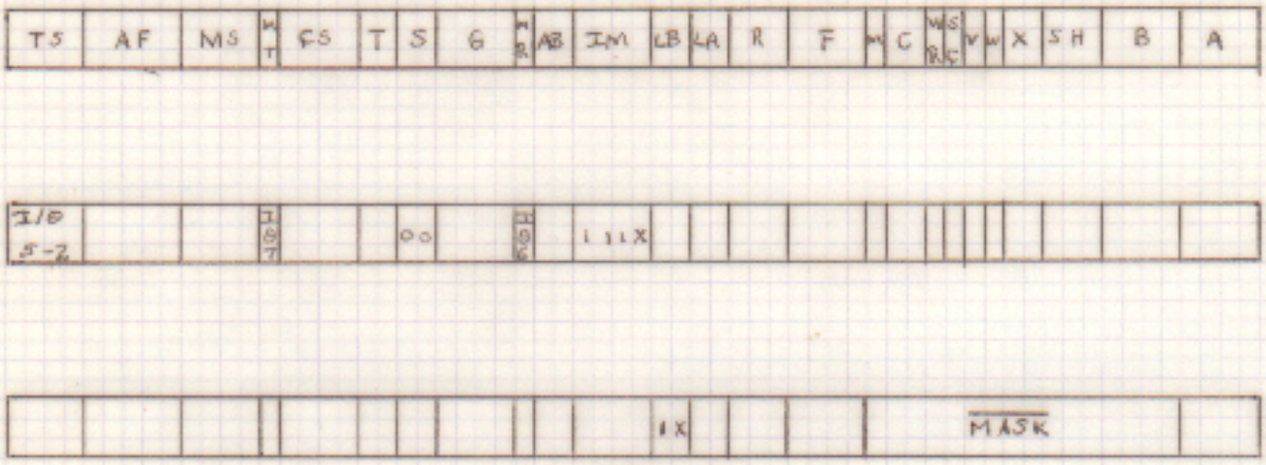
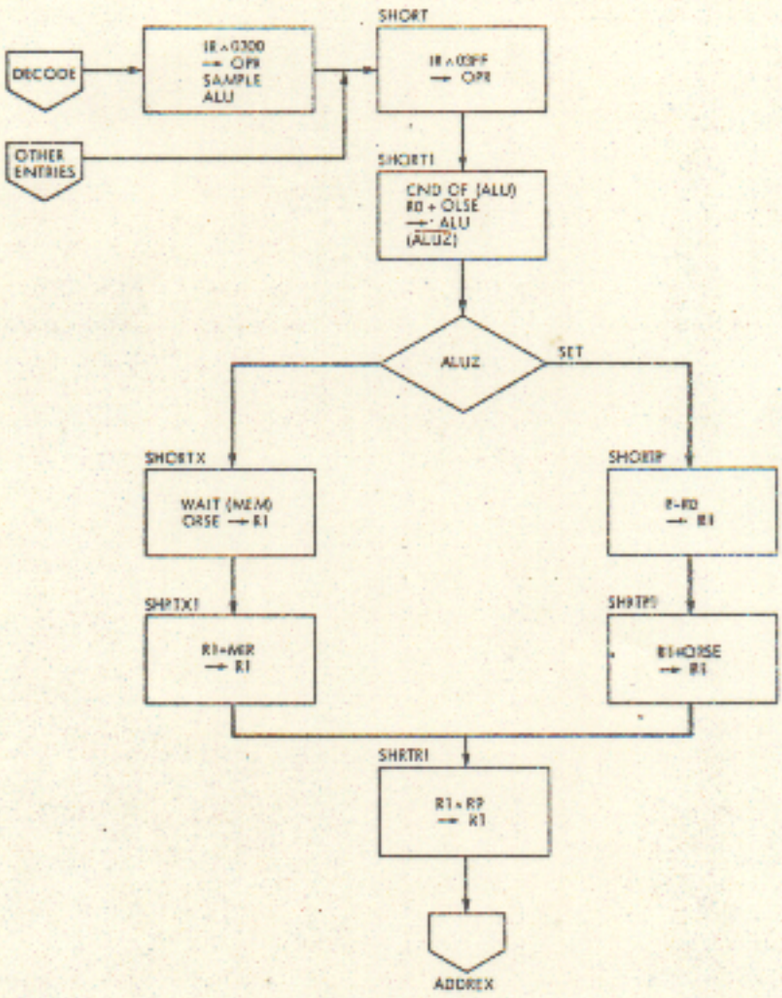
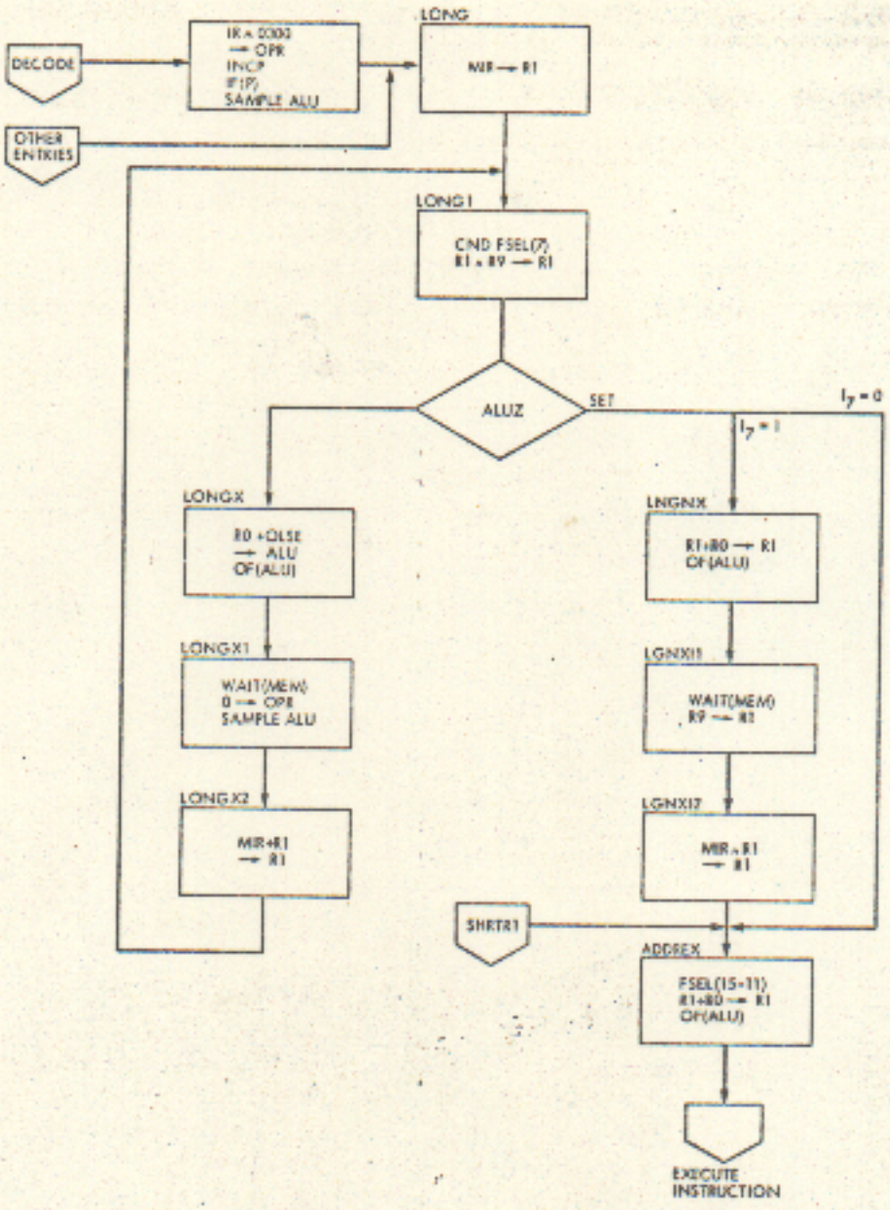


