SPECTRA 70

RADIO CORPORATION OF AMERICA • ELECTRONIC DATA PROCESSING

RCA

SYSTEM 70|45

PROCESSOR EO FLOW CHARTS

RCA 70/45 PROCESSOR
EO FLOW CHARTS

Prepared By

EDP ENGINEERING
and PUBLICATIONS
Camden, New Jersey

70-45-101

March 1966

TABLE OF CONTENTS

SECTION 1

SECTION 2

SECTION 3. EO FLOW CHARTS

SECTION 3. EO FLOW CHARTS (Cont'd.)

SECTION 3. EO FLOW CHARTS (Cont'd.)

SECTION 3. EO FLOW CHARTS (Cont'd.)

SECTION 3. EO FLOW CHARTS (Cont'd.)

SECTION 3. EO FLOW CHARTS (Cont'd.)

SECTION 3. EO FLOW CHARTS (Cont'd.)

SECTION 3. EO FLOW CHARTS (Cont'd.)

SECTION 1
INTRODUCTION

SPECTRA 70/45 ELEMENTARY OPERATIONS

INTRODUCTION

The 53-bit EO word pattern is grouped, functionally into 11 Fields as follows: FVC-MSD-TNA-E-I.

FVC Fields

$F = (F)_8 = F(2\text{-}0)$ -- Function

$V = (V)_{16} = V(3\text{-}0)$ -- Sub-Function.

$C = (C)_{16} \cdot (C)_{16} = C(7\text{-}4) \cdot (C3\text{-}0)$ -- Control.

C7,MSD Fields (Data Transfer) $F \neq 5$

$M = (M)_4 = M(1\text{-}0)$ -- FM Control

$S = (S)_4(S)_{16} = S(5\text{-}4)S(3\text{-}0)$ -- FM Address or Source Register

$D = (D)_4(D)_{16} = S(5\text{-}4)S(3\text{-}0)$ -- Source or Destination Register

TNA Fields (Branch Control)

$T = (T)_4(T)_{16} = T(5\text{-}4)T(3\text{-}0)$ -- Test Condition Code

$N = (N)_8(N)_8 = N(5\text{-}3)N(2\text{-}0)$ -- Next EO Address if Test False

$A = (A)_8(A)_8 = A(5\text{-}3)A(2\text{-}0)$ -- Next EO Address if Test True

E Field

$E = (E)_2(E)_{16} = E4 \cdot E(3\text{-}0)$ -- Exception Code

I Field

$I = (I)_2 = IO$ -- Inhibit I/O Servicing

FV = 00

C = 00/80

C7, MSD-TNA-E-I -- Standard

Special: If FVC-MSD-TNA-E-I = All 0's
And if RUN(1)CKROM(0)MAR2(0),
And if NSR(0)/LDFN(0)ISIM(0)IMMINT(0)NORST(0):
Set ROMERR.

This EO is a combination of an F = 7(SET F) EO and an F = 2(PER) EO. The Adder controls are set first as an F = 7 (SETF). The function is then performed once (SPER1) or twice (SPER2) as an F = 2(PER).

FV	Mnemonic	Same as FV #1	Same as FV #2	Description
10	SPER1-X	70	20	Function Unchanged
18	SPER2-X	70	28	Function Unchanged
11	SPER1-DADDf(IR03)	71	20	Decimal Add, IR03 ⟶ ICAR
19	SPER2-DADDf(IR03)	71	28	Decimal Add, IR03 ⟶ ICAR
12	SPER1-BADD	72	20	Binary Add
1A	SPER2-BADD	72	28	Binary Add
13	SPER1-BSUB	73	20	Binary Subtract
1B	SPER2-BSUB	73	28	Binary Subtract
14	SPER1-MUR	74	20	Move UR Unchanged
1C	SPER2-MUR	74	28	Move UR Unchanged
15	SPER1-$\overline{UR}$	75	20	Two's Complement of UR
1D	SPER2-$\overline{UR}$	75	28	Two's Complement of UR
16	SPER1-DADD	76	20	Decimal Add
1E	SPER2-DADD	76	28	Decimal Add
17	SPER1-DSUB	77	20	Decimal Subtract
1F	SPER2-DSUB	77	28	Decimal Subtract

C(6-4,0) -- BAA/BAB/BDB Are First Set/Reset Same As F = 7 (See F=7), Then:

C(6-0) -- The Same As F = 2. (See F = 2)

C7,MSD-TNA-E-I -- Standard; also C7-----WDB After Bus Transfer

This EO performs a function (with 8 bits of DR as the first operand and 8 bits of UR as the second operand) in the 8-bit Adder and places the result in 4/8/16 bit positions of IR. The function that is performed is defined by the Adder control flip-flops EDR, COMP, BCD, SIMC, ELAND, ELOR that have been pre-set by the last F = 1(SPER)/F = 7(SETF) EO and which remain unchanged by the F = 2(PER). The function may be performed once (PER1) or twice (PER2). A PER2 = 1.5 units of EO time.

During every Perform:

If V = 0/V = 8/BDB(0): CAR7 ⟶ SCAR
If $(\overline{V = 0/8})$ · BDB(1): CAR3 ⟶ SCAR
CAR7 ≠ CAR6 ⟶ OV
SUM 8 ⟶ B0
SUM 9 ⟶ B1
CAR9 ⟶ B3

If $\left[V = 0/V = 8/BDB(0)\right]$ · $\left[\text{Adder } (7\text{-}4) \neq 0\right]$: Reset RZ

If $\left[V = 0/V = 8/BDB(1)\right]$ · $\left[\text{Adder } (3\text{-}0) \neq 0\right]$: Reset RZ

After every Perform: SCAR ⟶ ICAR

Function Performed as follows:

DR(9-8) = EDR(1)C7(1)DR1(1-0)
DR(7-0) = EDR(1)C7(0)A(0)DR0(7-0)/EDR(1)C7(0)A(1)DR1(7-0)
DR(7-0) = EDR(1)C7(1)A(0)DR2(7-0)/EDR(1)C7(0)A(1)DR3(7-0)
UR(7-0) = B(0)UR0(7-0)/B(1)UR1(7-0)
$U(9\text{-}8) = COMP(0)\cdot G(1\text{-}0)/COMP(1)\cdot\overline{G(1\text{-}0)}$
$U(7\text{-}0) = COMP(0)BCD(0)UR(7\text{-}0)/COMP(1)\cdot\overline{UR(7\text{-}0)}$
$U(7\text{-}4) = COMP(0)BCD(1)\cdot\left[6 + UR(7\text{-}4)\right]$
$U(3\text{-}0) = COMP(0)BCD(1)\cdot\left[6 + UR(3\text{-}0)\right]$
P(9-8) = U(9-8)·DR(9-8)
$P(7\text{-}0) = \left[ELOR(1)/\overline{U(7\text{-}0)}\right]\cdot DR(7\text{-}0)$
$N(9\text{-}8) = \overline{U(9\text{-}8)}\cdot\overline{DR(9\text{-}8)}$
$N(7\text{-}0) = ELAND(0)\cdot\overline{U(7\text{-}0)}\cdot\overline{DR(7\text{-}0)}$
$SUM(9\text{-}1) = \left[P(9\text{-}1)/N(9\text{-}1)\right]\cdot CAR(8\text{-}0)/\left[\overline{P(9\text{-}1)/N(9\text{-}1)}\right]\cdot\overline{CAR(8\text{-}0)}$
$SUMO = (P0/N0)\cdot ICAR(1)/\overline{(P0/N0)}\cdot ICAR(0)$
$CAR(9\text{-}4) = \overline{N(9\text{-}4)}\cdot CAR(8\text{-}3)/P(9\text{-}4)/SIMC(1)$
$CAR3 = \overline{(V=0/8)}\cdot BDB(0)\cdot ICAR(1)$
$CAR3 = \left[V = 0/V = 8/BDB(1)\right]\cdot\left[\overline{N3}\cdot CAR2/P3/SIMC(1)\right]$
$CAR(2\text{-}1) = \overline{N(2\text{-}1)}\cdot CAR(1\text{-}0)/P(2\text{-}1)/SIMC(1)$
$CAR0 = \overline{N0}\cdot ICAR(1)/P0/SIMC(1)$
$ADDER(7\text{-}4) = \left[BCD(0)/CAR7\right]\cdot SUM(7\text{-}4)/BCD(1)\cdot\overline{CAR7}\cdot\left[A+SUM(7\text{-}4)\right]$
$ADDER(3\text{-}0) = \left[BCD(0)/CAR3\right)\cdot SUM(3\text{-}0)/BCD(1)\cdot\overline{CAR3}\cdot\left[A+SUM(3\text{-}0)\right]$

FVC FIELDS F = 2(PER)

FV	MNEM	BAA(0) → IR1(7-4) / BAA(1) → IR1(7-4)	BDB(0) IRO(3-0) / IR1(3-0)	BDB(1) IRO(7-4) / IR1(7-4)	BDB(1) IRO(3-0) / IR1(3-0)
20 28	PER1 PER2	Adder(7-4)	Adder(3-0)	Adder(7-4)	Adder(3-0)
21 29	PER1-ZONE4B PER2-ZONE4B	Zone	Sum(7-4)	Zone	Adder(3-0)
22 2A	PER1-4BRS PER2-4BRS	Adder(7-4)	RS	Sum(3-0)	RS
23 2B	PER1-4B$\overline{RS}$ PER2-4B$\overline{RS}$	Adder(7-4)	$\overline{RS}$	Sum(3-0)	$\overline{RS}$
24 2C	PER1-4BS PER2-4BS	Adder(7-4)	X	X	Adder(3-0)
25 2D	PER1-4BSZ PER2-4BSZ	Adder(7-4)	0	0	Adder(3-0)
26 2E	PER1-4BC PER2-4BC	X	Sum(7-4)	Sum(3-0)	X
27 2F	PER1-4BCZ PER2-4BCZ	0	Sum(7-4)	Sum(3-0)	0

X = Unchanged

If A(0): Zone = F
If A(1): Zone = 5

If A(0)·[OR7(0)US(0)/OR6(1)OR2(0)DS(0)/OR7(1)OR2(1)·US=DS]: RS=C, $\overline{RS}$=D

If A(0)·[OR7(0)US(1)/OR6(1)OR2(0)DS(1)/OR7(1)OR2(1)·US≠DS]: RS=D, $\overline{RS}$=C

If A(1)·[OR7(0)US(0)/OR6(1)OR2(0)DS(0)/OR7(1)OR2(1)·US=DS]: RS=A, $\overline{RS}$=B

If A(1)·[OR7(0)US(1)/OR6(1)OR2(0)DS(1)/OR7(1)OR2(1)·US≠DS]: RS=B, $\overline{RS}$=A

A(0) = BKAP13(0) = EBCDIC Code

A(1) = BKAP13(1) = ASCII Code

C(7-4)	Mnemonic	C(7-4)	Mnemonic	C(7-4)	Mnemonic	C(7-4)	Mnemonic
0/8	None	2/A	T-A	4/C	T-D	6/E	T-AD
1/9	T-B	3/B	T-AB	5/D	T-BD	7/F	T-ABD

If C(7-4)=2/A/3/B/6/E/7/F: $\overline{E=02\cdot T=1B/22}$ in the next EO.
If C(7-4)=1/9/3/B/5/D/7/F: $\overline{E=02\cdot T=18/23}$ in the next EO.
If C(7-4)=4/C/5/D/6/E/7/F: $\overline{E=02\cdot T=24}$ in the next EO.

The test is made before the Trigger in the current EO.

C(3-0)	OR7(0)/OR6(0) Mnemonic	OR7(1)OR6(1)OR5(0) Mnemonic	OR7(1)OR6(1)OR5(0) Mnemonic
0	None	None	None
4	R-G	R-G	R-G
5	G-1	G-1	G-1
6	G-2	G-2	G-2
7	G-4	G-4	G-4
8	S-R2	R1-1,S-R2	R1-1,S-R2
9	R2-1	R-1	R1,R2-1
A	R2-2	R-2	R1,R2-2
B	R2-4	R-4	R1,R2-4
C	S-R2,R-G	R1-1,S-R2,R-G	R1-1,S-R2,R-G
D	R2,G-1	R,G-1	R1,R2,G-1
E	R2,G-2	R,G-2	R1,R2,G-2
F	R2,G-4	R,G-4	R1,R2,G-4

Special: When C(3-0) = 8/C : T ≠ 09/0A/0B/0E/0F/13/21/2B
When C(3-0) = 4/C : T ≠ 0C/15/1D/1E/1F
G/R1/R2/R cannot be used as a Source/Destination and Triggered in the same EO.
When C(3-0) = 8/C: $\overline{M=(1/3)\cdot S=(16/17)}$
When C(3-0) = 4/C: $\overline{M=(1/3)\cdot S=(14/15/1C/1D/1E/1F)}$

Trigger of A/B/D Occurs After Each Perform; Trigger of R/R1/R2/G Occurs Once

C7,MSD: Standard Except
C7 ⟶ WDB After Bus Transfer

M = 1 · (D = 08/0C) · C7(0)WDB(1): $(S)_{01} \longrightarrow (S)_{23}$

M = 1 · (D = 08/0C) · C7(1)WDB(0): $(S)_{01} \longleftarrow (S)_{23}$

M = 3 · WDB(1): $D \longrightarrow (S)_{23}$ Independent of C7

M = 3 · WDB(0): $D \longrightarrow (S)_{01}$ Independent of C7

TNA-E-I: Standard

If M = 1/3, the least significant 18 bits of the FM (FM11-FM30), as addressed by S, are either incremented/decremented by 0/1/2.

The most significant 14 bits of FM and all 32 bits of DR remain unchanged.

Incrementing/decrementing occurs after any bus transfer.

FV	Mnemonic	Description	M
30	INC+0	No Change	0/1/3
31	INC+1	Increment by 1	1/3
32	INC+2	Increment by 2	1/3
35	INC-1	Decrement by 1	1/3
36	INC-2	Decrement by 2	1/3

C(7-4)	Mnemonic	C(7-4)	Mnemonic
0/8	None	4/C	BUS10 → D
1/9	BUS10 → B	5/D	BUS10 → B,D
2/A	BUS10 → A	6/E	BUS10 → A,D
3/B	BUS10 → A,B	7/F	BUS10 → A,B,D

C(3-0)	Mnemonic	Description Standard
0	INC	
1	MERGE	BUS0 $\xrightarrow{0}$ DR0
2	MERGE	BUS1 $\xrightarrow{1}$ DR1
3	MERGE	BUS $\xrightarrow{01}$ DR0, DR1

C7, MSD: Standard When C(3-0) = 0; When C(3-0) ≠ 0, See the Following:

M	C7		Standard Transfer	+	Second Transfer When C(3-0) ≠ 0
0	0		S → D		S → $DR_{0/1}$
1	1		$(S)_{23}$ → D		$(S)_{23}$ → $(S)_{0/1}$
3	0	$\overline{D = 0B}$	None		D → $(S)_{0/1}$
3	0	D = 0B	DR → $(S)_{\overline{0/1}}$		00 → $(S)_{0/1}$

TNA-E-I: Standard

FV	Mnemonic	Bus
39/3D	V2,C(6-0) —0→	BUS0(7-0)
3A/3E	V2,C(6-0) —1→	BUS1(7-0)
3B/3F	V2,C(6-0) —01→	BUS0(7-0) and BUS1(7-0)

V2,C(6-0): Dropped on Bus as a Constant during Bus Transfer

C7,MSD-TNA-E-I: Standard

Special: When the MSD Fields are used, the generated Constant OR's with the source. The source cannot be MR.

A 1-bit shift (left/right) occurs at the 32-bit FM regeneration matrix from the outputs of DR. The shift occurs after any bus transfer into or out of DR/BO/B2/US/IR, and after any tests on B2/B0 have been made. The shift does not alter the contents of DR.

SLL/SLA: If M = 1/3: $FM(2^{31}-2^{1}) \leftarrow DR(06-30)$, DR07 is Stored, and $FM2^{0}$ is Filled. If M = 0: Only DR07 is Stored.

SRL/SRA: If M = 1/3: $DR(07-31) \rightarrow FM\ (2^{30} - 2^{0})$, DR30 is Stored, and $FM2^{31}$ is Filled. If M = 0, Only DR30 is Stored.

FV	Mnemonic	DR07 Stored In	$FM2^{0}$ Filled From	
40	SLL	X	B2	
41	SLL	B0	B2	
42	SLL	B2	0	
43	SLL	B2	B0	
44	SLA	X	B2	If DR07 ≠ DR06 -- Set OV
45	SLA	B0	B2	If DR07 ≠ DR06 -- Set OV
46	SLA	B2	US≠IRO7	If DR07 ≠ DR06 -- Set OV
47	SLA	B2	US≠IRO7	If DR07 ≠ DR06 -- Set OV

FV	Mnemonic	$FM2^{31}$ Filled From	DR30 Stored In	
48	SRL	B2	X	
49	SRL	B2	B0	
4A	SRL	0	B2	
4B	SRL	B0	B2	
4E	SRA	DR07	B2	
4F	SRA	DR07f(OV)	B2	DR07f(OV) = $\overline{DR07}$ if OV(0) = DR07 if OV(1)

C(7-4)	Mnemonic	C(7-4)	Mnemonic	C(7-4)	Mnemonic	C(7-4)	Mnemonic
0/8	None	2/A	T-A	4/C	T-D	6/E	T-AD
1/9	T-B	3/B	T-AB	5/D	T-BD	7/F	T-ABD

If C(7-4)=2/A/3/B/6/E/7/F: $\overline{E=02 \cdot T=1B/22}$ in the next EO.
If C(7-4)=1/9/3/B/5/D/7/F: $\overline{E=02 \cdot T=18/23}$ in the next EO.
If C(7-4)=4/C/5/D/6/E/7/F: $\overline{E=02 \cdot T=24}$ in the next EO.

The test is made before the Trigger in the current EO.

C(3-0)	OR7(0)/OR6(0) Mnemonic	OR7(1)OR6(1)OR5(0) Mnemonic	OR7(1)OR6(1)OR5(0) Mnemonic
0	None	None	None
4	R-G	R-G	R-G
5	G-1	G-1	G-1
6	G-2	G-2	G-2
7	G-4	G-4	G-4
8	S-R2	R1-1,S-R2	R1-1,S-R2
9	R2-1	R-1	R1,R2-1
A	R2-2	R-2	R1,R2-2
B	R2-4	R-4	R1,R2-4
C	S-R2,R-G	R1-1,S-R2,R-G	R1-1,S-R2,R-G
D	R2,G-1	R,G-1	R1,R2,G-1
E	R2,G-2	R,G-2	R1,R2,G-2
F	R2,G-4	R,G-4	R1,R2,G-4

Special: When C(3-0) = 8/C : T ≠ 09/0A/0B/0E/0F/13/21/2B
When C(3-0) = 4/C : T ≠ 0C/15/1D/1E/1F
When C(3-0) = 8/C: $\overline{M=(1/3) \cdot S=(16/17)}$
When C(3-0) = 4/C: $\overline{M=(1/3) \cdot S=(14/15/1C/1D/1E/1F)}$
G/R1/R2/R cannot be used as a Source/Destination and triggered in the same EO.

C7,MSD-TNA-E-I: Standard

This EO is used for multiple branching rather than standard, 2-way branching.,

Test Field T	D	S	Branch Address	ROM Address	
T*	X*	X	A(5-0)	ROMAR(5-0)	Standard
F*	T	X	N(5-0)	ROMAR(5-0)	Standard
F	F	T	C(7-0),V(3-0)	ROMAR(11-0)	Absolute Address
F	F	F	Drop Thru	Set ROMAR0	Present Address Must Be Even

*T = True -- *F = False -- *X = Don't Care

V -- V(3-0) ⟶ ROMAR(3-0) if $(\overline{T/D})\cdot S$

C -- C(7-0) ⟶ ROMAR(11-4) if $(\overline{T/D})\cdot S$

M = 0

S	Test Condition = T
00	No Test
04	R1 ≠ CC
05	RGO
06	DR1 ≠ 20
07	DR0 ≠ 20

D	Test Condition = T
00	No Test
01	R = 00/OR7(1)OR6(1)OR5(1)·(R1 = 0/R2 = 0)
02	DR1 = 21
03	DR0 = 21

TNA-E-I: Standard

FVC FIELDS F = 6(SETCC)

FV	Mnemonic	OR Class	Set SREG(5-4) 00	01	10	11
60	SETCC-OR	1	RZ(1)	RZ(0)VS(1)	RZ(0)VS(0)	--
		2	RZ(1)OV(0)	RZ(0)OV(0) IR07(1)	RZ(0)OV(0) IR07(0)	OV(1)
		3	RZ(1)SCAR (1)	SCAR(0)	RZ(0)SCAR(1)	--
		4	RZ(1)	RZ(0)	--	--
		5	RZ(1)SCAR (0)	RZ(0)SCAR(0)	RZ(1)SCAR(1)	RZ(0)SCAR(1)
		6	RZ(1)OV(0)	--	RZ(0)OV(0)	OV(1)
		7	--	RZ(0)	--	RZ(1)
		8	RZ(1)	RZ(0)·$\overline{P}$*	RZ(0)·$\overline{P}$*	--

*P = RZ(0)US(1)OV(0)UR07≠IR07
= RZ(0)US(1)OV(1)UR07=IR07
= RZ(0)US(0)OV(0)UR07=IR07

FV	Mnemonic	OR Class	SREG(5-4)
		9	Unchanged
		1-9	00
64	SETCC-0	$\overline{9}$	00
		9	Unchanged
65	SETCC-1	$\overline{9}$	01
		9	Set SREG4
66	SETCC-2	$\overline{9}$	10
		9	Set SREG5
67	SETCC-3		11
68	SETCC-X		Unchanged

OR	Class
8A 8B 8E 8F	-1
12 13 1A 1B	-2
4A 4B 5A 5B	-2
15 1D 55 5D	-3
95 D5 DD	-3
11 14 16 17 1C	-4
54 56 57 5C	-4
94 96 97 D4	-4

OR	Class
D6 D7 DC DE DF	-4
1E 1F 5E 5F	-5
10 90	-6
91	-7
X9·$\overline{89}$	-8
08 18 48 58	-9
88 89 8C 89	-9

C(7-4)	Mnemonic	C(7-4)	Mnemonic	C(7-4)	Mnemonic	C(7-4)	Mnemonic
0/8	None	2/A	T-A	4/C	T-D	6/E	T-AD
1/9	T-B	3/B	T-AB	5/D	T-BD	7/F	T-ABD

If C(7-4)=2/A/3/B/6/E/7/F: $\overline{E=02 \cdot T=1B/22}$ in the next EO.
If C(7-4)=1/9/3/B/5/D/7/F: $\overline{E=02 \cdot T=18/23}$ in the next EO.
If C(7-4)=4/C/5/D/6/E/7/F: $\overline{E=02 \cdot T=24}$ in the next EO.

The test is made before the Trigger in the current EO.

C(3-0)	OR7(0)/OR6(0) Mnemonic	OR7(1)OR6(1)OR5(0) Mnemonic	OR7(1)OR6(1)OR5(0) Mnemonic
0	None	None	None
4	R-G	R-G	R-G
5	G-1	G-1	G-1
6	G-2	G-2	G-2
7	G-4	G-4	G-4
8	S-R2	R1-1,S-R2	R1-1,S-R2
9	R2-1	R-1	R1,R2-1
A	R2-2	R-2	R1,R2-2
B	R2-4	R-4	R1,R2-4
C	S-R2,R-G	R1-1,S-R2,R-G	R1-1,S-R2,R-G
D	R2,G-1	R,G-1	R1,R2,G-1
E	R2,G-2	R,G-2	R1,R2,G-2
F	R2,G-4	R,G-4	R1,R2,G-4

Special: When C(3-0) = 8/C : T ≠ 09/0A/0B/0E/0F/13/21/2B
When C(3-0) = 4/C : T ≠ 0C/15/1D/1E/1F

When C(3-0) = 8/C: $\overline{M=(1/3) \cdot S=(16/17)}$
When C(3-0) = 4/C: $\overline{M=(1/3) \cdot S=(14/15/1C/1D/1E/1F)}$
G/R1/R2/R cannot be used as a Source/Destination and triggered in the same EO.

C7,MSD-TNA-E-I: Standard

FVC FIELDS F = 7(SETF)

FV	Mnemonic	EDR	COMP	ICAR	BCD	SIMC	Eland	ELOR	SCAR	OV	RZ	Function
70	SETF-X	X	X	X	X	X	X	X	X	0	X	None
71	SETF-DADD f(IR03)	1	0	IR03	1	0	0	0	X	0	1	Decimal Add IR03 → ICAR
72	SETF-BADD	1	0	0	0	0	0	0	X	0	1	Binary Add
73	SETF-BSUB	1	1	1	0	0	0	0	X	0	1	Binary Sub
74	SETF-MUR	0	0	0	0	0	0	0	X	0	1	0 + UR
75	SETF-$\overline{UR}$	0	1	1	0	0	0	0	X	0	1	0 + $\overline{UR}$ + 1
76	SETF-DADD	1	0	0	1	0	0	0	X	0	1	Decimal Add
77	SETF-DSUB	1	1	1	1	0	0	0	X	0	1	Decimal Sub
79	SETF-OR	1	1	1	0	1	0	1	X	0	1	Logical OR
7A	SETF-AND	1	0	1	0	1	1	0	X	0	1	Logical AND
7B	SETF-EXO	1	1	1	0	1	0	0	X	0	1	Logical EXOR
7C	SETF-BSUB, $\overline{ICAR}$	1	1	0	0	0	0	0	X	0	1	DR + $\overline{UR}$

C		Mnemonic	C		Mnemonic	C		Mnemonic	
00/80	--	None				04/84	--	R-G	
10/90	--	R-B	11/91	--	S-B	14/94	--	R-G	R-B
20/A0	--	R-A	21/A1	--	S-A	24/A4	--	R-G	R-A
30/B0	--	R-AB	31/B1	--	S-AB	34/B4	--	R-G	R-AB
40/C0	--	R-D	41/C1	--	S-D	44/C4	--	R-G	R-D
50/D0	--	R-BD	51/D1	--	S-BD	54/D4	--	R-G	R-BD
60/E0	--	R-AD	61/E1	--	S-AD	64/E4	--	R-G	R-AD
70/F0	--	R-ABD	71/F1	--	S-ABD	74/F4	--	R-G	R-ABD

Note:

When C=2X/3X/6X/7X : T ≠ 1B/22

When C=1X/3X/5X/7X : T ≠ 18/23

When C=4X/5X/6X/7X : T ≠ 24

When C=X4 : T ≠ 0C/15/1D/1E/1F, $\overline{M=(1/3)\cdot S=(16/17)}$, G cannot be used as a source or destination.

C7, MSD-TNA-E-I: Standard; Also C7 → WDB After Bus Transfer.

Special: If FVC-MSD-TNA-E-I = All 1's

And if RUN(1)CKROM(0)MAR2(0),

And if NSR(0)/LDFN(0)ISIM(0)IMMINT(0)NORST(0):

Set ROMERR.

M FIELD

M	C7	Mnemonic	FM Address Code	Hardware Source Register Code	Hardware Destination Register Code	FM Cycle
0	X	S → D	None	S	D	0
1	0	$(S)_{01}$ → D	S	OB(DR)	D	1
1	1	$(S)_{23}$ → D	S	OB(DR)	D	1
3•D ≠ OB	0	D → $(S)_{01}$	S	D	08(DR)	1
3•D ≠ OB	1	D → $(S)_{23}$	S	D	08(DR)	1
3•D = OB	X	DR → (S)	S	00	00	1 Inhibit FM Strobe

The same Register cannot be used as both a Source and Destination.

M 13	Field S S	Mnemonic	FM Address 11 1	10 2	01 3	00 4 ← PS2.PS1 ← Program State
	00	U10	50	50	30	30 Utility #14/10
	01	U11	51	51	31	31 Utility #15/11
	02	U8	78	58	38	18 Utility #18/16/12/8
	03	Flag	23	23	23	23 Interrupt Flag Register
	04	Mask	20	24	28	0C Interrupt Mask Register
	05	INTST	21	25	29	0D Interrupt Status Register
	06	PC	22	26	2A	0E Program Counter
	07	Weight	2F	2F	2F	0F Interrupt Weight Register
	08	U1	00	00	00	00 Utility #1
	09	U2	01	01	01	01 Utility #2
	0A	U3	02	02	02	02 Utility #3
	0B	U4	03	03	03	03 Utility #4
	0C	U5	04	04	04	04 Utility #5
	0D	U6	05	05	05	05 Utility #6
	0E	AAD	06	06	06	06 First Operand Address
	0F	BAD	07	07	07	07 Second Operand Address

M 1 3	Field S S	Mnemonic	FM Address (I/O) Trunk No. T(2-0) 0 MUX	1 SEL1	2 SEL2	3 SEL3	4 SEL4	5 SEL5	6 SEL6	Register
	11	MU1	11	19*	31*	39*	51*	59*	79*	I/O Utility #1
	12	CAR	12	1A	32	3A	52	5A	7A	Channel Address Reg.
	13	CCR2	13	1B	33	3B	53	5B	7B	Channel Command Reg. 2
	14	CCR1	14	1C	34	3C	54	5C	7C	Channel Command Reg. 1
	15	ASSY/ST	15	1D	35	3D	55	5D	7D	Assembly Status Reg.
	16	MU2	16	1E	36	3E	56	5E	7E	I/O Utility #2
	17	MU3	17	1F	37	3F	57	5F	7F	I/O Utility #3

* Unused

FM Address ($\overline{I/O}$)

M 1 3	Field S	Mnemonic	11 1 FMAR (6-4)	10 2 FMAR (6-4)	01 3 FMAR (6-4)	00 ← PS2·PS1 4 ← Prog. State FMAR (6-4)	FMAR (3-0)		FM Register	
	10	B	6	4	2	0	B(3-0)		GP Reg 0-15	If B=0:Reset EDR (See F=7)
	11	B+1	6	4	2	0	B(3-0)	FMAR0=1	GP Reg 1-15	If B=0:Reset EDR (See F=7)
	12	R1	6	4	2	0	R1(3-0)		GP Reg 0-15	
	13	R1+1	6	4	2	0	R1(3-0)	FMAR0=1	GP Reg 1-15	
	14	R2	6	4	2	0	R2(3-0)		GP Reg 0-15	
	15	R2+1	6	4	2	0	R2(3-0)	FMAR0=1	GP Reg 1-15	
	16	G	G(7-5)	G(7-5)	G(7-5)	G(7-5)	G(4-1)		All Reg.	If G0(1): C7=1 If G0(0) C7=C7
	17	G+1	G(7-5)	G(7-5)	G(7-5)	G(7-5)	G(4-1)	FMAR0=1	All Odd Reg.	If G0(1): C7=1 If G0(0): C7=C7
	18	U9	7	5	3	1	9		Utility Reg. #19/17/13/9	
	1A	R1	7	7	7	7	R1(3-0)		FP Reg. 0/2/4/6	
	1B	R1+1	7	7	7	7	R1(3-0)	FMAR0=1	FP Reg. 1/3/5/7	
	1C	R2	7	7	7	7	R2(3-0)		FP Reg. 0/2/4/6	
	1D	R2+1	7	7	7	7	R2(3-0)	FMAR0=1	FP Reg. 1/3/5/7	
							FMAR (3-2)	FMAR (1-0)		
	1E	R	0	0	0	0	0	R2(3-2)	Utility Reg. 1-4	If R21(1): C7=1 If R21(0): C7=C7
	1F	R +1	0	0	0	0	0	R2(3-2) FMAR0=1	Utility Reg. 1-4	If R21(1): C7=1 If R21(0): C7=C7

M 0 3	Field S D	Mnemonic	Source Register	
	00	00	None	
	01	SAR	SAR (7-0) ⟶ BUS0(7-0)	
	02	IAR	IAR (7-0) ⟶ BUS0(7-0)	
	03	DIN	DIN (7-0) ⟶ BUS1(7-0)	See Maint. Manual
	04	STR	STR (7-0) ⟶ BUS1(7-0)	See Maint. Manual
	05	MPKC	MKEY(4-1) ⟶ BUS0(7-4) COMM(3-0) ⟶ BUS0(3-0)	
	06	CSR	PCI ⟶ BUS17 INCLGT ⟶ BUS16 PROGCK ⟶ BUS15 PROTCK ⟶ BUS14 DATACK ⟶ BUS13 CONTCK ⟶ BUS12 CHEND ⟶ BUS11 CHEND·INT ⟶ BUS10	See Maint. Manual
	07	R1	R1(3-0) ⟶ BUS1(3-0)	
	08	R2	R2(3-0) ⟶ BUS1(3-0)	
	09	R	R1(3-0) ⟶ BUS1(7-4) R2(3-0) ⟶ BUS1(3-0)	
	0A	IR	IR0(7-0) ⟶ BUS0(7-0) IR1(7-0) ⟶ BUS1(7-0)	
	0B	DR_{01}	DR0(7-0) ⟶ BUS0(7-0) DR1(7-0) ⟶ BUS1(7-0)	If C7(0)·M≠3 If C7(0)·M≠3
		DR23	DR1(1-0) ⟶ BUSA(1-0) DR2(7-0) ⟶ BUS0(7-0) DR3(7-0) ⟶ BUS1(7-0)	If C7(1)·M≠3 If C7(1)·M≠3 If C7(1)·M≠3
	0C	MR	MR0(8-0) ⟶ BUS0(8-0) MR1(8-0) ⟶ BUS1(8-0)	If BUS08·BUS0(7-0)/ BUS18·BUS1(7-0)=even·even/odd·odd: Set MRPE, INT02
	0D	PS	PS(2-1) ⟶ BUS0(6-5)	After Transfer, If R1=0·R23(0)·R22(0): Set PS=00 If $\overline{R1=0·R23(0)·R22(0)}$: Set PS=01
	0E	G	G(7-0) $\xrightarrow{0}$ BUS0(7-0)	
	0F	G	G(7-0) $\xrightarrow{1}$ BUS1(7-0)	
	10	UR	UR0(7-0) ⟶ BUS0(7-0) UR1(7-0) ⟶ BUS1(7-0)	
	11	B	B(3-0) ⟶ BUS0(3-0)	
	12	BIR	B(1-0) ⟶ BUSA(1-0) IR0(7-0) ⟶ BUS0(7-0) IR1(7-0) ⟶ BUS1(7-0)	

M 0 3	Field S D	Mnemonic	Source Register	
	13	S	SREG(7-0) → BUS0(7-0)	
	14	INT1	INT(32-25) → BUS0(7-0)	Reset INT(32-17) After Transfer
			INT(24-17) → BUS1(7-0)	
	15	INT2	INT(16-09) → BUS0(7-0)	Reset INT(16-01) After Transfer
			INT(08-01) → BUS1(7-0)	
	16	IR1	IR1(7-0) → BUS1(7-0)	
	17	DEVAD	DEVSW(7-0) → BUS0(7-0)	
	18	MAR	MAR0(7-0) → BUS0(7-0)	Set MARCON, MRCON
			MAR1(7-0) → BUS1(7-0)	
	19	CFF1	BCD → BUS07	Reset MARCON
			ICAR → BUS06	
			RZ → BUS05	
			SCAR → BUS04	
			FMAR(6-0) → BUS1(6-0)	
	1C	ORR	OR(7-0) → BUS0(7-0)	
			R1(3-0) → BUS1(7-4)	
			R2(3-0) → BUS1(3-0)	
	1D	SG	SREG(7-0) → BUS0(7-0)	
			G(7-0) → BUS1(7-0)	
	1E	BKAP	B(3-0) → BUS0(3-0)	
			K(7-4) → BUS1(7-4)	
			A → BUS13	
			PRIV → BUS10	
	1F	MACHE	MRPE → BUS01	Reset MRPE After Transfer
			DRPE → BUS00	Reset DRPE After Transfer
	20	MPMR	MPMR(3-0) → BUS1(7-4)	
	21	ETCC	1 → BUS00	If 60 cycle
			1 → BUS01	If 50 cycle

M 0 1 Field D D	Mnemonic	Destination Register	
00	None	None	
01	DOUT	BUS0(8-0) → DOUT(8-0)	If MAR10(0). See Maint. Manual
		BUS1(8-0) → DOUT(8-0)	If MAR10(1).
02	CHR	BUS07 → MCDF	See Maint. Manual
		BUS06 → MCCF	
		BUS05 → MSLIF	
		BUS04 → MSKF	
		BUS03 → MPCIF	
		BUS03 → ITSCAN	
		BUS03 → CHR5	
		BUS0(2-0) → CHR(2-0)	
03	IAR	BUS0(7-0) → IAR(7-0)	See Maint. Manual
04	KCR	BUS0(7-4) → MKEY(4-1)	See Maint. Manual
		BUS0(3-0) → COMM(3-0)	
05	FLR	BUS07 → MCDF	See Maint. Manual
		BUS06 → MCCF	
		BUS05 → MSLIF	
		BUS04 → MSKF	
		BUS03 → MPCIF	
		BUS1(7-1) → MCHS(7-1)	
06	TR		See Maint. Manual
07	TICR		See Maint. Manual
08	DR_{01}	BUS0(7-0) → DR0(7-0)	If F=2·M=1/3·WDB(0)
		BUS0(7-0) → DR0(7-0)	If $\overline{C7(0)\cdot F=2\cdot M=1/3\cdot F=3\ V3(0)\cdot \overline{C1(1)}}$
		BUS1(7-0) → DR1(7-0)	If F=2·M=1/3·WDB(0)
		BUS1(7-0) → DR1(7-0)	If $\overline{C7(0)\cdot F=2\cdot M=1/3\cdot F=3\cdot V3(0)\cdot \overline{C0(1)}}$

M 0 1	Field D D	Mnemonic	Destination Register	
	08	DR_{23}	00 ⟶ DR1(7-2)	If F=2•M=1/3•WDB(1)•(M=0•S=12/M=3•D=12)
			00 ⟶ DR1(7-2)	If C7(1)•(M=0•S=12/M=3•D=12)•$\overline{F=2•M=1/3}$•$\overline{F=3•V3(0)}$•[C1(1)/C0(1)]
			BUSA(1-0) ⟶ DR1(1-0)	If F=2 M=1/3 WDB(1) (M=0 S=12/M=3 D=12)
			BUSA(1-0) ⟶ DR1(1-0)	If C7(1)•(M=0•S=12/M=3•D=12)•$\overline{F=2•M=1/3}$•$\overline{F=3•V3(0)}$•[C1(1)/C0(1)]
			BUS0(7-0) ⟶ DR2(7-0)	If F=2•M=1/3•WDB(1)
			BUS0(7-0) ⟶ DR2(7-0)	If C7(1) F=2 M=1/3 F=3 V3(0) C1(1)/C0(1)
			BUS1(7-0) ⟶ DR3(7-0)	If F=2•M=1/3•WDB(1)
			BUS1(7-0) ⟶ DR3(7-0)	If C7(1)•F=2•M=1/3•F=3•V3(0)[C1(1)/C0(1)]
	09	PS	BUS0(6-5) ⟶ PS(2-1)	If R20(1)
			R2(2-1) ⟶ PS(2-1)	If R20(0)
	0A	R	BUS1(7-4) ⟶ R1(3-0) BUS1(3-0) ⟶ R2(3-0)	
	0B	OR,R	BUS0(7-0) ⟶ OR(7-0)	If M=0•S=0C: Set PCKROM
			BUS1(7-4) ⟶ R1(3-0)	T≠01/07/09/0C/15
			BUS1(3-0) ⟶ R2(3-0)	
	0C	IR,DR	Same As 08 & 0E	
	0D	IR,DS	Same As 0E. Also	
			BUS07 ⟶ DS	If E≠06/07
			DRX(3-0)=3/5/B/D ⟶ DS	If E=06/07
	0E	IR	BUS0(7-0) ⟶ IR0(7-0)	
			BUS1(7-0) ⟶ IR1(7-0)	
	0F	IR,UR	Same As 0E & 10	

M 0 1	Field D D	Mnemonic	Destination Register	
	10	UR	BUS0(7-0) → UR0(7-0) BUS1(7-0) → UR1(7-0)	Reset PCKROM
	11	UR,US	Same As 10. Also BUS07 → US URX(3-0)=3/5/B/D → US	If E≠06/07 If E=06/07
	12	UR,B	BUS0(7-4) → B(3-0) 0 → UR0(7-4) BUS0(3-0) → UR0(3-0) BUS1(7-0) → UR1(7-0)	
	13	GUR	Same As 10. Also 00 → G(7-2) BUSA(1-0) → G(1-0)	
	18	G	BUS1(7-0) → G(7-0)	
	1A	S	BUS0(7-0) → SREG(7-0)	
	1B	KEY,A	BUS1(7-4) → K(7-4) BUS13 → A BUS10 → PRIV	
	1C	MAR1	BUSA(1-0) → MARA(1-0) BUS0(7-0) → MAR0(7-0) BUS1(7-0) → MAR1(7-0)	Gen HSM clock. Code 1C/1D may not be used again in the next 2 EO's. I=1 in current EO and the next. If PCKROM(1)NSR(1)V(2-0)=2: Set CKROM. M=0·(S=0A/12/16)·F=(1/2)
	1D	MAR2	BUSA(1-0) → MARA(1-0) BUS0(7-0) → MAR0(7-0) BUS1(7-0) → MAR1(7-0)	Same as Code 1C. Also Set MAR2. Next EO pattern= zero and the I bit applies to the zero EO. The A/N fields may be odd/even
	1E	MR1	BUS1(8-0) → MR0(8-0) BUS1(8-0) → MR1(8-0)	If MAR10(0).Gen. parity → Bus 18 If MAR10(1).Gen. parity → Bus 18
	1F	MR2	BUS0(8-0) → MR0(8-0) BUS1(8-0) → MR1(8-0)	Gen. parity → BUS08,BUS18
	20	MPMAR	BUSA(1-0) → MPMAR BUS0(7-3) → MPMAR	
	21	MPMR	BUS1(7-4) → MPMR(3-0)	

Field

T: Selects various test conditions to be tested. If the test is true (T-True), the A Field is the address of the next EO within the block of 64. If the test is False/No Test (T-False), the N Field is the address of the next EO within the block of 64. The test and the next EO address is determined prior to the execution of the EO. (See Test Condition Codes.)

A: If T-True: A(5-0) ⟶ ROMAR(5-0). ROMAR(11-6) remains unchanged.

N: If F ≠ 5·T-False/F = 5·T-False·D-True: N(5-0) ⟶ ROMAR(5-0). ROMAR(11-6) remains unchanged.

Special

The T Field cannot select a Register or indicator to be tested which is also selected by the D Field (when M=0/1) as a destination.

