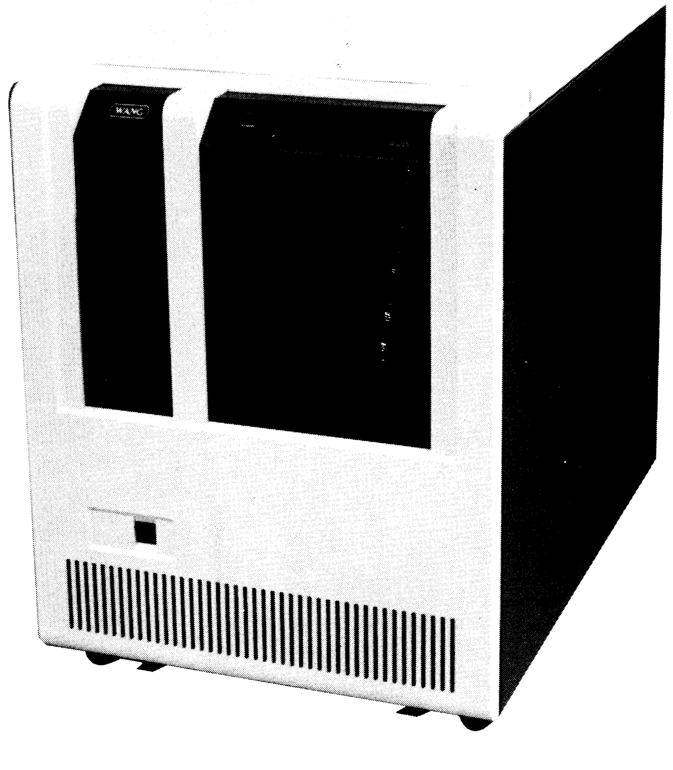
CUSTOMER
ENGINEERING

IV.A.3 M

# MODEL 2200LVP



# PRODUCT MAINTENANCE MANUAL

WANG

# CUSTOMER ENGINEERING

# MODEL 2200 LVP

# PRODUCT MAINTENANCE MANUAL

WANG LABORATORIES, INC., 1980

WANG LABORATORIES, INC.

ONE INDUSTRIAL AVENUE, LOWELL, MASSACHUSETTS 01851, TEL. (617) 459-5000, TWX 710 343-6769, TELEX 94-7421

## PREFACE

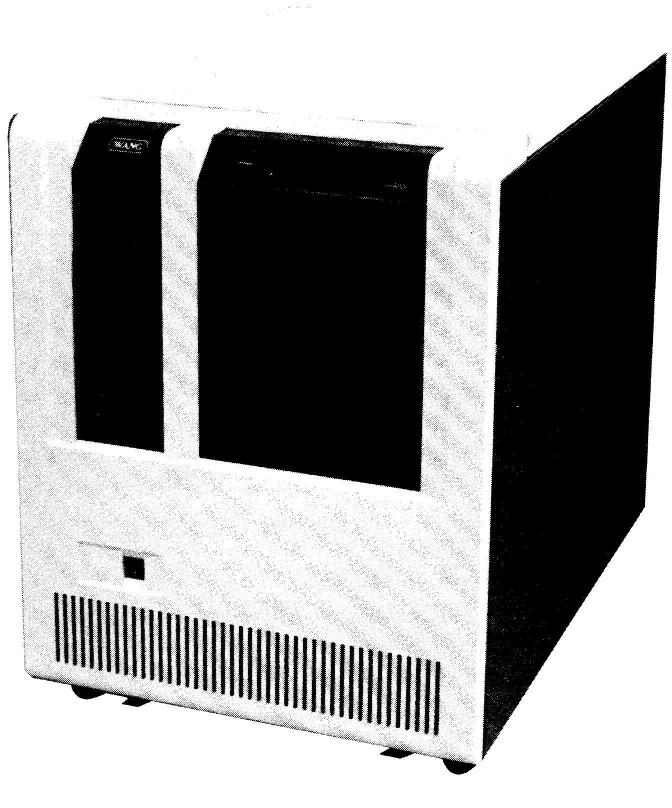
This manual provides field personnel with information needed to unpack, install, operate, and maintain the Model 2200LVP Central Processing Unit.

Following is a list of documentation categories referenced by this manual. In many cases, documentation from these other categories is directly required for the performance of certain maintenance tasks. Be sure to check the list of other required documentation at the beginning of each such task-section.

MODEL 2236MXD MULTIPLEXER/CONTROLLER -- IV.B.1
MODEL 22C32 TRIPLE CONTROLLER -- IV.B.1
I/O CONTROLLERS: SETTING DEVICE ADDRESS SWITCHES -- IV.B.1
I/O CONTROLLERS: PART #'S & I/O CABLE CONNECTION -- IV.B.1
I/O CABLE CONNECTOR INSTALLATTION -- I.B.0
2236DE INTERACTIVE TERMINAL -- III.D.1
DISK DRIVES -- III.A.11 AND III.A.12
PERIPHERALS -- Appropriate categories
OPERATING SYSTEM -- IV.C.4
MODEL 2200LVP CPU DIAGNOSTICS -- IV.C.1
PERIPHERAL AND DISK DRIVE DIAGNOSTICS -- IV.C.1
2200LVP CPU PREVENTIVE MAINTENANCE -- I.A.4
2200LVP CPU DATA MEMORY CAPACITY UPGRADES -- I.B.2
FIXED-DISK DRIVE CAPACITY UPGRADES -- I.B.2
SITE PLANNING & PREPARATION -- I.A.7
WANG BASIC-2 DISK REFERENCE MANUAL, WL #700-4081F -- III.A.0
2200VP BASIC-2 LANGUAGE REFERENCE MANUAL, WL #700-4080C -- IV.C.2
3740 DISKETTE COMPATIBILITY SOFTWARE RELEASE 2 USER MANUAL, WL #700-4369C

TABLE OF CONTENTS

LIST OF FIGURES



# SECTION 1

# GENERAL DESCRIPTION

## 1.1 SYSTEM OVERVIEW

The 2200LVP is an interactive, multi-user, multi-task, disk-based computer system. The LVP central processor supports up to five terminals and 16 jobs (tasks) concurrently, and is programmable in Wang BASIC-2.

The 2200LVP utilizes a user-defined, fixed-partition memory configuration along with a 600 ns cycle-time central processor to extend multiprogramming capabilities to system users. In a fixed-partition memory scheme, user memory is divided into a number of distinct areas called "partitions", each of which can contain a separate program. The central processor allocates intervals of processing time (time slices) to each partition in turn, permitting the program in an individual partition to execute for a brief time slice before servicing the next partition (called "interleaving"). Since programs performing input/output operations are not serviced until the operation is complete, they relinquish their central processing time to another partition. This method ensures a complete overlap of I/O processing, which is handled by peripheral controllers and CPU processing. By interleaving execution of different partitions, and bypassing those which cannot use central processing time, the 2200LVP response time decreases to create the illusion that each user has exclusive, continuous control of the system. Response time, an important consideration in a multi-user environment, is extremely fast for all users, regardless of the number of partitions or type of program currently executing.

State-of-the-art disk technology enhances the speed and versatility of the 2200LVP. Two new types of disk drives are available with the 2200LVP--a dual-sided, double-density (DSDD) diskette drive, which is IBM 3741-compatible, and a fixed, Winchester-style drive. Both storage devices represent the latest developments in cost-effective, high-speed, mass storage peripherals.

The expanded capacity diskette can be used to obtain faster backup with fewer platters. In addition to its backup capabilities, the DSDD diskette also serves as the medium for transferring system software and application packages. The fixed disk provides fast data access in a compact space without the mechanical or environmental problems associated with removable media-type drives. The DSDD diskette drive and the fixed-disk drive are mounted directly in the compact system housing, which also contains the central processor, thus saving space which separate drives would customarily require. (Refer to Section 1.7.2 for disk drive capacity specifications, and sector addressing scheme. Refer to Wang BASIC-2 Disk Reference Manual, WL #700-4081F--III.A.0, for an explanation of the BASIC-2 disk commands. Refer to 3740 Diskette Compatibility Software Release 2 User Manual, WL #700-4369C, for an explanation of 3740 diskette compatibility.)

System users communicate directly with the 2200LVP by using a 2236DE Interactive Terminal with business graphics capabilities. The terminal consists of a large, easy-to-read 80 X 24-character CRT-screen display and a typewriter-style keyboard. Up to five terminals can be attached (via one 2236MXD Multiplexer/Controller, and one 22C32 Triple Controller) either locally to the central processing unit at distances ranging up to 2,000 feet, or remotely by using modems and telephone lines. Line speeds, which range from 300 to 19,200 baud, are supported by using, asynchronous, full-duplex transmission. To accelerate communication and improve response time, the system performs automatic data compression on information transmitted to each terminal. Since each terminal has provision for connection of a local printer or plotter on the back of the unit, screen dumps may be output, and all standard printing operations may be performed. The 2236DE terminal also generates extensive bar and line graphics by standard program statements (ref: 2236DE Terminal documentation in category III.D.1).

At the customer's option, the 2200LVP can be equipped with telecommunications controllers to enable remote devices to be connected to the central processing unit. Both asynchronous and bisynchronous transmission are supported by the 2200LVP processor.

The 2200LVP also supports a wide range of peripheral devices, such as printers, plotters, disks, and tape drives. The current peripherals available for use with the 2200LVP are:

## PRINTERS

2201L Character
2221W Matrix (120 cps)
2231W Matrix (120 cps)
2251 Matrix (110 cps)
2261W Matrix (240 lpm)
2263W Chain (400/600 lpm)
2271 Bi-Directional (15 cps)
2273 Band (250/600 lpm)
2281 Diablo Daisy (30 cps)
2281W/WC Wang Daisy (40 cps)
IP41L Image (900 cps)

## PRINTER MULTIPLEXERS

2211M
2221M

## PLOTTERS

2232B Large Flatbed
2271P Bi-Directional
2272-2 Drum
2281P Daisy
2282 Graphic CRT

## TELECOMMUNICATIONS

2227B Asynchronous
2228B/C Bisynchronous

## TAPE DRIVE

2209A 9-Track (1600 bpi)

## DISK DRIVES *

2280 Cartridge Module

* 2230/60/70 model disk drives are supportable but are not sold in standard LVP-system configurations.

NOTE:

As of July, 1980, only 3 I/O slots are available in the 2200LVP, one of which is taken up by the 2236MXD Multiplexer/Controller. A field-upgradable, 9-I/O-slot version of the LVP is being designed.

FIGURE 1-1, on the following page, illustrates a typical 2200LVP system configuration.

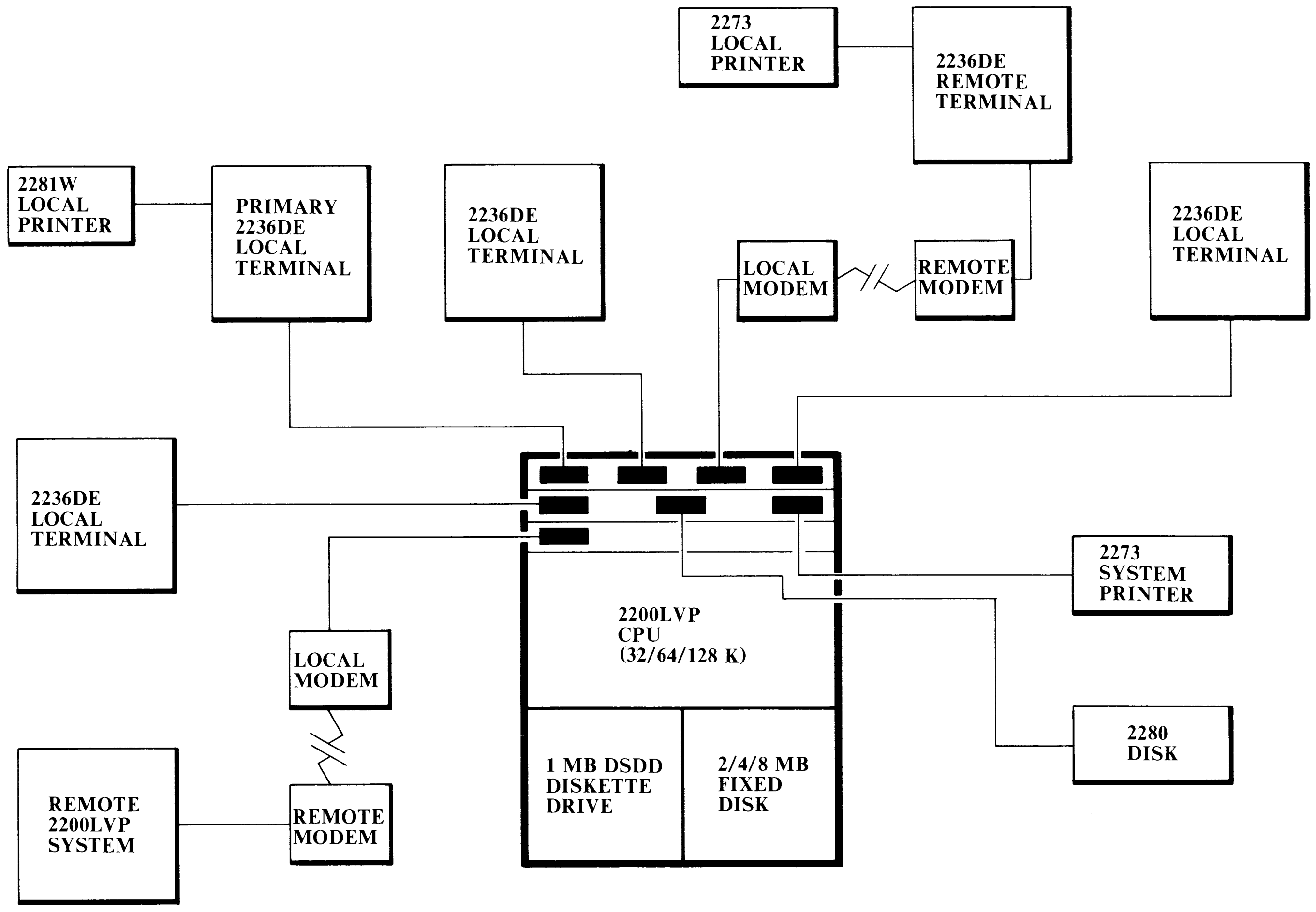


FIGURE 1-1 TYPICAL 2200LVP SYSTEM CONFIGURATION

## 1.2 PARTITION GENERATION AND SYSTEM CONFIGURATION

The number of partitions on the system and the size and characteristics of each are established initially in a process called "partition generation." (Wang provides a special utility program to facilitate the partition generation process.) When the number of partitions and the size limits of each partition have been defined, and when other system characteristics have been specified, a "system configuration" is created. The user can create/generate one or many such system configurations, each tailored to a specific set of processing requirements. All configurations can be uniquely named and then saved in a system disk file, to be accessed when needed. Optionally, the user can designate a particular predefined configuration to be automatically loaded and executed whenever the system is powered on.

The system configuration selected determines system and program operating parameters such as how many partitions will be created, how much memory will be allotted to each, and how many partitions will be assigned to each terminal. Once the system configuration is executed, each terminal on the system functions as if it were part of a single-user system. In general, each user can enter and run programs, interrogate and modify variables, and access common disk files as if there were no other users on the system.

## 1.3 MEMORY

The 2200LVP does not store its system programs (the BASIC-2 interpreter, operating system, and system diagnostics) in the same memory area used to store the application software. System programs are stored in a separate memory area called "control memory." The 2200LVP contains approximately 32K 24-bit words of control memory. When the system is powered on, the system programs are loaded into control memory from the system platter and remain resident in memory until the system is powered off or reinitialized. Control memory is a separate, protected memory area which cannot be accessed by the user or the user's programs. The system programs are, therefore, always protected against accidental interference or destruction by a user program.

User memory is the area of memory available to the user's programs and data. User memory may be 32K bytes, 64K bytes, or a maximum of 128K bytes. Because the system programs are stored separately, all user memory except for a small portion used for partition overhead, is available for user programs and data.

User memory consists of either one or two "banks" which contain a maximum 64K bytes each. The user may divide each bank into a number of partitions of fixed size, each of which can execute a separate program. The addressing scheme, however, does not permit partitions in the first bank to extend into the second bank. Within each bank, a fixed amount of memory is reserved for system overhead. In the first bank, 3K bytes are reserved for overhead and in the second bank, 8K bytes are unavailable to the user. Thus, a total of 61K bytes in Bank 1 and 56K bytes in Bank 2 are available to the user. The amount of system overhead is fixed, regardless of the total memory purchased for each bank. In addition, each partition in each bank requires approximately 1K bytes of partition overhead. All remaining memory in a single partition is available for user programs and data.

## 1.4 FOREGROUND/BACKGROUND OPERATION (ref: SECTION 2 for detailed information)

Since each terminal on the system may be assigned more than one memory partition, each terminal may be running several different jobs concurrently. Although the terminal may be running several jobs in different partitions, it can communicate with only one job at a time. The job which is currently communicating with the terminal is running in the "foreground." The job or jobs associated with a terminal but not currently communicating with it are running in the "background." A terminal may be switched from one partition to another, shifting the current foreground job into the background and shifting a particular background job into the foreground to permit operator communication with that program.

Foreground/background operation allows a user to run several jobs requiring varying degress of operator attention from a single terminal. A typical example would involve running a batch-type job requiring minimal operator interaction (such as payroll processing) in the background, while a more interactive job (such as order entry) runs in the foreground.

## 1.5 COMPATIBILITY WITH OTHER 2200 SYSTEMS

The 2200LVP has been designed to preserve compatibility with Wang's older, single and multi-user systems, as well as the more recent single-user systems. Since the 2200LVP is compatible with the 2200MVP, multiuser software written for the 2200MVP will function correctly on the 2200LVP. However, differences in the number of peripherals which can be attached to the system may affect some user programs.

Because the BASIC-2 language supported on the 2200LVP is identical to BASIC-2 on the 2200VP, there is 100% software compatibility between these systems for single-user programs. The 2200LVP also supports earlier Wang BASIC syntax, providing a significant degree of compatibility with non-VP and non-MVP systems.

## 1.6 MODEL CONFIGURATION

The 2200LVP is identified by a model number of the format:

2200LVP-xxy

where: 2200LVP represents the CPU, 1 megabyte of dual-sided double-density diskette, and the cabinet.

xx is a one or two digit number representing the actual CPU user-memory size when multiplied by 4.

| | |
|---|---|
| 8 - | 32 kilobytes of user-memory |
| 16 - | 64 kilobytes of user-memory |
| 32 - | 128 kilobytes of user-memory |

y is a capital letter representing the model for the fixed disk capacity option.

| | |
|---|---|
| B - | 2 megabyte fixed disk option |
| C - | 4 megabyte fixed disk option |
| D - | 8 megabyte fixed disk option |
| X - | No fixed disk option |

A summary of all model descriptions and numbers is given in TABLE 1-1.

TABLE 1-1 SUMMARY OF 2200LVP MODEL NUMBERS

| MODEL | DESCRIPTION | WL # 60 HERTZ | WL # 50 HERTZ |
|---|---|---|---|
| 2200LVP-8X | 32K Memory, 1 MB Floppy | 177-3204 | 157-3204 |
| 2200LVP-16X | 64K Memory, 1 MB Floppy | 177-3205 | 157-3205 |
| 2200LVP-32X | 128K Memory, 1 MB Floppy | 177-3206 | 157-3206 |
| 2200LVP-8B | 32K Memory, 1 MB Floppy, 2 MB Fixed Disk | 177-3207 | 157-3207 |
| 2200LVP-16B | 64K Memory, 1 MB Floppy, 2 MB Fixed Disk | 177-3208 | 157-3208 |
| 2200LVP-32B | 128K Memory, 1 MB Floppy, 2 MB Fixed Disk | 177-3209 | 157-3209 |
| 2200LVP-8C | 32K Memory, 1 MB Floppy, 4 MB Fixed Disk | 177-3201 | 157-3201 |
| 2200LVP-16C | 64K Memory, 1 MB Floppy, 4 MB Fixed Disk | 177-3202 | 157-3202 |
| 2200LVP-32C | 128K Memory, 1 MB Floppy, 4 MB Fixed Disk | 177-3203 | 157-3203 |
| 2200LVP-8D | 32K Memory, 1 MB Floppy, 8 MB Fixed Disk | 177-3210 | 157-3210 |
| 2200LVP-16D | 64K Memory, 1 MB Floppy, 8 MB Fixed Disk | 177-3211 | 157-3211 |
| 2200LVP-32D | 128K Memory, 1 MB Floppy, 8 MB Fixed Disk | 177-3212 | 157-3212 |

## 1.7 SPECIFICATIONS

### 1.7.1 2200LVP CPU

<u>Size</u>

Height - 27.0 in. (68.6 cm)
Width - 20.4 in. (51.8 cm)
Depth - 30.0 in. (76.2 cm)

<u>Memory Cycle Time</u>

600 nanoseconds

<u>User Memory Size</u>

32K bytes (standard)
Expandable to 64K or 128K bytes

<u>Control Memory Size</u>

32K 24-bit words

<u>Maximum Number of Partitions</u>

16

<u>Minimum Partition Size</u>

1.25K (1,280) bytes

<u>Maximum Number of Terminals</u>

5

System Overhead

3K (3,072) bytes for 32K and 64K machines
11K (11,264) bytes for 128K machines
1K (1,024) bytes per partition

Numeric Range

$10^{-100}$ to $10^{100}$, floating point with 13 significant digits

Power Requirements

115 or 230 VAC ± 10%
50 or 60 Hz ± 1.0 Hz
230 Watts

Fuses

5.0 amp (SB) for 115 V
3.0 amp (SB) for 230 V

Operating Environment

Temperature - 60° to 90°F (15° to 32°C)
Relative Humdity - 35% to 65% (noncondensing--recommended)
20% to 80% (noncondensing--allowable)

Heat Output

1,050 Btu/hr.

## 1.7.2 DISK DRIVES

| | 1 MB DSDD Diskette | 2 MB Fixed-Disk | 4 MB Fixed-Disk | 8 MB Fixed-Disk |
|---|---|---|---|---|
| Disk Surfaces | 2 | 1 | 2 | 4 |
| Sectors/Track | 26 | 32 | 32 | 32 |
| Tracks/Surface | 77 | **254 | ***255 | ***255 |
| Bytes/Sector | 374 | 320 | 320 | 320 |
| Bytes/Data Field | 256 | 256 | 256 | 256 |
| Sectors/Surface | 2002 | 8128 | 8160 | 8160 |
| Total Sectors | *3978 | 8128 | 16320 | 32640 |
| Total Bytes | 1,025,024 | 2,080,768 | 4,177,920 | 8,355,840 |
| Sector Addresses | *0-3977 | 0-8127 | 0-16319 | 0-32639 |

* The first track on side zero is single density, and is accessed by sector addresses 16384-16409. NOTE: The sector addresses for a single-density single-sided diskette are 16384-18386.

** Actually 256 -- the last track is reserved for alternate sector assignment, and the next to last track is reserved for diagnostic testing purposes.

*** Actually 256 -- the last cylinder is reserved for alternate sector assignment and for diagnostic testing purposes.

NOTES

# SECTION 2

# SYSTEM-LEVEL THEORY OF OPERATION

2200LVP operation is handled collectively by hardware, firmware, and software; however, the key to understanding how each major element of the system interacts with others comes by first understanding the method of memory control used in the Central Processor. This discussion will therefore commence in the general area of 2200LVP memory and memory control.

## 2.1 MEMORY RESOURCES IN THE 2200LVP

There are two random-access memory units in the 2200LVP--Control Memory and User Memory.

Microcode for the Operating System is contained in Control Memory. The Operating System is a software package dedicated to central processor time management, system memory management, and I/O operations management. Control Memory comprises thirty-two-thousand 24-bit words; that microcode is not accessible to users.

Physically separate from Control Memory is the RAM space allocated for User Memory--for storage of user programs, user data, and other information needed for correct user program execution. User Memory is divided into areas known as "banks"; a maximum of two banks are possible. A system containing 32 or 64 kilobytes of User Memory uses only bank #1; a system containing 128 kilobytes of User Memory uses both bank #1 and #2.

## 2.2 MEMORY MANAGEMENT IN MULTI-USER SYSTEMS

In a multiple-user system such as the 2200LVP, system resources must be shared. The simplest technique of sharing user memory space is called "partitioning".

Normally, the word "partition" means "a dividing wall". However, in the computer industry, the word has come to mean the space enclosed by the wall, rather than the wall itself. Henceforth, when discussing partitioned memory management, the "partition" is a block of memory space with specified address boundaries; it is not a boundary itself. The 2200LVP is configured such that each user is allocated one or more blocks (partitions) of User RAM which belong exclusively to him.

## 2.3 PARTITIONING 2200LVP USER MEMORY

### 2.3.1 MASTER INITIALIZATION

Before partitions are allocated for system users (during Master Initialization), a "system-use" block--comprising the first 3K in Bank #1 of User Memory--is established for Operating System housekeeping. (For this preliminary allocation, the Operating System might be loosely thought of as another "user" of User RAM space, requiring its own partition.)

During Master Initialization, the "MOUNT SYSTEM PLATTER" message is displayed at terminal #1; the operator at terminal #1 uses a Special Function key to load the Operating System from the system platter.

### 2.3.2 GENERATING THE PARTITIONS

The number of partitions to be created and the amount of User Memory to be allocated to each partition are specified by the user in a process called "partition generation". This process also involves specifying certain attributes for each partition (ref: SECTION 4) and supplying the addresses of peripheral devices connected to the system.

Once the Operating System has been loaded into Control Memory, the special utility program "@GENPART" is loaded and executed at terminal #1. This program leads the system operator through the necessary steps for "partition generation". A series of display prompts appear at terminal #1 that require the user to supply information pertinent to each partition and each shared peripheral device.

A "system configuration" is created by the @GENPART utility. Once created, a system configuration can be saved on disk for later recall. For this reason, a system configuration need be defined only once. A variety of system configurations can be created for different processing requirements; the operator can then select an appropriate configuration, as needed.

When the user has provided all of the information requested by @GENPART, or when the desired saved configuration is selected from the @GENPART display, the BASIC-2 statement $INIT must be executed. In the case of the Wang version of @GENPART, this is accomplished by keying SF 15. $INIT directs the Operating System to allocate resources as prescribed in @GENPART, in order to create a requested system configuration. Note that the $INIT statement alone may be used instead of the @GENPART program; but in either case, it is the $INIT statement which ultimately causes configuration to be carried out.

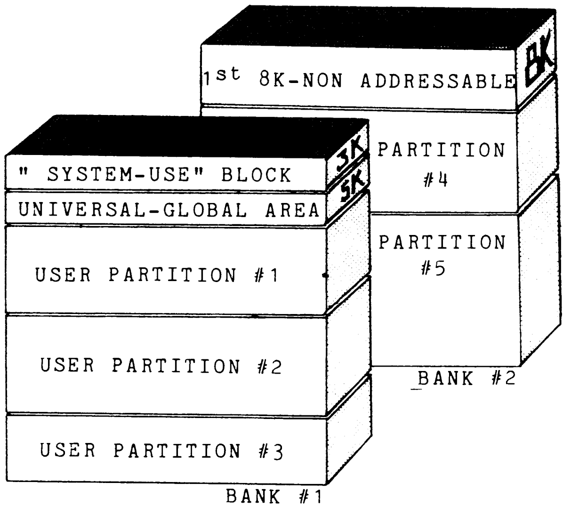
Once partition generation (partition allocation) has been implemented, each partition can be handled much like the entire user memory space of a single-user 2200 System: program text can be entered by a user, starting near the low end of his allocated partition, and his text entry progresses with ascending User Memory addresses; variable data for that program can be entered starting at the end (highest address) of his partition, and entry of that data progresses with descending addresses. This information will be illustrated in a partition diagram which appears in subsequent text of this section.

The LVP Operating System and CPU hardware will support a maximum of 16 partitions and 5 system users. All 16 partitions may be allocated to a single user, or multiple-parititon configurations may be created for each user. The 16 partitions (maximum configuration) may reside entirely within a single bank, or may be split up between both banks (as could be the case for a 128K LVP). One restriction, in regards to this latter statement, is that each partition must be defined wholly within the confines of a bank; that is, no user partition is allowed to extend from one bank to the next.

<u>NOTE</u>:

The first 8K of bank 2 is non-addressable, due to certain constraints of the LVP Operating System; this means that prior to partition generation time, there is only 56K (maximum loading) left for partitioning in bank #1. A 128-kilobyte LVP therefore provides an actual total of 117 kilobytes for partitioning of User Memory--61K in bank #1, plus 56K in bank 2.

2200LVP USER MEMORY



## 2.3.3 PARTITION SIZE & INTERNAL ALLOCATIONS

Partition sizes are specified in 256-byte (1/4K) increments. The first 947 bytes of each partition is used by the Operating System for "Operations Housekeeping" requirements of that paritition (this is not to be confused with the "system-use" block in bank #1). The minimum size that may be specified for any user partition is 1.25K.

Within each partition, there is also a User Program Text area, a Work Buffer, a Free Space area, a Value Stack, and a User Data Space (further explanation follows). Realize that neither the 947-byte housekeeping space, nor the Work Buffer, nor the Value Stack in each partition is addressable by the user; instead, values are stored in and retrieved from those blocks by the Operating System, according to the the conditions of execution existing in that partition at any given moment.

The Value Stack is not of fixed size; it expands and contracts in size during the course of program execution, and its size is zero prior to program execution. Typically, the Value Stack serves as a storage space for transient operands during the evaluation of mathematical expressions; subroutine return address information is also stored here, as required by the user's program.

The Work Buffer "floats" at the end of the Program Text area in memory. It is used to temporarily store information transferred into memory from the keyboard's input buffer, as well as for temporary storage of data for certain system functions such as LIST DC, MOVE and COPY. Immediate Mode lines and system commands transferred to the Work Buffer are immediately executed and then cleared; numbered program lines are moved from the Work Buffer area to the Program Text area so that they will be threaded into the user's program.

The Work Buffer can become as large as necessary (subject to available space) to contain an entered line. In every case, however, the system reserves a fixed minimum of 192 bytes for the Work Buffer. When the addition of a new program line or variable threatens to overlap into the minimum buffer area, a memory overflow error is signaled, and the program line or variable is not stored.

The actual amount of free space that exists in a partition at any given moment may be calculated by the two BASIC functions SPACE and END. Before computing this free space, the system automatically subtracts 192 bytes from the available space (for the minimum Work Buffer area). Thus, if END and SPACE return free space values of zero, there remains a minimum of 192 bytes still available for the Work Buffer.

It is important to recognize that a situation may arise in which initially, there _is_ sufficient free space to enter a program, but not enough free space to execute the program; this occurs when, specifically, the Value Stack requires more space than is available for the execution of the user's program. To determine how much free space actually is available, free space must be checked by SPACE during program execution when the Value Stack attains its _maximum_ size. Typically, this occurs when the program executes the innermost loop in series of nested loops. SPACE can be executed in the innermost loop to determine how much free space is available at the point.

The SPACE function returns, to the workstation screen, the amount of memory not currently occupied by program text or data, _minus_ the amount occupied by the Value Stack. This value represents the actual amount of free space in memory at any point during program execution.

The END function does not subtract the space taken up by the Value Stack.

## The Meaning of "Negative" Free Space

Although the system ensures that a minimum of 192 bytes always remain unoccupied by program text or variables in memory, it does permit the Value Stack to utilize a portion of this minimum buffer area. Up to 128 bytes of the 192-byte minimum Work Buffer can be used by the Value Stack. This fact implies that a program can be run even when memory is legally "full," since additions to the Value Stack during execution can overlap into the reserved Work Buffer area. Note that in this case the SPACE function would return a negative free space value.

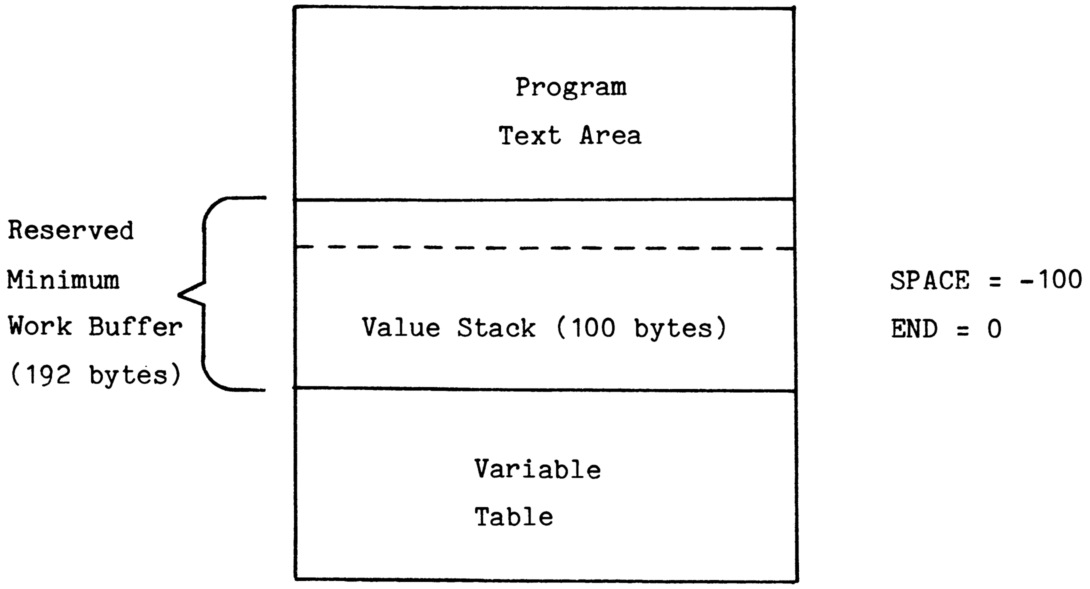


FIGURE 2-1 "NEGATIVE" FREE SPACE

To understand why this is so, consider the following:

When memory is so fully packed that the Value Stack must occupy part of the minimum buffer area, size of the Value Stack is subtracted from zero by SPACE, yielding a negative free space figure. Thus, a free space value of -100, returned by SPACE, indicates that memory is legally "full"; however, 100 bytes of the reserved minimum buffer have been used by the Value Stack. Since a maximum of 128 bytes of the minimum buffer area can be used by the Value Stack, SPACE cannot return a value less than -128. When the Value Stack requires more than 128 bytes of the buffer, a memory overflow error is signaled.

The following diagram offers a detailed summary of major allocations made in each partition (see next page):

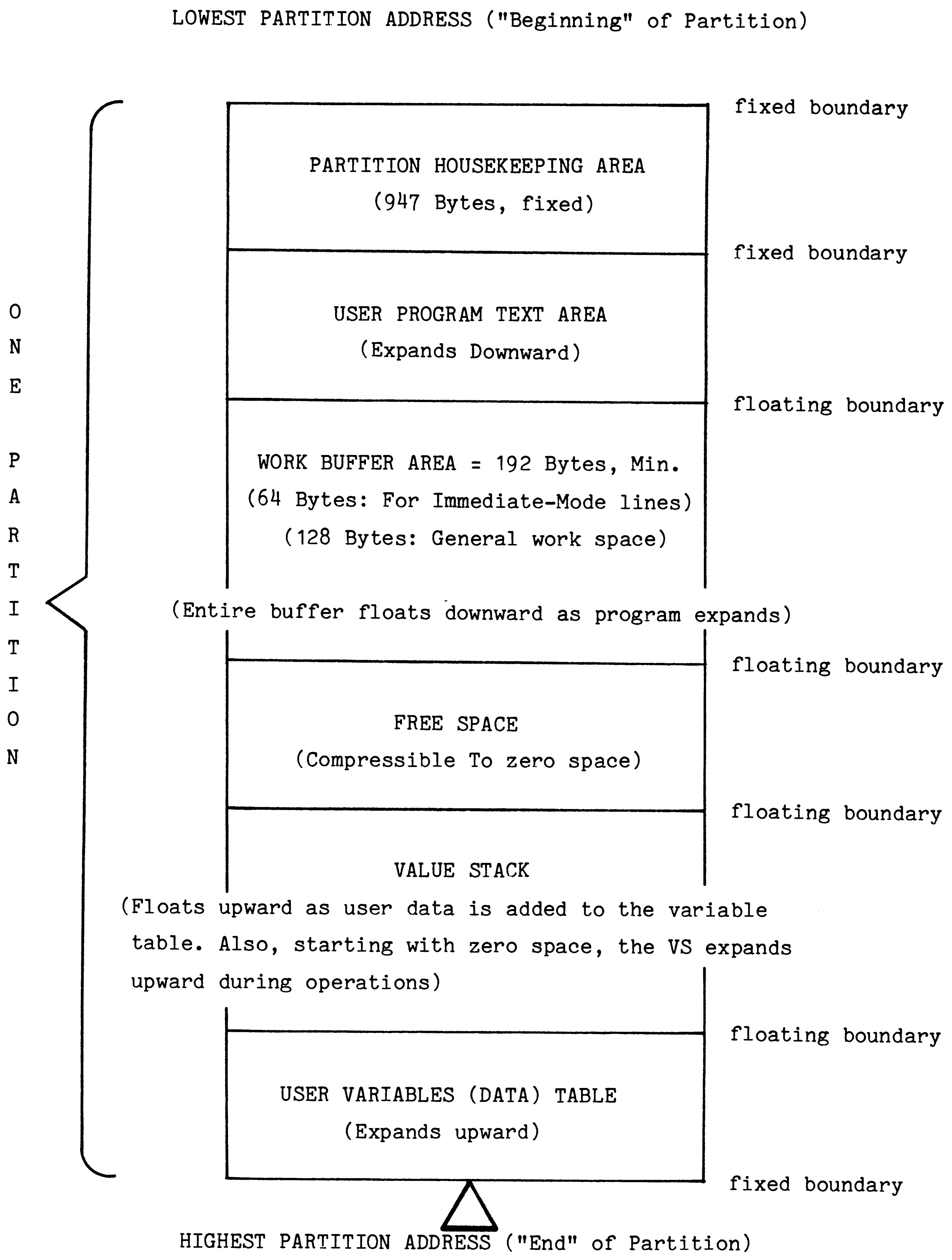

LOWEST PARTITION ADDRESS ("Beginning" of Partition)
fixed boundary
PARTITION HOUSEKEEPING AREA
(947 Bytes, fixed)
fixed boundary
USER PROGRAM TEXT AREA
(Expands Downward)
floating boundary
WORK BUFFER AREA = 192 Bytes, Min.
(64 Bytes: For Immediate-Mode lines)
(128 Bytes: General work space)
(Entire buffer floats downward as program expands)
floating boundary
FREE SPACE
(Compressible To zero space)
floating boundary
VALUE STACK
(Floats upward as user data is added to the variable table. Also, starting with zero space, the VS expands upward during operations)
floating boundary
USER VARIABLES (DATA) TABLE
(Expands upward)
fixed boundary
HIGHEST PARTITION ADDRESS ("End" of Partition)
ONE PARTITION

FIGURE 2-2 INTERNAL PARTITION ALLOCATIONS

## 2.4 THE SERVICING OF PARTITIONS

### 2.4.1 TIME-SLICE PROCESSING

The LVP CPU services each partition (max.= 16) in a repetitive, ordered sequence. Each partition is given a standard 30 ms. "time slice", during which exclusive use of the CPU is granted. A limited number of program or immediate-mode operations can be executed during this interval. For this purpose, the CPU has a 30-millisecond timer which is set at the beginning of each timeslice; this clock is checked periodically for expiration of the 30-ms limit. For reasons which will be explained in subsequent text of this section, note that time slices are not always allowed to last the full 30 ms.

When a partition's time slice ends, the Operating System saves all current status conditions for that partition. The Operating System then proceeds to load the status of the next partition into the CPU and begins a new 30-ms time slice. The exact moment when execution is halted in a partition is called the "breakpoint" of the time slice. The programmer cannot predict in advance when a breakpoint will take place, but the occurrence of breakpoints is of little or no concern to him. Further, since the ordered time slice arrangement is repeated at high speed, all user programs appear to operate simultaneously.

Whenever a partition is given a new time slice, conditions that existed at the end of that partition's previous time slice will be restored, and processing for that application resumes for the duration of the new time slice.

### 2.4.2 BREAKPOINTING

As previously mentioned, a time slice does not always last exactly 30 milliseconds. Unlike many operating systems, the LVP Operating System will cause breakpoints whenever it is convenient or advantageous, rather than only allowing breakpoints to occur upon expiration of the the CPU time-slice clock. Specifically, under the direction of the Operating System, a breakpoint may occur if a peripheral device being addressed is busy, or if the device being addressed is being "hogged" (explained later) by another partition; either condition is called an "I/O breakpoint".

For instance, if the partition that has the current time slice attempts a disk access, and if the disk is temporarily being "hogged" (used exclusively) by another partition, the hogging condition is quickly detected and a breakpoint occurs in the current partition time slice.

The term "I/O breakpoint" should not be confused with "program breakpoint". Program breakpoints are conditional, scheduled halts in a user's program; they are a means, for instance, of monitoring an I/O port for pending data entry requests. Program breakpoints are written into the user's program by the user.

I/O breakpoints differ from program breakpoints in that the partition interrupted by an I/O breakpoint is specifically marked "waiting for I/O". When that partition is given another time slice, the Operating System takes only microseconds to decide whether I/O processing may proceed or whether the partition is still waiting for the I/O device and must therefore be bypassed. The Operating System temporarily bypasses that partition as effectively as if it had been entirely removed from the system during the I/O waiting period.

The CPU is much faster than any of its peripherals, and for this reason, breakpointing during I/O allows the LVP to perform work with other partitions while the I/O operation is still being carried out. For example, when a program uses KEYIN to receive data from a keyboard, the CPU can give time slices to other partitions between operator keystrokes. In a similar manner, several partitions can be serviced by the CPU during a carriage return on a 2221W printer.

## 2.5 ASSIGNMENT, ATTACHMENT, AND FOREGROUND/BACKGROUND PROCESSING

### 2.5.1 ASSIGNMENT

Although system resources must be shared in the 2200LVP, each user is given the impression of having his own personal system--his own terminal, his own memory, his own peripherals. As previously explained, the exact configuration of each user's "personal system" is specified by an operator at partition generation time. Partitions and terminals configured into one such "personal" system are said to be assigned to each other; they belong to each other as integral parts of an independently-functioning personal computer system.

Assignment alone is only a prerequisite for the actual operation of each "personal" system. In order to use any facility of the system, attachment is required.

## 2.5.2 ATTACHMENT

"Attachment" is a state that exists when the Operating System establishes an active bidirectional communications link between a partition and a terminal that have previous assignment to one another. Unless attachment occurs, a user has no access to the LVP central processor. Without attachment, the user terminal is dumb, having no program mode, no immediate mode. At any given time, only one attachment is possible in each user's "personal" system configuration. Attachment with the lowest-numbered assigned partition occurs automatically on completion of @GENPART (ref: $INIT statement in the BASIC-2 Language Reference Manual, WL#700-4080).

To illustrate the states of assignment and attachment, consider the following:

Suppose that a program (arbitrarily called "program A") requires frequent operator interaction; another program, "B", belonging to the same user, requires only minimal interaction. The preliminary requirement is that of assignment. The two partitions (one with program "A", the other with program "B") and the user's terminal must have been previously assigned to each other by the Operating System, that they might function as an integral unit, a "personal system". So that program "A", the priority real-time program, can function on an interactive basis with the user terminal, the next requirement is that of attachment. The Operating System moves the partition holding program "A" into the "foreground". By this action, the Operating System attaches the user terminal and partition to one another. For the duration of each subsequent time slice given to the foreground (attached) partition, both program and user can communicate with one another, and both have access to the CPU. Also by time-slice processing, program "B" runs "simultaneously" in the background, communicating with the CPU, communicating with certain peripherals; however, since this partition is running in the background, it is unable, for the moment, to interact with its assigned terminal.

When a background partition (program) attempts to communicate with its assigned terminal, and if that terminal is currently in a state of attachment with another assigned partition, execution of the background program is suspended ("hangs") until the requested terminal is released (detached) from the foreground partition and is attached to the requesting partition.

Note that some background jobs may have no requirements for access to a terminal other than periodic display of current job status. To avoid having such jobs "hang" while awaiting availability of the terminal, the $IF ON statement can be used to determine if the terminal currently is attached to the partition. If $IF ON finds that the terminal is attached, the status information is displayed; if not, the program branches to perform further processing before testing for availability of the terminal again.

## 2.6 "RELEASING" A TERMINAL

"Release" of a terminal from a state of attachment is accomplished by executing the BASIC statement $RELEASE TERM in either the program mode or the immediate mode. Once a terminal has been released from a foreground partition, the assignment that existed between the terminal and the partition is still recognized and maintained by the Operating System. Further, when a $RELEASE TERM is executed, the foreground partition is moved immediately to the background by the Operating System. Simultaneously, the Operating System establishes a new state of attachment between the terminal and the lowest-numbered waiting (suspended, assigned) background partition. Of course, the partition selected for attachment is then considered to be in the foreground. Note that the term "background" implies only assignment; "foreground" implies both assignment and attachment.

All waiting background partitions may have a need for the terminal within their assignment; however, each of the assigned background partitions (programs) is sequentially given access to the terminal (i.e., is brought into the foreground for attachment) only when:

1) The program operating in the foreground partition executes a $RELEASE TERM statement; this means that the terminal is released, in the program mode, to the next-highest-numbered waiting partition. (Special case: If the $RELEASE TERM statement is executed in the highest-numbered assigned partition, the terminal is given to the lowest-numbered waiting partition in the assignment.)

OR When:

2) The user executes a $RELEASE TERM statement in the immediate mode; the terminal is released to the next- highest-numbered waiting partition. (Due to the fact that the processing order of partitions is repeated by the CPU, if the $RELEASE TERM statement is executed when the highest-numbered partition is in the foreground (attached), the terminal is given to the lowest-numbered waiting background partition in the assignment.)

If there are no assigned partitions actually waiting for a terminal after it been released, it is possible for the user at that terminal to request the Operating System to re-establish a state of attachment between his terminal and one of the partitions assigned in his "personal" system. This is accomplished by keying either RESET or HALT on the terminal. On that signal, the Operating System moves the user's lowest-numbered assigned background partition to the foreground, HALTs or RESETs any program operating in that partition, and then establishes a state of attachment between the terminal and the new partition.

<u>NOTE</u>:

In order to allow re-attachment, and in order to prevent the halting or resetting of an active background program, it is a good practice to generate a small control or "dummy" partition as the lowest-numbered assigned partition.

Optionally, the user himself may direct the swapping of terminal/partition attachments by executing a modified form of the $RELEASE TERM statement ($RELEASE TERM TO) in either the program mode or the immediate mode. A partition is named in the TO parameter, and that partition must, of course, be a partition that already shares assignment with his terminal. When a $RELEASE TERM TO statement is executed, the terminal is placed in attachment with the specified partition, even if that partition has not attempted to communicate with the terminal, and even if one or more other assigned partitions have attempted to communicate with the terminal. $RELEASE TERM TO does not halt the execution of programs running in either the current foreground partition or the target background partition specified in the TO parameter.

## 2.7 "RELEASING" A PARTITION

Release of a parititon from a state of attachment is accomplished by executing the BASIC statement $RELEASE PART in either the program mode or the immediate mode. A partition may also be considered "released" if, at partition generation time, an operator specifies terminal #0 (a non-existent terminal, sometimes called the "null" terminal) for any terminal/partition assignment in the system. Another term used in place of "released partition" is "available partition". In any case, the flag which signifies that a partition is released (i.e., available) is the terminal #0 assignment. A released partition does not belong to any user's "personal system"; it has no terminal associated with it; it has no terminal assignment. Note that if a program is running in a released partition, execution of that program will "hang" if any communications are attempted with a terminal.

The $RELEASE PART statement allows a partition to become available to any terminal connected to the system.

Consider the following:

1) If a terminal is in a state of attachment with some partition, and if that partition does not meet requirements for some new application (due to insufficient partition size, for instance), the operator may elect to use an "available" partition more suited to his needs. The characteristics of available partitions may be examined by executing a $PSTAT statement. When the available partition is found having characteristics most suited to the operator's needs, the user may then execute a $RELEASE TERM TO statement to the available partition; the newly-acquired partition will then be given a new assignment with the requesting terminal, and will be placed in a state of attachment with that terminal. Thus, the new partition becomes a new addition to the user's "personal" system.

2) An operator at a non-assigned terminal may also request assignment and attachment to a released (available) partition by keying RESET or HALT.

$RELEASE PART causes a present states of attachment and assignment between a terminal and a partition to be broken off. The terminal formerly belonging to that assignment can optionally be re-directed to a new partition for assignment and attachment (if a new assignment is specified in parameters of the $RELEASE PART statement). This carries the implication that, in addition to making a partition available, $RELEASE PART also performs a $RELEASE <u>TERM TO</u> for the terminal. If a new partition assignment is not specified for the terminal in the parameters of $RELEASE PART, that terminal will either be attached to a waiting partition already within the assignment (if there is one waiting), or the terminal will have no further assignment or attachment with any partition. In the latter case, the terminal becomes non-assigned, having no immediate mode, no means of executing programs, no access to system peripherals; it would no longer be part of the active LVP system.

Note that $RELEASE PART does not clear a partition, nor does it terminate a program running in that partition.

## 2.8 "GLOBAL" PARTITIONS

Partitions can also be "global"; that is, each partition so designated contains programs and/or data which become conditionally shareable. A foregound or background program that is running in a partition in one bank can access any global partition (i.e., global routine and/or global data) residing in that same bank. Additionally, a user terminal that is in a state of attachment with a partition in that same bank can access those global routines and/or data while in the immediate mode.

Although partitions function independently, there are situations in which it is highly expedient for two or more partitions to cooperate with one another, to share common information, common programs. This sharing eliminates needless duplication of applications software and data, thus allowing more efficient use of available User Memory space.

## 2.9 "UNIVERSAL GLOBAL" PARTITIONS

The first 5K of User Memory in bank #1 (immediately following the System-Use Block) constitutes a special section of User Memory known as the "universal-global" area. Partitions defined within this area are correspondingly called "universal-global partitions". A universal-global partition may be accessed by a program running in any foreground or background partition. Also, similar to standard global access, user terminals in a state of attachment are allowed access to universal-global routines and/or data while in the immediate mode. To summarize, a universal-global partition can be used to store programs and data that can be shared by all system users.

Note that the entire 5K universal-global area need not be used exclusively for universal-global partitions; the only restriction is that, for a partition to be universally global, it must reside entirely within the 5K universal-global address block in bank #1. When not required for universal-global purposes, that same 5K in bank #1 can be treated as all other partitionable memory.

## 2.10 USER PROGRAM EXECUTION

### 2.10.1 GENERAL

The term "job flow" refers to the path of execution followed by a job from beginning to end. In the 2200LVP, job flow may be confined within a single partition, or it may extend across several partitions via global subroutine calls. The term "job" is preferred to "program" here, because the term "program" is too closely associated with the contents of a single partition. A job consists of one or more program routines; each line of each routine in the job contains one or more program statements. In the normal execution of an individual routine, each statement is executed from left to right, from lowest line number to highest.

### 2.10.2 SUBROUTINES

The Operating System tracks execution of a job by using a "text pointer". The text pointer always points to the statement that is to be executed next in a particular job flow; the text pointer provides a "thread" leading from the statement currently being executed to the statement that is about to be executed.

If job execution is confined within a single partition, the text pointer contains all information required by the Operating System for the execution of a user's program. However, to execute global subroutines, the Operating System requires additional information that reveals which partition contains the currently-executing program text.

When a global subroutine call is made, the global text is executed as if that text were appended to the calling text within the originating partition. The "job" may therefore be thought of as the combination of all nonglobal and global program text, considered as a integral unit.

The "originating partition" is the partition in which the job is initiated; further, it is the partition that holds all status information pertinent to the execution (flow) of that job, even if that job extends across several partitions. Each job has only one "originating partition".

"Calling partition", which may be familiar to some readers, is simply a partition making the current global/universal-global subroutine or data call in a multi-partition job.

When a user program issues a non-global, global, or universal-global subroutine call (or requires global/universal-global variables), the status and return-address information for each successive subroutine level is stored in the originating partition sequentially. If a time slice expires while execution is taking place in an originating partition, or if the time slice is terminated by the occurence of a breakpoint, or if the time slice ends while execution is taking place in a called global/universal-global routine, the conditions of execution that exist at the moment the time-slice ends are also stored in the originating partition.

In order to track all of the various conditions that arise during subroutine calls, each partition has two internal "stacks" and a "pointer table"; users are not allowed access to these housekeeping elements. The CPU and the Operating System service each partition, and in the process, monitor, use, and update each pointer, each stack. Note that the text pointer for each job is maintained within the originating partition's pointer table.

## 2.10.3 TEXT POINTER, POINTER TABLE, & INTERNAL STACKS

Typically, when a subroutine call is issued (for instance, by a GOSUB' statement), the number of the statement following the GOSUB' becomes the current value in the text pointer. Simultaneously, the same number is saved on top of the value stack, one of the internal stacks previously mentioned in this discussion.

NOTE:

The value stack functions as a "push-down, pop-up" storage element. (The last, most recent entry in the value stack will be the first to be recalled at any given time by the Operating System.) The value stack can also be thought of as a "last-in, first-out" or "LIFO" storage element.

The Operating System searches the program for a DEFFN' that corresponds to the GOSUB' just issued. The statement number at which the DEFFN' is found becomes the a current value in the text pointer. The Operating System instantaneously passes execution to that point in the program.

The number in the value stack is unchanged; it is still the statement number following the GOSUB'. When a RETURN statement is executed in the subroutine, the Operating System retrieves the "old" text pointer entry from the top of the value stack. That entry is placed in the text pointer (in the pointer table), thus replacing the DEFFN' statement number, and then the Operating System passes execution back to the statement which immediately follows the GOSUB statement.

## Pointer Table Format

The following illustrates basic Pointer Table format:

| | |
|---|---|
| Text Pointer | |
| Text Partition # | |
| Data Partition # | |
| Global Partition # | |
| Current Partition # | |
| Terminal # | |

Basically, each text pointer consists of a line number and a statement number. For example, consider the following line of program text:

```
10  A = 100:  PRINT A
```

In line #10, when the statement "A = 100" is executed, the text pointer is automatically incremented to point to the next statement in that line, "PRINT A". Thus, during execution of the statement "A = 100," the text pointer would have the value "10,2", indicating that the next statement to be executed is the second statement in line 10.

Initially, all items in the Pointer table refer to the current partition. For example, immediately following Master Initialization, a system configuration could be established such that Partition #2 (in a state of assignment with Terminal #4 for this arbitrary example) would have the following values in its pointer table.

| | |
|---|---|
| Text Pointer | 0,0 |
| Text Partition # | 2 |
| Data Partition # | 2 |
| Global Partition # | 2 |
| Originating Partition # | 2 |
| Terminal # | 4 |

The last two items in the table, Originating Partition# and Terminal#, are constants that are set during Master Initialization. These values do not change unless the system is reconfigured; other items in the table can be modified frequently during job execution. The meanings and uses of each item in the pointer table follow:

The Text Pointer - is updated by the Operating System, each time a statement is executed, to point to the next sequential statement. Further, it is modified by any branch statement (GOSUB, GOTO, GOSUB', etc.), in order to point to the branched-to statement.

The Text Partition # - is the number of the partition to which the text pointer applies (i.e., it is the number of the partition containing the currently-executing text). It is modified by a GOSUB' statement whenever a branch is made to a DEFFN' in a global partition. In this case, GOSUB' sets the Text Partition# equal to the Global Partition#.

<u>The DATA Partition #</u> - is the number of the partition containing DATA statements referenced by READ. The DATA Partiton# can be modified by a RESTORE statement, which always sets that number equal to the current Text Partition#.

<u>The Global Partition #</u> - is the number of the currently- selected global partition. It is modified by a SELECT @ PART statement. It is the partition searched by GOSUB' for a corresponding DEFFN' when the DEFFN' cannot be found in the Text Partition. It is also the partition used for all global variable references.

<u>The Originating Partition #</u> - is the number of the partition in which execution of the job originates and the Pointer Table is stored. The Originating Partition # is a constant for each partition. It is used for all local variable references, for LOAD operations, and for all system commands issued from the user terminal. The Originating Partition # is returned by the #PART function.

<u>The Terminal #</u> - is the number of the terminal that is in a state of assignment with the originating partition. Like the Originating Partition #, it is set at configuration time and generally is not modified, except by reconfiguring the system. (Terminal # can be altered upon execution of a $RELEASE PART statement.) Terminal # is used for all CRT, keyboard, and local printer I/O operations performed during job execution; this includes CO, CI, PRINT, LIST, INPUT, LINPUT, KEYIN, etc. For any partition, the Terminal # is returned by the #TERM function.

## 2.11 ALLOCATION AND HANDLING OF PERIPHERALS

### 2.11.1 GENERAL

The mental image of multiple partitions and terminals functioning as completely independent "personal systems" may be clouded somewhat by the problem of competition (between partitions) for shared peripheral devices ("system peripherals"). This situation is familiar to programmers accustomed to working with single-user Wang 2200 systems that share one or more disk drives via disk multiplexers. In such systems, it is sometimes necessary for one CPU to request exclusive control of a disk (i.e., to "hog" the disk) while a file update is conducted.

With the 2200LVP, it may be necessary for a partition to exclusively control a printer. For example, if, during a report printout, a printer were not exclusively available to one partition, that partition's print lines might become unintelligibly mixed with those of another partition's, if both were allowed access to one system printer at the same time. To solve this problem, the concept of disk hog mode has, in the LVP, been extended to all shared I/O devices ("system peripherals").

To state the situation more specifically: prior to configuration of the system through $INIT, and with the exception of user terminals and local printers, peripherals connected directly to 2200 I/O controllers are available to all partitions i.e., such peripherals are "sharable". This implies, further, that printers connected to terminals would not be considered "shareable". A conflict arises when more than one user partition simultaneously attempts access to a shareable device.

In order to avoid such situations, the LVP Operating System enables a partition, under program control, to request exclusive use of a peripheral with a $OPEN statement; the address of that peripheral must be specified in that statement. Once "open", the device remains hogged by the requesting partition until either a $CLOSE or an END statement is executed or if a CLEAR, RESET, or LOAD RUN command is initiated. Thus, if a disk is "hogged" by the $OPEN statement, only the user who executed that statement may read or write disk files until the device is released by one of the above prescribed methods.

With the exception of terminals and local printers connected to them, all peripherals connected to the system must be specified in the Master Device Table at partition generation time. Using the Device Table, a device can be placed in exclusive assignment with a specific partition until a new system configuration is generated. This method involves use of the SELECT statement with its various options (Ref: 2200VP/MVP Language Manual; WL# 700-4080).

Basically, peripheral assignments are established at partition generation time by the entry of a number--the number of the partition which is to have control of a particular device--in the "Master Device Table". Such entries are carried out indirectly by the Operating System during the execution of @GENPART. Console device addresses--i.e., HEX 005 (CRT), 001 (Keyboard), 204 (terminal printers)--are not specified in the Master Device Table; these are specified in a partition device table. Each partition, in fact, has its own local device table which should not be confused with the Master Device Table; the partition device table keeps track of console devices in a user's "personal" system configuration.

If any partition attempts to use a shareable device that has not been allocated to it during @GENPART (i.e., use of that peripheral device was not specified in the Master Device Table), an error is signalled.

## 2.11.2 BACKGROUND PRINTING

As an additional feature of the LVP system, if a printer is connected to the rear apron of an "assigned" terminal (thus making the printer an assigned "local printer"), it is possible for a background program to send output to that printer while a foreground program simultaneously interacts with the keyboard and display of the attached terminal. The only requirement for background printing is that the terminal to which the local printer is connected must be in a state of assignment with both the foreground and the background partition. The simultaneous I/O required for this type of action is handled by the 2236MXD controller and the 2236DE firmware (PROM's).

## NOTES

# SECTION 3

# BOOTSTRAP OPERATION

## 3.1 BOOTSTRAP

A BOOTSTRAP, by definition, is "that part of a computer program used to establish another version of the computer program."

In general, the Wang LVP BOOTSTRAP, is a set of microcoded routines loaded in three 1024 x 8-bit Intel 2708 PROMs. The purpose of the BOOTSTRAP is to handle four system functions and make available certain subroutines which are used for I/O operations.

The four system functions handled by BOOTSTRAP are:

1) Master Initialization (Power-On).
2) Reset (Initiated by depressing the RESET key on the keyboard).
3) Control and Data Memory Parity Error Detection.
4) Loading the desired system software (i.e., standalone diagnostics, or BASIC-2) from disk and initiating their execution.

An explanation of each of the above functions follows.

### 3.1.1 MASTER INITIALIZATION

Master Initialization begins by turning the CPU power switch to the ON position. A branch to Control Memory address 8003 (HEX), located in the BOOTSTRAP PROMs, is executed and the BOOTSTRAP routine begins controlling and performing its various tasks.

The tasks performed by the Master Initialization routine in BOOTSTRAP are:

a) Exercise the CPU to determine if any obvious malfunctions exist.
b) Verify the BOOTSTRAP PROMs still maintain the desired data.
c) Write zeros to all locations in Data Memory in preparation for subsequent Data Memory Reads.

If all Master Initialization tasks are completed satisfactorily, the following prompt will be displayed at the system console:

```
MOUNT SYSTEM PLATTER
PRESS RESET
```

## 3.1.2 RESET

Reset is initiated by depressing the RESET key on the terminal keyboard. This action causes the execution of a branch to Control Memory address 8001 (HEX), located in BOOTSTRAP PROMs.

The tasks performed by Reset are:

(a) To pass control, from the present point of program execution, to the currently loaded system program, located in Control Memory (BOOTSTRAP, Microcode diagnostics, or BASIC-2).

(b) To allow the user to recover from any of the various system error conditions which may be encountered.

(c) To abort a BOOTSTRAP load.

Should task a) be called for, the user may expect those messages and/or actions designed into the particular system program. Activation of RESET would typically result in the occurrence of a display menu of user-selectable software options (key Special Function), or, for instance, an automatic return to some predetermined starting point in the software currently resident in Data Memory.

Generally speaking, whenever task b) is to be performed, the user is expected to inform the BOOTSTRAP of what action to take (by keying a Special Function, for instance).

### 3.1.3 CONTROL AND DATA MEMORY PARITY ERRORS

In both Data and Control Memory a bit has been set aside, called the parity bit, to aid in error detection.

In Control Memory, bit 24 is set aside for parity; it should be turned on by the programmer whenever an even number of the remaining 23 bits are turned on. (This is called ODD Parity.) This bit must be properly set when the microprogram is written.

In Data Memory, a ninth bit is set aside for parity; it is turned by the hardware whenever an even number of the 8 data bits are turned on. (This is also ODD parity.) The hardware determines and sets this bit, whenever data is written into Data Memory.

Whenever the system detects bad parity in Control Memory, during an instruction fetch, a branch is made to location 8000, located in the BOOTSTRAP PROMS. The BOOTSTRAP will then display the appropriate error message at the system console.

Similarly, whenever the system detects bad parity in Data Memory, during a read from Data Memory, a branch is made to location 8002, located in the BOOTSTRAP PROMS. The BOOTSTRAP will then display the appropriate error message at the system console.

### 3.1.4 LOAD SYSTEM FILES

Whenever the operator responds to the BOOTSTRAP request for a system file to be loaded, the following tasks are performed by the BOOTSTRAP.

a) Check for disk ready.

b) Verify whether the user-requested file exists on the mounted platter.

c) Determine whether the requested file should be loaded into Control Memory and/or Data Memory, and then load the file.

d) Verify Control Memory, checking instruction parity and built-in CRC and LRC checksums.

e) Check Data Memory Parity.

f) Pass control to the newly loaded system file.

## 3.2 BOOTSTRAP ERROR MESSAGES AND RECOVERY

Three types of errors and five possible error messages can be reported by BOOTSTRAP. The three error types--initalization, reset, and system--are discussed below.

### 3.2.1 INITIALIZATION ERRORS

The BOOTSTRAP, during Master Initialization, fails to display the complete

```
MOUNT SYSTEM PLATTER
PRESS RESET
```

message upon the CRT.

This error implies that some routine of the LVP BOOTSTRAP has failed. This may be due to either a CPU-related error or an I/O-related error. If an initialization error occurs, refer to SECTION 12.

In some cases, a device address may need to be corrected and the system powered on again.

The Master Initialization sequence is described on the following pages.