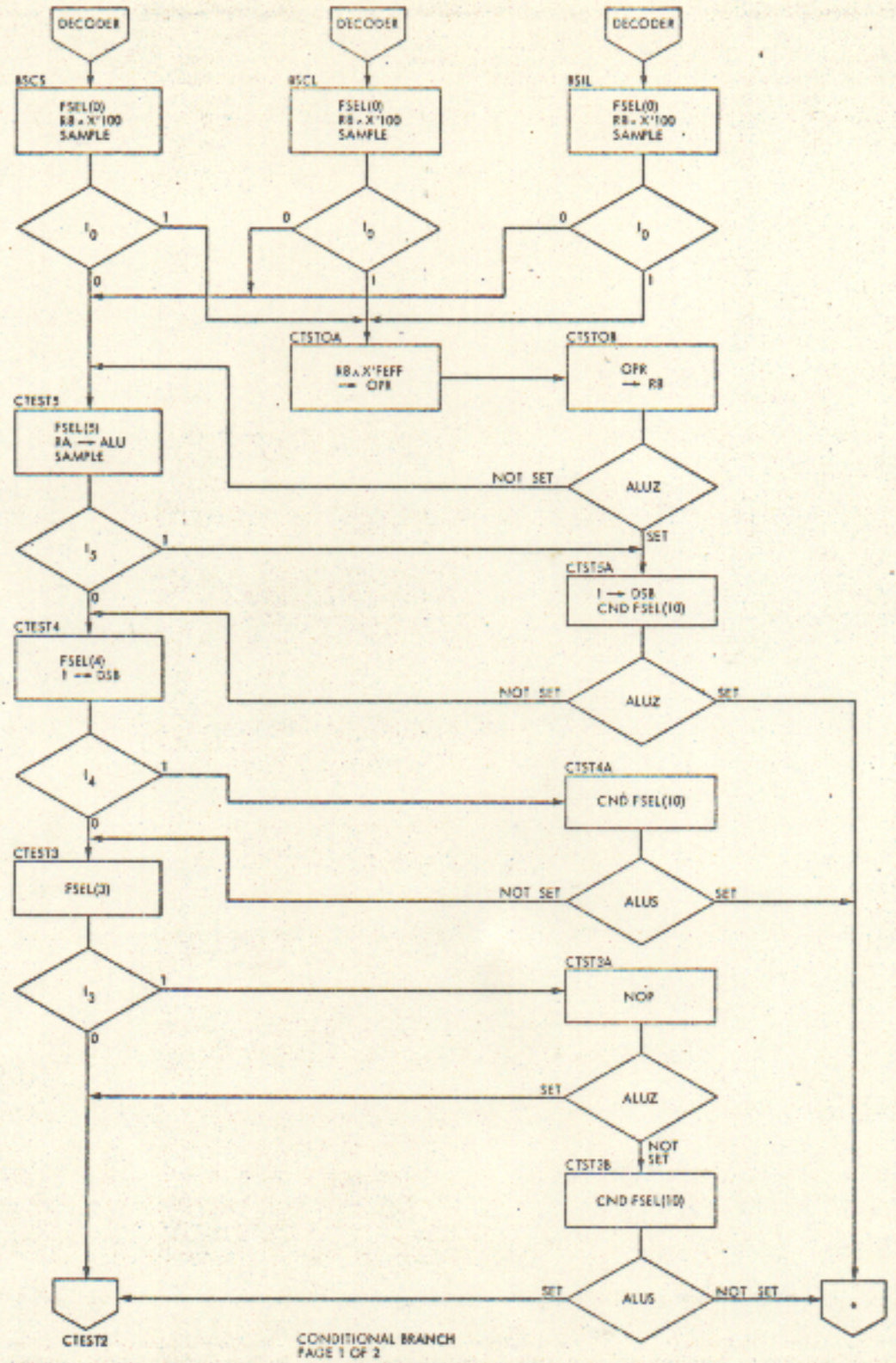
CENTRAL CONTROL STORE FORMATS



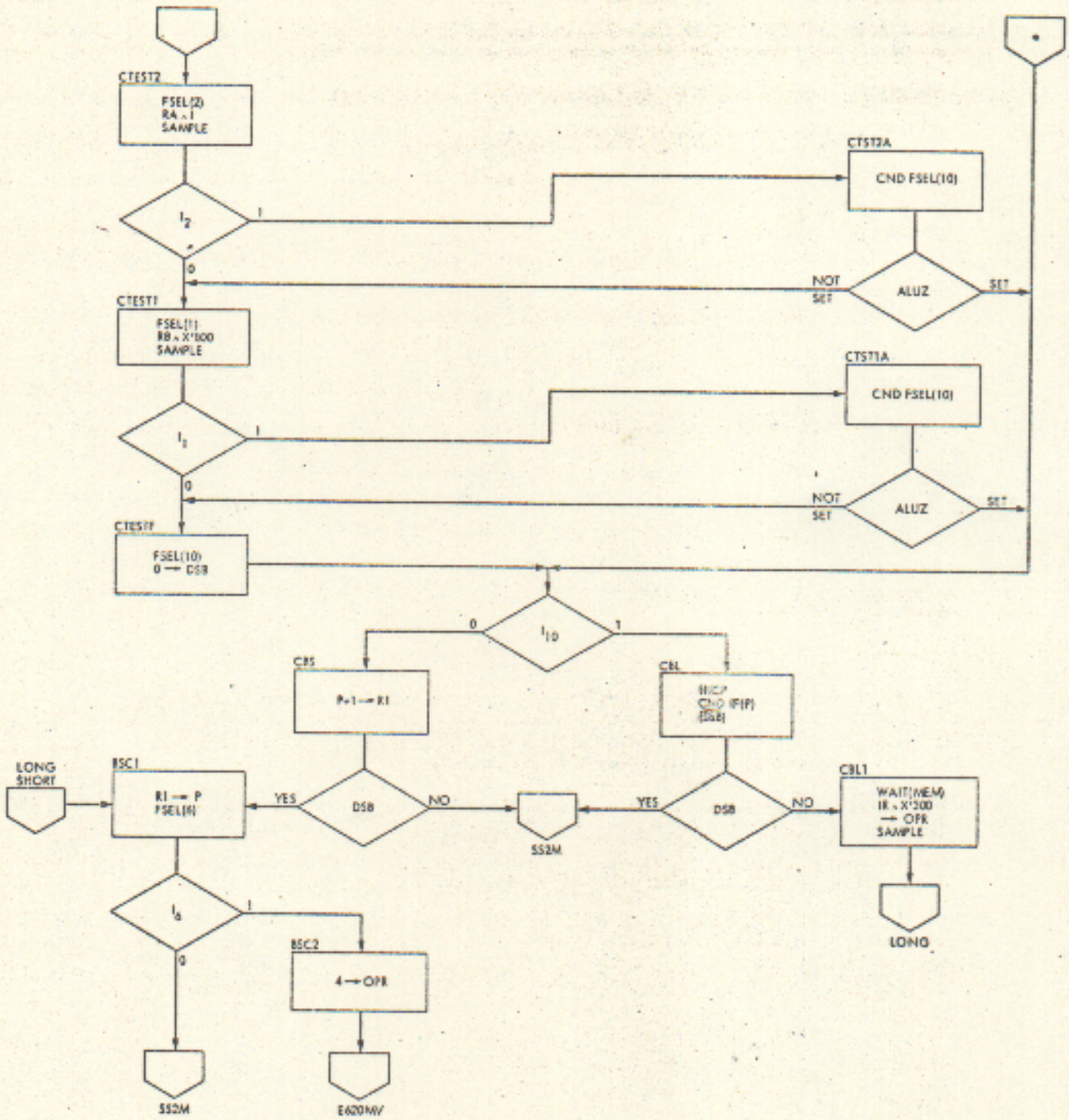
OPERAND FETCH
SHORT FORMAT INSTRUCTION



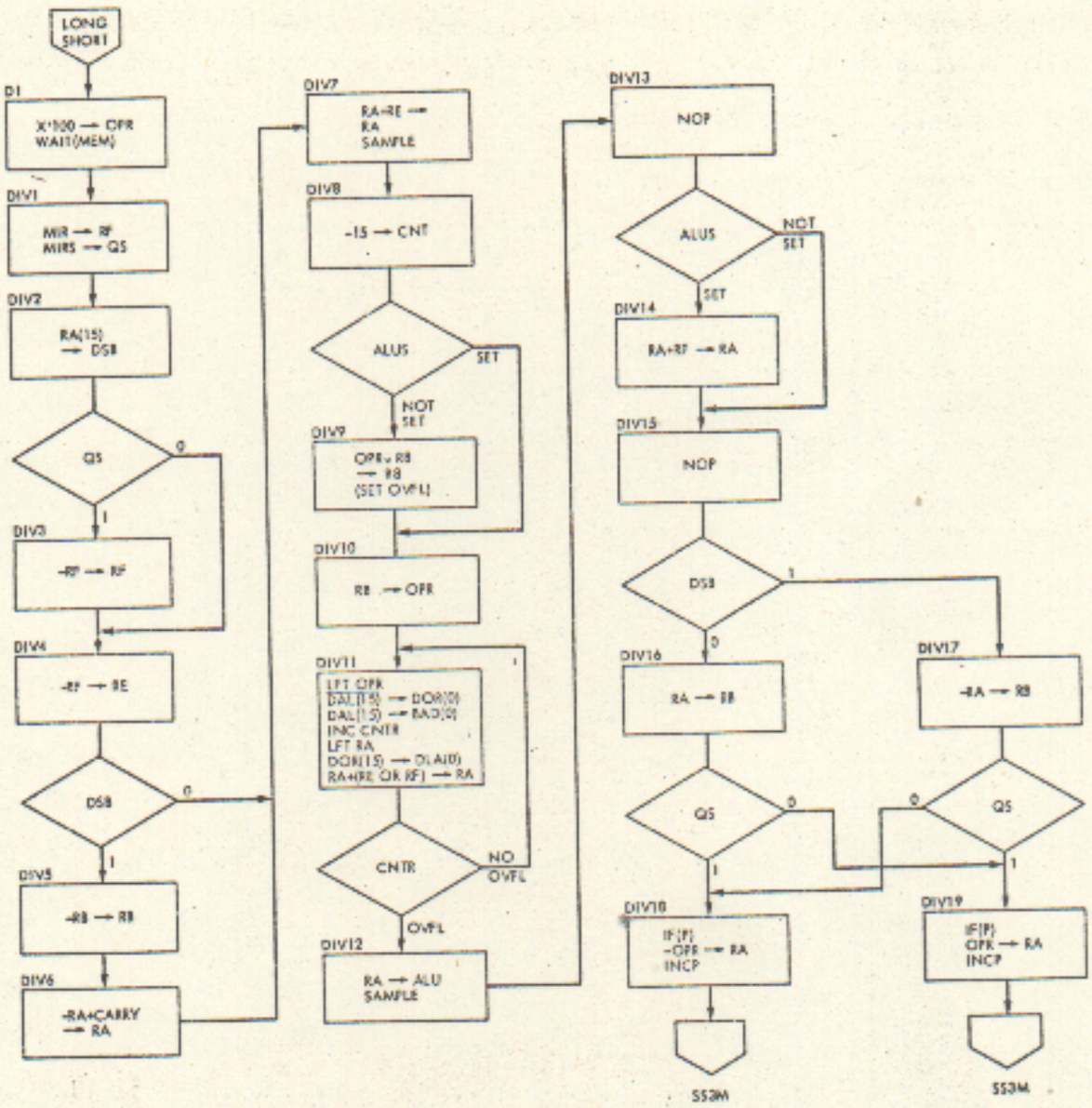
OPERAND FETCH
LONG FORMAT INSTRUCTION



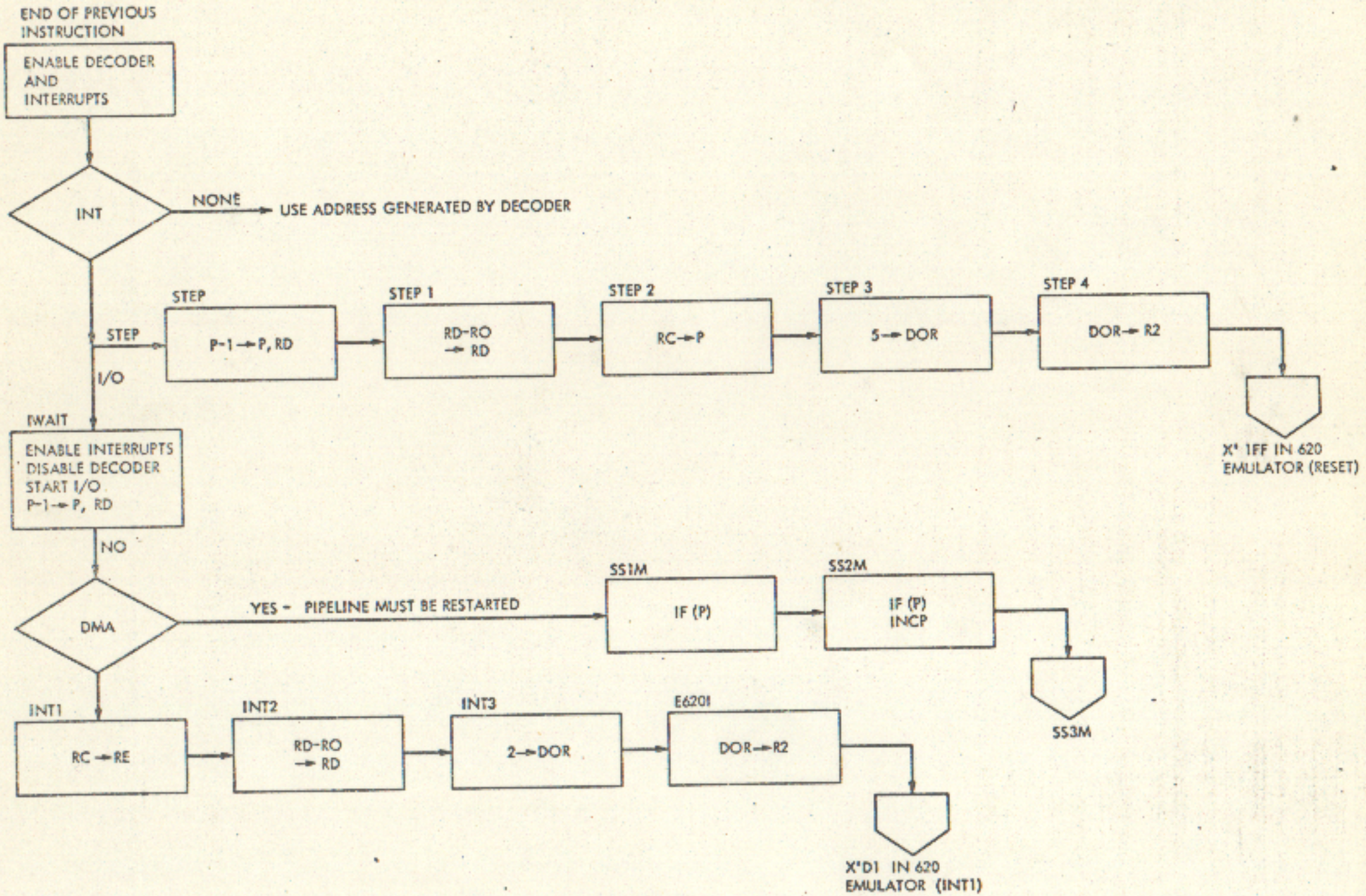
CONDITIONAL BRANCH
PAGE 1 OF 2

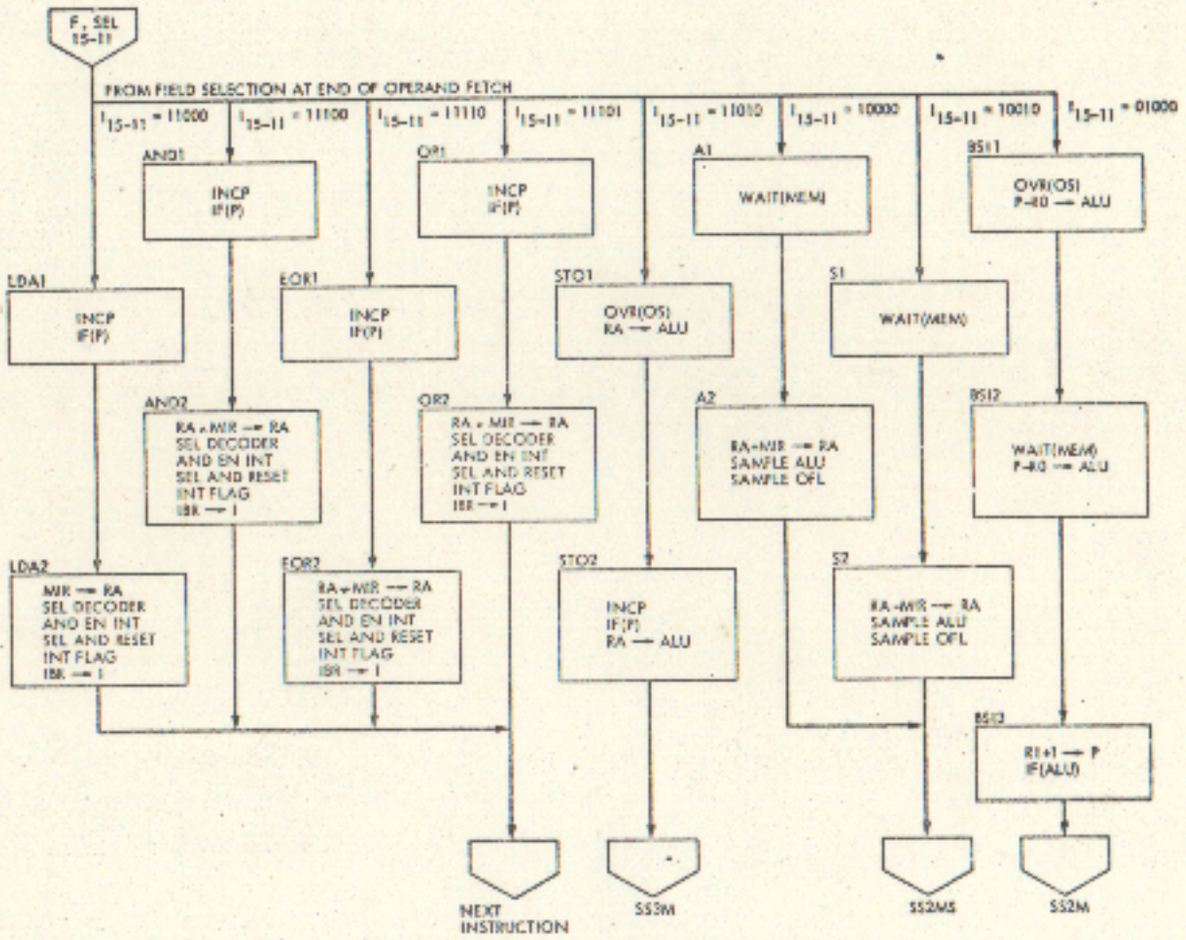


CONDITIONAL BRANCH
PAGE 2 OF 2

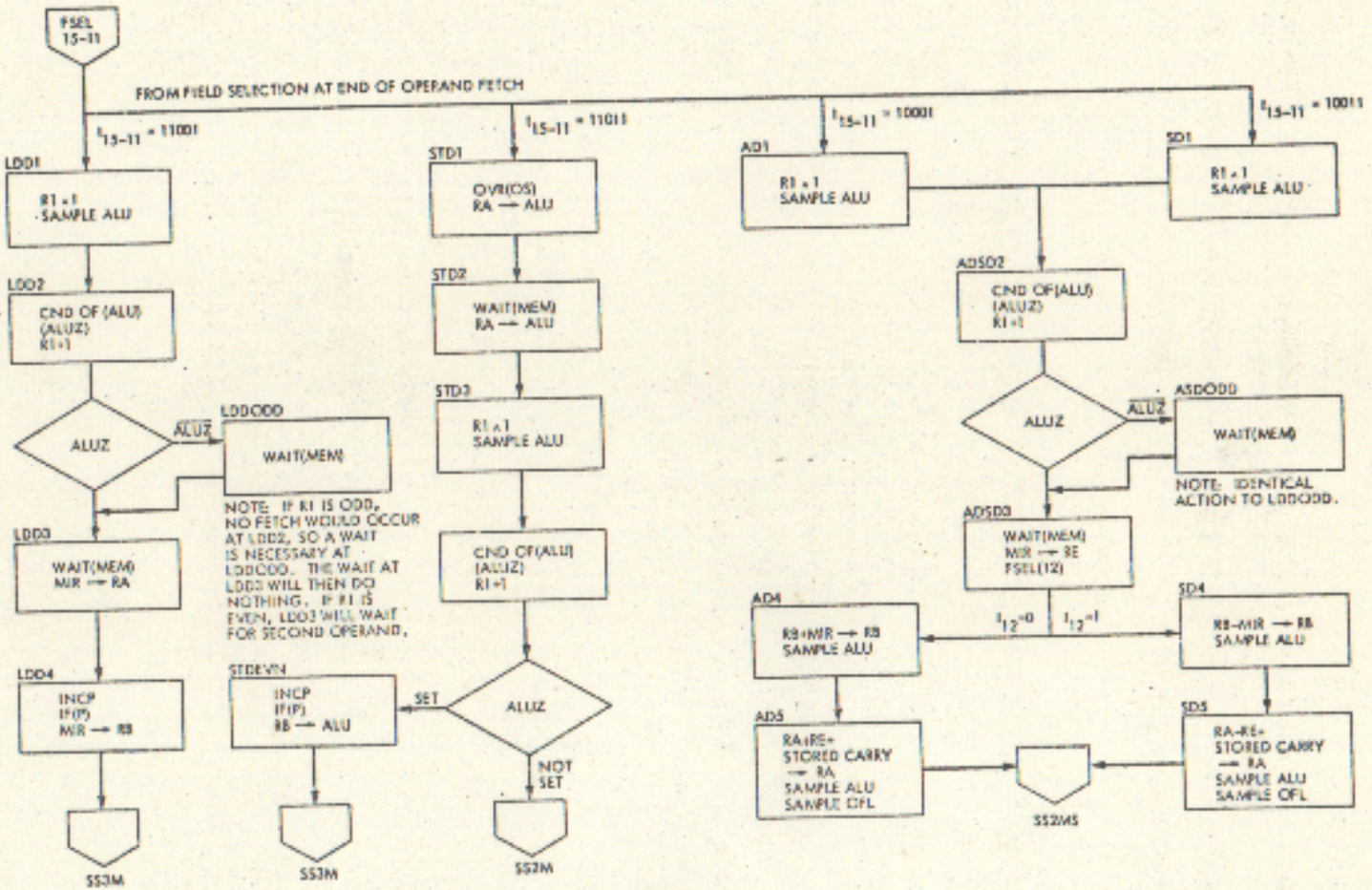

LONG
SHORT
D1
X'100 → OPR
WAIT(MEM)
DIV1
MIR → RF
MIRS → QS
DIV2
RA(15)
→ DSB
QS
0
1
DIV3
-RF → RF
DIV4
-RF → RE
DSB
0
1
DIV5
-RB → RB
DIV6
-RA+CARRY
→ RA
DIV7
RA=RE →
RA
SAMPLE
DIV8
-15 → CNT
ALUS
SET
NOT
SET
DIV9
OPR+RB
→ RB
(SET OVFL)
DIV10
RB → OPR
DIV11
LFT OPR
DAL(15) → DOR(0)
DAL(15) → BAD(0)
