DWG NO. 98A0887

REVISIONS

REV	EN	CHG CODE	DESCRIPTION	DR	APPD
X			PROTOTYPE RELEASE		
X2			ADDED SHEET 39 THRU 46		J.L.K. 3/27/73
A		—	PRODUCTION RELEASE EN 82239		G.E.K 8/1/73
B	83588	—	REVISED PER MARKED UP ENGRG DWG	PSH	11/27/75

NEXT ASSEMBLY	MODEL NO. V70 SERIES	varian data machines / a varian subsidiary 2722 michelson drive / irvine / california / 92664
DR K.ELLINOR 10/26/72 CHK R.A.D. 7/31/73 SGN ——— —— GR R.C.DRUMM 7/31/73 D A.WHITCOMB 8/1/73	CODE IDENT NO. 21101 THIS DOCUMENT MAY CONTAIN PROPRIETARY INFORMATION AND SUCH INFORMATION MAY NOT BE DISCLOSED TO OTHERS FOR ANY PURPOSE OR USED TO PRODUCE THE ARTICLE OR SUBJECT, WITHOUT PERMISSION FROM VDM.	TITLE V70 SERIES MICRO WORD FLOWCHARTS SIZE A | DWG NO. 98A0887 | REV B SHEET 1 OF 57

TABLE OF CONTENTS

SECTION 1
INTRODUCTION

1.1 Purpose

The function of this document is to provide microprogram information describing the standard V70 series "macro" instruction on set. This instruction set is implemented in page zero of the control store (512 words of Read-only memory), and is designed to emulate the 620F instruction set a maximum performance level. For a description of this standard instruction set, see the V70 series handbooks (V72, V73, or V74 System Handbooks).

Although the V70 series "macro" instruction set is fixed, pages of 256 or 512 words of user definable control store (random access memories) may be added, via a "Writable Control Store Option", thus allowing for user defined "macro" and ' micro" instructions on programs in a second environment. For information about the Writable Control Store Option, see the WCS manual (98A9906-08X).

1.2 Performance Specifications

The use of high speed logic; and read-only memory, allows for a micro-exicution time of 165 nanoseconds. This high speed micro-exicution time results in very efficient use of existing system resources, such as semiconductor memory with a cycle time of 330 nanoseconds (or two micro-instructions).

1.3 Reference Documents

For further information concerning V70 Series hardware and micro-programming refer to the following documents:

Title	Document Number
Writable Control Store Manual	98A9906-08X
V70 Series Processor Manual	98A9906-02X
V73 System Handbook	98A9906-01X
Varian Microprogramming Guide	98A9906-07X
Core Memory Manual	98A9906-03X
Semiconductor Memory	98A9906-04X
Option Board Manual	98A9906-05X
Power Supply Manual	98A9906-06X

SECTION 2
FUNCTIONAL DESCRIPTION

2.1 Flowchart Definitions

Flowchart definitions will consist of explanations for memory operation, control store addressing, data loop operations, and the use of processor flags. Included are examples of several types of operations performed in the flow diagrams to follow.

2.1.1 Memory Operations

When memory operations are to be performed, or initiated by a micro-command, a statement of the type of memory operation to be performed, as well as the sources and/or destinations for memory data and address, will be made. An example is given below:

Example: A.F. @ A, M = address fetch with ALU as address and MIL as the fetched data's destination.

2.1.1.1 Type of operations:

a. I.F. = instruction fetch

b. A.F. = address fetch

c. O.F. = opened fetch

d. O.S. or STO = store

e. Cond. = conditional memory initiation (condition specified by ◇).

2.1.1.2 Memory Address Sources:

a. P = program counter to memory address

b. A = ALU output to memory address

c. M = memory input latch (MIL register) to memory address

2.1.1.3 Memory Fetch Destinations:

a. I = MII and MIL register - data from memory is to be loaded into both the instruction buffer register (MII) and the memory input latch (MIL).

b. M = MIL register - data from memory is to be loaded into the memory input latch (MIL).

2.1.2 Field Selection

The operation of selecting and decoding up to a five bit field from the instruction register (C21), for use in control store branching operations, is known as field selecting. Field selecting can only take place from the instruction register - C21. Field Selection statements will appear as follows:

F. Sel I (00-02) or Fld Sel. 0-2 - Field selection from the instruction register (C21) bits 0 thru 2.

2.1.3 Data Loop Operations

When data loop operations are performed, statement describing the action taking place will occur. Examples are given below:

a. P + 1 → S1 - program register added to 1 and transferred to the S1 register.

b. A → DOR - The contents of the A-register are transferred to the Data Loop Operand register.

2.1.4 Instruction Register Transfer

Instruction register transfers are labeled as below:

Statement: I1 → I2 = M11 → C21 = transfer contents of instruction buffer (M11) to the instruction register (C21).

2.1.5 Flags

Several Flags are used for storing and testing the status of the machine A few are listed below:

a. Statement = operation on the interrupt flag, CINTF.
Example = RST CINTF - reset the interrupt flag.

b. Statement = Ofl., Ovrfl - overflow indicator

c. Statement = DSB - a utility flip-flop used to provide temporary storage of data or control information during the execution of various instructions.

2.1.6 Other Terms

Other terms used with the micro-flow are given in Table 1. See Sheet 7.

TABLE I

FLOWCHART DEFINITIONS

EXPRESSION	INTERPRETATION
Mem. Strt.	Memory start. A memory operation is being initiated.
Incr. P	Increment the Program Counter by one.
0 → DOR	The contents of the Data Loop Operand Register are set to zero.
A → DOR	The contents of the A Register are transferred to the Data Loop Operand Register.
AVB → DOR	The contents of the A and B Registers are inclusively OR'ed and are transferred to DOR.
Cond.	Conditional. Used to denote that the execution of a specific operation is dependent upon the outcome of a test being performed.
En.	Enable.
D-ROM	Instruction decoder read-only-memory.
M11	Preliminary instruction register. A buffer register used to receive all instructions from memory. This register provides input to both C21 and the D-ROM.
MDN	Memory Done. A signal which identifies the completion of a memory operation.
IDN	I/O done. A signal which identifies the completion of an I/O operation.
S1, S2	Two registers located in the Register File which are used for temporary storage of operands, addresses, etc. Also known as E (S1) and F (S2).

EXPRESSION	INTERPRETATION
Zero	A status indicator which is set during various instructions when the result of an arithmetic/logical operation is zero.
Enable "Jump" Signal.	A "Jump" signal is generated and sent to the Memory Protect option to signify that a program "jump" or "skip" operation is occurring.
Mask → DOR	The 16-bit "Mask" field of the micro-word is transferred to DOR (after inversion of each bit).
I/O Strt.	I/O Start. An I/O operation is being initiated or allowed to continue execution.
IOR	A buffer register used for temporary storage of data or control information during the execution of various I/O operations.
C21	Main instruction register. This register contains the current instruction during most of instruction execution.
CINTF	Interrupt flag. A flip-flop used to denote that an interrupt instruction execution is or has been in progress.
DOR $\xrightarrow[\text{Func}]{\text{Spec. ALU}}$ A	The contents of DOR are transferred through the adder (ALU) to the A Register. The adder operation will transfer, increment, decrement, or invert as a function of a hardware decoding of bits 6 and 7 of the current instruction.
Override Mem.	A modification to a previously initiated and current memory operation; e.g., a "Read" is modified to a "Write".
Sample Overflow	An operation in which the results of an arithmetic operation are allowed to set the Overflow Indicator if overflow conditions exist.
Mask I + P → DOR	A 16-bit word, generated by AND'ing the micro-word "Mask" field and the current instruction word (contents of C21), is added to the contents of the Program Counter. The result is placed in DOR.

EXPRESSION	INTERPRETATION
Mem. Contr.	Memory Control logic section.
MIL	A buffer register used to receive and hold operands and operand addresses from memory.
I/O Sense Flag	A flip-flop used to receive and hold the sense response bit transmitted from a device controller during a SEN instruction execution.
Intrpt.	Interrupt.

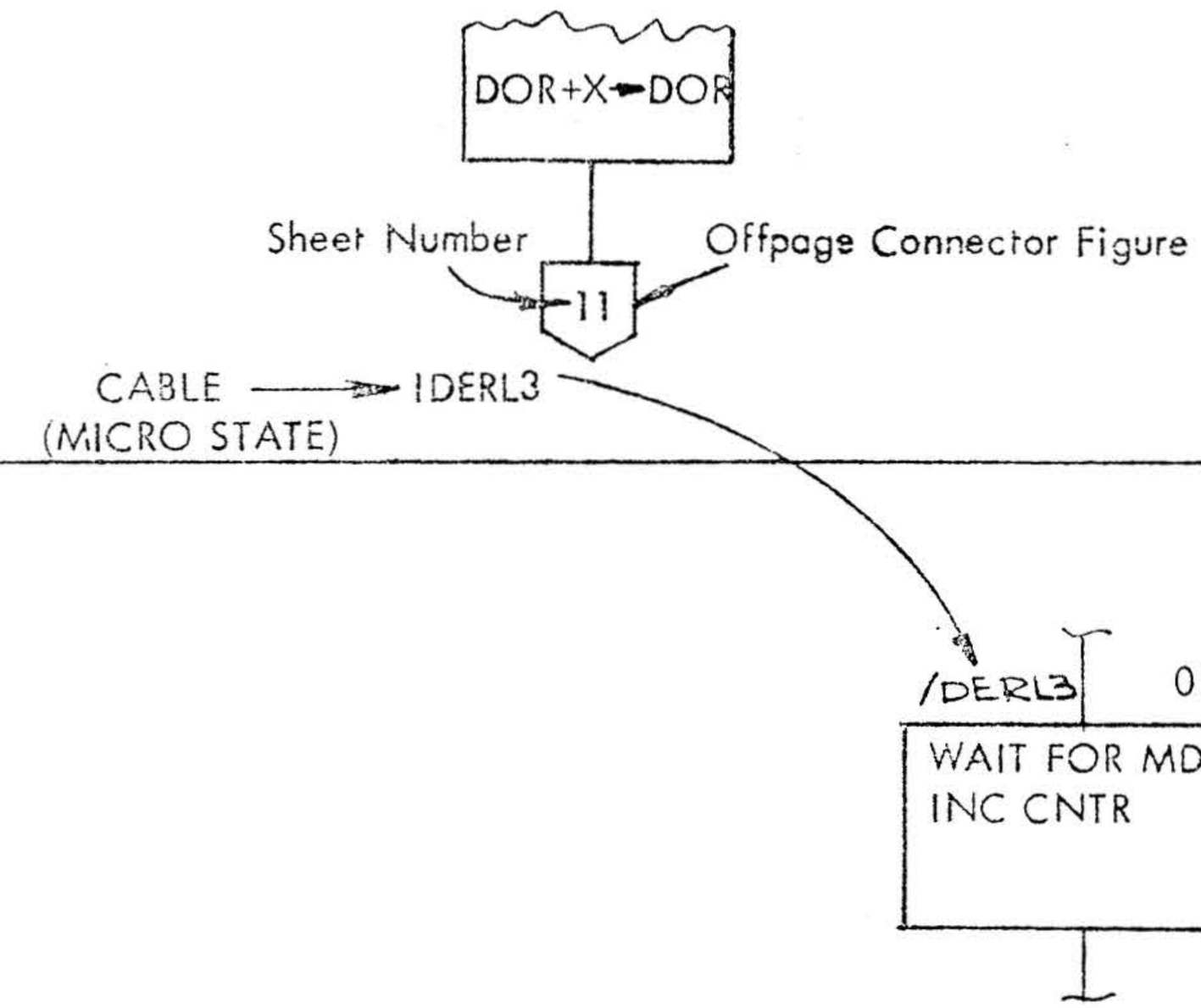
2.2 V70 Series Flowcharts

The V70 Series Flowcharts give detailed information for each class of instruction used in the standard V70 instruction set; front panel routines; interrupt routine; automatic bootstrap loader routines; V70 "Standard" micro-states; and Input/Output microflows.

For exact Binary Contents of each control store address, see listing for V70 Control Store #49A0195-000 and #49A0195-001.

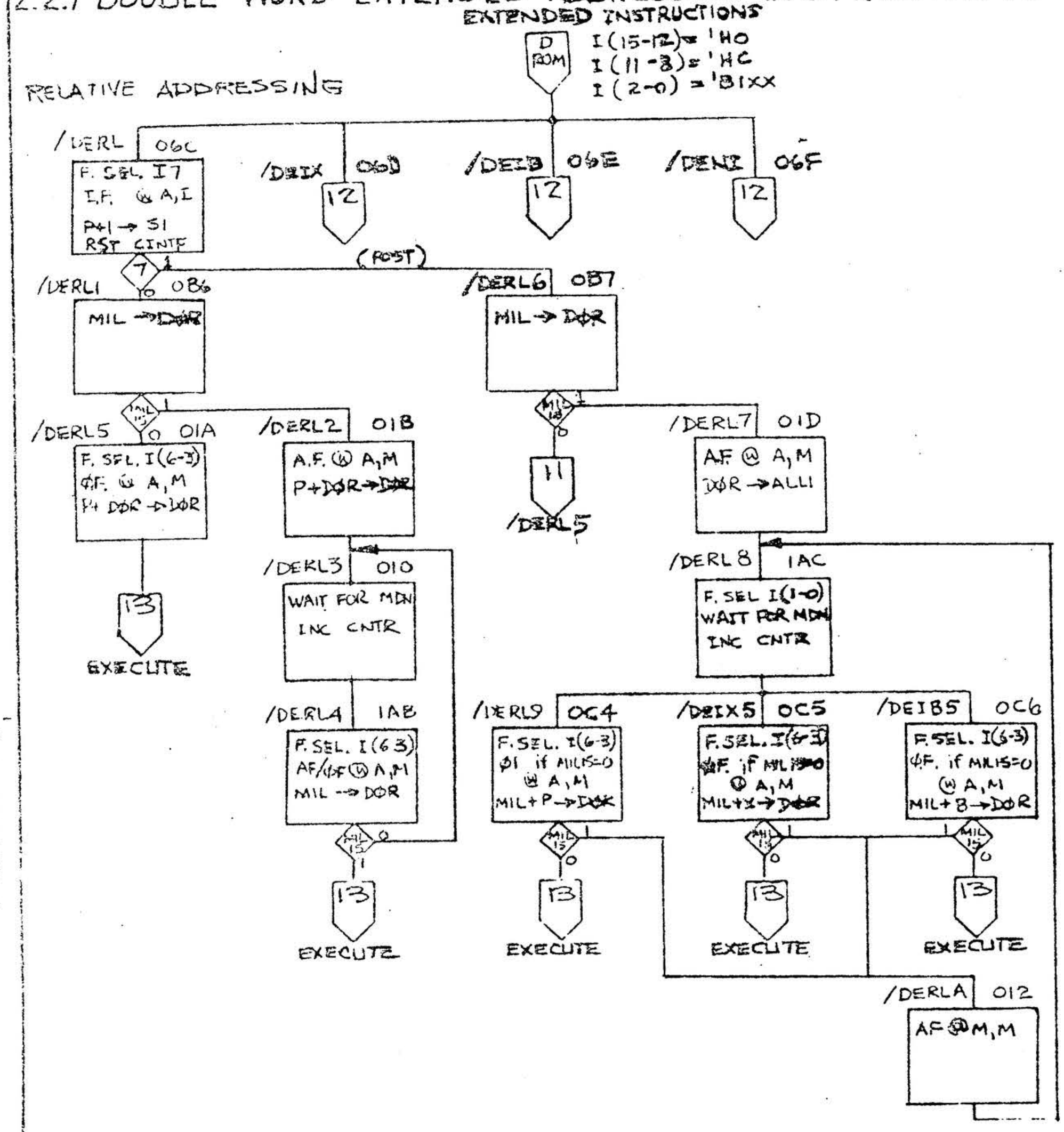
Offpage connectors are referenced by sheet number, located at bottom of each sheet (right hand side). Offpage connectors will reference to a specific Micro State, or will indicate some general operation to follow (i.e. Execute, Next Instruction, etc.). See example below.

Example:



PROGRAMMER	PROGRAM NO.	DATE	PAGE
CHART I.D.	CHART NAME	PROGRAM NAME	

2.2.1 DOUBLE WORD EXTENDED ADDRESSING INSTRUCTIONS



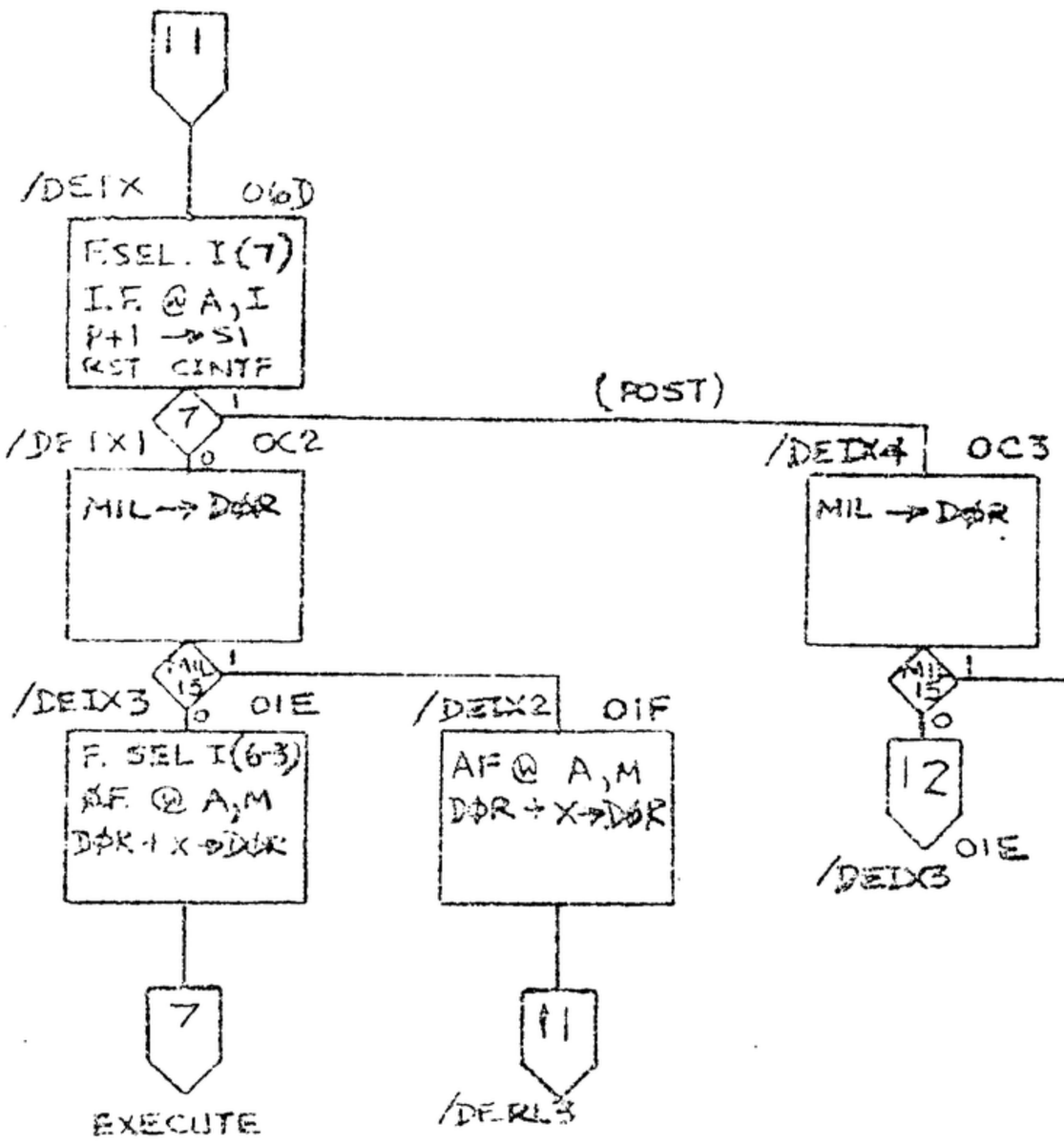
FOLD UNDER AT DOTTED LINE

FOLD UNDER AT DOTTED LINE

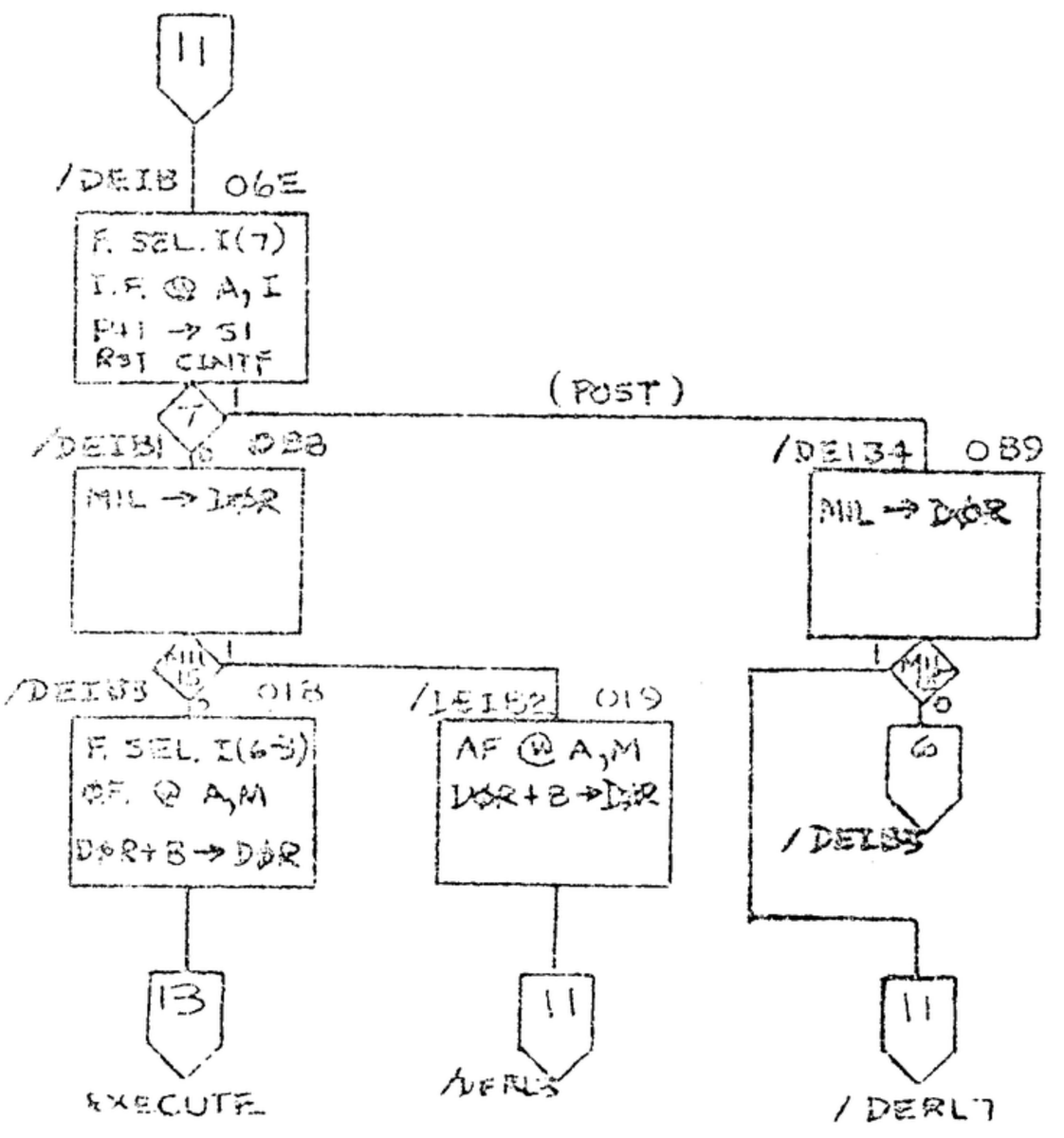
varian data machines a varian subsidiary	CODE IDENT. NO.	EXTENDED INSTRUCTIONS	98A0887	REV. B
	PREPARED BY	APPROVED BY	SHEET 11 OF 57	

