Publication Number 29-438R01

MODEL 8/32 MICRO-INSTRUCTION REFERENCE MANUAL

CONSISTS OF:

MICRO-PROGRAM DESCRIPTION	05-058R01A15
MICRO-PROGRAM LISTING	05-058R02A13
MICRO-PROGRAM LISTING	05-058F02A13
ROM MICRO-PROGRAM LISTING	05-059A13

THIS MANUAL CONTAINS PROPRIETARY INFORMATION AND IS SUPPLIED BY INTERDATA FOR THE SOLE PURPOSE OF USING AND MAINTAINING INTERDATA SUPPLIED EQUIPMENT AND SHALL NOT BE USED FOR ANY OTHER PURPOSE UNLESS SPECIFICALLY AUTHORIZED IN WRITING.

INTERDATA®
Subsidiary of PERKIN-ELMER
Oceanport, New Jersey 07757, U.S.A.

May 1976

PAGE REVISION STATUS SHEET

PUBLICATION NUMBER 29-438

TITLE Model 8/32 Micro Instruction Reference Manual

REVISION R01 DATE 5/76

PAGE	REV.	DATE	PAGE	REV.	DATE	PAGE	REV.	DATE
05-058A15 Model 8/32 Micro Program Description R01 5/76			05-058F02A13 Micro Program Listing R00 4/76					
1	R01	5/76	1-70	R00	4/76			
2-3	R00	5/75	05-059A13 ROM Micro Program Listing R00 1/75					
4	R01	5/76	1-8	R00	1/75			
5	R00	5/75						
6	R01	5/76						
7-13	R00	5/75						
14	R01	5/76						
15-32	R00	5/75						
33	R01	5/76						
34-35	R00	5/75						
36	R01	5/76						
37	R00	5/75						
38	R01	5/76						
39	R00	5/75						
40	R01	5/76						
41-44	R00	5/75						
45-63	R01	5/76						
64-67	R00	5/75						
68	R01	5/76						
69-86	R00	5/75						
87	R01	5/76						
88	R00	5/75						
89	R01	5/76						
90-91	R00	5/75						
92-95	R01	5/76						
96	R00	5/75						
97-100	R01	5/76						
101-104	R00	5/75						
105-106	R01	5/76						
05-058A13 Micro Program Listing R02 5/76								
1-58	R02	5/76						

05-058R01A15
January 1976

MODEL 8/32 MICRO-PROGRAM DESCRIPTION

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 program compatability in a family of Processors whose internal hardware varies from one member to the next.

Like its predecessors, the Model 8/32 is designed to execute micro-instructions stored in a Control Store Read-Only-Memory (ROM). Each micro-instruction causes one or more hardware functions to be performed, such as transferring the content of one register to another, arithmetic or logical operations between registers, controlling input/output operations, or initiating main memory accesses.

A series of micro-instructions is called a micro-program. The complete Model 8/32 micro-program is, by definition, an emulator, causing the Model 8/32 micro-processor to react to a user program in main memory and to external events as would the Processor described in the *Model 8/32 Processor User's Manual,* Publication Number 29-428. Every user level instruction, interrupt handling feature and Display Panel function is simulated by some portion of the Model 8/32 micro-program.

2. BLOCK DIAGRAM ANALYSIS

Refer to the Block Diagram in Figure 1.

2.1 System Organization

The Model 8/32 Processor is organized between three 32-bit busses. The A and B Busses are used to present the first and second operand data respectively 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 A, B, and S Busses, as well as the function performed by the ALU is controlled by micro-instructions contained in the Control Store Memory.

2.2 Control Store Memory

The Control Store Memory is a high speed, solid-state, non-destructive memory organized into a maximum of 16 pages of 256 words each. Each word is 32-bits long and represents one micro-instruction. The first five pages (1,280 words) in the Control Store Memory contain the entire Model 8/32 micro-program. Additional pages of writeable Control Store Memory can be added to the basic Model 8/32, allowing the user to supplement the standard instruction repertoire with special algorithms or application oriented functions without requiring hardware involvement.

Each micro-instruction read from the Control Store Memory is placed in the 32-bit ROM Instruction Register (RIR). Most micro-instructions are executed in one machine cycle of 240 nanoseconds. At the conclusion of each micro-instruction, the next micro-instruction to be performed is read out and placed in the RIR. The meaning of the micro-instruction word bits is explained later.

Locations in the Control Store Memory are addressed by the 12-bit output from the ROM Address Gate (RAG). Inputs to the RAG may be the ROM Location Counter (RLC) to select the next micro-instruction to be performed, certain bits of the ROM Instruction Register (RIR) for branches and transfers, the B Bus for data addressing and branches, the user level operation code for entering an emulation routine, or the interrupt control logic for entering interrupt service routines.

Micro-instructions are normally executed from sequential Control Store Memory locations. After a micro-instruction is read into the RIR, the RLC is loaded with the address of the next sequential micro-instruction. When it becomes necessary to jump to a different program sequence, the first micro-instruction in that sequence is addressed through the RAG from ROM Instruction Register (RIR) bits or B Bus bits. The new address is also loaded into the RLC so that sequential instructions can again be executed.

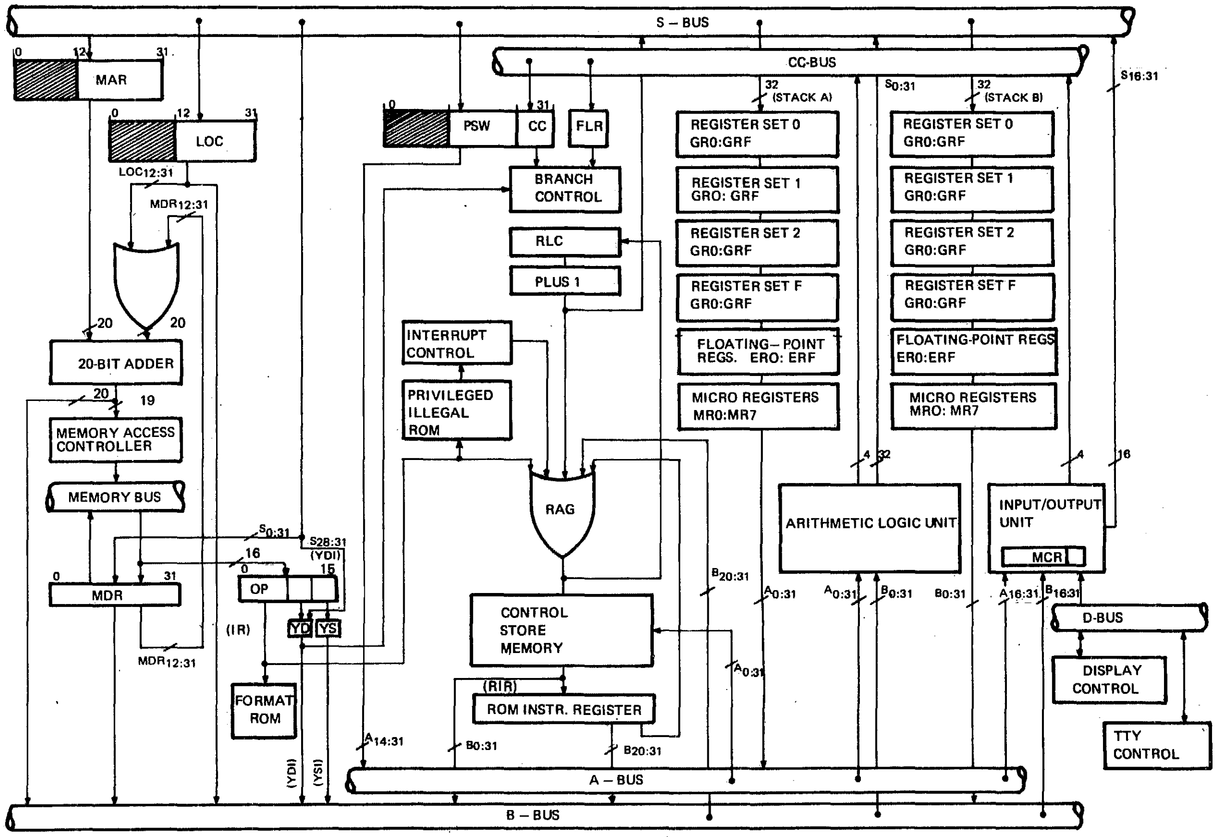


Figure 1. Model 8/32 Block Diagram

During user instruction decoding, the user's operation code times two is presented to the ROM Address Gate to address the first instruction of the sequence that emulates that user's instruction. Again, the new address is also loaded into the RLC.

In response to an interrupt, the interrupt control logic presents an address to the RAG. If the address is that of a branch and link type instruction, the hardware has time to save the current RLC value plus one in the designated link register before the RLC is updated from the ROM Address Gate. This way, the micro-code could return to the interrupted sequence after servicing the interrupt, if desired.

The execute type instructions are the only class where RLC is not updated. After executing the selected instruction, the next micro-instruction performed is the one following the Execute instruction.

2.3 Flag Register (FLR)

The Flag Register (FLR) is a 4-bit register containing the following flags: Carry (C), Overflow (V), Greater than Zero (G), and Less than Zero (L). These flags are modified from the Condition Code Bus at the conclusion of arithmetic and logical operations, and I/O operations to reflect the result of the operation.

2.4 Program Status Word (PSW)

The Program Status Word (PSW) is a 32-bit register used to indicate the system status relative to the user program being emulated. Bits 0:27 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 emulator. Bits 28:31 of the PSW make up the Condition Code field (CC) which reflects the result of the last user level instruction executed. The Condition Code may be updated from the Condition Code Bus, or when the PSW is the specified destination register. Only Bits 14:31 of PSW are implemented.

The Location Counter (LOC) is a 32-bit appendum to PSW, holding the address in main memory of the next user instruction to be performed. During an instruction memory read, the LOC is used to address main memory. For all other main memory accesses, the 32-bit Memory Address Register (MAR) is used.

Only the 20 least significant bits (Bits 12:31) of LOC and MAR are implemented.

2.5. Main Memory

Main Memory consists of a number of 128 KB (Kilo-Byte) 750 nanosecond core memory modules, providing storage for user instructions and data. Data read from or written into memory is buffered in the 32-bit Memory Data Register (MDR).

The micro-program initiates a main memory cycle by issuing a memory read or memory write 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 attempts to load MAR or MDR, or issue another memory command before the current memory cycle is complete, the Processor stops until the desired function can be performed.

After an instruction read has been issued, when the read-out becomes available, Bits 0:7 are placed in the register labeled OP, Bits 8:11 are placed in the register labeled YD, and Bits 12:15 are placed in the register labeled YS. These three registers–OP, YD, and YS– comprise the user's instruction register.

The OP register, containing the user's operation code, is used to address the Privileged/Illegal ROM and the Control Store Memory itself. Twice the user's operation code is the Control Store Memory address of the first micro-instruction of the appropriate emulation sequence. The Privileged/Illegal ROM is a separate Read-Only-Memory containing 256 4-bit words. This ROM is interrogated prior to entering the micro-sequence that emulates a user-level instruction. If the op-code is illegal, or is that of a privileged instruction and PSW Bit 23 is set, the Illegal Instruction Interrupt is generated.

2.6 General Registers

The two to eight sets of user level General Registers each contain 16 32-bit registers. The register sets (stacks) are duplicated for the A Bus and B Bus. Refer to Figure 1. Only one set of General Registers is active at a time, dependin' upon PSW Bits 25, 26, and 27. See Table 1.

TABLE 1. REGISTER SET SELECTION

PSW Bits			Active Register Set
25	26	27	
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	F

The micro-program usually accesses the user's General Registers without caring which of the 16 registers in the active set it gets. However, it does matter 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 the instruction read, the register addresses specified by the user are in the YD and YS registers, the micro-program can access the appropriate General Register by specifying the YD or YS register. The hardware then selects the General Register whose number is in the YD or YS register.

The user's General Registers are also directly addressable to the micro-program when it is necessary to access specific registers.

2.7 Floating-Point Registers

The 16 32-bit floating-point registers are directly addressable or indirectly addressable (through YD or YS) by the floating-point micro-instructions.

2.8 Micro-Registers

The eight 32-bit Micro-Registers (MR0:7) are available to the micro-program as general purpose registers.

2.9 Arithmetic Logic Unit (ALU)

The 32-bit A Bus holds the first operand for arithmetic and logical operations. The 32-bit B Bus holds the second operand. The A and B Busses are input to the Arithmetic Logic Unit (ALU). The ALU performs Addition, Subtraction, Multiplication, Division, Shifting, and Boolean connect functions. The output of the ALU is the 32-bit S Bus.

2.10 Input/Output

Input/output (I/O) operations are accomplished by gating data from the A and/or B Busses onto the 16-bit I/O Bus and gating data from the I/O Bus onto the S Bus.

The I/O Bus consists of 33 lines; 16 bi-directional data lines, 10 control lines to identify the type of data transfer, 6 test lines and an initialize line. See Section 6.

2.11 Interrupt Control

The interrupt control logic provides rapid response to internal and external events. Nine priority interrupt lines are available each with a unique trap location in the Control Store Memory. Recognition of an interrupt causes the micro-instruction at the trap location to be performed. Certain interrupts can be disabled or enabled by bits of the Program Status Word (PSW). Interrupts can also be disabled or enabled as a group by the micro-program. See Table 2.

TABLE 2. INTERRUPT TRAPS

INTERRUPT	TRAP ADRS (HEX)	MASK	GROUP ENABLE
Memory Access Controller (Instruction)	1FE	PSW21	NO
Memory Access Controller (Data)	207		
Primary Power Fail	206	NONE	YES
Machine Malfunction	205	PSW18	YES
Display Panel	204	NONE	YES
External Interrupt Level 0	203	see table 3	YES
External Interrupt Level 1	202		
External Interrupt Level 2	201		
External Interrupt Level 3	200		
Illegal Instruction	208	NONE	N/A
Privileged Instruction	208	PSW23	N/A

PSW Bits 17 and 20 define the external Interrupt enable status of the Processor as shown below:

PSW	BITS	
17	20	
0	0	All Levels Disabled
0	1	Higher Levels Enabled
1	0	All Levels Enabled
1	1	Current and Higher Levels Enabled

where the current level is a function of the currently active register set. See Table 3.

TABLE 3. EXTERNAL INTERRUPT ENABLE

PSW BITS					EXTERNAL INTERRUPT ENABLED			
17	20	25	26	27	LEVEL 0	LEVEL 1	LEVEL 2	LEVEL 3
0	0	X	X	X	NO	NO	NO	NO
0	1	0	0	0	NO	NO	NO	NO
0	1	0	0	1	YES	NO	NO	NO
0	1	0	1	0	YES	YES	NO	NO
0	1	0	1	1	YES	YES	YES	NO
0	1	1	0	0	YES	YES	YES	NO
0	1	1	0	1	YES	YES	YES	NO
0	1	1	1	0	YES	YES	YES	NO
0	1	1	1	1	YES	YES	YES	NO
1	0	X	X	X	YES	YES	YES	YES
1	1	0	0	0	YES	NO	NO	NO
1	1	0	0	1	YES	YES	NO	NO
1	1	0	1	0	YES	YES	YES	NO
1	1	0	1	1	YES	YES	YES	YES
1	1	1	0	0	YES	YES	YES	YES
1	1	1	0	1	YES	YES	YES	YES
1	1	1	1	0	YES	YES	YES	YES
1	1	1	1	1	YES	YES	YES	YES

2.12 Machine Control Register (MCR)

The 12-bit Machine Control Register (MCR) can be interrogated or cleared by the micro-program. The meaning of the MCR bits is shown in Table 4.

TABLE 4. MCR BIT DEFINITION

BIT	MNEMONIC	MEANING
15	EPE	Early Power Fail Detect
14	MM	Memory Malfunction (Parity Error)
13	MM	
12	Unused	
11	STF	Start Timer Failure
10	CATN	Console Attention
09	RSET	Register Sets Available
08	Spare	Spare (Strap)
07	SNGL	Console Single Mode
06	INIT	Initialize Switch on console
05	HWCRC	Hardware Assist CRC option
04	DFU	DFU option

3. DATA AND INSTRUCTION FORMATS

3.1 Data Formats

All internal data paths except those to the Input/Output control are 32-bits wide. The basic machine operand is consequently a 32-bit fullword. Positive fixed-point data is expressed in true binary form with a Sign bit of zero. Negative fixed-point data is expressed in two's complement notation with a Sign bit of one. Floating-point data is expressed as a signed magnitude fraction with a signed exponent. The quantity expressed is the product of the fraction and 16 raised to the power of the exponent. Each floating-point number requires a 32-bit fullword; 8-bits are used for the fraction sign and exponent, and 24-bits are used for the fraction.

Binary information is represented in hexadecimal notation (base 16) for simplicity.

3.2 Instruction Formats

Model 8/32 micro-instructions can be one of six formats designated Address Link, Register Link, Register to Register Transfer, Register to Register Control, Register to Register Immediate, and Register Write. The instruction formats are shown in Figure 2.

ADDRESS LINK

0 2	3	4	5	6 10	11 13	14 25	26	27	28 31
0 0 0	1	X	T	S	F	ADDRESS	E	D	MC

REGISTER BRANCH

0 2	3	4	5	6 10	11 13	14 19	20 24	25	26	27	28 31
0 0 0	0	X	T	NULL	F	▨	B	▨	E	D	MC

REGISTER TO REGISTER TRANSFER

0 2	3	4	5	6 10	11 15	16 19	20 24	25	26 31
Module	0	0	I	S	A	F	B	C	PAGE ADDRESS

REGISTER TO REGISTER CONTROL

0 2	3	4	5	6 10	11 15	16 19	20 24	25	26	27	28 31
Module	0	1	I	S	A	F	B	K	E	D	MC

REGISTER TO REGISTER IMMEDIATE

0 2	3	4	5	6 10	11 15	16 19	20 31
Module	1	0	I	S	A	F	DATA

REGISTER WRITE

0 2	3	4	5	6 10	11 15	16 19	20 24	25	26	27	28 31
0 0 1	1	1	I	NULL	A	F	B	K	E	D	MC

Figure 2. Instruction Word Formats

The basic instruction format provides the micro-processor with a three address capability, but various options of the repertoire can modify their range from two to four.

Bits 0, 1, and 2 of the micro-instruction, select the Processor module that will perform the specified function. The Address Link and Register Link micro-instructions are the only ones that select Module 0, the control module. The other micro instruction formats can be directed to any other module. The Model 8/32 Micro Code Assembler recognizes symbolic operation codes directed to Modules 0 (the control module), 1 (the ALU module), 2 (the I/O module), and 3 (the floating-point ALU module).

The meaning of the micro-instruction word fields is summarized in Table 5 and the following paragraphs.

TABLE 5. INSTRUCTION WORD FIELDS

FIELD	MEANING
A	Selects Register to be used as first operand.
B	Selects Register to be used as second operand.
C	If set, transfer is conditional.
D	Decode next user instruction.
E	Enable setting of Condition Code.
F	Specifies function of addressed module.
I	B Bus data addresses actual data in Control Store.
K	F field extention.
MC	Memory Control field.
S	Selects register to receive the result.
T	If set, item F must be true for transfer.
X	Execute

The F field in all formats specifies the function that the selected module is to perform. The X-bit in the Address Link and Register Link formats distinguishes Execute and Link instructions from Branch and Link instructions. The T-bit specifies whether the true or false state of the condition F is to be tested.

The S field selects the S Bus register to be loaded. The A field selects the first operand (A Bus) register. The B field selects the second operand (B Bus) register.

Setting the I-bit causes the operand developed on the B Bus to be taken as a Control Store Memory address. The fullword contents of the addressed location replaces the original B Bus data.

Setting the C-bit on Register to Register Transfer instructions makes the transfer occur only if no predefined signal is returned from the addressed module. For the ALU module, the signal is carry, meaning no transfer occurs if a carry is generated.

The K-bit is used as an extension of the F field, allowing more than 16 functions to be performed by the addressed module.

The E-bit allows the Condition Code (CC) field to be updated from data on the CC Bus from the addressed module. Once an instruction with the E-bit set has been performed, the Condition Code remains connected to the CC Bus until an instruction with the E-bit reset is fetched.

The D-bit enables the Privileged/Illegal ROM and the instruction decoding hardware. Unless a branch is taken or an interrupt occurs, a user instruction emulation sequence is entered.

The MC field controls main memory accesses, and MAR and LOC activities.

The most significant bit of the 12-bit immediate field on Register to Register Immediate instructions is propagated as the most significant 20-bits on the B Bus. For example, the immediate operand '400' produces the value '0000 0400' on the B Bus. The immediate operand '800' produces the value 'FFFFF800' on the B Bus.

The 6-bit Address field on Register to Register Transfer instructions can specify any address within the local 64 word page. For example, an instruction at address '131' can transfer to any other instruction from address '100' to '13F'.

3.2.1 Address Link. On the Address Link instructions, the incremented contents of the RLC are placed in the selected S Bus register. Then, if the condition specified by F and T is met, the next micro-instruction executed is the one at the location specified by the 12-bit ADDRESS field. If not, the next micro-instruction in sequence is executed. In addition, if the condition is met, any memory control or decode options specified are suppressed.

3.2.2 Register Branch. The Register Branch instructions are identical to the Address Link instructions except the address to transfer to is taken from the register specified by B. The destination register must be specified as NULL.

3.2.3 Register to Register Transfer. These instructions perform function F using the contents of the register specified by A as the first operand and an effective second operand specified by B. The result replaces the register specified by S. Then, if the C-bit is reset, or if a special signal is not returned from the addressed module, the next micro-instruction executed is from the Control Store Memory address specified by the PAGE ADDRESS field on the current page. If the C-bit is set and the special signal is returned from the addressed module, the next micro-instruction in sequence is executed. The PAGE ADDRESS field can only specify the least significant 6-bits of a Control Store Memory address. The remaining address bits are taken from the high order 6-bits of RLC. This means that a transfer can only occur to a location within the 64 word page defined by RLC Bits 4:9. An exception is when the micro-instruction is at the end of a page boundary (e.g., address '23F'). In this instance, the transfer occurs to the specified address on the next sequential page (e.g., addresses '240' through '27F').

The effective second operand, B_E, is the contents of the register specified by B if $I = 0$.

$$B_E = (B)$$

or the contents of the fullword Control Store Memory location whose address is in the register specified by B if $I = 1$.

$$B_E = [\ (B)\ ]$$

3.2.4 Register to Register Control. These instructions perform function F using the contents of the register specified by A as the first operand and an effective second operand specified by B. The result replaces the contents of the register specified by S.

The effective second operand, B_E, is the contents of the register specified by B if $I = 0$:

$$B_E = (B)$$

or the contents of the fullword Control Store Memory location whose address is in the register specified by B if $I = 1$:

$$B_E = [\ (B)\ ]$$

At the conclusion of the instruction, any specified MC or D options are performed.

3.2.5 Register to Register Immediate. The function specified by F is performed using the contents of the register specified by A as the first operand and an effective second operand specified by the DATA field. The result replaces the contents of the register specified by S.

The effective second operand, B_E, is the 12-bit value of the DATA field with the most significant 20 bits equal to Bit 20 if $I = 0$:

$$B_E = \text{DATA}$$

or the contents of the fullword Control Store Memory location whose address is DATA if $I = 1$:

$$B_E = [\text{DATA}]$$

3.2.6 Register Write. The Register Write instruction stores the contents of the register specified by A into the Dynamic Control Store (DCS) location whose address is in the register specified by B. After the write, any specified MC or D options are performed.

If the Processor is not equipped with DCS, only the MC or D options are performed.

3.3 Main Memory Control

The Processor's main memory is the source of user's instructions and data. Control over the main memory is provided in the MC field of the Address Link, Register Link, Register to Register Control, and Register Write micro-instructions.

Table 6 and the following paragraphs describe the MC field options.

TABLE 6. MC FIELD

BITS				MEANING	
28	29	30	31		
0	0	0	0	No Action	
0	0	0	1	IL	Increment LOC by Instruction Length
0	0	1	0	PW2	Privileged Write Halfword (two bytes)
0	0	1	1	DW2	Data Write Halfword
0	1	0	0	No Action	
0	1	0	1	I4DW4	Increment MAR by 4, Data Write Fullword
0	1	1	0	PW4	Privileged Write Fullword
0	1	1	1	DW4	Data Write Fullword
1	0	0	0	RAS	Read Halfword and Set Sign Bit
1	0	0	1	ILIR	Increment LOC by Length and Read Instruction
1	0	1	0	PR2	Privileged Read Halfword
1	0	1	1	DR2	Data Read Halfword
1	1	0	0	IR	Instruction Read
1	1	0	1	14DR4	Increment MAR by 4, Data Read Fullword
1	1	1	0	PR4	Privileged Read Fullword
1	1	1	1	DR4	Data Read Fullword

IL The Location Counter (LOC) is incremented by the length in bytes of the last user level instruction fetched.

PW2 The Memory Access Controller (MAC) is disabled and the halfword in MDR, Bits 16:31, is written into the addressed location.

DW2 The halfword in MDR, Bits 16:31, is written into the addressed location. MAC is not disabled.

I4DW4 The Memory Address Register (MAR) is incremented by four. Then the fullword in MDR, Bits 0:31, is written into the location addressed by MAR.

PW4 The MAC is disabled and the fullword in MDR, Bits 0:31, is written into the addressed location.

DW4 The fullword in MDR, Bits 0:31, is written into the addressed location.

RAS The halfword at the addressed location is read then re-written with Bit 0 of the halfword set. The original value of the halfword replaces MDR Bits 16:31. Bits 0:15 of the MDR are set equal to Bit 16 of MDR (sign extension).

ILIR LOC is incremented by the length in bytes of the last user instruction fetched. Then an Instruction Read is started from the address specified by the new value of LOC.

PR2 The MAC is disabled and the halfword at the addressed location is read and copied to MDR Bits 16:31. Bits 0:15 of MDR are set equal to MDR Bit 16.

DR2 The halfword at the addressed location is read and copied to MDR Bits 16:31. Bits 0:15 of MDR are set equal to MDR Bit 16.

IR An Instruction Read is started from the memory address specified by LOC.

I4DR4 MAR is incremented by four. Then the fullword at the location addressed by the new value of MAR is read and copied to MDR.

PR4 MAC is disabled. Then the fullword at the location addressed by MAR is read and copied to MDR.

DR4 The fullword at the location addressed by MAR is read and copied to MDR.

All Main Memory Control is conditional when used within Address Link and Register Branch micro-instructions. The control is only affected if the instruction does not result in a transfer.

All increment functions are done before the micro-instruction terminates. Memory read and write functions do not start until after the micro-instruction terminates. This allows the micro-program to use MAR or MDR as a destination and then begin a memory read or write in the same micro-instruction.

4. SOURCE AND DESTINATION REGISTERS

The Model 8/32 has 93 registers that are addressable by the micro-program. Most of these are available to the A, B and S Busses. Table 7 and the following paragraphs explain the exceptions and special cases.

TABLE 7. REGISTER ADDRESSES

HEX ADDRESS	S BUS	A BUS	B BUS	CATEGORY
00	0	0	0	USER'S GENERAL REGISTERS
01	1	1	1	
02	2	2	2	
03	3	3	3	
04	4	4	4	
05	5	5	5	
06	6	6	6	
07	7	7	7	
08	8	8	8	
09	9	9	9	
0A	10	10	10	
0B	11	11	11	
0C	12	12	12	
0D	13	13	13	
0E	14	14	14	
0F	15	15	15	
10	MR0	MR0	MR0	MICRO REGISTERS
11	MR1	MR1	MR1	
12	MR2	MR2	MR2	
13	MR3	MR3	MR3	
14	MR4	MR4	MR4	
15	MR5	MR5	MR5	
16	MR6	MR6	MR6	
17	MR7	MR7	MR7	
18	YS	YS	YS	SPECIAL PURPOSE
19	YD	YD	YD	
1A	LOC	YX	LOC	
1B	MDR	YDPI	MDR	
1C	MAR	–	MAR	
1D	PSW	PSW	YSI	
1E	YDI	–	YDI	
1F	NULL	NULL	NULL	

Although the user's General Registers in the register set specified by PSW Bits 25, 26 and 27 can be addressed directly by the micro-program, it is often more convenient to access the General Register specified in the user's instruction without regard to its physical number. The symbolic addresses YD, YDP1, YS, and YX allow just that. Specifying YD causes the General Register whose number appears in the YD field of IR (IR Bits 8:11) to be selected. Specifying YDP1 causes the odd member of the even-odd pair of General Registers, one of whose number appears in the YD field of IR, to be selected. Specifying YS causes the General Register whose number appears in the YS field of IR (IR Bits 12:15) to be selected. Specifying YX is the same as specifying YS except when the YS field of IR is zero. Then, all zeros are placed on the A Bus. This automatic feature is used to develop the index value for the user level RI1, RI2, and RX format instructions.

On micro-instructions that address the Floating-Point Module, the corresponding floating-point register is selected instead of a General Register.

Specifying YDI or YSI as a source causes the corresponding field of IR – YD or YS – to be placed on Bits 28:31 of the B Bus. The upper 28-bits of the B Bus are zero.

Specifying NULL as a source on the A or B Bus causes the corresponding bus to be set to zero. Specifying NULL as the S Bus destination causes the data to be lost.

Specifying MDR as a source after a memory read operation causes the Processor to wait until the memory data becomes available. Following an Instruction Read and Decode function, MDR participates in the formation of the effective address, if the user's instruction is one of the RX formats. Specifically, until MAR is loaded, any reference to MDR as a source causes the second level index register (SX2) to be accessed if the instruction format is RX3. Otherwise, MDR is accessed. Refer to Section 7 for details.

Specifying LOC, MAR, or MDR as a destination when a memory access is in progress causes the Processor to wait until the memory access is completed.

Specifying MAR as a source, produces meaningful results on the B Bus only after an Instruction Read and Decode function. Referring to the block diagram on Figure 1, when MAR is specified as the source, the output of the 20-bit adder is presented to the B Bus instead of MAR. This output is the actual contents of MAR, or the sum of MAR and LOC, or the sum of MAR and MDR, depending upon the format of the last user Instruction read. Refer to Section 7 for details.

The Condition Code field of PSW can be manipulated by any addressed module unless PSW is the explicit destination or a Condition Code change was inhibited by the E-bit in a micro-instruction.

The bits of PSW that have hardware implications are:

PSW17	Interrupt priority selection (see Table 3.)
PSW18	Machine Malfunction interrupt enable
PSW20	Interrupt priority selection (see Table 3.)
PSW21	Relocation/Protection interrupt enable
PSW23	Protect mode
PSW25,26,27	Register Set Selection
PSW28:31	Condition Code

5. INSTRUCTION REPERTOIRE

5.1 Introduction

The instruction repertoire has been grouped by function in this section. The operation of each instruction is presented in the following format.

1. An instruction word chart for each instruction including mnemonic operation code and operand designations in the correct Assembler format. The format type and an instruction diagram with operation code and the location of all fields is also provided.

2. A description of instruction operation.

3. A diagrammatic representation of instruction operation.

4. A chart showing the possible resultant flags.

5. The execution time in nanoseconds. On all microinstructions, add 180 nanoseconds if I = 1.

6. A programming note may be provided to add pertinent or clarifying information.

The symbols and abbreviations used in the instruction descriptions are defined as follows:

( ) Parenthesis or brackets. Read as "the content of –"
[ ]
← Arrow. Read as "is replaced by –" or "replaces–"
A The A field. First operand register specification.
B The B field. Second operand register specification.
S The S field. Destination register specification.
(0:7) A bit grouping within a word. Read as "Bits 0 through 7 inclusive", etc.

B_E The effective second operand. If the instruction format is RR Control or RR Transfer, the effective second operand is the contents of the register specified by B if the Indirect (I) bit is reset:

$$B_E = (B) \text{ if } I=0$$

If the Indirect bit is set, the effective second operand is the content of the fullword Control Store location whose address is contained in the register specified by B:

$$B_E=[(B)] \text{ if } I=1$$

If the instruction format is RR Immediate, the effective second operand is the 12-bit data field if the Indirect bit is reset:

$$B_E=\text{DATA if } I=0$$

If the Indirect bit is set, the effective second operand is the contents of the fullword Control Store location whose address is the Data field:

$$B_E=(\text{DATA}) \text{ if } I=1$$

5.2 Logical Instructions

The instructions described in this section are:

5.2.1	L	Load
	LX	Load and Transfer
	LI	Load Immediate
5.2.2	STR	Store
5.2.3	N	AND
	NX	AND and Transfer
	NI	AND Immediate
5.2.4	O	OR
	OX	OR and Transfer
	OI	OR Immediate
5.2.5	X	Exclusive OR
	XX	Exclusive OR and Transfer
	XI	Exclusive OR Immediate

5.2.1 Load

L S,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
0 0 1	0 1	I	S	1 1 1 1 1	0 0 0 1	B	0	E	D	MC

LX S,B,ADRS,I,C [RR TRANSFER]

0	3	5	6	11	16	20	25	26 31
0 0 1	0 0	I	S	1 1 1 1 1	0 0 0 1	B	C	PAGE ADDRESS

LI S,DATA,I [RR IMMEDIATE]

0	3	5	6	11	16	20 31
0 0 1	1 0	I	S	1 1 1 1 1	0 0 0 1	DATA

The second operand is loaded into the register specified by S.

L,LI : $(S) \leftarrow B_E$
LX : $(S) \leftarrow B_E$
then $(RLC_{10:15}) \leftarrow$ PAGE ADDRESS

Resulting Flags

C	V	G	L	
0	0	0	0	Result is zero
0	0	0	1	Result is less than zero
0	0	1	0	Result is greater than zero

Programming Note

The Load instruction assembles as an Add instruction with a NULL A field.

Execution Times

L,LI,LX : 240

5.2.2 Store

STR A,B,E,D,MC [R WRITE]

0	3	5	6	11	16	20	25	26	27	28 31
0 0 1	1 1	1	1 1 1 1 1	A	0 0 0 0	B	0	E	D	MC

The content of the register specified by A is stored in the Control Store Memory location whose address is in the register specified by B.

STR: (A) → [(B)]

Execution Time

STR: 420

5.2.3 AND

N S,A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
0 0 1	0 1	I	S	A	0 1 0 1	B	0	E	D	MC

NX S,A,B,ADRS,I,C [RR TRANSFER]

0	3	5	6	11	16	20	25	26 31
0 0 1	0 0	I	S	A	0 1 0 1	B	C	PAGE ADDRESS

NI S,A,DATA,I [RR IMMEDIATE]

0	3	5	6	11	16	20 31
0 0 1	1 0	I	S	A	0 1 0 1	DATA

The logical product of the first and second operand replaces the contents of the register specified by S. The 32-bit product is formed on a bit-by-bit basis.

N, NI : (S) ← (A) AND B_E
NX : (S) ← (A) AND B_E
Then ($RLC_{10:15}$) ← PAGE ADDRESS

Resulting Flags

C	V	G	L	
0	0	0	0	Product is zero
0	0	0	1	Product is not zero
0	0	1	0	

Execution Times

N,NI,NX : 240

5.2.4 OR

O S,A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
0 0 1	0 1	I	S	A	0 1 1 1	B	0	E	D	MC

OX S,A,B,ADRS,I,C [RR TRANSFER]

0	3	5	6	11	16	20	25	26 31
0 0 1	0 0	I	S	A	0 1 1 1	B	C	PAGE ADDRESS

OI S,A,DATA,I [RR IMMEDIATE]

0	3	5	6	11	16	20 31
0 0 1	1 0	I	S	A	0 1 1 1	DATA

The logical sum of the first and second operands replaces the contents of the register specified by S. The 32-bit sum is formed on a bit-by-bit basis.

O,OI : (S) ← (A) OR B_E
OX : (S) ← (A) OR B_E
then ($RLC_{10:15}$) ← PAGE ADDRESS

Resulting Flags

C	V	G	L	
0	0	0	0	Sum is zero
0	0	0	1	Sum is not zero
0	0	1	0	

Execution Times

O,OI,OX : 240

5.2.5 Exclusive OR

X S,A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
0 0 1	0 1	I	S	A	0 1 1 0	B	0	E	D	MC

XX S,A,B,ADRS,I,C [RR TRANSFER]

0	3	5	6	11	16	20	25	26 31
0 0 1	0 0	I	S	A	0 1 1 0	B	C	PAGE ADDRESS

XI S,A,DATA,I [RR IMMEDIATE]

0	3	5	6	11	16	20 31
0 0 1	1 0	I	S	A	0 1 1 0	DATA

The logical difference between the first and second operands replaces the contents of the register specified by S. The 32-bit difference is formed on a bit-by-bit basis.

X, XI : $(S) \leftarrow (A)$ XOR B_E
XX : $(S) \leftarrow (A)$ XOR B_E
then $(RLC_{10:15}) \leftarrow$ PAGE ADDRESS

Resulting Flags

C	V	G	L	
0	0	0	0	Difference is zero
0	0	0	1	Difference is not zero
0	0	1	0	

Execution Times

X, XI, XX : 240

5.3 Branch/Execute and Link Instructions

These instructions are programmed decisions providing entry to and return from subprograms, as well as testing the results of arithmetic, logical, and other machine operations.

Most Processor operations result in setting the micro-flag register. The state of this flag register is testable with the Branch/Execute and Link on condition instructions.

The Execute and Link instructions allow conditional execution of a single, non-sequential micro-instruction. No branch is actually taken, and unless the instruction executed is a Branch instruction or otherwise results in a transfer, control returns to the instruction following the Execute and Link.

The address plus 1 of the Branch/Execute and Link instruction is always saved in the specified link register, even if the condition for doing the Branch of Execute is not met.

The instructions described in this section are:

5.3.1	BAL	Branch and Link
	BALA	Branch and Link and Arm interrupts
	BALD	Branch and Link and Disarm interrupts
	BALZ	Branch and Link on Zero
	BALNZ	Branch and Link on Not Zero
	BALL	Branch and Link on Less
	BALNL	Branch and Link on Not Less
	BALG	Branch and Link on Greater
	BALNG	Branch and Link on Not Greater
	BALV	Branch and Link on Overflow
	BALNV	Branch and Link on No Overflow
	BALC	Branch and Link on Carry
	BALNC	Branch and Link on No Carry
	BALT	Branch and Link on True CC Match
	BALF	Branch and Link on False CC Match
5.3.2	EXL	Execute and Link
	EXLA	Execute and Link and Arm interrupts
	EXLD	Execute and Link and Disarm interrupts
	EXLZ	Execute and Link on Zero
	EXLNZ	Execute and Link on Not Zero
	EXLL	Execute and Link on Less
	EXLNL	Execute and Link on Not Less
	EXLG	Execute and Link on Greater
	EXLNG	Execute and Link on Not Greater
	EXLV	Execute and Link on Overflow
	EXLNV	Execute and Link on No Overflow
	EXLC	Execute and Link on Carry
	EXLNC	Execute and Link on No Carry
	EXLT	Execute and Link on True CC Match
	EXLF	Execute and Link on False CC Match

5.3.1 Branch and Link

F ADDRESS (LINK),E,D,MC [ADDRESS LINK]

0	3	5	6	11	14	26	27	28 31
0 0 0	10	T	LINK	F	ADDRESS	E	D	MC

F (B)(NULL),E,D,MC [REGISTER LINK]

0	3	5	6	11	14	20	25	26	27	28 31
0 0 0	00	T	NULL	F	▨	B	▨	E	D	MC

where F =

	T	F		
BALZ	0	0	0	0
BALL	0	0	0	1
BALG	0	0	1	0
BALF	0	0	1	1
BALC	0	1	0	0
BALV	0	1	0	1
BAL	0	1	1	0
BALA	0	1	1	1
BALNZ	1	0	0	0
BALNL	1	0	0	1
BALNG	1	0	1	0
BALT	1	0	1	1
BALNC	1	1	0	0
BALNV	1	1	0	1
BALD	1	1	1	1

The address of the next sequential micro-instruction replaces the contents of the register specified by LINK, then a transfer is conditionally taken to the address specified. In the Address Link format, the ADDRESS field of the instruction contains the branch address. In the Register Link format, the branch address is contained in the register specified by B. This format is used to return from subroutines.

Tested Condition True

(LINK) ← $(RLC_{4:15})$+1
$(RLC_{4:15})$ ← ADDRESS [Address Link]
$(RLC_{4:15})$ ← (B) [Register Link]

Tested Condition False

(LINK) ← $(RLC_{4:15})$+1
$(RLC_{4:15})$ ← $(RLC_{4:15})$+1

Programming Notes

For the BALT and BALF instructions, a logical AND is performed between each bit in the Condition Code field of PSW and the M1 field of the user's instruction $(IR_{8:11})$. If any resultant bit is a ONE, the BALT instruction will branch and the BALF instruction will not. If all resultant bits are ZERO, the BALF instruction will branch and the BALT instruction will not.

If any Memory Control function is specified in the MC field, the function is performed only if the branch is not taken. Similarly, if the Decode bit is set, the Decode function only occurs if no branch is taken.

The BALA and BALD instructions are used respectively to Arm and Disarm the interrupt system.

Execution Time

240 nanoseconds

5.3.2 Execute And Link

F ADDRESS (LINK),E,D,MC [ADDRESS LINK]

0	3	5	6	11	14	26	27	28 31
0 0 0	1 1	T	LINK	F	ADDRESS	E	D	MC

F (B)(NULL),E,D,MC [REGISTER LINK]

0	3	5	6	11	14	20	25	26	27	28 31
0 0 0	0 1	T	NULL	F	▨	B	▨	E	D	MC

where F =

	T	F
EXLZ	0	000
EXLL	0	001
EXLG	0	010
EXLF	0	011
EXLC	0	100
EXLV	0	101
EXL	0	110
EXLA	0	111
EXLNZ	1	000
EXLNL	1	001
EXLNG	1	010
EXLT	1	011
EXLNC	1	100
EXLNV	1	101
EXLD	1	111

The address of the next sequential micro-instruction replaces the contents of the register specified by LINK, then if the condition is met, the instruction at the specified address is executed. Any instruction may be executed including other Execute instructions. When the executed instruction is completed, the Processor continues with the micro-instruction following the Execute and Link.

Tested Condition True

(LINK) ← $(RLC_{4:15})+1$
Do instruction at ADDRESS [Address Link]
Do instruction at (B) [Register Link]
$(RLC_{4:15})$ ← $(RLC_{4:15})+1$

Tested Condition False

(LINK) ← $(RLC_{4:15})+1$
$(RLC_{4:15})$ ← $(RLC_{4:15})+1$

Programming Note

See Branch and Link

Execution Time

240 nanoseconds plus executed instruction

5.4 Shift/Rotate Instructions

The Shift and Rotate instructions provide for arithmetic and logical manipulation of information contained in the Processor registers. Bits shifted out of the high or low end of a register are passed through the Carry flag (C). After a shift instruction, the last bit which was shifted out is contained in the Carry flag.

A shift of zero positions causes the G and L flags to be set properly with no alteration to the data contained in the register. The Carry and Overflow flags are reset.

The instructions described in this section are:

5.4.1	SLL	Shift Left Logical
	SLLX	Shift Left Logical and Transfer
	SLLI	Shift Left Logical Immediate
5.4.2	SLHL	Shift Left Halfword Logical
5.4.3	SRL	Shift Right Logical
	SRLX	Shift Right Logical and Transfer
	SRLI	Shift Right Logical Immediate
5.4.4	SRHL	Shift Right Halfword Logical
5.4.5	SLA	Shift Left Arithmetic
	SLAX	Shift Left Arithmetic and Transfer
	SLAI	Shift Left Arithmetic Immediate
5.4.6	SLHA	Shift Left Halfword Arithmetic
5.4.7	SRA	Shift Right Arithmetic
	SRAX	Shift Right Arithmetic and Transfer
	SRAI	Shift Right Arithmetic Immediate
5.4.8	SRHA	Shift Right Halfword Arithmetic
5.4.9	RR	Rotate Right
	RRX	Rotate Right and Transfer
	RRI	Rotate Right Immediate
5.4.10	RL	Rotate Left
	RLX	Rotate Left and Transfer
	RLI	Rotate Left Immediate

5.4.1 Shift Left Logical

SLL S,A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
0 01	01	I	S	A	1001	B	O	E	D	MC

SLLX S,A,B,ADRS,I,C [RR TRANSFER]

0	3	5	6	11	16	20	25	26 31
0 0 1	0 0	I	S	A	1001	B	C	PAGE ADDRESS

SLLI S,A,DATA,I [RR IMMEDIATE]

0	3	5	6	11	16	20 31
0 0 1	10	I	S	A	1001	DATA

The contents of the register specified by A are shifted left the number of bit positions specified by the least significant 5 bits of the second operand. The result replaces the contents of the register specified by S.

High order bits shifted out of Position 0 are shifted through the Carry flag, then lost. Zeros shift into the low order bit position.

SLL,SLLI: $(S) \xleftarrow[B_E(27:31)]{L} (A)$

SLLX : $(S) \xleftarrow[B_E(27:31)]{L} (A)$

then $(RLC_{10:15}) \leftarrow$ PAGE ADDRESS if C = 0 or Carry = 0

$(RLC_{4:15}) \leftarrow (RLC_{4:15})+1$ if C = 1 and Carry = 1

Resulting flags

C	V	G	L	
	0	0	0	Result is zero
	0	0	1	Result is less than zero
	0	1	0	Result is greater than zero
0				Last bit shifted out was a zero
1				Last bit shifted out was a one

Execution Times (n = number of shifts)

SLL,SLLI	:360+60n
SLLX (No transfer)	:500+60n
SLLX (Transfer)	:360+60n

5.4.2 Shift Left Halfword Logical

SLHL S,A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
0 0 1	0 1	I	S	A	1 0 0 1	B	1	E	D	MC

The least significant 16-bits of the register specified by A are shifted left the number of bit positions specified by the least significant 4-bits of the second operand. The result replaces the least significant 16-bits of the register specified by S. The most significant 16-bits of the register specified by A replace the most significant 16-bits of the register specified by S. Bits shifted out of Position 16 are shifted through the Carry flag, then lost. Zeros shift into the low order bit position.

$$\text{SLHL:}\quad \begin{array}{l}(S_{0:15}) \longleftarrow (A_{0:15})\\ (S_{16:31}) \xleftarrow{L} (A_{16:31})\\ \qquad\quad B_E(28:31)\end{array}$$

Resulting flags

C	V	G	L	
	0	0	0	Halfword result is zero
	0	0	1	Halfword result is less than zero
	0	1	0	Halfword result is greater than zero
0				Last bit shifted out of bit-16 was a zero
1				Last bit shifted out of bit-16 was a one

Execution Times (n = number of shifts)

SLHL: 360+60n

5.4.3 Shift Right Logical

SRL S,A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
0 0 1	0 1	I	S	A	1 0 0 0	B	0	E	D	MC

SRLX S,A,B,ADRS,I,C [RR TRANSFER]

0	3	5	6	11	16	20	25	26 31
0 0 1	0 0	I	S	A	1 0 0 0	B	C	PAGE ADDRESS

SRLI S,A,DATA,I [RR IMMEDIATE]

0	3	5	6	11	16	20 31
0 0 1	10	I	S	A	1 0 0 0	DATA

The contents of the register specified by A are shifted right the number of bit positions specified by Bits 27:31 of the second operand. Low order bits shifted out of Position 31 are shifted through the Carry flag and then lost. Zeros shift into Position 0.

SRL,SRLI : $(S) \xleftarrow[B_E(27:31)]{R} (A)$

SRLX : $(S) \xleftarrow[B_E(27:31)]{R} (A)$

then $(RLC_{10:15}) \leftarrow$ PAGE ADDRESS if C=0 or Carry=0
$(RLC_{4:15}) \leftarrow (RLC_{4:15})+1$ if C=1 and Carry=1

Resulting flags

C	V	G	L	
	0	0	0	Result is zero
	0	0	1	Result is less than zero
	0	1	0	Result is greater than zero
0				Last bit shifted out was a zero
1				Last bit shifted out was a one

Execution Times (n = number of shifts)

SRL,SRLI	:	360+60n
SRLX (No transfer)	:	500+60n
SRLX (Transfer)	:	360+60n

5.4.4 Shift Right Halfword Logical

SRHL S,A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
001	01	I	S	A	1000	B	1	E	D	MC

The least significant 16-bits of the register specified by A are shifted right the number of bit positions specified by the least significant 4-bits of the second operand. The result replaces the least significant 16-bits of the register specified by S. The most significant 16-bits of the register specified by A replace the most significant 16-bits of the register specified by S. Bits shifted out of Position 31 are shifted through the Carry flag, then lost. Zeros shift into Position 16.

$$\text{SRHL} : (S_{0:15}) \leftarrow (A_{0:15})$$
$$(S_{16:31}) \xleftarrow[B_E(28:31)]{R} (A_{16:31})$$

Resulting flags

C	V	G	L	
	0	0	0	Halfword result is zero
	0	0	1	Halfword result is less than zero
	0	1	0	Halfword result is greater than zero
0				Last bit shifted out was a zero
1				Last bit shifted out was a one

Execution Times (n = number of shifts)

SRHL: 360+60n

5.4.5 Shift Left Arithmetic

SLA S,A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
0 0 1	0 1	I	S	A	1 1 0 1	B	0	E	D	MC

SLAX S,A,B,ADRS,I,C [RR TRANSFER]

0	3	5	6	11	16	20	25	26 31
0 0 1	0 0	I	S	A	1 1 0 1	B	C	PAGE ADDRESS

SLAI S,A,DATA,I [RR IMMEDIATE]

0	3	5	6	11	16	20 31
0 0 1	10	I	S	A	1 1 0 1	DATA

The contents of the register specified by A are shifted left the number of bit positions specified by the least significant 5-bits of the second operand. Only Bits 1:31 participate in the shift, Bit 0 remains unchanged. High order bits shifted out of Position 1 are shifted through the Carry flag, then lost. Zeros shift into the low order bit position.

SLA,SLAI : $(S_0) \leftarrow (A_0)$

$$(S_{1:31}) \xleftarrow[B_E(27:31)]{L} (A_{1:31})$$

SLAX : $(S_0) \leftarrow (A_0)$

$$(S_{1:31}) \xleftarrow[B_E(27:31)]{L} (A_{1:31})$$

then $(RLC_{10:15}) \leftarrow$ PAGE ADDRESS if C=0 or Carry=0

$(RLC_{4:15}) \leftarrow (RLC_{4:15})+$ if C=1 and Carry=1

Resulting flags

C	V	G	L	
	0	0	0	Result is zero
	0	0	1	Result is less than zero
	0	1	0	Result is greater than zero
0				Last bit shifted out was a zero
1				Last bit shifted out was a one

Execution Times (n = number of shifts)

SLA,SLAI	: 360+60n
SLAX (No transfer)	: 500+60n
SLAX (Transfer)	: 360+60n

5.4.6 Shift Left Halfword Arithmetic

SLHA S,A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
0 0 1	0 1	I	S	A	1 1 0 1	B	1	E	D	MC

The least significant 15-bits of the register specified by A are shifted left the number of bit positions specified by the least significant 4-bits of the second operand. The result replaces the least significant 15-bits of the register specified by S. The most significant 17-bits of the register specified by A replace the most significant 17-bits of the register specified by S. Bits shifted out of Position 17 are shifted through the Carry flag, then lost. Zeros shift into Position 31.

SLHA : $(S_{0:16}) \leftarrow (A_{0:16})$

$$(S_{17:31}) \xleftarrow[B_E(28:31)]{L} (A_{17:31})$$

Resulting flags

C	V	G	L	
	0	0	0	Halfword Result is zero
	0	0	1	Halfword Result is less than zero
	0	1	0	Halfword Result is greater than zero
0				Last bit shifted out of Position 17 was a zero
1				Last bit shifted out of Position 17 was a one

Execution Time (n = number of shifts)

SLHA : 360+60n

5.4.7 Shift Right Arithmetic

SRA S,A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
001	01	I	S	A	1100	B	0	E	D	MC

SRAX S,A,B,ADRS,I,C [RR TRANSFER]

0	3	5	6	11	16	20	25	26 31
001	00	I	S	A	1100	B	C	PAGE ADDRESS

SRAI S,A,DATA,I [RR IMMEDIATE]

0	3	5	6	11	16	20 31
001	10	I	S	A	1100	DATA

The contents of the register specified by A are shifted right the number of bit positions specified by the least significant 5-bits of the second operand. The result replaces the contents of the register specified by S. Only Bits 1:31 participate in the shift; Bit 0 remains unchanged and is propagated right into Position 1 on each shift. Low order bits shift through the Carry flag and are then lost.

SRA,SRAI : $(S_0) \leftarrow (A_0)$

$$(S_{1:31}) \xleftarrow[B_E(27:31)]{R} (A_{1:31})$$

SRAX : $(S_0) \leftarrow (A_0)$

$$(S_{1:31}) \xleftarrow[B_E(27:31)]{R} (A_{1:31})$$

then $(RLC_{10:15}) \leftarrow$ PAGE ADDRESS if C = 0 or Carry = 0
$(RLC_{4:15}) \leftarrow (RLC_{4:15})+1$ if C = 1 and Carry = 1

Resulting flags

C	V	G	L	
	0	0	0	Result is zero
	0	0	1	Result is less than zero
	0	1	0	Result is greater than zero
0				Last bit shifted out was a zero
1				Last bit shifted out was a one

Execution Times (n = number of shifts)

SRA,SRAI : 360+60n
SRAX (No transfer) : 500+60n
SRAX (Transfer) : 360+60n

5.4.8 Shift Right Halfword Arithmetic

SRHA S,A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
001	01	I	S	A	1100	B	1	E	D	MC

The least significant 15-bits of the register specified by A are shifted right the number of bit positions specified by the least significant 4-bits of the second operand. The result replaces the least significant 15-bits of the register specified by S. The most significant 17-bits of the register specified by A replace the most significant 17-bits of the register specified by S. Bit 16 is propogated right into bit Position 15 on each shift. Bits shifted out of Position 31 are shifted through the Carry flag and then lost.

SRHA: $(S_{0:16}) \leftarrow (A_{0:16})$
$(S_{17:31}) \xleftarrow[B_E(28:31)]{R} (A_{17:31})$

Resulting flags

C	V	G	L	
	0	0	0	Halfword result is zero
	0	0	1	Halfword result is less than zero
	0	1	0	Halfword result is greater than zero
0				Last bit shifted out was a zero
1				Last bit shifted out was a one

Execution Times (n = number of shifts)

SRHA : 360+60n

5.4.9 Rotate Left

RL S,A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
0 0 1	0 1	I	S	A	1 0 1 1	B	0	E	D	MC

RLX S,A,B,ADRS,I,C [RR CONTROL]

0	3	5	6	11	16	20	25	26 31
0 0 1	0 0	I	S	A	1 0 1 1	B	C	PAGE ADDRESS

RLI S,A,DATA,I [RR IMMEDIATE]

0	3	5	6	11	16	20 31
0 0 1	1 0	I	S	A	1 0 1 1	DATA

The contents of the register specified by A are shifted left, end around, the number of bit positions specified by the least significant 5-bits of the second operand. Bits shifted out of Position 0 are shifted into Position 31.

RL,RLI : $(S_{0:31}) \xleftarrow[B_E(27:31)]{L} (A_{0:31})$

RLX : $(S_{0:31}) \xleftarrow[B_E(27:31)]{L} (A_{0:31})$

:then $(RLC_{10:15}) \leftarrow$ PAGE ADDRESS

Resulting flags

C	V	G	L	
0	0	0	0	Result is zero
0	0	0	1	Result is not zero
0	0	1	0	

Execution Times (n = number of shifts)

RL,RLI,RLX: 360+60n

5.4.10 Rotate Right

RR S,A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
0 0 1	0 1	I	S	A	1 0 1 0	B	0	E	D	MC

RRX S,A,B,ADRS,I,C [RR TRANSFER]

0	3	5	6	11	16	20	25	26 31
0 0 1	0 0	I	S	A	1010	B	C	PAGE ADDRESS

RRI S,A,DATA,I [RR IMMEDIATE]

0	3	5	6	11	16	20 31
0 0 1	10	I	S	A	1 0 1 0	DATA

The contents of the register specified by A are shifted right, end around, the number of bit positions specified by the least significant 5-bits of the second operand. Bits shifted out of Position 31 are shifted into position 0.

RR,RRI : $(S_{0:31}) \xleftarrow[B_E(27:31)]{R} (A_{0:31})$

RRX : $(S_{0:31}) \xleftarrow[B_E(27:31)]{R} (A_{0:31})$

then $(RLC_{10:15}) \leftarrow$ PAGE ADDRESS

Resulting flags

C	V	G	L	
0	0	0	0	Result is zero
0	0	0	1	Result is not zero
0	0	1	0	

Execution Times (n = number of shifts)

RR,RRI,RRX: 360+60n

5.5 Fixed Point Arithmetic Instructions

The Fixed Point Arithmetic Instructions provide for addition, subtraction, multiplication and division of fixed point data contained in the Processor registers. The instructions described in this section are:

5.5.1	A	Add
	AX	Add and Transfer
	AI	Add Immediate
5.5.2	AINC	Add and Increment
	AINCX	Add and Increment and Transfer
5.5.3	S	Subtract
	SX	Subtract and Transfer
	SI	Subtract Immediate
5.5.4	SDEC	Subtract and Decrement
	SDECX	Subtract and Decrement and Transfer
5.5.5	M	Multiply
	MX	Multiply and Transfer
	MI	Multiply Immediate
5.5.6	D	Divide
	DX	Divide and Transfer
	DI	Divide Immediate

5.5.1 Add

A S,A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
001	01	I	S	A	0001	B	0	E	D	MC

AX S,A,B,ADRS,I,C [RR TRANSFER]

0	3	5	6	11	16	20	25	26 31
0 01	00	I	S	A	0001	B	C	PAGE ADDRESS

AI S,A,DATA,I [RR IMMEDIATE]

0	3	5	6	11	16	20 31
0 0 1	10	I	S	A	0001	DATA

The second operand is algebraically added to the first operand. The sum replaces the contents of the register specified by S.

A,AI : (S) ← (A) + B_E
AX : (S) ← (A) + B_E
then ($RLC_{10:15}$) ← PAGE ADDRESS if
C = 0 or Carry = 0
($RLC_{4:15}$) ← ($RLC_{4:15}$) + 1 if C = 1 and
Carry = 1

Resulting flags:

C	V	G	L	
		0	0	Sum is zero
		0	1	Sum is less than zero
		1	0	Sum is greater than zero
	1			Overflow
1				Carry

Execution Times:

A,AI	:	240
AX (No transfer)	:	380
AX (Transfer)	:	240

5.5.2 Add and Increment

AINC S,A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
0 0 1	0 1	I	S	A	0 0 1 1	B	0	E	D	MC

AINCX S,A,B,ADRS,I [RR TRANSFER]

0	3	5	6	11	16	20	25	26 31
0 0 1	0 0	I	S	A	0 0 1 1	B	C	PAGE ADDRESS

The second operand is algebraically added with the first operand and a forced carry in of one. The sum replaces the contents of the register specified by S.

AINC : $(S) \leftarrow (A) + B_E + 1$
AINCX : $(S) \leftarrow (A) + B_E + 1$
then $(RLC_{10:15}) \leftarrow$ PAGE ADDRESS if C = 0 or Carry = 0
$(RLC_{4:15}) \leftarrow (RLC_{4:15}) + 1$ if C = 1 and Carry = 1.

Resulting flags

C	V	G	L	
		0	0	Sum is zero
		0	1	Sum is less than zero
		1	0	Sum is greater than zero
	1			Overflow
1				Carry

Programming Note

Multiple precision addition operations require a carry forward from the least significant to the most significant operands. The following example shows a double word add operation.

```
*
* MR0 AND MR1 CONTAIN THE 64 BIT FIRST OPERAND
* MR2 AND MR3 CONTAIN THE 64 BIT SECOND OPERAND
* THE 64 BIT RESULT IS RETURNED IN MR0 and MR1
*

START   AX      MR1,MR1,MR3,SUM2,C      SUM LOW OPERANDS FIRST
*                                       TRANSFER IF NO CARRY, ELSE
*                                       FALL THROUGH, SUMMING
        AINCX   MR0,MR0,MR2,SUM3        HIGH OPERANDS THEN ADD ONE,
                                        SKIP TO SUM3.
*
SUM2    A       MR0,MR0,MR2             SUM HIGH OPERANDS
SUM3                                    MR0,MR1 = 64-BIT RESULT
```

Execution Times

AINC : 240
AINCX (No transfer): 380
AINCX (Transfer) : 240

5.5.3 Subtract

S S,A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
0 0 1	0 1	I	S	A	0 0 0 0	B	0	E	D	MC

SX S,A,B,ADRS,I,C [RR TRANSFER]

0	3	5	6	11	16	20	25	26 31
0 0 1	0 0	I	S	A	0 0 0 0	B	C	PAGE ADDRESS

SI S,A,DATA,I [RR IMMEDIATE]

0	3	5	6	11	16	20 31
0 0 1	1 0	I	S	A	0 0 0 0	DATA

The second operand is algebraically subtracted from the first operand. The difference replaces the contents of the register specified by S.

S,SI : $(S) \leftarrow (A) - B_E$
SX : $(S) \leftarrow (A) - B_E$
then $(RLC_{10:15}) \leftarrow$ PAGE ADDRESS if C = 0 or Carry = 0
$(RLC_{4:15}) \leftarrow (RLC_{4:15}) + 1$ if C = 1 and Carry = 1

Resulting flags

C	V	G	L	
		0	0	Difference is zero
		0	1	Difference is less than zero
		1	0	Difference is greater than zero
	1			Overflow
1				Borrow

Execution times

S,SI	: 240
SX (No Transfer)	: 380
SX (Transfer)	: 240

5.5.4 Subtract and Decrement

SDEC S,A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
0 0 1	0 1	I	S	A	0 0 1 0	B	0	E	D	MC

SDECX S,A,B,ADRS,I,C [RR TRANSFER]

0	3	5	6	11	16	20	25	26 31
0 0 1	0 0	I	S	A	0 0 1 0	B	C	PAGE ADDRESS

The second operand and a forced carry in of one are subtracted from the first operand. The result replaces the contents of the register specified by S.

SDEC : $(S) \leftarrow (A) - B_E - 1$
SDECX : $(S) \leftarrow (A) - B_E - 1$
then $(RLC_{10:15}) \leftarrow$ PAGE ADDRESS if $C = 0$ or $Carry = 0$
$(RLC_{4:15}) \leftarrow (RLC_{4:15}) + 1$ if $C = 1$ and $Carry = 1$

Resulting flags

C	V	G	L	
		0	0	Difference is zero
		0	1	Difference is less than zero
		1	0	Difference is greater than zero
	1			Overflow
1				Carry

Programming Note

See Add and Increment

Execution Times

SDEC	: 240
SDECX (No Transfer)	: 380
SDECX (Transfer)	: 240

5.5.5 Multiply

M S,A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
001	01	I	S	A	1110	B	0	E	D	MC

MX S,A,B,ADRS,I,C [RR TRANSFER]

0	3	5	6	11	16	20	25	26 31
0 01	0 0	I	S	A	1110	B	C	PAGE ADDRESS

MI S,A,DATA,I [RR IMMEDIATE]

0	3	5	6	11	16	20 31
001	1 0	I	S	A	1110	DATA

The 32-bit second operand is multiplied by the contents of the first operand register. The 32 most significant product bits replace the contents of the register specified by S. The 32 least significant product bits replace the contents of the first operand register, the register specified by A. The S field must specify an even numbered register. The A field must specify the next sequential register, an odd number. The sign of the product is determined by the rules of algebra.

M,MI : $(S,A) \leftarrow (A) * B_E$
MX : $(S,A) \leftarrow (A) * B_E$
then $(RLC_{10:15}) \leftarrow$ PAGE ADDRESS

Resulting flags

C	V	G	L
0	0	0	0

Execution Times

M,MI,MX: 2580/3480/4440 minimum/average/maximum

5.5.6 Divide

D S,A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
001	01	I	S	A	1111	B	0	E	D	MC

DX S,A,B,ADRS,I,C [RR TRANSFER]

0	3	5	6	11	16	20	25	26 31
001	00	I	S	A	1111	B	C	PAGE ADDRESS

DI S,A,DATA,I [RR IMMEDIATE]

0	3	5	6	11	16	20 31
001	10	I	S	A	1111	DATA

The 32-bit second operand is divided into the 64-bit dividend contained in the registers specified by S and A, an even/odd pair. The S field must specify an even numbered register and the A field must specify the next sequential odd register. The resulting 31-bit quotient with sign is contained in A and the 31-bit remainder with sign is contained in S. The sign of the quotient is determined by the rules of algebra; the sign of the remainder equals the sign of the dividend.

D,DI : $(A) \leftarrow (S,A)/B_E$
$(S) \leftarrow$ Remainder

DX : $(A) \leftarrow (S,A)/B_E$
$(S) \leftarrow$ Remainder
then $(RLC_{10:15}) \leftarrow$ PAGE ADDRESS

Resulting flags

C	V	G	L	
0	0	0	0	Normal
0	1	0	0	Divide fault

Programming Note

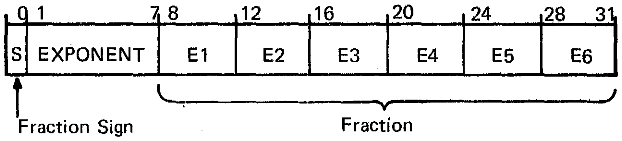
A quotient that would be greater than '7FFF FFFF' or less than '8000 0000' causes the division to be aborted with the V flag set and an unpredictable remainder in S. The register specified by A is unchanged.

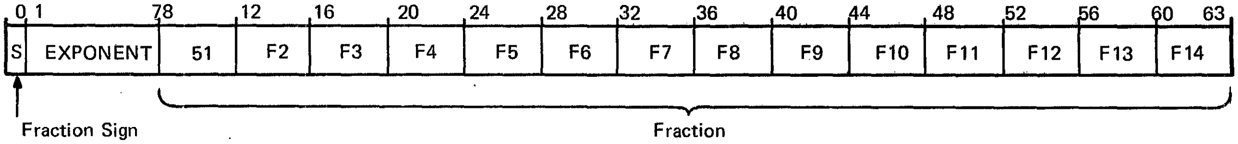
Execution Times

D,DI,DX : 4560

5.6 Floating Point Instructions

These instructions provide for the manipulation of single precision floating point data and double precision floating point data. A floating point quantity consists of a signed exponent and a signed magnitude fraction. The 7-bit exponent is expressed in excess 64 notation and can range in actual value from +63 through zero to -64. The value of the exponent field is that power of 16 that the fraction field is to be multiplied by. The 24 or 56 bit fraction is expressed as a hexadecimal number having a radix point to the left of the high order fraction digit. Bit 0 of the fullword or double word is the sign bit of the fraction.





The standard Model 8/32 has single precision floating point arithmetic capability included in the Arithmetic Logic Unit (ALU). The micro instructions associated with this feature of the ALU provide for the manipulation of single precision floating-point data. When the single precision floating point unit is addressed (Module Number 3), references to the user's General Registers, either directly or by way of YD or YS, select the corresponding single precision floating point registers instead. These registers are shown on the block diagram, Figure 1, as ER0 through ERF. There are 16 32-bit floating point registers, although in the user level architecture only the even numbered registers are actually used.

The optional precision floating point ALU (DFU), which is not shown on the block diagram, is also capable of performing single precision floating point operations with greater accuracy than the standard single precision floating point unit. When the DFU is installed, all of the existing single precision only floating point capability remains intact, including the 16 floating point registers. Thus the microcode can exclusively use the DFU or mix operations with the single precision unit.

The double precision floating point unit will be discussed in more detail later. First, the instructions associated with the single precision only floating point unit, and described in this section are:

5.6.1	CE	Compare
5.6.2	CEQ	Compare and Equalize
	CEQX	Compare and Equalize and Transfer
5.6.3	AE	Add
5.6.4	SE	Subtract
5.6.5	AU	Add Unnormalized
5.6.6	ME	Multiply
5.6.7	DE	Divide

5.6.1 Compare

CE S,A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
011	01	I	S	A	0101	B	0	E	D	MC

The first operand is compared to the second operand. The comparison is algebraic, taking into account the sign, exponent and fraction. The result is indicated by the resultant flags. The S Bus should be NULL selected.

CE (A):B_E

Resulting flags

C	V	G	L	
0	0	0	0	First operand equal to second operand
1	0	0	1	First operand less than second operand
0	0	1	0	First operand greater than second operand

Execution Times

CE: 240

5.6.2 Compare and Equalize

CEQ S,A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
0 1 1	0 1	I	S	A	1 1 0 0	B	0	E	D	MC

CEQX S,A,B,ADRS,I,C [RR TRANSFER]

0	3	5	6	11	16	20	25	26 31
0 1 1	0 0	I	S	A	1 1 0 0	B	C	PAGE ADDRESS

The first operand is compared to the second operand. The comparison is magnitude only—the sign bits of the two operands are ignored. The result of this comparison is indicated by the resultant flags.

The fraction field of the smaller of the two operands is shifted right hexadecimally (4-bits at a time) the number of times indicated by the exponent difference of the two operands. The result fraction replaces the contents of the register specified by S.

The affect of this instruction is to unnormalize the smaller number so that the radix points of the two arguments are aligned.

In the RR Transfer format, if the C bit is set, a transfer occurs if the second operand is smaller than the first operand.

CEQ: if $(A_{1:31}) \leq B_E\,(1{:}31)$,
$(S) \leftarrow (A_{8:31})$
$X \leftarrow B_E(1{:}7) - (A_{1:7})$

if $(A_{1:31}) > B_E\,(1{:}31)$,
$(S) \leftarrow (B_E(8{:}31)$
$X \leftarrow (A_{1:7}) - B_E\,(1{:}7)$

then $(S) \xleftarrow[4X]{R} (S)$

CEQX: if $(A_{1:31}) \leq B_E\,(1{:}31)$,
$(S) \leftarrow (A_{8:31})$
$X \leftarrow B_E\,(1{:}7) - (A_{1:7})$

if $(A_{1:31}) > B_E\,(1{:}31)$,
$(S) \leftarrow B_E\,(8{:}31)$
$X \leftarrow (A_{1:7}) - B_E\,(1{:}7)$

then $(S) \xleftarrow[4X]{R} (S)$

if $(A_{1:31}) > B_E(1{:}31)$ or $C = 0$
$(RLC_{10:15}) \leftarrow$ PAGE ADDRESS

if $(A_{1:31}) \leqslant B_E(1{:}31)$ and $C = 1$
$(RLC_{4:15}) \leftarrow (RLC_{4:15})+1$

Resulting flags

C	V	G	L	
0	0	0	0	First operand equal to second operand
1	0	0	1	First operand less than second operand
0	0	1	0	First operand greater than second operand

Programming Note

This instruction must precede the floating point Add or Subtract instructions to properly set-up the second operand.

Execution Time (E = number of equalizing shifts – maximum = 5)

CEQ:	480+60E
CEQX (no transfer):	660+60E
CEQ (transfer):	480+60E

5.6.3 Add

AE S,A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
0 1 1	0 1	I	S	A	1 0 0 1	B	0	E	D	MC

Proper execution of this instruction requires that a Compare and Equalize instruction has been performed on the two arguments and that the result of the Compare and Equalize–that is, the smaller of the two arguments–is on the B Bus.

The second operand fraction is algebraically added to the first operand fraction. The result replaces the contents of the register specified by S.

If the addition of fractions produces a carry, the result fraction is shifted right one hexadecimal position and the result exponent is incremented by one. If no carry was produced, the result fraction is normalized if necessary; the result exponent is decremented by one for each normalization cycle required.

AE: (S) ←— (A)+B_E

Resulting flags

C	V	G	L	
0	0	0	0	Result is zero
0	X	0	1	Result is less than zero
0	X	1	0	Result is greater than zero
0	1	X	X	Exponent overflow
0	1	0	0	Exponent underflow

Programming Notes

In the event of exponent overflow, the result flags show the proper sign for the result, but the data returned in the register specified by S is unpredictable. The microprogram should detect the exponent overflow case and then OR a value of '7FFFFFFF' into the register specified by S.

In the event of exponent underflow, the entire result is set to zero.

The following program sequence shows the correct way to Add two floating-point numbers.

```
* THE FLOATING POINT REGISTER SPECIFIED BY YD
* CONTAINS THE FIRST OPERAND.
* THE FLOATING POINT REGISTER SPECIFIED BY YS
* CONTAINS THE SECOND OPERAND.
*
*
AER     CEQX    MR0,YD,YS,AER1,C        SMALLER OPERAND TO MR0
*                                       TRANSFER IF YS CONTAINED
*                                       THE SMALLER OPERAND
*                                       FALL THROUGH IF YD WAS
*                                       EQUAL TO OR LARGER
*                                       THAN YS.
        AE      YD,YS,MR0,E             ADD SMALLER TO LARGER,
*                                       RESULT TO YD & SET CC
        BALV    FAULT (NULL),ILIR,D     TO FAULT IF V FLAG, ELSE
*                                       INCR. LOC, FETCH & DECODE
*                                       NEXT USER INSTR.
AER1    AE      YD,YD,MR0,E             ADD SMALLER TO LARGER
*                                       RESULT TO YD & SET CC
        BALV    FAULT (NULL),ILIR,D
*
```

Execution Times (n = number of normalize cycles)

AE: 360+60 n

5.6.4 Subtract

SE S,A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
0 1 1	0 1	I	S	A	1 0 0 0	B	0	E	D	MC

Proper execution of this instruction requires that a Compare and Equalize instruction has been performed on the two operands and that the result of the Compare and Equalize–that is, the smaller of the two operands–is on the B Bus.

The second operand fraction is algebraically subtracted from the fraction of the first operand. The result replaces the contents of the register specified by S.

If the subtraction of fractions produces a carry, the result fraction is shifted right one hexadecimal position and the result exponent is incremented by one. If no carry was produced, the result fraction is normalized if necessary; the result exponent is decremented by one for each normalization cycle required.

SE : (S) ← (A)–B_E

Resulting flags

C	V	G	L	
0	0	0	0	Result is zero
0	X	0	1	Result is less than zero
0	X	1	0	Result is greater than zero
0	1	X	X	Exponent Overflow
0	1	0	0	Exponent Underflow

Programming Note

See Add

Execution Times (n = number of normalize cycles)

SE : 360+60n

5.6.5 Add Unnormalized

AU S,A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
0 1 1	0 1	I	S	A	0 0 0 1	B	0	E	D	MC

The second operand is added to the first operand and the unnormalized result replaces the contents of the register specified by S. For this instruction, both arguments are assumed to be fixed-point 32-bit quantities. No Compare and Equalize instruction is needed.