INC CNTR
LFT RA
DOR(15) → DLAI(0)
RA=(RE OR RF) → RA
CNTR
NO
OVFL
OVFL
DIV12
RA → ALU
SAMPLE
DIV13
NOP
ALUS
NOT
SET
SET
DIV14
RA+RF → RA
DIV15
NOP
DSB
1
0
DIV16
RA → RB
QS
0
1
DIV18
IF(F)
-OPR → RA
INCP
SS3M
DIV17
-RA → RB
QS
0
1
DIV19
IF(F)
OPR → RA
INCP
SS3M

DIVIDE

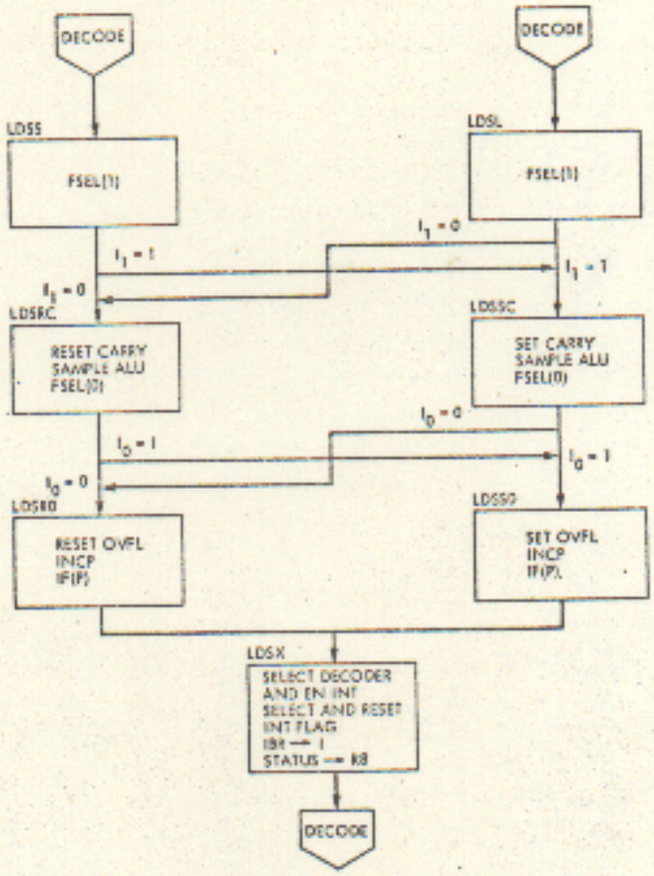

END OF PREVIOUS INSTRUCTION
ENABLE DECODER AND INTERRUPTS
INT
NONE
USE ADDRESS GENERATED BY DECODER
STEP
STEP
P-1→P, RD
STEP 1
RD-RO → RD
STEP 2
RC→P
STEP 3
5→DOR
STEP 4
DOR→R2
X' 1FF IN 620 EMULATOR (RESET)
I/O
WAIT
ENABLE INTERRUPTS
DISABLE DECODER
START I/O
P-1→P, RD
NO
DMA
YES - PIPELINE MUST BE RESTARTED
SS1M
IF (P)
SS2M
IF (P)
INCP
SS3M
INT1
RC→RE
INT2
RD-RO → RD
INT3
2→DOR
E620I
DOR→R2
X'D1 IN 620 EMULATOR (INT1)