T	Mnemonic	Test True(T) If --
00	None	Always False (F)
01	OR0=0	OR0(0)
02	OR1=0	OR1(0)
03	OR2=0	OR2(0)
04	OR3=0	OR3(0)
05	OR4=0	OR4(0)
06	OR5=0	OR5(0)
07	OR6=0	OR6(0)
08	MR07=1	MR07=(1)
09	R1/R2=0	R1=0•R2=0/OR7(1)OR6(1)OR5(1) • (R1=0/R2=0)
	R=0	R1=0 R2=0/OR7(1)OR6(1)OR5(1) • (R1=0/R2=0)
0A	RGO	$\overline{\text{R1=0•R2=0/R1=F•R2=F}}$
0B	RLO	R1=F•R2=F
0C	G5=1	G5(1)/OR5(1)/OR=41
	OR5(1)/LA	G5(1)/OR5(1)/OR=41
0D	R1L0	R1=F
	R1=F	R1=F
0E	R2=0	R2=0
0F	R2L0	R2=F
	R2=F	R2=F
10	DR1=FS	DR1=22
11	DR0=FS	DR0=22
12	DS=0	DS(0)
13	R21=0	R21(0)
14	R1=0	R1=0
15	G4=0	$\overline{\text{OR7(1)OR6(1)OR5(1)}}$ • G4(0)/OR7(1)OR6(1)OR5(1)G6(0)
	G6=0	$\overline{\text{OR7(1)OR6(1)OR5(1)}}$ • G4(0)/OR7(1)OR6(1)OR5(1)G6(0)
16	B0=1	B0(1)
17	IR07=1	IR07 (1)
18	URX1G9(B)	URX(3-0)=A/B/C/D/E/F
19	BIR= +1	B3(1)B1(0)B0(0)• IR=0001
1A	IR=0	IR=0000
1B	IRX1=0($\overline{\text{A}}$)	BAA(1)• IR0(3-0)=0/BAA(0) • IR1(3-0)=0
1C	R1=1	R1=1
1D	G=00/80	G=00/80
1E	GLO	G=FF
	G=FF	G=FF
1F	GG4	$\overline{\text{G=00/01/02/03/04}}$
20	RZ=1	RZ(1)
21	RLO/NC	RZ(0)/R1=F • R2=F/SCAR(0)
22	A=1	BAA(1)
23	B=1	BAB(1)
24	D=1	BDB(1)
25	SCAR=0	SCAR(0)
26	US=0	US(0)
27	MUX	
28	SEL	
29	OV=1	OV(1)
2A	US=IR07	US(0)IR07(0)/US(1)IR07(1)
2B	B2=1/R2=0	B2(1)/R2=0
	B2=1	B2(1)/R2=0

T	Mnemonic	Test True(T) If --
2C	BUSY	[BURST(1)/SELBUSY/M/SSERV(0)]·[I/O INTB(1)CHR= 02/04/06·SFL$_C$INT M/SSERV(0)]· [CHR=00·PBURST(1)· MUXINT]
2D	NPR/CCk	NPROGCK(1)/NCONTCK(1)
2E	SETINT	MUXSINT/SELSINT
2F	SCC	STR(3-0)=5·(MUX·MCCF/SEL·SCCF)
30	LOAD	LDFN(1)
31	READY	BREADY(1)
32	END	MUX·MT$_t$END/SEL·S$_C$T$_t$ END
33	SK/WR	MWRITE(1)
34	RR	MUX·COMM=02/SEL$_C$·ScREV(1)
35	SR	MUX·CATSUP(1)/STR7(1)/MCHEND(1)/SCHEND(1)
36	CDF	MCDF(1)/SCDF(1)
37	T=0	MTR(1)/SEL$_C$ · S$_C$TR(1)
38	TIC	MTIC(1)/STIC(1)
39	CCF	[STR(3-0)=4/5]·[MTERANY(0)·MCCF(1)/STERANY(0)·SCCF(1)]
3C	SIP	[M/SSERV(0)·I/OINTB(0)]·[CHR=01/02/03·SEL$_C$INT]
3D	SKP	MSKP(1)/SSKP(1)
3E	UNS	AVAIL/[I/O CCO]
3F	SELBUSY	[M/SSERV(0)·I/OINTB(0)] · [CHR=01/02/03·S$_C$BUSY(1)]

E	Mnemonic	Exception/Check
00	None	No Exception/Check
01	UJ	Unconditional Jump -- A(5-0) ⟶ ROMAR(11-6). If T-False: N(5-0) ⟶ ROMAR(5-0), Next EO=AANN. If T-True: A(5-0) ⟶ ROMAR(5-0), Next EO=AAAA. Used to Enter or Exit a Block of 64 EO's.
02	EI	End Instruction -- If T-False: N(5-0) ⟶ ROMAR(5-0), ROMAR(11-6) Remains Unchanged; Next EO Address = XXNN. If T-True: A(5-0) ⟶ ROMAR(5-0), Where AA = 00/01, and 1. If a Unit of Time has Passed on the Elapsed Time Clock, 15 ⟶ ROMAR(11-6); Next EO = 1500/1501, First EO of the Elapsed Time Clock Flow. 2. If not (1) and if any INT(32-1) is set, 13 ⟶ ROMAR (11-6); Next EO = 1300/1301, First EO of Interrupt 3. If Neither (1) nor (2), 11 ⟶ ROMAR(11-6); Next EO = 1100/1101, First EO of Staticizing
03	ES	End Staticizing -- If T-False: N(5-0) ⟶ ROMAR(5-0), ROMAR(11-6) Remains Unchanged and Next EO = XXNN. If T-True: A(5-0) ⟶ ROMAR(5-0) Where AA = 00/01 and 0 ⟶ ROMAR(11-9) and OR(7-0) ⟶ ROMAR(8-1); Next EO = 0XXX, First EO of Execution of an Instruction
04	ASM	Address Shaded Memory -- The HSM Address in MAR is the Address of a Portion of HSM, Non-addressable by Normal Programming; Used Mainly in MUX flows
05	IA	Instruction Address -- If MAR10(1): Set INT24
06	DEB	Data Error Both Digits -- If [F = (1/2)/RIUR] · URX(7-4) ≠ 0-9: Set INT25 If [F = (1/2)/RIUR] · $\overline{\text{RIUS}}$ URX(3-0) ≠ 0-9: Set INT25 If RIUS URX(3-0) = 0-9: Set INT25 If [F = (1/2)/RIDR] · DRX(7-4) ≠ 0-9: Set INT25 If [F = (1/2)/RIDR] · $\overline{\text{RIDS}}$ DRX(3-0) ≠ 0-9: Set INT25 RIDS · DRX(3-0) = 0-9: Set INT25 Where: RIUR = F ≠ 5 · M = (0/1) · D = (0F/10/11/12/13) RIUS = F ≠ 5 · M = (0/1) · D = 11 RIDR = F ≠ 5 · [M = (0/1) · D = (08/0C)/M = 3·D = 0B] RIDS = F ≠ 5 · M = (0/1) · D = 0D Also See Destination Register Codes #11 and #0D

E	Mnemonic	Exception/Check
07	DEM	Data Error Most Significant Digit -- If [F = (1/2)/RIUR] URX(7-4) ≠ 0-9: Set INT25 If [F = (1/2)/RIDR] DRX(7-4) ≠ 0-9: Set INT25 Where: RIUR = F ≠ 5 • M = (0/1) • D = (0F/10/11/12/13) RIDR = F ≠ 5 • [M = (0/1) • D = (08/0C)/M = 3 • D ≠ 0B] Also See Destination Register Codes #11 and #0D
08	EX	Execute -- If OR = 44: Set INT24
09	HW	Half Word Address -- If IR10(1): Set INT24
0A	FW	Full Word Address -- If IR11(1)/IR10(1): Set INT24
0B	DW	Double Word Address -- If IR12(1)/IR11(1)/IR10(1): Set INT24
0C	R	R Specification If R10(1)/OR5(1)R13(1): Set INT24 If OR6(0)OR5(1) • [R23(1)/R20(1)]: Set INT24
0D	PRIV	Privileged -- If PRIV(1): Set INT22
0E	NOOP	Illegal Instruction -- Set INT23
0F	R2	R2 Specification -- If R23(1)/RZ(1)/SCAR(0): Set INT24
10	DE	Data Error -- Set INT25
14	FOV	Fixed Pt. Overflow -- If OV(1) SREG3(1): Set INT31 If OR = (8B/8F) SREG3(1): Set INT31
15	EOV	Exponent Overflow -- Set INT26
16	ETC	Elapsed Time Clock -- Set INT17
17	ESRV	End I/O Servicing -- Indicates Next EO is Last EO in Present I/O Servicing Routine Flow; If No Service Requests Presents, Control Returns to Normal EO Flow
18	DIVI	Divide Error -- Set INT27
19	SIGE	Significance Error -- If SREG0(1): Set INT28

E	Mnemonic	Exception/Check
1A	EUN	Exponent Underflow -- If SREG1(1): Set INT29
1B	DOV	Decimal Overflow -- If SREG2(1): Set INT30
1C	BJX1	Block Jump Set -- Set ROMAR9; ROMAR(11-10,8-6) Remain Unchanged; TNA - Standard
1D	BJX0	Block Jump Reset -- Reset ROMAR9; ROMAR(11-10,8-6) Remain Unchanged; TNA - Standard

NOTE: When E = 06/07, conditions are tested after the data transfer and during any Performs.

When E = 05/09/0A/0B/0F/14, conditions are tested before execution.

INT(1-2,22-26,28-29) or OR1(1)INT27 being set cause an immediate jump to EO #1300, the first EO of the Interrupt routine.

INT(3-21,30-32) and OR1(0)INT27 being set wait until E = 02.

I FIELD INHIBIT I/O SERVICING

I	Mnemonic	Function
0	None	Allows MUX/SEL Servicing Between Current EO and Next EO
1	I	Inhibits MUX/SEL Servicing Between Current EO and Next EO

Whenever M = (0/1) • D = 1C/1D, I(1) in the current EO and next EO (a blank is considered an EO).

The I bit can not be used in consecutive EO's whose total time plus the time of the next EO without an I bit is greater than 4 EO time units.

If the next EO is a blank, the I bit also applies to the blank.

In the Normal mode, the next normal EO address is preserved in REGAD and MREGAD -- ROMAR(11-0) ⟶ REGAD(11-0), MREGAD(11-0). When MUX servicing breaks in, REGAD is undisturbed. During MUX servicing, the next MUX EO address is preserved in MREGAD -- ROMAR(11-0) ⟶ MREGAD(11-0). I bits are used during MUX servicing to inhibit SEL servicing. When SEL servicing breaks in, REGAD and MREGAD are undisturbed.

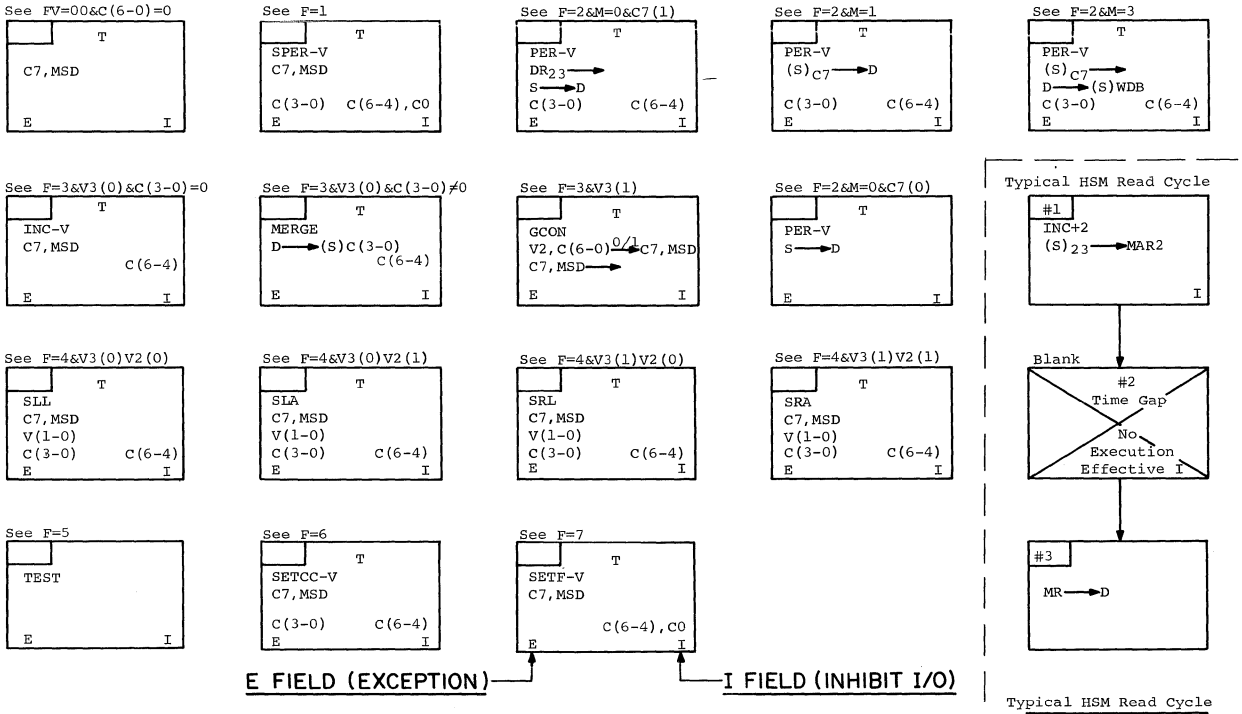
When returning from SEL service to MUX service, MREGAD(11-0) ⟶ ROMAR(11-0) and the BPU continues MUX service. When returning from SEL/MUX service to Normal mode, REGAD(11-0) ⟶ ROMAR(11-0) and the BPU continues in the Normal mode.

SECTION 2

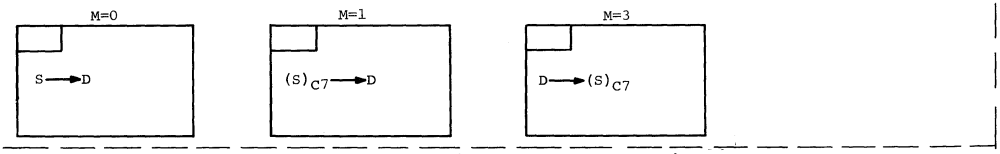
EO FLOW CHART NOTATION

EO FLOW CHART NOTATION

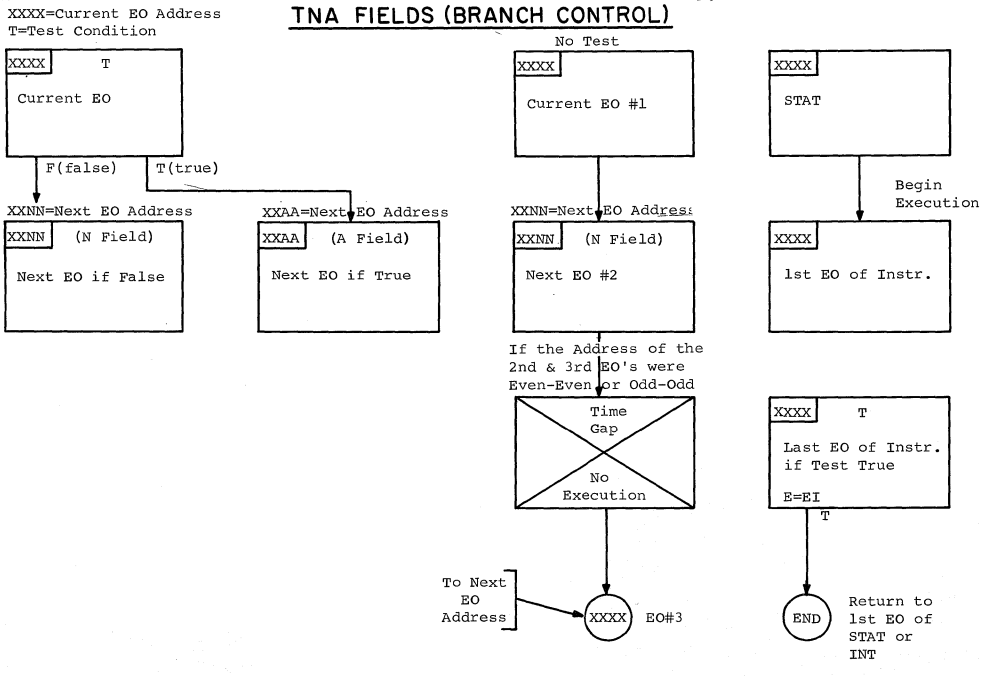
FVC FIELD (FUNCTION)

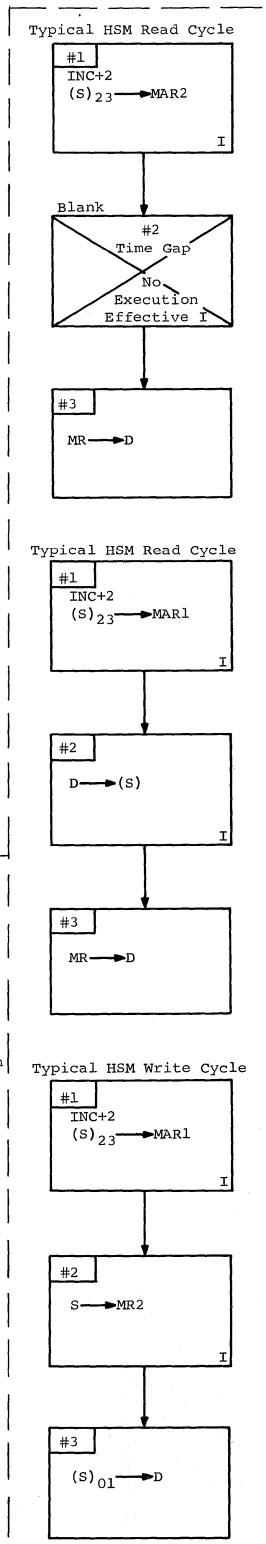


C7,MSD FIELDS (DATA TRANSFER) F≠5



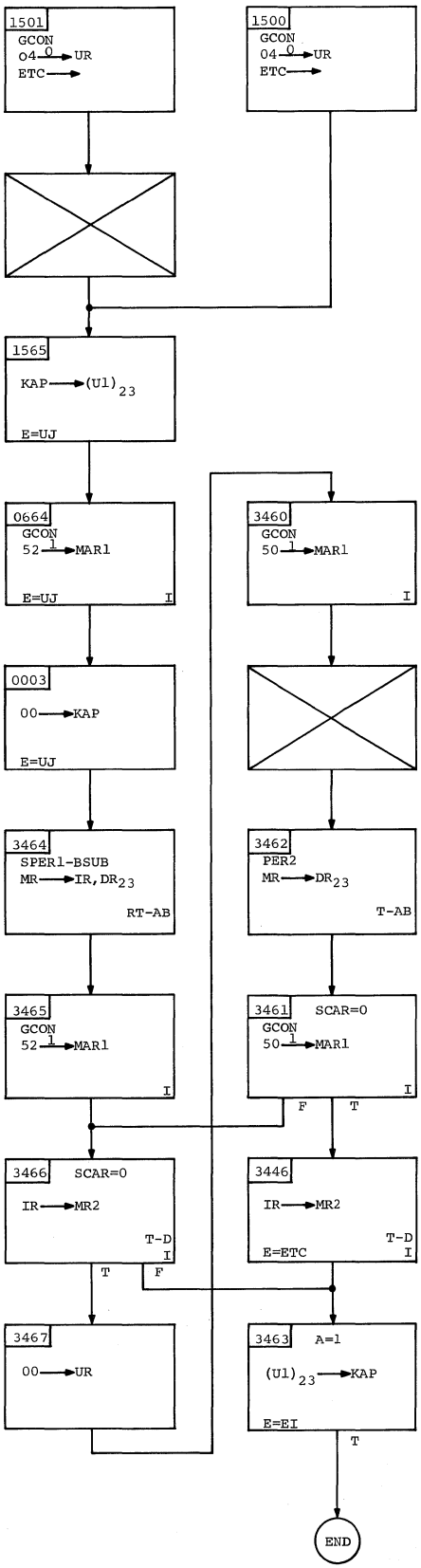
TNA FIELDS (BRANCH CONTROL)



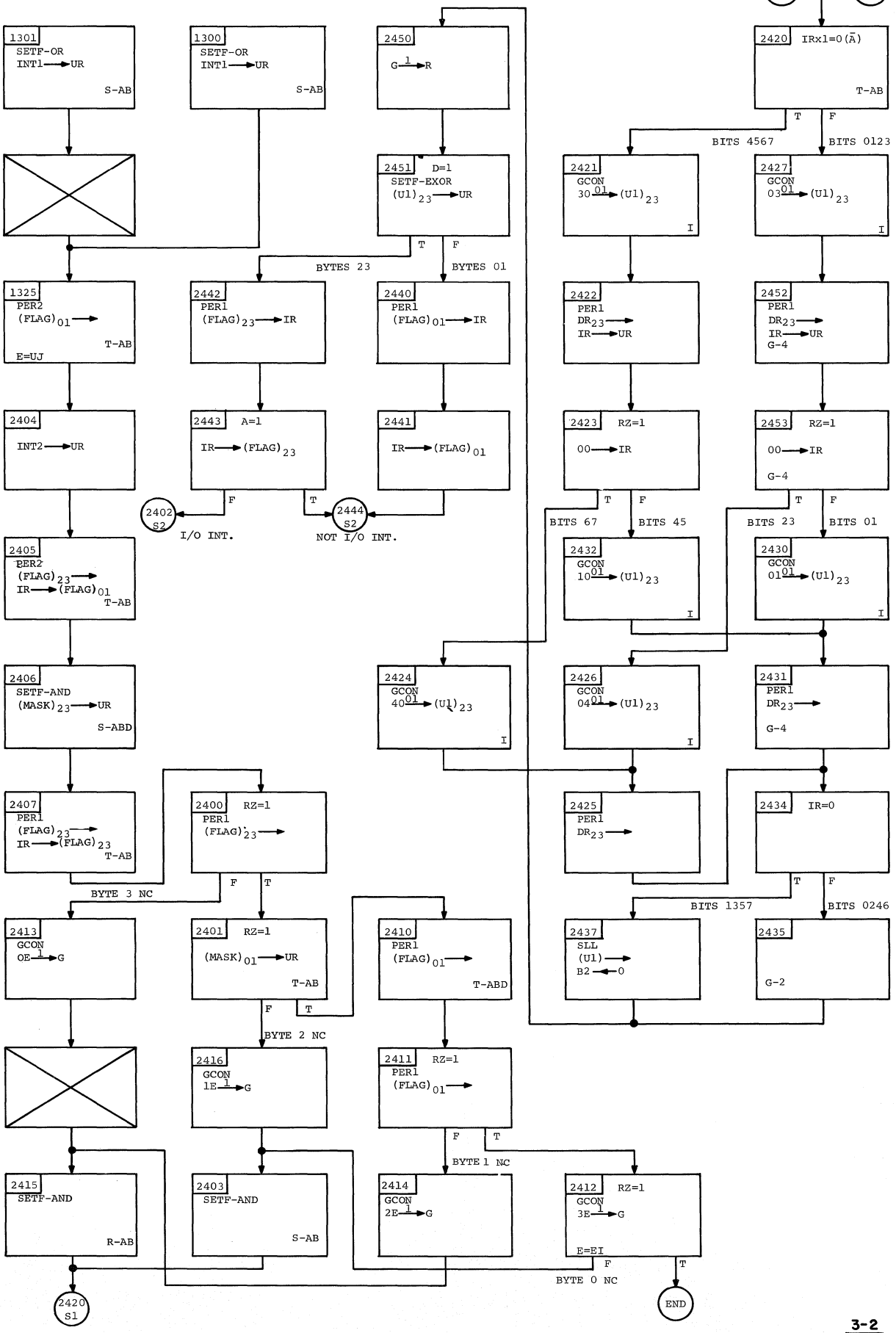


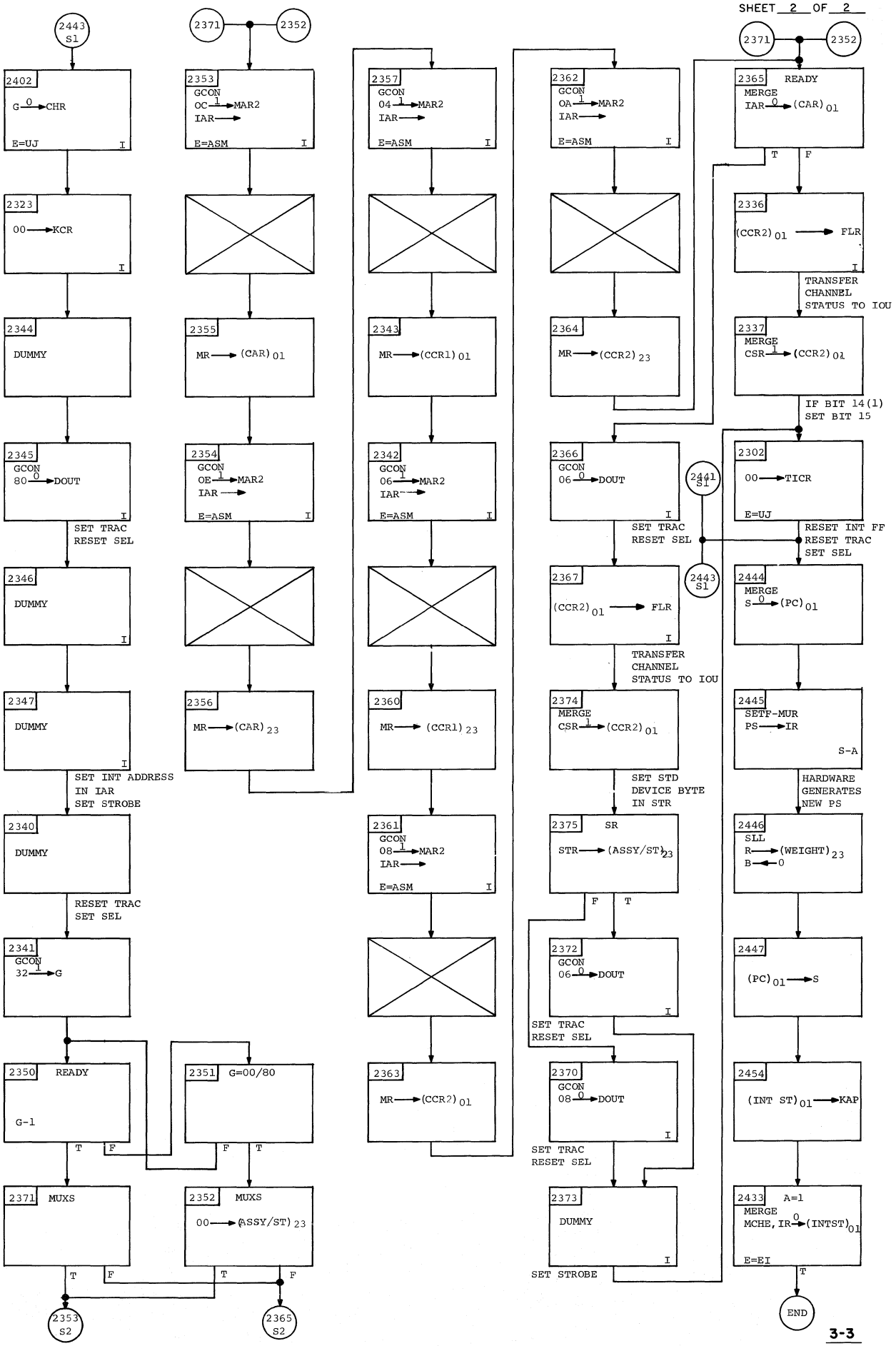
SECTION 3
EO FLOW CHARTS

ELAPSED TIME CLOCK ETC

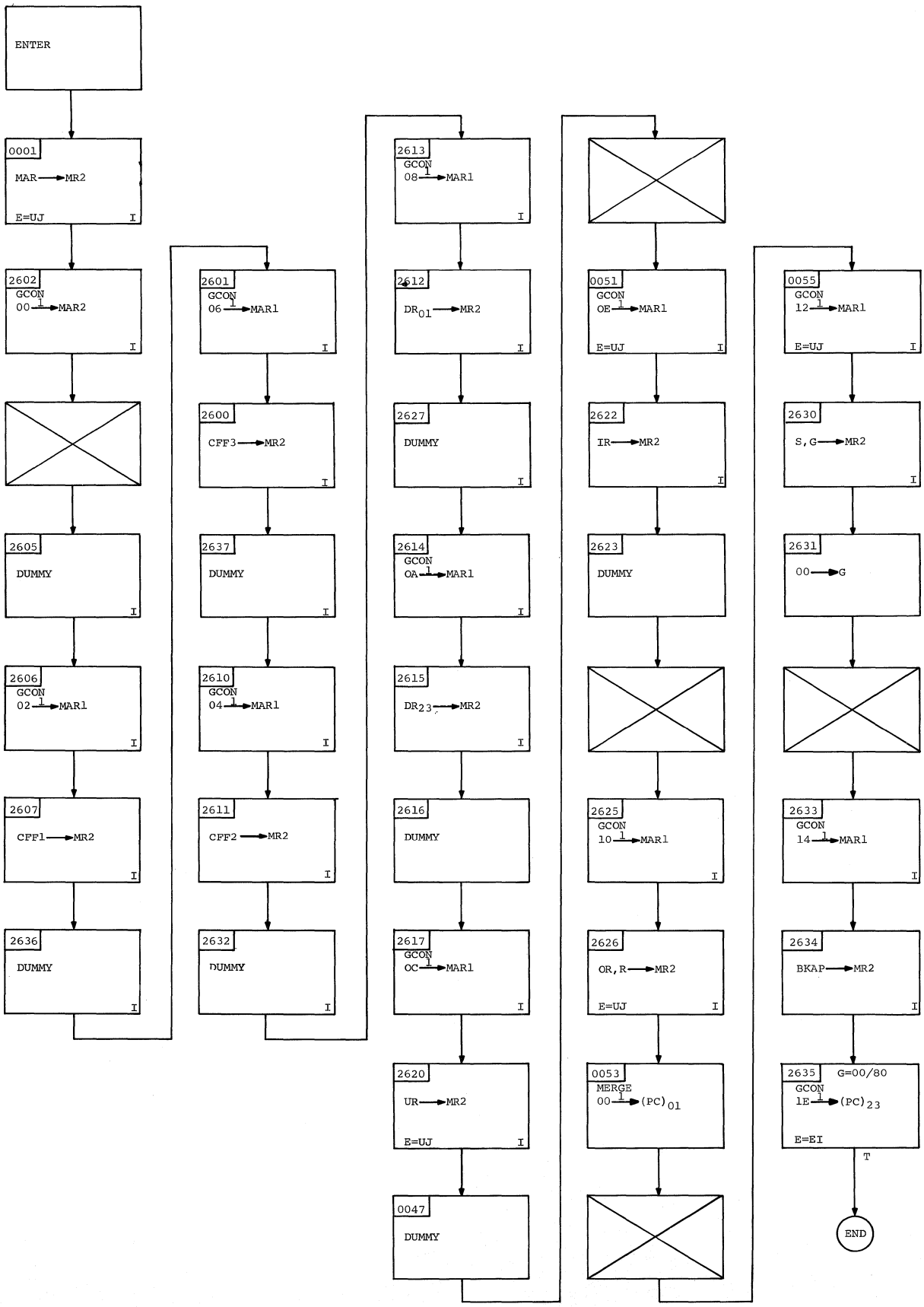


INTERRUPT

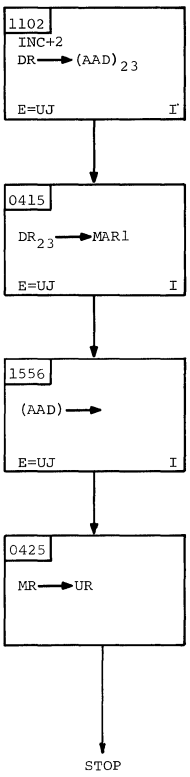

2415 S1
2403 S1
1301 SETF-OR INT1 → UR S-AB
1300 SETF-OR INT1 → UR S-AB
2450 G 1 → R
2420 IRx1=0 (Ā) T-AB
T F
BITS 4567
BITS 0123
2451 D=1 SETF-EXOR (U1)23 → UR
2421 GCON 30 01 → (U1)23 I
2427 GCON 03 01 → (U1)23 I
T F
BYTES 23
BYTES 01
1325 PER2 (FLAG)01 → T-AB E=UJ
2442 PER1 (FLAG)23 → IR
2440 PER1 (FLAG)01 → IR
2422 PER1 DR23 → IR → UR
2452 PER1 DR23 → IR → UR G-4
2404 INT2 → UR
2443 A=1 IR → (FLAG)23
2441 IR → (FLAG)01
2423 RZ=1 00 → IR
2453 RZ=1 00 → IR G-4
F T F T
2402 S2
I/O INT.
2444 S2
NOT I/O INT.
BITS 67
BITS 45
BITS 23
BITS 01
2405 PER2 (FLAG)23 → IR → (FLAG)01 T-AB
2432 GCON 10 01 → (U1)23 I
2430 GCON 01 01 → (U1)23 I
2406 SETF-AND (MASK)23 → UR S-ABD
2424 GCON 40 01 → (U1)23 I
2426 GCON 04 01 → (U1)23 I
2431 PER1 DR23 → G-4
2407 PER1 (FLAG)23 → IR → (FLAG)23 T-AB
2400 RZ=1 PER1 (FLAG)23 →
2425 PER1 DR23 →
2434 IR=0
F T
BYTE 3 NC
T F
BITS 1357
BITS 0246
2413 GCON OE 1 → G
2401 RZ=1 (MASK)01 → UR T-AB
2410 PER1 (FLAG)01 → T-ABD
2437 SLL (U1) → B2 ← 0
2435 G-2
F T
BYTE 2 NC
2416 GCON 1E 1 → G
2411 RZ=1 PER1 (FLAG)01 →
F T
BYTE 1 NC
2415 SETF-AND R-AB
2403 SETF-AND S-AB
2414 GCON 2E 1 → G
2412 RZ=1 GCON 3E 1 → G E=EI
F T
BYTE 0 NC
2420 S1
END


2443 S1
2402
G 0 → CHR
E=UJ I
2323
00 → KCR
I
2344
DUMMY
2345
GCON
80 0 → DOUT
I
SET TRAC
RESET SEL
2346
DUMMY
I
2347
DUMMY
I
SET INT ADDRESS
IN IAR
SET STROBE
2340
DUMMY
RESET TRAC
SET SEL
2341
GCON
32 1 → G
2350 READY
G-1
T F
2351 G=00/80
F T
2371 MUXS
T F
2352 MUXS
00 → (ASSY/ST) 23
T F
2353 S2
2365 S2
2371
2352
2353
GCON
OC 1 → MAR2
IAR →
E=ASM I
2355
MR → (CAR) 01
2354
GCON
OE 1 → MAR2
IAR →
E=ASM I
2356
MR → (CAR) 23
2357
GCON
04 1 → MAR2
IAR →
E=ASM I
2343
MR → (CCR1) 01
2342
GCON
06 1 → MAR2
IAR →
E=ASM I
2360
MR → (CCR1) 23
2361
GCON
08 1 → MAR2
IAR →
E=ASM I
2363
MR → (CCR2) 01
2362
GCON
0A 1 → MAR2
IAR →
E=ASM I
2364
MR → (CCR2) 23
2366
GCON
06 0 → DOUT
I
SET TRAC
RESET SEL
2367
(CCR2) 01 → FLR
I
TRANSFER
CHANNEL
STATUS TO IOU
2374
MERGE
CSR 1 → (CCR2) 01
SET STD
DEVICE BYTE
IN STR
2375 SR
STR → (ASSY/ST) 23
F T
2372
GCON
06 0 → DOUT
I
SET TRAC
RESET SEL
2370
GCON
08 0 → DOUT
I
SET TRAC
RESET SEL
2373
DUMMY
I
SET STROBE
2371
2352
2365 READY
MERGE
IAR 0 → (CAR) 01
T F
2336
(CCR2) 01 → FLR
I
TRANSFER
CHANNEL
STATUS TO IOU
2337
MERGE
CSR 1 → (CCR2) 01
IF BIT 14(1)
SET BIT 15
2441 S1
2302
00 → TICR
E=UJ
RESET INT FF
RESET TRAC
SET SEL
2443 S1
2444
MERGE
S 0 → (PC) 01
2445
SETF-MUR
PS → IR
S-A
HARDWARE
GENERATES
NEW PS
2446
SLL
R → (WEIGHT) 23
B ← 0
2447
(PC) 01 → S
2454
(INT ST) 01 → KAP
2433 A=1
MERGE
MCHE, IR 0 → (INTST) 01
E=EI
T
END