## MASTER INITIALIZATION
Step-By-Step Breakdown of Function

| CRT DISPLAY | SEQUENCE OF OPERATIONS | POSSIBLE FAILURES |
|---|---|---|
| | 1. Power On Trap to 8003 | 1. Hardware Trap, Branch Instruction |
| | 2. Enable CRT, Clear Screen and Display "M" | 2. CRT Address, I/O Register, I/O Lines, CIO Instruction |
| CLEAR SCREEN | | |
| "M" | | |
| | 3. Test 24-Bit Parity Trap. Execute IC 800F which has Bad Parity | 3. Parity Checking Logic, Hardware Trap, TSP Instruction (IC+1 stored in stack), PC's may not hold IC retrieved from Stack, Compare Instruction |
| "MO" | | |
| | 4. Test Subroutine Branch and Subroutine Return Instructions | 4. Subroutine Branch Instruction, Subroutine Return Instruction, Stack |
| "MOU" | | |
| | 5. Clear CH, CL Parity Bits | 5. Write/Read Data Memory |
| "MOUN" | | |
| | 6. Check File Registers | 6. Register Instruction, Register Chip, Compare Instruction |
| "MOUNT" | | |
| | 7. Check PC Incrementing on the A-BUS. | 7. PC Chip, LPI Instruction, Register Instruction, A-Bus Increment Hardware, Compare Instruction |
| "MOUNT S" | | |
| | 8. Test Auxiliary Registers | 8. Auxiliary/Stack Chip, PC Chip, Auxiliary Register Instruction, Compare Instruction |

MASTER INITIALIZATION
Step-By-Step Breakdown of Function
(Continued)

| CRT DISPLAY | SEQUENCE OF OPERATION | POSSIBLE FAILURES |
|---|---|---|
| "MOUNT SY" | 9. Test Binary ALU | 9. Binary ALU, AC, ACX, AI, SC or SCX Instruction, Compare Instruction |
| "MOUNT SYS" | 10. Test Stack. | 10. Auxiliary/Stack Chip, PC Chip, Stack Instruction, Compare Instruction |
| "MOUNT SYST" | 11. Test Decimal ALU | 11. Decimal ALU, DAC, DACI, DACX, DSC, DSCI or DSCX Instruction, Compare Instruction |
| "MOUNT SYSTE" | 12. Test Binary Multiply | 12. Multiply Logic, M OR MI Instruction, Compare Instruction |
| "MOUNT SYSTEM" | 13. Test Shift | 13. Shift Logic, Compare Instruction |
| "MOUNT SYSTEM " | 14. Verify PROM | 14. PROM Chip |
| "MOUNT SYSTEM P" | 15. Zero 8-Bit Data Memory | 15. SR Failure, Bad IC's |
| "MOUNT SYSTEM PLATTER" "PRESS RESET" | 16. Write/Read Control Memory | 16. WCM/RCM Instruction, Stack, Auxiliary Register, PC Chip, SB Instruction, Compare Instruction |

## 3.2.2 RESET ERRORS

If the hexdigit display of the keyed special function did not appear upon the CRT, during the Reset function, when the operator has properly responded to the "KEY SF'?" message by keying the desired special function key:

This implies that the special function key was not depressed sufficiently, or the 2236DE or 2236MXD may be defective, or an SF' key not defined was depressed. If a RESET error occurs, refer to SECTION 12.

NOTE:

During the RESET function, several of the SYSTEM ERROR messages may appear. If one does, consult the recovery procedure for that particular message, given in Section 3.2.3.

The system reset sequence is described on the following pages.

SYSTEM RESET
Step-By-Step Breakdown of Function

| CRT DISPLAY | SEQUENCE OF OPERATIONS | POSSIBLE FAILURES |
|---|---|---|
| | 1. Reset keyed while BOOTSTRAP is in control | 1. Reset Trap |
| CLEAR SCREEN | | |
| "KEY SF'?" | | |
| | 2. Enable Keyboard (address = 01) and accept Special Function key input; operator keys the desired SF key.<br><br>NOTE: if any undefined SF' key is despressed, the "KEY SF" message re-appears and step 2 must be repeated. | 2. Inactive SF is keyed,<br>I/O Register,<br>I/O Lines,<br>CRB or KFN,<br>Keyboard |
| *"KEY SF'?" (address) | 3. Enable specified disk | 3. Improper disk address,<br>I/O Register,<br>I/O Lines,<br>Disk Not Powered On,<br>Disk Not Ready |
| | 4. Search disk for desired file; if file cannot be found, Step 2 is repeated | 4. Wrong Special Function key depressed,<br>Wrong disk mounted |
| | 5. Load desired file into Memory<br><br>NOTE: System files should contain a comment block containing file date.<br>If a disk error results, the system error message will appear. Consult Error Recovery, for proper procedure. If a parity error occurs during loading, 'P' will be displayed and the previous sector will be reloaded. If no control memory data is found, skip to step 9. | 5. I/O Register,<br>I/O Lines,<br>Disk Problems |

*The name of the file to load and the platter to load from is displayed.

## SYSTEM RESET
Step-BY-Step Breakdown of Function
(continued)

| CRT DISPLAY | SEQUENCE OF OPERATIONS | POSSIBLE FAILURES |
|---|---|---|
| "KEY SF'?" (address) "COMMENT" | | |
| | 6. Verify Control Memory. (Parity, LRC & CRC). If an error results, the system error message will appear. Consult Error Recovery, for proper procedure. | 6. Memory, WCM/RCM Instruction |
| | 7. Check 8-Bit Data Memory. If an error results, the system error message will appear. Consult Error Recovery, for proper procedure. | 7. Memory, Read/Write Instruction |
| | 8. Control is passed to loaded system file which now takes over control. Consult proper system file documentation. (Address = 3000). | |
| | 9. Display Diagnostic Menu listing upon CRT. | |
| "KEY SF'?" | | |
| | 10. Enable Keyboard (address = 01) and accept Special function key input. Operator keys the SF key of the desired diagnostic | 10. Inactive SF is keyed, I/O Register, I/O Lines, CRB OR KFN, Keyboard |
| "KEY SF'?" (address) | | |
| | 11. Go to Step 4. | |

## 3.2.3 SYSTEM ERRORS

The third grouping of error conditions is reported to the operator via a SYSTEM ERROR message on the CRT.

First, should memory fail, the following message will appear:

```
*** SYSTEM ERROR MMMM XXXX ***
PRESS RESET
```

```
where: MMMM = PECM--Parity Error Control Memory
              PEDM--Parity Error Data Memory
              VECM--Verify Error Control Memory
              VEDM--Verify Error Data Memory
       XXXX = Various error information pertinent to the type of error.
```

Secondly, a disk error will result in the following message being displayed:

```
*** SYSTEM ERROR DISK 00XX ***
PRESS RESET
```

where: 00XX = is the Disk Error Code

The procedure used to recover from these SYSTEM ERRORS is similar. Therefore, the general procedure will be outlined and each error will be discussed.

The general procedure is:

a) Key RESET in response to the "PRESS RESET" message on line 2 of the CRT.

b) Choose one of the four following courses of action.

1. Key SF'15 to resume, using the currently loaded system program (usually BASIC-2).

2. Key SF'00-'05, '08-'013 to load BASIC-2 from disk 310, B10, 320, B20, 330, B30, 350, B50, 360, B60, 370 or B70.

3. Key SF'16-'19 to load the User diagnostic menu from disk 310, B10, 320, or B20, respectively.

4. Key SF'28-'31 to load the Field Service diagnostic menu from 310, B10, 320, or B20, respectively.

Use special caution when you choose #1 above: depending on what type of error and where it occured, BASIC-2 may not function properly in all cases.

The following discussion will outline each of the SYSTEM ERRORS and what may be done, in particular, to recover from them. (Also refer to SECTION 12 if a system error occurs.)

## 3.2.3.1 CONTROL MEMORY ERRORS

In both Data Memory and Control Memory, one bit has been set aside for parity error detection.

In Control Memory, the 24th bit (bit #23) of every microinstruction is set aside for parity (it is turned ON whenever an even number of the remaining 23 bits turns on). This is called ODD Parity. This bit must be properly set when writing the instruction into Control Memory.

```
*** SYSTEM ERROR (PECM aaaa dddddd) ***
```

| | | |
|---|---|---|
| Where: | aaaa = | The address of the instruction with bad parity. |
| | dddddd = | The instruction located at aaaa. The instruction is reread when displayed and thus may not be the same as when the error occurred. |

This error implies that bad parity was detected while the system was trying to execute an instruction from Control or BOOTSTRAP Memory.

Whenever the system detects bad parity in Control Memory (PECM message) during an instruction fetch, a branch is made to Control Memory address 8000 (HEX), located in the BOOTSTRAP PROMs. The BOOTSTRAP then performs its designated error routine and displays PECM aaaa, dddddd.

Bad parity may be the result of:

a) dropping of bits by Control/BOOTSTRAP Memory
b) picking up of bits by Control/BOOTSTRAP Memory
c) writing bad parity to Control Memory
d) defective parity-checking logic

This error should be serious enough to warrant the executing of a Control Memory diagnostic. However, it may be possible to resume execution of the currently loaded system program. If the error is reported again, a Control Memory diagnostic should be run to locate the defective memory chip.

*** SYSTEM ERROR VECM aaaa ***

| | | |
|---|---|---|
| Where: | aaaa = | An address in the section of Control Memory that does not verify correctly. |

<u>Case 1 (aaaa = 0000 thru 7FFF)</u>

This error implies that the load of Control Memory from the disk was not successful. However, bad memory locations cannot be entirely ruled out.

This error is reported prior to a system program being given control and is the result of the program not being loaded properly into Control Memory.

The operator should attempt to reload that particular system program. However, should successive failures be reported, a Control Memory diagnostic should be run to determine if there are any bad memory chips. If no chips are reported defective, a CPU instruction may be failing, requiring a CPU diagnostic to be run.

Should the error be reported in low memory (i.e., address between 0000 and OFFF) it may be necessary to change memory boards in order to load the diagnostic into memory.

<u>Case 2 (aaaa = 8000 thru 83FF)</u>

This error implies that the BOOTSTRAP Memory is not as expected.

This error may be caused from dropping or picking up bits by one or more of the three PROMs that make up the BOOTSTRAP.

Try to power on again, and if the problem still persists replace the BOOTSTRAP PROMs and perform a MASTER INITIALIZATION. If the error continues, the board may have failed or in some cases a microinstruction may have failed.

### 3.2.3.2 DATA MEMORY ERRORS

In Data Memory, a ninth bit allocated for each 8-bit byte is used in the same manner as described above. However, the CPU hardware determines the required state and sets this bit whenever a write is executed in Data Memory.

*** SYSTEM ERROR (PEDM ss.aaaa)***

| | | |
|---|---|---|
| Where: | ss = | Memory bank containing the error (00 = bank #1; 40 = bank #2; 80 = bank #3; C0 = bank #4) |
| | aaaa = | Data memory address (i.e., the current value of the PC's) at the time of the error. This is probably, but not necessarily, the address of the memory location with bad parity. |

This error implies that bad parity was detected during a read of Data Memory.

Whenever the system detects bad parity in Data Memory (PEDM message) during a read from Data Memory, a branch is made to Control Memory address 8002 (HEX), located in the BOOTSTRAP PROMs. The BOOTSTRAP then performs another error routine and displays PEDM ss.aaaa.

Bad parity may be the result of:

a) dropping of bits in Data Memory
b) picking up of bits in Data Memory
c) defective parity checking logic

This error should be serious enough to warrant the executing of a Data Memory diagnostic. However, it may be possible to resume execution of the currently loaded system program. If the error is reported again, a Data Memory diagnostic should be run to locate the defective memory chip.

`*** SYSTEM ERROR (VEDM ss.aaaa)***`

| | | |
|---|---|---|
| Where: | ss = | Memory bank containing the error (00 = bank #1; 40 = bank #2; 80 = bank #3; C0 = bank #4) |
| | aaaa = | Address of the data in error |

This error implies that the area of data memory used for system constants (verb tables, match constants, messages), was not loaded properly when BASIC-2 was loaded. However, bad memory locations cannot be entirely ruled out.

This error is reported prior to a system program being given control. The operator should attempt to reload BASIC-2. However, should successive failures be reported, Data Memory Diagnostics should be run to determine if there are any defective memory chips.

### 3.2.3.3 DISK ERRORS

`*** SYSTEM ERROR DISK 00XX ***`

There are several possible DISK errors that may occur while BOOTSTRAP is trying to load a particular system program. The only recovery procedure that should be taken is to attempt to reload the particular system program.

The possible disk errors are:

DISK 0082

Error: File not in catalog

Cause: The file to be loaded does not reside on the platter specified.

Recovery: Make sure that the proper platter is properly mounted, that the proper disk drive was specified, and that the proper special function key was pressed. Press RESET, as prompted, and select the appropriate special function.

DISK 0088

Error: Wrong record

Cause: Occurs during a load when the format of the record read does not conform to the bootstrap format.

Recovery: Make sure that the proper platter is properly mounted, the the proper disk drive was specified, and the proper special function key was pressed. Press RESET, as prompted, and select the appropriate special function.

DISK 0090

Error: Disk Hardware Error

Cause: The disk did not recognize or properly respond to the system at the beginning of a read or write operation (the read or write has not been performed).

DISK 0091

Error: Disk Hardware Error

Cause: A disk hardware error occurred; i.e., the disk is not in file-ready position. This could occur, for example, if the disk is in LOAD mode or power is not turned on.

Recovery: Ensure that the disk is turned on and properly set up for operation. Set the disk into LOAD mode and then back into RUN mode, with the RUN/LOAD selection switch.

DISK 0092

Error: Disk Hardware Error

Cause: The disk did not respond to the system at the beginning of a read or write operation in the proper amount of time (time-out). The read or write has not been performed.

Recovery: Run program again. If error persists, reinitialize disk.

DISK 0093

Error: Disk Format Error

Cause: A disk format error was detected during a disk read or write. The disk is not properly formatted. The error can be either in the disk platter or the disk hardware.

Recovery: Format the disk again.

DISK 0094

Error: Format Key Engaged

Cause: The disk format key is engaged (the key should be engaged only when formatting a disk).

Recovery: Turn off the format key.

DISK 0095

Error: Seek Error

Cause: A disk-seek error occurred; the specified sector could not be found on the disk.

Recovery: Run program again. If the error persists, reinitialize (reformat) the disk.

DISK 0096

Error: Cyclic Read Error

Cause: A cyclic redundancy check error occurred during a disk read operation; the sector being addressed has never been written to or was incorrectly written.

Recovery: If the disk has been formatted, rewrite the bad sector or reformat the disk.

DISK 0097

Error: Longitudinal Read Error

Cause: A longitudinal redundancy check error occurred when reading a sector.

Recovery: Make sure the SYSTEM PLATTER is properly mounted in the operator specified disk unit. Key RESET, as prompted, and try to reload. If the error persists, try a backup platter.

DISK 0098

Error: Disk Addressing Error

Cause: The disk sector being addressed is not on the disk.

Recovery:

a) Make sure that the disk is ready and the SYSTEM PLATTER is properly mounted in the operator specified disk unit. Key RESET, as prompted, and try to reload.

b) If the problem persists, then BOOTSTRAP may be bad or the disk may have a problem.

## NOTES

NOTES

# SECTION 4

# SYSTEM GENERATION

## 4.1 GENERAL

<u>NOTE</u>:

Any future changes in the Operating System that affect the system generation procedure will be documented in category IV.C.4.

When the 2200LVP is powered on, an operator at terminal #1 has the responsibility to "Master Initialize" the system and to load/execute the partition/peripheral configuration suited to the current application(s).

The process of Master Initialization (loading the BASIC-2 Operating System) creates a preliminary single-partition system that is controlled exclusively from terminal #1. No devices connected to the system--other than terminal #1 and the system disk--are available until total system configuration takes place. Configuration is performed either by execution of the BASIC-language system utility called @GENPART, or by the BASIC statement $INIT (discussed in later text). As a part of Master Initialization, the system microcode (BOOTSTRAP) automatically loads and runs @GENPART, which is a file stored on the system disk. If @GENPART is not on the system disk, a READY message is displayed at terminal #1.

A system configuration created by either the standard @GENPART utility or by a customized version of @GENPART (using the $INIT statement) remains in effect until the system is reinitialized. Note that @GENPART is always assumed (by the BASIC-2 Operating System) to be the name of the system generation/configuration utility, whether Wang-written or user-written.

When @GENPART is initiated, parameters from the previous configuration (called 'current') are automatically loaded. If the Wang version of @GENPART is used, a list of user-selectable options and previously-saved configurations is displayed.

On completion of Master Initialization and System Generation/ Configuration, terminal #1 functions like all other terminals connected to the LVP Central Processor. (Up to this point, terminal #1 functioned as the "system console".)

After configuring the system, at least one backup copy of the system disk should be made. By taking this step, a user might prevent system "down time" that could result from accidental damage to the original system disk. The COPY or MOVE statements are used for duplication of the system disk. (A detailed explanation of the COPY and MOVE statements is given in the Wang BASIC-2 Disk Reference Manual, WL #700-4081F (III.A.0).

## 4.2 POWER-UP, MASTER INITIALIZATION, AND SYSTEM GENERATION

The following explanation should provide the reader with enough information to power-up, Master Initialize, and configure ("generate") the system.

### 4.2.1 POWER-UP

To begin, switch AC power ON in Workstation #1 and in the Central Processor. After power is applied to the system, the prompt appears:

```
MOUNT SYSTEM PLATTER
PRESS RESET
```

The system disk contains the BASIC-2 Operating System, as well as a variety of hardware diagnostics. When the disk drive achieves the ready state, steps may be taken to load the Operating System or hardware diagnostics via Special Function Keys on terminal #1.

Mount the system disk, then press the RESET key (located in the upper-right corner of the keyboard). The following prompt is displayed:

```
KEY SF'?
```

## 4.2.2 LOADING THE OPERATING SYSTEM

A Special Function Key must be depressed to specify the address of the disk drive in which the system disk is loaded.

The following options are available:

Key SF '00 to load BASIC-2 from the disk @ address 310 (Hex).
Key SF '01 to load BASIC-2 from the disk @ address B10 (Hex).
Key SF '02 to load BASIC-2 from the disk @ address 320 (Hex).
Key SF '03 to load BASIC-2 from the disk @ address B20 (Hex).
Key SF '04 to load BASIC-2 from the disk @ address 330 (Hex).
Key SF '05 to load BASIC-2 from the disk @ address B30 (Hex).
Key SF '08 to load BASIC-2 from the disk @ address 350 (Hex).
Key SF '09 to load BASIC-2 from the disk @ address B50 (Hex).
Key SF '10 to load BASIC-2 from the disk @ address 360 (Hex).
Key SF '11 to load BASIC-2 from the disk @ address B60 (Hex).
Key SF '12 to load BASIC-2 from the disk @ address 370 (Hex).
Key SF '13 to load BASIC-2 from the disk @ address B70 (Hex).

<u>NOTE</u>:
Normally, the fixed-disk drive is assigned address 310 (HEX), and the DSDD diskette drive (removable) is assigned address B10 (HEX).

Approximately 15 seconds are required for the BASIC-2 Operating System to be loaded into Control Memory. While this takes place, the following message will appear on the display screen of terminal #1:

```
Loading:  2200LVP BASIC-2 Release X.X
```

When loading is complete, the system displays the "READY (BASIC-2) PARTITION 01" message, unless the @GENPART partition-generation program is resident on the system disk. If such is the case, the @GENPART Partition Generator is automatically loaded after the BASIC-2 Operating System is loaded. (The @GENPART data file is normally on the system diskette.) Terminal #1 should then be ready for limited use, the other terminals are enabled only after configuring the system as desired with @GENPART or $INIT.

If the wrong SF Key is depressed (i.e., if the system disk is mounted at address 310, but the operator depresses SF Key 02), an error message will be displayed:

```
*** SYSTEM ERROR (DISK 00XX) ***
PRESS RESET
```

Recovery from such errors may be accomplished by simply pressing RESET, followed by the correct Special Function key. If RESET fails, turn the Central Processor OFF then ON again. If this latter step is required, Master Initialization will be repeated.

In some instances, the Special Function key code is displayed. This may indicate that an incorrect disk address was specified, or that a disk controller has failed. Check the controller address, or replace the controller if that board is suspected to be defective.

## 4.2.3 PARTITION GENERATION

Configuration parameters must now be passed to the Operating System. As stated previously, the @GENPART program is automatically loaded and executed when it is resident on the system disk (no operator intervention required). If such is the case, immediately following Master Initialization (RESET, KEY SF'?) the @GENPART menu will be displayed at terminal #1, instead of the READY message. (The "READY (BASIC-2) PARTITION 01" message will appear once @GENPART has finished execution.) If so desired, the user may elect to customize the BASIC language @GENPART program, thus providing more suitable display prompts (etc.) for his specific needs.

Basically, using either method of partition generation (@GENPART or $INIT), the operator at terminal #1 has control over the following (explanations follow in subsequent text):

--Number of partitions
--Size of each partition
--The terminal associated with each partition
--The "programmability" of each partition
--The "bootstrap" program for each partition
--Addresses of the peripherals connected to the system
--Access to peripherals
--The "system message"

### Standard Partition Generation:

The standard Wang "@GENPART" program has two important provisions for user convenience:

1) If partition-generation modules have been previously defined, a list of those module names will be displayed on the @GENPART menu screen. The user can select and load one of these modules using the following procedure:

a) First, type in the name of a previously-saved configuration module, then press RETURN.

b) Depress Special Function key '15, causing the system to begin execution with the presently loaded partition-configuration module.

2) If the user wishes to define a new partition module, he can do so by depressing any of the other Special Function keys; this action initiates partition generation.

NOTE:

It may be useful to depress the large FN (HELP) key in the upper-left part of the workstation key pad; descriptive information will be automatically provided on the screen that explains the partition generation process. (Depress the RETURN key to see successive screenloads of instructions.)

When the BASIC-2 Operating System is fully loaded, the @GENPART menu should appear:

```
           *** WANG 2200MVP PARTITION GENERATION PROGRAM ***

                                                  LIST OF OPTIONS:
LIST OF STORED CONFIGURATIONS (#PARTITIONS)    SF'00 - clear partitions
              current         ( X )            SF'01 - clear device table
                                               SF'02 - divide mem. evenly
                                               SF'04 - edit partitions
                                               SF'05 - edit device table
                                               SF'06 - edit $MSG
                                               SF'08 - load configuration
                                               SF'09 - save configuration
                                               SF'10 - delete configuration
                                               SF'15 - execute
                                                  FN - help

Configuration 'current' loaded.  Name of configuration to load? ________
```

## DESCRIPTIONS OF @GENPART SPECIAL-FUNCTION OPTIONS

SF' 00 - clear partitions:

Clears partition-configuration parameters currently in memory, allows the user to specify the total number of terminals and the total number of partitions in each bank, then automatically advances to SF'04 (Edit Partitions). The Master Device Table (ref: SF' 05 - edit device table:) is not altered when this function is selected. Any number of partitions between one (1) and sixteen (16) that will not exceed the available memory capacity is allowable. (Note that since the smallest each partition can be is 1.25K (16 partitions, max.) and since there is a 3K Operating System overhead space to account for, the minimum User memory size that would accommodate 16 partitions is (1.25K x 16 partitions) + 3K = 23K. (The nearest RAM size to this, physically available, is 32K.)

SF' 01 - clear device table:

Clears Master Device-Table parameters currently stored in memory, resets default peripheral addresses to HEX 215 (printer), 310 (primary disk/disks), and 320 (secondary disk/disks), allocates these devices to all users (specifies common access), then advances to SF'05 (Edit Device Table). (Default device addresses can be edited, if necessary, using SF' 05.)

SF' 02 - divide mem. evenly:

Divides remaining User Memory equally among the number of partitions specified with SF' 04.

SF' 04 - edit partitions:

Displays and allows editing of partition parameters such as size, terminal assignment, programmability, and name of bootstrap program. SF'04 does not allow addition or deletion of defined partitions in an existing configuration. Descriptions of edit functions follow:

Number of partitions:

From one (1) to sixteen (16) partitions may be created.

## Size of partitions:

Any size greater than--or equal to--1.25 kilobytes is allowable. This specification is made in 256-byte (1/4K) increments. The maximum allowable size is 61K (64K minus 3K for housekeeping) in bank #1, and 56K (64K minus 8K that is not user-accessible) in bank #2.

## The terminal associated with each partition:

Any terminal number from 0 to 5 is valid; terminals 1 to 5 are the actual user-terminals connected to the system; terminal number 0 is a non-existent "dummy" or "null" terminal. All partitions must have a terminal assignment, even if the 0 (null; non-existent) terminal is specified, and even if there are partitions that will contain "background jobs" that, practically speaking, never print on the CRT or require keyboard entry. In general, any singular partition may be placed in assignment with any singular terminal; however, a singular terminal may be specified to be in assignment with several partitions, in order to create a multiple-partition "personal" system. In general, the lowest-numbered partition(s) to be placed in a state of assignment with a terminal should contain the foreground (interactive) jobs for that terminal. Background jobs should be placed in the higher-numbered partitions within that assignment. Only the terminal that has been specified to be in a state of assignment with a particular partition can list or modify the program in that partition. Finally, note that while it is possible for partitions to access global program text and modify global variables, it is not possible for non-global partitions to list or modify program text in a global or universal-global partition.

## Programmability of partitions:

Any partition can be specified for the "disabled programming" mode, whereby that partition is inhibited from certain operations. Terminals attached to "disabled programming" partition(s) are inhibited from entering or modifying program text, or from performing certain other system operations. Thus, the operator is prevented from inadvertent or unauthorized use of protected or restricted programs and data.

## Bootstrap programs for partitions:

Any program that resides on the system disk can be loaded into a partition and run automatically when a configuration is executed. When no bootstrap program is specified for a partition, the 'READY' display will appear on the CRT once the configuration has been executed.

## SF' 05 - edit device table:

Displays and allows editing of device addresses for all peripherals. All peripherals connected directly to I/O controllers must be specified in the Master Device Table, which is located in the System Overhead section of memory, (this, of course, excludes terminals and local printers connected to them). Console device addresses (i.e. HEX 005--CRT, 001--keyboard, 204--local printers) are not specified in the Master Device Table, nor may they be specified using SF'05; these are specified in each partition device table, which is located in the Partition Overhead section of memory. Partition Device-Table specifications and modifications are discussed later in this section.

By default, all system peripheral devices listed in the Master Device Table are available to all partitions. However, devices can be given exclusive assignmment with one partition until the next system configuration is executed. This is accomplished by entering, in the Master Device Table, the number of the partition that is to have control of the selected device. For disk controllers that respond to more than one address, only the primary address must be specified in the Master Device Table (i.e. HEX 310 but not B10, 350, 390, etc.). For all other multi-address controllers, all valid addresses must be listed.

## SF' 06 - edit $MSG:

Displays and allows editing of a user-defined broadcast message that will be displayed on each terminal's CRT whenever the READY message is displayed. The user-defined message is displayed on line 0 of the CRT, immediately above the "READY" message.

## SF' 08 - load configuration:

Loads a named configuration from the Configuration File, which is located on the system disk. To modify and/or execute any previously-defined configuration other than 'current', this option must be used.

## SF' 09 - save configuration:

Save a system configuration in the Configuration File under a user-specified name (up to eight characters in length). If the user specifies a configuration name already used, @GENPART will verify that the user desires to replace the old configuration on disk file with the configuration currently in memory.

## SF' 10 - delete config.:

Deletes a configuration from the Configuration File on the system disk.

## SF' 15 - execute:

Allows the operator to review first, and then to execute, a configuration. This configuration will be automatically saved in the Configuration File under the name 'current' when the configuration is executed. Once a configuration has been executed, the system may be reconfigured again only after the Master Initialization procedure has been repeated.

## FN - help:

Displays @GENPART operating instructions.

## 4.2.4 GENERATING A SAMPLE CONFIGURATION

The following example illustrates how, typically, @GENPART can be used to configure a system. In this example, a 2200LVP with 128K bytes of User Memory, three terminals, and telecommunications option are to be configured. The configuration (named "SAMPLE") will have four partitions. A 15K-byte telecommunications program will be designated for automatic bootstrapping, as a background job sharing terminal #1. Disabled programming will be specified for this partition so that it cannot be modified inadvertently. Remaining memory will be divided equally among the other three partitions.

In general, the order of executing @GENPART options is: (1) SF'08--to load a configuration, (2) SF'00--to modify this configuration by adding or deleting partitions, (3) SF'04--to create the new partition parameters, (4) SF'05--to create the Master Device Table, (5) SF'06--to create the broadcast message, (6) SF'09--to save the configuration with a name other than 'current', and (7) SF'15--to execute the configuration. Therefore, in the example that follows, these options are discussed in their probable order of use.

Load a configuration (SF'08)

(When @GENPART is first executed, this display occurs without pressing SF'08):

```
              *** WANG 2200LVP PARTITION GENERATION PROGRAM ***

                                                      LIST OF OPTIONS:
LIST OF STORED CONFIGURATIONS (#PARTITIONS)        SF'00 - clear partitions
                current          ( X )             SF'01 - clear device table
                                                   SF'02 - divide mem. evenly
                                                   SF'04 - edit partitions
                                                   SF'05 - edit device table
                                                   SF'06 - edit $MSG
                                                   SF'08 - load configuration
                                                   SF'09 - save configuration
                                                   SF'10 - delete configuration
                                                   SF'15 - execute
                                                      FN - help

Configuration 'current' loaded.  Name of configuration to load? ________
```

The last configuration executed (called 'current') is automatically loaded. To load any other configuration, enter its name, then press RETURN. Since, in this example, a completely new configuration is to be created, press SF'00--clear partition.

## Clear Partitions (SF'00)

The program responds with a display that requests the total number of terminals that are to be configured into the system and the number of partitions that will be created. Available User Memory is automatically calculated and displayed. Note that the 3K of Operating System overhead space in bank #1, and the 8K in bank #2 are automatically deducted from the available-memory quantity. Remaining memory is updated and displayed as memory is allocated to the partitions.

```
Available memory: 61.00 K    56.00 K
Remaining memory: 61.00 K    56.00 K

No. of terminals?
```

In this example, there will be 3 terminals; enter 3 in response to the "No. of terminals?" prompt, and then key RETURN. The following will be displayed:

```
Available memory: 61.00 K    56.00 K
Remaining memory: 61.00 K    56.00 K

No. of terminals? 3
No. of partitions in bank 1 ?
```

In this example, there will be four partitions--two in each bank; enter 2 in response to the "No. of partitions in bank 1 ?" prompt, and then key RETURN. The following will be displayed:

```
Available memory: 61.00 K    56.00 K
Remaining memory: 61.00 K    56.00 K

No. of terminals? 3
No. of partitions in bank 1 ? 2
No. of partitions in bank 2 ?
```

## Edit Partitions (SF'04)

This option displays default parameters for all partitions and initiates a cycle of prompts for the altering of these parameters. The cycle recurs until another option is selected. The user is thus allowed to modify parameters for each partition. The display is updated each time an item is entered.

```
    PARTITION SIZE(K) TERMINAL PROGRAMMABLE PROGRAM
        1        -        1        Y
        2        -        2        Y
        3        -        3        Y
        4        -        1        Y

Edit which partition (default = 1 )?
```

In this example, the telecommunications program will be run in partition #2. Begin, therefore, by editing the parameters for partition #2. Enter 2, then key RETURN. An asterisk (*) appears beside the number of the partition whose parameters are being edited, and the following series of prompts will be displayed in succession at the bottom of the screen:

```
Partition size (default = 0 K)?
```

Any value greater than 1.25K and less than the amount of remaining User Memory is a valid response. Note that the default value (zero kilobytes) is not a legal value.

The telecommunications program that is to be run in this partition will require 15K. To allocate 15K of User Memory to partition #2, enter 15, then key RETURN. The following prompt should be displayed at the bottom of the screen:

```
Terminal (default = 2 )?
```

The telecommunication program will be a background job controlled at terminal #1. To establish assignment between this partition (partition #2) and terminal #1, enter 1 and key RETURN. The following prompt then occurs.

```
Enable programming (Y or N) ?
```

By default, programming is allowed for all partitions; however, to prevent inadvertent modification of the telecommunications program, "disabled programming" will be specified for partition #2. To specify disabled programming mode for this partition, enter N, then key RETURN. The name of a program to be automatically loaded into this partition is now requested as follows:

```
Name of program to load?
```

The name of the telecommunication program that will be run in partition #2 is "TELE-COM". Enter TELE-COM and then key RETURN. When the configuration is executed, the telecommunications program will be automatically loaded from the system disk into partition #2, and will then begin running.

At this point, editing of the parameters for partition #2 is complete. Partitions #1, #3, and #4 require further modification. Remaining memory is to be divided evenly between those remaining partitions. Press SF'02 (divide mem. evenly) and the following prompt will be displayed:

```
Divide memory evenly in which bank (default = all)?
```

Key RETURN and the remaining 46K in bank #1 will be assigned to partition #1; the 56K in bank #2 will be divide evenly between partitions #3 and #4. The system returns to the initial "Edit which partition (default = 1 )?" prompt.

All that remains is to establish assignment between terminal #2 and partition #3, and between terminal #3 and partition #4. Enter these values into the table for partitions #3 and #4. Upon completion of this operation, the table should appear as follows:

| PARTITION | SIZE(K) | TERMINAL | PROGRAMMABLE | PROGRAM |
|---|---|---|---|---|
| 1 | 46.00 | 1 | Y | |
| 2 | 15.00 | 1 | N | TELE-COM |
| 3 | 28.00 | 2 | Y | |
| 4 | 28.00 | 3 | Y | |

Once all partitions have been edited, SF'05 is used to leave the "Edit Partition" cycle and then invoke the "Edit Master Device Table" option (SF'05). Note that it is legal to exit the Edit Partition Cycle (SF'04) without answering all prompts; in this case, the specified default values are used by @GENPART and the Operating System.

## Edit Device Table (SF'05)

This option displays the default values resident in the Master Device Table. Notice that by default, every device specified is available to all users.

```
            DEVICE PARTITION                  DEVICE PARTITION

         1.  /215     all                 17.
         2.  /310     all                 18.
         3.  /320     all                 19.
         4.                               20.
          .                                 .
          .                                 .
                                            .
        16.                               32.
Edit which entry (default = 1 )?
```

In this example configuration, a fourth device (telecommunications controller) is used, in addition to the three default devices. The device address of this controller is HEX 01C. To specify this device in the Master Device Table, enter "4", then key RETURN. An asterisk (*) will appear beside the number 4 in the table. Several prompts are displayed in succession at the bottom of the screen; the table is updated each time an item is edited. The user is requested to enter the device address with the following prompt:

```
Device address (default = /000, /000 to delete entry)?
```

Enter /01C, then key RETURN. Another prompt now appears, and the user is requested to specify assignment for the peripheral device with one or more partitions:

```
Allocate device to which partition (default = all)?
```

For this example, enter a "2", then key RETURN to allocate the peripheral and its controller to partition #2. This display cycle will continue, in order to allow the user to edit all entries in the Master Device Table. When the parameters for all peripheral/partition allocations have been specified, the user can select another S.F. option to exit the "Edit Device Table" mode.

Broadcast Message (SF'06)

When SF'06 is depressed, the following display occurs at the bottom of the CRT display.

```
Broadcast message:


-------------------------------------------------------------------
```

NOTE:

The system is in EDIT mode during entry of the broadcast message. While in EDIT mode, all S.F. Keys revert to their system-defined EDIT functions. The S.F. Keys cannot be used for their @GENPART-defined functions until the entry of the broadcast message is complete and the system leaves the EDIT mode.

Any message in which the number of characters and spaces does not exceed the number of dashes displayed on the CRT is valid. For this example, enter * * * THE SYSTEM WILL GO DOWN AT NOON * * *. Now key RETURN. When the broadcast message has been entered, all partition-generation parameters for the example configuration have been specified. This configuration can now be saved for later use (SF'09) or executed (SF'15). Pressing SF'09 allows the operator to save this configuration on disk under a unique name.

Save Configuration (SF'09)

When SF'09 is depressed, the following display occurs at the bottom of the CRT display.

```
Check configuration to save.  Configuration name? current
```

NOTE:

In order to save a configuration, the system diskette must be write-enabled (i.e., unprotected; the write-protect notch must be covered). If the system disk is a hard disk, note that the hard disk is always write-enabled.

The configuration currently in memory will automatically be saved under the name 'current' (if the system platter is write-enabled). However, each time a new configuration is executed, the new parameters replace the old parameters in the 'current' file. In order to save a configuration so that it can be retrieved for future use, it should be saved under a unique name. The name to be used for this sample configuration is, appropriately, "SAMPLE". Enter "SAMPLE", then key RETURN. The configuration is saved under the name SAMPLE.

## Execute Configuration (SF'15)

Once all parameters of a configuration have been defined, the system configuration can be executed. To execute a configuration, press SF'15. The configuration table will appear near the bottom of the CRT, along with a prompt requesting the operator to verify the configuration parameters to be executed.

Check configuration OK to execute (Y or N)?

If Y (RETURN) is entered, this configuration will be executed. If N (RETURN) is entered, the system returns to the beginning of the "Edit Partition" cycle (SF'04).

NOTE:

Once executed, a configuration can only be changed by first Master Initializing the system, and then, by specifying the new parameters.

Delete a Configuration (SF'10)

Since this exercise generates only a sample configuration, the configuration should be deleted, in order to save more space for actual configuration records. The following prompt will request which configuration to delete.

```
Delete which configuration? ________
```

Enter SAMPLE, then key RETURN; the configuration will be deleted from the system disk.

## 4.3 GENERATING EVENLY-DIVIDED PARTITIONS: A SAMPLE PROGRAM

Load the LVP BASIC-2 Operating System by keying the appropriate SF' key on terminal #1. Approximately 15 seconds later, the following should appear on terminal #1's display:

```
                    *** WANG 2200LVP PARTITION GENERATION PROGRAM ***

                                                          LIST OF OPTIONS:
LIST OF STORED CONFIGURATIONS (#PARTITIONS)          SF'00 - clear partitions
                 current          ( X )              SF'01 - clear device table
                                                     SF'02 - divide mem. evenly
                                                     SF'04 - edit partitions
                                                     SF'05 - edit device table
                                                     SF'06 - edit $MSG
                                                     SF'08 - load configuration
                                                     SF'09 - save configuration
                                                     SF'10 - delete configuration
                                                     SF'15 - execute
                                                        FN - help

Configuration 'current' loaded.  Name of configuration to load? ________
```

Key SF'00 to initialize all terminals and clear the partitions. The following will then appear:

```
    Available memory: 61.00 K    56.00 K
    Remaining memory: 61.00 K    56.00 K

    No. of terminals?
```

Answer the "No. of terminals?" prompt with the number of terminals on the system, then answer the "No. of partitions in bank #1?" prompt. Enter the appropriate number, then key RETURN.

```
Available memory: 61.00 K    56.00 K
Remaining memory: 61.00 K    56.00 K

No. of terminals? 3
No. of partitions in bank 1 ? 2
```

Answer the "No. of partitions in bank #2?" prompt. Enter the appropriate number, then key RETURN.

```
Available memory: 61.00 K    56.00 K
Remaining memory: 61.00 K    56.00 K

No. of terminals? 3
No. of partitions in bank 1 ? 2
No. of partitions in bank 2 ? 2
```

The "Edit Partition" screenload will automatically be displayed.

```
PARTITION SIZE(K) TERMINAL PROGRAMMABLE PROGRAM
    1        -       1          Y
    2        -       2          Y
-----------------------------------------------
    3        -       3          Y
    4        -       1          Y
```

Key SF'02 - Divide memory evenly. Key RETURN and the available memory should be apportioned equally among the number of partitions entered in the above step. The following should appear:

```
PARTITION SIZE(K) TERMINAL PROGRAMMABLE PROGRAM
    1      30.50     1          Y
    2      30.50     2          Y
-----------------------------------------------
    3      28.00     3          Y
    4      28.00     1          Y

Edit which partition (default = 1 )?
```

Finally key SF'15 (EXECUTE). A prompt will appear "CHECK CONFIGURATION. OK TO EXECUTE (Y OR N)?". Enter "Y" and key RETURN if the configuration is correct. All terminals should now display "READY (BASIC-2) PARTITION xx"; each terminal can now be used as an independent processor, a "personal" system.

## 4.4 CUSTOMIZED PARTITION GENERATION

The user may, if he so desires, write his own partition-generation utility. Further description of this approach is given below; also, refer to the 2200VP BASIC-2 Language Reference Manual, WL# 700-4080C (IV.C.2), for a detailed description of the $INIT statement.

### Streamlining the @GENPART Program:

Once initially defined and stored on disk, configuration parameters in a specified system configuration can be passed to the Operating System and executed automatically during Master Initialization, with no operator intervention. REM statements near the beginning of the @GENPART program tell the user how to streamline the program to operate in this manner.

### Use of the $INIT Statement:

When the Wang utility @GENPART does not meet a user's needs, it is also possible to create a customized configuration program using the BASIC-2 statement $INIT.

$INIT General Forms:

Program Mode Statement: (Pass initial configuration parameters to the Operating System)

```
$INIT (alpha-1, alpha-2, alpha-3, alpha-4, alpha-5, alpha-6)

Where:  alpha = literal-string
                alpha-variable
```

Immediate Mode Statement: (Reconfigure system)

```
$INIT "password"

Where:  password = System reconfiguration password; this must be a
                   literal string.
```

Once configured, the system can be reconfigured <u>only</u> by executing the $INIT "password" statement at terminal #1. Control is passed to the system bootstrap; the message

```
MOUNT SYSTEM PLATTER
PRESS RESET
```

is displayed, and the system can be loaded and reconfigured as if it had just been powered-up. In order to protect against inadvertent reconfiguration, <u>$INIT can be executed at terminal #1 only</u>.

Additionally, reconfiguration is password- protected. An error results if the proper password is not included in the immediate-mode $INIT command and reconfiguration does not occur. The default password is "SYSTEM"; thus, the operator on terminal #1 would enter:

```
$INIT "SYSTEM"
```

in order to pass control to the system BOOTSTRAP. The password can be changed via the 'alpha-6' parameter in the $INIT program statement (explanation follows). The password can be from 1 to 8 characters in length. However, if the system is powered off, or if an immediate mode $INIT is executed, the password reverts back to "SYSTEM".

The user need not be concerned with the complex form of $INIT, unless a customized partition-generator program is required. It is recommended that the Wang-supplied utility, "@GENPART," or a modified version of it be used for configuring the system, to ensure that the proper configuration parameters are passed to the Operating System. If $INIT parameters are not properly set, the system may be erroneously configured, produce unpredictable errors, and/or lock out all terminals. In order to restore operation following any of these error conditions, it may be necessary to power the CPU off and on (reinitialize the system).

Configuration parameters are defined as follows:

alpha-1 = size of each partition.

Length of string = 17.

Size = binary value indicating number of 256-byte pages of memory allocated for a partition.

Byte 1 = size of partition 1.

Byte 2 = size of partition 2.

•

•

Byte n = size of partition n.

Byte n+1 = HEX (00).

alpha-2 = terminal number for each partition.

Length of string = 16.

Terminal number = (in binary) of terminal assigned to a partition.

Byte 1 = terminal number for partition 1.

Byte 2 = terminal number for partition 2.

•

•

Byte n = terminal number for partition n.

Remaining bytes must = HEX (00).

alpha-3 = partition modes.

Length of string = 16.

Mode, bit 01 = 1 if and only if programming is not allowed on this partition.

Mode, bit 02 = 1 if and only if a program is to be bootstrapped into this partition.

Byte 1 = mode of partition 1.

Byte 2 = mode of partition 2.

•

•

Byte n = mode of partition n.

alpha-4 = bootstrap program name for each partition.

Length = 128 bytes.

Bootstrap program name = 8-byte literal-string specifying the program to be automatically loaded and run after partition generation.

1st 8 bytes = bootstrap name for partition 1.

2nd 8 bytes = bootstrap name for partition 2.

•

•

Nth 8 bytes = bootstrap name for partition n.

alpha-5 = device table.

Length of string = 99.

A device is specified by 3 bytes.

1st byte, low 4-bits = device-type (disk must be 3 or B).

2nd byte = physical device-address.

3rd byte = number of the partition for which the device is to be opened (0 if none).

1st 3 bytes = device specification for device 1.

2nd 3 bytes = device specification for device 2.

•

•

Nth 3 bytes = device specification for device n.

(N + 1) 3 bytes = $000000_{16}$

alpha-6 = reconfiguration password.

Length of string = 8

1st eight bytes are the password.

Example of Valid Syntax:

```
$INIT "SYSTEM"
10 $INIT (S$,T$,M$,N$(),D$)
20 $INIT (S$,T$,M$,N$(),D$, P$)
```

## 4.5 COPYING THE SYSTEM DISK

1. Be certain that the write-protect notch on the Operating System diskette is uncovered (write-disabled) and insert the diskette into the DSDD drive.

2. If the fixed-disk drive has not been formatted, do so by keying:

```
SELECT DISK B10 (RETURN)       -- selects diskette drive
LOAD RUN "@FORMAT" (RETURN)    -- loads format utility program from
                                  the Operating System diskette
```

3. Answer all screen prompts to format the fixed-disk drive at address 310 (HEX). (Formatting takes approximately 10 minutes.)

4. When formatting has been completed, enter

```
COPY RF (RETURN) or MOVE RF (RETURN)
```

   to create a backup copy of the system diskette on the fixed-disk.

5. Remove the Operating System diskette from the DSDD drive and insert a blank diskette (WL #177-0070) in its place. (Ensure that the write-protect notch on the new diskette is covered (write-enabled).

6. If the blank diskette has not been formatted, do so by keying:

```
RUN (RETURN)  -- runs the format utility that is already loaded
                 in CPU memory
```

7. Answer all screen prompts to format the diskette at address B10 (HEX). (Formatting takes approximately 1.5 minutes.)

8. When formatting has been completed, enter

```
COPY FR (RETURN) or MOVE FR (RETURN)
```

   to create a backup copy of the Operating System on diskette.

## 4.6 MODIFYING DEVICE TABLE ENTRIES

Master Device Table Modifications:

Refer to the EDIT DEVICE TABLE function (SF'05) in the @GENPART discussion given earlier in this section.

Partition Device Table Modifications:

Device Table entries can be modified either explicitly, with a SELECT statement, or implicitly with a CLEAR command, the RESET key, or Master Initialization of the system. In general, therefore, Partition Device Table entries remain in effect until one of the following operations is performed:

--A SELECT statement is executed explicitly redefining one or more specified entries

--A CLEAR command with no parameters is executed

--The system is Master Initialized (see below);

Whenever necessary, the Partition Device Table can be displayed for debugging purposes by using the BASIC-2 statement LIST DT (List Device Table).

LIST DT displays, in hexadecimal notation, the device table belonging to the partition that is attached to the requesting terminal. The Partition Device Table is displayed at the requesting terminal. More detailed information concerning partition device-table modifications can be found in the 2200VP BASIC-2 Language Reference Manual, WL#700-4080C (IV.C.2).

## 4.7 SPECIAL PROGRAMMING CONSIDERATIONS

### 4.7.1 TIME-DEPENDENT SOFTWARE

1. The execution time of a given program varies from one machine to another. Execution on the LVP depends upon the current load of the CPU.

2. 2236DE CRT refresh speed is much slower than in 2226 CRTs. Thus, programs written to update the entire screen may affect the operating speed of the system.

3. LINPUT rather than KEYIN is recommended for data entry, since response time with KEYIN will vary, and LINPUT requires no CPU processing between keystrokes.

4. Using FOR/NEXT loops for delaying, (e.g., maintaining a message on the screen for a specified amount of time) uses excessive CPU time. Delay time varies depending upon the current work load of the CPU. Use of the SELECT P statement is recommended.

5. Instrumentation that is critically timed by the program may not work properly.

### 4.7.2 PERIPHERALS

1. For line printers, plotters, 2228B and any other device that must be allocated to a specified user for a period of time, new $OPEN and $CLOSE statements are provided. Other than making certain that these statements are added, the programmer need not change the body of a program.

2. All Console Input, INPUT, and LINPUT statements utilize 2236MXD controllers. Therefore, these statements may not be used with the telecommunications-control boards. This means, further, that the echo characters may not be sent to the line printer.

## 4.7.3 $GIO RESTRICTIONS

1. CBS is not issued to the 2236MXD.

2. Input not allowed from 2236MXD (i.e., console keyboard).

3. Timeouts and delays are allowed for output; however, the timeout or delay value is a minimum time. The value applies to the execution time allocated to this program; if other programs are executing, the actual delay time will be longer than specified.

4. There is an implicit timeout (with error) of 1 millisecond for input (non-MXD). A timeout of up to 10 ms can be specified.

## 4.7.4 I/O STATEMENT RESTRICTIONS

The following chart defines which devices the LVP Operating System permits the statement to communicate with. ERR #48 results when a BASIC-2 statement addresses an illegal device.

| STATEMENT OR OPERATION: | 2236DE TERMINAL KEYBOARD | 2236DE TERMINAL CRT | 2236DE TERMINAL LOCAL PRINTER | DEVICES OTHER THAN 2236DE TERMINALS |
|---|---|---|---|---|
| Console Output* | | X | X | X |
| PRINT | | X | X | X |
| PRINTUSING | | X | X | X |
| HEXPRINT | | X | X | X |
| LIST | | X | X | X |
| PLOT | | X | X | X |
| Console Input | X | | | |
| INPUT | X | | | |
| LINPUT | X | | | |
| KEYIN | X | | | X |
| $IF ON/OFF | X | | | X |
| $GIO | | X | X | X |
| SELECT ON (interrupt) | | | | X |
| Disk Statements | | | | X |

* Console Output (keystroke echo, error, END, STOP messages, and LINPUT and INPUT prompts) is always directed to the terminal CRT except for TRACE output which can be selected to another device (such as a printer).

### 4.7.5 DEFAULT DISK ADDRESS

Unlike the 2200VP, whose default disk address is always /310 after power on, the LVP's default disk address after power on is set to the address of the disk from which the system was loaded. That is

|  |  |
|---|---|
| SF'00 sets default address to | /310 |
| '01 | /B10 |
| '02 | /320 |
| '03 | /B20 |

After partition generation, the default disk address for each partition is set to the default disk address of partition #1 at the time of partition generation.

### 4.7.6 CONTINUE

The LVP supports CONTINUE as an Immediate Mode statement rather than a command. Thus, CONTINUE need not be the only statement on a line; however, no statements may follow CONTINUE on the Immediate Mode line. This feature of CONTINUE is useful when program execution is to be continued with the terminal released to another partition. For example,

```
:$RELEASE TERMINAL : CONTINUE
```

## 4.8 PROGRAMMING THE 2209A ON THE 2200LVP

The present $GIO sequences, documented in table 4-1 of the 2209A manual, will lead to an input timeout error (I92) on the LVP. The LVP cannot allow one partition to wait for an input strobe (8607) for a long time, as this would be unfair to other users. The LVP hardware does not permit the LVP to switch users once an 860X microcommand has begun, because data may be lost in the process. The solution is to wait for the tape drive controller to become ready (1020) before asking the board for input. Thus the change to the $GIO sequence is to insert a 1020 microcommand after a CBS (44xx) that causes tape motion and before the single character input (8607) that follows the tape motion commands.

As mentioned in the 2209A manual, it is not necessary to keep the tape controller board enabled throughout an entire tape operation. The example of a look ahead read is given. In the example, the $IF ON statement is an acceptable substitute for the wait for ready micro-command (1020).

```
10 #GIO READ/07B (4400 1020 8607 442A C220, A$) B$ ()
or
20 $GIO LOOK AHEAD READ /07B (4400, A$)
.
.
.
30 $IF ON /07B, 500
.
.
.
500 $GIO READ CONTROLLER BUFFER /07B (1020 8607 442A C220, A$)
```

In the previous example, $IF ON and the 1020 microcommand in line 500 are redundant.

Another important LVP change is the increased importance of Master Reset (459C). The reset key on the 2236DE console <u>WILL NOT</u> reset the tape drive controller. If a reset from the console happens to occur in the middle of the execution of a tape drive $GIO sequence, the tape drive controller will be left in an unpredictable state. In such cases, it is important that tape drive controller be reset by sending a CBS of HEX (9C) without waiting for ready (459C).

The Status $GIO sequence is currently documented as allowable at any time (CBS of 88 without waiting for ready). Experience has shown that reading controller status during tape operations sometimes interferes with proper controller operation. The status sequence should be used to read tape drive status when the tape is not in motion (448B rather than 458B). $IF ON or the $GIO micocommand 1010 should be used to test for "tape operation complete".

On the LVP, the $GIO sequence 1300 A000 is a faster multi-character output than the A200 in the present tape drive manual.

To summarize, the new recommended LVP $GIO sequence for the 2209A tape drive are listed below:

| | |
|---|---|
| Backspace file | $GIO BSF /07B (4405 1020 8607, A$) |
| Backspace record | $GIO BSR /07B (4404 1020 8607, A$) |
| Forwardspace file | $GIO FSF /07B (4402 1020 8607, A$) |
| Forwardspace record | $GIO FSF /07B (4408 1020 8607, A$) |
| Read | $GIO READ /07B (4404 1020 8607 442A C220, A$) B$() |
| Rewind | $GIO REWIND /07B (4446 1020 8607, A$) |
| Write EOF | $GIO WEOF /07B (4403 1020 8607, A$) |
| Write Gap | $GIO WGAP /07B (4407 1020 8607, A$) |
| Write | $GIO WRITE /07B (4429 1300 A000 4401 1020 8607, A$) B$() |
| Look Ahead Read (Subset of Read) | $GIO LAR /07B (4400, A$) |
| Finish Read (Subset of Read) | $GIO FR /07B (1020 8607 442A C220, A$) B$ () |
| Buffer Write (Subset of Write) | $GIO BW /07B (4429 1300 A000 4401, A$) B$ () |
| Finish Write (Subset of Write) | $GIO FW /07B (1020 8607, B$) |
| Master Reset | $GIO RESET /07B (459C, B$) |
| Status | $GIO STATUS /07B (448B, 1020 8706, B$) |

## NOTES

NOTES

# SECTION 5
# HARDWARE THEORY OF OPERATION

## 5.1 FUNCTIONAL STRUCTURE OF THE 2200LVP COMPUTER SYSTEM

Three basic components make up the 2200LVP computer system: 1) a Central Processing Unit (CPU), 2) system memory, and 3) an Input/Output (I/O) subsystem.

### 5.1.1 CENTRAL PROCESSING UNIT

The Central Processing Unit (CPU) controls the operation of the 2200LVP computer system, and is basically comprised of: work registers, an Arithmetic/Logic Unit (ALU), and control circuitry. The work registers are normally used as temporary storage areas during program execution. The ALU contains the circuitry necessary for performing all arithmetic and logical operations required for program execution. The control circuitry allows the CPU to execute a program automatically, by reading an instruction from memory, decoding that instruction, and generating the proper control signals to execute that instruction. When the execution of the present instruction is complete, the control circuitry reads in the next instruction and the process continues.

### 5.1.2 SYSTEM MEMORY

The system memory is a high-speed storage unit which is used to hold executable instructions (a program), and the data/variables required for execution of that program. Therefore, it is necessary to load the instructions into memory before they can be executed. The data/variables must also be in memory before they can be referenced by the program. Memory is also used to hold computation results. The area of memory where these results are stored is refered to as the "scratch pad".

### 5.1.3 INPUT/OUTPUT SUBSYSTEM

The Input/Output subsystem is comprised of peripheral devices used for program and data input, data output, mass storage of high-volume information, and remote CPU/memory-access. Several types of I/O devices exist. For example: magnetic tape units, magnetic disk units, printers, and plotters.

## 5.2 FUNCTIONAL STRUCTURE OF THE 2200LVP CENTRAL PROCESSING UNIT

The primary objective or function of the 2200LVP Central Processing Unit (CPU) is to fetch, decode, and execute instructions that reside in memory. In the 2200LVP, instructions are executed sequentially.

The 2200LVP CPU consists of three interconnected functional units: 1) work registers, 2) an Arithmetic/Logic Unit (ALU), and 3) control circuitry.

### 5.2.1 WORK REGISTERS

A register is a high-speed temporary storage area. Some examples of registers in the 2200LVP CPU are: the "C" Register, the Instruction Register, or the File Registers. Not all CPU registers can be accessed by an executable instruction, but instead are used to sustain the operation of the CPU. One such register is the Instruction Register, which holds the instruction presently being executed, and thereby is involved in controlling the execution of that instruction, and is not involved in the actual calculation or operation that the instruction is performing.

The LVP CPU contains two main addressing-registers: the Program Counter--which is used to address the location in (Data) memory at which data is to be read or written (stored), and the Instruction Counter--which is used to address the location in (Control) memory where the next executable instruction is located. During program execution, the content of the memory location addressed by the Instruction Counter is loaded into the Instruction Register. The Instruction Register holds the instruction presently being executed so that the control circuitry can direct the CPU through the steps necessary for performing the operation(s) indicated by the instruction.

### 5.2.2 ARITHMETIC/LOGIC UNIT

The Arithmetic/Logic Unit (ALU), which performs the necessary math and logic operations, is made up of two sections--registers, and a Logical Adder/Multiplier. The registers temporarily hold or store the values calculated by the Logical Adder/Multiplier.

### 5.2.3 CONTROL CIRCUITRY

The control circuitry can be divided into two areas. One of these areas is Timing. The processor fetches an instruction, performs the operations required, fetches the next instruction, and so forth. This orderly sequence of events requires precise timing. This timing is supplied by a free-running oscillator clock, and its support circuitry. The timing circuitry furnishes a reference for all CPU actions. One basic unit of time for the CPU is the instruction cycle, which is the amount of time required to fetch and execute a single instruction.

The second area is the Instruction Decoder circuitry, which is comprised of the Instruction Register and the Instruction Decoder. As previously mentioned, the Instruction Register stores the instruction to be executed. This instruction, through the Instruction Decoder, directs the CPU's activities during the instruction cycle. The Instruction Decoder translates the instruction into CPU actions. The timing circuitry controls the precise occurrence of these actions.

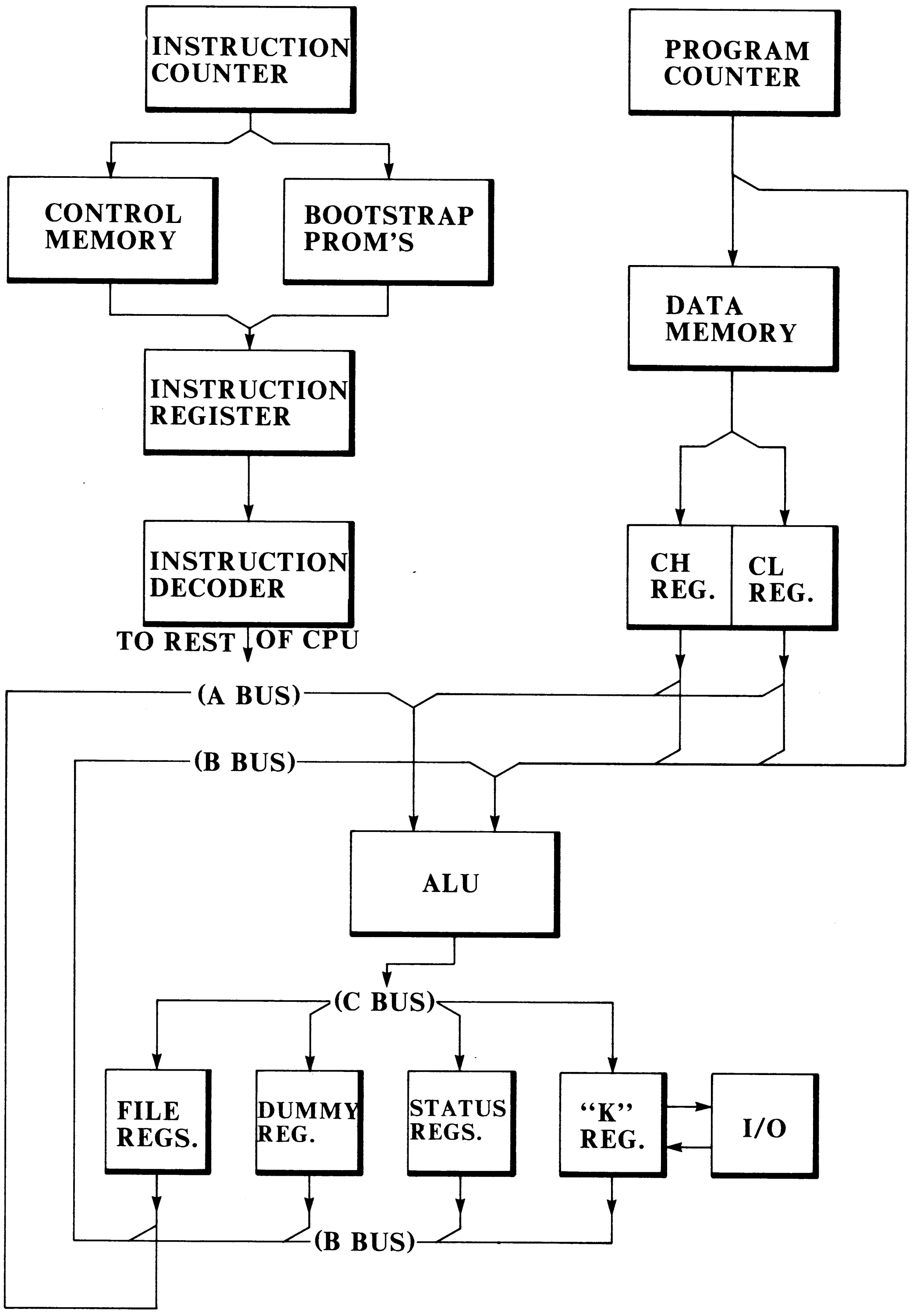


**FIGURE 5-1 2200LVP BLOCK DIAGRAM (BASIC)**

## 5.3 2200LVP CPU BLOCK DIAGRAM THEORY--BASIC (ref: FIGURE 5-1)

This section is intended to introduce some of the major concepts necessary for developing an overall understanding of the flow of information through the 2200LVP. CPU components covered in the following text are:

| | | |
|---|---|---|
| --BOOTSTRAP PROM's | --Instruction Counter | --Instruction Register |
| --Instruction Decoder | --Control Memory | --Data Memory |
| --Program Counter | --CH and CL Registers | --Dummy Register |
| --File Registers | --Status Registers | --"K" Register |
| --ALU | | |

The 2200LVP CPU is a versatile high-speed device with all the necessary functional units required for classification as a central processor. The initial instructions that control the Central Processor immediately after it is powered on are contained in the BOOTSTRAP PROM's. When power is applied to the Central Processing Unit, an initial address of 8003 (HEX) is loaded into the Instruction Counter. This action results in the CPU fetching the first instruction from the BOOTSTRAP PROM's.

The Instruction Counter points to (addresses) the location (in Control Memory) of the next instruction to be executed. The instruction addressed by the Instruction Counter is loaded into the Instruction Register. Once the instruction has been fetched, the Instruction Decoder directs the actions of the CPU through the rest of the instruction cycle.

The CPU continues to execute instructions from the BOOTSTRAP PROM's until such time as the system operator instructs the CPU to load a program (from a source such as a diskette drive) into Control Memory. Once an operating system program such as BASIC-2 has been loaded, the BOOTSTRAP PROM's transfer control (alter the Instruction Counter) to that program. The program begins execution at the appropriate location in Control Memory. The CPU always fetches and executes the next sequential instruction in Control Memory unless an instruction is decoded that directs it differently.

Data Memory, which in the LVP may be up to 128K bytes, can contain a variety of information, which is to be processed by the CPU. This information may be in the form of input data or BASIC-2 language instructions. Locations in Data Memory are addressed by the Program Counter. The information contained in Data Memory can be made available to the CPU under instruction control. When a Data Memory READ instruction is performed, the information contained in the location addressed by the Program Counter is transferred to the CH and CL Registers as directed by the instruction. To store information in Data Memory, a memory WRITE operation must be performed. The information is transferred to the Data Memory location addressed by the Program Counter.

There are several registers that are available under instruction control to allow the micro/machine language programmer to manipulate data within the CPU. One internal register of the CPU is the Dummy Register. This register is not a storage location. The Dummy Register, as its name implies, appears to be a register but is actually the source of a NULL byte, that is, the Dummy Register always contains all zeros. The Dummy Register is normally used as a a source-register for setting the contents of other registers to zero. A byte (a byte being eight bits) of information can, under instruction control, be sent to the Dummy Register. When this type of instruction is executed, the information sent to the Dummy Register is lost, because the Dummy Register, when read, always contain zeros.

The File Registers are a group of general purpose registers primarily used to hold intermediate arithimetic or logical operation results. There are eigth (8) File Registers, numbered zero (0) thru seven (7). Without File Registers, the CPU would have to write each portion of a calculation into memory and then read that partial result back from memory when the next portion of the calculation is to be performed.

The Status Registers (SH and SL) sense or indicate the state of various CPU and I/O operations. The SH Register is an eight (8) bit register that senses or sets various arithmetic, I/O, and keyboard status conditions by means of the microprogram and/or hardware. The SL Register is another eight (8) bit register which can be set only by the microprogram, and which indicates the phase of processing, mode and other conditions.

The "K" Register holds the data that is to be transfered to/from a peripheral device. The "K" Register is also used to hold the eight (8) high order bits read from or written to Control Memory.

The ALU is designed to accept two (2) eight bit binary words (from registers) as inputs to be manipulated. The two sources or inputs to the ALU are routed to the ALU via the "A" and "B" Busses. The output of the ALU is sent or routed to a register via the "C" Bus.

## 5.4 2200LVP CPU BLOCK DIAGRAM THEORY--DETAILED (ref: FIGURE 5-2)

This section explains the Detailed Block Diagram of the 2200LVP CPU. The approach taken in this section is to divide the block diagram into sections, then into blocks, and then explain the purpose of each block and its interaction with other blocks (circuits) in the section.

Prior to the description of each section is a list of the hardware components that are covered in that section.