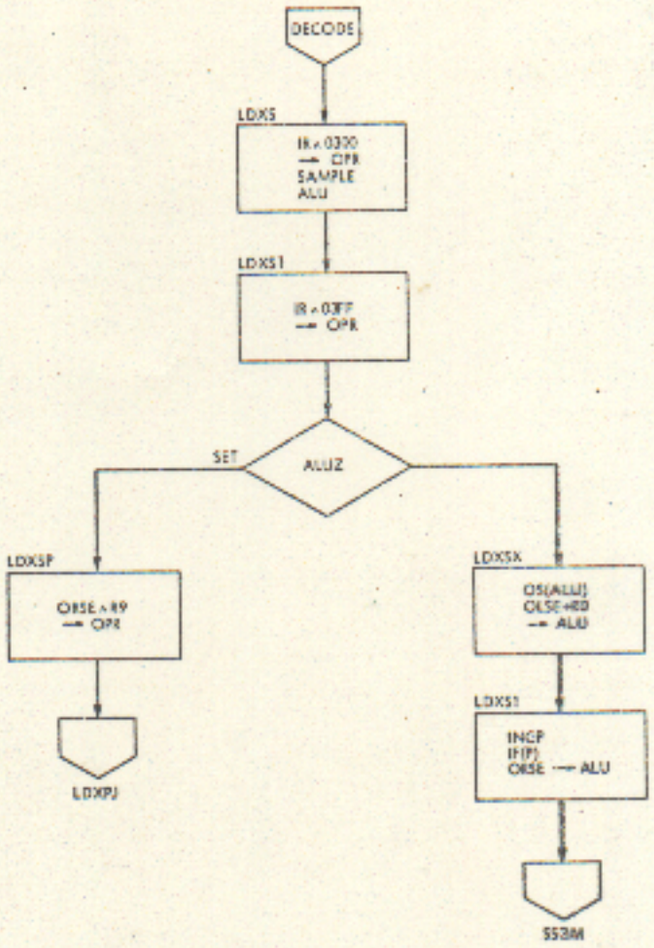
LDA, STO, A, S, AND, EOR, OR, BSI


FSEL 15-11
FROM FIELD SELECTION AT END OF OPERAND FETCH
I15-11 = 11001
I15-11 = 11011
I15-11 = 10001
I15-11 = 10011
LDD1
R1 ← 1
SAMPLE ALU
LDD2
CND OF (ALU)
(ALUZ)
R1=1
ALUZ
ALUZ
LDDODD
WAIT(MEM)
NOTE: IF R1 IS ODD, NO FETCH WOULD OCCUR AT LDD2, SO A WAIT IS NECESSARY AT LDDODD. THE WAIT AT LDD3 WILL THEN DO NOTHING. IF R1 IS EVEN, LDD3 WILL WAIT FOR SECOND OPERAND.
LDD3
WAIT(MEM)
MIR → RA
LDD4
INCP
IF(P)
MIR → RB
SS3M
STDEVN
INCP
IF(P)
RB → ALU
SS3M
STD1
OVR(OS)
RA → ALU
STD2
WAIT(MEM)
RA → ALU
STD3
R1 ← 1
SAMPLE ALU
CND OF(ALU)
(ALUZ)
R1=1
ALUZ
SET
NOT SET
SS2M
AD1
R1 ← 1
SAMPLE ALU
SD1
R1 ← 1
SAMPLE ALU
ADSD2
CND OF(ALU)
(ALUZ)
R1=1
ALUZ
ALUZ
ASDODD
WAIT(MEM)
NOTE: IDENTICAL ACTION TO LDDODD.
ADSD3
WAIT(MEM)
MIR → RE
FSEL(12)
I12=0
I12=1
AD4
RB+MIR → RB
SAMPLE ALU
SD4
RB-MIR → RB
SAMPLE ALU
AD5
RA±RE±
STORED CARRY
→ RA
SAMPLE ALU
SAMPLE OFL
SD5
RA-RE±
STORED CARRY
→ RA
SAMPLE ALU
SAMPLE OFL
SS2MS

