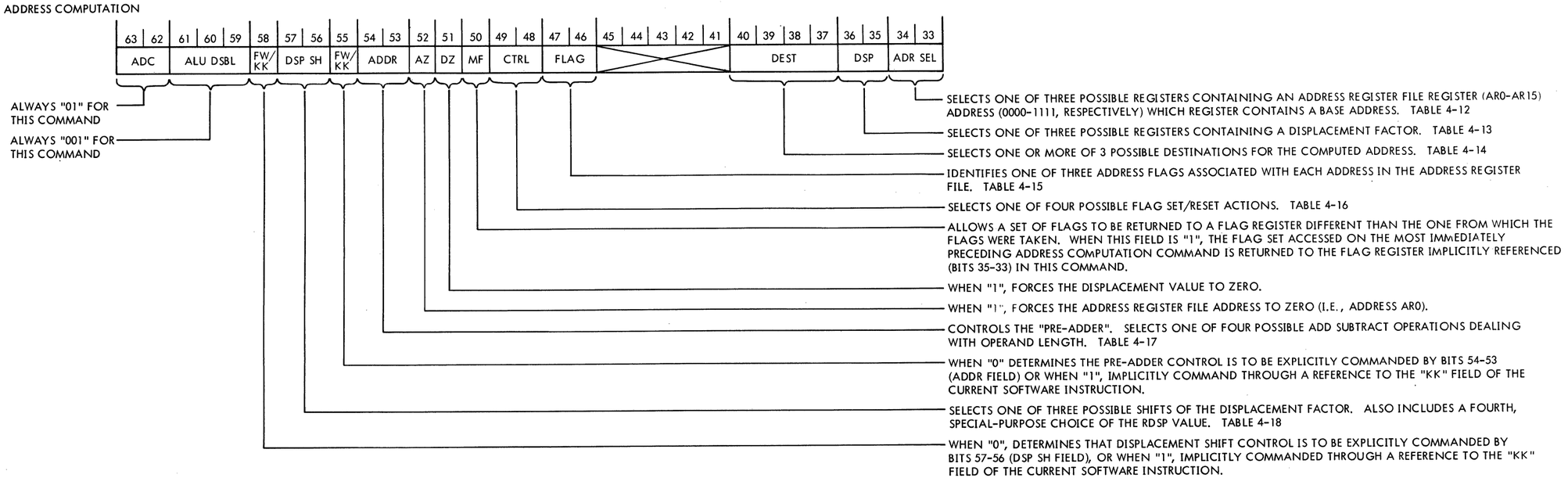


Figure 4-31. CPU Firmware Word: A Types ALU and Rotate

ADDRESS COMPUTATION



B TYPES

MEMORY READ/WRITE

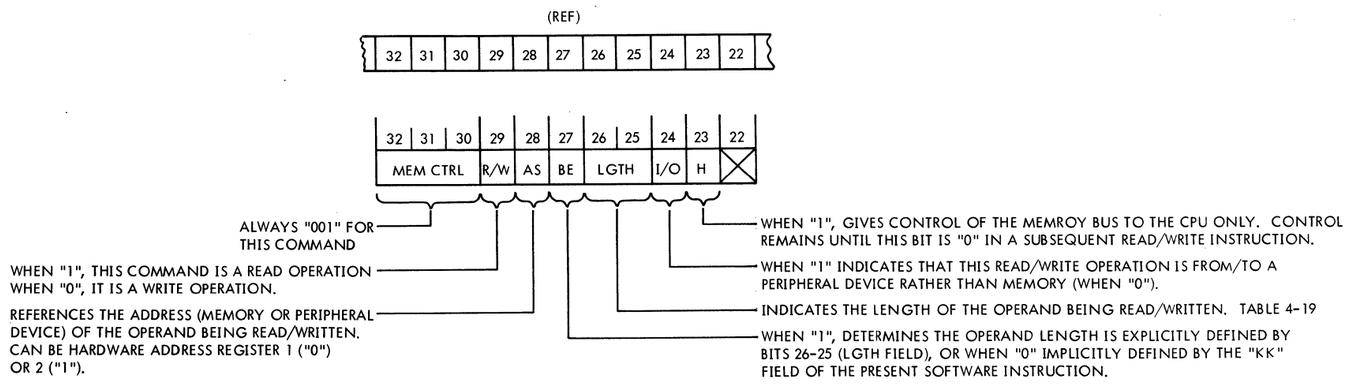


Figure 4-32. CPU Firmware Word: Address Comp and Memory R/W

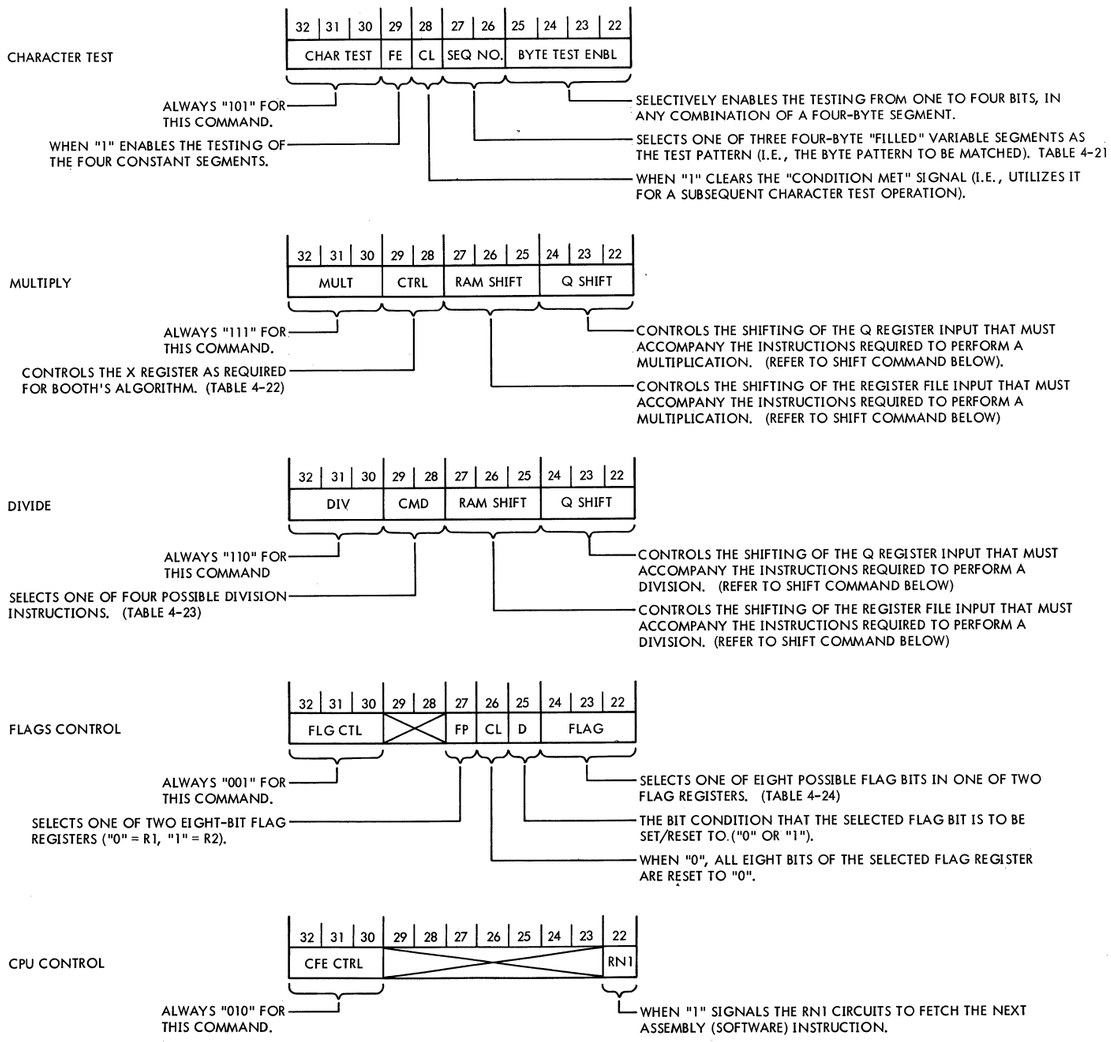


Figure 4-33. Character Test, Multiply, Divide, Flags Control and CPU Control

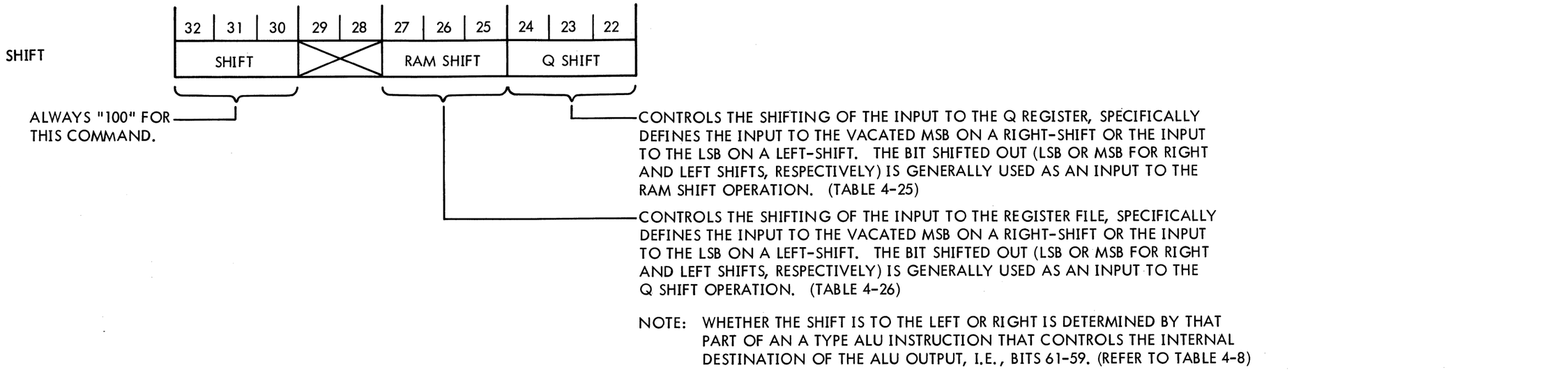


Figure 4-34. CPU Firmware Word: Shift and Microsequencer Type

A-Field FW Bits			A-Commanded Shifts
61	60	59	
1	0	0	Input to register file is shifted right 1 bit Input to Q register is shifted right 1 bit.
1	0	1	Input to register file is shifted right 1 bit.
1	1	0	Input to register file is shifted left 1 bit. Input to Q register is shifted left 1 bit.
1	1	1	Input to register file is shifted left 1 bit.

It should be noted in the above that when both the register file and Q register inputs are shifted in the same instruction, the shifts are in the same direction, both left or both right. One cannot be shifted left and the other right in the same instruction.

In any left or right 1-bit shifting operation, one bit is shifted out of the string at one end while the bit position at the other end is vacated. On a left shift, the most significant bit is shifted out while the least significant bit position is vacated, which effectively doubles the value. On a right shift, the least significant bit is shifted out while the most significant bit position is vacated, which incidentally, effectively halves the value. The bit shifted out is latched into a holding flip-flop for disposition during the current or next instruction cycle as needed.

B-type shifting control involves controlling the input to the vacated bit position (LSB on left shift, MXB on right shift). As the positions in the register file and/or Q register inputs are vacated by the A-type shift control, the B-type shift operation then loads the vacated position(s) with a predetermined bit system. Through hardware design, these latter bits include status bits (carry, sign) and the bits being shifted out of the register file and/or Q register inputs by the current shifting operation (cyclic shifts) of the ones from the previous shifting operation (linked shifts).

The input bit loading control for the register file and the Q register is controlled by bits FW27, FW26 and FW25 for the RAM shift, FW24, FW23, and FW22 for the Q shift. The two sets of firmware bits are employed in pairs in any appropriate combination. Refer to Table 4-26 for the RAM shift control functions. The hardware is implemented so that FW60 will enable the "left" shift, and FW60/ will enable the "right" shift of the 8:1 multiplexors. Refer to Table 4-25 for the Q register control functions.

The B-type shift control fields (bits FW27-FW25 and FW24-FW22) are also used in conjunction with B-type multiply and divide commands (see multiply and divide sections, respectively). These shift controls must be employed for the correct linking of the repetitive ALU functions that are invoked by the multiply and divide operations.

MEMORY READ/WRITE OPERATIONS

The CPU accesses main memory through the mechanism of the read/write operations. Data accesses to memory consist of full cycle reads or write, which take 600 nanoseconds to complete for either function. The memory address and control must be ready when the memory cycle starts for both read and write data. On a memory write, the data value can be loaded into the memory write register as late as the next firmware instruction after the write command. Refer to Figure 4-35 for an overview of the memory write control transfer function.

The memory read operation causes a 4-byte segment to be transferred from memory to the memory read register. From this register, it is routed to the ALU (where it can be used just asit came from memory) or most generally, it is routed to the rotate circuits for proper arrangement according to its length. The memory write operation causes a 1- to 4-byte segment in the memory write register to be transferred back to memory. This register is loaded from the ALU (when the ALU output can be returned to memory just as it is) or, most generally, it is loaded from the rotate circuits (counter-rotation) for proper arrangement for its return to memory. All memory reads involve four bytes, regardless of the length of the operand being accessed (1 bit, or 1, 2, 3, or 4 or more bytes). Refer to Figure 4-4.

The memory data address is contained in one of two hardware address registers (HAR), designated HAR1 and HAR2. These can be changed to another address on the third command following the start of the memory command without losing the address associated with the start memory command. The HAR can be set on the same command as the memory command. On a read memory operation, the data is automatically loaded into the memory read register on the fourth clock after the start memory pulse, so it is available for use by the CPU on the fifth clock.

HAR		KK or FWWDL		Byte Length	Write Byte Enables				Location Where Data Is Written Into Memory
1	0	1	0		3	2	1	0	
0	0	0	0	1	1	0	0	0	X
		0	1	2	1	1	0	0	X X
		1	0	4	1	1	1	1	X X X X
		1	1	3	1	1	1	0	X X X
0	1	0	0	1	0	1	0	0	X
		0	1	2	0	1	1	0	X X
		1	0	4	1	1	1	1	X X X X
		1	1	3	1	0	1	1	X X X
1	0	0	0	1	0	0	1	0	X
		0	1	2	0	0	1	1	X X
		1	0	4	1	1	1	1	X X X X
		1	1	3	1	0	1	1	X X X
1	1	0	0	1	0	0	0	1	X
		0	1	2	1	0	0	1	X X
		1	0	4	1	1	1	1	X X X X
		1	1	3	1	1	0	1	X X X

Figure 4-35. Memory Write ControlTransfer Function

On a write data command, each of the four byte fields in the memory can be separately enabled for a write, thus providing the capability to write 1 to 4 bytes in memory. The selected address register HAR points to the most significant byte of the data. The memory has a split field address bus that causes memory bytes to the left of the most significant byte to be addressed in the

next higher double word location of the memory. This makes it possible to read or write a full 32 bit double word in one memory cycle regardless of the byte address value, as long as the entire double word to be accessed is in the same 512 byte frame of memory.

Memory addresses are computed by the address computation logic, under direct control of the firmware, or can be loaded directly from the CPU into either memory address register. The rotate circuits are then employed, when required, to arrange the operand in the proper configuration for its length.

Memory writes involve only the exact number of bytes comprising the operand being committed to memory. The hardware address registers (HAR1 and HAR2) again provide the address where the operand will begin to be written, the most significant byte. The data being written is first loaded into the memory write register via the CPU destination bus either from the ALU or from the rotate circuits. With the write, all four bytes of the memory write register are impressed on the memory bus, but only the specific number of bytes comprising the operand are written to memory. Refer to Figure 4-36 for an overview of the memory read/write operation.

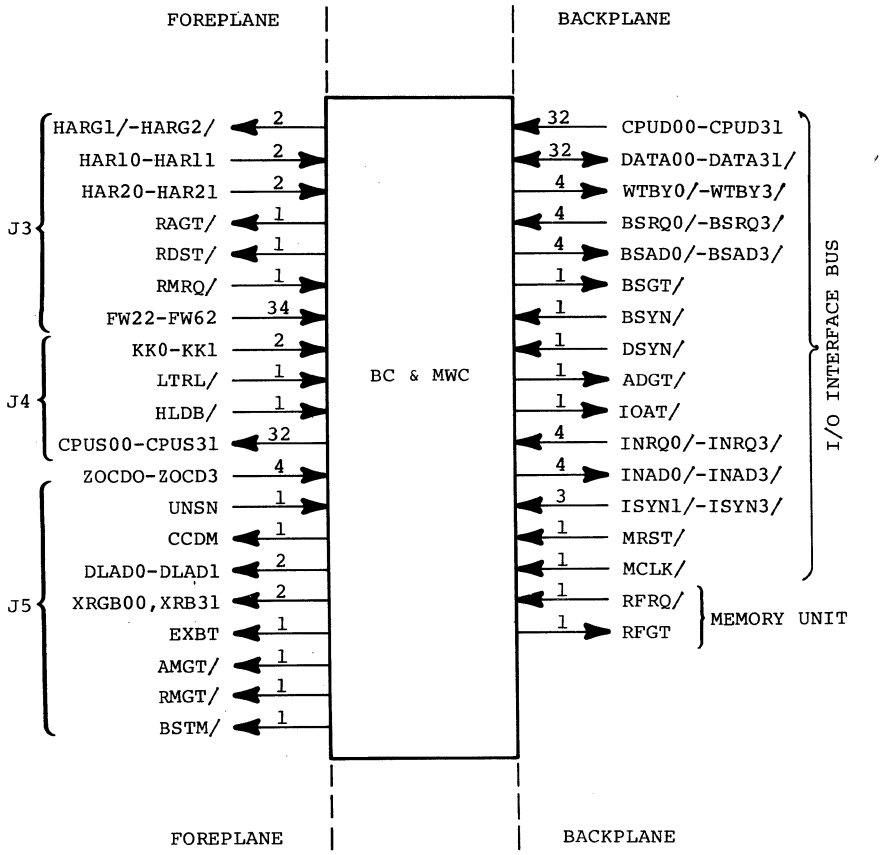


Figure 4-36. Bus Controller and Memory Word Control Interface diagram

Selection of a read/write operation is controlled by a "001" in bits FW32, FW31 and FW30 of the firmware word latched into the instruction register (refer to Figure 4-32). Selection of a read or write is controlled by bit FW29, where "0" commands a write and a "1" commands a read. Bit FW28 references the memory address of the data being read or written. When bit 28 equals "0," the memory address of concern is contained in hardware address register 1, when bit FW28 equals "1," the address is in hardware addresses register 2.

The length of the operand read from or written to memory is specified implicitly (bit FW27 equals zero) or explicitly (bit FW27 equals 1). When bit FW27 is zero, operand length is referenced from the "KK" field of the assembly instruction software. When bit FW27 equals 1, the operand length is specified by bits FW26 and FW25. The bit patterns are:

FW Bits		Operand Length
26	25	
0	0	1 byte
0	1	2 bytes
1	0	4 bytes
1	1	3 bytes

By way of reminder, the "KK" length specifiers are:

"KK"		Operand Length
0	1	1 byte
0	1	Word (or tally i.e., 2 bytes
1	0	Double word (or tally) i.e., 4 bytes
1	1	Triple word (or tally) i.e. 6 bytes (this will be interpreted as 3 bytes by the memory write control)

Memory Write Byte Enable

Four lines are provided to the memory to enable writing of any combination of bytes. These lines can be controlled directly by firmware, or the firmware can specify hardware control of write byte enable, according to the following:

Write Byte Enables

(a)	KK	HAR2	01
(b)	KK	HAR2	01
(c)	FW WD Length	HAR1	01
(d)	FW WD Length	HAR2	01
(e)	Direct firmware length		

Write byte enables and byte locations where data is written into memory are shown in Figure 4-35.

Bit FW24 of the firmware word latched into the instruction register is the input/output (I/O) attention flag. When this bit equals one, it indicates that data on the memory bus is being communicated to/from a peripheral device rather than memory. For a write operation (bit FW29 = 0), data is being transferred from the memory write register to a specified peripheral device. For a read operation (bit FW29 = 1), data is being transferred from a specified peripheral device to the memory read register. In both operations, the device is specified by the lower eight bits of HAR1 and HAR2 (as determined by bit FW28), whose eight bits constitute the device address.

Bit FW23 of the firmware word latched into the instruction register is the hold bus command (HDBS). When this bit equals one, the firmware directs that no other system component except the CPU (i.e., ALU) can gain control of the memory bus until the firmware "releases" it (with bit FW23 = 0) in a subsequent B-type memory command firmware word. During the control period, "burst mode" transfers to/from the CPU from/to memory of peripheral devices can occur. Also during the control period, the memory unit can "access" (but not control) the memory bus for the purpose of a "refresh memory."

CHARACTER TEST OPERATIONS

The character test operations permit the searching of memory for specific characters, or bytes. (Character and byte are synonomous in this contest.) The searching is accomplished by making a comparison between a pre-determined byte pattern and memory data in 4-byte segments in ascending (forward) or descending (backward) address order.

The byte patterns that can be searched consist of four constants representing delimiters, and three optional variable patterns supplied by the firmware. By design, special purpose hardware is implemented in the system that performs the comparisons for the matching of memory data with the constants.

The search (i.e., matching) methodology for the constants is as follows: The memory data being searched is retrived from memory in a 4-byte segment and loaded into the memory read register. This register is then simultaneously compared via the CPU source bus with each of the four constants. A match with each of the target bytes is compared.

The search (i.e., matching) methodology for the variable is as follows: The one to three target variables are "filled" into one to three file registers (i.e., each byte is duplicated four times). The memory data to be searched is retrieved from memory

in a 4-byte segment and loaded into the memory read register. Then the memory read register contents are compared via the CPU source bus with one, two, and/or three of the variables in the character test hardware for determination of a match.

The match test is done on a subsequent firmware clock. When a match occurs, a "conditional-met" signal is set along with the byte address (0-3) of the memory byte involved in the match. If more than one byte of the 4-byte memory segment matches the target byte, the condition-met signal and address are returned only for the "high-order" byte that made the match (i.e., the most significant matched byte in a forward search [ascending memory address order]) or the least significant matched byte in a backward search (descending memory order).

Character testing is controlled by a mask, which is set by the ALU. Refer to Figure 4-37 for the mask format. Notice that the first three outputs VRMB1, VRMB2 and VRMB3 are used to interrogate variable patterns. The terms FXMB1, FXMB2, FXMB3 and FXMB4 are used to test for "fixed patterns." The term "TORF" makes the decision to test for either a matched or non-matched condition (true or false).

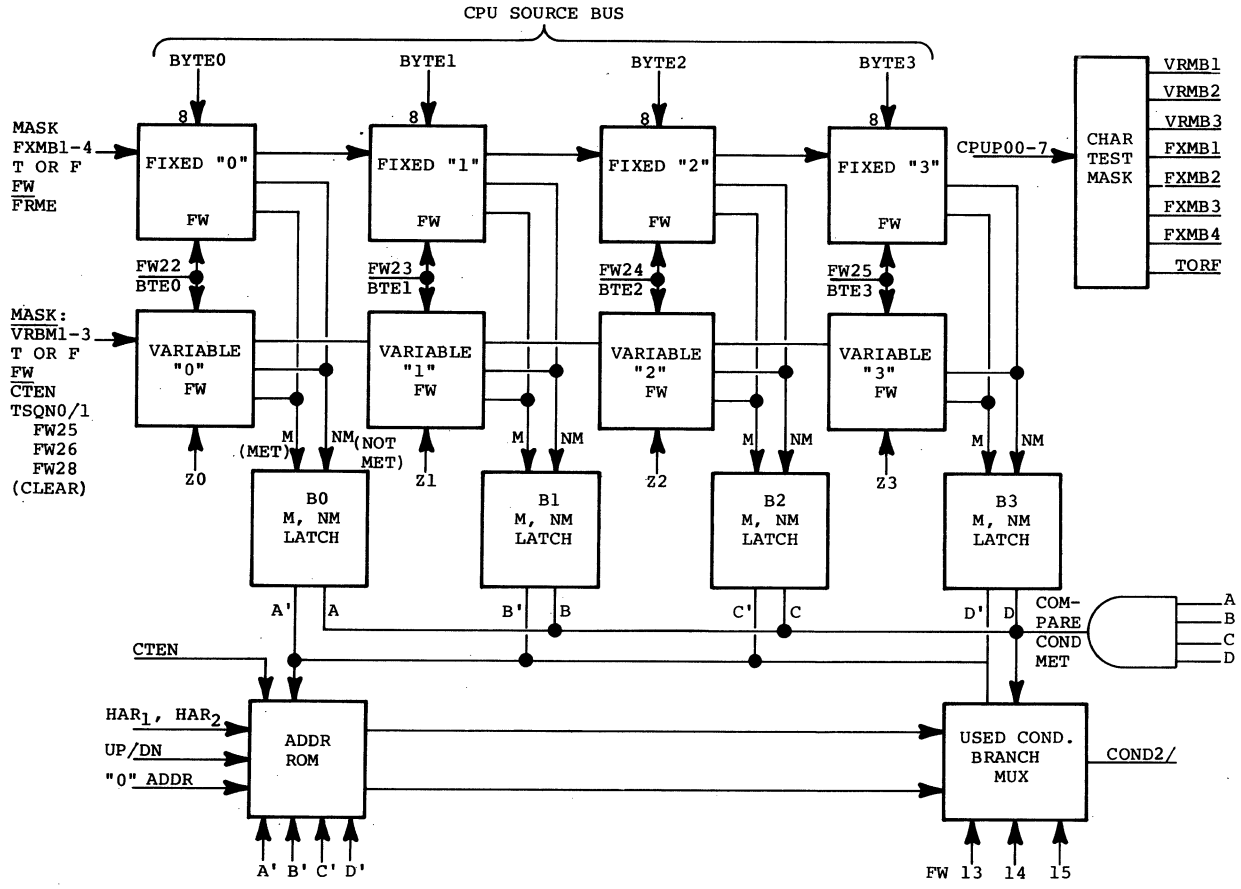


Figure 4-37. Character Test Block Diagram

A B-type character test operation is commanded by FW32, FW31/ amd FW30, respectively, of the firmware word latched into the instruction register. Testing of the memory data against the four constants is controlled by bit FW29. When bit FW29 = 1, the constants are tested for a match against memory data supplied to the memory read register on a previous instruction cycle. A variable to be tested for a match is controlled by bits FW27 and FW26. The bit patterns are:

Bits		Variables to be Used in Searching for a Match
27	26	
0	0	NOP
0	1	SC0
1	0	SC1
1	1	SC2

Note that in the same character test command (firmware word), all the constants can be tested along with one of the variables. When two or three of the variables are to be used in the test, the second and third variable must each be specified in a subsequent firmware word.

In the previous discussion, it was stated that all four bytes of memory data were simultaneously compared with four identical target bytes. Additionally, it is possible to specify in the firmware which bytes of memory data (from 1-4 in any combination) are enabled for testing. This is specified by bits FW25, FW24, FW23, and FW22. The bit patterns are:

FW Bits				Memory Data Bytes To Be Tested			
25	24	23	22	Byte3	Byte2	Byte1	Byte0
0	0	0	0	X	X	X	X
0	0	0	1	X	X	X	
0	0	1	0	X	X		X
0	0	1	1	X	X		
0	1	0	0	X		X	X
0	1	0	1	X		X	
0	1	1	0	X			X
0	1	1	1	X			
1	0	0	0		X	X	X
1	0	0	1		X	X	
1	0	1	0		X		X
1	0	1	1		X		
1	1	0	0			X	X
1	1	0	1			X	
1	1	1	0				X
1	1	1	1				

Bit FW28 of the firmware word is provided to initialize (i.e., reset) the condition-met signal. This signal should be reset prior to a character test operation. The reset is commanded when bit FW28 = 1.

> NOTE: This bit cannot be used to clear the condition-met signal in the same instruction cycle that calls for a character test.

MULTIPLY OPERATION

As described in the ALU section, the arithmetic logic unit performs only two basic mathematical functions, addition and subtraction, besides its shifting and logical functions. Therefore, the process of multiplication must be affected with these limited capabilities that imply a lengthy series of instructions. However, there exists a method where the number of required steps can be lessened. To permit use of this algorithm, special purpose hardware, in conjunction with a B-type command, is provided. For this usage, the "X" register is equipped with a 1-bit-right-shift capability and an extension bit latch which is a 1-bit holding device to receive the "X" register's right-most bit when this register's contents are successively shifted right one bit during the execution of the algorithm. A simplified functional block diagram of this multiplication implementation is shown in Figure 4-38. The methodology involved in Booth's algorithm can be briefly expressed as follows:

The multiplicand is placed in an ALU file register.

The multiplier is placed in the "X" register.

> NOTE: Multiplicand and multiplier are both 32-bit words. If either value was less than 32 bits, it would have been filled by the "sign extend" function of the rotate circuits.

One or more ALU file registers are cleared to receive the partial product (ultimately, the full product at the end of the operation). Then, the multiply set instruction (with appropriate ALU commands and shift intructions) is executed 32 times with each instruction execution, the "X" register is shifted right one bit with the extension bit playing the key role in the operation.

Multiplication Using Booth's Algorithm

Booth's algorithm was implemented to multiply two N-bit two's complemented numbers yielding a 2N-1 bit two's complement result. No correction cycle is required for negative numbers. Multiplication using this algorithm is implemented using the operand register that may be shifted right, and a single-bit extension on the least significant end of the operand register.

For a 32-bit by 32-bit multiply yielding a 64-bit result (duplicated sign bit) the steps are as follows:

(a) Place multiplier in operand register ("X" Reg), and clear extension bit.

(b) Place multiplicand in a reserved ALU file register.

(c) Multiply step (see Table 4-32) 32 times.

(d) Store the partial results in the ALU file register.

For a 64-bit by 64 bit multiply yielding a 128-bit result, the steps are as follows:

(a) Place lower 32-bits of the multiplier in operand register. Clear extension bit.

(b) Multiply step (see Table 4-32) 32 times.

(c) Place upper 32-bits of multiplier in operand register. Do not clear extension bit.

(d) Multiply step (see Table 4-32) 32 times.

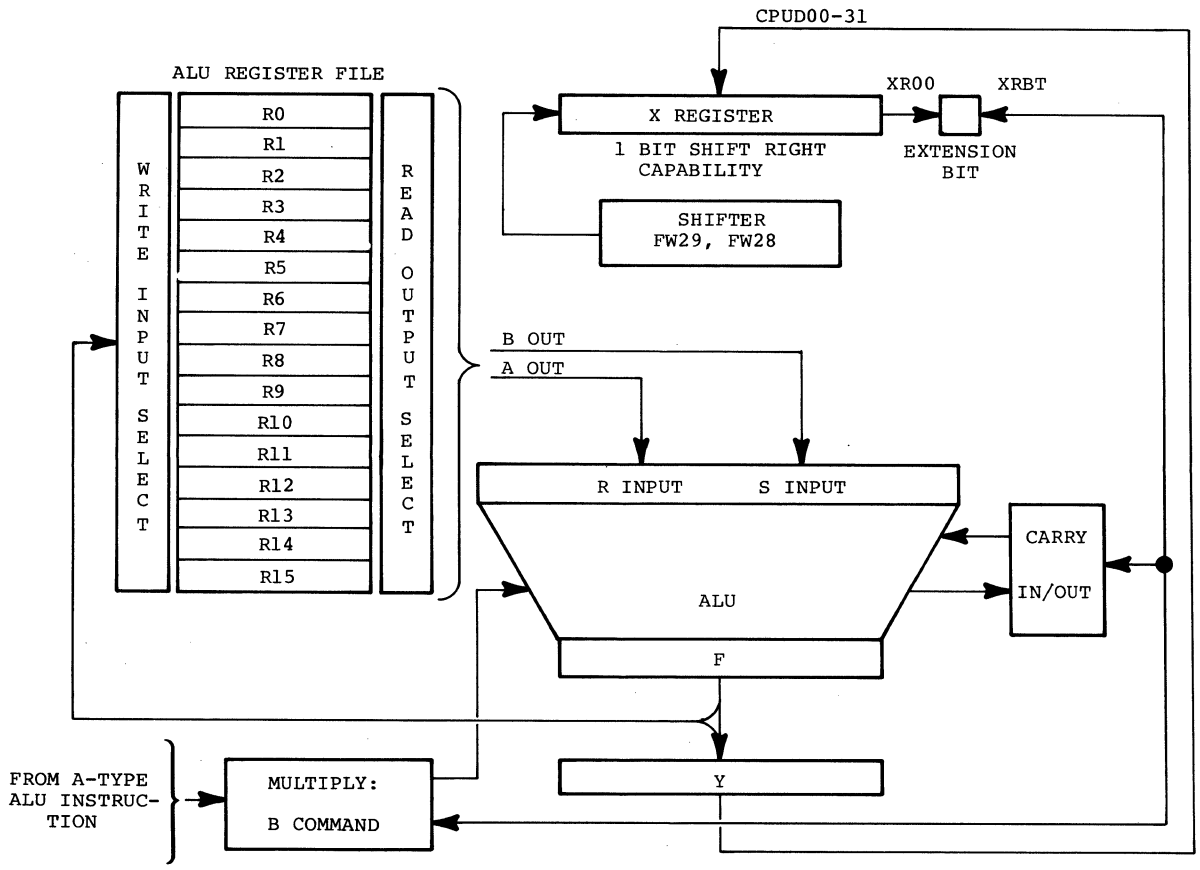


Figure 4-38. Multiply Block Diagram

TABLE 4-32

MULTIPLICATION EXAMPLE USING BOOTH'S ALGORITHM

Multiplicand	Product	Multiplier, Ext.	Operation
0100	00000000	1001,0	
	-0100		1,0 - Subtract
	11000000		
	11100000	0100,1	Shift
	+0100		0,1 - Add
	00100000		
	00010000	0010,0	Shift
	00001000	0001,0	0,0 - Shift only
	-0100		1,0 - Subtract
	11001000		
	11100100	000,1	Shift

0100 = +4

1001 = -7

11100100 = -28

Similar procedures may be used for other multiplier/multiplicand lengths. The multiply step used is a conditional instruction that executes differently depending upon the value of the least significant bit of the operand register and the extension bit. The following example lists the operation(s) performed for the four combinations of these two bits:

XR00	EXBT	Operation (modified Type A Function CTL)
0	0	Arithmetic shift right only
0	1	Add multiplicand then arithmetic shift right
1	0	Subtract multiplicand then arithmetic shift right
1	1	Arithmetic shift right only

Addition or subtraction of the multiplicand is done with the most significant portion of the product. At the same time the partial product is shifted right (always arithmetic) the operand register and extension bit are shifted right. For multiple word multiplicands, the multiply step is executed multiple times per multiplier bit. The operand register shift is therefore not automatic. The following example illustrates the algorithm by multiplying two four-bit numbers:

(+4 times -7 yielding -28)

Multiply Implementation

The B-type multiply function is complemented by a modified A-type ALU operation. The A-type commands affected by multiply are the source and function control. The two specific firmware bits are FW47 and FW53. The equations for these two control terms are:

$$*INST1 = FW57+MUL.FW29.\overline{XROO \oplus EXBT}$$

$$INST3 = FW53+MUL.FW29.XROO.\overline{EXBT}$$

The table for XROO (OPRO) and EXBT represents the effect of altered A-type operations. This modification occurs when the B-type command is in multiply step and firmware bits FW53, FW54, and FW55 are zero. Refer to Table 4-6 of the ALU operation.

The ALU input control selection is changed as indicated by the equation for INST1. Firmware bits FW57 and FW58 are zero and FW56 is one. Refer to Table 4-7 of the ALU input control to verify the operation.

The carry control operation is modified by the multiply function with a term called "SUBR." The equation for "SUBR" is:

SUBR = MUL.FW29.XROO.EXBT+DIV.QUOT

The "carry control" firmware bits FW43 and FW44 are zero during this multiply step. The carry out line "CO" will be the same as the "SUBR" term. When SUBR = 1, CC will = 1. Refer to Table 4-5 for the carry control operation.

Figure 4-39 depicts the "hardware multiply routine." This is an extract from the firmware diagnostic program, and should prove useful in understanding the multiply function.

---

*See Figure 4-40 for logic breakdown.

```
                                *
                                *
                                *
                                *    HARDWARE MULTIPLY ROUTINE
                                *
                                *
                                *    WR0: OPERAND
                                *    WR1: ACCUMULTOR
                                *    WR2,3: RESULT
                                *
                                *
                                *
                                *
                                * PERFORM MULTIPLY
                                *
                                * 32 BY 32 BIT MULTIPLY
                                * WR2,3 = (WR0 * WR1)
                                *
0047                            SB.HMULT  MOVE   WR1>XREG          MULTIPLIER IN XREG
0047 0C62000600028000                     ADV    *                 ********** SEPARATE TWO INSTRUCTIONS **********
0048 08000001C0C28000                     MUL    RESET             CLEAR EXTENSION BIT
0049 1A804000000028000                    ZERO   WR2               CLEAR PRODUCT UPPER
004A 1AR060000000028000                   ZERO   WR3               CLEAR PRODUCT LOWER
004B 0F6000FCFFFFFFE0                     MOVE   -32>LOOPCTR       LOOP COUNTER

                                *
                                * MULTIPLY LOOP
                                *
004C                            MUL.LOOP  ADD    WR0,WR2>WR2       MULTIPLY
004C 19004001F0028000                     MUL    SHIFTX            DO MULTIPLY/SHIFT/ADD OPERATION
004D                                      MOVE   WR2>WR2
004D 2664400010C028000                    SHFR   SIGN              ARITHMETIC RIGHT SHIFT ON PRODUCT
004E                                      MOVE   WR3>WR3
004E                                      SHFR   WLNK              LINKED RIGHT SHIFT ON PRODUCT LOWER
004E 2C666001040560 4C                    BF     LOOPCTR,MUL.LOOP  BRIF NOT DONE YET
                                *
                                * LOOP DONE
                                * PRODUCT IN WR2,WR3
                                *
004F 1C6640000002 8000                    MOVE   WR2>WR2           SHOW MS
0050                                      MOVE   WR3>WR3           SHOW LS
0050 1C666C000012A000                     RTN    *                 RETURN
                                *
                                *
                                *
                                *
                                *
```

Figure 4-39. Hardware Multiply Routine

DIVIDE OPERATION

The same limitations that applied for the multiply function affect the divide operation. The divide operation was implemented within the capabilities of the arithmetic logic unit. There are four B-type special purpose commands that are used in conjunction with the ALU commands to create a series of instructions that will permit a division operation.

The methodology involved in the divide function can be briefly expressed as follows:

The dividend and divisor (sign-extended as required) are placed in the ALU register file. The dividend must always have twice the number of bits as the divisor. For example, when the divisor has 32 bits and occupies one register, the dividend has 64 bits occupying two file registers. Then, the most significant 32 bits of the divisor are copied from the ALU register file to the "X" register for the purpose (only) of having the "X" register's most significant bit occupied by the divisor's sign bit. One or more ALU file registers are cleared to receive the quotient. The divisor is taken from its register, two's complemented, and the two's complemented number returned to the same (divisor) register. The "divide test" instruction is then performed on the dividend.

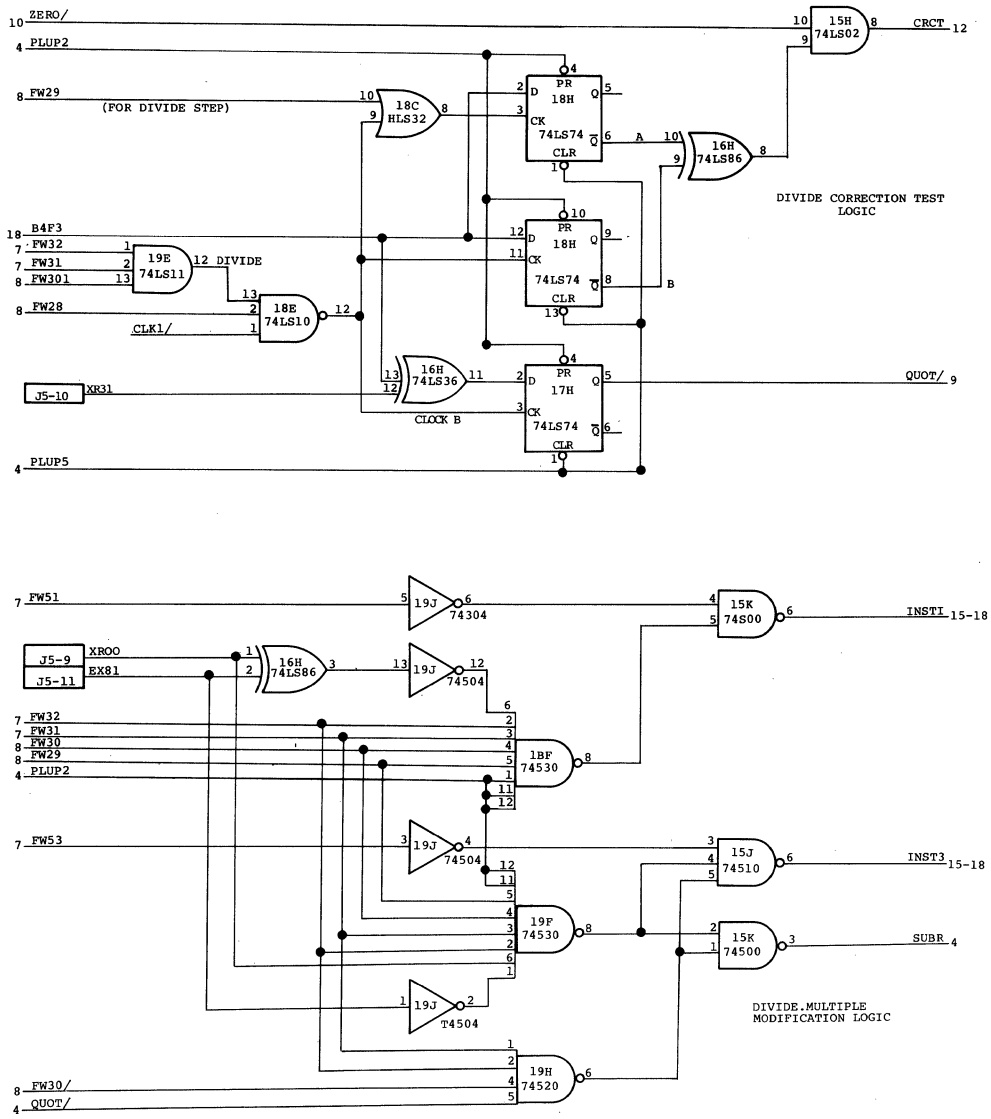


Figure 4-40. Multiply/Divide Modification Logic

The division procedure now enters a loop that is executed "N" times where "N" is the number of bits in the divisor. The loop consists of the following steps:

(a) Shift dividend left one bit.

(b) Shift quotient left one bit.

(c) Perform "divide quotient" instruction. This produces one bit of the answer (i.e., quotient). This bit placed in the LSB of the ALU file register assigned to the quotient.

(d) A "divide step" instruction is performed "N" times, where "N" is the number of "words" minus one of the quotient.

(e) A "divide test step" instruction is performed, ending the loop.

As stated previously, this loop is performed the same number of times as there are bits in the divisor. With each pass, one bit of the quotient is produced, from MSB to LSB, as defined in step (c). Each time this bit is produced, it is loaded into the LSB of the receiving register in conjunction with a 1-bit left shift of this register's contents. When the procedure ends, the full quotient resides in the receiving register.

For some cases in the divide procedure a "correction" (determined by hardware) is required. This correction involves passing through the "loop" one additional time and making the correction.

The hardware divide function is accomplished by performing an A-type function with modifications provided by the B-type divide command. Refer to Figure 4-41 for an overview of the B-type divide command. As shown, the divide command is invoked when firmware bits FW32, FW31, and FW31/ are raised simultaneously. This now permits other special purpose divide hardware to function during this time. A breakdown of the special divide commands is given in Table 4-23. Firmware bits FW28 and FW29 perform the initialization, arithmetic modification and finalization of the divide operation. The divide command and these mode terms are used to develop unique control functions. There are four control terms developed for divide which are:

QUOT/ = B4F3+XR31 latched

INST3 = DIV.QUOT/+FW53

SUBR = DIV.QUOT/+FW53

CRCT (See Table 4-33 and Figure 4-40.)

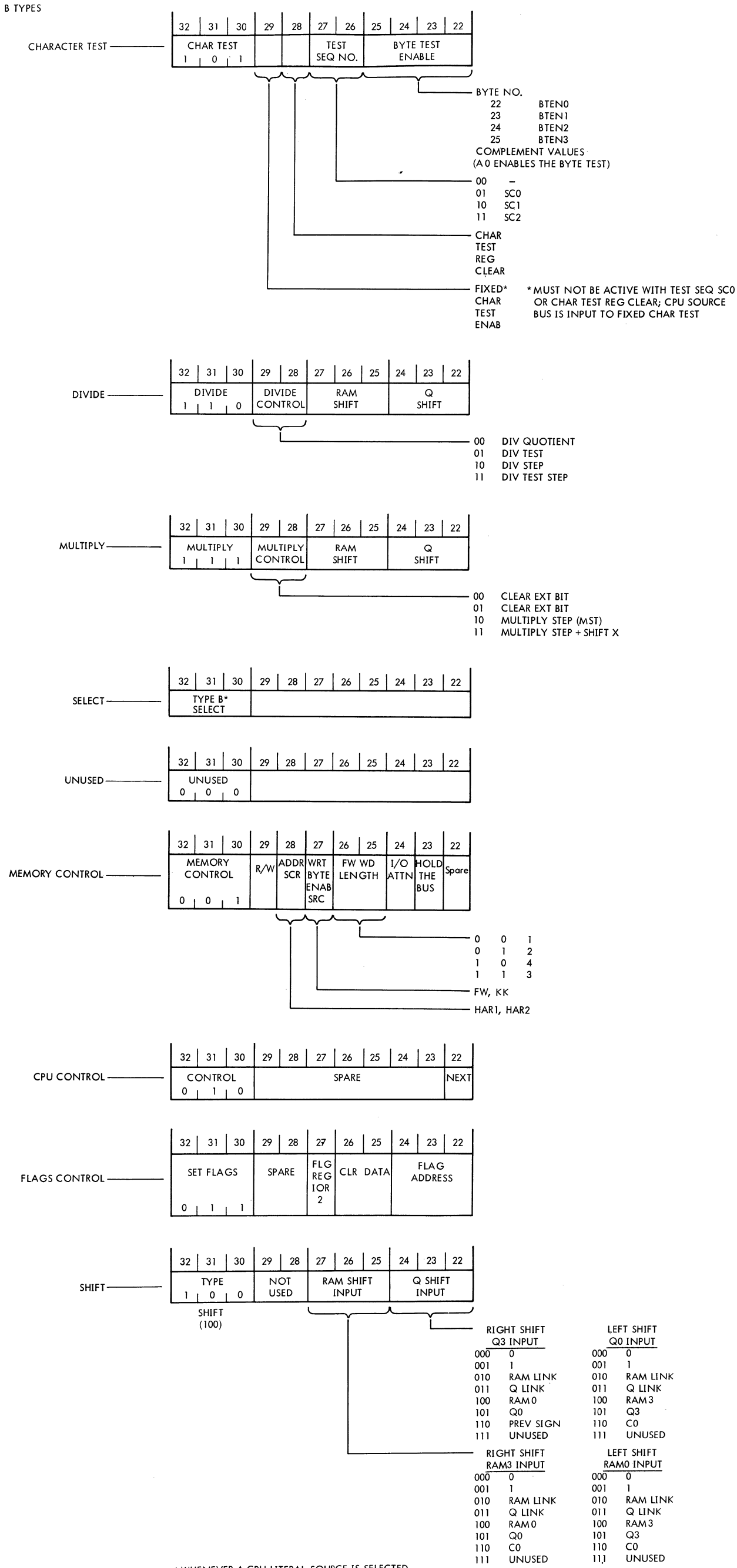
The quotient term "QUOT/" is data dependent. The data terms acting upon this are B4F3, the ALU sign bit, and XR31, the sign bit of the divisor stored in the "X" register. As indicated by the equations, QUOT/ is integral to developing SUBR and INST3.

The term "INST3" is used to modify the A-type ALU function. The ALU function operation is given in Table 4-6. Firmware bits FW53, FW54, and FW55 are all zero. The function is altered from command "000" to "001" by INST3.

During divide the ALU control for the "carry-in" logic is provided by the control line, "SUBR." Refer to Table 4-5, which is the carry control table. Firmware bits FW43 and FW44 are both zero during this modified ALU operation. The carry-out line "CO" will change from command "00" to "01" by SUBR.

Provision for correction to the divide process is provided by the CRCT control function. The CRCT set-up is shown in Table 4-33. As indicated on this table, the A and B outputs are clocked

Figure 4-41. B Type Divide Command Overview



during "divide test only" function. This will set up a sign reference for the ALU output. When the "divide test step" function is engaged, the A output is unaffected while the B output will be clocked through. The test for CRCT correction is enabled after the total divide iterations are completed.

TABLE 4-33

CRCT TEST WITH ZERO/ = 1

B4F3	$\overline{B4F3}$	Device Function	FW 29	FW 28	Clock A	Clock B	A	B	CRCT
0	1	Divide Test Only	0	1	1	1	1	1	0
1	0	Divide Test Only	0	1	1	1	0	0	0
0	1	Divide Test Step	1	1	0	1	0	1	1
0	1	Divide Test Only	0	1	1	1	1	1	0
1	0	Divide Test Step	1	1	0	1	1	0	1

CRCT=ZERO/ $\overline{F3}$ Clocked on Divide Test Only $\oplus$ $\overline{F3}$ Clocked on Divide Test or Divide Test Step

NOTE: Refer to Figure 4-40.

Divide Procedure

(a) Divide test.

(1) Establishes sign of the dividend, (2) determines the first quotient bit.

(b) Shift dividend (partial remainder) left one bit; also shift quotient.

(c) Divide quotient step.

(1) Updates quotient by addition or subtraction of 1; (2) for extended length quotients, divide step must be used for upper words of quotient.

(d) Divide step.

Updates the remainder by adding or subtracting the 2's complement of the divisor.

(e) Determine the next quotient bit.

(f) Repeat steps (a) through (e), a total of "N" times.

(g) Determine if divide correction is needed.

(1) Correct if dividend sign does not equal remainder sign unless the remainder sign is zero; (2) correct if remainder equals divisor; (3) correct if remainder equals the negative of the divisor.

(h) If correction is required, repeat steps (c) and (d).

(i) Overflow is not detectable.

Hardware Divide Subroutine

The "hardware divide subroutine" is shown in Figure 4-42. This extract from the firmware diagnostic programs should provide adequate information to fully understand the hardware

```
                          *     HARDWARE DIVIDE SUBROUTINE
                          *
                          *
0051 1C74800000002F000  SB.HDIV   MOVE   WR10>WR4          LOAD OPERAND
0052 1C76C0000002B000             MOVE   WR11>WR6          LOAD ACCUMULATOR
0053 1F6120E000000001             MOVE   1>WR9             USED TO DIV STEP QUOTIENT
0054 1A80A0000002B000             ZERO   WR5
0055 1C6CC2000002F000             MOVE   WR6>WR6,(U)       SET ACCUMULATOR SIGN
                          *
                          * WR4 = DIVISOR (TO BE COMPLEMENTED)
                          * WR6,WR5 = DIVIDEND
                          * WR7 = QUOTIENT
                          *
0056                              ZERO   WR7               CLEAR QUOTIENT
0056 1A80E0000004C058             BT     NEG>DIV.1         IF NEGATIVE, OVER
0057 0800000000002A050            B      DIV.10            ELSE DIV.10
0058 1820A0000002B000   DIV.1     DEC    WR5               MAKE WR5 = -1
0059 0C6800060002B000   DIV.10    MOVE.  WR4>XREG
005A 1C4888D000028000             SUB    ZERO,WR4>WR4      COMPLEMENT DIVISOR
005B                              OR     WR6,ZERO,(U)      RESET FLAGS
005C 0C6C020190028000             DIV    TEST              SET DIVIDEND SIGN AND REMAINDER SIGN
005C 0F6000FCFFFFFFE0             MOVE   -32>LOOPCTR
                          *
005D                      LOOP    MOVE   WR6>WR6
005D 3C6CC00100028000             SHFL   ZERO              SHIFT DIVIDEND LOWER
005E                              MOVE   WR5>WR5
005E 3C6AA00104028000             SHFL   WLNK              SHIFT DIVIDEND UPPER
005F                              MOVE   WR7>WR7
005F 3C6EE00100028000             SHFL   ZERO              SHIFT QUOTIENT
0060                              ADD    WR7,WR9>WR7
0060 1912E001A0028000             DIV    STEP              ADD/SUB 1 TO QUOTIENT
0061                              ADD    WR5,WR4>WR5,(U)
0061                              DIV    TESTSTEP          ADD/SUB (-DIVISOR) TO REMAINDER

0061 1908A2018005805D             BF     LOOPCTR,LOOP      BRIF NOT DONE
                          *
                          * TEST FOR CORRECTION
0062 0800000000028000             NOP    *          **********TIMING NO-OP**********
0063                              ADD    WR5,WR4,(U)       NO DESTINATION.  CHECK FOR BOTH ZERO.
0063                              DIV    STEP              TEST FOR CORRECTION
0063 090A8201A0070065             BT     CRCT>CORRECTION   BRIF CORRECTION
0064                              NOPA
0064 0800000000052067             BF     ZERO>END          IF CORRECTION NEEDED, FALL THROUGH
                          *
0065                      CORRECTION ADD   WR7,WR9>WR7
0065 1912E001A0028000             DIV    STEP              ADD/SUB 1 TO QUOTIENT
0066                              ADD    WR5,WR4>WR5
0066 1908A001A0028000             DIV    STEP              ADD/SUB (-DIVISOR) TO REMAINER
                          *
                          * QUOTIENT IN WR7, REMAINDER TO WR8
                          *
0067                      END     MOVE   WR5>WR8           REMAINDER
0067 1C6B00000012A000             RTN    *                 RETURN TO CALLER
                          *
                          *
                          *              FIGURE 5.6.2
                                  END
```

Figure 4-42. Hardware Divide Subroutine

implementation of the divide function. For ease in understanding the divide algorithm, refer to the examples provided in Table 4-34.

FLAGS CONTROL

There are two 8-bit registers used to hold special purpose flags, flag register 1 and flag register 2. These flags are:

Flag Register 1		Flag Register 2	
Register Bit Position	Flag Name	Register Bit Position	Flag Name
0	Flag 0	8	Flag 4
1	Flag 1	9	Flag 5
2	Flag 2 (Step/)	10	Flag 6
3	Flag 3	11	Flag 7
4	Diagnostic 1	12	Flag 8
5	Up-Down Scan	13	Flag 9
6	ANDRO	14	Flag 10
7	Diagnostic 2	15	Flag 11

The B-type flags control function serves to set/reset these flags. This function is commanded by FW32/, FW31, and FW30, respectively. Bit FW27 then selects which of the two flag registers is being altered. Bit FW27 = 0 selected flag register 1, bit FW27-0 selected flag register 2. Bit FW26 serves as a full register clear command, i.e., when FW26 = 0, all eight bits of the selected register are reset to zero. When bit FW26-1 then only one flag bit in the register is to be changed. The specific bit that will be changed is selected by bits FW24, FW23, and FW22 of the instruction register.

The bit patterns are:

Bits			Flag Register 1	Flag Register 2
			Flag Selected	Flag Selected
24	23	22	(Bit 27=0)	(Bit 27-1)
0	0	0	Flag 0	Flag 4
0	0	1	Flag 1	Flag 5
0	1	0	Flag 2	Flag 6
0	1	1	Flag 3	Flag 7
1	0	0	Diagnostic 1	Flag 8
1	0	1	Up/Down Scan	Flag 9
1	1	0	ADRO	Flag 10
1	1	1	Diagnostic 2	Flag 11

TABLE 4-34

DIVIDE ALGORITHM EXAMPLES

Example: +29 - +4 = +7 r +1

0100	00011101	0000	q = 1
	00111010	0000	Shift left
	1100	+1	-Divisor, +1
	11111010	0001	q = -1
	11110100	0010	Shift left
	0100	-1	+Divisor, -1
	00110100	0001	q = 1
	01101000	0010	Shift left
	1100	+1	-Divisor, +1
	00101000	0011	q = 1
	01010000	0110	Shift left
	1100	+1	Divisor, +1
	00010000	0111	Result +7 r +1
	+1	+7	

Example: +29 - +7 = +4 r +1

0111	00011101	0000	q = 1
	00111010	0000	Shift left
	1001	+1	-Divisor, +1
	11001010	0001	q = -1
	10010100	0010	Shift left
	0111	-1	+Divisor, -1
	00000100	0001	q = 1
	00001000	0010	Shift left
	1001	+1	-Divisor, +1
	10011000	0011	q = -1
	00110000	0110	Shift left*
	0111	-1	+Divisor, -1
	10100000	0101	q = -1
	0111	-1	Remainder sign dividend sign
	00010000	0100	Result +4 r +1
	+1	+4	

*Note that the sign bit changes with this shift. No significance is lost, however.

Example: -10 - -4 = +2 r -2

1100	11110110	0000	q = 1
	11101100	0000	Shift left
	0100	+1	-Divisor, +1
	00101100	0001	q = -1
	01011000	0010	Shift left
	1100	-1	+Divisor, -1
	00011000	0001	q = -1
	00110000	0010	Shift left
	1100	-1	+Divisor, -1
	11110000	0001	q = 1
	11100000	0010	Shift left
	0100	+1	-Divisor, +1
	00100000	0011	q = -1*
	1100	-1	Remainder sign dividend sign
	11100000	0010	Result +2 r -2
	-2	+2	

*Note that the q bit for the correction changes from the previous q bit.

Example: +12 - +4 = +3 r 0

0100	00001100	0000	q = 1
	00011000	0000	Shift left
	1100	+1	-Divisor, +1
	11011000	0001	q = -1
	10110000	0010	Shift left
	0100	-1	+Divisor, -1
	11110000	0001	q = -1
	11100000	0010	Shift left
	0100	-1	+Divisor, -1
	00100000	0001	q = 1
	01000000	0010	Shift left
	1100	+1	-Divisor, +1
	00000000	0011	Result +3 r 0

Whatever flag is selected and whatever condition it is in (0 or 1), it is then set/reset to match the contents of bit FW25 of the instruction register. Bit FW25 is referred to as the "data" bit.

CPU CONTROL

The B-type CPU control operation currently uses only four bits of the B field of the instruction registers. Firmware bits FW32/ FW31, and FW30/, respectively, command the function, and bit FW22 is used to signal the RNI circuits that the firmware is ready for the RNI to fetch the next assembler (software) instruction. A description of the RNI circuits appears later in this section.

When the ALU firmware execution is completed, the microprocessor signals the picoprocessor that it is ready to form another firmware instruction (request next instruction) and awaits the picoprocessor's instruction register full signal (IRF).

MICROSEQUENCER OPERATIONS

Microsequencer operations deal with firmware program control, i.e., the order in which firmware instructions are executed. There are three basic types of microsequencer operations:

Next Instruction: Causes the next sequential firmware instruction to executed next.

Unconditional Branch: Causes a jump to an instruction other than the next sequential instruction.

Conditional Branch: Specifies a "condition" to be tested and based on the results of that test:

(1) When the "condition is met, it causes a jump to an instruction other than the next sequential instruction.

(2) When the "condition" is not met, it causes the next sequential firmware instruction to be executed next (sometimes referred to as a "fall-through.").

A simplified functional block diagram of the microsequencer is shown in Figure 4-43.

FIRMWARE PROMS AND FIRMWARE ADDRESSING

The set of firmware instructions is contained in a group of proms that jointly provide storage for up to 8192 (i.e., 8K) individually addressable 64-bit firmware instructions. Instruction

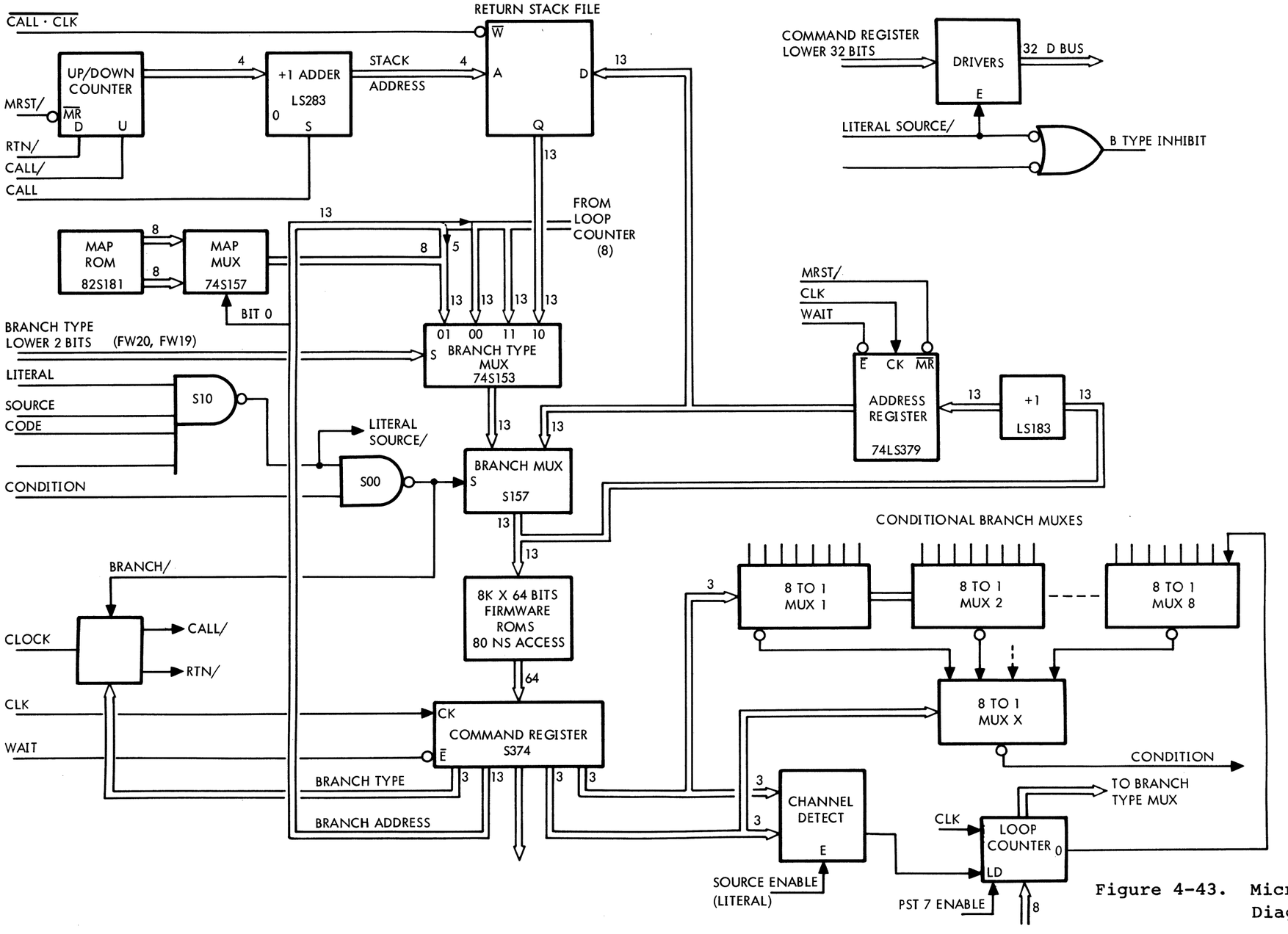


Figure 4-43. Microsequencer Block Diagram

addresses are provided by a 2:1 multiplexor whose inputs are the next sequential instruction address or a branch address.

Control of this multiplexor is through bits FW18 through FW13 of the firmware word latched into the instruction register. These six bits are organized into two, 3-bit octal groups that address one of 64 available "condition" channels (octal 00 through 77). With two exceptions, each of these channels represents a condition that can be true (1) or false (0).

When a conditional branch is specified, one of the "conditional" channels is addressed (via 64:1 multiplexing circuits), passing that channel to the 2:1 multiplexor. If the address condition channel is true, the 2:1 multiplexor passes the branch address. If the addressed condition channel is false, the 2:1 multiplexor passes the next sequential address.

Control of the 4:1 branch address multiplexor is affected with the lower two bits (FW20 and FW19) of the 3-bit command type field of the firmware word latched into the instruction register. The bit patterns are:

Bits		
20	19	Branch address selected
0	0	Firmware branch address
0	1	Mapped jump
1	0	Return address
1	1	Table jump

Bits FW20 and FW19, coupled with bit FW21 constitute the command type field of the microsequencer instruction part of the firmware word. Note the relationships with the branch address selection bit patterns given just previously. The bit patterns for this field are as follows:

Bits			
21	20	19	Microsequencer command type
0	0	0	Branch
0	0	1	Map
0	1	0	Return (with stack "pop")
0	1	1	Register
1	0	1	Unassigned
1	0	0	Call (with stack "push")
1	1	0	Unassigned
1	1	1	Unassigned

There is another "default-type microsequencer command type that is used when an A-type ALU operation specifies a "microprogram literal" as an ALU input, 1111 in bits FW40 - FW37, respectively. In this case the entire microsequencer field (bits FW21 - FW00) coupled with the entire literal. When this occurs, the microsequencer circuits are automatically forced to provide the incremented address as the next firmware instruction address.

The two conditional channel exceptions mentioned in the preceding paragraph deal with a "hard-wired" branch and no-branch condition. Channel 24 is always false and when addressed, forces the next address through the 2:1 multiplexor. Channel 25 is always true and when addressed, forces the branch address through the 2:1 multiplexor. The bit patterns for all 64 channels are shown in Table 4-29.

> NOTE: As indicated in Table 4-29, the channel to test the loop counter is 77 (octal). The counter is incremented automatically during the test cycle.

NEXT BRANCH ADDRESS

The next sequential address is developed by tapping the "current" address on its way to the prom addressing circuits and incrementing it by 1. Refer to Figure 4-43. The incremented address is then stored in the address register for use in the next cycle. In addition to being sent to the 2:1 multiplexor, the incremented address is also sent to a file register referred to as a "push-pop" stack. This will be described in the Return Address Stack paragraph.

The branch address input to the 2:1 multiplexor comes from the 4:1 multiplexing circuits that will pass one of four possible branch addresses.

- Return: The return address from the "push-pop" stack.
- Firmware branch ddress: An address provided by the microsequencer instruction (bits FW12-FW00 of the firmware word).
- Table jump: Consisting of the upper five bits of the firmware branch address (bits FW12-FW8) coupled with eight bits from the ALU (via the loop counter), provides the facility for a "computed go-to."

  NOTE: In this usage, the loop counter functions solely as a convenient data link between the CPU bus and the 4:1 branch address multiplexor).
- Mapped jump: Consisting of the upper five bits of the firmware branch address (bits FW12-FW8) coupled with

eight bits from the RNI-prom maps circuits; provides the capability for a mapped jump. As defined in the RNI discussion, there is a choice of first or second map branching. This selection is controlled by FW00.

RETURN ADDRESS STACK

The return address stack is a "push-pop" stack in which the last (i.e., most recently entered) address in ("pushed") is the first address out ("popped"). This stack is used for the storage and retrieval of addresses when "call" and "return" paired commands are used for subroutine linkage.

When a "call" is made, the next (i.e., incremented) address is "pushed" into the stack while a branch is made to the called subroutine. The last instruction in a called subroutine is always a "return" instruction which when encountered, causes the "pushed" address to be "popped" out of the stack and sent to the prom addressing circuits as a branch address.