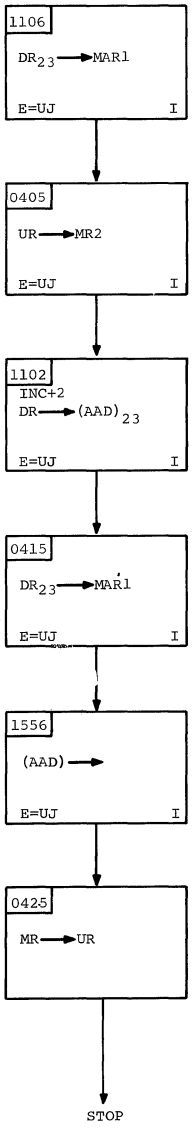
SNAPSHOT SNS



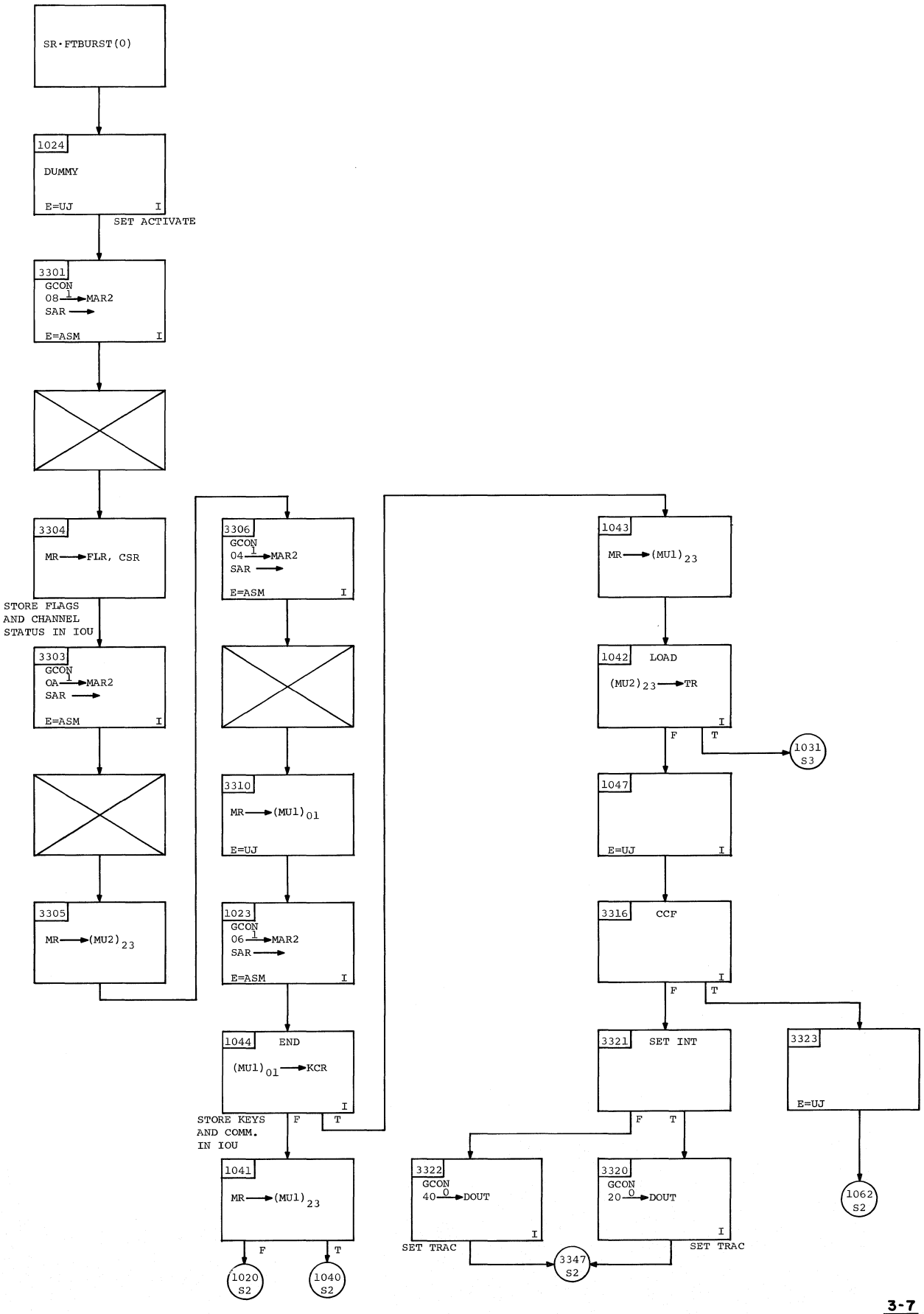
READ HIGH SPEED MEMORY RDM



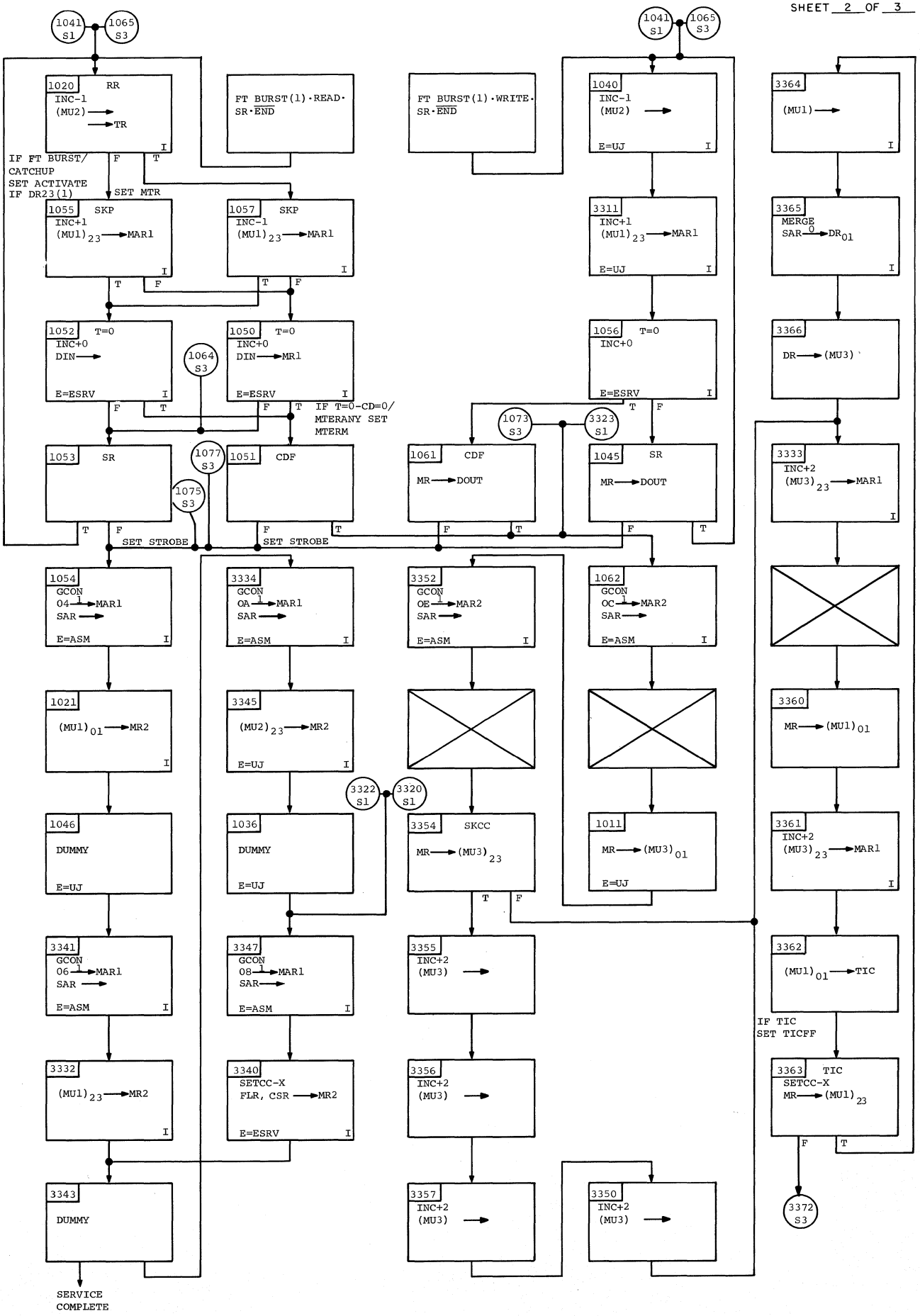
WRITE HIGH SPEED MEMORY WRM



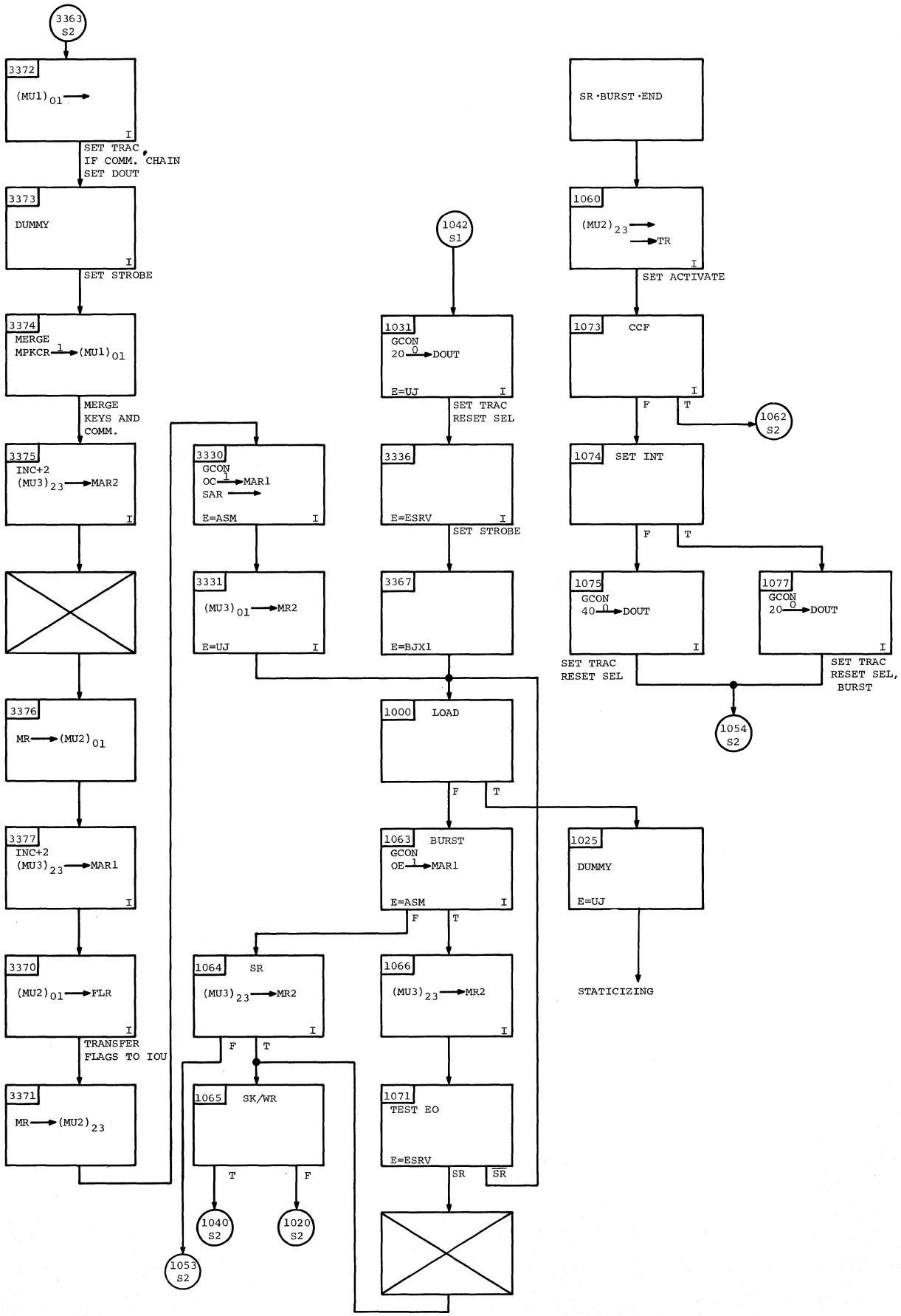
MULTIPLEXOR SERVICE MUXS



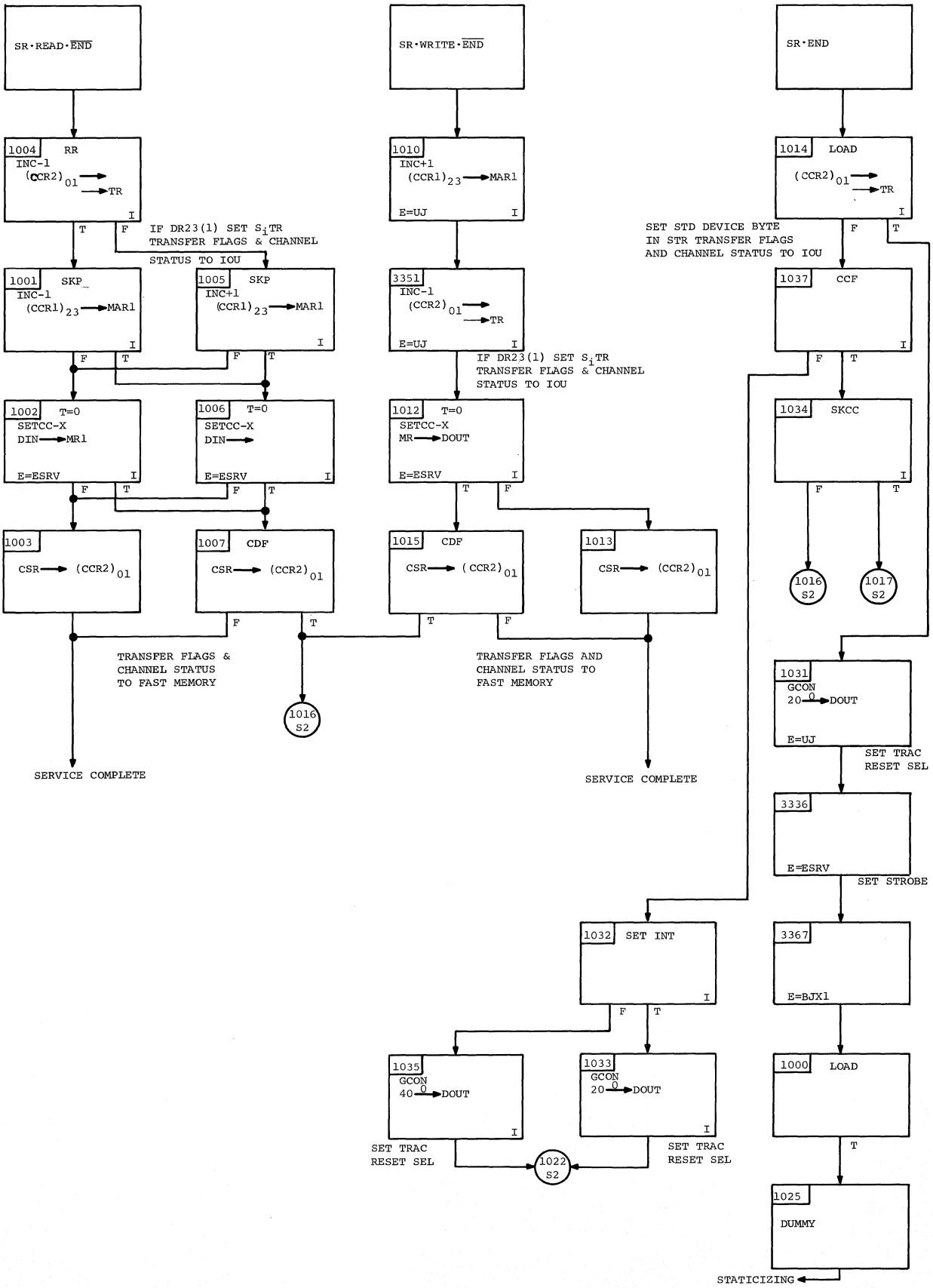
MULTIPLEXOR SERVICE MUXS



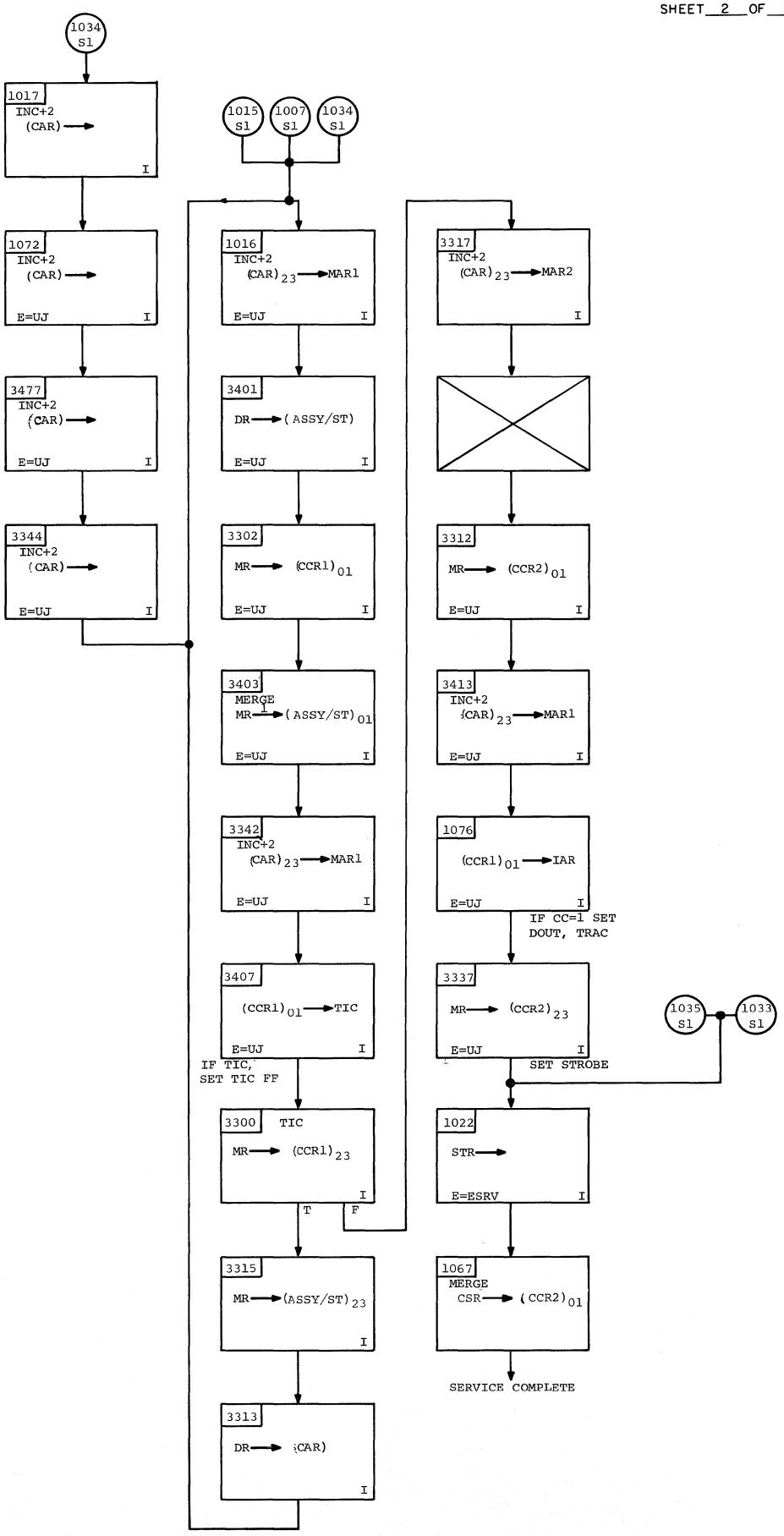
MULTIPLEXOR SERVICE MUXS



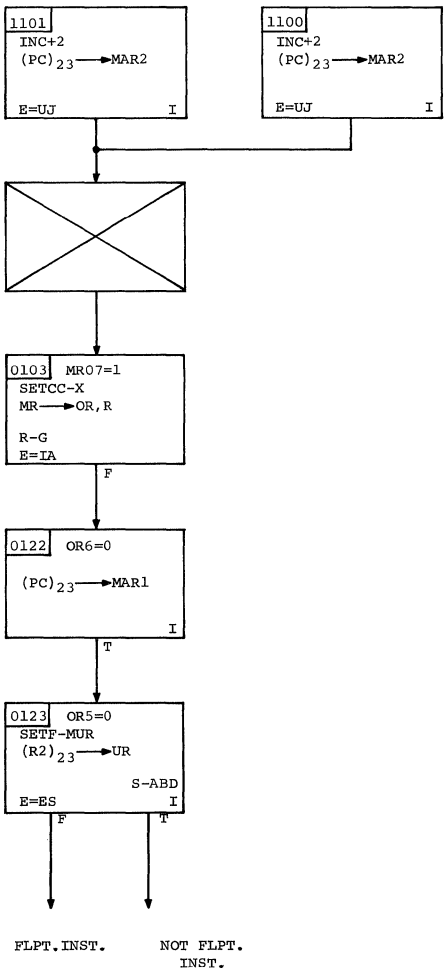
SELECTOR SERVICE SELS



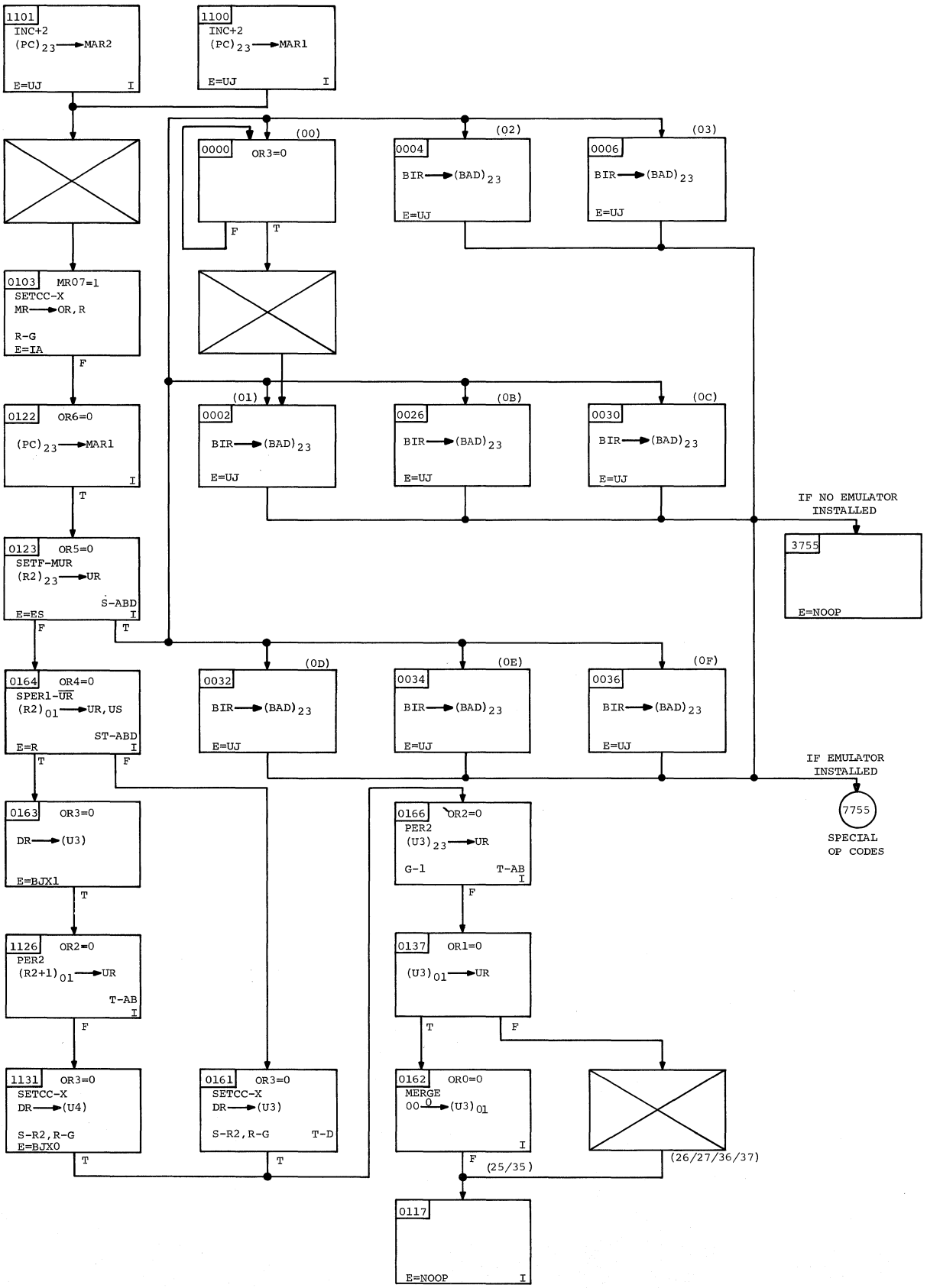
SELECTOR SERVICE SELS



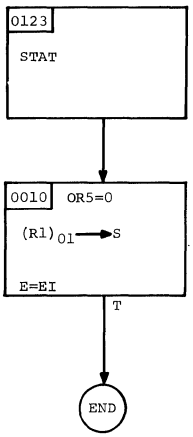
STATICIZING RR FORMAT



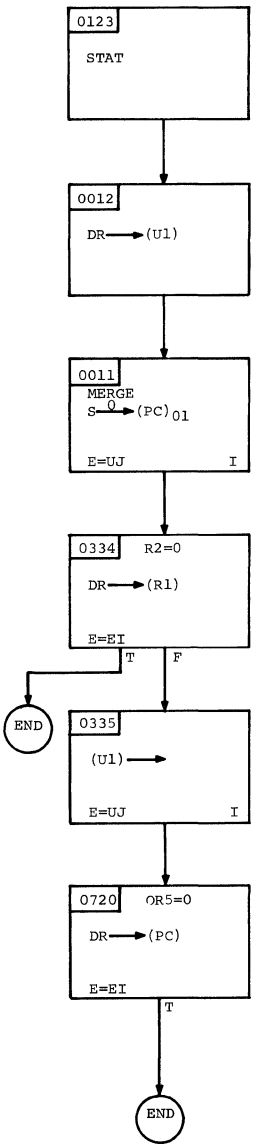
NO OPERATION INSTRUCTIONS RR FORMAT



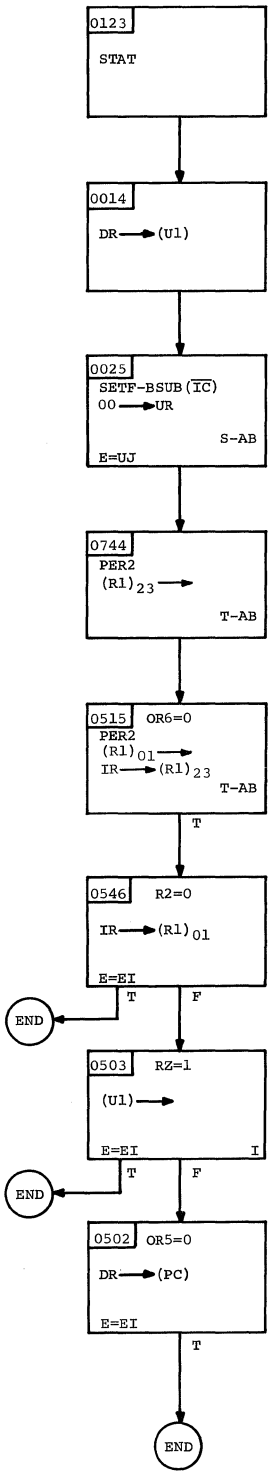
SET PROGRAM MASK SPM RR 04



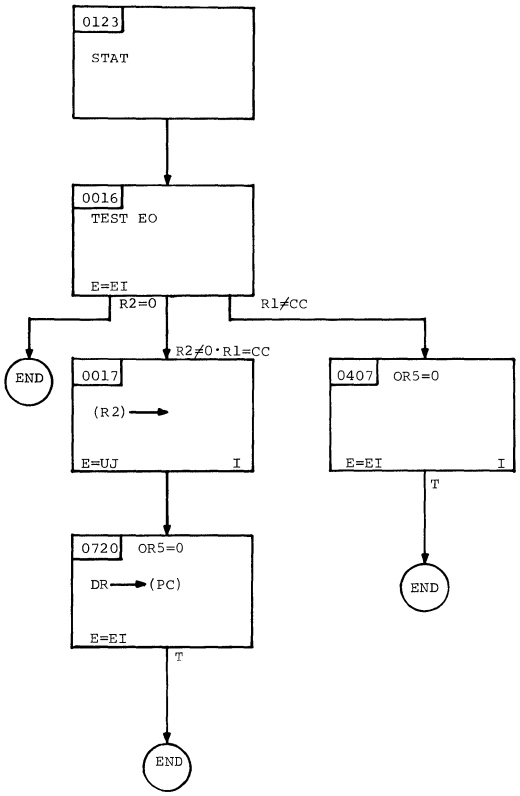
BRANCH AND LINK BALR RR 05



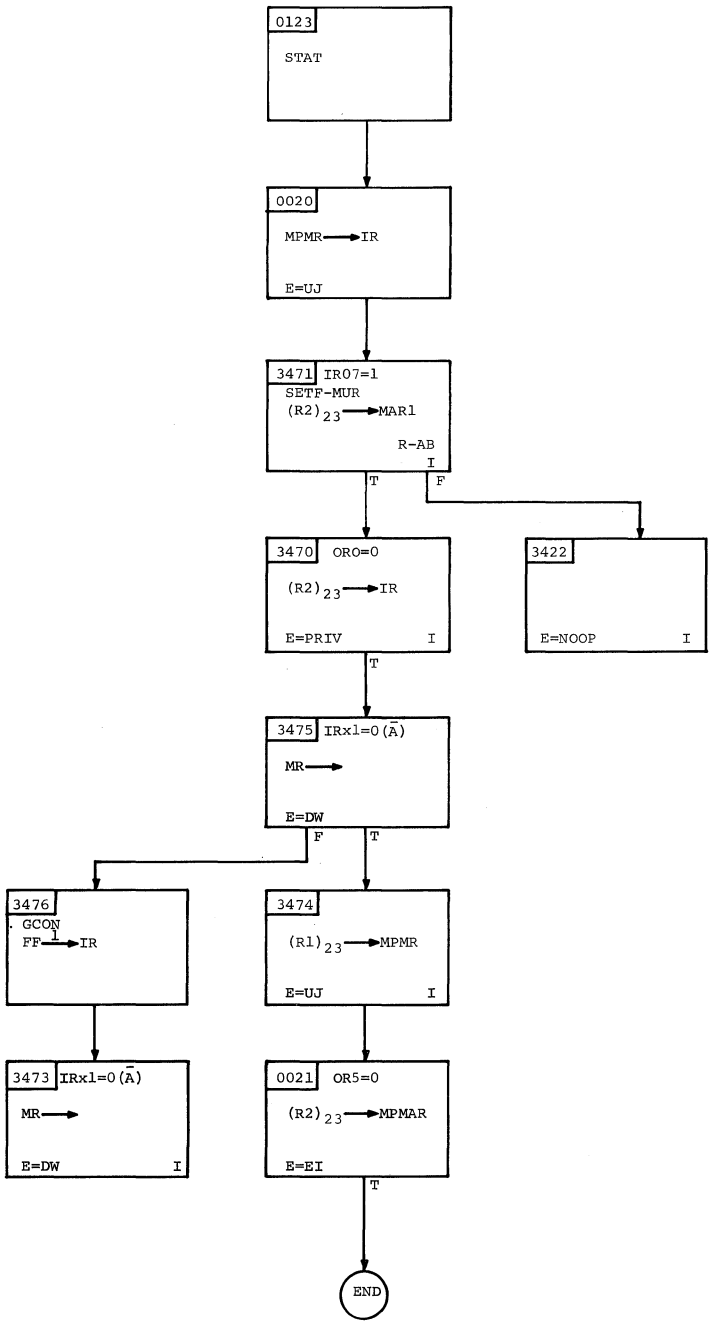
BRANCH ON COUNT BCTR RR 06



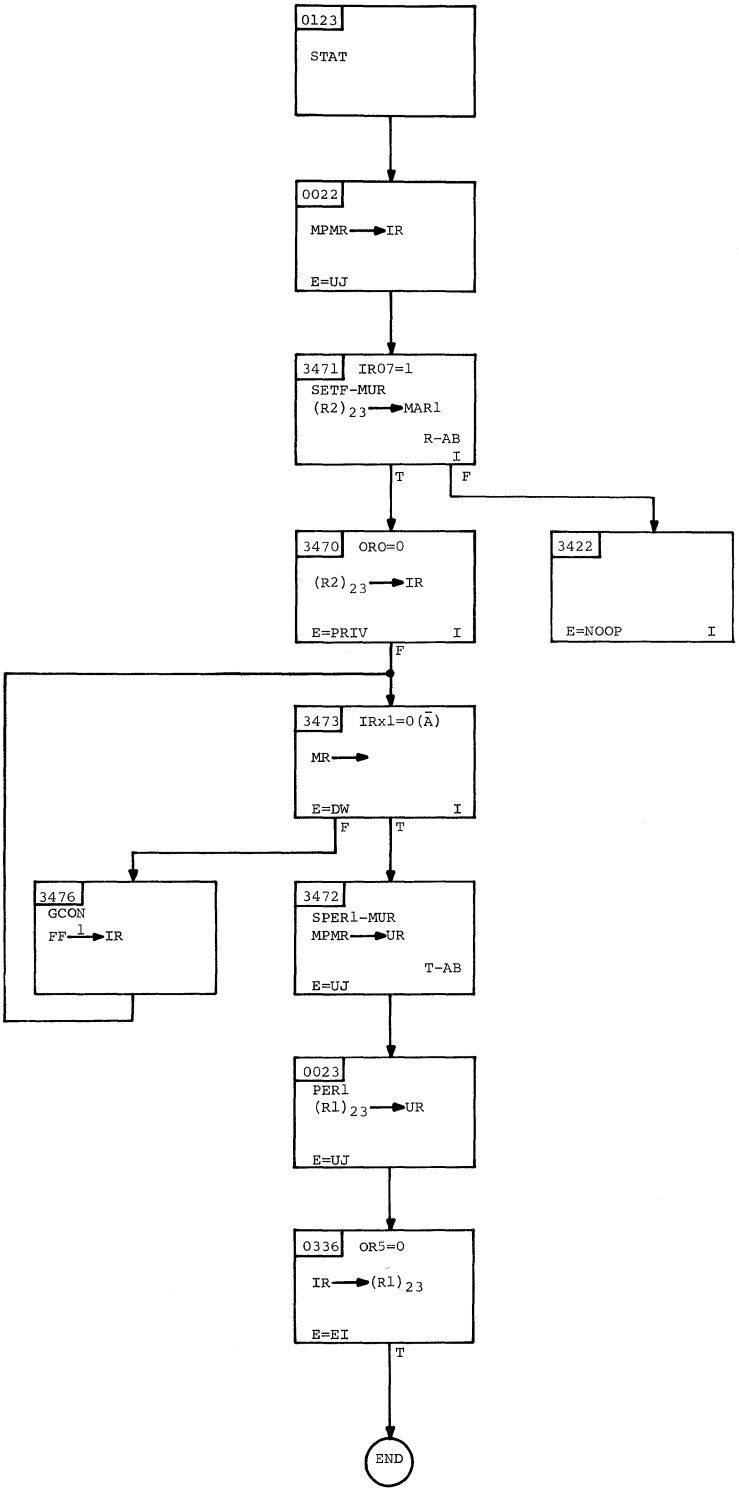
BRANCH ON CONDITION BCR RR 07



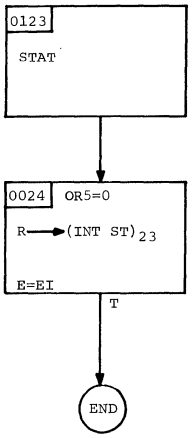
SET STORAGE KEY SSK RR 08



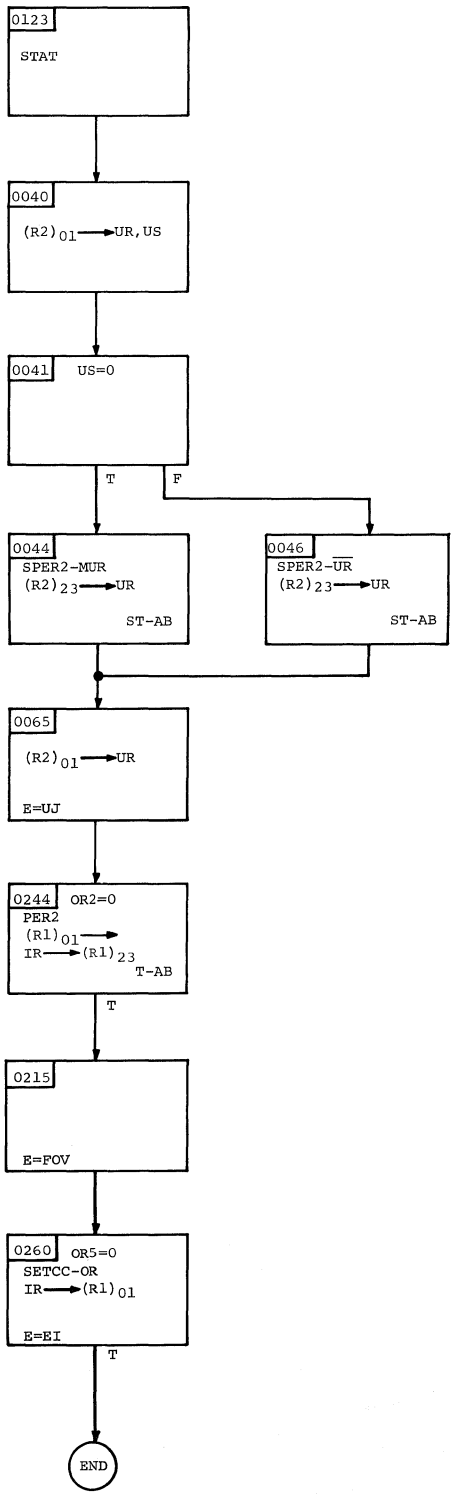
INSERT STORAGE KEY ISK RR 09



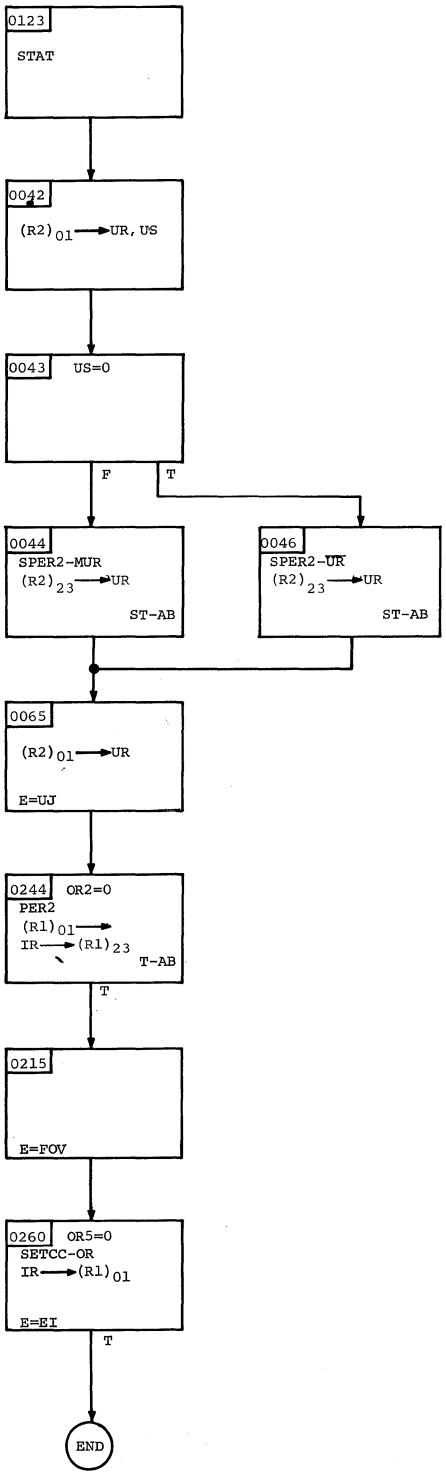
SUPERVISOR CALL SVC RR 0A



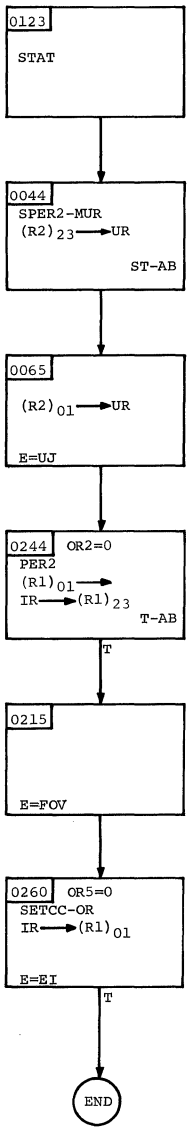
LOAD POSITIVE LPR RR 10



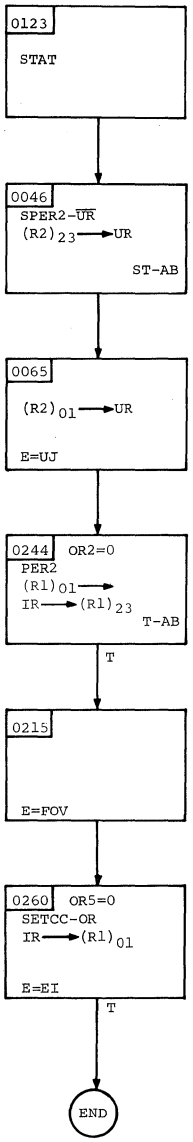
LOAD NEGATIVE LNR RR II



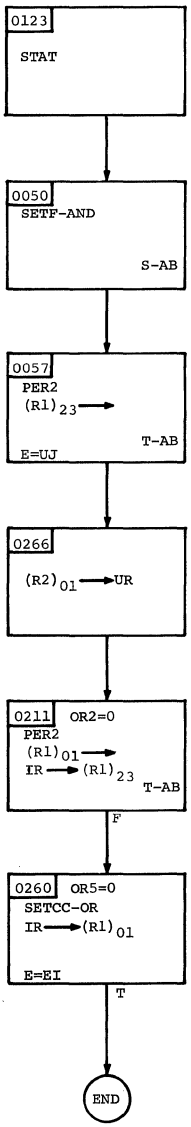
LOAD AND TEST LTR RR 12



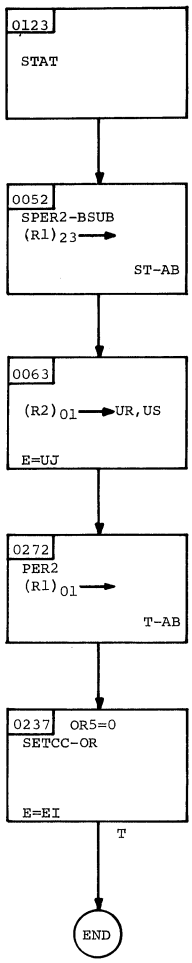
LOAD & COMPLEMENT LCR RR 13



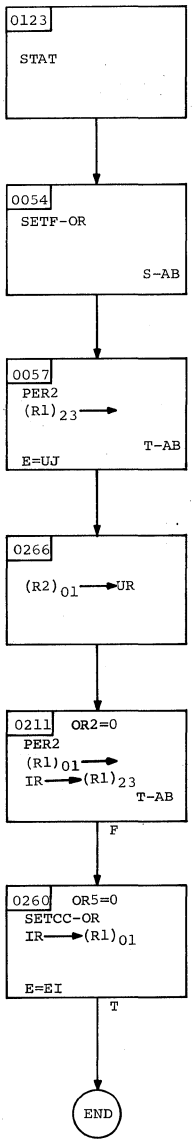
LOGICAL AND NR RR 14



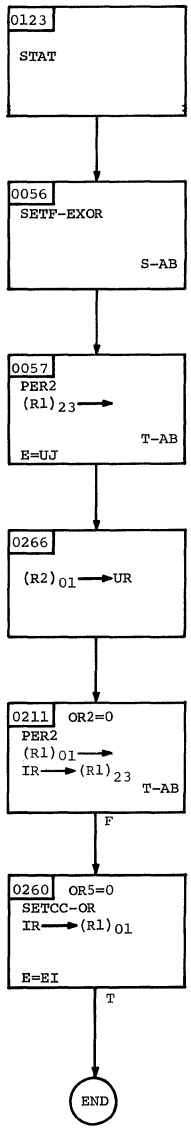
COMPARE LOGICAL CLR RR 15



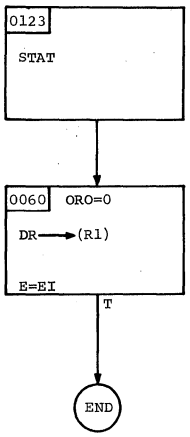
LOGICAL OR OR RR 16



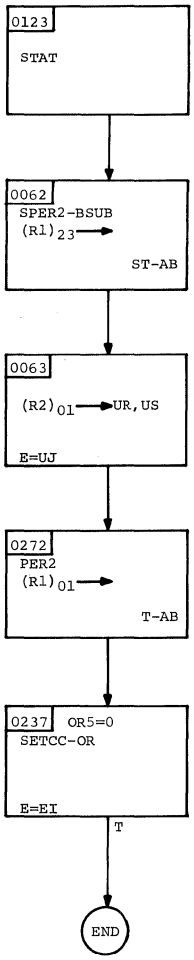
LOGICAL EXCLUSIVE OR XR RR 17



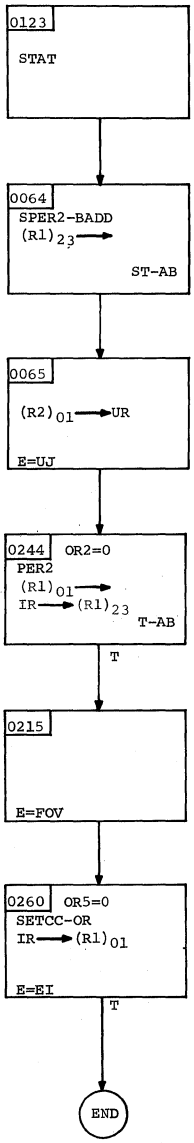
LOAD LR RR 18



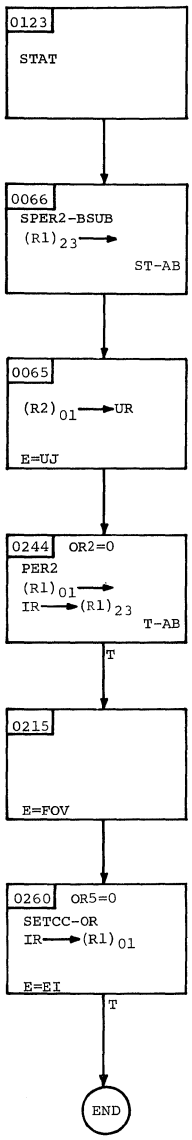
COMPARE CR RR 19



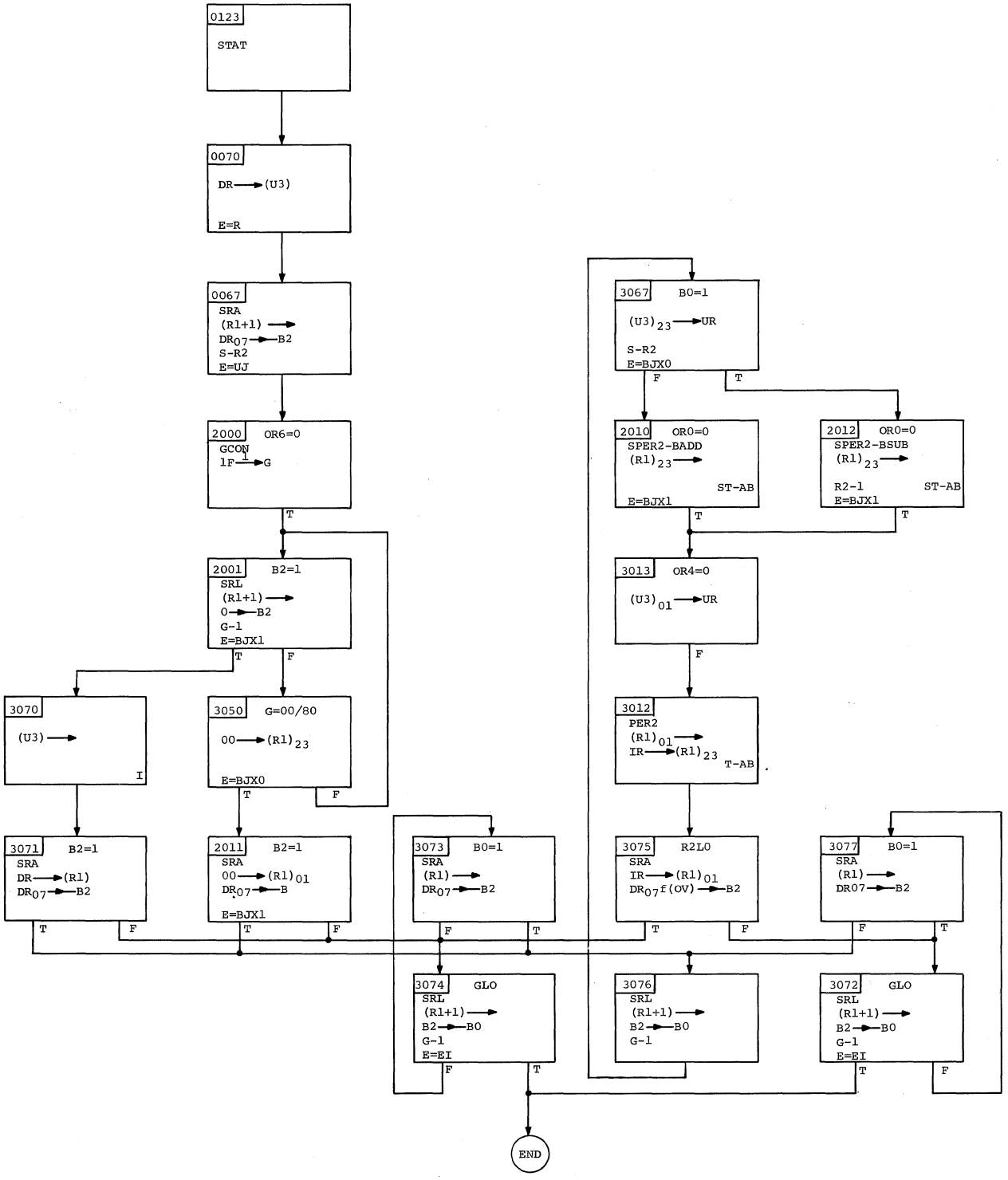
ADD AR RR IA


0123
STAT
0064
SPER2-BADD
(R1)23 →
ST-AB
0065
(R2)01 → UR
E=UJ
0244
OR2=0
PER2
(R1)01 →
IR → (R1)23
T-AB
T
0215
E=FOV
0260
OR5=0
SETCC-OR
IR → (R1)01
E=EI
T
END

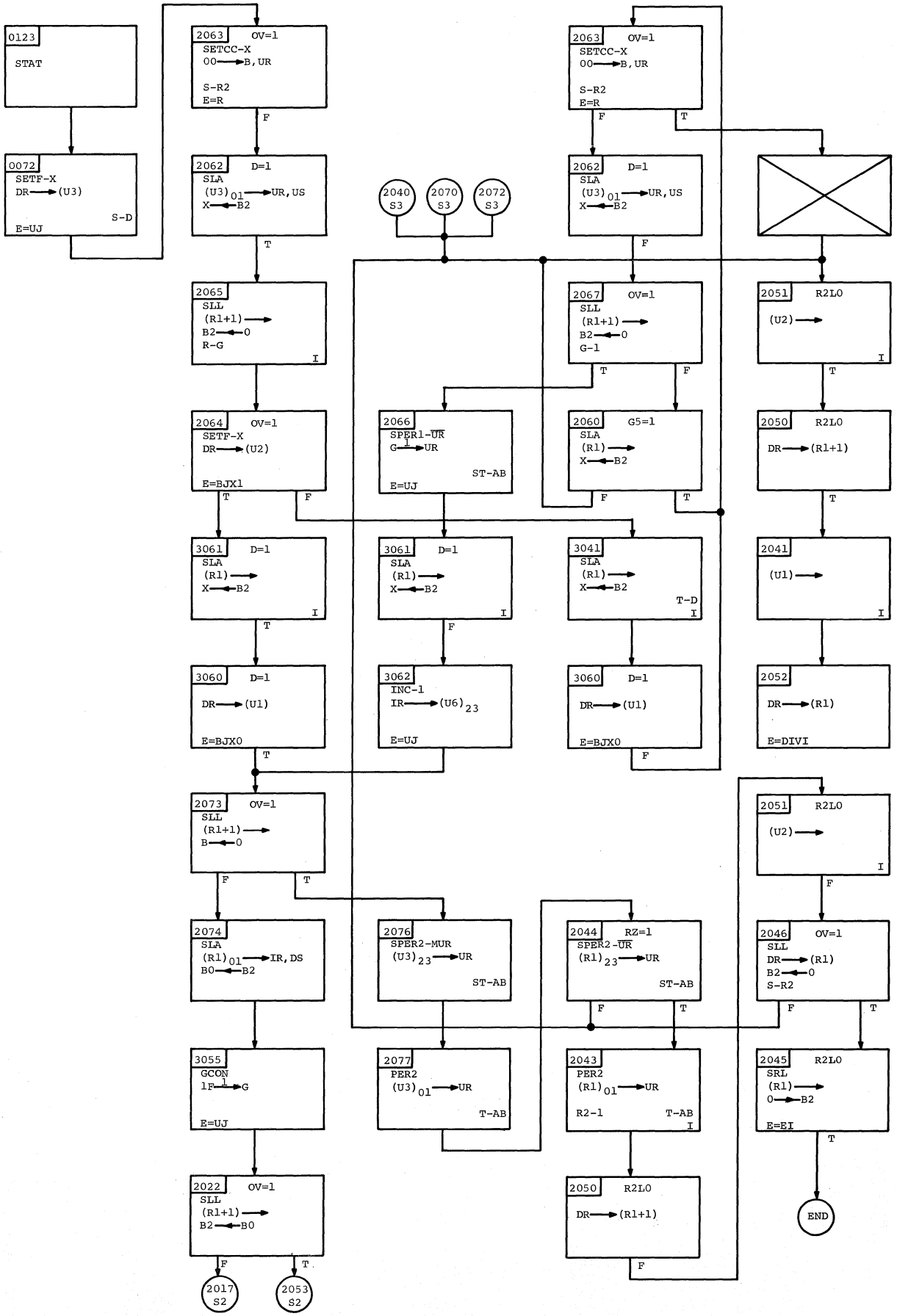
SUBTRACT SR RR IB

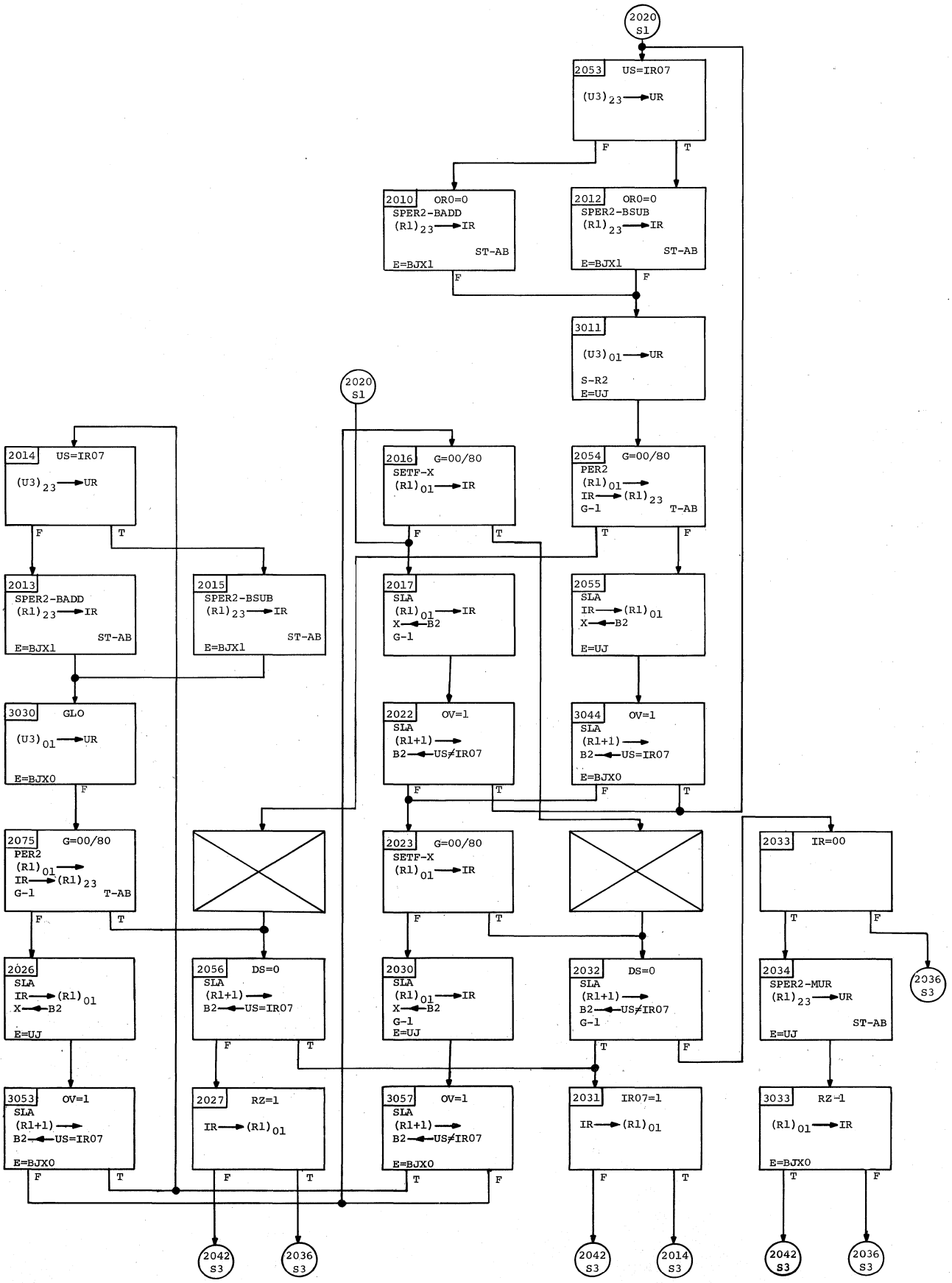


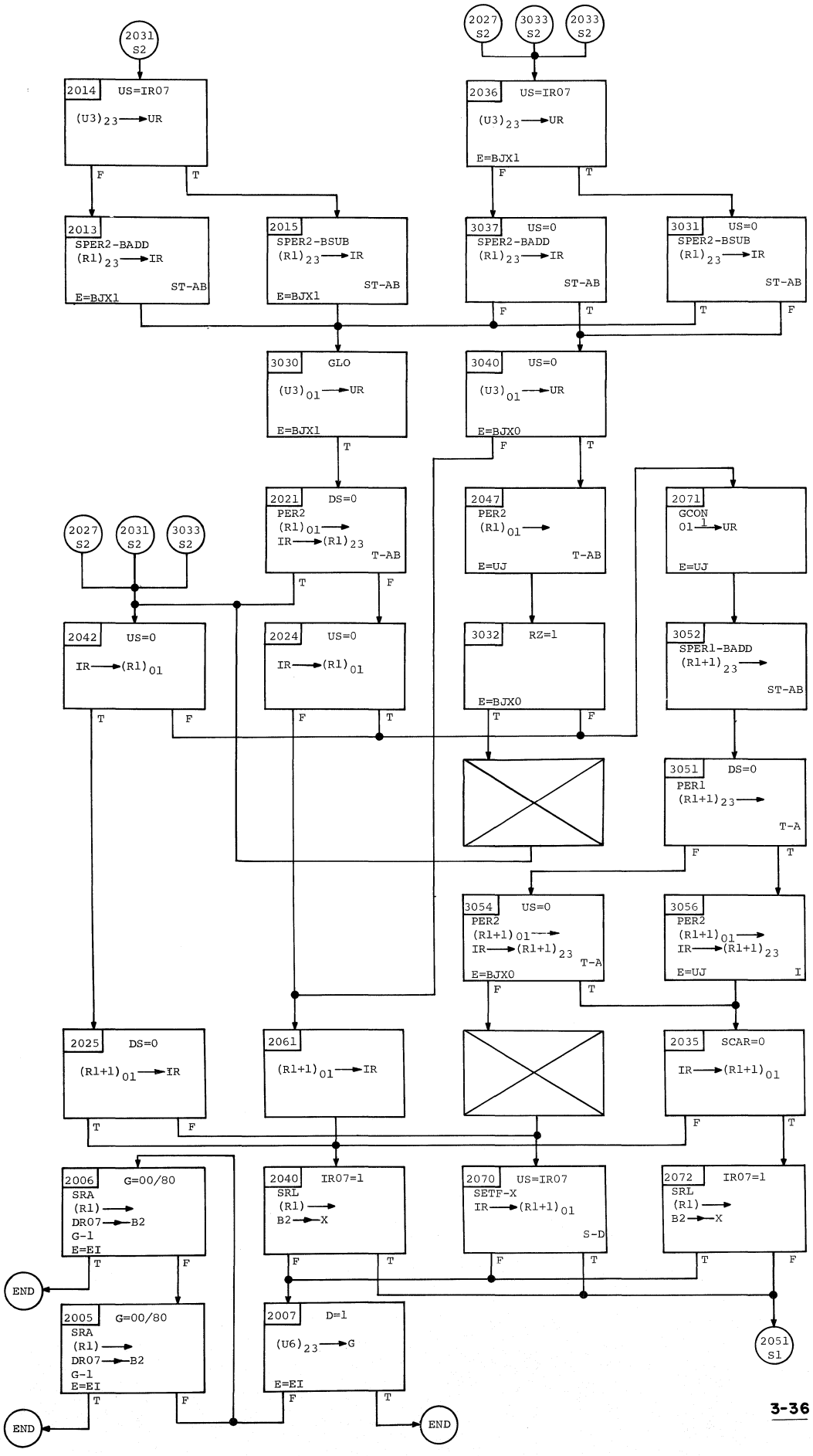
MULTIPLY MR RR IC



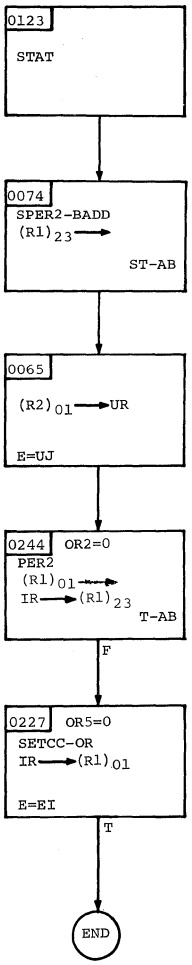
DIVIDE DR RR ID



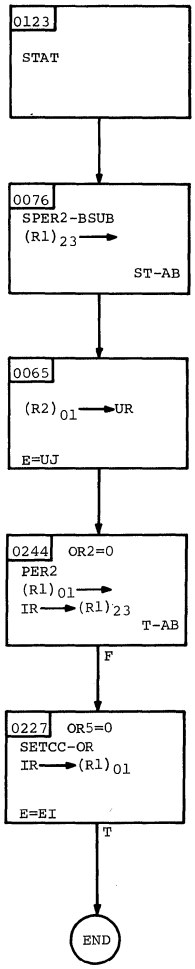

2020 S1
2053 US=IR07 (U3)23→UR
F T
2010 OR0=0 SPER2-BADD (R1)23→IR ST-AB E=BJX1
2012 OR0=0 SPER2-BSUB (R1)23→IR ST-AB E=BJX1
F F
3011 (U3)01→UR S-R2 E=UJ
2020 S1
2014 US=IR07 (U3)23→UR
2016 G=00/80 SETF-X (R1)01→IR
2054 G=00/80 PER2 (R1)01→ IR→(R1)23 G-1 T-AB
F T T F
2013 SPER2-BADD (R1)23→IR ST-AB E=BJX1
2015 SPER2-BSUB (R1)23→IR ST-AB E=BJX1
2017 SLA (R1)01→IR X←B2 G-1
2055 SLA IR→(R1)01 X←B2 E=UJ
3030 GLO (U3)01→UR E=BJX0
2022 OV=1 SLA (R1+1)→ B2←US≠IR07
3044 OV=1 SLA (R1+1)→ B2←US=IR07 E=BJX0
F T F T
2075 G=00/80 PER2 (R1)01→ IR→(R1)23 G-1 T-AB
2023 G=00/80 SETF-X (R1)01→IR
2033 IR=00
F T F T T F
2026 SLA IR→(R1)01 X←B2 E=UJ
2056 DS=0 SLA (R1+1)→ B2←US=IR07
2030 SLA (R1)01→IR X←B2 G-1 E=UJ
2032 DS=0 SLA (R1+1)→ B2←US≠IR07 G-1
2034 SPER2-MUR (R1)23→UR ST-AB E=UJ
2036 S3
3053 OV=1 SLA (R1+1)→ B2←US=IR07 E=BJX0
2027 RZ=1 IR→(R1)01
3057 OV=1 SLA (R1+1)→ B2←US≠IR07 E=BJX0
2031 IR07=1 IR→(R1)01
3033 RZ=1 (R1)01→IR E=BJX0
F T F T F T F T T F
2042 S3
2036 S3
2042 S3
2014 S3
2042 S3
2036 S3