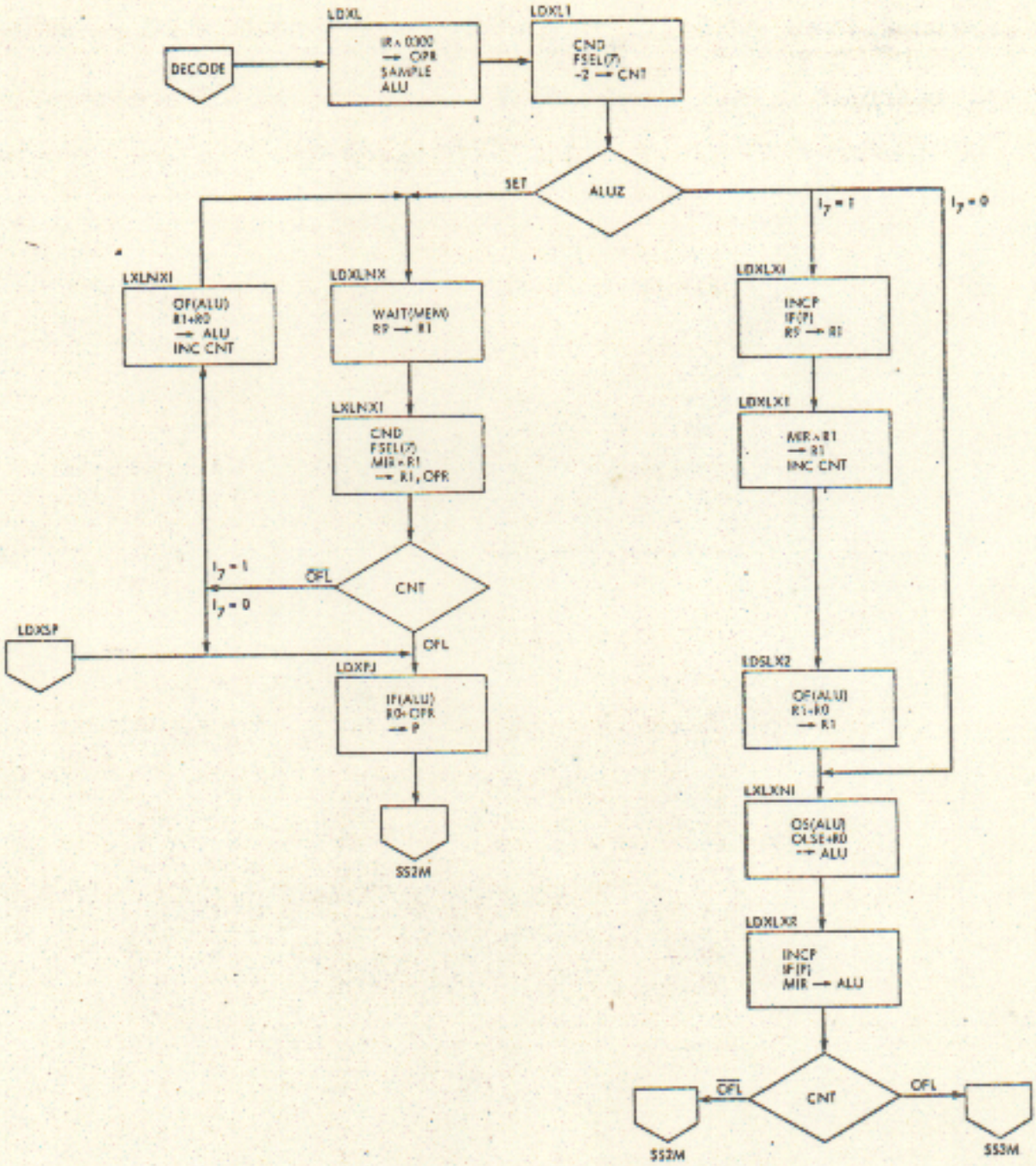
LDD STD
AD SD



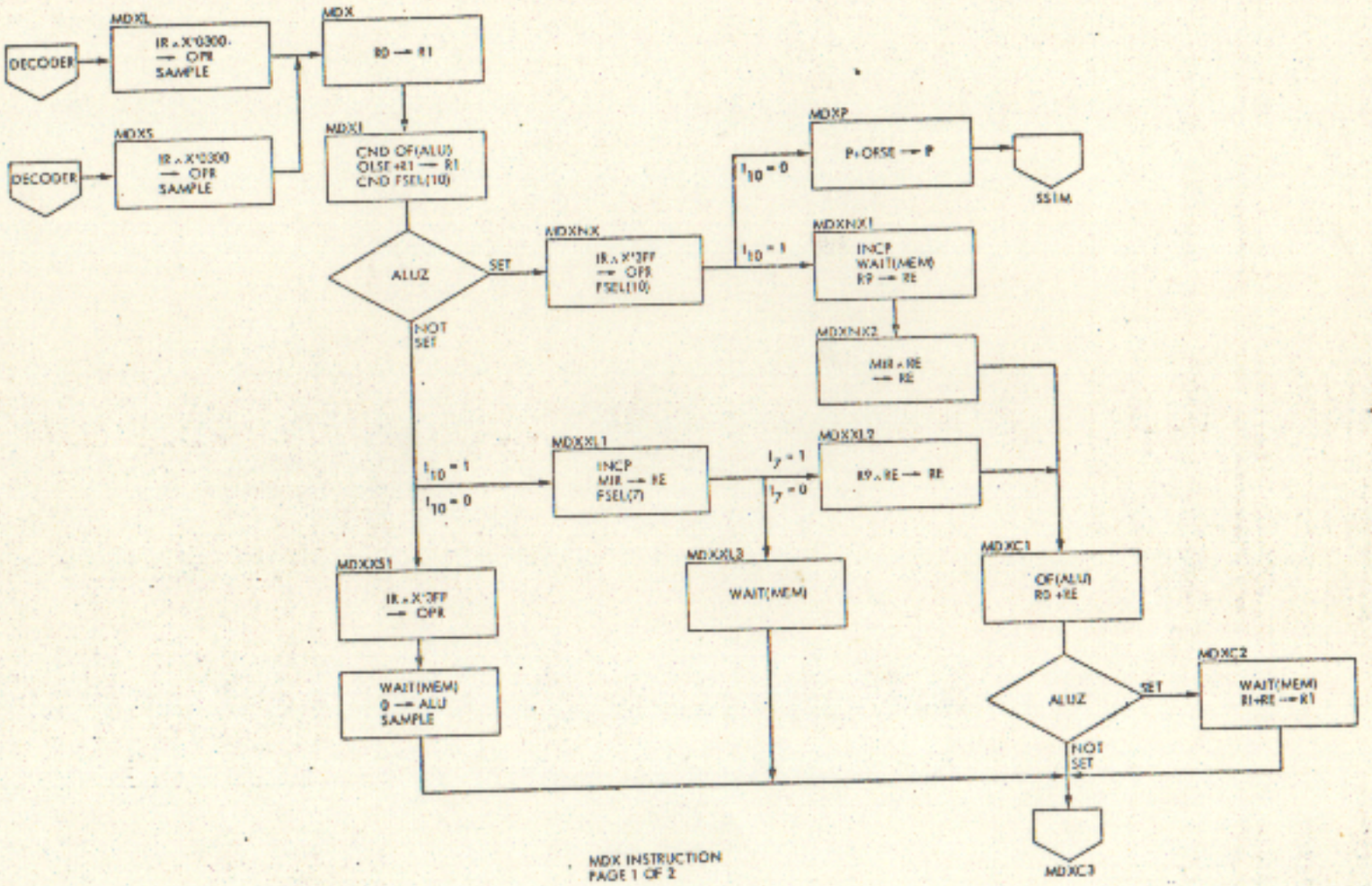
LDS
BOTH FORMATS



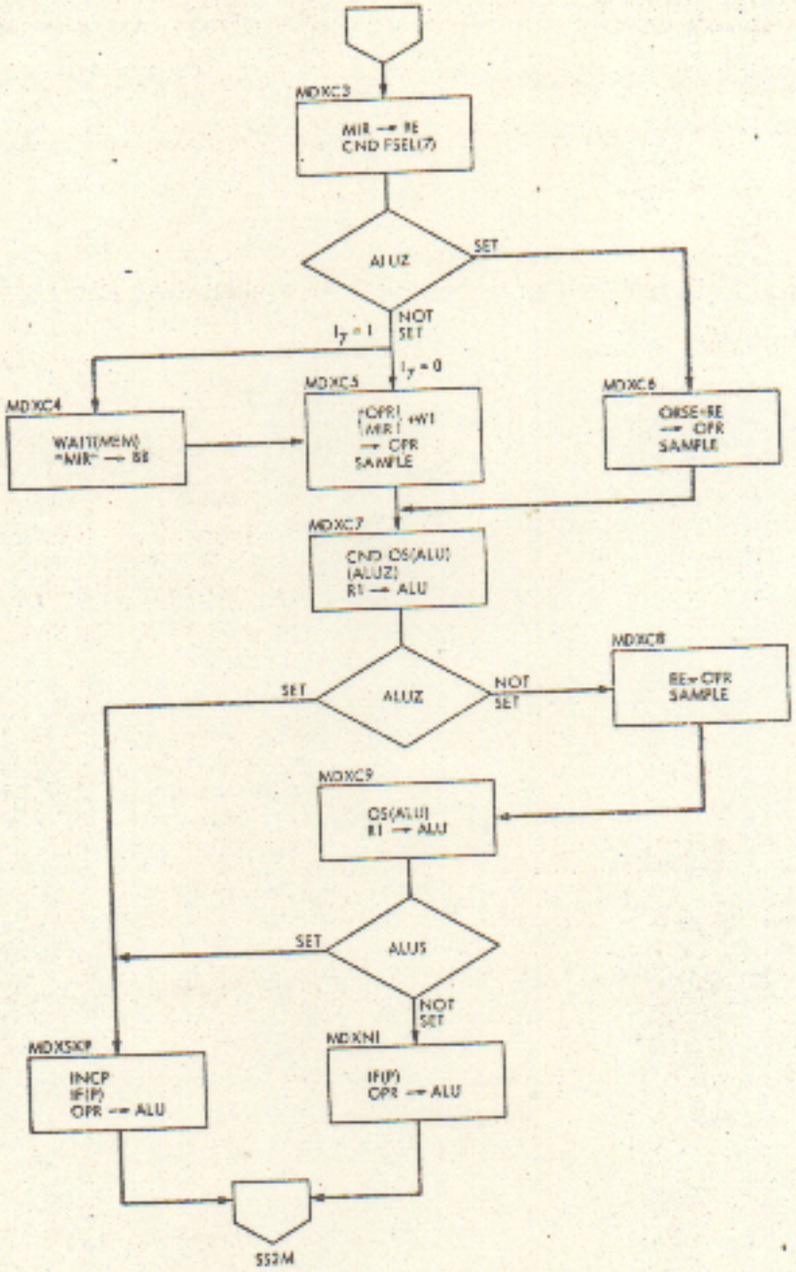
LDX
SHORT FORMAT



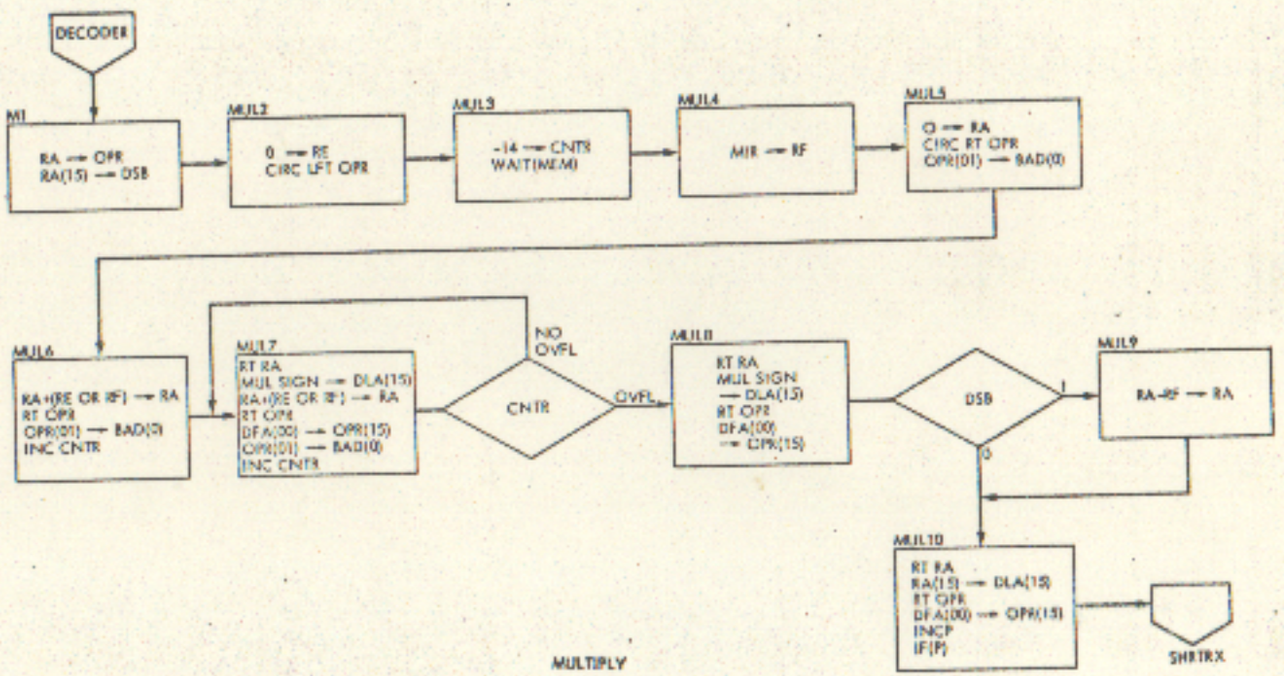
LDX
LONG FORMAT



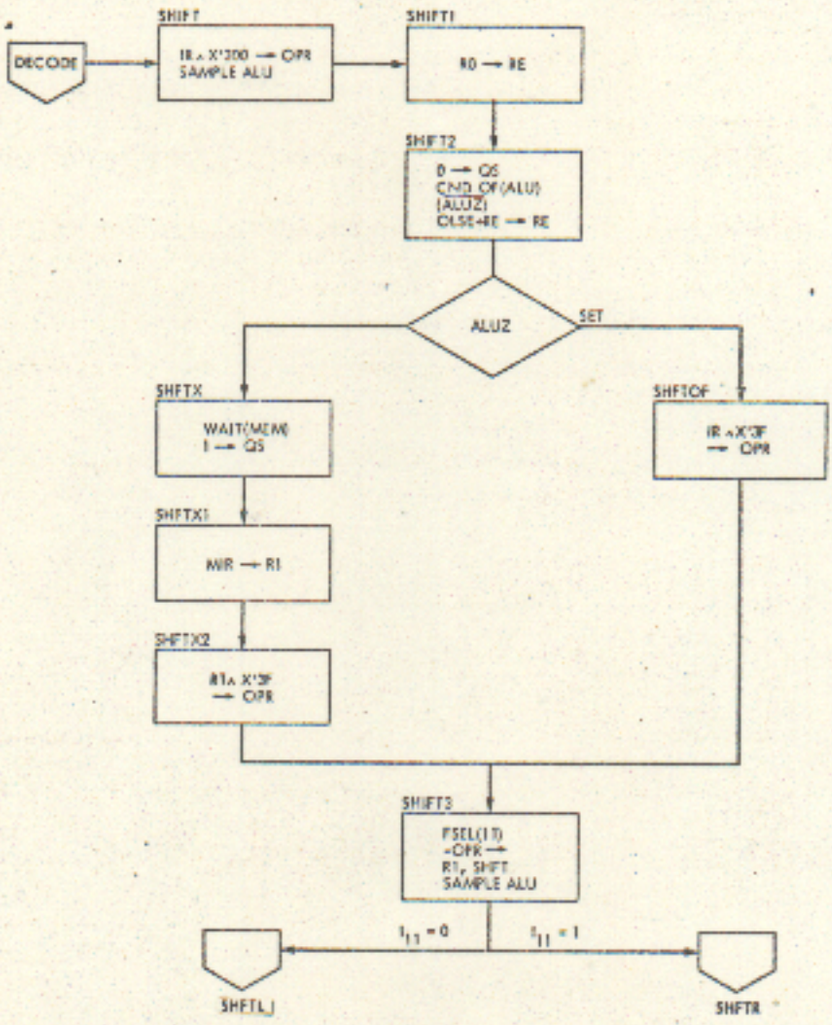
MDX INSTRUCTION
PAGE 1 OF 2



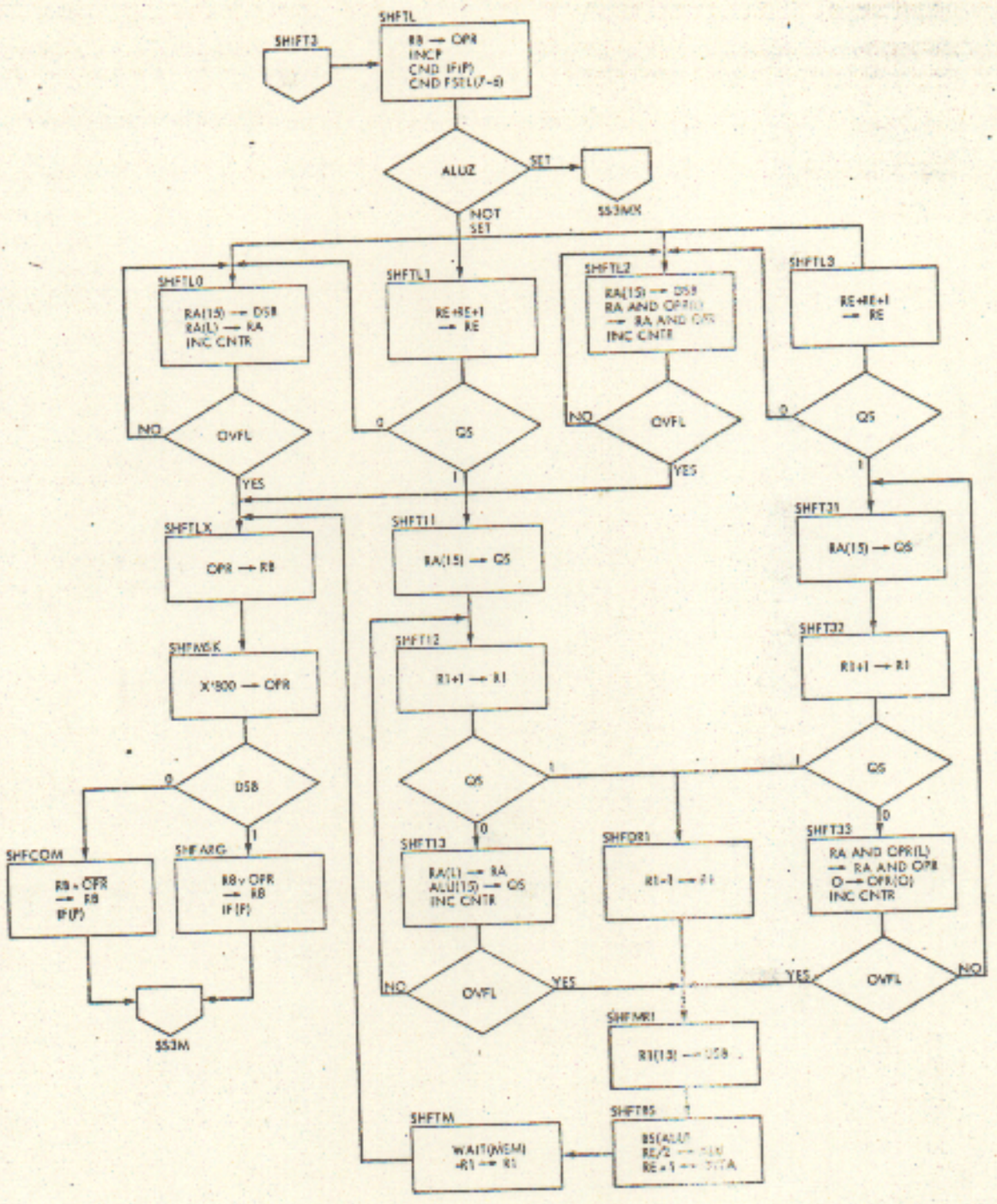
MDX INSTRUCTION
PAGE 2 OF 2


DECODER
MI
RA → OPR
RA(15) → DSB
MUL2
0 → RE
CIRC LFT OPR
MUL3
-14 → CNTR
WAIT(MEM)
MUL4
MIR → RF
MUL5
0 → RA
CIRC RT OPR
OPR(01) → BAD(0)
MUL6
RA+(RE OR RF) → RA
RT OPR
OPR(01) → BAD(0)
INC CNTR
MUL7
RT RA
MUL SIGN → DLA(15)
RA+(RE OR RF) → RA
RT OPR
DFA(00) → OPR(15)
OPR(01) → BAD(0)
INC CNTR
NO OVFL
CNTR
OVFL
MUL8
RT RA
MUL SIGN
→ DLA(15)
RT OPR
DFA(00)
→ OPR(15)
DSB
1
MUL9
RA-RF → RA
0
MUL10
RT RA
RA(15) → DLA(15)
RT OPR
DFA(00) → OPR(15)
INCP
IF(P)
SHRTRX