PROGRAMMER	PROGRAM NO.	DATE	PAGE
CHART I.D.	CHART NAME	PROGRAM NAME	

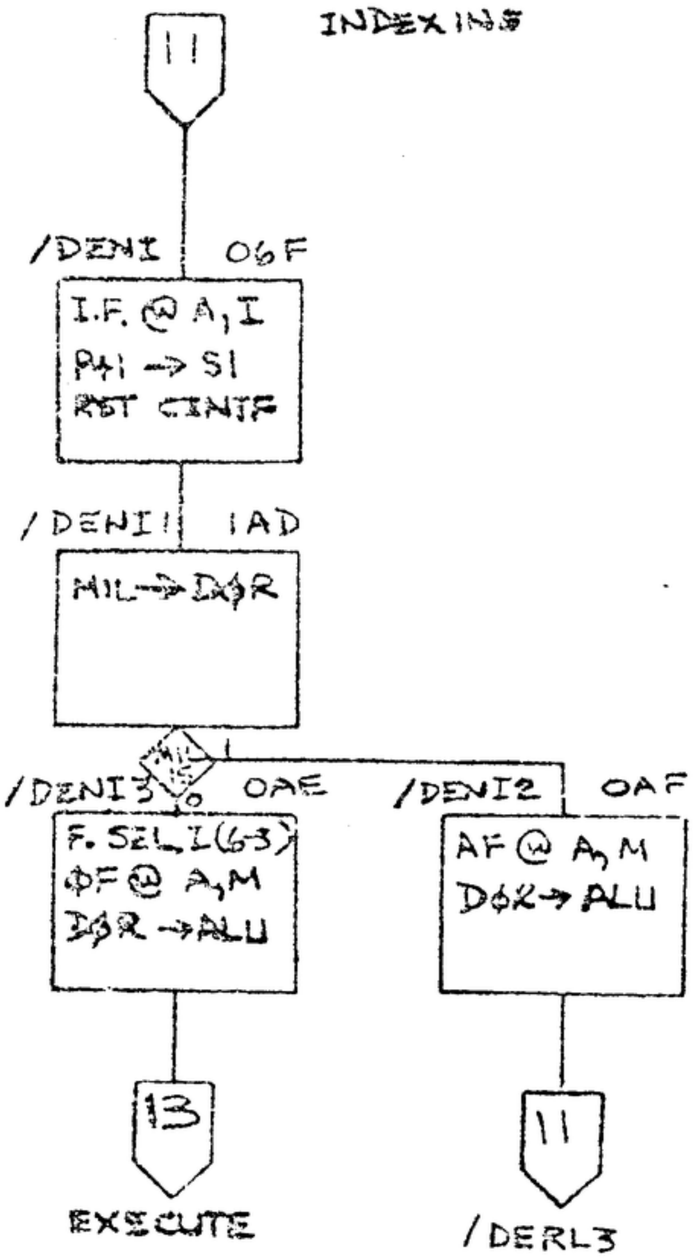
DOUBLE-WORD EXTENDED INDEX @ X



DOUBLE-WORD EXTENDED INDEX @ B



DOUBLE-WORD EXTENDED NO INDEXING



FOLD UNDER AT DOTTED LINE

FOLD UNDER AT DOTTED LINE

varian data machines, a varian subsidiary	CODE IDENT. NO.	EXTENDED INSTRUCTIONS	98A0387	REV. D
	PREPARED BY	APPROVED BY	SHEET 12 OF 57	

PROGRAMMER	PROGRAM NO	DATE	PAGE
CHART I.D.	CHART NAME	PROGRAM NAME	

EXTENDED EXECUTE

FIELD SEL. I(6→3)

/ELDA 001
I.F. @ A, I
SI+1 → P
/ELDA1 1AE
SEL. D-ROM & Samp INT's
I1 → I2
MIL → A
NEXT INST.

/ELDB 002
I.F. @ A, I
SI+1 → P
/ELDB1 1AF
SEL. D-ROM & SAMP INT's
I1 → I2
MIL → B
NEXT INST

/ELDX 003
I.F. @ A, I
SI+1 → P
/ELDX1 1B8
SEL D-ROM & SAMP. INT's
I1 → I2
MIL → X
NEXT INST

/EØRA 009
I.F. @ A, I
SI+1 → P
/EØRA1 1B9
SEL D-ROM & SAMP. INT's
I1 → I2
MILVA → A
NEXT INST.

/EADD 00A
I.F. @ A, I
SI+P → P
/EADD1 1BC
SEL. D-ROM & SAMP. INT's
I1 → I2
MIL+A → A
SAMP ØFL
NEXT INST

/EERA 00B
I.F. @ A, I
SI+1 → P
/EERA1 1BA
SEL. D-ROM & SAMP. INT's
I1 → I2
MIL ⊕ A → A
NEXT INST

/ESUB 00C
I.F. @ A, I
SI+1 → P
/ESUB1 1BD
SEL. D-ROM & SAMP INT's
I1 → I2
SAMP ØFL
A − MIL → A
NEXT INST

/EANA 00D
I.F. @ A, I
SI+1 → P
/EANA1 1BB
SEL. D-ROM & SAMP INT's
I1 → I2
MIL ∧ A → A
NEXT INST

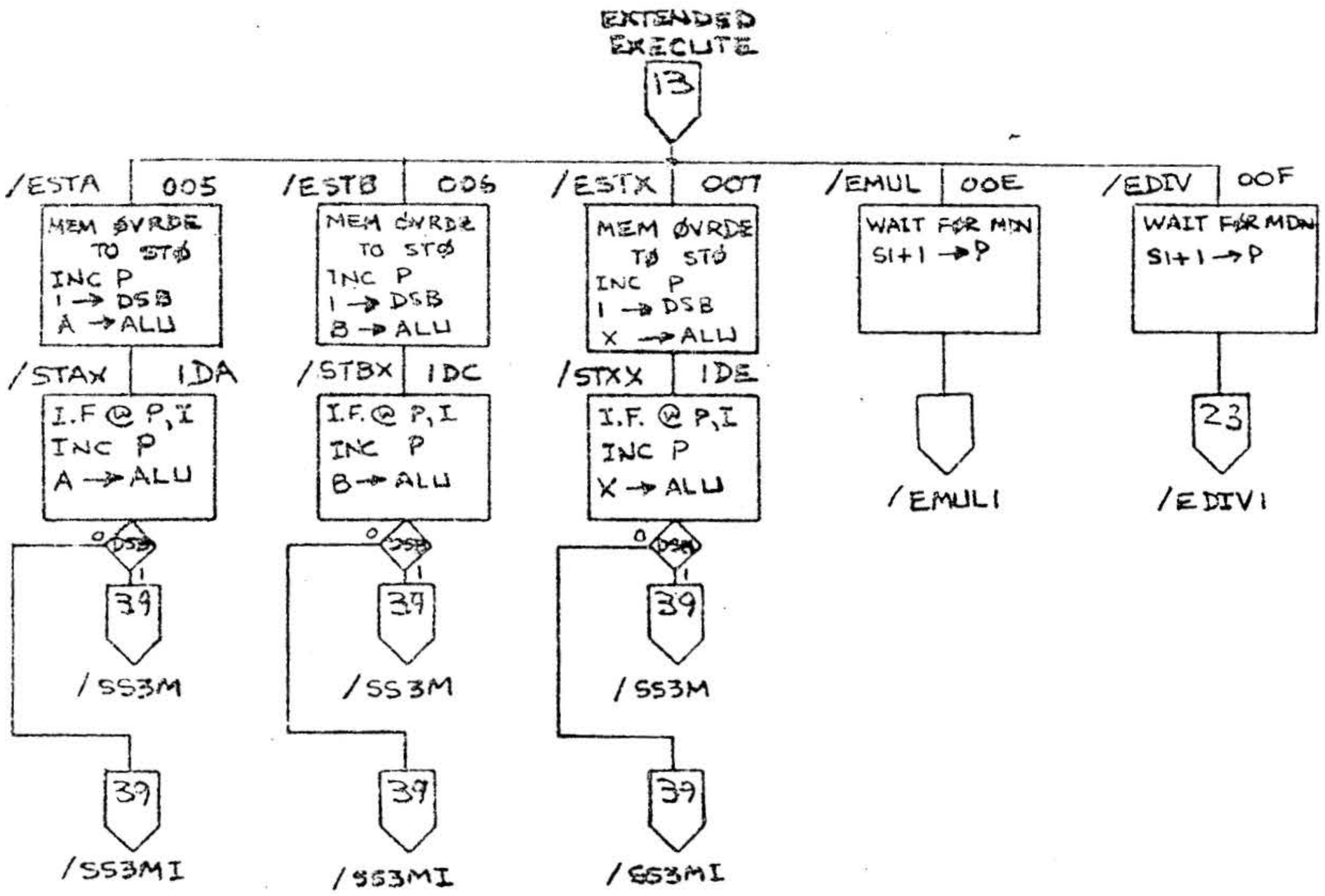
/EINR 004
STØ @ A
INC P
1 → DSB
DSR+1 → ALU
/INRX 1DF
SAMP ØFL
MIL+1 → ALU
/INR2 0BC
I.F. @ P, I
INC P
MIL+1 → ALU
0 (SKIP) 1
553M
553MI

14
EXTENDED EXECUTE

FOLD UNDER AT DOTTED LINE

varian data machines, a varian subsidiary	CODE IDENT. NO.	EXTENDED INSTRUCTIONS	93A0887	REV. E
	PREPARED BY	APPROVED BY	SHEET 13 OF 57	

PROGRAMMER	PROGRAM NO.	DATE	PAGE
CHART I.D.	CHART NAME	PROGRAM NAME	



FOLD UNDER AT DOTTED LINE

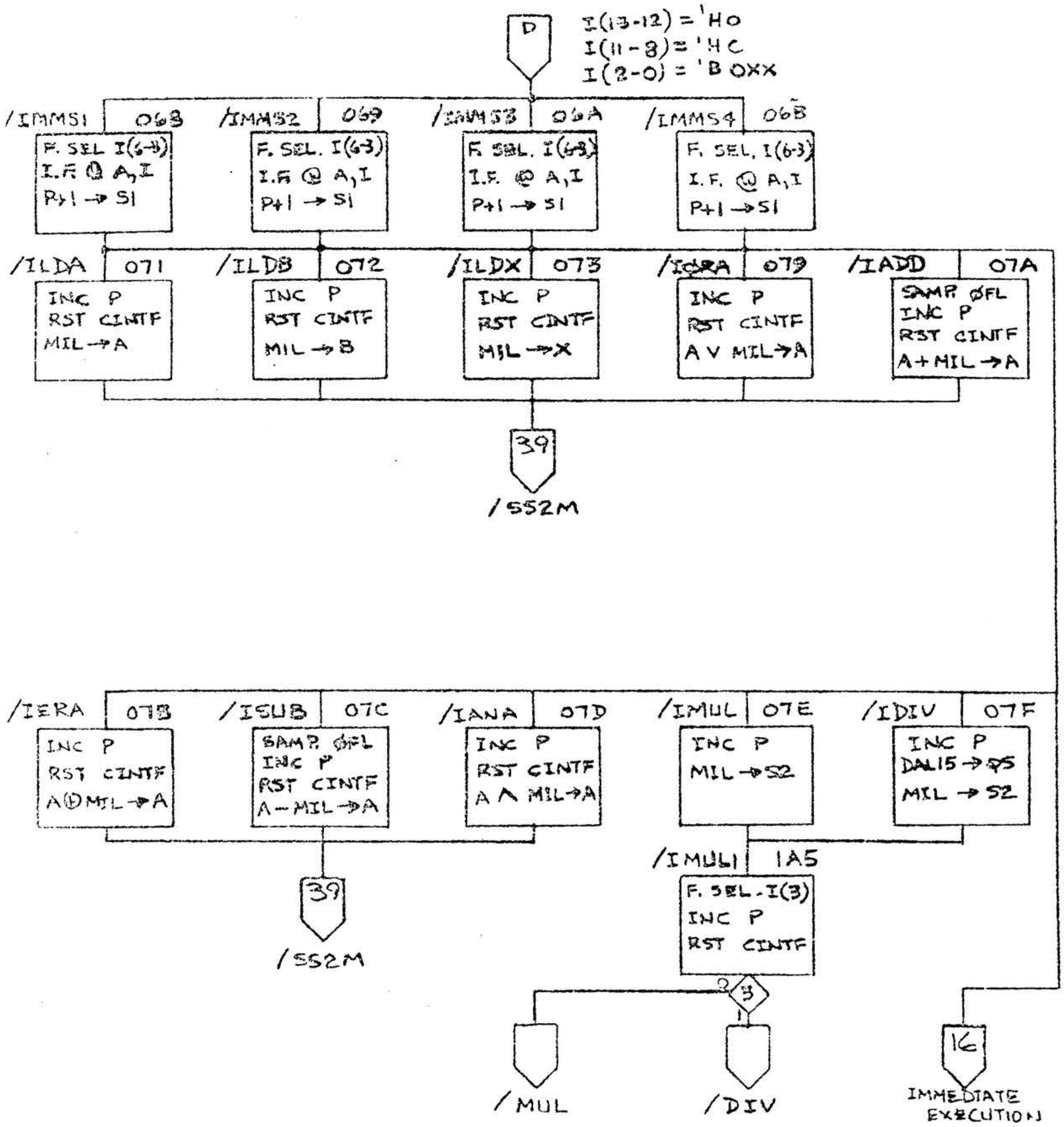
FOLD UNDER AT DOTTED LINE

varian data machines, a varian subsidiary	CODE IDENT. NO.	EXTENDED INSTRUCTIONS	98A0887	REV.
	PREPARED BY	APPROVED BY	SHEET 14 OF 57	

24A834)-000A

PROGRAMMER	PROGRAM NO.	DATE	PAGE
CHART I.D.	CHART NAME	PROGRAM NAME	

2.2.2 DOUBLE WORD IMMEDIATE INSTRUCTION

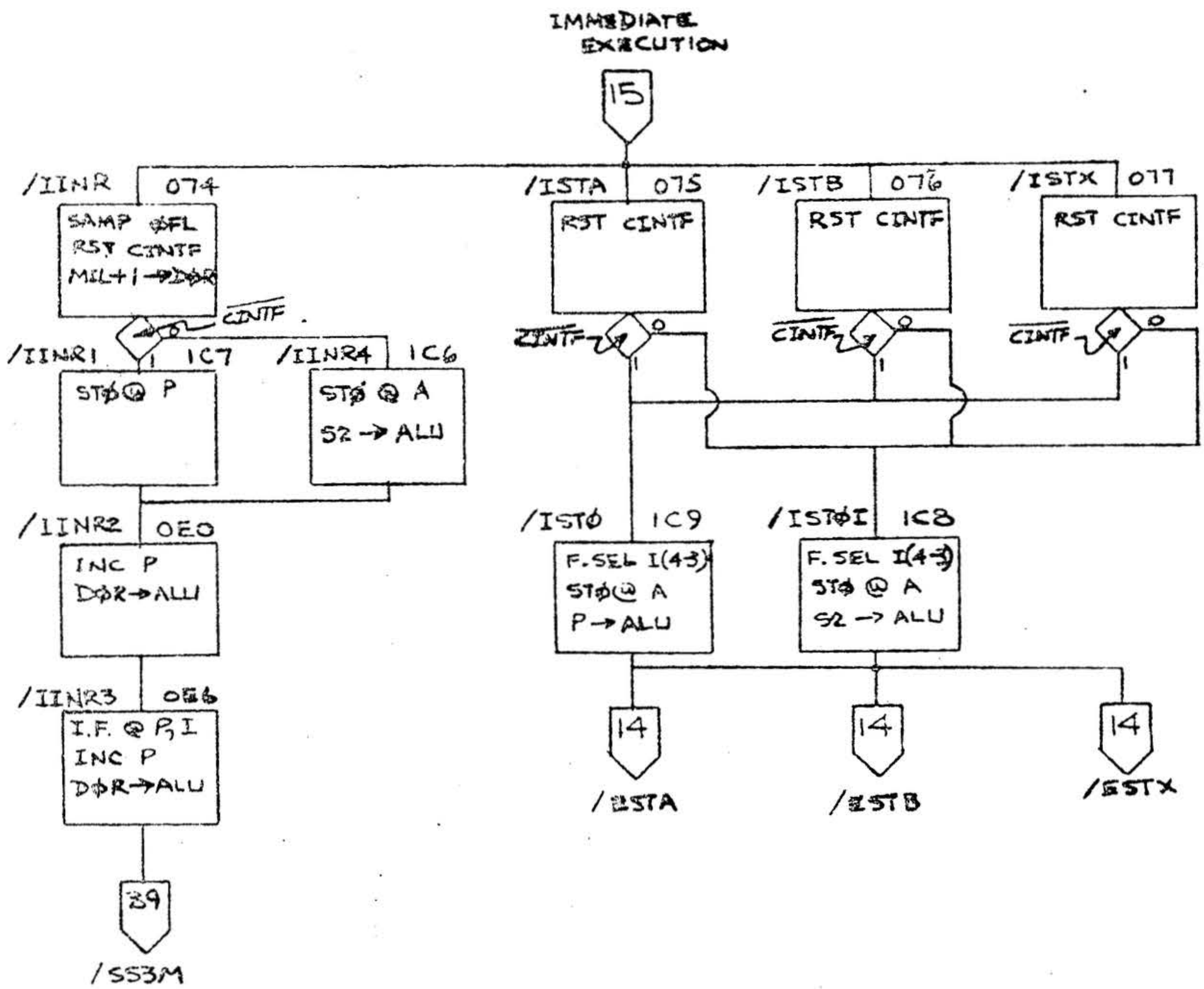


FOLD UNDER AT DOTTED LINE

FOLD UNDER AT DOTTED LINE

varian data machines, a varian subsidiary	CODE IDENT. NO.	IMMEDIATE INSTRUCTION	98A0887	REV. E
	PREPARED BY	APPROVED BY	SHEET 15 OF 57	

PROGRAMMER	PROGRAM NO.	DATE	PAGE
CHART I.D.	CHART NAME	PROGRAM NAME	



FOLD UNDER AT DOTTED LINE

FOLD UNDER AT DOTTED LINE

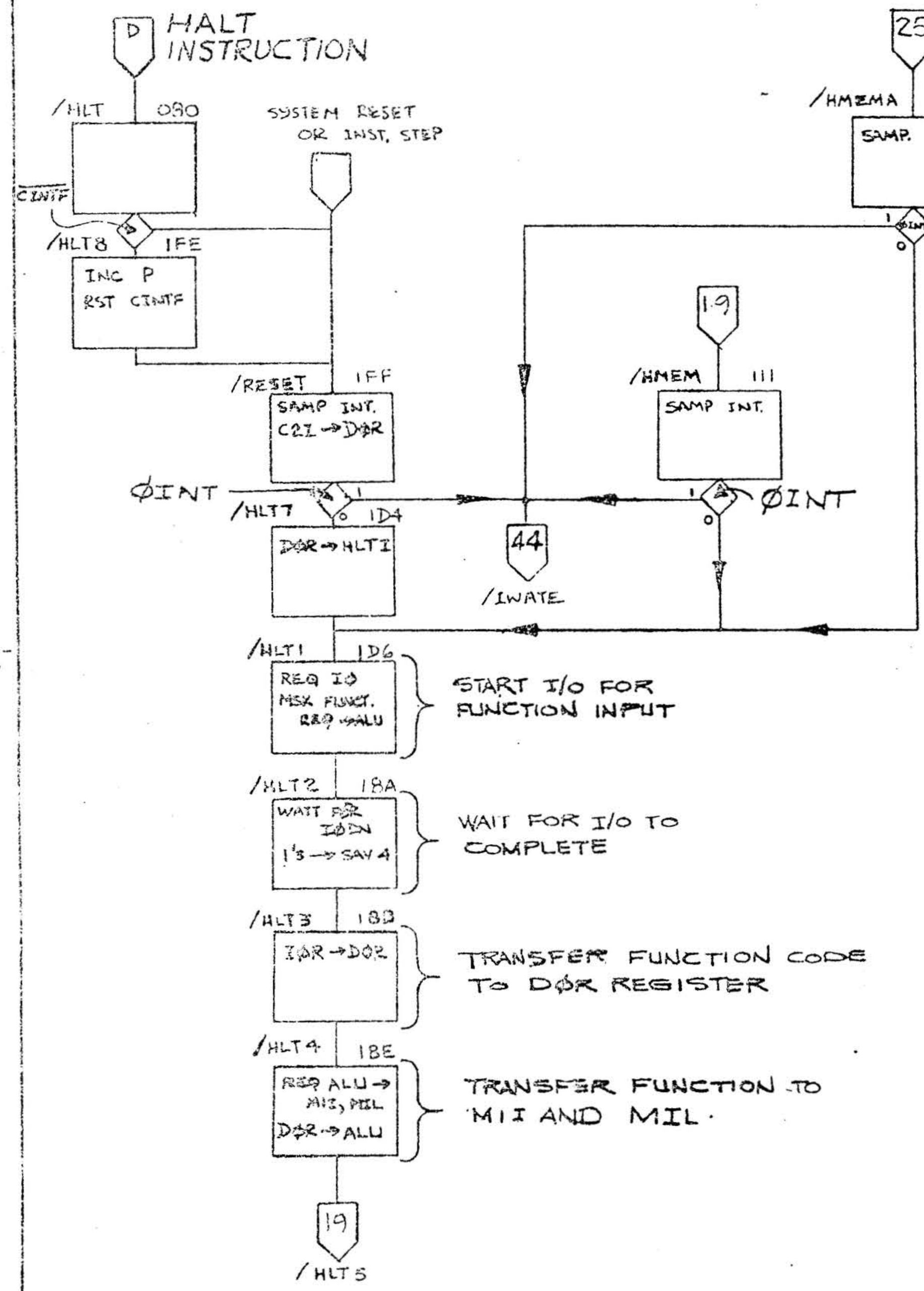
varian data machines, a varian subsidiary	CODE IDENT. NO.	IMMEDIATE INSTRUCTIONS	93AC887	REV. B
	PREPARED BY	APPROVED BY	SHEET 16 OF 57	