This effectively brings program execution back to the instruction immediately following the "call" when subroutine execution is completed. The "return" address stack is 16-addresses deep, allowing that many levels of "nested" subroutine calls.

Control of the "return" address stack is effected through bits FW21, FW20, and FW19 of the firmware word latched into the instruction register. These bits control both stack "pushing" and "popping" and as required, the appropriate selection of the throughput of the 4:1 branch address multiplexor. The bit patterns were given in the preceding paragraph.

READ NEXT INSTRUCTION (RNI) FUNCTION

The function of the RNI picoprocessor is the fetching from memory of an instruction stream and the reorganizing (parsing) of this instruction into a controlled, well-ordered function for execution by the main processor.

The logic implementing this function consists of the picoprocessor sequencer, control store, memory fetch and instruction register control logic, memory data register, program address registers, double ranked instruction register and the instruction mapping proms. Refer to Figure 4-44 for a function flow.

PICOPROCESSOR SEQUENCER AND CONTROL STORE

The control store consists of 1024 32-bit words of read only memory. Since the sequencer can only address 256 words, the control store appears as four separate files. One contains the

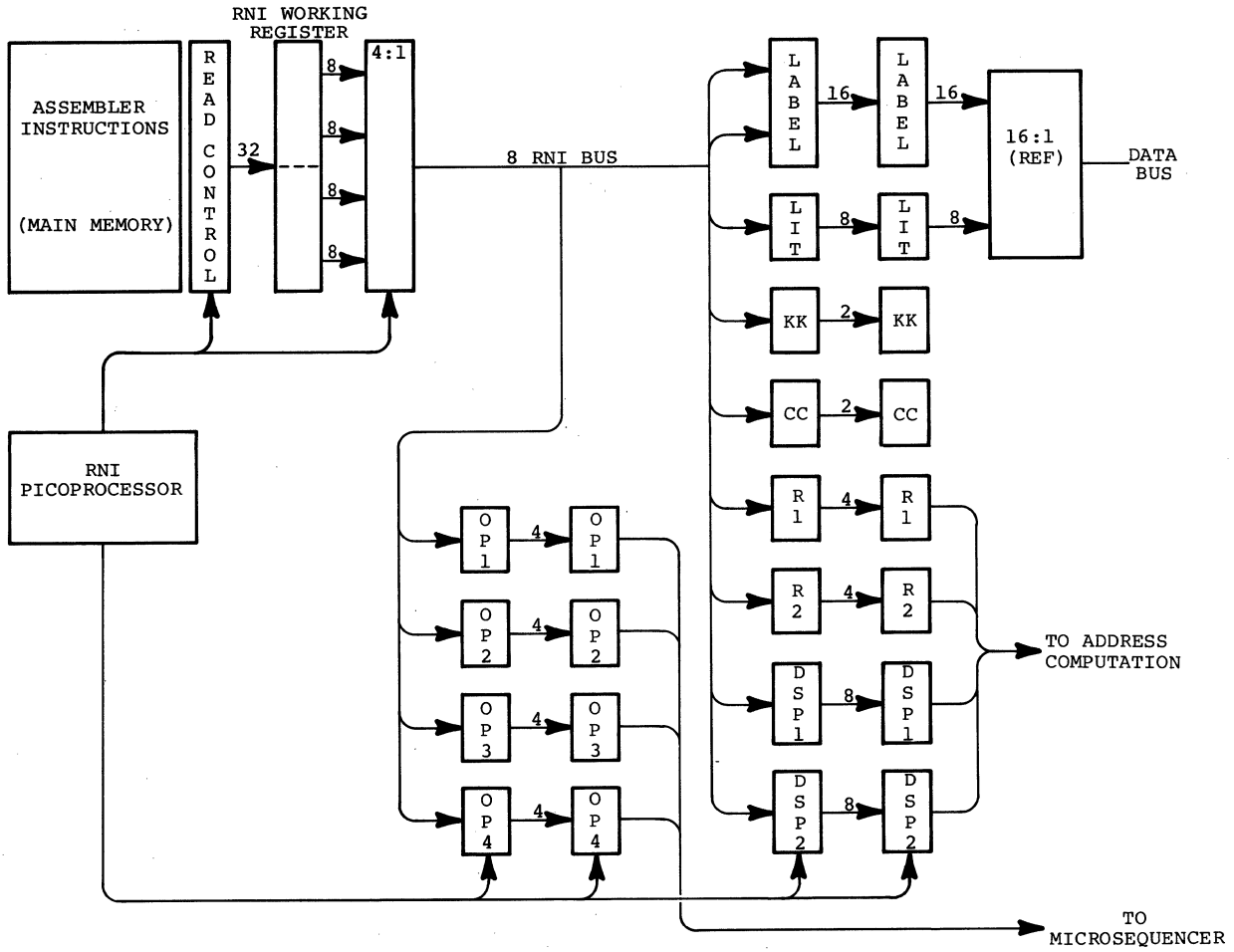


Figure 4-44. RNI Block Diagram

normal system RNI firmware and the other three may contain diagnostic or other RNI firmware. Each control store word consists of a 4-bit field used for overall control, (see memory fetch and instruction register control logic), a 12-bit field used for instruction register loading (see discussion of instruction register), a 3-bit field presently unused and a 13-bit field used for sequencer control. The breakdown of the control store word is illustrated in Figure 4-45.

The sequencer control field consists of four subfields:

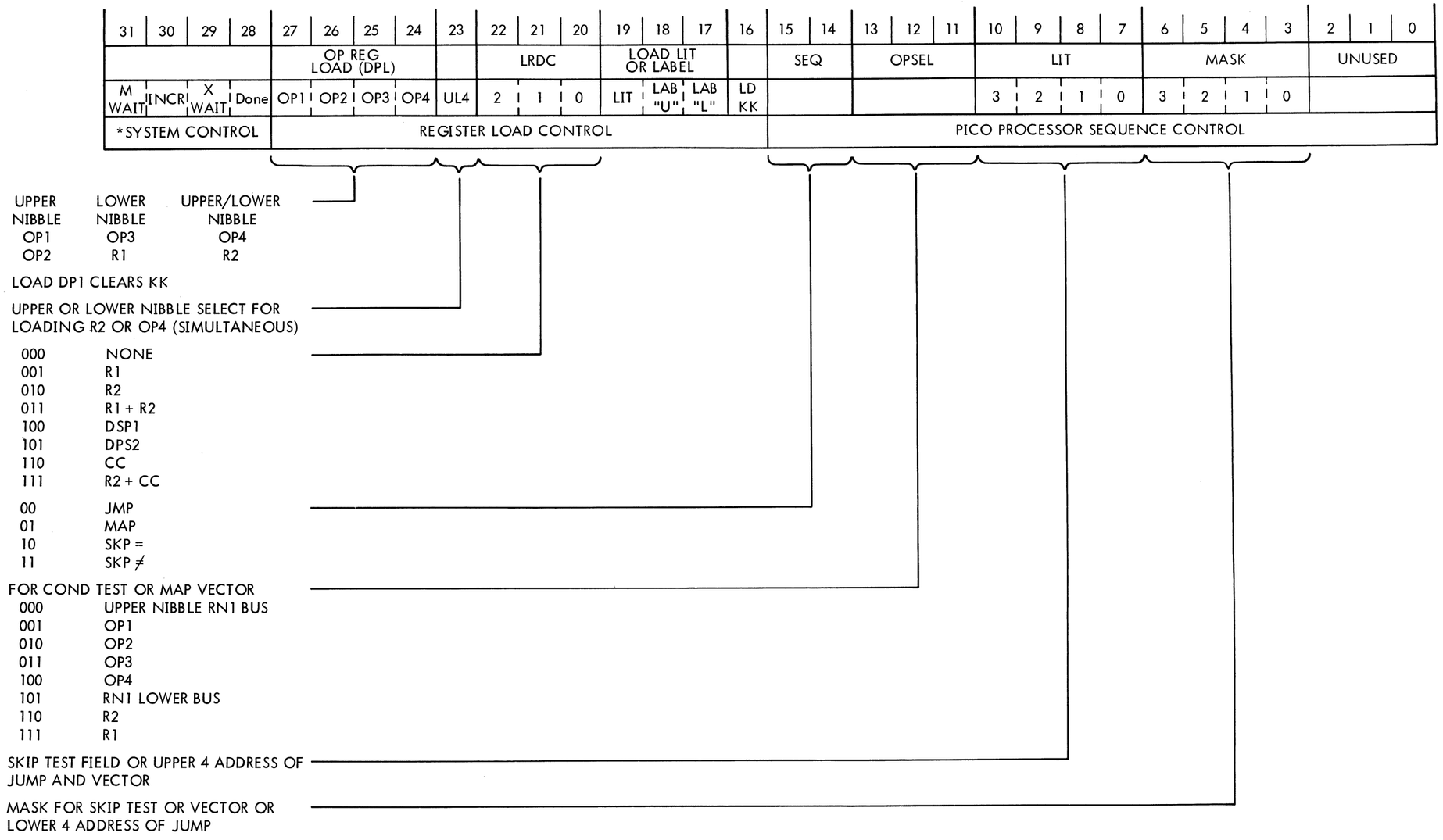
Sequencer operation field

Operand selection field

Literal field

Mask field

RN1 INSTRUCTION FORMAT



* SYSTEM CONTROL DEFINITIONS

M WAIT	MEMORY WAIT: HOLD INSTR UNTIL DATA IS READY ON RN1 BUS
INCR	INCREMENT: APAR TO SELECT NEXT RN1 BYTE
X WAIT	EXECUTION WAIT: HOLD INSTR UNTIL INST HOLD REG X FERRED
DONE	INDICATE THAT HOLDING REG HAVE BEEN FILLED
	CAN DO ALL AT SAME TIME

Figure 4-45. Control Store Word Breakdown.

The sequencer operation field specifies the type of branch or skip to be executed. The operand selection field is used to select OP1, OP2, OP3, OP4, R1, R2 or the upper or lower half of the RNI bus for testing.

> NOTE: R1 and R2 are implemented in the OP-code but are not used.

Jump Operation

If the sequencer operation field contains "00," the operation performed will be an unconditional jump. The upper four bits of the jump address are taken from the literal field. The lower four bits are taken from the mask field.

Vector Operation

If the sequencer operation field contains "01," the operation performed is a vectored jump. The upper four bits of the jump address are taken from the literal field. The lower four bits are derived from the logical "and" of the mask field and the selected operand. This operation performs a table jump.

Skip Operation

If the sequencer operation field contains "10" or "11," the operation performed is a skip if equal or skip if not equal, respectively. The upper four bits of the next address are unchanged from those of the current control store address. If the skip condition is not met, the lower four bits of the current address with one added becomes the next address.

If the skip condition is met, two is added. Only the lower four bits are affected by the skip operation under any circumstances. The equal condition is determined by a bit-for-bit comparison between the selected operand and the literal field. The mask field selects which combination of the four comparisons are used to determine a composite equal condition. For example, if the mask field were "0011," only the lower two bits of the selected operand and of the literal would be compared for equality.

Memory Fetch and Instruction Register Control Logic

This control logic provides for communication with the memory system and with the main processor. When either a system reset or an RNI reset (caused by loading the RPAR) occurs, this logic is initialized. This logic consists of memory interface control, RNI sequencer hold control, and main processor interface control (refer to Figure 4-46).

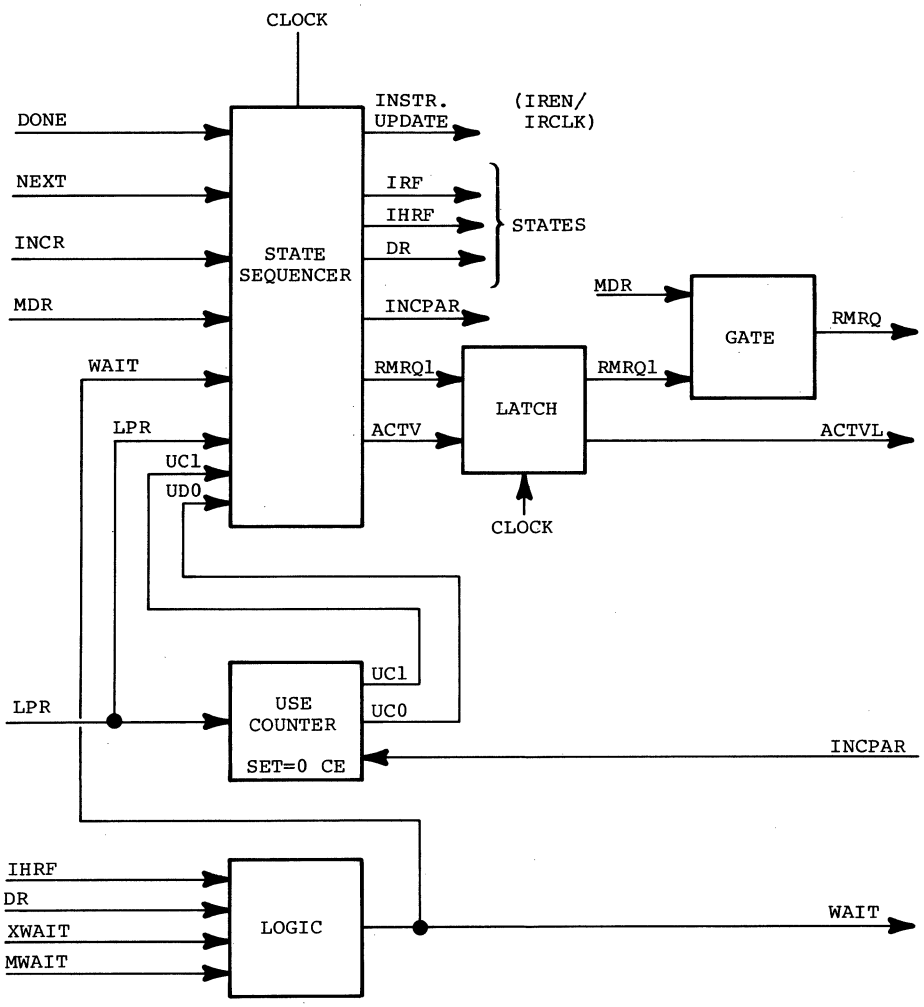


Figure 4-46. RNI Fetch and Control State Block Diagram

Memory Interface Control

The control signals involved in memory interface are the RNI memory request (RMRQ), the RNI memory data strobe (RDST) and the INCR signal from the RNI control store. When the RNI unit is initiated, a memory request is made which is removed when the data strobe is returned.

The data strobe also causes a 2-bit use counter to reset. The INCR signal increments the use counter to indicate the number of bytes of the memory data word that have been processed. A memory request is made when the use count reaches three and the INCR signal is asserted. Data ready is reset at this time, also to indicate that a memory cycle must be completed before additional data may be processed. The INCR signal is also used to generate INCPAR, which causes the RNI program address register to increment.

Instruction Register Control

There is a status associated with each rank of the instruction register. These are instruction holding registers full (IHRG) for the first (RNI) rank and instruction register full (IRF) for the second (ALU) rank. Two control signals, instruction register load enable (IREN) and the instruction register load clock (IRCLK), are used to control the transfer of data from the first rank to the second rank.

The control signal is decoded from the main processor command to initiate a transfer into the second rank. The done control signal from the RNI control initiates a transfer into the first rank. Upon initialization both IRF and IHRF are reset. The first done after this will cause the transfer of data from the first rank to the second rank and set IRF. If this is followed by a next, IRF will be reset; if followed by a simultaneous next and done, the transfer between instruction register ranks is carried out and no status changes. If IRF and IHRF are both reset, next should never occur but will be ignored if it does. If IRF and IHRF are both set, done should never occur but will be ignored if it does.

RNI Sequence Hold Control

During memory cycles (when data is not yet available) and when both ranks of the instruction register are filled (when there is nowhere to put data), RNI processing cannot proceed. To control this a signal called "wait" is generated. The XWAIT term from the control store causes the wait state if the IHRF is set, i.e., processing cannot continue until the main processor completes execution of the current instruction and either requests the next instruction or loads the RPAR. The MWAIT term from the control store causes the wait state if DR is reset, i.e., processing cannot continue until the memory data register is filled.

MAIN PROCESSOR INTERFACE CONTROL

The main processor interface consists of four signals:

Next from the main processor

LPR (load program address register) from the main processor

DIRF from the instruction register control logic

ACTV (active) from RNI

The next and IRF signals have been described above. The main processor should never assert next without first assuring that DIRF is set.

The LPR signal causes the contents of the CPU destination bus to be entered into the RNI program address register and causes the initialization of the RNI picoprocessor. Because of this, it is imperative that no RNI memory cycle be in progress then the main processor assets LPR. To ensure this, the ACTV signal must be tested by the main processor. This signal is high either if DR is reset (a memory request is being made) or if IHRF is reset (RNI is processing the next instruction and could initiate a memory cycle before the main processor can assert LPR).

Timing diagrams illustrating the RNI operations are shown in Figures 4-47 and 4-48. Figure 4-48 shows only the instruction register control functions. Figure 4-48 also shows memory cycles that would occur if processing a 3-byte instruction was followed by a 6-byte instruction.

State Diagram

A state diagram and description table is shown in Figure 4-47. The state variables are DR (data ready) for RNI processing, IRF (instruction register full) and IHRF (instruction holding register full). The controlling terms are MDR (memory data ready [same as RDST RNI data strobe]), done (RNI processing done), next

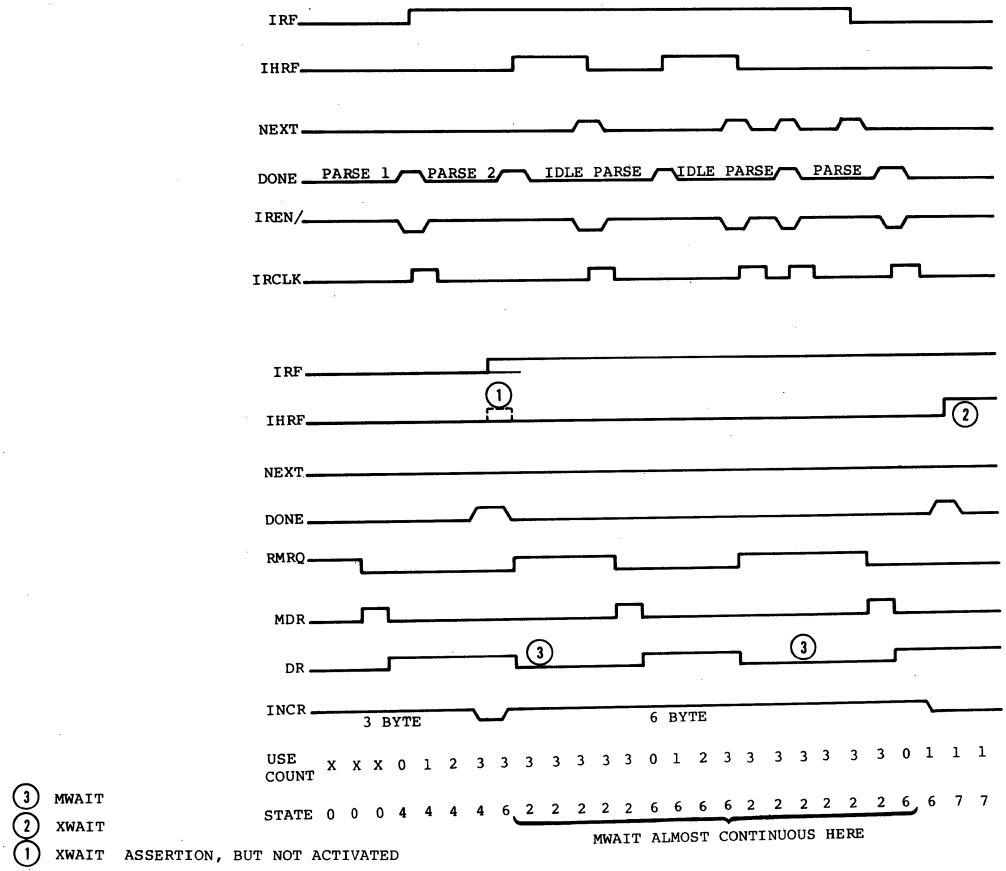


Figure 4-47. Read Next Instruction Timing

DR	IRF	IHRF	ACTIVE	DESC.
0	0	0	1	INITIAL PARS. (TO BEGIN)
0	0	1	1	UNUSUAL STATE
0	1	0	1	SECOND PARSE (TO BEGIN)
0	1	1	1	MEMORY CYCLE FOR THIRD PARSE
1	0	0	1	INITIAL PARSE (IN PROGRESS)
1	0	1	0	UNUSUAL STATE
1	1	0	1	SECOND PARSE (IN PROGRESS)
1	1	1	0	TWO PARSES COMPLETE, MEMORY CYCLE COMPLETE FOR THIRD PARSE

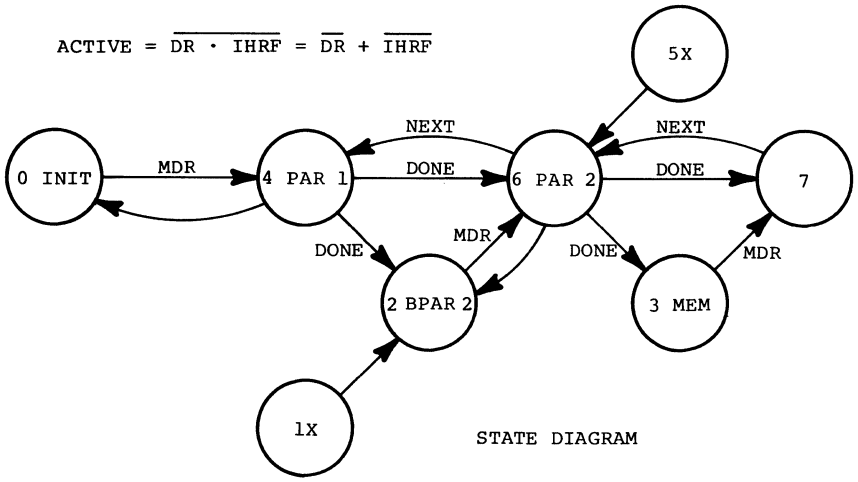


Figure 4-48. Read Next Instruction Sequence

(main processor request for next instruction) and a term derived from the use counter and INCR (4 bytes used on the diagram). The meaning of each state is explained in the following paragraphs.

1. State 0 Initiate

This is the initial state in which no instructions have been completely processed and RNI is waiting on a memory cycle to complete.

2. State 1

This state should never occur in proper operation but will be handled correctly. In this state the instruction holding register is full but the instruction register is empty.

3. State 2 MPAR2

In this state one instruction has been completely processed and RNI is waiting on a memory cycle before continuing with the processing of a second instruction.

4. State 3 MEM

In this state two instructions have been fully processed but a memory cycle is in progress because the second of these instructions used the last byte of the memory data register.

5. State 4 PAR1

In this state RNI is processing an initial instruction. The main processor is waiting for an instruction at this time.

6. State 5

This state is similar to state 1.

7. State 6 PAR 2

In this state RNI is processing an instruction to follow the one already in the instruction register.

8. State 7 COMP

In this state all registers are filled and RNI is inactive waiting for either a next request or initialization.

MAIN PROCESSOR INTERACTION

The main processor must execute the following steps to properly interact with the RNI.

(a) Wait for an instruction register full (IRF) to be set.

(b) Allow for propagation delay through mapping proms.

(c) Execute the instruction.

(d) Request the next instruction (NEXT), and go to step (a) if it is not a branch instruction.

(e) If it is a branch instruction, wait for RNI to become inactive (ACTV/ is set).

(f) Load RPAR.

(g) Go to step (a).

MEMORY DATA REGISTER

The memory data register is a 32-bit register which is accessed one byte at a time by RNI. The register is loaded from the memory data bus when a RNI data strobe (RDST) occurs, signaling the completion of a memory cycle. The lower two bits of the

RPAR are used to sequentially select one of the four bytes of memory data to be enabled onto the RNI bus from which the memory data is transferred to the instruction register.

As shown in Figure 4-49, there are three ranks of program address registers. Since program execution is restricted to the 512-byte frames, the upper 15-bits of all program address registers are always the same, requiring only one 15-bit register for all ranks with 9-bit registers representing each rank.

The first rank is the RNI program address register or RPAR. This register is loaded by the main processor to affect a software branch. Loading this register causes initialization of RNI. The RPAR provides the memory address for all RNI memory cycles. As each byte of the memory data is referenced, the RPAR is incremented (under control of the INCPAR signal which is controlled by INCR from the control store). When all 4-bytes of the memory data register have been processed, the RPAR then points to the next word in memory. Since memory words accessed by RNI do not necessarily lie on double word boundaries, the lower 2-bits of the RPAR are decoded to select one of the four bytes of memory data as the RPAR is incremented. This allows for sequential access of bytes for processing.

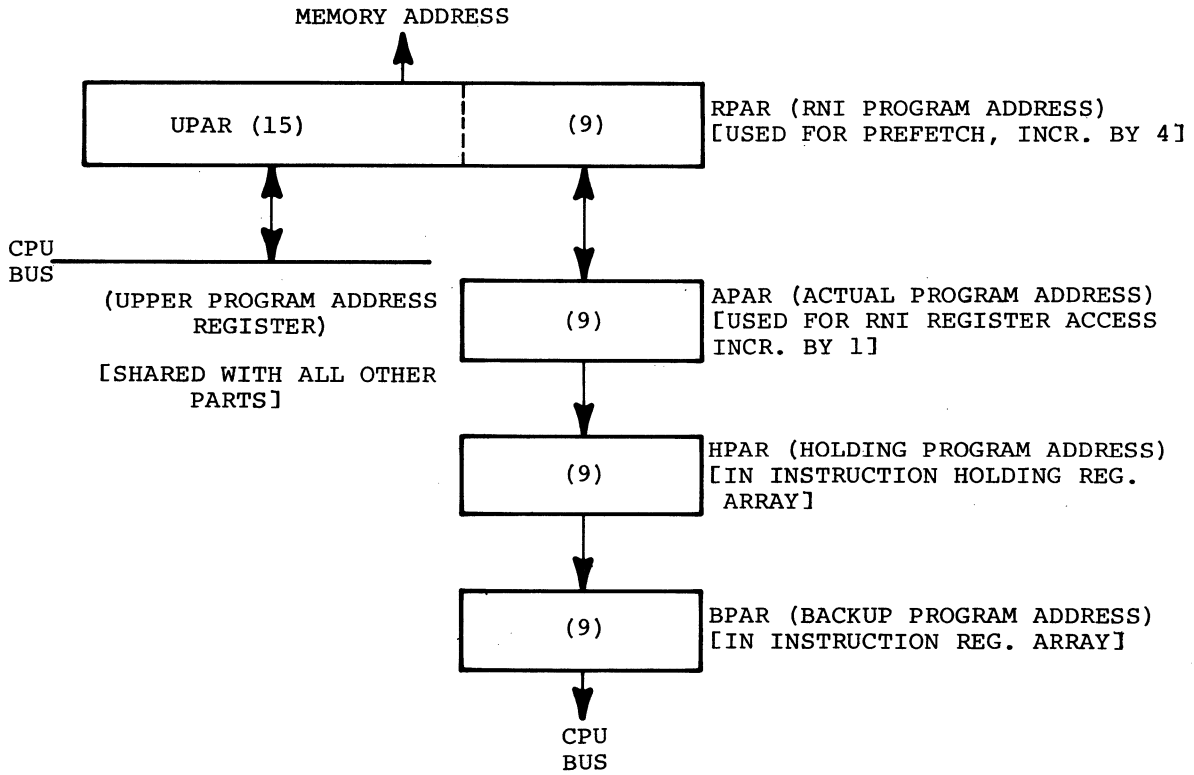


Figure 4-49. Program Address Register

At the start of processing for each execution, the OP1 register is loaded. At this time the contents of the RPAR are transferred into the holding program address register or HPAR. The HPAR is considered one of the instruction holding registers since it contains the starting address of the instruction in this first rank of instruction registers.

When the instruction holding registers are transferred into the instruction registers, the HPAR is transferred to the backup program address register or BPAR. The contents of BPAR are provided for allowing the main processor to return to the current instruction if an interrupt condition exists, i.e., frame fault condition or external interrupt condition.

Instruction Register

The instruction register is actually a double ranked set of several registers, each of which has a specific purpose oriented toward the REALITY instruction set. The first rank of registers is loaded from the memory data register from the RNI bus under control of the RNI firmware. A 12-bit field from the control store specifies the combination of registers to be loaded.

The second rank of registers is loaded from the first rank under the control of the instruction register control logic. Four of the registers in the second rank feed a set of proms that provide opcode branch addresses to main processor microsequencer. These are the opcode registers. Four other registers in the second rank are routed to the address computation portion of the main processor. These are the R1, R2, DSP1 and DSP2 registers. The remaining registers are fed to various parts of the main processor.

The organization of the instruction register is shown in Figure 4-45. The RNI bus source fields for the various registers is shown in Figure 4-44.

Opcode Registers

Five bits from the control store are used to control loading of the OP1, OP2, OP3 and OP4 registers. Four of these bits individually correspond to each of the registers. The fifth selects either the upper or lower 4-bits of the RNI bus for loading into OP4.

The OP1 register is loaded from the upper 4-bits of the RNI bus. This is always the primary opcode which is always in the first byte of the instruction; therefore, additional functions are performed when this register is loaded:

(a) The RPAR is transferred to HPAR.

(b) The KK register (see below) is reset to zeros.

The OP2 register is loaded from the upper 4-bits of the RNI bus. The OP3 register is loaded from the lower 4-bits of the RNI bus. The OP4 register is loaded from the upper 4-bits if the upper/lower select (UL4) is a 1; from the lower 4-bits if the UL4 is a 0. These three registers contain secondary opcodes.

The four opcode registers of the first rank are used in the picoprocessor sequencer for testing and sequence control. This allows the RNI firmware program to make decisions based on the primary and secondary opcodes of an instruction, the length of the instruction, and the organization of fields in the instruction.

The four opcode registers of the second rank go to the instruction mapping proms (see Figure 4-50). One of these proms is the primary map and uses OP1 and OP4 to generate a branch address for the microsequencer. OP1 and OP4 also are used by another prom to select which of two additional secondary map proms is the selected one for the secondary map branch address. The combination of either OP2 or OP3 and OP4 is used to generate the secondary map.

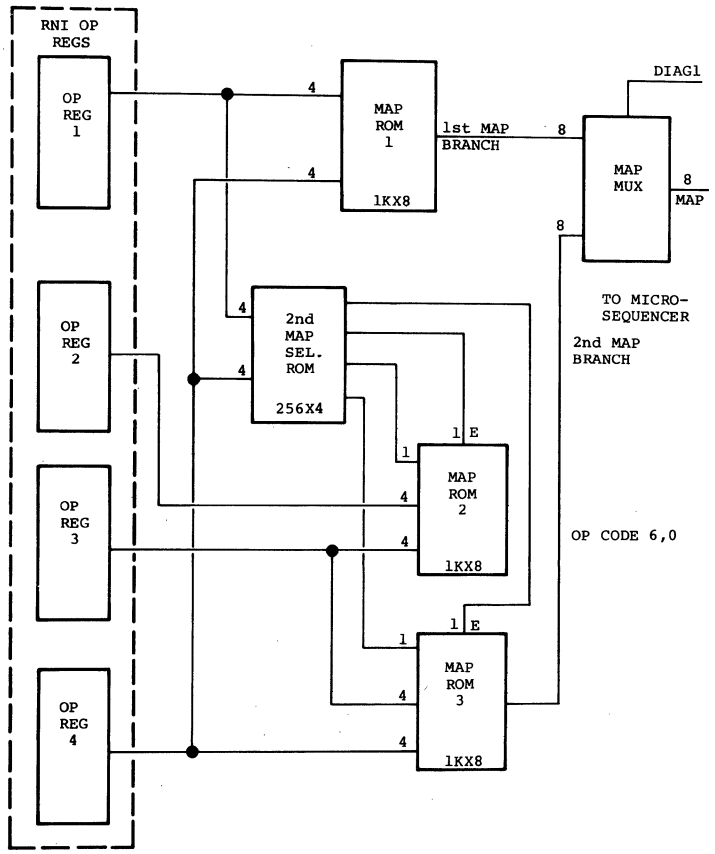


Figure 4-50. Read Next Instruction MAP ROM Organization

REGISTER USED FOR ADDRESS COMPUTATION

There are four pairs of registers that contain data intended exclusively for address computation. These are referred to as R1, R2, DSP1, and DSP2. In addition, the KK register is used by address computation.

The R1 and R2 registers each contain four bits. R1 is always loaded from the lower 4 bits of the RNI bus. R2 can be loaded from either the upper or lower 4 bits of the RNI bus as determined by the upper/lower select UL4) bit from the control store. The definition of this bit is inverted from its meaning for the loading of OP4, i.e., upper 4 bits if zero; lower 4 bits if 1.

The DSP1 and DSP2 registers are each 8 bits. They are loaded directly from the RNI bus. There are not individual bits for load control of these registers as there are for the opcode registers. Instead, three bits from the control store are decided to select one of seven combinations of registers to be loaded. The seven choices (not including the choice of no registers) are: R1 only, R2 only, R1 and R2, DSP1 only, DSP2 only, CC only, or R2 and CC.

OTHER REGISTERS

There are five pairs of registers which are used by various components of the main processor. These registers are referred to as the label, LIT, KK, CC and PAR registers. The KK and CC registers have been briefly mentioned above. The PAR registers have been described in detail as program address registers. HPAR and BPAR were described previously. Associated with the program address registers is a status bit called FRMX, which is passed on to the branch condition multiplexer in the microsequencer. This bit is set if the instruction in the second rank does not fit in the current program frame, i.e., an attempt was made to execute across a frame boundary.

The KK register is 2 bits, which are loaded from bits 7 and 6 of the RNI bus. The KK register is also cleared when OP1 is loaded. The contents of KK represent the operand word length--1, 2, 4 or 6 bytes. It is passed to and used by address computation, rotate and the branch condition logic. One bit from the control store controls loading of KK.

The CC register is 2 bits, which are loaded from bits 3 and 2 of the RNI bus. The contents of CC represent the condition to be tested by the ALU for conditional branch software instructions. The 2 bits of CC are individually available for use as microsequencer branch conditions. Load control of CC is decoded as previously described.

The label register is 16 bits wide. It is divided into two single byte registers (labelu [upper] and labell [lower]), each of which may be directly loaded from the RNI bus. Each byte has a corresponding control bit in the control store. The label register is used to pass instruction fields more than 8 bits in length to the ALU, i.e., branch addresses. The entire 16-bit register can be selected by the ALU as an external source.

The literal (LIT) register contains 8 bits which are loaded directly from the RNI bus. One bit from the control store is used to load this register. The LIT register is used to pass one-byte literal fields from instructions to the ALU, i.e., character scan masks. The LIT register is an external source for the ALU.

RNI CONTROL STORE INFORMATION FORMAT

The RNI control store instruction is diagrammed in Figure 4-45. Descriptions for each of the bits or fields contained in each RNI instruction follow.

MWAIT Bit 31 Memory Bit

No portion of the current instruction is executed until data has been strobed into the memory data register and is available on the RNI bus. This must be specified in any RNI instruction that loads a register or tests the RNI bus.

INCR Bit 30 Increment RPAR

The next byte of the instruction stream (in the memory data register) is to be selected for the next instruction. This indicates that processing of the current byte on the RNI bus will be completed with the current RNI instruction.

XWAIT Bit 29 Execution Wait

No portion of the current instruction is executed until the instruction holding register is transferred to the instruction register. This must be specified as the first step of processing before any changes are made to the instruction holding register.

Done Bit 28 Processing Done

The processing of the current software instruction holding register is complete and the instruction holding register is now full. No registers can be loaded on the same RNI instruction if done is indicated. This is the last operation to be executed for each software instruction.

OP1 Bits 27, 26, 25, 24 Opcode Register Load

Bit 27 controls OP1, clearing of KK, and loading of HPAR. Bits 26, 25, and 24 control OP2, OP3, and OP4, respectively.

UL4 Bit 23 Upper/Lower Nibble Select

If bit 23 is a zero, OP4 will be loaded from the lower 4 bits of the RNI bus and R2 will be loaded from the upper 4 bits of the RNI bus. If bit 23 is a 1, OP4 will be loaded from the upper 4 bits and R2 will be loaded from the lower 4 bits. This is a select control only; load control is defined by other bits.

LRCD Bits 22, 21, 20 Load RX, DSP and CC Registers

The R1, R2, DSP1, and DSP2 and CC registers are loaded per the following:

23	22	21	Function
0	0	0	None
0	0	1	R1 only
0	1	0	R2 only
0	1	1	R1 and R2
1	0	0	DSP1 only
1	0	1	DSP2 only
1	1	0	CC only
1	1	1	R2 and CC

LL Bits 19, 18, 17 Load Literal or Label Registers

Bit 19 controls LIT. Bit 18 controls labelu (label upper), and bit 17 controls label (label lower).

LK Bit 16 Load KK

Bit 16 controls the loading of KK.

SEQ Bits 15, 14 Sequence Instruction

These bits define the type of sequence operation to be executed as follows:

15	14	Function
0	0	Jump to address in LIT and mask fields
0	1	Vector to mapped address
1	0	Skip if selected operand (with mask) equals LIT
1	1	Skip if selected operand (with mask) does not equal LIT.

OPSEL Bits 13, 12, 11 Operand Select

These bits are used to select one of eight choices of operands for use in condition testing (skips) or vectoring. They are defined as follows:

13	12	11	Function
0	0	0	Upper 4 bits of RNI bus
0	0	1	OP1
0	1	0	OP2
0	1	1	OP3
1	0	0	OP4
1	0	1	Lower 4 bits of RNI bus
1	1	0	R2
1	1	1	R1

LIT Bits 10, 9, 8, 7 Literal Field

This field is used for the upper four bits of a jump or vector address and for comparison with the selected operand for skip tests.

Mask Bits 6, 5, 4, 3 Mask Field

This field is used in the comparison of skip tests, for masking a selected operand for vectoring, or for the lower four bits of a jump address.

Bits 2, 1 and 0 Not Used

RNI Parsing

A flow chart of RNI parsing is provided in Figure 4-51.

LIST OF SEQUEL FIRMWARE RESTRICTIONS

(a) A/R file not allowed as destination of ALU arithmetic operation.

(b) Wait for one clock to test conditional branch after setting it. (Two clocks for CRT.)

(c) Character test:

(1) For variable tests, the individual byte zero condition result of an ALU opertion is saved by specifying update ALU condition code.

(2) Test variable result on next clock or later.

(3) Fixed character test must be done along with second or third variable test.

(4) Fixed character test must not be enabled during character test clear.

(5) Character test changing address latch is loaded on next clock or later after a character test and only if the next or later command is a character test NOP or sequence 0.

(d) On a call or return call, allow two clocks before doing a return.

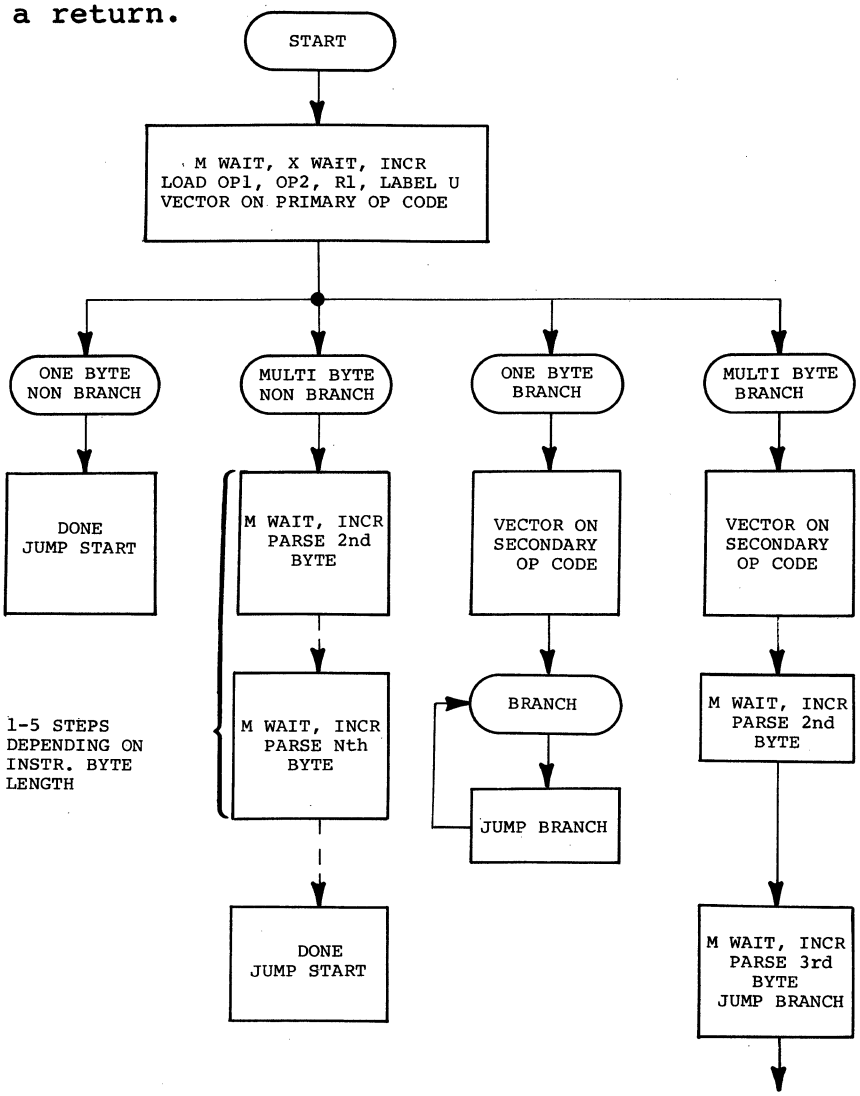


Figure 4-51. Read Next Instruction Parsing Flow Chart

(e) Memory access:

(1) Load HAR on or before memory command.

(2) HAR can be changed on third clock.

(3) Read data available on fifth clock after read command.

(4) Memory write register can be loaded on next clock after write memory command.

(5) Memory write register can be changed on third clock after write command.

(6) T holds are caused by the following: memory request while busy, memory request simultaneous with refresh, DMA, or RNI request.

(7) T hold immediately blocks all operations except address computation which is allowed to occur prior to the block.

(f) RNI commands followed by conditional branch on RNI status:

(1) Next wait 0 clocks before testing instabal (comes up first after load RPAR) prior to using IR contents wait 1 clock before testing RPARAVAL (prior to loading RPAR).

(2) Load wait 0 clocks before testing instaval (no intervening instruction request). Wait 1 clock before testing RPARAVAL ALU operation after multiply step shift NOP in between.

(g) ADC range of 1 frame.

The valid combinations of A-type and B-type commands are shown below:

A-Type	B-Type
Operand address Computation (ADC)	Memory cycles CPU control mode Set flags Character test No-op
Rotate, sign Extend, fill	Memory cycle CPU control mode Set flags Character test No-op

A-Type (Cont'd)	B-Type (Cont'd)
2901 operation	Memory cycle Shift Multiply step Divide step Character test control CPU control modes Set flags No-op
No-op	Memory cycle CPU control mode Set flags Character test control No-op

SECTION 5

MAINTENANCE

INTRODUCTION

This section presents information that will provide assistance in performing maintenance of the SEQUEL System. Maintenance is limited to checkout and corrective procedures that can be performed at the user site. Maintenance should be performed only by qualified service personnel.

For qualified personnel, fault isolation will be performed to the printed circuit board (PCB) level and to the replacement of the PCB as a unit. Fault isolation will also be performed to the submodule division of the power supply.

Preventive Maintenance

All peripherals associated with SEQUEL have existing maintenance manuals that include maintenance procedures. These manuals should be consulted for each of the peripherals. There are limited, but important, maintenance procedures for the SEQUEL that should be followed on a scheduled basis.

1. Cleaning Filters and Screens

The filter for the intake air blower should be vacuumed at least four times a year (see Figure 5-1). This same filter should be washed at least once a year in warm water containing a small amount of detergent. After washing, the filter should be thoroughly dry before it is put back into place.

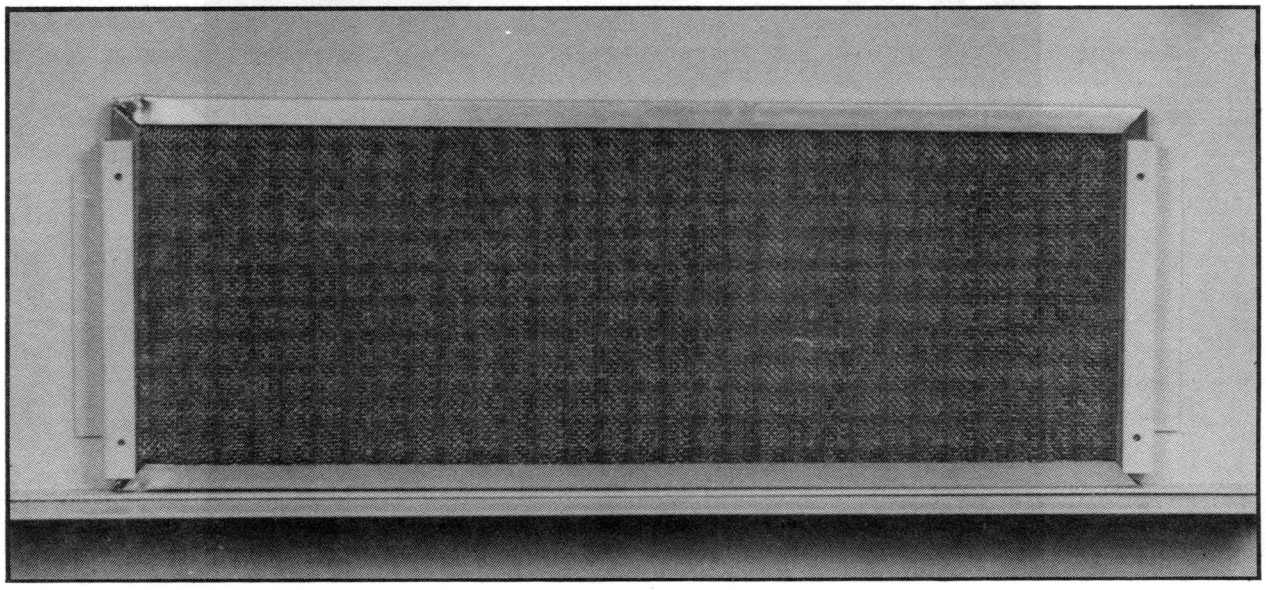


Figure 5-1. Intake Air Blower Filter

2. Cleaning Electro-Magnetic Interference (EMI) Screens

EMI screens should be vacuumed on a regular periodic basis of not less than once every three months. For best results, remove the screen and thoroughly vacuum both sides of the screen (see Figure 5-2).

Tool Requirements

Special tools needed for bulb replacement and diagnostic procedures include the following:

Tool	Part Number
Tool kit to replace front panel lights	T8905
ACLC diagnostic cable	A20032234
Rear door cabinet key	SP52450018

Lamp Replacement Procedures

Figure 5-3 includes the two main tools used to replace bulbs in the front panel. The black tube at the left is used to remove and replace bulbs while the tweezer-like instrument at the right is for removing the lens covering the bulb. The object located in the center of Figure 5-3 is a bulb similar to the ones used in all panel lights.

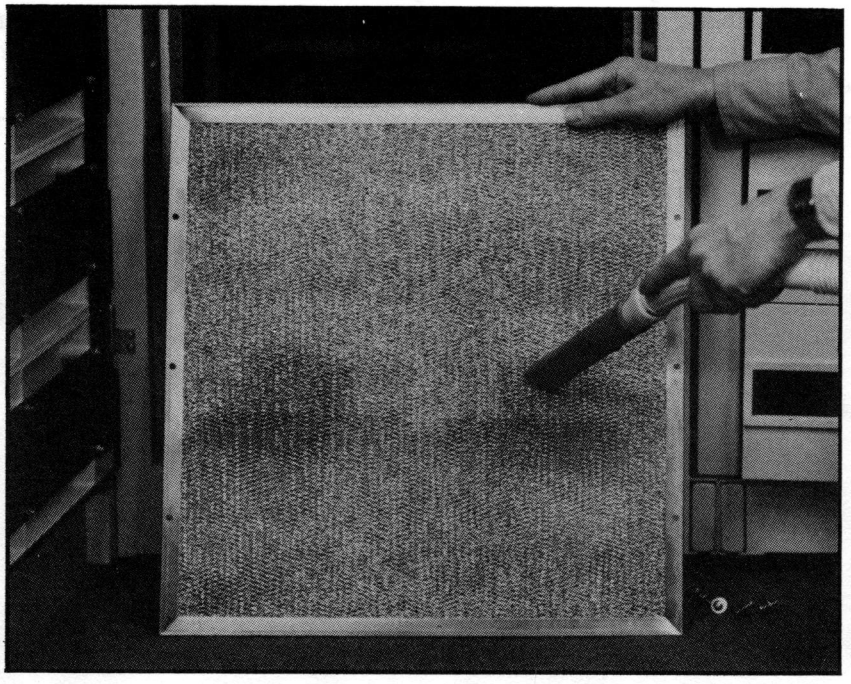


Figure 5-2. Cleaning Procedure for EMI Screens

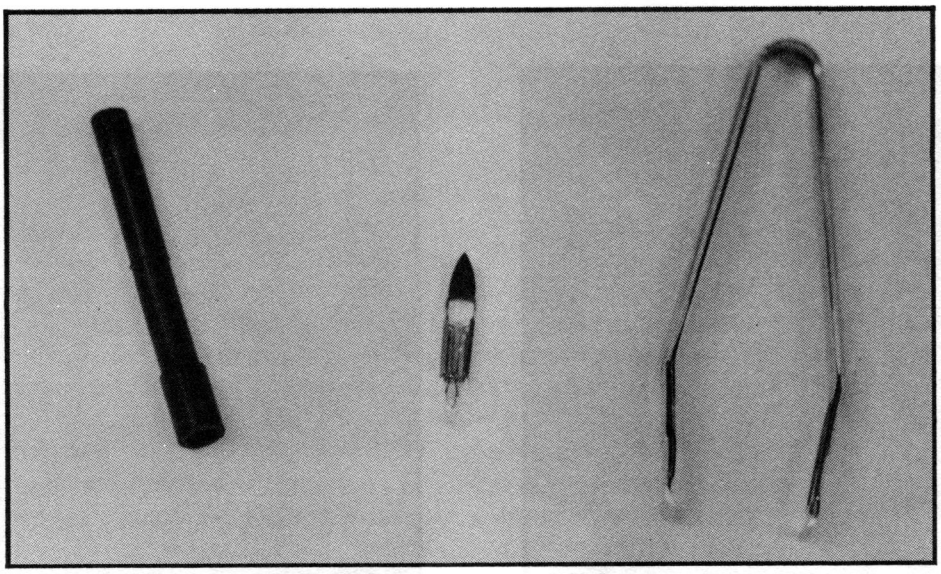


Figure 5-3. Tools to Replace Front Panel Lights

Care should be exercised in removing the lens. There are shallow grooves on each side of the lens that will accommodate the grasping instrument (see Figure 5-4). Once proper contact is made with these grooves, the lens should be carefully pulled straight out. The openings at the two ends of the small black tube have different diameters. The end without the collar should be used to remove the defective bulb (see Figure 5-5). Placement of a finger over the open end of the tube will create a partial vacuum and aid in the removal of the bulb.

The other end of the small black tube, the end with the collar, should be used to replace the bulb (see Figure 5-6). The bulb should be slipped into this open end, glass first. A finger placed over the open end of the tube will once again create a partial vacuum and decrease the likelihood of the bulb falling out during the replacement process. Once the bulb is inserted, push firmly with the black tube and carefully pull the tube staight back. Replace the lens to finish this maintenance procedure.

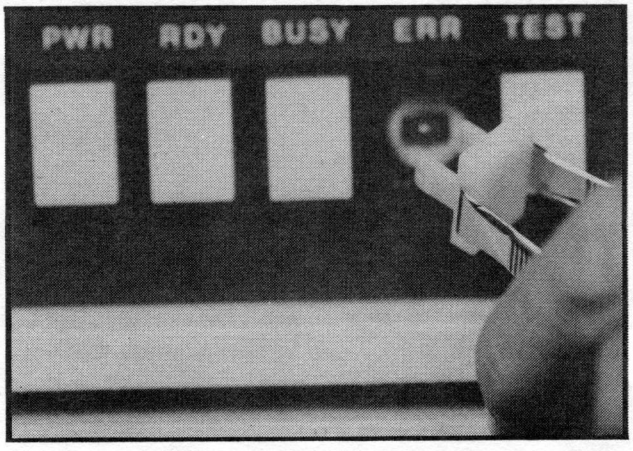


Figure 5-4. Procedure for Removing Panel Light Lens

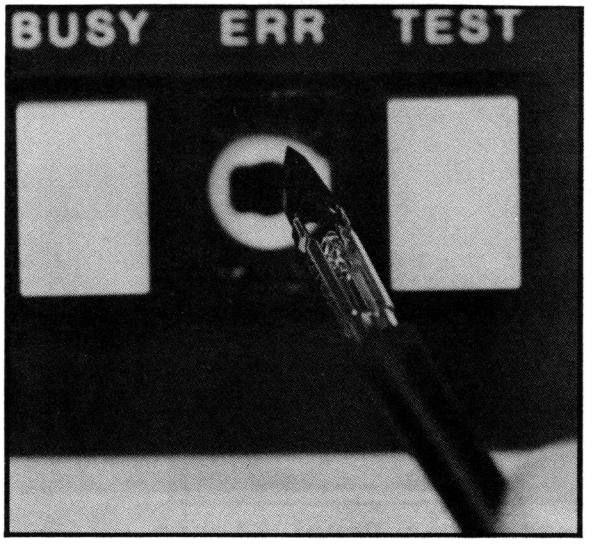


Figure 5-5. Removing Bulb

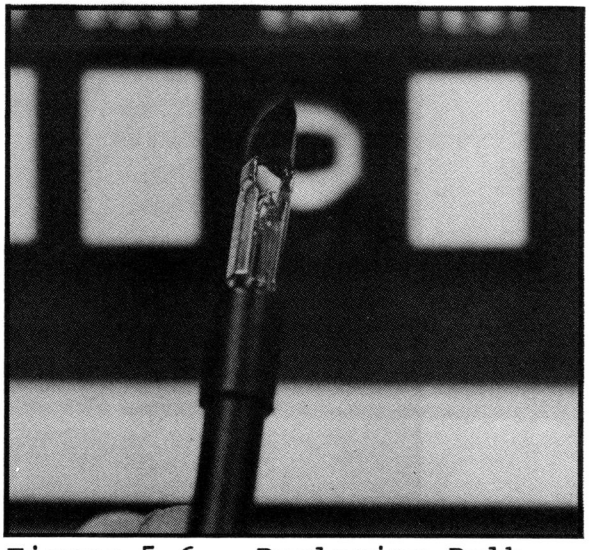


Figure 5-6. Replacing Bulb

Power Supply Maintenance

The power supply should also be vacuumed at least once every three months. An accumulation of dust may provide partial insulation and cause the power supply to malfunction.

Two fuses in the power supply will require replacement on rare occasions. One of the fuses is identified by the pointing finger in Figure 5-7, and the other is identified by an arrow at lower left in the photograph.

Batteries will require replacement on a periodic basis that will be dependent upon usage of the power supply. Battery replacements will be required more frequently in areas where seasonal power blackouts occur. The procedure for replacing batteries is similar to the REALITY battery replacement procedure.

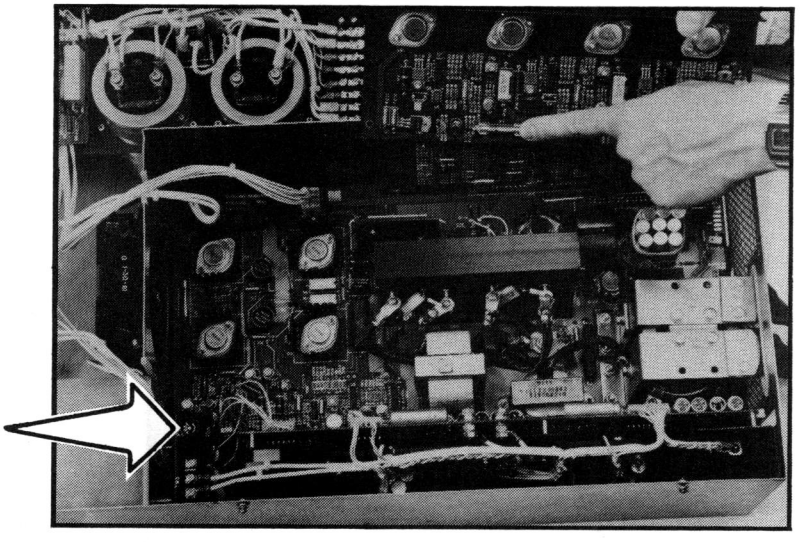


Figure 5-7. Power Supply Fuse Locations

SECTION 6

PARTS LIST, ASSEMBLY DRAWINGS AND SCHEMATICS

INTRODUCTION

This section presents information on service parts. Included are drawings and schematics that will provide maintenance assistance by identifying the service part and item location. SEQUEL spare parts are listed in Table 6-1 and spare parts that are unique to the first forty (40) SEQUEL Systems are listed in Table 6-2. Table 6-3 contains a listing of all drawings and schematics that are included in this manual.

TABLE 6-1

SEQUEL SPARE PARTS

Assembly Number	Description	
A-20032175	PCB MEMORY CONTROL BOARD	
20032007	PCB CPU/ALU (MUST HAVE MASTER SEQUEL LICENSE)	
20032008	PPB CPU/RNI	
20032009	PCB CPU/SPEC FUNCTION	
20032011	PCB DISC CONTROLLER	
20032012	PCB MAG/TAPE PRTR CONTROLLER	
20032013	PCB I/O PROCESSOR	
20032014	PCB ACLC (8-WAY)	
20032015	PCB DMP (DIAG. MAINT. PROCESSOR)	
20032038-001	CONTROL PANEL ASSY	
20032025-XXX	POWER SUPPLY ASSEMBLY	
	Power Requirement	Use Dash #
	50-60HZ 200 VAC	-009
	50-60HZ 208 VAC	-010
	50-60HZ 220 VAC	-011
	50-60HZ 240 VAC	-012
20032043-002	AC INPUT ASSY	
20032050	PCBA SWITCHING REGULATOR	
20032051	PCBA SERIES REGULATOR	

TABLE 6-1, SEQUEL SPARE PARTS (Cont'd)

20032052	PCBA PWR SUPPLY LOGIC
20032101	FUSE ASSY-MEM PWR MOD
20032205	PCB ASSY MEM-PWR MOD
20032206	MEMORY PWR MODULE ASSY
20032176	MEMORY ARRAY BOARD (1 MEG MOS)
20032176-002	MEMORY ARRAY BOARD (512K MOS)
20032065	BACKPLANE
20032117	EXPANSION BACKPLANE
20032132	EXPANSION AC DISTRIBUTION
20014832	CABLE ASSY INTCON CPU
20032034	INTERLOCK SWITCH ASSY
20032049	PCBA AC INPUT
20032062	PCBA - I/O CONN PANEL
20032107-001	CABLE ASSY I/O, 50 CNDCT
20032110-001	CABLE ASSY PWR SPLY/BKPLN
20032162	CABLE ASSY INTLK SWITCH
20032179-001	CABLE ASSY GND. BKPLN
20032186	BLOWER ASSY
CS-20014838	LEAD ACID BATTERY
SP-50820058	CONNECTION SYSTEM
A-20032085	PCB, ADAPT. MTU
20032102-001	CABLE, TERM
20032103-001	CABLE, I/O PRINTER
20032104-00	CABLE, DMP/ACLC
20032106-001	CABLE, MTP I/O
20032108-001	CABLE, CONT. PNL
20032112-001	CABLE, CONT. PNL
20032123-001	CABLE, EXT. BAT.
20032168-001	CABLE, I/O
20032187-001	CABLE, DISC/CTRL
20032163	CABLE, P/S SYNC
20032063-001	A/C POWER MOD

TABLE 6-2

UNIQUE SPARES FOR FIRST 40 SEQUEL SYSTEMS

Assembly No.	Description
A20032064	Memory Array Board 16K RAM 256K MOS
A20032055	Memory Pwr Module Assy
A20032053	PCB Assy Mem-Power Mod

TABLE 6-3

SEQUEL DRAWINGS AND SCHEMATICS

Drawing or Schematic No.	Description
A20032025	Power Supply Assembly
A20032034	Interlock Switch Assembly
A20032038	Control Panel, Assembly
A20032043	AC Input Assembly
A20032050	PCB Assy Switching Regulator
A20032051	PCB Assy Series Regulator
A20032052	PCB Assy Power Supply Logic
A20032062	PCB Assy - I/O Connector Panel
A20032063	AC Pwr Module Assy
A20032065	PCB Assembly Backplane
A20032085	PCB Assy Adapter MTU Ctrl
A20032101	Fuse Assy Mem Pwr Module
A20032117	PCB Assy Expansion Backplane
A20032118	Cable Assy DC Pwr/Gnd
A20032132	AC Pwr Expansion Module Assy
A20032162	Cable Assy, Interlock Switch
A20032163	Cable Assy, P/S Sync
A20032168	Cable Assy - I/O, 50 Conductor
A20032187	Cable Assy, 26 Conductor
A20032193	AC Power Module Assy - Side Access
A20032206	Memory Power Module, Assy
CD20032100	Cable Diagram (Basic System)
CD20032185	Cable Diagram, Expansion
CS20014838	Component Specification Lead Acid Batteries
I20032005	Information Drawing, SEQUEL Document Tree
SC20032025	Schematic Power Supply
SC20032048	PCB Schematic - Control Panel
SC20032062	PCB Schematic - I/O Connector Panel
SC20032065	PCB Schematic - Backplane
SC20032085	PCB Schematic - Adapter MTU Controller
SC20032117	PCB Schematic - Expansion Backplane

REV	EO	ZONE	DESCRIPTION	DWN	CHKD	APPD	DATE
AX	10108		PROTO RELEASE	[illegible]	MD	[illegible]	11-20-90
A	10231		PILOT RELEASE	[illegible]	MD	[illegible]	1/26/91
B	10335		INCORP E.O.	SEL	MD	[illegible]	4-28-91
C	10376		INCORP E.O.	SEL	MD	[illegible]	4-28-91
D	10519		INCORP EO	SPD	R.V.	[illegible]	9-19-91
E	10579		INCORP EO	SPD	R.V.	[illegible]	9-19-91
F	10576		INCORP EO	SPD	R.V.	[illegible]	9-19-91
G	10633		INCORP EO	SPD	R.V.	[illegible]	9-19-91
H	10818A		INCORP EO	GL.			

9	W/O MEMORY POWER	240V, 50-60Hz, 15A	A20032025-016	E
9	W/O MEMORY POWER	220V, 50-60Hz, 15A	A20032025-015	E
9	W/O MEMORY POWER	208V, 50-60Hz, 15A	A20032025-014	E
9	W/O MEMORY POWER	200V, 50-60Hz, 15A	A20032025-013	E
	W/ MEMORY POWER	240V, 50-60Hz, 15A	A20032025-012	E
	W/ MEMORY POWER	220V, 50-60Hz, 15A	A20032025-011	E
	W/ MEMORY POWER	208V, 50-60Hz, 15A	A20032025-010	E
	W/ MEMORY POWER	200V, 50-60Hz, 15A	A20032025-009	E
		A20032025-001 THRU -008		OBS
	DESCRIPTION	POWER REQUIREMENT	ASSY NO.	REV LTR

ASSY TABULATION CHART

10 DO NOT TIE OR ROUTE THIS CABLE WITH ANY OTHER CABLES.

9 OMIT ITEMS INDICATED FOR DASH NO.'S W/O MEMORY POWER.

8 HARDWARE (ITEM 90,91,92) TO BE INCLUDED IN THIS ASSY FOR INSTALLATION INTO CABINET AT HIGHER ASSY LEVEL.

7. WIRING AND COVER (ITEM 6) REMOVED FOR CLARITY ON TOP VIEW.

6 MARK PER Z20016001 WITH DESIGNATIONS SHOWN.

5 MARK PER Z20016001 WITH ASSY NO. REV LTR SERIAL NO. AND PERTINENT INFORMATION.

4. FOR WIRE LIST SEE WL20032025.
3. FOR SCHEMATIC SEE SC20032025.
2. ASSEMBLE PER MICRODATA WORKMANSHIP MANUAL.
1. INTERPRET DRAWNG PER MIL-STD-100.

NOTES: UNLESS OTHERWISE SPECIFIED

THE INFORMATION CONTAINED HEREIN IS PROPRIETARY TO AND CONSIDERED A TRADE SECRET OF MICRODATA CORPORATION AND SHALL NOT BE REPRODUCED, IN WHOLE OR PART, WITHOUT THE WRITTEN AUTHORIZATION OF MICRODATA CORPORATION.