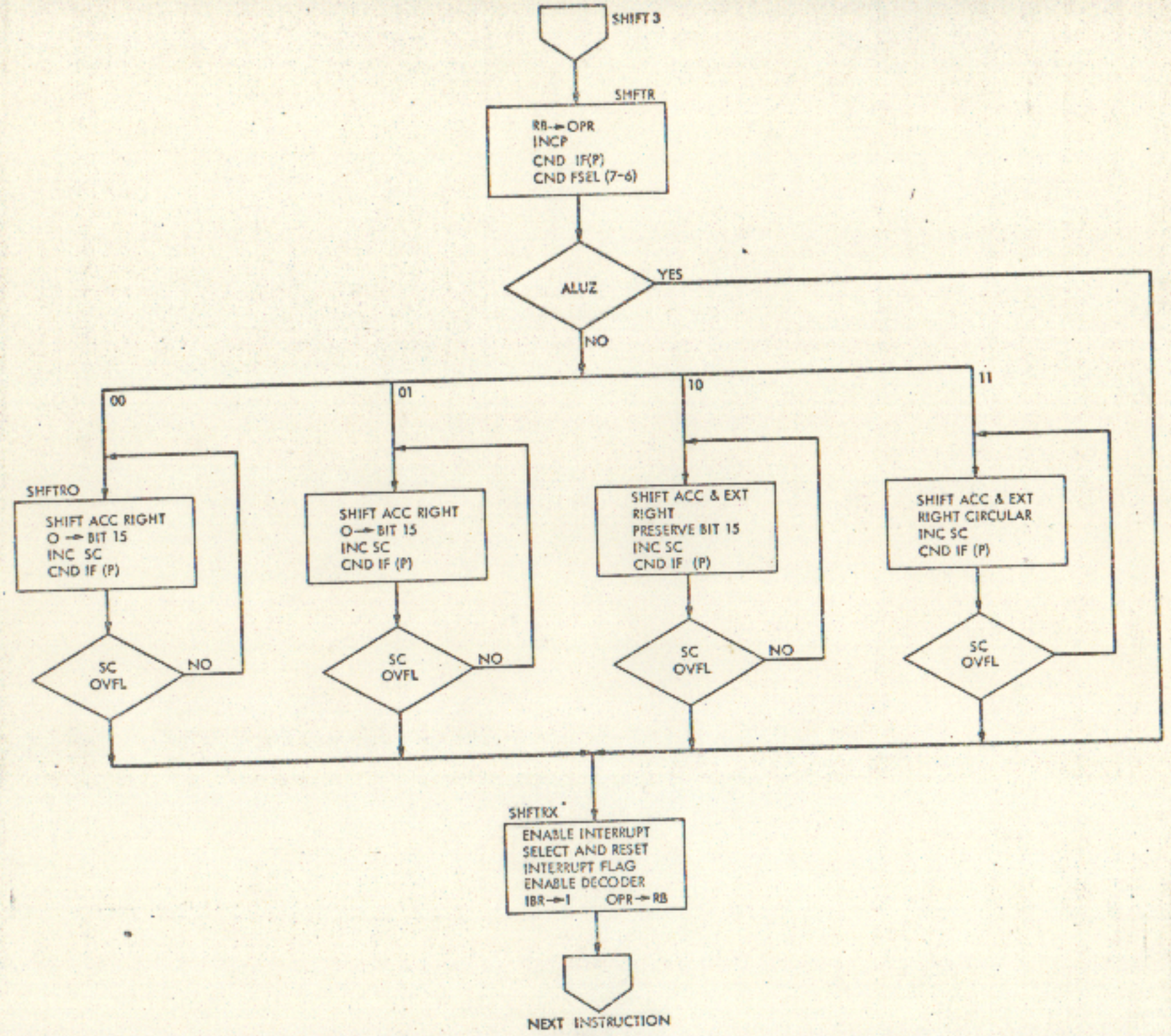
MULTIPLY



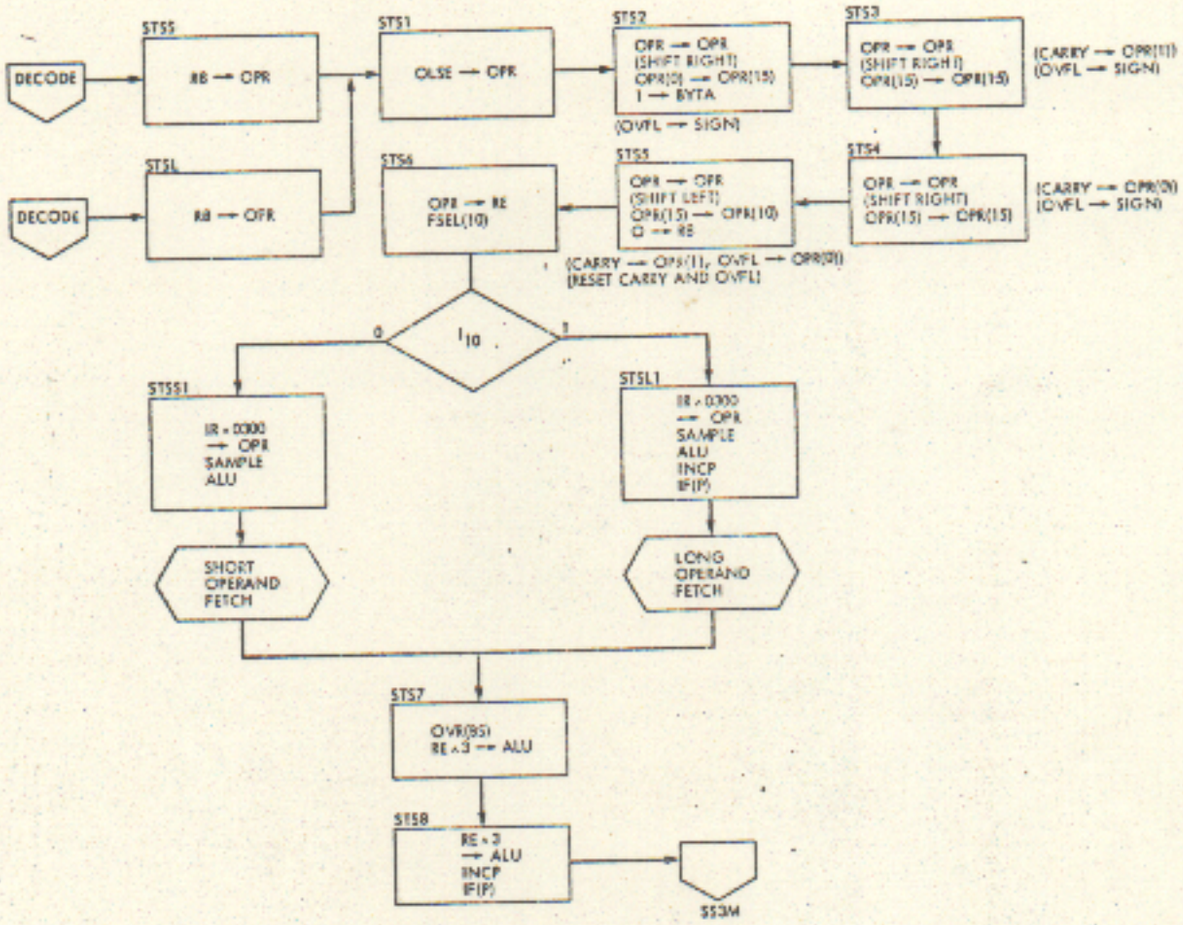
SHIFT COUNT
COMPUTATION


SHIFT3
SHFTL
RB → OPR
INCP
CND IF(P)
CND FSTL(7-0)
ALUZ
SET
SS3MX
NOT SET
SHFTL0
RA(15) → DSB
RA(L) → RA
INC CNTR
OVFL
NO
YES
SHFTL1
RE+RE+1 → RE
QS
SHFTL2
RA(15) → DSB
RA AND OPR(L) → RA AND QS
INC CNTR
OVFL
SHFTL3
RE+RE+1 → RE
QS
SHFTLX
OPR → RB
SHFMSK
X'800' → OPR
DSB
SHFCOM
RB ÷ OPR → RB IF(P)
SHFARG
RB × OPR → RB IF(P)
SS3M
SHFTL1
RA(15) → QS
SHFTL2
R1+1 → R1
QS
SHFTL3
RA(L) → RA
ALU(15) → QS
INC CNTR
OVFL
SHFQR1
R1-1 → R1
SHFMR1
R1(13) → DSB
SHFTM
WAIT(MEM)
-R1 → R1
SHFTL5
SHFTL1
RA(15) → QS
SHFTL2
R1+1 → R1
QS
SHFTL3
RA AND OPR(L) → RA AND OPR
0 → OPR(0)
INC CNTR
OVFL
YES
NO