93A7043-098A

2.2.3 FRONT PANEL ROUTINES

PROGRAMMER	PROGRAM NO.	DATE	PAGE
CHART I.D.	CHART NAME	PROGRAM NAME	

2.2.3.1 HALT LOOP



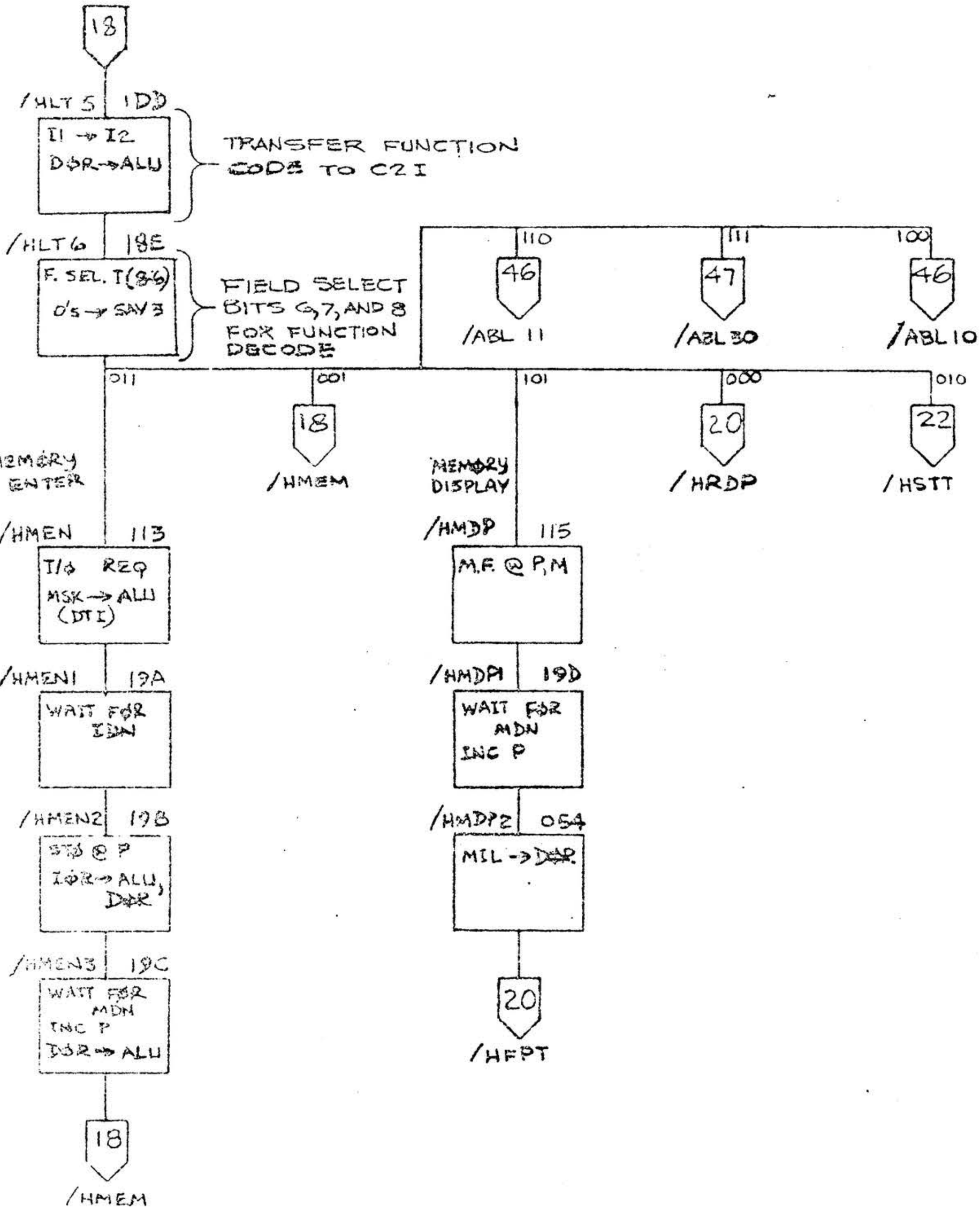
FOLD WHERE INDICATED

FOLD WHERE INDICATED AT DOTTED LINE

varian data machines, a varian subsidiary	IDENT. NO.	HALT LOOP	98A0887	REV. B
	PREPARED BY	APPROVED BY	SHEET 18 OF 57	

03A3843-003A

PROGRAMMER	PROGRAM NO.	DATE	PAGE
CHART I.D.	CHART NAME	PROGRAM NAME	



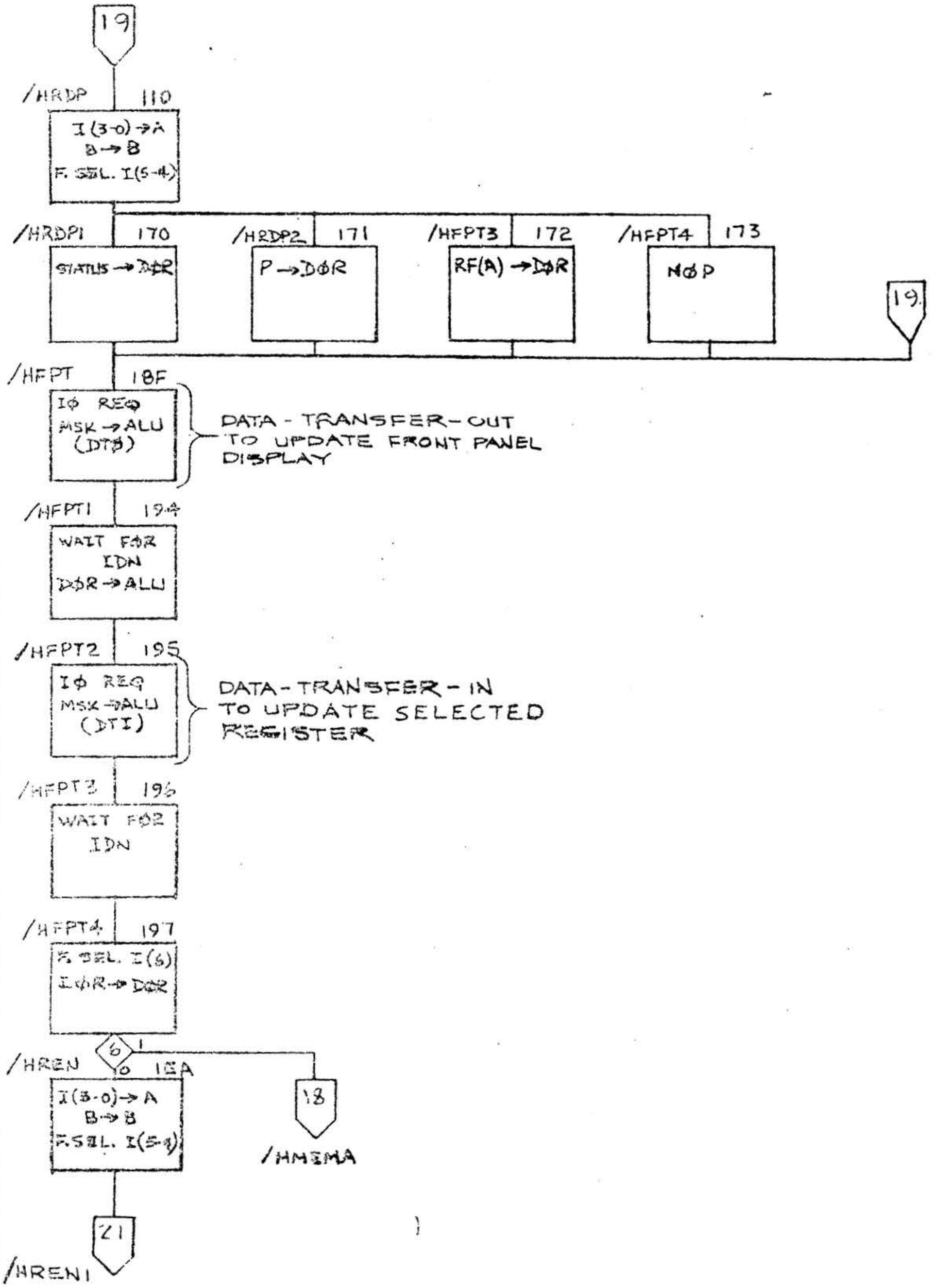
FOLD BACK AT DOTTED LINE

FOLD UNDER AT DOTTED LINE

varian data machines, a varian subsidiary	CODE IDENT. NO.	HALT LOOP	93A0887	REV. B
	PREPARED BY	APPROVED BY	SHEET 19 OF 57	

PROGRAMMER	PROGRAM NO.	DATE	PAGE
CHART I.D.	CHART NAME	PROGRAM NAME	

2.2.3.2 REGISTER DISPLAY / ENTER



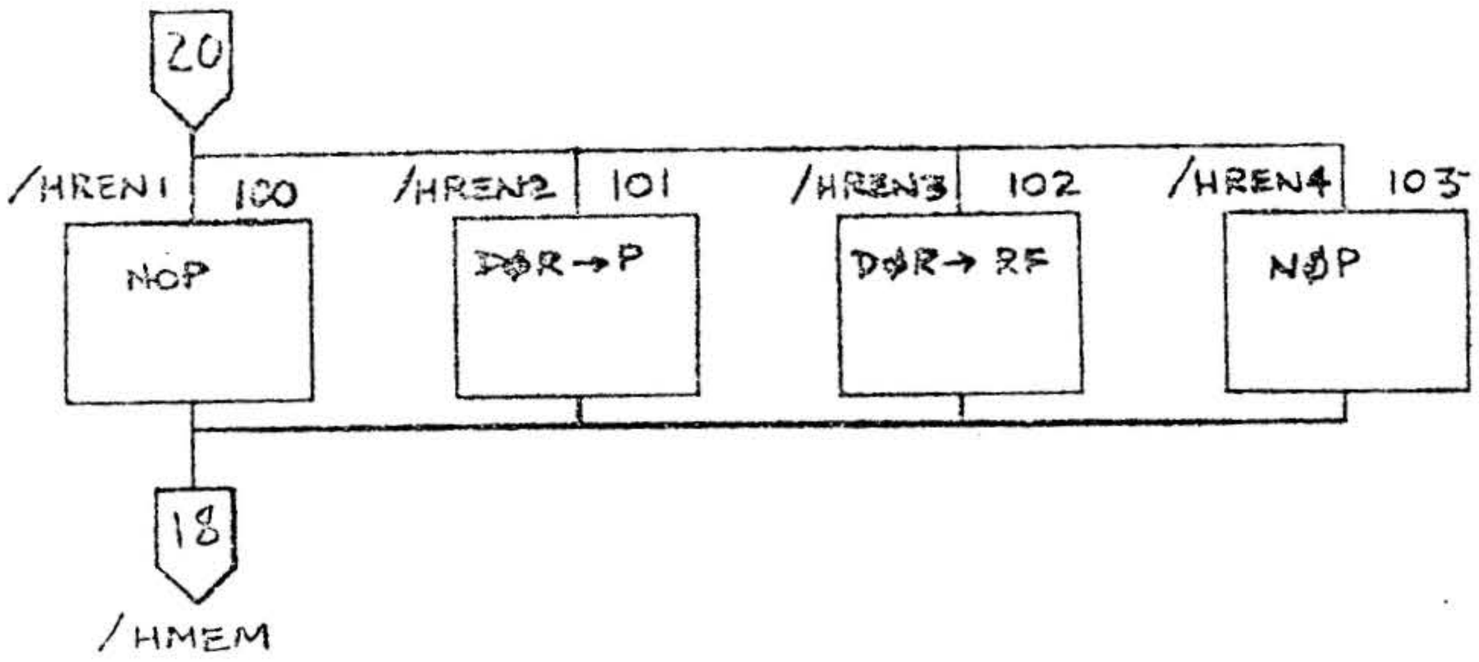
FOLD UNDER AT DOTTED LINE

FOLD UNDER AT DOTTED LINE

varian data machines, a varian subsidiary	CODE IDENT. NO.	REGISTER DISPLAY / ENTER	98A0887	REV. E
	PREPARED BY	APPROVED BY	SHEET 20 OF 57	

98A0884-019A

PROGRAMMER	PROGRAM NO.	DATE	PAGE
CHART I.D.	CHART NAME	PROGRAM NAME	



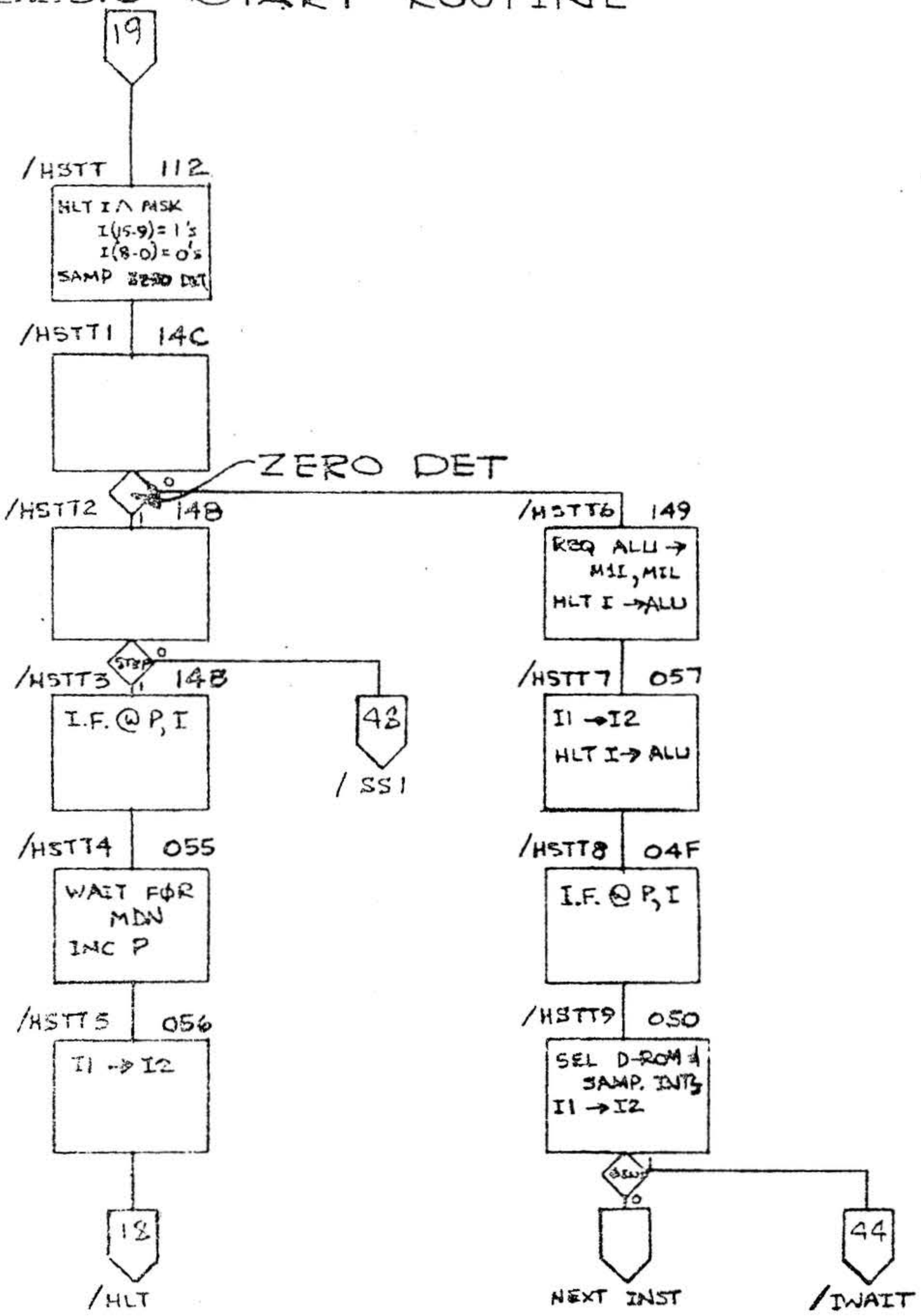
FOLD UNDER AT DOTTED LINE

FOLD UNDER AT DOTTED LINE

varian data machines, a varian subsidiary	CODE IDENT. NO.	REGISTER DISPLAY/ENTER	98A0887	REV. C
	PREPARED BY	APPROVED BY	SHEET 21 OF 57	

PROGRAMMER	PROGRAM NO.	DATE	PAGE
CHART I.D.	CHART NAME	PROGRAM NAME	

2.2.3.3 START ROUTINE



FOLD UNDER AT DOTTED LINE

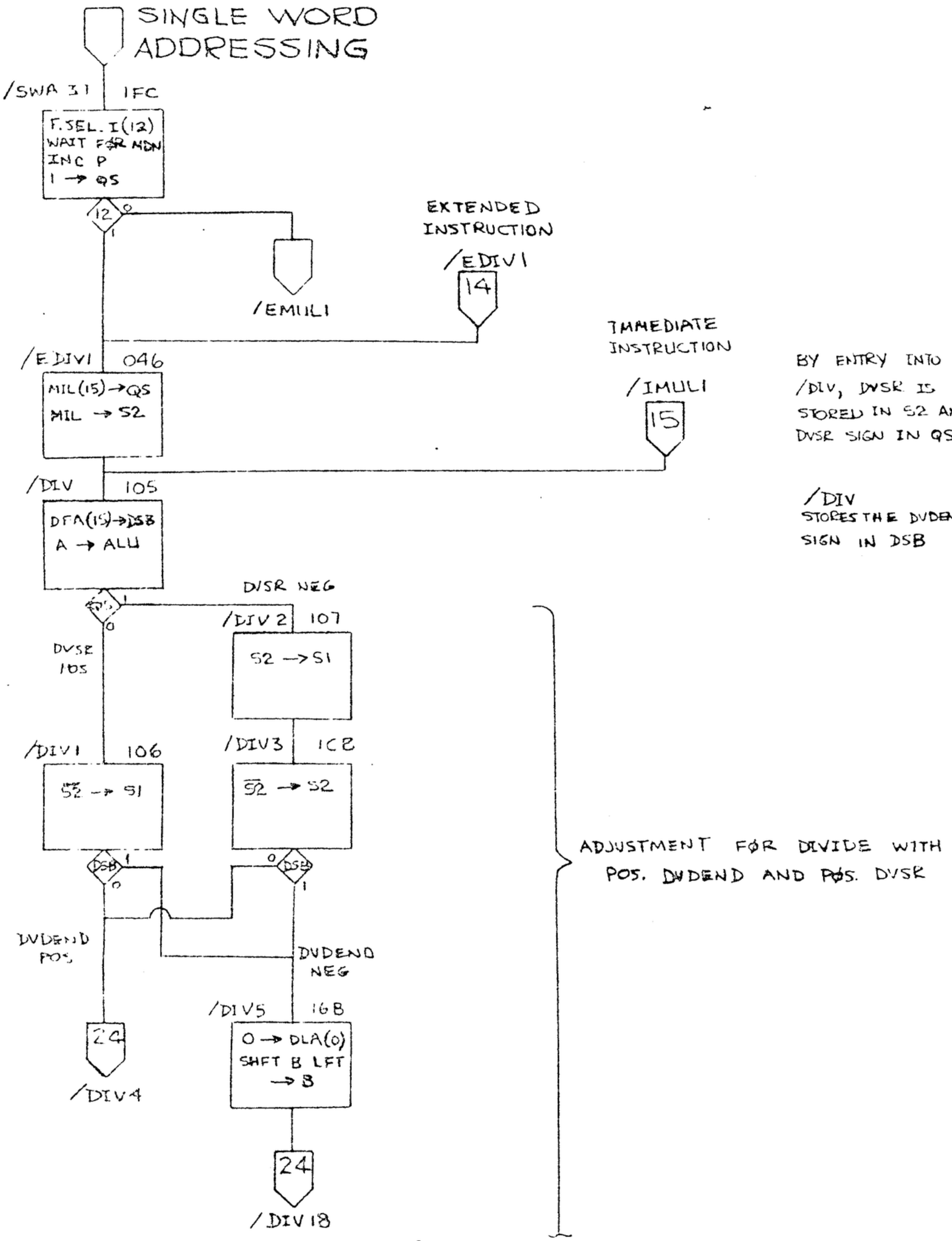
FOLD UNDER AT DOTTED LINE

varian data machines, a varian subsidiary	CODE IDENT. NO.	START ROUTINE	98A0887	REV.
	PREPARED BY	APPROVED BY	SHEET 22 OF 57	