AU: (S) ← (A) + B_E

Resulting flags

C	V	G	L	
0	0	0	0	Result is zero
0	0	0	1	Result is less than zero
0	0	1	0	Result is greater than zero

Programming Note

This instruction is included to provide a means to access the floating point registers without using the floating point arithmetic instructions.

Execution Time

AU : 240

5.6.6 Multiply

ME S,A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
0 1 1	0 1	I	S	A	1 1 1 0	B	0	E	D	MC

The exponents of the first and second operands are added and set aside as the exponent of the final result. The result sign is determined by the rules of algebra. The fractions of the first and second operands are then multiplied. If the product is zero, the entire result, sign and exponent included, is set to zero. If non-zero, the product is normalized or corrected as necessary. The sign, exponent and result fraction are combined and replace the contents of the register specified by S.

ME: (S) ← (A)*B_E

Resulting flags

C	V	G	L	
0	0	0	0	Result is zero
0	X	0	1	Result is less than zero
0	X	1	0	Result is greater than zero
0	1	X	X	Exponent Overflow
0	1	0	0	Exponent Underflow

Programming Notes

In the event of exponent underflow, if the result of the multiplication of fractions is not zero, it is forced to zero and the result exponent and sign are cleared.

In the event of exponent overflow, the result in the register specified by S has the proper sign bit, but the exponent and fraction are not predictable. The microprogram should detect the exponent overflow situation and OR a value of '7FFFFFFF' into the register specified by S.

Execution Times

ME: 1860/2580/3300 minimum/average/maximum

5.6.7 Divide

DE S,A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
0 1 1	0 1	I	S	A	1 1 1 1	B	0	E	D	MC

The exponents of the first and second operands are subtracted and the result set aside as the exponent of the final result. The result sign is determined by the rules of algebra. The second operand divisor is divided into the first operand dividend. If the result fraction is zero, the entire result is set to zero. If the result fraction is non-zero, it is normalized or corrected as necessary. The sign, exponent, and result fraction are combined and replace the contents of the register specified by S.

DE: $(S) \leftarrow (A)/B_E$

Resulting flags

C	V	G	L	
0	0	0	0	Result is zero
0	X	0	1	Result is less than zero
0	X	1	0	Result is greater than zero
	1	X	X	Exponent overflow
	1	0	0	Exponent underflow

Programming Notes

The hardware does not check for attempted division by zero. The microprogram must detect a zero divisor before doing the divide.

See Multiply.

Execution Times

DE : 3480/3540 Best/Worst

The DFU is a standard plug-in module to the 8/32. A unique set of 24 microinstructions is provided to access the DFU (Module Number 6).

Figure 3 shows a block diagram of the DFU. The DFU is situated between the 32-bit S Bus and the 32-bit B Bus. The A Bus does not connect to the DFU. The DFU contains its own set of eight 32-bit single precision registers and eight 64-bit double precision registers.

In microinstructions directed to the DFU, references to the user's General Registers, either directly or by way of the YD or YS fields of the user's instruction, cause the corresponding single-precision floating point register or half (32-bits) of a double precision floating point register to be accessed. The microinstruction itself distinguishes whether a single precision or a double precision operation is to be performed. Single precision operations can only use the single precision registers and double precision operations can only use the double precision registers.

The two halves of a double precision register are selected on an even/odd addressing scheme. For example, accessing double register 2 selects the most significant 32 bits of double register 2, and accessing double register 3 selects the least significant 32 bits of double register 2.

Note that the single precision registers in the DFU are separate and distinct from the single precision registers in the basic processor.

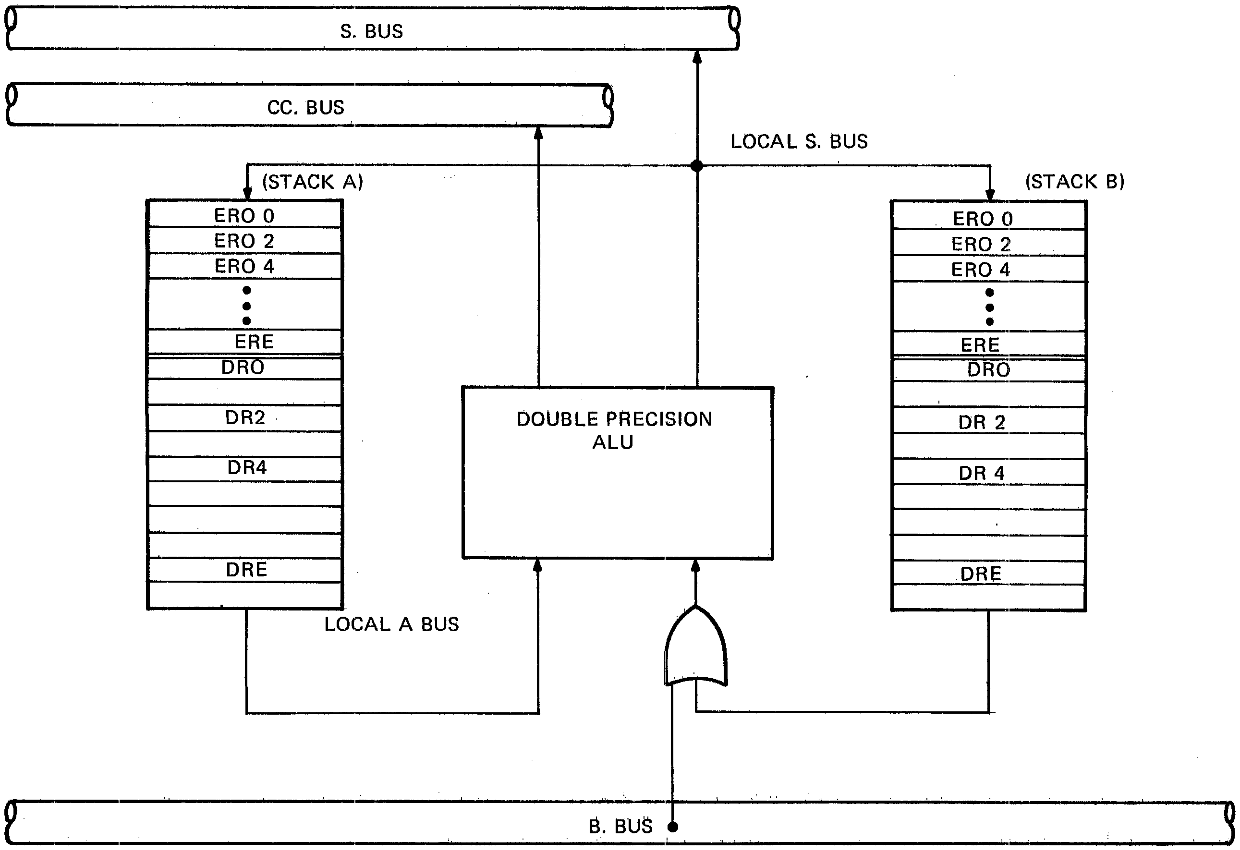


Figure 3. Double Precision Floating Point Unit Block Diagram

The floating point ALU operates in a fully autonomous mode having its own internal A, B, and S Busses. Given a load, add, subtract, multiply, or divide operation, the floating point module performs the function asynchronous of other Processor activity. The microprogram is free to do other microinstructions while the floating point module is finishing its task. If the microprogram attempts to test the result of an operation or begin another floating point operation before the last one is completed, the processor stops until the old function is completed before starting the next.

This feature has an impact on determination of execution times. The execution time shown for Divide Single Precision, for example, is 2920 nanoseconds. In practice, assuming the floating point module is not busy, the microinstruction that initiates the divide takes only 240 nanoseconds. The processor immediately begins the next sequential microinstruction. The floating point module is working on its own and will be busy for the next 2680 nanoseconds (2920-240). If this microinstruction does not reference the floating point module, the instruction is performed and the next microinstruction is begun. This continues until a microinstruction is fetched that does access the floating point module. At that time, the processor must wait out any of the divide execution time remaining – 2680 nanoseconds minus the execution time of all intervening microinstructions. If there was any time left at all, an additional 60 nanoseconds has to be added in for resynchronization.

The instructions described in this section are:

5.6.8	RCC	Read Condition Code
5.6.9	LE	Load Register Single Precision
	LEX	Load Register Single Precision and Transfer
	LEI	Load Register Single Precision Immediate
5.6.10	RRE	Read Register Single Precision
	RREX	Read Register Single Precision and Transfer
5.6.11	CER	Compare Single Precision
5.6.12	AER	Add Single Precision
5.6.13	SER	Subtract Single Precision
5.6.14	MER	Multiply Single Precision
5.6.15	DER	Divide Single Precision
5.6.16	LW	Load Unnormalized Double Precision
	LWX	Load Unnormalized Double Precision and Transfer
	LWI	Load Unnormalized Double Precision Immediate
5.6.17	LD	Load Register Double Precision
	LDX	Load Register Double Precision and Transfer
	LDI	Load Register Double Precision Immediate
5.6.18	RRD	Read Register Double Precision
	RRDX	Read Register Double Precision and Transfer
5.6.19	CDR	Compare Double Precision
5.6.20	ADR	Add Double Precision
5.6.21	SDR	Subtract Double Precision
5.6.22	MDR	Multiply Double Precision
5.6.23	DDR	Divide Double Precision

5.6.8 Read Condition Code

RCC S,B [,I,E,D,MC] [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
1 1 0	0 1	I	S	1 1 1 1 1	0 0 0 0	B	0	E	D	MC

This instruction is used to collect the flags that resulted from the last single precision or double precision floating point operation.

RCC: CCBUS ← Floating Point Flags

Resulting Flags

Determined by previous floating point operation

Programming Notes:

The S and B fields are not used and should be NULL selected.

Execution Times

240

5.6.9 Load Register Single Precision

LE A,B,I,K,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
1 1 0	0 1	I	1 1 1 1 1	A	0 0 1 0	B	K	E	D	MC

LEX A,B,ADRS,I,C [RR TRANSFER]

0	3	5	6	11	16	20	25	26 31
1 1 0	0 0	I	1 1 1 1 1	A	0 0 1 0	B	C	PAGE ADRS

LEI A,DATA,I [RR IMMEDIATE]

0	3	5	6	11	16	20 31
1 1 0	1 0	I	1 1 1 1 1	A	0 0 1 0	DATA

The second operand is normalized, if necessary and then loaded into the floating point register specified by A. Normalization involves shifting the fraction field left four bits at a time until the most significant four fraction bits are not zero. For each four place shift required, the exponent is decremented by one.

For the RR Control format, setting the K bit causes normalization to be avoided. The second operand is copied directly into the floating point register specified by A with no modification.

LE,LEI : $(A) \leftarrow B_E$
LEX : $(A) \leftarrow B_E$
then $(RLC_{10:15}) \leftarrow$ ADRS

Resulting flags (Second operand also a floating point register)

Not Meaningful

Resulting flags (Second operand not a floating point register)

C	V	G	L	
0	0	0	1	Result is less than zero
0	0	1	0	Result is greater than zero
0	1	X	X	Fraction was not normalized

Programming Notes

A Load microinstruction where both first and second operands are floating point registers does not produce meaningful flags. A Read Condition Code microinstruction is required if the flags need to be known. If the second operand is not from a floating point register, if the V flag is reset, the argument was already normalized. In this case the G and L flags are properly set. If the V flag does set, the argument fraction was not normalized. An argument of zero is considered to be an un-normalized quantity for this test.

The normalization process continues autonomous of other processor activity. When finished, a Read Condition Code microinstruction will collect the final flags.

Resulting flags (After Read Condition Code)

C	V	G	L	
0	0	0	0	Result is zero
0	0	0	1	Result is less than zero
0	0	1	0	Result is greater than zero
0	1	0	0	Exponent underflow

The A field must only specify a floating point register.

Execution Times

LE,LEX,LEI : 240+100n

5.6.10 Read Register Single Precision

RRE S,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
1 1 0	0 1	I	S	1 1 1 1 1	0 0 0 1	B	0	E	D	MC

RREX S,B,ADRS,I,C [RR TRANSFER]

0	3	5	6	11	16	20	25	26 31
1 1 0	0 0	I	S	1 1 1 1 1	0 0 0 1	B	C	PAGE ADRS

The contents of the floating point register specified by B are copied to the register specified by S.

RRE : $(S) \leftarrow (B)$
RREX : $(S) \leftarrow (B)$
then $(RLC_{10:15}) \leftarrow ADRS$

Resulting Flags

Not Meaningful

Programming Notes

Floating point register selection is not affected by the least significant B address bit. If an odd numbered register is specified, the next lower even numbered register is selected instead.

The S field may specify any register other than a floating point register.

Execution Times

RRE,RREX : 300

5.6.11 Compare Single Precision

CER A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
1 1 0	0 1	1	1 1 1 1 1	A	0 0 1 1	B	0	E	D	MC

The first operand is compared to the second operand. The comparison is algebraic, taking into account the sign, exponent, and fraction. The result is indicated by the resulting flags.

CER: (A): B_E

Resulting Flags after RCC

C	V	G	L	
0	0	0	0	First operand equal to second operand
1	0	0	1	First operand less than second operand
0	0	1	0	First operand greater than second operand

Execution Times

CER: 400

5.6.12 Add Single Precision

AER A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
1 1 0	0 1	I	1 1 1 1 1	A	0 1 0 0	B	0	E	D	MC

The first operand is compared to the second operand. The comparison is in magnitude only, ignoring the signs of the two operands. The fraction field of the smaller of the two operands is shifted right hexadecimally (4-Bits at a time) the number of times indicated by the difference of the exponents of the two operands. This is called equalization. Note that because the DFU has internal 56 bit arithmetic capability, hexadecimal digits shifted out of the low order end of the 24 bit fraction field are not lost. In practice, only the last digit shifted out is saved. This digit is called a "guard" digit and is provided for extended accuracy.

The affect of this process is to unnormalize the smaller operand enough so that the radix points of the two arguments are aligned. If the exponent difference exceeds six, the smaller operand loses significance and a value of zero is substituted.

The equalized fraction with its guard digit and the fraction of the other operand with trailing zeros are then algebraically added, taking into account the signs and order of the two operands. The result fraction with an exponent equal to that of the larger operand replaces the contents of the floating point register specified by A.

If the addition of fractions produces a carry, the result fraction is shifted right one hexadecimal position and the result exponent is incremented by one. If no carry was produced, the result fraction is normalized if necessary. The result exponent is decremented by one for each normalization cycle required.

AER: $(A) \leftarrow (A)+B_E$

Resulting Flags

C	V	G	L	
0	0	0	0	Result is Zero
0	0	0	1	Result is less than Zero
0	0	1	0	Result is greater than Zero
0	1	0	1	Exponent Overflow
0	1	1	0	
0	1	0	0	Exponent Underflow

Programming Notes

In the event of exponent overflow, the proper sign bit is generated and the rest of the result is set to all ones ± 7FFFFFFF .

In the event of exponent underflow, the entire result is set to zero.

Execution Times

AER: 750 + 100 (e+n) e = Equalize Cycles
n = Normalize Cycles

worst case total of e+n = 6

5.6.13 Subtract Single Precision [RR CONTROL]

SER A,B,I,E,D,MC

0	3	5	6	11	16	20	25	26	27	28 31
1 1 0	0 1	I	1 1 1 1 1	A	0 1 0 1	B	0	E	0	MC

The first operand is compared to the second operand. The comparison is in magnitude only, ignoring the signs of the two operands. The fraction field of the smaller of the two operands is shifted right hexadecimally (4-Bits at a time) the number of times indicated by the difference of the exponents of the two operands. Note that because the DFU has internal 56 bit arithmetic capability, hexadecimal digits shifted out of the low order end of the 24 bit fraction field are not lost. In practice, only the last digit shifted out is saved. This digit is called a "guard" digit and is provided for extended accuracy.

The affect of this process is to unnormalize the smaller operand enough so that the radix points of the two arguments are aligned. If the exponent difference exceeds five, the smaller argument loses significance and a value of zero is substituted.

The equalized fraction with its guard digit and the fraction of the other operand with trailing zeros are then algebraically subtracted, taking into account the signs and order of the two operands. The result fraction with an exponent equal to that of the larger operand replaces the contents of the floating point register specified by A.

If the subtraction of fractions produces a carry, the result fraction is shifted right one hexadecimal position and the result exponent is incremented by one. If no carry was produced, the result fraction is normalized if necessary. The result exponent is decremented by one for each normalization cycle required.

SER: $(A) \leftarrow (A)+B_E$

Resulting Flags

C	V	G	L	
0	0	0	0	Result is zero
0	0	0	1	Result is less than zero
0	0	1	0	Result is greater than zero
0	1	0	1	Exponent Overflow
0	1	1	0	
0	1	0	0	Exponent Underflow

Programming Notes

See Add Single Precision

Execution Times

SER: 750+100(e+n)

e = Equalize Cycles
n = Normalize Cycles

worst case total of e+n = 6

5.6.14 Multiply Single Precision

MER A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
1 1 0	0 1	I	1 1 1 1 1	A	0 1 1 0	B	0	E	D	MC

The exponents of the first and second operands are added then set aside as the exponent of the final result. The result sign is determined by the rules of algebra. The fractions of the first and second operands are then multiplied. If the product is zero, the entire result, sign and exponent included, is set to zero. If the product is non-zero, it is normalized or corrected as necessary. The sign, exponent, and result fraction are combined and replace the contents of the floating point register specified by A.

MER: $(A) \leftarrow (A)^{*} B_E$

Resulting Flags

C	V	G	L	
0	0	0	0	Result is zero
0	0	0	1	Result is less than zero
0	0	1	0	Result is greater than zero
0	1	0	1	Exponent Overflow
0	1	1	0	
0	1	0	0	Exponent Underflow

Programming Notes

In the event of exponent overflow, the proper sign bit is generated and the rest of the result is set to all ones ± 7FFFFFFF .

In the event of exponent underflow, the entire result is set to zero.

Execution Times

MER: 1170/1670/2170 Best/Average/Worst

5.6.15 Divide Single Precision

DER A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
1 1 0	0 1	I	1 1 1 1 1	A	0 1 1 1	B	0	E	D	MC

The exponents of the first and second operands are subtracted and the result set aside as the exponent of the final result. The result sign is determined by the rules of algebra. The second operand divisor is divided into the first operand dividend. If the result is zero, the entire result is set to zero. If the result fraction is non-zero, it is normalized or corrected as necessary. The sign, exponent, and result fraction are combined and replace the contents of the floating point register specified by A.

DER: (A)←(A)/B_E

Resulting Flags

C	V	G	L	
0	0	0	0	Result is zero
0	0	0	1	Result is less than zero
0	0	1	0	Result is greater than zero
0	1	0	1	Exponent Overflow
0	1	1	0	
0	1	0	0	Exponent Underflow
1	1	0	0	Divisor was zero

Programming Notes

In the event of attempted division by zero, the result destination register is unchanged and the operation aborts with a condition code of 1100_2.

See Multiply Single Precision

Execution Times

DER: 2950

5.6.16 Load Unnormalized Double Precision

LW A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	31
1 1 0	0 1	I	1 1 1 1 1	A	1 0 0 0	B	0	E	D	MC

LWX A,B,ADRS,I,C [RR TRANSFER]

0	3	5	6	11	16	20	25	26 31
1 1 0	0 0	I	1 1 1 1 1	A	1 0 0 0	B	C	PAGE ADRS

LWI A,DATA,I [RR IMMEDIATE]

0	3	5	6	11	16	20 31
1 1 0	1 0	I	1 1 1 1 1	A	1 0 0 0	DATA

This microinstruction is required when the second operand for a double precision function is not resident in one of the double precision registers. That is, the operand is contained in a pair of Micro Registers or in main memory or in the Control Store memory.

This instruction specifies the most significant 32-bits of the desired argument. These 32 bits are copied into a holding register in the floating point unit. A subsequent double precision microinstruction places the least significant 32-bits of the operand on the B Bus and the operation is performed.

LW,LWI : $\text{FALU} \leftarrow \text{B}_E$
LWX : $\text{FALU} \leftarrow \text{B}_E$
then $(\text{RLC}_{10:15}) \leftarrow \text{ADRS}$

Resulting flags

Unchanged

Programming Notes

The A field is not used and should be Null Selected.

If the subsequent double precision operation does specify a double precision register, the 32-bits presented by the Load Word microinstruction are ignored and the operation is carried out using the 64-bits of the double precision register specified.

Execution Times

LW,LWX,LWI : 240

5.6.17 Load Register Double Precision

LD A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
1 1 0	0 1	I	1 1 1 1 1	A	1 0 1 0	B	0	E	D	MC

LDX A,B,ADRS,I,C [RR TRANSFER]

0	3	5	6	11	16	20	25	26 31
1 1 0	0 0	I	1 1 1 1 1	A	1 0 1 0	B	C	PAGE ADRS

LDI A,DATA [,I] [RR IMMEDIATE]

0	3	5	6	11	16	20 31
1 1 0	1 0	I	1 1 1 1 1	A	1 0 1 0	DATA

If the second operand is resident in one of the double precision registers, the second operand is normalized, if necessary, and placed in the double precision register specified by A.

If the second operand is not resident in one of the double precision registers, prior to this instruction a Load Word microinstruction has presented the most significant 32-bits of the 64-bit second operand to the floating point unit. This instruction then places the least significant 32-bits of the second operand on the B Bus. The first 32-bits and the 32-bits of B Bus data form the effective second operand. The second operand is normalized, if necessary, and placed in the double precision register specified by A.

LD,LDI : $(A) \leftarrow B_E$
LDX : $(A) \leftarrow B_E$
then $(RLC_{10:15}) \leftarrow ADRS$

Resulting flags

C	V	G	L	
0	0	0	1	Result is less than zero
0	0	1	0	Result is greater than zero
0	1	X	X	Fraction was not normalized

Programming Notes

After a Load microinstruction, the V flag reset means that the argument was already normalized. In this case the G and L flags are properly set. If the V flag is set, the argument fraction was not normalized. An argument of zero is considered to be un-normalized for this test.

The normalization process continues autonomous of other Processor activity. When finished, a Read Condition Code microinstruction will collect the final flags.

Resulting flags (after Read Condition Code)

C	V	G	L	
0	0	0	0	Result is zero
0	0	0	1	Result is less than zero
0	0	1	0	Result is greater than zero
0	1	0	0	Exponent Underflow

The A field may only specify a double precision floating point register.

Execution Times

LD,LDX,LDI : 320+100n

5.6.18 Read Register Double Precision

RRD S,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
1 1 0	0 1	I	S	1 1 1 1 1	1 0 0 1	B	0	E	D	MC

RRDX S,B,ADRS,I,C [RR TRANSFER]

0	3	5	6	11	16	20	25	26 31
1 1 0	0 0	I	S	1 1 1 1 1	1 0 0 1	B	C	PAGE ADRS

The contents of the double precision floating point register half specified by B are copied into the register specified by S. The flags generated by this instruction equal the result flags of the last floating point operation.

RRD: (S)←(B)
RRDX: (S)←(B)
then $(RLC_{10:15})$←Page Address

Resulting Flags

Unchanged from last floating point operation.

Execution Times

RRD, RRDX: 240

When this instruction follows a floating point add, subtract, multiply, or divide, its effective execution times is 60 nanoseconds.

5.6.19 Compare Double Precision

CDR A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
1 1 0	0 1	I	1 1 1 1 1	A	1 0 1 1	B	0	E	D	MC

The first operand is compared to the second operand. The comparison is algebraic, taking into account the sign, exponent, and fraction. The result is indicated by the resulting flags.

CDR: (A): B_E

Resulting Flags

C	V	G	L	
0	0	0	0	First operand equal to second operand
1	0	0	1	First operand less than second operand
0	0	1	0	First operand greater than second operand

Execution Times

CDR: 400

5.6.20 Add Double Precision

ADR A,B,I,E,D,MC [RR CONTROL]

0	3	5 6	11	16	20	25	26	27	28 31
1 1 0	0 1	I 1 1 1 1 1	A	1 1 0 0	B	0	E	D	MC

The first operand is compared to the second operand. The comparison is in magnitude only, ignoring the signs of the two operands. The fraction field of the smaller of the two operands is shifted right hexadecimally (four bits at a time) the number of times indicated by the difference of the exponents of the two operands.

The affect of this process is to unnormalize the smaller operand enough so that the radix points of the two arguments are aligned. If the exponent difference exceeds thirteen, the smaller operand loses significance and a value of zero is substituted.

The equalized fraction and the fraction of the other operand are then algebraically added, taking into account the signs and order of the two operands. The result fraction with an exponent equal to that of the larger operand replaces the contents of the double precision floating point registers specified by A.

If the addition of fractions produces a carry, the result fraction is shifted right one hexadecimal position and the result exponent is incremented by one. If no carry was produced, the result fraction is normalized if necessary. The result exponent is decremented by one for each normalization cycle required.

ADR: $(A) \leftarrow (A)+B_E$

Resulting Flags

C	V	G	L	
0	0	0	0	Result is zero
0	0	0	1	Result is less than zero
0	0	1	0	Result is greater than zero
0	1	0	1	Exponent Overflow
0	1	1	0	
0	1	0	0	Exponent Underflow

Programming Notes

In the event of exponent overflow, the proper sign is generated and the rest of the result is set to all ones + 7FFFFFFFFFFFFFFF .

In the event of exponent underflow, the entire result is set to zero.

Execution Times

ADR: 750 + 100 (e+n) e = equalize cycles
n = normalize cycles

worst case total of e+n = 13

5.6.21 Subtract Double Precision

SDR A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
1 1 0	0 1	I	1 1 1 1 1	A	1 1 0 1	B	0	E	D	MC

The first operand is compared to the second operand. The comparison is in magnitude only, ignoring the signs of the two operands. The fraction field of the smaller of the two operands is shifted right hexadecimally (four bits at a time) the number of times indicated by the difference of the exponents of the two operands.

The affect of this process is to unnormalize the smaller operand enough so that the radix points of the two operands are aligned. If the exponent difference exceeds thirteen, the smaller operand loses significance and a value of zero is substituted.

The equalized fraction and the fraction of the other operand are then algebraically subtracted taking into account the signs and order of the two operands. The result fraction with an exponent equal to that of the larger operand replaces the contents of the double precision floating point register specified by A.

If the subtraction of fractions produces a carry, the result fraction is shifted right one hexadecimal position and the result exponent is incremented by one. If not carry was produced, the result fraction is normalized if necessary. The result exponent is decremented by one for each normalization cycle required.

SDR: $(A) \longleftarrow (A)+B_E$

Resulting Flags

C	V	G	L	
0	0	0	0	Result is zero
0	0	0	1	Result is less than zero
0	0	1	0	Result is greater than zero
0	1	0	1	Exponent overflow
0	1	1	0	
0	1	0	0	Exponent underflow

Programming Notes

See Add Double Precision

Execution Times

SDR: 750 + 100 (e+n) e = equalize cycles
n = normalize cycles

Worst case total of e+n = 13

5.6.22 Multiply Double Precision

MDR A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
1 1 0	0 1	I	1 1 1 1 1	A	1 1 1 0	B	0	E	D	MC

The exponents of the first and second operands are added then set aside as the exponent of the final result. The result sign is determined by the rules of algebra. The fractions of the two operands are then multiplied. If the product is zero, the entire result, sign and exponent included is set to zero. If the product is non-zero, it is normalized or corrected as necessary. The sign, exponent, and result fraction are combined and replace the contents of the double precision floating point register specified by A.

MDR; $(A) \leftarrow (A)*B_E$

Resulting Flags

C	V	G	L	
0	0	0	0	Result is zero
0	0	0	1	Result is less than zero
0	0	1	0	Result is greater than zero
0	1	0	1	Exponent Overflow
0	1	1	0	
0	1	0	0	Exponent Underflow

Programming Notes

In the event of exponent overflow, the proper sign bit is generated and the rest of the result is set to all ones + 7FFFFFFFFFFFFFFF .

In the event of exponent underflow, the entire result is set to zero.

Execution Times

MDR: 1800/3000/4200 Best/Average/Worst

5.6.23 Divide Double Precision

DDR A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
1 1 0	0 1	I	1 1 1 1 1	A	1 1 1 1	B	0	E	D	MC

The exponents of the first and second operands are subtracted and then set aside as the result exponent. The result sign is determined by the rules of algebra. The second operand divisor is divided into the first operand dividend. If the result is zero, the entire result is set to zero. If the result fraction is non-zero, it is normalized or corrected as necessary. The sign, exponent, and result fraction are combined and replace the contents of the double precision floating point register specified by A.

DDR: $(A) \leftarrow (A)/B_E$

Resulting Flags

C	V	G	L	
0	0	0	0	Result is zero
0	0	0	1	Result is less than zero
0	0	1	0	Result is greater than zero
0	1	0	1	Exponent Overflow
0	1	1	0	
0	1	0	0	Exponent Underflow
1	1	0	0	Divisor was zero

Programming Notes

In the event of attempted division by zero, the result destination register is unchanged and the operation aborts with a condition code of 1100_2.

See Multiply Double Precision

Execution Times

DDR: 7420

Implementation Note:

The DFU is normally strapped to respond as Processor Module 6. Consequently, the DFU microinstructions assemble with a module number of 6. If there are two DFUs in the system, the second DFU must be strapped to respond as Processor Module 4. DFU Module 6 will be the primary unit used by the basic microcode. DFU 4 will then be the auxiliary unit available to the WCS user. Since the two DFUs run in a fully autonomous mode, simultaneous calculations can be carried out to greatly improve the overall speed of any heavy floating point algorithms.

In order to allow microcode to be assembled for a DFU strapped as Module Number 4, the Common Microcode Assembler (MICROCAL) has a pair of special psuedo-operations that cause the module number of a DFU directed microinstruction to be switched from Module 6 to Module 4 and vice versa.

The assembler is normally in the DFU Module 6 mode. Consequently, an AER microinstruction will normally assemble with a module number of 6. The appearance in the source program of a DFU4 psuedo operation places the Assembler in the DFU Module 4 mode until a DFU6 psuedo operation is encountered. While in the DFU Module 4 mode, all micro instructions directed to the DFU (AER for example) will assemble with a module number of 4.

A psuedo operation is an instruction only to the assembler and, as such, causes no object code to be generated.

The following code sequence shows the use of two DFUs to calculate B=A^2 + B + C.

where A, B and C are micro registers

```
DFU6                SELECT DFU 6
LE        0,A       LOAD E0 IN DFU 6 WITH A
MER       0,0       FORM A SQUARED
DFU4                SELECT DFU 4
LE        0,B       LOAD E0 IN DFU 4 WITH B
AER       0,C       ADD C
```

*THE MULTIPLY STARTED IN DFU 6 IS STILL IN PROGRESS

```
RRE       B,0       READ B+C RESULT INTO B
DFU6                SWITCH BACK TO DFU 6
RRE       NULL,B    WAIT FOR MULTIPLY TO FINISH
AER       0,B       ADD B TO A SQUARED
RRE       B,0       FINAL RESULT TO B
```

In this example, the execution times of the microinstructions directed to DFU 4 (LU, AER & RRE) can be completely ignored because their total does not exceed the time it took DFU6 to do the initial multiply. The total time, using average execution times is 2.78 microseconds instead of 3.83 microseconds.

5.7 Byte Handling Instructions

These instructions use the I/O module to perform byte manipulations on the least significant 16-bits of A, B, and S Bus data. The instructions described in this section are:

5.7.1	LB LBR	Load Byte Load Byte Register
5.7.2	STB STBR	Store Byte Store Byte Register
5.7.3	EXB	Exchange Byte

5.7.1 Load Byte

LB S,A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
0 1 0	0 1	I	S	A	1 1 0 1	B	1	E	D	MC

LBR S,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
0 1 0	0 1	I	S	1 1 1 1 1	0 1 0 1	B	1	E	D	MC

Bits 16:23 of the first operand or Bits 24:31 of the second operand replace Bits 24:31 of the register specified by S. The most significant 24-bits of S are set to zero. For the LB instruction, whether the A or B operand is used depends upon the state of Memory Address Bit 31. For the LBR instruction, the B operand is always used.

LB: $(S_{0:23}) \leftarrow 0$
$(S_{24:31}) \leftarrow (A_{16:23})$ if $MA_{31} = 0$
$(S_{24:31}) \leftarrow B_E(24:31)$ if $MA_{31} = 1$

LBR: $(S_{0:23}) \leftarrow 0$
$(S_{24:31}) \leftarrow B_E(24:31)$

Resulting flags

C	V	G	L
0	0	0	0

Execution Times

LB,LBR: 250

5.7.2 Store Byte

STB S,A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
0 1 0	0 1	I	S	A	1 1 0 0	B	1	E	D	MC

STBR S,A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
0 1 0	0 1	I	S	A	0 1 0 0	B	1	E	D	MC

Bits 24:31 of A are copied to Bits 24:31 or Bits 16:23 of the register specified by S. The byte position not loaded is replaced by the corresponding byte position of the second operand. Byte steering on the STB instruction depends upon the state of Memory Address Bit 31.

STB: $(S_{0:15}) \leftarrow 0$

$$\left.\begin{array}{l}(S_{16:23}) \leftarrow (A_{24:31})\\ (S_{24:31}) \leftarrow B_E(24:31)\end{array}\right\} MA_{31} = 0$$

$$\left.\begin{array}{l}(S_{16:23}) \leftarrow B_E(16:23)\\ (S_{24:31}) \leftarrow (A_{24:31})\end{array}\right\} MA_{31} = 1$$

STBR: $(S_{0:15}) \leftarrow 0$

$(S_{16:23}) \leftarrow B_E(16:23)$

$(S_{24:31}) \leftarrow (A_{24:31})$

Resulting flags

C	V	G	L
0	0	0	0

Execution Times

STB,STBR : 250

5.7.3 Exchange Byte

EXB S,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
0 1 0	0 1	I	S	1 1 1 1 1	1 1 1 0	B	1	E	D	MC

The two low order bytes of the second operand are exchanged and loaded into the register specified by S.

EXB: $(S_{0:15}) \leftarrow 0$
$(S_{16:23}) \leftarrow B_E(24:31)$
$(S_{24:31}) \leftarrow B_E(16:23)$

Resulting flags

C	V	G	L
0	0	0	0

Execution Times

EXB : 250

5.8 Control Instructions

These instructions allow testing and clearing the Machine Control Register, control over the console interrupt and the console WAIT lamp, control over externally usable signals, and the Initialize relay. The instructions covered in this section are:

5.8.1	SMCR SMCRX	Sense Machine Control Register Sense Machine Control Register and Transfer
5.8.2	CMCR	Clear Machine Control Register
5.8.3	LWFF	Load the Wait Flip-Flop
5.8.4	POUT	Pulse Output Lines
5.8.5	POW	Power Down
5.8.6	BDC	Branch and Disable Console

5.8.1 Sense Machine Control Register

SMCR S,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
010	01	I	S	11111	0111	B	0	E	D	MC

SMCRX S,B,ADRS,I,C [RR TRANSFER]

0	3	5	6	11	16	20	25	26 31
010	00	I	S	11111	0111	B	C	PAGE ADDRESS

The contents of the Machine Control Register replace the contents of the register specified by S. Bits 12:15 of the MCR become available on the CC Bus and are copied into the micro flag register.

SMCR: (S) ← (MCR)
SMCRX: (S) ← (MCR)
then $(RLC_{10:15})$ ← PAGE ADDRESS

Resulting flags

C	V	G	L	
1				Unused
	1			Memory Error
		1		
			1	Early Power Failure

Programming Notes

The B field is not used and should be NULL selected.

The meaning of the Machine Control Register bits is summarized below.

04	DFU	Double Precision Floating Point ALU installed
05	HWCRC	Hardware Assist Cyclic Redundancy Check installed
06	INIT	Initialize Switch on Display Panel Depressed
07	SNGL	Console Single Step. This signal goes active when the SGL switch on the Hexadecimal Display Panel is depressed and remains active until any other functional switch is depressed.
08	Spare	Automatic Restart Option
09	ARST (STRAP)	
10	CATN	Console Attention. This signal goes active when any Display Panel function is selected and remains active until the Hexadecimal Display Panel is addressed.
11	STF	Start Time Failure. An addressed module (e.g., I/O Module) has failed to respond within 35 micro-seconds.
12	Spare	
13	MM	Memory Malfunction (e.g., parity error)
14	MM	
15	EPF	Early Power Failure.

Execution Times

SMCR,SMCRX : 290

5.8.2 Clear Machine Control Register

CMCR S,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
010	01	I	S	1 1 1 1 1	0 1 1 1	B	1	E	D	MC

The bits in the Machine Control Register that correspond to 'ones' in the second operand are set to zeros. The S field is not used and should be NULL selected.

CMCR: (MCR) ← (MCR) AND $\overline{B_E}$

Resulting flags

C	V	G	L
0	0	0	0

Programming Note

The MCR bits that are straps cannot be modified.

Execution Times

CMCR: 240

5.8.3 Load the Wait Flip-Flop

LWFF S,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
0 1 0	0 1	I	S	1 1 1 1 1	0 1 1 0	B	1	E	D	MC

Bit 16 of the second operand is copied into the Wait flip-flop. A 'one' sets the flip-flop and turns on the Console WAIT lamp. A 'zero' resets the flip-flop and turns off the Console WAIT lamp.

LWFF: WAIT ← $B_E(16)$

Resulting flags

C	V	G	L
0	0	0	0

Execution Time

LWFF: 240

5.8.4 Pulse Output Lines

POUT S,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
010	01	I	S	11111	1111	B	1	E	D	MC

The Model 8/32 I/O Module features four board stakes labeled PA0, PB0, PC0 and PD0 to be used for external signalling purposes. The POUT instruction generates a low active pulse on the Px0 lines that correspond to ones in Bits 28:31 of the second operand.

POUT: PA ← B_E(Bit 31)
PB ← B_E(Bit 30)
PC ← B_E(Bit 29)
PD ← B_E(Bit 28)

Resulting flags

C	V	G	L
0	0	0	0

Programming Note

The width of the pulse output (controlled by a timer on the Input/Output Module) can be lengthened if necessary by changing components. The execution times given are for the standard Input/Output Module without system modifications.

Execution Times

POUT: 1220

5.8.5 Power Down

POW S,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
010	01	I	S	11111	1111	B	0	E	D	MC

The Power Down micro-instruction releases the System Clear relay and initializes the system. The S, B and MC fields are not interpreted. The resulting Condition Code and execution time are meaningless.

When the System Clear relay is re-enabled, micro-code execution resumes at Control Store Memory address '001'

5.8.6 Branch and Disable Console

BDC ADRS(LINK),E,D,MC [ADDRESS LINK]

0	3	5	6	11	14	26	27	28 31
000	10	1	LINK	110	ADDRESS	E	D	MC

BDC (B) (LINK),E,D,MC [REGISTER LINK]

0	3	5	6	11	14	20	25	26	27	28 31
000	00	1	LINK	110	▨	B	▨	E	D	MC

Interrupts from the Console (CATN or SNGL) are ignored for the interval of this instruction so that interrupts of lower priority can be detected. No branch is actually taken, so MC and D field functions can occur.

BDC: (LINK) ← $(RLC_{4:15})+1$

Execution Time

BDC: 240 nanoseconds

6. INPUT/OUTPUT SYSTEM

This section discusses the Input/Output (I/O) System. There are several methods of communications between the Processor and peripheral devices or other system elements. The methods vary in speed, sophistication, and the amount of attention required by the Processor.

6.1 Multiplexor Bus

The Multiplexor Bus is a byte or halfword oriented I/O system which communicates with up to 1,023 peripheral devices. The Multiplexor Bus consists of 33 lines – 16 bi-directional data lines, 10 control lines, 6 test lines and 1 initialize line. The lines in the Multiplexor Bus are:

Data Lines	D00:15	(Processor ⇄ Device)	16 lines
Control Lines	SR	(——→)	1 line
	DR	(——→)	1 line
	CMD	(——→)	1 line
	DA	(——→)	1 line
	ADRS	(——→)	1 line
	ACK0	(——→)	1 line
	ACK1	(——→)	1 line
	ACK2	(——→)	1 line
	ACK3	(——→)	1 line
	CL07	(——→)	1 line
Test Lines	ATN0	(←——)	1 line
	ATN1	(←——)	1 line
	ATN2	(←——)	1 line
	ATN3	(←——)	1 line
	SYN	(←——)	1 line
	HW	(←——)	1 line
Initialize	SCLR	(——→)	1 line

6.1.1 Data Lines. The 16 bi-directional data lines are used to transfer one 8-bit byte, one 10-bit device address, or one 16-bit halfword between the Processor and the device. In actuality, 16 bits are always transferred, and the device or the Processor accepts as much of the data as is required for the particular operation.

6.1.2 Control Lines

ADRS Address. The Processor presents a 10-bit device address on Data Lines D06:15. The device controller that recognizes its address becomes the 'on-line' device and responds with a synchronize (SYN). Once a device has been addressed, it remains addressed until a different device is addressed or a system initialize occurs. If the device is halfword oriented, the Halfword test line (HW) is also active.

DA Data Available. The Processor presents data to be transferred to the addressed device on Data Lines D00:15. The addressed device controller accepts the low order byte or the entire halfword and responds with a SYN.

DR Data Request. The addressed device controller presents data on Data Lines D08:15 or D00:15, followed by a SYN.

SR Status Request. The addressed device controller presents status information on Data Lines D08:15, followed by a SYN.

CMD Output Command. The Processor presents a command byte on Data Lines D08:15 for the addressed device. The addressed device controller accepts the command byte and responds with a SYN.

ACK0
ACK1
ACK2
ACK3 Acknowledge. The microprogram generates an Acknowledge signal on the appropriate line in response to an active Attention (ATN) test line. The device controller nearest the Processor on the particular Acknowledge line that is activating the corresponding ATN line presents its address on Data Lines D06:15, followed by a SYN. That device controller then removes ATN.

CL07 This Control Line is activated by the Initialize key, the PWR OFF switch or when the optional Power fail detector determines that the Processor's primary power is failing. The line remains active as long as the PPF interrupt line is active.

6.1.3 Test Lines

ATN0 ATN1 ATN2 ATN3 — Attention. When so enabled, any device on one of the Attention lines desiring to interrupt the Processor will activate the ATN x line and hold it active until an Acknowledge is received from the Processor.

HW — Halfword. Any halfword oriented device will activate the Halfword test line when it becomes addressed and hold the line active for as long as the device is addressed. The HW line being active suppresses the byte steering done in the I/O module on DA or DR operations.

SYN — Synchronize. This signal is generated by the device controller to inform the Processor that it is responding to the active control line.

6.1.4 Initialize Line

SCLR — System Clear. This is a metallic contact to ground that occurs during Power fail or Initialize.

6.2 Input/Output Instructions

Communication over the Multiplexor Bus is on a request/response basis where each operation started by the Processor must receive a SYN response to terminate the operation. If no SYN response is received within approximately 35 micro-seconds, a False Sync (FSYN) is automatically generated to terminate the operation.

Input/Output micro-instructions generate one, two, or three Multiplexor Bus operations. Each operation lasts until a SYN is received from the device, meaning that the execution time on I/O instructions is solely device dependent.

NOTE

> All I/O instruction execution times are given using the following assumptions:
>
> 1. Average circuit delays, not max. or min.
> 2. SYN delay of 100 nanoseconds
> 3. No Bus Buffer delay in the system.

The instructions described in this section are:

6.2.1	AK	Acknowledge Interrupt
	AKX	Acknowledge Interrupt and Transfer
6.2.2	SSA	Address and Sense Status
	SSAX	Address and Sense Status and Transfer
	SSRA	Address and Sense Status Register
6.2.3	SS	Sense Status
	SSX	Sense Status and Transfer
	SSR	Sense Status Register
6.2.4	OCA	Address and Output Command
	OCAX	Address and Output Command and Transfer
	OCAI	Address and Output Command Immediate
	OCRA	Address and Output Command Register
6.2.5	OC	Output Command
	OCX	Output Command and Transfer
	OCI	Output Command Immediate
	OCR	Output Command Register
6.2.6	RDA	Address and Read Data
	RDAX	Address and Read Data and Transfer
	RDRA	Address and Read Data Register
6.2.7	RD	Read Data
	RDX	Read Data and Transfer
	RDR	Read Data Register
6.2.8	WDA	Address and Write Data
	WDAX	Address and Write Data and Transfer
	WDAI	Address and Write Data Immediate
	WDRA	Address and Write Data Register

6.2.9	WD WDX WDI WDR	Write Data Write Data and Transfer Write Data Immediate Write Data Register
6.2.10	RHA RHAX	Address and Read Halfword Address and Read Halfword and Transfer
6.2.11	RH RHX	Read Halfword Read Halfword and Transfer
6.2.12	WHA WHAX	Address and Write Halfword Address and Write Halfword and Transfer
6.2.13	WH WHX	Write Halfword Write Halfword and Transfer
6.2.14	THWX	Test Halfword Line and Transfer

6.2.1 Acknowledge Interrupt

AK S,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
0 1 0	0 1	I	S	1 1 1 1 1	0 1 1 0	B	0	E	D	MC

AKX S,B,ADRS,I,C [RR TRANSFER]

0	3	5	6	11	16	20	25	26 31
0 1 0	0 0	I	S	1 1 1 1 1	0 1 1 0	B	C	PAGE ADDRESS

Bits 30 and 31 of the effective second operand select the desired ACK Control line. The device number of the interrupting device replaces the contents of the register specified by S. The interrupt condition in the controller is cleared.

AK : $(S_{0:21}) \leftarrow 0$
$(S_{22:31}) \leftarrow$ DEVICE NUMBER

AKX : $(S_{0:21}) \leftarrow 0$
$(S_{22:31}) \leftarrow$ DEVICE NUMBER
then $(RLC_{10:15}) \leftarrow$ PAGE ADDRESS

Resulting flags

C	V	G	L	
0	0	0	0	Normal Execution
0	1	0	0	Instruction Time-Out (No Interrupts)

Programming Note

Each ACK Control Line passes through the interrupt circuits on all of its assigned controllers in a 'daisy chain' fashion. The execution time is increased by 100 nanoseconds for each controller between the Processor and the interrupting controller.

Execution Time

AK,AKX : 480

6.2.2 Address and Sense Status

SSA S,A,B,I,E,D,MC [RR CONTROL]

0	2	4	5	11	16	20	25	26	27	28 31
010	01	I	S	A	1010	B	0	E	D	MC

SSAX S,A,B,ADRS,I,C [RR TRANSFER]

0	2	4	5	11	16	20	25	26 31
010	00	I	S	A	1010	B	C	PAGE ADDRESS

SSRA S,A,B,I,E,D,MC [RR CONTROL]

0	2	4	5	11	16	20	25	26	27	28 31
010	01	I	S	A	1010	B	1	E	D	MC

The register specified by A contains the device address. The device is addressed and its 8-bit Status byte replaces the contents of the register specified by S. The right most four bits of the Status byte are available on the CC Bus and are copied into the micro flag register.

SSA: CC Bus $\leftarrow$ DEVICE STATUS (4:7)

$(S_{0:15}) \leftarrow 0$

$$\left.\begin{array}{l}(S_{16:23}) \leftarrow \text{DEVICE STATUS}\\ (S_{24:31}) \leftarrow B_E(24:31)\end{array}\right\} \text{if MA31} = 0$$

$$\left.\begin{array}{l}(S_{16:23}) \leftarrow B_E(16:23)\\ (S_{24:31}) \leftarrow \text{DEVICE STATUS}\end{array}\right\} \text{if MA31} = 1$$

SSAX: Same as SSA
then $(RLC_{10:15}) \leftarrow$ PAGE ADDRESS

SSRA CC Bus $\leftarrow$ DEVICE STATUS (4:7)

$(S_{0:15}) \leftarrow 0$

$(S_{16:23}) \leftarrow B_E(16:23)$

$(S_{24:31}) \leftarrow$ DEVICE STATUS

Resulting flags

C	V	G	L	
1				Device Busy (BSY)
	1			Examine Status (EX) or Time-Out
		1		End of Medium (EOM)
			1	Device Unavailable (DU)

Execution Times

SSA,SSAX,SSRA : 1180

6.2.3 Sense Status

SS S,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
010	01	I	S	11111	0010	B	0	E	D	MC

SSX S,B,ADRS,I,C [RR TRANSFER]

0	3	5	6	11	16	20	25	26 31
010	00	I	S	11111	0010	B	C	PAGE ADDRESS

SSR S,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
010	01	I	S	11111	0010	B	1	E	D	MC

The Sense Status instructions are identical to the Address and Sense Status instructions except that the address cycle is avoided. Once addressed, a device controller remains addressed until a different device controller is addressed or a System Clear occurs.

Execution Times

SS,SSX,SSR : 480

6.2.4 Address and Output Command

OCA S,A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
0 1 0	0 1	I	S	A	1 0 1 1	B	0	E	D	MC

OCAX S,A,B,ADRS,I,C [RR TRANSFER]

0	3	5	6	11	16	20	25	26 31
0 1 0	0 0	I	S	A	1 0 1 1	B	C	PAGE ADDRESS

OCAI S,A,DATA,I [RR IMMEDIATE]

0	3	5	6	11	16	20 31
0 1 0	1 0	I	S	A	1 0 1 1	DATA

OCRA S,A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
0 1 0	0 1	I	S	A	1 0 1 1	B	1	E	D	MC

The register specified by A contains the device address. The device is addressed and the 8-bit second operand command byte is sent to the device.

OCA,OCAI : DEVICE← $B_E(16:23)$ if MA31=0
DEVICE← $B_E(24:31)$ if MA31=1

OCAX : Same as OCA
then $(RLC_{10:15})$←PAGE ADDRESS

OCRA : DEVICE← $B_E(24:31)$

Resulting flags

C	V	G	L	
0	0	0	0	Normal Execution
0	1	0	0	Instruction Time-Out

Programming Note

The S field is not used and should be NULL selected.

Execution Times

OCA,OCAX,OCAI,OCRA : 1280

6.2.5 Output Command

OC S,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
010	01	I		11111	0011	B	0	E	D	MC

OCX S,B,ADRS,I,C [RR TRANSFER]

0	3	5	6	11	16	20	25	26 31
010	00	I	S	11111	0011	B	C	PAGE ADDRESS

OCI S,B,DATA,I [RR IMMEDIATE]

0	3	5	6	11	16	20 31
010	10	I	S	11111	0011	DATA

OCR S,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
010	01	I	S	11111	0011	B	1	E	D	MC

The Output Command instructions are identical to the Address and Output Command instructions except that the address cycle is avoided.

Execution Times

OC,OCX,OCI,OCR : 580

6.2.6 Address and Read Data

RDA S,A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
0 1 0	0 1	I	S	A	1 0 0 0	B	0	E	D	MC

RDAX S,A,B,ADRS,I,C [RR TRANSFER]

0	3	5	6	11	16	20	25	26 31
0 1 0	0 0	I	S	A	1 0 0 0	B	C	PAGE ADDRESS

RDRA S,A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
0 1 0	0 1	I	S	A	1 0 0 0	B	1	E	D	MC

The register specified by A contains the device address. The device is addressed and a single 8-bit data byte is transferred from the device to the register specified by S.

RDA: $(S_{0:15}) \leftarrow 0$

$(S_{16:23}) \leftarrow$ DEVICE DATA
$(S_{24:31}) \leftarrow B_E(24:31)$ } If MA31 = 0

$(S_{16:23}) \leftarrow B_E(16:23)$
$(S_{24:31}) \leftarrow$ DEVICE DATA } If MA 31 = 1

RDAX: Same as RDA
then $(RLC_{10:15}) \leftarrow$ PAGE ADDRESS

RDRA: $(S_{0:15}) \leftarrow 0$
$(S_{16:23}) \leftarrow B_E(16:23)$
$(S_{24:31}) \leftarrow$ DEVICE DATA

Resulting flags

C	V	G	L	
0	0	0	0	Normal Execution
0	1	0	0	Instruction Time-out

Execution Times

RDA,RDAX,RDRA : 1180

6.2.7 Read Data

RD S,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
010	01	I	S	11111	0000	B	0	E	D	MC

RDX S,B,ADRS,I,C [RR TRANSFER]

0	3	5	6	11	16	20	25	26 31
010	00	I	S	11111	0000	B	C	PAGE ADDRESS

RDR S,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
010	01	I	S	11111	0000	B	1	E	D	MC

The Read Data instructions are identical to the Address and Read Data instructions except that the address cycle is avoided.

Execution Times

RD,RDX,RDR : 480

6.2.8 Address and Write Data

WDA S,A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
010	01	I	S	A	1001	B	0	E	D	MC

WDAX S,A,B,ADRS,I,C [RR TRANSFER]

0	3	5	6	11	16	20	25	26 31
010	00	I	S	A	1001	B	C	PAGE ADDRESS

WDAI S,A,DATA,I [RR IMMEDIATE]

0	3	5	6	11	16	20 31
010	10	I	S	A	1001	DATA

WDRA S,A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
010	01	I	S	A	1001	B	1	E	D	MC

The register specified by A contains the device address. The device is addressed and a single 8-bit byte is transferred to the device.

WDA,WDAI : DEVICE $\leftarrow B_E(16:23)$ if MA 31=0
DEVICE $\leftarrow B_E(24:31)$ if MA 31=1

WDAX : Same as WDA
then $(RLC_{10:15}) \leftarrow$ PAGE ADDRESS

WDRA : DEVICE $\leftarrow B_E(24:31)$

Resulting flags:

C	V	G	L	
0	0	0	0	Normal Execution
0	1	0	0	Instruction Time-Out

Programming Note

The S field is not used and should be NULL selected.

Execution Times

WDA,WDAX,WDAI,WDRA : 1280

6.2.9 Write Data

WD S,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
0 1 0	0 1	I	S	1 1 1 1 1	0 0 0 1	B	0	E	D	MC

WDX S,B,ADRS,I,C [RR TRANSFER]

0	3	5	6	11	16	20	25	26 31
0 1 0	0 0	I	S	1 1 1 1 1	0 0 0 1	B	C	PAGE ADDRESS

WDI S,DATA,I [RR IMMEDIATE]

0	3	5	6	11	16	20 31
0 1 0	1 0	I	S	1 1 1 1 1	0 0 0 1	DATA

WDR S,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
0 1 0	0 1	I	S	1 1 1 1 1	0 0 0 1	B	1	E	D	MC

The Write Data instructions are identical to the Address and Write Data instructions except that the address cycle is avoided.

Execution Times

WD,WDX,WDI,WDR : 580

6.2.10 Address and Read Halfword

RHA S,A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
010	01	I	S	A	1100	B	0	E	D	MC

RHAX S,A,B,ADRS,I,C [RR TRANSFER]

0	3	5	6	11	16	20	25	26 31
010	00	I	S	A	1100	B	C	PAGE ADDRESS

The register specified by A contains the device address. The device is addressed and a 16-bit halfword is transferred from the device to the register specified by S. The Read Halfword instructions can be used with both byte and halfword oriented controllers. If the controller is byte oriented, the Halfword test line (HW) is inactive. The I/O module inputs two 8-bit bytes, one after the other. If the controller is halfword oriented, the Halfword test Line (HW) is active. The I/O module inputs one 16-bit Halfword in parallel.

RHA : $(S_{0:15}) \leftarrow 0$
$(S_{16:23}) \leftarrow$ First Data Byte
$(S_{24:31}) \leftarrow$ Second Data Byte } Byte oriented Controller
$(S_{16:31}) \leftarrow$ Halfword of Data Halfword oriented Controller

RHAX : Same as RHA
then $(RLC_{10:15}) \leftarrow$ PAGE ADDRESS

Resulting flags

C	V	G	L	
0	0	0	0	Normal Execution
0	1	0	0	Instruction Time-Out

Programming Note

The B field is not used and should be NULL selected.

Execution Times

RHA,RHAX :	1400	Byte oriented device
	1180	Halfword oriented device

6.2.11 Read Halfword

RH S,B,I,E,D,MC [RR CONTROL]

0	3	=	6	11	16	20	25	26	27	28 31
010	01	I	S	11111	0100	B	0	E	D	MC

RHX S,B,ADRS,I,C [RR TRANSFER]

0	3	5	6	11	16	20	25	26 31
010	00	I	S	11111	0100	B	C	PAGE ADDRESS

The Read Halfword instructions are identical to the Address and Read Halfword instructions except that the address cycle is avoided.

Execution Times

RH,RHX	700	Byte oriented device
	480	Halfword oriented device

6.2.12 Address and Write Halfword

WHA S,A,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	24	25	26	27	28 31
0 1 0	0 1	I	S	A	1 1 0 1	B	0	E	D	MC

WHAX S,A,B,ADRS,I,C [RR TRANSFER]

0	3	5	6	11	16	20	25	26 31
0 1 0	0 0	I	S	A	1 1 0 1	B	C	PAGE ADDRESS

The register specified by A contains the device address. The device is addressed and a 16-bit Halfword is transferred from the Processor to the device. The Write Halfword instructions can be used with either byte or halfword oriented controllers. If the controller is byte oriented, the Halfword test line (HW) is inactive. The I/O module outputs two 8-bit bytes, one after the other. If the controller is halfword oriented, the Halfword test line (HW) is active. The I/O Module outputs one 16-bit halfword in parallel.

$$\text{WHA} : \left.\begin{array}{l}\text{DEVICE} \leftarrow B_E(16{:}23)\\ \text{DEVICE} \leftarrow B_E(24{:}31)\end{array}\right\}\text{Byte oriented controller}$$

$$\text{DEVICE} \leftarrow B_E(16{:}31) \quad \text{Halfword oriented controller}$$

WHAX : Same as WHA
then $(RLC_{10:15}) \leftarrow$ PAGE ADDRESS

Resulting flags

C	V	G	L	
0	0	0	0	Normal Execution
0	1	0	0	Instruction Time-Out

Programming Note

The S field is not used and should be NULL selected.

Execution Times

WHA,WHAX :	1740	Byte oriented device
	1280	Halfword oriented device

6.2.13 Write Halfword

WH S,B,I,E,D,MC [RR CONTROL]

0	3	5	6	11	16	20	25	26	27	28 31
0 1 0	0 1	I	S	1 1 1 1 1	0 1 0 1	B	0	E	D	MC

WHX S,B,ADRS,I,C [RR TRANSFER]

0	3	5	6	11	16	20	25	26 31
0 1 0	0 0	I	S	1 1 1 1 1	0 1 0 1	B	C	PAGE ADDRESS

The Write Halfword instructions are identical to the Address and Write Halfword instructions except that the address cycle is avoided.

Execution Times

WH,WHX :	1040	Byte oriented device
	580	Halfword oriented device

6.2.14 Test Halfword Line and Transfer

THWX S,B,ADRS,I,C [RR TRANSFER]

0	3	5	6	11	16	20	25	26 31
0 1 0	0 0	I	S	1 1 1 1 1	1 1 1 0	B	C	PAGE ADDRESS

This micro-instruction is provided so that the micro-program can test the state of the Halfword test Line (HW). The Halfword test line is active for as long as any halfword oriented controller is addressed.

THWX: $(RLC_{10:15}) \leftarrow$ PAGE ADDRESS if C=0 or HW=0
$(RLC_{4:15}) \leftarrow (RLC_{4:15})+1$ if C=1 and HW=1

Resulting flags

C	V	G	L
0	0	0	0

Programming Note

The S and B fields are unnused and should be NULL selected.

Execution Times

THWX	(No transfer)	:	360
	(Transfer)	:	240

7. INTERRUPT SYSTEM

The hardware priority interrupt structure provides rapid response to internal and external events that require special program attention. In the interrupt procedure, the micro-program is steered to a unique Control Store address for each of the nine possible interrupts.

Certain interrupts can be individually enabled or disabled by bits in the PSW. All interrupts except the Memory Access Controller can be collectively enabled or disabled by the Branch/Execute and Link and Arm, Branch/Execute and Link and Disarm micro-instructions. Interrupts are automatically armed when a micro-instruction specifies the decode option so that interrupt service can occur before starting the next user instruction. The interrupts are then dis-armed into the subsequent emulation sequence until specifically armed by the micro-program.

When an interrupt occurs, the micro-instruction at its respective trap location is executed. The RLC is not changed so that the micro-program could return to the interrupted program sequence if desired. The standard Model 8/32 micro-program does not use this capability.

The interrupts with pertinent enabling PSW bits and trap locations are shown in Table 2. The descriptions that follow are oriented towards the emulator.

7.1 Internal Interrupts

Seven different internal interrupts may be generated. Of these, the Fixed-Point Divide Fault, Floating-Point Arithmetic Fault, Queue Service, and Supervisor Call are created by the emulator and consequently do not have dedicated trap addresses. The remaining three – Illegal Instruction, Memory Access Controller, and Machine Malfunction – are generated in the hardware.

7.1.1 Illegal Instruction Interrupt. The Illegal Instruction Interrupt is generated when an instruction not in the user level instruction repertoire is attempted or when execution of a "Privileged Instruction" is attempted and PSW Bit 23 is set.

As a result of an Instruction Read, the main memory gates its read-out into the user's Instruction Register (IR). When the Decode (D) option is also specified, at the conclusion of the present micro-instruction, the Processor waits until the next user-instruction is available in IR, at which time the Privileged/Illegal ROM is interrogated.

The Privileged/Illegal ROM is addressed by the Operation Code field of IR (IR Bits 0:7). There is a four-bit data entry in the Privileged/Illegal ROM for each of the 256 possible user op-codes.

The occurrance of the Illegal Instruction Interrupt causes the micro-instruction at Control Store location '208' to be executed.

7.1.2 Memory Access Controller Interrupt. The Memory Access Controller Interrupt, enabled by Bit 21 of the PSW, occurs when the currently running program violates any of the relocation and protection conditions in the Memory Access Controller. In response to the Memory Access Controller Interrupt, the micro-instruction at Control Store location '207' is executed for a data violation.

In addition, following an Instruction Read and Decode specification, if the instruction fetched is from a location that is identified as invalid, non-present or execute protected by the Memory Access Controller, the op-code field of IR is jammed to X'FF'. In this circumstance, op-code 'FF' is not interpreted as illegal, but rather as cause for a Memory Access Controller Interrupt. When this occurs, the instruction at '1FE' is executed in response to the Decode (D) option.

7.1.3 Machine Malfunction Interrupt. The Machine Malfunction Interrupt, enabled by Bit 18 of the PSW, occurs on Memory Parity Errors or Early Power Fail detect. The emulator also generates a Machine Malfunction Interrupt on Power Up if PSW Bit 18 is set.

Specifically, the Machine Malfunction Interrupt occurs if any of the right-most three bits of the Machine Control Register (MCR) are set. See Table 4.

In response to the Machine Malfunction Interrupt, the micro-instruction at Control Store location, '205' is executed.

Early Power Fail

The Early Power Fail bit sets if the Power Fail Detector determines that the primary line voltage is low, or when the Initialize key is depressed or the key-operated Power switch is turned to the off position. When any of the above events occurs, a one millisecond timer is started and the Early Power Fail bit in MCR is set. The user program may do any necessary system shutdown procedures during this one millisecond interval. PSW Bit 18 may again be set to look for parity errors or to prepare for the interrupt on Power Up. The Early Power Fail interrupt will not re-occur.

At the end of the EPF one millisecond time-out, the Primary Power Fail (PPF) interrupt is generated. In response to the Primary Power Fail interrupt, the micro-instruction at Control Store location '206' is executed. This instruction is a branch to a micro-routine that saves the PSW and LOC and all the user's registers. After this, a Power Down (POW) micro-instruction is executed which releases the System Clear relay, holding the system in an initialized state until power is restored.

If all interrupts had been collectively disabled by the micro-program when the PPF signal occurred, after another one millisecond interval, the hardware automatically releases the System Clear relay, initializing the system.

Memory Parity Error

If the Memory Parity option is present, the parity bit of each halfword in main memory is set or reset to maintain odd parity. The parity bit is generated on every Data Write and checked on every Instruction Read or Data Read. If a parity error occurs on an Instruction Read or on a Data Read, one of the Memory Malfunction bits in MCR is set.

If enabled by PSW Bit 18 and armed by the micro-program when the current micro-instruction is finished, the Machine Malfunction Interrupt is taken.

7.2 External Interrupts

If individually enabled by the user, a peripheral device controller is allowed to request Processor service when the device itself is ready to transfer data. The Processor has five priority interrupt lines related to peripheral device handling. These are Display Console requests, and Interrupt requests occurring on Interrupt Lines 0:3. Whenever an External Interrupt occurs, it remains pending until the Processor recognizes and services the interrupt.

7.2.1 Display Console. The Display Console generates an interrupt when an Address, Memory Read, Memory Write, Examine Register, or Examine Floating-Point Register operation is initiated from the console or when a function is selected, or Single Step or Run Mode is selected. The occurrance of any of the above causes the signal CATN to go active. Selecting Single Step also causes the signal SNGL to go active.

The signal CATN remains active until the Display Console is addressed by the micro-program. The signal SNGL remains active until a different operation is selected. The CATN and SNGL signals are copied in the Machine Control register so that the micro-program can distinguish between the two signals.

The Display Console interrupt is only tested during the Decode option of a micro-instruction. The implication is that a console interrupt can only be serviced at the end of a user's instruction. When the console interrupt is taken, the micro-instruction at Control Store location '204' is executed.

The Branch and Disable Console Interrupt micro-instruction momentarily disables the Console Attention or Single Step signal so that lower priority I/O interrupts can be examined. The interrupt is only disabled for this one micro-instruction.

7.2.2 Attention. The four I/O Attention lines are processed in the priority shown below:

Priority	Attention Line
First	0
Second	1
Third	2
Fourth	3

PSW Bits 17, 20, 26, and 27 affect the enable status of the four I/O Attention lines as shown in Table 3. The emulator handles I/O Interrupts in one of two manners, depending upon data in main memory.

8. INSTRUCTION EXECUTION

User instructions are maintained in the main memory. The user instruction to be executed next is at the Main Memory address specified by the Location Counter (LOC). The micro-program begins to emulate that user instruction by doing an Instruction Read. On the same micro-instruction or on a subsequent micro-instruction, the Decode option is specified. Because the micro-program need not specify Instruction Read and Decode in the same micro-instruction, the instruction fetch is discussed in two phases.

8.1 Instruction Read

In response to an Instruction Read, the halfword whose address is in the Location Counter is fetched and placed in the user's Instruction Register (IR). Loading IR has no immediate affect on the YD and YS registers. These registers are not modified until the Decode option is specified so that the micro-program can continue using YD and YS for selecting the user's registers. See Figure 1.

At the same time that the user's Operation Code is loaded into IR, a decision is made, based on Bit 1 of the first halfword, whether or not additional halfwords must be fetched from memory to make up the complete instruction word. As soon as the IR is filled, the Format ROM is interrogated to determine the instruction format. The format ROM is a separate Read-Only-Memory containing 256 4-bit words, one word for each possible user level operation code. The nature of the data in the format ROM is shown below.

1	0	0	0	RX format
0	0	0	1	RI1 format
0	1	0	0	RI2 format
0	0	1	0	RR or short format
0	0	0	0	Undefined

The user level instruction formats are shown in Figure 3.

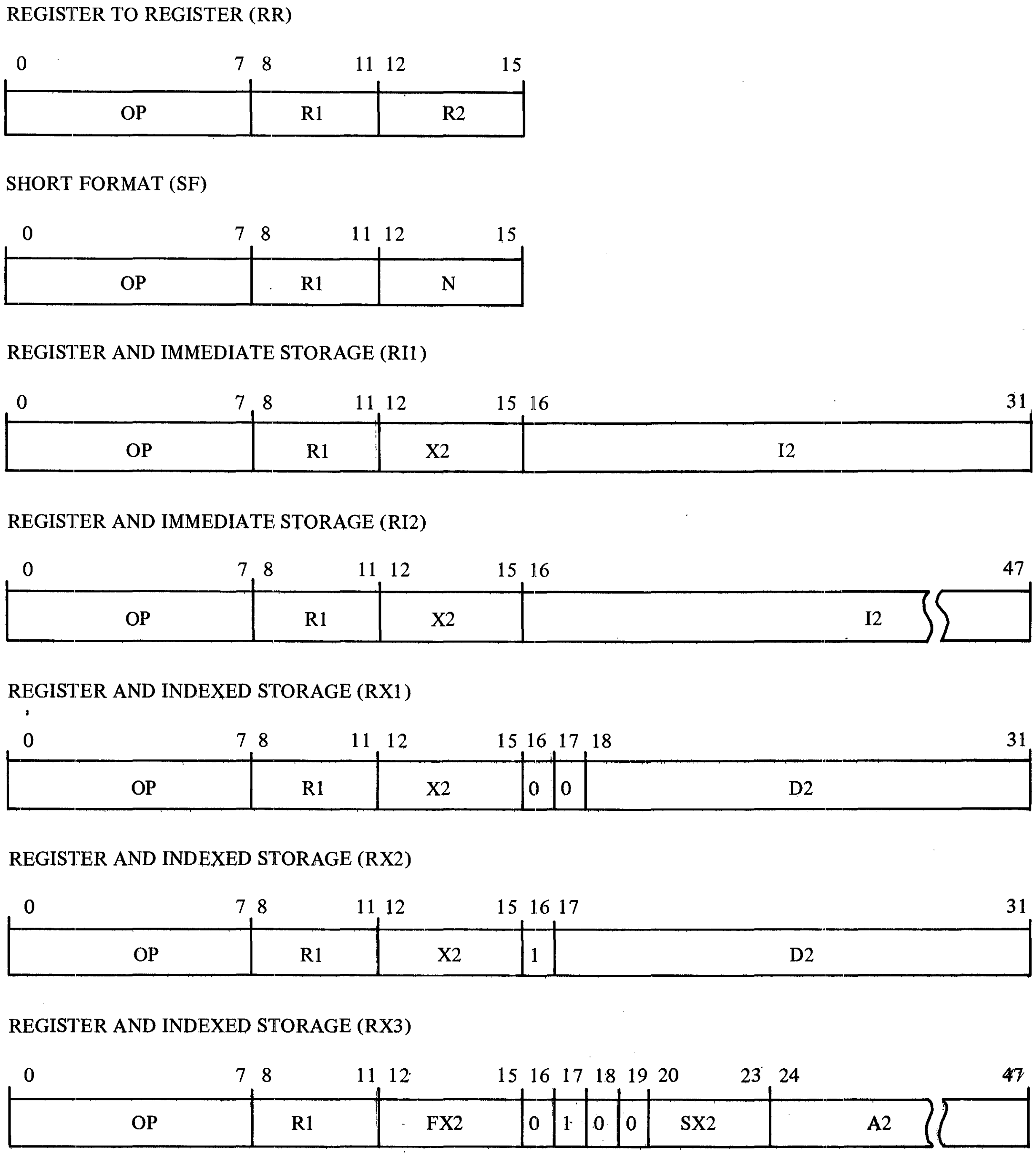


Figure 3. User Level Instruction Formats

The Processor knows from the output of the format ROM and from Bits 16 and 17 of the second halfword, if a third halfword, for RX3 and RI2 formats, is required.

The hardware automatically fetches the appropriate number of halfwords so that after the instruction read is performed, the user's Instruction Register contains the most significant 16-bits of the instruction and the Memory Data Register (MDR) contains the information shown on Table 8. However, when the micro-program attempts to unload MDR to the B Bus, the data shown on Table 9 is received instead of the actual MDR.

TABLE 8. STATE OF MDR AFTER INSTRUCTION READ

INSTRUCTION FORMAT	CONTENTS OF MDR
RR or SF	0 31 UNDEFINED
RI1	0 15 16 31 I2 FIELD OF INSTRUCTION I2 FIELD OF INSTRUCTION
RI2	0 31 I2 FIELD OF INSTRUCTION
RX1	0 1 2 16 17 18 31 0 0 D2 FIELD OF INSTRUCTION 0 0 D2 FIELD OF INSTRUCTION
RX2	0 1 16 17 31 1 D2 FIELD OF INSTRUCTION 1 D2 FIELD OF INSTRUCTION
RX3	0 3 4 7 8 31 0100 SX2 A2 FIELD OF INSTRUCTION

TABLE 9. B BUS GATING AFTER INSTRUCTION READ

INSTRUCTION FORMAT	STATE OF B BUS WHEN UNLOAD MDR
RR or SF	0 31 UNDEFINED
RI1	0 15 16 31 EXTENDED SIGN I2 FIELD OF INSTRUCTION
RI2	0 31 I2 FIELD OF INSTRUCTION
RX1	0 16 17 18 31 ZERO 0 D2 FIELD OF INSTRUCTION
RX2	0 16 17 31 EQUALS BIT 17 D2 FIELD OF INSTRUCTION
RX3	0 31 CONTENTS OF REGISTER SELECTED BY SX2

For the Register and Immediate Storage–RI1–format, Bits 0:15 of MDR are set equal to the sign bit of the halfword in Bits 16:31. For the RX1 format, Bits 0:16 of MDR are zero. For the RX2 format, Bits 0:16 of MDR are set equal to Bit 17. For the RX3 format, until a micro-instruction is performed that loads the Memory Address Register (MAR), any reference to MDR as a source will cause the contents of the General Register whose address is in the SX2 field of the instruction to appear on the B Bus instead of MDR. See 8.3.

8.2 Decode Option

In response to the Decode option, the Processor first tests for any pending interrupts. If an interrupt is pending, the instruction fetch is aborted and the interrupt is serviced. If no interrupt is pending, the YD and YS registers are updated. Twice the user's operation code is presented to the ROM Address Gate as the starting address of the appropriate emulation sequence and the Privileged/Illegal ROM is interrogated. This is a separate Read-Only-Memory containing 256 4-bit words, one for each possible user level operation code. The nature of the data in the Privileged/Illegal ROM is shown below.

1				Illegal Instruction, Model 8/32 with double precision floating-point
	1			Illegal Instruction, Model 8/32 with floating-point
		1		Illegal Instruction, basic Model 8/32
			1	Privileged Instruction

If the output of the Privileged/Illegal ROM indicates that the operation code presently in IR is illegal, the instruction fetch is aborted and the Illegal Instruction Interrupt is taken. If the output of the Privileged/Illegal ROM indicates that the operation code presently in IR is that of a 'Privileged' instruction and PSW Bit 23 is set, the Illegal Instruction Interrupt is taken.

If no interrupt occurs, the ROM Location Counter is set equal to twice the user's operation code and the emulation sequence begins.