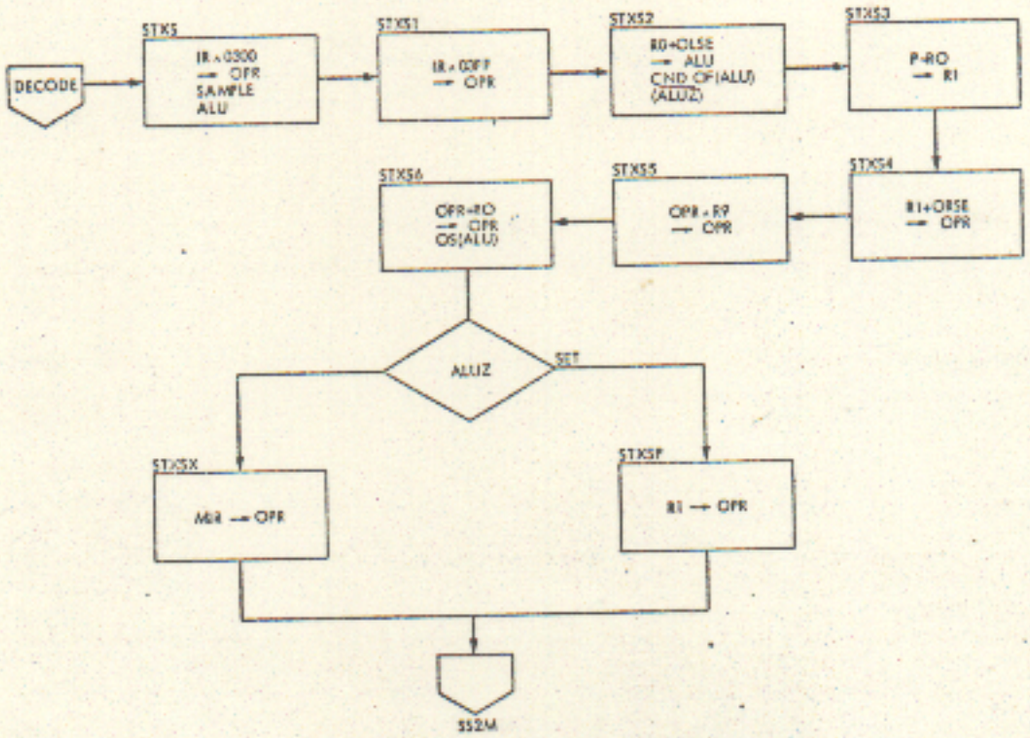
LEFT SHIFTS



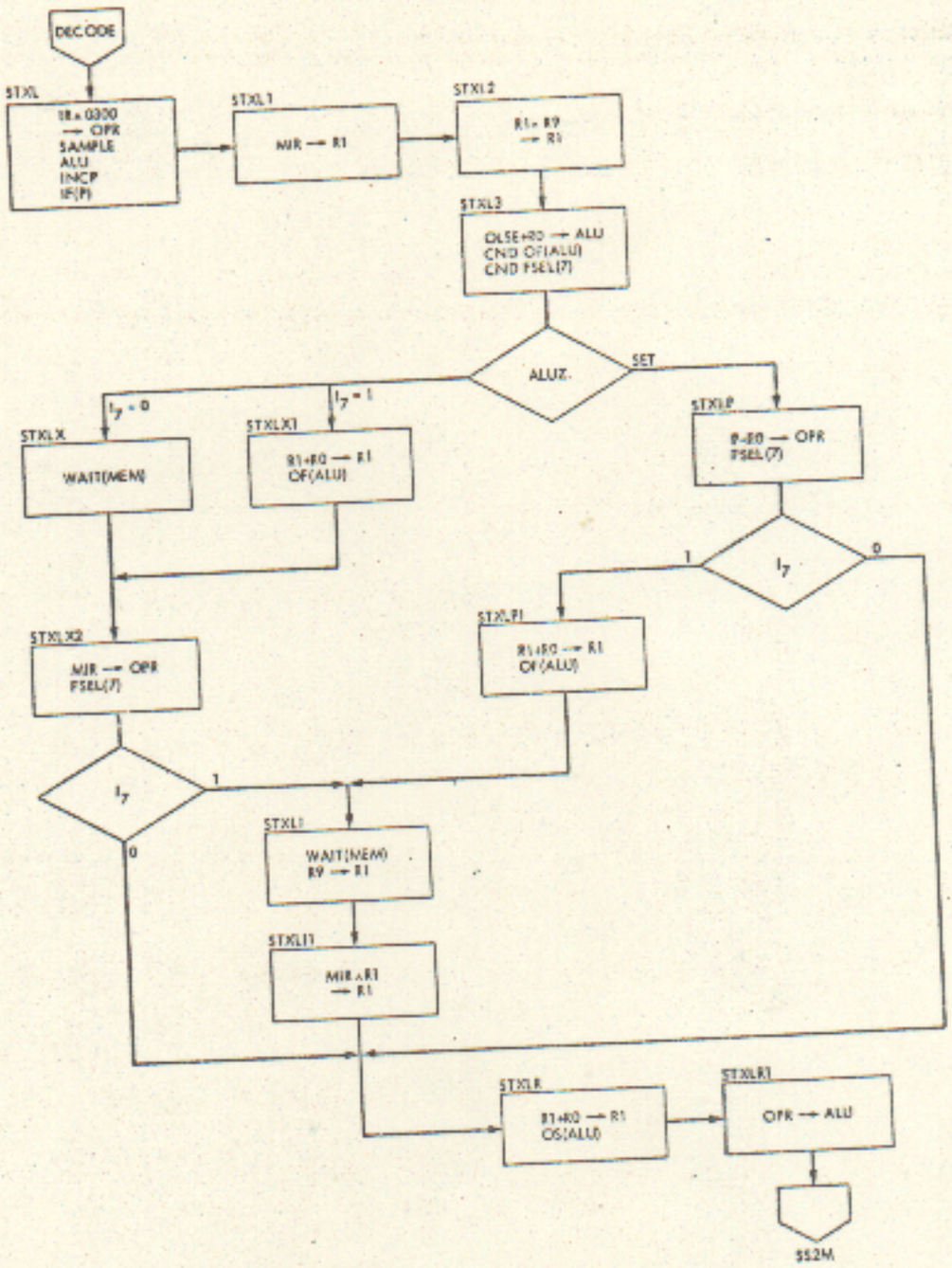
RIGHT SHIFTS


DECODE
ST55
RB → OPR
ST51
OLSE → OPR
ST52
OPR → OPR
(SHIFT RIGHT)
OPR(0) → OPR(15)
1 → BYTA
(OVFL → SIGN)
ST53
OPR → OPR
(SHIFT RIGHT)
OPR(15) → OPR(15)
(CARRY → OPR(1)
(OVFL → SIGN)
DECODE
ST5L
RB → OPR
ST56
OPR → RE
FSEL(10)
ST55
OPR → OPR
(SHIFT LEFT)
OPR(15) → OPR(10)
0 → RE
ST54
OPR → OPR
(SHIFT RIGHT)
OPR(15) → OPR(15)
(CARRY → OPR(0)
(OVFL → SIGN)
(CARRY → OPR(1), OVFL → OPR(0)
(RESET CARRY AND OVFL)
I10
0
1
ST551
IR = 0000
→ OPR
SAMPLE
ALU
ST5L1
IR = 0000
→ OPR
SAMPLE
ALU
INCP
IF(P)
SHORT
OPERAND
FETCH
LONG
OPERAND
FETCH
ST57
OVR(85)
RE ∧ 3 → ALU
ST58
RE ∧ 3
→ ALU
INCP
IF(P)
SS3M

