Publication Number 29-402R17

M71-SERIES MODEL 7/16 HSALU MAINTENANCE MANUAL

INTERDATA®
A DIVISION OF
THE PERKIN-ELMER CORPORATION
Oceanport, New Jersey 07757, U.S.A.

December 1977

PAGE REVISION STATUS SHEET

PUBLICATION NUMBER 29-402

TITLE 7/16 HSALU Maintenance Manual

REVISION R17 DATE 12/77

PAGE	REV.	DATE
29-402A12 1 thru A1-7/ A1-8	R00	7/74
01-079A20 1 thru 19/20	R00	7/74
01-079R03A21 1 thru A1-10	R03	12/77
02-232M01R04A20 1 thru 10	R04	9/77
02-232M01R01A21 1 thru 10	R01	9/77
02-360A20 1	R00	6/74
2	R00	6/74
02-360A21 1 thru 4	R00	6/74
02-360A22 1 thru A1-2	R00	6/74
02-276R02A12 1 thru 6	R02	7/74
02-352R01A12 1 thru A2-2	R01	7/74
05-051R04A13 1 thru 40	R01	3/77
05-052A13 1 thru 6	R00	7/74
05-053A13 1 thru 6	R00	7/74
09-065R01A12 1 thru 10	R01	7/74
01-079M01R23D08 1-36	R23	12/77
35-522M01R13E03 1 of 1	R13	12/77
35-544R03E03 1 of 1	R03	3/75
35-524M01R06E03 1 of 1	R06	4/77
02-232M01R06D08 1-7	R05	12/77
35-391M02R05E03 1 of 1	R01	12/77
02-236R04D08 1-3	R04	9/73
02-276R01D08 1 of 1	R01	7/73
02-352R01C08 1-2	R01	7/74
09-065R01D08 1-4	R01	1-75
35-519R03D03 1 of 1	R03	9/74
02-368R01C08 1 of 1	R01	12/76

PREFACE

This manual contains a general description of the Model 7/16 HSALU (High Speed - ALU), and installation and maintenance specifications for the Processor, Selector Channel, and Memory Protect. Programming specifications for the Memory Protect, and information specifications for Test Aid, Automatic Loader, and Hexadecimal Display are included. Micro-Program listings are provided along with applicable drawings which are located at the back of the manual.

QUICK REFERENCE INDEX

To aid in quickly locating a particular section, the index marks on the edge of this page are aligned with similar marks at the beginning of each section.

GENERAL DESCRIPTION

[redacted]

29-402A12
July 1974

M 71 SERIES
MODEL 7/16 HSALU
GENERAL DESCRIPTION

1. INTRODUCTION

The Model 7/16 HASLU (High Speed-ALU) combines advanced circuitry and packaging designs to give the user a price/performance optimized machine. The Model 7/16 HASLU is completely upward compatible with INTERDATA Model 3, 4, 5, 74 and 70 Processors user instructions, interrupt handling, input/output formats and control sequencing. Because of this compatibility, the Model 7/16 HSALU can use the wide range of existing software and peripheral devices.

The Model 7/16 HSALU offers a comprehensive set of 113 instructions making the system both easy to program and efficient to operate. Through multi-function instructions and direct core addressing, coding and debugging time is reduced to a minimum.

Memory is addressable to the eight-bit byte level. Memory is expandable from the basic 8,192 bytes to 65,536 bytes. All memory is directly addressable with the primary instructions, no paging or indirect addressing is required. Sixteen 16-bit General Registers can be used as Accumulators, fifteen of which can also be used as Index Registers. Register-to-Register instructions permit operations between any of the sixteen General Registers, eliminating redundant loads and stores.

The Protect Mode of the Model 7/16 HSALU enables Memory Protect and detection of Privileged instructions, and can be activated under program control. This mode is invaluable in process control, data communication, and time-sharing operations to guarantee that a running program cannot interfere with the integrity of the system.

The Model 7/16 HSALU also provides a flexible Input/Output system in addition to conventional means of programmed I/O. In the Automatic I/O Service mode, the Processor acknowledges all I/O interrupts and automatically performs much of the overhead prior to activating the Interrupt Service Routine. In conjunction with the Automatic I/O Service, an I/O Channel can perform data transfers and signal counting without interrupting the running program until the specified sequence is completed.

Up to four Direct Memory Access Channels can be added to a Model 7/16 HSALU Memory System. These channels operate over the common Memory Bus, on a cycle stealing basis, through a Direct Memory Access Port which is built into the Processor. Two types of Direct Memory Access Channels can be used with the Model 7/16 HSALU System. The Selector Channel, which permits direct data transfer between any standard oriented INTERDATA device controller and memory; and the Direct Memory Access Channel custom designed by the user for special applications.

2. SCOPE

This specification is intended to enable the digital technician to understand the INTERDATA documentation system. Number Notation, the Part Numbering System, and the Drawing System are described. Illustrations are provided to help understand these systems. Other publications which may be of interest to Model 7/16 HSALU users are shown in Table 1.

TABLE 1. RELATED PUBLICATIONS

Title	Publication Number
Universal Clock Instruction Manual	29-26b
Users Handbook	29-261
Model 7/16 HSALU Maintenance Manual	29-402*
Multiplexor Bus Buffer Instruction Manual	29-267
8 Line Interrupt Module Instruction Manual	29-268

*This General Description is a part of 29-402

3. BLOCK DIAGRAM

A Model 7/16 HSALU simplified block diagram is shown in Figure 1. The Model 7/16 HSALU is a 16-bit digital computer. The Processor logic is contained on three PC boards:

Part No.	Description	Card File Position
35-524	CPU-C	5
35-544	CPU-B	6
35-522	CPU-A	7

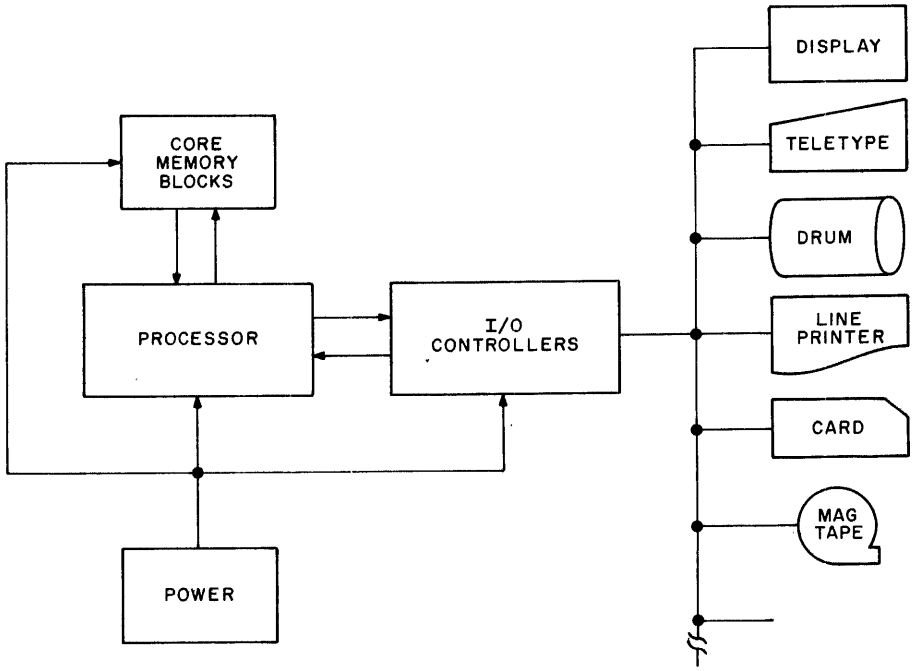


Figure 1. Model 7/16 HSALU Simplified Block Diagram

4. DOCUMENTATION

This section describes the style and conventions used with INTERDATA documentation.

4.1 Number Notation

The most common form of number notation used in INTERDATA documentation is hexadecimal notation. In this system, groups of four binary digits are represented by a single hexadecimal digit. Table 2 lists the hexadecimal characters employed.

TABLE 2. HEXADECIMAL NOTATION DATA

Binary	Decimal	Hexadecimal	Binary	Decimal	Hexadecimal	Binary	Decimal	Hexadecimal
0000	0	0	0110	6	6	1100	12	C
0001	1	1	0111	7	7	1101	13	D
0010	2	2	1000	8	8	1110	14	E
0011	3	3	1001	9	9	1111	15	F
0100	4	4	1010	10	A			
0101	5	5	1011	11	B			

To differentiate between decimal and hexadecimal numbers, hexadecimal numbers are preceded by the letter "X", and the number is enclosed in single quotation marks. Examples of hexadecimal numbers are: X'1234', X'2EC6', X'A340, X'EEFA', and X'10B9'.

4.2 Part Numbering System

INTERDATA parts, drawings, and publications employ a common numbering system. The part number and drawing numbers for drawings which describe the part are related. The publication number is also often related to the part number of the device or program described. Figure 2 shows the format used for INTERDATA part numbers. The fields are described in the following paragraphs.

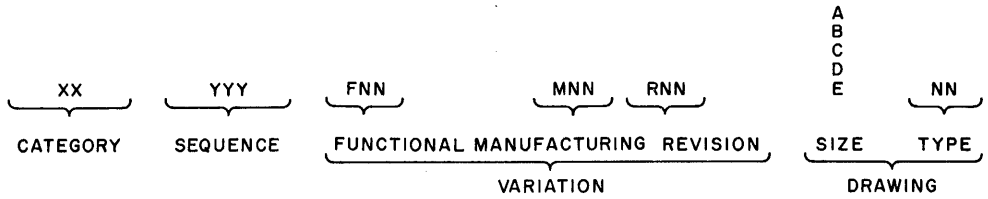


Figure 2. Part Number Format

4.2.1 Category Field. The two-digit Category number indicates the broad class or category to which a part belongs. Typical examples of category number assignments are:

01 - Basic Hardware Systems

02 - Basic Hardware Expansions

03 - Basic Software Systems

04 - Basic Software Expansions

05 - Major Application Programs

06 - Self-contained Utility Programs

07 - Subroutines of General Utility

10 - Spare Parts Packages

12 - Card File Assemblies

13 - Panels

17 - Wire and Cables

19 - Integrated Circuits

20 - Transistors

27 - Peripheral Equipment

29 - Manuals

34 - Power Supplies

35 - Assembled Printed Circuit Boards

36 - Electro-Mechanical Devices

4.2.2 Sequence Field. The Sequence number identifies a particular item within the category. Sequence numbers are assigned serially, and have no other significance.

NOTE

The Sequence Field, like all other part number fields, may be lengthened as required. The field lengths shown on Figure 2 are minimum lengths (insignificant zeros must be added to maintain these minimums).

4.2.3 Functional Variation Field. The optional Function Variation Field consists of the letter "F" followed by two digits. The F field is used to distinguish between parts which are not necessarily electrically or mechanically equivalent, but which are described by the same set of drawings. For examples, a power supply may be strapped internally to operate on either 110 VAC or 220 VAC. Except for this strap, all power supplies of this type are identical. The strapping option is easily described by a note on the assembly and test specification drawings. Therefore, this is a functional variation.

4.2.4 Manufacturing Variation Field. The optional Manufacturing Variation Field consists of the letter "M' followed by two digits.

NOTE

A part number must contain a Category number and a Sequence number. All other fields are optional.

The M Field is used to distinguish between parts which are electrically and mechanically equivalent (interchangeable), but which vary in method of manufacture. For example, if leads are welded instead of soldered on an assembly, the M Field changes.

An important exception to the meaning of the M Field exists for categories related to software. In Software, the M Field number, when used, indicates the form in which a particular program is presented. For example, define a program as a set of machine instructions. These same identical instructions may be presented on punched cards, paper tape, or magnetic tape; and for any of these they could be in symbolic form or in relative or absolute binary form. Thus, there are many ways to present the same identical program.

The format for the M field and its meaning for software is:

M x y

where x identifies the media selection (i.e., paper tape, mag tape, cassette, etc) and
y identifies object or source and the format.

Meaning of x		Meaning of y
Paper tape	1	1 Object program standard format 32 bit Processor
Cassette	2	4 Memory Image
Mag tape (800)	3	6 Object program standard format 16 bit Processor
Cards	4	7 Object non-standard format
Disc (2.5)	5	7 Object established task
		9 Source program

The above numbers refer to the physical program placed on an approved media for INTERDATA Software.

A paper tape object program in standard format and for a 16 bit Processor has an M16 identifier.

A magnetic tape object program in standard format and for a 32 bit Processor has an M31 identifier.

In addition to the above, there are three unique M numbers which have special meaning:

M99 always refers to a documentation package.
M00 always refers to a conceptual object program divorced from any media. This reference is used for all parts lists when object programs may be on any media.
M09 always refers to a conceptual source program and is used on all parts lists where any media may be used.

NOTE

M00 and M09 may only be used on parts lists and never identifies a physical program on any media.

4.2.5 Revision Field. The optional Revision Field consists of the letter "R" followed by two digits. The R Field is used to indicate minor electrical or mechanical changes to a part which does not change the part's original character. R Field changes often reflect improvements. A part with a revision level HIGHER than the one specified will work. A part with a revision level LOWER than specified should not be used.

4.2.6 Drawing Field. The optional Drawing Field consists of a letter from "A" to "E" followed by two digits. The letter indicates the size of the original drawing. The sizes for each letter are:

A - $8\frac{1}{2}$" X 11"
B - 11" X 17"
C - 17" X 22"
D - 22" X 34"
E - 34" X 44"

The two digits indicate the drawing type as follows:

01 - Parts List
02 - Machine Details
03 - Assembly Details
05 - Art Details
06 - Wire Run List
08 - Schematic
09 - Test Specification
10 - Purchase Specification
11 - Bill of Material
12 - Information
13 - Program Listing
14 - Abstracts
15 - Program Description
16 - Operating Instructions
17 - Program Design Specification
18 - Flow Charts
19 - Product Specification
20 - Installation Specification
21 - Maintenance Specification
22 - Programming Specification

4.2.7 Examples. The following list provides some examples of the part numbering system. The numbers were arbitrarily selected, and in most cases are fictitious.

35-060 The 60th printed-circuit board assigned a part number under this system.

35-060M01 A printed circuit board electrically and mechanically interchangeable with the 35-060, but differing in method of manufacture.

35-060F01 A printed-circuit board not electrically and/or mechanically interchangeable with the 35-060, but described by the same set of drawings.

35-060R01 A revised 35-060 printed-circuit board. Probably supercedes the 35-060.

35-060A01 The $8\frac{1}{2}$ by 11 inch parts list for a 35-060.

35-060B08 The 11 by 17 inch schematic for a 35-060.

06-072 The 72nd utility program assigned a part number.

06-072A13 An $8\frac{1}{2}$ by 11 inch listing of the 06-072 program.

06-072M03 An absolute binary deck of punched cards for the 06-072 program.

06-072A12 An $8\frac{1}{2}$ by 11 inch information drawing on the 06-072 program. Probably a part of the program.

29-060 The 60th manual assigned a number under this system. Note that this number is not referenced in any way to the part number of equipment described in the manual.

4.3 Drawing System

This section describes the drawings provided with INTERDATA equipment. Note that drawings provided with peripheral devices and other purchased items may vary from the system described in this Section.

A digital system may be divided into a collection of functionally independent circuits such as core memory, Processor, and I/O device controllers. These circuits may or may not be saleable units in their own right, but in the electrical sense they are essentially self contained and capable of performing their function with minimum dependence on other functional circuits in the system. Hence a functional circuit is treated as a building block. Each functional circuit is described electrically by a detailed functional schematic. Each schematic contains a variety of information including type and location of discrete integrated circuits (IC's), pin connections, all interconnections within the schematic connector pin numbers and connections to other schematics. Further, the schematics are drawn to reflect, in an orderly fashion, all logical operations performed by the circuits. Generally, symbols used on schematics conform to MIL-STD-806B.

Registers are named according to the following rules:

1. The register mnemonic name has a maximum of three letters.
2. Each bit in the register is numbered, usually starting at 00 on the left, or most significant position, and continuing to N-1 on the right, where N is the number of bits in the register.
3. The 00 bit is the Most Significant Bit and the N-1 is the Least Significant Bit.

The IC's, mounted directly on the logic board, are represented on the schematic drawings by logic symbols. Each symbol contains the reference designation, device part number (category and sequence), and symbol mnemonic designation. Refer to Figure 3.

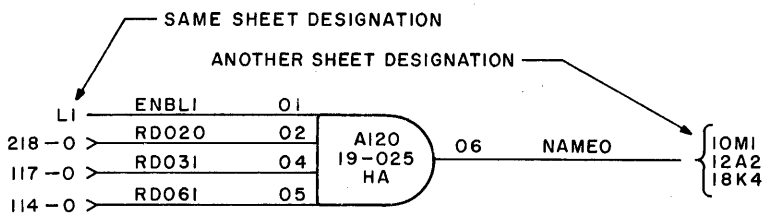


Figure 3. Example of a High Speed AND Gate.

The designations, numbers, and references shown in Figure 3 are:

A120 - This indicates the component location on the logic board. Figure 4 illustrates the method generally used to determine component location on a logic board. With the logic board oriented so that the header connectors (Conn 0 and Conn 1) are on the right, the components are numbered from left to right starting in the upper left corner. That is, the first IC in the upper left corner is A01 and the first capacitor is C1.

19-025- The number 19 is the category number of ICs, and the 025 is the sequence number of the component.

HA - Indicates this component is a high speed AND gate. Some other common designations used are:

P - Power Gate
HP - High Speed Power Gate
G - Gate
HG - High Speed Gate
HGO - High Speed Gate, Open Collector
B - Buffer
HB - High Speed Buffer

L1 - This input lead is from area L1 on the same schematic sheet.

10M1, 12A2, 18K4 - Indicate outputs to another logic schematic sheet.

218-0, 117-0, 114-0 - Indicate inputs from Connector 0.

Note that the pin numbers (01, 02, 04, 05, and 06) correspond directly to the actual IC pin numbers.

Figure 4 also shows the locations of the header connectors (Conn 0 and Conn 1) and the cable connectors (Conn 2 and Conn 3). All logic boards always contain Header Connectors 0 and 1, however, any combination (either, both, or none) of cable connectors (Conn 2 and Conn 3) may be provided.

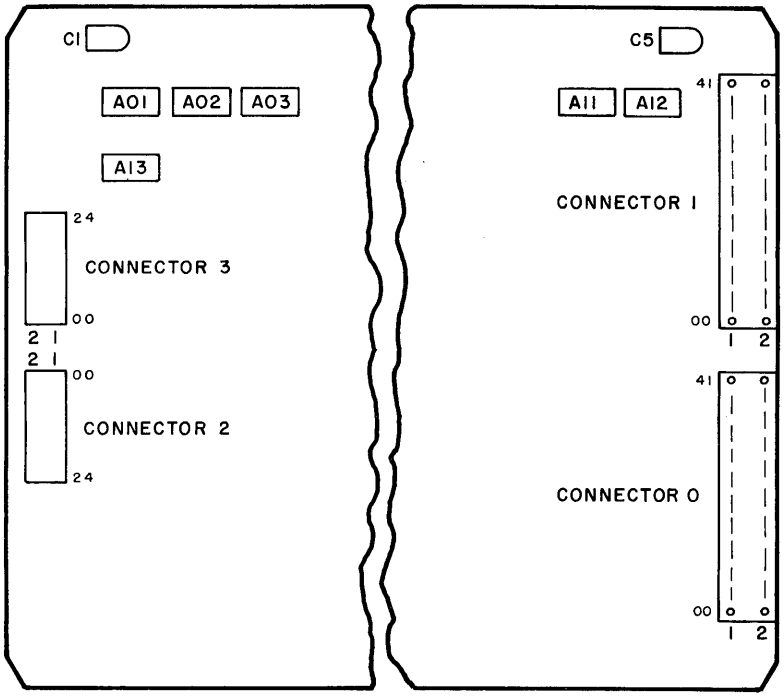


Figure 4. Example of a Logic Board Layout

Wherever possible, the immediate output of a flip-flop (1 or 0 side) will have a mnemonic name preceded by an 'F'. A flip-flop whose name is PSEL (Processor selected) will have an output mnemonic, on the 0 side, FPSEL0 (see Figure 5). This provides the digital technician with an indication when observing a mnemonic at the terminal end of a net, that the signal is the output of a flip-flop rather than a decoded function.

Clocked devices, flip-flops and counters in particular, are drawn in a manner which indicates information concerning their inputs. An input which has a circle adjacent to the pin designation implies a low active signal is required to perform the specified operation. In addition, an inverted V at the clock input shows that the device changes state on an edge. Thus, if no circle is present the chip is positive edge triggered. Refer to Figure 5 for examples.

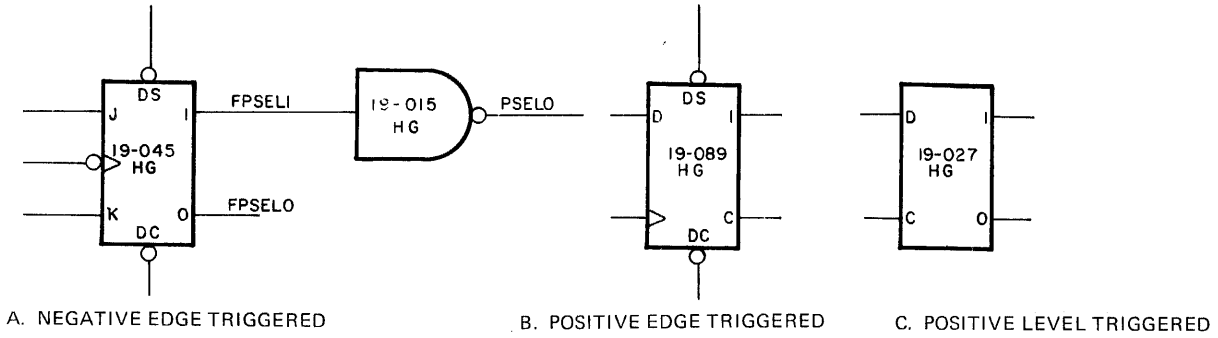


Figure 5. Examples of Clocked Devices

Figure 6 provides the pin numbering scheme for the header and cable connectors. Header connectors always have 2 rows of pins and 42 positions. Cable connectors always have 2 rows of pins but may vary in the number of positions. The number of positions may only vary in increments of five positions (10 contacts). For instance, if 24 positions are desired, five blocks of five positions each (25 positions) must be used.

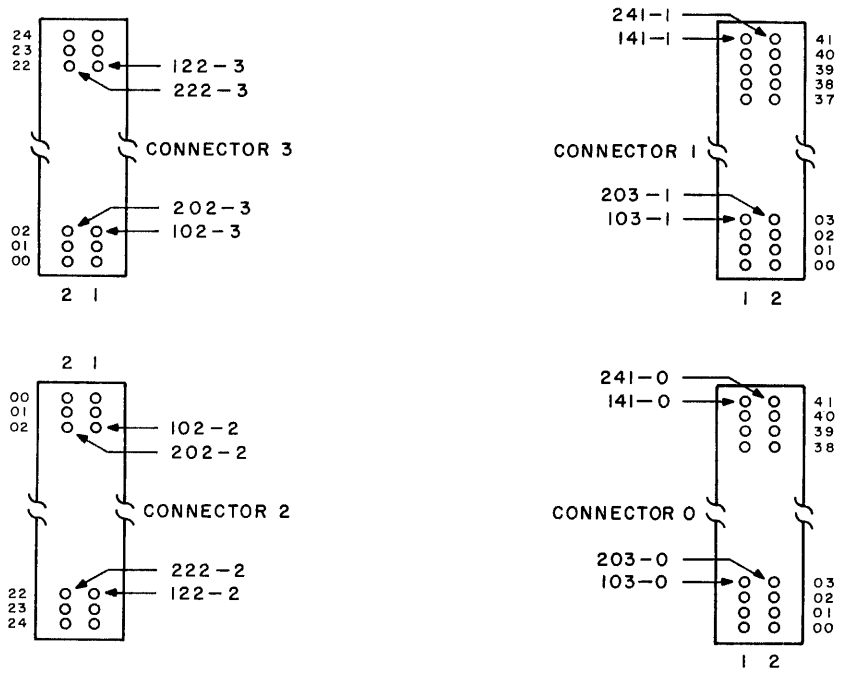


Figure 6. Connector Pin Numbering

A net is defined as an electrical connection between two or more points in a circuit. Ordinarily, a net has an originating end (usually a collector where the signal is generated) and one or more terminating ends. Often it is convenient to assign descriptive mnemonic names to nets as a way of identifying them on schematics. Whether a net is named or not is sometimes arbitrary. However, a net is always assigned a name if:

1. The net is contained on one drawing sheet but is not shown as a complete solid line on that sheet.
2. Part of the net appears on more than one sheet.
3. Part of the net connects with a different schematic.
4. Part of the net leaves a logic board.

If a net is named, the following rules are observed.

1. All mnemonic names are a maximum of six characters.
2. All decimal digits and upper case letters are permitted.
3. No other characters permitted.
4. Where possible, mnemonics are descriptive. However, it should be recognized that descriptive names are not always possible and a danger of misinterpreting a mnemonic exists.
5. Mnemonic names are not repeated within a schematic.
6. Every mnemonic is suffixed by a state indicator. This indicator consists of the digit "1" for the logically true state, or the digit "0" for the logically false state. For example, the set side of a flip-flop would have the "1" state indicator, while the reset side would have the "0" state indicator. The state indicator for a function changes each time that function is inverted. Thus, the state indicator permits assigning the same mnemonic to functions that are identical except for an inversion.
7. When a logical function is inverted, an inversion indicator is added after the state indicator. This allows for functionally equivalent, but electrically different nets to have the same mnemonic name. For example, assume a signal NAME1. NAME1 may be inverted to produce NAME0. If NAME0 is then inverted, NAME1A is produced. NAME1 and NAME1A are functionally equivalent, but physically different nets.

Sometimes a net fans-out to many sheets in a schematic. It is also possible for a net to fan-out to sheets in different schematics. In these situations, the net is assigned a mnemonic name. The net is also "zoned" from sheet to sheet to allow for properly identifying the originating and terminating ends of the net. The originating end of a net is defined as the collector at which a signal is generated. All other points to which the net connects are called terminating ends. When a lead leaves a sheet at the originating end, it is zoned to each and every sheet on which the net reappears, by indicating first the page number, followed by the schematic number that contains the page. For example, assume that the gate shown on Figure 3 is on a schematic, sheet 20. The output, NAME0, appears on sheets 10, 12 and 18 of the schematic. Note that the schematic number is implied. When a net enters a sheet from another sheet, it is labeled with the same mnemonic name, and is zoned back to the originating end of the net only. Thus, on Figure 3, the ENBL1 may, however, have many other terminations in addition to the one shown. Generally then, when a net leaves the sheet where it originates, it is zoned to every other sheet where the net terminates, while the terminating end is zoned only to the originating sheet. Note that in the Model 70 schematics, signals are co-ordinated between sheets only when the sheets are related to the same board. When a signal leaves a board, the Back Panel Map must be used.

When a lead leaves a logic board, it usually does so through a logic board back panel connector pin. These connector pins must be shown on the schematic even if the complete net is shown on one drawing sheet. Only the connector pin number need be indicated under the pin symbol, since the connector number itself is implied by the logic board location number in the logic symbol or in the footnote. Thus, on Figure 3, RD061 enters the logic board on Pin 114 of Header Connector 0.

Figure 7 is a typical schematic sheet with call-outs illustrating many of the conventions described in this section.

The schematic drawings for the basic Digital System and some of the more common expansions are commonly included in the rear of the appropriate Digital System Maintenance Manual. Schematic drawings for other expansions are included with the expansion or with the publications that describe the expansion.

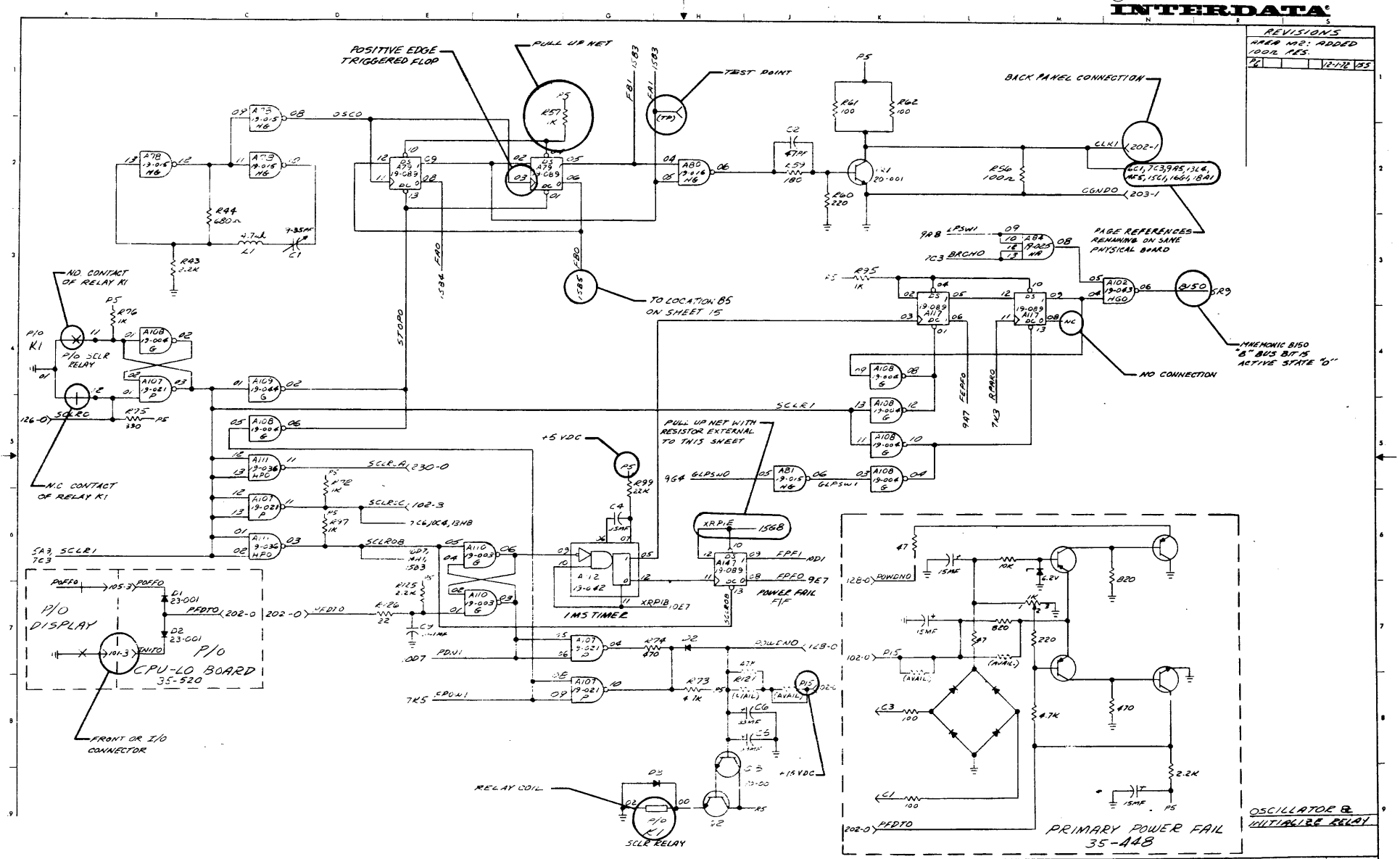


Figure 7. Functional Schematic Format Drawing

APPENDIX 1

PART NUMBER CROSS REFERENCE TABLE

INTERDATA Part Number	Type	JEDIC Number
19-001	Dual 4 Input Nand DTL	15861
19-002	Triple 3 Input Nand DTL	15863
19-003	Quad 2 Input Nand DTL	15849N
19-004	Hex 1 Input Nand DTL	15837N
19-005	Dual Power Gate DOC	8633N
19-006	Dual Buffer DTL	1582N
19-007	Flip-Flop DTL	15848N
19-008	Gate Expander Dual 4 Input DTL	15833N
19-009	8 Bit Stack DTL	903059 (Fairchild)
19-010	Differential Compartor LIN	72710L
19-012	Dual 4 Input Buffer TTL	74H40H
19-013	Quad 2 Input Nand DTL	15846
19-014	Dual J-K Flip-Flop DTL	158097N
19-015	Hex Inverter 1 Input	74H04H
19-016	Quad 2 Input TTL	74H00N
19-017	Triple 3 Input TTL	74H10N
19-018	Dual 4 Input TTL	74H20N
19-019	Single 8 Input TTL	MC3015 (Motorola)
19-020	Operational Amplifier LIN	MC1709C (Motorola)
19-021	Quad 2 Input Power DOC	15858N
19-022	Dual J-K Flip-Flop TTL	MC3061P (Motorola)
19-023	Selected Dual Buffer 19-006 with 20-30 nsec. delay DTL	15832N
19-024	Triple 3 Input AND TTL	74H11N
19-025	Dual 4 Input AND TTL	74H21N
19-026	2-2-2-3 Input AND-OR TTL	74H52

INTERDATA Part Number	Type	JEDIC Number
19-027	4 Bit Latch TTL	7475N
19-028	4 Bit Adder TTL	7483N
19-029	Quad Exclusive - OR TTL	7486N
19-030	4 Bit Shift Register TTL	7495N
19-031	One Shot TTL	7412N
19-032	1 out of 10 Decoder TOC	74145N 5445 7445
19-033	Sense Amplifier LIN	7524N
19-034	Retriggerable One Shot TTL	74122N
19-035	4 Bit Counter TTL	74193N
19-036	Quad 2 Input Open Collector TTL	7438N
19-037	High Performance Operational Amp	7748393 (Fairchild)
19-038	Dual 4 line to 1 line Mux TTL	74153
19-039	4 Bit ALU TTL	74181
19-040	Look Ahead Carry TTL	74182
19-041	4 x 4 Register Stack TTL	74170
19-042	Dual Retriggerable One Shot TTL	74123N
19-043	Quad 2 Input Open Collector TTL	74H01N
19-044	Hex Inverter Open Collector TTL	74H05N
19-045	Dual J-K Flip-Flop TTL	74H106
19-046	Quad RS-232C Line Driver	MC1488L (Motorola)
19-047	Quad RS-232C Line Receiver	MC1489AL (Motorola)
19-048	8 Bit Shifter	74198N
19-050	8 Input Nand TTL	74H30
19-051	1024 Bit PROM TTL	74187 (Fairchild)
19-055	Quad 2 Input Nand STTL	74S00
19-056	Quad 2 Input Nand Open Collector STTL	74S03
19-057	Hex 1 Input Inverter STTL	74S04
19-058	Triple 3 Input Nand STTL	74S10
19-059	Triple 3 Input AND STTL	74S11
19-060	Dual 4 Input Nand STTL	74S20

INTERDATA Part Number	Type	JEDIC Number
19-061	Dual 4 Input Buffer STTL	74S40
19-062	2-2-3-4 Input AND-OR Inverter STTL	74S64
19-063	Dual D Edge Triggered Flip-Flop STTL	74S74
19-064	Dual J-K Flip-Flop STTL	74S112
19-065	Quad 2:1 Max Non-inverting STTL	74S157
19-066	Quad 2:1 Mux Inverting STTL	74S158
19-067	4 Bit ALU STTL	74S181
19-068	Carry Look Ahead STTL	74S182
19-069	8 line to 1 line Mux STTL	74151
19-070	4 Bit Syncronous Counter TTL	74161
19-071	Quad D Edge Triggered Flip-Flop	74175
19-072	4 Bit Left/Right Shift Register TTL	74194
19-073	Dual 4:1 Mux Tri-State TTL	8214 (National)
19-074	8 Bit Priority Encoder TTL	9318 (Fairchild)
19-075	16 x 4 Register Stack TTL	3101A (Intel)
19-076	1024 Bit Memory MOS	TM54062
19-077	256 Bit Memory TTL	6531 (Monolithic Memories)
19-078	Dual 4 Input Nand-OC	74S22
19-080	High-Speed PROM	82S29 (Signetics)
19-081	Univ. Asynchronous Receiver/Transmitters	TR1042A (Western Digital)
19-082	2-2-3-4 Input AND-OR Invert Open Collector STTL	74S65
19-083	9 Bit Parity Generator/Checker STTL	82S62 (Signetics)
19-085	Monolithic Timing Circuit	MC1555 (Motorola) NE555V (Signetics)
19-086	741 C DIP Operational Amplifier	U6A7741393 (Fairchild)
19-087	747 DIP Operational Amplifier	U7A774 (Fairchild)
19-088	737 C DIP Operational Amplifier	U6A773393 (Fairchild)
19-089	Dual D Edge Triggered Flip-Flop	74H74
19-090	High Speed (710) Differential Comparator DIP	U6A771093 (Fiarchild)
19-091	Retriggerable Single One Shot	9600 (Fairchild)

INTERDATA Part Number	Type	JEDIC Number
19-092	Negative Voltages Regulator	MC1463R (Motorola)
19-093	Positive Voltages Regulator	MC1469R (Motorola)
19-094	Voltage Regulator	U6A7723393 (Fairchild) MC1723CL (Motorola)
19-095	Linear Positive Voltage Regulator	U9H7805393 (Fairchild)
19-096	First In-First Out Serial Memory 64 Word 4 Bit	3341 (Fairchild)
19-097	Amplifier	LH0002H (National)
19-098	Quad 2:1 Multiplexor Non-Inverting	74157
19-099	Dual Sense Amplifier	75234N
19-100	Driver	75452N
19-101	4-2 Input Buffer	7437N
19-102	6-1 Input Buffer OC	7407N
19-103	1 out of 10 Decoder	7442N
19-104	Current Switch	75325N
19-105	Dual Differential Driver	Fairchild 9614
19-106	Dual Differential Receiver	Fairchild 9615
19-107	Sense Amplifier	SN7520N
19-108	Quad 2 Input Nand	SN7400N
19-109	Hex Inverter Open Collector	SN7406N
19-110	Hex Inverter	SN7404N
19-111	Dual 4 Input Nand	SN7440N
19-112	Optically Coupled Isolator	TI-111 4N25
19-113	360 Dual Transmitter	TI 75123
19-114	360 Triple Receiver	TI 75124
19-115	Quad 2 Input AND	74H08
19-116	Dual 4:1 Multiplexor STTL	74S153
19-117	4 Bit Magnitude Comparitor STTL	74858
19-118	Quad Bus Transceiver TTL	26S12A
19-119	Expandable AND-OR Invert TTL	74H55
19-120	Dual Timer	Signetics NE556

INTERDATA Part Number	Type	JEDIC Number
19-121	Matched Pair 19-085 (P.S. Timing)	MC1555 (Motorola) NE555V (Signetics)
19-123	Dual Voltage Controlled Oscillator	TI 74S124
20-001	Transistor NPN High Speed Switch	2N3646
20-002	Transistor PNP 500 MA	MPS6534 (Motorola)
20-003	Transistor	2N3902
20-004	Transistor NPN	2N5189
20-006	Transistor NPN 15 Amps 100W T03 case	2N3055 (RCA)
20-007	Transistor NPN 3 Amps	TIP31A
20-008	Transistor PNP 3 Amps	TIP32A
20-009	Transistor Triac 2 Amps 100V	A03001 (Electronic Control Corp).
20-010	Transistor NPN 500 MA Code Driver	2N5845
20-011	Transistor Photo	2N5777
20-012	Transistor PNP High Current Switch	2N2907
20-013	Transistor NPN	2N3303
20-014	Transistor NPN	2N4238
20-015	Transistor PNP	2N4235
20-016	Transistor PNP	2N3740
20-017	Transistor NPN	2N3766
20-018	Transistor, Power Silicon NPN	2N3054
20-019	Transistor NPN Fast PWR Switch	2N6308 (Motorola)
20-020	Transistor Switching 1 Amp T05 can	2N3725
20-021	Transistor NPN Silicon	MPS3646 (Motorola)
20-022	Transistor NPN	1N1711
20-023	Transistor PNP	2N2905A
20-024	Transistor Switch	2N3776
20-025	PNP Hi Speed Switch	2N3467
20-026	Transistor Module, Quad	MPQ3725

INTERDATA Part Number	Type	JEDIC Number
20-027	Transistor	2N2369
20-029	Transistor	
21-025F01	1K ohm-15 to Common DIP	898-1-1K ohm (Beckman)
21-025F02	470 ohm-15 to Common DIP	898-1-470 ohm (Beckman)
21-025F03	330 ohm-15 to Common DIP	898-1-330 ohm (Beckman)
23-001	Diode High Speed-High Current	1N914
23-002	Diode 5.1 V Zener	1M5.1ZS5 (Motorola)
23-003	Diode 10V Zener	1M10ZS5 (Motorola)
23-004	Diode 6.2 V Zener	1M6.2ZS5 (Motorola)
23-007	Diode Mot Bridge	MDA962-2 (Motorola)
23-008	Diode Int. Rectifier	40HF-5R
23-009	Diode	1N4735
23-010	Diode Int. Rectifier	S1Y1P
23-011	Diode Rectifier	2N681
23-012	Diode Thermister	KA31J1 (Fenwall)
23-013	Diode 9.4V	1N2163
23-014	Diode	1N3880
23-015	Diode	1N3889
23-016	Diode Bridge Recitifer	VS448 (Varo)
23-017	Diode	1N2070
23-018	Diode 18 V Zener	1N4746A
23-019	Diode	1N3615
23-020	Diode 8.2V Zener	1N756A
23-021	Diode 9.1 V Zener	1N757A
23-022	Diode 3.3V Zener	1N746A
23-023	Diode Bridge Rectifier	KBH2506 (General Instrument)
23-024	Diode, Power Fast Rec. 30 Amps.	1N3909
23-025	Diode, Power Fast Rec. 3 Amps.	A115A (General Electric)
23-026	Triac 600V 30 Amps	2N6162

INTERDATA Part Number	Type	JEDIC Number
23-027	Diac 32V	1N5761
23-028	Power SCR Thyristor	2N4441
23-029	Diode	1N4607
23-030	Diode	1N4156
23-031	Diode 6.6 V Zener	1N4736
23-032	Diode 8.8 V Zener	1N4739
23-033	16 Diode Array	45190 (Litton)
30-018	100 nsec. Delay Line 10 taps	30-018 (Princeton Advanced Eng.)
30-019	50 nsec. Delay Line 10 taps	30-018 (Princeton Advanced Eng.)

PROCESSOR

01-079A20
July 1974

M71-SERIES
MODEL 7/16 HSALU
INSTALLATION SPECIFICATION

1. INTRODUCTION

The INTERDATA Model 7/16 HSALU (High Speed-ALU) Digital System features a highly modular structure which permits configurations to suit the user's exact needs. It provides the means for convenient expansion as the user's requirements grow. This document describes the Processor, System Expansion Chassis, and Power Supply Mounting. Filler and Display Panel mounting, and the interconnecting cables. Printed circuit boards are discussed with respect to cabling and location only. Circuit descriptions of these boards are provided in the appropriate maintenance or instruction manuals. Note that the following discussion assumes that the equipment is mounted in standard INTERDATA cabinets. In addition, this specification covers the installation of the Turnkey Console.

2. MECHANICAL COMPONENTS

This section is intended to familiarize the reader with the mechanical components that are discussed here (i.e., Cabinet Uprights, Chassis Support Rails, Filler Panels). Figures 1 through 4 provide the dimensions and mounting configurations for the Rack, Chassis Support Rails, and Filler/Display Panels. Note in Figure 4, that while $3\frac{1}{4}$", 7", and $10\frac{1}{2}$" Filler Panels and the Display Panel mount the same way (via retaining brackets), the smaller 1 3/4" Filler Panel mounts with spring clips.

3. PROCESSOR AND EXPANSION CHASSIS MOUNTING

Two Expansion chassis (10 inch and 15 inch) are available for expanding the Model 7/16 HSALU Digital System. The (15 inch) Expansion chassis has the same over-all dimensions as the Processor chassis. See Section 9 on Configuration.

The Expansion or Processor chassis slides into the rack on the two Chassis support rails (see Figures 2 and 3) from the front of the rack.

CAUTION

NO CHASSIS SHOULD BE MOUNTED IN CANTILEVER FASHION. CHASSIS SUPPORT RAILS MUST BE USED. IF A RACK CABINET OTHER THAN AN INTERDATA CABINET IS USED, CONSULT RACK MANUFACTURER FOR PROPER SUPPORT RAILS.

The chassis support rails are fastened to the mounting uprights at the front and rear of the rack. Slots are provided in the rails to allow vertical adjustment. The Expansion or Processor Chassis are screwed in place at the mounting uprights in front of the rack. All Expansion Chassis mount below the Processor Chassis. Expansion Chassis cabling is discussed later in this document. Figure 5 shows a front and rear view of the Processor or Expansion Chassis. Figure 6 shows Expansion Chassis location with respect to the filler panel and power supply.

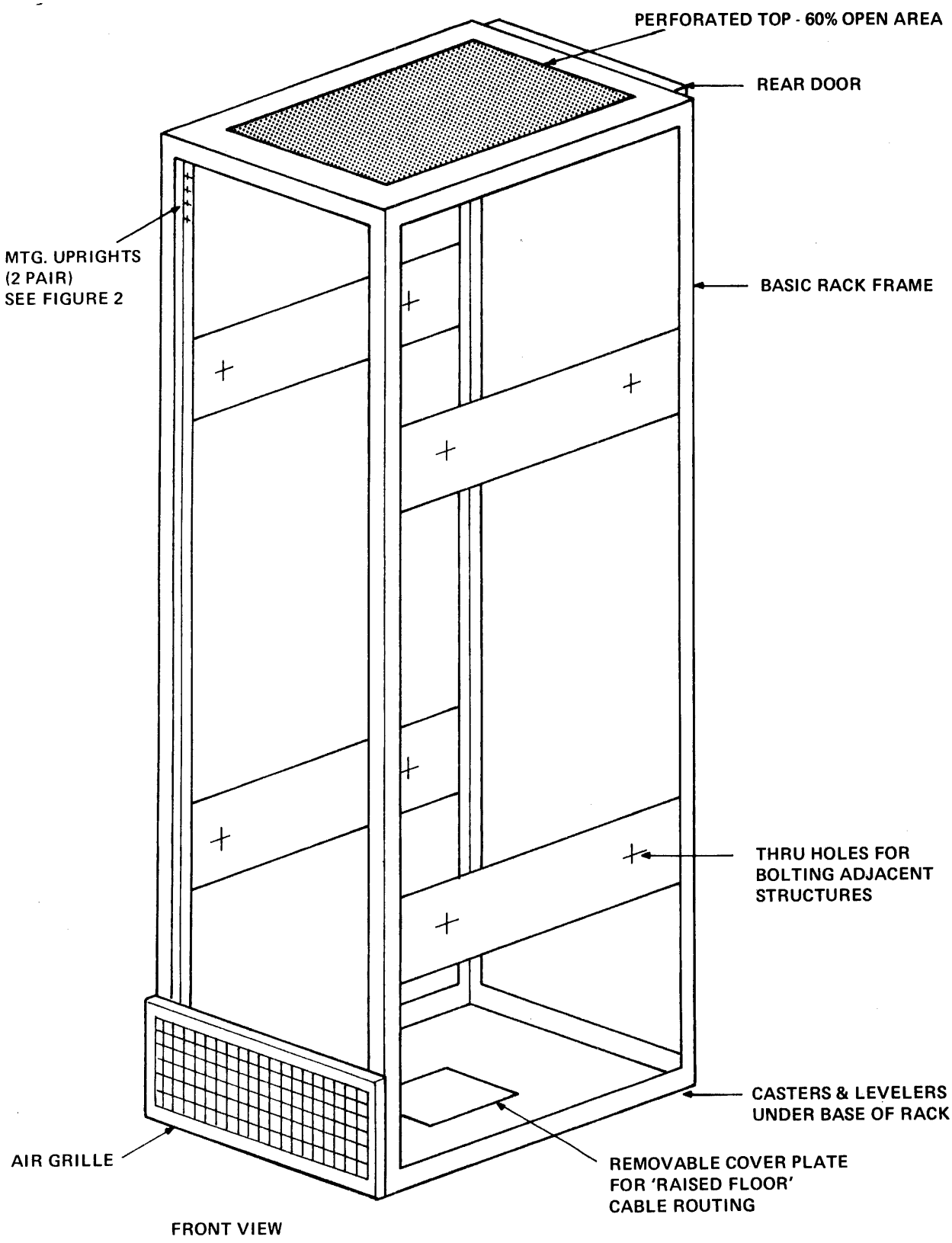


Figure 1. Basic Cabinet

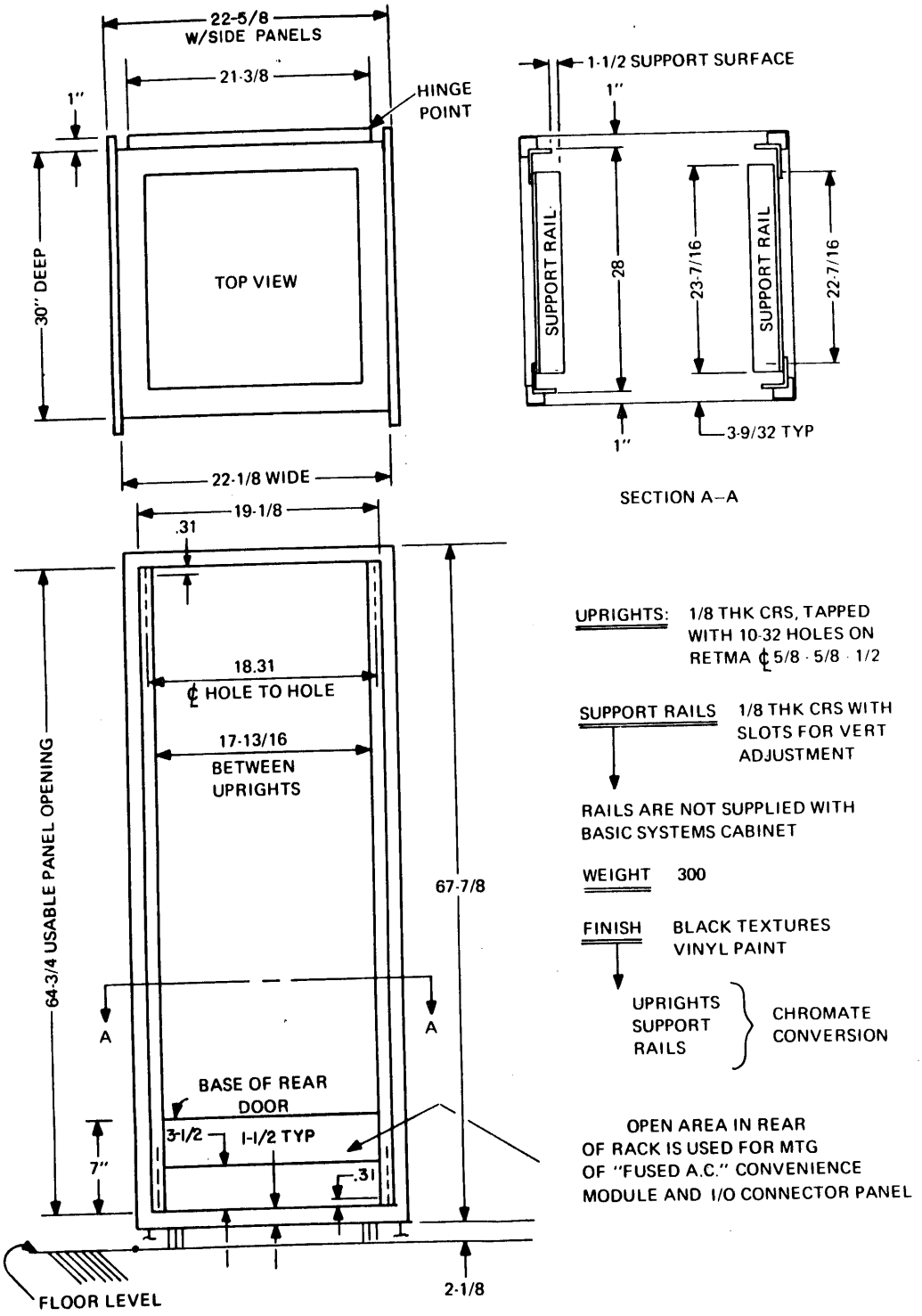


Figure 2. Basic Cabinet Physical Dimensions

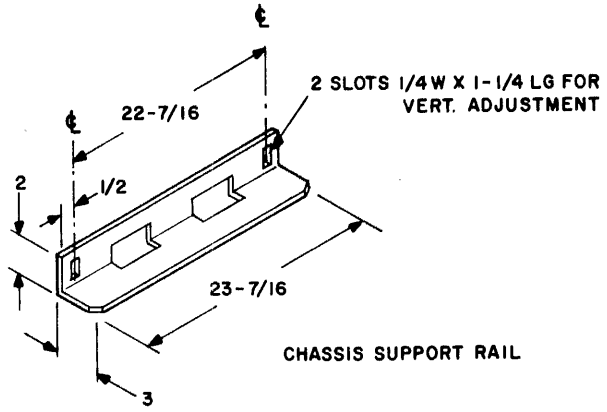


Figure 3. Chassis Support Rail

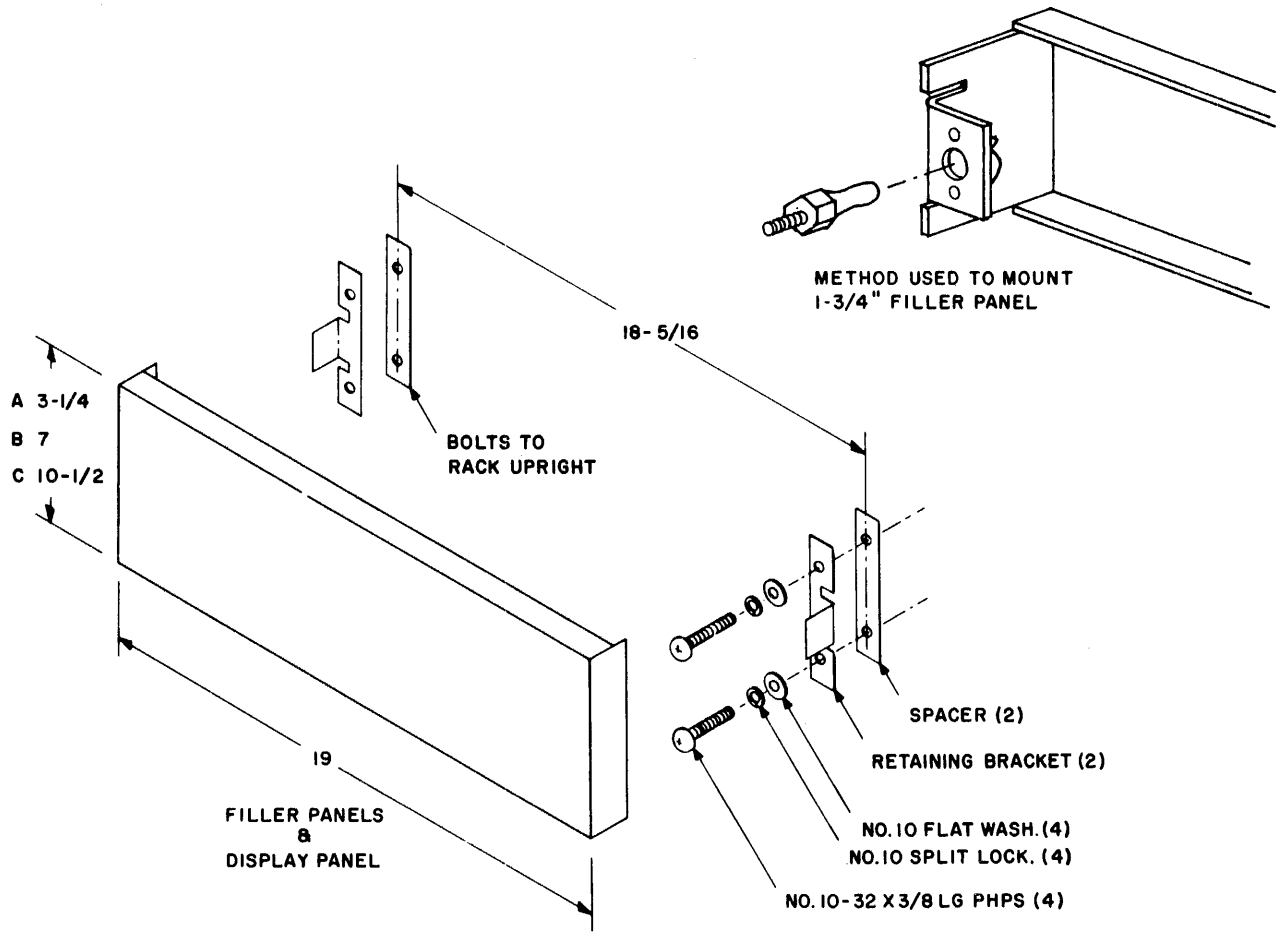
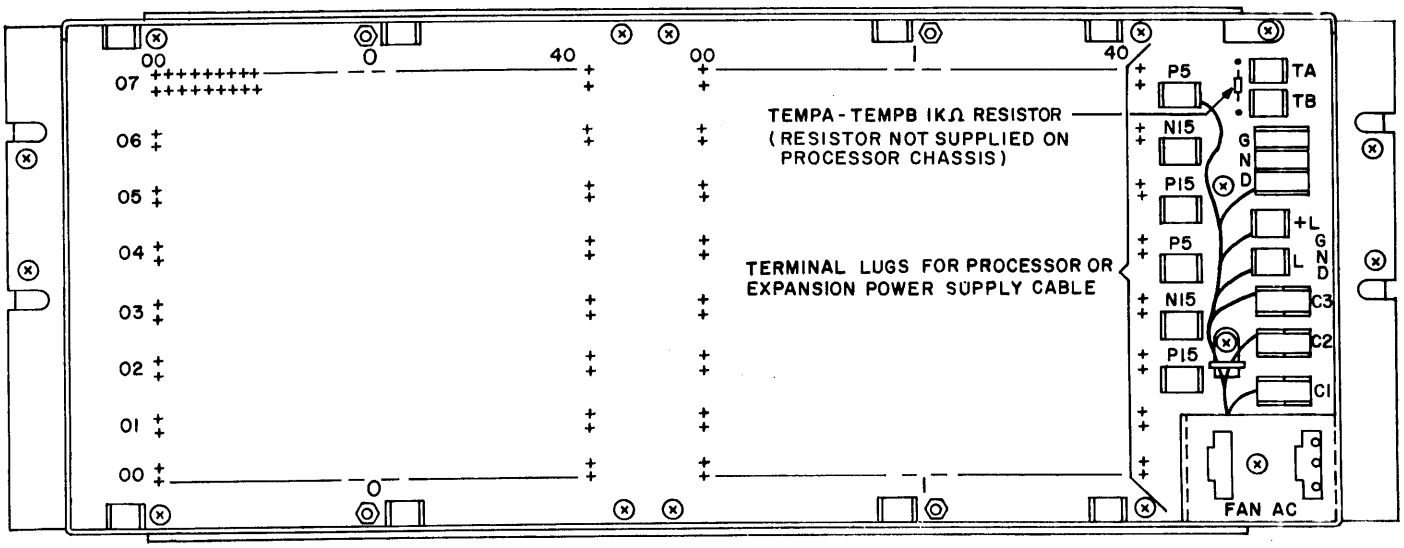
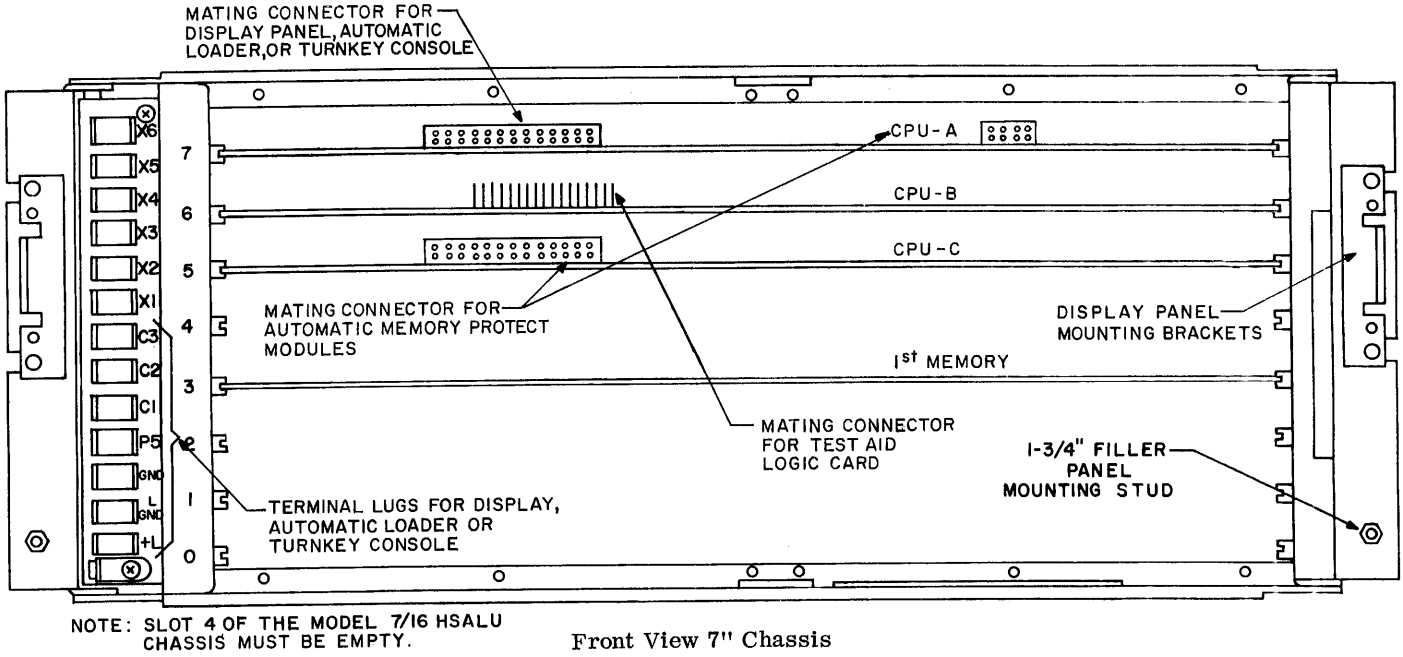


Figure 4. Typical Mounting Configuration for Display and Filler Panels



Rear View 7" Chassis



Front View 7" Chassis

Figure 5. View of Model 7/16 HSALU Processor and 15 Inch Expansion Chassis

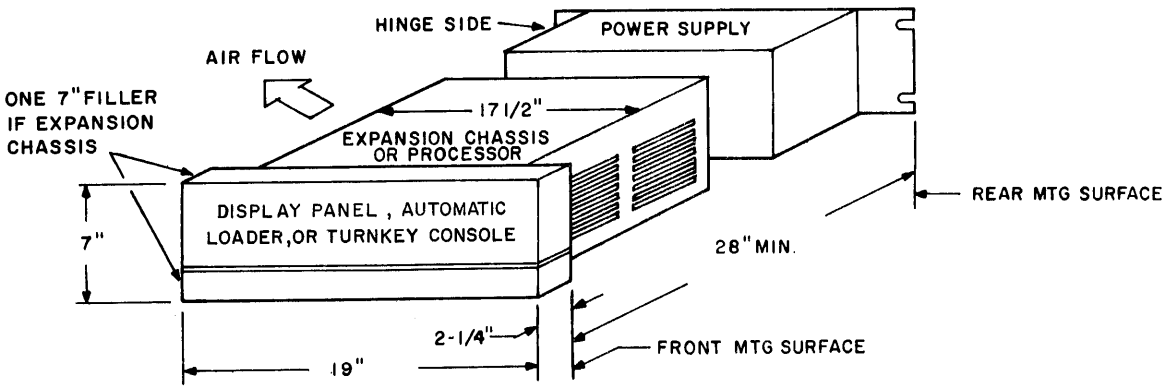


Figure 6. Processor or Expansion Chassis Location

3.1 15 Inch Expansion Chassis

The 15 inch Expansion Chassis contains eight universal expansion slots which can accept combinations of memory modules, single board peripheral controllers, system modules, Selector Channel, or user designed interfaces. Included with this chassis are the cooling fans and interconnecting cables. The chassis may be ordered with or without a power supply.

3.1.1 7 and 10 Inch Boards in a 15 Inch Chassis. A 10 inch I/O Controller (provided it does not use Connector 1) may be inserted in a 15 inch chassis via the 02-234 I/O Adapter Kit (see Figure 7). One or two 7 inch boards (half boards) may be inserted into a 15" chassis via the 16-398 Half Board Adapter Kit (see Figure 8). The Half Board Adapter Kit may hold two active 7" boards or one active and one blank 7" board, depending on requirements. No wiring takes place between the boards and the adapters. The adapters are designed such that the connectors on the boards plug directly into the Expansion Chassis.

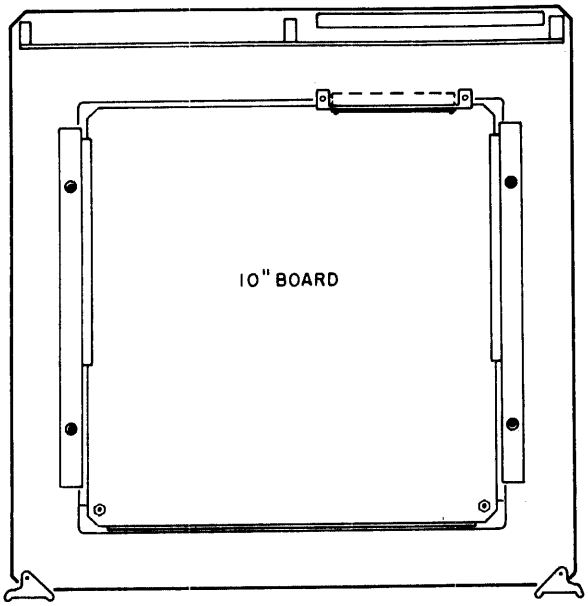


Figure 7. 02-234 I/O Adapter (Top View)

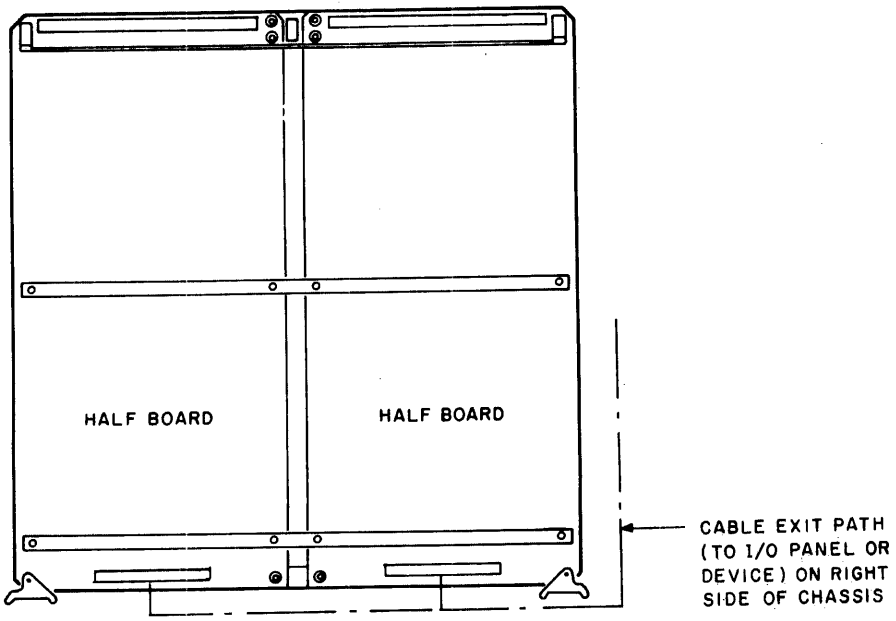


Figure 8. 16-398 Half Board Adapter

3.2 10 Inch Expansion Chassis

The 10 inch Expansion Chassis contains six 10 inch I/O expansion slots which can accept any combination of up to six 10 inch wire-wrap or copper peripheral controllers, systems, modules, or user designed interfaces. Included with the chassis are the cooling fans and system interconnecting cables. The Power Supply is separate.

4. POWER SUPPLY MOUNTING

The Power Supply mounts in the rear of the cabinet, behind the Processor or Expansion chassis. It is attached to the right mounting upright (looking from rear). Either of two Power Supplies may be supplied with the Model 7/16 HSALU System.

WARNING

BEFORE HINGING OUT THE POWER SUPPLIES, THE RACK LEVELLING FEET SHOULD BE LOWERED. AFTER THE LEVELLERS ARE IN CONTACT WITH THE FLOOR SURFACE, UP TO THREE POWER SUPPLIES MAY BE HINGED OUT AT ONE TIME. IF THE LEVELLERS ARE NOT DOWN, AND THREE POWER SUPPLIES ARE HINGED OUT, THE RACK WILL FALL OVER DUE TO THE WEIGHT OF THE POWER SUPPLIES.

34-017 and 34-020 Power Supplies. The 34-017 and 34-020 supplies attach to the mounting upright via four 10-32 X $\frac{1}{2}$ Lg PHPS screws. See Figure 9.

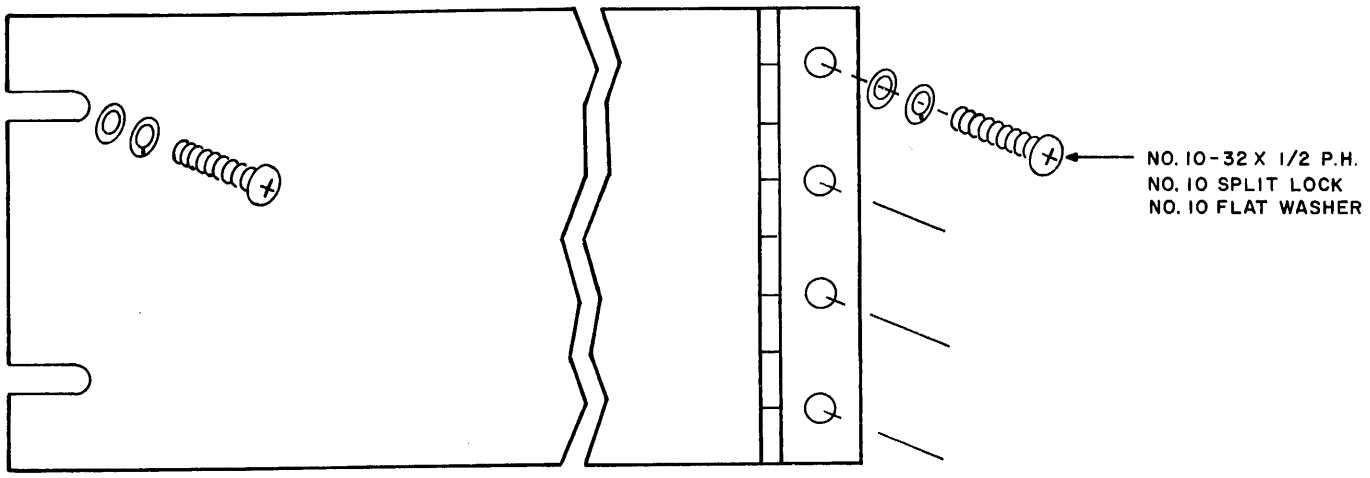


Figure 9. 34-017 and 34-020 Power Supply Mounting

When either power supply is in the installed operating position, it is secured to the left rear upright by two 10-32 screws. The power supply cable connects to terminal lugs at the right rear (looking from rear) of its respective Processor or Expansion Chassis via faston lugs and a connector for fan AC power. Refer to Figure 10. There is adequate slack provided in the cable to allow the Power Supply to hinge out freely. In order to prevent the cable from being pinched between the Power Supply and the Chassis Support Rails, a service loop is required. A maximum of five power supplies may be mounted in one rack.

WARNING

ALL AC FAN CONNECTORS ON POWER SUPPLIES WHICH ARE NOT CONNECTED TO MATING RECEPTACLES MUST REMAIN COVERED OR SHORTING MAY OCCUR. SEE FIGURE 12.

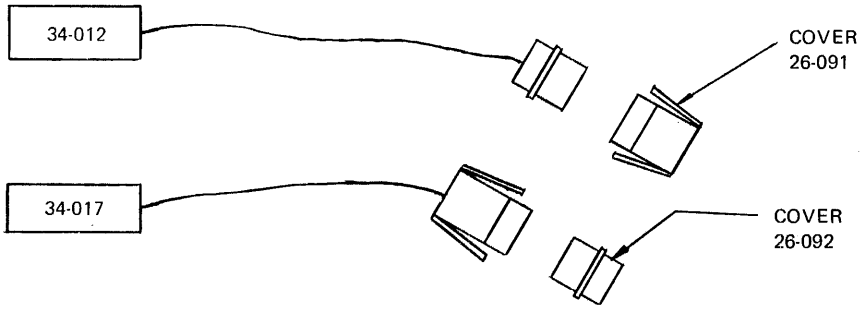


Figure 10. Fan Connector Caps

4.1 New Power Supply

INTERDATA now has several new Power Supplies available for use with its line of digital systems. These supplies are manufactured by INTERDATA, and may replace those previously used. This section provides information on these new supplies, and on several associated changes to INTERDATA systems.

There are three power supplies available:

Model Number 34-012 Power Supply. The 34-012 is the original power supply. This unit has a transformer that always supplies 115V for fan power at both 115V and 230V strappings. It supplies chassis fan power with a male connector (now discontinued).

Model Number 34-017 Power Supply. The 34-017 is the 25 ampere version of the INTERDATA supply. This unit supplies 115V or 230V depending on AC source for fan power. Chassis fan power is supplied with a U. L. approved female connector.

Model Number 34-020 Power Supply. The 34-020 is the 50 ampere version of the INTERDATA supply. The fan power consideration and the output connector are the same as for the 34-017 unit.

Two types of fan jumper cables are provided. Cable 17-181 is wired with two male connectors, while cable 17-287 is wired with one male and one female connector.

Because the newer power supplies supply line power for the fans, it is necessary to re-wire the fans in the chassis to operate with 230VAC. The New Chassis provides an AC power switch for switching fan power to either voltage. When using the newer supplies (34-017 or 34-020) with the former chassis, only 115V operation is possible, unless the chassis is modified.

4.2 Configuration Data

The 34-012 may be replaced in the following ways.

25 AMP	34-012 (1 each)	Replaced by 34-017 (1 each)
50 AMP	34-012 (2 each)	Replaced by 34-020 (1 each)

The following paragraphs outline the factors involved in using different combinations of chassis, power supplies, and cables.

1. The 34-012 with the "Former Chassis".

   This unit is used for 115/230V operation. The "Former Chassis" (1 or 2 fan) is wired for 115V operation and equipped with two female connectors for fan power. Refer to Figure 11.

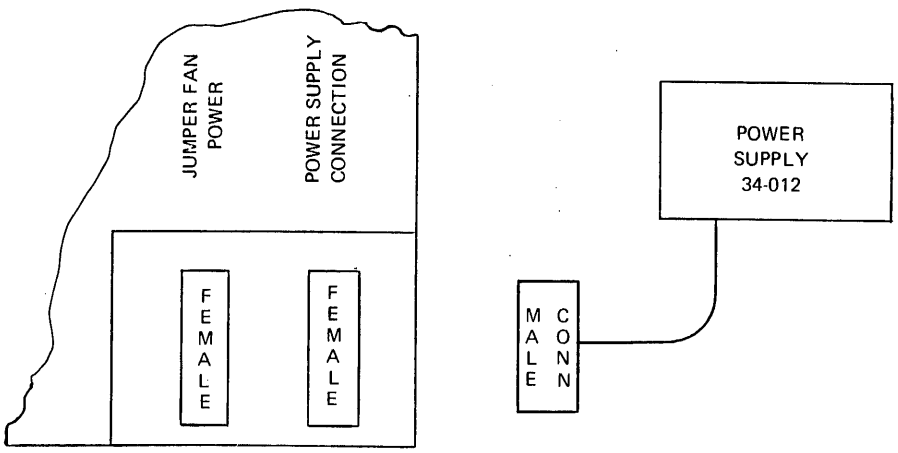


Figure 11. Former Chassis

2. The 34-012 Power Supply with a "New Chassis".

This unit is used for 115/230V operation, however, the 115/230V fan switch on the "new chassis" must remain in the 115V position for 115V or 230V operation. If two or more "new chassis" are powered, standard fan jumper cable, 17-181, must be replaced by cable 17-287.

NOTE

The "new chassis" (1 or 2 fan) is wired for 115/230V fan operation and equipped with one male and one female connector for fan power. See Figure 12.

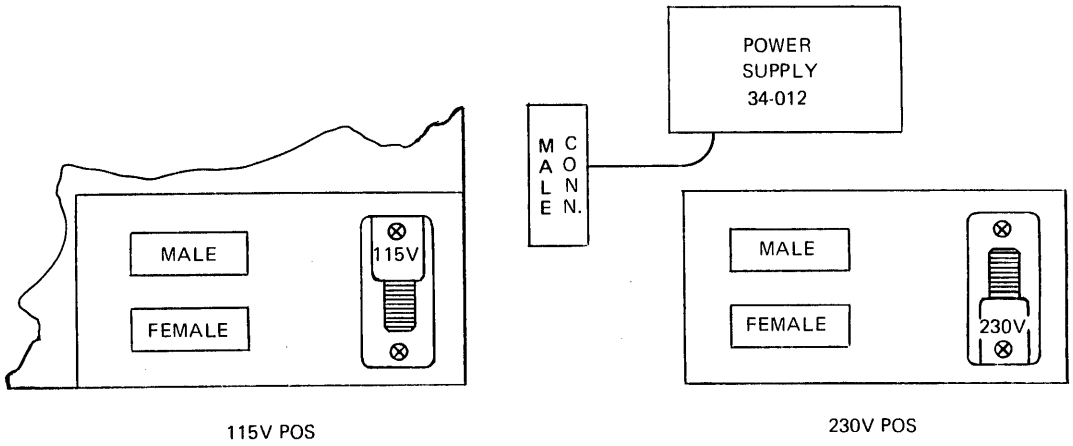


Figure 12. New Chassis

3. The 34-017 Power Supply with a "New Chassis".

   This unit is used for 115/230V operation, however, the 115/230V fan switch on the chassis must be matched with the 115V or 230V strapping in the Power Supply.

4. The 34-017 Power Supply with a "Former Chassis".

   This unit is used for 115V operation (ONLY) using cable 17-181.

NOTE

The "former chassis" must be rewired for 230V operation.
Kits 39-020F01 (1 fan) and 39-020F02 (2 fans) must be used
to convert a chassis wired for 115V to 115/230V fan operation.

4.3 Exhaust Fans

New exhaust fan plates are equipped with a switch to provide either 115 or 230VAC operation. See Figure 13.

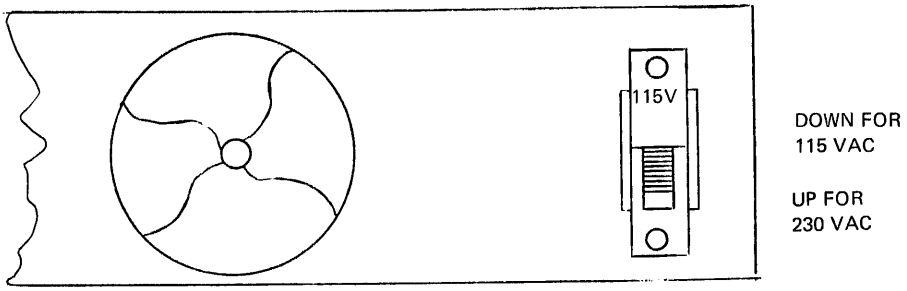


Figure 13. Fan Switch Setting

5. DISPLAY PANEL INSTALLATION

The optional Model 7/16 HSALU Hexadecimal Display Panel is electrically tied to the Processor via one connector and seven Faston lugs. The connector is installed on Connector 3 of the 35-522 CPU-A board and the seven terminal lugs mate into a terminal strip on the left side of the Processor Chassis. The terminal lugs are identified at the Faston connector and are mated to their corresponding terminal pin (C1, C2, etc). on the chassis. See Figure 5.

The Hexadecimal Display Panel is physically mounted to the brackets provided on the Processor Chassis. The 1 3/4 inch Filler Panel is mounted directly below the Hexadecimal Display Panel on this same chassis. Refer to Figure 6.

6. TURNKEY CONSOLE PANEL INSTALLATION

The Turnkey Console is connected to the Processor in the same manner as the Hexadecimal Display Panel discussed previously. Only two Faston connectors are provided with this assembly, but their installation is the same.

The panel on which the switches are installed may be mounted in the same manner as the Display Panel. The Hexadecimal Display Panel option, the Automatic Loader option, and the Turnkey Console option may not be installed together on the same Processor.

7. MEMORY INSTALLATION AND EXPANSION

The first memory module must be installed in Slot 3 of the Processor card file. Further memory expansions are installed in adjacent descending slots in this chassis. The first memory module installed in an Expansion chassis must be installed in Slot 7 of that chassis. This is necessary to insure proper temperature tracking characteristics for the memory voltages from the Expansion Chassis Power Supply. The additional modules are then installed in one of the slots pre-wired to accept that module, for module assignments. When memory is installed in an Expansion chassis, the 1K ohm resistor between TEMA and TEMB at the back panel must be removed. If for any reason the memory module installed in Slot 7 is removed (see Figure 14), the resistor must be replaced before applying power. The 35-533 parity option card must be installed on the wire wrap side of Slot 3 (Figure 14), Connector 1 of the Processor back panel in order for the Processor to detect parity errors.

NOTE

Parity and non-parity memories may not be mixed in a system.

8. PRIMARY POWER FAIL/AUTO-RESTART INSTALLATION

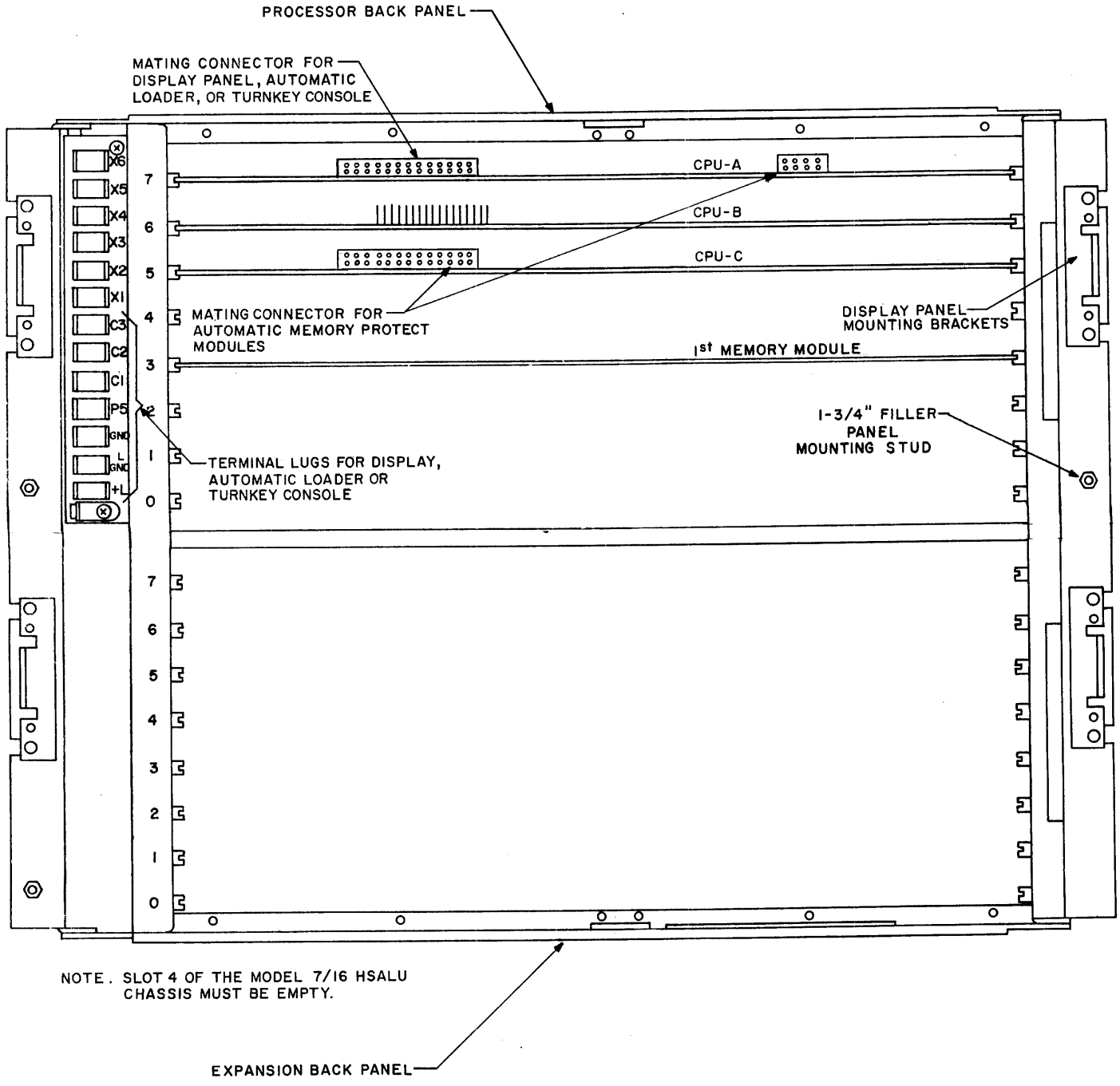
Install the 35-448 logic card for the Primary Power Fail/Auto-Restart option on the wire wrap side of the Processor chassis at Slot 7 (Figure 14). Connector 0 with the apparatus side up. The 17-182F01 and 17-182F02, cables which supply 12 VAC to the logic card, connect between C1 on the logic card and C1 on the back panel and C3 on the logic card and C3 on the back panel as indicated on the cables. On the 35-522 CPU-A board remoev the jumper between TP4 and TP5. The primary power fail option card is adjusted at the factory.

9. CONFIGURATION

9.1 System Expansion Chassis

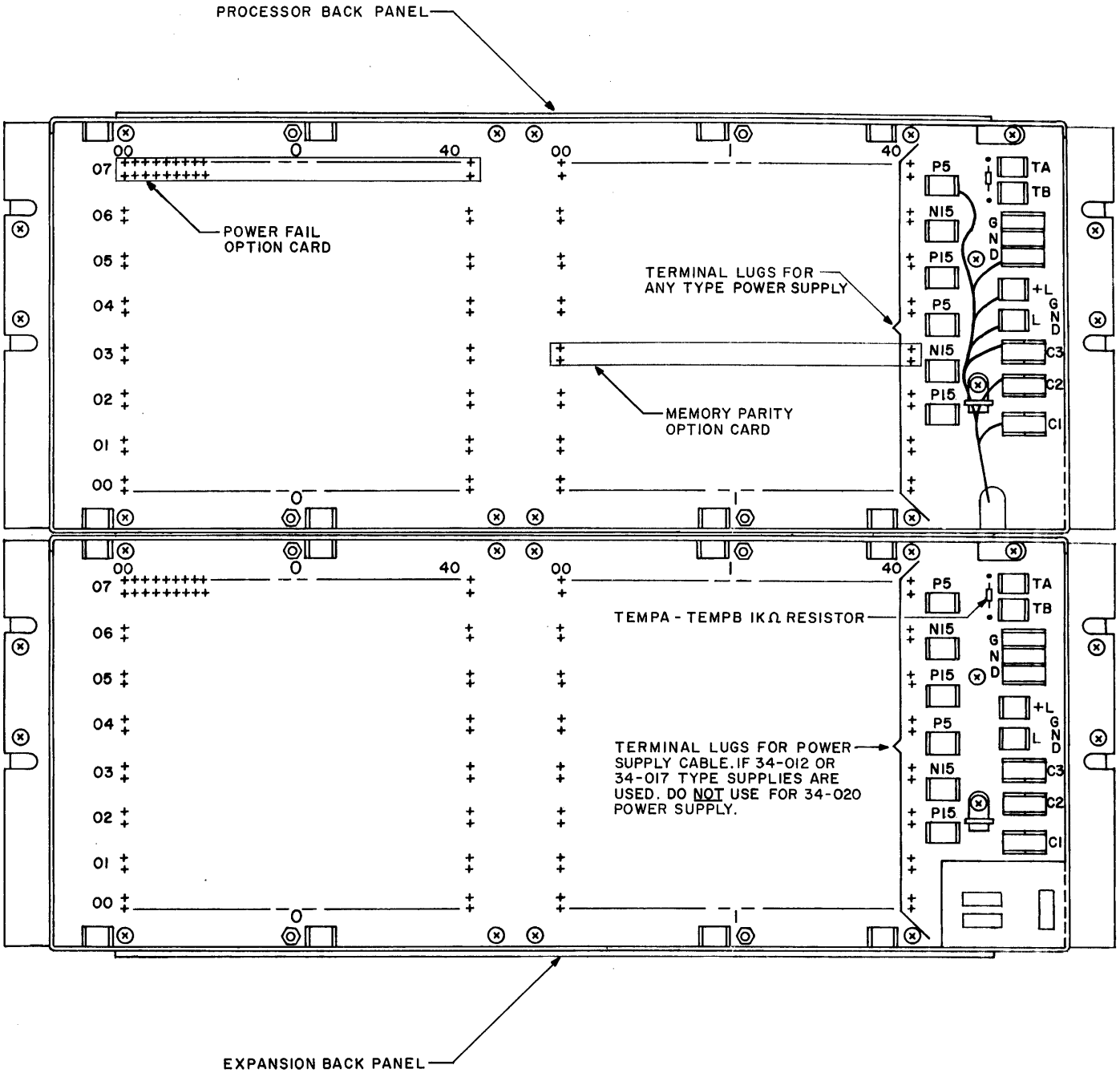
When configuring a multi-chassis system there are four rules that must be followed:

1. The system Expansion Chassis must be mounted below the basic Processor Chassis.
2. All chassis must be contiguous.
3. All 15 inch system expansion chassis must be mounted above any 10 inch system Expansion Chassis.
4. Multiboard peripheral device controllers (on 10 inch circuit boards) can only be used in the 10 inch system Expansion Chassis.



Front View

Figure 14. View of Model 7/16 HSALU Twin Chassis



Rear View

Figure 14. View of Model 7/16 HSALU Twin Chassis (Continued)

9.2 Circuit Board Distribution

Model 7/16 HSALU Digital Systems may be configured in a variety of ways. However, the following factors must be considered when determining circuit board distribution within the basic Processor and the system Expansion Chassis. See Figure 15.

1. The Selector Channel can be placed in Slot 0 of the Processor back panel, or Slots 6, 4, 2, or 0 of the system Expansion back panel.

2. All slots on Connector 1 below the position where the SELCH is inserted become SELCH Bus slots. (This only applies within the back panel containing the SELCH.) The SELCH Bus extends down the right side connectors (left view). Note that all device controllers on 10" adapter boards connect to the Multiplexor Bus from the left side connectors (rear view). Therefore, these device controllers may be inserted in vacant SELCH Bus slots, but will not be on the SELCH Bus. This also applies to all 7" boards on adapters, installed on the left side.

3. The SELCH Bus can be extended by cable to any even numbered slot in an I/O chassis adjacent to the chassis containing the SELCH controller.

4. All device addresses are hard-wired on the device controller cards, (device addresses may be changed at option) so that the distribution of I/O device controllers in the chassis normally need be considered as a matter of priority in the RACK0/TACK0 "daisy-chain" and convenience.

5. Slots 3, 2, 1 and 0 of the Processor back panel and all slots of the universal Expansion back panel are pre-wired with 8KB memory module addresses for up to 64KB. It is mandatory that Slot 3 of the Processor Chassis be used for memory. If memory is installed in an Expansion Chassis it is mandatory that one module is installed in Slot 7.

6. The 15 inch system Expansion Chassis, and the basic Processor Chassis may only be used for single board I/O device controllers unless the interconnection between boards takes place via cables installed on the other edge of the board. For multi-board 10 inch device controllers, the 10 inch system Expansion Chassis must be used.

7. The interrupt priority of a given device controlled is determined by its physical location on the serial RACK0/TACK0 line. Refer to 9.3 Interrupt Priority Back Panel Wiring to determine which physical location has what priority. When deciding which devices should have a higher or lower priority, devices that must be serviced in a certain amount of time or loss of data access should be given a higher priority than a device that with a high interrupt data and no data loss if not serviced.

9.3 Interrupt Priority Back Panel Wiring

The Acknowledge Control line from the Processor carries the Interrupt Acknowledge (ACK) signal. This line breaks up into a series of short lines to form the "daisy-chain" priority system. The ACK signal must pass through every controller that is equipped with Interrupt Control circuits.

Back panel wiring for interrupt control at a given position is: The Received ACK (RACK0) at Pin 122-1 or 0 and the Transmitted ACK (TACK0) at Pin 222-1 or 0. The daisy-chain bus is formed by a series of isolated lines which connect Terminal 222-1 or 0 of a given position to Terminal 122-1 or 0 of the next position (lower priority). On unequipped positions, a jumper shorts 122-1 or 0 and 222-1 or 0 of the same connector to complete the bus. Back panels are wired with jumpers on all positions. Whenever a card chassis position is equipped with a controller that has an interrupt capability, the jumper from 122-1 or 0 and 222-1 or 0 must be removed from the back panel at that position.

Figure 16, showing the standard interrupt priority wiring, assumes a Model 7/16 HSALU Processor and one memory module. The arrows indicate the direction of priority from the highest priority to the lowest. By changing the wires, crossing from Side 0 to Side 1 of the Processor and/or Expansion panels, interrupt priorities may be rearranged. An example of this is shown in Figure 17, Modified Interrupt Priority. Slot 3 of the Side 1 of the Processor panel has the highest priority and the Serial Port of the Expansion panel has the lowest priority. When Selector Channels (SELCH) are installed, the standard interrupt priority must be modified. Refer to Figure 18, Interrupt Priority with SELCH Installed. Note that if it is decided to install a SELCH in Slot 0 of the Processor panel or Slot 6 of the Expansion panel it is necessary to remove the wires connecting the Processor and Exapnsion panels together in addition to the wires called out by the SELCH

installation procedure. The Multiplexor Bus may be restored to the remaining slots on Side 1 that are not used for the private SELCH Bus. To prevent this, install the SELCH on the bus buffer as low as possible in the chassis. This may be accomplished by the use of a 17-182 cable. This same situation exists when installing a bus buffer into the chassis.

For controllers that occupy several positions, the jumper is removed only at the position where the controller board has ATN/ACK circuits. For details on the various devices, see the appropriate installation specification.

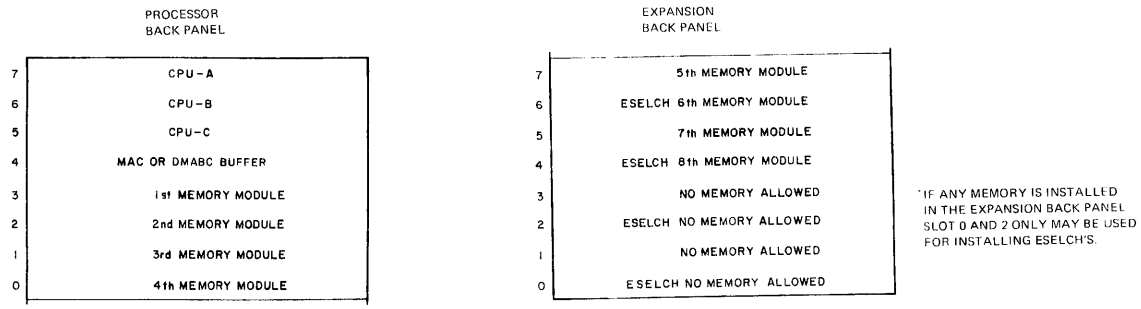


Figure 15. Model 7/32 Memory Configuration

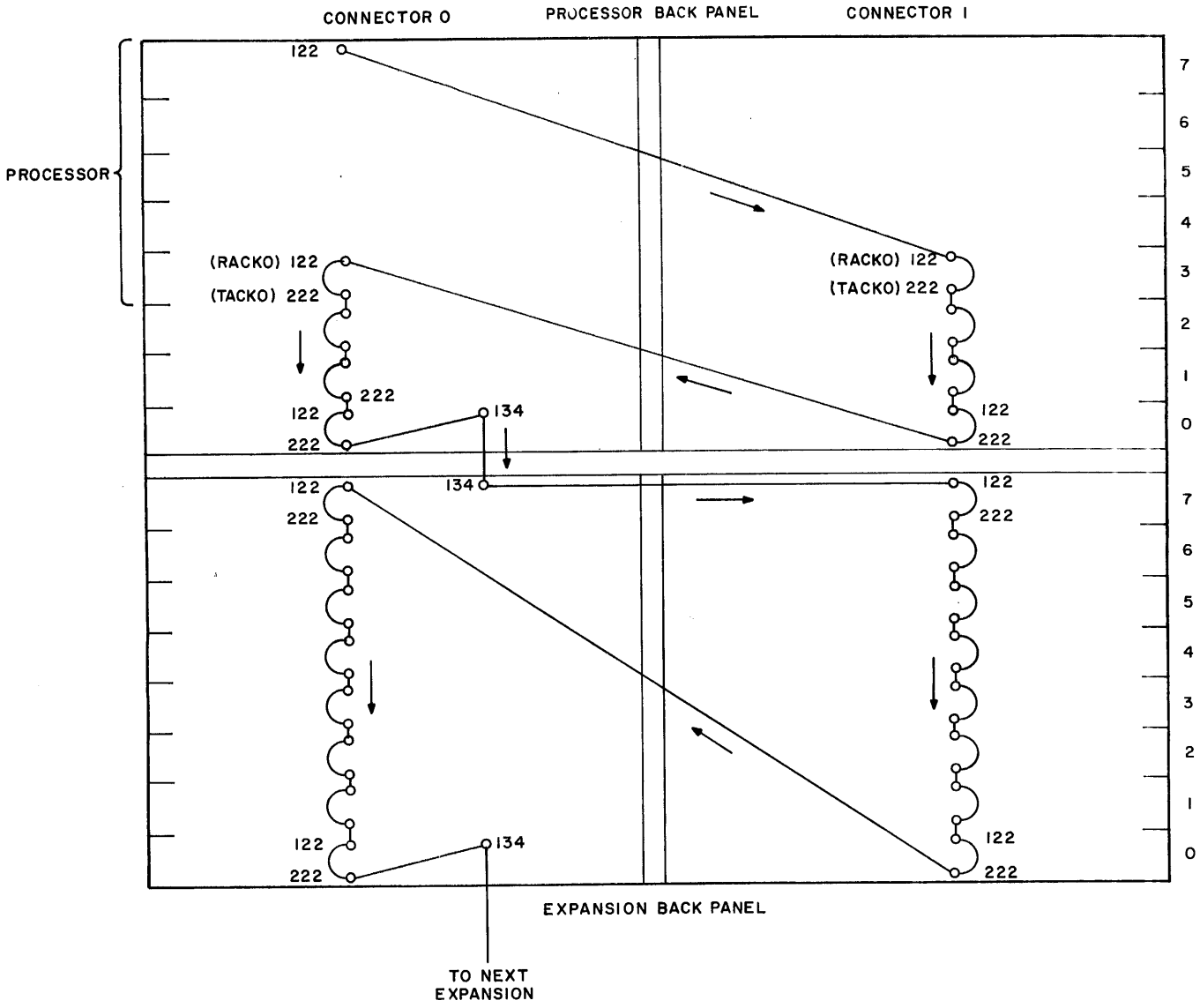


Figure 16. View of Model 7/16 HSALU Twin Chassis

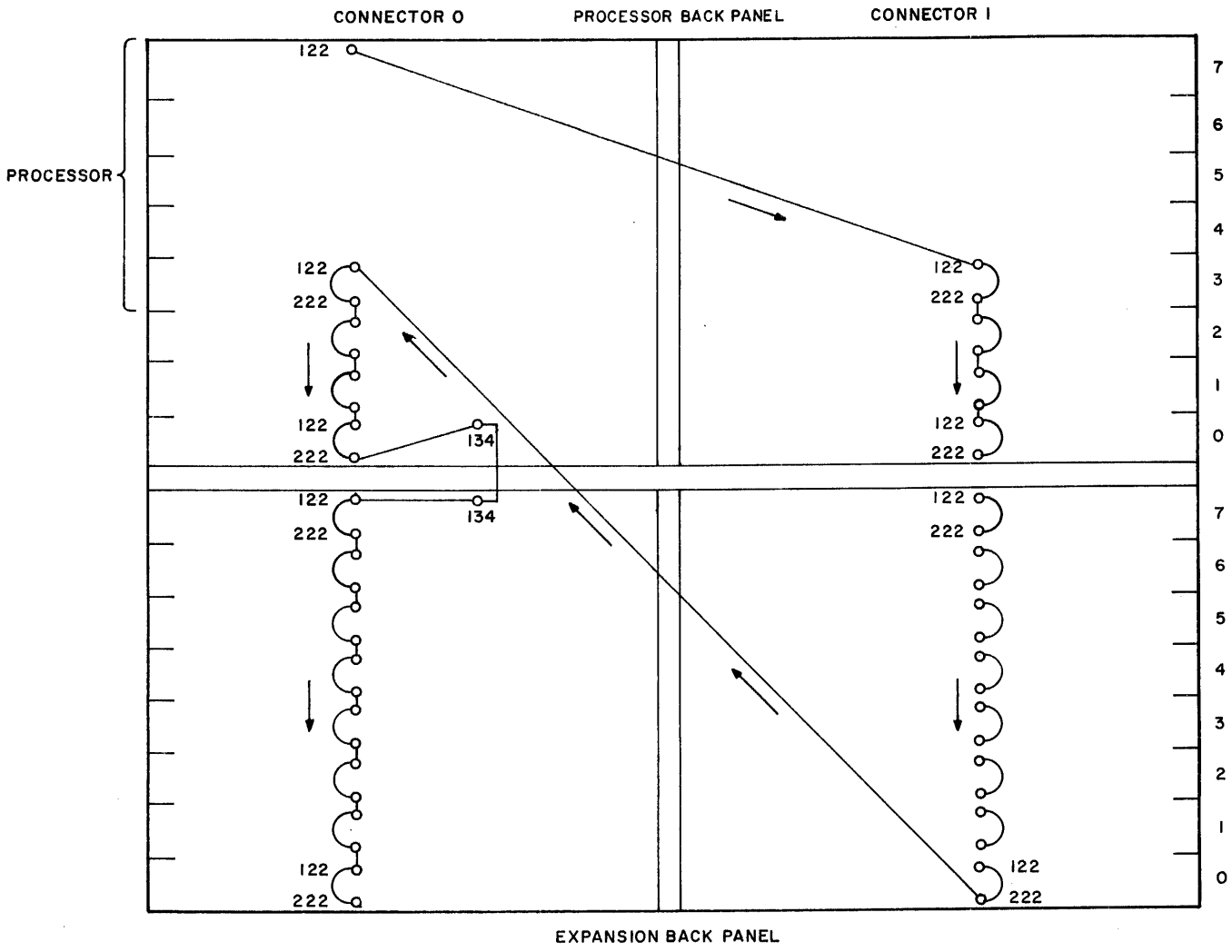


Figure 17. View of Model 7/16 HSALU Twin Chassis (Continued)

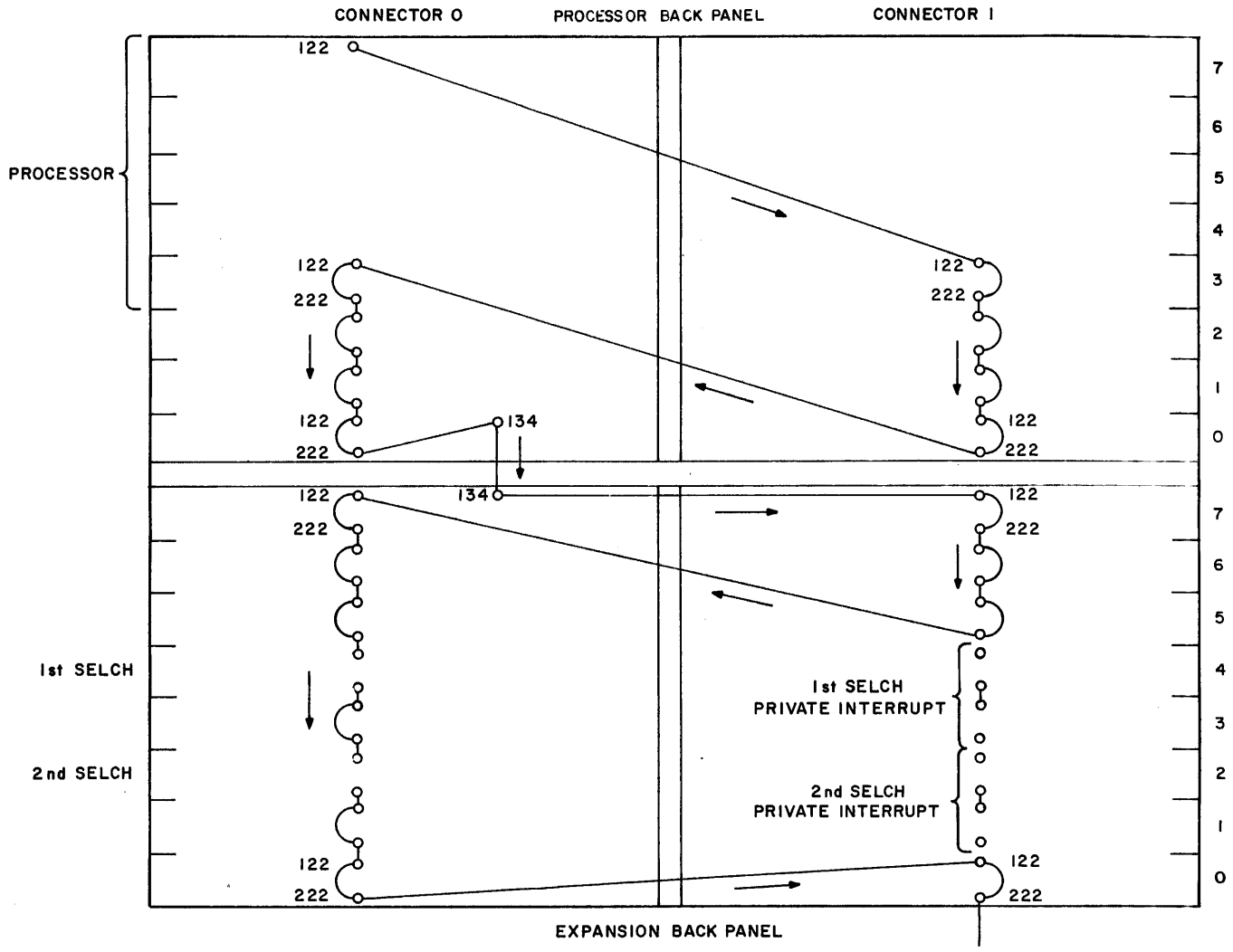


Figure 18. Interrupt Priority With SELCH Installed

9.4 System Configuration

System configuration data is provided in the User's Manual, Publication Number 29-261.

9.5 Terminators

The termination end of both legs, Connector 0 and 1, of the Multiplexor Bus must have a standard INTERDATA termination card (35-433) installed. These cards are installed, on the back panel at the lowest numbered slot of both connectors on the Multiplexor Bus that exists, e.g., If a Selector Channel or bus buffer is installed in Slot 4 on the first expansion chassis and only the Processor Chassis and one Expansion Chassis is used in the system, the Multiplexor Bus must be terminated at Slot 0, Connector 0, and Slot 5, Connector 1 of the Expansion Chassis. In addition, the buffered bus or the SELCH Bus should be terminated at Slot 0, Connector 1 of this chassis.

Depending upon system configuration, any SELCH Bus or buffered bus may be terminated by a 15" Terminator (35-433) or a 10" Terminator (35-434). The choice of terminators depends on the type of chassis in which the last slot of the bus is present.

10. CABLES

10.1 Power Cable

The standard INTERDATA Cabinet is wired for 30 Ampere service. On the main power cable (part of the AC Distribution Panel), there is a three wire, twist lock, grounding, 125VAC, 30 Ampere, UL, (Hubble #2610) plug. A three wire, grounding, 30 Ampere, 125 VAC receptacle (Hubbel #2611 or equivalent) is required to accept this plug.

10.2 System Expansion Cable

A number of standard cables are available for configuring systems made up of the INTERDATA Expansion Chassis discussed in Section 4. The choice of cables is dependent upon system configuration. The following cables are available:

1. 17-162 and 17-163: I/O and Memory Expansion Cables (see note)

The 17-162 cable is used to connect the "0" connector field and the 17-163 cable is used to connect the "1" connector field from a 7" Processor chassis to the corresponding connector in the first 15 inch expansion file. The expansion file must be mounted immediately below the basic Processor as these cables contain the memory bus which is restricted to the first 15 inch expansion only.

These cables are always used in pairs.

2. 17-193: I/O Expansion Cable, Connector "0"

This cable is used to connect the "0" connector field between two adjacent 15" card files.

3. 17-194: I/O Expansion Cable (see note)

This cable is used to connect the "1" connector I/O fields between two adjacent 15" card files.

4. 17-216: I/O Expansion Cable, 36 Inch Long

This is a 36" long cable. It can be used to connect two 15" files that are not adjacent.

It must not be used to extend the basic Processor Mutiplexor Bus.

It can be used to extend a buffered bus or a SELCH Bus. It plugs into a "1" side connector. The "receiving" end can plug into the "0" or "1" side of the expansion file.

5. 17-214: 15" to 10" Expansion Cable

This cable is used to connect the "0" connector field of a 15" card file to a lower adjacent 10" card file. It provides an 8 bit I/O bus to the 10" card file.

6. 17-166: 15" to 10" I/O Expansion Cable, 36 Inch Long

This cable is used to connect the "1" side of a 15" expansion file to a 10" expansion file. It provides an 8 bit I/O bus to a 10" card file.

It must not be connected to the basic Processor Multiplexor Bus.

It may be driven either by a Selector Channel or a bus buffer.

Can be used on the older 10" card file (13 I/O slot).

7. 17-183: "0" to "1" Connector

This cable can be used to interconnect the "0" field and the "1" field within a 15" card file.

It can also be used to connect a "0" side (Slot 0) of a file, to the "1" side (Slot 7) of the next adjacent file, or vice versa.

8. 17-215: 10" to 10" I/O Expansion Cable

This cable is used to connect two adjacent 10" card files.

NOTE

On the receive end of either a 17-163 or 17-194 cable a strap is installed in the factory. This strap must be removed unless the cable is being used to jumper a private I/O Bus (SELCH or bus buffer). This strap jumpers Pin 222-0001 of the upper chassis to Pin 122-0701 of the first expansion chassis. If these cables are used to extend a SELCH or bus buffer the following wiring changes are required on the lower chassis:

Remove the strap from Pin 134-0700 to Pin 122-0701

Add the strap from Pin 134-0700 to Pin 122-0700

M-71 SERIES MODEL 7/16 HSALU
M-73 SERIES MODEL 7/32
MAINTENANCE SPECIFICATION

1. INTRODUCTION

A family of Processors are represented by the 7/16 and 7/32 model numbers. This family covers the entire spectrum of performance levels from the 16-bit basic, Model 7/16 Basic, to the faster and more powerful 16-bit Model 7/16 HSALU (High Speed Arithmetic Logic Unit) and finally to the maxi-mini Model 7/32 which is a 32 bit Processor with the capability of directly addressing one million bytes of main memory.

Each machine is a fourth generation Processor suitable for use in data communications, process control, or stand-alone scientific applications. These Processors are modularly constructed for ease of maintenance and are compatible with all building blocks in the INTERDATA product line. In addition, field upgrading is available (as the application grows) from any machine to a higher performance Processor.

2. SCOPE

This specification describes the functional operation of both the Model 7/16 HSALU and the Model 7/32 Processors and provides maintenance information useful to the digital technician in maintaining these Processors. A block diagram analysis, a micro-program description, and functional analysis of major Processor areas are included.

3. BLOCK DIAGRAM ANALYSIS

Refer to the Block Diagram in Figure 1.

3.1 System Organization

The Model 7/16 HSALU and 7/32 Processors are organized between two 16-bit buses. The B Bus is used to present data to the Arithmetic Logic Unit (ALU). The S Bus then transfers the ALU output to the appropriate destination. The source and destination of data on the B Bus and S Bus, as well as the function performed by the ALU is controlled by micro-instructions contained in the Read-Only-Memory (ROM). In the 7/32 Processor an extention of four bits as appended to the B and S Busses which is used for address manipulation by the micro-program. This extention provides a 20-bit path in the machine for calculating an address.

3.2 Read-Only-Memory (ROM)

The Read-Only-Memory is a high speed, solid-state, non-destructive memory organized into five, six or seven pages of 256 words each depending on the Processor. Each word in ROM is 24 bits long and represents one micro-instruction. Each micro-instruction read out of ROM is placed in the 24 bit ROM Data Register (RD). RD is the Instruction Register for the micro-Processor. Most micro-instructions are executed in one machine cycle of 250 nanoseconds. RD bits are decoded to select a Source to be statically unloaded to the B Bus. The ALU then forms a result on the S Bus. This result becomes available some time before the end of the machine cycle; at the start of the next machine cycle the appropriate destination register is loaded and the next micro-instruction is fetched. The meaning of the micro-instruction word bits is explained later.

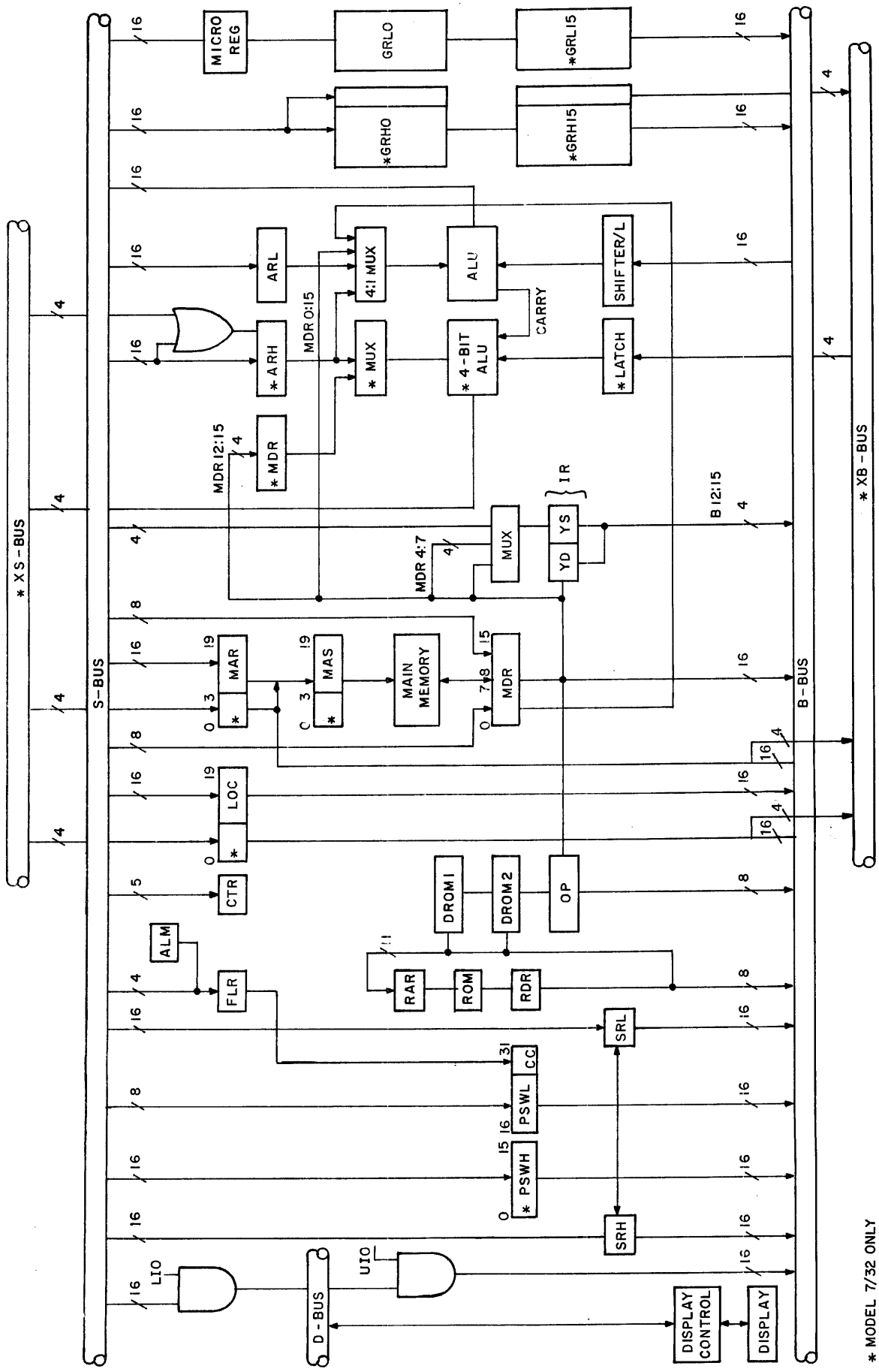


Figure 1. Block Diagram, 7/16 HSALU and 7/32 PROCESSOR

Locations in the ROM are addressed by the 11 bit ROM Address Register (RAR). Micro-instructions are normally located at sequential addresses in the ROM. The RAR is an up/down counter which increments by one as each new micro-instruction is read into RD. The RAR therefore holds the address of the next micro-instruction to be executed. When it becomes necessary to jump out of sequence, RAR can be loaded with a new address from the RD register, from the Decoded Read-Only-Memories, or it can be preset by the hardware.

3.3 Flag Register (FLR)

The Flag Register (FLR) is a four-bit register containing the following flags: Carry (C), Overflow (V), Greater than Zero (G), and Less than Zero (L). These flags are modified at the conclusion of arithmetic and logical micro-operations to reflect the result of the operation. The FLR is loaded from Bits 12 through 15 of the S Bus when either the FLR or the Program Status Word (PSW) is the specified Destination Register. The Flag Register may also be loaded from the Alarm Register (ALM). The ALM is a three bit register which is used to indicate the following: Parity Fail on a Data Read, Parity Fail on an Instruction Fetch and an Early Power Failure.

3.4 Program Status Word (PSW)

The Model 7/16 HSALU Program Status Word (PSW) is a 16-bit register used to indicate the system status relative to the user program being emulated. Bits 0 through 11 of the PSW define enabled interrupts and the operational status or mode of the user level Processor. Some of the PSW bits have hardware significance while others are of significance only to the micro-program. Bits 12 through 15 of the PSW make up the Condition Code field (CC) which reflects the result of the previous user instruction.