UNINCORPORATED ENGINEERING ORDERS

NEXT ASSY / USED ON / APPLICATION

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES
TOLERANCES .XX ± .02
TOLERANCES .XXX ± .010
TOLERANCES ANGLES ±½°

MATERIAL

FINISH

DRAFTSMAN [illegible] DATE 11-9-90
CHECKER [illegible] 11-18-90
ENGINEER [illegible] 11/19/90
APPD [illegible] 4/19/90
APPD [illegible] 1-21-91
APPD [illegible] 1-21-91
APPD

TITLE
POWER SUPPLY ASSY

SCALE 1:2 | IDENT CODE 52936 | DWG SIZE D | A20032025 H | SHEET 1 OF 2 | REV

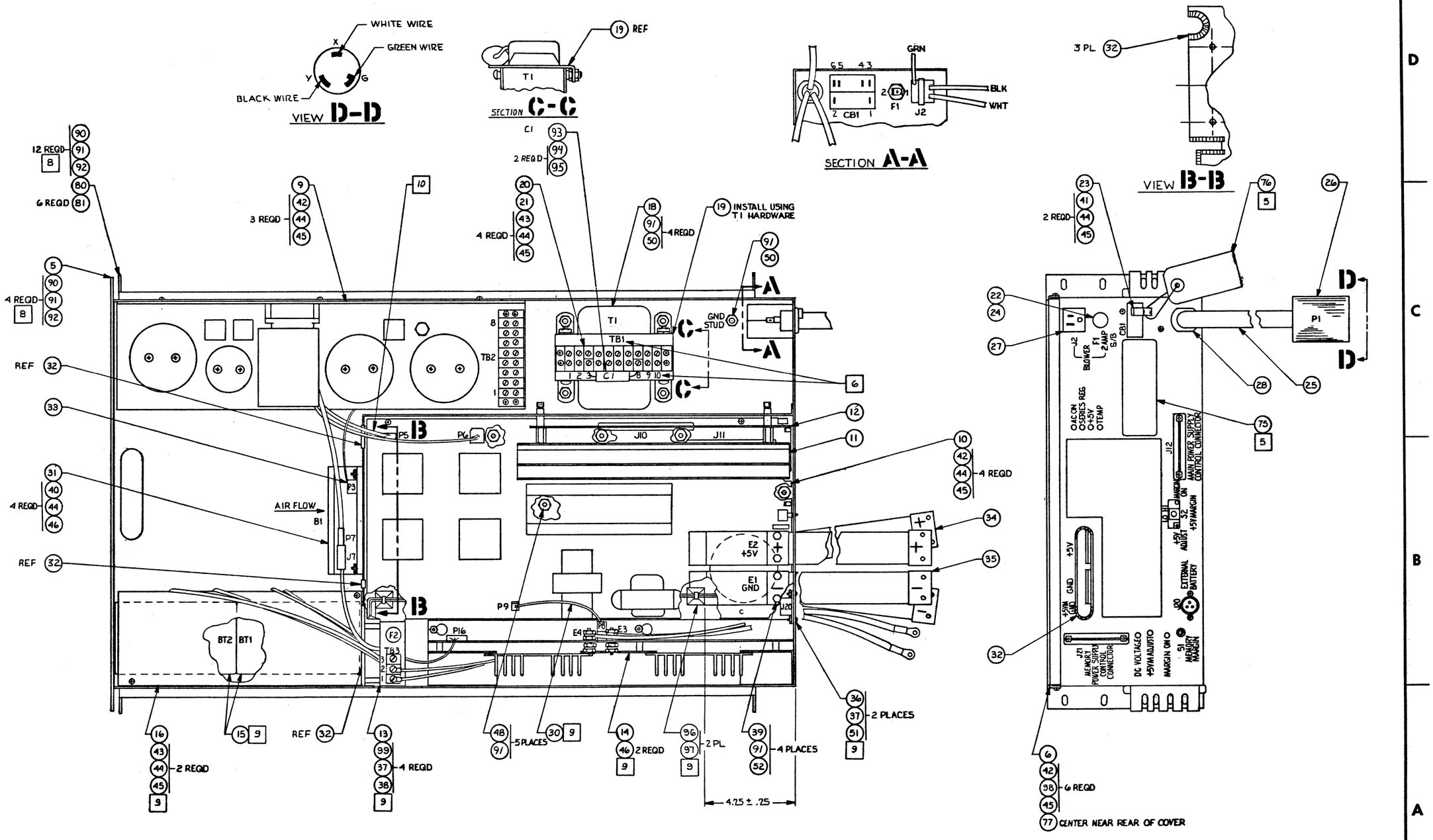
Microdata
IRVINE, CALIFORNIA

A20032025 B
H

6-5

TABULATED

FORM NO. 9-136

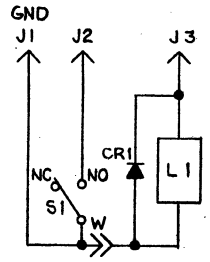


6-7

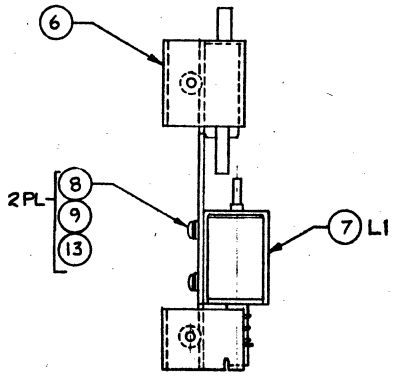
D	A20032025	G
DWG SIZE	SHEET 2 OF 2	REV

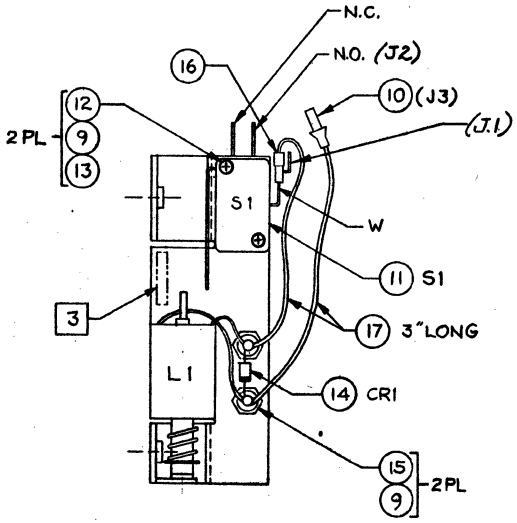
FORM NO. 9-135-1

REV	EO	ZONE	DESCRIPTION	DWN	CHKD	APPD	DATE
AX	9728		PROTO RELEASE	JEC	MD	HG	5-22-80
AX1	10077		REVISED & REDRAWN	PW	[illegible]	[illegible]	11-9-80
A	10172		PILOT RELEASE	PW	[illegible]	[illegible]	1-12-81
B	10452		INCORP EO	PW	MP	[illegible]	6-12-81
C	10580		INCORP E.O	KP	MP	[illegible]	8-13-81



SCHEMATIC
(DOOR OPEN)





A20032034
C

A20032034	C
ASSY NO	REV LTR
ASSY REV CHART	

THE INFORMATION CONTAINED HEREIN IS PROPRIETARY TO AND CONSIDERED A TRADE SECRET OF MICRODATA CORPORATION AND SHALL NOT BE REPRODUCED, IN WHOLE OR PART, WITHOUT THE WRITTEN AUTHORIZATION OF MICRODATA CORPORATION.

UNINCORPORATED ENGINEERING ORDERS

3. MARK PER Ž20016001 WITH ASSEMBLY NO., REV LETTER, AND SERIAL NUMBER
2. ASSEMBLE PER MICRODATA WORKMANSHIP MANUAL
1. INTERPRET DRAWING PER MIL-STD-100

NOTES: UNLESS OTHERWISE SPECIFIED

USED ON | NEXT ASSY
APPLICATION

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES
TOLERANCES .XX ±.02
TOLERANCES .XXX ±.010
TOLERANCES ANGLES ±½°
MATERIAL
FINISH

		DATE
DRAFTSMAN	J E CANTRELL	5/5/80
CHECKER	M. DOYLE	5/9/80
ENGINEER	D. BALFOUR	5/14/80
APPD	GARRETT HO	5/22/80
APPD	[illegible]	1/9/81
APPD	[illegible]	1-9-81
APPD		

TITLE
INTERLOCK SWITCH ASSEMBLY

SCALE 1/1 | IDENT CODE 52936

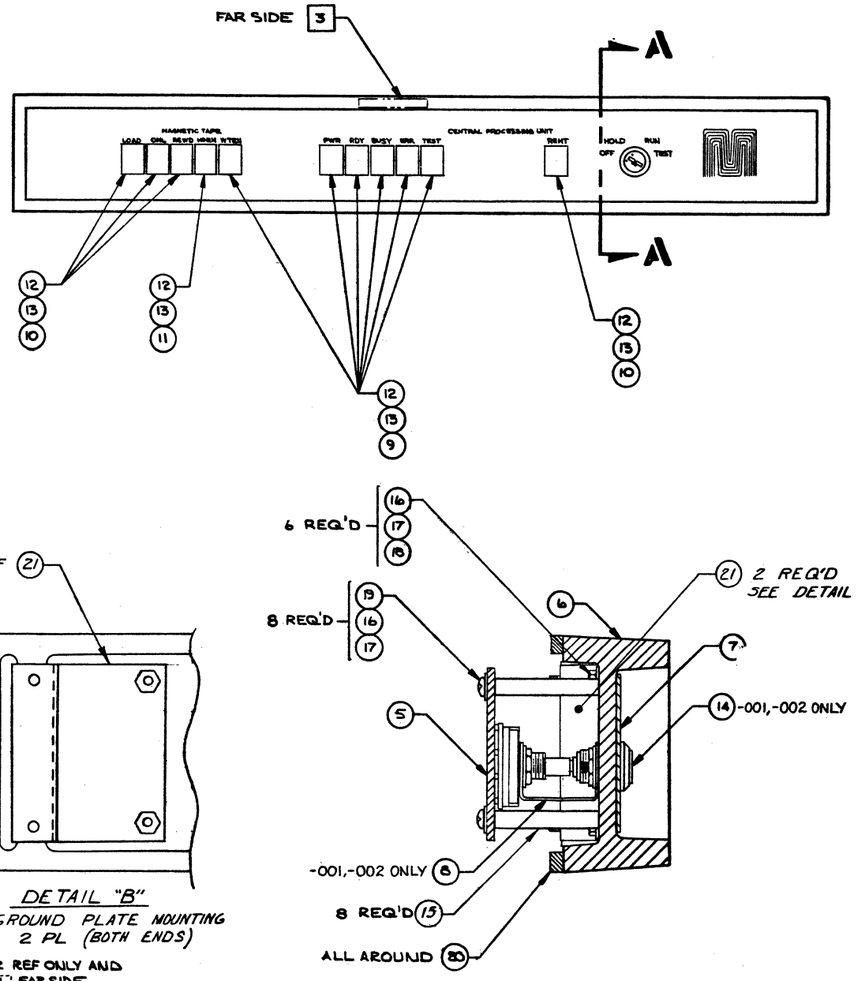
Microdata
IRVINE, CALIFORNIA

C	A20032034	C
DWG SIZE	SHEET 1 OF 1	REV

6-9

FORM 0-1350-3

REV	EO	ZONE	DESCRIPTION	DWN	CHKD	APPD	DATE
AX	9717		PROTOTYPE RELEASE	RB	[illegible]	[illegible]	5-15-80
AX1	10016		INCORPORATE E O	[illegible]	[illegible]	[illegible]	9/8/80
AX2	10143		INCORPORATE E O	AK	[illegible]	[illegible]	12-9-80
A	10138		PILOT RELEASE	AK	[illegible]	AK	1/0/81
B	11111		INCORPORATE E O	[illegible]	[illegible]	[illegible]	4-1-82
C	11199		INCORPORATE E O	[illegible]	[illegible]	[illegible]	4-1-82



SECTION A-A
SCALE: NONE

DETAIL "B"
GROUND PLATE MOUNTING
2 PL (BOTH ENDS)

4. SWITCH REFERENCE DESIGNATORS ARE FOR REF ONLY AND APPEAR ON SWITCH PANEL. EXAMPLE: S1 FAR SIDE
3. MARK PER 22001600I WITH PART NUMBER APPROPRIATE DASH NUMBER AND REVISION LETTER.
2. ASSEMBLE PER MICRODATA WORKMANSHIP MANUAL.
1. INTERPRET DRAWING PER MIL-STD-100.

NOTES: UNLESS OTHERWISE SPECIFIED

A20032038-003	CONTROL PANEL W/O CPU SWITCHES	WHITE	BROWN	A
A20032038-002	CONTROL PANEL W/O MTU SWITCHES	WHITE	BROWN	A
A20032038-001	CONTROL PANEL	WHITE	BROWN	B
PART NUMBER	DESCRIPTION	BEZEL COLOR	PANEL COLOR	REV LTR

ASSEMBLY TABULATION CHART

A20032038 B C

6-11

TABULATED

THE INFORMATION CONTAINED HEREIN IS PROPRIETARY TO AND CONSIDERED A TRADE SECRET OF MICRODATA CORPORATION AND SHALL NOT BE REPRODUCED, IN WHOLE OR PART, WITHOUT THE WRITTEN AUTHORIZATION OF MICRODATA CORPORATION.

UNINCORPORATED ENGINEERING ORDERS

NEXT ASSY
USED ON
APPLICATION

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES
TOLERANCES .XX ± .02
TOLERANCES .XXX ± .010
TOLERANCES ANGLES ±½°

MATERIAL

FINISH

DRAFTSMAN R. BIESCHKE DATE 4-21-80
CHECKER [illegible] 5/3/80
ENGINEER [illegible] 9/14/80
APPD [illegible] 5-12-80
APPD [illegible] 10/1/80
APPD [illegible]
APPD

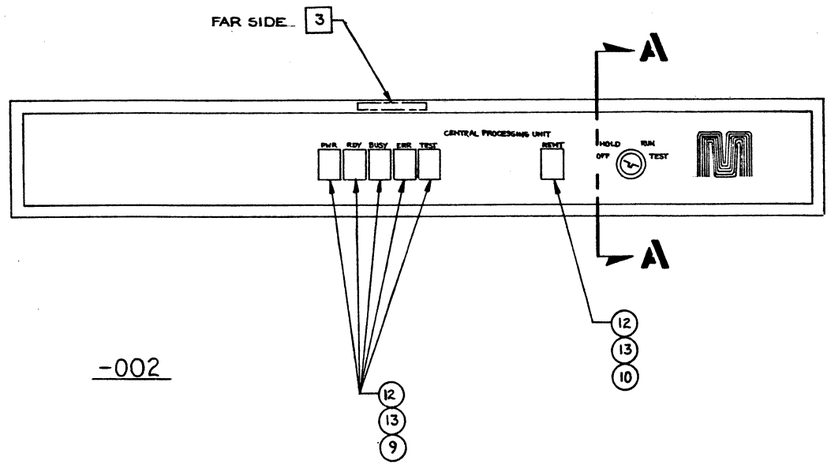
TITLE
CONTROL PANEL, ASSEMBLY

SCALE 1/2 & NOTED | IDENT CODE 52936

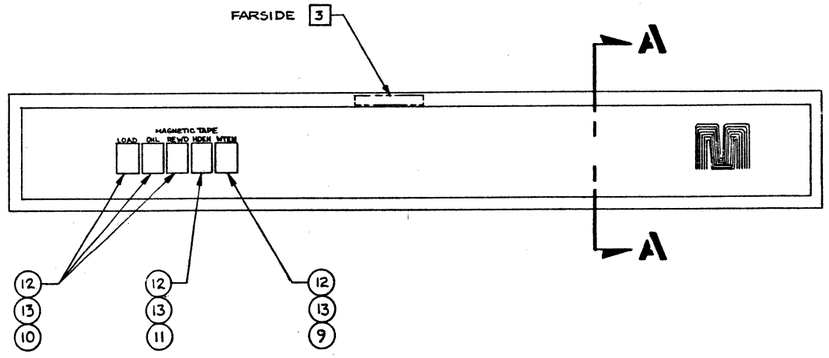
Microdata
IRVINE, CALIFORNIA

D | A20032038 | C
DWG SIZE | SHEET 1 OF 2 | REV

FORM NO. 9-138



-002



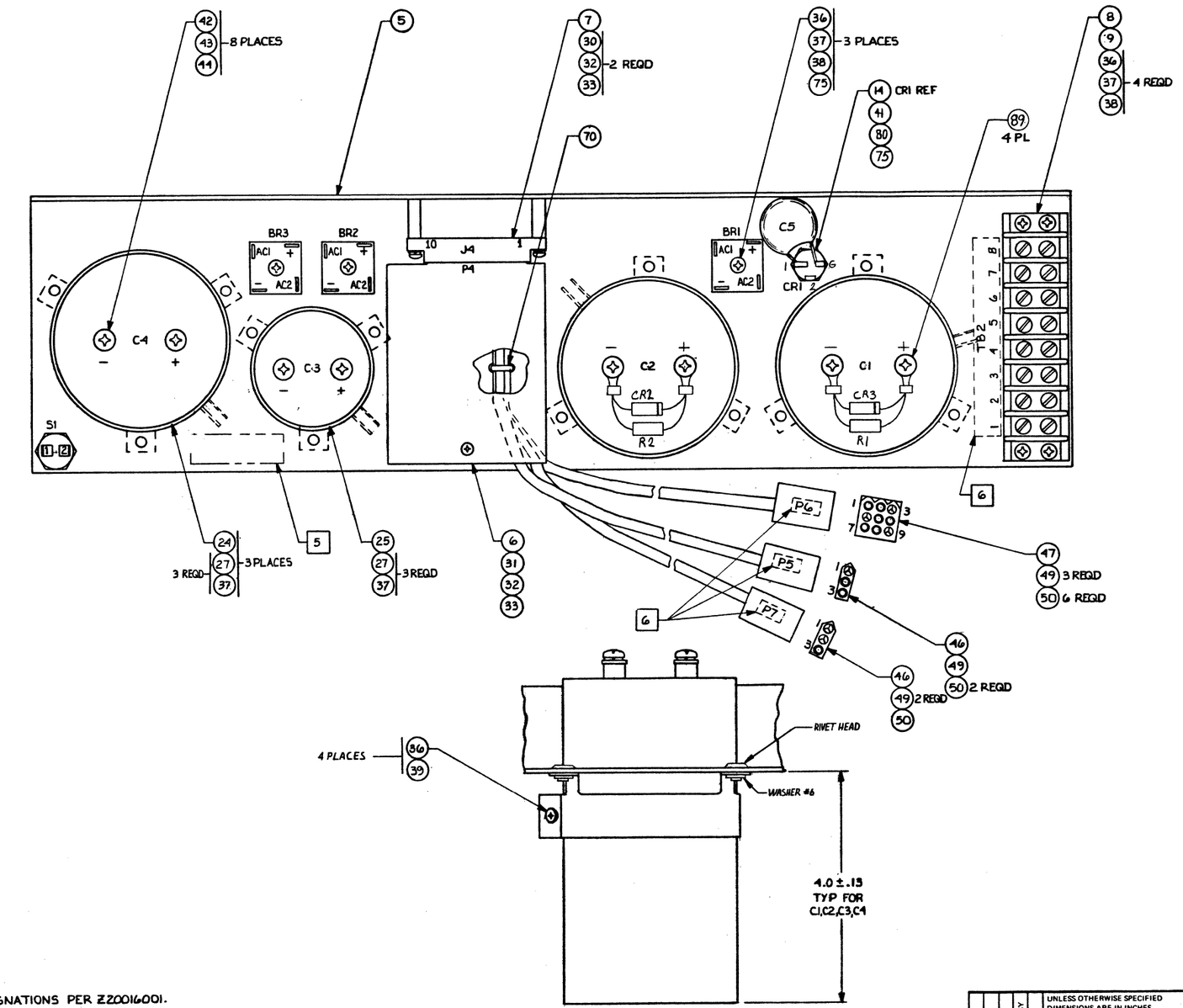
-003

6-13

D	A20032038	C
DWG SIZE	SHEET 2 OF 2	REV

FORM NO. 9-135-1

REV	EO	ZONE	DESCRIPTION	DWN	CHKD	APPD	DATE
AX	10081		PROTO RELEASE	[illegible]	mP	[illegible]	11-5-80
A	10177		PILOT RELEASE	[illegible]	mP	[illegible]	1-16-81
B	10334		INCORP EO	SPD	mP	[illegible]	9-25-81
C	10356		INCORP EO	SPD	mP	[illegible]	9-25-81
D	10542		INCORP EO	SPD	mP	[illegible]	9-25-81
E	10666		INCORP EO	SPD	mP	[illegible]	9-25-81
F	10734		INCORP EO	SPD	mP	[illegible]	3-17-82



6 MARK REF DESIGNATIONS PER Z20016001.

5 MARK PER Z20016001 WITH ASSY NO., REV LTR & SERIAL NO.

4. FOR WIRE LIST SEE WL20032043.
3. FOR SCHEMATIC SEE SC20032043.
2. ASSEMBLE PER MICRODATA WORKMANSHIP MANUAL.
1. INTERPRET DRAWING PER MIL-STD-100.

NOTES: UNLESS OTHERWISE SPECIFIED

WIRED FOR 220V & 240V	A20032043-002	C
WIRED FOR 100V & 115V	A20032043-001	085
DESCRIPTION	ASSY NO.	REV LTR

ASSY TABULATION CHART

6-15

TABULATED

UNINCORPORATED ENGINEERING ORDERS

THE INFORMATION CONTAINED HEREIN IS PROPRIETARY TO AND CONSIDERED A TRADE SECRET OF MICRODATA CORPORATION AND SHALL NOT BE REPRODUCED, IN WHOLE OR PART, WITHOUT THE WRITTEN AUTHORIZATION OF MICRODATA CORPORATION.

NEXT ASSY

USED ON

APPLICATION

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES
TOLERANCES .XX ± .02
TOLERANCES .XXX ± .010
TOLERANCES ANGLES ±½°

MATERIAL

FINISH

DRAFTSMAN — DATE 10-21-80
CHECKER — 11-3-80
ENGINEER — 11/4/80
APPD — 11/4/80
APPD — 1-13-81
APPD — 1-13-81

TITLE: AC INPUT ASSY

SCALE 1:1 | IDENT CODE 52936 | DWG SIZE D | SHEET 1 OF 1 | REV F

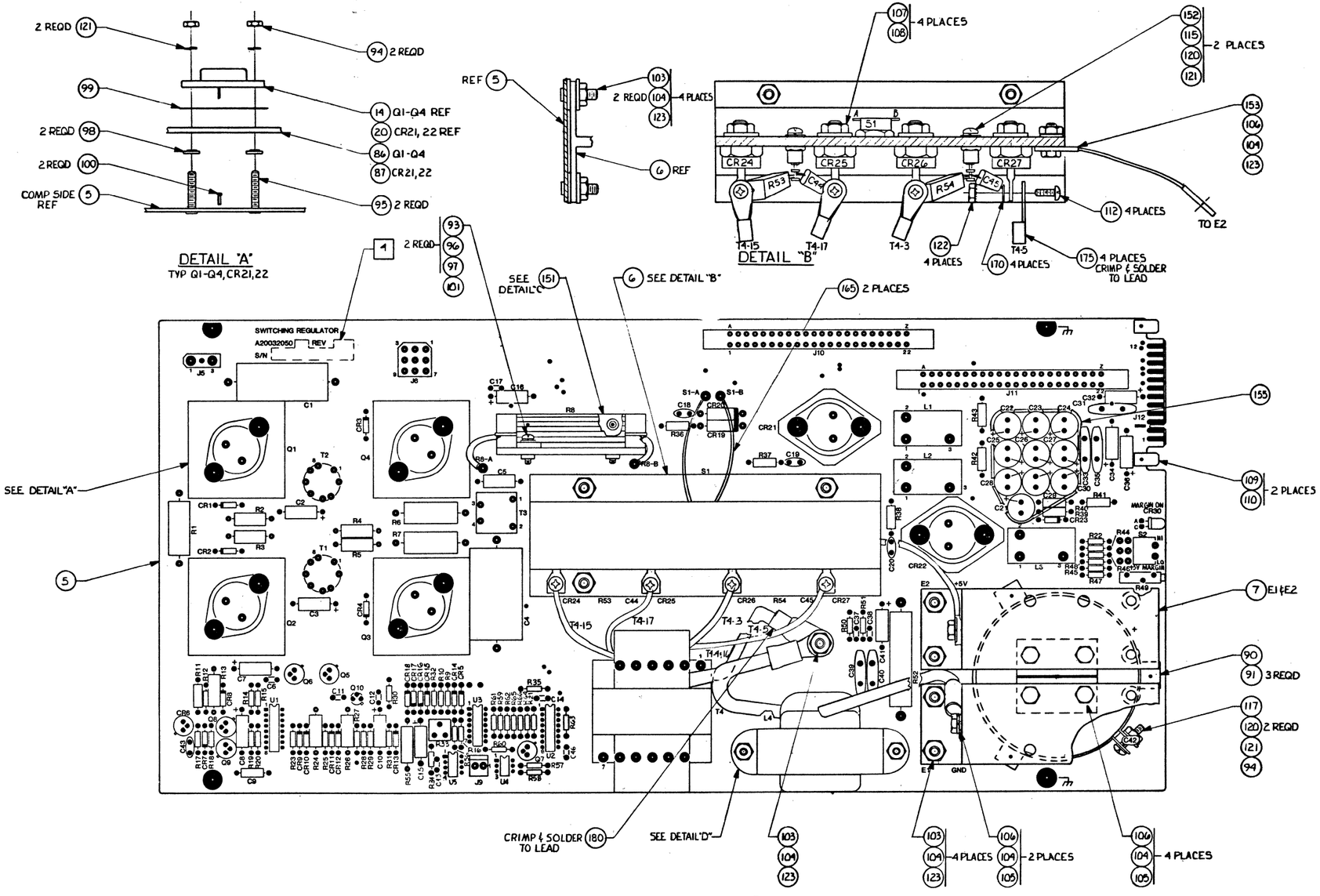
Microdata
IRVINE, CALIFORNIA

D A20032043 F

A20032043 B F

FORM NO 9-135

REV	EO	ZONE	DESCRIPTION	DWN	CHKD	APPD	DATE
AX	10172		PROTO RELEASE	[illegible]	MP	[illegible]	10-31-80
A	10165		PILOT RELEASE	[illegible]	MP	[illegible]	1-15-81



A20032050 B
A

A20032050	A
ASSY NO.	REV LTR
ASSY REV CHART	

6-17

UNINCORPORATED ENGINEERING ORDERS	
10424B	10674
10539	10795B
10611A	10939
	11050A

A

THE INFORMATION CONTAINED HEREIN IS PROPRIETARY TO AND CONSIDERED A TRADE SECRET OF MICRODATA CORPORATION AND SHALL NOT BE REPRODUCED, IN WHOLE OR PART, WITHOUT THE WRITTEN AUTHORIZATION OF MICRODATA CORPORATION.

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES
TOLERANCES .XX ± .02
TOLERANCES .XXX ± .010
TOLERANCES ANGLES ±½°
MATERIAL
FINISH
NEXT ASSY
USED ON
APPLICATION

DRAFTSMAN [illegible] DATE 10/1/80
CHECKED [illegible] 10/30/80
ENGINEER [illegible] 10/31/80
APPD [illegible] 10/31/80
APPD [illegible] 1-13-81
APPD [illegible] 1-13-81

TITLE
PCB ASSY
SWITCHING REGULATOR

SCALE 1:1 | IDENT CODE 52936 | DWG SIZE D | SHEET 1 OF 1 | REV A

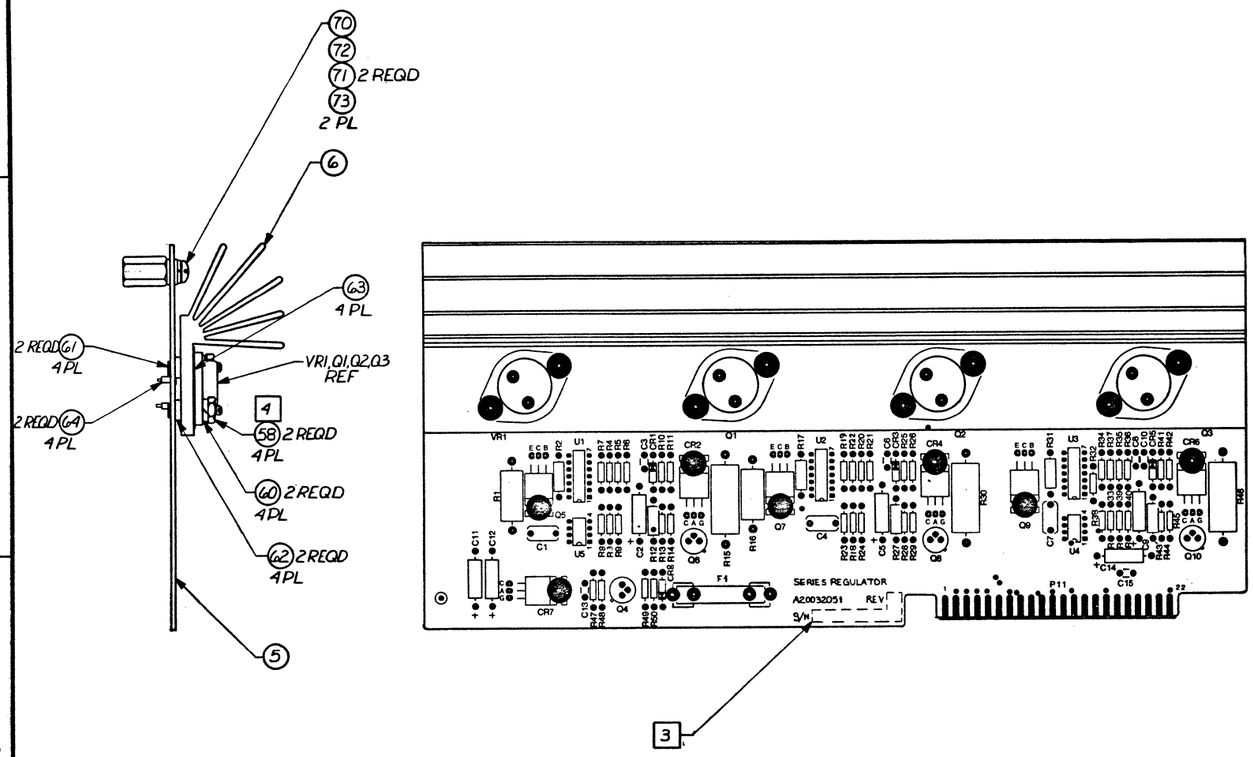
Microdata
IRVINE, CALIFORNIA

D A20032050 A

4 MARK PER Z2001G001 WITH REVISION LETTER AND SERIAL NO.
3. FOR SCHEMATIC SEE SC20032050.
2. ASSEMBLE PER MICRODATA WORKMANSHIP MANUAL.
1. INTERPRET DRAWING PER MIL-STD-100.
NOTES: UNLESS OTHERWISE SPECIFIED

FORM NO 9-135

REV	EO	ZONE	DESCRIPTION	DWN	CHKD	APPD	DATE
AX	9738		PROTO RELEASE	MJS	FO	RFS	5-2880
AXI	9919		REVISED & REDRAWN	MJS	MP	[illegible]	6-1840
A	10161		PILOT RELEASE	MJS	MP	[illegible]	1-1581



4 TORQUE TO 8 INCH POUNDS.
3 MARK PER Z20016001 WITH REV LTR AND SERIAL NO.
2. ASSEMBLE PER MICRODATA WORKMANSHIP MANUAL.
1. INTERPRET DRAWING PER MIL-STD-100.
NOTES: UNLESS OTHERWISE SPECIFIED

A20032051 A

ASSY NO.	REV LTR
ASSY REV CHART	6-19

THE INFORMATION CONTAINED HEREIN IS PROPRIETARY TO AND CONSIDERED A TRADE SECRET OF MICRODATA CORPORATION AND SHALL NOT BE REPRODUCED, IN WHOLE OR PART, WITHOUT THE WRITTEN AUTHORIZATION OF MICRODATA CORPORATION.

UNINCORPORATED ENGINEERING ORDERS
10358A
10485
11055

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES
TOLERANCES XX ± 02
TOLERANCES XXX ± 010
TOLERANCES ANGLES ±½°
MATERIAL
FINISH
NEXT ASSY
USED ON
APPLICATION

	DATE
DRAFTSMAN M. STEM	4-28-80
CHECKED F. OVERTON	5-12-80
ENGINEER N. SCOTT	5-22-80
APPD V. TULLAI	527-80
APPD Brock	1-13-81
APPD [illegible]	1-13-81
APPD	

TITLE
PCB ASSY
SERIES REGULATOR

SCALE 1:1 IDENT CODE 52936

Microdata
IRVINE, CALIFORNIA

D A20032051 A
DWG SIZE SHEET 1 OF 1 REV

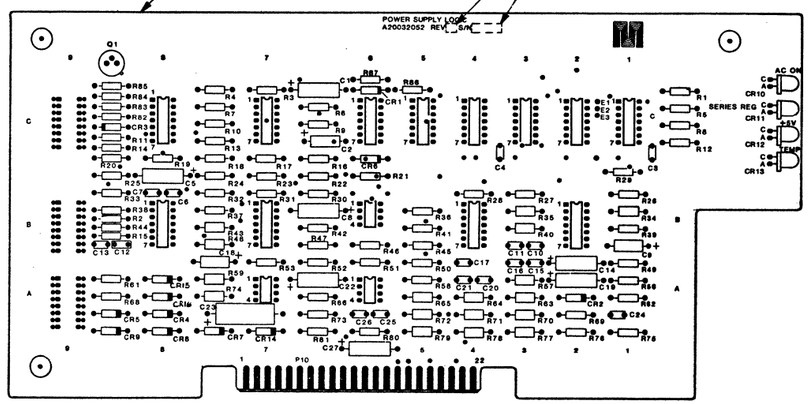
FORM NO 9-135

REV	EO	ZONE	DESCRIPTION	DWN	CHKD	APPD	DATE
AX	9762		PROTO RELEASE	JEC	FO	RFS	6-5-80
AXI	9932		REVISED & REDRAWN	[illegible]	mP	RFS	9/12/80
A	10139		PILOT RELEASE	[illegible]	mP	[illegible]	1-15-81

(85) 2 PLACES

(5)

3



A20032052 B

A

A20032052	A
ASSY NO	REV LTR
ASSY REV CHART	6-21

THE INFORMATION CONTAINED HEREIN IS PROPRIETARY TO AND CONSIDERED A TRADE SECRET OF MICRODATA CORPORATION AND SHALL NOT BE REPRODUCED, IN WHOLE OR PART, WITHOUT THE WRITTEN AUTHORIZATION OF MICRODATA CORPORATION.

UNINCORPORATED ENGINEERING ORDERS
10412 A
10919 A

3 MARK PER Z20016001 WITH REV LTR AND SERIAL NO.
2. ASSEMBLE PER MICRODATA WORKMANSHIP MANUAL.
1. INTERPRET DRAWING PER MIL-STD-100.
NOTES: UNLESS OTHERWISE SPECIFIED

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES
TOLERANCES XX ± 02
TOLERANCES XXX ± 010
TOLERANCES ANGLES ±½°
MATERIAL
FINISH

NEXT ASSY
USED ON
APPLICATION

	DATE
DRAFTSMAN J. CANTRELL	6-3-80
CHECKER F. GIVERTON	6-4-80
ENGINEER N. SCOTT	6-4-80
APPD J. TULLAI	6-4-80
APPD Brock	1-13-81
APPD [illegible]	1-13-81
APPD	

TITLE
PCB ASSY
POWER SUPPLY LOGIC

SCALE 1:1 | IDENT CODE 52936 | DWG SIZE D | SHEET 1 OF 1

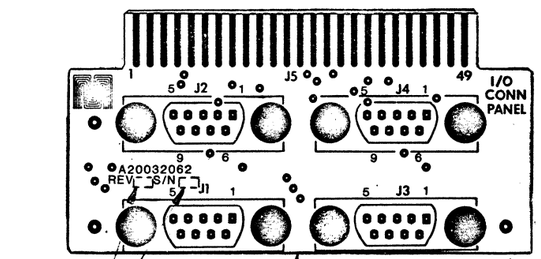
Microdata
IRVINE, CALIFORNIA

D A20032052 A
REV

FORM NO 5-135

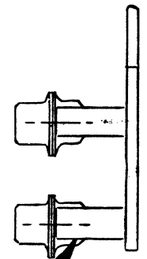
A20032062 A

REV	EO	ZONE	DESCRIPTION	DWN	CHKD	APPD	DATE
AX	9611		PROTO RELEASE	KCW	F.O.	H.B.G	3-12-80
A	10001		PILOT RELEASE	CW	mP	HBG	10-30-80



3

6



8 8 REQ'D

SWAGE IN PLACE

A20032062	A
ASSEMBLY NO.	REV LTR
ASSEMBLY REV CHART	

6-23

3. MARK PER Z20016001 WITH REV LETTER AND SERIAL NO.
2. ASSEMBLE PER MICRODATA WORKMANSHIP MANUAL.
1. INTERPRET DWG PER MIL-STD-100.

NOTES: UNLESS OTHERWISE SPECIFIED

THE INFORMATION CONTAINED HEREIN IS PROPRIETARY TO AND CONSIDERED A TRADE SECRET OF MICRODATA CORPORATION AND SHALL NOT BE REPRODUCED, IN WHOLE OR PART, WITHOUT THE WRITTEN AUTHORIZATION OF MICRODATA CORPORATION.

UNINCORPORATED ENGINEERING ORDERS

NEXT ASSY | USED ON | APPLICATION

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES
TOLERANCES .XX ± .02
TOLERANCES .XXX ± .010
TOLERANCES ANGLES ±½°

MATERIAL

FINISH

		DATE
DRAFTSMAN	K.C. WILTON	2-29-80
CHECKER	FRANKIE OVERTON	3-11-80
ENGINEER	R. KENT	3-12-80
APPD	W SCHRADER	3-12-80
APPD	R. Bonch	10/29/80
APPD	[signature]	10/29/80
APPD		

TITLE

PCB ASSEMBLY - I/O CONNECTOR PANEL

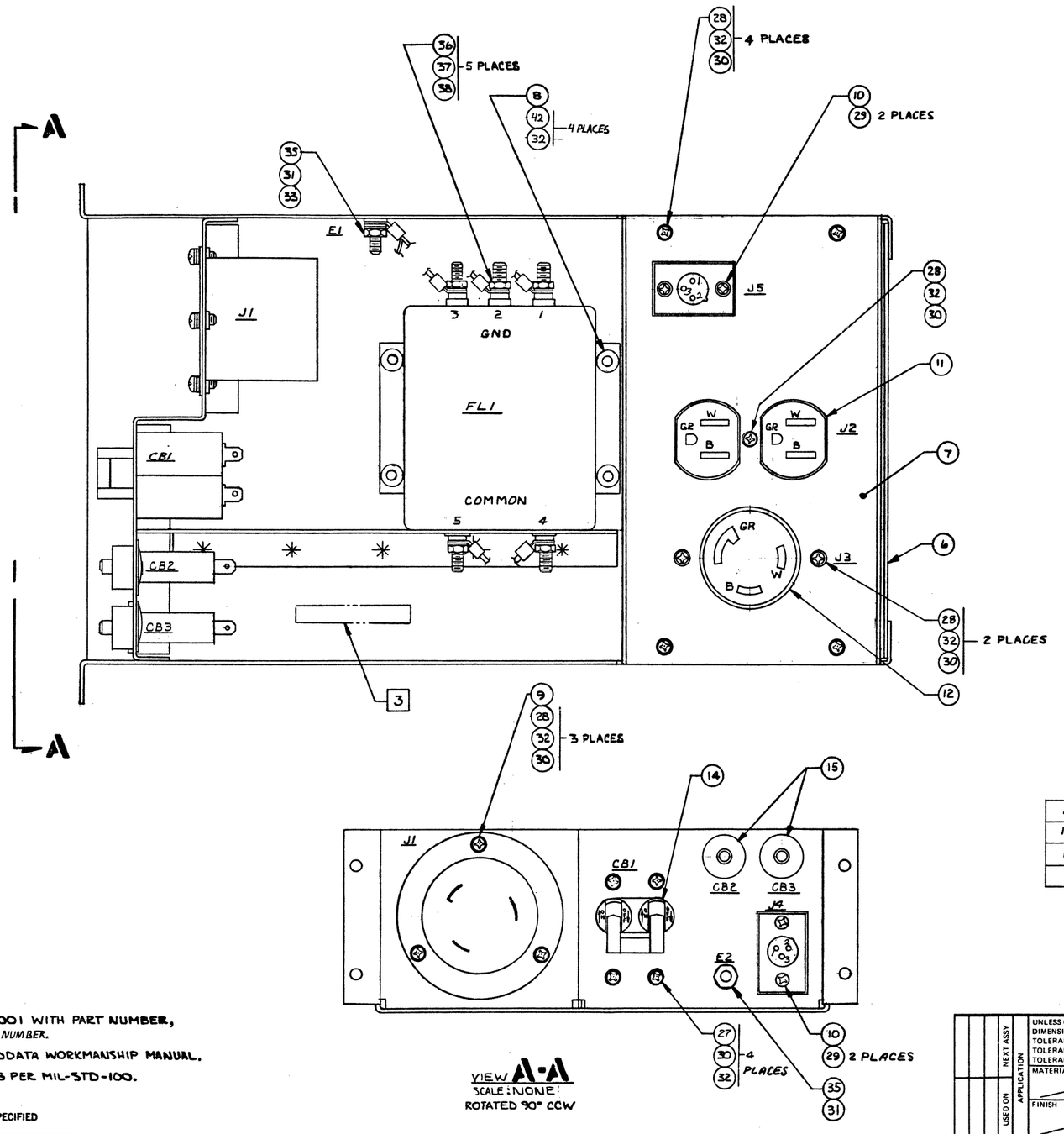
Microdata
IRVINE, CALIFORNIA

B	A20032062	A

SCALE	1:1	IDENT CODE	52936	DWG SIZE	SHEET 1 OF 1	REV

FORM NO. 9-137

REV	EO	ZONE	DESCRIPTION	DWN	CHKD	APPD	DATE
AX	9760		PROTOTYPE RELEASE	RB			4/7/80
AX1	9955		INCORP E.O.				9/9/80
AX2	10061		INCORP E.O.				11-3-80
A	10176		PILOT RELEASE				1-15-81
B	10476A		INCORP EO	SPD			9-3-81



VIEW A-A
SCALE: NONE
ROTATED 90° CCW

3. MARK PER 220016001 WITH PART NUMBER, REVISION LETTER & SERIAL NUMBER.
2. ASSEMBLE PER MICRODATA WORKMANSHIP MANUAL.
1. INTERPRET DRAWING PER MIL-STD-100.

NOTES: UNLESS OTHERWISE SPECIFIED

A20032063 B

B

240V, 50/60 HZ	A20032063-001	B
115V, 50/60 HZ	A20032063	B
DESCRIPTION	PART NUMBER	REV LTR
ASSEMBLY TABULATION CHART		

TABULATED 6-25

THE INFORMATION CONTAINED HEREIN IS PROPRIETARY TO AND CONSIDERED A TRADE SECRET OF MICRODATA CORPORATION AND SHALL NOT BE REPRODUCED, IN WHOLE OR PART, WITHOUT THE WRITTEN AUTHORIZATION OF MICRODATA CORPORATION.

UNINCORPORATED ENGINEERING ORDERS

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES
TOLERANCES XX ± .02
TOLERANCES XXX ± .010
TOLERANCES ANGLES ±½°
MATERIAL
FINISH
NEXT ASSY
USED ON
APPLICATION

DRAFTSMAN R. Bielschke DATE 5-10-80
CHECKER [illegible] 5/14/80
ENGINEER [illegible] 6/4/80
APPD [illegible] 6/4/80
APPD [illegible] 1/9/81
APPD [illegible] 1-9-81

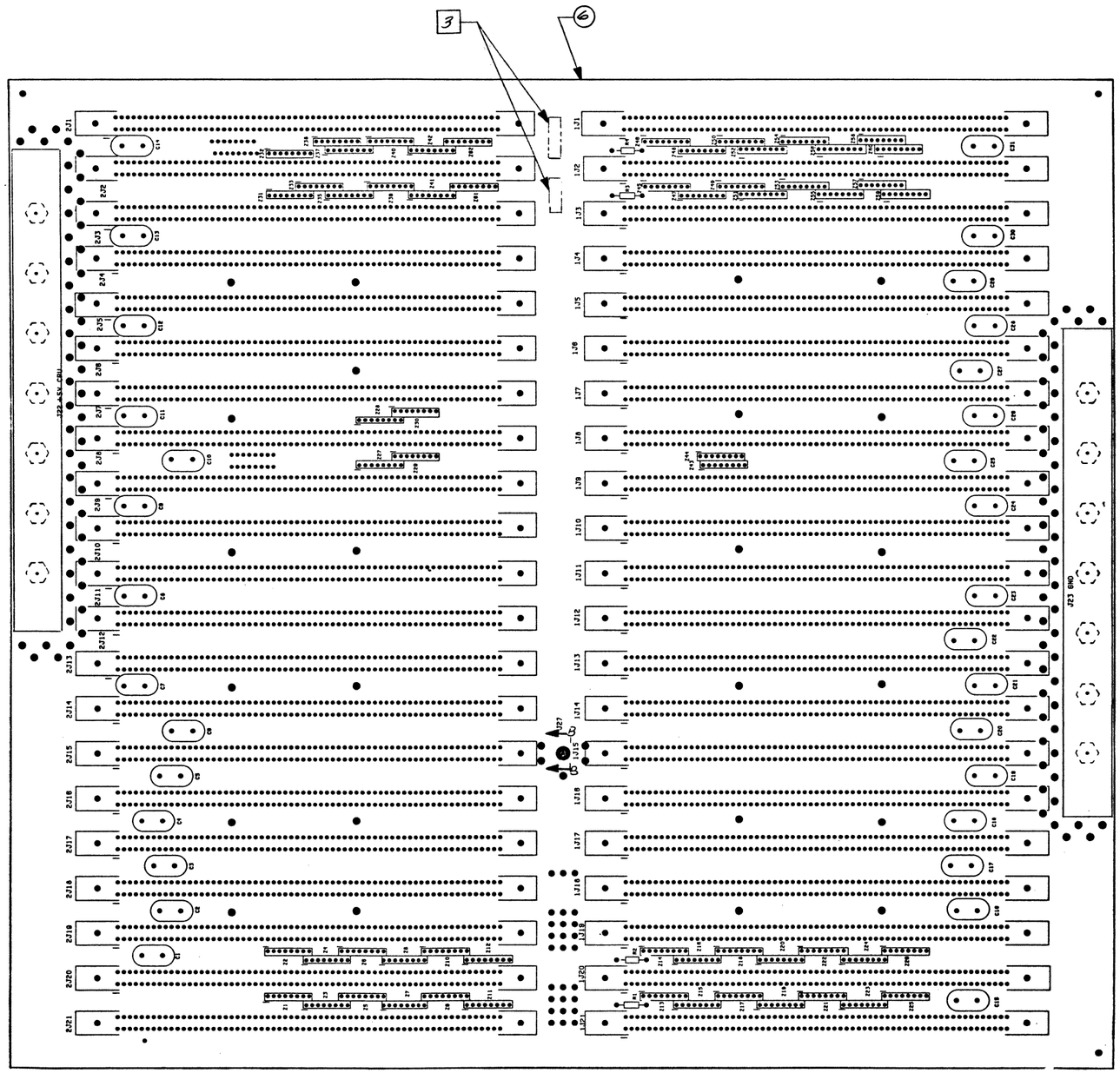
TITLE
AC PWR MODULE ASSEMBLY

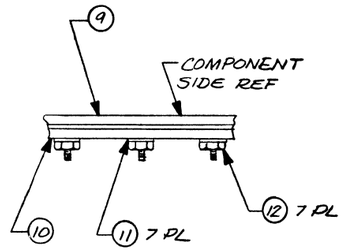
SCALE 1/1 & NOTED | IDENT CODE 52936 | DWG SIZE D | A20032063 | B | SHEET 1 OF 1 | REV

Microdata
IRVINE, CALIFORNIA

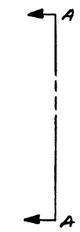
FORM NO. 9-136

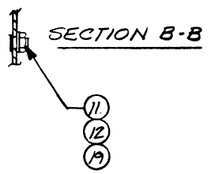
REV	EO	ZONE	DESCRIPTION	DWN	CHKD	APPD	DATE
A	9984		PILOT RELEASE	RH	MP	[illegible]	9-7-82





VIEW A-A
2 PLACES
ROTATED 90° CW





SECTION B-B

A20032065 B
A

A20032065	A
ASSEMBLY NO	REV
ASSEMBLY REV CHART	

6-27

THE INFORMATION CONTAINED HEREIN IS PROPRIETARY TO AND CONSIDERED A TRADE SECRET OF MICRODATA CORPORATION AND SHALL NOT BE REPRODUCED, IN WHOLE OR PART, WITHOUT THE WRITTEN AUTHORIZATION OF MICRODATA CORPORATION.

UNINCORPORATED ENGINEERING ORDERS	
10211 A	10650
10286	11080
9722	

3 MARK PER Z20016001 WITH REV LTR AND S/N.

2 ASSEMBLE PER MICRODATA WORKMANSHIP MANUAL.

1 INTERPRET DRAWING PER MIL-STD-100.

NOTES: UNLESS OTHERWISE SPECIFIED

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES
TOLERANCES XX ± 02
TOLERANCES XXX ± 010
TOLERANCES ANGLES ±½°
MATERIAL
FINISH

USED ON NEXT ASSY
APPLICATION

DRAFTSMAN [illegible] DATE 9/8/82
CHECKER E CHERTEN 9/14/82
ENGINEER R.J. BERG 9/14/82
APPD M. SCHRADER 9/14/82
APPD L. BROCK 9/14/82
APPD [illegible] 9-14-82
APPD

TITLE
PCB ASSEMBLY-
BACKPLANE

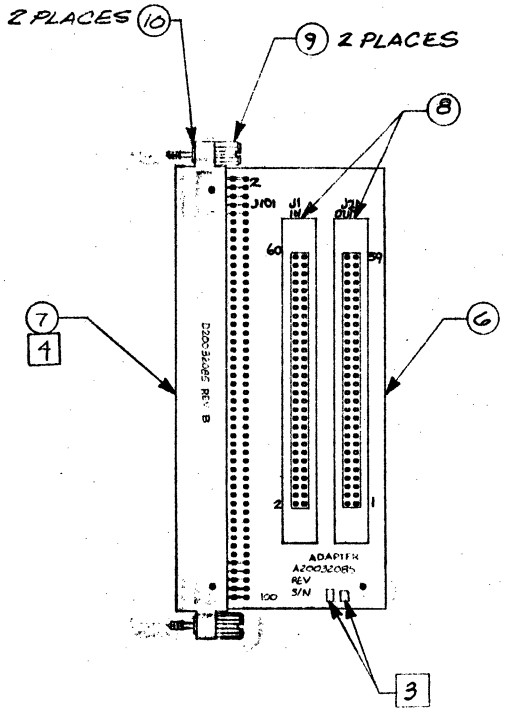
SCALE 1/1 IDENT CODE 52936

Microdata
IRVINE, CALIFORNIA

D	A20032065	A
DWG SIZE	SHEET 1 OF 1	REV

FORM NO 5-135

REV	EO	ZONE	DESCRIPTION	DWN	CHKD	APPD	DATE
AX	9754		PROTO RELEASE	[illegible]	-10	RJS	6/9/80
A	9961		PILOT RELEASE	CAD	-10	[illegible]	11-11-80
B	10755		INCORP EO	SPD	MD	[illegible]	3-17-82
C	10820		INCORP EO	SPD	MD	[illegible]	3-17-82



A200032085
C

6-29

4 USE SILKSCREEN CONNECTOR PIN NUMBERS ON PC. BOARD WHEN REFERRING TO CONNECTOR NUMBERING FROM SCHEMATIC

3 MARK PER 2200/6001 WITH SERIAL NO. AND REVISION LETTER.

2. ASSEMBLE PER MICRODATA WORKMANSHIP MANUAL.

1. INTERPRET DRAWING PER MIL-STD-100.

NOTES: UNLESS OTHERWISE SPECIFIED

A200032085	B
ASSEMBLY NO	REV
ASSEMBLY REV CHART	

THE INFORMATION CONTAINED HEREIN IS PROPRIETARY TO AND CONSIDERED A TRADE SECRET OF MICRODATA CORPORATION AND SHALL NOT BE REPRODUCED IN WHOLE OR PART, WITHOUT THE WRITTEN AUTHORIZATION OF MICRODATA CORPORATION.

UNINCORPORATED ENGINEERING ORDERS

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES
TOLERANCES XX = .02
TOLERANCES XXX = .010
TOLERANCES ANGLES =
MATERIAL
FINISH
NEXT ASSY
APPLICATION

DRAWN [illegible] 6/9/80
CHECKED
ENGR'G [illegible] 6/10/80
APPD
APPD [illegible] 6/10/80
APPD
APPD

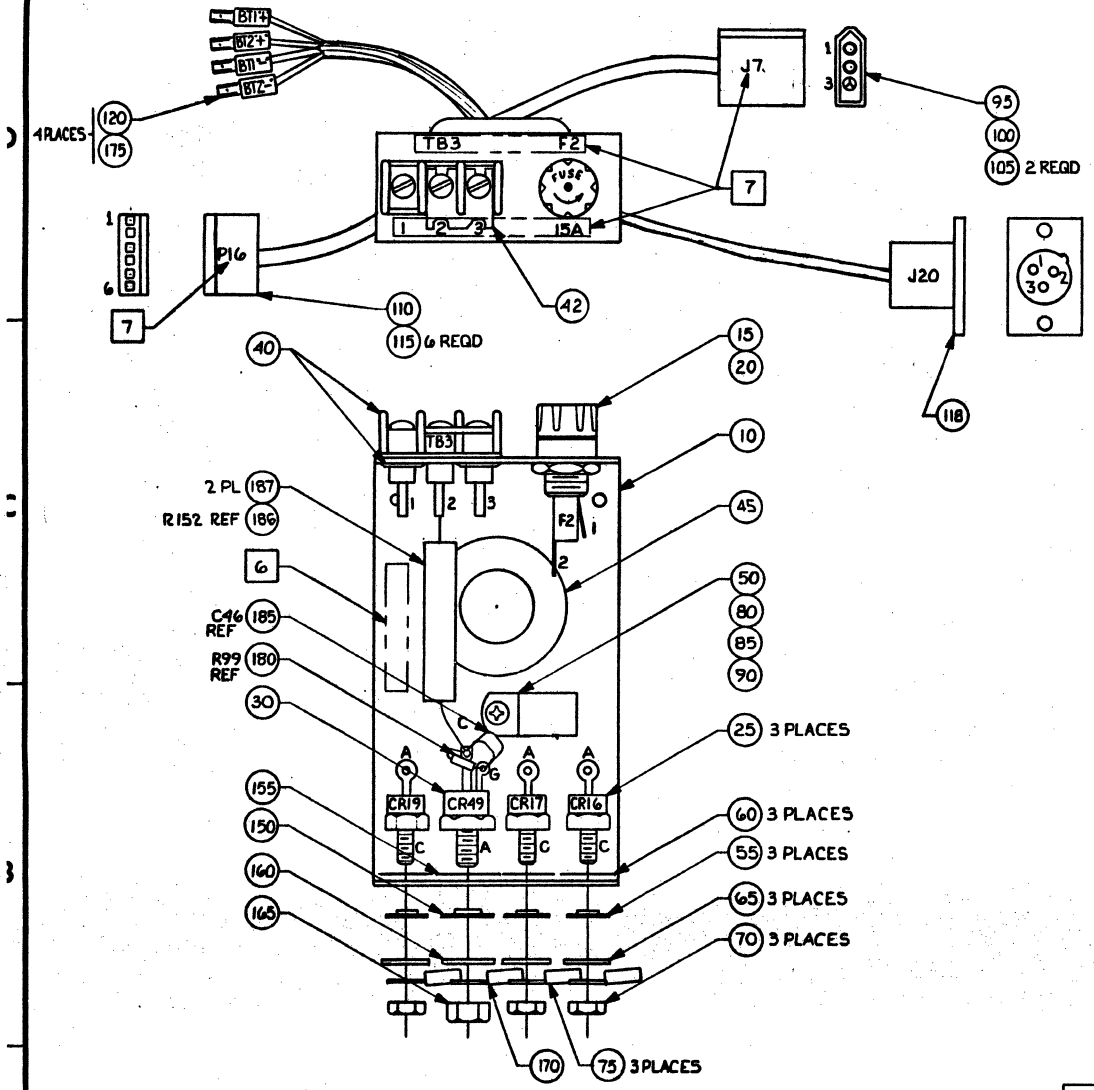
TITLE: PCB ASSEMBLY ADAPTER MTU CONTROLLER

Microdata
IRVINE, CALIFORNIA

C	A200032085	C

SCALE 1/1 | IDENT CODE 52936 | SIZE | SHEET 1 OF 1 | REV

REV	EO	ZONE	DESCRIPTION	DWN	CHKD	APPD	DATE
AX	10087		PROTO RELEASE	[illegible]	MP	[illegible]	11-7-80
A	10166		PILOT RELEASE	[illegible]	MP	[illegible]	1-16-81
B	10753		INCORP EO	G.L.			
C	10784		INCORP EO	G.L.			
D	10818A		INCORP EO	G.L.			



A20032101

A

A20032101	D
A20032101	A
ASSY NO.	REV LTR

ASSY REV CHART

7 MARK REF DESIGNATIONS PER Z20016001.

6 MARK PER Z20016001 WITH ASSY NO., REV LTR & SERIAL NO.

5. J7, P16, J20 & ALL WIRING REMOVED FOR CLARITY ON FRONT VIEW.
4. FOR WIRE LIST SEE WL20032101.
3. FOR SCHEMATIC SEE SC20032053.
2. ASSEMBLE PER MICRODATA WORKMANSHIP MANUAL.
1. INTERPRET DRAWING PER MIL-STD-100.

NOTES: UNLESS OTHERWISE SPECIFIED

THE INFORMATION CONTAINED HEREIN IS PROPRIETARY TO AND CONSIDERED A TRADE SECRET OF MICRODATA CORPORATION AND SHALL NOT BE REPRODUCED, IN WHOLE OR PART, WITHOUT THE WRITTEN AUTHORIZATION OF MICRODATA CORPORATION.

UNINCORPORATED ENGINEERING ORDERS

NEXT ASSY

USED ON

APPLICATION

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES
TOLERANCES .XX ± .02
TOLERANCES .XXX ± .010
TOLERANCES ANGLES ±½°
MATERIAL
FINISH

DRAFTSMAN [illegible] DATE 10.23.80
CHECKED [illegible] 11/3/80
ENGINEER [illegible] 11/4/80
APPD [illegible] 11/6/80
APPD [illegible] 1-13-81
APPD [illegible] 1-13-81
APPD

TITLE

FUSE ASSY
MEM PWR MOD

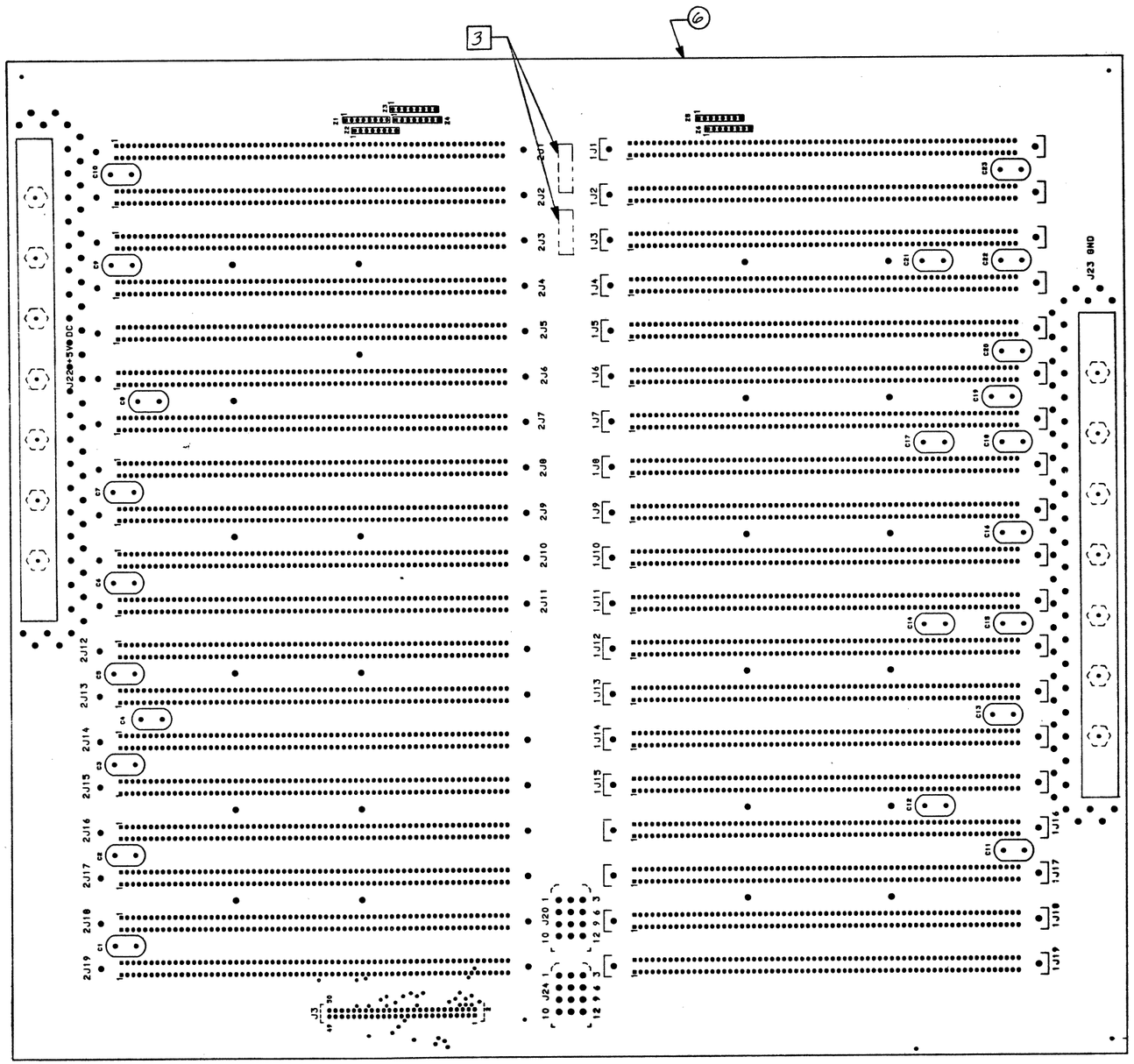
Microdata
IRVINE, CALIFORNIA

SCALE 1:1 | IDENT CODE 52936 | C | A20032101 | A. | DWG SIZE | SHEET 1 OF 1 | REV

FORM 8-136-2

4 3 2 1

REV	EO	ZONE	DESCRIPTION	DWN	CHKD	APPD	DATE
AX	10091		PROTO RELEASE	[illegible]	MP	[illegible]	11-12-80
A	10220		PILOT RELEASE	TSH	R.V.	[illegible]	1-29-81



A20032117 B
A

A20032117	A
ASSEMBLY NO	REV
ASSEMBLY REV CHART	

6-31

THE INFORMATION CONTAINED HEREIN IS PROPRIETARY TO AND CONSIDERED A TRADE SECRET OF MICRODATA CORPORATION AND SHALL NOT BE REPRODUCED, IN WHOLE OR PART, WITHOUT THE WRITTEN AUTHORIZATION OF MICRODATA CORPORATION.

UNINCORPORATED ENGINEERING ORDERS
10682

A

[3] MARK PER Z2001600I WITH REVISION LETTER AND SERIAL NUMBER.

2 ASSEMBLE PER MICRODATA WORKMANSHIP MANUAL.

1 INTERPRET DRAWING PER MIL-STD-100

NOTES: UNLESS OTHERWISE SPECIFIED

NEXT ASSY
USED ON
APPLICATION

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES
TOLERANCES XX ± 02
TOLERANCES XXX ± 010
TOLERANCES ANGLES ±°

MATERIAL
FINISH

DRAFTSMAN — DATE
CHECKER 11/10/80
ENGINEER 11/10/80
APPD 11/10/80
APPD 1/28/81
APPD 1-26-81
APPD

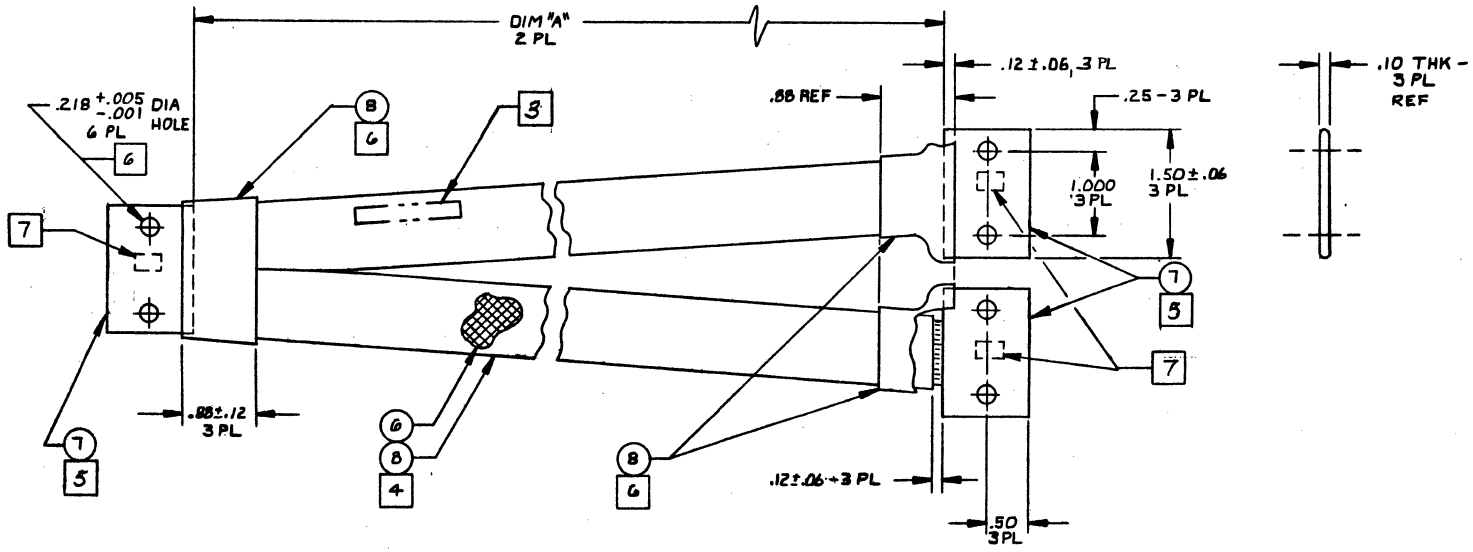
TITLE
PCB ASSEMBLY-EXPANSION BACKPLANE

Microdata
IRVINE, CALIFORNIA

SCALE 1/1 | IDENT CODE 52936 | D | A20032117 | A
DWG SIZE | SHEET 1 OF 1 | REV

FORM NO 5-135

REV	EO	ZONE	DESCRIPTION	DWN	CHKD	APPD	DATE
AX	10024		PROTO RELEASE	JEC			10/1/80
A	10181		PILOT RELEASE	CMB	MP		11-5-81
B	10462		INCORP EO	HM	MP		6-18-81



A20032118
B

A20032118-002	49.00 ± 2.00	B
A20032118-001	38.00 ± 1.00	B
ASSEMBLY NO.	LENGTH (DIM "A")	REV. LTR.

ASSEMBLY TABULATION CHART

TABULATED

7. POLARITY TO BE MARKED "+" FOR CONFIG. -002 & "-" FOR -001.
6. APPLY TUBING OVER CABLE ENDS & HEATSHRINK.
5. SWAGE TERMINALS ONTO THE CABLE ENDS, AS SHOWN.
4. APPLY TUBING ONTO CABLES & HEATSHRINK, FULL LENGTHS ONLY.
3. MARK PER Z20016001 WITH PART NO., DASH NO., REV. LTR.
2. ASSEMBLE PER MICRODATA WORKMANSHIP MANUAL.
1. INTERPRET DRAWING PER MIL-STD-100.

NOTES: UNLESS OTHERWISE SPECIFIED

THE INFORMATION CONTAINED HEREIN IS PROPRIETARY TO AND CONSIDERED A TRADE SECRET OF MICRODATA CORPORATION AND SHALL NOT BE REPRODUCED, IN WHOLE OR PART, WITHOUT THE WRITTEN AUTHORIZATION OF MICRODATA CORPORATION.