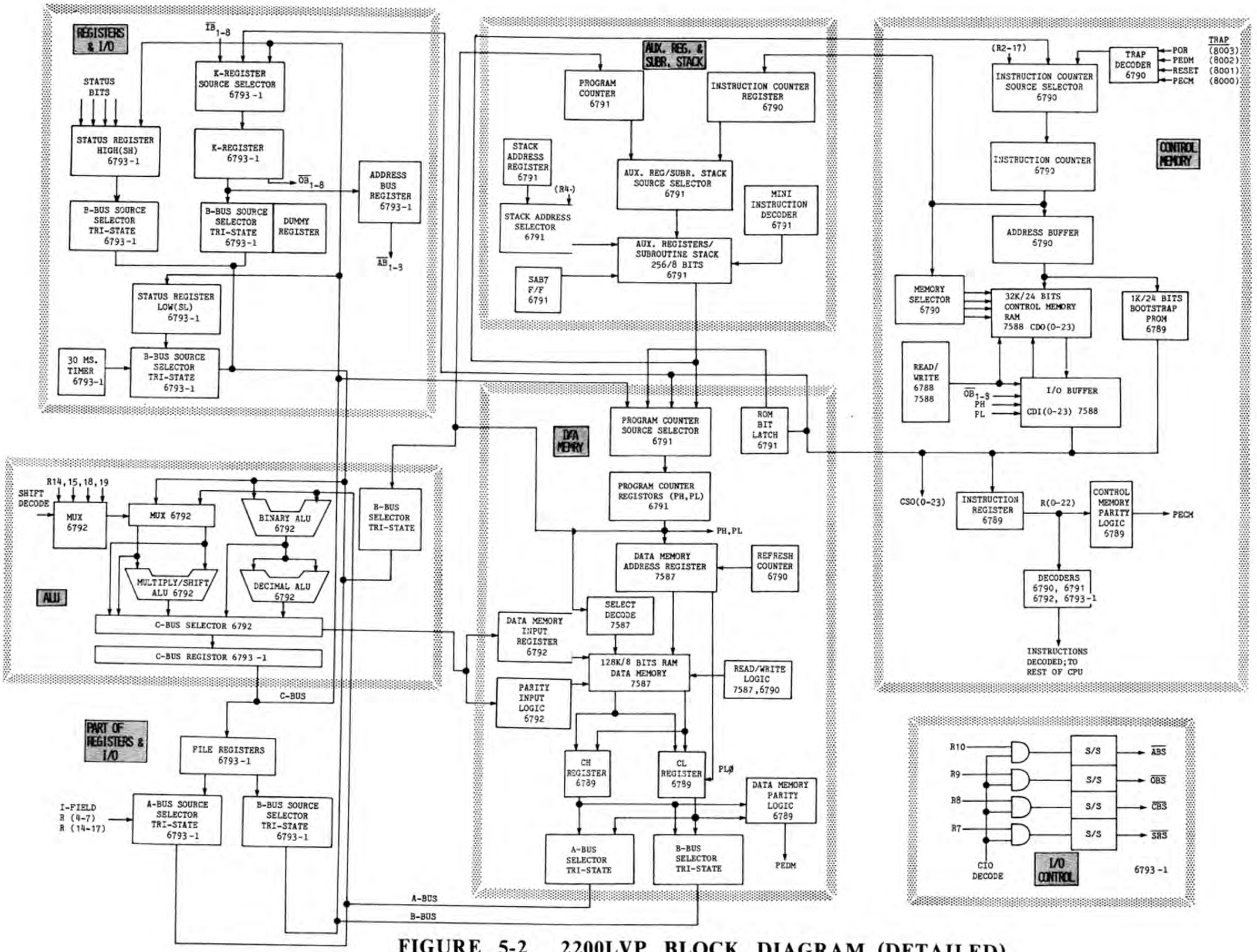

REGISTERS & I/O
$\overline{IB}_{1-8}$
STATUS BITS
K-REGISTER SOURCE SELECTOR 6793-1
STATUS REGISTER HIGH(SH) 6793-1
K-REGISTER 6793-1
$\overline{OB}_{1-8}$
ADDRESS BUS REGISTER 6793-1
B-BUS SOURCE SELECTOR TRI-STATE 6793-1
B-BUS SOURCE SELECTOR TRI-STATE 6793-1
DUMMY REGISTER
$\overline{AB}_{1-8}$
STATUS REGISTER LOW(SL) 6793-1
30 MS. TIMER 6793-1
B-BUS SOURCE SELECTOR TRI-STATE 6793-1
AUX. REG. & SUBR. STACK
PROGRAM COUNTER 6791
INSTRUCTION COUNTER REGISTER 6790
STACK ADDRESS REGISTER 6791
(R4-)
AUX. REG/SUBR. STACK SOURCE SELECTOR 6791
MINI INSTRUCTION DECODER 6791
STACK ADDRESS SELECTOR 6791
AUX. REGISTERS/ SUBROUTINE STACK 256/8 BITS 6791
SAB7 F/F 6791
(R2-17)
INSTRUCTION COUNTER SOURCE SELECTOR 6790
TRAP DECODER 6790
POR
PEDM
RESET
PECM
TRAP
(8003)
(8002)
(8001)
(8000)
INSTRUCTION COUNTER 6790
CONTROL MEMORY
ADDRESS BUFFER 6790
MEMORY SELECTOR 6790
32K/24 BITS CONTROL MEMORY RAM 7588 CDO(0-23)
1K/24 BITS BOOTSTRAP PROM 6789
READ/ WRITE 6788 7588
$\overline{OB}_{1-8}$
PH
PL
I/O BUFFER
CDI(0-23) 7588
PROGRAM COUNTER SOURCE SELECTOR 6791
ROM BIT LATCH 6791
DATA MEMORY
CSO(0-23)
INSTRUCTION REGISTER 6789
R(0-22)
CONTROL MEMORY PARITY LOGIC 6789
PECM
DECODERS 6790, 6791 6792, 6793-1
INSTRUCTIONS DECODED; TO REST OF CPU
PROGRAM COUNTER REGISTORS (PH,PL) 6791
PH,PL
DATA MEMORY ADDRESS REGISTER 7587
REFRESH COUNTER 6790
SELECT DECODE 7587
DATA MEMORY INPUT REGISTER 6792
128K/8 BITS RAM DATA MEMORY 7587
READ/WRITE LOGIC 7587,6790
PARITY INPUT LOGIC 6792
CH REGISTER 6789
CL REGISTER 6789
PLØ
DATA MEMORY PARITY LOGIC 6789
PEDM
A-BUS SELECTOR TRI-STATE
B-BUS SELECTOR TRI-STATE
SHIFT DECODE
R14,15,18,19
MUX 6792
MUX 6792
BINARY ALU 6792
B-BUS SELECTOR TRI-STATE
MULTIPLY/SHIFT ALU 6792
DECIMAL ALU 6792
ALU
C-BUS SELECTOR 6792
C-BUS REGISTOR 6793-1
C-BUS
PART OF REGISTERS & I/O
FILE REGISTERS 6793-1
I-FIELD
R (4-7)
R (14-17)
A-BUS SOURCE SELECTOR TRI-STATE 6793-1
B-BUS SOURCE SELECTOR TRI-STATE 6793-1
A-BUS
B-BUS
R10
R9
R8
R7
S/S
S/S
S/S
S/S
$\overline{ABS}$
$\overline{OBS}$
$\overline{CBS}$
$\overline{SBS}$
CIO DECODE
I/O CONTROL
6793-1

FIGURE 5-2 2200LVP BLOCK DIAGRAM (DETAILED)

### 5.4.1 CONTROL MEMORY

- --Trap Decoder
- --Bootstrap PROM
- --Memory Selector
- --Instruction Register
- --Instruction Decoders
- --Instruction Counter
- --Control Memory
- --Read/Write Control Logic
- --Control Memory Parity Logic

The Trap Decoder forces the Instruction Counter to one of four addresses (HEX 8000 to HEX 8003) when one of the following conditions exist: 1) a Parity Error is detected in Control Memory (PECM), 2) the reset button is depressed (RESET), 3) a Parity Error is detected in Data Memory (PEDM), or 4) Power-On Reset is initiated (POR). When one of these trap address conditions occurs, the Instruction Counter Source Selector is directed to pass the trap address (8000 - 8003) on to the Instruction Counter. The Instruction Counter will then contain an address above 8000 (HEX)--32K (Decimal).

The BOOTSTRAP is located in this area of Memory (above HEX 8000). The BOOTSTRAP, which is made up of three 1K by 8 bit PROM's forming a 24-bit instruction word, contains the microinstructions that control the initial operation of the CPU. The program contained in the BOOTSTRAP PROM's performs certain diagnostic routines, displays memory parity error faults, and allows the operator to load an operating system program (from an Input/Output device such as a diskette drive) into Control Memory. Once an operating system has been loaded into Control Memory, the BOOTSTRAP transfers control to the program contained in Control Memory. Control Memory, like the BOOTSTRAP PROM's, contains 24-bit instructions. Actually, twenty-three bits comprise the instruction; the twenty-fourth bit is a parity bit.

The Memory Selector circuitry supplies memory block select signals to Control Memory. These block select signals enable a 4K (4096) block of memory. There are eight memory select lines (MS1-MS8), each selecting a 4K block of memory, thus making it possible for the LVP to access up to 32K (HEX 0000-7FFF) locations in Control Memory. The remainder of the address necessary to access the desired location in the selected 4K block of memory is supplied by the Instruction Counter.

| Memory Select | Address block (HEX) | Decimal Equiv. (DEC) |
|---|---|---|
| MS1 | 0000 - 0FFF | 0000 - 4095 |
| MS2 | 1000 - 1FFF | 4096 - 8191 |
| MS3 | 2000 - 2FFF | 8192 - 12287 |
| MS4 | 3000 - 3FFF | 12288 - 16383 |
| MS5 | 4000 - 4FFF | 16384 - 20479 |
| MS6 | 5000 - 5FFF | 20480 - 24575 |
| MS7 | 6000 - 6FFF | 24576 - 28671 |
| MS8 | 7000 - 7FFF | 28672 - 32767 |

The Memory Select Decoder also generates the signal ROMS when an address of 8000 (HEX) or above is contained in the Instruction Counter. The signal ROMS selects the BOOTSTRAP PROM's, and slows the CPU to one half its normal operating speed to compensate for the slower memory access time of PROM memories.

The Read/Write Control Logic generates the control signals (R/W) necessary to transfer information to and from Control Memory. Control Memory can be read under two conditions. One condition occurs during the normal instruction fetch cycle performed by the CPU; the other condition occurs under instruction control. The micro/machine language programmer can instruct the CPU to read an instruction from Control Memory, and place it in registers K, PH, and PL.

Information can also be written to Control Memory under instruction control. Again, the micro/machine language programmer can instruct the CPU to write an instruction contained in registers K, PH, & PL into a location in Control Memory. This operation normally happens during the initialization of the system--under control of the BOOTSTRAP.

The Instruction Register holds (stores) the instruction to be executed by the CPU. The Instruction Register is loaded during the instruction fetch cycle of the CPU. After the instruction is loaded into the Instruction Register, the Control Memory Parity Logic checks for correct odd parity. If there is a parity problem, PECM is generated and the Instruction Counter is forced to the trap address of 8000 (HEX) in the BOOTSTRAP, and the location of the error in Control Memory is displayed on the "system console".

After the instruction contained in the Instruction Register has been checked for correct parity, it is up to the Instruction Decoder to control the actions or operation of the CPU to see that the instruction is executed.

## 5.4.2 DATA MEMORY

--Program Counter Source Selector
--Data Memory Address Register
--Data Memory Input Register
--Data Memory
--CH and CL Registers
--Refresh Counter
--Program Counter Register
--Select Decoder
--Parity Input Logic
--Read/Write Control Logic
--Data Memory Parity Logic
--"A" and "B" Bus Selector

The Program Counter Source Selector routes information from a specified source to the Program Counter Register. The Program Counter Register is a 16-bit register which addresses locations in Data Memory. The Program Counter Register is divided into two eight-bit registers called PH (H means high order) and PL (L means low order).

Program Counter Register bits PL 1 thru PL 7 and PH 0 thru PH 6 are passed (clocked) to the Data Memory Address Register, on the Data (RAM) Memory board, where they are used to help locate the particular piece of data requested.

The Select Decode circuitry selects the Data Memory board the information is to be read from or written to. With address lines PL 1 thru PL 7 and PH 0 thru PH 6, up to 32K locations can be selected. PH 7, which is not sent directly to the Data Memory board, is used in the generation of the Data Memory Select signals (DMS1 & DMS2).

The Data Memory Input Register holds (stores) the results of an ALU operation so that it can be written to (stored in) Data Memory.

The Parity Input Logic ensures that the data being written to Data Memory has odd parity by developing the correct parity bit (9th bit).

An LVP can have up to 128K bytes of Data Memory for holding data to be processed by the CPU. The data can be of many types. When an operator writes a User program in BASIC-2, that program resides in and will be executed from Data Memory. Also, any data that is to be processed by that BASIC-2 program will, sooner or later, reside in Data Memory. All of Data Memory, except for a small portion used by the CPU microprogram, is available for use by the operator.

The Read/Write Control Logic is similar to that used to read or write to and from Control Memory. Writing to Data Memory during Register Instructions and most Mini Instructions can be accomplished in two ways. One way is by performing a "Write 1". During a "Write 1", the information contained in the Data Memory Input Register is stored in the Data Memory location addressed by the Program Counter Register when PL0 is equal to zero.

The second way is to perform a "Write 2" which stores the information contained in the Data Memory Input register in the location addressed by the Program Counter Register when PL0 is equal to one.

When a Data Memory Read operation is performed, sixteen bits of information are transferred from Data Memory to the CL and CH Registers. Actually, two eight-bit data words are read as one sixteen-bit data word. The first word (DMO 0 - DMO 7) in the Data Memory location addressed by the Program Counter Register is loaded into the CL Register when PL0 equals zero. The second word (DMO 9 - DMO 16) in the Data Memory location addressed by the Program Counter Register is loaded into the CH Register when PL0 equals 1.

The CH and CL Registers hold information read from Data Memory so that it can be analyzed and manipulated by the CPU. Once the information from Data Memory is in the CH and CL Registers, it can be accessed by the ALU and other circuitry so that it can enter into calculations or decisions to be made by the CPU.

When the information from Data Memory is loaded into the CH and CL Registers, it, just like an instruction, is checked for correct odd parity. Information being read from Data Memory is checked by the Data Memory Parity Logic circuitry.

The Refresh Counter is necessary due to the use of Dynamic RAM memory devices in both Control and Data Memory. Dynamic RAMS must be "refreshed" periodically, row by row, or the data stored will be lost. The Refresh Counter keeps track of which row in each RAM chip is to be refreshed. One big advantage of Dynamic Memories is their fast access time.

The "A" and "B" Bus Selectors allow the CH and CL Registers to be output to either the "A" Bus or the "B" Bus.

## 5.4.3 REGISTERS

--SH Register (SH)
--"K" Register Source Selector
--Address Bus Register
--30 msec Timer
--"A" and "B" Bus Source Selectors
--SL Register (SL)
--"K" Register
--Dummy Register
--File Registers

The high order Status Register or the SH Register is an eight-bit register that senses or controls various hardware functions. (The SH Register can also be used for arithmetic/logic operations.) One example of a hardware sensing operation is: when the HALT key is depressed, SH Register bit 5 (SH5) is set. Under program control, the CPU can monitor this bit to determine whether the operator has depressed the HALT key.

| SH Register Bit | Function |
|---|---|
| SH0 | CARRY BIT (= 1 if CARRY) |
| SH1 | ENABLE/INHIBIT INPUT |
| SH1 | SFN (Special Function Key) |
| SH3 | READY/BUSY |
| SH4 | 30 MSEC TIMER |
| SH5 | HALT |
| SH6 | PEDM |
| SH7 | TRAP if PEDM (generates DMPI) |

The low order Status Register or SL Register is an eight-bit register which cannot be set by hardware. The SL Register can be modified only under program control. It can be used to indicate the CPU phase of processing, mode or other conditions. It can also be used for arithmetic/logic operations.

The "K" Register Source Selector will select, under program control, one of three possible sources for the data to be loaded into the "K" Register. The three possible sources of inputs to the "K" Register are the Input Bus, the "C" Bus, and Control Memory.

The "K" Register is another eight-bit register accessable to the CPU for arithmetic/logic operations. The "K" Register, unlike other registers, is also used to send data to and receive data from I/O devices. It is used as the I/O Input Bus (IB0-IB7), the I/O Output Bus (OB0-OB7), and holds data to be transferred to the I/O Address Bus Register (AB0-AB7). When performing read and write operations with Control Memory, the "K" Register is used to hold the eight most significant (high order) bits to be written to or read from Control Memory.

The Address Bus Register holds the I/O device address for output to the Address Bus. The Address Bus Register is an eight-bit register.

The Dummy Register supplies the CPU with a convenient register that can be used as a source for a null byte (zeros), or as a register that can receive the undesired results of an operation so that the contents of another register are not altered.

The Thirty Millisecond (30 msec) Timer is used in the 2200LVP for time sharing of the system between multiple users. The 30 msec Timer "sets" SH4 to let the operating system know that the 30 msec time slice allocated to the operators program presently being serviced has expired.

The last group of registers are the File Registers. The File Registers are temporary storage locations that can be used to hold a variety of different types of data. One use of the File Registers is storing intermediate results of arithmetic operations. There are eight 8-bit File Registers available which can be used as either source and/or destination registers for operations by the CPU.

The "A" and "B" Bus Source Selectors control the connection of the various registers to the "A" and "B" busses.

### 5.4.4 ALU

| | |
|---|---|
| --Binary ALU | --Decimal ALU |
| --Multiply/Shift ALU and Mux | --"C" Bus Source Selector and Register |

The Arithmetic Logic Unit section, known as the ALU, provides the CPU with the ability to perform arithmetic calculations and Boolean logic functions.

The ALU is basically divided into three sections: a Binary ALU, a Decimal (adjuster) ALU, and a Multiply/Shift ALU.

The Binary ALU performs the function specified by the instruction on two eight-bit source words. One eight-bit word is supplied to the ALU via the "A" Bus; the other eight-bit source is supplied to the ALU via the "B" Bus. The eight-bit data inputs are operated on by the ALU, and the results are output via the "C" Bus selector and the "C" Bus Register.

The Decimal ALU converts binary values to an equivalent BCD (Binary Coded Decimal) number. When a Decimal Add operation is performed, the ALU adds the two binary values, supplied by the "A" and "B" Busses, and then converts it to BCD.

The Multiply/Shift ALU is capable of multiplying, or performing a shift function on two 4-bit binary words. A shift function occurs when either the high or low four bits of the "A" and "B" Bus are combined for output (via the "C" Bus) to a destination register or Data Memory. The Multiply circuitry can only multiply half of the two selected source registers at a time. To multiply two eight-bit source words together requires the execution of more than one multiply instruction. Also, as indicated on the block diagram, immediate data contained within the instruction itself can be entered into the ALU via the "A" Bus.

## 5.4.5 AUXILIARY REGISTERS AND SUBROUTINE STACK

--Aux Reg/Subr Stack Source Selector
--Program Counter
--Stack Address Register
--SAB7 Flip Flop (F/F)
--Instruction Counter
--Auxiliary Registers/Subroutine Stack
--Stack Address Selector
--Mini Instruction Decoder

There are two inputs to the Aux. Reg/Subr. Stack Source Selector. One of these two inputs is the Instruction Counter Register, which contains an address that is one count higher than the address contained in the Instruction Counter. It was mentioned earlier that a CPU is normally setup to execute the instructions contained in Control Memory in sequential order. When a program is written, not all the operations that the CPU is to perform are going to occur in sequence. A computer is capable of making decisions, under program control of course, and the paths taken to solve a particular problem are going to vary depending on the information given to the computer. Depending on the operation being performed, a branch to another portion of the program may be necessary to handle that operation. This is known as branching to a Subroutine. If this branch is only temporary, the CPU will return to the point in the main program where it left off.

For this to be accomplished, the address to return to (IC + 1) must be saved. Storing this subroutine return address is the function of the Subroutine Stack. Remember, that in order to fetch any microinstruction contained in Control Memory, the Instruction Counter has to be changed to the address of the memory location containing that instruction. If the address in the Instruction Counter is changed to get to the subroutine, then the address of the next instruction in the main program routine must be put back into the Instruction Counter in order to return from the subroutine.

The Program Counter is the other input to the Aux. Reg. & Subr. Stack Selector. The Program Counter contains an address that is always one count higher than the address contained in the Program Counter Register. The Program Counter Register, which addresses Data Memory, is used by the microprogram to address the location of the next BASIC-2 Language instruction in Data Memory. When a BASIC-2 language subroutine is called, the Program Counter Register (PC+ 1) must be saved so that when the subroutine has been completed, a return to the main BASIC-2 program flow may be accomplished.

The Auxiliary Registers/Subroutine Stack is a 256 by 8-bit RAM memory used to form 96 Subroutine Registers and 32 Auxiliary Registers. Each of the Auxiliary and Subroutine Registers is capable of holding a sixteen bit address, which means that two eight-bit RAM locations are required to store the information.

The Stack Address Register addresses the next available location in the Subroutine Stack when a microcode subroutine branch instruction is performed, so that the contents of the Instruction Counter can be saved prior to the branch operation. The Stack Address Register also has the responsibility of locating the correct address information to be loaded back into the Instruction counter when a subroutine return microinstruction is executed. The Subroutine Stack Register works on a Last-In First-Out (LIFO) addressing scheme. In other words, the last address loaded into the Subroutine Stack, by a subroutine branch (SB) operation is the first address that will be read out when a subroutine return (SR) operation is performed.

Addressing of the Auxiliary Register is accomplished by (bits 4 thru 8 of) the microinstruction. The micro/machine language programmer must specify which Auxiliary Register is to be used.

The source of the address supplied to the Aux. Reg./Subr. Stack is controlled by the Stack Address Selector.

When the Stack Address Register addresses the Subroutine Stack Registers, or when the Auxiliary Registers are addressed via the microinstruction, only one eight bit location is accessed. Remember, that 16 bits of information are stored in the Auxiliary/Stack registers. The SAB7 Flip Flop (F/F) is used to supply the highest order address bit to allow access to the other eight bits of information that were stored. (The SAB7 F/F adds a count of 128 to the original address.)

The Mini Instruction Decoder is mainly used to decode and control the execution of microinstructions involving the transfer of information to and from the Auxiliary Registers, Subroutine Stack and Program Counter Register.

## 5.4.6 INPUT/OUTPUT CIRCUITY

|  |  |
|---|---|
| --Address Bus Strobe | --Output Bus Strobe |
| --Control Bus Strobe | --Input Bus Strobe |

The I/O Control section is responible for generating three output pulses: an Address Bus Strobe (ABS), an Output Bus Strobe (OBS), and a Control Bus Strobe (CBS). These three pulses or signals are necessary for the transfer of data to/from one of many peripheral devices.

The Address Bus Strobe clocks the address contained in the Address Bus Register onto the Address Bus. This address selects the appropriate peripheral device.

The Output Bus Strobe clocks the output data, which is contained in the "K" Register, to the selected peripheral device.

The Control Bus Strobe requests the selected peripheral device to generate an Input Bus Strobe to the CPU.

Data is clocked from a peripheral device to the CPU by the Input Bus Strobe (IBS). The Input Bus Strobe is generated by the peripheral, and clocks the data to be transferred to the CPU into the "K" Register.

## 5.5 DISK PROCESSING UNIT

The Disk Processing Unit (DPU) is a Z80A based microcomputer responsible for controlling all disk drive (DSDD diskette and fixed-disk) activities, and for supervising data transfer between the LVP CPU and the disk drives.

The DPU is comprised of three logic boards: 1) WL #210-7696 Microcomputer and Memory, 2) WL #210-7694 2200/Disk Interface, and 3) WL #210-7695 Disk Controller. The specific duties of each board, and the circuitry of each board are explained in the following text. A basic block diagram of each board is provided, and should be referenced to allow for easier understanding of the DPU hardware operation.

### 5.5.1 MICROCOMPUTER AND MEMORY (ref: FIGURE 5-3)

The Microcomputer and Memory board is a Z80A based controller dedicated to the 2200/Disk Interface and Disk Controller boards.

Z80A-CPU--

Directs all other logic circuitry--the heart of the DPU. The main components of the Z80A are: an Arithmetic/Logic Unit (ALU), sixteen 8-bit general purpose registers, four 16-bit special purpose registers, an instruction register, address bus and data bus control circuitry, and instruction decode and CPU control circuitry.

Z80A-CTC--

Performs all Z80A timing and counting functions. The CTC (Counter Timing Circuit) is a programmable component with four independent channels that may be selected to operate in the timer mode or the counting mode. Each channel is comprised of two registers, two counters, and an interrupt vector. An interrupt occurs when the counter reaches zero, or when a trigger pulse is received from an I/O device requesting service. When an interrupt occurs, the Z80A enters an acknowledge cycle during which the CTC places a vector address on the address bus (low-order byte). The high-order byte of the address is supplied by the interrupt register of the Z80A to form a pointer to an interrupt service routine located in memory.

EPROM--

Contains the microcode program that controls the Z80A. Four 2K X 8-bit 2716 EPROM's comprise the Read Only Memory.

Select decode circuit--

Monitors certain Z80A address-bus bits to determine which of the four EPROM's will be selected. When an address indicating a location in EPROM is detected, the data contained in that location is made available to the Z80A, via the data bus, for processing. The Z80A reads in the program instruction and then processes that instruction to achieve various pre-defined results.

RAM--

Stores information such as quantities, values, or status, which the microcode program needs to perform its specific task. RAM is also used to store data that is to be transferred between the LVP CPU and the disk drives. Eight 16K X 1-bit 4116 Dynamic RAM's comprise the Random Access Memory. Dynamic RAM, by nature, is a volatile memory, in that, if the RAM's are not "refreshed" in every 2 msec time period, the data contained in them will be lost. The Z80A provides automatic refreshing for RAM.

RAM address register--

Supplies the address (location) in RAM where data is to be read/written. The lower-order bits of the Z80A address bus pass through the address register as the "row address"; the higher-order bits pass through as the "column address".

Parity generating/checking circuit--

Generates a parity bit and writes it into a ninth RAM (parity RAM) whenever data is written to RAM. When data is read from RAM, a parity bit is once again generated and tested against the parity bit that was written to ensure data integrity. (The parity bit that was written is read along with the data.)

DMA controller--

Transfers data between the LVP CPU and RAM, and between the disk drives and RAM. The Direct Memory Access (DMA) controller is a 4-channel, 16-bit 9517-1 device. Although the DMA controller has four channels, only three are used. Channel 0 (highest priority) transfers data between the disk drives and RAM; Channel 1 transfers data from the LVP CPU to RAM; Channel 2 transfers data from RAM to the LVP CPU. By utilizing DMA, LVP-to-disk (and vice versa) data-transfer microprocessing time is cut in half (as compared to utilizing the Z80A).

DMA high-address latch--

Supplies the upper-address byte (to RAM) during DMA transfers; the lower-address byte is supplied by the DMA controller.

I/O port address decoders--

Determines which input or output port (latch) has access to the Z80A data bus. (This is necessary due to the fact that all DPU data travels on a bus line.) The Z80A address-bus bits are monitored by the decoders, and depending on the bit configuration, the desired ports are enabled (or strobed) by the decoder outputs.

Data bus buffer--

Interfaces the Z80A data bus and the 2200/disk interface and controller boards. All data passes through this buffer prior to going to (coming from) the interface and controller boards.

Diagnostic selection switch--

Allows a desired DPU diagnostic test (resident in PROM) to be run.

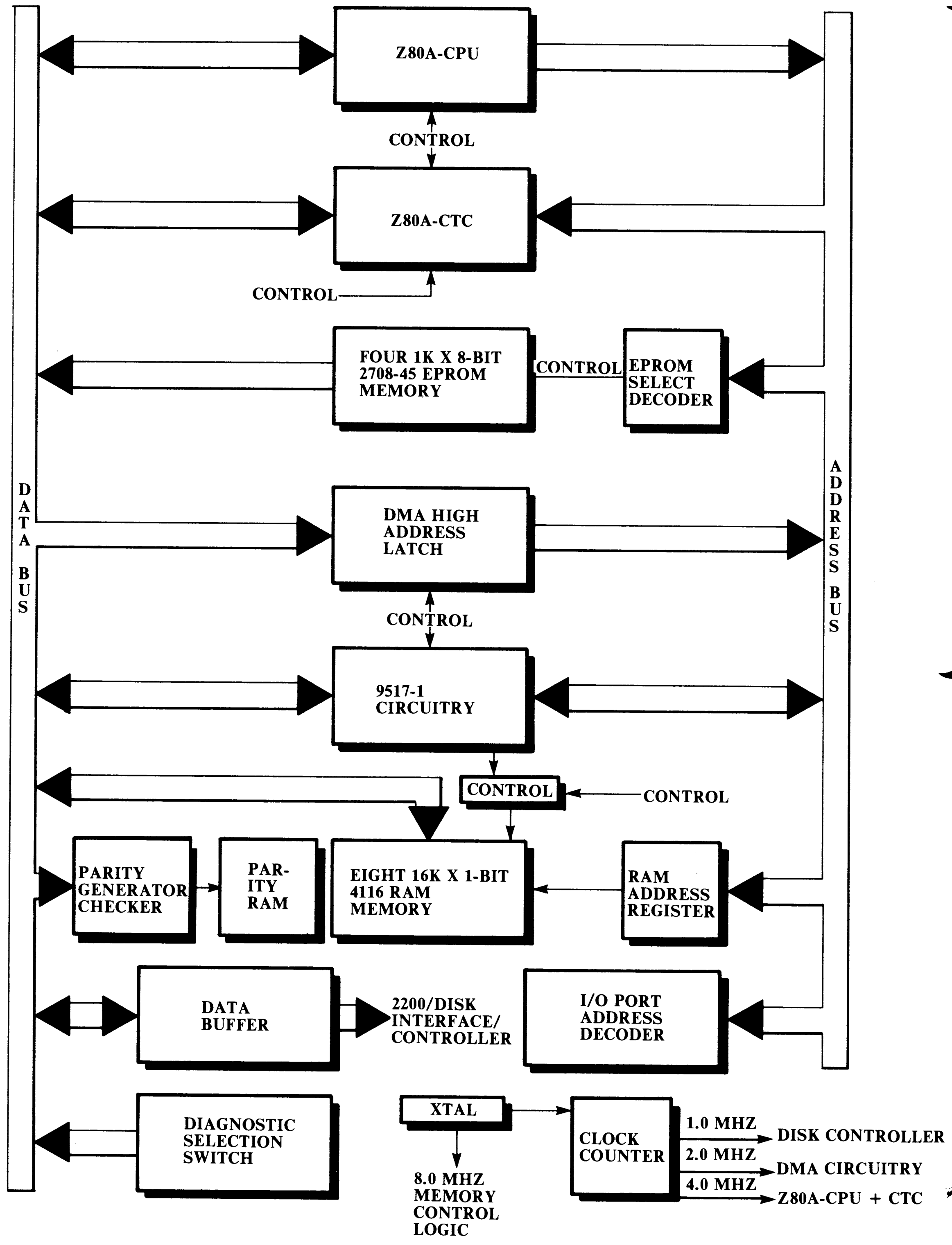


FIGURE 5-3 MICROCOMPUTER/ MEMORY BLOCK DIAGRAM

## 5.5.2 2200/DISK INTERFACE (ref: FIGURE 5-4)

The main functions of the 2200/Disk Interface board are: 1) provide an interface for LVP CPU communications, 2) provide an interface for disk communications, and 3) provide circuitry for separating read data and read clock.

<u>Address switch</u>--

Represents the disk drive address selected by the user. The outputs of the switch are applied to the address compare circuit.

<u>Address compare circuit</u>--

Compares the specified disk address (sent from the LVP via the CPU address bus) with the setting of the address switch. If the two compare, the select flip/flop is set.

<u>Select flip/flop</u>--

Enables the CPU strobe/status latch, which in turn allows data to be received from the CPU.

<u>Output bus latch</u>--

Receives the data sent to the DPU from the LVP.

<u>Output bus buffer</u>--

Transfers input data from the output bus latch to the Z80A data bus. The data is then sent to RAM under control of the DMA device.

<u>Input bus buffer</u>--

Sends data to the LVP CPU from the DPU. The data passes from the Z80A data bus through the buffer and onto the CPU input bus.

Drive status latch--

Transfers disk drive status information onto the Z80A data bus. From there, the information is scrutinized by the microprogram to determine the exact state of the disk drives. Some examples of disk drive status are write protect, drive ready, and seek complete.

Disk control latch--

Accepts disk drive control information from the Z80A, via the data bus, and applies the information to the drive control buffer.

Drive control buffer--

Sends the disk control information (received from the drive control latch) to the drives. Some examples of drive control are head load, drive select, and step.

Interrupt latch--

Accepts various interrupt signals generated by the control circuitry, and presents those signals to the Z80A via the data bus. The interrupt signals instruct the Z80A to perform the required service routine.

Read/write clock oscillator--

Generates the clock frequencies required by the read/write circuitry, the VCO, and other circuitry.

Phase locked loop--

Is a variable controlled oscillator (VCO) that synchronizes with read data/clock to allow for data/clock separation. The VCO is required due to the fact that every bit cell does not have a clock pulse--which is the nature of MFM encoding.

Data/clock separator--

Identifies and separates read data from read clock bits.

Address mark detect circuit--

Compares the read data bits with a one byte address mark, which is preloaded into a register from the Z80A data bus. When eight consecutive bits of read data match the address mark bits, an address mark found signal is generated. This circuit is used to find the beginning of a sector, and then the beginning of the data field in that sector.

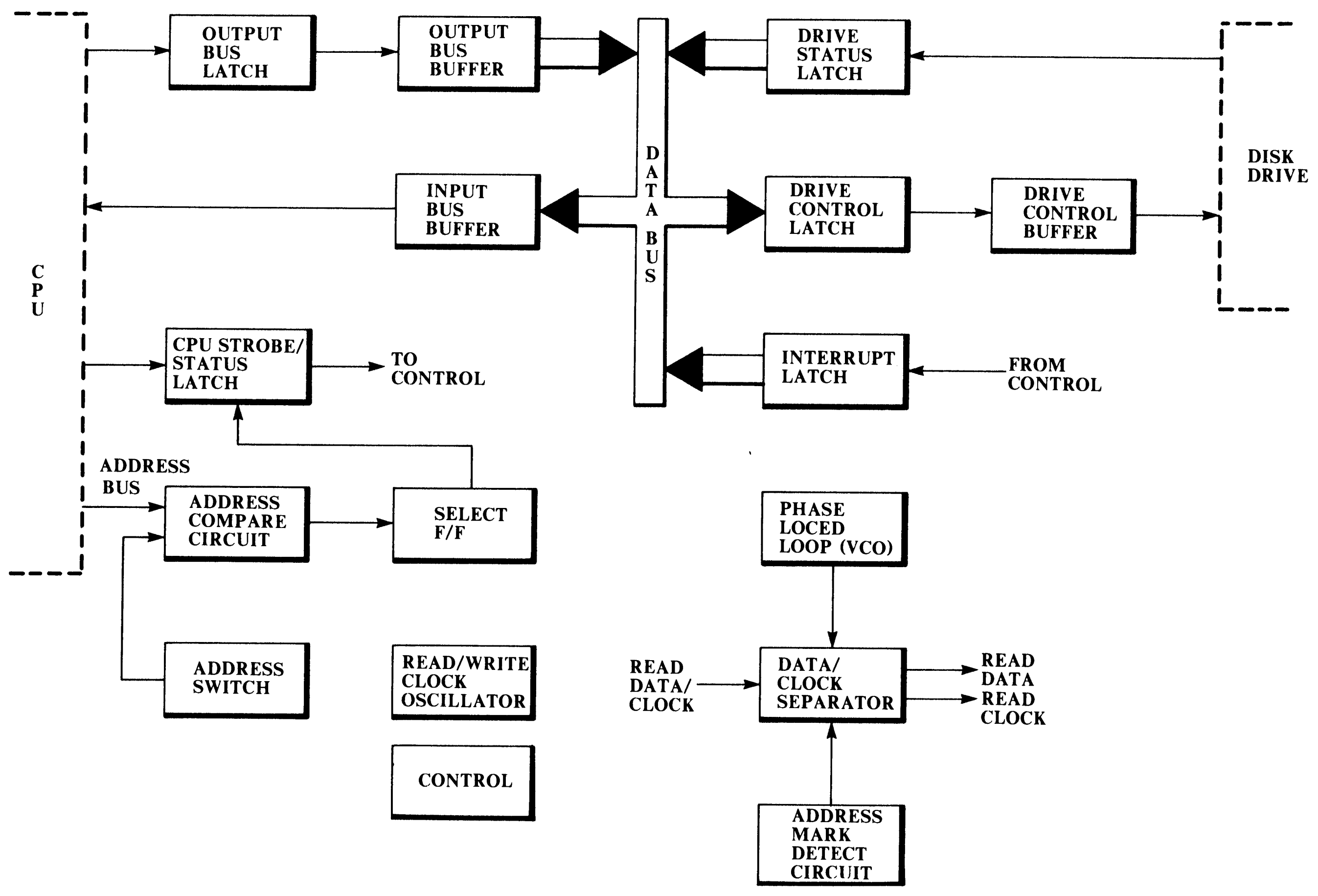


FIGURE 5-4 2200/DISK INTERFACE BLOCK DIAGRAM

## 5.5.3 DISK CONTROLLER (ref: FIGURE 5-5)

The main functions of the Disk Controller board are: 1) convert parallel write data to serial FM/MFM encoded data, 2) convert serial read data to parallel, and 3) monitor and check address/data header information and CRC bytes for errors.

### Bit/byte counter--

Counts read/write clock bits, and generates a byte count every eight bits. The outputs of the counter control (directly or indirectly) the majority of circuitry on the disk controller board. The bit portion of the counter (bit counter) provides input signals to the clock generator, supplies select lines to the parallel-to-serial converter that is responsible for writing address/data header information, and provides select lines to the HEX(4E) filler-byte-code generator. The byte portion of the counter (byte counter) increments the address to the Programmable Logic Array, indicates what type of address error was detected (via the control buffer), and selects (via the port 6X control mux) the appropriate location in the scratch pad where the reference address/data header bytes are stored. (These bytes are used for comparison during a read operation.)

### Programmable Logic Array (PLA)--

Generates instruction commands to all read/write control circuits. The PLA consists of two 2K X 8-bit 2716 EPROM's for control of the DSDD diskette and the fixed-disk drives, and one 1K X 8-bit 2708 EPROM for controlling read operations of single density diskettes on the DSDD drive. The locations in the PLA are addressed by the byte counter. The outputs of the PLA are applied to the PLA buffer.

### PLA buffer--

Routes the instruction commands received from the PLA to the appropriate read/write control circuitry.

Clock generator--

Provides timing signals for all read/write control circuitry. The active output of the clock generator is enabled by the bit counter.

4E code generator--

Generates a HEX(4E) filler-byte code which is written on the disk preceeding the preamble and following the last byte of information in the sector. The byte counter selects the appropriate parallel inputs through the parallel-to-serial converter in order to generate the 4E code. At preamble time, the generator is disabled and all zeros (preamble pattern) are forced out. The output of the generator is input to the "W data" select circuit.

Control buffer--

Accepts inputs from the data bus, and outputs control signals specifying the type of operation (read, write, or format) and the type of drive (fixed, DSDD, or SDF--single density format). The control signals select the appropriate PLA EPROM('s), and the location (address) in the PLA where the desired instruction routine (read, write, or format) is stored.

Control latch--

Applies error-type (CRC, address, etc.) information to the data bus for input to the Z80A. The byte counter inputs specify the exact type of address error.

Scratch pad--

Is a 16-word memory used to store address/data header information. The information is written into the memory via the data bus. The location in memory is specified by the port 6X control multiplexer. The scratch pad outputs go to the scratch pad buffer.

Port 6X control multiplexer--

Provides address (location) select signals to the scratch pad memory. The Z80A address bus inputs represent the address when writing into the scratch pad; the byte counter inputs represent the address when reading from the pad.

Scratch pad buffer--

Accepts the output from the scratch pad, and routes that data to the address compare circuit (read operation), and to a parallel-to-serial converter (write operation).

Address compare circuit--

Compares address/data header information that is read from the disk with reference header information that is stored in the scratch pad. If the reference and read data are identical, the read operation continues; if the information differs, an address error is flagged by the address error flip/flop.

Parallel-to-serial converter (header)--

Converts the address header and data header, received from the scratch pad, to serial data for transmission to the disk drive (write operation). The byte counter selects the appropriate parallel inputs through the converter to generate the serial data.

Serial-to-parallel converter--

Receives serial read data from the data/clock separator, and converts it to parallel data. If the data received is header information, it is sent to the address compare circuit for checking; if the data is the actual data field, it is sent to the FIFO input multiplexer, and the CRC device.

FIFO input multiplexer--

Provides the FIFO stack with data read (from serial-to-parallel converter), or data that is to be written (from the data bus).

FIFO stack--

Is a first-in, first-out (FIFO) memory for temporary storage of read/write data that is transfered between the DMA controller and the disk drive (disk controller).

FIFO buffer (read)--

Applies the read data received from the FIFO stack to the data bus for input to the DMA controller.

FIFO buffer (write)--

Accepts write data from the FIFO stack, and sends it to a parallel-to-serial converter.

Parallel-to-serial converter (data)--

Converts the parallel write data received from the FIFO buffer to serial data. The serial data is then sent to the "W data" select circuit.

"W data" select circuit--

Determines what type of write data is to be routed to the FM/MFM encoder for transmission to the disk drive. The "W data" circuit selects the HEX(4E) filler-byte code, the HEX(00) preamble code, the actual data field, or the CRC bytes. The desired write data is then sent to an OR gate prior to encoding. At the same time, the write data is input to the CRC device.

OR gate--

Routes either "W data" or address/data header information to the FM/MFM encoder.

FM/MFM encoder--

Takes serial write data from the OR gate, and converts it to FM/MFM prior to transmission to the disk drive.

CRC device--

Accepts write data and generates a two-byte check character for that data. The CRC bytes are written on the disk along with the data. When reading, the data is fed through the CRC device generating another check character which is compared with the CRC bytes for the data read. (The CRC bytes are read along with the data.) If the CRC written (then read) and the CRC generated during the read operation do not compare, a CRC error is flagged by the CRC error flip/flop.

8.0 MHZ crystal oscillator/counter--

Generates all timing signals for the DPU. A counter breaks the 8.0 MHZ frequency down into three other clock frequencies. The specific clocks and the circuitry they control are:

| | | |
|---|---|---|
| 8.0 MHZ | - | Memory control logic |
| 4.0 MHZ | - | Z80A-CPU and CTC |
| 2.0 MHZ | - | DMA controller |
| 1.0 MHZ | - | Disk control circuitry |

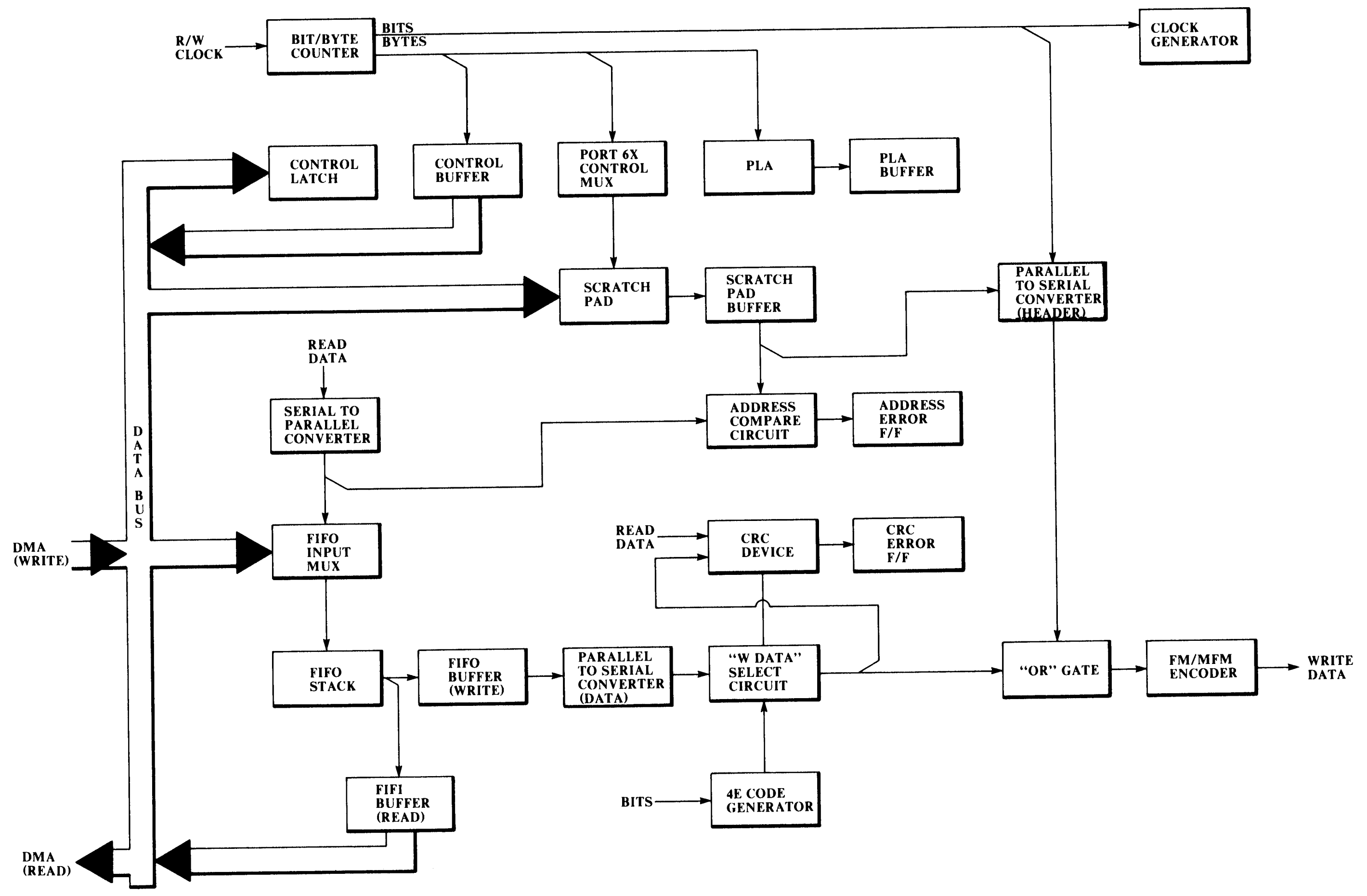


FIGURE 5-5 DISK CONTROLLER BLOCK DIAGRAM

## NOTES

NOTES

## SECTION 6
## SITE PREPARATION

For information concerning preinstallation site planning and preparations, refer to the corporate "Customer Site Planning Guide" WL #700-5978, its updates, and CE documentation category I.A.7.

## NOTES

# SECTION 7

# INSPECTION, UNPACKING, AND CABINET LEVELING

## 7.1 TOOLS REQUIRED

Heavy duty wire cutters -- WL #726-9416

Adjustable wrench -- WL #726-9425

## 7.2 PRE-UNPACKING INSPECTION

Before unpacking the 2200LVP, check the packing slip to ensure that the proper equipment has been delivered. After checking the packing slip, visually inspect the container carefully for any indications of possible shipping damage (crushed edges or corners, puncture holes, tears, etc.). If any shipping damage is noted, file an appropriate claim promptly with the carrier involved and notify the WLI Distribution Center (Department 90), Quality Assurance Department, of the nature and extent of that damage, making arrangements for equipment replacement, as necessary.

## 7.3 UNPACKING INSTRUCTIONS (ref: FIGURE 7-1)

1. Using heavy duty wire cutters (WL #726-9416), cut the two straps that secure the cardboard box cover to the shipping pallet.

2. Remove the cardboard box cover, and the protective cardboard filler.

3. Using an adjustable wrench (WL #726-9425), remove the three shipping bolts from the underside of the shipping pallet. (There is one bolt in the front of the unit, and two bolts in the rear of the unit.)

4. Carefully lift the unit off the shipping pallet. (Although it may be possible for one person to lift the unit off the palate, it is recommended that two people perform this step.)

5. Store the shipping pallet, cardboard box cover/filler, and shipping bolts for possible future reshipment.

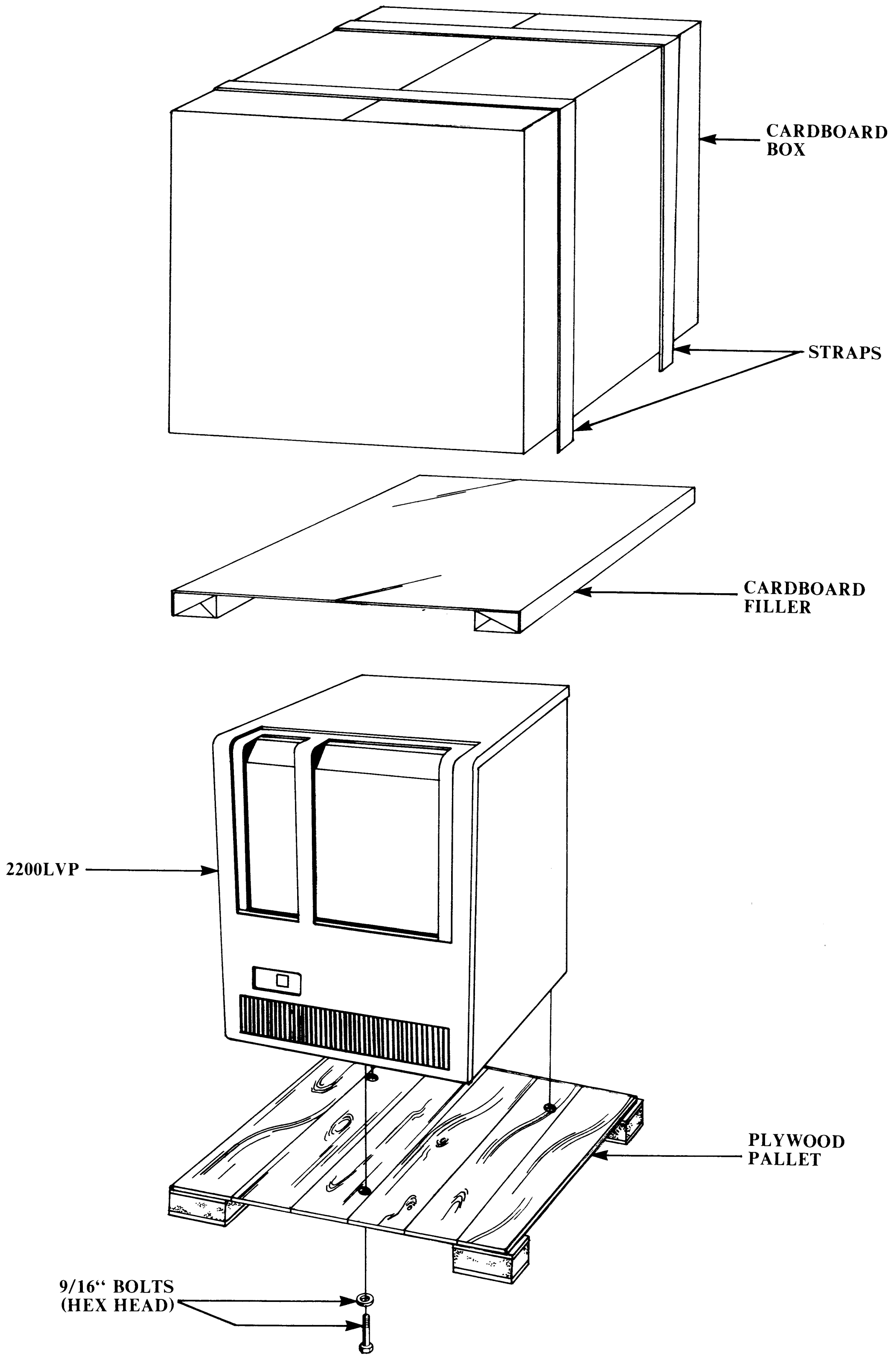


**FIGURE 7-1 2200LVP PACKAGING**

## 7.4 CABINET LEVELING PROCEDURE

Cabinet leveling and subsequent equipment power-on checkout procedures should not be performed until the unit is in its final operating location. Cabinet leveling consists of adjusting the four leveling-pad screw bolts (located on the under side of the cabinet assembly, next to each wheel caster--ref: FIGURE 7-2) as necessary to support the unit off its wheel casters and in level alignment with adjacent peripherals. A bubble level is desirable to confirm the final level setting but is not necessary for adequate performance of this procedure.

1. Move the unit to its permanent location.

2. Turn the leveling pads counterclockwise (down) until they support the full weight of the unit, which must be held off all wheel casters.

3. Coarse-adjust the leveling pads to align the unit with adjacent equipment, ensuring that the unit remains off all wheel casters.

4. Once the coarse-adjustment alignment appears satisfactory, fine-adjust the leveling pads as necessary to further level the unit to a solid, maximum-stability condition, with no rocking motion detectable when pushed. If any sort of bubble level is available, place the level on the top cover of the cabinet assembly and then level the unit, both front-to back and side-to-side.

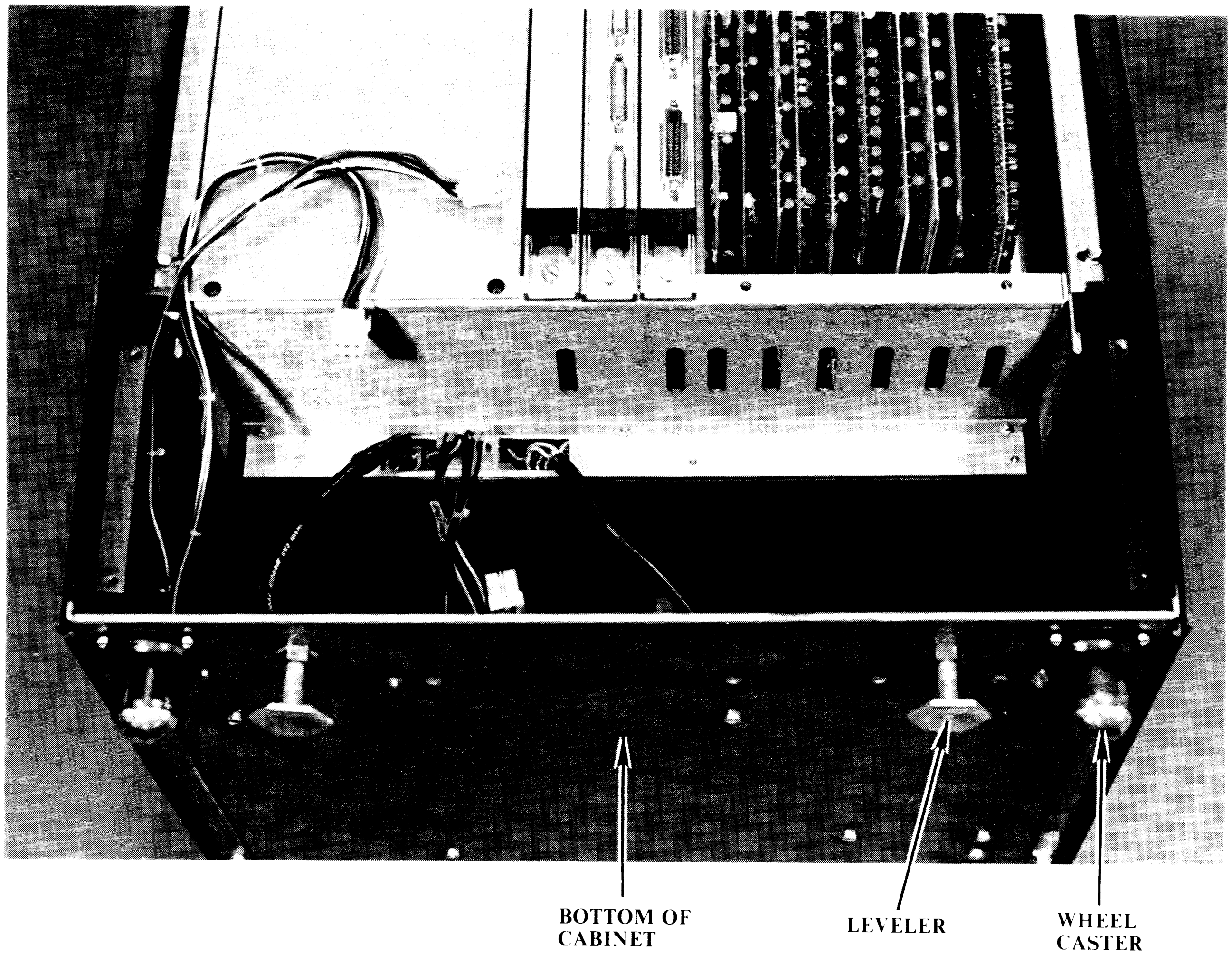


**FIGURE 7-2 LEVELING-PAD SCREW BOLTS**

NOTES

NOTES

# SECTION 8

# INSTALLATION

Following is a list of documentation categories referenced by this section. In many cases, documentation from these categories is required to ensure correct installation of a 2200LVP system.

MODEL 2236MXD MULTIPLEXER/CONTROLLER -- IV.B.1
MODEL 22C32 TRIPLE CONTROLLER -- IV.B.1
I/O CONTROLLERS: SETTING DEVICE ADDRESS SWITCHES -- IV.B.1
I/O CONTROLLERS: PART #'S & I/O CABLE CONNECTION -- IV.B.1
I/O CABLE CONNECTOR INSTALLATTION -- I.B.0
2236DE INTERACTIVE TERMINAL -- III.D.1
DISK DRIVES -- III.A.11 AND III.A.12
PERIPHERALS -- Appropriate categories

## 8.1 PRE-INSTALLATION INSPECTION

1. Be certain that the customer site has been prepared according to the guidelines referenced in SECTION 6, and then place the LVP unit in its assigned physical location, and perform the cabinet leveling procedure (ref: Section 7.4).

2. Remove the top cover from the cabinet assembly (ref: SECTION 11).

3. Remove the cover from the CPU chassis subassembly (ref: SECTION 11).

4. Inspect the CPU chassis and the entire cabinet assembly for damaged or loosened aasemblies. Also check for loose hardware or debris. If any shipping damage is noted, notify the WLI Distribution Center (Department 90), Quality Assurance Department, of the nature and extent of the damage, making arrangements for equipment replacement, as necessary.

5. Ensure that the unit is thoroughly clean. Use a soft bristle brush and a vacuum cleaner to remove dust from the inside of the unit. Use a mild detergent and a soft cloth or sponge to remove dirt and grime from the cabinet. Do not use abrasive or corrosive chemicals.

## 8.2 INITIAL SETUP

This section consists of:

-- photographs in which the major components of the 2200LVP are pointed out--to familiarize the Customer Engineer with the physical aspects of the LVP.

-- photographs of the circuit boards (CPU and DPU) showing PROM numbers and locations, switch settings, and component loading for the various versions of the same board (memory).

-- an explanation (with photographs) of the units internal cable connections.

-- references to the appropriate documentation categories that deal with equipment associated with (normally included with) the LVP system.

Section 8.3 (Installation and Power-On Procedures) helps link together the various information items contained in this section.