The Status portion of the Model 7/32 PSW is 32 bits long. Only 17 bits, however, are implemented in the hardware of this machine. For this Processor, Bit 11 and Bits 16 through 27 represent the present state of the machine and Bits 28 through 31 make up the Condition Code field which reflects the result of the previous user instruction.

The Condition Code may only be updated from the FLR. When PSW is the specified Destination Register, Bits 0 through 11 of the S Bus are loaded into Bits 0 through 11 of the PSW and S Bus Bits 12 through 15 are captured in the FLR. The Condition Code field remains unchanged until the micro-program causes it to be updated from the FLR or when the hardware, in the case of the instruction read, copies the contents of the FLR into the Condition Code.

The Location Counter (LOC) is a 16-bit (20-bit for the Model 7/32) appendum to PSW which holds the main memory address of the next user instruction to be performed.

3.5 Main Memory

The Main Memory consists of random access memory providing storage for user instructions and data. The Memory Address Register (MAR) is a 16-bit register (20 bit for the Model 7/32) which is loaded with the address of main memory locations. Memory is actually addressed by the Memory Access Slave Register (MAS). MAS is automatically updated from MAR at the start of each memory cycle. Data read from or written into memory is buffered in the Memory Data Register.(MDR). The micro-program initiates a main memory cycle by issuing a memory read, memory write, privileged write, or instruction read command. After issuing a memory command, the micro-program is free to do other instructions. The memory cycle is accomplished asynchronous of other Processor activity. If the micro-program, however, attempts to use the contents of MDR after a memory read or instruction read before memory data becomes available, or attempts to load MDR or issue another memory command before the current memory cycle is complete, the Processor stops until the desired function can be performed.

With core memory, a memory cycle consists of two phases. First, the contents of the specified location are read out and placed into the MDR and replaced by zeros (destructive read-out). The contents of MDR are then written into the specified location. A memory read consists of a read cycle that saves the contents of the specified location in MDR. The contents of the MDR are then written back to the specified location on the write cycle. A memory write does not save the read-out so that the specified location is written with the contents of MDR. An instruction read differs from a memory read in that after the data becomes available in MDR, it is automatically transferred to the user's Instruction Register.

3.6 Instruction Register (IR)

After an instruction read has been issued, when the read-out is available in MDR, MDR Bits 0 through 7 are placed in the register labeled OP, Bits 8 through 11 are placed in the register labeled YD, and Bits 12 through 15 are placed in the register labeled YS. These three registers (OP, YD, and YS) comprise the user's Instruction Register.

3.7 Decoder Read-Only Memory (DROM)

The OP Register is used to address locations in the pair of Decoder Read-Only-Memories (DROM1 and DROM2). Two DROM1s are used in the 7/32. Each DROM contains 256 12-bit words. The micro-program can interrogate either DROM1 or DROM2 at anytime other than on a Branch or Input/Output micro-instruction. The least significant 11 bits of the resulting read-out are jammed into the RAR, resulting in an automatic branch to an address that is related to the user's operation code. Bit 4 of DROM1 is used to indicate privileged instructions.

3.8 General Registers (GR)

It is most often the case that the micro-program accesses the user's General Registers without caring which of the 16 General Registers it gets. It matters that when the micro-program accesses a General Register for emulating a user instruction that it be the General Register specified in that user instruction. Since after instruction read, the register address or addresses specified by the user are in the YD and YS Register, the micro-program can access the appropriate General Register by specifying the YD or YS Instruction Register. The hardware then selects the General Register whose number is in the YS or YD Register.

The YD Register is an up/down counter so that sequential General Registers can be accessed. The micro-program can also clear the YD Register when it needs to access specific General Registers. The YS Register, in addition to being loaded at the time of an instruction fetch, may also be loaded by the micro-program from the S Bus.

In the Model 7/16, sixteen 16-bit General Registers are provided for the user programs while in the 7/32 two sets of sixteen 32-bit General Registers are provided. These registers are labeled GRH0 and GRL0 which is Set 0 and GRH15 and GRL15 which is Set 15.

3.9 Micro-Registers (MR)

The seven 16-bit registers MR0 through MR6 are available to the micro-program for general purpose use.

3.10 Arithmetic Registers High and Low (ARH and ARL)

Two 16-bit A Registers are used to hold the second operand for arithmetic and logical micro-operations. These registers plus the Memory Data Register (MDR) and the sign of MDR, Bit 0, are used as the 'B' input to the Arithmetic Logic Unit (ALU). The other input, input 'A', to the ALU is the output of the Shifter/Latch. The Shifter can Shift B Bus data left or right one bit position, do an eight-bit rotate, or gate the B Bus data directly into the ALU.

3.11 Arithmetic Logic Unit (ALU)

The ALU comprises a 16-bit parallel arithmetic/logic network with look ahead carry. The arithmetic or logical result is formed on the 16-bit S Bus. On the Model 7/32 a four bit extention to the ALU is used which forms its result on the four bit XS Bus.

3.12 Input/Output (I/O)

Input/Output operations are achieved by gating S Bus data onto the D Bus and activating an I/O Control line, or by activating an I/O Control line and gating the D Bus data onto the B Bus.

3.13 Shift Register High (SRH) and Shift Register Low (SRL)

These two registers, SRH and SRL, are 16-bit general purpose micro-registers which can also be used in combination as a 32-bit shift register. The extended shift capability is used in multiply, divide, and other 32-bit operations.

3.14 Counter

The Counter (CTR) is a five-bit decrementing register used on multiply, divide, repeat operations, and on the Branch on Counter micro-instruction. The counter is loaded by the micro-program from the S Bus with any value from 0 to 31. The state of this register, zero or non-zero, may be tested by the micro-program when a Branch on Counter is specified by the instruction.

4. MICRO PROGRAM DESCRIPTION

4.1 Introduction

Micro-programming is a means for implementing the control logic of a digital computer. At INTERDATA, micro-programming has been effectively used to maintain upward compatibility in a family of Processors whose internal hardware varies from one member to the next.

The Models 7/16 HSALU and 7/32 are designed to execute micro-instructions stored in a Read-Only Memory (ROM). Each micro-instruction causes one or more hardware functions to be performed, such as transferring the contents of one register to another, arithmetic or Boolean operations between registers, controlling input/output operations, or initiating main memory accesses.

A series of micro-instructions is called a micro-program. The complete micro-program for these machines are by definition, an emulator, causing the hardware to react to a user program in main memory and to external events as would the Processor described in the User's Manual, Publication Number 29-261 or 32 Bit Series Reference Manual, Publication Number 29-365. Every user instruction, interrupt handling feature, and Hexadecimal Display Panel function is simulated by some portion of the micro-program.

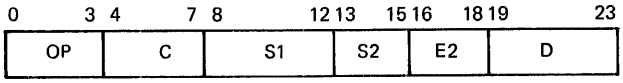
This section describes the micro-program for the Model 7/16 HSALU and 7/32 Processors. The microprocessor of the two machines are identical but the compliment of hardware available in these two Processors vary. For example; all references to 20 bit registers in this description are implemented as 16 bit registers in the Model 7/16 HSALU. In addition, only one set of 16 bit General Registers is equipped on the 7/16 while two sets of 32 bit General Registers are available on the Model 7/32.

4.2 Micro-Instruction Formats and Modifiers

Micro-instructions for the Models 7/16 HSALU and 7/32 have fixed length but their format is variable. All micro-instructions are 24 bits long and are highly encoded. The Processor has six basic micro-instruction formats. The operation code is found in the first four bits of every instruction format, but the interpretation of other bits varies from one format to the other.

4.2.1 Register to Retister (RR) Format.

RR FORMAT



OP	Operation Code
C	Control Modifier
S1	First Source Operand
S2	Second Source Operand
D	Destination Register
E2	Opcode Extension Two Modifiers

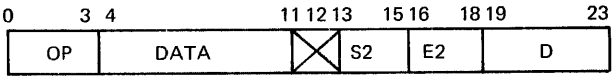
The following six micro-instructions use this RR format.

AND	(N)
OR	(O)
XOR	(X)
ADD	(A)
SUBTRACT	(S)
CALCULATE ADDRESS	(CA)

All instructions with this format contain a four bit operation code, a four bit control modifier, a five bit first operand specification, a three bit second source specification, a five bit destination specification, and three bit opcode extension modifiers. These six micro-instructions are described in Section 4.3, and a detailed description of instruction modifiers appears in Section 4.2.8. Section 4.2.7 describes all addressable sources and destinations.

4.2.2 Register Immediate (RI) Format.

RI FORMAT



OP	Operation Code
DATA	The least significant eight bits of the first operand.
S2	Second Source
D	Destination Register
E2	Opcode Extension Two Modifiers

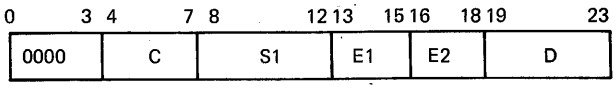
There are five micro-instructions with this format.

AND Immediate	(NI)
OR Immediate	(OI)
Exclusive OR Immediate	(XI)
Add Immediate	(AI)
Subtract Immediate	(SI)

All of the previous instructions contain a four bit operation code and eight bits of immediate data. The eight bit immediate field is expanded to a 20 bit value with the high order 12 bits forced to zero. This 20 bit value forms the first operand. The second operand source and the destination register are specified by the S2 and D fields. The E2 field specifies opcode extension modifiers (see Section 4.2.8). Bit 12 is not used in this format.

4.2.3 Load (L) Format.

L FORMAT



C Control Field

S1 First Operand

D Destination Register

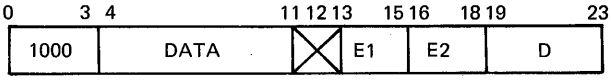
E2 Opcode Extension Two Modifiers

E1 Opcode Extension One Modifier

This format is used only by the Load micro-instruction. This micro-instruction does not require any second operand. Bits 13 through 15 of the micro-instruction, are therefore, used for specifying additional modifiers (see Section 4.2.8 for description of opcode Extension One modifiers).

4.2.4 Load Immediate (LI) Format.

LI FORMAT



DATA Least Significant Eight Bits of Source Operand

D Destination Register

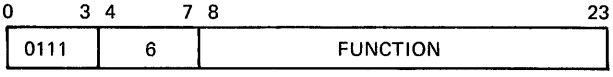
E1 Opcode Extension One Modifier

E2 Opcode Extension Two Modifiers

This format is used by the Load Immediate micro-instruction. The eight bit immediate field is expanded to a 20 bit value with a high order 12 bits forced to zero. This 20 bit value forms the source operand for the load operation. Bit 12 of the micro-instruction is not used by this format.

4.2.5 Command Format (C).

C FORMAT



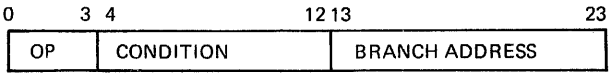
C Control Modifier

FUNCTION Function Field

The Command micro-instruction uses this format. A four bit field C specifies the control modifier, and a 16 bit function field specifies various functions to be performed by the Command micro-instruction.

4.2.6 Branch (BR) Format.

BR FORMAT



OP Operation Code

CONDITION Condition to be Tested

BRANCH ADDRESS Address of the next micro-instruction if branch is taken

This format is used by two Branch micro-instructions (Branch on True Bit 3 reset and Branch on False Bit 3 set).

4.2.7 Source and Destination Registers. There are several general purpose and special purpose micro-registers which can be used for sources and destinations. These sections describe addressable sources and destinations.

First Operand Source Registers

The sources that may be used as the first operand are shown in Table 1.

MR0, MR1, MR2, MR3, MR4, MR5, and MR6 are seven general purpose micro-registers. They can be used freely as 16 bit sources.

PSWH and PSWL are two 16 bit registers which are each used to hold a Program Status Word. These registers can be used as 16 bit sources.

Location Counter is a special micro-register which can be used as a 20 bit source. The most significant four bits of the Location Counter can be accessed separately by specifying LOCH as the first source. Specifying LOCH as the first source causes the most significant four bits of LOC to be gated on to the B Bus, Bits 12 through 15.

SRH and SRL are two special purpose shift registers which can be used as 16 bit sources.

Memory Address Register holds the address of the main memory location to be accessed. This register can be used as a 20 bit source register.

TABLE 1. ADDRESSABLE FIRST OPERAND SOURCES

RD BITS					SYMBOLIC REGISTER	MEANING
8	9	10	11	12		
0	0	0	0	0	MR0	Micro-register 0
0	0	0	0	1	MR1	Micro-register 1
0	0	0	1	0	MR2	Micro-register 2
0	0	0	1	1	MR3	Micro-register 3
0	0	1	0	0	MR4	Micro-register 4
0	0	1	0	1	MR5	Micro-register 5
0	0	1	1	0	MR6	Micro-register 6
0	0	1	1	1	PSWL	Program Status Word Low
0	1	0	0	1		
0	1	0	1	0		
0	1	0	1	1		
0	1	1	0	0	YSI	YS Field
0	1	1	0	1	YDI	YD Field
0	1	1	1	0	PSWH	Program Status Word High
0	1	1	1	1	OP	Operation Code
1	0	0	0	0	NULL	NULL Source
1	0	0	0	1	LOCH	Location Counter High
1	0	0	1	0	SRL	Shift Register Low
1	0	0	1	1	SRH	Shift Register High
1	0	1	0	0	LOC	Location Counter
1	0	1	0	1	IO	Input (D Bus to B Bus)
1	0	1	1	0	MDR	Memory Data Register
1	0	1	1	1	MAR	Memory Address Register
1	1	0	0	0	YDH	MS 16 bits of reg. specified by YD
1	1	0	0	1	YDL	LS 20 bits of reg. specified by YD
1	1	0	1	0	YDLP1	LS 20 bits of reg. specified by YD
1	1	0	1	1	YDLM1	LS 20 bits of reg. specified by YD
1	1	1	0	0	YSH	MS 16 bits of reg. specified by YS
1	1	1	0	1	YSL	LS 20 bits of reg. specified by YS
1	1	1	1	0	YSLX	LS bits of reg. specified by YS
1	1	1	1	1	YSHX	MS 16 bits of reg. specified by YS

Memory Data Register is available as a 20 bit first operand source. If MDR is specified as the source and the memory data is not available (after a memory read operation), execution of that micro-instruction is suspended until memory data is available.

The user's 16 general purpose registers do not have individual addresses. The General Registers are accessed indirectly through the YD or YS field. If YDL is specified as the first source, the least significant 20 bits of the General Register whose number is contained in the YD field (Bits 8:11 of the user instruction) are selected as the source value. Specifying YDLP1 (YDLM1) is the same as specifying YDL except that the YD field is incremented (or decremented) by one after the execution of the micro-instruction. If the current micro-instruction uses YDLP1 or YDLM1 as the source, the next **micro-instruction cannot specify YD for the source operand. Specifying YDH selects the most significant 16 bits of the General Register specified by the YD field (Model 7/32 only).**

YSL, when used as a source, selects the least significant 20 bits of the General Register specified by the YS field. If YSH is used as the source operand, the most significant **16 bits of the General Register are used as the source value (Model 7/32 only). Specifying YSLX or YSHX is the same as specifying YSL or YSH except that the source value is force to zero if the YS field (Bits 12 through 15 of the user instruction) is zero.**

If YDI or YSI is specified as the source, the YD field (Bits 8 through 11 of user instruction) or the YS field (Bits 12 through 15 of user instruction) is gated onto B Bus Bits 12 through 15.

When I/O appears as the source operand, an input operation is performed. Only in a Load micro-instruction can I/O be a source. The nature of the input request is specified by Bits 16 through 18 of the micro-instruction. When the device responds, the data is gated on to the B Bus. Completion of the micro-instruction is suspended until the device responds or a false sync **occurs (14 microsecond time set).**

When OP is specified as the first source operand, the operation code (Bits 0 through 7) of the user instruction is gated onto the **B Bus Bits 0 through 7.**

If the first operand is NULL all the bits on B and XB Busses are forced to zero.

NOTE

If the selected source has a 20 bit value the most significant four bits are gated onto the XB Bus and the least significant 16 bits are gated onto the B Bus. If the selected source has less than 20 bits, it is expanded to a 20 bit value by forcing high order bits to zero. The most significant four bits of this expanded value are gated to the XB Bus and the least significant 16 bits are gated to the B Bus.

Second Operand Sources

The sources that may be used as the second operand are shown in Table 2.

TABLE 2. ADDRESSABLE SECOND SOURCES

RD BITS			SYMBOLIC	MEANING
13	14	15		
0	0	0	NULL	NULL Value (ZERO)
0	0	1	ONE	Constant ONE
0	1	0	ARL	Arithmetic Register Low
0	1	1	TWO	Constant TWO
1	0	0	MDR	Memory Data Register
1	0	1	SIGN	SIGN of MDR
1	1	0	AR	Arithmetic Register
1	1	1	ARH	Arithmetic Register High

Specifying NULL for the second source forces the second operand value to zero.

If ONE or TWO is specified as the second operand, a constant of value 1 or 2 is used as the second operand value.

If ARL is specified as the second source, the least significant 16 bits of the 32 bit Arithmetic Register (ARH, ARL) are used as the second source operand. If ARH is specified as the second source, the most significant 16 bits of the 32 bit Arithmetic Register are used as the second operand value. If AR is specified as the second source, the least significant 20 bits of AR are used as the second operand.

Memory Data Register can be used as a 20 bit second operand. If the MDR has been used as the second source, and data is not available, execution of the micro-instruction is suspended until data is available. MDR should not be used as the second operand by a micro-instruction which specifies Memory Read operation or Instruction Read operation in the control field.

If SIGN is specified as the second operand and MDR Bit 4 (sign of 16 bit value) is zero, the second operand value is forced to zero. If SIGN is specified as the second source and MDR Bit 4 (Sign of 16 bit value) is 1, the second operand is forced to all 1s. SIGN should not be used as the second operand by a microinstruction which specifies memory read or instruction read.

Destination Registers

The registers which can be used for the destinations are shown in Table 3. Some of the destinations are 20 bit destinations and others are 16 bit destinations. If a 20 bit destination is specified, the most significant four bits of the destination register are loaded from the XS Bus and the least significant 16 bits of the destination register are loaded from the S Bus. If the specified register is 16 bits long, the destination register is loaded from the S Bus and the data on the XS Bus is ignored.

General purpose micro-registers MR0 through MR6 and shift registers SRH and SRL can be used as 16 bit destination registers in any arithmetic and logic micro-instruction.

TABLE 3. ADDRESSABLE DESTINATIONS

RD BITS					SYMBOLIC REGISTER	MEANING
19	20	21	22	23		
0	0	0	0	0	MR0	Micro-register 0
0	0	0	0	1	MR1	Micro-register 1
0	0	0	1	0	MR2	Micro-register 2
0	0	0	1	1	MR3	Micro-register 3
0	0	1	0	0	MR4	Micro-register 4
0	0	1	0	1	MR5	Micro-register 5
0	0	1	1	0	MR6	Micro-register 6
0	0	1	1	1	PSWL	Program Status Word Low
0	1	0	0	0	CTR	Counter
0	1	0	0	1	ARL	Arithmetic Register Low
0	1	0	1	0	ARH	Arithmetic Register High
0	1	0	1	1	AR	Arithmetic Register
0	1	1	0	0	YSI	YS Immediate
0	1	1	0	1		
0	1	1	1	0	PSWH	Program Status Word High
0	1	1	1	1	FLR	Flag Register
1	0	0	0	0	NULL	NULL Destination
1	0	0	0	1		
1	0	0	1	0	SRL	Shift Register Low
1	0	0	1	1	SRH	Shift Register High
1	0	1	0	0	LOC	Location Counter
1	0	1	0	1	IO	Output (S Bus to D Bus)
1	0	1	1	0	MDR	Memory Data Register
1	0	1	1	1	MAR	Memory Address Register
1	1	0	0	0	YDH	MS 16 bits of reg. specified by YD
1	1	0	0	1	YDL	LS 16 bits of reg. specified by YD
1	1	0	1	0	YDLP1	LS 16 bits of reg. specified by YD
1	1	0	1	1	YDLM1	LS 16 bits of reg. specified by YD
1	1	1	0	0	YSH	MS 16 bits of reg. specified by YS
1	1	1	0	1	YSL	LS 16 bits of reg. specified by YS
1	1	1	1	0		
1	1	1	1	1		

PSWH and PSWL are 16 bit destination registers. When PSWH or PSWL is specified as the destination, the least significant four bits from the S Bus are captured in the Flag Register (FLR). The least significant four bits of PSWL (Condition Code) can only be loaded from FLR by specifying the proper control modifier or by a command instruction.

The Location Counter is a 20 bit destination register. When LOC is loaded with a 20 bit value, MAR is loaded simultaneously with the same 20 bit value.

Memory Address Register (MAR) is a 20 bit destination register. If a micro-instruction specifies a main memory operation and uses MAR as the destination, the memory operation is started before changing the Memory Address Register.

Memory Data Register is available as a 16 bit destination (Bits 4:19 of MDR). If the MDR is specified as the destination, and memory is still busy (because of previous memory read or write operation), execution of that micro-instruction is suspended until memory is not busy.

General Registers can be specified as destination registers indirectly through YD and YS fields. Specifying YDH destination causes the most significant 16 bits of the General Register (specified by the YD field) to be loaded from the S Bus (Model 7/32 only). If YDL is specified as the destination the least significant 16 bits of the General Register (specified by YD field) are loaded from the S Bus. Specifying YDLP1 (or YDLM1) is the same as specifying YDL except that the YD field is incremented (or decremented) by one. If the current micro-instruction specifies YDLP1 or YDLM1 as the destination, the next micro-instruction cannot specify YD as the source operand.

If YSH (Model 7/32 only) or YSL is used for the destination, the most significant (or least significant) 16 bits of the General Register (specified by YS field) are loaded from the S Bus.

When the Counter is used as the destination register, the least significant five bits of the S Bus are loaded into the Counter Register.

If FLR is specified as the destination register, the least significant four bits of the S Bus are loaded into the Flag Register. Loading the Flag Register in an instruction that normally specifies flags causes ORing of the resulting C, V, and G flags and an ANDing of the L flag and the S Bus Data.

When YSI is the destination, the YS field of the Instruction Register is loaded from the least significant four bits of the S Bus.

ARL and ARH are two 16 bit destinations. AR is a 20 bit destination. When AR is specified as the destination, the least significant four bits of ARH are loaded from the XS Bus and ARL is loaded from the S Bus.

When I/O appears as the destination an I/O operation is performed. Only a Load or Load Immediate Micro-instruction can specify I/O as the destination. The nature of the output operation is encoded into the Operation Extension One Modifier Field.

4.2.8 Instruction Modifiers. Instruction modifiers can modify the function specified by the operation code. Modifiers are grouped into three categories as follows:

1. Control Modifiers (C)
2. Opcode Extension One Modifiers (E1)
3. Opcode Extension Two Modifiers (E2)

1. Control Modifiers (C)

These modifiers control the memory operations, DROM operations, and modification of the Condition Code. A micro-instruction can specify only one of these modifiers since they are mutually exclusive. The control modifier is specified by the C field of the micro-instruction. The following paragraphs of this section describe various control modifiers.

No Action

C Field	Symbolic
0000	

No control function is performed.

Memory Read and Increment LOC

C Field	Symbolic
0001	MRI

A memory read operation is started using the contents of MAR prior to execution of this micro-instruction as the memory address. The LOC and MAR are then incremented by two.

Vector through DROM1

C Field	Symbolic
0010	D1

The next sequential micro-instruction is executed and then control is transferred to the ROM address obtained by vectoring through DROM1 (using operation code of the user instruction as index). If the most significant bit of DROM1 entry is set, the corresponding **user instruction is a priviledged instruction. If PSW Bit 23 is set and the Most Significant Bit (MSB) of DROM1 entery is set, the least significant eight bits of the ROM Address Register are forced to hexadecimal value 'FF' (i.e., a branch is taken to the last word on the current ROM page).**

Vector through DROM2

C Field	Symbolic
0011	D2

The next sequential micro-instruction is executed and then control is transferred to the ROM address obtained by vectoring through DROM2. If the user instruction being emulated is RHR, RH, WHR, or WH and the I/O device is a byte oriented device, Bit 14 of the ROM Address Register is forced to 1 (the 11 bit address is contained in Bits 5 through 15 of ROM Address Register).

Instruction Read

C Field	Symbolic
0100	IR

Memory read operation for fetching the next user instruction is started. The following additional operations are performed.

- The next sequential micro-instruction is executed. The next micro-instruction should not be a Branch micro-instruction and it must not specify YSH, YSL, YSLX, YSHX, or YSI for source or destination. The control field of the next micro-instruction must specify no action.
- The data from main memory is loaded into the Instruction Register (IR) when data becomes available).
- The Flag Register is cleared after the completion of the next micro-instruction.
- LOC and MAR are incremented by two.
- If no interrupt is pending, control is transferred to START routine (at '001'). If any interrupt is pending or if control console is in the SINGLE mode, the control is transferred to HELP routine (at '045').
- XMDR is cleared.
- Abort flip-flop is reset.

Instruction Read and Jam Condition Code Half

C Field	Symbolic
0101	IRJH

The control function performed is same as IR except that the contents of the Flag Register is copied into the Condition Code after the next micro-instruction has been executed.

Instruction Read and Jam Condition Code

C Field	Symbolic
0111	IRJ

Same as IR except that the contents of the Flag Register is copied to the Condition Code after the next micro-instruction has been executed when in the extended mode (PSW Bit 11 reset). If the Processor is in the halfword mode (Bit 11 of PSW set) the Flag Register is copied to the Condition Code at the conclusion of the current micro-instruction.

Memory Read

C Field	Symbolic
1000	MR

Memory read operation is started before loading the destination register.

Memory Read and Increment MAR

C Field	Symbolic
1001	MR2

Memory read operation is started and then MAR is incremented by two.

Memory Read and Disable Memory Access Controller

C Field	Symbolic
10101	MRD

For any memory read or write operation the address contained in MAR is relocated by Memory Access Controller (MAC) if enabled by setting Bit 21 of PSW. If this option is specified, the address contained in MAR is not changed by the MAC and a memory read operation is initiated (Model 7/32 only).

Extended Read and Increment MAR by Two

C Field	Symbolic
1011	XR2

This control function is the same as MR2 except that the least significant four bits (Bits 16: 19) of the previous contents of MDR are copied to the most significant four bits (Bits 0:3) of the current MDR. For normal read operation, the most significant four bits of MDR are not **changed (Model 7/32 only).**

Memory Read with MAC Disabled, Increment MAR

C Field	Symbolic
1101	MRD2

Same as MR2 except that address contained in MAR is not changed by MAC. **(Model 7/32 only.)**

Memory Write

C Field	Symbolic
1110	MW

Memory write operation is started. If the current micro-instruction specifies a memory write operation and it uses MDR as the destination register, the new value of MDR is written into the addressed memory location.

Memory Write and Increment MAR by Two

C Field	Symbolic
1111	MW2

Memory write operation is started and then MAR is incremented by two.

Memory Write with MAC Disabled

C Field	Symbolic
1100	MWD

This control function is same as Memory Write (MW) except that memory address is not changed by MAC. (Model 7/32 only.)

NOTE

If a micro-instruction specifies a memory read or memory write operation, the address of the memory location to be accessed should be loaded into MAR prior to the execution of this micro-instruction. If MAR is specified as destination, the new value is loaded into MAR after starting the memory operation.

2. Opcode Extension One Modifiers (E1)

The E1 field is valid with the Load or the Load Immediate micro-instructions only. Four mutually exclusive modifiers are included in this class. These modifiers are described in the following paragraphs.

Shift Right

E1 Field	Symbolic
100	Shift Right

The source data (from the B Bus) is shifted by the B Bus shifter, right one place and copied into the specified destination register.

Shift Left

E1 Field	Symbolic
010	SL

The source data (from the B Bus) is shifted left one place and copied onto the specified destination register.

Cross Shift

E1 Field	Symbolic
110	CS

The source data (from the B Bus) is rotated eight bit positions and copied into the destination register. If MDR is the source or destination, the cross shift will occur only if MAR is even. If MDR is the destination and MAR is even, only the high byte of MDR (Bits 4 through 11) is loaded. If MAR is odd, only the low byte of MDR (Bits 12 through 19) is loaded. The bits not loaded remain unchanged. This special feature of MDR is used for emulating byte instructions.

Length

E1 Field	Symbolic
XX1	LEN

If the user instruction being emulated is of RX3 format (refer to Model 7/32 Reference Manual, Publication Number 29-399), the constant two is loaded into the destination register. If the user instruction is not of RX3 format zero value is loaded into the destination register. MDR must be used as the source operand when this modifier is specified. **This modifier is not for general use. The 7/32 micro-program uses this modifier to emulate some branch instructions.**

3. Opcode Extension Two Modifiers (E2)

The E2 field of a micro-instruction specifies this class of instruction modifiers.

If the source or destination is not I/O, the following three modifiers can be specified. These modifiers are not mutually exclusive, so two or more of them can be specified simulataneously.

Carry In

E2 Field	Symbolic
1XX	CI

If the micro-instruction using this modifier is a Load or Load Immediate micro-instruction, the state of the carry flag is shifted into the most significant (if shift right is specified) or the least significant bit (if shift left is specified) of the result.

If the micro-instruction using this modifier is an Add or Add Immediate instruction, the carry flag is added with the least significant bit of the sum.

If the micro-instruction specifying this modifier is a Subtract or Subtract Immediate Instruction, the carry flag represents the borrow situation from the least significant bit of the source data. This borrow participates in the subtraction operation.

If the micro-instruction specifying this modifier does not perform a load, add, or subtract operation, the modifier is ignored.

Carry Out

E2 Field	Symbolic
X1X	CO

If the micro-instruction specifying this modifier performs a load operation, the Carry flag stores the state of the bit shifted out (if shift operation is specified). If shift modifier is not specified by load micro-instruction, the Carry flag is reset.

If the micro-instruction specifying carry out performs an AND, OR, or Exclusive OR operation, the Carry flag is reset.

If the micro-instruction specifying carry out performs an AND, OR, or Exclusive OR operation, the carry flag is reset.

Flags

E2 Field	Symbolic
XX1	F

This modifier is used to enable modification of the flags (V, G, and L) if the micro-instruction specifying F modifier performs an add or subtract operation. G, L and V flags are adjusted to reflect the result of the operation. G and L flags are adjusted to reflect the algebraic value of the result. The V flag is adjusted to reflect the overflow condition.

If the micro-instruction specifying F modifier performs a load or logical operation, V flag is reset and G and L flags are adjusted to reflect the algebraic value of the result.

NOTE

The flags G, L and V reflect the result obtained from 16 bit ALU. The four bit extension to the ALU is used only for address arithmetic and it des not change any flags. The hardware provides a cumulative flage affect to facilitate multi-precision operations. Once the G or L flag becomes set, the G and L flags will never again be both zero unless the flag register is explicitly cleared.

Result	Flags Before Execution		Flags After Execution	
	G	L	G	L
ZERO	0	0	0	0
	0	1	1	0
	1	0	1	0
POSITIVE	0	0	1	0
	0	1	1	0
	1	0	1	0
NEGATIVE	0	0	0	1
	0	1	0	1
	1	0	0	1

If the source or destination of a Load micro-instruction is I/O the following seven modifiers can be specified by E2 Field.

Data Channel Acknowledge

E2 Field	Symbolic
000	DCAK

The source data is present on the data lines, and the address and data channel acknowledge control lines are active. The highest priority interrupting data channel device becomes the On-Line device.

Address

E2 Field	Symbolic
001	ADRS

The source data is present on the date lines, and the address control line is active. The device that detects its address becomes On-Line and responds with Sync.

Data Available

E2 Field	Symbolic
010	DA

The source data is present on the data lines, and the data available control line is active. The On-Line device accepts the data and responds with a Sync.

Output Command

E2 Field	Symbolic
011	OC

The source data is present on the data lines, and the output command control line is active.

Acknowledge Interrupt

E2 Field	Symbolic
100	ACK

The acknowledge interrupt control line is active. The highest priority interrupting device responds by placing its address on the data lines. The input data is copied to the destination register.

Data Request

E2 Field	Symbolic
101	DR

The data request control line is active. The On-Line device responds by placing data on the data lines. The input data is copied to the destination register.

Status Request

E2 Field	Symbolic
110	STAT

The status request control line is active. The On-Line device responds by placing status data on the data lines. The input data is copied to the destination register.

4.3 Micro-instructions

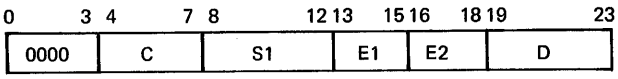
The 7/32 Processor executes 16 basic micro-instructions. This section describes these 16 micro-instructions. For each micro-instructions the assembler format is shown. The machine instruction format is diagrammed, a description of the instruction is provided, and allowed instruction modifiers are indicated. The execution time of a micro-instruction is given in terms of the number of machine cycles required to complete it. One machine cycle equals 250 nanoseconds.

Arithmetic and logic operations use 20 bit operands. If a specified operand is less than 20 bits long, it is expanded to a 20 bit value (with high order bits forced to zero) before the specified operation is performed. If the specified destination is a 20 bit register, a 20 bit result is loaded into the destination register. If the specified destination is a 16 bit register, the least significant 16 bits of the result are loaded into this register and most significant four bits of the result are ignored.

4.3.1 Load

L D, S1, OPTIONS

Timing: 1 Cycle (No I/O)
2 Cycles (Input I/O)
3 Cycles (Output I/O)



The contents of the register specified by the first source field (S1) are copied into the register specified by the destination field (D).

OPTIONS

If neither the source nor destination field specifies I/O, the following instruction modifiers can be used:

C Field	Any Control Modifier
E1 Field	SR, SL, CS, or LEN
E2 Field	One or more of the following:
	CI, CO, F

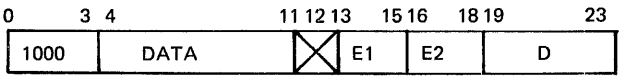
If the source or destination field specifies I/O, the following instruction modifiers are used:

C Field	Any Control Modifier
E1 Field	SR, SL, or CS
E2 Field	DCAK, ADRS, DA, OC, ACK, DR or STAT

4.3.2 Load Immediate

LI D, DATA, OPTIONS

Timing: 1 Cycle (No I/O)
3 Cycles (Output I/O)



The eight bits from the data field are copied into the least significant eight bits of the register specified by the D field. The most significant eight bits (12 bits for 20 bit destination) of the destination are forced to zero.

OPTIONS

If the destination field does not specify I/O, the following modifiers can be used:

E1 Field	SR, SL, or CS
E2 Field	One or more of the modifiers CI, CO, and F.

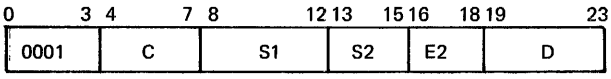
If the destination field specifies I/O, the following modifiers can be used:

E1 Field	SR, SL, or CS
E2 Field	DCAK, ADRS, DA, or OC

4.3.3 AND

N D,S1, S2, OPTIONS

Timing: 1 Cycle



The contents of the register specified by the first source field (S1) are logically 'ANDed' with the contents of the register specified by the second source field (S2). The logical product is loaded into the register specified by the destination field.

OPTIONS

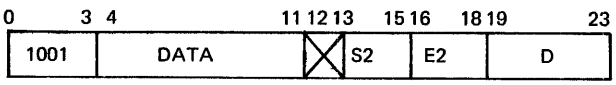
Any of the following instruction modifiers can be specified as options.

C Field	Any Control Modifier
E2 Field	One or more of CI, CO, and F

4.3.4 AND Immediate

NI D, DATA, S2, OPTIONS

Timing: 1 Cycle



The eight bit data field is expanded to a 20 bit value with high order 12 bits forced to zero. This 20 bit value is logically ANDed with the contents of the register specified by the second source field (S2). The logical product is loaded into the register specified by the destination field.

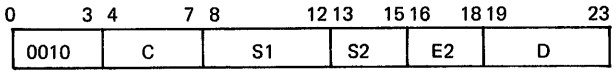
OPTIONS

E2 Field One or more of the following three modifiers:

CI, CO, and F

4.3.5 OR

O D, S1, S2, OPTIONS **Timing: 1 Cycle**



The first operand and second operand are logically added (or ORed). The logical sum replaces the contents of the specified destination register.

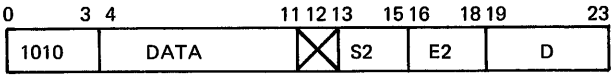
OPTIONS

C Field Any Control Modifier

E2 Field One or more of CI, CO, and F

4.3.6 OR Immediate

OI, D, DATA, S2, OPTIONS **Timing: 1 Cycle**



The eight data field is expanded to a 20 bit immediate value with high order 12 bits forced to zero. This immediate value is logically added (or ORed) to the second operand. The logical sum is loaded into the specified destination register.

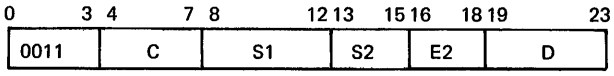
OPTIONS

E2 Field One or more of the following modifiers:

CI, CO, and F

4.3.7 Exclusive OR

X D, S1, S2, OPTIONS **Timing: 1 Cycle**



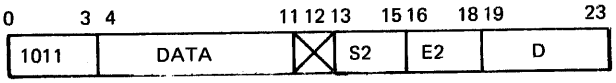
The first operand (specified by S1) is 'Exclusive ORed' with the second operand. The resulting logical difference is placed into the register specified by the destination field (D).

OPTIONS

C Field	Any Control Modifier
E2 Field	One or more of CI, CO, and F

4.3.8 Exclusive OR Immediate

XI D, DATA, S2, OPTIONS **Timing: 1 Cycle**



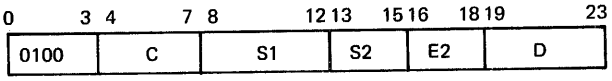
The eight bit data field is expanded to 20 bit immediate value with high order 12 bits forced to zero. This immediate value is 'Exclusive ORed' with the second operand. The resulting logical difference replaces the contents of the destination register.

OPTIONS

E2 Field	One or more of the following modifiers:
	CI, CO, and F

4.3.9 Add

A D, S1, S2, OPTIONS **Timing: 1 Cycle**



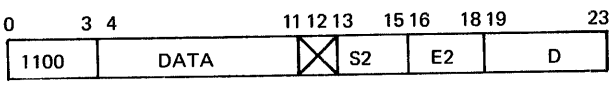
The second operand is algebraically added to the first operand. The sum replaces the contents of the destination register.

OPTIONS

C Field	Any Control Modifier
E2 Field	One or more the modifiers
	CI, CO, and F

4.3.10 Add Immediate

AT D, DATA, S2, OPTIONS **Timing: 1 Cycle**



The eight bit data field is expanded to 20 bit immediate value with high order bits forced to zero. This immediate value is added to the second operand. The sum replaces the contents of the destination register.

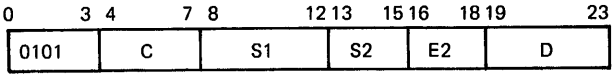
OPTIONS

E2 Field — One or more of the following modifiers can be specified as options:

CI, CO, F

4.3.11 Subtract

S D, S1, S2, OPTIONS — **Timing: 1 Cycle**



The second operand is algebraically subtracted from the first operand. The difference is loaded into the register specified by the destination field.

OPTIONS

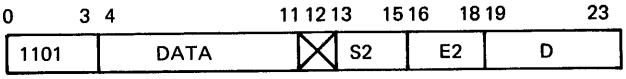
C Field — Any Control Modifier

E2 Field — One or more of the following modifiers:

CI, CO and F

4.3.12 Subtract Immediate

SI D, DATA, S2, OPTIONS — **Timing: 1 Cycle**



The eight bit data field is expanded to 20 bit immediate value with high order bits forced to zero. The second operand is subtracted from the expanded immediate operand, the difference is loaded into the register specified by the destination field.

OPTIONS

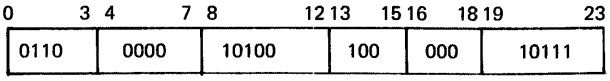
E2 Field — One or more of the modifiers

CI, CO and F

4.3.13 Calculate Address

CA MAR, LOC, MDR

3 Cycle (RX1 or RX2)
5 Cycles (RX3)



The Calculate Address instruction is a very powerful and special purpose micro-instruction. This micro-instruction has been specifically designed to improve the efficiency of the Instruction Fetch and operand Fetch processes of the RX format user instructions with the Model 7/32 emulation. The following conditions should be satisfied when the Calculate Address instruction is executed.

1. The first source and the second source registers must be LOC and MDR respectively. The **destination register must be MAR. The Control field (C) and Opcode Extension Two (E2) field must specify no action.**

2. Memory Data Register (MDR) should contain the second halfword of an RX instruction, or the memory read operation for fetching the second halfword should be in progress.

3. The Location Counter and the Memory Address Register must contain the current user instruction address plus four.

An RX instruction can specify any one of three RX formats (refer to Model 7/32 Reference Manual, Publication Number 29-399). The particular RX format is specified as follows:

Most Significant Two Bits of the Second Half Word of the User Instruction		RX Format
Bit 16	Bit 17	
0	0	RX1
1	X	RX2
0	1	RX3

The Calculate Address micro-instruction examines the most significant two bits of the second halfword of the user instruction (contained in Bits 4 and 5 of the MDR) and determines which one of the three RX formats has been used. If the user instruction is of the RX1 format, the least significant 14 bits of the second halfword (contained in Bits 6 through 19 of the MDR) are expanded to a 20 bit value by forcing the high order bits to zero. This 20 bit value is added to the least significant 20 bits of the index value specified by the YS field. The 20 bit sum is loaded into the MAR.

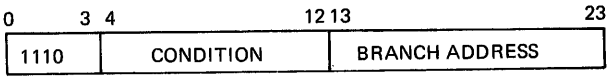
If the current user instruction is of the RX2 format the least significant 15 bits of the second halfword (contained in Bits 5 through 19 of the MDR) are expanded to a 20 bit value by propagating the most significant bit (Bits 5 of MDR) through the high order bits. This 20 bit value, 20 bit value contained in the Location Counter, and the least significant 20 bits of the index value are added to form the 20 bit effective address of the second operand. This 20 bit address is loaded into the MAR.

If the current user instruction is of the RX3 format, the **Calculate Address micro-instruction starts extended** memory read operation (XR2) to fetch the third halfword of the user instruction, and then increments LOC and MAR by two. While waiting for the memory data to become available, the Processor adds the least significant 20 bits of the first index value (contained in the General Register specified by Bits 12 through 15 of the user instruction) to the least significant 20 bits of the second index value (contained in the General Register specified by Bits 24 through 27) of the user instruction. The 20 bit sum forms the effective index value. When the memory data becomes available, **the MDR Bits 0 through 19 contain the 20 bit address** specified by Bits 28:47 of the user instruction. The Processor adds this 20 bit address to the 20 bit effective index value. The resulting 20 bit effective address is loaded in to the Memory Address Register.

4.3.14 Branch On True

BT CONDITION, ADDRESS

1 Cycle (No Branch)
2 Cycles (Branch)



The Branch on True micro-instruction results in a transfer to the Branch Address if any of the specified conditions is true. If none of the specified conditions is true, the next sequential micro-instruction is executed. The branch conditions to be tested are specified by Bits 4 through 12 of the micro-instruction. See Table 4 for the branch conditions.

If counter = 0 (CNTR) is specified as the branch condition, the transfer takes place only if the counter Register contains a zero value. The Counter is decremented by one after testing the condition.

TABLE 4. BRANCH CONDITIONS

CONDITION FIELD									SYMBOLIC CONDITION	MEANING
4	5	6	7	8	9	10	11	12		
0	0	1							C	Carry Flag Set
0	0		1						V	Over Flow Flag Set
0	0			1					G	Greater Than Flag Set
0	0				1				L	Less Than Flag Set
0	0					1			MSK1	Mask (AND Operation Between YD Field and Condition Code Produce Non-Zero Result)
0	0						1		WAIT	Processor is in Wait State (PSW Bit 16=1)
0	0							1		
0	1	1							ATNX	I/O Attention and PSW Bit 17=1 and No Higher Priority Interrupt is Present
0	1		1						HW	Processor in Halfword Mode (PSW Bit 11=1)
0	1			1					QUE	PSW Bit 22 is Set
0	1				1				RR	Current User Instruction is an RR or SF Format Instruction
0	1					1			ATN	I/O Attention and PSW Bit 17 Set
0	1						1		MAC	MAC Interrupt and PSW Bit 21 Set
0	1							1		
1	0	1							SNGL	Console Single Mode
1	0		1						CATN	Console Attention
1	0			1					DC	Data Channel Requests
1	0				1				DRD	Data Channel Read
1	0					1			MALF	Machine Malfunction
1	0						1		PPF	Primary Power Fail
1	0							1		
1	1	1							HW10	Halfword I/O Line is Active
1	1		1						NNORM	Not Normalized (SRH Bits 8 through 11 are 0)
1	1			1					CNTR	Counter Contains a Zero Value
1	1				1				ARST	Auto Restart Present
1	1	0	0	0	0	1	0	0	UT	Utility Flip-Flop Set
1	1						1		SHORT	User Instruction is of RX1 or RX2 Format
1	1							1		

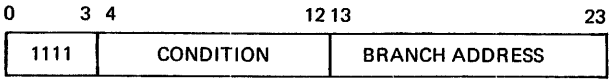
4.3.15 Branch on False

BF CONDITION, ADDRESS

Timing: 1 Cycle (No Branch)
2 Cycles (Branch)

or

B ADDRESS (condition field is forced to zero by the assembler)



The Branch on False micro-instruction results in a transfer to the Branch Address if none of the specified conditions is true. If any of the specified conditions is true, the next sequential micro-instruction is executed. The branch conditions to be tested are specified by Bits 4 through 12 of the micro-instruction. See Table 4 for the branch conditions.

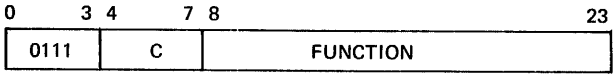
If Counter = 0 (CNTR) is specified as the branch condition, the transfer takes place only if the Counter Register does not contain a zero value. The counter is decremented by one after testing the branch condition.

4.3.16 Command

C FUNCTION + CONTROL OPTIONS

Timing: 1 Cycle (except MPY, DIV, RPT, and UNMPY)
10 Cycles (MPY or UNMPY)
16 Cycles (DIV)

16 Cycles (DIV)



The Command micro-instruction instructs the Processor to perform the functions specified by the function field (Bits 8 through 23 of the micro-instruction). The following paragraphs describe the various command functions (see Table 5).

Multiply (MPY)

The Processor multiplies the 16 bit multiplicand in MDR (Bits 4 through 19) by the 16 bit multiplier in SRL. The 32 bit signed product is obtained in SRH and SRL. The following set up conditions must exist prior to the execution of the command micro-instruction.

1. MDR contains the multiplicand in two's complement form.
2. ARL contains twice the multiplicand in two's complement form.
3. SRL contains the multiplier in two's complement form.
4. Carry flag contains the most significant bit of the multiplicand.

TABLE 5. FUNCTIONS (FOR COMMAND MICRO-INSTRUCTIONS)

FUNCTION FIELD																SYMBOLIC FUNCTION	MEANING
8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23		
1	0				1											MPY	Multiply
1	1				1											UMPY	Unsigned Multiply
		1				1										DIV	Divide
			1	0	0											SL1	Shift (SRH,SRL) Left One Place
			1	0	1											SL2	Shift (SHR,SRL) Left Two Places
			0	1	0											SR1	Shift (SRH,SRL) Right One Place
			0	1	1											SR2	Shift (SRH,SRL) Right Two Places
						1										CI	Carry In
									1							CO	Carry Out
								0									
								0		1	0	0				SUT	Set Utility Flip-Flop
								0		0	1	0				CUT	Clear Utility Flip-Flop
								0		1	1	0				TUT	Toggle Utility Flip-Flop
								0				1				RPT	Repeat
								0					1			SWA	Set Wait Indicator
								0					0			CWA	Clear Wait Indicator
								0						1		POW	Power Down
								0							1	ALRM	Load Alarm Flags
								1									
								1		1						TABT	Toggle Abort Flip-Flop
								1			1					PRIV	Privileged Memory Operation
								1				1				JH	Copy FLR to Condition Code
								1					1				
								1						1		TS	Test and Set
								1							1	CYD	Clear YD Field

The Processor used a fast multiplication algorithm to multiply two 16 bit numbers. The algorithm used by the Processor speeds up the multiplication by pairing the multiplier bits and inspecting one pair at a time. The multiplication process is completed in 10 machine cycles and the signed product in two's complement form is obtained in SRH and SRL.

Unsigned Multiply (UMPY)

This command performs the same operation as MPY except that it multiplies two unsigned 16 bit numbers, and the product obtained in SRH and SRL is an unsigned 32 bit number.

Divide (DIV)

The Processor divides the 32 bit dividend in SRH and SRL by the 16 bit divisor in ARL. To achieve this, the following setup condtions must exist.

1. The Counter contains 16
2. SRH, SRL contain a positive dividend that is less than 65,536 times the divisor magnitude.
3. The ARL contains the divisor in two's complement negative form.
4. Carry flag is reset.

Shift Left One Place (SL1)

The 32 bit shift register (SRH, SRL) is shifted left one place. Carry in and carry out can be specified.

Shift Left Two Places (SL2)

The 32 bit shift register (SRH, SRL) is shifted left two places. The carry in or carry out can be specified, but the carry produced at the end of the first shift is ignored.

Shift Right One Place (SR1)

The 32 bit shift register (SRH, SRL) is shifted right one place. Carry in and carry out can be specified.

Shift Right Two Places (SR2)

The 32 bit shift register (SRH, SRL) is shifted right two places. Carry in and carry out can be specified but the carry produced at the end of the first shift is ignored.

Carry In (CI)

Carry flag is shifted into the most significant bit of the SRH (if shift right) or the least significant bit of the SRL (if shift left).

Carry Out (CO)

The bit shifted out is saved in the Carry flag.

Set Utility Flip-Flop (SUT)

Utility flip-flop is set.

Clear Utility Flip-Flop (CUT)

Utility flip-flop is reset.

Toggle Utility Flip-Flop (TUT)

Utility flip-flop state is complemented

Repeat (RPT)

If the counter is not zero, the next sequential micro-instruction is repeated the number of times specified by the Counter Register. If the Counter is non-zero, reasonable micro-instructions that do not result in a branch can be executed.

Set Wait Alarm (SWA)

The Wait indicator is set

Clear Wait Alarm (CWA)

The Wait indicator is reset

Power Down (POW)

The system is initialized

Copy Alarm Bits (ALRM)

The bits set in the Alarm Register are 'ORed' with the Flag Register the next time PSWL is loaded.

Toggle Abort Flip-Flop

The ABORT flip-flop is complemented (the micro-program can be interrupted by the I/O interrupts if the ABORT flip-flop is set. This flip-flop is reset when Instruction Read is specified)

Privileged Memory Operation (PRIV)

If this function is specified, the address contained in MAR is not relocated by MAC, and all memory protection is disabled for this micro-instruction.

Test and Set (TS)

This command function is used to achieve synchronization in multi-processor systems. When this function is received in conjunction with a memory read operation, the most significant bit of the addressed word in the common memory is set during the write portion of the same memory cycle. (Model 7/32 only.)

Clear YD Field (CYD)

YD field of the Instruction Register is forced to zero.

4.4 Micro-program

The INTERDATA Model 7/32 is a 32 bit computer. The user instructions on the Model 7/32 are interpretively executed (emulated) by the micro-program. The micro-program is executed by the micro-processor which has 16 bit wide data paths. Since the host (micro-processor) and target (7/32 user machine) machine data widths do not match, the micro-program creates a "virtual match" through the proper manipulation of the facilities available on the micro-processor. The micro-program uses multiple precision micro-programming techniques to perform 32 bit operation with 16 bit data paths. The micro-program for the Model 7/32 also emulates the user instruction of the 16 bit processors. In the 16 bit emulation mode (called Halfword Mode, refer to 7/32 Reference Manual, Publication Number 29-399) most of the arithmetic and logical operations performed are 32 bit operations. Since only the least significant 16 bits of a General Register are meaningful in the Halfword Mode, the compatible 16 bit results are obtained without using separate execution routines for the Halfword mode instructions.

The listing of the Model 7/32 Firmware or micro-program is well documented and self-explanatory for many of the less involved user instructions requiring short execution routines. Section 4.4.1 explains the basic emulation process and the emulation of the simple arithmetic and logic user instruction. The subsequent sections describe the Firmware implementation of some of the more involved user instructions. Console support, interrupt support, and the Auto Driver Channel Firmware routines have also been explained. Most of this description is also applicable to the 7/16 HSALU micro- program.

4.4.1 Emulation Process. At the highest level, the process of emulation can be divided into three major tasks as depicted in Figure 2A: User Instruction **Fetch, Operand Fetch, and Execution.** In Figure 2B, the major tasks are broken into typical subtasks. This general functional diagram applies to the emulation of both the 7/32 and 7/16 HSALU user machines.

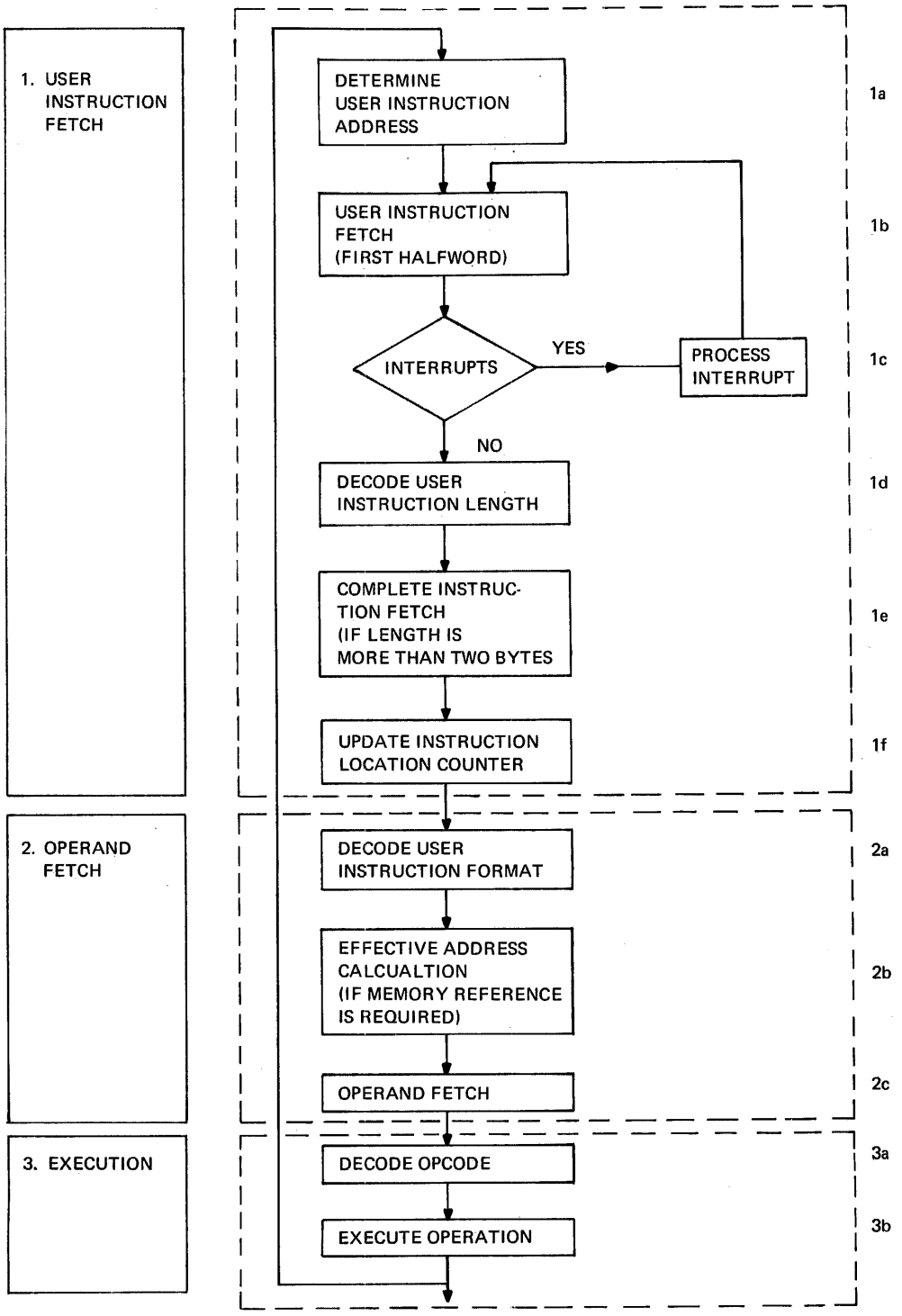


FIGURE 2A: HIGH LEVEL EMULATION TASKS

FIGURE 2B: EMULATION SUBTASKS

Subtasks of the emulation process can be further divided into multiple subtasks. For example, subtask 3B, Execute Operation, consists of many alternate processes, essentially one per operation code. Each of these processes may be represented as one or more subtasks, some of which may be similar to other subtasks in the emulation process.

1. User Instruction Fetch

User instruction fetch begins when a micro-instruction specifying instruction read (in Control Modifier field) is executed. The instruction fetch for the next user instruction is started by the next to last micro-instruction of the execution micro-routine. The execution micro-routine for the current user instruction copies the updated Location Counter (LOC) into the Memory Address Register (MAR) before the execution of the next to the last micro-instruction. The next to the last micro-instruction of this micro-routine then specifies IR, IRJ or IRJH (see Section 4.2.8 for the description of these control modifiers) in its Control Modifier field. The hardware executes the last micro-instruction of the micro-routing and then sets the ROM Address Register to '001' (corresponds to label START on the micro-program listing). If any interrupts are pending, the hardware sets the ROM Address Register to '045' (corresponds to label HELP) instead of '1'. Before transferring control to the micro-instruction at START the hardware increments the Location Counter by two, clears the Flag Register (FLR), resets the Abort flip-flop, and copies the first halfword of the user instruction from the MDR into the Instruction Register (IR)(OP, YD, and YS).

The following two micro-instructions at 'START' load the contents of the General Register specified by the YS field into ARH and ARL (only ARL is loaded for the 7/16 HSALU).

```
START    L     ARH, YSH, D1     (ARH, ARL) = (R2) VECTOR THROUGH
         L     ARL, YSL, MRI    DROM1; START MEMORY READ AND INCREMENT
  *                             LOC AND MAR BY TWO
```

The second micro-instruction uses the MRI Control Modifier to fetch the second halfword of the user instruction (see Section 4.2.8 for the description of MRI Control Modifier). If the user instruction is of the SF or RR format, (refer to the 7/32 Reference Manual, Publication Number 29-399 for the description of the user instruction formats) format, this control operation is not performed by the processor.

The instruction decoding can be easily performed using a table look-up on eight bits of the instruction (operation-code), which establishes not only the operation to be performed but also the format and the method of operand fetch. Since most instructions require an operand, a table can be devised to direct the reading of the operand and the subsequent operation to be performed. DROM1 and DROM2 have been designed to implement this look-up procedure. Since any DROM operation is specified one micro-instruction in advance, the table look-up on the op-code is achieved without any time penalty. The micro-instruction at 'START' specifies vectoring through DROM1. The Processor, therefore, interrogates DROM1 after the execution of the micro-instruction at START + 1. If the current user instruction is a fixed point RR or an SF instruction, the ROM address supplied by DROM1 is the execution routine address. If the current instruction is an RI1, RI2, RX1, RX2, RX3 or a Floating Point instruction, the DROM1 address is the Operand Fetch routine address.

2. Operand Fetch

Most user instructions require two operands to perform the specified function. The first operand is contained in a General Register and the second operand is contained either in a General Register or in a memory location. Normally, the execution micro-routine for any arithmetic or logical operation assumes that the second operand is contained in the Arithmetic Register (ARH, ARL). If the current user instruction is of the RR format, two micro-

instructions at 'START', load the Arithmetic Register (ARH, ARL) with the second operand value. If the current user instruction is not an RR format instruction, the appropriate Operand Fetch routine is entered by vectoring through DROM1. The following paragraphs describe the various Operand Fetch routines used by the 7/32 micro-program.

RI1 Operand Fetch

In an RI1 format instruction, the second operand is obtained by adding the contents of the index register specified by the X2 field to the value contained in the I2 field (refer to the 7/32 Reference Manual, Publication Number 29-399). Before adding the immediate value to the contents of the index register, the 16 bit immediate value is expended to a 32 bit fullword quantity by propagating the most significant bit through the high order bits. The following two micro-instructions at label RI1 perform this operand fetch operation.

```
RI1     A   ARL, YSLX, MDR, CO + D2     (ARH, ARL) = EXTENDED IMMEDIATE
*       A   ARH, YSHX, SIGN, CI         FIELD + INDEX VALUE; VECTOR THROUGH
                                        DROM 2
```

As explained previously, obtaining the second operand value requires a 32 bit addition. Since the Arithmetic Logic Unit (ALU) cannot perform a 32 bit operation, this addition is accomplished in two steps. The first micro-instruction adds the 16 bit immediate value (now contained in the MDR as a result of the MRI Control operation) to the least significant 16 bits of the 32 bit index value. The arithmetic carry produced is saved in the Carry Flag of the Flag Register. The first micro-instruction also specifies D2 control action (vector through DROM2) so that control is transferred to the appropriate execution routine after the fetch operation is complete. The second micro-instruction adds the most significant 16 bits of the index value. This second micro-instruction also specifies Carry In (CI) in its Opcode Extension Two Modifier (E2) field. Specifying Carry In (CI) instructs the Processor to add the Carry flag of the Flag Register with the least signficant bit of the sum. Thus, a 32 bit sum is obtained in the Arithmetic Register (ARH, ARL).

RI2 Operand Fetch

If the current user instruction is of the RI2 format, the second operand is obtained by adding the contents of the index register specified by X2 to the 32 bit immediate value contained in the I2 field. The following three micro-instructions load ARH and ARL with the second operand value.

```
RI2     L   ARH, MDR, MRI               (ARH) = MS 16 BITS OF IMM. VALUE;
*                                       START THIRD HALFWORD FETCH;
*                                       INCREMENT LOC AND MAR BY TWO
        A   ARL, YSLX, MDR, CO + D2     (ARH, ARL) = IMM VALUE + INDEX VALUE;
        A   ARH, YSHX, ARH, CI          VECTOR THROUGH DROM2
```

The first micro-instruction saves the most significant 16 bits of the immediate value (contained in MDR) in the Arithmetic Register High (ARH), starts the memory read operation for fetching the least significant 16 bits of the immediate value contained in the third halfword of the user instruction, and increments the contents of the Location Counter (LOC) and the Memory Address Register (MAR) by two. The second micro-instruction adds the least significant 16 bits of the immediate value (now contained in MDR) to the least significant 16 bits of the index value and saves the arithmetic carry in the carry Flag. The sum is stored in the Arithmetic Register Low (ARL). The second micro-instruction also specifies D2 in its Control field, so that control is transferred to the appropriate Execution routine

after the completion of the third micro-instruction. The third micro-instruction adds the most significant 16 bits of the immediate value (contained in ARH), the most significant 16 bits of the index value, and the carry from the previous addition. The sum is stored in the Arithmetic Register High (ARH).

RX Halfword Operand Fetch

If the current user instruction is an RX instruction and it requires only a halfword from memory (for the second operand), this routine is entered after vectoring through DROM1. The user instruction specifies the memory address for the second operand. The halfword located at the specified memory address is fetched and expanded to a 32 bit value by propagating the most significant bit through the high order bits. This fetch routine consists of the following four micro-instructions.

```
RXH     CA  MAR,LOC,MDR        CALCULATE EFFECTIVE ADDRESS OF THE
 *                              SECOND OPERAND AND PLACE IT INTO MAR
 *      L   MAR,LOC,MR         START MEMORY READ AND THEN COPY CON-
 *                              TENTS OF LOC INTO MAR
         L   ARL,MDR,D2         (ARH,ARL)== 32 BIT SECOND
         A   ARH,NULL,SIGN      OPERAND VALUE;VECTOR THROUGH DROM2
```

The first micro-instruction is a very powerful micro-instruction. This micro-instruction examines the most significant two bits of the second halfword of the user instruction, and determines which one of the three RX formats has been specified. The effective address of the second operand is then loaded into MAR as follows:

RX1 FORMAT: (MAR)←INDEX VALUE (BITS 12:31) + 14 BIT ABSOLUTE VALUE IN MDR (BITS 6:19)

RX2 FORMAT: (MAR)←INDEX VALUE (BITS 12:31) + CONTENTS OF LOCATION COUNTER + 15 BIT TWO'S COMPLEMENT NUMBER IN MDR (BITS 5:19)

RX3 FORMAT: (MAR)←FIRST INDEX VALUE (BITS 12:31) + SECOND INDEX VALUE (BITS 12:31) + 20 BIT ADDRESS SPECIFIED BY USER INSTRUCTION (BITS 28:47)

If the user instruction specifies RX3 format, the Calculate Address (CA) micro-instruction also fetches the third halfword, and increments MAR and LOC by two.

The second micro-instruction starts a memory read operation for fetching the halfword located at the effective address, and then copies the contents of the updated Location Counter (LOC) into the Memory Address Register (MAR). The third micro-instruction loads the ARL with the contents of the halfword located at the specified address. This micro-instruction also specifies a D2 control operation so that the appropriate execution routine is entered after the completion of the fourth micro-instruction. The fourth micro-instruction loads the ARH with the most significant 16 bits of the expanded second operand value.

RX Fullword Operand Fetch

If the current user instruction is an RX instruction and it requires a fullword from memory, this routine is entered after vectoring through DROM1. This routine consists of the following five micro-instructions.

```
RXF     CA  MAR, LOC, MDR
*                           (MAR) = EFFECTIVE ADDRESS OF SECOND
                            OPERAND
        L   NULL, NULL, MR2
                            START MEMORY READ AND INCR MAR BY TWO
*       L   ARH, MDR, MR
                            (ARH) = MS 16 BITS OF SECOND OPERAND;
                            START MEMORY READ OPERATION
        L   MAR, LOC, D2
                            (MAR) = (LOC); VECTOR THROUGH DROM2
        L   ARL, MDR
                            (ARL) = LS 16 BITS OF SECOND OPERAND
```

The first micro-instruction loads the effective second operand address into MAR. The second micro-instruction starts the memory read operation (for fetching the most significant 16 bits of the second operand) and increments MAR by two. The third micro-instruction loads the most significant 16 bits of the second operand into the ARH and starts the memory read operation for fetching the least significant 16 bits. The fourth micro-instruction copies the updated Location Counter (LOC) into the Memory Address Register and specifies a D2 control operation. The fifth micro-instruction loads the ARL with the least significant 16 bits of the second operand. After the completion of the last micro-instruction control is transferred to the appropriate Execution routine.

3. Execution

Most arithmetic and logic user instructions require two micro-instruction execution routines. The execution routine for a normal arithmetic or logic user instruction assumes that the first operand is contained in a General Register specified by the YD field of the Instruction Register (IR). It also assumes that the second operand value has been loaded into the

Arithmetic Register (ARH, ARL), and the updated Location Counter (LOC) value has been copied into MAR. If the current user instruction is an RR or SF instruction, the Execution routine is entered from DROM1. If the current user instruction is an RI1, RI2, or RX instruction, the Execution routine is normally entered from DROM2. The following two micro-instructions form the Execution routine for the Fixed Point Add Operation. This routine is used by AR, A, AH, AI, and AHI user instructions. (This routine is also used by Halfword mode AHR, AH, and AHI instructions.)

```
A       A   YDL, YDL, ARL, CO+F+IRJ  (R1) = (R1) + (ARH, ARL);
        A   YDH, YDH, ARH, CO+CI+F   FETCH NEXT INSTR. AND SET CONDITION
*                                    CODE
```

The first micro-instruction adds the least significant 16 bits of the second operand to the least significant 16 bits of the first operand, and deposits the 16 bit sum into the least significant 16 bits of the General Register specified by the YD field. This micro-instruction also saves the arithmetic Carry In (CI) the Carry flag, modifies the G, L and V flags to reflect the result of the addition and initiates the instruction fetch for the next user instruction. The modified flag register is copied to the Condition Code.

The second micro-instruction adds the most significant 16 bits of the second operand to the most significant 16 bits of the first operand. The Carry flag is added with the least significant bit of the sum and the result is loaded into the upper half of the General Register specified

by the YD field. The Arithmetic carry produced by the add operation is saved in the Carry flag, the G and L are adjusted to reflect the algebraic value of the result, and the V flag is adjusted to reflect the overflow condition. Since the hardware provides a cumulative flag affect to facilitate multiprecision operations (refer to Section 4.3.3), the Flag Register, at completion of the second micro-instruction, reflects the result of the 32 bit add operation. The contents of the Flag Register, at the conclusion of the second micro-instruction is copied into the user Condition Code (if the Processor is in Halfword mode, the second micro-instruction does not modify the Condition Code).

After the execution of the second micro-instruction LOC and MAR are incremented by two, Flag Register is cleared and control is transferred to 'START' (at address X'001'). Now the same emulation process is repeated for the next user instruction.

The execution routines for other Fixed Point Arithmetic and Logical Operations are similar and they can be understood from the micro-program listing (refer to 7/32 micro-program listing 05-054A13 and 7/16 with high speed ALU micro-program listing 05-051A13).

4.4.2 System Initialization. On power up, or following initialize, when the System Clear signal (SCLR) goes high, the Processor starts executing micro-instructions. The system clear signal forces the ROM Address Register to X'100' (see Figure 3 for flow chart of the System Initialization Routine).

The micro-program addresses the Loader Storage Unit (LSU) (Device Number 5). If the false sync does not occur (LSU is present), the micro-program branches to routine "LSU' otherwise normal power up operation is performed by the micro-program.

The PSW Save Pointer is fetched from location X'84' and saved in MR0. The Register Save Pointer is fetched from location '86', and both the register sets (Set 0 and Set 'F') are loaded from the memory area (of 128 bytes) pointed to by this pointer.

The Program Status and the Location Counter are restored from the memory area (of eight bytes) pointed to by the PSW Save Pointer. The micro-program examines the old console status (saved in location) X'28' by the CONSER routine before powerfail or initialize). If the Console is not in the Run mode or the Processor is not equipped with Auto-Restart, a branch is taken to routine LOCDIS, (shown in Figure 5). If the Console is in the Run mode and the Processor is equipped with Auto-Restart, the Machine Malfunction Enable bit (PSW Bit 18) is tested. If this bit is set, a branch is taken to MMFINT routine, otherwise the Wait bit (PSW Bit 16) is tested. If the Wait bit is set, routine WAIT is entered. If the Wait bit is reset, the Wait indicator is cleared and the user instruction pointed to by LOC is executed.

Routine LSU is entered from the System Initialize routine (PWRUP) if the false sync does not occur when the Loader Storage Unit (LSU) (Device Number 5) is addressed. The micro-program clears the most significant 16 bits of the Program Status, reads in two bytes from the LSU and loads them into the least significant 16 bits of the Program Status. The LSU routine reads in a 16 bit new LOC value, a 16 bit starting memory address, and a 16 bit ending memory address. The difference between the ending address and the starting address is formed in MR1. If the starting address is greater that the ending address, routine IDLE is entered. If the starting address is less than or equal to the ending address, the data bytes are read from the LSU and stored in the consecutive byte locations in the main memory. When a data byte has been loaded into the memory location corresponding to the ending address, PSW Bit 16 is examined. If PSW Bit 16 is set, the interruptable Wait loop is entered, otherwise the user instruction addressed by LOC is executed.

4.4.3 Interrupt System. During user instruction fetch, the hardware tests for interrupts. If any of the tested interrupts (PPF, MALF, MAC, ATN, or CATN) are active routine HELP is entered. Routine HELP determines the nature of the interrupt, and branches to the appropriate interrupt processing routine. See Figure 4 for the interrupt support routines.

4.4.3.1 Primary Power Fail. Routine PWRDWN is entered if the Primary Power Fail signal is active. Current Program Status and Location Counter are saved in the PSW save area (pointed to by the PSW Save Pointer) and the General Registers of both register sets (Set 0 and Set 'F') are saved in the Register Save area (pointed to by the Register Save Pointer). The Command Power Down micro-instruction (C POW) is performed which stops the Processor and closes the Initialize relay.

4.4.3.2 Machine Malfunction Interrupt. Machine Malfunction (MALF) can be caused by the Memory Parity Error or Early Power Fail if PSW Bit 18 is set, or by Primary Power Fail (PPF). If MALF is active and PPF is not active, routine MMFINT is entered. This interrupt routine saves the Current Program Status and Location Counter in memory locations '20' to '27'. The new Program Status and Location Counter values are loaded from the locations '38' to '3F'. This Routine copies the contents of the Alarm Register (ALRM) to the Flag Register (FLR). If the Machine Malfunction occurred during the Auto-Driver Channel operation, the C flag is set by the micro-program. Routine TWAIT is then entered. If PSW Bit 16 is reset, the next user instruction is executed. If PSW Bit 16 is set, the interruptable Wait loop is entered.

4.4.3.3 Memory Access Controller Interrupt (MACINT). This interrupt routine is entered from HELP routine, if the Memory Access Controller interrupt is active and PSW Bit 21 is set. The Memory Address Register (MAR) is loaded with the constant '90', the Utility flip-flop is cleared, and the Common Interrupt routine (COMINT) is entered.

4.4.3.4 Arithmetic Fault Routine. This routine is entered when a fixed point or floating point fault occurs. If this routine is entered as a result of the floating point fault, the register MR0 contains eight, otherwise MR0 contains zero. This routine examines PSW Bit 19. If Bit 19 is set, the MAR is loaded with address '48', the Utility flip-flop is set, and the Common Interrupt routine (COMINT) is entered. If Bit 19 of the PSW is reset, the user instruction addressed by the Location Counter is executed.

4.4.3.5 Illegal Instruction Interrupt (ILLEG). If the current user instruction specifies an illegal operation code or if a privileged user instruction is executed in the protect mode (PSW Bit 23 set), this routine is entered. This routine decrements the LOC value by two if the instruction format is RR or or four if the instruction format is not RR or SF, loads the MAR with the address '30', clears the Utility flip-flop and transfers control to the Common Interrupt routine (COMINT).

4.4.3.6 Queue Interrupt (QUEINT). This routine is entered after the execution of a LPSW, LPSWR or EPSR user instruction if PSW Bit 22 is set. The routine examines the system queue. If the system queue is empty, routine TWAIT is entered. If the system queue contains any entries, the queue interrupt is taken. The Memory Address Register (MAR) is loaded with '88'; the queue address is saved in Register 13 (of Set 0), the Utility flip-flop is cleared, and a branch is taken to the Common Interrupt routine (COMINT).

4.4.3.7 Common Interrupt Routine (COMINT). The Program Status and the Location Counter values are saved in Registers 14 and 15 of Set 0. The new values of the Program Status and Location Counter are loaded from the eight byte memory area pointed to by the contents of the MAR. If the Utility flip-flop is set, the least significant four bits of MR0 are copied into the Condition Code, and routine TWAIT is entered.

4.4.3.8 Routine TWAIT. Most interrupt routines enter TWAIT after performing the PSW swap. If PSW Bit 16 is reset, the next user instruction is fetched. If PSW Bit 16 is set, the WAIT indicator is set and CATN signal is examined. If the CATN signal is active, the Console Service routine (CONSER) is entered otherwise an interruptable Wait loop (WAIT) is executed. If ATN, PPF, MALF, or CATN becomes active, the Wait loop is exited. If SNGL is active, one user instruction is executed otherwise routine HELP is entered to determine the exact nature of the interrupt.

4.4.3.9 I/O Interrupt. If none of the PPF, MALF, or MAC interrupt signals are active, the microprogram tests for I/O Attention (ATN). If ATN is active and PSW Bit 17 is set, the following I/O service is performed.

If the Processor is in Halfword mode (PSW Bit 11 is set) and the Protect mode bit (PSW Bit 23) is reset, routine HWINT is entered. Routine HWINT performs a PSW swap with location '0040'.

If the Processor is not in Halfword mode or the Protect mode bit (PSW Bit 23) is set, routine AUTI0 is entered for performing Automatic I/O. The detailed flow chart of the Automatic I/O routine (AUTI0) is shown in Figure 6.

4.4.4 Console Support. The Hexadecimal Display Panel is serviced by routine CONSER. Routine CONSER is entered from HELP routine if CATN or SNGL is active and none of the higher priority interrupts (PPF, MALF, MAC or I/O interrupts) are pending. The detailed flow chart of the Console Support routine (CONSER) is shown in Figure 5.

4.4.5 Floating Point Instructions. This section describes the micro-program implementation of the Floating Point instructions. The 'load' routine normalizes an unnormalized number but all other micro-routines (except 'float') assume normalized floating point operands and produce normalized results (except FIX). The R1 (and R2 for an RR Floating Point instruction) field of the user instruction must specify an even floating point register. If an odd floating point register is specified, the result obtained is incorrect. All floating point registers require two halfwords in reserve memory area. The addresses of the first and second halfwords of the n^{th} floating point register are calculated as follows:

Address of first halfword = 2*n

Address of second halfword = 2*n+2

In floating point arithmetic micro-routines, the sign and magnitude processing is used, where the signs are stripped off and processed separately. The basic arithmetic processes positive operands and produces positive results (the fraction part of a floating point number is always in the true form).

If underflow occurs, a zero result (all bits zero) is generated; if overflow occurs the largest possible magnitude (all bits one) is generated.

The detailed flow charts of LME, STME, STE, LE, LER, AE, AER, SE, and SER instructions are shown in Figure 7.

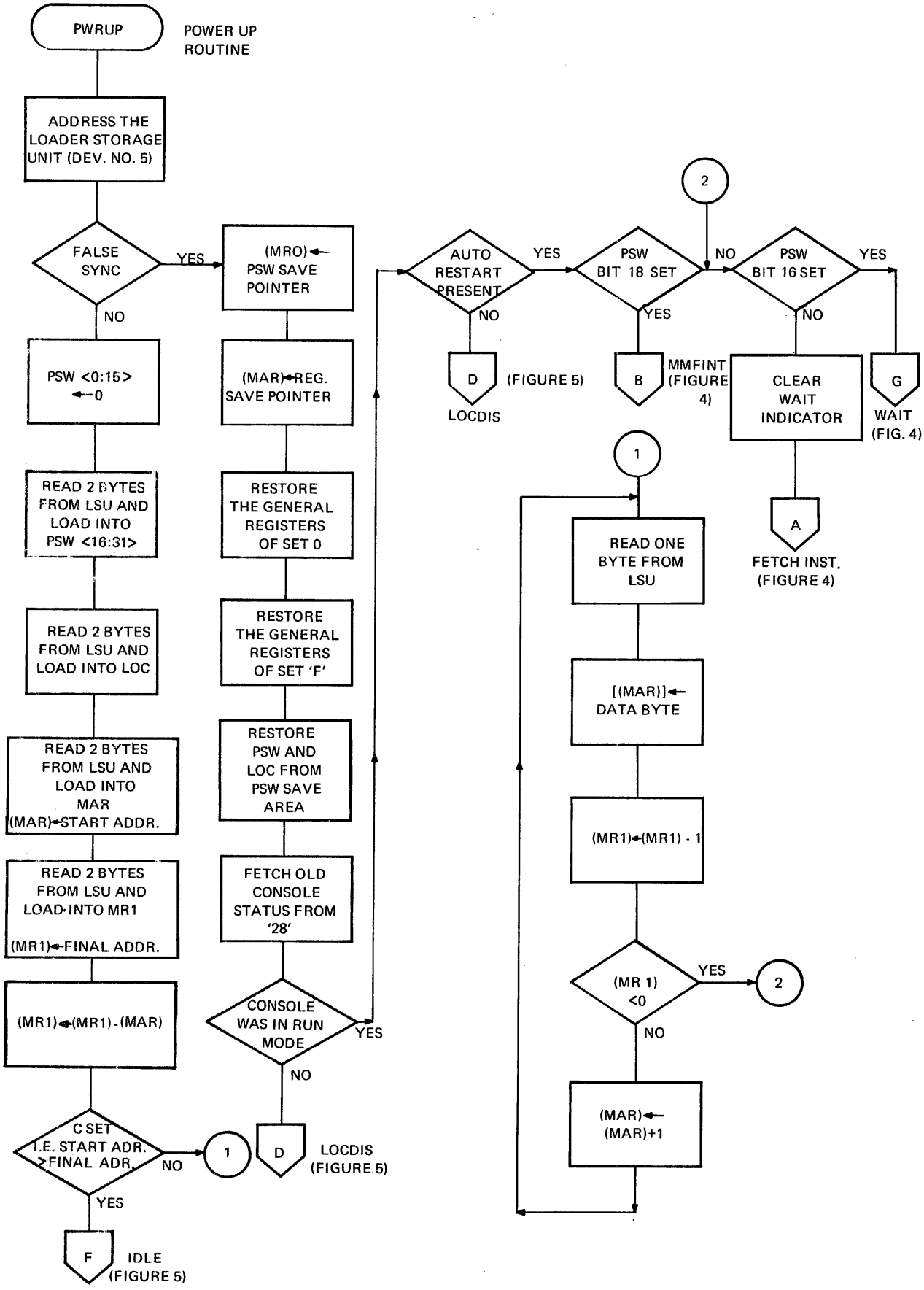


Figure 3. System Initialization

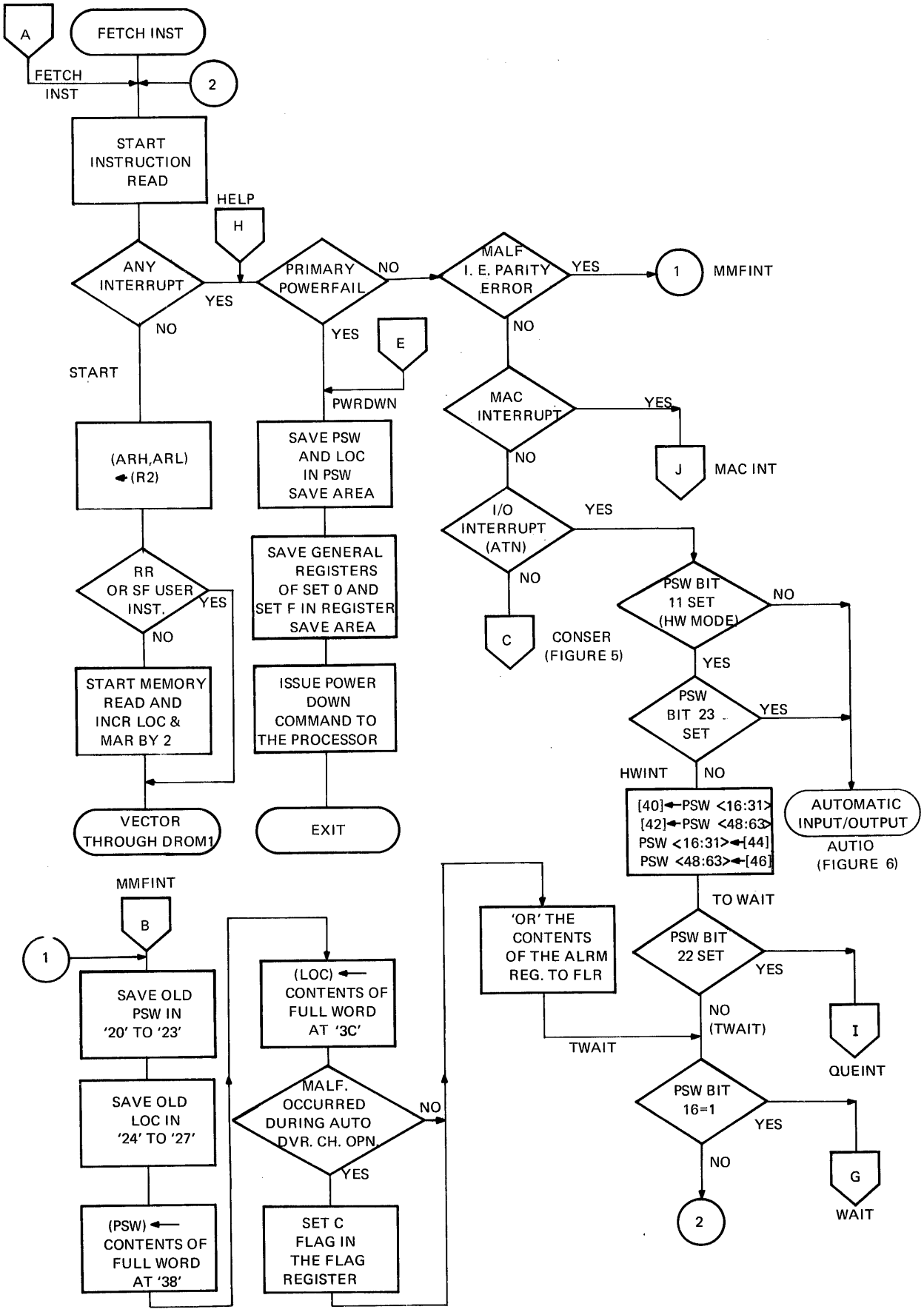


Figure 4. Interrupt Support

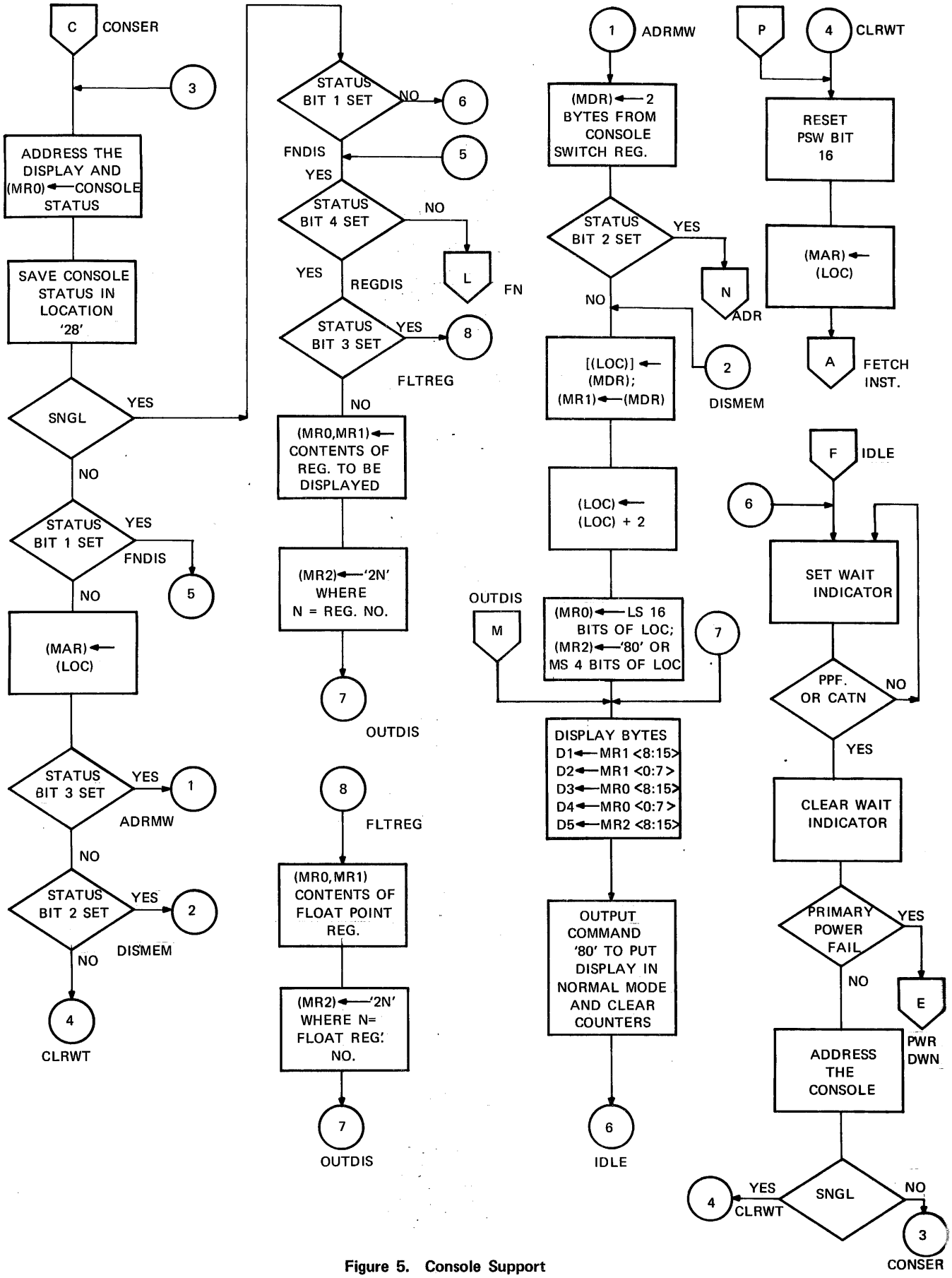


Figure 5. Console Support

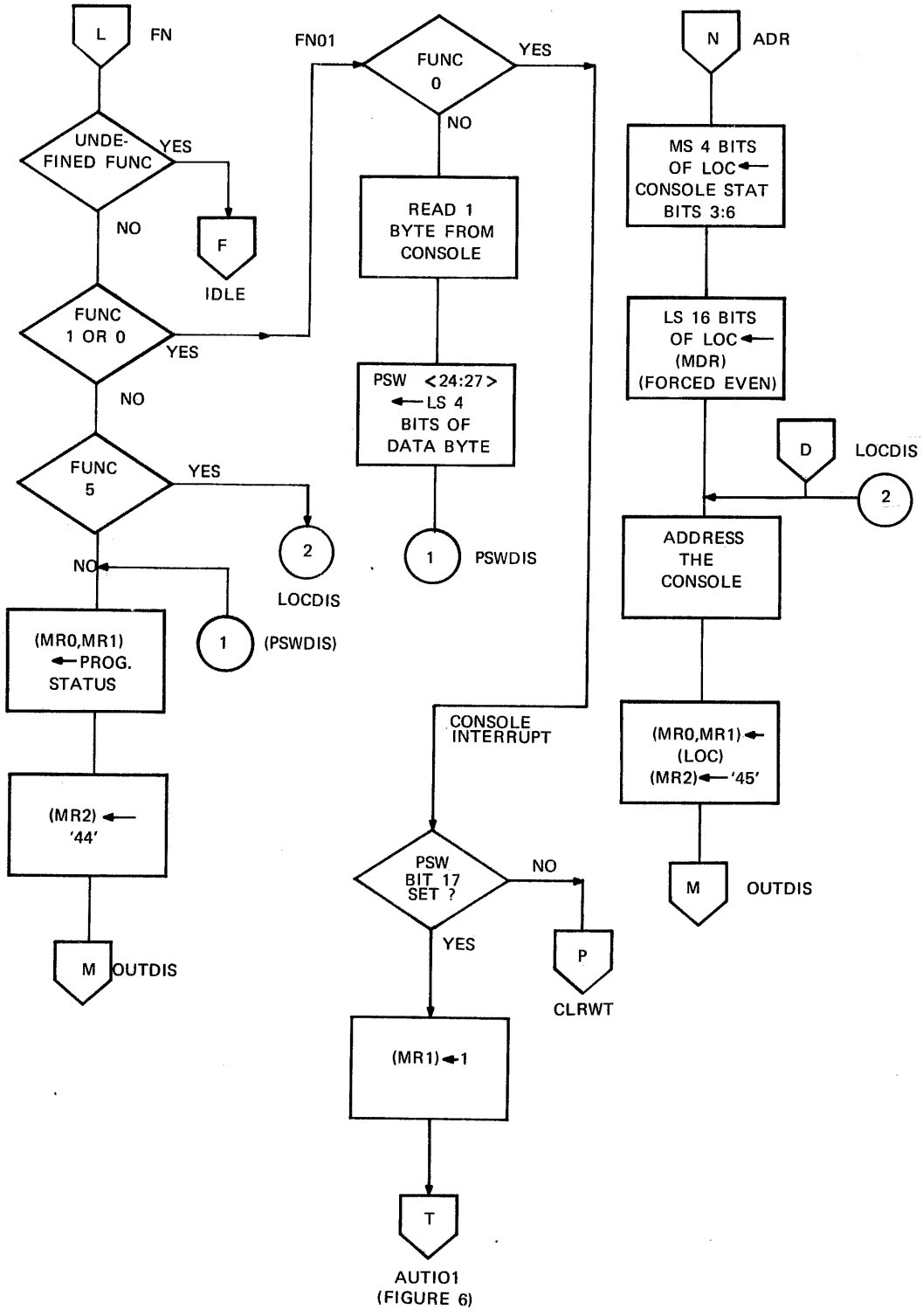


Figure 5. Console Support (Continued)

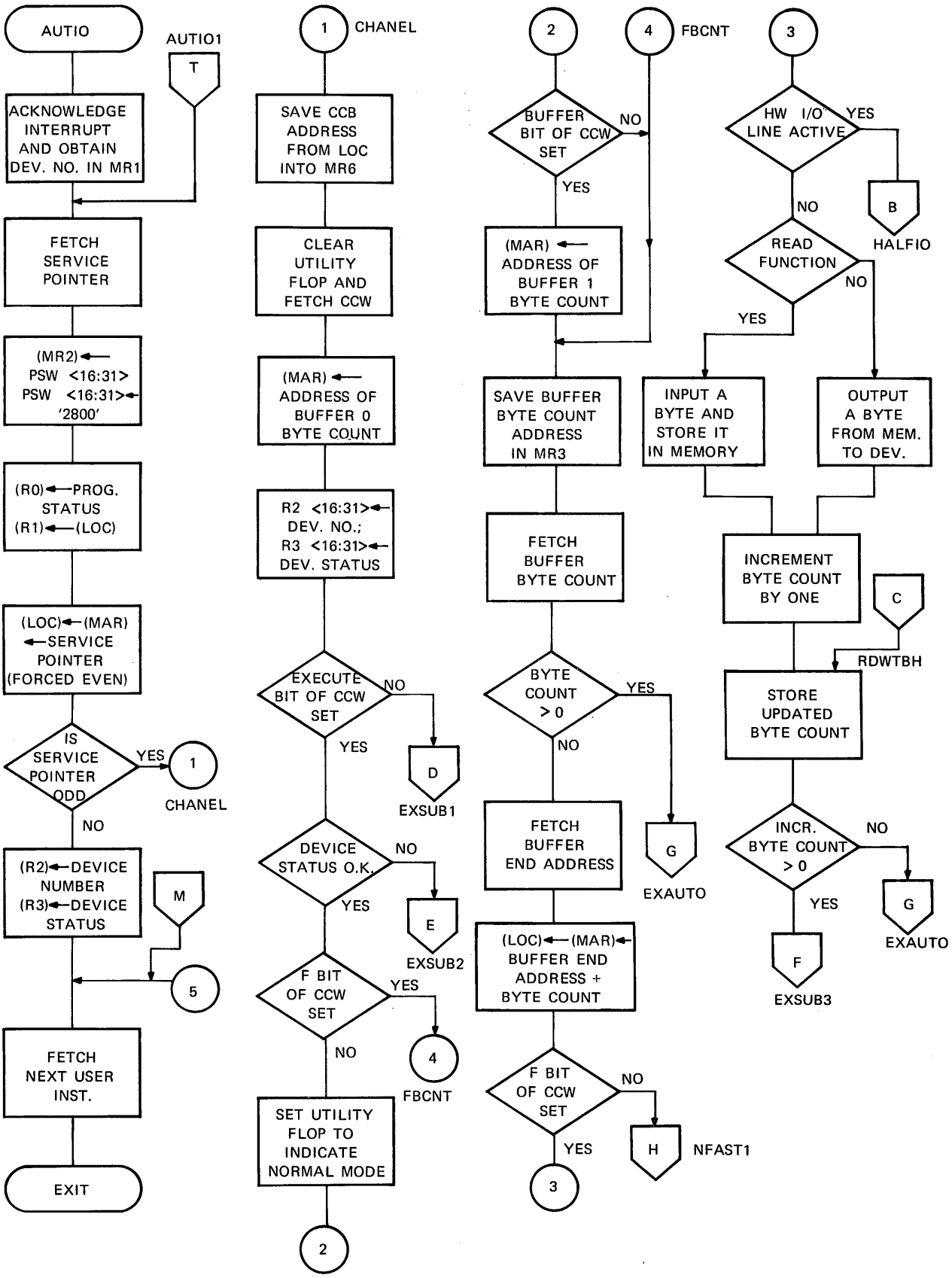


Figure 6. Automatic I/O

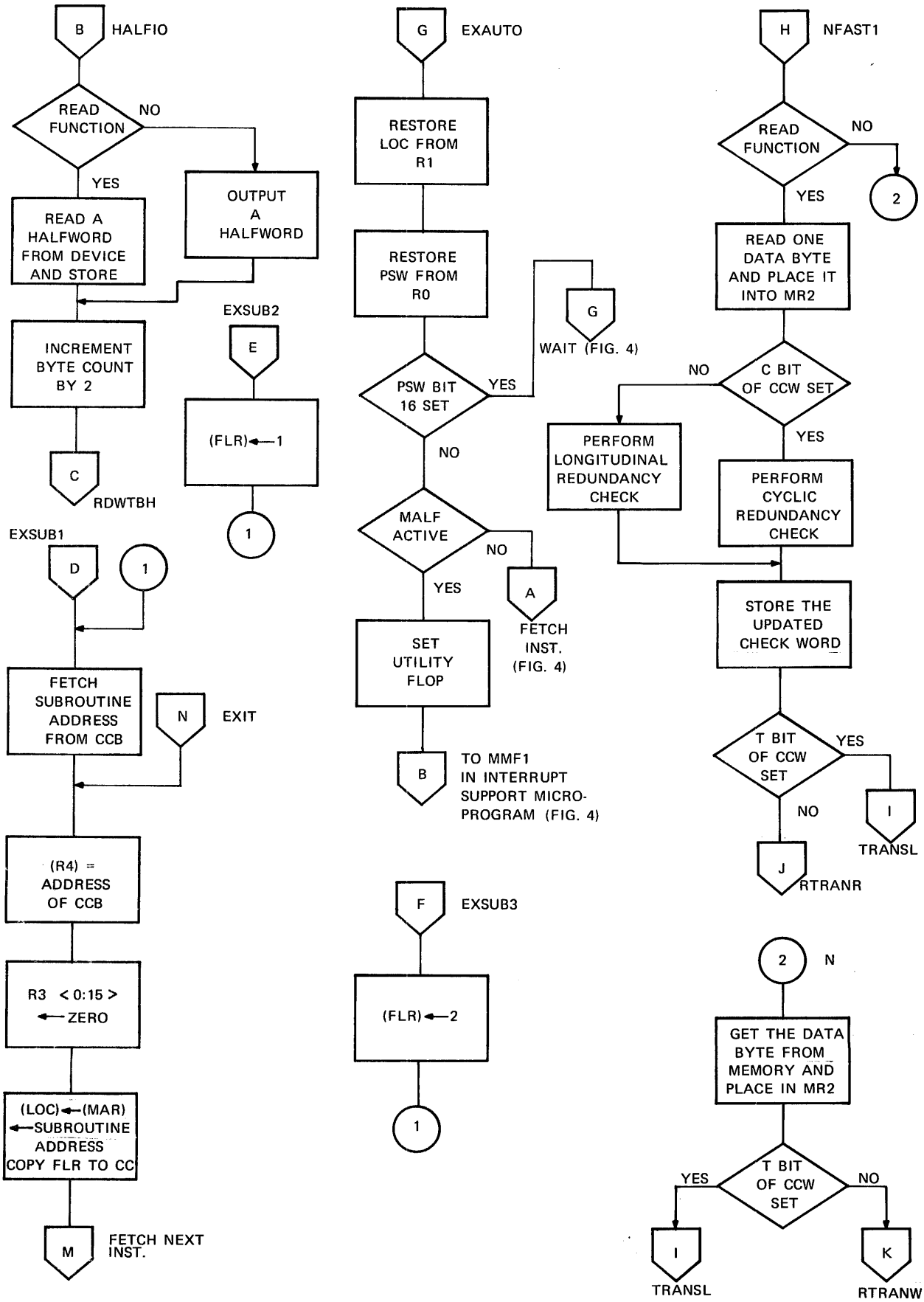


Figure 6. Automatic I/O (Continued)

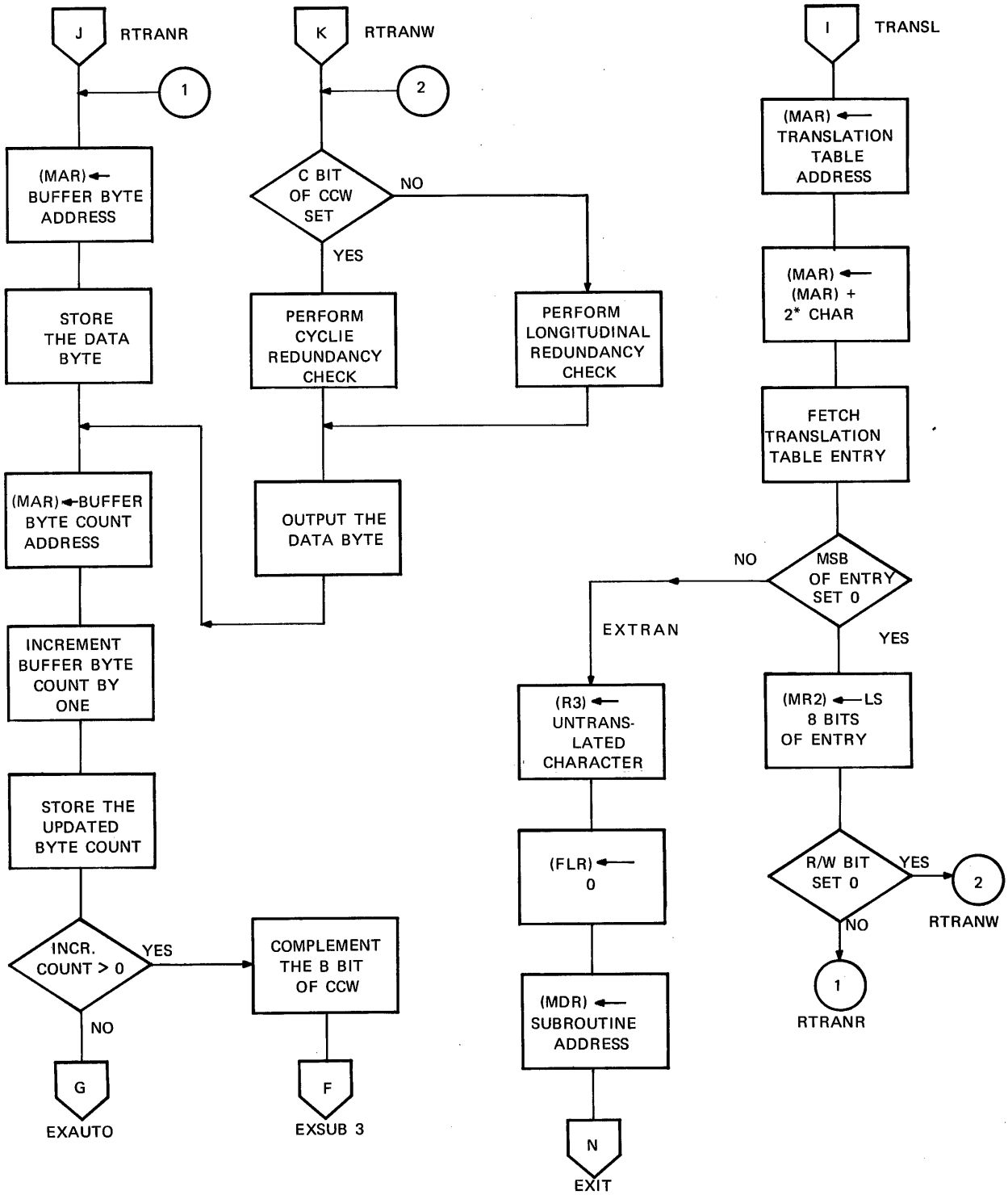


Figure 6. Automatic I/O (Continued)

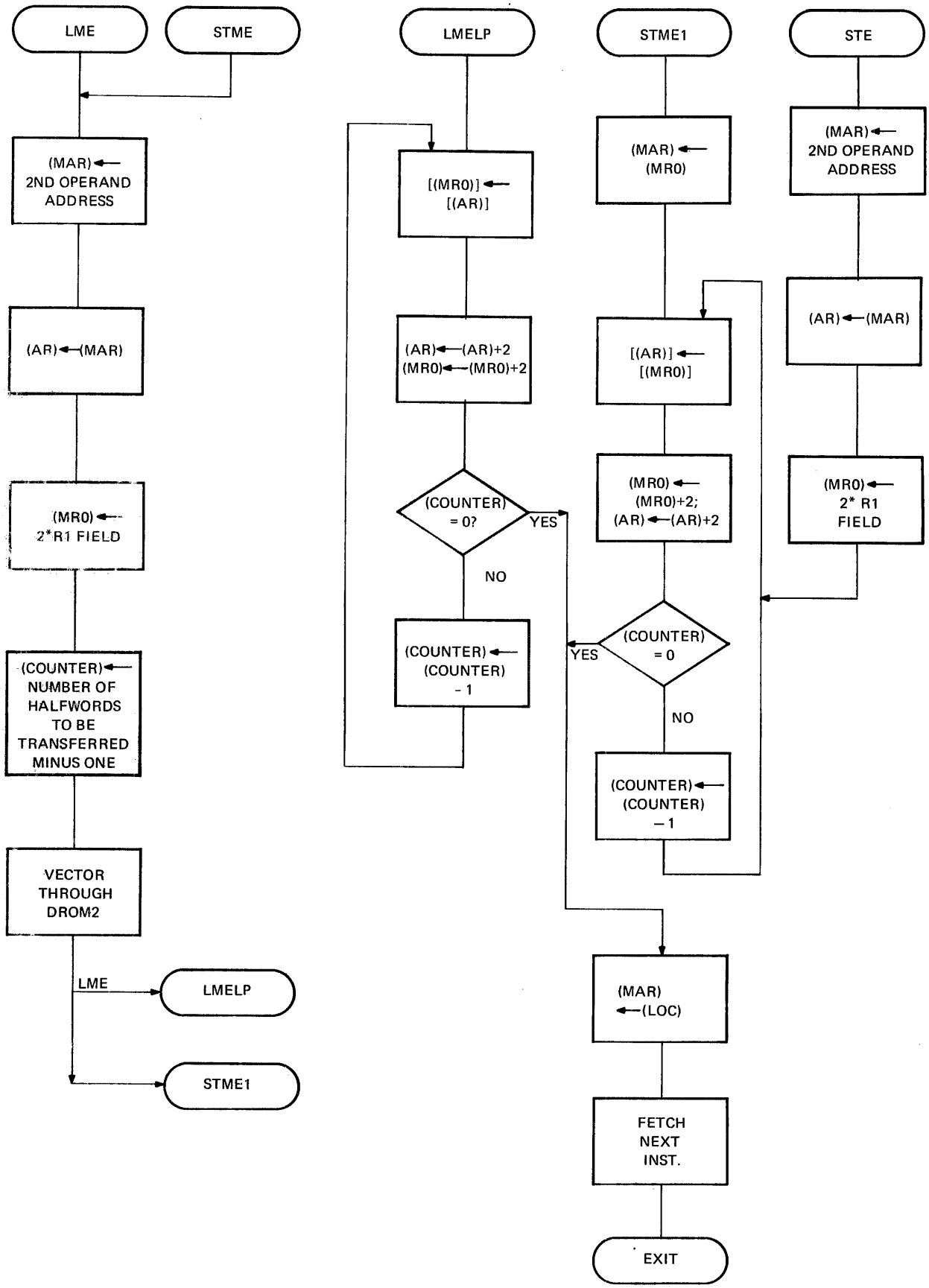


Figure 7. Floating Point Instructions

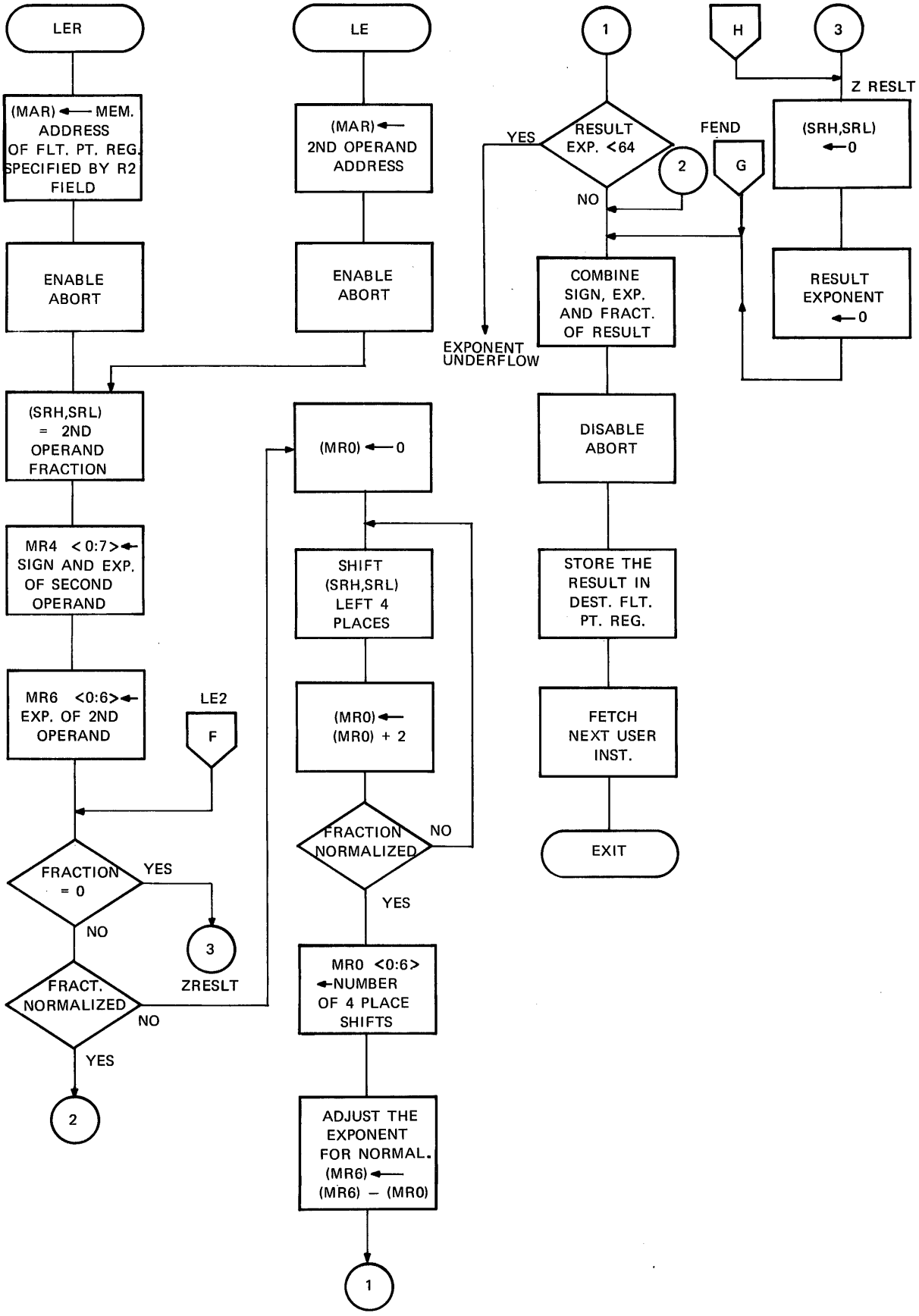


Figure 7. Floating Point Instructions (Continued)

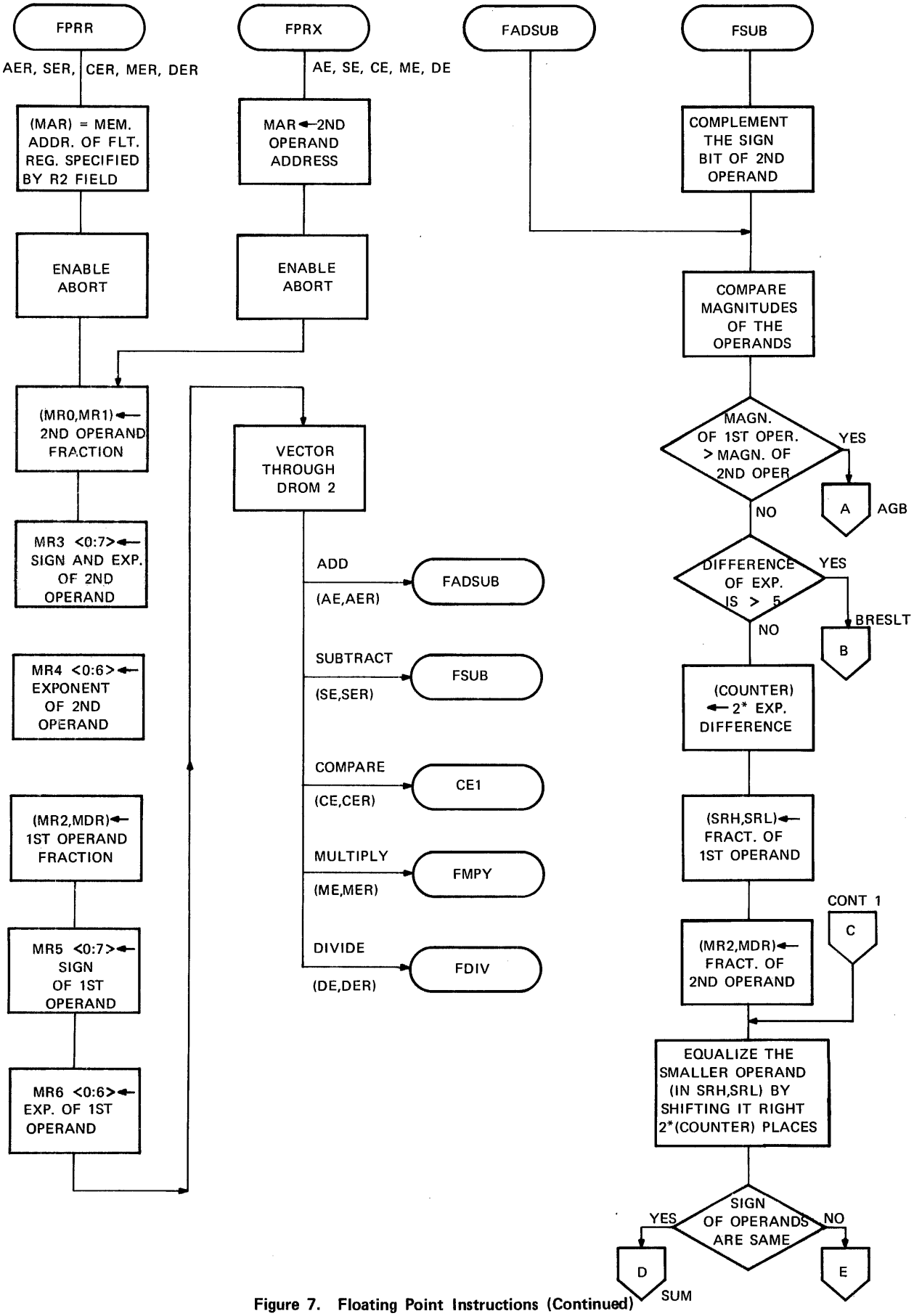


Figure 7. Floating Point Instructions (Continued)

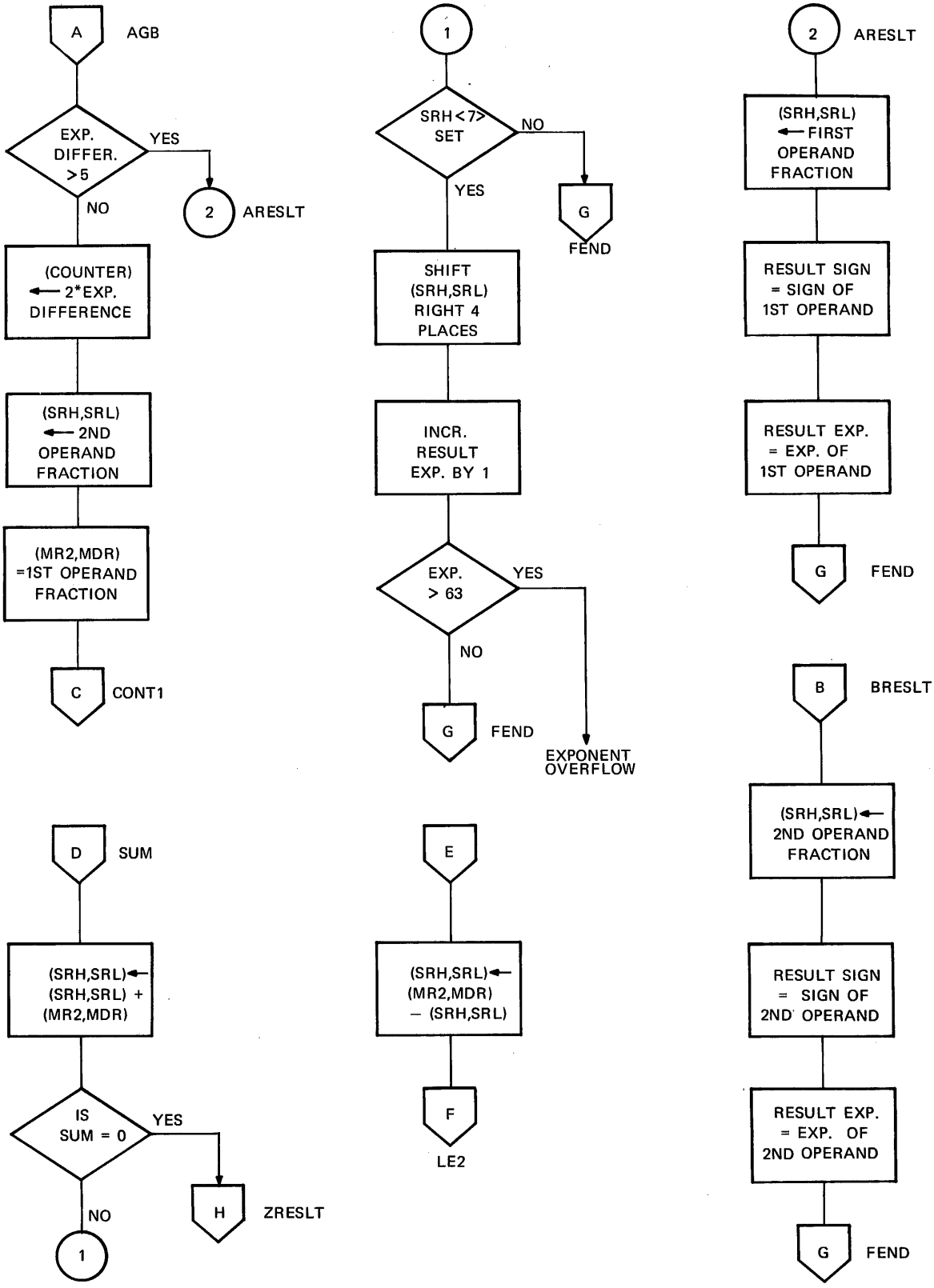


Figure 7. Floating Point Instructions (Continued)

NOTE

Two versions of the CPU-A and CPU-C boards have been manufactured. While functionally similar, the boards contain slight differences which affect the schematics. The two types of boards are differentiated by a "manufacturing" variation number. The boards are covered schematically as shown:

NAME	NUMBER	SCHEMATIC
CPU-A	35-522	01-079D08
CPU-A	35-522M01	01-079M01D08
CPU-C	35-524	01-079D08
CPU-C	35-524M01	01-079M01D08

5.1 Introduction

This section relates to Functional Schematic 01-079D08, Sheets 4 through 36. Note that in INTERDATA functional schematics, the last character in the mnemonic symbol designates the logic level when the signal is active. For example; D050 is Data Line Number 5 (D05). The last character (0) indicates that when D050 is active, the line is at a logical zero level. Refer to the General Description section of this manual for further information concerning the INTERDATA documentation system.