UNINCORPORATED ENGINEERING ORDERS

NEXT ASSY | USED ON | APPLICATION

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES
TOLERANCES .XX ± .02
TOLERANCES .XXX ± .010
TOLERANCES ANGLES ±½°

MATERIAL

FINISH

DRAFTSMAN S. Cantrell 8/27/80
CHECKER [illegible] 9/17/80
ENGINEER [illegible] 9/19/80
APPD [illegible] 10-8-80
APPD [illegible] 1-13-81
APPD [illegible] 1-13-81
APPD

TITLE
CABLE ASSY — DC PWR/GND

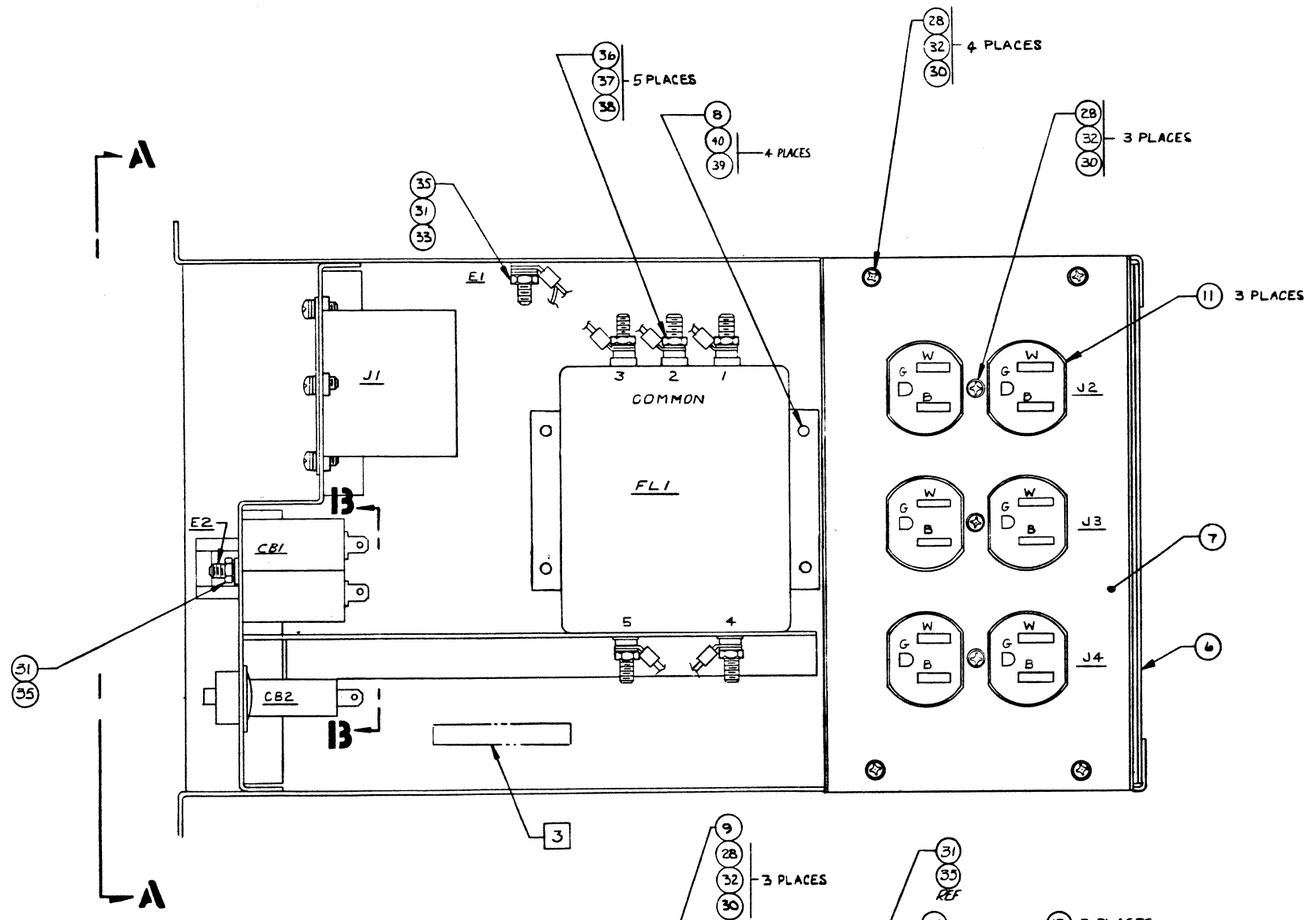
Microdata
IRVINE, CALIFORNIA

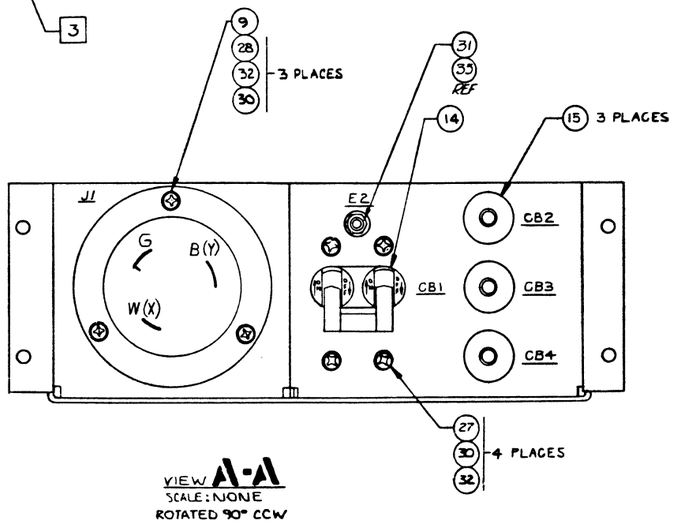
SCALE NONE | IDENT CODE 52936 | DWG SIZE C | A20032118 | B | SHEET 1 OF 1 | REV

FORM 9-136-3

6-33

REV	EO	ZONE	DESCRIPTION	DWN	CHKD	APPD	DATE
AX	10073		PROTOTYPE RELEASE				11-3-80
A	10158		PILOT RELEASE				1-14-81
B	10591		INCORP EO	SPD			11-12-81
C	10876		INCORP EO	SPD			





VIEW A-A
SCALE: NONE
ROTATED 90° CCW

4. WIRE PER WIRE LIST WL20032132.
3. MARK PER Z20016001 WITH PART NUMBER, REVISION LETTER AND SERIAL NUMBER.
2. ASSEMBLE PER MICRODATA WORKMANSHIP MANUAL.
1. INTERPRET DRAWING PER MIL-STD-100.

NOTES: UNLESS OTHERWISE SPECIFIED

A20032132 B
C

240V, 50/60HZ	A20032132-001	B
115V, 50/60HZ	A20032132	B
DESCRIPTION	ASSY NO.	REV LTR

ASSEMBLY TABULATION CHART

6-35

TABULATED

THE INFORMATION CONTAINED HEREIN IS PROPRIETARY TO AND CONSIDERED A TRADE SECRET OF MICRODATA CORPORATION AND SHALL NOT BE REPRODUCED, IN WHOLE OR PART, WITHOUT THE WRITTEN AUTHORIZATION OF MICRODATA CORPORATION.

UNINCORPORATED ENGINEERING ORDERS

A

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES
TOLERANCES XX ±.02
TOLERANCES XXX ±.010
TOLERANCES ANGLES ±½°
MATERIAL
FINISH
NEXT ASSY
USED ON
APPLICATION

DRAWN 8/21/80
CHECKED 10/17/80
ENGINEER
APPD 11/9/81
APPD 11/9/81
APPD 1-9-81
APPD

TITLE
AC PWR EXPANSION MODULE ASSEMBLY

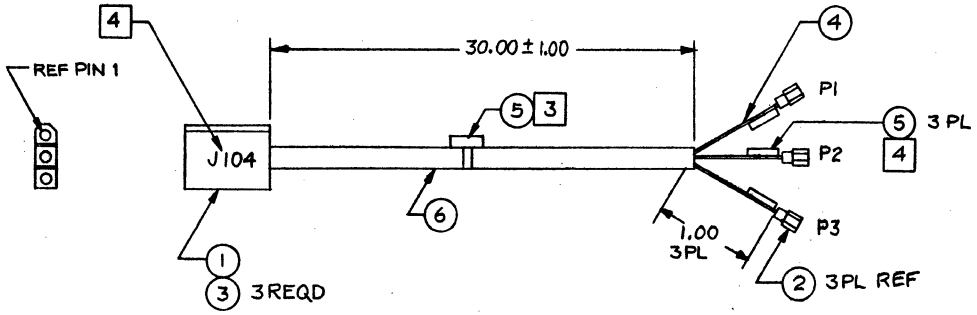
SCALE 1/1 NOTED
IDENT CODE 52936

Microdata
IRVINE, CALIFORNIA

D	A20032132	C
DWG SIZE	SHEET 1 OF 1	REV

FORM NO 9-138

REV	EO	ZONE	DESCRIPTION	DWN	CHKD	APPD	DATE
AX	10078		PROTOTYPE RELEASE	PW			11-4-80
A	10147		PILOT RELEASE	SEC	MP		1/13/81



CONTACT ITEM NO.	J104		TERM ITEM NO.
3	1	P1	2
3	2	P2	2
3	3	P3	2

WIRE LIST

A20032162 A

4 PERMANENTLY IDENTIFY REFERENCE DESIGNATORS J104 & P1, P2, P3 USING .12 HIGH, GOTHIC, BLACK INK LETTERING, CENTERED APPROXIMATELY AS SHOWN.

3. MARK PER Z 20016001 WITH PART NO. & REVISION LETTER

2. ASSEMBLE PER MICRODATA WORKMANSHIP MANUAL

1. INTERPRET DRAWING PER MIL-STD-100

NOTES: UNLESS OTHERWISE SPECIFIED

A20032162	A
PART NO.	REV LTR
ASSEMBLY REV CHART	

THE INFORMATION CONTAINED HEREIN IS PROPRIETARY TO AND CONSIDERED A TRADE SECRET OF MICRODATA CORPORATION AND SHALL NOT BE REPRODUCED, IN WHOLE OR PART, WITHOUT THE WRITTEN AUTHORIZATION OF MICRODATA CORPORATION.

UNINCORPORATED ENGINEERING ORDERS

NEXT ASSY | USED ON | APPLICATION

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES
TOLERANCES .XX ± .02
TOLERANCES .XXX ± .010
TOLERANCES ANGLES ±½°
MATERIAL
FINISH

DRAFTSMAN P. Wallace 10/27/80
CHECKER 10/29/80
ENGINEER 10-31-80
APPD 10-31-80
APPD 1/9/81
APPD 1-9-81

TITLE
CABLE ASSY, INTERLOCK SWITCH

Microdata
IRVINE, CALIFORNIA

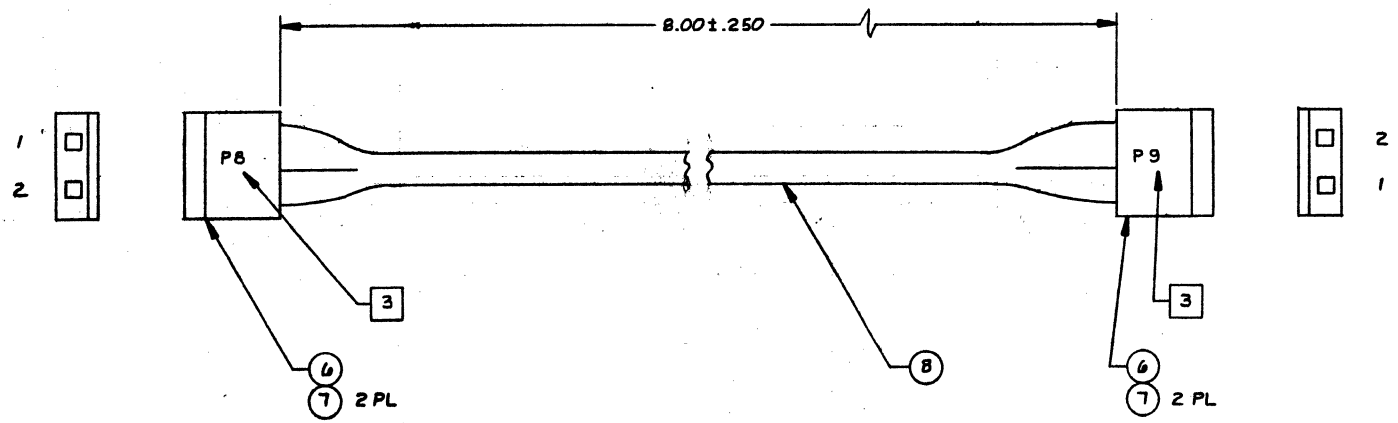
SCALE NONE | IDENT CODE 52936 | DWG SIZE C | A20032162 | A | SHEET 1 OF 1 | REV

6-37

FORM 9-186-5

REV	EO	ZONE	DESCRIPTION	DWN	CHKD	APPD	DATE
AX	10053		PROTO. RELEASE	DEC	[illegible]	[illegible]	10-27-80
A	10181		PILOT RELEASE	CMB	mP	[illegible]	1-15-81
B	10355		INCORP EO	DEC	mP	[illegible]	4-29-81

WIRE LIST	
FROM	TO
P8-1	P9-1
P8-2	P9-2



A20032163
B

4. BAG AND TAG PER MARKING SPEC. 220016001 WITH ASSY NO. AND REV LTR.
3. MARK REF DESIGNATIONS P8 & P9 PER 220016001.
2. ASSEMBLE PER MICRODATA WORKMANSHIP MANUAL.
1. INTERPRET DRAWING PER MIL-STD-100.

NOTES: UNLESS OTHERWISE SPECIFIED

A20032163	A
ASSEMBLY NUMBER	REV LTR
ASSEMBLY REVISION CHART	

THE INFORMATION CONTAINED HEREIN IS PROPRIETARY TO AND CONSIDERED A TRADE SECRET OF MICRODATA CORPORATION AND SHALL NOT BE REPRODUCED, IN WHOLE OR PART, WITHOUT THE WRITTEN AUTHORIZATION OF MICRODATA CORPORATION.

UNINCORPORATED ENGINEERING ORDERS

NEXT ASSY | USED ON | APPLICATION

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES
TOLERANCES .XX ± .02
TOLERANCES .XXX ± .010
TOLERANCES ANGLES ±½°
MATERIAL
FINISH

DRAFTSMAN J. CANTRELL DATE 10/22/80
CHECKER [illegible] 10/22/80
ENGINEER [illegible] 10/22/80
APPD [illegible] 10/24/80
APPD [illegible] 1-13-81
APPD [illegible] 1-13-81
APPD

TITLE: CABLE ASSY - P/S SYNC

SCALE 4:1 | IDENT CODE 52936

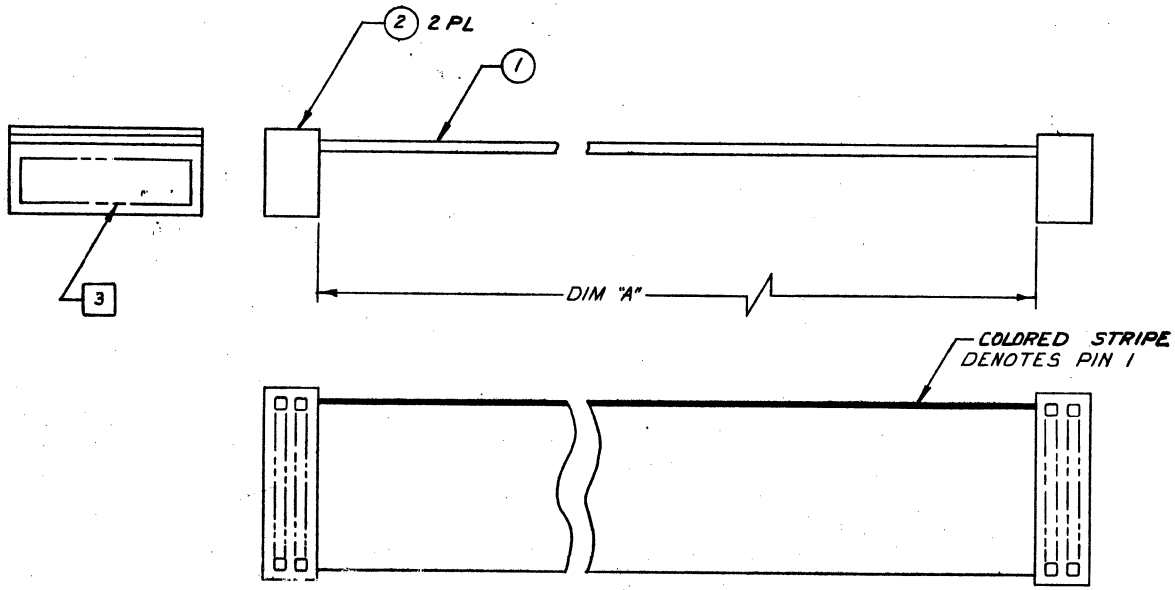
Microdata
IRVINE, CALIFORNIA

C	A20032163	B
DWG SIZE	SHEET 1 OF 1	REV

FORM 9-136-2

6-38

REV	EO	ZONE	DESCRIPTION	DWN	CHKD	APPD	DATE
AX	10131		PROTOTYPE RELEASE	AK	mP	[illegible]	12-3-80
A	10218		PILOT RELEASE	AK	mP	[illegible]	1-30-81



A20032168
A

6-39

A20032168-001	36.00 ± 1.00	A
ASSEMBLY NO.	DIM "A" (LENGTH)	REV LTR

ASSEMBLY TABULATION CHART

TABULATED

4. WIRING TO BE POINT TO POINT.

3. MARK PER Z20016001 WITH PART NUMBER, DASH NUMBER AND REVISION LETTER.

2. ASSEMBLE PER MICRODATA WORKMANSHIP MANUAL.

1. INTERPRET DRAWING PER MIL-STD-100.

NOTES: UNLESS OTHERWISE SPECIFIED

THE INFORMATION CONTAINED HEREIN IS PROPRIETARY TO AND CONSIDERED A TRADE SECRET OF MICRODATA CORPORATION AND SHALL NOT BE REPRODUCED, IN WHOLE OR PART, WITHOUT THE WRITTEN AUTHORIZATION OF MICRODATA CORPORATION.

UNINCORPORATED ENGINEERING ORDERS

NEXT ASSY | USED ON | APPLICATION

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES
TOLERANCES .XX ± .02
TOLERANCES .XXX ± .010
TOLERANCES ANGLES ±½°
MATERIAL
FINISH

		DATE
DRAFTSMAN	A. KULMA	11-26-80
CHECKER		11-26-80
ENGINEER		12-1-80
APPD		[illegible]
APPD		1/28/81
APPD		1-28-81
APPD		

TITLE: CABLE ASSY— I/O, 50 CONDUCTOR

Microdata
IRVINE, CALIFORNIA

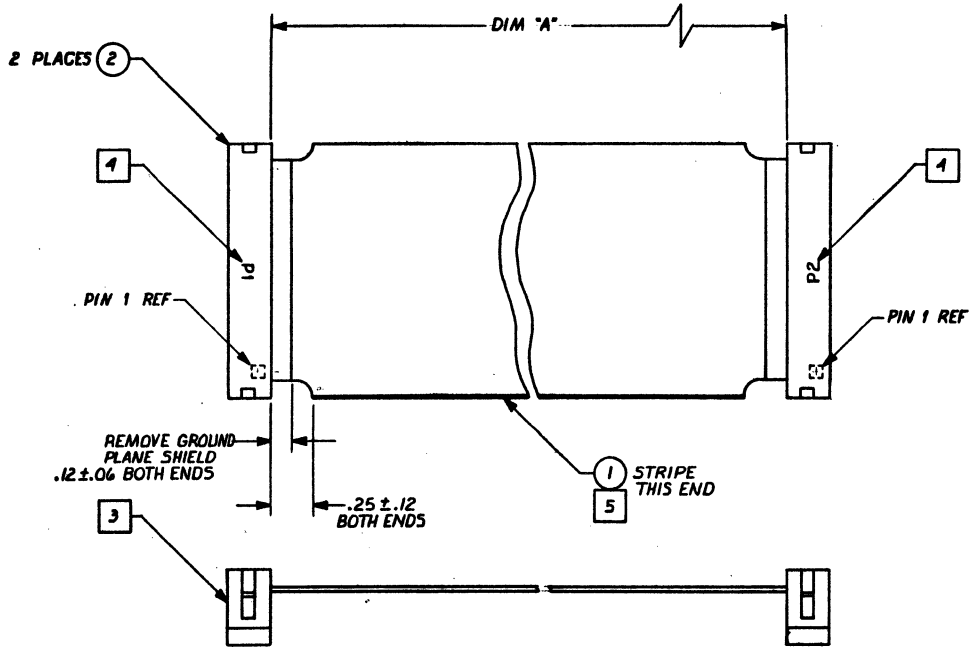
SCALE	IDENT CODE	DWG SIZE	SHEET	REV
NONE	52936	C	1 OF 1	A

C A20032168 A

FORM 9-1136-3

4 3 2 1

REV	EO	ZONE	DESCRIPTION	DWN	CHKD	APPD	DATE
A	10275		PILOT RELEASE	MJS	MP	JHH	3-2-81



A20032187

A

100 INCHES ±2 IN.	A20032187-001	A
DIM "A" (LENGTH)	ASSY NO.	REV LTR

ASSY TABULATION CHART

TABULATED

UNINCORPORATED ENGINEERING ORDERS

THE INFORMATION CONTAINED HEREIN IS PROPRIETARY TO AND CONSIDERED A TRADE SECRET OF MICRODATA CORPORATION AND SHALL NOT BE REPRODUCED, IN WHOLE OR PART, WITHOUT THE WRITTEN AUTHORIZATION OF MICRODATA CORPORATION.

[5] COLORED STRIPE INDICATES DRAIN WIRE.

[4] MARK REF DESIGNATION PER Z20016001 APPROXIMATELY WHERE SHOWN.

[3] MARK PER Z20016001 WITH ASSY NO., DASH NO., AND REV LTR APPROXIMATELY WHERE SHOWN.

2. ASSEMBLE PER MICRODATA WORKMANSHIP MANUAL.

1. INTERPRET DRAWING PER MIL-STD-100.

NOTES: UNLESS OTHERWISE SPECIFIED

NEXT ASSY

USED ON

APPLICATION

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES
TOLERANCES .XX ± .02
TOLERANCES .XXX ± .010
TOLERANCES ANGLES ±½°

MATERIAL

FINISH

DRAFTSMAN DATE 2-10-81

CHECKER 2/11/81

ENGINEER

APPD

APPD 2/26/81

APPD 2/26/81

TITLE

CABLE ASSY, 26 CONDUCTOR

SCALE

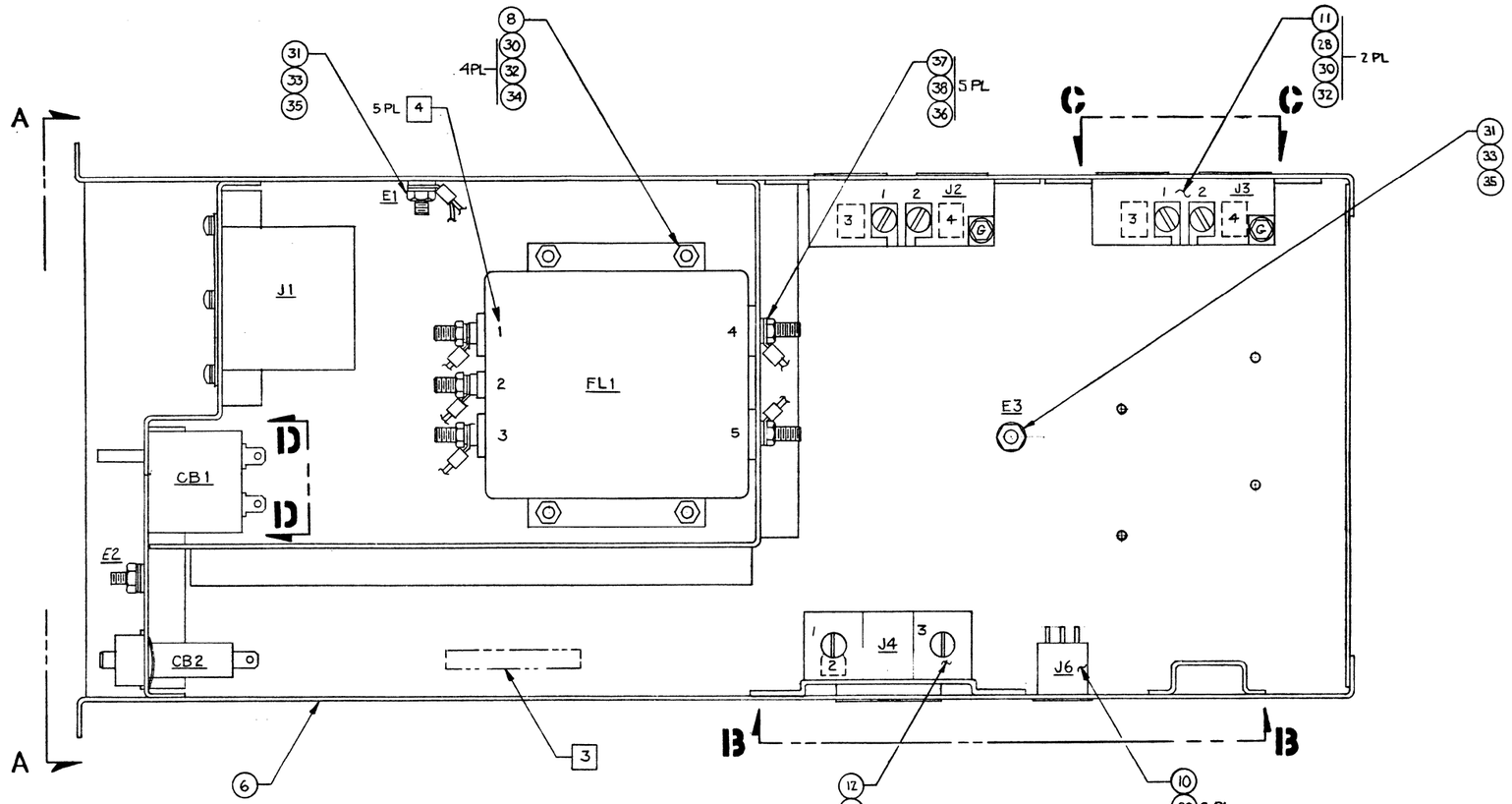
IDENT CODE 52936

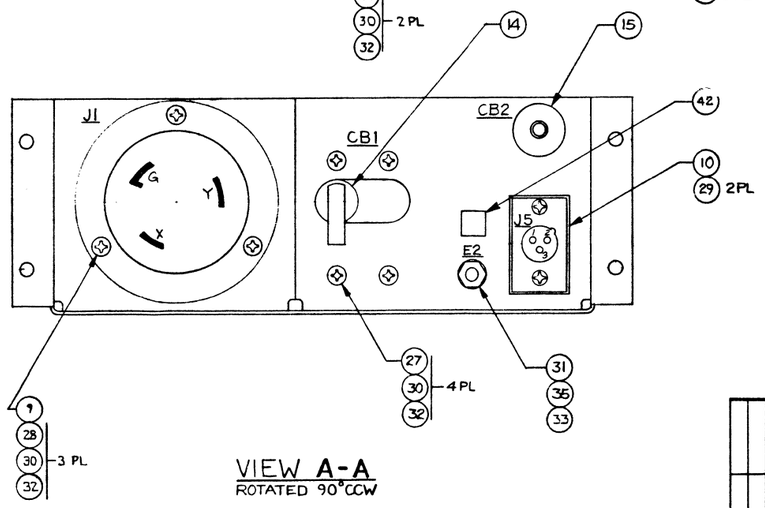
Microdata
IRVINE, CALIFORNIA

C	A20032187	A
DWG SIZE	SHEET 1 OF 1	REV

FORM 8-136-3

REV	EO	ZONE	DESCRIPTION	DWN	CHKD	APPD	DATE
AX	10594		PROTOTYPE RELEASE	[illegible]	[illegible]	[illegible]	6-10-81
AX1	11038		INCORP EO	DEC	[illegible]	[illegible]	12-2-81
A	11084		PILOT RELEASE	[illegible]	[illegible]	[illegible]	1-11-82





VIEW A-A
ROTATED 90° CCW

4 MARK REF DES. AS SHOWN

3. MARK PER Z20016001 WITH PART NUMBER AND REVISION LETTER + S/N

2. ASSEMBLE PER MICRODATA WORKMANSHIP MANUAL

1. INTERPRET DRAWING PER MIL-STD-100

NOTES: UNLESS OTHERWISE SPECIFIED

A20032193 A

190-250V 50/60 HZ	A20032193-001	A
DESCRIPTION	PART NUMBER	REV LTR
ASSEMBLY TABULATION CHART		

6-41

TABULATED

THE INFORMATION CONTAINED HEREIN IS PROPRIETARY TO AND CONSIDERED A TRADE SECRET OF MICRODATA CORPORATION AND SHALL NOT BE REPRODUCED, IN WHOLE OR PART, WITHOUT THE WRITTEN AUTHORIZATION OF MICRODATA CORPORATION.

UNINCORPORATED ENGINEERING ORDERS

USED ON / NEXT ASSY / APPLICATION

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES
TOLERANCES .XX ± .02
TOLERANCES .XXX ± .010
TOLERANCES ANGLES ±½°
MATERIAL
FINISH

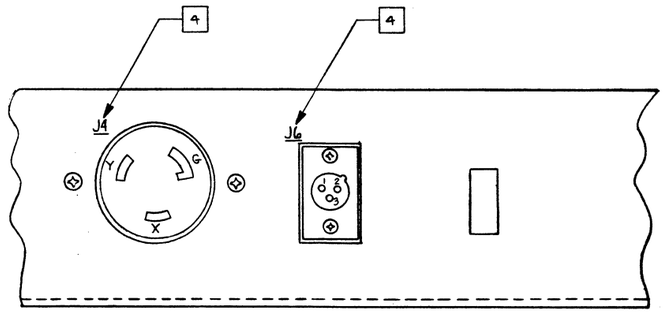
DRAFTSMAN P. Wallace DATE 5/27/81
CHECKER [illegible] 6/4/81
ENGINEER [illegible] 6-9-81
APPD [illegible]
APPD
APPD

TITLE: A.C. POWER MODULE ASSY - SIDE ACCESS

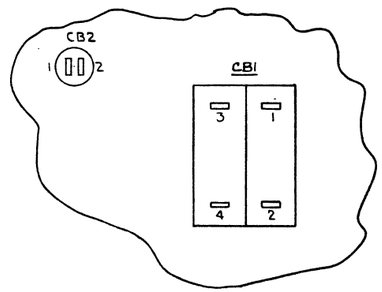
Microdata
IRVINE, CALIFORNIA

SCALE NONE | IDENT CODE 52936 | DWG SIZE D | A20032193 | REV A | SHEET 1 OF 2

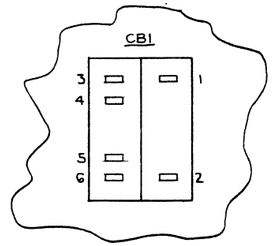
FORM NO 5-135



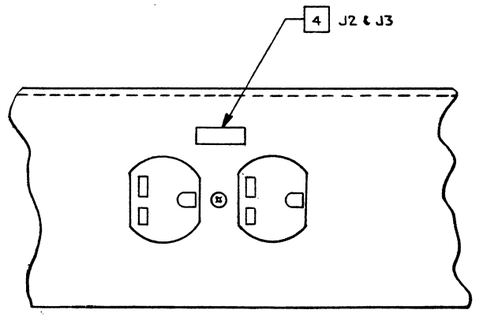
VIEW **B-B**



VIEW **D-D**
ROTATED 90° C.C.W.



ALTERNATE
CIRCUIT BREAKER



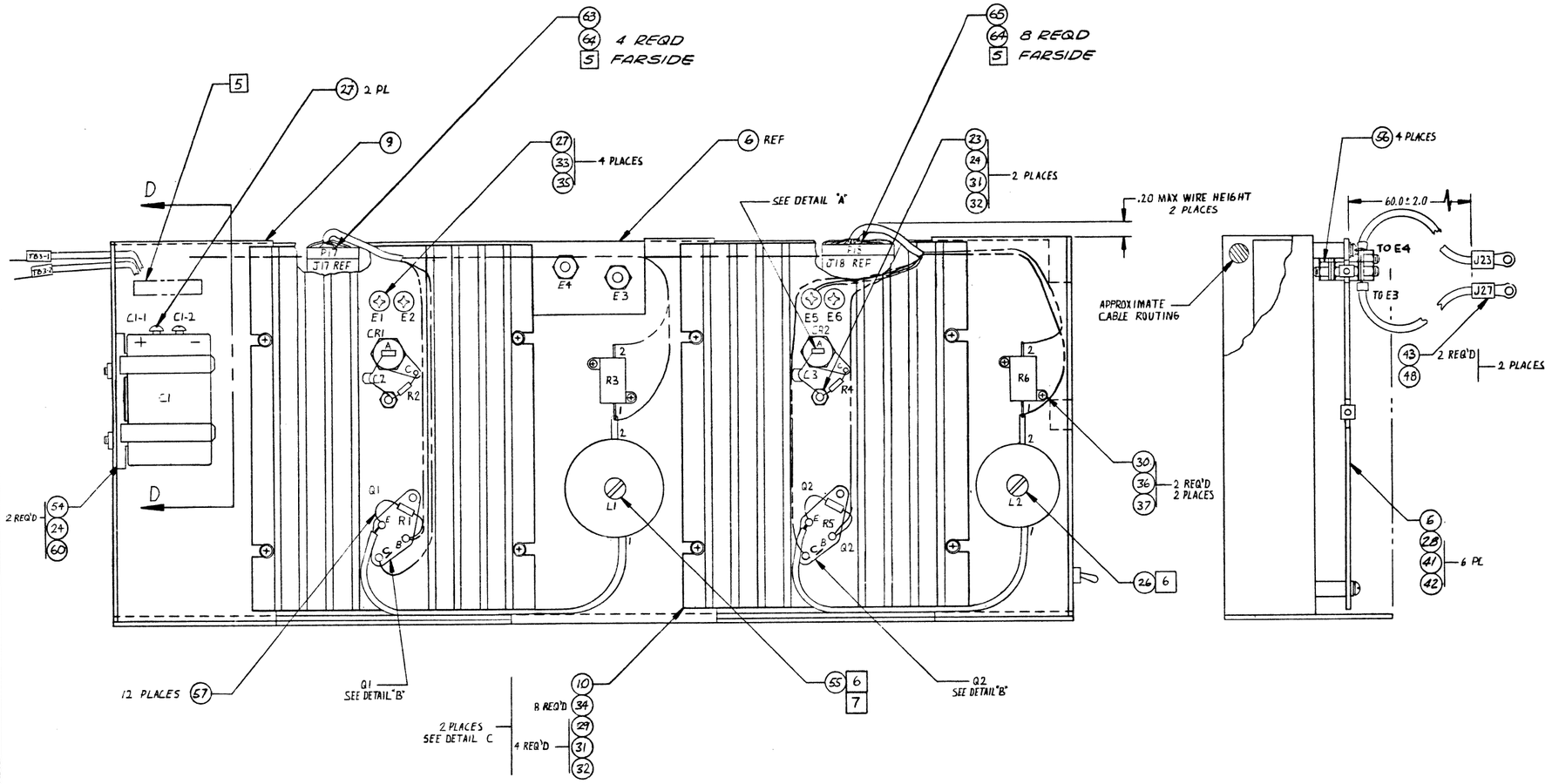
VIEW **C-C**
2 PL

6-43

D	A20032193	A
DWG SIZE	SHEET 2 OF 2	REV

FORM NO. B-136-1

REV	EO	ZONE	DESCRIPTION	DWN	CHKD	APPD	DATE
A	11013		PILOT RELEASE	SFD	MP	[illegible]	11-12-81



7 TRIM LENGTH AS REQUIRED.

6. MAX TORQUE IS 8.75 IN/LBS.

5 MARK PER E20016001 WITH ASSY NO. REV LTR, SERIAL NO. & APPROPRIATE "P" DESIGNATOR.

4. FOR WIRE LIST SEE WL20032206

3. FOR SCHEMATIC SEE SL20032206

2. ASSEMBLE PER MICRODATA WORKMANSHIP MANUAL.

1. INTERPRET DRAWING PER MIL-STD-100.

NOTES: UNLESS OTHERWISE SPECIFIED

A20032206 B
A

A20032206	A
ASSY NO.	REV LTR

ASSY REV CHART

6-45

THE INFORMATION CONTAINED HEREIN IS PROPRIETARY TO AND CONSIDERED A TRADE SECRET OF MICRODATA CORPORATION AND SHALL NOT BE REPRODUCED, IN WHOLE OR PART, WITHOUT THE WRITTEN AUTHORIZATION OF MICRODATA CORPORATION.

UNINCORPORATED ENGINEERING ORDERS

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES
TOLERANCES .XX ± .02
TOLERANCES .XXX ± .010
TOLERANCES ANGLES ±½°

MATERIAL

FINISH

NEXT ASSY

USED ON

APPLICATION

	DATE
DRAFTSMAN	
CHECKER	10/21/81
ENGINEER	10/21/81
APPD	10/21/81
APPD	11/4/81
APPD	11/4/81
APPD	

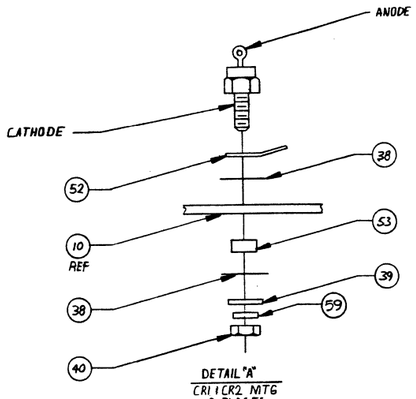
TITLE

MEMORY POWER MODULE, ASSY

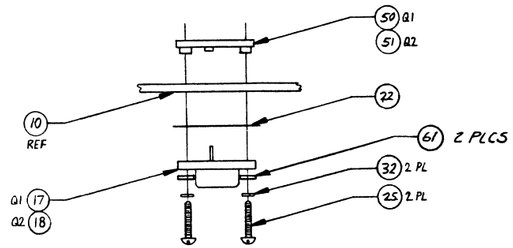
Microdata
IRVINE, CALIFORNIA

SCALE 1:1 | IDENT CODE 52936 | DWG SIZE D | A20032206 | REV A | SHEET 1 OF 2

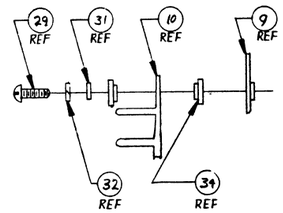
FORM NO. B-135



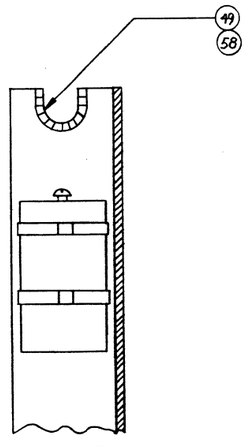
DETAIL "A"
CR1 I CR2 MTG
2 PLACES



DETAIL "B"
Q1 I Q2 MTG
2 PLACES



DETAIL "C"
8 PLACES



VIEW D-D

6-47

D	A20032206	A
DWG SIZE	SHEET 2 OF 2	REV

FORM NO. 9-130-1

REV	ECO	ZONE	DESCRIPTION	DWN	CHKD	APPD	DATE
A1	[illegible]		PROTOTYPE RELEASE	[illegible]			
A2	[illegible]		INCORP ECO	PB			
A3	[illegible]		INCORP ECO	[illegible]			
A	[illegible]		PILOT RELEASE	[illegible]			
B	10267		INCORP ECO	AK	RM	[illegible]	[illegible]
C	10388		INCORP ECO	PBU	[illegible]	[illegible]	[illegible]

POWER SUPPLY A20032025

(P/O A20031025)

(P/O- A20032025)

(P/O-A20032025)

(P/O-A20032025)

E3 E4

E1 +5V MEM

REAR VIEW

J22 +5V CPU

J23 GND

A20032110-001

P1 J1

BLOWER A20032186 SENSOR

P/O A20032186

A20032112-001

CONTROL PANEL A20032038

A20032108-001

CONNECTOR EXTERNAL BATTERY

A20032123-001 (P/O-A20032025)

AC PWR DIST MOD A20032063

A20000347

(P/O-MTU)

INTERLOCK ASSY A20032034

A20032116Z

(P/O MTU)

MTU

SP50820058

TO EXPANSION MTU

MTU CHASSIS GND

PCBA ADAPTER A20032085

A20000774-40

A20032187-001

A2001/3339-001

(P/O-DISC)

REFLEX 1ST DISC

TO 2ND DISC J2 OR TERMINATOR

E1

A20014684-002

A20014832

J21 MAB
J20 MAB
J19 MAB
J18 MAB
J17 MAB
J16 MAB
J15 MAB
J14 MAB
J13 MCB
J12 RN1
J11 ALU
J10 SF

J3 J4 J5

J1 P2 J2 J9 DMP

J5 P1 J1 J2 J3 J8 MTP J7 IOP

J3 P2 J1 P2 J2 J6 DISC J5 DISC J4 ACLC J3 ACLC

J4 J3 J2 ACLC J1 ACLC

FRONT VIEW

CARD CAGE A20032055

I/O PANEL ASSY A20032045

A20032102-001

A20032234 4

A20032104-001

A20032107-001

DMP TERMINAL

A20032102-001

TERMINAL

A20032107-001

A20032102-001

A20032106-001

A20032103-001

PRINTER

TERMINAL

A20032106-001

CD20032100 C

4. USED ONLY WHEN RUNNING ACLC DIAGNOSTICS.
3. FOR EXPANSION CABLE DIAGRAM, FOR TERMINAL & DISK SEE CD20032185
2. (P/O) = PART OF
1. INTERPRET DRAWING PER MIL-STD-100.

NOTES: UNLESS OTHERWISE SPECIFIED

THE INFORMATION CONTAINED HEREIN IS PROPRIETARY TO AND CONSIDERED A TRADE SECRET OF MICRODATA CORPORATION AND SHALL NOT BE REPRODUCED IN WHOLE OR PART WITHOUT THE WRITTEN AUTHORIZATION OF MICRODATA CORPORATION

UNINCORPORATED ENGINEERING ORDERS

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES TOLERANCES XX ± .01 XXX ± .010 TOLERANCES ANGLES ± 1° MATERIAL FINISH

NEXT ASSY APPLICATION

TITLE CABLE DIAGRAM- SEQUOIA (BASIC SYSTEM)

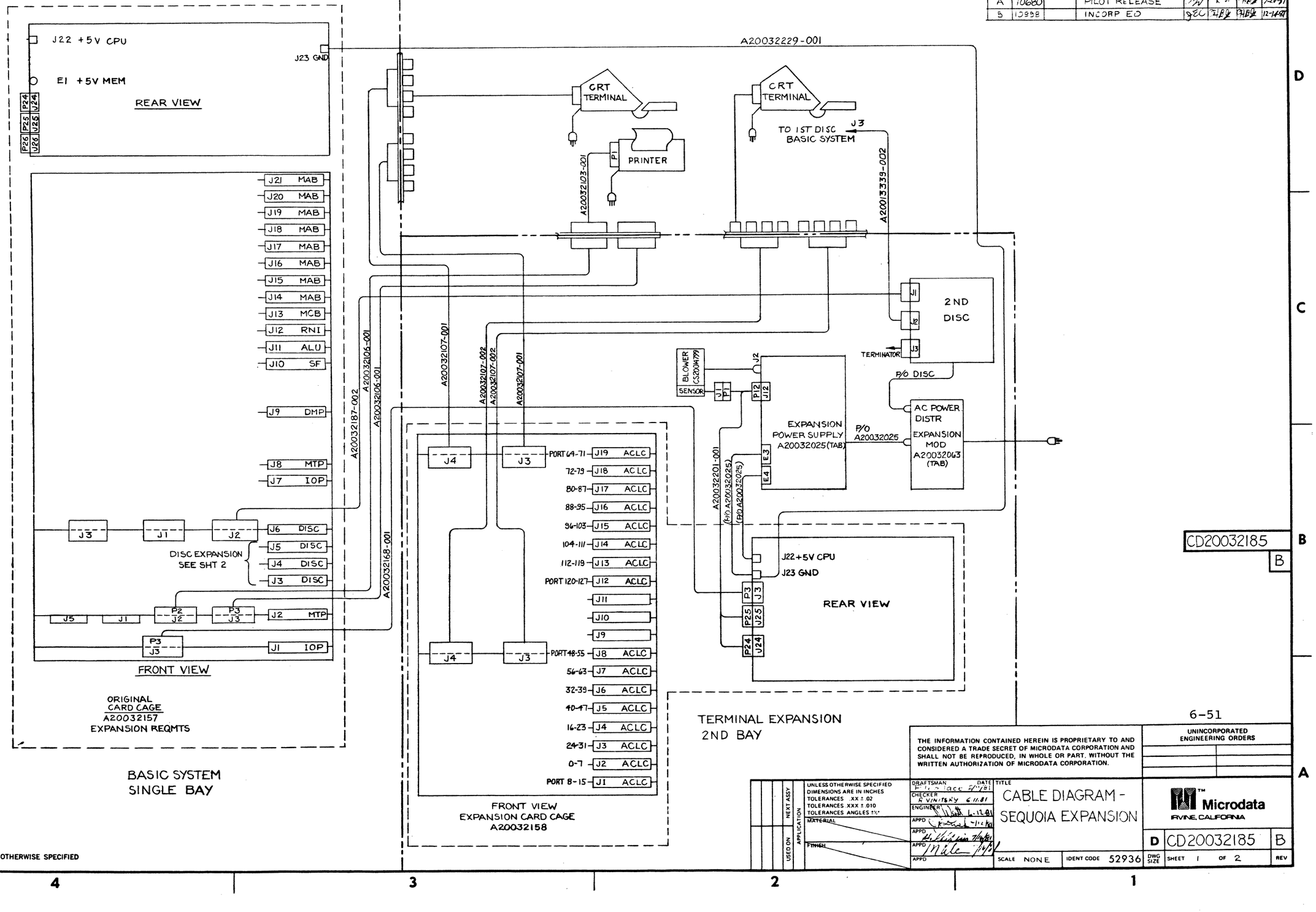
SCALE NONE IDENT CODE 52936

Microdata IRVINE, CALIFORNIA

D CD20032100 C

SHEET 1 OF 1 DWG SIZE REV

REV	EO	ZONE	DESCRIPTION	DWN	CHKD	APPD	DATE
A	10680		PILOT RELEASE				7-20-81
B	10998		INCORP ED				12-14-81



CD20032185 B

B

6-51

NOTES: UNLESS OTHERWISE SPECIFIED

FORM NO 9-135

THE INFORMATION CONTAINED HEREIN IS PROPRIETARY TO AND CONSIDERED A TRADE SECRET OF MICRODATA CORPORATION AND SHALL NOT BE REPRODUCED, IN WHOLE OR PART, WITHOUT THE WRITTEN AUTHORIZATION OF MICRODATA CORPORATION.

UNINCORPORATED ENGINEERING ORDERS

NEXT ASSY | USED ON | APPLICATION

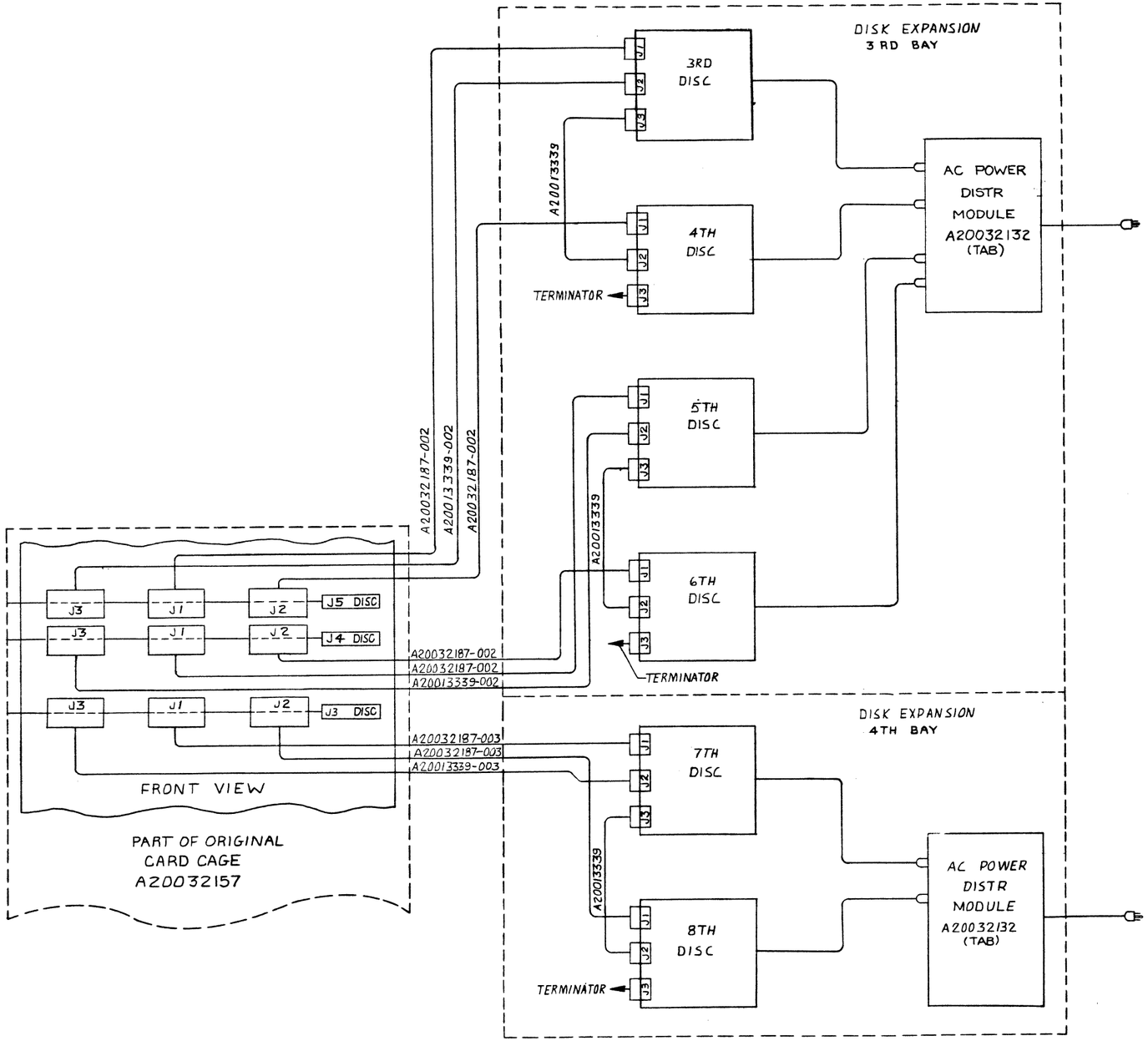
UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES
TOLERANCES .XX ± .02
TOLERANCES .XXX ± .010
TOLERANCES ANGLES ±½°
MATERIAL
FINISH

DRAFTSMAN DATE
CHECKER R. VINITSKY 6.11.81
ENGINEER
APPD
APPD
APPD
APPD

TITLE
CABLE DIAGRAM - SEQUOIA EXPANSION

Microdata
IRVINE, CALIFORNIA

SCALE NONE | IDENT CODE 52936 | DWG SIZE D | CD20032185 B | SHEET 1 OF 2 | REV



D	CD20032185	B
DWG SIZE	SHEET 2 OF 2	REV

FORM NO. 9-125-1

REV	EO	DESCRIPTION	DWN	CHKD	APPD	DATE
AX	10111	PROTO RELEASE	MJS	—	[illegible]	11-20-80
A	10181	PILOT RELEASE	MJS	MP	[illegible]	[illegible]

UNINCORPORATED ENGINEERING ORDERS	

THE INFORMATION CONTAINED HEREIN IS PROPRIETARY TO AND CONSIDERED A TRADE SECRET OF MICRODATA CORPORATION AND SHALL NOT BE REPRODUCED, IN WHOLE OR PART, WITHOUT THE WRITTEN AUTHORIZATION OF MICRODATA CORPORATION.

PREPARED J. MOORE DATE 11-14-80	TITLE COMPONENT SPECIFICATION - LEAD ACID BATTERIES	6-55	
CHECKER [illegible] 11-14-80		Microdata IRVINE, CALIFORNIA	
ENGINEER C. Baird 11/14/80			
APPD J.D. Schneider 11/19/80			
APPD [illegible] 1-13-81			
APPD [illegible] 1-13-81			
RECORDS [illegible] 1/20/80	IDENT CODE 52936	A / CS20014838 / A	DWG SIZE / SHEET 1 OF 5 / REV

Form 3-132-1

1.0 SCOPE

This specification defines an 18 volt, 5 ampere hour, battery pack using 9, 5AH, X cells.

2.0 APPLICABLE DOCUMENTS

Z20016001 - Marking Specification
UL-94V-0

3.0 REQUIREMENTS

3.1 Electrical

3.1.1 Open circuit voltage - (23°C)

Normal at 80% charge = 2.14 volts per cell
19.26 volts per pack

3.1.2 Capacity - Full charged at 23°C

CAPACITY	DISCHARGE CURRENT	DISCHARGE TIME
5.0 AH	500 MA	10 HR

3.2 Mechanical

3.2.1 Case Material: Uniroyal Kralastic Plastic or Equivalent material - UL94V.0 rated
Configuration Size: Figure 1

3.3 Environmental

3.3.1 Temperature Range

Storage	-65°C	to	+65°C
Discharged	-65°C	to	+65°C
Charge	-40°C	to	+65°C

A	CS20014838	A
DWG SIZE	SHEET 2 OF 5	REV

3.3.2 Storage Time Versus Temp (charged cells)

Temp	Time
65°C	60 Days
23°C	1200 Days
0°C	7200 Days

3.3.3 Humidity

Operating	5% to 90% Non Condensing
Storage	5% to 95% Non Condensing

3.3.4 Altitude

Operating	12,000 Feet
Storage and Shipping	40,000 Feet

3.3.5 Vibration

The unit shall be designed to withstand 0.3G's vibration from 93-300 Hz without any degradation.

3.3.6 Shock

The cell pack shall withstand a shock of 10G's on any axis.

4.0 QUALITY ASSURANCE

4.1 General

Suppliers to this procurement specification shall utilize the best practices in manufacturing and inspecting this product. Acceptable processes, controls and procedures are required to assure product integrity from parts/materials and assembly through testing and shipping.

Microdata Product Assurance reserves the right to approve the manufacturer's processes and procedures as they affect product quality in order to fully satisfy the requirements of this paragraph.

4.2 Inspection and Testing

The supplier has the responsibility to perform the necessary inspection and testing to insure compliance with the requirements of this specification. Certification of required inspection and testing shall be provided with the delivered product.

4.3 Source Inspection

Source inspection will be performed at the discretion of Microdata Product Assurance based on the quality level of the delivered product.

5.0 PACKING AND MARKING

5.1 Packaging shall be in accordance with Microdata specification Z20016000.

5.2 The battery packs shall be marked with Microdata part number and applicable revision letter, manufacturers designation (or mark), and date (or lot) code. All marking shall conform to Z20016001, and must with stand the processing specified in Z20016000.

A	CS20014838	A
DWG SIZE	SHEET 4 OF 5	REV

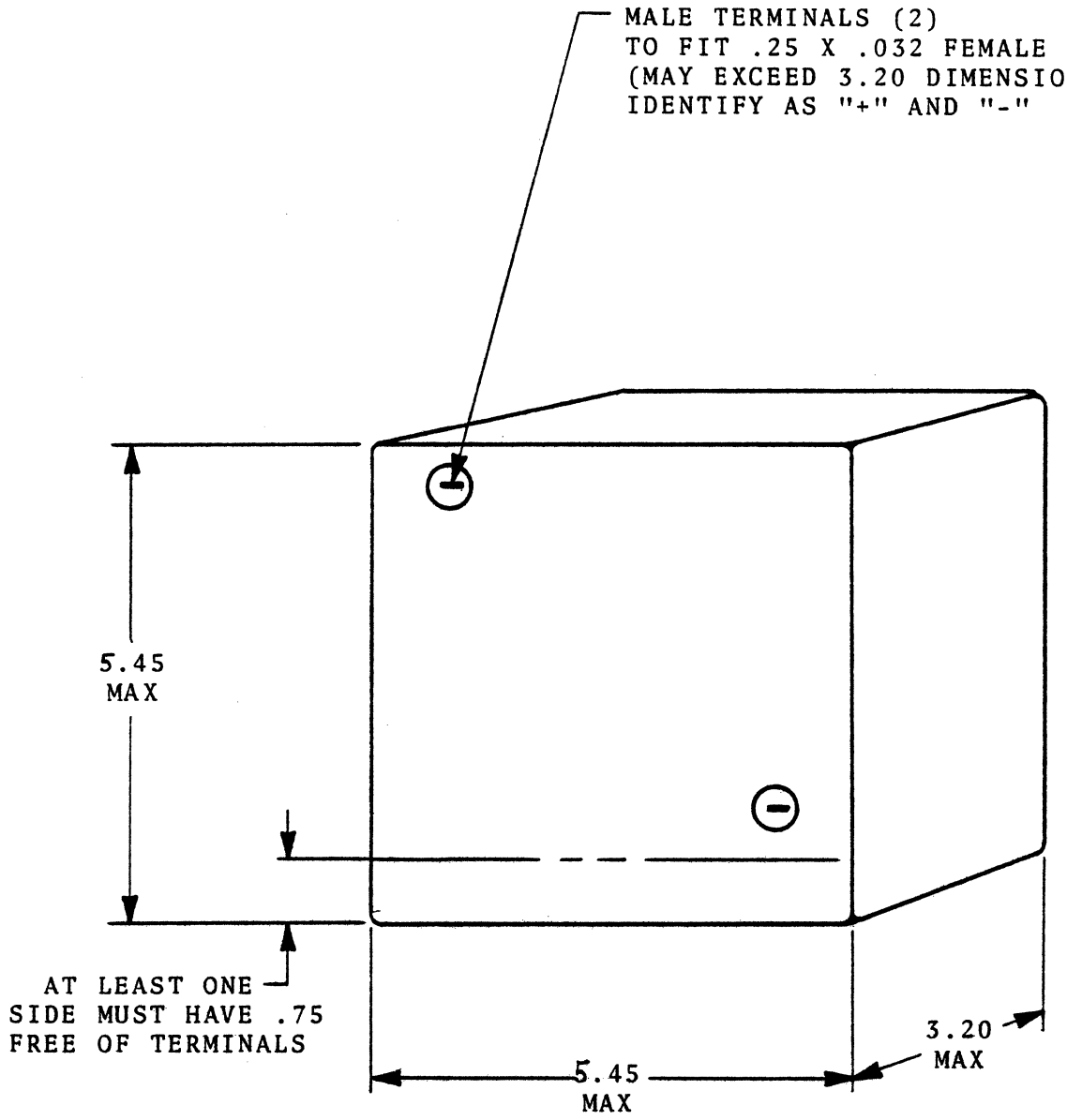


FIGURE 1

SCALE : NONE

A	CS20014838	A
DWG SIZE	SHEET 5 OF 5	REV

FORM 3-132-2

REV	EO	DESCRIPTION	DWN	CHKD	APPD	DATE
AX	9709	PROTO RELEASE	HCW	[illegible]	[illegible]	5-12-80
AX1	9912	INCORP EO	RS.	[illegible]	[illegible]	8-15-80
AX2	10060	REVISED & REDRAWN	RB	[illegible]	RL	[illegible]
AX3	10179	REVISED & REDRAWN	JEC	[illegible]	[illegible]	12-16-80
AX4	10237	REVISED & REDRAWN	CW	[illegible]	[illegible]	1-16-81
A	10206	PILOT RELEASE	GMB	[illegible]	[illegible]	1-30-81
B	10267	INCORP EO	AK	RV	[illegible]	2-20-81
C	10502	INCORP E.O.	NCW	mP	[illegible]	7-16-81
C	10664	INCORP EO	JEC	mP	[illegible]	12-14-81
E	10998	INCORP EO	JEC	[illegible]	[illegible]	12-14-81

UNINCORPORATED ENGINEERING ORDERS	

THE INFORMATION CONTAINED HEREIN IS PROPRIETARY TO AND CONSIDERED A TRADE SECRET OF MICRODATA CORPORATION AND SHALL NOT BE REPRODUCED, IN WHOLE OR PART, WITHOUT THE WRITTEN AUTHORIZATION OF MICRODATA CORPORATION.

PREPARED	DATE
K.C. WILTON	5-5-80
CHECKER [illegible]	5-9-80
ENGINEER [illegible]	5/12/80
APPD [illegible]	5/12/80
APPD [illegible]	1/28/81
APPD [illegible]	1-28-81
REVISED [illegible]	2/12/81

TITLE

INFORMATION DWG.-
SEQUOIA DOCUMENT TREE
(SINGLE DISC CABINET SYSTEM)

IDENT CODE 52936

6-60

Microdata
IRVINE, CALIFORNIA

A	I20032005	E
DWG SIZE	SHEET 1 OF 20	REV

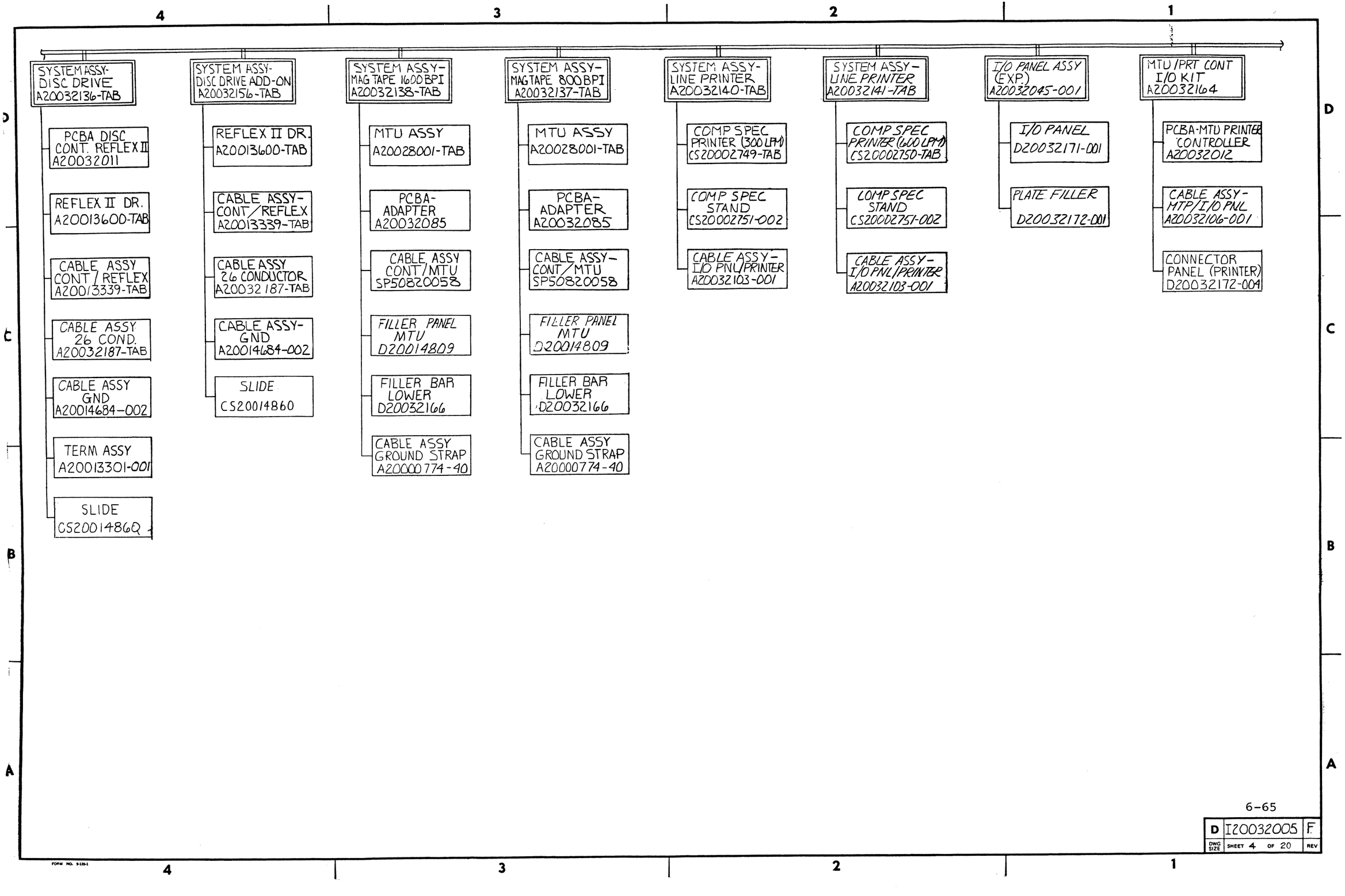
Form 3-132-1

INDEX

- SEQUOIA SYSTEM
  - BASIC A20032100-TAB
    - CABINET ASSY A20032039-TAB
      - CABINET CS20014786-TAB
        - BLOWER ASSY A20032227
          - COMP SPEC BLOWER CS20014862
      - A.C. DIST ASSY A20032063-TAB
        - CHASSIS - PWR. DIST. D20032070
        - BRACKET PWR DIST D20032071
      - VENT FILTER D20032041
      - BRACKET VENT D20014798
      - INTERLOCK SWITCH ASSY (ELECTRIC) A20032034
        - MOUNTING BRACKET D20032200
        - COMP SPEC SOLENOID CS20028343
      - RETAINER-CABLE D20032178
    - CONTROL PNL ASSY A20032038-001
      - PCBA CONTROL PANEL A20032048
      - BEZEL-CONTROL PNL D20032037-001
      - SWITCH PLATE CONTROL PNL D20032040-001
      - GROUND PLATE D20032169
      - BRACKET ROTARY SWITCH D20032087
      - CABLE ASSY INTERLOCK SWITCH A20032162
      - VENT PANEL ASSY A20032089-001
        - VENT PANEL D20032036-001
        - BAFFLE PLATE D20032079-001
        - HINGE-DOOR D20032090-TAB
    - HARNESS ASSY D.C. WIRING A20032110-001
      - CABLE ASSY I/O, 50 COND. A20032107(TAB)
      - CABLE ASSY DMP/ACLC CONN A20032104-001
      - CABLE ASSY CONT. PNL TO DMP A20032108-001
      - CABLE ASSY CONT PNL TO P/S A20032112-001
      - CABLE ASSY A.C. MOD TO P/S (EXT BATT) A20032123-001
      - CABLE ASSY POWER CORD A20000347-TAB
      - DOOR ASSY MTU A20032088-001
        - DOOR FRAME ASSY A20032181-001
          - DOOR FRAME D20032032-001
          - WINDOW D20032091-001
          - BACKING STRIP D20032130-TAB
    - CARD CAGE ASSY A20032157 (035)
      - CHASSIS D20032042 (60)
      - BACKPLANE ASSY A20032065
        - PLATE PWR ADPT W/STUD D20032080
        - PLATE PWR ADPT D20032081
      - CABLE ASSY GROUND A20032179
      - CABLE ASSY INTERCONNECT A20014832
      - FILTER, BLOWER D20032228
      - HINGE-DOOR D20032090-TAB
    - I/O PANEL ASSY A20032045-002
      - CONNECTOR PLATE BLANK D20032172-001
      - PCBA I/O A20032062
      - I/O PANEL D20032171-001
      - CONNECTOR PLATE (TERMINAL) D20032172-002
    - PCBA I/O PROC A20032013
      - PCBA MEMORY CONTROL A20032001
      - PCBA DIAG MAINT PROC A20032015
      - PCBA CPU/ALU A20032007
      - PCBA CPU/RNI A20032008
      - PCBA CPU SPCL FCTN A20032009
      - PANEL, LOUVERED D20032171-003
    - POWER SYSTEM A20032025-TAB
      - BATTERY PLATE D20032151
      - COMP SPEC BATTERY CS20014838
      - SYNC CABLE A20032163
      - COVER D20032082
      - FUSE ASSY A20032101
      - FUSE BRKT D20032116
      - PCBA SERIES REG A20032051
      - HEATSINK SERIES REG D20014824
      - A.C. INPUT ASSY A20032043-TAB
        - PCBA-A.C. INPUT A20032049
        - BRACKET-A.C. INPUT D20032077
      - CHASSIS D20032044
      - CABLE ASSY-D.C. POWER/GND A20032118-001/-002
      - TERMINAL DC PWR D20032148
      - MTG. BRKT TERM. BLOCK D20032086
      - FILTER INDUCTOR CS20032170
      - MEMORY PWR MODULE A20032206
        - HEATSINK MEM PWR MOD D20032115
        - INDUCTOR CS20014830
        - INDUCTOR CS20014831
        - MOUNTING BRKT MEM PWR MOD D20032113
        - PCBA MEMORY MODULE A20032053
          - COMP SPEC TRANSFORMER PULSE CS20014837
      - PCBA LOGIC A20032052
      - COMP SPEC TRAN PWR CS20032180
      - PCBA SWITCH REG A20032050
        - HEATSINK SWITCH REG D20014825
        - BUS BAR SWITCH REG D20032084
        - COMP SPEC TRANS CS20032028
        - COMP SPEC TRANS CS20032031
        - COMP SPEC INDUCTOR CS20032029
        - COMP SPEC CURRENT SENSE XFMR CS20032149
        - RESISTOR BRACKET D20032150
  - PRISM ASSY A20004600-014
  - CABLE ASSY PRISM/IC PANEL A20032102-001
  - PCBA MEMORY ARRAY A20032064-TAB
  - SYS-CONFIG PROM PR20005634-032
  - SYS GEN PROGRAM PM20001284-TAB
  - WORD MATE PROGRAM PM20001905-TAB

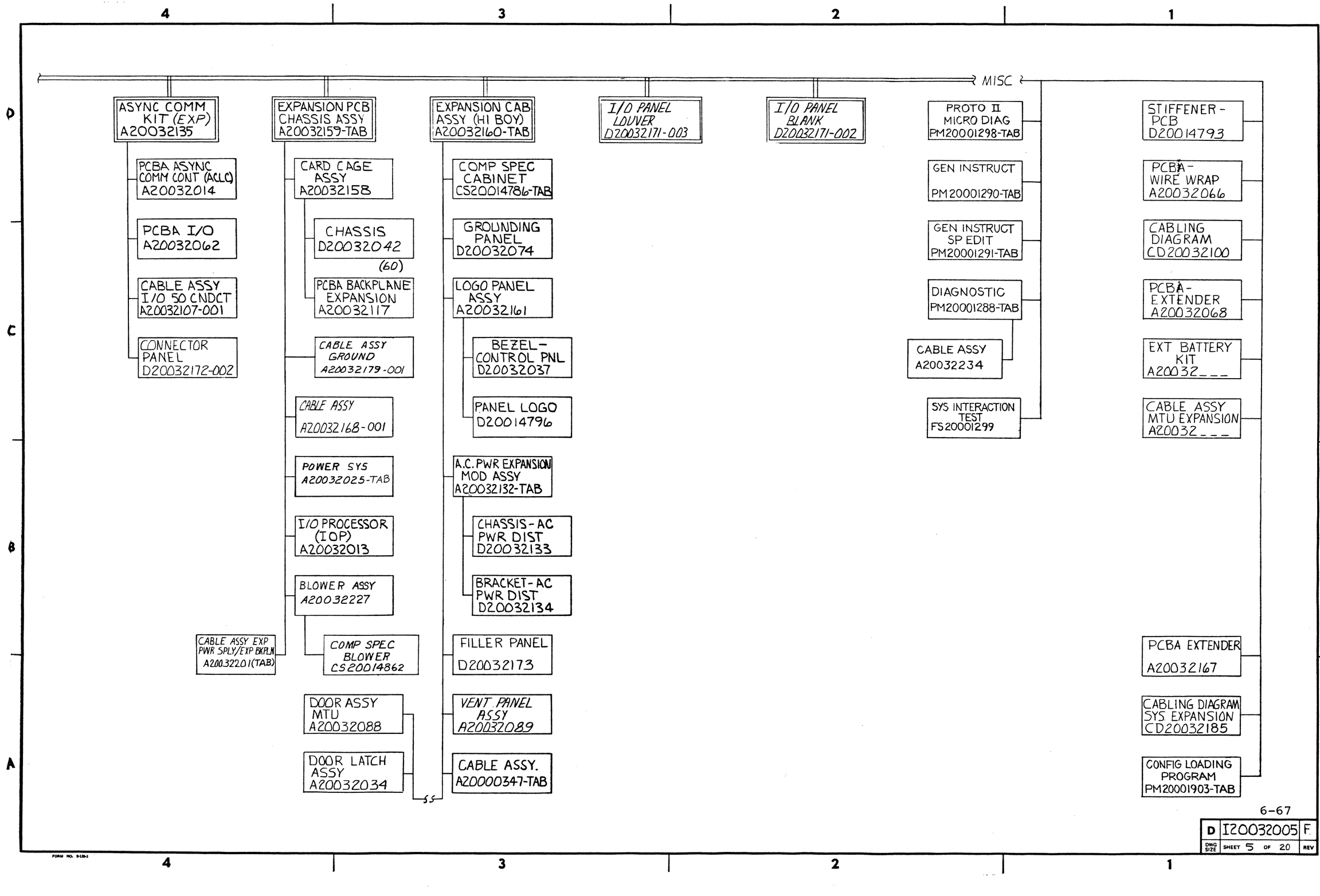
6-63

D	I20032005	F
DWG SIZE	SHEET 3 OF 20	REV.