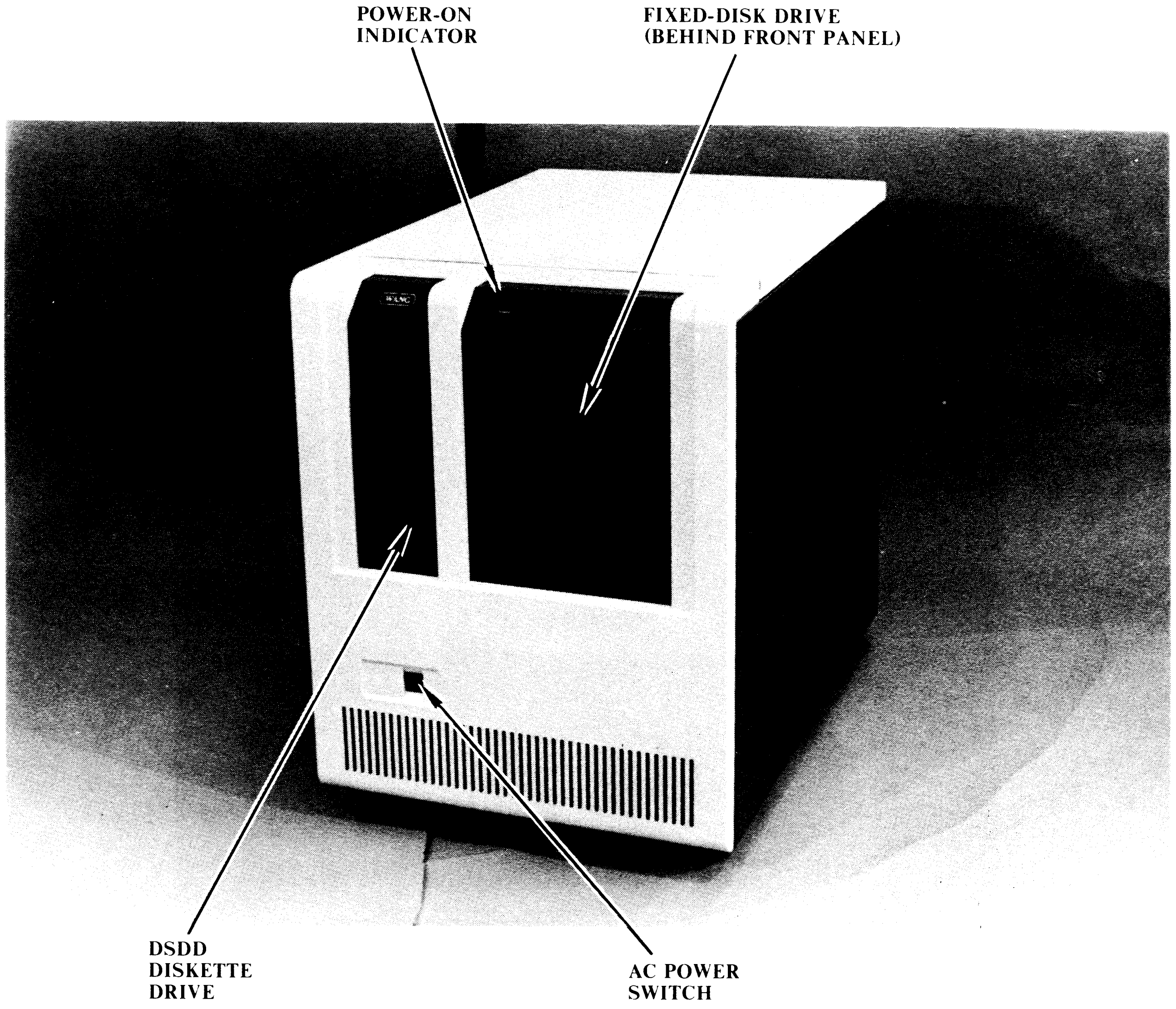


FIGURE 8-1 2200LVP (FRONT VIEW)

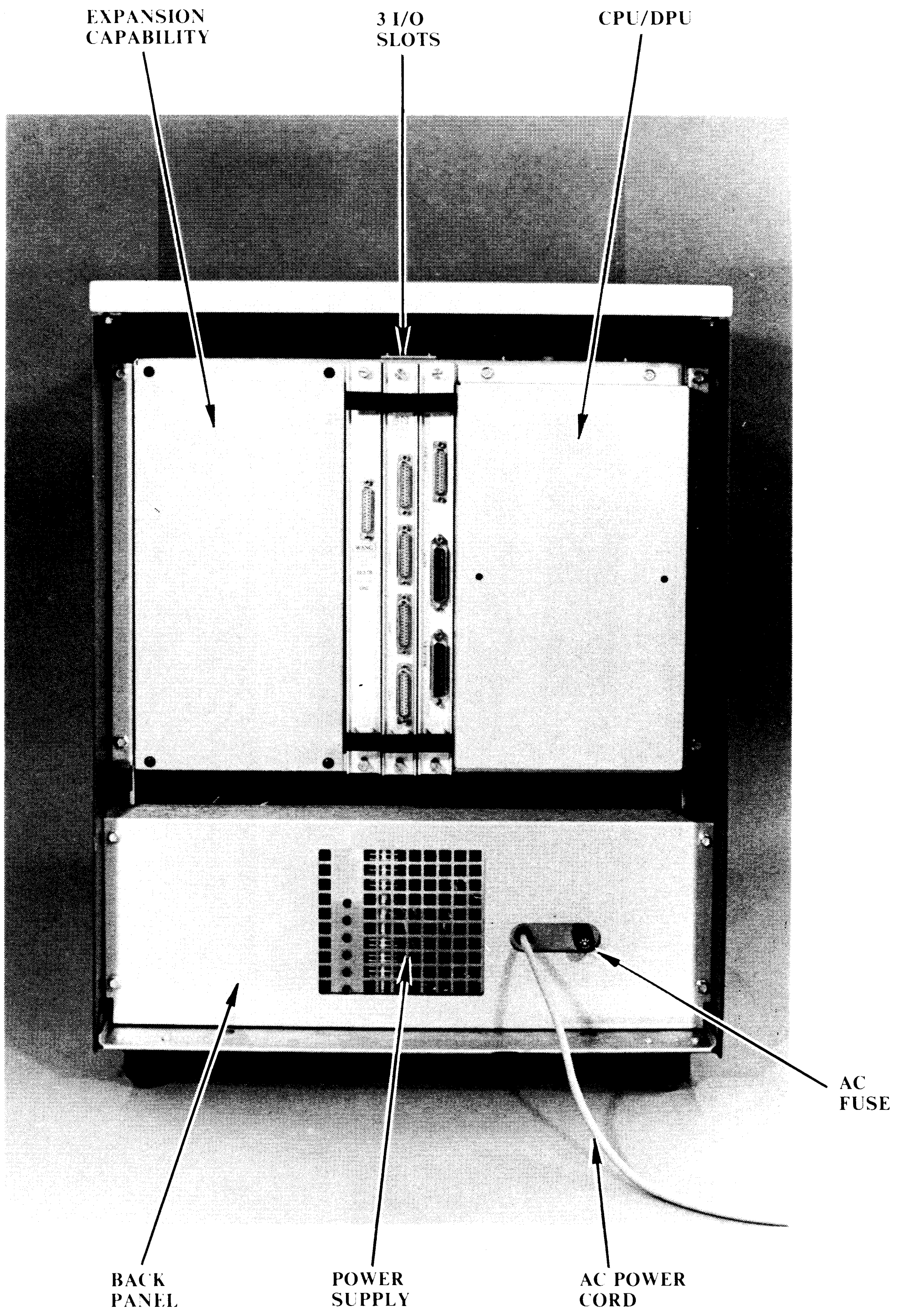


**FIGURE 8-2 2200LVP (REAR VIEW)**

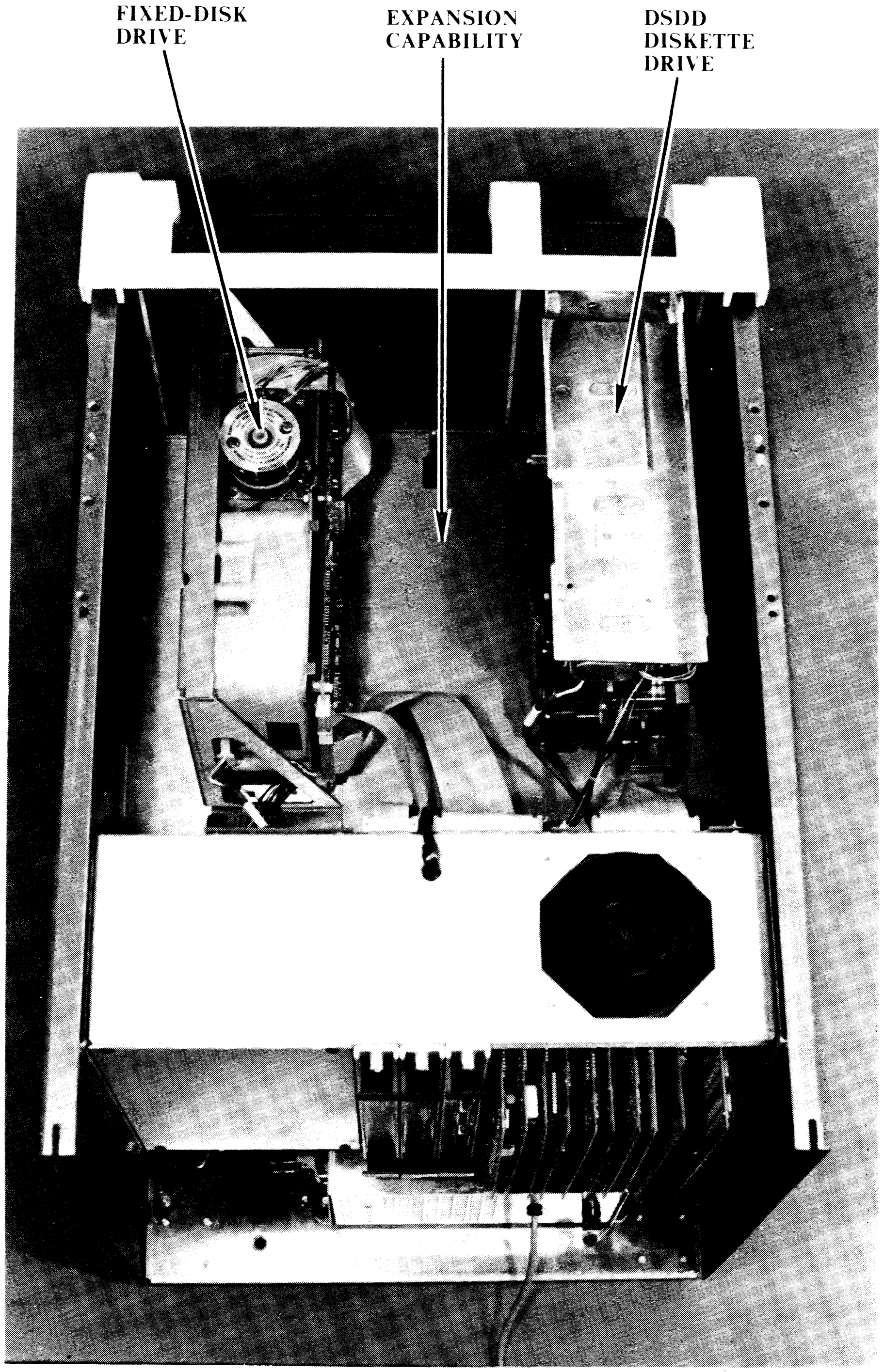


FIGURE 8-3 2200LVP (INSIDE VIEW)

## 8.2.1 2200LVP CIRCUIT BOARDS (W/LAYOUT)

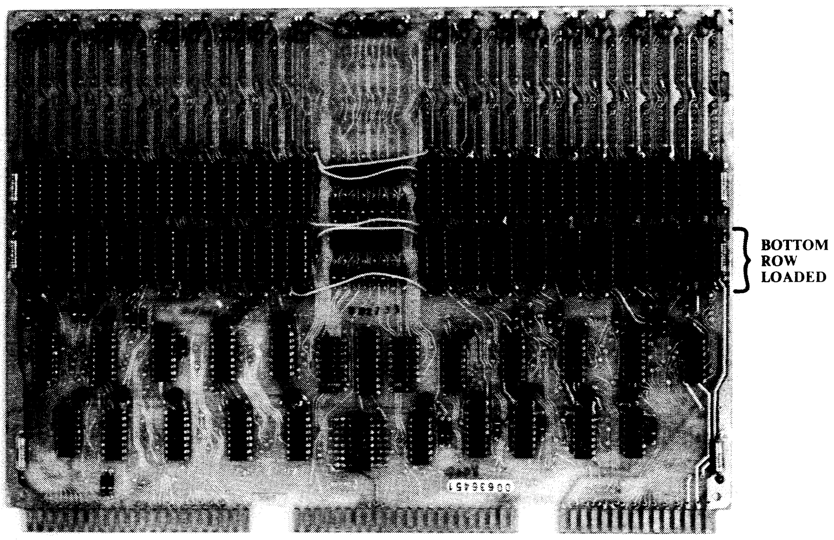


FIGURE 8-4 WL NO. 210-7587-1B DATA MEMORY (32K)

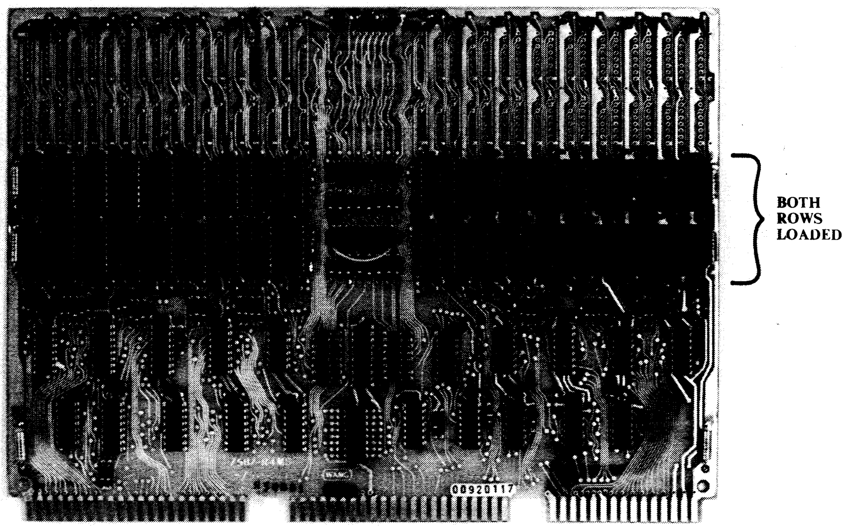


FIGURE 8-5 WL NO. 210-7587-1A DATA MEMORY (64K)

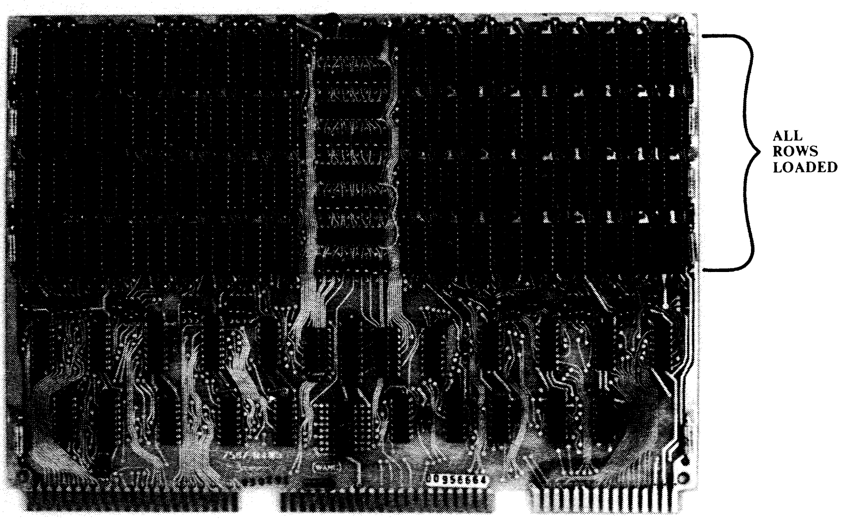


FIGURE 8-6 WL NO. 210-7587-3A DATA MEMORY (128K)

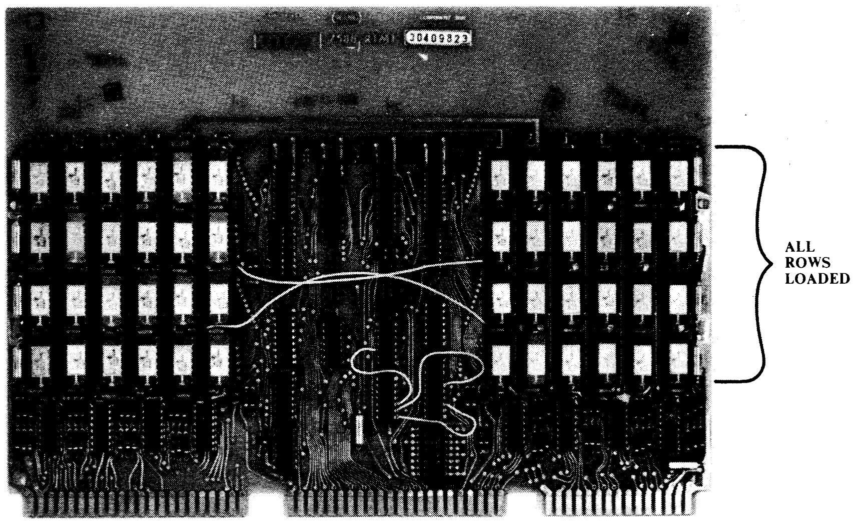
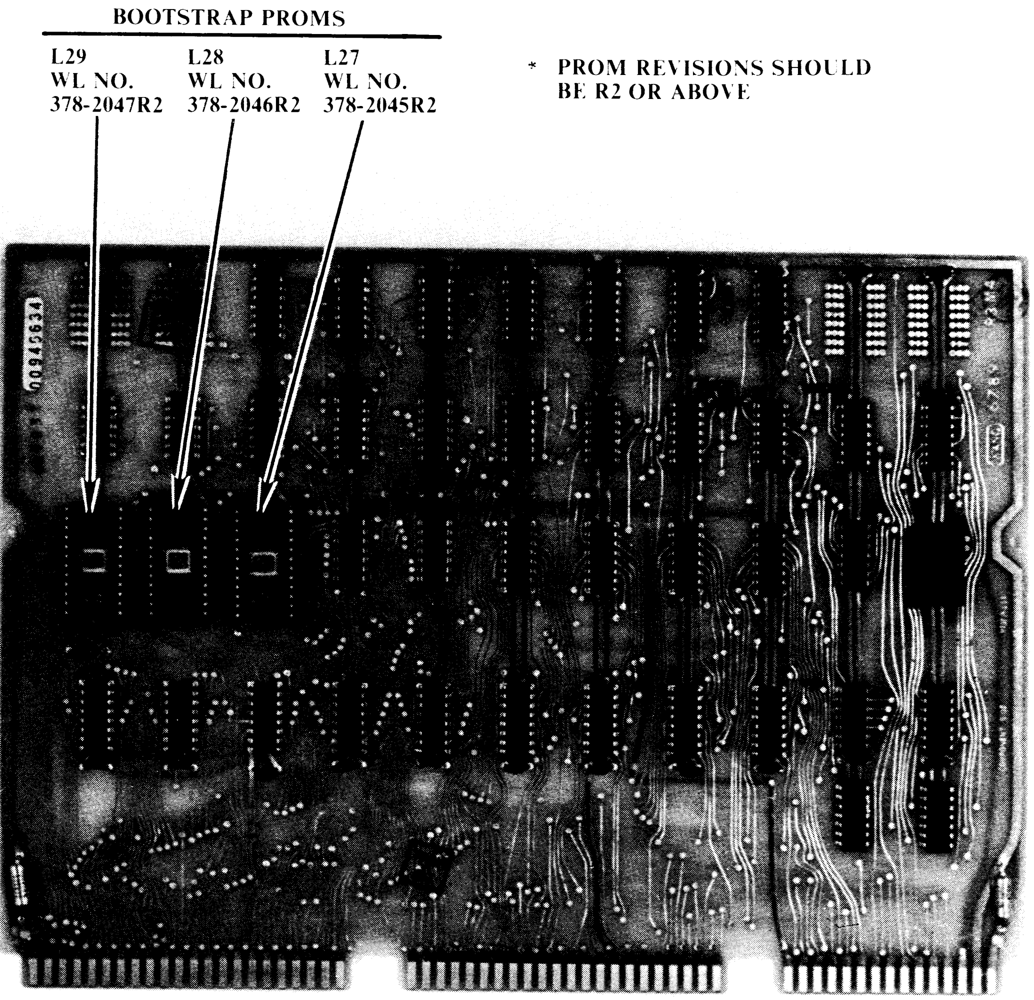


FIGURE 8-7 WL NO. 210-7588-1A CONTROL MEMORY (32K)



* PROM REVISIONS SHOULD BE R2 OR ABOVE

FIGURE 8-8 WL NO. 210-6789-A MEMORY CONTROL

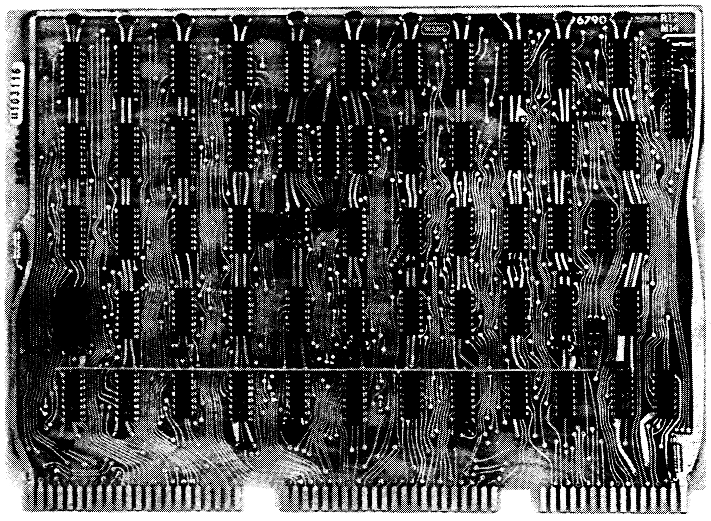


FIGURE 8-9 WL NO. 210-6790 INSTRUCTION COUNTER

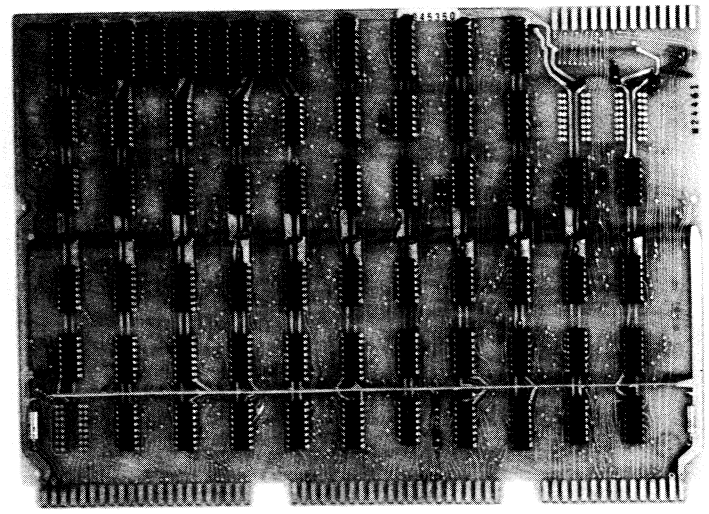


FIGURE 8-10 WL NO. 210-6791 STACK

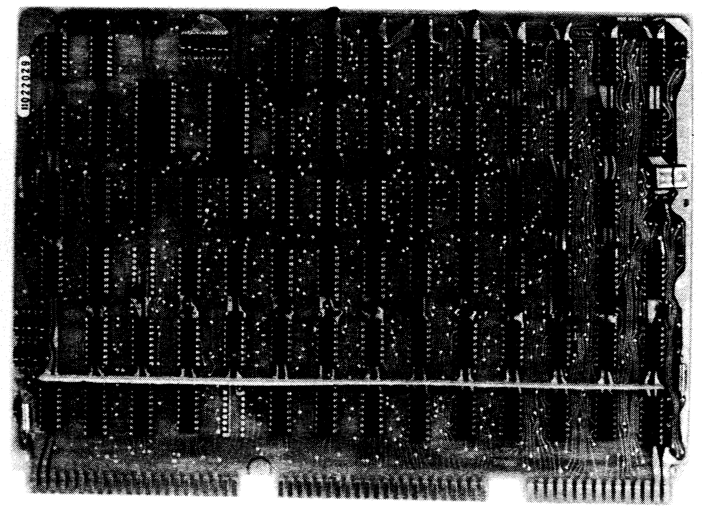


FIGURE 8-11 WL NO. 210-6792 ALU

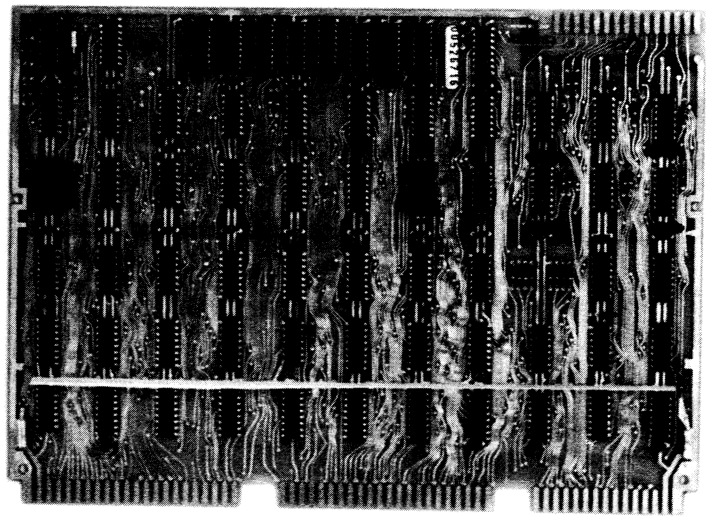


FIGURE 8-12 WL NO. 210 -6793-1 REGISTERS

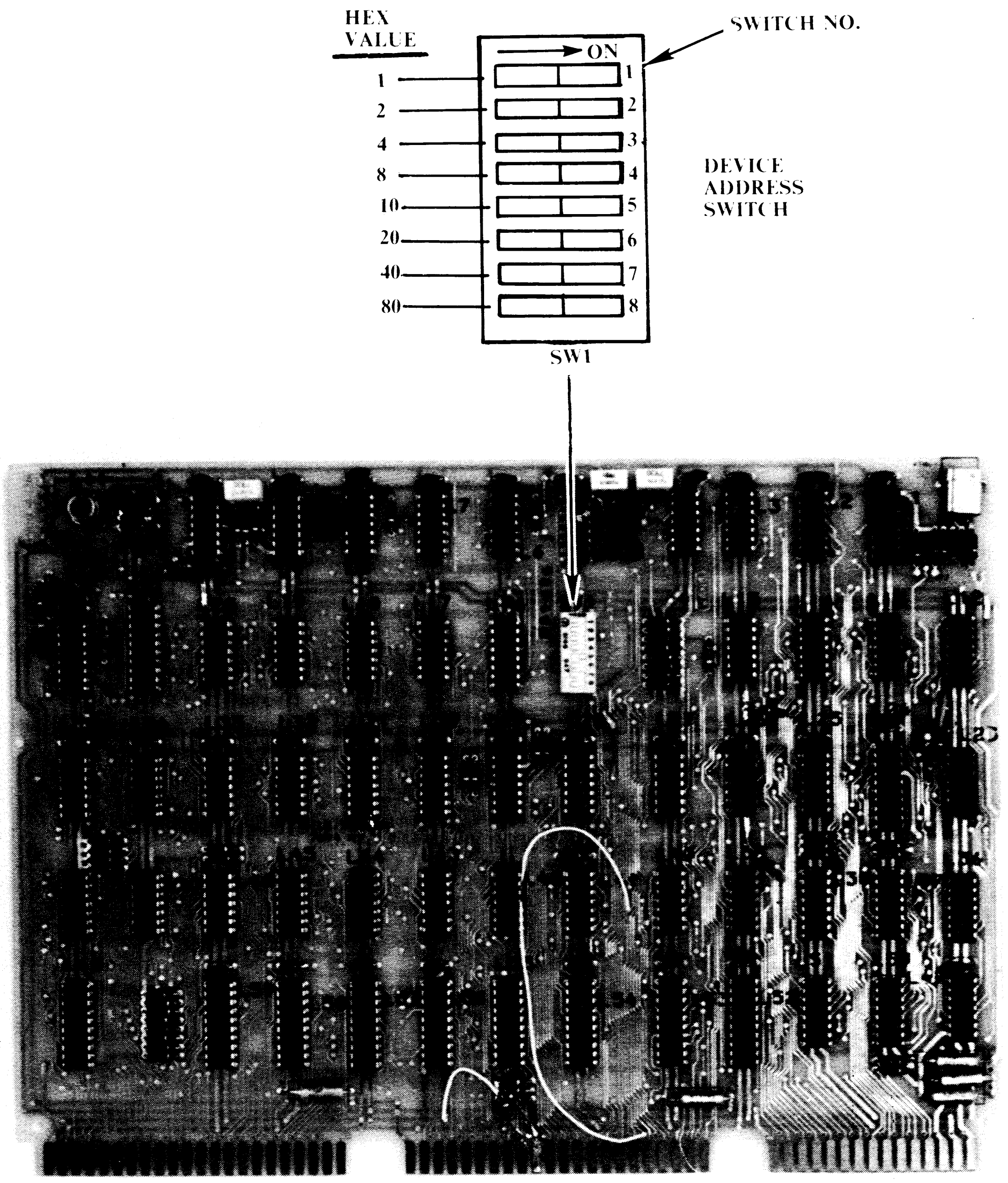


FIGURE 8-13 WL NO. 210-7694 2200/DISK INTERFACE (DPU)

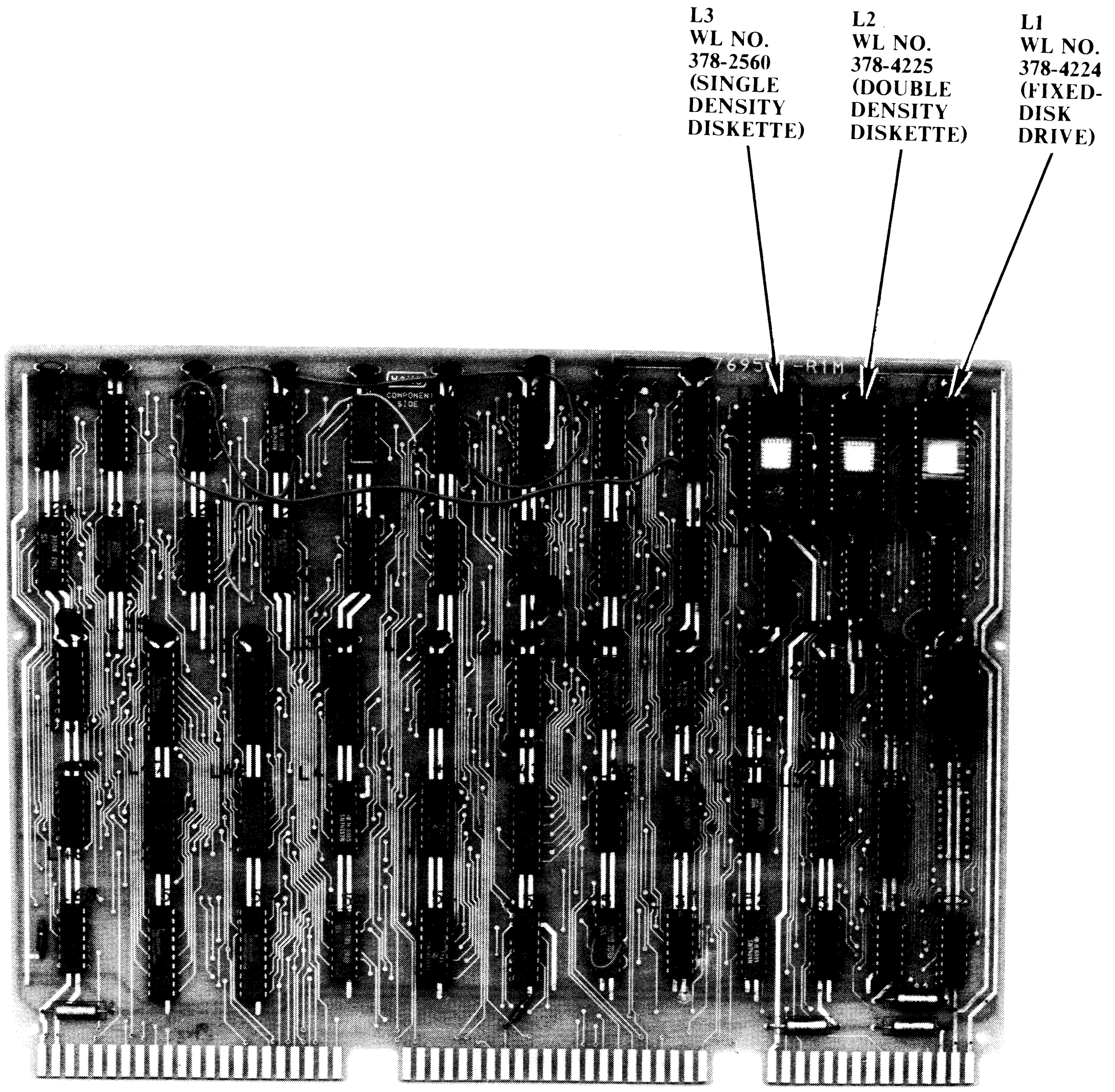


FIGURE 8-14 WL NO. 210-7695-A DISK CONTROLLER (DPU)

| SWITCH SIGNIFICANCE | → ON | SWITCH SETTING |
|---|---|---|
| FOR POWER-ON DIAGNOSTIC—MUST BE ON | 1 | ON |
| OFF IF 8 MB FIXED DRIVE | 2 | SEE SIGNIFICANCE |
| FOR POWER-ON DIAGNOSTIC—MUST BE ON | 3 | ON |
| OFF IF 2 OR 4 MB FIXED DRIVE | 4 | SEE SIGNIFICANCE |
| UNUSED—MUST BE ON | 5 | ON |
| OFF IF DSDD DRIVE IN UNIT | 6 | OFF |
| UNUSED—MUST BE ON | 7 | ON |
| ON TO CERTIFY FIXED PLATTER(S) DURING FORMAT (IF OFF, ONLY OPERATION THAT CAN BE PERFORMED IS FORMAT WITHOUT CERTIFICATION) | 8 | ON |

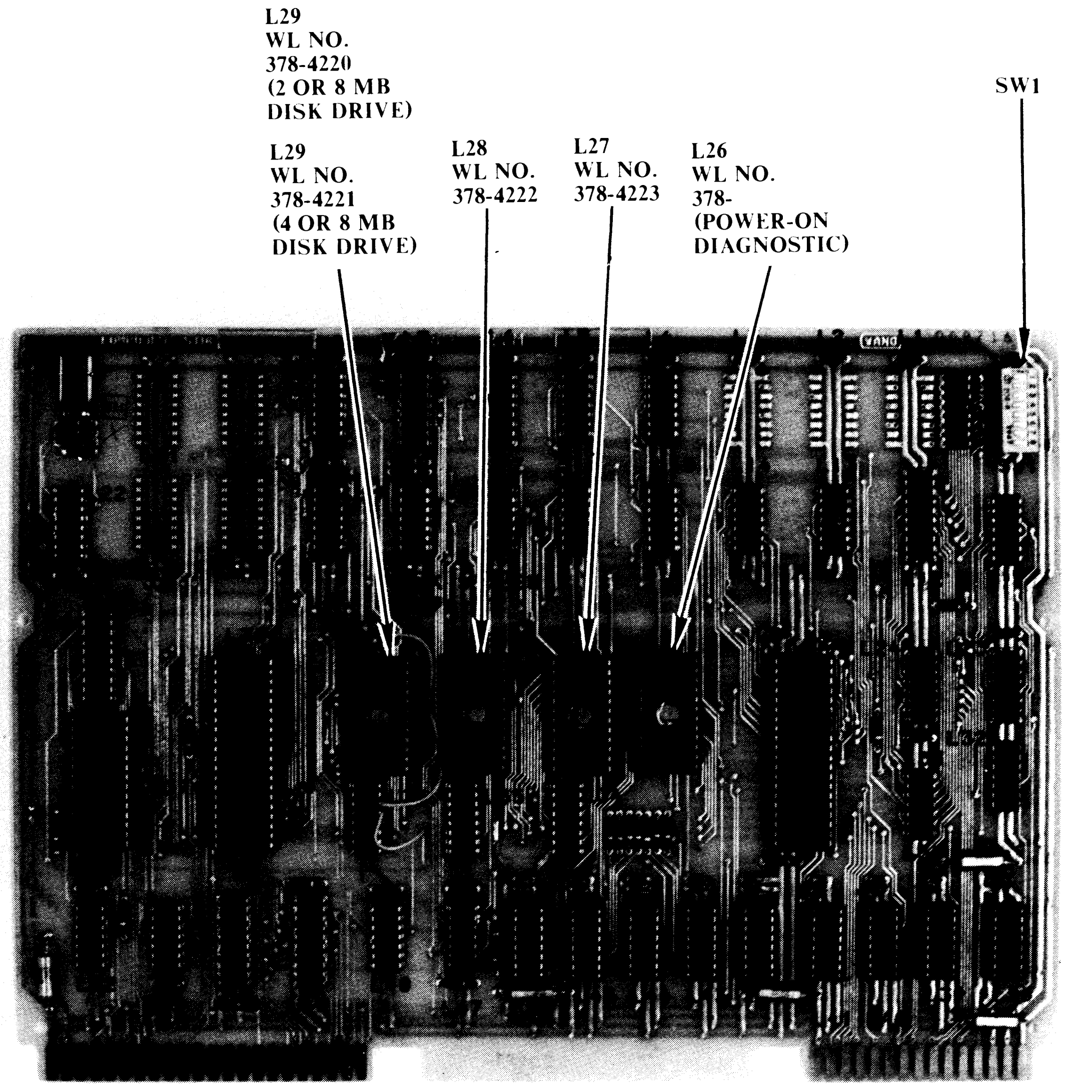


FIGURE 8-15 WL NO. 210-7696-A MICROCOMPUTER/MEMORY (DPU)

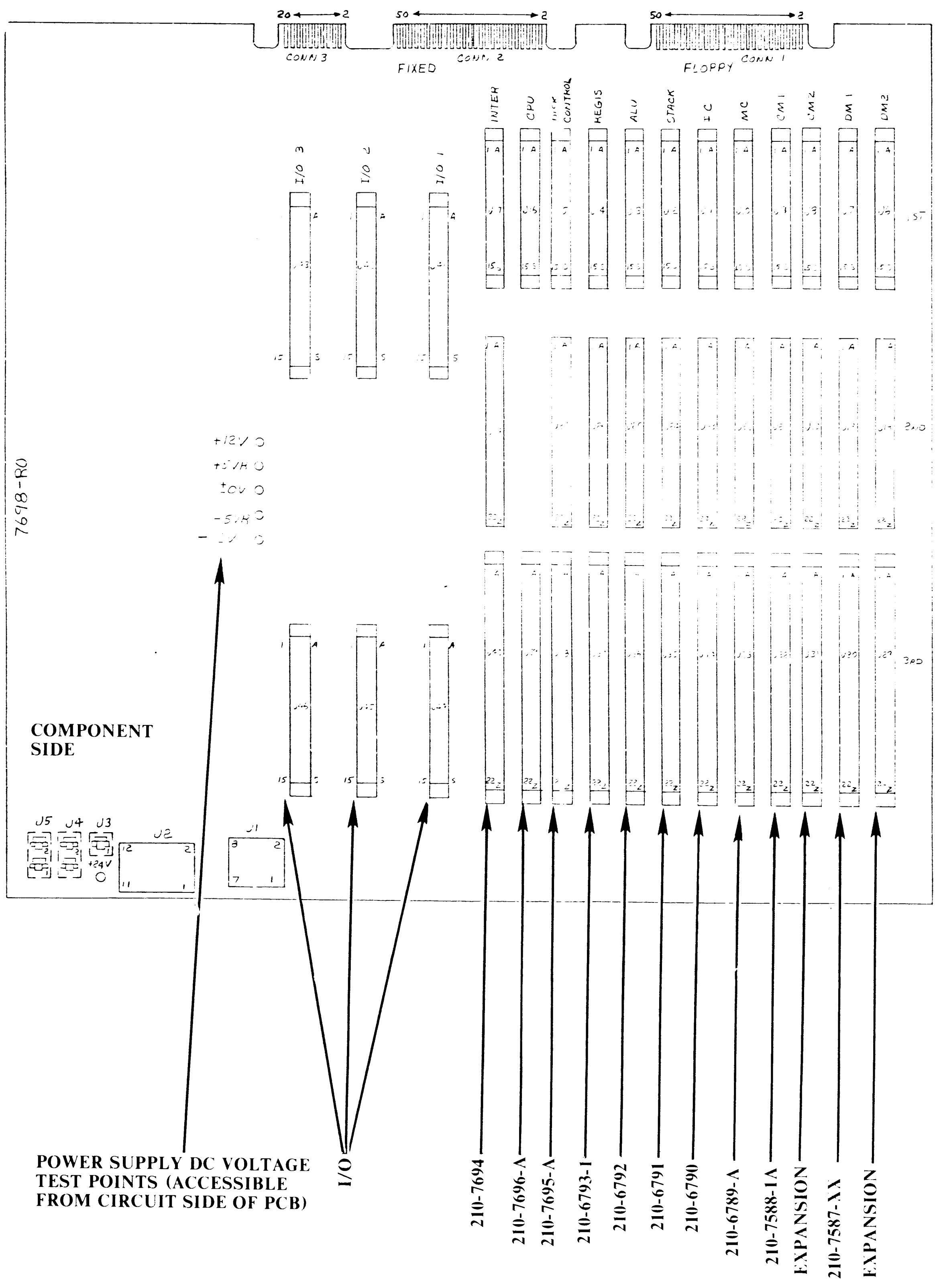


FIGURE 8-16 CIRCUIT BOARD LAYOUT AND VOLTAGE TEST POINTS

## 8.2.2 2200LVP POWER SUPPLY-TO-CPU-TO-DISK POWER CABLE CONNECTIONS

### CPU Motherboard-to-Power Supply DC Power Harnesses

The harness (WL #220-1428) connected to CPU motherboard jacks J3-J5 (ref: FIGURE 8-17) attaches to the 6-pin Mat 'N' Lock connector on the power supply (ref: FIGURE 8-18).

The harness (WL #220-1427) connected to CPU motherboard jack J1 (ref: FIGURE 8-17) attaches to the 10-pin Molex connector on the power supply (ref: FIGURE 8-18).

### CPU Motherboard-to-Disk Drives DC Power Harness

The harness (WL #220-1405) connected to CPU motherboard Jack J2 (ref: FIGURE 8-17) attaches to jack J5 on the DSDD diskette drive (ref: FIGURE 8-19), and to jack J5 on the fixed-disk drive (ref: FIGURE 8-19). (Both drive connector ends of the cable are identical--they may be interchanged.)

### Disk Drive AC Power Cords

The two disk drive ac power cords (WL #220-0251) from the power supply (ref: FIGURE 8-20) attach to jack J4 on the DSDD diskette drive (ref: FIGURE 8-19), and to jack J4 on the fixed-disk drive (ref: FIGURE 8-19).

### Fan Cord

The fan cord (WL #220-1425) from the power supply (ref: FIGURE 8-18) attaches to the fan cord (WL #220-1424) from the CPU chassis (ref: FIGURE 8-18).

### 8.2.3 DISK DRIVE I/O CABLE CONNECTIONS

A 50-pin ribbon cable (WL #220-3119) connects CPU motherboard connector 1 to jack J1 on the DSDD diskette drive (ref: FIGURE 8-21).

A 50-pin ribbon cable (WL #220-3119) connects CPU motherboard connector 2 to jack J1 on the fixed-disk drive (ref: FIGURE 8-21).

A 20-pin ribbon cable (WL #220-3118) connects CPU motherboard connector 3 to jack J2 on the fixed-disk drive (ref: FIGURE 8-21).

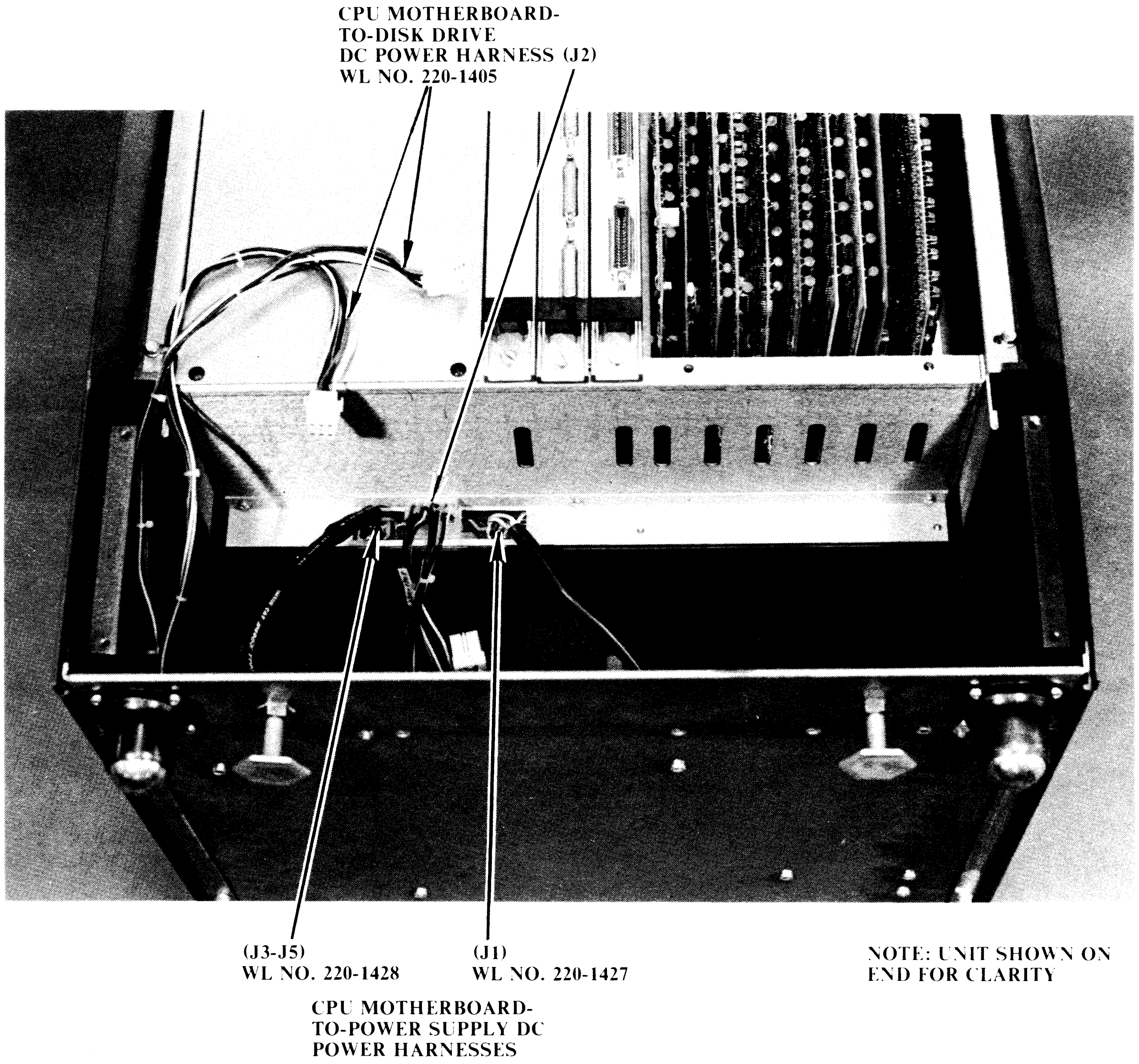
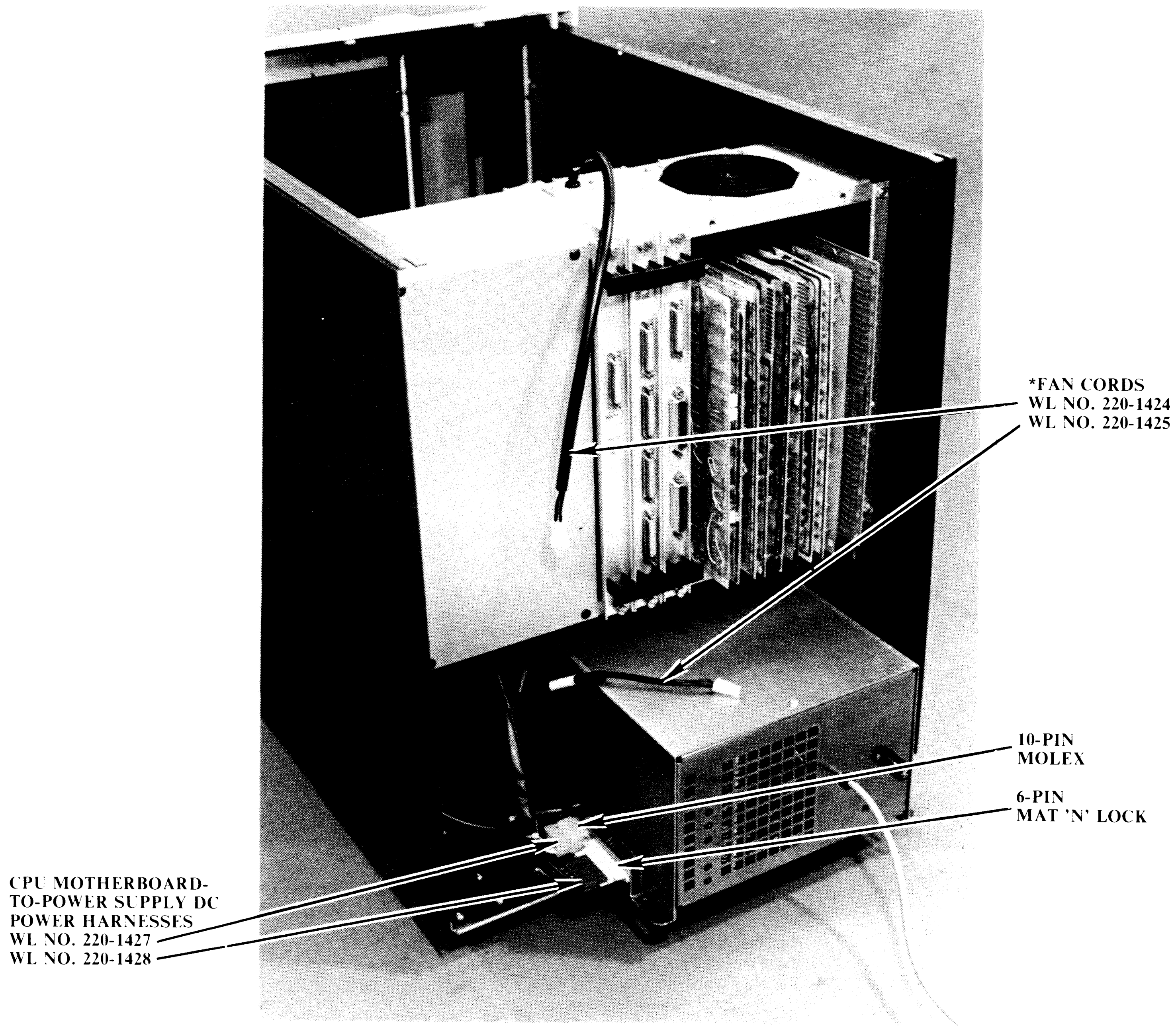
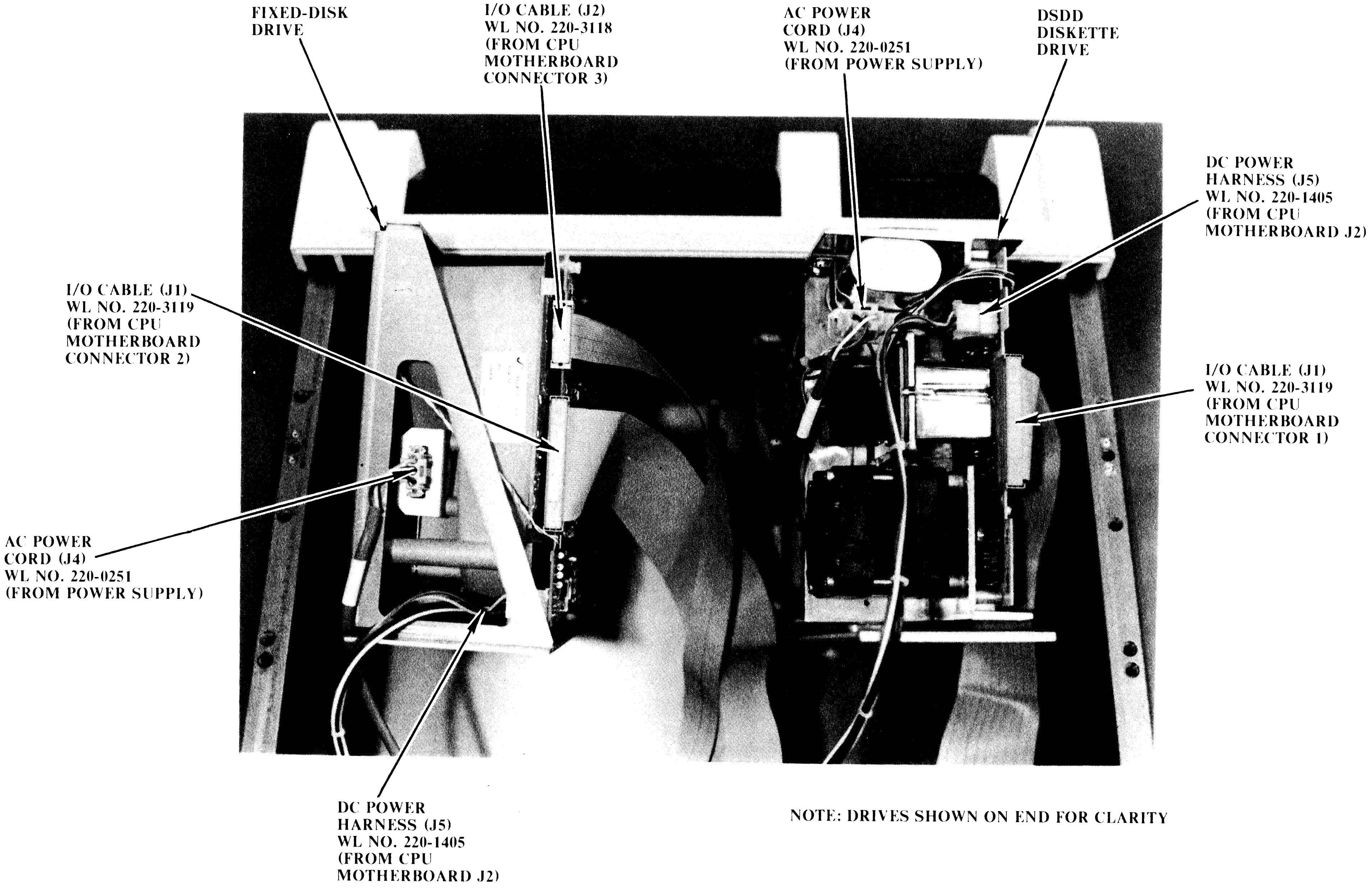


**FIGURE 8-17 CPU MOTHERBOARD POWER CABLE CONNECTIONS**



*FAN CORD CONNECTION IS MADE ON BACKSIDE OF CPU CHASSIS—NOT FRONTSIDE AS MAY BE INDICATED.

FIGURE 8-18 POWER SUPPLY CABLE CONNECTIONS


FIXED-DISK DRIVE
I/O CABLE (J2) WL NO. 220-3118 (FROM CPU MOTHERBOARD CONNECTOR 3)
AC POWER CORD (J4) WL NO. 220-0251 (FROM POWER SUPPLY)
DSDD DISKETTE DRIVE
DC POWER HARNESS (J5) WL NO. 220-1405 (FROM CPU MOTHERBOARD J2)
I/O CABLE (J1) WL NO. 220-3119 (FROM CPU MOTHERBOARD CONNECTOR 2)
I/O CABLE (J1) WL NO. 220-3119 (FROM CPU MOTHERBOARD CONNECTOR 1)
AC POWER CORD (J4) WL NO. 220-0251 (FROM POWER SUPPLY)
DC POWER HARNESS (J5) WL NO. 220-1405 (FROM CPU MOTHERBOARD J2)

NOTE: DRIVES SHOWN ON END FOR CLARITY

FIGURE 8-19 DISK DRIVE POWER AND I/O CABLE CONNECTIONS

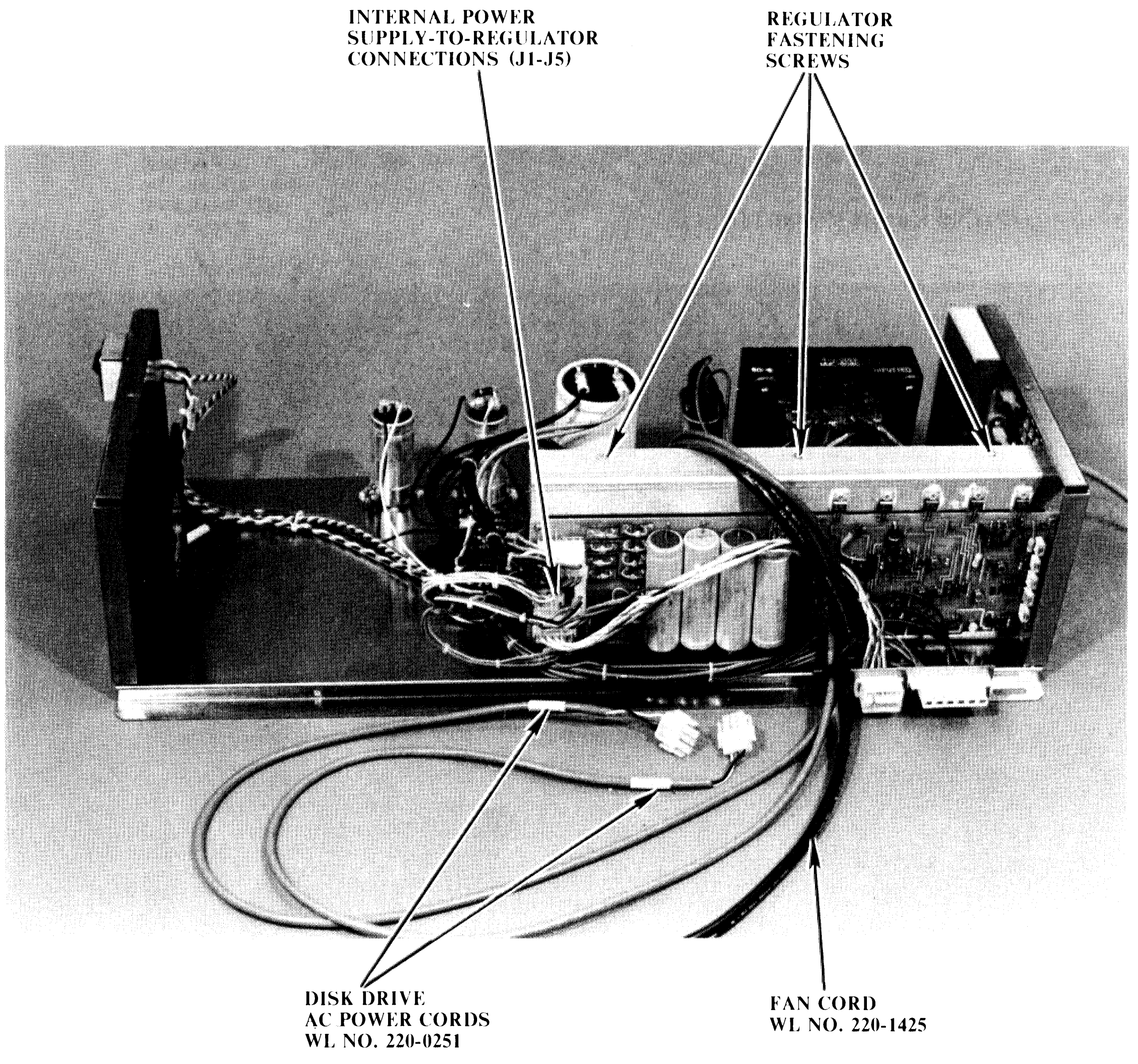


FIGURE 8-20 POWER SUPPLY CABLES

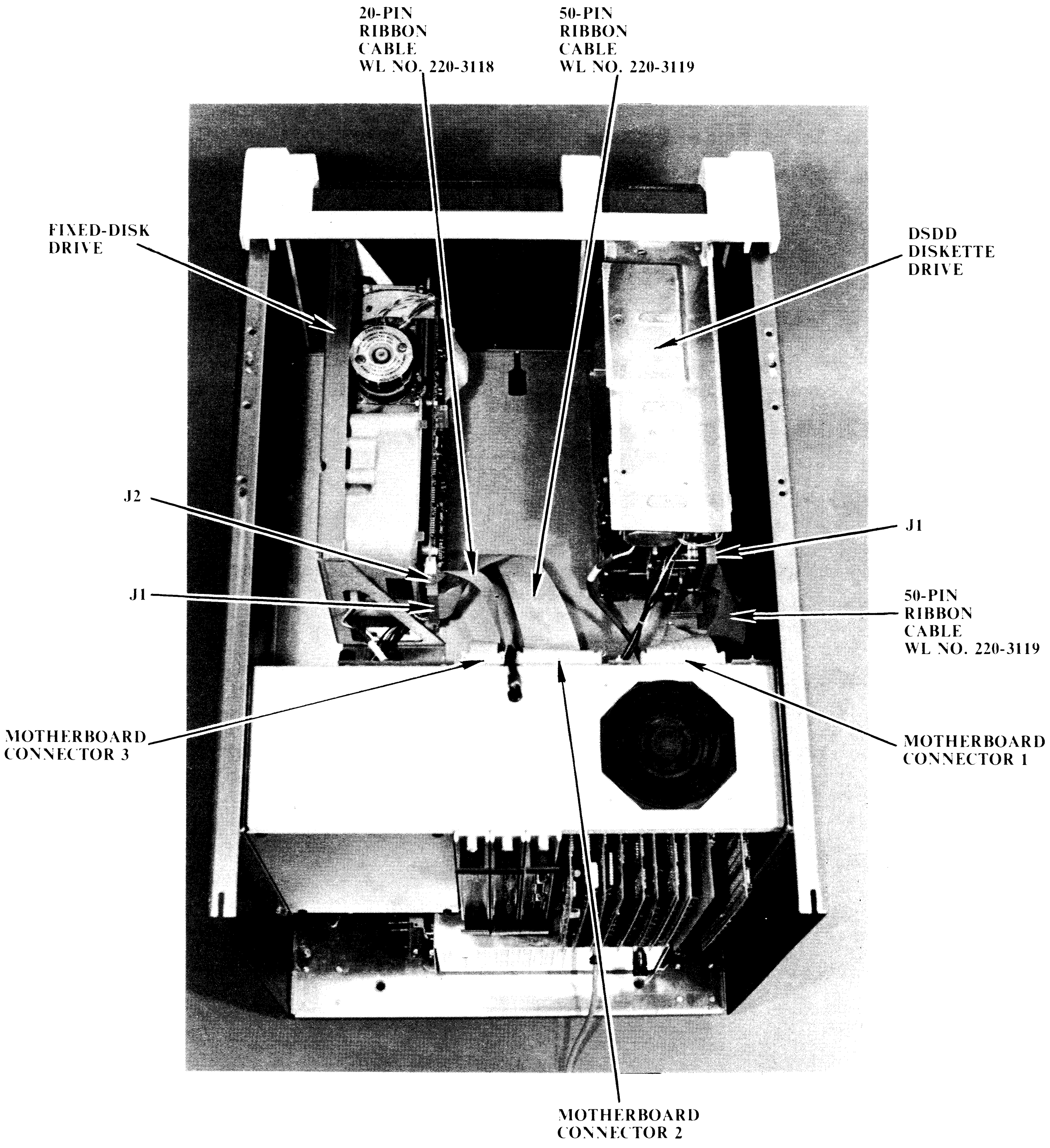


FIGURE 8-21 DISK DRIVE I/O CABLE CONNECTIONS

### 8.2.4 2200LVP POWER SUPPLY AC INPUT VOLTAGE SELECTION

There are two models of the 2200LVP power supply--one for 50 hertz applications and one for 60 hertz applications. The ac input voltage for both power supplies is switch selectable. The 115/230 input voltage selection switch is located on the rear of the power supply assemblies (ref: FIGURE 11-1). Be certain that the switch is positioned correctly for the supplied ac voltage.

### 8.2.5 2236MXD MULTIPLEXER/CONTROLLER

Refer to documentation category IV.B.1 for information concerning switch settings, PROM loading, etc.

### 8.2.6 22C32 TRIPLE CONTROLLER

Refer to documentation category IV.B.1 for information concerning switch settings, PROM loading, etc.

### 8.2.7 I/O CONTROLLERS

Refer to documentation category IV.B.1 for information concerning switch settings, PROM loading, etc.

### 8.2.8 2236DE INTERACTIVE TERMINAL

Refer to documentation category III.D.1 for information concerning unpacking, initial setup, adjustments, off-line diagnostic tests, system interconnection etc.

Refer to documentation category IV.B.1 for additional information concerning system interconnection.

### 8.2.9 DISK DRIVES

Refer to documentation categories III.A.11 and III.A.12 for information concerning initial setup, adjustments, off-line diagnostic tests, etc.

### 8.2.10 PERIPHERALS

Refer to appropriate documentation categories for information concerning unpacking, initial setups, adjustments, off-line diagnostic tests, system interconnection etc.

Refer to documentation category IV.B.1 for additional information concerning system interconnection.

## 8.3 INSTALLATION AND POWER-ON PROCEDURES

1. Ensure that the CPU power supply ac input voltage selection switch is positioned correctly for the supplied ac voltage (ref: Section 8.2.4).

2. Check to see that all power supply-to-CPU-to-disk power cables are firmly attached to the appropriate connectors (ref: Section 8.2.2).

3. Be certain that the CPU ac power switch (ref: FIGURE 8-1) is OFF, and then plug the CPU ac power cord in.

4. Check to see that all circuit boards are properly seated in their appropriate locations, and that all switches are set correctly (ref: Section 8.2).

5. Ensure that the DSDD diskette drive and the fixed disk are connected to the appropriate I/O jacks on the CPU chassis motherboard (ref: Section 8.2.3).

6. Remove the shipping diskette from the DSDD Diskette Drive. (Save the diskette for use when reshipping the drive/unit.)

7. Remove the shipping clamp that secures the Fixed-Disk Drive ac spindle motor. (This clamp is located on the side of the drive opposite the circuit board.) Save the clamp for use when reshipping the drive/unit.

8. Remove the spring clip that prevents the Fixed-Disk Drive head actuator damper from rotating. The spring clip fastens the tab on the actuator damper to the track 00 photocell. The damper can be readily identified by the yellow CAUTION label attached to it. Save the clip for use when reshipping the drive/unit.

9. Turn the CPU ac power switch ON, then check and adjust, if necessary, all CPU power supply voltages (ref: SECTION 11).

10. Turn the CPU ac power switch OFF.

11. Attach all system terminals and peripherals (ref: Section 8.2).

> <u>NOTE</u>:
>
> If peripheral I/O cables are routed through conduit, ceilings, walls, or floors, it will be necessary to install amphenol connectors on the ends of those cables. The procedure for amphenol connector installation is documented in category I.B.0.

11. Turn the terminal ac power switch(es) ON; turn the CPU ac power switch ON; turn the ac power switches of all peripherals ON.

12. At this point, the terminal connected to MXD channel #1 should have the "MOUNT SYSTEM PLATTER--PRESS RESET" prompt displayed (ref: SECTION 3). If this message is not displayed (possibly due to a malfunction), turn the CPU ac power switch OFF. After 2 or 3 seconds, turn the switch back ON. If the message is still not displayed, refer to SECTIONS 3 and 12.

13. After the power-on prompt is displayed, insert the 2200LVP Operating System diskette (WL #704-0002) into the DSDD diskette drive, and then press RESET on the keyboard of terminal #1.

14. The prompt "KEY SF'?" should now be displayed (ref: SECTION 3). If this message is not displayed, refer to SECTIONS 3 and 12.

15. After the "KEY SF'?" prompt is displayed, load and run the LVP User and Field Service diagnostics (ref: SECTION 9). If the system will not load diagnostic programs, refer to SECTIONS 3 and 12. If any diagnostic errors occur, refer to SECTION 12 for interpretations.

16. After successful completion of all Microcode diagnostics, load BASIC-2 (ref: SECTION 3), and configure the system such that all terminals evenly share the available user memory (ref: SECTION 4). (Edit the Master Device Table so that it is correct for the user's peripherals.)

17. When the system configuration has been generated, load and run the BASIC-2 Language diagnostics (ref: SECTION 9).

18. Upon completion of the BASIC-2 diagnostics, load and run the appropriate peripheral diagnostics (ref: SECTION 9).

19. After all peripherals are proven to be operational, format the fixed-disk drive by loading and running the format utility program "@FORMAT" (resident on the Operating System diskette).

20. When formatting has been completed, load and run the disk diagnostics (ref: SECTION 9).

21. Check all DSDD Diskette Drive adjustments/alignments to ensure that they are correct (ref: documentation category III.A.11).

22. Verify that all DPU adjustments are correct (ref: Section 11.3).

23. The system is now ready for customer use.

## NOTES

## SECTION 9
## DIAGNOSTICS

Following is a list of documentation categories referenced by this section. Diagnostic information in these categories is required to fully test a 2200LVP system.

2200LVP CPU, any disk drives, and any peripherals -- IV.C.1
DPU -- IV.A.3
2236DE Terminal -- III.D.1

### CPU Diagnostics

There are three classes of diagnostic tests available for the 2200LVP CPU: 1) "BOOTSTRAP" diagnostics (resident in the 2200LVP firmware), 2) Microcode diagnostics (contained on the 2200LVP Operating System diskette), and 3) BASIC-2 Language diagnostics (available on diskette). Refer to SECTION 3 of this manual for an explanation of the BOOTSTRAP diagnostics. Refer to documentation category IV.C.1 for information concerning the Microcode and BASIC-2 Language diagnostics.

### DPU Diagnostics

The Disk Processing Unit has a built-in power-on diagnostic. If a failure is detected, the activity LED in the door latch release button of the DSDD Diskette Drive will blink on and off. As of July, 1980, the diagnostic program and the DPU boards are not finalized. Refer to documentation category IV.A.3 for later developments on this system element.

### Terminal Diagnostics

Refer to documentation category III.D.1 for information concerning 2236DE Terminal power-on diagnostics.

## Disk Diagnostics

Refer to documentation category IV.C.1 for information concerning Disk diagnostics.

## Peripheral Diagnostics

Refer to documentation category IV.C.1 for information concerning Peripheral diagnostics.

## NOTES

## NOTES

## SECTION 10
## PREVENTIVE MAINTENANCE

To ensure trouble-free operation the 2200LVP must have periodic preventive maintenance (PM), consisting of inspection, cleaning, and adjustments. Since the DSDD Diskette Drive requires PM once a year minimally, that becomes the minimum criteria for the LVP mainframe. Certain peripherals attached to the LVP mainframe may require more frequent PM. Therefore, refer to documentation category I.A.4 for information concerning PM for the 2200LVP CPU.

## NOTES

# SECTION 11

# REMOVAL/REPLACEMENT AND ADJUSTMENT PROCEDURES

## 11.1 RECOMMENDED TEST EQUIPMENT/TOOL LIST

1. Digital Voltmeter (WL #726-9595), with an accuracy of at least $\pm$ 1% of full scale, and 1 mv resolution factor. Analog Multimeters have accuracy and resolution factors that are unacceptable for certain critical measurements.

   Acceptable Type/Equivalent: FLUKE #8000A

2. Multimeter, 20,000 ohms/volt (minimum); 2% or better full scale accuracy; for less critical measurements.

   Acceptable Type/Equivalent: TRIPLETT VOM #630NA

3. Oscilloscope, with two X1 probes and two X10 probes.

   Acceptable Type/Equivalent: TEKTRONIX #465

4. Heavy duty magnetic screwdriver with well-insulated handle (WL #726-9411).

5. Small screwdriver with insulated shank (WL #726-9406).

6. 5/16" nut driver (WL #726-9473).