5.2 Clock Control

The Clock Generator is shown on Sheet 32. The clock system employs a free running 16 MHz oscillator. The oscillator output is inverted to generate OSC0. The oscillator is adjustable (via the variable Capacitor C1), over the range of 55 to 120 nanoseconds.

OSC0 is used as the clock inputs to a pair of flip-flops arranged as a two bit counter. The outputs from this counter are ANDed to form the **Clock signal** (CLK1, CLK1A, CLK1B, and CLK1C). These clocks, **hereafter, are referred to** as CLK1. CLK1 is the basic clock of the Models 7/16 HSALU and 7/32. Another clock, FA1FB00, is decoded from the counter and is used primarily for controlling the enables to the DROMs and the loading of the ROM Address Register. The counter is initialized and held **in this state** by STOP0, on a power down or a power up, to inhibit the clocks. **Refer to Figure** 8 for clock timing.

The clock control logic of these Processors is shown on Sheet 33. Clock inputs to most flip-flop or registers are derived from CLK1 and are delayed from CLK1 by one and only one TTL gate delay. The clock system in the Processor is controller by clock stops. These stops prevent various functions from being performed within the machine at a given time.

Memory Stop (MSTOP0) and Input/Output Stop (ISTOP0) are the two clock stops in the Processor which inhibits all clocks in the machine. The clocks are stopped when the Processor is waiting for data from the memory or when waiting for memory to become available to begin a new memory operation. When MSTOP0 becomes active it halts all activity except the current memory operation. Likewise, during an I/O operation, when ISTOP0 is active, all clocks are inhibited until the current **I/O operation is complete.** In the case where both a memory operation and an I/O operation is specified by the same **micro-instruction, the** hierarchy of stops is **as follows; first any** previous memory operations must be completed, the I/O operation specified by the instruction is then executed, finally, upon completion of the I/O operation, the memory operation specified is started.

ROM Stop (RSTOP0) is used to prevent both incrementing the ROM Address Register and loading the ROM Data Register. This stop is activaetd by MSTOP0, ISTOP0, SKIP0, and SPSTOP0. MSTOP0 and ISTOP0 were discussed previous. SKIP0 is active during the first clock of any True Branch operation **(Figure 9)**, and SRSTOP0 is active on any Counter operation when the value in the counter is greater than one (refer to Section 5).

The Destination Stop (DSTOP0), when active, prevents loading any destination register. DSTOP0 is active when MSTOP0, ISTOP0, BRCH0, or SDSTOP0 is active. MSTOP0 and ISTOP0 were discussed previously. BRCH0 is active during any Branch operation whether or not the branch is taken and SDSTOP0 is active on any command mode operation when the Repeat flip-flop is reset (34M6). Refer to **Section 5.12 for** Command Mode operations.

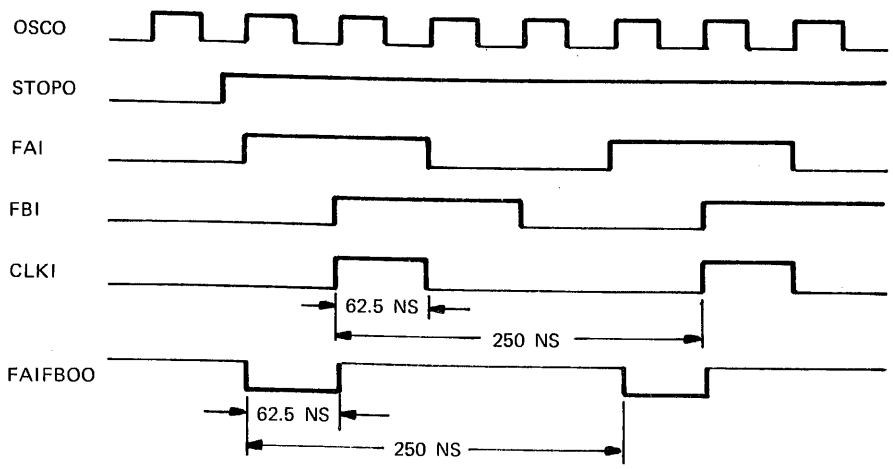


Figure 8. Nominal Clock Timing

5.3 Initialize Control

System initialization is performed by de-energizing the System Clear (SCLR) relay (32A4). This relay is de-energized as a result of one of the following conditions.

1. Placing the Processor ON-OFF-LOCK Switch in the OFF position.
2. Operating the Processor Initialize key.
3. Activating PFDT0 by the watchdog timer feature of the optional Loader Storage Unit (LSU) or other external source.
4. Activating PFDT0 from the optional Primary Power Fail (PPF) detection if the AC input level falls below a minimum operating level.
5. Loss of either +5VDC or +15VDC from the Processor power supply.

The SCLR function provides an orderly shut down of the Processor as well as a reset signal to both the memory and the Multiplexor Bus. On a power up, the SCLR relay remains deactivated until all DC voltages are in regulation. This assures predictable initial states of latched functions.

An Early Power Fail (EPF) indication is provided to the user program if Bit 2 of the PSW is set, (Bit 18 on the 7/32). This indication is provided by the micro-program by means of a machine malfunction interrupt swap.

Upon receipt of a power down indicator, PFDT0 (32C7) active, by the hardware, the one millisecond timer (32G7) is triggered. The leading edge of this pulse sets the Early Power Fail flip-flop (EPF) (32H5) which in turn enables, if PSW181 is set, a branch on machine malfunction to be taken by the micro-program. In the machine malfunction interrupt swap routine, the micro-program issues a Command Jam Alarm register (JALRM1) (34E8) active, which direct sets the Alarm flip-flop (28E8). The one output of this flip-flop is ANDed with Load PSW Low (LPSWL1) which causes the less than flag of the new PSW to be active following the machine malfunction interrupt swap. On the trailing edge of the one millisecond timer, the Power Fail (PF) flip-flop (32H7) becomes set, initiating a power down sequence.

The optional Primary Power Fail Detector (Sheet 12) monitors the AC input by sampling the secondaries of a 12VAC transformer, C1 and C3, from the Processor power supply. If the AC is lost or if the AC falls below a present level, PFDT0 and POWDN0 becomes active. PFDT0 initiates the power down sequence and POWDN0 provides a fast discharge path for Capacitors C5 and Cy which de-energizes the SCLR relay and holds the relay off in the event the AC is fluctuating about its preset power down level.

5.4 Read-Only-Memory

The Read-Only-Memory (ROM) is a high-speed, solid-state, non-destructive memory used to hold the micro-program. The ROM is organized into pages of 256 24-bit words. Each page of ROM contains six integrated circuit (IC) packages arranged such that each integrated circuit holds four-bits of each word on the associated page.

The Model 7/16 HSALU micro-program is complete in five ROM pages or 30 ROM integrated circuits. Seven additional ROM integrated circuits comprise the Decoder ROM (DROM). Seven pages of ROM are used in the Model 7/32 using 42 ROM ICs plus nine Integrated Circuits for the DROM.

Each ROM integrated circuit has two enable leads. Both enables have to be low before a read-out is obtained. If the enables are false, the four data output leads are high. Address decoding is done internal to the IC.

5.4.1 Decoder Read-Only-Memory. The Decoder Read-Only-Memory (DROM) logic of the Model 7/16 HSALU and Model 7/32 differ, refer to the strapping table for DROMs, Sheet 9. The DROM for these two machines are discussed in the following paragraphs.

The DROM for the Model 7/16 HSALU consist of several ROM integrated circuits shown on Sheet 5. Each IC contains 256 four bit words. The DROM is address by the outputs of the operation code (op-code) field of the Instruction Register (IR). Each of the 256 possible bit combinations in this op-code field address a unique word in the ROM. The DROM in the 7/16 HSALU is divided into two sections DROM1 and DROM2.

DROM1, enabled by FDEC10 (5A1), consists of four ROM ICs. The 11 least significant bits of the ROM chips labelled DROM1 (F) (5B5) are presented as inputs to the ROM Address Register (RAR) and represent an address in the micro-program of this machine. The most significant bit of this set indicates that a privileged instruction is decoded.

The two least significant bits of the ROM ICs labeled DROM1 (H) (5D5) are used to modify the control field of the next sequential instruction to be executed. Table 6 represents the bit meaning of these outputs. No other bits are used in DROM1 (H).

Table. 6.

Pin 11	Pin 12	Output Meaning
0	0	No Action
0	1	Memory Read and Increment
1	0	Instruction Read
1	1	Instruction Read and Jam CC

DROM2, enabled by FDEC20 (5E5) is used exclusively to modify the contents of the RAR and is accessed when the micro-program specifies D2 in the control field of the micro-instruction.

The Model 7/32 has three separate DROMs; DROM1 (F), DROM1 (H), and DROM2. These DROMs are addressed as discussed previously but the enabling of DROM1 is under control of PSW Bit 11. In the full-word mode of this machine, PSW111 (5A2) inactive, DROM1 (F) is enabled when D1 is specified in the control field of the micro-instruction and DROM1 (H) is enabled when PSW Bit 11 is active, halfword mode. The 11 least significant bits of all the DROMs are presented as inputs to the RAR and the most signfiicant bit of DROM1 (F) and DROM1 (H) indicates a privileged instruction when active.

In addition, Bits 9, 13, 14, and 15 can be forced low for vectoring to unique locations in the micro-program.

On an Instruction Read, FINR1A active (9H5), with no interrupts pending, INIT1 (9J4) inactive, address X'001' is forced. An Instruction Read with an interrupt pending causes location X'045' to be accessed. If a user instruction is being aborted during the execution of the instruction, GABORT1 (9J4) is forced active and during the time of a D2 on Read Halfword or Write Halfword instructions if the device is in the Byte (eight-bit) transfer mode SRA140 is forced low.

5.4.2 ROM Address Register. The ROM Address Register (RAR) (Sheet 5) is an 11-bit register which is loaded from the false SRAXX0 bus. This bus contains the data from the DROM during a Decode micro-operation and the RD register during a Branch micro-operation. The eight least significant bits of this register are arranged as a counter so that sequential ROM addresses, in a given page, may be selected.

5.4.3 ROM Data Register. The contents of the ROM from the selected address are loaded into the ROM Data Register (RD) (Sheet 7) on the trailing edge of Clock RD (CKRD0). The RD is a 20-bit register which can be thought of as the micro-instruction register. Refer to the Micro-Program Description section of the specification for the micro-instruction word format.

The RD is initialized by SCLR on a power up to an X'F00100'. This is decoded as an unconditional branch to address X'100' which starts the micro-program execution at the specified location.

5.5 Processor Registers

The majority of instructions in the micro-program are concerned with moving data from one Processor register to another. This transfer takes place by way of the 16-bit B and S Busses and modification of the data, under control of the micro-program, is done by either the Arithmetic Logic Unit (ALU) or the Shifter. Most of the Processor registers are general purpose but a few of them perform special functions. Each register is described in the following paragraphs.

5.5.1 Memory Address Register and Memory Address Slave. The memory address function of these Processors is accomplished in two steps. First, the selected address is loaded by the micro-program into the Memory Address Register (MAR) and then, the hardware copies the contents of the MAR into the Memory Address Slave (MAS) at the beginning of a memory cycle and presents this address to the memory. The micro-program is then free to modify the contents of the MAR.

Both registers are 20-bit registers and are shown on Sheets 10 and 16 (only 16-bits are provided in the 7/16 HSALU). The MAR is loaded from the S Bus whenever either the MAR or the LOC is specified as a destination and its outputs are dumped onto the B Bus if MAR is decoded as a source register or are loaded into the MAS at the beginning of a memory cycle on the leading edge of LMAS0.

5.5.2 Memory Data Register. The Memory Data Register (MDR) is shown on Sheets 18 and 19. This register is divided into two parts MDR High and MDR Low and is located on the CPU-C board LO. On a memory read operation, the MDR is first direct cleared by either CLMDH0 or CLMDL0 and then each active bit from the Memory Data Lines (MD000:160) direct sets its corresponding bit in the MDR. The MDR may also be loaded from the S Bus when it is specified as a destination register by the micro-program. When loading from the S Bus, if Cross Shift is specified, only MDR High is loaded when Bit 15 of the MAR is set and only MDR Low is loaded when Bit 15 is reset.

The outputs of the MDR are presented to the memory during the write portion of a memory cycle, to the B Bus if MDR is a source register and, to the B inputs to the ALU when MDR is specified as the second source.

On the Model 7/32 four additional bits of the MDR are provided, XMDR (9K7). This extention to the MDR is loaded from Bits 12:15 of the MDR during Phase two of a Calculated Address (CA) (Section 5) or when an Extended Read (ER) is specified by the micro-program.

5.5.3 Instruction Register. The Instruction Register (IR) is a 16-bit register which stores the user instruction presently being executed. The IR is divided into three parts or fields; OP code (Bits 0:7) YD field (Bits 8:11), and YS field (Bits 12:15). The IR is loaded from the Memory Data Bus, by the hardware, on an Instruction Read. Refer to Figure 9 for timing information.

The op code field (Sheet 9) contains the encoded instruction to be performed. Its outputs are decoded by the hardware and presented as address to the Decoder Read-Only-Memory. All 256 combinations have unique entry points in the micro-program.

The YD field is defined as the user destination field. YD selects one of the General Registers, in the Processor, in which the result of the user instruction is to be stored. This portion of the IR (Sheet 21) is arranged as an up/down counter. If YDP1 is specified as either the source or destination of a micro-instruction, the YD field of the IR is incremented by one at the end of the instruction. Likewise, if YDM1 is specified, YD is decremented by one.

YS is the user source field of the instruction being emulated. The second operand of the instruction is contained in the General Register specified by YS for RR format instructions. This field also contains the number of the General Register being used as the index register on an RX or an RS instruction or the actual hexadecimal number in a short form instruction. Refer to User's Manual, Publication Number 29-261 or Model 7/32 Reference Manual, Publication Number 29-399, for instruction format information.

The YS field of the IR can also be loaded from the S Bus by the micro-program if YSI is the selected destination register. The three fields of the IR can be selected separately as source registers by the micro-program.

5.5.4 Arithmetic Register. Two 16-bits Arithmetic Registers (ARH and ARL) (Sheets 21 and 22) are available to the micro-program to be used as second operand on the Add, Subtract, OR, AND, or Exclusive OR micro-instructions. These registers may be loaded from the S Bus when they are specified as the destination register of a micro-instruction. If AR is specified as the destination both ARH and ARL are loaded. Bits 0:11 of ARH are forced inactive and Bits 12:15 of ARH and Bits 0:15 of ARL loaded from the XS Bus and S Bus respectively. The output of the registers are presented to the 'B' inputs to the ALU when they are specified as the second source by the micro-instruction.

5.5.5 Flag Register and Condition Code. The Flag Register (FLR) (Sheet 30) is a four bit register which contains the Carry flag (C), the Overflow flag (V), the Greater than flag (G), and the Less than flag (L). The outputs from the FLR are copied into another four bits register, the Condition Code, at the end of each user instruction being emulated. These flags represent results of user instructions which cannot be indicated otherwise.

The FLR is loaded from the S Bus whenever either the FLR or the Program Status Word (PSW) register is specified as a destination. The contents of FLR is copied into the Condition Code on an Instruction Read (see Figure 9) or on a command if JAM CC is specified. The outputs from the Flag Register are also used by the Branch Circuit (Sheet 31) for conditional branches. The contents of the CC are copied onto the B Bus (Bits 12:15) when the PSW is specified as the source register.

The following conditions also modify the FLR:

1. Carry Flag - The C flag changes on any micro-instruction except Load I/O if carry out is specified. It sets if B Bus Bit 0 is set on a Shift Left, if B Bus Bit 15 is set on a Shift Right, if Carry Save (CSV1) from the ALU is set on an Add, or if CSV1 is inactive on a Subtract. For all other cases the C flag is reset.

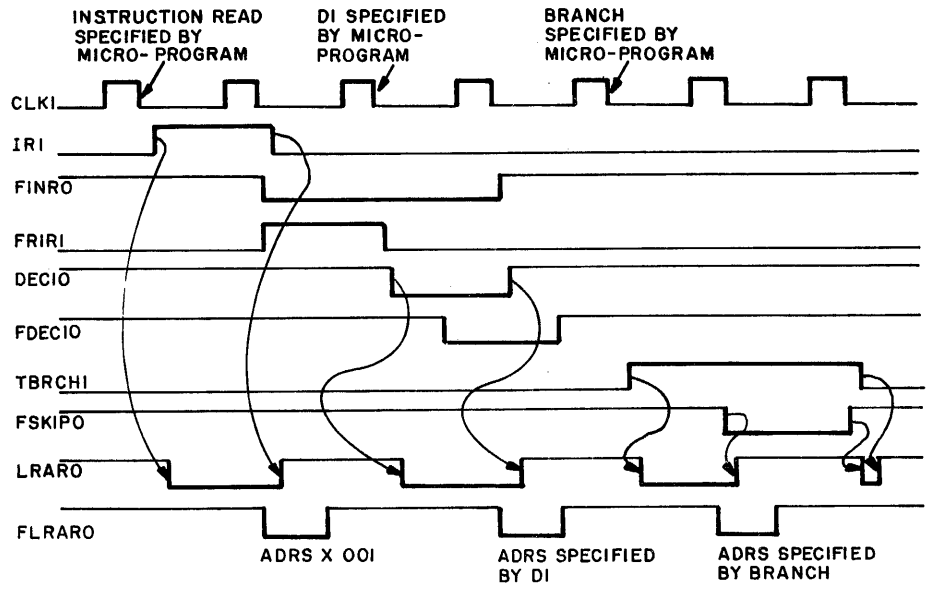


Figure 9. Micro-Program Decode and Branch Operations

2. Overflow Flag - The V flag is direct set if false sync is detected on an I/O operation and is changed on any micro-instruction except Branch, Command, or Load I/O if Flags (F) are specified. The V flag is set on an Add if the sign of the number of the B Bus is positive and the sign of the B Bus is the same as the sign of the A Bus (Arithmetic Register) and the resulting sign (S Bus) is negative or the number on the B Bus is negative and the sign of the B Bus is again the same as the sign of the A Bus and the result is positive. This flag is also set on a Subtract operation if the sign of the B Bus is positive and the signs of the B Bus and A Bus differ and the result sign is negative and the B and A bus signs differ and the result is positive. For all other combinations of A, B, and S Bus signs on Adds and Subtracts, the V flag becomes reset, The following Boolean expression also defines the setting of the V flag:

$$V = ADD \cdot B000 \cdot \overline{(B000 \oplus GA000)} \cdot S000$$

$$+ADD \cdot B0000 \cdot \overline{(B000 \oplus GA000)} \cdot \overline{S000}$$

$$+SUB \cdot \overline{B000} \cdot (B000 \oplus GA000) \cdot S000$$

$$+SUB \cdot B000 \cdot (B000 \oplus GA000) \cdot \overline{S000}$$

3. Greater than and Less than - These flags change on any micro-instruction except Branch Command or Load I/O as long as F is specified. The G flag is set if the result of the operation (S Bus) is positive or if the result is zero and either the G or L flag was set from a previous operation. The L flag is set if the resulting sign is negative. Either flag is reset if these conditions are not met.

5.5.6 Register Stacks. The register stacks in these machines are located on Sheets 12 and 13. Each IC used in these stacks are four bit by 16 word register files. Each stack has four select lines which are decoded internally to select one of 16 words. Associated with each IC are also two control leads Memory Enable (ME) and Write Enable (WE).

When the ME control line is at a low state, the outputs represent the contents of the selected word in the register file. During the time that the WE control line is active the selected register file is loaded from the S Bus. Each machine cycle is divided into a Read portion, READ1 active (11J3), and a Write portion, READ1 inactive. During the time that READ1 is inactive, data to the ALU is latched-up by the B Bus Shifter/Latch circuit (Section 5.6) to prevent changing the B Bus during the time that the outputs of the register stacks may be changing. Refer to Figure 10 for timing information.

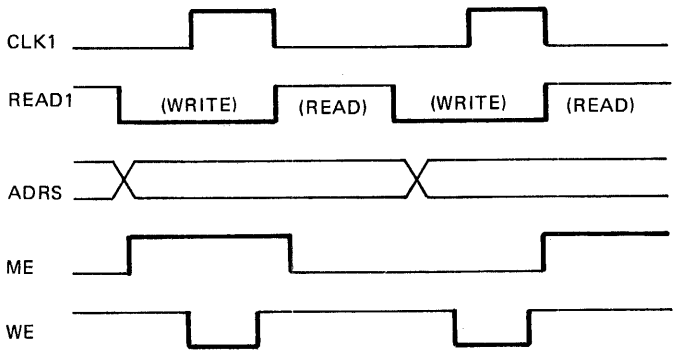


Figure 10. Register Stack Timing

The Model 7/16 HSALU uses eight register stack ICs in the register stack, four for the micro-registers MR0:6 and PSWL and four for the sixteen 16-bit General Registers. Twenty-two register files are used in the Model 7/32 four for the micro-register and the remaining 18 for the two sets of 32 bit General Registers.

The Memory Enable logic for the register stacks is found on Sheet 11. The AND gate on the left half of each AND/OR gate is used for source decoding and the AND gate on the right hand side is used for destination decoding. ROM data bits are used to address micro-registers while the YS and YD fields of the Instruction Register (IR) is used for addressing the General Registers. In the 7/32, selection of General Register Set 0 or 'F' is under control of PSW Bit 27 (11F2).

5.6 Shifter and Latch

The Shifter/Latch circuit (Sheet 20) takes the data on the B Bus, manipulates that data and stores it (when READ1 is inactive) prior to presenting it to the A inputs of the ALU. The shifter can load, shift left, shift right, or cross shift the B Bus data. The function performed is determined by the state of the A and B inputs to the eight shifter ICs. A truth table defining these inputs is provided on Sheet 1A. The 19-073 ICs which comprise the shifter are tri-state devices. When the S inputs to the shifter are at a logical ZERO level, the chips are enabled and the specified function is performed. If these inputs are high, the outputs of the shifter are disabled and assume a high impedance state.

The cross coupled flip-flops at the output of each state of the shifter latch the data on the Gated B Bus (GB000:150) during the time that READ1 is inactive. This prevents the data, which may be changing on the B Bus at this time, from being felt at the inputs to the ALU.

5.7 Arithmetic Logic Unit (ALU)

The Arithmetic Logic Unit (ALU) consists of four 19-039 four-bit ALU packs and one 19-046 Carry Look-Ahead pack. The ALU is shown on Sheets 21 and 22. In the Model 7/32 the ALU is extended to 20-bits for address calculations by additional ALU IC (10N3).

FUNCTION	M	S_0	S_1	S_2	S_3
LOAD	1	1	1	1	1
AND	1	0	1	1	1
OR	1	1	1	0	1
XOR	1	1	0	0	1
ADD	0	1	0	0	1
SUB	0	0	1	1	0
CMD	0	1	0	0	1

Each 19-038 ALU pack develops four-bits of the low active S Bus. The internal Carry Propagated (CP) from the most significant stage of the ALU pack and the Carry Generated (CG) for the most significant stage (CPXX1 and CGXX1) are applied to the 19-040 Carry Look-Ahead pack to develop the Carry into the next more significant ALU pack (CNXX1). Only the carry output of the most significant ALU chip is used (CSV1) (22B8). Each function of the ALU that is used is described inthe following pargaraphs. All gate references are to the arbitrary labels on Figure 11. The mnemonics indicated are the actual symbols used as reference on the ICs (22N7).

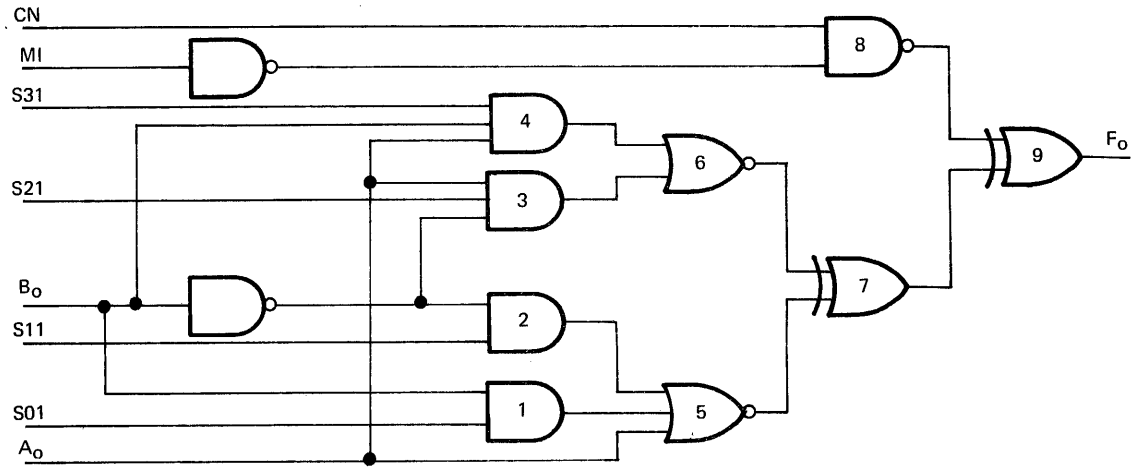


Figure 11. Least Significant ALU Stage

5.7.1 Load. The ALU is conditioned to the Load mode on any Load micro-instruction. In this mode, Gates 1, 2, 3, and 4 are enabled by S01, S11, S21, and S31 respectively, and Gate 8 is disabled by M1. Since both Gates 1 and 2 are enabled, at least one of their outputs are high producing a low at the output from Gate 5. The state of Gate 6 is the inverse of $\overline{Ao}$. If $\overline{Ao}$ is low, the output of Gate 7 is high and the output of Gate 8 is low ($\overline{Fo}$). For $\overline{Ao}$ high, the inverse is true at each stage causing $\overline{Fo}$ to also be high. Therefore, in this mode, the state of $\overline{Fo}$ is the same as the state of $\overline{Ao}$ independent of the $\overline{Bo}$ input. The state of the Gated B Bus is passed, unmodified, to the S Bus.

5.7.2 AND. The AND function produced by the AND micro-instruction conditions the ALU to logically AND each bit of the Gated B Bus with the gated outputs of the AR. In this mode the output equation for Gate 5 is (Bo·Ao) and the output equation for Gate 6 is (Ao). The simplified expression for the output from Gate 7 is then (Ao · Bo). Since Gate 8 is disabled by the M1 input to the ALU, its output is high causing the output from Gate 9 to be defined by the same equation as the output from Gate 7, the AND function.

5.7.3 OR. The OR micro-instruction causes each bit from the B Bus to be logically ORed with the corresponding bit from the gated output of the AR. Gate 5 produces a low because of the complimentary Bo inputs. The outputs equation for Gates 6 and 7 is (Ao + Bo) which corresponds to the $\overline{Fo}$ output from Gate 9.

5.7.4 Exclusive OR. The Exclusive OR micro-instruction produces a logical low at the S Bus if the corresponding bits on the Gated B Bus and the gated outputs of the AR are at different logic levels. The expressions for the outputs from Gates 5 and 6 are (Ao · Bo) and (AoBo) respectively. The function of the output from Gate 7 is; therefore, $Ao\overline{Bo} + \overline{Ao}Bo$, the Exclusive OR function. Since, once again, the output from Gate 8 is high, $\overline{Fo}$ is the same as the output from Gate 7.

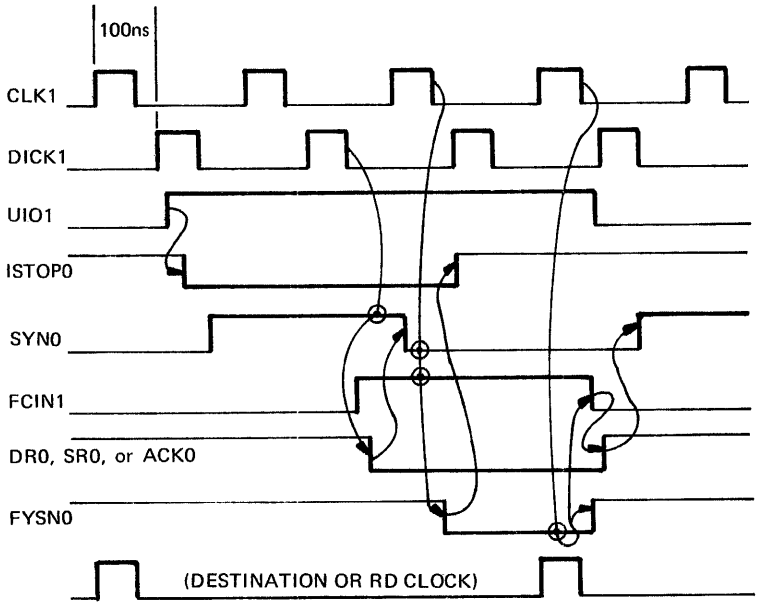
5.7.5 Add. The ALU is conditioned to the Add mode on either a command or an Add micro-instruction. Note that with the exception of the M1 control line, Add is the same as Exclusive OR. The M1 control line enables the carry network internal to the ALU device so that the output from Gate 8 is CN. Fo now becomes CN ($Ao\overline{Bo} + \overline{Ao}Bo$) + $\overline{CN}$ (AoBo + AoBo). Figure 11 shows only the least significant stage of the 19-039 four bit ALU. The next three stages are identical except for the internally propagated carry.

5.7.6 Subtract. The Subtract function produced by the four-bit ALU device is A-B-1. For this reason, the carry in to the least significant stage is inverted by the Exclusive OR gate (23B3) on a Subtract micro-instruction. The output equation for Gate 5 is (Ao · Bo) and the equation for Gate 6 is ($Ao + \overline{Bo}$). Gate 7 produces a high output when the equation (Ao · Bo + AoBo) is satisfied. The output function, $\overline{Fo}$ = CN $\overline{(Ao\overline{Bo})}$ + $\overline{CN}$ (AoBo + $Ao\overline{Bo}$), yields A-B.

5.8 I/O Control

An I/O operation is initiated if I/O is the Source or Destination of a Load micro-instruction. The I/O control logic is shown on Sheet 25. If I/O is a source, then an input operation is initiated if I/O is a destination, an output operation is indicated. I/O timing is discussed separately for input and output.

5.8.1 Input. Refer to Figure 12 for input timing information. When I/O is specified as a source, unload I/O (UI10) (25N5) is decoded and ISTOP0 goes active. On the trailing edge of the next Delayed Clock (DICK 1) the Control In flip-flop (CIN) (25L7) sets. On receipt of SYN0 (26L1) or the detection of False Sync, the 14 millisecond timer (25N7) timed out, the Sync flip-flop (25J7) sets which deactivates ISTOP0. On the trailing edge of the next CLK1 the destination register is loaded and both the Control In and Sync flip-flops become reset completing the operation.



*SYN0 is active due to a previous I/O instruction.

Figure 12. I/O Input

5.8.2 Output. Refer to Figure 13 for output timing information. An I/O out operation is very similar to the input operation. When Load I/O LDI01 (25N5) is detected, ISTOP0 goes active and on the leading edge of the next CLK1 the Data flip-flop, FDAT1 active (25F8), sets, the output of the Data flip-flop is used to gate data to the Data Bus, D000:160, (Sheet 15). On the trailing edge of the next DICK1 the Control Out flip-flop (25E7) is set which activates the specified output control line. The output operation now progresses in the same manner as the I/O input discussed previously.

To insure a minimum width of 350 nanoseconds on ADRS0 the FSR0 strap option (25H7) may be removed. Exercising this option allows the Catch flip-flop (25H7) to set on an address operation. The 0 output of this flip-flop delays the setting of the Sync flip-flop by 250 nanoseconds. All other sequencing is as discussed previously.

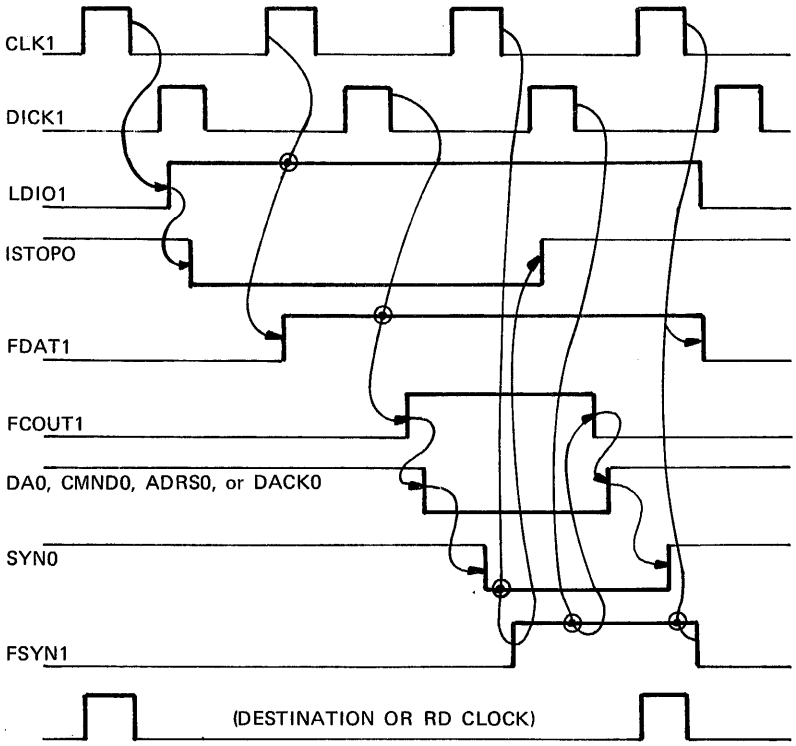


Figure 13. I/O Output

5.9 Memory Control and Timing

The memory control logic in the Processor is found on Sheet 27. Refer to Figure 14 A and B for memory control and timing information.

There are two different memory systems associated with this Processor, local memory and extended memory. Only local memory can be used with the Model 7/16 HSALU or the Model 7/32 without either the Memory Access Controller or the Direct Memory Access Bus Controller options. In addition, for local memory, two difefrent memory timings are provided under option control, depending on whether 1 microsecond or 750 nanoseconds memories are equipped.

Local memory timing is initiated by a Direct Memory Access request, REQ0 active, or a Processor request, MEM1 active, either of which activates Set Local Memory Busy (SLMBY1) (27M4). On the leading edge of CLK1, with SLMBY1 active, the Early Read (ER) flip-flop (27E5) is direct set. The setting of the ER flip-flop triggers the local memory timing (refer to the Local Memory Timing Diagrams Figure 14.A and B).

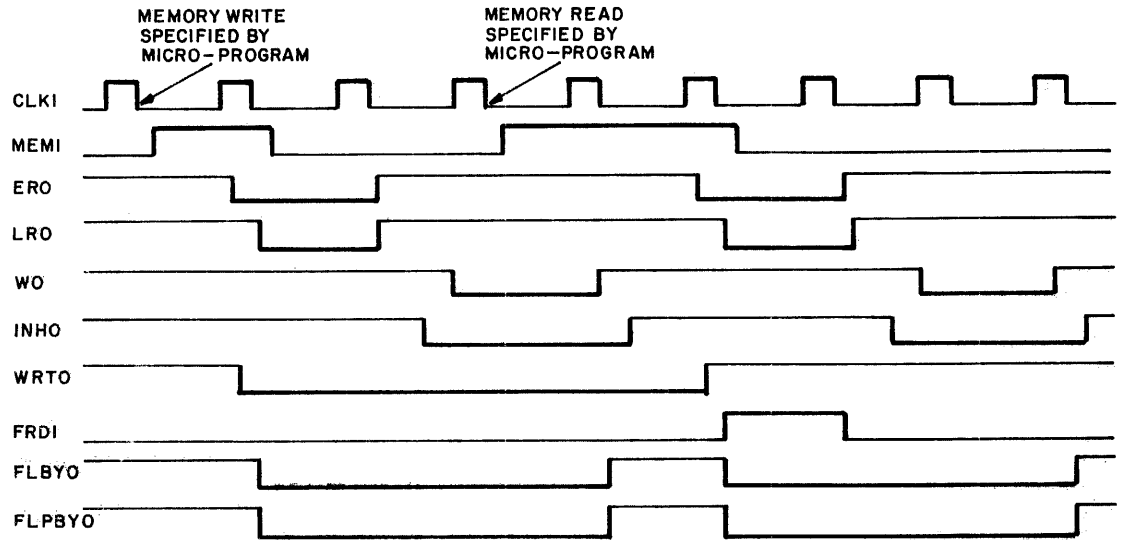


Figure 14.A Processor to Local Memory (1.0 Microsecond Memory)

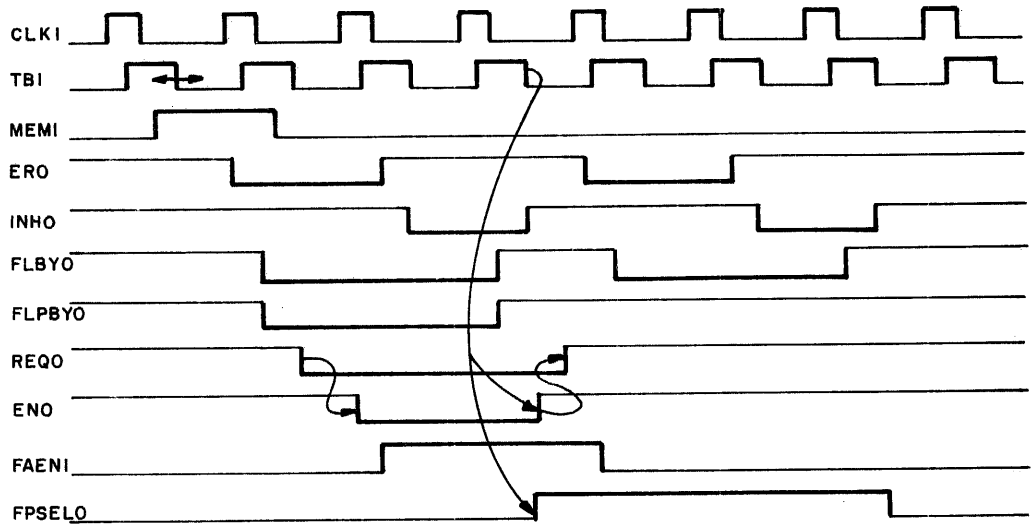


Figure 14.B Processor and DMA to Local Memory (750 Nanoseconds Memory)

On the trailing edge of the same clock which sets the ER flip-flop, the Local Busy (LBSY) flip-flop (27L5) is set, the Local Processor Busy (LPBSY) flip-flop (27K5), if the request for memory was from the Processor is set, and the Read (RD) flip-flop (27G5) for a Processor read operation is set. These flip-flops are used for controlling memory timing and clock stops within the Processor. The Read flip-flop is set for one clock and is used to indicate Data Unavailable (DU) (generate MSTOP0 if the MDR is specified as the source) when it is set. The two busy flip-flops are set for two clocks for 750 nanoseconds memory timing and three clocks for 1.0 microsecond memory timing. The Write (WT) flip-flop (26G7) is set on the leading edge of ER0 for write operations and reset on the same edge for read operations. This flip-flop is used to indicate to the memory system that a write operation is in progress and to inhibit the generation of Enable Memory Strobed data (ENMS1) (27 G9) when set.

A Direct Memory Access device (MAC, SELCH, etc.) requests a memory cycle by activating Request (REQ0) (27K1). On the leading edge of the first CLK1 after REQ0 becomes active. The Request flip-flop (27J3) is set thus activating the Enable (EN) flip-flop (27J5) and generating the EN0 control line to the DMA Bus. When the Enable (EN) flip-flop is set and memory is not busy, the Processor Select flip-flop (27H5) and the Enable flip-flop are reset, initiating a DMA memory cycle.

When the Processor attempts a memory operation beyond Bank 0 (the first 256KB of memory) on a Model 7/32 equipped with either a MAC or DMABC, XMEM0 is active. XMEM0 prevents the generation of local memory timing and sets the Extended Processor Busy (XPBY) (27N5) flip-flop. The MAC or DMABC then activates Extended Data Unavailable (EXDUA0) and Extended Busy (EXBYS0) until the memory cycle is complete. When the XPBY flip-flop is set a DMA device is still able to initiate a memory cycle to local memory. Refer to Figure 15.

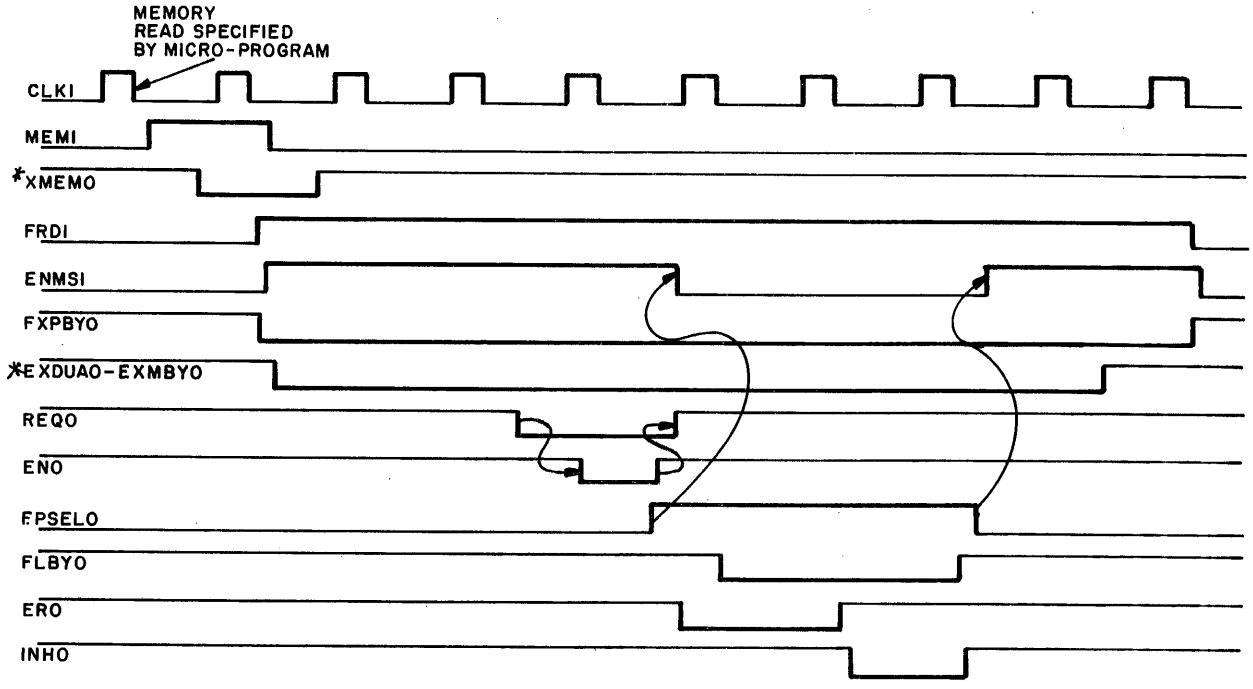


Figure 15. Extended Memory Timing and DMA to Local Memory

5.10 X Bus Logic (Model 7/32 only.)

On the Model 7/32,the S and B Busses are extended to 20 bits for the purpose of address calculations. This logic is found on Sheets 9 and 10.

An additional four bit ALU (10N9) is appended to the standard 16 bit ALU to provide this 20 bit arithmetic and logic capability during a single machine cycle. Carry Save (CSV1) (10L9) is used as the carry input to this ALU. The Flag Register (Section 5.5.5)is not modified as a result of any operation on the X Bus. The operation of this logic including its associated registers has been covered in more detail in their individual sections described previously.

5.11 Calculate Address (Model 7/32 Only)

The Calculate Address (CA) micro-instruction is used to generate an effective address of the second operand of a user instruction and load that address into the Memory Address Register. This micro-instruction requires a minimum of three machine cycles for its execution but could require more depending on the instruction format and the access and cycle time of the memory being used.

A phase counter (Sheet 8) is used to keep track of where, within the execution of the CA micro-instruction, the machine is at a given time. Refer to Figures 16A and 16B. Depending on the instruction format of the user instruction being executed different functions are performed. Refer to the 32 Bit Reference Manual, Publication Number 29-365, for descriptions of the various Memory Indexed (RX) instructions. The two most significant bits of the current contents of MDR determines the format being used.

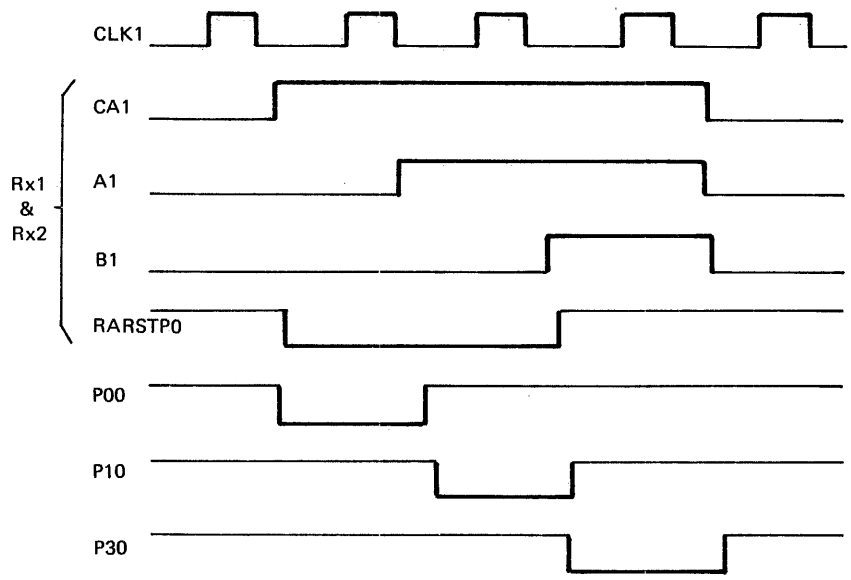


Figure 16A. Calculate Address RX1 or RX2

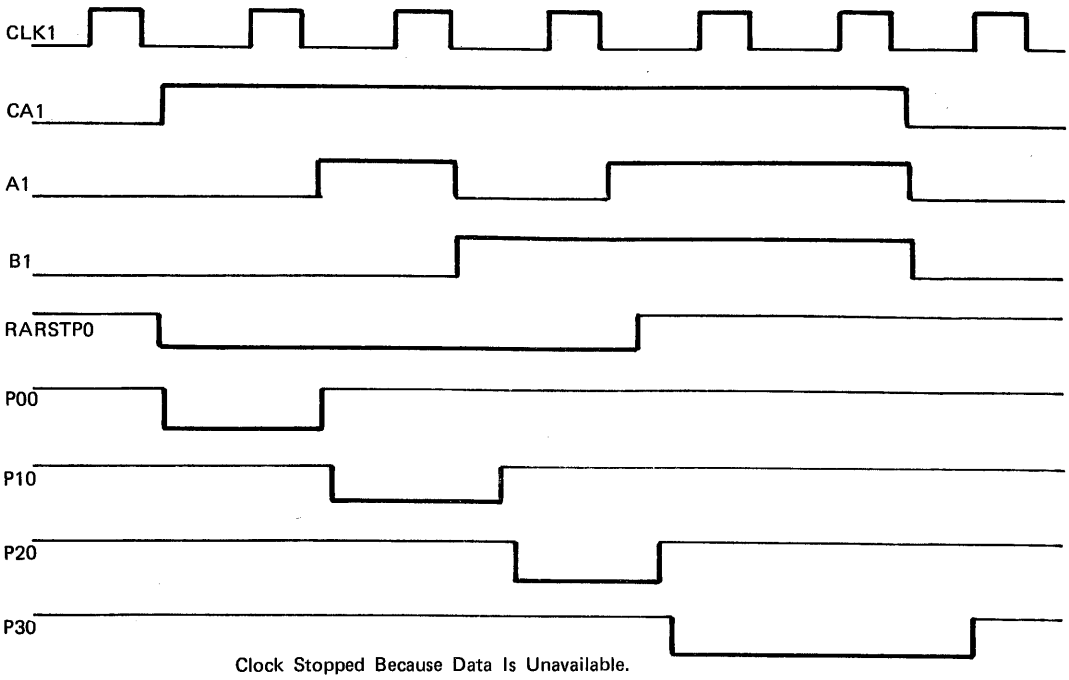


Figure 16B. Calculate Address RX3.

Phase zero, independent of the instruction format, is only used to decrement the ROM Address Register by one so that it points to the address of the CA micro-instruction. This is necessary because, set RD bits generated by the Calculate Address (CA) logic are ORed with the data from the ROM to modify the micro-instruction during the various phases of the CA.

During Phase zero of the CA instruction RARSTP0 is forced active. This prevents incrementing the RAR. Phase zero is also used to inhibit the RD clock (7D9) decremented the RAR by one (SK1) by generating a clock on the count up input to the RAR (negative logic is used). The Calculate Address instruction conditions the ALU to the Add mode. The following describes the execution of this instruction for each of the possible formats. Refer to Table 7 during the description.

TABLE 7. RX FORMATS FOR CALCULATE ADDRESS

FORMAT	PHASE	CONTENTS OF RDR	OPERATION
RX1	0	X'60A417'	Decrement RAR
RX1	1	X'60A417'	MDR→MAR
RX1	3	X'60BE17'	MAR + *AR→MAR
RX2	0	X'60A417'	Decrement RAR
RX2	1	X'60A417'	MDR + LOC→MAR
RX2	3	X'60BE17'	MAR + *AR→MAR
RX3	0	X'60A417'	Decrement RAR
RX3	1	X'60A417'	MDR04:07→YSI
RX3	2	X'61F617'	*AR + **YSLX→AR, MDR12:15→XMDR
RX3	3	X'60B617'	*AR + MDR→MAR

*Conditional on FAMOD
** Conditional on AMOD

5.11.1 RX1. The contents of the RDR during Phase one is an X'60A417'. This is normally decoded by the hardware as (A MAR, LOC, MDR). Gated RX1 or RX3 (GRX130) (8N8), however, disables the B Bus source causing the contents of MDR to be loaded into the MAR. Phase one also causes set RD bits (SRD110, SRD120, and SRD140) to go active which changes the contents of the RDR to an X'60A417' when Phase three is entered. In Phase one, for RX1, and RX2, RARSTP0 is deactivated which allows the RAR to increment to the next sequential instruction. Phase three executes the indicated operation (A MAR, MAR, AR). Since this instruction performs the indexing, the second source, AR, is conditional on the Address Modification (FAMOD0) (29N4) flip-flop. If the YS field of the user instruction contains all zeros the AMOD flip-flop is reset following an Instruction Read and Disable Source Two (DS20) (9K8) is active

5.11.2 RX2. The CA micro-instruction for a RX2 format works the same as the RX1 format except the operation specified by RDR in Phase one (A MAR, LOC, MDR) is performed, GRX130 inactive. The sign bit, MDR011, is extended through the XB-Logic (10N3) since the value in MDR is in the two's complement form.

5.11.3 RX3. Phase one, for RX3 formats, does not change the contents of MAR, RX3P10 active (14K1). Phase one jams Bits 7, 9, 11, and 14 of the SR0 lines causing an X'61F67' to be loaded into the RDR in Phase two. This is decoded as (A AR, YSLX, AR, MRI). RX3P10 also loads MDR04:07 into the YS field of the IR (29G2). In Phase two the first and second level index values are added and loaded into the AR. The second source is conditional on FAMOD0 as described in Section 5.11.1. Phase two, in addition to performing the indicated function, also loads MDR Bits 12 through 15 into the Extended Memory Data Register (XMDR) (9K9) and jams SRD Bits 11 and 14 which causes an X'60B67' to be loaded into RDR in Phase three. Phase three then executes the indicated function (A MAR, MDR, AR).

5.12 Counter Dependent Operations

A Command micro-instruction with RD Bit 6 set implies a counter dependent mode of operation that is maintained until the Counter Register (CTR) is zero. Command Repeat causes the next micro-instruction to be repeated the number of times specified in the CTR. Command Multiply and Command Divide cause the Command itself to be repeated with SRH forced to be the Source and Destination Register. These modes are implemented by the circuit shown on Sheet 34.

5.12.1 Repeat. When a Command Repeat micro-instruction is executed, CMND1 goes high (34N2). The Repeat flip-flop (REPT) (34N5) toggles set on the trailing edge of CLK1. On the same edge, the Counter Mode flip-flop (CMODE) will toggle set if the CTR is not zero (CEMT0) (34K6). If the CTR is zero, the next micro-instruction should be not executed. Since, at the same time RPT flip-flop toggles set, the next micro-instruction toggles into RD, the Processor has to execute the instruction. However, SDSTOP0 goes active (34N9) stopping the Destination Clock. No register is modified, nor is the FLR changed. RPT toggles (34N9) reset on the next CLK1.

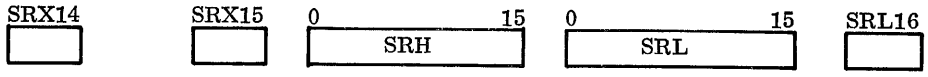
If the CTR is not zero, CMODE and RPT both toggle set. The target micro-instruction is in RD. Because CMODE is set, DECTR0 (34M9) goes low until the CTR is zero. The CTR will decrement on each Destination Clock (DC). As long as CMODE is set and the CTR does not equal zero or one (CTONE0) (34L9), SRSTP0 is low (34L9), disabling CLKRD0 and RDSTP0 (19L9). The RAR will not increment nor will another micro-instruction be strobed into RD until the Counter (CTR) decrements from one to zero. As soon as the CTR decrements from one to zero, CMODE and RPT both toggle reset, SRSTP0 is high and the next sequential micro-instruction is executed.

5.12.2 Multiply. Prior to executing the Command Multiply micro-instruction, the following preliminary conditions are assumed:

1. The MDR contains the multiplicand in two's compliment form.
2. ARL contains twice the multiplicand in two's compliment form.
3. SRL contains the multiplier in two compliment form.
4. The Carry flag (FLR121) contains the most significant bit of the multiplicand.

These registers are loaded by the micro-program.

SRH and SRL with a two bit extention on the left and a one bit extention on the right is used to contain partial products. When multiplication is complete SRH and SRL contains the 32 bit result.



SRL16 is used to remember the last bit shifted out and SRX14, SRX15, SRH, and SRL are used to contain the partial products (34 bits are required to represent signed partial products in two's compliment form). The least significant 18 bits of the first partial product are zero, SRL contains the 16 bit multiplier. SRL14, SRL15, and SRL16 are used to decide which operation is to be performed. After calculating a parital product, an arithmetic shift of two places is performed of the entire partial product. This procedure is repeated until all multiplier bits are shifted out.

The algorithm used in the multiplication instruction requires 10 machine cycles for its execution. The first clock of the instruction, Clear Shift Register High (CSRH1) (36C7) active, is used to clear SRH, SRL16, SRX14, and SRX15 and load the Counter with an X'09'. The next eight clocks perform the multiplication by executing either an Add or Subtract operation on the partial product contained in SRX and SRH and either the multiplicand contained in MDR, twice the multiplicand contained in ARL, or zero depending on the result of the shift operation SRL14:16. Refer to Table 8.

TABLE 8. OPERATION CODE

SRL14	SRL15	SRL16	OPERATION
0	0	0	ADD Zero to SRH
0	0	1	ADD Multiplicand to SRH
0	1	0	ADD Multiplicand to SRH
0	1	1	ADD 2x Multiplicand to SRH
1	0	0	SUB 2x Multiplicand from SRH
1	0	1	SUB Multiplicand from SRH
1	1	0	SUB Multiplcand from SRH
1	1	1	SUB Zero from SRH

On the last clock of the multiplication, the operation to be performed differs depending on whether a signed multiplication C MPY (RD091 inactive) or an unsigned multiplication C UMPY (RD091 active) is being executed.

If Signed Multiply:

L SRH, SRH, SR is performed.

If Unsigned Multiply:

L SRH, SRH, SR is performed if SRL161 is inactive.

or

A SRH, SRH, MDR, SR is performed if SRL161 is active.

Shift Register High (SRH) is conditioned to the load state during the first clock of the pulse pair of SHCLK0 and to the Shift Right mode during the second clock. The second shift is performed by the fact that the B Bus Multiplexor (Sheet 20) is conditioned to the Shift Right mode. SRL is conditioned to the Shift Right state for both clocks. Refer to timing diagram Figure 17.

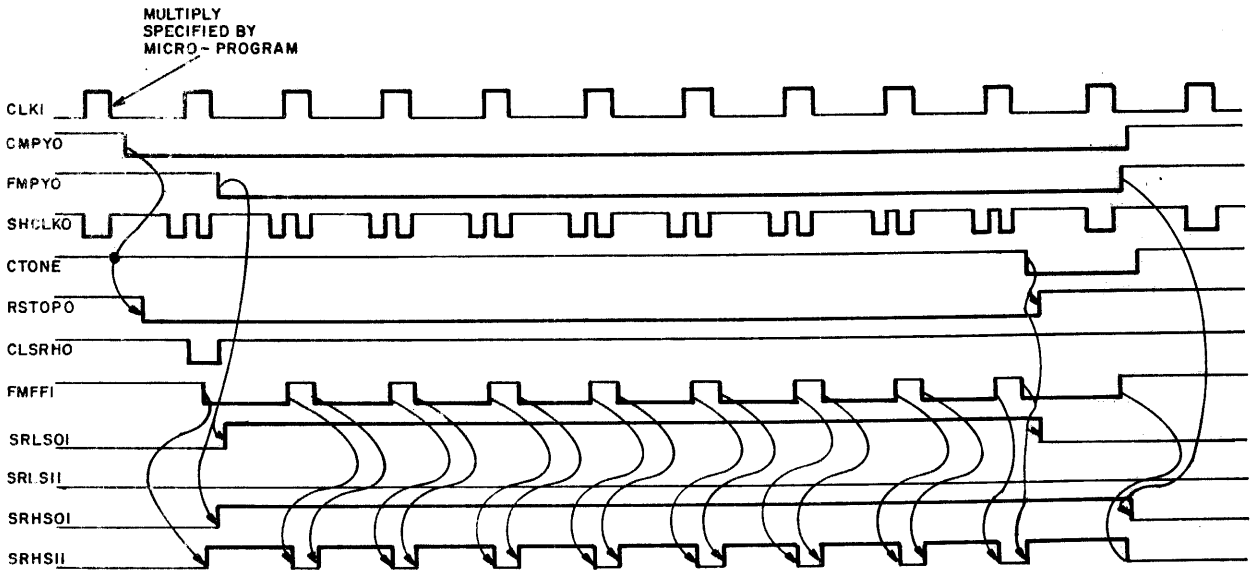


Figure 17. Multiply Timing

During Command operations the ALU is conditioned to the ADD mode. If a Subtract operation is required, Force Subtract (FRSUB0) (36M4) is activated, which conditions the ALU to the Subrtact mode. A four-bit Adder (36G5) is used to allow a 20-bit parallel operation to be performed during the execution of the Multiply algorithm. The 'A' input to this adder is SRX and the B input is a function of the Carry flag.

Table 9 shows the hexadecimal contents of each of the registers during each clock of the multiplication of 5*4. Prior to execution, MDR contains a X'5', ARL contains a X'A', and the Carry flag is reset.

Table 10 shows another example of a multiplication (X'1111'*X'1111'). Prior to execution, MDR contains a X'1111, ARL contains a X'2222' and the Carry flag is reset.

Refer to Appendix 1 for additional examples of multiply routines.

TABLE 9. MULTIPLICATION EXAMPLE (5 x 4)

	SRX14		SRH			SRL				SRL16
1	0	0	0	0	0	0	0	0	4	0
2	0	0	0	0	0	0	0	0	1	0
3	0	0	0	0	2	4	0	0	0	0
4	0	0	0	0	0	5	0	0	0	0
5	0	0	0	0	0	1	4	0	0	0
6	0	0	0	0	0	0	5	0	0	0
7	0	0	0	0	0	0	1	4	0	0
8	0	0	0	0	0	0	0	5	0	0
9	0	0	0	0	0	0	0	1	4	0
10	0	0	0	0	0	0	0	1	4	0

TABLE 10. MULTIPLICATION EXAMPLE (X' 1111' x '1111')

	SRX		SRH			SRL				SRL16
1	0	0	0	0	0	1	1	1	1	0
2	0	0	8	8	8	4	4	4	4	0
3	0	0	2	2	2	1	1	1	1	0
4	0	0	9	1	1	8	4	4	4	0
5	0	0	2	4	4	2	1	1	1	0
6	0	0	9	1	9	C	8	4	4	0
7	0	0	2	4	6	3	2	1	1	0
8	0	0	9	1	A	0	C	8	4	0
9	0	0	2	4	6	4	3	2	1	0
10	0	0	1	2	3	4	3	2	1	0

5.12.3 Divide. Prior to executing the Command Divide micro-instruction, the following preliminary conditions are assumed: The 32-bit positive dividend is in SRH and SRL, the divisor is in two's complement negative form in the ARL, the carry flag is reset, and the CTR contains 16. The Command Divide executes in 16 machine cycles. On each DCL0: the CTR is decremented, SRL is shifted left one position, FLR12 is updated from the ALU carry, and SRH is either shifted left one position or loaded from the S Bus. Refer to Figure 18.

The signal CDIV0 is low during the Command Divide. CDIV0 causes DECTR0 to go low until the Counter (CTR) decrements to zero. SRL is conditioned to the Shift Left mode, and SRH assumes the Load mode or the Shift Left mode depending upon the state of the ALU Carry (CSV1).

The most significant 16-bits of the dividend in the SRH are present on the B Bus. The data is shifted left one position by the B Bus Shifter and presented to the ALU. Note that SRL001 is carried into CISL0 (36R5). The ARL is unloaded to the ALU and an 'Add' is performed.

By adding the two's complement from the divisor in the ARL to the most significant half of the dividend in the SRH, the result on the S Bus is actually the difference between the two. If this 'trial subtraction' is successful, the ALU produces a Carry (CSV1 high), and the partial remainder on the S Bus is gated into the SRH. If the 'subtraction' is unsuccessful, CSV1 is low, and the SRH is shifted left one position; the S Bus data is ignored. The CSV1 signal is shifted into SRL15 to form the quotient bits. CSV1 is also saved in FLR12. After the Command Divide, the remainder is in the SRH and the quotient is in the SRL.

Refer to Appendix 1 for an example of a divide operation.

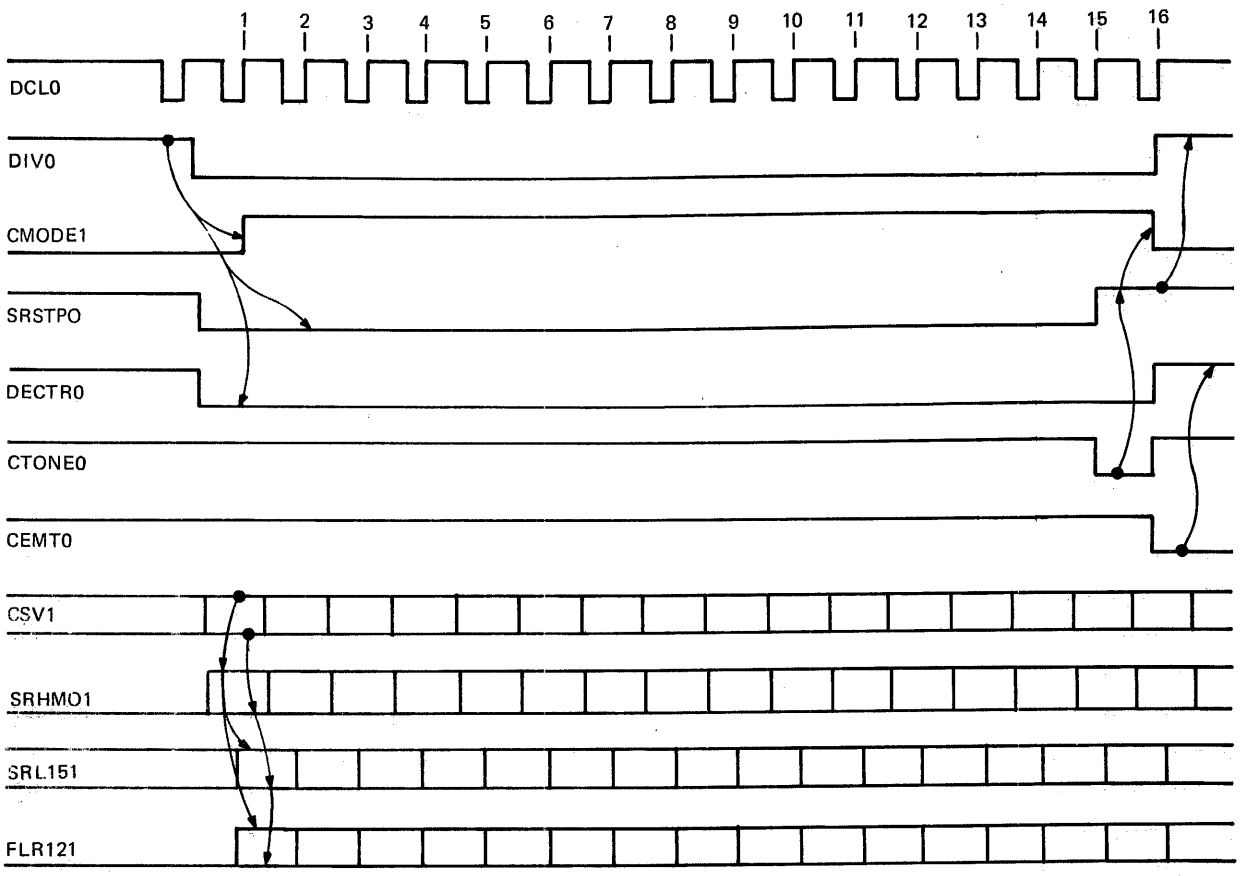


Figure 18. Command Divide Timing Diagram

5.13 Display System

The Display System provides, if the Hexadecimal Display Panel is present, a means for reading the contents of all the system registers and any core memory location, together with the capability of manually entering data and programs. Figure 19 shows the Hexadecimal Display Panel layout. Within the Hexadecimal Display Panel are five eight-bit byte Display Registers, D1 through D5, that hold data output from the Processor, and a 20-bit Switch Register which stores data input from the Hexadecimal Keyboard.

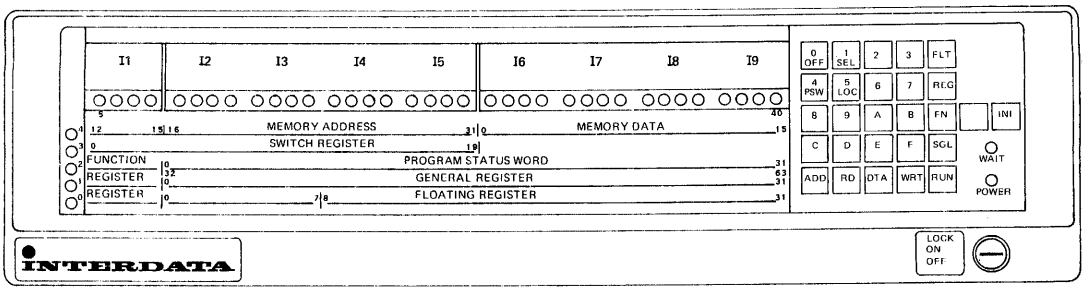


Figure 19. Hexadecimal Display Panel

Associated with each Display Register D1 through D4 are eight indicator lamps that provide a binary read-out and two optional hexadecimal read-out indicators. Associated with Display Register 5 are four indicator lamps for binary display and one optional hexadecimal read-out indicator.

The most significant four bits of Display Register D5 (Bits 0:3) control four of the five indicator lamps along the left edge of the Hexadecimal Display Panel. The fifth indicator lamp is controlled by logic internal to the Hexadecimal Display Panel. To the right of each of these five lamps is a diagram that defines what is being displayed. In general, only one of the diagram lamps is on at a time. If none of the diagram lamps are on, a user program has written data to the display registers.

The most significant 20-bits of the display show the contents of Display Registers D3 and D4 and the least significant four bits of Display Register D5 (Bits 4:7) or the contents of the 20-bit Switch Register. When the Switch Register is being displayed, the lamp next to the Switch Register diagram is illuminated. Any other diagram lamp that may have been on, remains on. When the Switch Register is no longer displayed, its diagrma lamp goes out and the most significant 20-bits of the display again shows the contents of Display Registers D3 and D4 and the least significant four bits of Display Register D4 (Bits 4:7).

The Key Operated Security Lock is a three-position, OFF-ON-LOCK, key-operated locking switch, which controls the primary power to the system. This switch can also disable the Hexadecimal Display Panel, thereby preventing any accidentical manual input to the system. The power indicator lamp (PWR) associated with the key lock is located in the lower right corner of the Hexadecimal Display Panel. The PWR lamp is on when the key lock is in the ON or LOCK position. The relationship between the key lock switch positions, primary power, the Control keys, and the Hexadecimal keys is:

OFF	The primary power is OFF.
ON	The primary power is ON and the Control keys and Hexadecimal keys are enabled.
LOCK	The primary power is ON and the Control keys and Hexadecimal keys are disabled.

The Hexadecimal Display Panel operating procedures may be found in the appropriate User's Manual.

The Display Controller, built into the Processor, is shown on Sheet 26. Unlike most I/O controllers, data transfer does not take place over the D Bus. Data from the Hexadecimal Display Panel is gated directly to the B Bus, B08:14, and the content of the S Bus, S08:15, is gated by DAG1 and sent to the Hexadecimal Display Panel. Data is transmitted between the Hexadecimal Display Panel and the Display Controller one byte at a time.

5.13.1 Data Transfer. When the display is in the Normal mode, all data outputs are directed into Display Register D1. Conditioning the controller to the Incremental mode, via an Output command, causes the two bit counter (26H7) to be incremented at the trailing edge of DAG1. The output of this counter is decoded to activate LA0, in response to the first DAG1 and then LB0 for all subsequent DAG1's until the counter is initialized. In this mode, the first DA loads Display Register D1, the next DA loads Display Register D2. The next two DAs load Display Registers D3 and D4. This counter is initialized by SCLR0, by an Output command placing the controller in the Incremental mode, or whenever the display is addressed and the Normal mode is selected.

Input data from the Switch Register on the Hexadecimal Display Panel is handled in a similar manner as output data. In the Normal mode or on the first Data Request (DR), if in the Incremental mode, Switch Register Bits 12:19 are read. The second DR, in the Incremental mode, reads Switch Register Bits 4:11. The two bit counter (26H5) directs the DR to the appropriate group of Switch Registers. This counter is initialized by the same function as the four bit counter discussed above and is incremented at the trailing edge of DRG1.

NOTE

Bits 0:3 are gated out as part of the status byte when address is read.

5.13.2 Control Logic. When the display requires micro-program support, it generates two outputs, ESNO0 and ESNC0, which are latched in the RS flip-flop at 26C2. The output of this flip-flop sets the Console Attention flip-flop (CATN) at 26F1. This flip-flop is reset by GADR0 when the Processor addresses the display.

When the SGL function switch is depressed, SSGL1 becomes active (26A2) and ESNC0 and ESNO0 are generated which caused the Single flip-flop (26G3) to become set. This flip-flop remains set until another execute is generated and the SGL function is not selected.

5.13.3 Status Input. The status byte encoding is shown in Table 11. The status byte is gated onto the SD00:07 lines by the SRG0 lead. SRG0 gates the SD00:07 lines onto Bits 08:15 of the B Bus.

TABLE 11. DISPLAY STATUS AND COMMAND ENCODING

STATUS

	0	1	2	3	4	5	6	7	
Run	X	0	0	0	X	X	X	X	
Memory Write	X	0	0	1	X	X	X	X	
Memory Read	X	0	1	0	X	X	X	X	
Address	X	0	1	1	X	X	X	X	
Fixed Register	X	1	0	0	X	X	X	X	Single or Halt
Floating Register	X	1	0	1	X	X	X	X	Single or Halt
Function	X	1	0	0	X	X	X	X	Single or Halt

General Register									Floating Register
	0	X	X	X	1	0	0	0	0
	1	X	X	X	1	0	0	0	
	2	X	X	X	1	0	0	1	2
	3	X	X	X	1	0	0	1	
	4	X	X	X	1	0	1	0	4
	5	X	X	X	1	0	1	0	
	6	X	X	X	1	0	1	1	6
	7	X	X	X	1	0	1	1	
	8	X	X	X	1	1	0	0	8
	9	X	X	X	1	1	0	0	
	A	X	X	X	1	1	0	1	A
	B	X	X	X	1	1	0	1	
	C	X	X	X	1	1	1	0	C
	D	X	X	X	1	1	1	0	
	E	X	X	X	1	1	1	1	Floating Register E
General Register	F	X	X	X	1	1	1	1	

Function									
	0	X	X	X	0	0	0	0	Console Interrupt
	1	X	X	X	0	0	0	0	Register Set Select
	2	X	X	X	0	0	0	1	
	3	X	X	X	0	0	0	1	
	4	X	X	X	0	0	1	0	PSW
	5	X	X	X	0	0	1	0	LOC
	6	X	X	X	0	0	1	1	
	7	X	X	X	0	0	1	1	
	8	X	X	X	0	1	0	0	
	9	X	X	X	0	1	0	0	
	A	X	X	X	0	1	0	1	
	B	X	X	X	0	1	0	1	
	C	X	X	X	0	1	1	0	
	D	X	X	X	0	1	1	0	
	E	X	X	X	0	1	1	1	
Function	F	X	X	X	0	1	1	1	

COMMAND

Normal	1	0	0	0	0	0	0	0
Incremental	0	1	0	0	0	0	0	0

6.4 Fast Memory Timing Adjustment for 35-522M01, CPU A

The fast memory timing adjustment is set up at the factory and should not require field adjustments. Potentiometer R2 adjusts the start time of INH0. Refer to Figure 20. Grounding REQ0 at the Processor back panel Slot 7, Pin 139-0 of the CPU-A board causes constant memory cycles to occur. Before grounding REQ0, remove MAC or extended DMA buffer if installed. ER0 may be found at Slot 7, Pin 204-0 of CPU-A. INH0 may be found at Slot 7, Pin 104-0 of CPU-A. Adjust R26 for 280NS width on INH0.

CAUTION

WHEN SETTING UP FAST MEMORY TIMING, UNPLUG ALL MEMORY STACKS UNTIL THE MEMORY TIMING IS ADJUSTED. FAILURE TO DO SO MAY CAUSE DAMAGE TO THE MEMORIES.

WITH ALL MEMORY STACKS UNPLUGGED, THE PROCESSOR MAY NOT POWER-UP. THIS IS BECAUSE THE -16.5VDC CROWBAR IN THE POWER SUPPLY HAS TRIPPED, REMOVING THE -16.5VDC. LOWER THE -16.5VDC TO CORRECT THIS. BE SURE TO READJUST THE -16.5VDC AFTER PLUGGING THE MEMORY MODULES BACK IN.

6.5 DMA Clock Adjustment for the 35-522M00, CPU A

The DMA clock (TB1) adjustment is set at the factory and should not require any further adjustment. The DMA clock (TB1) determines, when EN0 is active, when to turn EN0 OFF. To properly adjust TB1 for the 7/16 HSALU, the adjustment should be made while observing the time difference between EN0 and ER0. The adjustment must be made such that the time between the rising edge of EN0 and the falling edge of ER0 is 90 nanoseconds. Adjusting TB1 on the 7/32 Processor is done by observing FPSEL0 and ER0 on the back panel and adjusting TB1 such that the rising edge of FPSEL0 occurs 115 nanoseconds before the falling edge of ER0. The variable resistor R56 adjusts the TB1 timing.

6.6 DMA Clock Adjustment for 35-522M01, CPU A

The DMA clock (TB1) adjustment is set at the factory and should not require any further adjustment. The DMA clock (TB1) determines, when EN0 is active, when to turn EN0 OFF. To properly adjust TB1 for the 7/16 HSALU, the adjustment should be made while observing the time difference between EN0 and ER0. The adjustment must be made such that the time between the rising edge of EN0 and the falling edge of ER0 is 90 nanoseconds. Adjusting TB1 on the 7/32 Processor is done by observing FPSEL0 and ER0 on the back panel and adjusting TB1 such that the rising edge of FPSEL0 occurs 115 nanoseconds before the falling edge of ER0. The variable resistor R1 adjusts the TB1 timing.

6.7 Overall Processor Test

Use the 06-106 Processor Test Program to perform a comprehensive test of the 7/16 HSALU Processor.

Use the 06-154 Series 32 Processor Test Part 1, 06-155 Series 32 Processor Test Part 2, 06-153 Model 7/32 Halfword Processor Test and the 06-156 F01, F02, and F03 Series 32 Memory Test.

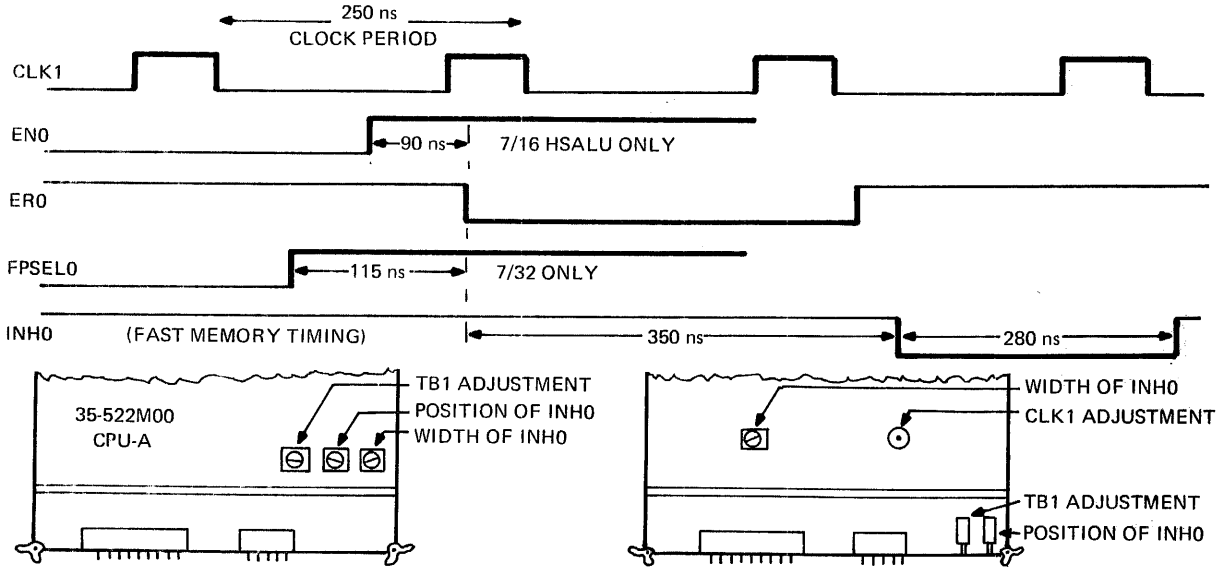


Figure 21. Clock Timing

7. Mnemonics List

The following list provides a brief description of each mnemonic found in the 7/16 HSALU and 7/32 Processor. The source of each signal on Schematic Drawing 01-079D08, is also provided.

MNEMONICS	MEANING	SCHEMATIC LOCATION
ACK0	Acknowledge Control Line. This signal starts the Rack0/Tack0 Daisy Chain.	25D9
ADD0	Add micro-instruction Decoded	14D9
ADRS0	Address Control line	25A9
AMOD0	Address modification	29R3
ARST1	Automatic Restart	31K6
ATN0	Attention Test line from the I/O Bus	31G1
ATNX1	I/O attention but no higher peiority interrupts	33B9
BCLK0	Basic Clock. A non-stoppable clock used for memory timing and test aid control.	11J1
B000:150	The B Bus which transmits data from the specified source to the shifter.	12A9-12K9
BRCH1	Branch. This indicated a Branch micro-instruction	14F9
BRCMND0	Branch or Command micro-instruction decoded	30F4
CA0	Calculate address Micro-instruction decode	14E9
CATN1	Console Attention. Special Interrupt from the Display Panel	31G9
CC12:151	Condition Code Bits 12 through 15; Carry, Overflow, Greater than, and Less than	30G9-30N9
CDIV0	Command Divide micro-instruction decoded	34R9
CEMT1	Counter empty	34E9
CIN21	Carry into ALU Bits 8 through 11	22B9
CIN31	Carry into ALU Bits 12 through 15	22A4
CISHI	Carry into Shift Register High	36K9
CISLI	Carry into Shift Register Low	36M9
CISL0	Carry in Shift Left to 'B' Bus Latch	36N5
CISR0	Carry in Shift Right to 'B' Bus Latch	36N5

MNEMONICS	MEANING	SCHEMATIC LOCATION
CKMDH0	Clock Memory Data High	18B4
CKMDL0	Clock Memory Data Low	19B4
CKRD0	The Clock used to Load the ROM Data Register and increment the ROM address	7F8
CLABT0	Clear the Abort flip-flop	33D9
CLK1	The major Processor Clock from which all other clocks are derived	32L3
CLMDR0	Clear Memory Data Register	19B-27F9
CLO70	Control Line 7. This function provides an Early Power Fail indication to devices on the I/O Bus.	25A9
CLRF0	Clear Alarm Register flip-flops	28G7
CLSRH0	Clear Shift Register High ANDed with Processor Clocks	36M3
CMDTS0	Command Test and Set to Memory Access Controller	34H9
CMD0	Command Control line	25C9
CMND0	Command micro-instruction Decoded	14E9
CMPY0	Command Multiply Micro-instruction Decoded	34R9
CPLOC1	Carry propagate for the Location Counter	17A1
CPMAR1	Carry propagate for the Memory Address Register	16D1
CRYIN0	Carry into the least significant bit of the ALU	33M9
CS0	Cross Shift. Conditions the B Bus shifter to perform a Cross Shift.	30R3
CSL0	Command Shift Left micro-instruction	34R9
CSRH0	Clear Shift Register High	36C7
CSR0	Command Shift Right micro-instruction	34S9
CSVDIV0	Divide Carry Save	36F7
CSVI	Carry Save. Carry out from the ALU	22C8
CTONE0	Counter equal to One (1).	34L9
DA0	Data Availabel Control line	25C9
DACK0	Data Channel Acknowledge interrupt	25B9
DC0	Data Channel interrupt	31J1
DCR0	Data Channel Read operation	31J1
DEC10	Decode ROM #1. Decoded from micro-instruction.	35G1

MNEMONICS	MEANING	SCHEMATIC LOCATION
DEC20	Decode ROM #2. Decoded from micro-instruction	35F9
DECTR0	Decrement the Counter	34M9
DIR0	Second Time the same Instruction.	33A9
DMAC0	Disable MAC or DMABC Controller.	35C9
D000-D150	I/O Bus data lines.	Sheet 15
DON0	Timing signal derived from functions of the Processor Clock Counter.	32N3
DS20	Disable Source Two	8L9
DSA0	Disable Source A.	35M9
DSB1	Disable Source B.	14R9
DR0	Data Request Control line.	25B9
DSTOP0	Destination Stop. Prevents the loading of any destination register when active.	33G9
ENDA1	Enable Destination 'A'.	14G1
ENDB1	Enable Destination 'B'.	14J9
ENDD1	Enable Destination 'D'.	14K9
ENH1	Enable High. Enables the loading of the Memory Data Register Low.	16S8
ENMS1	Enable Memory Sense. During data time of memory read cycle.	27G9
ENSA0	Enable Source 'A'.	35S9
ENSB0	Enable Source 'B'.	14S9
EN0	Enable signal to the direct Memory Access Port.	27K9
EPF0	Early Power Failure detected.	32K4
ER0	Early Read, used for Read Memory Timing.	27F9
ESNO0	Execute switch normally open contact.	26A2
ESNC0	Execute switch normally closed contact.	26A1
EXBYS0	External or Extended Memory Busy.	27L1
EXDUA0	External or Extended Memory Data Unavailable.	27G1
FABORT1	Abort flip-flop, set during Abortable instruction.	34F9
FAEN0	Auxillary EN0 flip-flop	27J9
FA1	Output from the Processor Timing flip-flop, State A.	32H1

MNEMONICS	MEANING	SCHEMATIC LOCATION
FAIFBO0	Timing signal derived from Processor Clock flip-flop	32J3
FAMOD0	Address Modified flip-flop.	29R4
FB1	Output from the Processor Timing flip-flop, State B.	32G1
FCATN1	Output from the Console Attention flip-flop indicating a request for Console Service.	26H1
FDAT0	I/O Data Output flip-flop.	25G9
FDECI0	Decode Type #1 flip-flop.	35B9
FDEC20	Decode Type #2 flip-flop.	35C9
FEPF0	Output from the Early Power Fail relay.	32J5
FHLD0	Hold flip-flop for slow memory timing.	27C6
FINR0	Output from the Instruction Read flip-flop.	35K9
FLRAR0	Load ROM Address register flip-flop.	8G4
FLBY0	Local Memory Busy flip-flop.	27L9
FLGL0	Flag Register Greater than or Less than set.	30N9
FLPBY0	Local Processor Memory Busy flip-flop.	27K9
FL121-151	Flag Register Bits 12 through 15.	30H7-30N7
FMBY0	Memory Busy flip-flop.	36B7
FMFF1	Multiply Operation flip-flop.	36K8
FPF1	Power Fail Detection flip-flop.	32J6
FPOW1	The one output from the Power Down flip-flop. Set by the micro-program to Initialize the system.	34D9
FRSUB0	Force Subtract on multiply.	36M3
FSKIP0	Output from the Skip flip-flop.	33F5
FSNGL1	Single flip-flop from Display Controller.	26H3
FSYN1	Output from the Sync flip-flop.	25J9
FTIT0	Clock stop from test aid.	33J1
FUT1	Utility flip-flop.	34B9
FWAIT0	Active when the Processor is in the Wait state.	34C9
FXPBY0	Exetended Processor Memory Busy flip-flop.	27N7

MNEMONICS	MEANING	SCHEMATIC LOCATION
GABB1	'A' or 'B', but not both sources negative	29B4
GABORT1	Gated Abort.	33C9
GATN1	Gated Attention.	31G9
GB00-150	Gated B Bus. The output from the Shifter/Latch circuit to the ALU.	20A9-20S9
GFLR121	Gated Flag Register Bit 12.	36F3
GLOAD1	Load Micro-operation and RD151 Decoded.	18B9
GMDR000	Gated Memory Data Register Bit 0.	24N9
GPSEL1	Gated Processor selected.	10H4
GRX130	Gated RX1 or RX3 format user instruction decoded.	8N9
HALT0	Signal from MAC preventing the setting of the processor Selected flip-flop.	27J1
HW0	Halfword I/O Test line from the active device.	33K1
ICLK1	Buffered Processor Clock used primarily in I/O timing.	25K1
ILOC1	Increment Location Counter.	35E9
IMAR1	Increment Memory Address Register	35D9
IMCMD0	Immediate or Command micro-instruction decoded.	14F9
INIT1	Output from the system Initialize switch.	33C9
INH0	Inhibit, used for write memory timing.	27C9
IR001-071	Instruction Register Bits 0 through 7. Operation Code portion of a user instruction.	A9-8F9
IR031-111	The outputs from the YD field of the Instruction Register.	29N5-29N7
IR1	Instruction Read decoded from a micro-instruction.	35H2
IRG1	Second Clock of Instruction Read. Use to load Memory Data Register to the YD portion of the Instruction Register.	35J9
IR121-151	The outputs from the YS portion of the Instruction Register.	29L1-29L4
ISTOP0	I/O Processor Clock Stop. Active during I/O Transfer, keeps ROM Data from Changing.	25R9
JALARM	Prepares the Alarm Register to jam to the B-Bus with its contents when a Load PSW is specified.	34E9
JARLB1	Force ARL to be the second source during Multiply.	24B9

MNEMONICS	MEANING	SCHEMATIC LOCATION
LA0	Load display Byte A, to Display 2 Bits 8:15.	26J7
LAR1	Load Arithmetic Register Control line.	14M9
LARH120:150	Data Inputs to the AR High	10E3
LARH0	Load Arithmetic Register High.	24C9
LARL0	Load Arithmetic Register Low.	24D9
LARX0	Load the extended portion of the Arithmetic Register.	24D9
LB0	Load display byte B, to Display 2 Bits 0:7.	26J7
LCNTR1	Load the Counter.	35M9
LD100	Load I/O micro-instruction decoder.	30S4
LIO1	I/O operation in progress.	15G9
LFLR0	Load the Flag Register.	35K9
LCC0	Load the Condition Code Register.	34G9
LINH0	Fast Memory Timing INH0.	27N9
LLOC0	Load the Location Counter.	14L9
LMAR0	Load the Memory Address Register from the S Bus.	14L9
LMAS0	Load the Memory Address Slave Register.	27E9
LMDR1	Load the Memory Data Register from the S Bus.	14L9
LOAD0	Any Load micro-instruction decoded.	14B9
LPSWL1	Load Program Status Word Low Order Bits 26 through 31.	31F9
LRAR0	Load ROM Address Register. Enables the loading of the RAR.	33E9
LR0	Late Read, used for read memory timing.	27E9
LSRH1	Load Shift Register High	35L9
LSRL1	Load Shift Register Low	35L9
LYSI0	Lead YS portion of the Instruction Register from the S Bus.	14L9
MA000:140	Output from the Memory Address Register Bits 0:15.	Sheet 16
MAR001:151	Output from the Memory Address Register Bits 0:15.	Sheet 16
MCI0	Multiply Carry into the Shift Register.	36G8
MD0	Multiply or divide operation decoded.	36L3
MD000:150	Memory Data Bus to the memory system.	Sheets 18-19
MDR000:070	Outputs from the Memory Data Register.	Sheets 18-19

MNEMONICS	MEANING	SCHEMATIC LOCATION
MDS20	Multiply operation, disable the second source.	24R9
MEGRL00	Memory enable General Register Low, Set 0.	11D8
MEGRLF0	Memory enable General Register Low, Set F.	11N8
MEGRH00	Memory enable General Register High, Set 0.	11H8
MEGRHF0	Memory enable General Register High, Set F.	11R8
MEM1	Memory Operation decoded from micro-instruction.	35E9
MEMS0	Memory enable for the micro-register stacks.	11B8
MEXR00	Memory enable for the Extended Registers Set 0.	11M8
MEXRF0	Memory enable for the Extended Register Set F.	11R8
M1	The M input to the ALU decoded.	14J9
MMAL1	Machine Malfunction-Parity Error or Power Failure	28L6
MW1	Memory Write Decode from Micro-instruction.	35A9
NORM0	Floating point number not normalized.	23F9
OSC1	Output from the System Clock Oscillator.	32C1
OVAS1	Overflow enable function on an add or subtract.	29G6
OVA1	Overflow enable function on an add.	29G7
OVS1	Overflow enable function on a subtract.	29G5
PO0	Phase 0 of Calculate Address micro-instruction.	8G9
P11	Phase 1 of a Calculate Address micro-instruction.	8N9
P20	Phase 2 of a Calculate Address micro-instruction.	8M9
PBY1	Local Processor memory Busy or Extended Processor Memory Busy	29B6
PERR0	Parity error on Memory Data Bus.	28B6
PFDT0	Power Failure detected.	32D9
POFF0	Output from the On/Off Power switch controlling system power.	32B7
POWDN0	Power is down. Resets the SCLR relay.	32J7
PRIV0	Privileged instruction decoded from DROM.	34G9
PRTECT0	Memory Protected and Processor is doing memory write.	35A9
PSEL1	Processor selected for Memory Bus.	18B3

MNEMONICS	MEANING	SCHEMATIC LOCATION
PSW111	Program Status Word Bit 11.	31C4
PSW161-231	Program Status Word Bit 16 through 23.	31A8-31E8
PSW271	Program Status Word Bit 27.	31D4
RAR050:150	The outputs from the RAR.	Sheet 5
RARSTP0	ROM Address Register clock stop.	8F9
RD000:230	ROM Data Register Bit through 23.	Sheet 7
READ1	Register stack read time.	11N1
REQ0	Request from the Direct Memory Access Port.	27K1
RR1	Register to Register format user instruction decoder.	9N3
RSTOP0	ROM Stop. Stops CKRD0 when active.	33F9
SA1	General Register stacks addressing signal.	12A4
SB1	General Register stacks addressing signal.	12E1
SC1	General Register stacks addressing signal.	12E1
SCHRY1	Shifted Carry. Enables the setting of the carry flag on either a shift right or shift left.	29D4
SCLR0	System Clear. Signal used to initialize the system on a power up or power down.	32A5
SD1	General Register stacks addressing signal.	12H1
SD001:071	Status and Data Bus to the Display Console.	26R1-26R7
SDSTOP0	Set Destination Stop Clock during Repeat Mode Micro-instruction.	34M9
SELA1	Source address line for A Bus.	24J9
SELB1	Source address line for A Bus.	24G9
SEQ01	S Bus equals zero.	22N8
SGN10	Second source equal to one.	24L9
SGN20	Second source equal to two.	24K9
SGNMDR0	Sign bit of Memory Data Register.	24K9
SHCLK0	32 bit Shift Register Clock	36A9
SHL1	Shift Left. Conditions the B Bus Shifter to shift data left one place.	16N9
SHR1	Shift Right. Conditions the B Bus Shifter to shift data right one place.	16N9

MNEMONICS	MEANING	SCHEMATIC LOCATION
SHORT1	32 Bit RX instruction being executed	8R9
SHI0	Input high data switches	26J6
SLO0	Input low data switches	26J6
SMALF1	A Machine Malfunction is pending.	31H9
SMPY0	Set multiply flip-flop	36A1
SNGL1	Single. Test Point to the micro-program indicating status of the Control Console.	31G9
S01:S31	S01 The S0 input to the ALU decoded. S11 The S1 input to the ALU decoded. S20 The S2 input to the ALU decoded. S31 The S3 input to the ALU decoded.	14G9:14
S001:S151	S Bus. Outputs from the ALU.	Sheets 21-22
SRA050:150	The Set ROM Address lines to the RAR.	5F4:5F9
SRD000:230	The outputs from the ROM used to load the ROM Data Register.	6S2-6S9
SRH091:111	Shift Register High bits 9 through 11.	23D9
SRHS01	Shift Register High Control lines.	36E7
SRHS11	Shift Register High Control line.	36E7
SRLCI1	Shift Register Low Carry In.	36L9
SRLS01	Shift Register Low Control line.	36C7
SRLS11	Shift Register Low Control line.	36D7
SR0	Status Request Control line.	25D6
SRSTOP0	Set ROM Clock Stop.	34L9
SSGL1	Singl Switch from Display Panel	26A2
SUB1	Subtract micro-instruction decoded.	14H9
SV0	Set Overflow flag. Direct sets the Overflow flag when False Sync is detected.	25L1
SYN0	Sync response from external device controller.	26J4
TBRCH1	True Branch	31S8
ULOCH1	Unload Location Counter high	14N9
ULOC1	Unload Location Counter.	35R9
UMAR1	Unload the Memory Address Register to the B Bus.	35M9

MNEMONICS	MEANING	SCHEMATIC LOCATION
UMDR0	Unload the Memory Data Register to the B Bus.	24F9
UMDR1	Unload the Memor Data Regist to the B Bus.	35M9
USRH1	Unload the Shift Register High.	23M1
USRLOC1	Unload Shift Register Low.	22M2
USRLOC1	Unload Shift Register or Location Counter.	14N9
WAIT	Controls the state of the Wait indicator on the Control Console.	26J3
WRT0	Memory Bus signal indicating that a Processor memory write is taking place.	27H9
WO	Memory Timing signal defining write time.	27B9
XB00:30	Extended B Bus 0 through 3.	9N9
XMA120:150	Extended Memory Address Register Bits 12 through 15	10K5
XMDR00:30	Extended Memory Data Register Bits 0 through 3.	9N9
XMEM0	Extended Memory operation taking place. Comes from MAC or DMABC.	9N9
XS01:31	Extended S Bus Bits 0 through 3.	10N9
YDM1	YD minus one. Decrement YD.	29G8
YDP1	YD plue one. Increment YD.	29G7
YSH1	General Register source address line.	12L1
YLS1	General Register stack source address line.	12H1
YA121:YS151	User Source Bus for loading YS field of Instruction Register and for loading the micro-register stack.	29K1-29K5

Appendix 1.
Multiply-Divide Examples

This appendix to 01-079A21 shows examples of both Multiply and Divide operations indicating the state of the various busses and registers during each clock of the operation. These are examples of the operands used in the test programs.

TABLE A1-1. SIGNED MULTIPLY (MPY) EXAMPLE WITH THE FOLLOWING OPERANDS:

Multiplier = X'FFFF'
Multiplicand = X'1111'

Prior to execution of the Multiply micro-instruction the following registers have been loaded by the micro-program as follows:

MDR = X'1111'
ARL = X'2222'
CARRY Flag = 0

		SRX 14	SRX 15		SRH				SRL				SRL 16
	1	0	0	-	0	0	0	0	F	F	F	F	0
	2			B	0	0	0	0					
				GB	0	0	0	0					
		1	1	S	E	E	E	F	F	F	F	F	1
				SRH	F	7	7	7	F	F	F	F	1
	3			B	F	7	7	7					
				GB	F	B	B	B					
		1	1	S	F	B	B	B	F	F	F	F	1
		1	1	SRH	F	D	D	D	F	F	F	F	1
	4			B	F	D	D	D					
				GB	F	E	E	E					
		1	1	S	F	E	E	E	7	F	F	F	1
		1	1	SRH	F	F	7	7	B	F	F	F	1
	5			B	F	F	7	7					
				GB	F	F	B	B					
		1	1	S	F	F	B	B	D	F	F	F	1
		1	1	SRH	F	F	D	D	E	F	F	F	1
	6			B	F	F	D	D					
				GB	F	F	E	E					
		1	1	S	F	F	E	E	7	7	F	F	1
		1	1	SRH	F	F	F	7	B	B	F	F	1
	7			B	F	F	F	7					
				GB	F	F	F	B					
		1	1	S	F	F	F	B	D	D	F	F	1
		1	1	SRH	F	F	F	D	E	E	F	F	1
	8			B	F	F	F	D					
				GB	F	F	F	E					
		1	1	S	F	F	F	E	7	7	7	F	1
		1	1	SRH	F	F	F	F	B	B	B	F	1
	9			B	F	F	F	F					
				GB	F	F	F	F					
		1	1	S	F	F	F	F	D	D	D	F	1
		1	1	SRH	F	F	F	F	E	E	E	F	1
L SRH, SRH, SR	10			B	F	F	F	F	E	E	E	F	
				GB	F	F	F	F					
				S	F	F	F	F					
RESULT				SRH	F	F	F	F					

TABLE A1-2. SIGNED MULTIPLY (MPY) EXAMPLE WITH THE FOLLOWING OPERANDS:

Multiplier = X'1111'
Multiplicand = X'FFFF'

Prior to execution of the Multiply micro-instruction the following registers have been loaded by the micro-program as follows:

MDR = X'FFFF'
ARL = X'FFFE'
CARRY Flag = 1

		SRX 14	SRX 15		SRH				SRL				SRL 16
	1	0	0	-	0	0	0	0	1	1	1	1	0
				B	0	0	0	0					
				GB	0	0	0	0					
				S	F	F	F	F	8	8	8	8	1
	2	1	1	SRH	F	F	F	F	C	4	4	4	0
				B	F	F	F	F					
				GB	F	F	F	F					
		1	1	S	F	F	F	F	E	2	2	2	0
	3	1	1	SRH	F	F	F	F	F	1	1	1	0
				B	F	F	F	F					
				GB	F	F	F	F					
	4	1	1	S	F	F	F	E	7	8	8	8	1
		1	1	SRH	F	F	F	F	B	C	4	4	0
				B	F	F	F	F					
	5			GB	F	F	F	F					
		1	1	S	F	F	F	F	D	E	2	2	0
		1	1	SRH	F	F	F	F	E	F	1	1	0
				B	F	F	F	F					
				GB	F	F	F	F					
	6	1	1	S	F	F	F	E	7	7	8	8	1
		1	1	SRH	F	F	F	F	B	B	C	4	0
				B	F	F	F	F					
				GB	F	F	F	F					
	7	1	1	S	F	F	F	F	D	D	E	2	0
		1	1	SRH	F	F	F	F	E	E	F	1	0
				B	F	F	F	F					
				GB	F	F	F	F					
	8	1	1	S	F	F	F	E	7	7	7	8	1
		1	1	SRH	F	F	F	F	B	B	B	C	0
				B	F	F	F	F					
				GB	F	F	F	F					
	9	1	1	S	F	F	F	F	D	D	D	E	0
		1	1	SRH	F	F	F	F	E	E	E	F	0
I, SRH, SRH, SR	10			B	F	F	F	F	E	E	E	F	-
				GB	F	F	F	F					
				S	F	F	F	F					
RESULT				SRH	F	F	F	F					

TABLE A1-3. UNSIGNED MULTIPLY (UMPY) EXAMPLE WITH THE FOLLOWING OPERANDS:

Multiplier = X'FFFF'
Multiplicand = X'1111'

Prior to execution of the Multiply micro-instruction the following registers have been loaded by the micro-program as follows:

MDR = X'1111'
ARL = X'2222'
CARRY Flag = 0

		SRX 14	SRX 15		SRH				SRL				SRL 16
	1	0	0	-	0	0	0	0	F	F	F	F	0
				B	0	0	0	0					
				GB	0	0	0	0					
				S	E	E	E	F	F	F	F	F	1
	2	1	1	SRH	F	7	7	7	F	F	F	F	1
				B	F	7	7	7					
	3			GB	F	B	B	B					
				S	F	B	B	B	F	F	F	F	1
		1	1	SRH	F	D	D	D	F	F	F	F	1
				B	F	D	D	D					
				GB	F	E	E	E					
	4			S	F	E	E	E	7	F	F	F	1
		1	1	SRH	F	F	7	7	B	F	F	F	1
				B	F	F	7	7					
				GB	F	F	B	B					
	5			S	F	F	B	B	D	F	F	F	1
		1	1	SRH	F	F	D	D	E	F	F	F	1
				B	F	F	D	D					
				GB	F	F	E	E					
	6			S	F	F	E	E	7	7	F	F	1
		1	1	SRH	F	F	F	7	B	B	F	F	1
				B	F	F	F	7					
				GB	F	F	F	B					
	7			S	F	F	F	B	B	B	F	F	1
		1	1	SRH	F	F	F	D	D	D	F	F	1
				B	F	F	F	D					
				GB	F	F	F	E					
	8			S	F	F	F	E	E	E	F	F	
		1	1	SRH	F	F	F	F	B	B	B	F	
				B	F	F	F	F					
				GB	F	F	F	F					
	9			S	F	F	F	F	D	D	D	F	1
		1	1	SRH	F	F	F	F	E	E	E	F	
A SRH, SRH, MOR, SR	10			B	F	F	F	F	E	E	E	F	
				GB	F	F	F	F					
				S	1	1	1	0					
RESULT				SRH	1	1	1	0					

TABLE A1-4. SIGNED MULTIPLY (MPY) EXAMPLE WITH THE FOLLOWING OPERANDS:

Multiplier = X'7FFF'
Multiplicand = X'FFFF'

Prior to execution of the Multiply micro-instruction the following registers have been loaded by the micro-program as follows:

MDR = X'FFFF'
ARL = X'FFFE'
CARRY Flag = 1

		SRH 14	SRH 15		SRH				SRL				SRL 16
	1	0	0	0	0	0	0	0	7	F	F	F	0
	2			B	0	0	0	0					
				GB	0	0	0	0					1
				S	0	0	0	1	B	F	F	F	1
		1	1	SRH	8	0	0	0	5	F	F	F	1
	3			B	8	0	0	0					
				GB	C	0	0	0					
				S	C	0	0	0	2	F	F	F	1
		1	1	SRH	E	0	0	0	1	7	F	F	1
				B	E	0	0	0					
	4			GB	F	0	0	0					
				S	F	0	0	0	0	B	F	F	1
		1	1	SRH	F	8	0	0	0	5	F	F	1
				B	F	8	0	0					
				GB	F	C	0	0					
	5			S	F	C	0	0	0	2	F	F	1
		1	1	SRH	F	E	0	0	0	1	7	F	1
				B	F	E	0	0					
				GB	F	F	0	0					
	6			S	F	F	0	0	0	0	B	F	1
		1	1	SRH	F	F	8	0	0	0	5	F	1
				B	F	F	8	0					
				GB	F	F	C	0					
	7			S	F	F	C	0	0	0	2	F	1
		1	1	SRH	F	F	E	0	0	0	1	7	1
				B	F	F	E	0					
				GB	F	F	F	0					
	8			S	F	F	F	0	0	0	0	B	1
		1	1	SRH	F	F	F	8	0	0	0	5	1
				B	F	F	F	8					
				GB	F	F	F	C					
	9			S	F	F	F	C	0	0	0	2	1
		1	1	SRH	F	F	F	E	8	0	0	1	0
L SRH, SRH, SR	10			B	F	F	F	F	8	0	0	1	-
SIGN				GB	F	F	F	F					
				S	F	F	F	F					
RESULT				SRH	F	F	F	F					

TABLE A1-5. UNSIGNED MULTIPLY (UMPY) EXAMPLE WITH THE FOLLOWING OPERANDS:

Multiplier = X'FFFF'
Multiplicand = X'FFFF'

Prior to execution of the Multiply micro-instruction the following registers have been loaded by the micro-program as follows:

MDR = X'FFFF'
ARL = X'FFFE'
CARRY Flag = 1

		SRX 14	SRX 15		SRH				SRL				SRL 16
	1	0	0	-	0	0	0	0	F	F	F	F	0
				B	0	0	0	0					
				GB	0	0	0	0					
				S	0	0	0	1	F	F	F	F	1
	2	1	1	SRH	8	0	0	0	7	F	F	F	1
				B	8	0	0	0					
				GB	C	0	0	0					
				S	C	0	0	0	3	F	F	F	1
	3	1	1	SRH	E	0	0	0	1	F	F	F	1
				B	E	0	0	0					
				GB	F	0	0	0					
				S	F	0	0	0	0	F	F	F	1
	4	1	1	SRH	F	8	0	0	0	7	F	F	1
				B	F	8	0	0					
				GB	F	C	0	0					
				S	F	C	0	0	0	3	F	F	1
	5	1	1	SRH	F	E	0	0	0	1	F	F	1
				B	F	E	0	0					
				GB	F	F	0	0					
				S	F	F	0	0	0	0	F	F	1
	6	1	1	SRH	F	F	8	0	0	0	7	F	1
				B	F	F	8	0					
				GB	F	F	C	0					
				S	F	F	C	0	0	0	3	F	1
	7	1	1	SRH	F	F	E	0	0	0	1	F	1
				B	F	F	E	0					
				GB	F	F	F	0					
				S	F	F	F	0	0	0	0	F	1
	8	1	1	SRH	F	F	F	8	0	0	0	7	1
				B	F	F	F	8					
				GB	F	F	F	C					
				S	F	F	F	C	0	0	0	3	1
	9	1	1	SRH	F	F	F	E	0	0	0	1	1
A SRH, SRH, MDR, SR	10			B	F	F	F	E					
				GB	F	F	F	F					
				S	F	F	F	E	0	0	0	1	-
RESULT				SRH	F	F	F	E					

TABLE A1-6. SIGNED MULTIPLY (MPY) EXAMPLE WITH THE FOLLOWING OPERANDS:

Multiplier = X'8000'
Multiplicand = X'8000'

Prior to execution of the Multiply micro-instruction the following registers have been loaded by the micro-program as follows:

MDR = X'8000'
ARL = X'0000'
CARRY Flag = 1

		SRX 14	SRX 15		SRH				SRL				SRL 16
	1	0	0	-	0	0	0	0	8	0	0	0	0
				B	0	0	0	0					
				GB	0	0	0	0					
	2			S	0	0	0	0	4	0	0	0	0
		0	0	SRH	0	0	0	0	2	0	0	0	0
				B	0	0	0	0					
				GB	0	0	0	0					
	3	0	0	S	0	0	0	0	1	0	0	0	0
		0	0	SRH	0	0	0	0	0	8	0	0	0
				B	0	0	0	0					
				GB	0	0	0	0					
	4	0	0	S	0	0	0	0	0	4	0	0	0
		0	0	SRH	0	0	0	0	0	2	0	0	0
				B	0	0	0	0					
				GB	0	0	0	0					
	5	0	0	S	0	0	0	0	0	1	0	0	0
		0	0	SRH	0	0	0	0	0	0	8	0	0
				B	0	0	0	0					
				GB	0	0	0	0					
	6	0	0	S	0	0	0	0	0	0	4	0	0
		0	0	SRH	0	0	0	0	0	0	2	0	0
				B	0	0	0	0					
				GB	0	0	0	0					
	7	0	0	S	0	0	0	0	0	0	1	0	0
		0	0	SRH	0	0	0	0	0	0	0	8	0
				B	0	0	0	0					
				GB	0	0	0	0					
	8	0	0	S	0	0	0	0	0	0	0	4	0
		0	0	SRH	0	0	0	0	0	0	0	2	0
				B	0	0	0	0					
				GB	0	0	0	0					
	9	1	1	S	0	0	0	0	0	0	0	1	0
				SRH	8	0	0	0	0	0	0	0	1
L SRH,SRH,SR	10			B	4	0	0	0	0	0	0	0	0
				GB	4	0	0	0					
				S	4	0	0	0					
RESULT				SRH	4	0	0	0					

TABLE A1-7. SIGNED MULTIPLY (MPY) EXAMPLE WITH THE FOLLOWING OPERANDS:

Multiplier = X'FFFF'
Multiplicand = X'8000'

Prior to execution of the Multiply micro-instruction the following registers have been loaded by the micro-program as follows:

MDR = X'8000'
ARL = X'000'
CARRY Flag = 1

		SRX 14	SRX 15		SRH				SRL				SRL 16
	1	0	0	-	0	0	0	0	F	F	F	F	0
				B	0	0	0	0					
				GB	0	0	0	0					
				S	8	0	0	0	7	F	F	F	1
	2	0	0	SRH	4	0	0	0	3	F	F	F	1
				B	4	0	0	0					
				GB	2	0	0	0					
	3	0	0	S	2	0	0	0	1	F	F	F	1
		0	0	SRH	1	0	0	0	0	F	F	F	1
				B	1	0	0	0					
				GB	0	8	0	0					
	4	0	0	S	0	0	0	0	0	7	F	F	1
		0	0	SRH	0	4	0	0	0	3	F	F	1
				B	0	4	0	0					
				GB	0	2	0	0					
	5			S	0	2	0	0	0	1	F	F	1
		0	0	SRH	0	1	0	0	0	0	F	F	1
				B	0	1	0	0					
				GB	0	0	8	0					
	6	0	0	S	0	0	8	0	0	0	7	F	1
				SRH	0	0	4	0	0	0	3	F	1
				B	0	0	4	0					
				GB	0	0	2	0					
	7	0	0	S	0	0	2	0	0	0	1	F	1
		0	0	SRH	0	0	1	0	0	0	0	F	1
				B	0	0	1	0					
				GB	0	0	0	8					
	8	0	0	S	0	0	0	8	0	0	0	7	1
		0	0	SRH	0	0	0	4	0	0	0	3	1
				B	0	0	0	4					
				GB	0	0	0	2					
	9	0	0	S	0	0	0	2	0	0	0	1	1
		0	0	SRH	0	0	0	1	8	0	0	0	0
L SRH, SRH, SR	10			B	0	0	0	0	8	0	0	0	-
				GB	0	0	0	0					
				S	0	0	0	0					
RESULT				SRH	0	0	0	0					

TABLE A1-8. UNSIGNED MULTIPLY (UMPY) EXAMPLE WITH THE FOLLOWING OPERANDS:

Multiplier = X'8000'
Multiplicand = X'FFFF'

Prior to execution of the Multiply micro-instruction the following registers have been loaded by the micro-program as follows:

MDR = X'FFFF'
ARL = X'FFFE'
CARRY Flag = 1

A = FFFF
B = 8000
D = 7FFFF 8000

(UMPY) D = 4X3

	SRX 14	SRX 15		SRH				SRL				SRL 16
1	0	0	-	0	0	0	0	8	0	0	0	0
			B	0	0	0	0					
			GB	0	0	0	0					
			S	0	0	0	0	4	0	0	0	0
2	0	0	SRH	0	0	0	0	2	0	0	0	0
			B	0	0	0	0					
			GB	0	0	0	0					
			S	0	0	0	0	1	0	0	0	0
3	0	0	SRH	0	0	0	0	0	8	0	0	0
			B	0	0	0	0					
			GB	0	0	0	0					
4			S	0	0	0	0	0	4	0	0	0
	0	0	SRH	0	0	0	0	0	2	0	0	0
			B	0	0	0	0					
			GB	0	0	0	0					
5			S	0	0	0	0	0	1	0	0	0
	0	0	SRH	0	0	0	0	0	0	8	0	0
			B	0	0	0	0					
			GB	0	0	0	0					
6			S	0	0	0	0	0	0	4	0	0
	0	0	SRH	0	0	0	0	0	0	2	0	0
			B	0	0	0	0					
			GB	0	0	0	0					
7			S	0	0	0	0	0	0	1	0	
	0	0	SRH	0	0	0	0	0	0	0	8	0
			B	0	0	0	0					
			GB	0	0	0	0					
			S	0	0	0	0	0	0	0	4	0
8	0	0	SRH	0	0	0	0	0	0	0	2	0
			B	0	0	0	0					
			GB	0	0	0	0					
			S	0	0	0	2	0	0	0	1	0
9	1	1	SRH	0	0	0	1	8	0	0	0	1
10			B	0	0	0	1					
A SRH, SRH, MDR SR			GB	8	0	0	0					
			S	7	F	F	F	8	0	0	0	
RESULT			SRH	7	F	F	F					

TABLE A1-9. DIVIDE EXAMPLE WITH THE FOLLOWING OPERANDS:

Divisor = X'7FFF'
Dividend = X'3FFF 7FFF'

Prior to execution of the Divide micro-instruction the following registers have been loaded by the micro-program as follows:

SRH = X'3FFF'
SRL = X'7FFF'
ARL = X'8000'

CLOCK	CARRY SAVE FROM ALU			SRL
1	C=0	B GB S SRH	3FFF 7FFE FFFE 7FFE	FFFE
2	C=1	B GB S SRH	7FFE FFFD 7FFD 7FFD	FFFD
3	C=1	B GB S SRH	7FFD FFFB 7FFB 7FFB	FFFB
4	C=1	B GB S SRH	7FFB FFF7 7FF7 7FF7	FFF7
5	C=1	B GB S SRH	7FF7 FFEF 7FEF 7FEF	FFEF
6	C=1	B GB S SRH	7FEF FFDF 7FDF 7FDF	FFDF
7	C=1	B GB S SRH	7FDF FFBF 7FBF 7FBF	FFBF
8	C=1	B GB S SRH	7FBF FF7F 7F7F 7F7F	FF7F
9	C=1	B GB S SRH	7F7F FEFF 7EFF 7EFF	FEFF
10	C=1	B GB S SRH	7EFF FDFF 7DFF 7DFF	FDFF
11	C=1	B GB S SRH	7DFF FBFF 7BFF 7BFF	FBFF

CLOCK	CARRY SAVE FROM ALU			SRL
12	C=1	B GB S SRH	7BFF F7FF 77FF 77FF	F7FF
13	C=1	B GB S SRH	77FF EFFF 6FFF 6FFF	EFFF
14	C=1	B GB S SRH	6FFF DFFF 5FFF 5FFF	DFFF
15	C=1	B GB S SRH	5FFF BFFF 3FFF 3FFF	BFFF
16	C=0	B GB S SRH	3FFF 7FFF FFFF 7FFF	7FFE

Legend:

B	B-Bus
GB	Gated B Bus
S	S-Bus
SRH	Shift Register High
SRL	Shift Register Low
SRX	Extension to Shift Register High

SELECTOR CHANNEL

M70-103
NS SELECTOR CHANNEL
INSTALLATION SPECIFICATION

1. INTRODUCTION

This specification provides the necessary information for the installation of the 02-232 Selector Channel (SELCH) (Product Number M70-103) in a Model 70, 74, 80, 7/16 or 7/16 HSALU Processor System. The NS Selector Channel is complete on one 35-391M02 printed circuit board.

2. PHYSICAL CHARACTERISTICS

2.1 Dimensions 15 3/8 x 14 7/8"

2.2 Weight $2\frac{1}{2}$ pounds maximum

3. INSTALLATION

The NS SELCH may be installed in any even numbered universal expansion slot (i.e., 0, 2, 4, or 6) in the Central Processor Unit (CPU) or in the first memory-I/O expansion chassis. See Figure 1. On 7/16 HSALU the NS SELCH may only be installed in Slot 0 of the CPU back panel.

To install a NS Selector Channel on a Model 74 or a 7/16 BASIC, the Selector Channel must be a 35-391M02. To install a Selector Channel on a 7/16 HSALU the Selector Channel must be a 35-391M02, R02 or higher.

NOTE

A SELCH may be installed in slots 0, 1, or 2 of a Model 80/85 CPU chassis only. In this case cutting of the Multiplexor Bus is not necessary.