2031 S2
2014 US=IR07
(U3)23 → UR
F
T
2013
SPER2-BADD
(R1)23 → IR
ST-AB
E=BJX1
2015
SPER2-BSUB
(R1)23 → IR
ST-AB
E=BJX1
2027 S2
3033 S2
2033 S2
2036 US=IR07
(U3)23 → UR
E=BJX1
3037 US=0
SPER2-BADD
(R1)23 → IR
ST-AB
3031 US=0
SPER2-BSUB
(R1)23 → IR
ST-AB
3030 GLO
(U3)01 → UR
E=BJX1
3040 US=0
(U3)01 → UR
E=BJX0
2021 DS=0
PER2
(R1)01 →
IR → (R1)23
T-AB
2047
PER2
(R1)01 →
T-AB
E=UJ
2071
GCON
01 1 → UR
E=UJ
2027 S2
2031 S2
3033 S2
2042 US=0
IR → (R1)01
2024 US=0
IR → (R1)01
3032 RZ=1
E=BJX0
3052
SPER1-BADD
(R1+1)23 →
ST-AB
3051 DS=0
PER1
(R1+1)23 →
T-A
3054 US=0
PER2
(R1+1)01 →
IR → (R1+1)23
T-A
E=BJX0
3056
PER2
(R1+1)01 →
IR → (R1+1)23
E=UJ
I
2025 DS=0
(R1+1)01 → IR
2061
(R1+1)01 → IR
2035 SCAR=0
IR → (R1+1)01
2006 G=00/80
SRA
(R1) →
DR07 → B2
G-1
E=EI
2040 IR07=1
SRL
(R1) →
B2 → X
2070 US=IR07
SETF-X
IR → (R1+1)01
S-D
2072 IR07=1
SRL
(R1) →
B2 → X
END
2005 G=00/80
SRA
(R1) →
DR07 → B2
G-1
E=EI
2007 D=1
(U6)23 → G
E=EI
END
END
2051 S1