98A0343-099A

PROGRAMMER	PROGRAM NO.	DATE	PAGE
CHART I.D.	CHART NAME	PROGRAM NAME	

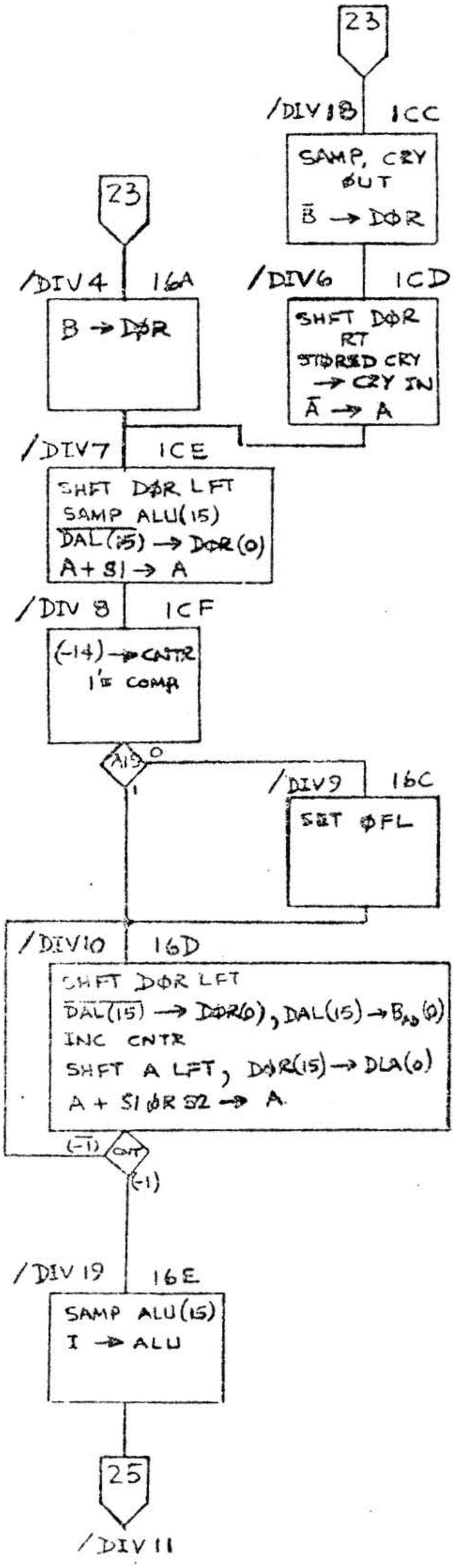
2.2.4 DIVIDE ROUTINE



varian data machines, a varian subsidiary	CODE IDENT. NO.	DIVIDE ROUTINE	98A0887	REV.
	PREPARED BY	APPROVED BY	SHEET 23 OF 57	

98A0343-990A

PROGRAMMER	PROGRAM NO.	DATE	PAGE
CHART I.D.	CHART NAME	PROGRAM NAME	



FIRST ADD AND TEST FØR ØFL

FOLD BACK AT DOTTED LINE

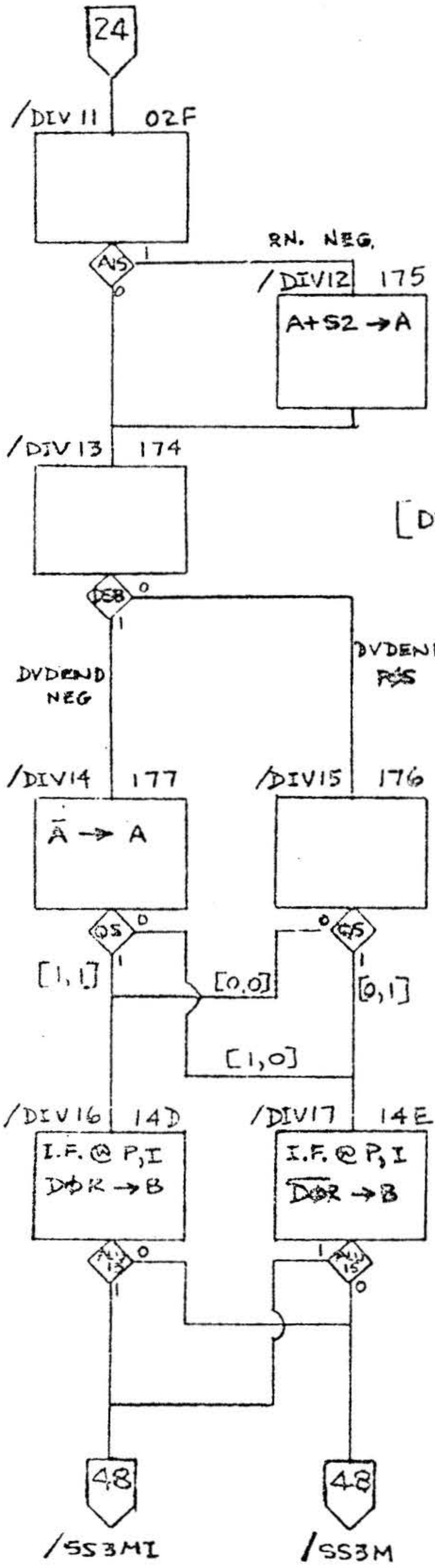
FOLD BACK AT DOTTED LINE

varian data machines, a varian subsidiary	CODE IDENT. NO.	DIVIDE	98A0887	REV. E
	PREPARED BY	APPROVED BY	SHEET 24 OF 57	

98A3243-909A

Instructions

PROGRAMMER	PROGRAM NO.	DATE	PAGE
CHART I.D.	CHART NAME	PROGRAM NAME	



REMAINDER TEST AND CORRECTION

RESTORE PROPER SIGN OF QUOT. AND RN.

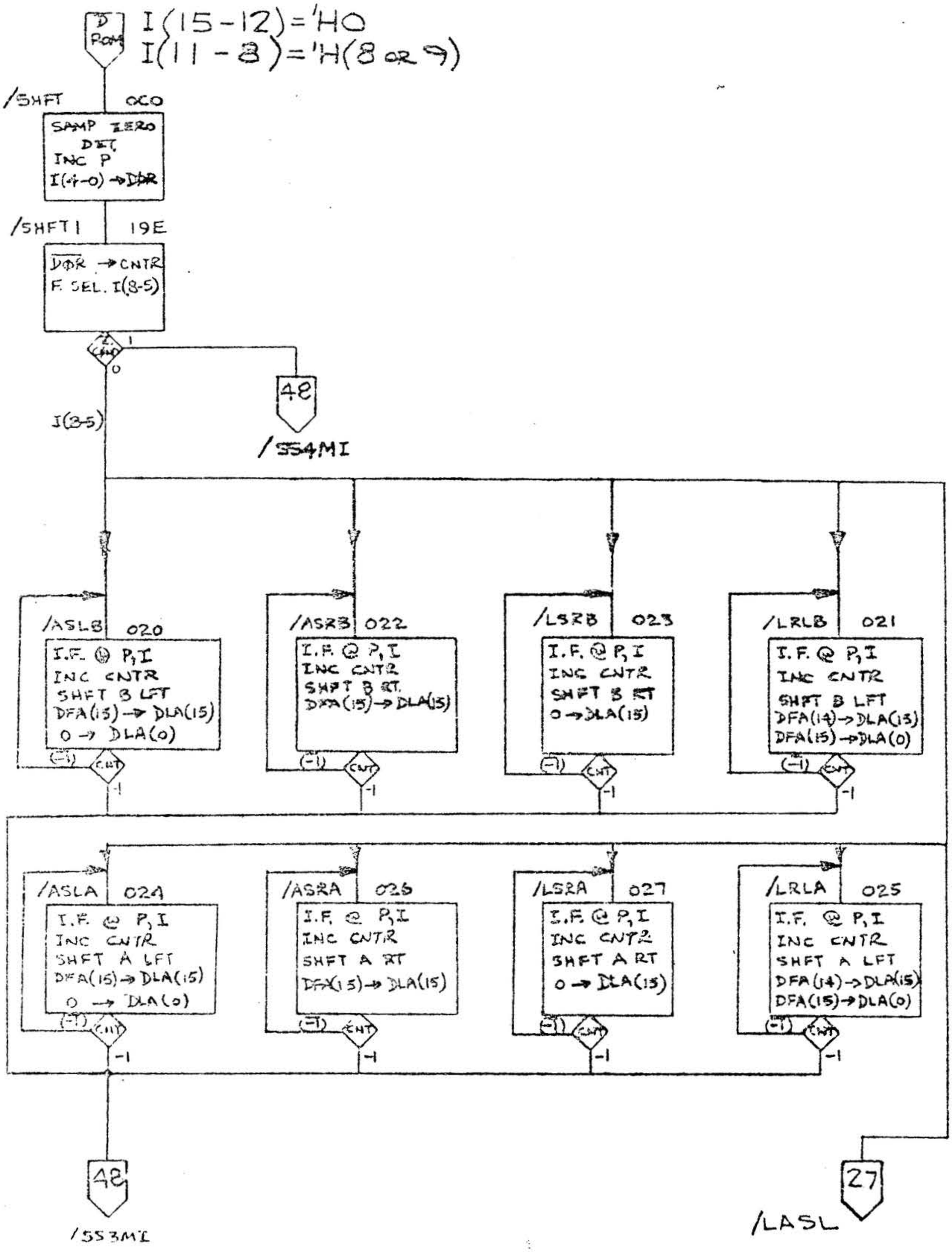
FOLD UNDER AT DOTTED LINE

FOLD UNDER AT DOTTED LINE

varian data machines, a varian subsidiary	CODE IDENT. NO.	DIVIDE	98A0887	REV. D
	PREPARED BY	APPROVED BY	SHEET 25 OF 57	

PROGRAMMER	PROGRAM NO.	DATE	PAGE
CHART I.D.	CHART NAME .	PROGRAM NAME	

2.2.5 SHIFT INSTRUCTIONS



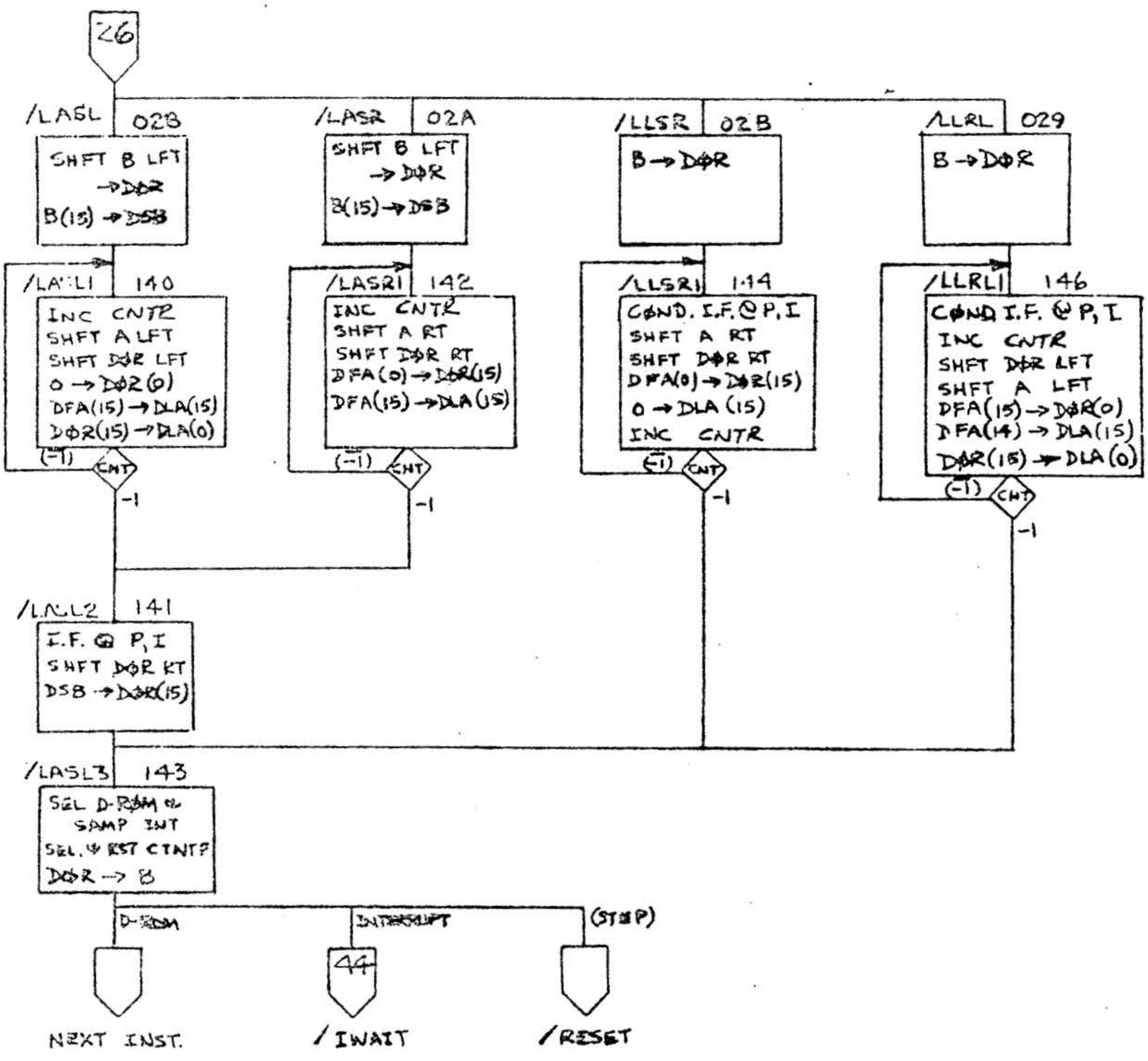
FOLD UNDER AT DOTTED LINE

FOLD UNDER AT DOTTED LINE

varian data machines, a varian subsidiary	IDENT. NO.	SHIFT INSTRUCTIONS	98A0887	REV. D
	PREPARED BY	APPROVED BY	SHEET 26 OF 57	

PROGRAMMER	PROGRAM NO.	DATE	PAGE
CHART I.D.	CHART NAME	PROGRAM NAME	

SHFT DECODE
I(8-5)



FOLD UNDER AT DOTTED LINE

FOLD UNDER AT DOTTED LINE

varian data machines, a varian subsidiary	CODE IDENT. NO.	SHIFT INSTRUCTIONS	98A0887	REV. B
	PREPARED BY	APPROVED BY	SHEET 27 OF 57	

98A0349-209A

PROGRAMMER	PROGRAM NO.	DATE	PAGE
CHART I.D.	CHART NAME	PROGRAM NAME	

2.2.6 JMP, JMARK AND EXECUTE INSTRUCTIONS

JUMP INSTRUCTION GROUP

D FROM I(15-12) = 'H0
I(11-8) = 'H(2 OR 3)

/JMP 090
INC P
COND IF/AF @ M,I
RST CINTF

0 → 48 → /551M
1 →

/JMP1 120
MIL → P
EN JMP 316

MIL 15: 0 →
1 → /JMP2 095
I.F/AF @ M,I
INC CNTR

/JMP2: 48 → /552M

JMARK INSTRUCTION GROUP

D FROM I(15-12) = 'H0
I(11-8) = 'H(4 OR 5)

/JMRK 0A0
COND. AF@ M,M
RST CINTF

0 → /SS5M 168
I.F.@ P,I
INC P
→ 48 → /552M

1 →

/JMRK1 169
COND ST0 ON MIL(15) = 0
P+1 → ALU

MIL 15: 0 →
1 → /JMRK4 0A2
AF @ M,M
INC CNTR

/JMRK2 0A1
P+1 → ALU
WAIT FOR MDN
→ /JMRK3 1CA
I.F. @ A,I
MIL+1 → P, ALU
→ 42 → /552M

EXECUTE INSTRUCTION GROUP

D FROM I(15-12) = 'H0
I(11-8) = 'H(6 OR 7)

/XEC 0B0
COND LF/AF @ M,I
RST CINTF

MIL 14: 0 →
1 → /SS5 060
I.F. @ P,I
INC P

/XEC1 061

MIL 13: 0 →
1 → /XEC2 0B1
I.F/AF @ M,I
INC CNTR

/XEC2: 48 → /552

FOLD UNDER AT DOTTED LINE

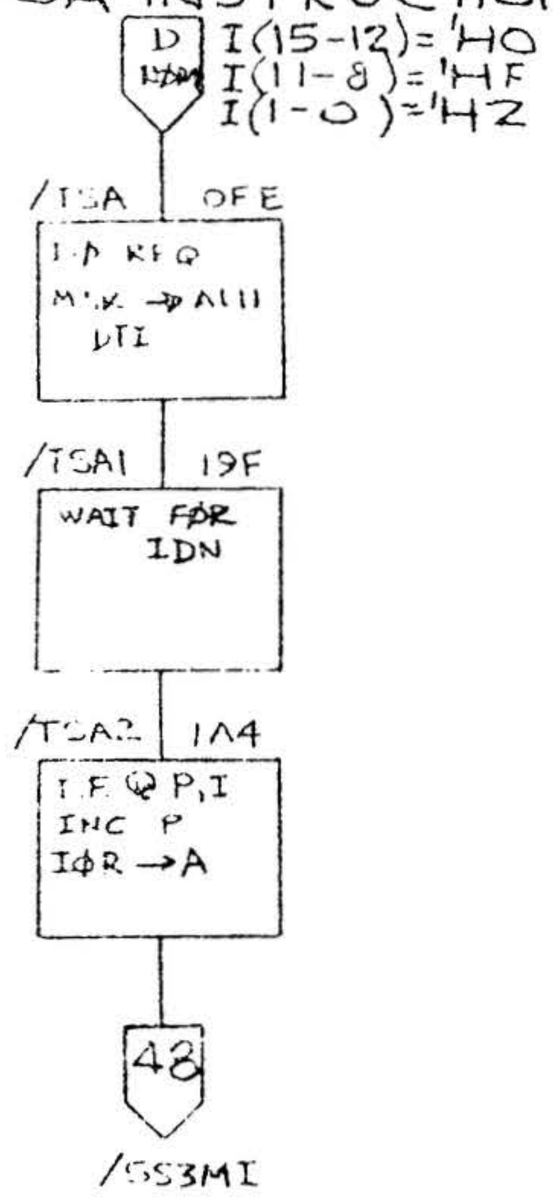
FOLD UNDER AT DOTTED LINE

varian data machines, a varian subsidiary	CODE IDENT. NO.	JMP, JMARK, & EXECUTE INSTRUCTIONS	98A0887	REV. D
	PREPARED BY	APPROVED BY	SHEET 28 OF 57	

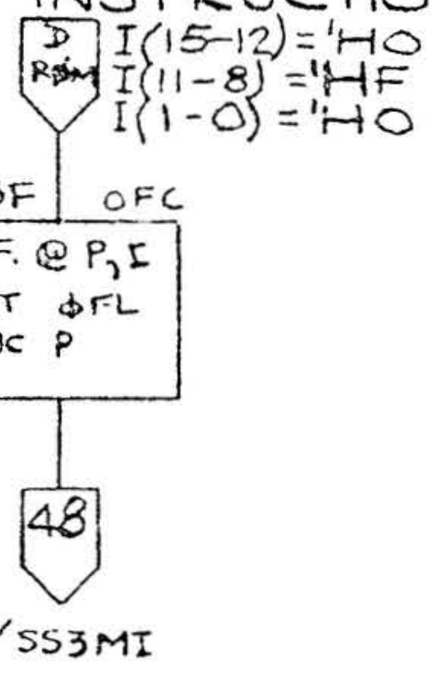
PROGRAMMER	PROGRAM NO.	DATE	PAGE
CHART I.D.	CHART NAME	PROGRAM NAME	

2.2.7 TSA, RØF, AND SØF INSTRUCTIONS

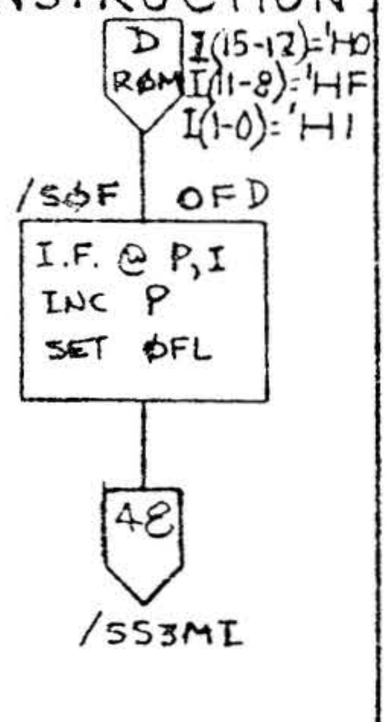
TSA INSTRUCTION



RØF INSTRUCTION



SØF INSTRUCTION



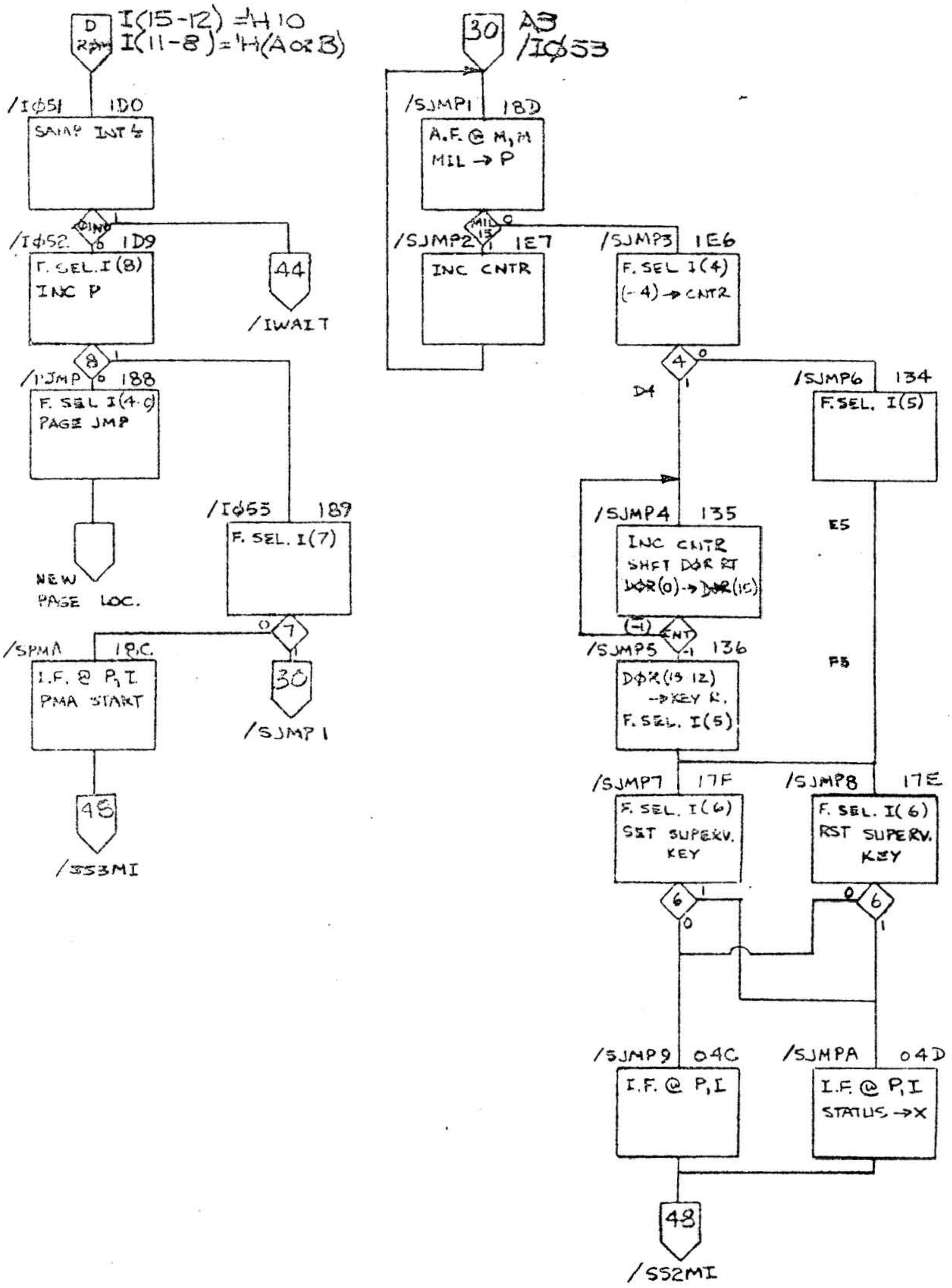
FOLD UNDER AT DOTTED LINE

FOLD UNDER AT DOTTED LINE

varian data machines, a varian subsidiary	CODE IDENT. NO.	TSA, RØF, AND SØF INSTRUCTIONS	98A0887	REV. D
	PREPARED BY	APPROVED BY	SHEET 29 OF 57	