3.1 Back Panel Wiring

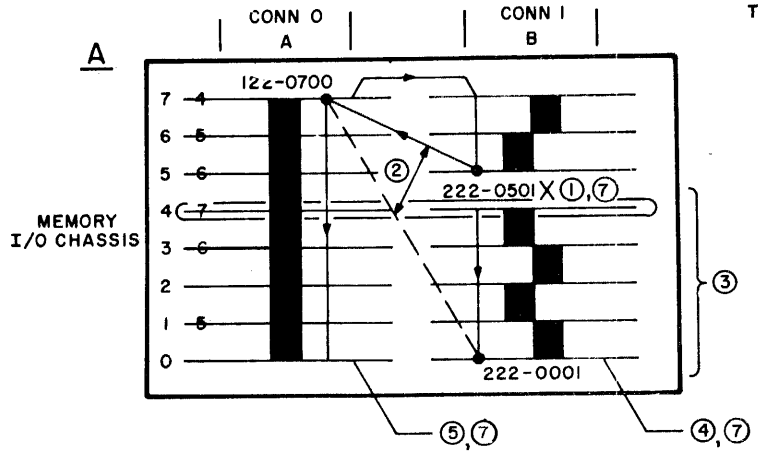
3.1.1 Multiplexor Channel Bus. **At the time of installation it is necessary to cut the Multiplexor Bus wiring between the even numbered slot accepting the SELCH and the next higher numbered slot on the One (1) connector only. The RACK0/TACK0 daisy chain wiring on the back panel is rerouted according to Figure 1. The lower numbered card slots in the chassis become part of the private SELCH Bus on the One (1) connector only.**

For the convenience of cutting the Multiplexor Bus, the connections between every other slot are made using "top" wire wraps. (This refers to wire wrap back panels only.) This allows the cutting of the bus by simply lifting the top wraps when the SELCH is installed in an even numbered slot. Refer to Figure 1 A during the following example.

To install a SELCH in Slot 4:

1. **Remove all wires on Connector One (1), between Slots 4 and 5, on Pins 11 through 26, Rows 1 and 2. (See backpanel map in 02-232M01D08 Sheet 7.)**
2. **Remove the wire between 222-0001 and 122-0700.**
3. **Remove the RACK0/TACK0 jumper between Pins 122 and 222 on both the Zero (0) and One (1) connectors of Slot 4.**
4. **Connect 122-0700 to 222-0501.**
5. **Install the SELCH into Slot 4 of the chassis. The private SELCH Bus now appears on the Connector One (1) side of Slots 4, 3, 2, 1, and 0. All slots on the Connector Zero (0) side and Slots 7, 6, and 5 on Connector One (1) side remain as standard Multiplexor Bus slots.**

To install a SELCH in any other even numbered slot of a CPU chassis or a Memory-I/O chassis, a similar procedure is followed. Refer to Figure 1 B, C, D, and E.



NOTE: THE CIRCLED NUMBERS ON ILLUSTRATIONS A, B, AND C REFER TO THE CORRESPONDING NUMBERS IN THE FOLLOWING INSTALLATION PROCEDURES.

TO INSTALL A SELECTOR CHANNEL IN SLOT 4 OF THE MEMORY I/O CHASSIS ——

① CUT THE MULTIPLEXOR BUS BY REMOVING THE TOP WRAPS.

② JUMPER RACKO/TACKO AS SHOWN, REMOVE DASHED JUMPER.

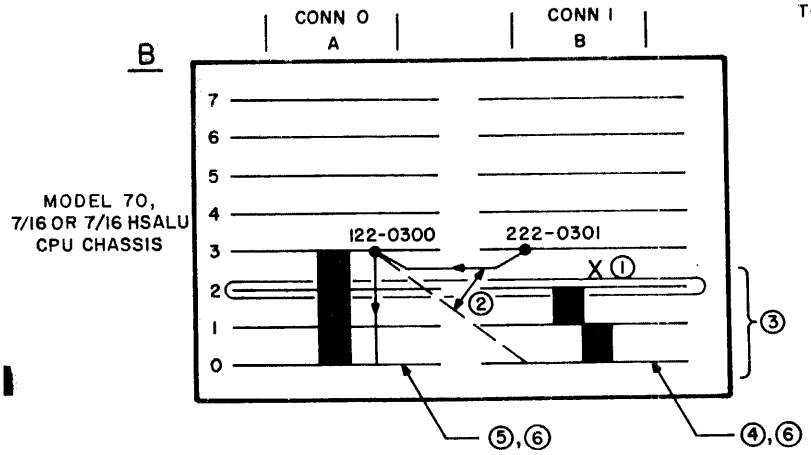
③ THIS SECTION BECOMES THE PRIVATE SELECTOR BUS ON THE CONNECTOR ONE (CONN.1) SIDE ONLY. ALL SLOTS ON THE CONNECTOR ZERO (CONN.0) SIDE, AND SLOTS 7, 6 AND 5 ON CONNECTOR ONE SIDE REMAIN AS STANDARD MULTIPLEXOR BUS SLOTS.

④ IF REQUIRED, EXTEND THE SELECTOR CHANNEL BUS TO OTHER CHASSIS BY INSTALLING A CABLE HERE.

⑤ EXTEND THE MULTIPLEXOR BUS TO OTHER CHASSIS BY INSTALLING A CABLE HERE.

⑥ MEMORY MODULE 7, IF IT EXISTS, MUST BE LOCATED IN SLOT 3.

⑦ INSTALL I/O TERMINATORS 35-433 ROI HERE.



TO INSTALL A SELECTOR CHANNEL IN SLOT 2 OF THE CPU CHASSIS ——

① CUT THE MULTIPLEXOR BUS BY REMOVING THE TOP WRAPS.

② JUMPER RACKO/TACKO AS SHOWN, REMOVE DASHED JUMPER.

③ THIS SECTION BECOMES THE PRIVATE SELECTOR BUS ON THE CONNECTOR ONE (CONN.1) SIDE ONLY. ALL SLOTS ON THE CONNECTOR ZERO (CONN.0) SIDE REMAIN AS STANDARD MULTIPLEXOR BUS SLOTS.

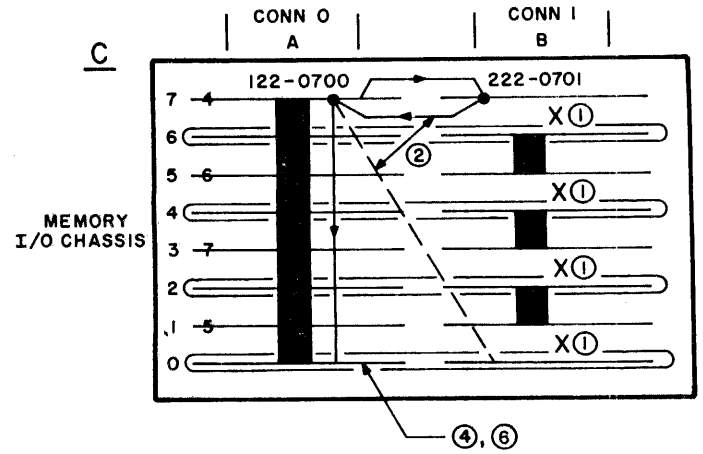
④ IF REQUIRED, EXTEND THE SELECTOR CHANNEL BUS TO OTHER CHASSIS BY INSTALLING A CABLE HERE.

⑤ EXTEND THE MULTIPLEXOR BUS TO OTHER CHASSIS BY INSTALLING A CABLE HERE.

⑥ INSTALL I/O TERMINATORS 35-433 ROI HERE.

NOTE 1:
IF A MEMORY I/O CHASSIS IS USED IN THE SYSTEM, ANY SELECTOR CHANNELS MUST BE INSTALLED IN THAT CHASSIS.

NOTE: 2
A SELECTOR CHANNEL MAY NOT BE INSTALLED IN SLOT 2 OF THE 7/16 HSALU. SLOT 0 IS THE ONLY SLOT IN WHICH A SELCH MAY BE INSTALLED.



TO INSTALL 4 SELECTOR CHANNELS (IN SLOTS 6, 4, 2 AND 0) OF THE MEMORY I/O CHASSIS ——

① CUT THE MULTIPLEXOR BUS IN FOUR PLACES.

② JUMPER RACKO/TACKO AS SHOWN, REMOVE DASHED JUMPER.

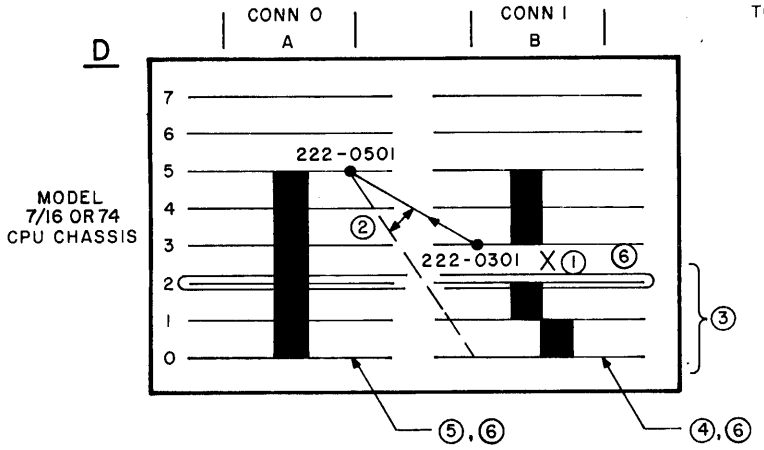
③ EACH SELCH, EXCEPT THE ONE IN SLOT 0, HAS ONE SLOT AVAILABLE ON IT'S PRIVATE BUS. THE PRIVATE BUSSES CAN BE EXTENDED TO OTHER CHASSIS BY INSTALLING CABLES IN SLOT POSITIONS 0, 1, 3 AND 5 ON CONNECTOR ONE (CONN.1) SIDE.

④ ALL SLOTS ON THE CONNECTOR ZERO (CONN.0) SIDE REMAIN AS THE STANDARD MULTIPLEXOR BUS. THIS BUS CAN BE EXTENDED BY INSTALLING A CABLE HERE.

⑤ MEMORY MODULES 5 AND 7, IF THEY EXIST, MUST BE LOCATED IN SLOTS 1 AND 3.

⑥ INSTALL I/O TERMINATORS 35-433 ROI HERE.

Figure 1. Backpanel Modifications



TO INSTALL A SELECTOR CHANNEL IN SLOT 2 OF THE CPU CHASSIS——

① CUT THE MULTIPLEXOR BUS BY REMOVING THE TOP WRAPS.

② JUMPER RACKO/TACKO AS SHOWN, REMOVE DASHED JUMPER.

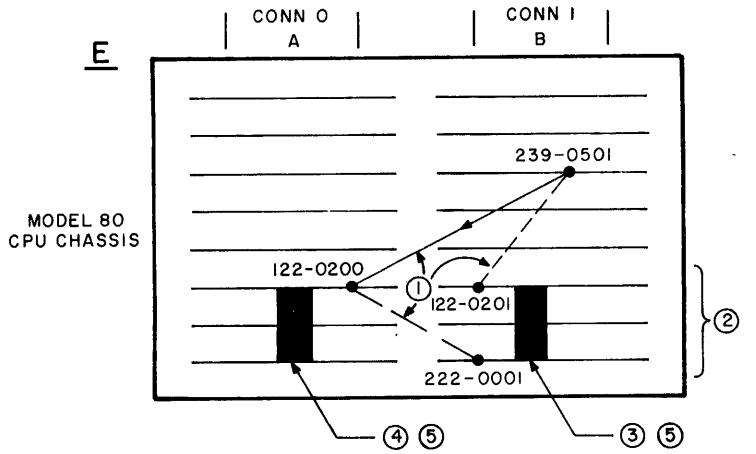
③ THIS SECTION BECOMES THE PRIVATE SELECTOR BUS ON THE CONNECTOR ONE (CONN.1) SIDE ONLY. ALL SLOTS ON THE CONNECTOR ZERO (CONN.0) SIDE REMAIN AS STANDARD MULTIPLEXOR BUS SLOTS.

④ IF REQUIRED, EXTEND THE SELECTOR CHANNEL BUS TO OTHER CHASSIS BY INSTALLING A CABLE HERE.

⑤ EXTEND THE MULTIPLEXOR BUS TO OTHE CHASSIS BY INSTALLING A CABLE HERE.

⑥ INSTALL I/O TERMINATORS 35-433 ROI HERE.

NOTE:
IF A MEMORY I/O CHASSIS IS USED IN THE SYSTEM, THE SELECTOR CHANNEL MUST BE IN SLOT 0 OF THE CPU CHASSIS OR IN SOME SLOT OF THE EXPANSION.



TO INSTALL A SELECTOR CHANNEL IN SLOT 0, 1 OR 2 CHASSIS——

① JUMPER RACKO/TACKO AS SHOWN, REMOVE DASHED JUMPERS.

② THIS SECTION BECOMES THE PRIVATE SELCH BUS ON THE CONNECTOR ONE (CONN.1) SIDE ONLY. ALL SLOTS ON THE CONNECTOR (CONN.0) SIDE REMAIN AS STANDARD MULTIPLEXOR SLOTS.

③ EXTEND THE SELCH BUS TO OTHER CHASSIS BY INSTALLING A CABLE HERE.

④ EXTEND THE MULTIPLEXOR BUS TO OTHER CHASSIS BY INSTALLING A CABLE HERE.

⑤ INSTALL I/O TERMINATORS 35-433 ROI HERE.

NOTES:
1. IF A 17-183 CABLE IS INSTALLED BETWEEN CONNECTORS ZERO AND ONE IN THE CPU CHASSIS, THIS CABLE MUST BE REMOVED PRIOR TO INSTALLING SELCH.
2. THE INSTALLATION OF A SELCTOR CHANNEL OR OTHER I/O DEVICE CONTROLLER IN THE M80 CPU CHASSIS REDUCES THE MAXIMUM MEMORY SIZE BY 16K BYTES (ONE MSU) FOR EACH SLOT USED!
3. ONLY ONE SELECTOR CHANNEL MAY BE CONFIGURED IN THE MODEL 80 PROCESSOR CHASSIS.

Figure 1. Backpanel Modifications
(Continued)

3.1.2 ACT0/TAC0. The ACT0/TAC0 jumper between Pins 137-0 and 237-0 must be removed from the slot used by the SELCH controller. If the Selector Channel is not the first Direct Memory Access (DMA) channel on the Memory Bus, jumper "K" on the SELCH controller must be removed. Note that on a Model 74 only one DMA device is permitted.

NOTE (Not Applicable on Model 74)

On installations with Multiple SELCH's, remove the "EN0" and the "INH0" filters on all but the last SELCH (Remove the following: R70, R72, C59 and C61).

3.2 Cabling

The cabling necessary for the SELCH depends on the system's physical configuration. When the SELCH Bus does not extend outside the chassis, no cabling is required. When the SELCH Bus must be extended to another chassis, a number of cable configurations can be used. See Figure 2. Care should be taken to minimize bus lengths.

See Figure 2 for a summary of cables.

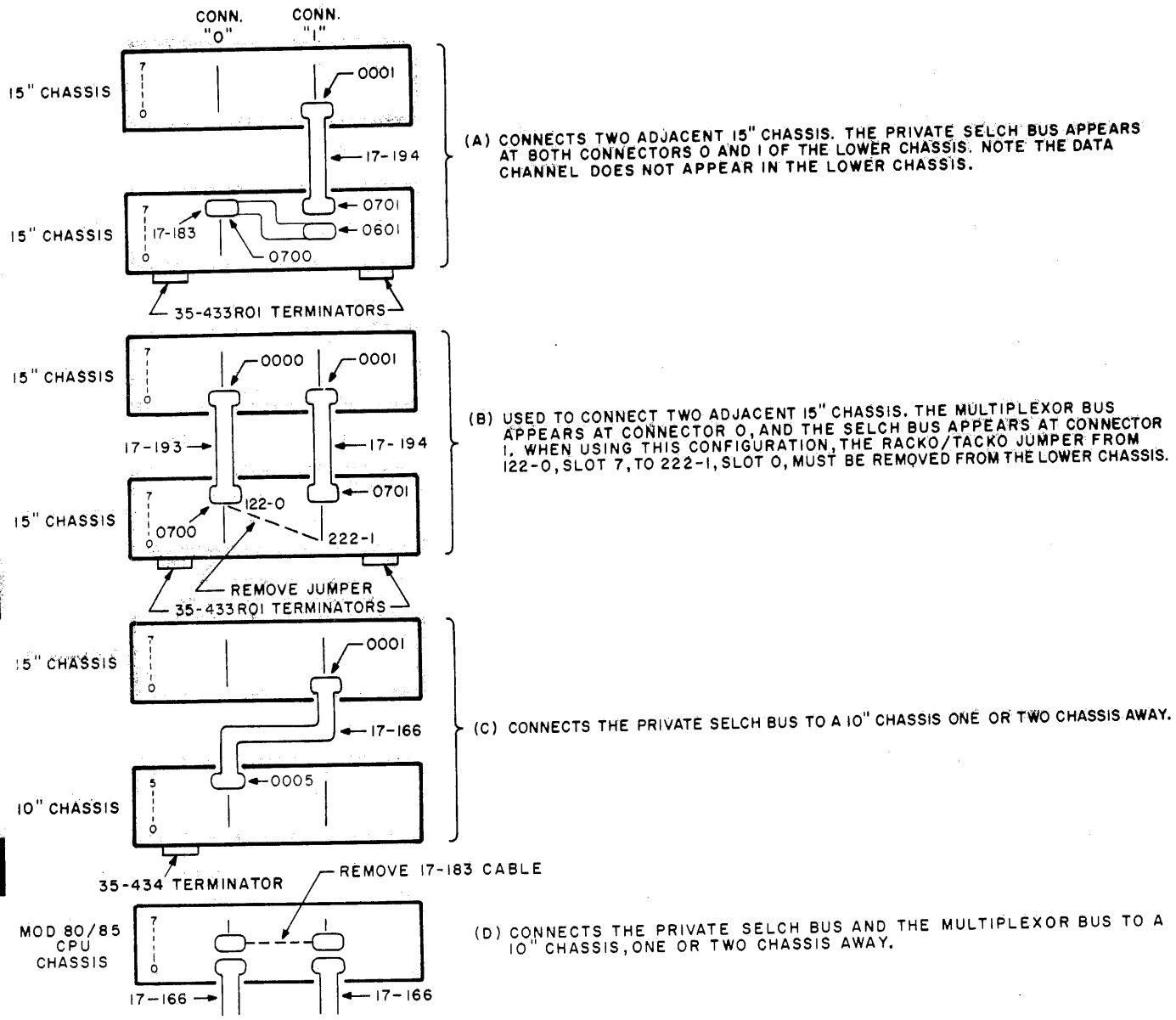


Figure 2. Cabling

4. ADDRESS STRAPPING

The preferred address of the NS Selector Channel is X'F0'. The controller is strapped for this address at the factory. To change the address, refer to Functional Schematic 02-232M01D08. The number and letter designations shown on the schematic refer to the designations on the apparatus side of the SELCH controller board.

5. MODEL 80/85 STRAPPING

For use with the Model 80 or 85 the following options must be exercised:

1. Remove the jumper labled J located between IC 14 and 15.
2. Change the jumper, above IC 53, from (L to X) to (L to 2).

6. INSTALLATION CHECKS

The NS SELCH is factory tested using a special high speed device. Therefore, field checks are contingent upon the user having appropriate hardware and software available with which to exercise the Selector Channel. When the SELCH is used with Model 80 or 85, insure that the strap options on the SELCH have been made according to Section 5.

M70-103
NS SELECTOR CHANNEL
MAINTENANCE SPECIFICATION

1. INTRODUCTION

The 02-232M01 NS Selector Channel (SELCH) (Product Number M70-103) is a Direct Memory Access port (DMA) which provides block data transfer between a device controller and memory. Once initiated, the transfer is independent of the Processor. The Processor sets up the device controller, loads the SELCH with the starting and final addresses of the memory block, specifies the type of operation (Read or Write), and issues a GO Command. The SELCH then handles the transfer without further direction by the Processor.

The NS Selector Channel is complete on one printed circuit board and occupies one slot in a system chassis. The SELCH provides the drivers, receivers and termination resistors for the private SELCH Bus. This bus originates at Connector One (1) of the SELCH slot and extends to each lower numbered slot in the system chassis on the Connector One (1) side only. The private SELCH Bus can be extended to other chassis, as required. For installation information, refer to Installation Specification 02-232M01A20.

2. SCOPE

This specification describes the operation of the SELCH in its various modes; Setup, Memory Read, Memory Write, and Termination. Where necessary, this specification references the Multiplexor Channel Bus and Memory Bus operations.

3. BLOCK DIAGRAM ANALYSIS

Refer to the SELCH block diagram on Sheet 7 of Functional Schematic 02-232M01D08, and the SELCH Flow Chart, Figure 1, during the following analysis. Before initiating a data transfer via the SELCH, the device controller and the SELCH must be set up. The setup procedure is implemented by the Processor via the Multiplexor Bus (MPX-Bus). When the SELCH is in the Idle mode, the MPX-Bus is tied directly to the private SELCH Bus through the SELCH. This allows the Processor to communicate directly with any device on the private SELCH Bus.

To prepare the SELCH for data transfer, the Address Register (AR) and Auxiliary Address Register (AAR) must be loaded with the starting address in memory where the transfer is to begin, and the Final Address Register (FAR) must be loaded with the address of the last memory location to be accessed. These registers are loaded from the eight least significant Data Lines D080:150 by four consecutive Data Availables (DAs) from the Processor. The first two Data Availables simultaneously load the AR and AAR, which are 16-bit incrementing registers. The AR is incremented, by two, after each halfword is transferred to/from memory, and the AAR is incremented, by one, with each byte transferred to/from the device. Data transfer is terminated when the AAR is equal to the FAR or when the AAR increments past its maximum value, X'FFFF'.

Data transfer is begun by the Processor issuing a GO Command to the SELCH. Transfer to/from the device is now independent of the Processor. The GO Command also prevents communication between the Processor and any device on the private SELCH Bus until the transfer is terminated and the SELCH is addressed.

Data transfer is controlled in the Move Data circuit by inspection of the four least significant bits of the Status Byte presented by the active device on the private SELCH Bus. When any one of the three least significant bits are set, (EX, EOM, or DU), the transfer is terminated. Bit-12 (Busy) regulates the rate of data transfer. In the Memory Read mode, the actual data transfer begins with a memory request, REQ0 active, as soon as a GO Command is issued. When the memory request is serviced by the Processor, the SELCH Memory Bus Control circuit activates Select (SEL), which gates the contents of the Address Register (AR) onto the Memory Address Bus, and gates a halfword of data from memory into the Data Register (DR). At the termination of the memory transfer, the data is loaded from the DR to the Data Buffer (DB) and the AR is incremented.

NOTE

Unless the SELCH has dropped REQ0 in time to remain selected during the next memory cycle, the SELCH is deselected by the rising edge of Inhibit (INH0) after the halfword has been transferred.

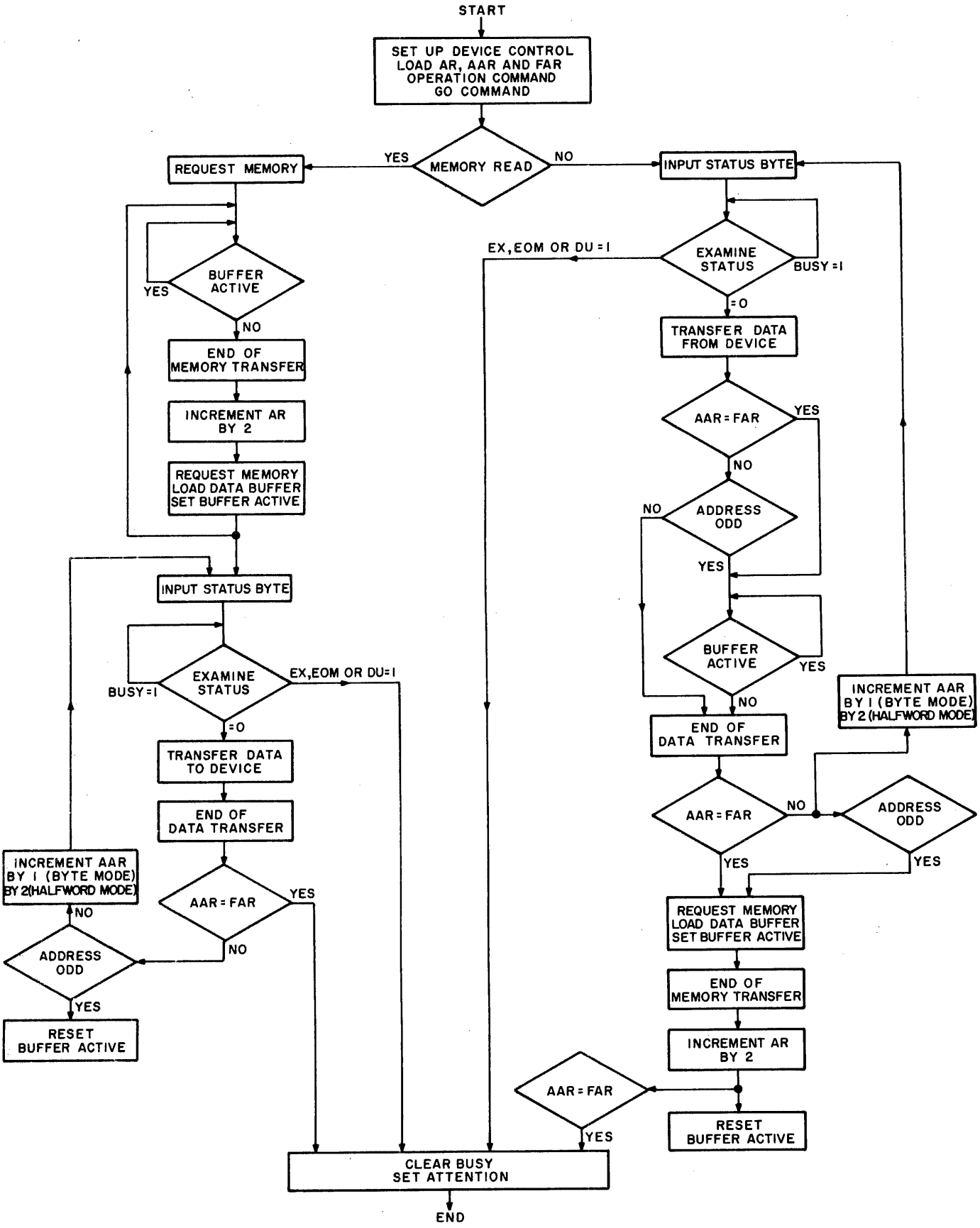


Figure 1. Flow Chart

Once the DB is loaded, data transfer to the device over Private Data Lines PD000:150 is initiated and, when applicable, a request is made to fetch the next halfword from memory. This cycle continues until either a match is detected between the contents of the Auxiliary Address Register and contents of the Final Address Register, or until the transfer is aborted due to an error condition.

In the Memory Write mode, the data transfer sequence described previously for Memory Read mode is reversed. That is, two bytes of data are loaded into the DR from the device prior to a memory request and the data flow is from the device to the DR, DR to the DB, and finally into memory.

The Branch Gate circuit and the Move Data circuit control the flow of data between memory and the device. The Branch Gate supervises the overall data flow, while the Move Data circuit performs the handshaking between the SELCH and the active device on the private SELCH Bus.

Upon termination of the data transfer, the program is notified via an interrupt and by the inactive state of the SELCH Busy flip-flop which is presented to the program as Bit-12 of the SELCH Status Byte.

Selector Channel Status and Command Byte Data is shown on Table 1.

TABLE 1. SELECTOR CHANNEL STATUS AND COMMAND BYTE DATA

BIT NUMBER	0	1	2	3	4	5	6	7
STATUS BYTE					BSY			
COMMAND BYTE			READ	GO	STOP			

BSY — When this bit is set, a one shot generates the EBS1 pulse to start the appropriate SELCH operation. When this bit is cleared an interrupt is generated.

READ — This command, Bit-2, sets the Memory Write (WT) flip-flop. The controller on the SELCH Bus is setup for a device Read operation.

GO — This command, Bit-3, clears the MSC flip-flop and sets the BSY flip-flop to initiate the Data Transfer mode.

STOP — This command, Bit-4, from the Processor clears the BSY flip-flop and initializes the Load/Unload Sequencer. During the Data Transfer mode, execution of the command is delayed until the end of a memory cycle, if one is in progress.

4. FUNCTIONAL DIAGRAM ANALYSIS

4.1 Introduction

This section relates to Functional Schematic 02-232M01D08, Sheets 1 through 6. Note that in INTERDATA functional schematics, the last character in the mnemonic symbol designates the logic level when the signal is active. For example; D080 is Data Line Number 8 (D08). The last character (0) inidicates that when D080 is active, the line is at a logical Zero level.

4.2 SELCH Control Circuit

In the Idle mode, the SELCH Address (2M8), Busy (3F3), and Multiplexor-SELCH (MSC) (3FA) flip-flops are reset and the private SELCH Bus is tied directly to the Multiplexor Bus. This allows the Processor to communicate, via the Multiplexor Bus, with any device on the private SELCH Bus.

To communicate with the SELCH, it must first be addressed. The SELCH Address (X'F0' preferred) is placed on Data Lines D080:150 (1A3-8) and the Address control line is activated (ADRS0)(4B8). The SELCH Address is decoded by the four input NAND gate (1F4) and the Address flip-flop is set (2M8). The set output from the Address flip-flop (AD1)(2J7), when active, prevents the control signals on the MPX-Bus from passing onto the private SELCH Bus by holding the Control Line Gate inactive (CLG1) (1B2). Capacitors C33 and C34 (4D8) delay the propagation of the Private Address control line (PADRS0) (4F9) to the SELCH Bus, so that when the SELCH is being addressed PADRS0 does not become active. This delay allows the SELCH to be addressed without resetting the Address flip-flop of the active device on the private SELCH Bus.

The simultaneous loading of the Address Register (AR) and Auxiliary Address Register (AAR), and the loading of the Final Address Register (FAR) is accomplished by four consecutive Data Availables (DAs) from the Processor. The Load/Unload Sequencer (2L2) controls the loading of these registers (AR, AAR, and FAR) and the unloading of the AAR. The sequencer is set to its initial state by the termination of the last data transfer, a Stop Command, or a System Clear (SCLR0)(4MS) so that the first DA, through Data Available Gate (DAG0)(2L4), will activate Load Address Register High (LARH0)(2S2).

LARH0 gates Data Lines DA081:151 (1D3-8) into the eight most significant bits of the AR and AAR. The rising edge of the first and each successive Data Available Gate (DAG0)(2L4) increments the sequencer and allows the next DA to activate the next load line. The second DA loads the eight least significant bits of these registers. The third and fourth DAs will then load the Final Address Register in the same order. The contents of the AAR may be inspected, via the program, by issuing two Data Requests (DR) to the SELCH whenever the Load/Unload Sequencer is in its initial state. (e.g., Upon termination of a SELCH transfer, sequencer initialized, the FAR may be inspected to determine if the entire block of data had been transferred.)

If a Memory Write operation is desired, an Output Command with Bit-10 set must be issued to set the Write flip-flop (3F5). Since the Write flip-flop is reset by the Data Available/Request Gate (DARG1)(2L5) whenever a DA or DR is sent to the SELCH (setup procedure), no command is necessary to initiate a Memory Read operation.

Data transfer commences with a GO Command from the Processor, which is an Output Command with Bit-11 set. The GO Command sets both the Busy (3F3) and MSC (3F4) flip-flops. The setting of the Busy flip-flop causes an End of Busy Set pulse (EBS1) (3H4) to be generated. This pulse is generated from the falling edge of BSY0 (3F3), and is used by the Branch Gate circuit to initiate the transfer cycle. The Busy latch circuit remains set until the Selector Channel detects the termination of transfer and its state is presented to the program via Bit-12 of the Sense Status Byte.

The MSC flip-flop is reset by SCLR0A or by addressing the SELCH, Set Gate active (SGAD1)(2L9), when the Busy flip-flop is reset. The resetting of the MSC flip-flop, MSC0 active, clears any pending interrupt in the Selector Channel.

Information is steered from the SELCH to both the Data Lines D080:150 (1B4-9) and the Private Data Lines PD080:150 (1S4-9) by the proper gating of four each, four-to-one line multiplexors (Sheet 1). For example; with the SELCH idle, Busy reset, all Data Lines are tied directly to the Private Data Lines in both directions.

4.3 Memory Bus Control Circuit

Memory Bus Control timing relationships are shown in Figure 2. A SELCH request for memory is started by activating Set Request (SREQ0)(3S4). SREQ0 is activated by the Branch Gate circuit (3M8) when either the SELCH has received a halfword of data from the device or, in the Memory Read mode, whenever the Memory Data Register is available to accept the next halfword.

SREQ0 is applied to the direct set input of the Request flip-flop (REQ)(4L5) which sets the flip-flop and sends REQ0 to the Processor. When REQ0 is received, the Processor generates Enable (EN0). EN0, on the first DMA device, is jumpered to Accept (ACT0) which generates the daisy chain priority loop through all DMAs in the system. The daisy chain begins at the highest priority DMA as EN0, and propagates to the lower priority DMAs as Transmit Accept (TAC0) until it is captured by the first DMA requesting a memory cycle. When the REQ flip-flop is set and ACT0 (EN0) is active, the ACT0/TAC0 contention circuit (4H2) blocks the propagation of TAC0, and provide highs on the J input to the SEL flip-flop and the K input to the REQ flip-flop. Thus, on the rising edge of EN0, the Select flip-flop becomes set and the Request flip-flop is reset. When the SELCH REQ flip-flop is reset and ACT0 (EN0) is active, the ACT0 signal is propagated as TAC0 to the DMA with a lower priority.

The trailing edge of Inhibit (INH0) (4H5), from the Processor, indicates the end of the memory cycle. This edge, unless the SELCH has dropped REQ0 in time to remain selected during the next cycle, causes the SEL flip-flop to reset. Two pulses, End of Memory Transfer (EMX0) (4M7) and Inhibit (INH0P) (4M2), are generated by the leading and trailing edges of INH0 respectively. EMX0 is used by the Branch Gate circuit to indicate the end of the memory transfer. The Address Register is toggled by the AND function of SEU and INH1.

In the Memory Read mode, the Memory Data Register (MDR) is cleared by Clear Data Register (CDR0) (4R2) which is generated by the trailing edge of REQ1. Write Not (WT0A) (3F5) is ANDed with SEL1 to form Enable Memory Data Read (ENMDR1) (4R6), which gates the contents of the MDR onto the Memory Data Lines MD000:150 (Sheet 6) for the restore portion of the memory cycle. (This function is disabled when using solid state memory.) The contents of the memory location accessed is gated to the direct set inputs of the MDR by Enable Memory Strobe (ENMS1) (4R4). This function is WT•SEL•CDR0 for use with core memory and WT•SEL•INH when using solid state memory (4R4).

When writing to memory, the contents of the Data Buffer is gated onto Memory Data Lines MD000:150 (Sheet 6) by Enable Memory Data Write (ENMDW1)(4R3), when selected. A Memory Write operation is indicated to the Processor by activating WRT0A (4S3).

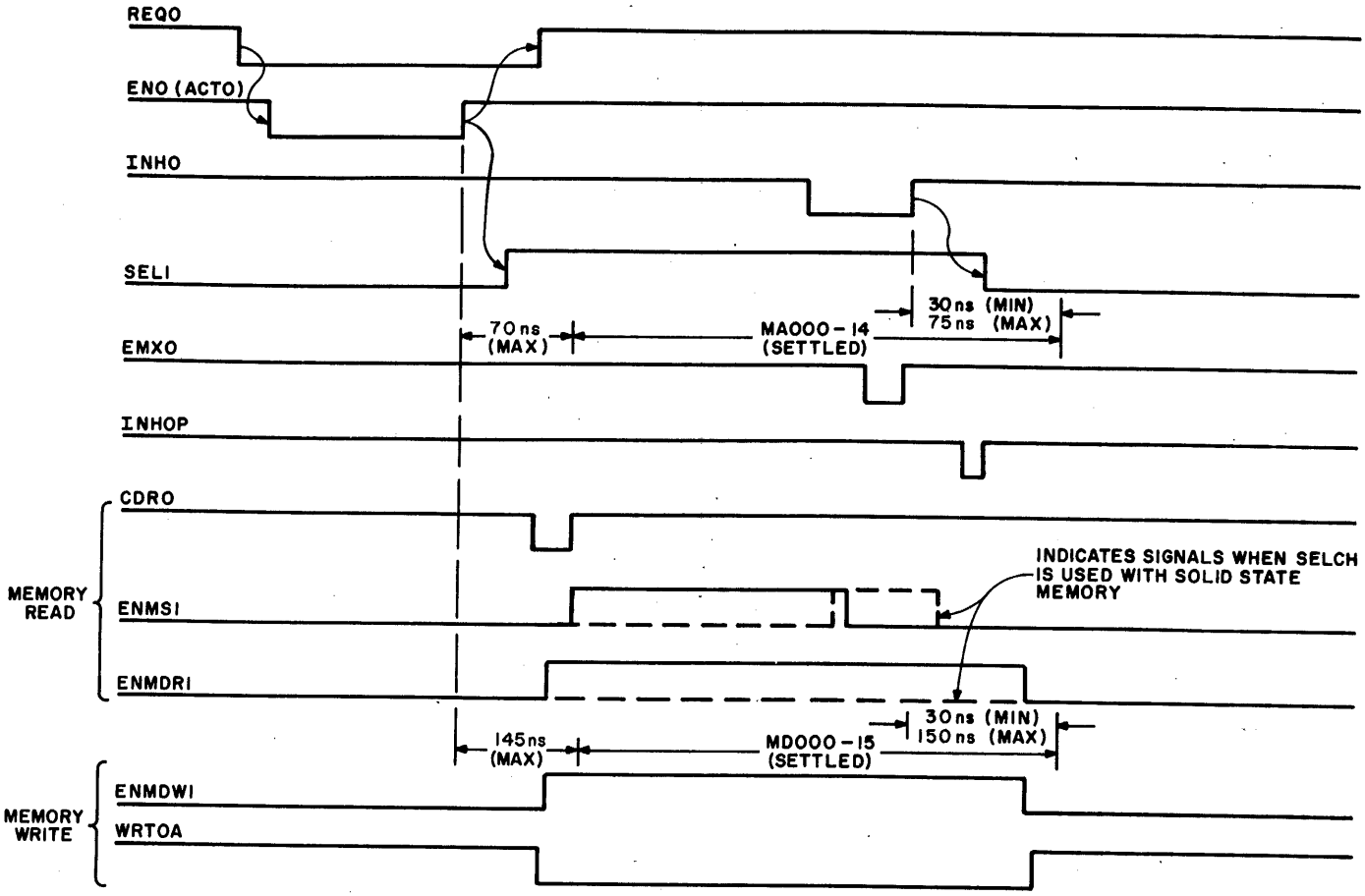


Figure 2. Memory Bus Control Timing Diagram

4.4 Address Register and Auxiliary Address Register.

The Address Register (AR) and Auxiliary Address Register (AAR) (Sheet 5) each consist of four, four-bit counters. These registers are loaded simultaneously by the Processor from Data Lines D080:150 (1A3-8), under control of the Load/Unload Sequencer (as discussed in Section 4.2), with the starting address from which the block transfer is to begin. The contents of the AR (Sheet 5) is gated onto Memory Address Lines MA000:140 (5R1-8) whenever the SELCH is selected, SEL flip-flop set. The AR is incremented with each memory transfer by Select Inhibit (SINH0) (4K8). The AAR (Sheet 5) keeps track of the transfer between the SELCH and the device. This register is incremented, by one, for each byte of data transferred by Toggle Auxiliary Address Register (TAAR0) (3M1). When the transfer is in the Half-word mode, TAAR0 is generated twice for each transfer. The outputs of the AAR are used by the Match circuit to determine the end of the data block. Its contents may be examined, via the program, by issuing two consecutive DRs to the SELCH when the sequencer is initialized. In addition, AAR151 is used in the Byte Transfer mode to determine whether the byte being transferred is odd or even, for byte steering. Carry Out from the most significant stage of the AAR (CO0) (541) terminates the transfer, clear Busy, when a transfer is attempted past the maximum memory address. This feature prevents 'wrap-around' in memory.

4.5 Final Address Register

The Final Address Register (FAR) is implemented by four quad latches (Sheet 5). This register, like AR and AAR, is loaded by the Processor under control of the Load/Unload Sequencer. The outputs of this register are used exclusively by the Match circuit to determine when the final address of the transfer is reached.

4.6 Memory Data Register and Data Buffer

The Memory Data Register (MDR)(Sheet 6) is a16-bit register composed of 16 edge triggered JK flip-flops. In the Memory Read mode, the MDR is first cleared by Clear Data Register (CDR0) (4R2) and then direct set by each active bit on Memory Strobed Data Lines MS000:150 (Sheet 6). During a Memory Write, the data, in double rail format, present at the J and K inputs to the MDR, is toggled into the flip-flops on the trailing edge of either Load Data Register High (LDRH0)(649) or Load Data Register Low (LDRL0)(6J9).

As soon as the MDR is loaded, if the Data Buffer (DB) is empty (as determined by the inactive state of the Buffer Active flip-flop)(3H1), the MDR contents are loaded into the DB (Sheet 6) by Load Data Buffer (LDB1)(357). Information present in the DB is, in turn, either written into memory via Memory Data Lines MD000:150 or sent to the device on Private Data Lines PD000:150.

4.7 Data Transfer Circuit

Refer to Figure 3 for Memory Read timing diagrams and Figure 4 for Memory Write timing diagrams. The Memory Read timing diagram shows the timing of a three byte transfer, in the Byte mode, of 2,000,000 bytes/second. Figure 4 shows a transfer of two halfwords, in the Halfword mode, to a slower device.

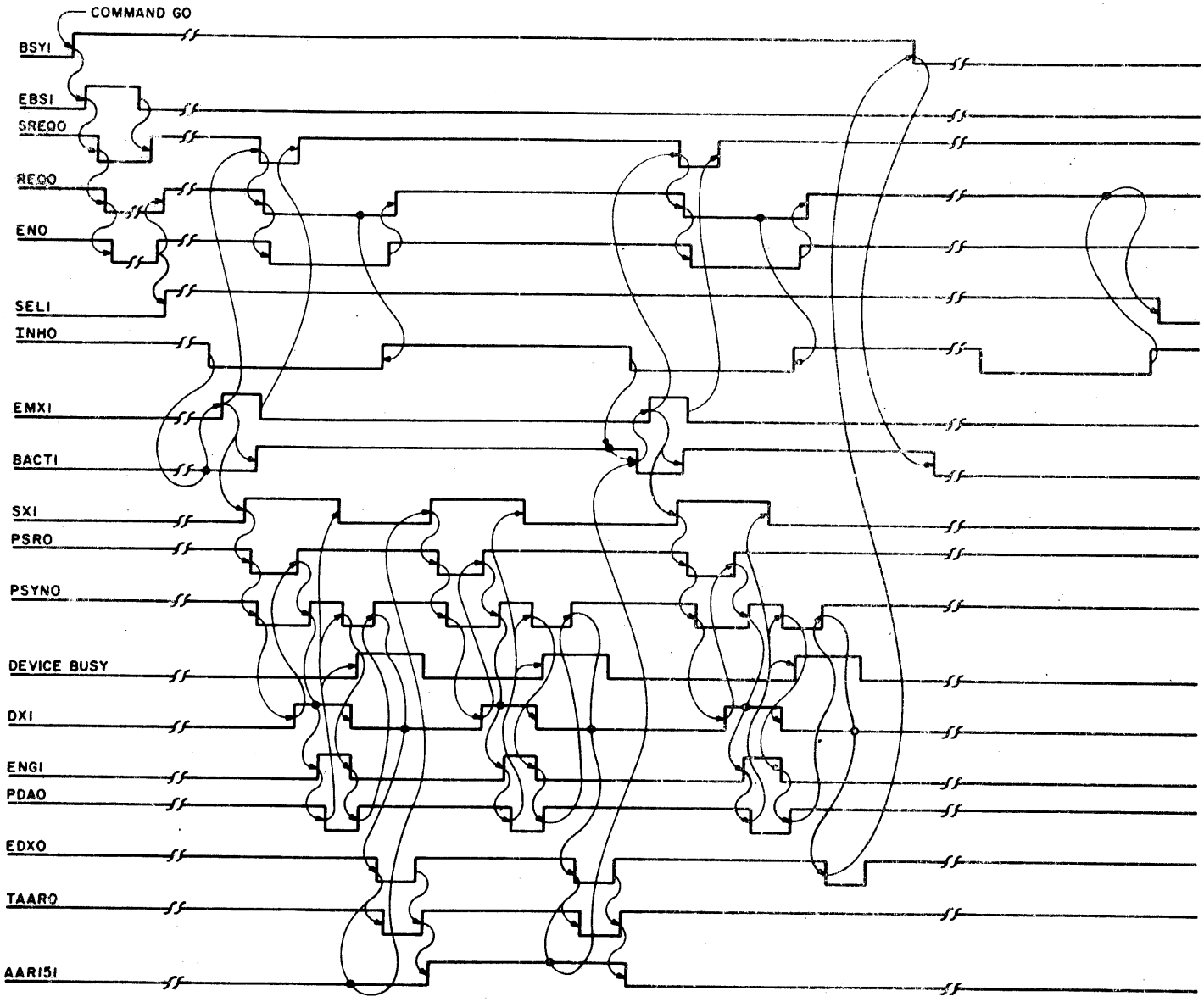


Figure 3. Memory Read (Byte Mode)

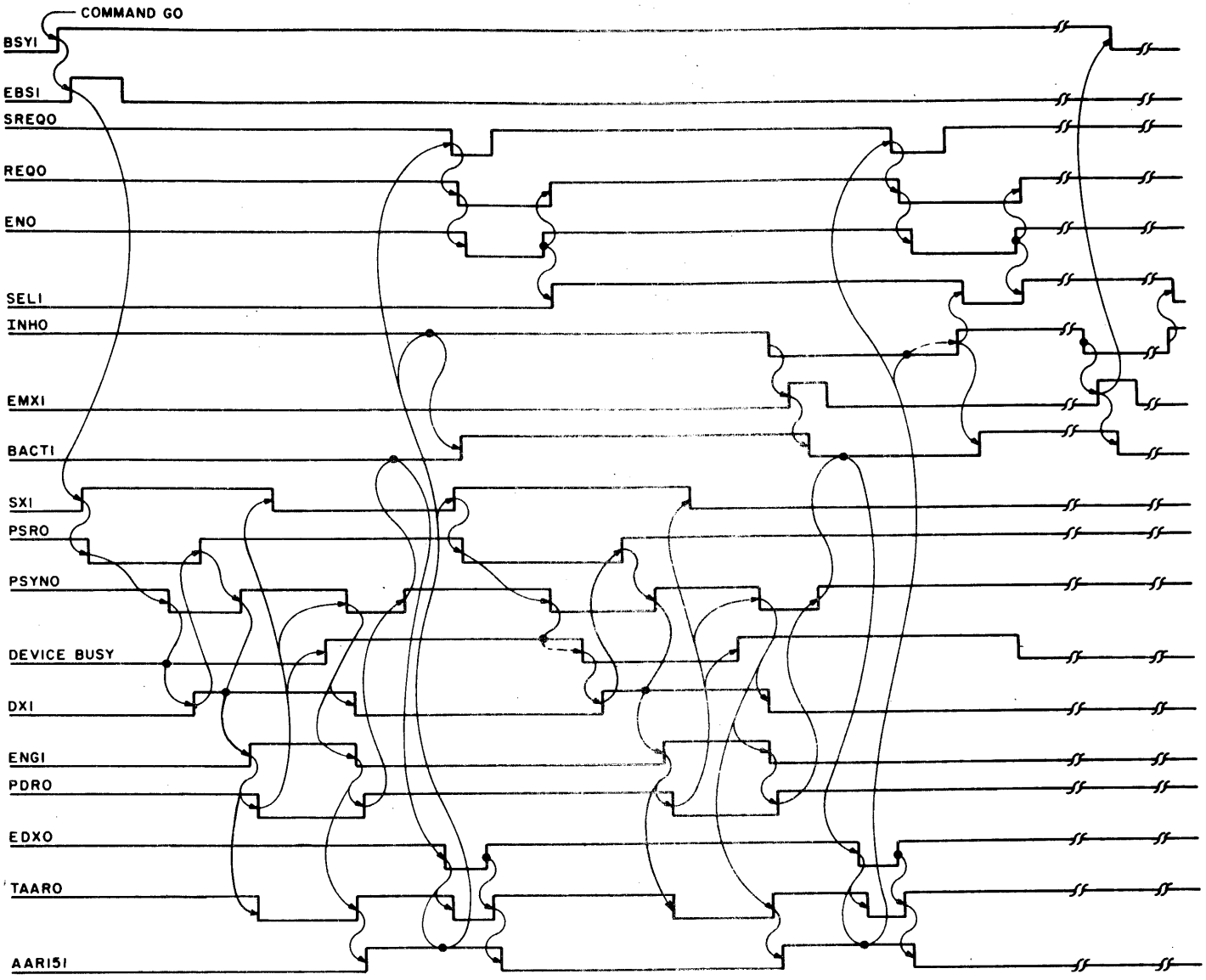


Figure 4. Memory Write (Halfword I/O Mode)

A GO Command to the Selector Channel sets the Busy flip-flop which generates the End of Busy Set pulse (EBS1)(3K8).

In the Memory Read mode, EBS1 is decoded by the Branch Gate circuit and SREQ0 is generated. Thus, a request for memory is initiated. When the halfword of data is present in the MDR, the End of Memory Transfer pulse (EMX1)(4R6) becomes active and the Branch Gate circuit once again requests memory and generates Set Status Transfer (SSX0)(3S5) and Load Data Buffer (LDB1) (3S7). These signals initiate the transfer to the device and load the Data Buffer (DB) respectively.

SSX0 sets the Status Request flip-flop (3F6) which activates the Private Status Request control line (PSR0)(3G6) to the active device on the private SELCH Bus. This Status Request examines the four least significant bits of the Status Byte. If any of the three least significant bits (EX, EOM, or DU) are set, the transfer is terminated by resetting the Busy flip-flop (3F3). The assumption is made that each of these status bits remain reset for the remainder of this discussion. With Bit-12 (Busy) of the Status Byte reset, the Data Transfer flip-flop becomes set (3F7). Data Transfer (DX0)(3E1) inhibits the generation of PSR0, which causes Private Sync (PSYN1)(4B4) from the device to become inactive. This enables Engage to go high (ENG1)(3D8), which allows the Private Data Available control line (PDA0)(3H5) to become active. The Private Data Available/Request signal (PDAR1)(3H5), generated whenever a Private Data Available (PAD0) or Private Data Request (PDR0) signal is active, will then clear the Status Request flip-flop. Upon receipt of Sync from the device, PSYN1 active, the Data Transfer flip-flop becomes reset and ENG1 goes low, disabling PAD0. When the Sync is removed by the device, an 80 nanosecond End of Data Transfer pulse is generated (EDX0) (3J8) which increments the Auxiliary Address Register and is used by the Branch Gate circuit to generate a function in accordance with the truth table for EDX on Sheet 3. This cycle continues until termination of the transfer is detected.

In the Memory Write mode, WT1 active (3F5), EBS1 is used to generate SSX0, and the Branch Gate circuit directs the loading of a halfword of data into the DB before a memory request is made. The transfer of data from the device is the same as described in the Memory Read mode, except that ENG1 is used to generate the Private Data Request control line (PDR0)(3H5) rather than PDA0. Data from the device is loaded into the MDR on the trailing edge of either Load Data Register High (LDRH0)(2R2) or Load Data Register Low (LDRL0)(2R2), depending on which eight bits are being loaded. In the Halfword Transfer mode, both LDRH0 and LDRL0 are generated simultaneously. With WT1 active, the generation of EDX1 is delayed by activating the clear input to the one-shot (3H8) when the Buffer Active flip-flop is set (BACT1)(3H1), if either the transfer to the device is on an odd boundary or when a Match is detected (MCH0)(5J2). This prevents the reloading of the Data Buffer (DB) before the last halfword has been written into memory.

4.8 RACK0/TACK0 Contention Circuit

The Selector Channel directs the propagation of the Acknowledge signal to lower priority devices on the Multiplexor Channel Bus as well as devices on the private SELCH Bus. If the Selector Channel Attention flip-flop (4B4) is set, the SELCH captures the Receive Acknowledge signal (RACK0) (4B3), place its device address on the Data Lines and return Sync to the Processor, Attention Sync (ATSYN0)(4F4) active. If the Attention flip-flop is reset, RACK0 is propagated as either Private Transmit Acknowledge (PTACK0)(4F2) or Transmit Acknowledge (TACK0)(4F3). Since devices on the private SELCH Bus have a higher interrupt priority than devices below the SELCH on the MPX Bus; if the Private Attention Test line is active (PATN0) (4B1), PTACK0 is generated rather than TACK0. Note that when MSC0 is high (3F4), PATN0 is disabled so that a device on the private SELCH Bus may not interrupt the Processor while the SELCH is active.

5. MAINTENANCE, TROUBLE SHOOTING, AND TEST

Before attempting any maintenance or testing, insure that the necessary back panel modifications and SELCH board option strapping have been made in accordance with the NS Selector Channel Installation Specification 02-232M01A20.

To insure a 2,000,000 Byte/second transfer rate in the Byte Transfer Mode, it is necessary to limit the maximum delay between PDA0, PDR0, and PSR0 and the return of Sync from the device (PSYN0), to 30 nanoseconds. In addition, the device must be ready for the next byte of data, Busy Status Bit reset, whenever a Status Request is made. Field testing of this device is contingent upon the user having appropriate software and hardware available with which to exercise the Selector Channel. There are no adjustments associated with this device. Do not install Terminator Boards 35-433 or 35-434 on the SELCH bus if a transfer rate of 2,000,000 Bytes/second is to be maintained in the Byte (8 Bit) Mode. The SELCH Bus should be contained on a single 15 inch chassis if no terminators are used.

6. MNEMONICS

The following list provides a brief description of each mnemonic found in the SELCH. The source of each signal on Functional Schematic 02-232M01D08 is also provided.

MNEMONIC	MEANING	SCHEMATIC LOCATION
AAR001:151	Outputs of the Auxiliary Address Register	5F1-5F9
ACT0	Accept - Request for memory accepted by Processor	4H1
AD1	Address - Active when SELCH is addressed	2K7
ADRS0	Address control line from MPX-Bus	4B8
AG081:151	Address Gated Lines - Output of Address Strap	1E3 - 1E8
ATN0	Attention - Attention to Processor	4F1
ATSYN	Attention Sync - Generated by an Acknowledge Attention from Processor	4F4
BACT1	Buffer Active - Indicates that valid data is present in the DB	3H1
BSY	Busy - Indicates a data transfer is in progress	3F3
CDR0	Clear Data Register - Clears MDR prior to loading from MS000:150	4R2
CBSY0	Clear Busy - Terminates transfer when a match is detected	3M3
CL070	Control Line 7 - Control Line from MPX-Bus	4B6
CLG1	Control Line Gate - Gates Private Control Lines	1C2

MNEMONIC	MEANING	SCHEMATIC LOCATION
CLUS0	Clear Load/Unload Sequencer - Clears Sequencer	3S2
CMD0	Command Control Line from MPX-Bus	4B8
CMG	Command Gated by AD1	2S7
CO0	Carry Out of the AAR - Prevents Memory 'Wrap-around'	5A1
D000:150	Data Lines from MPX-Bus	1A3 - 1A8 2A1 - 2A8
DA0	Data Available Control Line from MPX-Bus	4B7
DLG0	Data Line Gate - Gates Data Lines and Private Data Lines	1R2 - 1D3
DB001:151	Outputs of Data Buffer	6G2 - 6G9 6R2 - 6R9
DR0	Data Request Control Line from MPX-Bus	4B7
DRG0	Data Request Gated by AD1	2L5
DX	Data Transfer flip-flop	3F8
EBS1	End of Busy Set - Signals the start of a SELCH transfer	3J4
EDX1	End of Data Transfer - Signals the end of a device transfer	3J8
EMX1	End of Memory Transfer - Signals the end of a memory transfer	4R7
EN0	Enable from Memory Bus	4H2
ENG1	Engage - Gates either PDS0 or PDR0	3D8
ENMDR1	Enable Memory Data Register Read - Gates contents of MDR to MD000:150	4R6
ENMDW1	Enable Memory Data Register Write - Gates contents of DB to MD000:150	4R3
ENMS1	Enable Memory Strobe - Gates contents of MS000:150 to MDR	4R4
HW0	Halfword Control Line from MPX-Bus	1D2
INH0	Inhibit from Memory Bus	4H5
LARH0	Load Address Register High - Loads AAR and AR, Bits 00:07	2R2
LARL0	Load Address Register Low - Lards AAR and AR, Bits 08:15	2R2
LDB1	Load Data Buffer - Load contents of MDR to DB	3S2
LDRH0	Load Data Register High - Loads MDR Bits 00:07	3F8
LDRL0	Load Data Register Low - Loads MDR Bits 08:15	3H9
LFRH0	Load Final Address Register High - Loads FAR Bits 00:07	2S3
LFRL0	Load Final Address Register Low - Loads FAR Bits 08:15	2S3
MA000:140	Memory Address Lines to Memory Bus	5R1 - 5R8
MCH1	Match - Indicates a match between AAR and FAR	5J6

MNEMONIC	MEANING	SCHEMATIC LOCATION
MD000:150	Memory Data Lines to Memory Bus	6G1 - 6G8 6R1 - 6R8
MS000:150	Memory Strobed Data Lines from Memory Bus	6A1 - 6A8 6H1 - 6H8
MSC0	Multiplexor SELCH Control flip-flop	3F4
PADRS0	Private Address Control Line to SELCH Bus	4F8
PATN0	Private Attention from SELCH Bus	4B1
PCL070	Private Control Line 7 to SELCH Bus	4R6
PCMD0	Private Command Control Line to SELCH Bus	4F8
PD000:150	Private Data Lines - SELCH Bus	2F1 - 2F8 1S3 - 1S9
PDA0	Private Data Available Control Line to SELCH Bus	3H5
PHW0	Private Halfword Control Line from SELCH Bus	1A1
PSR0	Private Status Request Control Line to SELCH Bus	3H6
PSYN0	Private Sync from SELCH Bus	4B5
PTACK0	Private Transmit Acknowledge to SELCH Bus	4F2
RACK0	Receive Acknowledge from MPX-Bus	4B5
RBA0	Reset Buffer Active - Resets Buffer Active flip-flop	3M2
REQ0	Request - Request for memory cycle to Memory Bus	4R5
SCLR0	System Clear - Initialize Signal	4H4
SGAD1	Set Gate - Sets Address flip-flop	2LR
SR0	Status Request Control line from MPX-Bus	4B6
SREQ0	Set Request - Initiates a request for memory	3S4
SSX0	Set Status Transfer - Sets the Status Request flip-flop	3S5
SX	Status Transfer - Status Request flip-flop	3F6
SYN0	Sync to MPX-Bus	2S5
TAC0	Transmit Accept - To lower priority DMAs	4R1
TACK0	Transmit Acknowledge - To lower priority devices on MPX Bus	4F3
TAAR0	Toggle Auxiliary Address Register - Increments AAR	3M1
TAR0	Toggle Address Register - Increments AR	4S7
UAAH0	Unload Auxiliary Address Register High - Unloads AAR Bits 00:07	2R4
UAARL0	Unload Auxiliary Address Register Low - Unloads AAR Bits 08:15	2R4
WT	Write flip-flop	3F5
WRT0A	Write to Memory Bus, when selected	4R3

MEMORY PROTECT

[redacted]

MODEL 7/16 HSALU MEMORY PROTECT INSTALLATION SPECIFICATION

1. INTRODUCTION

This specification provides the necessary information for the installation of the 02-360 Memory Protect Module in a Model 7/16 HSALU Processor System. The module assembly consists of one 35-396 7" board and one 17-325 cable. The 35-396 7" board must be strapped to a blank 7" board or an active 7" board (e.g., Universal Clock Module) by an INTERDATA 16-398 Half Board Kit, so that it may be installed in a chassis designed for standard 15" boards. The Memory Protect Module must be installed in left half position to gain highest priority in the interrupt daisy chain. See Figure 1.

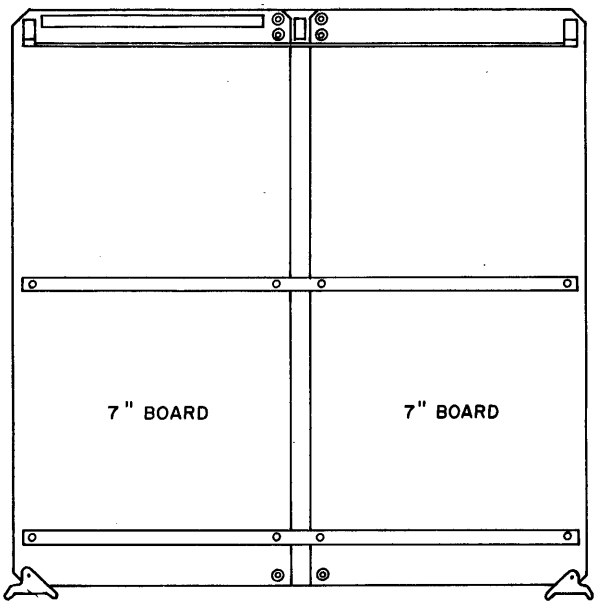


Figure 1. 7" Board Assembly

2. UNPACKING

When the Memory Protect Module is shipped with a system, it is installed at the factory so there is no special unpacking procedure. If the module assembly is purchased separately, it should be inspected for damage prior to installation.

3. LOCATION

It is required that the Memory Protect Module be installed as the highest priority device controller (with respect to I/O interrupts) on the Multiplexor Bus, to insure proper recognition and response to Memory Protect violation interrupts that may occur in the system. This is accomplished by installing the Memory Protect Module in the first available I/O slot on Connector 1.

4. CABLE CONNECTION

Install the 17-325 cable between the 35-396 Memory Protect board, the CPU-A board, and the CPU-C board. See Figure 2. The 35-524 CPU-C board is located in Slot 5 and the 35-522 CPU-A board is located in Slot 7 of the Processor back panel.

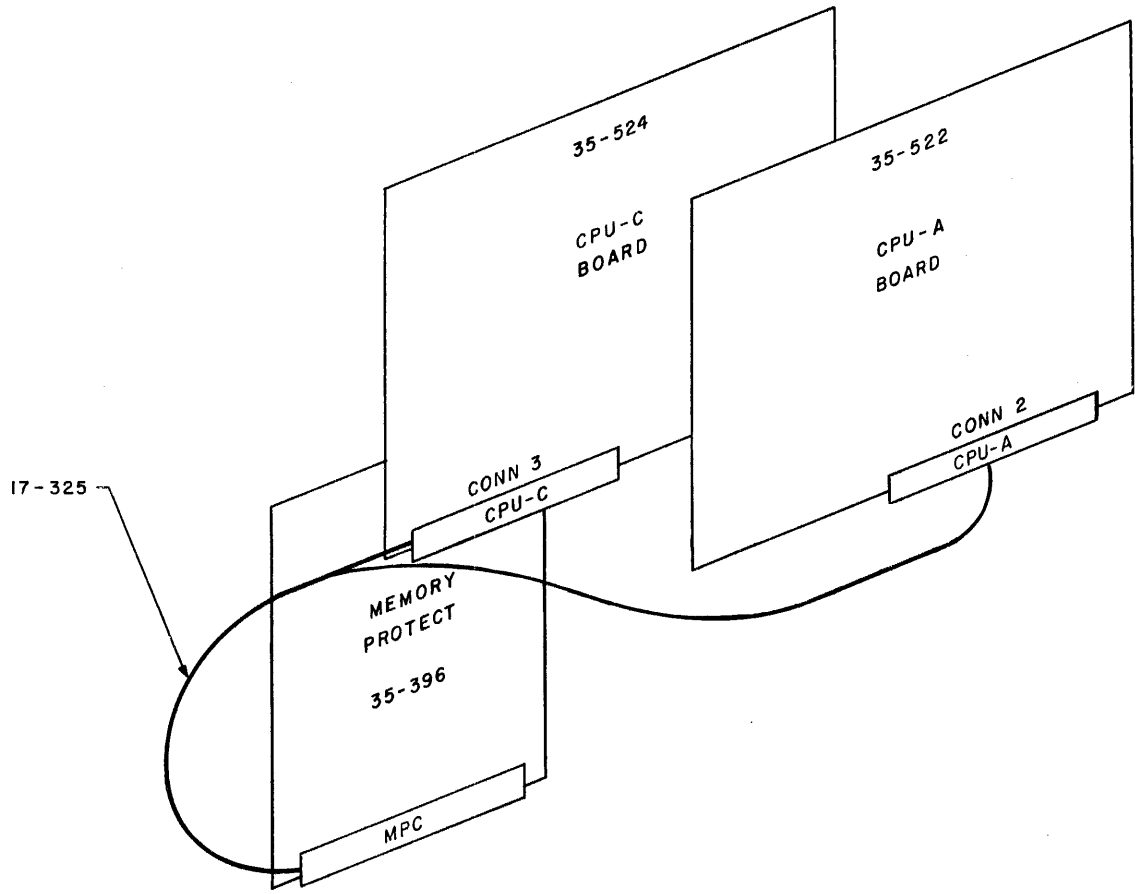


Figure 2. Cable Connection

5. STRAP OPTIONS

5.1 Address Strapping

The preferred address of the Memory Protect Controller is X'AE'. The module is strapped for this address at the factory. To change the address, refer to Functional Schematic 02-360D08, Sheet 1.

5.2 Block Size Option

The Memory Protect Controller is strapped for 1,024 Byte blocks at the factory. To change the block size, refer to the Block Size Table on Sheet 2 of Functional Schematic 02-360D08.

NOTE

The alpha-numeric designations referred to on the functional schematic, indicate similar designations on the apparatus side of the printed circuit board.

6. INSTALLATION CHECKS

To insure proper operation of the Model 7/16 HSALU Memory Protect, Test Program 06-135 should be executed in accordance with its test program description.

MODEL 7/16 HSALU MEMORY PROTECT MAINTENANCE SPECIFICATION

1. INTRODUCTION

The 02-360 Model 7/16 HSALU Memory Protect provides a means of allocating selected blocks of memory to be protected. The Memory Protect Controller (MPC) converts a Write operation to any location in a protected block to a Read operation, notifies the Processor via an interrupt, and sets a status bit in the controller. The memory may be partitioned into a maximum of 64 blocks with individual protect control for each block. An overall protection override provides a means of loading any area in memory.

The Model 7/16 HSALU Memory Protect consists of one Memory Control mother-board and one cable. The cable interconnects the Memory Protect Controller with the 35-524 CPU-C and the 35-522 CPU-A boards of the Processor. Refer to Installation Specification, 02-360A20, for necessary installation information.

2. SCOPE

This specification describes the functional operation of the Memory Protect Controller and its associated circuits in the Processor. This specification contains a block diagram analysis, a functional diagram analysis, timing information, and a mnemonic list for the Memory Protect Controller. Strapping information for block size selection is found on Sheet 2 of Functional Schematic 02-360D08.

3. BLOCK DIAGRAM ANALYSIS

Refer to the block diagram of the Memory Protect Controller (MPC) on Sheet 3 of Functional Schematic 02-360D08.

Prior to installation, the Memory Protect should be strapped for the desired block size. The following block sizes may be selected; 512 bytes, 1,024 bytes, or 2,048 bytes. Refer to the table on Sheet 2 of Functional Schematics 02-360D08 for strapping information.

Having selected the desired block size, the MPC is setup by the Processor via the Multiplexor Channel Bus. This setup includes the loading of the 64-bit Mask Register with the block or blocks to be protected, either selecting or overriding the protect function, and either arming or disarming the I/O interrupts at the controller. The loading of the 64 bit Mask Register is accomplished by steering up to eight Data Availables (DAs), by the Control Circuit, to the Mask Register. The most significant bit of the first DA corresponds to block one, the next less significant bit of the first DA corresponds to block two, etc.

The Memory Protect Controller constantly monitors the true outputs from the Memory Address Register in the Processor (MA001A:MA061A) and generates Protected Address (PRTADl) whenever a protected memory address, as defined by the 64-bit Mask Register, is accessed. Meanwhile, if protect is enabled at the Processor, and a Write operation is attempted, Protect (PRTECT0) is generated on the Memory Control board and sent to the MPC. PRTECT0 is ANDed with Protected Address (PRTAD1) and Override Protect (ORP1), by the MPC Control Circuit, to generate CWR0. CWR0 is in turn sent to the Read flip-flop on the Memory Control board. It sets the Read flip-flop which disables the Write attempt and causes a normal Read operation to be performed.

CWR0 also sets a status bit in the controller and generates an interrupt, when enabled, to indicate to the program that an attempt was made to Write to a protected area in memory.

4. FUNCTIONAL DIAGRAM ANALYSIS

4.1 Introduction

To understand INTERDATA functional schematics, it must be noted that the last character in the mnemonic symbol designates the logic level when the signal is active. For example; D080 is Data Line Number 8 and the last character 0 means that when D080 is active, the line is at a logical Zero level.

This section relates to Functional Schematic 02-360D08.

4.2 Multiplexor Bus Communication

When the Memory Protect Controller is addressed by the Processor, all highs are presented to the inputs of the NAND gate at 1G3. The output from this gate is inverted and applied to the J input of the Address flip-flop (AD)(1L3) while the non-inverted signal is applied to the K input of the Address flip-flop. The trailing edge of ADRS0 toggles the Address flip-flop set producing AD1.

The Processor is notified that the address was recognized by a device controller by activating SYN0 (1H7). SYN0 is generated by ADSYN0 (1D6), which is the ANDed output of ADRS1 and the recognition of the address. SYN0 is generated for each of the other control signals (CMD, DR, DA, and SR) by gating them with an active AD1 and applying the gated output to the NAND gate at 1E7.

If an interrupt is generated by this controller, the Attention flip-flop (1J9) is set, and Attention (ATN0) is sent to the Processor. When the Processor acknowledges this interrupt, Receive Acknowledge (RACK0)(1L8) becomes active. RACK0 is inverted and applied to the contention circuit (1R5) which generates either Attention Sync (ATSYN0), when ATN1 is active or Transmit Acknowledge (TACK0), when ATN1 is inactive. ATSYN1 (1H8) gates the address onto Data Lines 8:15, and its falling edge resets both the ATN and the Write Attempt (WATT) flip-flops. Interrupts may be enabled and disabled, via program control, by the setting or clearing of the ARM/DISARM flip-flop (1M7). Initialize (SCLR0 active) or an Output Command with Bit-8 active and Bit-9 inactive, sets this flip-flop to the Disarm state, and an Output Command with Bit-8 inactive and Bit-9 active, places it in the Arm state. In the Disarm state, a low is presented to the direct clear input of the ATN flip-flop, preventing it from becoming set.

4.3 Status and Command

The Status and Command Byte data is shown in Table 1.

TABLE 1. MEMORY PROTECT STATUS AND COMMAND BYTE DATA.

BIT NUMBER	0	1	2	3	4	5	6	7
STATUS BYTE			PON	PWF		EX		
COMMAND BYTE	DISARM	ARM	PON	POFF				

PON	Indicates that protect is enabled.
PWF	Indicates that an attempt was made to Write into a protected memory area. This bit is reset by an Output Command, or an Acknowledge Interrupt instruction.
EX	Examine is set when PWF is set.
DISARM	Disables interrupts. They will not be queued.
ARM	Enables interrupts.
PON	Enables the protect function on the controller.
POFF	Disables the protect function on the controller.

4.4 Load Mask Sequencer

The load mask sequencer steers up to eight Data Availables (DAs) to load the 64-bit Mask Register with the desired protect pattern. The four-bit counter (2L7) is initialized by CL1 which is the OR output from SCLR0 and CMG0 (1H9). When initialized, all outputs from the counter are at a logical Zero level. This causes the first DA to activate BADG20 which enables the loading of the two 4 x 4 register stacks located at 2G2 and 2G6. Since BAD01 and BAD11 are both low, the first DA loads Word '0' into the two registers. The trailing edge of DAG0 increments the counter, allowing the next DA to load Word '1' into the same two registers (BAD01 active). For the last four DA outputs, output C from the counter is high causing these DAs to activate BADG30 which enables the loading of the remaining two register stacks. If more than eight DAs are issued, wrap-around occurs (i.e., the ninth DA will once again load Word '0' into the first pair of registers).

4.5 Protect Pattern Recognition

The Memory Protect Controller constantly monitors the true outputs from the Memory Address Register (MAR) in the Processor (MA001A:MA061A). The MAR outputs are strapped in accordance with the table on Sheet 2 of Functional Schematic 02-236D08 for the desired block size, to generate Memory Address Lines MA01 to MA051 (2C3-267). The four most significant signals, MA01:MA031, are decoded by the four-to-four line decoder (2D2-2D9) to enable one of the four register stacks and to select one of the four-bit words of that register for a particular group of memory addresses.

Each of the four outputs from the four-bit register stacks is OR tied with the corresponding output from the other register stacks (i.e., the 1 output from register 35 is OR tied with the 1 output from register stacks 34, 33, and 32), but only the selected stack may be active. The two least significant memory address bits monitored by the Memory Protect Module for any given block size, MA41 and MA51, are used to select one of the four outputs of the selected register stack (2L3). This selection is accomplished by the four-to-one line multiplexor (2M8). If the selected data input to the multiplexor is high, the Protected Address signal (PRTAD1) is active.

Each time a Memory Write function is decoded by the Processor with Program Status Word (PSW) Bit-7 set, PRTECT0 is generated unless the Write function is a privileged write. The one-shot at 2K5 generates a pulse approximately 60 nanoseconds in duration on the trailing edge of PRTECT0. This pulse, when the protect function is enabled in the module, ORP1 and PRTAD1 active, generates CWR0 and ASATN0. These signals (CWR0 and ASATN0) cause the Write attempt by the Processor to be converted into a Read operation, and the Attention flip-flop in the Memory Protect Module to be set respectively.

5. TIMING

Refer to Figure 1. Worst case delays are indicated.

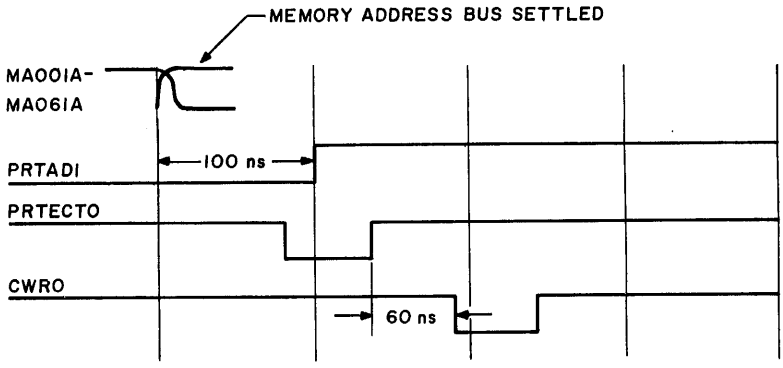


Figure 1. Memory Protect Timing

6. MAINTENANCE AND TESTS

The Memory Protect Module requires no adjustments. Before attempting any maintenance or testing, insure that the cable and the controller are installed properly. Refer to the Installation Specification, 02-360A20, for necessary installation information.

To test the Memory Protect Module, run the Memory Protect Test 06-135 in accordance with its test program description.

7. MNEMONICS

The following list provides a brief description of each mnemonic in the Memory Protect Module. The source of each signal on Schematic Drawing 02-360D08 is also provided.

MNEMONIC	MEANING	SCHEMATIC LOCATION
AD	Address flip-flop	1M3
ADRS0	Address control line from CPU	1J4
ADSYN0	Address Sync - Causes Sync to be returned on an ADRS0	1M4
ASATN0	Set Attention - Sets the Attention flip-flop on a protect violation	2R4
ATN	Attention flip-flop	1J8
ATSYN0	Attention Sync - Causes Sync to be returned on an Acknowledge Interrupt	1S5
BAD01-11	Block Address - Selects the indicated word in the 64-bit Mask Register	2M6
BADG20-30	Block Address Gated - Enables the indicated pair of registers in the 64-bit Mask Register	2S6
CL1	Clear - Initialize the load mask sequencer on a Command or System Clear	1K9
CMD0	Command control line from the CPU	1A6
CWR0	Convert Write to Read - To CPU to change the Write operation into a Read operation	2S4
D080:D150	Data Lines from the CPU	1A1 - 1A5
DA0	Data Available control line from the CPU	1A7
DR0	Data Request control line from the CPU	1A7
MA001A:061A	Ungated Memory Address from the Memory Address Register	2A3 - 2A9
MA01:051	Memory Address lines selected by the Block Size strapping	2C4 - 2C7
ORP1	Override Protect - Overrides the protect function of the Memory Protect Module	1L1
PRTAD1	Protected Address - Indicates that the Memory Address Register in the CPU contains a protected address when active.	2N3
PRTECT0	Protect - Generated by the Processor when a non-privileged Write attempt is made	2J5
RACK0	Receive Acknowledge - Acknowledge from the lower priority device in the RACK0/TACK0 daisy chain	1L8
SCLR0	System Clear from the CPU	1A8
SR0	Status Request control line from the CPU	1A8
SYN0	Sync - Composite Sync signal to the CPU	1G7
TACK0	Transmit Acknowledge - Acknowledge to the higher priority device in the RACK0/TACK0 daisy chain	1S6

5.14 Turnkey Console (Figure 20)

Refer to Functional Schematic 02-352C08. The Basic Switch Control Panel provides a means by which a program, previously loaded into memory, can be executed without the aid of the optional Control Console.

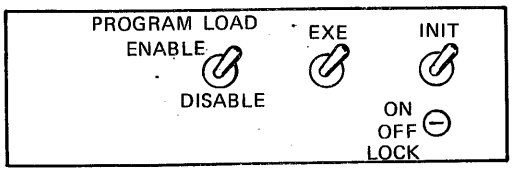


Figure 20. Turnkey Console

This panel provides a means of controlling the system power, initializing the system, and generating a Console Attention (FCATN1) to start program execution, if the Primary Power Fail/Auto Restart option is not installed.

This option conditions the Processor to the Run mode by grounding SSGL1, SD011, SD021, and SD031 at the Control Console connector. The status of the display is X'8F'. If a data byte is read from the display it is X'8F'. The Display Controller, when addressed by the micro-program in the power up sequence, indicates the Run mode. With the Auto-Restart option present, program execution commences at the address specified in the Location Counter (LOC), when the system is turned on and without the Auto-Restart option the micro-program performs a normal power sequence and then goes to the un-interruptable idle Loop until the Execution (EXE) switch is operated. When the EXE switch is operated, program execution then begins as described previously.

6. MAINTENANCE

This section describes maintenance procedures which may be used to check and, if necessary, adjust the Processor.

6.1 Clock Timing

There is only one adjustment associated with the Processor clocks. The variable Capacitor C89 on the CPU-A board 35-522 is very stable and should not require field adjustment. The adjustment should only be changed after the test indicates that it is out of tolerance and there are no faulty components in the system.

Clock timing is checked with an oscilloscope. By monitoring CLK1 B at the Processor back panel, Slot 7, Pin 202-1 of the CPU-A board. The period of CCK1B should be 250 nanoseconds. Refer to Figure 21. Adjust Capacitor C89 to obtain the 250 nanosecond period.

6.2 Clock Timing for 35-522 M01, CPU A

There is only one adjustment associated with the Processor clocks. The variable Capacitor C1 on the CPU-A board 35-522 is very stable and should not require field adjustment. The adjustment should only be changed after the test indicates that it is out of tolerance and there are no faulty components in the system.

Clock timing is checked with an oscilloscope by monitoring CLK1 B at the Processor back panel, Slot 7, Pin 202-1 of the CPU-A board. The period of CLK1B should be 250 nanoseconds. Refer to Figure 21. Adjust Capacitor C1 to obtain the 250 nanosecond period.

6.3 Fast Memory Timing Adjustment

The fast memory timing adjustment is set up at the factory and should not require field need adjustments. There are two adjustments for fast memory timing: (1) adjust the time between the falling edge of the ER0 and the falling edge of INH0. (2) adjust the width of INH0. Potentiometer R57 adjusts the start time of INH0. Potentiometer R58 adjusts the width of INH0. Refer to Figure 20. Grounding REQ0 at the Processor back panel Slot 7, Pin 139-0 of the CPU-A board causes constant memory cycles to occur. Before grounding REQ0, remove MAC or extended DMA buffer if installed. ER0 may be found at Slot 7, Pin 204-0 of CPU-A. INH0 may be found at Slot 7, Pin 104-0 of CPU-A.

MODEL 7/16 HSALU MEMORY PROTECT PROGRAMMING SPECIFICATION

1. INTRODUCTION

The 02-360 Model 7/16 HSALU Memory Protect provides a means of allocating selected blocks of memory to be protected from user writing. The Memory Protect Controller (MPC) converts a Write operation to any location in a protected block to a Read operation, notifies the Processor via an interrupt, and sets a status bit in the controller. The memory may be partitioned into a maximum of 64 blocks with individual protect control for each block. Block sizes of 512 bytes, 1,024 bytes, or 2,048 bytes may be selected by strap options on the controller. Refer to Installation Specification 02-360A20, for block size strapping information. An overall protection override provides a means of disabling the protect function at the controller. Note that the protect function is enabled at the Processor only when Bit 7 of the current Program Status Word is set. When Bit 7 is reset, all Writes are treated the same as Privileged Writes (i.e., protect disabled).

2. CONFIGURATION

The Model 7/16 HSALU Memory Protect consists of one Memory Control mother-board and one cable. The cable interconnects the Memory Protect Controller with the Memory Control Board in the Central Processor Unit (CPU). The priority of the Memory Protect Controller on the Multiplexor Bus is determined by its installation in the system chassis. Refer to 02-360A20 for installation information.

> NOTE
>
> It is imperative that the Memory Protect Controller have the highest priority on the Multiplexor Bus, to avoid ambiguity in identifying the source of a protect violation when executing interrupt routines with I/O interrupts enabled.

3. OPERATING PROCEDURES

The Model 7/16 HSALU Memory Protect is controlled by programmed I/O sequences using the Multiplexor Bus. See Sections 5 and 6.

4. DATA FORMAT

Refer to Appendix 1 for the definition of the Load Mask Bytes.

5. PROGRAMMING INSTRUCTIONS

The Output Command instruction (OC or OCR) causes a command byte, as defined in Table 1, to be sent to the controller. Any command causes all status except Protect On (PON) to be reset, and the Load Mask Sequencer to be initialized. The Sense Status instruction (SS or SSR) is used to read the status byte, as defined in Table 1, from the Memory Protect Controller. The status byte reflects the operational state of the controller. The least significant four bits (4:7) of the status byte are copied into the Condition Code of the current Program Status Word so they can be tested directly by the use of Branch instructions.

The Write Data (WD or WDR), Write Halfword (WH or WHR), or the Write Block (WB or WBR) instructions may be used to load the protect pattern into the controller.

TABLE 1. MEMORY PROTECT STATUS AND COMMAND BYTE DATA

BIT NUMBER	0	1	2	3	4	5	6	7
STATUS BYTE			PON	PWF		EX		
COMMAND BYTE	DISARM	ARM	PON	POFF				

STATUS

PON	Indicates that protect is enabled.
PWF	Indicates that an attempt was made to Write into a protected memory area. This bit is reset by an Output Command, or an Acknowledge Interrupt instruction.
EX	Examine is set when PWF is set.

COMMAND

DISARM	Disables interrupts. They will not be queued.
ARM	Enables interrupts.
PON	Enables the protect function at the controller.
POFF	Disables the protect function at the controller.

If an interrupt is pending, the Acknowledge Interrupt instruction (AI or AIR) clears the interrupt and causes the interrupting device address and status to be read into the Processor. Executing an Acknowledge Interrupt instruction when no interrupt is pending, results in a device address of zero and a status of X'04'.

6. PROGRAMMING SEQUENCE

The setting up of the Memory Protect Controller consists of loading the desired protect pattern into the controller, enabling the protect function, and either Arming or Disarming the interrupts. Any Output Command initializes the load sequencer in the controller so that the first byte to the protect module, following the Output Command, will load the Mask Register, Blocks 0:7. Refer to Appendix 1 for the correspondence between the load mask bytes, block size, and memory addresses protected. An active bit in the load mask byte masks (protects) the selected addresses. If more than eight bytes are used to load the Mask Register, wrap-around occurs (i.e. the ninth byte once again loads blocks 0:7). For the 2K byte block size option only four bytes are required to protect maximum memory, 64K. If more than four bytes are issued, bytes 4:7 load the Mask Register but are not used. The Protect Enable Command and Arm or Disarm Command may be issued simultaneously.

After setting up the Protect Controller, Bit-7 of the Program Status Word must be set to enable the protect function at the Processor.

7. INTERRUPTS

An interrupt can be used to signal the Processor that an attempt was made to Write into a protected area in memory. In the Arm state, interrupts are armed and enabled. In the Disarm state, interrupts are neither enabled not queued.

When an interrupt is acknowledged by an Acknowledge Interrupt instruction (AI or AIR) the interrupt is cleared and all status bits except PON are reset. The normal status returned by an Acknowledge Interrupt is X'20'.

8. INITIALIZATION

During power up, power down, or whenever the Initialize (INT) switch on the Display Panel is depressed, the Memory Protect Controller is placed in the Disarm state with all status conditions reset. The Load Mask Sequencer is initialized and the protect function is disabled at the controller.

9. DEVICE NUMBER

The Model 7/16 HSALU Memory Protect is normally assigned device address X'AE'. This address can be changed by the modification of straps on the controller. Refer to the Installation Specification, 02-360A20, for details about the wiring alteration.

10. SAMPLE PROGRAM(S)

Not applicable to the Model 7/16 HSALU Memory Protect.

APPENDIX 1
TABLE OF MEMORY ADDRESSES VS. BLOCKS

BLOCK	LOAD MASK DATA		MEMORY ADDRESSES (HEX)		
	BYTE	BIT	.5K BYTE	1K BYTE	2K BYTE
1	0	0	0000-01FF	0000-03FF	0000-07FF
2	0	1	0200-03FF	0400-07FF	0800-0FFF
3	0	2	0400-05FF	0800-0BFF	1000-17FF
4	0	3	0600-07FF	0C00-0FFF	1800-1FFF
5	0	4	0800-09FF	1000-13FF	2000-27FF
6	0	5	0A00-0BFF	1400-17FF	2800-2FFF
7	0	6	0C00-0DFF	1800-1BFF	3000-37FF
8	0	7	0E00-0FFF	1C00-1FFF	3800-3FFF
9	1	0	1000-11FF	2000-23FF	4000-47FF
10	1	1	1200-13FF	2400-27FF	4800-4FFF
11	1	2	1400-15FF	2800-2BFF	5000-57FF
12	1	3	1600-17FF	2C00-2FFF	5800-5FFF
13	1	4	1800-19FF	3000-33FF	6000-67FF
14	1	5	1A00-1BFF	3400-37FF	6800-6FFF
15	1	6	1C00-1DFF	3800-3BFF	7000-77FF
16	1	7	1E00-1FFF	3C00-3FFF	7800-7FFF
17	2	0	2000-21FF	4000-43FF	8000-87FF
18	2	1	2200-23FF	4400-47FF	8800-8FFF
19	2	2	2400-25FF	4800-4BFF	9000-97FF
20	2	3	2600-27FF	4C00-4FFF	9800-9FFF
21	2	4	2800-29FF	5000-53FF	A000-A7FF
22	2	5	2A00-2BFF	5400-57FF	A800-AFFF
23	2	6	2C00-2DFF	5800-5BFF	B000-B7FF
24	2	7	2E00-2FFF	5C00-5FFF	B800-BFFF
25	3	0	3000-31FF	6000-63FF	C000-C7FF
26	3	1	3200-33FF	6400-67FF	C800-CFFF
27	3	2	3400-35FF	6800-6BFF	D000-D7FF
28	3	3	3600-37FF	6C00-6FFF	D800-DFFF
29	3	4	3800-39FF	7000-73FF	E000-E7FF
30	3	5	3A00-3BFF	7400-77FF	E800-EFFF
31	3	6	3C00-3DFF	7800-7BFF	F000-F7FF
32	3	7	3E00-3FFF	7C00-7FFF	F800-FFFF
33	4	0	*	8000-83FF	↑
34	4	1		8400-87FF	|
35	4	2		8800-8BFF	|
36	4	3		8C00-8FFF	|
37	4	4		9000-93FF	|
38	4	5		9400-97FF	|
39	4	6		9800-9BFF	|
40	4	7		9C00-9FFF	N/A
41	5	0		A000-A3FF	|
42	5	1		A400-A7FF	|
43	5	2		A800-ABFF	|
44	5	3		AC00-AFFF	|
45	5	4		B000-B3FF	|
46	5	5		B400-B7FF	|
47	5	6		B800-BBFF	|
48	5	7		BC00-BFFF	↓

* If more than 32K Bytes of memory exists, and the 512 Byte Block option is used, the pattern defined for the first 32K repeats itself for the second 32K.

APPENDIX 1
TABLE OF MEMORY ADDRESSES VS. BLOCKS (Continued)

BLOCK	LOAD MASK DATA		MEMORY ADDRESSES (HEX)		
	BYTE	BIT	.5K BYTE	1K BYTE	2K BYTE
49	6	0		C000-C3FF	N/A
50	6	1		C400-C7FF	
51	6	2		C800-CBFF	
52	6	3		CC00-CFFF	
53	6	4		D000-D3FF	
54	6	5		D400-D7FF	
55	6	6		D800-DBFF	
56	6	7		DC00-DFFF	
57	7	0		E000-E3FF	
58	7	1		E400-E7FF	
59	7	2		E800-EBFF	
60	7	3		EC00-EFFF	
61	7	4		F000-F3FF	
62	7	5		F400-F7FF	
63	7	6		F800-FBFF	
64	7	7		FC00-FFFF	

TEST AID [redacted]

02-276R03A12
February 1976

M49-410
TEST AID
INFORMATION SPECIFICATION

1. INTRODUCTION

This Information Specification covers installation, operation, and maintenance of the M49-410 Test Aid (02-276) and the associated logic in the Processor. Refer to 02-276D08 for schematics of the M49-410 Test Aid.

2. GENERAL DESCRIPTION

The Test Aid consists of a switch display panel and a 17-283 logic card which attaches to the following boards:

35-446 CPU-HI Model 74 Processor
35-446F01 and 35-446F02 CPU-HI 7/16 Basic Processor
35-524 CPU-B Model 7/16 HSALU Processor
35-523F01 and 35-523F02 CPU-B Model 7/32
35-624F01, F02, F03 CPU-B Model 7/32C Processor

The Test Aid provides the ability to examine the address of the micro-code and to stop Processor clocks at option.

3. INSTALLATION

This section provides the information necessary to install the Test Aid on the Processor. The 02-276R01 or higher revision level Test Aid may be used on the Model 74, 7/16 Basic, 7/16 HSALU, 7/32, and 7/32 C Processors. The 02-276R00 Test Aid may be used on the Model 74 and the Model 7/16 Basic. The installation procedure is:

1. Remove the display from the chassis.

2. Place Test Aid logic card over pins on CPU-HI board or CPU-B board (installed in Slot 6 of the Processor back panel, refer to Figure 1) and press down until Test Aid logic card rests on spacers on the Processor board. The switch/display panel assembly may be placed on a table or mounted on the chassis as shown in Figure 2.

3. On the 7/16 HSALU Processor, a jumper must be installed between TP1 on the 35-544 CPU-B board and TP2 on the 35-522 CPU-A board.

4. On the 7/32, a jumper must be installed between TP1 on the 35-523F01 or F02 CPU-B board and TP2 on the 35-522 CPU-A board.

5. On the 7/32 C Processor, a jumper must be installed between TP1 on the 35-625F01, F02, or F03 CPU-B board and TP2 on the 35-624 CPU-A board.

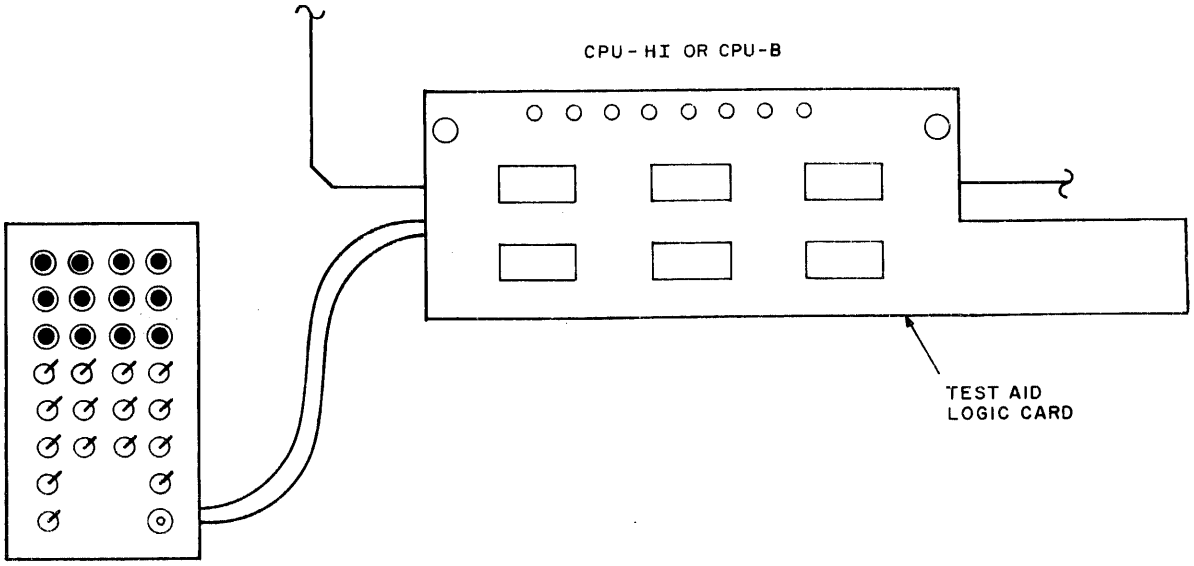


Figure 1. Test Aid Installation

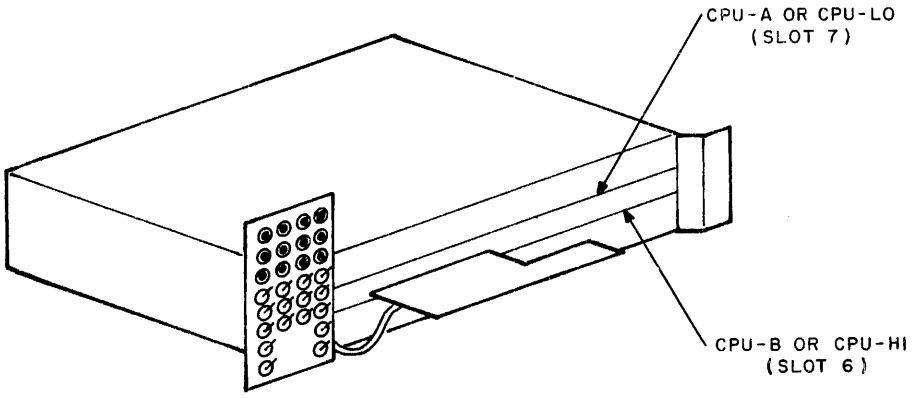


Figure 2. Switch Panel Mounting

4. POWER

Power and ground are supplied by the Processor board. There are no other power requirements.

5. OPERATION

Refer to Figure 3 during the operating description. The 12 Light-Emitting Diodes (LEDs) numbered 4:15 display the contents of the ROM Address Register. The numbers assigned to the LEDs correspond to the ROM Address Register bits. The 12 toggle switches labelled 4:15 provide the ability to set-up a match address. The Test Aid logic stops the Processor clocks when the selected "match address" is in the ROM Address Register and the Address Match switch is in the ON (up) position.

6. ADDRESS MATCH SWITCH

After selecting an address on the Address switches, place the Address Match switch in the ON (up) position. When the ROM Address Register of the Processor contains the "match address", the Processor clocks are stopped on the next clock. The Address Match switch feature can also be used to interrupt and continue micro-code loops. Follow the procedure for address matching. Select an address within a micro-code loop. Once the match has been found, depressing the Clock Advance (ADV) switch once allows the micro-code to go through the loop and match on the selected address again.

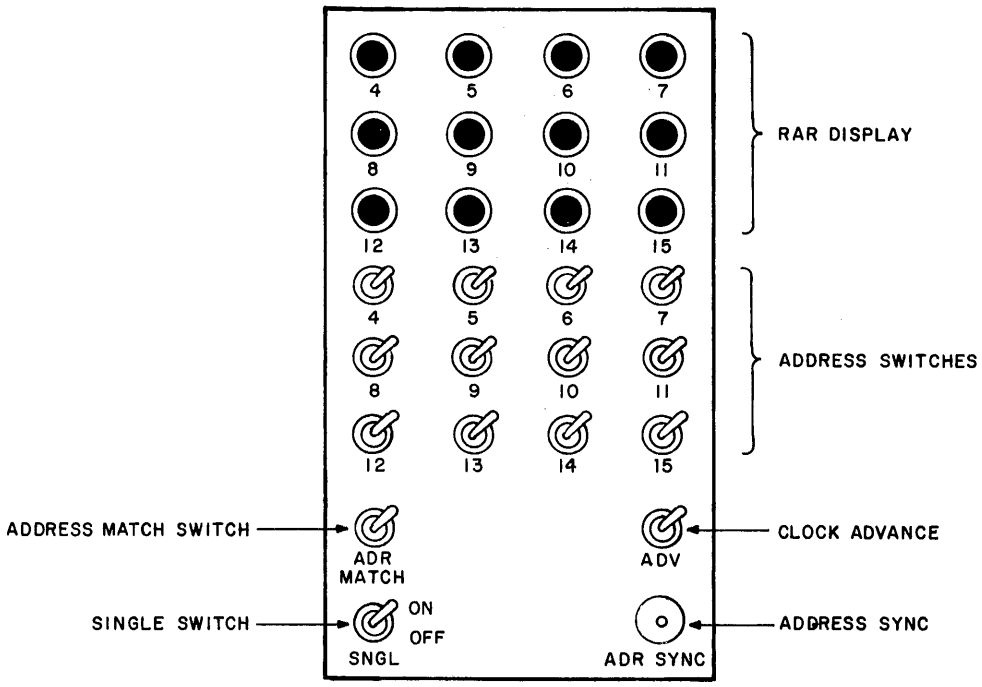


Figure 3. Switch Panel

NOTE

The LED display in most cases is one increment ahead of the match address. The micro-code instruction at the address selected has been executed or is one clock into execution when the match occurs and the Processor clocks stop.

7. CLOCK ADVANCE SWITCH

The Clock Advance switch allows the Processor to generate one clock each time it is depressed when the Address Match or Single switch is in the ON (up) position.

8. SINGLE SWITCH

When the Single switch is in the ON (up) position, the Processor clocks are stopped. With this switch ON, the micro-program may be executed one micro-instruction at a time by depressing the Clock Advance switch.

9. ADDRESS SYNC

Address Sync is a BNC connector whose output is a low going signal that becomes active when the Address switches and the contents of the ROM Address Register compare. The contents of the ROM (specified by the ROM Address Register) will not be loaded into the ROM Data Register until the next Clock which loads the ROM Data Register.

10. OPTION

Pins 'A' and 'B' are normally wired together. Pin 'A' is the output of a comparator that compares the ROM Address Reg ister and the Address switches. When they compare, the signal on Pin A goes high (+5 VDC) causing Processor clocks to stop. Removing the wire between Pins 'A' and 'B' provides a means to bring in any high active signal on Pin 'B' to stop Processor clocks. To match on any high active signal, the Address Match switch must still be placed in the ON (up) position when a match is desired. Removing the wire between 'A' and 'B' will remove the capability to stop Processor clocks on address match. See Figure 4.

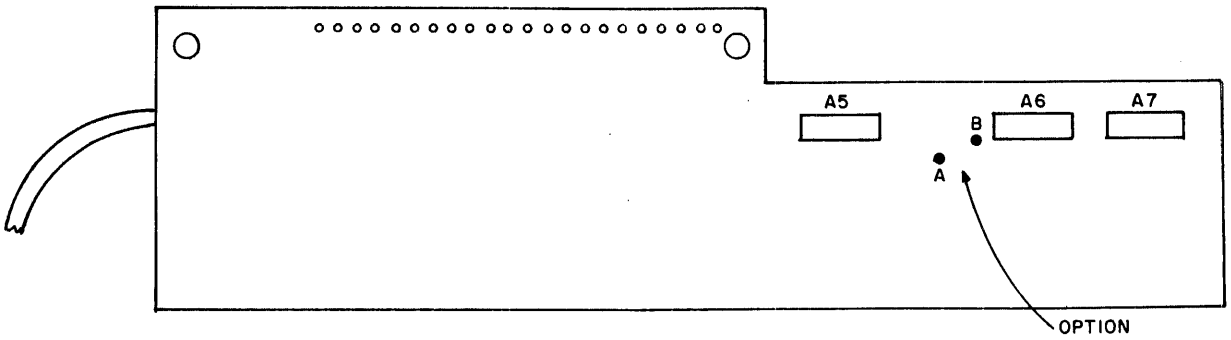
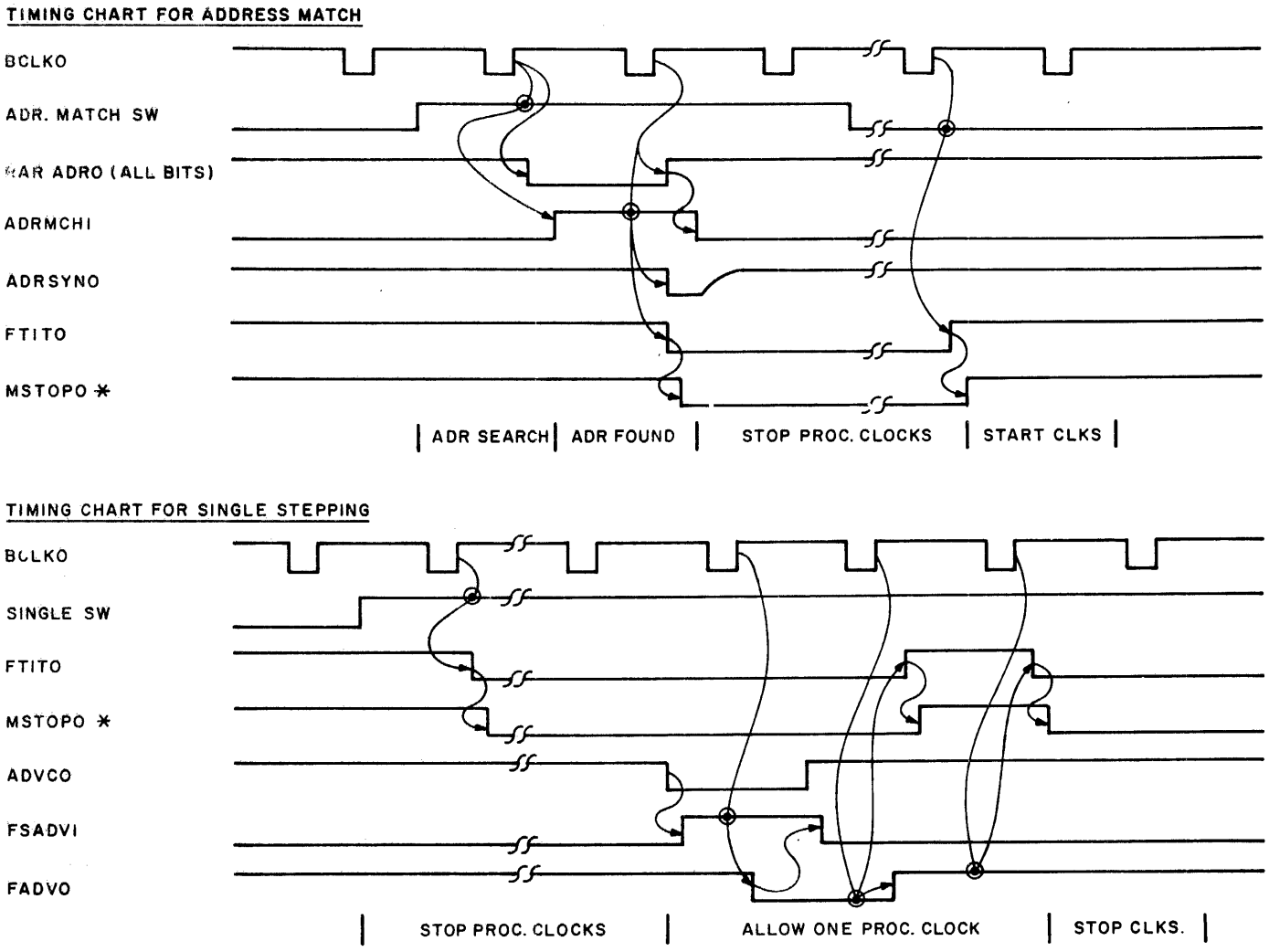


Figure 4. Option Connections

11. TEST AID MAINTENANCE

11.1 Timing

This section defines timing sequences (Figure 5) in the logic of the Test Aid and associated logic in the Processor. Refer to the Processor Functional Schematic, Clock Control sheet, for logic detail of the clock stop.



* MSTOPO IS A CLOCK STOPPING SIGNAL INTERNAL TO THE MODEL PROCESSOR. WHEN ACTIVE ALL PROCESSOR CLOCKS EXCEPT CLKI, BCLKI AND BCLKO ARE STOPPED.

Figure 5. Test Aid Timing

11.2 Mnemonic Definitions

ADRMCH1 - This signal is active when the contents of the ROM Address Register and the Address switches are equal.

ADVC0 - Flip-flop output which goes active when the ADV switch is depressed, inactive when the ADV switch is released.

BCLK0 - Derived from the Processor. This is a clock that cannot be stopped by any clock stop in the Processor. BCLK0 width is typically 60 nanoseconds and the period is typically 250 nanoseconds.

FADV0 - When active, allows FTIT0 to be inactive for one clock period. If ADVC0 and BCLK0 are active at the same time, the FADV0 flip-flop sets.

FTIT0 - This flip-flop is reset by Single switch ON, Address Match switch ON, and a match address.

MCH04-150 - When active, indicates that a particular address switch has been selected.

RAR04-150 - ROM Address Register outputs which indicates the address of the micro-instruction to be executed on the next clock.

12. USE OF MODEL 70 EXTENDER BOARD (11-103) ON THE PROCESSOR

12.1 Hazards

All Model 70 extender boards, below revision level 11-103R02, when used to extend Processor boards, present two hazards.

1. All stiffening metal on the extender board when being plugged in becomes +5VDC. This hazard exists with either Processor board on the extender.

2. When the Test Aid is installed and the CPU-LO or CPU-A is on the extender board, a stiffening bar located on the underside of the extender board rests on top of the Test Aid logic card and forces it down possibly causing a short.

12.2 Modification

The following information describes how to modify the 11-103R01 extender board:

1. Pins 200-0, 200-1, 241-0 and 241-1 are tied into the ground bus of the extender board. These pins in the Processor are +5VDC. Both ends of the extender board tie these pins to the extender board ground but via feedthrough holes causing the ground bus to become +5VDC. Cut the copper between these feed-through holes and the extender board ground bus. Add a strap from the copper run of Pins 101-0, 101-1, 140-0 and 140-1 to the adjacent ground shield to restore the continuity of back panel ground to extender board ground.

2. Remove stiffening bar on underside of extender board. Three new clearance holes must be drilled so that the stiffening bar mounts horizontally rather than vertically. The original screws may bottom out; if so, use #4-40 x 5/8 screws. Refer to Figure 6.

   After this change is made, care should still be taken to insure that the Test Aid logic card is not shorting to the stiffening bar.

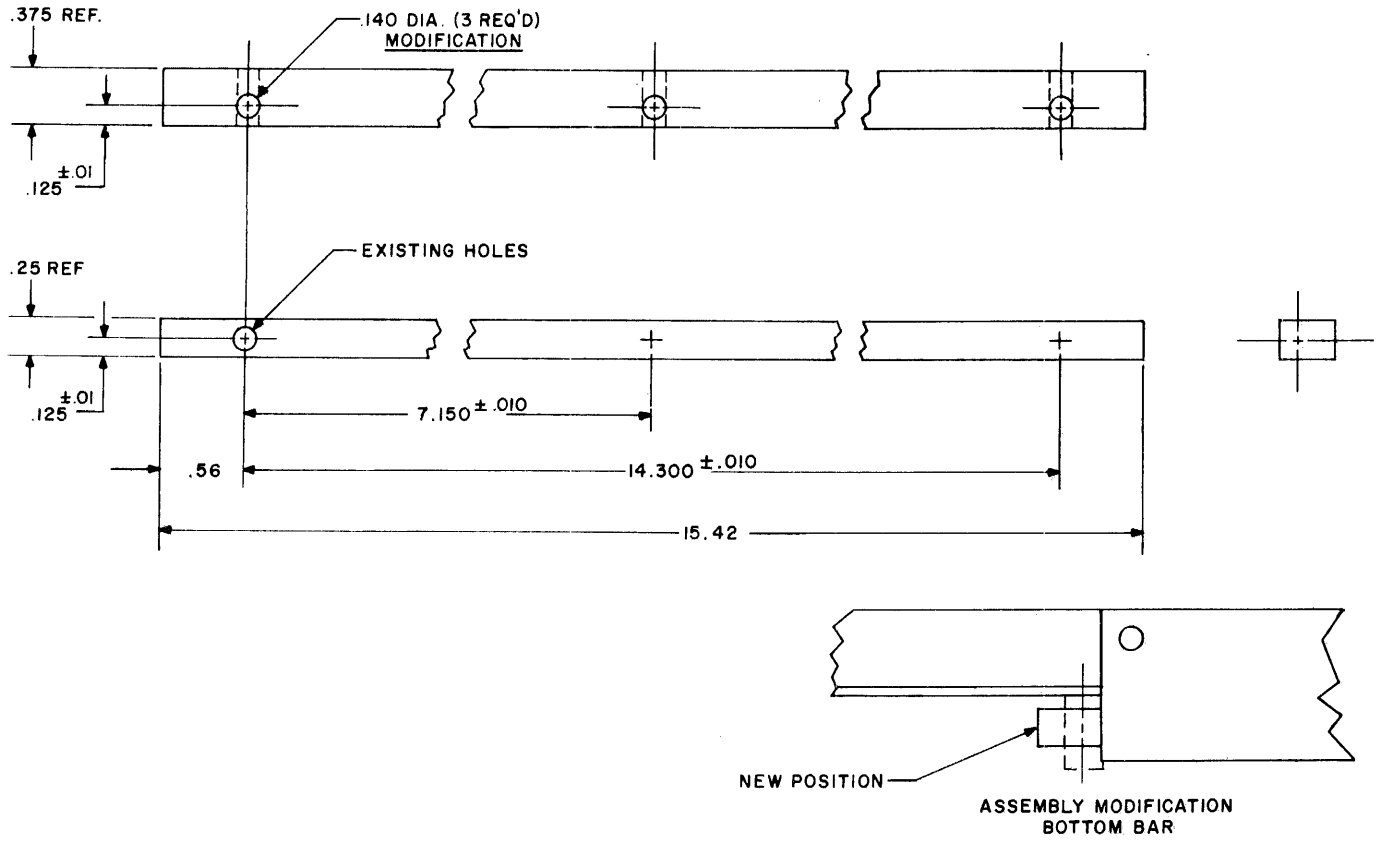


Figure 6. Stiffening Bar

AUTO LOADER

[redacted]

M71-103
AUTOMATIC LOADER
INFORMATION SPECIFICATION

1. INTRODUCTION

The Automatic Loader consists of a logic card connector, cable, and switch panel which connects onto the Processor's display connector in place of a display. The Automatic Loader provides a means of applying power to the Processor via a LOCK/ON/OFF switch, initializing the Processor (INIT), and starting a resident program (EXE). The micro-program in the Processor fetches a device address and a command byte from the Automatic Loader on its power up sequence, when enabled. This information is used to load a program from the specified device.

This information specification covers installation and operation of the Autmoatic Loader.

2. INSTALLATION

2.1 Prior to Installation

The Automatic Loader must be modified to specify the appropriate device number and command byte. The Automatic Loader is initially equipped with a device number of X'FF' and a command of X'FF'. Refer to Functional Schematic 02-352C08 for the information necessary to change the device and command designations.

2.2 Model 7/16 Basic Installation

The Automatic Loader is installed on CPU-LO Board Connector 3, and to the 25-327 Terminal Strip, Positions C1 and C2. Refer to Figure 1.

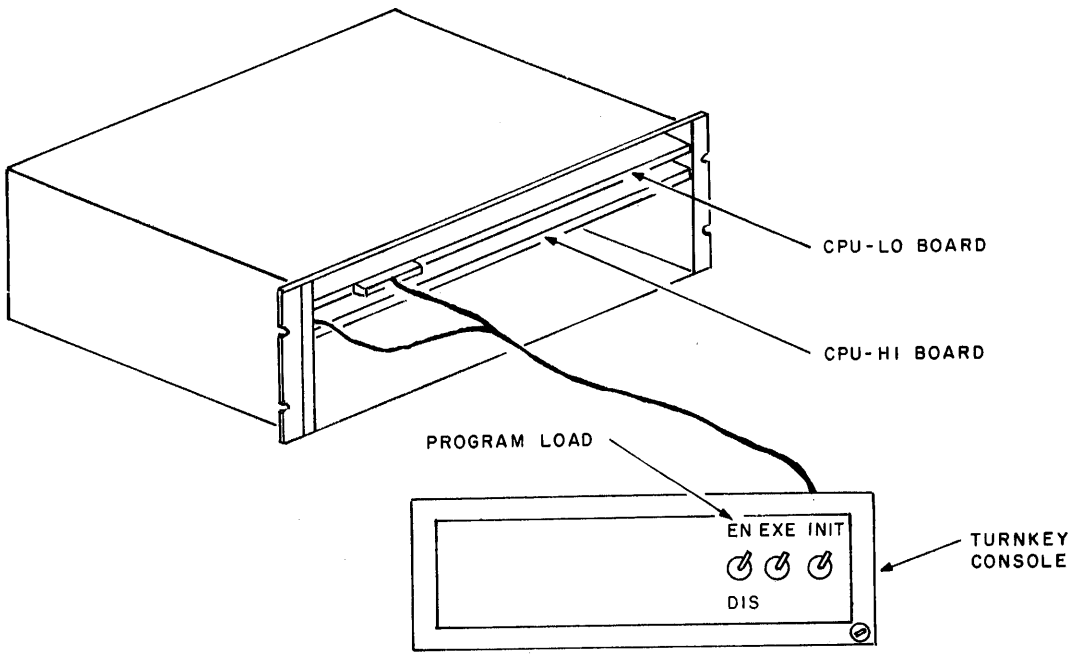


Figure 1. 7/16 Basic Installation

2.3 Model 7/16 HSALU Installation

The Automatic Loader is installed on the 35-522 CPU-A board, and to the 25-327 Terminal Strip, Positions C1 and C2. Refer to Figure 2.

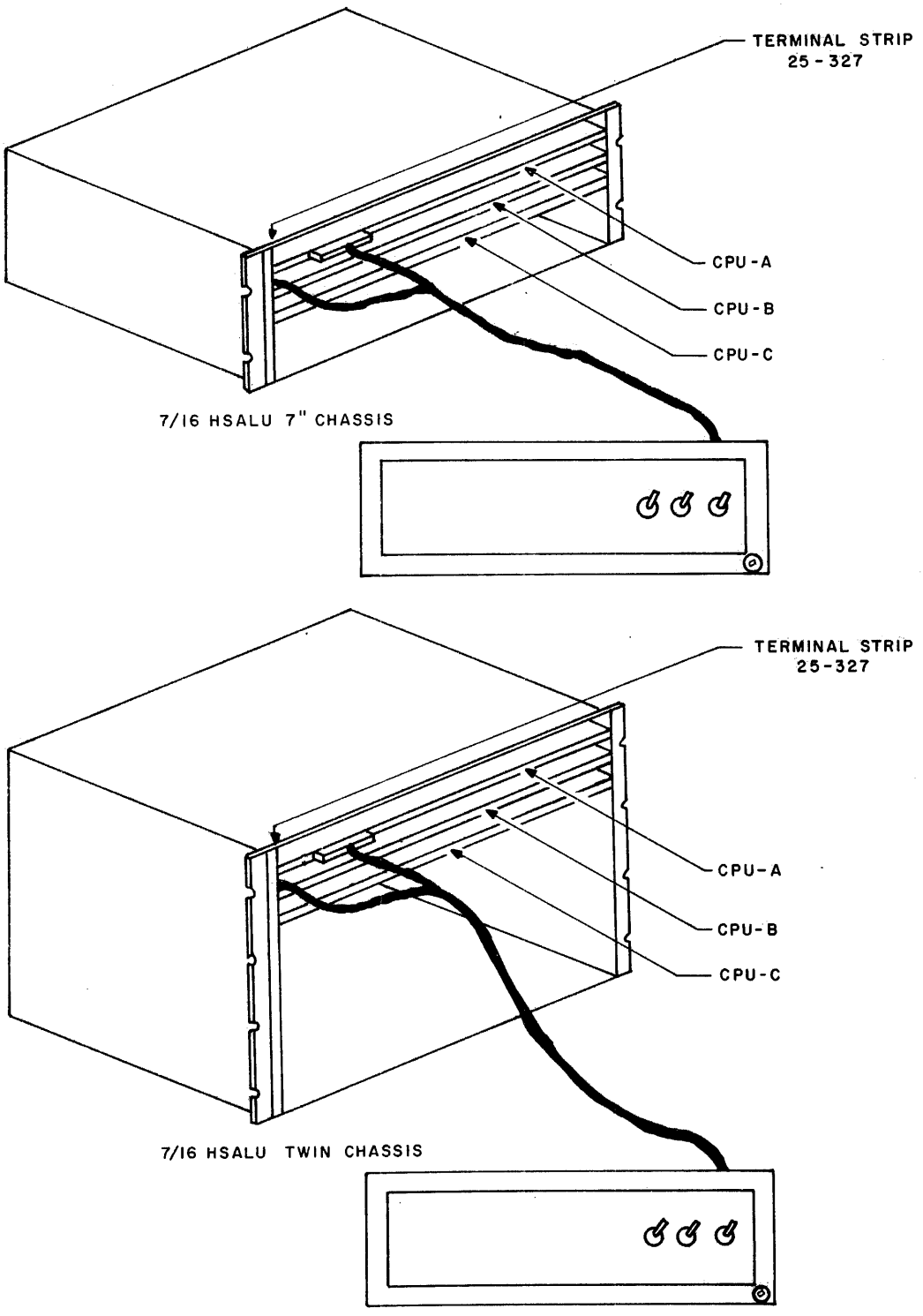


Figure 2. 7/16 HSALU Installation

3. OPERATION

Refer to Figure 3 for the following controls:

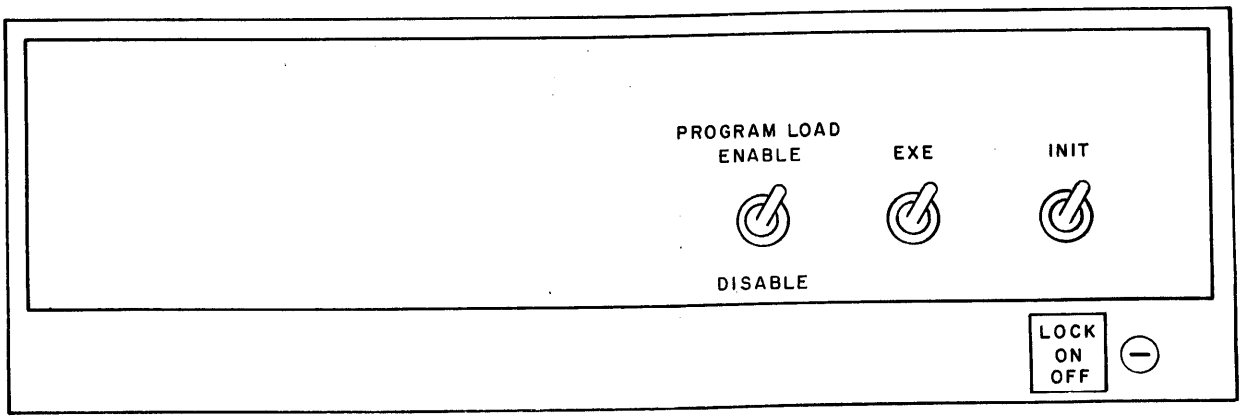


Figure 3. Switch Features

OFF/ON/LOCK — Power switch. This is a three position key operated security lock switch which controls primary power to the system. When placed in the OFF position, the testable Primary Power Fail signal (PPF) goes active which causes a power fail indication and removes the DC voltages to the system. In the ON position, DC power is applied to the Processor. In the LOCK position, the power remains on but the EXE and INIT switches are disabled.