8.3 Operand Fetch

If the user level instruction is Register to Register or Short Format, the second operand is already available in a General Register or in the instruction word itself. If the user level instruction is one of the Register and Immediate Storage formats, the immediate operand is available in MDR. All that remains to be done is to add in the contents of the specified index register. The first micro-instruction of a Register and Immediate Storage format instruction could be:

```
      A       MDR,YX,MDR
```

After the instruction, MDR contains the sum of the I2 field of the instruction and the contents of the General Register specified by the X2 field of the instruction.

If the user level instruction is one of the Register and Indexed Storage formats, and the micro-program does not need to know what the effective second operand address is, the first micro-instruction of the emulation sequence should be:

```
      A       MAR,YX,MDR,DR2

or    A       MAR,YX,MDR,DR4
```

depending upon whether the second operand is to be a halfword (2 bytes) or a fullword (4 bytes).

If the instruction format is RX1, the sum of MDR and the contents of the General Register specified by the X2 field of the instruction replaces the contents of MAR. Referring to Figure 1, the output of MAR is passed, unaltered, through the 20-bit adder to the Memory Access Controller (MAC). The MAC presents the address or a translated address to the memory bus and the memory read is started. As soon as the data becomes available in MDR, the instruction fetch is over.

If the instruction format is RX2, the sum of MDR and the contents of the General Register specified by the X2 field of the instruction replaces the contents of MAR. The output of MAR is added to the contents of the Location Counter. This sum, plus four is presented via the MAC to the memory bus and the memory read is started.

If the instruction format is RX3, the sum of the contents of the General Register specified by the SX2 field of the instruction and the contents of the General Register specified by the FX2 field of the instruction replaces the contents of MAR. Remember that if the format is RX3, until MAR is loaded, any reference to MDR as a source causes the second level index register (SX2) to be accessed instead. The output of MAR is added to the contents of MDR and this sum is presented via the MAC to the Memory Bus. The memory read is then begun.

For those user level instructions where the micro-program does need to know the effective second operand address, or those user instructions that will change MDR before a memory operation is begun (such as Store), additional steps must be taken.

The first micro-instruction of the RX type emulation sequence can still be:

```
A        MAR,YX,MDR
```

Now the contents of MAR is:

FORMAT	
RX1	(YX) + $(MDR_{12:31})$
RX2	(YX) + $(MDR_{12:31})$
RX3	(YX) + (SX2)

and the output of the 20-bit ALU is:

FORMAT	
RX1	$(MAR_{12:31}) + 0$
RX2	$(MAR_{12:31}) + (LOC_{12:31})$
RX3	$(MAR_{12:31}) + (MDR_{12:31})$

Except for the RX2 case, it would be sufficient at this point to just unload MAR and save the address temporarily in a micro-register. However, when the instruction format is RX2, the output of the 20-bit ALU is incorrect. The output should be:

$$(YX) + (MDR_{12:31}) + (LOC_{12:31}) +4$$
or $$(MAR_{12:31}) + (LOC_{12:31}) +4$$

The address is being incremented by four as it passes through the MAC, but when the micro-program needs to know the exact address, it must add four to the output of the 20-bit ALU if the instruction format is RX2.

The micro-program can know the format of the user instruction by making the first micro-instruction a RR transfer format. Then the proper micro-sequence could be:

```
RX        AX     MAR,YX,MDR,RX1OR3,C     TRANSFER IF RX1 OR RX3
*                                        FALL THROUGH IF RX2
          LI     MR0,4                   MR0 = '00000004'
          AX     MR0,MR0,MAR,RXX         MR0 = YX+MDR+LOC+4,
*                                        SKIP TO RXX
*
RX1OR3    L      MR0,MAR                 MR0=EFFECTIVE ADDRESS
RXX
```

Note that only in the situation where the Add and Transfer micro-instruction is the first micro-instruction of an RX emulation sequence, the condition for transfer is whether or not the user instruction format is RX2 rather than the state of the ALU carry.

An alternate approach when the emulation sequence cannot be interrupted is to increment the Location Counter and thus let the 20-bit ALU do all the work. The proper micro-sequence for doing this is:

```
RX        A      MAR,YX,MDR,IL           MAR = YX+MDR or YX+SX2
*                                        LOC IS INCREMENTED BY 4 OR 6
          L      MR0,MAR                 MR0 = EFFECTIVE ADDRESS
```

9. THE MICRO-PROGRAM

The following sections describe major sections of the Model 8/32 micro-program. The micro-program listing is well documented and generally self explanatory for the simpler micro-code sequences.

9.1 System Initialization

On power up or following initialize, micro-code execution begins at Control Store address '000'. Referring to Figure 4, Micro Register 0 is set to 05 , the Loader Storage Unit device number. The Machine Control Register is copied into (MRD) and a 2 millisecond microcode delay loop is performed. The Loader Storage Unit (LSU) is addressed. If a false sync occurs, no LSU is present and routine POWRUP is entered for a normal power-up sequence.

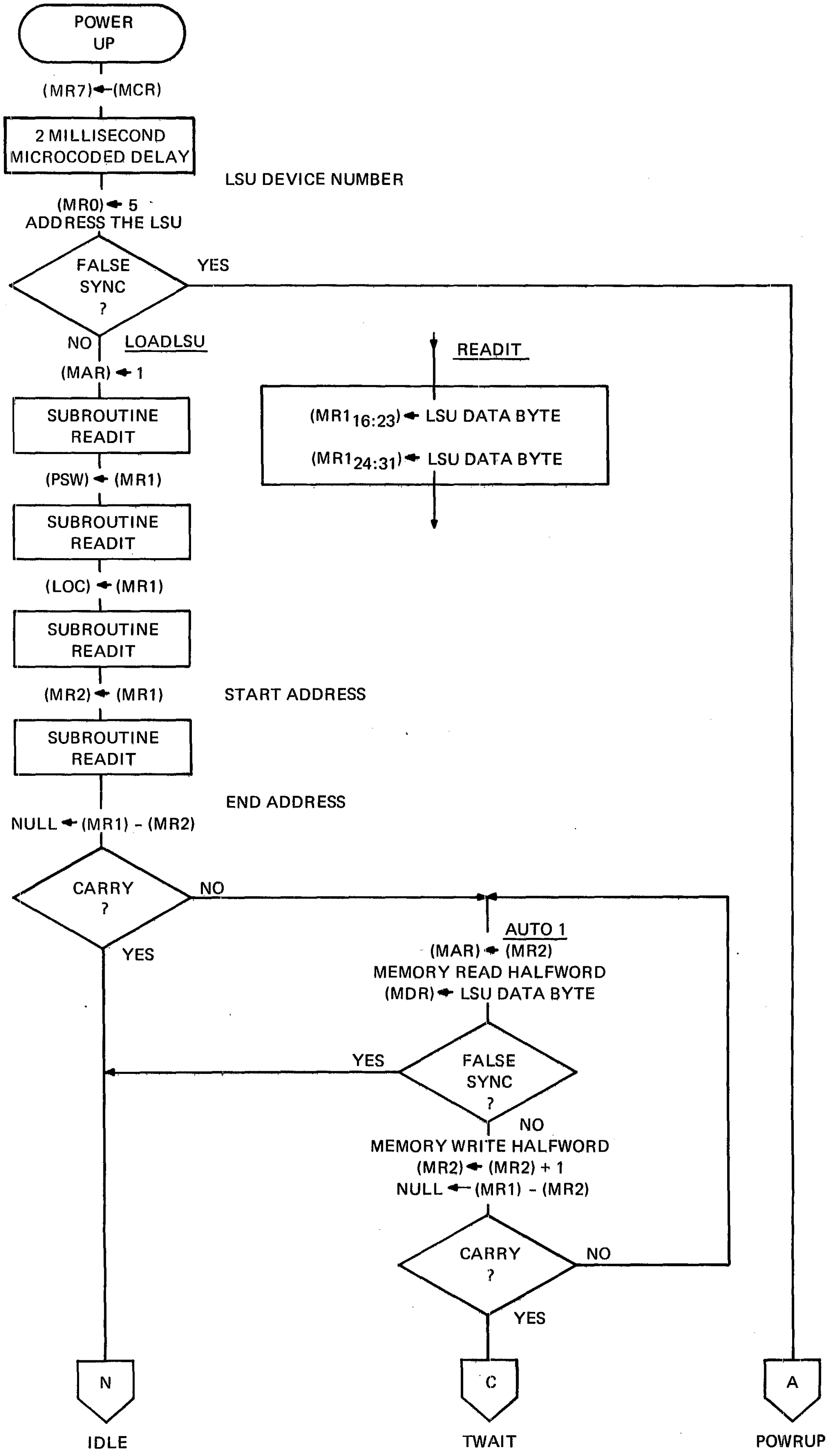
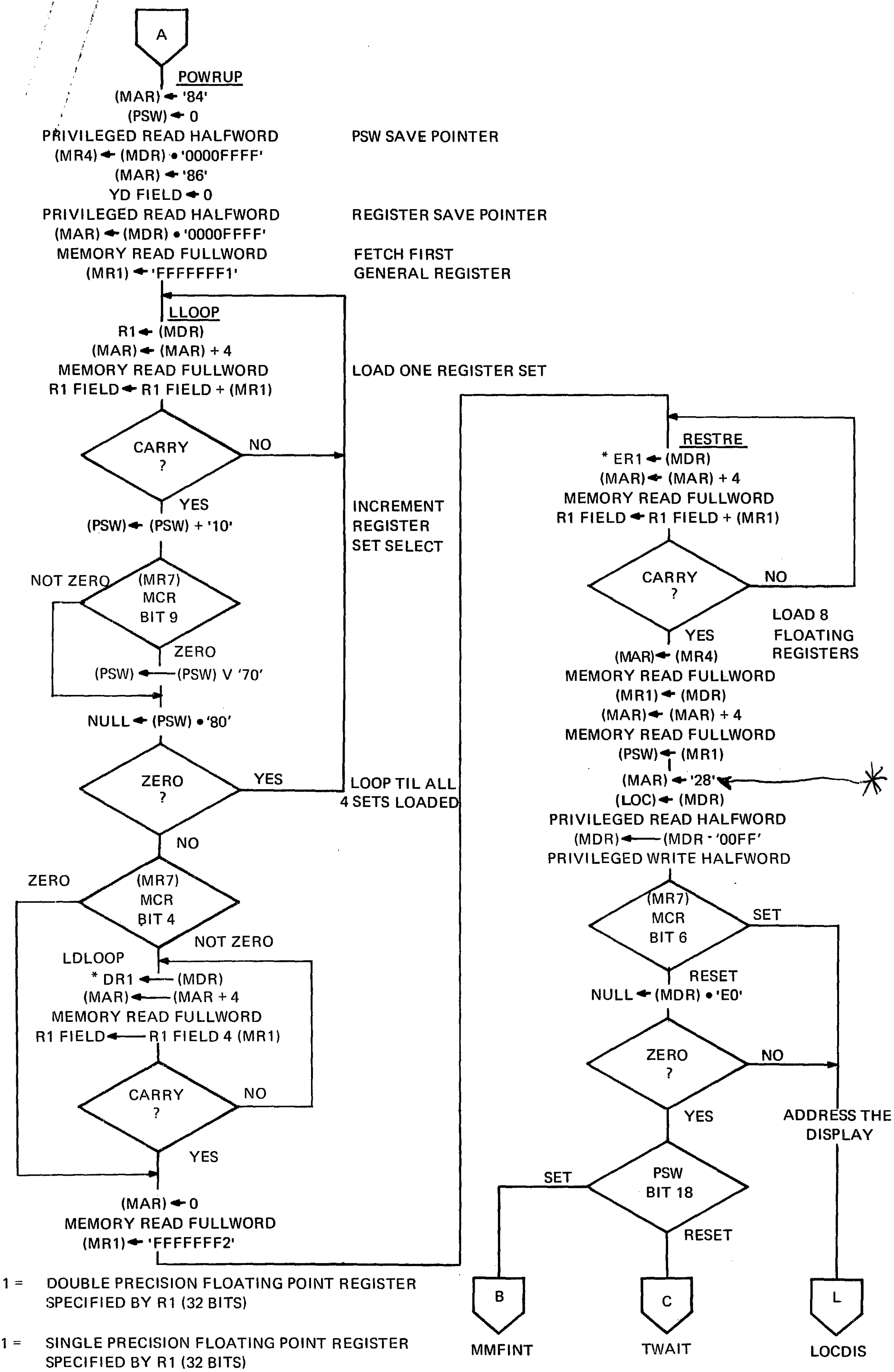


Figure 4. Power Up



* DR1 = DOUBLE PRECISION FLOATING POINT REGISTER SPECIFIED BY R1 (32 BITS)

* ER1 = SINGLE PRECISION FLOATING POINT REGISTER SPECIFIED BY R1 (32 BITS)

Figure 4. Power Up (Continued)

If the LSU is present, the micro-program proceeds to read the first eight bytes of data from the LSU. The nature of this data is as follows:

First two bytes	=	Least significant 16-bits of a new PSW
Second two bytes	=	Least significant 16-bits of a new LOC
Third two bytes	=	Least significant 16-bits of a start address
Fourth two bytes	=	Least significant 16-bits of an end address

The most significant 16-bits of PSW and LOC are set to zero. As a consequence, the Location Count value can only address a location within the first 64 KB of main memory. The start and end addresses identify an area in the first 64KB of main memory to be loaded with the ninth and successive bytes of data from the LSU.

If the start address is initially greater than the end address, the IDLE Loop is entered. Otherwise, data bytes are read from the LSU and stored at successive byte locations in main memory. The start address is incremented by one for each byte read. Reading continues until the start address becomes greater than the end address, at which time routine TWAIT is entered.

Routine POWRUP. Routine POWRUP is entered after power-up if no Loader Storage Unit is present. The halfword PSW save pointer is fetched from absolute address '84' and set aside in Micro-Register 4. Note that after doing a halfword main memory read, the halfword sign bit is propagated through the most significant 16 bits in MDR, necessitating 'ANDing' the fullword with '0000 FFFF' to keep the resultant address within the first 64 KB of main memory.

The halfword General Register save pointer is fetched from absolute address '86' and copied into the Memory Address Register (MAR). The register save pointer is the starting address in the first 64 KB of the General Register save area. General Register Set 0 is restored, followed by Register Set 7 or Register Sets 1 through 7 depending on the state of MCR bit 9. Then if MCR bit 4 is set, the 8 Double precision floating point Registers are restored from memory locations immediately following the General Register Save area. After this, the eight even-numbered floating-point registers are restored from absolute addresses 00 through 1F.

The saved PSW is fetched from the address retained in Micro-Register 4 and set aside in Micro-Register 1. The saved LOC is fetched from the address plus four contained in MR4. The PSW and LOC are then restored.

* The byte location at absolute address 28 is reset, then the saved Console status byte at absolute address 29 is fetched and examined. If the console was not in the RUN mode before power went down, or if the initialize switch on the console is depressed, the display is addressed and routine LOCDIS is entered. If the display was in the RUN mode, the Machine Malfunction enable bit in PSW (PSW18) is tested. If set, routine MMFINT is entered. If reset, the user instruction whose address is in LOC is fetched and executed unless the WAIT bit (Bit 16) of PSW is set, in which case routine WAIT is entered.

9.2 Interrupt Support

Routine MMFINT. Routine MMFINT, shown on Figure 5, is entered when a memory parity error or an early power failure is detected or if, after the normal power up sequence, the micro-code determines that the display had been in the Run mode before power went down and the Automatic Restart option is present and PSW Bit 18 is set. The PSW is saved in absolute location '20', the LOC is saved in absolute location '24', and a new PSW and LOC are fetched from absolute locations '38' and '3C' respectively. The least significant three bits from the MCR are 'ORed' into the Condition Code field of the new PSW, and routine TWAIT is entered.

Routine TWAIT. This routine tests the WAIT bit of the current PSW (PSWI6). If set, the WAIT loop is entered. If reset, the user instruction whose address is in LOC is fetched and executed.

WAIT Loop. The WAIT loop is a high speed loop consisting of a single micro-instruction branching to itself. The interrupts are collectively armed and the occurrance of any interrupt will cause the micro-instruction at its respective trap location to be executed.

Also shown on Figure 5 is a Common Interrupt routine (COMINT) shared by Memory Access Controller interrupts, Illegal Instruction interrupts, Arithmetic fault and Queue Service interrupts. The routine fetches a new PSW and LOC from dedicated memory locations then saves the old PSW and LOC in General Registers 14 and 15 respectively of the register set selected by the new PSW.

The three instructions capable of causing a System Queue interrupt are also shown on Figure 5. These are LPSW, LPSWR and EPSR. After performing the instruction, Routine TEST1 is entered. There the number of slots used, halfword at the absolute address plus two of the fullword address contained in absolute location '80', is examined. If the halfword is zero, routine TWAIT is entered. If non-zero, the system queue is not empty. A new PSW and LOC are fetched from absolute location '88'. The old PSW and LOC are saved in General Registers 14 and 15 of the register set selected by the new PSW and the address of the System Queue is saved in General Register 13.

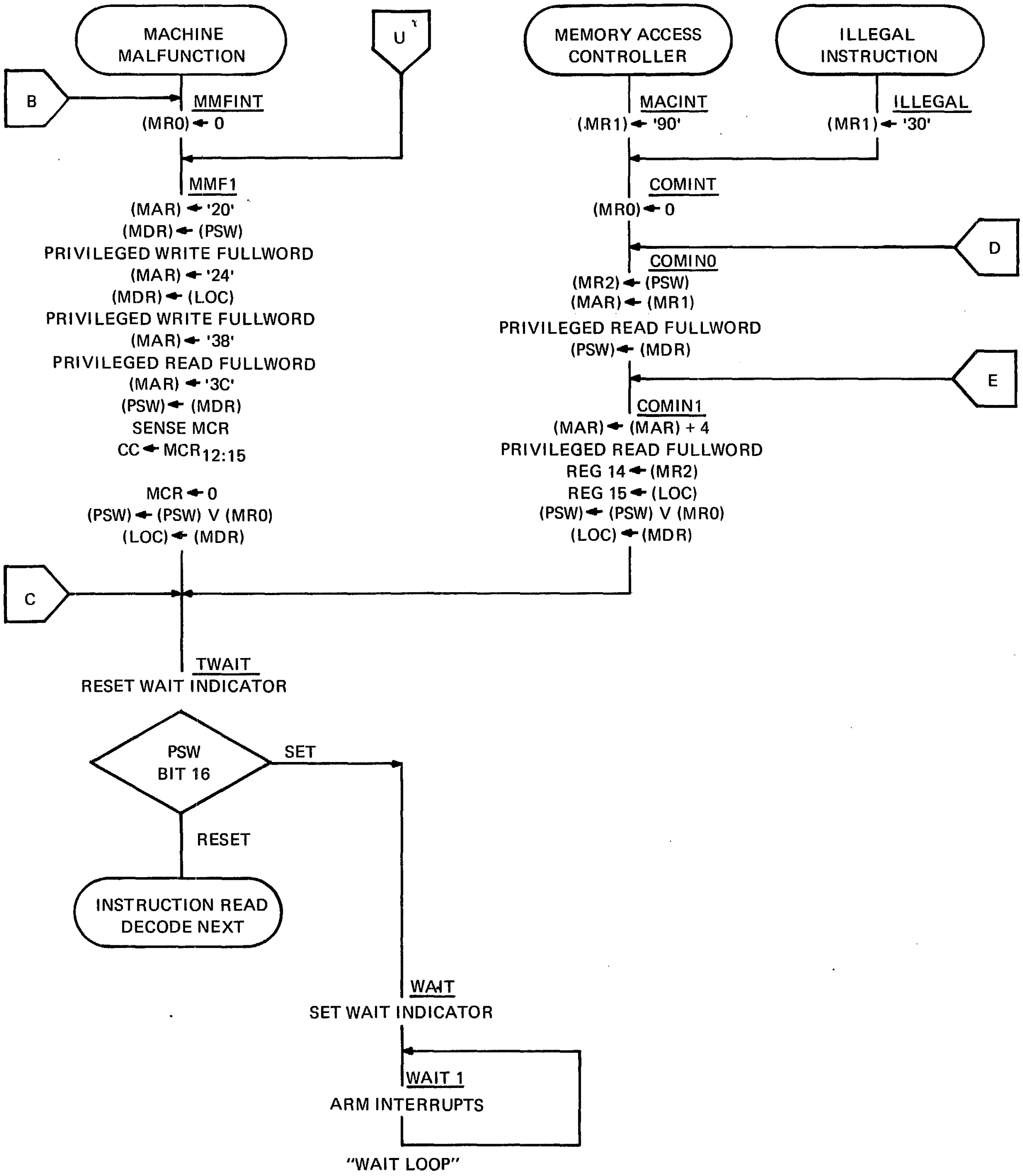


Figure 5. Interrupt Support

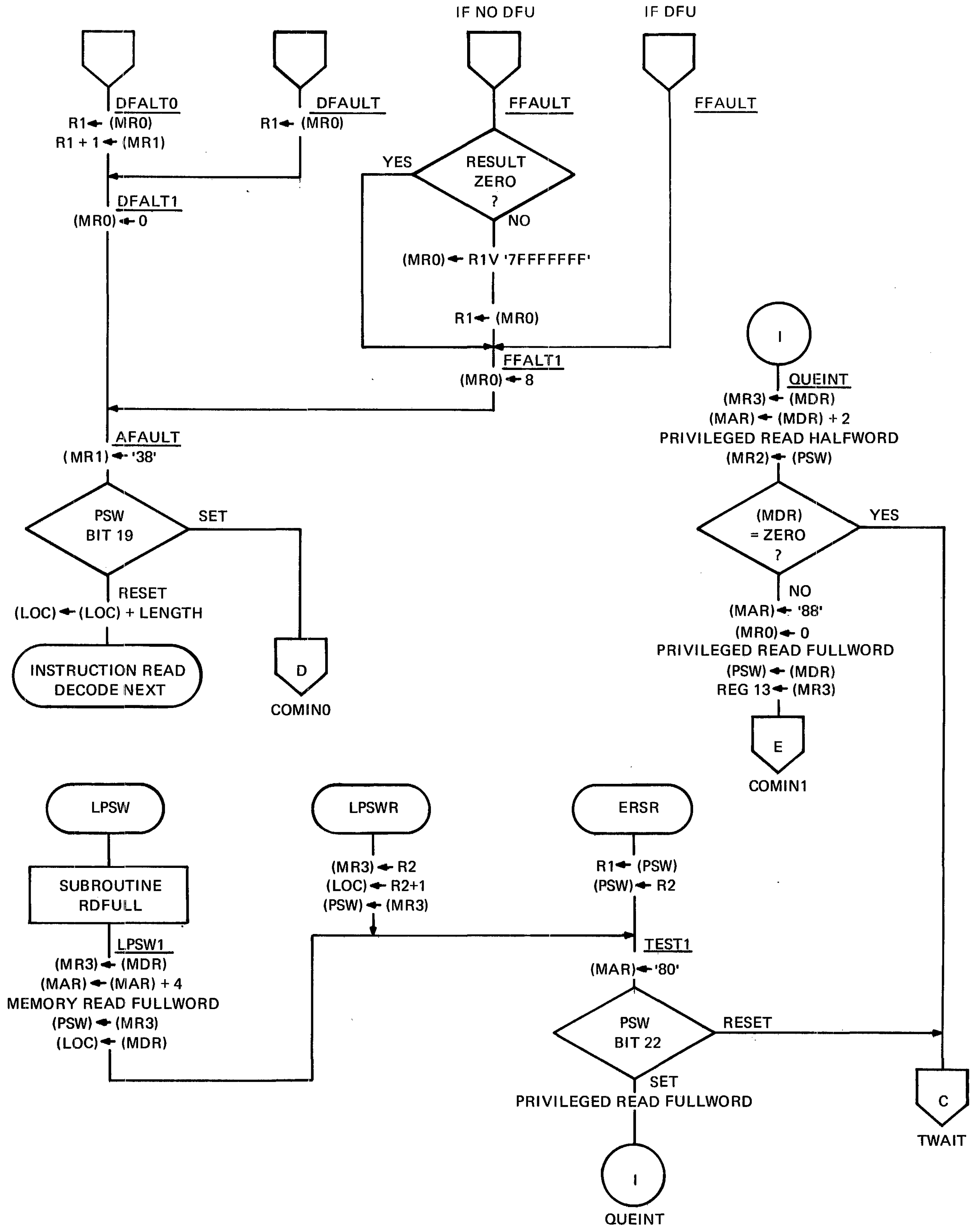


Figure 5. Interrupt (Continued)

9.3 Console Support

The Routine CONSER handles all the console service functions. This routine is entered when a Console Attention is generated, or at the conclusion of a user level instruction if the signal SNGL is active. (Refer to Figure 6.)

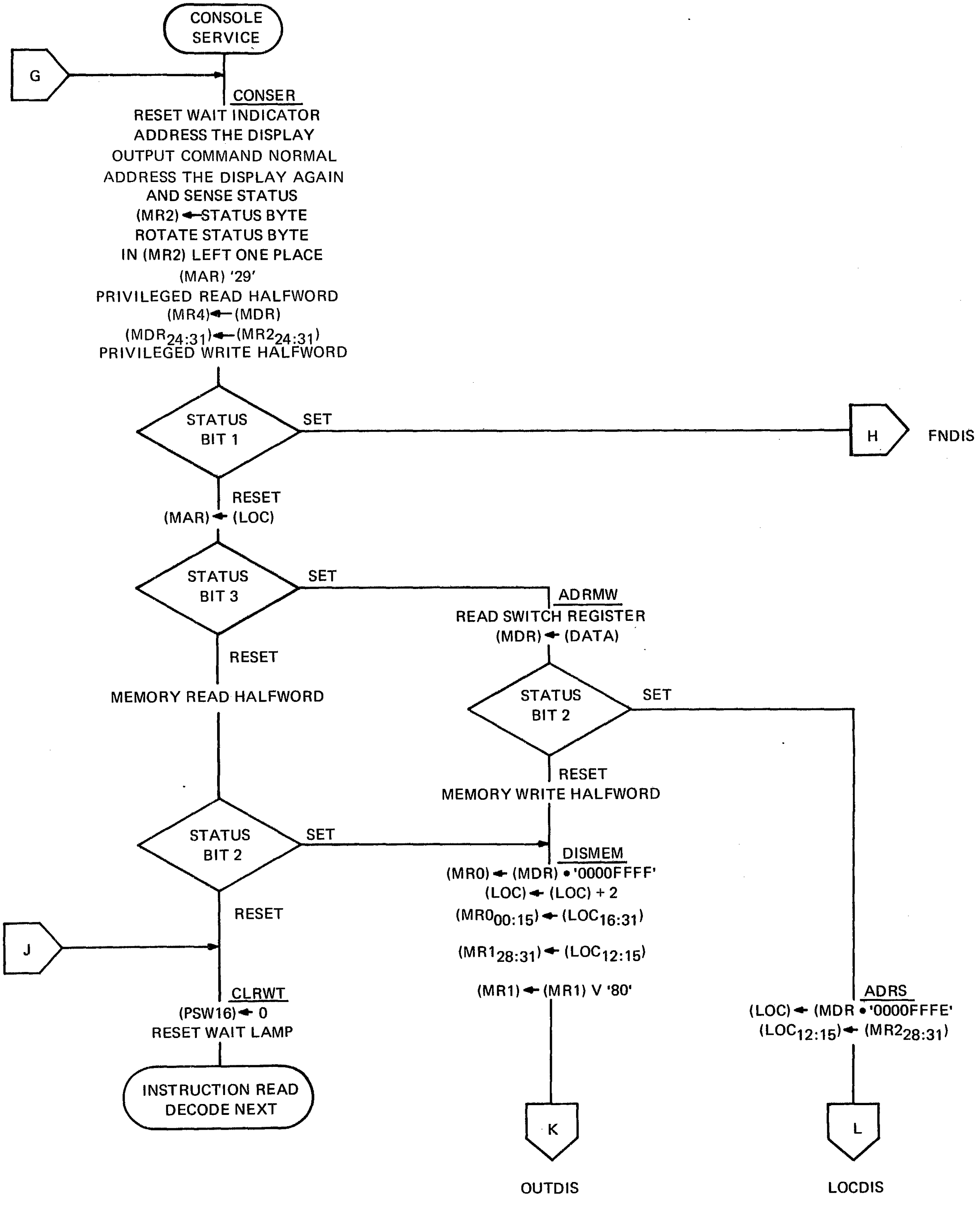


Figure 6. Console Support

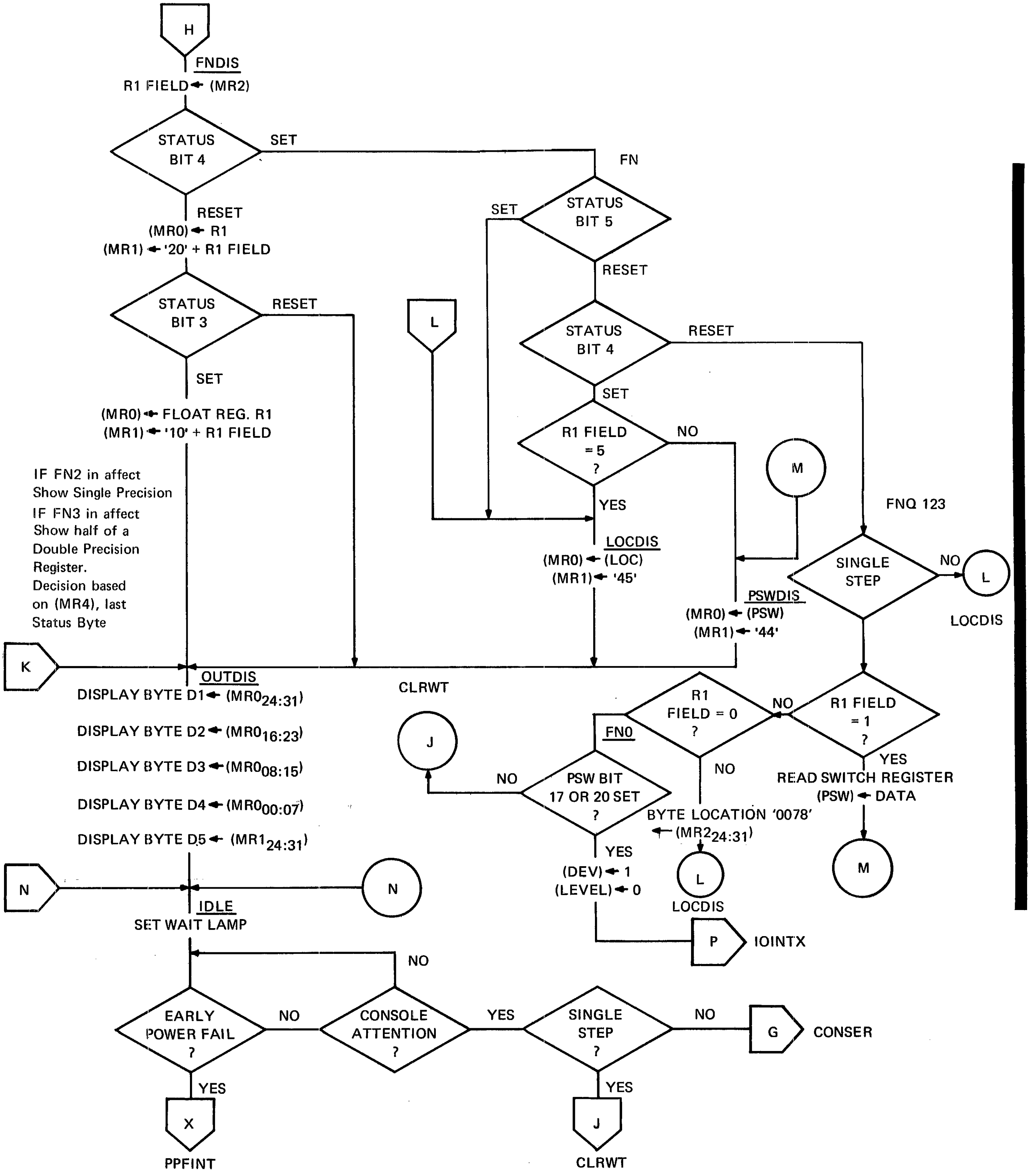


Figure 6. Console Support (Continued)

The WAIT indicator on the console is reset, the display controller is addressed and the status sensed. The status byte is copied into Micro-Register 2 and rotated left one position so that MR2 Bits 24:31 have the following significance:

	24	25	26	27	28	29	30	31	
Run	0	0	0	X	X	X	X	X	
Memory Write	0	0	1	X	X	X	X	X	
Memory Read	0	1	0	X	X	X	X	X	
Address	0	1	1	X	MS 4 ADDRESS BITS				
Fixed Register	1	0	0	1	REGISTER NO.				Single or Halt
Floating Register	1	0	1	1	REGISTER NO.				
Function	1	0	0	0	FUNCTION NO.				

MR2 is saved in the absolute byte location 29 so that the power-up sequence can determine what mode the console was last placed in before power down.

The status byte in MR2 is then examined. If status Bit 1 is set (status Bit 1 corresponds to MR2 Bit 24), a register has been selected for display or a function has been selected. Routine FNDIS is entered. If status Bit 1 is reset, Address, Memory Read, Memory Write or Run mode has been selected. If Address or Memory Write, the least significant 16 switch register bits are read into MDR. If it is 'Memory Write', the halfword in MDR is written into the main memory location whose address is the contents of LOC. Routine DISMEM is entered.

If Address, Bits 28:31 of MR2 are appended to the top of the halfword in MDR, becoming Bits 12:15 of the new location count value. This 20-bit address is forced even and copied to LOC. Routine LOCDIS is then entered.

If the status was Memory Read, the halfword whose address is the contents of LOC is read and Routine DISMEM is entered.

Routine DISMEM increments LOC then packs MR0 and MR1 with the contents of MDR and LOC. Routine OUTDIS is then entered.

Routine FNDIS determines if the selected function is legal. The only legal functions are 0, 1, 2, 3, 4 and 5. If the function is illegal, Routine LOCDIS is entered. If the function is zero, and PSW Bit 17 or 20 is set, an interrupt from Device Number 1 is simulated through Routine IOINTX. If the function is one, the least significant 16 switch register bits are copied into the least significant 16-bits of PSW. The most significant 16-bits of PSW are unchanged. Routine PSWDIS is then entered. If the function is 2 or 3, (MR2) is saved in byte location 00028 and LOCDIS is entered. If the function is four, Routine PSWDIS is also entered. There, the current PSW is copied to MR0 and MR1 is set to 44 . Routine OUTDIS is entered.

If the function is five, routine LOCDIS is entered. There, the current LOC is copied to MR0, and MR1 is set to '45'. Routine OUTDIS is entered.

Routine OUTDIS writes the contents of MR0 and MR1, a total of five bytes, to the display.

Routine IDLE sets the WAIT indicator, then the high-speed Idle loop is entered. This loop can only be exited if a power-fail is detected or another console attention is generated. If a power-fail occurs, routine PPFINT is entered. If a console attention is detected, MCR Bit 7 is tested. If reset, the console is not in single mode. The micro-program re-enters routine CONSER. If MCR Bit 7 is set, the console is in single step mode. Routine CLRWT is entered.

Routine CLRWT resets PSW Bit 16, the wait bit, and an instruction read is performed from the address specified by LOC.

9.4 I/O Interrupts (Refer to Figure 7.)

The occurrance of one of the four I/O Interrupts causes the micro-instruction at the respective trap location to be executed. Register "LEVEL" is set equal to the number of the interrupt line and the interrupt is acknowledged. The returned device number is placed in register "DEV" and Routine IOINTX is entered.

The device number in "DEV" is masked to the least significant 10 bits. It is then doubled and added to X'D0', forming the address of the appropriate interrupt service pointer. The halfword service-pointer table entry is fetched. The current PSW is set aside in register "TEMP", then Bits 18 and 20 are set and all others reset. The register set number specified by register "LEVEL" is copied to the register set select field of PSW. General Register 0 of the newly selected set is set equal to the old PSW; General Register 1 is set equal to LOC, and General Register 2 is set equal to the device number. The device is addressed and a sense status is performed. The device status byte is copied to General Register 3. The service-pointer table entry, in MDR, is copied to LOC and the least significant bit is examined. If reset, an "immediate interrupt" is to be performed. The user instruction whose address is LOC is fetched and executed. If the least significant bit of the service-pointer table entry is set, the service-pointer is the address in the first 64 KB of a Channel Command Block (CCB). Routine CHANEL is entered.

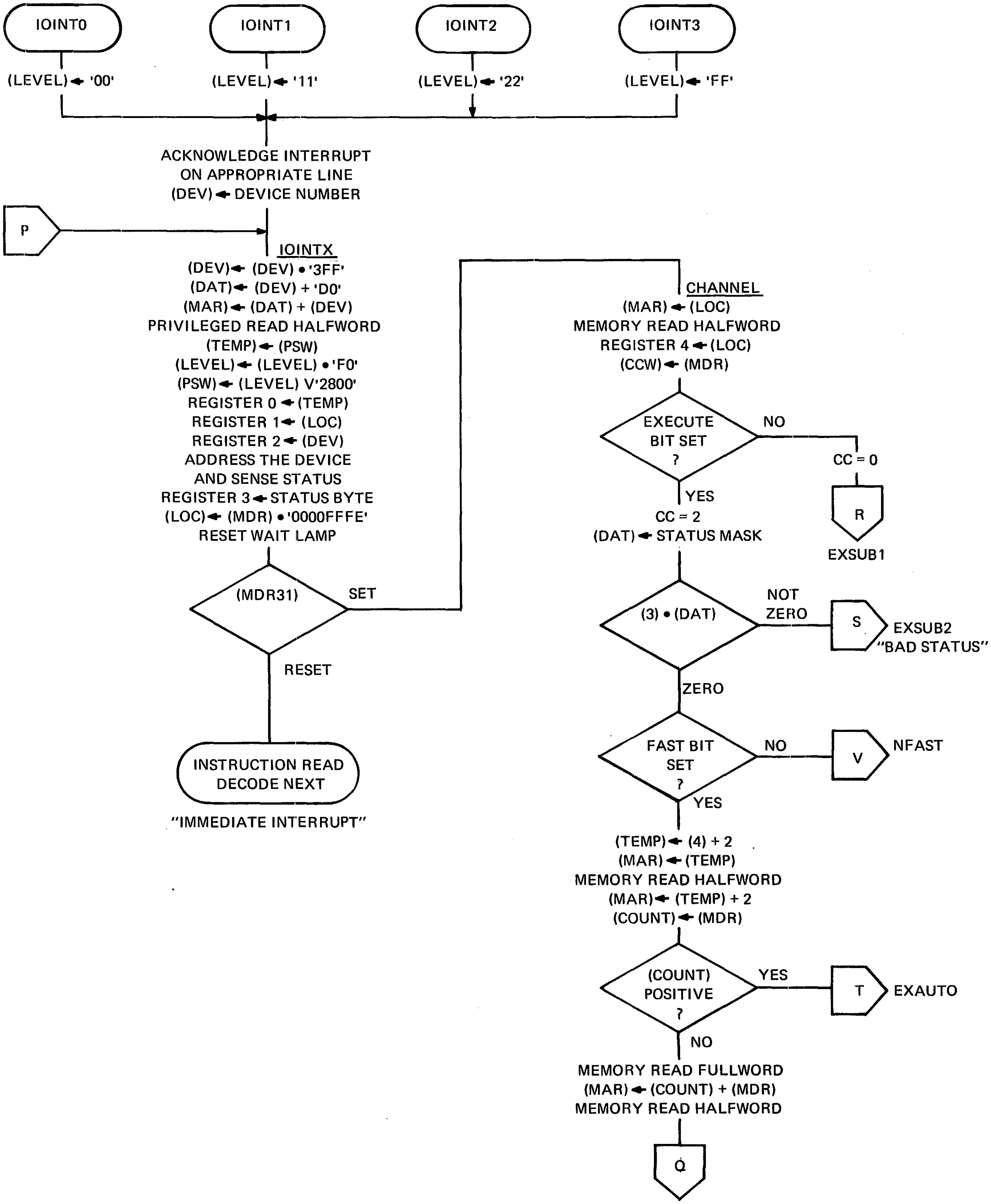


Figure 7. I/O Interrupts

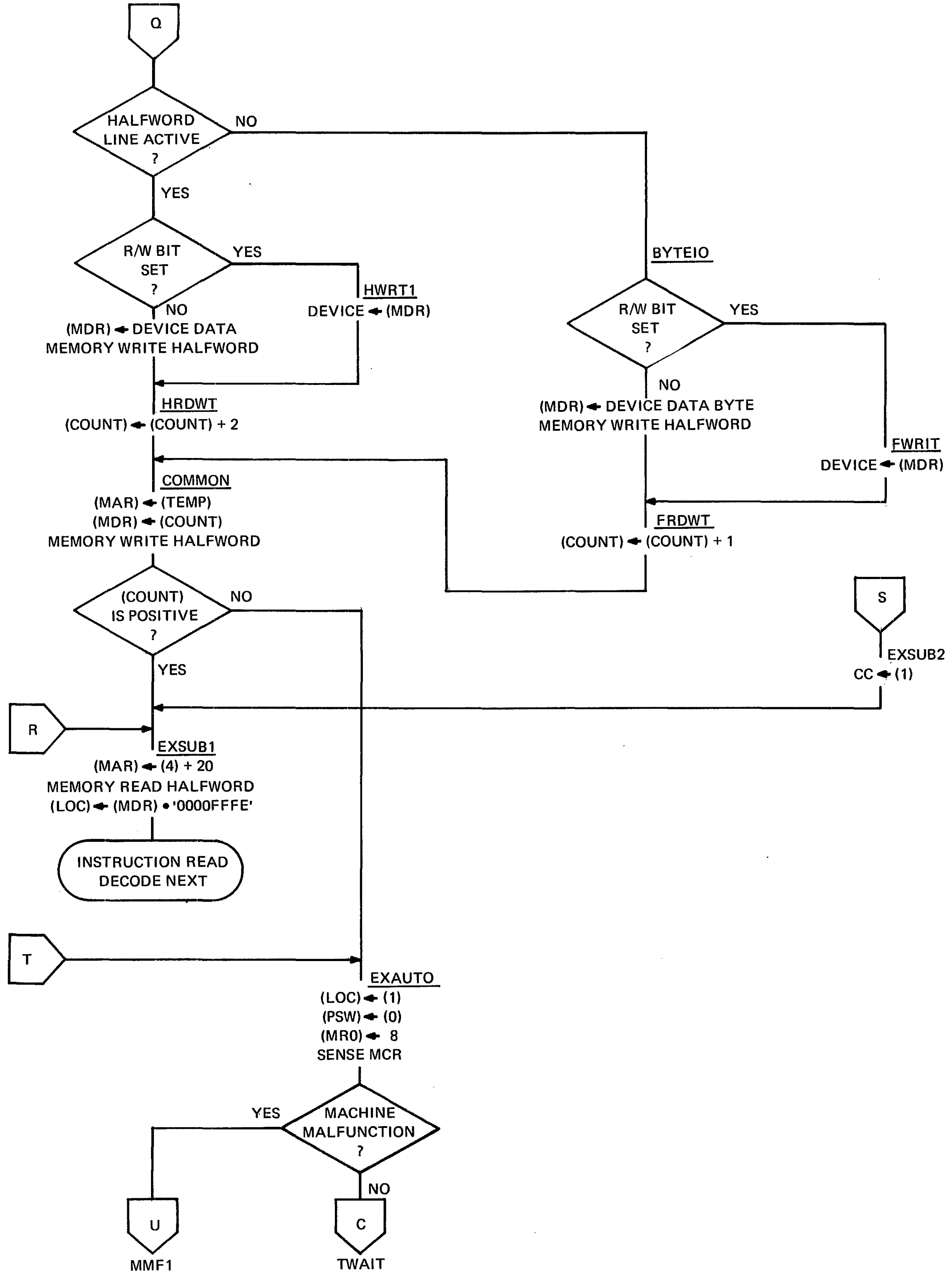


Figure 7. I/O Interrupts (Continued)

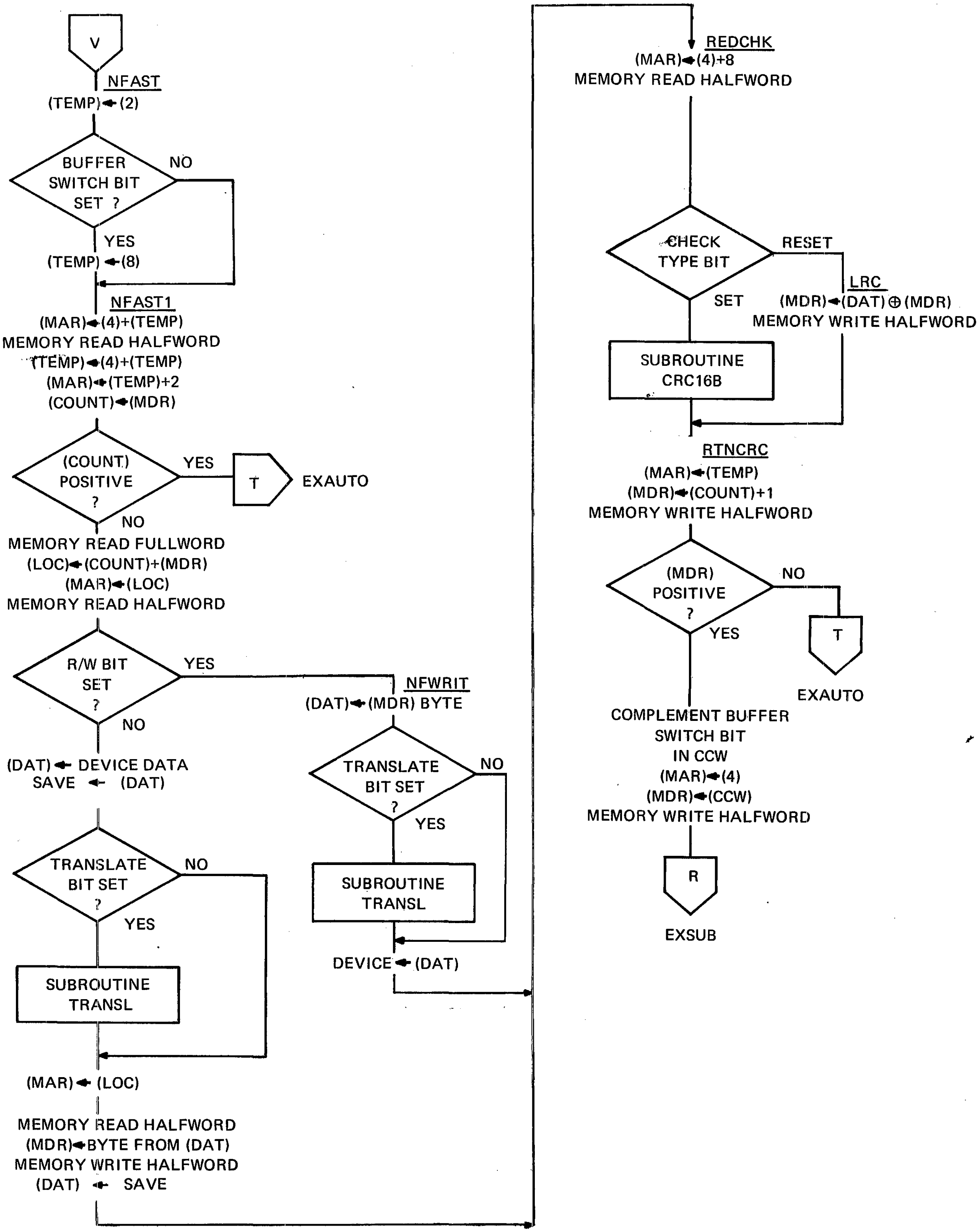


Figure 7. I/O Interrupts (Continued)

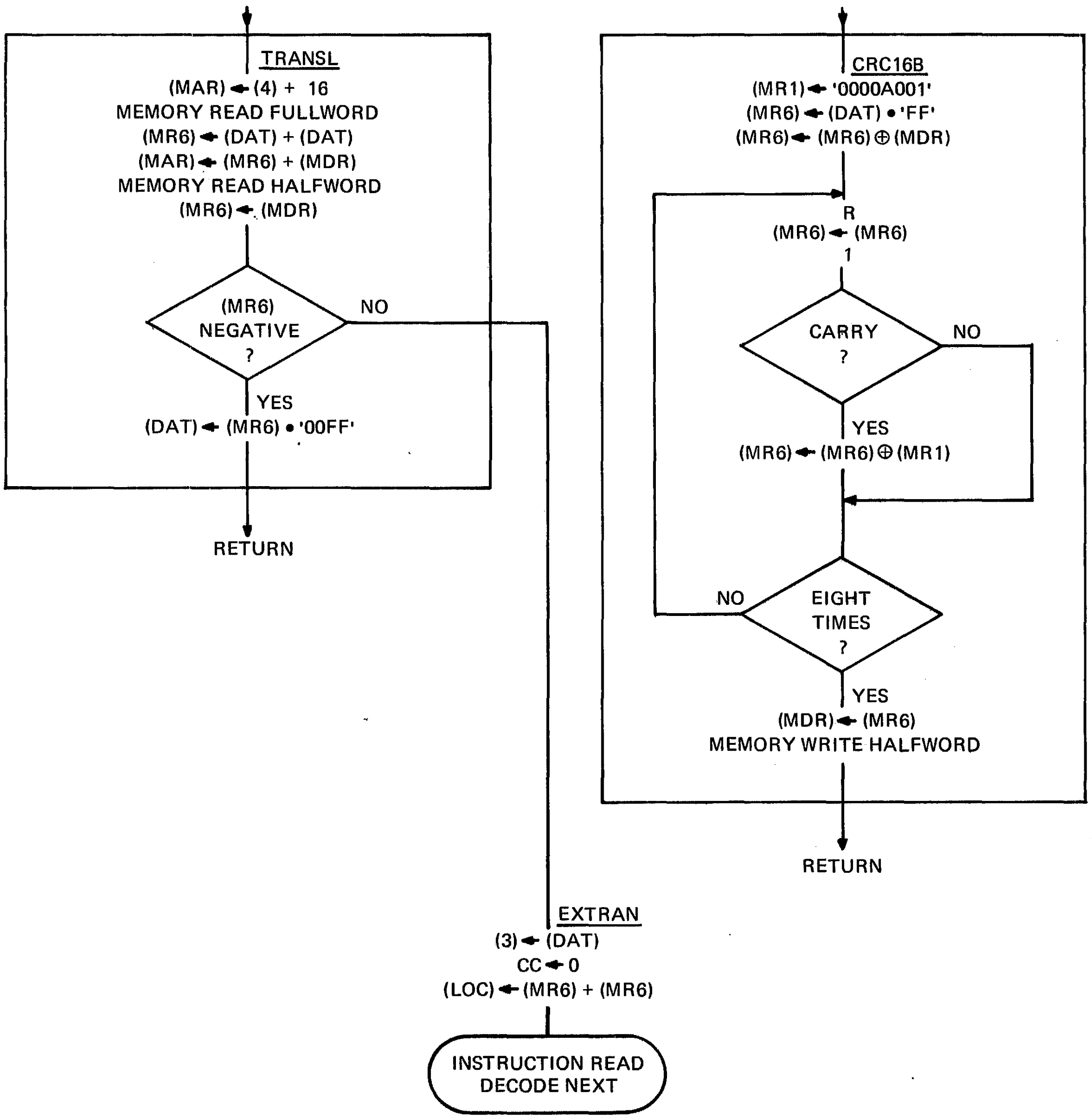


Figure 7. I/O Interrupts (Continued)

Routine CHANEL can perform a variety of functions, depending upon bits in the Channel Command Word (CCW) which is the first halfword in the CCB. (See Figure 8.)

	0 … 7	8	9 10	11	12	13	14	15
0	STATUS MASK	E	▨	C	B	R/W	T	F
2	BUFFER 0 BYTE COUNT							
4	BUFFER 0 END ADDRESS							
8	CHECK WORD							
10	BUFFER 1 BYTE COUNT							
12	BUFFER 1 END ADDRESS							
16	TRANSLATION TABLE ADDRESS							
20	SUBROUTINE ADDRESS							

Channel Command Word

Figure 8. Channel Command Block

The Channel Command Word is fetched and placed in a register labeled CCW. Setting of the Condition Code is enabled and the 'EXECUTE' bit of CCW is tested. If reset, Routine EXSUB1 is entered (the Condition Code is clear), if set, the G flag sets. The micro-code then ANDS the status mask with the actual device status in General Register 3. If any bit matches, the status check has failed. Routine EXSUB2 is entered. If the status test is good, the 'FAST' bit in CCW is tested. If not set, Routine NFAST is entered for 'normal mode' CCB activities. If reset, the Buffer 0 byte count is fetched and copied to register "COUNT". If the count is already positive, the assumption is that it has not yet been set up, so Routine EXAUTO is entered. If the count is zero or negative, the Buffer 0 end Address is fetched and added to the count. The halfword addressed is fetched and the halfword test line is examined. If inactive, routine BYTEIO is entered. If active, a halfword device controller is addressed. A Read Halfword or Write Halfword, depending upon the state of the R/W bit in the CCW, is performed and the count is incremented by two. Routine COMMON is entered. At Routine BYTEIO, a single byte is transferred, the count is incremented by one and Routine COMMON is entered. There, the incremented count is restored to the CCB and if not positive, Routine EXAUTO is entered. If the count is positive, Routine EXSUB1 is entered.

Routine EXAUTO restores PSW and LOC to return control to the point in the user program where the interrupt occurred. If a Machine Malfunction Interrupt is present, Routine MMF1 is entered. If no MALF, the wait bit of PSW is tested. If set, Routine WAIT is entered; or else the user instruction whose address is LOC is fetched and executed.

Routine EXSUB2 is entered if the device status is improper. There, the L flag in PSW is forced set and Routine EXSUB1 is entered. There, the subroutine address from the CCB is fetched and copied to LOC and an Instruction Read is performed.

At Routine NFAST, the buffer switch bit in the CCW is examined to determine which Buffer byte count and End address to use. The appropriate Buffer byte Count is fetched and copied to Register "COUNT". If the count is already positive, Routine EXAUTO is entered. If not positive, the byte count is added to the Buffer End address. The halfword addressed is fetched from main memory.

The R/W bit in the CCW is tested to determine Input or Output. If Output, and the Translate bit in the CCW is set, the byte to output is first translated by way of subroutine TRANSL. After returning from TRANSL the translated byte is output to the device and Routine REDCHK is entered. There, depending upon the TYPE bit in the CCW, either a longitudinal (Exclusive-OR) checksum is performed or a 16th order cyclic redundancy check is performed. The result is returned to the check-word of the CCB. Routine NFRW is then entered.

If input, the data byte is input from the device and saved. If the Translate bit in the CCW is set, the byte is translated by way of subroutine TRANSL. After returning from TRANSL, the translated byte is stored in main memory. Register DAT is set equal to the saved untranslated byte and Routine REDCHK is entered.

The count is incremented by one and if not yet positive, Routine EXAUTO is entered. If the count has become positive, the buffer switch bit in the CCW is complemented and Routine EXSUB1 is entered.

9.5 Power Fail Interrupt

In response to a power fail interrupt (PPF), the micro-program enters Routine PPFINT, shown on Figure 9. This routine is a reversal of the power up routine. The PSW and LOC are saved in their power fail save locations, all sets of General Registers and, if present, the eight Double Precision floating point Registers are saved in their power fail block of memory and the eight single-precision floating-point registers are saved at memory locations '000' through '01F'. After this, the Power Down micro-instruction is issued to release the system clear relay and initialize the system.

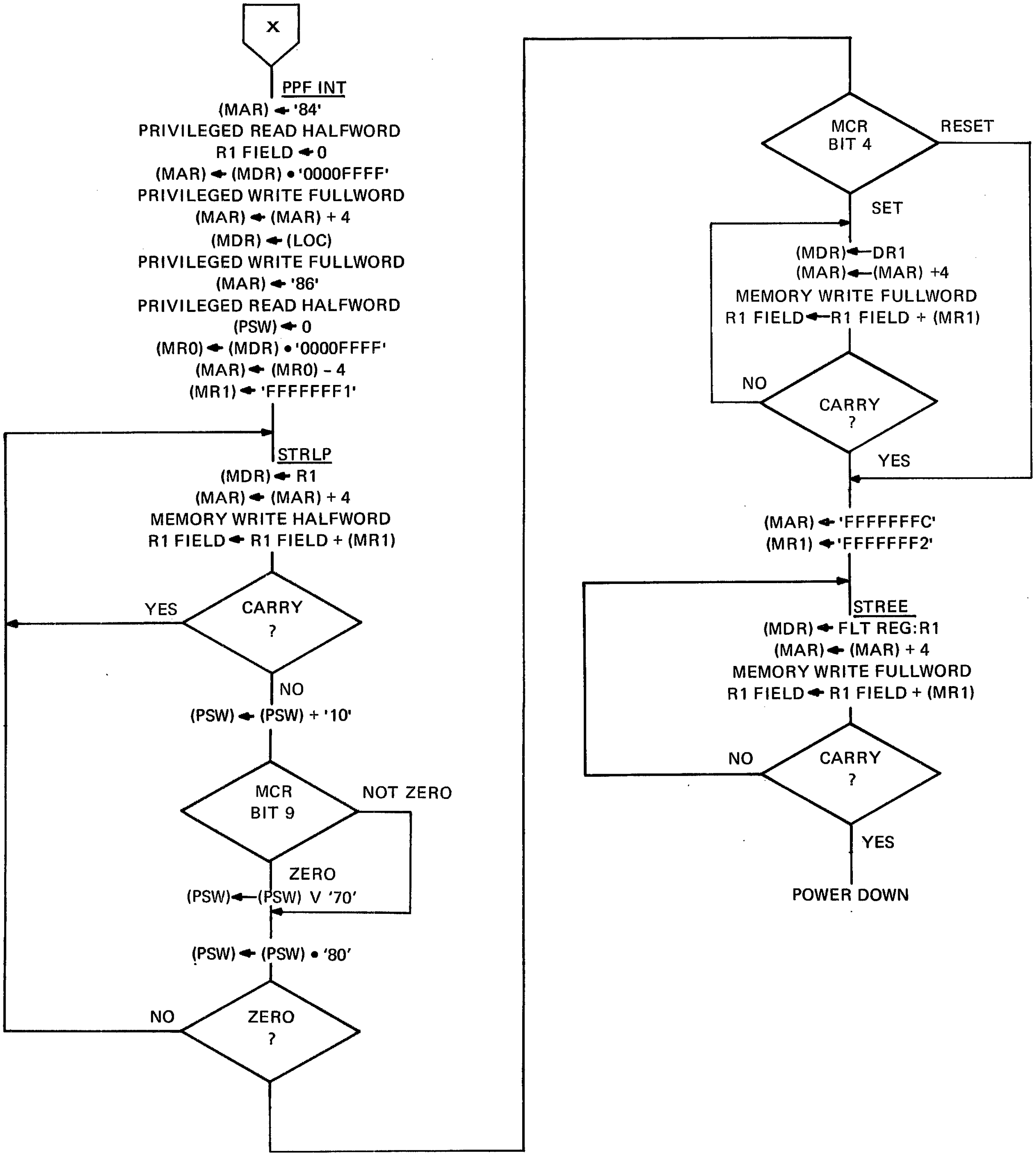


Figure 9. Power Fail Interrupt

```
                         OPT   SCRT,GO,CROSS,SQCHK
                 *
                 * COPYRIGHT INTERDATA INC. SEPTEMBER,  1975
                 *
                 * SEPTEMBER 4, 1975
                 * IRA GABBERT
                 *
                 *
                 *
                 * * * * * * * * * * * * * * * * * * * * * * * * * * * *
                 *                                                     *
                 * PROGRAM COMPRISES THE FOLLOWING ROM CHIPS:          *
                 *                                                     *
                 * 19-142R00F01,19-142R00F02,19-142R02F03,19-142R01F04 * R01,R02
                 * 19-142R00F05,19-142R01F06,19-142R01F07,19-142R00F08 *     R02
                 *                                                     *
                 * 19-142R00F09,19-142R00F10,19-142R01F11,19-142R01F12 *     R02
                 * 19-142R00F13,19-142R01F14,19-142R00F15,19-142R00F16 *     R02
                 *                                                     *
                 * 19-142R00F17,19-142R00F18,19-142R00F19,19-142R00F20 *
                 * 19-142R00F21,19-142R00F22,19-142R00F23,19-142R00F24 *
                 *                                                     *
                 * 19-142R00F25,19-142R00F26,19-142R00F27,19-142R00F28 *
                 * 19-142R00F29,19-142R00F30,19-142R00F31,19-142R00F32 *
                 *                                                     *
                 * 19-142R01F33,19-142R01F34,19-142R00F35,19-142R01F36 *     R02
                 * 19-142R01F37,19-142R01F38,19-142R00F39,19-142R01F40 *     R02
                 *                                                     *
                 * * * * * * * * * * * * * * * * * * * * * * * * * * * *
                 *
                 *
                 *
                 * IN ALL CASES WHERE A BRANCH OR TRANSFER COULD OCCUR TO A
                 * LISTING PAGE OTHER THAN THE CURRENT LISTING PAGE, THE TARGET
                 * PAGE NUMBER IS SHOWN IN PARENTHESIS IN THE COMMENT FIELD.
                 *
                 *                                   ON POWER-UP, OR AFTER
                 *                                   INITIALIZE, MICRO CODE
000  321F 1005           L1    MR0,5                 EXECUTION BEGINS AT '001'
001  17FD 0000           BALD  TLSU(NULL)            GO TO POWER UP ROUTINE (P.43)
                 *
                 *
                 * * * * * * * * * * * * * * * * * * * * * * * * * * * *
                 *                                                     *
                 * USER  LEVEL  INSTRUCTION  EMULATION  ENTRY-POINTS   *
                 * FOLLOW.  IN RESPONSE TO AN INSTRUCTION READ COMMAND *
                 * THE HARDWARE, READS THE NEXT USER INSTRUCTION FROM  *
                 * THE MAIN MEMORY LOCATION SPECIFIED BY (LOC). TWO,   *
                 * FOUR OR SIX BYTES ARE READ, DEPENDING UPON THE      *
                 * INSTRUCTION TYPE.  TWICE THE USER'S OPERATION CODE  *
                 * IS THE STARTING ADDRESS IN ROM OF THE APPROPRIATE   *
                 * EMULATION SEQUENCE.  THE OP-CODE IS  SHOWN IN THE   *
                 * COMMENT FIELD AND THE USER'S MNEMONIC IS THE LABEL. *
                 *                                                     *
                 * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

```
                   *                                                                   83200580
                   *                                                       * 01 *      83200590
002  2A1F 1C01  BALR   L      MR0,YS,IL                 SAVE BRANCH ADDRESS            83200600
003  233F 1D05         LX     YD,LOC,BALR1              INCREMENTED LOC TO YD          83200610
                   *                                                       * 02 *      83200620
004  17EC 01D9  BTCR   BALT   BRR(NULL),ILIR,D          BRANCH IF MASK TRUE            83200630
005  235F 1B37  BALR1  LX     LOC,MR0,BC2               LOAD LOC (P.3)                 83200640
                   *                                                       * 03 *      83200650
006  13EC 01D9  BFCR   BALF   BRR(NULL),ILIR,D          BRANCH IF MASK FALSE           83200660
007  235F 1C37  BRR    LX     LOC,YS,BC2                LOAD LOC (P.3)                 83200670
                   *                                                       * 04 *      83200680
008  2B39 5C39  NR     N      YD,YD,YS,ILIR,E,D                                        83200690
009  0000 0000         DC     0,0                                                      83200700
                   *                                                       * 05 *      83200710
00A  2BF9 0C39  CLR    S      NULL,YD,YS,ILIR,E,D                                      83200720
00B  233B FC0D  DR1    DX     YD,YDP1,YS,DR2            DO DIVIDE                      83200730
                   *                                                       * 06 *      83200740
00C  2B39 7C39  OR     O      YD,YD,YS,ILIR,E,D                                        83200750
00D  13F4 B110  DR2    BALV   DFAL10(NULL),D            (P.28)                         83200760
                   *                                                       * 07 *      83200770
00E  2B39 6C39  XR     X      YD,YD,YS,ILIR,E,D                                        83200780
00F  0001 0000  BIT15  DC     '00010000'                                               83200790
                   *                                                       * 08 *      83200800
010  2B3F 1C39  LR     L      YD,YS,ILIR,E,D                                           83200810
011  0000 0000         DC     0,0                                                      83200820
                   *                                                       * 09 *      83200830
012  2B7F 1C00  CR     A      MDR,NULL,YS               2ND OP TO MDR                  83200840
013  23F9 6DBC  C1     XX     NULL,YD,MDR,C2            COMPARE SIGNS (P.3)            83200850
                   *                                                       * 0A *      83200860
014  2B39 1C39  AR     A      YD,YD,YS,ILIR,E,D                                        83200870
015  FFFF 0000  HIHALF DC     'FFFF0000'                                               83200880
                   *                                                       * 0B *      83200890
016  2B39 0C39  SR     S      YD,YD,YS,ILIR,E,D                                        83200900
017  0000 FFFF  LOHALF DC     '0000FFFF'                                               83200910
                   *                                                       * 0C *      83200920
018  1398 D6C0  MHR    BAL    MHR1(MAR)                 (P.37)             R01         83200930
019  0000 0000         DC     0,0                                                      83200940
                   *                                                       * 0D *      83200950
01A  1398 D8C0  DHR    BAL    DHR1(MAR)                 (P.37)             R01         83200960
                                                                                       83200970
01B  17E0 3740  DER1   BALNZ DER2(NULL)                 CONTINUE IF NOT ZERO (P.8)     83200980
                   *                                                                   83200990
01C  33BD 700C  DEZ    OI     PSW,PSW,'C'               DIVISOR WAS ZERO, SET CC       83201000
01D  13F8 B300         BAL    FFALT1(NULL)              (P.28)                         83201010
                   *                                                                   83201020
01E  6B38 8829  SER2   SE     YD,YS,MR0,ILIR,E          DO B MINUS A                   83201030
01F  13F4 B210         BALV   FFAULT(NULL),D            (P.28)                         83201040
```

```
                *                                                                       83201060
                *                                                           * 10 *      83201070
020  2B39 8EB9  SRLS   SRL    YD,YD,YS1,ILIR,E,D                                        83201080
021  0000 4800  BI1720 DC     '00004800'                                                83201090
                *                                                           * 11 *      83201100
022  2B39 9EB9  SLLS   SLL    YD,YD,YS1,ILIR,E,D                                        83201110
023  FFFF 7FFF  BIT160 DC     'FFFF7FFF'                                                83201120
                *                                                           * 12 *      83201130
024  321D 5008  CHVR   NI     MR0,PSW,8              SAVE PREVIOUS CARRY                83201140
025  3658 7015         OI     MR2,YS,H1HALF,I        SET MS 16 BITS                     83201150
026  3652 6327         XI     MR2,MR2,BIT16,I        INVERT THE SIGN BIT                83201160
027  3652 1327         AI     MR2,MR2,BIT16,I        AND EXTEND IT                      83201170
028  2A38 6909         X      MR1,YS,MR2,ILIR        RE-CREATE HW OVERFLOW BIT          83201180
029  37F1 500F         NI     NULL,MR1,BIT15,I       BY COMPARING BITS 15 AND           83201190
02A  13E0 0B00         BALZ   CHVR1(NULL)            16 OF R2                           83201200
02B  3210 7004         OI     MR0,MR0,4              OVERFLOW IF DIFFER                 83201210
02C  2B3F 1920  CHVR1  L      YD,MR2,E               LOAD RESULT, ADJUST G & L          83201220
02D  2BBD 7810         O      PSW,PSW,MR0,D          OR IN C & V                        83201230
                *                                                                       83201240
02E  2B3F 1E00  LA1    L      YD,MAR                 EFFECTIVE ADRS TO R1      R02      83201250
02F  2BFF 1F9C         L      NULL,NULL,IR,D         FETCH NEXT INSTR.         R02      83201260
                *                                                           * 18 *      83201270
030  2A7F 1C00  LPSWR  L      MR3,YS                 SET NEW PSW ASIDE IN MR3           83201280
031  2BDF 3E80         AINC   YD1,NULL,YS1           POINT TO R2+1                      83201290
032  2B5F 1C80         L      LOC,YD                 LOAD NEW LOC                       83201300
033  13F8 A000         BAL    TEST1(NULL)            (P.27)                             83201310
                *                                                                       83201320
034  2BFF 1F81  BC1    L      NULL,NULL,IL           INCREMENT LOC                      83201330
035  2B9A 1080         A      MAR,YX,MDR             CALCULATE EFFECTIVE ADDRESS        83201340
036  2B5F 1E00         L      LOC,MAR                LOAD NEW LOC                       83201350
037  2BFF 1F9C  BC2    L      NULL,NULL,IR,D         FETCH NEXT INSTRUCTION             83201360
                *                                                           * 1C *      83201370
038  2B3B EC19  MR     M      YD,YDP1,YS,ILIR,D                                         83201380
039  0000 FFFE  CFFFE  DC     '0000FFFE'                                                83201390
                *                                                           * 1D *      83201400
03A  2A1F 1C89  DR     L      MR0,YD,ILIR            SAVE MS DIVIDEND                   83201410
03B  223B 1F8B         AX     MR1,YDP1,NULL,DR1      SAVE LS DIVIDEND (P.2)             83201420
                *                                                                       83201430
03C  17E4 2AC9  C2     BALNL  CL1(NULL),ILIR         SIGNS ALIKE (P.7)                  83201440
03D  3219 C001         SRAI   MR0,YD,1               PROPOGATE 1ST OP SIGN              83201450
03E  3210 7001         OI     MR0,MR0,1              FORCE SOME MAGNITUDE               83201460
03F  2A10 1830         A      MR0,MR0,MR0,E,D        SET CONDITION CODE                 83201470
                *                                                                       83201480
```

```
                   *                                                                        83201500
                   *                                                          * 20 *        83201510
040  17EC 1059  BTBS    BALT  BBS(NULL),ILIR,D        BRANCH IF MASK TRUE                   83201520
041  221F 0E83  BBS     SX    MR0,NULL,YSI,BBS1       DECREMENT BY TWICE R2                 83201530
                   *                                                          * 21 *        83201540
042  17EC 1159  BTFS    BALT  BFS(NULL),ILIR,D        BRANCH IF MASK TRUE                   83201550
043  2210 0E8D  BBS1    SX    MR0,MR0,YSI,DOSBR       MR0=BYTE DISPLACEMENT                 83201560
                   *                                                          * 22 *        83201570
044  13EC 1059  BFBS    BALF  BBS(NULL),ILIR,D        BRANCH IF MASK FALSE                  83201580
045  221F 1E87  BFS     AX    MR0,NULL,YSI,BFS1       INCREMENT BY TWICE R2                 83201590
                   *                                                          * 23 *        83201600
046  13EC 1159  BFFS    BALF  BFS(NULL),ILIR,D        BRANCH IF MASK FALSE                  83201610
047  2210 1E8D  BFS1    AX    MR0,MR0,YSI,DOSBR       MR0=BYTE DISPLACEMENT                 83201620
                   *                                                          * 24 *        83201630
048  2B3F 1EB9  LIS     L     YD,YSI,ILIR,E,D                                               83201640
049  221F 1E2F  CADRS2  LX    MR0,MAR,CADRS3          EFFECTIVE ADRS TO MR0 (P.5)           83201650
                   *                                                          * 25 *        83201660
04A  2B3F 0E89  LCS     S     YD,NULL,YSI,ILIR        SUBTRACT TO TWO'S COMP                83201670
04B  2B3F 1CB0          L     YD,YD,E,D               SET G,L                               83201680
                   *                                                          * 26 *        83201690
04C  2B39 1EB9  AIS     A     YD,YD,YSI,ILIR,E,D                                            83201700
04D  2350 1D0F  DOSBR   AX    LOC,MR0,LOC,DOSBR1      FORM BRANCH ADDRESS                   83201710
                   *                                                          * 27 *        83201720
04E  2B39 0EB9  SIS     S     YD,YD,YSI,ILIR,E,D                                            83201730
04F  2BFF 1F9C  DOSBR1  L     NULL,NULL,IR,D          FETCH NEXT INSTR                      83201740
                   *                                                          * 28 *        83201750
050  63F8 CC22  LER     CEQX  NULL,YS,YS,LER1         SET UP FALU (P.5)                     83201760
051  0000 0000          DC    0,0                                                           83201770
                   *                                                          * 29 *        83201780
052  6BF9 5C39  CER     CL    NULL,YD,YS,ILIR,E,D                                           83201790
053  0000 2800  B11820  LC    '00002800'                                                    83201800
                   *                                                          * 2A *        83201810
054  6219 CC64  AER     CEQX  MR0,YD,YS,AER1,C        SMALLER TO MR0(P.5)                   83201820
055  13F8 1980          BAL   AER2(NULL)              (P.5)                                 83201830
                   *                                                          * 2B *        83201840
056  6219 CC6A  SER     CEQX  MR0,YD,YS,SER1,C        SMALLER TO MR0 (P.5)                  83201850
057  13F8 0780          BAL   SER2(NULL)              (P.2)                                 83201860
                   *                                                          * 2C *        83201870
058  6B39 EC29  MER     ME    YD,YD,YS,ILIR,E         DO MULTIPLY,SET CC                    83201880
059  13F4 B210          BALV  FFAULT(NULL),D          (P.28)                                83201890
                   *                                                          * 2D *        83201900
05A  6A18 1FA9  DER     AU    MR0,YS,NULL,ILIR,E      TEST DIVISOR                          83201910
05B  13F8 06C0          BAL   DER1(NULL)              (P.2)                                 83201920
                   *                                                          * 2E *        83201930
05C  6A18 1F80  FXR     AU    MR0,YS,NULL             ARGUMENT TO MR0                       83201940
05D  13F8 E1C0          BAL   FXR1(NULL)              (P.39)                                83201950
                   *                                                          * 2F *        83201960
05E  2A1F 1C00  FLR     L     MR0,YS                  ARGUMENT TO MR0                       83201970
05F  13FC E640          BALA  FLR1(NULL)              ARM INTERRUPTS (P.40)                 83201980
```