7. Nut driver handle (WL #726-9478).

8. A 4-inch length of jumper wire.

## 11.2 CPU VOLTAGE CHECK/ADJUSTMENT PROCEDURE

1. Turn the CPU ac power switch OFF.

2. Remove the cabinet top cover (ref: Section 11.4.1).

3. Turn the CPU ac power switch ON.

4. Check the dc voltages with a digital voltmeter for the values listec in TABLE 11-1. (The test points for monitoring the voltages are shown in FIGURE 8-16. The test points are accessible from the circuit side of the motherboard. To check +24V, monitor J5 pin 1 of both the diskette and fixed-disk drives (ref: FIGURE 8-19). The yellow wires in the CPU motherboard-to-disk drive dc power harness connect to those pins.) Adjust the trimpots where indicated in FIGURE 11-1 to obtain correct voltage levels where necessary.

   IMPORTANT:

   Be sure to connect the COMMON lead of the voltmeter to a ± 0V connection, NOT the chassis or I/O controller rail. Erroneous readings will result if chassis ground is used as the voltmeter reference. The oscilloscope ground clip should also be attached to ± 0V, NOT chassis ground.

5. Using an oscilloscope, with the vertical sensitivity set at 5V/cm, and a X1 probe, measure the ripple at the points indicated in FIGURE 8-16. No ac ripple should be observed. If any voltage or ripple measurement is out of specification, troubleshoot the CPU power supply.

6. Note that when increasing RAM capacity by field conversion, or when adding extra I/O capabilities to the CPU, all voltages must be rechecked and readjusted when necessary.

TABLE 11-1 DC VOLTAGE SPECIFICATIONS

| VOLTAGE | LIMITS |
|---|---|
| +5V1 * | +4.95 to +5.05 |
| +5V2 ** | +4.95 to +5.05 |
| +12V | +11.95 to +12.05 |
| +24V | +21.60 to +26.40 |
| -5V | -4.95 to -5.05 |
| -12V *** | -11.50 to -12.50 |

* If +5V1 drops below +4.7V dc, +24V will be shut off.

** +5V2 is only used in the 9 I/O-slot version of the LVP-- +5V2 supplies the last 6 I/O slots. This voltage does not have to be correct for the 3-slot version of the LVP.

*** -12V is not adjustable.

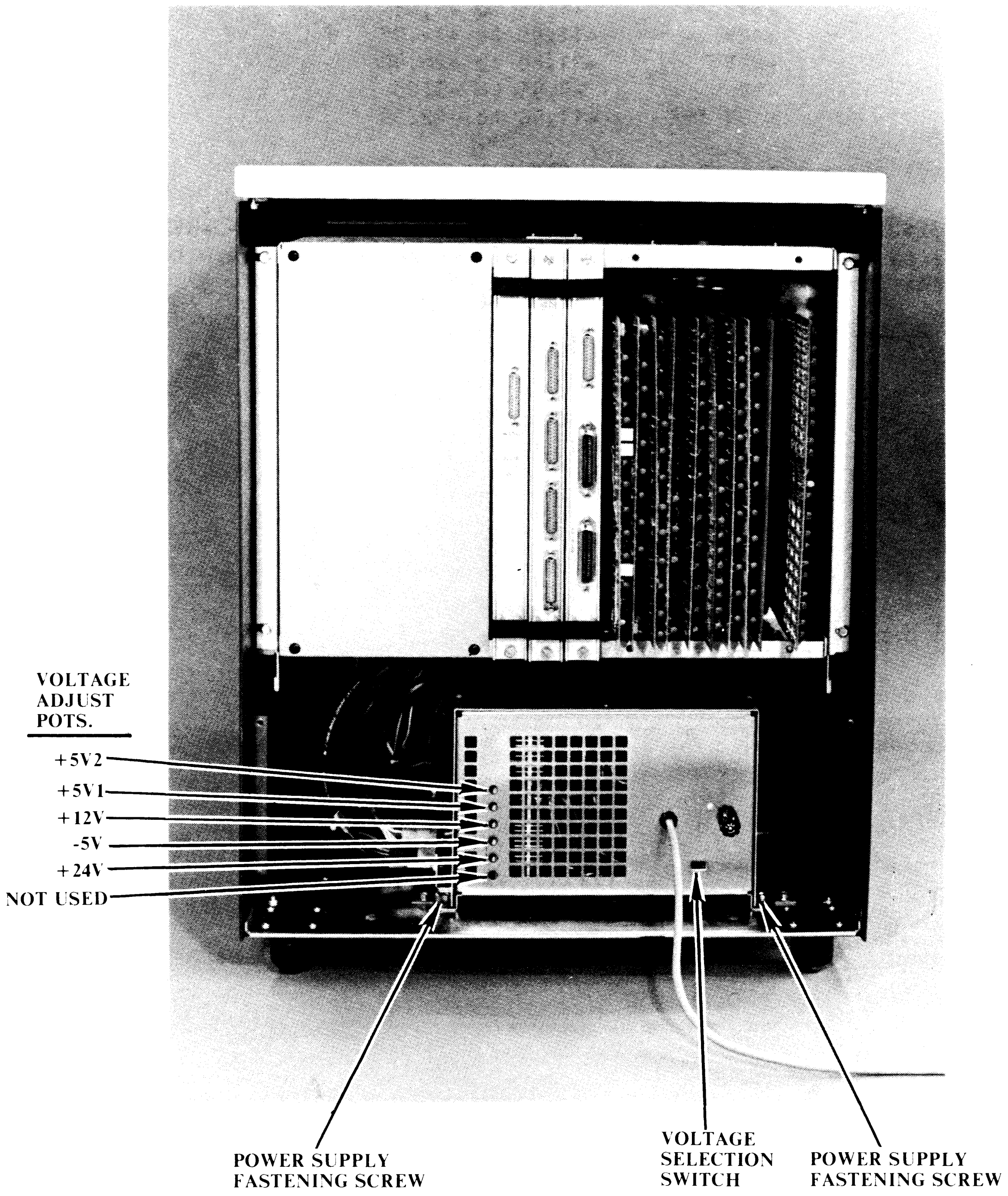


FIGURE 11-1 POWER SUPPLY REGULATOR ADJUSTMENT POTENTIOMETERS

## 11.3 DISK PROCESSING UNIT ADJUSTMENT PROCEDURE

This section explains the procedure for adjusting the phase-locked loop in the Disk Processing Unit (DPU). This adjustment should be performed whenever a disk read/write problem is suspected.

1. Check and adjust (if necessary) the CPU power supply voltages (ref: Section 11.2), and then turn the CPU ac power switch OFF.

2. Remove the cabinet top cover, and the CPU chassis cover (ref: Section 11.4).

NOTE:

Do not place the 210-7694 2200/Disk Interface board on an extender board when performing this adjustment.

3. Using a 4-inch length of jumper wire, connect L8 pin 3 on the 210-7694 board to $\pm 0$V (ref: FIGURE 11-2).

4. Connect the Channel 1 probe of the oscilloscope to L5 pin 1 or 2 on the 210-7694 board (ref: FIGURE 11-2).

5. Set the oscilloscope such that a +3.0V dc level can be observed. Be sure to ground the oscilloscope probe.

6. Turn potentiometers R7 and R8 on the 210-7694 board to their midrange points, and turn potentiometer R16 on the 210-7694 board fully counterclockwise (ref: FIGURE 11-2).

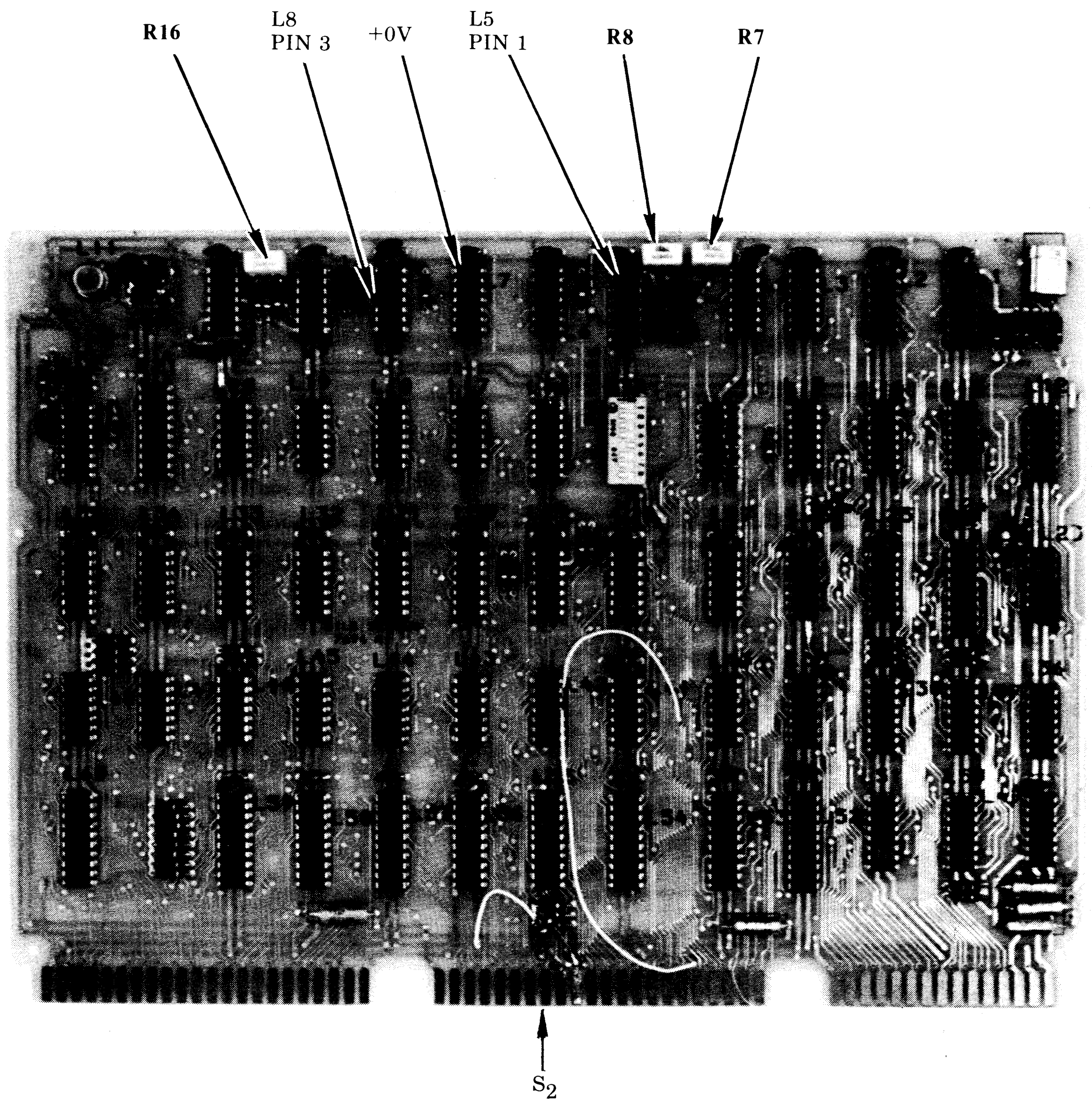


**FIGURE 11-2 DPU ADJUSTMENT TEST POINTS AND POTENTIOMETERS ON 210-7694 BOARD**

7. Turn the CPU ac power switch ON.

8. Adjust R16 to obtain a +2.5V to +3.0V (preferably +3.0V) dc level (ref: FIGURE 11-2A). Some oscillation (noise) may be noticed. Adjust R16 until the oscillation is minimized and the dc level is between +2.5V and +3.0V. (Lowering the dc level reduces the amount of oscillation.)

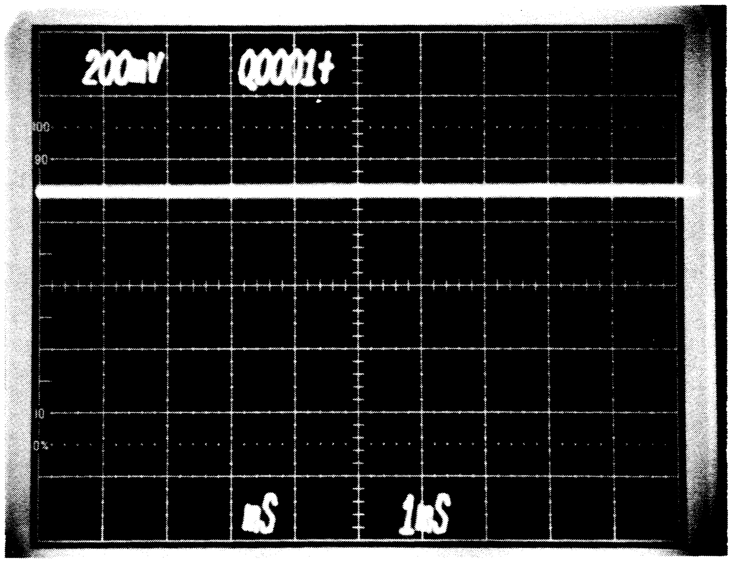


**FIGURE 11-2A +3.0V DC LEVEL**

9. Carefully remove the jumper wire from L8 pin 3.

10. Connect the oscilloscope external trigger probe to $S_2$ (INDEX) on the 210-7694 board (ref: FIGURE 11-2). ($S_2$ can be reached on the circuit side of the motherboard.)

11. Set the oscilloscope as follows:
    Trigger Source: External
    Trigger Mode: Normal (DC)
    Trigger Slope: Negative
    Time Base: 100 usec/div
    Vertical Sensitivity: 2V/cm
    Input Coupling: DC
    Ground Reference Point: Center Line

12. Insert an Operating System diskette into the DSDD drive, and depress RESET on the system console.

13. Depress SF'01 on the system console to load the Operating System from B10 (removable disk).

14. While watching the system console screen, adjust R8 until the "Loading BASIC-2" prompt is observed (if it is not already). Repeat steps 12 and 13 as needed.

15. If the Operating System will not completely load, fine tune R8 to obtain a waveform similar to FIGURE 11-2E (located in the R8 adjustment section following). Repeat steps 12 and 13 as needed. FIGURES 11-2F and 11-2G (also located in the R8 adjustment section following) show typical waveforms when R8 is turned too far counterclockwise (FIGURE 11-2F) or too far clockwise (FIGURE 11-2G).

NOTE:

It may be helpful to listen to the diskette retries (heads stepping back to track 0 on a reseek) while trying to load the Operating System. By fine tuning R8 to reduce the number of retries, the Operating System should load. Continue adjusting R8 until the Operating System does load.

16. After the Operating System is loaded, remove the diskette, and enter the following program on the system console:

```
10 VERIFYF(X,X):PRINT".";:GOTO 10
```

```
where X = 8127 if the the Fixed-Disk Drive is 2MB
        = 16319 if the Fixed-Disk Drive is 4MB
        = 32639 if the Fixed-Disk Drive is 8MB
```

17. Depress RETURN, RUN, RETURN to load and run the verify program. (The program verifies the last sector on the Fixed-Disk Drive, and prints a "period" on the system console when the operation is completed. An ERROR message is displayed if a read error is detected.)

18. Adjust R7 to obtain a waveform similar to FIGURE 11-2B. Adjust R7 to minimize noise, as in FIGURE 11-2B. FIGURES 11-2C and 11-2D show typical waveforms when R7 is turned too far counterclockwise (FIGURE 11-2C) or too far clockwise (FIGURE 11-2D).

NOTE:

If R7 is adjusted properly, FIGURE 11-2B will be observed, and the "periods" (.) from the verify program in step 16 will be displayed at a constant rate of speed. If R7 is turned in either direction (CW or CCW) until a verify ERROR occurs, either FIGURE 11-2C or 11-2D will be observed (dependent on direction). Watching the rate at which the "periods" (or the "ERRORs") are being displayed may help in setting R7 correctly. It may be helpful to purposefully create errors (by turning R7 to its extremes) in order to see what the waveform (noise) looks like when R7 is maladjusted.

19. When it appears that R7 is adjusted properly, change the oscilloscope Vertical Sensitivity to 1V/cm.

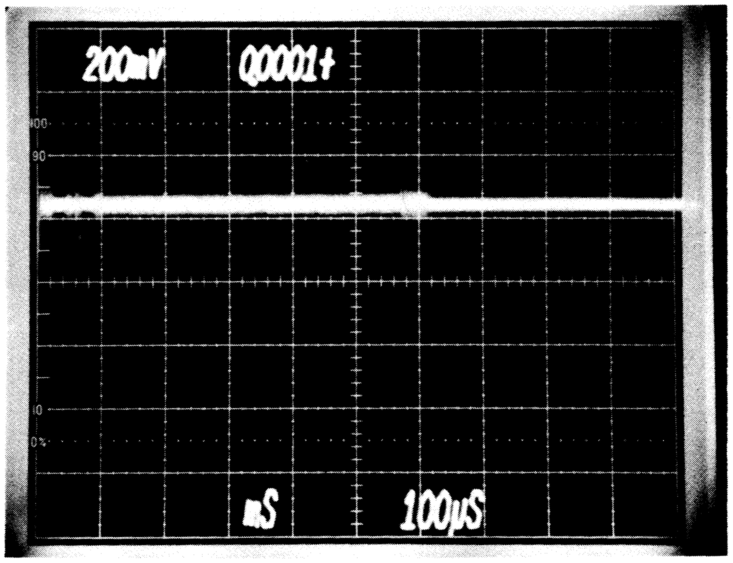


**FIGURE 11-2B R7 ADJUSTED PROPERLY**

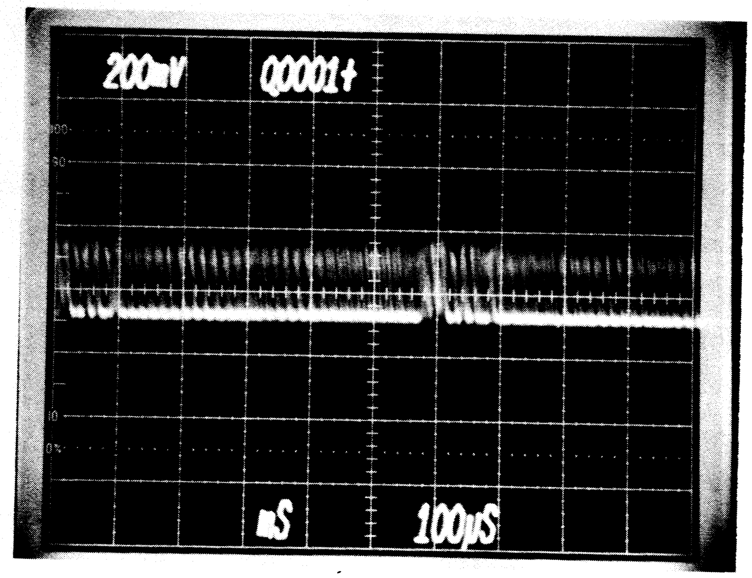


FIGURE 11-2C  R7 SET TOO FAR COUNTERCLOCKWISE

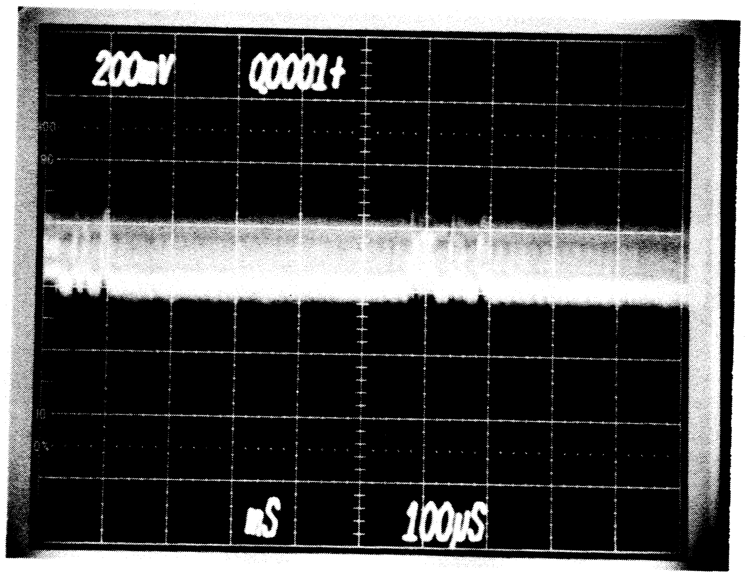


FIGURE 11-2D  R7 SET TOO FAR CLOCKWISE

20. Carefully (slightly) turn R7 in both directions (CW and CCW) taking note of the exact amplitudes where the signal starts to become noisy. (There will be about a .6V difference between the amplitudes.) Adjust R7 such that the waveform amplitude is in the middle of this .6V range.

21. After R7 is set properly, verify the entire fixed disk to ensure complete operation of the DPU. (Actually, the only way to guarantee complete operation is by formatting--with certification--the fixed disk. IF, AND ONLY IF, THE CUSTOMER WILL ALLOW THIS, then do so.

CAUTION:

If R7 appears to be adjusted properly (ref: FIGURE 11-2B), yet verify ERRORs still occur, the fixed-disk drive--or the disk sector that is being verified--is probably bad. Verify some other sectors on the fixed-disk to determine whether the drive or the platter is bad.

22. Insert a known-good, formatted diskette into the DSDD drive, and then enter the following program on the system console.

```
10 VERIFYR(3977,3977):PRINT".";:GOTO 10
```

23. Depress RETURN, RUN, RETURN to load and run the verify program. (The program verifies the last sector on the DSDD Diskette Drive, and prints a "period" on the system console when the operation is completed. An ERROR message is displayed if a read error is detected.)

24. Set the oscilloscope Vertical Sensitivity to 2V/cm.

25. Adjust R8 to obtain a wave form similar to FIGURE 11-2E. Adjust R8 until noise on the waveform is slight, as in FIGURE 11-2E. FIGURES 11-2F and 11-2G show typical wave forms when R8 is turned too far counterclockwise (FIGURE 11-2F) or too far clockwise (FIGURE 11-2G). (It may be possible to adjust R8 such that a waveform that has less noise than the one shown in FIGURE 11-2E is obtained. However, errors may occur at this setting, because the adjustment of the loop will be too close to the "noise limit"--the point where the clean waveform begins to break up).

NOTE:

If R8 is adjusted properly, FIGURE 11-2E will be observed, and the "periods" (.) from the verify program in step 22 will be displayed at a constant rate of speed. If R8 is turned in either direction (CW or CCW) until a verify ERROR occurs, either FIGURE 11-2F or 11-2G will be observed (dependent on direction). Watching the rate at which the "periods" (or the "ERRORS") are being displayed may help in setting R8 correctly. Again, it may be helpful to purposefully create errors (by turning R8 to its extremes) in order to see what the waveform (noise) looks like when R8 is maladjusted.

26. After R8 is set properly, verify the entire diskette to ensure complete operation of the DPU.

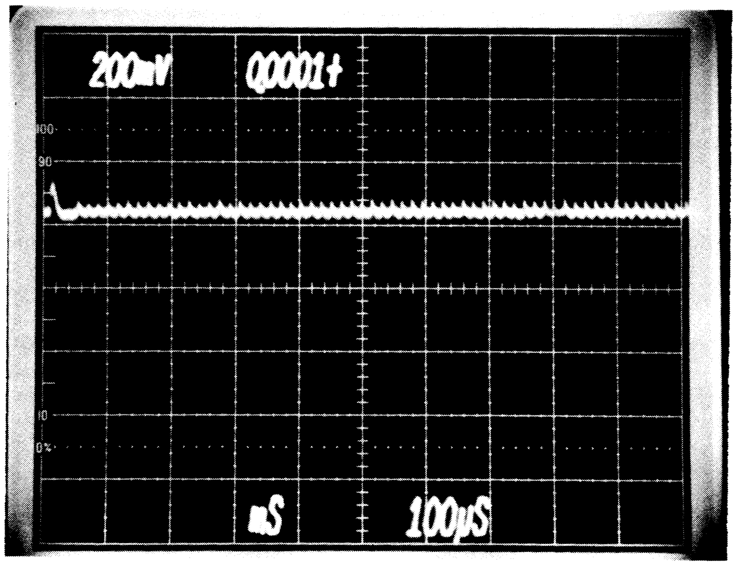


FIGURE 11-2E R8 ADJUSTED PROPERLY

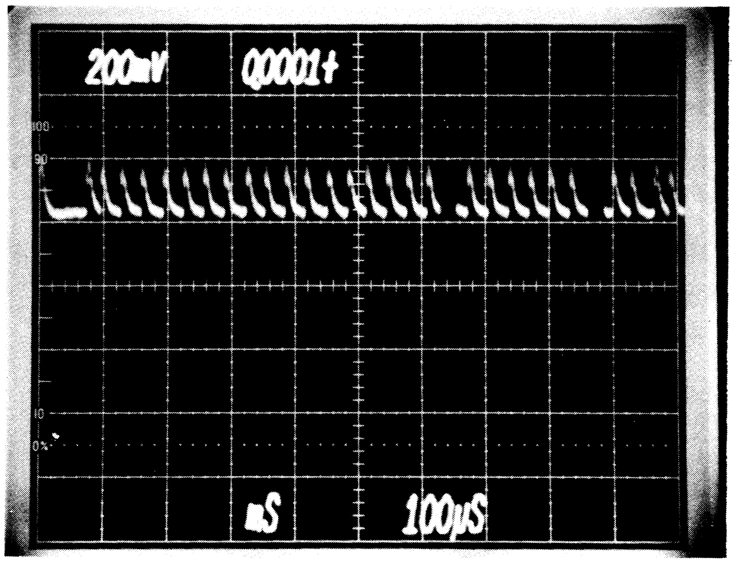


FIGURE 11-2F  R8 SET TOO FAR COUNTERCLOCKWISE

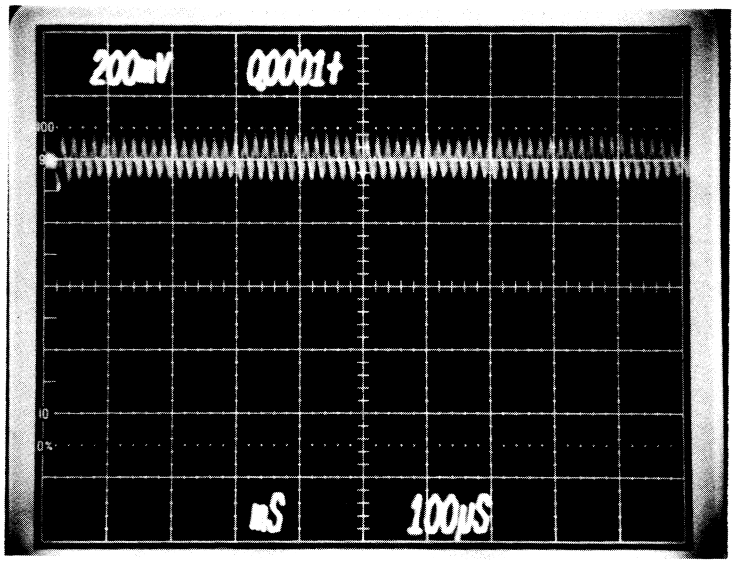


FIGURE 11-2G  R8 SET TOO FAR CLOCKWISE

## 11.4 REMOVAL/REPLACEMENT PROCEDURES

### 11.4.1 CABINET TOP COVER

1. Using a 5/16" nut driver (WL #726-9473), remove the two screws (ref: FIGURE 11-3, items A) from the rear-underneath sides of the top cover.

2. Lift the top cover out of the two snap locks (ref: FIGURE 11-4) and off the unit.

### 11.4.2 CABINET BACK PANEL

Remove the four screws (ref: FIGURE 11-3, items B) securing the back panel and remove that panel.

### 11.4.3 CPU CHASSIS COVER

Remove the four screws (ref: FIGURE 11-3, items C) securing the CPU chassis cover and remove the cover.

### 11.4.4 CPU CHASSIS

1. Remove the cabinet top cover (ref: Section 11.4.1).

2. Disconnect the fan cord (ref: Section 8.2.2).

3. Disconnect the power-on LED cable.

4. Disconnect the three disk drive I/O ribbon cables (ref: FIGURE 8-21) from the CPU motherboard.

5. Disconnect the two CPU motherboard-to-power supply dc power harnesses (ref: FIGURE 8-18) from the power supply.

6. Disconnect the CPU motherboard-to-disk drive dc power harness (ref: FIGURE 8-17) from the CPU motherboard.

7. Remove the four screws (ref: FIGURE 11-3, items D) securing the CPU chassis and lift the chassis out of the cabinet.

## 11.4.5 POWER SUPPLY

1. Remove the cabinet back panel (ref: Section 11.4.2).

2. Disconnect the fan cord (ref: Section 8.2.2).

3. Disconnect the two disk drive ac power cords (ref: FIGURE 8-19) from the disk units.

4. Disconnect the two CPU motherboard-to-power supply dc power harnesses (ref: FIGURE 8-18) from the power supply.

5. Remove the two screws (ref: FIGURE 11-1) from the rear sides of the power supply and pull the power supply (from the rear) out of the cabinet.

## 11.4.6 POWER SUPPLY COVER

Remove the screws securing the power supply cover and remove the cover.

## 11.4.7 POWER SUPPLY REGULATOR

1. Remove the power supply and power supply cover (ref: Sections 11.4.5 and 11.4.6).

NOTE:

There are two 3-pin connectors (J3 and J5) on the regulator board. Note the orientation of the two cables connected to these jacks before performing the following step. When installing a regulator, refer to the interconnection diagram in Appendix C to ensure that the correct cable is connected to each of these jacks.

2. Disconnect the power harnesses attached to regulator board jacks J1-J5 (ref: FIGURE 8-20).

3. Remove the screws (ref: FIGURE 8-20) securing the regulator board to the power supply chassis and remove the regulator board.

## 11.4.8 DISK DRIVES

NOTE:

Refer to documentation categories III.A.11 (SA851) and III.A.12 (SA1000) for information concerning disk drive handling procedures.

1. Remove the cabinet top cover (ref: Section 11.4.1).

2. Disconnect the two ac power cords (ref: FIGURE 8-19) from the disk drives.

3. Disconnect the dc power harnesses (ref: FIGURE 8-19) from the disk drives.

4. Disconnect the I/O ribbon cables (ref: FIGURE 8-19) from the disk drives.

CAUTION:

Be careful not to damage the motherboard when performing the following step.

5. Remove the 5/16" screws (ref: FIGURE 11-5) from the rear of the drive mounting plates and lift the drives back, up and out of the cabinet.

6. When replacing a disk drive, insert the plastic mounting/guide block, located on the bottom of the drive, into the mounting hole in the cabinet shelf, and then push the drive toward the front of the cabinet until the drive is seated properly and the fastening-screw holes line up.

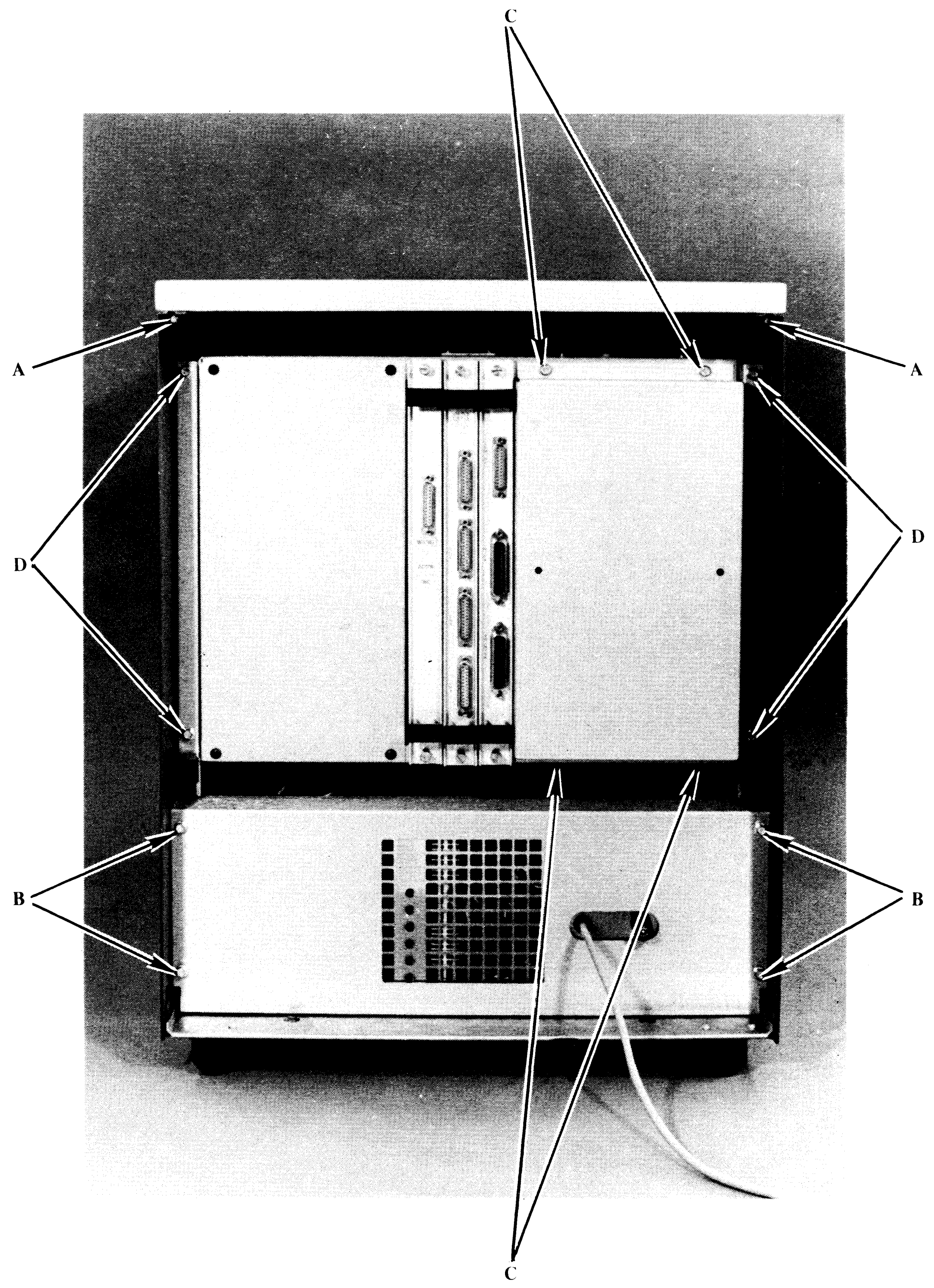


FIGURE 11-3 FASTENING SCREWS

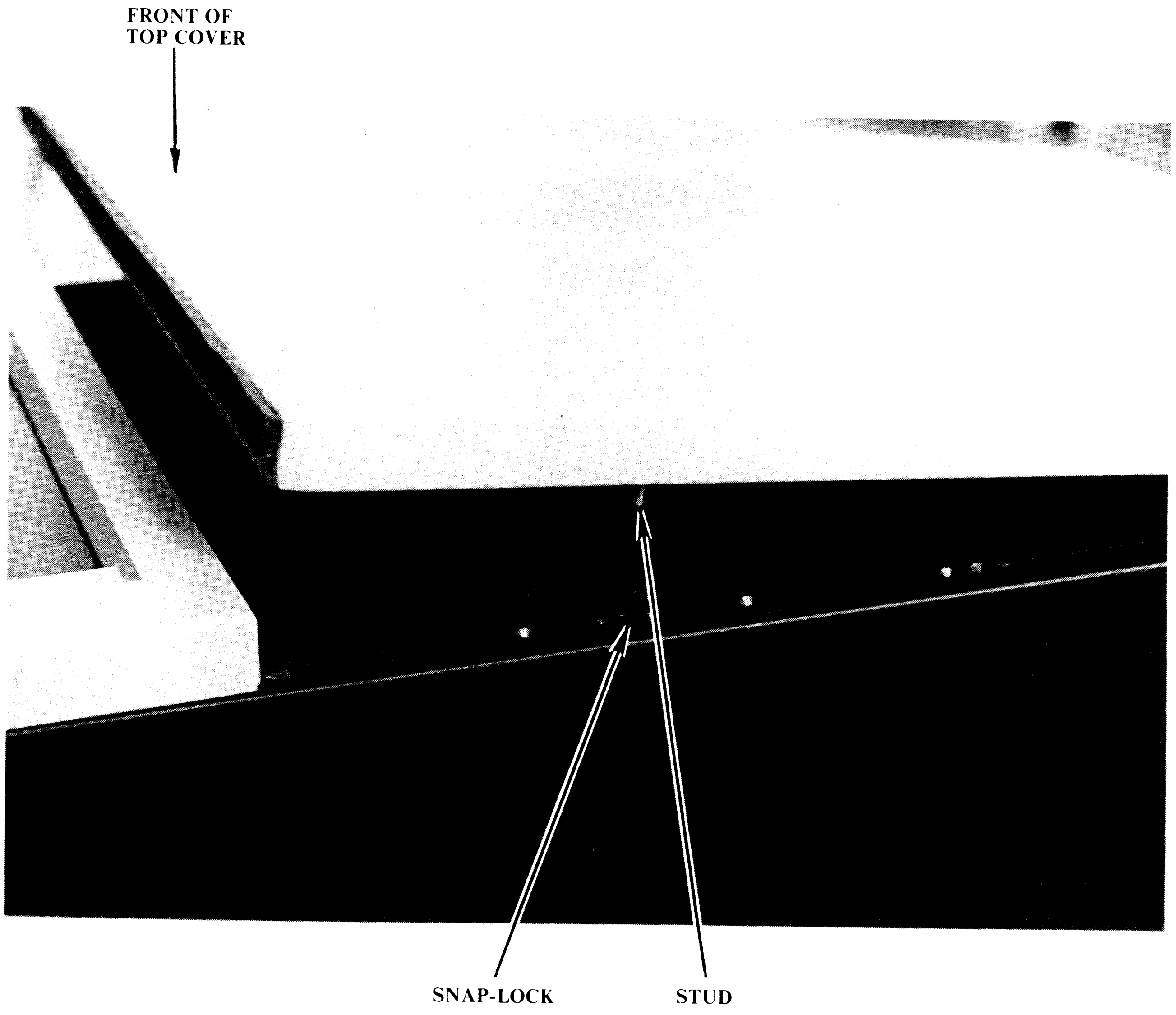


FIGURE 11-4 TOP COVER SNAP-LOCKS

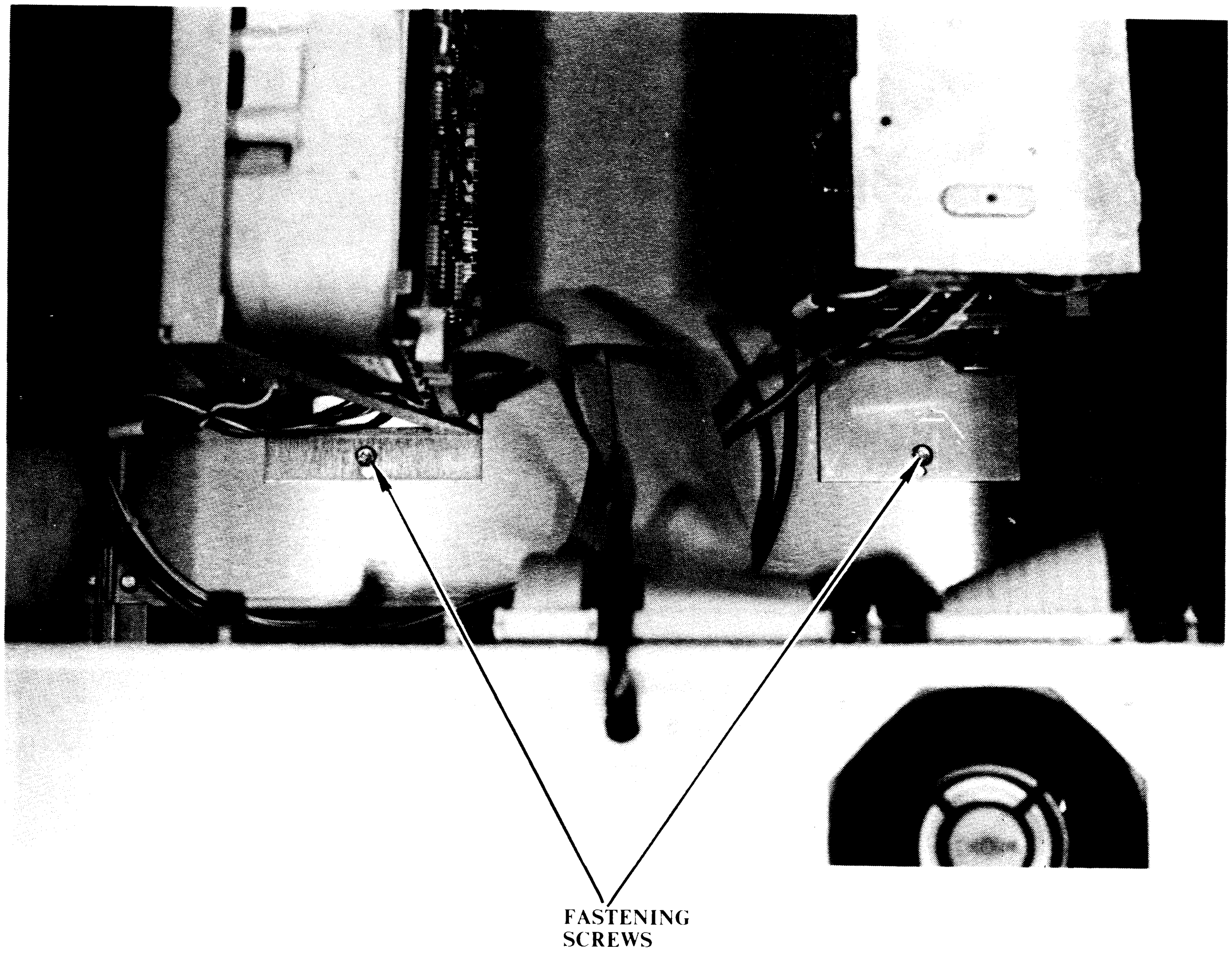


FIGURE 11-5 DISK DRIVE FASTENING SCREWS

## NOTES

SECTION 12

# TROUBLESHOOTING

## 12.1 GENERAL

This section provides troubleshooting aids that will be helpful in identifying the more common 2200LVP faults. Use a logical approach to troubleshoot the system: observe the problem symptoms carefully, and then isolate the problem by logical deduction.

NOTE:
Be certain to verify or install all required ECN's.

From a system standpoint, troubleshooting the CPU involves a relatively simple procedure. The following steps should be performed.

1. Remove all peripheral controllers from the CPU. Check the address switch settings (ref: SECTION 8). Ensure that all cables from peripherals to controllers are correct and secured. Replace only the 2236MXD and the disk controllers(s). Check all voltages for proper levels (ref: SECTION 11). If the problem persists, continue with ROUTINE A (in Troubleshooting Flowchart to follow in this section). Otherwise continue with step 3.

2. If the problem persists, replace each board presently in the CPU (and I/O) with a known good board (latest E-REV) until the problem disappears (never rule out the possibility of mutiple problems). If the problem still persists, there may be a software problem.

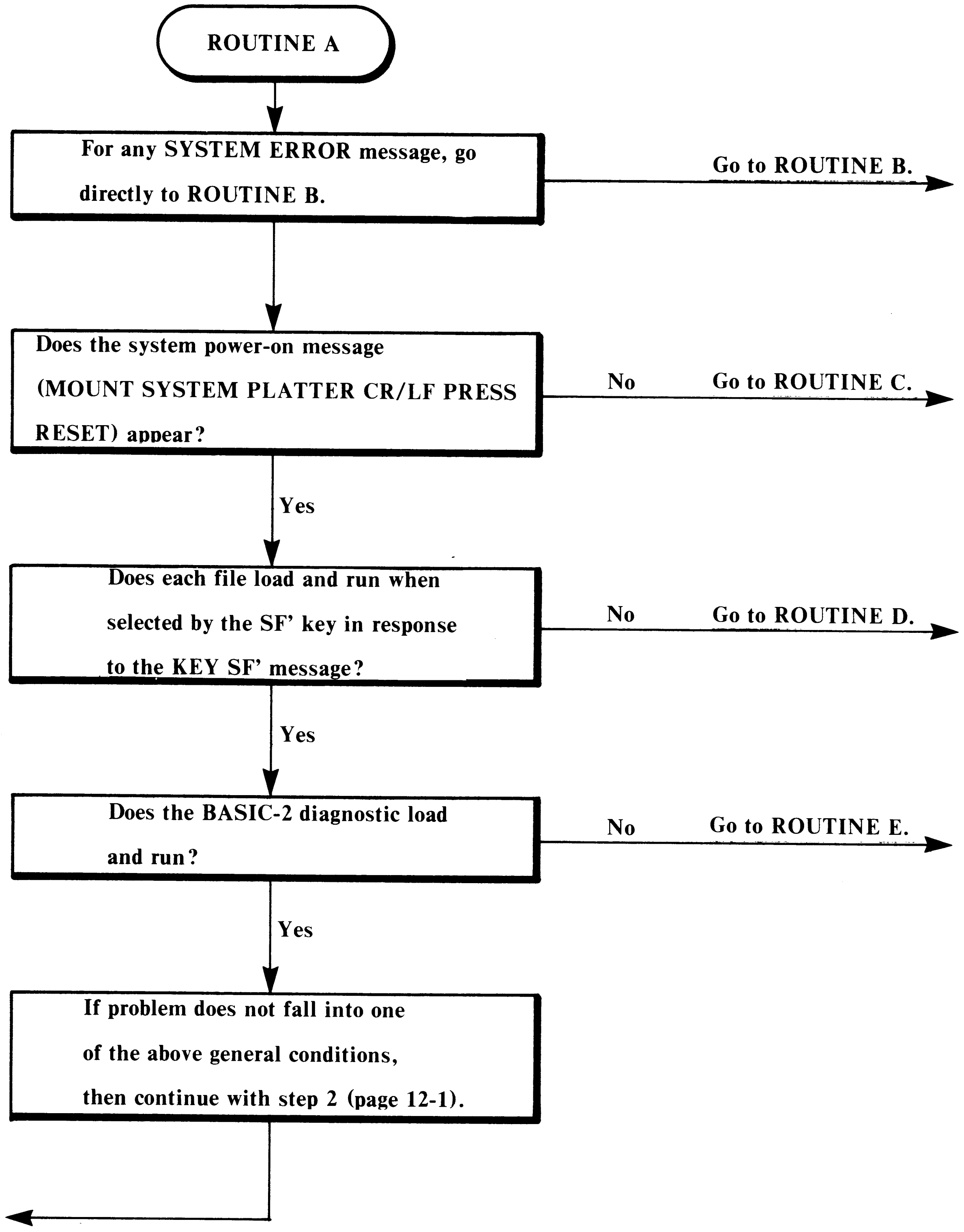
3. Once the problem has been removed, run all System and BASIC-2 Diagnostics. If any further errors are discovered from these diagnostics, follow the procedures outlined in the 2200LVP Troubleshooting Flowchart (following). Otherwise, continue with step 5.

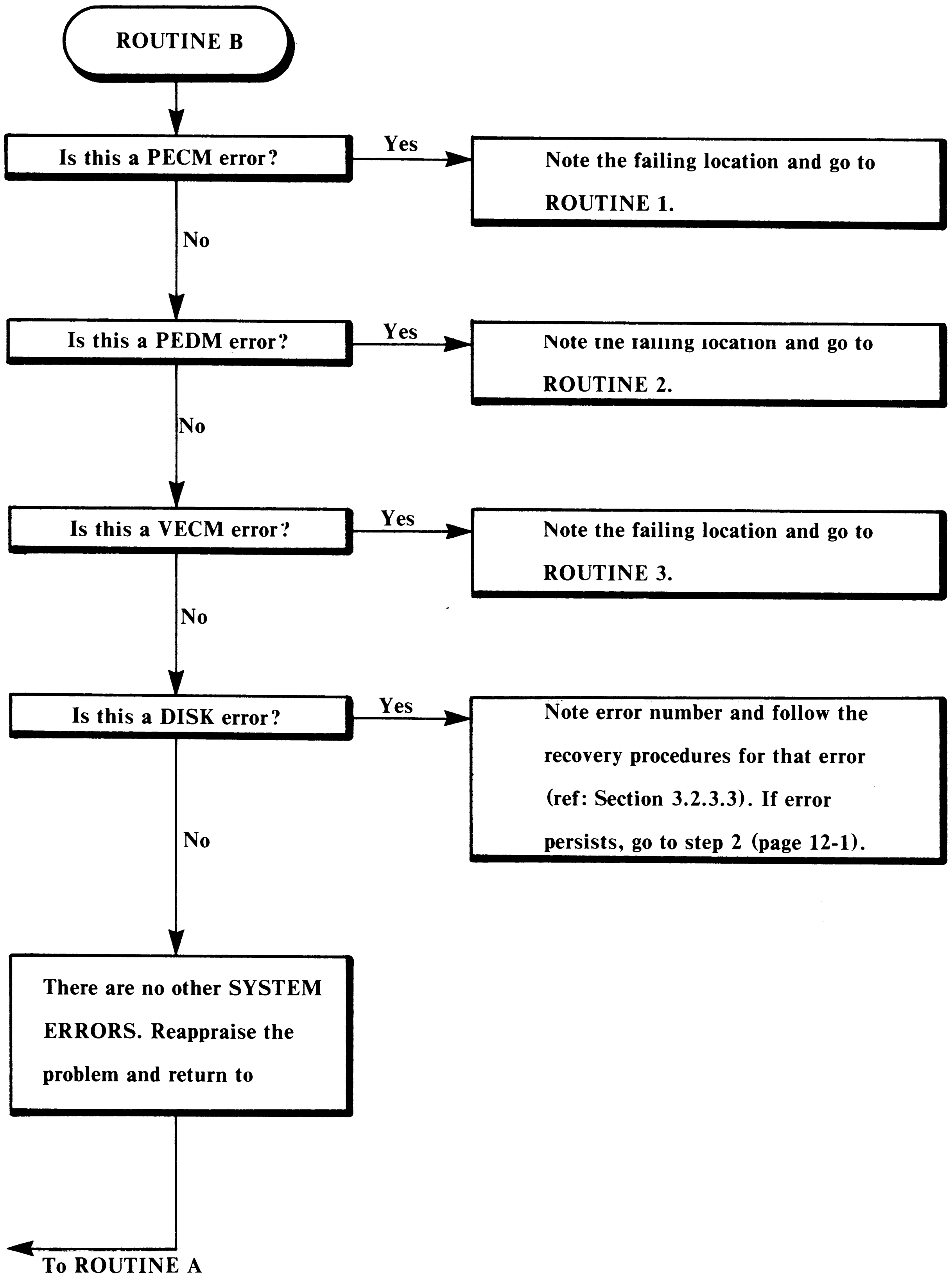
4. Replace only the suspected bad peripheral controller with a known good one in the CPU. If the problem recurs and appears to be in the peripheral, troubleshoot that peripheral according to the procedures given in the specific maintenance manual for that peripheral.

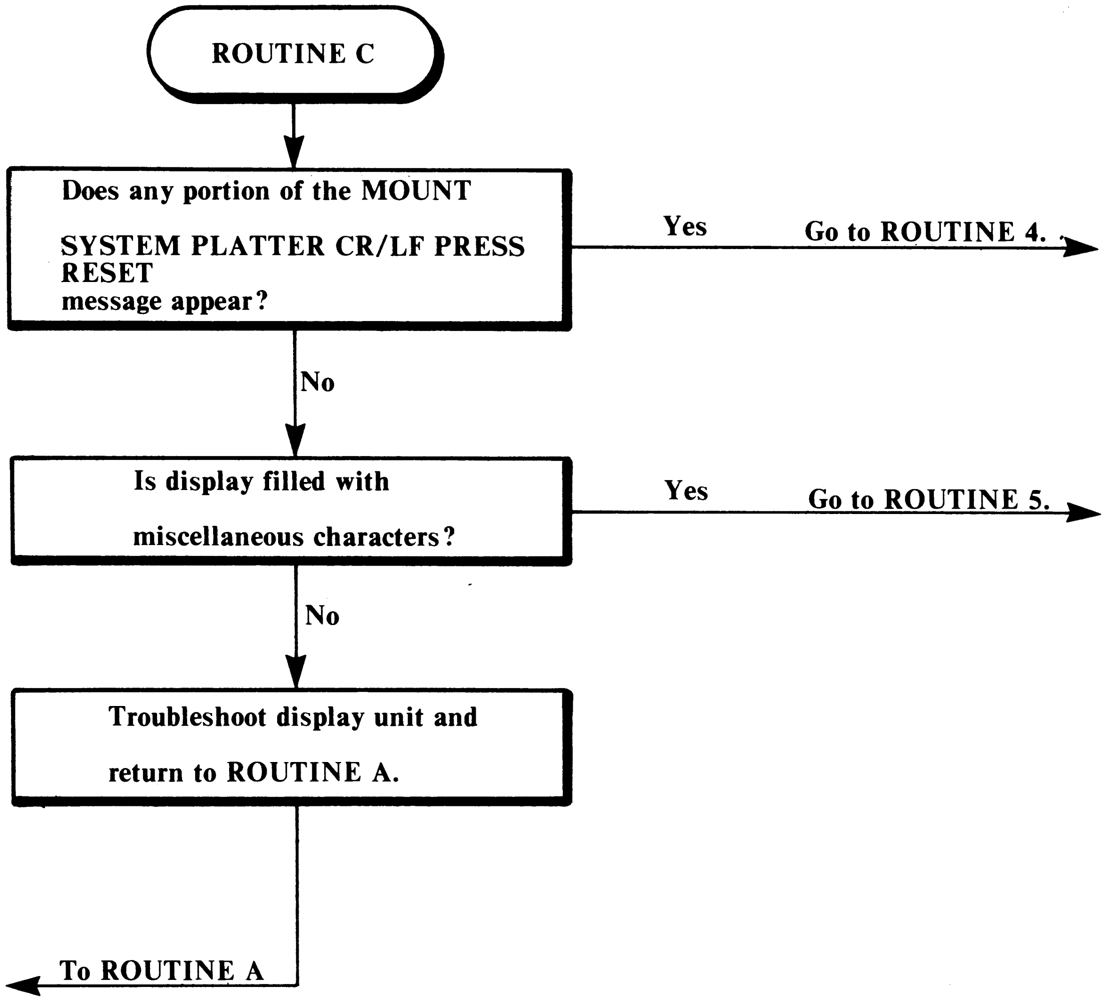
5. Plug all peripheral controllers into the CPU and recheck all voltages. Run all system and peripheral diagnostics to ensure that the system operates properly in its final configuration.

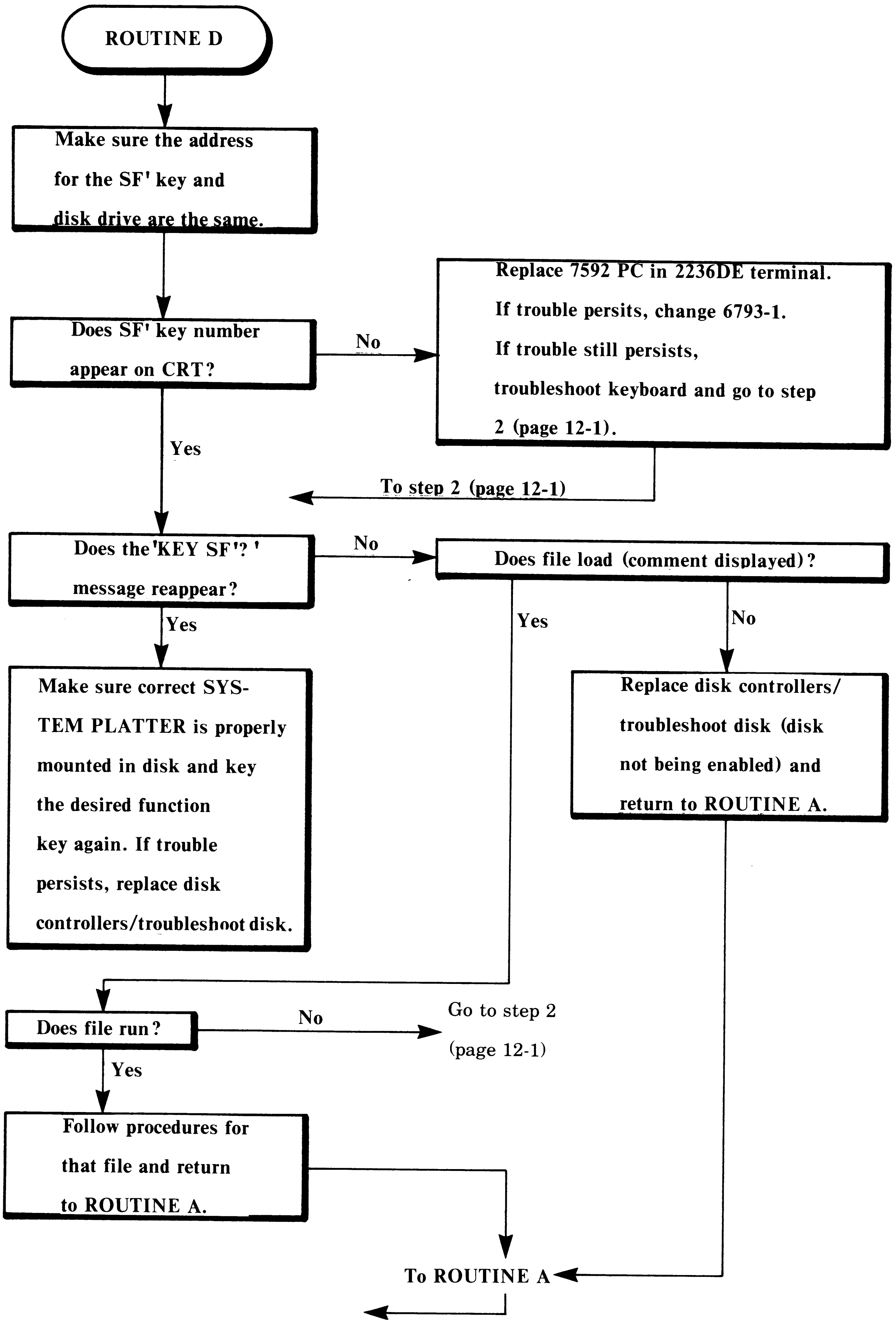
The 2200MVP Troubleshooting Flowchart, starting on the next page, presents a logical approach to troubleshooting the system.

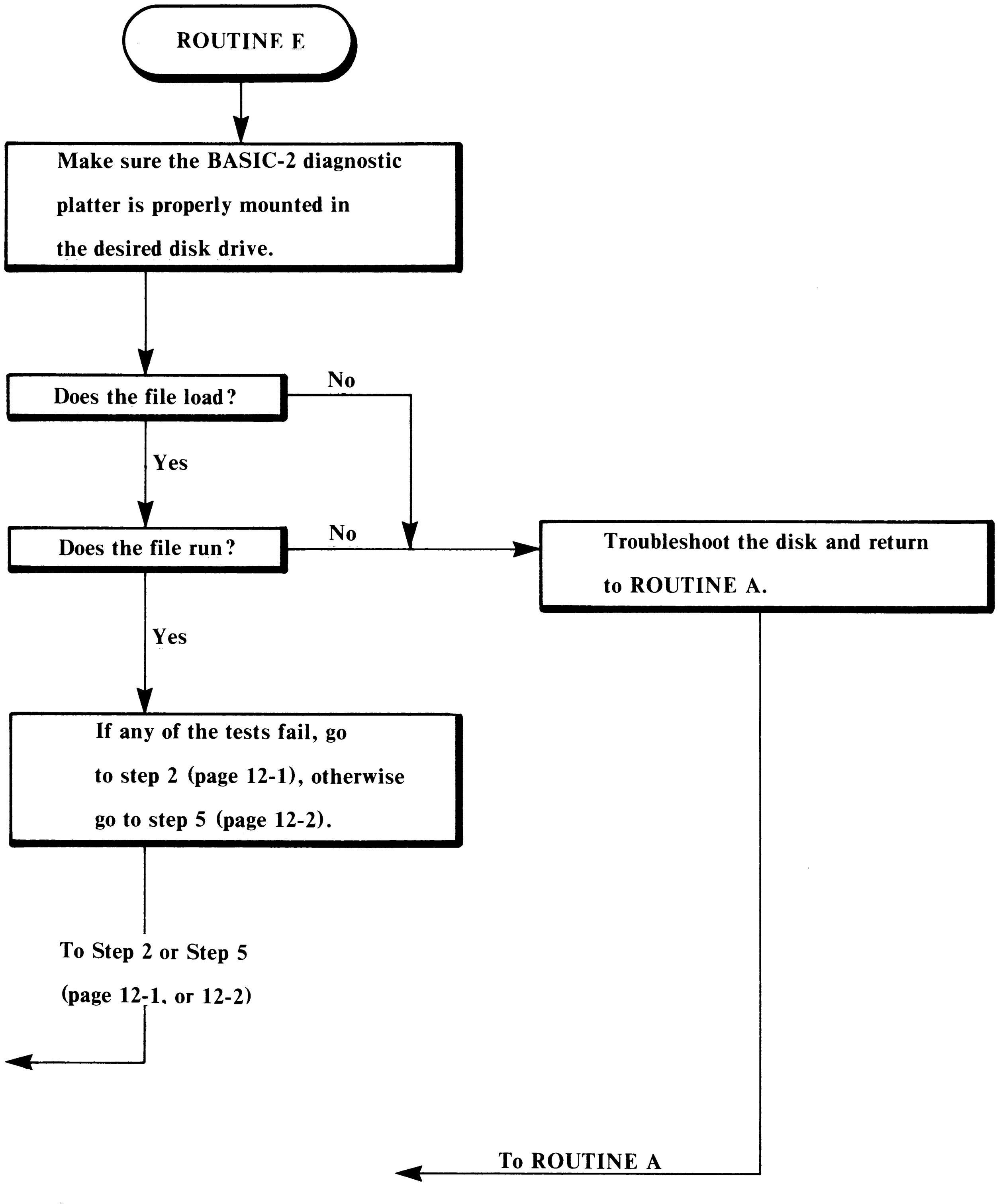
# 2200 LVP TROUBLESHOOTING FLOWCHART

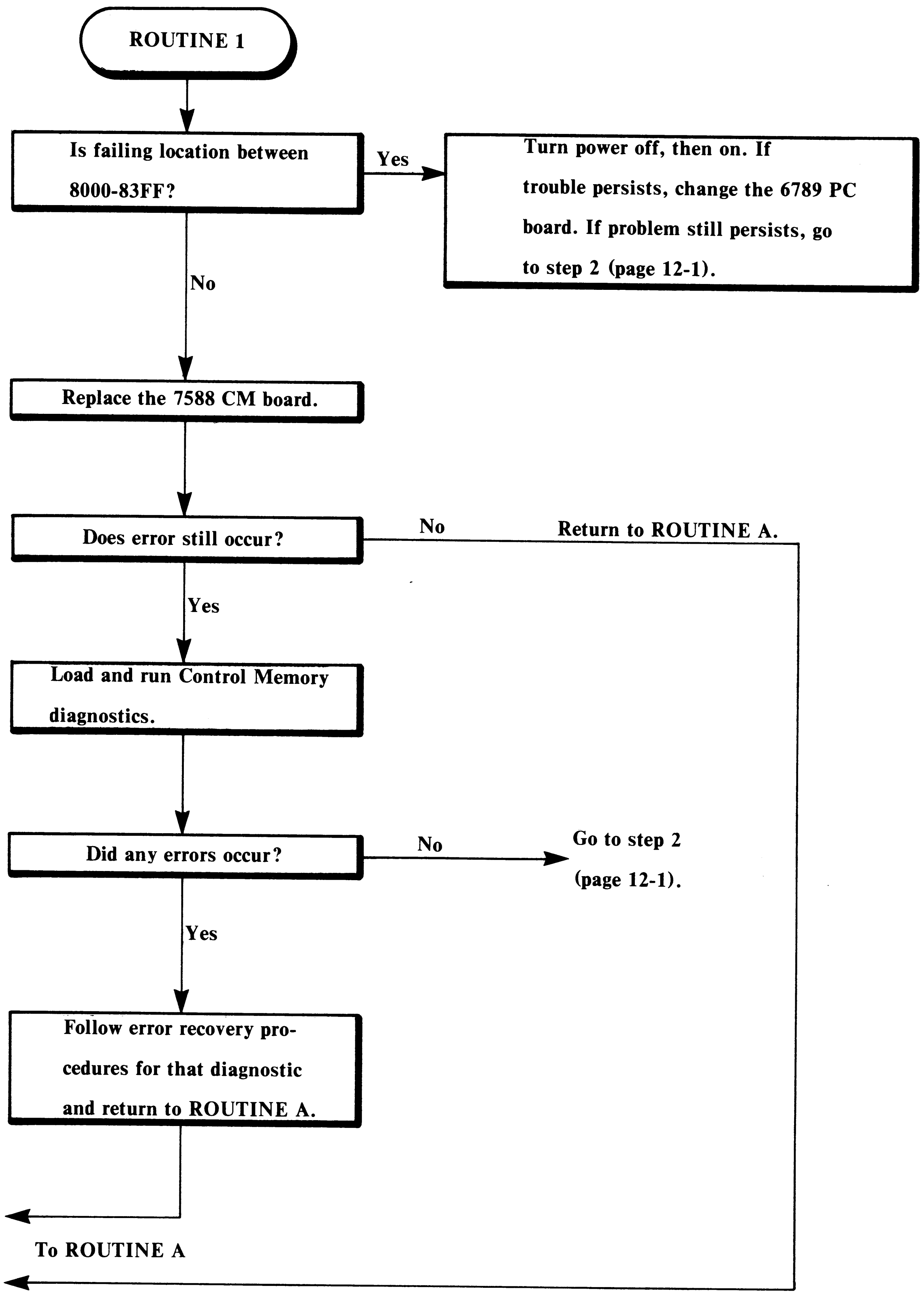

ROUTINE A
For any SYSTEM ERROR message, go directly to ROUTINE B.
Go to ROUTINE B.
Does the system power-on message (MOUNT SYSTEM PLATTER CR/LF PRESS RESET) appear?
No
Go to ROUTINE C.
Yes
Does each file load and run when selected by the SF' key in response to the KEY SF' message?
No
Go to ROUTINE D.
Yes
Does the BASIC-2 diagnostic load and run?
No
Go to ROUTINE E.
Yes
If problem does not fall into one of the above general conditions, then continue with step 2 (page 12-1).