INIT — Initialize switch. The Initialize switch generates the testable PPF signal. The micro-program responds to PPF by saving the Processor status in main memory and then doing a Command Power Down to initialize the system.

EXE — Execute switch. The Execute switch generates the testable Console Attention signal (CATN0). The micro-program always assumes the Run mode in response to the Automatic Loader CATN0 (EXE), except when the Program Load switch is in the Enable position.

PROGRAM LOAD — Program Load switch. The Program Load switch determines whether or not an auto-load is to be performed after the system is initialized or upon power up. If the Program Load switch is in the Enable position, upon initialization or power up, an automatic load is performed from the device number specified by the Automatic Loader logic card.

The micro-program determines that the Automatic Loader option has been selected by the unique status of the display controller; namely a zero in Bit 4 of the status and a one in Bit 7. The state of the Program Load switch is presented to the Processor as Bit 6 of the display status. On power up or initialize, the micro-program determines that the Automatic Loader option is present, then does two data requests from the device number '01' (Automatic Loader). The first data byte received is the device number from which an automatic load is to be performed and the second data byte is an appropriate output command for that device. The device number and command byte are a function of the Automatic Loader logic card.

NOTE

The type device used must be able to respond with data with one Output command. This device must also have a valid status (busy only or all zeros) immediately after the Output command. The first data byte read must be valid data.

Figure 4 shows the data format for automatic loading. This data may be appended to a system program or may be a stand-alone program used as a general purpose boot loader. Appendix 1 is an example of the latter.

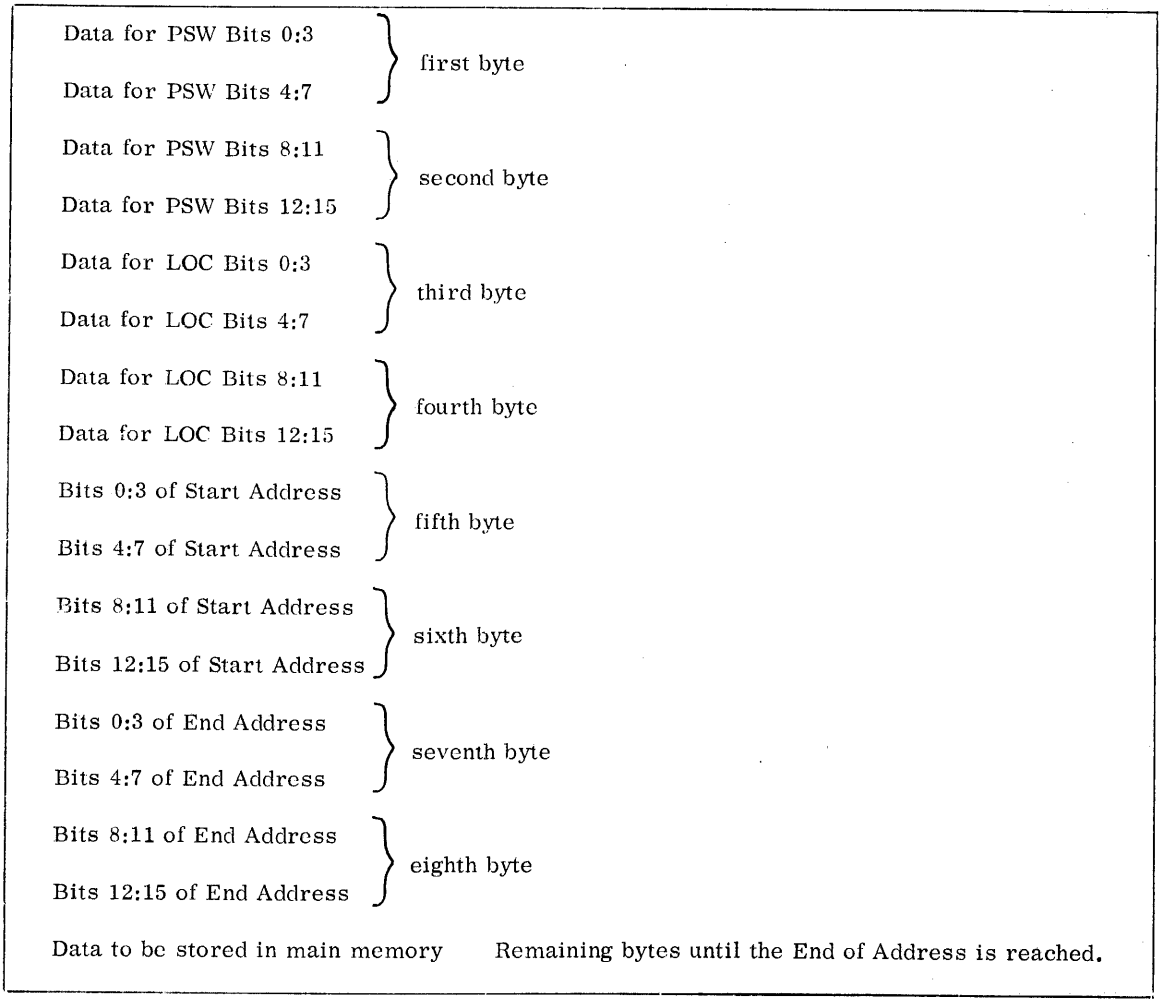


Figure 4. Data Format for Automatic Loading

On the completion of the Auto-Load, the micro-program commences to execute the User program from the Memory Address specified by LOC (third and fourth bytes). Note that if Bit 0 of the PSW, as specified by the first byte, is set, the Processor goes to the "Wait" state. Program execution then begins when the Execute switch is operated.

APPENDIX 1

GENERAL BOOT LOADER PROGRAM

1. PROGRAM DESCRIPTION

The General Boot Loader Program automatically sets the Binary Input Device definition to the device number and command byte associated with the Automatic Loader logic card.

General Boot Loader Program

```
                                ORG   X'0048'
0048   8000                     DC    X'8000' START   SET WAIT BIT
004A   0050
004C   0050                     DC    START,END
004E   005D                     .
0050   2401     START           LIS   0,1             SET REGISTER 0
0052   D900                     RH    0,X'78'         READ DEVNO AND COMMAND
0054   0078
0056   D500                     AL    X'CF'           DO AUTO LOAD
0058   00CF
005A   4300                     B     X'80'
005C   0080
               END              EQV   *-1
```

The above program places the Processor in the Wait state after loading. When the Execute switch is operated, the device number and command byte associated with the Boot Loader logic card is read into location X'0078', the binary input device specification. Then, the Auto Load instruction is executed.

2. PROGRAM CREATION

The user can prepare an eight bit tape of this program on any INTERDATA Processor equipped with a display panel by loading CLUB (03-013) or an equivalent program; then, key the program into memory. Use the 'Q' directive with limits of X'0048' through X'005D'. (Any program can be used which generates an eight bit tape. Note that the tape must not be punched with a leading X'F0' as occurs with some operating systems when punching eight bit paper tape.)

3. OPERATION PROCEDURES

Use the following procedure to load programs using the General Boot Loader Program.

1. Place the PROGRAM LOAD switch in the ENABLE position.
2. Apply power to the appropriate input device (e.g., Teletype) and place the General Boot Loader tape in the device with the first character over the read station.
3. Turn the Power switch to the ON position.
4. Note that if a Teletype is used as the input device, place the lever on the TTY reader to the "START" position.
5. The Processor reads the tape and then halts. Any program that can be loaded with a standard 50 Sequence, (AL X'CF') can now be loaded.
   (B X'80')
6. Place the appropriate paper tape in the Binary Input Device and momentarily depress the EXE switch. The 50 Sequence which was loaded by the Automatic Loader now begins execution.

4. USAGE

Using a Boot Loader Program that establishes the standard 50 Sequence in memory, in general, the following programs can be run on a Model 7/16 with the Automatic Loader option.

1. Any M10 Tape can be loaded immediately after loading the 50 Sequence (i.e., General Loader, Rel Loader, Basic Assembly, BOSS, or DOS object tape).
2. Any M08, M09, M16, or M17 program tape can be loaded with the General Loader or Rel Loader, provided it has an end transfer address (i.e., CLUB, Memory test, or Processor test).
3. Any M14 Tape can be loaded after CLUB has been loaded and executed to modify the 50 Sequence with an appropriate ending address.

APPENDIX 2

ALTERNATE BOOT LOADER PROGRAM

The following listing shows an alternate program designed to be loaded from a TTY. The information loaded comprises a standard '50 Sequence' with device definition table. After loading, the program issues an X-OFF to the TTY to stop the paper tape. Then an asterisk is printed to indicate that the 50 Sequence is in memory and ready. Execution of the 50 Sequence is begun by depressing the Break key on the TTY. This program issues an X-ON to the TTY to start the tape advancing through the reader before the 50 Sequence is begun.

```
INTERDATA MODEL 7/16 AUTO-BOOT LOADER                          PAGE    1

                        *
                        *
                        * ROUTINE IS AUTOMATICALLY LOADED FROM TTY ON
                        * POWER UP OR INITIALIZE.  AFTER LOAD, TTY READER
                        * IS TURNED OFF AND AN ASTERISK IS PRINTED.  THE
                        * OPERATOR THEN PLACES GENERAL LOADER OR REL LOADER
                        * PAPER TAPE IN TTY READER.  DEPRESS BREAK KEY
                        * ON KEYBOARD TO START AUTOLOAD SEQUENCE.
                        *
                        *
0008                    RET     EQU     8
0009                    STAT    EQU     9
000A                    TTY     EQU     10
000B                    TWRT    EQU     11
000C                    TREAD   EQU     12
000D                    DAT     EQU     13
000E                    OUT     EQU     14
000F                    XON     EQU     15
                        *
0048                            ORG     X'0048'
                        *
                        *
0048  0100                       DC      X'0100',START   PSW AND LOC
      0058
004C  0050                       DC      LOAD            START ADDRESS
004E  0093                       DC      END             END ADDRESS
                        *
0050  D500              LOAD     AL      X'CF'
      00CF
0054  4300                       B       X'80'
      0080
                        *
0058  D1A0              START    LM      TTY,DATA        SET UP REGISTERS
      0080
005C  9EAB                       OCR     TTY,TWRT        WRITE MODE
005E  018E                       BALR    RET,OUT         XOFF
0060  94DD                       EXBR    DAT,DAT
0062  018E                       BALR    RET,OUT         *
0064  9EAC                       OCR     TTY,TREAD       READ MODE
0066  9DA9                       SSR     TTY,STAT
0068  2241                       BFBS    4,1             WAIT FOR BREAK
006A  9EAB                       OCR     TTY,TWRT        WRITE MODE
006C  08DF                       LHR     DAT,XON
006F  018E                       BALR    RET,OUT         XON
0070  9DA9                       SSR     TTY,STAT        WAIT FOR
0072  2091                       BTBS    9,1             XON CHARACTER
0074  4300                       B       LOAD            DO AUTO LOAD
      0050
                        *
0078  0294                       DC      X'0294'         BINDV
007A  0298                       DC      X'0298'         BOUTDV
007C  0294                       DC      X'0294'         SINDV
007E  0298                       DC      X'0298'         LISTDV
                        *
0080  0002              DATA     DC      2
0082  0098                       DC      X'98'
```

```
0084  0094              DC     X'94'
0086  2A93              DC     X'2A93'
0088  008C              DC     OUTPUT
008A  0091              DC     X'91'
                  *
008C  9DA9      OUTPUT  SSR    TTY,STAT
008E  20F1              BIBS   15,1
0090  9AAD              WDR    TTY,DAT
0092  0308              BR     RET
                  *
0093            END     EQU    *-1
0094                    END
```



```
NO ERRORS

  DAT      000D
  DATA     0080
  END      0093
  LOAD     0050
  OUT      000E
  OUTPUT   008C
  RET      0008
  START    0058
  STAT     0009
  TREAD    000C
  TTY      000A
  TWRT     000B
 XON       000F
```

MICRO–PROGRAMS

PROG= 05.05104 ASSEMBLED BY 03-103R01M91 6/10/77

```
                                1          SCRAT
                                3          CROSS
                                4          TARGT 7/16H              7/16 WITH HIGH SPEED ALU
                                5  PAGE.0  PARTS 19-122R02F13,19-122R02F12,12-122R03F11          R04 **
                                6          PARTS 19-122R01F10,19-122R02F09,19-122R03F08          R04 **
                                7  PAGE.1  PARTS 19-122R02F19,19-122R00F18,19-122R02F17
                                8          PARTS 19-122R02F16,19-122R02F15,19-122R03F14          R04 **
                                9  PAGE.2  PARTS 19-122R00F25,19-122R00F24,-9-122R00F23
                               10          PARTS 19-122R00F22,19-122R00F21,19-122R00F20
                               11  PAGE.3  PARTS 19-122R00F31,19-122R00F30,19-122R01F29
                               12          PARTS 19-122R01F28,19-122R00F27,19-122R01F26
                               13  PAGE.4  PARTS 19-122R00F37,19-122R00F36,19-122R00F35
                               14          PARTS 19-122R00F34,19-122R00F33,19-122R00F32
                               15          SQCHK
                               16  * MODEL 7/16 WITH HIGH SPEED ALU MICROPROGRAM                    71600040
                               17  * COPYRIGHT INTERDATA INC. DEC. 1974                             71600050
                               18  * FEB. 1977  ECN 3135                                R04 **      71600060
                               19  * DURGA AGARWAL                                                  71600070
                               20  *                                                               71600080
                               21  *                                                               71600090
                               22  *                                                               71600100
                               23  *                                                               71600110
                               24  *                                                               71600120
                               25  *                                                               71600130
                               26  *                                                               71600140
                               27  * ILLEGAL INSTRUCTION OR PROTECT MODE VIOLATION INTERRUPT        71600150
                               28  * LOC HAS BEEN INCREMENTED BY TWO                                71600160
                               29  *                                                               71600170
000      F0 0005               30          B     ILLEG                                            71600180
                               31  * COMMON ENTRY POINT FOR ALL INSTRUCTIONS                       71600190
                               32  * INSTRUCTION READ IS IN PROGRESS                               71600200
                               33  *                                                               71600210
001      02 8010               34  START   L     NULL,NULL,D1      VECTOR THROUGH DROM1           71600220
002      00 E809               35          L     ARL,YSL           (ARL)=(R2)                     71600230
                               36  *                               CONTROL FIELD IS TAKEN FROM    71600240
                               37  *                               LS 4 BITS OF DROM1             71600250
                               38  * SEE DROM1 LISTING FOR BRANCH ADDRESS CORRESPONDING TO EACH USER INSTR  71600255
                               39  *       COME HERE IF PROCESSOR IS IN USER MODE AND THE          71600260
                               40  *       CURRENT INSTRUCTION IS A PRIVILEGED INSTRUCTION         71600270
003      50 A314               41  PRIV    S     LOC,LOC,TWO       DECR. LOC BY 2                 71600280
004      E4 4006               42          BT    RR,*+2            SKIP NEXT INST. IF RR OR SHORT INST.  71600290
005      50 A314               43  ILLEG   S     LOC,LOC,TWO       DECREMENT LOC BY TWO           71600300
006      83 0017               44          LI    MAR,'30'          (MAR) = ADDRESS OF ILLEGAL OR  71600310
                               45  *                               PROTECT MODE VIOLATION INT. NEW PSW  71600320
007      F0 0059               46          B     GENSWP                                           71600330
008                            47          ORG   X'40'                                           71600340
                               48  * COME HERE IF CURRENT INSTRUCTION IS TO BE ABORTED            71600350
                               49  * LOCATION COUNTER IS POINTING TO THE NEXT INSTRUCTION         71600360
                               50  * IF CURRENT INSTRUCTION IS OF RR FORMAT LOC SHOULD BE DECREMENTED  71600370
                               51  * BY 2 OTHERWISE LOC SHOULD BE DECREMENTED BY 4                 71600380
                               52  *                                                               71600390
040      50 A314               53  ABORT   S     LOC,LOC,TWO       DECR. LOC BY TWO               71600400
041      E4 4045               54          BT    RR,HELP           BRANCH TO COMMON ROUTINE IF CURRENT  71600410
                               55  *                               INST. IS OF RR FORMAT          71600420
```

```
042  50 A314          56          S     LOC,LOC,TWO        DECR. LOC BY 2                          71600430
043  00 8010          57          L     NULL,NULL          FILLER                                  71600440
044  00 8010          58          L     NULL,NULL          FILLER                                  71600450
                      59  * COMMON INTERRUPT SUPPORT ROUTINE                                       71600460
                      60  *                                                                       71600470
045  F6 0054          61  HELP    BF    ATNX,HELP1         BRANCH TO HELP1 IF AUTOMATIC I/O        71600480
                      62  *                                SERVICE IS NOT REQUIRED                 71600490
                      63  * COME HERE FOR AUTOMATIC I/O SERVICE                                    71600500
046  00 A881          64  AUTIO   L     MR1,IO,ACK         (MR1)=DEVICE NUMBER                     71600510
047  00 0A09          65  AUTIO1  L     ARL,MR1,SL         (ARL)=2*DEVICE NUMBER                   71600520
048  CD 0217          66  AUTIO3  AI    MAR,'D0',ARL       (MAR)=INT. SERV. POINTER ADDRESS        71600530
049  08 8010          67              L     NULL,NULL,MR     FETCH INT. SERVICE POINTER              71600540
04A  00 B443          68  AUTIO2  L     MR3,MDR,SR+C0      (MR3)= SERV.POINTER/2                   71600550
04B  00 1A17          69              L     MAR,MR3,SL       (MAR) = SERV. POINTER (FORCED EVEN)     71600560
04C  E2 01C9          70              BT    C,CHANEL         BRANCH TO CHANEL I/O ROUTINE IF         71600570
                      71  *                                SERVICE POINTER IS ODD                  71600580
                      72  * COME HERE FOR IMMEDIATE INTERRUPT                                      71600590
                      73  * (MAR) = SERVICE POINTER                                               71600600
04D  00 3816          74  IMMINT  L     MDR,PSWL           STORE PROGRAM STATUS                    71600610
04E  7F 0090          75              C     MW2+PRIV                                             71600620
04F  00 A016          76              L     MDR,LOC          STORE LOCATION COUNTER                 71600630
050  7F 0090          77              C     MW2+PRIV                                             71600640
051  09 8007          78              L     PSWL,NULL,MR2    CLEAR PSW;FETCH NEW PROG. STATUS       71600650
052  05 B007          79              L     PSWL,MDR,IRJH    LOAD NEW STATUS AND FETCH NEW          71600660
053  00 B814          80              L     LOC,MAR          INST. FROM SERV.POINTER +6             71600670
054  E8 10D7          81  HELP1   BT    PPF,PWRDWN         TO PWRDWN IF POWER FAIL                 71600680
055  E9 206B          82              BT    MALF+CATN,MMFINT EPF+MPE+CATN                  R03     71600690
056  E8 80CD          83              BT    DC,DCR           TO DCR IF DATA CHANNEL REQUEST         71600700
057  F4 20FA          84              BF    ATN,CONSER1      BRANCH IF NOT ATN             R03      71600710
                      85  * EXTERNAL INTERRUPT                                                    71600720
                      86  * AUTOMATIC I/O SERVICE BIT IS RESET                                    71600730
058  84 0017          87  EXINT   LI    MAR,'40'           (MAR)=ADDR. OF EXT. INT. OLD PSW        71600740
                      88  * GENERAL PSW SWAP ROUTINE                                             71600750
                      89  * THIS ROUTINE STORES THE CURRENT PSW IN THE FULL WORD LOCATION         71600760
                      90  * POINTED TO BY MAR. NEW PSW IS OBTAINED FROM (MAR)+4                   71600770
                      91  *                                                                       71600780
                      92  *                                                                       71600790
059  00 3816          93  GENSWP  L     MDR,PSWL           STORE PROG. STATUS                      71600800
05A  7F 0090          94              C     MW2+PRIV                                             71600810
05B  00 A016          95              L     MDR,LOC          STORE LOC                              71600820
05C  7F 0090          96              C     MW2+PRIV                                             71600830
05D  09 8010          97              L     NULL,NULL,MR2    FETCH NEW PROG. STATUS                 71600840
                      98  * ENTER FROM LPSW FOR LOAD PROGRAM STATUS WORD                          71600850
                      99  * CONTINUE FOR GENSWAP ROUTINE                                         71600860
                     100  *                                                                       71600870
05E  08 B007         101  LPSW1   L     PSWL,MDR,MR        LOAD NEW PROG. STATUS                  71600880
05F  70 0088         102              C     JH               COPY FLR TO COND. CODE                 71600890
060  00 B014         103              L     LOC,MDR          LOAD NEW LOC VALUE                     71600900
                     104  *                                                                       71600910
061  E4 80F8         105  TEST    BT    QUE,QUETST         QUEUE INTRPT IF PSW 06         R03     71600920
062  F0 1279         106  TEST1   BF    WAIT,NBR           FETCH NEXT INSTRUCTION IF WAIT          71600930
                     107  *                                BIT IS RESET                           71600940
                     108  * INTERRUPTABLE WAIT LOOP                                              71600950
063  70 0004         109  WAIT    C     SWA                SET WAIT INDICATOR                     71600960
```

```
064       EA 0081              110           BT    SNGL,CONSER          GO TO CONSER IF SNGL IS ACTIVE          71600970
065       00 8010              111  WAIT1    L     NULL,NULL            NOP                                     71600980
066       E4 2068              112           BT    ATN,WAIT2                                                    71600990
067       F9 B065              113           BF    MALF+PPF+CATN+DC,WAIT1                                       71601000
068       70 0000              114  WAIT2    C     CWA                  CLEAR WAIT INDICATOR                    71601010
069       EA 0094              115           BT    SNGL,CLRWT                                                   71601020
06A       F0 0045              116           B     HELP                                                         71601030
                               117  * MACHINE MALFUNCTION INTERRUPT                                             71601040
06B       F8 2081              118  MMFINT   BF    MALF,CONSER          TO CONSER IF NO MALF          R04**    71601050
06C       83 8017              119  MMFINT1  LI    MAR,X'38'            (MAR)=X'38' A(MMF OLD PSW)    R04 **   71601055
06D       00 3816              120           L     MDR,PSWL             STORE PROG.STATUS                      71601060
06E       7F 0090              121           C     MW2+PRIV                                                     71601070
06F       00 A016              122           L     MDR,LOC              STORE LOC                              71601080
070       7F 0090              123           C     MW2+PRIV                                                     71601090
071       09 8010              124           L     NULL,NULL,MR2        FETCH NEW STATUS                       71601100
                               125  *                                                                           71601110
                               126  *                                                                           71601120
072       F0 00F0              127           B     MMF2                 BRANCH TO TEST CATN           R03      71601130
                               128  * QUEUE SERVICE INTERRUPT ENABLED                                           71601140
                               129  * TEST  TERMINATION QUEUE                                                  71601150
073       88 0017              130  QUEINT   LI    MAR,'80'             (MAR)=ADDRESS OF TERM.QUE.POINTER      71601160
074       08 8010              131           L     NULL,NULL,MR         FETCH QUEUE POINTER                    71601170
075       00 B017              132           L     MAR,MDR              (MAR)= TERM. QUEUE POINTER             71601180
076       08 800F              133           L     FLR,NULL,MR          CLEAR FLR; FETCH 1ST HW OF LIST        71601190
077       9F F430              134           NI    NULL,'FF',MDR,F      SET G FLAG IF NO. OF SLOTS USED >0     71601200
078       88 2017              135           LI    MAR,'82'             (MAR)=ADDR.QUE.SER.INT. OLD PSW        71601210
079       E0 8059              136           BT    G,GENSWP             GO TO PSW SWAP ROUTINE IF LIST IS      71601220
                               137  *                                  NOT EMPTY                              71601230
07A       F0 0062              138           B     TEST1                TEST WAIT BIT                          71601240
                               139  * CONSOLE SERVICE                                                          71601250
                               140  *                                                                           71601260
                               141  * UN-INTERRUP TABLE IDLE LOOP                                              71601270
07B       70 0004              142  IDLE     C     SWA                  SET WAIT INDICATOR                     71601280
07C       F9 107B              143  IDLE1    BF    PPF+CATN,IDLE        LOOP IF NOT PPF OR CATN                71601290
07D       70 0000              144           C     CWA                  CLEAR WAIT INDICATOR                   71601300
07E       E8 10D7              145           BT    PPF,PWRDWN           TO PWRDWN IF PPF                       71601310
07F       80 1035              146           LI    IO,'1',ADRS          ADDRESS THE CONSOLE                    71601320
080       EA 0094              147           BT    SNGL,CLRWT           SINGLE STEP                            71601330
                               148  * FALL THROUGH TO CONSER                                                   71601340
                               149  *                                                                           71601350
081       80 1035              150  CONSER   LI    IO,'1',ADRS          ADDRESS THE CONSOLE                    71601360
082       00 A8C0              151           L     MR0,IO,STAT          (MR0)=CONSOLE STATUS                   71601370
083       00 0413              152           L     SRH,MR0,SR                                                   71601380
084       70 0CD0              153           C     SR2                  SHIFT RIGHT TWO PLACES                 71601390
085       00 9C01              154           L     MR1,SRH,SR           MR1(12:15)= STATUS (0:3)               71601400
086       82 0017              155           LI    MAR,'20'             (MAR)='20'                             71601402
087       00 0016              156           L     MDR,MR0              (MDR)=CONSOLE STATUS                   71601404
088       7E 0090              157           C     MW+PRIV              STORE CONSOLE STATUS                   71601406
                               158  *        TEST FOR AUTO LOADER                                              71601410
089       80 F009              159           LI    ARL,'F'                                                      71601420
08A       30 0A0F              160           X     FLR,MR1,ARL                                                  71601430
08B       F0 00FC              161           B     PATCH                *                             R03      71601440
08C       80 1035              162  RETURN   LI    IO,'01',ADRS         RE-ADDRESS, CLEAR COUNTERS    R03      71601450
08D       00 080F              163           L     FLR,MR1              (FLR)=STATUS .0:3>                     71601460
```

```
08E  EA 00AC  164          BT   SNGL,DISPLY        TO DISPLAY IF SINGLE          R03   71601470
08F  E1 00AD  165 NOSNGL   BT   V,FNDIS           FUNCTION OR REG.DISPLAY IF V IS SET  71601480
090  00 A017  166          L    MAR,LOC            (MAR)=(LOC)                          71601490
091  08 8010  167          L    NULL,NULL,MR       START MEMORY READ                    71601500
092  E0 4097  168          BT   L,ADRMW            ADDRESS OR WRITE IF L IS SET         71601510
093  E0 809A  169          BT   G,DISMEM           DISPLAY A HW FROM MEMORY             71601520
               170 * RUN MODE                                                          71601530
094  00 3A06  171 CLRWT    L    MR6,PSWL,SL        RESET WAIT BIT OF PSW                71601540
095  00 3407  172          L    PSWL,MR6,SR                                             71601550
096  F0 0279  173          B    NBR                FETCH NEXT INSTRUCTION               71601560
097  00 A8B6  174 ADRMW    L    MDR,IO,DR          INPUT LOW BYTE                       71601570
098  00 AEB6  175          L    MDR,IO,DR+CS       INPUT HIGH BYTE                      71601580
099  E0 809F  176          BT   G,ADR              ADRS MODE IF G IS SET                71601590
               177 *                                                                   71601600
               178 *                                                                   71601610
               179 * MEMORY WRITE                                                     71601620
09A  0E B001  180 DISMEM   L    MR1,MDR,MW         (MR1)=DISPLAY BYTES D2 AND D1:       71601630
               181 *                               MEMORY WRITE                        71601640
09B  40 A314  182          A    LOC,LOC,TWO        INCR. LOC BY 2                       71601650
09C  00 A000  183          L    MR0,LOC            (MR0)=DISPLAY BYTES D4 AND D3        71601660
09D  88 0002  184          LI   MR2,'80'           MR2(8:15) =DISPLAY BYTE D5           71601670
               185 *                               MSB OF D5 IS SET TO INDICATE        71601680
               186 *                               ADDRESS MODE                        71601690
09E  F0 00A5  187          B    OUTDIS             DISPLAY BYTES D5,D4,D3,D2 & D1       71601700
               188 * ADDRESS                                                           71601710
09F  00 B414  189 ADR      L    LOC,MDR,SR         (LOC)=(MAR)= SELECTED                71601715
0A0  00 A214  190          L    LOC,LOC,SL         ADDRESS(FORCED EVEN)                 71601720
0A1  80 1035  191 LOCDIS   LI   IO,'1',ADRS        ADDRESS THE CONSOLE                  71601730
0A2  00 8000  192          L    MR0,NULL           (MR0)=DISPLAY BYTES D4 & D3          71601740
0A3  00 A001  193          L    MR1,LOC            (MR1) = DISPLAY BYTES D2 & D1        71601750
0A4  84 5002  194          LI   MR2,'45'           (DISPLAY BYTE D5)=X'45'              71601760
0A5  00 0855  195 OUTDIS   L    IO,MR1,DA          OUTPUT BYTES D1                      71601770
0A6  00 0E55  196          L    IO,MR1,DA+CS       OUTPUT D2                            71601780
0A7  00 0055  197          L    IO,MR0,DA          OUTPUT D3                            71601790
0A8  00 0655  198          L    IO,MR0,DA+CS       OUTPUT D4                            71601800
0A9  00 1055  199          L    IO,MR2,DA          OUTPUT D5                            71601810
0AA  88 0075  200          LI   IO,'80',OC         DISPLAY IN NORMAL MODE:              71601820
               201 *                               CLEAR COUNTERS                      71601830
0AB  F0 007B  202          B    IDLE               BRANCH TO IDLE IF V IS RESET         71601840
0AC  F1 007B  203 DISPLY   BF   V,IDLE             * COME HERE FOR FUNCTION OR REGISTER DISPLAY  71601850
               204                                                                     71601860
               205 *                                                                   71601870
0AD  00 028C  206 FNDIS    L    YSI,MR0,SL+CI      YSI(3)=STATUS(0);YSI(0:2)=STATUS(5:7)  71601880
0AE  00 000F  207          L    FLR,MR0            (FLR)=STATUS(4:7)                    71601890
0AF  F2 00C0  208          BF   C,FN               FUNCTION IF C IS RESET               71601900
               209 * GEN REG. OR FLOAT REGISTER DISPLAY                                71601910
0B0  00 080F  210 REGDIS   L    FLR,MR1            (FLR)=STATUS(0:3)                    71601920
0B1  E0 40B7  211          BT   L,FLTREG           FLOAT. REG. DISPLAY IF L IS SET      71601930
               212 * GENERAL REGISTER DISPLAY                                          71601940
               213 *                                                                   71601950
               214 *                                                                   71601960
0B2  00 E000  215          L    MR0,YSH            (MR0)=HIGH REG.                      71601970
0B3  00 E801  216          L    MR1,YSL            (MR1)=LOW REG.                       71601980
0B4  82 0009  217          LI   ARL,'20'           (ARL)=X'20'                          71601990
```

```
0B5     20 6202     218            O     MR2,YSI,ARL      MR2(8:15)= D5(ILLUMINATE FIXED REG   71602000
                    219  *                                INDICATOR AND DISPLAY HEX. REG.#)    71602010
0B6     F0 00A5     220            B     OUTDIS           OUTPUT DISPLAY BYTES                 71602020
0B7     80 E009     221  FLTREG    LI    ARL,'E'                                               71602030
0B8     10 6202     222            N     MR2,YSI,ARL      (MR2)=FLOAT. REG. #/2                71602040
0B9     00 1217     223            L     MAR,MR2,SL       (MAR)=ADDRESS OF THE FLOAT. REG.     71602050
0BA     09 8010     224            L     NULL,NULL,MR2    START MEMORY READ                    71602060
                    225  *                                AND INCREMENT MAR BY 2               71602070
0BB     08 B000     226            L     MR0,MDR,MR       (MR0) = HIGH FLOAT. REG.             71602080
0BC     00 B001     227            L     MR1,MDR          (MR1) = LOW FLOAT. REG.              71602090
0BD     81 0009     228            LI    ARL,'10'                                              71602100
0BE     20 1202     229            O     MR2,MR2,ARL      (MR2)=D5(ILLUMINATE FLOAT.REG.IND.   71602110
                    230  *                                AND DISPLAY HEX REG. #)              71602120
0BF     F0 00A5     231            B     OUTDIS           OUTPUT DISPLAY BYTES                 71602130
                    232  * FUNCTION                                                            71602140
0C0     00 600F     233  FN        L     FLR,YSI          (FLR)=FUNCTION #                     71602150
0C1     E2 807B     234            BT    C+G,IDLE         GO TO IDLE IF ANY UNDEFINED FUNCT.   71602160
0C2     F1 00C8     235            BF    V,FN01           FUNCT 0 OR 1 IF V IS RESET           71602170
                    236  * COME HERE FOR FUNCTION 4 OR 5                                       71602180
0C3     E0 40A1     237            BT    L,LOCDIS         DISPLAY (LOC) IF L IS SET            71602190
                    238  * DISPLAY PROGRAM STATUS                                             71602200
0C4     00 8000     239  PSWDIS    L     MR0,NULL         (MR0)= 0                             71602210
0C5     00 3801     240            L     MR1,PSWL         (MR1) = LOW PSW                      71602220
0C6     84 4002     241            LI    MR2,'44'         (MR2) = DISPLAY BYTE D5 (IND. FN4)   71602230
0C7     F0 00A5     242            B     OUTDIS           OUTPUT DISPLAY BYTES                 71602240
                    243  * FUCNTION 0 OR 1                                                    71602250
0C8     E0 407B     244  FN01      BT    L,IDLE           GO TO IDLE IF FUNC. 1(UNDEFINED )    71602260
                    245  * FUNCTION 0(CONSOLE INTERRUPT)                                      71602270
0C9     80 1001     246  FN0       LI    MR1,'1'          (MR1) = DISPLAY DEVICE NUMBER        71602280
0CA     00 3E0F     247            L     FLR,PSWL,CS      (FLR)=PSW BITS 4:7                   71602290
0CB     E2 0047     248            BT    C,AUTI01         BRANCH TO AUTI01 IF BIT 4=1          71602300
0CC     F0 0094     249            B     CLRWT            EXECUTE NEXT INSTRUCTION             71602310
                    250  *  DATA CHANNEL REQUEST                                              71602320
0CD     00 8015     251  DCR       L     IO,NULL,DCAK     DATA CHANNEL ACKNOWLEDGE             71602330
0CE     E8 40D3     252            BT    DRD,DCREAD                                            71602340
                    253  *  DATA CHANNEL WRITE                                                71602350
0CF     00 A8B7     254            L     MAR,IO,DR        HALF WORD ADDRESS TO MAR             71602360
0D0     7A 0000     255            C     MR0              READ A HALF WORD FROM MEMORY         71602370
0D1     00 B055     256            L     IO,MDR,DA        OUTPUT A HALF WORD                   71602380
0D2     F0 0045     257            B     HELP                                                  71602390
                    258  *  DATA CHANNEL READ                                                 71602400
0D3     00 A8B7     259  DCREAD    L     MAR,IO,DR        HALF WORD ADDRESS TO MAR             71602410
0D4     00 A8B6     260            L     MDR,IO,DR        INPUT A HW                           71602420
0D5     7E 0090     261            C     MW+PRIV          WRITE TO MEMORY                      71602430
0D6     F0 0045     262            B     HELP             TEST FOR INTERRUPTS                  71602440
                    263  *  POWER DOWN                                                        71602450
0D7     82 2017     264  PWRDWN    LI    MAR,'22'         (MAR)=X'22'                          71602460
                    265  *                                                                     71602470
                    266  *                                                                     71602480
0D8     09 8010     267            L     NULL,NULL,MR2    FETCH REG. SAVE POINTER              71602490
0D9     00 B000     268            L     MR0,MDR          (MR0)=REG. SAVE POINTER              71602500
0DA     00 3816     269            L     MDR,PSWL         STORE PROGRAM STATUS                 71602510
0DB     7F 0090     270            C     MW2+PRIV                                              71602520
0DC     00 A016     271            L     MDR,LOC          STORE LOC:CLEAR YD FIELD             71602530
```

```
00DD  7E 0091   272          C    MW+PRIV+CYD                                          71602540
00DE  00 0017   273          L    MAR,MR0     (MAR)=ADDRESS OF REG. SAVE AREA          71602550
00DF  80 F008   274          LI   CTR,'F'     (COUNTER)=15                             71602560
                275  *  STORE REGISTERS                                               71602570
00E0  0F D016   276  SLOOP1   L    MDR,YDLP1,MW2  STORE ONE REG.                       71602580
00E1  FC 80E0   277          BF   CNTR,SLOOP1    REPEAT UNTIL DONE                   71602590
00E2  00 B016   278          L    MDR,MDR        WAIT UNTIL WRITE OPER. IS COMPLETE   71602595
00E3  70 0002   279          C    POW            SYSTEM RESET                        71602596
00E4            280          ORG  '0F0'                                      R03     71602598
00F0  70 0001   281  MMF2    C    ALRM           (FLR)=(CC)V(ALRM REG)       R03     71602600
                282  *                           WHEN PSWL IS LOADED         R03     71602601
00F1  08 B009   283          L    ARL,MDR,MR     SAVE NEW PSW IN ARL         R03     71602602
00F2  00 8007   284          L    PSWL,NULL      CLEAR CURRENT PSW            R03    71602603
00F3  70 0088   285          C    JH             PROPOGATE ALARM BITS        R03     71602604
00F4  20 3A07   286          O    PSWL,PSWL,ARL  COMBINE NEW PSW VALUE        R03     71602605
00F5  00 B014   287          L    LOC,MDR        LOAD NEW LOC VALUE           R03    71602606
00F6  E9 0081   288          BT   CATN,CONSER    LEAVE IF CONSOLE ATTENTION   R03    71602607
00F7  F0 0061   289          B    TEST           GO TEST QUEUE                R03    71602608
00F8  F9 0073   290  QUETST  BF   CATN,QUEINT     DO INTERRUPT IF NO CATN     R03     71602609
00F9  F0 0081   291          B    CONSER         ELSE SERVICE CONSOLE         R03    71602610
00FA  EA 0081   292  CONSER1 BT   SNGL,CONSER     TO CONSER IF SINGLE STEP     R03    71602611
00FB  F0 0062   293          B    TEST1          IGNORE SPURIOUS INTERRUPT    R03    71602612
00FC  F3 C094   294  PATCH   BF   C+V+G+L,CLRWT   TO CLRWT IF AUTOLOADER       R03    71602613
00FD  88 0075   295          LI   IO,'80',0C     COMMAND "NORMAL" MODE        R03    71602614
00FE  F0 008C   296          B    RETURN         *                            R03    71602615
```

```
0FF                    298            ORG    '100'                                                    71602617
                       299    * INITIALIZE / POWERUP ROUTINE                                          71602620
100    82 0077         300    PWRUP   LI     MAR,'20',CO+F        (MAR)='20':REST V FLAG              71602630
101    80 5035         301            LI     IO,'5',ADRS          ADDRESS THE DEV. NUMBER 5           71602640
102    F1 0119         302            BF     V,AUTO1              DO AUTOLOAD FROM LOADER STORAGE     71602650
                       303    *                                     UNIT IF NO FALSE SYNC             71602660
                       304    * NORMAL POWERUP                                                        71602670
103    09 800F         305    PWRUP1  L      FLR,NULL,MR2         CLEAR FLR;FETCH SAVED CONS. STAT.   71602680
104    09 B00A         306            L      ARH,MDR,MR2          (MDR)=OLD CONSOLE STATUS            71602690
105    09 B000         307            L      MR0,MDR,MR2          (MR0)=REG. SAVE POINTER             71602700
106    09 B007         308            L      PSWL,MDR,MR2         LOAD PROGRAM STATUS AND             71602710
107    70 0089         309            C      JH+CYD               CLEAR YD FIELD AND COPY FLR TO CC   71602720
108    00 B014         310            L      LOC,MDR              RESTORE LOC                         71602730
109    00 0017         311            L      MAR,MR0              (MAR)=ADDR. OF REG. SAVE AREA       71602734
10A    80 F008         312            LI     CTR,'F'              (COUNTER)=15                        71602738
10B    09 8010         313            L      NULL,NULL,MR2                                            71602742
                       314    * RESTORE ALL REGISTERS                                                 71602746
10C    09 B01A         315    LLOOP1  L      YDLP1,MDR,MR2        LOAD ONE REGISTER                   71602750
10D    FC 810C         316            BF     CNTR,LLOOP1          REPEAT UNTIL ALL REGS. HAVE         71602760
                       317    *                                   BEEN LOADED                         71602770
10E    80 1035         318            LI     IO,'1',ADRS          ADDREESS THE CONSOLE                71602780
10F    00 A8CF         319            L      FLR,IO,STAT          SENSE STATUS                        71602790
110    E0 8141         320            BT     G,PWRUP2             BOOT LOADER IS DISABLED OR          71602800
                       321    *                                   BASIC CONTROL SWITCH                71602810
111    E0 4115         322            BT     L,BOOTL              TO BOOTL IF BOOT LOADER ENABLED     71602820
                       323    * DISPLAY PANEL IS PRESENT                                              71602830
112    97 0730         324            NI     NULL,'70',ARH,F      TEST BITS 1:3 OF OLD CONS. STATUS   71602840
113    F0 C141         325            BF     G+L,PWRUP2           TO PWRUP2 IF CONS. WAS IN RUN MODE  71602850
                       326    * DISPLAY WAS NOT IN RUN MODE                                           71602860
114    F0 00A1         327            B      LOCDIS               GO TO LOCDIS                        71602870
                       328    * DISPLAY IS NOT PRESENT (STATUS=XXXX XX01)                             71602880
                       329    * BOOT LOAD FROM AUTOMATIC LOADER IS REQUIRED                           71602890
115    00 A8A0         330    BOOTL   L      MR0,IO,DR            AUTO-BOOT DEV. NO.                  71602900
116    00 A8A1         331            L      MR1,IO,DR            COMMAND FOR AUTOBOOT DEVICE         71602910
117    00 0035         332            L      IO,MR0,ADRS          ADDRESS THE AUTO-BOOT DEVICE        71602920
118    00 0875         333            L      IO,MR1,OC            OUTPUT COMMAND TO AUTO-BOOT DEVICE  71602930
                       334    * AUTO LOAD FROM LOADER STORAGE UNTI OR AUTO-BOOT DEVICE                71602940
119    00 A8CF         335    AUTO1   L      FLR,IO,STAT                                              71602950
11A    E2 0119         336            BT     C,AUTO1                                                  71602960
11B    00 AEA9         337            L      ARL,IO,DR+CS                                             71602970
11C    00 A8CF         338    AUTO2   L      FLR,IO,STAT                                              71602980
11D    E2 011C         339            BT     C,AUTO2                                                  71602990
11E    00 A8A1         340            L      MR1,IO,DR                                                71603000
11F    20 0A07         341            O      PSWL,MR1,ARL         LOAD PSW                            71603010
120    70 0088         342            C      JH                                                       71603020
121    00 A8CF         343    AUTO3   L      FLR,IO,STAT                                              71603030
122    E2 0121         344            BT     C,AUTO3                                                  71603040
123    00 AEA9         345            L      ARL,IO,DR+CS                                             71603050
124    00 A8CF         346    AUTO4   L      FLR,IO,STAT                                              71603060
125    E2 0124         347            BT     C,AUTO4                                                  71603070
126    00 A8A1         348            L      MR1,IO,DR                                                71603080
127    20 0A14         349            O      LOC,MR1,ARL          LOAD LOC                            71603090
128    00 A8CF         350    AUTO5   L      FLR,IO,STAT                                              71603100
129    E2 0128         351            BT     C,AUTO5                                                  71603110
```

```
12A  00 AEA9  352          L    ARL,IO,DR+CS                                  71603120
12B  00 A8CF  353  AUT06   L    FLR,IO,STAT                                   71603130
12C  E2 012B  354          BT   C,AUT06                                       71603140
12D  00 A8B7  355          L    MAR,IO,DR                                     71603150
12E  20 BA17  356          O    MAR,MAR,ARL      (MAR)=START ADDRESS          71603160
12F  00 A8CF  357  AUT07   L    FLR,IO,STAT                                   71603170
130  E2 012F  358          BT   C,AUT07                                       71603180
131  00 AEA9  359          L    ARL,IO,DR+CS                                  71603190
132  00 A8CF  360  AUT08   L    FLR,IO,STAT                                   71603200
133  E2 0132  361          BT   C,AUT08                                       71603210
134  00 A8A1  362          L    MR1,IO,DR                                     71603220
135  20 0A01  363          O    MR1,MR1,ARL      (MR1) = END ADDRESS          71603230
136  00 B809  364          L    ARL,MAR                                       71603240
137  58 0A41  365          S    MR1,MR1,ARL,CO+MR (MR1) =FINAL ADDR. - START ADDR.;MR 71603250
138  E2 007B  366          BT   C,IDLE      BRANCH TO IDLE IF FINAL ADDRESS   71603260
             367  *                           IS LESS THAN START ADDRESS      71603270
139  00 A8CF  368  AUTOL   L    FLR,IO,STAT      SENSE STATUS                  71603280
13A  E1 C07B  369          BT   V+G+L,IDLE  TO IDLE IF BAD STATUS             71603290
13B  E2 0139  370          BT   C,AUTOL     LOOP ON BUSY                      71603300
13C  0E AEB6  371          L    MDR,IO,DR+CS+MW  INPUT A BYTE & STORE IT      71603310
13D  50 0941  372          S    MR1,MR1,ONE,CO   DECR. BYTE COUNT BY 1        71603320
13E  E2 0062  373          BT   C,TEST1     TEST WAIT BIT IF DONE             71603330
13F  40 B917  374          A    MAR,MAR,ONE      INCR. MAR BY 1               71603340
140  F0 0139  375          B    AUTOL            REPEAT                       71603350
             376  *  DISPLAY WAS IN RUN MODE ON POWER DOWN/INITIALIZE         71603360
141  FC 40A1  377  PWRUP2  BF   ARST+LOCDIS      DISPLAY LOC AND ENTER IDLE LOOP 71603370
             378  *                           IF NO AUTO RESTART              71603380
142  00 3A13  379          L    SRH,PSWL,SL                                   71603390
143  70 1440  380          C    SL2+CO           CARRY=PSW BIT 2              71603400
144  F2 0279  381          BF   C,NBR       FETCH NEXT INST. IF MACHINE       71603410
             382  *                           MALF INT. HAS NOT BEEN ENABLED  71603420
145  F0 006C  383          B    MMFINT1     TAKE MACHINE MALF INTERRUPT  R04 ** 71603430
             384  * LOAD PROGRAM STATUS WORD                                  71603440
             385  * ENTER FROM D2                                            71603450
146  09 8010  386  LPSW    L    NULL,NULL,MR2    FETCH NEW PROGRAM STATUS     71603460
147  F0 005E  387          B    LPSW1            BRANCH TO COMMON ROUTINE     71603470
             388  * EXCHANGE PROGRAM STATUS REGISTER                          71603480
             389  * ENTER FROM D1                                            71603490
148  00 3819  390  EPSR    L    YDL,PSWL         (R1)=OLD PSW                 71603500
149  00 E807  391          L    PSWL,YSL         LOAD NEW PSW FROM R2         71603510
14A  70 0088  392          C    JH               COPY FLR TO COND CODE        71603520
14B  F0 0061  393          B    TEST                                          71603530
             394  * SIMULATE INTERRUPT                                       71603540
             395  *  ENTER FROM DROM1                                        71603550
14C  40 F409  396  SINT    A    ARL,YSLX,MDR     (ARL)=SECOND OPERAND         71603560
14D  9F F201  397          NI   MR1,'FF',ARL     (MR1)=DEV. NUMBER            71603570
14E  F0 0047  398          B    AUTIO1           BRANCH TO AUTOMATIC I/O ROUTINE 71603580
             399  *                                                          71603590
             400  * SUPERVISOR CALL                                          71603600
             401  * ENTER FROM D2                                            71603610
             402  *                                                          71603620
14F  00 B816  403  SVC     L    MDR,MAR          (MDR)=EFFECTIVE ADDRESS      71603630
150  89 4017  404          LI   MAR,'94'                                      71603640
151  7F 0090  405          C    MW2+PRIV         STORE A+(X2)                 71603650
```

```
152  00 3816   406           L     MDR,PSWL                                    71603660
153  7F 0090   407           C     MW2+PRIV          STORE PROGRAM STATUS      71603670
154  00 A016   408           L     MDR,LOC                                     71603680
155  7F 0090   409           C     MW2+PRIV          STORE LOC                 71603690
156  09 6A09   410           L     ARL,YDI,SL+MR2    (ARL)=2*R1                71603700
157  40 BA17   411           A     MAR,MAR,ARL       (MAR)='9C'+2*R1           71603710
158  08 B007   412           L     PSWL,MDR,MR       LOAD PROGRAM STATUS       71603720
159  00 B014   413           L     LOC,MDR           LOAD LOC                  71603730
15A  05 8010   414           L     NULL,NULL,IRJH    FETCH NEXT INST, AND COPY FLR TO CC 71603740
15B  00 8010   415           L     NULL,NULL                                   71603750
15C  00 C835   416  RDR      L     IO,YDL,ADRS       ADDRESS THE DEVICE        71603760
15D  05 A8BD   417           L     YSL,IO,DR+IRJH    (YSL)=INPUT DATA;FETCH NEXT INSTR. 71603770
15E  00 8010   418           L     NULL,NULL         NOP                       71603780
               419  * READ DATA                                               71603790
               420  *  ENTER FROM D2                                          71603800
15F  08 8010   421  RD       L     NULL,NULL,MR      START MEM. READ           71603810
160  00 C835   422           L     IO,YDL,ADRS       ADDRESS THE DEVICE        71603820
161  0E AEB6   423           L     MDR,IO,DR+CS+MW   INPUT DATA, AND STORE IT  71603830
162  00 A017   424           L     MAR,LOC           (MAR)=(LOC)               71603840
163  07 8010   425           L     NULL,NULL,IRJ                               71603850
164  00 8010   426           L     NULL,NULL         START INST.FETCH;COPY FLR TO CC 71603860
               427  *  WRITE  DATA  REGISTER                                   71603870
               428  *                                                         71603880
               429  *                             (MAR)=EFFECTIVE ADDRESS     71603890
165  00 C835   430  WDR      L     IO,YDL,ADRS       ADDRESS THE DEVICE        71603900
166  05 E855   431           L     IO,YSL,DA+IRJH    OUTPUT DATA AND FETCH NEXT 71603910
167  00 8010   432           L     NULL,NULL         INSTRUCTION;COPY FLR TO CC 71603920
               433  *          WRITE   DATA                                    71603930
               434  *  ENTER FROM D2                                          71603940
168  08 8010   435  WD       L     NULL,NULL,MR      START MEM. READ           71603950
169  00 C835   436           L     IO,YDL,ADRS       ADDRESS THE DEVICE        71603960
16A  00 B655   437           L     IO,MDR,CS+DA      OUTPUT DATA               71603970
16B  00 A017   438           L     MAR,LOC           (MAR)= (LOC)              71603980
16C  05 8010   439           L     NULL,NULL,IRJH    FETCH NEXT INST, AND COPY 71603990
16D  00 8010   440           L     NULL,NULL         FLR TO COND. CODE         71604000
               441  *                                                         71604010
               442  *                                                         71604020
               443  *                                                         71604030
               444  *                                                         71604040
               445  *  AIR                                                    71604050
16E  00 A899   446  AIR      L     YDL,IO,ACK        (R1)= DEVICE NUMBER       71604060
               447  * SENSE  STATUS REGISTER                                  71604070
16F  00 C835   448  SSR      L     IO,YDL,ADRS       DEVICE ADDRESS            71604080
170  00 A8DD   449           L     YSL,IO,STAT       INPUT STATUS              71604090
171  05 E8UF   450           L     FLR,YSL,IRJH      COPY LS 4 BITS OF STATUS TO CC ; 71604100
172  00 8010   451           L     NULL,NULL         FETCH NEXT INST           71604110
               452  * AI                                                      71604120
               453  *  ENTER FROM D2                                          71604130
173  00 A899   454  AI       L     YDL,IO,ACK        (R1)= DEVICE NUMBER       71604140
               455  *  SENSE  STATUS                                          71604150
               456  *  ENTER FROM D2                                          71604160
               457  * CONTINUE FOR AI                                         71604170
174  08 C835   458  SS       L     IO,YDL,ADRS+MR    ADDRESS THE DEVICE        71604180
175  00 AED6   459           L     MDR,IO,STAT+CS    INPUT STATUS              71604190
```

```
176  0E B60F   460          L     FLR,MDR,CS+MW      (FLR)=LS 4 BITS OF STAT.        71604200
177  F0 0294   461          B     FETCHJ             CCOPY FLR TO CC AND FETCH       71604210
               462  *                               NEXT INSTRUCTION                71604220
               463  *  OUTPUT  COMMAND  REGISTER                                    71604230
178  00 C835   464  OCR    L     IO,YDL,ADRS        ADDRESS THE DEVICE              71604240
179  05 E875   465          L     IO,YSL,OC+IRJH     OUTPUT COMMAND:START INST.FETCH 71604250
               466  *                               COPY FLR TO CC                  71604260
17A  00 8010   467          L     NULL,NULL          NOP                             71604270
               468  * OUTPUT COMMAND                                               71604280
               469  * ENTER FROM D2                                                71604290
17B  08 C835   470  OC     L     IO,YDL,ADRS+MR     ADDRESS THE DEVICE              71604300
17C  00 B675   471          L     IO,MDR,CS+OC       OUT PUT COMMAND                 71604310
17D  F0 0294   472          B     FETCHJ             FETCH NEXT INSTRUCTION          71604320
               473  *  WRITE HALFWORD REGISTER                                     71604330
               474  *  ENTER AFTER D1                                              71604340
17E  03 C835   475  WHR    L     IO,YDL,ADRS+D2     ADDRESS THE DEVICE              71604350
17F  00 E816   476          L     MDR,YSL            (MDR) = LOW 2ND OPERAND         71604360
               477  *  WRITE HALFWORD                                              71604370
               478  *  ENTER AFTER D1                                              71604380
180  40 F417   479  WH     A     MAR,YSLX,MDR       (MAR)=2ND OPERAND ADDR.         71604390
181  08 8010   480  WH1    L     NULL,NULL,MR       START MEM. READ                 71604400
182  03 C835   481          L     IO,YDL,ADRS+D2     ADDRESS THE DEVICE              71604410
183  00 A017   482          L     MAR,LOC                                            71604420
               483  * COME HERE (AFTER D2) FOR RH  IF DEVICE IS HALFWORD ORIENTED   71604430
184  0E A8B6   484  RHH    L     MDR,IO,DR+MW       INPUT A HW AND STORE IT         71604440
185  F0 0294   485          B     FETCHJ             CCOPY FLR TO COND. CODE AND FETCH 71604450
               486  *                               NEXT INSTRUCTION                71604460
               487  * COME HERE (AFTER D2) FOR RH IF THE DEVICE IS BYTE ORIENTED    71604470
186  00 AEA1   488  RHB    L     MR1,IO,DR+CS       INPUT HIGH BYTE                 71604480
187  00 A8A9   489          L     ARL,IO,DR          INPUT LOW BYTE                  71604490
188  2E 0A16   490          O     MDR,MR1,ARL,MW     COMBINE,AND STORE TWO BYTES     71604500
189  F0 0294   491          B     FETCHJ             CCOPY FLR TO CC AND FETCHETCH   71604510
               492  *  READ HALFWORD REGISTER (ENTER AFTER D1)                      71604520
18A  03 C835   493  RHR    L     IO,YDL,ADRS+D2     ADDRESS THE DEVICE              71604530
18B  00 8010   494          L     NULL,NULL          NOP                             71604540
               495  *  COME  HERE  (AFTER D2) FOR RHR IF DEVICE IS HALFWORD ORIENTED 71604550
18C  05 A8BD   496  RHRH   L     YSL,IO,DR+IRJH     INPUT A HALFWORD:INST.FETCH:COPY 71604560
18D  00 8010   497          L     NULL,NULL          FLR TO COND. CODE               71604570
               498  * COME  HERE (AFTER D2) FOR RHR IF DEVICE IS BYTE ORIENTED     71604580
18E  00 AEBD   499  RHRB   L     YSL,IO,DR+CS       INPUT HIGH BYTE                 71604590
18F  00 A8A9   500          L     ARL,IO,DR          INPUT LOW BYTE                  71604600
190  25 EA1D   501          O     YSL,YSL,ARL,IRJH   COMBINE LOW AND HIGH BYTES      71604610
191  00 8010   502          L     NULL,NULL                                          71604620
               503  * READ HALFWORD  (ENTER AFTER D1)                             71604630
192  03 C835   504  RH     L     IO,YDL,ADRS+D2     ADDRESS THE DEVICE              71604640
193  40 F417   505  RH1    A     MAR,YSLX,MDR       (MAR) = SECOND OPERAND ADDRESS  71604650
               506  *                               NEXT INSTRUCTION                71604670
               507  * COMMON FOR  WHR AND WH                                       71604680
               508  * ENTER AFTER D2 FOR HW DEVICES (ONLY)                         71604690
194  05 B055   509  WHH    L     IO,MDR,DA+IRJH     OUTPUT DATA:INST. FETCH:COPY FLR TO CC 71604700
195  00 8010   510          L     NULL,NULL          NOP                             71604710
               511  *                                                              71604720
               512  * COMMON FOR WHR & WH. ENTER AFTER D2 (FOR BYTE ORIENTED DEVICES) 71604730
               513  *                                                              71604740
```

```
196       00 B655            514  WHB      L     IO,MDR,DA+CS        OUTPUT HIGH                                71604750
197       05 B055            515           L     IO,MDR,DA+IRJH      OUTPUT LOW BYTE; START INST.               71604760
                             516  *                                  FETCH;COPY FLR TO CC                       71604770
198       00 8010            517           L     NULL,NULL                                                      71604780
                             518  *  COMMON ROUTINE FOR READ BLOCK AND WRITE BLOCK                               71604790
                             519  * ENTER AFTER  D1                                                             71604800
199       40 F417            520  RWB      A     MAR,YSLX,MDR        (MAR)=SECOND OPERAND ADDRESS               71604810
19A       09 8010            521           L     NULL,NULL,MR2       START  MR & INC.MAR BY 2                   71604820
19B       00 C835            522           L     IO,YDL,ADRS         ADDRESS THE DEVICE                         71604830
19C       08 8009            523           L     ARL,MDR,MR          (ARL)=START ADDRESS                        71604840
19D       43 8217            524           A     MAR,NULL,ARL,D2     (MAR)=START ADDR.;VECTOR THRU D2            71604850
19E       50 B241            525           S     MR1,MDR,ARL,CO      (MR1)=END ADDR. -START ADDR.               71604860
                             526  * COMMON ROUTINE FOR RBR AND WBR                                             71604870
                             527  * ENTER   AFTER D1                                                           71604880
19F       00 C835            528  RWBR     L     IO,YDL,ADRS         ADDRESS THE DEVICE                         71604890
1A0       00 E80B            529           L     AR,YSL              (AR)=START ADDRESS                         71604900
1A1       40 610C            530           A     YSI,YSI,ONE         INCR. YS FIELD BY 1                        71604910
1A2       53 EA41            531           S     MR1,YSL,ARL,CO+D2   (MR1)=END ADR.-START ADR.;               71604920
                             532  *                                  VECTOR THROUGH D2                          71604930
1A3       40 8217            533           A     MAR,NULL,ARL        (MAR)=START  ADDRESS                       71604940
                             534  * COMMON ROUTINE FOR RB & RBR                                                 71604950
                             535  * ENTER AT 'STARTR' AFTER VECTORING THROUGH DROM2                             71604960
                             536  * (MR1)=END ADDRESS-START ADDRESS;C IS SET IF END ADR.<START ADR.            71604970
1A4       08 8010            537  LOOPR    L     NULL,NULL,MR        FETCH HALFWORD                             71604980
1A5       00 A8CF            538  STATR    L     FLR,IO,STAT         (FLR) 4 LS 4 BITS OF STATUS                71604990
1A6       E3 C1B5            539           BT    C+V+G+L,TESTR       TEST STATUS IF NOT ZERO                    71605000
1A7       0E AEB6            540           L     MDR,IO,DR+CS+MW     INSERT BYTE;RESTORE HALFWORD               71605010
1A8       50 0941            541  LOOPR1   S     MR1,MR1,ONE,CO      (MR1)=(MR1)-1                              71605020
1A9       40 B917            542           A     MAR,MAR,ONE         INCR.  MAR BY ONE                          71605030
1AA       F2 01A4            543  STARTR   BF    C,LOOPR             REPEAT IF TRANSFER IS NOT COMPLETE         71605040
1AB       F0 01B5            544           B     FINIS                                                          71605050
                             545  *  COMMON ROUTINE  FOR WB & WBR                                              71605060
                             546  * ENTER (AFTER D2) AT STARTW                                                  71605070
1AC       08 8010            547  LOOPW    L     NULL,NULL,MR        START MEM. READ                            71605080
1AD       00 A8CF            548  STATW    L     FLR,IO,STAT         (FLR) = 4 LS BITS OF STATUS                71605090
1AE       E3 C1B7            549           BT    C+V+G+L,TESTW       TEST STATUS IF NOT ZERO                    71605100
1AF       00 B655            550           L     IO,MDR,DA+CS        OUTPUT A BYTE                              71605110
1B0       50 0941            551           S     MR1,MR1,ONE,CO      (MR1)=(MR1)-1                              71605120
1B1       40 B917            552           A     MAR,MAR,ONE         INCR. MAR BY 1                             71605130
1B2       F2 01AC            553  STARTW   BF    C,LOOPW             REPEAT IF NOT DONE                         71605140
1B3       00 800F            554  FINIS    L     FLR,NULL            CLEAR FLR                                  71605150
1B4       F0 0294            555  FINIS1   B     FETCHJ              COPY FLR TO CC AND FETCH                   71605160
                             556  *                                  NEXT INSTRUCTION                           71605170
1B5       E1 C1B4            557  TESTR    BT    V+G+L,FINIS1        BAD STATUS                                 71605180
1B6       F0 01A5            558           B     STATR               JUST BUSY                                  71605190
1B7       E1 C1B4            559  TESTW    BT    V+G+L,FINIS1        BAD STATUS                                 71605200
1B8       F0 01AD            560           B     STATW               JUST BUSY                                  71605210
                             561  *  AUTO LOAD                                                                   71605220
                             562  *  ENTER FROM DROM2                                                          71605230
1B9       88 0009            563  AL       LI    ARL,'80'                                                       71605240
1BA       50 BA41            564           S     MR1,MAR,ARL,CO      (MR1)=END ADDRESS-START ADDRESS            71605250
1BB       87 8017            565           LI    MAR,'78'                                                       71605260
1BC       48 8217            566           A     MAR,NULL,ARL,MR     (MAR)='80';FETCH DEV. NO.                  71605270
1BD       E2 01B3            567           BT    C,FINIS                                                        71605280
```

```
1BE  00 B635  568          L    IO,MDR,CS+ADRS     ADDRESS THE DEVICE                          71605290
1BF  08 B075  569          L    IO,MDR,OC+MR       OUTPUT COMMAND START MEM READ               71605300
1C0  00 A8CF  570 INAL     L    FLR,IO,STAT        (FLR)= STATUS (4:7)                         71605310
1C1  E1 C1B4  571          BT   V+G+L,FINIS1       BAD STATUS                                  71605320
1C2  E2 01C0  572          BT   C,INAL             JUST BUSY                                   71605330
1C3  00 A8A6  573          L    MR6,IO,DR          READ A BYTE                                 71605340
1C4  00 3616  574          L    MDR,MR6,CS         INSERT THE BYTE                             71605350
1C5  00 3030  575          L    NULL,MR6,F         TEST INPUT BYTE                             71605360
1C6  F0 C1C0  576          BF   G+L,INAL           SKIP LEADER                                 71605370
              577 * A NONZERO CHARACTER HAS BEEN READ                                          71605380
1C7  7E 0000  578          C    MW                 STORE                                       71605390
1C8  F0 01A8  579          B    LOOPR1             BRANCH TO LOOPR1 ( A ROUTINE USED BY        71605400
              580 *                                                                            71605410
              581 *                                                                            71605420
              582 *                                                                            71605430
              583 *                                                                            71605440
              584 *                                                                            71605450
              585 *                                                                            71605460
              586 *                                                                            71605470
              587 * I/O CHANNEL                                                               71605480
              588 * (MAR)=SERVICE POINTER                                                     71605490
              589 * (MR1)=DEVICE NUMBER                                                      71605500
1C9  00 B80A  590 CHANEL   L    ARH,MAR            (ARH)=CHANNEL COMMAND WORD ADDRESS          71605510
1CA  09 0E01  591          L    MR1,MR1,CS+MR2     MR1<0:7>=DEV.NO.                            71605520
              592 *                                FETCH CCW                                   71605530
1CB  00 0E35  593          L    IO,MR1,CS+ADRS     ADDRESS THE DEVICE                          71605540
1CC  08 B002  594          L    MR2,MDR,MR         (MR2)=CCW;FETCH START ADDR OR COUNT         71605550
1CD  00 1612  595          L    SRL,MR2,CS         SRL<8:15>=CCW<0:7> ;SRL<0:7>=BITS           71605560
              596 *                                8 TO 15 OF CCW                              71605570
1CE  70 0C00  597          C    SR2                                                            71605580
1CF  70 0C00  598          C    SR2                SRL<12:15>=CCW<0:3>;SRL<4:11>=BITS          71605590
              599 *                                8:15 OF CCW                                 71605600
1D0  00 900F  600          L    FLR,SRL            (FLR)= CCW BIT 0:3                          71605610
1D1  E3 82DF  601          BT   C+V+G,NI01         BRANCH TO NI01 IF ANY OF THE BITS           71605620
              602 *                                0 THRU 2 IS SET                             71605630
              603 * READ OR WRITE FUNCTION                                                    71605640
1D2  00 A8C9  604 RDWRT    L    ARL,IO,STAT        SENSE STATUS                                71605650
1D3  20 0A01  605          O    MR1,MR1,ARL        (MR1)=DEV.NO./STATUS                        71605660
1D4  00 080F  606          L    FLR,MR1            (FLR) = BITS 4:7 OF STATUS                  71605670
1D5  E3 C3DD  607          BT   C+V+G+L,BSTAT      BRANCH TO BSTAT IF BAD STATUS               71605680
              608 * DEVICE STATUS IS O.K.                                                    71605690
              609 * READ OR WRITE FUNCTION                                                   71605700
1D6  00 B003  610          L    MR3,MDR            (MR3)=START ADDRESS                         71605710
1D7  50 1109  611          S    ARL,MR2,ONE        ARL<12:15>=BYTE COUNT - 1                   71605720
1D8  90 F208  612          NI   CTR,'F',ARL        (COUNTER)=BYTE COUNT -1                     71605730
              613 *                                (COUNTER)=15 IF BYTE COUNT IS ZERO          71605740
1D9  90 F209  614          NI   ARL,'F',ARL                                                    71605750
1DA  C0 1209  615          AI   ARL,'1',ARL        (ARL)=BYTE COUNT (16 IF BYTE CNT=0)         71605760
1DB  40 B216  616          A    MDR,MDR,ARL        (MDR)=INCREMENTED START ADDRESS             71605770
1DC  7F 0090  617          C    MW2+PRIV           STORE INCREMENTED START ADDRESS             71605780
1DD  08 1817  618          L    MAR,MR3,MR         FETCH END ADDRESS ;(MAR)=START ADDR.        71605790
1DE  00 900F  619          L    FLR,SRL            (FLR)=CCW BITS  0:3                         71605800
1DF  00 B004  620          L    MR4,MDR            (MR4) = END ADDRESS                         71605810
1E0  E0 42C1  621          BT   L,WRITE                                                        71605820
```

```
                        622  * READ FUNCTION                                                    71605830
1E1      78 0000        623  READ     C     MR                 READ FROM MEMORY               71605840
1E2      00 AEB6        624           L     MDR,IO,DR+CS       INSERT DATA BYTE               71605850
1E3      7E 0090        625           C     MW+PRIV            STORE                          71605860
1E4      40 B917        626           A     MAR,MAR,ONE        INCREMENT MAR BY 1             71605870
1E5      FC 81E1        627           BF    CNTR,READ          REPEAT UNTIL SPECIFIED NUMBER  71605880
                        628  *                                  OF BYTES HAVE BEEN READ        71605890
1E6      F0 02C5        629           B     END1                                              71605900
1E7                     630           ORG   '1FF'                                             71605904
                        631  * A PRIV. INST. HAS BEEN ENCOUNTERED IN PROTECT MODE              71605906
1FF      F0 0003        632           B     PRIV                                              71605908
```

```
200                     ORG    '200'                                          71605910
                  * COMMON ROUTINE  FOR SLHL,SRHL,SLHA,AND SRHA INSTRUCTIONS  71605920
                  * ENTER  AFTER  VECTORING THROUGH DROM1                     71605930
200   43 F409     HWSHFT  A     ARL,YSLX,MDR,D2     (ARL) = LOW SECOND OPERAND  71605940
                  *                                 VECTOR THROUGH DROM2        71605950
201   90 F208             NI    CTR,X'F',ARL        (COUNTER) = SHIFT COUNT     71605960
                  * SHIFT RIGHT LOGICAL (HALF WORD) SHORT                     71605970
202   00 6008     SRLS    L     CTR,YSI             (COUNTER)=SHIFT COUNT       71605980
                  * COMMON FOR SRLHS AND SRHL INSTRUCTIONS                    71605990
                  * FOR SRHL ENTER AFTER VECTORING THROUGH DROM2              71606000
203   70 0008     SRHL    C     RPT                 REPEAT NEXT INST. (COUNTER) TIMES  71606010
204   00 CC59             L     YDL,YDL,SR+CO       SHIFT RIGHT 1 PLACE; MODIFY C      71606020
205   05 C839             L     YDL,YDL,F+IRJH      MODIFY  G&L; FETCH NEXT            71606030
                  *                                 INSTR. AND COPY FLR TO CC          71606040
                  * SHIFT LEFT LOGICAL (HALF WORD) SHORT                      71606050
206   00 6008     SLLS    L     CTR,YSI             (COUNTER) = SHIFT COUNT     71606060
                  * COMMON FOR SLLHS AND SLHL INSTRUCTIONS                    71606070
                  * FOR     SLHL  ENTER AFTER D2                              71606080
207   70 0008     SLHL    C     RPT                 REPEAT NEXT INSTR. (COUNTER) TIMES  71606090
208   00 CA59             L     YDL,YDL,SL+CO       SHIFT LEFT ONE PLACE; MODIFY C FLAG  71606100
209   05 C839             L     YDL,YDL,F+IRJH      MODIFY G&L;FETCH NEXT               71606110
20A   00 8010             L     NULL,NULL           INSTR. AND COPY FLR TO CC           71606120
                  *  SHIFT LEFT  HALFWORD ARITHMETIC                          71606130
20B   00 CA59     SLHA    L     YDL,YDL,SL+CO       SIGN TO CARRY               71606140
20C   00 84C9             L     ARL,NULL,SR+CI+CO   ARL(0) = SIGN               71606150
20D   70 0008             C     RPT                 REPEAT NEXT INSTR. (COUNTER TIMES)  71606160
20E   00 CA59             L     YDL,YDL,SL+CO       SHIFT LEFT ONE PLACE;MODIFY C FLAG  71606170
20F   05 CC19             L     YDL,YDL,SR+IRJH     COMPENSATE FOR PRE-SHIFT    71606180
                  *                                START INST. FETCH            71606190
210   20 CA59             O     YDL,YDL,ARL,F       RESTORE SIGN ; MOD. FLR AND COPY  71606200
                  *                                 IT TO CONDITION CODE        71606210
                  * SHIFT   RIGHT  HALFWORD ARITHMETIC                        71606220
211   00 CA59     SRHA    L     YDL,YDL,SL+CO       MOVE SIGN TO CARRY FLAG     71606230
212   70 0008             C     RPT                 REPEAT NEXT INSTR. (COUNTER) TIMES  71606240
213   00 CC99             L     YDL,YDL,SR+CI       SHIFT RIGHT ONE PLACE;SHIFT IN CARRY BI71606250
214   05 CCF9             L     YDL,YDL,SR+CO+CI+F+IRJH  COMPENSATE PRESHIFT;MOD. CARRY  71606260
                  *                                      AND FLAGS;FETCH NEXT INSTR. &  71606270
                  *                                      COPY FLR TO HW CC         71606280
                  * COMMON FOR    SRL,SLL,SRA,SLA,RRL,RLL                     71606290
                  * ENTER   FROM   D1                                         71606300
215   00 D013     RTSHFT  L     SRH,YDLP1           (SRH,SRL)=(R1);(COUNTER)=SHIFT COUNT  71606310
216   43 F408             A     CTR,YSLX,MDR,D2     VECTOR THROUGH DROM2        71606320
217   00 C812             L     SRL,YDL                                         71606330
                  * SHIFT RIGHT LOGICAL                                       71606340
218   70 0008     SRL     C     RPT                 REPEAT NEXT INST. COUNTER TIMES  71606350
219   70 0840             C     SR1+CO              SHIFT RIGHT ONE PLACE       71606360
21A   05 903B             L     YDLM1,SRL,F+IRJH    STORE  THE  RESULT          71606370
21B   00 9839             L     YDL,SRH,F                                       71606380
                  * SHIFT   LEFT  LOGICAL                                     71606390
21C   70 0008     SLL     C     RPT                 REPEAT NEXT INSTR.  (COUNTER)TIMES  71606400
21D   70 1040             C     SL1+CO              SHIFT LEFT ONE PLACE        71606410
21E   05 903B             L     YDLM1,SRL,F+IRJH    STORE  THE RESULT           71606420
21F   00 9839             L     YDL,SRH,F                                       71606430
                  *  SHIFT RIGHT ARITHMETIC                                   71606440
```

```
220   70 1040          688  SRA     C   SL1+CO          SHIFT (SRH,SRL) LEFT   ONE PLACE        71606450
221   70 0008          689          C   RPT             REPEAT NEXT INST. (COUNTER)TIMES        71606460
222   70 0A00          690          C   SR1+CI          SHIFT RIGHT ONE PLACE                   71606470
223   70 0A40          691          C   SR1+CI+CO       COMPENSATE PRESHIFT                     71606480
224   05 903B          692          L   YDLM1,SRL,F+IRJH STORE  THE  RESULT AND FETCH NEXT     71606490
225   00 9839          693          L   YDL,SRH,F       INSTRUCTION                             71606500
                       694  * SHIFT LEFT ARITHMETIC                                          71606510
226   70 1040          695  SLA     C   SL1+CO          SHIFT LEFT  ONE PLACE                   71606520
227   00 84C9          696          L   ARL,NULL,SR+CI+CO SAVE  SIGN  IN ARL                   71606530
228   70 0008          697          C   RPT             REPEAT NEXT INSTR. (COUNTER) TIMES      71606540
229   70 1040          698          C   SL1+CO          SHIFT LEFT 1 PLACE                      71606550
22A   70 0800          699          C   SR1             COMPENSATE  PRESHIFT                    71606560
22B   05 903B          700          L   YDLM1,SRL,F+IRJH STORE THE RESULT  WITH SIGN ;         71606570
22C   20 9A39          701          U   YDL,SRH,ARL,F   FETCH  NEXT  INSTRUCTION              71606580
                       702  *  ROTATE RIGHT LOGICAL                                          71606590
22D   EC 8231          703  RRL     BT  CNTR,RRX1       NO ROTATION  IFF(COUNTER)=0          71606600
22E   00 9450          704  RRL2    L   NULL,SRL,SR+CO   (CARRY)= LSB OF  FIRST  OPERAND     71606610
22F   70 0A40          705          C   SR1+CO+CI       ROTATE  RIGHT ONE PLACE              71606620
230   FC 822E          706          BF  CNTR,RRL2       REPEAT UNTIL (COUNTER)= 0            71606630
231   05 907B          707  RRX1    L   YDLM1,SRL,F+CO+IRJH (DEST.)=(SRH,SRL);MODIFY G&L;RESET 71606640
232   00 9879          708          L   YDL,SRH,F+CO    C;FETCH NEXT INST.;COPY FLR TO CC       71606650
                       709  * ROTATE LEFT LOGICAL                                            71606660
233   EC 8237          710  RLL     BT  CNTR,RLX1       NO ROTATION  IF (COUNTER)=0          71606670
234   00 9A50          711  RLL2    L   NULL,SRH,SL+CO  (CARRY)= MSB                         71606680
235   70 1240          712          C   SL1+CO+CI       ROTATE LEFT ONE  PLACE               71606690
236   FC 8234          713          BF  CNTR,RLL2       REPEAT UNTIL (COUNTER)=0             71606700
237   05 907B          714  RLX1    L   YDLM1,SRL,F+CO+IRJH STORE THE RESULT INTO DEST. REGISTER 71606710
238   00 9879          715          L   YDL,SRH,F+CO    START INST. FETCH;COPY FLR TO CC        71606720
                       716  * ENTER HERE(AFTER VECTORING THROUGH DROM1) FOR THE FOLLOWING INSTR. 71606730
                       717  * LH,AH,SH,NH,OH,XH,CLH                                          71606740
                       718  *                                                                71606750
239   40 F417          719  RX      A   MAR,YSLX,MDR    (MAR)=SECOND OPERAND ADDRESS         71606760
23A   08 A017          720          L   MAR,LOC,MR     (MAR)=(LOC);FETCH OPERAND            71606770
23B   03 8010          721          L   NULL,NULL,D2    VECTOR THROUGH DROM2                 71606780
23C   05 B009          722          L   ARL,MDR,IRJH    (ARL)= SECOND OPERAND ADDRESS        71606790
                       723  *                           START INST. READ                    71606800
                       724  * COMMON FOR   NHI,OHI,XHI,CLHI,LHI,AHI,SHI,THI                  71606810
                       725  * METER  FROM  DROM1                                            71606820
23D   43 F409          726  RI      A   ARL,YSLX,MDR,D2  (ARL)=2ND OPERAND;VECTOR THRU DROM2 71606830
23E   05 8010          727          L   NULL,NULL,IRJH   START INSTR. FETCH                  71606840
                       728  * ENTER  FROM  DROM1                                           71606850
                       729  * COMMON FOR STH,STB,LB,CLB,AHM,RD,WD,SS,OC,BAL,LPSW,AL,AI,SVC 71606860
23F   03 8010          730  RXA     L   NULL,NULL,D2    VECTOR THROUGH DROM2                 71606870
240   40 F417          731          A   MAR,YSLX,MDR    (MAR)=SECOND  OPERAND ADDRESS        71606880
                       732  * BRANCH ON TRUE CONDITION REGISTER (ENTER FROM DROM1)           71606890
241   F0 224F          733  BTCR    BF  MSK1,FETCH      FETCH NEXT INST. IF COND. IS FALSE   71606900
242   40 8214          734          A   LOC,NULL,ARL    NLOC)=(MAR)=BRANCH ADDRESS           71606910
243   04 8010          735          L   NULL,NULL,IR    FETCH INSTRUCTION                    71606920
244   00 8010          736          L   NULL,NULL       NOP                                  71606930
                       737  *                                                                71606940
                       738  *                                                                71606950
                       739  * BRANCH ON FALSE CONDITION REGISTER                             71606960
                       740  * ENTER  FROM  D1                                               71606970
245   E0 224F          741  BFCR    BT  MSK1,FETCH      FETCH NEXT INST. IF COND. IS TRUE    71606980
```

```
246  40 8214   742          A     LOC,NULL,ARL       (LOC)=(MAR)=BRANCH ADDRESS          71606990
247  04 8010   743          L     NULL,NULL,IR       FETCH INSTRUCTION                   71607000
248  00 8010   744          L     NULL,NULL          NOP                                 71607010
               745  *                                                                    71607020
               746  *                                                                    71607030
               747  * BRANCH ON TRUE CONDITION                                          71607040
               748  * ENTER  FROM  DROM1                                                71607050
249  F0 224F   749  BTC      BF    MSK1,FETCH         FETCH NEXT INST. IF COND. IS FALSE  71607060
24A  40 F414   750          A     LOC,YSLX,MDR       (LOC)=(MAR)=BRANCH ADDRESS          71607070
24B  04 8010   751          L     NULL,NULL,IR       FETCH INSTRUCTION                   71607080
24C  00 8010   752          L     NULL,NULL          NOP                                 71607090
               753  *                                                                    71607100
               754  *                                                                    71607110
               755  * BRANCH ON FALSE CONDITION (ENTER FROM DROM1)                       71607120
24D  E0 224F   756  BFC      BT    MSK1,FETCH         FETCH NEXT INST. IF COND. IS TRUE   71607130
24E  40 F414   757          A     LOC,YSLX,MDR       (LOC)=(MAR)=BRANCH ADDRESS          71607140
               758  * THIS ROUTINE FETCHES NEXT INSTRUCTION                            71607150
24F  04 8010   759  FETCH    L     NULL,NULL,IR       START INSTRUCTION READ              71607160
250  00 8010   760          L     NULL,NULL          NOP                                 71607170
               761  *                                                                    71607180
               762  *                                                                    71607190
               763  * BRANCH ON TRUE BACKWARD SHORT                                     71607200
               764  * ENTER FROM D1                                                     71607210
251  F0 224F   765  BTBS     BF    MSK1,FETCH         FETCH NEXT INST. IF COND. IS FALSE  71607220
252  50 A314   766          S     LOC,LOC,TWO        DECR. LOC BY TWO                    71607230
253  00 6209   767          L     ARL,YSI,SL         (ARL) = 2 * HW DISPLACEMENT         71607240
254  50 A214   768          S     LOC,LOC,ARL        (LOC)=(MAR)=BRANCH ADDRESS          71607250
255  04 8010   769          L     NULL,NULL,IR       FETCH INSTRUCTION                   71607260
256  00 8010   770          L     NULL,NULL          NOP                                 71607270
               771  *                                                                    71607280
               772  *                                                                    71607290
               773  * BRANCH ON TRUE FORWARD SHORT                                      71607300
               774  * ENTER FROM DROM1                                                  71607310
257  F0 224F   775  BTFS     BF    MSK1,FETCH         FETCH NEXT INST. IF COND. IS FALSE  71607320
258  50 A314   776          S     LOC,LOC,TWO        DECR. LOC BY TWO                    71607330
259  00 6209   777          L     ARL,YSI,SL         (ARL)=2* HW DISPLACEMENT            71607340
25A  40 A214   778          A     LOC,LOC,ARL        (LOC)=(MAR)=BRANCH ADDRESS          71607350
25B  04 8010   779          L     NULL,NULL,IR       FETCH INSTRUCTION                   71607360
25C  00 8010   780          L     NULL,NULL          NOP                                 71607370
               781  *                                                                    71607380
               782  *                                                                    71607390
               783  * BRANCH ON FALSE BACKWARD SHORT                                    71607400
               784  * ENTER FROM DROM1                                                  71607410
25D  E0 224F   785  BFBS     BT    MSK1,FETCH         FETCH NEXT INST. IF COND. IS TRUE   71607420
25E  50 A314   786          S     LOC,LOC,TWO        DECR. LOC BY 2                      71607430
25F  00 6209   787          L     ARL,YSI,SL         (ARL)=2* HW DISPLACEMENT            71607440
260  50 A214   788          S     LOC,LOC,ARL        (LOC)=(MAR)=BRANCH ADDRESS          71607450
261  04 8010   789          L     NULL,NULL,IR       FETCH NEXT INSTRUCTION              71607460
262  00 8010   790          L     NULL,NULL          NOP                                 71607470
               791  *                                                                    71607480
               792  *                                                                    71607490
               793  * BRANCH ON FALSE FORWARD HALFWORD SHORT                            71607500
               794  * ENTER  FROM  DROM1                                                71607510
263  E0 224F   795  BFFS     BT    MSK1,FETCH         FETCH NEXT INST. IF COND. IS TRUE   71607520
```

```
                                  796           S     LOC,LOC,TWO         DECR. LOC BY 2                                  71607530
264      50 A314                  797           L     ARL,YSI,SL          (ARL)=2*HW DISPLACEMENT                         71607540
265      00 6209                  798           A     LOC,LOC,ARL         (LOC)=(MAR)=BRANCH ADDRESS                      71607550
266      40 A214                  799           L     NULL,NULL,IR        FETCH INSTRUCTION                               71607560
267      04 8010                  800           L     NULL,NULL           NOP                                             71607570
268      00 8010                  801   *                                                                                 71607580
                                  802   * BRANCH AND LINK REGISTER (ENTER FROM DROM1)                                    71607590
269      00 E817                  803   BALR    L     MAR,YSL             (MAR) = (R2)                                    71607600
                                  804   * COMMON FOR BALR AND BAL (ENTER FROM DROM2 FOR BAL)                            71607610
                                  805   * MAR  CONTAINS BRANCH ADDRESS                                                  71607620
26A      04 A019                  806   BAL     L     YDL,LOC,IR          (R1)=(LOC) ;                                    71607630
26B      00 B814                  807           L     LOC,MAR             UPDATE LOC AND FETCH NEXT INST.                 71607640
                                  808   * COMMON FOR BXH AND BXLE                                                       71607650
                                  809   * ENTER  FROM  D1                                                              71607660
26C      40 F417                  810   BXHLE   A     MAR,YSLX,MDR        (MAR)=SECOND OPERAND ADDRESS                    71607670
26D      00 D009                  811           L     ARL,YDLP1           (ARL) = (R1)                                    71607680
26E      40 690C                  812           A     YSI,YDI,ONE         YS FIELD =R1+1                                  71607690
26F      40 DA09                  813           A     ARL,YDLM1,ARL       (ARL)=(R1)+(R1+1)                               71607700
270      43 8219                  814           A     YDL,NULL,ARL,D2     (R1)=(R1)+(R1+1);VECTOR THRU D2                 71607710
271      50 EA50                  815           S     NULL,YSL,ARL,CO     SET CARRY IF INCR. INDEX>FINAL INDEX            71607720
                                  816   * BRANCH ON INDEX HIGH                                                          71607730
                                  817   * ENTER  FROM DROM2                                                             71607740
                                  818   *                                                                               71607750
272      F2 0279                  819   BXH     BF    C,NBR               NO BRANCH IF INDEX IS NOT>FINAL IND             71607760
273      04 B814                  820           L     LOC,MAR,IR          TAKE BRANCH                                     71607770
274      00 8010                  821           L     NULL,NULL           NOP                                             71607780
                                  822   * BRANCH  ON INDEX LESS THAN OR EQUAL (BXLE)                                    71607790
                                  823   * ENTER FROM D2                                                                 71607800
275      E2 0279                  824   BXLE    BT    C,NBR               NO BRANCH,IF INDEX IS > FINAL VALUE             71607810
276      04 B814                  825           L     LOC,MAR,IR          TAKE BRANCH                                     71607820
277      00 8010                  826           L     NULL,NULL           NOP                                             71607830
                                  827   * STORE HALF WORD (ENTER FROM DROM2)                                            71607840
                                  828   *                                                                               71607850
278      0E C816                  829   STH     L     MDR,YDL,MW          STORE  (R1)                                     71607860
                                  830   * THIS ROUTINE FETCHES INSTR. AT ADDRESS SPECIFIED BY LOC                       71607870
279      00 A017                  831   NBR     L     MAR,LOC             (MAR)=(LOC)                                     71607880
27A      04 8010                  832           L     NULL,NULL,IR        START INST. READ                                71607890
27B      00 8010                  833           L     NULL,NULL           NOP                                             71607900
                                  834   * NH AND NHI (ENTER FROM D2)                                                    71607910
                                  835   * NHR (ENTER FROM D1)                                                           71607920
                                  836   * INSTRUCTION READ IS IN PROGRESS                                               71607930
                                  837   *                                                                               71607940
27C      10 CA39                  838   NH      N     YDL,YDL,ARL,F       (R1) =(R1)  AND  (ARL);MODIFY FLAGS             71607950
                                  839   *                                                                               71607960
                                  840   * OH AND OHI (ENTER FROM D2)                                                     71607970
                                  841   * OHR (ENTER FROM D1)                                                           71607980
                                  842   *                                                                               71607990
27D      20 CA39                  843   OH      O     YDL,YDL,ARL,F       (R1)=(R1) OR ARL) ;MODIFY FLAGS                 71608000
                                  844   *                                                                               71608010
                                  845   * XH AND XHI (FROM D2)                                                          71608020
                                  846   * XHR (FROM D1)                                                                 71608030
                                  847   *                                                                               71608040
27E      30 CA39                  848   XH      X     YDL,YDL,ARL,F       (R1)=(R1) EXCLUSIVE OR (ARL)                    71608050
                                  849   *                                 MODIFY FLAGS                                    71608060
```

```
                                850  *                                                                          71608070
                                851  * CLH AND CLHI (ENTER FROM DROM2)                                         71608080
                                852  * CLHR (ENTER FROM DROM1)                                                 71608090
                                853  *                                                                          71608100
27F     50 CA70                 854  CLH       S     NULL,YDL,ARL,CO+F   (R1) - (ARL) ; MODIFY FLAGS & CARRY   71608110
                                855  *                                                                          71608120
                                856  *                                                                          71608130
                                857  * LH AND LHI (ENTER FROM D2)                                             71608140
                                858  * LHR (ENTER FROM D1)                                                    71608150
                                859  *                                                                          71608160
280     40 8239                 860  LH        A     YDL,NULL,ARL,F      (R1)= (ARL); MODIFY FLAGS             71608170
                                861  *                                                                          71608180
                                862  * LIS (ENTER FROM D1)                                                    71608190
                                863  *                                                                          71608200
281     00 8010                 864            L     NULL,NULL           FILLER(NOT USED)                      71608210
                                865  *                                                                          71608220
                                866  * LCS (ENTER FROM D1)                                                    71608230
                                867  *                                                                          71608240
282     05 6009                 868  LCS       L     ARL,YSI,IRJH        (ARL)=SHORT IMMEDIATE OPERAND         71608250
283     50 8239                 869            S     YDL,NULL,ARL,F      (R1) = NEGATIVE OF IMM. OPERAND       71608260
                                870  *                                   MODIFY G&L FLAGS                      71608270
                                871  *                                                                          71608280
                                872  * CH AND CHI (ENTER FROM DROM2)                                          71608290
                                873  * CHR (ENTER FROM DROM1)                                                 71608300
284     30 CA50                 874  CH        X     NULL,YDL,ARL,F      SET L FLAG IF SIGNS ARE DIFFERENT     71608310
285     E0 4288                 875            BT    L,DIFFER                                                  71608320
                                876  * SIGNS OF OPERANDS ARE SAME                                            71608330
                                877  * SUBTRACT 2ND OPERAND FROM 1ST OPERANDS TO SET FLAGS                    71608340
                                878  *                                                                          71608350
286     05 800F                 879            L     FLR,NULL,IRJH       CLEAR FLAG REG.&START INST.FETCH      71608360
287     50 CA70                 880            S     NULL,YDL,ARL,CO+F   SET CARRY & FLAGS TO INDICATE COMPAR  71608370
                                881  * SIGNS OF OPERANDS ARE DIFFERENT                                       71608380
288     05 CA50                 882  DIFFER    L     NULL,YDL,SL+CO+IRJH SET FLAGS & CARRY TO INDICATE         71608390
289     00 C830                 883            L     NULL,YDL,F          COMPARISON;START INST. FETCH          71608400
                                884  * AIS  (ENTER FROOM DROM1 AT AIS)                                       71608410
                                885  * AH AND AHI (ENTER FROM DROM2 AT AH)                                   71608420
                                886  * AHR  (ENTER FROM DROM1 AT AH)                                         71608430
                                887  *                                                                          71608440
28A     05 6009                 888  AIS       L     ARL,YSI,IRJH        (ARL)=IMM.OPERAND;START INST.FETCH     71608450
28B     40 CA79                 889  AH        A     YDL,YDL,ARL,CO+F    (R1)=(R1)+(ARL);MODIFY C,V,G&L         71608460
                                890  *                                                                          71608470
                                891  * ACH (ENTER FROM D2)                                                    71608480
                                892  * ACHR (ENTER FROM D1)                                                   71608490
                                893  *                                                                          71608500
28C     05 380F                 894  ACH       L     FLR,PSWL,IRJH       CARRY=PSWL<12>;START INST. READ       71608510
28D     40 CA79                 895            A     YDL,YDL,ARL,CI+CO+F (R1)=(R1)+(ARL)+CARRY;MOD. C,V,G&L    71608520
                                896  *                                                                          71608530
                                897  *                                                                          71608540
                                898  * SH AND SHI (ENTER FROM DROM2 AT SH)                                    71608550
                                899  * SHR (ENTER FROM DROM1 AT SH)                                           71608560
                                900  * SIS (ENTER FROM DROM1 AT SIS)                                          71608570
                                901  *                                                                          71608580
28E     05 6009                 902  SIS       L     ARL,YSI,IRJH        (ARL)=IMM.OPERAND;STTART INST. FETCH   71608590
28F     50 CA79                 903  SH        S     YDL,YDL,ARL,CO+F    (R1)=(R1)-(ARL);MODIFY C,V,G & L       71608600
```

```
                                  904  *                                                                    71608610
                                  905  * SCH (ENTER FROM DROM2)                                             71608620
                                  906  * SCHR(ENTER FROM DROM1)                                            71608630
290       05 380F                 907  SCH       L     FLR,PSWL,IRJH       CARRY=PSWL<12>;START INST. READ  71608640
291       50 CAF9                 908            S     YDL,YDL,ARL,CI+CO+F (R1)=(R1)-(ARL)-CARRY;MODIFY C,V,G&L 71608650
                                  909  *                                                                    71608660
                                  910  *                                                                    71608670
                                  911  * AHM (ENTER FROM D2)                                               71608680
                                  912  *                                                                    71608690
292       08 8010                 913  AHM       L     NULL,NULL,MR        START MEMORY READ                71608700
293       4E CC76                 914            A     MDR,YDL,MDR,CO+F+MW (MDR)=(R1)+2ND OPERAND;START MW  71608710
                                  915  *  THIS ROUTINE COPIES (FLR) TO COND.CODE,AND FETCHES INST. AT (LOC) 71608720
294       00 A017                 916  FETCHJ    L     MAR,LOC             (MAR)=(LOC)                      71608730
295       05 8010                 917  INSTRJ    L     NULL,NULL,IRJH      COPY FLR TO COND CODE & FETCH    71608740
296       00 8010                 918            L     NULL,NULL           AT (LOC)                         71608750
                                  919  * LOAD BYTE (LB)                                                     71608760
                                  920  * ENTER FROM D2                                                     71608770
297       08 8010                 921  LB        L     NULL,NULL,MR        START MEM. READ                  71608780
298       00 B609                 922            L     ARL,MDR,CS          (ARL) = 2ND OPERAND              71608790
299       00 A017                 923            L     MAR,LOC             (MAR)=(LOC)                      71608800
                                  924  * ENTRY POINT FOR LBR (FROM D1)                                      71608810
29A       9F F219                 925  LBR       NI    YDL,'FF',ARL        R1.0:7>=0 ; R1<8:15>= 2ND OPERAND 71608820
29B       04 8010                 926            L     NULL,NULL,IR        FETCH NEXT INSTR.                71608830
                                  927  *                                                                    71608840
                                  928  * STBR  (ENTER FROM D1)                                             71608850
                                  929  *                                                                    71608860
29C       8F F609                 930  STBR      LI    ARL,'FF',CS         (ARL)='FF00'                     71608870
29D       10 EA01                 931            N     MR1,YSL,ARL         CLEAR LOW 8 BITS OF 2ND OPERAND  71608880
29E       8F F009                 932            LI    ARL,'FF'            (ARL)='00FF'                     71608890
29F       10 CA09                 933            N     ARL,YDL,ARL         ARL<0:7>=0; ARL<8:15>=R1<8:15>   71608900
2A0       24 0A1D                 934            O     YSL,MR1,ARL,IR      R2<8:15>=R1<8:15>; FETCH NEXT INST. 71608910
2A1       00 8010                 935            L     NULL,NULL           NOP                              71608920
                                  936  * STB (ENTER FROM D2)                                               71608930
2A2       08 8010                 937  STB       L     NULL,NULL,MR        START MEM. WRITE                 71608940
2A3       0E CE16                 938            L     MDR,YDL,CS+MW       INSERT BYTE & STORE              71608950
2A4       00 A017                 939            L     MAR,LOC             (MAR)=(LOC)                      71608960
2A5       04 8010                 940            L     NULL,NULL,IR        FETCH INSTR.                     71608970
                                  941  *          TEST HALFWORD IMMEDIATE (ENTER FROM DROM2)                71608980
2A6       10 CA30                 942  THI       N     NULL,YDL,ARL,F      MODIFY CONDITION CODE            71608990
                                  943  * CLB (ENTER FROM D2)                                               71609000
2A7       8F F009                 944  CLB       LI    ARL,'FF'            (ARL)='00FF'                     71609010
2A8       18 CA00                 945            N     MR0,YDL,ARL,MR      (MR0) = 1ST OPERAND              71609020
2A9       00 B601                 946            L     MR1,MDR,CS          MR1<8:15>=2ND OPERAND            71609030
2AA       00 A017                 947            L     MAR,LOC             (MAR)=(LOC)                      71609040
2AB       15 0A09                 948            N     ARL,MR1,ARL,IRJH    (ARL)=2ND OPERAND;START I INST.FETCH 71609050
2AC       50 0270                 949            S     NULL,MR0,ARL,CO+F   SET COND. CODE TO INDICATE COMPARE 71609060
                                  950  * EXBR (ENTER FROM D1)                                              71609070
2AD       04 EE19                 951  EXBR      L     YDL,YSL,CS+IR       R1<0:7>=R2<8:15> ; R1<8:15>=R2<0:7>; 71609080
2AE       00 8010                 952            L     NULL,NULL           FILLER(NOT USED).                71609090
                                  953  * ENTER HERE FOR LM AND STM  (FROM DROM1)                           71609100
                                  954  *                                                                    71609110
2AF       00 6809                 955  LSTM      L     ARL,YDI             (ARL)=R1 FIELD VALUE             71609120
2B0       43 F417                 956            A     MAR,YSLX,MDR,D2     (MAR)=2ND OPERAND ADDR.;VECTOR THRU 71609130
2B1       D0 F208                 957            SI    CTR,'F',ARL         DROM2 ;(COUNTER)=NO.OF REG. TO BE 71609140
```

```
                                    958  *                                                                                    71609150
                                    959  *                                                                                    71609160
                                    960  * LM (ENTER FROM D2)                                                               71609170
2B2     09 8010                     961  LM      L     NULL,NULL,MR2      START MEM. READ AND INCR. MAR BY 2                71609180
2B3     09 B01A                     962  LM1     L     YDLP1,MDR,MR2      LOAD REGISTER ; INCR. MAR BY 2                    71609190
2B4     FC 82B3                     963          BF    CNTR,LM1           REPEAT UNTIL ALL REGS.ARE LOADED                  71609200
2B5     F0 0279                     964          B     NBR                FETCH NEXT INSTRUCTION                            71609210
                                    965  *                                                                                    71609220
                                    966  *                                                                                    71609230
                                    967  * STM (ENTER FROM D2)                                                              71609240
                                    968  *                                                                                    71609250
2B6     0F D016                     969  STM     L     MDR,YDLP1,MW2      STORE REG.;INCR MAR BY 2                          71609260
2B7     FC 82B6                     970          BF    CNTR,STM           REPEAT UNTIL ALL REGS. ARE STORED                 71609270
2B8     F0 0279                     971          B     NBR                FETCH NEXT INSTRUCTION                            71609280
                                    972  *       COMMON FOR ACH,SCH,MH,DH,CH,MHU  (ENTER FROM DROM2)                       71609290
2B9     40 F417                     973  RXNOIR  A     MAR,YSLX,MDR       (MAR)=SECOND OPERAND ADDRESS                      71609300
2BA     08 A017                     974          L     MAR,LOC,MR         (MAR)=(LOC);FETCH OPERAND                         71609310
2BB     03 8010                     975          L     NULL,NULL,D2                                                         71609320
2BC     00 B009                     976          L     ARL,MDR            (ARL)=SECOND OPERAND                              71609330
                                    977  * COMMON FOR THI,CHI  (ENTER FROM DROM1)                                          71609340
2BD     03 8010                     978  RINOIR  L     NULL,NULL,D2                                                         71609350
2BE     40 F409                     979          A     ARL,YSLX,MDR       (ARL)=SECOND OPERAND                              71609360
                                    980  *                                                                                    71609370
                                    981  * LOAD IMMEDIATE SHORT (ENTER FROM DROM1)                                          71609380
2BF     05 6039                     982  LIS     L     YDL,YSI,IRJH+F     (R1)=SHORT IMM. 2ND OPERAND                       71609390
2C0     00 8010                     983          L     NULL,NULL                                                            71609400
                                    984  * WRITE FUNCTION (PART OF CHANNEL I/O)                                            71609410
2C1     78 0000                     985  WRITE   C     MR                 READ FROM MEMORY                                  71609420
2C2     00 B655                     986          L     IO,MDR,CS+DA       OUTPUT DATA BYTE                                  71609430
2C3     40 B917                     987          A     MAR,MAR,ONE        INCR. MAR BY ONE                                  71609440
2C4     FC 82C1                     988          BF    CNTR,WRITE         REPEAT UNTIL SPECIFIED NUMBER OF                  71609450
                                    989  *                                BYTES HAVE BEEN OUTPUT TO THE DEV.                71609460
                                    990  * COME HERE AFTER TRANSFERRING SPECIFIED NUMBER OF BYTES                           71609470
2C5     00 B809                     991  END1    L     ARL,MAR            (ARL)=INCREMENTED START ADDRESS                   71609480
2C6     50 2250                     992          S     NULL,MR4,ARL,CO    FINAL ADDRESS - START ADDRESS                     71609490
2C7     E2 02D2                     993          BT    C,TERM1            DONE IF INCR. START ADDR.>FINAL ADDR              71609500
                                    994  * COME HERE AFTER COMPLETION OF DATA TRANSFER                                     71609510
                                    995  * INCREMENTED START ADDRESS IS NOT GREATER THAN FINAL ADDRESS                     71609520
2C8     00 160F                     996          L     FLR,MR2,CS         (FLR) = BITS 4:7 OF CCW                           71609530
2C9     F1 04E6                     997          BF    V,END3             TO END3 IF BIT 5 OF CCW IS RESET                  71609540
                                    998  * COMPARE LAST BYTE TRANSFERRED WITH TERMINAL CHAR.                               71609550
2CA     50 B917                     999          S     MAR,MAR,ONE        (MAR)= ADDRESS OF LAST BYTE                       71609560
2CB     00 B609                    1000          L     ARL,MDR,CS         ARL<8:15>= LAST BYTE                              71609570
2CC     C0 6717                    1001          AI    MAR,'6',ARH        (MAR)=ADDR. OF COMMAND BYTE/TERM CH.              71609580
2CD     08 800F                    1002          L     FLR,NULL,MR        CLEAR FLAG REGISTER ;FETCH TERM.CHAR              71609590
2CE     9F F209                    1003          NI    ARL,'FF',ARL       ARL<0:7>=0; ARL<8:15>=LAST BYTE                   71609600
2CF     9F F405                    1004          NI    MR5,'FF',MDR       MR5<0:7>=0; MR5<8:15>=TERM. CHAR.                 71609610
2D0     30 2A30                    1005          X     NULL,MR5,ARL,F     COMPARE LAST BYTE AND TERM. CHAR.                 71609620
2D1     E0 84E6                    1006          BT    G,END3             BRANCH TO END3 IF DIFFERENT                       71609630
                                   1007  * TERMINATION PHASE                                                               71609640
                                   1008  *  DATA TRANSFER IS COMPLETE                                                      71609650
2D2     84 0609                    1009  TERM1   LI    ARL,'40',CS        (ARL)='4000'                                      71609660
2D3     20 1216                    1010          O     MDR,MR2,ARL        SET NOP ;(MDR)=MODIFIED CCW                       71609670
2D4     40 8717                    1011  TERM4   A     MAR,NULL,ARH       (MAR)=CCW ADDRESS                                 71609680
```

```
2D5   7E 0090   1012          C     MW+PRIV          STORE UPDATED CCW                    71609690
2D6   40 8717   1013  TERM2   A     MAR,NULL,ARH     (MAR)=CCW ADDRESS                    71609700
2D7   50 BB17   1014          S     MAR,MAR,TWO                                           71609710
2D8   00 0816   1015          L     MDR,MR1          (MR1)=DEV NO./STATUS                 71609720
2D9   7E 0090   1016          C     MW+PRIV          STORE DEV.NO. /DEV.STATUS            71609730
2DA   00 160F   1017          L     FLR,MR2,CS       (FLR) = CCW BITS 4:7                 71609740
2DB   E0 84D0   1018          BT    G,QUEUE          BRANCH TO QUEUE IF QUEUE BIT IS 1    71609750
                1019  *       COME HERE AFTER ENTERING CCW ADDR. IN TERM QUEUE            71609760
2DC   00 960F   1020  TERM3   L     FLR,SRL,CS       (FLR)=CCW BITS 8:11                  71609770
2DD   E0 83EB   1021          BT    G,CHAIN          BRANCH TO CHAIN IF CCW 10 IS SET     71609780
2DE   F0 04E6   1022          B     END3             EXIT FROM CHANNEL I/O                71609790
                1023  * FUNCTION  IS  NOT  READ  OR  WRITE                               71609800
                1024  *                                                                   71609810
2DF   E1 04E6   1025  NI01    BT    V,END3           NOP  IF CCW<1> IS SET                71609820
2E0   E2 02E7   1026          BT    C,NI02           INIT FUNCT. IF CCW<0> IS SET         71609830
                1027  * NULL OR DMT FUNCTION                                             71609840
                1028  *                                                                   71609850
2E1   E0 42D6   1029  NI03    BT    L,TERM2          NULL OPERATION  IF BIT 3 OF CCW=1    71609860
                1030  * DECREMENT   MEMORY AND TEST  OPERATION                           71609870
2E2   00 800F   1031          L     FLR,NULL         CLEAR FLAG REGISTER                  71609880
2E3   50 B136   1032          S     MDR,MDR,ONE,F    DECREMENT  COUNT BY 1               71609890
2E4   7E 0090   1033          C     MW+PRIV          STORE UPDATED COUNT                  71609900
2E5   E0 C4E6   1034          BT    G+L,END3         BRANCH  TO END3  IF NOT ZERO         71609910
2E6   F0 02D6   1035          B     TERM2                                                 71609920
                1036  * INITIALIZATION  PHASE                                           71609930
                1037  *                                                                   71609940
2E7   00 12D6   1038  NI02    L     MR6,MR2,SL                                            71609950
2E8   00 3416   1039          L     MDR,MR6,SR       (MDR)=CCW  WITH BIT0(INIT) RESET     71609960
2E9   40 8717   1040          A     MAR,NULL,ARH     (MAR)=CCW  ADDRESS                   71609970
2EA   7E 0090   1041          C     MW+PRIV          STORE UPDATED  CCW                   71609980
2EB   00 960F   1042          L     FLR,SRL,CS       (FLR)= CCW BITS 8:11                 71609990
2EC   E2 03E7   1043          BT    C,CC2            OUTPUT COMMAND IF BIT 8 =1           71610000
2ED   C0 2717   1044          AI    MAR,'2',ARH      (MAR)= ADDR. OF START ADR  OR COUNT  71610010
2EE   08 900F   1045          L     FLR,SRL,MR       (FLR)=CCW BITS 0:3                   71610020
                1046  *                              FETCH START ADDRESS OR COUNT         71610030
2EF   E0 82E1   1047          BT    G,NI03           TO NI03  IF DMT OR NULL              71610040
2F0   F0 01D2   1048          B     RDWRT            BRANCH TO RDWRT                      71610050
2F1             1049          ORG   '2FF'                                                71610052
                1050  * A PRIV. INST. HAS BEEN ENCOUNTERED IN PROTECT MODE               71610054
2FF   F0 0003   1051          B     PRIV             BRANCH TO PRIV ROUTINE               71610056
```

```
                                1053            ORG   '300'                                                    71610060
                                1054  * FLOATING POINT INSTRUCTIONS                                            71610070
                                1055  *  FLOAT. POINT STORE    (STE)                                          71610080
                                1056  *  COME HERE AFTER VECTORING THROUGH DROM1                             71610090
300             40 F417         1057  STE       A     MAR,YSLX,MDR        (MAR)= 2ND OPERAND ADDRESS         71610100
301             00 B809         1058            L     ARL,MAR             (ARL)= 2ND OPERAND ADDR.           71610110
302             00 6A17         1059            L     MAR,YDI,SL          (MAR)=1ST OPERAND ADDRESS          71610120
303             08 6A00         1060            L     MR0,YDI,SL+MR       START MEM READ;(MR0)=1ST OP. ADDR. 71610130
304             40 8217         1061            A     MAR,NULL,ARL        (MAR)= 2ND OPERAND ADDR.           71610140
305             4E 0317         1062            A     MAR,MR0,TWO,MW      START MEM.WRITE;(MAR)=1ST OP.ADR.+2 71610150
306             C0 2209         1063            AI    ARL,'2',ARL                                            71610160
307             48 8217         1064            A     MAR,NULL,ARL,MR     START MEM READ;(MAR)=1ST OP.ADR.+2 71610170
308             0E A017         1065            L     MAR,LOC,MW          START MEM. WRITE;(MAR)=(LOC)       71610180
309             04 8010         1066            L     NULL,NULL,IR        FETCH NEXT INSTRUCTION             71610190
30A             00 8010         1067            L     NULL,NULL           NOP                                71610200
                                1068  *  FLOAT . POINT LOAD  (LE)                                            71610210
30B             40 F417         1069  LE        A     MAR,YSLX,MDR        (MAR)=EFFECTIVE 2ND OPERAND ADDR.  71610220
30C             F0 030E         1070            B     LE1                 BRANCH TO LE1                      71610230
                                1071  * FLOAT. POINT LOAD REGISTER LER                                        71610240
30D             00 6217         1072  LER       L     MAR,YSI,SL          (MAR)=SECOND OPERAND ADDRESS       71610250
                                1073  * COMMON FOR LE AND LER.  MAR CONTAINS SECOND OPERAND ADDRESS          71610260
30E             79 00A0         1074  LE1       C     TABT+MR2            ENABLE ABORT;START MR & INCR. MAR  71610270
30F             00 6A00         1075            L     MR0,YDI,SL          (MR0) = 1ST OPERAND ADDRESS        71610280
310             9F F433         1076            NI    SRH,'FF',MDR,F      (SRH) = HIGH FRACTION; MODIFY G&L   71610290
311             30 9C04         1077            X     MR4,SRH,MDR         MR4<0:7>=SIGN & EXPONENT           71610300
312             48 0317         1078            A     MAR,MR0,TWO,MR      (MAR)=1ST OPERAND ADDR.+2           71610310
313             03 B032         1079            L     SRL,MDR,D2+F        (SRH,SRL)= FRACTION; VECTOR THROUGH 71610320
                                1080  *                                   DROM2 TO LE2; MODIFY G&L FLAGS     71610330
314             00 2206         1081            L     MR6,MR4,SL          MR6<0:6>=EXPONENT                  71610340
                                1082  * COME HERE (AFTER VECTORING THROUGH DROM1) FOR                         71610350
                                1083  * AE,SE,CE,ME, AND DE INSTRUCTIONS                                     71610360
315             40 F417         1084  FPRX      A     MAR,YSLX,MDR        (MAR)=SECOND OPERAND ADDRESS       71610370
316             F0 0318         1085            B     COMRRX              BRACH TO COMMON ROUTINE            71610380
                                1086  * ENTER HERE(AFTER D1) FOR AER, SER,MER,DER & CER INSTRUCTIONS         71610390
317             00 6217         1087  FPRR      L     MAR,YSI,SL          (MAR)=SECOND OPERAND ADDRESS       71610400
                                1088  * COMMON FOR AER,AE,SER,SE,CER,CE,MER,ME,DER & DE INSTRUCTIONS         71610410
                                1089  * (MAR) = SECOND OPERAND ADDRESS                                       71610420
318             79 00A0         1090  COMRRX    C     TABT+MR2            ENABLE ABORT;START MR& INCR. MAR   71610430
319             8F F609         1091            LI    ARL,X'FF',CS        (ARL)=X'FF00'                      71610440
31A             9F F400         1092            NI    MR0,'FF',MDR        (MR0)=HIGH SECOND OPERAND FRACTION  71610450
31B             18 B203         1093            N     MR3,MDR,ARL,MR      MR3<0:7>=SIGN & EXP. OF 2ND OPERAND 71610460
31C             00 1A04         1094            L     MR4,MR3,SL          MR4<0:6>=EXP. OF 2ND OPERAND       71610470
31D             00 6A17         1095            L     MAR,YDI,SL          (MAR)=1 ST OPERAND ADDRESS         71610480
31E             09 B001         1096            L     MR1,MDR,MR2         (MR1) = LOW SECOND OPERAND FRACT.; 71610490
                                1097  *                                   START MEM. READ & INCR. MAR BY 2   71610500
31F             9F F402         1098            NI    MR2,'FF',MDR        (MR2)=HIGH 1ST OPERAND FRACTION    71610510
                                1099  *                                                                      71610520
320             13 B205         1100            N     MR5,MDR,ARL,D2      MR5<0:7>=SIGN & EXP. OF 1ST OPER.; 71610530
                                1101  *                                   VECTOR THROUGH DROM2               71610540
321             08 2A06         1102            L     MR6,MR5,SL+MR       MR6<0:6>=EXP. OF 1ST OPERAND;MR    71610550
                                1103  *  FLOAT. POINT SUBTRACT                                               71610560
                                1104  *  ENTER  AFTER D2                                                     71610570
322             88 060A         1105  FSUB      LI    ARH,'80',CS         (ARH) = X'8000'                    71610580
323             30 1F03         1106            X     MR3,MR3,ARH         COMPLEMENT THE SIGN BIT OF THE     71610590
```

```
                                1107  *                                                SECOND OPERAND                                  71610600
                                1108  *  ENTRY  FOR  FLOAT. ADR AFTER  D2                                                      71610610
                                1109  *  ENTRY  FOR  FLOAT. SUBTRACT AFTER COMPLEMENTING THE SIGN BIT                          71610620
                                1110  *  OF  THE  SECOND  OPERAND                                                              71610630
                                1111  *  (MR0,MR1) = SECOND OPERAND FRACTIONS                                                  71610640
                                1112  *  (MR2,MDR) = FIRST OPERAND FRACTION                                                    71610650
                                1113  *  MR3<0:7> = SIGN & EXP. OF THE SECOND OPERAND;MR4<0:6>=EXP.2NDOP                       71610660
                                1114  *  MR5<0:7> = SIGN & EXP. OF THE FIRST OPERAND                                           71610670
                                1115  *  MR6<0:6> = EXPONENT OF FIRST OPERAND                                                  71610680
                                1116  *  FIRST OPERAND IS CALLED A & SECOND OPERAND IS CALLED B                                 71610690
324     00 1009                 1117  FADSUB   L     ARL,MR2              (ARL)=HIGH A.FRACTION                                    71610700
325     50 0C50                 1118           S     NULL,MR1,MDR,C0      SUBTRACT  A.FRACTION FROM B.FRACT.                       71610710
326     50 02D0                 1119           S     NULL,MR0,ARL,CI+C0   SET CARRY IF A.FRACT.> B.FRACT.                          71610720
327     00 2613                 1120           L     SRH,MR4,CS           (SRH)=2*(B.EXP.)                                         71610730
328     00 3609                 1121           L     ARL,MR6,CS           (ARL)=2*(A.EXP)                                          71610740
329     50 9AD0                 1122           S     NULL,SRH,ARL,CI+C0   SET C IF A.MAGNITUDE > B.MAGNITUDE                       71610750
32A     50 9A09                 1123           S     ARL,SRH,ARL          (ARL) = 2*(B.EXP-A.EXP)                                  71610760
32B     E2 03C7                 1124           BT    C,AGB                BRANCH TO AGB IF A.MAGN.> B.MAGN.                        71610770
                                1125  * COME  HERE IF A. MAGNITUDE IS LESS THAN OR EQUAL TO B.MAGNITUDE                          71610780
                                1126  * (ARL) = B.EXP-A.EXP                                                                      71610790
32C     D0 A250                 1127           SI    NULL,'A',ARL,C0      BRANCH TO BRESLT IF EXPONENT                             71610800
32D     E2 03C2                 1128           BT    C,BRESLT             DIFFERENCE TIMES 2 IS >10                                71610810
32E     C0 0208                 1129           AI    CTR,'0',ARL          (COUNTER)=2* EXP. DIFFERENCE                             71610820
32F     00 B012                 1130           L     SRL,MDR              (SRH,SRL)= FRACTION OF THE                               71610830
330     00 1013                 1131           L     SRH,MR2              SMALLER OPERAND                                          71610840
331     00 0816                 1132           L     MDR,MR1              (MR2,MDR) = FRACTION OF THE LARGER                       71610850
332     00 0002                 1133           L     MR2,MR0              OPERAND                                                  71610860
333     00 2006                 1134           L     MR6,MR4              MR6 <0:6> = EXP. OF LARGER OPERAND                       71610870
334     00 1804                 1135           L     MR4,MR3              MR4<0:7>=SIGN & EXP OF LARGER OPER.                      71610880
                                1136  * (COUNTR) = 2*EXPONENT DIFFERENCE                                                         71610890
                                1137  * (SRH,SRL)=FRACTION OF THE SMALLER OPERAND                                               71610900
                                1138  * (MR2,MDR)=FRACTION OF THE LARGER OPERAND                                                71610910
                                1139  * MR4<0:7>=SIGN AND EXPONENT OF THE LARGER OPERAND                                        71610920
                                1140  * MR6 <0:6>=EXPONENT OF THE LARGER OPERAND                                                71610930
                                1141  * MR3 <0:7>=SIGN AND EXPONENT OF THE SECOND OPERAND                                       71610940
                                1142  * MR5 <0:7>=SIGN AND EXPONENT OF THE FIRST OPERAND                                        71610950
335     70 0008                 1143  CONT1    C     RPT                  EQUALIZE THE                                             71610960
336     70 0C00                 1144           C     SR2                  SMALLER OPERAND                                          71610970
337     00 2809                 1145           L     ARL,MR5              ARL<0:7>=SIGN & EXP. OF 1ST OPERAND                      71610980
338     30 1A30                 1146           X     NULL,MR3,ARL,F       SET L FLAG IF SIGNS ARE DIFFERENT                        71610990
339     F0 4352                 1147           BF    L,SUM                BRANCH TO SUM IF L FLAG IS RESET                         71611000
                                1148  *  COME HERE IF DIFFERENCE IS REQUIRED (SIGNS ARE NOT SAME)                                71611010
33A     80 000F                 1149           LI    FLR,'0'              CLEAR FLAG REGISTER                                      71611020
33B     00 9009                 1150           L     ARL,SRL                                                                       71611030
33C     50 B272                 1151           S     SRL,MDR,ARL,F+C0                                                              71611040
33D     00 9809                 1152           L     ARL,SRH                                                                       71611050
33E     50 12F3                 1153           S     SRH,MR2,ARL,CI+C0+F  (SRH,SRL)=DIRRERENCE FRACTION                          71611060
                                1154  * ENTER HERE FOR LE AND LER AFTER D2                                                       71611070
33F     F0 C389                 1155  LE2      BF    G+L,ZRESLT           RESULT IS ZERO IF G&L ARE RESET                          71611080
                                1156  * COME HERE  FOR  NORMALIZATION                                                            71611090
                                1157  * (SRH,SRL)=FRACTION ; MR4<0:7>=SIGN & EXP.; MR6<0:6> = EXPONENT                           71611100
340     FD 0349                 1158  NRMAL    BF    NNORM,FEND           TAKE COMMON EXIT IF NORM,NOT REQD.                       71611110
341     80 0000                 1159           LI    MR0,'0'              (MR0)=0                                                  71611120
                                1160  *                                                                                          71611130
```

```
                                1161  NRLOOP   C     SL2                 SHIFT LEFT                                71611140
                                1162           C     SL2                 4 PLACES                                  71611150
342      70 1400                1163           A     MR0,MR0,TWO         (MR0)=(MR0)+2                             71611160
343      70 1400                1164           BT    NNORM,NRLOOP        REPEAT IF NOT NORMALIZED                  71611170
344      40 0300                1165           L     ARL,MR0,CS          ARL<0:6>=NORMALIZATION COUNT              71611180
345      ED 0342                1166           S     MR6,MR6,ARL,C0      MR6<0:6>=RESULT EXPONENT                  71611190
346      00 0609                1167           BT    C,EXPUF             EXPONENT UNDERFLOW IF C IS SET            71611200
347      50 3246                1168  *  COMMON EXIT FOR FLOAT.POINT ADD,SUBTRACT,MULTIPLY,DIVIDE & LOAD        71611210
348      E2 03B5                1169  * (SRH,SRL)=NORM. FRACT.;(MAR)=1ST OPER. ADDR. +2                         71611220
                                1170  *  MR4<0>= SIGN OF  THE RESULT; MR6<0:6>= EXP. OF THE RESULT              71611230
349      80 000F                1171  FEND     LI    FLR,'0'             CLEAR FLAG REGISTER                       71611240
34A      00 9036                1172           L     MDR,SRL,F           (MDR)= LOW RESULT ;MODIFY FLAGS           71611250
34B      7E 00B0                1173           C     MW+PRIV+TABT        START PRIVLEGED WRITE;DISABLE ABORT       71611260
34C      50 BB17                1174           S     MAR,MAR,TWO         DECREMENT MAR BY 2                        71611270
                                1175  *                                                                           71611280
34D      00 2250                1176           L     NULL,MR4,SL+C0      (CARRY)=SIGN OF THE RESULT                71611290
34E      00 3489                1177           L     ARL,MR6,SR+CI       ARL<0:7>=SIGN & EXP. OF RESULT            71611300
34F      20 9A76                1178           O     MDR,SRH,ARL,F+CO    (MDR)=HIGH RESULT;MODIFY G&L               71611310
                                1179  *                                 AND RESET CARRY                           71611320
350      7E 0090                1180           C     MW+PRIV             PRIVILEGED WRITE                          71611330
351      F0 0294                1181           B     FETCHJ              FETCH NEXT INSTRUCTION                    71611340
                                1182  *                                 COPY FLR TO CONDITION CODE                71611350
                                1183  * COME HERE IF SUM IS REQUIRED (SIGNS ARE SAME)                           71611360
352      00 1009                1184  SUM      L     ARL,MR2             (ARL) = HIGH LARGER OPERAND               71611370
353      40 9472                1185           A     SRL,SRL,MDR,F+C0    (SRH,SRL)=SUM OF TWO FRACTIONS ;          71611380
354      40 9AB3                1186           A     SRH,SRH,ARL,CI+F    MODIFY FLAGS                              71611390
355      F0 83B9                1187           BF    G,ZRESLT            RESULT IS ZERO G FLAG IS RESET            71611400
                                1188  * COME HERE FOR FLOAT. MULT.&DIV.(AFTER ROUNDING IF RESILT IS NOT 0)      71611405
356      00 9E0F                1189  ASMDE    L     FLR,SRH,CS          L FLAG = SRH<7>                           71611410
357      F0 4349                1190           BF    L,FEND              EXIT TO COMMON ROUTINE IF L IS            71611420
                                1191  *                                 RESET                                     71611430
358      70 0C00                1192           C     SR2                 SHIFT RIGHT                               71611440
359      70 0C00                1193           C     SR2                 4 PLACES                                  71611450
35A      80 2609                1194           LI    ARL,'2',CS          (ARL)=X'0200'                             71611460
35B      40 3246                1195           A     MR6,MR6,ARL,C0      MR6<0:6> = RESULT EXPONENT                71611470
35C      E2 03CF                1196           BT    C,EXPOF             EXP. OVERFLOW IF CARRY IS SET             71611480
35D      F0 0349                1197           B     FEND                TAKE COMMON EXIT                          71611490
                                1198  * FLOAT  POINT  COMPARE CE & CER (COMMON FOR HW & EXT. MODES)            71611500
                                1199  * ENTER AFTER D2 ; (MR0,MR1)=2ND OPERAND FRACT.; (MR2,MDR)= 1ST           71611510
                                1200  * OPERAND FRACT.; MR3<0:7>=SIGN & EXP. OF 2ND OPER.;MR4<0:6>=EXP OF 2ND   71611520
                                1201  * OPERAND;MR5<0:7>=SIGN & EXP. OF 1ST OPER.;MR6<0:6>=EXP.OF 1ST OPER.     71611530
35E      00 1809                1202  CE1      L     ARL,MR3             ARL<0:7> = SIGN & EXP. OF 2ND OPER.       71611540
35F      00 A017                1203           L     MAR,LOC             (MAR)=(LOC)                               71611550
360      30 2A30                1204           X     NULL,MR5,ARL,F                                                71611560
361      E0 4370                1205           BT    L,CE2               SIGNS ARE DIFFERENT IF L IS SET           71611570
                                1206  * OPERANDS HAVE LIKE SIGNS                                                  71611580
362      80 000F                1207           LI    FLR,'0'             CLEAR FLAG REGISTER                       71611590
363      00 0809                1208           L     ARL,MR1             (ARL) = LOW 2ND OP. FRACTION              71611600
364      00 000A                1209           L     ARH,MR0             (ARH)=HIGH 2ND OP. FRACTION               71611610
365      50 B270                1210           S     NULL,MDR,ARL,CO+F   SET FLAG. REGISTER TO INDICATE           71611620
366      50 17F0                1211           S     NULL,MR2,ARH,CO+CI+F      SIGN & MAGNITUDE OF FRACT.DIFF.   71611630
367      00 200A                1212           L     ARH,MR4             ARH<0:7>=EXPONENT OF 2ND OPER.            71611640
368      50 37F0                1213           S     NULL,MR6,ARH,CI+CO+F      SET FLR TO INDICATE MAGN.COMPAR  71611650
                                1214  * FLAG REGISTER INDICATES SIGN & MAGNITUDE OF(1ST OP.MAG.-2ND OP.MAG.)    71611660
```

```
369       F0 C373          1215            BF    G+L,CE4              BRANCH TO CE4 IF DIFFERENCE=0          71611670
36A       E2 036D          1216            BT    C,CE3                BRANCH TO CE3 IF MAGN. OF 1ST          71611680
                           1217  *                                        OPERAND IS < MAGN. OF 2ND OPER.    71611690
36B       00 2A50          1218            L     NULL,MR5,SL+CO       CARRY = SIGN OF 1ST OPERAND            71611700
36C       55 8080          1219            S     NULL,NULL,NULL,CI+F+IRJH START INST, FETCH&SET CC           71611710
                           1220  *  1ST OPERAND MANITUDE < SECOND OPERAND MAGNITUDE                          71611720
36D       88 0609          1221  CE3       LI    ARL,'80',CS          (ARL)=X'8000'                          71611730
36E       55 2A70          1222            S     NULL,MR5,ARL,CO+F+IRJH   START INST.FETCH & SET CC          71611740
36F       50 80F0          1223            S     NULL,NULL,NULL,CI+CO+F  RESET V FLAG WITHOUT CHANGING       71611750
                           1224  *                                        OTHER FLAGS                        71611760
                           1225  *  SINGS  ARE  DIFFERENT                                                   71611770
370       00 2A50          1226  CE2       L     NULL,MR5,SL+CO       CARRY=FIRST OPERAND SIGN               71611780
371       05 2830          1227            L     NULL,MR5,F+IRJH      MODIFY G&L;START INST.FETCH            71611790
372       00 8010          1228            L     NULL,NULL            NOP                                    71611800
                           1229  *   1ST  OPERAND = SECOND OPERAND                                           71611810
373       05 800F          1230  CE4       L     FLR,NULL,IRJH        START ISNT. FETCH & SET CC = 0         71611820
374       00 8010          1231            L     NULL,NULL            NOP                                    71611830
                           1232  *  FLOATING POINT MULTIPLY  ROUTINE                                        71611840
                           1233  *  ENTER  AFTER  VECTORING  THROUGH DROM2                                  71611850
                           1234  *  (MR0,MR1)=SECOND OPERAND FRACTION                                       71611860
                           1235  *  (MR2,MDR)= FIRST OPERAND FRACTION                                       71611870
                           1236  *  MR3<0:7> =SIGN & EXP. OF SECOND OPERAND                                 71611880
                           1237  *  MR4<0:6> = EXPONENT OF SECOND OPERAND                                   71611890
                           1238  *  MR5<0:7> = SIGN & EXP. OF FIRST OPERAND                                 71611900
                           1239  *  MR6<0:6> = EXPONENT OF FIRST OPERAND                                    71611910
                           1240  *  FIRST  OPERAND IS CALLED A & SECOND OPERAND IS CALLED B                 71611920
375       88 0609          1241  .FMPY     LI    ARL,X'80',CS         (ARL)=X'8000'                          71611930
376       30 220A          1242            X     ARH,MR4,ARL          ARH<0:6> = B.EXP. IN 2'S COMPLEMENT    71611940
377       30 3206          1243            X     MR6,MR6,ARL          MR6<0:6>=A.EXP. IN 2'S COMPLEMENT      71611950
378       40 3726          1244            A     MR6,MR6,ARH,F        MR6<0:6>=SUM OF EXP. IN 2'S COMPLE.;   71611960
                           1245  *                                        MODIFY G,L, & V FLAGS              71611970
                           1246  *                                                                           71611980
379       30 3206          1247            X     MR6,MR6,ARL          MR6<0:6>=EXP.SUM IN EXCESS 64          71611990
37A       00 2809          1248            L     ARL,MR5              ARL<0> = SIGN OF A                     71612000
37B       30 1A04          1249            X     MR4,MR3,ARL          MR4<0> = SIGN OF THE RESULT            71612010
37C       E1 0380          1250            BT    V,EXPOUF             EXPONENT  OVERFLOW OR UNDERFLOW        71612020
                           1251  *                                        IF V FLAG IS SET                   71612025
                           1252  * MULTIPLY  24 BIT A. FRACTION AND 24 BIT B. FRACTION                      71612030
37D       00 8005          1253            L     MR5,MDR              (MR5) = LOW  A.FRACTION                71612040
37E       00 0812          1254            L     SRL,MR1              (SRL)=LOW B. FRACTION                  71612050
37F       00 B249          1255            L     ARL,MDR,SL+CO        (ARL) = 2* LOW A.FRACTION              71612060
380       70 C400          1256            C     UMPY                 (SRH,SRL)= LOW A.FRACTION * LOW        71612070
                           1257  *                                        B. FRACTION                        71612080
                           1258  *  16 LEAST SIGNIFICANT BITS OF THE PRODCUT ARE IGNORED                     71612090
381       00 980A          1259            L     ARH,SRH              (ARH)=16 MSB OF (LOWA.FRACTION*LOW     71612100
                           1260  *                                        B.FRACTION)                        71612110
382       00 0012          1261            L     SRL,MR0              (SRL) = HIGH B.FRACTION                71612120
383       00 2816          1262            L     MDR,MR5              (MDR) = LOW A. FRACTION                71612130
384       00 B249          1263            L     ARL,MDR,SL+CO        (ARL) = 2*LOW A.FRACTION               71612140
385       70 C400          1264            C     UMPY                 (SRH,SRL)=HIGH B.FRACTION * LOW        71612150
                           1265  *                                        A. FRACTION                        71612160
386       40 9743          1266            A     MR3,SRL,ARH,CO       (ARH,MR3)=(LOW A * LOW B) +            71612170
                           1267  *                                        (LOW A * HIGH B)                   71612180
387       40 988A          1268            A     ARH,SRH,NULL,CI                                             71612190
```

```
388     00 1012            L     SRL,MR2              (SRL) = HIGH A.FRACTION                 71612200
389     00 0816            L     MDR,MR1              (MDR) = LOW B. FRACTION                 71612210
38A     00 8249            L     ARL,MDR,SL+CO        (MDR) = 2* LOW B FRACTION               71612220
38B     70 C400            C     UMPY                 (SRH,SRL)=HIGH A.FRACT * LOW B.FRAC     71612230
38C     00 1809            L     ARL,MR3                                                      71612240
38D     40 9243            A     MR3,SRL,ARL,CO       (ARH,MR3) = (LOW A * LOWB) +(LOW A *    71612250
                   *                                  HIGH B) + (HIGH A * LOW B)              71612260
38E     40 9F8A            A     ARH,SRH,ARH,CI                                               71612270
38F     00 0012            L     SRL,MR0              (SRL) = HIGH B.FRACTION                 71612280
390     00 1016            L     MDR,MR2              (MDR) = HIGH A.FRACTION                 71612290
391     00 B249            L     ARL,MDR,SL+CO        (ARL) = 2 * HIGH A.FRACTION             71612300
392     70 C400            C     UMPY                 (SRH,SRL)=HIGH B.FRACT*HIGH A FRACT     71612310
                   *  SRH MUST BE ZERO BECAUSE 8 MSB OF HIGH FRACTIONS ARE ZERO               71612320
393     40 9713            A     SRH,SRL,ARH          (SRH,SRL) = (LOW A*LOW B)+(LOW A *      71612330
                   *                                  HIGH B)+(HIGH A*LOWB)+(HI A * HI B)     71612340
394     00 1812            L     SRL,MR3                                                      71612350
                   *  (SRH,SRL) CONTAINS MOST SIGNIFICANT 32 BITS OF A.FRACT*B.FRACT          71612360
                   * IF THE MOST SIGNIFICANT HEX.DIG. OF THE PRODUCT IS NONZERO,EXP.          71612370
                   *  NEED NOT BE ADJUSTED &(SRH,SRL) SHOULD BE SHIFTED RIGHT 8 PLACES        71612380
                   * IF THE MOST SIGNIFICANT HEX. DIGIT OF 32 BIT PRODUCT IS ZERO,EXP.        71612390
                   * IS DECREMENTED BY ONE & (SRH,SRL) IS SHIFTED RIGHT 4 PLACES              71612400
395     80 4008            LI    CTR,'4'              (COUNTER) = 4                           71612410
396     80 000F            LI    FLR,'0'              CLEAR FLAG REGISTER                     71612420
397     8F 0609            LI    ARL,'F0',CS          (ARL)=X'F000'                           71612430
398     10 9A30            N     NULL,SRH,ARL,F       TEST MOST SIGNIFICANT HEX DIGIT         71612440
399     E0 C39E            BT    G+L,NOCOR            BRANCH TO 'NOCOR' IF NOT ZERO           71612450
39A     80 2008            LI    CTR,'2'              (COUNTER) = 2                           71612460
39B     80 2609            LI    ARL,'2',CS                                                   71612470
39C     50 3246            S     MR6,MR6,ARL,CO       DECR. EXP BY 1                          71612480
39D     E2 03B1            BT    C,EXPUFZ             UNDERFLOW IF C IS SET                   71612490
39E     80 000F    NOCOR   LI    FLR,'0'              CLEAR FLAG REGISTER                     71612500
                   *  COMMON FOR FLOAT. POINT MULTIPLY AND DIVIDE                             71612510
39F     70 0008    FMDIV   C     RPT                  SHIFT (SRH,SRL) RIGHT                   71612520
3A0     70 0C40            C     SR2+CO               2*(COUNTER ) PLACES                     71612530
3A1     40 90F2            A     SRL,SRL,NULL,CI+CO+F      PERFORM ROUNDING                   71612540
3A2     40 98F3            A     SRH,SRH,NULL,CI+CO+F                                         71612550
3A3     E0 C356            BT    G+L,ASMDE            GO TO COMMON ROUTINE IF (SRH,SRL)       71612560
                   *                                  IS NOT EQUAL TO ZERO                    71612570
3A4     80 0004            LI    MR4,'0'              SIGN BIT OF RESULT = 0                  71612580
3A5     80 0006            LI    MR6,'0'              EXPONENT OF RESULT = 0                  71612590
3A6     F0 0349            B     FEND                 TAKE COMMON EXIT                        71612600
                   * FLOATING POINT DIVISION BY ZERO                                          71612610
3A7     70 00A0    DIVZRO  C     TABT                 DISABLE ABORT                           71612620
3A8     80 C00F            LI    FLR,'C'              SET C & V FLAGS                         71612630
3A9     70 0088            C     JH                   COPY FLR TO COND. CODE                  71612640
                   * FLOATING POINT ARITHMETIC FAULT                                          71612650
3AA     00 800F    FFAULT  L     FLR,NULL             CLEAR FLAG REGISTER                     71612660
3AB     80 4609            LI    ARL,'4',CS                                                   71612670
3AC     10 3A30            N     NULL,PSWL,ARL,F      TEST PSW BIT 5                          71612680
3AD     F0 8279            BF    G,NBR                IGNORE FLAT. FAULT IF ZERO              71612690
3AE     82 8017            LI    MAR,'28'                                                     71612700
3AF     F0 0059            B     GENSWP                                                       71612710
                   * EXPONENT OVERFLOW OR UNDERFLOW OR ZERO RESULT(FOR MULT. OR DIVIDE)       71612720
3B0     E0 43CF    EXPOUF  BT    L,EXPOF              OVERFLOW IF L FLAG IS SET               71612730
```

```
                                1323  * ZERO RESULT OR UNDERFLOW(FOR FLOAT. MULT. OR DIVIDE)                    71612734
3B1     50 0150                 1324  EXPUFZ   S     NULL,MR0,ONE,CO     SET CARRY IF 2ND OPERAND IS ZERO     71612738
3B2     E2 03B9                 1325           BT    C,ZRESLT                                                 71612742
3B3     50 1150                 1326           S     NULL,MR2,ONE,CO     SET CARRY IF 1ST OPERAND IS ZERO     71612746
3B4     E2 03B9                 1327           BT    C,ZRESLT                                                 71612748
                                1328  * EXPONENT UNDERFLOW(FORCE THE RESULT TO ZERO)                            71612750
3B5     00 8013                 1329  EXPUF    L     SRH,NULL            (SRH,SRL)=0                          71612760
3B6     00 8012                 1330           L     SRL,NULL                                                 71612770
3B7     80 400F                 1331           LI    FLR,4               CLEAR G&L,AND SET V FLAG             71612780
3B8     F0 03D7                 1332           B     OUEXIT              BRANCH TO OUEXIT                     71612790
                                1333  *  PART OF FLOAT. POINT ADD SUBTRACT ROUTINE                            71612800
                                1334  *  COME HERE IF RESULT IS ZERO                                          71612810
3B9     80 0013                 1335  ZRESLT   LI    SRH,'0'             (SRH,SRL)=0                          71612815
3BA     80 0012                 1336           LI    SRL,'0'                                                  71612820
3BB     80 0006                 1337           LI    MR6,'0'             (MR6) = 0                            71612825
3BC     80 0004                 1338           LI    MR4,'0'             (MR4) =  0                           71612830
3BD     F0 0349                 1339           B     FEND                GO TO COMMON EXIT ROUTINE            71612840
                                1340  *  PART OF FLOAT. POINT ADD SUBTRACT ROUTINE                            71612850
                                1341  * COME HERE IF RESULT = FIRST OPERAND                                    71612860
3BE     00 1013                 1342  ARESLT   L     SRH,MR2             (SRH,SRL)=FIRST OPERAND              71612870
3BF     00 B012                 1343           L     SRL,MDR                                                  71612880
3C0     00 2804                 1344           L     MR4,MR5             MR<0:7>=SIGN & EXP. OF 2ND OPER.     71612890
3C1     F0 0349                 1345           B     FEND                GO TO COMMON EXIT ROUTINE            71612900
                                1346  *  PART OF FLOAT. POINT ADD SUBTRACT ROUTINE                            71612910
                                1347  * COME   HERE   IF RESULT = SECOND OPERAND                              71612920
3C2     00 0013                 1348  BRESLT   L     SRH,MR0             (SRH,SRL)=SECOND OPERAND             71612930
3C3     00 0812                 1349           L     SRL,MR1                                                  71612940
3C4     00 2006                 1350           L     MR6,MR4             MR6<0:6> = EXP. OF 2ND OPERAND       71612950
3C5     00 1804                 1351           L     MR4,MR3             MR4(0:7>=SIGN & EXP. OF 2ND OPER.    71612960
3C6     F0 0349                 1352           B     FEND                GO TO COMMON EXIT ROUTINE            71612970
                                1353  *  PART OF FLOAT. POINT ADD SUBTRACT ROUTINE                            71612980
                                1354  * COME HERE IF A.MAGNITUDE IS GREATER THAN B.MAGNITUDE                   71612990
3C7     D0 0209                 1355  AGB      SI    ARL,'0',ARL         (ARL)=EXP. DIFFERENCE                71613000
3C8     D0 A250                 1356           SI    NULL,'A',ARL,CO                                          71613010
3C9     E2 03BE                 1357           BT    C,ARESLT            BRANCH TO ARESLT IF DIFF. *2 IS >10  71613020
3CA     C0 0208                 1358           AI    CTR,'0',ARL         (COUNTER)=2*EXP. DIFFERENCE          71613030
3CB     00 0013                 1359           L     SRH,MR0             (SRH,SRL)= SMALLER OPERAND FRACT.    71613040
3CC     00 0812                 1360           L     SRL,MR1                                                  71613050
3CD     00 2804                 1361           L     MR4,MR5             MR4<0:7> = SIGN AND EXP. OF A        71613060
3CE     F0 03E5                 1362           B     CONT1                                                    71613070
                                1363  * EXPONENT OVERFLOW                                                      71613080
                                1364  * FORCE THE RESULT TO +'7FFFFFFF' OR 'FFFFFFFF'                          71613090
3CF     50 8113                 1365  EXPOF    S     SRH,NULL,ONE        (SRH,SRL)='FFFF FFFF'                71613100
3D0     00 9812                 1366           L     SRL,SRH                                                  71613110
3D1     80 500F                 1367           LI    FLR,'5'             SET V FLAG & L FLAG                  71613120
3D2     00 2250                 1368           L     NULL,MR4,SL+CO      TEST RESULT SIGN                     71613130
3D3     E2 03D6                 1369           BT    C,EXPOF1            SKIP NEXT TWO INST. IF MINUS         71613140
3D4     70 0800                 1370           C     SR1                 (SRH,SRL)='7FFF FFFF'                71613150
3D5     80 600F                 1371           LI    FLR,'6'             SET V & G FLAGS                      71613160
3D6     00 8050                 1372  EXPOF1   L     NULL,NULL,CO        RESET CARRY FLAG                     71613170
                                1373  *  EXIT FOR EXPONENT OVERFLOW OR UNDERFLOW                               71613180
                                1374  *  (SRH,SRL)=RESULT      (MAR)=DEST REG. ADDRESS+2                      71613190
3D7     00 9016                 1375  OUEXIT   L     MDR,SRL             (MDR)=LOW RESULT                     71613200
3D8     7E 00B0                 1376           C     MW+PRIV+TABT        START PRIV. WRITE; DISABLE ABORT     71613210
```

```
3D9     50 BB17   1377          S     MAR,MAR,TWO       DECREMENT MAR BY 2                    71613220
3DA     00 9816   1378          L     MDR,SRH           (MDR)=HIGH RESULT                     71613230
3DB     7E 009C   1379          DC    '7E009C'          =   C   JH+JF+MW+PRIV                71613240
3DC     F0 03AA   1380          B     FFAULT            B FLOAT POINT FAULT ROUTINE           71613250
                  1381  * DEVICE STATUS IS NOT O.K.                                           71613260
3DD     8C 2609   1382  BSTAT   LI    ARL,'C2',CS       (ARL)= 'C200'                         71613270
3DE     20 1202   1383          O     MR2,MR2,ARL       (MRL)= CCW WITH INIT, NOP AND         71613280
                  1384  *                               QUEUE BIT SET                         71613290
3DF     89 8209   1385          LI    ARL,'98',SL       (ARL)= '0130'                         71613300
3E0     20 1202   1386          O     MR2,MR2,ARL                                             71613310
3E1     30 1216   1387          X     MDR,MR2,ARL       (MDR)=(MR2)=CCW WITH HI/LO,CHAIN &    71613315
3E2     00 B002   1388          L     MR2,MDR           CONTINEUE BIT RESET                   71613320
3E3     00 1612   1389          L     SRL,MR2,CS                                              71613325
3E4     70 0C00   1390          C     SR2               SRL<12:15>=CCW<0:3>;                  71613330
3E5     70 0C00   1391          C     SR2               SRL<4:11>=CCW<8:15>                   71613335
3E6     F0 02D4   1392          B     TERM4                                                   71613340
                  1393  * OUTPUT COMMAND                                                      71613350
                  1394  *                                                                     71613360
3E7     C0 6717   1395  OC2     AI    MAR,'6',ARH                                           71613370
3E8     78 0000   1396          C     MR                FETCH OUTPUT COMMAND BYTE/TERM CHAR   71613380
3E9     00 B675   1397          L     IO,MDR,CS+OC      OUTPUT COMMAND                        71613390
3EA     F0 04E6   1398          B     END3                                                    71613400
                  1399  * CHAIN BIT IS SET                                                    71613410
3EB     40 8717   1400  CHAIN   A     MAR,NULL,ARH      (MAR)=CCW ADDRESS                     71613420
3EC     50 BB17   1401          S     MAR,MAR,TWO                                             71613430
3ED     50 BB17   1402          S     MAR,MAR,TWO       (MAR)=ADDR. OF CHAIN VALUE            71613440
3EE     08 0E09   1403          L     ARL,MR1,CS+MR     (ARL)=DEV. STATUS/DEV. NO.            71613450
                  1404  *                               FETCH CHAIN VALUE                     71613460
3EF     9F F201   1405          NI    MR1,'FF',ARL      (MR1) = DEV. NO.                      71613470
3F0     00 0A09   1406          L     ARL,MR1,SL        (ARL) = 2* DEV. NUMBER                71613480
3F1     CD 0217   1407          AI    MAR,'D0',ARL      (MAR)= ADDRESS OF SERV. POINTER       71613490
3F2     7E 0090   1408          C     MW+PRIV           STORE NEW POINTER (CHAIN VALUE)       71613500
3F3     E0 404A   1409          BT    L,AUTI02          GENERATE AN IMMEDIATE INTERRUPT OR    71613510
                  1410  *                               START ANOTHER CHANNEL PROGRAM IF      71613520
                  1411  *                               CONTINUE BIT IS SET                   71613530
3F4     F0 04E6   1412          B     END3              EXIT I/O CHANNEL                      71613540
```

```
3F5                                 1414            ORG   X'400'                                                    71613550
                                    1415  * FLOATING POINT DIVIDE ROUTINE                                           71613560
                                    1416  * ENTER AFTER VECTORING THROUGH DROM2                                      71613570
                                    1417  * (MR0,MR1)=SECOND OPERAND FRACTION                                       71613580
                                    1418  * (MR2,MDR)=FIRST OPERAND FRACTION                                        71613590
                                    1419  * MR3<0:7>=SIGN & EXP. OF SECOND OPERAND                                  71613600
                                    1420  * MR4<0:6>=EXPONENT OF SECOND OPERAND                                     71613610
                                    1421  * MR5<0:7>=SIGN & EXPONENT OF FIRST OPERAND                               71613620
                                    1422  * MR6<0:6>=EXPONENT OF FIRST OPERAND                                      71613630
                                    1423  * FIRST OPERAND IS CALLED A & SECOND OPERAND IS CALLED B                  71613640
400     88 0609                     1424  FDIV      LI    ARL,'80',CS         (ARL)= X'8000'                      71613650
401     30 220A                     1425            X     ARH,MR4,ARL         ARH<0:6>=B.EXP. IN 2'S COMPLEMENT   71613660
402     30 3206                     1426            X     MR6,MR6,ARL         MR6<0:6> = A.EXP. IN 2'S COMPLEMENT 71613670
403     50 3726                     1427            S     MR6,MR6,ARH,F       MR6<0:6> = A.EXP. - B.EXP.          71613680
404     00 280A                     1428            L     ARH,MR5                                                 71613685
405     30 1F04                     1429            X     MR4,MR3,ARH         MR4<0>=SIGN  OF THE RESULT          71613690
406     00 000A                     1430            L     ARH,MR0                                                 71613700
407     50 8750                     1431            S     NULL,NULL,ARH,CO    SET CARRY IF 2ND OPER. IS NOT ZERO  71613710
408     F2 03A7                     1432            BF    C,DIVZRO            BRANCH TO DIVZRO IF 2ND OPER. = 0   71613720
409     E1 03B0                     1433            BT    V,EXPOUF            EXP. OVERFLOW OR UNDERFLOW IF V IS SET 71613730
40A     30 3206                     1434            X     MR6,MR6,ARL         MR6<0:6>=EXP.DIFFERENCE IN EXCESS 64 71613740
40B     00 0600                     1435            L     MR0,MR0,CS                                              71613750
40C     00 0E09                     1436            L     ARL,MR1,CS                                              71613760
40D     9F F20A                     1437            NI    ARH,'FF',ARL                                            71613770
40E     20 0700                     1438            O     MR0,MR0,ARH                                             71613780
40F     00 040A                     1439            L     ARH,MR0,SR          (ARH)=MS 15 BITS OF B.FRACTION      71613790
410     80 1609                     1440            LI    ARL,'1',CS                                              71613800
411     AF F209                     1441            OI    ARL,'FF',ARL        (ARL) ='01FF'                       71613810
412     10 0A01                     1442            N     MR1,MR1,ARL         (MR1) = LS 9 BITS OF B.FRACTION     71613820
                                    1443  * DIVISOR HAS BEEN BROKEN INTO TWO PARTS BHIGH (15 BITS)&BLOW(9 BITS)     71613830
                                    1444  * (ARH)= BHIGH  (WITH SCALE FACTOR= 2**-15)                               71613840
                                    1445  * (MR1) = BLOW   (WITH SCALE FACTOR= 2**-24)                              71613850
413     00 1013                     1446            L     SRH,MR2             (SRH,SRL)=A.FRACTION                71613860
414     00 B012                     1447            L     SRL,MDR                                                 71613870
415     50 8709                     1448            S     ARL,NULL,ARH        (ARL) = 2'S COMP. OF BHIGH          71613880
416     81 0008                     1449            LI    CTR,'10'            (COUNTER)=16                        71613890
417     70 2200                     1450            C     DIV                 DIVIDE                              71613900
418     00 9003                     1451            L     MR3,SRL             (MR3)= MS QUOTIENT OF (A.FRACT/BHIGH) 71613910
                                    1452  *                                   WITH SCALE FACTOR=2**-9             71613920
419     00 8012                     1453            L     SRL,NULL            (SRH,SRL)=(2**15)*(REMAINDER)       71613930
41A     70 0800                     1454            C     SR1                                                     71613940
41B     50 8709                     1455            S     ARL,NULL,ARH        (ARL) =2'S COMP. OF BHIGH           71613950
41C     81 0008                     1456            LI    CTR,'10'            (COUNTER)=16                        71613960
41D     70 2200                     1457            C     DIV                 DIVIDE                              71613970
41E     00 9005                     1458            L     MR5,SRL             (MR5) =LS QUOT. OF (A.FRAC./BHIGH)  71613980
                                    1459  *                                   WITH SCALE FACTOR = 2**-24           71613990
                                    1460  * REMAINDER OF 2ND DIVISION IS IGNORED                                     71614000
41F     00 0813                     1461            L     SRH,MR1             (SRH) = BLOW (LS9 BITS OF B.FRAC.)  71614010
420     00 8012                     1462            L     SRL,NULL            (SRL) = 0                           71614020
421     50 8709                     1463            S     ARL,NULL,ARH        (ARL) =2'S COMPLEMENT OF BHIGH      71614030
422     81 0008                     1464            LI    CTR,'10'            (COUNTER)=16                        71614040
423     70 2200                     1465            C     DIV                 DIVIDE  ; (SRL)=16 BIT QUOTIENT     71614050
                                    1466  *                                   OF BLOW/BHIGH (CALLED C)            71614060
                                    1467  *                                                                       71614070
```

```
 424  00 9002  1468          L     MR2,SRL           (MR2)=C (WITH SCALE FACTOR=2**-25)   71614080
 425  00 9016  1469          L     MDR,SRL           (MDR) = C                            71614090
 426  00 B249  1470          L     ARL,MDR,SL+CO     (ARL) = 2*C                          71614100
 427  70 C400  1471          C     UMPY              UNSIGNED MULTIPLY                    71614110
               1472  *  (SRH,SRL) = C*C (WITH SCALE FACTOR =2**-50)                       71614120
               1473  *  ADJUST THE SCALE FACTOR OF  C*C BY SHIFTING SRH 9 PLACES RIGHT    71614130
               1474  *  IGNORE LS 16 BITS OF C*C                                         71614140
               1475  *                                                                   71614150
 428  00 9E09  1476          L     ARL,SRH,CS        SHIFT MS 16 BITS OF C*C RIGHT        71614160
 429  9F F212  1477          NI    SRL,'FF',ARL      9 PLACES                             71614170
 42A  00 9409  1478          L     ARL,SRL,SR        (APL)=C*C(WITH SCALE FACTOR=2**-25)  71614180
 42B  50 12D2  1479          S     MR2,MR2,ARL       (MR2)=C- C*C (WITH S.F.=2**-25)      71614190
 42C  00 1012  1480          L     SRL,MR2           (SRL)= C-C*C = D (SAY) S.F.=2**-25   71614200
 42D  00 2816  1481          L     MDR,MR5           (MDR)= LS QUOTIENT (S.F.=2**-24)     71614210
 42E  00 B249  1482          L     ARL,MDR,SL+CO     (ARL)= 2*LS QUOTIENT                 71614220
 42F  70 C400  1483          C     UMPY              (SRH,SRL)=D*LS QUOT.(S.F.=2**-49)    71614230
 430  00 9252  1484          L     SRL,SRL,SL+CO     SHIFT (SRH,SRL) RIGHT 15 PLACES AND  71614240
 431  00 9AC1  1485          L     MR1,SRH,SL+CO+CI  STORE RESULT IN (ARH,MR1)            71614250
 432  40 808A  1486          A     ARH,NULL,NULL,CI  (ARH,MR1)=D*LS QUOT.(S.F.=2**-34)    71614260
 433  00 1012  1487          L     SRL,MR2           (SRL) = D (WITH S.F. = 2**-25)       71614270
 434  00 1816  1488          L     MDR,MR3           (MDR)=MS QUOTIENT (WITH S.F=2**-9)   71614280
 435  00 B249  1489          L     ARL,MDR,SL+CO     (ARL)=2*MS QUOTIENT                  71614290
 436  70 C400  1490          C     UMPY              UNSIGNED MULTIPLY                    71614300
               1491  *                              (SRH,SRL)=D*MS QUOT.(S.F.=2**-34)    71614310
 437  00 0809  1492          L     ARL,MR1           (ARH,ARL)=D*LS QUOT.(S.F.=2**-34)    71614320
 438  40 9252  1493          A     SRL,SRL,ARL,CO    ACCUMULATE DOUBLE LENGTH CORRECT.    71614330
 439  40 9F93  1494          A     SRH,SRH,ARH,CI    QUOTIENT (WITH S.F.=2**-34)          71614340
 43A  80 4008  1495          LI    CTR,4             (COUNTER)=4                          71614350
 43B  70 0008  1496          C     RPT               SHIFT THE CORRECTION QUOTIENT        71614360
 43C  70 0CU0  1497          C     SR2               RIGHT 9 PLACES TO                    71614370
 43D  00 9C4A  1498          L     ARH,SRH,SR+CO     ADJUST THE SCALE FACTOR              71614380
 43E  00 9489  1499          L     ARL,SRL,SR+CI     (ARH,ARL)=CORRECTION QUOT(SF=2**-25)  71614390
 43F  00 2A05  1500          L     MR5,MR5,SL        (MR5)= LS QUOT.(WITH S.F.=2**-25)    71614400
               1501  *  (MR3,MR5) = UNCORRECTED QUOTIENT WITH S.F. = 2**-25               71614410
 440  50 2A52  1502          S     SRL,MR5,ARL,CO    (SRH,SRL)=CORRECTED QUOTIENT         71614420
 441  50 1F93  1503          S     SRH,MR3,ARH,CI    WITH SCALE FACTOR=2**-25             71614430
               1504  * FOR CORRECT RESULT IN FLOAT.POINT FRACTION FORM S.F. SHOULD BE 2**-24  71614440
               1505  * RESULTANT QUOTIENT MUST BE SHIFTED RIGHT 1 PLACE IF SRH<0:6>=0.    71614450
               1506  * RESULTANT QUOTIENT MUST BE  SHIFTED RIGHT 5 PLACES AND EXPONENT    71614460
               1507  * SHOULD BE INCREMENTED BY 1 IF SRH<0:6> IS NOT EQUAL TO 0           71614470
               1508  *                                                                   71614480
 442  80 000F  1509          LI    FLR,'0'           CLEAR FLAG REGISTER                  71614490
 443  8F E609  1510          LI    ARL,'FE',CS       (ARL)='FE00'                         71614500
 444  80 0008  1511          LI    CTR,'0'           COUNTER=0                            71614510
 445  10 9A30  1512          N     NULL,SRH,ARL,F    TEST SRH BITS 0 THRU 6               71614520
 446  F0 C44B  1513          BF    G+L,NOCOR1        NO CORRECTION IF 0                   71614530
               1514  * BITS   0 THRU 6 OF SRH DO NOT EQUAL ZERO                          71614540
 447  80 2008  1515          LI    CTR,'2'           (COUNTER)=2                          71614550
 448  80 2609  1516          LI    ARL,'2',CS        INCREMENT RESULT EXPONENT            71614560
 449  40 3246  1517          A     MR6,MR6,ARL,CO    BY ONE                               71614570
 44A  E2 03CF  1518          BT    C,EXPOF           OVERFLOW IF CARRY OUT                71614580
 44B  80 000F  1519  NOCOR1  LI    FLR,'0'           CLEAR FLAG REGISTER                  71614590
 44C  70 0840  1520          C     SR1+CO            SHIFT RESULT FRACTION RIGHT 1 PLACE  71614600
 44D  F0 039F  1521          B     FMDIV             BRANCH TO COMMON ROUTINE             71614610
```

```
                                                1522  *  COMMON FOR MH AND MHR                                                    71614620
                                                1523  * ENTER FROM D1 FOR MHR                                                   71614630
                                                1524  *  ENTER FROM D2 FOR MH                                                  71614640
                                                1525  * (ARL)= SECOND OPERAND                                                   71614650
 44E      00 D010                               1526  MH        L     NULL,YDLP1          INCR. YD FIELD BY 1                71614660
 44F      40 8216                               1527            A     MDR,NULL,ARL        (MDR)=SECOND OPERAND               71614670
 450      00 C812                               1528            L     SRL,YDL             (SRL)= FIRST HALF WORD OPERAND     71614680
 451      00 B249                               1529            L     ARL,MDR,SL+CO       (ARL)=2* SECOND OPERAND            71614690
 452      70 8400                               1530            C     MPY                 MULTIPLY  (SIGNED)                 71614700
 453      04 901B                               1531            L     YDLM1,SRL,IR        STORE SIGNED  PRODUCT ;            71614710
 454      00 9819                               1532            L     YDL,SRH             FETCH NEXT  INSTR.                 71614720
                                                1533  * COMMON  FOR   MHU & MHUR                                               71614730
                                                1534  * ENTER   AFTER D2  FOR MHU. ENTER AFTER D1  FOR MHUR                    71614740
                                                1535  * (ARL)=SECOND OPERAND                                                   71614750
 455      00 D010                               1536  MHU       L     NULL,YDLP1          INCR. YD FIELD BY 1                71614760
 456      40 8216                               1537            A     MDR,NULL,ARL        (MDR)=SECOND OPERAND               71614770
 457      00 C812                               1538            L     SRL,YDL             (SRL)=FIRST HALF WORD OPERAND      71614780
 458      00 B249                               1539            L     ARL,MDR,SL+CO       (ARL)=2* SECOND OPERAND            71614790
 459      70 C400                               1540            C     UMPY                MULTIPLY  (UNSIGNED)               71614800
 45A      04 901B                               1541            L     YDLM1,SRL,IR        STORE UNSIGNED PRODUCT ;           71614810
 45B      00 9819                               1542            L     YDL,SRH             FETCH NEXT INSTR.                  71614820
                                                1543  * DIVIDE HALF WORD                                                       71614830
                                                1544  * COMMON FOR DH(AFTER D2) AND DHR(AFTER D1)                              71614840
 45C      40 8222                               1545  DH        A     MR2,NULL,ARL,F      (MR2)=DIVISOR                      71614850
 45D      70 0020                               1546            C     SUT                 SET UTILITY FLOP                   71614855
 45E      81 0008                               1547            LI    CTR,'10'            (COUNTER)='10'                     71614860
 45F      E0 4479                               1548            BT    L,OKDIVS            DIVISOR DOES NOT REQUIRE  ANY CHANGE 71614870
                                                1549  *                                   IF IT IS  NEGATIVE                 71614880
 460      E0 8467                               1550            BT    G,COMSOR            FORM 2'S  COMPLEMENT  OF DIVISOR   71614890
                                                1551  *                                   IF IT IS  POSITIVE                 71614900
                                                1552  * FIXED   POINT DIVIDE  FAULT                                            71614910
 461      00 800F                               1553  OVDIV     L     FLR,NULL            CLEAR FLAG REGISTER                71614920
 462      81 0609                               1554            LI    ARL,'10',CS         (ARL)=X'1000'                      71614930
 463      10 3A30                               1555            N     NULL,PSWL,ARL,F     TEST BIT 3 OF PSW                  71614940
 464      F0 8279                               1556            BF    G,NBR               IGNOR DIV. FAULT IF PSW BIT 3=0    71614950
 465      84 8017                               1557            LI    MAR,'48'            (MAR)= '48'                        71614960
 466      F0 0059                               1558            B     GENSWP                                                 71614970
 467      50 8202                               1559  COMSOR    S     MR2,NULL,ARL        (MR2)=2'S COMPLEMENT OF DIVISOR    71614980
 468      00 D033                               1560            L     SRH,YDLP1,F         (SRH,SRL)=DIVIDENT ;               71614990
 469      70 0010                               1561            C     CUT                 CLEAR UTILITY FLIP FLOP            71615000
 46A      00 D812                               1562            L     SRL,YDLM1           TEST SIGN  OF THE DIVIDEND         71615010
 46B      E0 8481                               1563            BT    G,DIVOP             QUOTIENT WILL BE PLUS IF POSITIVE  71615020
 46C      00 9009                               1564            L     ARL,SRL                                                71615030
 46D      00 980A                               1565            L     ARH,SRH                                                71615040
 46E      50 8252                               1566            S     SRL,NULL,ARL,CO     (SRH,SRL) =2'S COMP. OF DIVIDEND    71615050
 46F      50 8793                               1567            S     SRH,NULL,ARH,CI                                        71615060
 470      70 0030                               1568  DIVOM     C     TUT                 TOGGLE UTILITY FLIP FLOP           71615070
 471      00 1009                               1569            L     ARL,MR2                                                71615080
 472      40 9A42                               1570            A     MR2,SRH,ARL,CO      TRAIL SUBSTRACT                    71615090
 473      E2 0461                               1571            BT    C,OVDIV             OVERFLOW  IF  CARRY IS SET          71615100
 474      70 2200                               1572            C     DIV                 DIVIDE                             71615110
 475      00 9009                               1573            L     ARL,SRL                                                71615120
 476      50 8252                               1574            S     SRL,NULL,ARL,F      2'S COMP. OF QUOTIENT              71615130
 477      E0 4487                               1575            BT    L,BCKDIV            OK  IF QUOTIENT IS NEGATIVE        71615140
```

```
478  F0 048C  1576          B     CHOVF              CHECK FOR QUOTIENT OVERFLOW          71615150
479  00 D033  1577  OKDIVS  L     SRH,YDLP1,F        LOAD DIVIDEND ;TEST SIGN ,           71615160
47A  00 8010  1578          L     NULL,NULL          NOP                                  71615170
47B  00 D812  1579          L     SRL,YDLM1                                               71615180
47C  E0 8470  1580          BT    G,DIVOM            QUOTIENT WILL BE MINUS IF POSITIVE   71615190
47D  00 9009  1581          L     ARL,SRL                                                 71615200
47E  00 980A  1582          L     ARH,SRH                                                 71615210
47F  50 8252  1583          S     SRL,NULL,ARL,CO    (SRH,SRL)=2'S COMP. OF DIVIDEND      71615220
480  50 87D3  1584          S     SRH,NULL,ARH,CI+CO                                      71615230
481  00 1009  1585  DIVOP   L     ARL,MR2                                                 71615240
482  40 9A42  1586          A     MR2,SRH,ARL,CO     TRIAL SUBTRACT                       71615250
483  E2 0461  1587          BT    C,OVDIV            OVERFLOW  IF  C IS SET              71615260
484  70 2200  1588          C     DIV                DIVIDE                               71615270
485  00 9032  1589          L     SRL,SRL,F          TEST SIGN                            71615280
486  E0 4461  1590          BT    L,OVDIV            OVER FLOW IF NEGATIVE                71615290
487  FC 248A  1591  BCKDIV  BF    UT,EXDIV           REMAINDER SHOULD BE POSITIVE IF      71615300
                1592  *                            UTILITY  FLIP  FLOP  IS RESET        71615310
488  00 9809  1593          L     ARL,SRH            2'S COMP. OF REMAINDER               71615320
489  50 8293  1594          S     SRH,NULL,ARL,CI                                         71615330
                1595  *                                                                  71615340
48A  04 981A  1596  EXDIV   L     YDLP1,SRH,IR       STORE RESULT;FETCH NEXT INST.        71615350
48B  00 9019  1597          L     YDL,SRL                                                 71615360
              1598  *  PART OF DIVIDE HALF WORD RROUTINE                                 71615370
              1599  * QUOTIENT OVER  FLOW  CHECK (DIVIDEND  AND  DIVISOR HAVE DIFF. SIGNS) 71615380
              1600  * 2'S  COMPLEMENT OF QUOTIENT MAGNITUDE  IS CONTAINED IN   SRL        71615390
48C  80 000F  1601  CHOVF   LI    FLR,0              CLEAR FLAG REGISTER                  71615400
48D  00 9032  1602          L     SRL,SRL,F          TEST QUOTIENT (2'S COMP. OF MAGN.)   71615410
48E  E0 8461  1603          BT    G,OVDIV            OVERFLOW IF POSITIVE                 71615420
48F  F0 0487  1604          B     BCKDIV                                                  71615430
              1605  * HALF WORD MODE LIST INSTRUCTIONS                                   71615440
              1606  * MAR IS ASSUMED TO CONTAIN AN EVEN ADDRESS                          71615450
              1607  * COMMON FOR HW MODE ATL AND ABL INSTRUCTIONS (HALF WORD MODE)       71615460
              1608  * ENTER AFTER D1                                                    71615470
490  40 F417  1609  ATBLH1  A     MAR,YSLX,MDR       (MAR)=LIST ADDRESS                   71615480
491  70 0010  1610          C     CUT                CLEAR UTILITY FLOP                   71615490
              1611  *       COMMON FOR LIST INST. AND I/O CHANNEL                       71615500
492  08 8010  1612  ATBLC   L     NULL,NULL,MR       START MEMEORY READ                   71615510
493  00 B600  1613          L     MR0,MDR,CS         (MR0)=NO. USED/NO. OF SLOTS IN LIST  71615520
494  50 0450  1614          S     NULL,MR0,MDR,CO    (MR0)-(MDR);MODIFY CARRY FLAG        71615530
495  F2 04CD  1615          BF    C,LSTOF            LIST OVERFLOW IF CARRY IS NOT SET    71615540
496  4F B116  1616          A     MDR,MDR,ONE,MW2    INCREMENT NO.OF SLOTS USED TALLY;    71615550
              1617  *                              START MEM. WRITE & INCR. MAR         71615560
497  8F F009  1618          LI    ARL,'FF'           (ARL)=X'00FF'                        71615570
498  EC 24D7  1619          BT    UT,CHAL1                                                71615580
499  03 8010  1620          L     NULL,NULL,D2       VECTOR THROUGH D2                    71615590
49A  18 0201  1621          N     MR1,MR0,ARL,MR     (MR1)=NO. OF SLOTS IN LIST;START MR   71615600
              1622  * ADD TO TOP OF LIST                                                71615610
              1623  * (ARL)=X'00FF'; (MR1) = NO. OF SLOTS IN THE LIST                   71615620
              1624  * ENTER  AFTER  D2                                                  71615630
49B  00 B603  1625  ATLH    L     MR3,MDR,CS         (MR3) = NEXT BOTTOM/CURRENT TOP      71615640
49C  10 1A03  1626          N     MR3,MR3,ARL        (MR3)=CURR. TOP POINTER              71615650
49D  50 1943  1627          S     MR3,MR3,ONE,CO     (MR3) = CURR. TOP POINTER-1          71615660
49E  F2 04A0  1628          BF    C,*+2              SKIP NEXT INST. IF C IS RESET        71615670
49F  50 0903  1629          S     MR3,MR1,ONE        (MR3) = MAX. SLOT NUMBER (LIST WRAP) 71615680
```

```
4A0  0F 1E16  1630          L     MDR,MR3,CS+MW2      INSERT UPDATED POINTER & RESOTRE     71615690
              1631  * ABOVE  INSTRUCTION CHANGES ONLY HIGH BYTE OF MDR SINCE MAR IS EVEN    71615700
4A1  F0 04AA  1632          B     ATBLH2              BRANCH TO COMMON ROUTINE             71615710
              1633  * ADD TO BOTTOM OF LIST                                                71615720
              1634  * (ARL)=X'00FF'; (MR1)=NO. OF SLOTS IN THE LIST                        71615730
              1635  * ENTER AFTER D2                                                       71615740
              1636  * (ARL)=X'00FF' ; (MR1) = NO. OF SLOTS IN THE LIST                    71615750
4A2  10 B203  1637  ABLH    N     MR3,MDR,ARL         (MR3)=NEXT BOTTOM POINTER            71615760
4A3  40 1902  1638          A     MR2,MR3,ONE         (MR2)=NEXT BOTTOM POINTER +1         71615770
4A4  00 080A  1639          L     ARH,MR1             (MRH)=NO. OF SLOTS IN LIST           71615780
4A5  50 1750  1640          S     NULL,MR2,ARH,CO     (MR2)-(ARH)                          71615790
4A6  E2 04A8  1641          BT    C,*+2               SKIP NEXT INST. IF C IS SET          71615800
4A7  00 8002  1642          L     MR2,NULL            (MR2)=0  (LIST WRAP OPERATION)       71615810
4A8  30 1C16  1643          X     MDR,MR3,MDR         CLEAR RIGHT HALF OF MDR              71615820
4A9  2F 1416  1644          O     MDR,MR2,MDR,MW2     INSERT UPDATED POINTER & RESTORE     71615830
              1645  * COMMON FOR   ATL AND ABL (IN HW MODE)                               71615840
              1646  * (MAR)=SLOT 0 ADDR.; (MR3)= SLOT POINTER                             71615850
4AA  00 1A09  1647  ATBLH2  L     ARL,MR3,SL          (ARL) = 2* SLOT POINTER              71615860
4AB  40 BA17  1648          A     MAR,MAR,ARL         (MAR)=REQUIRED SLOT ADDRESS          71615870
4AC  EC 24DB  1649          BT    UT,CHAL2            TO CHAL2 IF FROM I/O CHANNEL         71615880
4AD  0E C856  1650          L     MDR,YDL,MW+CO       STORE ELEMENT AND RESET CARRY        71615890
4AE  F0 0294  1651          B     FETCHJ              COPY FLR TO BOTH CC AND FETCH        71615900
              1652  *                                NEXT INSTRUCTION                     71615910
              1653  * COMMON FOR HW   MODE RTL  AND RBL  INSTRUCTIONS                      71615920
              1654  * ENTER AFTER D1                                                       71615930
4AF  40 F417  1655  RTBLH1  A     MAR,YSLX,MDR        (MAR)=LIST ADDRESS                   71615940
4B0  08 8010  1656          L     NULL,NULL,MR        START MEM. READ                      71615950
4B1  9F F400  1657          NI    MR0,'FF',MDR        (MR0)= NO. OF SLOTS USED             71615960
4B2  50 0160  1658          S     MR0,MR0,ONE,CO+F    (MR0)=(MR0)-1;MODIFY FLG             71615970
4B3  E2 04CE  1659          BT    C,LSTOUF            LIST IS EMPTY IF C IS SET            71615980
4B4  5F B116  1660          S     MDR,MDR,ONE,MW2     INCR. NO. OF SLOTS USED TALLY BY 1;  71615990
              1661  *                                INCR. MAR BY 2                       71616000
4B5  8F F009  1662          LI    ARL,'FF'            (ARL)=X'00FF'                        71616010
4B6  03 B601  1663          L     MR1,MDR,CS+D2       (MR1)=NO. USED/NO. OF SLOTS          71616020
              1664  *                                VECTOR THROUGH DROM2                 71616030
4B7  18 0A01  1665          N     MR1,MR1,ARL,MR      (MR1)=NO. OF SLOTS IN THE LIST       71616040
              1666  * REMOVE FROM BOTTOM OF LIST (HALF WORD MODE)                         71616050
              1667  * ENTER AFTER D2                                                      71616060
              1668  * (ARL)=X'00FF';(MR1)=NO. OF SLOTS IN THE LIST                       71616070
4B8  10 B202  1669  RBLH    N     MR2,MDR,ARL         (MR2)=NEXT BOTTOM POINTER            71616080
4B9  50 1143  1670          S     MR3,MR2,ONE,CO      (MR3)=NEXT BOTTOM POINTER-1          71616090
4BA  F2 04BC  1671          BF    C,*+2               SKIP NEXT INST. IF C IS RESET        71616100
4BB  50 0903  1672          S     MR3,MR1,ONE         (MR3)=MAX. SLOT NUMBER(LIST WRAP)    71616110
4BC  30 1416  1673          X     MDR,MR2,MDR         CLEAR RIGHT HALF OF MDR              71616120
4BD  2F 1C16  1674          O     MDR,MR3,MDR,MW2     INSERT UPDATED POINTER & RESTORE     71616130
4BE  F0 04C7  1675          B     RTBLH2              BRANCH TO COMMON ROUTINE             71616140
              1676  * REMOVE FROM TOP OF LIST (HALF WORD MODE)                           71616150
              1677  * ENTER AFTER D2                                                      71616160
              1678  * (ARL)=X'FF00' ; (MR1) =NO. OF SLOTS IN THE LIST                    71616170
4BF  00 B603  1679  RTLH    L     MR3,MDR,CS          (MR3)=NEXT BOTTOM/CURRENT TOP        71616180
4C0  10 1A03  1680          N     MR3,MR3,ARL         (MR3)= CURRENT TOP POINTER           71616190
4C1  40 1902  1681          A     MR2,MR3,ONE         (MR2)=CURRENT TOP POINTER+1          71616200
4C2  00 080A  1682          L     ARH,MR1             (ARH)=NO. OF SLOTS IN LIST           71616210
4C3  50 1750  1683          S     NULL,MR2,ARH,CO     (MR2)-(ARH);MODIFY CARRY             71616220
```

```
4C4       E2 04C6   1684            BT    C,*+2              SKIP NEXT INST. IF C IS SET          71616230
4C5       00 8002   1685            L     MR2,NULL           (MR2)=0                              71616240
4C6       0F 1616   1686            L     MDR,MR2,CS+MW2     INSERT UPDATED POINTER & RESTORE     71616250
                    1687  * COMMON FOR RTL AND RBL (IN HW MODE)                                   71616260
                    1688  * (MAR)=SLOT 0 ADDRESS; (MR3)=SLOT POINTER                             71616270
4C7       00 1A09   1689  RTBLH2  L     ARL,MR3,SL         (ARL)=2*SLOT POINTER                 71616280
4C8       40 BA17   1690            A     MAR,MAR,ARL        (MAR)=REQUIRED SLOT ADDRESS          71616290
4C9       08 8050   1691            L     NULL,NULL,MR+CO    START MEMORY READ & RESET CARRY      71616300
4CA       00 A017   1692            L     MAR,LOC            (MAR) = (LOC)                        71616310
4CB       05 B019   1693            L     YDL,MDR,IRUH       (DEST. REG.)=LIST ELEMENT;START      71616320
4CC       00 8010   1694            L     NULL,NULL          INSTR. FETCH & COPY FLR TO BOTH CC   71616330
                    1695  *                                  FETCH NEXT INST;COPY FLR TO BOTH CC  71616340
                    1696  *       COME HERE FOR LIST OVERFLOW                                    71616350
4CD       EC 24E1   1697  LSTOF   BT    UT,QUEOV           TO QUEOV IF FROM I/O CHANNEL         71616360
                    1698  * COME HERE FOR LIST OVERFLOW OR LIST UNDERFLOW                         71616370
                    1699  * LIST HAS NOT BEEN CHANGED                                            71616380
4CE       80 400F   1700  LSTOUF  LI    FLR,'4'            SET V FLAG & RESET OTHER FLAGS;      71616390
4CF       F0 0294   1701            B     FETCHJ             COPY FLR TO BOTH CC AND FETCH        71616400
                    1702  *                                NEXT INSTRUCTION                     71616410
                    1703  *                                                                     71616420
                    1704  *                                                                     71616430
                    1705  *                                                                     71616440
                    1706  *                                                                     71616450
                    1707  *                                                                     71616460
                    1708  *                                                                     71616470
                    1709  *                                                                     71616480
                    1710  *                                                                     71616490
                    1711  *                                                                     71616500
                    1712  * QUEUE  BIT  IS SET                                                  71616510
                    1713  *                                                                     71616520
                    1714  *                                                                     71616530
4D0       88 0017   1715  QUEUE   LI    MAR,'80'           (MAR)=ADDRESS OF QUEUE POINTER       71616540
4D1       78 0020   1716            C     SUT+MR             SET UTILITY FLOP;FETCH QUEUE POINTER 71616550
4D2       40 8704   1717            A     MR4,NULL,ARH       (MR4)=CCW ADDRESS                    71616560
4D3       00 3805   1718            L     MR5,PSWL           (MR5)=CURRENT PSW                    71616570
4D4       00 8007   1719            L     PSWL,NULL          FORCE SUPERVISOR MODE                71616580
4D5       00 B017   1720            L     MAR,MDR            (MAR)= QUEUE   ADDRESS               71616590
4D6       F0 0492   1721            B     ATBLC              BRANCH TO A ROUTINE COMMON TO        71616600
                    1722  *                                  CHANEL AND LIST INSTRUCTIONS         71616610
                    1723  *                                                                       71616620
4D7       18 0201   1724  CHAL1   N     MR1,MR0,ARL,MR     (MR1)=NO. OF SLOTS IN LIST           71616630
4D8       00 160F   1725            L     FLR,MR2,CS         (FLR)=BITS 4:7 OF CCW                71616640
4D9       E0 44A2   1726            BT    L,ABLH                                                  71616650
4DA       F0 049B   1727            B     ATLH                                                    71616660
                    1728  * RETURN FROM LIST ROUTINE                                            71616670
                    1729  * (MAR)= ADDRESS OF SLOT TO BE USED                                   71616680
                    1730  *                                                                       71616690
4D8       0E 2016   1731  CHAL2   L     MDR,MR4,MW         STORE CCW ADDRESS IN SYSTEM QUEUE    71616700
4DC       00 200A   1732            L     ARH,MR4            (ARH)=CCW ADDRESS                    71616710
4DD       50 2317   1733            S     MAR,MR4,TWO        (MAR)=CCW ADDRESS-2                  71616720
4DE       08 2807   1734            L     PSWL,MR5,MR        RESTORE PSW                          71616730
4DF       00 B001   1735            L     MR1,MDR            (MR1)= DEV NO./DEV.STATUS            71616740
4E0       F0 02DC   1736            B     TERM3                                                   71616750
                    1737  *       TERMINATION QUEUE WAS ALREADY FULL                             71616760
```

```
                        1738  *                                                                    71616770
4E1      88 A017        1739  QUEOV   LI    MAR,'8A'          (MAR)=ADDR. OF OVERFLOW TERM.POINTER 71616780
4E2      0F 2016        1740          L     MDR,MR4,MW2       STORE CCW ADDR; (MAR)='8C'           71616790
4E3      00 2807        1741          L     PSWL,MR5          RESTORE PSW                          71616800
4E4      70 0088        1742          C     JH                                                     71616810
4E5      F0 0059        1743          B     GENSWP            GO TO GENERAL PSW SWAP ROUTINE       71616820
                        1744  * EXIT  FROM  CHANEL  I/O                                            71616830
                        1745  * UTILITY FLOP IS SET IF AN ENTRY HAS BEEN MADE IN TERM. QUEUE       71616840
4E6      FC 2062        1746  END3    BF    UT,TEST1          TEST WAIT BIT IF NO ENTRY WAS MADE   71616850
                        1747  *                               IN TERMINATION QUEUE                 71616860
4E7      F4 8062        1748          BF    QUE,TEST1         TEST WAIT BIT IF PSW BIT 6 =0        71616870
4E8      88 2017        1749          LI    MAR,'82'          TAKE CHANEL I/O TERM. INTERRUPT      71616880
4E9      F0 0059        1750          B     GENSWP                                                 71616890
4EA                     1751          END                                                          71616900
```