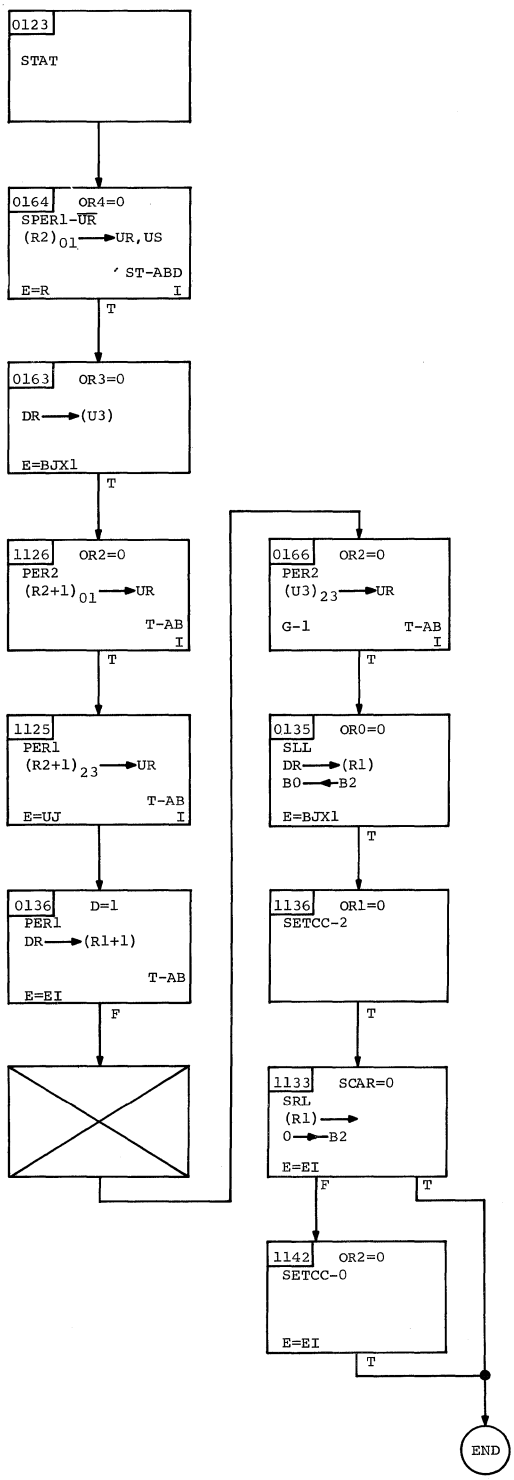
ADD LOGICAL ALR RR IE



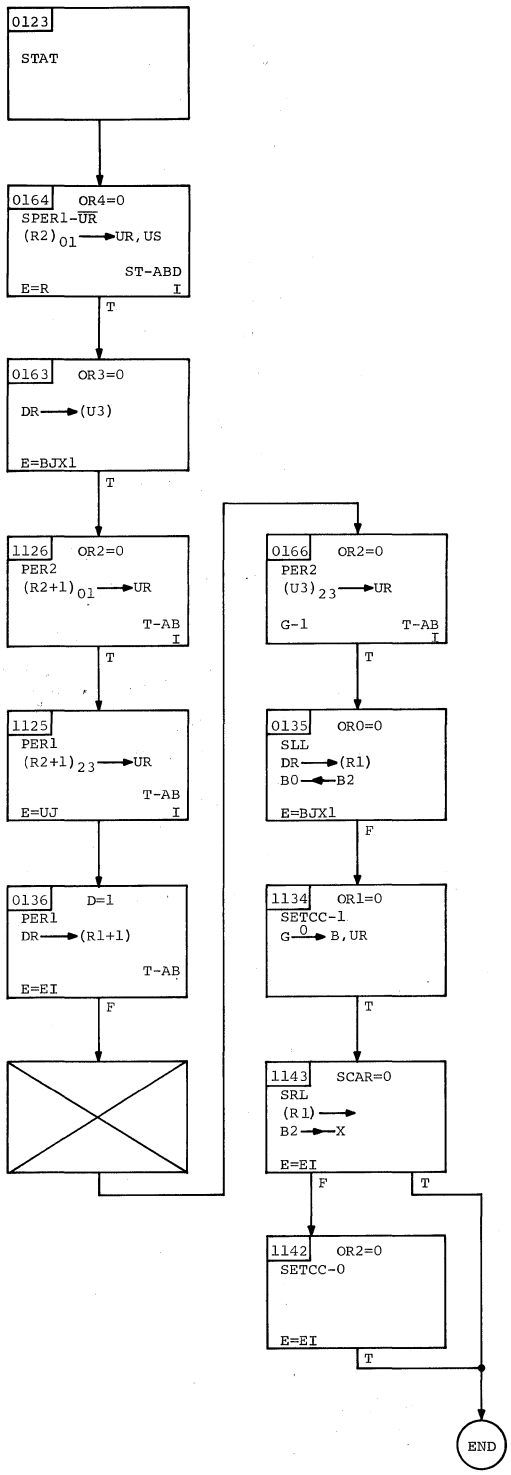
SUBTRACT LOGICAL SLR RR IF



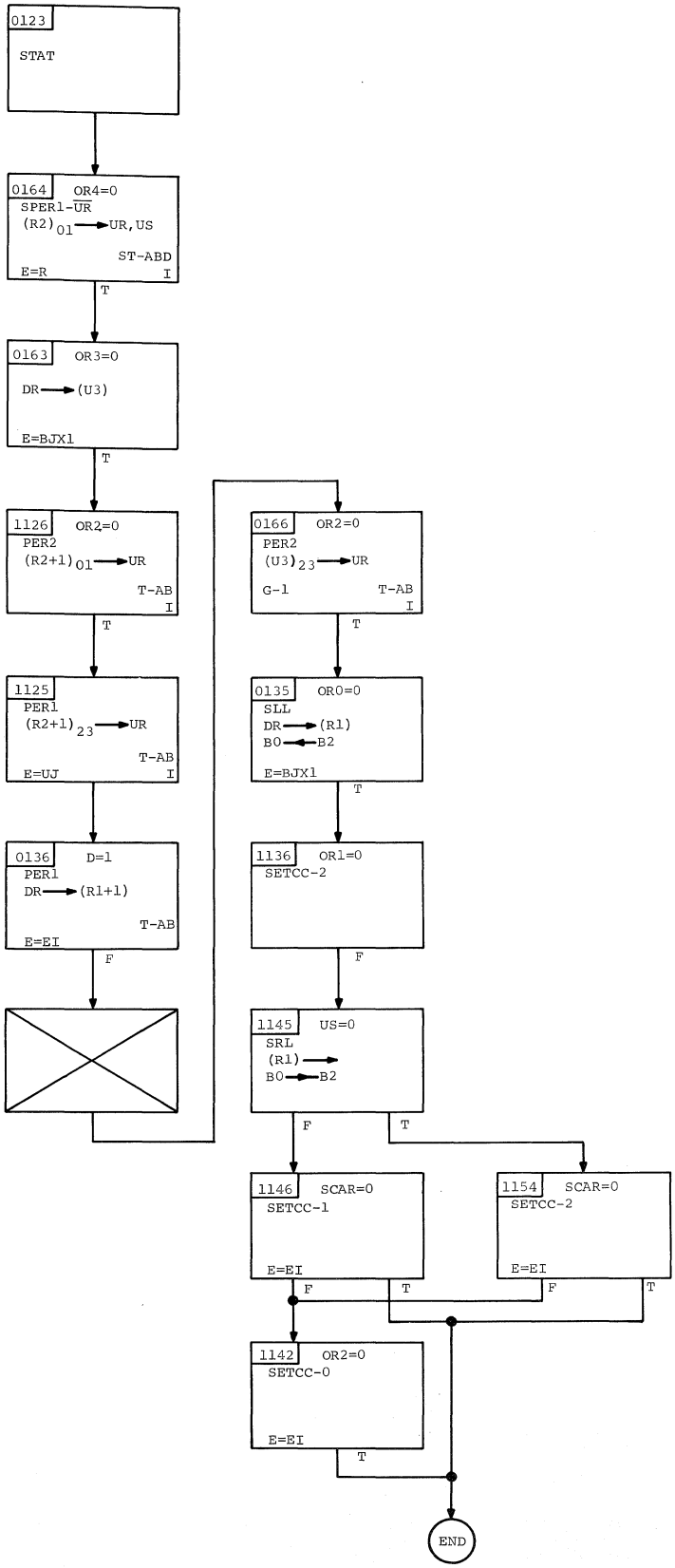
LOAD POSITIVE LONG LPDR RR 20



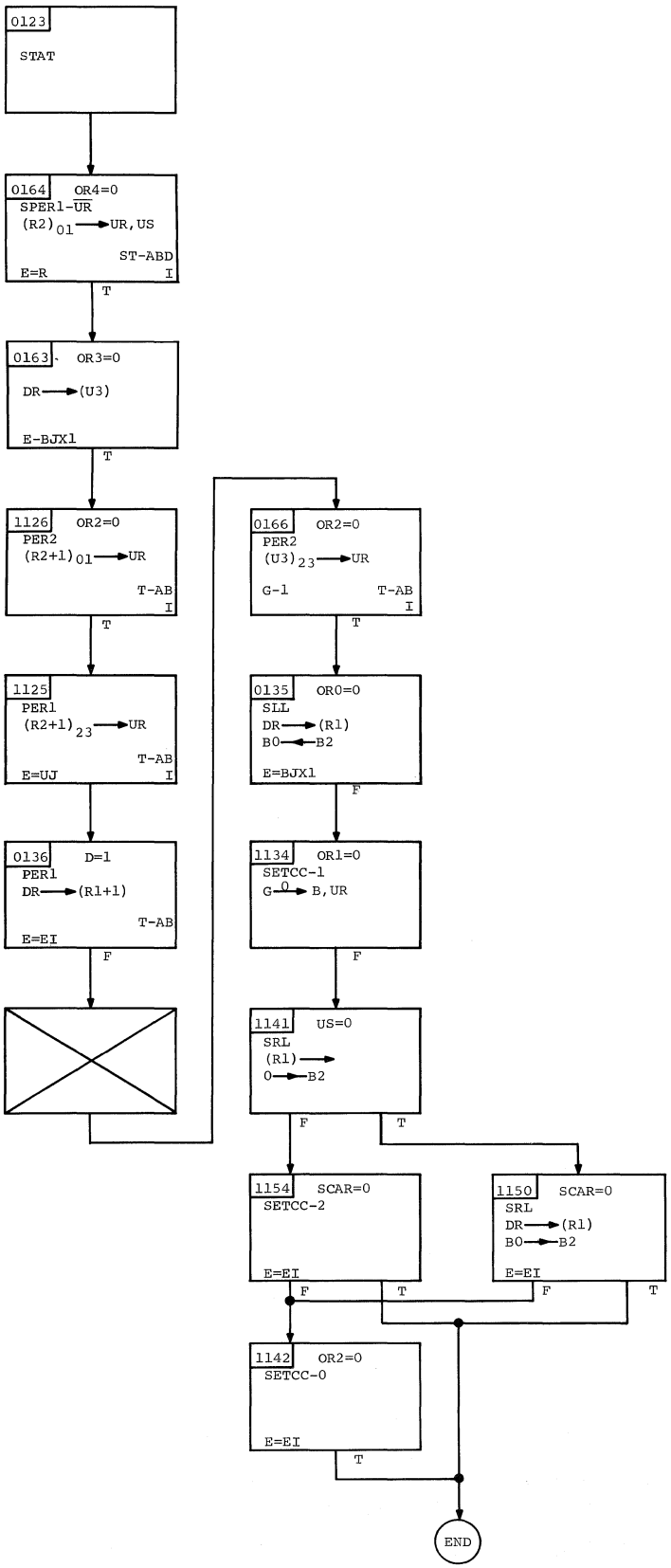
LOAD NEGATIVE LONG LNDR RR 21



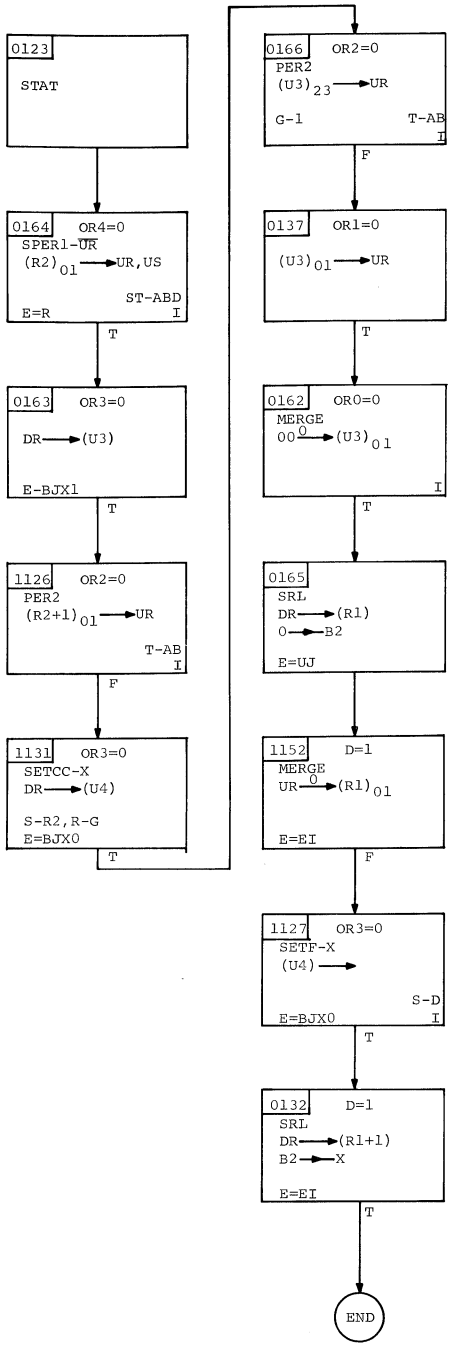
LOAD AND TEST LONG LTDR RR 22



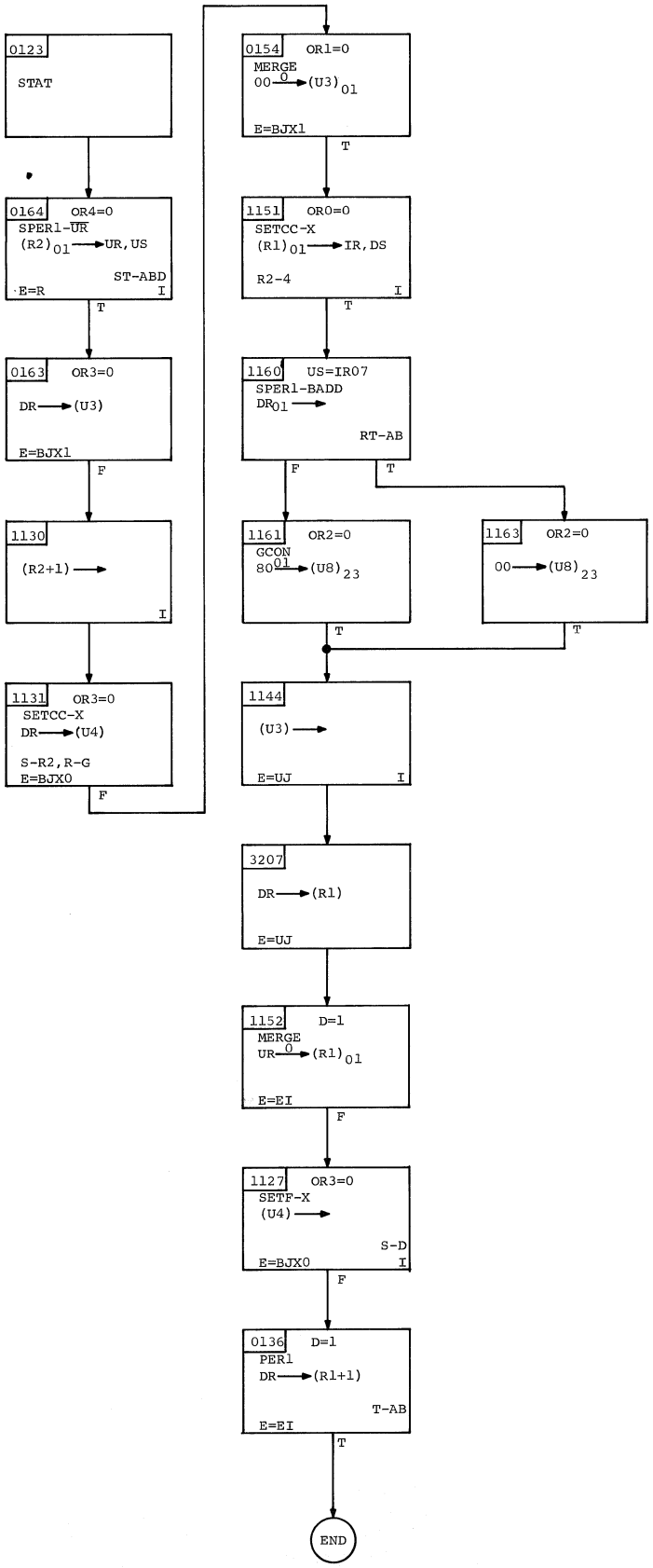
LOAD COMPLEMENT LONG LCDR RR 23



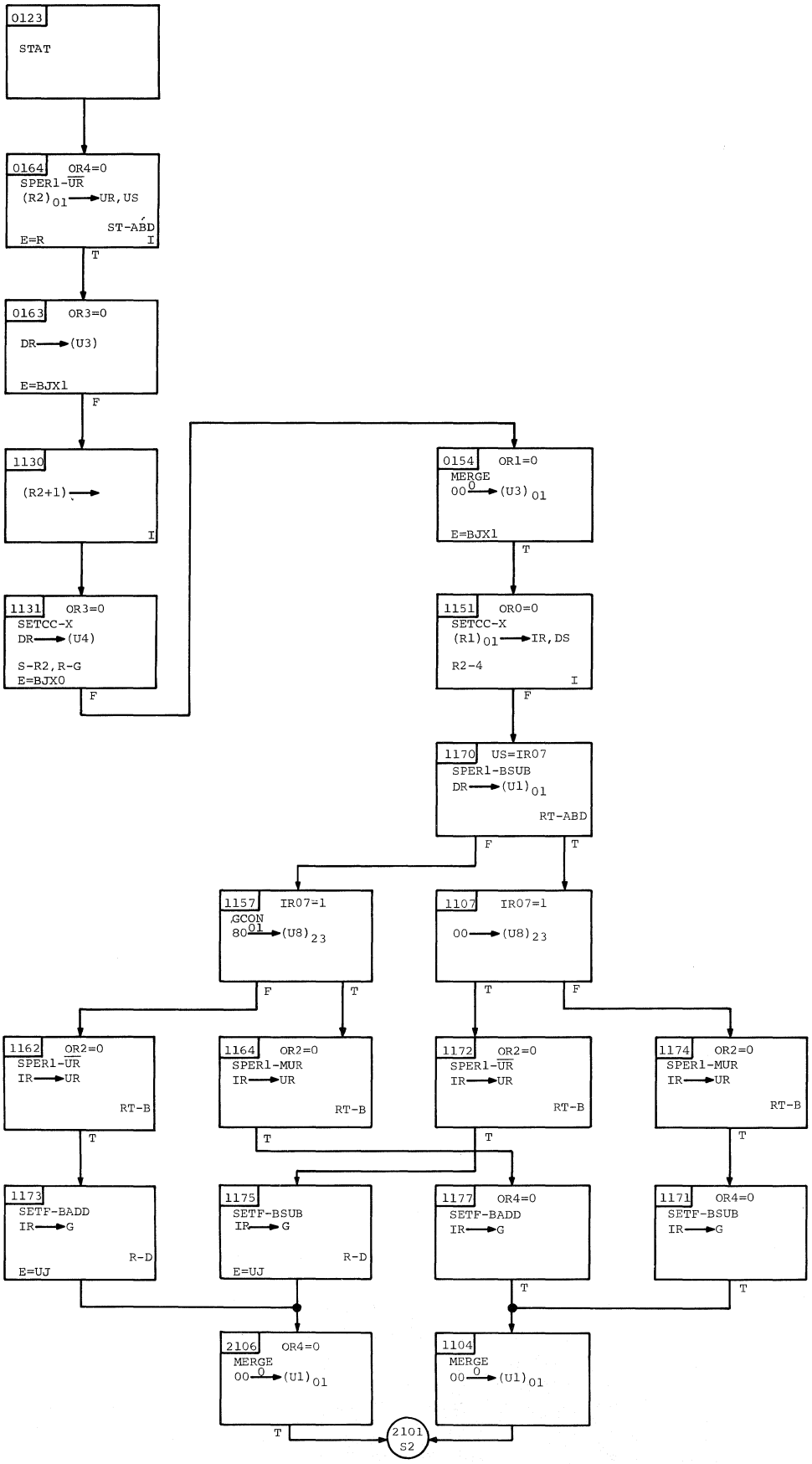
HALVE LONG HDR RR 24



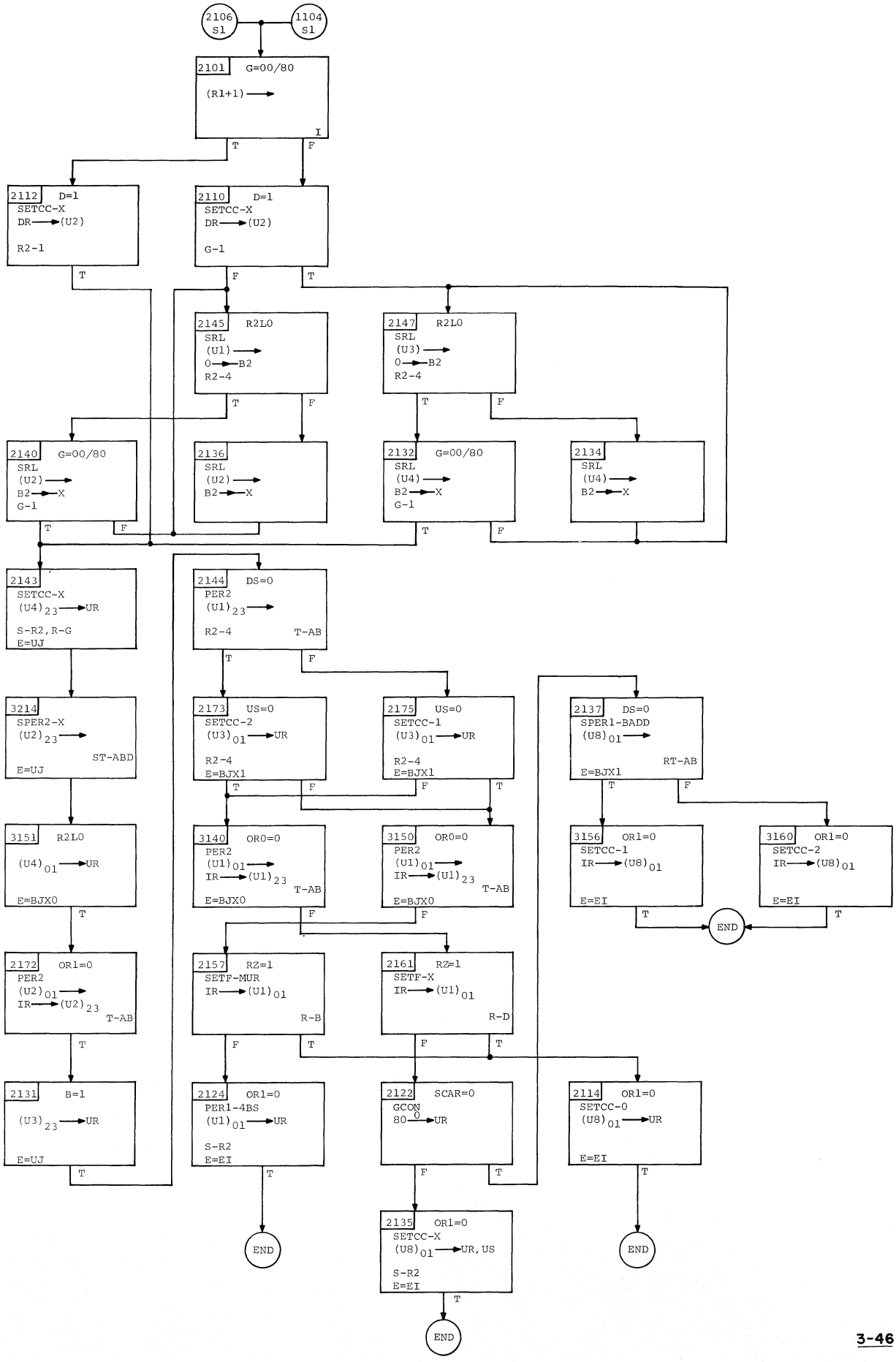
LOAD LONG LDR RR 28



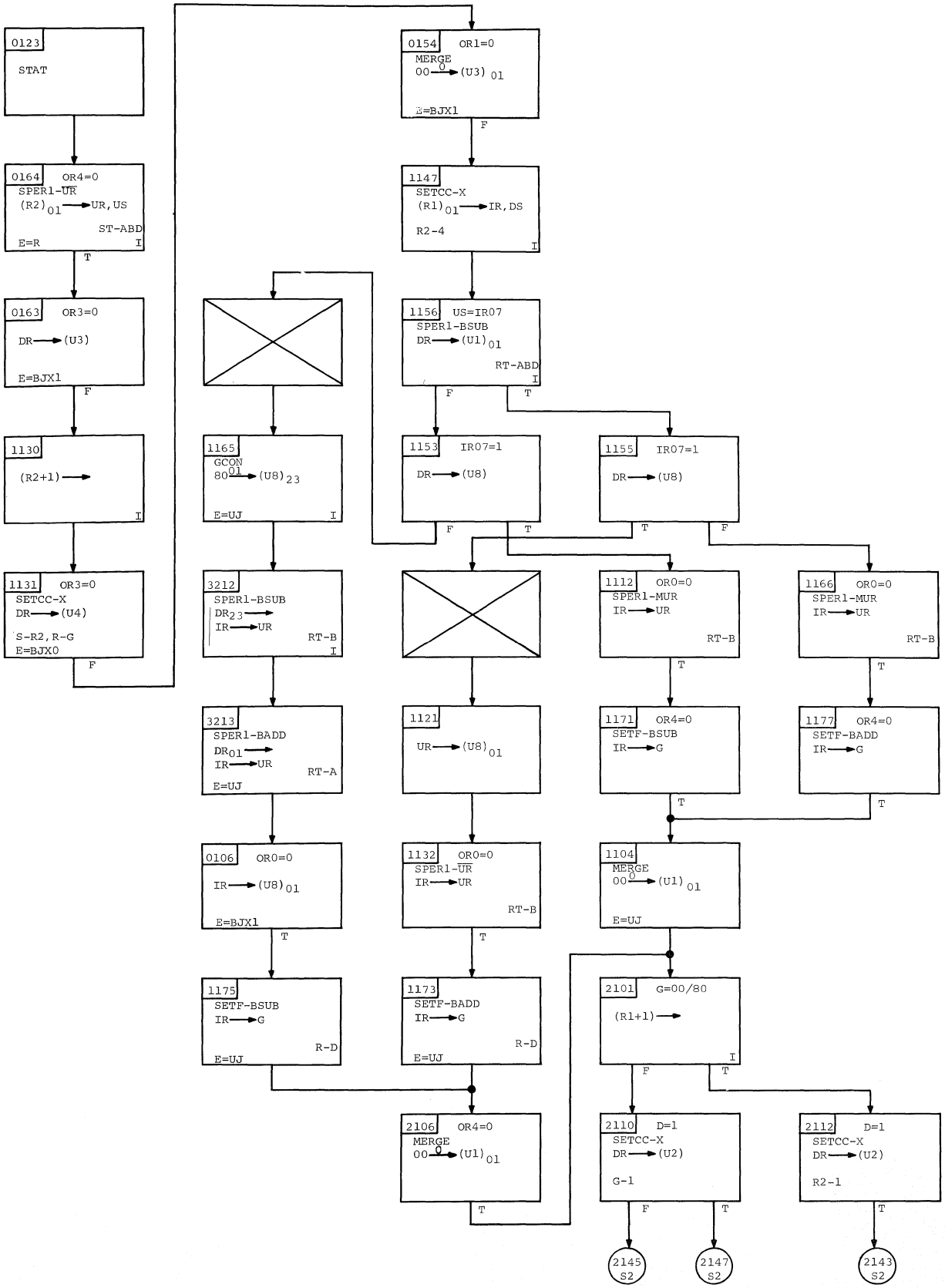
COMPARE LONG CDR RR 29



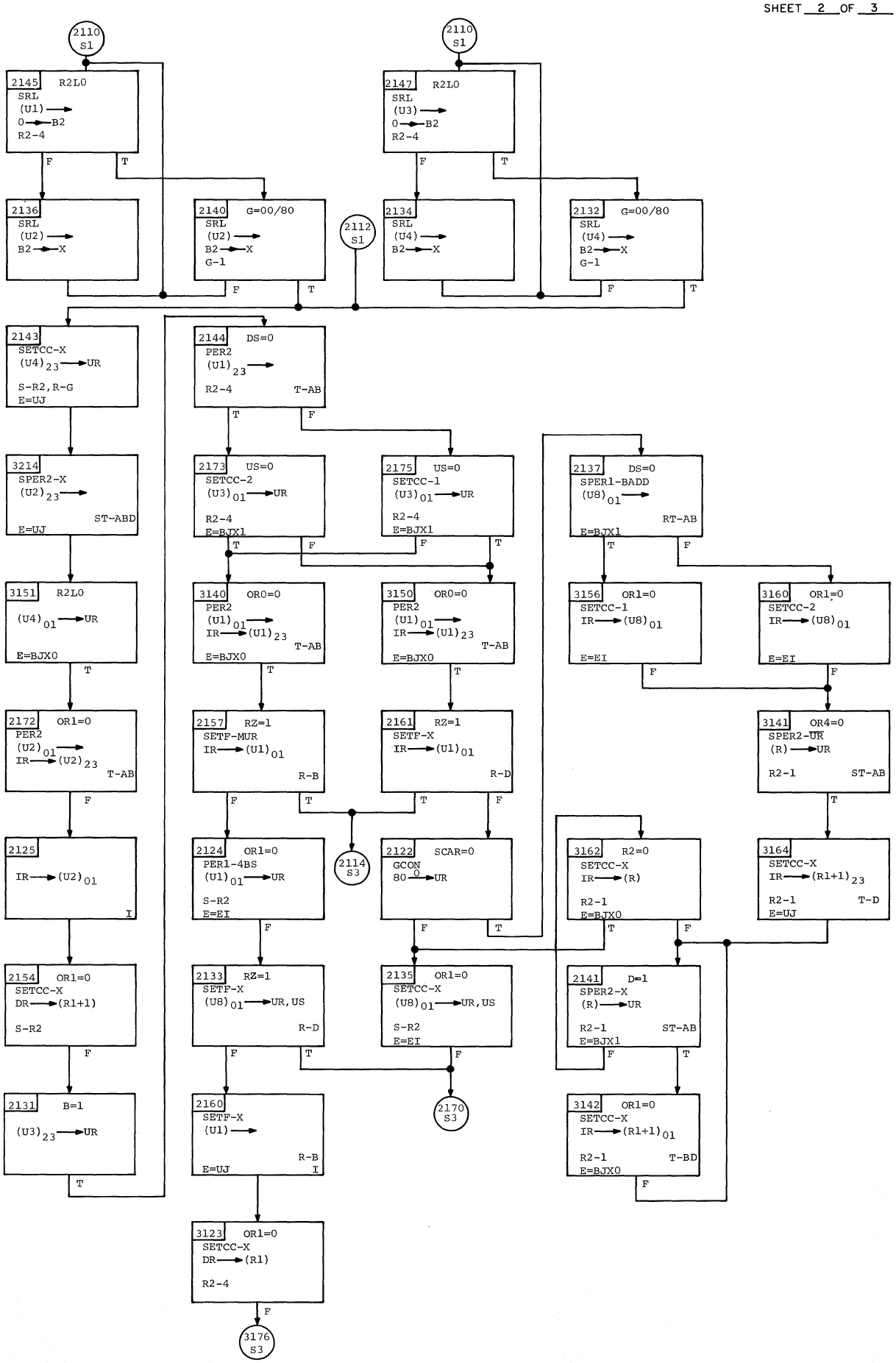
COMPARE LONG CDR RR 29



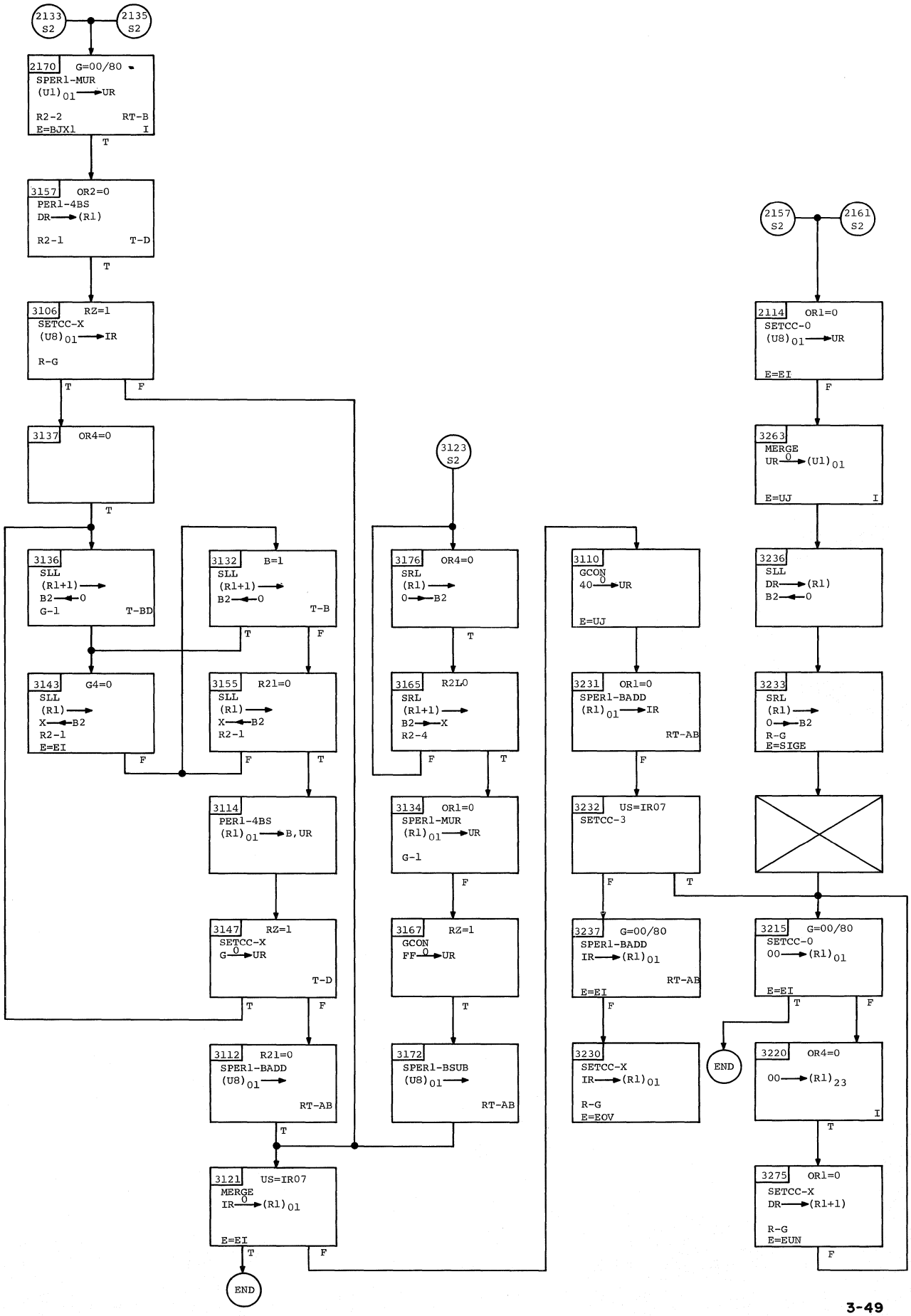
ADD NORMALIZED LONG ANDR RR 2A



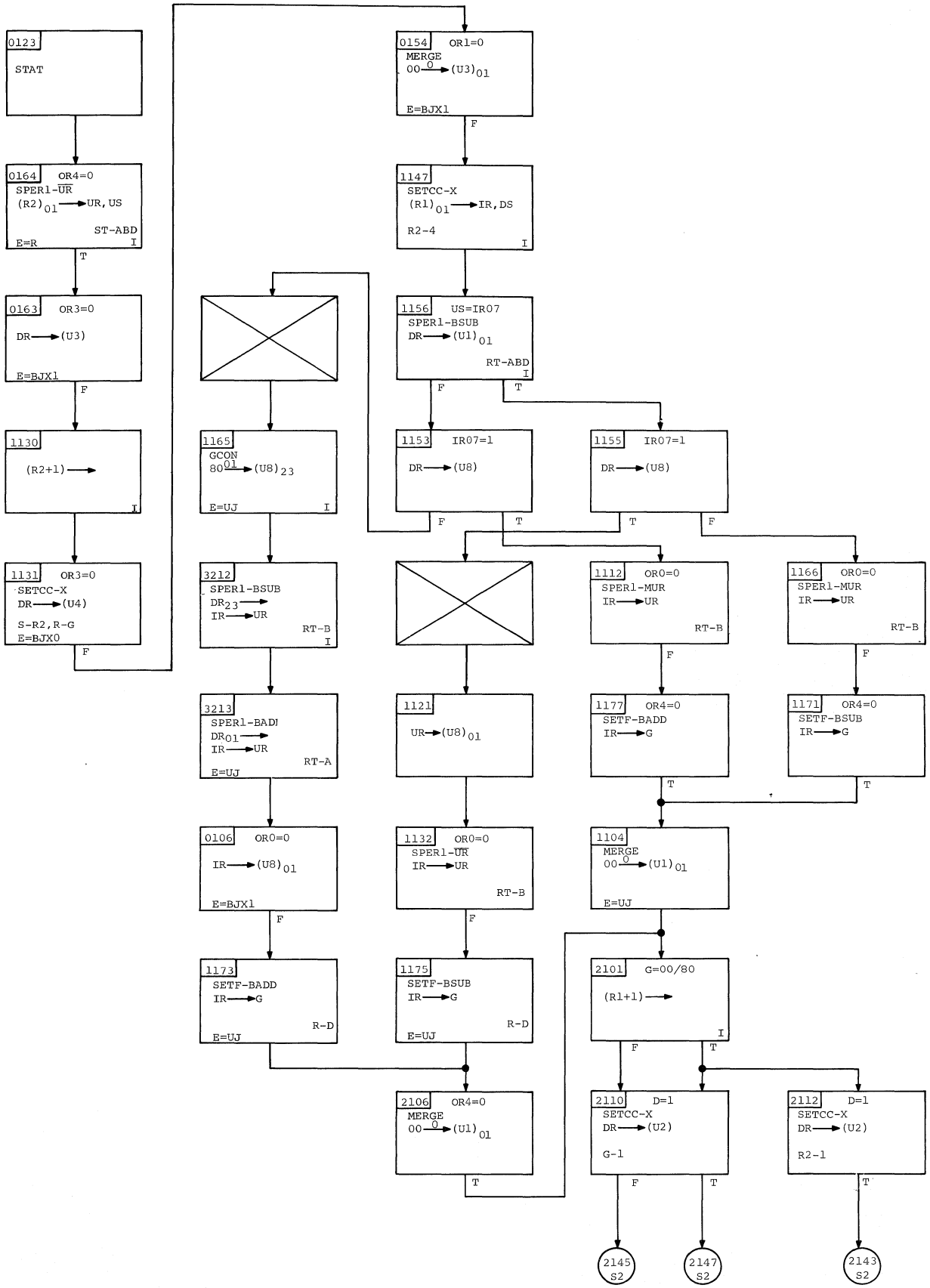
ADD NORMALIZED LONG ANDR RR 2A



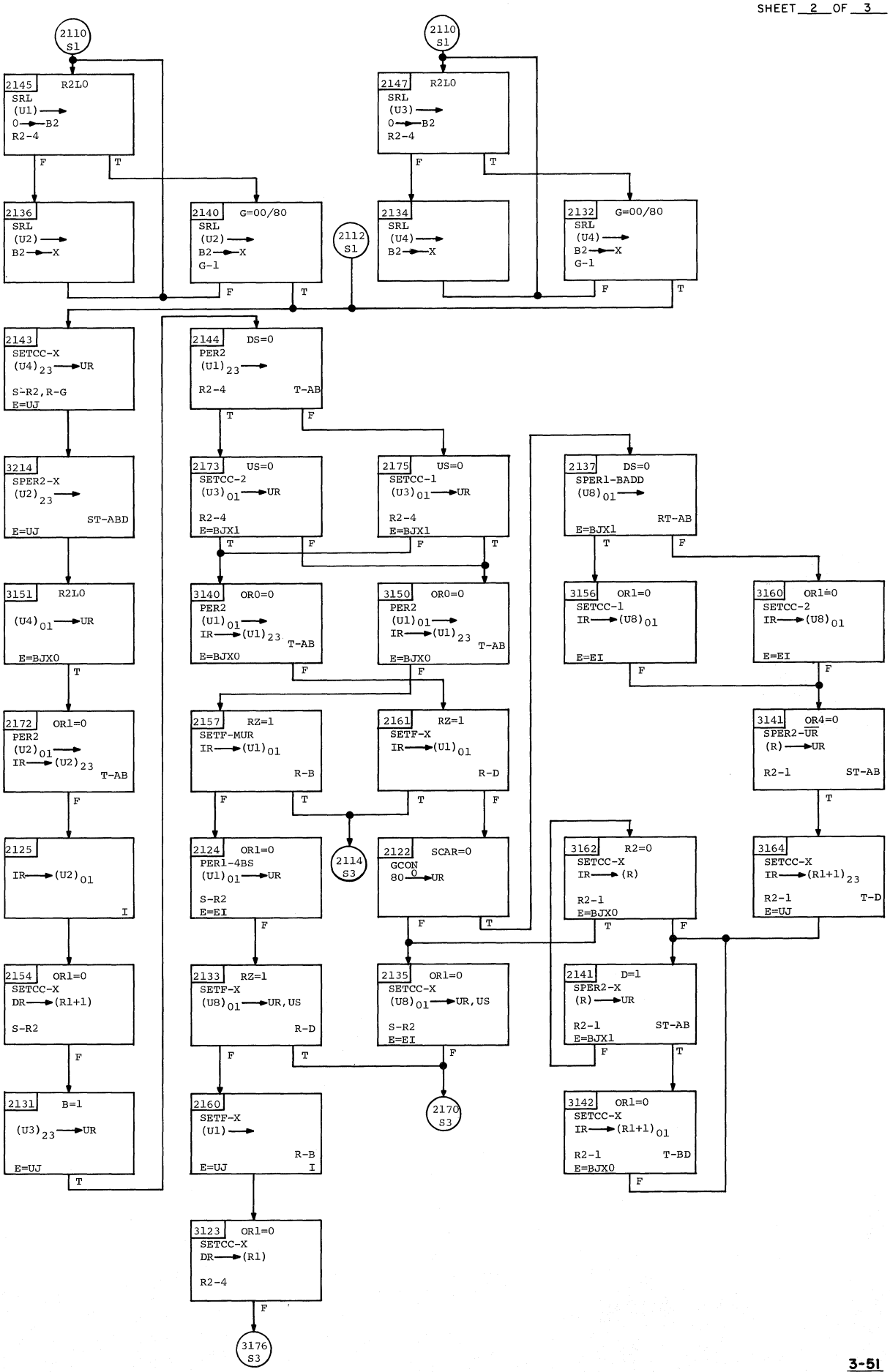
ADD NORMALIZED LONG ANDR RR 2A



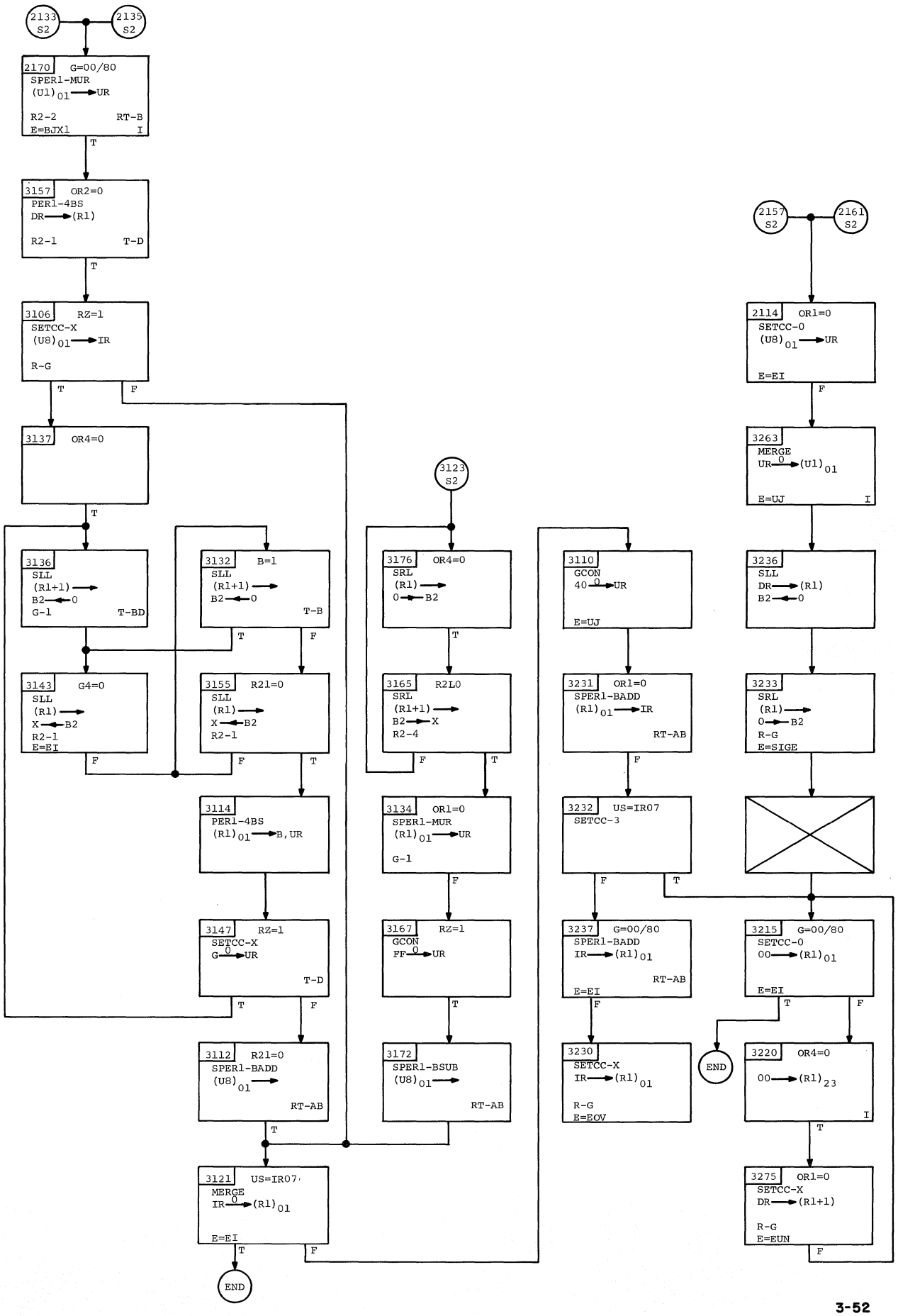
SUBTRACT NORMALIZED LONG SNDR RR 2B



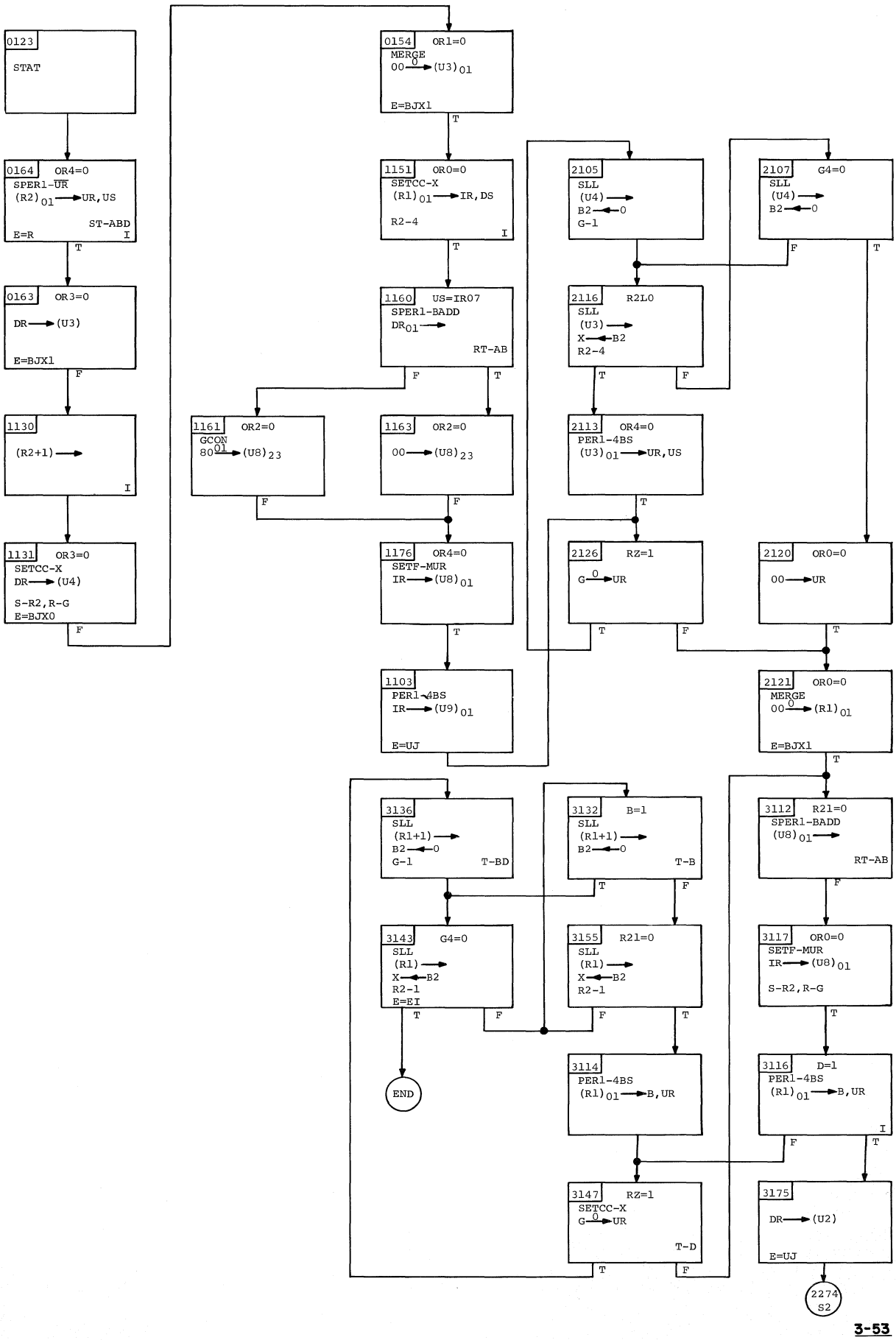
SUBTRACT NORMALIZED LONG SNDR RR 2B



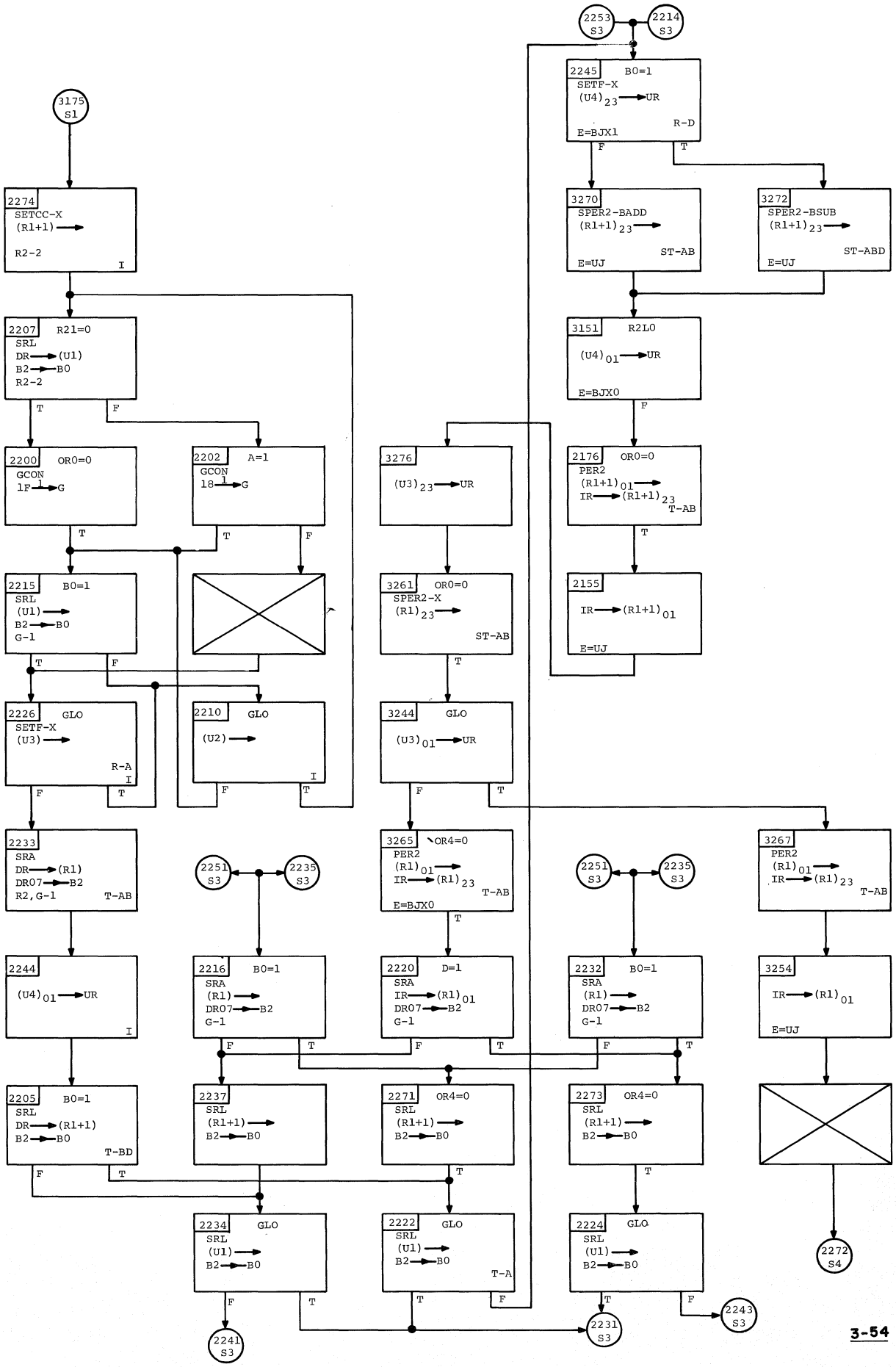
SUBTRACT NORMALIZED LONG SNDR RR 2B

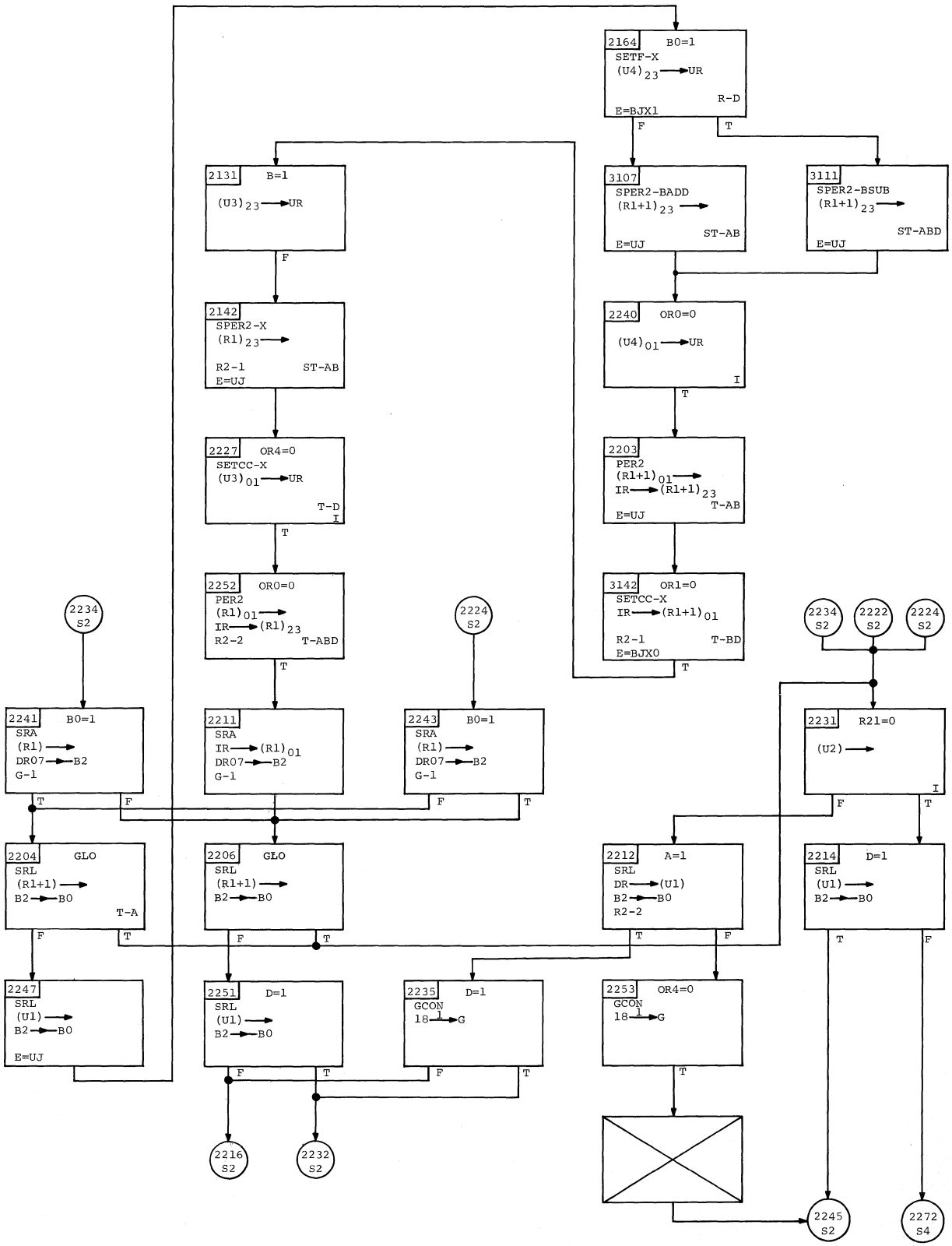


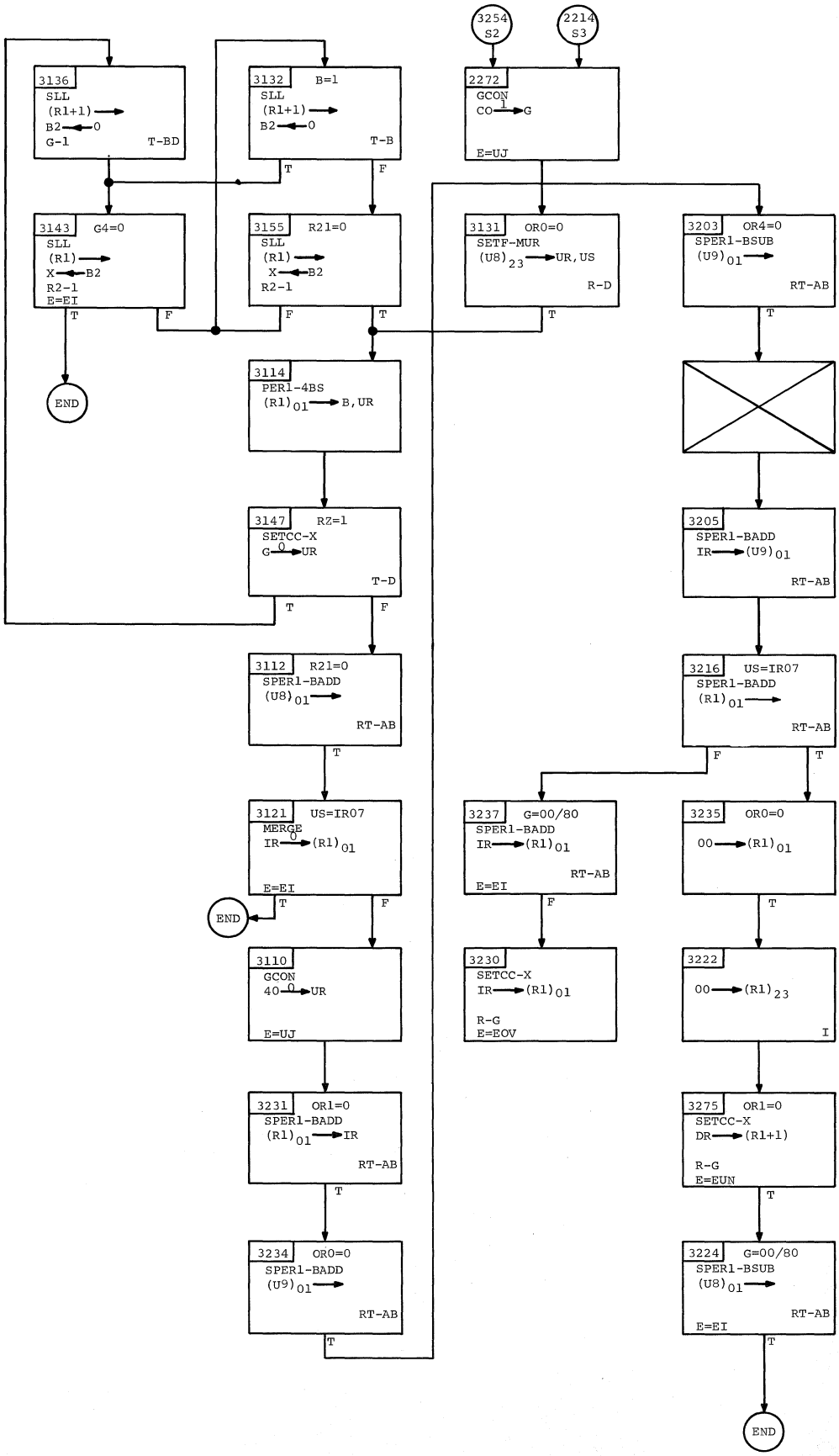
MULTIPLY LONG MDR RR 2C



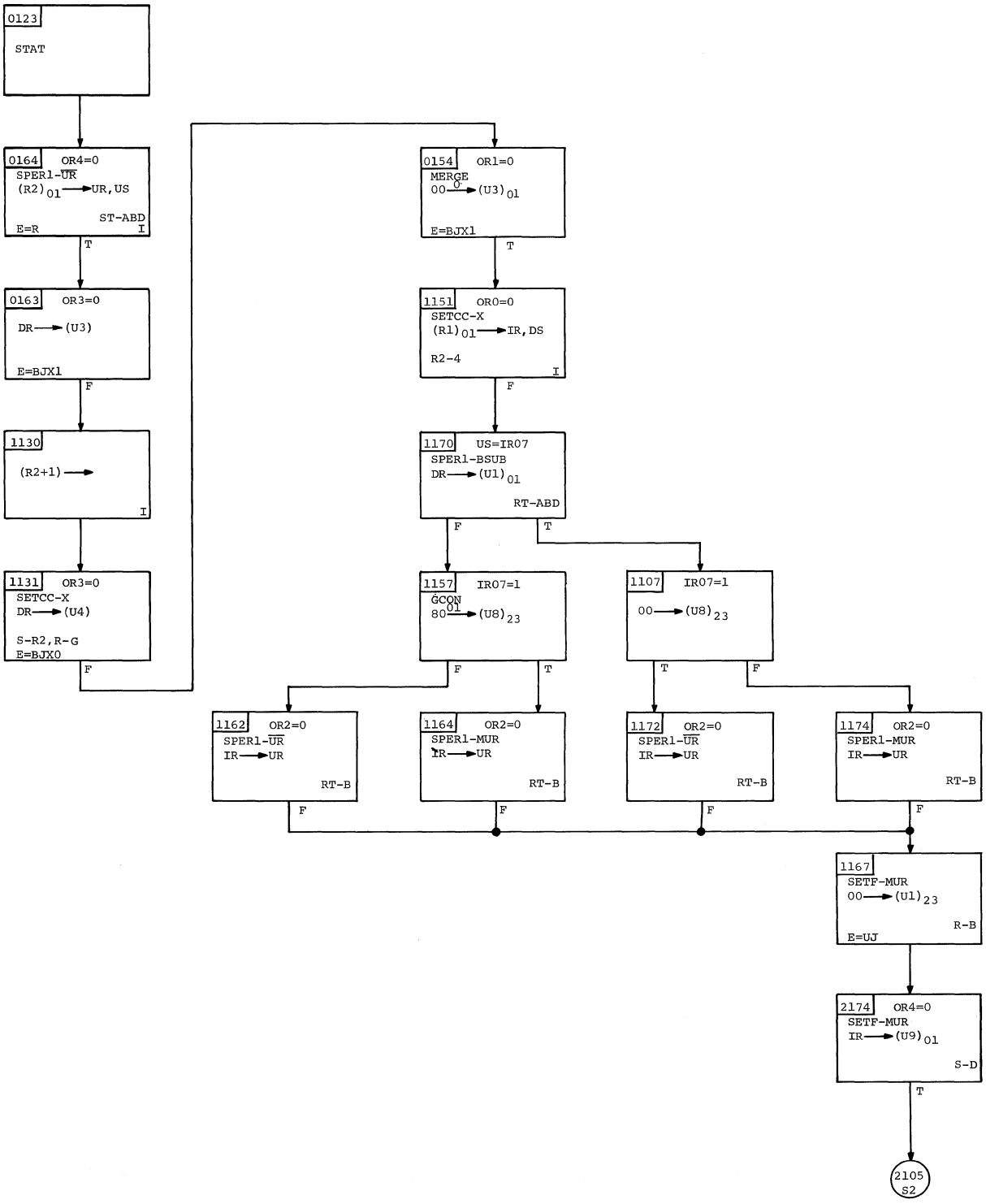
MULTIPLY LONG MDR RR 2C



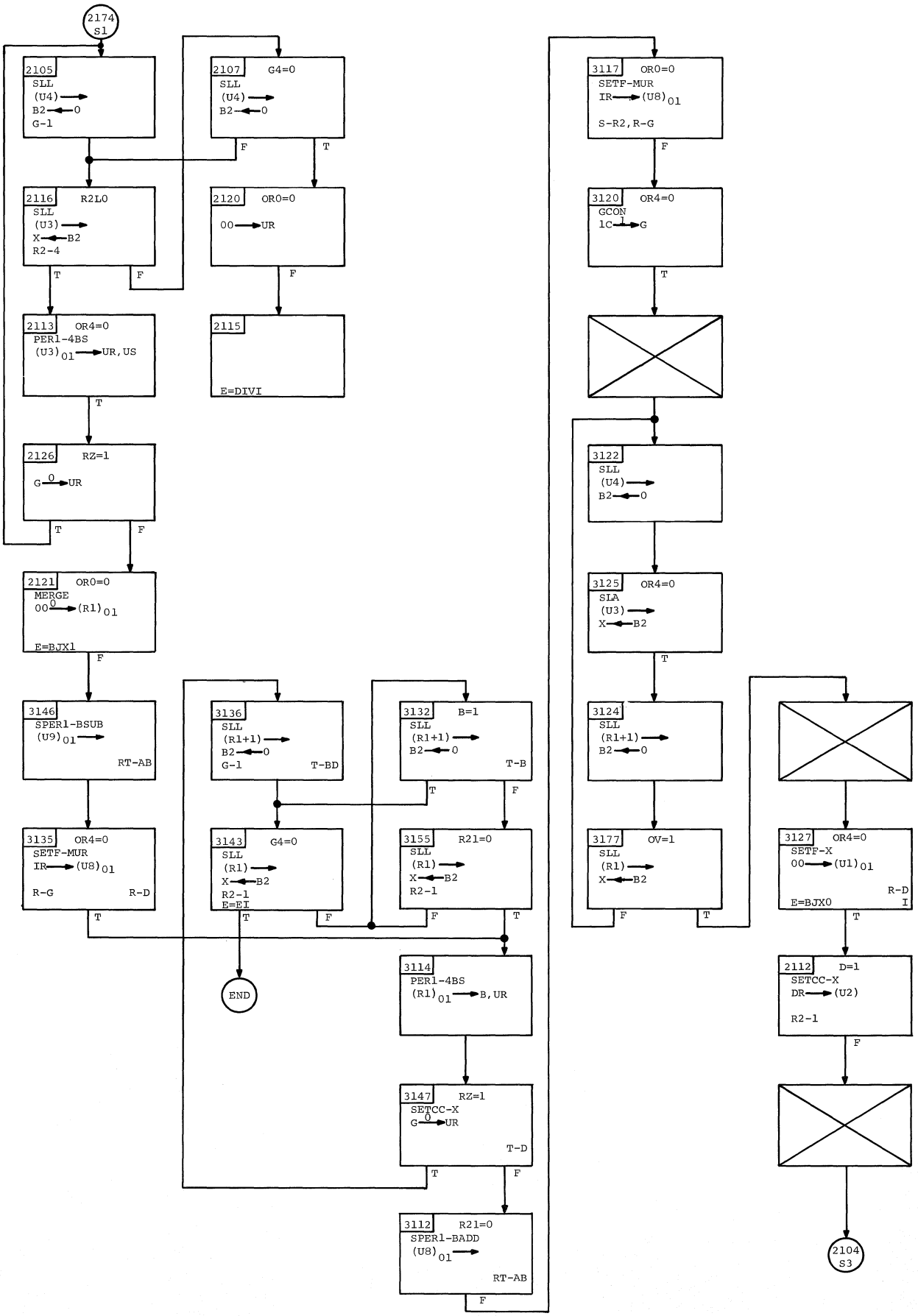

2164 B0=1
SETF-X
(U4)23 → UR
R-D
E=BJX1
F
T
2131 B=1
(U3)23 → UR
3107
SPER2-BADD
(R1+1)23 →
ST-AB
E=UJ
3111
SPER2-BSUB
(R1+1)23 →
ST-ABD
E=UJ
2142
SPER2-X
(R1)23 →
R2-1 ST-AB
E=UJ
2240 OR0=0
(U4)01 → UR
I
2227 OR4=0
SETCC-X
(U3)01 → UR
T-D
I
2203
PER2
(R1+1)01 →
IR → (R1+1)23
T-AB
E=UJ
2252 OR0=0
PER2
(R1)01 →
IR → (R1)23
R2-2 T-ABD
3142 OR1=0
SETCC-X
IR → (R1+1)01
R2-1 T-BD
E=BJX0
2234 S2
2224 S2
2234 S2
2222 S2
2224 S2
2241 B0=1
SRA
(R1) →
DR07 → B2
G-1
2211
SRA
IR → (R1)01
DR07 → B2
G-1
2243 B0=1
SRA
(R1) →
DR07 → B2
G-1
2231 R21=0
(U2) →
I
2204 GLO
SRL
(R1+1) →
B2 → B0
T-A
2206 GLO
SRL
(R1+1) →
B2 → B0
2212 A=1
SRL
DR → (U1)
B2 → B0
R2-2
2214 D=1
SRL
(U1) →
B2 → B0
2247
SRL
(U1) →
B2 → B0
E=UJ
2251 D=1
SRL
(U1) →
B2 → B0
2235 D=1
GCON
18 1 → G
2253 OR4=0
GCON
18 1 → G
2216 S2
2232 S2
2245 S2
2272 S4


3254 S2
2214 S3
3136 SLL (R1+1) B2 G-1 T-BD
3132 B=1 SLL (R1+1) B2 T-B
2272 GCON CO 1 G E=UJ
3143 G4=0 SLL (R1) X B2 R2-1 E=EI
3155 R21=0 SLL (R1) X B2 R2-1
3131 OR0=0 SETF-MUR (U8)23 UR,US R-D
3203 OR4=0 SPER1-BSUB (U9)01 RT-AB
END
3114 PER1-4BS (R1)01 B,UR
3147 RZ=1 SETCC-X G 0 UR T-D
3205 SPER1-BADD IR (U9)01 RT-AB
3112 R21=0 SPER1-BADD (U8)01 RT-AB
3216 US=IR07 SPER1-BADD (R1)01 RT-AB
3121 US=IR07 MERGE IR 0 (R1)01 E=EI
3237 G=00/80 SPER1-BADD IR (R1)01 RT-AB E=EI
3235 OR0=0 00 (R1)01
END
3110 GCON 40 0 UR E=UJ
3230 SETCC-X IR (R1)01 R-G E=EOV
3222 00 (R1)23 I
3231 OR1=0 SPER1-BADD (R1)01 IR RT-AB
3275 OR1=0 SETCC-X DR (R1+1) R-G E=EUN
3234 OR0=0 SPER1-BADD (U9)01 RT-AB
3224 G=00/80 SPER1-BSUB (U8)01 RT-AB E=EI
END