```
                *                                                                                  83202000
                *                                                                                  83202010
060  0000 0000         DC    0,0                                                                   83202020
061  0000 0000         DC    0,0                                                                   83202030
                *                                                                                  83202040
062  6E38 8FA9  LER1   SE    YD,YS,NULL,ILIR,E        SUBTRACI TO NORMALIZE                        83202050
063  13F4 B210         BALV  FFAULT(NULL),D           (P.28)                                       83202060
                *                                                                                  83202070
064  6B39 9829  AER1   AE    YD,YD,MR0,ILIR,E         DO A PLUS B                                  83202080
065  13F4 B210         BALV  FFAULT(NULL),D           (P.28)                                       83202090
                *                                                                                  83202100
066  6B38 9829  AER2   AE    YD,YS,MR0,ILIR,E         DO B PLUS A                                  83202110
067  13F4 B210         BALV  FFAULT(NULL),D           (P.28)                                       83202120
                *                                                                     * 34 *       83202130
068  3338 A010  EXHR   RRI   YD,YS,16                 EXCHANGE HALFWORDS                           83202140
069  2BFF 1C99         L     NULL,YD,ILIR,D                                                        83202150
                *                                                                                  83202160
06A  6B39 8829  SER1   SE    YD,YD,MR0,ILIR,E         DO A MINUS B                                 83202170
06B  13F4 B210         BALV  FFAULT(NULL),D           (P.28)                                       83202180
                *                                                                                  83202190
                * CALCULATE EFFECTIVE RX ADDRESS                                                   83202200
                *                                                                                  83202210
06C  239A 1DC9  CADRS  AX    MAR,YX,MDR,CADRS2,C      TRANSFER IF RX1 OR RX3 (P.4)                 83202220
06D  2A1F 1E00         L     MR0,MAR                  D2+(X2)+(LOC) TO MR0                         83202230
                *                                                                                  83202240
06E  3210 1004  CADRS1 AI    MR0,MR0,4                ADD 4 FOR LOC INCREMENT                      83202250
06F  03F8 0800  CADRS3 BAL   (MR6)(NULL)              RETURN TO CALL                               83202260
                *                                                                                  83202270
                *                                                                     * 38 *       83202280
070  0000 0000  LDR    DC    0,0                                                                   83202290
071  0000 0000         DC    0,0                                                                   83202300
                *                                                                     * 39 *       83202310
072  0000 0000  CDR    DC    0,0                                                                   83202320
073  0000 0000         DC    0,0                                                                   83202330
                *                                                                     * 3A *       83202340
074  0000 0000  ADR    DC    0,0                                                                   83202350
075  0000 0000         DC    0,0                                                                   83202360
                *                                                                     * 3B *       83202370
076  0000 0000  SDR    DC    0,0                                                                   83202380
077  0000 0000         DC    0,0                                                                   83202390
                *                                                                     * 3C *       83202400
078  0000 0000  MDR    DC    0,0                                                                   83202410
079  0000 0000         DC    0,0                                                                   83202420
                *                                                                     * 3D *       83202430
07A  0000 0000  DDR    DC    0,0                                                                   83202440
07B  0000 0000         DC    0,0                                                                   83202450
                *                                                                     * 3E *       83202460
07C  0000 0000  FXDR   DC    0,0                                                                   83202470
07D  0000 0000         DC    0,0                                                                   83202480
                *                                                                     * 3F *       83202490
07E  0000 0000  FLDR   DC    0,0                                                                   83202500
07F  0000 0000         DC    0,0                                                                   83202510
```

```
                 *                                                                         83202530
                 *                                                               * 40 *    83202540
080   12D8 5E80  STH     BAL    STORE(MR6)          COMMON ROUTINE (P.14)                  83202550
081   2B7F 1C83          L      MDR,YD,DW2          EXECUTED INSTR.                        83202560
                 *                                                               * 41 *    83202570
082   2BFF 1F81  BAL     L      NULL,NULL,IL        INCREMENT LOC                          83202580
083   235A 1DA5          AX     MAR,YX,MDR,BAL1     CALCULATE BRANCH ADDRESS (P.7)         83202590
                 *                                                               * 42 *    83202600
084   17EC 0D19  BTC     BALT   BC1(NULL),ILIR,D    BRANCH IF MASK TRUE (P.3)              83202610
085   2BFF 1F9C  BAL2    L      NULL,NULL,IR,D                                             83202620
                 *                                                               * 43 *    83202630
086   13EC 0D19  BFC     BALF   BC1(NULL),ILIR,D    BRANCH IF MASK FALSE (P.3)             83202640
087   0000 0000          DC     0,0                                                        83202650
                 *                                                               * 44 *    83202660
088   2B9A 1D8B  NH      A      MAR,YX,MDR,DR2                                             83202670
089   2B39 5DB9          N      YD,YD,MDR,ILIR,E,D                                         83202680
                 *                                                               * 45 *    83202690
08A   2B9A 1D8B  CLH     A      MAR,YX,MDR,DR2                                             83202700
08B   2BF9 0DB9          S      NULL,YD,MDR,ILIR,E,D                                       83202710
                 *                                                               * 46 *    83202720
08C   2B9A 1D8B  OH      A      MAR,YX,MDR,DR2                                             83202730
08D   2B39 7DB9          O      YD,YD,MDR,ILIR,E,D                                         83202740
                 *                                                               * 47 *    83202750
08E   2B9A 1D8B  XH      A      MAR,YX,MDR,DR2                                             83202760
08F   2B39 6DB9          X      YD,YD,MDR,ILIR,E,D                                         83202770
                 *                                                               * 48 *    83202780
090   2B9A 1D8B  LH      A      MAR,YX,MDR,DR2                                             83202790
091   2B3F 1DB9          L      YD,MDR,ILIR,E,D                                            83202800
                 *                                                               * 49 *    83202810
092   2B9A 1D8B  CH      A      MAR,YX,MDR,DR2                                             83202820
093   13F8 04C0          BAL    C1(NULL)            (P.2)                                  83202830
                 *                                                               * 4A *    83202840
094   2B9A 1D8B  AH      A      MAR,YX,MDR,DR2                                             83202850
095   2B39 1DB9          A      YD,YD,MDR,ILIR,E,D                                         83202860
                 *                                                               * 4B *    83202870
096   2B9A 1D8B  SH      A      MAR,YX,MDR,DR2                                             83202880
097   2B39 0DB9          S      YD,YD,MDR,ILIR,E,D                                         83202890
                 *                                                               * 4C *    83202900
098   2B9A 1D8B  MH      A      MAR,YX,MDR,DR2      FETCH MULTIPLIER                       83202910
099   13F8 D780          BAL    MH1(NULL)           (P.37)                                 83202920
                 *                                                               * 4D *    83202930
09A   2B9A 1D8B  DH      A      MAR,YX,MDR,DR2      FETCH DIVISOR                          83202940
09B   13F8 D980          BAL    DH1(NULL)           (P.37)                                 83202950
                 *                                                                         83202960
09C   2A3B 1F80  D0      A      MR1,YDP1,NULL                                              83202970
09D   2B3B FD89          D      YD,YDP1,MDR,ILIR                                           83202980
09E   13F4 B110          BALV   OFALT0(NULL),D      (P.28)                                 83202990
09F   0000 0000          DC     0,0                                                        83203000
```

```
                     *
                     *                                                              * 50 *
0A0   12D8 5E80  ST       BAL    STORE(MR6)                COMMON ROUTINE (P.14)
0A1   2B7F 1C87           L      MDR,YD,DW4                EXECUTED INSTR.
                     *                                                              * 51 *
0A2   12D8 80C0  AM       BAL    RDFULL(MR6)               FETCH FULLWORD (P.31)
0A3   2B79 10A7           A      MDR,YD,MDR,DW4,E
0A4   2BFF 1F99           L      NULL,NULL,ILIR,D
                     *
0A5   2A1F 1E00  BAL1     L      MR0,MAR                   BRANCH ADRS TO MR0
0A6   2B3F 1D00           L      YD,LOC                    INCREMENTED LOC TO R1
0A7   235F 1805           LX     LOC,MR0,BAL2              LOAD LOC (P.6)
                     *
                     *                                                              * 54 *
0A8   2B9A 1D8F  N        A      MAR,YX,MDR,DR4
0A9   2B39 5DB9           N      YD,YD,MDR,ILIR,E,D
                     *                                                              * 55 *
0AA   2B9A 1D8F  CL       A      MAR,YX,MDR,DR4
0AB   2BF9 0DB9  CL1      S      NULL,YD,MDR,ILIR,E,D
                     *                                                              * 56 *
0AC   2B9A 1D8F  O        A      MAR,YX,MDR,DR4
0AD   2B39 7DB9           O      YD,YD,MDR,ILIR,E,D
                     *                                                              * 57 *
0AE   2B9A 1D8F  X        A      MAR,YX,MDR,DR4
0AF   2B39 6DB9           X      YD,YD,MDR,ILIR,E,D
                     *                                                              * 58 *
0B0   2B9A 1D8F  L        A      MAR,YX,MDR,DR4
0B1   2B3F 1DB9           L      YD,MDR,ILIR,E,D
                     *                                                              * 59 *
0B2   2B9A 1D8F  C        A      MAR,YX,MDR,DR4
0B3   13F8 04C0           BAL    C1(NULL)                  (P.2)
                     *                                                              * 5A *
0B4   2B9A 1D8F  A        A      MAR,YX,MDR,DR4
0B5   2B39 1DB9           A      YD,YD,MDR,ILIR,E,D
                     *                                                              * 5B *
0B6   2B9A 1D8F  S        A      MAR,YX,MDR,DR4
0B7   2B39 0DB9           S      YD,YD,MDR,ILIR,E,D
                     *                                                              * 5C *
0B8   2B9A 1D8F  M        A      MAR,YX,MDR,DR4
0B9   2B3B ED99           M      YD,YDP1,MDR,ILIR,D
                     *                                                              * 5D *
0BA   2B9A 1D8F  D        A      MAR,YX,MDR,DR4
0BB   221F 1C9C           LX     MR0,YD,D0                 (P.6)
                     *                                                              * 5E *
0BC   12D8 BC40  CRC12    BAL    RDHALF(MR6)               FETCH CHECKWORD (P.31)
0BD   13FC F2C0           BALA   CRC12A(NULL)              ARM INTERRUPTS (P.42)
                     *                                                              * 5F *
0BE   12D8 BC40  CRC16    BAL    RDHALF(MR6)               FETCH CHECKWORD (P.31)
0BF   13FC F480           BALA   CRC16A(NULL)              ARM INTERRUPTS (P.42)
```

```
                    *                                                                  83203520
                    *                                                    * 60 *        83203530
0C0   12D8 5E80   STE      BAL    STORE(MR6)            COMMON ROUTINE (P.14)          83203540
0C1   6B79 1F87            AU     MDR,YD,NULL,DW4       EXECUTED INSTRUCTION           83203550
                    *                                                    * 61 *        83203560
0C2   12D8 BC40   AHM      BAL    RDHALF(MR6)           FETCH HALFWORD (P.31)          83203570
0C3   2B79 1DA3            A      MDR,YD,MDR,DW2,E                                     83203580
0C4   2BFF 1F99            L      NULL,NULL,ILIR,D                                     83203590
                    *                                                                  83203600
0C5   2BFF 1F83   CBT2     L      NULL,NULL,DW2                                        83203610
0C6   2BFF 1F99            L      NULL,NULL,ILIR,D                                     83203620
0C7   0000 0000            DC     0,0                                                  83203630
                    *                                                                  83203640
                    *                                                    * 64 *        83203650
0C8   12D8 C000   ATL      BAL    ATBL(MR6)             COMMON OVERHEAD (P.32)         83203660
0C9   13F8 C400            BAL    ATL1(NULL)            (P.33)                         83203670
                    *                                                    * 65 *        83203680
0CA   12D8 C000   ABL      BAL    ATBL(MR6)             COMMON OVERHEAD (P.32)         83203690
0CB   13F8 C700            BAL    ABL1(NULL)            (P.33)                         83203700
                    *                                                    * 66 *        83203710
0CC   12D8 CDC0   RTL      BAL    RTBL(MR6)             COMMON OVERHEAD (P.35)         83203720
0CD   13F8 D1C0            BAL    RTL1(NULL)            (P.36)                         83203730
                    *                                                    * 67 *        83203740
0CE   12D8 CDC0   RBL      BAL    RTBL(MR6)             COMMON OVERHEAD (P.35)         83203750
0CF   13F8 D440            BAL    RBL1(NULL)            (P.36)                         83203760
                    *                                                    * 68 *        83203770
0D0   2B9A 1D8F   LE       A      MAR,YX,MDR,DR4                                       83203780
0D1   13F8 4500            BAL    LE1(NULL)             (P.10)                         83203790
                    *                                                    * 69 *        83203800
0D2   2B9A 1D8F   CE       A      MAR,YX,MDR,DR4                                       83203810
0D3   6BF9 50B9            CE     NULL,YD,MDR,ILIR,E,D                                 83203820
                    *                                                    * 6A *        83203830
0D4   2B9A 1D8F   AE       A      MAR,YX,MDR,DR4                                       83203840
0D5   13F8 4600            BAL    AE1(NULL)             (P.10)                         83203850
                    *                                                    * 6B *        83203860
0D6   2B9A 1D8F   SE       A      MAR,YX,MDR,DR4                                       83203870
0D7   13F8 4700            BAL    SE1(NULL)             (P.10)                         83203880
                    *                                                    * 6C *        83203890
0D8   2B9A 1D8F   ME       A      MAR,YX,MDR,DR4                                       83203900
0D9   13F8 59C0            BAL    ME1(NULL)             (P.13)                         83203910
                    *                                                    * 6D *        83203920
0DA   2B9A 1D8F   DE       A      MAR,YX,MDR,DR4                                       83203930
0DB   2A1F 1DA9            L      MR0,MDR,ILIR,E        TEST DIVISOR                   83203940
0DC   13E0 0700            BALZ   DEZ(NULL)             FAULT IF ZERO (P.2)            83203950
0DD   6B39 F820   DER2     DE     YD,YD,MR0,E           DO DIVIDE, SET CC              83203960
0DE   13F4 B210            BALV   FFAULT(NULL),D        (P.28)                         83203970
0DF   0000 0000            DC     0,0                                                  83203980
                    *                                                                  83203990
                    *                                                                 83204010
                    *                                                     * 70 *      83204020
0E0  0000 0000  STD     DC     0,0                                                    83204030
0E1  0000 0000          DC     0,0                                                    83204040
                    *                                                     * 71 *      83204050
0E2  12B8 1800  STME    BAL    CADRS(MR6)            CALCULATE ADDRESS (P.5)          83204060
0E3  13F8 41C0          BAL    STME1(NULL)           (P.10)                           83204070
                    *                                                     * 72 *      83204080
0E4  12D8 BDC0  LME     BAL    RDFULL(MR6)           READ 1ST FULLWORD (P.31)         83204090
0E5  13F8 4340          BAL    LME1(NULL)            (P.10)                           83204100
                    *                                                     * 73 *      83204110
0E6  2B9A 1D8B  LHL     A      MAR,YX,MDR,DR2                                         83204120
0E7  13F8 BF40          BAL    LHL1(NULL)            (P.31)                           83204130
                    *                                                     * 74 *      83204140
0E8  12B8 B940  TBT     BAL    COMBIT(MR5)           (P.30)                           83204150
0E9  2BFF 1F99          L      NULL,NULL,ILIR,D                                       83204160
                    *                                                     * 75 *      83204170
0EA  12B8 B940  SBT     BAL    COMBIT(MR5)           (P.30)                           83204180
0EB  2372 7D85          OX     MDR,MR2,MDR,CBT2      (P.8)                            83204190
                    *                                                     * 76 *      83204200
0EC  12B8 B940  RBT     BAL    COMBIT(MR5)           (P.30)                           83204210
0ED  2372 7DAF          OX     MDR,MR2,MDR,CBT1                                       83204220
                    *                                                     * 77 *      83204230
0EE  12B8 B940  CBT     BAL    COMBIT(MR5)           (P.30)                           83204240
0EF  2372 6D85  CBT1    XX     MDR,MR2,MDR,CBT2      (P.8)                            83204250
                    *                                                     * 78 *      83204260
0F0  0000 0000  LD      DC     0,0                                                    83204270
0F1  0000 0000          DC     0,0                                                    83204280
                    *                                                     * 79 *      83204290
0F2  0000 0000  CD      DC     0,0                                                    83204300
0F3  0000 0000          DC     0,0                                                    83204310
                    *                                                     * 7A *      83204320
0F4  0000 0000  AD      DC     0,0                                                    83204330
0F5  0000 0000          DC     0,0                                                    83204340
                    *                                                     * 7B *      83204350
0F6  0000 0000  SD      DC     0,0                                                    83204360
0F7  0000 0000          DC     0,0                                                    83204370
                    *                                                     * 7C *      83204380
0F8  0000 0000  MD      DC     0,0                                                    83204390
0F9  0000 0000          DC     0,0                                                    83204400
                    *                                                     * 7D *      83204410
0FA  0000 0000  DD      DC     0,0                                                    83204420
0FB  0000 0000          DC     0,0                                                    83204430
                    *                                                     * 7E *      83204440
0FC  0000 0000  STMD    DC     0,0                                                    83204450
0FD  0000 0000          DC     0,0                                                    83204460
                    *                                                     * 7F *      83204470
0FE  0000 0000  LMD     DC     0,0                                                    83204480
0FF  0000 0000          DC     0,0                                                    83204490
                    *                                                                 83204500
```

```
                          ORG    '100'                                                     83204510
                   *                                                                       83204520
                   *                                                                       83204530
100  0000 0000            DC     0,0                                                       83204540
                   *                                                                       83204550
101  13F0 44CD     COMLML BALC   EXLSTM(NULL),I4DR4     EXIT IF DONE, ELSE READ NEXT       83204560
102  0BF8 0800     COMLM  EXL    (MR6)(NULL)            EXECUTE LOAD                       83204570
103  23D1 1F01            AX     YDI,MR1,YD1,COMLML     BUMP R1 FIELD & LOOP               83204580
                   *                                                                       83204590
                   *                                                                       83204600
104  323F 000F     STM1   SI     MR1,NULL,15            MR1='FFFFFFF1'                     83204610
105  12D8 4400            BAL    COMSTM(MR6)            TO COMMON ROUTINE                  83204620
106  2B7F 1C85            L      MDR,YD,I4DW4           EXECUTED INSTRUCTION               83204630
                   *                                                                       83204640
107  323F 000E     STME1  SI     MR1,NULL,14            MR1='FFFFFFF2'                     83204650
108  12D8 4400            BAL    COMSTM(MR6)            TO COMMON ROUTINE                  83204660
109  6B79 1F85            AU     MDR,YD,NULL,I4DW4      EXECUTED INSTRUCTION               83204670
                   *                                                                       83204680
10A  323F 000F     LM1    SI     MR1,NULL,15            MR1='FFFFFFF1'                     83204690
10B  12D8 4080            BAL    COMLM(MR6)             TO COMMON ROUTINE Br'd to 122 instead 83204700  B = FFFF  F80F
10C  2B3F 1D80            L      YD,MDR                 EXECUTED INSTRUCTION of 101        83204710
                   *                                                                       83204720
10D  323F 000E     LME1   SI     MR1,NULL,14            MR1='FFFFFFF2'                     83204730
10E  12D8 4080            BAL    COMLM(MR6)             TO COMMON ROUTINE                  83204740
10F  6B3F 1D80            AU     YD,NULL,MDR            EXECUTED INSTRUCTION               83204750
                   *                                                                       83204760
                   *                                                                       83204770
                   *                                                                       83204780
110  3390 0004     COMSTM SI     MAR,MR0,4              TEMPORARY DECREMENT                83204790
111  0BF8 0800     CMSTML EXL    (MR6)(NULL)            EXECUTE STORE                      83204800
112  23D1 1F51            AX     YDI,MR1,YD1,CMSTML,C   BUMP R1 FIELD & LOOP               83204810
113  2BFF 1F99     EXLSTM L      NULL,NULL,ILIR,D       FETCH NEXT INSTRUCTION             83204820
                   *                                                                       83204830
                   *                                                                       83204840
                   *                                                                       83204850
114  2A1F 1D80     LE1    L      MR0,MDR                ARGUMENT TO MR0                    83204860
115  6BF0 C800     LE2    CEQ    NULL,MR0,MR0           SET UP FALU                        83204870
116  6B30 8FA9            SE     YD,MR0,NULL,ILIR,E     SUBTRACT TO NORMALIZE              83204880
117  13F4 B210            BALV   FFAULT(NULL),D         (P.28)                             83204890
                   *                                                                       83204900
                   *                                                                       83204910
118  2A3F 1D80     AE1    L      MR1,MDR                                                   83204920
119  6219 C8F7            CEQX   MR0,YD,MR1,AE2,C       SMALLER OPERAND TO MR0 (P.11)      83204930
11A  6B31 9829            AE     YD,MR1,MR0,ILIR,E      DO B PLUS A                        83204940
11B  13F4 B210            BALV   FFAULT(NULL),D         (P.28)                             83204950
                   *                                                                       83204960
                   *                                                                       83204970
11C  2A3F 1D80     SE1    L      MR1,MDR                2ND OPERAND TO MR1                 83204980
11D  6219 C8FD            CEQX   MR0,YD,MR1,SE2,C       SMALLER TO MR0 (P.11)              83204990
11E  6B31 8829            SE     YD,MR1,MR0,ILIR,E      DO B MINUS A                       83205000
11F  13F4 B210            BALV   FFAULT(NULL),D         (P.28)                             83205010
```

```
                 *                                                            83205030
                 *                                                   * 90 *   83205040
120  2B39 8EF9   SRHLS   SRHL  YD,YD,YSI,ILIR,E,D                             83205050
121  0000 0F01   C0F01   DC    '00000F01'                                     83205060
                 *                                                   * 91 *   83205070
122  2B39 9FF9   SLHLS   SLHL  YD,YD,YSI,ILIR,E,D                             83205080
123  0000 A001   CA001   DC    '0000A001'                                     83205090
                 *                                                   * 92 *   83205100
124  3618 5015   STBR    NI    MR0,YS,H1HALF,I      SAVE MS 16 BITS           83205110
125  13F8 5A40           BAL   STBR1(NULL)          (P.13)                    83205120
                 *                                                   * 93 *   83205130
126  4B3F 5C59   LBR     LBR   YD,YS,ILIR,D                                   83205140
127  13F8 A000   EPSR1   BAL   TEST1(NULL)          (P.27)                    83205150
                 *                                                   * 94 *   83205160
128  3619 5015   EXBR    NI    MR0,YD,H1HALF,I      SAVE MS 16 BITS           83205170
129  13F8 5B40           BAL   EXBR1(NULL)          (P.13)                    83205180
                 *                                                   * 95 *   83205190
12A  2B3D 1F81   EPSR    A     YD,PSW,NULL,IL       PSW TO R1, INC. LOC       83205200
12B  227F 1C27           LX    MR3,YS,EPSR1         NEW PSW TO MR3            83205210
                 *                                                   * 96 *   83205220
12C  12D8 4FC0   WBR     BAL   RWBRR(MR6)           (P.12)                    83205230
12D  4BFF 1D80           WD    NULL,MDR                                       83205240
                 *                                                   * 97 *   83205250
12E  12D8 4FC0   RBR     BAL   RWBRR(MR6)           (P.12)                    83205260
12F  467F 0D83           RD    MDR,MDR,DW2                                    83205270
                 *                                                   * 98 *   83205280
130  4BF9 DC39   WHR     WHA   NULL,YD,YS,ILIR,E,D                            83205290
131  F000 0000   DIGIT1  DC    'F0000000'                                     83205300
                 *                                                   * 99 *   83205310
132  4B19 CFB9   RHR     RHA   YS,YD,NULL,ILIR,E,D                            83205320
133  0F00 0000   DIGIT2  DC    '0F000000'                                     83205330
                 *                                                   * 9A *   83205340
134  4BF9 9C79   WDR     WDRA  NULL,YD,YS,ILIR,E,D                            83205350
135  0000 0000           DC    0,0                                            83205360
                 *                                                   * 9B *   83205370
136  4B19 8FF9   RDR     RDRA  YS,YD,NULL,ILIR,E,D                            83205380
                 *                                                            83205390
137  6B39 9829   AE2     AE    YD,YD,MR0,ILIR,E     DO A PLUS B               83205400
138  13F4 B210           BALV  FFAULT(NULL),D       (P.28)                    83205410
139  0000 0000           DC    0,0                                            83205420
                 *                                                   * 9D *   83205430
13A  4B19 AFF9   SSR     SSRA  YS,YD,NULL,ILIR,E,D                            83205440
13B  7FFF FFFF   ONES    DC    '7FFFFFFF'                                     83205450
                 *                                                   * 9E *   83205460
13C  4BF9 BC79   OCR     OCRA  NULL,YD,YS,ILIR,E,D                            83205470
                 *                                                            83205480
13D  6B39 8829   SE2     SE    YD,YD,MR0,ILIR,E     DO A MINUS B              83205490
13E  13F4 B210           BALV  FFAULT(NULL),D       (P.28)                    83205500
```

```
                     *                                                                83205520
                     * COMMON RBR,WBR                                                 83205530
                     *                                                                83205540
13F   2A5F 1C00   RWBRR  L      MR2,YS                MR2=START ADRS                  83205550
140   4B79 AFC0          SSRA   MDR,YD,NULL           ADDRESS THE DEVICE              83205560
141   2BDF 3E80          AINC   YDI,NULL,YSI          POINT TO R2+1                   83205570
142   227F 1C8A          LX     MR3,YD,CMNBRW         MR3=ENDING ADRS                 83205560
                     *                                                                83205590
                     *                                                                83205600
143   12D8 5180   WB1    BAL    RWBRX(MR6)            COMMON SET-UP ROUTINE           83205610
144   4BFF 1080          WD     NULL,MDR              EXECUTED INSTRUCTION            83205620
                     *                                                                83205630
                     *                                                                83205640
145   32DF 1166   RB1    LI     MR6,RB2                                               83205650
                     *                                                                83205660
                     *                                                                83205670
                     * COMMON RB,WB                                                   83205680
                     *                                                                83205690
146   2B7F 1D80   RWBRX  L      MDR,MDR                                               83205700
147   2A5F 1D8D          L      MR2,MDR,I4DR4         MR2=START ADDRESS               83205710
148   4BF9 AFC0          SSRA   NULL,YD,NULL          ADDRESS THE DEVICE              83205720
149   2A7F 1D80          L      MR3,MDR               MR3=ENDING ADDRESS              83205730
                     *                                                                83205740
14A   33DF 1007   CMNBRW LI     YDI,7                 CC MASK                         83205750
14B   23F3 094E          SX     NULL,MR3,MR2,BRW2,C   COMPARE START:END               83205760
14C   33BD 5FF0   TS3    NI     PSW,PSW,'FF0'         CLEAR CONDITION CODE            83205770
14D   2BFF 1F99          L      NULL,NULL,ILIR,D      AND EXIT                        83205780
                     *                                                                83205790
14E   2B9F 190B   BRW2   L      MAR,MR2,DR2           FETCH HALFWORD                  83205800
14F   4BFF 2FE0   BRW3   SSR    NULL,NULL,E           SENSE DEVICE STATUS             83205810
150   17EC 5540          BALT   ENDBRW(NULL)          EXIT IF BAD STATUS              83205820
151   13F0 53C0          BALC   BRW3(NULL)            LOOP ON BUSY                    83205830
152   0BF8 0B00          EXL    (MR6)(NULL)           DO INSTR AT (MR6)               83205840
153   3252 1001   BRW4   AI     MR2,MR2,1             INCREMENT ADDRESS               83205850
154   23F3 094E          SX     NULL,MR3,MR2,BRW2,C   COMPARE TO FINAL & LOOP         83205860
                     *                                                                83205870
155   2BFF 1F99   ENDBRW L      NULL,NULL,ILIR,D      EXIT                            83205880
```

```
                   *                                                                  83205900
                   * AUTO-LOAD                                                        83205910
                   *                                                                  83205920
156  339F 1078  AL1      LI     MAR,X'78'                                             83205930
                   *                              KILL ADRS CALCULATION LOGIC         83205940
157  2E7F 1F8A           L      MDR,NULL,PR2      AND FETCH BINDV SPECIFICATION       83205950
158  325F 1080           LI     MR2,'80'          MR2=START ADDRESS                   83205960
159  2A7F 1800           L      MR3,MR0           MR3=FINAL ADDRESS                   83205970
15A  2A3F 1D80           L      MR1,MDR                                               83205980
15B  4A31 DFC0           LB     MR1,MR1,NULL      MR1=DEVICE NUMBER                   83205990
15C  4BF1 BDC0           OCRA   NULL,MR1,MDR                                          83206000
15D  2B9F 190B           L      MAR,MR2,DR2       ADRS & READ 1ST HW                  83206010
15E  33DF 1007           LI     YDI,7             SET CC MASK                         83206020
                   *                                                                  83206030
15F  4BFF 2FE0  AL2      SSR    NULL,NULL,E                                           83206040
160  17EC 5540           BALT   ENDBRW(NULL)      EXIT IF BAD STATUS (P.12)           83206050
161  13F0 57C0           BALC   AL2(NULL)         LOOP ON BUSY                        83206060
162  4A1F 0FC0           RDR    MR0,NULL          INPUT DATA BYTE                     83206070
163  23FF 085F           SX     NULL,NULL,MR0,AL2,C    LOOP IF BLANK                  83206080
164  4B70 CDC3           STB    MDR,MR0,MDR,DW2   STORE FIRST NON-ZERO BYTE AND       83206090
165  12D8 54C0           BAL    BRW4(MR6)         CONTINUE INPUT (P.12)               83206100
166  4B7F 0D83  RB2      RD     MDR,MDR,DW2       EXECUTED INSTRUCTION                83206110
                   *                                                                  83206120
                   *                                                                  83206130
167  6B39 EDA9  ME1      ME     YD,YD,MDR,ILIR,E  DO MPY, START INSTR FETCH           83206140
168  13F4 B210           BALV   FFAULT(NULL),D    (P.28)                              83206150
                   *                                                                  83206160
                   *                                                                  83206170
169  4B19 4C49  STBR1    STBR   YS,YD,YS,ILIR     COPY R1 24:31 TO R2 24:31           83206180
16A  2B18 7810           O      YS,YS,MR0,D       RESTORE R2 0:15                     83206190
                   *                                                                  83206200
                   *                                                                  83206210
16B  4B79 CDC3  STB1     STB    MDR,YD,MDR,DW2    COPY R1 24:31 TO MEMORY             83206220
16C  2BFF 1F99           L      NULL,NULL,ILIR,D                                      83206230
                   *                                                                  83206240
                   *                                                                  83206250
16D  4B3F EC49  EXBR1    EXB    YD,YS,ILIR        SWAP BYTES                          83206260
16E  2B39 7810           O      YD,YD,MR0,D       RESTORE R1 0:15                     83206270
                   *                                                                  83206280
                   *                                                                  83206290
16F  2A1F 1D80  LB1      L      MR0,MDR           COPY HW TO MR0                      83206300
170  4B30 DDC0           LB     YD,MR0,MDR        MS OR LS BYTE TO R1                 83206310
171  2BFF 1F99           L      NULL,NULL,ILIR,D                                      83206320
                   *                                                                  83206330
                   *                                                                  83206340
172  3239 50FF  CLB1     NI     MR1,YD,'FF'       ISOLATE 1ST OP BYTE                 83206350
173  2A1F 1D80           L      MR0,MDR           2ND OP BYTE IS IN MDR               83206360
174  4A10 DDC0           LB     MR0,MR0,MDR       GET IT                              83206370
175  2BF1 0839           S      NULL,MR1,MR0,ILIR,E,D SUBTRACT TO COMPARE             83206380
                   *                                                                  83206390
                   *                                                                  83206400
176  4BF9 9DA0  WD1      WDA    NULL,YD,MDR,E     ADRS DEV & OUTPUT BYTE              83206410
177  2BFF 1F99           L      NULL,NULL,ILIR,D                                      83206420
                   *                                                                  83206430
                   *                                                                  83206440
178  4B79 ADA3  SS1      SSA    MDR,YD,MDR,DW2,E  ADRS DEV & INPUT STATUS             83206450
179  2BFF 1F99           L      NULL,NULL,ILIR,D                                      83206460
```

```
                * * * * * * * * * * * * * * * * * * * * * * * * *                       83206480
                *                                               *                       83206490
                * COMMON SUBROUTINE FOR STH,ST,STE,RH            *                       83206500
                *                                               *                       83206510
                * * * * * * * * * * * * * * * * * * * * * * * * *                       83206520
                *                                                                       83206530
                *                                                                       83206540
                *                                                                       83206550
17A  239A 1DFD  STORE  AX    MAR,YX,MDR,STORE1,C   TRANSFER IF RX1 OR RX3               83206560
17B  321F 1004         LI    MR0,4                 ADD 4 IF RX2                         83206570
17C  2390 1E3E         AX    MAR,MR0,MAR,STORE2    UPDATE MAR                           83206580
17D  2B9F 1E00  STORE1 L     MAR,MAR               MAR=EFFECTIVE ADDRESS                83206590
17E  0BF8 0B00  STORE2 EXL   (MR6)(NULL)           EXECUTE INSTRUCTION                  83206600
17F  2BFF 1F99         L     NULL,NULL,ILIR,D      DO NEXT INSTR                        83206610
                *                                                                       83206620
```

```
                   *                                                                    83206640
                   *                                                         * C0 *     83206650
180  12D8 7800  BXH    BAL    BXLH(MR6)                  COMMON ROUTINE (P.18)            83206660
181  13F0 0D19         BALC   BC1(NULL),ILIR,D           (P.3)                            83206670
                   *                                                         * C1 *     83206680
182  12D8 7800  BXLE   BAL    BXLH(MR6)                  COMMON ROUTINE (P.18)            83206690
183  17F0 0D19         BALNC  BC1(NULL),ILIR,D           (P.3)                            83206700
                   *                                                         * C2 *     83206710
184  12D8 BDC0  LPSW   BAL    RDFULL(MR6)                FETCH FW PSW (P.31)              83206720
185  13F8 AC80         BAL    LPSW1(NULL)                (P.27)                           83206730
                   *                                                         * C3 *     83206740
186  2A1A 1D89  THI    A      MR0,YX,MDR,ILIR                                             83206750
187  2BF9 5830         N      NULL,YD,MR0,E,D                                             83206760
                   *                                                         * C4 *     83206770
188  2A1A 1D89  NHI    A      MR0,YX,MDR,ILIR                                             83206780
189  2B39 5830         N      YD,YD,MR0,E,D                                               83206790
                   *                                                         * C5 *     83206800
18A  2A1A 1D89  CLHI   A      MR0,YX,MDR,ILIR                                             83206810
18B  2BF9 0830         S      NULL,YD,MR0,E,D                                             83206820
                   *                                                         * C6 *     83206830
18C  2A1A 1D89  OHI    A      MR0,YX,MDR,ILIR                                             83206840
18D  2B39 7830         O      YD,YD,MR0,E,D                                               83206850
                   *                                                         * C7 *     83206860
18E  2A1A 1D89  XHI    A      MR0,YX,MDR,ILIR                                             83206870
18F  2B39 6830         X      YD,YD,MR0,E,D                                               83206880
                   *                                                         * C8 *     83206890
190  2A1A 1D89  LHI    A      MR0,YX,MDR,ILIR                                             83206900
191  2B3F 1830         L      YD,MR0,E,D                                                  83206910
                   *                                                         * C9 *     83206920
192  2B7A 1D80  CHI    A      MDR,YX,MDR                                                  83206930
193  13F8 04C0         BAL    C1(NULL)                   (P.2)                            83206940
                   *                                                         * CA *     83206950
194  2A1A 1D89  AHI    A      MR0,YX,MDR,ILIR                                             83206960
195  2B39 1830         A      YD,YD,MR0,E,D                                               83206970
                   *                                                         * CB *     83206980
196  2A1A 1D89  SHI    A      MR0,YX,MDR,ILIR                                             83206990
197  2B39 0830         S      YD,YD,MR0,E,D                                               83207000
                   *                                                         * CC *     83207010
198  2A1A 1D89  SRHL   A      MR0,YX,MDR,ILIR                                             83207020
199  2B39 8870         SRHL   YD,YD,MR0,E,D                                               83207030
                   *                                                         * CD *     83207040
19A  2A1A 1D89  SLHL   A      MR0,YX,MDR,ILIR                                             83207050
19B  2B39 9870         SLHL   YD,YD,MR0,E,D                                               83207060
                   *                                                         * CE *     83207070
19C  2A1A 1D89  SRHA   A      MR0,YX,MDR,ILIR                                             83207080
19D  2B39 C870         SRHA   YD,YD,MR0,E,D                                               83207090
                   *                                                         * CF *     83207100
19E  2A1A 1D89  SLHA   A      MR0,YX,MDR,ILIR                                             83207110
19F  2B39 D870         SLHA   YD,YD,MR0,E,D                                               83207120
```

```
                    *                                                                 83207140
                    *                                                      * D0 *     83207150
1A0  1208 1B00  STM      BAL      CADRS(MR6)                CALCULATE ADDRESS (P.5)   83207160
1A1  13F8 4100           BAL      STM1(NULL)                (P.10)                    83207170
                    *                                                      * D1 *     83207180
1A2  1208 BDC0  LM       BAL      RDFULL(MR6)               GET 1ST REGISTER (P.31)   83207190
1A3  13F8 4280           BAL      LM1(NULL)                 (P.10)                    83207200 B=1A3, S=2FD
                    *                                                      * D2 *     83207210
1A4  1208 BC40  STB      BAL      RDHALF(MR6)               FETCH HALFWORD (P.31)     83207220
1A5  13F8 5AC0           BAL      STB1(NULL)                (P.13)                    83207230
                    *                                                      * D3 *     83207240
1A6  1208 BC40  LB       BAL      RDHALF(MR6)               FETCH HALFWORD (P.31)     83207250
1A7  13F8 5BC0           BAL      LB1(NULL)                 (P.13)                    83207260
                    *                                                      * D4 *     83207270
1A8  1208 BC40  CLB      BAL      RDHALF(MR6)               FETCH HALFWORD (P.31)     83207280
1A9  13F8 5C80           BAL      CLB1(NULL)                (P.13)                    83207290
                    *                                                      * D5 *     83207300
1AA  1208 1B00  AL       BAL      CADRS(MR6)                CALCULATE ADDRESS (P.5)   83207310
1AB  13F8 5580           BAL      AL1(NULL)                 (P.13)                    83207320
                    *                                                      * D6 *     83207330
1AC  1208 BDC0  WB       BAL      RDFULL(MR6)               GET ADRS OF LIMITS (P.31) 83207340
1AD  13F8 50C0           BAL      WB1(NULL)                 (P.12)                    83207350
                    *                                                      * D7 *     83207360
1AE  1208 BDC0  RB       BAL      RDFULL(MR6)               GET ADRS OF LIMITS (P.31) 83207370
1AF  13F8 5140           BAL      RB1(NULL)                 (P.12)                    83207380
                    *                                                      * D8 *     83207390
1B0  1208 BC40  WH       BAL      RDHALF(MR6)               FETCH HALFWORD (P.31)     83207400
1B1  4BF9 DDB9           WHA      NULL,YD,MDR,ILIR,E,D                                83207410
                    *                                                      * D9 *     83207420
1B2  1208 5E80  RH       BAL      STORE(MR6)                COMMON ROUTINE (P.14)     83207430
1B3  4B79 CFA3           RHA      MDR,YD,NULL,DW2,E         EXECUTED INSTR.           83207440
                    *                                                      * DA *     83207450
1B4  1208 BC40  WD       BAL      RDHALF(MR6)               FETCH HALFWORD (P.31)     83207460
1B5  13F8 5D80           BAL      WD1(NULL)                 (P.13)                    83207470
                    *                                                      * DB *     83207480
1B6  1208 BC40  RD       BAL      RDHALF(MR6)               FETCH HALFWORD (P.31)     83207490
1B7  4B79 8DA3           RDA      MDR,YD,MDR,DW2,E                                    83207500
1B8  2BFF 1F99           L        NULL,NULL,ILIR,D                                    83207510
1B9  0000 0000           DC       0,0                                                 83207520
                    *                                                                 83207530
                    *                                                      * DD *     83207540
1BA  1208 BC40  SS       BAL      RDHALF(MR6)               FETCH HALFWORD (P.31)     83207550
1BB  13F8 5E00           BAL      SS1(NULL)                 (P.13)                    83207560
                    *                                                      * DE *     83207570
1BC  1208 BC40  OC       BAL      RDHALF(MR6)               FETCH HALFWORD (P.31)     83207580
1BD  4BF9 BDA0           OCA      NULL,YD,MDR,E                                       83207590
1BE  2BFF 1F99           L        NULL,NULL,ILIR,D                                    83207600
1BF  0000 0000           DC       0,0                                                 83207610
```

```
                    *                                                                   83207630
                    *                                                       * E0 *      83207640
1C0  12D8 1800  TS      BAL   CADRS(MR6)             CALCULATE ADDRESS (P.5)            83207650
1C1  239F 1839          LX    MAR,MR0,TS1            MAR=EFFECTIVE ADRS (P.18)          83207660
                    *                                                       * E1 *      83207670
1C2  2BFF 1F81  SVC     L     NULL,NULL,IL           INCREMENT LOC                      83207680
1C3  13F8 B540          BAL   SVC1(NULL)             (P.29)                             83207690
                    *                                                       * E2 *      83207700
1C4  2A3A 1D80  SINT    A     DEV,YX,MDR             DEV = I2+(X2)                      83207710
1C5  13F8 B440          BAL   SINT1(NULL)            (P.29)                             83207720
                    *                                                       * E3 *      83207730
1C6  12D8 1800  SCP     BAL   CADRS(MR6)             CALCULATE ADDRESS (P.5)            83207740
1C7  2B9F 1800          L     MAR,MR0                ADDRESS CCW                        83207750
1C8  13F8 ED80          BAL   SCP1(NULL)             (P.41)                             83207760
                    *                                                                   83207770
1C9  23DF 1811  BXLH1   LX    YDI,MR0,BXLH2          POINT BACK TO R1                   83207780
                    *                                                       * E5 *      83207790
1CA  12D8 1800  BDCS    BAL   CADRS(MR6)             CALCULATE ADDRESS (P.5)            83207800
1CB  03F8 0800          BAL   (MR0)(NULL)            DO BRANCH                          83207810
                    *                                                       * E6 *      83207820
1CC  2B9A 1D81  LA      A     MAR,YX,MDR,IL          CALCULATE ADDRESS          R02     83207830
1CD  13F8 0680          BAL   LA1(NULL)              (P.3)                              83207840
                    *                                                       * E7 *      83207850
1CE  2B9A 1D8F  TLATE   A     MAR,YX,MDR,DR4         FETCH TABLE ADDRESS                83207860
1CF  13F8 DD40          BAL   TLATE1(NULL)           (P.38)                             83207870
                    *                                                       * E8 *      83207880
1D0  13FD 2740  CCS     BALA  CCS1(NULL)             (P.49)                             83207890
1D1  233F 1893  BXLH2   LX    YD,MR1,BXLH3           INDEX TO R1                        83207900
                    *                                                       * E9 *      83207910
1D2  13F9 22C0  ECS     BAL   ECS1(NULL)             (P.49)                             83207920
1D3  23F2 08B8  BXLH3   SX    NULL,MR2,MR1,BXLH4     SUBTRACT FOR COMPARISON (P.18)     83207930
                    *                                                       * EA *      83207940
1D4  2A1A 1D89  RRL     A     MR0,YX,MDR,ILIR                                           83207950
1D5  2B39 A830          RR    YD,YD,MR0,E,D                                             83207960
                    *                                                       * EB *      83207970
1D6  2A1A 1D89  RLL     A     MR0,YX,MDR,ILIR                                           83207980
1D7  2B39 B830          RL    YD,YD,MR0,E,D                                             83207990
                    *                                                       * EC *      83208000
1D8  2A1A 1D89  SRL     A     MR0,YX,MDR,ILIR                                           83208010
1D9  2B39 8830          SRL   YD,YD,MR0,E,D                                             83208020
                    *                                                       * ED *      83208030
1DA  2A1A 1D89  SLL     A     MR0,YX,MDR,ILIR                                           83208040
1DB  2B39 9830          SLL   YD,YD,MR0,E,D                                             83208050
                    *                                                       * EE *      83208060
1DC  2A1A 1D89  SRA     A     MR0,YX,MDR,ILIR                                           83208070
1DD  2B39 C830          SRA   YD,YD,MR0,E,D                                             83208080
                    *                                                       * EF *      83208090
1DE  2A1A 1D89  SLA     A     MR0,YX,MDR,ILIR                                           83208100
1DF  2B39 D830          SLA   YD,YD,MR0,E,D                                             83208110
```

```
                    *                                                                        83208130
                    * COMMON SUBROUTINE FOR BXH,BXLE                                         83208140
                    *                                                                        83208150
1E0   2A39 1F80   BXLH    A       MR1,YD,NULL           MR1 = INDEX                          83208160
1E1   2A1F 1F00           L       MR0,YDI                                                    83208170
1E2   33D0 1001           AI      YDI,MR0,1             POINT TO R1+1                        83208180
1E3   2A39 1680           A       MR1,YD,MR1            INDEX PLUS INCREMENT                 83208190
1E4   33D0 1002           AI      YDI,MR0,2             POINT TO R1+2                        83208200
1E5   2259 1F89           AX      MR2,YD,NULL,BXLH1     MR2 = COMPARAND (P.17)               83208210
                    *                                                                        83208220
                    *                                                          * F3 *        83208230
1E6   2A1A 1D89   TI      A       MR0,YX,MDR,ILIR                                            83208240
1E7   2BF9 5830           N       NULL,YD,MR0,E,D                                            83208250
                    *                                                          * F4 *        83208260
1E8   2A1A 1D89   NI      A       MR0,YX,MDR,ILIR                                            83208270
1E9   2B39 5830           N       YD,XD,MR0,E,D                                              83208280
                    *                                                          * F5 *        83208290
1EA   2A1A 1D89   CLI     A       MR0,YX,MDR,ILIR                                            83208300
1EB   2BF9 0830           S       NULL,YD,MR0,E,D                                            83208310
                    *                                                          * F6 *        83208320
1EC   2A1A 1D89   OI      A       MR0,YX,MDR,ILIR                                            83208330
1ED   2B39 7830           O       YD,YD,MR0,E,D                                              83208340
                    *                                                          * F7 *        83208350
1EE   2A1A 1D89   XI      A       MR0,YX,MDR,ILIR                                            83208360
1EF   2B39 6830           X       YD,YD,MR0,E,D                                              83208370
                    *                                                          * F8 *        83208380
1F0   2A1A 1D89   LI      A       MR0,YX,MDR,ILIR                                            83208390
1F1   2B3F 1830           L       YD,MR0,E,D                                                 83208400
                    *                                                          * F9 *        83208410
1F2   2B7A 1D80   CI      A       MDR,YX,MDR                                                 83208420
1F3   13F8 04C0           BAL     CI(NULL)              (P.2)                                83208430
                    *                                                          * FA *        83208440
1F4   2A1A 1D80   AI      A       MR0,YX,MDR                                                 83208450
1F5   2B39 1839           A       YD,YD,MR0,ILIR,E,D                                         83208460
                    *                                                          * FB *        83208470
1F6   2A1A 1D80   SI      A       MR0,YX,MDR                                                 83208480
1F7   2B39 0839           S       YD,YD,MR0,ILIR,E,D                                         83208490
                    *                                                                        83208500
                    *                                                                        83208510
1F8   03F8 0D00   BXLH4   BAL     (MR6)(NULL)           RETURN TO CALL                       83208520
                    *                                                                        83208530
                    *                                                                        83208540
1F9   2B7F 1F88   TS1     L       MDR,NULL,RAS          READ HALFWORD & SET BIT              83208550
1FA   2BFF 1DA0           L       NULL,MDR,E            TEST IT                              83208560
1FB   17E4 7F19           BALNL   TS2(NULL),ILIR,D      TO TS2 IF NOT NEGATIVE               83208570
                    *                                                                        83208580
1FC   2B7F 2F83   TS2     SDEC    MDR,NULL,NULL,DW2     WRITE ALL ONES IF NOT SET            83208590
1FD   13F8 5300           BAL     TS3(NULL)             (P.12)                               83208600
                    *                                                                        83208610
                    *                                                                        83208620
                    *                                                                        83208630
```

```
                  * * * * * * * * * * * * * * * * * * * * * * * * *                 83208650
                  *                                               *                 83208660
                  *          M A C   I N T E R R U P T            *                 83208670
                  *                                               *                 83208680
                  * * * * * * * * * * * * * * * * * * * * * * * * *                 83208690
                  *                                                                 83208700
                  *                                                                 83208710
                  *                                                                 83208720
1FE  323F 1090  MACINT LI    MR1,'90'             ADRS OF MAC INTERRUPT NEW PSW     83208730
1FF  1378 8280         BAL   COMINT(MDR)          TO COMMON ROUTINE (P.21)          83208740
                  *                                                                 83208750
                  *                                                                 83208760
                  * ON INSTRUCTION READS, IF ENABLED BY PSW BIT 21,                 83208770
                  * THE OCCURANCE OF INVALID ADDRESS, NON-PRESENT                   83208780
                  * ADDRESS OR EXECUTE PROTECT JAMS 'FF' INTO THE                   83208790
                  * OP-CODE FIELD OF IR, CAUSING THE VECTOR TO '1FE'.               83208800
                  *                                                                 83208810
```

```
                          ORG    '200'
                     *
                     * * * * * * * * * * * * * * * * * * * * * * * * *
                     *                                               *
                     *      I N T E R R U P T   V E C T O R S        *
                     *                                               *
                     * * * * * * * * * * * * * * * * * * * * * * * * *
                     *
                     *
                     *
200  177C A780            BALD   IOINT3(MDR)           I/O INTERRUPT LEVEL 3 (P.26)
201  177C A700            BALD   IOINT2(MDR)           I/O INTERRUPT LEVEL 2 (P.26)
202  177C A680            BALD   IOINT1(MDR)           I/O INTERRUPT LEVEL 1 (P.26)
203  177C A640            BALD   IOINT0(MDR)           I/O INTERRUPT LEVEL 0 (P.26)
204  177C 9000            BALD   CONSER(MDR)           CONSOLE ATTENTION (P.23)
205  177C 8C00            BALD   MMFINT(MDR)           MACHINE MALFUNCTION (P.22)
206  177C 84C0            BALD   PPFINT(MDR)           PRIMARY POWER FAIL (P.21)
207  177C 7F80            BALD   MACINT(MDR)           MEMORY ACCESS CONTROLLER
208  177C 8240            BALD   ILEGAL(MDR)           ILLEGAL INSTRUCTION (P.21)
                     *
                     * GENERAL REGISTER ASSIGNMENTS
                     *
          0000       R0   EQU    0
          0001       R1   EQU    1
          0002       R2   EQU    2
          0003       R3   EQU    3
          0004       R4   EQU    4
          000D       R13  EQU    13
          000E       R14  EQU    14
          000F       R15  EQU    15
```

```
                * * * * * * * * * * * * * * * * * * * * * * * * *                 83209030
                *                                               *                 83209040
                *    I L L E G A L   I N S T R U C T I O N      *                 83209050
                *                                               *                 83209060
                * * * * * * * * * * * * * * * * * * * * * * * * *                 83209070
                *                                                                 83209080
                *                                                                 83209090
                *                                                                 83209100
209  323F 1030  ILLEGAL LI    MR1,'30'              ADRS OF ILLEGAL INSTR. NEW PSW      83209110
                *                                                                 83209120
                *                                                                 83209130
                *                                                                 83209140
20A  321F 1000  COMINT  LI    MR0,0                 MR0=CC TO OR IN                     83209150
20B  289F 1880  COMIN0  L     MAR,MR1                                                   83209160
20C  2A5D 1F8E          A     MR2,PSW,NULL,PR4      SAVE CURRENT PSW                    83209170
20D  3391 1004  COMIN1  AI    MAR,MR1,4                                                 83209180
20E  2BBF 108E          L     PSW,MDR,PR4           LOAD NEW PSW                        83209190
20F  29DF 1900  COMIN2  L     R14,MR2               REG 14 = OLD PSW                    83209200
210  29FF 1000          L     R15,LOC               REG 15 = OLD LOC                    83209210
211  2BBD 7800          O     PSW,PSW,MR0           SET CONDITION CODE                  83209220
212  235F 10BD          LX    LOC,MDR,TWAIT         LOAD NEW LOC (P.22)                 83209230
                *                                                                 83209240
                * GO TO TEST WAIT BIT OF NEW PSW                                  83209250
                *                                                                 83209260
                *                                                                 83209270
                *                                                                 83209280
                * * * * * * * * * * * * * * * * * * * * * * * * *                 83209290
                *                                               *                 83209300
                *     P R I M A R Y   P O W E R   F A I L       *                 83209310
                *                                               *                 83209320
                * * * * * * * * * * * * * * * * * * * * * * * * *                 83209330
                *                                                                 83209340
                *                                                                 83209350
                *                                                                 83209360
213  339F 1084  PPFINT  LI    MAR,'84'              ADRS OF CURRENT PSW SAVE POINTER    83209370
214  2BDF 1F8A          L     YD1,NULL,PR2          FETCH POINTER                       83209380
215  361F 1017          LI    MR0,LOHALF,I                                              83209390
216  2A10 5D80          N     MR0,MR0,MDR           USE LS 16 BITS                      83209400
217  289F 1800          L     MAR,MR0                                                   83209410
218  2B7D 1F86          A     MDR,PSW,NULL,PW4      SAVE PSW                            83209420
219  3390 1004          AI    MAR,MR0,4             INCREMENT                           83209430
21A  2B7F 1006          L     MDR,LOC,PW4           SAVE LOC                            83209440
21B  339F 1086          LI    MAR,X'86'             ADRS OF REGISTER SAVE POINTER       83209450
21C  2BBF 1F8A          L     PSW,NULL,PR2          SELECT REGISTER SET 0               83209460
21D  2A1F 1D80          L     MR0,MDR                                                   83209470
21E  3610 5017          NI    MR0,MR0,LOHALF,1      USE LS 16 BITS                      83209480
21F  3390 0004          SI    MAR,MR0,4             TEMPORARY DECREMENT                 83209490
220  323F 000F          SI    MR1,NULL,15           MR1='FFFFFFF1'                      83209500
                *                                                                 83209510
221  2B7F 1C85  STRLP   L     MDR,YD,I4DW4                                              83209520
222  23D1 1F61          AX    YDI,MR1,YDI,STRLP,C                                       83209530
223  33BD 1010          AI    PSW,PSW,'010'         INCREMENT REGISTER SET NUMBER       83209540
224  4AFF 7F80          SMCR  MR7,NULL              TEST IF 2 OR 8 REGISTER SETS        83209*43
225  33F7 5040          NI    NULL,MR7,'40'         MCR BIT 9 = 0 IF 2 SETS             83209544
226  17E0 8A00          BALNZ **+2(NULL)            MCR BIT 9 = 1 IF 8 SETS             83209545
227  33BD 7070          OI    PSW,PSW,'70'          IF ZERO, FORCE LAST SET             83209546
228  33FD 5080          NI    NULL,PSW,'80'         TEST IF LAST SET STORED             83209550
```

```
229  13E0 8840        BALZ  STRLP(NULL)           LOOP UNTIL ALL SETS (P.21)           83209560
                *                                                                    83209570
                *                                                                    83209580
22A  13F8 8AC0        BAL   *+1(NULL)             FOR D FLOAT                          83209590
22B  339F 0004        SI    MAR,NULL,4            MAR='FFFFFFFC'                       83209600
22C  323F 000E        SI    MR1,NULL,14           MR1='FFFFFFF2'                       83209610
                *                                                                    83209620
22D  6B79 1F85  STREE AU    MDR,YD,NULL,I4DW4     SAVE FLOATING POINT REGISTER         83209630
22E  23D1 1F6D        AX    YD1,MR1,YDI,STREE,C                                        83209640
                *                                                                    83209650
22F  487F FF80        POW   MDR,NULL              POWER DOWN                           83209660
                *                                                                    83209670
                *                                                                    83209680
                *                                                                    83209690
                * * * * * * * * * * * * * * * * * * * * * * * * *                    83209700
                *                                               *                    83209710
                *     M A C H I N E   M A L F U N C T I O N     *                    83209720
                *                                               *                    83209730
                * * * * * * * * * * * * * * * * * * * * * * * * *                    83209740
                *                                                                    83209750
                *                                                                    83209760
                *                                                                    83209770
230  321F 1000  MMFINT LI   MR0,0                                                      83209780
231  339F 1020  MMF1   LI   MAR,'20'                                                   83209790
232  2B7D 1F86         A    MDR,PSW,NULL,PW4                                           83209800
233  339F 1024         LI   MAR,'24'                                                   83209810
234  2B7F 1D06         L    MDR,LOC,PW4           SAVE LOC                             83209820
235  339F 1038         LI   MAR,'38'                                                   83209830
236  2BFF 1F8E         L    NULL,NULL,PR4         FETCH NEW PSW                        83209840
237  339F 103C         LI   MAR,'3C'                                                   83209850
238  2BBF 1080         L    PSW,MDR               LOAD NEW PSW                         83209860
239  4AFF 7BAE         SMCR MR7,MR7,PR4,E         JAM MCR 12:15 TO CC                  83209870
23A  4AFF 7BC0         CMCR MR7,MR7               CLEAR MCR BITS                       83209880
23B  2BBD 7800         O    PSW,PSW,MR0           SET C FLAG TO INDICATE               83209890
23C  2B5F 1080         L    LOC,MDR               MALF DURING AUTO DRIVER              83209900
                *                                                                    83209910
                *                                                                    83209920
23D  4BFF 6FC0  TWAIT  LWFF NULL,NULL             RESET WAIT INDICATOR                 83209930
23E  37FD 5327         NI   NULL,PSW,BIT16,1      TEST PSW WAIT BIT                    83209940
23F  17E0 ABDC         BALNZ WAIT(NULL),IR,D      (P.27)                               83209950
```

```
                 * * * * * * * * * * * * * * * * * * * * * * * * *                      83209970
                 *                                               *                      83209980
                 *        C O N S O L E   S E R V I C E          *                      83209990
                 *                                               *                      83210000
                 * * * * * * * * * * * * * * * * * * * * * * * * *                      83210010
                 *                                                                      83210020
                 *                                                                      83210030
                 *                                                                      83210040
240  323F 1001   CONSER LI     MR1,1                  DISPLAY DEVICE NUMBER             83210050
241  4BFF 6FC0          LWFF   NULL,NULL              RESET WAIT INDICATOR              83210060
242  4A51 AFC0          SSRA   MR2,MR1,NULL           ADDRESS & SENSE STATUS            83210070
243  3252 7F00          OI     MR2,MR2,'F00'          EXTEND BIT 0 OF STATUS BYTE       83210080
244  3252 6080          XI     MR2,MR2,'80'                                             83210090
245  3252 1080          AI     MR2,MR2,'80'                                             83210100
246  3252 B001          RLI    MR2,MR2,1              8 BIT ROTATE LEFT                 83210110
247  339F 1028          LI     MAR,'28'                                                 83210120
248  2B7F 1902          L      MDR,MR2,PW2            SAVE STATUS BYTE                  83210130
249  33F2 5080          NI     NULL,MR2,'080'         TEST STATUS BIT 1                 83210140
24A  17E0 9E40          BALNZ  FNDIS(NULL)            FUNCTION OR REGISTER (P.24)       83210150
24B  2B9F 1D00          L      MAR,LOC                                                  83210160
24C  33F2 5020          NI     NULL,MR2,'020'         TEST STATUS BIT 3                 83210170
24D  17E0 94CB          BALNZ  ADRMW(NULL),DR2        ADDRESS OR MEMORY WRITE           83210180
24E  33F2 5040          NI     NULL,MR2,'040'         TEST STATUS BIT 2                 83210190
24F  17E0 95C0          BALNZ  DISMEM(NULL)           MEMORY READ                       83210200
                 *                                                                      83210210
                 * RUN MODE                                                             83210220
                 *                                                                      83210230
                 *                                                                      83210240
250  37B0 5023   CLRWT  NI     PSW,PSW,BIT160,1       RESET PSW 16                      83210250
251  4BFF 6FC0          LWFF   NULL,NULL              RESET WAIT                        83210260
252  17F8 941C          BDC    CLRWT(NULL),IR,D       EXECUTE NEXT USER INSTR           83210270
                 *                                                                      83210280
                 *                                                                      83210290
                 * ADDRESS OR MEMORY WRITE                                              83210300
                 *                                                                      83210310
253  4B7F 0FC0   ADRMW  RDR    MDR,NULL               READ SWITCH REGISTER 12:19        83210320
254  4B7F 0D80          RD     MDR,MDR                READ SWITCH REGISTER 04:11        83210330
255  33F2 5040          NI     NULL,MR2,'040'         TEST STATUS BIT 2                 83210340
256  17E0 9843          BALNZ  ADRS(NULL),DW2         ADDRESS (P.24)                    83210350
                 *                                                                      83210360
                 *                                                                      83210370
                 * MEMORY WRITE                                                         83210380
                 *                                                                      83210390
                 * DISPLAY MEMORY REGISTERS                                             83210400
                 *                                                                      83210410
257  2A1F 1D80   DISMEM L      MR0,MDR                LS 16 BITS TO MR0                 83210420
258  3610 5017          NI     MR0,MR0,LOHALF,1                                         83210430
259  2A3F 1D00          L      MR1,LOC                                                  83210440
25A  3351 1002          AI     LOC,MR1,2              INCREMENT LOC                     83210450
25B  3231 1002          AI     MR1,MR1,2                                                83210460
25C  3271 9010          SLLI   MR3,MR1,16             POSITION MS 16 BITS               83210470
25D  2A10 7980          O      MR0,MR0,MR3            MR0=DISPLAY BYTES D4,D3,D2,D1     83210480
25E  3231 8010          SRLI   MR1,MR1,16             FORM DISPLAY BYTE D5, EQUAL       83210490
25F  3231 7080          OI     MR1,MR1,'080'          TO LOC 0:3 & INDICATOR FOR        83210500
260  13F8 9A00          BAL    OUTDIS(NULL)           ADDRESS & MEMORY DATA (P.24)      83210510
                 *                                                                      83210520
                 *                                                                      83210530
```

```
                    *                                                                  83210540
                    *                                                                  83210550
                    *                                        ADDRESS                   83210560
261   2A1F 1080   ADRS    L      MR0,MDR                                               83210570
262   3610 5039           NI     MR0,MR0,CFFFE,I             LS 16 BITS, FORCED EVEN   83210580
263   3232 500F           NI     MR1,MR2,'0F'                MS 4 BITS IN STATUS BYTE  83210590
264   3231 9010           SLLI   MR1,MR1,16                  POSITION  BITS            83210600
265   2851 7800           O      LOC,MR1,MR0                 LOAD 20 BIT LOCATION COUNT 83210610
                    *                                                                  83210620
                    *                                                                  83210630
                    * DISPLAY THE LOCATION COUNTER                                     83210640
                    *                                                                  83210650
266   2A1F 1000   LOCDIS  L      MR0,LOC                     MR0=DISPLAY BYTES D4,D3,D2,D1 83210660
267   323F 1045           LI     MR1,'45'                    MR1=D5=FUNCTION 5         83210670
                    *                                                                  83210680
                    *                                                                  83210690
                    * OUTPUT DATA TO THE DISPLAY                                       83210700
                    *                                                                  83210710
268   4BFF 1840   OUTDIS  WDR    NULL,MR0                    DISPLAY BYTE D1           83210720
269   4BFF 1800           WD     NULL,MR0                    DISPLAY BYTE D2           83210730
26A   3210 8010           SRLI   MR0,MR0,16  ALU OP                                    83210740
26B   4BFF 1840           WDR    NULL,MR0                    DISPLAY BYTE D3           83210750
26C   4BFF 1800           WD     NULL,MR0                    DISPLAY BYTE D4           83210760
26D   4BFF 18C0           WDR    NULL,MR1                    DISPLAY BYTE D5: 1N=FLT REG N  83210770
                    *                                                      2N=GEN REG N  83210780
                    *                                                     4N=FUNCTION N  83210790
                    *                                                      8N=ADRS/DATA  83210800
26E   321F 1080           LI     MR0,'80'                                              83210810
26F   4BFF 3840           OCR    NULL,MR0                    OUTPUT COMMAND NORMAL     83210820
                    *                                                                  83210830
270   321F 0001   IDLE    SI     MR0,NULL,1                                            83210840
271   4BFF 6840           LWFF   NULL,MR0                    SET WAIT INDICATOR        83210850
272   4AFF 7B80   IDLE1   SMCR   MR7,MR7                                               83210860
273   13E4 84C0           BALL   PPFINT(NULL)                EARLY POWER FAIL (P.21)   83210870
274   33F7 5020           NI     NULL,MR7,'20'               CATN?                     83210880
275   13E0 9C80           BALZ   IDLE1(NULL)                 NO, LOOP                  83210890
                    *                                                                  83210900
276   33F7 5100           NI     NULL,MR7,'100'              SINGLE STEP?              83210910
277   13E0 9000           BALZ   CONSER(NULL)                NO (P.23)                 83210920
278   13F8 9400           BAL    CLRWT(NULL)                 SINGLE STEP (P.23)        83210930
                    *                                                                  83210940
                    *                                                                  83210950
                    * FUNCTION OR REGISTER DISPLAY                                     83210960
                    *                                                                  83210970
279   2BDF 1900   FNDIS   L      YDI,MR2                                               83210980
27A   33F2 5010           NI     NULL,MR2,'010'              TEST STATUS BIT 4         83210990
27B   13E0 A180           BALZ   FN(NULL)                    FUNCTION (P.25)           83211000
                    *                                                                  83211010
                    * GENERAL REGISTER OR FLOATING REGISTER                            83211020
                    *                                                                  83211030
27C   2A1F 1C80           L      MR0,YD                      FETCH GENERAL REGISTER    83211040
27D   323F 1020           LI     MR1,'20'                                              83211050
27E   2A31 7F00           O      MR1,MR1,YDI                                           83211060
27F   33F2 5020           NI     NULL,MR2,'020'              TEST STATUS BIT 3         83211070
280   13E0 9A00           BALZ   OUTDIS(NULL)                GENERAL REGISTER          83211080
                    *                                                                  83211090
                    *                                                                  83211100
```

```
                     *                                                                 83211110
                     * SINGLE PRECISION FLOATING-POINT REGISTER                         83211120
                     *                                                                 83211130
281  33D2 500E  SFLOAT NI    YDI,MR2,'E'          FORCE EVEN REGISTER NUMBER             83211140
282  3231 50FE         NI    MR1,MR1,'0FE'                                             83211150
283  6A19 1F80         AU    MR0,YD,NULL          MR0=FLOATING POINT REGISTER            83211160
284  3231 0010         SI    MR1,MR1,'10'                                              83211170
285  13F8 9A00         BAL   OUTDIS(NULL)         (P.24)                                83211180
                     *                                                                 83211190
                     *                                                                 83211200
                     * FUNCTION                                                        83211210
                     *                                                                 83211220
286  33F2 500A  FN     NI    NULL,MR2,'00A'                                            83211230
287  17E0 9C00         BALNZ IDLE(NULL)           IDLE IF UNDEFINED (P.24)              83211240
288  33F2 5004         NI    NULL,MR2,'004'                                            83211250
289  13E0 A3C0         BALZ  FN01(NULL)           FUNCTION 0 OR 1                       83211260
                     *                                                                 83211270
                     * FUNCTION 4 OR 5                                                 83211280
                     *                                                                 83211290
28A  33F2 5001         NI    NULL,MR2,'001'                                            83211300
28B  17E0 9980         BALNZ LOCDIS(NULL)         FUNCTION 5 = LOC (P.24)               83211310
                     *                                                                 83211320
                     * FUNCTION 4 = PSW                                                83211330
                     *                                                                 83211340
28C  2A1D 1F80  PSWDIS A     MR0,PSW,NULL         LOAD PSW                              83211350
28D  323F 1044         LI    MR1,'44'             FUNCTION 4                            83211360
28E  13F8 9A00         BAL   OUTDIS(NULL)         (P.24)                                83211370
                     *                                                                 83211380
                     * FUNCTION 0 OR 1                                                 83211390
                     *                                                                 83211400
28F  33F2 5001  FN01   NI    NULL,MR2,'001'                                            83211410
290  13E0 A540         BALZ  FN0(NULL)            FUNCTION 0                            83211420
                     *                                                                 83211430
                     * FUNCTION 1 = SELECT PSW BITS                                    83211440
                     *                                                                 83211450
291  37BD 5015         NI    PSW,PSW,HIHALF,I     CLEAR LS 16 BITS                      83211460
292  4A5F 0FC0         RDR   MR2,NULL             INPUT LS BYTE                         83211470
293  4A5F 0900         RD    MR2,MR2              INPUT MS BYTE                         83211480
294  23BD 790C         OX    PSW,PSW,MR2,PSWDIS   OR INTO PSW THEN DISPLAY PSW          83211490
                     *                                                                 83211500
                     * FUNCTION 0 = CONSOLE INTERRUPT                                  83211510
                     *                                                                 83211520
295  37FD 5021  FN0    NI    NULL,PSW,BI1720,I    TEST PSW BIT 17 OR 20                 83211530
296  13E0 9400         BALZ  CLRWI(NULL)          IGNORE IF NO ENABLE (P.23)            83211540
297  323F 1001         LI    DEV,1                DEVICE NUMBER 1                       83211550
298  225F 1FA0         LX    LEVEL,NULL,IOINTX    REGISTER SET 0 (P.26)                 83211560
                     *                                                                 83211570
                     * IMMEDIATE INTERRUPT                                             83211580
                     *                                                                 83211590
```

```
                  * * * * * * * * * * * * * * * * * * * * * * * * *
                  *                                               *
                  *   I / O   D E V I C E   I N T E R R U P T S   *
                  *                                               *
                  * * * * * * * * * * * * * * * * * * * * * * * * *
                  *
                  *
                  *
                  * REGISTER ASSIGNMENTS FOR CHANNEL I/O
                  *
                  *
             0010 TEMP   EQU   '10'
             0011 DEV    EQU   '11'
             0012 LEVEL  EQU   '12'
             0012 CCW    EQU   '12'
             0013 DAT    EQU   '13'
             0014 COUNT  EQU   '14'
             0015 RETURN EQU   '15'
                  *
                  *
299  225F 1F9F    IOINT0 LX    LEVEL,NULL,AUTOIO      SELECT REGISTER SET 0
                  *
29A  325F 1011    IOINT1 LI    LEVEL,'11'             SELECT REGISTER SET 1
29B  423F 6920           AKX   DEV,LEVEL,IOINTX       ACKNOWLEDGE INTERRUPT
                  *
29C  325F 1022    IOINT2 LI    LEVEL,'22'             SELECT REGISTER SET 2
29D  423F 6920           AKX   DEV,LEVEL,IOINTX       ACKNOWLEDGE INTERRUPT
                  *
29E  325F 1033    IOINT3 LI    LEVEL,'33'             SELECT REGISTER SET 3
29F  4A3F 6900    AUTOIO AK    DEV,LEVEL              ACKNOWLEDGE INTERRUPT
                  *
                  *
2A0  3231 53FF    IOINTX NI    DEV,DEV,'3FF'          MASK 10 BIT DEVICE NUMBER
2A1  3271 10D0           AI    DAT,DEV,'D0'           2X DEVICE NUMBER + 'D0'
2A2  2B93 188A           A     MAR,DAT,DEV,PR2        INDEXES SERVICE POINTER TABLE
                  *                                   FETCH APPROPRIATE ENTRY
                  *
2A3  2A1D 1F80           A     TEMP,PSW,NULL          SAVE PSW
2A4  3252 50F0           NI    LEVEL,LEVEL,'F0'       MASK REGISTER SET BITS
2A5  37B2 7053           OI    PSW,LEVEL,BI1820,I     SET PSW BITS 18 & 20 AND
                  *                                   SELECT REGISTER SET
                  *
2A6  281F 1800           L     R0,TEMP                REG 0 = PSW
2A7  283F 1000           L     R1,LOC                 REG 1 = LOC
2A8  285F 1880           L     R2,DEV                 REG 2 = DEVICE NUMBER
2A9  4871 AFE0           SSRA  R3,DEV,NULL,E          REG 3 = DEVICE STATUS, SET CC
                  *
2AA  2A7F 1D80           L     DAT,MDR                TABLE ENTRY TO DAT
2AB  3753 5039           NI    LOC,DAT,CFFFE,I        LOAD LOC JUST IN CASE
2AC  4BFF 6FC0           LWFF  NULL,NULL              RESET WAIT INDICATOR
2AD  33F3 5001           NI    NULL,DAT,1             TEST LSB OF SERVICE POINTER
2AE  17E1 0E1C           BALNZ CHANEL(NULL),IR,D      (P.45)
                  *                                   IF SET, SERVICE POINTER IS
                  *                                   ADDRESS PLUS ONE OF CCB
                  *                                   IF RESET, SERVICE POINTER IS
                  *                                   ADDRESS IN FIRST 64K OF A
                  *                                   USER'S SERVICE SUBROUTINE
                  *                                                                    83212180
                  * * * * * * * * * * * * * * * * * * * * * * * * *                    83212200
                  *                                               *                    83212210
                  *           I N T E R R U P T A B L E           *                    83212220
                  *                                               *                    83212230
                  *               W A I !   L O O P               *                    83212240
                  *                                               *                    83212250
                  * * * * * * * * * * * * * * * * * * * * * * * * *                    83212260
                  *                                                                    83212270
                  *                                                                    83212280
                  *                                                                    83212290
2AF   321F 0001   WAIT   SI     MR0,NULL,1                                             83212300
2B0   4BFF 6840          LWFF   NULL,MR0              SET WAIT INDICATOR               83212310
2B1   13FC AC50   WAIT1  BALA   WAIT1(NULL),D         LOOP                             83212320
                  *                                                                    83212330
                  *                                                                    83212340
                  * L O A D   P R O G R A M   S T A T U S   W O R D                    83212350
                  *                                                                    83212360
2B2   2A7F 1D8D   LPSW1  L      MR3,MDR,14DR4         SET PSW ASIDE IN MR3             83212370
2B3   2B5F 1D80          L      LOC,MDR                                                83212380
                  *                                                                    83212390
                  *                                                                    83212400
2B4   339F 1080   TEST1  LI     MAR,'80'                                               83212410
2B5   2BBF 1980          L      PSW,MR3               LOAD NEW PSW                     83212420
2B6   337D 5200          NI     MDR,PSW,'200'         TEST IF QUEUE SERVICE ENABLED    83212430
2B7   13E0 8F4E          BALZ   TWAIT(NULL),PR4       TEST WAIT BIT IF NOT (P.22)      83212440
                  *                                                                    83212450
                  *                                                                    83212460
                  * QUEUE SERVICE IS ENABLED. TEST THE QUEUE                           83212470
                  *                                                                    83212480
2B8   321F 1002   QUEINT LI     MR0,2                                                  83212490
2B9   2A7F 1D80          L      MR3,MDR               SAVE ADDRESS OF QUEUE            83212500
2BA   2B90 1D8A          A      MAR,MR0,MDR,PR2       FETCH NO. USED HALFWORD          83212510
2BB   2A5D 1F80          A      MR2,PSW,NULL          SAVE PSW                         83212520
2BC   2BFF 1D80          L      NULL,MDR              TEST TALLY                       83212530
2BD   13E0 8F40          BALZ   TWAIT(NULL)           EXIT IF ZERO (P.22)              83212540
                  *                                                                    83212550
                  * QUEUE IS NOT EMPTY, DO INTERRUPT                                   83212560
                  *                                                                    83212570
2BE   339F 1088          LI     MAR,'88'              ADRS OF QUEUE SERVICE NEW PSW    83212580
2BF   2A1F 1F8E          L      MR0,NULL,PR4          FETCH NEW PSW                    83212590
2C0   2BBF 1D80          L      PSW,MDR               LOAD NEW PSW                     83212600
2C1   339F 108C          LI     MAR,'8C'                                               83212610
2C2   29BF 198E          L      R13,MR3,PR4           REG 13 = ADRS OF QUEUE           83212620
2C3   13F8 83C0          BAL    COMIN2(NULL)          TO COMMON ROUTINE (P.21)         83212630
```