D	I20032005	F
DWG SIZE	SHEET 4 OF 20	REV

FORM NO. 9-135-1



D	I20032005	F
DWG SIZE	SHEET 5 OF 20	REV

FORM NO. 9-135-1

BASIC SYSTEM TABULATION		
ASSEMBLY NUMBER	LINE VOLTAGE	DESCRIPTION
A20032100-003	220V 50/60 Hz	BROWN/WHITE
A20032100-004	240V 50/60 Hz	BROWN/WHITE
A20032100-005	200V 50/60 Hz	BROWN/WHITE
A20032100-006	208V 50/60 Hz	BROWN/WHITE

A	I20032005	E
DWG SIZE	SHEET 6 OF 20	REV

CABINET ASSEMBLY TABULATION		
PART NUMBER	VOLTAGE	DESCRIPTION
A20032039-002	250V 50/60 Hz	White

A	I20032005	E
DWG SIZE	SHEET 7 OF 20	REV

BASIC POWER SUPPLY TABULATIONS			
PART NUMBER	DESCRIPTION	VOLTAGE	HZ
A20032025-009	POWER SUPPLY ASSY	200V	50/60
A20032025-010	POWER SUPPLY ASSY	208V	50/60
A20032025-011	POWER SUPPLY ASSY	220V	50/60
A20032025-012	POWER SUPPLY ASSY	240V	50/60
A20032025-013	POWER SUPPLY ASSY W/O MEM	200V	50/60
A20032025-014	POWER SUPPLY ASSY W/O MEM	208V	50/60
A20032025-015	POWER SUPPLY ASSY W/O MEM	220V	50/60
A20032025-016	POWER SUPPLY ASSY W/O MEM	240V	50/60

A	I20032005	E
DWG SIZE	SHEET 8 OF 20	REV

MEMORY USAGE CHART			
MEMORY	PART NUMBER DOUBLE BIT DETECTION	PART NUMBER SINGLE BIT DETECTION	QTY
256K	A20032064	A20032064-001	1
512K	A20032064	A20032064-001	2
768K	A20032064	A20032064-001	3
1024K	A20032064	A20032064-001	4
1280K	A20032064	A20032064-001	5
1536K	A20032064	A20032064-001	6
1792K	A20032064	A20032064-001	7
2048K	A20032064	A20032064-001	8

A	I20032005	E
DWG SIZE	SHEET 9 OF 20	REV

PRISM USAGE CHART			
PRISM PART NUMBER	QTY	CABLE PART NUMBER	QTY REQUIRED
A20004600-014	1	A20032102-001	1
A20004600-014	2	A20032120-001	2
A20004600-014	3	A20032102-001	3
A20004600-014	28	A20032102-001	128

A	I20032005	E
DWG SIZE	SHEET 10 OF 20	REV

FORM 3-132-2

SYS-GEN PROGRAM TABULATIONS	
PART NUMBER	DESCRIPTION (MAG TAPE REEL)
PM20001284-001	9 TRACK,NRZ1, 800 BPI
PM20001284-002	9 TRACK,PE, 1600 BPI

SYSTEM ASSEMBLY-DISC DRIVE USAGE AND TABULATIONS

MEG BIT	BASIC DISC SYSTEM		DR. QTY	DR. NO.	ADD-ON DISC SYSTEM		DR. QTY	DR. NO.	VOLTAGE/HZ	
	PART NUMBER	DESCRIPTION			PART NUMBER	DESCRIPTION				
132	A20032136-007	SYS ASSY	1	1	NONE	NONE	NONE	NONE	200V 50 Hz	
	A20032136-008	SYS ASSY	1	1	NONE	NONE	NONE	NONE	200V 60 Hz	
	A20032136-009	SYS ASSY	1	1	NONE	NONE	NONE	NONE	208V 50 Hz	
	A20032136-010	SYS ASSY	1	1	NONE	NONE	NONE	NONE	208V 60 Hz	
	A20032136-005	SYS ASSY	1	1	NONE	NONE	NONE	NONE	220V 50 Hz	
	A20032136-006	SYS ASSY	1	1	NONE	NONE	NONE	NONE	240V 50 Hz	
	A20032136-017	SYS ASSY	1	1	NONE	NONE	NONE	NONE	240V 60 Hz	
264	A20032136-007	SYS ASSY	1	1	A20032156-028	SYS ASSY-ADD-ON	1	2	200V 50 Hz	*
	A20032136-008	SYS ASSY	1	1	A20032156-029	SYS ASSY-ADD-ON	1	2	200V 60 Hz	*
	A20032136-009	SYS ASSY	1	1	A20032156-030	SYS ASSY-ADD-ON	1	2	208V 50 Hz	*
	A20032136-010	SYS ASSY	1	1	A20032156-031	SYS ASSY-ADD-ON	1	2	208V 60 Hz	*
	A20032136-005	SYS ASSY	1	1	A20032156-026	SYS ASSY-ADD-ON	1	2	220V 50 Hz	*
	A20032136-006	SYS ASSY	1	1	A20032156-027	SYS ASSY-ADD-ON	1	2	240V 50 Hz	*
	A20032136-017	SYS ASSY	1	1	A20032156-032	SYS ASSY-ADD-ON	1	2	240V 60 Hz	*
										*
										*
										*
										*
396	A20032136-007	SYS ASSY	1	1	A20032156-028	SYS ASSY-ADD-ON	1	2	200V 50 Hz	*
	A20032136-008	SYS ASSY	1	1	A20032156-029	SYS ASSY-ADD-ON	1	2	200V 60 Hz	*
	A20032136-009	SYS ASSY	1	1	A20032156-030	SYS ASSY-ADD-ON	1	2	208V 50 Hz	*
	A20032136-010	SYS ASSY	1	1	A20032156-031	SYS ASSY-ADD-ON	1	2	208V 60 Hz	*
	A20032136-005	SYS ASSY	1	1	A20032156-026	SYS ASSY-ADD-ON	1	2	220V 50 Hz	*
	A20032136-006	SYS ASSY	1	1	A20032156-027	SYS ASSY-ADD-ON	1	2	240V 50 Hz	*
	A20032136-017	SYS ASSY	1	1	A20032156-032	SYS ASSY-ADD-ON	1	2	240V 60 Hz	*
	A20032136-013	SYS ASSY	1	3	NONE	NONE	NONE	NONE	200V 50 Hz	*
	A20032136-014	SYS ASSY	1	3	↕	↕		↕	200V 60 Hz	*
	A20032136-015	SYS ASSY	1	3					208V 50 Hz	*
	A20032136-016	SYS ASSY	1	3					208V 60 Hz	*
	A20032136-011	SYS ASSY	1	3					220V 50 Hz	*
	A20032136-012	SYS ASSY	1	3					240V 50 Hz	*
	A20032136-018	SYS ASSY	1	3	NONE	NONE	NONE	NONE	240V 60 Hz	*

* EXPANSION CABINET REQUIRED

6-75

A	I20032005	E
DWG SIZE	SHEET 12 OF 20	REV

FORM 3-132-2

SYSTEM ASSEMBLY-DISC DRIVE USAGE AND TABULATIONS

MEG BIT	BASIC DISC SYSTEM PART NUMBER	DESCRIPTION	DR. QTY	DR. NO.	ADD-ON DISC SYSTEM PART NUMBER	DESCRIPTION	DR. QTY	DR. NO.	VOLTAGE/HZ	
528	A20032136-007	SYS ASSY	1	1	A20032156-028	SYS ASSY-ADD-ON	1	2	200V 50 Hz	*
	A20032136-008	SYS ASSY	1	1	A20032156-029	SYS ASSY-ADD-ON	1	2	200V 60 Hz	*
	A20032136-009	SYS ASSY	1	1	A20032156-030	SYS ASSY-ADD-ON	1	2	208V 50 Hz	*
	A20032136-010	SYS ASSY	1	1	A20032156-031	SYS ASSY-ADD-ON	1	2	208V 60 Hz	*
	A20032136-005	SYS ASSY	1	1	A20032156-026	SYS ASSY-ADD-ON	1	2	220V 50 Hz	*
	A20032136-006	SYS ASSY	1	1	A20032156-027	SYS ASSY-ADD-ON	1	2	240V 50 Hz	*
	A20032136-017	SYS ASSY	1	1	A20032156-032	SYS ASSY-ADD-ON	1	2	240V 60 Hz	*
	A20032136-013	SYS ASSY	1	3	A20032156-013	SYS ASSY-ADD-ON	1	4	200V 50 Hz	*
	A20032136-014	SYS ASSY	1	3	A20032156-014	SYS ASSY-ADD-ON	1	4	200V 60 Hz	*
	A20032136-015	SYS ASSY	1	3	A20032156-015	SYS ASSY-ADD-ON	1	4	208V 50 Hz	*
	A20032136-016	SYS ASSY	1	3	A20032156-016	SYS ASSY-ADD-ON	1	4	208V 60 Hz	*
	A20032136-011	SYS ASSY	1	3	A20032156-011	SYS ASSY-ADD-ON	1	4	220V 50 Hz	*
	A20032136-012	SYS ASSY	1	3	A20032156-012	SYS ASSY-ADD-ON	1	4	240V 50 Hz	*
	A20032136-018	SYS ASSY	1	3	A20032156-018	SYS ASSY-ADD-ON	1	4	240V 60 Hz	*
660	A20032136-007	SYS ASSY	1	1	A20032156-028	SYS ASSY-ADD-ON	1	2	200V 50 Hz	*
	A20032136-008	SYS ASSY	1	1	A20032156-029	SYS ASSY-ADD-ON	1	2	200V 60 Hz	*
	A20032136-009	SYS ASSY	1	1	A20032156-030	SYS ASSY-ADD-ON	1	2	208V 50 Hz	*
	A20032136-010	SYS ASSY	1	1	A20032156-031	SYS ASSY-ADD-ON	1	2	208V 60 Hz	*
	A20032136-005	SYS ASSY	1	1	A20032156-026	SYS ASSY-ADD-ON	1	2	220V 50 Hz	*
	A20032136-006	SYS ASSY	1	1	A20032156-027	SYS ASSY-ADD-ON	1	2	240V 50 Hz	*
	A20032136-017	SYS ASSY	1	1	A20032156-032	SYS ASSY-ADD-ON	1	2	240V 60 Hz	*
	A20032136-013	SYS ASSY	2	3,5	A20032156-013	SYS ASSY-ADD-ON	1	4	200V 50 Hz	*
	A20032136-014	SYS ASSY	2	3,5	A20032156-014	SYS ASSY-ADD-ON	1	4	200V 60 Hz	*
	A20032136-015	SYS ASSY	2	3,5	A20032156-015	SYS ASSY-ADD-ON	1	4	208V 50 Hz	*
	A20032136-016	SYS ASSY	2	3,5	A20032156-016	SYS ASSY-ADD-ON	1	4	208V 60 Hz	*
	A20032136-011	SYS ASSY	2	3,5	A20032156-011	SYS ASSY-ADD-ON	1	4	220V 50 Hz	*
	A20032136-012	SYS ASSY	2	3,5	A20032156-012	SYS ASSY-ADD-ON	1	4	240V 50 Hz	*
	A20032136-018	SYS ASSY	2	3,5	A20032156-018	SYS ASSY-ADD-ON	1	4	240V 60 Hz	*
792	A20032136-007	SYS ASSY	1	1	A20032156-028	SYS ASSY-ADD-ON	1	2	200V 50 Hz	*
	A20032136-008	SYS ASSY	1	1	A20032156-029	SYS ASSY-ADD-ON	1	2	200V 60 Hz	*
	A20032136-009	SYS ASSY	1	1	A20032156-030	SYS ASSY-ADD-ON	1	2	208V 50 Hz	*
	A20032136-010	SYS ASSY	1	1	A20032156-031	SYS ASSY-ADD-ON	1	2	208V 60 Hz	*
	A20032136-005	SYS ASSY	1	1	A20032156-026	SYS ASSY-ADD-ON	1	2	220V 50 Hz	*
	A20032136-006	SYS ASSY	1	1	A20032156-027	SYS ASSY-ADD-ON	1	2	240V 50 Hz	*
	A20032136-017	SYS ASSY	1	1	A20032156-032	SYS ASSY-ADD-ON	1	2	2 0V 0 Hz	*
	A20032136-013	SYS ASSY	2	3,5	A20032156-013	SYS ASSY-ADD-ON	2	4,6	200V 50 Hz	*
	A20032136-014	SYS ASSY	2	3,5	A20032156-014	SYS ASSY-ADD-ON	2	4,6	200V 60 Hz	*
	A20032136-015	SYS ASSY	2	3,5	A20032156-015	SYS ASSY-ADD-ON	2	4,6	208V 50 Hz	*
	A20032136-016	SYS ASSY	2	3,5	A20032156-016	SYS ASSY-ADD-ON	2	4,6	208V 60 Hz	*
	A20032136-011	SYS ASSY	2	3,5	A20032156-011	SYS ASSY-ADD-ON	2	4,6	220V 50 Hz	*
	A20032136-012	SYS ASSY	2	3,5	A20032156-012	SYS ASSY-ADD-ON	2	4,6	240V 50 Hz	*
	A20032136-018	SYS ASSY	2	3,5	A20032156-018	SYS ASSY-ADD-ON	2	4,6	240V 60 Hz	*

* EXPANSION CABINET REQUIRED

6-76

DWG SIZE		REV
A	I20032005	E
	SHEET 13 OF 20	

FORM 3-132-8

SYSTEM ASSEMBLY-DISC DRIVE USAGE AND TABULATIONS

MEG BIT	BASIC DISC SYSTEM PART NUMBER	BASIC DISC SYSTEM DESCRIPTION	DR. QTY	DR. NO.	ADD-ON DISC SYSTEM PART NUMBER	ADD-ON DISC SYSTEM DESCRIPTION	DR. QTY	DR. NO.	VOLTAGE/HZ
924	A20032136-007	SYS ASSY	1	1	A20032156-028	SYS ASSY-ADD-ON	1	2	200V 50 Hz
	A20032136-008	SYS ASSY	1	1	A20032156-029	SYS ASSY-ADD-ON	1	2	200V 60 Hz
	A20032136-009	SYS ASSY	1	1	A20032156-030	SYS ASSY-ADD-ON	1	2	208V 50 Hz
	A20032136-010	SYS ASSY	1	1	A20032156-031	SYS ASSY-ADD-ON	1	2	208V 60 Hz
	A20032136-005	SYS ASSY	1	1	A20032156-026	SYS ASSY-ADD-ON	1	2	220V 50 Hz
	A20032136-006	SYS ASSY	1	1	A20032156-027	SYS ASSY-ADD-ON	1	2	240V 50 Hz
	A20032136-017	SYS ASSY	1	1	A20032156-032	SYS ASSY-ADD-ON	1	2	240V 60 Hz
	A20032136-013	SYS ASSY	2	3,5	A20032156-013	SYS ASSY-ADD-ON	2	4,6	200V 50 Hz
	A20032136-014	SYS ASSY	2	3,5	A20032156-014	SYS ASSY-ADD-ON	2	4,6	200V 60 Hz
	A20032136-015	SYS ASSY	2	3,5	A20032156-015	SYS ASSY-ADD-ON	2	4,6	208V 50 Hz
	A20032136-016	SYS ASSY	2	3,5	A20032156-016	SYS ASSY-ADD-ON	2	4,6	208V 60 Hz
	A20032136-011	SYS ASSY	2	3,5	A20032156-011	SYS ASSY-ADD-ON	2	4,6	220V 50 Hz
	A20032136-012	SYS ASSY	2	3,5	A20032156-012	SYS ASSY-ADD-ON	2	4,6	240V 50 Hz
	A20032136-018	SYS ASSY	2	3,5	A20032156-018	SYS ASSY-ADD-ON	2	4,6	240V 60 Hz
	A20032136-022	SYS ASSY	1	7	NONE	NONE	NONE	NONE	200V 50 Hz
	A20032136-023	SYS ASSY	1	7					200V 60 Hz
	A20032136-024	SYS ASSY	1	7					208V 50 Hz
	A20032136-025	SYS ASSY	1	7					208V 60 Hz
	A20032136-020	SYS ASSY	1	7					220V 50 Hz
	A20032136-021	SYS ASSY	1	7					240V 50 Hz
	A20032136-019	SYS ASSY	1	7	NONE	NONE	NONE	NONE	240V 60 Hz
1056	A20032136-007	SYS ASSY	1	1	A20032156-028	SYS ASSY-ADD-ON	1	2	200V 50 Hz
	A20032136-008	SYS ASSY	1	1	A20032156-029	SYS ASSY-ADD-ON	1	2	200V 60 Hz
	A20032136-009	SYS ASSY	1	1	A20032156-030	SYS ASSY-ADD-ON	1	2	208V 50 Hz
	A20032136-010	SYS ASSY	1	1	A20032156-031	SYS ASSY-ADD-ON	1	2	208V 60 Hz
	A20032136-005	SYS ASSY	1	1	A20032156-026	SYS ASSY-ADD-ON	1	2	220V 50 Hz
	A20032136-006	SYS ASSY	1	1	A20032156-027	SYS ASSY-ADD-ON	1	2	240V 50 Hz
	A20032136-017	SYS ASSY	1	1	A20032156-032	SYS ASSY-ADD-ON	1	2	240V 60 Hz
	A20032136-013	SYS ASSY	2	3,5	A20032156-013	SYS ASSY-ADD-ON	2	4,6	200V 50 Hz
	A20032136-014	SYS ASSY	2	3,5	A20032156-014	SYS ASSY-ADD-ON	2	4,6	200V 60 Hz
	A20032136-015	SYS ASSY	2	3,5	A20032156-015	SYS ASSY-ADD-ON	2	4,6	208V 50 Hz
	A20032136-016	SYS ASSY	2	3,5	A20032156-016	SYS ASSY-ADD-ON	2	4,6	208V 60 Hz
	A20032136-011	SYS ASSY	2	3,5	A20032156-011	SYS ASSY-ADD-ON	2	4,6	220V 50 Hz
	A20032136-012	SYS ASSY	2	3,5	A20032156-012	SYS ASSY-ADD-ON	2	4,6	240V 50 Hz
	A20032136-018	SYS ASSY	2	3,5	A20032156-018	SYS ASSY-ADD-ON	2	4,6	240V 60 Hz
	A20032136-022	SYS ASSY	1	7	A20032156-022	SYS ASSY-ADD-ON	1	8	200V 50 Hz
	A20032136-023	SYS ASSY	1	7	A20032156-023	SYS ASSY-ADD-ON	1	8	200V 60 Hz
	A20032136-024	SYS ASSY	1	7	A20032156-024	SYS ASSY-ADD-ON	1	8	208V 50 Hz
	A20032136-025	SYS ASSY	1	7	A20032156-025	SYS ASSY-ADD-ON	1	8	208V 60 Hz
	A20032136-020	SYS ASSY	1	7	A20032156-020	SYS ASSY-ADD-ON	1	8	220V 50 Hz
	A20032136-021	SYS ASSY	1	7	A20032156-021	SYS ASSY-ADD-ON	1	8	240V 50 Hz
	A20032136-019	SYS ASSY	1	7	A20032156-019	SYS ASSY-ADD-ON	1	8	240V 60 Hz

6-77

A	I20032005	E
DWG SIZE	SHEET 14 OF 20	REV

FORM 3-133-2

TABULATIONS		
SYSTEM ASSEMBLY-MAGNETIC TAPE 45 IPS 800 BPI		
PART NUMBER	VOLTAGE/Hz	
A20032137-005	200/208V 50/60Hz	
A20032137-003	220V 50/60 Hz	
A20032137-004	240V 50/60 Hz	

A	I20032005	E
DWG SIZE	SHEET 15 OF 20	REV

SYSTEM ASSEMBLY-LINE PRINTER TABULATIONS		
PART NUMBER	VOLTAGE/Hz	DESCRIPTION
A20032140-001	100V 50/60 Hz	300 LPM
A20032140-002	115V 50/60 Hz	300 LPM
A20032140-003	220V 50 Hz	300 LPM
A20032140-004	240V 50 Hz	300 LPM

SYSTEM ASSEMBLY-LINE PRINTER TABULATIONS		
PART NUMBER	VOLTAGE/Hz	DESCRIPTION
A20032141-001	100V 50/60 Hz	600 LPM
A20032141-002	115V 50/60 Hz	600 LPM
A20032141-003	220V 50 Hz	600 LPM
A20032141-004	240V 50 Hz	600 LPM

I/O PANEL ASSY (EXPANSION)			
PART NUMBER	QTY	ADDITIONAL PRT	ADDITIONAL PORTS
A20032045-001	1	4	32

TO BE USED WHEN MORE PORTS OR PRINTERS ARE REQUIRED THAN ARE AVAILABLE WITH STANDARD TERMINAL PORTS CONFIGURATION, AND WILL REQUIRE AN EXPANSION CABINET.

ASYNC COMM KIT		
PART NUMBER	QTY	PORT QTY'S
A20032135	*	1 THRU 64
A20032135	*	65 THRU 128

* 1 ASYNC COMM KIT FOR EACH GROUP OF 8 PORTS

A	I20032005	E
DWG SIZE	SHEET 17 OF 20	REV

MAG TAPE/PRT CONTROLLER I/O KIT			
PART NUMBER	QTY	MAG TAPE SYSTEM	PRINTER SYSTEM
A20032164	1	1	2

TO BE USED WITH EVERY MAG TAPE SYSTEM AND/OR (2) PRINTER SYSTEM

A	I20032005	E
DWG SIZE	SHEET 18 OF 20	REV

EXPANSION PCB CHASSIS TABULATIONS		
PART NUMBER	VOLTAGE/Hz	P/S PART NUMBER
A20032159-005	200V 50/60 Hz	A20032025-013
A20032159-006	208V 50/60 Hz	A20032025-014
A20032159-003	220V 50/60 Hz	A20032025-015
A20032159-004	240V 50/60 Hz	A20032025-016

EXPANSION CABINET TABULATION		
PART NUMBER	VOLTAGE	DESCRIPTION
A20032160-002	240V 50/60 Hz	WHITE, DISC EXP
A20032160-004	240V 50/60 Hz	WHITE, W/MTL DOOR, DISC EXP
A20032160-006	240V 50/60 Hz	WHITE, ACLC EXP
A20032160-008	240V 50/60 Hz	WHITE, W/MTL DOOR, ACLC EXP

A	I20032005	E
DWG SIZE	SHEET 20 OF 20	REV

REV	EO	ZONE	DESCRIPTION	DWN	CHKD	APPD	DATE
AX	9683		PROTO RELEASE	MJS	HBG	RFS	4-22-80
AX1	9739		INCORP E.C.	MJS	FO	RFS	5-28-80
AX2	10108		REVISED & REDRAWN	MJS	MP	[illegible]	11-20-80
A	10231		PILOT RELEASE	[illegible]	MP	[illegible]	12-31-80
B	12335		INCORP E.O.	DEV	MP	[illegible]	4-28-81
C	10576		INCORP E.O.	SPD	R.V.	[illegible]	6-20-81
D	10633		INCORP E.O.	SPD	R.V.	[illegible]	6-20-81

AC INPUT ASSY
A20032043-002

PCB ASSY AC INPUT
A20032049

P4 J4: +200V, ACSW, DCN, VB, -200V, ACG, ACH, TPON/, AC ON/

TB2 FS2: ACM1, ACM2, CK1, CK2, ACB1, ACB2, AC L, AC H

115VAC L, 115VAC H

T1
CS20032180

240V, 220V, 208V, 200V

F1 2AMP S/B

C1 1UF ± 20% 400V

FS1

CB1 15 AMP

TB1

B1 FAN — P3

J2 BLOWER POWER — WHT, GRN, BLK

P1 — WHT, BLK, GRN

GND STUD

BT1 18V, BT2 18V

P7 J7: VA, VARTN

FUSE ASSY
A20032101

VIN, VIN RTN — TB3

MEMORY POWER MODULE ASSY
A20032055

P5 J5: +200V, -200V

P6 J6: ACON/, CK2, CK1, +VB, TPON/, TXX, +VB RTN

PCB ASSY LOGIC CARD
A20032052

P10 J10: CK1, CK2, ACON/, TPON/, TXX, DCON/, +5VB, +VB RTN, +12V SENSE, +5V SENSE, +5V SENSE, BLOWER 2, BLOWER 1, -12V SENSE, PRDY, PFAT/, CPSS/, RTCK/

PCB ASSY SERIES REG
A20032051

P11 J11: +5VB, +VB, +VB RTN, GND, -12V BIAS, COMM SENSE, +12V SENSE, -12V UNREG, +12V, +5V, -5V SENSE, -5V UNREG, -5V, +12V SENSE, +12V UNREG, +12V

J12: RTCK/, CPSS/, PFAT/, PRDY, BLOWER 1, BLOWER 2, +5V SENSE, COMM SENSE, -5V SENSE, -12V SENSE, +12V SENSE, +12V, -12V, -5V

P18 J18: +12VM, H2VM RTN, +1 SEN 12, -1 SEN 12, BDR 12, VIN

P17 J17: +1 SEN 5, -1 SEN 5, BDR 5, VIN

P16 J16: OV TRIG, BAT 1, BAT 2, EXT BAT

CABLE ASSY SYNC
A20032163

SYNC RTN — J8 P8 — P9 J9

PCB ASSY MEM PWR MOD
A20032053

J20: EXT BAT, EXT BAT RTN

E3 +5VM, E4 +5VM RTN

J21: +12VM, +12VM RTN, -5VM, H2VM SENSE, +5VM SENSE, COMM SENSE, MPFL/, MPSS/, PFIR/

MEMORY POWER SECTION

PCB ASSY SWITCHING REG
A20032050

GND E1 — CABLE ASSY PWR/GND A20032118-001

+5V E2 — CABLE ASSY PWR/GND A20032118-002

SC20032025 D

6-85

2 FOR 200V INPUT CONNECT TB1-10 TO TB1-4, 208V=TB1-10 TO TB1-5, 220V=TB1-10 TO TB1-6, 240V=TB1-10 TO TB1-7

1. INTERPRET DRAWING PER MIL-STD-100.

NOTES: UNLESS OTHERWISE SPECIFIED

FORM NO 9-135

THE INFORMATION CONTAINED HEREIN IS PROPRIETARY TO AND CONSIDERED A TRADE SECRET OF MICRODATA CORPORATION AND SHALL NOT BE REPRODUCED, IN WHOLE OR PART, WITHOUT THE WRITTEN AUTHORIZATION OF MICRODATA CORPORATION.

UNINCORPORATED ENGINEERING ORDERS

NEXT ASSY / USED ON / APPLICATION

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES
TOLERANCES XX ± .02
TOLERANCES XXX ± .010
TOLERANCES ANGLES ±½°
MATERIAL
FINISH

		DATE
DRAFTSMAN	M. STEM	4-17-80
CHECKER	H. CLASSICK	4-22-80
ENGINEER	N. SCOTT	4-22-80
APPD	J. TULLAI	4-22-80
APPD	[illegible]	1-21-81
APPD	[illegible]	1-22-81
APPD	[illegible]	1/22/81

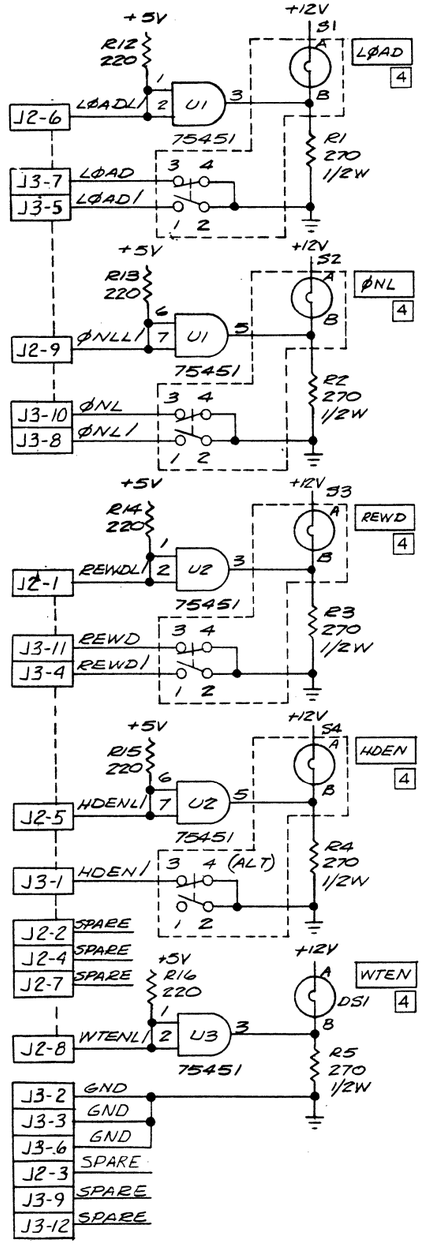
TITLE: SCHEMATIC POWER SUPPLY

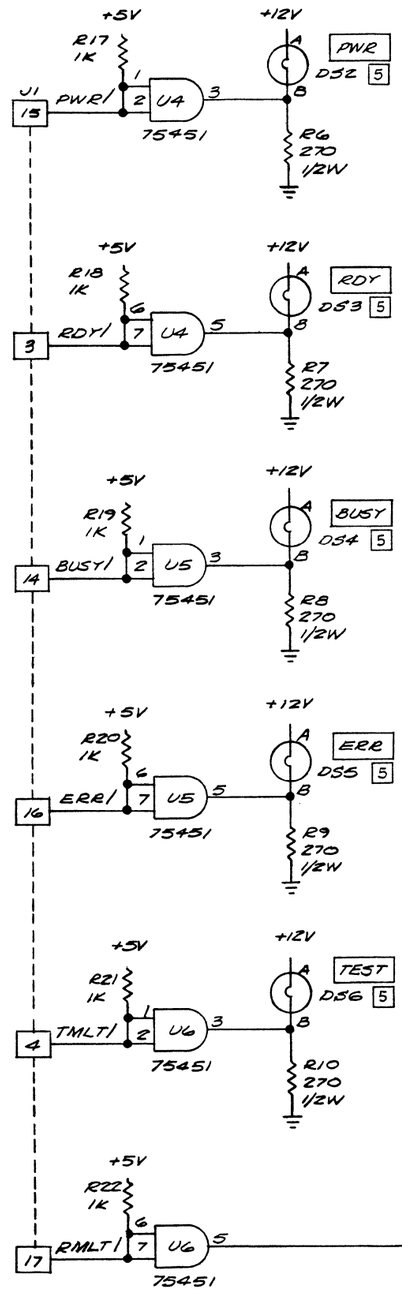
Microdata IRVINE, CALIFORNIA

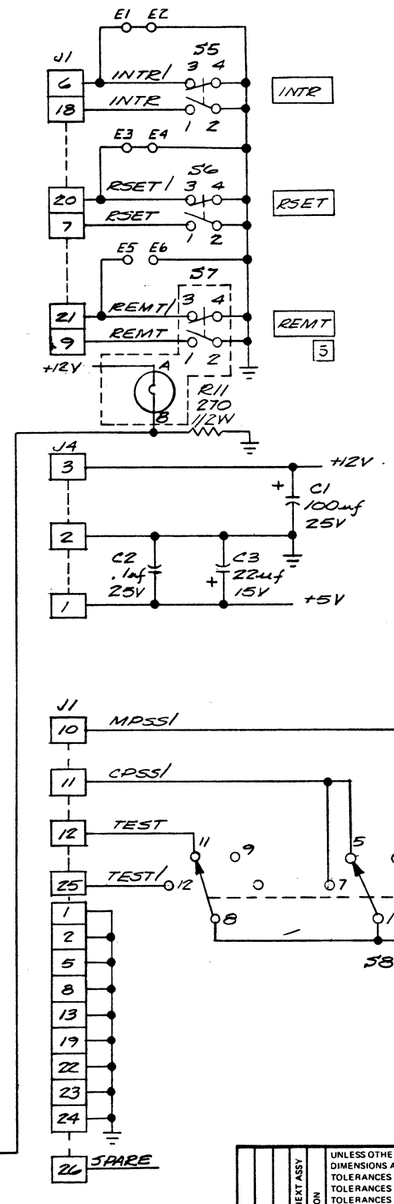
D SC20032025 D

SCALE NONE | IDENT CODE 52936 | DWG SIZE | SHEET 1 OF 1 | REV

REV	EO	ZONE	DESCRIPTION	DWN	CHKD	APPD	DATE
AK	9698		PROTO RELEASE	PW	MP	[illegible]	5/14
AK1	9785		INCORP E.O.	[illegible]	MP	[illegible]	6/17
A	9982		PILOT RELEASE	[illegible]	MP	[illegible]	1-4
B	11199		INCORPORATE EO	PW	MP	[illegible]	4-A







4. USED ON A20032048-001 & -003

3. ALL SWITCHES ARE MOMENTARY EXCEPT S4.

2. INTERPRET PER MIL-STD-100.

1. ALL RESISTORS ARE 1/4W, 5%, VALUE IN OHMS.

NOTES: UNLESS OTHERWISE SPECIFIED

5. USED ON A20032048-001 & -002

5C20032048

B

6-87

THE INFORMATION CONTAINED HEREIN IS PROPRIETARY TO AND CONSIDERED A TRADE SECRET OF MICRODATA CORPORATION AND SHALL NOT BE REPRODUCED, IN WHOLE OR PART, WITHOUT THE WRITTEN AUTHORIZATION OF MICRODATA CORPORATION.

UNINCORPORATED ENGINEERING ORDERS

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES
TOLERANCES XX ± 02
TOLERANCES XXX ± 010
TOLERANCES ANGLES ±½°

MATERIAL

FINISH

NEXT ASSY

USED ON

APPLICATION

DRAFTS'MAN [illegible] DATE 4-1-80
CHECKER [illegible]
ENGINEER [illegible]
APPD [illegible]
APPD [illegible] 1-9-81
APPD [illegible] 1-9-81
APPD

TITLE
PCB SCHEMATIC-
CONTROL PANEL

SCALE —

IDENT CODE 52936

Microdata
IRVINE, CALIFORNIA

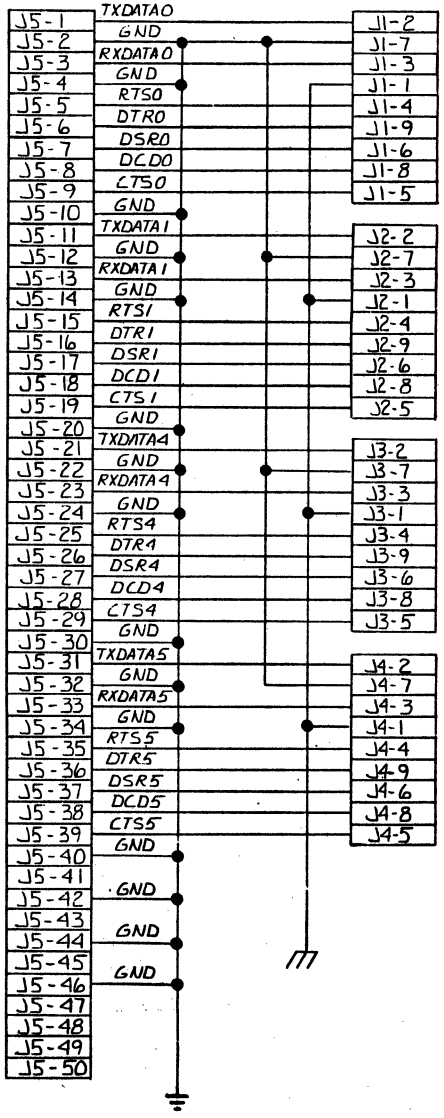
D 5C20032048

DWG SIZE

SHEET 1 OF 1

FORM NO. 9-138

REV	EO	ZONE	DESCRIPTION	DWN	CHKD	APPD	DATE
AX	9611		PROTO RELEASE	CD	JO	[illegible]	3-12-80
A	10001		PILOT RELEASE	CD	MP	[illegible]	10-3-80



SC20032062
A

6-89

1. INTERPRET DRAWING PER MIL-STD-100.

NOTES: UNLESS OTHERWISE SPECIFIED

THE INFORMATION CONTAINED HEREIN IS PROPRIETARY TO AND CONSIDERED A TRADE SECRET OF MICRODATA CORPORATION AND SHALL NOT BE REPRODUCED, IN WHOLE OR PART, WITHOUT THE WRITTEN AUTHORIZATION OF MICRODATA CORPORATION.

UNINCORPORATED ENGINEERING ORDERS

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES
TOLERANCES .XX ± .02
TOLERANCES .XXX ± .010
TOLERANCES ANGLES ±½°
MATERIAL
FINISH

NEXT ASSY | USED ON | APPLICATION

DRAFTSMAN C. Ruble 1-28-80
CHECKER [illegible] 3-12-80
ENGINEER [illegible] 3-12-80
APPD [illegible]
APPD [illegible] 10/2/80
APPD [illegible]
APPD

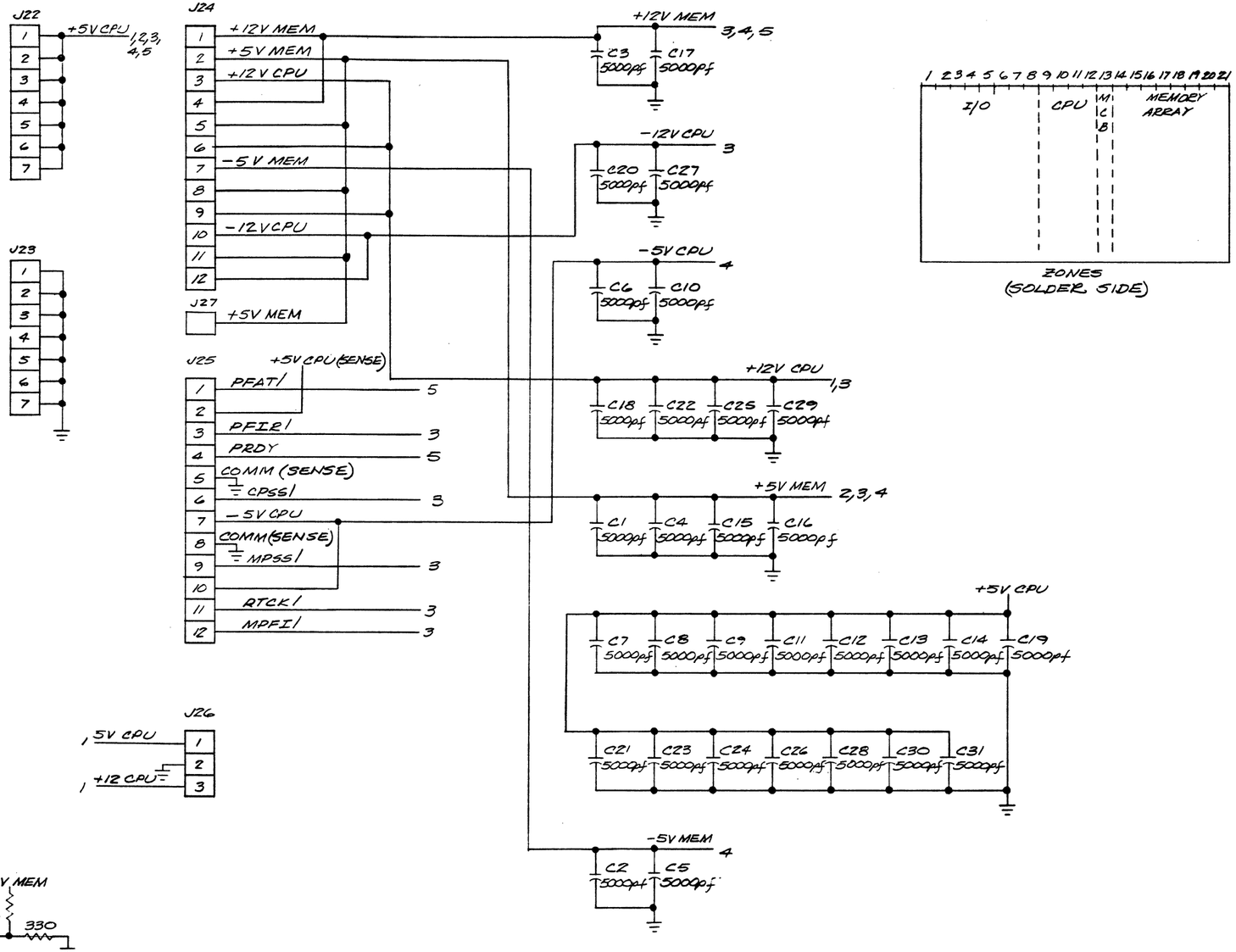
TITLE: PCB SCHEMATIC - I/O CONNECTOR PANEL

Microdata
IRVINE, CALIFORNIA

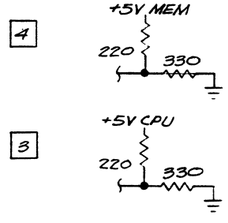
SCALE NONE | IDENT CODE 52936 | DWG SIZE C | SC20032062 | A | SHEET 1 OF 1 | REV

FORM 9-130-3

REV	EO	ZONE	DESCRIPTION	DWN	CHKD	APPD	DATE
AX	9722		PROTOTYPE RELEASE	RSH	RD		5/4/80
A	9984		PILOT RELEASE	RSH	MP		9/2/80



ZONES
(SOLDER SIDE)



2 ALL RESISTOR MODULES PIN 1 GOES TO GND, PIN 8 TO +5V.

1 INTERPRET DRAWING PER MIL-STD-100

NOTES: UNLESS OTHERWISE SPECIFIED

5C20032065 B
A

6-91

THE INFORMATION CONTAINED HEREIN IS PROPRIETARY TO AND CONSIDERED A TRADE SECRET OF MICRODATA CORPORATION AND SHALL NOT BE REPRODUCED, IN WHOLE OR PART, WITHOUT THE WRITTEN AUTHORIZATION OF MICRODATA CORPORATION.

UNINCORPORATED ENGINEERING ORDERS
10211 A
11080
11211

NEXT ASSY
USED ON
APPLICATION

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES
TOLERANCES .XX ± .02
TOLERANCES .XXX ± .010
TOLERANCES ANGLES ±½°
MATERIAL
FINISH

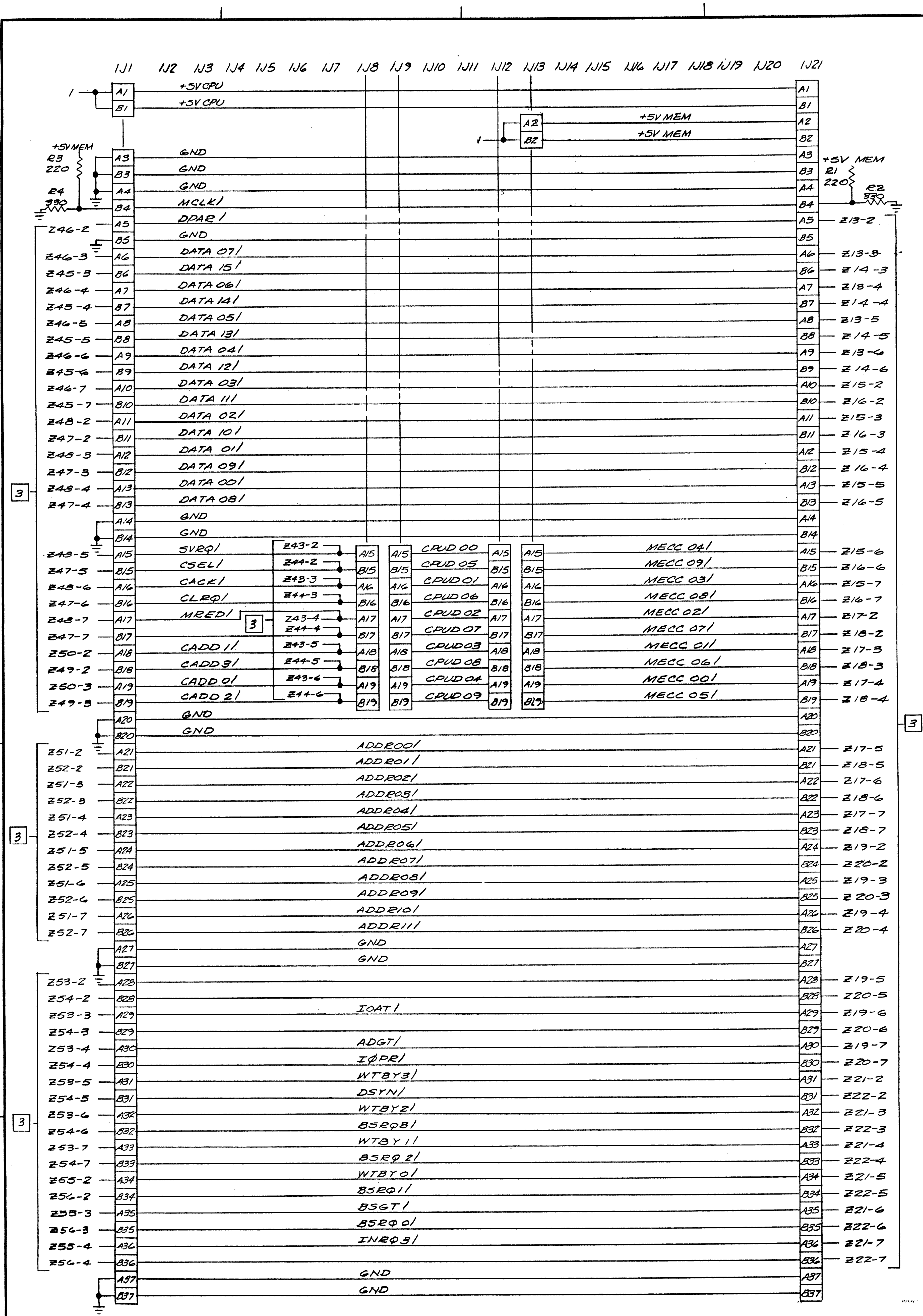
DRAFTSMAN — DATE
CHECKER 5/14/80
ENGINEER 5/14/80
APPD 5/14/80
APPD 9/4/80
APPD

TITLE
PCB SCHEMATIC - BACKPLANE

Microdata
IRVINE, CALIFORNIA

SCALE — IDENT CODE 52936
D 5C20032065 A
DWG SIZE SHEET 1 OF 5 REV

FORM NO 9-135



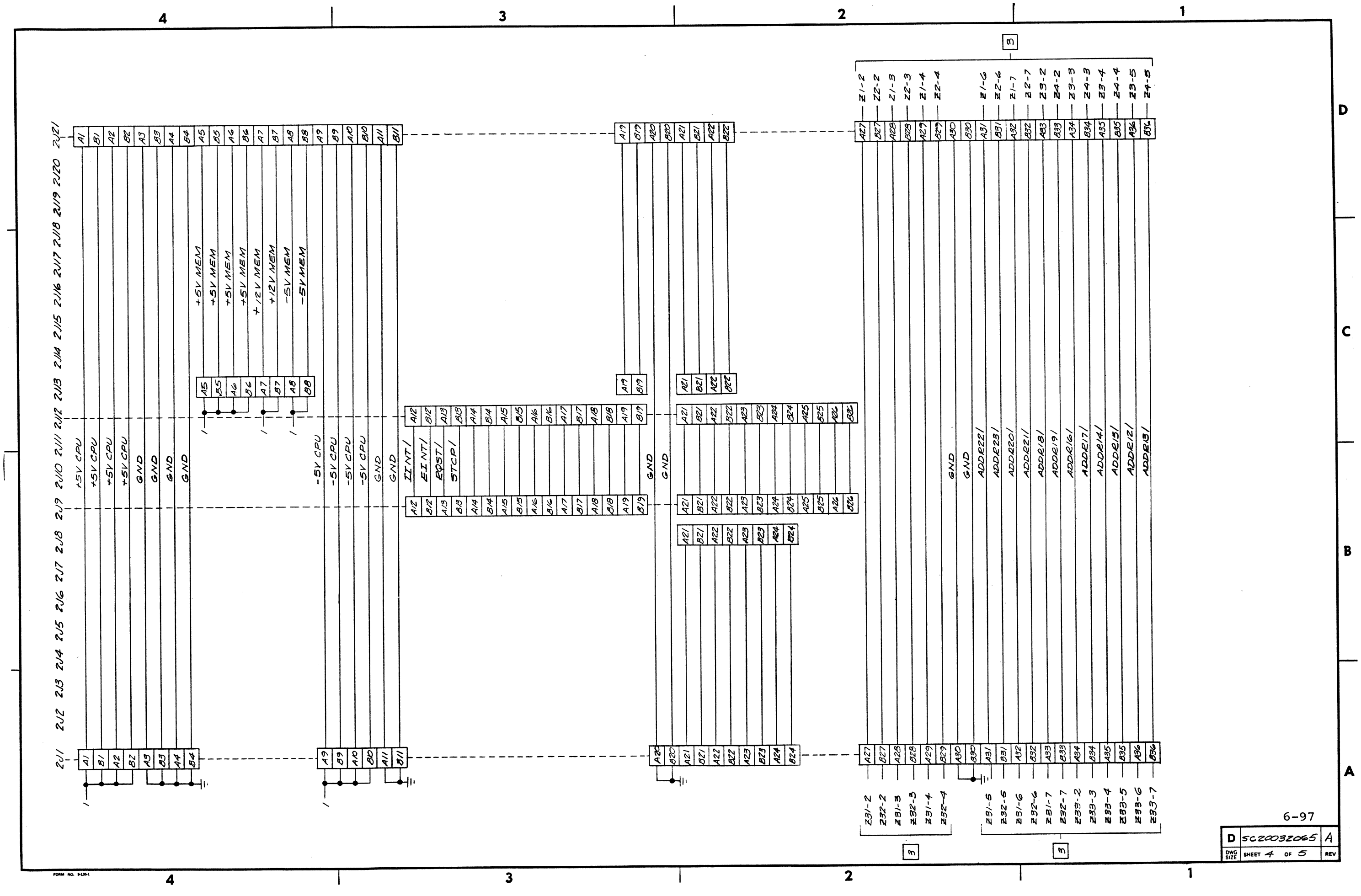
DWG SIZE D | 5C20032065 | SHEET 2 OF 5 | REV A

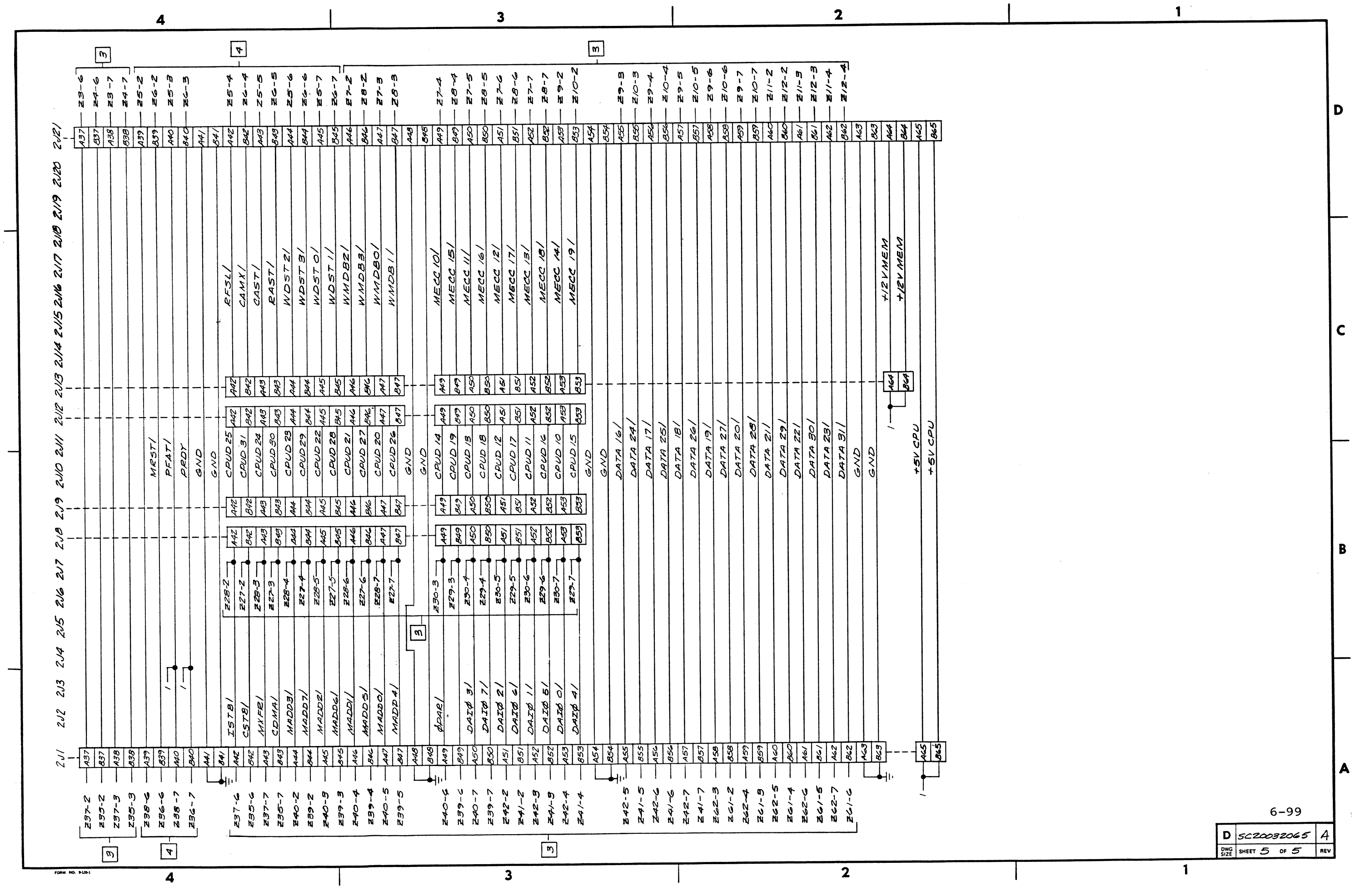
6-93

IJ1 IJ2 IJ3 IJ4 IJ5 IJ6 IJ7 IJ8 IJ9 IJ10 IJ11 IJ12 IJ13 IJ14 IJ15 IJ16 IJ17 IJ18 IJ19 IJ20 IJ21

Z55-5	A38	INBP 2/																				Z23-2
Z52-5	B38	BSAD 3/																				Z24-2
Z55-6	A39	INBP 1/																				Z23-3
Z54-4	B39	BSAD 2/																				Z24-3
Z55-7	A40	INBP 0/																				Z23-4
Z54-7	B40	ISYN3/																				Z24-4
Z57-2	A41	INAD 1/																				Z23-5
Z58-2	B41	ISYN 2/																				Z24-5
Z57-3	A42	INAD 0/																				Z23-6
Z58-3	B42	ISYN 1/																				Z24-6
Z57-4	A43	BSAD 1/																				Z23-7
Z58-4	B43	INAD 3/																				Z24-7
	A44	GND																				
	B44	GND																				
Z57-5	A45	BSAD 0/																				Z25-2
Z58-5	B45	INAD 2/																				Z26-2
Z58-6	A46	PFIR/																				
Z57-7	B46																					Z26-3
Z58-7	A47	BSYN/																				Z25-4
Z55-7	B47																					Z26-4
	A48	CPSS/																				
	B48	MPSS/																				
Z59-5	A49	MERR/																				Z25-6
Z59-6	B49	MPCI/																				
Z60-7	A50	RTCK/																				
	B50	GND																				
EFGT/	A51	GND																				
EFSP/	B51	GND																				
	A52	GND																				
	B52	GND																				
	A53	GND																				
	B53	GND																				
+12V CPU	A54	+12V CPU																				
+12V CPU	B54	+12V CPU																				
+12V CPU	A55	+12V CPU																				
+12V CPU	B55	+12V CPU																				
+12V CPU	A56	+12V CPU																				
+12V CPU	B56	+12V CPU																				
-12V CPU	A57	-12V CPU																				
-12V CPU	B57	-12V CPU																				
-12V CPU	A58	-12V CPU																				
-12V CPU	B58	-12V CPU																				
+12V MEM	A59	+12V MEM																				
+12V MEM	B59	+12V MEM																				
+5V MEM	A60	+5V MEM																				
+5V MEM	B60	+5V MEM																				
+5V MEM	A61	+5V MEM																				
+5V MEM	B61	+5V MEM																				
+5V CPU	A62	+5V CPU																				
+5V CPU	B62	+5V CPU																				
+5V CPU	A63	+5V CPU																				
+5V CPU	B63	+5V CPU																				
GND	A64	GND																				
GND	B64	GND																				
GND	A65	GND																				
GND	B65	GND																				

DWG SIZE		REV
D	SCZ00326G5	A
	SHEET 5 OF 5	


4
3
2
1
D
C
B
A
2J1 2J2 2J3 2J4 2J5 2J6 2J7 2J8 2J9 2J10 2J11 2J12 2J13 2J14 2J15 2J16 2J17 2J18 2J19 2J20 2J21
+5V CPU
-5V CPU
GND
+5V MEM
-5V MEM
+12V MEM
INT/
EINT/
EOST/
STOP/
ADDE22/
ADDE23/
ADDE20/
ADDE21/
ADDE18/
ADDE19/
ADDE16/
ADDE17/
ADDE14/
ADDE15/
ADDE12/
ADDE13/
6-97
D
SC20032065
A
DWG SIZE
SHEET 4 OF 5
REV
FORM NO. 9-139-1

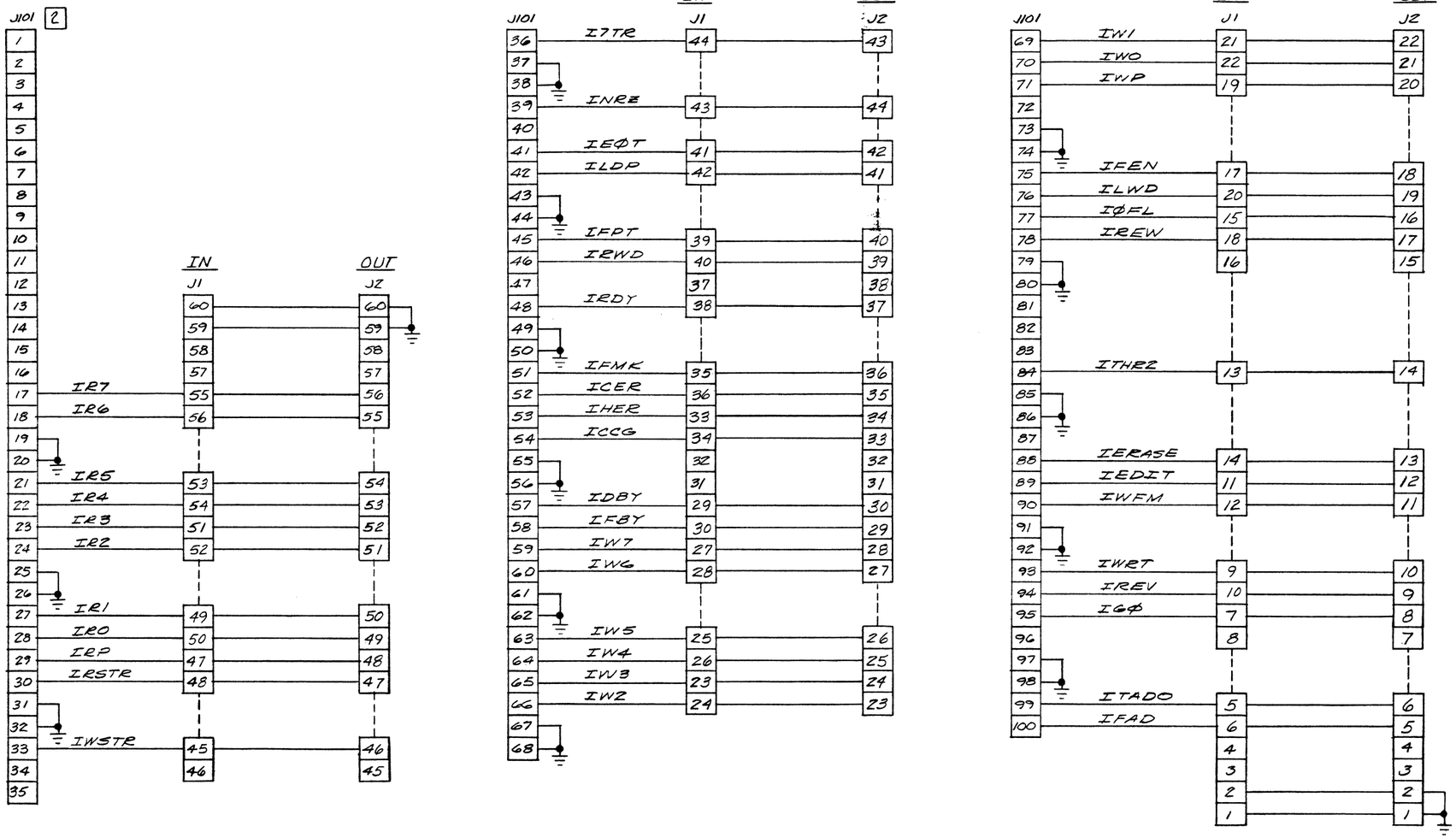


6-99

D	SC20032065	4	
DWG SIZE	SHEET 5 OF 5	REV	

FORM NO. 9-135-1

REV	EO	ZONE	DESCRIPTION	DWN	CHKD	APPD	DATE
AX	9754		PROTO RELEASE	[illegible]	[illegible]	[illegible]	6/6/80
A	9961		PILOT RELEASE	[illegible]	[illegible]	[illegible]	11/1/80
B	10320		INCORP EO	SPD	[illegible]	[illegible]	3-17-82



2 CONNECTOR PIN NUMBERS AGREE WITH NUMBERS ON P.C. BOARD AND NOT MOLDED PIN NUMBERS ON SOME CONNECTORS

1) INTERPRET DRAWING PER MIL-STD-100.

NOTES: UNLESS OTHERWISE SPECIFIED

6-101

THE INFORMATION CONTAINED HEREIN IS PROPRIETARY TO AND CONSIDERED A TRADE SECRET OF MICRODATA CORPORATION AND SHALL NOT BE REPRODUCED, IN WHOLE OR PART, WITHOUT THE WRITTEN AUTHORIZATION OF MICRODATA CORPORATION.