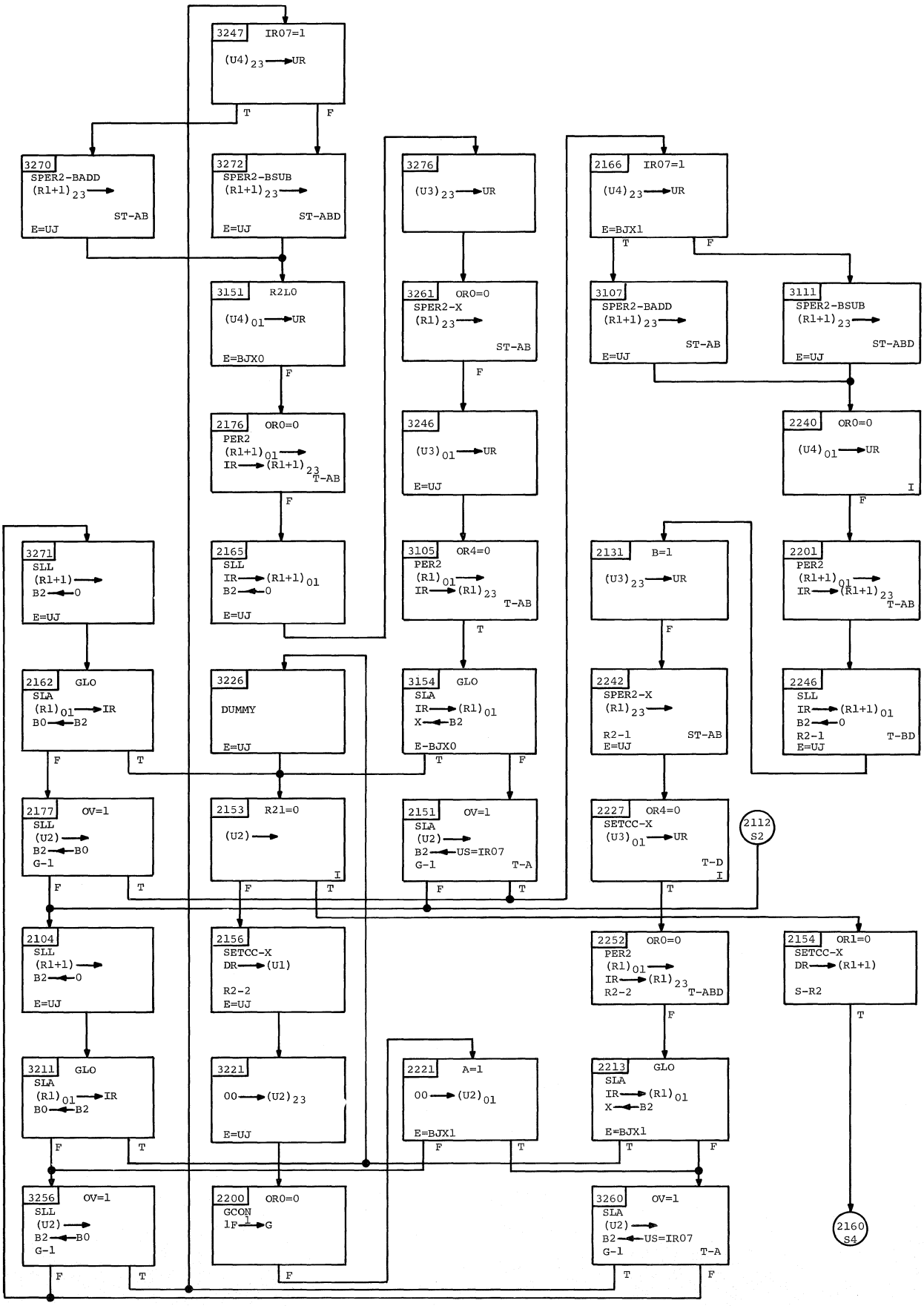
DIVIDE LONG DDR RR 2D



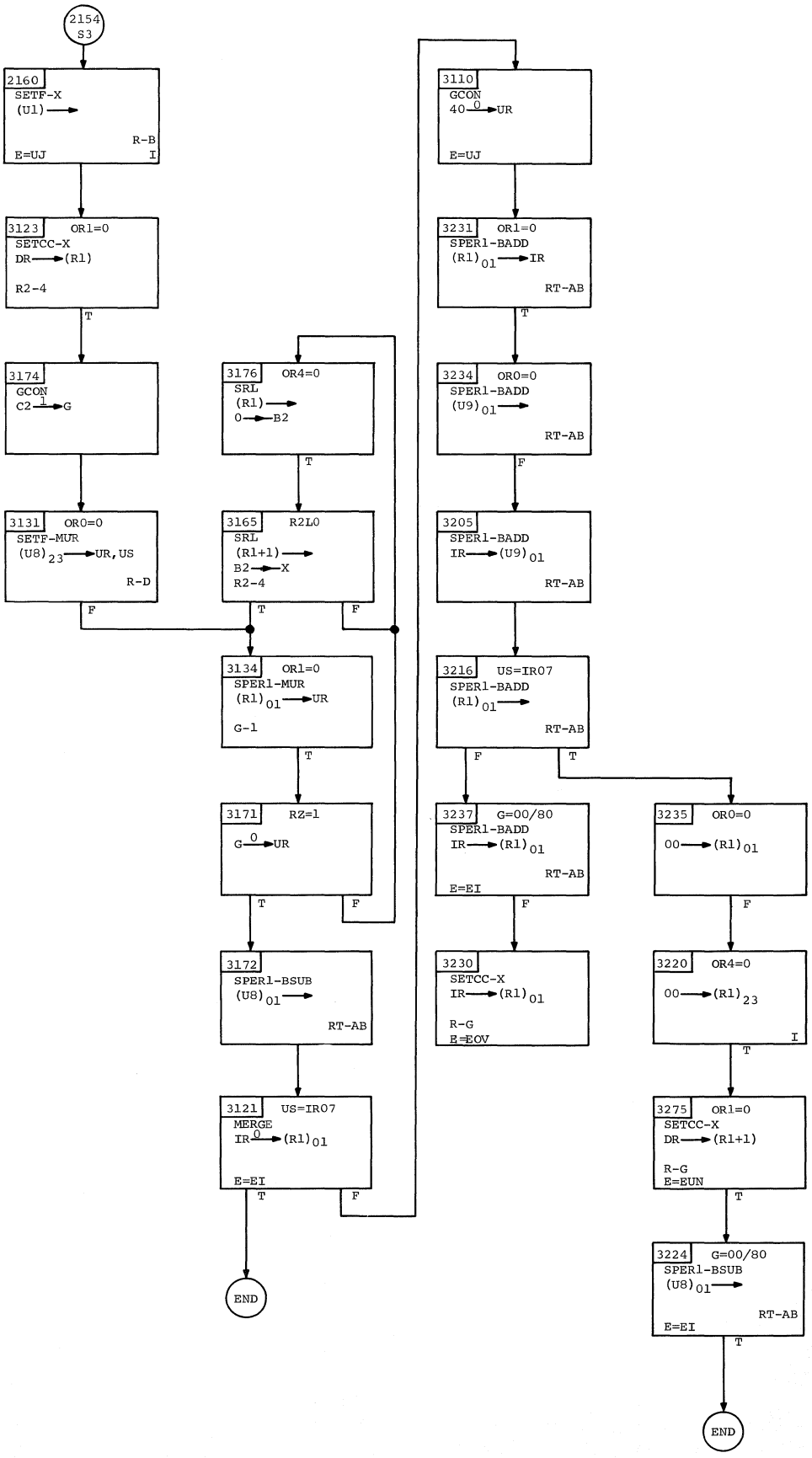
DIVIDE LONG DDR RR 2D



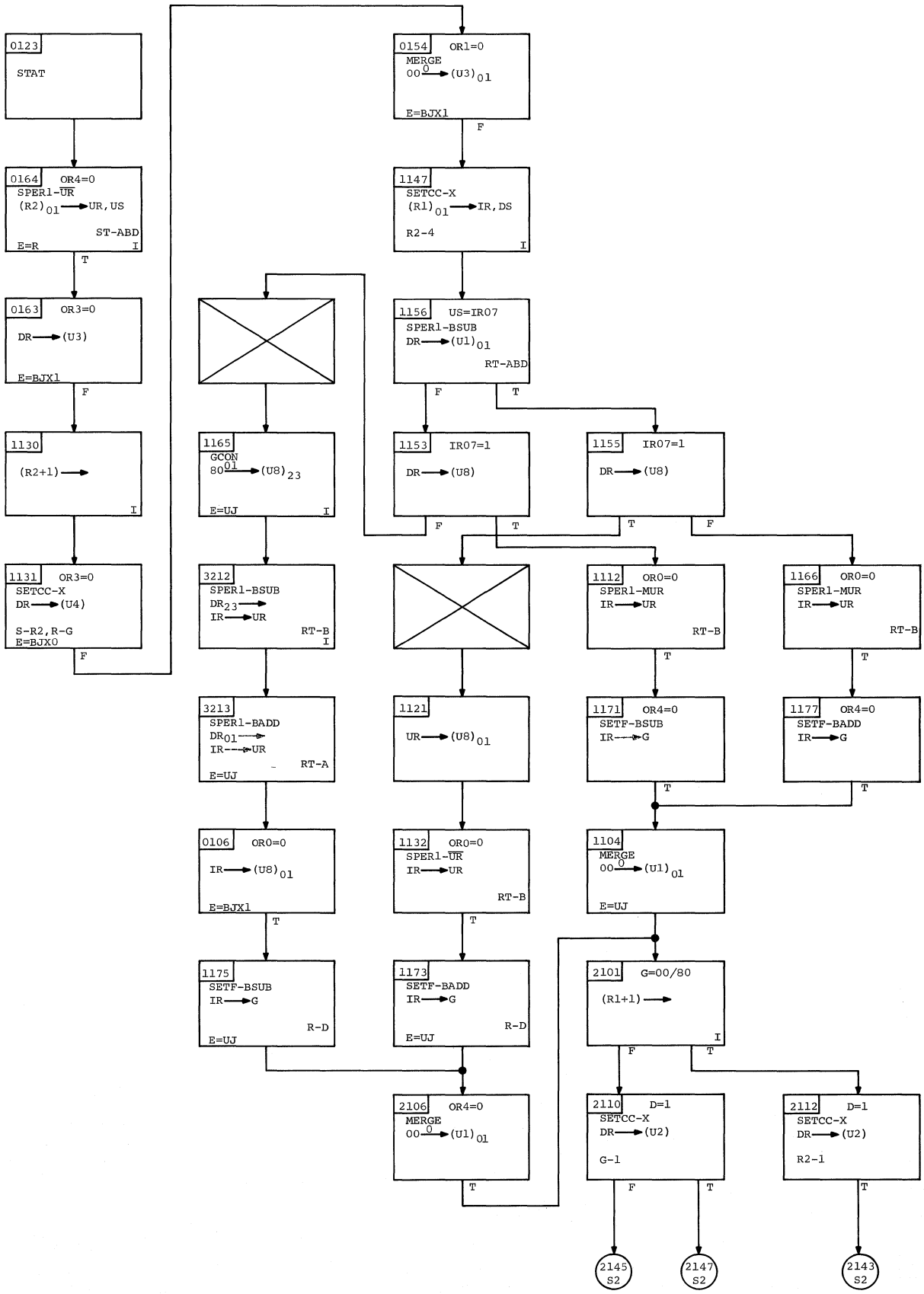
DIVIDE LONG DDR RR 2D



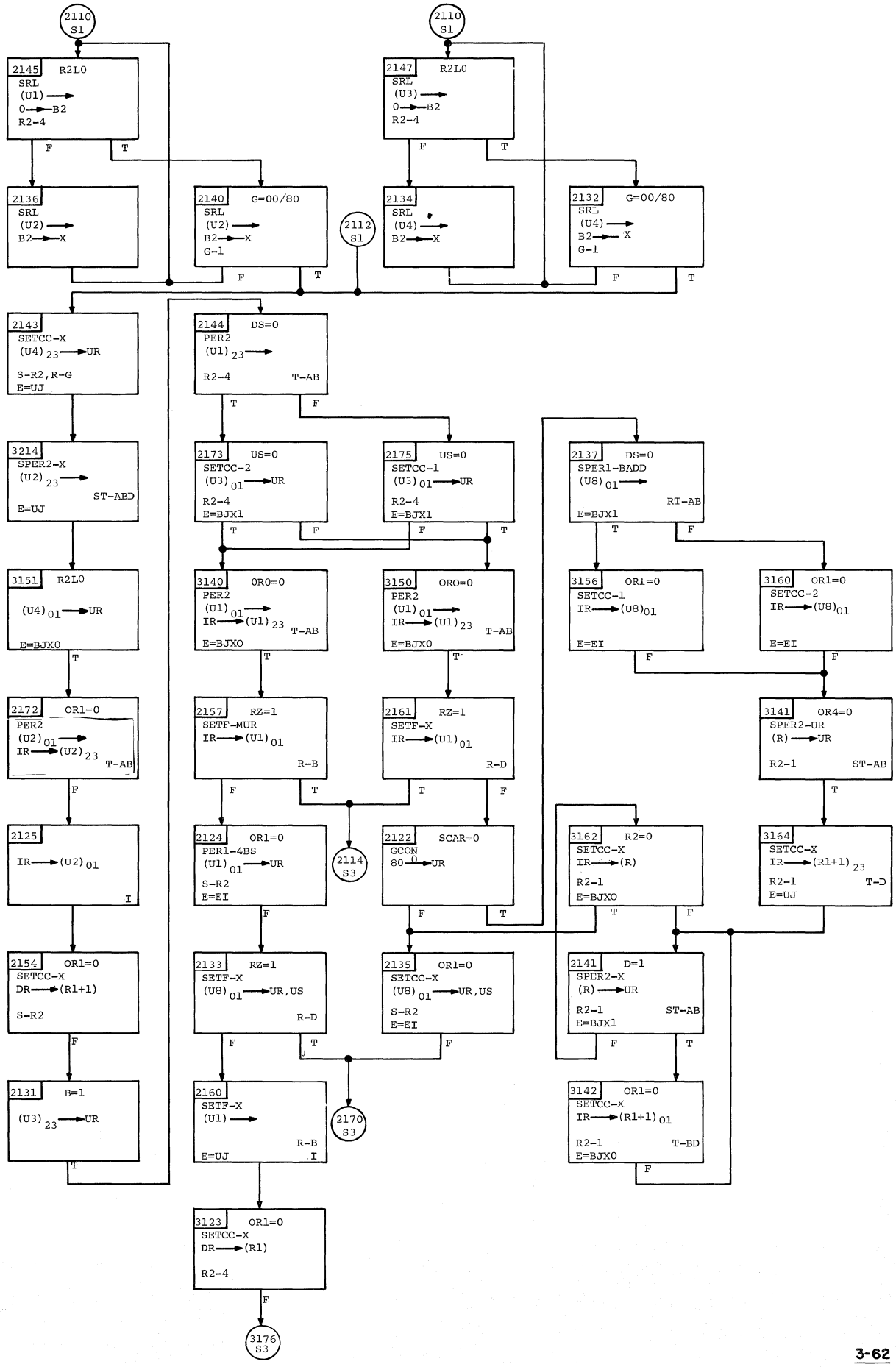
DIVIDE LONG DDR RR 2D



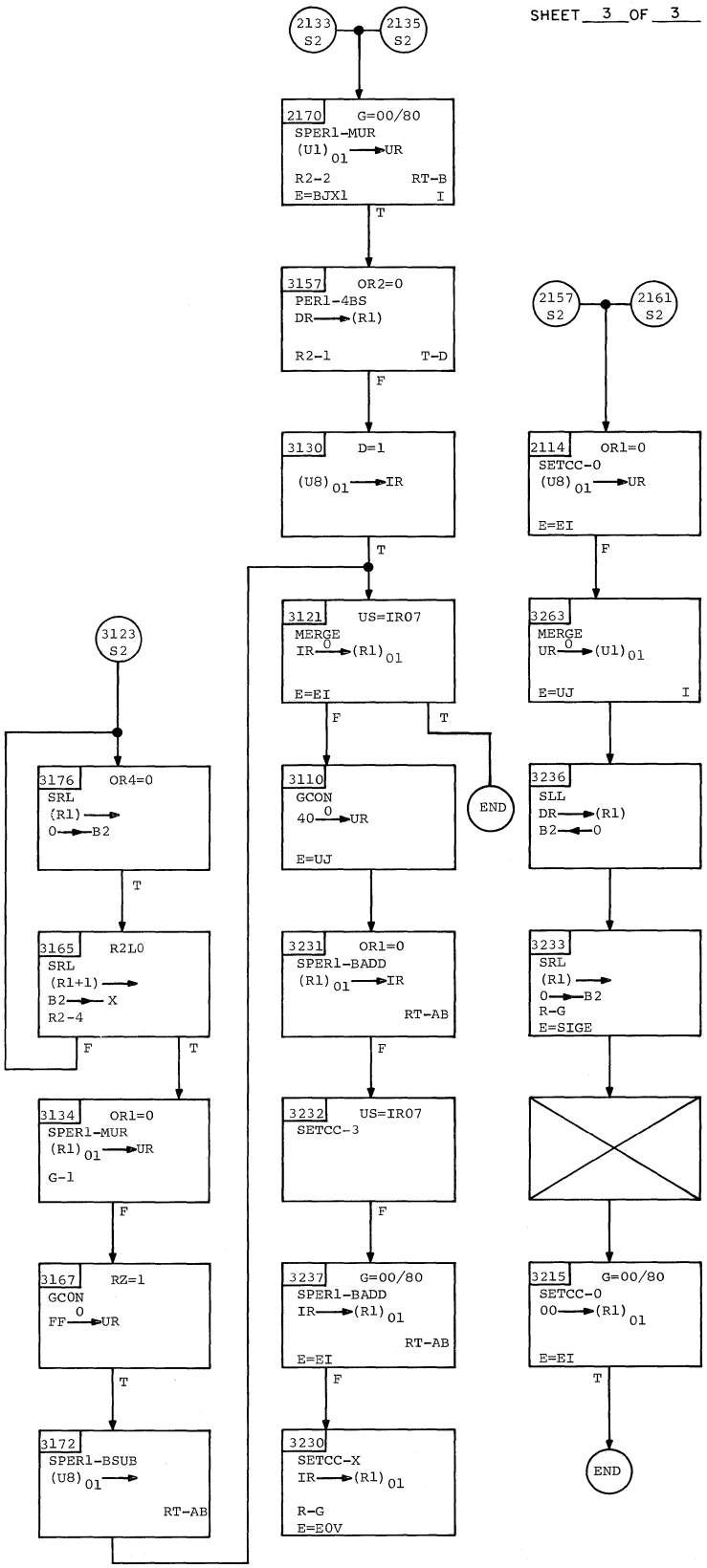
ADD UNNORMALIZED LONG AWR RR 2E



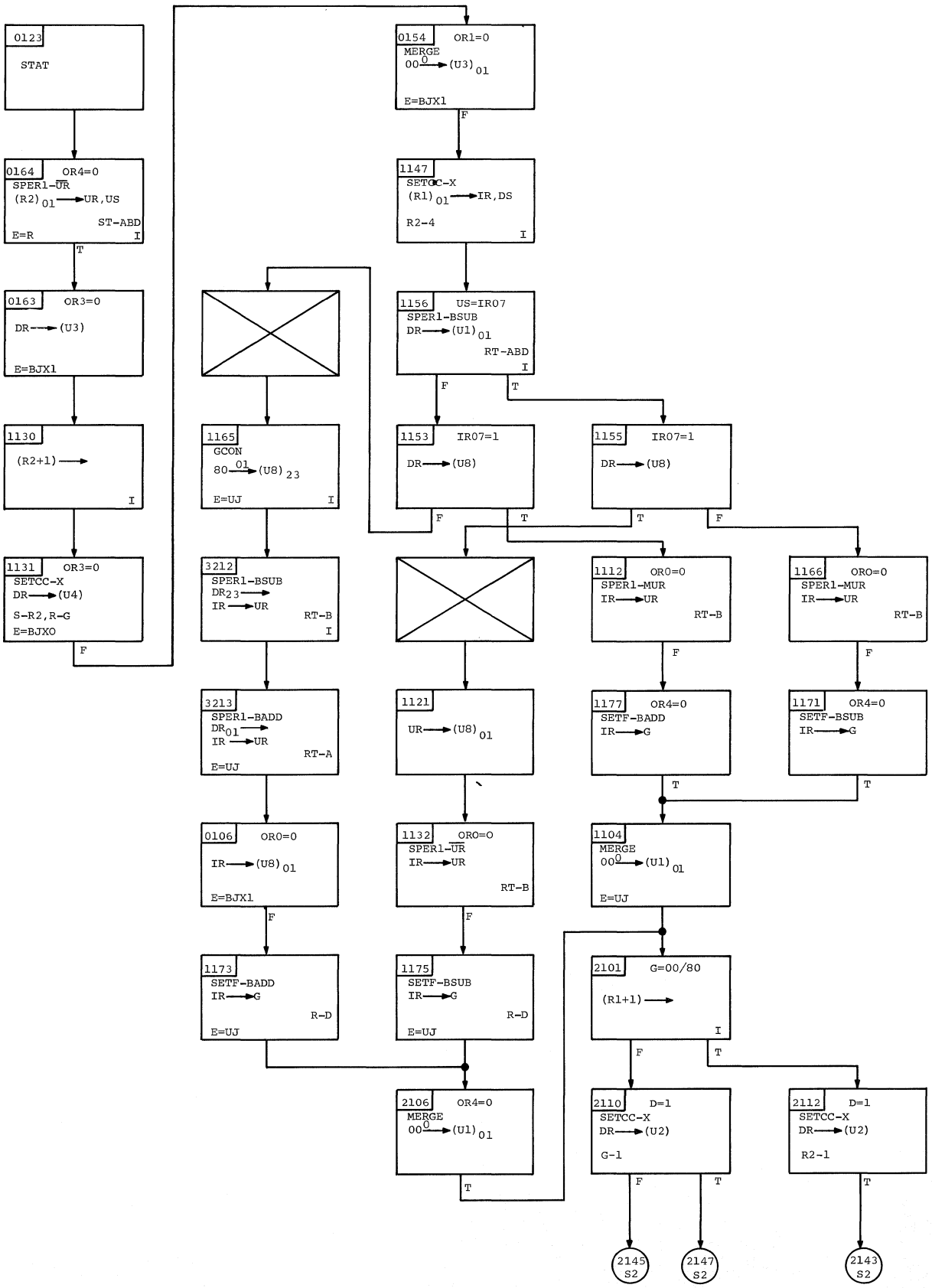
ADD UNNORMALIZED LONG AWR RR 2E



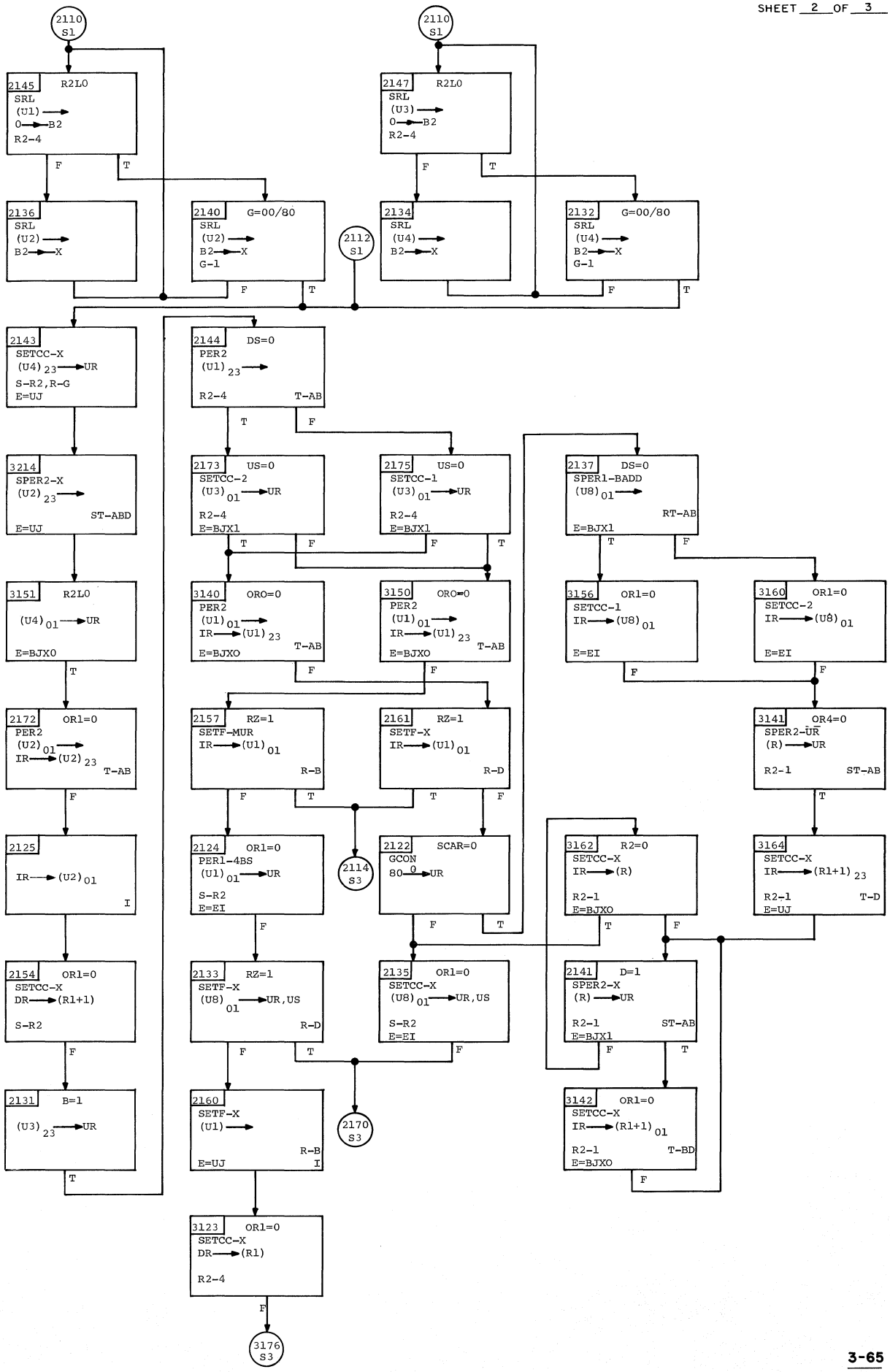
ADD UNNORMALIZED LONG AWR RR 2E



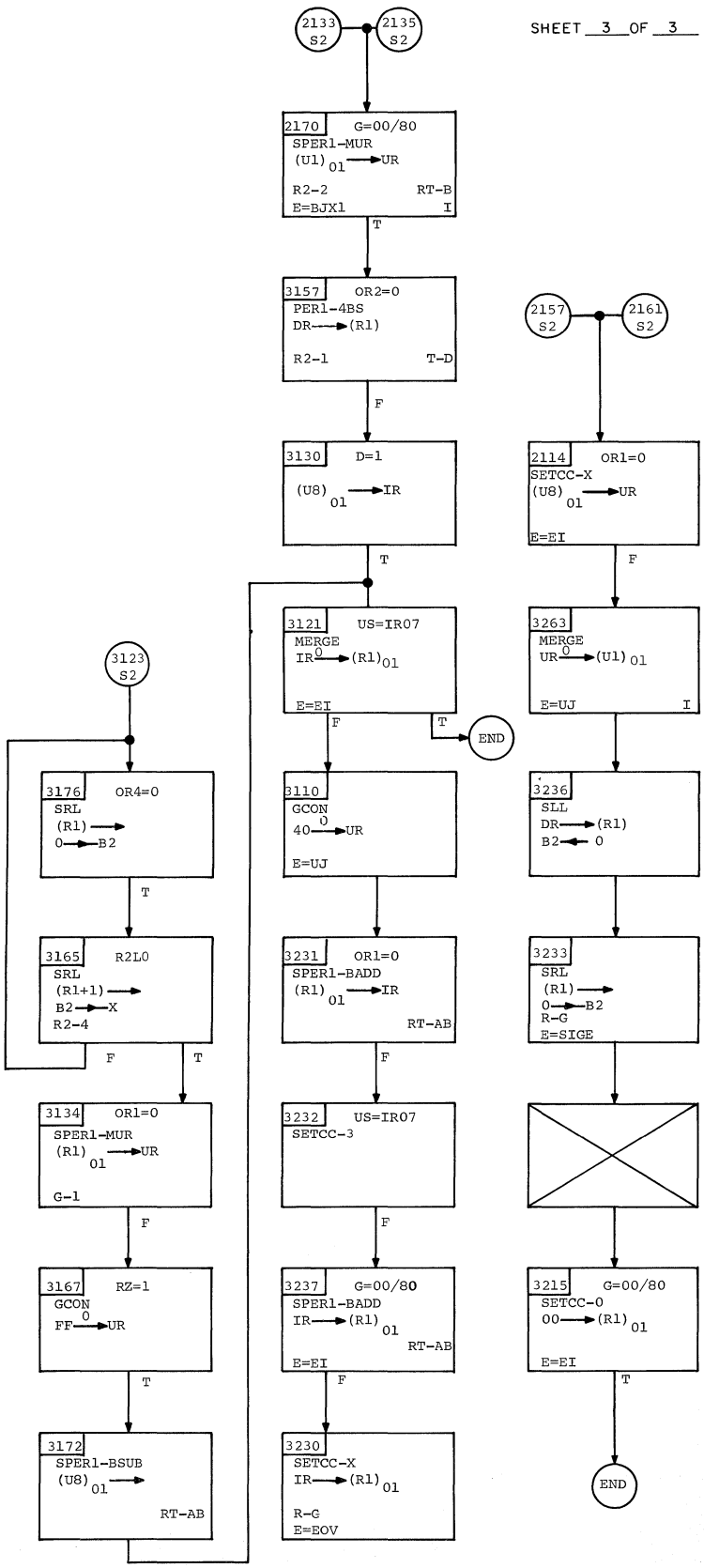
SUBTRACT UNNORMALIZED LONG SWR RR 2F



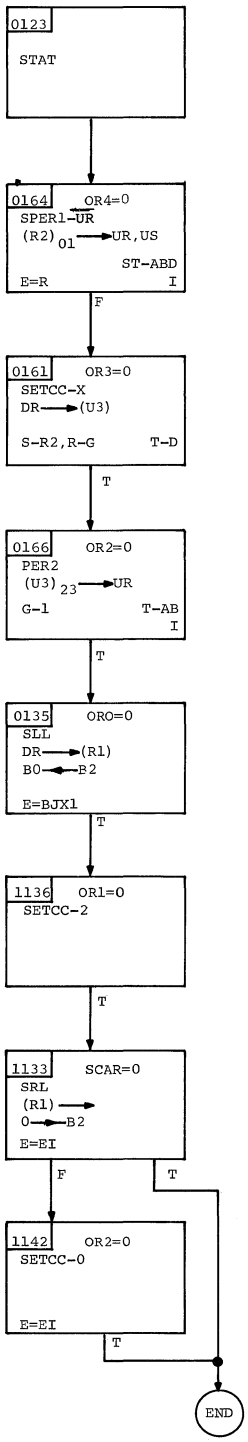
SUBTRACT UNNORMALIZED LONG SWR RR 2F



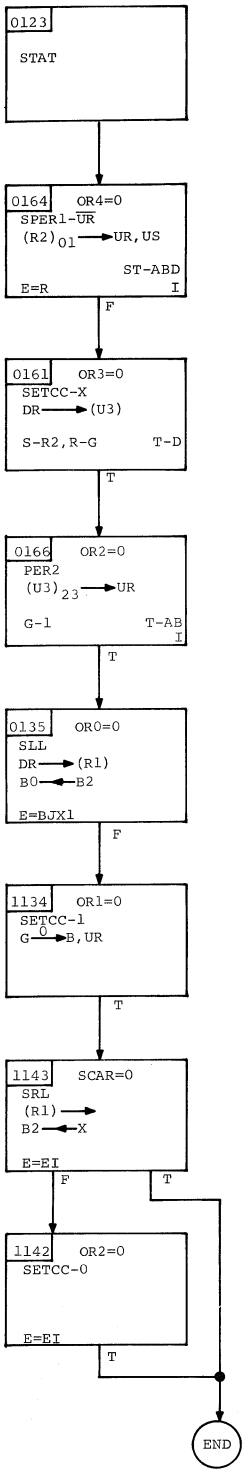
SUBTRACT UNNORMALIZED LONG SWR RR 2F



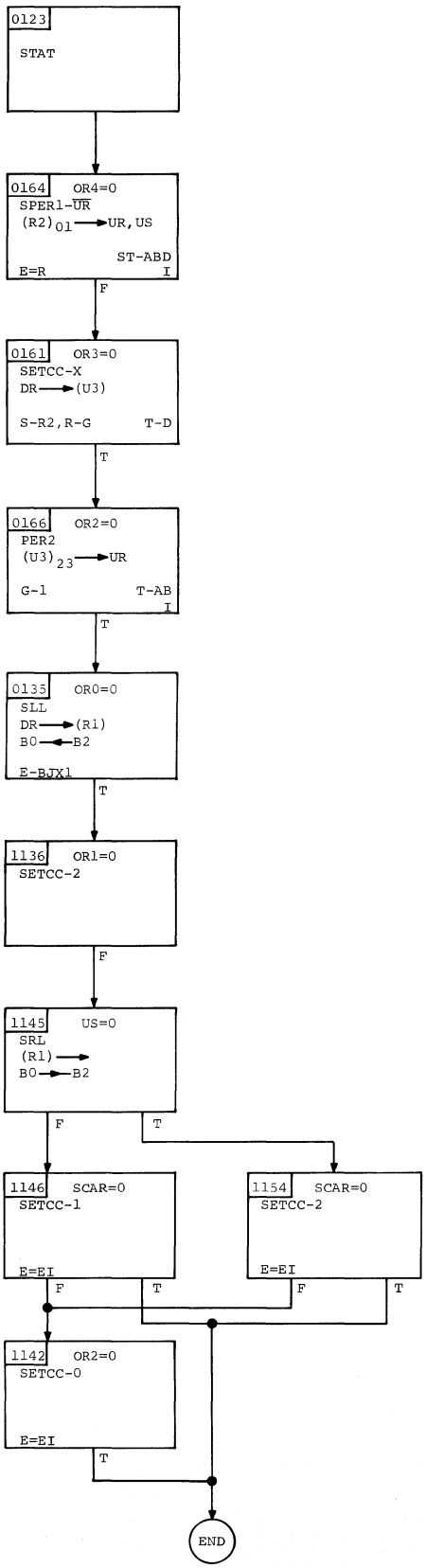
LOAD POSITIVE SHORT LPER RR 30



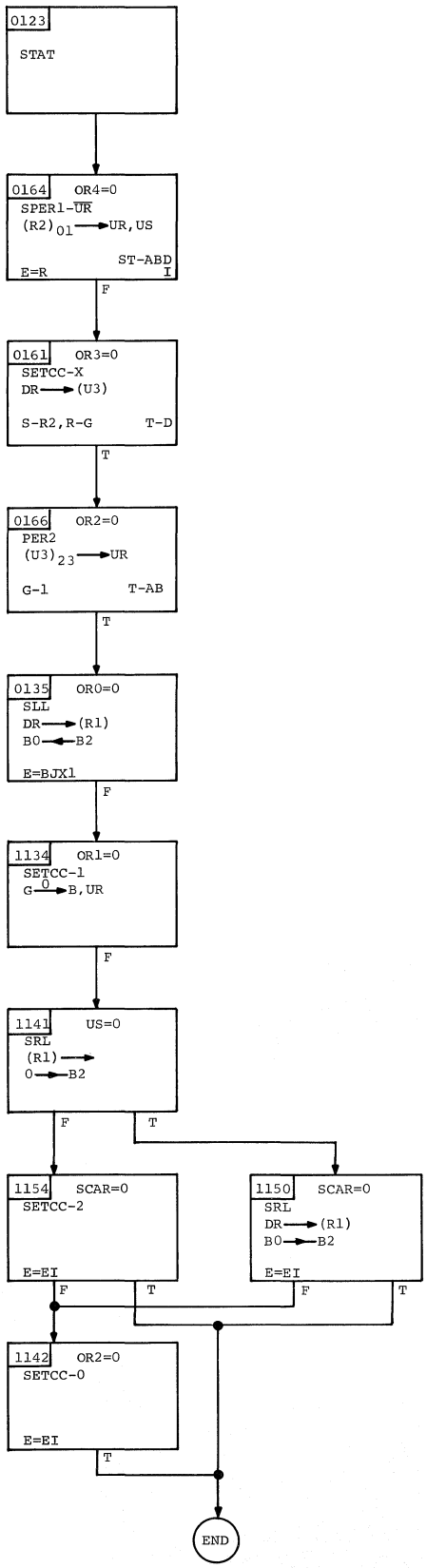
LOAD NEGATIVE SHORT LNER RR 31



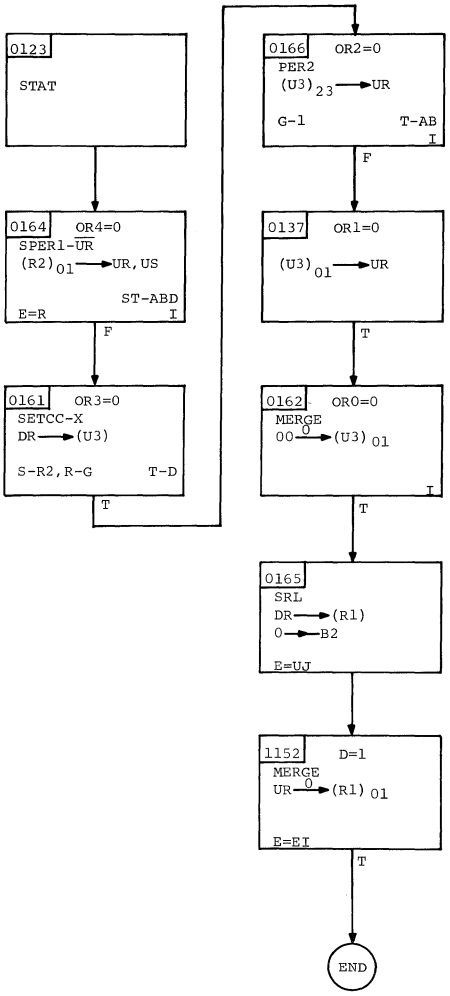
LOAD AND TEST SHORT LTER RR 32



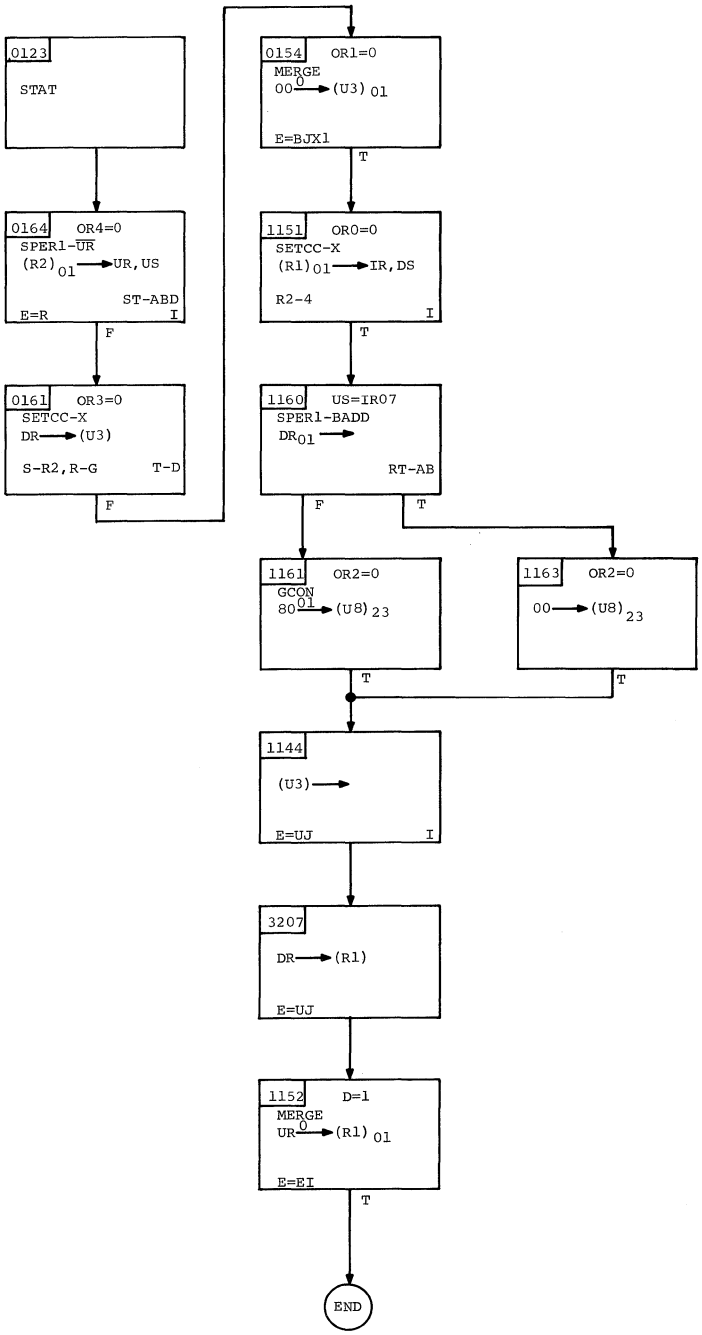
LOAD COMPLEMENT SHORT LCER RR 33



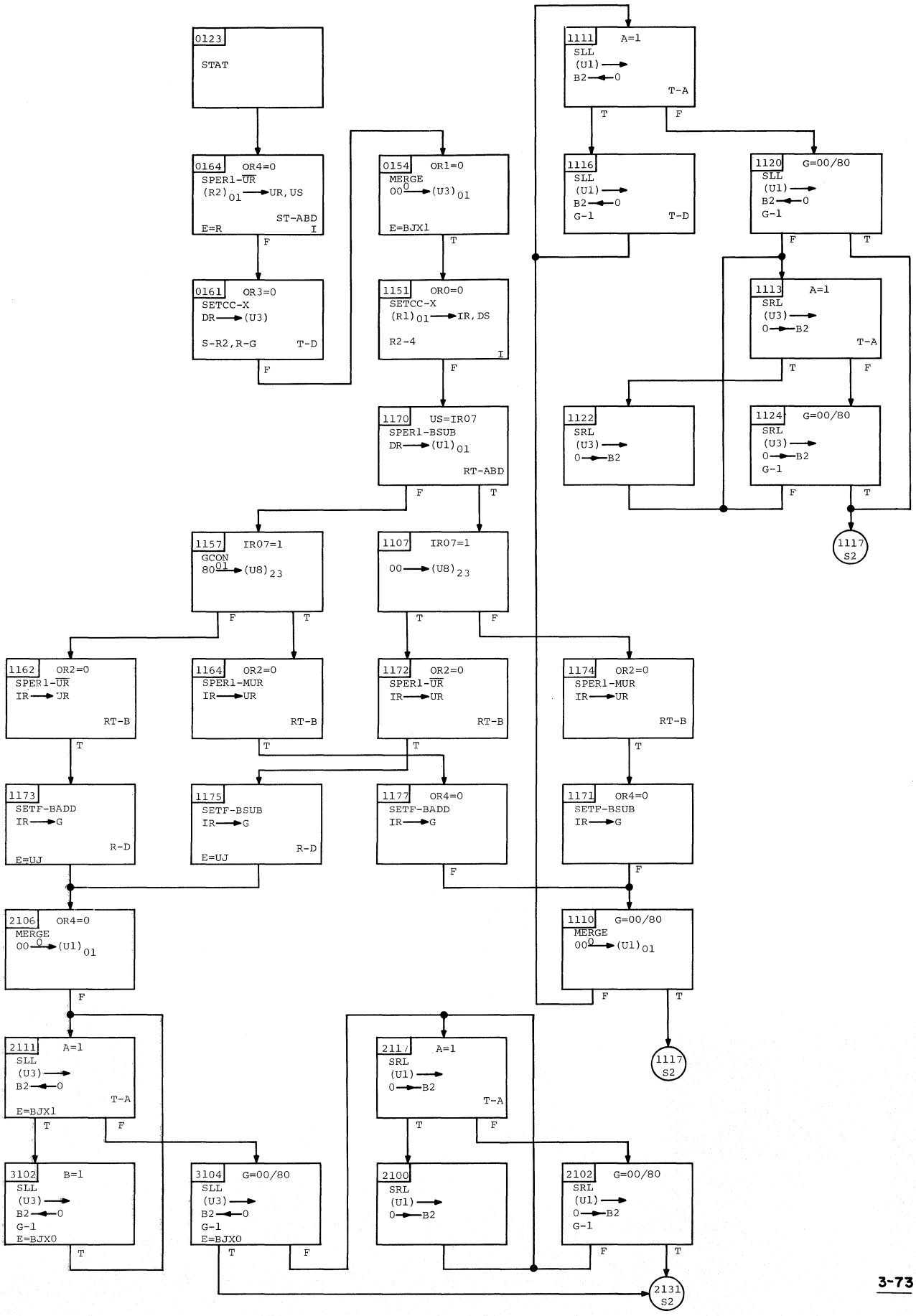
HALVE SHORT HER RR 34



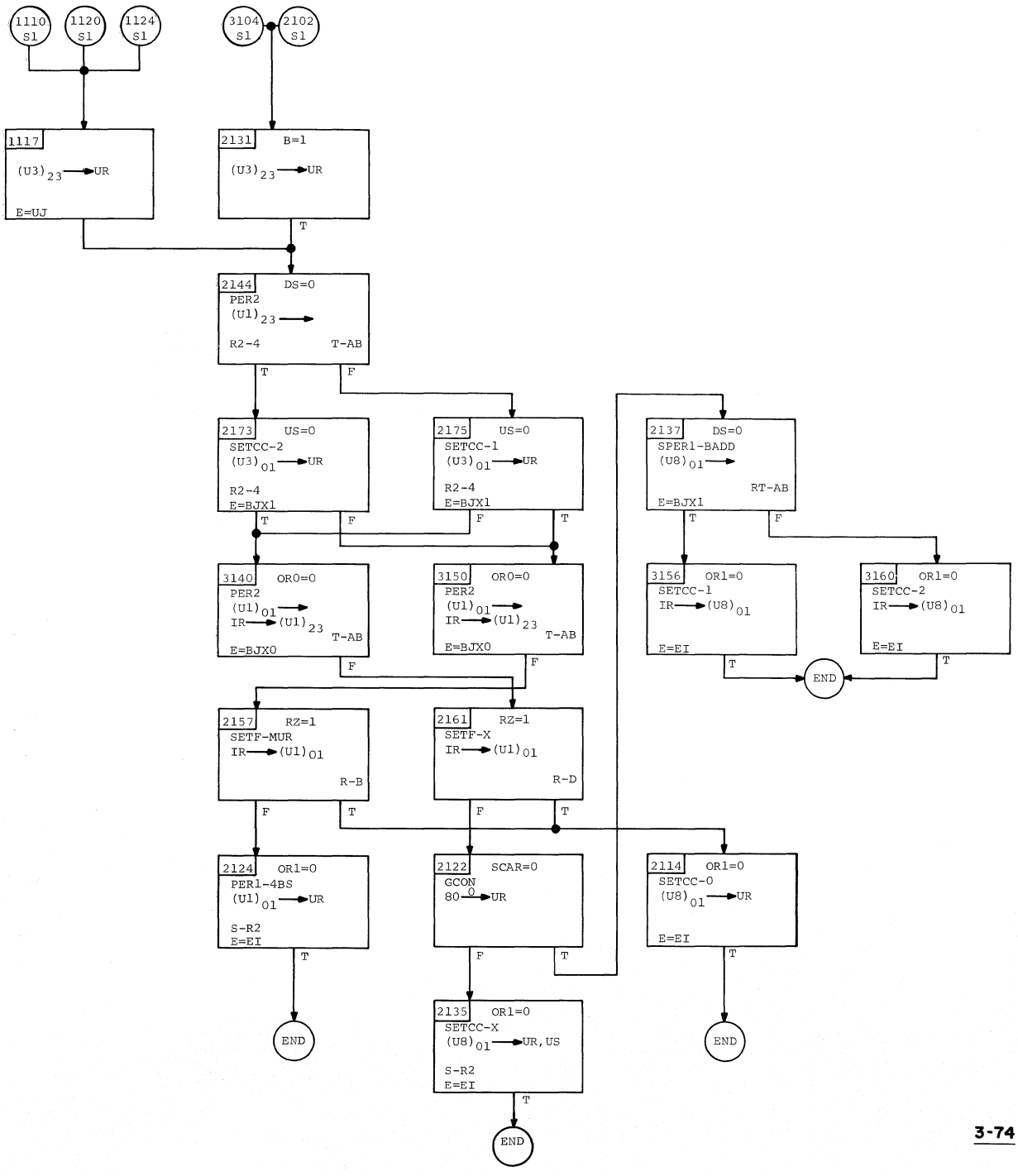
LOAD SHORT LER RR 38