```
                * * * * * * * * * * * * * * * * * * * * * * * * *                       83212650
                *                                               *                       83212660
                *        A R I T H M E T I C   F A U L T        *                       83212670
                *                                               *                       83212680
                * * * * * * * * * * * * * * * * * * * * * * * * *                       83212690
                *                                                                       83212700
                *                                                                       83212710
                *                                                                       83212720
                *                                D,DR DIVIDE FAULT                      83212730
2C4  283F 1800  DFALT0 L      YD,MR0                RESTORE (R1)                        83212740
2C5  2BDF 3F00         AINC   YD1,NULL,YD1                                              83212750
2C6  283F 1880         L      YD,MR1                RESTORE (R1+1)                      83212760
                *                                  DH,DHR DIVIDE FAULT                  83212770
2C7  221F 1F8D  DFALT1 LX     MR0,NULL,AFAULT       NO CC                               83212780
                *                                                                       83212790
                *                                                                       83212800
                *                                                                       83212810
2C8  13E0 B300  FFAULT BALZ   FFALT1(NULL)          ZERO = UNDERFLOW                    83212820
2C9  6A19 1F80         AU     MR0,YD,NULL           IF OVERFLOW                         83212830
2CA  3610 713B         OI     MR0,MR0,ONES,I        FORCE LARGEST MAGNITUDE             83212840
2CB  6B30 1F80         AU     YD,MR0,NULL                                               83212850
2CC  321F 1008  FFALT1 LI     MR0,8                 FLOATING POINT FAULT                83212860
                *                                  CARRY FLAG WILL BE SET               83212870
                *                                                                       83212880
2CD  325F 1048  AFAULT LI     MR1,'48'                                                  83212890
                *                                                                       83212900
2CE  37FD 532A         NI     NULL,PSW,BIT19,I      TEST ARITHMETIC FAULT ENABLE        83212910
2CF  17E0 B410         BALNZ  *+1(NULL),D           TAKE INTERRUPT IF ENABLED           83212920
                *                                                                       83212930
                * DO NEXT USER INSTRUCTION IF BIT 19 RESET                              83212940
2D0  1378 82C0         BAL    COMINO(MDR)           COMMON ROUTINE (P.21)               83212945
```

```
                      *                                                                        83212960
                      *      S I M U L A T E   I N T E R R U P T                               83212970
                      *                                                                        83212980
                      *                                                                        83212990
                      *                                                                        83213000
2D1   2A5F 1F01  SINT1   L     LEVEL,YDI,IL          R1 FIELD TO "LEVEL"                       83213010
2D2   1360 A800          BALZ  IOINTX(MDR)           R1 FIELD IS ZERO (P.26)                   83213020
2D3   3259 9004          SLLI  LEVEL,YD,4            IF NON-ZERO, SELECT REGISTER SET          83213030
2D4   13F8 A800          BAL   IOINTX(NULL)          DO I/O INTERRUPT (P.26)                   83213040
                      *                                                                        83213050
                      *                                                                        83213060
                      *                                                                        83213070
                      *      S U P E R V I S O R   C A L L                                     83213080
                      *                                                                        83213090
2D5   2B9A 1D80  SVC1    A     MAR,YX,MDR            CALCULATE ADDRESS                         83213100
2D6   2A1F 1E00          L     MR0,MAR               SET ASIDE IN MR0                          83213110
2D7   339F 1098          LI    MAR,'98'              ADDRESS SVC NEW PSW                       83213120
2D8   2B7F 1F8E          L     MDR,NULL,PR4          FETCH NEW PSW STATUS                      83213130
2D9   2A5D 1F80          A     MR2,PSW,NULL          SAVE CURRENT PSW                          83213140
2DA   2BBF 1D80          L     PSW,MDR               LOAD NEW PSW                              83213150
2DB   29BF 1600          L     R13,MR0               REG 13 = EFFECTIVE ADDRESS                83213160
2DC   29DF 1900          L     R14,MR2               REG 14= PSW                               83213170
2DD   29FF 1D00          L     R15,LOC               REG 15 = LOC                              83213180
2DE   2A3F 1F00          L     MR1,YDI               COLLECT R1 FIELD                          83213190
2DF   2A31 1880          A     MR1,MR1,MR1           2X R1 FIELD PLUS X'9C'                    83213200
2E0   3391 109C          AI    MAR,MR1,'9C'          IS HALFWORD ADRS                          83213210
2E1   2BFF 1F8A          L     NULL,NULL,PR2         OF SVC NEW LOC                            83213220
2E2   363F 1039          LI    MR1,CFFFE,I           MR1 = '0000FFFE'                          83213230
2E3   2B51 5080          N     LOC,MR1,MDR           LOAD NEW LOC                              83213240
2E4   2BFF 1F9C          L     NULL,NULL,IR,D                                                  83213250
```

```
                  * * * * * * * * * * * * * * * * * * * * * * * * *                 83213270
                  *                                               *                 83213280
                  * COMMON SUBROUTINE FOR TBT, SBT, RBT & CBT     *                 83213290
                  *                                               *                 83213300
                  * * * * * * * * * * * * * * * * * * * * * * * * *                 83213310
                  *                                                                 83213320
                  *                                                                 83213330
                  *                                                                 83213340
2E5  239A 1DE8    COMBIT AX     MAR,YX,MDR,COMBT1,C   TRANSFER IF RX1 OR RX3        83213350
2E6  321F 1D04           LI     MR0,4                                               83213360
2E7  2210 1E29           AX     MR0,MR0,MAR,COMBT2    ADD 4 IF RX2                  83213370
2E8  2A1F 1E00    COMBT1 L      MR0,MAR                                             83213380
                  *                                                                 83213390
                  * MR0 = MATRIX START ADDRESS                                      83213400
                  * R1 CONTAINS DISPLACEMENT TO DESIRED BIT                         83213410
                  *                                                                 83213420
2E9  3239 8004    COMBT2 SRLI   MR1,YD,4              ON HALFWORD BOUNDARY,         83213430
2EA  2B71 1880           A      MDR,MR1,MR1           ADDRESS A HALFWORD IN         83213440
2EB  2B90 1D8B           A      MAR,MR0,MDR,DR2       THE ARRAY & FETCH IT          83213450
2EC  3259 500F           NI     MR2,YD,'F'            MASK LS 4 BITS TO TEST        83213460
                  *                                   A BIT IN THE HALFWORD         83213470
2ED  3252 1327           AI     MR2,MR2,BTABLE        FORM VECTOR ADDRESS           83213480
2EE  2E5F 1900           L      MR2,MR2,1             FETCH BIT MASK                83213490
2EF  2BF2 5CA0           N      NULL,MR2,MDR,E        TEST THE BIT, SET CC          83213500
2F0  03F8 0A80           BAL    (MR5)(NULL)           RETURN                        83213510
                  *                                                                 83213520
                  * * * * * * * * * * * * * * * * * * * * * * * * *
                  *                                               *
                  * COMMON ROUTINE CALCULATES EFFECTIVE ADDRESS   *
                  * THEN READS HALFWORD FROM MAIN MEMORY          *
                  *                                               *
                  * * * * * * * * * * * * * * * * * * * * * * * * *
                  *
                  *
                  *
2F1  239A 1DF4    RDHALF AX     MAR,YX,MDR,RHALF1,C    TRANSFER IF RX1 OR RX3
2F2  321F 1004           LI     MR0,4                  ADD 4 IF RX2
2F3  2390 1E35           AX     MAR,MR0,MAR,RHALF2     LOAD ADDRESS
2F4  2B9F 1E00    RHALF1 L      MAR,MAR                LOAD ADDRESS
2F5  2B7F 1F8B    RHALF2 L      MDR,NULL,DR2           FETCH HALFWORD
2F6  03F8 0800           BAL    (MR6)(NULL)            RETURN TO CALL
                  *
                  *
                  *
                  * * * * * * * * * * * * * * * * * * * * * * * * *
                  *                                               *
                  * COMMON ROUTINE CALCULATES EFFECTIVE ADDRESS   *
                  * THEN READS FULLWORD FROM MAIN MEMORY          *
                  *                                               *
                  * * * * * * * * * * * * * * * * * * * * * * * * *
                  *
                  *
                  *
2F7  239A 1DFA    RDFULL AX     MAR,YX,MDR,RFULL1,C    TRANSFER IF RX1 OR RX3     S=1A3
2F8  321F 1004           LI     MR0,4                  ADD 4 IF RX2
2F9  2390 1E3B           AX     MAR,MR0,MAR,RFULL2     LOAD ADDRESS
2FA  2B9F 1E00    RFULL1 L      MAR,MAR                LOAD ADDRESS               B=7B
2FB  2B7F 1F8F    RFULL2 L      MDR,NULL,DR4           FETCH FULLWORD
2FC  03F8 0800           BAL    (MR6)(NULL)            RETURN TO CALL
                  *
                  *
                  *
                  * LOAD HALFWORD LOGICAL
                  *
2FD  361F 1017    LHL1   LI     MR0,LOHALF,I           MR0='0000FFFF'
2FE  2B30 5DB9           N      YD,MR0,MDR,ILIR,E,D
                  *
                  *
                  *
                  *
                  *
```

```
                                 ORG    '300'
                           * * * * * * * * * * * * * * * * * * * * * * * * *
                           *                                               *
                           *        S U B R O U T I N E   A T B L          *
                           *                                               *
                           * COMMON SUBROUTINE FOR ATL, ABL                *
                           *                                               *
                           * CALL IS:      BAL  ATBL(MR6)                  *
                           *                                               *
                           * * * * * * * * * * * * * * * * * * * * * * * * *
                           *
                           *
                           *
  300  239A 10C3  ATBL     AX     MAR,YX,MDR,ATBL1,C     TRANSFER IF RX1 OR RX3
  301  321F 1004           LI     MR0,4                  ADD 4 IF RX2
  302  2210 1E04           AX     MR0,MR0,MAR,ATBL2
  303  2A1F 1E00  ATBL1    L      MR0,MAR                MR0=LIST START ADRS
  304  2B9F 1800  ATBL2    L      MAR,MR0
  305  2B7F 1F8F           L      MDR,NULL,DR4           FETCH MAX SLOT/NO. USED
                           *
  306  2ABF 1D80           L      MR5,MDR
  307  3275 A010           RRI    MR3,MR5,16             NO. USED/MAX SLOT
  308  2293 0ACD           SX     MR4,MR3,MR5,LSTOVF,C
                           *
                           * LIST NOT FULL
                           *
  309  2B7F 3D87           AINC   MDR,NULL,MDR,DW4       INCREMENT NO.USED
  30A  3293 0001           SI     MR4,MR3,1              MAX SLOT - 1
  30B  2BFF 1F8D           L      NULL,NULL,I4DR4        FETCH CUR.TOP/NEXT BOTTOM
  30C  03F8 0600           BAL    (MR6)(NULL)            RETURN TO CALL
                           *
                           *
                           *
                           *
  30D  336D 5FF0  LSTOVF   NI     PSW,PSW,'FF0'          CLEAR CONDITION CODE
  30E  33BD 7004           OI     PSW,PSW,4              SET V FLAG
  30F  2BFF 1F99           L      NULL,NULL,ILIR,D
```

```
                   *                                                              83214370
                   * ADD TO TOP OF LIST                                           83214380
                   *                                                              83214390
 310  2A5F 1D80  ATL1   L      MR2,MDR                                            83214400
 311  3252 8010         SRLI   MR2,MR2,16            CURRENT TOP                  83214410
 312  2252 2FD4         SDECX  MR2,MR2,NULL,ATL2,C   DECREMENT BY 1               83214420
                   *                                 TRANSFER IF NO CARRY         83214430
                   *                                                              83214440
                   * LIST WRAP, SET CURRENT TOP TO MAX                            83214450
                   *                                                              83214460
 313  2A5F 1A00         L      MR2,MR4                                            83214470
 314  3652 5017  ATL2   NI     MR2,MR2,LOHALF,I      REMOVE HI ORDER BITS         83214480
 315  2B7F 1903         L      MDR,MR2,DW2           INSERT UP-DATED POINTER      83214490
                   *                                                              83214500
 316  2A52 1900  ADDIT  A      MR2,MR2,MR2           2X SLOT NUMBER               83214510
 317  2A52 1900         A      MR2,MR2,MR2           4X SLOT NUMBER               83214520
 318  3210 1008         AI     MR0,MR0,8             ADDRESS OF SLOT ZERO         83214530
 319  2B90 1900         A      MAR,MR0,MR2           PLUS 4X SLOT NUMBER          83214540
 31A  2B7F 1C87         L      MDR,YD,DW4            ADD ELEMENT TO LIST          83214550
 31B  2BFF 1FB9         L      NULL,NULL,ILIR,E,D    CLEAR CC & EXIT              83214560
                   *                                                              83214570
                   *                                                              83214580
                   *                                                              83214590
                   * ADD TO BOTTOM OF LIST                                        83214600
                   *                                                              83214610
 31C  2A5F 1D80  ABL1   L      MR2,MDR                                            83214620
 31D  3652 5017         NI     MR2,MR2,LOHALF,I      MR2=NEXT BOTTOM POINTER      83214630
 31E  2A3F 1900         L      MR1,MR2               SAVE ORIGINAL VALUE          83214640
 31F  3252 1001         AI     MR2,MR2,1             INCREMENT NEXT BOTTOM        83214650
 320  2A74 6880         X      MR3,MR4,MR1           COMPARE ORIGIONAL TO MAX     83214660
 321  3673 5017  ABL2   NI     MR3,MR3,LOHALF,I      TEST LS 16 BITS              83214670
 322  17E0 C900         BALNZ  ABL3(NULL)            NO WRAP                      83214680
                   *                                                              83214690
                   * LIST WRAP, SET NEXT BOTTOM TO ZERO                           83214700
                   *                                                              83214710
 323  325F 1000         LI     MR2,0                                              83214720
 324  3390 1006  ABL3   AI     MAR,MR0,6                                          83214730
 325  2B7F 1903         L      MDR,MR2,DW2           STORE UPDATED POINTER        83214740
 326  225F 1896         LX     MR2,MR1,ADDIT         USE ORIGIONAL VALUE          83214750
```

```
                        *                                                                  83214770
                        * BIT TABLE USED BY TBT, SBT, RBT, & CBT                           83214780
                        *                                                                  83214790
                  0327  BTABLE EQU    *                                                    83214800
    327  0000 8000  BIT16  DC     '00008000'                                               83214810
    328  0000 4000  BIT17  DC     '00004000'                                               83214820
    329  0000 2000  BIT18  DC     '00002000'                                               83214830
    32A  0000 1000  BIT19  DC     '00001000'                                               83214840
    32B  0000 0800  BIT20  DC     '00000800'                                               83214850
    32C  0000 0400  BIT21  DC     '00000400'                                               83214860
    32D  0000 0200  BIT22  DC     '00000200'                                               83214870
    32E  0000 0100  BIT23  DC     '00000100'                                               83214880
    32F  0000 0080  BIT24  DC     '00000080'                                               83214890
    330  0000 0040  BIT25  DC     '00000040'                                               83214900
    331  0000 0020  BIT26  DC     '00000020'                                               83214910
    332  0000 0010  BIT27  DC     '00000010'                                               83214920
    333  0000 0008  BIT28  DC     '00000008'                                               83214930
    334  0000 0004  BIT29  DC     '00000004'                                               83214940
    335  0000 0002  BIT30  DC     '00000002'                                               83214950
    336  0000 0001  BIT31  DC     '00000001'                                               83214960
```

```
                   * * * * * * * * * * * * * * * * * * * * * * * * *            83214980
                   *                                               *            83214990
                   *          S U B R O U T I N E   R T B L        *            83215000
                   *                                               *            83215010
                   * COMMON SUBROUTINE FOR RTL,RBL                 *            83215020
                   *                                               *            83215030
                   * CALL IS:      BAL   RTBL(MR7)                 *            83215040
                   *                                               *            83215050
                   * * * * * * * * * * * * * * * * * * * * * * * * *            83215060
                   *                                                            83215070
                   *                                                            83215080
                   *                                                            83215090
 337  239A 1DFA  RTBL    AX    MAR,YX,MDR,RTBL1,C    TRANSFER IF RX1 OR RX3     83215100
 338  321F 1004          LI    MR0,4                 ADD 4 IF RX2               83215110
 339  2210 1E3B          AX    MR0,MR0,MAR,RTBL2                                83215120
 33A  2A1F 1E00  RTBL1   L     MR0,MAR               MR0=LIST START ADRS        83215130
 33B  2B9F 1800  RTBL2   L     MAR,MR0                                          83215140
 33C  2B7F 1F8F          L     MDR,NULL,DR4          FETCH MAX SLOT/NO. USED    83215150
 33D  2A7F 1D80          L     MR3,MDR                                          83215160
 33E  3273 A010          RRI   MR3,MR3,16            NO. USED/MAX SLOT          83215170
 33F  3693 000F          SI    MR4,MR3,BIT15,I                                  83215180
 340  13F0 C340          BALC  LSTOVF(NULL)          LIST IS EMPTY (P.32)       83215190
                   *                                                            83215200
                   * LIST NOT EMPTY                                             83215210
                   *                                                            83215220
 341  2A9F 1D80          L     MR4,MDR                                          83215230
 342  2B74 2F87          SDEC  MDR,MR4,NULL,DW4      DECREMENT NO.USED          83215240
 343  2A94 2FA0          SDEC  MR4,MR4,NULL,E        ENABLE CC UPDATE           83215250
 344  37F4 5017          NI    NULL,MR4,LOHALF,I     SET CC : G IF NOT EMPTY, ELSE ZERO 83215260
 345  2A93 2F8D          SDEC  MR4,MR3,NULL,14DR4    MAX SLOT - 1               83215270
 346  03F8 0800          BAL   (MR6)(NULL)           RETURN TO CALL             83215280
```

```
                    *                                                      83215300
                    *                                                      83215310
                    * REMOVE FROM TOP OF LIST                              83215320
                    *                                                      83215330
 347  2A5F 1080     RTL1   L     MR2,MDR                                   83215340
 348  3252 8010            SRLI  MR2,MR2,16          CURRENT TOP           83215350
 349  2A3F 1900            L     MR1,MR2             ORIGINAL VALUE        83215360
 34A  3252 1001            AI    MR2,MR2,1           INCREMENT BY 1        83215370
 34B  2A74 6880            X     MR3,MR4,MR1         COMPARE TO MAX        83215380
 34C  3673 5017            NI    MR3,MR3,LOHALF,I    TEST LS 16 BITS       83215390
 34D  17E0 D3C0            BALNZ RTL2(NULL)          NO WRAP               83215400
                    *                                                      83215410
                    * LIST WRAP, SET CURRENT TOP TO ZERO                   83215420
                    *                                                      83215430
 34E  325F 1000            LI    MR2,0                                     83215440
 34F  2B7F 1903     RTL2   L     MDR,MR2,DW2         INSERT UPDATED POINTER 83215450
 350  225F 1897            LX    MR2,MR1,REMOV1      USE ORIGIONAL VALUE   83215460
                    *                                                      83215470
                    *                                                      83215480
                    *                                                      83215490
                    * REMOVE FROM BOTTOM OF LIST                           83215500
                    *                                                      83215510
 351  2A5F 1080     RBL1   L     MR2,MDR                                   83215520
 352  3652 5017            NI    MR2,MR2,LOHALF,I    NEXT BOTTOM POINTER   83215530
 353  2252 2FD5            SDECX MR2,MR2,NULL,REMOV,C  DECREMENT, XFER IF NO WRAP 83215540
 354  3654 5017            NI    MR2,MR4,LOHALF,I    SET NEXT BOTTOM TO MAX 83215550
                    *                                                      83215560
 355  3390 1006     REMOV  AI    MAR,MR0,6                                 83215570
 356  2B7F 1903            L     MDR,MR2,DW2         RESTORE               83215580
 357  3252 9002     REMOV1 SLLI  MR2,MR2,2           4X SLOT NUMBER        83215590
 358  3210 1008            AI    MR0,MR0,8           ADDRESS OF SLOT 0     83215600
 359  2B90 190F            A     MAR,MR0,MR2,DR4     PLUS 4X SLOT NUMBER   83215610
 35A  2B3F 1099            L     YD,MDR,ILIR,D       LOAD ELEMENT          83215620
                    *                                                      83215630
                    *                                                      83215640
                    *                                                      83215650
```

```
                  * MULTIPLY HALFWORD                                               83215670
                  *                                                                 83215680
 35B  3658 7015   MHR1   OI     MR2,YS,HIHALF,I       SET MS 16 BITS                83215690
 35C  3652 6327          XI     MR2,MR2,BIT16,I       INVERT SIGN THEN              83215700
 35D  3772 1327          AI     MDR,MR2,BIT16,I       EXTEND IT                     83215710
 35E  3679 7015   MH1    OI     MR3,YD,HIHALF,I       SET MS 16 BITS                83215720
 35F  3673 6327          XI     MR3,MR3,BIT16,I       INVERT THE SIGN               83215730
 360  3673 1327          AI     MR3,MR3,BIT16,I       EXTEND IT                     83215740
 361  2A53 ED89          M      MR2,MR3,MDR,ILIR      DO MULTIPLY                   83215750
 362  2B3F 1990          L      YD,MR3,D              LOAD LS 32 BITS               83215760
                  *                                                                 83215770
                  *                                                                 83215780
                  *                                                                 83215790
                  *                                                                 83215800
                  * DIVIDE HALFWORD                                                 83215810
                  *                                                                 83215820
 363  3658 7015   DHR1   OI     MR2,YS,HIHALF,I       SET MS 16 BITS                83215830
 364  3652 6327          XI     MR2,MR2,BIT16,I       INVERT THE SIGN               83215840
 365  3772 1327          AI     MDR,MR2,BIT16,I       EXTEND IT                     83215850
                  *                                   MDR = DIVISOR                 83215860
 366  2A5F 1F80   DH1    L      MR2,NULL                                            83215870
 367  2A7F 1C80          L      MR3,YD                MR3=DIVIDEND                  83215880
 368  17E4 DA80          BALNL  *+2(NULL)                                           83215890
 369  325F 0001          SI     MR2,NULL,1            MR2=SIGN OF DIVIDEND          83215900
 36A  2A53 FD89          D      MR2,MR3,MDR,ILIR      DO DIVIDE                     83215910
 36B  13F4 B1C0          BALV   DFALT1(NULL)          ARITHMETIC FAULT (P.28)       83215920
 36C  3693 5015          NI     MR4,MR3,HIHALF,I      SAVE MS 16 BITS               83215930
 36D  37F3 5327          NI     NULL,MR3,BIT16,I      DIVIDE FAULT IF MS 16         83215940
 36E  13E0 DD00          BALZ   DH3(NULL)             QUOTIENT BITS DON'T EQUAL     83215950
 36F  37F4 6015          XI     NULL,MR4,HIHALF,I     BIT 16 OF THE QUOTIENT        83215960
 370  17E0 B1C0   DH2    BALNZ  DFALT1(NULL)                                        83215970
 371  2B3F 1900          L      YD,MR2                REMAINDER TO R1               83215980
 372  2BDF 3F00          AINC   YDI,NULL,YDI                                        83215990
 373  2B3F 1990          L      YD,MR3,D              QUOTIENT TO R1+1              83216000
                  *                                                                 83216010
 374  23FF 1A30   DH3    LX     NULL,MR4,DH2                                        83216020
```

```
                *                                                                     83216040
                * TRANSLATE                                                           83216050
                *                                                                     83216060
                *                                                                     83216070
375  32B9 50FF  TLATE1 NI    MR6,YD,'FF'           BYTE TO TRANSLATE                  83216080
376  2AD6 1800         A     MR6,MR6,MR6           2X THE BYTE PLUS ADRS              83216090
377  2B96 108B         A     MAR,MR6,MDR,DR2       OF TRANSLATION TABLE               83216100
                *                                  FETCH HALFWORD ENTRY               83216110
378  2ADF 1080         L     MR6,MDR                                                  83216120
379  17E4 E149         BALNL EXTLAT(NULL),ILIR     EXIT IF NOT NEGATIVE               83216130
37A  4B36 4FD0         STBR  YD,MR6,NULL,D         TRANSLATED BYTE TO R1              83216140
                *                                                                     83216150
                *                                                                     83216160
                *                                                                     83216170
                * PART OF AUTO DRIVER CHANNEL                                         83216180
                *                                                                     83216190
37B  3384 1010  TRANSL AI    MAR,R4,16             REG 4 = ADRS OF CCB                83216200
37C  2BFF 1F8F         L     NULL,NULL,DR4         FETCH ADRS OF TRANSLATION TABLE    83216210
37D  2AD3 1980         A     MR6,DAT,DAT           2X DATA BYTE                       83216220
37E  2B96 108B         A     MAR,MR6,MDR,DR2       FETCH HALFWORD ENTRY               83216230
37F  2ADF 1080         L     MR6,MDR                                                  83216240
380  17E4 E0C0         BALNL EXTRAN(NULL)          EXIT IF NOT NEGATIVE               83216250
381  3276 50FF         NI    DAT,MR6,'0FF'         MASK TRANSLATED BYTE               83216260
382  03F8 0B80         BAL   (MR7)(NULL)           RETURN TO CALL                     83216270
                *                                                                     83216280
                *                                                                     83216290
                *                                                                     83216300
383  287F 1980  EXTRAN L     R3,DAT                REG 3 = UN-TRANSLATED CHAR         83216310
384  33BD 5FF0         NI    PSW,PSW,'FF0'         CLEAR CC                           83216320
385  2B56 1800  EXTLAT A     LOC,MR6,MR6           2X TABLE ENTRY                     83216330
386  2BFF 1F9C         L     NULL,NULL,IR,D        IS SUBROUTINE ADDRESS              83216340
```

```
                   *                                                                83216360
                   * CONVERT FLOATING POINT TO FIXED POINT                           83216370
                   *                                                                83216380
                   *                                                                83216390
                   *                                  INTERRUPTS ARE ENABLED        83216400
                   *                                                                83216410
387   3230 B008  FXR1   RLI    MR1,MR0,8                                            83216420
388   3251 5F00         NI     MR2,MR1,'F00'          MR2 = FRACTION LEFT 8         83216430
389   3231 507F         NI     MR1,MR1,'07F'          MR1 = EXPONENT                83216440
38A   327F 1048         LI     MR3,'48'                                             83216450
38B   2233 08D2         SX     MR1,MR3,MR1,FXR4,C     COMPARE EXPONENT TO LIMIT     83216460
                   *                                  TRANSFER IF LESS THAN OR      83216470
                   *                                  EQUAL TO '48'. MR1            83216480
                   *                                  CONTAINS DIFFERENCE.          83216490
                   *                                                                83216500
38C   365F 113B  OVF1   LI     MR2,ONES,I             OVERFLOW WHEN ARGUMENT EQUAL TO 83216510
                   *                                  OR GREATER THAN '48800000'. SET 83216520
38D   327F 1004         LI     MR3,4                  RESULT TO MAX & QUEUE V FLAG  83216530
                   *                                                                83216540
                   *                                                                83216550
38E   2210 1850  FXR2   AX     MR0,MR0,MR0,FXR3,C     TEST RESULT SIGN              83216560
38F   2A5F 0900         S      MR2,NULL,MR2           2'S COMPLEMENT IF NEGATIVE    83216570
390   2B3F 1929  FXR3   L      YD,MR2,ILIR,E          LOAD REGISTER, SET CC         83216580
391   2BBD 7990         O      PSW,PSW,MR3,D          OR IN V FLAG QUEUE            83216590
                   *                                                                83216600
                   *                                                                83216610
392   33F1 0008  FXR4   SI     NULL,MR1,8             COMPARE COUNT TO LIMIT        83216620
393   13E4 E540         BALL   *+2(NULL)              ZERO RESULT IF EXPONENT WAS   83216630
394   225F 1F98         LX     MR2,NULL,FXR5          LESS THAN '41'. ELSE, FORM    83216640
395   3231 9002         SLLI   MR1,MR1,2              HEX SHIFT COUNT FROM DELTA    83216650
396   2A52 8880         SRL    MR2,MR2,MR1            SHIFT INTEGER 0:7 HEX PLACES  83216660
397   13E4 E300         BALL   OVF1(NULL)             BRANCH IF '48800000' OR OVER  83216670
398   227F 1F8E  FXR5   LX     MR3,NULL,FXR2          NO V FLAG QUEUE               83216680
```

```
                  *                                                               83216700
                  * CONVERT FIXED-POINT DATA TO FLOATING POINT                    83216710
                  *                                                               83216720
                  *                                                               83216730
                  *                                 INTERRUPTS ARE ENABLED        83216740
                  *                                                               83216750
 399  2A3F 1F80   FLR1   L      MR1,NULL            MR1 TO CONTAIN RESULT SIGN    83216760
 39A  23F0 185D          AX     NULL,MR0,MR0,FLR2,C TEST ARGUMENT SIGN            83216770
                  * ARGUMENT IS NEGATIVE                                          83216780
                  *                                                               83216800
 39B  2A1F 0800          S      MR0,NULL,MR0        2'S COMP                      83216810
 39C  323F 1080          LI     MR1,'80'            MR1 = RESULT SIGN             83216820
 39D  3231 7048   FLR2   OI     MR1,MR1,'48'        OR IN LARGEST POSSIBLE EXPONENT 83216830
 39E  325F 1004          LI     MR2,4                                             83216840
 39F  37F0 5131          NI     NULL,MR0,DIGIT1,I   TEST BITS 0:3                 83216850
 3A0  13E0 E880          BALZ   FLR3(NULL)                                        83216860
 3A1  2210 8925          SRLX   MR0,MR0,MR2,FLR4    SHIFT RIGHT 4                 83216870
 3A2  3231 0001   FLR3   SI     MR1,MR1,1           DECREMENT EXPONENT            83216880
 3A3  37F0 5133          NI     NULL,MR0,DIGIT2,I   TEST BITS 4:7                 83216890
 3A4  13E0 E980          BALZ   FLR5(NULL)                                        83216900
 3A5  2210 8927   FLR4   SRLX   MR0,MR0,MR2,FLR6    SHIFT RIGHT 4                 83216910
 3A6  3231 0001   FLR5   SI     MR1,MR1,1           DECREMENT EXPONENT            83216920
 3A7  3231 A008   FLR6   RRI    MR1,MR1,8           POSITION SIGN & EXPONENT      83216930
 3A8  2A10 78A0          O      MR0,MR0,MR1,E       COMBINE WITH FRACTION         83216940
                  *                                 MR0 = UNNORMALIZED ARGUMENT   83216950
 3A9  17FC 4540          BALD   LE2(NULL)           GO TO LE2 TO NORMALIZE (P.10) 83216960
                  *                                 INTERRUPTS ARE DISARMED       83216970
                  *                                                               83216980
                  *                                                               83216990
                  * SUBROUTINE USED BY LSU MICRO-CODE                             83217000
                  *                                                               83217010
 3AA  4A30 8FC0   READIT RDRA   MR1,MR0,NULL        INPUT MS BYTE                 83217020
 3AB  4A3F E8C0          EXB    MR1,MR1             LEFT 8                        83217030
 3AC  4A30 8880          RDA    MR1,MR0,MR1         INPUT LS BYTE                 83217040
 3AD  03F8 0800          BAL    (MR6)(NULL)                                       83217050
                  *                                                               83217060
                  *                                                               83217080
                  *                                                               83217090
                  *                                                               83217100
                  *                                                               83217110
                  *                                                                 83217130
                  * SIMULATE CHANNEL PROGRAM                                        83217140
                  *                                                                 83217150
 3AE  2B7F 1F8B  SCP1   L     MDR,NULL,DR2          FETCH CCW                      83217160
 3AF  32B0 1002         AI    MR5,TEMP,2            POINT TO BUFFER 0 BYTE COUNT   83217170
 3B0  2A5F 1D80         L     CCW,MDR                                              83217180
 3B1  33F2 5008         NI    NULL,CCW,BBIT         TEST BUFFER SWITCH BIT         83217190
 3B2  13E0 ED00         BALZ  SCP2(NULL)            USE BUFFER 0                   83217200
 3B3  32B0 100A         AI    MR5,TEMP,10           POINT TO BUFFER 1 BYTE COUNT   83217210
 3B4  2B9F 1A8B  SCP2   L     MAR,MR5,DR2           FETCH BUFFER BYTE COUNT        83217220
 3B5  2A7F 3D80         AINC  DAT,NULL,MDR          COUNT+1 TO "DAT"               83217230
 3B6  2A9F 1D80         L     COUNT,MDR             IF COUNT IS POSITIVE           83217240
 3B7  13E8 C340         BALG  LSTOVF(NULL)          SET V FLAG & EXIT (P.32)       83217250
 3B8  2B7F 19A3         L     MDR,DAT,DW2,E         STORE INCREMENTED COUNT, SET CC 83217260
 3B9  2BFF 1F80         L     NULL,NULL             DISABLE CC UPDATE              83217270
 3BA  3395 1002         AI    MAR,MR5,2                                            83217280
 3BB  2BFF 1F8F         L     NULL,NULL,DR4         FETCH BUFFER END ADDRESS       83217290
 3BC  2B94 1D8B         A     MAR,COUNT,MDR,DR2     ADD COUNT & FETCH HALFWORD     83217300
 3BD  33F2 5004         NI    NULL,CCW,RWBIT        TEST R/W BIT                   83217310
 3BE  17E0 F1C0         BALNZ WRTSC(NULL)           WRITE IF R/W = 1               83217320
                  *                                                                 83217330
                  * MOVE A BYTE TO THE BUFFER (READ)                                83217340
                  *                                                                 83217350
 3BF  4B79 CDC3         STB   MDR,YD,MDR,DW2                                       83217360
                  *                                                                 83217370
 3C0  2A7F 1980         L     DAT,DAT               TEST INCREMENTED COUNT         83217380
 3C1  13E8 F099         BALG  RWSC1(NULL),ILIR,D                                   83217390
                  *                                                                 83217400
                  * EXECUTE NEXT USER INSTRUCTION IF COUNT NOT POSITIVE             83217410
                  *                                                                 83217420
                  *                                                                 83217430
 3C2  33F2 5001  RWSC1  NI    NULL,CCW,FBIT         IF "FAST" MODE, DO             83217440
 3C3  13E0 F119         BALZ  RWSC2(NULL),ILIR,D    NEXT USER INSTRUCTION          83217450
                  *                                                                 83217460
 3C4  3372 6008  RWSC2  XI    MDR,CCW,BBIT          IF COUNT HAS GONE POSITIVE,    83217470
 3C5  2B9F 1803         L     MAR,TEMP,DW2          COMPLEMENT BUFFER SWITCH BIT   83217480
 3C6  2BFF 1F99         L     NULL,NULL,ILIR,D                                     83217490
                  *                                                                 83217500
                  *                                                                 83217510
 3C7  2A3F 1D80  WRTSC  L     DEV,MDR               MOVE A  BYTE FROM THE          83217520
 3C8  4B31 DDC0         LB    YD,DEV,MDR            BUFFER TO R1                   83217530
 3C9  2A7F 1980         L     DAT,DAT               TEST INCREMENTED COUNT         83217540
 3CA  13E8 F099         BALG  RWSC1(NULL),ILIR,D                                   83217550
                  *                                                                 83217560
                  * EXECUTE NEXT USER INSTRUCTION IF COUNT NOT POSITIVE             83217570
```

```
                       *                                                          83217590
                       * CYCLIC REDUNDANCY CHECK                                  83217600
                       *                                                          83217610
 3CB  32D9 503F  CRC12A NI    MR6,YD,'3F'          MASK 6 BITS                    83217620
 3CC  363F 1121         LI    MR1,C0F01,I          POLY CHECK                     83217630
 3CD  2AD6 6D80         X     MR6,MR6,MDR          XOR IN RESIDUAL                83217640
 3CE  36D6 5017         NI    MR6,MR6,LOHALF,1     MASK LS 16 BITS                83217650
 3CF  32FF 1001         LI    MR7,1                                               83217660
 3D0  12B8 F740         BAL   CRC12B(RETURN)                                      83217670
 3D1  2BFF 1F99         L     NULL,NULL,ILIR,D                                    83217680
                       *                                                          83217690
                       *                                                          83217700
 3D2  3279 50FF  CRC16A NI    DAT,YD,'FF'          LOAD BYTE                      83217710
 3D3  12B8 F540         BAL   CRC16B(RETURN)       TO COMMON ROUTINE              83217720
 3D4  2BFF 1F99         L     NULL,NULL,ILIR,D                                    83217730
                       *                                                          83217740
                       * SUBROUTINE SHARED BY AUTO DRIVER CHANNEL                 83217750
                       *                                                          83217760
 3D5  363F 1123  CRC16B LI    MR1,CA001,I          POLY CHECK                     83217770
 3D6  2AD3 6D80         X     MR6,DAT,MDR          XOR IN RESIDUAL                83217780
 3D7  36D6 5017         NI    MR6,MR6,LOHALF,I     MASK LS 16 BITS                83217790
 3D8  32FF 1001         LI    MR7,1                                               83217800
                       *                                                          83217810
                       *                                                          83217820
 3D9  22D6 8BDB         SRLX  MR6,MR6,MR7,*+2,C    DATA & RESIDUAL EQUAL?         83217830
 3DA  2AD6 6880         X     MR6,MR6,MR1          YES, XOR IN FEEDBACK           83217840
 3DB  22D6 8BDD         SRLX  MR6,MR6,MR7,*+2,C    DATA & RESIDUAL EQUAL?         83217850
 3DC  2AD6 6880         X     MR6,MR6,MR1          YES, XOR IN FEEDBACK           83217860
 3DD  22D6 8BDF  CRC12B SRLX  MR6,MR6,MR7,*+2,C    DATA & RESIDUAL EQUAL?         83217870
 3DE  2AD6 6880         X     MR6,MR6,MR1          YES, XOR IN FEEDBACK           83217880
 3DF  22D6 8BE1         SRLX  MR6,MR6,MR7,*+2,C    DATA & RESIDUAL EQUAL?         83217890
 3E0  2AD6 6880         X     MR6,MR6,MR1          YES, XOR IN FEEDBACK           83217900
 3E1  22D6 8BE3         SRLX  MR6,MR6,MR7,*+2,C    DATA & RESIDUAL EQUAL?         83217910
 3E2  2AD6 6880         X     MR6,MR6,MR1          YES, XOR IN FEEDBACK           83217920
 3E3  22D6 8BE5         SRLX  MR6,MR6,MR7,*+2,C    DATA & RESIDUAL EQUAL?         83217930
 3E4  2AD6 6880         X     MR6,MR6,MR1          YES, XOR IN FEEDBACK           83217940
 3E5  22D6 8BE7         SRLX  MR6,MR6,MR7,*+2,C    DATA & RESIDUAL EQUAL?         83217950
 3E6  2AD6 6880         X     MR6,MR6,MR1          YES, XOR IN FEEDBACK           83217960
 3E7  22D6 8BE9         SRLX  MR6,MR6,MR7,*+2,C    DATA & RESIDUAL EQUAL?         83217970
 3E8  2AD6 6880         X     MR6,MR6,MR1          YES, XOR IN FEEDBACK           83217980
                       *                                                          83217990
 3E9  17FC FA80         BALD  *+1(NULL)                                           83218000
 3EA  2B7F 1B03         L     MDR,MR6,DW2          STORE RESULT                   83218010
 3EB  03F8 0A80         BAL   (RETURN)(NULL)       RETURN                         83218020
                       *                                                          83218030
```

```
                      ORG    '400'                                            83218040
                  *                                                           83218050
   400  1379 2980 TLSU   BAL    DELAY(MDR)  4A6    2 MS DELAY (P.50)          83218060
LSU 401  4BF0 AFC0        SSRA   NULL,MR0,NULL      ADDRESS & SENSE STATUS     83218070
   402  13F5 04C0        BALV   POWRUP(NULL)       LSU NOT PRESENT IF FALSE SYNC  83218080
                  *                                                           83218090
                  *                                                           83218100
   403  339F 1001        LI     MAR,1                                         83218110
   404  12D8 EA80        BAL    READIT(MR6)        (P.40)                     83218120
   405  2BBF 1880        L      PSW,MR1            PSW 16:31                  83218130
   406  12D8 EA80        BAL    READIT(MR6)        (P.40)                     83218140
   407  2B5F 1880        L      LOC,MR1            LOC 16:31                  83218150
   408  12D8 EA80        BAL    READIT(MR6)        (P.40)                     83218160
   409  2A5F 1880        L      MR2,MR1            START ADDRESS              83218170
   40A  12D8 EA80        BAL    READIT(MR6)        (P.40)                     83218180
                  *                                MR1 = END ADDRESS          83218190
   40B  23F1 094D        SX     NULL,MR1,MR2,AUTO1,C  COMPARE START & END     83218200
   40C  13F8 9C00        BAL    IDLE(NULL)         IDLE IF START NOT LESS (P.24)  83218210
                  *                                THAN END ADDRESS           83218220
   40D  2B9F 190B AUTO1  L      MAR,MR2,DR2                                   83218230
   40E  4B7F 0D80        RD     MDR,MDR            INPUT DATA BYTE            83218240
   40F  13F4 9C00        BALV   IDLE(NULL)         IDLE IF BAD STATUS (P.24)  83218250
   410  2A52 3F83 AUTO2  AINC   MR2,MR2,NULL,DW2   INCREMENT START ADRS       83218260
   411  23F1 094D        SX     NULL,MR1,MR2,AUTO1,C  LOOP TILL REACH END ADDRESS  83218270
   412  13F8 8F40        BAL    TWAIT(NULL)        TEST NEW PSW(P.22)         83218280
                  *                                                           83218290
                  *                                                           83218300
                  * NO LSU, NORMAL POWER UP SEQUENCE                          83218310
                  *                                                           83218320
   413  339F 1084 POWRUP LI     MAR,'84'                                      83218330
   414  2BBF 1F8A        L      PSW,NULL,PR2       FETCH PSW SAVE POINTER     83218340
   415  369F 1017        LI     MR4,LOHALF,I       USE ONLY LS 16 BITS        83218350
   416  2A94 5D8F        N      MR4,MR4,MDR,DR4    FETCH REGISTER SAVE POINTER  83218360
Bombs → 417  367F 1017   LI     MR3,LOHALF,I                                  83218370
OK ─ 418  2A73 5D80      N      MR3,MR3,MDR        USE ONLY LS 16 BITS — A=ØØØØFFFF  83218380 B=FFFFBFB9  S=ØØØØBFB9"G" f1a
   419  3393 0004        SI     MAR,MR3,4 ~ DEC 514  TEMPORARY DECREMENT A=BFB9,B=4,  83218390 S=BFB5
   41A  323F 000F        SI     MR1,NULL,15        MR1 = 'FFFFFFF1' A=Ø, B=F, S=-2  83218400
   41B  2BDF 1F80        L      YDI,NULL           SELECT R0, SET 0           83218410 C8L
                  *                                                           83218420
   41C  13F1 07CD LLOOP  BALC   ENDSET(NULL),I4DR4  FETCH NEXT IF NOT DONE    83218430
   41D  2B3F 1D80        L      YD,MDR             LOAD REGISTER              83218440
   41E  23D1 1F1C        AX     YDI,MR1,YDI,LLOOP  BUMP R1 FIELD & LOOP       83218450
                  *                                                           83218460
                  * ONE REGISTER SET LOADED                                   83218470
                  *                                                           83218480
   41F  33BD 1010 ENDSET AI     PSW,PSW,'10'       INCREMENT REGISTER SET NUMBER  83218490
   420  4AFF 7F80        SMCR   MR7,NULL           TEST IF 2 OR 8 REGISTER SETS  83218493
   421  32F7 5040        NI     MR7,MR7,'40'       MCR BIT 9 = 0 IF 2 SETS    83218494
   422  17E1 0900        BALNZ  *+2(NULL)          MCR BIT 9 = 1 IF 8 SETS    83218495
   423  33BD 7070        OI     PSW,PSW,'70'       IF ZERO, FORCE LAST SET    83218496
   424  33FD 5080        NI     NULL,PSW,'80'      TEST IF LAST SET LOADED    83218500
   425  13E1 0700        BALZ   LLOOP(NULL)        LOOP UNTIL ALL SETS LOADED  83218510
   426  13F9 09C0        BAL    *+1(NULL)          FOR D FLOAT                83218520
                  *                                                           83218530
   427  2B9F 1F8F        L      MAR,NULL,DR4       FETCH FLOATING REGISTER 0  83218540
   428  323F 000E        SI     MR1,NULL,14        MR1='FFFFFFF2'             83218550
                  *                                                           83218560
```

```
 429  6B3F 1D8D  RESTRE AU    YD,NULL,MDR,I4DR4      LOAD, THEN FETCH NEXT            83218570
~~42A  23D1 1F69         AX    YDI,MR1,YDI,RESTRE,C~~                                   83218580
                  *                                                                    83218590
 42B  2B9F 1A0F         L     MAR,MR4,DR4            FETCH SAVED PSW                  83218600
*42C  2A3F 1D8D         L     MR1,MDR,I4DR4          FETCH SAVED LOC                  83218610
 42D  339F 1028         LI    MAR,'28'                                                83218620
 42E  2B5F 1D8A         L     LOC,MDR,PR2            LOAD LOC, FETCH                  83218630
                  *                                  SAVED CONSOLE STATUS             83218640
 42F  2BBF 1880         L     PSW,MR1                LOAD PSW                         83218650
 430  323F 1001         LI    MR1,1                                                   83218660
 431  4BF1 AFC0         SSRA  NULL,MR1,NULL          ADDRESS THE DISPLAY              83218670
 432  2A3F 1D80         L     MR1,MDR                                                 83218680
 433  33F1 50E0         NI    NULL,MR1,'0E0'         TEST STATUS BITS 1,2,3           83218690
 434  17E0 9980         BALNZ LOCDIS(NULL)           SHOW LOC, GO TO IDLE (P.24)      83218700
                  *                                                                    83218710
                  *                                                                    83218720
                  *                                                                    83218730
                  *                                                                    83218740
                  *                                                                    83218750
                  *                                                                    83218760
 435  37FD 5329         NI    NULL,PSW,BIT18,I       TEST MALF ENABLE                 83218770
 436  17E0 8C00         BALNZ MMFINT(NULL)           MACHINE MALFUNCTION ~~(P.24)~~ P,22 83218780
                  *                                  INTERRUPT IF ENABLED             83218790
 437  13F8 8F40         BAL   TWAIT(NULL)            TEST PSW WAIT BIT (P.22)         83218800
                  *                                                                    83218810
                  *                                                                    83218820
```

```
                    *A U T O   D R I V E R   C H A N N E L                                     83218840
                    *                                                                          83218850
                    *                                                                          83218860
                    * CCW BIT DESIGNATIONS                                                     83218870
                    *                                                                          83218880
          0080 EBIT    EQU  '80'                 EXECUTE        TEMP    = MR0                 83218890
          0010 CBIT    EQU  '10'                 CHECK TYPE     DEV     = MR1                 83218900
          0008 BBIT    EQU  '08'                 BUFFER SWITCH  CCW     = MR2                 83218910
          0004 RWBIT   EQU  '04'                 READ/WRITE     DAT     = MR3                 83218920
          0002 TBIT    EQU  '02'                 TRANSLATE      COUNT   = MR4                 83218930
          0001 FBIT    EQU  '01'                 FAST MODE      RETURN  = MR5                 83218940
                    *                                                                          83218950
                    *                                                                          83218960
                    *                                                                          83218970
 438  2B9F 1D0B CHANEL L     MAR,LOC,DR2         FETCH CHANNEL COMMAND WORD                    83218980
 439  289F 1D00        L     R4,LUC              REG 4 = ADRS FORCED EVEN OF CCB               83218990
 43A  2A5F 1DA0        L     CCW,MDR,E                                                         83219000
 43B  33F2 5080        NI    NULL,CCW,EBIT       TEST THE EXECUTE BIT                          83219010
 43C  13E1 1400        BALZ  EXSUB1(NULL)        NO EXECUTE, CC=0 (P.46)                       83219020
                    *                            IF FALL THRU, E=1 & CC=0010                   83219030
                    *                                                                          83219040
 43D  4A7F E940        EXB   DAT,CCW             ISOLATE STATUS MASK                           83219050
 43E  2BE3 5980        N     NULL,R3,DAT         TEST DEVICE STATUS AGAINST MASK               83219060
 43F  17E1 1540        BALNZ EXSUB2(NULL)        BAD STATUS (P.46)                             83219070
                    *                                                                          83219080
 440  33F2 5001        NI    NULL,CCW,FBIT       TEST IF FAST MODE                             83219090
 441  13E1 1940        BALZ  NFAST(NULL)         NOT FAST MODE (P.47)                          83219100
                    *                                                                          83219110
                    *                                                                          83219120
                    *F A S T   M O D E                                                         83219130
                    *                                                                          83219140
 442  3204 1002        AI    TEMP,R4,2           ADRS BUFFER 0 BYTE COUNT                      83219150
 443  2B9F 180B        L     MAR,TEMP,DR2        FETCH IT                                      83219160
 444  3390 1002        AI    MAR,TEMP,2                                                        83219170
 445  2A9F 1D80        L     COUNT,MDR                                                         83219180
 446  13E9 16CF        BALG  EXAUTO(NULL),DR4    EXIT, COUNT POSITIVE (P.46)                   83219190
 447  2B94 1D8B        A     MAR,COUNT,MDR,DR2   BUFFER END ADDRESS + COUNT                    83219200
                    *                                                                          83219210
                    * BUFFER 0 BYTE COUNT IN REGISTER "COUNT"                                  83219220
                    * ADDRESS OF BUFFER 0 BYTE COUNT IN "TEMP"                                 83219230
                    * BUFFER 0 END ADRS + BYTE COUNT IN "MAR"                                  83219240
                    * BYTE/HALFWORD TO TRANSFER IN "MDR"                                       83219250
                    *                                                                          83219260
                    *                                                                          83219270
 448  43FF EFD7        THWX  NULL,NULL,BYTEIO,C  TEST HW LINE (P.46)                           83219280
                    *                                                                          83219290
                    * FALL THROUGH IF LINE IS ACTIVE                                           83219300
                    *                                                                          83219310
                    *                                                                          83219320
 449  33F2 5004        NI    NULL,CCW,RWBIT      TEST R/W BIT                                  83219330
 44A  17E1 18C0        BALNZ HWRT1(NULL)         WRITE HW, RW=1 (P.46)                         83219340
                    *                                                                          83219350
 44B  4B7F 4F83        RH    MDR,NULL,DW2        READ HALFWORD                                 83219360
 44C  3294 1002 HRDWT  AI    COUNT,COUNT,2       INCREMENT OF BUFFER 0 BYTE COUNT              83219370
 44D  2B9F 1800 COMMON L     MAR,TEMP            RE-ADDRESS BUFFER BYTE COUNT                  83219380
 44E  2B7F 1A03        L     MDR,COUNT,DW2                                                     83219390
 44F  17E9 16C0        BALNG EXAUTO(NULL)        EXIT IF NOT POSITIVE (P.46)                   83219400
```

```
                    *                                                                  83219410
                    * EXIT TO SUBROUTINE IF COUNT HAS BECOME POSITIVE                  83219420
                    *                                                                  83219430
                    *                                                                  83219440
                    *                                                                  83219450
 450  3384 1014  EXSUB1 AI     MAR,R4,20              REG 4 = ADRS OF CCB              83219460
 451  2BFF 1F8B         L      NULL,NULL,DR2          FETCH SUBROUTINE ADDRESS         83219470
 452  361F 1039         LI     TEMP,CFFFE,I           TEMP='0000FFFE'                  83219480
 453  2B50 5D80         N      LOC,TEMP,MDR                                            83219490
 454  2BFF 1F9C         L      NULL,NULL,IR,D         FETCH USER INSTRUCTION           83219500
                    *                                                                  83219510
                    *                                                                  83219520
 455  33BD 5FF0  EXSUB2 NI     PSW,PSW,'FF0'          CLEAR CC THEN                    83219530
 456  23BD 3F90         AINCX  PSW,PSW,NULL,EXSUB1    SET L FLAG (BAD STATUS)          83219540
                    *                                                                  83219550
                    *                                                                  83219560
                    *                                                                  83219570
 457  33F2 5004  BYTEIO NI     NULL,CCW,RWBIT         TEST R/W BIT                     83219580
 458  17E1 1900         BALNZ  FWRIT(NULL)            WRITE BYTE IF RW = 1             83219590
                    *                                                                  83219600
 459  4B7F 0D83         RD     MDR,MDR,DW2            READ BYTE                        83219610
                    *                                                                  83219620
                    *                                                                  83219630
 45A  2294 3F8D  FRDWT  AINCX  COUNT,COUNT,NULL,COMMON INC BUFF 0 BYTE COUNT (P.45)    83219640
                    *                                                                  83219650
                    *                                                                  83219660
                    *                                                                  83219670
 45B  2B5F 1080  EXAUTO L      LOC,R1                 RESTORE LOC                      83219680
 45C  2B8F 1000         L      PSW,R0                 RESTORE PSW                      83219690
 45D  321F 1008         LI     MR0,8                  QUEUE C FLAG                     83219700
 45E  4AFF 7F8C         SMCR   MR7,NULL,IR            SENSE MACHINE CONTROL REG.       83219710
 45F  33F7 5007         NI     NULL,MR7,7             TEST FOR EPF OR PARITY ERR       83219715
 460  17E0 8C40         BALNZ  MMF1(NULL)             DO MALF INTERRUPT (P.22)         83219720
 461  37FD 5327         NI     NULL,PSW,BIT16,I       TEST WAIT BIT                    83219730
 462  17E0 ABD0         BALNZ  WAIT(NULL),D           TO WAIT IF SET (P.27)            83219740
                    *                                                                  83219750
                    *                                                                  83219760
                    *                                                                  83219770
 463  43FF 5D8C  HWRT1  WHX    NULL,MDR,HRDWT         WRITE HALFWORD (P.45)            83219780
                    *                                                                  83219790
                    *                                                                  83219800
 464  43FF 1D9A  FWRIT  WDX    NULL,MDR,FRDWT         WRITE A BYTE                     83219810
```

```
                            *                                                          83219830
                            * N O R M A L   M O D E                                    83219840
                            *                                                          83219850
 465   321F 1002   NFAST   LI     TEMP,2                SET UP FOR BUFFER 0            83219860
 466   33F2 5008           NI     NULL,CCW,BBIT         TEST BUFFER SWITCH BIT         83219870
 467   13E1 1A40           BALZ   NFAST1(NULL)                                         83219880
 468   321F 100A           LI     TEMP,10               SET UP FOR BUFFER 1            83219890
                            *                                                          83219900
 469   2B84 180B   NFAST1  A      MAR,R4,TEMP,DR2       FETCH BUFFER BYTE COUNT        83219910
 46A   2A04 1800           A      TEMP,R4,TEMP          TEMP = ADRS OF BUFFER BYTE COUNT 83219920
 46B   3390 1002           AI     MAR,TEMP,2                                           83219930
 46C   2A9F 108F           L      COUNT,MDR,DR4         FETCH BUFFER END ADDRESS       83219940
 46D   13E9 16C0           BALG   EXAUTO(NULL)          EXIT IF POSITIVE (P.46)        83219950
 46E   2B54 1D80           A      LOC,COUNT,MDR         BUFFER END ADRS + COUNT        83219960
 46F   2B9F 1D0B           L      MAR,LOC,DR2                                          83219970
                            *                                                          83219980
                            * BUFFER BYTE COUNT IN REGISTER "COUNT"                    83219990
                            * ADDRESS OF BUFFER BYTE COUNT IN "TEMP"                   83220000
                            * BUFFER END ADRS + BYTE COUNT IN "MAR"                    83220010
                            * BYTE/HALFWORD TO TRANSFER IN "MDR"                       83220020
                            *                                                          83220030
                            * NOTE: IN NON-FAST MODE, ONLY BYTE TRANSFERS ARE ALLOWED  83220040
                            *                                                          83220050
                            *                                                          83220060
 470   33F2 5004           NI     NULL,CCW,RWBIT        TEST R/W BIT                   83220070
 471   17E1 1E40           BALNZ NFWRIT(NULL)           WRITE A BYTE IF R/W = 1        83220080
                            *                                                          83220090
 472   4A7F 0FC0           RDR    DAT,NULL              INPUT THE BYTE                 83220100
 473   2ABF 1980           L      RETURN,DAT            SAVE IT                        83220110
 474   33F2 5002           NI     NULL,CCW,TBIT         TRANSLATION REQUIRED ?         83220120
 475   16E0 DEC0           BALNZ TRANSL(MR7)            DO IT (P.38)                   83220130
                            *                                                          83220140
 476   2B9F 1D0B           L      MAR,LOC,DR2           RE-FETCH HALFWORD              83220150
 477   4B73 CDC3           STB    MDR,DAT,MDR,DW2       INSERT BYTE                    83220160
 478   227F 1ABE           LX     DAT,RETURN,REDCHK     DAT = UNTRANSLATED BYTE        83220170
                            *                                                          83220180
                            *                                                          83220190
                            *                                                          83220200
 479   2A7F 1D80   NFWRIT  L      DAT,MDR                                              83220210
 47A   4A73 DDC0           LB     DAT,DAT,MDR           BYTE TO OUTPUT                 83220220
 47B   33F2 5002           NI     NULL,CCW,TBIT         TEST TRANSLATE BIT             83220230
 47C   16E0 DEC0           BALNZ TRANSL(MR7)            DO IT (P.38)                   83220240
                            *                                                          83220250
                            * TRANSLATION NOT REQUIRED                                 33220260
                            *                                                          83220270
 47D   4BFF 19C0           WDR    NULL,DAT              OUTPUT THE BYTE                33220280
                            *                                                          83220290
                            * ONLY THE BYTE ACTUALLY TRANSFERRED IS INCLUDED IN THE    83220300
                            * LRC OR CRC.  SPECIAL CHARACTERS ARE NOT INCLUDED.        83220305
                            *                                                          83220310
 47E   3384 1008   REDCHK  AI     MAR,R4,8                                             83220320
 47F   2BFF 1F8B           L      NULL,NULL,DR2         FETCH CHECK-WORD               83220330
 480   33F2 5010           NI     NULL,CCW,CBIT         CHECK TYPE BIT                 83220340
 481   13E1 2240           BALZ   LRC(NULL)             LONGITUDINAL CHECKSUM (P.48)   83220350
                            *                                                          83220360
 482   12B8 F540           BAL    CRC16B(RETURN)        CYCLIC REDUNDANCY CHECK (P.42) 83220370
                            *                                                          83220380
```

```
                  *                                                                 83220390
                  *                                                                 83220400
 483  2B9F 1800  RTNCRC L     MAR,TEMP                                              83220410
 484  2B74 3F83         AINC  MDR,COUNT,NULL,DW2    INCREMENT & STORE COUNT         83220420
 485  17E9 16C0         BALNG EXAUTO(NULL)          EXIT IF NOT POSITIVE (P.46)     83220430
                  *                                                                 83220440
 486  3372 6008         XI    MDR,CCW,BBIT          COMPLEMENT BUFFER SWITCH BIT    83220450
 487  2B9F 1203         L     MAR,R4,DW2            RESTORE CCW                     83220460
 488  13F9 1400         BAL   EXSUB1(NULL)          EXIT TO SUBROUTINE (P.46)       83220470
                  *                                                                 83220480
                  *                                                                 83220490
                  * LONGITUDINAL CHECKSUM                                           83220500
                  *                                                                 83220510
 489  2B73 6D83  LRC    X     MDR,DAT,MDR,DW2       EXCLUSIVE OR CHECKSUM           83220520
 48A  13F9 20C0         BAL   RTNCRC(NULL)                                          83220530
                  *                                                                 83220540
```

```
                    *                                                                   83220560
                    * WRITEABLE CONTROL STORE INSTRUCTIONS                              83220570
                    *                                                                   83220580
                    *                                                                   83220590
                    *                                                                   83220600
                    * ENTER CONTROL STORE                                               83220610
                    *                                                                   83220620
  48B  32FF 1800    ECS1    LI     MR7,'800'            SELECT THE FIRST DCS            83220630
                    *                                   MODULE AT ADRS '800'            83220640
  48C  2AF7 1F00            A      MR7,MR7,YDI          ADD R1 FIELD                    83220650
  48D  03F8 0880            BAL    (MR7)(NULL)          BRANCH TO ONE FIRST             83220660
                    *                                   SIXTEEN LOCATIONS IN DCS        83220670
                    *                                                                   83220680
                    *                                                                   83220690
                    *                                                                   83220700
                    * READ/WRITE CONTROL STORE                                          83220710
                    *                                                                   83220720
  48E  2BDF 3F00    WDCS    AINC   YDI,NULL,YDI         BUMP R1 FIELD                   83220730
  48F  2B9F 180F    WDCS1   L      MAR,MR0,DR4          FETCH FULLWORD                  83220740
  490  3210 0004            SI     MR0,MR0,4            DECREMENT MEMORY ADDRESS        83220750
  491  2A5F 1D80            L      MR2,MDR              COPY DATA TO MR2                83220760
  492  3FF2 0880            STR    MR2,MR1              AND STORE IN DCS                83220770
  493  2A31 2F80            SDEC   MR1,MR1,NULL         DECREMENT DCS ADDRESS           83220780
  494  2339 2FCF            SDECX  YD,YD,NULL,WDCS1,C   DECREMENT COUNT                 83220790
  495  2BFF 1F99            L      NULL,NULL,ILIR,D     EXIT IF DONE                    83220800
                    *                                                                   83220810
                    *                                                                   83220820
  496  2BDF 3F00    RDCS    AINC   YDI,NULL,YDI         BUMP R1 FIELD                   83220830
  497  2B9F 1800    RDCS1   L      MAR,MR0                                              83220840
  498  2F7F 1887            L      MDR,MR1,I,DW4        MOVE DCS DATA TO MAIN MEMORY    83220850
  499  3210 0004            SI     MR0,MR0,4            DECREMENT MEMORY ADDRESS        83220860
  49A  2A31 2F80            SDEC   MR1,MR1,NULL         DECREMENT DCS ADDRESS           83220870
  49B  2339 2FD7            SDECX  YD,YD,NULL,RDCS1,C   DECREMENT COUNT                 83220880
  49C  2BFF 1F99            L      NULL,NULL,ILIR,D                                     83220890
                    *                                                                   83220900
                    *                                                                   83220910
                    * (R1)=DCS ADDRESS,(R1+1)=COUNT, (R2)=MAIN MEMORY ADRS              83220920
                    *                                                                   83220930
                    *                                                                   83220940
  49D  2A1B 1C00    CCS1    A      MR0,YDP1,YS          MR0=COUNT PLUS MEMORY ADDRESS   83220950
  49E  321B 9002            SLLI   MR0,YDP1,2           MR0=4X COUNT         R02        83220960
  49F  2A10 1C00            A      MR0,MR0,YS           PLUS MEMORY ADDRESS       R02   83220970
  4A0  2A3B 1C80            A      MR1,YDP1,YD          MR1=COUNT PLUS DCS ADDRESS      83220980
  4A1  2A5F 1E00            A      MR2,NULL,YDI         TEST R1 FIELD                   83220990
  4A2  13E1 2380            BALZ   WDCS(NULL)           0 = WRITE DCS                   83221000
  4A3  33F2 6002            XI     NULL,MR2,2                                           83221010
  4A4  13E1 2580            BALZ   RDCS(NULL)           2 = READ DCS                    83221020
  4A5  17FC 8240            BALD   ILEGAL(NULL)         ILLEGAL FUNCTION                83221030
                    *                                                                   83221040
                    *                                                       83221060
                    * TWO MILLISECOND DELAY FOR SCLRO TO BE RELEASED ON LMI 83221070
                    *                                                       83221080
                    *                                                       83221090
4A6  321F 1208  DELAY  LI    MR0,'208'                                     83221100
4A7  3210 9004         SLLI  MR0,MR0,4             MR0 = 8320              83221110
4A8  2210 2FE8         SDECX MR0,MR0,NULL,*,C      SINGLE INSTRUCTION LOOP 83221120
4A9  321F 1005         LI    MR0,5                 LOAD UP LSU DEVICE NUMBER 83221130
4AA  13F9 0040         BAL   TLSU+1(NULL)          (P.43)                  83221140
                    *                                                       83221150
4AB                    END                                                  83221160
```

```
  NO ERRORS

 A        00B4
 ABL      00CA
 ABL1     031C     8
 ABL2     0321
 ABL3     0324    33
 AD       00F4
 ADDIT    0316    33
 ADR      0074
 ADRMW    0253    23
 ADRS     0261    23
 AE       00D4
 AE1      0118     8
 AE2      0137    10
 AER      0054
 AER1     0064     4
 AER2     0066     4
 AFAULT   02CD    28
 AH       0094
 AHI      0194
 AHM      00C2
 AI       01F4
 AIS      004C
 AL       01AA
 AL1      0156    16
 AL2      015F    13,    13
 AM       00A2
 AR       0014
 ATBL     0300     8,     8
 ATBL1    0303    32
 ATBL2    0304    32
 ATL      00C8
 ATL1     0310     8
 ATL2     0314    33
 AUTO1    040D    43,    43
 AUTO2    0410
 AUTOIO   029F    26
 BAL      0082
 BAL1     00A5     6
 BAL2     0085     7
 BALR     0002
 BALR1    0005     2
 BBIT     0008    41,    41,    47,    48
 BBS      0041     4,     4
 BBS1     0043     4
 BC1      0034     6,     6,    15,    15
 BC2      0037     2,     2
 BDCS     01CA
 BFBS     0044
 BFC      0086
 BFCR     0006
 BFFS     0046
 BFS      0045     4,     4
 BFS1     0047     4
```

```
BI1720   0021      25
BI1620   0053      26
BIT15    000F       3,    35
BIT16    0327       3,     3,    22,    37,    37,    37,    37,    37,
                   37,    37,    46
BIT160   0023      23
BIT17    0328
BIT18    0329      44
BIT19    032A      28
BIT20    032B
BIT21    032C
BIT22    032D
BIT23    032E
BIT24    032F
BIT25    0330
BIT26    0331
BIT27    0332
BIT28    0333
BIT29    0334
BIT30    0335
BIT31    0336
BRR      0007       2,     2
BRW2     014E      12,    12
BRW3     014F      12
BRW4     0153      13
BTABLE   0327      30
BTBS     0040
BTC      0084
BTCR     0004
BTFS     0042
BXH      0180
BXLL     0182
BXLH     01E0      15,    15
BXLH1    01C9      18
BXLH2    01D1      17
BXLH3    01D3      17
BXLH4    01F8      17
BYTEIO   0457      45
C        00B2
C0F01    0121      42
C1       0013       6,     7,    15,    18
C2       003C       2
CA001    0123      42
CADRS    006C       9,    16,    16,    17,    17,    17
CADRS1   006E
CADRS2   0049       5
CADRS3   006F       4
CBIT     0010      47
CBT      00EE
CBT1     00EF       9
CBT2     00C5       9,     9
CCS      01D0
CCS1     049D      17
CCW      0012      41,    41,    41,    41,    41,    45,    45,    45,
                   45,    45,    46,    47,    47,    47,    47,    47,
                   48
CD       00F2
```

```
CDR      0072
CE       00D2
CER      0052
CFFFE    0039     24,    26,    29,    46
CH       0092
CHANEL   0438     26
CHI      0192
CHVR     0024
CHVR1    002C      3
CI       01F2
CL       00AA
CL1      00AB      3
CLB      01A8
CLB1     0172     16
CLH      008A
CLHI     018A
CLI      01EA
CLR      000A
CLRWT    0250     23,    24,    25
CMNBRW   014A     12
CMSTML   0111     10
COMBIT   02E5      9,     9,     9,     9
COMBT1   02E8     30
COMBT2   02E9     30
COMIN0   0208     28
COMIN1   020D
COMIN2   020F     27
COMINT   020A     19
COMLM    0102     10,    10
COMLML   0101     10
COMMON   044D     46
COMSTM   0110     10,    10
CONSER   0240     20,    24
COUNT    0014     41,    41,    45,    45,    45,    45,    45,    46,
                  46,    47,    47,    48
CR       0012
CRC12    00BC
CRC12A   03CB      7
CRC12B   03D0     42
CRC16    00BE
CRC16A   03D2      7
CRC16B   03D5     42,    47
D        00BA
D0       009C      7
DAT      0013     26,    26,    26,    26,    26,    38,    38,    38,
                  38,    41,    41,    41,    41,    41,    41,    42,
                  42,    45,    45,    47,    47,    47,    47,    47,
                  47,    47,    47,    48
DD       00FA
DDR      007A
DE       00DA
DELAY    04A6     43
DER      005A
DER1     001B      4
DER2     00DD      2
DEV      0011     17,    25,    26,    26,    26,    26,    26,    26,
                  26,    26,    26,    41,    41
```

```
DEZ      001C      8
DFALT0   02C4      2,     6
DFALT1   02C7     37,    37
DH       009A
DH1      0366      6
DH2      0370     37
DH3      0374     37
DHR      001A
DHR1     0363      2
DIGIT1   0131     40
DIGIT2   0133     40
DISMEM   0257     23
DOSBR    004D      4,     4
DOSBR1   004F      4
DR       003A
DR1      000B      3
DR2      000D      2
EBIT     0080     45
ECS      0102
ECS1     0486     17
ENDBRK   0155     12,    13
ENDSET   041F     43
EPSR     012A
EPSR1    0127     11
EXAUTO   045E     45,    45,    47,    48
EXBR     0126
EXBR1    016D     11
EXHR     0068
EXLSTM   0113     10
EXSUB1   0450     45,    46,    48
EXSUB2   0455     45
EXTLAT   0385     38
EXTRAN   0383     38
FBIT     0001     41,    45
FFALT1   02CC      2,    28
FFAULT   02C5      2,     4,     5,     5,     5,     5,     8,    10,
                  10,    10,    11,    11,    13
FLDR     007E
FLR      005E
FLR1     0399      4
FLR2     039D     40
FLR3     03A2     40
FLR4     03A5     40
FLR5     03A6     40
FLR6     03A7     40
FN       0286     24
FN0      0295     25
FN01     028F     25
FNDIS    0279     23
FRDWT    045A     46
FWRIT    0464     46
FXDR     007C
FXR      005C
FXR1     0387      4
FXR2     038E     39
FXR3     0390     39
FXR4     0392     39
```

```
FXR5     0398     39
HIHALF   0015      3,     11,     11,     25,     37,     37,     37,     37,
                  37
HRDWT    044C     46
HWRT1    0463     45
IDLE     0270     25,     43,     43
IDLE1    0272     24
ILEGAL   0209     20,     49
IOINT0   0299     20
IOINT1   029A     20
IOINT2   029C     20
IOINT3   029E     20
IOINTX   02A0     25,     26,     26,     29,     29
L        00B0
LA       01CC
LA1      002E     17
LB       01A6
LB1      016F     16
LBR      0126
LCS      004A
LD       00F0
LDR      0070
LE       00D0
LE1      0114      8
LE2      0115     40
LER      0050
LER1     0062      4
LEVEL    0012     25,     26,     26,     26,     26,     26,     26,     26,
                  26,     26,     26,     29,     29
LH       0090
LHI      0190
LHL      00E6
LHL1     02FD      9
LI       01F0
LIS      0048
LLOOP    041C     43,     43
LM       01A2
LM1      010A     16
LMD      00FE
LME      00E4
LME1     010D      9
LOCDIS   0266     25,     44
LOHALF   0017     21,     21,     23,     31,     33,     33,     33,     35,
                  36,     36,     36,     42,     42,     43,     43
LPSW     0184
LPSW1    02B2     15
LPSWR    0030
LR       0010
LRC      0489     47
LSTOVF   030D     32,     35,     41
M        00B8
MACINT   01FE     20
MD       00F8
MDR      0078
ME       00D8
ME1      0167      8
MER      0058
```

```
MH        0098
MH1       035E      6
MHR       0018
MHR1      035B      2
MMF1      0231     46
MMFINT    0230     20,    44
MR        0038
N         00A8
NFAST     0465     45
NFAST1    0469     47
NFWRIT    0479     47
NH        0088
NHI       0188
NI        01E8
NR        0008
O         00AC
OC        01BC
OCR       013C
OH        008C
OHI       018C
OI        01EC
ONES      013B     28,    39
OR        000C
OUTDIS    0268     23,    24,    25,    25
OVF1      038C     39
POWRUP    0413     43
PPFINT    0213     20,    24
PSWDIS    028C     25
QUEINT    02B8
R0        0000     26,    46
R1        0001     26,    46
R13       000D     27,    29
R14       000E     21,    29
R15       000F     21,    29
R2        0002     26
R3        0003     26,    38,    45
R4        0004     38,    45,    45,    46,    47,    47,    47,    48
RB        01AE
RB1       0145     16
RB2       0166     12
RBL       00CE
RBL1      0351      8
RBR       012E
RBT       00EC
RD        01B6
RDCS      0496     49
RDCS1     0497     49
RDFULL    02F7      7,     9,    15,    16,    16,    16
RDHALF    02F1      7,     7,     8,    16,    16,    16,    16,    16,
                   16,    16,    16
RDR       0136
READIT    03AA     43,    43,    43,    43
REDCHK    047E     47
REMOV     0355     36
REMOV1    0357     36
RESTRE    0429     44
RETURN    0015     42,    42,    42,    47,    47,    47
```