UNINCORPORATED ENGINEERING ORDERS

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES
TOLERANCES .XX ± .02
TOLERANCES .XXX ± .010
TOLERANCES ANGLES ±½°
MATERIAL
FINISH

NEXT ASSY
USED ON
APPLICATION

DRAFTSMAN [illegible] DATE 9/24/79
CHECKER [illegible] 5/23/80
ENGINEER [illegible] 5/23/80
APPD [illegible] 4/30/80
APPD [illegible] 4/10/80
APPD [illegible] 11/10/80

TITLE: PCB SCHEMATIC - ADAPTER MTU CONTROLLER

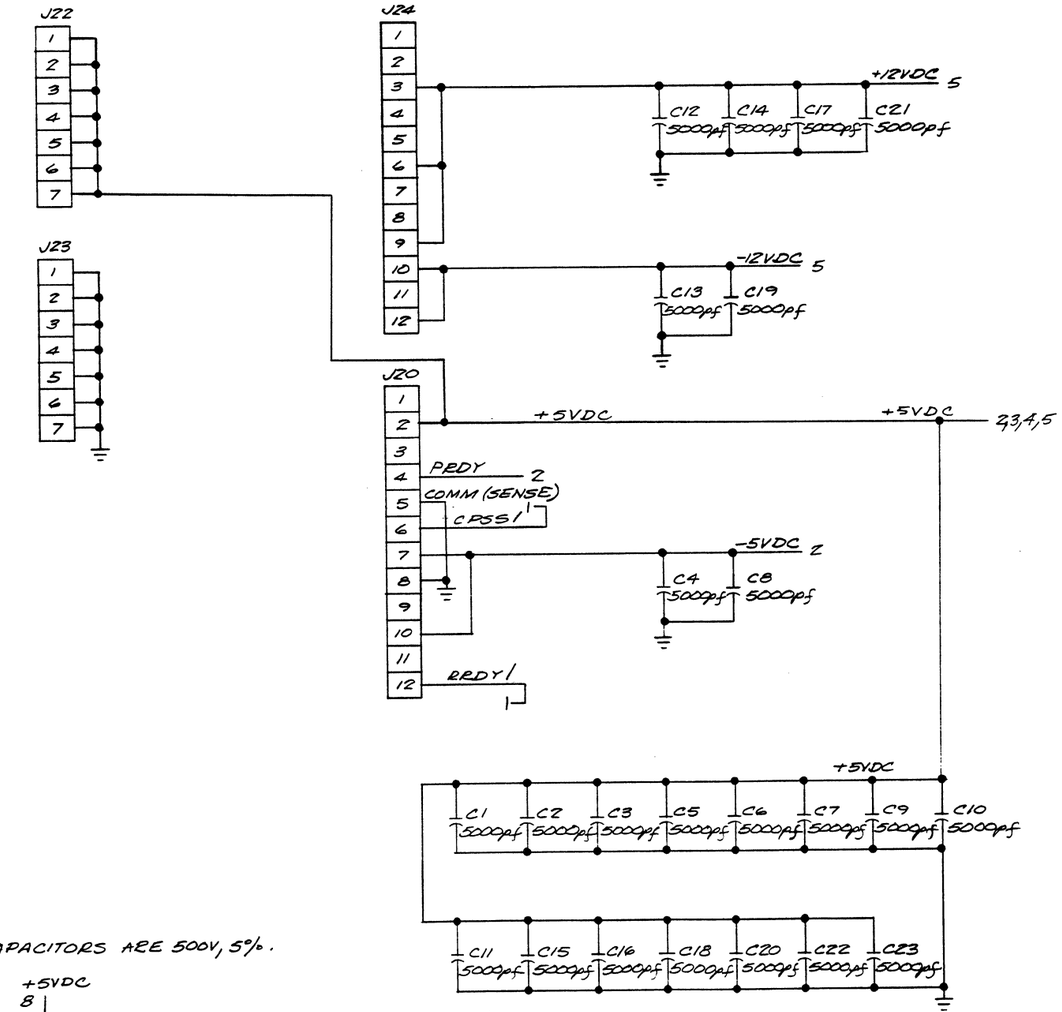
Microdata
IRVINE, CALIFORNIA

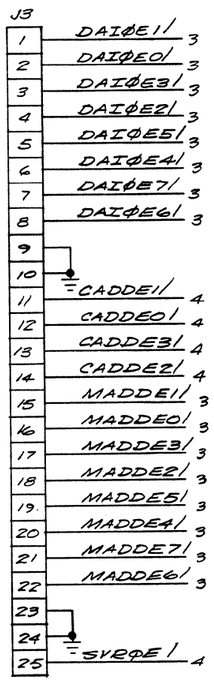
SCALE —
IDENT CODE 52936
DWG SIZE D
SHEET 1 OF 1
SCZ0032085 B
REV

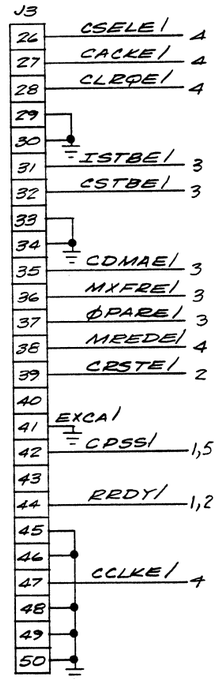
SCZ0032085 B
B

FORM NO. 9-135

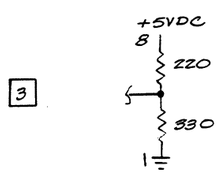
REV	EO	ZONE	DESCRIPTION	DWN	CHKD	APPD	DATE
AX	10091		PROTOTYPE RELEASE	RSH	mP	[illegible]	11-12-80
A	10220		PILOT RELEASE	RSH	R.V.	[illegible]	1-29-81
B	10447		INCORP E.O.	HJW	R.V.	[illegible]	7-1-81







4 ALL CAPACITORS ARE 500V, 5%.



2 ALL RESISTOR MODULES PIN 1 GOES TO GND, PIN 8 TO +5V.

1 INTERPRET DRAWING PER MIL-STD-10C.

NOTES: UNLESS OTHERWISE SPECIFIED

SCZ0032117 B

B

6-103

A

THE INFORMATION CONTAINED HEREIN IS PROPRIETARY TO AND CONSIDERED A TRADE SECRET OF MICRODATA CORPORATION AND SHALL NOT BE REPRODUCED, IN WHOLE OR PART, WITHOUT THE WRITTEN AUTHORIZATION OF MICRODATA CORPORATION.

UNINCORPORATED ENGINEERING ORDERS: 10652

NEXT ASSY | USED ON | APPLICATION

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES
TOLERANCES .XX ± .02
TOLERANCES .XXX ± .010
TOLERANCES ANGLES ±½°

MATERIAL

FINISH

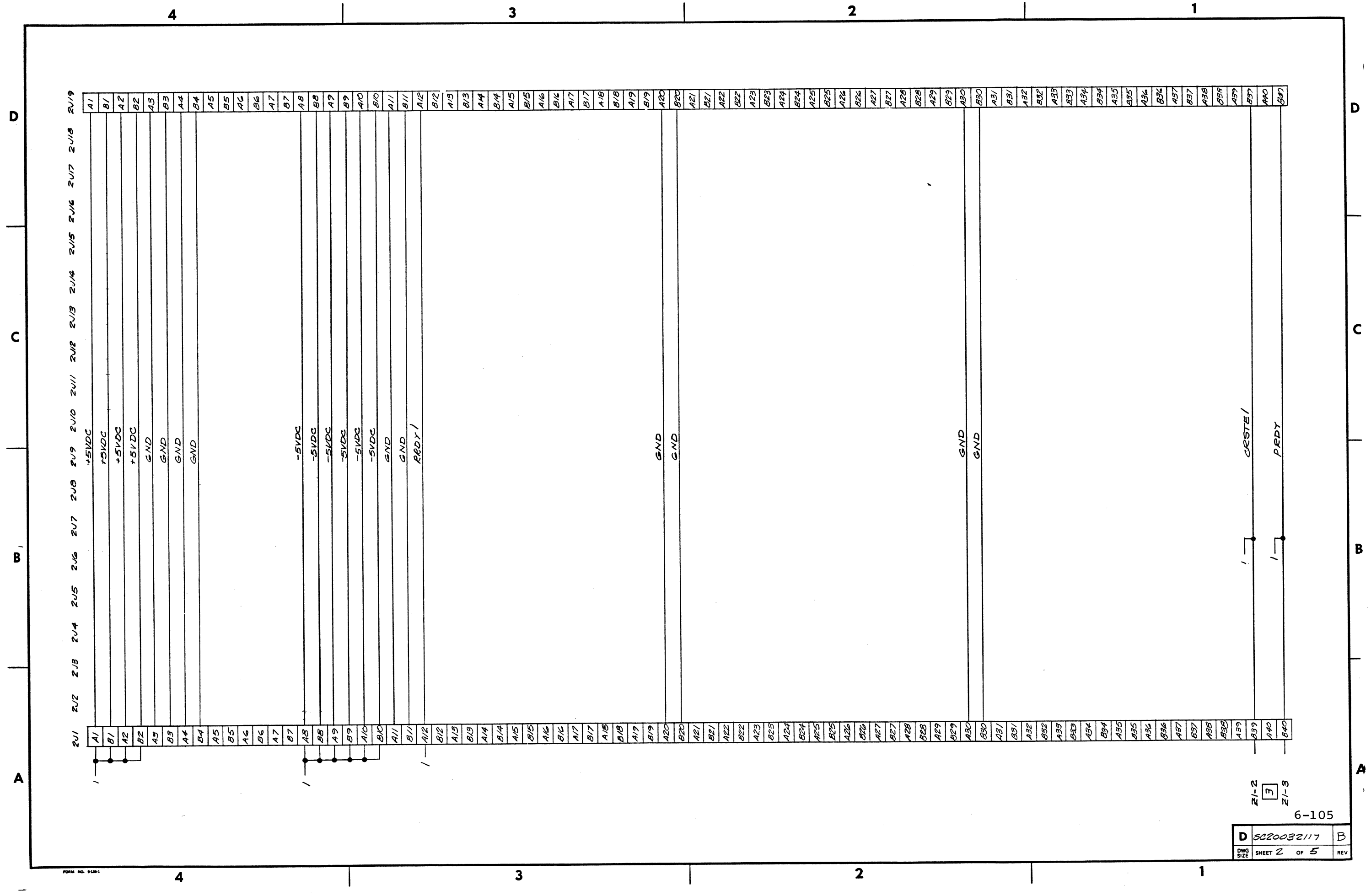
DRAFTSMAN [illegible] DATE 9/10/80
CHECKER [illegible] 11/10/80
ENGINEER [illegible] 11/11/80
APPD [illegible] 11/11/80
APPD [illegible] 11/20/80
APPD [illegible] 1-26-81

TITLE: PCB SCHEMATIC - EXPANSION BACKPLANE

Microdata IRVINE, CALIFORNIA

SCALE — | IDENT CODE 52936 | DWG SIZE D | SCZ0032117 | REV B | SHEET 1 OF 5

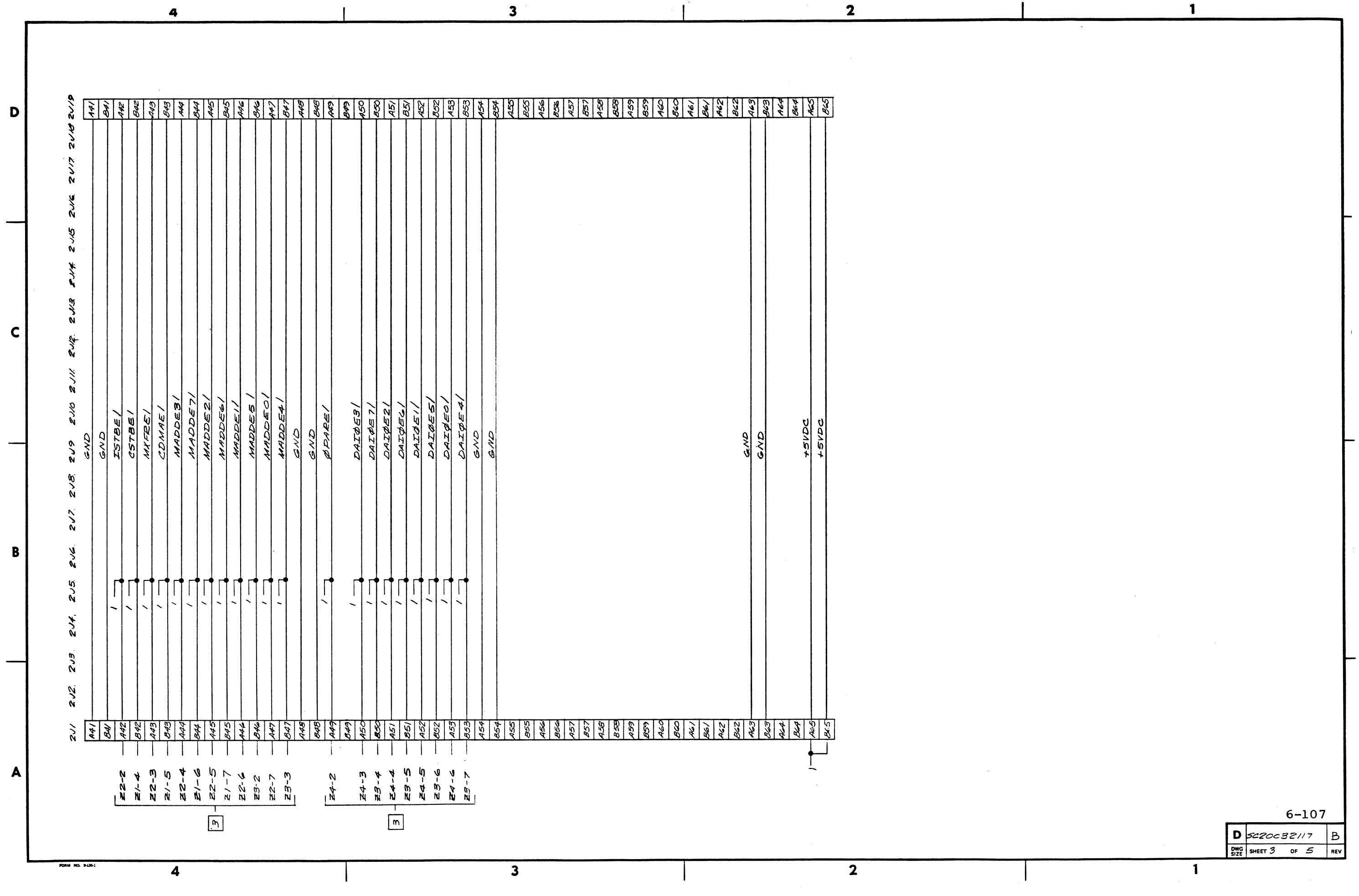
FORM NO 9-135



6-105

D	5C20032117	B
DWG SIZE	SHEET 2 OF 5	REV

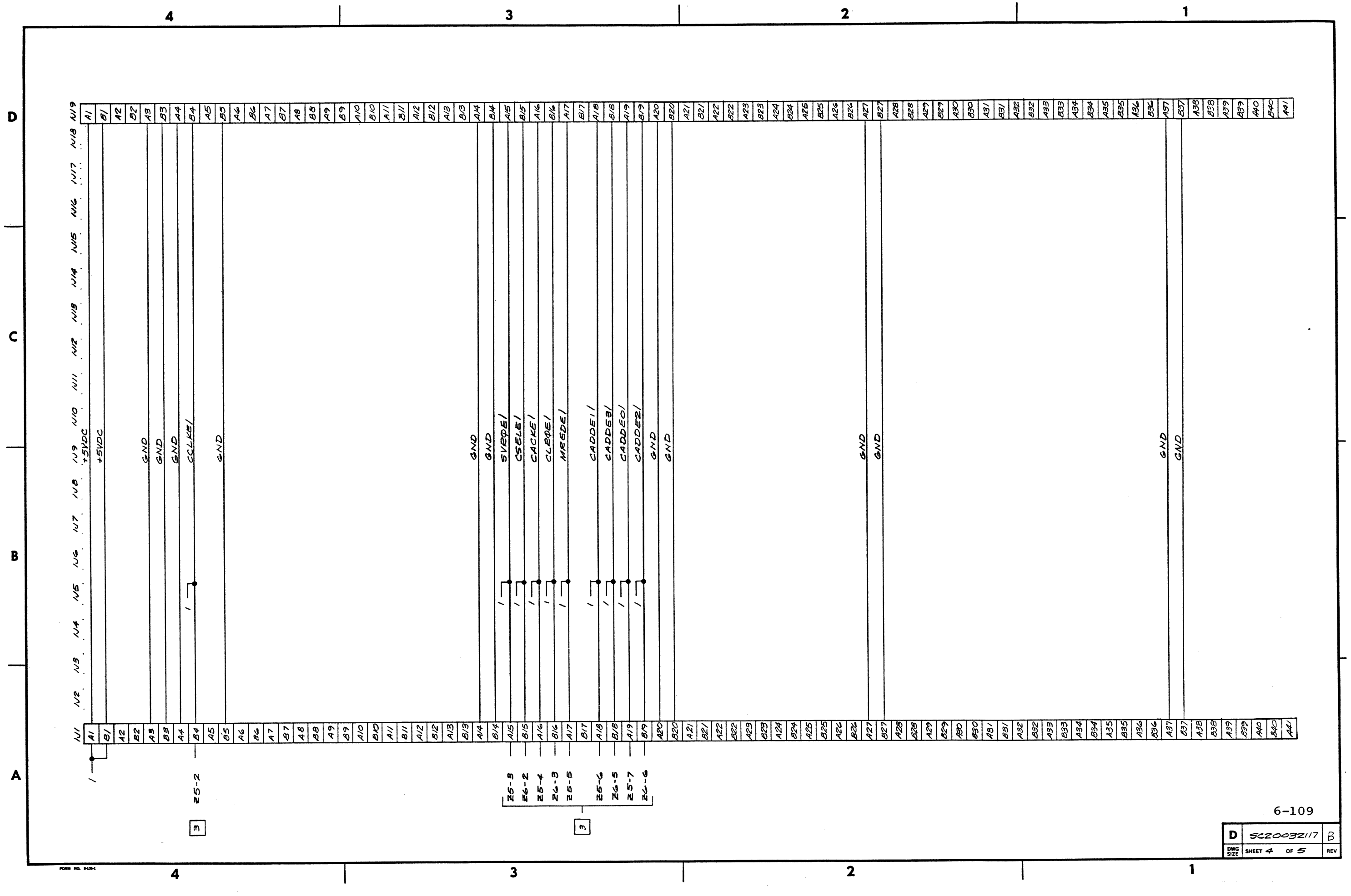
FORM NO. 9-135-1



6-107

D	SC20C32117	B
DWG SIZE	SHEET 3 OF 5	REV

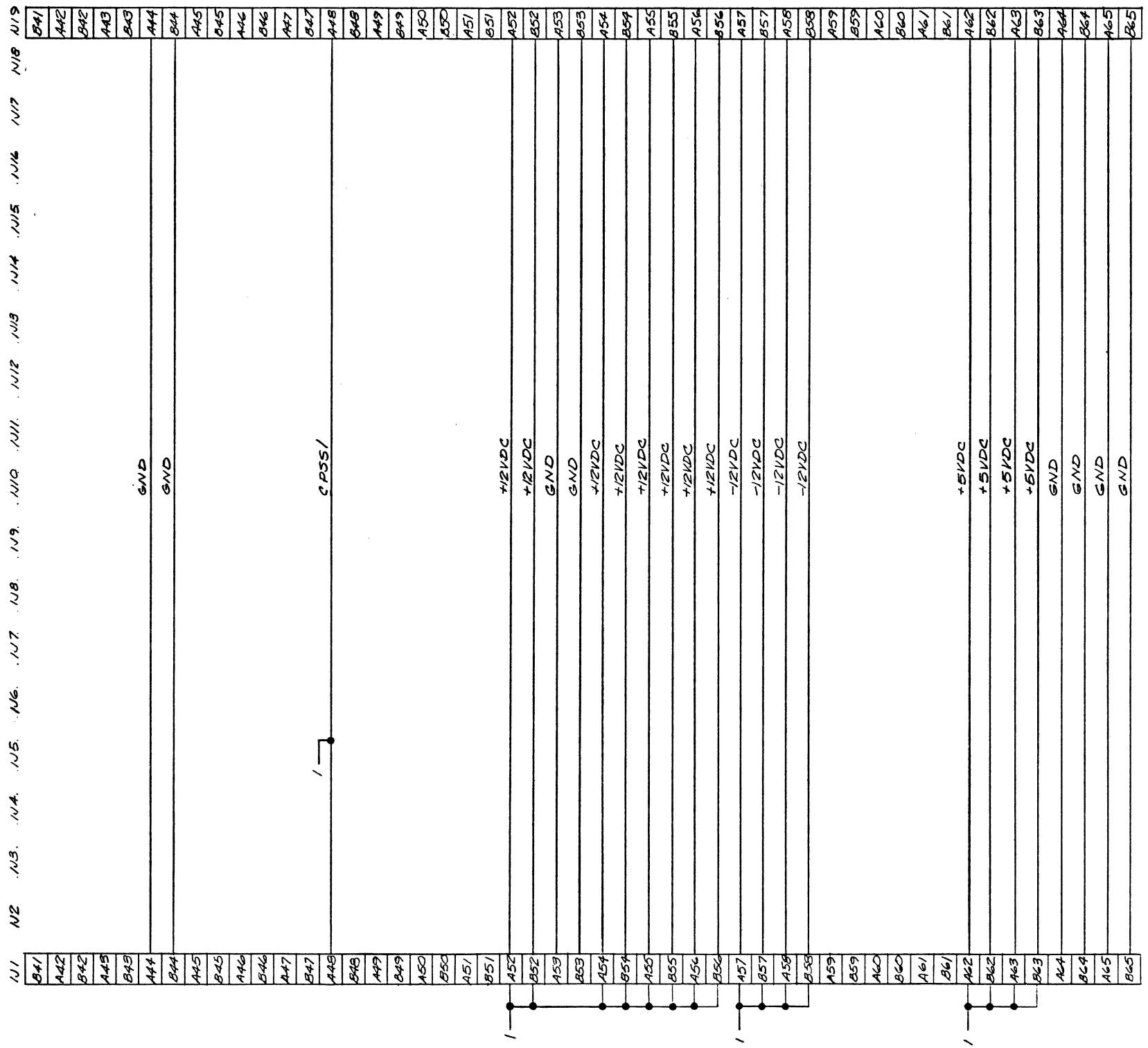
FORM NO. 9-130-1



6-109

D	SCZ0032117	B
DWG SIZE	SHEET 4 OF 5	REV

FORM NO. 9-135-1



6-111

D	5C20032117	B
DWG SIZE	SHEET 5 OF 5	REV

FORM NO. 9-130-1

APPENDIX A

TROUBLESHOOTING GUIDE

TROUBLESHOOTING

The system information provided in this section is not definitive but it highlights those areas where most problems will occur. Discussed are the buffer management tables, the layout of the monitor tables and the process identification blocks (PIBs). The monitor's logic flow and key elements of the I/O system are also presented.

The troubleshooting techniques include a list of procedures for eliminating hardware problems. The software techniques spotlight main memory locations that provide information in diagnosing problems.

Brief descriptions of the diagnostic programs help the user to select those programs that might fit the symptoms of a problem. The descriptions of symptoms guide the user from some evidence of a problem to an explanation of possible causes.

Terms

Terms and definitions that appear in Table A-1 will be useful in utilizing this section of the manual.

FIRMWARE DEBUGGER OVERVIEW

The firmware debugger is primarily designed for the debugging of the monitor, and is in fact often called the monitor debugger. Debugging of virtual programs is the function of the software debugger. The firmware debugger is also used to debug the software debugger. At customer sites the firmware debugger is useful for gathering information after the system fails or hangs up.

The debugger has several commands for controlling instruction execution. Other commands allow the user to change the contents of main memory locations and of address registers.

Firmware Instruction Execution Logic

When the firmware debugger is enabled, it is entered before the execution of each software instruction. Upon entry it tests to see if any of the following are true:

(a) The program counter is at a break point;
(b) A data match has occurred
(c) A specified number of instructions have executed
(d) The BREAK key on terminal zero was depressed.

If any of these conditions are true, the debugger will display the current software instruction counter and the first byte of the instruction next to be executed. If none of the conditions are true, the next instruction is executed.

TABLE A-1

TERMS AND DEFINITIONS

Term	Definition
Abort	The system traps to either the firmware debugger or the software debugger because of an error condition
Assembler debugger	Software debugger
BQ	Buffer queue
BSAFT	Buffer status and FID table
Buffer number	A two-byte value that points to a buffer
DCB	Debug control block, also known as the tertiary control block (TCB)
DMP	Diagnostic maintenance processor
FID	Frame identification; the frame number
Firmware debugger	A debugger, written in firmware, used mainly for debugging the monitor.
Hang (up)	Failure of the system or a terminal to respond correctly
Hash	A method for converting a FID to an index by truncating some of the bits
HAT	Hash address table
HLT	Hash link table
Interactive debugger	Firmware debugger
MTP	Magnetic tape/printer controller
PIB	Process identification block
PIB link	The number of the process corresponding to the PIB
Priority queue	A double-linked list of PIBs used by the monitor to select the next virtual process to run
RNI	Read next instruction; a loop in the instruction-execution firmware
SNU queue	Priority queue (so called because the select next user instruction is used)
Software debugger	A debugger, written in software, that substitutes for a process and uses the DCB instead of the PCB

The following is an overview of the firmware's instruction-execution logic:

```
Begin
 If there was an interrupt,
Then
 If the interval timer interrupted,
Then
 Detach the current process.
 Select a new process.
 Attach the new process.
 Load the new process.
Else
 Process interrupts.
Else
 If firmware debugger is enabled,
Then
 If program counter is at a break point address, or data match
 has occured, or a specified number of instructions have executed,
 or the BREAK key on terminal zero has been struck,
Then
 Enter the firmware debugger.
 Fetch and decode software instruction.
 Call the instruction routine.
 If a register is detached or if the instruction is crossing a
  frame boundary,
Then
 If a frame is needed from disc,
Then
 Detach the process.
 Start disc I/O.
 Select a new process.
 Attach the new process.
 Load the program counter.
Else
 Correct the detachment or crossing frame condition.
Else
 Process the instruction.
 Increment the instruction counter.
End
```

Activating the Firmware Debugger

There are several ways to enter the firmware debugger. The normal way is to turn the front panel rotary switch to TEST and type "Control-F" on port zero's terminal. A second, similar way is to type "Control-D" instead of "Control-F" on terminal zero, causing the DMP's control menu to be displayed. Typing a "3" in response to the menu activates the debugger.

The firmware debugger is also activated by an error in the monitor that in a virtual process would cause an abort. When the debugger is so entered, the message "MONITOR ERROR," followed by a decimal number is displayed. The number corresponds to an entry point in the software debugger that would have been taken had the error occurred in a virtual process.

Another way to activate the firmware debugger is by means of the software instruction FWDB (X'1C'). When executed by a process, this instruction initializes the debugger. Once the debugger is active, it can be re-entered at any time by striking the BREAK key. Note that this is preferable to using Control-F, which re-initializes the debugger, clearing break points, match bytes and the instruction trace.

Initial Display

When the firmware debugger is activated at the beginning of any instruction it produces a display on the port zero terminal with the following elements:

(1) An indication of why the debugger was entered:
E indicates instruction counter interrupt
B1 indicates break address 1 encountered
B2 indicates break address 2 encountered
BKEY indicates BREAK key activated or FWDB executed
T indicates match byte encountered
MASK indicates mask-mode match byte

(2) The location (six hexadecimal digits) of the instruction to be executed next.

(3) The first two hexadecimal digits of the next instruction.

(4) V for virtual process if not in monitor, followed by the port number if not port zero.

(5) SWDB if the process is in transition to the DCB; that is, to the software debugger.

Examples:

```
E 07E098 1E VE
BKEY 07E0B3 64 V5
```

Commands

The debugger prompt is a ">". Commands are those strings that are entered in response to the prompt character. Some commands have subcommands. These will be defined under the command.

1. Address Format

In the descriptions of the commands a main memory address is represented as "xxxxxx". This stands for a one to six digit hexadecimal number. Thus, when using command Mxxxxxx to display memory location X'1AC', enter any of the following:

```
M1AC
M01AC
M001AC
```

2. Control Commands

Control commands with their effects are described as follows:

Bxxxxxx - Sets the address of a break point. If an instruction is about to be executed at that address, the debugger will be entered and will solicit a command. There are two break points available. This command will set the first break point only if it is undefined. If the first break point is set, this command will set the second break point whether or not it is set. The state of the break points may be checked by giving the "D" command (described below).

Note that the break points are cleared when the debugger is entered with a Control-F.

Example:

```
>B7E0B5
>G
B1 07E0B5 F0 V5
```

K - The "K" (KILL) command turns off both break points.

Exxx - This command sets a counter which is decremented by 1 at each software instruction. When the counter reaches 0, the debugger will interrupt and solicit a command. It will then reset the "E" counter to the last value entered by an operator. The range of xxx is 1 to 3 decimal digits.

An entry of "E0" will turn this feature off. An entry of "E1" will cause an interrupt for each software instruction.

The counter is automatically set to "E1" when the debugger is entered with a Control-F.

G or Gxxxxxx - The "G" (GO) command, if followed only by a RETURN, causes the execution of the instruction at the location displayed at the beginning of the break.

If a one to six digit hexadecimal address is appended to the "G" and followed by a RETURN, instruction execution begins at that address.

Note that the "G" command does not change R1 of a virtual process. Hardware address register R1 would have to be changed by the "AR" command before the "G" is entered. Also, the register's FID field (R1FID) would have to be set with the FID using one of the memory-modify commands ("M" or "W").

A "G" and a LINE FEED are functionally equivalent

LINE FEED _ A LINE FEED causes the execution of the software instruction at the address printed by the debugger at the beginning of the break. This is functionally equivalent to a "G" with no parameters.

Txxxxxxzyy - This command sets one of the two match bytes. The one to six digit address is defined by xxxxxx. The "z" defines the test to be made, "=" causes an "equal to" test, and "#" causes a "not equal to" test. The "yy" is a one or two hexidecimal digit value which is used to compare the byte at the defined memory address.

If one digit is given, the debugger supplies a leading zero. If the first match byte is not set, this command will set it. If the first match byte is set, this command will cause the second to be set regardless of its current status.

Note that the match bytes are cleared when the debugger is entered with a Control-F.

This command functions in an "OR" mode or in an "AND" mode. Initially, the debugger is in the "OR" mode, but the "AND" and "OR" commands can be used to switch from one mode to the other.

Example:

```
>T1234#1
>G
T 07E0B3 64 V5
```

The "T" command can be used in a special way to mask the contents of a byte before comparing it with a match byte. This is done in AND mode my making both match bytes reference the same byte address. When this is done, the first match byte is used to mask the data, and the result is compared against the second match byte. The "z" parameter of the first match byte has no meaning, but it must be

present in the command. The "z" parameter of the second match byte may be "=" or "#". and the compare will be done accordingly.

Example: To break when bits two and six of location X'1480' equal zero and one, respectively, enter the following commands:

```
>T1480=22
>T1480=02
```

OR - OR causes a break if either match byte is true.

AND - AND causes the break to occur if, and only if, both match bytes are evaluated as true. If both match bytes are not set, there can never be a match byte break while in the AND mode.

U (UNTRACE) - U causes the match bytes to be turned off.

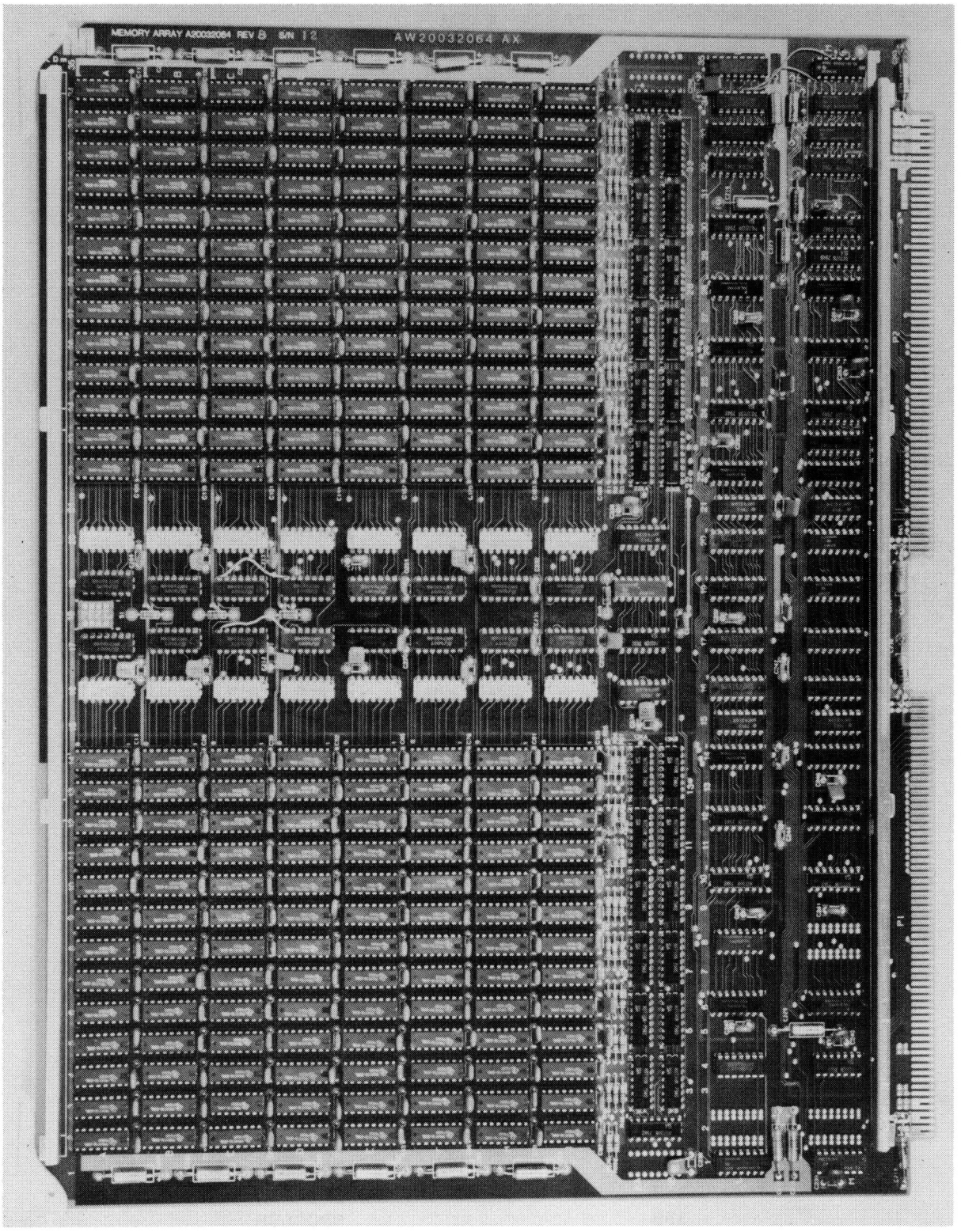
X - X causes the firmware debugger to be turned off. Execution of software instructions is resumed at the location displayed when the break occurred.

3. Display and Modify Memory Commands

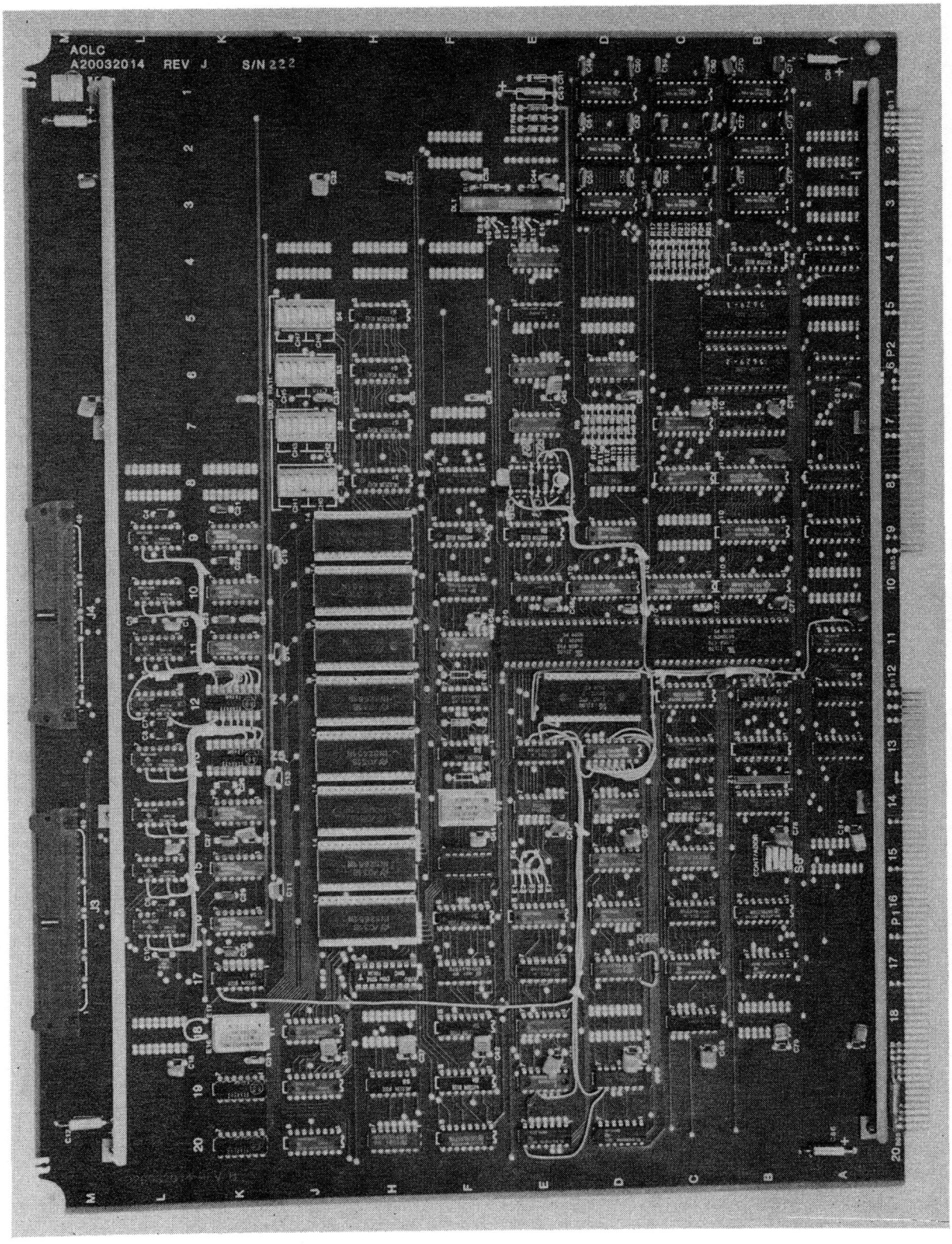
These commands display the contents of memory. The display is followed by an equal sign (=), which invites new data to be entered into the memory entity displayed. The size of the entity and the format of the displayed data and replacement data depend on the command. Whether or not replace data is entered, the action to be taken next is specified by one of the following delimiters:

Delimiter	Description
RETURN	Terminates the display mode; goes to next command
LINE FEED Or space	Displays the next entity's data
Control-N	Displays the next entity's address and data
Control-P	Displays the previous entity's address and data.

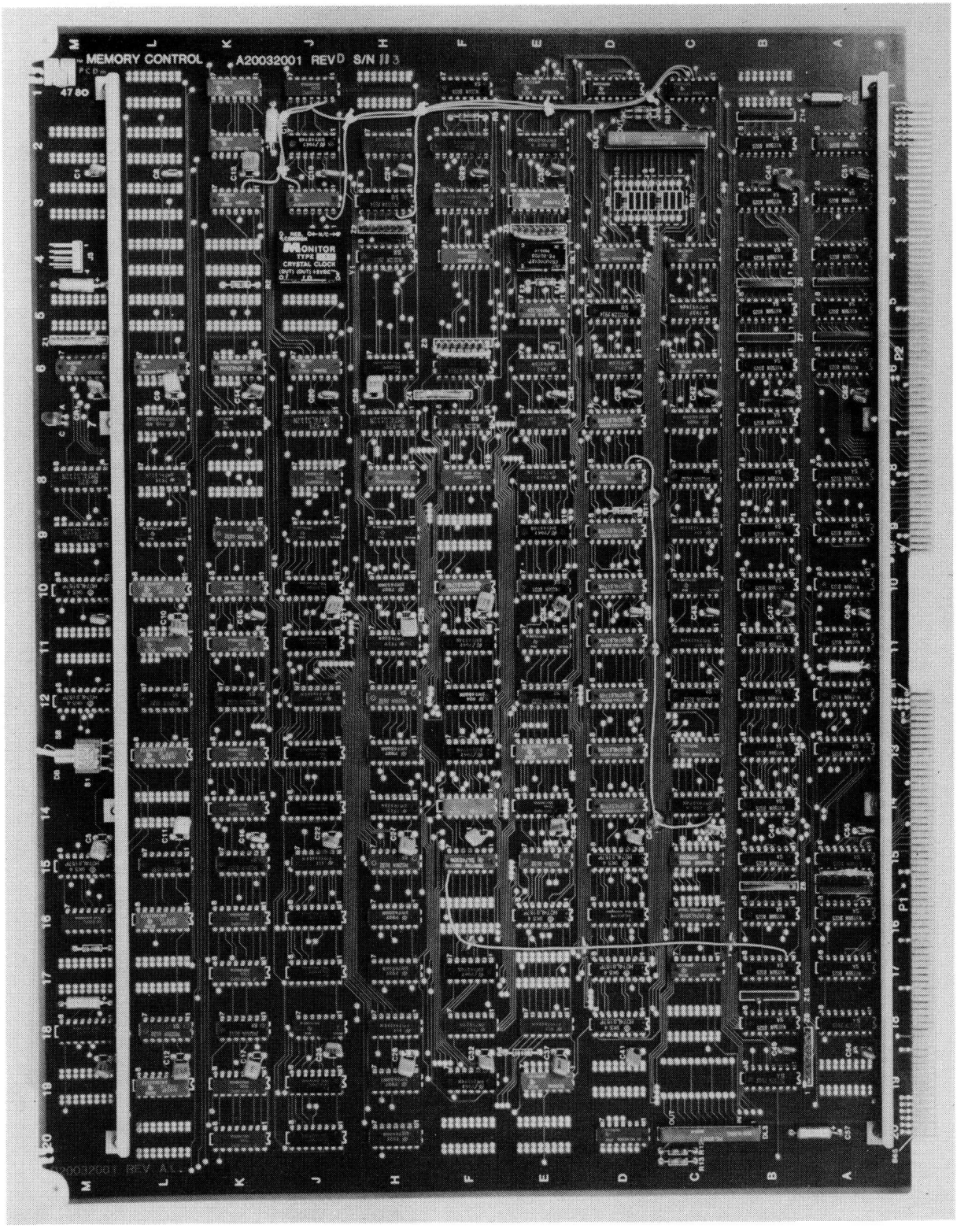
Cxxxxxx - The C command causes the memory byte at "xxxxxx" to be displayed. The contents of the byte are displayed as a single ASCII character if it is a displayable ASCII character. If it is not, the contents of the byte will be displayed in the format X'hh', where "hh" is a pair of hexadecimal digits.



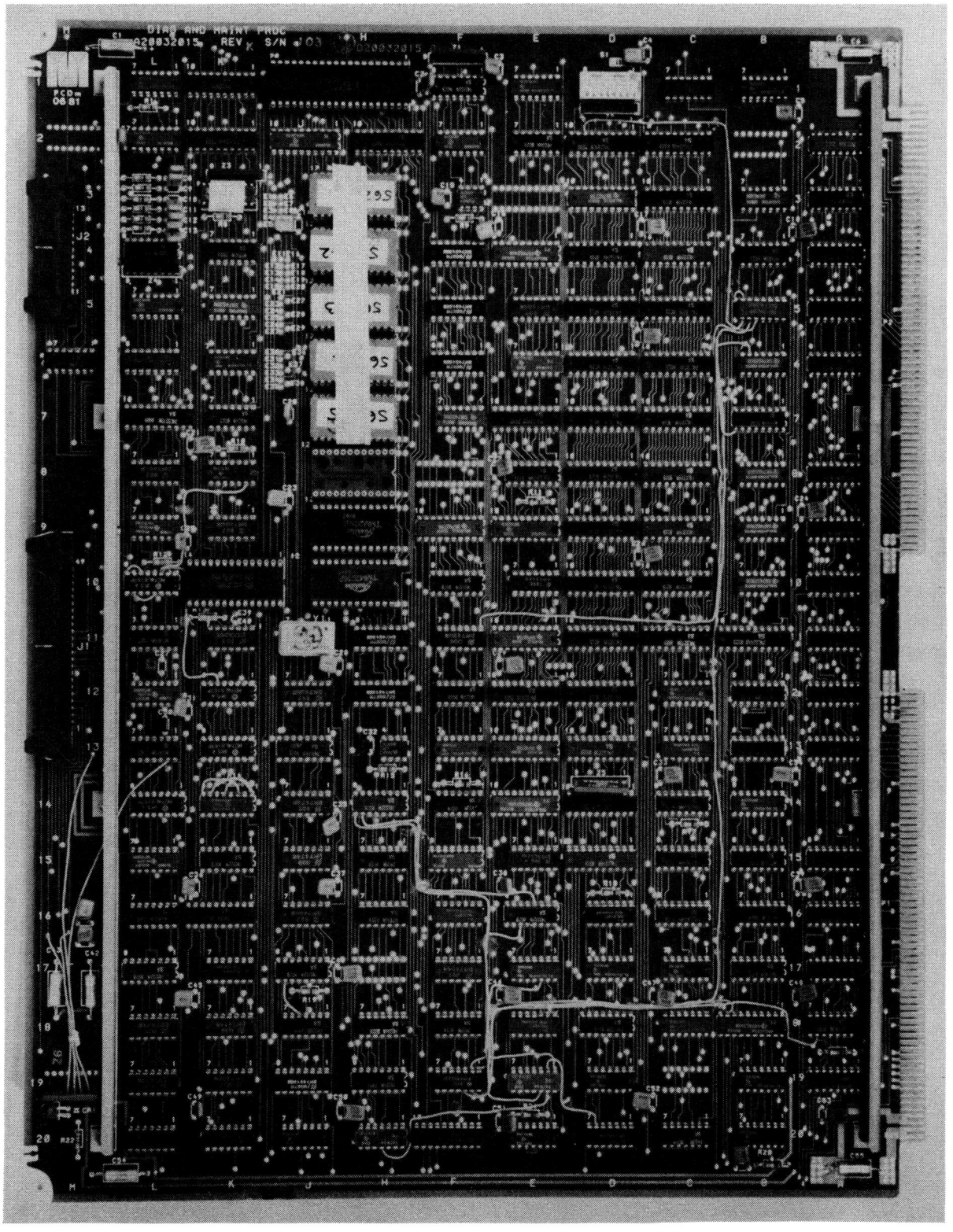
MEMORY ARRAY BOARD (MAB)
A20032064



ASYNC COMM CTRLR (ACLC)
A20032014



MEMORY CONTROL BOARD
A20032001



DIAGN MAINT PROCESSOR (DMP)
A20032015

The data displayed may be replaced by any displayable ASCII character by typing that character, followed by a delimiter. To insert other than a displayable ASCII character, use the "M" command.

Examples:

```
>C123E H=
>CDE34 X'00'=
```

Mxxxxxx - The M command displays the contents of memory byte "xxxxxx" as two hexadecimal digits. To change the byte's contents enter a one or two digit hexadecimal number followed by a delimiter.

Example:

```
>MA4F80 9C=
```

Wxxxxxx - The W command displays the contents of the 32-bit memory word at location "xxxxxx" as eight hexadecimal digits. The word address is always modified as necessary to begin on a word boundary; that is, the last digit is rounded down to a 0, 4, 8, or C. To change the contents of the word displayed, enter a one to eight digit hexadecimal number followed by a delimiter. The debugger will supply leading zeros when the data is less than eight digits long.

Examples:

```
>W1234
001234 5F9A430B=
>W67ED
>0067EC 3062076
```

4. Display Information Commands

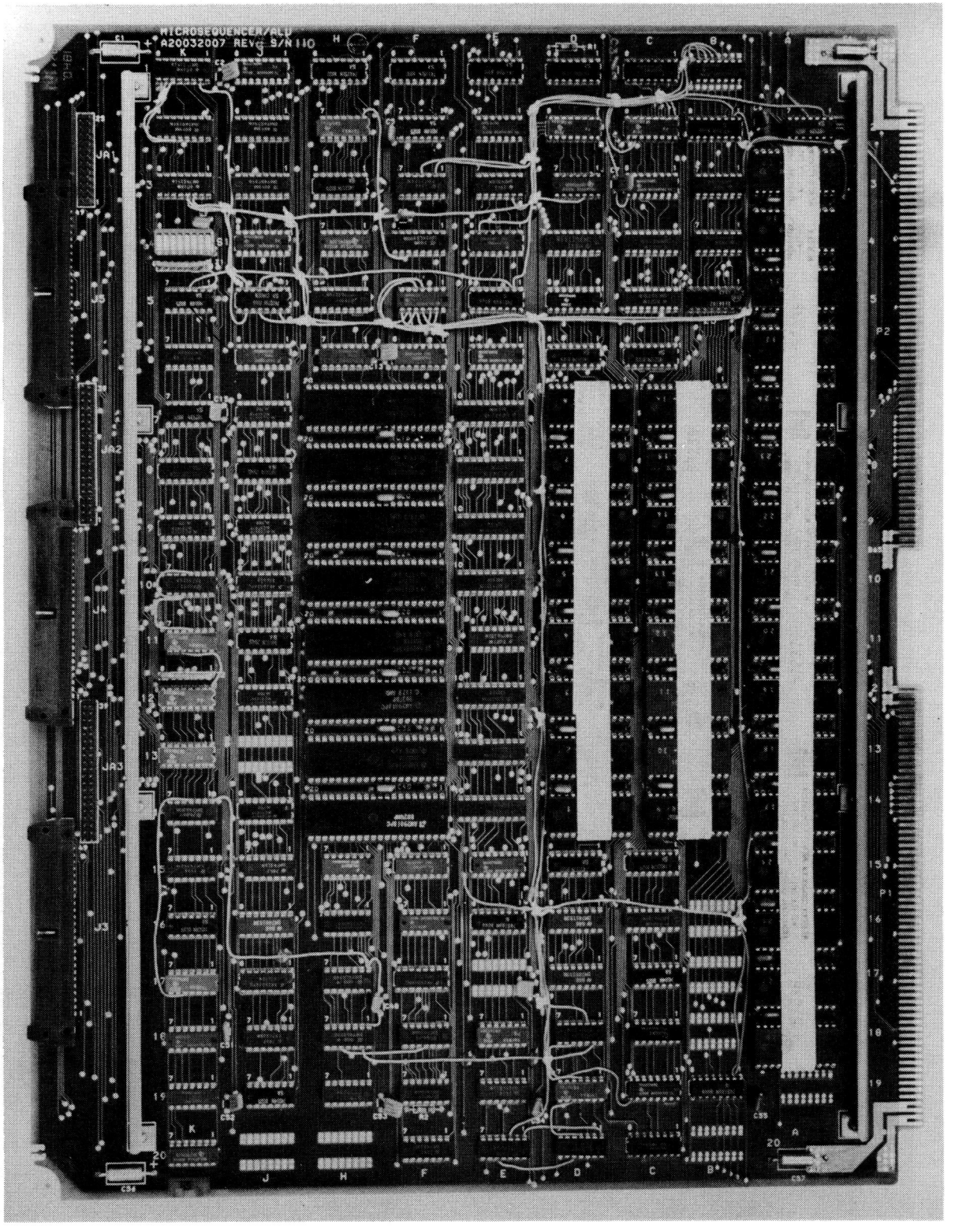
Display information commands and their effects are as follows:

D - This command (which does not require a RETURN or other terminator), causes the display of all information set by the debugger to control the occurrence of breaks. The display is self-explanatory to those familiar with the commands described herein.

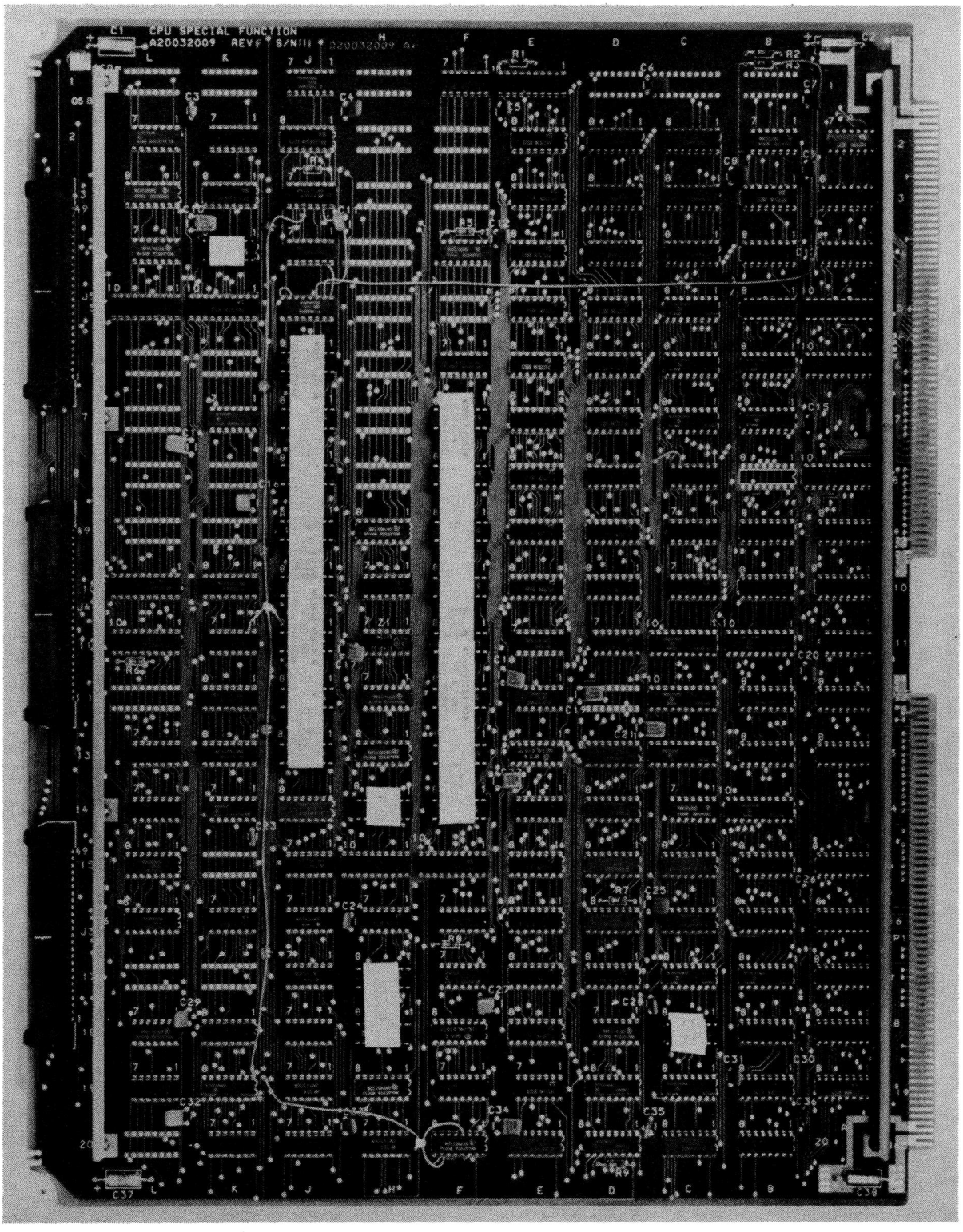
Example:

```
>D E=OFF 0000 BP1=ON  07E0B5  BP2=ON 07E0BF
MATCH IS MB1 OR MB2=ON 0D1B34 01 MB2=OFF 000000 00
```

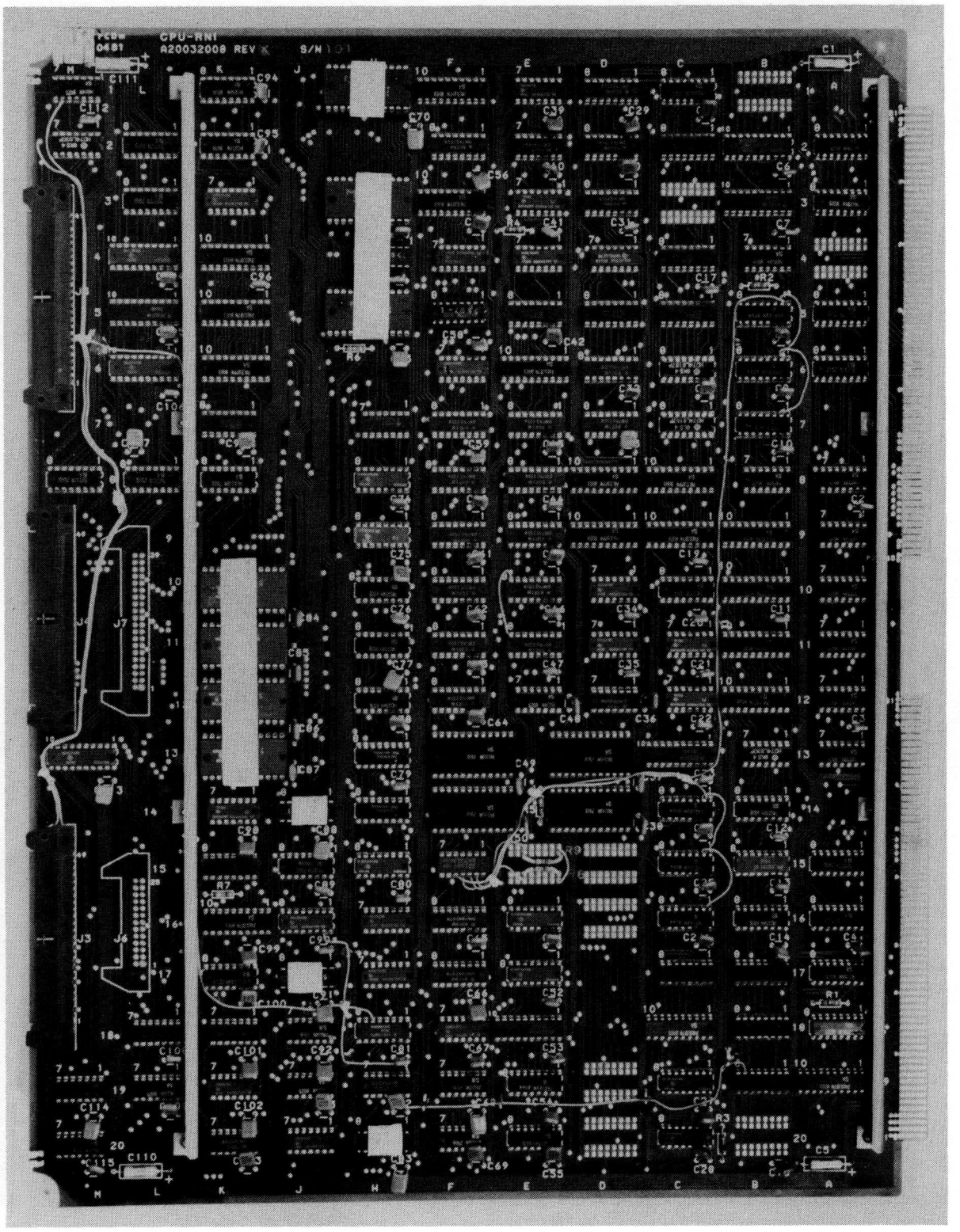
Fxxxxxx - The "F" (FID) command converts the memory address "xxxxxx" to the FID and offset of that location. The command is activitated by a RETURN.



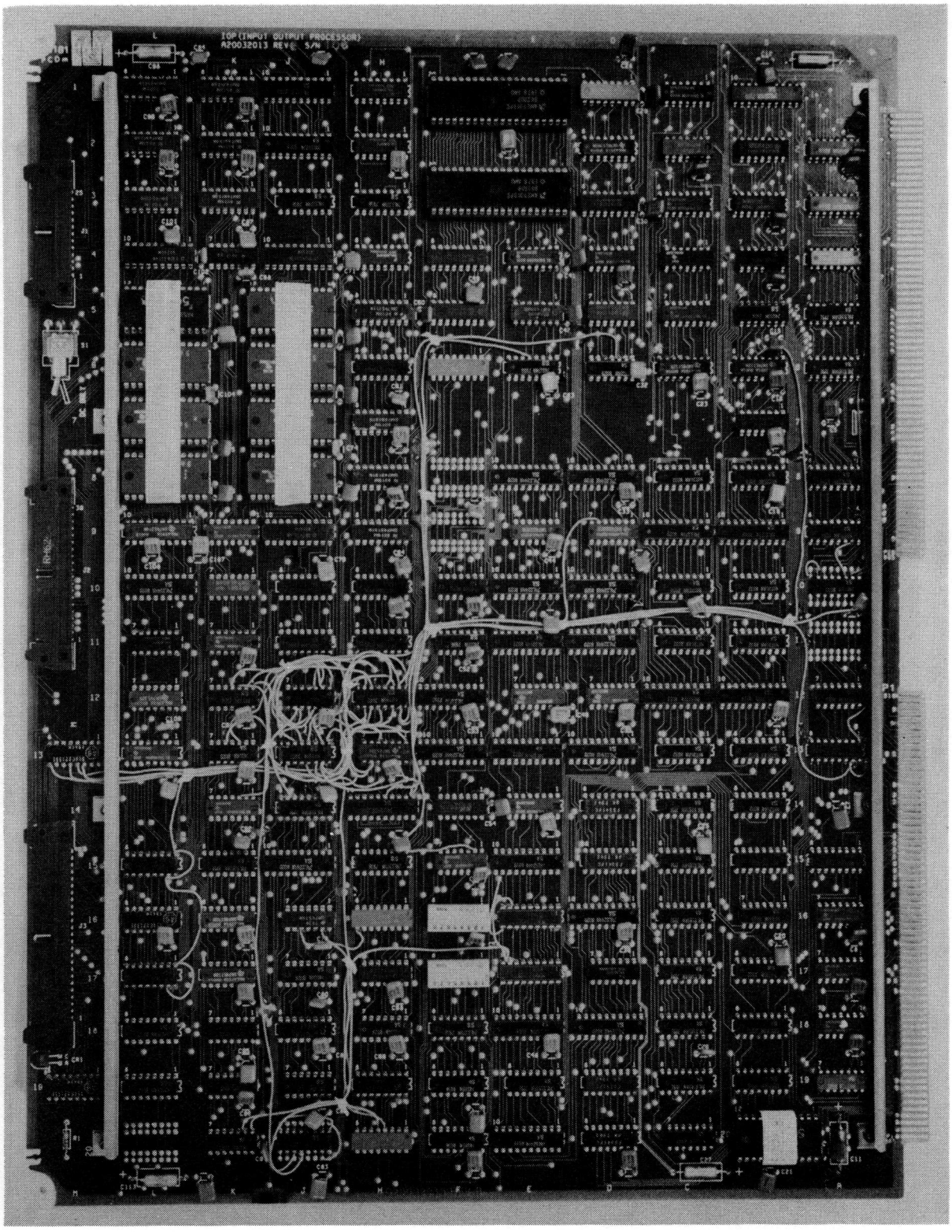
ARITHMETIC LOGIC UNIT (ALU)
A20032007



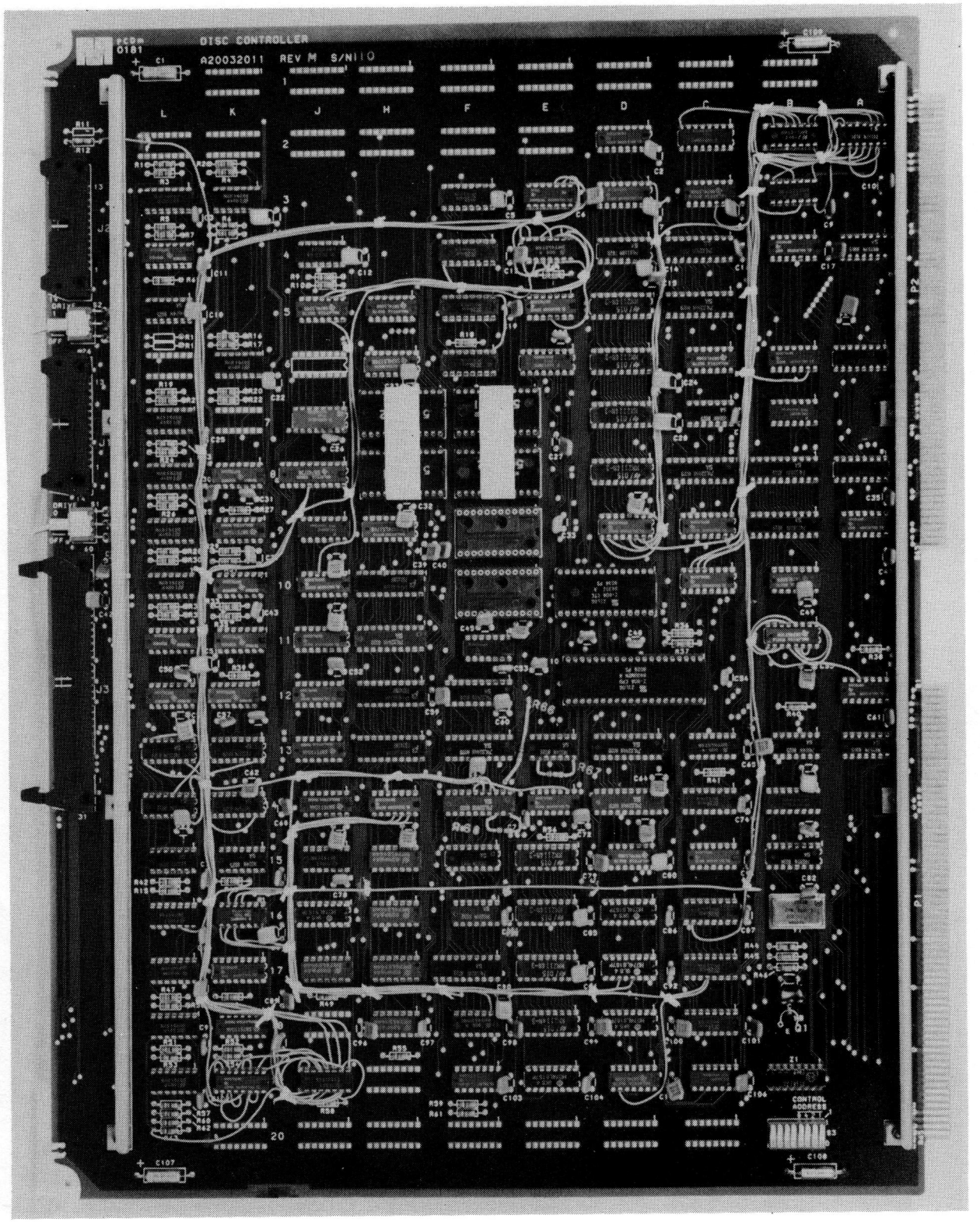
SPECIAL FUNCTION
A20032009



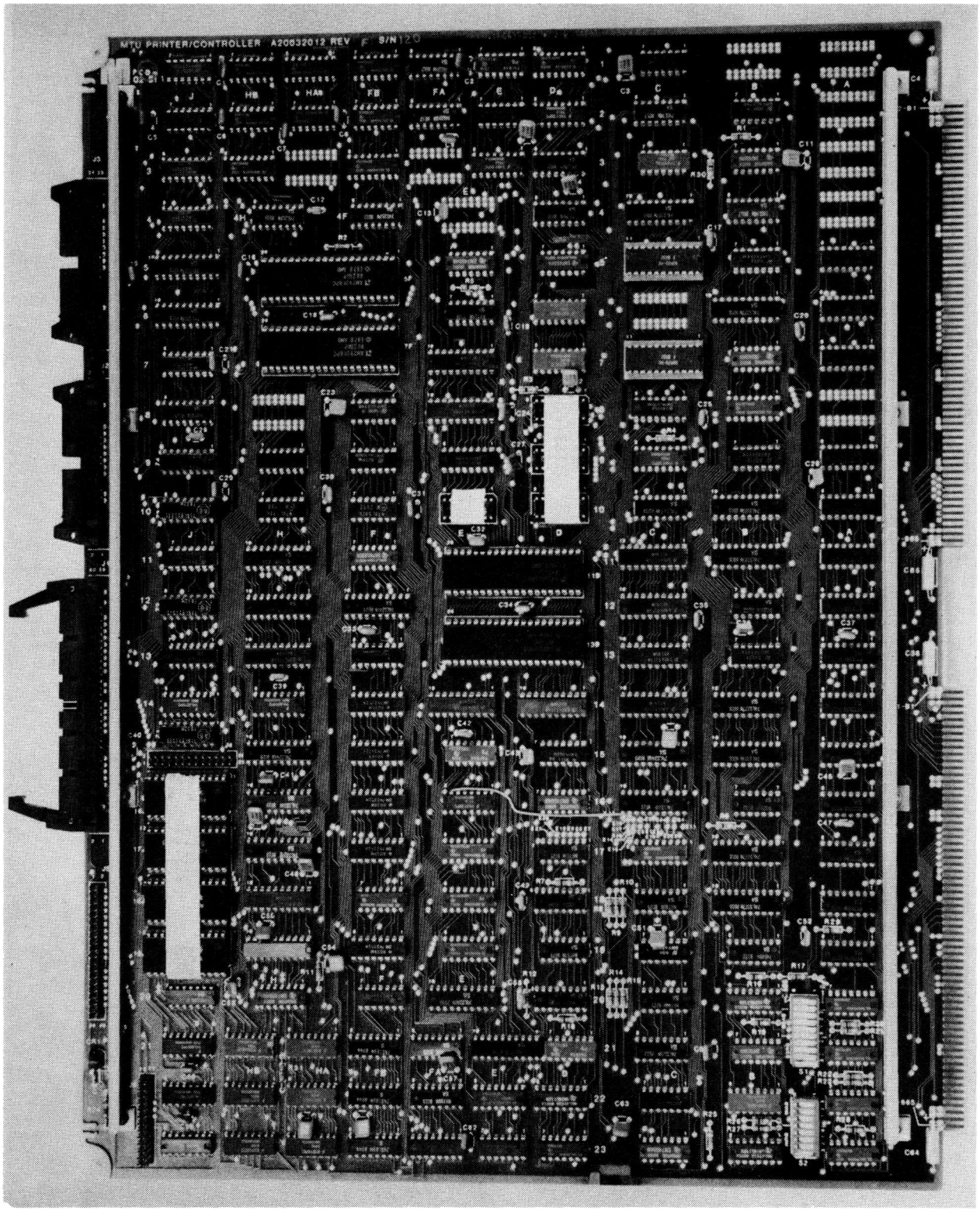
READ NEXT INSTRUCTION (RNI)
A20032008



I/O PROCESSOR (IOP)
A20032013



DISC CONTROLLER
A20032011



MAG/TAPE PRINTER CTRLR (MTP)
A20032012

If the memory address is in a buffer that is I/O busy, a message reporting that will be printed.