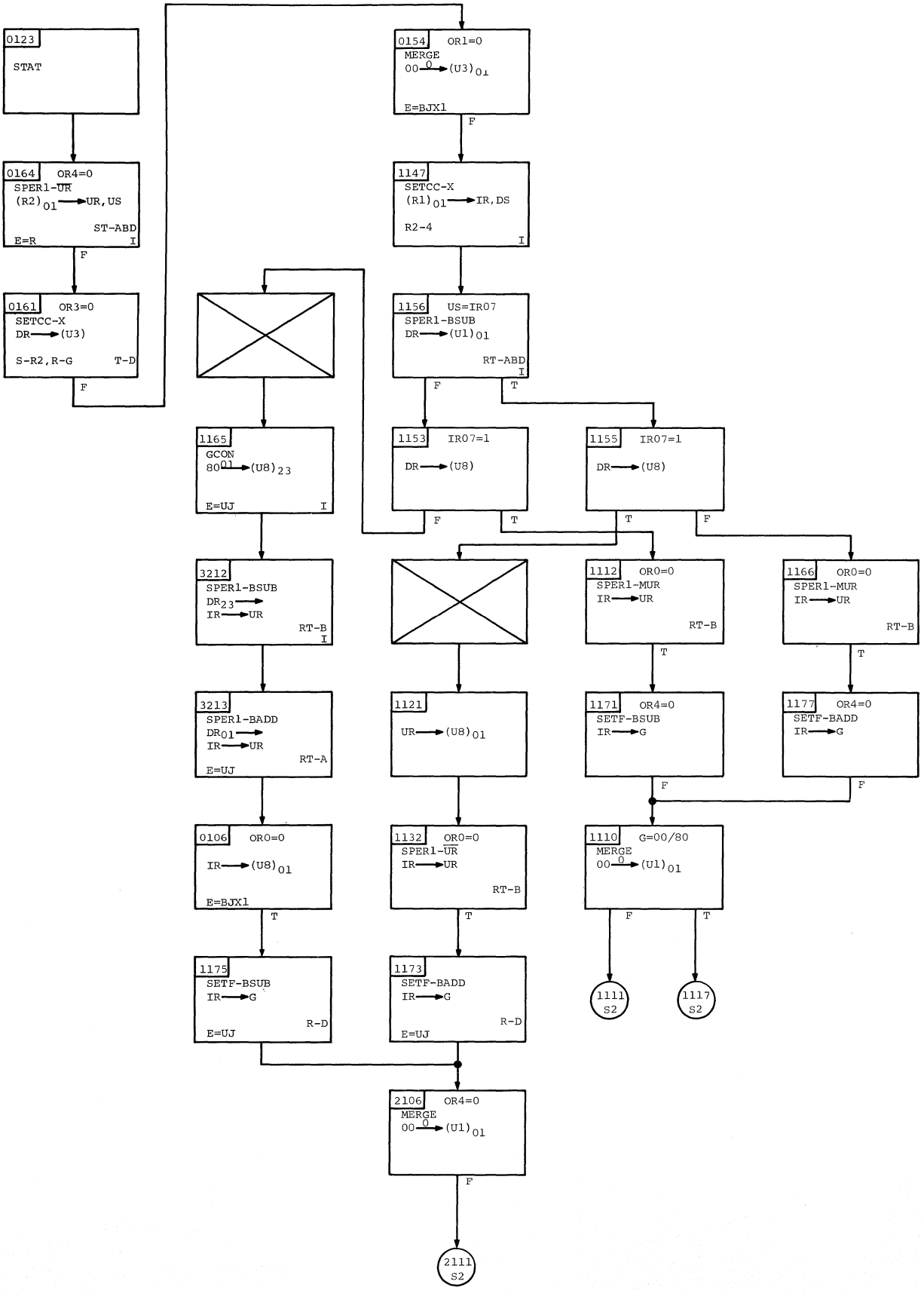
COMPARE SHORT CER RR 39



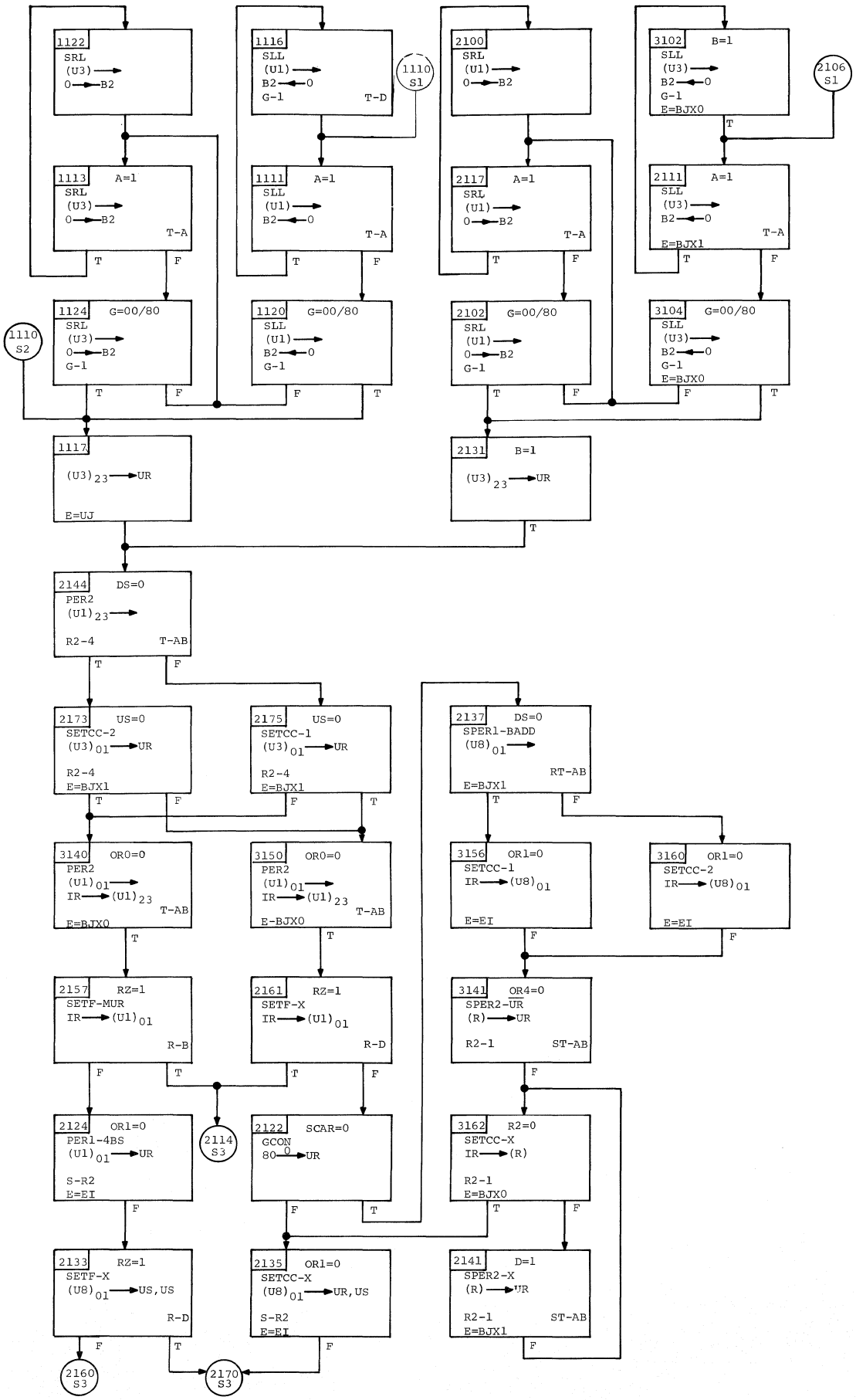
COMPARE SHORT CER RR 39



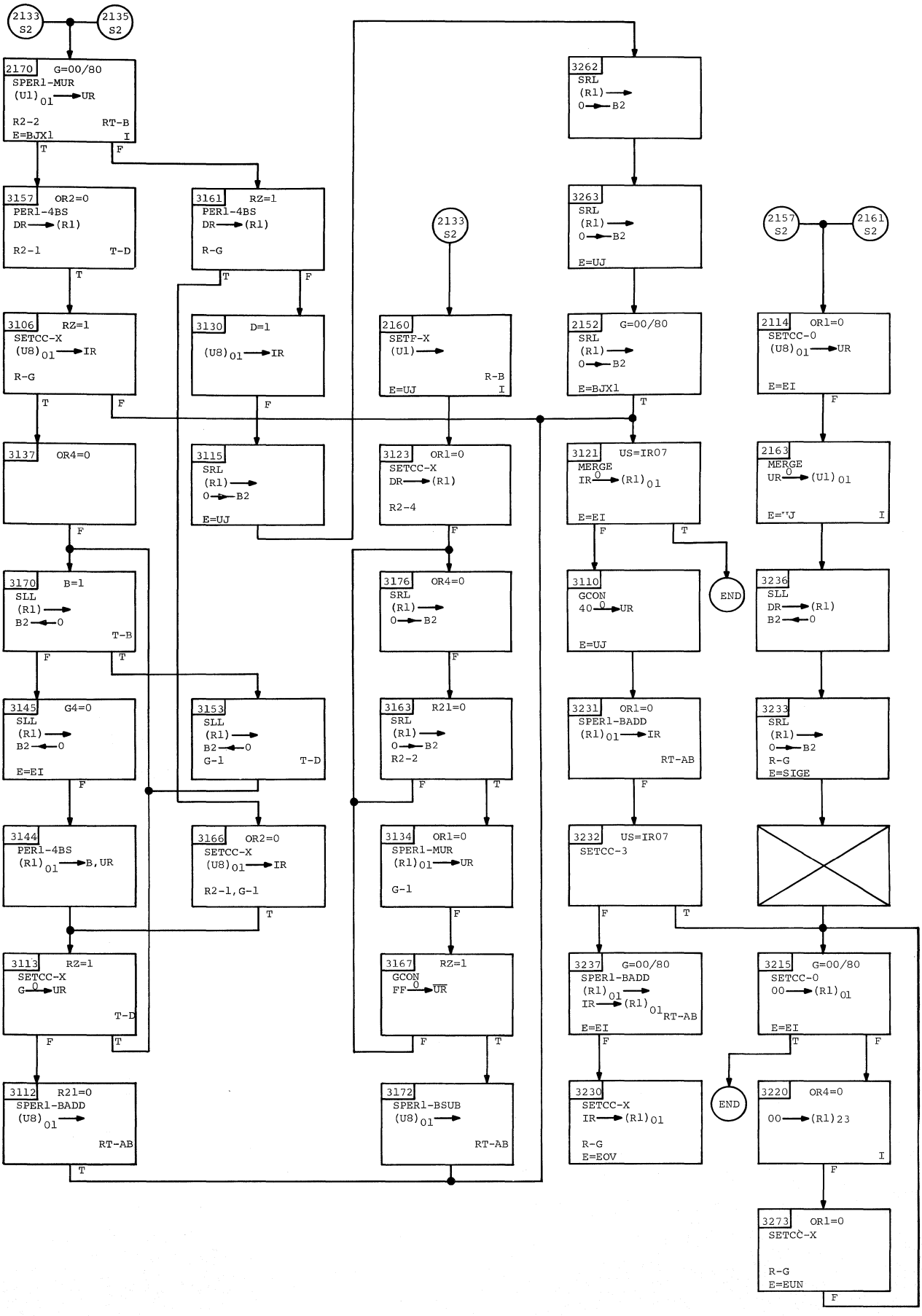
ADD NORMALIZED SHORT ANER RR 3A



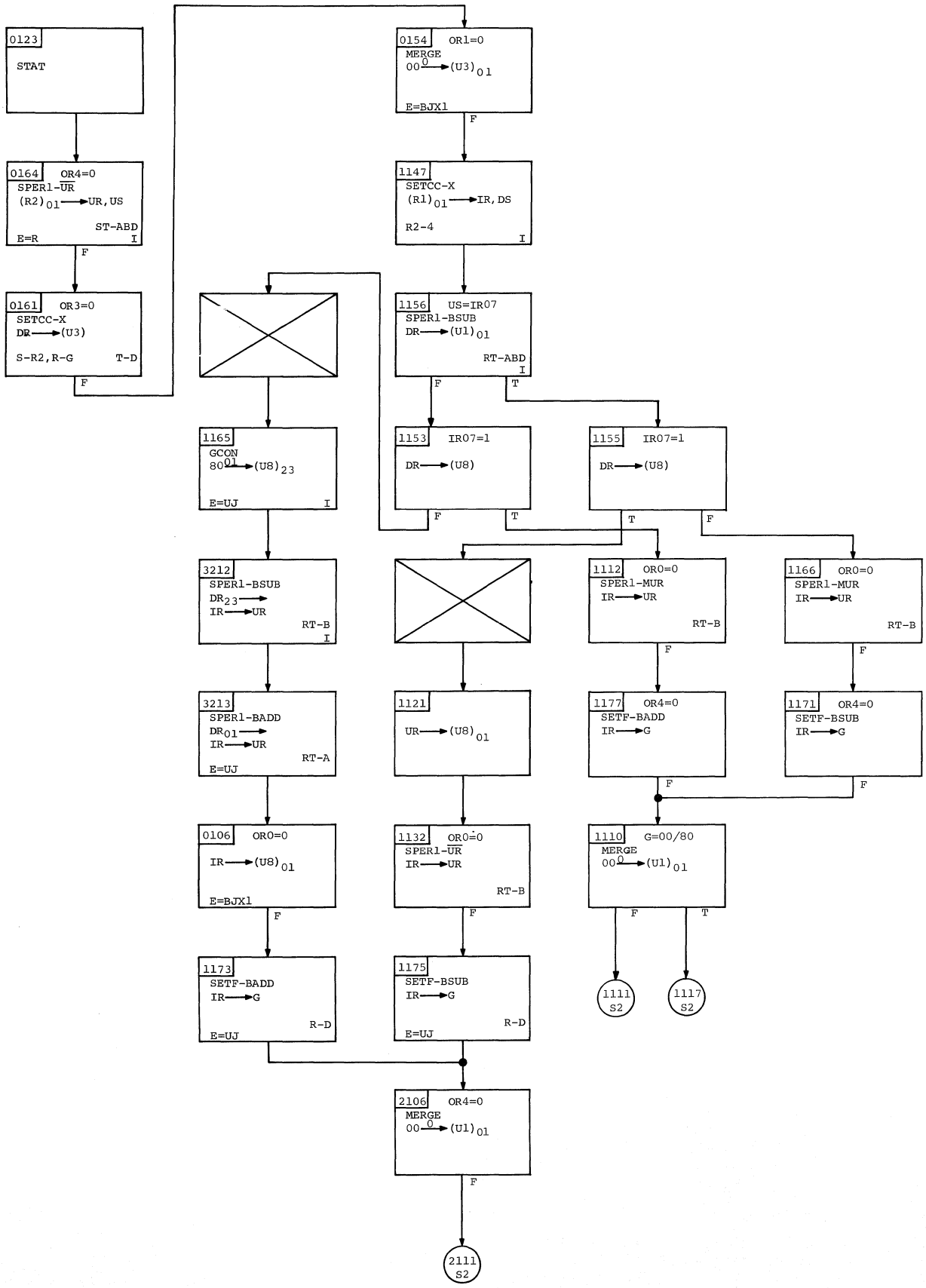
ADD NORMALIZED SHORT ANER RR 3A



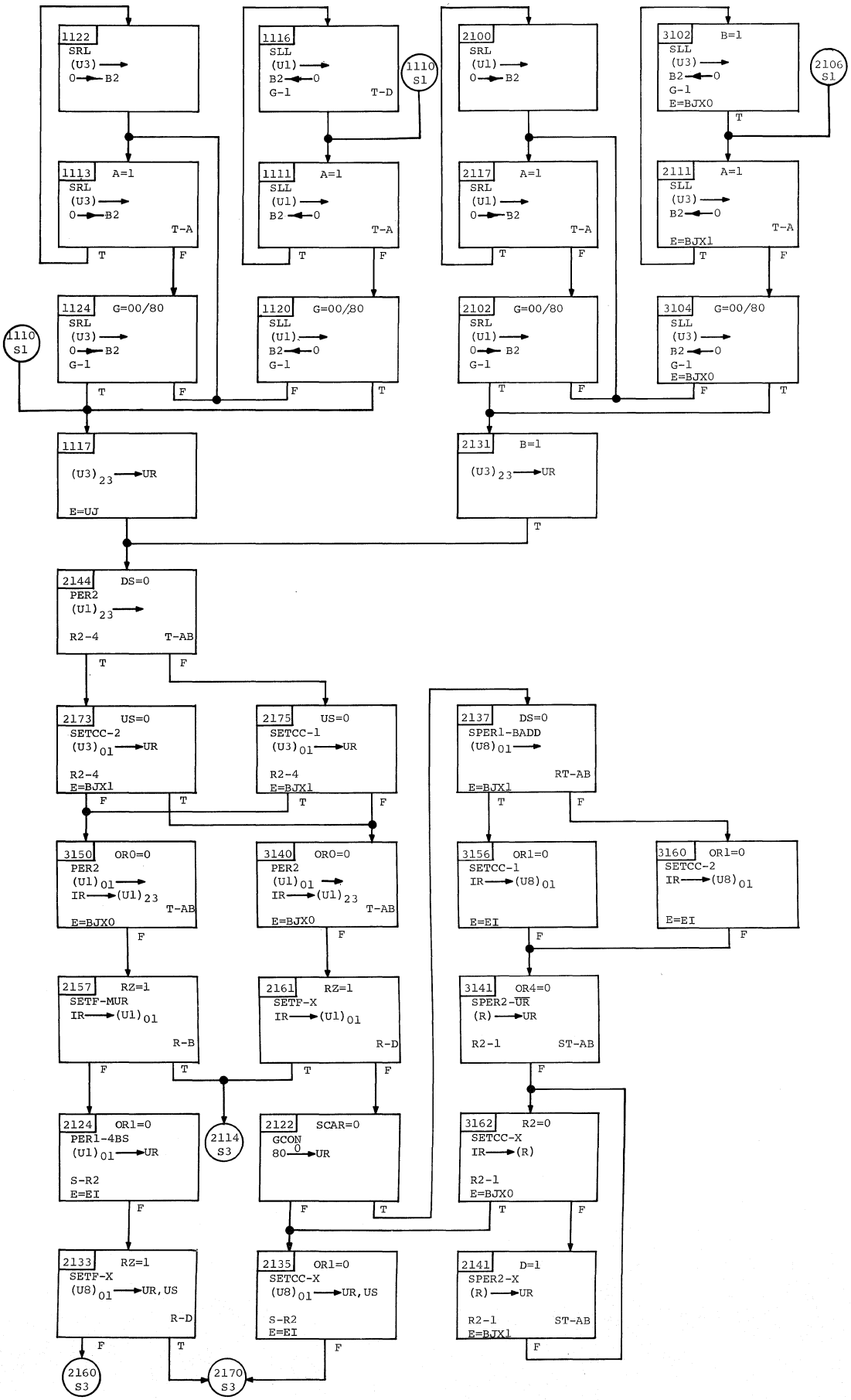
ADD NORMALIZED SHORT ANER RR 3A



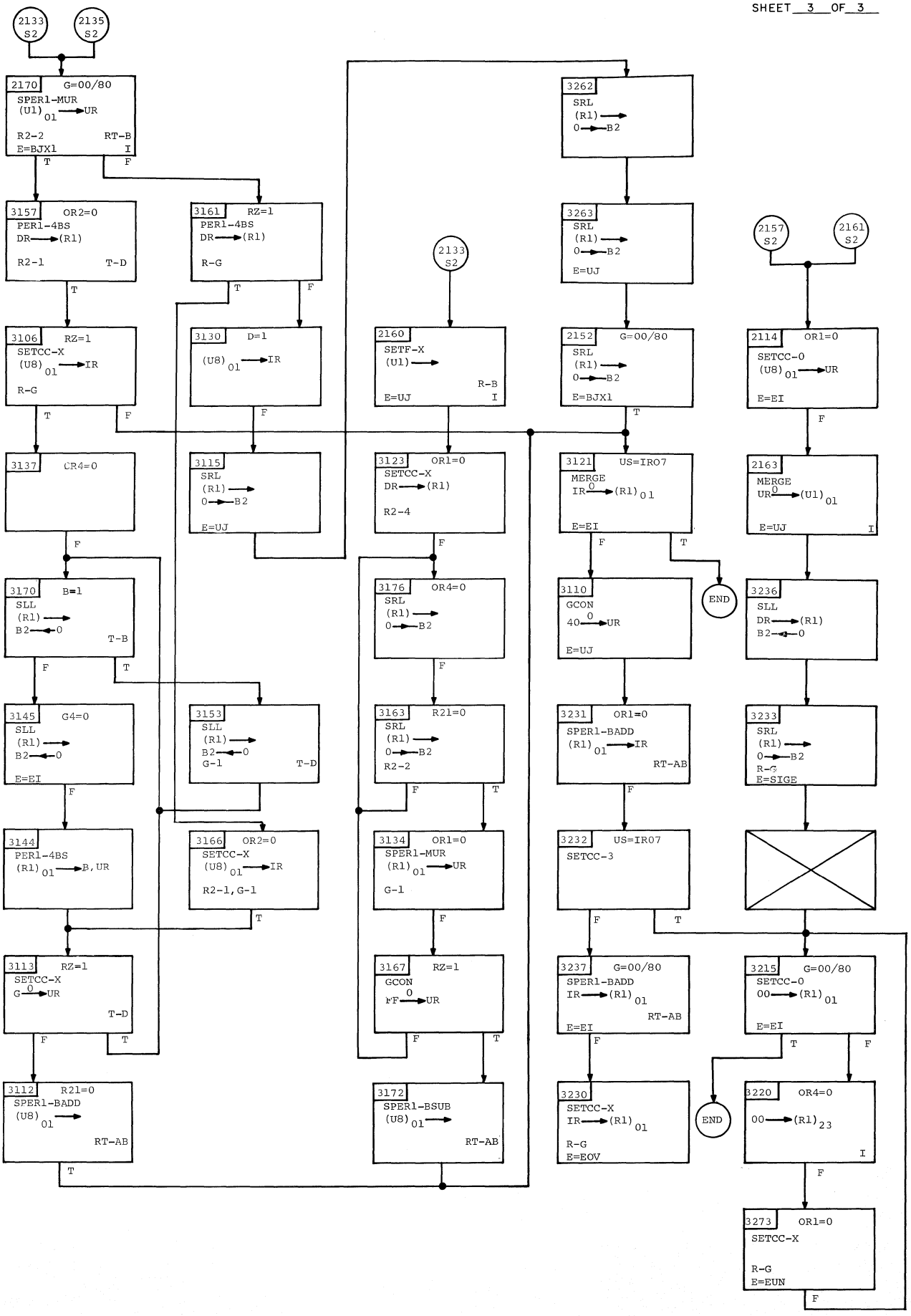
SUBTRACT NORMALIZED SHORT SNER RR 3B



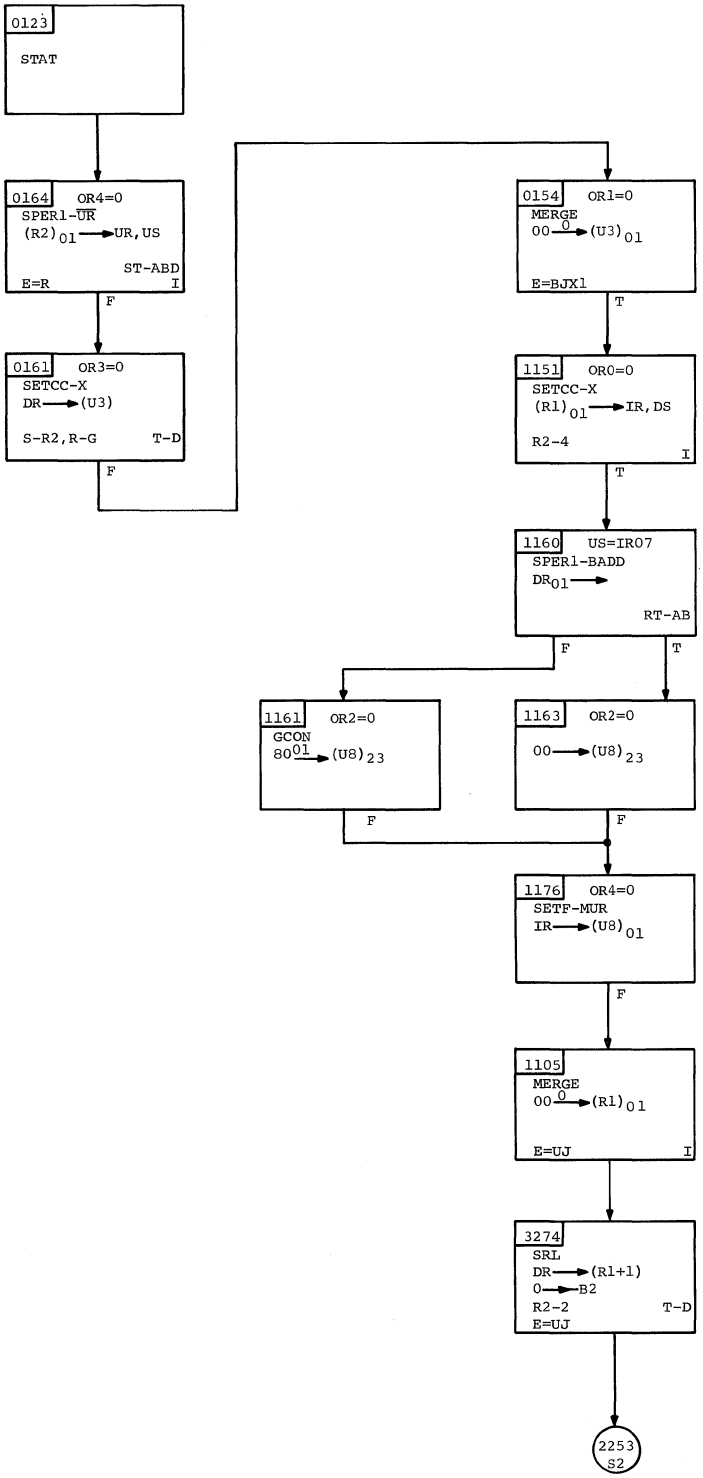
SUBTRACT NORMALIZED SHORT SNER RR 3B



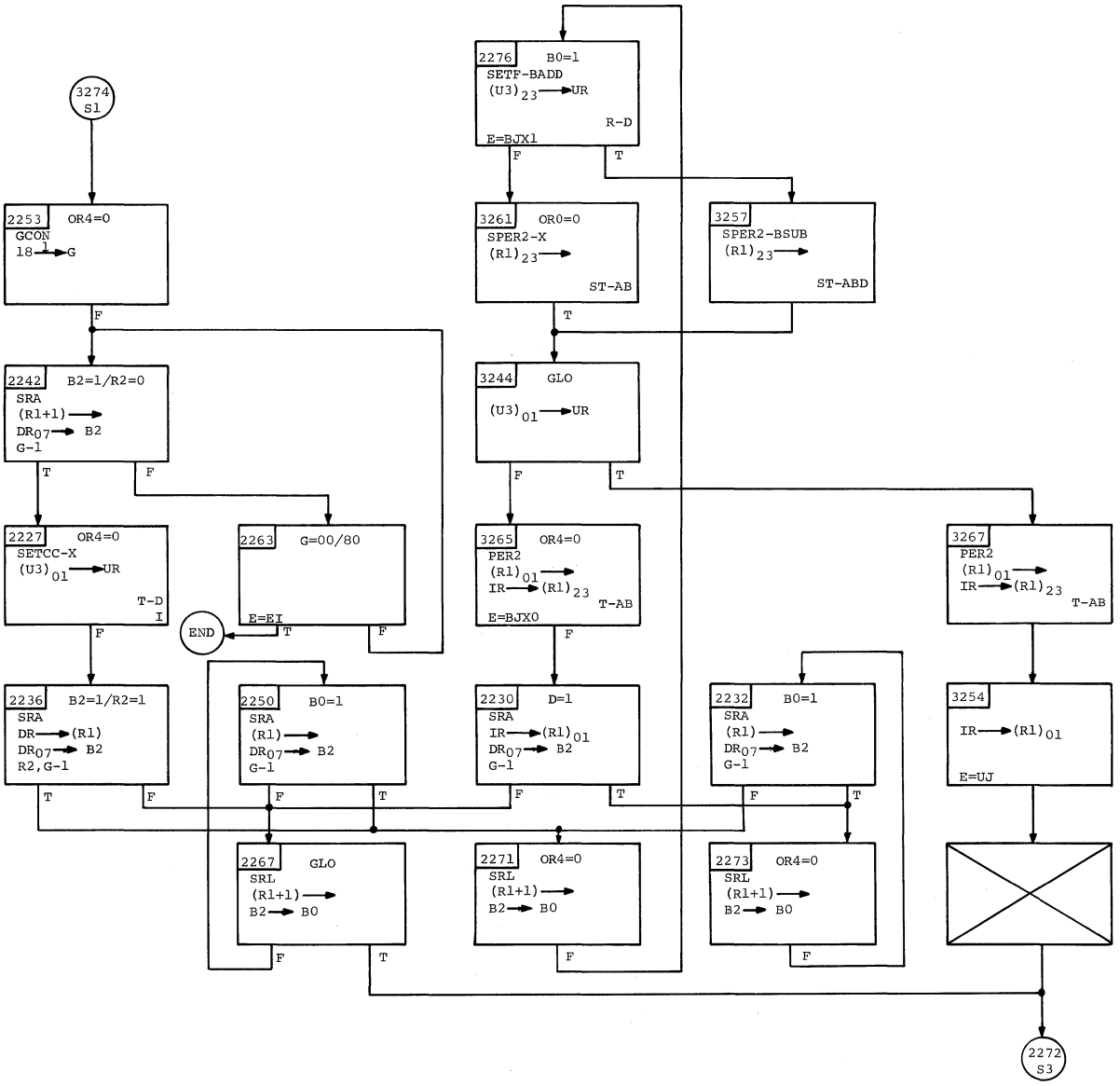
SUBTRACT NORMALIZED SHORT SNER RR 3B



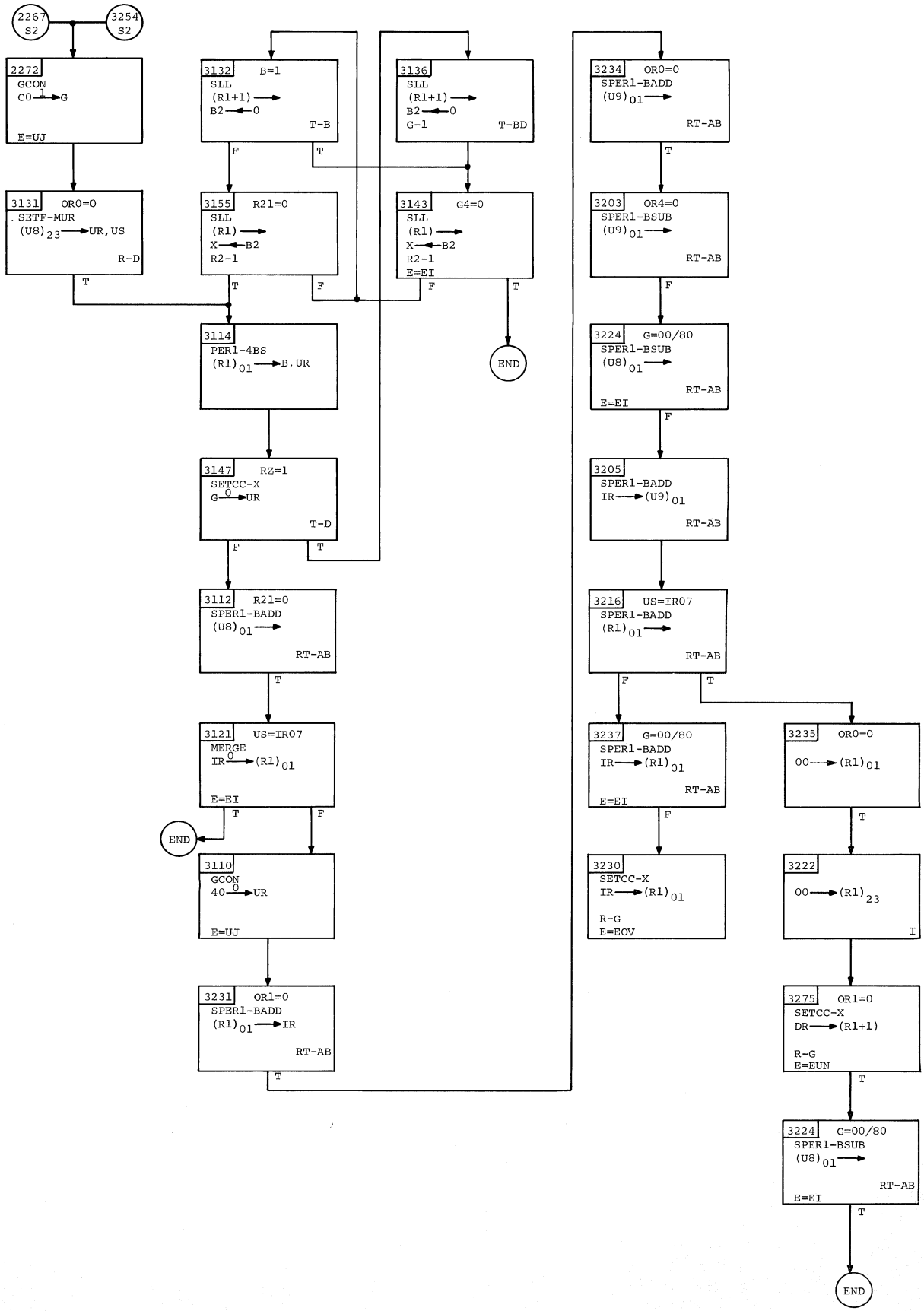
MULTIPLY SHORT MER RR 3C



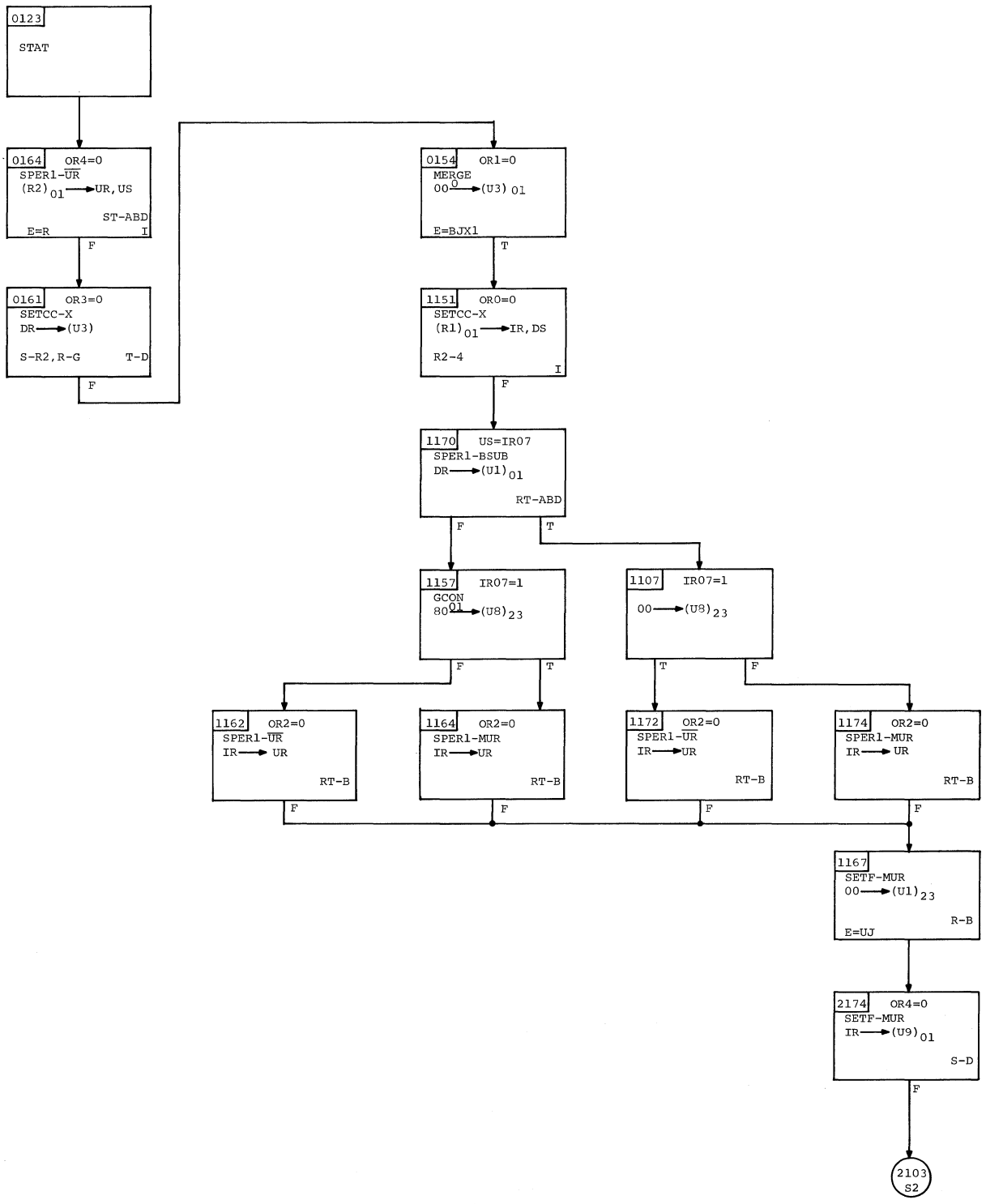
MULTIPLY SHORT MER RR 3C



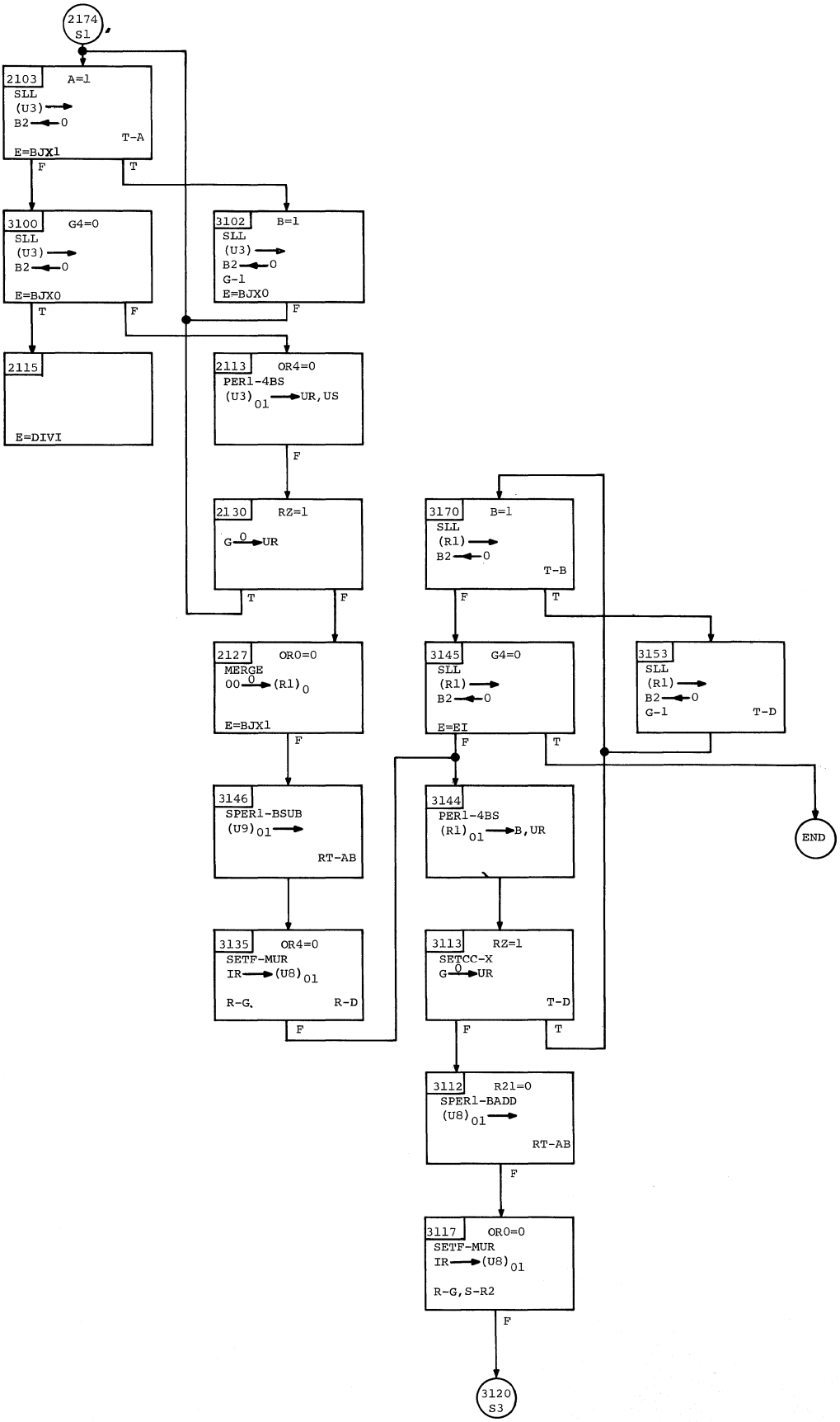
MULTIPLY SHORT MER RR 3C



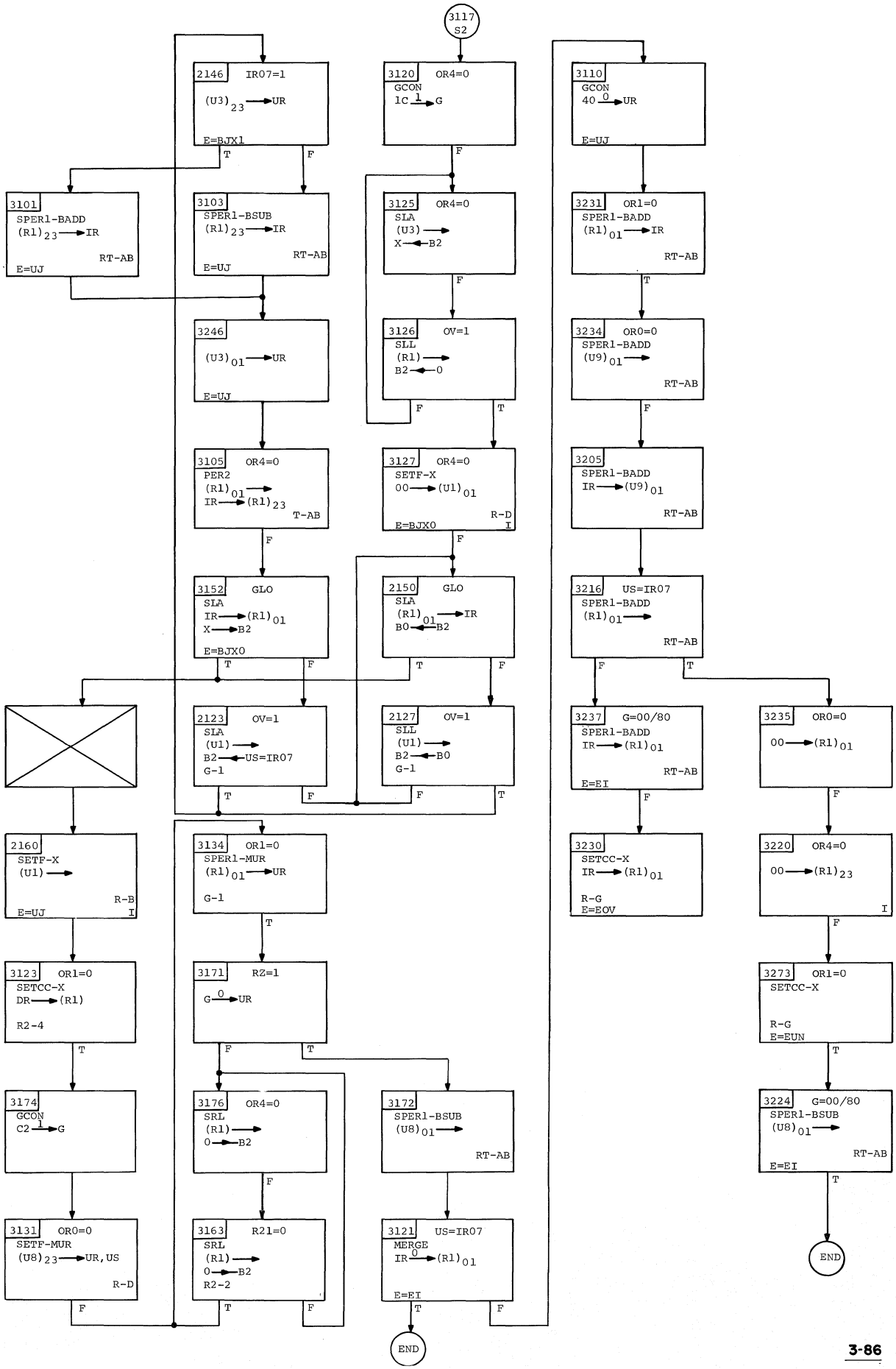
DIVIDE SHORT DER RR 3D



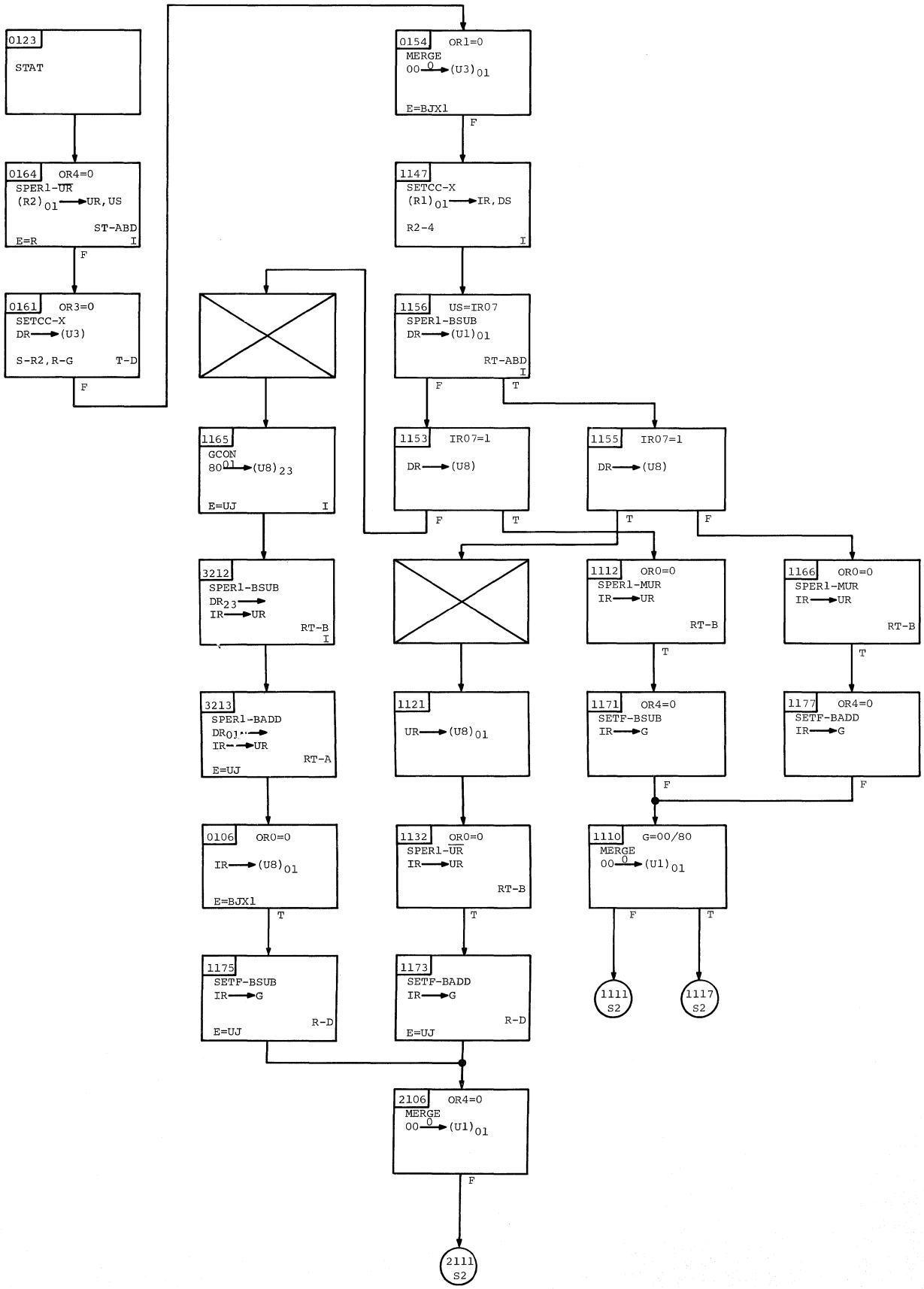
DIVIDE SHORT DER RR 3D



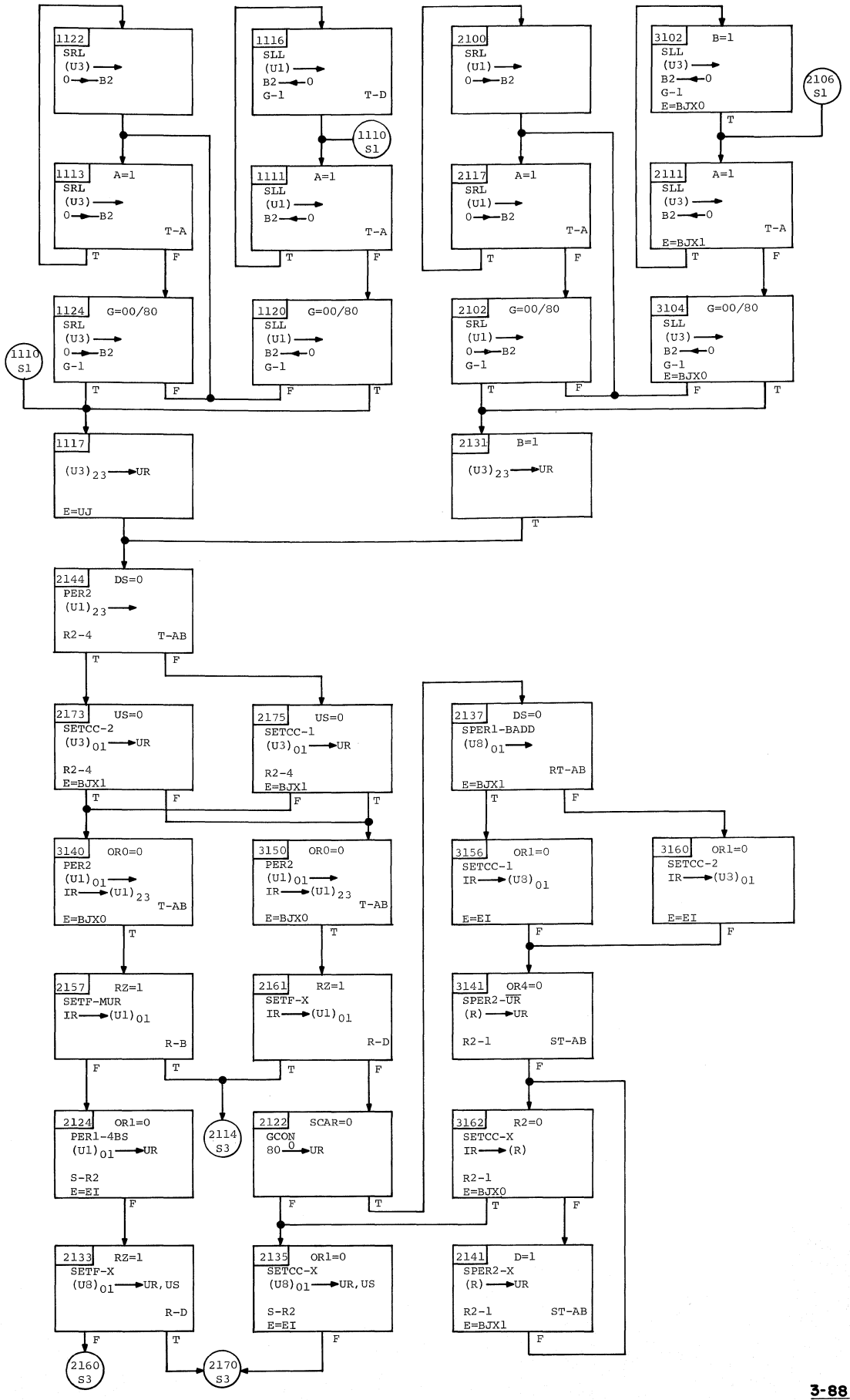
DIVIDE SHORT DER RR 3D



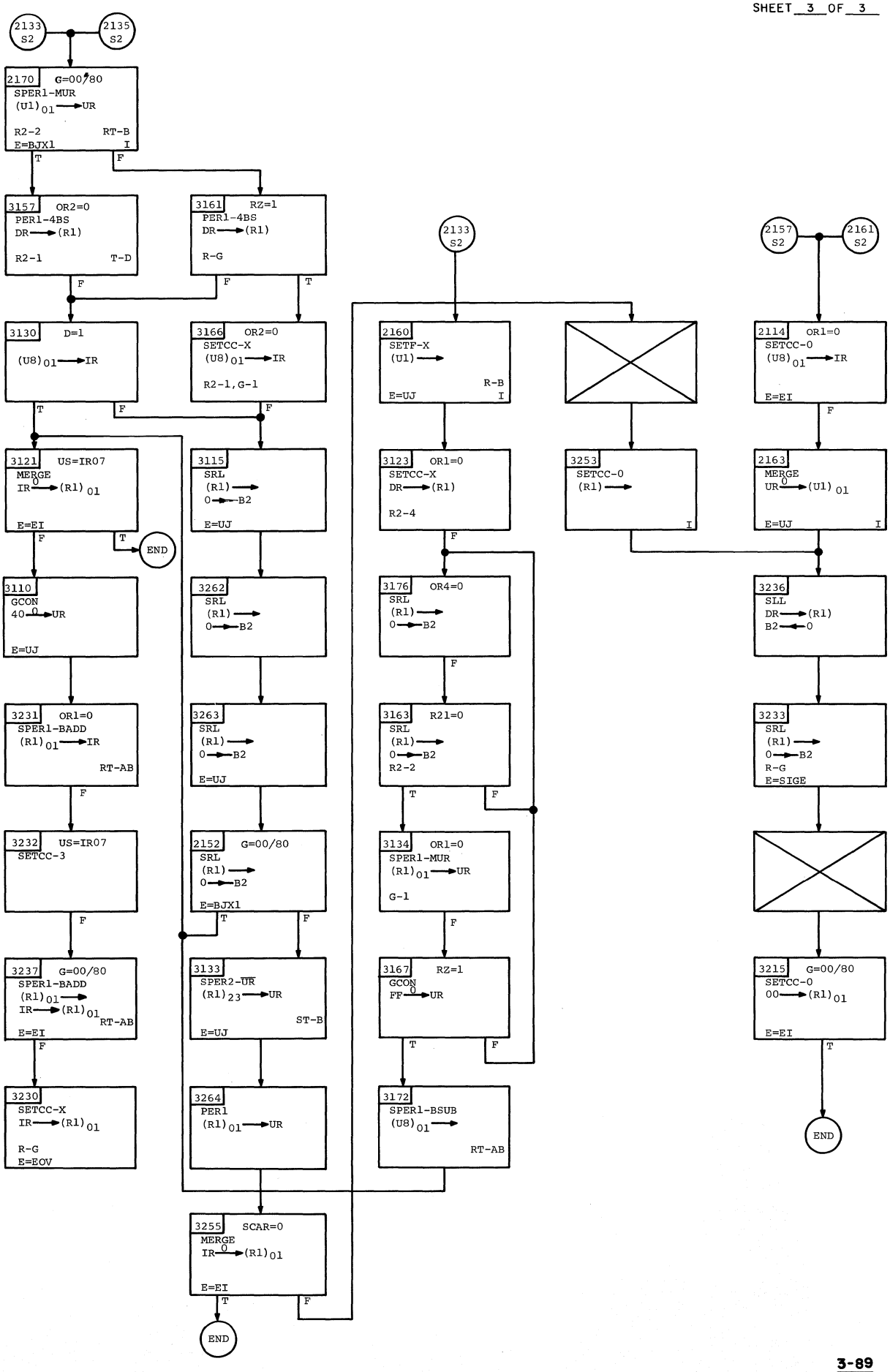
ADD UNNORMALIZED SHORT AUR RR 3E



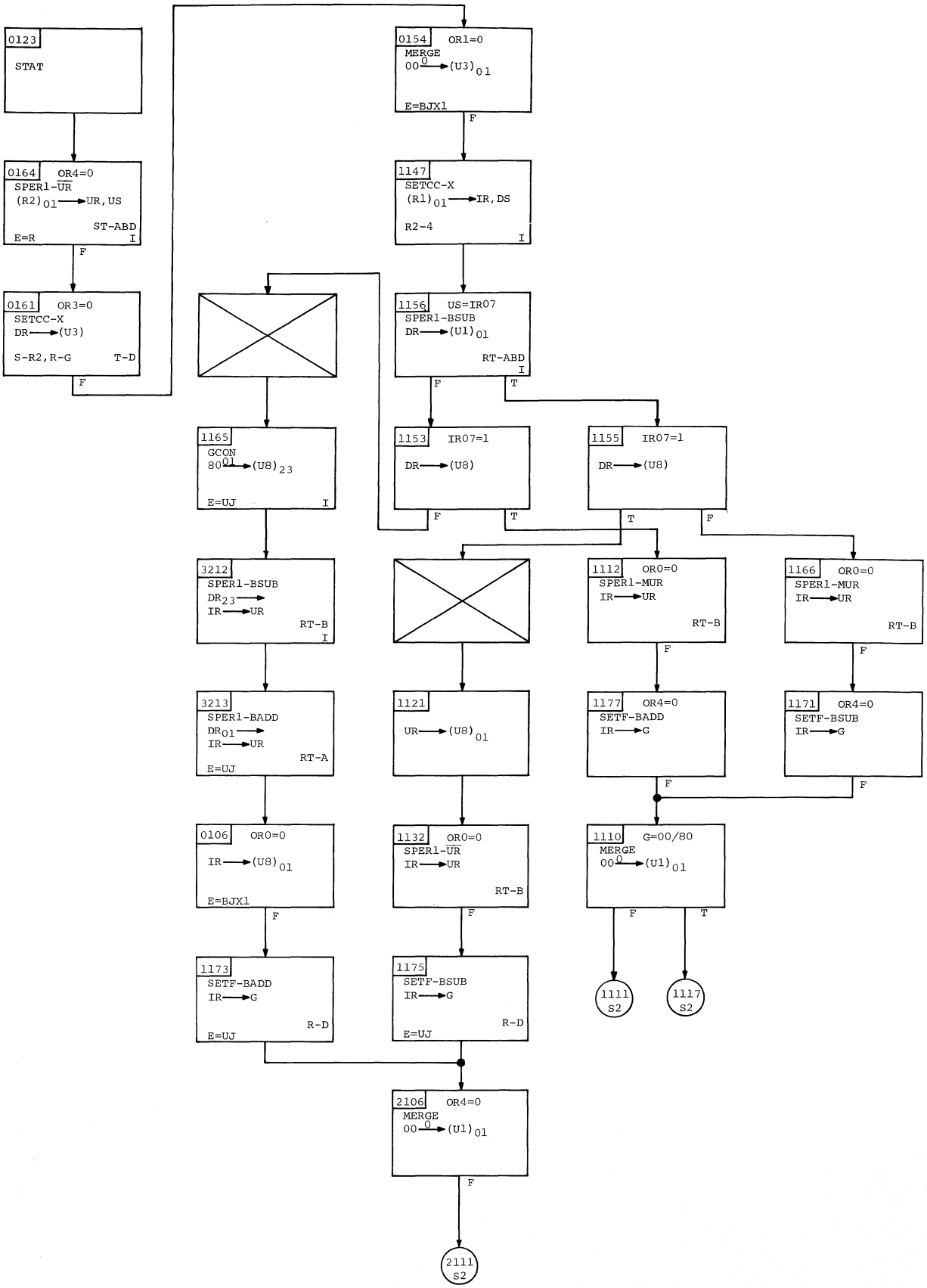
ADD UNNORMALIZED SHORT AUR RR 3E



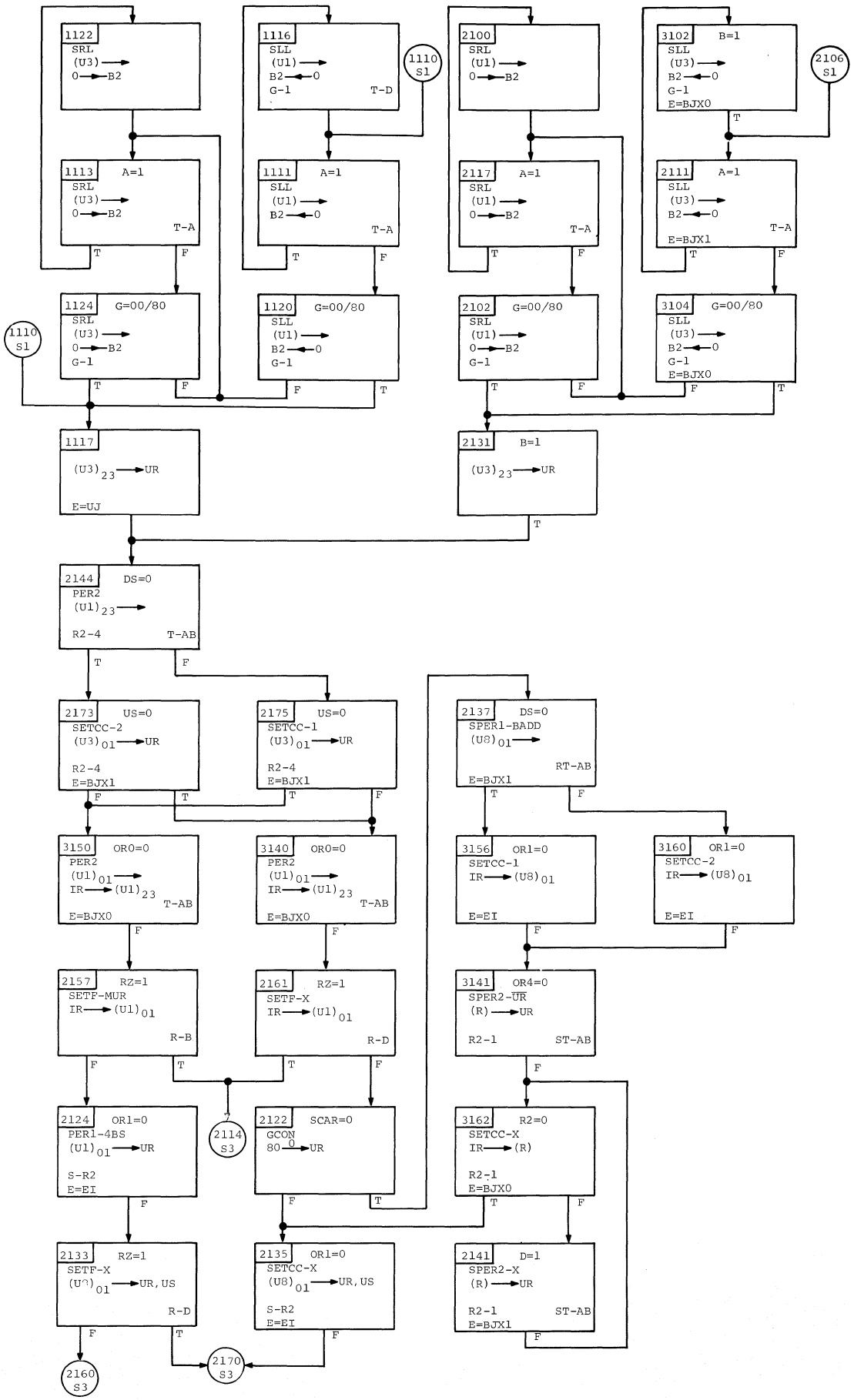