PROGRAMMER	PROGRAM NO.	DATE	PAGE
CHART I.D.	CHART NAME	PROGRAM NAME	

2,2,8 SPMA, BCS, SJUMP INSTRUCTIONS

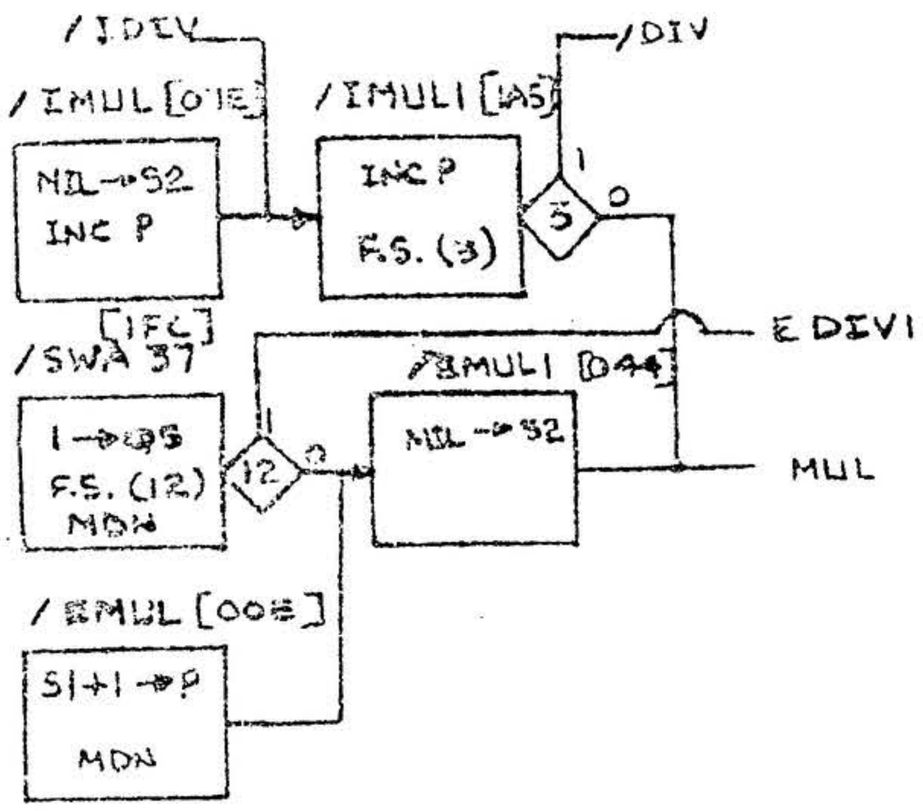


FOLD UNDER AT DOTTED LINE

FOLD UNDER AT DOTTED LINE

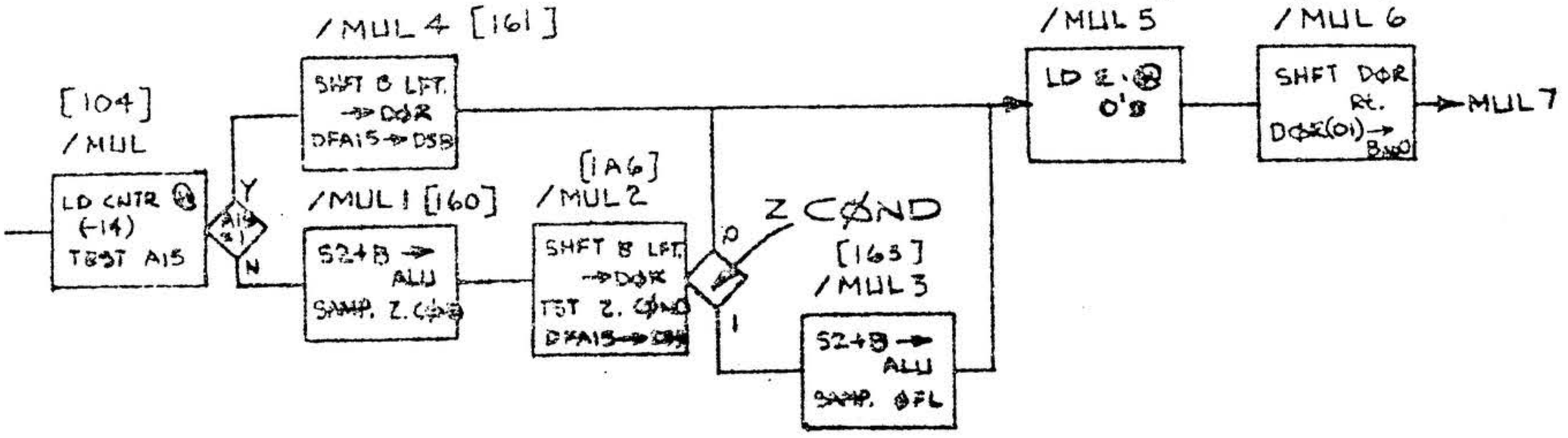
varian data machines, a varian subsidiary	CODE IDENT. NO.	SPM, BCS, SJUMP INSTRUCTIONS	98AC887	REV. B
	PREPARED BY	APPROVED BY	SHEET 30 OF 57	

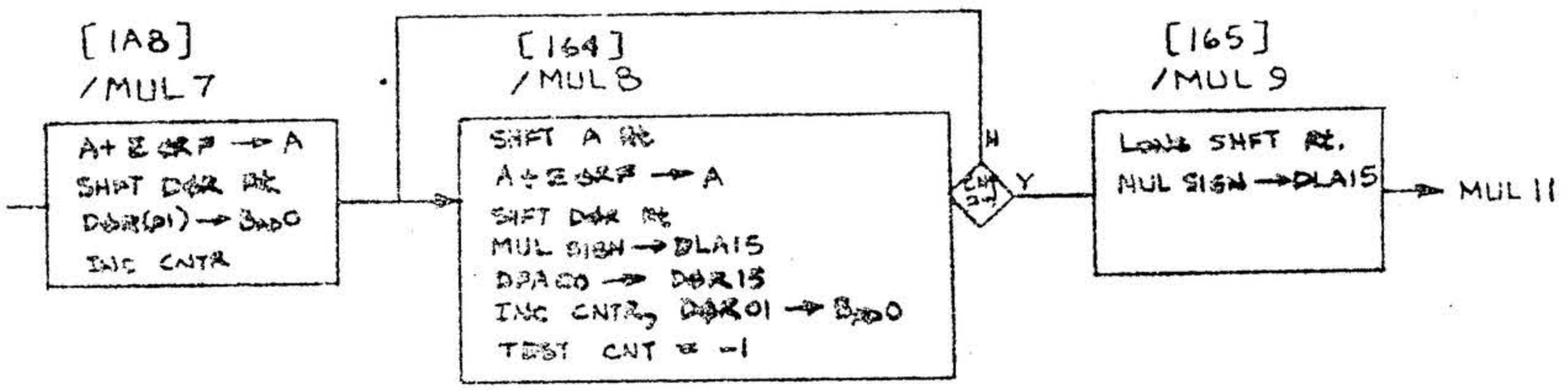
2.2.) MULTIPLY

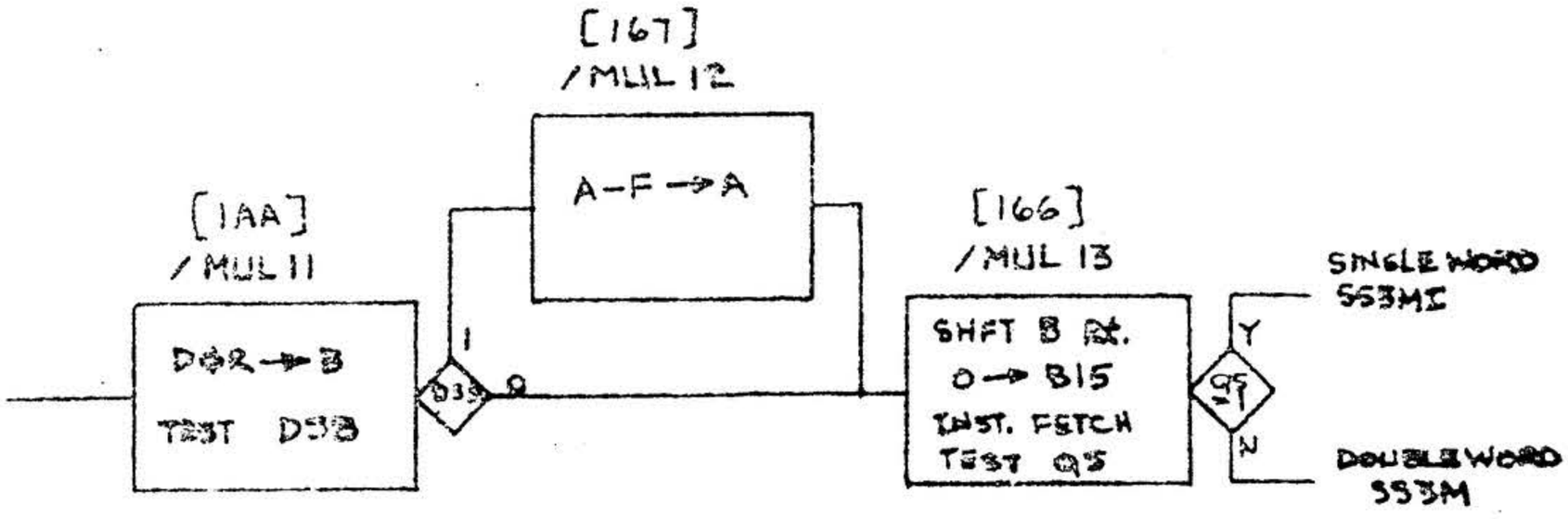


NOTE:

1. [XXX] ≡ HEX LOCATION OF MICROWORD
2. /XXXXX ≡ MICROWORD LABEL

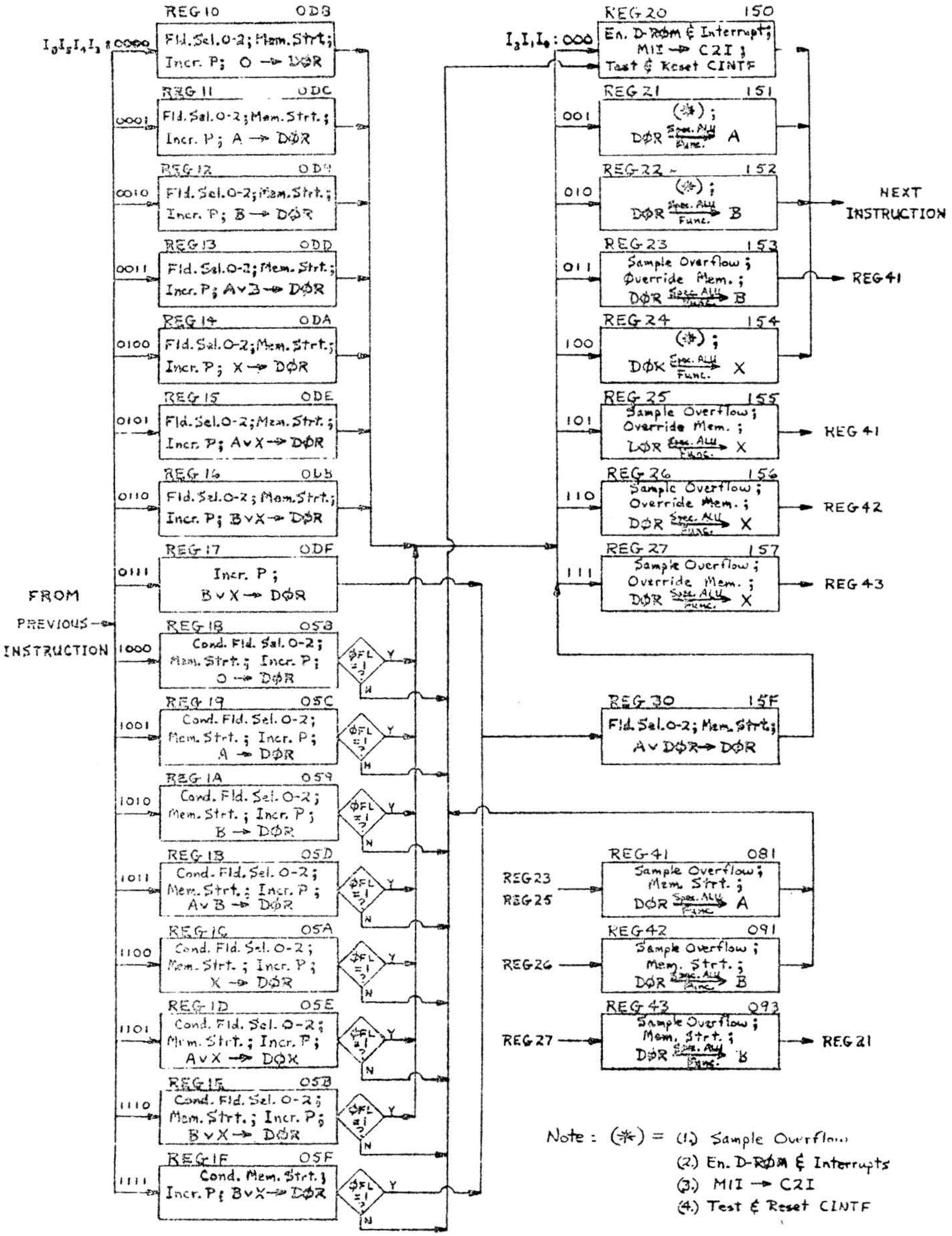






varian data machines	CODE IDENT NO.	MULTIPLY	98A0887	REV B
	PREPARED	APPR	SHEET 31 OF 57	

22.10 REGISTER TO REGISTER INSTRUCTIONS



Note: (*) = (1.) Sample Overflow
(2.) En. D-RØM & Interrupts
(3.) MII → C2I
(4.) Test & Reset CINTF

varian data machines	CODE IDENT NO	REGISTER TO REGISTER INSTRUCTIONS	98A0387	REV B
	PREPARED D.M.	APPR	SHEET 32 OF 57	

2.2.11 SINCE - WORD ADDRESSING INSTRUCTIONS

FROM PREVIOUS INSTRUCTION

$I_{11}I_{10}I_9$:

- 000 → SWA10 1E2: Mem. Strt.; Mask I → DØR
- 001 → SWA11 192: Mem. Strt.; Mask I → DØR
- 010 → SWA12 1A2: Mem. Strt.; Mask I → DØR
- 011 → SWA13 1B2: Mem. Strt.; Mask I → DØR
- 100 → SWA14 1C2: Mem. Strt.; Mask I + P → DØR
- 101 → SWA15 1D2: Mem. Strt.; Mask I + X → DØR
- 110 → SWA16 1E2: Mem. Strt.; Mask I + B → DØR
- 111 → SWA17 1F2: Mem. Strt.; Mask I → Mem. Contr.

SWA20 12F: Field Select 13-15

SWA21 1C5: Mem. Start

SWA22 13F: Cond. Fld. Sel. 13-15; MIL → DØR

SWA26 1FA: Mem. Start; Incr. Shift Count.

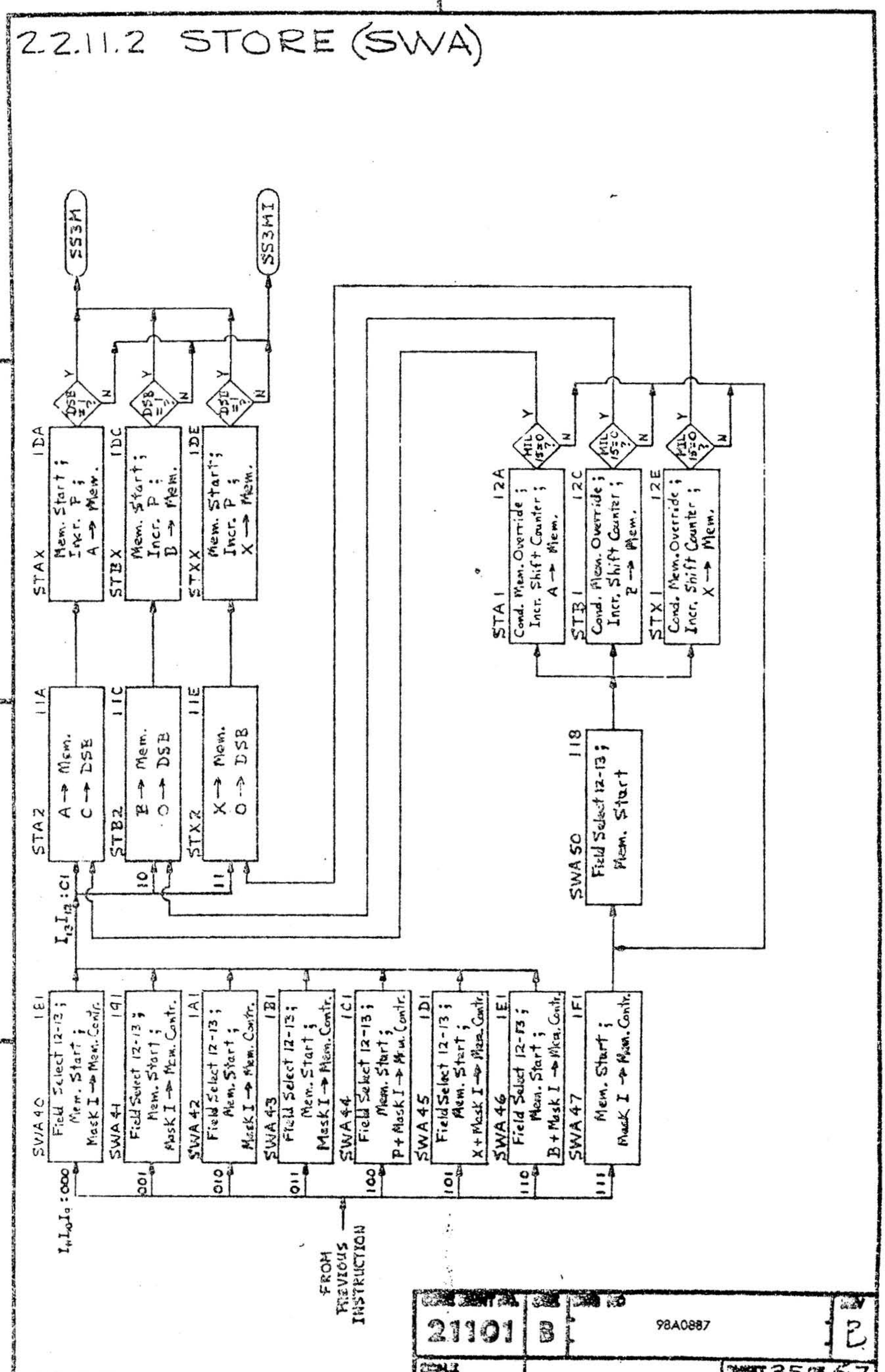
(MIL = 0 ?) N → SWA22; Y → $I_{15}I_{14}I_{13}$:

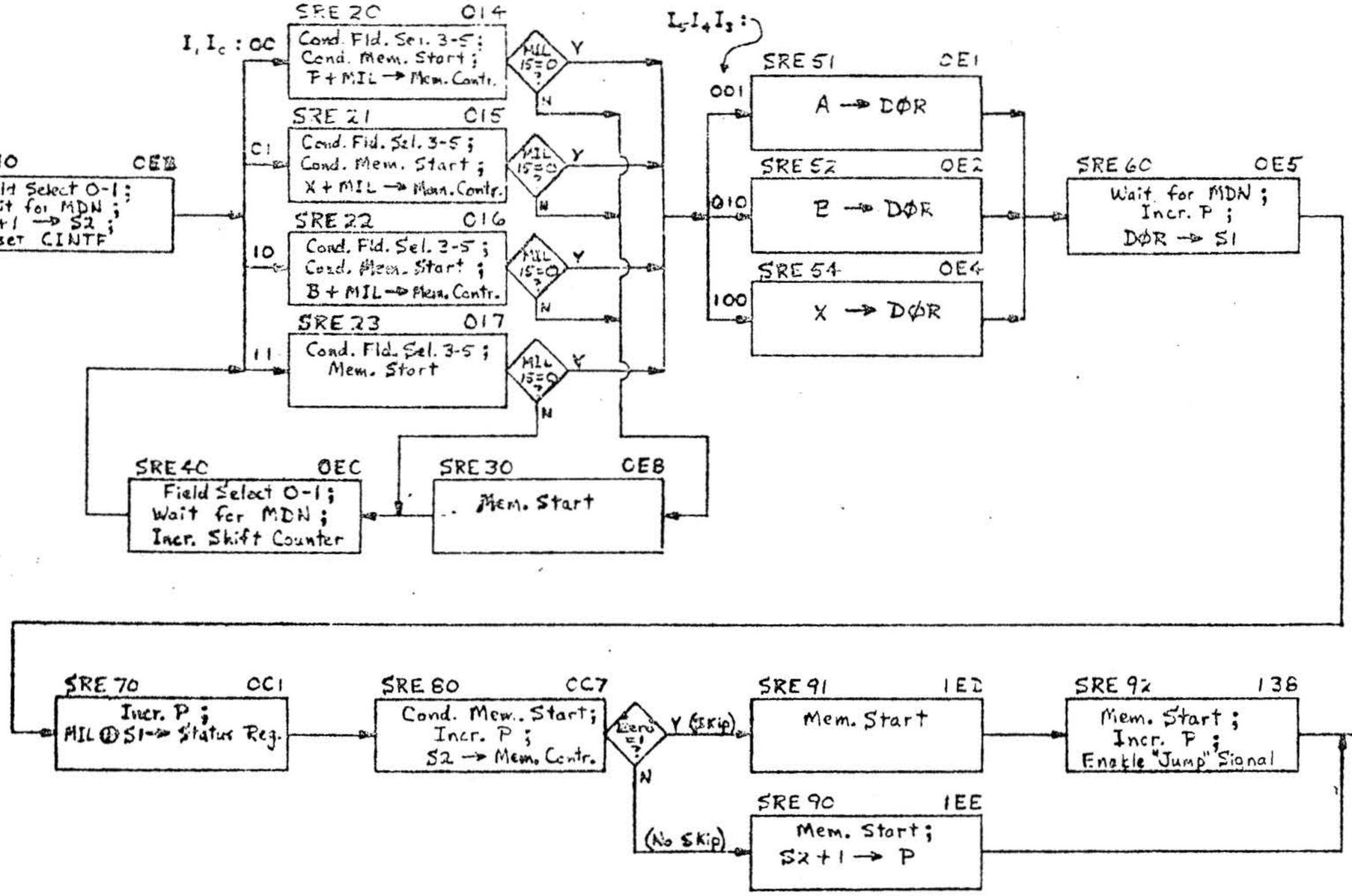
- 000 → LDA1 1E0: Mem. Start; Incr. P → LDA2 0B5: (*); MIL → A
- 001 → SWA31 1E4: Field Select 12; Mem. Strt.; Incr. P → I_{12}: 0 → LDB 084: (*); MIL → B; 1 → LDX 086: (*); MIL → X
- 010 → INR1 1E8: Mem. Start; DØR → Mem. Contr.; 0 → DSB → INRX 1DF: Sample Overflow; MIL+1 → Mem. → INR2 0BC: Mem. Start; Incr. P; MIL+1 → Mem. → (DSB = 0 ?) N → SS3MI; Y → SS3M
- 100 → ØRA1 1FC: Mem. Start; Incr. P → ØRA2 0BF: (*); A ∨ MIL → A
- 101 → SWA35 1F4: Field Select 12; Mem. Start; Incr. P → I_{12}: 0 → ADD 094: Sample Overflow; (*); A + MIL → A; 1 → ERA 096: (*); A ⊕ MIL → A
- 110 → SWA36 1F8: Field Select 12; Mem. Start; Incr. P → I_{12}: 0 → SUB 0A4: Sample Overflow; (*); MIL − A → A; 1 → ANA 0A6: (*); A • MIL → A
- 111 → SWA37 1FC: See MUL/DIV Flowcharts

NEXT INSTRUCTION

Note: (*) = (1.) Enable D-RØM & Interrupt
(2.) MII → C2I
(3.) Test & Reset CINTF

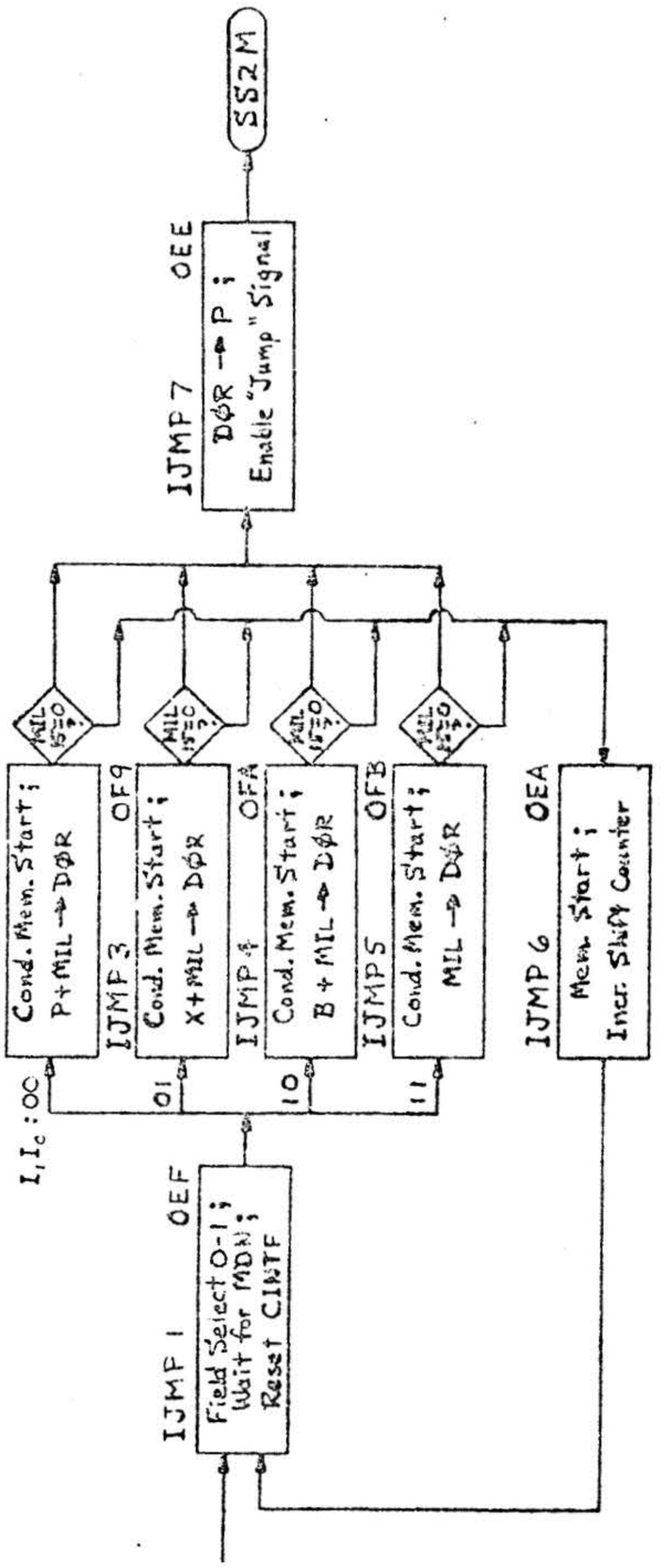
2.2.11.2 STORE (SWA)