NO ERRORS

ABLH	04A2	1726	
ABORT	0040		
ACH	028C		
ADR	009F	176	
ADRMW	0097	168	
AGB	03C7	1124	
AH	028B		
AHM	0292		
AI	0173		
AIR	016E		
AIS	028A		
AL	01B9		
AMFSLT	03BE	1357	
ASMDE	0356	1305	
ATBLC	0492	1721	
ATBLH1	0490		
ATBLH2	04AA	1632	
ATLH	049B	1727	
AUTIO	0046		
AUTIO1	0047	248	398
AUTIO2	004A	1409	
AUTIO3	0048		
AUTO1	0119	302	336
AUTO2	011C	339	
AUTO3	0121	344	
AUTO4	0124	347	
AUTO5	0128	351	
AUTO6	012B	354	
AUTO7	012F	358	
AUTO8	0132	361	
AUTOL	0139	370	375
BAL	026A		
BALR	0269		
BCKDIV	0487	1575	1604
BFRS	025D		
BFC	024D		
BFCR	0245		
BFFS	0263		
BOOTL	0115	322	
BRFSLT	03C2	1128	
BSTAT	03DD	607	
BTRS	0251		
BTC	0249		
BTCR	0241		
BTFS	0257		
BXH	0272		
BXHLE	026C		
BXLE	0275		
CE1	035E		
CE2	0370	1205	
CE3	036D	1216	
CE4	0373	1215	

SKLS	0202						
SS	0174						
SSR	016F						
START	0001						
STARTR	01AA						
STARTW	01B2						
STATR	01A5	558					
STATW	01AD	560					
STB	02A2						
STBR	029C						
STE	0300						
STH	0278						
STM	02B6	970					
SUM	0352	1147					
SVC	014F						
TERM1	02D2	993					
TERM2	02D6	1029	1035				
TERM3	02DC	1736					
TERM4	02D4	1392					
TEST	0061	289	393				
TEST1	0062	138	293	373	1746	1748	
TESTR	01B5	539					
TESTW	01B7	549					
THI	02A6						
WAIT	0063						
WAIT1	0065	113					
WAIT2	0068	112					
WD	0168						
WDR	0165						
WH	0180						
WH1	0181						
WHB	0196						
WHH	0194						
WHR	017E						
WRITE	02C1	621	986				
XH	027E						
ZRFSLT	03B9	1155	1187	1325	1327		

```
INTERDATA MODEL 7/16 WITH HSALU DROM1 05-052A13          PAGE   1
                OPT   PASS1,SQCHK
               * COPY RIGHT INTERDATA INC. JULY 1974                                    16D10040
               * DURGA AGARWAL                                                          16D10050
               * JULY 16,1974                                                           16D10060
               *                                                                        16D10070
               * THE LEAST SIGNIFICANT HEX DIGIT SPECIFIES THE CONTROL FIELD FOR       16D10080
               * THE NEXT MICROINSTRUCTION(LOCATED AT START+1)                         16D10090
               * IF LS HEX DIGIT =0 NO CONTROL ACTION                                  16D10100
               * IF LS HEX DIGIT=1 MRI CONTROL ACTION                                  16D10110
               * IF LS HEX DIGIT =2 IR  CONTROL ACTION                                 16D10120
               * IF LS HEX DIGIT=3  IRJH  CONTROL ACTION                               16D10130
               * IF THE MSB OF DROM1 IS SET,THE CORRESPONDING USER INSTRUCTION IS A    16D10140
               * PRIVILEGED INSTRUCTION . IF PSW BIT 7 IS SET (PROTECT MODE) AND       16D10150
               * MSB OF DROM1 ENTRY IS SET, LS 8 BITS OF THE ROM ADDRESS REGISTER ARE  16D10160
               * FORCED TO 1'S                                                         16D10170
               *                                                                       16D10180
               *                                                                       16D10190
000  000000          DC    0                  ILLEGAL                                  16D10200
001  002690          DC    '2690'             BALR                                     16D10210
002  002410          DC    '2410'             BTCR                                     16D10220
003  002450          DC    '2450'             BFCR                                     16D10230
               *                                                                       16D10240
004  0027C3          DC    '27C3'             NHR                                      16D10250
005  0027F3          DC    '27F3'             CLHR                                     16D10260
006  0027D3          DC    '27D3'             OHR                                      16D10270
007  0027E3          DC    '27E3'             XHR                                      16D10280
               *                                                                       16D10290
008  002803          DC    '2803'             LHR                                      16D10300
009  002840          DC    '2840'             CHR                                      16D10310
00A  0028B3          DC    '28B3'             AHR                                      16D10320
00B  0028F3          DC    '28F3'             SHR                                      16D10330
               *                                                                       16D10340
00C  0044E0          DC    '44E0'             MHR                                      16D10350
00D  0045C0          DC    '45C0'             DHR                                      16D10360
00E  0028C0          DC    '28C0'             ACHR                                     16D10370
00F  002900          DC    '2900'             SCHR                                     16D10380
               *                                                                       16D10390
               *                                                                       16D10400
010  000000          DC    0                  ILLEGAL                                  16D10410
011  000000          DC    0                  ILLEGAL                                  16D10420
012  000000          DC    0                  ILLEGAL                                  16D10430
013  000000          DC    0                  ILLEGAL                                  16D10440
014  000000          DC    0                  ILLEGAL                                  16D10450
015  000000          DC    0                  ILLEGAL                                  16D10460
016  000000          DC    0                  ILLEGAL                                  16D10470
017  000000          DC    0                  ILLEGAL                                  16D10480
018  000000          DC    0                  ILLEGAL                                  16D10490
019  000000          DC    0                  ILLEGAL                                  16D10500
01A  000000          DC    0                  ILLEGAL                                  16D10510
01B  000000          DC    0                  ILLEGAL                                  16D10520
01C  000000          DC    0                  ILLEGAL                                  16D10530
01D  000000          DC    0                  ILLEGAL                                  16D10540
01E  000000          DC    0                  ILLEGAL                                  16D10550
01F  000000          DC    0                  ILLEGAL                                  16D10560
               *                                                                       16D10570
               *                                                                       16D10580
020  002510          DC    '2510'             BTBS                                     16D10590
021  002570          DC    '2570'             BTFS                                     16D10600
```

```
022  0025D0          DC    '25D0'      BFBS          16D10610
023  002630          DC    '2630'      BFFS          16D10620
                                                     16D10630
                *
024  002BF0          DC    '2BF0'      LIS           16D10640
025  002820          DC    '2820'      LCS           16D10650
026  0028A0          DC    '28A0'      AIS           16D10660
027  0028E0          DC    '28E0'      SIS           16D10670
                                                     16D10680
                *
028  0030D0          DC    '30D0'      LER           16D10690
029  003170          DC    '3170'      CER           16D10700
02A  003170          DC    '3170'      AER           16D10710
02B  003170          DC    '3170'      SER           16D10720
                                                     16D10730
                *
02C  003170          DC    '3170'      MER           16D10740
02D  003170          DC    '3170'      DER           16D10750
02E  000000          DC    0           ILLEGAL       16D10760
02F  000000          DC    0           ILLEGAL       16D10770
                                                     16D10780
                *
                *                                    16D10790
030  000000          DC    0           ILLEGAL       16D10800
031  000000          DC    0           ILLEGAL       16D10810
032  000000          DC    0           ILLEGAL       16D10820
033  000000          DC    0           ILLEGAL       16D10830
034  000000          DC    0           ILLEGAL       16D10840
035  000000          DC    0           ILLEGAL       16D10850
036  000000          DC    0           ILLEGAL       16D10860
037  000000          DC    0           ILLEGAL       16D10870
038  000000          DC    0           ILLEGAL       16D10880
039  000000          DC    0           ILLEGAL       16D10890
03A  000000          DC    0           ILLEGAL       16D10900
03B  000000          DC    0           ILLEGAL       16D10910
03C  000000          DC    0           ILLEGAL       16D10920
03D  000000          DC    0           ILLEGAL       16D10930
03E  000000          DC    0           ILLEGAL       16D10940
03F  000000          DC    0           ILLEGAL       16D10950
                                                     16D10960
                *
                *                                    16D10970
040  0023F1          DC    '23F1'      STH           16D10980
041  0023F1          DC    '23F1'      BAL           16D10990
042  002491          DC    '2491'      BTC           16D11000
043  0024D1          DC    '24D1'      BFC           16D11010
                                                     16D11020
                *
044  002391          DC    '2391'      NH            16D11030
045  002391          DC    '2391'      CLH           16D11040
046  002391          DC    '2391'      OH            16D11050
047  002391          DC    '2391'      XH            16D11060
                                                     16D11070
                *
048  002391          DC    '2391'      LH            16D11080
049  002B91          DC    '2B91'      CH            16D11090
04A  002391          DC    '2391'      AH            16D11100
04B  002391          DC    '2391'      SH            16D11110
                                                     16D11120
                *
04C  002B91          DC    '2B91'      MH            16D11130
04D  002B91          DC    '2B91'      DH            16D11140
04E  002B91          DC    '2B91'      ACH           16D11150
04F  002B91          DC    '2B91'      SCH           16D11160
                                                     16D11170
                *
```

```
                *                                                       16D11180
050  000000          DC    0           ILLEGAL                          16D11190
051  000000          DC    0           ILLEGAL                          16D11200
052  000000          DC    0           ILLEGAL                          16D11210
053  000000          DC    0           ILLEGAL                          16D11220
054  000000          DC    0           ILLEGAL                          16D11230
055  000000          DC    0           ILLEGAL                          16D11240
056  000000          DC    0           ILLEGAL                          16D11250
057  000000          DC    0           ILLEGAL                          16D11260
058  000000          DC    0           ILLEGAL                          16D11270
059  000000          DC    0           ILLEGAL                          16D11280
05A  000000          DC    0           ILLEGAL                          16D11290
05B  000000          DC    0           ILLEGAL                          16D11300
05C  000000          DC    0           ILLEGAL                          16D11310
05D  000000          DC    0           ILLEGAL                          16D11320
05E  000000          DC    0           ILLEGAL                          16D11330
05F  000000          DC    0           ILLEGAL                          16D11340
                *                                                       16D11350
                *                                                       16D11360
060  003001          DC    '3001'      STE                              16D11370
061  0023F1          DC    '23F1'      AHM                              16D11380
062  000000          DC    0           ILLEGAL                          16D11390
063  000000          DC    0           ILLEGAL                          16D11400
                *                                                       16D11410
064  004901          DC    '4901'      ATL                              16D11420
065  004901          DC    '4901'      ABL                              16D11430
066  004AF1          DC    '4AF1'      RTL                              16D11440
067  004AF1          DC    '4AF1'      RBL                              16D11450
                *                                                       16D11460
068  0030B1          DC    '30B1'      LE                               16D11470
069  003151          DC    '3151'      CE                               16D11480
06A  003151          DC    '3151'      AE                               16D11490
06B  003151          DC    '3151'      SE                               16D11500
                *                                                       16D11510
06C  003151          DC    '3151'      ME                               16D11520
06D  003151          DC    '3151'      DE                               16D11530
06E  000000          DC    0           ILLEGAL                          16D11540
06F  000000          DC    0           ILLEGAL                          16D11550
                *                                                       16D11560
                *                                                       16D11570
070  000000          DC    0           ILLEGAL                          16D11580
071  000000          DC    0           ILLEGAL                          16D11590
072  000000          DC    0           ILLEGAL                          16D11600
073  000000          DC    0           ILLEGAL                          16D11610
074  000000          DC    0           ILLEGAL                          16D11620
075  000000          DC    0           ILLEGAL                          16D11630
076  000000          DC    0           ILLEGAL                          16D11640
077  000000          DC    0           ILLEGAL                          16D11650
078  000000          DC    0           ILLEGAL                          16D11660
079  000000          DC    0           ILLEGAL                          16D11670
07A  000000          DC    0           ILLEGAL                          16D11680
07B  000000          DC    0           ILLEGAL                          16D11690
07C  000000          DC    0           ILLEGAL                          16D11700
07D  000000          DC    0           ILLEGAL                          16D11710
07E  000000          DC    0           ILLEGAL                          16D11720
07F  000000          DC    0           ILLEGAL                          16D11730
                *                                                       16D11740
```

```
                *
080 000000          DC    0          ILLEGAL                    16D11750
081 000000          DC    0          ILLEGAL                    16D11760
082 000000          DC    0          ILLEGAL                    16D11770
083 000000          DC    0          ILLEGAL                    16D11780
084 000000          DC    0          ILLEGAL                    16D11790
085 000000          DC    0          ILLEGAL                    16D11800
086 000000          DC    0          ILLEGAL                    16D11810
087 000000          DC    0          ILLEGAL                    16D11820
088 000000          DC    0          ILLEGAL                    16D11830
089 000000          DC    0          ILLEGAL                    16D11840
08A 000000          DC    0          ILLEGAL                    16D11850
08B 000000          DC    0          ILLEGAL                    16D11860
08C 000000          DC    0          ILLEGAL                    16D11870
08D 000000          DC    0          ILLEGAL                    16D11880
08E 000000          DC    0          ILLEGAL                    16D11890
08F 000000          DC    0          ILLEGAL                    16D11900
                *                                               16D11910
                *                                               16D11920
090 002020          DC    '2020'     SRLS                       16D11930
091 002060          DC    '2060'     SLLS                       16D11940
092 0029C0          DC    '29C0'     STBR                       16D11950
093 0029A0          DC    '29A0'     LBR                        16D11960
                *                                               16D11970
094 002AD0          DC    '2AD0'     EXBR                       16D11980
095 009480          DC    '9480'     EPSR                       16D11990
096 0099F0          DC    '99F0'     WBR                        16D12000
097 0099F0          DC    '99F0'     RBR                        16D12010
                *                                               16D12020
098 0097E0          DC    '97E0'     WHR                        16D12030
099 0098A0          DC    '98A0'     RHR                        16D12040
09A 009650          DC    '9650'     WDR                        16D12050
09B 0095C0          DC    '95C0'     RDR                        16D12060
09C 004550          DC    '4550'     MHUR                       16D12070
09D 0096F0          DC    '96F0'     SSR                        16D12080
09E 009780          DC    '9780'     OCR                        16D12090
09F 0096E0          DC    '96E0'     AIR                        16D12100
                *                                               16D12110
                *                                               16D12120
0A0 000000          DC    0          ILLEGAL                    16D12130
0A1 000000          DC    0          ILLEGAL                    16D12140
0A2 000000          DC    0          ILLEGAL                    16D12150
0A3 000000          DC    0          ILLEGAL                    16D12160
0A4 000000          DC    0          ILLEGAL                    16D12170
0A5 000000          DC    0          ILLEGAL                    16D12180
0A6 000000          DC    0          ILLEGAL                    16D12190
0A7 000000          DC    0          ILLEGAL                    16D12200
0A8 000000          DC    0          ILLEGAL                    16D12210
0A9 000000          DC    0          ILLEGAL                    16D12220
0AA 000000          DC    0          ILLEGAL                    16D12230
0AB 000000          DC    0          ILLEGAL                    16D12240
0AC 000000          DC    0          ILLEGAL                    16D12250
0AD 000000          DC    0          ILLEGAL                    16D12260
0AE 000000          DC    0          ILLEGAL                    16D12270
0AF 000000          DC    0          ILLEGAL                    16D12280
                *                                               16D12290
                *                                               16D12300
                                                                16D12310
```

```
0B0  000000          DC    0          ILLEGAL                16D12320
0B1  000000          DC    0          ILLEGAL                16D12330
0B2  000000          DC    0          ILLEGAL                16D12340
0B3  000000          DC    0          ILLEGAL                16D12350
0B4  000000          DC    0          ILLEGAL                16D12360
0B5  000000          DC    0          ILLEGAL                16D12370
0B6  000000          DC    0          ILLEGAL                16D12380
0B7  000000          DC    0          ILLEGAL                16D12390
0B8  000000          DC    0          ILLEGAL                16D12400
0B9  000000          DC    0          ILLEGAL                16D12410
0BA  000000          DC    0          ILLEGAL                16D12420
0BB  000000          DC    0          ILLEGAL                16D12430
0BC  000000          DC    0          ILLEGAL                16D12440
0BD  000000          DC    0          ILLEGAL                16D12450
0BE  000000          DC    0          ILLEGAL                16D12460
0BF  000000          DC    0          ILLEGAL                16D12470
                *                                           16D12480
                *                                           16D12490
0C0  0026C1          DC    '26C1'     BXH                    16D12500
0C1  0026C1          DC    '26C1'     BXLE                   16D12510
0C2  00A3F1          DC    'A3F1'     LPSW                   16D12520
0C3  0023D1          DC    '23D1'     THI                    16D12530
                *                                           16D12540
0C4  0023D1          DC    '23D1'     NHI                    16D12550
0C5  0023D1          DC    '23D1'     CLHI                   16D12560
0C6  0023D1          DC    '23D1'     OHI                    16D12570
0C7  0023D1          DC    '23D1'     XHI                    16D12580
                *                                           16D12590
0C8  0023D1          DC    '23D1'     LHI                    16D12600
0C9  002BD1          DC    '2BD1'     CHI                    16D12610
0CA  0023D1          DC    '23D1'     AHI                    16D12620
0CB  0023D1          DC    '23D1'     SHI                    16D12630
                *                                           16D12640
0CC  002001          DC    '2001'     SRHL                   16D12650
0CD  002001          DC    '2001'     SLHL                   16D12660
0CE  002001          DC    '2001'     SRHA                   16D12670
0CF  002001          DC    '2001'     SLHA                   16D12680
                *                                           16D12690
                *                                           16D12700
0D0  002AF1          DC    '2AF1'     STM                    16D12710
0D1  002AF1          DC    '2AF1'     LM                     16D12720
0D2  0023F1          DC    '23F1'     STB                    16D12730
0D3  0023F1          DC    '23F1'     LB                     16D12740
                *                                           16D12750
0D4  0023F1          DC    '23F1'     CLB                    16D12760
0D5  00A3F1          DC    'A3F1'     AL                     16D12770
0D6  009991          DC    '9991'     WB                     16D12780
0D7  009991          DC    '9991'     RB                     16D12790
                *                                           16D12800
0D8  009801          DC    '9801'     WH                     16D12810
0D9  009921          DC    '9921'     RH                     16D12820
0DA  00A3F1          DC    'A3F1'     WD                     16D12830
0DB  00A3F1          DC    'A3F1'     RD                     16D12840
                *                                           16D12850
0DC  002B91          DC    '2B91'     MHU                    16D12860
0DD  00A3F1          DC    'A3F1'     SS                     16D12870
0DE  00A3F1          DC    'A3F1'     OC                     16D12880
```

```
0DF  00A3F1          DC    'A3F1'              AI                    16D12890
                     *                                               16D12900
                     *                                               16D12910
0E0  000000          DC    0                   ILLEGAL               16D12920
0E1  0023F1          DC    '23F1'              SVC                   16D12930
0E2  0094C1          DC    '94C1'              SINT                  16D12940
0E3  000000          DC    0                   ILLEGAL               16D12950
                     *                                               16D12960
0E4  000000          DC    0                   ILLEGAL               16D12970
0E5  000000          DC    0                   ILLEGAL               16D12980
0E6  000000          DC    0                   ILLEGAL               16D12990
0E7  000000          DC    0                   ILLEGAL               16D13000
                     *                                               16D13010
0E8  000000          DC    0                   ILLEGAL               16D13020
0E9  000000          DC    0                   ILLEGAL               16D13030
0EA  002151          DC    '2151'              RRL                   16D13040
0EB  002151          DC    '2151'              RLL                   16D13050
                     *                                               16D13060
0EC  002151          DC    '2151'              SRL                   16D13070
0ED  002151          DC    '2151'              SLL                   16D13080
0EE  002151          DC    '2151'              SRA                   16D13090
0EF  002151          DC    '2151'              SLA                   16D13100
                     *                                               16D13110
                     *                                               16D13120
0F0  000000          DC    0                   ILLEGAL               16D13130
0F1  000000          DC    0                   ILLEGAL               16D13140
0F2  000000          DC    0                   ILLEGAL               16D13150
0F3  000000          DC    0                   ILLEGAL               16D13160
0F4  000000          DC    0                   ILLEGAL               16D13170
0F5  000000          DC    0                   ILLEGAL               16D13180
0F6  000000          DC    0                   ILLEGAL               16D13190
0F7  000000          DC    0                   ILLEGAL               16D13200
0F8  000000          DC    0                   ILLEGAL               16D13210
0F9  000000          DC    0                   ILLEGAL               16D13220
0FA  000000          DC    0                   ILLEGAL               16D13230
0FB  000000          DC    0                   ILLEGAL               16D13240
0FC  000000          DC    0                   ILLEGAL               16D13250
0FD  000000          DC    0                   ILLEGAL               16D13260
0FE  000000          DC    0                   ILLEGAL               16D13270
0FF  000000          DC    0                   ILLEGAL               16D13280
                     *                                               16D13290
                     *                                               16D13300
                     *                                               16D13310
```



```
100                  END                                             16D13320
```

```
                OPT   PASS1,SQCHK
               * COPYRIGHT INTERDATA INC. JULY 1974                                  16D20040
               * DURGA AGARWAL                                                       16D20050
               * JULY 16,1974                                                        16D20060
               * THE LEAST SIGNIFICANT 11 BITS SPECIFY ROM ADDRESS                   16D20070
               * IF THE MSB OF DROM2 ENTRY IS SET AND THE HALFWORD I/O LINE IS       16D20080
               * NOT ACTIVE, BIT 14 OF ROM ADDRESS REGISER  IS FORCED TO 1           16D20090
               *                                                                     16D20100
000  000000          DC    0              ILLEGAL                                    16D20110
001  000000          DC    0              BALR                                       16D20120
002  000000          DC    0              BTCR                                       16D20130
003  000000          DC    0              BFCR                                       16D20140
               *                                                                     16D20150
004  000000          DC    0              NHR                                        16D20160
005  000000          DC    0              CLHR                                       16D20170
006  000000          DC    0              OHR                                        16D20180
007  000000          DC    0              XHR                                        16D20190
               *                                                                     16D20200
008  000000          DC    0              LHR                                        16D20210
009  000000          DC    0              CHR                                        16D20220
00A  000000          DC    0              AHR                                        16D20230
00B  000000          DC    0              SHR                                        16D20240
               *                                                                     16D20250
00C  000000          DC    0              MHR                                        16D20260
00D  000000          DC    0              DHR                                        16D20270
00E  000000          DC    0              ACHR                                       16D20280
00F  000000          DC    0              SCHR                                       16D20290
               *                                                                     16D20300
               *                                                                     16D20310
010  000000          DC    0              ILLEGAL                                    16D20320
011  000000          DC    0              ILLEGAL                                    16D20330
012  000000          DC    0              ILLEGAL                                    16D20340
013  000000          DC    0              ILLEGAL                                    16D20350
014  000000          DC    0              ILLEGAL                                    16D20360
015  000000          DC    0              ILLEGAL                                    16D20370
016  000000          DC    0              ILLEGAL                                    16D20380
017  000000          DC    0              ILLEGAL                                    16D20390
018  000000          DC    0              ILLEGAL                                    16D20400
019  000000          DC    0              ILLEGAL                                    16D20410
01A  000000          DC    0              ILLEGAL                                    16D20420
01B  000000          DC    0              ILLEGAL                                    16D20430
01C  000000          DC    0              ILLEGAL                                    16D20440
01D  000000          DC    0              ILLEGAL                                    16D20450
01E  000000          DC    0              ILLEGAL                                    16D20460
01F  000000          DC    0              ILLEGAL                                    16D20470
               *                                                                     16D20480
               *                                                                     16D20490
020  000000          DC    0              BTBS                                       16D20500
021  000000          DC    0              BTFS                                       16D20510
022  000000          DC    0              BFBS                                       16D20520
023  000000          DC    0              BFFS                                       16D20530
               *                                                                     16D20540
024  000000          DC    0              LIS                                        16D20550
025  000000          DC    0              LCS                                        16D20560
026  000000          DC    0              AIS                                        16D20570
027  000000          DC    0              SIS                                        16D20580
               *                                                                     16D20590
028  00033F          DC    '33F'          LER                                        16D20600
```

```
029 00035E          DC    '35E'          CER                  16D20610
02A 000324          DC    '324'          AER                  16D20620
02B 000322          DC    '322'          SER                  16D20630
                    *                                         16D20640
02C 000375          DC    '375'          MER                  16D20650
02D 000400          DC    '400'          DER                  16D20660
02E 000000          DC    0              ILLEGAL              16D20670
02F 000000          DC    0              ILLEGAL              16D20680
                    *                                         16D20690
                    *                                         16D20700
030 000000          DC    0              ILLEGAL              16D20710
031 000000          DC    0              ILLEGAL              16D20720
032 000000          DC    0              ILLEGAL              16D20730
033 000000          DC    0              ILLEGAL              16D20740
034 000000          DC    0              ILLEGAL              16D20750
035 000000          DC    0              ILLEGAL              16D20760
036 000000          DC    0              ILLEGAL              16D20770
037 000000          DC    0              ILLEGAL              16D20780
038 000000          DC    0              ILLEGAL              16D20790
039 000000          DC    0              ILLEGAL              16D20800
03A 000000          DC    0              ILLEGAL              16D20810
03B 000000          DC    0              ILLEGAL              16D20820
03C 000000          DC    0              ILLEGAL              16D20830
03D 000000          DC    0              ILLEGAL              16D20840
03E 000000          DC    0              ILLEGAL              16D20850
03F 000000          DC    0              ILLEGAL              16D20860
                    *                                         16D20870
                    *                                         16D20880
040 000278          DC    '278'          STH                  16D20890
041 00026A          DC    '26A'          BAL                  16D20900
042 000000          DC    0              BTC                  16D20910
043 000000          DC    0              BFC                  16D20920
                    *                                         16D20930
044 00027C          DC    '27C'          NH                   16D20940
045 00027F          DC    '27F'          CLH                  16D20950
046 00027D          DC    '27D'          OH                   16D20960
047 00027E          DC    '27E'          XH                   16D20970
                    *                                         16D20980
048 000280          DC    '280'          LH                   16D20990
049 000284          DC    '284'          CH                   16D21000
04A 00028B          DC    '28B'          AH                   16D21010
04B 00028F          DC    '28F'          SH                   16D21020
                    *                                         16D21030
04C 00044E          DC    '44E'          MH                   16D21040
04D 00045C          DC    '45C'          DH                   16D21050
04E 00028C          DC    '28C'          ACH                  16D21060
04F 000290          DC    '290'          SCH                  16D21070
                    *                                         16D21080
                    *                                         16D21090
050 000000          DC    0              ILLEGAL              16D21100
051 000000          DC    0              ILLEGAL              16D21110
052 000000          DC    0              ILLEGAL              16D21120
053 000000          DC    0              ILLEGAL              16D21130
054 000000          DC    0              ILLEGAL              16D21140
055 000000          DC    0              ILLEGAL              16D21150
056 000000          DC    0              ILLEGAL              16D21160
057 000000          DC    0              ILLEGAL              16D21170
```

```
058  000000          DC    0          ILLEGAL                    16D21180
059  000000          DC    0          ILLEGAL                    16D21190
05A  000000          DC    0          ILLEGAL                    16D21200
05B  000000          DC    0          ILLEGAL                    16D21210
05C  000000          DC    0          ILLEGAL                    16D21220
05D  000000          DC    0          ILLEGAL                    16D21230
05E  000000          DC    0          ILLEGAL                    16D21240
05F  000000          DC    0          ILLEGAL                    16D21250
                *                                                 16D21260
                *                                                 16D21270
060  000000          DC    0          STE                        16D21280
061  000292          DC    '292'      AHM                        16D21290
062  000000          DC    0          ILLEGAL                    16D21300
063  000000          DC    0          ILLEGAL                    16D21310
                *                                                 16D21320
064  00049B          DC    '49B'      ATL                        16D21330
065  0004A2          DC    '4A2'      ABL                        16D21340
066  0004BF          DC    '4BF'      RTL                        16D21350
067  0004B8          DC    '4B8'      RBL                        16D21360
                *                                                 16D21370
068  00033F          DC    '33F'      LE                         16D21380
069  00035E          DC    '35E'      CE                         16D21390
06A  000324          DC    '324'      AE                         16D21400
06B  000322          DC    '322'      SE                         16D21410
                *                                                 16D21420
06C  000375          DC    '375'      ME                         16D21430
06D  000400          DC    '400'      DE                         16D21440
06E  000000          DC    0          ILLEGAL                    16D21450
06F  000000          DC    0          ILLEGAL                    16D21460
                *                                                 16D21470
                *                                                 16D21480
070  000000          DC    0          ILLEGAL                    16D21490
071  000000          DC    0          ILLEGAL                    16D21500
072  000000          DC    0          ILLEGAL                    16D21510
073  000000          DC    0          ILLEGAL                    16D21520
074  000000          DC    0          ILLEGAL                    16D21530
075  000000          DC    0          ILLEGAL                    16D21540
076  000000          DC    0          ILLEGAL                    16D21550
077  000000          DC    0          ILLEGAL                    16D21560
078  000000          DC    0          ILLEGAL                    16D21570
079  000000          DC    0          ILLEGAL                    16D21580
07A  000000          DC    0          ILLEGAL                    16D21590
07B  000000          DC    0          ILLEGAL                    16D21600
07C  000000          DC    0          ILLEGAL                    16D21610
07D  000000          DC    0          ILLEGAL                    16D21620
07E  000000          DC    0          ILLEGAL                    16D21630
07F  000000          DC    0          ILLEGAL                    16D21640
                *                                                 16D21650
                *                                                 16D21660
080  000000          DC    0          ILLEGAL                    16D21670
081  000000          DC    0          ILLEGAL                    16D21680
082  000000          DC    0          ILLEGAL                    16D21690
083  000000          DC    0          ILLEGAL                    16D21700
084  000000          DC    0          ILLEGAL                    16D21710
085  000000          DC    0          ILLEGAL                    16D21720
086  000000          DC    0          ILLEGAL                    16D21730
087  000000          DC    0          ILLEGAL                    16D21740
```

```
088 000000          DC    0          ILLEGAL                    16D21750
089 000000          DC    0          ILLEGAL                    16D21760
08A 000000          DC    0          ILLEGAL                    16D21770
08B 000000          DC    0          ILLEGAL                    16D21780
08C 000000          DC    0          ILLEGAL                    16D21790
08D 000000          DC    0          ILLEGAL                    16D21800
08E 000000          DC    0          ILLEGAL                    16D21810
08F 000000          DC    0          ILLEGAL                    16D21820
                *                                               16D21830
                *                                               16D21840
090 000000          DC    0          SRLS                       16D21850
091 000000          DC    0          SLLS                       16D21860
092 000000          DC    0          STBR                       16D21870
093 000000          DC    0          LBR                        16D21880
                *                                               16D21890
094 000000          DC    0          EXBR                       16D21900
095 000000          DC    0          EPSR                       16D21910
096 000182          DC    '182'      WBR                        16D21920
097 0001AA          DC    '1AA'      RBR                        16D21930
                *                                               16D21940
098 000994          DC    '994'      WHR                        16D21950
099 00098C          DC    '98C'      RHR                        16D21960
09A 000000          DC    0          WDR                        16D21970
09B 000000          DC    0          RDR                        16D21980
                *                                               16D21990
09C 000000          DC    0          MHUR                       16D22000
09D 000000          DC    0          SSR                        16D22010
09E 000000          DC    0          OCR                        16D22020
09F 000000          DC    0          AIR                        16D22030
                *                                               16D22040
                *                                               16D22050
0A0 000000          DC    0          ILLEGAL                    16D22060
0A1 000000          DC    0          ILLEGAL                    16D22070
0A2 000000          DC    0          ILLEGAL                    16D22080
0A3 000000          DC    0          ILLEGAL                    16D22090
0A4 000000          DC    0          ILLEGAL                    16D22100
0A5 000000          DC    0          ILLEGAL                    16D22110
0A6 000000          DC    0          ILLEGAL                    16D22120
0A7 000000          DC    0          ILLEGAL                    16D22130
0A8 000000          DC    0          ILLEGAL                    16D22140
0A9 000000          DC    0          ILLEGAL                    16D22150
0AA 000000          DC    0          ILLEGAL                    16D22160
0AB 000000          DC    0          ILLEGAL                    16D22170
0AC 000000          DC    0          ILLEGAL                    16D22180
0AD 000000          DC    0          ILLEGAL                    16D22190
0AE 000000          DC    0          ILLEGAL                    16D22200
0AF 000000          DC    0          ILLEGAL                    16D22210
                *                                               16D22220
                *                                               16D22230
0B0 000000          DC    0          ILLEGAL                    16D22240
0B1 000000          DC    0          ILLEGAL                    16D22250
0B2 000000          DC    0          ILLEGAL                    16D22260
0B3 000000          DC    0          ILLEGAL                    16D22270
0B4 000000          DC    0          ILLEGAL                    16D22280
0B5 000000          DC    0          ILLEGAL                    16D22290
0B6 000000          DC    0          ILLEGAL                    16D22300
0B7 000000          DC    0          ILLEGAL                    16D22310
```

```
0B8  000000          DC    0          ILLEGAL                    16D22320
0B9  000000          DC    0          ILLEGAL                    16D22330
0BA  000000          DC    0          ILLEGAL                    16D22340
0BB  000000          DC    0          ILLEGAL                    16D22350
0BC  000000          DC    0          ILLEGAL                    16D22360
0BD  000000          DC    0          ILLEGAL                    16D22370
0BE  000000          DC    0          ILLEGAL                    16D22380
0BF  000000          DC    0          ILLEGAL                    16D22390
                *                                                 16D22400
                *                                                 16D22410
0C0  000272          DC    '272'      BXH                        16D22420
0C1  000275          DC    '275'      BXLE                       16D22430
0C2  000146          DC    '146'      LPSW                       16D22440
0C3  0002A6          DC    '2A6'      THI                        16D22450
0C4  00027C          DC    '27C'      NHI                        16D22460
0C5  00027F          DC    '27F'      CLHI                       16D22470
0C6  00027D          DC    '27D'      OHI                        16D22480
0C7  00027E          DC    '27E'      XHI                        16D22490
0C8  000280          DC    '280'      LHI                        16D22500
0C9  000284          DC    '284'      CHI                        16D22510
0CA  00028B          DC    '28B'      AHI                        16D22520
0CB  00028F          DC    '28F'      SHI                        16D22530
0CC  000203          DC    '203'      SRHL                       16D22540
0CD  000207          DC    '207'      SLHL                       16D22550
0CE  000211          DC    '211'      SRHA                       16D22560
0CF  00020B          DC    '20B'      SLHA                       16D22570
                *                                                 16D22580
                *                                                 16D22590
0D0  0002B6          DC    '2B6'      STM                        16D22600
0D1  0002B2          DC    '2B2'      LM                         16D22610
0D2  0002A2          DC    '2A2'      STB                        16D22620
0D3  000297          DC    '297'      LB                         16D22630
                *                                                 16D22640
0D4  0002A7          DC    '2A7'      CLB                        16D22650
0D5  0001B9          DC    '1B9'      AL                         16D22660
0D6  0001B2          DC    '1B2'      WB                         16D22670
0D7  0001AA          DC    '1AA'      RB                         16D22680
                *                                                 16D22690
0D8  000994          DC    '994'      WH                         16D22700
0D9  000984          DC    '984'      RH                         16D22710
0DA  000168          DC    '168'      WD                         16D22720
0DB  00015F          DC    '15F'      RD                         16D22730
                *                                                 16D22740
0DC  000455          DC    '455'      MHU                        16D22750
0DD  000174          DC    '174'      SS                         16D22760
0DE  000178          DC    '178'      OC                         16D22770
0DF  000173          DC    '173'      AI                         16D22780
                *                                                 16D22790
                *                                                 16D22800
0E0  000000          DC    0          ILLEGAL                    16D22810
0E1  00014F          DC    '14F'      SVC                        16D22820
0E2  000000          DC    0          SINT                       16D22830
0E3  000000          DC    0          ILLEGAL                    16D22840
                *                                                 16D22850
0E4  000000          DC    0          ILLEGAL                    16D22860
0E5  000000          DC    0          ILLEGAL                    16D22870
0E6  000000          DC    0          ILLEGAL                    16D22880
```

```
0E7  000000            DC    0              ILLEGAL                    16D22890
                *                                                       16D22900
0E8  000000            DC    0              ILLEGAL                    16D22910
0E9  000000            DC    0              ILLEGAL                    16D22920
0EA  00022D            DC    '22D'          RRL                        16D22930
0EB  000233            DC    '233'          RLL                        16D22940
                *                                                       16D22950
0EC  000218            DC    '218'          SRL                        16D22960
0ED  00021C            DC    '21C'          SLL                        16D22970
0EE  000220            DC    '220'          SRA                        16D22980
0EF  000226            DC    '226'          SLA                        16D22990
                *                                                       16D23000
                *                                                       16D23010
0F0  000000            DC    0              ILLEGAL                    16D23020
0F1  000000            DC    0              ILLEGAL                    16D23030
0F2  000000            DC    0              ILLEGAL                    16D23040
0F3  000000            DC    0              ILLEGAL                    16D23050
0F4  000000            DC    0              ILLEGAL                    16D23060
0F5  000000            DC    0              ILLEGAL                    16D23070
0F6  000000            DC    0              ILLEGAL                    16D23080
0F7  000000            DC    0              ILLEGAL                    16D23090
0F8  000000            DC    0              ILLEGAL                    16D23100
0F9  000000            DC    0              ILLEGAL                    16D23110
0FA  000000            DC    0              ILLEGAL                    16D23120
0FB  000000            DC    0              ILLEGAL                    16D23130
0FC  000000            DC    0              ILLEGAL                    16D23140
0FD  000000            DC    0              ILLEGAL                    16D23150
0FE  000000            DC    0              ILLEGAL                    16D23160
0FF  000000            DC    0              ILLEGAL                    16D23170
                *                                                       16D23180
                *                                                       16D23190
```



```
100                    END                                             16D23200
```

HEXADECIMAL DISPLAY

M71-102
HEXADECIMAL DISPLAY
INFORMATION SPECIFICATION

1. INTRODUCTION

The optional Hexadecimal Display Panel provides a means to manually control the Processor, interrogate and display various Processor registers and machine status, set and display Processor memory locations, and may be programmed as an I/O device by the user.

This specification describes the 09-065F02 Hexadecimal Display Panel (Product Number M71-102). It is also applicable to the 09-065F01 Binary Display Panel (Product Number M71-101), which is identical to the Hexadecimal Display Panel except for the omission of the hexadecimal indicators. The Hexadecimal Display Panel provides the following functions:

Displays five bytes of programmable digital information.

Registers and displays five hexadecimal digits of manually entered keyboard data.

Displays the WAIT and Power (PWR) indicators for the Processor.

Provides a 26 key control keyboard for manual input to the display.

Provides two bytes of unbuffered Switch Register data to the Processor.

Provides one byte of status to the Processor.

Provides a three position OFF-ON-LOCK key type switch capable of switching three separate power supply control lines.

Provides a control signal to the Processor that the display requires micro-program support.

2. GENERAL DESCRIPTION

A complete description of the operation of the Hexadecimal Display Panel is provided in the appropriate User's Manual. This specification describes the display from a maintenance view point. Figure 1 shows the Hexadecimal Display Panel.

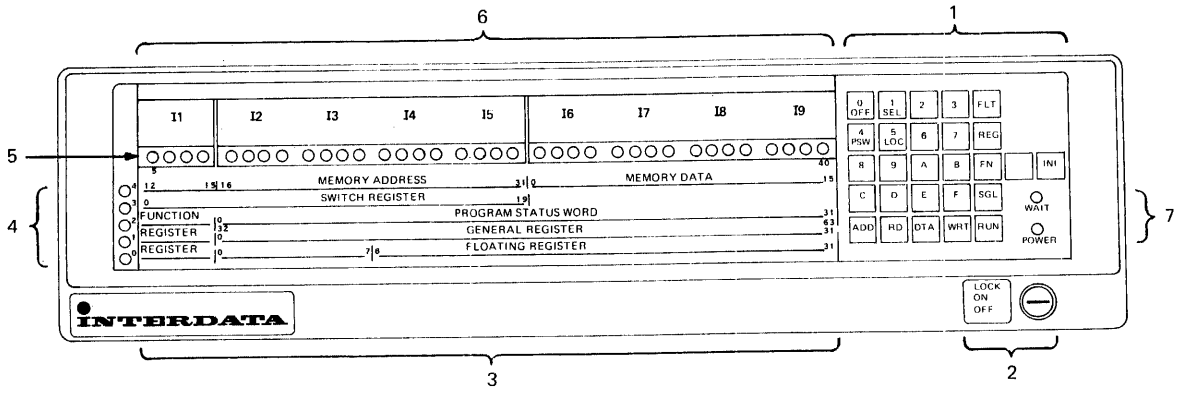


Figure 1. Hexadecimal Display Panel

Various parts of the Hexadecimal Display Panel in Figure 1 are numbered to correlate to the following descriptions.

1. Control Keyboard. The keyboard is the operators manual input to the Processor. The function of the specific keys are:

DTA — The function of the Data (DTA) key is to clear the Switch Register, connect the Switch Register to the display indicators, and enable hexadecimal data to be entered into the register. The Switch Register remains enabled and connected to the display indicators until any non-hexadecimal key other than DTA is depressed.

Hexadecimal Keys 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F supply data to the Switch Register when it is enabled, and the function number or register number for the Processor supported display (see Section 2.2).

ADD — The Address (ADD) key causes the Processor to read the five hexadecimal characters of the Switch Register, store them in the address portion of the Program Status Word (PSW), and display PSW 32:63 on the indicators.

RD — The Read (RD) key causes the Processor to read the memory location specified by the PSW, increment the PSW address by two, and display on the indicators the new address and the data read from memory.

WRT — Depressing the Write (WRT) key causes the data contained in the Switch Register to be written into the address specified by the PSW, the PSW to be incremented by two, and the new address and the data written to be displayed on the indicators.

FLT — Depressing the Floating-Point Register (FLT) key, followed by any hexadecimal key n, causes Floating-Point Register n to be displayed on the indicators.

REG — Depressing the Register (REG) key, followed by any hexadecimal key n, causes general register n to be displayed.

FN — Depressing the Function (FN) key, followed by any hexadecimal key n, causes the Processor to perform "Function n" as described in the appropriate User's Manual.

SGL — Depressing the Single Step (SGL) key causes the Processor to execute one user instruction and display the last register or function selected.

RUN — Depressing the Run (RUN) key causes the Processor to enter the Run mode at the address specified by the PSW.

INI — Depressing the Initialize (INI) key initializes the Processor.

SEL — Depress DTA, then 0 or F, for selection of Register Set 0 or 1 respectively. Then depress the Function (FN) Key followed by SEL to enable the selected register set to be displayed.

NOTE

The display requires support from the micro-program for all functions other than entering or displaying Switch Register data.

2. OFF-ON-LOCK Key Operated Locking Switch. This switch controls the power to the Processor and allows the keyboard to be completely disabled in the LOCK position.

3. Indicator Formats. These formats aid the user in interpreting the display indicators.

4. Format Selectors L0:4. Light Emitting Diode (LED) indicators L0:4 determine the format to be used to interpret display indicators L5:40.

5. Display Indicators L5:40. These LED indicators are used to display the PSW, general registers, etc., as described by the indicator formats.

6. Display Indicators I1:9. These indicators display the corresponding values displayed on L5:40 in the hexadecimal format.

7. WAIT and PWR. These indicators are illuminated when Processor is in the Wait state and Power is supplied to the Processor.

2.1 Switch Register Entries

When the operator is manipulating the Switch Register, there is no interaction between the display and the Processor. Data is entered into this register by first depressing the DTA key. This operation clears the Switch Register; connects the Switch Register to L5:24 of the display, and allows subsequent hexadecimal keyboard entries to be left shifted into the least significant digit of the resigter. The register is disconnected from the display and disabled when any non-hexadecimal key other than DTA is depressed. The register can be momentarily examined when it is disabled without affecting the Processor operation by depressing any hexadecimal key.

2.2 Processor Intervention

Depressing the following single keys causes the signals ESNC0 and ESNO0 to be complimentarily pulsed (ESNC0 is a positive going pulse):

ADD
RD
WRT
SGL
RUN

Depressing one of the following sequences of two keys causes a similar action:

FLT n (n is any hexadecimal digit)
REG n
FN n

3. FUNCTIONAL DIAGRAM ANALYSIS AND CIRCUIT DESCRIPTION

Refer to Figure 2. Hexadecimal Display Panel Block Diagram and Functional Schematic 09-065D08.

3.1 OFF-ON-LOCK Switch

This switch (2K1) controls power to the Processor by completing the circuit between CONT2 and CONT1 in the ON and LOCK positions. The switch is factory wired to provide one set of closures. This switch also provides a hard ground to the Processor as POFF0 in the OFF position which may be used as an early power down indication. When the switch is in the ON position, LP5 (2L1) is provided to the keyboard to enable the sensing of these switch closures.

3.2 Keyboard

The keyboard (Sheet 2) has a 5 x 5 switch array which is used to enter information to the Hexadecimal Display Panel logic plus an Initialize (INI) key used to transmit this condition to the Processor (2G1). The keyboard is a self-contained unit and connects to the 35-520 logic board by 27 stakes, 00-1 through 26-1. These normally open switches are encoded by diode logic (Sheet 2) to form HEX01:31 (2B8) and FUN00:30 (2C8), plus a few additional control signals mentioned later in this description. The switches are designed to be high active when a switch is depressed by biasing all receiving gates low with a 220 ohm input resistor. A switch being depressed causes an input gate to go high by supplying LP5 through a current limiting resistor from the common input, Pin 0, if the OFF-ON-LOCK switch is in the ON position. There is no keyboard rollover protection and if more than one key is simultaneously depressed, the result is unspecified.

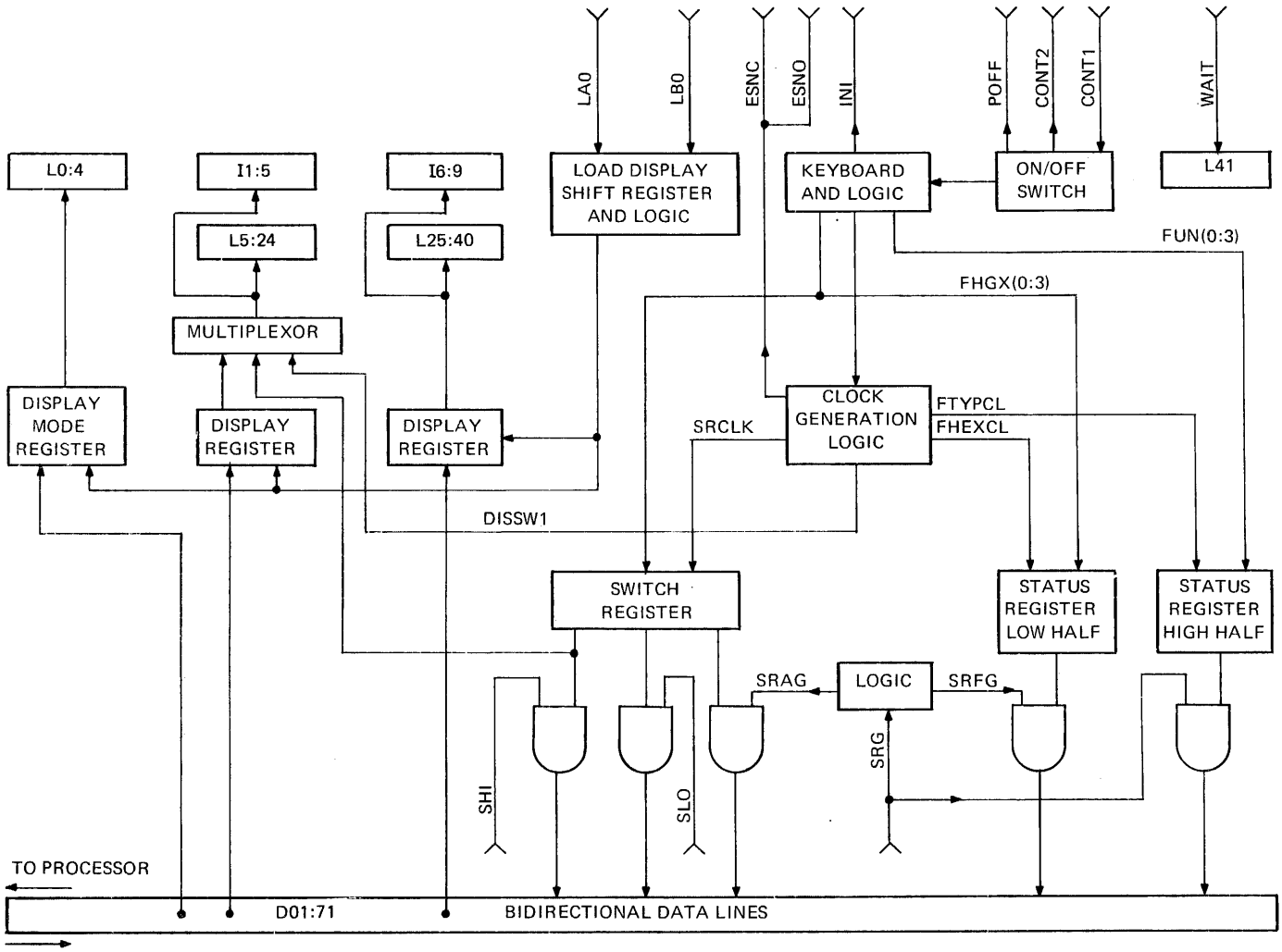


Figure 2. Hexadecimal Display Panel Block Diagram

3.3 Matrix Encoding

The diode matrix is encoded to drive signals HEX01:31 to the hexadecimal equivalent of the respective key 0:F (HEX31 is the LSB) when it is depressed. Depressing any function key other than DTA causes FUN00:30 to yield the codes specified by Table 1.

TABLE 1. FUNCTION KEY ENCODING (FUN00:30)

Key Depressed	FUN00	FUN10	FUN20	FUN30
SGL	0	1	1	1
RUN	1	1	1	1
WRT	1	1	0	1
RD	1	0	1	1
ADD	1	0	0	1
REG	0	1	1	0
FLT	0	1	0	0
FN	0	1	1	1

3.4 Clocking

Depressing any keyboard key other than DTA or INI generates one of three types of clocks used by the Hexadecimal Display Panel logic. This is accomplished by a positive transition of signal KEY1 (2F8) whenever one of these keys is depressed. The one shot triggered by this transition (2G8) is used to allow a one to two millisecond interval for switch bounce to subside before triggering the second one shot STRB1 (2K8) which is used to generate one of the three clocks. Since contact bounce is likely to retrigger these one shots when a key is released, the occurrence of signal KEY1 (any key depressed), HKEY1 (2F9 a hexadecimal key depressed), or FKEY1 (2H7 a function key depressed) being true in coincedence with the one shot is used to derive the clocks.

3.5 Switch Register Clocks

The Switch Register is enabled for clocking by depressing the DTA key. This is accomplished by direct clearing the Switch Register Enable flip-flop (SRENB) (2L6) when DTA is depressed and ANDing the zero output of the flip-flop plus HKEY1 and STRB1 to drive the Switch Register Clock (SRCLK0) (2M7). This clock is disabled by setting SRENB with the occurrence of FKEY1 when any function key is depressed.

3.6 Status Register Clocks

Two different clocks are used to load the status register. FTYPCL0 (2M8) is generated whenever any function key other than DTA is depressed and is used to load FUN00:30 into one half of the status register. The second clock FHEXCL0 (2N8) is generated whenever a hexadecimal key is depressed if the previously depressed key was FN, REG, or FLT. In this case, the hexadecimal input would be the register number or function number desired and FHEXCL0 is used to clock HEX01:31 into the second half of the status register.

3.7 Processor Intervention

The logic of the display signals the Processor that a response is necessary to a console function by signal ESNC0 (2R7) and its compliment ESNO0 (2R7). These signals are complimentarily pulsed whenever a function key other than DTA, FN, REG, or FLT is depressed, or whenever a hexadecimal key is depressed following FN, REG, or FLT (the occurrence of FHEXCL0).

3.8 Switch Register Loading

The Switch Register (4B1, 4D1, 4G1, 4J1, and 4M1) is loaded with a hexadecimal character with the occurrence of each SRCLK0 as mentioned previously. Data is entered into the least significant character (4B1) from the switches (HEX01:31) and left shifted through the register with each clock. The register is cleared whenever the DTA key is depressed.

3.9 Status Register

The status register is loaded in two parts as described previously. One half is loaded from FUN00:30 when a Function (FN) key is depressed by the occurrence of FTYPCL0. The least significant bit of this register is re-circulated on SGL or RUN and the second LSB is re-circulated on SSL to conform to the status codes indicated in Table 2. The second half of the register is loaded from HEX01:31 with the occurrence of FHEXCL0. These registers are initialized by SCLR0 from the Processor.

TABLE 2. STATUS CODES

KEY	DL1	DL2	DL3	DL4	DL5	DL6	DL7	DL0
SGL	1	U	X	X	X	X	X	X
INITIALIZE	U	U	U	U	U	U	O	U
RUN	0	0	0	X	X	X	X	X
WRT	0	0	1	U	U	U	U	U
RD	0	1	0	U	U	U	U	U
ADR	0	1	1	U	A_1	A_2	A_3	A_4
REG n	1	0	0	1	n_1	n_2	n_3	n_4
FLT n	1	0	1	1	n_1	n_2	n_3	n_4
FN n	1	0	0	0	n_1	n_2	n_3	n_4

A = Most significant hexadecimal digit of Switch Register
U = Unspecified
X = Unchanged
n = Hexadecimal digit associated with function (see Section 6)

The display status is presented to the Processor on the data lines (DL01:71) for the duration of time that control signal SRG0 is at a logical zero level. The data presented for status is in accordance with Table 2.

3.10 Display Register Loading

The Hexadecimal Display Panel registers and displays five bytes of data transmitted from the Processor. Two control signals are transmitted from the Processor to direct the loading of these registers. LA0 (2K5) is a low active pulse which signifies that data is available on bi-directional Data Lines D01:71 and it is to be loaded into the least significant byte of the display register. LA0 is used to initialize a four bit shift register (2M4) to 1000_2 which is used to load subsequent bytes, and generate a load pulse LA1 which is used to load the data into the LSB of the display register (2B6 and 3E6). Four subsequent LB0 pulses sent from the Processor gates data from D01:71 into successive bytes of the display register (3G6 and 3J6, 4C5 and 4E5, 4G5 and 4K5, 4N5 and 3N2). This is accomplished as each LB0 pulse is inverted and gated as LDB1, LDC1, LDD1 and LDE1 (2N4) respectively as controlled by the sequencing shift register (2M4) which is right shifted with each LB0 pulse.