Example:

```
>F67ED
118.01ED
```

PDyyyy - PD converts the decimal port number ("yyyy") to a display of the decimal port number, the hexadecimal port number and the hexadecimal main memory address of the port's PIB.

Examples:

```
>PD3  03  03  001580
>PD15  15  OF  001B80
```

PXxx - PX converts the hexadecimal port number ("xx") to a display of the decimal port number, the hexadecimal, port number and the hexadecimal main memory address of the port's PIB.

Examples:

```
>PX3  03  03  001580
>PXF  15  0F  001B80
```

AFxx - This command causes the display of an address register and its associated flags. The value "xx", which may be 0 to 15 decimal or 0 to F hexadecimal, is the register number.

The debugger will display the contents of the address register as a six-digit hexadecimal number, followed by two hexadecimal digits. The low-order digit of these two represents the flags. The meaning of the flag bits is as follows:

Bit	Meaning
First	Attached
Second	Linked
Third	Write required
Fourth	In process of attachment

The two digits are followed by an "=", which invites a change. If no change is required, enter a RETURN. To change the value of the flags, enter a one or two digit hexadecimal number followed by a RETURN. The last digit entered before the RETURN will be used to update the flags.

Examples:

```
>AF10  001600  00=
>AF4   044984  08=34
>AF4   044984  04=
```

ARxx - AR causes the display of an address register. The value "xx", which may be 0 to 15 decimal or 0 to F hexadecimal, is the register number. The register contents are displayed as six hexadecimal digits followed by an "-". If no change to the contents is desired, strike the RETURN key. To make a change enter one to six hexadecimal digits followed by RETURN.

Examples:

```
>AR10  001600=
>ARE   0D13C7=001680
```

Z - Z causes a trace of the previous 62 instructions and the current instruction to be displayed. If fewer instructions have been operated since the firmware debugger has been activated, only those will be displayed.

Monitor instructions are displayed with an M followed by the memory address of the instruction. Virtual instructions are displayed with a leading "V" followed by the main memory address and the virtual storage FID and displacement.

Note that the instruction trace is cleared when the debugger is entered with a Control-F.

Cancelling Keystrokes and Erroneous Data

The firmware debugger terminal I/O uses the DMP rather than the ACLC. Consequently, it does not allow the use of the BACK-SPACE to cancel the previous character as does normal terminal I/O. However, any improper input will cancel the debugger's input request. Improper input is non-numeric data when numeric data (0-9 for decimal, 0-F for hexadecimal) is required, or it is too many characters being entered. No error indication is given when improper data is entered. Most of the commands allowing changes to be entered return to the command mode when erroneous data is entered. The M, W, and C commands, however, do not return to the command mode when erroneous data is encountered. They simply request more data for the same location but without any obvious indication that it is for the same location.

Since there are no error messages provided for improper input, verify changes by displaying the locations changed.

Virtual Program Debugging

Although the firmware debugger is intended for debugging the monitor, some virtual program debugging must be done with it. In particular, the software debugger is normally debugged with the firmware debugger. Caution is required when attempting this. There is no provision in the firmware debugger to locate the absolute address of a FID. This can normally be acquired by the display which accompanies locking of the FID. The firmware debugger allows branching to any absolute address, but does not attach any buffer which has been branched to, as the software debugger does. In fact, the firmware debugger never changes the status of any buffer or register except by direct user command.

The execution of a branch instruction in virtual mode to a monitor instruction will not change the internal mode to monitor mode, and therefore execution of a monitor-only instruction will cause a "PRIVILEDGED INSTRUCTION" abort.

If there is a branch while in monitor mode to virtual instructions, and an abort condition ensues, the message "MONITOR ERROR" will be displayed, and the firmware debugger will be entered.

SYSTEM INFORMATION

All of main memory is divided into 512-byte buffers. The buffer management tables enable the monitor and the firmware to record each buffer's status and keep track of the frames that have been read into main memory.

Buffer Management Tables

There are four buffer management tables:

Buffer status and FID table (BSAFT) contains the status of each buffer and the FID of the frame currently residing in the buffer.

Hash address table (HLT) is a group of hash buckets pointing to lists of FIDs in the has link table (HLT).

Hash link table consists of doubly-linked lists of buffers containing frames whose FIDs hash to the same bucket.

Buffer queue is a doubly-linked list of all usable buffers ordered accoring to their time of attachment by the firmware.

BSAFT, HLT, and BQ each have one four-byte entry for each buffer in the system. The table entries are ordered according to

the order of the buffers in memory. HAT has two-byte entries, one for each possible hash result.

Buffer Numbers

A buffer number is a two-byte value that points to a buffer. Since buffers are 512 bytes long, their beginning addresses are multiples of 512 (X'200'). The buffer number is the buffer address divided by X'100'. Stated another way, a buffer number is the buffer address with the low-order eight bits lopped off. For example, the buffer number of the buffer beginning at X'00AC00 is X'00AC'.

Buffer Status and FID Table (BSAFT)

The BSAFT contains the status of each buffer and FID of the frame currently occupying the buffer. The first byte of each entry contains the status; the last three bytes, the FID. In the REALITY system this information is maintained in two tables.

The BSAFT begins at main memory location X'A000' for all systems. The length of the table is determined by the number of buffers in a system.

The status is updated automatically by assembly instructions which detach registers, the FAR instruction, terminal I/O instructions and the terminal interrupt-handling firmware. The information contained in the status byte is as follows:

Bits (Left to right)	PSYM Name	Description
0	IOBUSY/	Zeroes when I/O is in progress for this buffer; set when I/O completed. Firmware prevents attachment by a virtual process to a buffer with this bit zero, causing a trap to entry 3 of the monitor.
1	PERMLOCK/	Zero indicates the buffer is to remain permanently core-locked.
2	MWRITE/	Zero indicates that the monitor is in the process of writing this buffer out to disc. Virtual processes may still attach to it and modify it. (Formerly TEMPLOCK/ or CORELOCK/) This bit has no special meaning to the firmware, so the software may use it differently in the future.
4-7	WRTZD/	These bits are used by the FAR instruction.

Null BSAFT entries contain X'FF000000'.

Hash Address Table (HAT)

In the HAT, a group of hash buckets pointing to lists of FIDs in the HLT, each entry is two bytes long. The HAT must begin on a 256-byte boundary and the number of bytes in the table must be a multiple of 256 because the HLT, which follows it, must be on a 256-byte boundary. Locations X'200-X201' contain the buffer number of the buffer at which the HAT begins.

The register attachment firmware uses a hash code search to look up FIDs in the BSAFT. The low-order bits of the FID (times two) are used as the hash code to select one of the entrees (i.e., hash buckets) in the HAT. The HAT entry points to the head of a list of buffers in the HLT. The firmware searches this list to determine if a particular FID is in main memory.

The number of FID bits used to access the HAT depends on the length of the HAT, which the firmware determines by taking the difference between the HLT's buffer number in locations X'202-X'203 and the HAT's buffer number in X'200-X'201'. For instance, if the HLT's buffer number is X'01B8' and the HAT's is X'01B0', the difference between their buffer addresses would be X'01B800' - X'01B000' = X'800'. This value would be divided by two to get the number of entries in the HAT (X'800' / 2 = X'400'). One would be subtracted from the number of entries to yield a mask for ANDing with the FID (X'400' - 1 = X'3FF'). After the FID was masked (e.g., X'3FF AND X'BAC23' = X'023'), the result would be multiplied by two to point to a HAT entry (X'023' *2 = X'046'). If the result of the ANDing is zero, one is used instead, because HAT entry zero cannot be used.

The maximum length of the HAT is 2**15 entries because the backward link of the head of a HLT list uses 15 bits to point back to its associated HAT entry. There are only 15 bits because the low-order bit of the backward link is used to indicate the head of the list. Furthermore, HAT entry 0 is unusable as a bucket because a HLT backward link of X'0001' indicates an empty list.

The HAT and HLT are updated using the change FID option of the FAR instruction.

The following values may exist for a HAT entry:

Value	Description
X'FFF'	Indicates that there are no FIDs in memory buffers that hash to this address.
X'NNNN'	X'NNNN' is the offset address (a value divisible by four) of the head of a list of buffers in the HLT, indicating that at least one FID hashes to this address.

Unused HAT entries contain X'FFFF'.

Hash Link Table (HLT)

The HLT contains one list of memory buffers for each HAT entry. Each list consists of buffers whose FIDs hash to the same HAT entry. The HLT is located immediately after the HAT is main memory. The buffer number of the HLT's location is stored in memory locations X'202' - X'203'.

The first two bytes of each HLT entry contain the backward link; the next two bytes contain the forward link. This is the opposite order from REALITY.

The following values may exist for the forward link:

Value	Description
X'FFFF'	Indicates that this HLT entry is the last entry (or tail) of the HLT list.
X'NNNN	X'NNNN' is the offset address (a value divisible by four) of the next HLT entry in the list.

The following values may exist for the backward link:

Value	Description
X'NNNN'	If the low-order bit is set, X'NNNN' indicates that this HLT entry is the first entry (or head) of the list. In this case, the upper fifteen bits of X'NNNN are the entry number of the HAT entry that points to this list. Simply zeroing the low-order bit of the sixteen bits gives the HAT offset address. If the value is X'0001', it indicates an empty list.
X'NNNN'	If the two low-order bits are zero, X'NNNN' is the HLT offset of the previous HLT entry in the list.

HLT entries that are unused contain X'FFFF' in the forward link and X'0001' in the backward link.

Buffer Queue (BQ)

The BQ consists of a doubly-linked list of buffers ordered according to their time of attachment by the firmware. Each time a register is attached to a buffer, the register attachment firmware moves the attached buffer to the head of the BQ. When the contents of a buffer must be replaced because of a frame fault, the BQ is used to identify the

least recently attached buffer for replacement. Buffers are removed from the queue when they become either permanently corelocked or I/O busy, and they are reinserted at the top when they become both unlocked and not I/O busy. I/O busy buffers are removed from the queue to prevent the READ instruction from clogging the tail of the queue and to prevent them from being immediately swapped out when I/O completes.

The first two bytes of each BQ entry point to the next more recently attached entry. The next two bytes of each BQ entry point to the next less recently attached entry. Note that if the first two bytes of a BQ entry are X'000', the entry is either on top of the queue or out of the queue entirely. The move to the top (MTP) firmware routine will do nothing if the first two bytes are zeros.

The following values may exist in a BQ entry's link fields:

Value	Description
X'NNNN'	X'NNNN' is the offset address (a number divisible by four) of the next BQ entry in the list.
X'0000'	Indicates that this entry is the head or tail of the BQ list.

The BQ starts at main memory location X'1A000' with the four bytes at locations X'1A000' - X'1A003' containing the base pointer of the BQ (therefore BQ[0] is unusable). The first two bytes of the base pointer contain the pointer to the tail of the BQ (i.e., the least recently attached buffer). The next two bytes contain the pointer to the head of the BQ (i.e., the most recently attached buffer).

Memory Map Layout

Reserved buffers, table layouts and PIB layouts are presented in the following pages.

1. Reserved Buffers

The buffers listed in Table A-2 should be corelocked by the configurator except for those marked with an asterisk (*) which may be available to the virtual memory manager if there are less than 128 processes.

TABLE A-2

RESERVED BUFFERS

Location	Contents
0-1FF	MTABLES0
200-3FF	MTABLES1
400-5FF	MTABLES2 (DEVICE CONTROL TABLES)
600-7FF	MONITOR1
800-9FF	MONITOR2
A00-BFF	MONITOR3
C00-DFF	MONITOR4
E00-FFF	MONITOR5
1000-11FF	MONITOR6
1200-13FF	RESERVED FOR MONITOR EXPANSION
1400-93FF	* RESERVED FOR UP TO 256 128-BYTE PIBS
9400-95FF	TERMIO CR/LF/DELAY BUFFER
9800-99FF	FIRMWARE DEBUGGER TRACE
9A00-9FFF	3 DIAGNOSTIC BUFFERS
A000-	BUFFER STATUS AND FID TABLE
1A000-	BUFFER QUEUE
Anywhere	HASH ADDRESS TABLE
After HAT	HASH LINK TABLE

2. Layout of Tables

The layout of tables is by hexadecimal location. The type is the assembler data type. The symbol in the name column is defined in PSYM, the assembler's permanent symbol table. The layout of MTABLES0 is presented in Table A-3.

TABLE A-3

LAYOUT OF MTABLES0

Location	Type	Name	Description
001	C	ACF	Arithmetic condition flag bits
008-00B	D	D1	Accumulator extension
00C-00F	D	D0	Accumulator
010	H	SHPP	Port number of the super high priority process. This is used by SNU firmware.
011	H		Miscellaneous bits
011:B1	B	BINARY	Set when binary save/restore is running.
011:B2	B	MEMERR	Set by the firmware upon a memory error interrupt.
011:B3	B	MONREAD	Set when the monitor does terminal I/O.
011;B4	B	COLDFLAG	Set during a coldstart.
011;B5	B	DERR	Set if disc errors are to be recorded.
011;B6	B	FARBIT	This bit is used by the FAR instruction.
011;B7	B	DATEFLAG	Set by 24-hour interrupt handling firmware to tell the spooler to update the externally-formatted copy of the system date.
012	H		Miscellaneous bits
012;B5	B	REFAIL2	Flag used by power restart software.
012;B6	B	REFAIL1	Flag used by power restart software.
012;B7	B	REFAIL0	Flag set by firmware upon a power fail and reset by the software once it has finished with the power up routine. If these is a power fail while this flag is set the firmware will not save the state of the machine.
014	H	PIBTOP	Port number of the first process on the PIB queue.
015	H	PIBBOTTOM	Port number of the last process on the P13 queue.
01A-01D	D	ICW	Interrupt control word. When the firmware receives an interrupt for a device in the range X'D0' - X'EF', it sets a bit in this word. The high order bit corresponds to device X'D0', and the low bit to device X'EF'.
01A-01B	T	DISC.ICW	Portion of interrupt control word corresponding to disc devices.
020-07F			Reserved for bootloader code.
080-0BF	D (16)		Monitor register word addresses. These 16 double tallies contain the word addresses of monitor registers when the monitor is inactive.
0C0-0CF	H (16)		System locks
0C0		OVRFLW*	
0C1		GLOCK*	
0C2		SPOOLER*	Spooler queue
0C3		MSG*	
0C4		LOGON*	
0C5		TAPE*	Tape attachment and detachment
0C6		CREATEFILE*	
0C7		DEVICE*	Spooler device table
0C8		ASSIGN*	Spooler assignment table
0D0-0D1	T	DSCERRS	Count of the number of disc errors since the last bootload.
0D5	H	SPLINE	Spooler's line number
0DC-0DD	T	NCOMM	Number of communication lines
0DE-0E1	D	SYSTEMBASE	First FID of low disc space

TABLE A-3 (Cont'd) LAYOUT OF MTABLES0

0E2-0E3	T	PCB0	Frame number of port zero's PCB
0E4-0E5	T	CORESIZE	Number of K bytes of memory
0E6-0E7	T	ABSFRAMES	Number of ABS frames
0E8	H	MAXDISC	The highest disc device address present on the system
0E9	H	NDISCS	Number of disc drives
0EA	H	PHCOUNT	
0EB	H	PHLINE	
0EC-0EF	D	MAXFID	Maximum frame-ID on the system
0F0-0F3	C (4)		System release level
0F4-0F7	D	FFSEARCH#	Number of buffer queue entries to search for an available buffer before forcing a disc write
0F8-0FB	D	FARWA	This is used by the FAR instruction
100-17F	R (16)		Monitor register displacement, flag, and FID fields
180-181	T	RSEND	Points one byte past the last possible return stack entry
182-183	T	RSCWA	Points to the next available entry in the return stack
184-1AF	D (11)		Monitor return stack
1B0-1B3	D	SAVER10	Location where monitor register 10 is saved by the power fail/restart software
1B4-1BF			Miscellaneous constants
1C0-1C3	D	SPOOLPIB	PIB address of the spooler
1C4-1C7	D	AWFID	Automatic write FID. Periodically the monitor tests to see if this frame is in main memory and write required. If so, it is written to disc, regardless of its position in the BQ. Typically this is used to write the overflow table to disc.
1C8-1DF			Miscellaneous constants
1E0-1EF			Bootloader tape task descriptor
1F0-1F1			Miscellaneous constants
1F2-1F7	F	FPY	Six-byte accumulator extension
1F8-1F9			Miscellaneous constants
1FA-1FB	T	DRCT.BASE	Base of the table of disc read counters for each drive.
1FC-1FD			Miscellaneous constants
1FE-1FF	T	DCFT.BASE	Base of the disc configuration table

TABLE A-4

LAYOUT OF MTABLES1

Location	Type	Name	Description
200-201	T	HATBASE	Upper two bytes of the beginning address of the HAT.
202-203	T	HLTBASE	Upper two bytes of the beginning address of the HLT.
210-21F	H (16)		Disc configuration table. The nth byte is the actual device address used for disc n.
230-23F			Miscellaneous constants
240-24F			Reserved for the IOP mailbox and task descriptor. See PS20032013 for details.
250-24F	D (16)		Disc read counter table. Each four bytes contain the disc read count for the corresponding drive.
380-3FF	D (32)		Monitor register power-fail save table. The contents of the monitor ARFILE and the MR fields are saved here during a power fail.

TABLE A-5

LAYOUT OF MTABLES2

Location	Type	Name	Description
400-43F	16B (4)		Tape I/O mailboxes
400	H	TOMAIL	Tape unit zero mailbox control byte.
401-403	3B	TOFIRST	Main memory location of the first task descriptor on the chain for tape unit zero.
405-407	3B	TOCURRENT	Main memory location of the task descriptor of the current (executing) task for tape unit zero.
409-40B	3B	TOLAST	Main memory location of the last task descriptor on the chain for tape unit zero.
40C-40F	D	Unused	
440-47F	16B (4)		I/O mailboxes for printers. The format is identical to that of the tape mailboxes.
480-4FF	16B (8)		Reserved for mailboxes for future devices whose addresses are in the range X'E8 - X'EF'.
500-53F	D (16)		Disc mailboxes
500	H	DOFLGS	Mailbox control byte for disc unit zero.
501	H	D0FIRST	PIB number of the first PIB on the task descriptor chain for disc unit zero.
502	H	DOCURRENT	P13 number of the current (executing) task for disc unit zero.
503	H	DOLAST	Number of the last PIB on the task descriptor chain for disc unit zero.
580-5BF			Firmware debugger work area.
5CO-5CF	H (16)		Status byte for each device in the range X'DO' - X'DF'. These bytes are supplied by the DMP to the configurator as a means of sensing the number of terminals present.
5DO	H		Highest available device address in the range X'00' - X'7F'. This byte is suppled by the DMP to the configurator as a means of sensing the number of terminals present.
5D1-5DF			Reserved for further configuration information.
5EO-5EF			Tape task descriptor used by the bootloader firmware.
5FO-5F3	D		Memory error logger word
5F4-5F7	D		Power-fail/restart control word for DMP.
5FA-5FB	T	DATE	System date (internal format). The location of this field is fixed by the 24-hour interrupt firmware.
5FC-5FF	D	TIME	System time in deciseconds. This location is fixed by the DMP firmware.

TABLE A-6

LAYOUT OF PIB

Byte	Bit	Name	Meaning
0	0	ACTIVE	This bit is set when the process may be activated despite the presence of other roadblocks.
	1	SLEEP/	Zeroed when the process is asleep.
	2	DIOBLK/	Zeroed when disc I/O is in progress for the process.
	3	FAKERD/	Zero when a write initiated by a fake read is in progress.
	4	OBYTEBLK/	Zeroed while terminal output is in progress for the process. The process may be activated if ACTIVE is set.
	5	IBYTEBLK/	Zeroed while terminal input is in progress. The process may be activated if ACTIVE is set.
	6	RSTREAD	Set when the break key terminates a READ instruction. This allows the debugger to set the program counter back to the READ instruction.
	7	PHANTOM	When set, terminal I/O traps to the software debugger for a TIPH process.
1	0	ECHO/	Set to not echo terminal input
	1	INDEBUG	Set when executing from the debugger control block.
	2	PRGERR	Set by the firmware on an abort condition. This also serves to inhibit the break key when the process is on its way to the debugger. The ZPI instruction zeroes it.
	3-7		Error number. Note that this field is expanded to 5 bits from REALITY's 4 bits.
2	0-7	NEXTPIB	Number of the next process in the SNU queue. A value of X'FE' indicates that the process is out of the queue. A value of X'FF' indicates that this process is the last in the queue.
3	0-7	PREVPIB	Number of the previous process in the SNU queue. A value of X'FE' indicates that the process is out of the queue. A value of X'FF' indicates that this process is the first in the queue.
4	0-7 0 1 2	DCMD	Disc command byte Set for read commands Set for write commands Set for control commands
5	0-7	IRPTCONT	Disc interrupt control byte
	4	D.ISYN3	Set when the controller should interrupt a device to be specified.

TABLE A-6 (Cont'd) LAYOUT OF PIB

	5	D.ISYN2	Set when the controller should interrupt the DMP.
	6	D.ISYN1	Set when the controller should interrupt the CPU.
	7	D.ENABLE	Set to enable interrupts.
6-7	0-15	PIBBUFF	Upper two types of the starting main memory address for the disc transfer.
4-7	0-31		This word may be used by the power restart software temporarily to store PIB bytes 2C-2F.
8	0-7	SECCNT	Number of sectors to transfer.
9-B	0-23	RSN	Relative sector number of the frame to be transferred. This number is relative to absolute sector zero on the given disc.
8-B	0-31		This word may be used by the power restart software temporarily to store PIB bytes 24-27.
C	0-7	MAJOR	Major status
	0		Busy bit
	1		Normal end of transfer
	2-7		If any of these bits are set, an error occurred. See PS20032011 for further details.
D	0-7	MINOR	Minor (error) status. See PS20032011 for a breakdown of the individual bits.
C-D	0-16		This word may be used by the power restart software temporarily to store PIB bytes 28-29.
E	0-7	DDIAG	Diagnostic disc status byte. See PS20032011 for a breakdown of the individual bits.
F	0-7	NXTDQ	PIB number of the next PIB on the task descriptor chain for this disc.
10	5		Read-after-write when using the disc diagnostic embedded in the monitor.
	6	BATCH	Set if this is a batch job.
	7	READBIT	Flag for moving the TERM.LEN field from the PIB to TO. This is set upon receipt of an interrupt that terminated a read and cleared by search buffer pool once it has attached to the PCB and moved TERM.LEN to TO.
11-13	0-23	FFAULT.FID	FID which caused the frame fault.
14-17	0-31	SLEEPTIME	Time to wake up the process when it is asleep. This field is also used to store the FID being written when a frame fault requires a write.
18-1B	0-31	DREADCTR	Counter of number of user's disc reads.

TABLE A-6 (Cont'd) LAYOUT OF PIB

1C	0-7	LOCKN	Number of the system lock that this user has set.
1D			Unassigned and reserved
1E-1F	0-15	PCB.BIF	Upper two bytes of the main memory address of the user's PCB buffer. This field is set up when the process is activated.
20	0-7	MCAL.LIT	Literal field from an MCAL instruction. When a disc error occurs this field is used to store the disc command that resulted in the error. The debugger should look at this field, rather than at DCMD, which may contain a correcting command.
21-23	0-23	MCAL.BUF	Main memory address referenced by the register in an MCAL instruction.
21-23	0-23	ERR.STAT	Drive, major and minor status when a disc error occurred.
24	0-7	TERM.CMD	Terminal command byte
25-27	0-23	TERM.ADDR	Beginning main memory address of the data buffer before the operation. Last byte transferred to the buffer after the operation.
28-29	0-15	TERM.LEN	Maximum number of bytes to read or write. After the terminal operation is complete, the ACLC decrements this field by the number of bytes transferred.
2A-2B	0-15	TERM.STATUS	Status word of the completed operation
2C-2E	0-7 8-15 16-23	TSC1 TSC2 TSC3	Terminal I/O delimiters
2F	0-7		Reserved for use by the firmware; if a BREAK key is hit while no I/O is occurring, the ACLC sets the low-order bit of this byte and interrupts the CPU. When the CPU acknowledges the interrupt, it will zero the byte.
30	0-7		Used by the WRITE instruction when only one byte is output.
30-39			Available. These bytes are used as an I/O buffer for logon.
3A	0-7		Terminal backspace character
3B	0-7	DDA	Disc device address
3C-3F	D	DEACT	Number of times the process has been deactivated since logging on.

TABLE A-6 (Cont'd) LAYOUT OF PIB

40-41	0-15	PCISW	Programmable communication interface switch settings for the ACLC. These control baud rate, character length, etc.
42-43	0-15	PIB.TIMESLICE	User's timeslice in half-milliseconds. The DMP firmware restricts this time to less than 1024 milliseconds.
44-45			Unassigned and reserved
46-47	0-15		Buffer number used by disc I/O verification software.
48-4B	0-31		Buffer address used by disc I/O verification software.
4C-4F	0-31		Used to save DCMD and PIBBUFF during disc verification.
50-7F			Unassigned and reserved

Common PIB Status Values

PIB bytes 0, 1 and X'10' must all be displayed to determine the status of a process. However, the following hexadecimal values in byte 0 are usual (the values in parenthese will be seen when bit seven is set for a TIPH process):

Byte 0	Meaning
3C (3D)	Asleep until time in PIB bytes X'14' - X'17" or BREAK key
4C (4D)	Fake frame fault (doing a write to make a buffer available for a fake read).
5C (5D)	Frame fault, waiting for a frame to be brought into main memory from disc
74 (75)	Terminal I/O output roadblocked
78	Terminal I/O input roadblocked
7C (7D)	Can be activated or is active
F4	Can be activated or is active while characters are being transferred to terminal.

Priority Queue

The priority queue is a doubly-linked list of PIBs used by the monitor to select the next virtual process to run. Because the monitor used the select next user (SNU instruction to select the next process, the priority queue is often called the SNU queue. Location X'14' points to the top of the queue; location X'15, to the bottom. The pointers are the numbers of the processes, or PIBs, called PIB links.

Identifying the PIB

A PIB link is the number of the process that corresponds to the PIB. The PIB link is used to link the PIBs in the priority queue and in disc I/O task descriptor chains.

When the firmware debugger is available, a PIB link can be converted to the address of its PIB by using the "PD" or "PX" commands. Otherwise, the address can be calculated by multiplying the PIB link by X'80' and adding the result to X'1400'. For example, the PIB address of the port 8 process is

```
8 * X'80' + X'1400' = X;1800'
```

What Is Loaded By Bootload

When the DMP signals the CPU that a bootload has been requested, the CPU firmware loads the first two tape records into main memory:

Mode	Location
MTABLES0	0
MBOOT	X'600'

Software instruction execution starts at location X'20'. Control transfers to MBOOT, which display the options message. After the operator replies with an option, MBOOT loads the following:

Mode	Location
MTABLES1	X'200'
MSETUP1	X'800'
MSETUP2	X'A00'
MSETUP3	X'C00'
MSETUP4	X'E00'

MTABLES2 is not loaded because it contains fields that should not be destroyed during a warmstart; for example, the I/O mailboxes and the system time. Control then transfers into MSETUP1 to begin the configuration.

Initialization Done by the Configurator

The configurator determines the system configuration by using the information supplied by the DMP in locations X'5C0' - X'5CF' and X'5D0' and by asking for information of the operator. No tables are configured, however, until after the operator replies "Y" to the message "CONFIGURATION CORRECT?". After an affirmative reply to that question the configurator:

Initializes the buffer management tables,
Initializes the PIBs,
Reads in the coldfids,
Reads in the monitor, and
Executes the first virtual process.

To initialize the buffer management tables the configurator sets all buffers, except buffer zero, to empty and available. Then the configurator, using the FAR instruction, corelocks the system buffers listed in the section titled "Reserved Buffers." Buffers containing PIBs are also set write-required. The BSAFT is set with a FID of X'FFFFFC' for the buffer containing PIBs 0-3, X'FFFFFB' for the buffer with PIBs 4-7, X'FFFFFA' for the buffer with PIBs 8-11, and so forth, the absolute value decreasing by one for each buffer.

PIB initialization begins by zeroing all PIB buffers. Each PIB is then initialized in turn:

Byte	Initialization
0	Status is set to input roadblocked (X'78'). PIB0 status is set to can-be-activated (X'7C').
1	Break inhibit bit is set.
2	Forward link is set to not in priority queue (X'FE'). PIB0 link is set to queue bottom (X'FF').
3	Backward link is set to "Not in priority queue (X'FE')." PIB0 is set to queue top (X'FF').
F	Task descriptor link is set to not in the task descriptor list (X'FF').
1C	Lock number is set to indicate no system locks (-1)..
1D	Time slice is set to X'14'.
25-27	Terminal input address is set to PIB address plus X'30'.
28-29	Length of terminal input is set to 1.

The priority queue is set to point to PIB0; that is, locations X'14' and X'15' are both set to zero. The coldfids, modes necessary for starting the system, are loaded from tape into buffers which are then set write-required. The frames are written to disc in the course of the monitor's normal operation.

Control transfers to MBOOT to load the monitor modes as follows:

Mode	Location
MONITOR 2	X'800'
MONITOR 3	X'A00
MONITOR 4	X'C00'
MONITOR 5	X'E00'

Control then transfers to MTABLES0, which loads MONITOR1 into location X'600'. If a coldstart is being performed, execution transfers to the monitor's select-next-user routine (See BLOCK 020 in section titled "Overview of Monitor's Logic"); if warmstart had been selected, control branches to a clean-up

TABLE A-7

MONITOR REGISTER ASSIGNMENTS

Register Number	Permanent Value	Use
0	000	Permanent pointer to MTABLES0. This register value must never be changed.
1	400	Pointer to MTABLES2
2	800	Pointer to MONITOR2
3	A00	Pointer to MONITOR3.
4		Address of the current PIB. This register is accessed by the firmware in several instructions and must always be used as the PIB pointer.
5	100	Second half of MTABLES0
6	160	Pointer to monitor register 12
7	C00	Pointer to MONITOR4.
8	Z00	Pointer to MTABLES1.
9		Unassigned, but reserved for future use.
10		Scratch. This points to the ABS frame being executed when the system is running virtual monitor code.
11		Scratch
12		Scratch
13		Scratch. Several common uses are as a pointer to the user's PCB, a pointer to disc mailboxes, and the FAR instruction's BQ search register.
14		Scratch. Common uses are as the FAR instruction's BQ search register and a pointer to the current disc I/O buffer.
15		Scratch. This register is set up by the IO and IONB instructions to point to the task descriptor.

TABLE A-8

MONITOR ENTRY POINTS

Entry Point	Address	Use
0	601	Entered by firmware when a virtual process frame faults
1	603	Reserved for entry when an external interrupt is received. This entry is not presently implemented.
2	605	Reserved for entry when an internal interrupt is received. This entry is not presently implemented.
3	607	Firmware release quantum. The monitor is entered here upon any of the following conditions; attempted attachment to an I/O busy buffer. Attempted execution of a READ or WRITE instruction when OBYTEBLK/ was set, execution of a WRITE instruction that crosses a frame boundary, execution of the LOCK instruction when the lock was already set, or timeslice runout.
4	609	Reserved for entry when a monitor I/O instruction cannot be executed because of an inconsistency in a chain.
5	60B	Initial entry point when a power restart occurs.
6	60D	Used internally by the monitor
7	60F	Entered from a virtual READ instruction. By using this entry point instead of 607, the firmware prevents the PIB from being inserted into the links.
8	611	IO and IONB instructions (MCAL)
9	613	RQM instruction (MCAL)
A	615	MCAL for time and date
B	617	Common MCALs

procedure for restarting I/O operations, which transfers to the select-next-user routine.

Since the PIB for port zero is the only one in the priority queue, it is the one activated. Three coldfids that were loaded are PCB and DCB (control blocks for port zero) and ABSL1 (the ABS loader). Register one of PCB0 is preset pointing to frame 47, entry 1, which is in ABSL1, so the ABS loader is started. It initializes the remainder of port zero's work space and poses the "OTHER OPTIONS?" question.

MONITOR

The monitor register assignments are shown in Table 3-7. Monitor entry points are given in Table A-8. All entry points to the monitor are entered directly by the firmware.

Overview of Monitor's Logic

In the discussion of the monitor's logic, ENTRY POINT n refers to the monitor's entry point n; BLOCK n refers to a block of logic within this discussion. The monitor functions as follows:

ENTRY POINT 0: This point is entered by the firmware when a virtual process frame faults. The process is moved to the bottom of the priority queue. The monitor searches for an available buffer for the frame to be read into. If a buffer is available, all necessary information for the disc read is placed in the process's PIB, and the PIB is linked to the end of the task descriptor chain for the appropriate disc. The buffer map is changed immediately to show the buffer as I/O-busy and containing the new FID. This puts any other process that might frame fault on the same frame to sleep rather than scheduling a duplicate disc read.

If there is no available buffer, the entire buffer queue is searched for a buffer that is write-required. If there is one, its disc address is computed and placed in the virtual process's PIB (bytes 9-X'B'). Then the PIB is linked to the task descriptor chain for the appropriate drive. The read is started by the disc-interrupt handling logic. If no write-required buffer can be found, the process will be activated at the bottom of the priority queue, at which time this entry point will be entered again. Control transfers to the logic described in BLOCK 020.

ENTRY POINT 3: The firmware enters this entry point to release the current process's time quantum. Control transfers to the logic at BLOCK 010.

ENTRY POINT 5: This entry point is entered when there is a power restart. The I/O task descriptors for disc, tape, printer and terminals are cleared up and the tasks restarted. Control returns to the entry on the top of the return stack, which has been set by the firmware to entry point 7, if a virtual process was interrupted by the power fail, or to the point of interruption, if the monitor was interrupted.

ENTRY POINT 7: This entry is used by the firmware for a READ instruction and as a return point when a virtual process is interrupted by a power fail. Control transfers to the logic described in BLOCK 020.

ENTRY POINT 8: The IO and IONB instructions (MCALs) enter here. The requested I/O is started and execution continues at BLOCK 020.

ENTRY POINTS 9, A, B: These are the entry points for the other MCAL instructions. They perform their specified tasks and then transfer control to BLOCK 010.

BLOCK 010: The process is moved to the bottom of the priority queue. Control falls through to BLOCK 020.

BLOCK 020: If there are any disc interrupts, the monitor performs the logic described in BLOCK 100. The priority queue is checked for a process ready to run. If there is one, the monitor exits to the virtual process. If there are no processes ready to run and no disc reads are being performed, the monitor tries to write a write-required buffer to disc as discussed in BLOCK 110. When that procedure has finished its operation, control is transferred to the logic at the top of BLOCK 020.

BLOCK 100: Since the firmware acknowledges interrupts from devices and sets a flag in main memory without stopping current processing, the monitor looks at the flags every time it is given control. A successful virtual disc read causes the process to be placed on top of the priority queue unless the batch bit (PIB byte X'10', bit 6) is set, in which case the process is put on the queue bottom.

When a virtual write completes its operation without an error, the buffer is used to start a read to satisfy the frame fault. Errors on virtual reads and writes cause the process to be sent to the software debugger. If a write was

taking place, the buffer is reset to write-required. Eventually, the operation will be retried when either the process re-executes the frame fault (read) or some process tries to write the buffer.

When a monitor write finishes, the monitor resets I/O-busy, frees the monitor PIB by resetting the disc-roadblock bit, and records any error status. Control is transferred to ENTRY POINT 7.

BLOCK 110: There are eight PIBs reserved for the monitor's use. Whenever a new user cannot be selected and no disc reads are being performed, the monitor sees if one of its own PIBs is not disc roadblocked. If there is such a PIB, the buffer queue is searched for a write-required buffer. If one is found, a monitor PIB is set up with a write command and added to the end of the task descriptor chain for the appropriate disc.

In the process of looking for a write-required buffer if disc interrupts are reported, control transfers to BLOCK 100; otherwise, execution continues at BLOCK 020.

Input/Output

SEQUEL's architecture is based on two busses. One bus, the I/O interface bus (or SEQUEL bus), has 32 data lines and 24 address lines. It connects the CPU, the diagnostic maintenance process (DMP), the memory controller, all memory arrays, the magnetic tape/printer (MTP) controller, and I/O processor (IOP). The other bus, the IOP bus, connects the IOP to the disc controllers and the asynchronous communications line controllers (ACLC), which control the terminals.

All external device controllers on SEQUEL use direct memory access (DMA) to accomplish data transfers. The controllers use command blocks called task descriptors. The system sets up a task descriptor in main memory with all the information necessary for an I/O operation and issues one instruction, start I/O. The controller activated by the start I/O instruction reads the task descriptor from memory and executes the task. Task descriptors for disc and terminals are reserved fields in the PIBs. Tape and printer task descriptors may be anywhere in main memory.

Since a terminal's device address is the same as its associated process's number, the ACLC uses the device address from the start I/O instruction as an index into the corelocked PIBs to find the task descriptor.

Disc, tape and printer controllers support command chaining. It is possible to link several task descriptors and have the controller execute a task and then automatically continue to the next task. Chaining is accomplished using mailboxes. Each of

the mailboxes contains pointers to the first task descriptor, the task descriptor being executed, and the last task descriptor in the chain. One mailbox is allocated to each device. A mailbox contains a control byte for synchronizing access to the chain. Disc mailboxes use PIB numbers to chain together task descriptors. Tape and printer mailboxes use main memory addresses to link their task descriptors together.

When an interrupt occurs, the firmware sets a bit in main memory indicating that a particular device has interrupted. The currently active process is not deactivated. The double word at location X'1A' - X'1D' is used to flag interrupts for devices in the address range X'D0 - X'DF' and the least significant sixteen bits to devices X'E0' - X'EF'. Whenever the monitor is activated, it examines the interrupt flag word and services the descriptor chains of any devices that have generated an interrupt since the last time the monitor was active.

Mailbox Control Byte

The bits of a mailbox control byte are listed and described in TABLE A-9.

TABLE A-9

MAILBOX CONTROL BYTES

Bit	Name	Description
0	LOCK	Used to prohibit simultaneous access to the mailbox by the CPU and the controller.
2	QUEUED	Set by the CPU when it is queueing up a task descriptor, and it is cleared by the controller. Since a start I/O instruction signals the controller to execute the chain, this bit's state is not significant.
3	ATTENTION	Set by the controller when it has completed a task; the CPU clears it after servicing the chain. Since this bit indicates the completion of only one task, it could be set at the same time as either BUSY or ABORTED.
4	BUSY	Indicates that the controller is executing the tasks on the chain; the controller clears this bit when it has executed all tasks on the chain.
5	ABORTED	Set by the controller when it has to abort a task, which halts processing of the chain. Hence, both ABORTED and BUSY should not be set at the same time.

Terminal I/O

SEQUEL uses ACLC-monitored DMA transfers to move data between main memory and a terminal. The ACLC transfers data in 256-byte segments.

For input the ACLC is given a byte count and some delimiter characters that may halt the operation early. Since the controller performs the backspace, break, echo, cancel and retype operations on each 256-byte segment, inputting these characters may yield erroneous results if they are entered during a long read. PIB byte X'2C' contains a mask for the active delimiters. Each bit tells the ACLC that it should terminate the input if the corresponding character is entered. Active delimiters are not echoed. The ACLC terminates input either when the maximum byte count runs out or when a delimiter is entered.

In practice, a 140-byte buffer, called the IB workspace, is used for most input. Some routines, such as logon and entering a character at the end of a page, use a small buffer in the PIB (X'30 - X'39'). Terminal output in most cases also uses a 140-byte buffer (OB). Screenpro, however, outputs character strings that may cross linked frame boundaries.

Terminal I/O is started, generally, with a READ or a WRITE instruction. The firmware sets the PIB roadblock in the PIB status in PIB byte 0. Bit 4 is zeroed for output; bit 5 for input. The buffer status is set to I/O busy. For an input operation, the PIB is removed from the priority queue.

Status bit 0 (ACTIVE) is used to keep a process running as long as possible during output to the terminal. Since output is done on strings, status bit 0 (ACTIVE) is set when output is requested. This allows the process to continue running. If the process attempts to perform another terminal I/O operation or if it references the buffer in which the data resides, the ACTIVE bit will be turned off, and the process will be roadblocked until the I/O operation is complete.

When an interrupt occurs, the firmware looks at the PIB status byte. If bit 4 is zero, indicating that the process was outputting, the firmware sets the bit, clearing the roadblock. If the PIB is not output roadblocked, the input roadblock (bit 5) is checked. If it is zero, it is set. In either case the buffer status is reset from I/O busy, and for input the PIB is put into the priority queue.

Note that numerous parts of the system software (e.g., the power restart routine) assume that when a process is output roadblocked, it is in the priority queue. On the other hand, if a process is input roadblocked, the system software (the LOGON verb, for instance) assumes that there is no way to activate the process.

Terminal errors cause control to transfer either to software debugger entry point 10 if the BREAK key is pressed or to entry point 12 if any other terminal error occurs.

SEQUEL's monitor, with certain restrictions, uses the standard terminal I/O instructions (READ and WRITE). Port zero's PIB is borrowed by the firmware to accomplish the monitor's terminal I/O. Virtual processes are not running while the monitor is doing terminal I/O. A bit called MONREAD, location X'11', bit 3, is set when the monitor starts a terminal operation, and it is cleared when the end-of-transfer interrupt signaling occurs.

Task Descriptors

The task descriptor for each process is in its PIB, starting at byte X'24'. The task descriptors format is shown in Table A-10.

Key Locations

In addition to the task descriptors, the locations shown in Table A-11 are also significant in terminal I/O.

TAPE AND PRINTER I/O

To begin a tape or line printer I/O operation, a virtual process prepares an I/O buffer and task descriptor. The task descriptor must be placed in main memory locations that will not be paged out. The process then executes an IO or IONB instruction with the operand registers pointing at the first byte of the I/O area and the first byte of the task descriptor.

To minimize software changes, REALITY's status-driven scheme has been copied in SEQUEL's tape and printer I/O operation. Tape and line printer interrupts are disabled so that the monitor is not interrupted. To obtain the results of an I/O operation the virtual process must use one of the STATUS instructions to read the status from the task descriptor. Instruction STATUSW waits until the operation is complete. STATUS returns immediately so that the software may perform other functions and re-execute STATUS at a later time. While bit 28 of the status is off, the I/O operation is not complete. Once status bit 28 is set on, the virtual process may execute the delete top task from chain (DTP) instruction to clear the finished entry from the task descriptor chain and remove the I/O busy status of the memory buffer.

Physical errors (tape parity errors, printer off-line) are handled by the virtual process that initiated the I/O. When the controller detects an error, it suspends processing of the remainder of the chain so that the software can modify the chain as necessary and restart processing.

Table A-10

TASK DESCRIPTOR'S FORMAT

Pib Byte	Bit	Description
24		Terminal command byte. See PS20032014 for the commands.
25-27		Beginning main memory address of the data buffer before the operation. After the operation the address of the last byte transferred to or from the buffer.
28-29		Maximum number of bytes to read or write. After the operation is complete, ACLC decrements this field by the number of bytes transferred.
2A-2B		Status of the completed operation. See PS20032014.
2C		Terminal I/O delimiters:
	0	Unassigned
	1	Stop input on Control-I (Tab)
	2	Stop input on ESCape
	3	Stop input on Control-P
	4	Stop input on Control-N
	5	Stop input on either carriage return or line feed
	6	Stop input on the character in PIB byte X'2E'.
	7	Stop input on the character in PIB byte X'2D'.
2D		Delimiter character specified by byte X'2C', bit 7.
2E		Delimiter character specified by byte X'2C', bit 6.
2F		Reserved for the firmware; if a BREAK key is hit while no I/O is occurring, the ACLC sets the low-order bit of this byte and interrupts the CPU. When it acknowledges the interrupt, the CPU zeros the byte.

TABLE A-11

KEY LOCATIONS

Location	Bit	Description
11	3	Set when the monitor does terminal I/O
DC-DD		Number of communications lines
5D0		Highest available device address in the range X'00' - X'7F'. This byte is supplied by the DMP to the configurator as a means of sensing the number of terminals present.
PIB Byte	Bit	Description
0		PIB status:
	0	This bit is set when the process may be activated despite the presence of roadblocks.
	4	Zeroed while terminal output is in progress for the process. The process may be activated if bit 0 is set.
	5	Zeroed while terminal input is in progress. The process may be activated if bit 0 is set.
	6	Set when the break key terminates a READ instruction. This allows the software debugger to set the program counter back to the READ instruction.
	7	When set, terminal I/O traps to the software debugger for a TIPH process.
1	0	Set to not echo terminal input.
1	2	Set by the firmware on an abort condition. This also serves to inhibit the break key while the process is on its way to the debugger.
10	7	Flag for moving the byte count in PIB bytes X'28' - X'29' to T0. This is set upon receipt of an interrupt that terminates a read; it is cleared by the search buffer pool logic after it has attached to the PCB and moved the byte count to T0.
30		Used by the WRITE instruction when only one byte is output.
30-39		These bytes are used as an I/O buffer for logon.
3A		Terminal backspace character.
40-41		Programmable communications interface switch settings.

Mailboxes

The tape mailboxes are located in X'400' - X'43F'. The printer mailboxes are in locations X'440' - X'47F'. The mailboxes, each requiring 16 bytes, are ordered according to device addresses. To calculate the address of a mailbox, multiply the low-order nibble of the device address by 16 (X'10') and add the product to X'400'. For example, the mailbox for device X'E5', a printer, is at X'5' * X'5' * X'10' + X'500 = X'450'.

The mailbox for tape device X'E0' at location X'400' displays the formal for all tapes and printers:

Location	Bit	Description
400		Control byte:
	0	Lock
	2	Queued
	3	Attention
	4	Busy
	5	Aborted
401-403		Main memory location of the first task descriptor on the chain for this device.
404		Unused
405-407		Main memory location of the task descriptor of the currently executing or aborted task for this device.
408		Unused
409-40B		Main memory location of the last task descriptor on the chain for this device.
40C-40F		Unused

Task Descriptors

The task descriptors for tape and printers may be anywhere in main memory, but they must begin on a double word boundary; that is, the address must be divisible by four. During I/O execution, however, the buffer in which a task descriptor resides must be corelocked. A task descriptor's format is as follows:

Byte	Bit	Description
0		Command. See PS20032012
1		Interrupt control. See PS20032012
2-3		Byte count: the number of bytes to transfer
4		Not used
5-7		Main memory address of data
8-11		Status. See PS20032012
12		Not used
13-15		Main memory address of next task descriptor or -1

Tape and Printer I/O Key Locations

In addition to the mailboxes and task descriptors, the following locations are also significant in tape and printer I/O:

Location	Bit	Description
01C		Portion of interrupt control word corresponding to tape and printer devices
0C2		System lock for spooler queue
0C7		System lock for spooler device table
0C8		System lock for spooler assignment table
0D5		Spooler's line number
1C0-1C3		PIB address of the spooler
5E0-5EF		Tape task descriptor used by the bootloader firmware and software

Disc I/O

When a frame fault occurs, the firmware deactivates the process, sets the PIB disc-roadblocked, loads the absent frame's FID into the PIB and into the monitor's D0, and enters the monitor with R4 pointing to the PIB. The monitor sets up the PIB with all the necessary information for the disc transfer and executes a link and start disc I/O (LSDIO) instruction. The LSDIO locks the mailbox, inserts the PIB at the bottom of the chain, unlocks the mailbox, and may issue a start I/O instruction, depending on the state of the descriptor chain. Before the monitor selects the next process to be activated with the SNU instruction, it processes all recorded disc interrupts since the last activation of the monitor.

Monitor disc I/O is supported by eight additional PIBs located after the virtual process PIBs in main memory. The buffers beginning a location X'5400' are reserved for monitor PIBs.

Physical errors, like a disc dropping ready, cause the disc controller to stop processing the chain. When the monitor is activated, it tests for disc errors in the completed task descriptor chain, notes any error, and activates the software debugger instead of the process that received the error. The debugger outputs the ampersand disc-error indicator and re-executes the instruction that generated the original frame fault. Where possible, the monitor issues commands that attempt to correct error conditions. The monitor finally issues a command to the controller to resume execution of the chain.

Disc Mailboxes

The disc mailboxes are in locations X'500' - X'53F'. The mailboxes, each requiring four halfwords, are ordered according to disc device addresses. To calculate the address of a mailbox, multiply the low-order nibble of the disc's device address by four and add the product to X'500' (e.g., the mailbox for X'DB' is at X'B' * 4 + X'500' = X'52C').

The mailbox for disc unit X'D0' at location X'500' is representative of all the disc mailboxes:

Location	Bit	Description
500		Control byte:
	0	Lock
	2	Queued
	3	Attention
	4	Busy
	5	Aborted
501		PIB number of the first PIB on this drive's task descriptor chain
502		PIB number of the executing or aborted task descriptor on this drive's chain
503		PIB number of the last PIB on this drive's task descriptor chain

Disc I/O Task Descriptors

The task descriptor for each process is in its PIB starting at byte four. Note that bytes 4-X'D' of the PIB are used by the power-restart logic as a save area so these bytes may not contain disc-related data after a power restart. A task descriptor's format is given in Table A-12.

Disc I/O Key Locations

In addition to the mailboxes and task descriptors, the locations shown in Table A-13 are also significant in disc I/O.

TABLE A-12

DISC I/O TASK DESCRIPTOR'S FORMAT

PIB Byte	Bit	Description
4		Disc command byte (see also byte X'20'), See PS20032011 for commands, but typical values are: X'80 - read X'40' - write X'22' - recalibrate
5		Disc interrupt control byte (should always contain X'03'):
	4	Set when the controller should interrupt a device to be specified.
	5	Set when the controller should interrupt the DMP.
	6	Set when the controller should interrupt the CPU.
	7	Set to enable interrupts
6-7		Buffer number of the main memory address for the disc data transfer.
8		Number of sectors to transfer
9-B		Relative sector number of the frame to be transferred. This number is relative to absolute sector zero on the given disc.
C		Major status (see also bytes X'21' - X'23'):
	0	Busy bit
	1	Normal end of transfer
	2-7	If any of these bits are set, an error occurred. See PS20032011 for further details.
D		Minor (error) status. See PS20032011 for a breakdown of the individual bits.
E		Drive status byte. See PS20032011 for a breakdown of the individual bits.
F		Number of the next PIB on the task descriptor chain for this disc.

TABLE A-13

DISC I/O KEY LOCATIONS

Location	Bit	Description
011	5	Set if disc errors are to be recorded.
01A-01B		Portion of interrupt control word corresponding to disc devices.
0D0-0D1		Number of disc errors since last bootload.
0E8		Highest disc device address in system.
0E9		Number of disc drives
1FA-1FB		Base of the table of disc read counters for each drive.
1FE-1FF		Base of the disc configuration table
210-21F		Disc configuration table. The nth byte is the actual device address used for disc n.
250-28F		Disc read counter table. Each four bytes contain the disc read count for the corresponding drive.
5C0-5CF		DMP status for each device in the range X'D0' - X'DF':
	0-3	Disc type: 0 = Reflex II A 1 = Reflex II B 2 = Reflex II C 3 = Reflex II D
	4-7	Status: 0 = Not present 1 = Present 2 = Present but defective
5400-57FF		Additional PIBs for monitor disc I/O.
PIB Byte	**Bit**	**Description**
0	2	Zero when disc I/O is in progress.
11-13		FID which caused the frame fault.
14-17		FID being written when a frame fault requires a write. This FID is required in case a disc error causes the write to be cancelled. Note: This area is also used for a sleep wake up time, which implies that a process cannot be doing disc I/O and be asleep at the same time.
18-1B		Count of user's disc reads.
20		When a disc error occurs, this field is used to store the disc command that resulted in the error. When debugging, look at this field rather than PIB byte 4, which may contain a correcting command. Note: This field is also used to hold the literal from a MCAL instruction.
21-23		Drive, major, and minor status when a disc error occurs. See PS20032011 for status information.
3B		Disc device address

TRAPS

Certain conditions occuring in virtual mode cause the system to trap to the interactive debugger. When a condition requiring a trap arises, the firmware sets bit 2 of PIB byte 1 and stores the trap number in byte 0 of the PCB. If there is an address register error, the firmware stores the register number in byte ACF of the process's PCB. The debugger moves this byte to ACFSAV in the PCB.

Each trap has an entry point in virtual storage frame one, which is entered by the firmware performing a branch and stack location instruction.

Most of the reasons for trapping are determined by the firmware. However, there are some reasons which are determined by the software. Two of these are disc errors and message processor requests. The PIB error number field (byte 1, bit 4-7) is used to communicate to the firmware the need and the reason for a trap. For disc errors the monitor places X'9' in the error number field. When one process has a message to send to another process, the sender puts a X'D' in the receiver's PIB error number field. Whenever the firmware senses that the error number field is non-zero, it moves the value to the process's PCB byte zero, sets the error number field to zero, and traps to the debugger.

When the debugger is first entered, it transfers control to the monitor, which sets the conditions necessary for the debugger to run. The monitor sets PIB byte 1, bit 2 to zero. After setting INDEBUG (bit 1 of byte 1) to one, the monitor deactivates the process that has been trapped. When that process is next activated, because INDEBUG is set, the firmware will use DCB, rather than PCB, as the control block for the process. This will automatically put the debugger in control. This condition, in which INDEBUG is set to one and DCB is the control block, is called the debug state.

List of Traps

The traps that cause the system to transfer control to the debugger are listed below with the messages that the debugger outputs to the terminal. Note that not all traps result in a message and abort condition. The firmware stores the error number in byte zero of the process's PCB before entering the debugger.

For traps that involve address register errors, the following message will also be returned.

REG = n

Where "n" is the decimal register number of the register causing the trap.

When a trap results in an abort, the following message will also be returned.

ABORT @ f.d

Where "f" is the decimal FID of the frame and "d" is the hexadecimal displacement of the location where the trap occurred. This corresponds to the location counter in the assembly listing of the corresponding program. For traps 10 and 15 only the "f.d" are output.

The entry address into the debugger can be determined by doubling the trap number and adding 1. For example, the entry for number 5 is 11 (X'B'). Note that the debugger has more than 16 entry points. This is possible because the firmware can enter a frame at any location. Normally a mode is entered by a BSL or ENT with a mode-id, which is limited to 16 entries. The error numbers in Table A-14 are hexadecimal.

HARDWARE CHECKLIST

Some problems are caused by hardware that is not properly connected. Here are some techniques for checking proper hardware installation.

Voltages

Measure all supply voltages. They must be within tolerances specified by the applicable product specification. Measure system clock.

P. C. Board

Ensure that all PCBs are located in the correct chassis zone, i.e., memory boards in memory zone, processor in ALU zone, etc. Check that all PCBs are physically seated in the backplane. Loose boards cause intermittent errors.

Device Address Conflicts

Ensure that there are no device address conflicts, e.g., all address switches set correctly.

Newly Shipped Systems

On newly shipped systems inspect PCBs for solder splashes, loose components, ICs, etc., no cold solder joints, correct EO and latest revision levels.

TABLE A-14

TRAPS AND DESCRIPTIONS

Error No.	Message	Description
0	ILLEGAL OPCODE ABORT @ f.d	An illegal (undefined operation has been encountered
1	RTN STACK EMPTY ABORT @ f.d	A RTN (return) instruction was executed when the return-stack was empty (stack pointer was at X'0184').
2	RTN STACK FULL ABORT @ f/.d	A BSL or BSL1 (subroutine call) instruction was executed when the return-stack was full (stack pointer was at X'01B0'); the return-stack has been reset to an "empty" condition before the trap.
3	REFERENCING FRAME ZERO: REG=n	The register number containing the reference has been placed into the arithmetic condition flag byte (ACF) before the trap. An address register has a FID of zero when not in monitor mode.
4	CROSSING FRAME LIMIT; REG = n ABORT @ f.d	An address register in the "unlinked" format 1) has been incremented or decremented off the boundary of a frame, or (2) has been used in a relative address computation that causes the generated relative address to cross a frame boundary. The number of the register containing the reference has been placed into the arithmetic condition flag byte (ACT) before the trap.
5	FORWARD LINK ZERO: REG = n ABORT @ f/d	An address register in the "linked" format has been incremented past the last frame in the linked frame set. The number of the register containing the reference has been placed into the arithmetic-condition-flag byte (ACF) before the trap.
6	BACKWARD LINK ZERO: REG = n ABORT @ f.d	An address register in the "linked" format has been decremented prior to the first frame in the linked frame set. The number of the register containing the reference has been placed into the arithmetic-condition-flag byte (ACF) before the trap.
7	PRIVILEGED OPCODE ABORT @ f.d	A privileged operation code has been found. This means that an instruction that is allowed in monitor mode only has been used in virtual mode; or an instruction that may be used only in virtual mode has been used in monitor mode.
8	REFERENCING ILLEGAL FRAME ABORT @ f.d	An address register has an FID that exceed the maximum value allowable in the current disc configuration
9	None	A disc error has occurred. The operation will be retried later.

TABLE A-14 (Cont'd) TRAPS AND DESCRIPTION

A	I f.d	The break key on the terminal was activated.
B	None	Formerly, this error meant that the return-stack pointers were in an illegal format. At present the firmware does not check for this error.
C	None	This is a parity or terminal overrun and is presently ignored by the software.
D	None	Some process is sending a message to another. This is not usually an error.
E	END PROCESS	Terminate the primary process and execute a go. Usually not an error.
F	B f.d	When a trace mode is set, the firmware causes traps to the debugger using this error number.
10	None	A divide by zero has occurred. The debugger sets the overflow bit in user's ACF and returns to user.
11	None	When a phantom process performs a READ instruction, the firmware enters the debugger at this entry.
12	UNNORM. S/R ABORT @ f.d	A storage register with an unnormalized displacement was referenced.
13	None	When a phantom process performs a WRITE instruction, the firmware enters the debugger at this entry.
14	I/O HANDLING ERROR ABORT @ f.d	An error was detected during an I/O operation Usually, the error is in the mailbox. For example, if a mailbox has both the busy bit and the queued bit set, it indicates an error because when a controller starts an I/O operation, it should clear the queued bit and set the busy bit. If the firmware is unable to lock a mailbox within a million attempts, it causes this abort. Also, note that referencing an illegal device causes this abort.

Ribbon Cables

Inspect all ribbon cables for proper connector seating, no wear on ribbon jacket, and connector alignment on ribbon.

System Grounds

Ensure that all system grounds are connected, e.g., cabinet to REFLEX drives and tape drive, and intercabinet grounding on multiple cabinet systems. Check that the +5V and ground connections, (the big black power bus ribbons from the power supply) are properly secured and tight on the backplane.

Terminal Connectors

Ensure that all terminal connectors on the I/O panel are screwed down tight. This is a must for correct system operation on port 0.

Socketized ICs

Ensure that all socketized ICs (e.g., ALU PROMs and 2901s, RNI PROMs, Z80s, UARTs) are properly seated. Ensure that no leads are bent under or barely touching contacts.

AC Power

Ensure that proper AC power is installed at site and that site is properly wired in accordance with national and local electrical codes.

REFLEX Disc Drives

Ensure that REFLEX drive(s) are properly formatted.

Tape Drives

Ensure that tape drive read/write heads are clean.

Expansion Cabinet

On an expansion cabinet system ensure that bus priority and INAD switches are set correctly. Remember that ACLCs are addressed 0 through X'F' in the expansion cabinet. With two IOPs in the system, the one that interfaces with the discs only must have the switch set to DC-only position.

Cooling Fans

Cooling fans must push air through chassis.

Power Supply Margin Switches

Ensure that power supply margin switches are centered and not in either high margin or low margin.

Foreplane Connectors

Check to see that the foreplane connectors J3, J4, J5 on ALU, RNI, and SPCL function do not have bent pins.

Minimum configuration

If things really get bad--e.g., everything you do does not work--drop down to minimum configuration: 1 memory board, 1 ACLC, 1 disc, 1 disc controller, 1 MTP.

IOP Device Address

When there is only one IOP, it is used for both the disc controller and ACLCs. Its address must be X'FB'. This address is selected by setting toggle switch S1 away from the DC position (i.e., away from the backplane).

When there are two IOP boards, one (address X'FB') is used only for the ACLCs. The other (address X'FC') is used for the disc controller only. Address X'FC' is specified by setting switch S1 to the "DC" position (i.e., towards the backplane.

Controller Addresses

Each controller on the SEQUEL bus must have a unique bus address and a unique interrupt address. The bus address and the interrupt address for a given controller need not be the same value. Associated with the bus address is a bus request. Associated with the interrupt address is an interrupt request.

The address, which is a four-bit value, determines the priority. Binary address 0000 has the highest priority; 1111, the lowest. The addresses are selected by setting the address select switches on the controller board except for the DMP, which is hardwired to bus address 1111 and bus request 0 (BSRQ0). Hence, no other controller can have bus address 1111. The DMP has neither an interrupt address nor an interrupt request.

The bus request switches are mutually exclusive; that is, only one may be off (or open or 1); the other three must be on (or closed or 0). Likewise, the interrupt request switches are mutually exclusive.

1. IOP Bus Address and Bus Request Switches

The IOP bus address and bus request switches (S3) are located on the board at position 16F. Switches 8 through 5 specify the bus address (high-order to low-order), and switches 4 through 1 specify the bus request. The bus request switch setting must correspond to the two high-order switches of the bus address (on=closed=0; off=open=1):

Switch	8	7	4	3	2	1	Bus request name
	on	on	on	on	on	off	BSRQ3
	on	off	on	on	off	on	BSRQ2
	off	on	on	off	on	on	BSRQ1
	off	off	off	on	on	on	BSRQ0

2. IOP Interrupt Address and Interrupt Request Switches

The IOP interrupt address and interrupt request switches (S2) are located on the board at position 17F. Switches 8 through 5 specify the interrupt address (high-order to low-order), and switches 4 through 1 specify the interrupt request. The interrupt request switch settings must correspond to the two high-order switches of the interrupt address (on=closed=0; off=open=1):

Switch	8	7	4	3	2	1	Interrupt request name
	on	on	on	on	on	off	INRQ3
	on	off	on	on	off	on	INRQ2
	off	on	on	off	on	on	INRQ1
	off	off	off	on	on	on	INRQ0

3. MTP Bus Address and Bus Request Switches

The MTP bus address and bus request switches (S1) are located on the board between columns A and B and between rows 20 and 21. Switches 8 through 5 specify the bus address (high-order to low-order), and switches 4 through 1 specify the bus request. The bus request switch settings must correspond to the two high-order switches of the bus address (on=closed=0; off=open=1):

Switch	8	7	4	3	2	1	Bus request name
	on	on	on	on	on	off	BSRQ3
	on	off	on	on	off	on	BSRQ2
	off	on	on	off	on	on	BSRQ1
	off	off	off	on	on	on	BSRQ0

4. MTP Interrupt Address and Interrupt Request Switches

The MTP interrupt address and interrupt request switches (S2) are on the board between columns A and B and between rows 21 and 22. Switches 8 through 5 specify the interrupt address (high-order to low-order), and switches 4 through 1 specify the interrupt request. The interrupt request switch settings must correspond to the two high-order switches of the interrupt address (on=closed=0; off=open=1):

Switch	8	7	4	3	2	1	Interrupt request name
	on	on	on	on	on	off	INRQ3
	on	off	on	on	off	on	INRQ2
	off	on	on	off	on	on	INRQ1
	off	off	off	on	on	on	INRQ0

5. Settings for Bus Addresses and Requests

The following settings are recommended:

I/O Device	Bus Address	Bus Request Bit Off	Bus Request Name	Priority Level
MTP addresses E0, E1, E4, E5	0000	1	BSRQ3	highest
IOP address FC	0100	2	BSRQ2	
IOP address FB	1000	3	BSRQ1	
MTP addresses E2, E3, E6, E7	1100	4	BSRQ0	
DMP (hardwired)	1111	wired	BSRQ0	lowest

6. Settings for Interrupt Addresses and Requests

The following settings are recommended:

I/O Device	Interrupt Address	Interrupt Request Bit Off	Interrupt Request Name	Priority Level
MTP addresses E0, E1, E4, E5	0000	1	INRQ3	highest
IOP address FC	0100	2	INRQ2	
IOP address FB	1000	3	INRQ1	
MTP addresses E2 E3, E6, E7	1100	4	INRQ0	lowest

MTP Controller Number

Switch 10 on S1 determines the controller number. If switch 10 is on (or closed or 0), the controller number is 0, which handles devices X'EO', X'El', X'E4, and X'E5'. If switch 10 is off (or open or 1), the controller number is 1, which interfaces with devices X'E2', X'E3', X'E6', and X'E7'.