```
RFULL1  02FA     31
RFULL2  02FB     31
RH      01B2
RHALF1  02F4     31
RHALF2  02F5     31
RHR     0132
RLL     01D6
RRL     01D4
RTBL    0337      8,     8
RTBL1   033A     35
RTBL2   033B     35
RTL     00CC
RTL1    0347      8
RTL2    034F     36
RTNCRC  0483     48
RWBIT   0004     41,    45,    46,    47
RWBRR   013F     11,    11
RWBRX   0146     12
RWSC1   03C2     41,    41
RWSC2   03C4     41
S       00B6
SBT     00EA
SCP     01C6
SCP1    03AE     17
SCP2    03B4     41
SD      00F6
SDR     0076
SE      00D6
SE1     011C      6
SE2     013D     10
SER     0056
SER1    006A      4
SER2    001E      4
SFLOAT  0281
SH      0096
SHI     0196
SI      01F6
SINT    01C4
SINT1   02D1     17
SIS     004E
SLA     01DE
SLHA    019E
SLHL    019A
SLHLS   0122
SLL     01DA
SLLS    0022
SR      0016
SRA     01DC
SRHA    019C
SRHL    0198
SRHLS   0120
SRL     01D8
SRLS    0020
SS      01BA
SS1     0178     16
SSR     013A
ST      00A0
```

```
STB       01A4
STB1      016B      16
STBR      0124
STBR1     0169      11
STD       00E0
STE       00C0
STH       0080
STM       01A0
STM1      0104      16
STMD      00FC
STME      00E2
STME1     0107       9
STORE     017A       6,     7,     8,    16
STORE1    017D      14
STORE2    017E      14
STREE     022D      22
STRLP     0221      21,    21
SVC       01C2
SVC1      02D5      17
TBIT      0002      47,    47
TBT       00E8
TEMP      0010      26,    26,    41,    41,    41,    45,    45,    45,
                    45,    46,    46,    47,    47,    47,    47,    47,
                    47,    48
TEST1     02B4       3,    11
THI       0186
TI        01E6
TLATE     01CE
TLATE1    0375      17
TLSU      0400       1,    50
TRANSL    037D      47,    47
TS        01C0
TS1       01F9      17
TS2       01FC      18
TS3       014C      18
TWAIT     023D      21,    27,    27,    43,    44
WAIT      02AF      22,    46
WAIT1     02B1      27
WB        01AC
WB1       0143      16
WBR       012C
WD        01B4
WD1       0176      16
WDCS      048E      49
WDCS1     048F      49
WDR       0134
WH        01B0
WHR       0130
WRTSC     03C7      41
X         00AE
XH        008E
XHI       018E
XI        01EE
XR        000E
```

```
                                   1          SCRAT                                                   83200010
      0001                         2  DFU     EQU   1                 SYSGEN CARD                     83200020
                                   3          NLSTC                                                   83200030
 000                               4          IFNZ  DFU                                               83200040
                                   7          ENDC                                                    83200080
                                   8          SQCHK                                                   83200090
                                   9  * COPYRIGHT INTERDATA INC.  APRIL, 1976                         83200110
                                  10  *                                                               83200120
                                  11  * IRA GABBERT                                                   83200130
                                  12  *                                                               83200140
                                  13          TARGT 8/32                                              83200150
                                  14          CROSS                                                   83200160


                                  33          ELSE                                                    83200350
                                  34  PAGE0   PARTS 19-142R00F68,19-142R00F69,19-142R00F70,19-142R00F71  83200360
                                  35          PARTS 19-142R00F72,19-142R00F73,19-142R00F74,19-142R00F75  83200370
                                  36  PAGE1   PARTS 19-142R00F76,19-142R00F77,19-142R00F78,19-142R00F79  83200380
                                  37          PARTS 19-142R00F80,19-142R00F81,19-142R00F82,19-142R00F83  83200390
                                  38  PAGE2   PARTS 19-142R00F84,19-142R00F85,19-142R00F86,19-142R00F87  83200400
                                  39          PARTS 19-142R00F88,19-142R00F89,19-142R00F90,19-142R00F91  83200410
                                  40  PAGE3   PARTS 19-142R00F92,19-142R00F93,19-142R00F94,19-142R00F95  83200420
                                  41          PARTS 19-142R00F96,19-142R00F97,19-142R00F98,19-142R00F99  83200430
                                  42  PAGE4,5 PARTS 19-195R00F05,19-195R00F06,19-195R00F07,19-195R00F08  83200440
                                  43          PARTS 19-195R00F09,19-195R00F10,19-195R00F11,19-195R00F12  83200450
                                  44  *                                                               83200460
                                  45  *                                                               83200470
                                  46  *                                                               83200480
                                  47  *                                                               83200490
                                  48  *                                                               83200500
                                  49  *                                                               83200510
                                  50          ENDC                                                    83200520
                                  51  *                                                               83200530
                                  52  * IN ALL CASES WHERE A BRANCH OR TRANSFER COULD OCCUR TO A     83200540
                                  53  * LISTING PAGE OTHER THAN THE CURRENT LISTING PAGE, THE TARGET 83200550
                                  54  * PAGE NUMBER IS SHOWN IN PARENTHESIS IN THE COMMENT FIELD.    83200560
                                  55  *                                                               83200570
                                  56  *                           ON POWER-UP, OR AFTER               83200580
                                  57  *                           INITIALIZE, MICRO CODE              83200590
 000      321F 1005               58          LI    MR0,5         EXECUTION BEGINS AT '001'           83200600
 001      17FD 0000               59          BALD  TLSU(NULL)    GO TO POWER UP ROUTINE (P.44)       83200610


                                  61  * USER  LEVEL  INSTRUCTION  EMULATION  ENTRY-POINTS   *         83200630
                                  62  * FOLLOW.  IN RESPONSE TO AN INSTRUCTION READ COMMAND *         83200640
                                  63  * THE HARDWARE, READS THE NEXT USER INSTRUCTION FROM  *         83200650
                                  64  * THE MAIN MEMORY LOCATION SPECIFIED BY (LOC). TWO,   *         83200660
                                  65  * FOUR OR SIX BYTES ARE READ, DEPENDING UPON THE      *         83200670
                                  66  * INSTRUCTION TYPE.  TWICE THE USER'S OPERATION CODE  *         83200680
                                  67  * IS THE STARTING ADDRESS IN ROM OF THE APPROPRIATE   *         83200690
                                  68  * EMULATION SEQUENCE.  THE OP-CODE IS  SHOWN IN THE   *         83200700
                                  69  * COMMENT FIELD AND THE USER'S MNEMONIC IS THE LABEL. *         83200710
                                  71  *                                                         * 01 *        83200730
002     2A1F 1C01                 72  BALR      L     MR0,YS,IL             SAVE BRANCH ADDRESS                83200740
003     233F 1D05                 73            LX    YD,LOC,BALR1          INCREMENTED LOC TO YD              83200750
                                  74  *                                                         * 02 *        83200760
004     17EC 01D9                 75  BTCR      BALT  BRR(NULL),ILIR,D      BRANCH IF MASK TRUE                83200770
005     235F 1837                 76  BALR1     LX    LOC,MR0,BC2           LOAD LOC (P.3)                     83200780
                                  77  *                                                         * 03 *        83200790
006     13EC 01D9                 78  BFCR      BALF  BRR(NULL),ILIR,D      BRANCH IF MASK FALSE               83200800
007     235F 1C37                 79  BRR       LX    LOC,YS,BC2            LOAD LOC (P.3)                     83200810
                                  80  *                                                         * 04 *        83200820
008     2B39 5C39                 81  NR        N     YD,YD,YS,ILIR,E,D                                        83200830
009     000F FF00                 82  B12,23    DC    '000FFF00'                                               83200840
                                  83  *                                                         * 05 *        83200850
00A     2BF9 0C39                 84  CLR       S     NULL,YD,YS,ILIR,E,D                                      83200860
00B     233B FC0D                 85  DR1       DX    YD,YDP1,YS,DR2        DO DIVIDE                          83200870
                                  86  *                                                         * 06 *        83200880
00C     2B39 7C39                 87  OR        O     YD,YD,YS,ILIR,E,D                                        83200890
00D     13F4 B090                 88  DR2       BALV  DFALT0(NULL),D        (P.28)                             83200900
                                  89  *                                                         * 07 *        83200910
00E     2B39 6C39                 90  XR        X     YD,YD,YS,ILIR,E,D                                        83200920
00F     0001 0000                 91  BIT15     DC    '00010000'                                               83200930
                                  92  *                                                         * 08 *        83200940
010     2B3F 1C39                 93  LR        L     YD,YS,ILIR,E,D                                           83200950
011     4E00 0000                 94  CONSTANT  DC    '4E000000'                                               83200960
                                  95  *                                                         * 09 *        83200970
012     2B7F 1C00                 96  CR        A     MDR,NULL,YS           2ND OP TO MDR                      83200980
013     23F9 6DBC                 97  C1        XX    NULL,YD,MDR,C2        COMPARE SIGNS (P.3)                83200990
                                  98  *                                                         * 0A *        83201000
014     2B39 1C39                 99  AR        A     YD,YD,YS,ILIR,E,D                                        83201010
015     FFFF 0000                100  HIHALF    DC    'FFFF0000'                                               83201020
                                 101  *                                                         * 0B *        83201030
016     2B39 0C39                102  SR        S     YD,YD,YS,ILIR,E,D                                        83201040
017     0000 FFFF                103  LOHALF    DC    '0000FFFF'                                               83201050
                                 104  *                                                         * 0C *        83201060
018     1398 D6C0                105  MHR       BAL   MHR1(MAR)             (P.37)                             83201070
019     F000 0000                106  DIGIT1    DC    'F0000000'                                               83201080
                                 107  *                                                         * 0D *        83201090
01A     1398 D8C0                108  DHR       BAL   DHR1(MAR)             (P.37)                             83201100
01B                              109            IFNZ  DFU                                                      83201110
01B     0000 0000                110            DC    0                                                        83201120
                                 111  *                                                                       83201130
01C     CBF9 2DA9                112  LE1       LE    YD,MDR,ILIR,E         LOAD                               83201140
01D     13F4 1990                113            BALV  EEXIT1(NULL),D        READ CC IF V FLAG (P.5)            83201150
                                 119            ENDC                                                           83201210
                                 120  *                                                                       83201220
01E     2B79 1DA3                121  AHM1      A     MDR,YD,MDR,DW2,E      ADD R1, STORE RESULT               83201230
01F     2BFF 1F99                122            L     NULL,NULL,ILIR,D      FETCH NEXT INSTRUCTION             83201240
```

```
                            124  *                                                        * 10 *       83201260
020     2B39 8EB9           125  SRLS     SRL   YD,YD,YSI,ILIR,E,D                                     83201270
021     0000 4800           126  BI1720   DC    '00004800'                                             83201280
                            127  *                                                        * 11 *       83201290
022     2B39 9EB9           128  SLLS     SLL   YD,YD,YSI,ILIR,E,D                                     83201300
023     FFFF 7FFF           129  BIT160   DC    'FFFF7FFF'                                             83201310
                            130  *                                                        * 12 *       83201320
024     3210 5008           131  CHVR     NI    MR0,PSW,8           SAVE PREVIOUS CARRY                83201330
025     3658 7015           132           OI    MR2,YS,HIHALF,I     SET MS 16 BITS                     83201340
026     3652 6327           133           XI    MR2,MR2,BIT16,I     INVERT THE SIGN BIT                83201350
027     3652 1327           134           AI    MR2,MR2,BIT16,I     AND EXTEND IT                      83201360
028     2A38 6909           135           X     MR1,YS,MR2,ILIR     RE-CREATE HW OVERFLOW BIT          83201370
029     37F1 500F           136           NI    NULL,MR1,BIT15,I    BY COMPARING BITS 15 AND           83201380
02A     13E0 0B00           137           BALZ  CHVR1(NULL)         16 OF R2                           83201390
02B     3210 7004           138           OI    MR0,MR0,4           OVERFLOW IF DIFFER                 83201400
02C     2B3F 1920           139  CHVR1    L     YD,MR2,E            LOAD RESULT, ADJUST G & L          83201410
02D     2BBD 7810           140           O     PSW,PSW,MR0,D       OR IN C & V                        83201420
                            141  *                                                                     83201430
02E     2B3F 1E00           142  LA1      L     YD,MAR              ADDRESS VALUE TO R1                83201440
02F     2BFF 1F9C           143           L     NULL,NULL,IR,D      FETCH NEXT INSTRUCTION             83201450
                            144  *                                                        * 18 *       83201460
030     2A7F 1C00           145  LPSWR    L     MR3,YS              SET NEW PSW ASIDE IN MR3           83201470
031     2BDF 3E80           146           AINC  YDI,NULL,YSI        POINT TO R2+1                      83201480
032     2B5F 1C80           147           L     LOC,YD              LOAD NEW LOC                       83201490
033     13F8 FC00           148           BAL   TEST1(NULL)         (P.43)                             83201500
034     0000 0000           149           DC    0                                                      83201510
                            150  *                                                                     83201520
035     2B9A 1D81           151  BC1      A     MAR,YX,MDR,IL       CALCULATE EFFECTIVE ADDRESS        83201530
036     2B5F 1E00           152  BC3      L     LOC,MAR             LOAD NEW LOC                       83201540
037     2BFF 1F9C           153  BC2      L     NULL,NULL,IR,D      FETCH NEXT INSTRUCTION             83201550
                            154  *                                                        * 1C *       83201560
038     2B3B EC19           155  MR       M     YD,YDP1,YS,ILIR,D                                      83201570
039     0000 FFFE           156  CFFFE    DC    '0000FFFE'                                             83201580
                            157  *                                                        * 1D *       83201590
03A     2A1F 1C89           158  DR       L     MR0,YD,ILIR         SAVE MS DIVIDEND                   83201600
03B     223B 1F8B           159           AX    MR1,YDP1,NULL,DR1   SAVE LS DIVIDEND (P.2)             83201610
                            160  *                                                                     83201620
03C     17E4 2AC9           161  C2       BALNL CL1(NULL),ILIR      SIGNS ALIKE (P.7)                  83201630
03D     3219 C001           162           SRAI  MR0,YD,1            PROPOGATE 1ST OP SIGN              83201640
03E     3210 7001           163           OI    MR0,MR0,1           FORCE SOME MAGNITUDE               83201650
03F     2A10 1830           164           A     MR0,MR0,MR0,E,D     SET CONDITION CODE                 83201660
```

```
                              166  *                                                     * 20 *        83201680
040     17EC 1059             167  BTBS     BALT  BBS(NULL),ILIR,D    BRANCH IF MASK TRUE                 83201690
041     221F 0E83             168  BBS      SX    MRO,NULL,YSI,BBS1   DECREMENT BY TWICE R2               83201700
                              169  *                                                     * 21 *        83201710
042     17EC 1159             170  BTFS     BALT  BFS(NULL),ILIR,D    BRANCH IF MASK TRUE                 83201720
043     2210 0E8D             171  BBS1     SX    MRO,MRO,YSI,DOSBR   MRO=BYTE DISPLACEMENT               83201730
                              172  *                                                     * 22 *        83201740
044     13EC 1059             173  BFBS     BALF  BBS(NULL),ILIR,D    BRANCH IF MASK FALSE                83201750
045     221F 1E87             174  BFS      AX    MRO,NULL,YSI,BFS1   INCREMENT BY TWICE R2               83201760
                              175  *                                                     * 23 *        83201770
046     13EC 1159             176  BFFS     BALF  BFS(NULL),ILIR,D    BRANCH IF MASK FALSE                83201780
047     2210 1E8D             177  BFS1     AX    MRO,MRO,YSI,DOSBR   MRO=BYTE DISPLACEMENT               83201790
                              178  *                                                     * 24 *        83201800
048     2B3F 1EB9             179  LIS      L     YD,YSI,ILIR,E,D                                         83201810
049     221F 1E2F             180  CADRS2   LX    MRO,MAR,CADRS3      EFFECTIVE ADRS TO MRO (P.5)         83201820
                              181  *                                                     * 25 *        83201830
04A     2B3F 0E89             182  LCS      S     YD,NULL,YSI,ILIR    SUBTRACT TO TWO'S COMP              83201840
04B     2B3F 1CB0             183           L     YD,YD,E,D           SET G,L                             83201850
                              184  *                                                     * 26 *        83201860
04C     2B39 1EB9             185  AIS      A     YD,YD,YSI,ILIR,E,D                                      83201870
04D     2350 1D0F             186  DOSBR    AX    LOC,MRO,LOC,DOSBR1  FORM BRANCH ADDRESS                 83201880
                              187  *                                                     * 27 *        83201890
04E     2B39 0EB9             188  SIS      S     YD,YD,YSI,ILIR,E,D                                      83201900
04F     2BFF 1F9C             189  DOSBR1   L     NULL,NULL,IR,D      FETCH NEXT INSTR                    83201910
050                           190                 IFNZ  DFU                                               83201920
                              191  *                                                     * 28 *        83201930
050     CBF9 2C29             192  LER      LE    YD,YS,ILIR,E        LOAD                                83201940
051     13F8 1990             193                 BAL   EEXIT1(NULL),D      READ CC (P.5)                   83201950
                              194  *                                                     * 29 *        83201960
052     CBF9 3C09             195  CER      CER   YD,YS,ILIR          COMPARE                             83201970
053     CBFF 0FB0             196                 RCC   NULL,NULL,E,D       SET CONDITION CODE              83201980
                              197  *                                                     * 2A *        83201990
054     CBF9 4C29             198  AER      AER   YD,YS,ILIR,E        ADD                                 83202000
055     C3FF 0FA7             199                 RCCX  NULL,NULL,EEXIT2      COLLECT FLAGS (P.5)         83202010
                              200  *                                                     * 2B *        83202020
056     CBF9 5C29             201  SER      SER   YD,YS,ILIR,E        SUBTRACT                            83202030
057     C3FF 0FA7             202                 RCCX  NULL,NULL,EEXIT2      COLLECT FLAGS (P.5)         83202040
                              203  *                                                     * 2C *        83202050
058     CBF9 6C09             204  MER      MER   YD,YS,ILIR          MULTIPLY                            83202060
059     13F8 1980             205                 BAL   EEXIT1(NULL)        TO COMMON EXIT (P.5)            83202070
                              206  *                                                     * 2D *        83202080
05A     CBF9 7C09             207  DER      DER   YD,YS,ILIR          DIVIDE                              83202090
05B     13F8 1980             208                 BAL   EEXIT1(NULL)        TO COMMON EXIT (P.5)            83202100
                              209  *                                                     * 2E *        83202110
05C     CA1F 1C00             210  FXR      RRE   MRO,YS              ARGUMENT TO MRO                     83202120
05D     13F8 E240             211                 BAL   FXR1(NULL)          (P.39)                          83202130
                              212  *                                                     * 2F *        83202140
05E     361F 1011             213  FLR      LI    MRO,CONSTANT,I      MRO='4E000000'                      83202150
05F     13F9 35C0             214                 BAL   FLR1(NULL)          (P.54)                          83202160
                              240                 ENDC                                                    83202420
```

```
                               242  *                                                           * 30 *        83202440
060     2A0F 1F00              243  MPBSR    L     MR6,YDI             SAVE R1 FIELD                              83202450
061     1399 45C0              244           BAL   MPBSR1(MAR)         (P.56)                                     83202460
                               245  *                                                                         83202470
062                            246           IFNZ  DFU                                                        83202480
062     CBF9 4DA9              247  AE1      AER   YD,MDR,ILIR,E       ADD                                        83202490
063     C3FF 0FA7              248           RCCX  NULL,NULL,EEXIT2    COLLECT FLAGS                              83202500
                               252           ENDC                                                             83202540
                               253  *                                                           * 32 *        83202550
064     1399 43C0              254  PBR      BAL   PBR1(MAR)           (P.55)                                     83202560
065                            255           IFNZ  DFU                                                        83202570
065     CBF9 5D89              256  SE1      SER   YD,MDR,ILIR         SUBTRACT                                   83202580
                               257  *                                                                         83202590
066     CBFF 0FA0              258  EEXIT1   RCC   NULL,NULL,E         TEST RESULT FLAGS, SET CC                  83202600
067     13F4 B190              259  EEXIT2   BALV  FFAULT(NULL),D      ERROR IF V FLAG (P.28)                     83202610
                               265           ENDC                                                             83202670
                               266  *                                                           * 34 *        83202680
068     3338 A010              267  EXHR     RRI   YD,YS,16            EXCHANGE HALFWORDS                         83202690
069     2BFF 1C99              268           L     NULL,YD,ILIR,D                                             83202700
                               269  *                                                                         83202710
06A                            270           IFNZ  DFU                                                        83202720
06A     CBF9 6D89              271  ME1      MER   YD,MDR,ILIR         MULTIPLY                                   83202730
06B     13F8 1980              272           BAL   EEXIT1(NULL)                                               83202740
                               276           ENDC                                                             83202780
                               277  *                                                                         83202790
06C     239A 1DC9              278  CADRS    AX    MAR,YX,MDR,CADRS2,C TRANSFER IF RX1 OR RX3 (P.4)               83202800
06D     2A1F 1E00              279           L     MR0,MAR             D2+(X2)+(LOC) TO MR0                       83202810
06E     3210 1004              280  CADRS1   AI    MR0,MR0,4           ADD 4 FOR LOC INCREMENT                    83202820
06F     03F8 0B00              281  CADRS3   BAL   (MR6)(NULL)         RETURN TO CALL                             83202830
                               282  *                                                           * 38 *        83202840
070     CBF9 AC29              283  LDR      LD    YD,YS,ILIR,E        LOAD                                       83202850
071     13F8 1990              284           BAL   EEXIT1(NULL),D      READ CC                                    83202860
                               285  *                                                           * 39 *        83202870
072     CBF9 BC09              286  CDR      CDR   YD,YS,ILIR          COMPARE                                    83202880
073     CBFF 0FB0              287           RCC   NULL,NULL,E,D       SET CONDITION CODE                         83202890
                               288  *                                                           * 3A *        83202900
074     CBF9 CC09              289  ADR      ADR   YD,YS,ILIR          ADD                                        83202910
075     13F8 1980              290           BAL   EEXIT1(NULL)        TO COMMON EXIT                             83202920
                               291  *                                                           * 3B *        83202930
076     CBF9 DC09              292  SDR      SDR   YD,YS,ILIR          SUBTRACT                                   83202940
077     13F8 1980              293           BAL   EEXIT1(NULL)        TO COMMON EXIT                             83202950
                               294  *                                                           * 3C *        83202960
078     CBF9 EC09              295  MDR      MDR   YD,YS,ILIR          MULTIPLY                                   83202970
079     13F8 1980              296           BAL   EEXIT1(NULL)        TO COMMON EXIT                             83202980
                               297  *                                                           * 3D *        83202990
07A     CBF9 FC09              298  DDR      DDR   YD,YS,ILIR          DIVIDE                                     83203000
07B     13F8 1980              299           BAL   EEXIT1(NULL)        TO COMMON EXIT                             83203010
                               300  *                                                           * 3E *        83203020
07C     CA1F 9C00              301  FXDR     RRD   MR0,YS              ARGUMENT TO MR0                            83203030
07D     13F9 3000              302           BAL   FXDR1(NULL)         (P.52)                                     83203040
                               303  *                                                           * 3F *        83203050
07E     361F 1011              304  FLDR     LI    MR0,CONSTANT,I      MR0='4E000000'                             83203060
07F     13F9 32C0              305           BAL   FLDR1(NULL)         (P.53)                                     83203070
```

```
                         *                                                  * 40 *     83203090
080   12D8 5E80          STH    BAL    STORE(MR6)          COMMON ROUTINE (P.14)       83203100
081   2B7F 1C83                 L      MDR,YD,DW2          EXECUTED INSTR.             83203110
                         *                                                  * 41 *     83203120
082   2B9A 1D81          BAL    A      MAR,YX,MDR,IL       CALCULATE EFFECTIVE ADDRESS 83203130
083   221F 1E05                 LX     MR0,MAR,BAL1        BRANCH ADDRESS TO MR0       83203140
                         *                                                  * 42 *     83203150
084   17EC 0D59          BTC    BALT   BC1(NULL),ILIR,D    BRANCH IF MASK TRUE (P.3)   83203160
085   233F 1D07          BAL1   LX     YD,LOC,BAL2         INCREMENTED LOC TO R1       83203170
                         *                                                  * 43 *     83203180
086   13EC 0D59          BFC    BALF   BC1(NULL),ILIR,D    BRANCH IF MASK FALSE (P.3)  83203190
087   235F 181F          BAL2   LX     LOC,MR0,BAL3        LOAD NEW LOC                83203200
                         *                                                  * 44 *     83203210
088   2B9A 1D8B          NH     A      MAR,YX,MDR,DR2                                  83203220
089   2B39 5DB9                 N      YD,YD,MDR,ILIR,E,D                              83203230
                         *                                                  * 45 *     83203240
08A   2B9A 1D8B          CLH    A      MAR,YX,MDR,DR2                                  83203250
08B   2BF9 0DB9                 S      NULL,YD,MDR,ILIR,E,D                            83203260
                         *                                                  * 46 *     83203270
08C   2B9A 1D8B          OH     A      MAR,YX,MDR,DR2                                  83203280
08D   2B39 7DB9                 O      YD,YD,MDR,ILIR,E,D                              83203290
                         *                                                  * 47 *     83203300
08E   2B9A 1D8B          XH     A      MAR,YX,MDR,DR2                                  83203310
08F   2B39 6DB9                 X      YD,YD,MDR,ILIR,E,D                              83203320
                         *                                                  * 48 *     83203330
090   2B9A 1D8B          LH     A      MAR,YX,MDR,DR2                                  83203340
091   2B3F 1DB9                 L      YD,MDR,ILIR,E,D                                 83203350
                         *                                                  * 49 *     83203360
092   2B9A 1D8B          CH     A      MAR,YX,MDR,DR2                                  83203370
093   13F8 04C0                 BAL    C1(NULL)            (P.2)                       83203380
                         *                                                  * 4A *     83203390
094   2B9A 1D8B          AH     A      MAR,YX,MDR,DR2                                  83203400
095   2B39 1DB9                 A      YD,YD,MDR,ILIR,E,D                              83203410
                         *                                                  * 4B *     83203420
096   2B9A 1D8B          SH     A      MAR,YX,MDR,DR2                                  83203430
097   2B39 0DB9                 S      YD,YD,MDR,ILIR,E,D                              83203440
                         *                                                  * 4C *     83203450
098   2B9A 1D8B          MH     A      MAR,YX,MDR,DR2      FETCH MULTIPLIER            83203460
099   13F8 D780                 BAL    MH1(NULL)           (P.37)                      83203470
                         *                                                  * 4D *     83203480
09A   2B9A 1D8B          DH     A      MAR,YX,MDR,DR2      FETCH DIVISOR               83203490
09B   13F8 D980                 BAL    DH1(NULL)           (P.37)                      83203500
                         *                                                             83203510
09C   2A3B 1F80          DO     A      MR1,YDP1,NULL                                   83203520
09D   2B3B FD89                 D      YD,YDP1,MDR,ILIR                                83203530
09E   13F4 B090                 BALV   DFALTO(NULL),D      (P.28)                      83203540
09F   2BFF 1F9C          BAL3   L      NULL,NULL,IR,D                                  83203550
```

```
                            355  *                                                           * 50 *       83203570
0A0     12D8 5E80           356  ST       BAL   STORE(MR6)          COMMON ROUTINE (P.14)                 83203580
0A1     2B7F 1C87           357           L     MDR,YD,DW4          EXECUTED INSTR.                       83203590
                            358  *                                                           * 51 *       83203600
0A2     12D8 BC00           359  AM       BAL   RDFULL(MR6)         FETCH FULLWORD (P.31)                 83203610
0A3     2B79 1DA7           360           A     MDR,YD,MDR,DW4,E                                          83203620
0A4     2BFF 1F99           361           L     NULL,NULL,ILIR,D                                          83203630
                            362  *                                                                        83203640
0A5     CBFF 8D8D           363  LD1      LW    NULL,MDR,I4DR4      LOAD MS 32 BITS, FETCH LS 32          83203650
0A6     CBF9 ADA9           364           LD    YD,MDR,ILIR,E       LOAD LS 32 BITS                       83203660
0A7     13F4 1990           365           BALV  EEXIT1(NULL),D      TO COMMON EXIT IF V FLAG              83203670
                            366  *                                                                        83203680
                            367  *                                                           * 54 *       83203690
0A8     2B9A 1D8F           368  N        A     MAR,YX,MDR,DR4                                            83203700
0A9     2B39 5DB9           369           N     YD,YD,MDR,ILIR,E,D                                        83203710
                            370  *                                                           * 55 *       83203720
0AA     2B9A 1D8F           371  CL       A     MAR,YX,MDR,DR4                                            83203730
0AB     2BF9 0DB9           372  CL1      S     NULL,YD,MDR,ILIR,E,D                                      83203740
                            373  *                                                           * 56 *       83203750
0AC     2B9A 1D8F           374  O        A     MAR,YX,MDR,DR4                                            83203760
0AD     2B39 7DB9           375           O     YD,YD,MDR,ILIR,E,D                                        83203770
                            376  *                                                           * 57 *       83203780
0AE     2B9A 1D8F           377  X        A     MAR,YX,MDR,DR4                                            83203790
0AF     2B39 6DB9           378           X     YD,YD,MDR,ILIR,E,D                                        83203800
                            379  *                                                           * 58 *       83203810
0B0     2B9A 1D8F           380  L        A     MAR,YX,MDR,DR4                                            83203820
0B1     2B3F 1DB9           381           L     YD,MDR,ILIR,E,D                                           83203830
                            382  *                                                           * 59 *       83203840
0B2     2B9A 1D8F           383  C        A     MAR,YX,MDR,DR4                                            83203850
0B3     13F8 04C0           384           BAL   C1(NULL)            (P,2)                                 83203860
                            385  *                                                           * 5A *       83203870
0B4     2B9A 1D8F           386  A        A     MAR,YX,MDR,DR4                                            83203880
0B5     2B39 1DB9           387           A     YD,YD,MDR,ILIR,E,D                                        83203890
                            388  *                                                           * 5B *       83203900
0B6     2B9A 1D8F           389  S        A     MAR,YX,MDR,DR4                                            83203910
0B7     2B39 0DB9           390           S     YD,YD,MDR,ILIR,E,D                                        83203920
                            391  *                                                           * 5C *       83203930
0B8     2B9A 1D8F           392  M        A     MAR,YX,MDR,DR4                                            83203940
0B9     2B3B ED99           393           M     YD,YDP1,MDR,ILIR,D                                        83203950
                            394  *                                                           * 5D *       83203960
0BA     2B9A 1D8F           395  D        A     MAR,YX,MDR,DR4                                            83203970
0BB     221F 1C9C           396           LX    MR0,YD,D0           (P,6)                                 83203980
                            397  *                                                           * 5E *       83203990
0BC     12D8 BA80           398  CRC12    BAL   RDHALF(MR6)         FETCH CHECKWORD (P.31)                83204000
0BD     13FC F340           399           BALA  CRC12A(NULL)        ARM INTERRUPTS (P.42)                 83204010
                            400  *                                                           * 5F *       83204020
0BE     12D8 BA80           401  CRC16    BAL   RDHALF(MR6)         FETCH CHECKWORD (P.31)                83204030
0BF     13FC F500           402           BALA  CRC16A(NULL)        ARM INTERRUPTS (P.42)                 83204040
```

```
                            404  *                                                         * 60 *       83204060
0C0     12D8 5E80           405  STE       BAL   STORE(MR6)          COMMON ROUTINE (P.14)                 83204070
0C1                         406            IFNZ  DFU                                                       83204080
0C1     CB7F 1C87           407            RRE   MDR,YD,DW4          EXECUTED INSTRUCTION                  83204090
                            410            ENDC                                                            83204120
                            411  *                                                         * 61 *       83204130
0C2     12D8 BA80           412  AHM       BAL   RDHALF(MR6)         FETCH HALFWORD (P.31)                 83204140
0C3     13F8 0780           413            BAL   AHM1(NULL)          (P.2)                                 83204150
                            414  *                                                         * 62 *       83204160
0C4     12D8 BA80           415  PB        BAL   RDHALF(MR6)         FETCH RESIDUAL CHECKSUM (P.31)        83204170
0C5     13F9 4200           416            BAL   PB1(NULL)           (P.55)                                83204180
                            417  *                                                         * 63 *       83204190
0C6     12D8 1B00           418  LRA       BAL   CADRS(MR6)          CALCULATE ADDRESS (P.5)               83204200
0C7     1378 E6C0           419            BAL   LRA1(MDR)           (P.40)                                83204210
                            420  *                                                         * 64 *       83204220
0C8     12D8 C000           421  ATL       BAL   ATBL(MR6)           COMMON OVERHEAD (P.32)                83204230
0C9     13F8 C400           422            BAL   ATL1(NULL)          (P.33)                                83204240
                            423  *                                                         * 65 *       83204250
0CA     12D8 C000           424  ABL       BAL   ATBL(MR6)           COMMON OVERHEAD (P.32)                83204260
0CB     13F8 C700           425            BAL   ABL1(NULL)          (P.33)                                83204270
                            426  *                                                         * 66 *       83204280
0CC     12D8 CDC0           427  RTL       BAL   RTBL(MR6)           COMMON OVERHEAD (P.35)                83204290
0CD     13F8 D1C0           428            BAL   RTL1(NULL)          (P.36)                                83204300
                            429  *                                                         * 67 *       83204310
0CE     12D8 CDC0           430  RBL       BAL   RTBL(MR6)           COMMON OVERHEAD (P.35)                83204320
0CF     13F8 D440           431            BAL   RBL1(NULL)          (P.36)                                83204330
0D0                         432            IFNZ  DFU                                                       83204340
                            433  *                                                         * 68 *       83204350
0D0     2B9A 1D8F           434  LE        A     MAR,YX,MDR,DR4                                            83204360
0D1     13F8 0700           435            BAL   LE1(NULL)           (P.2)                                 83204370
                            436  *                                                         * 69 *       83204380
0D2     2B9A 1D8F           437  CE        A     MAR,YX,MDR,DR4      FETCH COMPARAND                       83204390
0D3     13F8 4680           438            BAL   CE1(NULL)           (P.10)                                83204400
                            439  *                                                         * 6A *       83204410
0D4     2B9A 1D8F           440  AE        A     MAR,YX,MDR,DR4      FETCH ADDEND                          83204420
0D5     13F8 1880           441            BAL   AE1(NULL)           (P.5)                                 83204430
                            442  *                                                         * 6B *       83204440
0D6     2B9A 1D8F           443  SE        A     MAR,YX,MDR,DR4      FETCH SUBTRAHEND                      83204450
0D7     13F8 1940           444            BAL   SE1(NULL)           (P.5)                                 83204460
                            445  *                                                         * 6C *       83204470
0D8     2B9A 1D8F           446  ME        A     MAR,YX,MDR,DR4      FETCH MULTIPLIER                      83204480
0D9     13F8 1A80           447            BAL   ME1(NULL)           (P.10)                                83204490
                            448  *                                                         * 6D *       83204500
0DA     2B9A 1D8F           449  DE        A     MAR,YX,MDR,DR4      FETCH DIVISOR                         83204510
0DB     CBF9 7D89           450            DER   YD,MDR,ILIR         DIVIDE                                83204520
0DC     13F8 1980           451            BAL   EEXIT1(NULL)        TO COMMON EXIT (P.5)                  83204530
0DD     0000 0000           452            DC    0                                                         83204540
                            474            ENDC                                                            83204760
                            475  *                                                                         83204770
0DE     2BFF 1F83           476  CBT2      L     NULL,NULL,DW2       STORE MODIFIED BIT                    83204780
0DF     2BFF 1F99           477            L     NULL,NULL,ILIR,D    FETCH NEXT INSTRUCTION                83204790
```

```
                               479  *                                                          * 70 *         83204810
0E0     2B9A 1D81              480  STD      A     MAR,YX,MDR,IL        CALCULATE ADDRESS                     83204820
0E1     13F9 2C80              481           BAL   STD1(NULL)           (P.52)                                83204830
                               482  *                                                          * 71 *         83204840
0E2     12D8 1B00              483  STME     BAL   CADRS(MR6)           CALCULATE ADDRESS (P.5)               83204850
0E3     13F8 4180              484           BAL   STME1(NULL)          (P.10)                                83204860
                               485  *                                                          * 72 *         83204870
0E4     12D8 BC00              486  LME      BAL   RDFULL(MR6)          READ 1ST FULLWORD (P.31)              83204880
0E5     13F8 4300              487           BAL   LME1(NULL)           (P.10)                                83204890
                               488  *                                                          * 73 *         83204900
0E6     2B9A 1D8B              489  LHL      A     MAR,YX,MDR,DR2                                             83204910
0E7     13F8 8D80              490           BAL   LHL1(NULL)           (P.31)                                83204920
                               491  *                                                          * 74 *         83204930
0E8     12B8 B780              492  TBT      BAL   COMBIT(MR5)          (P.30)                                83204940
0E9     2BFF 1F99              493           L     NULL,NULL,ILIR,D                                           83204950
                               494  *                                                          * 75 *         83204960
0EA     12B8 B780              495  SBT      BAL   COMBIT(MR5)          (P.30)                                83204970
0EB     2372 7D9E              496           OX    MDR,MR2,MDR,CBT2     (P.8)                                 83204980
                               497  *                                                          * 76 *         83204990
0EC     12B8 B780              498  RBT      BAL   COMBIT(MR5)          (P.30)                                83205000
0ED     2372 7DAF              499           OX    MDR,MR2,MDR,CBT1                                           83205010
                               500  *                                                          * 77 *         83205020
0EE     12B8 B780              501  CBT      BAL   COMBIT(MR5)          (P.30)                                83205030
0EF     2372 6D9E              502  CBT1     XX    MDR,MR2,MDR,CBT2     (P.8)                                 83205040
                               503  *                                                          * 78 *         83205050
0F0     12D8 BC00              504  LD       BAL   RDFULL(MR6)          FETCH MS 32 BITS (P.31)               83205060
0F1     13F8 2940              505           BAL   LD1(NULL)            (P.7)                                 83205070
                               506  *                                                          * 79 *         83205080
0F2     12D8 44C0              507  CD       BAL   CDADSDMD(MR6)        FETCH COMPARAND (P.10)                83205090
0F3     13F8 BE00              508           BAL   CD1(NULL)            (P.31)                                83205100
                               509  *                                                          * 7A *         83205110
0F4     12D8 44C0              510  AD       BAL   CDADSDMD(MR6)        FETCH ADDEND (P.10)                   83205120
0F5     13F8 4700              511           BAL   AD1(NULL)            (P.10)                                83205130
                               512  *                                                          * 7B *         83205140
0F6     12D8 44C0              513  SD       BAL   CDADSDMD(MR6)        FETCH SUBTRAHEND (P.10)               83205150
0F7     13F8 4780              514           BAL   SD1(NULL)            (P.10)                                83205160
                               515  *                                                          * 7C *         83205170
0F8     12D8 44C0              516  MD       BAL   CDADSDMD(MR6)        FETCH MULTIPLIER (P.10)               83205180
0F9     13F8 4F40              517           BAL   MD1(NULL)            (P.11)                                83205190
                               518  *                                                          * 7D *         83205200
0FA     12D8 44C0              519  DD       BAL   CDADSDMD(MR6)        FETCH DIVISOR (P.10)                  83205210
0FB     13F8 59C0              520           BAL   DD1(NULL)            (P.13)                                83205220
                               521  *                                                          * 7E *         83205230
0FC     12D8 1B00              522  STMD     BAL   CADRS(MR6)           CALCULATE ADDRESS (P.5)               83205240
0FD     13F9 2DC0              523           BAL   STMD1(NULL)          (P.52)                                83205250
                               524  *                                                          * 7F *         83205260
0FE     12D8 BC00              525  LMD      BAL   RDFULL(MR6)          FETCH MS 32 BITS (P.31)               83205270
0FF     13F9 2E80              526           BAL   LMD1(NULL)           (P.52)                                83205280
```

```
100                          528         ORG   '100'                                                      83205300
                             529   *                                                                      83205310
100     13F0 448D            530   COMLML    BALC  EXLSTM(NULL),I4DR4  EXIT IF DONE, ELSE READ NEXT       83205320
101     0BF8 0B00            531   COMLM     EXL   (MR6)(NULL)         EXECUTE LOAD                       83205330
102     23D1 1F00            532             AX    YDI,MR1,YDI,COMLML  BUMP R1 FIELD & LOOP               83205340
                             533   *                                                                      83205350
                             534   *                                                                      83205360
103     323F 000F            535   STM1      SI    MR1,NULL,15         MR1='FFFFFFF1'                     83205370
104     12D8 43C0            536             BAL   COMSTM(MR6)         TO COMMON ROUTINE                  83205380
105     2B7F 1C85            537             L     MDR,YD,I4DW4        EXECUTED INSTRUCTION               83205390
                             538   *                                                                      83205400
106     323F 000E            539   STME1     SI    MR1,NULL,14         MR1='FFFFFFF2'                     83205410
107     12D8 43C0            540             BAL   COMSTM(MR6)         TO COMMON ROUTINE                  83205420
108                          541             IFNZ  DFU                                                    83205430
108     CB7F 1C85            542             RRE   MDR,YD,I4DW4        EXECUTED INSTRUCTION               83205440
                             545             ENDC                                                         83205470
                             546   *                                                                      83205480
109     323F 000F            547   LM1       SI    MR1,NULL,15         MR1='FFFFFFF1'                     83205490
10A     12D8 4040            548             BAL   COMLM(MR6)          TO COMMON ROUTINE                  83205500
10B     2B3F 1D80            549             L     YD,MDR              EXECUTED INSTRUCTION               83205510
                             550   *                                                                      83205520
10C     323F 000E            551   LME1      SI    MR1,NULL,14         MR1='FFFFFFF2'                     83205530
10D     12D8 4040            552             BAL   COMLM(MR6)          TO COMMON ROUTINE                  83205540
10E                          553             IFNZ  DFU                                                    83205550
10E     CBF9 2DC0            554             LE    YD,MDR,K            EXECUTED INSTRUCTION               83205560
                             557             ENDC                                                         83205590


10F     3390 0004            559   COMSTM    SI    MAR,MR0,4           TEMPORARY DECREMENT                83205610
110     0BF8 0B00            560   CMSTML    EXL   (MR6)(NULL)         EXECUTE STORE                      83205620
111     23D1 1F50            561             AX    YDI,MR1,YDI,CMSTML,C  BUMP R1 FIELD & LOOP             83205630
112     2BFF 1F99            562   EXLSTM    L     NULL,NULL,ILIR,D    FETCH NEXT INSTRUCTION             83205640


113     239A 1DD6            564   CDADSDMD  AX    MAR,YX,MDR,CASMD1,C TRANSFER IF RX1 OR RX3             83205660
114     321F 1004            565             LI    MR0,4               ADD 4 IF RX2                       83205670
115     2390 1E17            566             AX    MAR,MR0,MAR,CASMD2  LOAD ADDRESS                       83205680
116     2B9F 1E00            567   CASMD1    L     MAR,MAR             LOAD ADDRESS                       83205690
117     2B7F 1F8F            568   CASMD2    L     MDR,NULL,DR4        FETCH MS 32 BITS                   83205700
118     CBFF 8D8D            569             LW    NULL,MDR,I4DR4      LOAD MS 32 BITS, FETCH LS 32       83205710
119     03F8 0B00            570             BAL   (MR6)(NULL)         RETURN                             83205720
                             571   *                                                                      83205730
11A                          572             IFNZ  DFU                                                    83205740
11A     CBF9 3D89            573   CE1       CER   YD,MDR,ILIR         COMPARE                            83205750
11B     CBFF 0FB0            574             RCC   NULL,NULL,E,D       SET CONDITION CODE                 83205760
                             578             ENDC                                                         83205800
                             579   *                                                                      83205810
11C     CBF9 CD89            580   AD1       ADR   YD,MDR,ILIR         ADD                                83205820
11D     13F8 1980            581             BAL   EEXIT1(NULL)        TO COMMON EXIT                     83205830
                             582   *                                                                      83205840
11E     CBF9 DD89            583   SD1       SDR   YD,MDR,ILIR         SUBTRACT                           83205850
11F     13F8 1980            584             BAL   EEXIT1(NULL)        TO COMMON EXIT                     83205860
```

```
                            586  *                                                      * 90 *       83205880
120     2B39 8EF9           587  SRHLS    SRHL  YD,YD,YSI,ILIR,E,D                                   83205890
121     0000 0F01           588  C0F01    DC    '00000F01'                                            83205900
                            589  *                                                      * 91 *       83205910
122     2B39 9EF9           590  SLHLS    SLHL  YD,YD,YSI,ILIR,E,D                                   83205920
123     0000 A001           591  CA001    DC    '0000A001'                                            83205930
                            592  *                                                      * 92 *       83205940
124     3618 5015           593  STBR     NI    MR0,YS,HIHALF,I     SAVE MS 16 BITS                   83205950
125     13F8 5A40           594           BAL   STBR1(NULL)         (P,13)                            83205960
                            595  *                                                      * 93 *       83205970
126     4B3F 5C59           596  LBR      LBR   YD,YS,ILIR,D                                          83205980
127     13F8 FC00           597  EPSR1    BAL   TEST1(NULL)         (P,43)                            83205990
                            598  *                                                      * 94 *       83206000
128     3619 5015           599  EXBR     NI    MR0,YD,HIHALF,I     SAVE MS 16 BITS                   83206010
129     13F8 5B40           600           BAL   EXBR1(NULL)         (P,13)                            83206020
                            601  *                                                      * 95 *       83206030
12A     2B3D 1F81           602  EPSR     A     YD,PSW,NULL,IL      PSW TO R1, INC, LOC               83206040
12B     227F 1C27           603           LX    MR3,YS,EPSR1        NEW PSW TO MR3                    83206050
                            604  *                                                      * 96 *       83206060
12C     12D8 5000           605  WBR      BAL   RWBRR(MR6)          (P,12)                            83206070
12D     4BFF 1D80           606           WD    NULL,MDR                                              83206080
                            607  *                                                      * 97 *       83206090
12E     12D8 5000           608  RBR      BAL   RWBRR(MR6)          (P,12)                            83206100
12F     4B7F 0D83           609           RD    MDR,MDR,DW2                                           83206110
                            610  *                                                      * 98 *       83206120
130     4BF9 DC39           611  WHR      WHA   NULL,YD,YS,ILIR,E,D                                   83206130
131     0000 2800           612  BI1820   DC    '00002800'                                            83206140
                            613  *                                                      * 99 *       83206150
132     4B19 CFB9           614  RHR      RHA   YS,YD,NULL,ILIR,E,D                                   83206160
133     8000 0000           615  BIT0     DC    '80000000'                                            83206170
                            616  *                                                      * 9A *       83206180
134     4BF9 9C79           617  WDR      WDRA  NULL,YD,YS,ILIR,E,D                                   83206190
135     0000 0000           618           DC    0                                                     83206200
                            619  *                                                      * 9B *       83206210
136     4B19 8FF9           620  RDR      RDRA  YS,YD,NULL,ILIR,E,D                                   83206220
137     0000 0000           621           DC    0                                                     83206230
                            622  *                                                                   83206240
138                         623           IFNZ  DFU                                                   83206250
138     0000 0000           624           DC    0                                                     83206260
139     0000 0000           625           DC    0                                                     83206270
                            629           ENDC                                                        83206310
                            630  *                                                      * 9D *       83206320
13A     4B19 AFF9           631  SSR      SSRA  YS,YD,NULL,ILIR,E,D                                   83206330
13B     7FFF FFFF           632  ONES     DC    '7FFFFFFF'                                            83206340
                            633  *                                                      * 9E *       83206350
13C     4BF9 BC79           634  OCR      OCRA  NULL,YD,YS,ILIR,E,D                                   83206360
                            635  *                                                                   83206370
13D                         636           IFNZ  DFU                                                   83206375
13D     CBF9 ED89           637  MD1      MDR   YD,MDR,ILIR         MULTIPLY                          83206380
13E     13F8 1980           638           BAL   EEXIT1(NULL)        TO COMMON EXIT (P,5)              83206385
13F     0000 0000           639           DC    0                                                     83206390
                            645           ENDC                                                        83206416
```

```
                                647  * COMMON RBR,WBR                                                         83206430
                                648  *                                                                       83206440
140     2A5F 1C00               649  RWBRR     L     MR2,YS               MR2=START ADRS                     83206450
141     4B79 AFC0               650            SSRA  MDR,YD,NULL          ADDRESS THE DEVICE                 83206460
142     2BDF 3E80               651            AINC  YDI,NULL,YSI         POINT TO R2+1                      83206470
143     227F 1C8B               652            LX    MR3,YD,CMNBRW        MR3=ENDING ADRS                    83206480
                                653  *                                                                       83206490
                                654  *                                                                       83206500
144     12D8 51C0               655  WB1       BAL   RWBRX(MR6)           COMMON SET-UP ROUTINE              83206510
145     4BFF 1D80               656            WD    NULL,MDR             EXECUTED INSTRUCTION               83206520
                                657  *                                                                       83206530
                                658  *                                                                       83206540
146     32DF 1166               659  RB1       LI    MR6,RB2                                                 83206550
                                660  *                                                                       83206560
                                661  *                                                                       83206570
                                662  * COMMON RB,WB                                                          83206580
                                663  *                                                                       83206590
147     2B7F 1D80               664  RWBRX     L     MDR,MDR                                                 83206600
148     2A5F 1D8D               665            L     MR2,MDR,I4DR4        MR2=START ADDRESS                  83206610
149     4BF9 AFC0               666            SSRA  NULL,YD,NULL         ADDRESS THE DEVICE                 83206620
14A     2A7F 1D80               667            L     MR3,MDR              MR3=ENDING ADDRESS                 83206630
                                668  *                                                                       83206640
14B     33DF 1007               669  CMNBRW    LI    YDI,7                CC MASK                            83206650
14C     23F3 094E               670            SX    NULL,MR3,MR2,BRW2,C  COMPARE START:END                  83206660
                                671  *                                    CLEAR CONDITION CODE               83206670
14D     2BFF 1FB9               672            L     NULL,NULL,ILIR,E,D   AND EXIT                           83206680
                                673  *                                                                       83206690
14E     2B9F 190B               674  BRW2      L     MAR,MR2,DR2          FETCH HALFWORD                     83206700
14F     4BFF 2FE0               675  BRW3      SSR   NULL,NULL,E          SENSE DEVICE STATUS                83206710
150     17EC 5540               676            BALT  ENDBRW(NULL)         EXIT IF BAD STATUS                 83206720
151     13F0 53C0               677            BALC  BRW3(NULL)           LOOP ON BUSY                       83206730
152     0BF8 0B00               678            EXL   (MR6)(NULL)          DO INSTR AT (MR6)                  83206740
153     3252 1001               679  BRW4      AI    MR2,MR2,1            INCREMENT ADDRESS                  83206750
154     23F3 094E               680            SX    NULL,MR3,MR2,BRW2,C  COMPARE TO FINAL & LOOP            83206760
                                681  *                                                                       83206770
155     2BFF 1F99               682  ENDBRW    L     NULL,NULL,ILIR,D     EXIT                               83206780
```

```
                                 684  * AUTO-LOAD                                                                   83206800
                                 685  *                                                                           83206810
156     339F 1078                686  AL1       LI    MAR,X'78'                                                   83206820
                                 687  *                                     KILL ADRS CALCULATION LOGIC           83206830
157     2B7F 1F8A                688            L     MDR,NULL,PR2          AND FETCH BINDV SPECIFICATION         83206840
158     325F 1080                689            LI    MR2,'80'              MR2=START ADDRESS                     83206850
159     2A7F 1800                690            L     MR3,MR0               MR3=FINAL ADDRESS                     83206860
15A     2A3F 1D80                691            L     MR1,MDR                                                     83206870
15B     4A31 DFC0                692            LB    MR1,MR1,NULL          MR1=8-BIT DEVICE NUMBER               83206880
15C     4BF1 8DC0                693            OCRA  NULL,MR1,MDR                                                83206890
15D     2B9F 190B                694            L     MAR,MR2,DR2           ADRS & READ 1ST HW                    83206900
15E     330F 1007                695            LI    YDI,7                 SET CC MASK                           83206910
                                 696  *                                                                           83206920
15F     4BFF 2FE0                697  AL2       SSR   NULL,NULL,E                                                 83206930
160     17EC 5540                698            BALT  ENDBRW(NULL)          EXIT IF BAD STATUS (P.12)             83206940
161     13F0 57C0                699            BALC  AL2(NULL)             LOOP ON BUSY                          83206950
162     4A1F 0FC0                700            RDR   MR0,NULL              INPUT DATA BYTE                       83206960
163     23FF 085F                701            SX    NULL,NULL,MR0,AL2,C LOOP IF BLANK                           83206970
164     4B70 CDC3                702            STB   MDR,MR0,MDR,DW2       STORE FIRST NON-ZERO BYTE AND         83206980
165     12D8 54C0                703            BAL   BRW4(MR6)             CONTINUE INPUT (P.12)                 83206990
166     4B7F 0D83                704  RB2       RD    MDR,MDR,DW2           EXECUTED INSTRUCTION                  83207000
                                 705  *                                                                           83207010
167     CBF9 FD89                706  DD1       DDR   YD,MDR,ILIR           DIVIDE                                83207020
168     13F8 1980                707            BAL   EEXIT1(NULL)          TO COMMON EXIT                        83207030
                                 708  *                                                                           83207040
169     4B19 4C49                709  STBR1     STBR  YS,YD,YS,ILIR         COPY R1 24:31 TO R2 24:31             83207050
16A     2B18 7810                710            O     YS,YS,MR0,D           RESTORE R2 0:15                       83207060
                                 711  *                                                                           83207070
16B     4B79 CDC3                712  STB1      STB   MDR,YD,MDR,DW2        COPY R1 24:31 TO MEMORY               83207080
16C     2BFF 1F99                713            L     NULL,NULL,ILIR,D                                            83207090
                                 714  *                                                                           83207100
16D     4B3F EC49                715  EXBR1     EXB   YD,YS,ILIR            SWAP BYTES                            83207110
16E     2B39 7810                716            O     YD,YD,MR0,D           RESTORE R1 0:15                       83207120
                                 717  *                                                                           83207130
16F     2A1F 1D80                718  LB1       L     MR0,MDR               COPY HW TO MR0                        83207140
170     4B30 DDC0                719            LB    YD,MR0,MDR            MS OR LS BYTE TO R1                   83207150
171     2BFF 1F99                720            L     NULL,NULL,ILIR,D                                            83207160
                                 721  *                                                                           83207170
172     3239 50FF                722  CLB1      NI    MR1,YD,'FF'           ISOLATE 1ST OP BYTE                   83207180
173     2A1F 1D80                723            L     MR0,MDR               2ND OP BYTE IS IN MDR                 83207190
174     4A10 DDC0                724            LB    MR0,MR0,MDR           GET IT                                83207200
175     2BF1 0839                725            S     NULL,MR1,MR0,ILIR,E,D  SUBTRACT TO COMPARE                  83207210
                                 726  *                                                                           83207220
176     4BF9 9DA0                727  WD1       WDA   NULL,YD,MDR,E         ADRS DEV & OUTPUT BYTE                83207230
177     2BFF 1F99                728            L     NULL,NULL,ILIR,D                                            83207240
                                 729  *                                                                           83207250
178     4B79 ADA3                730  SS1       SSA   MDR,YD,MDR,DW2,E      ADRS DEV & INPUT STATUS               83207260
179     2BFF 1F99                731            L     NULL,NULL,ILIR,D                                            83207270
```

```
                              733  * COMMON SUBROUTINE FOR STH,ST,STE,RH          *                   83207290


17A     239A 1DFD             735  STORE     AX     MAR,YX,MDR,STORE1,C TRANSFER IF RX1 OR RX3             83207310
17B     321F 1004             736            LI     MR0,4               ADD 4 IF RX2                       83207320
17C     2390 1E3E             737            AX     MAR,MR0,MAR,STORE2  UPDATE MAR                         83207330
17D     2B9F 1E00             738  STORE1    L      MAR,MAR             MAR=EFFECTIVE ADDRESS              83207340
17E     0BF8 0B00             739  STORE2    EXL    (MR6)(NULL)         EXECUTE INSTRUCTION                83207350
17F     2BFF 1F99             740            L      NULL,NULL,ILIR,D    DO NEXT INSTR                      83207360
```

```
                                 742  *                                                      * C0 *       83207380
180      12D8 7800               743  BXH       BAL   BXLH(MR6)            COMMON ROUTINE (P.18)               83207390
181      13F0 0D9C               744            BALC  BC3(NULL),IR,D       (P.3)                               83207400
                                 745  *                                                      * C1 *       83207410
182      12D8 7800               746  BXLE      BAL   BXLH(MR6)            COMMON ROUTINE (P.18)               83207420
183      17F0 0D9C               747            BALNC BC3(NULL),IR,D       (P.3)                               83207430
                                 748  *                                                      * C2 *       83207440
184      12D8 BC00               749  LPSW      BAL   RDFULL(MR6)          FETCH FW PSW (P.31)                 83207450
185      13F8 FB80               750            BAL   LPSW1(NULL)          (P.43)                              83207460
                                 751  *                                                      * C3 *       83207470
186      2A1A 1D89               752  THI       A     MR0,YX,MDR,ILIR                                          83207480
187      2BF9 5830               753            N     NULL,YD,MR0,E,D                                          83207490
                                 754  *                                                      * C4 *       83207500
188      2A1A 1D89               755  NHI       A     MR0,YX,MDR,ILIR                                          83207510
189      2B39 5830               756            N     YD,YD,MR0,E,D                                            83207520
                                 757  *                                                      * C5 *       83207530
18A      2A1A 1D89               758  CLHI      A     MR0,YX,MDR,ILIR                                          83207540
18B      2BF9 0830               759            S     NULL,YD,MR0,E,D                                          83207550
                                 760  *                                                      * C6 *       83207560
18C      2A1A 1D89               761  OHI       A     MR0,YX,MDR,ILIR                                          83207570
18D      2B39 7830               762            O     YD,YD,MR0,E,D                                            83207580
                                 763  *                                                      * C7 *       83207590
18E      2A1A 1D89               764  XHI       A     MR0,YX,MDR,ILIR                                          83207600
18F      2B39 6830               765            X     YD,YD,MR0,E,D                                            83207610
                                 766  *                                                      * C8 *       83207620
190      2A1A 1D89               767  LHI       A     MR0,YX,MDR,ILIR                                          83207630
191      2B3F 1830               768            L     YD,MR0,E,D                                               83207640
                                 769  *                                                      * C9 *       83207650
192      2B7A 1D80               770  CHI       A     MDR,YX,MDR                                               83207660
193      13F8 04C0               771            BAL   C1(NULL)             (P.2)                               83207670
                                 772  *                                                      * CA *       83207680
194      2A1A 1D89               773  AHI       A     MR0,YX,MDR,ILIR                                          83207690
195      2B39 1830               774            A     YD,YD,MR0,E,D                                            83207700
                                 775  *                                                      * CB *       83207710
196      2A1A 1D89               776  SHI       A     MR0,YX,MDR,ILIR                                          83207720
197      2B39 0830               777            S     YD,YD,MR0,E,D                                            83207730
                                 778  *                                                      * CC *       83207740
198      2A1A 1D89               779  SRHL      A     MR0,YX,MDR,ILIR                                          83207750
199      2B39 8870               780            SRHL  YD,YD,MR0,E,D                                            83207760
                                 781  *                                                      * CD *       83207770
19A      2A1A 1D89               782  SLHL      A     MR0,YX,MDR,ILIR                                          83207780
19B      2B39 9870               783            SLHL  YD,YD,MR0,E,D                                            83207790
                                 784  *                                                      * CE *       83207800
19C      2A1A 1D89               785  SRHA      A     MR0,YX,MDR,ILIR                                          83207810
19D      2B39 C870               786            SRHA  YD,YD,MR0,E,D                                            83207820
                                 787  *                                                      * CF *       83207830
19E      2A1A 1D89               788  SLHA      A     MR0,YX,MDR,ILIR                                          83207840
19F      2B39 D870               789            SLHA  YD,YD,MR0,E,D                                            83207850
```

```
                                    791  *                                                  * D0 *          83207870
1A0      12D8 1B00                  792  STM      BAL   CADRS(MR6)          CALCULATE ADDRESS (P.5)                 83207880
1A1      13F8 40C0                  793           BAL   STM1(NULL)          (P.10)                                  83207890
                                    794  *                                                  * D1 *          83207900
1A2      12D8 BC00                  795  LM       BAL   RDFULL(MR6)         GET 1ST REGISTER (P.31)                 83207910
1A3      13F8 4240                  796           BAL   LM1(NULL)           (P.10)                                  83207920
                                    797  *                                                  * D2 *          83207930
1A4      12D8 BA80                  798  STB      BAL   RDHALF(MR6)         FETCH HALFWORD (P.31)                   83207940
1A5      13F8 5AC0                  799           BAL   STB1(NULL)          (P.13)                                  83207950
                                    800  *                                                  * D3 *          83207960
1A6      12D8 BA80                  801  LB       BAL   RDHALF(MR6)         FETCH HALFWORD (P.31)                   83207970
1A7      13F8 5BC0                  802           BAL   LB1(NULL)           (P.13)                                  83207980
                                    803  *                                                  * D4 *          83207990
1A8      12D8 BA80                  804  CLB      BAL   RDHALF(MR6)         FETCH HALFWORD (P.31)                   83208000
1A9      13F8 5C80                  805           BAL   CLB1(NULL)          (P.13)                                  83208010
                                    806  *                                                  * D5 *          83208020
1AA      12D8 1B00                  807  AL       BAL   CADRS(MR6)          CALCULATE ADDRESS (P.5)                 83208030
1AB      13F8 5580                  808           BAL   AL1(NULL)           (P.13)                                  83208040
                                    809  *                                                  * D6 *          83208050
1AC      12D8 BC00                  810  WB       BAL   RDFULL(MR6)         GET ADRS OF LIMITS (P.31)               83208060
1AD      13F8 5100                  811           BAL   WB1(NULL)           (P.12)                                  83208070
                                    812  *                                                  * D7 *          83208080
1AE      12D8 BC00                  813  RB       BAL   RDFULL(MR6)         GET ADRS OF LIMITS (P.31)               83208090
1AF      13F8 5180                  814           BAL   RB1(NULL)           (P.12)                                  83208100
                                    815  *                                                  * D8 *          83208110
1B0      12D8 BA80                  816  WH       BAL   RDHALF(MR6)         FETCH HALFWORD (P.31)                   83208120
1B1      4BF9 D0B9                  817           WHA   NULL,YD,MDR,ILIR,E,D                                        83208130
                                    818  *                                                  * D9 *          83208140
1B2      12D8 5E80                  819  RH       BAL   STORE(MR6)          COMMON ROUTINE (P.14)                   83208150
1B3      4B79 CFA3                  820           RHA   MDR,YD,NULL,DW2,E   EXECUTED INSTR.                         83208160
                                    821  *                                                  * DA *          83208170
1B4      12D8 BA80                  822  WD       BAL   RDHALF(MR6)         FETCH HALFWORD (P.31)                   83208180
1B5      13F8 5D80                  823           BAL   WD1(NULL)           (P.13)                                  83208190
                                    824  *                                                  * DB *          83208200
1B6      12D8 BA80                  825  RD       BAL   RDHALF(MR6)         FETCH HALFWORD (P.31)                   83208210
1B7      4B79 8DA3                  826           RDA   MDR,YD,MDR,DW2,E                                            83208220
1B8      2BFF 1F99                  827           L     NULL,NULL,ILIR,D                                            83208230
1B9      0000 0000                  828           DC    0                                                           83208240
                                    829  *                                                                          83208250
                                    830  *                                                  * DD *          83208260
1BA      12D8 BA80                  831  SS       BAL   RDHALF(MR6)         FETCH HALFWORD (P.31)                   83208270
1BB      13F8 5E00                  832           BAL   SS1(NULL)           (P.13)                                  83208280
                                    833  *                                                  * DE *          83208290
1BC      12D8 BA80                  834  OC       BAL   RDHALF(MR6)         FETCH HALFWORD (P.31)                   83208300
1BD      4BF9 BDA0                  835           OCA   NULL,YD,MDR,E                                               83208310
1BE      2BFF 1F99                  836           L     NULL,NULL,ILIR,D                                            83208320
1BF      0000 0000                  837           DC    0                                                           83208330
```

```
                              839  *                                                     * E0 *       83208350
1C0      12D8 1B00            840  TS       BAL   CADRS(MR6)          CALCULATE ADDRESS (P.5)               83208360
1C1      239F 183A            841           LX    MAR,MR0,TS1         MAR=EFFECTIVE ADRS (P.18)             83208370
                              842  *                                                     * E1 *       83208380
1C2      2B9A 1D81            843  SVC      A     MAR,YX,MDR,IL       CALCULATE EFFECTIVE ADDRESS           83208390
1C3      13F8 B3C0            844           BAL   SVC1(NULL)          (P.29)                                83208400
                              845  *                                                     * E2 *       83208410
1C4      2A3A 1D80            846  SINT     A     DEV,YX,MDR          DEV = I2+(X2)                         83208420
1C5      13F8 B2C0            847           BAL   SINT1(NULL)         (P.29)                                83208430
                              848  *                                                     * E3 *       83208440
1C6      12D8 1B00            849  SCP      BAL   CADRS(MR6)          CALCULATE ADDRESS (P.5)               83208450
1C7      2B9F 1800            850           L     MAR,MR0             ADDRESS CCW                           83208460
1C8      13F8 EC00            851           BAL   SCP1(NULL)          (P.41)                                83208470
                              852  *                                                                  83208480
1C9      2259 1F91            853  BXLH1    AX    MR2,YD,NULL,BXLH2   (MR2)=COMPARAND                       83208490
                              854  *                                                     * E5 *       83208500
1CA      12D8 1B00            855  BDCS     BAL   CADRS(MR6)          CALCULATE ADDRESS (P.5)               83208510
1CB      03F8 0800            856           BAL   (MR0)(NULL)         DO BRANCH                             83208520
                              857  *                                                     * E6 *       83208530
1CC      2B9A 1D81            858  LA       A     MAR,YX,MDR,IL       CALCULATE ADRS & INCREMENT LOC        83208540
1CD      13F8 0B80            859           BAL   LA1(NULL)           (P.3)                                 83208550
                              860  *                                                     * E7 *       83208560
1CE      2B9A 1D8F            861  TLATE    A     MAR,YX,MDR,DR4      FETCH TABLE ADDRESS                   83208570
1CF      13F8 DD40            862           BAL   TLATE1(NULL)        (P.38)                                83208580
                              863  *                                                     * E8 *       83208590
1D0      13FD 2580            864  CCS      BALA  CCS1(NULL)          (P.50)                                83208600
1D1      23DF 1813            865  BXLH2    LX    YDI,MR0,BXLH3       POINT BACK TO R1                      83208610
                              866  *                                                     * E9 *       83208620
1D2      13F9 2180            867  ECS      BAL   ECS1(NULL)          (P.50)                                83208630
1D3      233F 18B8            868  BXLH3    LX    YD,MR1,BXLH4        INDEX TO R1 (P.18)                    83208640
                              869  *                                                     * EA *       83208650
1D4      2A1A 1D89            870  RRL      A     MR0,YX,MDR,ILIR                                           83208660
1D5      2B39 A830            871           RR    YD,YD,MR0,E,D                                             83208670
                              872  *                                                     * EB *       83208680
1D6      2A1A 1D89            873  RLL      A     MR0,YX,MDR,ILIR                                           83208690
1D7      2B39 B830            874           RL    YD,YD,MR0,E,D                                             83208700
                              875  *                                                     * EC *       83208710
1D8      2A1A 1D89            876  SRL      A     MR0,YX,MDR,ILIR                                           83208720
1D9      2B39 8830            877           SRL   YD,YD,MR0,E,D                                             83208730
                              878  *                                                     * ED *       83208740
1DA      2A1A 1D89            879  SLL      A     MR0,YX,MDR,ILIR                                           83208750
1DB      2B39 9830            880           SLL   YD,YD,MR0,E,D                                             83208760
                              881  *                                                     * EE *       83208770
1DC      2A1A 1D89            882  SRA      A     MR0,YX,MDR,ILIR                                           83208780
1DD      2B39 C830            883           SRA   YD,YD,MR0,E,D                                             83208790
                              884  *                                                     * EF *       83208800
1DE      2A1A 1D89            885  SLA      A     MR0,YX,MDR,ILIR                                           83208810
1DF      2B39 D830            886           SLA   YD,YD,MR0,E,D                                             83208820
```

```
                            888  * COMMON SUBROUTINE FOR BXH,BXLE                                         83208840
                            889  *                                                                        83208850
1E0     2A39 1F80           890  BXLH     A      MR1,YD,NULL          MR1 = INDEX                         83208860
1E1     2A1F 1F01           891           L      MR0,YDI,IL           INCREMENT LOC, SAVE R1 FIELD        83208870
1E2     33D0 1001           892           AI     YDI,MR0,1            POINT TO R1+1                       83208880
1E3     2A39 1880           893           A      MR1,YD,MR1           INDEX PLUS INCREMENT                83208890
1E4     33D0 1002           894           AI     YDI,MR0,2            POINT TO R1+2                       83208900
1E5     239A 1D89           895           AX     MAR,YX,MDR,BXLH1     (MAR)=BRANCH ADDRESS (P.17)         83208910
                            896  *                                                                        83208920
                            897  *                                                          * F3 *        83208930
1E6     2A1A 1D89           898  TI       A      MR0,YX,MDR,ILIR                                          83208940
1E7     2BF9 5830           899           N      NULL,YD,MR0,E,D                                          83208950
                            900  *                                                          * F4 *        83208960
1E8     2A1A 1D89           901  NI       A      MR0,YX,MDR,ILIR                                          83208970
1E9     2B39 5830           902           N      YD,YD,MR0,E,D                                            83208980
                            903  *                                                          * F5 *        83208990
1EA     2A1A 1D89           904  CLI      A      MR0,YX,MDR,ILIR                                          83209000
1EB     2BF9 0830           905           S      NULL,YD,MR0,E,D                                          83209010
                            906  *                                                          * F6 *        83209020
1EC     2A1A 1D89           907  OI       A      MR0,YX,MDR,ILIR                                          83209030
1ED     2B39 7830           908           O      YD,YD,MR0,E,D                                            83209040
                            909  *                                                          * F7 *        83209050
1EE     2A1A 1D89           910  XI       A      MR0,YX,MDR,ILIR                                          83209060
1EF     2B39 6830           911           X      YD,YD,MR0,E,D                                            83209070
                            912  *                                                          * F8 *        83209080
1F0     2A1A 1D89           913  LI       A      MR0,YX,MDR,ILIR                                          83209090
1F1     2B3F 1830           914           L      YD,MR0,E,D                                               83209100
                            915  *                                                          * F9 *        83209110
1F2     2B7A 1D80           916  CI       A      MDR,YX,MDR                                               83209120
1F3     13F8 04C0           917           BAL    C1(NULL)             (P.2)                               83209130
                            918  *                                                          * FA *        83209140
1F4     2A1A 1D80           919  AI       A      MR0,YX,MDR                                               83209150
1F5     2B39 1839           920           A      YD,YD,MR0,ILIR,E,D                                       83209160
                            921  *                                                          * FB *        83209170
1F6     2A1A 1D80           922  SI       A      MR0,YX,MDR                                               83209180
1F7     2B39 0839           923           S      YD,YD,MR0,ILIR,E,D                                       83209190
                            924  *                                                                        83209200
                            925  *                                                                        83209210
1F8     2BF2 0880           926  BXLH4    S      NULL,MR2,MR1         SUBTRACT TO COMPARE                 83209220
1F9     03F8 0B00           927           BAL    (MR6)(NULL)          RETURN TO CALL                      83209230
                            928  *                                                                        83209240
                            929  *                                                                        83209250
1FA     2B7F 1F88           930  TS1      L      MDR,NULL,RAS         READ HALFWORD & SET BIT             83209260
1FB     2BFF 1DA0           931           L      NULL,MDR,E           TEST IT                             83209270
1FC     17E5 3559           932           BALNL  TS2(NULL),ILIR,D     TO TS2 IF NOT MINUS (P.53)          83209280
                            933  *                                                                        83209290
1FD     0000 0000           934           DC     0                                                        83209300
```

```
                               936  *         M A C   I N T E R R U P T          *                 83209320


1FE     323F 1090              938  MACINT  LI    MR1,'90'          ADRS OF MAC INTERRUPT NEW PSW        83209340
1FF     1378 8280              939          BAL   COMINT(MDR)       TO COMMON ROUTINE (P.21)             83209350


                               941  * ON INSTRUCTION READS, IF ENABLED BY PSW BIT 21,                  83209370
                               942  * THE OCCURANCE OF INVALID ADDRESS, NON-PRESENT                    83209380
                               943  * ADDRESS OR EXECUTE PROTECT JAMS 'FF' INTO THE                    83209390
                               944  * OP-CODE FIELD OF IR, CAUSING THE VECTOR TO '1FE'.                83209400
```

```
200                          946       ORG   '200'                                            83209420
                             947  *      I N T E R R U P T   V E C T O R S      *             83209430


200     177C AB80            949       BALD  IOINT3(MDR)        I/O INTERRUPT LEVEL 3 (P.27)  83209450
201     177C AB00            950       BALD  IOINT2(MDR)        I/O INTERRUPT LEVEL 2 (P.27)  83209460
202     177C AA80            951       BALD  IOINT1(MDR)        I/O INTERRUPT LEVEL 1 (P.27)  83209470
203     177C AA40            952       BALD  IOINT0(MDR)        I/O INTERRUPT LEVEL 0 (P.27)  83209480
204     177C 90C0            953       BALD  CONSER(MDR)        CONSOLE ATTENTION (P.23)      83209490
205     177C 84C0            954       BALD  MMFINT(MDR)        MACHINE MALFUNCTION (P.21)    83209500
206     177C 88C0            955       BALD  PPFINT(MDR)        PRIMARY POWER FAIL (P.22)     83209510
207     177C 7F80            956       BALD  MACINT(MDR)        MEMORY ACCESS CONTROL (P.19)  83209520
208     177C 8240            957       BALD  ILEGAL(MDR)        ILLEGAL INSTRUCTION (P.21)    83209530