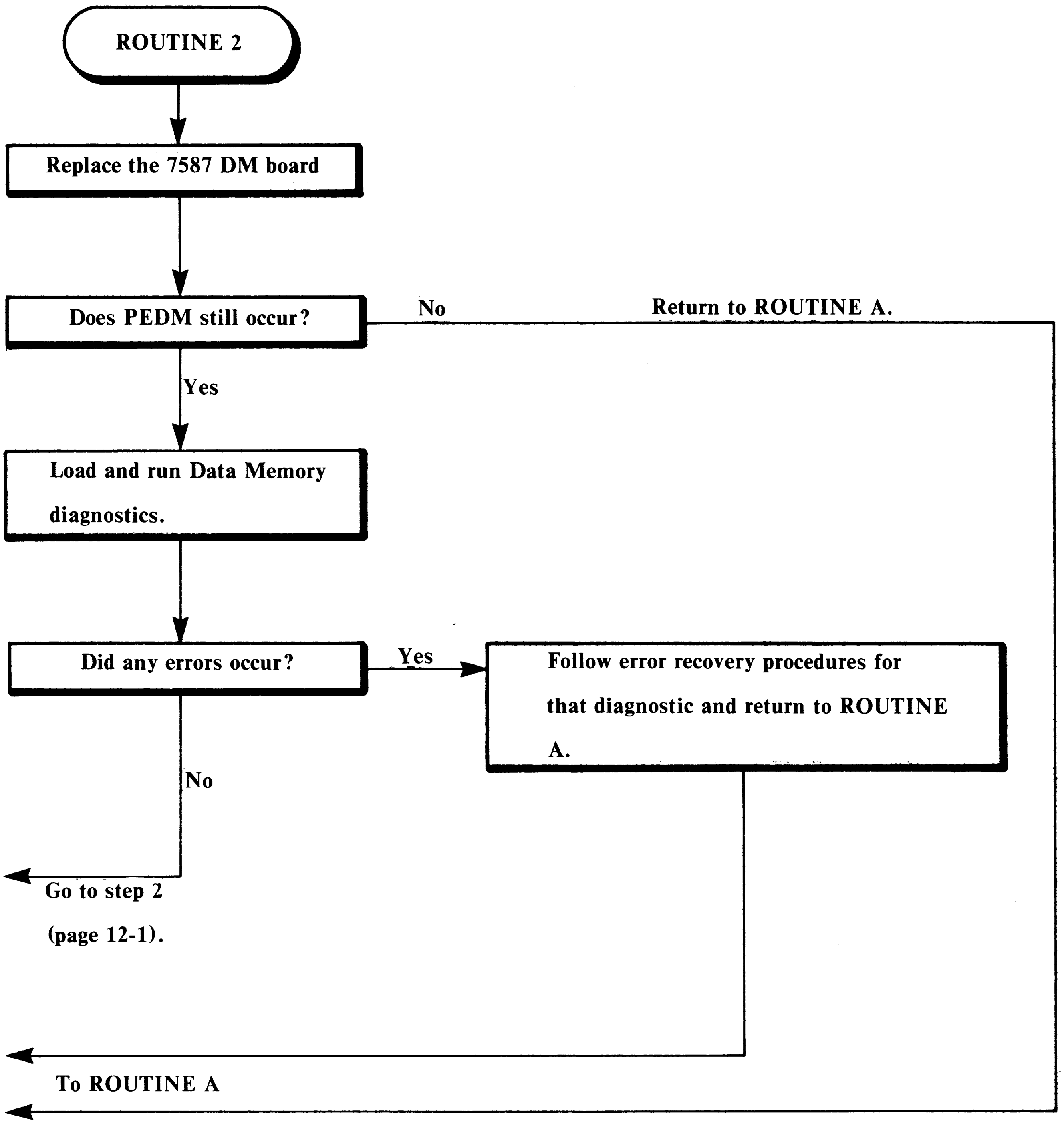

ROUTINE B
Is this a PECM error?
Yes
Note the failing location and go to ROUTINE 1.
No
Is this a PEDM error?
Yes
Note the failing location and go to ROUTINE 2.
No
Is this a VECM error?
Yes
Note the failing location and go to ROUTINE 3.
No
Is this a DISK error?
Yes
Note error number and follow the recovery procedures for that error (ref: Section 3.2.3.3). If error persists, go to step 2 (page 12-1).
No
There are no other SYSTEM ERRORS. Reappraise the problem and return to
To ROUTINE A

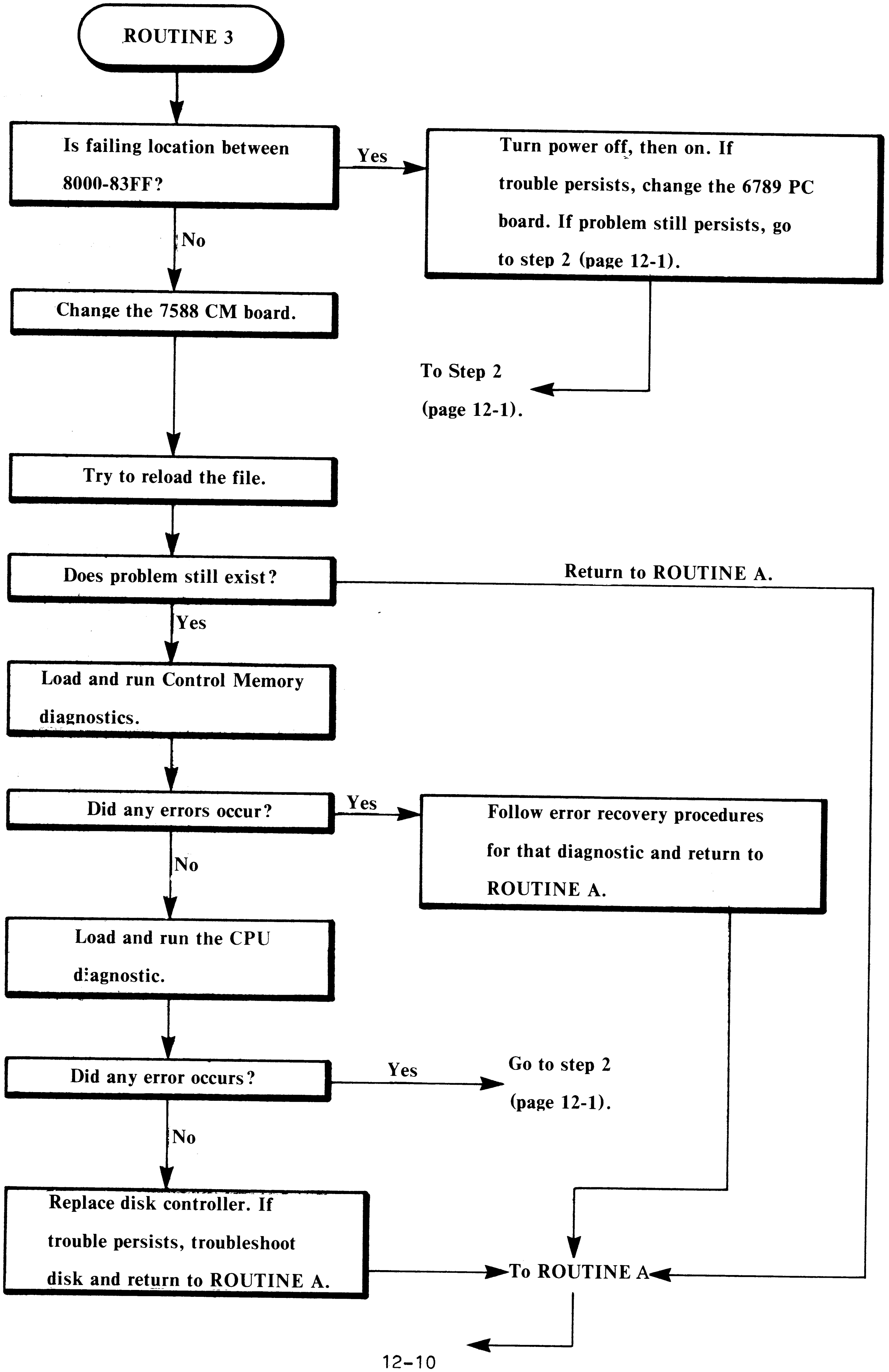

ROUTINE C
Does any portion of the MOUNT SYSTEM PLATTER CR/LF PRESS RESET message appear?
Yes
Go to ROUTINE 4.
No
Is display filled with miscellaneous characters?
Yes
Go to ROUTINE 5.
No
Troubleshoot display unit and return to ROUTINE A.
To ROUTINE A

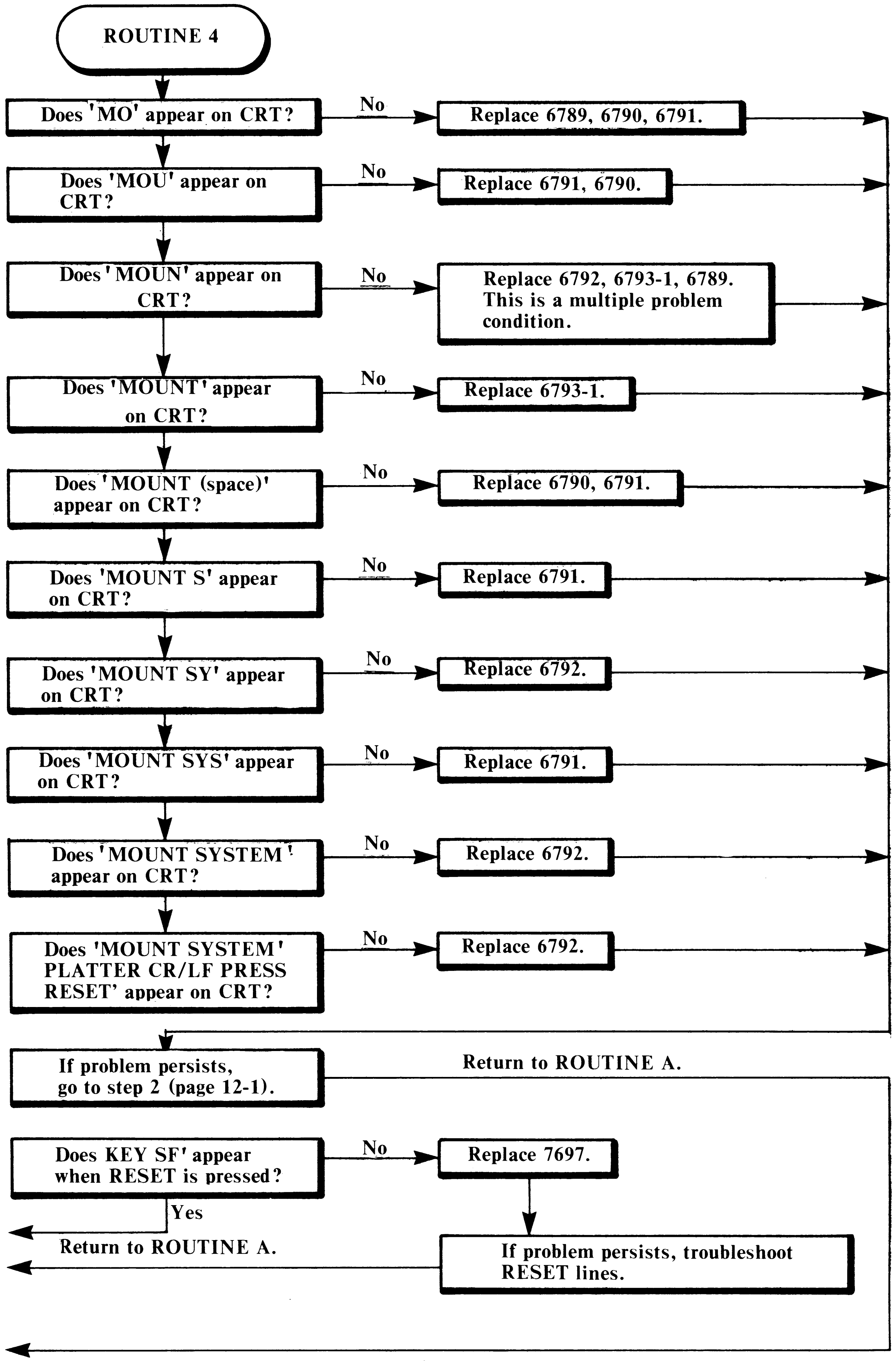

ROUTINE D
Make sure the address for the SF' key and disk drive are the same.
Does SF' key number appear on CRT?
No
Replace 7592 PC in 2236DE terminal. If trouble persits, change 6793-1. If trouble still persists, troubleshoot keyboard and go to step 2 (page 12-1).
Yes
To step 2 (page 12-1)
Does the 'KEY SF'? ' message reappear?
No
Does file load (comment displayed)?
Yes
Yes
No
Make sure correct SYSTEM PLATTER is properly mounted in disk and key the desired function key again. If trouble persists, replace disk controllers/troubleshoot disk.
Replace disk controllers/ troubleshoot disk (disk not being enabled) and return to ROUTINE A.
Does file run?
No
Go to step 2 (page 12-1)
Yes
Follow procedures for that file and return to ROUTINE A.
To ROUTINE A

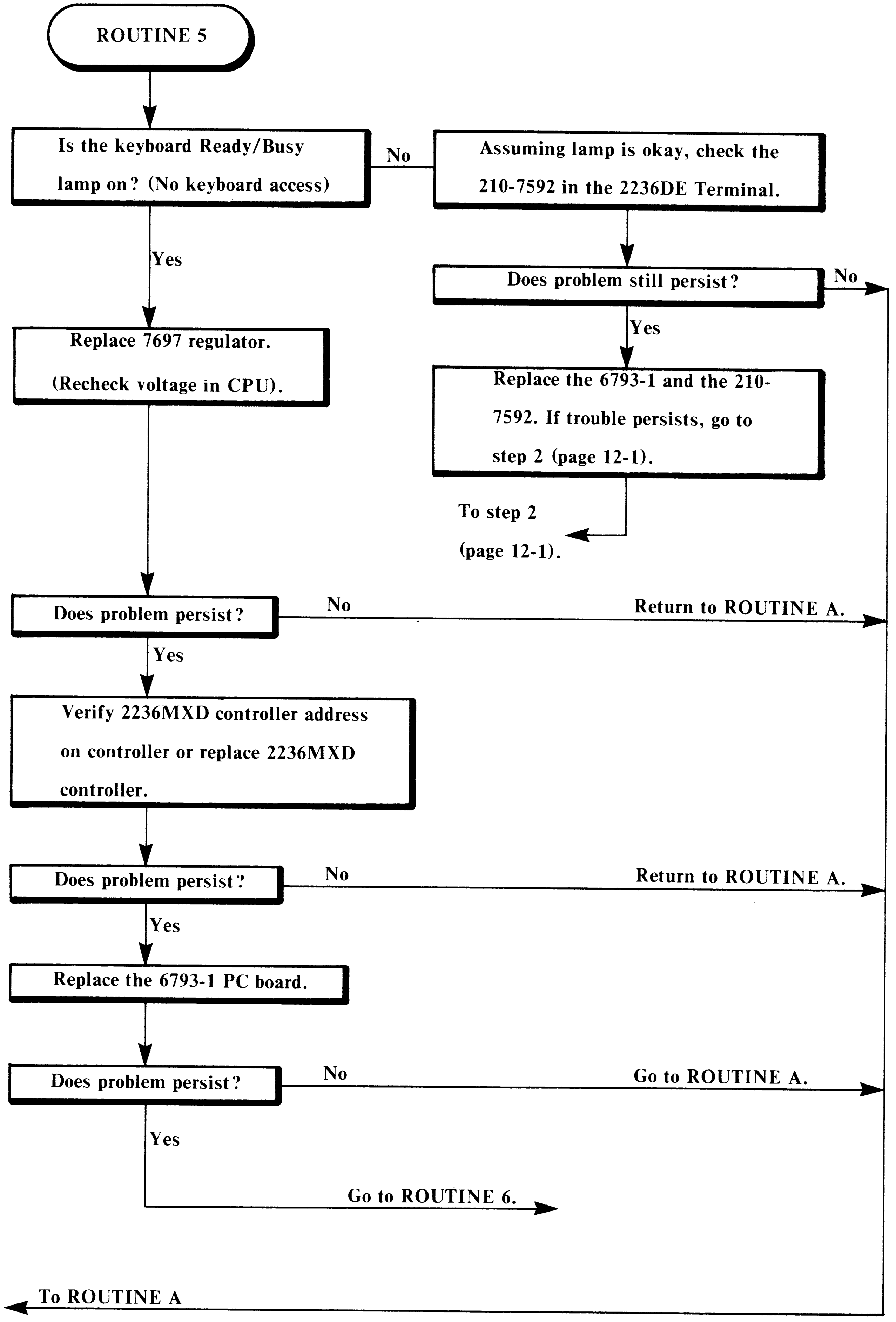

ROUTINE E
Make sure the BASIC-2 diagnostic platter is properly mounted in the desired disk drive.
Does the file load?
No
Yes
Does the file run?
No
Troubleshoot the disk and return to ROUTINE A.
Yes
If any of the tests fail, go to step 2 (page 12-1), otherwise go to step 5 (page 12-2).
To Step 2 or Step 5 (page 12-1, or 12-2)
To ROUTINE A

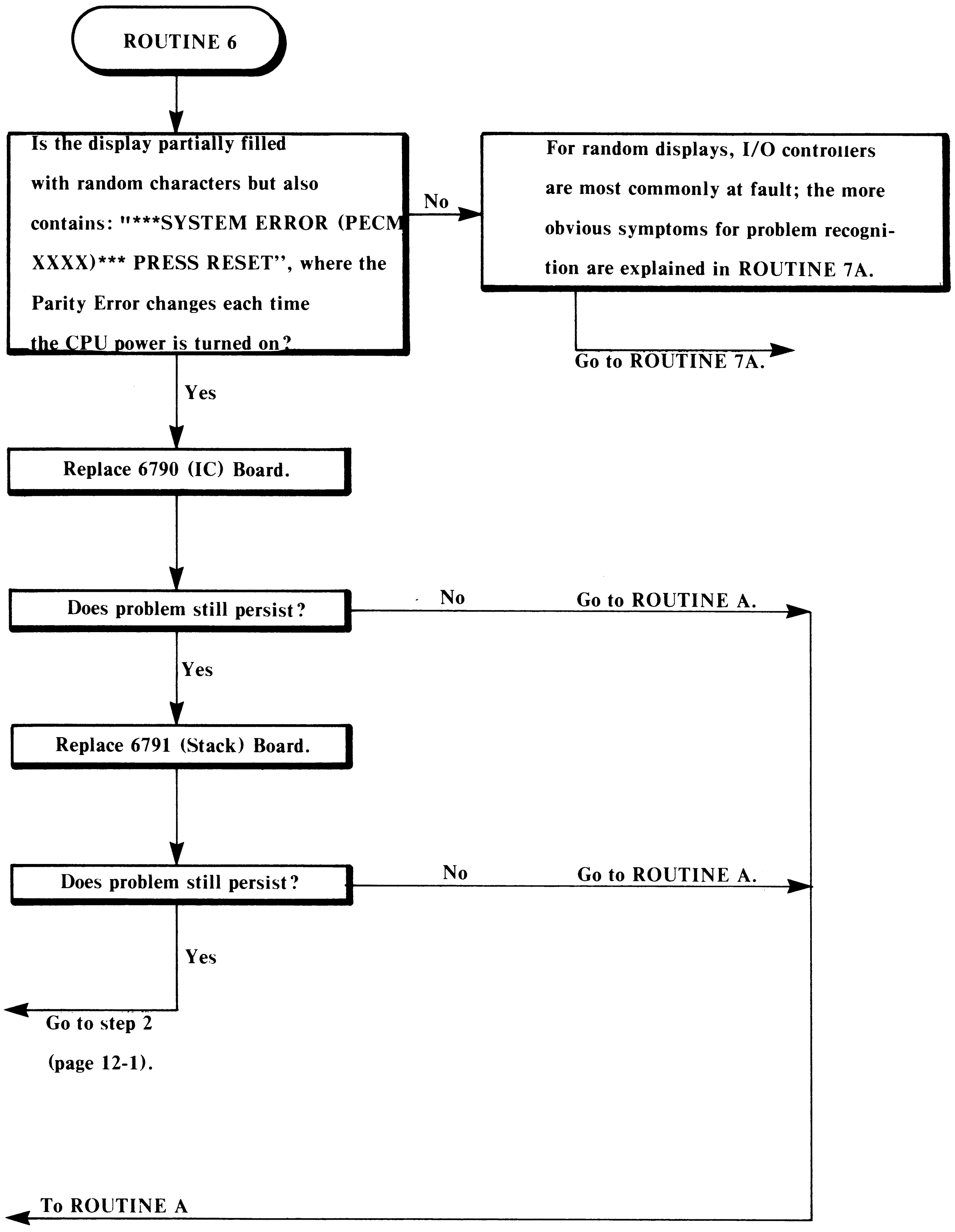

ROUTINE 1
Is failing location between 8000-83FF?
Yes
Turn power off, then on. If trouble persists, change the 6789 PC board. If problem still persists, go to step 2 (page 12-1).
No
Replace the 7588 CM board.
Does error still occur?
No
Return to ROUTINE A.
Yes
Load and run Control Memory diagnostics.
Did any errors occur?
No
Go to step 2 (page 12-1).
Yes
Follow error recovery procedures for that diagnostic and return to ROUTINE A.
To ROUTINE A

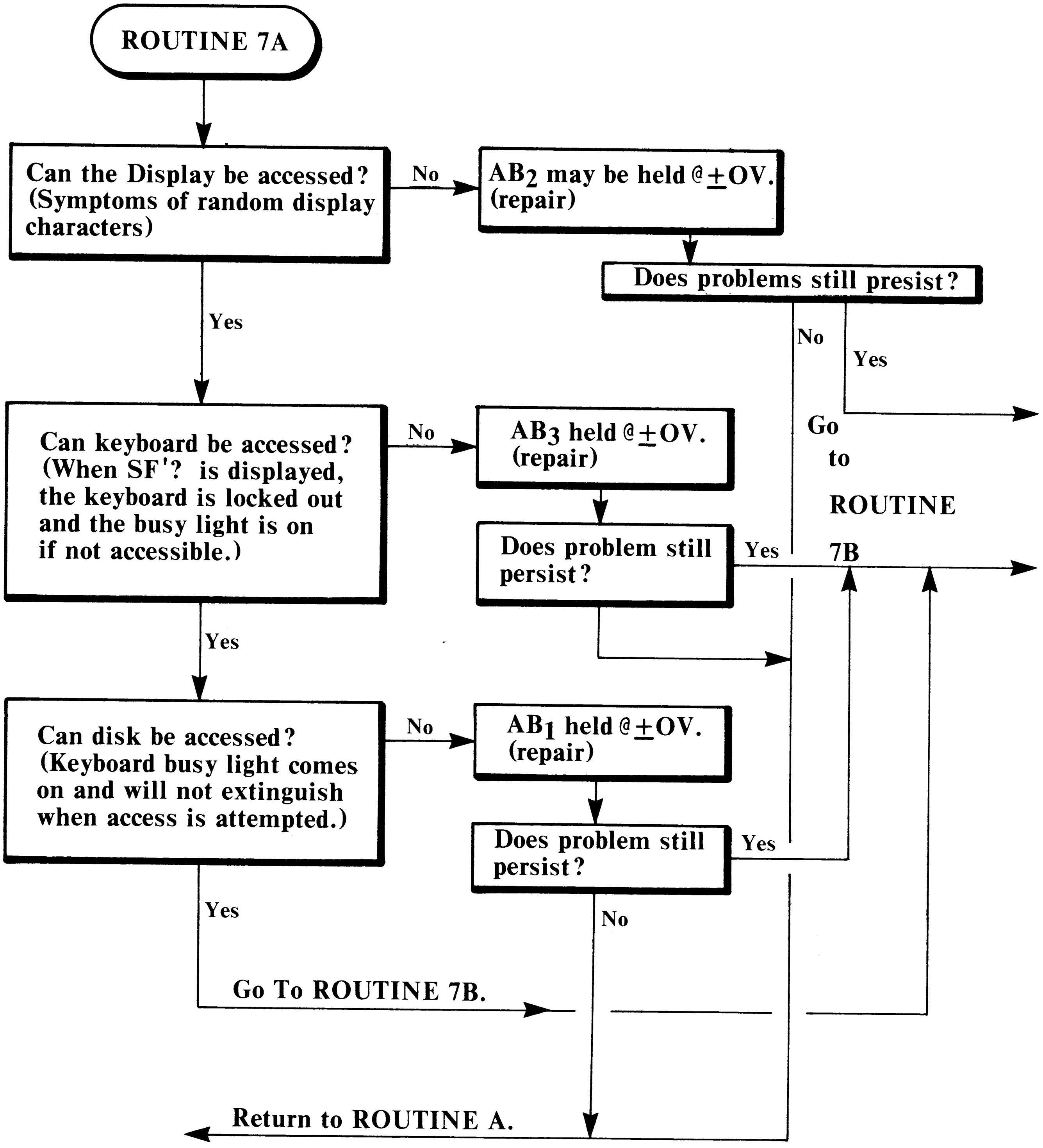

ROUTINE 2
Replace the 7587 DM board
Does PEDM still occur?
No
Return to ROUTINE A.
Yes
Load and run Data Memory diagnostics.
Did any errors occur?
Yes
Follow error recovery procedures for that diagnostic and return to ROUTINE A.
No
Go to step 2 (page 12-1).
To ROUTINE A

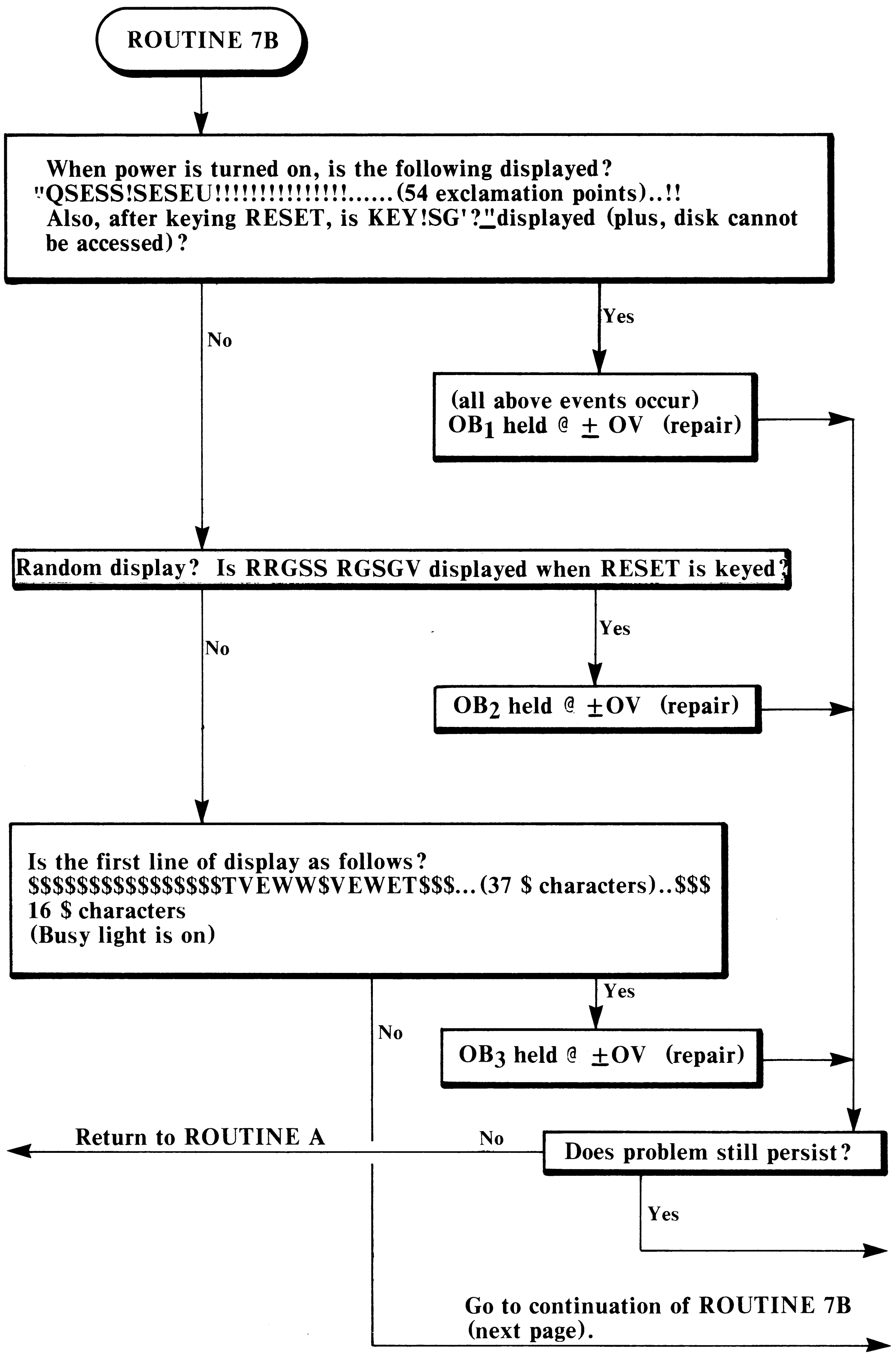

ROUTINE 3
Is failing location between 8000-83FF?
Yes
Turn power off, then on. If trouble persists, change the 6789 PC board. If problem still persists, go to step 2 (page 12-1).
No
Change the 7588 CM board.
To Step 2 (page 12-1).
Try to reload the file.
Does problem still exist?
Return to ROUTINE A.
Yes
Load and run Control Memory diagnostics.
Did any errors occur?
Yes
Follow error recovery procedures for that diagnostic and return to ROUTINE A.
No
Load and run the CPU diagnostic.
Did any error occurs?
Yes
Go to step 2 (page 12-1).
No
Replace disk controller. If trouble persists, troubleshoot disk and return to ROUTINE A.
To ROUTINE A

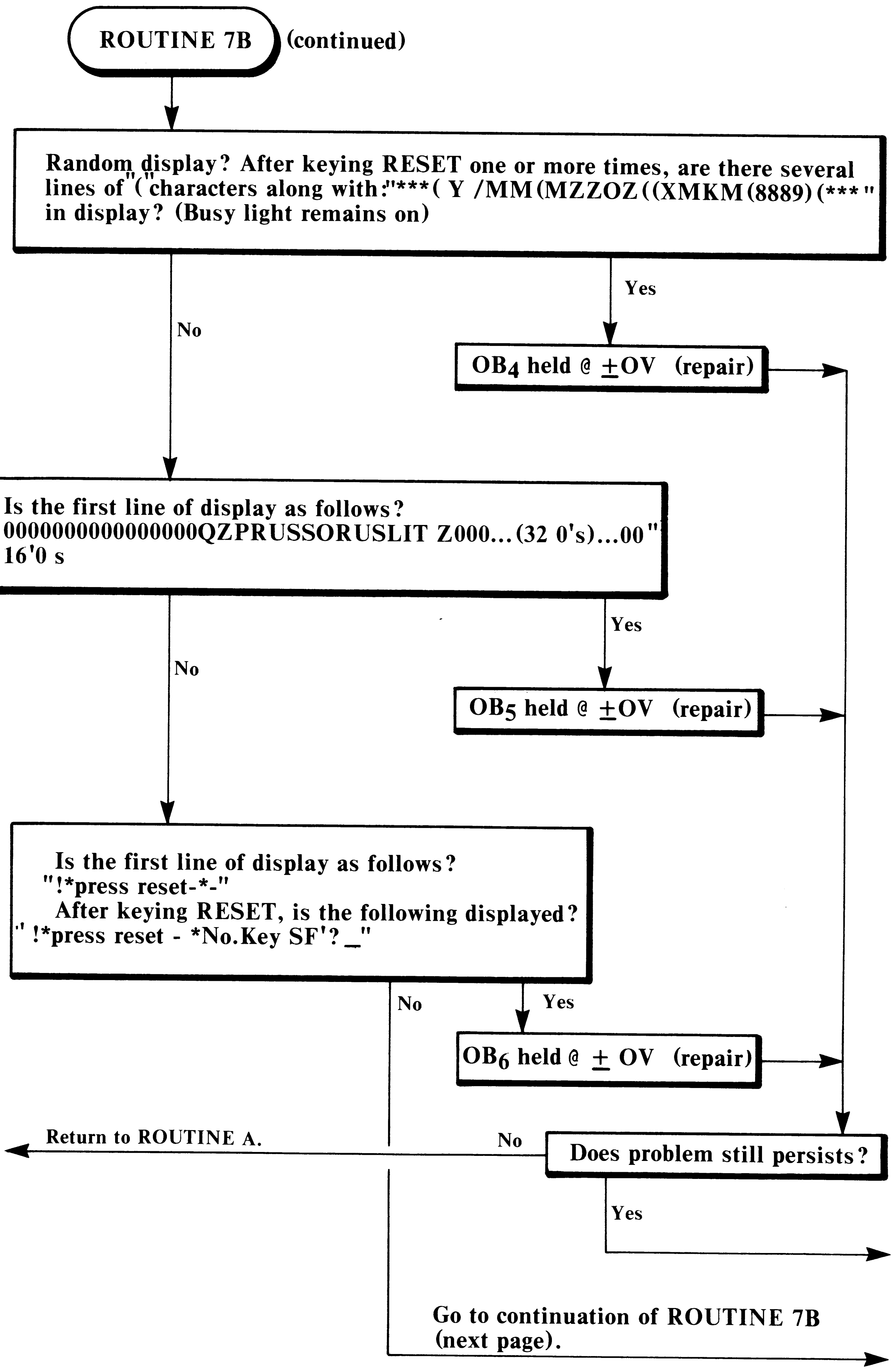

ROUTINE 4
Does 'MO' appear on CRT?
No
Replace 6789, 6790, 6791.
Does 'MOU' appear on CRT?
No
Replace 6791, 6790.
Does 'MOUN' appear on CRT?
No
Replace 6792, 6793-1, 6789. This is a multiple problem condition.
Does 'MOUNT' appear on CRT?
No
Replace 6793-1.
Does 'MOUNT (space)' appear on CRT?
No
Replace 6790, 6791.
Does 'MOUNT S' appear on CRT?
No
Replace 6791.
Does 'MOUNT SY' appear on CRT?
No
Replace 6792.
Does 'MOUNT SYS' appear on CRT?
No
Replace 6791.
Does 'MOUNT SYSTEM' appear on CRT?
No
Replace 6792.
Does 'MOUNT SYSTEM' PLATTER CR/LF PRESS RESET' appear on CRT?
No
Replace 6792.
If problem persists, go to step 2 (page 12-1).
Return to ROUTINE A.
Does KEY SF' appear when RESET is pressed?
No
Replace 7697.
Yes
Return to ROUTINE A.
If problem persists, troubleshoot RESET lines.

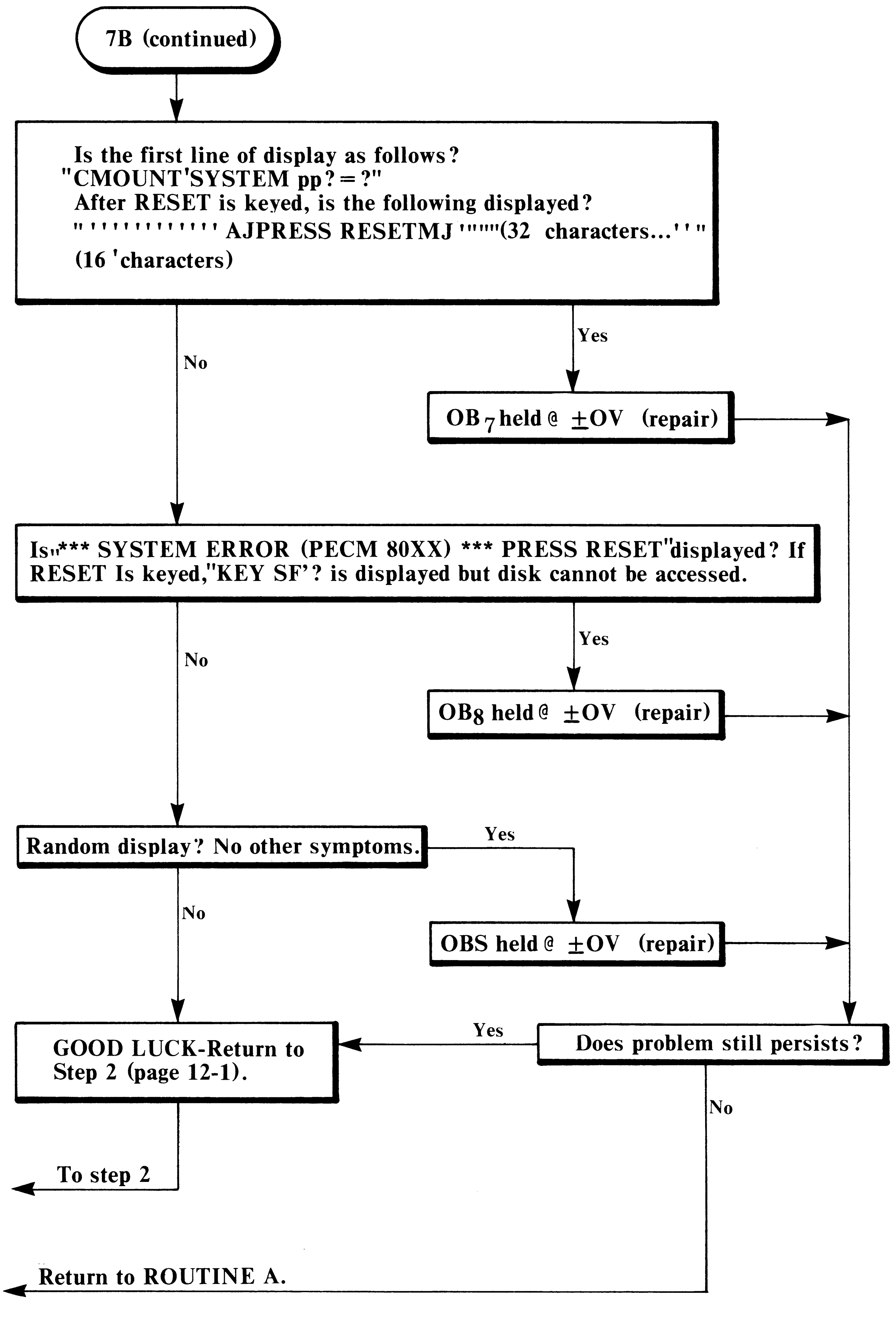

ROUTINE 5
Is the keyboard Ready/Busy lamp on? (No keyboard access)
No
Assuming lamp is okay, check the 210-7592 in the 2236DE Terminal.
Does problem still persist?
No
Yes
Replace the 6793-1 and the 210-7592. If trouble persists, go to step 2 (page 12-1).
To step 2 (page 12-1).
Yes
Replace 7697 regulator. (Recheck voltage in CPU).
Does problem persist?
No
Return to ROUTINE A.
Yes
Verify 2236MXD controller address on controller or replace 2236MXD controller.
Does problem persist?
No
Return to ROUTINE A.
Yes
Replace the 6793-1 PC board.
Does problem persist?
No
Go to ROUTINE A.
Yes
Go to ROUTINE 6.
To ROUTINE A


ROUTINE 6
Is the display partially filled with random characters but also contains: "***SYSTEM ERROR (PECM XXXX)*** PRESS RESET", where the Parity Error changes each time the CPU power is turned on?
No
For random displays, I/O controllers are most commonly at fault; the more obvious symptoms for problem recognition are explained in ROUTINE 7A.
Go to ROUTINE 7A.
Yes
Replace 6790 (IC) Board.
Does problem still persist?
No
Go to ROUTINE A.
Yes
Replace 6791 (Stack) Board.
Does problem still persist?
No
Go to ROUTINE A.
Yes
Go to step 2 (page 12-1).
To ROUTINE A


ROUTINE 7A
Can the Display be accessed? (Symptoms of random display characters)
No
AB2 may be held @±OV. (repair)
Does problems still presist?
No
Yes
Yes
Can keyboard be accessed? (When SF'? is displayed, the keyboard is locked out and the busy light is on if not accessible.)
No
AB3 held @±OV. (repair)
Go to ROUTINE 7B
Does problem still persist?
Yes
Yes
Can disk be accessed? (Keyboard busy light comes on and will not extinguish when access is attempted.)
No
AB1 held @±OV. (repair)
Does problem still persist?
Yes
No
Yes
Go To ROUTINE 7B.
Return to ROUTINE A.


ROUTINE 7B
When power is turned on, is the following displayed? "QSESS!SESEU!!!!!!!!!!!!!!!!......(54 exclamation points)..!! Also, after keying RESET, is KEY!SG'?_"displayed (plus, disk cannot be accessed)?
No
Yes
(all above events occur) $OB_1$ held @ ± OV (repair)
Random display? Is RRGSS RGSGV displayed when RESET is keyed?
No
Yes
$OB_2$ held @ ±OV (repair)
Is the first line of display as follows? $$$$$$$$$$$$$$$$TVEWW$VEWET$$$...(37 $ characters)..$$$ 16 $ characters (Busy light is on)
No
Yes
$OB_3$ held @ ±OV (repair)
Return to ROUTINE A
No
Does problem still persist?
Yes
Go to continuation of ROUTINE 7B (next page).


ROUTINE 7B
(continued)
Random display? After keying RESET one or more times, are there several lines of "(" characters along with:"***( Y /MM(MZZOZ((XMKM(8889)(***" in display? (Busy light remains on)
No
Yes
OB4 held @ ±OV (repair)
Is the first line of display as follows?
"0000000000000000QZPRUSSORUSLIT Z000...(32 0's)...00"
16'0 s
No
Yes
OB5 held @ ±OV (repair)
Is the first line of display as follows?
"!*press reset-*-"
After keying RESET, is the following displayed?
" !*press reset - *No.Key SF'?_"
No
Yes
OB6 held @ ± OV (repair)
Return to ROUTINE A.
No
Does problem still persists?
Yes
Go to continuation of ROUTINE 7B (next page).



7B (continued)
Is the first line of display as follows?
"CMOUNT'SYSTEM pp? = ?"
After RESET is keyed, is the following displayed?
" ' ' ' ' ' ' ' ' ' ' ' ' AJPRESS RESETMJ '"""(32 characters...' ' "
(16 'characters)
Yes
No
OB 7 held @ ±OV (repair)
Is"*** SYSTEM ERROR (PECM 80XX) *** PRESS RESET"displayed? If RESET Is keyed,"KEY SF'? is displayed but disk cannot be accessed.
Yes
No
OB8 held @ ±OV (repair)
Random display? No other symptoms.
Yes
No
OBS held @ ±OV (repair)
GOOD LUCK-Return to Step 2 (page 12-1).
Yes
Does problem still persists?
No
To step 2
Return to ROUTINE A.

## 12.2 SYSTEM ERRORS

This section details all system errors (with the exception of system disk errors) regardless of whether they normally would appear during system initialization, the RESET function, the loading of a program, or the running of memory diagnostics. To clarify, some system errors described in this section (AECM, BECM, AEDM, BEDM, and REDM) do not appear in the discussion of system errors in SECTION 3 because they normally will not occur during initialization, RESET, or program loading.

### 12.2.1 CONTROL MEMORY ERRORS

When the system detects a Control Memory failure, one of the following error messages is displayed:

AECM -- Addressing Error in Control Memory
BECM -- Bit Error in Control Memory
PECM -- Parity Error in Control Memory
VECM -- Verify Error in Control Memory

1) AECM aaaa bbbb xxxxxx

| | | |
|---|---|---|
| Where: | aaaa = | The address of the instruction in error |
| | bbbb = | The conflicting address |
| | xxxxxx = | An XOR of the expected and actually-read instruction |

This error indicates that writing to Control Memory location bbbb seems to modify location aaaa. The "1" bits in the xxxxxx field of the display indicate which bit(s) have been modified. The error could also occur if a chip at location aaaa had a marginal failure.

2) BECM aaaa xxxxxx

| | | |
|---|---|---|
| Where: | aaaa = | The address of the instruction in error |
| | xxxxxx = | An XOR of the instruction actually read from memory with the instruction that was expected to be there. |

This error implies that a bit error was detected while reading Control Memory. The "1" bits in the xxxxxx field of the display indicate which bit(s) are incorrect.

3) PECM aaaa dddddd

| | | |
|---|---|---|
| Where: | aaaa = | The address of the instruction with bad parity. |
| | dddddd = | The instruction located at aaaa. The instruction is reread when displayed and thus may not be the same as when the error occurred. |

This error implies that bad parity was detected during execution of the diagnostic. Bad parity may be the result of:

a) Bits dropped
b) Bits picked up
c) Bad parity written
d) Bad parity-check logic

4) VECM aaaa

| | | |
|---|---|---|
| Where: | aaaa = | An address in the section of Control Memory that does not verify correctly. |

Normally, this error will only be reported when the loading of a system program from disk into Control Memory is not successful.

## 12.2.2 DATA MEMORY ERRORS

When the system detects a Data Memory failure, one of the following error messages is displayed:

AEDM -- Addressing Error in Data Memory
BEDM -- Bit Error in Data Memory
PEDM -- Parity Error in Data Memory
REDM -- Read Error in Data Memory
VEDM -- Verify Error in Data Memory

1) AEDM ss.aaaa ss.bbbb xx

| | | |
|---|---|---|
| Where: | ss = | Memory bank containing the error (00 = bank #1; 40 = bank #2; 80 = bank #3; C0 = bank #4) |
| | aaaa = | Address of the data in error |
| | bbbb = | Conflicting address |
| | xx = | XOR of the expected and actually-read data. |

This error indicates that writing to location bbbb seems to modify location aaaa. The "1" bits in the xx field of the display indicate which bits have been modified. The error could also occur if a chip at location aaaa had a marginal failure.

2) BEDM ss.aaaa xxyy

| | | |
|---|---|---|
| Where: | ss = | Memory bank containing the error (00 = bank #1; 40 = bank #2; 80 = bank #3; C0 = bank #4) |
| | aaaa = | Address of the data in error |
| | xxyy = | XOR of the data actually read from User/Data memory with the data that was expected to be there. |
| | xx = | Corresponds to the byte at location aaaa |
| | yy = | Corresponds to the byte at location aaaa+1 |

This error implies that a memory error was detected while reading User/Data Memory. The "1" bits in the xxyy field of the display indicate which bit(s) are not correct. If all the bits are zero, one of the two parity bits associated with the pair of bytes read is incorrect.

3) PEDM ss.aaaa

| | | |
|---|---|---|
| Where: | ss = | Memory bank containing the error (00 = bank #1; 40 = bank #2; 80 = bank #3; C0 = bank #4) |
| | aaaa = | Data memory address (i.e., the current value of the PC's) at the time of the error. This is probably, but not necessarily, the address of the memory location with bad parity. |

This error implies that bad parity was detected during a read of 8-bit User/Data Memory. Bad parity may be the result of:

a) Bits dropped
b) Bits picked up
c) Bad parity written
d) Bad parity-check logic

NOTE:

In order to determine which bit is bad, a technician may ground L41 pin 3 on the 6789 board; this action disables parity-error logic. If this is performed, a different error message will be displayed.

4) REDM ss.aaaa xx

| | | |
|---|---|---|
| Where: | ss = | Memory bank containing the error (00 = bank #1; 40 = bank #2; 80 = bank #3; C0 = bank #4) |
| | aaaa = | Address of the data in error |
| | xx = | XOR of the data in memory with the data that was expected to be there. |

This error implies that a memory error was detected while reading User/Data Memory. The "1" bits on the xx field of the display indicate which bits are not correct. If all the bits are zero, a bit in the other byte of the pair of bytes is incorrect.

5) VEDM ss.aaaa

| | | |
|---|---|---|
| Where: | ss = | Memory bank containing the error (00 = bank #1; 40 = bank #2; 80 = bank #3; C0 = bank #4) |
| | aaaa = | Address of the data in error |

Normally, this error will only be reported when the loading of system program constants from disk into Data Memory is not successful.

## 12.3 MEMORY DIAGNOSTIC ERROR INTERPRETATION

This section contains charts and explanations of how to decode the memory diagnostic error messages (ref: Section 12.2) to point out exactly which RAM (WL #377-0345) has failed. There is a chart and an error example for both Control and Data Memory.

### 12.3.1 CONTROL MEMORY

EXAMPLE: error message--BECM 02D7 0C0000

The error was a Bit Error in Control Memory.
The address of the failing location was 02D7 (HEX).
The XOR of the expected and actually-read data was 0C0000 (HEX).

FIGURE 12-1 shows that the failing location is contained in the top two rows of RAM's (error was between 0000 and 3FFFF (HEX)).

By inserting the XOR result (0C0000) into the blocks depicting the ROW SECTIONS, it can be seen that row section E (located in the second row from the top of the board) contains the faulty RAM's.

The bits that failed were 4 and 8 (from XOR--HEX(C) is equal to bits 4 and 8 ON); therefore, the bad RAM's are in the second row from the top of the board, and are the fifth and sixth RAM's from the right side of the board.

If the failing address was between 4000 and 7FFF (HEX), the procedure for determining which RAM was faulty is the same as described above, except the RAM would be located in the bottom two rows.

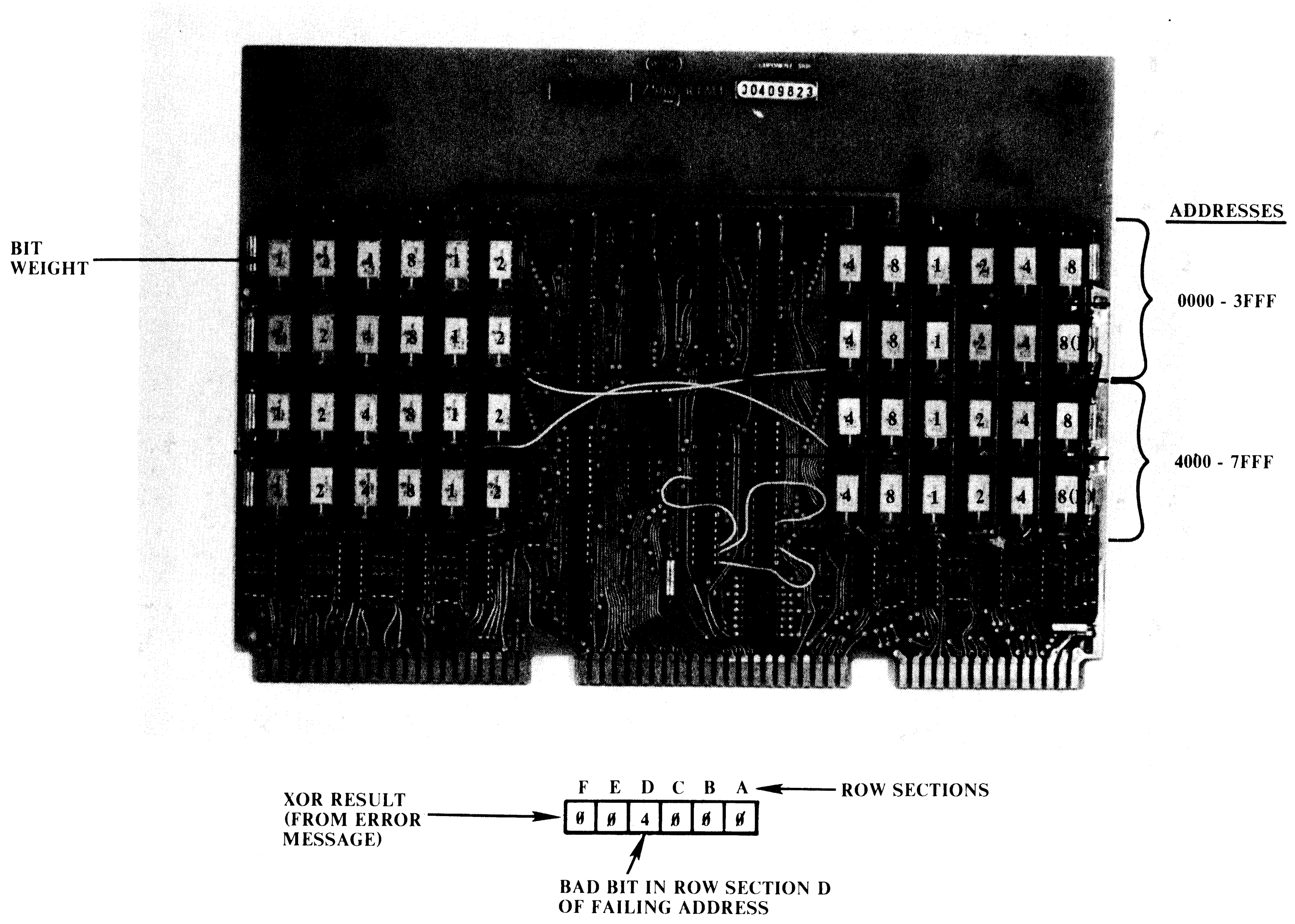


FIGURE 12-1 CONTROL MEMORY DIAGNOSTIC ERROR INTERPRETATION

## 12.3.2 DATA MEMORY

EXAMPLE: error message--BEDM 40.A3C4 8204

The error was a Bit Error in Data Memory.
The failing address is located in bank #2 (40.).
The address of the failing location was A3C4 (HEX).
The XOR of the expected and actually-read data was 8204 (HEX).

FIGURE 12-2 shows that the failing address is in the top two rows of RAM's (error was in bank #2).

The figure also shows that the failing location is contained in the top row of RAM's (error was between 8000 and FFFF (HEX)).

The failing address was even (A3C4); therefore, the faulty RAM('s) is somewhere on the left side of the top row of RAM's.

The bits that failed were 2 and 80 (from first two digits of XOR--HEX(82) is equal to bits 2 and 80 ON); therefore, the bad RAM's are in the top row, and are the second and eighth RAM's from the left side of the board.

In this example there is one additional RAM that failed.

The last two digits of the XOR (04) indicate that bit 4 in location A3C5 (address + 1) also failed.

The corresponding RAM is in the top row, seventh from the right side (odd address) of the board.

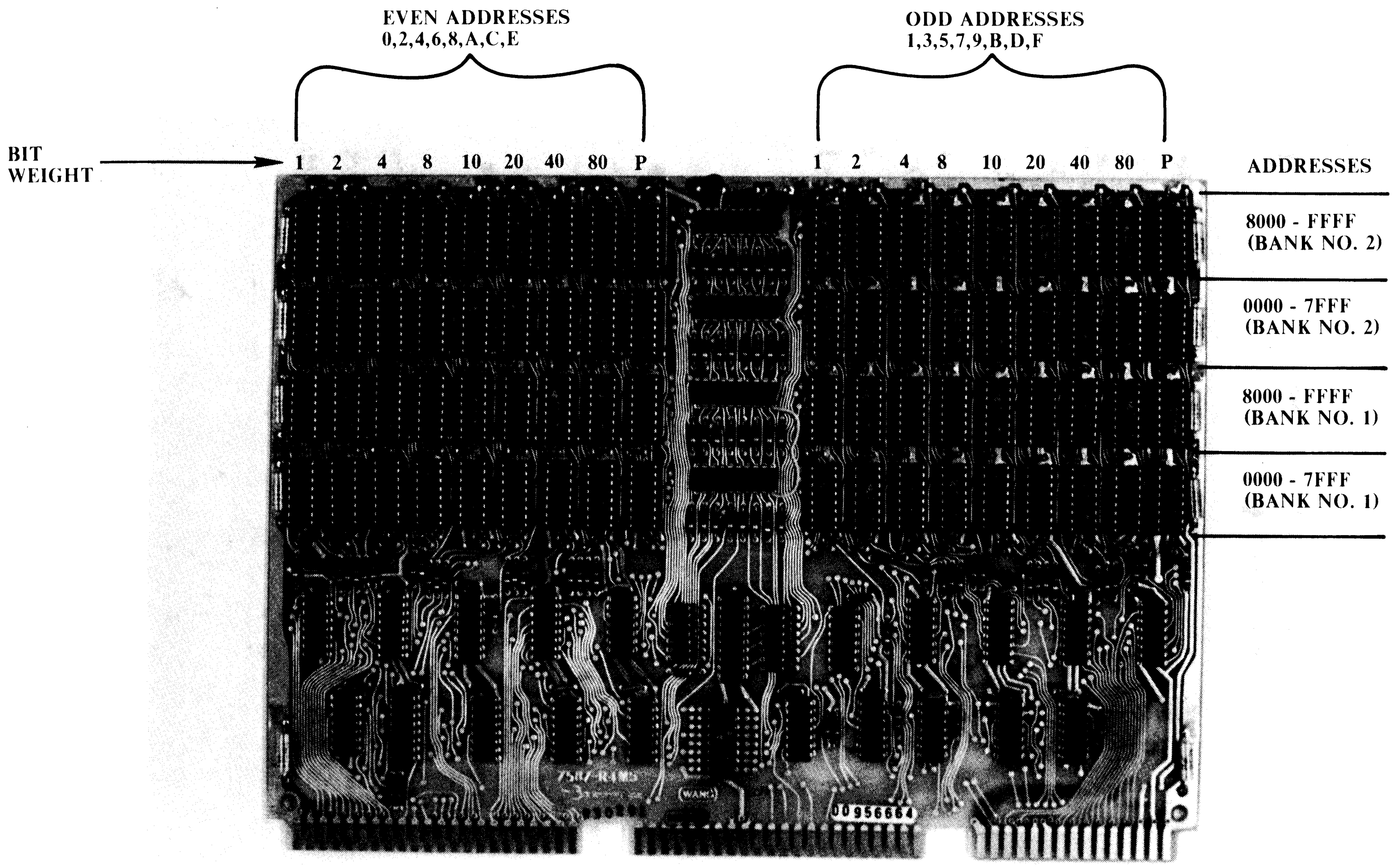


FIGURE 12-2 DATA MEMORY DIAGNOSTIC ERROR INTERPRETATION

## NOTES

## NOTES

## SECTION 13
## CONVERSIONS

Refer to documentation category I.B.2 for information concerning data-memory, and fixed-disk-drive capacity upgrades.