3.11 Display Indicators

The two least significant bytes of the display register are gated directly to LEDs L25:40 and the hexadecimal indicators I6:9 (Sheet 3). LEDs L5:24 and hexadecimal indicators I1:5 are used to display either the most significant bytes of the display registers or the Switch Register. These sets of registers are selected through the 2:1 multiplexors (4C6, 4E6, 4H6, 4K6 and 4N6) as determined by the state of the DISSW1 (2N6). DISSW1 is high whenever the Switch Register is enabled (SRENB1) or a hexadecimal key is depressed (HKEY1).

3.12 Processor Inputs

Data is gated to the Processor in response to control signals SHI0, SLO0 or SRG0. SLO0 gates the two least significant digits of the Switch Register onto the bi-directional Data Lines D01:71 (4C3 and 4C4). SHI0 gates the next two Switch Register digits onto the bi-directional Data Lines D01:71 (4H3 and 4K3). SRG0 causes the status register bits to be gated (3D4) as per Table 2. Note that either the most significant Switch Register character is gated (4N3) if DL11 is low or the hexadecimal portion of the status register if DL11 is high (3H4).

4. PROCESSOR INTERFACING

4.1 Processor Connector

Signals from the display are terminated at a 26-080F06 type connector per the following list:

SIGNAL	PIN	SIGNAL	PIN	*X1-X4	PIN
D01	109	LA0	203	X1	207
D11	110	LB0	114	X2	211
D21	111	SHI0	200	X3	210
D31	112	SLO0	206	X4	209
D41	202	WAIT1	102		
D51	204	SRG0	113		
D61	205	ESNC0	103		
D71	208	ESNO0	104		
POFF0	105	INIT0	101		
CONT1	DB1-C1 & 214	SSGL1	106		
CONT2	DB1-C2	GND	100-3		
CONT3	DB-C3 & 213	GND	108		
SCLR0	107	GND	212	twisted with 114	
		GND	201	twisted with 203	

*X1-X4 A1-8 leads to front terminal strip of chassis.

4.2 Timing

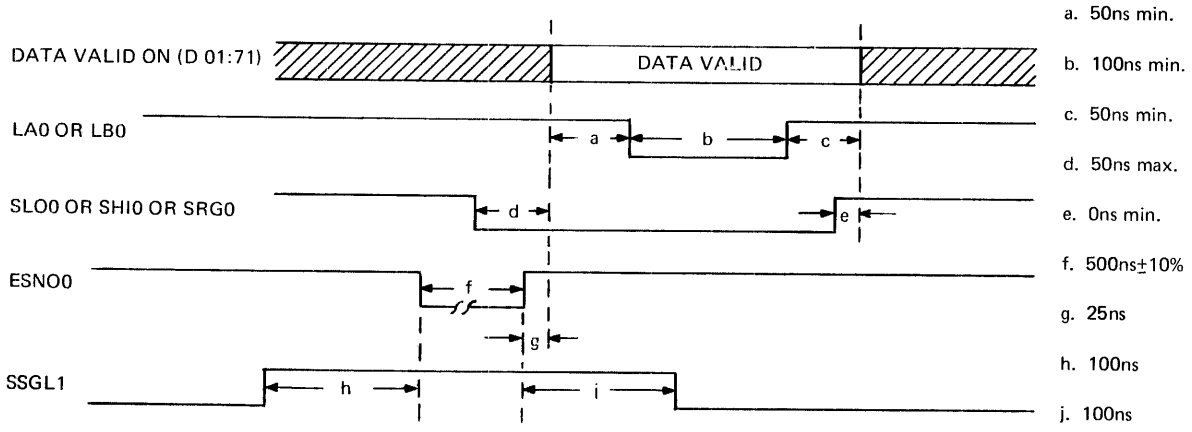


Figure 3. Hexadecimal Display Panel Timing

5. INSTALLATION PROCEDURE

The Hexadecimal Display Panel is connected to the Processor via a 17-305 cable. The 26-080F06 30-pin connector of the Hexadecimal Display Panel plugs into the mating connector as shown in Figures 4, 5 and 6.

CNTL1, CNTL2, P5, GND, LGND, +L jumpers go to corresponding lugs on the Processor chassis display terminal strip as shown in Figure 4.

6. POWER

The Hexadecimal Display Panel draws its power from the P5 and +L lugs on the Processor chassis display terminal strip. See Figure 4.

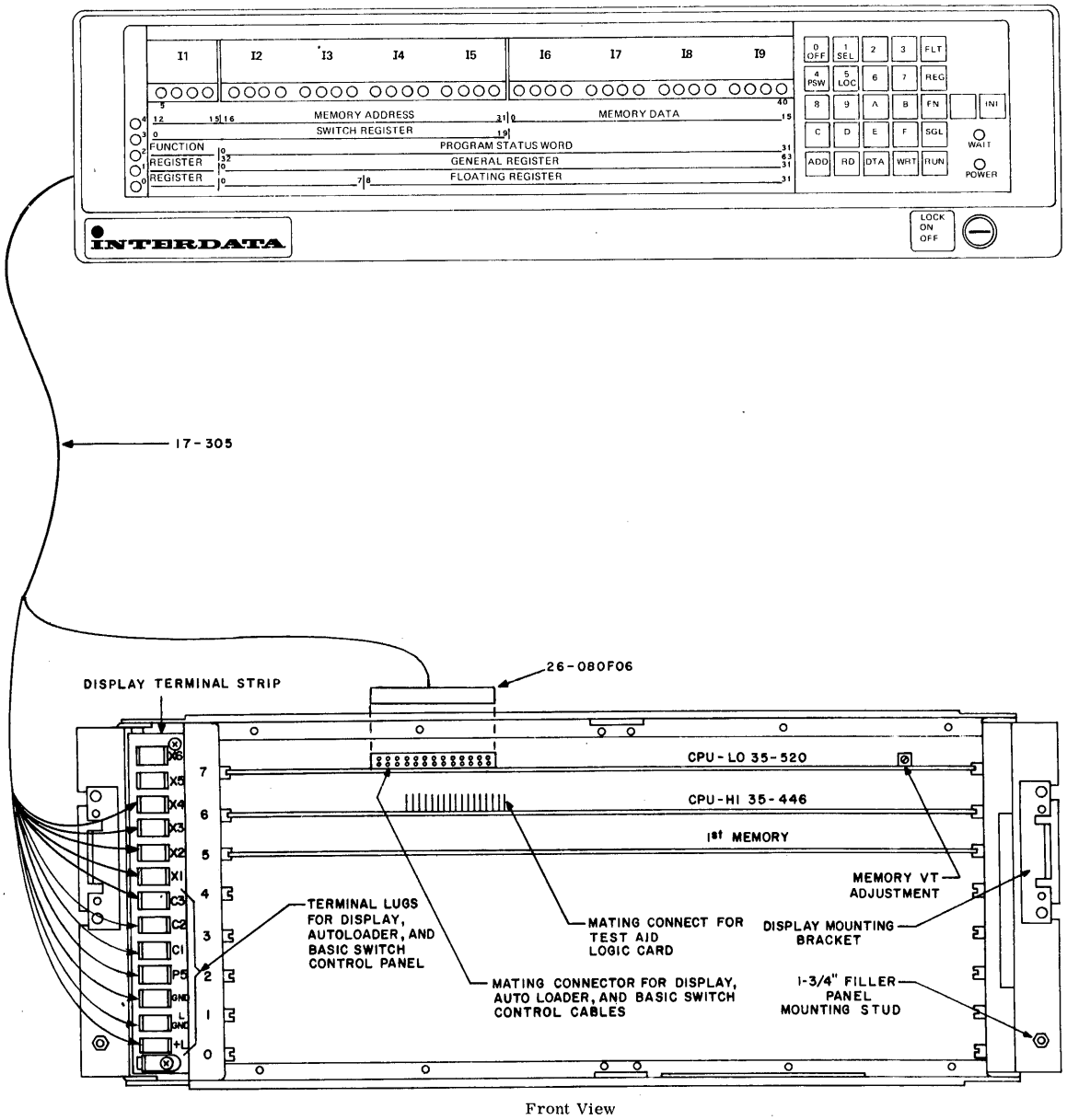
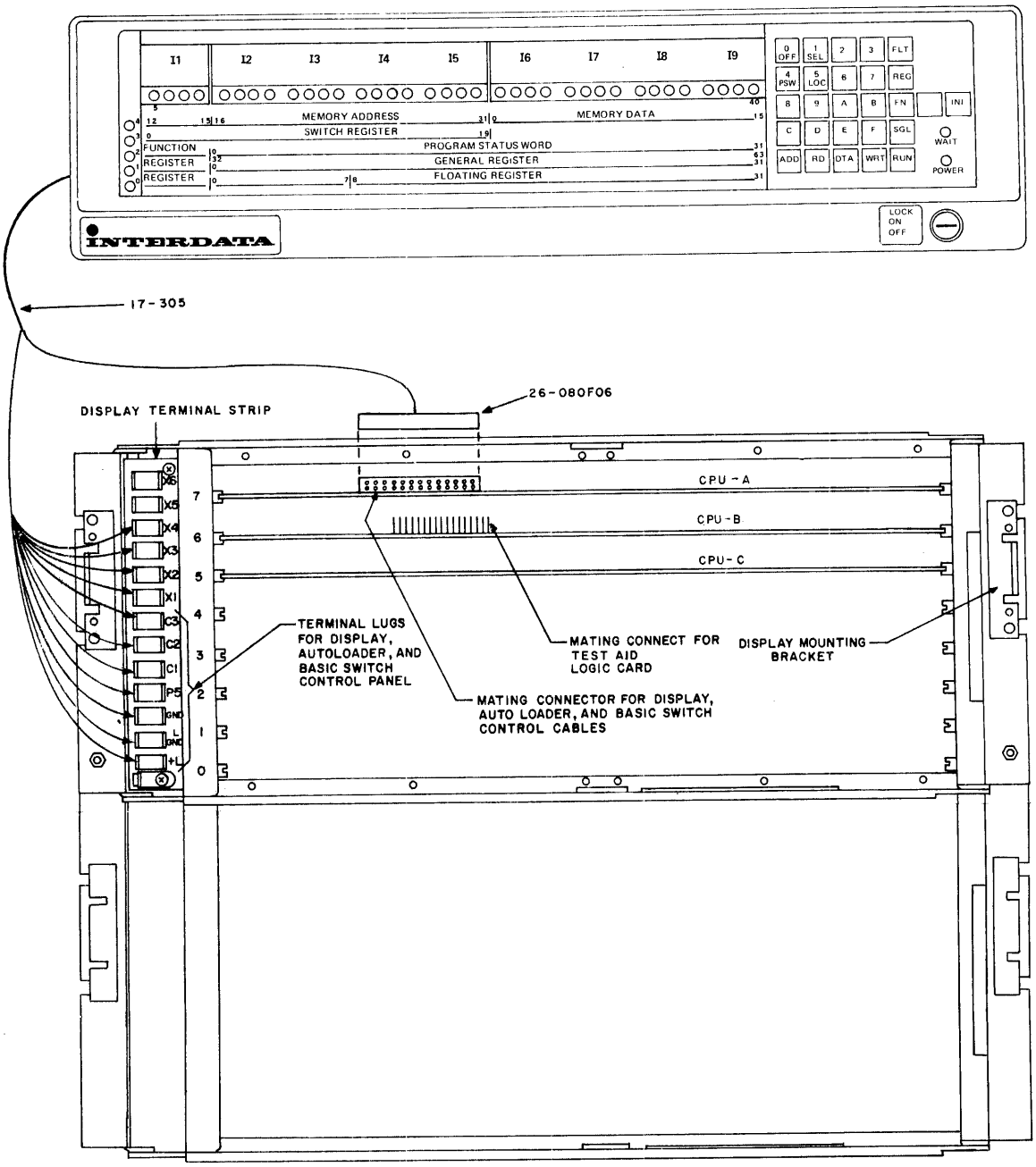
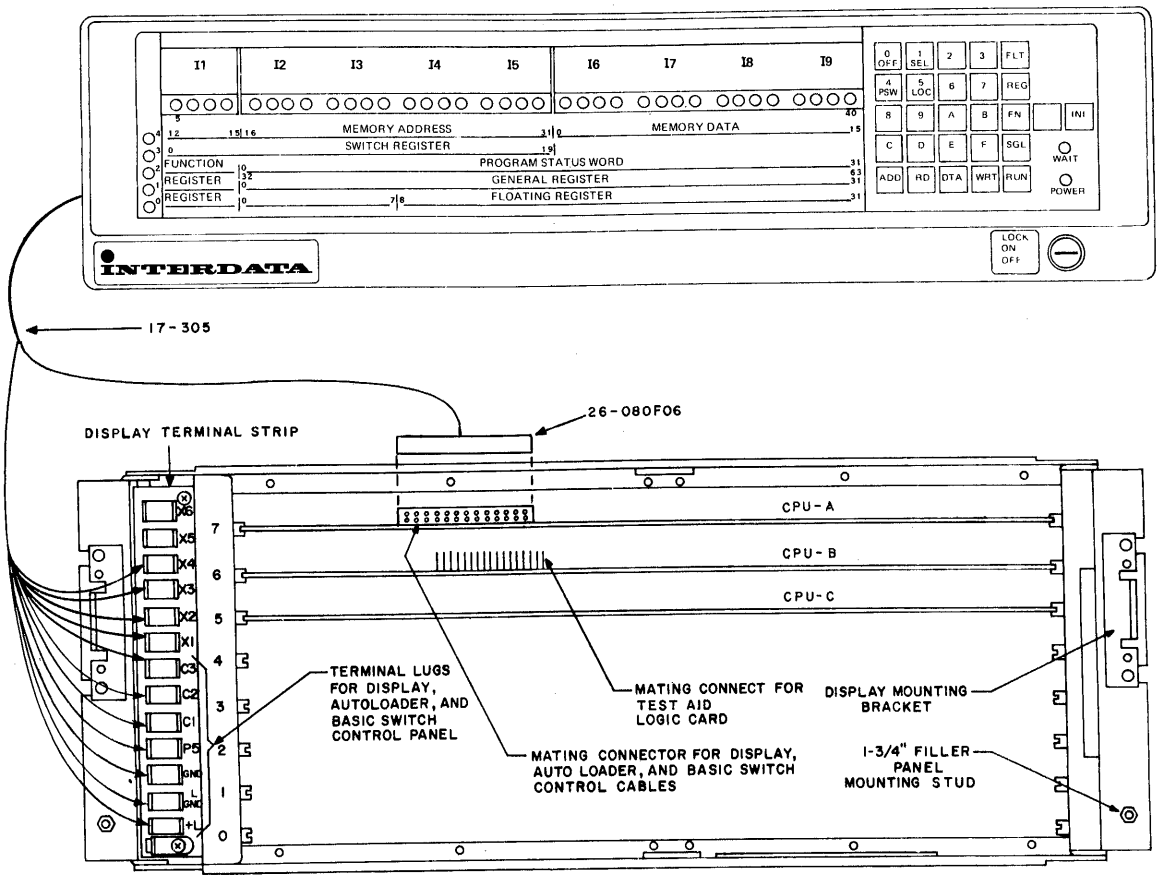


Figure 4. 7/16 Basic Display Installation



7/16 HSALU OR 7/32 TWIN CHASSIS INSTALLATION

Figure 5. Model 7/16 HSALU or 7/32 Installation



Front View

7/16 HSALU INSTALLATION

Figure 6. Model 7/16 HSALU Installation 7" Chassis

7. MNEMONICS

The following list provides a brief description of each mnemonic found in the Hexadecimal Display Panel. The source of each signal on Functional Schematic 09-065D08 is also provided.

MNEMONIC	MEANING	SCHEMATIC LOCATION
CONT1	12 VAC to turn on power supply	2L1
CONT2	12 VAC to turn off power supply	2M1
DISSW1	Controls Display Multiplexors for L5:24	2R6
ESNC0	Execute switch normally open	2R7
ESNO0	Execute switch normally closed	2R7
FTYPCL0	Function type status register clock	2N7
FHEXCL0	Hexadecimal type status register clock	2N8
FUN00:30	Encoded functional keys	Sheet 2
HEX01:31	Encoded hexadecimal keys	Sheet 2
INIT0	Initialize Processor	2H2
LA0	Low active signal from Processor which initializes the loading sequence and loads the least significant byte of the Hexadecimal Display Panel	2K5
LB0	Low active signal from Processor used to control loading of display registers by generating LDB1, LDC1, LDD1, LDE1	2L5
LDB1	Load display registers	2R3
LDC1		2R4
LDD1		2R4
LDE1	Loads display mode register and most significant hexadecimal digit of the display	2R4
POFF0	Early power OFF failure	2K1
SCLR0	System Clear, initialize status registers	3J1
SDA0	DTA key depressed	2J2
SHI0	Switch Register high half gate command	2M2
SLO0	Switch Register low half gate command	2L3
SOR0	SGL or RUN keys depressed	2K2
SRAG1	Switch Register most significant hexadecimal digit gate command	2R2
SRCLK0	Switch Register clock	2M7
SRFG1	Status Register Function high half gate command	2R2
SRG1	Status Register low half gate command	2M2
SSL1	SGL key depressed	2J6
WAIT1	Wait light control	2M6

DRAWINGS

[redacted]

MNEM.	SCHEMATIC LOCATION			TYPE
	CONTROL 'A'	ROM 'B'	ALU 'C'	
ACKO	25D9 - SOURCE			I/O
ADDO	29A6 - INPUT	14D8 - SOURCE		INT
ADRSO	25A9 - SOURCE			I/O
ARHI20		10G1 - INPUT	21H3 - SOURCE	INT
ARHI30		10G2 - INPUT	21H3 - SOURCE	INT
ARHI40		10G2 - INPUT	21H3 - SOURCE	INT
ARHI50		10G3 - INPUT	21K1 - SOURCE	INT
ATNO	31G1 - INPUT			I/O
B000		13D9 - SOURCE	20B1 - INPUT	INT
B010		13E9	20C1	
B020		13E9	20C1	
B030		13E9	20D1	
B040		13G9	20E1	
B050		13G9	20F1	
B060		13G9	20G1	
B070		13G9	20H1	
B080	26M1 - SOURCE	13J9	20J1	
B090	26M2	13J9	20K1	
B100	26M3	13J9	20L1	
B110	26M3	13J9	20M1	
B120	26M4	13J9	20M1	
B130	26M5	13M9	20N1	
B140	26M6	13M9	20R1	
B150	26M6 - SOURCE	13M9 - SOURCE	20S1 - INPUT	INT
BB001	29C5 - INPUT		29E1 - SOURCE	INT
BRCH1	33G1 - INPUT	14F8 - SOURCE		INT
CAO		14E8 - SOURCE	24N3 - INPUT	INT
CDIVO	36E6 - INPUT		36R1 - INPUT	INT
CLO70	25A9 - SOURCE			I/O
CLKIB	32M1 - SOURCE	12A9 - INPUT		INT
CLKIC	32M2 - SOURCE		17C1 - INPUT	INT
CLMDRO	27F9 - SOURCE		19B5 - INPUT	INT
CMDO	25C9 - SOURCE			I/O
CMNDO	30F1 - INPUT	14E8 - SOURCE	36L1 - INPUT	INT
CPLOC1		10A8 - INPUT	17A1 - SOURCE	
CPMAR1		10A5 - INPUT	16D1 - SOURCE	
CRYINO	33M9 - SOURCE		36N1 - INPUT	
CSRH1	36C6 - SOURCE		36J1 - INPUT	
CSV1	30G1 - INPUT	10L9 - INPUT	22C8 - SOURCE	
CSVDIVO	36F6 - SOURCE		36L5 - INPUT	INT
D000			15A1 - SOURCE	I/O
D010			15A2	
D020			15A3	
D030			15A4	
D040			15A5	
D050			15A6	
D060			15A7	
D070			15A8	
D080			15H1	
D090			15H2	
D100			15H3	
D110			15H4	
D120			15H5	
D130			15H6	
D140			15H7	
D150			15H8 - SOURCE	I/O
DACKO	25B9 - SOURCE			I/O
DAO	25C9 - SOURCE			I/O
DCO	31J1 - INPUT			DATA CHA
DCRO	31J1 - INPUT			DATA CHA
DONO	32R5 - SOURCE	11K1 - INPUT		INT
DRO	25B9 - SOURCE			I/O
DS20		8K8 - SOURCE	29J8 - SOURCE	INT
DSB1		19R8 - SOURCE	17A1 - INPUT	INT
DSTOPO	33G9 - SOURCE	12A9 - INPUT	17C1 - INPUT	INT
ENDA1	35M1 - INPUT	19G1 - SOURCE	29C3 - INPUT	INT
ENDB1	35K1 - INPUT	19J8 - SOURCE		INT
ENDC1	35L1 - INPUT	19K8 - SOURCE		INT
ENDD1	35L1 - INPUT	19K8 - SOURCE	24F3 - INPUT	INT
ENMS1	27G9 - SOURCE		19B3 - INPUT	INT
ENO	27K9 - SOURCE			DMA
ENSBO		19S8 - SOURCE	17B6 - INPUT	INT
ERO	27F9 - SOURCE			MEM
ESA1	31A3 - SOURCE	8A1 - INPUT		INT
EXBSYO	27N1 - INPUT			EXT MAC
EXDUAO	27G1 - INPUT			EXT MAC
FAIFB00	32J3 - SOURCE	8G1 - INPUT		INT
FDECIO	35B9 - SOURCE	5A1 - INPUT		INT

MNEM.	SCHEMATIC LOCATION			TYPE
	CONTROL 'A'	ROM 'B'	ALU 'C'	
FDEC20	35C9 - SOURCE	5F1 - INPUT	" .	INT
FINRO	35J9 - SOURCE	8D1 - INPUT		INT
FLRI21	30H7 - SOURCE		36L5 - INPUT	INT
FMFFI	36E1 - INPUT		36K8 - SOURCE	INT
FMPYO	36B6 - SOURCE		36J1 - INPUT	INT
FPSELO	27H6 - SOURCE	10G4 - INPUT	18C1 - INPUT	EXT MAC
FRSUBO		14D8 - INPUT	36M3 - SOURCE	INT
GABB1	29C5 - INPUT		29C4 - SOURCE	INT
GABORT1	33C9 - SOURCE	9J9 - INPUT		INT
HWO	33K1 - INPUT	9J1 - INPUT		I/O
ILOC1	35E9 - SOURCE	10A8 - INPUT	17C1 - INPUT	INT
IMAR1	35D9 - SOURCE	10A5 - INPUT	16M1 - INPUT	INT
INHO	27C9 - SOURCE			MEM
INIT1	33C9 - SOURCE	9J4 - INPUT		INT
IR081	29N7 - SOURCE	12C1 - INPUT		
IR091	29N6 - SOURCE	12D1		
IR101	29N6 - SOURCE	12F1		
IR111	29N5 - SOURCE	12G1		
IR121	30B6 - INPUT	12C1		
IR131	30C6 - INPUT	12D1		
IR141	30D6 - INPUT	12F1		
IR151	30E6 - INPUT	12G1 - INPUT		INT
LARHI20		10E3 - SOURCE	2H1 - INPUT	INT
LARHI30		10E3 - SOURCE	2J1 - INPUT	
LARHI40		10E3 - SOURCE	2J1 - INPUT	
LARHI50		10E3 - SOURCE	2J1 - INPUT	
LMARO		19K8 - SOURCE	16B1 - INPUT	
LMASO	27D9 - SOURCE	10E5 - INPUT	16A9 - INPUT	
LOADO	30R1 - INPUT	19B8 - SOURCE	16R4 - INPUT	
LPSWLI	31F4 - SOURCE	29J4 - INPUT		
LRARO	33E9 - SOURCE	8F1 - INPUT		INT
LRO	27E9 - SOURCE			MEM
M1		19J8 - SOURCE	22M7 - INPUT	INT
MA000			16B9 - SOURCE	MEM/MAC
MA001			16A9	
MA010			16B9	
MA011			16B9	
MA020			16C9	
MA021			16C9	
MA030			16D9	
MA040			16E9	
MA050			16E9	
MA060			16E9	
MA070			16F9	MEM/MAC
MA080			16H9	MEM
MA090			16H9	
MA100			16J9	
MA110			16J9	
MA120			16L9	
MA130			16L9	
MA140			16L9 - SOURCE	
MD000		8B1 - INPUT	18D7 - SOURCE	
MD010		8B1	18E7	
MD020		8C1	18G7	
MD030		8C1	18H7	
MD040		8E1	18K7	
MD050		8E1	18L7	
MD060		8F1	18N7	
MD070		8F1 - INPUT	18R7	
MD080	29H6 - INPUT		19D7	
MD090	29H6 - INPUT		19E7	
MD100	29H6 - INPUT		19G7	
MD110	29H5 - INPUT		19H7	
MD120		29G3 - INPUT	19K7	
MD130		29G3 - INPUT	19L7	
MD140		29G4 - INPUT	19N7	
MD150		29G4 - INPUT	19R7 - SOURCE	MEM
MDR000		8H1 - INPUT	18C7 - SOURCE	INT
MDR011		8H1 - INPUT	18E7 - SOURCE	INT
MSTOPO	33H9 - SOURCE	8G1 - INPUT		INT
NORMO	31K3 - INPUT		23F2 - SOURCE	INT
P00	33L1 - INPUT	8G8 - SOURCE		INT
P20		8N8 - SOURCE		INT
PERRO	28B6 - INPUT			EXT PAR
PFDTO	32D9 - INPUT			EXT PPF
POWDNO	32J7 - SOURCE			EXT PPF
PSW11	31C4 - SOURCE	5A2 - INPUT		INT
PSW21	31D4 - SOURCE	11G1 - INPUT		INT

MNEM.	SCHEMATIC LOCATION			TYPE
	CONTROL 'A'	ROM 'B'	ALU 'C'	
RD000	35A1 - INPUT	7E1 - SOURCE		INT
RD030	3158	7E4		
RD041	35G1	7E5		
RD051	35C1	7E6		
RD061	35C1	7E7		
RD071	35B1	7E7		
RD080	29A8	7K1		
RD091	29B8	7K2	36G1 - INPUT	
RD101	29A9	7K3		
RD110	25N1	7K4	23G1 - INPUT	
RD120	29C9	7K5	23G1 - INPUT	
RD131	30R1	7K6	24M3 - INPUT	
RD141	30R1 - INPUT	7K7	24M3 - INPUT	
RD151		7K7	24J3 - INPUT	
RD161	34E1 - INPUT	7S1		
RD171	25B1	7S2		
RD181	25B1	7S3		
RD191	25S1	7S4		
RD201	25R1	7S5		
RD211	2551	7S6		
RD221	39M1	7S7	24E3 - INPUT	
RD231	35L1 - INPUT	7S7 - SOURCE	24E3 - INPUT	INT
READ1		11K1 - SOURCE	20A1 - INPUT	INT
REQO	27K1 - INPUT			EXT DMA
RR1	35E1 - INPUT	9N3 - SOURCE		INT
RSTOPO	33F9 - SOURCE	7D8 - INPUT		INT
S01		14G8 - SOURCE	22R4 - INPUT	INT
S11		14H8 - SOURCE	22M7 - INPUT	INT
S21		14H8 - SOURCE	22M7 - INPUT	INT
S31		14J8 - SOURCE	22M7 - INPUT	INT
S001	31A4 - INPUT	13E2 - INPUT	22C9 - SOURCE	INT
S011	31B4 - INPUT	13E2	22D9	
S021	31B4 - INPUT	13E2	22E9	
S031		13E2	22F9	
S041	31D4 - INPUT	13G2	22J9	
S051	31D4	13H2	22K9	
S061	31D4	13H2	22L9	
S071	31D4	13H2	22M9	
S081	26K1	13K2	21C9	
S091	26K1	13K2	21D9	
S101	26K2	13K2	21E9	
S111	26K3	13K2	21F9	
S121	26K4	13R4	21J9	
S131	26K5	13R6	21K9	
S141	26K5	13R4	21L9	
S151	26K6 - INPUT	13R6 - INPUT	21M9 - SOURCE	INT
SCLRO	32A5 - SOURCE			I/O
SCLROA	32E5 - SOURCE	7A8 - INPUT		INT
SELA1		10G2 - INPUT	24J8 - SOURCE	
SEQO1	30M1 - INPUT		22N8 - SOURCE	
SHCLKO	36A9 - SOURCE		23R1 - INPUT	
SHCRY1	30H1 - INPUT		29E4 - SOURCE	
SHORT1	31K9 - INPUT	8R8 - SOURCE		
SMPYO	36A1 - SOURCE		36N1 - INPUT	
SRHSO1	34D6 - SOURCE		23B4 - INPUT	
SRHS11	36E6 - SOURCE		23B4 - INPUT	
SRLSO1	36C6 - SOURCE		23J4 - INPUT	
SRLS11	36D6 - SOURCE		23H4 - INPUT	
SRO	25D9 - SOURCE			I/O
SUB1	30H1 - INPUT	14H8 - SOURCE		INT
SYNO	23L1 - INPUT			I/O
ULOC1	35R9 - SOURCE	10E7 - INPUT		INT
UMAR1	35N9 - SOURCE	10G5 - INPUT		INT
UMDR1	35N9 - SOURCE	9G9 - INPUT		INT
USRLOC1		14N8 - SOURCE	23G2 - INPUT	INT
WO	27B9 - SOURCE			MEM
WRTO	27H9 - SOURCE			MEM
XMA120		10L5 - SOURCE		MEM/MAC
XMA130		10L5 - SOURCE		
XMA140		10L4 - SOURCE		
XMA141		10R5 - SOURCE		
XMA150		10L4 - SOURCE		MEM/MAC
P15	32J8 - 15VDC			

RELEASED FOR PRODUCTION
ENG. DJC DATE 9-20-74

	CHANGED SHTS 25 THRU 36 TO REFLECT CHAN 35-322 M01, IN SUP. INFO TBL BLK BELOW, CHAN WAS 35-522 R09.			1
X	JEH	2290	- 10-1-74	R01
	REVISED SHTS 2 & 28.			
E	JEH	2365	- 12-17-74	R02
	REVISED SHTS 25, 26, 27 & 28			
E	JEH	2376	- 1-28-75	R03
	REVISED SHTS, 5, 8, 10, 11, 12, 13, 14, 16, 23, 26, 31 & 36			2
E	JEH	2392	- 2-13-75	R04
	REVISED SHTS 9, 25, 30, 32			
E	JEH	2434	- 3-17-75	R05
	REVISE SHT. 16.			
E	JEH	2491	- 3-20-75	R06
	REVISED SHT 27.			
E	JEH	2496	- 3-21-75	R07
	REVISED SHT 21.			
E	JEH	2553	- 4-26-75	R08
	REVISE SHT 27. SHT 27 REV LEVEL WAS R03. AREA 5-7 35-522 M01 R09 REV LEVEL WAS R03.			3
E	JEH	2569	- 5-19-75	R09
	REVISED SHTS. 25, 27 & 28 AREA 57, 35-522 M01 R05 WAS R04.			
E	MM	2552	- 6-24-75	R10
	REVISED SHTS 25 & 26. AREA 57, 35-522 M01 R06 WAS R05.			4
E	MM	2573	- 7-14-75	R11
	REVISED SHT. 32 35-522 M01 WAS R06			
E	JEH	2619	- 9-23-75	R12
	REVISED SHT 25 & 28			
E	MM	2733	- 2-21-76	R13
	REVISED SHT 32			5
E	MM	2650	- 2-21-76	R14
	REVISED SHTS 2 & 27, & SUPPLEMENTARY INFO. TABLE & SHT 32			
E	MM	2697	- 2-23-76	R15
	REVISED SHT 32 & SUP. INFO TABLE, 35-522 M01 WAS R10			
E	MM	2838	- 5-5-76	R16
	CONTINUED BELOW			6

SUPPLEMENTARY INFORMATION

PRINTED CIRCUIT BOARDS AGREEING WITH THIS SCHEMATIC MUST BE AT LEAST THE FOLLOWING REVISION LEVEL

CPU-A	35-522 M01 R13
CPU-B 7/16 HSALU	35-544 R10
CPU-B 7/32 w/F.P.	35-523 F01 R15
CPU-B 7/32 w/o F.P.	35-523 F02 R15
CPU-C	35-524 M01 R06
16" TERMINATOR	35-433 R02
PRI. POWER FAIL	35-448

THE REVISION LEVEL OF THIS SHEET IS CONSIDERED TO BE THE REVISION

SHEET INDEX	REV. LEVEL	SHEET NO.
	24	1
	2	2
	1	3
	1	4
	1	5
	1	6
	1	7
	2	8
	2	9
	1	10
	1	11
	2	12
	1	13
	1	14
	1	15
	1	16
	1	17
	1	18
	1	19
	1	20
	1	21
	2	22
	2	23
	2	24
	2	25
	2	26
	7	27
	9	28
	7	29
	6	30
	1	31
	2	32
	2	33
	6	34
	1	35
	2	36

NOTES:

REVISED SHT 12. REV. LEVEL OF 35-544, 35-523 F01 & 35-523 F02 WAS R05, R04 & R04. GP 11 3478 11-12-76 R24

CPU-C 35-524 M01 IN THE SUPPL INFO TABLE WAS R05. GS 11 3319 9-26-77 R22

REVISED SHT 25. CPU-A BRD. WAS R2. GS 11 3314 12-1-77 R23

WAS R00. GS 11 3151 9-19-77 R20

REVISED SHTS 1, 18 & 19. CPU-C BOARD WAS R04. GS 11 3355 9-16-77 R21

REVISED SHT'S 27 & 32 35-522 M01 WAS R11. PM 11 30.31 3-26-77 R13

REVISED SHT 1, 18 & 19. CPU-C BOARD, 35-524 M01 REV LEVEL WAS R05. GS 11 3002 12-15-76 R18

REVISED SHT 34. US 7 2941 9-9-76 R17

REVISED SHEETS 18 & 28. DELETED 35-533 PROM SUPPL. INFO. TABLE. US 7 2697 2-23-76 R15

CHRIS JENSEN 6-27-74
R. CERO 9-20-74
D. FRANKENBERGER 9-20-74
H. ROSS Q.C. 9-20-74
D. FRANKENBERGER MGR 9-20-74

FUNCTIONAL SCHEMATIC
MODEL 7/16 HSALU & 7/32 PROCESSOR
03097
01-079M01 REV 008
36

CONN 2 PIN INDEX

MNEM	CPU-A	CONN. NO.	TYPE
RDSETO	34H9 - SOURCE	101	EXT MAC
CWRO	27G1 - INPUT	111	EXT PRT
DMAC0	35C9 - SOURCE	100	EXT MAC
HALTO	27J1 - INPUT	113	EXT MAC
IRI	35H1 - SOURCE	103	EXT MAC
MACO	31F1 - INPUT	104	EXT MAC
MEMI	35E9 - SOURCE	110	EXT MAC
MWI	35A9 - SOURCE	108	EXT PRT
PBY0	28D7 - SOURCE	109	EXT MAC
PRTECTO	35A9 - SOURCE	107	EXT PRT
XMEMO	27N1 - INPUT	112	EXT MAC

CONN 3 PIN INDEX

MNEM	CPU-A	CONN.NO.	TYPE
ESNOO	26A2 - INPUT	104	EXT DIS
ESNCO	26A1 - INPUT	103	
INITO	32B7 - INPUT	101	
LAO	26J6 - SOURCE	203	
LBO	26J6 - SOURCE	114	
POFFO	32B7 - INPUT	105	
SCLROD	26J9 - SOURCE	107	
SD001	26R1 - SOURCE	109	
SD011	26R2	110	
SD021	26R2	111	
SD031	26R3	112	
SD041	26R4	202	
SD051	26R5	204	
SD061	26R5	205	
SD071	26R6 - SOURCE	208	
SHIO	26J5 - SOURCE	200	
SLOO	26J5 - SOURCE	206	
SRGO	26F3 - SOURCE	113	
SSGLI	26A2 - INPUT	106	
WAITI	26J3 - SOURCE	102	EXT DIS
P5			
GND			
GND			
GND			
GND			

CONN. 3 PIN INDEX

MNEM	CPU-B	CONN.NO.	TYPE
CLKO	11J1 - SOURCE	104	EXT TEST
RAR050	5S7 - SOURCE	119	
RAR060	5N6	108	
RAR070	5N6	109	
RAR080	5N6	110	
RAR090	5N5	107	
RAR100	5M3	113	
RAR110	5M3	118	
RAR120	5L1	117	
RAR130	5L1	116	
RAR140	5M1	114	
RAR150	5M1 - SOURCE	115	
SCLROA	7A8 - INPUT	134	EXT TEST
P5	11N1		
P5	11N1		
GND	11N1		
GND	11M1		
GND	11M1		

CONN 3 PIN INDEX

MNEM	CONN.NO.	CPU-A	TYPE
MA001A	221	16A6 - SOURCE	EXT PRT
MA011A	223	16A6	
MA021A	222	16A6	
MA031A	122	16A6	
MA041A	123	16H5	
MA051A	124	16H9	
MA061A	224	16H9 - SOURCE	EXT PRT

	PRIMARY POWER FAIL	
C1	32K9 12VAC	EXT
C3	32K8 12VAC	
PFDTO	32K9 SOURCE	
POWDNO	32K7 SOURCE	
PIS	32K7 15VDC	EXT

	PARITY OPTION BOARD	
LRO	28E4 - INPUT	MEM
MD000	28A1 - INPUT	MEM
MD010	28A1	
MD020	28A1	
MD030	28A1	
MD040	28A1	
MD050	28A1	
MD060	28A2	
MD070	28A2	
MD080	28A3	
MD090	28A3	
MD100	28A3	
MD110	28A3	
MD120	28A3	
MD130	28A3	
MD140	28A4	
MD150	28A4	
MD160	28A4 - INPUT	MEM
PERRO	28D2 - SOURCE	EXT PAR
WRTO	28E3 - INPUT	MEM
WRTOA	28E2 - INPUT	DMA
P5	28R2	
GND	28R3	

TEST POINT INDEX

DESIG.	MNEM	LOC.
	CONTROL A	
TP1	SCATNO	26F1
TP2	FTITO	33J1
TP3	FPSELI	27J6
TP4	ARSTI	31K6
TP5	FLR131	30K7
TP6	FLR151	30R7
TP7	FLR141	30L7
TP8	FLR121	30H7
TP9	STPSYNO	25L1
TP10	FSTO	25H7
TP11	XRD2	27N8
TP12	GND	27M1
TP13	GND	31K6
TP14	GND	26F1
TP15	GND	25H8
	ROM B	
TP1	FTITO	11R1
TP2	XRSTO	TD9
TP3	RD020	7E3
TP4	RD010	7E2
TP5	X511	10S8
TP6	XS31	10S7
TP7	XS21	10S7
TP8	XS01	10S8

TRUTH TABLES

19-038

SELECT INPUTS B	SELECT INPUTS A	DATA INPUTS C0	C1	C2	C3	STROBE G	OUTPUT Y
X	X	X	X	X	X	H	L
L	L	L	X	X	X	L	L
L	L	H	X	X	X	L	H
L	H	X	L	X	X	L	L
L	H	X	H	X	X	L	H
H	L	X	X	L	X	L	L
H	L	X	X	H	X	L	H
H	H	X	X	X	L	L	L
H	H	X	X	X	H	L	H

SELECT INPUTS A AND B ARE COMMON TO BOTH SECTIONS. H = HIGH LEVEL
L = LOW LEVEL X = IRRELEVANT

19-073

ADDRESS INPUT B	ADDRESS INPUT A	DATA INPUTS C0	C1	C2	C3	STROBE G	OUTPUT Y
X	X	X	X	X	X	1	HiZ
0	0	0	X	X	X	0	0
0	0	1	X	X	X	0	1
0	1	X	0	X	X	0	0
0	1	X	1	X	X	0	1
1	0	X	X	0	X	0	0
1	0	X	X	1	X	0	1
1	1	X	X	X	0	0	0
1	1	X	X	X	1	0	1

X = DON'T CARE

19-066 + 19-065

INPUTS STROBE	SELECT	A	B	OUTPUT Y 19-065	19-066
H	X	X	X	L	H
L	L	L	X	L	H
L	L	H	X	H	L
L	H	X	L	L	H
L	H	X	H	H	L

H=HIGH LEVEL, L=LOW LEVEL
X=IRRELEVANT

19-028

INPUT A1 / A3	B1 / B3	A2 / A4	B2 / B4	WHEN CO=L, WHEN C2=L Σ1 / Σ3	Σ2 / Σ4	C2 / C4	WHEN CO=H, WHEN C2=L Σ1 / Σ3	Σ2 / Σ4	C2 / C4
L	L	L	L	L	L	L	H	L	L
H	L	L	L	H	L	L	L	H	L
L	H	L	L	H	L	L	L	H	L
H	H	L	L	L	H	L	H	H	L
L	L	H	L	H	L	L	L	L	H
H	L	H	L	H	L	L	L	L	H
L	H	H	L	H	L	L	L	L	H
H	H	H	L	L	H	L	H	L	H
L	L	L	H	H	L	L	H	L	H
H	L	L	H	H	L	L	L	L	H
L	H	L	H	H	L	L	L	L	H
H	H	L	H	L	H	L	H	L	H
L	L	H	H	H	L	L	H	L	H
H	L	H	H	H	L	L	L	L	H
L	H	H	H	H	L	L	L	L	H
H	H	H	H	L	H	H	H	H	H

H=HIGH LEVEL, L=LOW LEVEL
INPUT CONDITIONS AT A3, A2, B2, AND CO ARE USED TO DETERMINE OUTPUTS Σ1 AND Σ2 AND THE VALUE OF THE INTERNAL CARRY C2. THE VALUES AT C2, A3, B3, A4, AND B4 ARE THEN USED TO DETERMINE OUTPUTS Σ3, Σ4, AND C4.

J. HELVATY 7-10-74

FUNCTIONAL SCHEMATIC
MODEL 7/16 HSALU & 7/32 PROCESSOR
03097
01-079 M01 D08 1A 36

BACK PANEL MAP

TERM NO.	CONTROL "A" 07 ROW 1	CONTROL "A" 07 ROW 2	ROM "B" 06 ROW 1	ROM "B" 06 ROW 2	ALU "C" 05 ROW 1	ALU "C" 05 ROW 2	MAC 04 ROW 1	MAC 04 ROW 2	TERM NO.	MEM / I/O 03 ROW 1	MEM / I/O 03 ROW 2	02 ROW 1	02 ROW 2	01 ROW 1	01 ROW 2	00 ROW 1	00 ROW 2	TERM NO.
TITLE BD. LOC.									TITLE BD. LOC.									TITLE BD. LOC.
CONN.									CONN.									CONN.
41	P5	P5	P5	P5	P5	P5	P5	GND	41	P5	GND					P5	GND	41
40	GND	GND	GND	MD150	MD150	SHCLKO	GND	GND	40	GND	GND					GND	GND	40
39	REQO	FDEC20	FDEC20	MD140	MD140	CRYINO			39	P15	P15					P15	P15	39
38	ENO	FDEC10	FDEC10	MD130	MD130	FMPYO	MD150	MD160	38	N15	N15					N15	N15	38
37	EXDUAO	PSW231	PSW231	MD120	MD120	CDIVO	MD130	MD140	37	MD150	MD160					MD150	MD160	37
36	UMDRI	WRTO	FRSUBO	MD110	CAO	MD110	MD110	MD120	36	MD130	MD140					MD130	MD140	36
35	PSW111	PERRO	PSW111	M1	M1	MD100	MD090	MD100	35	MD110	MD120					MD110	MD120	35
34	MPARO	SCLROA	SCLROA	SO1	SO1	MD090	MD070	MD080	34	MD090	MD100					MD090	MD100	34
33	LPSWL1	POO	LPSWL1	MA130	MA130	MD080	MD050	MD060	33	MD070	MD080					MD070	MD080	33
32	DONO	MSTOPO	MSTOPO	MA110	MA120	MD070	MD030	MD040	32	MD050	MD060					MD050	MD060	32
31	SMPYO	UMARI	SELAI	MA090	MA100	MD060	MD010	MD020	31	MD030	MD040					MD030	MD040	31
30		SRHSO1		MA070	MA080	MD050		MD000	30	MD010	MD020					MD010	MD020	30
29	ILOC1	IMARI	ILOC1	IMAR1	ILOC1	SRHS11	MA050	MA060	29	EXVT	MD000					EXVT	MD000	29
28	POWONO	SRHS11	XMA141	S11	S11	MA030	MAR030	MXR141	28	TEMPA	VT					TEMPA	VT	28
27	DCO	DCRO	XMA120	S21	S21	SELAI	SCLRO		27	WRTO	TEMPB					WRTO	TEMPB	27
26	SCLRO	HWO	HWO	S31	S31	MA020	MAR020	MXR130	26	SCLRO	HWO					SCLRO	HWO	26
25	GND	GND	GND	GND	GND	MA010	MAR010	MXR140	25	PERRO	GLRO							25
24	LMA50	ENDD1	LMA50	ENDD1	ENDD1	MA000	MAR000	MXR150	24	MPARO	ENMSI							24
23	SYNO	ATNO	XMA130	DS20	DS20	SELAI			23	SYNO	ATNO					SYNO	ATNO	23
22	ACKO	B150	XMA140	B150	B150	MA010	DGND	DMA160	22	RACKO	TACKO					RACKO	TACKO	22
21	CLO70	DAO	XMA150	DS81	DS81	MA000	DMA150	DMA140	21	CLO70	DAO					CLO70	DAO	21
20	DRO	CMDO	LARH140	LARH150	LARH140	LARH150	DMA130	DMA120	20	DRO	CMDO					DRO	CMDO	20
19	SRO	ADRSO	LARH120	LARH130	LARH120	LARH130	DGND	DMA110	19	SRO	ADRSO					SRO	ADRSO	19
18	BRCHI		BRCHI	P20	D140	D150	DMA110	DMA100	18	D140	D150					D140	D150	18
17	IR141	IR151	IR141	IR151	D120	D130	DMA090	DMA080	17	D120	D130					D120	D130	17
16	IR121	IR131	IR121	IR131	D100	D110	DMA070	DMA060	16	D100	D110					D100	D110	16
15	IR101	IR111	IR101	IR111	D080	D090	DMA050	DGND	15	D080	D090					D080	D090	15
14	IR081	IR091	IR081	IR091	D060	D070	DMA030	DMA040	14	D060	D070					D060	D070	14
13	SHORTI	ULOC1	SHORTI	ULOC1	D040	D050	DMA010	DMA020	13	D040	D050					D040	D050	13
12	SUBI	CMNDO	SUBI	CMNDO	D020	D030	DGND	DMA000	12	D020	D030					D020	D030	12
11	PSW271	ESAI	PSW271	ESAI	D000	D010	DMX150	DMX140	11	D000	D010					D000	D010	11
10	B130	B140	B130	B140	B130	B140	DMX130	DGND	10	WRTOA	MA040					MA030	MA040	10
09	B120	ENDC1	B120	ENDC1	B120	MA081	M3B20	M2B20	09	MS010	MA020					MA020	MA021	09
08	B100	B110	B100	B110	B100	B110	M1B20	M0B20	08	MS030	MA010					MA01	MA02	08
07	B090	ENDB1	B090	ENDB1	B090	MA011	LOADO	AN50	07	MS050	MA000					MA010	MA011	07
06	B080	RR1	B080	RR1	B080	MA001	LMARO	NJRMO	06	MS070	MA001					MA000	MA001	06
05	EXBSYO	DACKO	B060	B070	B060	B070	CSVDIVO	CPLOC1	05	MS090	XMA141					PARO	MA000	05
04	INMO	ERO	B040	B050	B040	B050	CPLOC1	LMARO	04	MS110	ERO					INHO	ERO	04
03	WO	LRO	B020	B030	B020	B030	LMARO	SOTO	03	MS130	LRO					WO	LRO	03
02	PI5	PFDTO	B000	B010	B000	B010	SOTO	EOTO	02	MS150	NI5					PI5	NI5	02
01	GND	GND	GND	GND	GND	GND	XREQO	QUEO	01	GND	GND					GND	GND	01
00	P5	P5	P5	P5	P5	P5	P5	GND	00	P5	GND					P5	GND	00
	1	2	1	2	1	2	1	2		1	2	1	2	1	2	1	2	

TERM NO.	MAC 04 ROW 1	MAC 04 ROW 2	TERM NO.	MEM / I/O 03 ROW 1	MEM / I/O 03 ROW 2	02 ROW 1	02 ROW 2	01 ROW 1	01 ROW 2	00 ROW 1	00 ROW 2	TERM NO.
41	P5	GND	41	P5	GND					P5	GND	41
40	GND	GND	40	GND	GND					GND	GND	40
39	P15	REQO	39	P15	REQO					P15	REQO	39
38	N15	ENO	38	N15	ENO					N15	ENO	38
37	ACTO	TACO	37	ACTO	TACO					ACTO	TACO	37
36			36									36
35			35									35
34			34							XRACKO		34
33	MA130	MA140	33							MA130	MA140	33
32	MA110	MA120	32							MA110	MA120	32
31	MA090	MA100	31							MA090	MA100	31
30	MA070	MA080	30							MA070	MA080	30
29	MA050	MA060	29							MA050	MA060	29
28	RDACKO	TDACKO	28							RDACKO	TDACKO	28
27	DCO	DCRO	27							DCO	DCRO	27
26	SCLRO	HWO	26							SCLRO	HWO	26
25			25									25
24			24									24
23	SYNO	ATNO	23							SYNO	ATNO	23
22	RACKO	TACKO	22							RACKO	TACKO	22
21	CLO70	DAO	21							CLO70	DAO	21
20	DRO	CMDO	20							DRO	CMDO	20
19	SRO	ADRSO	19							SRO	ADRSO	19
18	D140	D150	18							D140	D150	18
17	D120	D130	17							D120	D130	17
16	D100	D110	16							D100	D110	16
15	D080	D090	15							D080	D090	15
14	D060	D070	14							D060	D070	14
13	D040	D050	13							D040	D050	13
12	D020	D030	12							D020	D030	12
11	D000	D010	11							D000	D010	11
10	MA030	MA040	10							MA030	MA040	10
09	MA020	MA021	09							MA020	MA021	09
08	XMA130	XMA150	08							MA01	MA02	08
07	MA010	MA011	07							MA010	MA011	07
06	MA000	MA001	06							MA000	MA001	06
05	PARO	XMA141	05							PARO	MA000	05
04	INHO	ERO	04							INHO	ERO	04
03	WO	LRO	03							WO	LRO	03
02	PI5	NI5	02							PI5	NI5	02
01	GND	GND	01							GND	GND	01
00	P5	GND	00							P5	GND	00
	1	2		1	2	1	2	1	2	1	2	

NOTE:

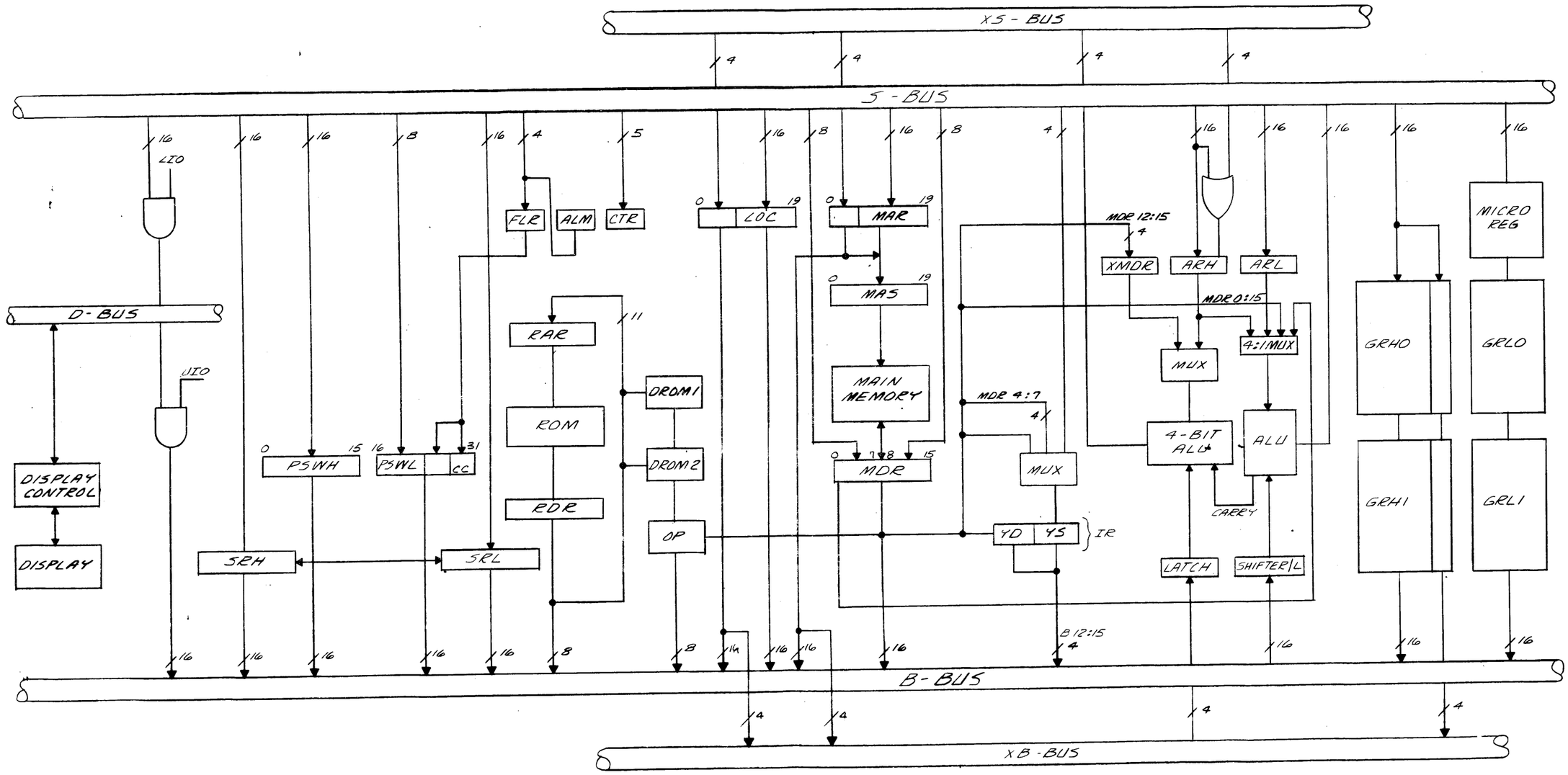
AREA B3: GLRO WAS NOT SPEC'D AT 126-0701 AREA J3 GLRO WAS SPEC'D AS LROAT 225-0301

45 | 12 | 2647 | - | 2-23-76 | P02

APLN J-3, B-6 & F-3 "MPARO" WAS SPEC'D AS "PARO" THREE PLACES.

P | 44 | 2365 | — | 12-17-74 | P01

MFG ENGR	
DIR ENG	
TITLE	FUNCTIONAL SCHEMATIC MODEL 7/16 HSALU & 7/32 PROCESSOR
TASK NO.	03047
DWG NO.	01-079N01R02D08
SHEET OF	2 - 36



BLOCK DIAGRAM

NOTES

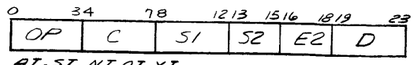
NAME	TITLE	DATE	TITLE
V. PERRI	DRAFT		FUNCTIONAL SCHEMATIC
	CHK		MODEL 7/16 HSALU & 7/32 PROCESSOR
	ENGR		TASK NO. 03047
	DIR ENG		DWG NO. 01-079 M01 D08
			SHEET 3 OF 36

INTERDATA®

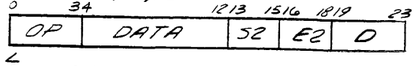
REVISIONS
AREA DB, OP EXTENSION TABLE, 18, DA. WAS 1.
A ECR 2434 / 3-17-75 / R01

MICRO INSTRUCTION FORMATS

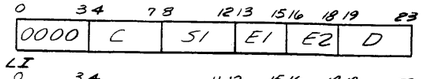
1. A, S, N, O, X, C, A



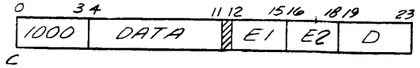
2. AI, SI, NI, OI, XI



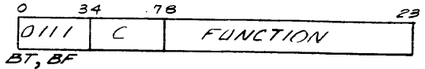
3. L



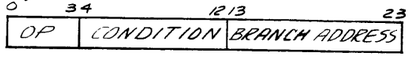
4. LI

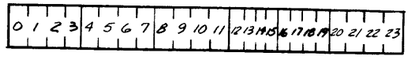


5. C



6. BT, BF





OP FIELD

0	1	2	3	
0	0	0	0	LOAD-L
0	0	0	1	AND-N
0	0	1	0	OR-O
0	0	1	1	XOR-X
0	1	0	0	ADD-A
0	1	0	1	SUB-S
0	1	1	0	CALCULATE ADR-CA
0	1	1	1	COMMAND
1	0	0	0	LOAD IMM.
1	0	0	1	AND IMM.
1	0	1	0	OR IMM.
1	0	1	1	XOR IMM.
1	1	0	0	ADD IMM.
1	1	0	1	SUB IMM.
1	1	1	0	BRANCH ON TRUE
1	1	1	1	BRANCH ON FALSE

CONTROL (C)

4	5	6	7	
0	0	0	0	NO ACTION
0	0	0	1	MRI
0	0	1	0	DI
0	0	1	1	DZ
0	1	0	0	IR
0	1	0	1	IRJH
0	1	1	0	IRJF
0	1	1	1	IRJ
1	0	0	0	MR
1	0	0	1	MR2
1	0	1	0	MRD
1	0	1	1	XR2
1	1	0	0	MWD
1	1	0	1	MRDR
1	1	1	0	MW
1	1	1	1	MW2

FIRST SOURCE (S1)

	8	9	10	11	12	
	0	0	0	0	0	MR0
	0	0	0	0	1	MR1
	0	0	0	1	0	MR2
	0	0	0	1	1	MR3
	0	0	1	0	0	MR4
	0	0	1	0	1	MR5
	0	0	1	1	0	MR6
	0	0	1	1	1	PSWL
	0	1	0	0	0	
	0	1	0	0	1	
	0	1	0	1	0	
	0	1	0	1	1	
	0	1	*1	0	0	YS1
SA	0	1	1	0	1	YD1
	0	1	1	1	0	PSWH
	0	1	1	1	1	OP
	1	0	0	0	0	NULL
	1	0	0	0	1	LOCH
	1	0	0	1	0	SRL
	1	0	0	1	1	SRH
	1	0	1	0	0	LOC
SB	1	0	1	0	1	IO
	1	0	1	1	0	MDR
	1	0	1	1	1	MAR
	1	1	0	0	0	YDH
	1	1	0	0	1	YDL
	1	1	0	1	0	YDLP1
	1	1	0	1	1	YDLM1
	1	1	1	0	0	YSH
	1	1	1	0	1	YSL
	1	1	1	1	0	YSLX
	1	1	1	1	1	YSHX

SECOND SOURCE (S2)

13	14	15	
0	0	0	NULL
0	0	1	ONE
0	1	0	ARL
0	1	1	TWO
1	0	0	MDR
1	0	1	SIGN
1	1	0	AR
1	1	1	ARH

OP EXTENSION (E1)

13	14	15	
1	0	0	SR
0	1	0	SL
1	1	0	CS
X	X	1	LEN

OP EXTENSION (E2)

16	17	18	
1	X	X	CARRY IN
X	1	X	CARRY OUT
X	X	1	FLAGS

OP EXTENSION 1 (E2) FOR I/O

16	17	18	
0	0	0	DCAK
0	0	1	ADRS
0	1	0	DA
0	1	1	OC
1	0	0	ACK
1	0	1	DR
1	1	0	STAT
1	1	1	

DESTINATION

	19	20	21	22	23	
	0	0	0	0	0	MR0
	0	0	0	0	1	MR1
	0	0	0	1	0	MR2
	0	0	0	1	1	MR3
	0	0	1	0	0	MR4
	0	0	1	0	1	MR5
	0	0	1	1	0	MR6
	0	0	1	1	1	PSWL
	0	1	0	0	0	CTR
	0	1	0	0	1	ARL
DA	0	1	0	1	0	ARH
	0	1	0	1	1	AR
	0	1	1	0	0	YSI
	0	1	1	0	1	
DB	0	1	1	1	0	PSWH
	0	1	1	1	1	PLR
	1	0	0	0	0	NULL
	1	0	0	0	1	
DC	1	0	0	1	0	SRL
	1	0	0	1	1	SRH
	1	0	1	0	0	LOC
	1	0	1	0	1	IO
DD	1	0	1	1	0	MDR
	1	0	1	1	1	MAR
	1	1	0	0	0	YDH
	1	1	0	0	1	YDL
	1	1	0	1	0	YDLP1
	1	1	0	1	1	YDLM1
	1	1	1	0	0	YSH
	1	1	1	0	1	YSL
	1	1	1	1	0	
	1	1	1	1	1	

FUNCTION (FOR COMMAND)

8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
1	0				1											MPY
1	1				1											UNMPY
		1				1										DIV
			1	0	0											SL1
			1	0	1											SL2
			0	1	0											SR1
			0	1	1											SR2
						1										CI
									1							CO
								0								
								0		1	0	0				SUT
								0		0	1	0				CUT
								0		1	1	0				TUT
								0				1				RPT
								0					1			SWA
								0					0			CWA
								0						1		POW
								0							1	ALRM
								1								
								1		1						TABT
								1			1					PW
								1				1				JH
								1					1			
								1						1		TS
								1							1	CYD

BRANCH CONDITION

4	5	6	7	8	9	10	11	12	
0	0	1							C
0	0		1						V
0	0			1					G
0	0				1				L
0	0					1			MSKI
0	0						1		WAIT
0	0							1	
0	1	1							ATNX
0	1		1						HWL
0	1			1					QUE
0	1				1				RR
0	1					1			ATN
0	1						1		MAC
0	1							1	
1	0	1							SNGL
1	0		1						CATN
1	0			1					DC
1	0				1				DRD
1	0					1			MALF
1	0						1		PPF
1	0							1	
1	1	1							HWIO
1	1		1						NORM
1	1			1					CNTR
1	1				1				ARST
1	1	0	0	0	0	1	0	0	UT
1	1						1		SHORT
1	1							1	

NOTES

NAME	TITLE	DATE
V. PEREZ	DRAFT	8-21-73
	CHK	
	ENGR	
	DIR ENG	

TITLE FUNCTIONAL SCHEMATIC
MODEL 7/16 HSALU & 7/32 PROCESSOR

TASK NO. 03047
DWG NO. 01-079M01 A/D/D08
SHEET 4 OF 36

MICRO PROGRAM FORMAT

INTERDATA

FDEC10
IRO01
IRO11
IRO21
IRO31
IRO41
IRO51
IRO61
IRO71
SEE NOTE 2
FDEC20
PSW110
PSW111
(NOTE 3)
PSW110
PRIVO
BRCH1
RD131
RD141
RD151
RD161
RD171
RD181
RD191
RD201
RD211
RD221
A04 7/16 HSALU-19-122F01 7/32 W/F.P.-19-122F38 7/32 W/O F.P.-19-122F47
A05 7/16 HSALU-19-122F04 7/32 W/F.P.-19-122F41 7/32 W/O F.P.-19-122F50
A06 7/16 HSALU-19-122F05 7/32 - 19-122F44
A16 7/16 HSALU-19-122F02 7/32 W/F.P.-19-122F39 7/32 W/O F.P.-19-122F48
A17 7/16 HSALU-N/EQUIPPED 7/32 W/F.P.-19-122F42 7/32 W/O F.P.-19-122F51
A18 7/16 HSALU-19-122F06 7/32 - 19-122F45
A27 7/16 HSALU-19-122F03 7/32 W/F.P.-19-122F40 7/32 W/O F.P.-19-122F49
A28 7/16 HSALU-N/EQUIPPED 7/32 W/F.P.-19-122F43 7/32 W/O F.P.-19-122F52
A29 7/16 HSALU-19-122F07 7/32 - 19-122F46
DROM 1 (F-7/16)
DROM 1 (H)
DROM 2
SEE DROM STRAPPING OPTIONS, SHEET 9
A01 19-005 HG
A30 19-043 HGO
A19 19-043 HGO
A07 19-043 HGO
SRA50
SRA60
SRA70
SRA80
SRA90
SRA100
SRA110
SRA120
SRA130
SRA140
SRA150
9N4
9N2
9N5
R11 470
R13 470
R12 470
R7 470
R10 470
R8 470
R9 470
R3 470
R4 470
(FALSE BUS)
RARSTPO
RSTOPO
CLK1B
A103 19-017 HG
A101 19-058 HG
A03 19-035
A15 19-035
A26 19-035
COUNT UP
COUNT DOWN
CARRY
BORROW
LOAD
CLEAR
N.C.
R1 470
R2 470
R6 1K
A02 19-101 B
A01 19-015 HG
A13 19-015 HG
A14 19-101 B
A25 19-015 HG
A24 19-016 HG
A32 19-015 HG
A33 19-024 HA
RAR151 6B1
RAR141 6A2
RAR131 6B1
RAR121 6A2
RAR111 6B1
RAR101 6A1
RAR091 6B1
RAR081 6A1
RAR090 107-3
RAR080 110-3
RAR060 108-3
RAR070 109-3
RAR151A 6E2
RAR141A 6S2
RAR131A 6E2
RAR121A 6S2
RAR111A 6E1
RAR101A 6S1
RAR091A 6E1
RAR081A 6S1
RAR051 6C1
RAR050 6N1
EN000 6K1
EN010 6D1
EN100 6F1
EN110 6H1
EN ECSTB 6G1
FLRARO
PSW231
119-3

NOTES
1. ALL APPARATUS THIS SHEET LOCATED ON CPU-B BOARD, MODEL 7/32 USES 35-523, MODEL 7/16 HSALU USES 35-549.
2. FOR 7/32 OPTION TERMINAL 'F' IS STRAPPED TO TERMINAL 'E'; FOR 7/16 HSALU TERMINAL 'F' IS STRAPPED TO TERMINAL 'G'.
3. FOR 7/16 HSALU THIS POINT IS GROUNDED AT THE BACK PANEL.

REVISIONS
A REA J1, A101 P/O WAS 19-017.
[illegible] 2392 — 2-12-75 [illegible]

NAME | DATE
DRAFT
CHK
ENGR
DIR ENG

TITLE FUNCTIONAL SCHEMATIC MODELS 7/16 HSALU & 7/32 PROCESSOR
ASA NO. 03047
DWG NO. 01-079101 R01 D08
SHEET 5 OF 36

INTERDATA®

SN5 RAR08I
SN4 RAR10I
SN3 RAR12I
SN2 RAR14I

554 RAR08IA
553 RAR11IA
552 RAR13IA
551 RAR15IA

RAR08IA 555
RAR10IA 554
RAR12IA 553
RAR14IA 552

A31 7/16 HSALU-19-122-F08 7/32 - 19-122 - F53	A32 7/16 HSALU-19-122-F14 7/32 - 19-122 - F59	A33 7/16 HSALU-19-122-F20 7/32 - 19-122 - F65	A34 7/16-HSALU-19-122-F26 7/32 - 19-122 - F71	A35 7/16-HSALU-19-122-F32 7/32 - 19-122 - F77
A42 7/16 HSALU-19-122-F09 7/32 - 19-122 - F54	A43 7/16 HSALU-19-122-F15 7/32 - 19-122 - F60	A44 7/16 HSALU-19-122-F21 7/32 - 19-122 - F66	A45 7/16 HSALU-19-122-F27 7/32-19-122 - F72	A46 7/16-HSALU-19-122-F33 7/32 - 19-122 - F78
A53 7/16 HSALU-19-122-F10 7/32 - 19-122 - F55	A54 7/16 HSALU-19-122-F16 7/32-19-122 - F61	A55 7/16 HSALU-19-122-F22 7/32 - 19-122 - F67	A56 7/16 HSALU-19-122-F28 7/32 - 19-122 - F73	A57 7/16-HSALU-19-122-F34 7/32 - 19-122 - F79
A64 7/16 HSALU-19-122-F11 7/32-19-122 - F56	A65 7/16 HSALU-19-122-F17 7/32 - 19-122 -F62	A66 7/16 HSALU-19-122-F23 7/32 -19-122 -F68	A67 7/16 HSALU-19-122-F29 7/32 - 19-122- F74	A68 7/16 HSALU-19-122-F35 7/32 - 19-122 - F80
A75 7/16 HSALU-19-122-F12 7/32 -19-122 - F57	A76 7/16 HSALU-19-122-F18 7/32 - 19-122 - F63	A77 7/16 HSALU-19-122-F24 7/32 - 19-122 - F69	A78 7/16 HSALU-19-122-F30 7/32 - 19-122 - F75	A79 7/16 HSALU-19-122-F36 7/32 - 19-122 - F81
A86 7/16 HSALU-19-122-F13 7/32 -19-122 -F58	A87 7/16 HSALU-19-122-F19 7/32 - 19-122 -F64	A88 7/16 HDALU-19-122-F25 7/32 - 19-122 -F70	A89 7/16 HSALU-19-122-F31 7/32-19-122- F76	A90 7/16 HSALU-19-122-F37 7/32-19-122 -F82

ADDRESS'000-0FF — ADDRESS 100-1FF — ADDRESS 200-2FF — ADDRESS 300-3FF — ADDRESS 400- 4FF

A36 7/16 HSALU-19-122-N/EQUIP 7/32 - 19-122 - F83	A37 7/16 HSALU-19-122-N/EQUIP 7/32 - 19-122 - F89 7/32 w/o F.P. N/EQUIP	12 SRD230 7L7 11 SRD220 7L6 10 SRD210 7L6 09 SRD200 7L5	
A47 7/16 HSALU-19-122-N/EQUIP 7/32— 19-122 - F84	A48 7/16 HSALU-19-122-N/EQUIP 7/32 - 19-122 - F90 7/32 w/o F.P. N/EQUIP	12 SRD190 7L4 11 SRD180 7L3 10 SRD170 7L2 09 SRD160 7L1	
A58 7/16 HSALU-19-122-N/EQUIP 7/32 - 19-122 - F85	A59 7/16 HSALU-19-122-N/EQUIP 7/32 - 19-122 - F91 7/32 w/o F.P. N/EQUIP	12 SRD150 7F7 11 SRD140 7F6,8L9 10 SRD130 7F6 09 SRD120 7F5,8L9	
A69 7/16 HSALU-19-122-N/EQUIP 7/32 - 19-122 - F86	A70 7/16 HSALU-19-122-N/EQUIP 7/32 - 19-122 - F92 7/32 w/o F.P. N/EQUIP	12 SRD110 7F4,8L9 11 SRD100 7F3 10 SRD090 7F2,8L9 09 SRD080 7F1	
A80 7/16 HSALU-19-122-N/EQUIP 7/32 - 19-122 - F87	A81 7/16 HSALU-19-122-N/EQUIP 7/32 - 19-122 - F93 7/32 w/o F.P. N/EQUIP	12 SRD070 7A4,8L9 11 SRD060 7A6 10 SRD050 7A6, 09 SRD040 7A5	T V
A91 7/16 HSALU-19-122-N/EQUIP 7/32 - 19-122 - F88	A92 7/16 HSALU-19-122-N/EQUIP 7/32 -19-122 - F92 7/32 w/o F.P. N/EQUIP	12 SRD030 7A4 11 SRD020 7A3 10 SRD010 7A2 09 SRD000 7A1	

ADDRESS 500-5FF — ADDRESS 600-6FF

READ ONLY MEMORY

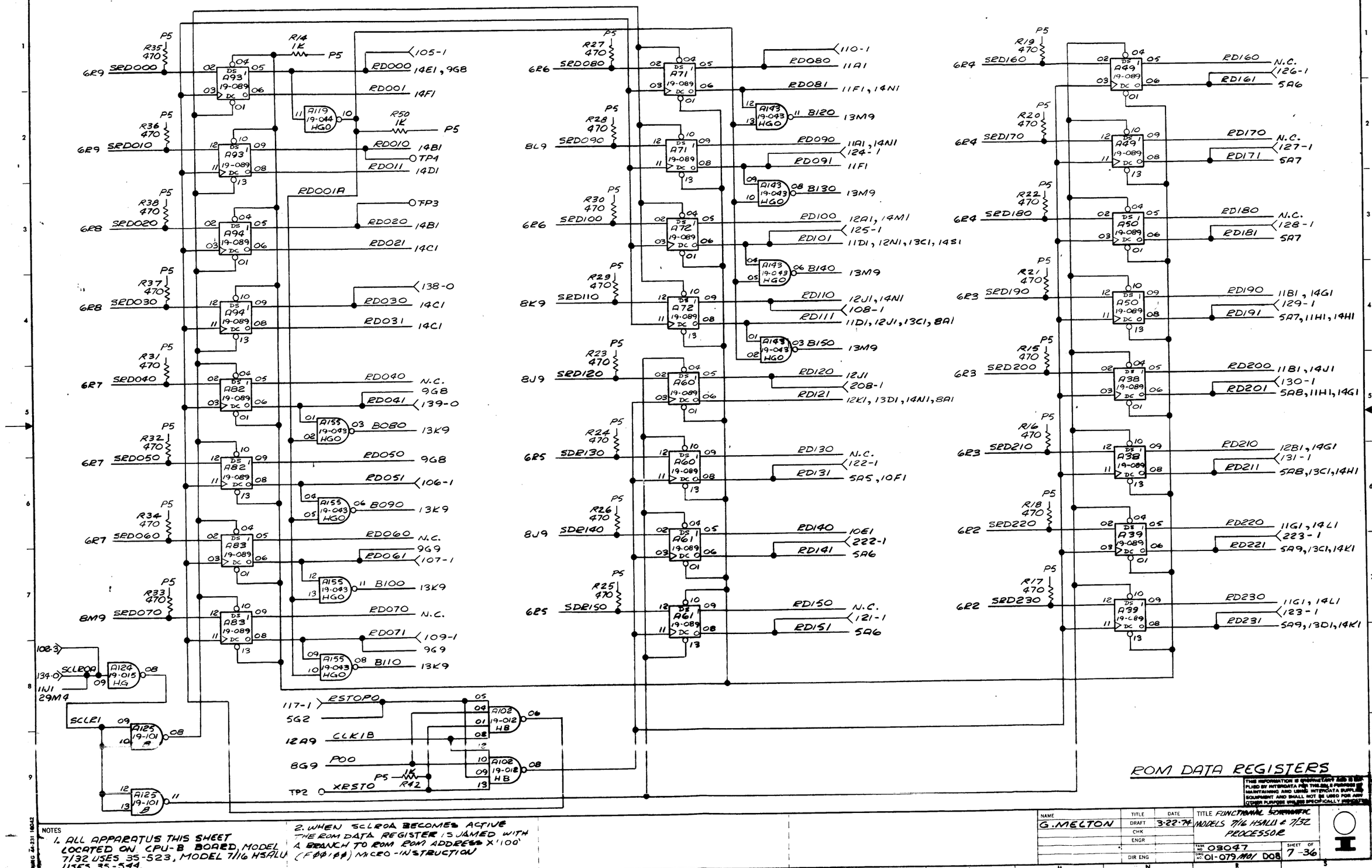
NOTES
1. ALL APPARATUS THIS SHEET LOCATED ON CPU-B BOARD, MODEL 7/32 USES 35-523, MODEL 7/16 HSALU USES 35-544.

NAME	TITLE	DATE
B GRAY	DRAFT	10-19-73
	CHK	
	ENGR	
	DIR ENG	

TITLE FUNCTIONAL SCHEMATIC MODELS 7/16 HSALU & 7/32 PROCESSOR

TASK NO. 03047 DWG NO. 01-079 M01 D08 SHEET 6 OF 36

INTERDATA®



ROM DATA REGISTERS

NOTES
1. ALL APPARATUS THIS SHEET LOCATED ON CPU-B BOARD, MODEL 7/32 USES 35-523, MODEL 7/16 HSALU USES 35-544.
2. WHEN SCLR0A BECOMES ACTIVE THE ROM DATA REGISTER IS LOADED WITH A BRANCH TO ROM ROM ADDRESS X'100' (F00/00) MICRO-INSTRUCTION

NAME	TITLE	DATE	TITLE
G.MELTON	DRAFT	3-22-74	FUNCTIONAL SCHEMATIC MODELS 7/16 HSALU & 7/32 PROCESSOR
	CHK		
	ENGR		
	DIR ENG		DWG NO 08047
			SHEET OF 7-36
			SYS NO 01-079/M01 D08

INTERDATA®

CALCULATE ADDRESS
AND
IR OP CODE LOGIC

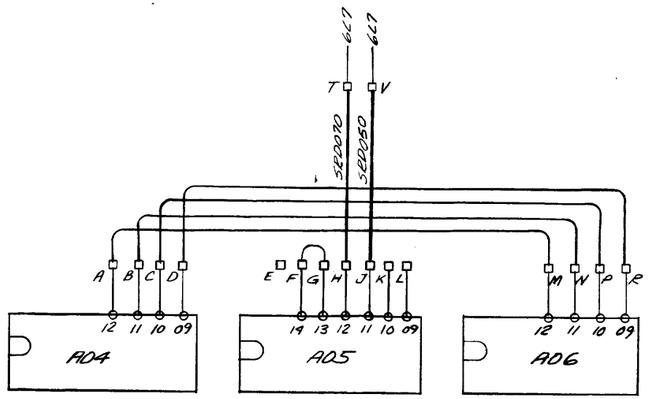
NOTES
1. ALL APPARATUS THIS SHEET LOCATED ON CPU-B BOARD, MODEL 7/32 USES 35-529, MODEL 7/16 HSALU USES 35-549.

REVISIONS
AREA G2, ANN PIN WAS 19-017.
2392 12-18-73 EDT

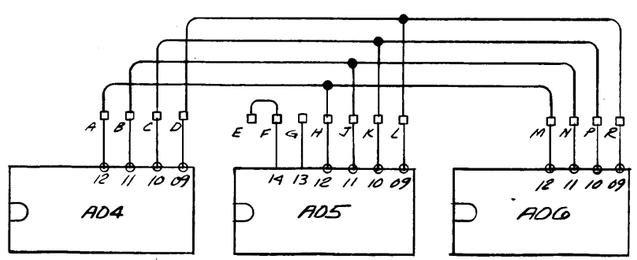
NAME	TITLE	DATE
G. MELTON	DRAFT	3-20-74
	CHK	
	ENGR	
	DIR ENG	

TITLE FUNCTIONAL SCHEMATIC
MODELS 7/16 HSALU & 7/32 PROCESSOR
TASK NO. 03047
DWG NO. 01-079[illegible]D08
SHEET OF 8-36

DRAWING 44-231 16043

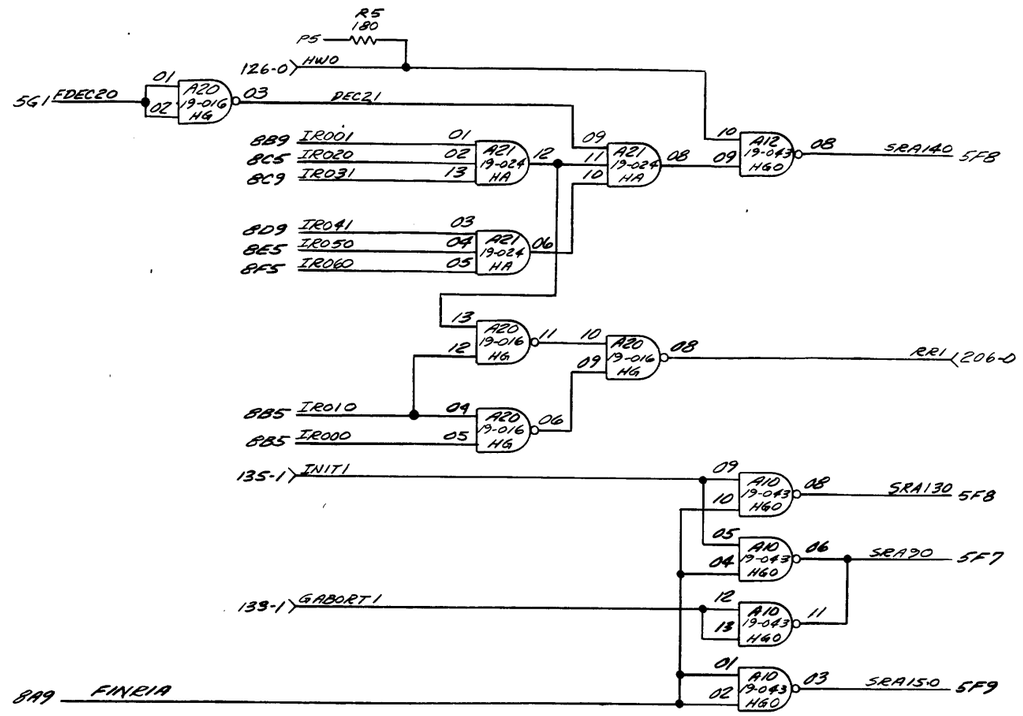


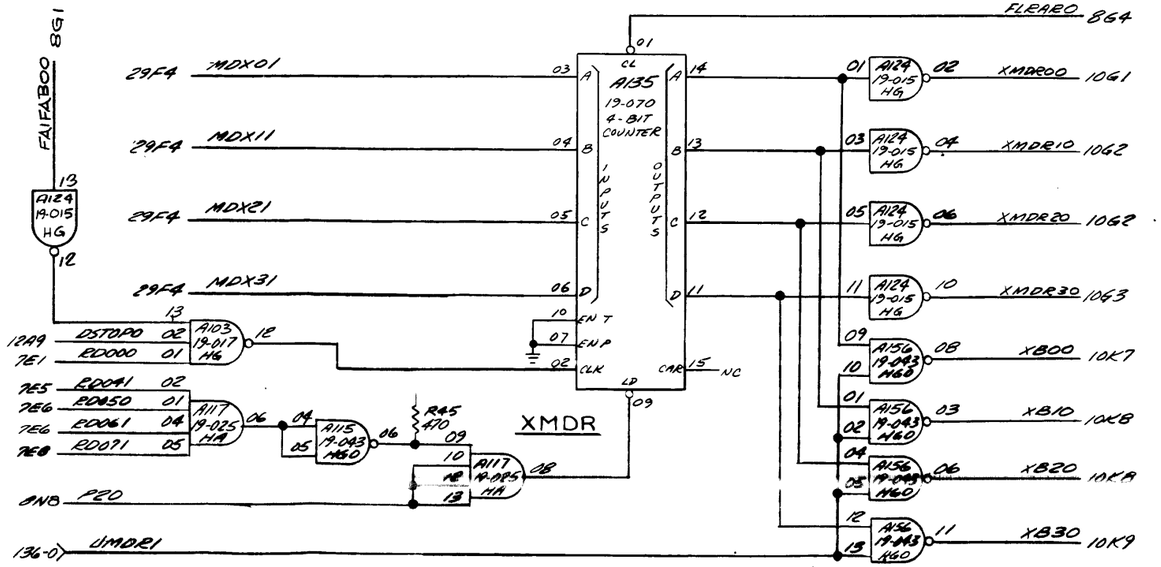
DEOM STRAPPING FOR 7/16 HSALU



DEOM STRAPPING FOR 7/32

STRAPPING OPTIONS FOR 7/32 & 7/16 HSALU





NOTES
1. ALL APPARATUS THIS SHEET LOCATED ON CPU-B BOARD, MODEL 7/32 USES 35-523, MODEL 7/16 HSALU USES 35-544.

NAME	TITLE	DATE	TITLE FUNCTIONAL SCHEMATIC
V. PERRI	DRAFT		MODELS 7/16 HSALU & 7/32 PROCESSOR
	CHK		
	ENGR		
	DIR ENG		TASK NO. 03047
			DWG NO. 01-079 M01 D08
			SHEET OF 9 36

INTERDATA®

STRAPPING OPTIONS	
7/32 W/ MEMORY ACCESS CONTROL OR DMABC	7/32 W/O MEMORY ACCESS CONTROL OR DMABC
TERMINAL 'A1' TO TERMINAL 'A3'	TERMINAL 'A2' TO TERMINAL 'A3'
TERMINAL 'B1' TO TERMINAL 'B3'	TERMINAL 'B2' TO TERMINAL 'B3'
TERMINAL 'C1' TO TERMINAL 'C3'	TERMINAL 'C2' TO TERMINAL 'C3'
TERMINAL 'D1' TO TERMINAL 'D3'	TERMINAL 'D2' TO TERMINAL 'D3'

X BUS LOGIC

X MAR

X MAS

236-0 ARH120, XMDR00 (9N6)
RDI31, RDI40 (7K6, 7K7)
SELA1 (131-0)
237-0 ARH130, XMDR10 (9N7)
238-0 ARH140, XMDR20 (9N7)
239-0 ARH150, XMDR30 (9N8)
A122 19-038 4 TO 1 MUX → XA00, XA10
A112 19-038 4 TO 1 MUX → XA20, XA30
A95 19-024 HA → MDR011 (8J1), PI1 (8M8)

13N1 S121, S131, S141, S151
A169 19-066 2:1 MUX
LARH120 (119-0), LARH130 (219-0), LARH140 (120-0), LARH150 (220-0)
14M9 LAR1
12A9 CLK1B, DSTOPO
A101 19-058 HG
11L1 DCLKO
229-0 INAR1
232-1 CPMAR1
14K9 LMARO
A146 19-070 4 BIT COUNT
MXR121, MXR131, MXR141, MXR151

29G1 GPSELI
P5 R49 470
103-1 FPSELO
A119 19-044 HGO
5A3 PSW111
SEE NOTE 2
R46 1K
A134 19-035 4 BIT UP/DWN CNTR
SEE LEGEND IN AREA A1 FOR STRAPPING OPTIONS.
124-0 LMAS0
11J4 ERADO
A144 19-015 HG
231-0 UMAR1
A145 19-043 HGO
R55 470, R53 470, R52 470, R54 470
A118 19-101 B
XMA150 (121-0), XMA140 (122-0), XMA130 (123-0), XMA120 (127-0), XMA141 (128-0)
A133 19-027 QUAD LATCH
XB-BUS LATCH
XB30 (12N9, 9N9), XB20 (12N9, 9N9), XB10 (12N9, 9N9), XB00 (12M9, 9N8)

8H8 GEX130
2I3-0 ULOC1
A123 19-024 HA
A157 19-043 HGO
A168 19-043 HGO
B150, B140, B130, B120 → 13M9
14N9 ULOCH1
P5 R48 1K
X LOC A158 19-070 4 BIT COUNTER
129-0 ILOC1
209-1 CPLOC1
14L9 LLOCO

14G8 S01, 14H8 S11, 14H8 S21, 14J8 S31, 14J8 M1, 221-1 C5V1
A121 19-039 4 BIT ALU
N.C., A=B, CN+4
FO, F1, F2, F3 → TP6, TP7, TP5, TP8
X531, X521, X511, X501

NOTES
1. ALL APPARATUS THIS SHEET LOCATED ON CPU-B BOARD, MODEL 7/32 USES 35-523, MODEL 7/16 HSALU USES 35-544.
2. REMOVE JUMPER 'X' TO 'Y' W/MAC OR DMABC.
* NOT EQUIPPED ON 7/16 WITH HSALU.

REVISIONS			
A8ER B4, A101 PIN WAS 19-017. A8ER A5, MNEM DCLKO WAS NOT SPEC'D.			
2N	2392	2-12-75	201

NAME	TITLE	DATE
G. MELTON	DRAFT	3-20-75
	CHK	
	ENGR	
	DIR ENG	

TITLE: FUNCTIONAL SCHEMATIC MODELS 7/16 HSALU & 7/32 PROCESSOR
TASK NO. 03047
DWG NO. 01-079MXENDO8
SHEET OF 10-36

BLUWING 44-231 11642

INTERDATA®

REVISIONS			
CUT CONN. AREA B4, A126-12, FROM A126-13. AREA B1, MNEM. CMNDO NOT SPEC'D. CUT CONN. BETWEEN, A108-9 AREA L3 & A108-11 AREA L2. AREA L1, MNEM, DCLKO WAS NOT SPEC'D AREA N3, A08 PINS. 1, 13, 2 & 12 WAS NOT SPEC'D. AREA R3, MNEM. DCLK1A WAS NOT SPEC.			
	2392	12-14-75	R01

IMCMDO 14F9
ED080 7K1
ED090 7K2
ED190 7J4
ED200 7J5
CMNDO 14E9
EMODO 29E4
ED101 7K3
ED111 7K4
YLS1 12H1
ED081 7K1
ED091 7K2
SEE NOTE 2
PSW271 111-0
ED220 7J6
ED230 7J7
YHS1 12M1
XEP1
RA01 1K
P5
ED191 7J4
ED201 7J5
13D, 12B5, 10H5
READO 7A8
SCLEFO 7A8
CLK1B 12A9
CLKO 104-3
DONO 13D2-0
EEAD1 20D7-1
1PH1
OSTCPO 12A9
DCLKO 10A5
GND 105-3
GND 112-3
GND 100-3
P5 101-3
P5 106-3
N.C. 120-3
N.C. 111-3
TPI ETITO 103-3
DCLK1A 12A8, 13A3
MEMSO 1394
MEGRLOO 1396
MEGRLFO 1398
MEGEHOO 1206
MEGEHFO 1298
MEXROO 12M9
MEXRFO 12N9

REGISTER STACK DECODING

NOTES
1. ALL APPARATUS THIS SHEET LOCATED ON CPU-B BOARD; MODEL 7/32 USES 35-523; MODEL 7/16 HSALU USES 35-544.
2. FOR 7/16 HSALU THIS POINT IS GROUNDED AT THE BACK PANEL.
* NOT EQUIPPED ON 7/16 WITH HSALU

NAME	TITLE	DATE	TITLE
G. MELTON	DRAFT	3-21-79	FUNCTIONAL SCHEMATIC MODELS 7/16 HSALU & 7/32 PROCESSOR
	CHK		
	ENGR		
	DIR ENG		TASK NO 03047; DRD NO 01-079MMB0008; SHEET 11 OF 36

INTERDATA®

REVISIONS
AREA A8, A109-10 WAS CONN TO 102-1; MNEM. DCLKIA HAS NOT SPEC'D.
23392 - 2-A-75 201
AREA C2-G2. R58-R61 WERE ADDED. C68-C71 WERE ADDED. GNDS WERE ADDED.
GP 3478 H 12-78 201

READ 11K1 RD00 K1, 7K3 · READ 11J1 RD210 7S6 · C68 220pf · R58 2.2Ω · IR08 I14-0 · IR12 I16-0 · IR09 I14-1 · IR13 I16-1 · I9M2 · SEL 13A5 · SCI 13A5 · C70 220 pf · R60 22Ω · IR10 I15-0 · IR14 I17-0 · C69 220pf · R59 2.2Ω · IR11 I15-1 · IR15 I17-1 · I9M3 · C71 220pf · R61 22Ω · I9M4 · SDI 13A5 · YLSI 11D1 · RD111 7K4 · RD120 7K5 · RD110 7K4 · RD121 7K5 · RD100 9I · YHSI 11G1 · RD101 9K3

A107 19-056 S60 · R44 330 · PS · A150 19-065 2:1 MUX · A138 19-065 2:1 MUX · SEL · STB G · 1A 1B 2A 2B 3A 3B 4A 4B · 1Y 2Y 3Y 4Y · 13A5 SAI

A109 19-026 HG · A100 19-026 HG · 09 N.C. · 08

S001 13F1 · S011 13F1 · S021 13F1 · S031 13F1 · S041 13H1 · S051 13H1 · S061 13H1 · S071 13J1 · S081 13K1 · S091 13L1 · S101 13L1 · S111 13L1 · S151 13N1 · S141 13N1 · S131 13N1 · S121 13N1

A166 · A165 · A164 · A163 · A162 · A177 · A176 · A175 · A174 · A173 · A109 19-017 HG · 11J9 MEGRH00 · 11K9 MEGRH10 · 11J3 DCLKIA

GRH (00-15) SET Ø · GRH (00-15) SET F

111,9B0 SST,8N1 · D32-1 DSTOD0 · 14E9 CMNDO · E02,1 CLK1B · S61,8N1 DMA,INT 13A5,7D9

B000 B010 B020 B030 · B040 B050 B060 B070 · B080 B090 B0A0 B0B0 · B120 B130 B140 B150 · MEXR00 MEXR10 · XB000 XB010 XB020 XB030

STACK DECODING AND GENERAL REG. HIGH

NOTES
1. ALL APPARATUS THIS SHEET LOCATED ON CPU-B BOARD, MODEL 7/32 USES 35-523, MODEL 7/16 HSALU USES 35-594.

* NOT EQUIPPED ON 7/16 WITH HSALU.

NAME	TITLE	DATE
B GRAY	DRAFT	
	CHK	
	ENGR	
	DIR ENG	

TITLE FUNCTIONAL SCHEMATIC
MODELS 7/16 HSALU & 7/32 PROCESSOR

TASK NO. 03047 · DWG NO. 01-079M01R02D08 · SHEET OF 12-36

FORM 0 44-231 10M42

INTERDATA®

REVISIONS			
ADDED B3, MNEM. DCLK1A, WAS DSTOPO.			
A	DEN 2392	2-12-75	R01

A127 19-065 2:1 MUX STTL — ENB, SEL, 1A 1B 2A 2B 3A 3B 4A 4B, 1Y 2Y 3Y 4Y; NC NC RD101 RD211 RD111 RD221 RD121 RD231; 7K3 7J6 7K4 7J7 7K5 7J8; READO 1171

A131 19-075 16x9 REG. STACK — DATA IN D1 D2 D3 D4; ADRS A B C D; WE; ME OUTPUTS S1 S2 S3 S4; 1189 MEMSO; 1AE9 CMNDO; 1IR3 DCLK1A; 1BA9 CLK1B; A104 19-017 HG

A130 19-075 16x9 REG. STACK

A129 19-075 16x9 REG. STACK

A128 19-075 16x4 REG. STACK

MR0 MR1 MR2 MR3 MR4 MR5 MR6 PSWL

A142 19-075 16x9 REG. STACK — 1189 MEMSO; 1109 MEGRLOO; 1284 SA1; 12E1 SB1; 12E1 SC1; 12H1 SD1

A141 19-075 16x9 REG. STACK

A140 19-075 16x9 REG. STACK

A139 19-075 16x4 REG. STACK

GRL (16-31) SET Ø

A154 19-075 16x9 REG. STACK * — 11F9 MEGRL1O

A153 19-075 16x9 REG. STACK *

A152 19-075 16x9 REG. STACK *

A151 19-075 16x4 REG. STACK *

GRL (16-31) SET F

S121 → 112-1, 10A3, 29G2, 12R5

S141 → 111-1, 10A3, 29G2, 12R4

S131 → 212-1, 10A3, 29G2, 12R4

S151 → 211-1, 10A3, 29G2, 12R4

C1 THRU C10 .001 MED → D5

C11 THRU C67 33 MED. → D6

DECOUPLING CAPACITORS

REGISTER STACKS

NOTES
1. ALL APPARATUS THIS SHEET LOCATED ON CPU-B BOARD; MODEL 7/32 USES 35-523; MODEL 7/16 HSALU USES 35-549.

* NOT EQUIPPED ON 7/16 WITH HSALU

NAME	DATE	TITLE FUNCTIONAL SCHEMATIC
CHRIS JENSEN	30OCT73	MODELS 7/16 HSALU & 7/32 PROCESSOR
	ENGR	03047
	DWG ENG	01-0179NOI R01D08
		SHEET OF 13-36

DWG NO. 44-231 16042

INTERDATA®

DECODING LOGIC

NOTES
1. ALL APPARATUS THIS SHEET LOCATED ON CPU-B BOARD; MODEL 7/32 USES 35-523, MODEL 7/16 HSALU USES 35-544.

REVISIONS

	ADDED: AT LOC E9 CODO 1181			
—	MCH	2392	3-4-75	R01

NAME	TITLE	DATE
G. MELTON	DRAFT	
	CHK	
	ENGR	
	DIR ENG	

TITLE: FUNCTIONAL SCHEMATIC MODELS 7/16 HSALU & 7/32 PROCESSOR

TASK NO. 03047
DWG NO. 01-079M01R01D03
SHEET 14 OF 36

111-0 > D000 — A154 19-036 HPO pins 06, 05, 04 — 5001A — 16B1, 17B1, 18C1, 22D9, 23A4

PS — R85 330 — A153 19-015 HG pins 03, 04 — D001 — 17A5

211-0 > D010 — A154 19-036 HPO pins 03, 02, 01 — 5011A — 16C1, 17B1, 18E1, 22E9, 23A4

PS — R86 330 — A153 19-015 HG pins 05, 06 — D011 — 17B5

112-0 > D020 — A154 19-036 HPO pins 11, 13, 12 — 5021A — 16C1, 17B1, 18G1, 22F9, 23A4

PS — R87 330 — A153 19-015 HG pins 09, 08 — D021 — 17C5

212-0 > D030 — A154 19-036 HPO pins 08, 10, 09 — 5031A — 16C1, 17C1, 18H1, 22G9, 23B4

PS — R84 330 — A153 19-015 HG pins 11, 10 — D031 — 17D5

113-0 > D040 — A152 19-036 HPO pins 06, 05, 04 — 5041A — 16E1, 17E1, 18J1, 22K9, 23C4

PS — R73 330 — A151 19-015 HG pins 03, 04 — D041 — 17E5

213-0 > D050 — A152 19-036 HPO pins 03, 02, 01 — 5051A — 16E1, 17F1, 18L1, 22K9, 23C4

PS — R61 330 — A151 19-015 HG pins 05, 06 — D051 — 17F5

114-0 > D060 — A152 19-036 HPO pins 11, 13, 12 — 5061A — 16E1, 17F1, 18M1, 22M9, 23C4

PS — R62 330 — A151 19-015 HG pins 09, 08 — D061 — 17F5

214-0 > D070 — A152 19-036 HPO pins 08, 10, 09 — 5071A — 16F1, 17F1, 18R1, 22M9, 23C4

PS — R72 330 — A151 19-015 HG pins 11, 10 — D071 — 17G5

224-1 > FUN16 — AB9 19-101 HPO pins 13, 12, 1 — EDIS1

115-0 > D080 — A141 19-036 HPO pins 06, 05, 04 — 5081A — 16H1, 17J1, 19C1, 21D9, 23D4

PS — R58 330 — A140 19-015 HG pins 03, 04 — D081 — 17H5

215-0 > D090 — A141 19-036 HPO pins 03, 02, 01 — 5091A — 16H1, 17J1, 19E1, 21E1, 23E4

PS — R60 330 — A140 19-015 HG pins 05, 06 — D091 — 17J5

116-0 > D100 — A141 19-036 HPO pins 11, 13, 12 — 5101A — 16J1, 17J1, 19F1, 21C1, 23E4

PS — R59 330 — A140 19-015 HG pins 09, 08 — D101 — 17K5

216-0 > D110 — A141 19-036 HPO pins 08, 10, 09 — 5111A — 16J1, 17J1, 19H1, 21D1, 23E4

PS — R57 330 — A140 19-015 HG pins 11, 10 — D111 — 17L5

117-0 > D120 — A135 19-036 HPO pins 06, 05, 04 — 5121A — 16L1, 17M1, 19J1, 23F4, 21K9

PS — R55 330 — A139 19-015 HG pins 03, 04 — D121 — 17L5

217-0 > D130 — A135 19-036 HPO pins 03, 02, 01 — 5131A — 16L1, 17N1, 19L1, 21K9, 23F4

PS — R56 330 — A139 19-015 HG pins 05, 06 — D131 — 17M5

118-0 > D140 — A135 19-036 HPO pins 11, 13, 12 — 5141A — 16L1, 17N1, 19M1, 21M9, 23F4

PS — R53 330 — A139 19-015 HG pins 09, 08 — D141 — 17N5

218-0 > D150 — A135 19-036 HPO pins 08, 10, 09 — 5151A — 16N1, 19R1, 21M9, 23G4, 36N5, 17R1

PS — R54 330 — A139 19-015 HG pins 11, 10 — D151 — 17R5

I/O DRIVERS & RECEIVERS

NOTES
1. ALL APPARATUS THIS SHEET LOCATED ON CPU-C BOARD, 3S-3C4 MAU.

NAME	DATE
DRAFT	
ENGR	
MFG ENG	

TITLE FUNCTIONAL SCHEMATIC
MODEL 5 7/16 HSALU & 7/32 PROCESSOR
DWG 23097
DWG NO 01-079 M01 D08
SHEET 15 OF 36

INTERDATA®

MAR

MAS

MAR & MAS

W/ MAC OR DMABC	W/O MAC OR DMABC
A1 - A3	A2 - A3
B1 - B3	B2 - B3
C1 - C3	C2 - C3
D1 - D3	D2 - D3
E1 - E3	E2 - E3
F1 - F3	F2 - F3
G1 - G3	G2 - G3
H1 - H3	H2 - H3
REMOVE J1 - J2	J1 - J2

STRAPPING OPTIONS

NOTES
1. ALL APPARATUS THIS SHEET LOCATED ON CPU-C BOARD, 35-5214 MOI.

REVISIONS
AT LOC N3 A44 WAS 19-089
- ECR 2392 3-7-75 R01
AREA FOR P5, 2 CAPS
A30 PIN WAS 19-044.
R02 ECR 2441 - 3-26-75

	NAME	TITLE	DATE
		DRAFT	
		CHK	
		ENGR	
		DIR ENG	

TITLE FUNCTIONAL SCHEMATIC
MODELS 7/16 HSALU & 7/32 PROCESSOR

TASK NO 23097 SHEET OF
DWG NO 01-079MOIRORDAG 16-36

INTERDATA®

LOC

A62 19-070 4-BIT CNTR
A63 19-070 4-BIT CNTR
A65 19-070 4-BIT CNTR
A66 19-070 4-BIT CNTR
A71 19-016 HG
A72 19-038 4 TO 1 MUX
A73 19-038 4 TO 1 MUX
A74 19-038 4 TO 1 MUX
A75 19-038 4 TO 1 MUX
A76 19-038 4 TO 1 MUX
A77 19-038 4 TO 1 MUX
A78 19-038 4 TO 1 MUX
A79 19-038 4 TO 1 MUX
A80 19-017 HG
A81 19-024 HA
A82 19-043 HG
A84 19-043 HG
A86 19-043 HG
A88 19-043 HG
A92 19-036 HBO
A85D DEC 19-039 DS DIFF.
R32 1K
R33 1K
R105 180Ω

NOTES
1. ALL APPARATUS THIS SHEET LOCATED ON CPU-C BOARD, 35-524 M01.

LOCATION COUNTER & B BUS MUX

NAME	TITLE	DATE
E. ROE	DRAFT	5-8-74
	CHK	
	ENGR	
	DIR ENG	

TITLE FUNCTIONAL SCHEMATIC
MODEL 7/16 HSALU-7/32 PROCESSOR
TASK NO. 03047
DWG NO. 01-079 M01 D08
SHEET OF 17-36

INTERDATA®

CLK/IC 78I
LMDR/ 24E9
DSTORA/7DI
ENHI/ 16S8
FPSELO/ 103-1
16GG
SOOIA/ 15FI
SOIIA/ 15F2
SO2IA/ 15F3
SO3IA/ 15F4
SO4IA/ 15F5
SO5IA/ 15F6
SO6IA/ 15F7
SO7IA/ 15F8
16MT/ PSELI
1955
ENMSI/ 1983
CKMOHO
1985 CLMDRO
RDISI/ 24J3
LOADI/ 16V4
A70 19-012 HB
A89 19-101 HB
A40 19-015 HG
A34 19-163 LH
A21 19-163 LH
A28 19-015 HG
A53 19-089
A54 19-089
A69 19-089
A63 19-089
A09 19-017 HG
A42 19-043 HG
A33 19-029 HA
A42 19-043 HBO
A41 19-036 HPO
A29 19-036 HPO
R112 22
C63 220pf
R26 470
R27 470
R25 470
R28 470
R21 470
R22 470
R20 470
R23 470
R29 330
205-I/ MDR000
2413, 1745 MDR001
124-I/ MDD00
22E4 MDR010
206-I/ MDR011
2413, 1785 MDR011
125-I/ MDD01
22E4 MDR020
17C5 MDR021
126-I/ MDD20
2214 MDR030
1705 MDR031
127-I/ MDD30
22H4 MDR040
1705 MDR041
128-I/ MDD40
22K4 MDR050
17E5 MDR051
129-I/ MDD50
22L4 MDR060
17F5 MDR061
130-I/ MDD60
2214 MDR070
1705 MDR071
131-I/ MDD70
17H1 BLOADI/
20RI, BIAO

MEMORY DATA REGISTER HIGH

NOTES 1. ALL APPARATUS THIS SHEET LOCATED ON CPU-C BOARD 35-524 M01.

A34-02
MS/AV 3/51 - 4-19-77 R01
AREA D3-R3 GATES A34 & A21 WERE 19-153 LS.
GS/AV 13355 9-16-77 R02

REVISIONS
AREA D3-R3. GATES A34 & A21 WERE SPEC AS 19-06, HG DEVICES
AREA E3. ADDED RIIE & C62 TO

NAME	TITLE	DATE
E. ROE	DRAFT	5-4-74
	CHK	
	ENGR	
	DIR ENG	

TITLE FUNCTIONAL SCHEMATIC
MODEL 7/16 HSALU-7/32 PROCESSOR
TASK NO. 03047
DWG NO. 01-079M01R02D08
SHEET 18 OF 96

DRAWING 44-231 16042

CLKIC/1781
LMDBL/2429
DSTOB2/1D1
ENLL/1658
A32 19-015 HG
1624 LMDBO
A70 19-012 HB
CKMDLO
105-1 CLMDBO
18A4
PS
R4 1K
225-1 ENMS1
1852
SCBIA/1SN1
SOBIA/1SN2
S1OIA/1SN3
S11IA/1SN4
S12IA/1SN5
S13IA/1SN6
S14IA/1SN7
S15IA/1SN8
A1B 19-015 HG
A15 19-163 LH
A57 19-089
A67 19-089
A68 19-089
A58 19-089
A07 19-015 HG
A05 19-163 LH
A19 19-036 HPD
A08 19-036 HPD
2/8S MDR00
17H5 MDR01
232-1 MDD00
2/6S MDR01
17H5 MDR11
233-1 MDD01
2/6S MDR10
17H5 MDE101
234-1 MDD10
2/6S MDR11
17H5 MDE111
235-1 MDD11
2/1S MDR12
17L5 MDR21
136-1 MDD20
2/1S MDR13
17H5 MDR31
137-1 MDD30
2/1S MDR14
17H5 MDR41
138-1 MDD40
2/1S MDR15
17H5 MDR51
139-1 MDD50
R14 470
R15 470
R13 470
R16 470
R7 470
R8 470
R6 470
R9 470
P5
P5ELI
1883

MEMORY DATA REGISTER LOW

NOTES 1. ALL APPARATUS THIS SHEET LOCATED ON CPU-C BOARD 35-524 MDI.

AREA D3-R3, GATES A15 & A05 WERE 19-153LS
9S1W | 3355 | 19-16-77 | A02

REVISIONS
AREA D3-R3, A15 & A05 WERE SPEC'D AS 19-036 HG DEVICES.
WS1 | 3151 | 4-19-77 | A01

NAME | TITLE | DATE
DRAFT
ENGR
DIR ENG

TITLE FUNCTIONAL SCHEMATIC
MODEL 7/16 MSALU-7/32
PROCESSOR
DRAWING NO 63047
SHEET 19-36
DWG NO 01-079 MDI REV A00

INTERDATA®

B-BUS SHIFTER & LATCH

NOTES
1. ALL APPARATUS ON THIS SHEET LOCATED ON CPU-C BOARD. 35-524 M01.

NAME	TITLE	DATE	TITLE FUNCTIONAL SCHEMATIC MODELS 7/16 H SALU & 7/32 PROCESSOR
	DRAFT		
	CHK		
	ENGR		
	DIR ENG		TASK NO. 03047
			DWG NO. 01-079 M01 D08
			SHEET OF 20-36

REVISIONS

AREA B9, FUNCTION TABLE WAS SHOWN THUS:

FUNCTION	M	S_0	S_1	S_2	S_3
LOAD	0	0	X	1	1
AND	0	1	0	1	1
OR	0	1	1	1	0
XOR	1	0	0	1	1
ADD	1	0	1	0	0
SUB	1	1	0	0	1
CMP	1	1	1	0	0
CA				0	0

2	MCR 2455	—	4-26-75	B01

FUNCTION	$\overline{RD}_0$	$\overline{RD}_1$	$\overline{RD}_2$	$\overline{RD}_3$	M	S_0	S_1	S_2	S_3
LOAD	0	0	0		1	1	1	1	1
AND	0	0	1		1	0	1	1	1
OR	0	1	0		1	1	1	0	1
XOR	0	1	1		1	1	0	0	1
ADD	1	0	0		0	1	0	0	1
SUB	1	0	1		0	0	1	1	0
CMD	1	1	0		0	1	0	0	1
CA	1	1	0		0	1	0	0	1

NOTES:
1. ALL APPARATUS THIS SHEET LOCATED ON CPU-C BOARD, 35-529 MOD.
2. A# CORRESPONDING TO IC IN POS. A119 THRU A122 INDICATES IC'S ARE INSTALLED WITH DOT 1 IN UPPER RIGHT HAND CORNER.

ALU LOW

FUNCTIONAL SCHEMATIC
MODEL 7/16 HSALU-7/32
PROCESSOR
030A47
01-07500101D08 21 3/6

INTERDATA

A129
19-040
4 STAGE CARRY/
LOOK AHEAD

A136
19-039
4 BIT ALU

A137
19-039
4 BIT ALU

A115 * 19-070 ARH
A116 * 19-070 ARL
A117 * 19-070 ARH
A118 * 19-070 ARL

A125 19-038
A126 19-038
A127 19-038
A128 19-038

A146 19-029 XMR

A164 19-016 HG

A106 19-016 HG
A108 19-016 HG
A107 19-015 HG
A50 19-015 HG
A51 19-015 HG
A109 19-015 HG

R63 330

ALU HIGH

NOTES
ALL APPARATUS THIS SHEET LOCATED ON CPU-C BOARD, 33-524 MOI.
[illegible] LOADING 28 IC IN POS. A115 THRU [illegible] INSTALLED WITH [illegible] HAND CORNER.

NAME	TITLE	DATE
	DRAFT	
	CHK	
	ENGR	
	DIR ENG	

TITLE FUNCTIONAL SCHEMATIC
MODEL 7/16 NSALU-7/32
PROCESSOR
TASK NO. 03047
DWG NO. 01-079 [illegible] 828
SHEET OF 22-36

INTERDATA®

TRUTH TABLE 19-048

S1	S0	FUNCTION
0	0	NO OP
0	1	SHIFT RIGHT
1	0	SHIFT LEFT
1	1	LOAD

A98 SRH 19-048 SHIFT REG.

A102 SRH 19-048 SHIFT REG.

A83 SRL 19-048 SHIFT REG.

A87 SRL 19-048 SHIFT REG.

SHIFT REGISTER

NOTES 1. ALL APPARATUS THIS SHEET LOCATED ON CPU-C BOARD 95-524 M01.

REVISIONS
AREA 36, A37 P/N XMS/7-0892
E392 / 2-13-75 / EDJ

NAME / DRAFT / CHK / ENGR / DIR ENG / DATE

TITLE FUNCTIONAL SCHEMATIC
MODEL 7/16 HSALU - 7/32 PROCESSOR
TASK NO. 03067
DWG NO 01-079A01 REV 08
SHEET OF 23-36

INTERDATA

SRLIS1 2329, E3
SRLI40 53
SRLI41 2329, N3
SRLI60 2359, 53
SRL161 2354, E3
ED0091 36G1
SMPY10 36N1
FMPY1 36J1
ENDA1 204-1
RD231 123-1
RD221 223-1
ENDD1 124-0
EASB01 1145
ED1119 2361
ED1120 2361
DSB01 1149
SMPY11 23, 36N4
JARLA1 49
JARLB1 89
NULL0 79
RD151 121-1
1885
RD131 122-1
16N4
ED1411 222-1
16N4, 36E1
MDR011 1828
MDR001 1829
CA01 135-1
SRLI41 B3
SRLI51 A3
SRLI81 B3
SMPY11 G3
SRLI40 A3
SRLI50 A9
SRLI60 B3

A37 19-015 HG
A24 19-024 HA
A14 19-016 HG
A04 19-043 HGO
A16 19-101 HB
A39 19-015 HG
A48 19-016 HG
A46 19-018 HG
A38 19-026
A26 19-017 HG
A27 19-024 HA
A25 19-024 HA
A17 19-093 HGO
A06 19-093 HGO
A23 19-119

R5 470 P5
R11 470 P5
R12 470 P5
R1 470 P5

SRLI50 3B
JTARLA1 G3
JTARLB1 H3
LARH0 218I, 228I
LARL0 216I, 226I
LARX0 218I, 228I, 22H1
LLOC0 17C1
LMDR1 189I, 199I
UNDR0 16F4
N.C. N.C.
SELB1 214I, 224I, 36H1
DS201 123-0
226-0
SELA1 214I, 224I
NULL0 H3
SGNM0R0 214I, 226I
SGN20 21N4
SGN10 21N4
GMDR000 228I
MD320 36F1

N.C. N.C. N.C.

SECOND SOURCE DECODING

NOTES
1. ALL APPARATUS THIS SHEET LOCATED ON CPU-C BOARD, 35-524 M01.

THIS INFORMATION IS PROPRIETARY AND IS SUPPLIED BY INTERDATA FOR THE SOLE PURPOSE OF MAINTAINING AND USING INTERDATA SUPPLIED EQUIPMENT AND SHALL NOT BE USED FOR ANY OTHER PURPOSE UNLESS SPECIFICALLY AUTHORIZED.

NAME	TITLE	DATE
CHRIS JENSEN	DRAFT	29OCT73
	CHK	
	ENGR	
	DIR ENG	

TITLE FUNCTIONAL SCHEMATIC MODELS 7/16 HSALU & 7/32 PROCESSOR

TASK NO. 03047
DWG NO. 01-079 M01 D08
SHEET OF 29-36

INTERDATA

I/O TIMING & CONTROL

NOTES: 1) ALL APPARATUS ON THIS SHEET LOCATED ON 02-A BRD 36-522 M01
2) REMOVE THIS JUMPER TO INCREASE MIN. WIDTH OF RSO & TACKO.

REVISIONS
AREA AB-DB ADDED
RESISTORS R96-R100.
FOR PREVIOUS REV LEVELS
SEE VOID SHT 25 IN FILES
GS 3314 12-1-77 R07

	NAME	DATE
DRAFT		
CHK		
ENGR		
DIR ENG		

TITLE FUNCT. SCHEMATIC
MODEL 7/16 HSALU 7/32
PROCESSOR
ASY NO 03047
DWG NO 01-079M01 R07 SHEET 25 OF 36

INTERDATA®

DISPLAY CONTROL LOGIC

NOTES
1. ALL APPARATUS THIS SHEET LOCATED ON CPU-A BOARD 35-522 M01.