                             959  * GENERAL REGISTER ASSIGNMENTS                              83209550
                             960  *                                                           83209560
    0000                     961  R0       EQU   0                                            83209570
    0001                     962  R1       EQU   1                                            83209580
    0002                     963  R2       EQU   2                                            83209590
    0003                     964  R3       EQU   3                                            83209600
    0004                     965  R4       EQU   4                                            83209610
    000D                     966  R13      EQU   13                                           83209620
    000E                     967  R14      EQU   14                                           83209630
    000F                     968  R15      EQU   15                                           83209640
```

```
                                970  *     I L L E G A L   I N S T R U C T I O N     *            83209660


209     323F 1030               972  ILEGAL   LI    MR1,'30'             ADRS OF ILLEGAL INSTR. NEW PSW        83209680


20A     321F 1000               974  COMINT   LI    MR0,0                MR0=CC TO OR IN                       83209700
20B     2B9F 1880               975  COMIN0   L     MAR,MR1                                                    83209710
20C     2A5D 1F8E               976           A     MR2,PSW,NULL,PR4     SAVE CURRENT PSW                      83209720
20D     3391 1004               977  COMIN1   AI    MAR,MR1,4                                                  83209730
20E     2BBF 1D8E               978           L     PSW,MDR,PR4          LOAD NEW PSW                          83209740
20F     29DF 1900               979  COMIN2   L     R14,MR2              REG 14 = OLD PSW                      83209750
210     29FF 1D00               980           L     R15,LOC              REG 15 = OLD LOC                      83209760
211     2BBD 7800               981           O     PSW,PSW,MR0          SET CONDITION CODE                    83209770
212     235F 1DA0               982           LX    LOC,MDR,TWAIT        LOAD NEW LOC                          83209780
                                983  *                                                                         83209790
                                984  * GO TO TEST WAIT BIT OF NEW PSW                                          83209800


                                986  *     M A C H I N E   M A L F U N C T I O N     *                        83209820


213     321F 1000               988  MMFINT   LI    MR0,0                                                      83209840
214     339F 1020               989  MMF1     LI    MAR,'20'                                                   83209850
215     2B7D 1F86               990           A     MDR,PSW,NULL,PW4                                           83209860
216     339F 1024               991           LI    MAR,'24'                                                   83209870
217     2B7F 1D06               992           L     MDR,LOC,PW4          SAVE LOC                              83209880
218     339F 1038               993           LI    MAR,'38'                                                   83209890
219     2BFF 1F8E               994           L     NULL,NULL,PR4        FETCH NEW PSW                         83209900
21A     339F 103C               995           LI    MAR,'3C'                                                   83209910
21B     2BBF 1D80               996           L     PSW,MDR              LOAD NEW PSW                          83209920
21C     4AFF 7BAE               997           SMCR  MR7,MR7,PR4,E        JAM MCR 12:15 TO CC                   83209930
21D     4AFF 7BC0               998           CMCR  MR7,MR7              CLEAR MCR BITS                        83209940
21E     2BBD 7800               999           O     PSW,PSW,MR0          SET C FLAG TO INDICATE                83209950
21F     2B5F 1D80              1000           L     LOC,MDR              MALF DURING AUTO DRIVER               83209960


220     4BFF 6FC0              1002  TWAIT    LWFF  NULL,NULL            RESET WAIT INDICATOR                  83209980
221     37FD 5327              1003           NI    NULL,PSW,BIT16,I     TEST PSW WAIT BIT                     83209990
222     17E0 AFDC              1004           BALNZ WAIT(NULL),IR,D      (P.28)                                83210000
```

```
                              1006  *        P R I M A R Y   P O W E R   F A I L      *                   83210020


223     339F 1084             1008  PPFINT   LI    MAR,'84'            ADRS OF CURRENT PSW SAVE POINTER       83210040
224     2BDF 1F8A             1009           L     YDI,NULL,PR2        FETCH POINTER                          83210050
225     361F 1017             1010           LI    MR0,LOHALF,I                                               83210060
226     2A10 5D80             1011           N     MR0,MR0,MDR         USE LS 16 BITS                         83210070
227     2B9F 1800             1012           L     MAR,MR0                                                    83210080
228     2B7D 1F86             1013           A     MDR,PSW,NULL,PW4    SAVE PSW                               83210090
229     3390 1004             1014           AI    MAR,MR0,4           INCREMENT                              83210100
22A     2B7F 1D06             1015           L     MDR,LOC,PW4         SAVE LOC                               83210110
22B     339F 1086             1016           LI    MAR,X'86'           ADRS OF REGISTER SAVE POINTER          83210120
22C     2BBF 1F8A             1017           L     PSW,NULL,PR2        SELECT REGISTER SET 0                  83210130
22D     2A1F 1D80             1018           L     MR0,MDR                                                    83210140
22E     3610 5017             1019           NI    MR0,MR0,LOHALF,I    USE LS 16 BITS                         83210150
22F     3390 0004             1020           SI    MAR,MR0,4           TEMPORARY DECREMENT                    83210160
230     323F 000F             1021           SI    MR1,NULL,15         MR1='FFFFFFF1'                         83210170
231     4AFF 7F80             1022           SMCR  MR7,NULL                                                   83210180
                              1023  *                                                                         83210190
232     2B7F 1C85             1024  STRLP    L     MDR,YD,I4DW4                                               83210200
233     23D1 1F72             1025           AX    YDI,MR1,YDI,STRLP,C                                        83210210
234     33BD 1010             1026           AI    PSW,PSW,'010'       INCREMENT REGISTER SET NUMBER          83210220
                              1027  *                                  TEST IF 2 OR 8 REGISTER SETS           83210230
235     33F7 5040             1028           NI    NULL,MR7,'40'       MCR BIT 9 = 0 IF 2 SETS                83210240
236     17E0 8E00             1029           BALNZ *+2(NULL)           MCR BIT 9 = 1 IF 8 SETS                83210250
237     33BD 7070             1030           OI    PSW,PSW,'70'        IF ZERO, FORCE LAST SET                83210260
238     33FD 5080             1031           NI    NULL,PSW,'80'       TEST IF LAST SET STORED                83210270
239     13E0 8C80             1032           BALZ  STRLP(NULL)         LOOP UNTIL ALL SETS                    83210280
                              1033  *                                                                         83210290
23A     37F7 532B             1034  PPFSTD   NI    NULL,MR7,BIT20,I    TEAT MCR BIT 4                         83210300
23B     17E0 9080             1035           BALNZ PWRDWN(NULL)        SKIP IF NO DFU                         83210310
23C     CB7F 9C85             1036  STRDLP   RRD   MDR,YD,I4DW4        STORE 32 BITS                          83210320
23D     23D1 1F7C             1037           AX    YDI,MR1,YDI,STRDLP,C INCREMENT TO NEXT 32 BITS             83210330
                              1038  *                                                                         83210340
23E     339F 0004             1039  PPFSTE   SI    MAR,NULL,4          MAR='FFFFFFFC'                         83210350
23F     323F 000E             1040           SI    MR1,NULL,14         MR1='FFFFFFF2'                         83210360
                              1041  *                                                                         83210370
240                           1042           IFNZ  DFU                                                        83210380
240     CB7F 1C85             1043  STREE    RRE   MDR,YD,I4DW4        SAVE FLOATING POINT REGISTER           83210390
                              1046           ENDC                                                             83210420
241     23D1 1F40             1047           AX    YDI,MR1,YDI,STREE,C                                        83210430
                              1048  *                                                                         83210440
242     4B7F FF80             1049  PWRDWN   POW   MDR,NULL            POWER DOWN                             83210450
```

```
                                1051  *       C O N S O L E   S E R V I C E           *                     83210470


243      323F 1001              1053  CONSER    LI    MR1,1               DISPLAY DEVICE NUMBER                 83210490
244      321F 1080              1054            LI    MR0,'80'                                                  83210500
245      4BF1 B840              1055            OCRA  NULL,MR1,MR0        OUTPUT COMMAND NORMAL MODE            83210510
246      4BFF 6FC0              1056            LWFF  NULL,NULL           RESET WAIT INDICATOR                  83210520
247      4A51 AFC0              1057            SSRA  MR2,MR1,NULL        ADDRESS & SENSE STATUS                83210530
248      3252 7F00              1058            OI    MR2,MR2,'F00'       EXTEND BIT 0 OF STATUS BYTE           83210540
249      3252 6080              1059            XI    MR2,MR2,'80'                                              83210550
24A      3252 1080              1060            AI    MR2,MR2,'80'                                              83210560
24B      3252 B001              1061            RLI   MR2,MR2,1           8 BIT ROTATE LEFT                     83210570
24C      339F 1028              1062            LI    MAR,'28'                                                  83210580
24D      2B7F 1F8A              1063            L     MDR,NULL,PR2        COLLECT PREVIOUS STATUS               83210590
24E      2A9F 1D80              1064            L     MR4,MDR             SAVE ALL 16 BITS IN MR4               83210600
24F      4B72 4DC2              1065            STBR  MDR,MR2,MDR,PW2     SAVE NEW STATUS IN LOW BYTE           83210610
250      33F2 5080              1066            NI    NULL,MR2,'080'      TEST STATUS BIT 1                     83210620
251      17E0 9F80              1067            BALNZ FNDIS(NULL)         FUNCTION OR REGISTER (P.24)           83210630
252      2B9F 1D00              1068            L     MAR,LOC                                                   83210640
253      33F2 5020              1069            NI    NULL,MR2,'020'      TEST STATUS BIT 3                     83210650
254      17E0 968B              1070            BALNZ ADRMW(NULL),DR2     ADDRESS OR MEMORY WRITE               83210660
255      33F2 5040              1071            NI    NULL,MR2,'040'      TEST STATUS BIT 2                     83210670
256      17E0 9780              1072            BALNZ DISMEM(NULL)        MEMORY READ                           83210680
                                1073  *                                                                         83210690
                                1074  * RUN MODE                                                                83210700
                                1075  *                                                                         83210710
257      37BD 5023              1076  CLRWT     NI    PSW,PSW,BIT160,I    RESET PSW 16                          83210720
258      4BFF 6FC0              1077            LWFF  NULL,NULL           RESET WAIT                            83210730
259      17F8 95DC              1078            BDC   CLRWT(NULL),IR,D    EXECUTE NEXT USER INSTR               83210740


                                1080  * ADDRESS OR MEMORY WRITE                                                 83210760
                                1081  *                                                                         83210770
25A      4B7F 0FC0              1082  ADRMW     RDR   MDR,NULL            READ SWITCH REGISTER 12:19            83210780
25B      4B7F 0D80              1083            RD    MDR,MDR             READ SWITCH REGISTER 04:11            83210790
25C      33F2 5040              1084            NI    NULL,MR2,'040'      TEST STATUS BIT 2                     83210800
25D      17E0 9A03              1085            BALNZ ADRS(NULL),DW2      ADDRESS (P.24)                        83210810
                                1086  *                                   ELSE, MEMORY WRITE                    83210820


                                1088  *                                                                         83210840
                                1089  * DISPLAY MEMORY REGISTERS                                                83210850
                                1090  *                                                                         83210860
25E      2A1F 1D80              1091  DISMEM    L     MR0,MDR             LS 16 BITS TO MR0                     83210870
25F      3610 5017              1092            NI    MR0,MR0,LOHALF,I                                          83210880
260      2A3F 1D00              1093            L     MR1,LOC                                                   83210890
261      3351 1002              1094            AI    LOC,MR1,2           INCREMENT LOC                         83210900
262      3231 1002              1095            AI    MR1,MR1,2                                                 83210910
263      3271 9010              1096            SLLI  MR3,MR1,16          POSITION MS 16 BITS                   83210920
264      2A10 7980              1097            O     MR0,MR0,MR3         MR0=DISPLAY BYTES D4,D3,D2,D1         83210930
265      3231 8010              1098            SRLI  MR1,MR1,16          FORM DISPLAY BYTE D5, EQUAL           83210940
```

```
266     3231 7080              1099          OI    MR1,MR1,'080'        TO LOC 0:3 & INDICATOR FOR                 83210950
267     13F8 9BC0              1100          BAL   OUTDIS(NULL)         MEMORY ADDRESS, MEMORY DATA                83210960



                               1102  *                                  ADDRESS                                    83210980
268     2A1F 1D80              1103  ADRS    L     MR0,MDR                                                         83210990
269     3610 5039              1104          NI    MR0,MR0,CFFFE,I      LS 16 BITS, FORCED EVEN                    83211000
26A     3232 500F              1105          NI    MR1,MR2,'0F'         MS 4 BITS IN STATUS BYTE                   83211010
26B     3231 9010              1106          SLLI  MR1,MR1,16           POSITION  BITS                             83211020
26C     2B51 7800              1107          O     LOC,MR1,MR0          LOAD 20 BIT LOCATION COUNT                 83211030
                               1108  *                                                                             83211040
                               1109  *                                                                             83211050
                               1110  * DISPLAY THE LOCATION COUNTER                                                83211060
                               1111  *                                                                             83211070
26D     2A1F 1D00              1112  LOCDIS  L     MR0,LOC              MR0=DISPLAY BYTES D4,D3,D2,D1              83211080
26E     323F 1045              1113          LI    MR1,'45'             MR1=D5=FUNCTION 5                          83211090
                               1114  *                                                                             83211100
                               1115  *                                                                             83211110
                               1116  * OUTPUT DATA TO THE DISPLAY                                                  83211120
                               1117  *                                                                             83211130
26F     4BFF 1840              1118  OUTDIS  WDR   NULL,MR0             DISPLAY BYTE D1                            83211140
270     4BFF 1800              1119          WD    NULL,MR0             DISPLAY BYTE D2                            83211150
271     3210 8010              1120          SRLI  MR0,MR0,16                                                      83211160
272     4BFF 1840              1121          WDR   NULL,MR0             DISPLAY BYTE D3                            83211170
273     4BFF 1800              1122          WD    NULL,MR0             DISPLAY BYTE D4                            83211180
274     4BFF 18C0              1123          WDR   NULL,MR1             DISPLAY BYTE D5: 1N=FLT REG N              83211190
                               1124  *                                                  2N=GEN REG N               83211200
                               1125  *                                                 4N=FUNCTION N               83211210
                               1126  *                                                  8N=ADRS/DATA               83211220
                               1127  *                                                                             83211230
275     321F 0001              1128  IDLE    SI    MR0,NULL,1                                                      83211240
276     4BFF 6840              1129          LWFF  NULL,MR0             SET WAIT INDICATOR                         83211250
277     4AFF 7B80              1130  IDLE1   SMCR  MR7,MR7                                                         83211260
278     1364 88C0              1131          BALL  PPFINT(MDR)          EARLY POWER FAIL (P.22)                    83211270
279     33F7 5020              1132          NI    NULL,MR7,'20'        CATN?                                      83211280
27A     13E0 9DC0              1133          BALZ  IDLE1(NULL)          NO, LOOP                                   83211290
                               1134  *                                                                             83211300
27B     33F7 5100              1135          NI    NULL,MR7,'100'       SINGLE STEP?                               83211310
27C     13E0 90C0              1136          BALZ  CONSER(NULL)         NO (P.23)                                  83211320
27D     13F8 95C0              1137          BAL   CLRWT(NULL)          SINGLE STEP (P.23)                         83211330



                               1139  * FUNCTION OR REGISTER DISPLAY                                                83211350
                               1140  *                                                                             83211360
27E     2BDF 1900              1141  FNDIS   L     YDI,MR2                                                         83211370
27F     33F2 5010              1142          NI    NULL,MR2,'010'       TEST STATUS BIT 4                          83211380
280     13E0 A380              1143          BALZ  FN(NULL)             FUNCTION (P.25)                            83211390
                               1144  *                                                                             83211400
                               1145  * GENERAL REGISTER OR FLOATING REGISTER                                       83211410
                               1146  *                                                                             83211420
281     2A1F 1C80              1147          L     MR0,YD               FETCH GENERAL REGISTER                     83211430
282     323F 1020              1148          LI    MR1,'20'                                                        83211440
```

```
283     2A31 7F00          1149          O     MR1,MR1,YDI                                        83211450
284     33F2 5020          1150          NI    NULL,MR2,'020'      TEST STATUS BIT 3              83211460
285     13E0 9BC0          1151          BALZ  OUTDIS(NULL)        GENERAL REGISTER (P.24)        83211470
286                        1152          IFNZ  DFU                                                83211480
286     CA1F 1C80          1153          RRE   MR0,YD              COLLECT SINGLE PRECISION REG.  83211490
                           1156          ENDC                                                     83211520
287     33F2 5001          1157          NI    NULL,MR2,1          TEST IF ODD # FLT REG.         83211530
288     17E0 A2C0          1158          BALNZ DFLOAT(NULL)        YES, SHOW LOW 32 BITS OF DOUBLE 83211540
289     33F4 5100          1159          NI    NULL,MR4,'100'      SEE IF FN 3 IN AFFECT          83211550
28A     13E0 A300          1160          BALZ  SFLOAT(NULL)        SINGLE PRECISION IF NO         83211560
28B     CA1F 9C80          1161 DFLOAT   RRD   MR0,YD              GET 32 BITS OF A DOUBLE REGISTER 83211570
28C     3231 0010          1162 SFLOAT   SI    MR1,MR1,'10'        ADJUST D5                      83211580
28D     13F8 9BC0          1163          BAL   OUTDIS(NULL)        (P.24)                         83211590


                           1165 * FUNCTION                                                        83211610
                           1166 *                                                                 83211620
28E     33F2 5008          1167 FN       NI    NULL,MR2,'008'      TEST STATUS BIT 5              83211630
28F     17E0 9B40          1168          BALNZ LOCDIS(NULL)        DEFAULT TO FN5 IF FN 8:F (P.24) 83211640
290     33F2 5004          1169          NI    NULL,MR2,'004'                                     83211650
291     13E0 A5C0          1170          BALZ  FN0123(NULL)        FUNCTION 0,1,2 OR 3            83211660
                           1171 *                                                                 83211670
                           1172 * FUNCTION 4 OR 5                                                 83211680
                           1173 *                                                                 83211690
292     33F2 5001          1174          NI    NULL,MR2,'001'                                     83211700
293     17E0 9B40          1175          BALNZ LOCDIS(NULL)        FUNCTION 5 = LOC (P.24)        83211710
                           1176 *                                                                 83211720
                           1177 * FUNCTION 4 = PSW                                                83211730
                           1178 *                                                                 83211740
294     2A1D 1F80          1179 PSWDIS   A     MR0,PSW,NULL        LOAD PSW                       83211750
295     323F 1044          1180          LI    MR1,'44'            FUNCTION 4                     83211760
296     13F8 9BC0          1181          BAL   OUTDIS(NULL)        (P.24)                         83211770
                           1182 *                                                                 83211780
                           1183 * FUNCTION 0,1,2 OR 3                                             83211790
                           1184 *                                                                 83211800
297     4AFF 7B80          1185 FN0123   SMCR  MR7,MR7             SENSE MACHINE CONTROL REGISTER 83211810
298     33F7 5100          1186          NI    NULL,MR7,'100'      TEST SNGL FLAG IN MCR          83211820
299     17E0 9B40          1187          BALNZ LOCDIS(NULL)        DEFAULT TO FN5 IF SNGL         83211830
29A     33F2 5002          1188          NI    NULL,MR2,2          TEST STATUS BIT 7              83211840
29B     13E0 A7C0          1189          BALZ  FN01(NULL)          FUNCTION 0 OR 1                83211850
                           1190 *                                                                 83211860
29C     339F 1028          1191          LI    MAR,'28'                                           83211870
29D     4B72 CDC2          1192          STB   MDR,MR2,MDR,PW2     FN 2 OR 3, UPDATE HIGH BYTE    83211880
29E     13F8 9B40          1193          BAL   LOCDIS(NULL)        OF LOCATION '00028'  (P.24)    83211890
                           1194 *                                                                 83211900
29F     33F2 5001          1195 FN01     NI    NULL,MR2,'001'                                     83211910
2A0     13E0 A940          1196          BALZ  FN0(NULL)           FUNCTION 0 (P.26)              83211920
                           1197 *                                                                 83211930
                           1198 * FUNCTION 1 = SELECT PSW BITS                                    83211940
                           1199 *                                                                 83211950
2A1     37BD 5015          1200          NI    PSW,PSW,HIHALF,I    CLEAR LS 16 BITS               83211960
2A2     4A5F 0FC0          1201          RDR   MR2,NULL            INPUT LS BYTE                  83211970
2A3     4A5F 0900          1202          RD    MR2,MR2             INPUT MS BYTE                  83211980
```

```
2A4     23BD 7914          1203           OX     PSW,PSW,MR2,PSWDIS  OR INTO PSW THEN DISPLAY (P.25)       83211990
                           1205  *                                                                          83212010
                           1206  * FUNCTION 0 = CONSOLE INTERRUPT                                           83212020
                           1207  *                                                                          83212030
2A5     37FD 5021          1208  FN0      NI     NULL,PSW,BI1720,I   TEST PSW BIT 17 OR 20                 83212040
2A6     13E0 95C0          1209           BALZ   CLRWT(NULL)         IGNORE IF NO ENABLE (P,23)            83212050
2A7     323F 1001          1210           LI     DEV,1               DEVICE NUMBER 1                       83212060
2A8     225F 1FB0          1211           LX     LEVEL,NULL,IOINTX   REGISTER SET 0 (P,27)                 83212070
                           1212  *                                                                          83212080
                           1213  * IMMEDIATE INTERRUPT                                                      83212090
```

```
                               1215  *   I / O   D E V I C E   I N T E R R U P T S   *                  83212110


                               1217  * REGISTER ASSIGNMENTS FOR CHANNEL I/O                              83212130
                               1218  *                                                                  83212140
    0010                       1219  TEMP     EQU   '10'                                                83212150
    0011                       1220  DEV      EQU   '11'                                                83212160
    0012                       1221  LEVEL    EQU   '12'                                                83212170
    0012                       1222  CCW      EQU   '12'                                                83212180
    0013                       1223  DAT      EQU   '13'                                                83212190
    0014                       1224  COUNT    EQU   '14'                                                83212200
    0015                       1225  RETURN   EQU   '15'                                                83212210


2A9      225F 1FAF             1227  IOINT0   LX    LEVEL,NULL,AUTOIO   SELECT REGISTER SET 0           83212230
                               1228  *                                                                  83212240
2AA      325F 1011             1229  IOINT1   LI    LEVEL,'11'          SELECT REGISTER SET 1           83212250
2AB      423F 6930             1230           AKX   DEV,LEVEL,IOINTX    ACKNOWLEDGE INTERRUPT           83212260
                               1231  *                                                                  83212270
2AC      325F 1022             1232  IOINT2   LI    LEVEL,'22'          SELECT REGISTER SET 2           83212280
2AD      423F 6930             1233           AKX   DEV,LEVEL,IOINTX    ACKNOWLEDGE INTERRUPT           83212290
                               1234  *                                                                  83212300
2AE      325F 1033             1235  IOINT3   LI    LEVEL,'33'          SELECT REGISTER SET 3           83212310
2AF      4A3F 6900             1236  AUTOIO   AK    DEV,LEVEL           ACKNOWLEDGE INTERRUPT           83212320
                               1237  *                                                                  83212330
                               1238  *                                                                  83212340
2B0      3231 53FF             1239  IOINTX   NI    DEV,DEV,'3FF'       MASK 10 BIT DEVICE NUMBER       83212350
2B1      3271 10D0             1240           AI    DAT,DEV,'D0'        2X DEVICE NUMBER + 'D0'         83212360
2B2      2B93 188A             1241           A     MAR,DAT,DEV,PR2     INDEXES SERVICE POINTER TABLE   83212370
                               1242  *                                  FETCH APPROPRIATE ENTRY         83212380
                               1243  *                                                                  83212390
2B3      2A1D 1F80             1244           A     TEMP,PSW,NULL       SAVE PSW                        83212400
2B4      3252 50F0             1245           NI    LEVEL,LEVEL,'F0'    MASK REGISTER SET BITS          83212410
2B5      37B2 7131             1246           OI    PSW,LEVEL,BI1820,I  SET PSW BITS 18 & 20 AND        83212420
                               1247  *                                  SELECT REGISTER SET             83212430
                               1248  *                                                                  83212440
2B6      281F 1800             1249           L     R0,TEMP             REG 0 = PSW                     83212450
2B7      283F 1D00             1250           L     R1,LOC              REG 1 = LOC                     83212460
2B8      285F 1880             1251           L     R2,DEV              REG 2 = DEVICE NUMBER           83212470
2B9      4871 AFE0             1252           SSRA  R3,DEV,NULL,E       REG 3 = DEVICE STATUS, SET CC   83212480
                               1253  *                                                                  83212490
2BA      2A7F 1D80             1254           L     DAT,MDR             TABLE ENTRY TO DAT              83212500
2BB      3753 5039             1255           NI    LOC,DAT,CFFFE,I     LOAD LOC JUST IN CASE           83212510
2BC      4BFF 6FC0             1256           LWFF  NULL,NULL           RESET WAIT INDICATOR            83212520
2BD      33F3 5001             1257           NI    NULL,DAT,1          TEST LSB OF SERVICE POINTER     83212530
2BE      17E1 0C9C             1258           BALNZ CHANEL(NULL),IR,D   (P.46)                          83212540
                               1259  *                                  IF SET, SERVICE POINTER IS      83212550
                               1260  *                                  ADDRESS PLUS ONE OF CCB         83212560
                               1261  *                                  IF RESET, SERVICE POINTER IS    83212570
                               1262  *                                  ADDRESS IN FIRST 64K OF A       83212580
                               1263  *                                  USER'S SERVICE SUBROUTINE       83212590
```

```
                              1265  *        I N T E R R U P T A B L E          *              83212610
                              1266  *                                           *              83212620
                              1267  *            W A I T   L O O P              *              83212630


2BF     321F 0001             1269  WAIT     SI    MR0,NULL,1                                   83212650
2C0     4BFF 6840             1270           LWFF  NULL,MR0           SET WAIT INDICATOR        83212660
2C1     13FC B050             1271  WAIT1    BALA  WAIT1(NULL),D      LOOP                      83212670


                              1273  *      A R I T H M E T I C   F A U L T      *              83212690


                              1275  *                                D,DR DIVIDE FAULT          83212710
2C2     2B3F 1800             1276  DFALT0   L     YD,MR0              RESTORE (R1)             83212720
2C3     2BDF 3F00             1277           AINC  YDI,NULL,YDI                                 83212730
2C4     2B3F 1880             1278           L     YD,MR1              RESTORE (R1+1)           83212740
                              1279  *                                  DH,DHR DIVIDE FAULT      83212750
2C5     221F 1F87             1280  DFALT1   LX    MR0,NULL,AFAULT     NO CC                    83212760


2C6                           1282           IFNZ  DFU                                          83212780
      02C6                    1283  FFAULT   EQU   *                                            83212790
                              1284           ENDC                                               83212800
2C6     321F 1008             1285  FFALT1   LI    MR0,8               FLOATING POINT FAULT     83212810
                              1286  *                                  CARRY FLAG WILL BE SET   83212820
                              1287  *                                                           83212830
2C7     323F 1048             1288  AFAULT   LI    MR1,'48'                                     83212840
2C8     37FD 532A             1289           NI    NULL,PSW,BIT19,I    TEST ARITHMETIC FAULT ENABLE  83212850
2C9     17E0 B290             1290           BALNZ *+1(NULL),D         TAKE INTERRUPT IF ENABLED     83212860
                              1291  *                                                           83212870
                              1292  * DO NEXT USER INSTRUCTION IF BIT 19 RESET                  83212880
2CA     1378 82C0             1293           BAL   COMIN0(MDR)         COMMON ROUTINE (P.21)    83212890
```

```
                             1295  *        S I M U L A T E   I N T E R R U P T                     83212910
                             1296  *                                                                83212920
2CB     2A5F 1F01            1297  SINT1    L     LEVEL,YDI,IL       R1 FIELD TO "LEVEL"            83212930
2CC     1360 AC00            1298           BALZ  IOINTX(MDR)        R1 FIELD IS ZERO (P.27)        83212940
2CD     3259 9004            1299           SLLI  LEVEL,YD,4         IF NON-ZERO, SELECT REGISTER SET 83212950
2CE     13F8 AC00            1300           BAL   IOINTX(NULL)       DO I/O INTERRUPT (P.27)        83212960


                             1302  *        S U P E R V I S O R   C A L L                           83212980
                             1303  *                                                                83212990
2CF     2A1F 1E00            1304  SVC1     L     MR0,MAR            SAVE ADDRESS IN MR0            83213000
2D0     339F 1098            1305           LI    MAR,'98'           ADDRESS SVC NEW PSW            83213010
2D1     2B7F 1F8E            1306           L     MDR,NULL,PR4       FETCH NEW PSW STATUS           83213020
2D2     2A5D 1F80            1307           A     MR2,PSW,NULL       SAVE CURRENT PSW               83213030
2D3     2BBF 1D80            1308           L     PSW,MDR            LOAD NEW PSW                   83213040
2D4     29BF 1800            1309           L     R13,MR0            REG 13 = EFFECTIVE ADDRESS     83213050
2D5     29DF 1900            1310           L     R14,MR2            REG 14= PSW                    83213060
2D6     29FF 1D00            1311           L     R15,LOC            REG 15 = LOC                   83213070
2D7     2A3F 1F00            1312           L     MR1,YDI            COLLECT R1 FIELD               83213080
2D8     2A31 1880            1313           A     MR1,MR1,MR1        2X R1 FIELD PLUS X'9C'         83213090
2D9     3391 109C            1314           AI    MAR,MR1,'9C'       IS HALFWORD ADRS               83213100
2DA     2BFF 1F8A            1315           L     NULL,NULL,PR2      OF SVC NEW LOC                 83213110
2DB     363F 1039            1316           LI    MR1,CFFFE,I        MR1 = '0000FFFE'               83213120
2DC     2B51 5D80            1317           N     LOC,MR1,MDR        LOAD NEW LOC                   83213130
2DD     2BFF 1F9C            1318           L     NULL,NULL,IR,D                                    83213140
```

```
                  * COMMON SUBROUTINE FOR TBT, SBT, RBT & CBT     *


2DE   239A 1DE1   COMBIT   AX    MAR,YX,MDR,COMBT1,C TRANSFER IF RX1 OR RX3
2DF   321F 1004            LI    MR0,4
2E0   2210 1E22            AX    MR0,MR0,MAR,COMBT2  ADD 4 IF RX2
2E1   2A1F 1E00   COMBT1   L     MR0,MAR
                  *
                  * MR0 = MATRIX START ADDRESS
                  * R1 CONTAINS DISPLACEMENT TO DESIRED BIT
                  *
2E2   3239 8004   COMBT2   SRLI  MR1,YD,4            ON HALFWORD BOUNDARY,
2E3   2B71 1880            A     MDR,MR1,MR1         ADDRESS A HALFWORD IN
2E4   2B90 1D8B            A     MAR,MR0,MDR,DR2     THE ARRAY & FETCH IT
2E5   3259 500F            NI    MR2,YD,'F'          MASK LS 4 BITS TO TEST
                  *                                  A BIT IN THE HALFWORD
2E6   3252 1327            AI    MR2,MR2,BTABLE      FORM VECTOR ADDRESS
2E7   2E5F 1900            L     MR2,MR2,I           FETCH BIT MASK
2E8   2BF2 5DA0            N     NULL,MR2,MDR,E      TEST THE BIT, SET CC
2E9   03F8 0A80            BAL   (MR5)(NULL)         RETURN
```

```
                                1340  * COMMON ROUTINE CALCULATES EFFECTIVE ADDRESS   *                 83213360
                                1341  * THEN READS HALFWORD FROM MAIN MEMORY          *                 83213370


2EA     239A 1DED               1343  RDHALF   AX    MAR,YX,MDR,RHALF1,C TRANSFER IF RX1 OR RX3        83213390
2EB     321F 1004               1344           LI    MR0,4               ADD 4 IF RX2                  83213400
2EC     2390 1E2E               1345           AX    MAR,MR0,MAR,RHALF2  LOAD ADDRESS                  83213410
2ED     2B9F 1E00               1346  RHALF1   L     MAR,MAR             LOAD ADDRESS                  83213420
2EE     2B7F 1F8B               1347  RHALF2   L     MDR,NULL,DR2        FETCH HALFWORD                83213430
2EF     03F8 0B00               1348           BAL   (MR6)(NULL)         RETURN TO CALL                83213440


                                1350  * COMMON ROUTINE CALCULATES EFFECTIVE ADDRESS   *                 83213460
                                1351  * THEN READS FULLWORD FROM MAIN MEMORY          *                 83213470


2F0     239A 1DF3               1353  RDFULL   AX    MAR,YX,MDR,RFULL1,C TRANSFER IF RX1 OR RX3        83213490
2F1     321F 1004               1354           LI    MR0,4               ADD 4 IF RX2                  83213500
2F2     2390 1E34               1355           AX    MAR,MR0,MAR,RFULL2  LOAD ADDRESS                  83213510
2F3     2B9F 1E00               1356  RFULL1   L     MAR,MAR             LOAD ADDRESS                  83213520
2F4     2B7F 1F8F               1357  RFULL2   L     MDR,NULL,DR4        FETCH FULLWORD                83213530
2F5     03F8 0B00               1358           BAL   (MR6)(NULL)         RETURN TO CALL                83213540


2F6     361F 1017               1360  LHL1     LI    MR0,LOHALF,I        MR0='0000FFFF'                83213560
2F7     2B30 5DB9               1361           N     YD,MR0,MDR,ILIR,E,D                               83213570
                                1362  *                                                                 83213580
2F8     CBF9 BD89               1363  CD1      CDR   YD,MDR,ILIR         COMPARE                       83213590
2F9     CBFF 0FB0               1364           RCC   NULL,NULL,E,D       SET CONDITION CODE            83213600
2FA     0000 0000               1365           DC    0                                                 83213610
2FB     0000 0000               1366           DC    0                                                 83213620
2FC     0000 0000               1367           DC    0                                                 83213630
2FD     0000 0000               1368           DC    0                                                 83213640
2FE     0000 0000               1369           DC    0                                                 83213650
2FF     0000 0000               1370           DC    0                                                 83213660
                                1371  *                                                                 83213670
```

```
300        .                1373          ORG   '300'                                          83213690
                            1374  *        S U B R O U T I N E   A T B L        *              83213700
                            1375  *                                             *              83213710
                            1376  * COMMON SUBROUTINE FOR ATL, ABL              *              83213720
                            1377  *                                             *              83213730
                            1378  * CALL IS:      BAL  ATBL(MR6)               *              83213740


300     239A 1DC3           1380  ATBL    AX    MAR,YX,MDR,ATBL1,C  TRANSFER IF RX1 OR RX3     83213760
301     321F 1004           1381          LI    MR0,4               ADD 4 IF RX2               83213770
302     2210 1E04           1382          AX    MR0,MR0,MAR,ATBL2                              83213780
303     2A1F 1E00           1383  ATBL1   L     MR0,MAR             MR0=LIST START ADRS        83213790
304     2B9F 1800           1384  ATBL2   L     MAR,MR0                                        83213800
305     2B7F 1F8F           1385          L     MDR,NULL,DR4        FETCH MAX SLOT/NO. USED    83213810
306     2A8F 1D80           1386          L     MR5,MDR                                        83213820
307     3275 A010           1387          RRI   MR3,MR5,16          NO. USED/MAX SLOT          83213830
308     2293 0ACD           1388          SX    MR4,MR3,MR5,LSTOVF,C                           83213840
                            1389  *                                                            83213850
                            1390  * LIST NOT FULL                                              83213860
                            1391  *                                                            83213870
309     2B7F 3D87           1392          AINC  MDR,NULL,MDR,DW4    INCREMENT NO.USED          83213880
30A     3293 0001           1393          SI    MR4,MR3,1           MAX SLOT - 1               83213890
30B     2BFF 1F8D           1394          L     NULL,NULL,I4DR4     FETCH CUR.TOP/NEXT BOTTOM  83213900
30C     03F8 0B00           1395          BAL   (MR6)(NULL)         RETURN TO CALL             83213910


30D     33BD 5FF0           1397  LSTOVF  NI    PSW,PSW,'FF0'       CLEAR CONDITION CODE       83213930
30E     33BD 7004           1398          OI    PSW,PSW,4           SET V FLAG                 83213940
30F     2BFF 1F99           1399          L     NULL,NULL,ILIR,D                               83213950
```

```
                               1401  * ADD TO TOP OF LIST                                                       83213970
                               1402  *                                                                         83213980
310     2A5F 1D80              1403  ATL1      L     MR2,MDR                                                     83213990
311     3252 8010              1404            SRLI  MR2,MR2,16           CURRENT TOP                            83214000
312     2252 2FD4              1405            SDECX MR2,MR2,NULL,ATL2,C  DECREMENT BY 1                         83214010
                               1406  *                                    TRANSFER IF NO CARRY                   83214020
                               1407  *                                                                         83214030
                               1408  * LIST WRAP, SET CURRENT TOP TO MAX                                       83214040
                               1409  *                                                                         83214050
313     2A5F 1A00              1410            L     MR2,MR4                                                     83214060
314     3652 5017              1411  ATL2      NI    MR2,MR2,LOHALF,I     REMOVE HI ORDER BITS                   83214070
315     2B7F 1903              1412            L     MDR,MR2,DW2          INSERT UP-DATED POINTER                83214080
                               1413  *                                                                         83214090
316     2A52 1900              1414  ADDIT     A     MR2,MR2,MR2          2X SLOT NUMBER                         83214100
317     2A52 1900              1415            A     MR2,MR2,MR2          4X SLOT NUMBER                         83214110
318     3210 1008              1416            AI    MR0,MR0,8            ADDRESS OF SLOT ZERO                   83214120
319     2B90 1900              1417            A     MAR,MR0,MR2          PLUS 4X SLOT NUMBER                    83214130
31A     2B7F 1C87              1418            L     MDR,YD,DW4           ADD ELEMENT TO LIST                    83214140
31B     2BFF 1FB9              1419            L     NULL,NULL,ILIR,E,D   CLEAR CC & EXIT                        83214150


                               1421  * ADD TO BOTTOM OF LIST                                                   83214170
                               1422  *                                                                         83214180
31C     2A5F 1D80              1423  ABL1      L     MR2,MDR                                                     83214190
31D     3652 5017              1424            NI    MR2,MR2,LOHALF,I     MR2=NEXT BOTTOM POINTER                83214200
31E     2A3F 1900              1425            L     MR1,MR2              SAVE ORIGINAL VALUE                    83214210
31F     3252 1001              1426            AI    MR2,MR2,1            INCREMENT NEXT BOTTOM                  83214220
320     2A74 6880              1427            X     MR3,MR4,MR1          COMPARE ORIGIONAL TO MAX               83214230
321     3673 5017              1428  ABL2      NI    MR3,MR3,LOHALF,I     TEST LS 16 BITS                        83214240
322     17E0 C900              1429            BALNZ ABL3(NULL)           NO WRAP                                83214250
                               1430  *                                                                         83214260
                               1431  * LIST WRAP, SET NEXT BOTTOM TO ZERO                                      83214270
                               1432  *                                                                         83214280
323     325F 1000              1433            LI    MR2,0                                                       83214290
324     3390 1006              1434  ABL3      AI    MAR,MR0,6                                                   83214300
325     2B7F 1903              1435            L     MDR,MR2,DW2          STORE UPDATED POINTER                  83214310
326     225F 1896              1436            LX    MR2,MR1,ADDIT        USE ORIGIONAL VALUE                    83214320
```

```
                                  1438  * BIT TABLE USED BY TBT, SBT, RBT, & CBT                        83214340
                                  1439  *                                                               83214350
      0327                        1440  BTABLE    EQU    *                                              83214360
327     0000 8000                 1441  BIT16     DC     '00008000'                                     83214370
328     0000 4000                 1442  BIT17     DC     '00004000'                                     83214380
329     0000 2000                 1443  BIT18     DC     '00002000'                                     83214390
32A     0000 1000                 1444  BIT19     DC     '00001000'                                     83214400
32B     0000 0800                 1445  BIT20     DC     '00000800'                                     83214410
32C     0000 0400                 1446  BIT21     DC     '00000400'                                     83214420
32D     0000 0200                 1447  BIT22     DC     '00000200'                                     83214430
32E     0000 0100                 1448  BIT23     DC     '00000100'                                     83214440
32F     0000 0080                 1449  BIT24     DC     '00000080'                                     83214450
330     0000 0040                 1450  BIT25     DC     '00000040'                                     83214460
331     0000 0020                 1451  BIT26     DC     '00000020'                                     83214470
332     0000 0010                 1452  BIT27     DC     '00000010'                                     83214480
333     0000 0008                 1453  BIT28     DC     '00000008'                                     83214490
334     0000 0004                 1454  BIT29     DC     '00000004'                                     83214500
335     0000 0002                 1455  BIT30     DC     '00000002'                                     83214510
336     0000 0001                 1456  BIT31     DC     '00000001'                                     83214520
```

```
                            1458  *         S U B R O U T I N E  R T B L          *                        83214540
                            1459  *                                               *                        83214550
                            1460  * COMMON SUBROUTINE FOR RTL,RBL                   *                        83214560
                            1461  *                                               *                        83214570
                            1462  * CALL IS:     BAL  RTBL(MR7)                    *                        83214580


337     239A 1DFA           1464  RTBL     AX    MAR,YX,MDR,RTBL1,C  TRANSFER IF RX1 OR RX3                  83214600
338     321F 1004           1465           LI    MR0,4               ADD 4 IF RX2                            83214610
339     2210 1E3B           1466           AX    MR0,MR0,MAR,RTBL2                                           83214620
33A     2A1F 1E00           1467  RTBL1    L     MR0,MAR             MR0=LIST START ADRS                     83214630
33B     2B9F 1800           1468  RTBL2    L     MAR,MR0                                                     83214640
33C     2B7F 1F8F           1469           L     MDR,NULL,DR4        FETCH MAX SLOT/NO. USED                 83214650
33D     2A7F 1D80           1470           L     MR3,MDR                                                     83214660
33E     3273 A010           1471           RRI   MR3,MR3,16          NO. USED/MAX SLOT                       83214670
33F     3693 000F           1472           SI    MR4,MR3,BIT15,I                                             83214680
340     13F0 C340           1473           BALC  LSTOVF(NULL)        LIST IS EMPTY (P.32)                    83214690
                            1474  *                                                                          83214700
                            1475  * LIST NOT EMPTY                                                           83214710
                            1476  *                                                                          83214720
341     2A9F 1D80           1477           L     MR4,MDR                                                     83214730
342     2B74 2F87           1478           SDEC  MDR,MR4,NULL,DW4    DECREMENT NO.USED                       83214740
343     2A94 2FA0           1479           SDEC  MR4,MR4,NULL,E      ENABLE CC UPDATE                        83214750
344     37F4 5017           1480           NI    NULL,MR4,LOHALF,I   SET CC : G IF NOT EMPTY, ELSE ZERO      83214760
345     2A93 2F8D           1481           SDEC  MR4,MR3,NULL,I4DR4  MAX SLOT - 1                            83214770
346     03F8 0B00           1482           BAL   (MR6)(NULL)         RETURN TO CALL                          83214780
```

```
                      * REMOVE FROM TOP OF LIST                                       83214800
                      *                                                               83214810
347   2A5F 1D80       RTL1      L      MR2,MDR                                        83214820
348   3252 8010                 SRLI   MR2,MR2,16          CURRENT TOP                83214830
349   2A3F 1900                 L      MR1,MR2             ORIGINAL VALUE             83214840
34A   3252 1001                 AI     MR2,MR2,1           INCREMENT BY 1             83214850
34B   2A74 6880                 X      MR3,MR4,MR1         COMPARE TO MAX             83214860
34C   3673 5017                 NI     MR3,MR3,LOHALF,I    TEST LS 16 BITS            83214870
34D   17E0 D3C0                 BALNZ  RTL2(NULL)          NO WRAP                    83214880
                      *                                                               83214890
                      * LIST WRAP, SET CURRENT TOP TO ZERO                            83214900
                      *                                                               83214910
34E   325F 1000                 LI     MR2,0                                          83214920
34F   2B7F 1903       RTL2      L      MDR,MR2,DW2         INSERT UPDATED POINTER     83214930
350   225F 1897                 LX     MR2,MR1,REMOV1      USE ORIGIONAL VALUE        83214940


                      * REMOVE FROM BOTTOM OF LIST                                    83214960
                      *                                                               83214970
351   2A5F 1D80       RBL1      L      MR2,MDR                                        83214980
352   3652 5017                 NI     MR2,MR2,LOHALF,I    NEXT BOTTOM POINTER        83214990
353   2252 2FD5                 SDECX  MR2,MR2,NULL,REMOV,C DECREMENT, XFER IF NO WRAP 83215000
354   3654 5017                 NI     MR2,MR4,LOHALF,I    SET NEXT BOTTOM TO MAX     83215010
                      *                                                               83215020
355   3390 1006       REMOV     AI     MAR,MR0,6                                      83215030
356   2B7F 1903                 L      MDR,MR2,DW2         RESTORE                    83215040
357   3252 9002       REMOV1    SLLI   MR2,MR2,2           4X SLOT NUMBER             83215050
358   3210 1008                 AI     MR0,MR0,8           ADDRESS OF SLOT 0          83215060
359   2B90 190F                 A      MAR,MR0,MR2,DR4     PLUS 4X SLOT NUMBER        83215070
35A   2B3F 1D99                 L      YD,MDR,ILIR,D       LOAD ELEMENT               83215080
```

```
                                    1514  * MULTIPLY HALFWORD                                                             83215100
                                    1515  *                                                                               83215110
35B     3658 7015                   1516  MHR1      OI     MR2,YS,HIHALF,I     SET MS 16 BITS                             83215120
35C     3652 6327                   1517            XI     MR2,MR2,BIT16,I     INVERT SIGN THEN                           83215130
35D     3772 1327                   1518            AI     MDR,MR2,BIT16,I     EXTEND IT                                  83215140
35E     3679 7015                   1519  MH1       OI     MR3,YD,HIHALF,I     SET MS 16 BITS                             83215150
35F     3673 6327                   1520            XI     MR3,MR3,BIT16,I     INVERT THE SIGN                            83215160
360     3673 1327                   1521            AI     MR3,MR3,BIT16,I     EXTEND IT                                  83215170
361     2A53 ED89                   1522            M      MR2,MR3,MDR,ILIR    DO MULTIPLY                                83215180
362     2B3F 1990                   1523            L      YD,MR3,D            LOAD LS 32 BITS                            83215190


                                    1525  * DIVIDE HALFWORD                                                               83215210
                                    1526  *                                                                               83215220
363     3658 7015                   1527  DHR1      OI     MR2,YS,HIHALF,I     SET MS 16 BITS                             83215230
364     3652 6327                   1528            XI     MR2,MR2,BIT16,I     INVERT THE SIGN                            83215240
365     3772 1327                   1529            AI     MDR,MR2,BIT16,I     EXTEND IT                                  83215250
                                    1530  *                                    MDR = DIVISOR                              83215260
366     2A5F 1F80                   1531  DH1       L      MR2,NULL                                                       83215270
367     2A7F 1C80                   1532            L      MR3,YD              MR3=DIVIDEND                               83215280
368     17E4 DA80                   1533            BALNL  *+2(NULL)                                                      83215290
369     325F 0001                   1534            SI     MR2,NULL,1          MR2=SIGN OF DIVIDEND                       83215300
36A     2A53 FD89                   1535            D      MR2,MR3,MDR,ILIR    DO DIVIDE                                  83215310
36B     13F4 B140                   1536            BALV   DFALT1(NULL)        ARITHMETIC FAULT (P.28)                    83215320
36C     3693 5015                   1537            NI     MR4,MR3,HIHALF,I    SAVE MS 16 BITS                            83215330
36D     37F3 5327                   1538            NI     NULL,MR3,BIT16,I    DIVIDE FAULT IF MS 16                      83215340
36E     13E0 DD00                   1539            BALZ   DH3(NULL)           QUOTIENT BITS DON'T EQUAL                  83215350
36F     37F4 6015                   1540            XI     NULL,MR4,HIHALF,I   BIT 16 OF THE QUOTIENT                     83215360
370     17E0 B140                   1541  DH2       BALNZ  DFALT1(NULL)        (P.28)                                     83215370
371     2B3F 1900                   1542            L      YD,MR2              REMAINDER TO R1                            83215380
372     2B0F 3F00                   1543            AINC   YDI,NULL,YDI                                                   83215390
373     2B3F 1990                   1544            L      YD,MR3,D            QUOTIENT TO R1+1                           83215400
                                    1545  *                                                                               83215410
374     23FF 1A30                   1546  DH3       LX     NULL,MR4,DH2                                                   83215420
```

```
                               1548  * TRANSLATE                                                           83215440
                               1549  *                                                                     83215450
375      32D9 50FF             1550  TLATE1    NI    MR6,YD,'FF'         BYTE TO TRANSLATE                 83215460
376      2AD6 1B00             1551            A     MR6,MR6,MR6         2X THE BYTE PLUS ADRS             83215470
377      2B96 1D8B             1552            A     MAR,MR6,MDR,DR2     OF TRANSLATION TABLE              83215480
378      3239 5F00             1553            NI    MR1,YD,'F00'        FETCH HALFWORD ENTRY              83215490
379      2ADF 1D80             1554            L     MR6,MDR                                               83215500
37A      17E4 E1C9             1555            BALNL EXTLAT(NULL),ILIR   EXIT IF NOT NEGATIVE              83215510
37B      32D6 50FF             1556            NI    MR6,MR6,'0FF'       MASK TRANSLATED BYTE              83215520
37C      2B31 7B10             1557            O     YD,MR1,MR6,D        OR INTOR1                         83215530


                               1559  * PART OF AUTO DRIVER CHANNEL                                         83215550
                               1560  *                                                                     83215560
37D      3384 1010             1561  TRANSL    AI    MAR,R4,16           REG 4 = ADRS OF CCB               83215570
37E      26FF 1F8F             1562            L     NULL,NULL,DR4       FETCH ADRS OF TRANSLATION TABLE   83215580
37F      2AD3 1980             1563            A     MR6,DAT,DAT         2X DATA BYTE                      83215590
380      2B96 1D8B             1564            A     MAR,MR6,MDR,DR2     FETCH HALFWORD ENTRY              83215600
381      2ADF 1D80             1565            L     MR6,MDR                                               83215610
382      17E4 E140             1566            BALNL EXTRAN(NULL)        EXIT IF NOT NEGATIVE              83215620
383      3276 50FF             1567            NI    DAT,MR6,'0FF'       MASK TRANSLATED BYTE              83215630
384      03F8 0B80             1568            BAL   (MR7)(NULL)         RETURN TO CALL                    83215640


385      287F 1980             1570  EXTRAN    L     R3,DAT              REG 3 = UN-TRANSLATED CHAR        83215660
386      33BD 5FF0             1571            NI    PSW,PSW,'FF0'       CLEAR CC                          83215670
387      2B56 1B00             1572  EXTLAT    A     LOC,MR6,MR6         2X TABLE ENTRY                    83215680
388      2BFF 1F9C             1573            L     NULL,NULL,IR,D      IS SUBROUTINE ADDRESS             83215690
```

```
                    * CONVERT FLOATING POINT TO FIXED POINT                                   83215710
                    *                                                                         83215720
389     3230 B008   FXR1   RLI   MR1,MR0,8                                                    83215730
38A     3251 5F00          NI    MR2,MR1,'F00'       MR2 = FRACTION LEFT 8                    83215740
    038B            FXDR2  EQU   *                                                            83215750
38B     3231 507F          NI    MR1,MR1,'07F'       MR1 = EXPONENT                           83215760
38C     327F 1048          LI    MR3,'48'                                                     83215770
38D     2233 08D4          SX    MR1,MR3,MR1,FXR4,C  COMPARE EXPONENT TO LIMIT                83215780
                    *                                TRANSFER IF LESS THAN OR                 83215790
                    *                                EQUAL TO '48'. MR1                       83215800
                    *                                CONTAINS DIFFERENCE.                     83215810
                    *                                                                         83215820
38E     365F 113B   OVF1   LI    MR2,ONES,I          OVERFLOW WHEN ARGUMENT EQUAL TO          83215830
                    *                                OR GREATER THAN '48800000', SET          83215840
38F     327F 1004          LI    MR3,4               RESULT TO MAX & QUEUE V FLAG             83215850
                    *                                                                         83215860
                    *                                                                         83215870
390     2210 1852   FXR2   AX    MR0,MR0,MR0,FXR3,C  TEST RESULT SIGN                         83215880
391     2A5F 0900          S     MR2,NULL,MR2        2'S COMPLEMENT IF NEGATIVE               83215890
392     2B3F 1929   FXR3   L     YD,MR2,ILIR,E       LOAD REGISTER, SET CC                    83215900
393     2BBD 7990          O     PSW,PSW,MR3,D       OR IN V FLAG QUEUE                       83215910
                    *                                                                         83215920
                    *                                                                         83215930
394     33F1 0008   FXR4   SI    NULL,MR1,8          COMPARE COUNT TO LIMIT                   83215940
395     13E4 E5C0          BALL  *+2(NULL)           ZERO RESULT IF EXPONENT WAS              83215950
396     225F 1F9A          LX    MR2,NULL,FXR5       LESS THAN '41', ELSE, FORM               83215960
397     3231 9002          SLLI  MR1,MR1,2           HEX SHIFT COUNT FROM DELTA               83215970
398     2A52 8880          SRL   MR2,MR2,MR1         SHIFT INTEGER 0:7 HEX PLACES             83215980
399     13E4 E380          BALL  OVF1(NULL)          BRANCH IF '48800000' OR OVER             83215990
39A     227F 1F90   FXR5   LX    MR3,NULL,FXR2       NO V FLAG QUEUE                          83216000
```

```
                         * LOAD REAL ADDRESS                                                   83216020
                         *                                                                     83216030
                         *                                                                     83216040
                         *                                                                     83216050
39B     3239 800E        LRA1     SRLI  MR1,YD,14           THE REGISTER SPECIFIED BY R1      83216060
                         *                                  CONTAINS THE PROGRAM ADDRESS      83216070
                         *                                  THAT IS TO BE CONVERTED TO A      83216080
                         *                                  REAL ADDRESS. THE CONVERSION      83216090
                         *                                  IS ACCOMPLISHED BY SIMULATING     83216100
                         *                                  THE OPERATION OF THE MEMORY       83216110
                         *                                  ACCESS CONTROLLER, USING AN       83216120
                         *                                  IMAGE IN MEMORY OF THE            83216130
                         *                                  SEGMENTATION REGISTERS. BITS      83216140
                         *                                  12 THROUGH 15 OF R1 SPECIFY       83216150
39C     33BD 5FF0                 NI    PSW,PSW,'FF0'       THE SEGMENT NUMBER. CLEAR CC.     83216160
39D     3231 503C                 NI    MR1,MR1,'3C'        (MR1)=4X SEGMENT NUMBER.          83216170
39E     2B91 18CB                 A     MAR,MR1,MR0,DR2     ADD IMAGE START ADDRESS AND       83216180
                         *                                  FETCH BITS 0:15 OF THE            83216190
                         *                                  SEGMENTATION REGISTER.            83216200
39F     3639 5017                 NI    MR1,YD,LOHALF,I     BLOCK NO/BYTE DISPLACEMENT        83216210
3A0     3211 8004                 SRLI  MR0,MR1,4                                             83216220
3A1     3210 5FF0                 NI    MR0,MR0,'FF0'       (MR0)= BLOCK NUMBER               83216230
3A2     2A5F 1D8F                 L     MR2,MDR,DR4         4:11 = SEGMENT LIMIT FIELD        83216240
                         *                                  FETCH ENTIRE SEG. REGISTER        83216250
3A3     23F2 0866                 SX    NULL,MR2,MR0,LRA2,C COMPARE BLOCK NUMBER TO LIMIT     83216260
3A4     33BD 7008                 OI    PSW,PSW,'8'         IF LIMIT VIOLATION, DON'T         83216270
3A5     2BFF 1F99                 L     NULL,NULL,ILIR,D    CHANGE R1.  EXIT WITH CC = 1000   83216280
3A6     2A1F 1D89        LRA2     L     MR0,MDR,ILIR                                          83216290
3A7     3650 5009                 NI    MR2,MR0,B12,23,I    MASK SEGMENT RELOCATION FIELD     83216300
3A8     2A52 1880                 A     MR2,MR2,MR1         ADD BLOCK NO/BYTE DISPLACEMENT    83216310
3A9     33F0 5010                 NI    NULL,MR0,'10'       TEST CONTROL BITS                 83216320
3AA     13E1 3480                 BALZ  SETV(NULL)          NOT PRESENT, CC=0100 (P.53)       83216330
3AB     2B3F 1900                 L     YD,MR2              REAL ADDRESS REPLACES             83216340
3AC     33F0 5060                 NI    NULL,MR0,'60'       ORIGINAL PROGRAM ADDRESS.         83216350
3AD     17E1 34C0                 BALNZ SETG(NULL)          WRITE PROTECT, CC=0010 (P.53)     83216360
3AE     33F0 5080                 NI    NULL,MR0,'80'                                         83216370
3AF     17E1 3510                 BALNZ SETL(NULL),D        EXECUTE PROTECT, CC=0001 (P.53)   83216380
                         *                                  NO RESTRICTIONS, CC=0000          83216390
```

```
                  * SIMULATE CHANNEL PROGRAM                                               83216410
                  *                                                                        83216420
3B0   2B7F 1F8B   SCP1   L     MDR,NULL,DR2        FETCH CCW                               83216430
3B1   32B0 1002          AI    MR5,TEMP,2          POINT TO BUFFER 0 BYTE COUNT            83216440
3B2   2A5F 1D80          L     CCW,MDR                                                     83216450
3B3   33F2 5008          NI    NULL,CCW,BBIT       TEST BUFFER SWITCH BIT                  83216460
3B4   13E0 ED80          BALZ  SCP2(NULL)          USE BUFFER 0                            83216470
3B5   32B0 100A          AI    MR5,TEMP,10         POINT TO BUFFER 1 BYTE COUNT            83216480
3B6   2B9F 1A8B   SCP2   L     MAR,MR5,DR2         FETCH BUFFER BYTE COUNT                 83216490
3B7   2A7F 3D80          AINC  DAT,NULL,MDR        COUNT+1 TO "DAT"                        83216500
3B8   2A9F 1D80          L     COUNT,MDR           IF COUNT IS POSITIVE                    83216510
3B9   13E8 C340          BALG  LSTOVF(NULL)        SET V FLAG & EXIT (P.32)                83216520
3BA   2B7F 19A3          L     MDR,DAT,DW2,E       STORE INCREMENTED COUNT, SET CC         83216530
3BB   2BFF 1F80          L     NULL,NULL           DISABLE CC UPDATE                       83216540
3BC   3395 1002          AI    MAR,MR5,2                                                   83216550
3BD   2BFF 1F8F          L     NULL,NULL,DR4       FETCH BUFFER END ADDRESS                83216560
3BE   2B94 1D88          A     MAR,COUNT,MDR,DR2   ADD COUNT & FETCH HALFWORD              83216570
3BF   33F2 5004          NI    NULL,CCW,RWBIT      TEST R/W BIT                            83216580
3C0   17E0 F240          BALNZ WRTSC(NULL)         WRITE IF R/W = 1                        83216590
                  *                                                                        83216600
                  * MOVE A BYTE TO THE BUFFER (READ)                                       83216610
                  *                                                                        83216620
3C1   4B79 CDC3          STB   MDR,YD,MDR,DW2                                              83216630
                  *                                                                        83216640
3C2   2A7F 1980          L     DAT,DAT             TEST INCREMENTED COUNT                  83216650
3C3   13E8 F119          BALG  RWSC1(NULL),ILIR,D                                          83216660
                  *                                                                        83216670
                  * EXECUTE NEXT USER INSTRUCTION IF COUNT NOT POSITIVE                    83216680
                  *                                                                        83216690
                  *                                                                        83216700
3C4   33F2 5001   RWSC1  NI    NULL,CCW,FBIT       IF "FAST" MODE, DO                      83216710
3C5   13E0 F199          BALZ  RWSC2(NULL),ILIR,D  NEXT USER INSTRUCTION                   83216720
                  *                                                                        83216730
3C6   3372 6008   RWSC2  XI    MDR,CCW,BBIT        IF COUNT HAS GONE POSITIVE,             83216740
3C7   2B9F 1803          L     MAR,TEMP,DW2        COMPLEMENT BUFFER SWITCH BIT            83216750
3C8   2BFF 1F99          L     NULL,NULL,ILIR,D                                            83216760
                  *                                                                        83216770
                  *                                                                        83216780
3C9   2A3F 1D80   WRTSC  L     DEV,MDR             MOVE A  BYTE FROM THE                   83216790
3CA   4B31 DDC0          LB    YD,DEV,MDR          BUFFER TO R1                            83216800
3CB   2A7F 1980          L     DAT,DAT             TEST INCREMENTED COUNT                  83216810
3CC   13E8 F119          BALG  RWSC1(NULL),ILIR,D                                          83216820
                  *                                                                        83216830
                  * EXECUTE NEXT USER INSTRUCTION IF COUNT NOT POSITIVE                    83216840
```

```
                    * CYCLIC REDUNDANCY CHECK                                    83216860
                    *                                                            83216870
3CD   32D9 503F     CRC12A   NI     MR6,YD,'3F'          MASK 6 BITS             83216880
3CE   363F 1121              LI     MR1,C0F01,I          POLY CHECK              83216890
3CF   2AD6 6D80              X      MR6,MR6,MDR          XOR IN RESIDUAL         83216900
3D0   36D6 5017              NI     MR6,MR6,LOHALF,I     MASK LS 16 BITS         83216910
3D1   32FF 1001              LI     MR7,1                                        83216920
3D2   12B8 F7C0              BAL    CRC12B(RETURN)                               83216930
3D3   2BFF 1F99              L      NULL,NULL,ILIR,D                             83216940
                    *                                                            83216950
                    *                                                            83216960
3D4   3279 50FF     CRC16A   NI     DAT,YD,'FF'          LOAD BYTE               83216970
3D5   12B8 F5C0              BAL    CRC16B(RETURN)       TO COMMON ROUTINE       83216980
3D6   2BFF 1F99              L      NULL,NULL,ILIR,D                             83216990
                    *                                                            83217000
                    * SUBROUTINE SHARED BY AUTO DRIVER CHANNEL                   83217010
                    *                                                            83217020
3D7   363F 1123     CRC16B   LI     MR1,CA001,I          POLY CHECK              83217030
3D8   2AD3 6D80              X      MR6,DAT,MDR          XOR IN RESIDUAL         83217040
3D9   36D6 5017              NI     MR6,MR6,LOHALF,I     MASK LS 16 BITS         83217050
3DA   32FF 1001              LI     MR7,1                                        83217060
                    *                                                            83217070
                    *                                                            83217080
3DB   22D6 8BDD              SRLX   MR6,MR6,MR7,*+2,C    DATA & RESIDUAL EQUAL?  83217090
3DC   2AD6 6880              X      MR6,MR6,MR1          YES, XOR IN FEEDBACK    83217100
3DD   22D6 8BDF              SRLX   MR6,MR6,MR7,*+2,C    DATA & RESIDUAL EQUAL?  83217110
3DE   2AD6 6880              X      MR6,MR6,MR1          YES, XOR IN FEEDBACK    83217120
3DF   22D6 8BE1     CRC12B   SRLX   MR6,MR6,MR7,*+2,C    DATA & RESIDUAL EQUAL?  83217130
3E0   2AD6 6880              X      MR6,MR6,MR1          YES, XOR IN FEEDBACK    83217140
3E1   22D6 8BE3              SRLX   MR6,MR6,MR7,*+2,C    DATA & RESIDUAL EQUAL?  83217150
3E2   2AD6 6880              X      MR6,MR6,MR1          YES, XOR IN FEEDBACK    83217160
3E3   22D6 8BE5              SRLX   MR6,MR6,MR7,*+2,C    DATA & RESIDUAL EQUAL?  83217170
3E4   2AD6 6880              X      MR6,MR6,MR1          YES, XOR IN FEEDBACK    83217180
3E5   22D6 8BE7              SRLX   MR6,MR6,MR7,*+2,C    DATA & RESIDUAL EQUAL?  83217190
3E6   2AD6 6880              X      MR6,MR6,MR1          YES, XOR IN FEEDBACK    83217200
3E7   22D6 8BE9              SRLX   MR6,MR6,MR7,*+2,C    DATA & RESIDUAL EQUAL?  83217210
3E8   2AD6 6880              X      MR6,MR6,MR1          YES, XOR IN FEEDBACK    83217220
3E9   22D6 8BEB              SRLX   MR6,MR6,MR7,*+2,C    DATA & RESIDUAL EQUAL?  83217230
3EA   2AD6 6880              X      MR6,MR6,MR1          YES, XOR IN FEEDBACK    83217240
                    *                                                            83217250
3EB   17FC FB00              BALD   *+1(NULL)                                    83217260
3EC   2B7F 1B03              L      MDR,MR6,DW2          STORE RESULT            83217270
3ED   03F8 0A80              BAL    (RETURN)(NULL)       RETURN                  83217280
```

```
                            1734 *L O A D   P R O G R A M   S T A T U S   W O R D                    83217300
                            1735 *                                                                   83217310
3EE      2A7F 1D8D          1736 LPSW1     L     MR3,MDR,I4DR4       SET PSW ASIDE IN MR3            83217320
3EF      2B5F 1D80          1737           L     LOC,MDR                                             83217330
3F0      339F 1080          1738 TEST1     LI    MAR,'80'                                            83217340
3F1      2BBF 1980          1739           L     PSW,MR3             LOAD NEW PSW                    83217350
3F2      337D 5200          1740           NI    MDR,PSW,'200'       TEST IF QUEUE SERVICE ENABLED   83217360
3F3      13E0 880E          1741           BALZ  TWAIT(NULL),PR4     TEST WAIT IF NOT (P.21)         83217370
                            1742 * QUEUE SERVICE IS ENABLED. TEST THE QUEUE                          83217380
3F4      321F 1002          1743 QUEINT    LI    MR0,2                                               83217390
3F5      2A7F 1D80          1744           L     MR3,MDR             SAVE ADDRESS OF QUEUE           83217400
3F6      2B90 1D8A          1745           A     MAR,MR0,MDR,PR2     FETCH NO. USED HALFWORD         83217410
3F7      2A5D 1F80          1746           A     MR2,PSW,NULL        SAVE PSW                        83217420
3F8      2BFF 1D80          1747           L     NULL,MDR            TEST TALLY                      83217430
3F9      13E0 8800          1748           BALZ  TWAIT(NULL)         EXIT IF ZERO (P.21)             83217440
                            1749 * QUEUE IS NOT EMPTY, DO INTERRUPT                                  83217450
3FA      339F 1088          1750           LI    MAR,'88'            ADRS OF QUEUE SERVICE NEW PSW   83217460
3FB      2A1F 1F8E          1751           L     MR0,NULL,PR4        FETCH NEW PSW                   83217470
3FC      2BBF 1D80          1752           L     PSW,MDR             LOAD NEW PSW                    83217480
3FD      339F 108C          1753           LI    MAR,'8C'                                            83217490
3FE      29BF 198E          1754           L     R13,MR3,PR4         REG 13 = ADRS OF QUEUE          83217500
3FF      13F8 83C0          1755           BAL   COMIN2(NULL)        TO COMMON ROUTINE (P.21)        83217510
```

```
400                    1757            ORG    '400'                                          83217530
      0400             1758  TLSU      EQU    *                                              83217540
400     4AFF 7F80      1759            SMCR   MR7,NULL            COLLECT MACHINE CONTROL REGISTER     83217550
                       1760  * TWO MILLISECOND DELAY FOR SCLRO TO BE RELEASED ON LMI          83217560
401     321F 1208      1761  DELAY     LI     MR0,'208'                                      83217570
402     3210 9004      1762            SLLI   MR0,MR0,4           MR0 = 8320                 83217580
403     2210 2FC3      1763            SDECX  MR0,MR0,NULL,*,C    SINGLE INSTRUCTION LOOP    83217590
404     321F 1005      1764            LI     MR0,5               LOAD UP LSU DEVICE NUMBER  83217600
405     4B70 AFC0      1765            SSRA   MDR,MR0,NULL        ADDRESS & SENSE STATUS     83217610
406     17F5 2780      1766            BALNV  LOADLSU(NULL)       LSU EXISTS IF NO FALSE SYNC (P,51)   83217620


                       1768  * NO LSU, NORMAL POWER UP SEQUENCE                              83217640
                       1769  *                                                               83217650
407     339F 1084      1770  POWRUP    LI     MAR,'84'                                       83217660
408     2B8F 1F8A      1771            L      PSW,NULL,PR2        FETCH PSW SAVE POINTER     83217670
409     369F 1017      1772            LI     MR4,LOHALF,I        USE ONLY LS 16 BITS        83217680
40A     2A94 5D8F      1773            N      MR4,MR4,MDR,DR4     FETCH REGISTER SAVE POINTER          83217690
40B     367F 1017      1774            LI     MR3,LOHALF,I                                   83217700
40C     2B93 5D80      1775            N      MAR,MR3,MDR         USE ONLY LS 16 BITS        83217710
40D     323F 000F      1776            SI     MR1,NULL,15         MR1 = 'FFFFFFF1'           83217720
40E     2B0F 1F8F      1777            L      YDI,NULL,DR4        SELECT R0, FETCH FIRST     83217730
                       1778  *                                                               83217740
40F     2B3F 1D8D      1779  LLOOP     L      YD,MDR,I4DR4        LOAD ONE, FETCH NEXT       83217750
410     23D1 1F4F      1780            AX     YDI,MR1,YDI,LLOOP,C BUMP R1 FIELD, LOOP FOR 16 REGS      83217760
                       1781  *                                                               83217770
                       1782  * ONE REGISTER SET LOADED                                       83217780
                       1783  *                                                               83217790
411     33BD 1010      1784  ENDSET    AI     PSW,PSW,'10'        INCREMENT REGISTER SET NUMBER        83217800
                       1785  *                                    TEST IF 2 OR 8 REGISTER SETS         83217810
412     33F7 5040      1786            NI     NULL,MR7,'40'       MCR BIT 9 = 0 IF 2 SETS    83217820
413     17E1 0540      1787            BALNZ  *+2(NULL)           MCR BIT 9 = 1 IF 8 SETS    83217830
414     33BD 7070      1788            OI     PSW,PSW,'70'        IF ZERO, FORCE LAST SET    83217840
415     33FD 5080      1789            NI     NULL,PSW,'80'       TEST IF LAST SET LOADED    83217850
416     13E1 03C0      1790            BALZ   LLOOP(NULL)         LOOP UNTIL ALL SETS LOADED           83217860
417     323F 000E      1791            SI     MR1,NULL,14         MR1='FFFFFFF2'             83217870
418     37F7 532B      1792            NI     NULL,MR7,BIT20,I    TEST MCR BIT 4             83217880
419     17E1 0740      1793            BALNZ  ENDDLD(NULL)        SKIP IF NO DFU             83217890
41A     CBF9 8D8D      1794  LDLOOP    LW     YD,MDR,I4DR4        LOAD 32 BITS, FETCH NEXT   83217900
41B     CBFB ADCD      1795            LD     YDP1,MDR,I4DR4,K    LOAD LS 32 BITS            83217910
41C     23D1 1F5A      1796            AX     YDI,MR1,YDI,LDLOOP,C POINT TO NEXT HALF REG, LOOP        83217920
      041D             1797  ENDDLD    EQU    *                                              83217930
41D     2B9F 1F8F      1798            L      MAR,NULL,DR4        FETCH FLOATING REGISTER 0  83217940
                       1799  *                                                               83217950
41E                    1800            IFNZ   DFU                                            83217960
41E     CBF9 2DCD      1801  RESTRE    LE     YD,MDR,I4DR4,K      LOAD ONE, FETCH NEXT       83217970
                       1804            ENDC                                                  83218000
41F     23D1 1F5E      1805            AX     YDI,MR1,YDI,RESTRE,C                           83218010
                       1806  *                                                               83218020
420     2B9F 1A0F      1807            L      MAR,MR4,DR4         FETCH SAVED PSW            83218030
421     2A3F 1D8D      1808            L      MR1,MDR,I4DR4       FETCH SAVED LOC            83218040
422     339F 1028      1809            LI     MAR,'28'                                       83218050
423     2B5F 1D8A      1810            L      LOC,MDR,PR2         LOAD LOC, FETCH            83218060
```

```
                          1811  *                                       SAVED CONSOLE STATUS                83218070
424     2B8F 1880         1812            L     PSW,MR1                 LOAD PSW                            83218080
425     4B7F CDC2         1813            STB   MDR,NULL,MDR,PW2        CLEAR HIGH BYTE OF '00028'          83218090
426     323F 1001         1814            LI    MR1,1                                                       83218100
427     4BF1 AFC0         1815            SSRA  NULL,MR1,NULL           ADDRESS THE DISPLAY                 83218110
428     33F7 5200         1816            NI    NULL,MR7,'200'          TEST MCR BIT 6                      83218120
429     17E0 9B40         1817            BALNZ LOCDIS(NULL)            IDLE IF INITIALIZE SWITCH (P.24)    83218130
42A     2A3F 1D80         1818            L     MR1,MDR                                                     83218140
42B     33F1 50E0         1819            NI    NULL,MR1,'0E0'          TEST STATUS BITS 1,2,3              83218150
42C     17E0 9B40         1820            BALNZ LOCDIS(NULL)            SHOW LOC, GO TO IDLE (P.24)         83218160
42D     37FD 5329         1821            NI    NULL,PSW,BIT18,I        TEST MALF ENABLE                    83218170
42E     17E0 84C0         1822            BALNZ MMFINT(NULL)            MACHINE MALFUNCTION (P.21)          83218180
                          1823  *                                       INTERRUPT IF ENABLED                83218190
42F     13F8 8800         1824            BAL   TWAIT(NULL)             TEST PSW WAIT BIT (P.21)            83218200


                          1826  *                                                                           83218220
                          1827  * LONGITUDINAL CHECKSUM                                                     83218230
                          1828  *                                                                           83218240
430     2B73 6D83         1829  LRC       X     MDR,DAT,MDR,DW2         EXCLUSIVE OR CHECKSUM               83218250
431     13F9 2000         1830            BAL   RTNCRC(NULL)            (P.49)                              83218260
```