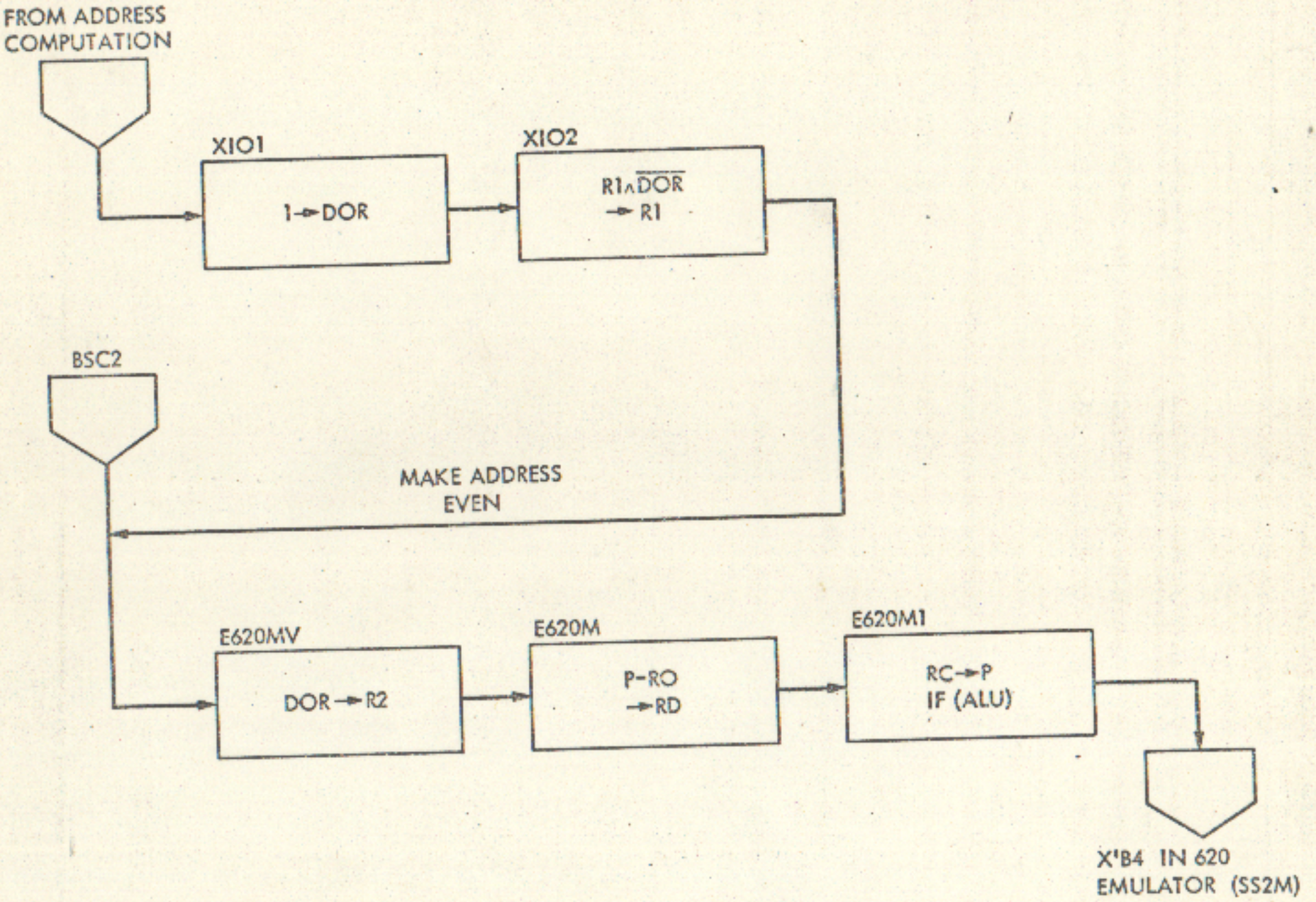
ST5



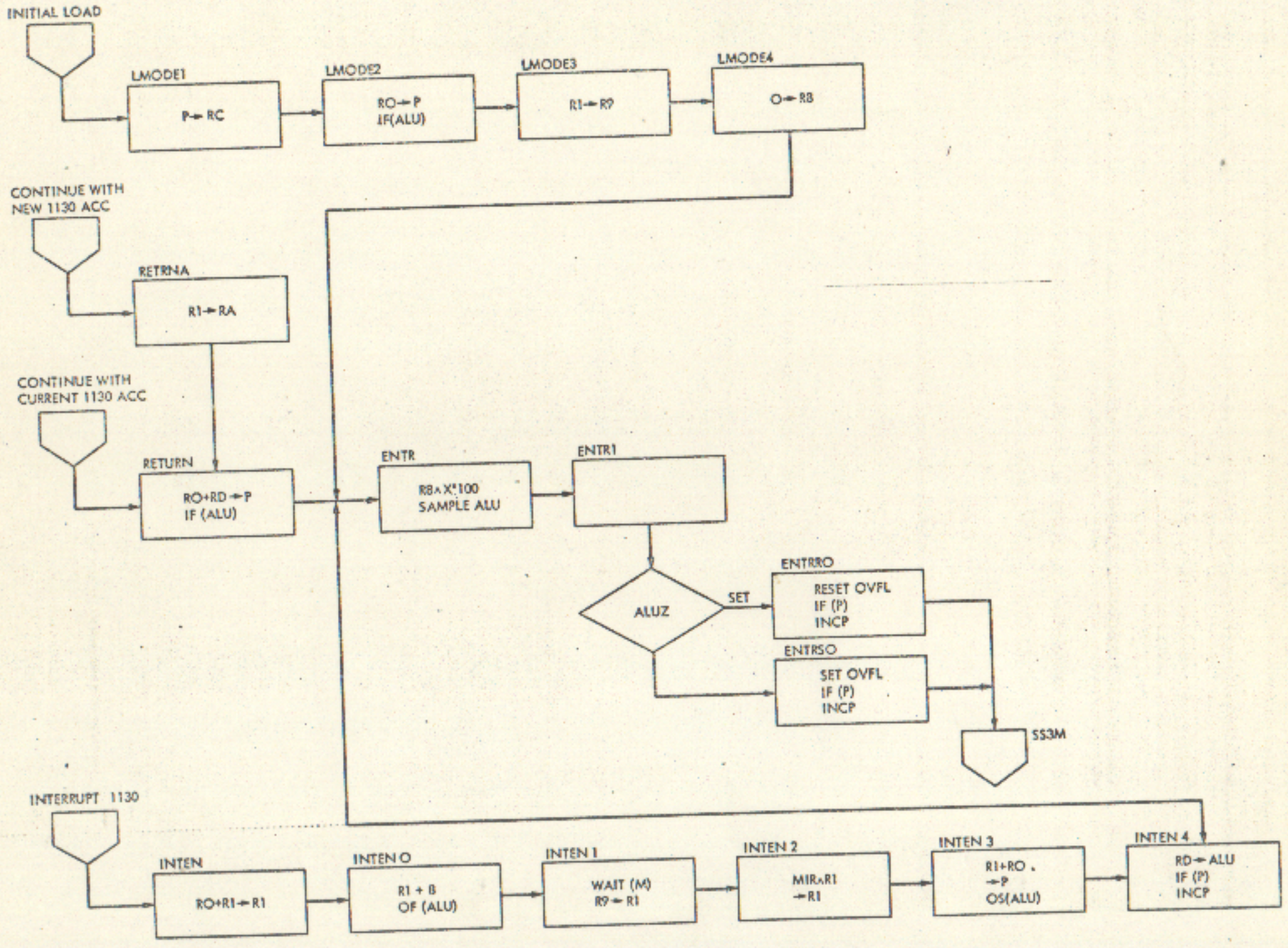
STX
SHORT FORMAT



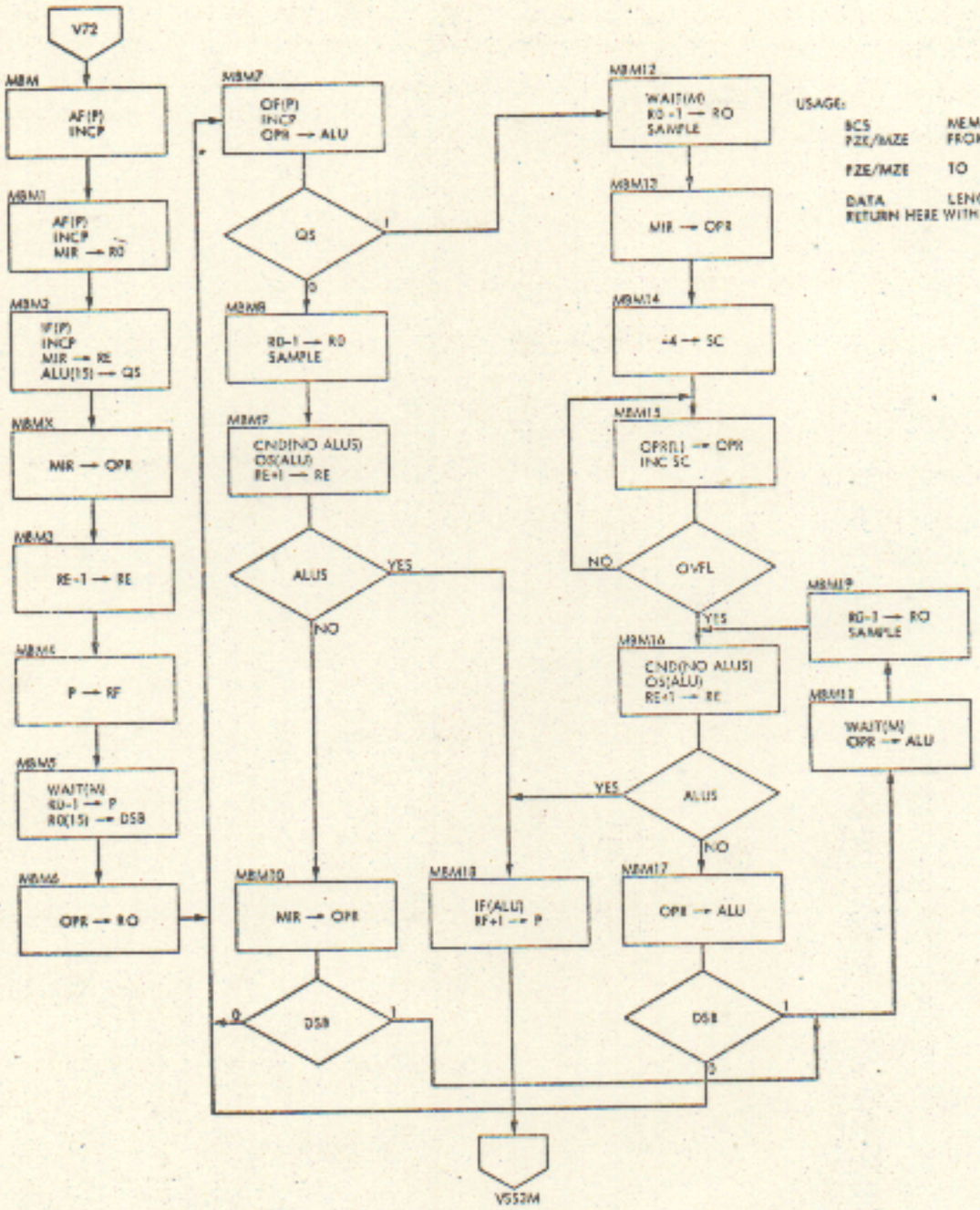
STX
LONG FORMAT



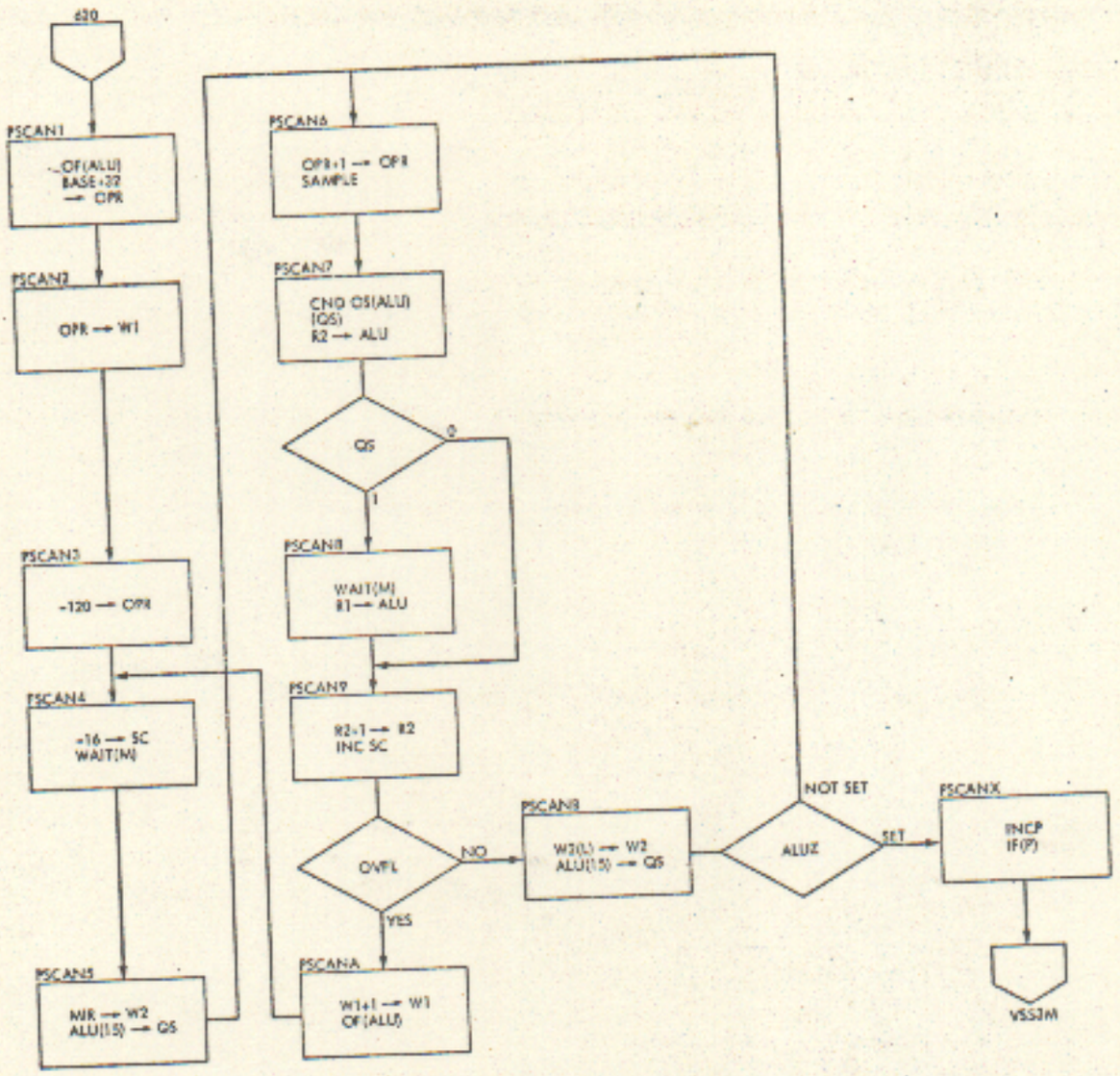
1130→620 COMMUNICATION



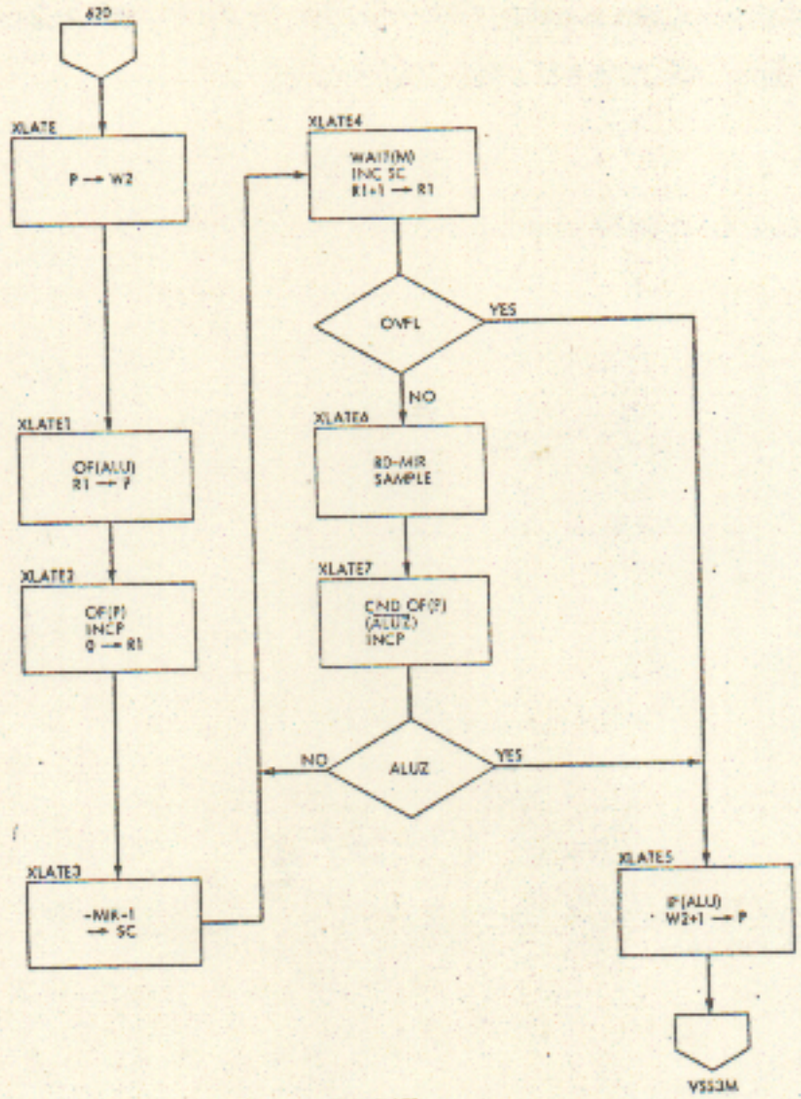
620→1130 COMMUNICATION



MEMORY-TO-MEMORY BLOCK MOVE
FOR I/O SIMULATOR



PRINTER BITMAP SCAN
FOR I/O SIMULATION



TRANSLATE
FOR I/O SIMULATOR