SRE INSTRUCTION

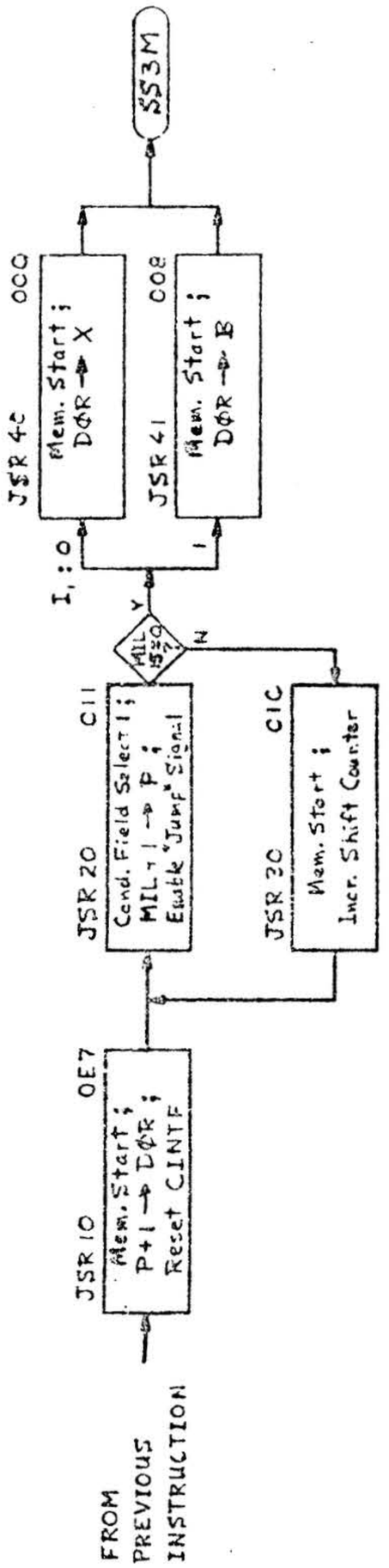
2.2.13 IJMP INSTRUCTION



IJMP INSTRUCTION

21101	B	98A0887	B
		SHEET 37 OF 57	

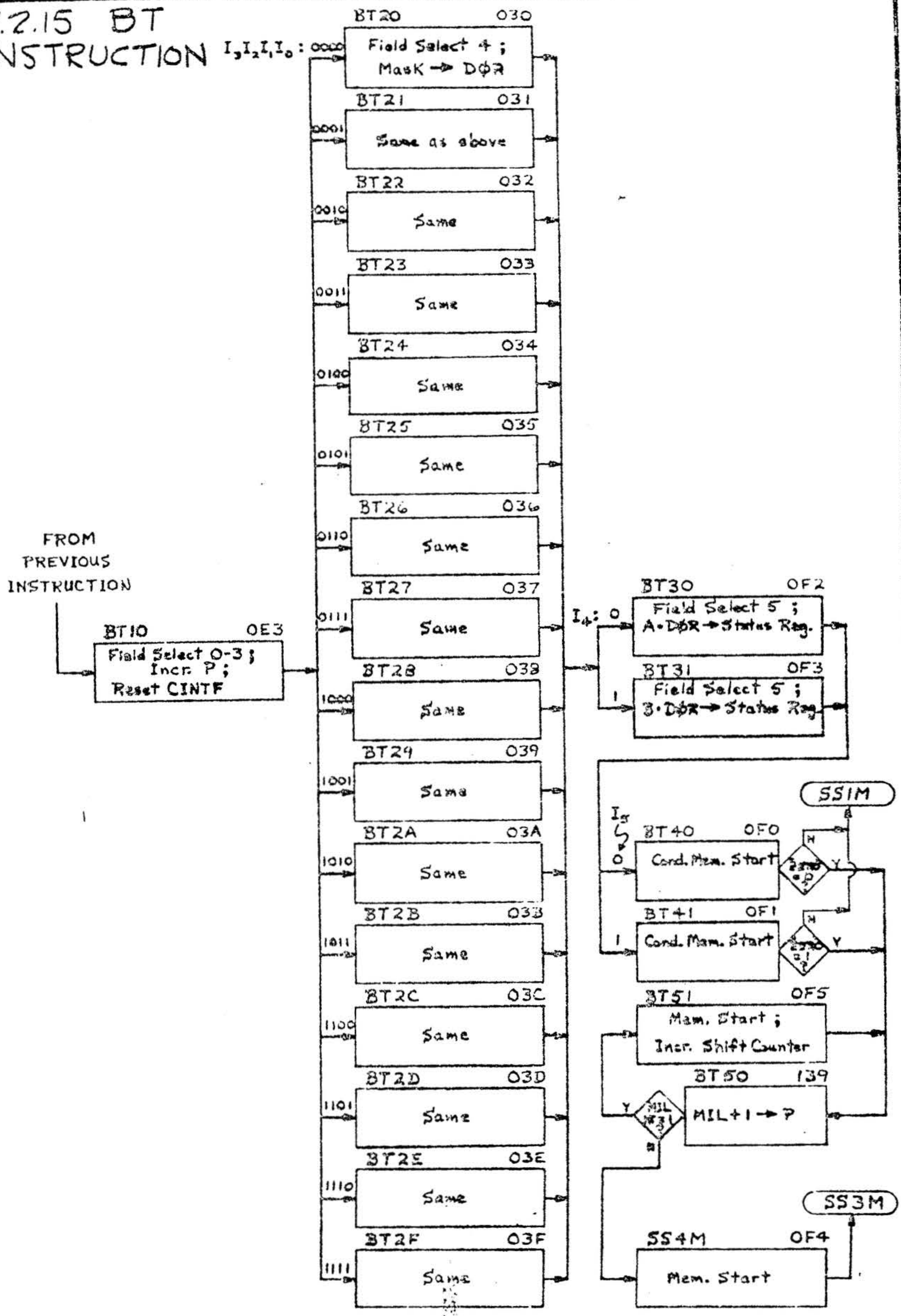
2.2.14 JSR INSTRUCTION



JSR INSTRUCTION

DOCUMENT NO.	SIZE	DWG NO	REV
21101	B	98A0867	B
SCALE		SHEET 38 OF 57	

2.2.15 BT INSTRUCTION



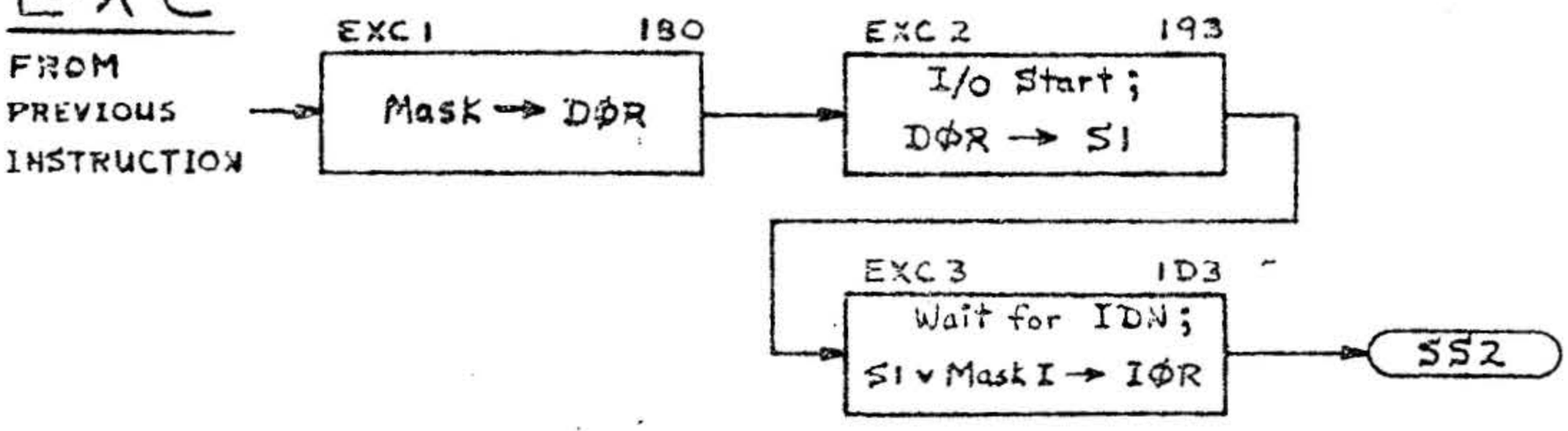
BT INSTRUCTION

21101	B	98A0837	B
		SHEET 39 OF 57	

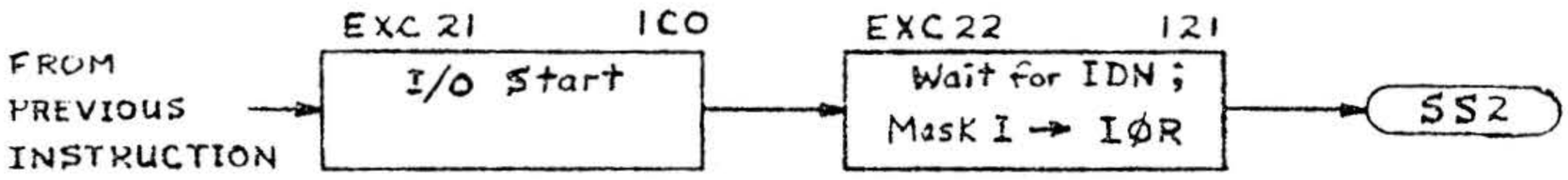
2.2.16 INPUT/OUTPUT (I/O) INSTRUCTIONS

2.2.16.1 I/O - EXC, EXC2, SEN

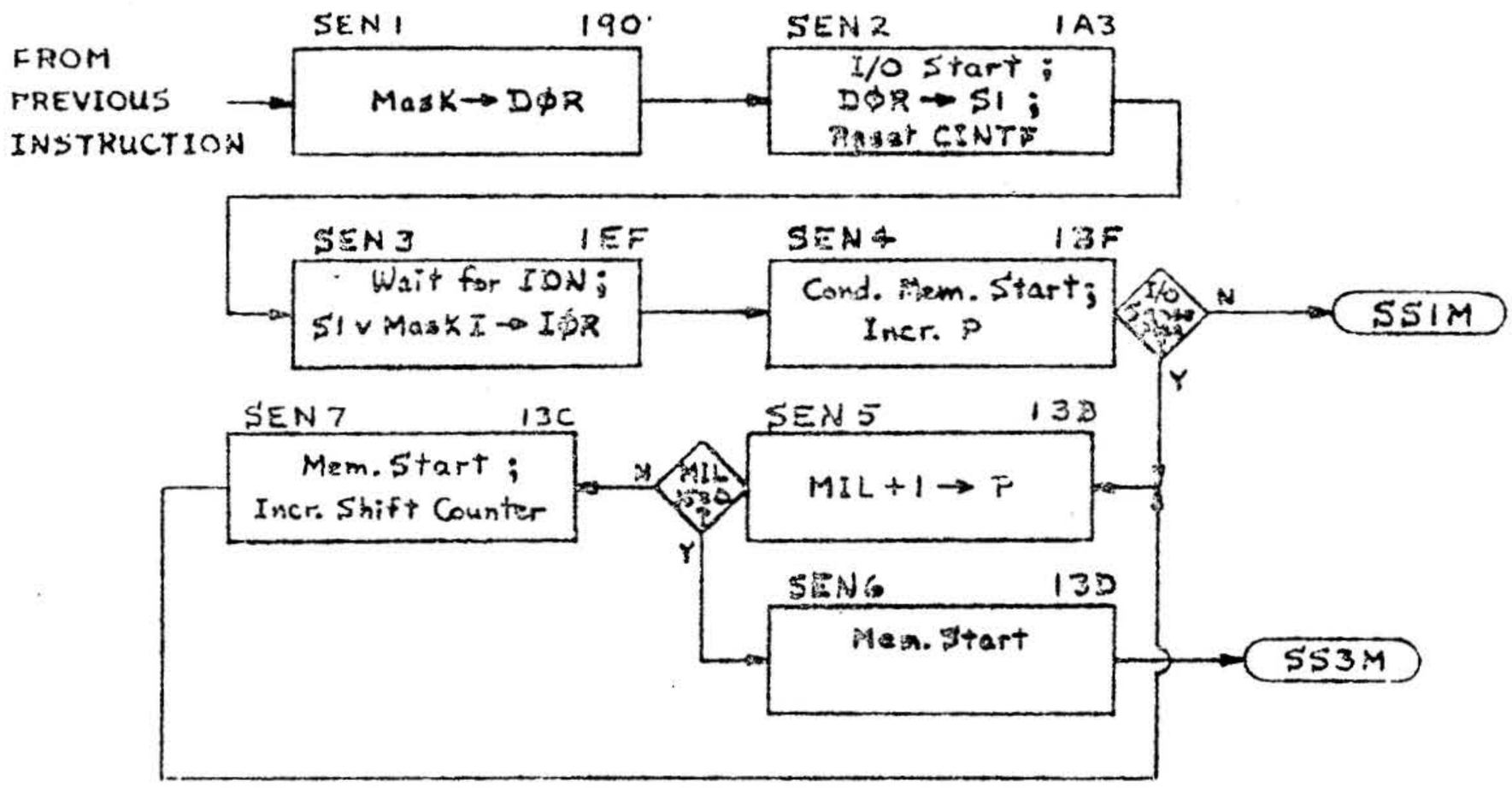
EXC



EXC 2

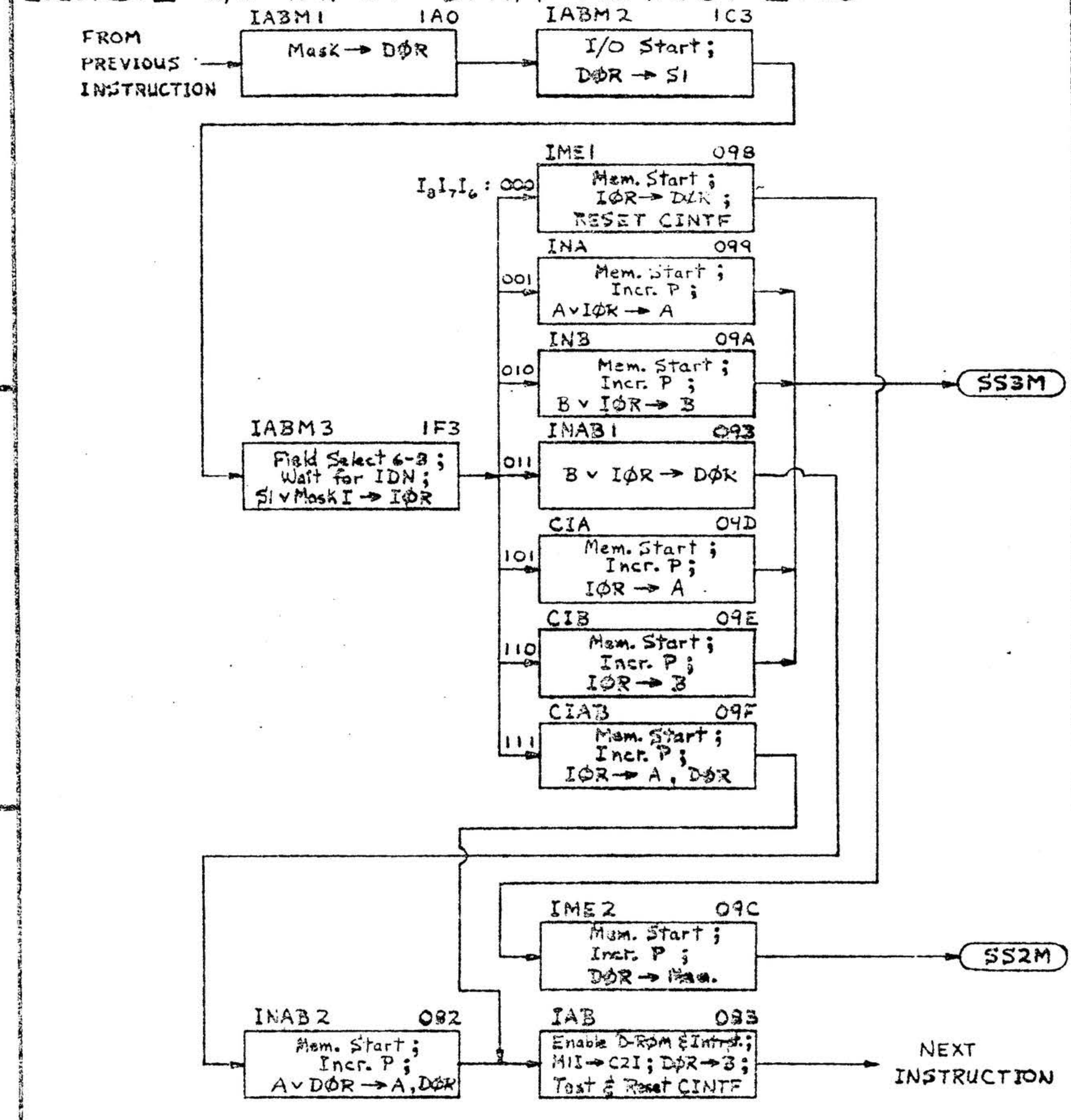


SEN



CODE IDENT NO.	SIZE	DWG NO.	REV
21101	B	98A0887	D
SCALE			SHEET 41 OF 57

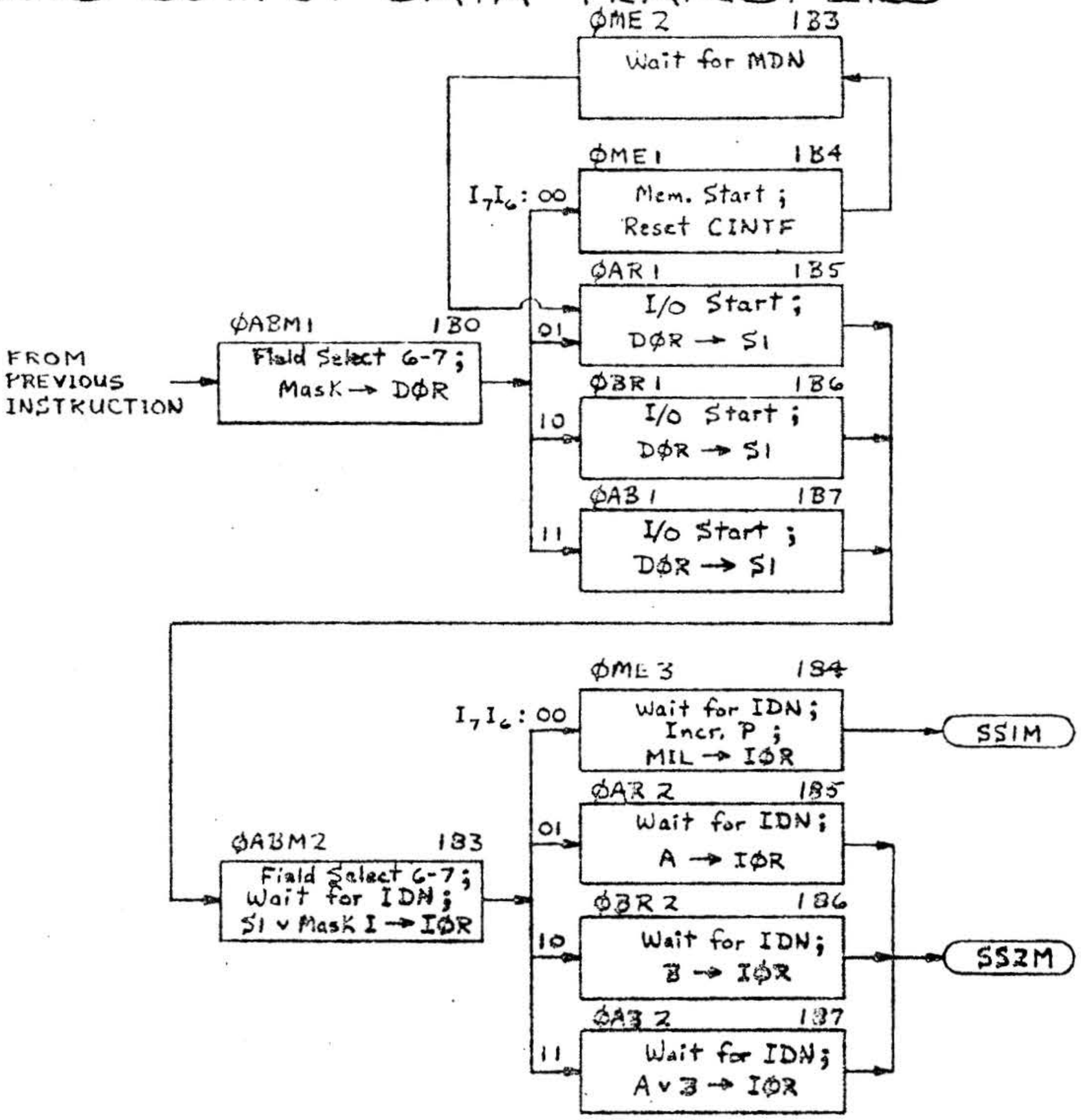
2.2.16.2 I/O INPUT DATA TRANSFERS



I/O INPUT DATA TRANSFER (CIA, CIB, CIAB, INA, INB, INAB, IME)

21101	B	98A0887	E
SCALE			SHEET 42 OF 57

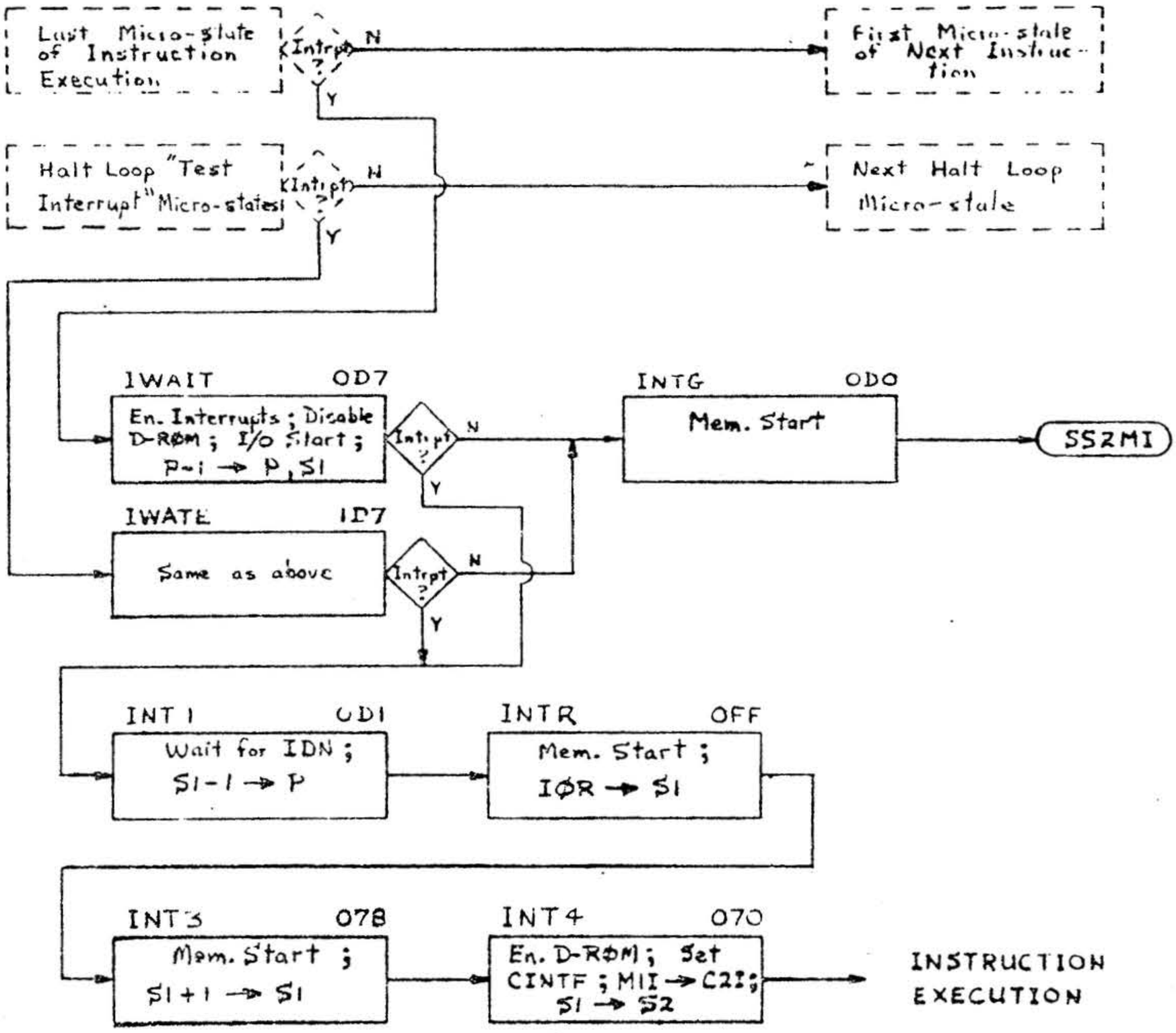
2.2.16.3 OUTPUT DATA TRANSFERS



OUTPUT DATA TRANSFERS
(ØAR, ØER, ØAB, ØME)

21101	B	98A0887	B
SCALE		SHEET 43 OF 57	

2.2.17 INTERRUPT ROUTINE



INTERRUPT ROUTINE

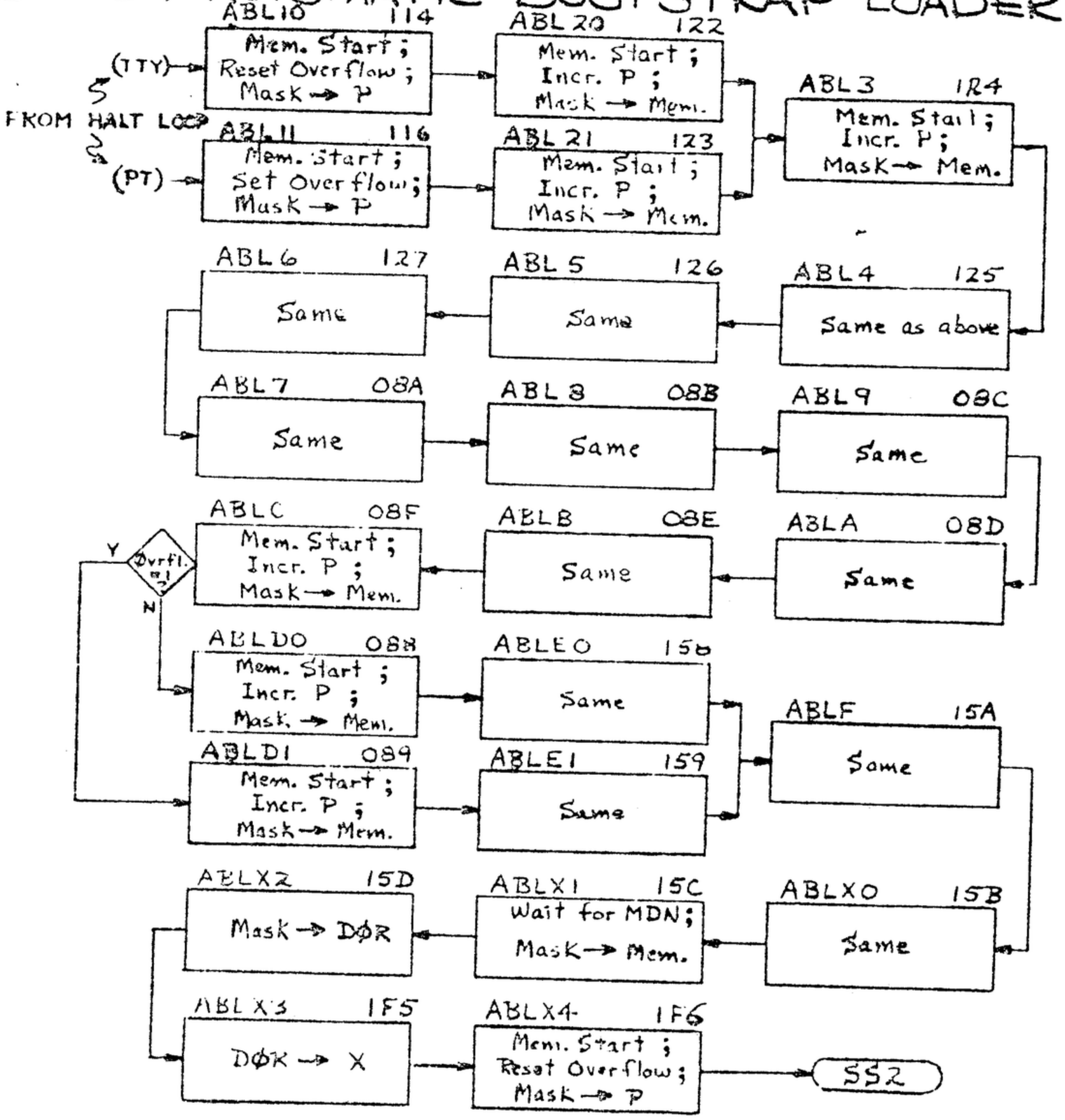
CODE IDENT NO.	SIZE	DWG NO	REV
21101	B	98A0887	D

SCALE

SHEET 44 of 57

2.2.18 AUTOMATIC BOOTSTRAP LOADER ROUTINES

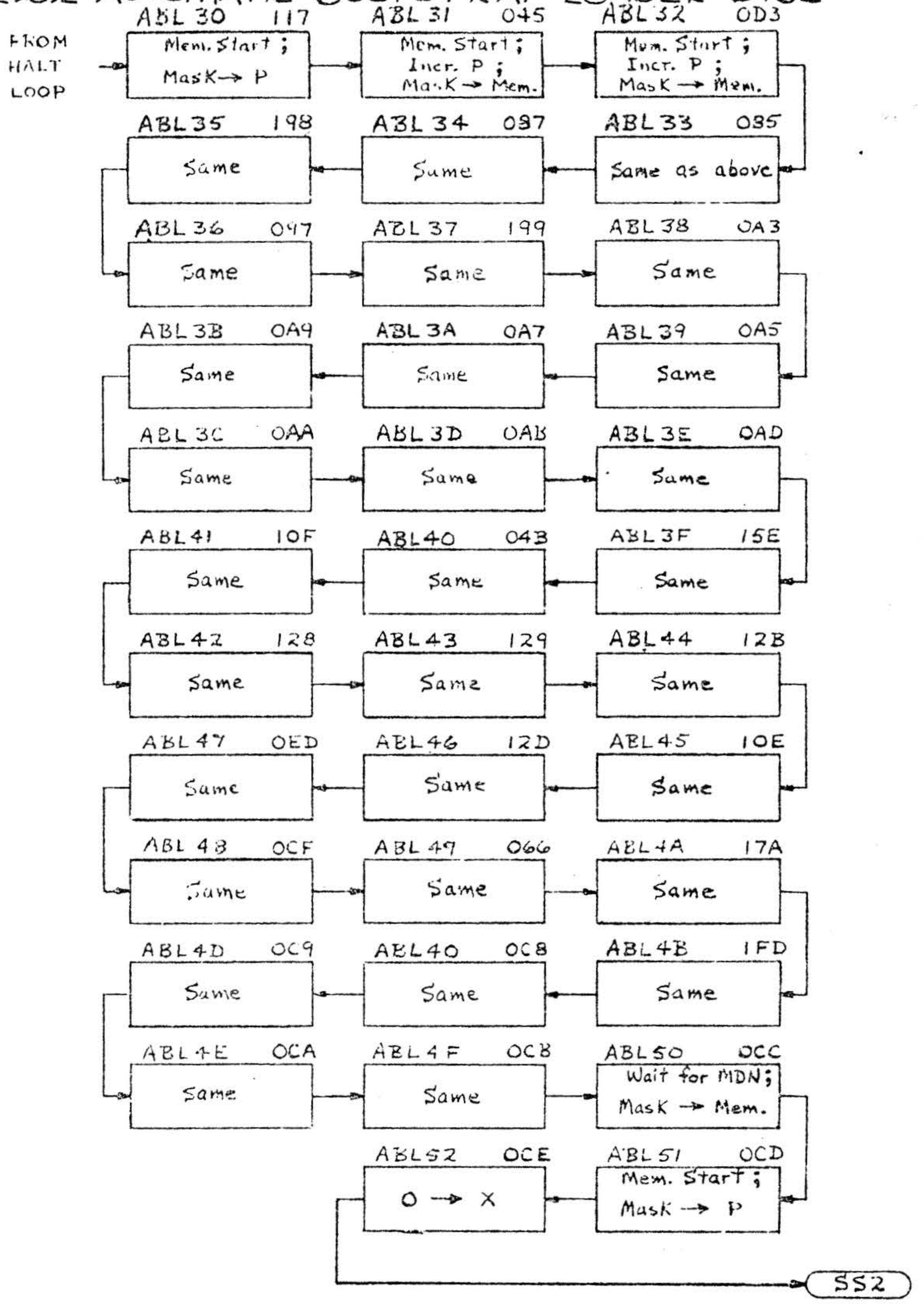
2.2.18.1 AUTOMATIC BOOTSTRAP LOADER TTY,PT