REVISIONS				
CHANGED TO REFLECT 35-522(M01) FOR PREVIOUS CONFIGURATION OF 35-522(M04) SEE 01-079(M04) D08				
4. STRAP FROM A62 PIN E TO PIN 6 WAS NOT SPEC'D.	2573	—	7-19-75	PDH
AREA B5, A71 WAS NOT SPEC'D. AREA N6, M4690, A04 TO A05 WAS DELETED.	2392	/	2-13-75	PDH
AREA N8: AG4-03 WAS TO A50-12. ADDED A02.	2376	—	1-23-75	PDH
AREA N5 DELETED JUMPER FROM A62 PIN 2 TO A62 PIN 1	2290	—	10-1-74	PDH

NAME	TITLE	DATE
	DRAFT	
	CHK	
	ENGR	
	DIR ENG	

TITLE: FUNCTIONAL SCHEMATIC MODEL 7/16 MSALU 7/32 PROFESSOR
TASK NO. 03047
DWG NO. 01-079 M01 R04 D08
SHEET 26 OF 36

INTERDATA®

XRPIB

SCLROB

SLMBY1

POSITION

WIDTH

MEMORY TIMING & CONTROL

NOTES
UNLESS OTHERWISE SPECIFIED, ALL APPARATUS
IN SHEET LOCATED ON CPU-A BOARD 35-522 M01

REVISIONS			
AREAS B7, C8, D7 & E7 CHANGED: A168 GATES WERE 19-101B, AREA 3F, A122-10 WENT TO A95-06	AREA A8: A69 (19-016) & RELATED CIRCUITRY WAS NOT SPEC'D	"ADDED" AREA M1: FRDO, REMOVED: CONN. BETWEEN A135-13 TO A194-15	AREA G8: A99-04 WAS FPSEL1. AREA L7: ADDED A123. AREA S7: C6 WAS 47PF. R26 WAS 10K, AREA J8: ADD CROSS REFERENCE 25N7 TO FAENO.
		AREA G8: R26 WAS SHOWN AS A 6.8K FIXED RES.	AREA H3: A85 WAS A97, 19-115. A97-09 WAS FLBY0. A97-10 WAS TO A97-11. A97-08 WAS RSET1. AREA E1: ADDED A169. AREA K8: A145-01 WAS SCLROB
M ADM 3031 — 3-24-77 R07	M JS 2647 — 2-23-76 R06	M ADM 2946 — 3-21-75 R05	M ADM 2509 — 5-19-75 R04
AREA E7: A170 WAS A158, 19-016. A158-09 WAS FERO. A158-10 WAS FXPBY0. A158-08 WAS TO A98-03.			
M ADM 2376 — 1-27-75 R02			
AREA I11: MNEM1 PBY0 WAS NOT SPEC'D. KEEP E1, MNEM1 GL1T00 WAS FIXED. AREA S1, MNEM1 GPBY0 & PBY0 WERE NOT SPEC'D.			
M ADM 2552 — 6-24-75 R05			

NAME	
DRAFT	
CHK	
ENGR	
DIR ENG	
TITLE FUNC. SCHEMATIC	
MODEL 7/16 HSALU 7/32	
PROCESSOR	
TASK NO. 03047	SHEET OF 27-36
DWG NO. 01-079M01R07008	

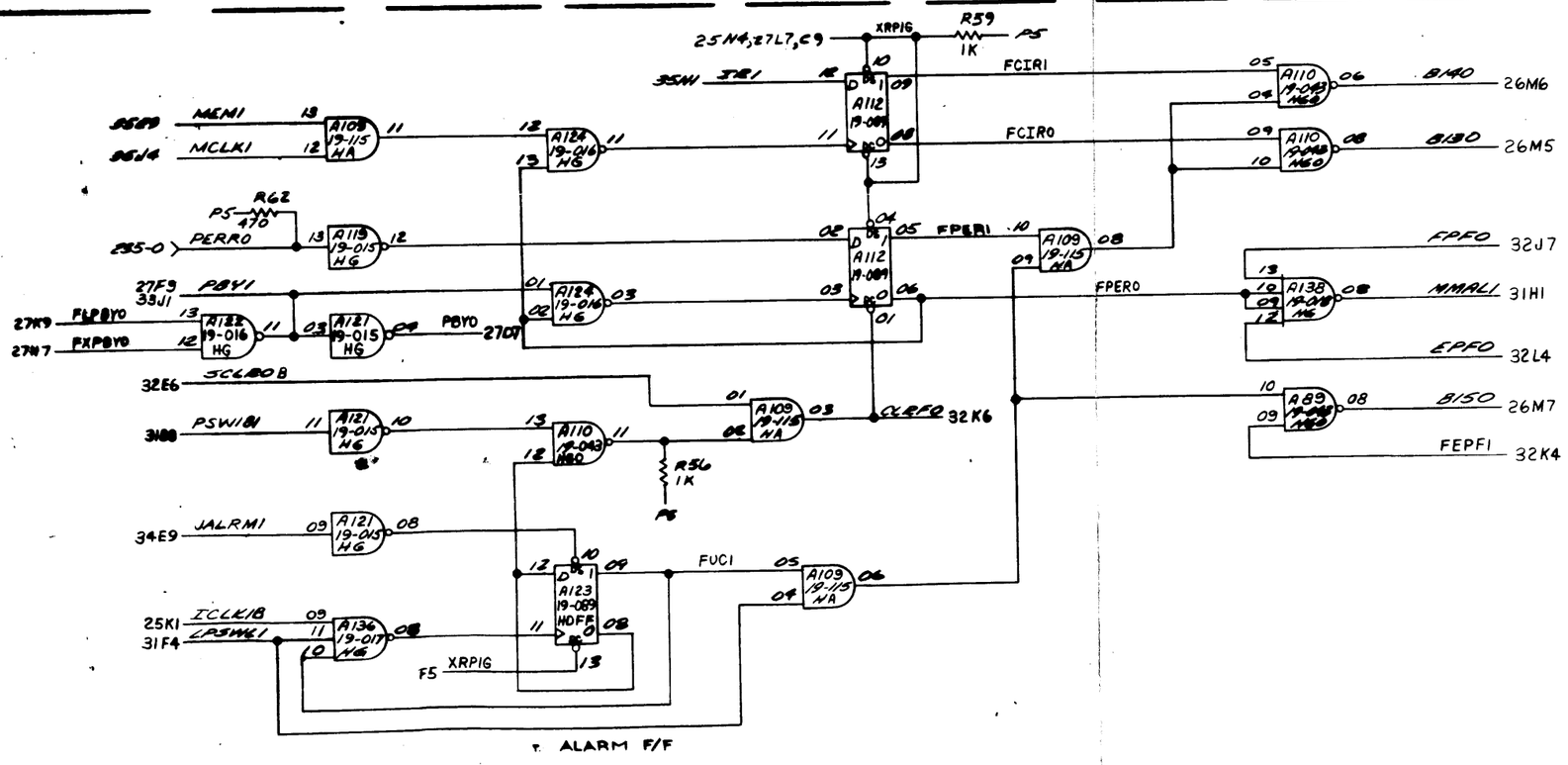
INTERDATA

7/32 PARITY OPTION BOARD

35-568 (SEE NOTE 2)

FOR SCHEMATIC INFO. SEE 02-368 C08

REVISIONS					
CHANGED TO REFLECT 35-522 M01; FOR PREVIOUS CONFIGURATION OF 35-522(M00) SEE 01-079(M00)D08					
1	JRH	2290	—	10-1-74	R01
AREA N14, "REFER TO 02-368 ETC." WAS NOT SPEC'D. AREA D-9. LAST SENTENCE OF NOTE 2 WAS NOT SPEC'D.					
2	JRH	2365	—	12-17-74	R02
AREA F5; ADDED CROSS REF; 27L2 TO XRPIG.					
F5	JRH	2376	—	1-28-75	R03
AREA C7: DELETED "PBY0" TO 109-2					
JM	HM	2552	—	6-27-75	R04
AREA F5, XRPIG ADDED 25N4					
WS	WJ	2733		3-5-76	R05
REVISED PER ECN 3002 FOR PREVIOUS REV INFO SEE VOIDED COPY 01-079/M01 R05 SHT 28.					
WS	WJ	3002	-	12-15-76	R06



MEMORY PARITY & ALARM REGISTER

NOTES
1. UNLESS OTHERWISE SPECIFIED ALL APPARATUS THIS SHEET LOCATED ON CPU-A BOARD 35-522 M01.

2. THIS OPTION, WHEN EQUIPPED, IS INSTALLED AT THE BACK PANEL ON SLOT 3. CONNECTOR 1.

NAME	TITLE	DATE
	DRAFT	4-23-74
	CHK	
	ENGR	
	DIR ENG	

TITLE: FUNCTIONAL SCHEMATIC MODEL 7/16 HSALU 7/32 PROCESSOR

TASK NO. 03047 DWG NO. 01-079M01 R06 D08 SHEET 28 OF 36

INTERDATA

PART OF
CPU-C BOARD
35-524 M01

PART OF
CPU-B BOARD
MOD 7/32 ——— 35-523
MOD 7/16 HSALU — 35-544

A MOD F/F

PART OF
CPU-A BOARD
35-522 M01

YD & YS LOGIC

NOTES
1. ALL APPARATUS THIS SHEET LOCATED ON CPU-A BOARD EXCEPT FOR AREAS DESIGNATED CPU-B & CPU-C BOARDS.

REVISIONS
CHANGED TO REFLECT 35-522 M01; FOR PREVIOUS CONFIGURATION OF 16-522(mod) SEE 01-079(mod) D08.
X | REV | 2290 | — | 10-1-74 | R01

NAME | TITLE | DATE
DRAFT
CHK
ENGR
DIR ENG

TITLE: FUNCTIONAL SCHEMATIC MODELS 7/16 HSALU & 7/32 PROCESSOR
TASK NO: 03047
DWG NO: 01-079 M01 R01 D08
SHEET OF: 29 - 36

INTERDATA®

FLAGS AND CONDITION CODES

A155 19-038

A167 19-038

A46 19-026 HG

A57 19-026 HG

A45 19-026 HG

A153 19-070 4 BIT COUNTER

INPUTS A B C D

OUTPUTS A B C D

CLR, EN P, EN T, LD, CLK, CARRY

NOTES
1. ALL APPARATUS THIS SHEET LOCATED ON CPU-A BOARD 35-522 M01.

ADDED: AT E9 TO "CLKO" LOC 25N7

REVISIONS
(MODIFIED) TO REFLECT 35-522 M01; FOR PREVIOUS CONFIGURATION OF 35-522(A00) SEE 01-079(A00) D08.

THIS INFORMATION IS PROPRIETARY AND IS SUPPLIED BY INTERDATA FOR THE SOLE PURPOSE OF MAINTAINING AND USING INTERDATA SUPPLIED EQUIPMENT AND SHALL NOT BE USED FOR ANY OTHER PURPOSE UNLESS SPECIFICALLY INDICATED.

NAME	TITLE	DATE
	DRAFT	
	CHK	
	ENGR	
	DIR ENG	

TITLE FUNCTIONAL SCHEMATIC MODEL 7/16 HSALU 7/32 PROCESSOR

TASK NO. 03047 SHEET OF

DWG NO. 01-079 M01 R02 D05 30-36

INTERDATA

REVISIONS

CHANGED TO REFLECT RE-ENGINEERING FOR PREVIOUS COUNTING OPERATION OF 35-582 (mag) SEE CH-37			
X	MCH 2890	—	10-1-74
ADDED A17, A18, A19, & A20 IC.			
2X	MCH 2892	—	12-13-74

BRANCH CIRCUITS AND PSW

NOTES
1. ALL APPARATUS THIS SHEET LOCATED ON CPU-A BOARD, 35-522 M01.
2. REMOVE STRAP FROM TP13 TO TP4 FOR POWER FAIL / AUTO RESTART OPTION

FUNC. SCHEMATIC
MODEL 7/16 HSALU 7/32
PROCESSOR
03047
01-079M01R02
SHEET OF 31 - 36

INTERDATA®

REVISIONS				
CHANGED TO REFLECT 35-522 M01; FOR PREVIOUS CONFIGURATION OF 35-522(M00) SEE 01-079(M00) DOB.				
7	MA	2290	—	10-4-74 R01
AREA L8, 22 THRU. DS-P/N 23-017 WAS NOT SPEC'D.				
2A	MEH	2434	—	3-17-75 R02
AREA E8: Q2-C WAS TO Q1-C & D5				
4S	VL	2650	-	2-27-76 R03
AREA N1 "FB0" ADDED 27A7				
7K	RK	2647		3-8-76 R04
AREA E8: Q2-C WAS TO D5				
4S	VL	2838	-	5-3-76 R05
AREA N3, ADDED 27F1 TO A84-03				
PM	VL	3031		3-28-77 R06

OSC1 36A5 L5
A125 19-075 HG
OSCO
R44 1K
XRPIF
F81 27B1
FA1 27A1, 36A5
FB0 27A7, 27A7
A77 19-063
A77 19-263
NC
A84 19-116 HA
CLK1B 202-1
CLK1C 203-1
CLK1A 25E1, 27F1, 27D7, 34A1, 35K1
CLK1 25H1, 30F1, 31C1, 36A5, 31H1, 27F1
R69 470
22-018 4-35PF
4.7K L1
C1
R67 2.2K
A81 19-016 HG
FA1FB00 237-0
A69 19-015 HG
FA1FB01
35A1
PSW1B1 31B9
A100 19-016 HG
EPF0 28L6
FEPF1 28L7
R94 1K
P/O SCLR RELAY A137
A126 19-136 HPO
SCLR0 126-0
R77 330
R43 XRPIH 1K
A88 19-089
FEPF1
FEPF0 N C
CLRF0 28K8
FA1FB01
DSTOP0
OSC1
A93 19-017 HG
DON0 132-0
33G9 C1
A76 19-044 HG0
R38 1K
STOP0
R49 10K
C7 2.2UF
D2
A87 19-092
XRPIH
SCLR0A 234-0
SCLR0L 26A3, 35B9
SCLR0B 27A1, 28B7
35A1, 33D9, 35J9, 36C1, 34C1, 34M1
A126 19-101 B
R42 22K
C6 15MF
EARLY POWER FAIL F/F
1MS TIMER
A87 19-042
SCLR0B
FPF1 31K9
FPF0 28L5
POWER FAIL F/F
P/O DISPLAY
P0FF0 105-3
P0FF0
D1 23-001
PFDT0
INIT0 101-3
R57 22
R58 2.2K
C8 1000 pf
A120 19-046 HPO
R73 470
D4
POWDN0 128-0
25A1 PDN1
R90 47K
R91
R92
34D9 FP0W1
R72 4.7K
C13 33MF
C14 .5MF
Q1 20-001
Q2 20-001
N.C. N.C. N.C. N.C. N.C.
PRR0 P20 PFDT0
236-1 126-1 134-0 130-0 218-0 202-0
R5 47
C3 15MF
R7 10K
Q1 20-002
Q2 20-002
D1 6.2V
R2 820
128-0 P0WDN0
R12 1K
C1 15MF
R1 820
R8 220
R9 47 (AVAIL.)
102-0
D5 D6 D7 D8 23-017
Q3 20-002
Q4 20-001
R3 470
R6 4.7K
R11 100
R10 00
R4 2.2K
C2 15MF
202-0 PFDT0
PRIMARY POWER FAIL 35-448 (NOTE 2)
ULTIMATE RELAY

NOTES
1. UNLESS OTHERWISE SPECIFIED ALL APPARATUS THIS SHEET LOCATED ON CPU 'A' BOARD, 35-522 M01.
2. THIS OPTION, WHEN EQUIPPED, IS INSTALLED AT THE BACKPANEL ON SLOT 7, CONNECTOR 0.

DRAWN
CHK
ENGR
DIR ENG

TITLE: FINAL SCHEMATIC MODEL 7/6 HSALU 7/32 PROCESSOR
TASK NO. 03447
DWG NO. 01-079M01 R06 D08
SHEET OF 32 36

44-231 16042

INTERDATA®

REVISIONS				
CHANGED TO REFLECT 35-522M01; FOR PREVIOUS CONFIGURATION OF 35-522(M00) SEE 01-079(M00)D08.				
X	DEN	2290	—	10-1-74	R01

CLOCK STOPS.

NOTES
1. ALL APPARATUS THIS SHEET LOCATED ON CPU-A BOARD, 35-522 M01.

NAME	TITLE	DATE	TITLE FUNC. SCHEMATIC
	DRAFT		MODEL 7/16 H5ALU 7/37
	CHK		PROCESSOR
	ENGR		TASK NO. J3047
	DIR ENG		SHEET OF 33-36
			DWG NO. 01-079M01R01D08

BRUNING 44-231 16042

INTERDATA

REVISIONS			
CHANGED TO REFLECT 35-522 M01 FOR PREVIOUS CONFIGURATION OF 35-522 (M00) SEE 01-079(M00)R08.			
1	JEH	2290	—	10-1-74	R01
AREA F7: ADDED CONNECTION TO GND TO PAGE 5-11					
WS	1	2991	—	9-9-76	R02

COMMAND LOGIC & COUNTER

NOTES
1. ALL APPARATUS THIS SHEET LOCATED ON CPU-'A' BOARD, 35-522 M01.

FUNC. SCHEMATIC
MODEL 7/16 HSALU 7/32 PROCESSOR
03047
01-079M01R02D08
SHEET 34 OF 36

INTERDATA

FIELD CONTROL LOGIC

NOTES
1. ALL APPARATUS THIS SHEET LOCATED ON CPU-A BOARD, 35-522M01.

REVISIONS			
CHANGED TO REFLECT 35-582 M01; FOR PREVIOUS CONFIGURATION OF 35-582M00 SEE 01-079(M00)D08			
#	ECM 2290	—	10-4-74 R01

NAME	TITLE	DATE
	DRAFT	
	CHK	
	ENGR	
	DIR ENG	

TITLE FUNC. SCHEMATIC
MODEL 7/16 HSALU 7/32
PROCESSOR
TASK NO. 03047
DWG NO. 01-079M01 R01 D08
SHEET OF 35-36

INTERDATA

REVISIONS				
CHANGED TO REFLECT 35-522M01; FOR PREVIOUS CONFIGURATION OF 35-522(M00) SEE 01-07				
7	SH	2290	—	10-1-74 R01
AREA K6, A99 PAN WAS 19-089. AREA L3, A01 PINS 8,9, 10 WAS NOT SPEC'D. AREA N3, DELETED CONN. FROM A13-2 TO A12-9.				
8	DEN	2392	—	12-19-75 R02
AREA E8: DELETED C82, 1MF				
8	RC	2619	—	9-23-75 R03

PART OF CPU-A BOARD 35-522M01

SHIFT REGISTER CONTROL

MULTIPLY LOGIC

PART OF CPU-C BOARD 35-524

DECOUPLING CAPACITORS

P5 — .1MF — C26 THRU C81, C83 THRU C97 — .1MF

P5 — 33MF — C16 THRU C25 — 33MF

DECOUPLING CAPACITORS

P5 — .1MF — C11 THRU C56 — .1MF

P5 — 33MF — C1 THRU C10 — 33MF

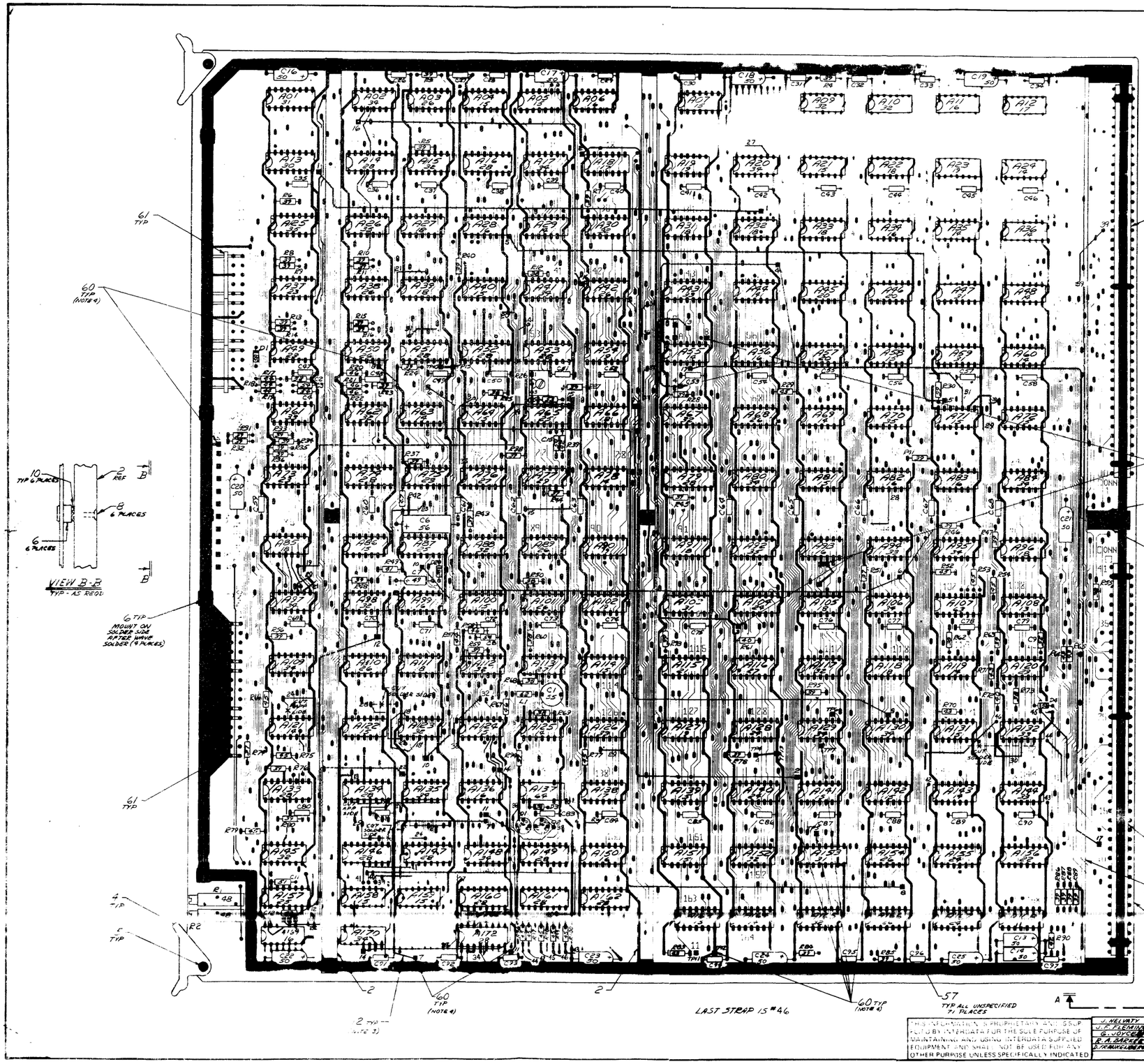
NOTES
1. ALL APPARATUS THIS SHEET LOCATED ON CPU-A BOARD, 35-522 M01 EXCEPT FOR AREA DESIGNATED CPU-C BOARD 35-524 M01.

	TITLE	DATE
NAME	DRAFT	
	CHK	
	ENGR	
	DIR ENG	

TITLE FUNCTIONAL SCHEMATIC MODEL 7/16 HSALU 7/32 PROCESSOR

SIZE D | 03047 | SHEET 36 OF 30

DWG 01-079M01 REV R03



PARTIAL VIEW A-A

TYP - 3 PLACES

NOTES:
1. HEADER STIFFENER (ITEM 9) TO BE SOLDERED TO GND BUS AT 2 ENDS AND CENTER (APPX SIZE).
2. CONTACTS CLOSEST TO EDGE OF BOARD TO BE BENT INWARD PRIOR TO SOLDERING.
3. AT STRAP ITEM 12 IS NO. 18 ADDITIONAL MW OPTIONS AND TEST POINTS (ITEM 13) ARE A TOTAL: A-C, E-F, TP10-TP17, J-P, R-T, MW STRAPS ARE ETC. ALL TEST POINTS TABLE ALL STRAP LOCATIONS I THRU 17 AND OPTIONS A THRU P.

COMPONENT	REF DESIGNATION
CAPACITOR	C1-C9, C11-C81, C83-C97
DIODE	D1-D4
INDUCTOR	L1
RESISTOR	R1-R80, R82-R90, R93-R100
TRANSISTOR	Q1, Q2

REVISIONS

REVISED AT PARTITION STRAP, CHANGED PIN NUMBERS TO AGREE WITH REVISED WIRE LIST, ETC. PARTIAL AS NECESSARY (TRACE, ETC.) ITEM 16 (CABLE); ADDED ITEM 17 (CLIP), ITEM 18 (STRAP), ADDED NOTE 4, CHANGED NOTE 3 TO ITEM 18 & ADDED ITEM 19 (STRAP) FROM ECO.

RELEASED FOR PRODUCTION
MFG ENG ___ DATE ___

SET COMPAT UNDER REF. BETWEEN PINS 9 & 10. CUT BETWEEN HOLES H62 & H63 NOT SPACED AS SHOWN. REVISED SHAPE TO AGREE WITH ECO 999-202 COS.

REVISED CIRCUITRY TO REFLECT NEW COPPER. ADDED 5 TPS (BLANK) STRAPS 42-46 WERE NOT SPEC'D.

A/16 WAS ITEM NO. 33

REVISED CIRCUITRY TO REFLECT NEW COPPER. A73 ADDED 5 TPS (BLANK) STRAPS 42-46 WERE NOT SPEC'D.

J. HELVATY	DRAFT	11-7-74	7/32 AND 7/16 HSALU	
J. R. FLOWERS	CHK	11-8-74	CPU-A	
G. JOYCE	ENGR	11-8-74		
P. A. MEEKER	QC	11-8-74		
D. RAUWELEERS	MGR	11-8-74	CAD/DE	1/1

SIZE B | SIZE/DWG NO. 36003

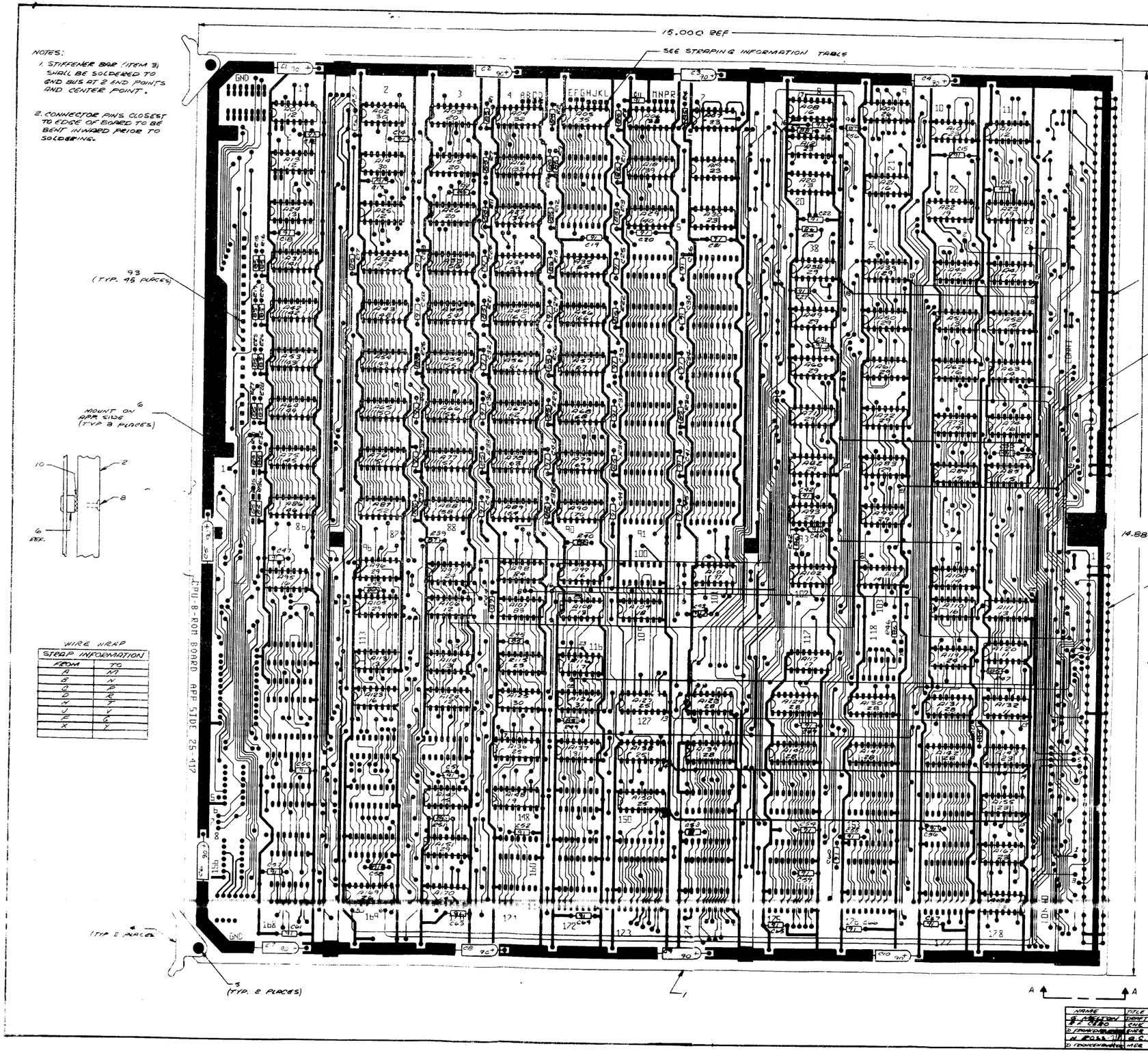
THIS INFORMATION IS PROPRIETARY AND SUPPLIED BY INTERDATA FOR THE SOLE PURPOSE OF MAINTAINING AND USING INTERDATA SUPPLIED EQUIPMENT AND SHALL NOT BE USED FOR ANY OTHER PURPOSE UNLESS SPECIFICALLY INDICATED

C-01

NOTES:

1. STIFFENER BAR (ITEM 3) SHALL BE SOLDERED TO GND BUS AT 2 END POINTS AND CENTER POINT.

2. CONNECTOR PINS CLOSEST TO EDGE OF BOARD TO BE BENT INWARD PRIOR TO SOLDERING.



WIRE WRAP STRAP INFORMATION

FROM	TO
A	M
B	N
C	P
D	R
H	T
J	V
E	G
X	Y

PWB-8-ROM BOARD APP SIDE 25-417

REVISIONS

PRE PRODUCTION APPROVAL	DEV PROD	INIT.	DATE

ADDED 1.77 JUMPER. 11 THRU 3 RE ROUTED. JUMPERS 8 & 19 LAST JUMPER WAS 10

RELEASED FOR PRODUCTION

MFG. ENG. [illegible] DATE 9-6-77

ADDED STRAPS A&J1 TO J2 PIN 3

ADDED 33 [illegible] -1.5-77 [illegible]

DELETED STRAP 126 (PIN 14 TO PIN 26). DELETED NOTE 3. ADDED STRAPS TO CONN P&T. DELETE TABLE REVISED CIRCUITRY TO REFLECT INITIAL CHANGES MADE WAS NEED TO ADD ITEM 16. FOR IMSD ROM INFO SEE MACHINE READABLE COPY 35-544 R03 E03

[illegible] 2997 -12-20-76 [illegible]

RESISTORS	R1-90, R92-97
CAPACITORS	C1-67
STRAPS	1-10, 13-22
I.C.	90-R16, R18-R87
	R89-R93, R98-R96
	R149-R97, R60-R68
	R71-R79, R82-R90
	R193-R99, R100-R111
	R113-R117, R119, R180
	R123-R132, R134-R143
	R147, R148, R150, R166
	R159, R161, R169, R170, R178

NAME	TITLE	DATE
E. PERSON	DESIGN	5-25-76
P. CHIPPO	CHK	5-30-76
D. FRANKENSTEIN	ENGR	8-5-76
H. BOWLING	Q.C.	11-1-76
D. FRANKENSTEIN	MFG	9-6-77

ASSEMBLY CPU-8 BOARD MOD 7/16 HSRLW

35-544 R04 E03 1-1

C-01

NOTES:
1. STIFFENER BAR (ITEM 3) SHALL BE SOLDERED TO GND BUS AT 2 END POINTS & CENTER POINT.
2. CONNECTOR PINS CLOSEST TO EDGE OF BOARD TO BE BENT INWARD PRIOR TO SOLDERING.

15.380 REF.

14.880 REF.

SEE NOTE 1

44

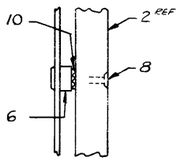
42
SEE NOTE 2

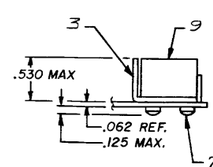
6
MOUNT ON SOLDER SIDE 2 PLACES

45 TYP 18 PLACES

SEE NOTE 1

ON ALL I.C. PACKS MARKED WITH AN ASTERISK (*), NOTE LOCATION OF I.C. ORIENTATION NOTCH, FACING NEARER, 14 PLACES.





PARTIAL VIEW A-A

43 TYP 26 PLACES

46 TYP 9 PLACES

42
SEE NOTE 2

3
SEE NOTE 1

5 TYP.

4 TYP.

1

A A

REVISIONS

RELEASED FOR PRODUCTION MFG. ENG. [illegible] DATE 4/4/75			
IN ADDR A1A9, 235 WIRE ITEM 3A.			
A	3335	12-11-76	KBN
IN ADDR A1A2, STRAPS 9-13/16 WRAP TO STRAP 10. REVISED DWG TO SHOW COPPER CHG. OF 22-083. ADD (3)CR10, (10) CR13, (5) CR14.			
B	3335	12-30-76	KBN
REVISED CIRCUITRY TO REFLECT NEW COPPER. STRAP 7 WAS NOT SPEC'D.			
C	2971	2-4-76	RDG
REVISED CIRCUITRY TO REFLECT NEW COPPER. AREA A9: DELETED C11. AREA A5 & A6 DELETED C23 & C28. ADDED R106 THRU R111 & C57 THRU C62.			
D	2971	9-23-76	RDG
REVISED CIRCUITRY TO REFLECT NEW COPPER. ADDED A1B & C63, 9 STRAP ADJ'S, A1A2, E23 & E40 TO E35, C44 TO C45 TO (E34) & 3 STRAP ADJ'S.			
E	3327	12-27-76	RDG
REVISED CIRCUITRY TO REFLECT NEW COPPER. AREA A4-A6, STRAP 10 WAS NOT SPEC'D.			
F	3335	9-13-77	RDG

COMPONENT	REF DESIGNATION
CAPACITOR	C1 THRU 10, C16 THRU
	C62, C65 THRU C83
RESISTORS	R1 THRU R112
I.C.	A1 THRU A144
JUMPERS	1 THRU 18

	NAME	TITLE	DATE
	V. CARUSO	DRAFT	9-9-76
	R. CERO	CHECK	9-10-76
	TERWILLIGER	ENG'R	9-24-76
	H. KOHN	M.C.	9-27-76
	TERWILLIGER	ENG'R MGR.	9-27-76

TITLE: ASSY. DRT. 4534 MOD 7/16 4534/0 & 7138 CPU - C BOARD

032047 SHEET 1 OF 1

35-32 REV F

C-01

INTERDATA

REVISIONS
CHANGES WERE MADE TO AREA N1-R2. FOR PREVIOUS REV. LEVELS SEE VOID SHT. 1 IN FILES. REVISED SHT 4.
GS 3350 12-19-77 R06

R69 100 C58 470PF

226-1 PHW0 — 54 19-036 HP — 123 19-015 HG — PHW1 — 2A9,3G9

P5 — R28 180 — P5 — R34 470 — 79 19-021 P — HW0 > 226-0

3F4,4A2,F2 MSCO — 85 19-036 HP — CLG1 > 2C9,4B9

2J7 AD1 — 85 19-036 HP — R48 1K — P5

3A2 SX1 — 30 19-036 HP — DLG0 — N3,2E9

3A2 DX0 10

2J5,3AG,M2 BSY0 02 — 64 19-021 P — R42 1K — P5

R2 DLG1

K3 DO81 — 136 19-036 HP — 135 19-016 HG — 134 19-015 HG — 5A1,K3 DA081 — 02 DA08OA — A AG081 — H3

115-0 DO80

K4 DO91 — 136 19-036 HP — 135 19-016 HG — 134 19-015 HG — 5A1,K4 DA091 — 04 DA09OA — B AG091 — H4

215-0 DO90

K5 DI01 — 136 19-036 HP — 135 19-016 HG — 134 19-015 HG — 3A5,5A1,K5 DA101 — 06 DA10OA — C AG101 — H5

116-0 DI00

K6 DI11 — 136 19-036 HP — 135 19-016 HG — 134 19-015 HG — 3A4,5A1,K6 DA111 — 08 DA11OA — D AG111 — H6

216-0 DI10

K6 DI21 — 113 19-036 HP — 124 19-016 HG — 134 19-015 HG — 3A3,5A3,K6 DA121 — 10 DA12OA — E AG121 — DI4 — J1

117-0 DI20

K7 DI31 — 113 19-036 HP — 124 19-016 HG — 134 19-015 HG — 5A3,K7 DA131 — 12 DA13OA — F AG131 — DI5 — M1

217-0 DI30

K8 DI41 — 113 19-036 HP — 124 19-016 HG — 123 19-015 HG — 5A3,K8 DA141 — 12 DA14OA — G AG141 — DI7

118-0 DI40

K9 DI51 — 113 19-036 HP — 124 19-016 HG — 123 19-015 HG — 5A3,K9 DA151 — 10 DA15OA — H AG151 — DI6

218-0 DI50

2M9,4G4,F3,F2 ATSYNO

133 19-001 G — 06 SGAD0 — 2J8

2J8,4F9 ADRS1A — DI8

DATA LINES 08-15

B9 ATSYN0 04 — 42 19-016 HG — ENDL1 — 96 19-015 HG

A2 MSCO 05 — F7

5F7 AAR121 — 96 19-015 HG — 122 19-016 HG — BUSY0 — F8

BSY1 05 — 122 19-016 HG

SRG1 04 — 98 19-002 G — BSY1

2L6 SRG0 05 — 64 19-021 P

G2 ENDL1 06

2R4 UAARH0 — 78 19-003 G

B9 ATSYN0 — 98 19-002 G

G7 BUSY0 03

2R4 UAARL0 04

6A2 PHW0A — 78 19-003 G — AAR1S1 — H9

PHW1 09 — 103 19-004 G — 2A9,3F3,F6 BSY1

3E9 WT1 4R2 — 122 19-016 HG — 31 19-016 HG

3A9 AAR1S0 10 — 122 19-016 HG — 78 19-003 G

78 19-003 G

NOTE 2

6A1,5,3 PD0B1 — 0 DATA INPUTS OUTPUT 7 — DO81 — A3

5F1 AAR001 — 1

5F5 AAR081 — 2 / 132 / 1

E4 AG081 — 3

STROBE 1

SELECT A

SELECT B

STROBE 2

6A2,5,4 PD091 — 0 DATA INPUTS OUTPUT 9 — DO91 — A4

5F1 AAR011 — 1

5F5 AAR091 — 2 / 2

E4 AG091 — 3 / 19-038

6A3,5,5 PD101 — 121 — 7 DI01 — A5

5F2 AAR021

5F6 AAR101

E5 AG101

19-038

6A4,5,6 PD111 — 9 DI11 — A5

5F2 AAR031

5F6 AAR111

E6 AG111

6A5,5,6 PD121 — 110 — 7 DI21 — AG

5F3 AAR041

E6 AG121

19-038

6A6,5,7 PD131 — 9 DI31 — A7

5F3 AAR051

5F7 AAR131

E7 AG131

6A7,5,8 PD141 — 99 — 7 DI41 — A7

5F4 AAR061

5F8 AAR141

E8 AG141

19-038

6A8,5,9 PD151 — 9 DI51 — A8

5F4 AAR071

3A8,5F8,H2 AAR151

AG151

C8

4 TO 1 MULTIPLEXOR

P5 R66 1K

2C2,6G2 DB0 01 — GR2 DA0 01 — C4

2C3,6G3 DB0 11 — GR3 DA0 91 — C4

2C4,6G4 DB0 21 — GR4 DB 1 01 — DA 1 01 — C5

2C5,6G5 DB0 31 — GR5 DB 1 11 — DA 1 11 — C6

2C6,6G6 DB0 91 — GR6 DB 1 21 — DA 1 21 — C6

2C6,6G7 DB0 51 — GR7 DB 1 31 — DA 1 31 — C7

2C7,6G8 DB0 61 — GR8 DB 1 41 — DA 1 41 — C8

2C8,6G9 DB0 71 — DB1 51 — DA 151

46 19-038 — 07

45 19-038 — 07

35 19-038 — 07

34 19-038 — 07

4 TO 1 MULTIPLEXOR

2S6,F7 SRG1 — 89 19-026 HG — 08 DLG1 — A3

4A4 ACK1

4F2 RENB1

C3 BSY0

2J6,4D6 SR1

2J5,4D7 DR1

09 — N.C.

DLG0 E2

48 19-036 HP — 47 19-015 HG — PD0B1 — H3

P5 R17 330 — PD080 < 115-1

48 19-036 HP — 47 19-015 HG — PD091 — H4

P5 R11 330 — PD090 < 215-1

48 19-036 HP — 36 19-015 HG — PD101 — H4

P5 R15 330 — PD100 < 116-1

48 19-036 HP — 36 19-015 HG — PD111 — H5

P5 R12 330 — PD110 < 216-1

37 19-036 HP — 36 19-015 HG — PD121 — H6

P5 R13 330 — PD120 < 117-1

37 19-036 HP — 36 19-015 HG — PD131 — H7

P5 R16 330 — PD130 < 217-1

37 19-036 HP — 36 19-015 HG — PD141 — H8

P5 R5 330 — PD140 < 118-1

37 19-036 HP — 36 19-015 HG — PD151 — H8

P5 R14 330 — PD150 < 218-1

PRIVATE DATA LINES 08-15

NOTES 1. PREFERRED ADDRESS X'F0'
ALL APPARATUS ON THIS SHEET IS LOCATED ON 35-391 MOZ SELECTOR CHANNEL.

SHEET INDEX	SHEET	1	2	3	4	5	6	7
	REV NO.	6	3	4	4	2	1	B

REVISION LEVEL OF SHEET 1 IS CONSIDERED TO BE THE REVISION LEVEL OF THIS DOCUMENT.

NAME	TITLE	DATE
W ZILLGER	DRAFT	10-5-71
R CERO	CHK	2-6-73
D.FRANKENBERGER	ENGR	2-7-73
N. MASI	SYS TST	2-7-73
R.E JONES	DIR ENG	

TITLE: NS SELECTOR CHANNEL

DWG NO. 03075 REV 08 — SHEET OF 1-7

ENG NO. 02-232

SIZE A — SCALE NONE — 44 231 15043

INTERDATA

GOR1

J5 DR1

111-0 D000 — H2 PD001
211-0 D010 — H3 PD011
112-0 D020 — H4 PD021
212-0 D030 — H5 PD031
113-0 D040 — HG PD041
213-0 D050 — H7 PD051
114-0 D060 — H8 PD061
214-0 D070 — H8 PD071

1K3 DB001 · 1K4 DB011 · 1K5 DB021 · 1K6 DB031 · 1K6 DB041 · 1K7 DB051 · 1K8 DB061 · 1K9 DB071

DA000 · DA010 · DA020 · DA030 · DA040 · DA050 · DA060 · DA070

1D2 CLG1

1J1 BSY1 · 1E2, F9 PHW1 · 3X6, 4H7 WTOA → UDBH0

DATA LINES
00-07

P5 — R31 330 — PD000 (111-1)
P5 — R32 330 — PD010 (211-1)
P5 — R19 330 — PD020 (112-1)
P5 — R18 330 — PD030 (212-1)
P5 — R20 330 — PD040 (113-1)
P5 — R22 330 — PD050 (213-1)
P5 — R23 330 — PD060 (114-1)
P5 — R21 330 — PD070 (214-1)

1E2 DLG0 · A9 PHW1

PDR000 6A2 · PD001 A2
PDR010 6A3 · PD011 A2
PDR020 6A4 · PD021 A3
PDR030 6A5 · PD031 A4
PDR040 6A6 · PD041 A5
PDR050 6A7 · PD051 A6
PDR060 6A8 · PD061 A7
PDR070 6A9 · PD071 A8

PRIVATE DATA LINES
00-07

P5 — R65 1K

93 19-003 G — 3A5 DARG1 · 4M9, 3K2 CLUS0

DS J 72 19-014 K DC (×2) — NC

LOAD/UNLOAD SEQUENCER

82 19-004 G — 1C

82 19-004 G — 1D

1 OF 10 DECODER

INPUTS A B C D · OUTPUTS 0 1 2 3 4 5 6 7 8 9

83 19-032

NC · NC · NC · NC

05 LARHO 5A4 — R55 470 — P5
06 LARLO 5A8 — R56 470 — P5
07 LFRHI 5D1 — R54 470 — P5
09 LFRLI 5D5 — R47 470 — P5
10 UAARHO 1F8 — R46 470 — P5
11 UAARLO 1F8 — R45 470 — P5

4D8 DA1 → 73 19-002 G → DAG0

DAG0 5J8

81, 1M2, 4D7 1C3 DR1 · BSY0 → 73 19-002 G → DRG0

4D8 CMD1 · NC → 23 19-002 G → CMG0

1M2, 4D6 SR1 → 71 19-003 G → SRG0

1F7

133 19-001 G

64 19-021 P → 10 SYN0 4F5
C21 470 PF

82 19-004 G → 06 SRG1 1M1

82 19-004 G → 08 CMG1 3A4

TPA

1A2 ADI

SCLR0A 3A4, 4L5

1E9 ADRSIA · 1G5 SGAD0 → 93 19-003 G

93 19-003 G → 85 19-036 HP

ADDRESS FF

D12 · D13

82 19-004 G → 10 SGAD1 3A4

1B9 ATSYN0

NOTES
ALL APPARATUS ON THIS SHEET IS LOCATED ON 35-391 MOZ SELECTOR CHANNEL.

AREA K5 DELETED 'SELO', TO A73-09
BY NM 2307 / 12-27-79 R03

AREA B1: ADDED, 93A.
AREA A8: A101-13, DLG1, WAS CONN TO DLG1 @ 1A3
BY RB 2132 / 4-16-79 R02
AREA N2, A78 WERE REVERSE

NAME	TITLE	DATE
W ZILLGER	DRAFT	10-5-71
	CHK	
	ENGR	
	DIR ENG	

TITLE NS SELECTOR CHANNEL
TASK NO. 03075
DWG NO. 02-232MOZ-D08
SHEET OF 2-7

INTERDATA

CIRCUIT NOTES

MAP FOR EDX (END OF DATA XFER)

MCH	WTO	WTI	
MCHO	000 SSX	100 SSX	(EVEN) AAR150
	001 RBA	101 SREQ,SSX,LDB	(ODD) AAR151
MCH1	011 CBSY	111 SREQ,LDB	
	010 CBSY	110 SREQ,LDB	(EVEN) AAR150

MAP FOR EMX (END MEM. XFER)

MCH	WTO	WTI
MCHO	SREQ,SSX,LDB	RBA
MCH1	SREQ,SSX,LDB	RBA

MAP FOR EBS (END BSY SET)

WTO	WTI
SREQ	SSX

BUFFER ACTIVE FF

BUSY FF

MUX/SELCH FF

WRITE FF

STATUS REQ FF

DATA XFER FF

ONE SHOT

19-032 1 OF 10 DECODER

BRANCH GATES

NOTES

ALL APPARATUS ON THIS SHEET IS LOCATED ON 35-391 MO2 SELECTOR CHANNEL.

REVISIONS

CHANGED: LEAD FROM 71-05 TO 71-N WAS FROM 71-05 TO 71-09	CMW	2197	6-19-74 R02
AERA J9, EBS1 NOT USED	WH	2307	12-27-74 R03
AERA F3: VALUE OF R27 WAS 470Ω, AERA H7: VALUE OF R10 WAS 10K	WSH	2824	5-26-76 R04

NAME	TITLE	DATE
W ZILLGER	DRAFT	
	CHK	
	ENGR	
	PTRNG	

TITLE: NS SELECTOR CHANNEL

TASK NO. 03075 | DWG NO. 02-232MM04D08 | SHEET 3-7

INTERDATA

REVISIONS				
LOC G6 ADDED R72, C61				
BG	DEN	2114	11/74	R01
REMOVED: CONN. BET. 17-08 @ L-13				
ADDED "CDRO" TO "L-X"				
GM	MH	2197	11/74	6-19-74 R02

P5 R49 180 R67 100 C57 470 PF 223-1 PATNO 01 76 19-004 G 02 01 88 19-021 P 03 ATNO 223-0
1A2, A5 MSCO 03 76 19-004 G 04 01 87 19-016 HG 03 E4 SINTO 10 54 19-036 HP 08 RENBI IM2
R30 1K P5 09 02 87 19-016 HG 04 11 109 19-004 G 10 06 88 19-021 P 04 RG1 1K P5 PTACKO 222-1
05 76 19-004 G 06 05 87 19-016 HG 06 08 88 19-021 P 10 RG0 1K P5 TACKO 222-0
03 109 19-004 G 04 05 109 19-004 G 06 09
C60 470 PF R71 100 122-0 RACKO 09 76 19-004 G 08 11 76 19-004 G 10 13 76 19-004 G 12
01 93 19-003 G 03 R64 680 P5 12 71 19-036 HP0 13 11 02 93 19-003 G 03
1M1 ACKI 09 87 19-016 HG 08 02 85 19-036 HP 03 01 96 19-015 HG 02 R57 1K P5 ATSYNO 1B9
3U3 SATNO 10 87 19-016 HG 11 SINTO DI 02 04 DS J 1 86 19-045 K DC 0 15 16 14 NC 03
12 87 19-016 HG 11 A2 MSCO 13 88 19-021 P 11

C56 100 PF 10 09 27 19-003 G 08 3A3 SCLRI 137-0 ACTO 05 17 19-057 SG 06 K
D7 CLO7I 01 77 19-055 SG 03 04 77 19-055 SG 06 03 96 19-015 HG 04 SELECT FF 05 96 19-015 HG 06
54 SELO 02 08 126-1 SCLRO 126-0 238-0 ENO R70 100 ENO H5 C59 100 PF S5 AEQI 01 02 03 19-003 G 03
NC 06 R40 10K P5 09 07 62 19-042 1 0 05 CDRI 3M7 P5R 10 NC 12 11
32 19-055 SG 10 08 09 32 19-055 SG 04 05 06 32 19-055 SG 02 01 03 17 19-057 SG 12 13
P5R 3F1 TP(I) 04 02 DS J 1 75 19-045 K DC 0 15 01 16 14 SEL1 03
H1 ENO 10 104-0 INHO 11 17 19-057 SG 10 09 17 19-057 SG 08 77 19-055 SG 09 10
DS J 1 86 19-045 K DC 0 09 07 11 06 12 10 NC 08 SCLROA 2M7 REQUEST FF
CDRO S2 2 1 L INH1 R72 100 C61 100 PF 04 27 19-003 G 06 NC 05 C55 100 PF
13 14 19-016 HG 11 01 15 19-036 HP 03 ENMDRI GR1 R3 470 P5 TP(K)
13 32 19-055 SG 11 TACO 237-0 12 01 97 19-036 HP 03 R58 1K P5 CDRO 6G9, K5
WTI WI, 3E9 12 30 19-036 11 WRTOA 110-1 13
04 A03 19-003 G 06 05 A41 19-044 HG0 01 02 R68 470 P5
01 14 19-016 HG 03 02 04 97 19-036 HP 06 05 TP(F) ENMDWI GR1 R62 470 P5
SELO H4 13 02 53 19-017 HG 12 01 09 97 19-036 HP 08 10 TP(J) ENMSI GA1 R59 470 P5
SREQO 3S4 12 97 19-036 HP 11 REQO 239-0 13 REQI H2 TP(G)
EMXI 3K5 NC 06 R29 10K P5 07 09 43 19-042 1 0 05 12 NC 11
5R5 SELIA 12 13 77 19-055 SG 11 10 2A9 WTOA 01 3H1 BACTI 02 42 19-016 HG 03
5G8 AAR15I 13 02 01 01 19-024 HA 12 13 16 19-015 HG 12
R36 1K P5 02 04 DS J 1 60 19-045 HF K DC 0 15 01 16 14 NC 03 3E8 ENG1
5R9 SELI 04 05 14 19-016 HG 06 5J8, 3K8 SINHO
3S7 LDBIA 07 09 DS J 1 60 19-045 HF K DC 0 11 06 12 10 08 ALDB1 3A8 NC
2K2 CLUSO

364 MSCI 3A7 123-1 PSYNO 03 123 19-015 HG 04 3A2 PSYNI R4 180 P5
05 79 19-021 P 04 SYNO 2S5 06 123-0
119-0 SRO 01 102 19-004 G 02 IM2, 2J6 SRI 08 79 19-021 P 10 PSRO 3H6 09 119-1 R44 1K P5
121-0 CLO7O 03 102 19-004 G 04 H4 CLO7I 12 79 19-021 P 11 PCLO7O 121-1 13 R43 1K P5
120-0 DRO 05 102 19-004 G 06 1M2, 2J5 DRI 01 90 19-021 P 03 PDRO 3H5 02 120-1 R51 1K P5
221-0 DAO 09 102 19-004 G 08 2J4 DAI 05 90 19-021 P 04 PDAO 3H5 06 221-1 R50 1K P5
220-0 CMDO 11 102 19-004 G 10 2J5 CMDI 08 90 19-021 P 10 PCMDO 220-1 09 R52 1K P5
09 123 19-015 HG 08 13 102 19-004 G 12 C34 100 PF C33 47 PF 12 90 19-021 P 11 PADRSO 219-1 13 R53 1K P5
219-0 ADRSO 05 123 19-015 HG 06 ADRSIA 1E9 1D2 CLG1

THIS INFORMATION IS PROPRIETARY AND IS SUPPLIED BY INTERDATA FOR THE SOLE PURPOSE OF MAINTAINING AND USING INTERDATA SUPPLIED EQUIPMENT AND SHALL NOT BE USED FOR ANY OTHER PURPOSE UNLESS SPECIFICALLY INDICATED

NOTES
ALL APPARATUS ON THIS SHEET IS LOCATED ON 35-391 MO2 SELECTOR CHANNEL.

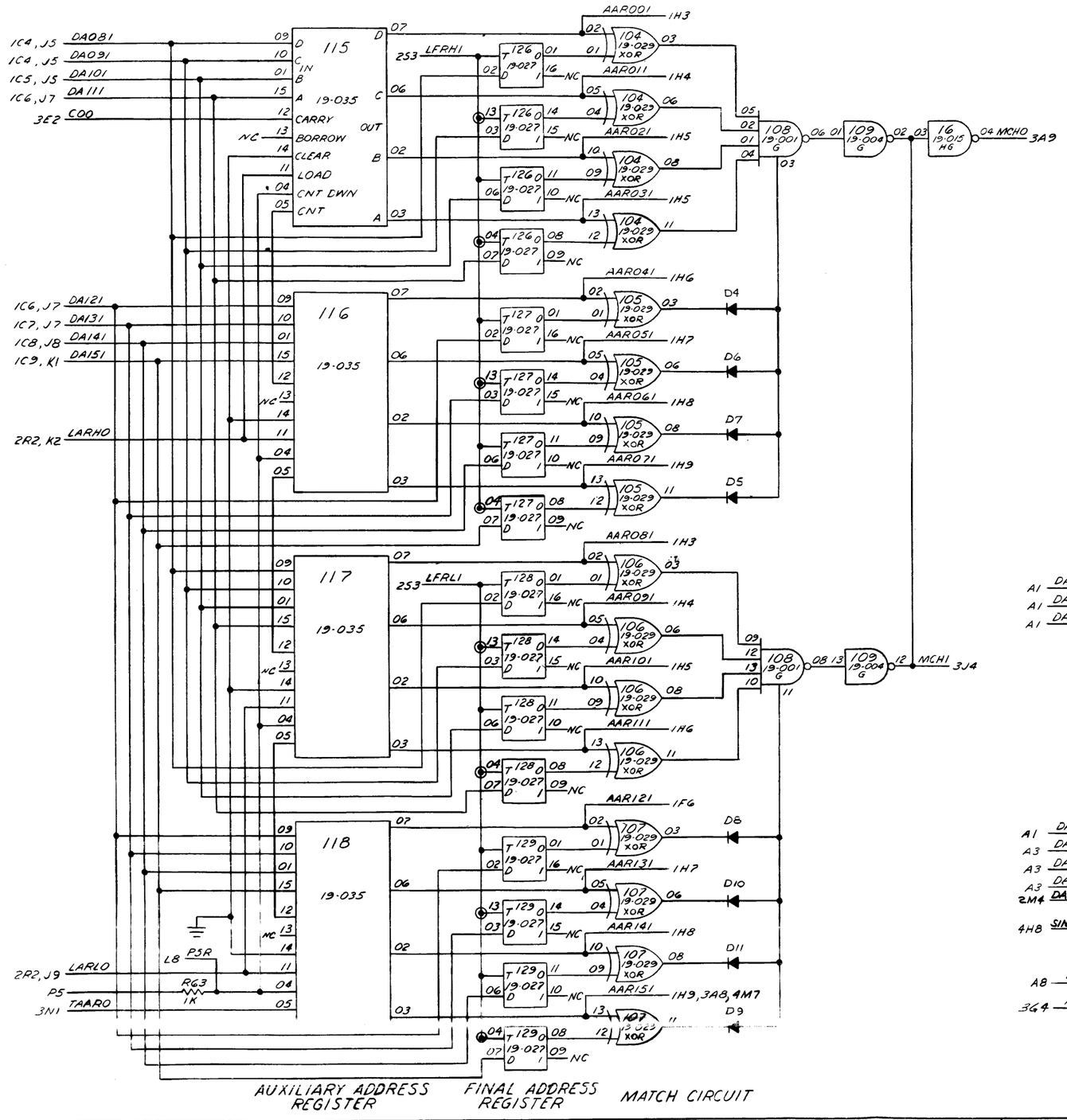
REVISIONS				
AREA A4, DELETED CORR. 2J5 FROM SELO				
VH	DEN	2307	1-20-75	R08
AREA E2: ADDED GATE 19-036 (ITEM54) & 1K RESISTOR.				

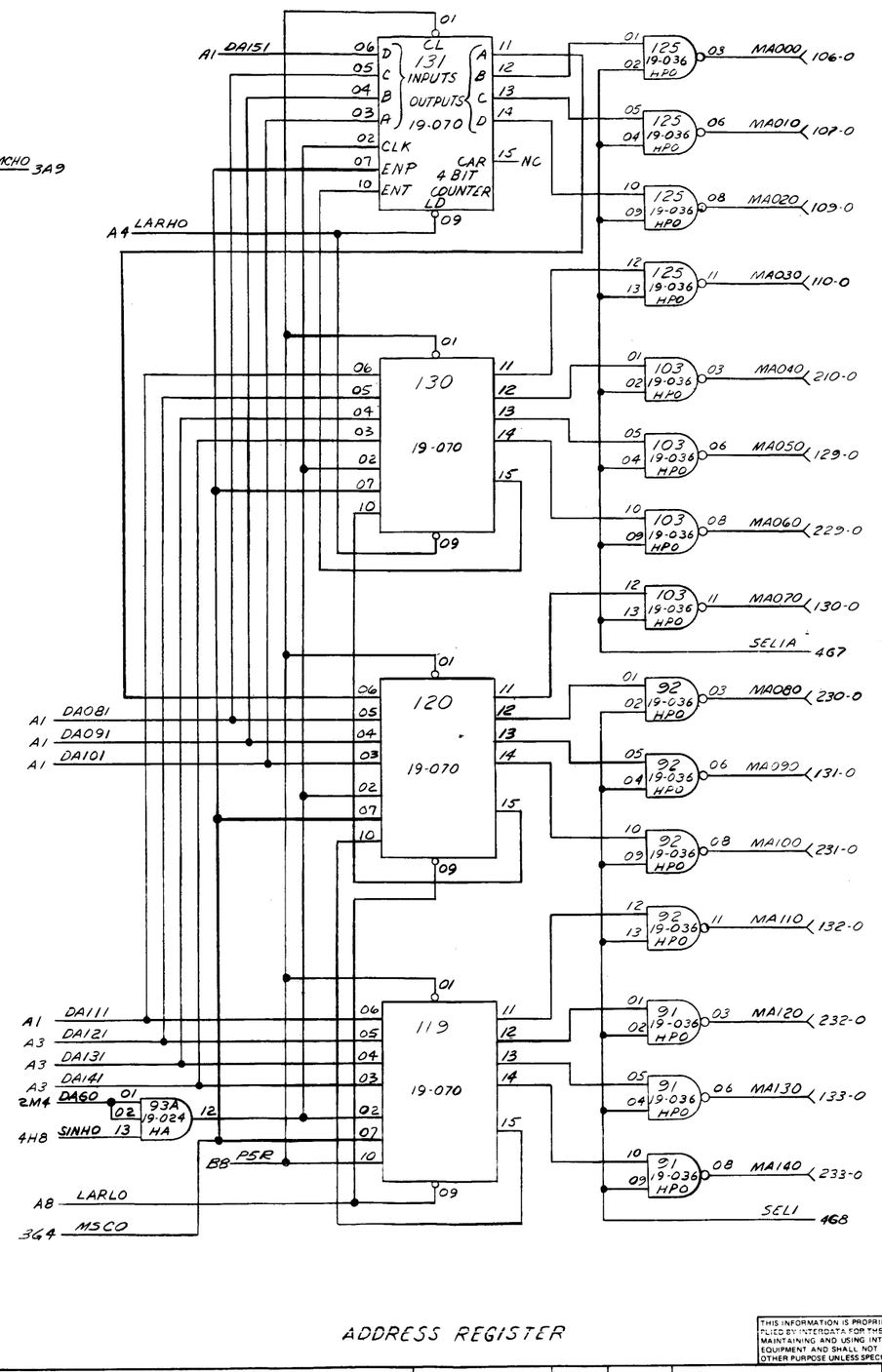
AREA E4: ADDED "SINTO" TO PIN 11 OF 19-016 (ITEM87)				
GS	JJ	3350	12-19-77	R04

NAME	TITLE	DATE
W. ZIELGER	DRAFT	10-5-71
	CHK	
	ENGR	
	DIR ENG	

TITLE: NS SELECTOR CHANNEL

DWG NO 03075 SPEC NO 02-232 MOV REV R08 SHEET 4-7





NOTES
ALL APPARATUS ON THIS SHEET IS LOCATED ON 35-391 MO2 SELECTOR CHANNEL.

REVISIONS
AREA MI, DESIG. W15, PIN 6 TO A, PIN 5 TO B, PIN 4 TO C, PIN 3 TO D, PIN 7 TO ENT, & PIN 10 TO ENP.
ECN 2307 | 12-27-74 | BO2

NAME	TITLE	DATE	TITLE
W. ZIELGER	DRAFT	10-5-71	SELECTOR CHANNEL
	CHK		
	ENGR		
	DIR ENG		TAB 03075 SHEET 5-7
			DWG NO 02-232MA01A02D08

INTERDATA®

DATA REG HIGH

DATA BUFFER

DATA REG LOW

DATA BUFFER

4R4,H1 ENMS1

210-1 MS000 — 58 19-015 HG — 01 — 6 19-016 HG — 03

1H3,H2 PD081 — 5 19-043 OC — 03 — 4 19-015 HG — 02

1H2 PHWOA

2H1 PDR000

109-1 MS010 — 58 19-015 HG — 04 — 6 19-016 HG — 06

1H4,H3 PD091 — 5 19-043 OC — 06 — 4 19-015 HG — 04

2H2 PDR010

209-1 MS020 — 58 19-015 HG — 06 — 6 19-016 HG — 08

1H4,H4 PD101 — 5 19-043 OC — 08 — 4 19-015 HG — 06

2H3 PDR020

108-1 MS030 — 58 19-015 HG — 08 — 6 19-016 HG — 11

1H5,H5 PD111 — 5 19-043 OC — 11 — 4 19-015 HG — 08

2H4 PDR030

208-1 MS040 — 58 19-015 HG — 10 — 13 19-016 HG — 03

1H6,H6 PD121 — 18 19-043 OC — 03 — 4 19-015 HG — 10

2H5 PDR040

107-1 MS050 — 58 19-015 HG — 12 — 19 19-016 HG — 06

1H7,H7 PD131 — 18 19-043 OC — 06 — 4 19-015 HG — 12

2H6 PDR050

207-1 MS060 — 59 19-015 HG — 02 — 19 19-016 HG — 08

1H7,H8 PD141 — 18 19-043 OC — 08 — 17 19-057 SG — 02

2H7 PDR060

106-1 MS070 — 59 19-015 HG — 04 — 19 19-016 HG — 11

1H8,H9 PD151 — 18 19-043 OC — 11 — 17 19-057 SG — 04

2H8 PDR070

LDRH0 — 54 19-036 HP — 03

3G8 LDRH1

R7 470Ω — P5

DS J 1 7 19-045 K 0 DC — 15 / 14 NC

DS J 1 8 19-045 K 0 DC — 15 / 14 NC

DS J 1 20 19-045 K 0 DC — 15 / 14 NC

DS J 1 21 19-045 K 0 DC — 15 / 14 NC

D 9 0 19-027 T 1 — 01 NC / 16

D 22 0 19-027 T 1 — 01 NC / 16

D 9 0 19-027 T 1 — 14 NC / 15

D 22 0 19-027 T 1 — 14 NC / 15

D 9 0 19-027 T 1 — 11 NC / 10

D 22 0 19-027 T 1 — 11 NC / 10

D 9 0 19-027 T 1 — 08 NC / 09

D 22 0 19-027 T 1 — 08 NC / 09

24 19-036 HP — 03 MD000 (229-1)

26 19-036 HP — 03 DB001 1K3

24 19-036 HP — 06 MD010 (130-1)

26 19-036 HP — 06 DB011 1K4

24 19-036 HP — 08 MD020 (230-1)

26 19-036 HP — 08 DB021 1K5

24 19-036 HP — 11 MD030 (131-1)

26 19-036 HP — 11 DB031 1K5

23 19-036 HP — 03 MD040 (231-1)

25 19-036 HP — 03 DB041 1K6

23 19-036 HP — 06 MD050 (132-1)

25 19-036 HP — 06 DB051 1K7

23 19-036 HP — 08 MD060 (232-1)

25 19-036 HP — 08 DB061 1K8

23 19-036 HP — 11 MD070 (133-1)

25 19-036 HP — 11 DB071 1K9

LDB1 R9,3S7

CDR0 R9,4R2

A1 ENMS1

206-1 MS080 — 59 19-015 HG — 05 — 57 19-016 HG — 02

A1 PD081 — 47 19-015 HG — 11 — 10

105-1 MS090 — 59 19-015 HG — 08 — 57 19-016 HG — 05

A2 PD091 — 47 19-015 HG — 13 — 12

205-1 MS100 — 59 19-015 HG — 11 — 57 19-016 HG — 10

A3 PD101 — 65 19-015 HG — 13 — 12

104-1 MS110 — 59 19-015 HG — 13 — 57 19-016 HG — 13

A4 PD111 — 65 19-015 HG — 11 — 10

204-1 MS120 — 70 19-015 HG — 01 — 68 19-016 HG — 02

A5 PD121 — 65 19-015 HG — 09 — 08

3A1 PDR120

103-1 MS130 — 70 19-015 HG — 03 — 68 19-016 HG — 05

A6 PD131 — 65 19-015 HG — 05 — 06

3A1 PDR130

203-1 MS140 — 70 19-015 HG — 05 — 68 19-016 HG — 10

A7 PD141 — 65 19-015 HG — 03 — 04

3A1 PDR140

102-1 MS150 — 70 19-015 HG — 09 — 68 19-016 HG — 13

A8 PD151 — 65 19-015 HG — 01 — 02

3A1 PDR150

3J9 LDRL0 — 54 19-036 HP — 05 — 06

R35 470Ω — P5

DS J 1 56 19-045 K 0 DC — 15 / 14 NC

DS J 1 55 19-045 K 0 DC — 15 / 14 NC

DS J 1 67 19-045 K 0 DC — 15 / 14 NC

DS J 1 66 19-045 K 0 DC — 15 / 14 NC

D 44 0 19-027 T 1 — 01 NC / 16

D 33 0 19-027 T 1 — 01 NC / 16

D 44 0 19-027 T 1 — 14 NC / 15

D 33 0 19-027 T 1 — 14 NC / 15

D 44 0 19-027 T 1 — 11 NC / 10

D 33 0 19-027 T 1 — 11 NC / 10

D 44 0 19-027 T 1 — 08 NC / 09

D 33 0 19-027 T 1 — 08 NC / 09

ENMDR1 4RG

ENMDW1 4R3

10 19-036 HP — 03 MD080 (233-1)

12 19-036 HP — 03 DB081 1K3

10 19-036 HP — 06 MD090 (134-1)

12 19-036 HP — 06 DB091 1K4

10 19-036 HP — 08 MD100 (234-1)

12 19-036 HP — 08 DB101 1K5

10 19-036 HP — 11 MD110 (135-1)

12 19-036 HP — 11 DB111 1K6

11 19-036 HP — 03 MD120 (235-1)

13 19-036 HP — 03 DB121 1K6

11 19-036 HP — 06 MD130 (136-1)

13 19-036 HP — 06 DB131 1K7

11 19-036 HP — 08 MD140 (236-1)

13 19-036 HP — 08 DB141 1K8

11 19-036 HP — 11 MD150 (137-1)

13 19-036 HP — 11 DB151 1K9

LDB1 G9

CDR0 G8

NOTES

ALL APPARATUS ON THIS SHEET IS LOCATED ON 35-391 M02 SELECTOR CHANNEL.

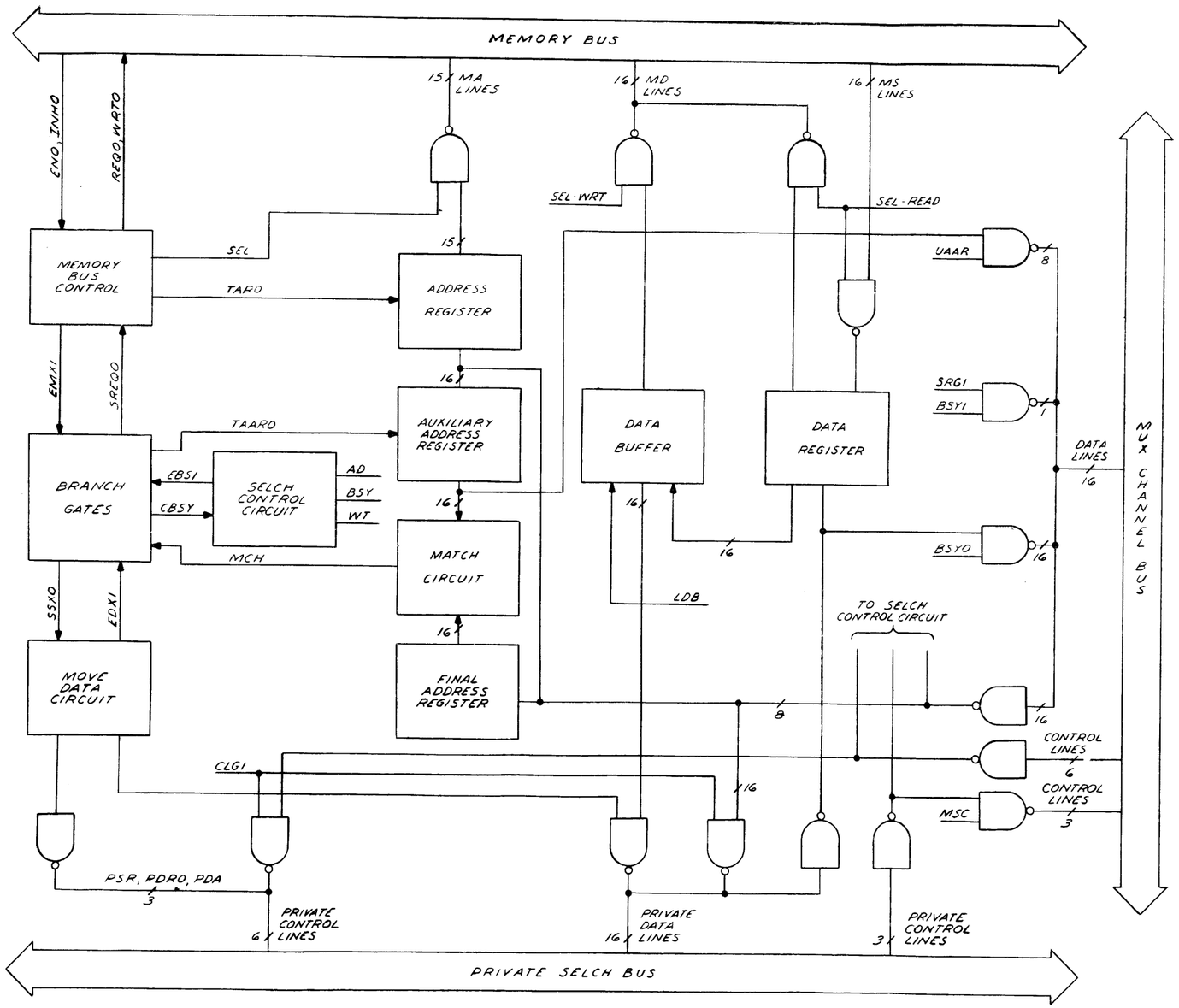
NAME	TITLE	DATE
W. ZILLGER	DRAFT-	10-5-71
	CHK	
	ENGR	
	DIR ENG	

TITLE: NS SELECTOR CHANNEL

TASK NO. 03075

DWG NO. 02-232M01R01D08

SHEET 6-7



BACK PANEL PIN INDEX

MNEM	LOC.	MNEM.	LOC.
ACTO	4H1	MDI30	6R6
ADRSO	4B8	MDI40	6R7
ATNO	4F1	MDI50	6R8
CLO7O	4B6	MSDOO	6A1
CMDO	4B3	MSD10	6A2
DOOO	2A1	MSD20	6A3
DO1O	2A2	MSD30	6A4
DO2O	2A3	MSD40	6A5
DO3O	2A4	MSD50	6A6
DO4O	2A5	MSD60	6A7
DO5O	2A6	MSD70	6A8
DO6O	2A7	MSD80	6H1
DO7O	2A8	MSD90	6H2
DO8O	1A3	MS100	6H3
DO9O	1A4	MS110	6H4
DI00	1A5	MS120	6H5
DI10	1A6	MS130	6H6
DI20	1A6	MS140	6H7
DI30	1A7	MS150	6H8
DI40	1A8	PADRSO	4F8
DI50	1A8	PATNO	4B1
DAO	4B7	PCLO7O	4F6
DRO	4B7	PCMDO	4F8
ENO	4H1	PDOOO	2F1
HWO	1D1	PDO10	2F2
INHO	4H5	PDO20	2F3
MAOOO	5R1	PDO30	2F4
MAO10	5R1	PDO40	2F5
MAO20	5R2	PDO50	2F6
MAO30	5R2	PDO60	2F7
MAO40	3R3	PDO70	2F8
MAO50	5R3	PDO80	1R4
MAO60	5R4	PDO90	1R4
MAO70	5R4	PD100	1R5
MAO80	5R5	PD110	1R6
MAO90	5R6	PD120	1R7
MA100	5R6	PD130	1R7
MA110	5R7	PD140	1R8
MA120	5R7	PD150	1R8
MA130	5R8	PDAO	4F7
MA140	5R8	PDRO	4F8
MDOOO	6G1	PHWO	1A1
MDO10	6G2	PSRO	4F6
MDO20	6G3	PSYNO	4B5
MDO30	6G4	PTACKO	4F2
MDO40	6G5	RACKO	4B3
MDO50	6G6	REQO	4R5
MDO60	6G7	SCLRO	4H4
MDO70	6G8	SRO	4B6
MDO80	6R1	SYNO	4F5
MDO90	6R2	TACKO	4H3
MD100	6R3	TACO	4R1
MD110	6R4	WRTOA	4R3
MD120	6R5		

BACK PANEL MAP

ROW 1	ROW 2	TERM NO.	CONN
P5	GND	41	
GND	GND	40	
		39	
		38	
MD150		37	
MD130	MD140	36	
MD110	MD120	35	
MD090	MD100	34	
MD070	MD080	33	
MD050	MD060	32	
MD030	MD040	31	
MD010	MD020	30	
	MDOOO	29	
		28	
		27	
SCLRO	PHWO	26	
		25	
		24	
PSYNO	PATNO	23	
	PTACKO	22	
PCLO7O	PDAO	21	/
PDRO	PCMDO	20	
PSRO	PADRSO	19	
PD140	PD150	18	
PD120	PD130	17	
PD100	PD110	16	
PD080	PD090	15	
PD060	PD070	14	
PD040	PD050	13	
PD020	PD030	12	
PDOOO	PD010	11	
WRTOA	MSDOO	10	
MS010	MS020	09	
MS030	MS040	08	
MS050	MS060	07	
MS070	MS080	06	
MS090	MS100	05	
MS110	MS120	04	
MS130	MS140	03	
MS150		02	
GND	GND	01	
P5	GND	00	
P5	GND	41	
GND	GND	40	
	REQO	39	
	ENO	38	
ATCO	TACO	37	
		36	
		35	
		34	
MA130	MA140	33	
MA110	MA120	32	
MA090	MA100	31	
MA070	MA080	30	
MA050	MA060	29	
		28	
		27	
SCLRO	HWO	26	
		25	
		24	
SYNO	ATNO	23	
RACKO	TACKO	22	
CLO7O	DAO	21	
DRO	CMDO	20	
SRO	ADRSO	19	
DI40	DI50	18	
DI20	DI30	17	
DI00	DI10	16	
DO80	DO90	15	
DO60	DO70	14	
DO40	DO50	13	
DO20	DO30	12	
DOOO	DO10	11	
MAO30	MAO40	10	
MAO20		09	
		08	
MAO10		07	
MAOOO		06	
		05	
INHO		04	
		03	
		02	
GND	GND	01	
P5	P5	00	

TEST POINT INDEX

DESIG	MNEM	LOC.
A	ADI	2L6
B	WTI	3G5
C	SXI	3G6
D	MSCO	3G4
E	BSYO	3G3
F	ENMDINI	4S3
G	REQI	4S5
H	DXI	3G7
I	SELI	4L3
J	ENMSI	4S4
K	ENMDRI	4S6

SUPPLEMENTARY INFORMATION

PRINTED CIRCUIT BOARDS AGREEING WITH THIS SCHEMATIC MUST BE AT LEAST THE FOLLOWING REVISION

NS SELCH	35-391M02 R03

NOTES

REVISIONS

IN SUPP INFO 35-392 M02 WAS R00					
-	DEH	2114	4/4	3/14/74	R01
IN SUPP INFO 35-392 M02 WAS R01					

ADD'A M02 DR. REV OF 35-391 M02 WAS R02					
MM	WH	2197	WH	8/22/74	E03
PE	WH	2132	WH	11-16-74	E02

NAME	TITLE	DATE
W ZILLGER	DRAFT	10-13-71
	CHK	
	ENGR	
	DIR ENG	

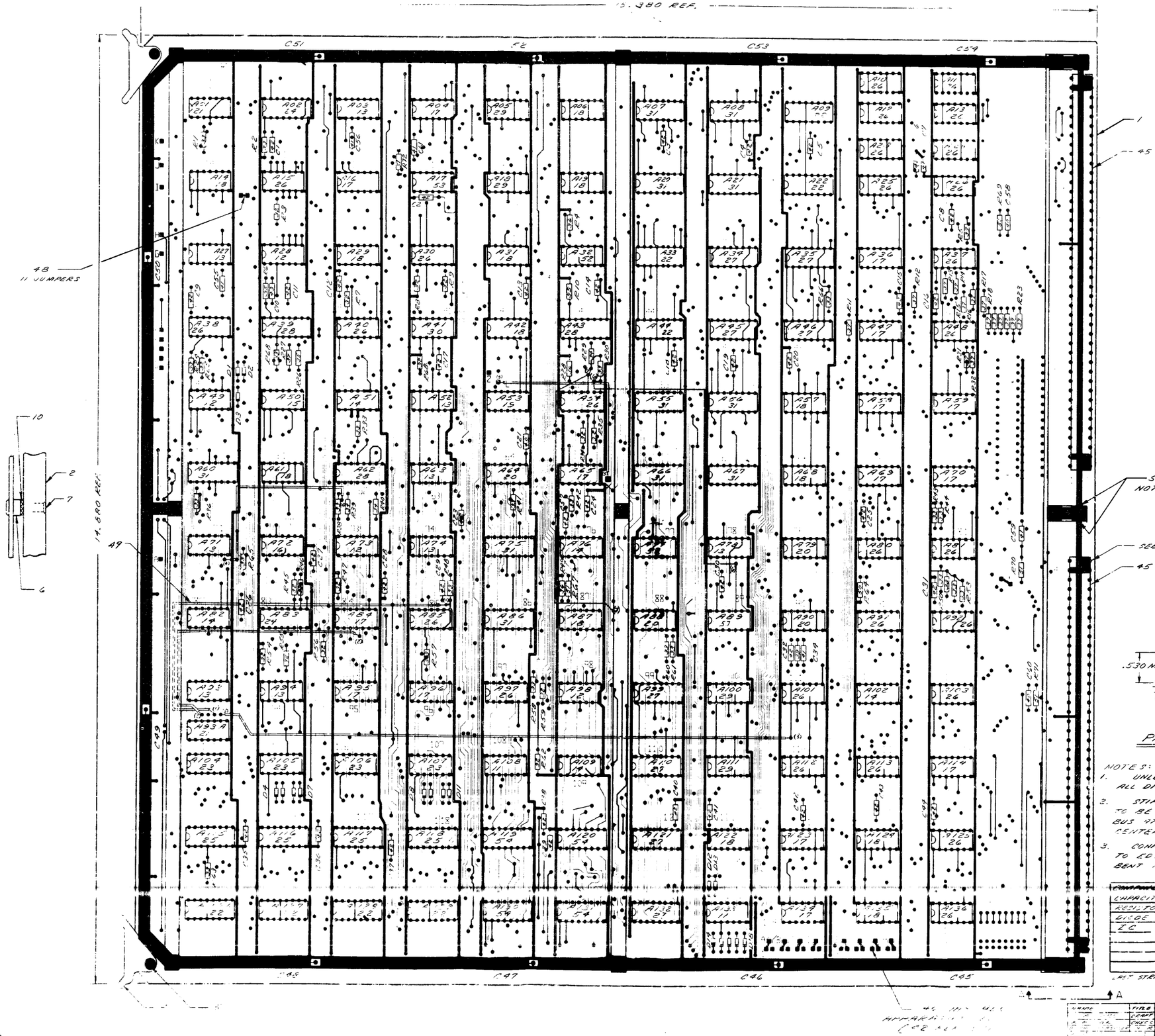
TITLE: NS SELECTOR CHANNEL

DWG NO. 03075

SHEET 7 OF 7

WORK NO. 02-232M01R03D08

DRAWING 44-231 1504.1



NOTES:

1. UNLESS OTHERWISE SPECIFIED ALL DIODES ARE ITEM 49.
2. STIFFENER U-SR (ITEM 3) TO BE SOLDERED TO GROUND BUS AT 2 END POINTS AND CENTER POINT.
3. CONNECTOR PINS CLOSEST TO EDGE OF BOARD TO BE BENT INWARD PRIOR TO SOLDERING.

CAPACITOR	C1-21 & 23-60
RESISTOR	R1-36, 38-62 & 67-71
DIODE	D1 THRU D18
I.C.	A01 THRU A136 & 93A

LAST STRAP NO. IS (7)

REVISIONS

ADDED R72 & C61
B 12/14 GM
MOVED STRAPS (253, REVISE TO REFLECT ADD JUMPER
C 2/8/77 VM
ADDED: STRAPS 4 & 5
D 2/17/77 VM
R10 WAS ITEM 34, A87 WAS ITEM 32
E 11/12/77 BOM
REVISED CIRCUITRY TO REFLECT 90 COMPER ADDED STRAPS 6&7 A89 WAS SPEC'D ITEM 13.
G SW 3350 12-12-77

	TITLE	SIZE	
	DRAFT 1/18/77		
	CHKD 2/2/77		
		E	03098
			SHEET 1

ASSEMBLY, ELECTOR
CHANNEL

INTERDATA

SHTS 1 & 2 CHG: IC No. A29, A25 & A28 WAS 19-015 HG. A26 WAS 19-016 HG. PIN 01 OF A27 WAS N.C, ADDED R26 TO A19 PIN 04 & 05

PM	DT	1483	VK	7/11/72	R01

CHANGED: SHT 1 LOC A1-A5 IC A30 & 31 WAS 19-016 HG
ADDED: SHT 1 LOC M9, C1 & R26

PM	DT	1507	VK	7/11/72	R02

CHG SHT 1 LOC J7 A02-16 HAS CONN TO A02-14 NON CONN TO A02-15

PM	VK	1646	VK	7/17/72	R03

DT: REMOVE: 2V7 & 10 FROM A17 PIN 12
ON SHTS 1 & 2 ER D3-G VIA: SYNIN
130 MA01 13 A28 08 MA30
120 MA11 01 A27 02 03
110 MA21 11 A28 10 13
100 MA31 13 A28 12 MA30
WHEN N7 = VIL, H3 PIN 5 & 6 BREAK K7 A19 PIN 5 HAS IF & EAGO

TC P	VK H	1904 1863	VK	9-21-73	R04

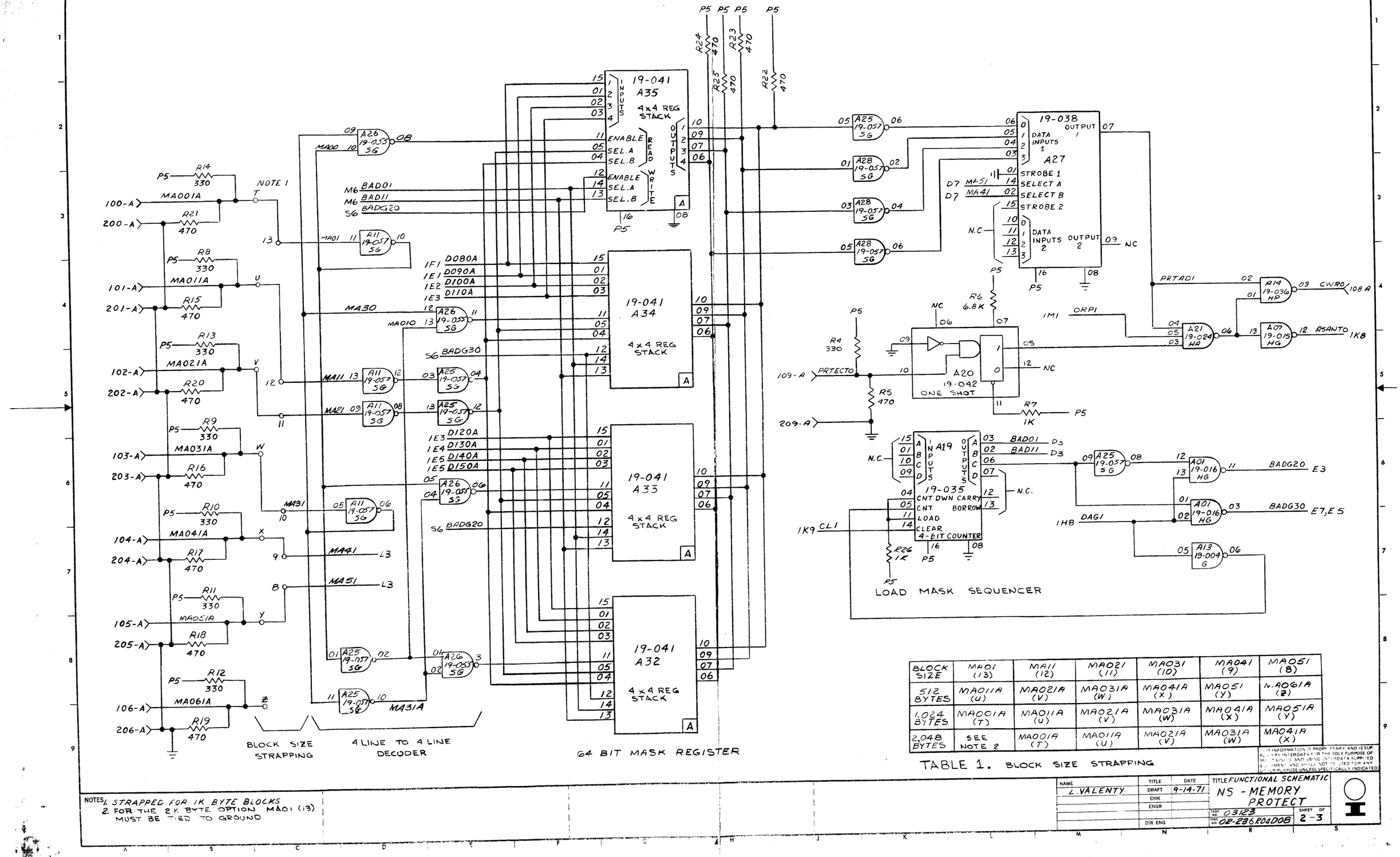
NOTES: 1. PREFERRED ADDRESS X'AE'

NAME	TITLE	DATE
L VALENTY	DRAFT	9-10-71
G J HOMEFIELD	CHK	1-7-72
D. FRANKENBERGER	ENGR	1-14-72
D. W. YOUNG	SYS	1-14-72
R. E. JONES	DIR ENG	1-17-72

TITLE FUNCTIONAL SCHEMATIC NS -MEMORY PROTECT

TASK NO. 03123 DWG NO. 02-23604D08 SHEET 1 OF 3

INTERDATA



BLOCK SIZE	MA01 (13)	MA11 (12)	MA021 (11)	MA031 (10)	MA041 (9)	MA051 (8)
512 BYTES	MA011A (U)	MA021A (V)	MA031A (W)	MA041A (X)	MA051 (Y)	N.A061A (Z)
1,024 BYTES	MA001A (T)	MA011A (U)	MA021A (V)	MA031A (W)	MA041A (X)	MA051A (Y)
2,048 BYTES	SEE NOTE 2	MA001A (T)	MA011A (U)	MA021A (V)	MA031A (W)	MA041A (X)

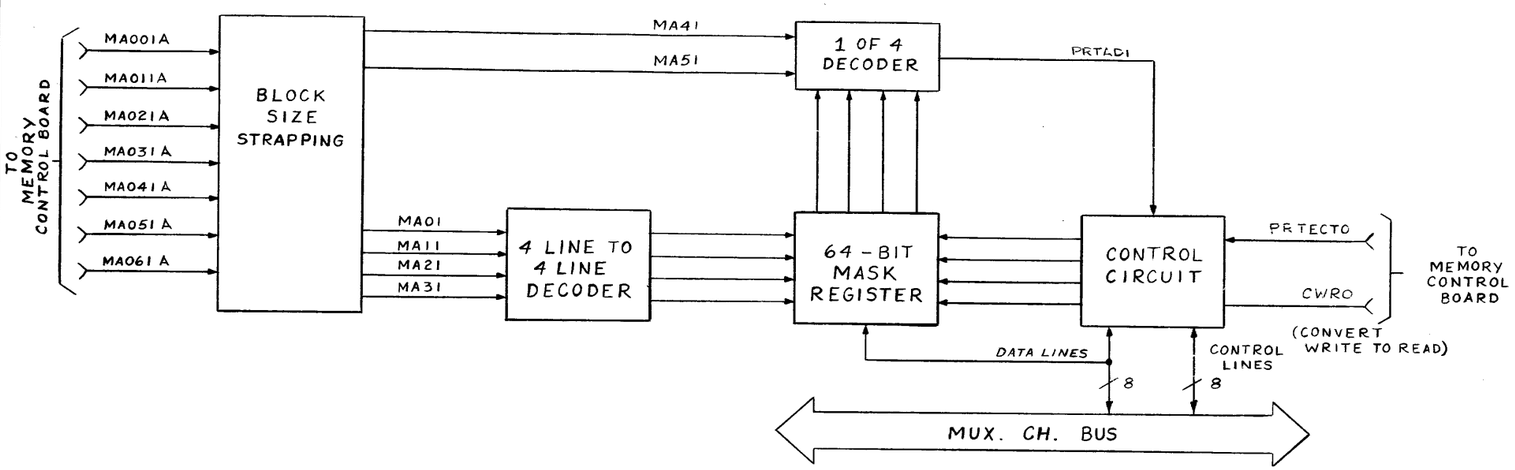
TABLE 1. BLOCK SIZE STRAPPING

NOTES: 1. STRAPPED FOR 1K BYTE BLOCKS
2. FOR THE 2K BYTE OPTION MA01 (13) MUST BE TIED TO GROUND

NAME	TITLE	DATE
L. VALENTY	DRAFT	9-14-71
	CHK	
	ENGR	
	DIR ENG	

TITLE FUNCTIONAL SCHEMATIC
NS - MEMORY PROTECT

TASK NO. 03123
DWG NO. 02-236 ROADOB
SHEET OF 2-3



BLOCK DIAGRAM

HORIZ. POS.	VERTICAL POSITION 2	VERTICAL POSITION 1
00	GND	MA001A
01	GND	MA011A
02	GND	MA021A
03	GND	MA031A
04	GND	MA041A
05	GND	MA051A
06	GND	MA061A
07		CWRO
08	GND	PROTECTO
09	GND	

CABLE MAP

HORIZ. POS.	VERTICAL POSITION 2	VERTICAL POSITION 1
00	GND	P5
01	GND	GND
02	D090	D080
03	D110	D100
04	D130	D120
05	D150	D140
06	ADR50	SR0
07	CMD0	DR0
08	DA0	CLOT0
09	TACK0	RACK0
10	ATN0	SYN0
11		
12	HW0	SCLR0
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		
25		
26		
27		
28		
29		
30		
31		
32		
33		
34		
35		
36		
37		
38		
39	GND	GND
40	GND	GND
41		

BACK PANEL MAP

NOTES

NAME	TITLE	DATE
L.VALENTY	DRAFT	9-14-71
	CHK	
	ENGR	
	DIR ENG	

TITLE FUNCTIONAL SCHEMATIC
NS - MEMORY PROTECT

TASK NO. 03123
DWG NO. 02-236RM08
SHEET OF 3 - 3

INTERDATA®

REVISIONS
FOR PREVIOUS REVISION SEE REV R00 VOIDED COPY IN FILE
03-8 | MCH | 2040 | [illegible] | 7-18-73 | R01

NOTE: SWITCHES SHOWN ARE IN "OFF" POSITION.

PC2-A09 … PC2-A04 — P5 — R13 … R20 4.7K — PCI-12 … PCI-19 — MCH040 … MCH100

PC2-A03 … PC2-A00 — P5 — R19 … R22 4.7K — PCI-18 … PCI-21 — MCH140 … MCH070

PC2-A07 … PC2-A04 — P5 — R21 … R24 4.7K — PCI-20 … PCI-23 — MCH080 … MCH040

PC2-A09 — P5 — R24 4.7K — PCI-23 — MCH090 — P/O 17-283 — GB

MCH040 … MCH140 — A8 19-029 XOR — A9 19-029 XOR — A10 19-029 XOR — A11 19-029 XOR — A12 19-029 XOR

A13 19-024 HA

SEE NOTE 3

P5 — R27 1K. — A3 19-089 DC D — N.C. — N.C.

W/W STAKES — A — B — OPTION SEE NOTE 2

P5 — R33 2.2K — C6 .1uf

A6 19-043 HGO — A5 19-002 G — A7 19-029 XOR

P/O 17-283 PC2 — PC2-ADR MATCHO — GB — SNGLO — GB — NOTE: SWITCHES ARE IN OFF POSITION

P/O 17-283

R32 2.2K — P5 — R32 22 — PCI-29 — CS .1uf — R38 22 — PCI-28

R29 1K — P5 — R36 22 — PCI-30 — PC2-ADR SYNC — TP

3CLROC — 102-3

FTITO — 103-3

R28 1K — P5

BCLKO — 104-3

TO CPU-HI BOARD 35-446

A12 19-025 HA — A11 19-025 HA — A13 19-024 HA

FSADVI — N.C. — N.C. — FADVO

P/O 17-283 PC2 — ADVO — GB — ADVI — PCI-26 — PCI-27 — R34 22 — R35 22 — C3 .1uF — C4 .1uF — R30 2.2K — R31 2.2K — ADVCO

P/O 17-283

C7 THRU C15 .1uF, 30V — P5 — DECOUPLING CAPACITORS

P/O 17-283 — PC2-GA GB — PCI-24 — PCI-25 — C2 2.2uf — C1 2.2uf — 1 mh. ± — 106-3 — 101-3 — 105-3 — 117-3 — TO CPU-HI BOARD 35-446

P/O 17-283 — P/O 17-283 — PC2 — ALL DIODES 33-027

R1 … R12 330 — P5 — PCI-00 … PCI-11 — LED15 … LED08

R2 … R9 330 — P5 — PCI-01 … PCI-08 — LED13 … LED06

R3 … R10 330 — P5 — PCI-02 … PCI-09 — LED12 … LED09

R4 … R11 330 — P5 — PCI-03 … PCI-10 — LED11 … LED08

R7 330 — P5 — PCI-06 — LED10

R8 330 — P5 — PCI-07 — LED07

R5 … R6 330 — P5 — PCI-04 … PCI-05 — LED05 … LED04

A1 … A2 19-044 HGO

R26 — P5 — R25 4.7K

115-3 RAR150 — 114-3 RAR140 — 116-3 RAR130 — 117-3 RAR120 — 118-3 RAR110 — 113-3 RAR100 — 107-3 RAR090 — 110-3 RAR080 — 109-3 RAR070 — 108-3 RAR060 — 119-3 RAR050 — 120-3 RAR040

TO CPU-HI BOARD 35-446

NOTES
1. UNLESS OTHERWISE SPECIFIED ALL APPARATUS IS LOCATED ON TEST CONTROL BOARD 35-505.
2. REFER TO 02-276 TEST AID INCA SPEC.
3. THIS FLIP FLOP-COULD BE USED AS A TEST AID.

NAME	TITLE	DATE
V. PERRI	DRAFT	7-17-73
E CERO	CHK	7-18-73
G. JOYCE	ENGR	
H. ROSS	QA	
J. FRANKENBERGER	DIR ENG	

TITLE FUNCTIONAL SCHEMATIC
TEST AID
SIZE D — CODE IDENT NO 03047 — DWG NO 02-276 R01 — SHEET 1 OF 1

INTERDATA

RELEASED FOR PRODUCTION
ENG. KB DATE 4-23-74

AREA A16 - S1 WAS "ENABL TO PIN 1 & 3 ADDED "DISABLE" & "ENABLE"

01	0264	2212	—	7-31-74	R01

SEE NOTE 1

STRAP OPTION COMMAND

STRAP OPTION ADDRESS

A5 19-021 P — R1 1K — P5
A6 19-021 P — R2 1K — P5
A5 19-021 P — R3 1K — P5
A6 19-021 P — R4 1K — P5

A3 19-021 P
A1 19-021 P
A4 19-021 P
A2 19-021 P
A6 19-021 P

PC00 SH10 (200-3)
PC06 SD001 (109-3)
PC07 SD011 (110-3)
PC08 SD021 (111-3)
PC09 SD031 (112-3)
PC01 SD041 (202-3)
PC02 SD051 (204-3)
PC03 SD061 (205-3)
PC05 SD071 (208-3)
PC04 SL00 (206-3)

TO DISPLAY CONNECTOR CPU-LO (35-440)

CABLE 17-314

PC13 ENABL
PC10 SRQO (113-3)
ESNOO (104-3)
ESNCO (103-3)
INITO (101-3)
C1 } 12VAC
C2 }
POFFO (105-3)

S1 PROGRAM LOAD ENABLE — DISABLE
S2 EXE
S3 INIT
S4 — 01W, 05W, 09W — OFF ON LOCK

PC11 (211-3)
C1 2.2 uf
PC12 (212-3)
(SSGLI)

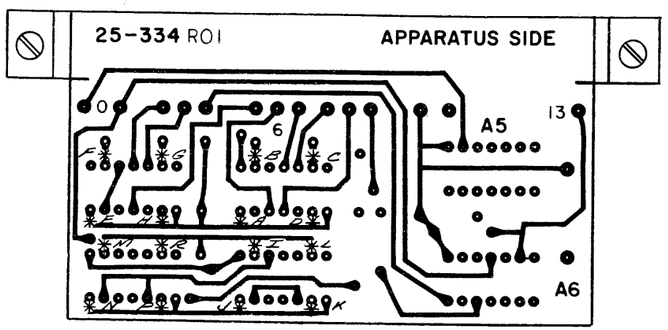
TO DISPLAY CONNECTOR CPU-LO (35-440)

CABLE 17-314

NOTES
1. CIRCUIT AS SHOWN WILL GATE ADDRESS AS X'FF' AND COMMAND AS X'FF'.
2. ALL IC's, CAPS AND RESISTORS ARE PART OF P.C. BOARD 35-430.

NAME	TITLE	DATE
R E CERO	DRAFT	3-18-74
R E CERO	ENR	3-22-74
G JOYCE	ENGR	3-4-74
D BOPP	QA	3-5-74
P ABITANTE	MGR	3-4-74

TITLE FUNCTIONAL SCHEMATIC AUTO/LOAD W/CONSOLE (MODEL 7/16)
TASK NO. 03095
DWG. NO. 02-332 E01 C08
SHEET 1 OF 2



*INDICATES WHERE OPTIONAL CUTS MUST BE ADDED.

NOTES

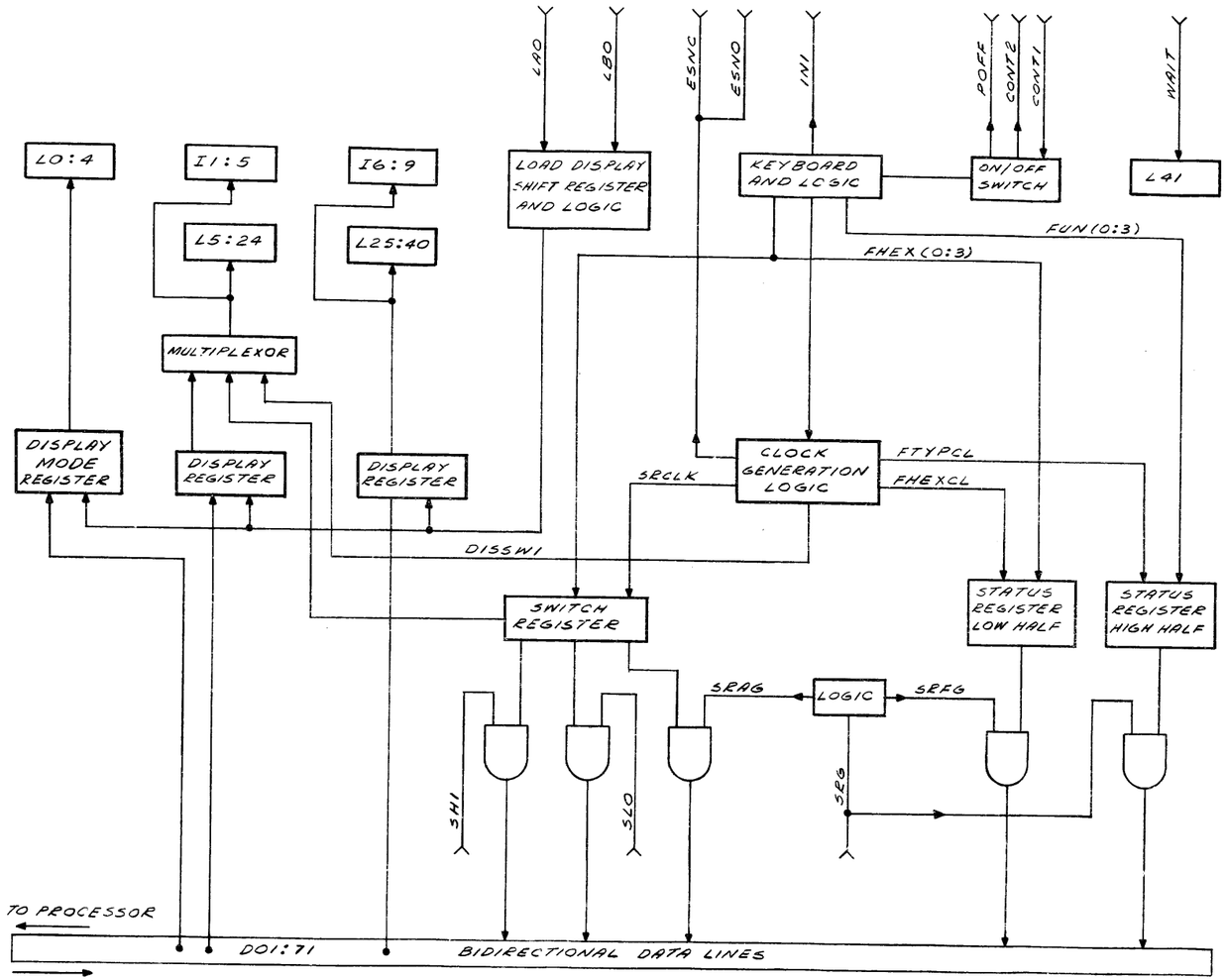
NAME	TITLE	DATE

TITLE: FUNCTIONAL SCHEMATIC AUTOLOAD W/CONSOLE (MODEL 7/16)

DWG 03095

02-334R01/C01

SHEET 2 OF 2



BLOCK DIAGRAM

CONN-3		
TERM	ROW 1	ROW 2
00	GND	SHIO
01	INITO	GND
02	WAITI	D41
03	ESNCO	LAO
04	ESNOO	D51
05	POFFO	D61
06	SSGLI	SLOO
07	SCLRO	
08	GND	D71
09	D01	
10	D11	
11	D21	
12	D31	GND
13	SRGO	CONT3
14	LBO	

REVISIONS

RELEASED FOR PRODUCTION MFG. ENG. [illegible] DATE 4/16/74			
REVISED SHTS 2 & 3			
2	AEH	2323	- 1-31-75 R01
REVISED SHTS 2 & 3			
WS	F	2635	- 12-10-75 R02
REVISED SHT 2			
GP	JT	3590	H 2-15-78 R03

SHEET INDEX	REV	1	2	3	4
	SHT	2	3	4	

REVISION LEVEL OF THIS SHEET IS CONSIDERED THE DOCUMENT

55-519F02 R06	HEXADECIMAL DISPLAY

NOTES

NAME	TITLE	DATE
H. MATTER	DRAFT	1-28-74
H. MATTER	CHK	1-28-74
S. MESSINA	ENGR	2-3[illegible]-74
L. JOHMANN	TEST	2-3[illegible]-74
	DIR ENG	

TITLE: FUNCTIONAL SCHEMATIC HEXADECIMAL DISPLAY

ASSY NO. 03081 SHEET OF 1-4

DWG NO. 09-065R03D08

DRAWING 44 231 16AA2

INTERDATA®

REVISIONS
AREA L1: CONNECTIONS 21, 22, & 23 WERE DIRECTLY CONNECTED TO 1W, 5W & 9W RESP. CROSS REF ALL CONN 3 PIN NOS. TO SHEET 3. AREA G8: C7 WAS 4.7uF. C8 WAS 15uF
PC | MCM | 2323 | / | 1-31-75 | R01
AREA K1: ADDED JUMPER BETWEEN POINTS 03 & 05 AREA L6-L8, A6 10 2-04 WERE TO A6 1 10 & 13
WS | F | 2635 | - | 12-10-75 | R02
AREA R5 R42 WAS 150 Ω
GP | MM | 3590 | 2-15-78 | R03

ALL DIODES 23-001

NOTES
1. LO THRU L42 ARE 33-0227

NAME	TITLE	DATE
P. EDWARDS	DRAFT	12-5-73
	CHK	
	ENGR	
	D.R. ENG	

TITLE FUNCTIONAL SCHEMATIC
HEXADECIMAL DISPLAY

TASK NO. 63081
DWG NO. 09-065-03D08
SHEET 2 OF 4

INTERDATA®

REVISIONS				
AREA A1-A4, B1-B4: ADDED CROSS REF. FROM SHT. 2.				
PE	VGH	2323	—	1-31-75 R01
AREA B2, ADDED SCLRO REF. FROM SHT 2				
WS	✗	2635	-	12-11-75 R02

DBI-C3 CONT3
213-3 TP
214-3 TP
DBI-C2 23 CONT2 2J1
DBI-C1 CONT1 2L2
106-3 14 SSGL1 2J6
107-3 9 SCLRO J1,2J6
105-3 22 POFFO 2J1
101-3 12 INITO 2H1
103-3 15 ESNCO 2S7
104-3 16 ESNOO 2S7
113-3 17 SRGO 2L2
102-3 10 WAITI 2M5
206-3 18 SLOO 2L3
2003 20 SHIO 2L3
114-3 19 LBO 2L5
203-3 13 LAO 2E5
208-3 D71 4
205-3 D61 3
204-3 D51 2
202-3 D41 1
112-3 D31 8
111-3 D21 7
110-3 D11 6
109-3 D01 5
XRPI
2L3 LA1
2L4

FUN00 2D9
FUN10 2C9
SSLO 2G7
FUN20 2C9
SORO 2K7
FUN30 2D9
FTYPCLO 2N7
SCLRO J1
SORI 2J7
SSLI 2G7
HEX01 2B9
HEX11 2B9
HEX21 2A9
HEX31 2A9
FHEXCLO 2N8
SCLRO B2
E1

A46 19-071
A56 19-071
A57 19-043 HGO
A35 19-036 HPO
A45 19-036 HPO
R81 1K P5
R80 1K P5
R78 1K P5

2M2 FR001
2M2 FR000
2M2 SRG1
2R2 SRFG1

A1 19-071
A8 19-109 HGO
L1 L2 L3 L4
R2 R3 R4 R7 150
LP5
LAOA 2N5
LDE1 2R4

P5
R64 R63 R65 R62 R61 R60 R59 R58 330 (TYP.)
R68 R69 R70 R71 R72 R75 R73 R74 470 (TYP.)

D71 4A4
D61 4A4
D51 4A4
D41 4A4
D31 4A4
D21 4A4
D11 4A4
D01 4A4

2R4 LDB1
R79 1K P5

A26 4BIT SHIFT REG. 19-030
A25 4 BIT SHIFT REG. 19-030
A24 4 BIT SHIFT REG. 19-030
A23 4 BIT SHIFT REG 19-030
SHR SHL SER IN MODE CONT

A7 19-109 HGO
A6 19-109 HGO
A5 19-109 HGO
L40 L39 L38 L37
L36 L35 L34 L33
L32 L31 L30 L29
L28 L27 L26 L25
R38 R39 R41 R40 150
R34 R35 R37 R36 150
R30 R31 R33 R32 150
R26 R27 R29 R28 150
LP5 4A8

A17 33-029 HEX DISPLAY
A16 33-029 HEX DISPLAY
A15 33-029 HEX DISPLAY
A14 33-029 HEX DISPLAY

LP5 — L GND
33MF, 10V
C1 & C2

P5 — GND
33MF, 10V
C3 TO C6

P5 — L GND
0.1MF, 30V
C10 TO C14

0.1MF, 30V
C15 TO C39

DECOUPLING CAPACITORS

NOTES

NAME	TITLE	DATE	TITLE FUNTIONAL SCHEMATIC
EDWARDS	DRAFT	12-3-73	HEXADECIMAL DISPLAY
	CHK		
	ENGR		
	DIR ENG		TASK NO. 03081 DWG NO. 09-065R02 D08 SHEET OF 3-4

DRAWING 64 231 16042

INTERDATA®

REVISIONS

2B9 HEX01
2B9 HEX11
2A9 HEX21
2A9 HEX31
2M7 SRCLK0
2H2 SDA0
2N3 SL01
2N2 SHI1
2R2 SRAS1
3R4 D71
3R4 D61
3R5 D51
3R5 D41
3R5 D31
3R5 D21
3R5 D11
3R5 D01
2R4 LDC1
LDD1 2R4
LDE1 2R3
XRP1 2L3
3L8 LP5
DISSW1 2N6

SW161 SW171 SW181 SW191
SW121 SW131 SW141 SW151
SW081 SW091 SW101 SW111
SW041 SW051 SW061 SW071
SW001 SW011 SW021 SW031

A55 A54 A53 A52 A51
19-071

A44 19-036 HPO
A43 19-036 HPO
A42 19-036 HPO
A41 19-036 HPO
A40 19-036 HGO

A34 19-030 4 BIT SHIFT REG
A33 19-030 4 BIT SHIFT REG
A32 19-030 4 BIT SHIFT REG
A31 19-030 4 BIT SHIFT REG
A30 19-030 4 BIT SHIFT REG

A22 19-053 2:1 MUX.
A21 19-053 2:1 MUX.
A20 19-053 2:1 MUX
A19 19-053 2:1 MUX
A18 19-053 2:1 MUX

A5 A4 A3 A2 A8 19-109 HGO
L24 L23 L22 L21
L20 L19 L18 L17
L16 L15 L14 L13
L12 L11 L10 L9
L8 L7 L6 L5
R23 R24 R25 R22
R19 R18 R20 R21
R16 R15 R14 R17
R12 R11 R10 R13
R8 R9 R6 R5
150

A13 A12 A11 A10 A9
33-029 HEX DISPLAY

NOTES

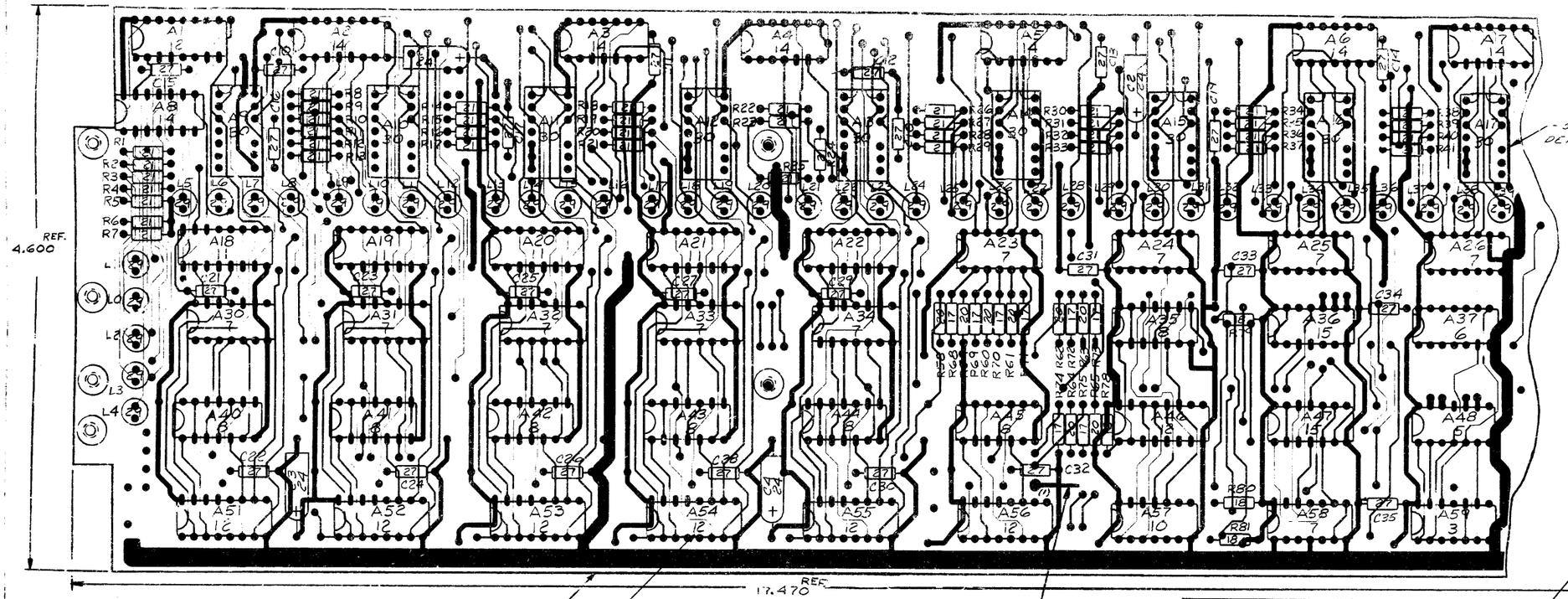
NAME	TITLE	DATE
P EDWARDS	DRAFT	12-3-73
	CHK	
	ENGR	
	DIR ENG	

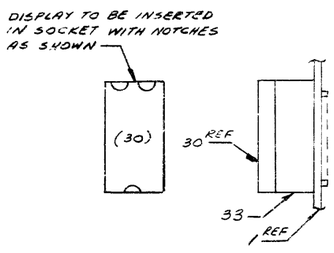
TITLE FUNCTIONAL SCHEMATIC HEXADECIMAL DISPLAY

TASK NO. 03081
DWG NO. 09-065 DOB
SHEET 4 OF 4

INTERDATA

RELEASED FOR PRODUCTION			
MFG. ENG. [illegible]	DATE 2-20-74		
REVISED TO REFLECT E02 COPPER			
PC	WH	[illegible]	4-15-74 E01
CHANGED: ORIENTATION NOTE FOR LED'S DID NOT USE FLAT SIDE			
DM	WH	2074	5-28-74 E02
REVISED R86, WAS IN HORIZONTAL POSITION AREA K8			
DM	WH	2261	9-5-74 E03
REVISED CIRCUITRY TO REFLECT NEW COPPER, STRAP #1 WAS NOT SPEC'D (AREAS H4, L9)			
WS	T	2635	12-11-75 E04
REVISED CIRCUITRY TO REFLECT E05 COPPER. AREA K6 R92 WAS ITEM 21 150Ω.			
GP	WH	3590	2-16-78 E05





DETAIL A
FOR USE ON E02 ONLY

E02	AS SHOWN
E01	LESS ITEMS 30 & 33

ASSEMBLY INFORMATION

NOTES:
1. UNSPECIFIED COMPONENTS ARE ITEM 28.

NAME	TITLE	DATE
F EDWARDS	DRAFT	2-16-74
H MATTER	CHK	19 FEB 74
S MESSINA	ENGR	2-15-74
S MESSINA	SUP	2-15-74

TITLE ASS'Y PRT. CKT. HEXIDECIMAL DISPLAY

03008
35-519-0580

INTERDATA

PRODUCTION

AREAS E17, G11, K10 & M13: P5 WAS SPEC'D AS P2 19-043. AREA N11: R4 WAS 470Ω. AREA H11: R6 WAS 470Ω.

WS	4V	3032	122336 (01)

229-1 MD000 01 P1
130-1 MD010 02 P2
230-1 MD020 03 P3
131-1 MD030 04 P4
231-1 MD040 10 P5
132-1 MD050 11 P6
232-1 MD060 12 P7
133-1 MD070 13 P8
05 P9 INH

A2 19-083

EVEN OUT 09 — EVEN 1
ODD OUT 06 — N.C.
08

233-1 MD080 01 P1
134-1 MD090 02 P2
234-1 MD100 03 P3
135-1 MD110 04 P4
235-1 MD120 10 P5
136-1 MD130 11 P6
236-1 MD140 12 P7
137-1 MD150 13 P8
237-1 MD160 05 P9 INH

A1 19-083

EVEN OUT 09
ODD OUT 06 — N.C.
08 — P5

DECOUPLING CAPACITORS

100-1, 141-1 — P5
C4 33 MFD
C1 .001
C3 .001
200-1, 201-1, 101-1, 241-1, 240-1, 140-1

129-1 MPARO
R7 10K
A9 19-044 HG0 (13, 12)
A6 19-015 HG (3, 4)

A5 19-125 S60 (02, 01 → 03) — PERRO 125-1

A6 19-015 HG (05, 06)
A5 19-125 S60 (07, C5 → 06)
R6 220Ω
R9 220Ω
R5 470

127-1 WRTO
A9 19-044 HG0 (03, 04)
A3 19-095 J K CLK PC (09, 06, 12; 07, 10, 08)
A5 19-125 S60 (09, 10 → 08)
NC

225-1 LRO
A9 19-044 HG0 (01, 02)

R1 1K
R8 470
R3 470
R2 1K
C2 330 PF

229-1 ENMSI
A6 19-015 HG (11, 10)
A5 19-125 S60 (13, 12 → 11)
A6 19-015 HG (01, 02)
A3 19-095 (02, 09, 16; 15, 14, 03) — NC
A9 19-044 HG0 (11, 10)
A9 19-044 HG0 (09, 08)

NOTES
1. UNLESS OTHERWISE SPECIFIED ALL DESIGNATIONS THIS SHEET LOCATED ON 35-216.
2. THIS OPTION, WHEN EQUIPPED IS INSTALLED AT THE BACKPANEL ON SLOT 3, CONNECTOR 1.

NAME	TITLE	DATE
J E FELD	DRAFT	11-1-78
R E ZERO	CHK	11-1-78
G JOYCE	ENGR	12-5-78
C MINUDRI		12-5-78
D FRANKENBERGER	MGR	12-5-78

TITLE SCHEMATIC PARITY OPTION BOARD 7/32
TASK NO. 03132
DWG NO. 02-3684M008
SHEET 1 OF 1

© DAVIS REPROGRAPHICS 40068-3

PUBLICATION COMMENT FORM

Please use this postage-paid form to make any comments, suggestions, criticisms, etc. concerning this publication.

From _____ Date _____

Title _____ Publication Title _____

Company _____ Publication Number _____

Address _____

_____

_____

FOLD FOLD

Check the appropriate item.

☐ Error Page No. _____ Drawing No. _____

☐ Addition Page No. _____ Drawing No. _____

☐ Other Page No. _____ Drawing No. _____

Explanation:

FOLD FOLD

CUT ALONG LINE

Fold and Staple

No postage necessary if mailed in U.S.A.

STAPLE

STAPLE

FOLD

FOLD

FIRST CLASS
PERMIT No. 22
OCEANPORT, N. J.

BUSINESS REPLY MAIL
NO POSTAGE NECESSARY IF MAILED IN U.S.A.

POSTAGE WILL BE PAID BY:

INTERDATA®

Subsidiary of PERKIN-ELMER
Oceanport, New Jersey 07757, U.S.A.

TECH PUBLICATIONS DEPT. MS 322

FOLD

FOLD

STAPLE