Tape Drive Daisy Chain Switch

The MTP has a switch to specify whether there is a formatter for each tape drive or whether two drives are connected (daisy-chained) to one formatter. The switch is number 9 on S1. If the switch is on (or closed or 0), each tape drive is connected to its own formatter (the normal case). If the switch is off (or closed or 1), two drives are connected to one formatter.

TROUBLESHOOTING TECHNIQUES

These are procedures that can be used to gather data whenever there is a problem. Brief titles in parentheses refer to earlier paragraphs in this section that give more detailed information on the subject discussed. See the Table of Contents for quick referral.

Strategy

Debugging programs is an art requiring an eye for details and for inconsistencies. Many times it is the inconsistent detail that gives an early clue to a problem. In pursuing any system problem, observe the following guidelines:

(a) WRITE DOWN EVERYTHING.
(b) Note all details.
(c) Make sure all details are consistent with each other. Check that error messages and system data are consistent with known quantities, such as memory size, number of discs, number of terminals, etc.
(d) The firmware debugger is the most useful tool when the system hangs up. The software debugger is probably more useful for intermittent problems affecting only one virtual process. When neither debugger is operational, remember the DMP's display/alter memory function.
(e) The most probable areas for problems is with disc and with terminals.
(f) When a system with recently installed equipment will not operate properly, run the hardware diagnostics until you are confident that the hardware is working correctly. Then assume that there is a software problem. When a system has been fully operational for more than a few hours, assume any problems are software-related and use one of the debuggers to gather information.

Looking At a PIB

The PIBs start at location X'1400'. Each PIB is X'80' (128) bytes long. A PIB and its corresponding port both have the same number (also called a PIB link).

To look at a PIB using the firmware debugger convert the PIB number to the PIB address with the PD or PX command. Display the desired bytes with the M or W command.

Example:

```
>PD117  117   75  4E80
>M4E80  7D=
>W4E98  000C19B6=
```

When the software debugger is available, use the FID and displacement to display PIB bytes. Table A-15 shows the commands to use for displaying the first four bytes of various PIBs. The first four entries are for ports 0-3. The remaining entries are for the first PIB of each buffer:

Once a window has been established (";4"), it need not be repeated on subsequent commands. If the DMP display and alter commands are being used, the PIB or port number can be converted to a PIB address by multiplying the number by X'80' and adding the result to the address of PIB 0, X'1400'. For example, the address of the PIB for line 68 (X'44') is:

X'44' * X'80' + X'1400' = X'3600'

Checking PIB Links (Priority Queue)

The PIB links of the priority queue are in PIB bytes two and three. Byte two contains the number of the next PIB in the queue or X'FF' if it is the last PIB. Byte three contains the number of the previous PIB in the queue or X'FF' if it is the first. The PIB number is the same as the port number for the process.

Determining Which Process Is Running

When a virtual process is running, its PIB address is held in the monitor's R4, which is saved in locations X'90' - X'93'. To determine which process is running, first note the firmware debugger's initial message (Initial Display).

When the firmware debugger is entered, it prints a message telling the address of the next instruction. If the message includes a V, a virtual process was running. If the V is absent, the monitor was running.

TABLE A-15

DISPLAY COMMANDS

Key In	Port No. (Dec.)	Port No. (Hex.)	ACLC No. (Dec.)
X.FFFFFC.00;4	0	0	0
X.FFFFFC.80;4	1	1	0
X.FFFFFC.100;4	2	2	0
X.FFFFFC.180;4	3	3	0
X.FFFFFB.00;4	4	4	0
X.FFFFFA.00;4	8	8	1
X.FFFFF9.00;4	12	C	1
X.FFFFF8.00;4	16	10	2
X.FFFFF7.00;4	20	14	2
X.FFFFF6.00;4	24	18	3
X.FFFFF5.00;4	28	1C	3
X.FFFFF4.00;4	32	20	4
X.FFFFF3.00;4	36	24	4
X.FFFFF2.00;4	40	28	5
X.FFFFF1.00;4	44	2C	5
X.FFFFF0.00;4	48	30	6
X.FFFFEF.00;4	52	34	6
X.FFFFEE.00;4	56	38	7
X.FFFFED.00;4	60	3C	7
X.FFFFEC.00;4	64	40	8
X.FFFFEB.00;4	68	44	8
X.FFFFEA.00;4	72	48	9
X.FFFFE9.00;4	76	4C	9
X.FFFFE8.00;4	80	50	10
X.FFFFE7.00;4	84	54	10
X.FFFFE6.00;4	88	58	11
X.FFFFE5.00;4	92	5C	11
X.FFFFE4.00;4	96	60	12
X.FFFFE3.00;4	100	64	12
X.FFFFE2.00;4	104	68	13
X.FFFFE1.00;4	108	6C	13
X.FFFFE0.00;4	112	70	14
X.FFFFDF.00;4	116	74	14
X.FFFFDE.00;4	120	78	15
X.FFFFDD.00;4	124	7C	15

If a virtual process was running, its PIB address is in locations X'90' - X'93'. Use the W command (Wxxxxxx) to display these locations (W90). The PIB address is in the three low-order bytes of the display. The two PIB bytes at displacement X'1E' contain the buffer number of the process's PCB (Layout of PIB). Multiply the buffer number by X'100' to get the PCB's main memory address (X'100' * X'0F49' = X'0F4900').

If the monitor was running when the firmware debugger was entered, the AF command (AFxx) can be used to display R4's contents and flags. The low-order digit of the two flag-digits reflects the flag settings. If the high-order bit of this digit is set, the register is attached, and the displayed address is being referenced by the monitor. If the register is not attached, display the four bytes at X'90' to see what the monitor was last referencing. When the monitor is running, R4 need not point to a PIB, but it usually will.

Looking at Buffer Status

Buffer status is recorded in the BSAFT, which begins at memory location X'A000'. To see the status of a buffer, calculate its entry in the BSAFT. First, convert the buffer address to a buffer number by masking off the two low-order hexadecimal digits of the buffer address (X'123400' becomes X'1234'). Multiply the buffer number by two (2 * X'1234' = X'2468' + X'A000' = X'C468'). This yields the byte address of the buffer's status byte. Use the W command (Wxxxxxx) to display the status byte and the FID in the next three bytes.

A section titled "Buffer Status and FID Table (BSAFT)" contains an explanation of the status bits. Remember that a zero means the condition is true.

The firmware debugger's F command (Fxxxxxx) can be used only if a display of the FID is desired. The F command also reports the buffer's I/O-busy status with a message if the buffer is I/O busy and with no message if it is not.

Following the Buffer Queue Links

The BQ links can be followed from the most recently attached buffer to the least recently attached buffer, or they can be followed from the least recently attached buffer to the most recently attached buffer.

To start with the most recently attached buffer, display the offset in locations X'1A002 - X'1A003'. Add this offset to the 3Q base address, X'1A000' to yield the address of the four-byte entry of the most recently attached buffer. The third and fourth bytes contain the offset of the next most recently attached buffer entry unless they contain zero, which signals the end of the list.

To follow the link from the least recently attached buffer, start with the offset in locations X'1A000' - X'1A001'. The first two bytes of each entry point to the next least recently attached buffer entry. A link of zero indicates the end of the list.

Determining Whether a Frame Is in Memory

To determine whether a frame is in memory, convert the size of the HAT into a mask for hashing the FID. The size of the HAT is calculated by converting the HAT's buffer number from locations X'200' - X'201' and the HLT's buffer number from locations X'202 - X'203' into addresses (Buffer Numbers). Subtract the HAT's address from the HLT's address. Divide the difference by two, yielding the number of HAT entries. Subtract one from the dividend to develop a mask. AND this mask with FID and multiply the result by two. Add the result to the HAT's address. The sum is the byte address of the HAT entry for the FID.

If the HAT entry contains X'FFFF', the frame is not in memory. Otherwise, the entry contains an offset into the BSAFT and the HLT. Add the offset to the base of the BSAFT, X'A000'. Display the four bytes at the resulting address. The first byte contains the buffer status; the last three bytes contain the FID.

If the FID is not the one you are searching for, add the offset to the base address of the HLT to obtain the address of the next entry in the list. If the first two bytes of this entry contain X'FFFF', the frame is not in memory. Otherwise, the two bytes contain another offset into the BSAFT and HLT.

If the FID is found, its memory address can be calculated from the offset into the BSAFT. Divide the offset by two to yield a buffer number, and multiply the buffer number by X'100'. For a one-step operation, multiply the offset by X'80'.

CHECKING ON DISC I/O

To check on disc I/O, examine the mailboxes, the chains of the task descriptors, and the interrupt byte (X'1A').

Examining a Disc Mailbox

The disc mailboxes, each four bytes long, are at locations X'500' - X'53F'. Use the low-order hexacdecimal digit of the device address to calculate the mailbox address. For example, the mailbox for device X'D5' would be:

5 * 4 = 500' = X'514'

Use the debugger to look at locations X'514' - X'517'.

Examining a Disc Task Descriptor Chain

The second byte of a disc mailbox contains the PIB link of the first task descriptor in the chain. Convert the PIB link to the PIB's main memory address (Looking at a PIB). PIB bytes 4-'X'F' are the task descriptor. Byte X'F' contains the PIB link of the next PIB in the chain.

Examining the Disc Interrupt Byte

Display location X'1A' to see if any discs have interrupted. The bits of the byte, when equal to one, indicate that the corresponding disc has interrupted. Bit 0 corresponds to device X'D0'; bit 1, to device X'D1'; and so forth.

Look at the disc I/O chains (Examining a Disc Mailbox and Examining a Disc Task Description Chain). If there are chains, it implies that there is disc activity. If there is disc I/O going on, there should be disc interrupts. If byte X'1A' contains zero, it indicates that no interrupts have been received. However, interrupts are posted to X'1A' between instructions only. When you first activate the debugger, record the contents of location X'1A'. Most of the debugger's E command (Exxx) single step the system. After each instruction, record the contents of X'1A'. One or more interrupts should have been posted.

Look at X'1A' before and after each instruction that is single-stepped, checking for interrupt bits to be set on. If you start stepping immediately, you may lose the original contents of the byte.

If there were interrupts posted in location X'1A', when you first started the firmware debugger, perhaps the monitor is not servicing interrupts correctly.

MONITOR BUILT-IN TROUBLESHOOTING ROUTINES

The monitor has several troubleshooting routines built in. They can be enabled by patching appropriate places in the monitor. The routines do the following:

(a) Record the last 42 disc start I/O instructions.

(b) Initialize a memory buffer to a known pattern before reading disc data into it.

(c) Perform a read to a different buffer and compare data to verify all disc operations.

Refer to Table A-18 for details on performing these routines in each release.

SYSTEM PERFORMANCE AND OPERATION TOOLS

System performance and operation tools are presented in the following paragraphs.

DISCIO Verb

The DISCIO verb displays a five-second "snapshot" of system disc I/O activity and provides a measurement of system disc performance. This display aids in locating causes of system performance problems. DISCIO generates two reports depending on the specified options. The first report (Report 1) consists of disc unit I/Os per second, disc I/Os per second, disc reads per second, disc writes per second, cumulative disc unit I/Os, cumulative disc I/Os, cumulative disc reads, and cumulative disc writes. The cumulative values are zero at the DISCIO operation's start. The second report (Report 2) consists of disc unit I/O totals. The general verb form is:

DISCIO (options)

To ensure accurate results during Report 1 displays, all ports must be logged on.

Three options exist for the DISCIO verb. The "T" option indicates the report type. When this option is specified, Report 2 is generated. Without this option, Report 1 is generated. The "n" option indicates the number of display iterations with "n" being any decimal number. This option only affects Report 1 displays. If this option isn't specified, one display iteration is performed. Report 2 always performs one display iteration. The "P" option indicates the output device. When this option is specified, the report is output to the printer. Without this option, the report is output to the terminal.

DISCIO reveals system performance problems. A low number of disc I/Os per second and a high number of active processes may indicate a disc-I/O bottleneck. This problem may be caused by an inadequate number of disc units or controllers, a slow disc-seeking mechanism, or inefficient monitor disc-handling code. A high number of reads per second and a high proportion of I/Os on one disc unit may imply a permanent disc error.

BUFFERS Verb

The BUFFERS verb displays main memory contents and provides a measurement of buffer I/O activity. This verb displays two reports depending on the specified options. The first report (Report 1) contains buffer locations, FIDs and the status of frames in buffers. The buffer status consists of I/O busy, monitor write, permanently memory-locked, and write-required indicators. The second report (Report 2) contains the number of buffers occupied by the monitor. ABS frames, work space frames, user program

data frames, I/O-busy frames, memory-locked buffers, and write-required buffers. The general verb form is:

BUFFERS (options)

Five options exist for the BUFFERS verb. The "T" option indicates the report type. If this option is specified, Report 2 in generated. Without this option, Report 1 is generated. The "S" option affects only Report 1 displays, creating a listing sorted by FID. Without this option, Report 1's display is in buffer memory-location order. The "Z" option is also used with Report 1. If "Z" is specified, the Report 1 listing includes the status of buffers with a FID of zero; that is, monitor code. If the "Z" is not specified, Report 1 lists only buffers with FIDs that are non-zero. The "P" option indicates the output device. If this option is specified, the report is output to the printer. Without this option, the report is output to the terminal. The "N" option inhibits automatic paging of terminal output. Without this option, the terminal display pauses at the page and until some terminal input is entered. With this option, no pausing occurs.

The BUFFERS verb aids in system operation. Before coldstart operations no write-required frames may be memory resident to reduce the possibility of GROUP FORMAT ERRORS. The monitor automatically flushes memory when no active processes exist. Because Report 1 displays all memory-resident, write-required frames, buffers may be used to determine if a coldstart operation may be safely performed.

Where Verb

WHERE prints one line for every process logged on (including the spooler). It is more accurate for telling which ports are logged on the LISTU becasue LISTU depends on the ACC (accounting) file being correct.

The output of WHERE shows the port number, PCB for that port (in hexadecimal), PIB status byte, and the software return stack. See Appendix D for actual addresses listed by release.

Example:

```
;WHERE <CR
  PORT      PS      RTN STACK.
*02 0500    FC      121.0CC 121.073
 04 0540    78        6.092   6.04B   5.072
 10 06C0    78       21.032   6.08F   6.04B   5.072
 104111C0    3C     164.07C 164.06D
```

The return stack is particularly valuable. It tells you precisely what a process is doing at any time by showing the locations in ABS of the instructions being executed. Each return stack entry is part of a subroutine that was called by the next return stack entry.

Consider the return stack for port 4 in the previous example:

```
6.092   6.04B   5.072
```

This says that the port is executing an instruction in frame 6 at a displacement of X'92' bytes into the frame. That instruction is part of a subroutine which, when it is finished executing, will return to another instruction in frame 6 at displacement X'4B' bytes. That, in turn, is part of a subroutine which will return to an instruction in frame 5 at displacement X'72' bytes into the frame. This port, by the way, is at TCL.

If you know what parts of the operating system reside in which frames, you can get a good idea of what the system is doing at any time by looking at the WHERE output. In the above example, frame 6 is called TERMIO and frame 5, TCL-1. The instructions in frame 6 are waiting for terminal input and have been called by the TCL processor.

Port 2 is executing in frame 121, which is called WHERESUBS. That is the port that is actually doing this WHERE command. Port 104, the spooler, is executing frame 164, which is called SPOOLOUT.

Whenever the first entry in the return stack is in frame 21, that process is in the software debugger, either because someone hit the break key or the process aborted. Port 10 is in the debugger and was waiting for terminal input at TCL immediately before the debugger was entered. In this case someone simply hit break while the process was at TCL.

A few sample return stacks from a 1.0 system and descriptions of what the processes are doing follows:

```
122.188    122.12C
  Asleep.  This is often seen in ATP.

6.092    6.04B    156.09C    13.050
  Waiting for terminal input in the editor.

6.092    6.04B    181.1AF
  Waiting for terminal input in BASIC.

164.07C    164.06D
  The spooler

202.0AA    201.191
  Doing a file save
```

NOTE: There will be return stack when the MONITOR PROC is run in ATP. The return stack addresses will be one byte less than the ones WHERE gives. MONITOR shows the address of the last byte of the last instruction executed (the subroutine call) while WHERE shows the address of the first byte of the next instruction. That is, MONITOR prints where an instruction was called from and WHERE prints where it will return to.

Examples:

The MONITOR return stack for TCL is
6.091 6.04A 5.071

The MONITOR return stack for the spooler is
164.07B 164.06C

BRIEF DESCRIPTIONS OF DIAGNOSTICS

The diagnostic programs are tools for determining hardware errors. They should be used after a review of software data indicates that a problem is not caused by an error in the software.

Diagnostics Selected Through the DMP

In addition to system functions and terminal control functions, the DMP control menu allows selection of several diagnostics. Number 8, RUN CPU FIRMWARE DIAGNOSTIC, can be used to test the CPU boards. The ninth selection on the control menu, RUN DMP FIRMWARE DIAGNOSTICS, runs diagnostics in the controllers: IOP, ACLC, MTP, and disc. Function 10, DISPLAY/ALTER MEMORY, provides the capability to display and change main memory. This is useful when the firmware and software debuggers cannot be activated.

Note that the control menu's terminal control functions allow you to switch the DMP to a remote terminal and to change the baud rate. The REMT switch on the system's front panel also provides the means to switch from the local terminal to a remote one and back again.

General Instruction Test (GIT)

The purpose of the GIT, which is composed of software instructions, is to test as many of the software instructions as possible. Most of the instructions are tested. Some such as the tape I/O instructions are simply used to load other parts of the test. If these do not work, the diagnostic will not run, indicating a problem. Disc I/O and DATA/BASIC instructions are not tested. Also, some of the system instructions are not tested.

This diagnostic runs in a stand-alone environment; no other program can be running while it is testing the system. Minimally, it requires the CPU, the DMP, one memory array, the memory controller, the terminal on port zero, the ACLC, the IOP, tape unit zero, and the MTP. The program can be run in less than three minutes. There are two versions of the diagnostic. The procedures for running them are in the following:

PM20001290 General Instruction Test for SEQUEL - Complete Edition

PM20001291 SEQUEL General Instruction Test - Special Edition

The special edition tests only the user instructions in both monitor and virtual mode: system-oriented instructions, such as process scheduling and memory management instructions, are not tested.

Diagnostic Monitor

The diagnostic monitor provides a system environment for the software diagnostic programs. Besides loading the diagnostics, the monitor starts them and processes error messages for them. It also provides the capability of chaining together selected tests.

The monitor takes complete control of the CPU while running. In addition, it requires the memory controller, at least one memory array, port zero's terminal, the ACLC, the IOP, the MTP and tape unit zero. Operating procedures for the diagnostic monitor can be found in FS20001297.

Disc Formatter

The purpose of the disc formatter is to format the discs so that they may be used by the system. Although not strictly a diagnostic program, it does diagnose the disc surfaces and recover bad spots that it finds, assigning alternate tracks.

The program, which runs under the diagnostic monitor, requires at least the CPU, the memory controller, one memory array, the port zero terminal, the ACLC, the IOP, a disc controller, and one or two discs. Information on the run procedure can be found in document FS20001293.

Disc Controller Diagnostic

The purpose of the disc controller diagnostic is to check out the functional capability of the disc controller. All functions of the disc controller are tested. The disc drives are not tested.

This diagnostic runs under the diagnostic monitor using at minimum the CPU, the memory controller, a memory array, the port zero terminal, the ACLC, the IOP, one or more disc controllers with one or more discs on each controller. The program, whose procedure is described in document FS20001292, allows selection of several pre-defined tests. The operator may also run these tests in other combinations under the diagnostic monitor. See the above mentioned document.

REFLEX Drive Diagnostic

The purpose of the disc diagnostic is to provide tests to verify the proper operation of the REFLEX II disc drives. The program tests all disc drive functions. It also formats tracks and analyzes surfaces quickly for production testing. However, it is not a complete formatter program and should not be used as such.

The disc diagnostic runs under the diagnostic monitor, using the CPU, the memory controller, at least one memory array, the port zero terminal, the ACLC, the IOP, one or more disc controllers with one or two disc drives on each controller. See document FS20001296 for operating procedures.

ACLC Diagnostic

The ACLC diagnostic tests all functions of the ACLC. It also tests some of the IOP's functions. The CRT terminals are not tested; a loop-back cable is used to move the data from one ACLC port to another.

The diagnostic, which runs under the diagnostic monitor, requires the CPU, the memory controller, at least one memory array, the IOP, one or more ACLCs, and the port zero terminal. The run procedure is described in document FS20001901.

Magnetic Tape/Printer Diagnostic

This diagnostic tests most functions of the magnetic tape units and printers and all controller functions. The program runs under the diagnostic monitor, using the CPU, the memory controller, at least one memory array, the port zero terminal, the ACLC, the IOP, the magnetic tape/printer controller, one or more magnetic tapes, and one or more printers. Document FS20001902 describes the operating procedure.

Memory Diagnostics

There are two memory diagnostic programs: the resident memory diagnostic that runs in the DMP and the software memory diagnostic that runs in in the CPU.

1. Resident Memory Diagnostic

The resident memory diagnostic tests only that portion of memory occupied by the diagnostic monitor and the software memory diagnostic when they are running. Only the data-store capability of the memory array boards is tested, not the error-checking circuitry. Since memory contents are changed by this diagnostic, no program can be running in the CPU while this program is executing.

The resident memory diagnostic runs in the DMP using the memory controller, the first memory array, the ACLC, the IOP, and the port zero terminal.

2. Software Memory Diagnostic

The software memory diagnostic runs under the diagnostic monitor to test all memory array boards except for the area in which this diagnostic and the diagnostic monitor reside. All memory functions, including the error-detection networks, are tested.

The diagnostic runs as a stand-along program using the CPU, the ACLC, the IOP, the port zero terminal, and the memory controller. All memory array boards in the system are tested. There are a variety of tests, each requiring a different length of time. For details on running the tests see FS20001900.

SYMPTOMS

The usual external symptom of a problem is when some or all of the terminals either beep (in response to input) or do not respond at all. This section concentrates on internal software symptoms since the external symptoms, generally, do not give enough specific information to pursue analysis of a problem.

Please remember that these descriptions explain what previous investigations have led to. The same symptoms together with others not mentioned here could lead to a different cause for a problem.

All Terminals Beep Only

If all of the terminals are beeping in response to input, turn the rotary switch to TEST and type CTRL-F on the port zero terminal. Since the DMP does not use the ACLC for terminal I/O, you can usually break into the firmware debugger. Look at the status of several PIBs ("Looking at a PIB," "Layout of PIBs," and "Common PIB Status Values").

1. All Processes Output Roadblocked

If all processes have a status of either X'74' or X'F4', they are all output roadblocked.

2. Case History

All the ACLCs had the same mailbox status, indicating that either all ACLCs were in error or the IOP was in error. The IOP status in location X'240' was X'82', indicating a flag-register error.

What had happened was that the IOP and an ACLC had a handshaking error. The IOP set the flag-register error in its status word. The IOP looked in location X'244' for interrupt instructions: which processor to interrupt, the CPU, the DMP, or the undefined processor. This location had not been initialized; it contained zeros. The IOP tried to interrupt on line zero, which does not exist, and went into a loop.

3. Some Processes Output Roadblocked

If some of the PIBs have a status of X'74' or X'F4'. indicating output roadblocked, check to see if the PIBs are grouped according to ACLC ("Looking at a PIB"). Perhaps only one of the ACLCs is bad.

4. All Terminals But Zero Input Roadblocked

When the configurator initializes the system, it sets all PIBs except the one for line zero to input roadblocked status (X'78'). If the systems hang up with PIB zero frame faulted (X'5C') and all other PIBs input roadblocked, it indicates that the system is hanging up on its first frame fault.

Sleeping PIBS

If many PIBs have a sleep time of X'7FFFFFFF' in PIB bytes X'14' - X'17', a process may have locked a system lock and then aborted. The lock will not be unlocked until an END or OFF is entered at the terminal where the abort was reported.

Display byte X'1C' of several PIBS. If they all contain the same lock number, it strongly suggests that this is the problem. The number displayed, which should be in the range X'C0' - X'CF', is the main memory address of the system lock. Display the byte. It contains the PIB number, which is the same as the port number, of the process that aborted. Entering an END or OFF at the PIB's terminal will clear the lock. However, you may want to determine the cause of the abort before allowing other processes to use the logic protected by the lock.

All R1s Contain Same FID

If R1 for every process points to the same ABS frame, perhaps the processes cannot be attached because the buffer has been flagged I/O-busy in the BSAFT ("Buffer Status and FID Table" and "Determining Which Process Is Running") The I/O-busy is set so that data in the buffer cannot be changed.

If an ABS frame were to write data from itself at the same time that the destination terminal was disconnected, the ACLC, detecting the missing terminal, would wait to output the data. Meanwhile, the main memory buffer would remain I/O-busy and other processes could not attach to it.

All Disc I/O on One Disc or Controller

A bad disc or a bad controller will attract processes on a system with more than one disc. Initially, there is an even distribution of disc operations, but eventually every process is going to frame fault on the bad disc or controller.

When you look at the disc chains ("Mailboxes" and "Examining a Disc Mailbox"), you will see all chains are empty except for one or two. If only one mailbox has a chain, it indicates a problem with one disc. If two discs connected to the same controller have chains, it suggests that there is something wrong with the controller.

When You Cannot Enter The Firmware Debugger

When you try to enter the firmware debugger and cannot (i.e., the DMP prints "FIRMWARE DEBUG REQUEST," but nothing else is printed), it indicates that the CPU is in a loop that never returns to RNI. Use the display and alter memory function of the DMP to diagnose the reason for the hang-up. The symptom might be caused by a loop in the BQ links or by a loop in the HLT links.

BQ Links Form a Loop

If the BQ links have been incorrectly changed so that they form an endless chain, the system could loop a long time in the FAR instruction. The monitor's write-routine moves the value X'FFFFFFFF' into D1, which is at main memory locations X'8' - X'B', to search the entire queue. Since the queue loops back on itself, the FAR will not terminate execution until the count decrements to zero in a little over two hours. When the instruction terminates, the firmware debugger will be entered.

Usually, if the monitor's D1 contains X'FFFFFFFF', it is a good sign that this is the problem. Follow the BQ links to find out if they loop ("Following the BUFFER Queue Links"). Change

the two bytes at location X'1A002' to point to one of the entries in the loop. Then enter a CTRL-F to enter the firmware debugger.

HLT Links Form a Loop

If the HLT has been changed incorrectly so that the links form a loop, the system will loop forever, looking for a buffer. Determine which process is running ("Determining Which Process Is Running"). Display the PCB's registers to see which FIDs are being attached. Follow the HLT links of these FIDs to determine if they loop ("Hash Address Table" and "Hash Link Table"). If a loop is found, change the last entry's forward link to X'FFFF'. Then enter a CTRL-F to break into the firmware debugger.

ASC II CODE CHART

The ASC II Code Chart is given in Table A-16. A summary of firmware debugger commands appears in Table A-17. A function of the most commonly used frames makes up Table A-18.

Sample Return Stacks

Here are a few sample return stacks from this system and descriptions of what the processes are doing:

122.188 122.12C
Asleep. You often see this in ATP.

6.092 6.04B 5.072
Waiting for terminal input in TCL.

6.092 6.04B 156.09C 13.050
Waiting for terminal input in the editor.

6.092 6.04B 181.1AF
Waiting for terminal input in BASIC.

21.032 6.08F 6.04B 5.072
Process entered software debugger while waiting for terminal input in TCL.

121.0CC 121.073
Process is executing WHERE verb.

164.07C 164.06D
The spooler.

202.0AA 201.191
Doing a file save

MONITOR BUILT-IN TROUBLESHOOTING ROUTINES

Recording Disc Start I/O Instruction

To record disc start I/O instructions in a circular buffer, change the NOPs at the end of the !COUNT.DISCIO routine in MONITOR1 to a branch to !RECORD.DISCIO.

Change main memory location X'719' from X'0000' to X'1F78'. Each disc I/O will be recorded in a circular buffer at memory location X'9600'. The format of the table is:

9600 - Point to the last used table entry or zero if no disc I/O activity has been recorded.

9604 - Four unused bytes (zeroed by the configurator)

9608 - Beginning of data

The format of each entry is:

Bytes	Contents
0	Command
1	Next PIB in the chain
2-3	Buffer number
4	Device address
5-7	Sector number on the drive
8-9	Unused
10-11	PIB address

Initializing a Disc Buffer [RELEASE 1.0]

To initialize a memory buffer change, the NOPs in !GOTBUF in MONITOR1 to BSL to INIT.BUFFER.

Change the contents of X'639' from X'0000' to X'1960'.

The pattern used is stored in constants within two instructions in the INIT.BUFFER routine. The default pattern is X"CC", but it may be changed by replacing the constants.

Change locations X'780' and X'784' from X'CC' to whatever one-byte value (two hexadecimal digits) is desired.

Enabling Disc I/O Verification [RELEASE 1.0]

Enabling disc I/O verification requires two steps:

(a) When a system is coldstarted (X or A or AF), buffers must be reserved for the verify operations. Before answering "CONFIGURATION CORRECT?", type CTRL-F to break into the firmware debugger. In routine PIBINIT.2 of mode SMSETUP4, change the branch to PIBINIT.25 to NOPs.

Change the contents of X'F44' from X'1F69' to X'0000'.

Also, set a break point at location X'60F' in MONITOR1. Type LINE FEED to continue.

(b) When the break point at X'60F' is reached, the monitor will be loaded. In MONITOR2 at label DINT50, change the first two bytes after ZB DLOCK to a branch to DINT3350.

Change the contents of X'866" from X'FOOD' to X'1EB9'.

Initializing a Disc Buffer [RELEASE 1.0, Revision C]

To initialize a memory buffer, change the NOPs in !GOTBUF in MONITOR1 to BSL to INIT.BUFFER.

Change the contents of X'6B9' from X'0000' to X'196 D'.

The pattern used is stored in constants within two instructions in the INIT.BUFFER routine. The default pattern is X'CC', but it may be changed by replacing the constants.

Change locations X'780' and X'784' from X'CC' to whatever one-byte value (two hexadecimal digits) is desired.

Enabling Disc I/O Verification [RELEASE 1.0, Revision C]

Enabling disc I/O verification requires two steps:

(a) When a coldstarted (X or A or AF), buffers must be reserved for the verify operations. Before answering "CONFIGURATOR CORRECT?", type CTRL-F to break into the firmware debugger. In routine PIBINIT.2 of mode SMSETUP4, change the branch to PIBINIT.25 to NOPs.

Change the contents of X'FBD' from X'1fE2' to X'0000'.

Also, set a break point at location X'60F' in MONITOR1. Type LINE FEED to continue.

(b) When the break point at X'60F' is reached, the monitor will be loaded. in MONITOR2 at label DINT50, change the first two bytes after 2B DLOCK to a branch to DINT350.

Change the contents of X'866' from X'FOOD' to X'1EBC.

TABLE A-16

ASCII CODE CHART

DECIMAL	HEX	EBCIDIC EQUILVALENT	ASCII	PRISM CHARACTER	SPECIAL USE IN ROYALE DISPLAY
0	00	00	NUL	NONE P cs	DELAY CHAR. SORT KEY DELIMITER
1	01	01	SOH	NONE A c	PRISM HOME CHARACTER
2	02	02	STX	NONE B c	
3	03	03	ETX	NONE C c	END OF TEXT
4	04	37	EOT	NONE D c	
5	05	2D	ENQ	NONE E c	
6	06	2E	ACK	NONE F c	CURSOR FORWARD ON PRISM
7	07	2F	BEL	NONE G c	BELL ON PRISM
8	08	16	BS	NONE H c	BACKSPACE ON PRISM
9	09	05	HT	NONE I c	TAB
10	0A	25	LF	NONE J c	CURSOR DOWN ON PRISM
11	0B	0B	VT	NONE K c	VERTICAL ADDRESS ON PRISM
12	0C	0C	FF	NONE L c	SCREEN ERASE ON PRISM
13	0D	0D	CR	NONE M c	CARRIAGE RETURN
14	0E	0E	SO	NONE N c	
15	0F	0F	SI	NONE O c	
16	10	10	DLE	NONE P c	HORIZONTAL ADDRESS ON PRISM, BLANK COMPRESSION CHARACTER
17	11	11	DC1	NONE Q c	
18	12	12	DC2	NONE R c	RETYPE ENTIRE LINE. ENABLE STATE PRINTER
19	13	3A	DC3	NONE S c	DUMP PRISM SCREEN TO SAVE PRINTER
20	14	3C	DC4	NONE T c	DISABLE SLAVE PRINTER
21	15	3D	NAK	NONE U c	CURSOR BACK ON PRISM
22	16	32	SYN	NONE V c	
23	17	26	ETB	NONE W c	
24	18	18	CAN	NONE X c	CANCEL LINE
25	19	19	EM	NONE Y c	
26	1A	3F	SUB	NONE Z c	CURSOR UP ON PRISM
27	1B	27	ESC	NONE	
28	1C	1C	FS	NONE	
29	1D	1D	GS	NONE	
30	1E	1E	RS		
31	1F	1F	US	NONE	
32	20	40	SPACE	b SPACE	
33	21	5A	!	! A cs	
34	22	7F	"	" B cs	STRING DELIMITER IN ENGLISH AND BASIC
35	23	7B	#	# C cs	
36	24	5B	$	$ D	
37	25	6C	%	% E	
38	26	50	&	& F	
39	27	7D	'	' G	STRING DELIMITER EN ENGLISH AND BASIC
40	28	4D	(	(H	
41	29	5D	)	) I	
42	2A	5C	*	* J	
43	2B	4E	+	+ K	
44	2C	6B	,	, L	
45	2D	60	-	- M	

Table A-16 (Cont'd) ASCII Code Chart

46	2E	4B	.	.	N
47	2F	61	/	,	O
48	30	F0	0	0	P
49	31	F1	1	1	Q
50	32	F2	2	2	R
51	33	F3	3	3	S
52	34	F4	4	4	T
53	35	F5	5	5	U
54	36	F6	6	6	V
55	37	F7	7	7	W
56	38	F8	8	8	X
57	39	F9	9	9	Y
58	3A	7A	:	:	Z
59	3B	5E	;	;	
60	3C	4C	<	<	
61	3D	7E	=	=	
62	3E	6E	>	>	
63	3F	6F	?	?	
64	40	7C	@	@	
65	41	C1	A	A	
66	42	C2	B	B	
67	43	C3	C	C	
68	44	C4	D	D	
69	45	C5	E	E	
70	46	C6	F	F	
71	47	C7	G	G	
72	48	C8	H	H	
73	49	C9	I	I	
74	4A	D1	J	J	
75	4B	D2	K	K	
76	4C	D3	L	L	
77	4D	D4	M	M	
78	4E	D5	N	N	
79	4F	D6	O	O	
80	50	D7	P	P	
81	51	D8	Q	Q	
82	52	D9	R	R	
83	53	E2	S	S	
84	54	E3	T	T	
85	55	E4	U	U	
86	56	E5	V	V	
87	57	E6	W	W	
88	58	E7	X	X	
89	59	E8	Y	Y	
90	5A	E9	Z	Z	
91	5B	80	[	[	STRING SEARCH DELIMITER
92	5C	E0	/	/	
93	5D	90	]	[	ENGLISH STRING SEARCH DELIMITER
94	5E	5F	©	©	ENGLISH STRING SEARCH DELIMITER
95	5F	6D	_	_	
96	60	79		NONE	0 cs
97	61	81		NONE	1 cs
98	62	82		NONE	2 cs
99	63	83		NONE	3 cs
100	64	84		NONE	4 cs
101	65	85		NONE	5 cs

Table A-16 (Cont'd) ASCII Code Chart

102	66	86		NONE	6 cs
103	67	87		NONE	7 cs
104	68	88		NONE	8 cs
105	69	89		NONE	9 cs
106	6A	91		NONE	
107	6B	92		NONE	
108	6C	93		NONE	
109	6D	94		NONE	
110	6E	95		NONE	
111	6F	96		NONE	
112	70	97		NONE	0 cs
113	71	98		NONE	1 cs
114	72	99		NONE	2 cs
115	73	A2		NONE	3 cs
116	73	A3		NONE	4 cs
117	75	A4		NONE	5 cs
118	76	A5		NONE	6 cs
119	77	A6		NONE	7 cs
120	78	A7		NONE	8 cs
121	79	A8		NONE	9 cs
122	7A	A9		NONE	
123	7B	C0		NONE	
124	7C	6A		NONE	
125	7D	D0		NONE	
126	7E	A1		NONE	SORT KEY DELIMITER
127	7F	07	DEL	NONE	
128	80	04		NONE	
129	81	06		NONE	
130	82	08		NONE	
131	83	09		NONE	
132	84	0A		NONE	
133	85	13		NONE	
134	86	14		NONE	
135	87	15		NONE	
136	88	17		NONE	
137	89	1A		NONE	
138	8A	1B		NONE	
139	8B	20		NONE	
140	8C	21		NONE	
141	8D	22		NONE	
142	8E	23		NONE	
143	8F	24		NONE	
144	90	28		NONE	
145	91	29		NONE	
146	92	2A		NONE	
147	93	2B		NONE	
148	94	2C		NONE	
149	95	30		NONE	
150	96	31		NONE	
151	97	33		NONE	
152	98	34		NONE	
153	99	35		NONE	
154	9A	36		NONE	
155	9B	38		NONE	
156	9C	39		NONE	
157	9D	3B		NONE	

Table A-16 (Cont'd) ASCII Code Chart

158	9E	3E		NONE
159	9F	41		NONE
160	A0	42		NONE
161	A1	43		NONE
162	A2	44		NONE
163	A3	45		NONE
164	A4	46		NONE
165	A5	47		NONE
166	A6	48		NONE
167	A7	49		NONE
168	A8	4A		NONE
169	A9	4F		NONE
170	AA	51		NONE
171	AB	52		NONE
172	AC	53		NONE
173	AD	54		NONE
174	AE	55		NONE
175	AF	56		NONE
176	B0	57		NONE
177	B1	58		NONE
178	B2	59		NONE
179	B3	62		NONE
180	B4	63		NONE
181	B5	64		NONE
182	B6	65		NONE
183	B7	66		NONE
184	B8	67		NONE
185	B9	68		NONE
186	BA	69		NONE
187	BB	70		NONE
188	BC	71		NONE
189	BD	72		NONE
190	BE	73		NONE
191	BF	74	b	NONE
192	C0	75	@	@
193	C1	76	A	A
194	C2	77	B	B
195	C3	78	C	C
196	C4	8A	D	D
197	C5	8B	E	E
198	C6	8C	F	F
199	C7	8C	G	G
200	C8	8E	H	H
201	C9	8F	I	I
202	CA	9A	J	J
203	CB	9B	K	K
204	CC	9C	L	L
205	CD	9D	M	M
206	CE	9E	N	N
207	CF	9F	O	O
208	D0	A0	P	P
209	D1	AA	Q	Q
210	D2	AB	R	R
211	D3	AC	S	S
212	D4	AD	T	T
213	D5	AE	U	U
214	D6	AF	V	V
215	D7	B0	W	W
216	D8	B1	X	X
217	D9	B2	Y	Y

Table A-16 (Cont'd) ASCII Code Chart

218	DA	B3	z	Z		
219	DB	B4	[	[		
220	DC	B5	/	/		
221	DD	B6	]	]		
222	DE	B7	©	©		
223	DF	B8				
224	E0	B9	@	@		
225	E1	BA	a	A		
226	E2	BB	b	B		
227	E3	BC	c	C		
228	E4	BD	d	D		
229	E5	BE	e	E		
230	E6	CA	f	F		
231	E7	CA	g	G		
232	E8	CB	h	H		
233	E9	CC	i	I		
234	EA	CD	j	J		
235	EB	CE	k	K		
236	EC	CF	l	L		
237	ED	DA	m	M		
238	EE	DB	n	N		
239	EF	DC	o	O		
240	F0	DD	p	P		
241	F1	DE	q	Q		
242	F2	DF	r	R		
243	F3	E1	s	S		
244	F4	EA	t	T		
245	F5	EB	u	U		
246	F6	EC	v	V		
247	F7	ED	w	W		
248	F8	EE	x	X		
249	F9	EF	y	Y		
250	FA	FA	z	Z		
251	FB	FB	SB	[	K cs	START BUFFER
252	FC	FC	SVM	/	L cs	SUBVALUE MARK
253	FD	FD	VM	]	M cs	VALUE MARK
254	FE	FE	AM	©	N cs	ATTRIBUTE MARK
255	FF	FF	SM	_	O cs	SEGMENT MARK

TABLE A-17

SUMMARY OF FIRMWARE DEBUGGER COMMANDS

AFxx	Display address register and flags - modify flags.
AND	Break when both Match Byte compares are true.
ARxx	Display address register and flags - modify register.
Bxxxxxx	Break at main memory location (two allowed).
Cxxxxxx	Display and modify memory byte in ASCII.
D	Display all execution control parameters.
Exxx	Execute xxx (decimal number) instructions: then break into the debugger. E0 indicates no break requested.
Fxxxxxx	Display FID and displacement of memory address xxxxxx.
Gxxxxxx	Execute instruction at main memory xxxxxx. G without an address continues execution at the last break address.
K	Kill both break points.
LINE FEED	Continue execution at the last break address.
Mxxxxxx	Display and modify memory byte in hexadecimal.
OR	Break if either Match Byte compare is true.
PDyyyy	Convert decimal port number to PIB address.
PXxx	Convert hexadecimal port number to PIB address.
Txxxxxxzyy	Set up Match Byte yy for compare operation z (equal, =; unequal, #) against contents of main memory xxxxxx. Two Match Bytes may be specified.
U	Deactivate and reset both Match Bytes.
Wxxxxxx	Display and modify 32-bit memory word in hexidecimal.
X	Turn off Firmware Debugger and continue execution at the last break address.
Z	Display trace of previous 62 instructions.

TABLE A-18

COMMONLY USED FRAMES

FRAMES	FUNCTION
5	TCL
6	TERMINAL I/O
13-16	EDITOR
17-21	SOFTWARE DEBUGGER
33-34	OVERFLOW TABLE
35-36	TAPE I/O
44	PROC
46	SOFTWARE DEBUGGER
66-67	SOFTWARE DEBUGGER
109-110	DATA/BASIC
115	FILE-SAVE
119	DATA/BASIC
122	SLEEP
131-134	DATA/BASIC
137	DATA/BASIC
138	SOFTWARE DEBUGGER
140-143	COPY
147	DATA/BASIC
149	EDITOR
153	TAPE I/O
156	EDITOR
158	DATA/BASIC
161	SOFTWARE DEBUGGER
163-164	SPOOLER
166	SPOOLER
173-174	DATA/BASIC
176	TAPE I/O
180-186	DATA/BASIC
188	DATA/BASIC
200-206	FILE-SAVE
210-219	FILE-RESTORE
228	EDITOR
231-246	PROC
253-254	SPOOLER
277	COPY
285	TAPE I/O
287	EDITOR
336	EDITOR
338	SPOOLER
357	EDITOR
358	SPOOLER
369	SPOOLER
371-372	SPOOLER
376	SPOOLER

APPENDIX B

GLOSSARY OF MNEMONIC TERMS

APPENDIX

GLOSSARY OF MNEUMONIC TERMS

Term	Location	Definition
ADDR00-23		Data address Bus
ADR		Address
ALU		Arithmetic Logic Unit
BPAR0-8		RNI Second of Saved Program Address Registers
BXF3		ALU Byte X Sign Bit
BXOV		Byte °X° Overflow
CHAR		Character
CMD		Command
CPU		Central Processor Unit
CPUDXX		CPU Destination Bus (32 Lines)
CPUSXX		CPU Source Bus (32 Lines)
DDD		Logic Term Used For Fixed Character Tests to Indicate Bits 26 thru 31=1
DMP		Diagnostic Maintenance Processor
DYSN		
FID		Frame Identification
FLGXX		Programmable Flags, Set and Tested By CPU
FW		Firmware Word
HLD 81		Timing Hold to Prevent CPU Memory Access When Memory is Busy
I/O		Input/Output
LCFF		Loop Count Output = °FF° (A Micro-sequencer Branch Condition)
LCTX		Loop Count Value. (Used As A Micro-sequencer Branch Value)
MEM		Memory
MHLD/		Map Branch T Hold For RNI Not Ready
MSA		Mode Select °A°
MSB		Mode Select °B°
MUX		Multiplexer
PAR		Program Address Register
PROM		Programmable Read Only Memory
RAM		Random Access Memory
RNI		Read Next Instruction Or RNI Board
ROM		Read Only Memory
SF		Special Functions Board
STDT		Term to Load Data Transfer Control Latch
ADRO	A05	Control of Character Test Force Reference Address to 0
AROM	A05	Alterable Read Only Memory Enable
CLK1,2	A05	CPU Clocks Which Can be Disabled by Thold
DHLD	A05	Delayed T Hold

DIAG1,2	A05	Firmware Diagnostic Modes Set by CPU Affecting RNI, Branch Map and Memory
ECLK	A05	Buffered Master Clock to AROM
HLDA/	A05	Latched T Hold To Synchronize And Delay One Clock
HOLD/	A05	Timing Hold For Memory Busy And RNI Not Ready
MRST	A05	System Master Reset
STOP/	A05	CPU Stop Command From AROM
UDSN	A05	Control of Character Test Up/Down Scan
LTRL	A06	ALU Literal Source Decode Which Disables All Type B Operations On Its Clock
LTRL/	A06	Decoded Literal Source To CPU Bus
INST1,3	A09	Control Terms into 2901 ALU Which Are Modified By MUL/DIV Conditions
LSTK/	A09	Load Microsequencer Stack
PTRX	A09	Microsequencer Stack Pointer (4 Lines)
STKXX	A09	Microsequencer Stack Output (13 Lines)
SUBR	A09	Control Term To ALU Carry In MUX, Which Is Modified By Divide Conditions
ARXX	A10	Microsequencer Command Address +1 (Latches) 13 Lines
F3	A10	ALU Sign Bit
NEGA	A10	Latched ALU Negative Condition
OVFL	A10	ALU Overflow (Latched)
OVR	A10	ALU Overflow
ZERO	A10	Latched ALU Zero Condition
ZOCDO-3	A10	Latched ALU Byte X
BRCH/	A11	Microsequencer Branch is Occurring
CAXX	A11	Selected Microsequencer Command Address
COND	A12	Microsequencer Branch Condition
CONDX	A12	Intermediate Branch Conditions From First Rank of Branch Multiplexers
EINT	A12	External Interrupt Microsequencer Branch
IINT	A12	Internal Interrupt Microsequencer Branch
INTR	A12	Composite Interrupt Conditional Branch Term
SSWX	A12	Sense Switches for Conditional Branch
QO	A13	ALU Q Shift Input or Output
QLNK	A13	Latched ALU Q Shift Output
RAMO	A13	ALU RAM Shift Input or Output
RLNK	A13	Latched ALU RAM Shift Output
ALU/	A14	2901 Output Enable for ALU Operation
CO	A14	Output of ALU Carry in MUX
CRCT	A14	Divide Modification Logic Branch Term
CX	A14	ALU Carry Look Ahead Terms
LINK	A14	Used In ALU Operation To Link Previous Carry Out To Carry In
QUOT/	A14	Quotient Sign For Divide Modification Logic

GLOSSARY OF MNEUMONIC TERMS (CON'T)

PX,GX	A15	Carry Look Ahead Terms From 2901 ALU
ZX	A15	ALU BYTE X Zero Condition (4 Lines)
Q3	A16	ALU Q Shift Output Or Input
RAM3	A16	ALU RAM Shift Output Or Input
RDXX	A19/A26	Firmware ROM Output (64 Lines)
MCLK	MEM CONT.	System Master Clock
DONE	R02	RNI Picoprocessor Term Which Indicates A Parsing Sequence Is Complete
INCR	R02	RNI Picoprocessor Term Which Requests The Program Address Reg To Increment
LDKK	R02	RNI Load Term For KK
LDLBL	R02	RNI Load Term For Lower Label Register
LDLBU	R02	RNI Load Term For Upper Label Register
LDLIT	R02	RNI Load Term For Literal Register Enable
LDOP1/4	R02	RNI Picoprocessor Control For Loading Of The Appropriate First Rank, OP1-4 Reg
LITO-3	R02	RNI Literal Value For Modification Branch And For Jump Vector
LRDCO-2	R02	Undecoded RNI Load Terms For R, DSPL And CC
MWAIT	R02	RNI Picoprocessor Control Term Which Causes A Wait For Memory Complete
OPSELO-2	R02	RNI Control Terms For Selection of Picoprocessor Branch Data
SEQO-1	R02	RNI Load Terms For Picoprocessor Firmware Address
UL4	R02	RNI Upper/Lower Nibble Select For R2 And OP4
XWAIT	R02	RNI Picoprocessor Term Which Causes A Wait For Parsing Complete
MASKO-3	R02	RNI Mask Values For Modifying Branch Data
OP10-3	R03	RNI Second Rank Instruction Register OP1
OP20-23	R03	ENI Second Rank Instruction Register OP2
OP30-33	R03	RNI Second Rank Instruction Register OP3
OP40-43	R03	RNI Second Rank Instruction Register OP4
ROP10-13	R03	RNI First Rank Instruction Register OP1
ROP20-23	R02	RNI First Rank Instruction Register OP2
ROP30-33	R03	RNI First Rank Instruction Register OP3
LDCC/	R04	RNI Load Control For First Rank Instruction Register CC
LDDSP1/2	R04	RNI Load Control For First Rank Instruction Registers DSP1,DSP2
LDRRIX-2X	R04	RNI Load Control For First Rank Instruction Registers R1,R2
R10-13	R04	RNI Second Rank Instruction Register R1
R20-23	R04	RNI Second Rank Instruction Register R2
ROP40-43	R04	RNI First Rank Instruction Register OP4
RR10-13	R04	RNI First Rank Instruction Register R1
RR20-23	R04	RNI First Rank Instruction Regoster R2
KKO,1	R05	Word Length Extracted From Reality Software Instruction By RNI

GLOSSARY OF MNEUMONIC TERMS (CON'T)

NEXT/	R05	RNI Command For ALU To Load The Next Instruction Into The Second Rank Register
XFE	R05	Latched Frame Error Term From Address Computation
LBL-LOWER	R06	RNI Label Lower Register
LBL-UPPER	R06	RNI Label Upper Register
LITERAL	R06	RNI Literal Register
DSP10-17	R07	RNI Displacement Register (1)
DSP20-27	R07	RNI Displacement Register (2)
ACTV/	R08	RNI Memory Cycle In Process Branch Condition
DIRF/	R08	RNI Instruction Register Full Status Delayed One Clock
DR	R08	RNI Memory Activity Status
IHRF	R08	RNI Instruction Holding Register Full Status
INCPAR	R08	RNI Control Term To Increment The Program Address Register
IRCLK	R08	RNI Load CLock For Second Rank Of Instruction Register
IREN/	R08	RNI Load Enable For Second Rank Instruction Register
IRF	R08	RNI Instruction Register Full Status
RMRQ/	R08	RNI Memory Request
WAIT/	R08	RNI Wait Term For Memory Or Parsing Activity
MDR	R09	RNI Memory Cycle Complete
RBO-7	R09	RNI Instruction Parsing Bus
FRMX	R10	ADC Detected Frame Error Of Program Address Register
RPARO-8	R10	RNI First Rank Of Saved PAR (Program Address Register)
RRSO-3	R10	RNI Enables Of BYTES 0-3 From The Memory Data Register To The Parsing Bus
LPR/	R11	Load The RNI Program Address Register (CPU Destination Bus Function)
RPAR9-23	R11	RNI Program Address Register Upper Part
EQDCOND	R12	RNI Picoprocessor Conditional Branch Test Condition
OPMP-3	R12	RNI Selected OP Register Terms Used For Pico-Processor Branching
LRO/7	R13	RNI Picoprocessor Firmware Address
OPXO	R13	RNI Selected Or Register Term Used For Picoprocessor Branching
JVO/-3/	R14	RNI Picoprocessor Firmware Address Jump Vector
LRPO/-3/	R14	RNI Picoprocessor Conditional Branch Test Condition
MFLGO/-3/	R14	ADC A/R Flag Updata Terms, Old Values Or Modified Values
NFLGO/-3/	R15	ADC A/R Flag Update Terms
LFLGO-3	R16	ADC A/R Flag Old Values, Output Of Anti-Race Latch
RFLGO-3	R16	ADC A/R Flags, Loaded When ever Destination Is Not HARl

RNO-3	R16	ADC A/R Flags, Loaded Whenever Destination Is HAR1
XFLGO-23	R16	ADC A/R Flags, Loaded Whenever Destination Is HAR1
AROO-23	R18	Output Of Address Register Files (A/R)
ACEN/	R19	ADC Enable Term Combining Decoded Command With Thold
ADRC	R19	Decoded Address Computation Command
DSTE/	R19	ALU Destination °E° Loop Center
HAR1D	R19	ALU Destination Decode For HAR1
HAR2D	R19	ALU Destination Decode For HAR2
LDF1/-2/	R19	ADC Flag Register (XFLG,RFLG) Load Enables
LDSA/	R19	ALU Destination Decode, Load ALU Displacement
LHAR1/	R19	HAR1 Load Enable
LHAR2/	R19	HAR2 Load Enable
LPRL/	R19	Load RNI Program Address Register Lower Part
LPRU/	R19	Load RNI Program Address Register Upper Part
LRA/	R19	ALU Destination Decode, Load ALUR
LWE/	R19	ADC Lower A/R File Write Enable
UWE/	R19	ADC Upper A/R File Write Enable
DSPAO-7	R20	ADC Displacement Register Loaded By The ALU (ALUDSPL)
RAO-3	R20	ADC A/R File Register Number Which Is Loaded By The ALU (ALUR)
RDSPO-3	R20	ADC Selected °R° Used To Generate The RDSPL Term
SDSPO-7	R21	ADC Displacement Term Prior To Shifting Term
DSPZ	R22	Term Which Forces Displacement To Zero
HDSPO-8	R22	ADC Selected And Shifted Displacement Term
SBPAR/	R23	CPU Source Bus Enable For RNI Program Address Register
SLBL/	R23	CPU Source Bus Enable For RNI Label Register
SLIT/	R23	CPU Source Bus Enable For RNI Literal Register
ADDC	R24	ADC Adder Carry Out
CADRO-8	R24	Output Of ADC Adder
PADC	R24	ADC Preadder Carry Out
NARO/-11/	R25	ADC Lower A/R File Source
FERRO	R26	ADC Latched Error For Not HAR1 Destination Of ADC Operation
FERR1	R26	ADC Latched Frame Error For HAR1 Destination Of ADC Operation
FERRD	R26	ADC frame Error Detect
PADD1-2	R26	ADC Preadder Constant Term
PADD3X	R26	ADC Preadder Constant
PADD3Y	R26	ADC Preadder Constant
SHSLO-1	R26	ADC Displacement Shift Control Terms
MAPX	R26	Microsequencer Map Branch (8 Lines)
BSADO-3	S05	I/O Bus Priority Sequence Lines
BSRQO-3	S05	I/O Bus Request Lines

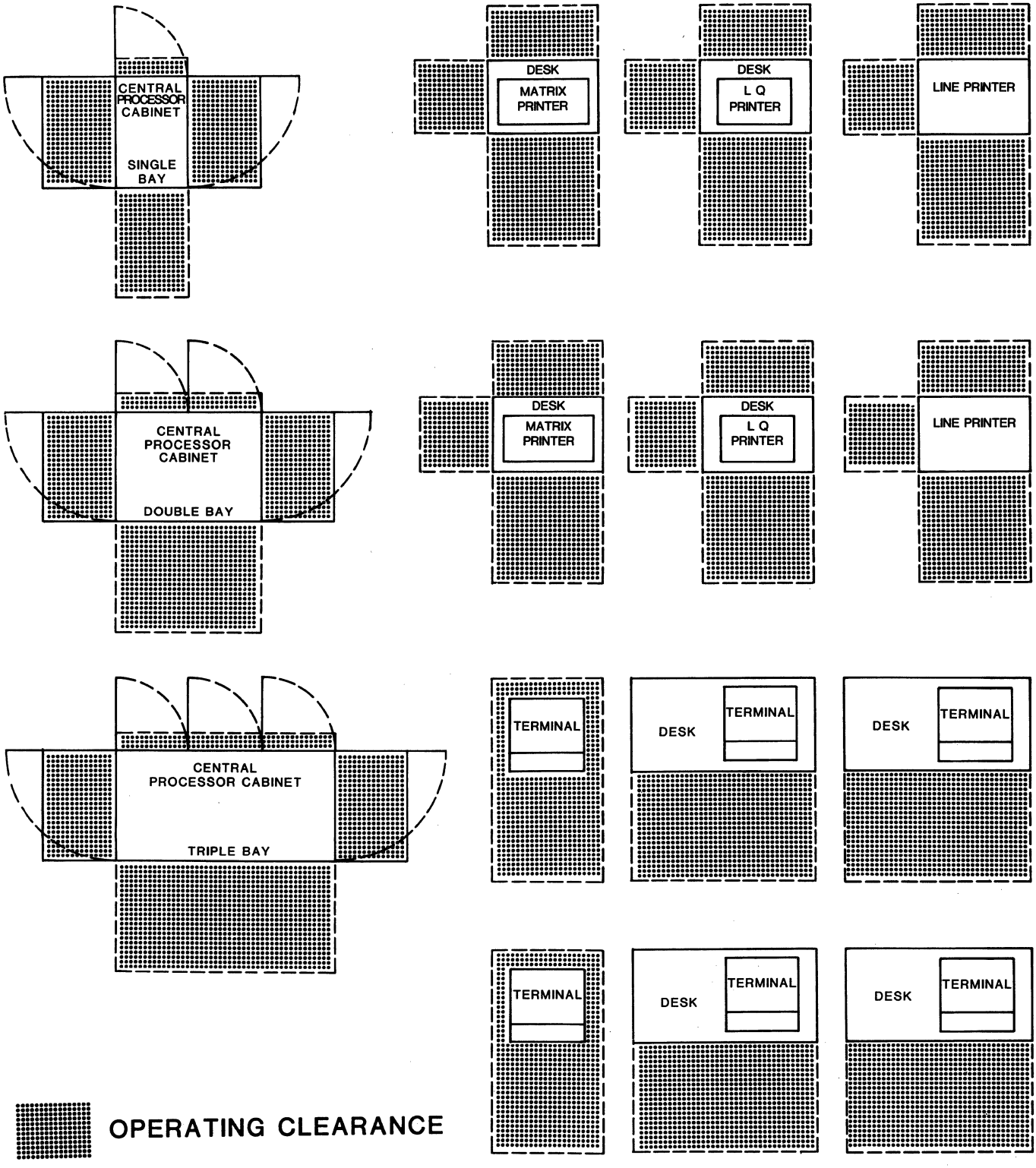
GLOSSARY OF MNEUMONIC TERMS (CON'T)

BSYN	S05	Signal Indicating Selected I/O Device Has The BUS
IOST/	S05	Inverted BSYN/
INADO-3	S06	I/O Interrupt Priority Sequence Lines
INRQO-3	S06	I/O Interrupt Request Lines
ISYN1-3	S06	Signal Indicating Selected Interrupting Device Is Active
ADGT	S07	CPU Addres Gate
AMGT	S07	CPU/ALU Grant
BGST	S07	I/O Bus Grant
DAGT	S07	CPU Data Gate
GHAR1-2	S07	CPU/ALU Operand Address Register Enables
GRNT	S07	BUS Cycle In Process
IOGT	S07	Signal Indicating That Memory Cycle Is For Requesting I/O Device
RAGT/	S07	RNI Address Gate For Memory Reads
RDCK	S07	CPU/ALU Read Data Strobe
RDST	S07	RNI Read Data Strobe
RFGT	S07	Refresh Grant
RFRQ/	S07	Refresh Request From Special Functions Board
RQST	S07	Refresh, RNI Or CPU/ALU Memory Request
WTEN	S07	CPU/ALU Write Data Enable
ADSO	S08	Operand Source Select For CPU/ALU Cycles (HAR1 Or HAR2)
AMRQ	S08	CPU/ALU BUS Cycle Request
BFW22-25	S08	Buffered Firmware Bits Used In Character Test For Byte Test Enables
CTEN/	S08	Decoded Type °B° Command Used For Character Test Enable
EXBT	S08	Latched X Register Bit O
FRME/	S08	Fixed Character Test Enable
FWLNO-1	S08	Firmware Designation Of Operand Word Length For Memory Write
HRBS	S08	CPU Generated BUS Hold For Read/Modify Write Operations
IOAT	S08	I/O Attention
LCTP/	S08	CPU/ALU Destination Character Test Mask
MWRC	S08	Memory Write Register Decode
RDWT	S08	CPU/ALU Read Or Write Select
TSQNO-1	S08	Buffered FW Bits 26,27 Used To Control Character Test Sequence Number
WBES	S08	Write Byte Enable Source Select
CLKA-B-C-D	S09	Repowered Master Clock
DPAR/	S09	Memory Data Parity
LKOT	S09	Blocks I/O Memory Requests While The ALU Holds The BUS
RSTA/	S09	Repowered System Reset
AMON	S10	Branch Condition, CPU Memory Cycle In Process
BSTM	S10	BUS Transfer Terminal Signal

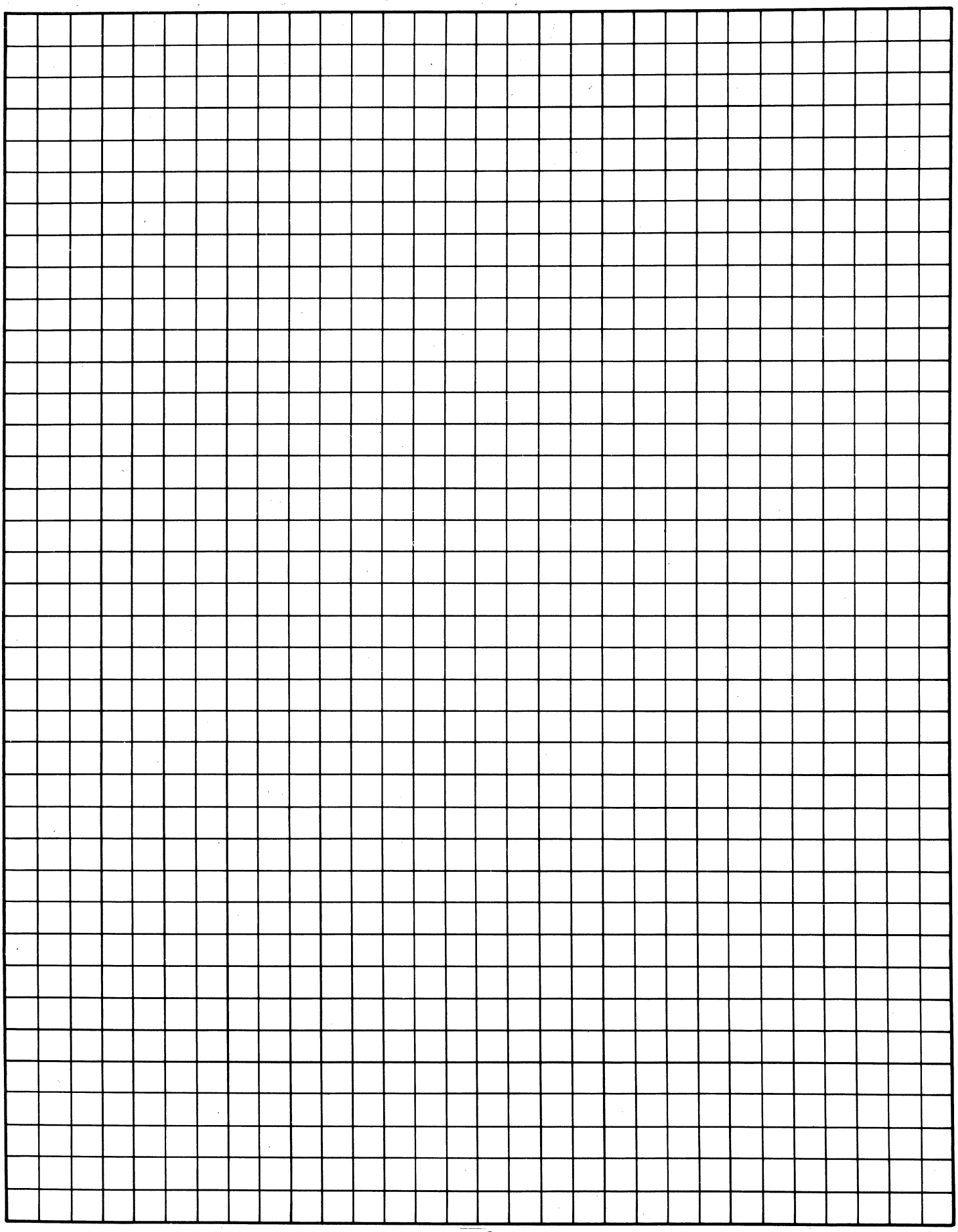
DYSN	S10	Signal From Memory Or I/O Device Signifying End Of Cycle
MBSY/	S10	Memory Busy Status From Special Functions Board
TOUT	S10	Time-Out Of BUS Access Which Causes A Termination To Prevent Lock-up
WTBY0-3	S10	Memory Write Byte Enables
DATA00-31	S11	Data Output Of Memory Write Register
FXFC-FXFF	S13	Fixed Character Test Mask Bits
MRRB00-31	S13	Data Output Of Memory Read Register
TORF	S13	True Or False Character Test Mask Bit
VSQI-2-3	S13	Variable Character Test Mask Bits
XR00-31	S14	Data Output Of °X° Register
XR31	S14	°X° Register Bit 31
XR00	S14	X Register Bit 0
XOMR	S15	CPU/ALU Source Select For °X° Or Memory Read Register
LWDLNO-1	S16	Overflow And Sign Selection
EXTD1-3	S17	Sign Extend Bit Select Control Terms
HAR100-101	S19	Hardware Address Register Bits 0 And 1
HAE100-123	S19	Memory Address register 1
HAR200-201	S19	Hardware Address Register Bits 0 And 1
HAR200-223	S19	Memory Address Register 2
BTAO-7	S20	Rotate Output Byte 0 (LS Byte)
BTBO-7	S20	Rotate Output Byte 1
BTCO-7	S20	Rotate Output Byte 2
BTDO-7	S20	Rotate Output Byte 3 (MS Byte)
RAMTO-1	S20	Rotate Amount Control Term For Rotate ROMS
WDLNO-1	S21	Word Amount Control Term For Sign Extension
ROTA/	S21	Type °A° Decode, Rotate Command
SELB1-2-3	S21	Sign Extend Byte Select Control Terms
CCDM	S23	Character Test Condition Met
CMBO-3	S23	Individual Byte Character Test For Condition Met
DLADO-1	S23	Latched 2 Bit Added To Memory Address To Calculate Of Test Condition

APPENDIX C

SITE LAYOUT WORKSHEET


CENTRAL PROCESSOR CABINET
SINGLE BAY
DESK
MATRIX PRINTER
DESK
L Q PRINTER
LINE PRINTER
CENTRAL PROCESSOR CABINET
DOUBLE BAY
DESK
MATRIX PRINTER
DESK
L Q PRINTER
LINE PRINTER
CENTRAL PROCESSOR CABINET
TRIPLE BAY
TERMINAL
DESK
TERMINAL
DESK
TERMINAL
TERMINAL
DESK
TERMINAL
DESK
TERMINAL
OPERATING CLEARANCE
SERVICE CLEARANCE
SCALE 1/4" - 1 FOOT

SITE LAYOUT WORKSHEET



SCALE 1/4" -1 FOOT