```
                                     1832  *     A U T O   D R I V E R   C H A N N E L     *         83218280


                                     1834  * CCW BIT DESIGNATIONS                                    83218300
                                     1835  *                                                       83218310
     0080                            1836  EBIT     EQU    '80'            EXECUTE        TEMP   = MR0  83218320
     0010                            1837  CBIT     EQU    '10'            CHECK TYPE     DEV    = MR1  83218330
     0008                            1838  BBIT     EQU    '08'            BUFFER SWITCH  CCW    = MR2  83218340
     0004                            1839  RWBIT    EQU    '04'            READ/WRITE     DAT    = MR3  83218350
     0002                            1840  TBIT     EQU    '02'            TRANSLATE      COUNT  = MR4  83218360
     0001                            1841  FBIT     EQU    '01'            FAST MODE      RETURN = MR5  83218370


432       2B9F 1D0B                  1843  CHANEL   L      MAR,LOC,DR2     FETCH CHANNEL COMMAND WORD     83218390
433       289F 1D00                  1844           L      R4,LOC          REG 4 = ADRS FORCED EVEN OF CCB 83218400
434       2A5F 1DA0                  1845           L      CCW,MDR,E                                      83218410
435       33F2 5080                  1846           NI     NULL,CCW,EBIT   TEST THE EXECUTE BIT           83218420
436       13E1 1280                  1847           BALZ   EXSUB1(NULL)    NO EXECUTE, CC=0 (P.47)        83218430
                                     1848  *                               IF FALL THRU, E=1 & CC=0010    83218440
                                     1849  *                                                              83218450
437       4A7F E940                  1850           EXB    DAT,CCW         ISOLATE STATUS MASK            83218460
438       2BE3 5980                  1851           N      NULL,R3,DAT     TEST DEVICE STATUS AGAINST MASK 83218470
439       17E1 13C0                  1852           BALNZ  EXSUB2(NULL)    BAD STATUS (P.47)              83218480
                                     1853  *                                                              83218490
43A       33F2 5001                  1854           NI     NULL,CCW,FBIT   TEST IF FAST MODE              83218500
43B       13E1 17C0                  1855           BALZ   NFAST(NULL)     NOT FAST MODE (P.48)           83218510


                                     1857  *                F A S T   M O D E              *         83218530


43C       3204 1002                  1859           AI     TEMP,R4,2       ADRS BUFFER 0 BYTE COUNT       83218550
43D       2B9F 180B                  1860           L      MAR,TEMP,DR2    FETCH IT                       83218560
43E       3390 1002                  1861           AI     MAR,TEMP,2                                     83218570
43F       2A9F 1D80                  1862           L      COUNT,MDR                                      83218580
440       13E9 154F                  1863           BALG   EXAUTO(NULL),DR4   EXIT, COUNT POSITIVE (P.47) 83218590
441       2B94 1D8B                  1864           A      MAR,COUNT,MDR,DR2  BUFFER END ADDRESS + COUNT  83218600
                                     1865  *                                                              83218610
                                     1866  * BUFFER 0 BYTE COUNT IN REGISTER "COUNT"                     83218620
                                     1867  * ADDRESS OF BUFFER 0 BYTE COUNT IN "TEMP"                    83218630
                                     1868  * BUFFER 0 END ADRS + BYTE COUNT IN "MAR"                     83218640
                                     1869  * BYTE/HALFWORD TO TRANSFER IN "MDR"                          83218650
                                     1870  *                                                              83218660
442       43FF EFD1                  1871           THWX   NULL,NULL,BYTEIO,C  TEST HW LINE (P.47)        83218670
                                     1872  *                                                              83218680
                                     1873  * FALL THROUGH IF LINE IS ACTIVE                              83218690
                                     1874  *                                                              83218700
443       33F2 5004                  1875           NI     NULL,CCW,RWBIT  TEST R/W BIT                   83218710
444       17E1 1740                  1876           BALNZ  HWRT1(NULL)     WRITE HW, RW=1 (P.47)          83218720
                                     1877  *                                                              83218730
```

```
445     4B7F 4F83           1878          RH     MDR,NULL,DW2         READ HALFWORD                        83218740
446     3294 1002           1879  HRDWT   AI     COUNT,COUNT,2        INCREMENT OF BUFFER 0 BYTE COUNT     83218750
447     2B9F 1800           1880  COMMON  L      MAR,TEMP             RE-ADDRESS BUFFER BYTE COUNT         83218760
448     2B7F 1A03           1881          L      MDR,COUNT,DW2                                             83218770
449     17E9 1540           1882          BALNG  EXAUTO(NULL)         EXIT IF NOT POSITIVE                 83218780
                            1883  *                                                                        83218790
                            1884  * EXIT TO SUBROUTINE IF COUNT HAS BECOME POSITIVE                         83218800


44A     3384 1014           1886  EXSUB1  AI     MAR,R4,20            REG 4 = ADRS OF CCB                  83218820
44B     2BFF 1F8B           1887          L      NULL,NULL,DR2        FETCH SUBROUTINE ADDRESS             83218830
44C     361F 1039           1888          LI     TEMP,CFFFE,I         TEMP='0000FFFE'                      83218840
44D     2B50 5D80           1889          N      LOC,TEMP,MDR                                              83218850
44E     2BFF 1F9C           1890          L      NULL,NULL,IR,D       FETCH USER INSTRUCTION               83218860
                            1891  *                                                                        83218870
                            1892  *                                                                        83218880
44F     33BD 5FF0           1893  EXSUB2  NI     PSW,PSW,'FF0'        CLEAR CC THEN                        83218890
450     23BD 3F8A           1894          AINCX  PSW,PSW,NULL,EXSUB1  SET L FLAG (BAD STATUS)              83218900


451     33F2 5004           1896  BYTEIO  NI     NULL,CCW,RWBIT       TEST R/W BIT                         83218920
452     17E1 1780           1897          BALNZ  FWRIT(NULL)          WRITE BYTE IF RW = 1                 83218930
                            1898  *                                                                        83218940
453     4B7F 0D83           1899          RD     MDR,MDR,DW2          READ BYTE                            83218950
                            1900  *                                                                        83218960
                            1901  *                                                                        83218970
454     2294 3F87           1902  FRDWT   AINCX  COUNT,COUNT,NULL,COMMON INC. BUFF 0 BYTE COUNT            83218980


455     2B5F 1080           1904  EXAUTO  L      LOC,R1               RESTORE LOC                          83219000
456     2BBF 1000           1905          L      PSW,R0               RESTORE PSW                          83219010
457     321F 1008           1906          LI     MR0,8                QUEUE C FLAG                         83219020
458     4AFF 7F8C           1907          SMCR   MR7,NULL,IR          SENSE MACHINE CONTROL REG.           83219030
459     33F7 5007           1908          NI     NULL,MR7,7           TEST FOR EPF OR PARITY ERR           83219040
45A     17E0 8500           1909          BALNZ  MMF1(NULL)           MACHINE MALFUNCTION (P.21)           83219050
45B     37FD 5327           1910          NI     NULL,PSW,BIT16,I     TEST WAIT BIT                        83219060
45C     17E0 AFD0           1911          BALNZ  WAIT(NULL),D         TO WAIT IF SET (P.28)                83219070


45D     43FF 5D86           1913  HWRT1   WHX    NULL,MDR,HRDWT       WRITE HAFLWORD                       83219090
                            1914  *                                                                        83219100
                            1915  *                                                                        83219110
45E     43FF 1D94           1916  FWRIT   WDX    NULL,MDR,FRDWT       WRITE A BYTE                         83219120
```

```
                        *              N O R M A L   M O D E            *                 83219140

45F     321F 1002       NFAST    LI     TEMP,2               SET UP FOR BUFFER 0               83219160
460     33F2 5008                NI     NULL,CCW,BBIT        TEST BUFFER SWITCH BIT            83219170
461     13E1 18C0                BALZ   NFAST1(NULL)                                           83219180
462     321F 100A                LI     TEMP,10              SET UP FOR BUFFER 1               83219190
                        *                                                                      83219200
463     2B84 180B       NFAST1   A      MAR,R4,TEMP,DR2      FETCH BUFFER BYTE COUNT           83219210
464     2A04 1800                A      TEMP,R4,TEMP         TEMP = ADRS OF BUFFER BYTE COUNT  83219220
465     3390 1002                AI     MAR,TEMP,2                                             83219230
466     2A9F 1D8F                L      COUNT,MDR,DR4        FETCH BUFFER END ADDRESS          83219240
467     13E9 1540                BALG   EXAUTO(NULL)         EXIT IF COUNT POSITIVE (P.47)     83219250
468     2B54 1D80                A      LOC,COUNT,MDR        BUFFER END ADRS + COUNT           83219260
469     2B9F 1D0B                L      MAR,LOC,DR2                                            83219270
                        *                                                                      83219280
                        * BUFFER BYTE COUNT IN REGISTER "COUNT"                                83219290
                        * ADDRESS OF BUFFER BYTE COUNT IN "TEMP"                               83219300
                        * BUFFER END ADRS + BYTE COUNT IN "MAR"                                83219310
                        * BYTE/HALFWORD TO TRANSFER IN "MDR"                                   83219320
                        *                                                                      83219330
                        * NOTE: IN NON-FAST MODE, ONLY BYTE TRANSFERS ARE ALLOWED              83219340
                        *                                                                      83219350
                        *                                                                      83219360
46A     33F2 5004                NI     NULL,CCW,RWBIT       TEST R/W BIT                      83219370
46B     17E1 1CC0                BALNZ  NFWRIT(NULL)         WRITE A BYTE IF R/W = 1           83219380
                        *                                                                      83219390
46C     4A7F 0FC0                RDR    DAT,NULL             INPUT THE BYTE                    83219400
46D     2ABF 1980                L      RETURN,DAT           SAVE IT                           83219410
46E     33F2 5002                NI     NULL,CCW,TBIT        TRANSLATION REQUIRED ?            83219420
46F     16E0 DF40                BALNZ  TRANSL(MR7)          DO IT (P.38)                      83219430
                        *                                                                      83219440
470     2B9F 1D0B                L      MAR,LOC,DR2          RE-FETCH HALFWORD                 83219450
471     4B73 CDC3                STB    MDR,DAT,MDR,DW2      INSERT BYTE                       83219460
472     227F 1AB8                LX     DAT,RETURN,REDCHK    DAT = UNTRANSLATED BYTE           83219470


473     2A7F 1D80       NFWRIT   L      DAT,MDR                                                83219490
474     4A73 DDC0                LB     DAT,DAT,MDR          BYTE TO OUTPUT                    83219500
475     33F2 5002                NI     NULL,CCW,TBIT        TEST TRANSLATE BIT                83219510
476     16E0 DF40                BALNZ  TRANSL(MR7)          DO IT (P.38)                      83219520
                        *                                                                      83219530
                        * TRANSLATION NOT REQUIRED                                             83219540
                        *                                                                      83219550
477     4BFF 19C0                WDR    NULL,DAT             OUTPUT THE BYTE                   83219560
                        *                                                                      83219570
                        * ONLY THE BYTE ACTUALLY TRANSFERRED IS INCLUDED IN THE                83219580
                        * LRC OR CRC. SPECIAL CHARACTERS ARE NOT INCLUDED.                     83219590
                        *                                                                      83219600
478     3384 1008       REDCHK   AI     MAR,R4,8                                               83219610
479     2BFF 1F8B                L      NULL,NULL,DR2        FETCH CHECK-WORD                  83219620
47A     33F2 5010                NI     NULL,CCW,CBIT        CHECK TYPE BIT                    83219630
```

```
47B     13E1 0C00          1968          BALZ  LRC(NULL)            LONGITUDINAL CHECKSUM (P.45)         83219640
                           1969  *                                                                        83219650
47C     4AFF 7F80          1970          SMCR  MR7,NULL                                                   83219660
47D     33F7 5400          1971          NI    NULL,MR7,'400'       TEST MCR BIT 5                        83219670
47E     13E1 4000          1972          BALZ  HWASSIST(NULL)       IF ZERO, USE HW ASSIST                83219680
47F     12B8 F5C0          1973  NOASSIST BAL  CRC16B(RETURN)       CYCLIC REDUNDANCY CHECK (P.42)        83219690


480     2B9F 1800          1975  RTNCRC  L     MAR,TEMP                                                   83219710
481     2B74 3F83          1976          AINC  MDR,COUNT,NULL,DW2   INCREMENT & STORE COUNT               83219720
482     17E9 1540          1977          BALNG EXAUTO(NULL)         EXIT IF NOT POSITIVE (P.47)           83219730
                           1978  *                                                                        83219740
483     3372 6008          1979          XI    MDR,CCW,BBIT         COMPLEMENT BUFFER SWITCH BIT          83219750
484     2B9F 1203          1980          L     MAR,R4,DW2           RESTORE CCW                           83219760
485     13F9 1280          1981          BAL   EXSUB1(NULL)         EXIT TO SUBROUTINE (P.47)             83219770
```

```
                        * WRITABLE CONTROL STORE INSTRUCTIONS                                83219790
                        *                                                                    83219800
                        *                                                                    83219810
                        * ENTER CONTROL STORE                                                83219820
                        *                                                                    83219830
486    32FF 1800        ECS1     LI    MR7,'800'          SELECT THE FIRST DCS               83219840
                        *                                 MODULE AT ADRS '800'               83219850
487    2AF7 1F00                 A     MR7,MR7,YDI        ADD R1 FIELD                       83219860
488    03F8 0B80                 BAL   (MR7)(NULL)        BRANCH TO ONE FIRST                83219870
                        *                                 SIXTEEN LOCATIONS IN DCS           83219880


                        * READ/WRITE CONTROL STORE                                           83219900
                        *                                                                    83219910
     0489               WDCS     EQU   *                                                     83219920
489    2B9F 180F        WDCS1    L     MAR,MR0,DR4        FETCH FULLWORD                     83219930
48A    3210 0004                 SI    MR0,MR0,4          DECREMENT MEMORY ADDRESS           83219940
48B    2A5F 1D80                 L     MR2,MDR            COPY DATA TO MR2                   83219950
48C    3FF2 0880                 STR   MR2,MR1            AND STORE IN DCS                   83219960
48D    2A31 2F80                 SDEC  MR1,MR1,NULL       DECREMENT DCS ADDRESS              83219970
48E    2021 2FC9                 SDECX 1,1,NULL,WDCS1,C   DECREMENT COUNT                    83219980
48F    2BFF 1F99                 L     NULL,NULL,ILIR,D   EXIT IF DONE                       83219990
                        *                                                                    83220000
                        *                                                                    83220010
     0490               RDCS     EQU   *                                                     83220020
490    2B9F 1800        RDCS1    L     MAR,MR0                                               83220030
491    2F7F 1887                 L     MDR,MR1,I,DW4      MOVE DCS DATA TO MAIN MEMORY       83220040
492    3210 0004                 SI    MR0,MR0,4          DECREMENT MEMORY ADDRESS           83220050
493    2A31 2F80                 SDEC  MR1,MR1,NULL       DECREMENT DCS ADDRESS              83220060
494    2063 2FD0                 SDECX 3,3,NULL,RDCS1,C   DECREMENT COUNT                    83220070
495    2BFF 1F99                 L     NULL,NULL,ILIR,D                                      83220080
                        *                                                                    83220090
                        *                                                                    83220100
                        * (R1)=DCS ADDRESS,(R1+1)=COUNT, (R2)=MAIN MEMORY ADRS               83220110
                        *                                                                    83220120
                        *                                                                    83220130
496    321B 9002        CCS1     SLLI  MR0,YDP1,2         4X COUNT                           83220140
497    2A10 1C00                 A     MR0,MR0,YS         PLUS MEMORY ADDRESS                83220150
498    2A3B 1C80                 A     MR1,YDP1,YD        MR1=COUNT PLUS DCS ADDRESS         83220160
499    2A5F 1F00                 A     MR2,NULL,YDI       TEST R1 FIELD                      83220170
49A    13E1 2240                 BALZ  WDCS(NULL)         0 = WRITE DCS                      83220180
49B    33F2 6002                 XI    NULL,MR2,2                                            83220190
49C    13E1 2400                 BALZ  RDCS(NULL)         2 = READ DCS                       83220200
49D    17FC 8240                 BALD  ILEGAL(NULL)       ILLEGAL FUNCTION (P.21)            83220210
```

```
                              2027  * LOADER STORAGE UNIT INPUT                                        83220230
                              2028  *                                                                  83220240
      049E                    2029  LOADLSU  EQU   *                                                   83220250
49E     339F 1001             2030           LI    MAR,1                                               83220260
49F     12D9 2B80             2031           BAL   READIT(MR6)                                         83220270
4A0     2BBF 1880             2032           L     PSW,MR1            PSW 16:31                        83220280
4A1     12D9 2B80             2033           BAL   READIT(MR6)                                         83220290
4A2     2B5F 1880             2034           L     LOC,MR1            LOC 16:31                        83220300
4A3     12D9 2B80             2035           BAL   READIT(MR6)                                         83220310
4A4     2A5F 1880             2036           L     MR2,MR1            START ADDRESS                    83220320
4A5     12D9 2B80             2037           BAL   READIT(MR6)                                         83220330
                              2038  *                                 MR1 = END ADDRESS                83220340
4A6     23F1 0968             2039           SX    NULL,MR1,MR2,AUTO1,C COMPARE START & END            83220350
4A7     13F8 9D40             2040           BAL   IDLE(NULL)         IDLE IF START NOT LESS (P.24)    83220360
                              2041  *                                 THAN END ADDRESS                 83220370
4A8     2B9F 190B             2042  AUTO1    L     MAR,MR2,DR2                                         83220380
4A9     4B7F 0080             2043           RD    MDR,MDR            INPUT DATA BYTE                  83220390
4AA     13F4 9D40             2044           BALV  IDLE(NULL)         IDLE IF BAD STATUS (P.24)        83220400
4AB     2A52 3F83             2045  AUTO2    AINC  MR2,MR2,NULL,DW2   INCREMENT START ADRS             83220410
4AC     23F1 0968             2046           SX    NULL,MR1,MR2,AUTO1,C LOOP TILL REACH END ADDRESS    83220420
4AD     13F8 8800             2047           BAL   TWAIT(NULL)        TEST NEW PSW (P.21)              83220430


4AE     4A30 8FC0             2049  READIT   RDRA  MR1,MR0,NULL       INPUT MS BYTE                    83220450
4AF     4A3F E8C0             2050           EXB   MR1,MR1            LEFT 8                           83220460
4B0     4A30 8880             2051           RDA   MR1,MR0,MR1        INPUT LS BYTE                    83220470
4B1     03F8 0B00             2052           BAL   (MR6)(NULL)                                         83220480
```

```
4B2       2B9F 1E00              2054  STD1    L     MAR,MAR              EFFECTIVE ADDRESS TO MAR                83220500
4B3       CB7F 9C87              2055          RRD   MDR,YD,DW4           STORE MS 32 BITS                        83220510
4B4       2BDF 3F00              2056          AINC  YDI,NULL,YDI         POINT TO LOW HALF                       83220520
4B5       CB7F 9C85              2057          RRD   MDR,YD,I4DW4         STORE LS 32 BITS                        83220530
4B6       2BFF 1F9C              2058          L     NULL,NULL,IR,D                                               83220540
                                 2059  *                                                                          83220550
4B7       323F 000F              2060  STMD1   SI    MR1,NULL,15          MR1='FFFFFFF1'                          83220560
4B8       12D8 43C0              2061          BAL   COMSTM(MR6)          TO COMMON ROUTINE (P.10)                83220570
4B9       CB7F 9C85              2062          RRD   MDR,YD,I4DW4         EXECUTED INSTRUCTION                    83220580
                                 2063  *                                                                          83220590
4BA       CBF9 8D8D              2064  LMD1    LW    YD,MDR,I4DR4         LOAD MS 32 BITS, FETCH LS 32 BITS       83220600
4BB       323F 000E              2065          SI    MR1,NULL,14          SET MR1 = 'FFFFFFF2'                    83220610
4BC       CBFB ADC0              2066          LD    YDP1,MDR,K           LOAD LS 32 BITS                         83220620
4BD       2BD1 1F00              2067          A     YDI,MR1,YDI          INCREMENT R1 FIELD BY 2                 83220630
4BE       13F0 448D              2068          BALC  EXLSTM(NULL),I4DR4   EXIT IF FINISHED                        83220640
4BF       13F9 2E80              2069          BAL   LMD1(NULL)           ELSE, FETCH NEXT 32 BITS                83220650
                                 2070  *                                                                          83220660
4C0       2ADF 1F00              2071  FXDR1   L     MR6,YDI                                                      83220670
4C1       2BDF 3E80              2072          AINC  YDI,NULL,YSI         POINT TO R2+1                           83220680
4C2       CA7F 9C80              2073          RRD   MR3,YD               MR0,MR3=ARGUMENT                        83220690
4C3       3230 B008              2074          RLI   MR1,MR0,8            ROTATE MS 32 BITS                       83220700
4C4       3273 B008              2075          RLI   MR3,MR3,8            ROTATE LS 32 BITS                       83220710
4C5       3251 5F00              2076          NI    MR2,MR1,'F00'        FRACTION BITS 0:23                      83220720
4C6       3273 50FF              2077          NI    MR3,MR3,'0FF'        FRACTION BITS 24:31                     83220730
4C7       2A52 7980              2078          O     MR2,MR2,MR3          COMBINED (8 HEX DIGITS)                 83220740
4C8       2BDF 1B00              2079          L     YDI,MR6              RESTORE R1 FIELD                        83220750
4C9       13F8 E2C0              2080          BAL   FXDR2(NULL)          (P.39)                                  83220760
```

```
                              2082 * CONVERT FIXED-POINT DATA TO FLOATING POINT                      83220780
                              2083 *                                                                 83220790
                              2084 *                                                                 83220800
4CA     CBF9 8F80             2085            LW    YD,NULL               NO-OP                      83220810
4CB     2A3F 1C29             2086 FLDR1      L     MR1,YS,ILIR,E         ARGUMENT TO MR1            83220820
4CC     13E1 3440             2087            BALZ  ZEROD(NULL)           EXIT IF ZERO               83220830
4CD     13E9 3400             2088            BALG  FLDR2(NULL)           SKIP IF POSITIVE           83220840
4CE     3610 1133             2089            AI    MR0,MR0,BIT0,I        CONSTANT BECOMES NEGATIVE  83220850
4CF     2A3F 0880             2090            S     MR1,NULL,MR1          2'S COMP ARGUMENT          83220860
4D0     CBF9 8800             2091 FLDR2      LW    YD,MR0                MS 32 BITS                 83220870
      04D1                    2092 ZEROD      EQU   *                                                83220880
4D1     CBFB A890             2093            LD    YDP1,MR1,D            ARGUMENT=LS 32 BITS        83220890
                              2094 *                                      NORMALIZE THE WHOLE THING  83220900


                              2096 * PART OF LRA INSTRUCTION                                         83220920
                              2097 *                                                                 83220930
4D2     33BD 7002             2098 SETV       OI    PSW,PSW,2             SET CC = 4                 83220940
4D3     2BBD 3F80             2099 SETG       AINC  PSW,PSW,NULL          SET CC = 2                 83220950
4D4     2BBD 3F90             2100 SETL       AINC  PSW,PSW,NULL,D                                   83220960


4D5     2B7F 2F83             2102 TS2        SDEC  MDR,NULL,NULL,DW2     WRITE ALL ONES IF NOT SET  83220980
4D6     2BFF 1FB9             2103            L     NULL,NULL,ILIR,E,D    CLEAR CC, FETCH NEXT INSTR 83220990


4D7                           2105            IFNZ  DFU                                              83221010
```

```
4D7      2A3F 1C29              2107  FLR1     L     MR1,YS,ILIR,E      ARGUMENT TO MR1                     83221030
4D8      13E1 3780              2108           BALZ  ZEROE(NULL)        EXIT IF ZERO                        83221040
4D9      13E9 3700              2109           BALG  FLR2(NULL)         SKIP IF POSITIVE                    83221050
4DA      3610 1133              2110           AI    MR0,MR0,BIT0,I     CONSTANT BECOMES NEGATIVE           83221060
4DB      2A3F 0880              2111           S     MR1,NULL,MR1       2'S COMP ARGUMENT                   83221070
4DC      CBFF 8800              2112  FLR2     LW    NULL,MR0           MS 32 BITS CONSTANT                 83221080
4DD      CBF9 2890              2113           LE    YD,MR1,D           ARGUMENT = LS 32 BITS               83221090
                                2114  *                                 NORMALIZE THE WHOLE THING           83221100
4DE      CBF9 2F90              2115  ZEROE    LE    YD,NULL,D          ZERO RESULT                         83221110
```

```
                             2168          ENDC                                                          83221640
4DF                          2170          ORG    '500'                                                  83221660
      0006                   2171 CRC      EQU    X'06'              HARDWARE ASSIST DEVICE NUMBER       83221670
500     32DF 1006            2172 HWASSIST LI     MR6,CRC            (MR6)=CRC DEVICE NUMBER             83221680
501     32F2 8005            2173          SRLI   MR7,CCW,5          POSITION CRC TYPE BITS              83221690
502     32F7 5001            2174          NI     MR7,MR7,1          (MR7) = 0 IF CRC16                  83221700
                             2175 *                                  (MR7) = 1 IF SDLC                   83221710
503     4BF6 BBC0            2176          OCRA   NULL,MR6,MR7       ADDRESS CRC ASSIST, SEND CONTROL    83221720
504     4BFF 5D80            2177          WH     NULL,MDR           OUTPUT OLD RESIDUAL                 83221730
505     4BFF 19C0            2178          WDR    NULL,DAT           OUTPUT NEW DATA BYTE                83221740
506     4B7F 4F83            2179          RH     MDR,NULL,DW2       INPUT & STORE RESULT                83221750
507     13F9 2000            2180          BAL    RTNCRC(NULL)       (P,49)                              83221760


                             2182 * PROCESS BYTE RX                                                      83221780
                             2183 *                                                                      83221790
508     3219 8010            2184 PB1      SRLI   MR0,YD,16          BITS 8:15 OF THE REGISTER           83221800
                             2185 *                                  SPECIFIED BY R1 CONTAIN A           83221810
                             2186 *                                  CONTROL CODE INDICATING             83221820
                             2187 *                                  TYPE OF ERROR CHECKING TO           83221830
509     323F 1006            2188          LI     MR1,CRC            BE PERFORMED.  ADDRESS THE          83221840
50A     4BF1 B840            2189          OCRA   NULL,MR1,MR0       CRC HARDWARE & OUTPUT THE           83221850
                             2190 *                                  CONTROL INFORMATION TO IT.          83221860
50B     4BFF 5D80            2191          WH     NULL,MDR           OUTPUT OLD RESIDUAL                 83221870
50C     4BFF 1CC0            2192          WDR    NULL,YD            OUTPUT THE DATA BYTE IN R1          83221880
                             2193 *                                  TO BE INCLUDED IN THE ERROR         83221890
50D     4B7F 4F83            2194          RH     MDR,NULL,DW2       CHECK.  INPUT THE RESULT            83221900
                             2195 *                                  AND STORE IT.                       83221910
50E     2BFF 1F99            2196          L      NULL,NULL,ILIR,D   FETCH NEXT INSTRUCTION              83221920


                             2198 * PROCESS BYTE RR                                                      83221940
                             2199 *                                                                      83221950
50F     3219 8010            2200 PBR1     SRLI   MR0,YD,16          BITS 0:15 OF THE REGISTER           83221960
                             2201 *                                  SPECIFIED BY R1 CONTAIN A           83221970
                             2202 *                                  CONTROL CODE INDICATING             83221980
                             2203 *                                  TYPE OF ERROR CHECKING TO           83221990
510     323F 1006            2204          LI     MR1,CRC            BE PERFORMED. ADDRESS THE           83222000
511     4BF1 B840            2205          OCRA   NULL,MR1,MR0       CRC HARDWARE & OUTPUT THE           83222010
                             2206 *                                  CONTROL INFORMATION TO IT           83222020
512     4BFF 5C00            2207          WH     NULL,YS            OUTPUT OLD RESIDUAL FROM            83222030
                             2208 *                                  R2 BITS 16:31.                      83222040
513     4BFF 1CC9            2209          WDR    NULL,YD,ILIR       OUTPUT DATA BYTE FROM R2            83222050
                             2210 *                                  TO BE INCLUDED IN THE               83222060
514     4A1F 4F80            2211          RH     MR0,NULL           ERROR CHECK.  INPUT THE             83222070
515     3718 5015            2212          NI     YS,YS,HIHALF,I     RESULT AND STORE IN R2 16:31        83222080
516     2B18 7810            2213          O      YS,YS,MR0,D        WITHOUT CHANGING BITS 0:15          83222090
```

```
                        * MOVE AND PROCESS BYTE STRING                                        83222110
                        *                                                                     83222120
517     2B9F 1C80       MPBSR1   L     MAR,YD              THE REGISTER SPECIFIED BY R1       83222130
                        *                                  CONTAINS THE START ADDRESS OF      83222140
                        *                                  BYTE STRING A.  THE REGISTER       83222150
                        *                                  SPECIFIED BY R1+1 CONTAINS A       83222160
                        *                                  TRANSLATION TABLE ADDRESS.         83222170
                        *                                  THE REGISTER SPECIFIED BY R1+2     83222180
518     33D6 1002                AI    YDI,MR6,2           CONTAINS IN BITS 16:31, A          83222190
519     2A19 8FC0                SRHL  MR0,YD,NULL         POSITIVE BYTE COUNT.  COPY TO      83222200
51A     13E4 C34B                BALL  LSTOVF(NULL),DR2    MR0.  EXIT IF NEGATIVE (P.32)      83222210
                        *                                  ELSE, FETCH 1ST EVEN/ODD BYTE PAIR 83222220
                        *                                                                     83222230
51B     3610 5017       MPBSR2   NI    MR0,MR0,LOHALF,I    BITS 0:15 OF THE REGISTER          83222240
                        *                                  SPECIFIED BY R1+2 CONTAIN          83222250
                        *                                  CONTROL CODES TO IDENTIFY          83222260
51C     3239 8010                SRLI  MR1,YD,16           THE ORDER IN WHICH TRANSLATION     83222270
51D     325F 1006                LI    MR2,CRC             AND ERROR CHECKING ARE TO BE       83222280
51E     33F1 5400                NI    NULL,MR1,'400'      PERFORMED.                         83222290
51F     17E1 5680                BALNZ TRONLY(NULL)        TRANSLATE ONLY (P.59)              83222300
520     33D6 1003                AI    YDI,MR6,3           ELSE, POINT TO R1+3                83222310
521     4BF2 B8C0                OCRA  NULL,MR2,MR1        OUTPUT CONTROL CODE TO CRC BOX     83222320
522     4BFF 5C80                WH    NULL,YD             OUTPUT OLD RESIDUAL FROM R1+3      83222330
523     33D6 1001                AI    YDI,MR6,1                                              83222340
524     33F1 5800                NI    NULL,MR1,'800'      TEST CONTROL CODE                  83222350
525     13E1 5BC0                BALZ  CKONLY(NULL)        ERROR CHECK ONLY (P.60)            83222360
526     33F1 5200                NI    NULL,MR1,'200'                                         83222370
527     13E1 5080                BALZ  CKTR(NULL)          CHECK THEN TRANSLATE (P.58)        83222380
528     13F9 4A80                BAL   TRCK(NULL)          ELSE, TRANSLATE THEN CHECK (P.57)  83222390
```

```
                   * TRANSLATE THEN ERROR CHECK                                        83222410
                   *                                                                   83222420
529   2B9F 1B8B    TRCKL    L      MAR,MR7,DR2          FETCH NEXT BYTE PAIR           83222430
52A   13FD 4AC0    TRCK     BALA   *+1(NULL)            ARM INTERRUPTS                 83222440
52B   33D6 1001             AI     YDI,MR6,1            POINT TO R1+1                  83222450
52C   2A5F 1D80             L      MR2,MDR                                             83222460
52D   4A52 DDC0             LB     MR2,MR2,MDR          ISOLATE APPROPRIATE BYTE       83222470
52E   2A72 1900             A      MR3,MR2,MR2          (MR3)=2X SOURCE BYTE           83222480
52F   2B93 1C8B             A      MAR,MR3,YD,DR2       PLUS TRANSLATION TABLE ADRS    83222490
530   2BDF 1B00             L      YDI,MR6              FETCH TABLE ENTRY              83222500
531   17FD 4C80             BALD   *+1(NULL)            DISARM INTERRUPTS              83222510
532   2B9F 1C00             L      MAR,YS               (MAR)=DESTINATION ADDRESS      83222520
533   2ABF 1D80             L      MR5,MDR              (MR5)=TABLE ENTRY              83222530
534   17E5 604B             BALNL  SUBR(NULL),DR2       TO SUBROUTINE IF NOT MINUS (P.60)  83222540
535   4BFF 1AC0             WDR    NULL,MR5             TRANSLATED BYTE TO CRC BOX     83222550
536   2B39 3F80             AINC   YD,YD,NULL           INCREMENT SOURCE ADDRESS       83222560
537   4B75 CDC3             STB    MDR,MR5,MDR,DW2      INSERT & STORE BYTE            83222570
538   2AFF 1C80             L      MR7,YD               SAVE NEW SOURCE ADDRESS        83222580
539   33D6 1003             AI     YDI,MR6,3            POINT TO R1+3                  83222590
53A   4B3F 4F80             RH     YD,NULL              INPUT NEW CHECKWORD            83222600
53B   3318 1001             AI     YS,YS,1              INCREMENT DESTINATION ADDRESS  83222610
53C   33D6 1002             AI     YDI,MR6,2            POINT TO R1+2                  83222620
53D   3339 0001             SI     YD,YD,1              DECREMENT COUNT FIELD          83222630
53E   2210 2FE9             SDECX  MR0,MR0,NULL,TRCKL,C DECREMENT & TEST COUNT         83222640
53F   2B3F 1800             L      YD,MR0               SET ALL OF R1+2 TO ONES        83222650
540   2BFF 1FB9             L      NULL,NULL,ILIR,E,D   EXIT! CC=0000                  8322266
```

```
                    * ERROR CHECK THEN TRANSLATE                                               83222680
                    *                                                                          83222690
541   2B9F 1B8B     CKTRL   L      MAR,MR7,DR2           FETCH NEXT BYTE PAIR                  83222700
542   13FD 50C0     CKTR    BALA   *+1(NULL)             ARM INTERRUPTS                        83222710
543   33D6 1001             AI     YDI,MR6,1             POINT TO R1+1                         83222720
544   2A5F 1D80             L      MR2,MDR                                                     83222730
545   4A52 DDC0             LB     MR2,MR2,MDR           ISOLATE APPROPRIATE BYTE              83222740
546   2A72 1900             A      MR3,MR2,MR2           MR3=2X SOURCE BYTE                    83222750
547   2B93 1C8B             A      MAR,MR3,YD,DR2        PLUS TRANSLATION TABLE ADDRESS        83222760
548   2BDF 1B00             L      YDI,MR6               FETCH TABLE ENTRY                     83222770
549   17F0 5280             BALD   *+1(NULL)             DISARM INTERRUPTS                     83222780
54A   2B9F 1C00             L      MAR,YS                (MAR)=DESTINATION ADDRESS             83222790
54B   2ABF 1D80             L      MR5,MDR               (MR5)=TABLE ENTRY                     83222800
54C   17E5 604B             BALNL  SUBR(NULL),DR2        TO SUBROUTINE IF NOT MINUS (P.60)     83222810
54D   4BFF 1940             WDR    NULL,MR2              UNTRANSLATED BYTE TO CRC BOX          83222820
54E   2B39 3F80             AINC   YD,YD,NULL            INCREMENT SOUCE ADDRESS               83222830
54F   4B75 CDC3             STB    MDR,MR5,MDR,DW2       STORE TRANSLATED BYTE                 83222840
550   2AFF 1C80             L      MR7,YD                SAVE NEW SOURCE ADDRESS               83222850
551   33D6 1003             AI     YDI,MR6,3             POINT TO R1+3                         83222860
552   4B3F 4F80             RH     YD,NULL               INPUT NEW CHECKWORD                   83222870
553   2B18 3F80             AINC   YS,YS,NULL            INCREMENT DESTINATION ADDRESS         83222880
554   33D6 1002             AI     YDI,MR6,2             POINT TO R1+2                         83222890
555   3339 0001             SI     YD,YD,1               DECREMENT COUNT FIELD                 83222900
556   2210 2FC1             SDECX  MR0,MR0,NULL,CKTRL,C DECREMENT & TEST COUNT                 83222910
557   2B3F 1800             L      YD,MR0                SET ALL OF R1+2 TO ONES               83222920
558   2BFF 1FB9             L      NULL,NULL,ILIR,E,D    EXIT: CC=0000                         83222930
```

```
                     * TRANSLATE ONLY                                                        83222950
                     *                                                                       83222960
559    2B9F 1B8B     TRONLYL  L      MAR,MR7,DR2          FETCH NEXT BYTE PAIR                   83222970
55A    13FD 56C0     TRONLY   BALA   *+1(NULL)            ARM INTERRUPTS                         83222980
55B    33D6 1001              AI     YDI,MR6,1            POINT TO R1+1                          83222990
55C    2A5F 1D80              L      MR2,MDR                                                     83223000
55D    4A52 DDC0              LB     MR2,MR2,MDR          ISOLATE APPROPRIATE BYTE               83223010
55E    2A72 1900              A      MR3,MR2,MR2          2X SOURCE BYTE                         83223020
55F    2B93 1C8B              A      MAR,MR3,YD,DR2       PLUS TRANSLATION TABLE ADDRESS         83223030
560    2BDF 1B00              L      YDI,MR6              FETCH TABLE ENTRY                      83223040
561    17FD 5880              BALD   *+1(NULL)            DISARM INTERRUPTS                      83223050
562    2B9F 1C00              L      MAR,YS               (MAR)=DESTINATION ADDRESS              83223060
563    2ABF 1D80              L      MR5,MDR              (MR5)=TABLE ENTRY                      83223070
564    17E5 604B              BALNL  SUBR(NULL),DR2       TO SUBROUTINE IF NOT MINUS (P,60)      83223080
565    2B39 3F80              AINC   YD,YD,NULL           INCREMENT SOURCE ADDRESS               83223090
566    2AFF 1C80              L      MR7,YD               SAVE NEW START ADDRESS                 83223100
567    4B75 CDC3              STB    MDR,MR5,MDR,DW2      STORE TRANSLATED BYTE                  83223110
568    2B18 3F80              AINC   YS,YS,NULL           INCREMENT DESTINATION ADDRESS          83223120
569    33D6 1002              AI     YDI,MR6,2            POINT TO R1+2                          83223130
56A    3339 0001              SI     YD,YD,1              DECREMENT COUNT FIELD                  83223140
56B    2210 2FD9              SDECX  MR0,MR0,NULL,TRONLYL,C DEC REMENT & TEST COUNT               83223150
56C    2B3F 1800              L      YD,MR0               SET R1+2 TO ALL ONES                   83223160
56D    2BFF 1FB9              L      NULL,NULL,ILIR,E,D   EXIT; CC=0000                          83223170
```

```
                              2323  * ERROR CHECK ONLY                                                          83223190
                              2324  *                                                                         83223200
56E     2B9F 1B8B             2325  CKONLYL  L     MAR,MR7,DR2        FETCH NEXT BYTE PAIR                     83223210
56F     13FD 5C00             2326  CKONLY   BALA  *+1(NULL)          ARM INTERRUPTS                           83223220
570     2A5F 1D80             2327           L     MR2,MDR                                                     83223230
571     4A52 0DC0             2328           LB    MR2,MR2,MDR        ISOLATE APPROPRIATE BYTE                 83223240
572     2B9F 1C0B             2329           L     MAR,YS,DR2         (MAR)=DESTINATION ADDRESS                83223250
573     4BFF 1940             2330           WDR   NULL,MR2           SEND SOURCE BYTE TO CRC BOX              83223260
574     17FD 5D40             2331           BALD  *+1(NULL)          DISARM INTERRUPTS                        83223270
575     2BDF 1B00             2332           L     YDI,MR6            POINT TO R1                              83223280
576     3339 1001             2333           AI    YD,YD,1            INCREMENT SOURCE ADDRESS                 83223290
577     2AFF 1C80             2334           L     MR7,YD                                                      83223300
578     33D6 1003             2335           AI    YDI,MR6,3          POINT TO R1+3                            83223310
579     4B72 CDC3             2336           STB   MDR,MR2,MDR,DW2    INSERT & STORE BYTE                      83223320
57A     4B3F 4F80             2337           RH    YD,NULL            INPUT NEW CHECKWORD                      83223330
57B     2B18 3F80             2338           AINC  YS,YS,NULL         INCREMENT DESTINATION ADDRESS            83223340
57C     33D6 1002             2339           AI    YDI,MR6,2          POINT TO R1+2                            83223350
57D     3339 0001             2340           SI    YD,YD,1            DECREMENT COUNT FIELD                    83223360
57E     2210 2FEE             2341           SDECX MR0,MR0,NULL,CKONLYL,C DECREMENT & TEST COUNT               83223370
57F     2B3F 1800             2342           L     YD,MR0             SET R1+2 TO ALL ONES                     83223380
580     2BFF 1FB9             2343           L     NULL,NULL,ILIR,E,D EXIT; CC=0000                            83223390


                              2345  * EXIT TO SUBROUTINE                                                        83223410
                              2346  *                                                                         83223420
581     33D6 1004             2347  SUBR     AI    YDI,MR6,4          POINT TO R1+4                            83223430
582     2B3F 1D00             2348           L     YD,LOC             SAVE UNINCREMENTED LOC                   83223440
583     2B55 1A80             2349           A     LOC,MR5,MR5        LOAD SUBROUTINE ADDRESS                  83223450
584     2BFF 1F9C             2350           L     NULL,NULL,IR,D                                              83223460
585                           2351           END                                                               83223470
```

```
NO ERRORS

  A           00B4
  ABL         00CA
  ABL1        031C                 425
  ABL2        0321
  ABL3        0324                1429
  AD          00F4
  AD1         011C                 511
  ADDIT       0316                1436
  ADR         0074
  ADRMW       025A                1070
  ADRS        0268                1085
  AE          00D4
  AE1         0062                 441
  AER         0054
  AFAULT      02C7                1280
  AH          0094
  AHI         0194
  AHM         00C2
  AHM1        001E                 413
  AI          01F4
  AIS         004C
  AL          01AA
  AL1         0156                 808
  AL2         015F                 699   701
  AM          00A2
  AR          0014
  ATBL        0300                 421   424
  ATBL1       0303                1380
  ATBL2       0304                1382
  ATL         00C8
  ATL1        0310                 422
  ATL2        0314                1405
  AUTO1       04A8                2039  2046
  AUTO2       04AB
  AUTOIO      02AF                1227
  B12.23      0009                1634
  BAL         0082
  BAL1        0085                 312
  BAL2        0087                 315
  BAL3        009F                 318
  BALR        0002
  BALR1       0005                  73
  BBIT        0008                1650  1678  1921  1979
  BBS         0041                 167   173
  BBS1        0043                 168
  BC1         0035                 314   317
  BC2         0037                  76    79
  BC3         0036                 744   747
  BDCS        01CA
  BFBS        0044
  BFC         0086
  BFCR        0006
```

```
BFFS            0046
BFS             0045               170     176
BFS1            0047               174
BI1720          0021              1208
BI1820          0131              1246
BIT0            0133              2089    2110
BIT15           000F               136    1472
BIT16           0327               133     134    1003    1517    1518    1520    1521    1528    1529    1538    1910
BIT160          0023              1076
BIT17           0328
BIT18           0329              1821
BIT19           032A              1289
BIT20           032B              1034    1792
BIT21           032C
BIT22           032D
BIT23           032E
BIT24           032F
BIT25           0330
BIT26           0331
BIT27           0332
BIT28           0333
BIT29           0334
BIT30           0335
BIT31           0336
BRR             0007                75      78
BRW2            014E               670     680
BRW3            014F               677
BRW4            0153               703
BTABLE          0327              1335
BTBS            0040
BTC             0084
BTCR            0004
BTFS            0042
BXH             0180
BXLE            0182
BXLH            01E0               743     746
BXLH1           01C9               895
BXLH2           01D1               853
BXLH3           01D3               865
BXLH4           01F8               868
BYTEIO          0451              1871
C               00B2
C0F01           0121              1693
C1              0013               336     384     771     917
C2              003C                97
CA001           0123              1707
CADRS           006C               418     483     522     792     807     840     849     855
CADRS1          006E
CADRS2          0049               278
CADRS3          006F               180
CASMD1          0116               564
CASMD2          0117               566
CBIT            0010              1967
CBT             00EE
```

```
CBT1        00EF                  499
CBT2        00DE                  496   502
CCS         01D0
CCS1        0496                  864
CCW         0012                 1649  1650  1662  1675  1678  1845  1846  1850  1854  1875  1896  1921    1941
                                 1946  1955  1967  1979  2173
CD          00F2
CD1         02F8                  508
CDADSDMD    0113                  507   510   513   516   519
CDR         0072
CE          00D2
CE1         011A                  438
CER         0052
CFFFE       0039                 1104  1255  1316  1888
CH          0092
CHANEL      0432                 1258
CHI         0192
CHVR        0024
CHVR1       002C                  137
CI          01F2
CKONLY      056F                 2240
CKONLYL     056E                 2341
CKTR        0542                 2242
CKTRL       0541                 2295
CL          00AA
CL1         00AB                  161
CLB         01A8
CLB1        0172                  805
CLH         008A
CLHI        018A
CLI         01EA
CLR         000A
CLRWT       0257                 1078  1137  1209
CMNBRW      014B                  652
CMSTML      0110                  561
COMBIT      02DE                  492   495   498   501
COMBT1      02E1                 1322
COMBT2      02E2                 1324
COMIN0      020B                 1293
COMIN1      020D
COMIN2      020F                 1755
COMINT      020A                  939
COMLM       0101                  548   552
COMLML      0100                  532
COMMON      0447                 1902
COMSTM      010F                  536   540  2061
CONSER      0243                  953  1136
CONSTANT    0011                  213   304
COUNT       0014                 1655  1661  1862  1864  1879  1879  1881  1902  1902  1928  1930  1976
CR          0012
CRC         0006                 2172  2188  2204  2232
CRC12       00BC
CRC12A      03CD                  399
CRC12B      03DF                 1697
```

```
CRC16       00BE
CRC16A      03D4                  402
CRC16B      03D7                 1702  1973
D           00BA
DO          009C                  396
DAT         0013                 1240  1241  1254  1255  1257  1563  1563  1567  1570  1654  1657  1669   1669
                                 1685  1685  1701  1708  1829  1850  1851  1944  1945  1950  1951  1953   1954
                                 1954  1960  2178
DD          00FA
DD1         0167                  520
DDR         007A
DE          00DA
DELAY       0401
DER         005A
DEV         0011                  846  1210  1230  1233  1236  1239  1239  1240  1241  1251  1252  1683   1684

DFALTO      02C2                   88   352
DFALT1      02C5                 1536  1541
DFLOAT      028B                 1158
DFU         0001                    4    16   109   190   246   255   270   406   432   541   553   572    623
                                  636  1042  1152  1282  1800  2105
DH          009A
DH1         0366                  348
DH2         0370                 1546
DH3         0374                 1539
DHR         001A
DHR1        0363                  108
DIGIT1      0019
DISMEM      025E                 1072
DOSBR       004D                  171   177
DOSBR1      004F                  186
DR          003A
DR1         000B                  159
DR2         000D                   85
EBIT        0080                 1846
ECS         01D2
ECS1        0486                  867
EEXIT1      0066                  113   193   205   208   272   284   290   293   296   299   365   451    581
                                  584   638   707
EEXIT2      0067                  199   202   248
ENDBRW      0155                  676   698
ENDDLD      041D                 1793
ENDSET      0411
EPSR        012A
EPSR1       0127                  603
EXAUTO      0455                 1863  1882  1929  1977
EXBR        0128
EXBR1       016D                  600
EXHR        0068
EXLSTM      0112                  530  2068
EXSUB1      044A                 1847  1894  1981
EXSUB2      044F                 1852
EXTLAT      0387                 1555
EXTRAN      0385                 1566
```

```
FBIT        0001                   1675  1854
FFALT1      02C6
FFAULT      02C6                    259
FLDR        007E
FLDR1       04CB                    305
FLDR2       04D0                   2088
FLR         005E
FLR1        04D7                    214
FLR2        04DC                   2109
FN          028E                   1143
FN0         02A5                   1196
FN01        029F                   1189
FN0123      0297                   1170
FNDIS       027E                   1067
FRDWT       0454                   1916
FWRIT       045E                   1897
FXDR        007C
FXDR1       04C0                    302
FXDR2       038B                   2080
FXR         005C
FXR1        0389                    211
FXR2        0390                   1604
FXR3        0392                   1592
FXR4        0394                   1582
FXR5        039A                   1600
HIHALF      0015                    132   593   599  1200  1516  1519  1527  1537  1540  2212
HRDWT       0446                   1913
HWASSIST    0500                   1972
HWRT1       045D                   1876
IDLE        0275                   2040  2044
IDLE1       0277                   1133
ILEGAL      0209                    957  2025
IOINT0      02A9                    952
IOINT1      02AA                    951
IOINT2      02AC                    950
IOINT3      02AE                    949
IOINTX      02B0                   1211  1230  1233  1298  1300
L           0080
LA          01CC
LA1         002E                    859
LB          01A6
LB1         016F                    802
LBR         0126
LCS         004A
LD          00F0
LD1         00A5                    505
LDLOOP      041A                   1796
LDR         0070
LE          00D0
LE1         001C                    435
LER         0050
LEVEL       0012                   1211  1227  1229  1230  1232  1233  1235  1236  1245  1245  1246  1297    1299

LH          0090
```

```
LHI            0190
LHL            00E6
LHL1           02F6                 490
LI             01F0
LIS            0048
LLOOP          040F                1780   1790
LM             01A2
LM1            0109                 796
LMD            00FE
LMD1           04BA                 526   2069
LME            00E4
LME1           010C                 487
LOADLSU        049E                1766
LOCDIS         026D                1168   1175   1187   1193   1817   1820
LOHALF         0017                1010   1019   1092   1360   1411   1424   1428   1480   1491   1503   1505   1625     1695
                                   1709   1772   1774   2228

LPSW           0184
LPSW1          03EE                 750
LPSWR          0030
LR             0010
LRA            00C6
LRA1           039B                 419
LRA2           03A6                1630
LRC            0430                1968
LSTOVF         030D                1388   1473   1656   2225
M              00B8
MACINT         01FE                 956
MD             00F8
MD1            013D                 517
MDR            0078
ME             00D8
ME1            006A                 447
MER            0058
MH             0098
MH1            035E                 345
MHR            0018
MHR1           035B                 105
MMF1           0214                1909
MMFINT         0213                 954   1822
MPBSR          0060
MPBSR1         0517                 244
MPBSR2         051B
MR             0038
N              00A8
NFAST          045F                1855
NFAST1         0463                1922
NFWRIT         0473                1942
NH             0088
NHI            0188
NI             01E8
NOASSIST       047F
NR             0008
O              00AC
OC             01BC
```

```
OCR         013C
OH          008C
OHI         018C
OI          01EC
ONES        013B                  1587
OR          000C
OUTDIS      026F                  1100  1151  1163  1181
OVF1        038E                  1603
PB          00C4
PB1         0508                   416
PBR         0064
PBR1        050F                   254
POWRUP      0407
PPFINT      0223                   955  1131
PPFSTD      023A
PPFSTE      023E
PSWDIS      0294                  1203
PWRDWN      0242                  1035
QUEINT      03F4
R0          0000                  1249  1905
R1          0001                  1250  1904
R13         000D                  1309  1754
R14         000E                   979  1310
R15         000F                   980  1311
R2          0002                  1251
R3          0003                  1252  1570  1851
R4          0004                  1561  1844  1859  1886  1925  1926  1965  1980
RB          01AE
RB1         0146                   814
RB2         0166                   659
RBL         00CE
RBL1        0351                   431
RBR         012E
RBT         00EC
RD          01B6
RDCS        0490                  2024
RDCS1       0490                  2011
RDFULL      02F0                   359   486   504   525   749   795   810   813
RDHALF      02EA                   398   401   412   415   798   801   804   816   822   825   831   834
RDR         0136
READIT      04AE                  2031  2033  2035  2037
REDCHK      0478                  1951
REMOV       0355                  1504
REMOV1      0357                  1498
RESTRE      041E                  1805
RETURN      0015                  1697  1702  1732  1945  1951  1973
RFULL1      02F3                  1353
RFULL2      02F4                  1355
RH          01B2
RHALF1      02ED                  1343
RHALF2      02EE                  1345
RHR         0132
RLL         01D6
RRL         01D4
```

```
RTBL        0337                427   430
RTBL1       033A               1464
RTBL2       033B               1466
RTL         00CC
RTL1        0347                428
RTL2        034F               1492
RTNCRC      0480               1830  2180
RWBIT       0004               1662  1875  1896  1941
RWBRR       0140                605   608
RWBRX       0147                655
RWSC1       03C4               1670  1686
RWSC2       03C6               1676
S           00B6
SBT         00EA
SCP         01C6
SCP1        03B0                851
SCP2        03B6               1651
SD          00F6
SD1         011E                514
SDR         0076
SE          00D6
SE1         0065                444
SER         0056
SETG        04D3               1640
SETL        04D4               1642
SETV        04D2               1637
SFLOAT      028C               1160
SH          0096
SHI         0196
SI          01F6
SINT        01C4
SINT1       02CB                847
SIS         004E
SLA         01DE
SLHA        019E
SLHL        019A
SLHLS       0122
SLL         01DA
SLLS        0022
SR          0016
SRA         01DC
SRHA        019C
SRHL        0198
SRHLS       0120
SRL         01D8
SRLS        0020
SS          01BA
SS1         0178                832
SSR         013A
ST          00A0
STB         01A4
STB1        016B                799
STBR        0124
STBR1       0169                594
```

```
STD         00E0
STD1        04B2                     481
STE         00C0
STH         0080
STM         01A0
STM1        0103                     793
STMD        00FC
STMD1       04B7                     523
STME        00E2
STME1       0106                     484
STORE       017A                     308   356   405   819
STORE1      017D                     735
STORE2      017E                     737
STRDLP      023C                    1037
STREE       0240                    1047
STRLP       0232                    1025  1032
SUBR        0581                    2258  2285  2312
SVC         01C2
SVC1        02CF                     844
TBIT        0002                    1946  1955
TBT         00E8
TEMP        0010                    1244  1249  1648  1652  1679  1859  1860  1861  1880  1888  1889  1920      1923
                                    1925  1926  1926  1927  1975
TEST1       03F0                     148   597
THI         0186
TI          01E6
TLATE       01CE
TLATE1      0375                     862
TLSU        0400                      59
TRANSL      037D                    1947  1956
TRCK        052A                    2243
TRCKL       0529                    2268
TRONLY      055A                    2234
TRONLYL     0559                    2319
TS          01C0
TS1         01FA                     841
TS2         04D5                     932
TWAIT       0220                     982  1741  1748  1824  2047
WAIT        02BF                    1004  1911
WAIT1       02C1                    1271
WB          01AC
WB1         0144                     811
WBR         012C
WD          01B4
WD1         0176                     823
WDCS        0489                    2022
WDCS1       0489                    2002
WDR         0134
WH          01B0
WHR         0130
WRTSC       03C9                    1663
X           00AE
XH          008E
XHI         018E
```

```
 XI          01EE
 XR          000E
 ZEROD       04D1                2087
 ZEROE       04DE                2108
```

```
                   OPT   SCRT,GO,SQCHK
              *                                                                    83200024
              * COPYRIGHT INTERDATA INC.  JANUARY, 1975                            83200025
              *                                                                    83200026
              * JANUARY 10, 1975                                                   83200027
              *                                                                    83200028
              * * * * * * * * * * * * * * * * * * * * * * * * * * * *              83200030
              *                                                     *              83200040
              * THE PURPOSE OF THIS PROGRAM IS TO DEFINE THE        *              83200050
              * DATA CONTAINED IN THE SEVEN EXTRA ROM CHIPS         *              83200060
              * USED THROUGHOUT THE MODEL 8/32 PROCESSOR.           *              83200070
              *                                                     *              83200080
              * THESE ROM CHIPS ARE:                                *              83200090
              *                                                     *              83200100
              *       19-142R00F41    INSTRUCTION DECODE ROM  CPA   *              83200110
              *       19-142R00F42    FUNCTION DECODE ROM     ALU   *              83200120
              *       19-142R00F43    PRIVILEGED/ILLEGAL ROM  CPB   *              83200130
              *       19-142R00F44    INTERRUPT CONTROL ROM   CPB   *              83200140
              *       19-142R00F45    S-BUS (H) ROM           IOU   *              83200150
              *       19-142R00F46    S-BUS (L) ROM           IOU   *              83200160
              *       19-142R00F47    D-BUS ROM               IOU   *              83200170
              *                                                     *              83200180
              * * * * * * * * * * * * * * * * * * * * * * * * * * * *              83200190
              *                                                                    83200200
              *                                                                    83200210
              *                                                                    83200220
000   26E0 84FF        DC    '26E084FF'                                            83200230
001   29E0 84FF        DC    '29E084FE'                                            83200240
002   26F0 587F        DC    '26E0587E'                                            83200250
003   29E0 58FF        DC    '29E058FF'                                            83200260
004   2F00 08F0        DC    '2F0008F0'                                            83200270
005   2D00 08F0        DC    '2D0008F0'                                            83200280
006   2600 0870        DC    '26000870'                                            83200290
007   2700 08F0        DC    '270008F0'                                            83200300
008   2000 00F0        DC    '200000F0'                                            83200310
009   2008 00F0        DC    '200800F0'                                            83200320
00A   200C 0070        DC    '200C0070'                                            83200330
00B   200F 00F0        DC    '200F00F0'                                            83200340
00C   200E 00F0        DC    '200F00F0'                                            83200350
00D   20EF 00FE        DC    '20EE00FE'                                            83200360
00F   29EF 007E        DC    '29EE007F'                                            83200370
00F   26EE 00FE        DC    '26EE00FE'                                            83200380
              *                                                                    83200390
010   2F0F 84F0        DC    '2F0F84F0'                                            83200400
011   290F 84F0        DC    '290F84F0'                                            83200410
012   260F 5870        DC    '260F5870'                                            83200420
013   290F 58F0        DC    '290F58F0'                                            83200430
014   2F0F 08F0        DC    '2F0F08F0'                                            83200440
015   260F 08F0        DC    '260F08F0'                                            83200450
016   2FEF 0870        DC    '2FEF0870'                                            83200460
017   2FFF 08F1        DC    '2FFF08F1'                                            83200470
018   2608 00F0        DC    '260800F0'                                            83200480
019   290C 00F0        DC    '290C00F0'                                            83200490
01A   2FFE 0071        DC    '2FFE0071'                                            83200500
01B   2FFF 00FF        DC    '2FEF00FF'                                            83200510
01C   261F 00F1        DC    '261F00F1'                                            83200520
01D   2F1F 00F1        DC    '2F1F00F1'                                            83200530
01E   290F 0070        DC    '290F0070'                                            83200540
```

```
01F  260F 00F0          DC    '260F00F0'                                      83200550
020  26E0 28FE          DC    '26E028FE'                                      83200570
021  2910 28F1          DC    '291028F1'                                      83200580
022  2610 2871          DC    '26102871'                                      83200590
023  29E0 28DE          DC    '29E028DE'                                      83200600
024  2F10 59F1          DC    '2F1059F1'                                      83200610
025  2D10 59F1          DC    '2D1059F1'                                      83200620
026  2610 5971          DC    '26105971'                                      83200630
027  2710 59D1          DC    '271059D1'                                      83200640
028  2010 00F1          DC    '201000F1'                                      83200650
029  2010 00F1          DC    '201000F1'                                      83200660
02A  2010 0071          DC    '20100071'                                      83200670
02B  2000 00A0          DC    '200000A0'                                      83200680
02C  2000 04F0          DC    '200004F0'                                      83200690
02D  2000 04F0          DC    '200004F0'                                      83200700
02E  2600 0470          DC    '26000470'                                      83200710
02F  2900 04A0          DC    '290004A0'                                      83200720
               *                                                              83200730
030  2F00 28F0          DC    '2F0028F0'                                      83200740
031  2900 28F0          DC    '290028F0'                                      83200750
032  2600 2870          DC    '26002870'                                      83200760
033  2900 28A0          DC    '290028A0'                                      83200770
034  2F00 00F0          DC    '2F0000F0'                                      83200780
035  2600 00F0          DC    '260000F0'                                      83200790
036  2F00 0070          DC    '2F000070'                                      83200800
037  2F00 00A0          DC    '2F0000A0'                                      83200810
038  2600 33F0          DC    '260033F0'                                      83200820
039  2900 33F0          DC    '290033F0'                                      83200830
03A  2F00 3370          DC    '2F003370'                                      83200840
03B  2F00 33A0          DC    '2F0033A0'                                      83200850
03C  2600 00F0          DC    '260000F0'                                      83200860
03D  2F10 00F1          DC    '2F1000F1'                                      83200870
03E  2900 0070          DC    '29000070'                                      83200880
03F  2600 00A0          DC    '260000A0'                                      83200890
```

```
040  86E0 84FE              DC     '86E084FE'                                                     83200910
041  89E0 84FE              DC     '89E084FE'                                                     83200920
042  86E0 587E              DC     '86E0587E'                                                     83200930
043  89E0 58FF              DC     '89E058FF'                                                     83200940
044  8FE0 08FE              DC     '8FE008FE'                                                     83200950
045  8DE0 08FF              DC     '8DE008FE'                                                     83200960
046  86E0 087E              DC     '86E0087F'                                                     83200970
047  87E0 08FF              DC     '87E008FE'                                                     83200980
048  80E0 00FE              DC     '80E000FE'                                                     83200990
049  80E0 00FF              DC     '80E000FE'                                                     83201000
04A  80E0 007E              DC     '80E0007F'                                                     83201010
04B  80F0 00FF              DC     '80F000FE'                                                     83201020
04C  80E0 00FE              DC     '80E000FE'                                                     83201030
04D  80E0 00FE              DC     '80E000FE'                                                     83201040
04E  89E0 007E              DC     '89E0007E'                                                     83201050
04F  89F0 00FE              DC     '89F000FF'                                                     83201060
                  *                                                                               83201070
050  8FE0 84FF              DC     '8FE084FE'                                                     83201080
051  89E0 84FE              DC     '89F084FF'                                                     83201090
052  86E0 587E              DC     '86E0587E'                                                     83201100
053  89F0 58FE              DC     '89E058FF'                                                     83201110
054  8FF0 08FF              DC     '8FE008FF'                                                     83201120
055  86F0 08FF              DC     '86F008FE'                                                     83201130
056  8FE0 087F              DC     '8FE0087F'                                                     83201140
057  8FE0 08FF              DC     '8FE008FE'                                                     83201150
058  86E0 00FF              DC     '86E000FF'                                                     83201160
059  89E0 00FE              DC     '89E000FE'                                                     83201170
05A  8FF0 007F              DC     '8FE0007F'                                                     83201180
05B  8FE0 00FF              DC     '8FE000FF'                                                     83201190
05C  86F0 00FF              DC     '86F000FF'                                                     83201200
05D  8FE0 00FF              DC     '8FE000FE'                                                     83201210
05E  89E0 007E              DC     '89E0007E'                                                     83201220
05F  86E0 00FE              DC     '86E000FF'                                                     83201230
```

```
060   86E0 28FE           DC     '86E028FE'                                       83201250
061   8910 28F1           DC     '891028F1'                                       83201260
062   8610 2871           DC     '86102871'                                       83201270
063   89E0 28DE           DC     '89E028DF'                                       83201280
064   8F10 94F1           DC     '8F1094F1'                                       83201290
065   8D10 94F1           DC     '8D1094F1'                                       83201300
066   8610 5971           DC     '86105971'                                       83201310
067   8710 59D1           DC     '871059D1'                                       83201320
068   8010 00F1           DC     '801000F1'                                       83201330
069   8010 00F1           DC     '801000F1'                                       83201340
06A   8010 0071           DC     '80100071'                                       83201350
06B   8000 00A0           DC     '800000A0'                                       83201360
06C   8000 02F0           DC     '800002F0'                                       83201370
06D   8000 02F0           DC     '800002F0'                                       83201380
06E   8600 0470           DC     '86000470'                                       83201390
06F   8600 04A0           DC     '860004A0'                                       83201400
                    *                                                             83201410
070   8FE0 00FE           DC     '8FE000FF'                                       83201420
071   89F0 00FE           DC     '89E000FE'                                       83201430
072   86F0 007E           DC     '86E0007E'                                       83201440
073   89E0 00AE           DC     '89E000AE'                                       83201450
074   8FE0 45FF           DC     '8FE045FF'                                       83201460
075   86E0 45FE           DC     '86E045FE'                                       83201470
076   8FE0 457E           DC     '8FE0457E'                                       83201480
077   8FE0 45AE           DC     '8FE045AF'                                       83201490
078   86E0 00FE           DC     '86E000FE'                                       83201500
079   89F0 00FE           DC     '89E000FE'                                       83201510
07A   8FF0 007E           DC     '8FE0007F'                                       83201520
07B   8FE0 00AE           DC     '8FE000AF'                                       83201530
07C   86E0 00FE           DC     '86F000FE'                                       83201540
07D   8FE0 00FF           DC     '8FE000FF'                                       83201550
07E   89F0 007E           DC     '89E0007F'                                       83201560
07F   86E0 00AE           DC     '86E000AF'                                       83201570
```

```
080  0660 00F6          DC    '066000F6'                                           83201590
081  0960 00F6          DC    '096000F6'                                           83201600
082  0660 0076          DC    '06600076'                                           83201610
083  0960 00F6          DC    '096000F6'                                           83201620
084  0F60 00F6          DC    '0F6000F6'                                           83201630
085  0D60 00F6          DC    '0D6000F6'                                           83201640
086  0660 0076          DC    '06600076'                                           83201650
087  0760 00F6          DC    '076000F6'                                           83201660
088  0000 00F0          DC    '000000F0'                                           83201670
089  0000 00F0          DC    '000000F0'                                           83201680
08A  0000 0070          DC    '00000070'                                           83201690
08B  0000 00F0          DC    '000000F0'                                           83201700
08C  0000 00F0          DC    '000000F0'                                           83201710
08D  0020 00F2          DC    '002000F2'                                           83201720
08E  0920 0072          DC    '09200072'                                           83201730
08F  0660 00F6          DC    '066000F6'                                           83201740
                   *                                                               83201750
090  2FE0 00FE          DC    '2FE000FE'                                           83201760
091  29E0 00FE          DC    '29E000FE'                                           83201770
092  2620 00F2          DC    '262000F2'                                           83201780
093  2920 00F2          DC    '292000F2'                                           83201790
094  2F20 00F2          DC    '2F2000F2'                                           83201800
095  2620 00F2          DC    '262000F2'                                           83201810
096  2F20 00F2          DC    '2F2000F2'                                           83201820
097  2F20 00F2          DC    '2F2000F2'                                           83201830
098  2900 00F0          DC    '290000F0'                                           83201840
099  2600 00F0          DC    '260000F0'                                           83201850
09A  2F00 00F0          DC    '2F0000F0'                                           83201860
09B  2F00 00F0          DC    '2F0000F0'                                           83201870
09C  26E0 00FE          DC    '26E000FE'                                           83201880
09D  2FE0 00FE          DC    '2FE000FE'                                           83201890
09E  2900 00F0          DC    '290000F0'                                           83201900
09F  2620 00F2          DC    '262000F2'                                           83201910
```

```
0A0  0600 00F0          DC     '060000F0'                                    83201930
0A1  0900 00B0          DC     '090000B0'                                    83201940
0A2  0600 0070          DC     '06000070'                                    83201950
0A3  0900 00D0          DC     '090000D0'                                    83201960
0A4  0F00 00F0          DC     '0F0000F0'                                    83201970
0A5  0D00 00F0          DC     '0D0000F0'                                    83201980
0A6  0600 0070          DC     '06000070'                                    83201990
0A7  0700 00A0          DC     '070000A0'                                    83202000
0A8  0000 00F0          DC     '000000F0'                                    83202010
0A9  0000 00B0          DC     '000000B0'                                    83202020
0AA  0000 0070          DC     '00000070'                                    83202030
0AB  0000 00D0          DC     '000000D0'                                    83202040
0AC  00E0 00FE          DC     '00E000FE'                                    83202050
0AD  00E0 00FE          DC     '00E000FE'                                    83202060
0AE  0600 0070          DC     '06000070'                                    83202070
0AF  0900 00A0          DC     '090000A0'                                    83202080
                  *                                                          83202090
0B0  0FE0 00FE          DC     '0FE000FE'                                    83202100
0B1  09E0 00FE          DC     '09E000FE'                                    83202110
0B2  0600 00F0          DC     '060000F0'                                    83202120
0B3  0900 00F0          DC     '090000F0'                                    83202130
0B4  0F00 00F0          DC     '0F0000F0'                                    83202140
0B5  0600 00F0          DC     '060000F0'                                    83202150
0B6  0F00 00F0          DC     '0F0000F0'                                    83202160
0B7  0F00 00F0          DC     '0F0000F0'                                    83202170
0B8  0900 00F0          DC     '090000F0'                                    83202180
0B9  0600 00F0          DC     '060000F0'                                    83202190
0BA  0F00 00F0          DC     '0F0000F0'                                    83202200
0BB  0F00 00F0          DC     '0F0000F0'                                    83202210
0BC  0600 00F0          DC     '060000F0'                                    83202220
0BD  0F00 00F0          DC     '0F0000F0'                                    83202230
0BE  0900 00F0          DC     '090000F0'                                    83202240
0BF  0600 00F0          DC     '060000F0'                                    83202250
```

```
0C0  8660  00F6            DC      '866000F6'                                                  83202270
0C1  8960  00F6            DC      '896000F6'                                                  83202280
0C2  8660  00F6            DC      '866000F6'                                                  83202290
0C3  1960  00F6            DC      '196000F6'                                                  83202300
0C4  1F60  00F6            DC      '1F6000F6'                                                  83202310
0C5  1D60  00F6            DC      '1D6000F6'                                                  83202320
0C6  1660  00F6            DC      '166000F6'                                                  83202330
0C7  1760  00F6            DC      '176000F6'                                                  83202340
0C8  10E0  00FE            DC      '10E000FE'                                                  83202350
0C9  10E0  00FE            DC      '10E000FE'                                                  83202360
0CA  10E0  00FE            DC      '10E000FE'                                                  83202370
0CB  1000  00F0            DC      '100000F0'                                                  83202380
0CC  10E0  00FE            DC      '10E000FE'                                                  83202390
0CD  10E0  00FE            DC      '10E000FE'                                                  83202400
0CE  19E0  00FF            DC      '19E000FF'                                                  83202410
0CF  19E0  00FE            DC      '19E000FE'                                                  83202420
                   *                                                                           83202430
0D0  8F20  00F2            DC      '8F2000F2'                                                  83202440
0D1  8920  00F2            DC      '892000F2'                                                  83202450
0D2  8620  00F2            DC      '862000F2'                                                  83202460
0D3  8920  00F2            DC      '892000F2'                                                  83202470
0D4  8F20  00F2            DC      '8F2000F2'                                                  83202480
0D5  8620  00F2            DC      '862000F2'                                                  83202490
0D6  8F20  00F2            DC      '8F2000F2'                                                  83202500
0D7  8F20  00F2            DC      '8F2000F2'                                                  83202510
0D8  8900  00F0            DC      '890000F0'                                                  83202520
0D9  8600  00F0            DC      '860000F0'                                                  83202530
0DA  8F00  00F0            DC      '8F0000F0'                                                  83202540
0DB  8F00  00F0            DC      '8F0000F0'                                                  83202550
0DC  8600  00F0            DC      '860000F0'                                                  83202560
0DD  8F00  00F0            DC      '8F0000F0'                                                  83202570
0DE  8900  00F0            DC      '890000F0'                                                  83202580
0DF  8600  00F0            DC      '860000F0'                                                  83202590
```

```
0E0  86E0 00FE            DC    '86E000FE'                                        83202610
0E1  89E0 00FE            DC    '89E000FE'                                        83202620
0E2  1600 00F0            DC    '160000F0'                                        83202630
0E3  8900 00F0            DC    '890000F0'                                        83202640
0E4  8FE0 00FE            DC    '8FE000FE'                                        83202650
0E5  8DE0 00FE            DC    '8DE000FE'                                        83202660
0E6  86E0 00FE            DC    '86E000FE'                                        83202670
0E7  8710 00F1            DC    '871000F1'                                        83202680
0E8  20E0 00FE            DC    '20E000FE'                                        83202690
0E9  10E0 00FE            DC    '10E000FE'                                        83202700
0EA  10E0 00FE            DC    '10E000FE'                                        83202710
0EB  10E0 00FE            DC    '10E000FE'                                        83202720
0EC  10E0 00FE            DC    '10E000FE'                                        83202730
0ED  1000 00F0            DC    '100000F0'                                        83202740
0EE  1600 00F0            DC    '160000F0'                                        83202750
0EF  1600 00F0            DC    '160000F0'                                        83202760
                *                                                                 83202770
0F0  0FE0 00FE            DC    '0FE000FE'                                        83202780
0F1  09E0 00FE            DC    '09E000FE'                                        83202790
0F2  0600 00F0            DC    '060000F0'                                        83202800
0F3  4900 00F0            DC    '490000F0'                                        83202810
0F4  4F00 00F0            DC    '4F0000F0'                                        83202820
0F5  4600 00F0            DC    '460000F0'                                        83202830
0F6  4F00 00F0            DC    '4F0000F0'                                        83202840
0F7  4F00 00F0            DC    '4F0000F0'                                        83202850
0F8  4900 00F0            DC    '490000F0'                                        83202860
0F9  4600 00F0            DC    '460000F0'                                        83202870
0FA  4F00 00F0            DC    '4F0000F0'                                        83202880
0FB  4F00 00F0            DC    '4F0000F0'                                        83202890
0FC  0600 00F0            DC    '060000F0'                                        83202900
0FD  0F00 00F0            DC    '0F0000F0'                                        83202910
0FE  0900 00F0            DC    '090000F0'                                        83202920
0FF  06E0 00FE            DC    '06E000FE'                                        83202930
                *                                                                 83202940
100                       END                                                     83202950
```

CUT ALONG LINE

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

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 229

FOLD FOLD

STAPLE STAPLE