## NOTES

## SECTION 14

## PARTS LIST

| DESCRIPTION | WL # |
|---|---|
| Memory Control Board (CPU) | 210-6789-A |
| Instruction Counter Board (CPU) | 210-6790 |
| Stack Board (CPU) | 210-6791 |
| ALU Board (CPU) | 210-6792 |
| Register Board (CPU) | 210-6793-1 |
| Data Memory Board--32K (CPU) | 210-7587-1B |
| Data Memory Board--64K (CPU) | 210-7587-1A |
| Data Memory Board--128K (CPU) | 210-7587-3A |
| Control Memory Board--32K (CPU) | 210-7588-1A |
| 2200/Disk Interface Board (DPU) | 210-7694 |
| Disk Controller Board (DPU) | 210-7695-A |
| Microcomputer/Memory Board (DPU) | 210-7696-A |
| Power Supply Regulator Board | 210-7697 |
| CPU/DPU Motherboard | 210-7698 |
| Fixed-Disk Drive--2/4MB (60 Hz) | 278-4013 |
| Fixed-Disk Drive--2/4MB (50 Hz) | 278-4013-1 |
| Fixed-Disk Drive--8MB (60 Hz) | 278-4014 |
| Fixed-Disk Drive--8MB (50 Hz) | 278-4014-1 |
| DSDD Diskette Drive (60 Hz) | 278-4015 |
| DSDD Diskette Drive (50 Hz) | 278-4015-1 |
| DSDD Diskette (10-pack) | 177-0070-1 |
| Fuse, 3.0A (230 VAC) | 360-1031SB |
| Fuse, 5.0A (115 VAC) | 360-1051SB |
| LED, Power-On | 370-0031 |
| Bootstrap PROM #1 | 378-2045R2 |
| Bootstrap PROM #2 | 378-2046R2 |
| Bootstrap PROM #3 | 378-2047R2 |
| DPU PROM #1 (Power-On Diagnostic) | 378- |
| DPU PROM #2 | 378-4223 |
| DPU PROM #3 | 378-4222 |
| DPU PROM #4 (2 or 8 MB disk drive) | 378-4220 |
| DPU PROM #4 (4 or 8 MB disk drive) | 378-4221 |
| PLA PROM #1 (Fixed-disk drive) | 378-4224 |
| PLA PROM #2 (Double-density diskette) | 378-4225 |
| PLA PROM #3 (Single-density diskette) | 378-2560 |
| RAM, 16K x 1-Bit Dynamic | 377-0345 |
| Power Supply Chassis Assembly (60 Hz) | 270-0616 |
| Power Supply Chassis Assembly (50 Hz) | 270-0616-1 |
| Power Supply Heatsink Assembly | 270-0157 |
| Transformer (60 Hz) | 270-3153 |
| Transformer (60 Hz) | 270-3166 |
| Transformer (50 Hz) | 270-3165 |
| Transformer (50 Hz) | 270-3167 |
| LVP Cabinet Assembly | 279-4122 |
| CPU/DPU Card Case (3 I/O slot) | 279-0371 |
| Fan, Rotron WR2H1, 75CFM | 400-1013 |
| Cable, 20-Pin Ribbon (Disk I/O) | 220-3118 |
| Cable, 50-Pin Ribbon (Disk I/O) | 220-3119 |
| Cable, AC Power (Disk) | 220-0251 |
| Cable, Fan (From CPU/DPU Card Case) | 220-1424 |
| Cable, Fan (From Power Supply) | 220-1425 |

## NOTES

## SECTION 15

## BILL OF MATERIALS

A preliminary Bill of Materials starts on the following page.

```
POSITION  IN  LEGEND    COMPONENT           DESCRIPTION                              E C N   QUANTITY    U/M
 STRUCTURE     1  2 3   PART  NUMBER                                                         PER ASSY

2                IN   220-3118-   - -   20 COND CARD EDGE ASSY  C06482-0612                  1.0000   EACH
 3               IN   000-0004-   - -   LABOR SUB-SYSTEMS                                     .1130   EACH
 3               IN   000-0011-   - -   LABOR   QUALITY CONTROL                               .0230
 3               IN   350-0432-   - -   CONN 10-10 POS CARD EDGE W/O MT EAR                  2.0000   EACH
 3               FS   420-0076-   - -   CABLE,FLAT 20 COND 3M #3365/20                       1.6600   FEET

2                IN   220-3119-   - -   50 COND CARD EDGE ASSY  C06482-0613                  2.0000   EACH
 3               IN   000-0004-   - -   LABOR SUB-SYSTEMS                                     .0630   EACH
 3               IN   000-0011-   - -   LABOR   QUALITY CONTROL                               .0130
 3               IN   350-0433-   - -   CONN 25-25 POS CARD EDGE W/O MT EAR                  2.0000   EACH
 3               FS   420-0056-   - -   50 COND FLAT CABLE 3M 3365/50                        1.6600   FEET

2                IN   278-4013-   - -   TESTED 4MB FIXED DISK WINCH 60HZ                     1.0000   EACH
 3               IN   451-7040-XC- -    OBS EC15885D USE 451-7040-XD                         1.0000   EACH
 3               IN   451-7054-XB- -    BRKT LOCK SHIPPING       B10004-5050                 1.0000   EACH
 3               FS   650-3140-   - -   SCR 6-32 X .41 KNURLED THUMB                         1.0000   EACH
 3               FS   650-3200-   - -   SCR   6-32   5/8  PHIL PH MS SS                      1.0000   EACH
 3               FS   652-0104-   - -   NUT PUSH TINNERMAN C12043-012-3B                     3.0000   EACH
 3               FS   653-0039-   - -   WASHER SPRING BELLEVILLE  B0500-018                  1.0000   EACH
 3               FS   653-0052-XA- -    WASHER SHOULDER          B10004-5027                 1.0000   EACH
 3               FS   653-4000-   - -   WASH    8 .174ID .375OD .016 FL  SS                  3.0000   EACH
 3               IN   725-0086-   - -   DISK 8 INCH RIGID 5 MEG 60 HZ                        1.0000   EACH

2                IN   278-4015-   - -   TESTED 1MB DSDD FLOPPY DISK 60HZ                     1.0000   EACH
 3               IN   452-0190-XE- -    OBS EC15885D USE 452-0190-XF                         1.0000   EACH
 3               FS   650-3200-   - -   SCR   6-32   5/8  PHIL PH MS SS                      1.0000   EACH
 3               FS   650-4123-   - -   8-32X3/8 TRUSS HD PHL MS SS                          3.0000   EACH
 3               FS   653-0039-   - -   WASHER SPRING BELLEVILLE  B0500-018                  1.0000   EACH
 3               FS   653-0052-XA- -    WASHER SHOULDER          B10004-5027                 1.0000   EACH
 3               FS   653-4005-   - -   WASH    8 .168ID .296OD SPLIT    SS                  3.0000   EACH
 3               IN   725-0088-   - -   DISK 8 INCH FLOPPY DSDD DRIVE 60HZ                   1.0000   EACH
```

```
2      IN   279-4122-   - -   2200 LVP CABINET ASSY                                1.0000  EACH
 3     IN   210-7693-   - -   PCA 2200LVP INDICATOR BOARD                          1.0000  EACH
  4    IN   220-1444-   - -   FRONT PANEL LIGHT CABLE B06482-0630 E15750           1.0000  EACH
   5   IN   000-0004-   - -   LABOR SUB-SYSTEMS                                     .0830  EACH
   5   IN   000-0011-   - -   LABOR   QUALITY CONTROL                               .0160
   5   IN   601-0000-   - -   WIRE 18 GA BLK TEFLON STRANDED                       1.0000  FEET
   5   IN   601-0002-   - -   WIRE 18 GA RED TEFLON STRANDED                       1.0000  FEET
   5   FS   605-1004-   - -   CABLE TYE, PAN-TY PLTIM-M                            1.0000  EACH
   5   IN   605-1011-   - -   TY-WRAP IDENT MARKER                                 1.0000  EACH
   5   FS   654-1164-R  - -   PIN TERM 20-14 GA(REEL)AMP 61118-4                   2.0000  EACH
   5   IN   654-1184-   - -   CONN 6 POS HOUSING AMP 1-470271-0   E15607           1.0000  EACH

  4    FS   330-2047-4B- -    RES 470 OHM 1/4W 10% FIXED COMP                      1.0000  EACH
   5   FS * 330-2047-   - -   RES 470 OHM 1/4W 10% FIXED COMP                      1.0000  EACH

  4    IN   370-0031-   - -   LAMP RED LED RECTANGULAR CM4-264                     1.0000  EACH
  4    IN   510-7693-   - -   PCB 2200LVP INDICATOR BOARD                          1.0000  EACH

 3     IN   220-1444-   - -   FRONT PANEL LIGHT CABLE B06482-0630                  1.0000  EACH
  4    IN   000-0004-   - -   LABOR SUB-SYSTEMS                                     .0830  EACH
  4    IN   000-0011-   - -   LABOR   QUALITY CONTROL                               .0160
  4    IN   601-0000-   - -   WIRE 18 GA BLK TEFLON STRANDED                       1.0000  FEET
  4    IN   601-0002-   - -   WIRE 18 GA RED TEFLON STRANDED                       1.0000  FEET
  4    FS   605-1004-   - -   CABLE TYE, PAN-TY PLTIM-M                            1.0000  EACH
  4    IN   605-1011-   - -   TY-WRAP IDENT MARKER                                 1.0000  EACH
  4    FS   654-1164-R  - -   PIN TERM 20-14 GA(REEL)AMP 61118-4                   2.0000  EACH
  4    IN   654-1184-   - -   CONN 6 POS HOUSING AMP 1-470271-0   E15607           1.0000  EACH

 3     IN   279-0371-   - -   2200 LVP CARD CAGE SHORT 3 I/O                       1.0000  EACH
  4    IN   210-7698-   - -   PCA 2200LVP MOTHERBOARD                              1.0000  EACH
   5   IN   000-0004-   - -   LABOR SUB-SYSTEMS                                    1.5000  EACH
   5   IN   000-0011-   - -   LABOR   QUALITY CONTROL                               .3000
   5   IN   220-1423-   - -   FRONT PANEL LIGHT ASSY  B06482-0610 E15033           1.0000  EACH
    6  IN   000-0004-   - -   LABOR SUB-SYSTEMS                                     .0990  EACH
    6  IN   000-0011-   - -   LABOR   QUALITY CONTROL                               .0200
    6  IN   601-0000-   - -   WIRE 18 GA BLK TEFLON STRANDED                       3.0000  FEET
 6          IN   601-0001-   - -   WIRE 18 GA BRN TEFLON STRANDED       E15607      3.0000   FEET
 6          IN   601-0002-   - -   WIRE 18 GA RED TEFLON STRANDED                   3.0000   FEET
 6          IN   601-0006-   - -   WIRE 18 GA BLU TEFLON STRANDED       E15607      3.0000   FEET
 6          IN   601-0009-   - -   WIRE 18 GA WHT TEFLON STRANDED       E15607      3.0000   FEET
 6          FS   605-1004-   - -   CABLE TYE, PAN-TY PLTIM-M                        5.0000   EACH
 6          FS   654-1163-R  - -   SOCKET 20-14 GA.(REEL) AMP 61117-4               2.0000   EACH
 6          IN   654-1185-   - -   6 POS SOC HOUSING AMP 1-480270-0     E15607      1.0000   EACH

5           IN   220-1427-   - -   MOTHER BD CABLE ASSY    B06482-0617 E15314      1.0000   EACH
 6          IN   000-0004-   - -   LABOR SUB-SYSTEMS                                 .3220   EACH
 6          IN   000-0011-   - -   LABOR   QUALITY CONTROL                           .0640
 6        P FS   600-7000-   - -   16 GA BLACK STRANDED WIRE                        1.1600   FEET
  7         FS   600-7009-   - -   16 GA WHITE STRANDED WIRE                        1.0000   FEET

 6        P FS   600-7002-   - -   16 GA RED STRANDED WIRE                          1.1600   FEET
  7         FS   600-7009-   - -   16 GA WHITE STRANDED WIRE                        1.0000   FEET

 6        P FS   600-7003-   - -   16 GA ORANGE STRANDED WIRE                       1.1600   FEET
  7         FS   600-7009-   - -   16 GA WHITE STRANDED WIRE                        1.0000   FEET

 6        P FS   600-7006-   - -   16 GA BLUE STRANDED WIRE                         1.1600   FEET
  7         FS   600-7009-   - -   16 GA WHITE STRANDED WIRE                        1.0000   FEET

 6          FS   600-7009-   - -   16 GA WHITE STRANDED WIRE                        1.1600   FEET
 6          FS   600-7092-   - -   WIRE,16 AWG WHITE RED                            1.1600   FEET
 6        P FS   600-7095-   - -   16 GA WHITE/GREEN STRANDED WIRE                  1.1600   FEET
  7         FS   600-7009-   - -   16 GA WHITE STRANDED WIRE                        1.0000   FEET

 6          FS   600-7096-   - -   WIRE 16 ANG WHITE BLU                            1.1600   FEET
 6        P FS   600-7097-   - -   16 GA WHITE/VIOLET STRANDED WIRE                 1.1600   FEET
  7         FS   600-7009-   - -   16 GA WHITE STRANDED WIRE                        1.0000   FEET

 6          FS   605-0103-   - -   TUBING 3/8 BLACK                                  .9100   FEET
 6          IN   605-1011-   - -   TY-WRAP IDENT MARKER                             1.0000   EACH
 6          FS   654-1163-R  - -   SOCKET 20-14 GA.(REEL) AMP 61117-4               9.0000   EACH
 6          IN   654-1187-   - -   10 POS SOCKET HSNG AMP 1-480285-0                1.0000   EACH
```

```
5        IN  220-1428-  - -  MOTHER BD CABLE ASSY    B06482-0616 E15314    1.0000  EACH
 6       IN  000-0004-  - -  LABOR SUB-SYSTEMS                              .3400  EACH
 6       IN  000-0011-  - -  LABOR   QUALITY CONTROL                        .0680
 6     P FS  600-6002-  - -  14 GA RED STRANDED WIRE                       1.1600  FEET
  7      FS  600-6009-  - -  14 GA WHITE STRANDED WIRE                     1.0000  FEET

 6     P FS  600-6100-  - -  12 GA BLACK STRANDED WIRE                     1.1600  FEET
  7      FS  600-6109-  - -  12 GA WHITE STRANDED WIRE                     1.0000  FEET

 6     P FS  600-7004-  - -  16 GA YEL STRANDED WIRE                       1.1600  FEET
  7      FS  600-7009-  - -  16 GA WHITE STRANDED WIRE                     1.0000  FEET

 6       FS  605-0104-  - -  TUBING 1/2 BLACK                               .9100  FEET
 6       IN  605-1011-  - -  TY-WRAP IDENT MARKER                          1.0000  EACH
 6       IN  654-2090-  - -  CONN HOUSING 6 POS W/MT FLANGE     E15312     1.0000  EACH
 6       IN  654-3000-R - -  CONTACT MALE 30AMP 53892-2                    5.0000  EACH

5        IN  350-0011-  - -  225-21521-110 PC CONN SOLDER TYPE   PATREL   18.0000  EACH
5        IN  350-0021-  - -  225-22221-110 SOL TYPE                       12.0000  EACH
5        IN  350-0039-  - -  44 POS P.C.CONN SOLDER TYPE(CINCH)  PATREL   11.0000  EACH
5        IN  510-7698-  - -  PCB 2200LVP MOTHERBOARD                       1.0000  EACH
5        IN  654-1154-  - -  GROUND BUS 2 COND.LAMINATED 7 1/2"  PATREL    1.0000  EACH
5        IN  654-1172-  - -  12 POS PIN HEADER ASSY AMP 350213-1           1.0000  EACH
5        IN  654-1190-  - -  8 POS PIN HEADER AMP 350212-1       PATREL    1.0000  EACH
5        IN  654-3001-  - -  CONTACT MALE 30AMP 54326-2          PATREL    5.0000  EACH

4        IN  220-1405-  - -  DISK POWER CABLE ASSY   B06482-0589           1.0000  EACH
5        IN  000-0004-  - -  LABOR SUB-SYSTEMS                              .5000  EACH
5        IN  000-0011-  - -  LABOR   QUALITY CONTROL                        .1000
5      P FS  600-0000-  - -  WIRE 18 GA BLACK UL                 E15131   18.0000  FEET
5      P FS  600-0002-  - -  WIRE 18 GA RED UL                   E15131    6.0000  FEET
 6       FS  600-0009-  - -  WIRE 18 GA WHITE UL                           1.0000  FEET
5      P FS  600-0004-  - -  WIRE 18 GA YELLOW UL                E15131    6.0000  FEET
 6       FS  600-0009-  - -  WIRE 18 GA WHITE UL                           1.0000  FEET

5        FS  600-0009-  - -  WIRE 18 GA WHITE UL                 E15131    6.0000  FEET
5        FS  605-1004-  - -  CABLE TYE, PAN-TY PLTIM-M           E15131   20.0000  EACH
5        IN  605-1011-  - -  TY-WRAP IDENT MARKER                E14704    2.0000  EACH
5        FS  654-1163-R - -  SOCKET 20-14 GA.(REEL) AMP 61117-4           24.0000  EACH
```

```
 5          IN   654-1171-   - -  12 POS SOCKET HOUSING AMP 1-4802870                1.0000  EACH
 5          IN   654-1185-   - -  6 POS SOC HOUSING AMP 1-480270-0                   2.0000  EACH

4           IN   220-1424-   - -  FAN CORD ASSY           B06482-0614                1.0000  EACH
 5          IN   000-0004-   - -  LABOR SUB-SYSTEMS                                   .0990  EACH
 5          IN   000-0011-   - -  LABOR  QUALITY CONTROL                              .0200
 5          IN   420-1005-   - -  POWER CORD ROTRON FAN 16415                        1.0000  EACH
 5          IN   452-2191-XA- -  PLATE CORD              B10004-5033                1.0000  EACH
 5          FS   605-0102-   - -  TUBING 5/16 BLACK                                  1.6600  FEET
 5          IN   606-1424-   - -  1/2"DIA WHT SHRINK BLK NUM 220-1424                1.0000  EACH
 5          IN   654-1147-   - -  PIN HOUSING 1-480319-0               E15157        1.0000  EACH
 5          FS   654-1164-R  - -  PIN TERM 20-14 GA(REEL)AMP 61118-4   E15157        2.0000  EACH

4           IN   400-1013-   - -  FAN,ROTRON WR2H1  75CFM                            1.0000  EACH
4           IN   449-0101-   - -  FAN GUARD 4"(BLACK)D5300-1085                      1.0000  EACH
4           IN   449-0255-   - -  PLATE,NUT MOLDED            C6800-112              2.0000  EACH
4           IN   451-2800-XB- -  BOX SHORT CARD            E10004-5019              1.0000  EACH
4           IN   451-7044-XC- -  OBS EC15885D USE 451-7044-XD                       1.0000  EACH
4           IN   462-0141-   - -  SPCR, PHENOLIC CURRENT 4-250                       4.0000  EACH
4           FS   650-2120-   - -  4-40 X 3/8 PAN HD PHL MS SS SEMS                   1.0000  EACH
4           FS   650-2121-   - -  SCR   4-40   3/8  PHIL FLAT H MS SS                6.0000  EACH
4           FS   650-2160-   - -  4-40 X 1/2 PAN HD PHL MS SS SEMS                   2.0000  EACH
4           FS   650-2240-   - -  4-40 X 3/4 PAN HD PHL MS SS SEMS                  24.0000  EACH
4           FS   650-2320-   - -  SCR   4-40 1       PHIL PH MS SS                   2.0000  EACH
4           FS   650-3120-   - -  6-32 X 3/8 PAN HD PHL MS SS SEMS                   5.0000  EACH
4           FS   650-3247-   - -  SCR,6-32 X 3/4 PAN HD PHIL SEMS                    4.0000  EACH
4           IN   650-4120-   - -  8-32 X 3/8 PAN HD PHL MS SS SEMS                   4.0000  EACH
4           FS   652-0032-   - -  6-32 LOCK-NUT KEPS 511-061800-00                   4.0000  EACH
4           FS   652-2000-   - -  NUT   4-40UNC HEX REG PAT        SS                6.0000  EACH
4           FS   652-2005-   - -  4-40 LOCK-NUT KEPS SS                             26.0000  EACH
4           FS   652-4001-   - -  8-32 SQUARE NUT SS                                10.0000  EACH
4           FS   653-0003-   - -  WASHER, NO.4 NYLON 1/8 ID X 3/8 OD                25.0000  EACH
4           FS   653-2002-   - -  WASH    4 .123ID .2650D INT T    ST                2.0000  EACH
4           FS   653-2006-   - -  WASH    4 .125ID .3120D .016 FL  SS               37.0000  EACH
4           FS   653-3001-   - -  WASH    6 .150ID .2880D INT T    ST                4.0000  EACH
4           IN   654-1238-   - -  HEYCO STRAIN RELIEF SR5P-4                         1.0000  EACH
4           IN   655-0271-   - -  BUMPER RUBBER GREENE #2082                         4.0000  EACH
```

```
   3            IN   449-0348-XE- -   OBS EC15885D USE 449-0348-XF                         1.0000  EACH
   3            IN   451-0033-XB- -   CABINET REAR COVER                          070280   1.0000  EACH
   3            IN   451-0408-XB- -   SHELF                       D10004-5023              1.0000  EACH
   3            IN   451-1268-XC- -   BASE                        D10004-5017              1.0000  EACH
   3            IN   451-2315-XC- -   COVER TOP (WELDMENT)        E10004-5024              1.0000  EACH
   3            IN   451-2316-XC- -   COVER MEMORY BOARDS         D10004-5034              1.0000  EACH
   3            IN   451-2317-XC- -   COVER SIDE WLDMNT RH        E10004-5016              1.0000  EACH
   3            IN  '451-2318-XC- -   COVER SIDE WLDMNT LH        E10004-5016              1.0000  EACH
   3            IN   451-2320-XA- -   COVER I.O. BOARDS           C10004-5046              1.0000  EACH
   3            IN   451-3109-XB- -   OBS EC15885D USE 451-3109-XC                         1.0000  EACH
   3            IN   451-3110-XC- -   OBS EC15885D USE 451-3110-XD                         1.0000  EACH
   3            IN   451-3111-XB- -   OBS EC15885D USE 451-3111-XC                         1.0000  EACH
   3            IN   451-7041-XB- -   OBS EC15885D USE 451-7041-XC                         1.0000  EACH
   3            IN   451-7042-XB- -   OBS EC15885D USE 451-7042-XC                         1.0000  EACH
   3            IN   452-2202-XB- -   DEAD FRONT SILKSCRENING C10004-5048                  1.0000  EACH
   3            FS   650-3080-   - -  6-32 X 1/4 PAN HD PHL MS SS SEMS                     6.0000  EACH
   3            FS   650-3120-   - -  6-32 X 3/8 PAN HD PHL MS SS SEMS                    16.0000  EACH
   3            FS   650-3135-   - -  SCR PNL 6-32 X 3/8 PAN HD PH                         8.0000  EACH
   3            IN   650-4120-   - -  8-32 X 3/8 PAN HD PHL MS SS SEMS                     4.0000  EACH
   3            FS   650-6122-   - -  10-32X3/8 FLANGE WHIZ-LOCK MS ZINC                  26.0000  EACH
   3            FS   651-0406-   - -  RIVET POP 3/32 X .212                                8.0000  EACH
   3            FS   652-2005-   - -  4-40 LOCK-NUT KEPS SS                                6.0000  EACH
   3            IN   654-0035-   - -  BALL STUD P 116-447-8-495 TIN 701(6                  4.0000  EACH
   3            IN   654-0036-   - -  STUD REC,R C-4893-022-3B OR -24 TIN                  4.0000  EACH
   3            IN   655-0031-   - -  CASTER BALL 11/4D FLUSH W/RD NECK                    4.0000  EACH
   3            IN   655-0272-   - -  LEVELER 3/8-16X2" LG                                 2.0000  EACH
 1              IN   449-0348-XF- -   COVER FRONT (MOLDED)        E10004-5035              1.0000  EACH
 1              IN   451-2323-XA- -   OBS EC15894D   ECN15894                              1.0000  EACH
 1              IN   451-3109-XC- -   PNL LFT FRNT (MLD CVR)      D10004-5041              1.0000  EACH
 1              IN   451-3110-XD- -   PANEL CONTROL               C10004-5044              1.0000  EACH
 1              IN   451-3111-XC- -   PANEL RT FRNT (MLD CVR)     E10004-5040              1.0000  EACH
 1              IN   451-3637-XA- -   OBS EC15894D   ECN15894                              1.0000  EACH
 1              IN   451-3638-XA- -   OBS EC15894D   ECN15894                              1.0000  EACH
 1              IN   451-7040-XD- -   MTG BRKT FIXD DISK WELD     E10004-5039              1.0000  EACH
 1              IN   451-7041-XC- -   RTNR BKT PWR SUPPLY RH      C10004-5037              1.0000  EACH
 1              IN   451-7042-XC- -   RTNR BKT PWR SUPPLY LH      C10004-5037              1.0000  EACH
 1              IN   451-7043-XD- -   BRKT CONN LONG CARD BOX     C10004-5038              1.0000  EACH
```

```
1          IN    451-7044-XD- -   BRKT CONN SH CARD BOX    C10004-5032                  1.0000  EACH
1          IN    452-0190-XF- -   PLATE MTG DISKETTE       D10004-5026                  1.0000  EACH
1          IN    462-0471-  - -   STANDOFF M/F 3/8 HEX 1"L 6-32                         3.0000  EACH
1          IN    462-0473-  - -   STDF 4-40 .312 LG CL THU                              3.0000  EACH
1          IN    462-0474-  - -   STDF 6-32 X .625 LG CL BLIND                          2.0000  EACH
1          IN    478-1026-XA- -   OBS EC15894D   ECN15894                               1.0000  EACH
1          IN    605-1011-  - -   TY-WRAP IDENT MARKER                E15529            1.0000  EACH
1          IN    615-1586-XA- -   LABEL POWER SUPPLY     C-06611-0330 E15931            1.0000  EACH
1          FS    651-1058-  - -   STUD 4-40 X.312 CL FLUSH HD ST      E14831            8.0000  EACH
1          FS    652-0102-  - -   10-32 DODGE ULTRASRT 6041-3BRX.375  041880            4.0000  EACH
1          IN    654-1249-  - -   STRAIN RELIEF .300 HEYCO SR-6W-1                      1.0000  EACH
1          IN    660-0914-XA- -   RETAINER P.C.            C10004-5052                  2.0000  EACH

1          IN    270-0616-X - -   2200 LVP PWR SUPPLY CHASS ASSY 60HZ 030680            1.0000  EACH

 2         IN    220-0251-  - -   DISK POWER CORD          C06482-0609 041480           2.0000  EACH
  3        IN    000-0004-  - -   LABOR SUB-SYSTEMS                                      .4370  EACH
  3        IN    000-0011-  - -   LABOR   QUALITY CONTROL                                .0870
  3        FS    420-1003-  - -   POWER CABLE 3 COND 18 GA W/SVT                        4.5000  FEET
  3        IN    606-0251-  - -   3/8"DIA WHT SHRINK BLK NUM 220-0251                   1.0000  EACH
  3        FS    654-0050-R - -   #6 RING TONGUE RED BA16-6M(2K/REEL) E15131            1.0000  EACH
  3        FS    654-0062-R - -   #4 FORK LUG RED BA16F-6M (2K/REEL)  E15312            2.0000  EACH
  3        IN    654-2088-  - -   HOUSING 3 POS INTERNATIONAL SERIES                    1.0000  EACH
  3        IN    654-2089-R - -   CONTACT SOC FOR INTERNATIONAL SERIE                   3.0000  EACH

 2         IN    220-1008-  - -   9" FAN CABLE ASSY B6494-20 W/OFF/79 041480            1.0000  EACH
  3        IN    000-0004-  - -   LABOR SUB-SYSTEMS                                      .0830  EACH
  3        IN    000-0011-  - -   LABOR   QUALITY CONTROL                                .0170
  3      P FS    600-0000-  - -   WIRE 18 GA BLACK UL                 EC8399             .7500  FEET
  3        FS    600-0009-  - -   WIRE 18 GA WHITE UL                                    .7500  FEET
  3        FS    605-0015-  - -   #3 CLEAR TUBING                     EC8399             .5800  FEET
  3        IN    606-1008-  - -   3/8" DIA WHT SHRNK BLK NUM 220-1008 E11776            1.0000  EACH
  3        IN    654-1148-  - -   SOCKET HOUSING 1-480318-0                             1.0000  EACH
  3        FS  * 654-1163-R - -   SOCKET 20-14 GA.(REEL) AMP 61117-4  EC5402            2.0000  EACH
```

```
2      IN   220-1400-  - -  POWER CORD ASSY          B06482-0581 040480   1.0000  EACH
 3     IN   000-0004-  - -  LABOR SUB-SYSTEMS                              .0950  EACH
 3     IN   000-0011-  - -  LABOR   QUALITY CONTROL                        .0190
 3     IN   420-1096-  - -  POWER CORD,10 FT 18AWG                        1.0000  EACH
 3     IN   606-1400-  - -  1/2 DIA WHT SHRINK BLK NUM 220-1400           1.0000  EACH
 3     FS   654-0050-R - -  #6 RING TONGUE RED BA16-6M(2K/REEL)           1.0000  EACH
 3     IN   654-0187-R - -  FASTON LUG RT ANG INS RED 22-18GA             2.0000  EACH

2      IN   220-1411-  - -  AC CABLE ASSY            B06482-0604 040480   1.0000  EACH
 3     IN   000-0004-  - -  LABOR SUB-SYSTEMS                              .0680  EACH
 3     IN   000-0011-  - -  LABOR   QUALITY CONTROL                        .0140
 3   P FS   600-0000-  - -  WIRE 18 GA BLACK UL                  E15131   3.2500  FEET
 3     FS   600-0009-  - -  WIRE 18 GA WHITE UL                  E15131   3.2500  FEET
 3     FS   605-0015-  - -  #3 CLEAR TUBING                               2.7500  FEET
 3     IN   605-1011-  - -  TY-WRAP IDENT MARKER                          1.0000  EACH
 3     FS   654-0134-R - -  FASTON TERM 14-16 BLU AMP3-350819-2           4.0000  EACH

2      IN   220-1412-  - -  AC POWER CORD ASSY       B06482-0605 040480   1.0000  EACH
 3     IN   000-0004-  - -  LABOR SUB-SYSTEMS                              .3310  EACH
 3     IN   000-0011-  - -  LABOR   QUALITY CONTROL                        .0660
 3   P FS   600-0000-  - -  WIRE 18 GA BLACK UL                  E15131   2.7500  FEET
 3     FS   600-0009-  - -  WIRE 18 GA WHITE UL                  E15131   2.7500  FEET
 3     FS   605-0015-  - -  #3 CLEAR TUBING                               2.2000  FEET
 3     IN   605-1011-  - -  TY-WRAP IDENT MARKER                          1.0000  EACH
 3     FS   654-0134-R - -  FASTON TERM 14-16 BLU AMP3-350819-2           2.0000  EACH
 3     IN   654-1147-  - -  PIN HOUSING 1-480319-0                        1.0000  EACH
 3     FS   654-1164-R - -  PIN TERM 20-14 GA(REEL)AMP 61118-4            2.0000  EACH

2      IN   220-1413-  - -  WIRE & LUG ASSY PO 165               041480   1.0000  EACH
 3     IN   000-0004-  - -  LABOR SUB-SYSTEMS                              .0090  EACH
 3     IN   000-0011-  - -  LABOR   QUALITY CONTROL                        .0020
 3   P FS   600-0000-  - -  WIRE 18 GA BLACK UL                           1.1200  FEET
 3     FS   654-0068-R - -  #6 FORK LUG BLUE BA14-F6M (2K/REEL)           1.0000  EACH

2      IN   220-1414-  - -  WIRE & LUG ASSY PO 166               041480   1.0000  EACH
 3     IN   000-0004-  - -  LABOR SUB-SYSTEMS                              .0130  EACH
 3     IN   000-0011-  - -  LABOR   QUALITY CONTROL                        .0030
```

```
3       FS   600-0009-   - -   WIRE 18 GA WHITE UL                                         1.2500  FEET
3       FS   654-0068-R  - -   #6 FORK LUG BLUE BA14-F6M (2K/REEL)                         1.0000  EACH

2       IN   220-1415-   - -   WIRE & LUG ASSY PO 167                      041480          1.0000  EACH
 3      IN   000-0004-   - -   LABOR SUB-SYSTEMS                                            .0090  EACH
 3      IN   000-0011-   - -   LABOR   QUALITY CONTROL                                      .0020
 3    P FS   600-0000-   - -   WIRE 18 GA BLACK UL                                         1.2900  FEET
 3      FS   654-0068-R  - -   #6 FORK LUG BLUE BA14-F6M (2K/REEL)                         1.0000  EACH

2       IN   220-1416-   - -   WIRE & LUG ASSY PO 168                      041480          1.0000  EACH
 3      IN   000-0004-   - -   LABOR SUB-SYSTEMS                                            .0170  EACH
 3      IN   000-0011-   - -   LABOR   QUALITY CONTROL                                      .0030
 3      FS   600-0009-   - -   WIRE 18 GA WHITE UL                                         1.3300  FEET
 3      FS   654-0068-R  - -   #6 FORK LUG BLUE BA14-F6M (2K/REEL)                         1.0000  EACH

2       IN   220-1417-   - -   WIRE & LUG ASSY PO 169                      052780          2.0000  EACH
 3      IN   000-0004-   - -   LABOR SUB-SYSTEMS                                            .0170  EACH
 3      IN   000-0011-   - -   LABOR   QUALITY CONTROL                                      .0030
 3    P FS   600-7000-   - -   16 GA BLACK STRANDED WIRE                                    .7200  FEET
  4     FS   600-7009-   - -   16 GA WHITE STRANDED WIRE                                   1.0000  FEET

 3      FS   654-0068-R  - -   #6 FORK LUG BLUE BA14-F6M (2K/REEL)                         1.0000  EACH
 3      FS   654-0133-R  - -   FASTON TERM 18-22 RED AMP2-350803-2                         1.0000  EACH

2       IN   220-1418-   - -   WIRE & LUG ASSY PO 170                      042380          1.0000  EACH
 3      IN   000-0004-   - -   LABOR SUB-SYSTEMS                                            .0130  EACH
 3      IN   000-0011-   - -   LABOR   QUALITY CONTROL                                      .0030
 3    P FS   600-6100-   - -   12 GA BLACK STRANDED WIRE                                    .7500  FEET
  4     FS   600-6109-   - -   12 GA WHITE STRANDED WIRE                                   1.0000  FEET

 3      FS   654-0518-R  - -   TERM #6 12-10WIRE RING TONGUE INSUL E15458                  1.0000  EACH

2       IN   220-1425-   - -   FAN CORD EXTENTION ASSY B06482-0615 041480                  1.0000  EACH
 3      IN   000-0004-   - -   LABOR SUB-SYSTEMS                                            .0990  EACH
 3      IN   000-0011-   - -   LABOR   QUALITY CONTROL                                      .0200
 3      FS   420-0037-   - -   BLACK ZIP CORD 18  GA.     W/OFF-76 E15157                  3.0000  FEET
 3      FS   605-0102-   - -   TUBING 5/16 BLACK                                           2.5000  FEET
 3      IN   606-1425-   - -   1/2"DIA WHT SHRINK BLK NUM 220-1425 E15157                  1.0000  EACH
 3         FS   654-0062-R - -   #4 FORK LUG RED BA16F-6M (2K/REEL)              2.0000  EACH
 3         IN   654-1148-  - -   SOCKET HOUSING 1-480318-0             E15157    1.0000  EACH
 3         FS   654-1163-R - -   SOCKET 20-14 GA.(REEL) AMP 61117-4    E15157    2.0000  EACH

2          IN   220-1429-  - -   WIRE & LUG ASSY PO 173                041480    1.0000  EACH
 3         IN   000-0004-  - -   LABOR SUB-SYSTEMS                                .0170  EACH
 3         IN   000-0011-  - -   LABOR   QUALITY CONTROL                          .0030
 3       P FS   600-7000-  - -   16 GA BLACK STRANDED WIRE                        .6200  FEET
  4        FS   600-7009-  - -   16 GA WHITE STRANDED WIRE                       1.0000  FEET

 3         IN   654-0188-R - -   FASTON LUG RT ANG INS BLUE 16-14GA              1.0000  EACH

2          IN   220-1430-  - -   FAN CABLE ASSY          B06482-0477 042380      1.0000  EACH
 3         IN   000-0004-  - -   LABOR SUB-SYSTEMS                                .0760  EACH
 3         IN   000-0011-  - -   LABOR   QUALITY CONTROL                          .0150
 3         IN   420-1005-  - -   POWER CORD ROTRON FAN 16415                     1.0000  EACH
 3         FS   605-0015-  - -   #3 CLEAR TUBING                                 1.3300  FEET
 3         IN   606-1430-  - -   3/8"DIA WHT SHRINK BLK NUM 220-1430             1.0000  EACH
 3         FS   654-0062-R - -   #4 FORK LUG RED BA16F-6M (2K/REEL)              2.0000  EACH

2          IN   220-1431-  - -   MOTHER BD CBL EXTENTION C06482-0621 042380      1.0000  EACH
 3         IN   000-0004-  - -   LABOR SUB-SYSTEMS                                .2030  EACH
 3         IN   000-0011-  - -   LABOR   QUALITY CONTROL                          .0410
 3       P FS   600-7000-  - -   16 GA BLACK STRANDED WIRE                       1.2500  FEET
  4        FS   600-7009-  - -   16 GA WHITE STRANDED WIRE                       1.0000  FEET

 3       P FS   600-7002-  - -   16 GA RED STRANDED WIRE                         1.2500  FEET
  4        FS   600-7009-  - -   16 GA WHITE STRANDED WIRE                       1.0000  FEET

 3       P FS   600-7003-  - -   16 GA ORANGE STRANDED WIRE                      1.2500  FEET
  4        FS   600-7009-  - -   16 GA WHITE STRANDED WIRE                       1.0000  FEET

 3       P FS   600-7006-  - -   16 GA BLUE STRANDED WIRE                        1.2500  FEET
  4        FS   600-7009-  - -   16 GA WHITE STRANDED WIRE                       1.0000  FEET

 3         FS   600-7009-  - -   16 GA WHITE STRANDED WIRE                       1.2500  FEET
 3         FS   600-7092-  - -   WIRE,16 AWG WHITE RED                           1.2500  FEET
 3       P FS   600-7095-  - -   16 GA WHITE/GREEN STRANDED WIRE                 1.2500  FEET
  4        FS   600-7009-  - -   16 GA WHITE STRANDED WIRE                       1.0000  FEET
```

```
3         FS  600-7096-   - -  WIRE 16 ANG WHITE BLU                              1.2500  FEET
 3      P FS  600-7097-   - -  16 GA WHITE/VIOLET STRANDED WIRE                   1.2500  FEET
  4       FS  600-7009-   - -  16 GA WHITE STRANDED WIRE                          1.0000  FEET

3         FS  605-1004-   - -  CABLE TYE, PAN-TY PLTIM-M                          3.0000  EACH
3         IN  605-1011-   - -  TY-WRAP IDENT MARKER                               1.0000  EACH
3         FS  654-1163-R  - -  SOCKET 20-14 GA.(REEL) AMP 61117-4                 9.0000  EACH
3         FS  654-1164-R  - -  PIN TERM 20-14 GA(REEL)AMP 61118-4                 9.0000  EACH
3         IN  654-1187-   - -  10 POS SOCKET HSNG AMP 1-480285-0                  1.0000  EACH

2         IN  270-0157-   - -  2200 LVP HEATSINK ASSY                 062380      1.0000  EACH
 3        IN  270-3164-   - -  HEATSINK HARNESS        D06482-0606                1.0000  EACH
  4       IN  000-0004-   - -  LABOR SUB-SYSTEMS                                  2.7500  EACH
  4       IN  000-0011-   - -  LABOR   QUALITY CONTROL                             .5500
  4     P FS  600-6002-   - -  14 GA RED STRANDED WIRE                E15157      4.0000  FEET
   5      FS  600-6009-   - -  14 GA WHITE STRANDED WIRE                          1.0000  FEET

  4     P FS  600-6003-   - -  14 GA ORANGE STRANDED WIRE             E15157      3.9200  FEET
   5      FS  600-6009-   - -  14 GA WHITE STRANDED WIRE                          1.0000  FEET

  4     P FS  600-6100-   - -  12 GA BLACK STRANDED WIRE              E15157      5.0000  FEET
   5      FS  600-6109-   - -  12 GA WHITE STRANDED WIRE                          1.0000  FEET

  4     P FS  600-7000-   - -  16 GA BLACK STRANDED WIRE              E15157      8.5000  FEET
   5      FS  600-7009-   - -  16 GA WHITE STRANDED WIRE                          1.0000  FEET
  4     P FS  600-7001-   - -  16 GA BROWN STRANDED WIRE              E15157      2.3100  FEET
   5      FS  600-7009-   - -  16 GA WHITE STRANDED WIRE                          1.0000  FEET

  4     P FS  600-7002-   - -  16 GA RED STRANDED WIRE                E15157      5.6800  FEET
   5      FS  600-7009-   - -  16 GA WHITE STRANDED WIRE                          1.0000  FEET

  4     P FS  600-7003-   - -  16 GA ORANGE STRANDED WIRE             E15157      1.4600  FEET
   5      FS  600-7009-   - -  16 GA WHITE STRANDED WIRE                          1.0000  FEET

  4     P FS  600-7004-   - -  16 GA YEL STRANDED WIRE                E15157      1.8300  FEET
   5      FS  600-7009-   - -  16 GA WHITE STRANDED WIRE                          1.0000  FEET
 4        P FS   600-7006-   - -   16 GA BLUE STRANDED WIRE             E15157      1.2000  FEET
  5         FS   600-7009-   - -   16 GA WHITE STRANDED WIRE                        1.0000  FEET

 4        P FS   600-7007-   - -   16 GA VIOLET STRANDED WIRE           E15157      1.2500  FEET
  5         FS   600-7009-   - -   16 GA WHITE STRANDED WIRE                        1.0000  FEET

 4        P FS   600-7008-   - -   16 GA GRAY STRANDED WIRE             E15157      1.5800  FEET
  5         FS   600-7009-   - -   16 GA WHITE STRANDED WIRE                        1.0000  FEET

 4          FS   600-7009-   - -   16 GA WHITE STRANDED WIRE            E15157      1.7500  FEET
 4        P FS   600-7030-   - -   16 GA WIRE COLOR ORN/BLK             E15157      2.1000  FEET
 4          FS   600-7090-   - -   WIRE,16 AWG WHITE BLACK              E15157      2.2900  FEET
 4          FS   600-7091-   - -   WIRE,16 AWG WHITE BRN                E15157      1.1200  FEET
 4          FS   600-7092-   - -   WIRE,16 AWG WHITE RED                E15157      1.7500  FEET
 4        P FS   600-7095-   - -   16 GA WHITE/GREEN STRANDED WIRE      E15157      2.1000  FEET
  5         FS   600-7009-   - -   16 GA WHITE STRANDED WIRE                        1.0000  FEET

 4          FS   600-7096-   - -   WIRE 16 ANG WHITE BLU                E15157      1.7500  FEET
 4          FS   605-1004-   - -   CABLE TYE, PAN-TY PLTIM-M            E15312     14.0000  EACH
 4          FS   605-1006-   - -   DES CHG. CABLE TIE PLATE 2S          E15312      2.0000  EACH
 4          IN   605-1011-   - -   TY-WRAP IDENT MARKER                             1.0000  EACH
 4          FS   654-0068-R  - -   #6 FORK LUG BLUE BA14-F6M (2K/REEL)              6.0000  EACH
 4          FS   654-0075-R  - -   #10 RNG TNG YLO BA10-10 MIK          E15157      1.0000  EACH
 4          FS   654-0134-R  - -   FASTON TERM 14-16 BLU AMP3-350819-2              1.0000  EACH
 4          FS   654-0518-R  - -   TERM #6 12-10WIRE RING TONGUE INSUL E15157      2.0000  EACH
 4          IN   654-1150-   - -   SOCKET HOUSING 1-480303-0            E14902      2.0000  EACH
 4          FS   654-1163-R  - -   SOCKET 20-14 GA.(REEL) AMP 61117-4              19.0000  EACH
 4          IN   654-1171-   - -   12 POS SOCKET HOUSING AMP 1-4802870 E14902      1.0000  EACH
 4          IN   654-1185-   - -   6 POS SOC HOUSING AMP 1-480270-0     E14902      1.0000  EACH
 4          IN   654-2090-   - -   CONN HOUSING 6 POS W/MT FLANGE       E15312      1.0000  EACH
 4          IN   654-3000-R  - -   CONTACT MALE 30AMP 53892-2           E14902      6.0000  EACH

3           IN   334-0031-   - -   RES .006 OHM  3W  5% WW NON-IND                  1.0000  EACH
3           IN   334-0032-   - -   RES .020 OHM  3W  3% WW NON-IND                  1.0000  EACH
3           IN   375-1048-   - -   TSTR 2N5301  200W  40V AP NPN S T03              2.0000  EACH
3           IN   375-1072-   - -   TRANSISTOR 2N5685 300W S NPN TO-3                1.0000  EACH
3           IN   375-9014-   - -   INSULATOR XTOR MOUNT WECKESSER TM-1              3.0000  EACH
3           IN   375-9020-   - -   MICA WSHR (LARGE) FOR POWER X1STORS              3.0000  EACH
3           IN   380-3000-   - -   DIO   1N1200A 100V  12A RECT S D04               2.0000  EACH
3           IN   380-3007-   - -   DIO   1N1183A  50V  35A RECT S D05               2.0000  EACH
```

```
3        IN   380-9001-  - -  MICA WSHF .26IDX.880DX.003THK(3007)                4.0000  EACH
3        IN   380-9002-  - -  MICA WSHR .19IDX.630DX.003THK(3000)                2.0000  EACH
3        IN   462-0193-  - -  INSULATED STANDOFF (FORKED)                        3.0000  EACH
3        IN   462-0476-  - -  STDF 6-32 M/F 1/4 HX 1"LG 9743A0632                3.0000  EACH
3        IN   478-1025-XP- -  HEATSINK REG BOARD      D10004-5030                1.0000  EACH
3        FS   600-9015-  - -  WIRE 14 GA TINNED COPPER BUS (UL)                  2.0833  FEET
3        FS   605-0000-  - -  TUBING #10 CLEAR                                   2.0000  FEET
3        FS   650-3100-  - -  SCR   6-32   5/16 PHIL PH MS SS                    3.0000  EACH
3        FS   650-3120-  - -  6-32 X 3/8 PAN HD PHL MS SS SEMS                   2.0000  EACH
3        FS   650-3160-  - -  6-32 X 1/2 PAN HD PHL MS SS SEMS                   5.0000  EACH
3        FS   650-3200-  - -  SCR   6-32   5/8  PHIL PH MS SS                    1.0000  EACH
3        FS   652-0045-  - -  1/4-28 HEX NUT 7/16 A.F.X 3/16 TH                  2.0000  EACH
3        FS   652-3004-  - -  NUT   6-32UNC HEX SMALL PAT      SS                6.0000  EACH
3        FS   652-6000-  - -  NUT  10-32UNC HEX REG PAT        SS                2.0000  EACH
3        FS   653-3000-  - -  WASH    6 .149ID .3750D .016 FL  SS                9.0000  EACH
3        FS   653-3001-  - -  WASH    6 .150ID .2880D INT T    ST                7.0000  EACH
3        FS   653-6000-  - -  WASH   10 .203ID .4380D .032 FL  SS                2.0000  EACH
3        FS   653-6001-  - -  WASH   10.204ID .3810D INT T    ST                 2.0000  EACH
3        FS   653-6006-  - -  WASH  1/4 .265ID .5000D .032 FL  SS                2.0000  EACH
3        FS   653-6009-  - -  WASH  1/4 .267ID .4780D INT T    ST                2.0000  EACH
3        IN   653-7002-  - -  WASH,.27 IDX.31 ODX.06THK FL TEF                   4.0000  EACH
3        IN   654-0190-  - -  SOLDER GROUND LUG DO-5                             2.0000  EACH
3        IN   654-0191-  - -  SOLDER GROUND LUG DO-4                             2.0000  EACH
3        IN   654-1006-  - -  #6 GROUND LUG                                      4.0000  EACH
3        IN   654-1256-  - -  CLAMP, CABLE 1/2 INCH                              2.0000  EACH
3        IN   654-1319-  - -  SHOULDER BUSHING DO-4                              2.0000  EACH
3        FS   660-0123-  - -  THERMAL COMPOUND DOW#340(14 OZ TUBE                 .0110  EACH

2        IN   270-3158-  - -  XFMR HARNESS ASSY 60 HZ D06482-C590 040480         1.0000  EACH
 3       IN   000-0004-  - -  LABOR SUB-SYSTEMS                                   .2410  EACH
 3       IN   000-0011-  - -  LABOR   QUALITY CONTROL                             .0480
 3       IN   410-0169-  - -  XFMR CVT MDL 2200LVP        C5068-169              1.0000  EACH
 3     P FS   600-0002-  - -  WIRE 18 GA RED UL                       E14770      .7500  FEET
  4      FS   600-0009-  - -  WIRE 18 GA WHITE UL                                1.0000  FEET

 3     P FS   600-0004-  - -  WIRE 18 GA YELLOW UL                    E14770      .7500  FEET
  4      FS   600-0009-  - -  WIRE 18 GA WHITE UL                                1.0000  FEET
```

```
3      FS   605-0123-   - -   SHRINK TUBING TYPE RNF 3/16 ID BLK  E14770       .3300  FEET
3      FS   654-0068-R  - -   #6 FORK LUG BLUE BA14-F6M (2K/REEL)             6.0000  EACH
3      FS   654-0084-R  - -   TERMINAL FASTON .250X.032 PQ14R258M             2.0000  EACH
3      FS   654-0518-R  - -   TERM #6 12-10WIRE RING TONGUE INSUL             1.0000  EACH
3      FS   654-1163-R  - -   SOCKET 20-14 GA.(REEL) AMP 61117-4              4.0000  EACH

2      IN   300-3074-   - -   7300 UF 40V ELECTROLYTIC CAP        052780      2.0000  EACH
2      IN   300-3087-   - -   CAP 161K UF 10V ELECT ESR.07        030680      1.0000  EACH
2      IN   300-3203-   - -   4UF,660 VAC,60HZ,NON PCB            030680      1.0000  EACH
2      IN   300-9006-   - -   CAP CLAMP 2 1/2 INCH 3 LUG          030680      1.0000  EACH
2      IN   300-9009-   - -   CAP CLAMP 1 1/4 INCH 2 LUG CMC-22   052780      2.0000  EACH
2      IN   300-9024-   - -   CAP BOOT FOR #300-3042 GE #614      040480      1.0000  EACH
2      IN   300-9026-   - -   CLAMP CAPACITOR OVAL 2.062"X 1.250" 030680      1.0000  EACH
2      IN   310-1206-   - -   TERM BLOCK  6 TWIN 1.125W           040480      1.0000  EACH
2      IN   310-1210-   - -   TERM BLOCK 10 TWIN 1.125W           030680      1.0000  EACH
2      IN   325-0055-   - -   SWITCH DPST ROCKER ON/OFF           030680      1.0000  EACH
2      IN   325-2117-   - -   SW SLIDE DPDT AC LINE 115/230       030680      1.0000  EACH
2      IN   332-3010-   - -   RES   1K OHM   1W 10% FIXED COMP    030680      2.0000  EACH
2      IN   360-0000-   - -   FUSE HOLDER 90 DEGREE CONTACT       030680      1.0000  EACH
2      IN   360-1040-SB- -    4.0 AMP 250V SB GLASS 3AG           030680      1.0000  EACH
2      IN   360-9000-   - -   RUBBER WSHR FOR 360-0000 / 360-0001 040480      1.0000  EACH
2      IN   360-9002-   - -   HEX NUT FOR 360-0000 / 360-0001     040480      1.0000  EACH
2      IN   360-9003-   - -   LOCK WSHR LF#905023(FOR 360-0000/1) 040480      1.0000  EACH
2      IN   380-5001-   - -   250 VOLT VARISTOR V250LA20          030680      1.0000  EACH
2      IN   400-1003-   - -   FAN MUFFIN M747 MARK IV             030680      1.0000  EACH
2      IN   410-2005-   - -   LINE FILTER 5 AMP CORCOM 5K1        032680      1.0000  EACH
2      IN   449-0101-   - -   FAN GUARD 4"(BLACK)D5300-1085       030680      1.0000  EACH
2      IN   449-0247-   - -   HANDLE,FACEPLATE        B6815-123   052780      1.0000  EACH
2      IN   451-1146-XE- -    OBSOLETE USE 451-1146-XF E15530     043080      1.0000  EACH
2      IN   451-2319-XB- -    COVER POWER SUPPLY     D10004-5031  040480      1.0000  EACH
2      IN   451-2321-XB- -    COVER LINE FILTER WLDMT D10004-5049 040880      1.0000  EACH
2      IN   451-7045-XB- -    OBSOLETE USE 451-7045-XC E15350D    042380      1.0000  EACH
2      IN   462-0015-   - -   SPCR  .140 ID  .250 OD  .250 L RD B 040480      2.0000  EACH
2      IN   478-1030-XA- -    CVR TRMNL BLK (LG 5.25) B10004-5051 042380      1.0000  EACH
2      IN   510-7749-   - -   PCB A.C. SWITCH UNIVERSAL BD        040480      1.0000  EACH
2      FS   650-3120-   - -   6-32 X 3/8 PAN HD PHL MS SS SEMS    052780     17.0000  EACH
2      FS   650-3160-   - -   6-32 X 1/2 PAN HD PHL MS SS SEMS    052780     10.0000  EACH
```

```
2          FS   650-3247-  - -  SCR,6-32 X 3/4 PAN HD PHIL SEMS        042280    4.0000  EACH
2          FS   650-3280-  - -  SCR   6-32   7/8  PHIL PH MS SS        040480    2.0000  EACH
2          FS   650-6122-  - -  10-32X3/8 FLANGE WHIZ-LOCK MS ZINC     040480    4.0000  EACH
2          FS   651-0030-  - -  .SCREW,SELF TAP T-B #4X1/2"L PNHD PH   041480    2.0000  EACH
2          FS   652-0032-  - -  6-32 LOCK-NUT KEPS 511-061800-00       052780   10.0000  EACH
2          FS   652-2004-  - -  NUT   4-40UNC HEX SMALL PAT         SS 040480    2.0000  EACH
2          FS   652-3004-  - -  NUT   6-32UNC HEX SMALL PAT         SS 040480    5.0000  EACH
2          FS   653-2006-  - -  WASH    4 .125ID .312OD .016 FL     SS 042280    8.0000  EACH
2          FS   653-3000-  - -  WASH    6 .149ID .375OD .016 FL     SS 052780   23.0000  EACH
2          FS   653-3001-  - -  WASH    6 .150ID .288OD INT T       ST 040480    8.0000  EACH
2          FS   653-3003-  - -  WASH    6 .141ID .253OD SPLIT       SS 040480    3.0000  EACH
2          IN   654-0125-  - -  TERMINAL LUG,.200 HOLE 1/4 SPADE       052780    4.0000  EACH
2          IN   654-1010-  - -  #10 GROUND LUG                         040480    4.0000  EACH
2          IN   654-1082-  - -  TERMINAL BLOCK JUMPER CJ 141-J         040480    9.0000  EACH
2          IN   654-1202-  - -  GROMMET 3/8 ID FOR 1/2 HOLE            040480    1.0000  EACH
2          IN   654-1222-  - -  GROMMET 1/2 ID FOR 3/4 HOLE 3/32PNL    041480    1.0000  EACH
2          IN   654-1238-  - -  HEYCO STRAIN RELIEF SR5P-4             030680    1.0000  EACH
2          IN   654-1317-  - -  GROMMET EXTRUDED CHANNEL .037-.105     041480     .4167  FEET
2          IN   654-2031-  - -  PIN HOUSING 10 POS AMP 1-480286-0      052780    1.0000  EACH
```

## NOTES

## NOTES

# APPENDIX A

# 2200LVP ERROR CODES

Refer to 2200VP BASIC-2 Language Reference Manual, WL #700-4080C (IV.C.2), for more detailed information concerning the nature of, and the recovery from the following errors.

| Error Code: | NONRECOVERABLE ERRORS |
|---|---|
| **Misc. Errors:** | |
| A01 | memory exceeded (overlap: text & symbol table) |
| A02 | memory exceeded (overlap: text & value stack) |
| A03 | not enough memory (LISTDC, MOVE, COPY) |
| A04 | stack overflow (operator stack) |
| A05 | line too long |
| A06 | program protected |
| A07 | illegal immediate mode statement |
| A08 | statement not legal here |
| A09 | program not resolved |
| **Syntax Errors:** | |
| S10 | missing left parenthesis |
| S11 | missing right parenthesis |
| S12 | missing equal sign |
| S13 | missing comma |
| S14 | missing asterisk |
| S15 | missing angle brackets |
| S16 | missing letter |
| S17 | missing hex digit |
| S18 | missing relation operator |
| S19 | missing required word |
| S20 | expected end of statement |
| S21 | missing line number |
| S22 | illegal PLOT argument |
| S23 | missing literal string |
| S24 | illegal expression or missing variable |
| S25 | missing numeric scalar variable |
| S26 | missing array variable |
| S27 | missing numeric array |
| S28 | missing alpha array |
| S29 | missing alpha variable |
| **Program Errors:** | |
| P32 | starting address greater than ending address |
| P33 | line number conflict |
| P34 | illegal value |
| P35 | no program |
| P36 | underfined line number or CONTINUE illegal |
| P37 | underfined special function subroutine |
| P38 | underfined FN function |
| P39 | FN nested too deep |
| P40 | NEXT without FOR |
| P41 | RETURN without GOSUB |

| | |
|---|---|
| P42 | illegal image |
| P43 | illegal matrix operand |
| P44 | matrix not square |
| P45 | operand dimensions not compatible |
| P46 | illegal microcommand |
| P47 | missing buffer variable |
| P48 | illegal device specification |
| P49 | interrupt table full |
| P50 | illegal dimensions or variable length |
| P51 | variable or value too short |
| P52 | variable or value too long |
| P53 | noncommon variables already defined |
| P54 | common variable required |
| P55 | undefined array |
| P56 | illegal subscripts |
| P57 | illegal STR () arguments |
| P58 | illegal field/delimiter specification |
| P59 | illegal redimension |

Error Code: RECOVERABLE ERRORS

Computation Errors:

| | |
|---|---|
| C60 | underflow |
| C61 | overflow |
| C62 | division by zero |
| C63 | zero divided by zero, or zero raised to zero power |
| C64 | zero raised to negative power |
| C65 | negative number raised to noninteger power |
| C66 | SQR of negative power |
| C67 | LOG of zero |
| C68 | LOG of negative power |
| C69 | argument too large |

Execution Errors:

| | |
|---|---|
| X70 | insufficient data |
| X71 | value exceeds format |
| X72 | singular matrix |
| X73 | illegal INPUT data |
| X74 | wrong variable type |
| X75 | illegal number |
| X77 | invalid partition reference |

Disk Errors:

| | |
|---|---|
| D80 | file not open |
| D81 | file full |
| D82 | file not in catalog |
| D83 | file already catalogued |
| D84 | file not scratched |
| D85 | index full |
| D86 | catalog end error |
| D87 | no end file |
| D88 | wrong record type |
| D89 | sector address beyond EOF |

I/O Errors:

| | |
|---|---|
| I90 | disk hardware error (X'CO' not rec'd) |
| I91 | disk hardware error |
| I92 | disk hardware error (timeout) |
| I93 | disk format error |
| I94 | format key engaged |
| I95 | seek error |
| I96 | CRC error |
| I97 | LRC error |
| I98 | illegal sector address |
| I99 | read-after-write error |

NOTES

## APPENDIX B

## MECHANICAL DRAWINGS

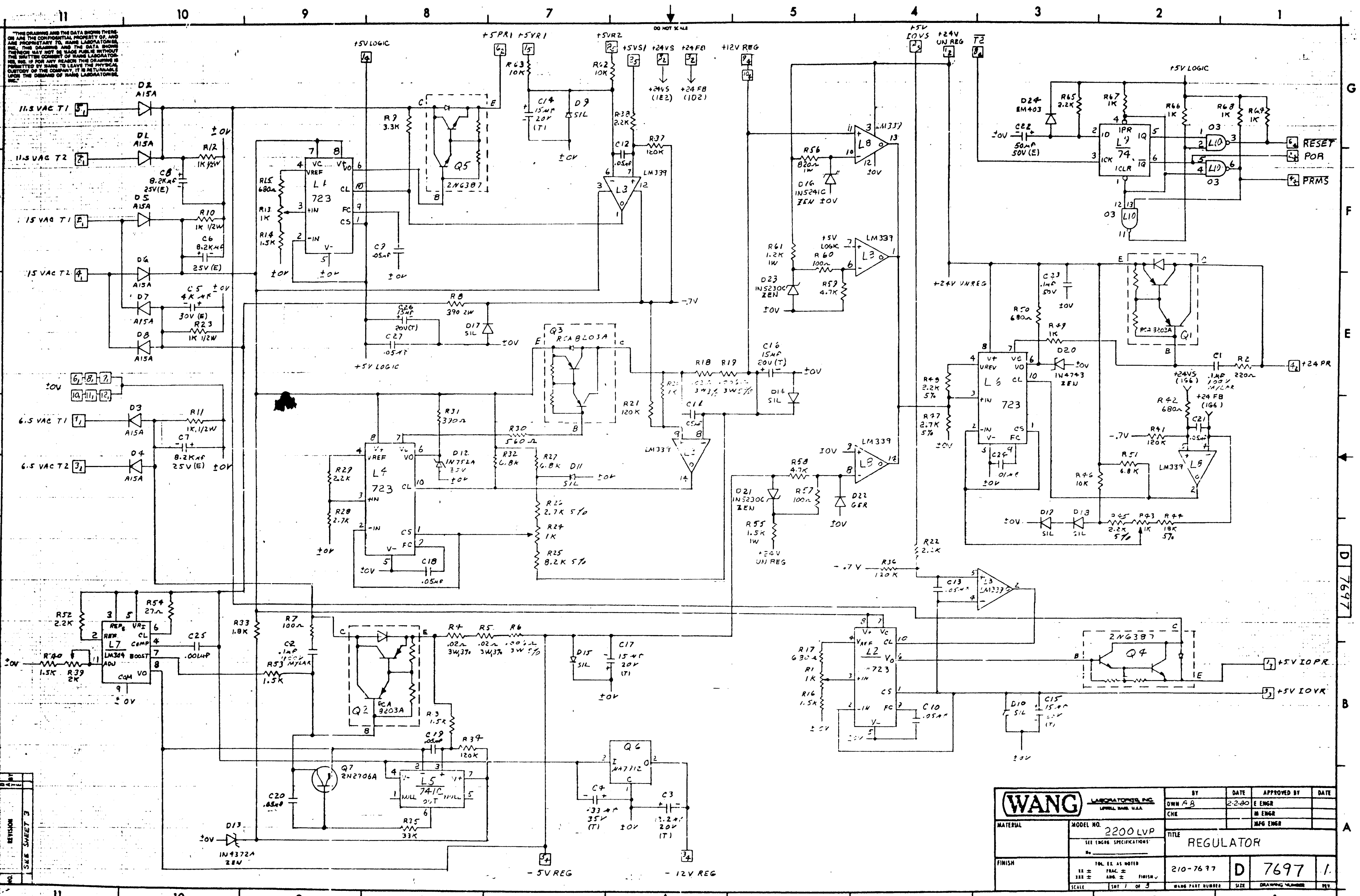
TO BE SUPPLIED

## NOTES

# APPENDIX C
# SCHEMATICS

The following schematics are contained in this Appendix.

| | |
|---|---|
| 210-7697 | Regulator |
| 7698-100 | Interconnection Diagram |
| 210-7698 | Motherboard |



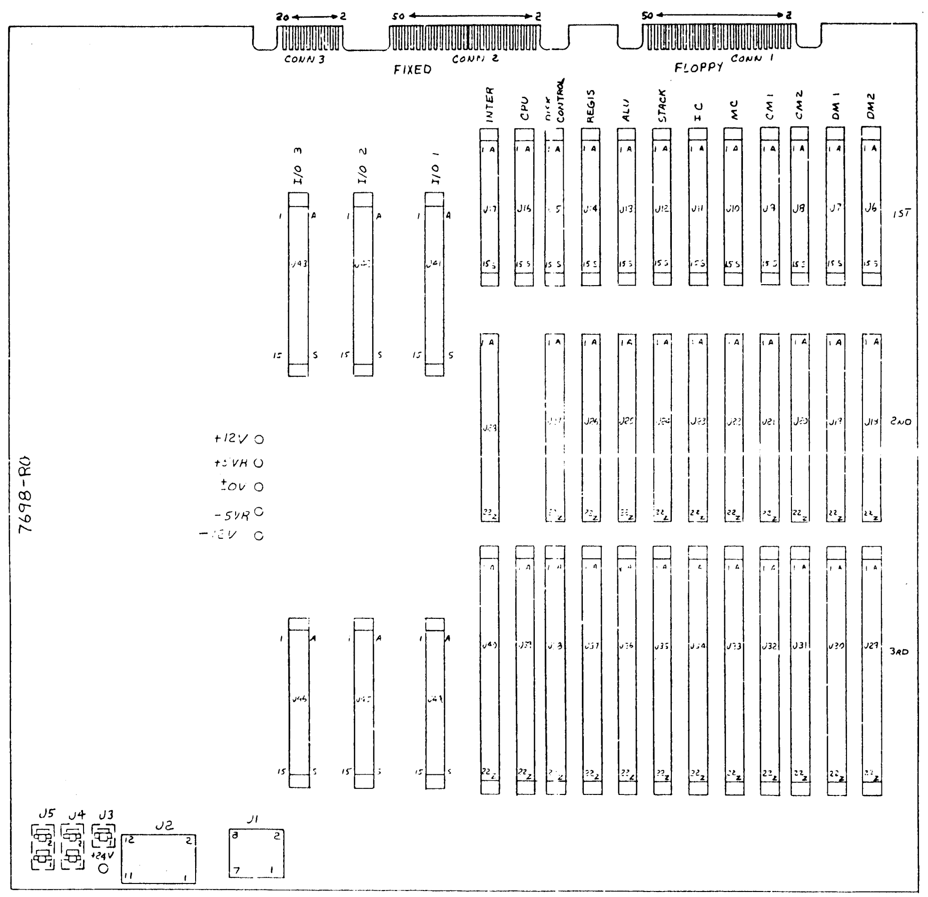
| | | | | | |
|---|---|---|---|---|---|
| **WANG** LABORATORIES, INC. LOWELL, MASS. U.S.A. | | BY | DATE | APPROVED BY | DATE |
| | | DWN A.B. | 2-2-80 | E ENGR | |
| | | CHK | | M ENGR | |
| | | | | MFG ENGR | |
| MATERIAL | MODEL NO. 2200 LVP | TITLE REGULATOR | | | |
| | SEE ENGRG. SPECIFICATIONS No. | | | | |
| FINISH | TOL. EX. AS NOTED XX ± FRAC. ± XXX ± ANG. ± FINISH ✓ | 210-7697 | D | 7697 | 1. |
| | SCALE / SHT 1 OF 3 | WANG PART NUMBER | SIZE | DRAWING NUMBER | REV. |

| NO. | REVISION |
|---|---|
| | SEE SHEET 3 |

D 7697

DO NOT SCALE

| COMPONENT | TYPE | W.L PART NO. |
|---|---|---|
| J1 | 8 POS PIN HEADER | 654-1170 |
| J2 | 12 POS PIN HEADER | 654-1172 |
| J3,4,5 | MALE CONTACT | 654-3001 |
| | | |
| J6-17, 41-46 | 30 PIN CONN | 350-0011 |
| J18 - 28 | 44 PIN CONN | 350-0039 |
| J29 - J40 | 4? PIN CONN | 350-0021 |
| | | |



E-REV
0

| REVISION | | |
|---|---|---|
| NO. | 0 | 1 |
| REVISION | ORIGINATED PER DWR # E920 APP'D. [illegible] | REV PER ECN # 14541 APP'D [illegible] |
| DATE | 2-7-80 | 3-3-80 |
| BY | A.B | [illegible] |

| | | BY | DATE | APPROVED BY | DATE |
|---|---|---|---|---|---|
| WANG LABORATORIES, INC. LOWELL, MASS. U.S.A. | | DWN [illegible] | 2/7/80 | E ENGR [illegible] | [illegible] |
| | | CHK [illegible] | [illegible] | M ENGR | |
| | | | | MFG ENGR | |
| MATERIAL | MODEL NO. 2200 LVP | TITLE MOTHERBOARD | | | |
| | SEE ENGRG. SPECIFICATIONS No. | | | | |
| FINISH | TOL. EX. AS NOTED .XX ± .XXX ± FRAC. ± ANG. ± FINISH ✓ | 210-7698 | D | 7698 | 1 |
| | SCALE / SHT 6 OF 6 | WANG PART NUMBER | SIZE | DRAWING NUMBER | REV |