AUTO BOOTSTRAP LOADER TTY, PT

CODE IDENT. NO.	SIZE	DWG NO.	REV
21101	B	98A0887	B
SCALE			

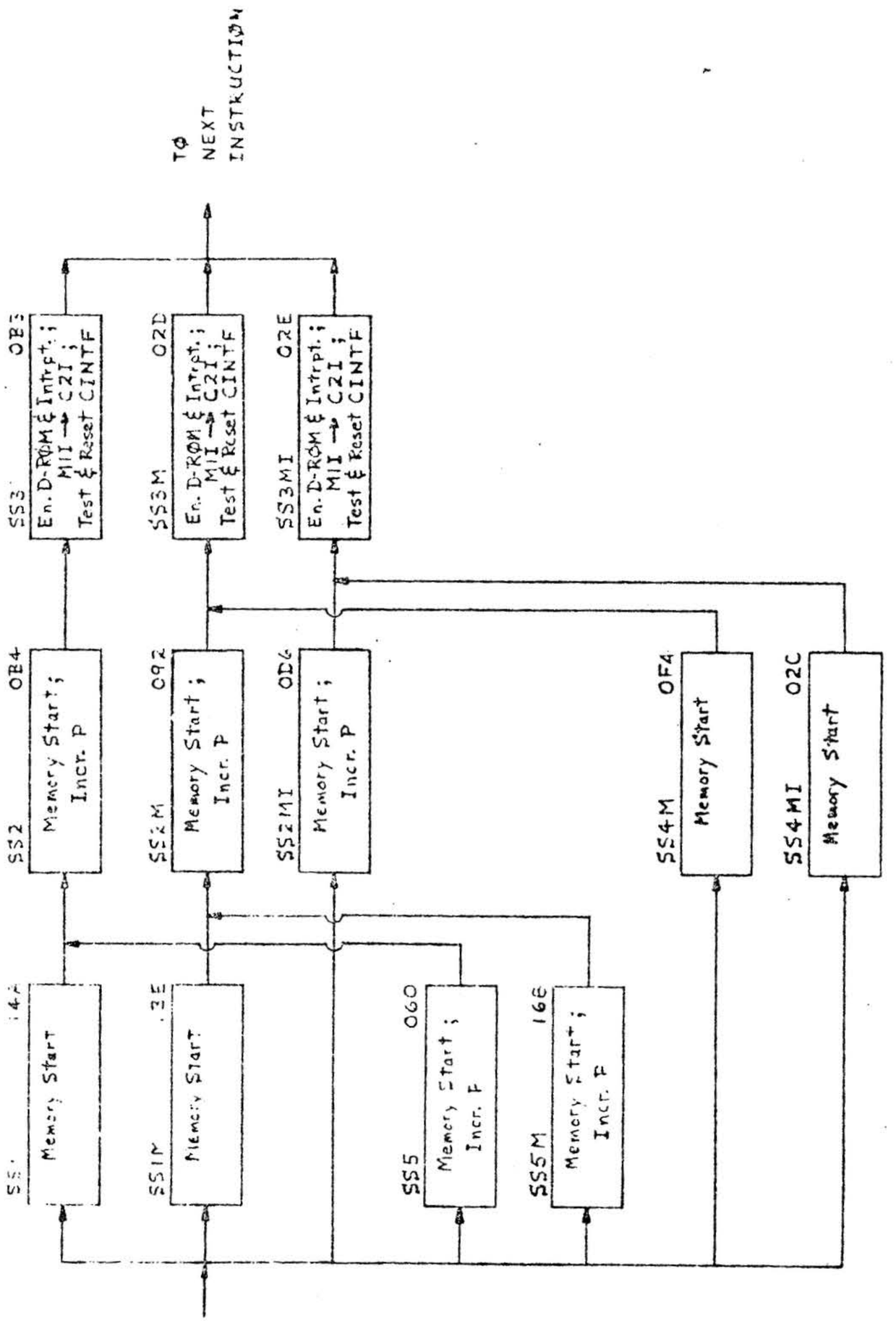
2.2.8.2 AUTOMATIC BOOTSTRAP LOADER DISC



AUTOMATIC BOOTSTRAP LOADER DISC

21101	B	98A0887	B
SCALE		SHEET 47 OF 57	

2.2.19 V70 "STANDARD" MICRO STATE



V70 "STANDARD" MICRO STATE

21101	B	98A0687	B
SCALE			

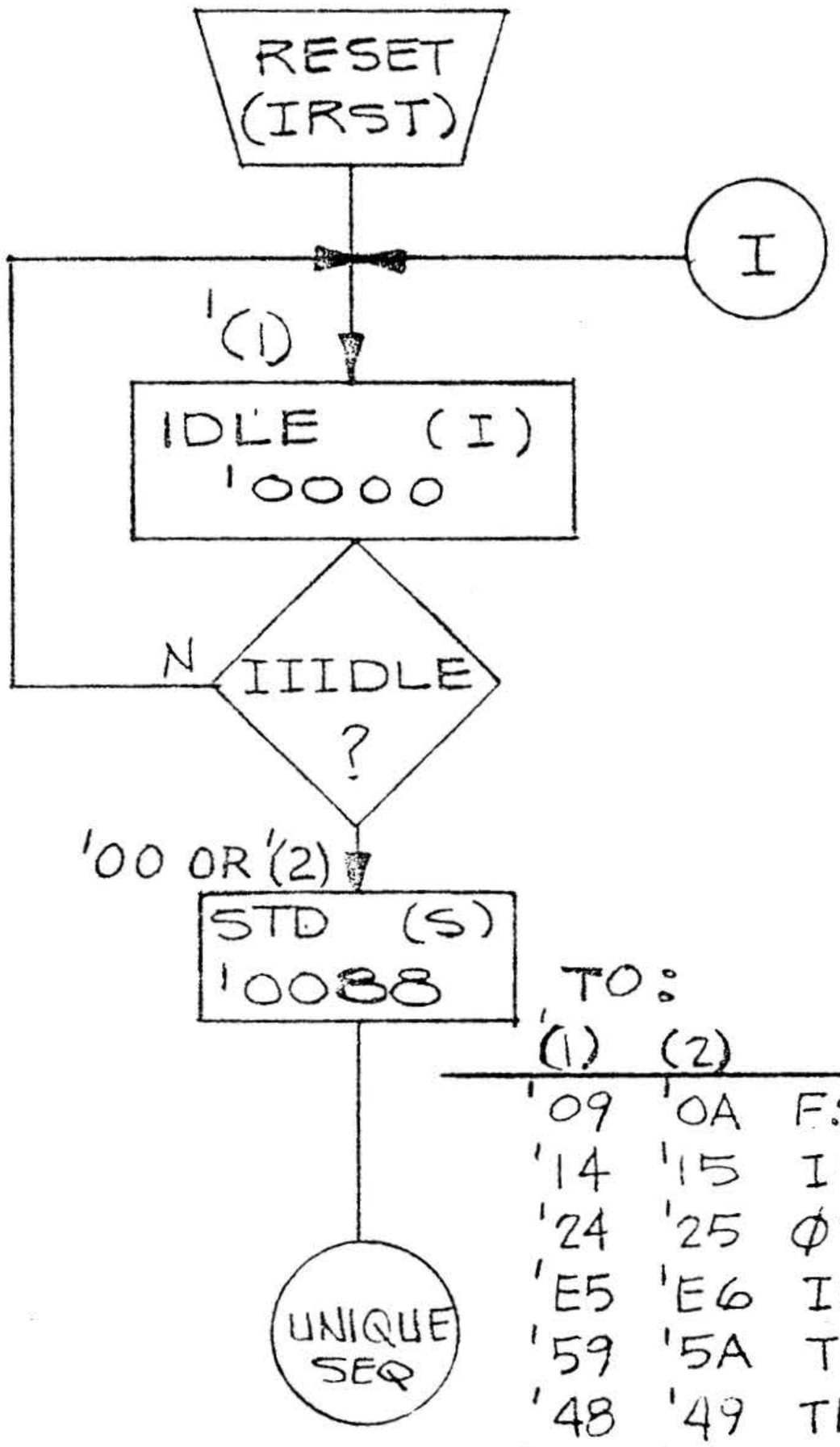
2.2.20 INPUT/OUTPUT (OPTION BOARD) MICRO FLOWS

2.2.20.1 IDLE STATE

NOTE: 1. 'XX INDICATES SOURCE ADDRESS OF I/O CONTROL BUFFER WORD

2. 'XXXX INDICATES I/O CONTROL BUFFER CONTENTS

3. ALL NUMBERS ARE IN HEX.

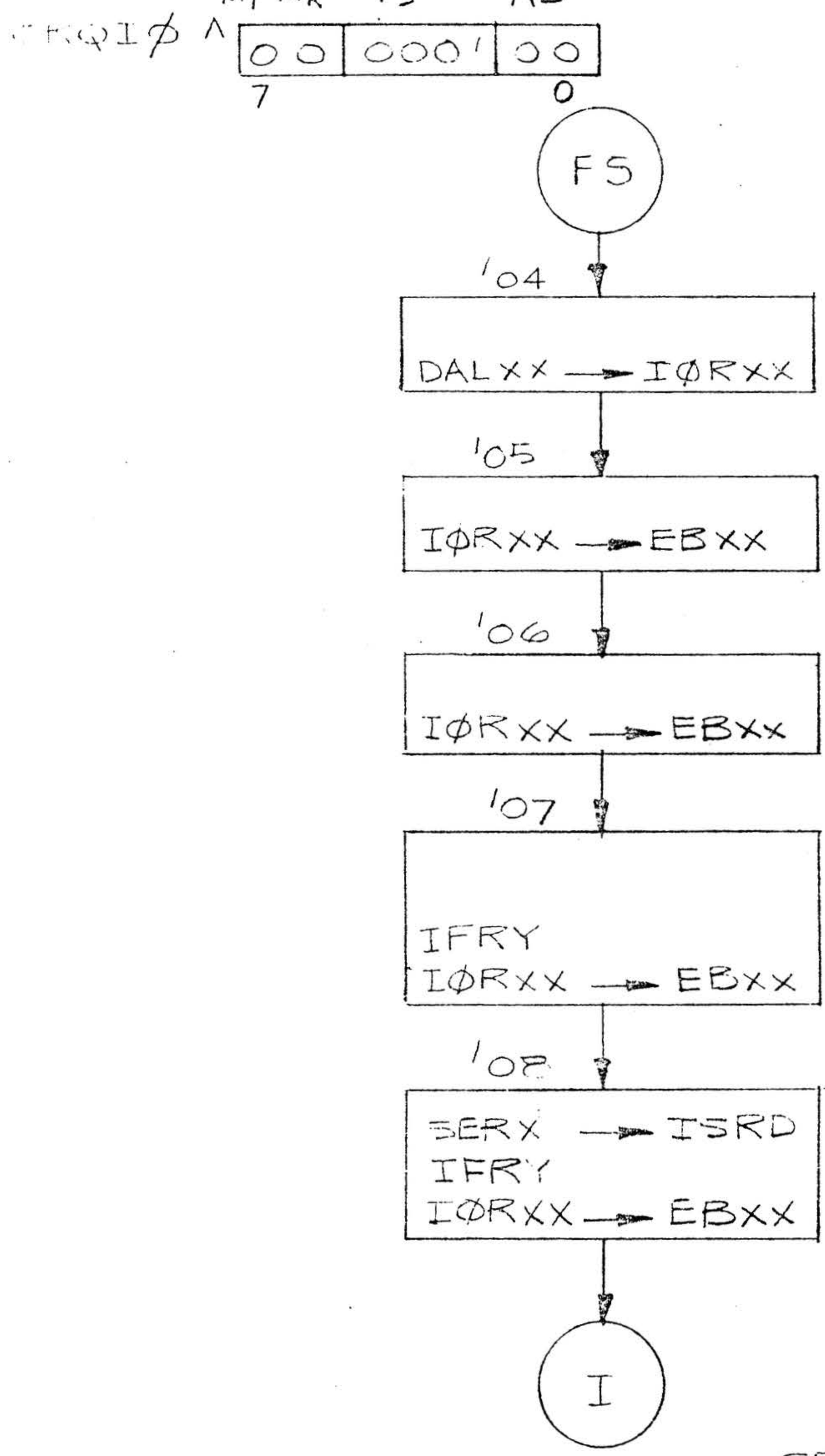


TO:

'(1)	'(2)	
'09	'0A	F.S. FUNCTION STEP
'14	'15	IS INPUT "
'24	'25	OS OUTPUT "
'E5	'E6	INT.S. INTERRUPT "
'59	'5A	TPIX DMA IN "
'48	'49	TPOX OUT "
'74	'75	TPIF HSDMA IN "
'64	'65	TPOF OUT "

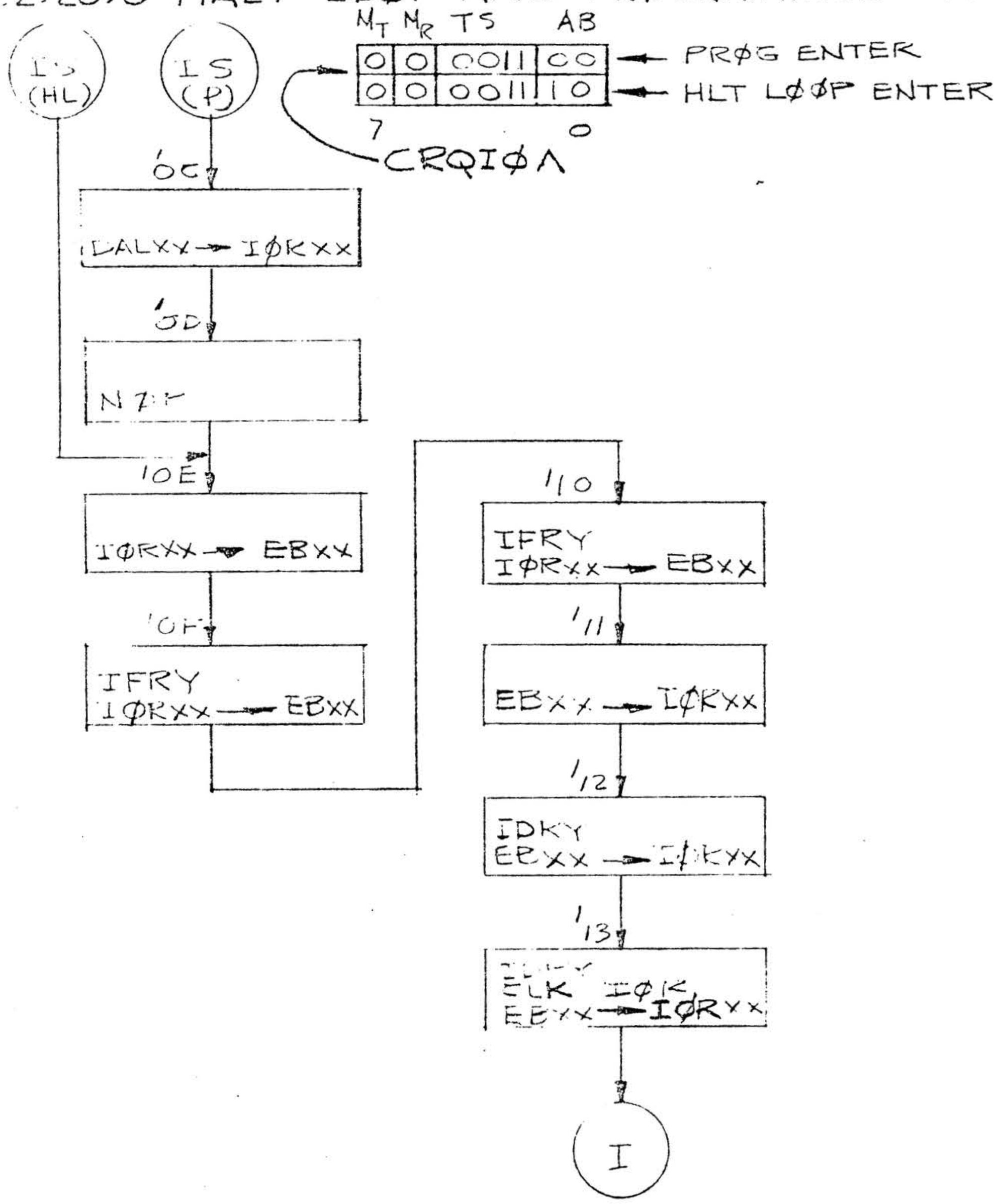
IDLE STATE

2.2.20.2 SEN, EXC, EXCA



SEN, EXC, EXCA

2.2.20.3 HALT LØØP AND PRØGRAMMED INPUT

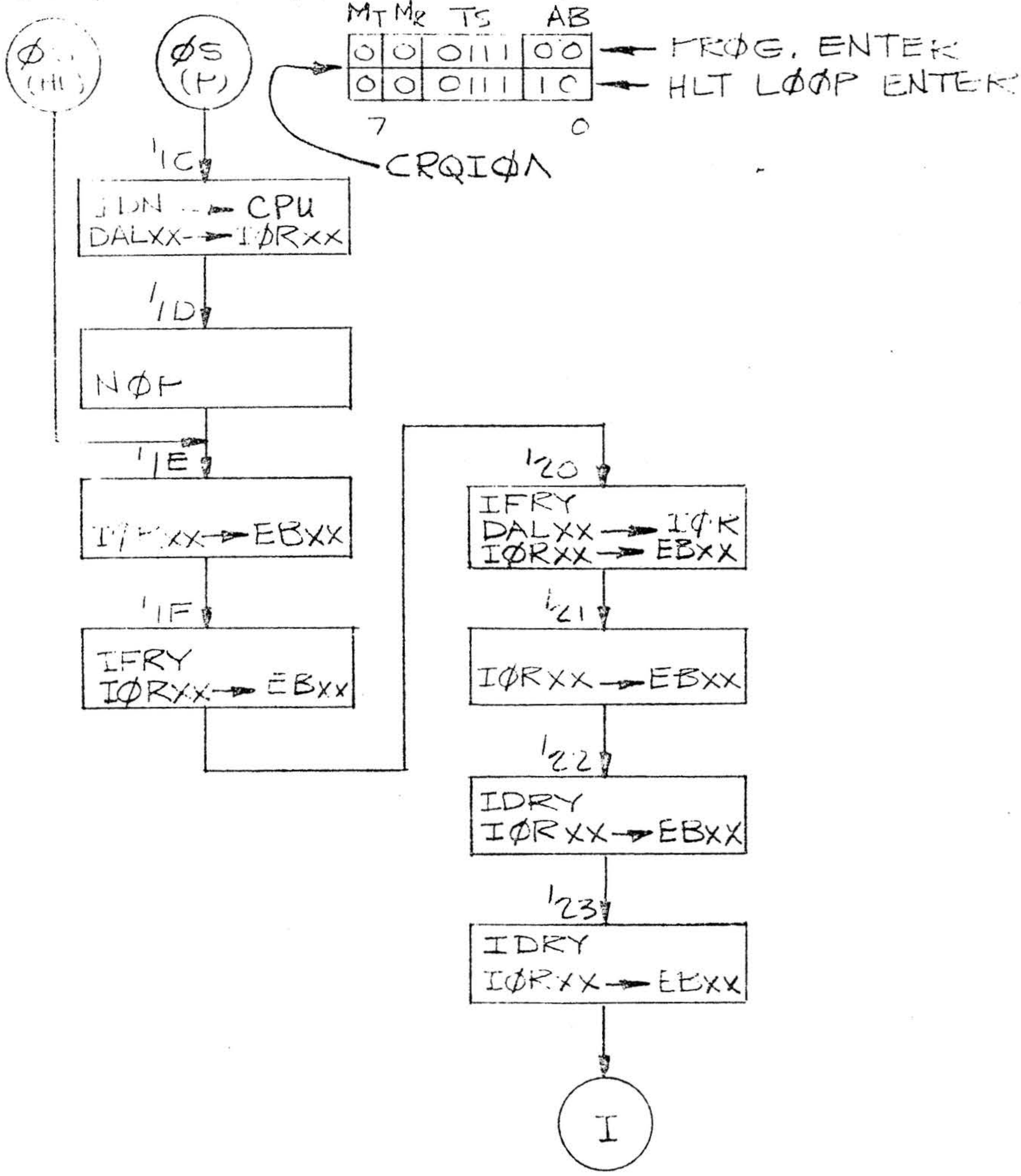


HALT LØØP AND
PRØGRAMMED INPUTS

varian data machines a varian subsidiary	CODE IDENT NO. 21101		98A0887 B
			SH 52 OF 57

96A0039-0008

2.2.20,4 HALT LØØP AND PRØGRAMMED ØUTPUT



HALT LØØP AND
PRØGRAMMED ØUTPUTS

varian data machines a varian subsidiary	CODE IDENT NO. 21101		96A0039-000	B
			SH 53 OF 57	REV

2.2.20.5
INTERRUPT

INT 5

'DC

EBXX → IØKXX

'DD

IFRY
EBXX → IØKXX

'DE

IFRY
CLK IØR
EBXX → IØRXX

IDN→CPU

'DF

WAIT FOR
CPU (CINTF)

N ← CINTF ? → 'E0

IUJX (IDRY)
IF IJMRK

CRQIØ ∧ IEMI ∧

MT	ME	TS	AB
1	1	0111	00
7			0

'E1

IUJK (IDRY)
IF IJMRK
RESET IUAX
REENAB. IUCX

'E2

NØP

'E3

NØP

'E4

NØP

I

INTERRUPT

2.2.20.6
DMA INPUT

ITRPN (TPIX-1)

0	1	0	1	0	0	0	0

7 0

HARDWARE GEN

(TPIX)

'50
EBXX → IØRXX

'51
IFRY
EBXX → IØRXX

'52
CLK IØR
IFRY
EBXX → IØRXX

'53
NØP

'54
NØP

'55
IRQM
EBXX → M.C.
WAIT FØR M.C. ACK

N MAKIØ?

'56
EBXX → M.C.

'57
IDRY
WAIT FØR M.C. DØNE
EBXX → M.C.

N MAKIØ?

'58
RESET IUAX
REENAB IUCX
IDRY
EBXX → M.C.

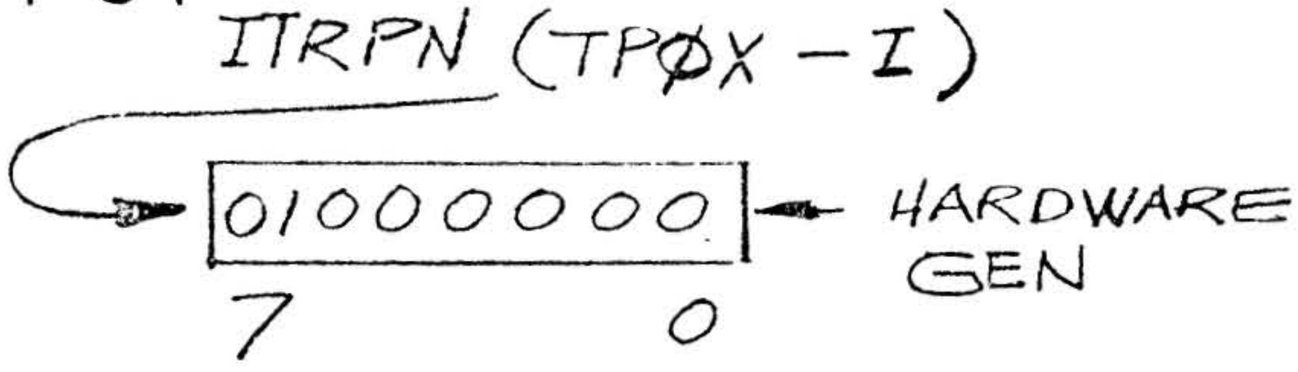
(I)

DMA INPUT (TPIX-I)

	CODE IDENT NO 21101		98A0[illegible] B
			SH 55 OF 57

2.2.20.7 DMA ØUTPUT

ITRPN (TPØX - I)



DMA ØUTPUT (TPØX- I)

varian data machines a varian subsidiary	CODE IDENT NO 21101		98A0887	B
			SH 56 OF 57	REV

2.2.20.8 HIGH SPEED DMA (TPØF, TPIF)

ITRPF (TPØF-I)^

01100000

7 0

ITRPF (TPIF-I)^

01110000

7 0

TPØF

'60

EBXX → IØRXX
CLK IØR
IFRY, IRQM
WAIT FØR M.C.
ACK

MAKIØ ? N

'61

NØP

'62

MIØLXX → IØRXX
CLK IØR
IØRXX → EBXX
IDRY, WAIT
FØR M.C. DØNE

MAKIØ ? N

'63

RESET IUAF
REENAB IUCF
IDRY
IØRXX → EBXX

I

TPIF

'70

EBXX → IØRXX
CLK IØR
IFRY, IRQM
WAIT FOR M.C.
ACK

MAKIØ ? N

'71

EBXX → M.C.

'72

EBXX → M.C.
IDRY
WAIT FØR M.C.
DØNE

MAKIØ ? N

'73

RESET IUAF
REENAB IUCF

I

HS DMA (TPØF, TPIF)