D 7698

| NO. | REVISION |
|---|---|
| | SEE SHEET 6 |

| SIGNAL | I/O 1,2,3 | S.P. | JACK | INTER-FACE 7694 | CPU 7696 | DISK C 7695 | REG 6793 | ALU 6792 | STACK 6791 | I.C. 6790 | MC 6789 | CM 1 7588 | CM 2 7586 | DM 1 7587 | DM2 7587 | CONN 1 | CONN 2 | CONN 3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ±0V | $C_1$ | ±0V | J5-1 | $1_1$ | $1_1$ | $A_1$ | $1_1$ | $1_1$ | $1_1$ | $1_1$ | $1_1$ | $1_1$ | $1_1$ | $1_1$ | $1_1$ | 1,3,5, | 1,3,5, | 3,4, |
| | $3_1$ | | J5-2 | $A_1$ | $A_1$ | $A_1$ | $A_1$ | $A_1$ | $A_1$ | $A_1$ | $A_1$ | $A_1$ | $A_1$ | $A_1$ | $A_1$ | 7,9,11, | 7,9,11, | 7,8, |
| | $P_3$ | | J2-1 | $Z_3$ | $Z_3$ | $Z_3$ | $Z_3$ | $Z_3$ | $Z_3$ | $Z_3$ | $Z_3$ | $Z_3$ | $Z_3$ | $Z_3$ | $Z_3$ | 13,15, | 13,15 | 11,12, |
| | $13_3$ | | J2-3 | $22_3$ | $22_3$ | $22_3$ | $22_3$ | $22_3$ | $22_3$ | $22_3$ | $22_3$ | $22_3$ | $22_3$ | $22_3$ | $22_3$ | 17,19,21 | 17,19, | 15,16, |
| | | | J2-5 | | | | | | | | $8_3$ | | | $S_2$ | | 23,25, | 21,23 | 19,20 |
| | | | J2-7 | | | | | | | | | | | | | 27,29, | 25,27, | |
| | | | J2-9 | | | | | | | | | | | | | 31,33, | 29,31, | |
| | | | J2-11 | | | | | | | | | | | | | 35,37, | 33,35, | |
| | | | | | | | | | | | | | | | | 39,41, | 37,39, | |
| | | | | | | | | | | | | | | | | 43,45, | 41,43, | |
| | | | | | | | | | | | | | | | | 47,49, | 45,47, | |
| | | | | | | | | | | | | | | | | | 49 | |
| +5VR | $B_1$ | +5VR | J2-6 | $B_1$ | $B_1$ | $B_1$ | $B_1$ | $B_1$ | $B_1$ | $B_1$ | $B_1$ | $B_1$ | $B_1$ | $B_1$ | $B_1$ | | | |
| | $2_1$ | | J2-12 | $2_1$ | $2_1$ | $2_1$ | $2_1$ | $2_1$ | $2_1$ | $2_1$ | $2_1$ | $2_1$ | $2_1$ | $2_1$ | $2_1$ | | | |
| | $R_3$ | | J1-2 | $Y_3$ | $Y_3$ | $Y_3$ | $Y_3$ | $Y_3$ | $Y_3$ | $Y_3$ | $Y_3$ | $Y_3$ | $Y_3$ | $Y_3$ | $Y_3$ | | | |
| | $19_3$ | | | $21_3$ | $21_3$ | $21_3$ | $21_3$ | $21_3$ | $21_3$ | $21_3$ | $21_3$ | $21_3$ | $21_3$ | $21_3$ | $21_3$ | | | |
| −5VR | | −5VR | J1-6 | $C_1$ | $C_1$ | $C_1$ | $C_1$ | $C_1$ | $C_1$ | $C_1$ | $C_1$ | $C_1$ | $C_1$ | $C_1$ | $C_1$ | | | |
| | | | J2-2 | $3_1$ | $3_1$ | $3_1$ | $3_1$ | $3_1$ | $3_1$ | $3_1$ | $3_1$ | $3_1$ | $3_1$ | $3_1$ | $3_1$ | | | |
| | | | J2-8 | | | | | | | | | | | | | | | |
| +12V | $S_3$ | +12V | J1-4 | $S_1$ | $S_1$ | $S_1$ | $S_1$ | $S_1$ | $S_1$ | $S_1$ | $S_1$ | $S_1$ | $S_1$ | $S_1$ | $S_1$ | | | |
| | $15_3$ | | | $15_1$ | $15_1$ | $15_1$ | $15_1$ | $15_1$ | $15_1$ | $15_1$ | $15_1$ | $15_1$ | $15_1$ | $15_1$ | $15_1$ | | | |
| −12V | $N_3$ | −12V | J1-8 | $Z_2$ | | $Z_2$ | | | | | | | | | | | | |
| | $12_3$ | | | $22_2$ | | $22_2$ | | | | | | | | | | | | |
| +24V | | +24V | J3-1 | | | | | | | | | | | | | | | |
| | | | J2-4 | | | | | | | | | | | | | | | |
| | | | J2-10 | | | | | | | | | | | | | | | |
| CH GND | $M_3$ | CH GND | | | | | | | | | | | | | | | | |
| | $11_3$ | | | | | | | | | | | | | | | | | |
| SPARE | $4_1$ | | | | | | | | | | | | | | | | | |
| ↑ | $7_1$ | | | | | | | | | | | | | | | | | |
| | $8_1$ | | | | | | | | | | | | | | | | | |
| | $9_1$ | | | | | | | | | | | | | | | | | |
| | $10_1$ | | | | | | | | | | | | | | | | | |
| | $11_1$ | | | | | | | | | | | | | | | | | |
| | $12_1$ | | | | | | | | | | | | | | | | | |
| | $13_1$ | | | | | | | | | | | | | | | | | |
| ↓ | $9_3$ | | | | | | | | | | | | | | | | | |
| SPARE | $10_3$ | | | | | | | | | | | | | | | | | |

WANG LABORATORIES, INC.

| | |
|---|---|
| MATERIAL | |
| FINISH | |
| MODEL NO. | 2200 LVP |
| SCALE | SHEET 3 OF 6 |
| BY | DWN A.B |
| DATE | 2/6/80 |
| TITLE | MOTHER BOARD |
| WANG PART NUMBER | 210-7698 |
| SIZE | D |
| DRAWING NUMBER | 7698 |
| REV. | 1 |

D 7698

DO NOT SCALE

| SIGNAL | I/O 1,2,3 | JACK | INTERFACE 7694 | CPU 7696 | DISK C 1645 | REG 6793 | ALU 6792 | STACK 6791 | I.C 6790 | MC 6789 | CM1 7588 | CM2 7588 | DM1 7587 | DM2 7587 | CONN 1 | CONN 2 | CONN 3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R0 | | | | | | $1_3$ | $1_3$ | | | $P_3$ | | | | | | | |
| R1 | | | | | | $22_2$ | $22_2$ | | | $13_1$ | | | | | | | |
| R2 | | | | | | $18_3$ | $18_3$ | | $16_3$ | $16_3$ | | | | | | | |
| R3 | | | | | | $V_3$ | $V_3$ | | $T_3$ | $15_3$ | | | | | | | |
| R4 | | | | | | $S_3$ | $M_3$ | | $Z_2$ | $Y_2$ | | | | | | | |
| R5 | | | | | | $15_3$ | $N_3$ | | $E_3$ | $1_3$ | | | | | | | |
| R6 | | | | | | $21_2$ | $21_2$ | | $1_3$ | $22_2$ | | | | | | | |
| R7 | | | | | | $C_3$ | $C_3$ | | $D_3$ | $A_3$ | | | | | | | |
| R8 | | | | | | $E_3$ | | | $F_3$ | $2_3$ | | | | | | | |
| R9 | | | | | | $18_2$ | | | $20_2$ | $19_2$ | | | | | | | |
| R10 | | | | | | $X_2$ | | | $Y_2$ | $X_2$ | | | | | | | |
| R11 | | | | | | $L_3$ | | | $L_3$ | $N_3$ | | | | | | | |
| R12 | | | | | | | | | $18_3$ | $20_3$ | | | | | | | |
| R13 | | | | | | | | | $19_3$ | $X_3$ | | | | | | | |
| R14 | | | | | | $W_3$ | $W_3$ | | $W_3$ | $W_3$ | | | | | | | |
| R15 | | | | | | $U_3$ | $U_3$ | | $U_3$ | $V_3$ | | | | | | | |
| R16 | | | | | | | $6_3$ | $6_3$ | $6_3$ | $F_3$ | | | | | | | |
| R17 | | | | | | | | | $10_3$ | $M_3$ | | | | | | | |
| R18 | | | | | | | $4_1$ | | $4_1$ | $4_1$ | | | | | | | |
| R19 | | | | | | | $12_3$ | | | $12_3$ | | | | | | | |
| R20 | | | | | | | $5_1$ | | $5_1$ | $5_1$ | | | | | | | |
| R21 | | | | | | | $E_1$ | | $E_1$ | $E_1$ | | | | | | | |
| R22 | | | | | | | $D_1$ | | $D_1$ | $D_1$ | | | | | | | |
| | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | |
| $\overline{SB}$ | | | | | | | | $17_2$ | $19_2$ | | | | | | | | |
| SDF | | | $J_2$ | | $J_2$ | | | | | | | | | | | | |
| $\overline{SELECTED}$ | | | $16_2$ | | | | | | | | | | | | | | 1 |
| SEEK COMP | | | $14_2$ | | | | | | | | | | | | | 8 | |
| SEEK CLK + | | | $M_2$ | | | | | | | | | | | | | | 9 |
| SEEK CLK − | | | $11_2$ | | | | | | | | | | | | | | 10 |
| SERIAL ADM CLK | | | $D_2$ | | $D_2$ | | | | | | | | | | | | |
| SIDE SEL | | | $18_2$ | | | | | | | | | | | | 14 | | |
| SHO | | | | | | $7_2$ | $7_2$ | | | | | | | | | | |
| S0 | | | | | | | | $7_3$ | $7_3$ | | | | | | | | |
| S1 | | | | | | | | $11_3$ | $11_3$ | | | | | | | | |
| S2 | | | | | | | | $M_3$ | $M_3$ | | | | | | | | |
| S3 | | | | | | | | $S_3$ | $S_3$ | | | | | | | | |
| S4 | | | | | | | | $17_3$ | $17_3$ | | | | | | | | |
| S5 | | | | | | | | $V_3$ | $V_3$ | | | | | | | | |
| S6 | | | | | | | | $Z_2$ | $A_3$ | | | | | | | | |
| S7 | | | | | | | | $B_3$ | $C_3$ | | | | | | | | |
| STEP | | | $21_2$ | | | | | | | | | | | | 36 | 36 | |
| $\overline{STOP}$ | | | | | | | $D_3$ | | | | $14_2$ | $14_2$ | | | | | |
| $\overline{SWITCH}$ | | | | | | | | | $22_2$ | | $13_2$ | $13_2$ | | | | | |
| $\overline{SR}$ | | | | | | | $19_2$ | $19_2$ | $21_2$ | | | | | | | | |
| | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | |
| $\overline{TAP}$ | | | | | | | | $H_1$ | $H_1$ | | | | | | | | |
| $\overline{TCMDR}$ | | | | | | $J_1$ | $J_1$ | $J_1$ | | $6_1$ | | | | | | | |
| | | | | | | | | | | | | | | | | | |
| $\overline{T1}$ | | | | | | $14_1$ | $14_1$ | $14_1/12_1$ | $14_1$ | | | | | | | | |
| $\overline{T2}$ | | J1-1 | | | | | 12 | | | | | | | | | | |
| $\overline{T3}$ | | | | | | $N_1$ | $N_1$ | $N_1$ | $N_1$ | $P_1$ | | | | | | | |
| $\overline{T4}$ | | | | | | | $P_1$ | $4_1$ | $P_1$ | $M_1$ | | | | | | | |
| $\overline{T5}$ | | | | | | | $1_2$ | | $1_2$ | | | | | | | | |
| $\overline{T6}$ | | | | | | $F_1$ | $F_1$ | $F_1$ | | $F_1$ | | | | | | | |
| $\overline{T7}$ | | | | | | $9_1$ | $M_1$ | | $12_1$ | | | | | | | | |
| $\overline{T8}$ | | | | | | | $A_2$ | | $A_2$ | | | | | | | | |
| $\overline{T9}$ | | | | | | | $2_2$ | $2_2$ | | | $4_2$ | $4_2$ | | | | | |
| $\overline{T10}$ | | | | | | | $C_2$ | $C_2$ | | | | | | | | | |
| $\overline{T11}$ | | | | | | | $3_2$ | $3_2$ | $13_1$ | | | | | | | | |
| $\overline{T12}$ | | | | | | $13_1$ | $13_1$ | $13_1$ | | | | | | | | | |
| T16 | | | | | | $R_1$ | R | | $R_1$ | | | | | | | | |
| | | | | | | | | | | | | | | | | | |
| $\overline{TSP}$ | | | | | | | | $7_1$ | $7_1$ | | | | | | | | |
| $\overline{TRACK\ 00}$ | | | $13_2$ | | | | | | | | | | | | 42 | 42 | |
| TWO SIDED | | | $R_2$ | | | | | | | | | | | | 10 | | |
| | | | | | | | | | | | | | | | | | |
| $\overline{WAMF}$ | | | $E_2$ | | $E_2$ | | | | | | | | | | | | |
| WRITE DATA | | | $20_2$ | | | | | | | | | | | | 38 | | |
| WRITE GATE | | | $X_2$ | | | | | | | | | | | | 40 | 40 | |
| WRITE PROTECT | | | $P_2$ | | | | | | | | | | | | 44 | 44 | |
| $\overline{WTG}$ | | | $S_2$ | | $S_2$ | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | |
| XPA | | | | | | | | $6_1$ | $6_1$ | | | | | | | | |
| XOP | | | | | | | | $L_1$ | $L_1$ | $12_1$ | | | | | | | |
| | | | | | | | | | | | | | | | | | |
| $\overline{ZNMI}$ | | | $W_3$ | $19_3$ | | | | | | | | | | | | | |
| $\overline{ZRD}$ | | | $B_3$ | $B_3$ | $B_3$ | | | | | | | | | | | | |
| $\overline{ZWR}$ | | | $1_3$ | $1_3$ | $1_3$ | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | |
| 2200 INT | | | $V_3$ | $U_3$ | | | | | | | | | | | | | |
| 1MHZ | | | $U_3$ | $17_3$ | | | | | | | | | | | | | |
| 2X FREQ | | | $3_2$ | | $3_2$ | | | | | | | | | | | | |

SEE SHEET 6

WANG LABORATORIES, INC. LOWELL, MASS. U.S.A.

MODEL NO. 2200 LVP

TITLE MOTHER BOARD

DWN R B 2/6/80

210-7698 D 7698 1

D 7698

| SIGNAL | I/O 1,2,3 | JACK | INTERFACE 1694 | CPU 7696 | DISK.C 7645 | REG 6793 | ALU 6792 | STACK 6791 | I C 6790 | MC 6789 | CM 1 7588 | CM 2 7588 | DM 1 7587 | DM 2 7587 | CONN 1 | CONN 2 | CONN 3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ICC 0 | | | | | | | | $14_3$ | $14_3$ | | | | | | | | |
| ICC 1 | | | | | | | | $15_3$ | $15_3$ | | | | | | | | |
| ICC 2 | | | | | | | | $13_3$ | $13_3$ | | | | | | | | |
| ICC 3 | | | | | | | | $A_3$ | $B_3$ | | | | | | | | |
| ICC 4 | | | | | | | | $R_3$ | $R_3$ | | | | | | | | |
| ICC 5 | | | | | | | | $P_3$ | $P_3$ | | | | | | | | |
| ICC 6 | | | | | | | | $N_3$ | $N_3$ | | | | | | | | |
| ICC 7 | | | | | | | | $12_3$ | $12_3$ | | | | | | | | |
| ICC 8 | | | | | | | | $K_3$ | $K_3$ | | | | | | | | |
| ICC 9 | | | | | | | | $9_3$ | $9_3$ | | | | | | | | |
| ICC 10 | | | | | | | | $J_3$ | $J_3$ | | | | | | | | |
| ICC 11 | | | | | | | | $8_3$ | $8_3$ | | | | | | | | |
| ICC 12 | | | | | | | | $5_3$ | $5_3$ | | | | | | | | |
| ICC 13 | | | | | | | | $4_3$ | $4_3$ | | | | | | | | |
| ICC 14 | | | | | | | | $3_3$ | $3_3$ | | | | | | | | |
| ICC 15 | | | | | | | | $2_3$ | $2_3$ | | | | | | | | |
| L8K | | | | | | | | $H_2$ | | | | | $L_2$ | $L_2$ | | | |
| $\overline{\text{LLPC}}$ | | | | | | | | $M_1$ | $M_1$ | | | | | | | | |
| $\overline{\text{LHPC}}$ | | | | | | | | $11_1$ | $11_1$ | | | | | | | | |
| $\overline{\text{LPI}}$ | | | | | | | $9_1$ | $9_1$ | $9_1$ | | | | | | | | |
| LOP | | | | | | $T_2$ | | $T_2$ | $U_2$ | $T_2$ | | | | | | | |
| $\overline{\text{MAR CLK}}$ | | | | | | | $F_3$ | | $H_3$ | $14_3$ | | | $R_3$ | $R_3$ | | | |
| MFM WRITE DATA | | | $L_2$ | | $L_2$ | | | | | | | | | | | | |
| $\overline{\text{MM}}$ | | | | | | | | | | | $E_2$ | $E_2$ | | | | | |
| $\overline{\text{MS1}}$ | | | | | | | | | $E_2$ | | $18_2$ | | | | | | |
| $\overline{\text{MS2}}$ | | | | | | | | | $D_2$ | | $T_2$ | | | | | | |
| $\overline{\text{MS3}}$ | | | | | | | | | $C_2$ | | $S_2$ | $T_2$ | | | | | |
| $\overline{\text{MS4}}$ | | | | | | | | | $17_2$ | | $16_2$ | $18_2$ | | | | | |
| $\overline{\text{MS5}}$ | | | | | | | | | $T_2$ | | $17_2$ | | | | | | |
| $\overline{\text{MS6}}$ | | | | | | | | | $16_2$ | | $15_2$ | $17_2$ | | | | | |
| $\overline{\text{MS7}}$ | | | | | | | | | $X_3$ | | $R_2$ | | | | | | |
| $\overline{\text{MS8}}$ | | | | | | | | | $20_3$ | | $N_2$ | | | | | | |
| $\overline{\text{OBS}}$ | $H_1$ | | $H_1$ | | | $H_1$ | | | | | | | | | | | |
| $\overline{\text{OB1}}$ | $A_3$ | | $K_3$ | | | $6_3$ | | | | | $1_3$ | $1_3$ | | | | | |
| $\overline{\text{OB2}}$ | $B_3$ | | $L_3$ | | | $11_3$ | | | | | $Z_2$ | $Z_2$ | | | | | |
| $\overline{\text{OB3}}$ | $C_3$ | | $M_3$ | | | $M_3$ | | | | | $R_1$ | $R_1$ | | | | | |
| $\overline{\text{OB4}}$ | $D_3$ | | $N_3$ | | | $12_3$ | | | | | $12_1$ | $12_1$ | | | | | |
| $\overline{\text{OB5}}$ | $J_3$ | | $20_3$ | | | $14_3$ | | | | | $11_1$ | $11_1$ | | | | | |
| $\overline{\text{OB6}}$ | $H_3$ | | $17_3$ | | | $13_3$ | | | | | $6_1$ | $6_1$ | | | | | |
| $\overline{\text{OB7}}$ | $F_3$ | | $T_3$ | | | $P_3$ | | | | | $5_1$ | $5_1$ | | | | | |
| $\overline{\text{OB8}}$ | $E_3$ | | $P_3$ | | | $N_3$ | | | | | $4_1$ | $4_1$ | | | | | |
| PARITY ERROR | | | $S_3$ | $S_3$ | | | | | | | | | | | | | |
| $\overline{\text{PAØ}}$ | | | | | | $H_3$ | | $8_2$ | | | | | $M_2$ | $M_2$ | | | |
| $\overline{\text{PA1}}$ | | | | | | $R_3$ | | $1_3$ | | | | | $N_2$ | $N_2$ | | | |
| PEDM | | | | | | $3_2$ | | | $3_2$ | $J_2$ | | | | | | | |
| PECM | | | | | | | | $16_2$ | $18_2$ | $17_2$ | | | | | | | |
| $\overline{\text{PCCC}}$ | | | | | | | | $2$ | $G_1$ | | | | | | | | |
| $\overline{\text{PINC}}$ | | | | | | | | $K_1$ | $K_1$ | | | | | | | | |
| POR | | J1-7 | $T_2$ | $J_3$ | $T_2$ | $V_2$ | | $V_2$ | $W_2$ | | $D_3$ | $D_3$ | | | | | |
| $\overline{\text{PRMS}}$ | $3_3$ | J1-5 | $11_3$ | | | | | | | | $E_3$ | $E_3$ | | | | | |
| $\overline{\text{PORT 0}}$ | | | $10_1$ | $9_1$ | | | | | | | | | | | | | |
| $\overline{\text{PORT 2}}$ | | | $5_1$ | $6_1$ | | | | | | | | | | | | | |
| $\overline{\text{PORT 3}}$ | | | $A_2$ | $F_1$ | $A_2$ | | | | | | | | | | | | |
| $\overline{\text{PORT 4}}$ | | | | $H_1$ | $H_1$ | | | | | | | | | | | | |
| $\overline{\text{PORT 5}}$ | | | $F_1$ | $7_1$ | | | | | | | | | | | | | |
| $\overline{\text{PORT 6}}$ | | | $J_1$ | $10_1$ | | | | | | | | | | | | | |
| $\overline{\text{PORT 60}}$ | | | | $D_1$ | $D_1$ | | | | | | | | | | | | |
| PH 0 | | | | | | | $20_3$ | $19_3$ | | | $M_1$ | $M_1$ | $9_3$ | $9_3$ | | | |
| PH 1 | | | | | | | $3_3$ | $22_2$ | | | $F_1$ | $F_1$ | $8_3$ | $8_3$ | | | |
| PH 2 | | | | | | | $X_3$ | $W_3$ | | | $E_1$ | $E_1$ | $K_3$ | $K_3$ | | | |
| PH 3 | | | | | | | $E_3$ | $18_3$ | | | $D_1$ | $D_1$ | $J_3$ | $J_3$ | | | |
| PH 4 | | | | | | | $2_3$ | $W_2$ | | | $T_3$ | $T_3$ | $F_3$ | $F_3$ | | | |
| PH 5 | | | | | | | $14_2$ | $11_2$ | | | $S_3$ | $S_3$ | $C_3$ | $C_3$ | | | |
| PH 6 | | | | | | | $Z_2$ | $R_2$ | | | $R_3$ | $R_3$ | $5_3$ | $5_3$ | | | |
| PH 7 | | | | | | | $L_2$ | $M_2$ | | | $2_3$ | $2_3$ | $K_2$ | $K_2$ | | | |
| PL0 | | | | | | | $T_3$ | $T_3$ | | $R_3$ | $16_3$ | $16_3$ | | | | | |
| PL1 | | | | | | | $V_2$ | $S_2$ | | | $15_3$ | $15_3$ | $6_3$ | $6_3$ | | | |
| PL2 | | | | | | | $15_2$ | $12_2$ | | | $14_3$ | $14_3$ | $D_3$ | $D_3$ | | | |
| PL3 | | | | | | | $N_2$ | $9_2$ | | | $B_3$ | $B_3$ | $2_3$ | $2_3$ | | | |
| PL4 | | | | | | | $F_2$ | $1_2$ | | | $A_3$ | $A_3$ | $14_1$ | $14_1$ | | | |
| PL5 | | | | | | | $J_2$ | $7_2$ | | | $22_2$ | $22_2$ | $A_3$ | $A_3$ | | | |
| PL6 | | | | | | | $P_2$ | $10_2$ | | | $P_1$ | $P_1$ | $B_3$ | $B_3$ | | | |
| PL7 | | | | | | | $A_3$ | $X_2$ | | | $N_1$ | $N_1$ | $H_3$ | $H_3$ | | | |
| READY | | | $1_2$ | $K_1$ | | | | | | | | | | | | | |
| $\overline{\text{READY/BUSY}}$ | $K_3$ | | $X_3$ | | | $X_3$ | | | | | | | | | | | |
| READ DATA | | | $6_2$ | | $6_2$ | | | | | | | | | | | | |
| $\overline{\text{REF}}$ | | | | | | | $B_2$ | | $B_2$ | | $Y_2$ | $Y_2$ | $L_3$ | $L_3$ | | | |
| $\overline{\text{RESET}}$ | | J1-3 | | | | | | | $S_2$ | | | | | | | | |
| REF CLK | | | | | | | $13_2$ | | $15_2$ | | | | | | | | |
| $\overline{\text{RDG}}$ | | | $7_2$ | | $7_2$ | | | | | | | | | | | | |
| $\overline{\text{ROMS}}$ | | | | | | | $11_1$ | | $X_2$ | $W_2$ | | | | | | | |
| R/$\overline{\text{W}}$ | | | | | | | | | $F_1$ | | $K_2$ | $K_2$ | | | | | |
| RWCLK | | | $4_2$ | | $4_2$ | | | | | | | | | | | | |
| RA0 | | | | | | | | | $J_2$ | | | | $E_3$ | $E_3$ | | | |
| RA1 | | | | | | | | | $K_2$ | | | | $7_3$ | $7_3$ | | | |
| RA2 | | | | | | | | | $8_2$ | | | | $3_3$ | $3_3$ | | | |
| RA3 | | | | | | | | | $10_2$ | | | | $R_1$ | $R_1$ | | | |
| RA4 | | | | | | | | | $14_2$ | | | | $1_3$ | $1_3$ | | | |
| RA5 | | | | | | | | | $R_2$ | | | | $4_3$ | $4_3$ | | | |

SEE SHEET 6

WANG LABORATORIES INC.

2200 LVP

MOTHER BOARD

D 7698

210-7598

D 7698

SEE SHEET 6

| SIGNAL | I/O 1.2.3 | JACK | INTERFACE 7694 | CPU 7696 | DISK. C 7695 | REG 6793 | ALU 6792 | STACK 6791 | I.C 6790 | MC 6789 | CM 1 7588 | CM 2 7588 | DM 1 7587 | DM 2 7587 | CONN1 | CONN 2 | CONN 3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DACK 0 | | | | $P_1$ | $P_1$ | | | | | | | | | | | | |
| DACK 1 | | | $P_1$ | $14_1$ | | | | | | | | | | | | | |
| DACK 2 | | | $N_1$ | $13_1$ | | | | | | | | | | | | | |
| DISK RDY | | | $14_1$ | | | | | | | | | | | | 22 | 22 | |
| D0 | | | $5_3$ | $5_3$ | $5_3$ | | | | | | | | | | | | |
| D1 | | | $F_3$ | $F_3$ | $F_3$ | | | | | | | | | | | | |
| D2 | | | $4_3$ | $4_3$ | $4_3$ | | | | | | | | | | | | |
| D3 | | | $C_3$ | $C_3$ | $C_3$ | | | | | | | | | | | | |
| D4 | | | $2_3$ | $2_3$ | $2_3$ | | | | | | | | | | | | |
| D5 | | | $3_3$ | $3_3$ | $3_3$ | | | | | | | | | | | | |
| D6 | | | $D_3$ | $D_3$ | $D_3$ | | | | | | | | | | | | |
| D7 | | | $E_3$ | $E_3$ | $E_3$ | | | | | | | | | | | | |
| DISK CHANGE | | | $15_2$ | | | | | | | | | | | | 12 | | |
| DISK DONE | | | | $18_3$ | $18_2$ | | | | | | | | | | | | |
| DIRECTN | | | $Y_2$ | | | | | | | | | | | | 34 | 34 | |
| DIF | | | | | | | $13_3$ | | | | | | $D_1$ | $D_1$ | | | |
| | | | | | | | | | | | | | $13_3$ | $13_3$ | | | |
| DI0 | | | | | | | $16_3$ | | | | | | $13_1$ | $13_1$ | | | |
| | | | | | | | | | | | | | $16_3$ | $16_3$ | | | |
| DI1 | | | | | | | $17_3$ | | | | | | $P_1$ | $P_1$ | | | |
| | | | | | | | | | | | | | $17_3$ | $17_3$ | | | |
| DI2 | | | | | | | $R_3$ | | | | | | $12_1$ | $12_1$ | | | |
| | | | | | | | | | | | | | $S_3$ | $S_3$ | | | |
| DI3 | | | | | | | $S_3$ | | | | | | $6_1$ | $6_1$ | | | |
| | | | | | | | | | | | | | $T_3$ | $T_3$ | | | |
| DI4 | | | | | | | $19_3$ | | | | | | $5_1$ | $5_1$ | | | |
| | | | | | | | | | | | | | $U_3$ | $U_3$ | | | |
| DI5 | | | | | | | $15_3$ | | | | | | $E_1$ | $E_1$ | | | |
| | | | | | | | | | | | | | $15_3$ | $15_3$ | | | |
| DI6 | | | | | | | $14_3$ | | | | | | $F_1$ | $F_1$ | | | |
| | | | | | | | | | | | | | $14_3$ | $14_3$ | | | |
| DI7 | | | | | | | $P_3$ | | | | | | $4_1$ | $4_1$ | | | |
| | | | | | | | | | | | | | $P_3$ | $P_3$ | | | |
| DMFM WRITE DATA+ | | | $10_2$ | | | | | | | | | | | | | | 13 |
| DMFM WRITE DATA- | | | $K_2$ | | | | | | | | | | | | | | 14 |
| DMFM READ DATA+ | | | $N_2$ | | | | | | | | | | | | | | 11 |
| DMFM READ DATA- | | | $12_2$ | | | | | | | | | | | | | | 18 |
| $\overline{\text{DMS1}}$ | | | | | | | | $20_3$ | | | | | $10_3$ | | | | |
| $\overline{\text{DMS2}}$ | | | | | | | | $X_3$ | | | | | | $10_3$ | | | |
| $\overline{\text{DMPI}}$ | | | | | | $2_2$ | | | $2_2$ | | | | | | | | |
| | | | | | | | | | | | | | | | | | |
| DMO Ø | | | | | | | | | | $T_3$ | | | $19_3$ | $19_3$ | | | |
| DMO 1 | | | | | | | | | | $U_3$ | | | $20_3$ | $20_3$ | | | |
| DMO 2 | | | | | | | | | | $17_3$ | | | $V_3$ | $V_3$ | | | |
| DMO 3 | | | | | | | | | | $18_3$ | | | $W_3$ | $W_3$ | | | |
| DMO 4 | | | | | | | | | | $S_3$ | | | $18_3$ | $18_3$ | | | |
| DMO 5 | | | | | | | | | | $19_3$ | | | $X_3$ | $X_3$ | | | |
| DMO 6 | | | | | | | | | | $9_3$ | | | $11_3$ | $11_3$ | | | |
| DMO 7 | | | | | | | | | | $H_3$ | | | $M_3$ | $M_3$ | | | |
| DMO 8 | | | | | | | | | | $10_3$ | | | $12_3$ | $12_3$ | | | |
| DMO 9 | | | | | | | | | | $11_1$ | | | $M_1$ | $M_1$ | | | |
| DMO 10 | | | | | | | | | | $10_1$ | | | $L_1$ | $L_1$ | | | |
| DMO 11 | | | | | | | | | | $L_1$ | | | $10_1$ | $10_1$ | | | |
| DMO 12 | | | | | | | | | | $7_1$ | | | $H_1$ | $H_1$ | | | |
| DMO 13 | | | | | | | | | | $J_1$ | | | $8_1$ | $8_1$ | | | |
| DMO 14 | | | | | | | | | | $8_1$ | | | $J_1$ | $J_1$ | | | |
| DMO 15 | | | | | | | | | | $9_1$ | | | $K_1$ | $K_1$ | | | |
| DMO 16 | | | | | | | | | | $H_1$ | | | $7_1$ | $7_1$ | | | |
| DMO 17 | | | | | | | | | | $K_1$ | | | $9_1$ | $9_1$ | | | |
| | | | | | | | | | | | | | | | | | |
| $\overline{\text{DREQ 0}}$ | | | | $R_1$ | $R_1$ | | | | | | | | | | | | |
| DREQ 1 | | | $9_1$ | $11_1$ | | | | | | | | | | | | | |
| DREQ 2 | | | $M_1$ | $12_1$ | | | | | | | | | | | | | |
| DRIVE SEL 1 | | | $19_2$ | | | | | | | | | | | | 26 | 26 | |
| DRIVE SEL 2 | | | $W_2$ | | | | | | | | | | | | 28 | 28 | |
| D-$\overline{\text{R}}$/W2 | | | | | | | | | | $12_2$ | | | $N_1$ | $N_1$ | | | |
| D-$\overline{\text{R}}$/W1 | | | | | | | | | | $B_3$ | | | $N_3$ | $N_3$ | | | |
| | | | | | | | | | | | | | | | | | |
| $\overline{\text{EOP}}$ | | | $A_3$ | $A_3$ | $A_3$ | | | | | | | | | | | | |
| EWC | | | $8_2$ | | $8_2$ | | | | | | | | | | 2 | 2 | |
| | | | | | | | | | | | | | | | | | |
| $\overline{\text{FB128}}$ | | | | | | | | | | | | | $14_2$ | $R_2$ | | | |
| FLOPPY | | | $9_2$ | | $9_2$ | | | | | | | | | | | | |
| $\overline{\text{F READ DATA}}$ | | | $K_1$ | | | | | | | | | | | | 46 | | |
| | | | | | | | | | | | | | | | | | |
| $\overline{\text{HALT}}$ | $K_1$ | | | | | $K_1$ | | | | | | | | | | | |
| HEAD LOAD | | | $V_2$ | | | | | | | | | | | | 18 | | |
| HEAD SEL 2' | | | $17_2$ | | | | | | | | | | | | | 18 | |
| HEAD SEL 2° | | | $U_2$ | | | | | | | | | | | | | 14 | |
| | | | | | | | | | | | | | | | | | |
| $\overline{\text{IBS}}$ | $1_3$ | | $9_3$ | | | $F_3$ | | | | | | | | | | | |
| I/O CLK | $L_3$ | | | | | | $1_3$ | | | | | | | | | | |
| $\overline{\text{INDEX}}$ | | | $S_2$ | | $S_2$ | | | | | | | | | | 20 | 20 | |
| | | | | | | | | | | | | | | | | | |
| $\overline{\text{IB1}}$ | $15_1$ | | $13_1$ | | | $P_1$ | | | | | | | | | | | |
| $\overline{\text{IB2}}$ | $S_1$ | | $R_1$ | | | $B_2$ | | | | | | | | | | | |
| $\overline{\text{IB3}}$ | $14_1$ | | $C_2$ | | | $4_2$ | | | | | | | | | | | |
| $\overline{\text{IB4}}$ | $R_1$ | | $12_1$ | | | $12_1$ | | | | | | | | | | | |
| $\overline{\text{IB5}}$ | $N_1$ | | $2_2$ | | | $C_2$ | | | | | | | | | | | |
| $\overline{\text{IB6}}$ | $P_1$ | | $11_1$ | | | $11_1$ | | | | | | | | | | | |
| $\overline{\text{IB7}}$ | $L_1$ | | $B_2$ | | | $A_2$ | | | | | | | | | | | |
| $\overline{\text{IB8}}$ | $M_1$ | | $L_1$ | | | $M_1$ | | | | | | | | | | | |
| $\overline{\text{IB9}}$ | $2_3$ | | $10_3$ | | | $10_1$ | | | | | | | | | | | |

WANG LABORATORIES, INC. LOWELL, MASS. U.S.A.

| | |
|---|---|
| MATERIAL | |
| FINISH | |
| MODEL NO. | 2200 LVP |
| | SEE ENGRG SPECIFICATIONS |
| TOL | EE AS NOTED |
| SCALE | |
| SHT | 2 OF 6 |
| DWN | AJ |
| DATE | 2/4/80 |
| CHK | |
| E ENGR | |
| M ENGR | |
| MFG ENGR | |
| APPROVED BY | |
| TITLE | MOTHER BOARD |
| WANG PART NUMBER | 210-7698 |
| SIZE | D |
| DRAWING NUMBER | 7698 |
| REV | 1 |

D 7698

SEE SHEET 6

| SIGNAL | I/O 1,2,3 | JACK | INTERFACE 7694 | CPU 7696 | DISK C 7695 | REG 6793 | ALU 6792 | STACK 6791 | IC 6790 | MC 6789 | CM 1 7588 | CM 2 7588 | DM 1 7587 | DM 2 7587 | COMM 1 | COMM 2 | COMM 3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AØ | | | | $T_3$ | $T_3$ | | | | | | | | | | | | |
| A1 | | | | $15_3$ | $15_3$ | | | | | | | | | | | | |
| A2 | | | | $P_3$ | $P_3$ | | | | | | | | | | | | |
| A3 | | | | $R_3$ | $R_3$ | | | | | | | | | | | | |
| $\overline{AB1}$ | $D_1$ | | $D_1$ | | | $D_1$ | | | | | | | | | | | |
| $\overline{AB2}$ | $5_1$ | | $4_1$ | | | $4_1$ | | | | | | | | | | | |
| $\overline{AB3}$ | $6_1$ | | $6_1$ | | | $6_1$ | | | | | | | | | | | |
| $\overline{AB4}$ | $E_1$ | | $E_1$ | | | $E_1$ | | | | | | | | | | | |
| $\overline{AB5}$ | $F_1$ | | $7_1$ | | | $7_1$ | | | | | | | | | | | |
| $\overline{AB6}$ | $6_3$ | | $14_3$ | | | $T_3$ | | | | | | | | | | | |
| $\overline{AB7}$ | $7_3$ | | $16_3$ | | | $16_3$ | | | | | | | | | | | |
| $\overline{AB8}$ | $8_3$ | | $19_3$ | | | $17_3$ | | | | | | | | | | | |
| $\overline{ABS}$ | $J_1$ | | $8_1$ | | | $8_1$ | | | | | | | | | | | |
| ABUS 0 | | | | | | $B_3$ | $Y_2$ | | | $S_2$ | | | | | | | |
| ABUS 1 | | | | | | $3_3$ | $20_2$ | | | $14_2$ | | | | | | | |
| ABUS 2 | | | | | | $A_3$ | $X_2$ | | | $R_2$ | | | | | | | |
| ABUS 3 | | | | | | $2_3$ | $B_3$ | | | $13_2$ | | | | | | | |
| ABUS 4 | | | | | | $Z_2$ | $18_2$ | | | $P_2$ | | | | | | | |
| ABUS 5 | | | | | | $20_2$ | $17_2$ | | | $M_2$ | | | | | | | |
| ABUS 6 | | | | | | $U_2$ | $U_2$ | | | $K_2$ | | | | | | | |
| ABUS 7 | | | | | | $D_2$ | $6_2$ | | | $14_1$ | | | | | | | |
| ADDEQU | | | $F_2$ | | $F_8$ | | | | | | | | | | | | |
| $\overline{ALUCLK}$ | | | | | | $W_2$ | $W_2$ | | | | | | | | | | |
| $\overline{AMF}$ | | | $H_2$ | | $H_2$ | | | | | | | | | | | | |
| $.\overline{B}$ | | | | | | | | $U_2$ | $V_2$ | | | | | | | | |
| $\overline{BLKCA}$ | | | | | | $10_2$ | $10_2$ | | | | | | | | | | |
| BRANCH | | | | | | | $5_2$ | | $9_2$ | | | | | | | | |
| BO | | | | | | $10_3$ | $L_3$ | | | $Z_2$ | | | | | | | |
| B1 | | | | | | $K_3$ | $K_3$ | | | $21_2$ | | | | | | | |
| B2 | | | | | | $9_3$ | $9_3$ | | | $20_2$ | | | | | | | |
| B3 | | | | | | $J_3$ | $J_3$ | | | $18_2$ | | | | | | | |
| B4 | | | | | | $8_3$ | $8_3$ | | | $V_2$ | | | | | | | |
| B5 | | | | | | $7_3$ | $7_3$ | | | $U_2$ | | | | | | | |
| B6 | | | | | | $D_3$ | $5_3$ | | | $16_2$ | | | | | | | |
| B7 | | | | | | $4_3$ | $4_3$ | | | $15_2$ | | | | | | | |
| CA | | | | | | $8_2$ | $8_2$ | | | | | | | | | | |
| $\overline{CBS}$ | $5_3$ | | $13_3$ | | | $20_3$ | | | | | | | | | | | |
| CEN | | | | | | $9_2$ | $9_2$ | | | | | | | | | | |
| $\overline{CE}$ | | | | | | | $T_2$ | | | | $U_2$ | $U_2$ | $11_1$ | $11_1$ | | | |
| $\overline{CIO}$ | | | | | | $5_1$ | | $5_1$ | | | | | | | | | |
| $\overline{CNTD}$ | | | | | | | | $10_1$ | $10_1$ | | | | | | | | |
| $\overline{CPB}$ | $4_3$ | | $12_3$ | | | $19_3$ | | | | | | | | | | | |
| CO | | | | | | $16_2$ | $16_2$ | | | | | | | | | | |
| C1 | | | | | | $S_2$ | $S_2$ | | | | | | | | | | |
| C2 | | | | | | $R_2$ | $R_2$ | | | | | | | | | | |
| C3 | | | | | | $12_2$ | $12_2$ | | | | | | | | | | |
| C4 | | | | | | $11_2$ | $11_2$ | | | | | | | | | | |
| C5 | | | | | | $M_2$ | $M_2$ | | | | | | | | | | |
| C6 | | | | | | $K_2$ | $K_2$ | | | | | | | | | | |
| C7 | | | | | | $H_2$ | $H_2$ | | | | | | | | | | |
| CA 0 | | | | | | | | | $6_2$ | $7_2$ | $10_2$ | $10_2$ | | | | | |
| CA 1 | | | | | | | | | $H_2$ | $H_2$ | $P_2$ | $P_2$ | | | | | |
| CA2 | | | | | | | | | $5_2$ | $6_2$ | $L_2$ | $L_2$ | | | | | |
| CA3 | | | | | | | | | $7_2$ | $9_2$ | $11_2$ | $11_2$ | | | | | |
| CA4 | | | | | | | | | $F_2$ | $F_2$ | $M_2$ | $M_2$ | | | | | |
| CA5 | | | | | | | | | $4_2$ | $5_2$ | $9_2$ | $9_2$ | | | | | |
| CA6 | | | | | | | | | $12_2$ | $11_2$ | $20_2$ | $20_2$ | $J_2$ | $J_2$ | | | |
| CA7 | | | | | | | | | $M_2$ | $L_2$ | $V_2$ | $V_2$ | | | | | |
| CA8 | | | | | | | | | $P_2$ | $N_2$ | $X_2$ | $X_2$ | | | | | |
| CA9 | | | | | | | | | $11_2$ | $10_2$ | $19_2$ | $19_2$ | | | | | |
| CA10 | | | | | | | | | $13_2$ | | $21_2$ | $21_2$ | | | | | |
| CA11 | | | | | | | | | $N_2$ | | $W_2$ | $W_2$ | | | | | |
| CBUS 0 | | | | | | $17_2$ | | $18_2$ | | | | | | | | | |
| CBUS 1 | | | | | | $15_2$ | | $15_2$ | | | | | | | | | |
| CBUS 2 | | | | | | $14_2$ | | $14_2$ | | | | | | | | | |
| CBUS 3 | | | | | | $13_2$ | | $13_2$ | | | | | | | | | |
| CBUS 4 | | | | | | $P_2$ | | $P_2$ | | | | | | | | | |
| CBUS 5 | | | | | | $N_2$ | | $N_2$ | | | | | | | | | |
| CBUS 6 | | | | | | $L_2$ | | $L_2$ | | | | | | | | | |
| CBUS 7 | | | | | | $J_2$ | | $J_2$ | | | | | | | | | |
| CSO Ø | | | | | | | | $L_3$ | | $L_3$ | $20_3$ | $20_3$ | | | | | |
| CSO 1 | | | | | | | | $H_3$ | | $K_3$ | $19_3$ | $19_3$ | | | | | |
| CSO 2 | | | | | | | | $F_3$ | | $J_3$ | $18_3$ | $18_3$ | | | | | |
| CSO 3 | | | | | | | | $E_3$ | | $E_3$ | $M_3$ | $M_3$ | | | | | |
| CSO 4 | | | | | | | | $D_3$ | | $D_3$ | $L_3$ | $L_3$ | | | | | |
| CSO 5 | | | | | | | | $C_3$ | | $3_3$ | $10_3$ | $10_3$ | | | | | |
| CSO 6 | | | | | | | | $D_2$ | | $E_2$ | $C_2$ | $C_2$ | | | | | |
| CSO 7 | | | | | | | | $4_2$ | | $D_2$ | $B_2$ | $B_2$ | | | | | |
| CSO 8 | | | | | | | | $B_2$ | | $C_2$ | $A_2$ | $A_2$ | | | | | |
| CSO 9 | | | | | | | | $A_2$ | | $1_2$ | $10_1$ | $10_1$ | | | | | |
| CSO 10 | | | | | | | | $R_1$ | | $A_2$ | $K_1$ | $K_1$ | | | | | |
| CSO 11 | | | | | | | | $P_1$ | | $N_1$ | $J_1$ | $J_1$ | | | | | |
| CSO 12 | | | | | | | | $16_3$ | | $11_3$ | $X_3$ | $X_3$ | | | | | |
| CSO 13 | | | | | | | | $10_3$ | | $7_3$ | $W_3$ | $W_3$ | | | | | |
| CSO 14 | | | | | | | | $21_2$ | | $6_3$ | $V_3$ | $V_3$ | | | | | |
| CSO 15 | | | | | | | | $20_2$ | | $5_3$ | $N_3$ | $N_3$ | | | | | |
| CSO 16 | | | | | | $5_3$ | | | | $4_3$ | $11_3$ | $11_3$ | | | | | |
| CSO 17 | | | | | | $Y_2$ | | $Y_2$ | | $C_3$ | $K_3$ | $K_3$ | | | | | |
| CSO 18 | | | | | | $L_1$ | | $E_1$ | | $4_2$ | $3_2$ | $3_2$ | | | | | |
| CSO 19 | | | | | | $F_2$ | | $F_2$ | | $3_2$ | $2_2$ | $2_2$ | | | | | |
| CSO 20 | | | | | | $6_2$ | | $6_2$ | | $2_2$ | $1_2$ | $1_2$ | | | | | |
| CSO 21 | | | | | | $E_2$ | | $E_2$ | | $B_2$ | $L_1$ | $L_1$ | | | | | |
| CSO 22 | | | | | | $5_2$ | | $5_2$ | | $R_1$ | $9_1$ | $9_1$ | | | | | |
| CSO 23 | | | | | | $1_2$ | | | | $13_1$ | $8_1$ | $8_1$ | | | | | |

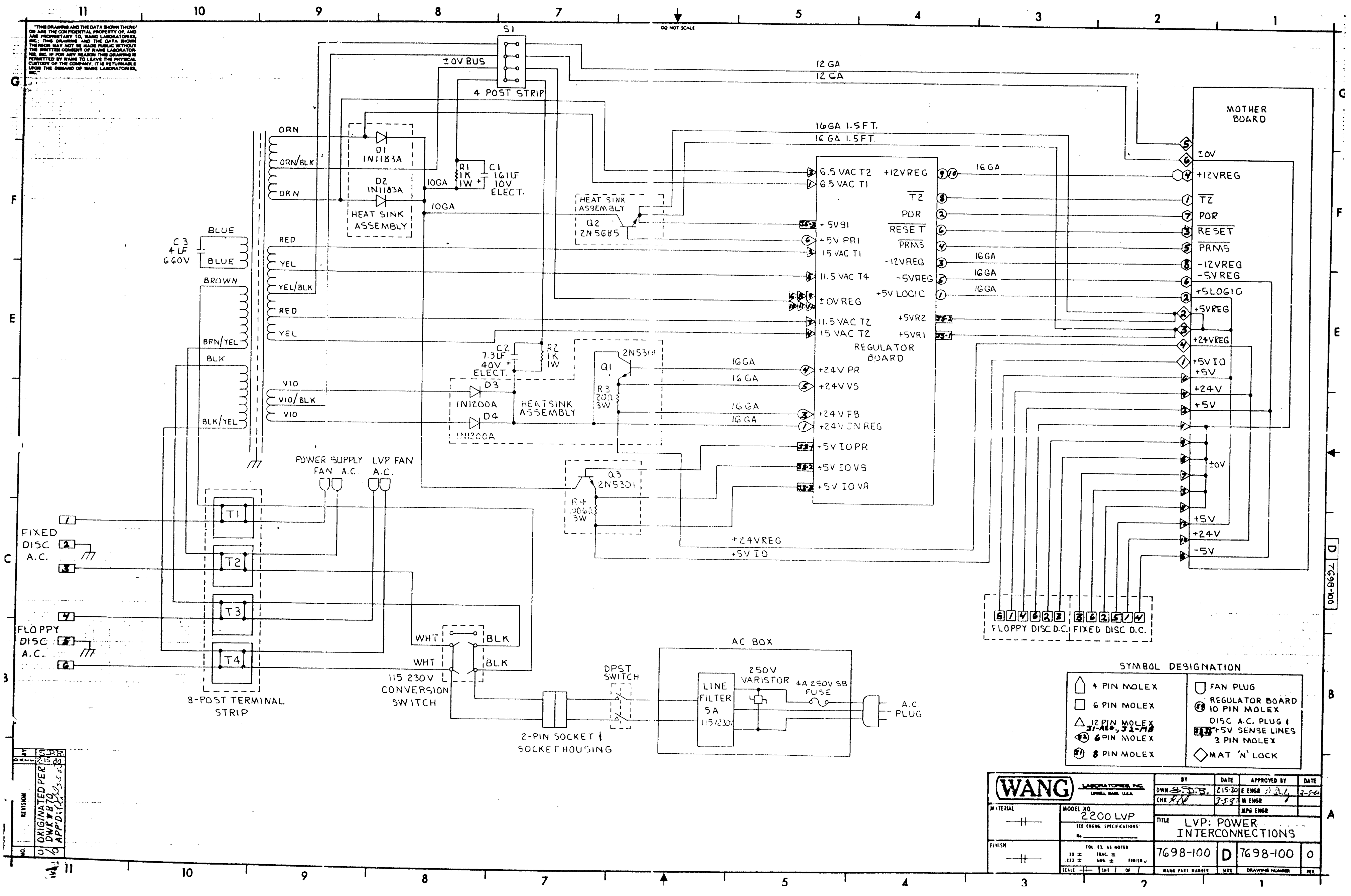
WANG LABORATORIES, INC.

MODEL NO. 2200 LVP

| DWN | AB | 2/6/80 |
|---|---|---|

TITLE MOTHER BOARD

210-7698 D 7698



SYMBOL DESIGNATION

| | |
|---|---|
| △ 4 PIN MOLEX | FAN PLUG |
| □ 6 PIN MOLEX | REGULATOR BOARD 10 PIN MOLEX |
| △ 12 PIN MOLEX J1-A&B, J2-A&B | DISC A.C. PLUG & +5V SENSE LINES 3 PIN MOLEX |
| 6 PIN MOLEX | ◇ MAT 'N' LOCK |
| 8 PIN MOLEX | |

"THIS DRAWING AND THE DATA SHOWN THEREON ARE THE CONFIDENTIAL PROPERTY OF, AND ARE PROPRIETARY TO, WANG LABORATORIES, INC.; THIS DRAWING AND THE DATA SHOWN THEREON MAY NOT BE MADE PUBLIC WITHOUT THE WRITTEN CONSENT OF WANG LABORATORIES, INC. IF FOR ANY REASON THIS DRAWING IS PERMITTED BY WANG TO LEAVE THE PHYSICAL CUSTODY OF THE COMPANY, IT IS RETURNABLE UPON THE DEMAND OF WANG LABORATORIES, INC."

REVISION: 0 ORIGINATED PER DWR #870 APP'D [illegible] 2-15-80

WANG LABORATORIES, INC. LOWELL, MASS. U.S.A.

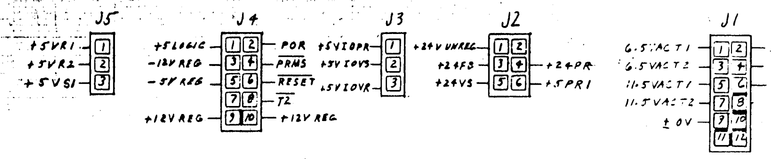
MODEL NO. 2200 LVP

TITLE LVP: POWER INTERCONNECTIONS

DWN S.D.B. 2-15-80; CHK [illegible] 3-5-80

7698-100 D 7698-100 0

DO NOT SCALE



| I.C. LOCATION | W.L. PART No. | TYPE |
|---|---|---|
| L1, 2, 4, 6 | 376-0066 | 723 |
| L3, 8 | 376-0240 | LM339 |
| L5 | 376-0074 | LM741C |
| L7 | 376-0134 | LM304 |
| L9 | 376-0006 | 7474 |
| L10 | 376-0028 | 7403 |

| I.C. TYPE | LOCATION | SPARES |
|---|---|---|
| LM339 | L3 | 1 |
| 7403 | L10 | 1 |
| 7474 | L9 | 1 |

| COMPONENT | W.L. PART No. | TYPE |
|---|---|---|
| D1-8 | 380-3008 | A15A RECTIFIER |
| D9, 10, 11, 14, 15, 17, 18, 19 | 380-1001 | SIL DIODE |
| D12 | 380-2056 | 1N752A ZEN 5.6V |
| D13 | 380-2129 | 1N4372A ZEN 3V |
| D16 | 380-2111 | 1N5241C ZEN 11V |
| D20 | 380-2113 | 1N4743 ZEN 13V |
| D21, 23 | 380-2048 | 1N5230C ZEN 4.7V |
| D22 | 380-0000 | GER DIODE |
| D24 | 380-4000 | 1N4004 |
| | | |
| | | |
| Q1, 2, 3 | 375-1053 | RCA8203A |
| Q4, 5 | 375-1052 | 2N6387 |
| Q6 | 374-0005 | MA1112 |
| Q7 | 375-1017 | 2N2906A |
| | | |
| | | |
| J1 | 654-1172 | 12 POS. HEADER |
| J2 | 654-1186 | 6 POS. HEADER |
| J3, 5 | 654-1193 | 3 POS HEADER |
| J4 | 654-1183 | 10 POS HEADER |

| COMPONENT | W.L. PART No. | TYPE |
|---|---|---|
| R1, 13, 24, 43 | 336-1001 | 1K POT |
| R2 | 330-2022 | 220Ω 1/4W 10% |
| R3, 14, 16, 40, 53 | 330-3015 | 1.5K 1/4W 10% |
| R4, 5, 18 | 334-0032 | .02Ω 3W 3% |
| R6, 19 | 334-0031 | .006Ω 3W 5% |
| R7, 57, 60 | 330-2010 | 100Ω 1/4W 10% |
| R8 | 337-2039 | 390Ω 2W 10% |
| R9 | 330-3033 | 3.3K 1/4W 10% |
| R10, 11, 12, 23 | 331-3010 | 1K 1/2W 10% |
| R15, 17, 42, 50 | 330-2068 | 680Ω 1/4W 10% |
| | | |
| | | |
| R20, 49, 64, 66, 67, 68 | 330-3010 | 1K 1/4W 10% |
| R21, 34, 36, 37, 41 | 330-5012 | 120K 1/4W 10% |
| R22, 29, 38, 52, 65 | 330-3022 | 2.2K 1/4W 10% |
| R25 | 330-3083 | 8.2K 1/4W 5% |
| R26, 47 | 330-3028 | 2.7K 1/4W 5% |
| R27, 32, 51 | 330-3068 | 6.8K 1/4W 10% |
| R28 | 330-3027 | 2.7K 1/4W 10% |
| R30 | 330-2056 | 560Ω 1/4W 10% |
| R31 | 330-2039 | 390Ω 1/2W 10% |
| R33 | 330-3018 | 1.8K 1/2W 10% |
| R35 | 330-4033 | 33K 1/4W 10% |
| | | |
| R39 | 336-1026 | 2K POT |
| R44 | 330-4019 | 18K 1/4W 5% |
| R45, 48 | 330-3023 | 2.2K 1/4W 5% |
| R46, 62, 63 | 330-4010 | 10K 1/4W 10% |
| R54 | 330-1027 | 27Ω 1/4W 10% |
| R55 | 332-3015 | 1.5K 1W 10% |
| R56 | 332-2082 | 820Ω 1W 10% |
| R58, 59 | 330-3047 | 4.7K 1/4W 10% |
| R61 | 332-3012 | 1.2K 1W 10% |
| | | |
| C5 | 300-3030 | 4K μF 30V (E) |
| C1, 2 | 300-2210 | .1μF 100V MYLAR |
| C3 | 300-4017 | 2.2μF 20V (T) |
| C4 | 300-4008 | .33μF 35V (T) |
| C6, 7, 8 | 300-3078 | 8.2K μF 25V (E) |
| C9-13, 18-21, 27 | 300-1938 | .05μF 16V CER |
| C14-17, 26 | 300-4022 | 15μF 20V (T) |
| C22 | 300-3010 | 50μF 50V (E) |
| C23 | 300-1930 | .1μF 50V CER |
| C24 | 330-1903 | .01μF 25V CER |
| C25 | 300-1821 | .001μF 1000V CER |

| MNEMONIC | COORDINATE |
|---|---|
| POR | 1F1 |
| ~~PRMS~~ | 1F1 |
| | |
| RESET | 1G1 |
| | |
| T2 | 1G4 |
| | |
| +24FB | 1G6 |
| +24PR | 1E1 |
| +24V UNREG | 1G4 |
| +24VS | 1G6 |
| | |
| 15VAC T1 | 1F11 |
| 15VAC T2 | 1F11 |
| | |
| +12V REG | 1G5 |
| -12V REG | 1A6 |
| | |
| 11.5VAC T1 | 1G11 |
| 11.5VAC T2 | 1G11 |
| | |
| 6.5VAC T1 | 1D11 |
| 6.5VAC T2 | 1D11 |
| | |
| +5V IOPR | 1B1 |
| +5V IOVR | 1B1 |
| +5V IOVS | 1G4 |
| +5V LOGIC | 1G8 |
| +5PRI | 1G7 |
| +5VR1 | 1G7 |
| +5VR2 | 1G6 |
| +5VS1 | 1G6 |
| | |
| -5V REG | 1A7 |
| | |
| ±0V | 1E11 |

E-REV C

| NO. | REVISION | DATE | BY |
|---|---|---|---|
| 0/0 | ORC PER PWR# E795 APP'D [illegible] | 2-2-80 | A B |
| 1/0 | REVISED PER ECN # 14326 APP'D BY [illegible] | 3-25-80 | [illegible] |

WANG LABORATORIES, INC. LOWELL, MASS. U.S.A.

| | | | |
|---|---|---|---|
| BY | DATE | APPROVED BY | DATE |
| DWN [illegible] | 3/?/80 | E ENGR D. DALY | 4/?/80 |
| CHK [illegible] | [illegible] | M ENGR | |
| | | MFG ENGR | |

MATERIAL

MODEL NO. 2200LVP

SEE ENGRG. SPECIFICATIONS No.

TITLE REGULATOR

FINISH

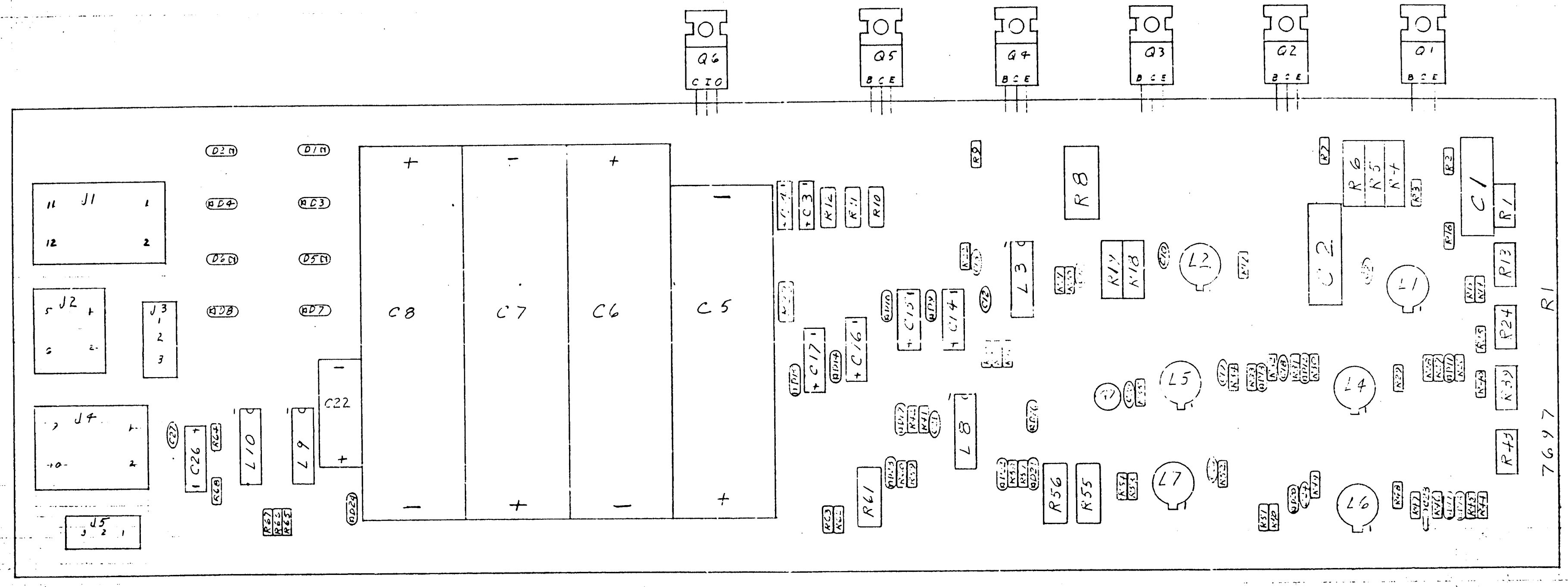
TOL. EX. AS NOTED XX ± FRAC. ± XXX ± ANG. ± FINISH ✓

SCALE — SHT 3 OF 3

| WANG PART NUMBER | SIZE | DRAWING NUMBER | REV. |
|---|---|---|---|
| 210-7697 | D | 7697 | 1 |

D 7697

DO NOT SCALE



| NO. | REVISION | DATE | BY |
|---|---|---|---|
| 1 | SEE SHT 3 | | |

| | | BY | DATE | APPROVED BY | DATE |
|---|---|---|---|---|---|
| WANG LABORATORIES, INC. LOWELL, MASS. U.S.A. | | DWN JB | 9/2/76 | E ENGR | |
| | | CHK | | M ENGR | |
| | | | | MFG ENGR | |
| MATERIAL | MODEL NO. 2200LVP | TITLE REGULATOR | | | |
| | SEE ENGRG. SPECIFICATIONS No. | | | | |
| FINISH | TOL. EX. AS NOTED .XX ± .XXX ± FRAC. ± ANG. ± FINISH ✓ | 210-7697 | D | 7697 | 1. |
| | SCALE — SHT 2 OF 2 | WANG PART NUMBER | SIZE | DRAWING NUMBER | REV. |

D 7697

## United States

**Alabama**
Birmingham
Mobile

**Alaska**
Anchorage

**Arizona**
Phoenix
Tucson

**California**
Culver City
Fountain Valley
Fresno
Inglewood
Sacramento
San Diego
San Francisco
Santa Clara
Ventura

**Colorado**
Englewood

**Connecticut**
New Haven
Stamford
Wethersfield

**District of Columbia**
Washington

**Florida**
Miami
Hialeah
Jacksonville
Orlando
Tampa

**Georgia**
Atlanta
Savannah

**Hawaii**
Honolulu

**Idaho**
Idaho Falls

**Illinois**
Chicago
Morton
Park Ridge
Rock Island
Rosemont

**Indiana**
Indianapolis
South Bend

**Kansas**
Overland Park
Wichita

**Kentucky**
Louisville

**Louisiana**
Baton Rouge
Metairie

**Maryland**
Rockville
Towson

**Massachusetts**
Billerica
Boston
Burlington
Chelmsford
Lawrence
Littleton
Lowell
Tewksbury
Worcester

**Michigan**
Kentwood
Okemos
Southfield

**Minnesota**
Eden Prairie

**Missouri**
Creve Coeur

**Nebraska**
Omaha

**Nevada**
Las Vegas
Reno

**New Hampshire**
Manchester

**New Jersey**
Toms River
Mountainside
Clifton

**New Mexico**
Albuquerque

**New York**
Albany
Buffalo
Fairport
Lake Success
New York City
Syracuse

**North Carolina**
Charlotte
Greensboro
Raleigh

**Ohio**
Cincinnati
Cleveland
Middleburg Heights
Toledo
Worthington

**Oklahoma**
Oklahoma City
Tulsa

**Oregon**
Eugene
Portland

**Pennsylvania**
Allentown
Camp Hill
Erie
Philadelphia
Pittsburgh
Wayne

**Rhode Island**
Cranston

**South Carolina**
Charleston
Columbia

**Tennessee**
Chattanooga
Knoxville
Memphis
Nashville

**Texas**
Austin
Dallas
Houston
San Antonio

**Utah**
Salt Lake City

**Vermont**
Montpelier

**Virginia**
Newport News
Norfolk
Richmond

**Washington**
Richland
Seattle
Spokane
Tacoma

**Wisconsin**
Brookfield
Madison
Wauwatosa

## International Offices

**Australia**
Wang Computer Pty., Ltd.
Adelaide, S.A.
Brisbane, Qld.
Canberra, A.C.T.
Darwin N.T.
Perth, W.A.
South Melbourne, Vic 3
Sydney, NSW

**Austria**
Wang Gesellschaft, m.b.H.
Vienna

**Belgium**
Wang Europe, S.A.
Brussels
Erpe-Mere

**Canada**
Wang Laboratories (Canada) Ltd.
Burnaby, B.C.
Calgary, Alberta
Don Mills, Ontario
Edmonton, Alberta
Hamilton, Ontario
Montreal, Quebec
Ottawa, Ontario
Winnipeg, Manitoba

**China**
Wang Industrial Co., Ltd.
Taipei
Wang Laboratories Ltd.
Taipei

**France**
Wang France S.A.R.L.
Paris
Bordeaux
Lyon
Marseilles
Nantes
Strasbourg
Toulouse

**Great Britain**
Wang (U.K.) Ltd.
Richmond
Birmingham
London
Manchester
Northwood Hills

**Hong Kong**
Wang Pacific Ltd.
Hong Kong

**Japan**
Wang Computer Ltd.
Tokyo

**Netherlands**
Wang Nederland B.V.
IJsselstein
Gronigen

**New Zealand**
Wang Computer Ltd.
Auckland
Wellington

**Panama**
Wang de Panama (CPEC) S.A.
Panama City

**Singapore**
Wang Computer (Pte) Ltd.
Singapore

**Sweden**
Wang Skandinaviska AB
Stockholm
Gothenburg
Malmo

**Switzerland**
Wang A.G.
Zurich
Basel
Geneva

**Wang Trading A.G.**
Zug

**United States**
Wang International Trade, Inc.
Lowell, Mass.

**West Germany**
Wang Laboratories, GmbH
Frankfurt
Berlin
Cologne
Dusseldorf
Essen
Freiburg
Hamburg
Hannover
Kassel
Munich
Nurnberg
Saarbrucken
Stuttgart

## International Representatives

Abu-Dhabi
Argentina
Bahrain
Bolivia
Brazil
Canary Islands
Chile
Colombia
Costa Rica
Cyprus
Denmark
Dominican Republic
Ecuador
Egypt
El Salvador
Finland
Ghana
Greece
Guatemala
Haiti
Honduras
Iceland
India
Indonesia
Ireland
Israel
Italy
Jamaica
Japan
Jordan
Kenya
Korea
Kuwait
Lebanon
Liberia
Malaysia
Malta
Mexico
Morocco
Nicaragua
Nigeria
Norway
Paraguay
Peru
Phillippines
Portugal
Saudi Arabia
Scotland
Spain
Sri Lanka
Sudan
Syria
Thailand
Turkey
United Arab Emirates
Venezuela

WANG LABORATORIES, INC.

ONE INDUSTRIAL AVENUE, LOWELL, MASSACHUSETTS 01851, TEL. (617) 459-5000, TWX 710 343-6769, TELEX 94-7421