ADD UNNORMALIZED SHORT AUR RR 3E



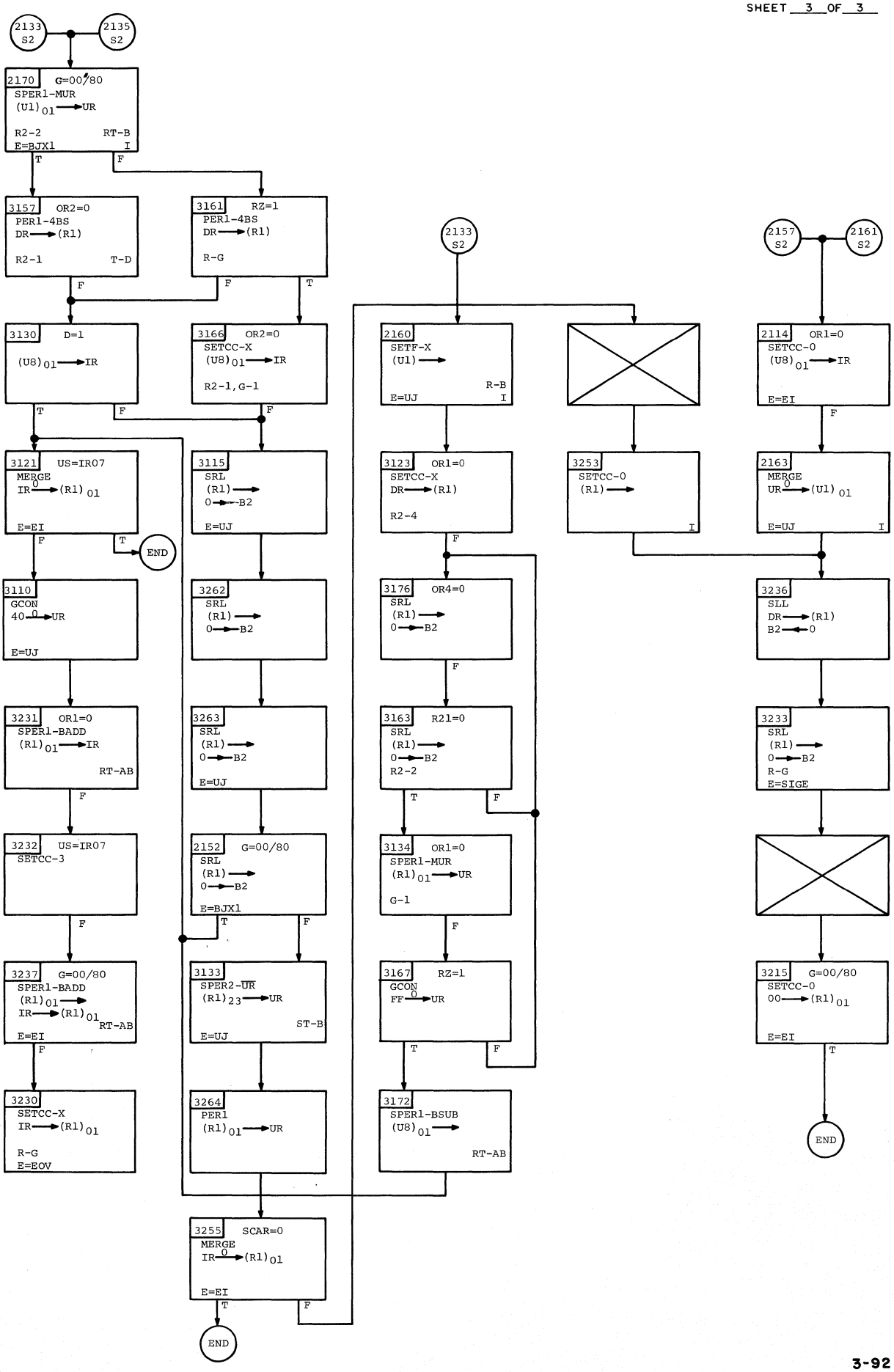
SUBTRACT UNNORMALIZED SHORT SUR RR 3F



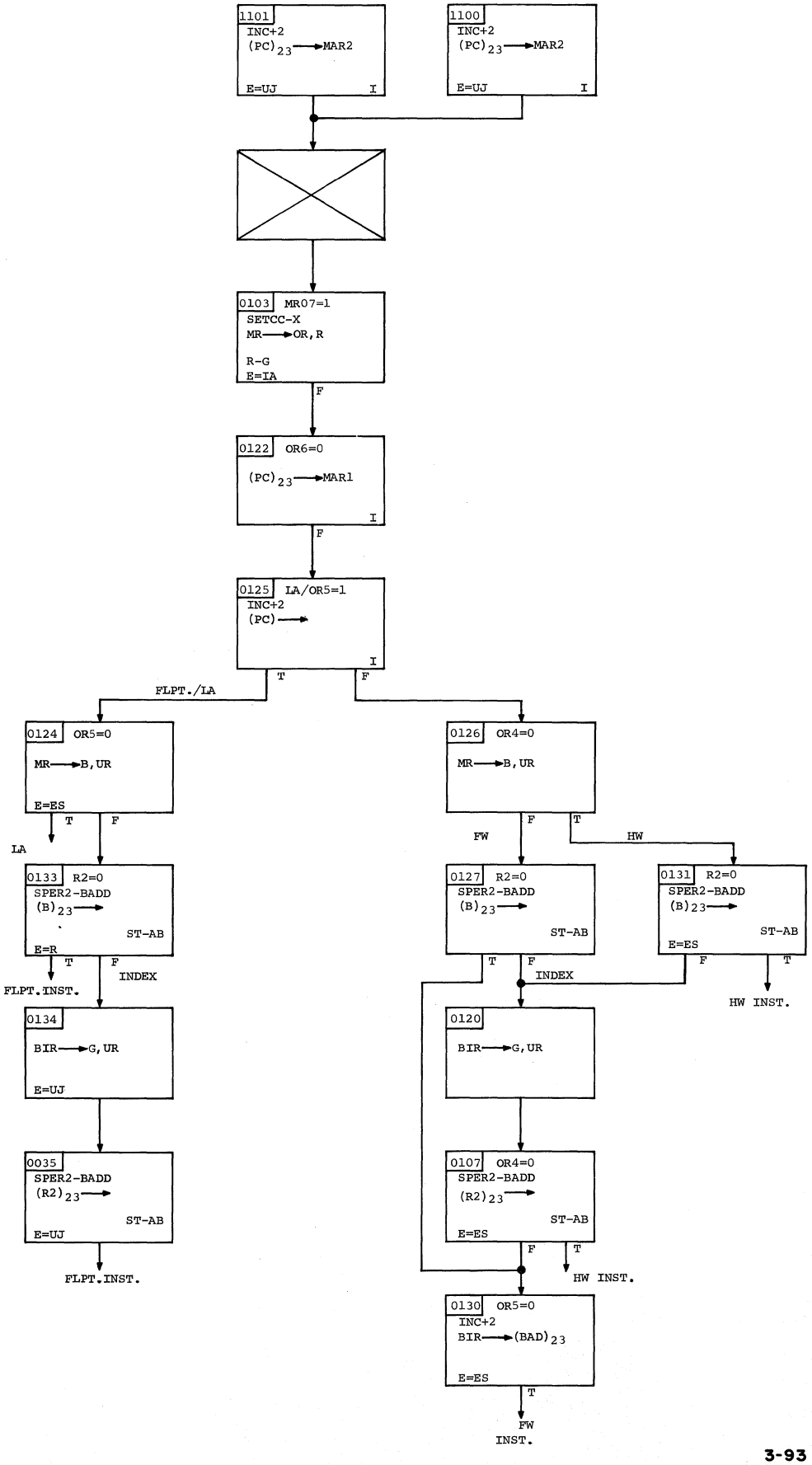
SUBTRACT UNNORMALIZED SHORT SUR RR 3F



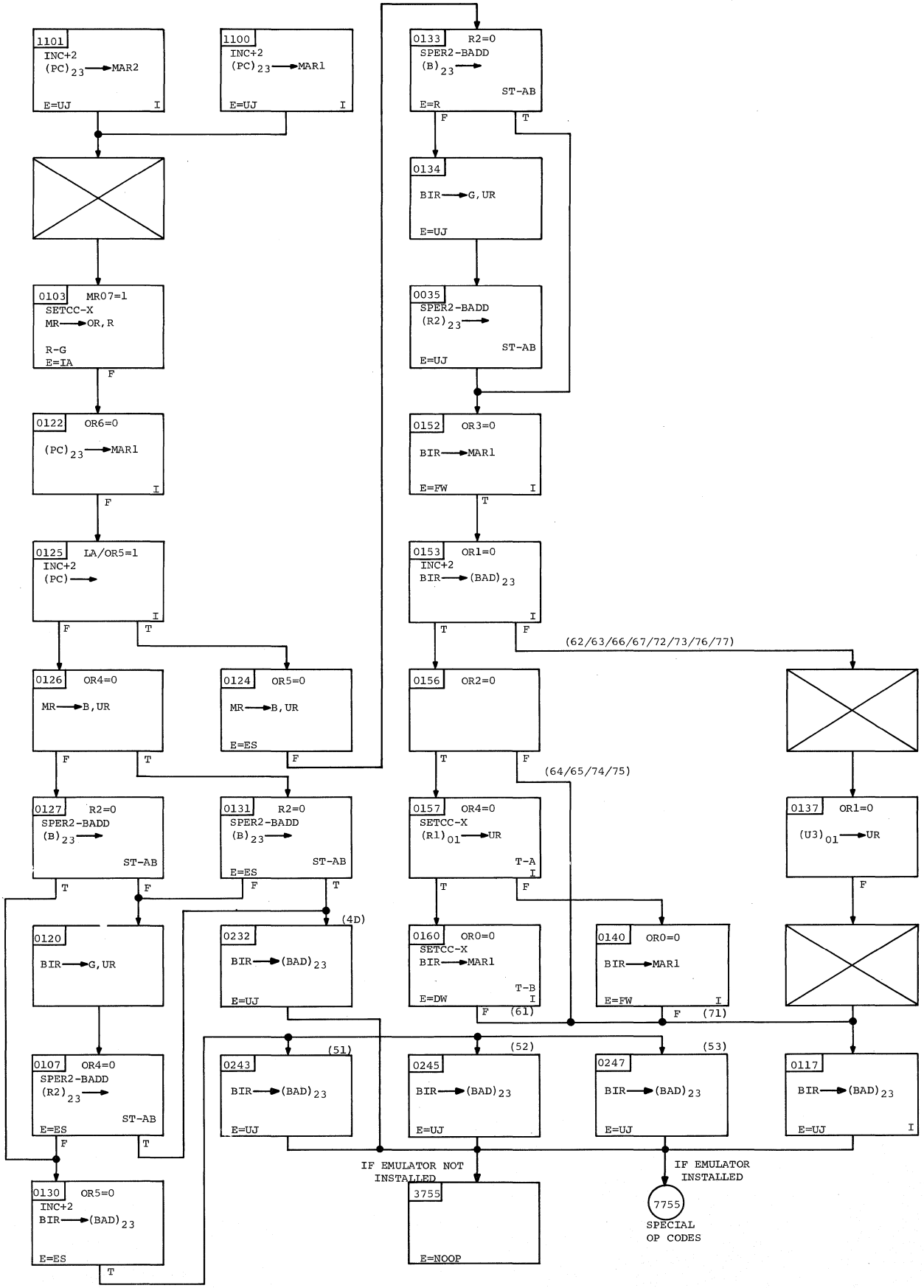
SUBTRACT UNNORMALIZED SHORT SUR RR 3F



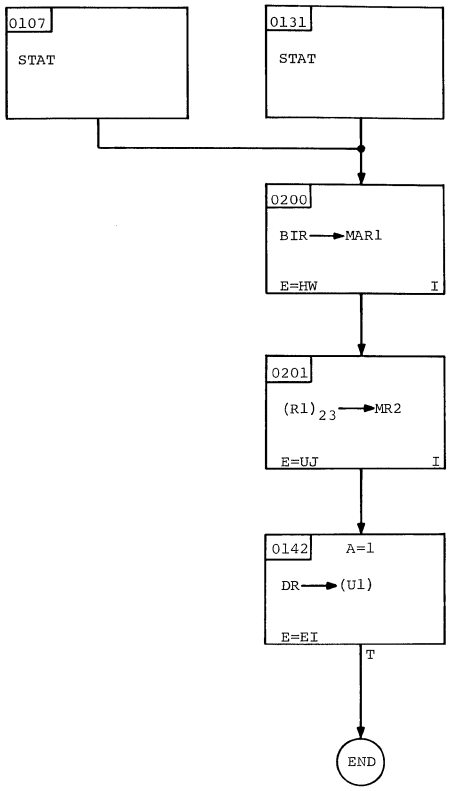
STATICIZING RX FORMAT



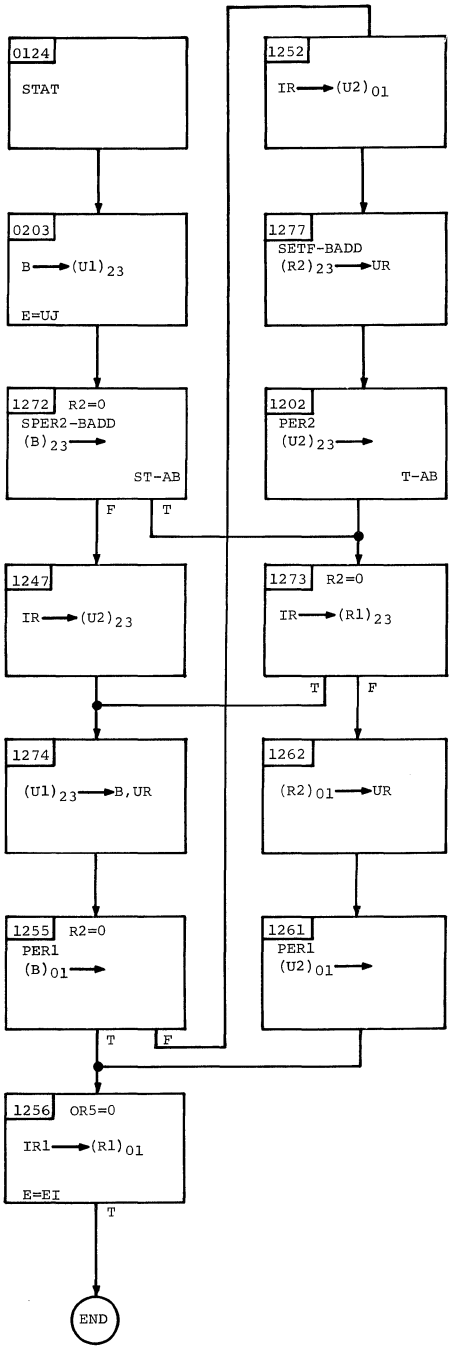
NO OPERATION INSTRUCTIONS RX FORMAT



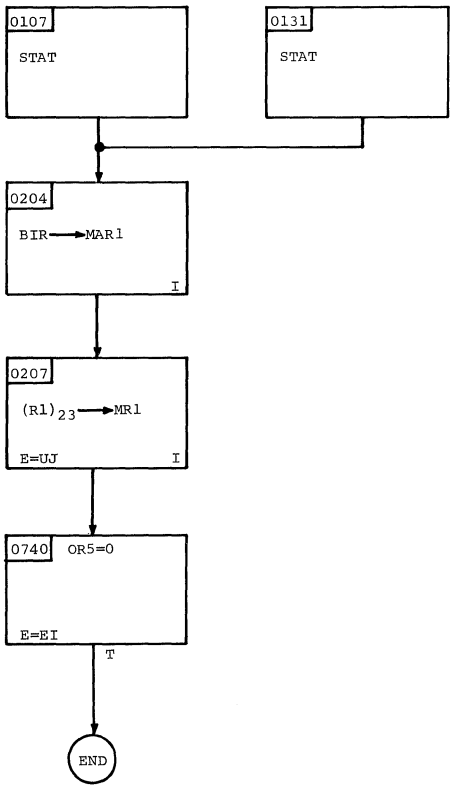
STORE HALFWORD STH RX 40



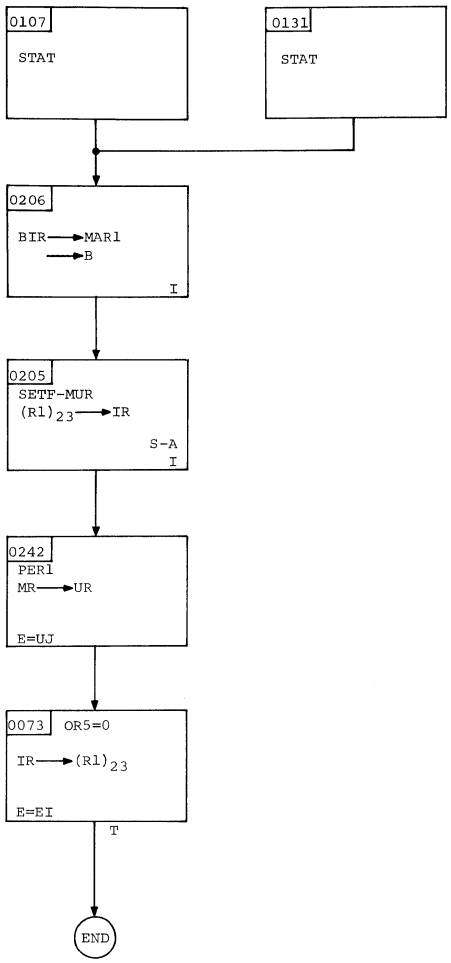
LOAD ADDRESS LA RX 41



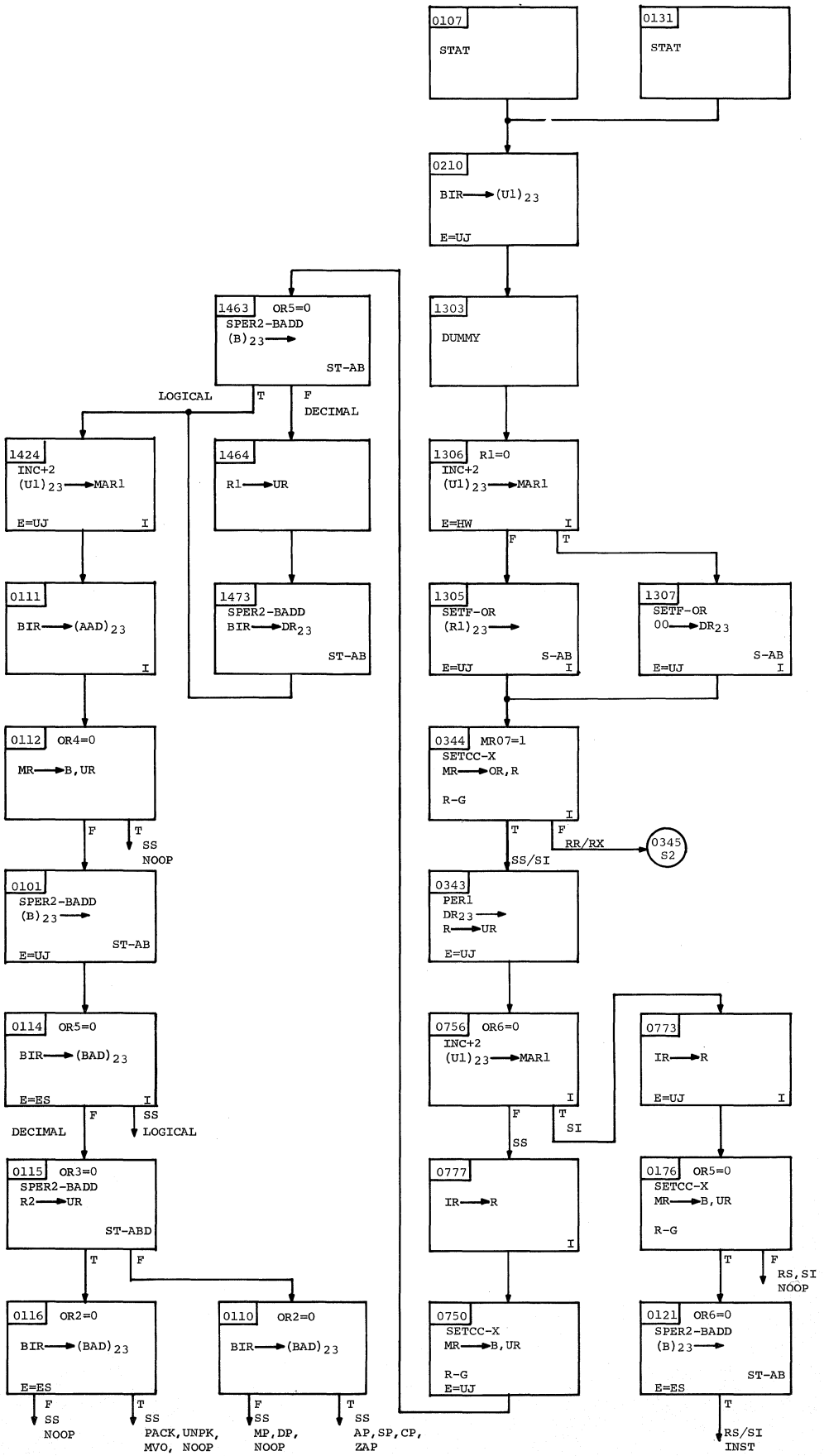
STORE CHARACTER STC RX 42



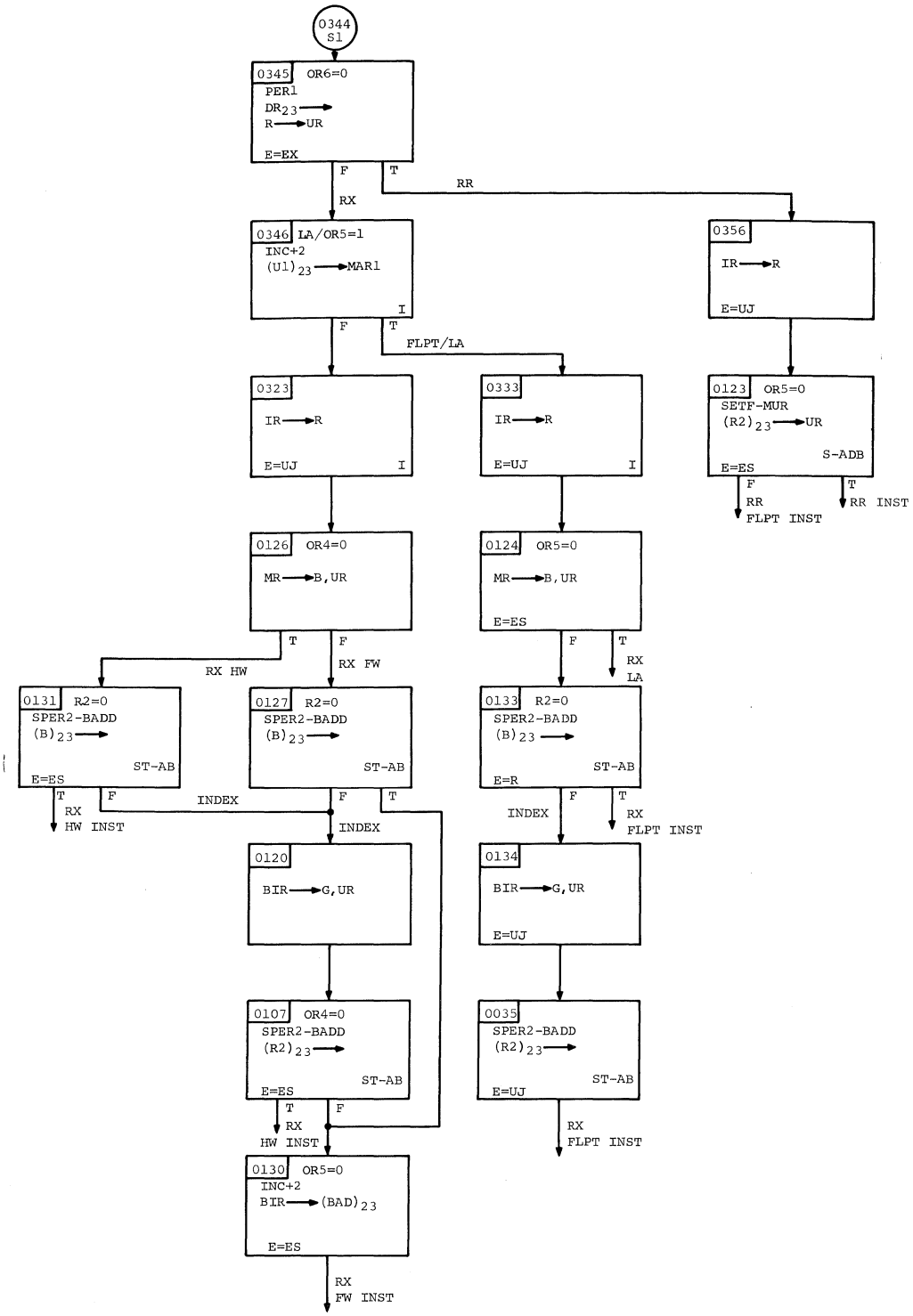
INSERT CHARACTER IC RX 43



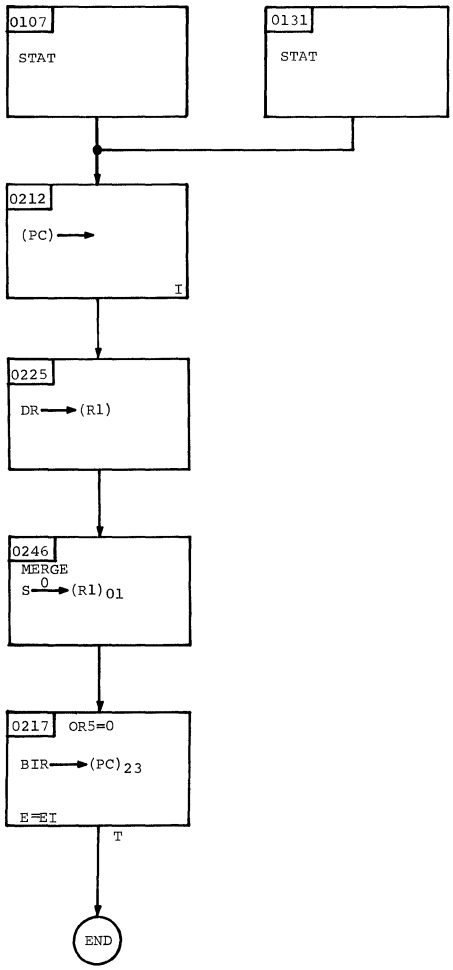
EXECUTE EX RX 44



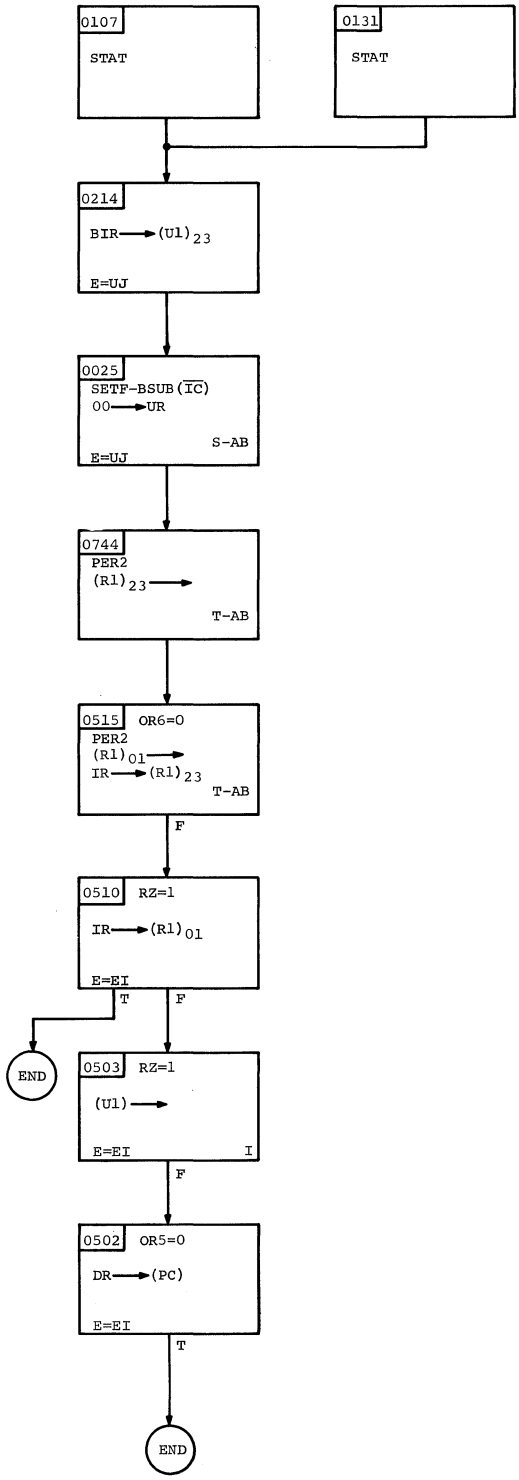
EXECUTE EX RX 44



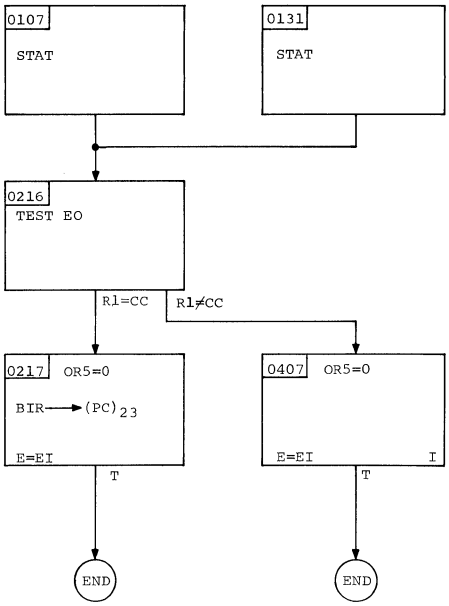
BRANCH AND LINK BAL RX 45



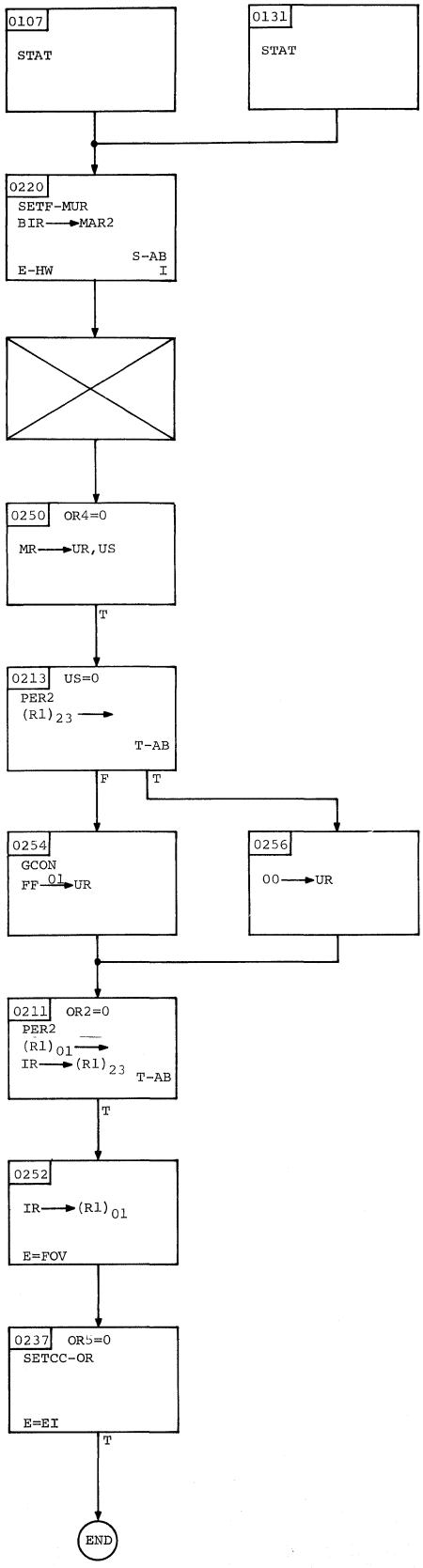
BRANCH ON COUNT BCT RX 46



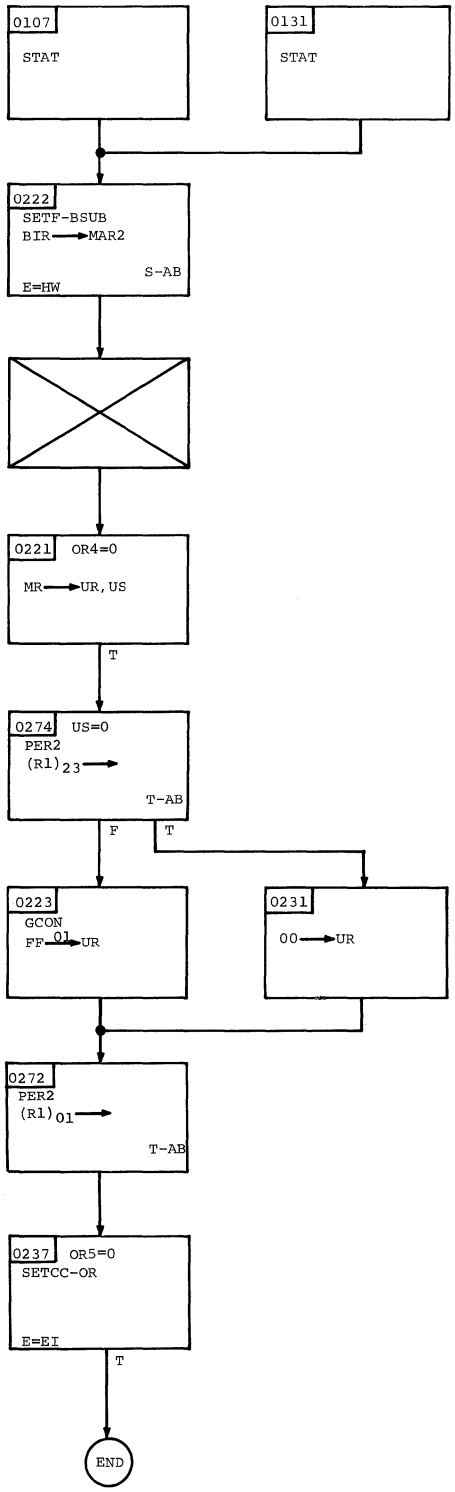
BRANCH ON CONDITION BC RX 47



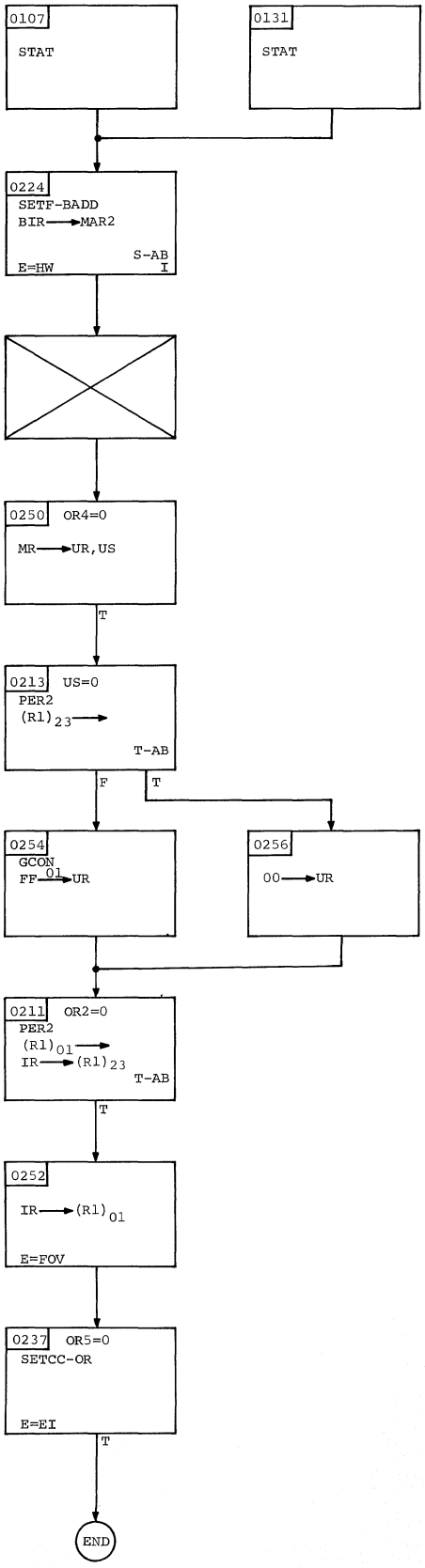
LOAD HALFWORD LH RX 48



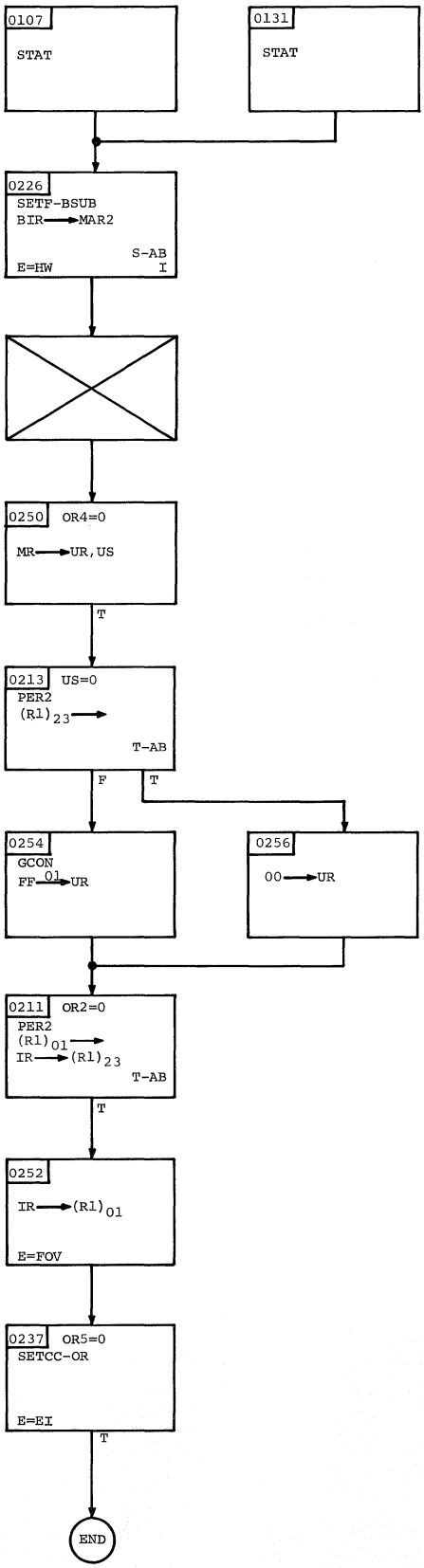
COMPARE HALFWORD CH RX 49



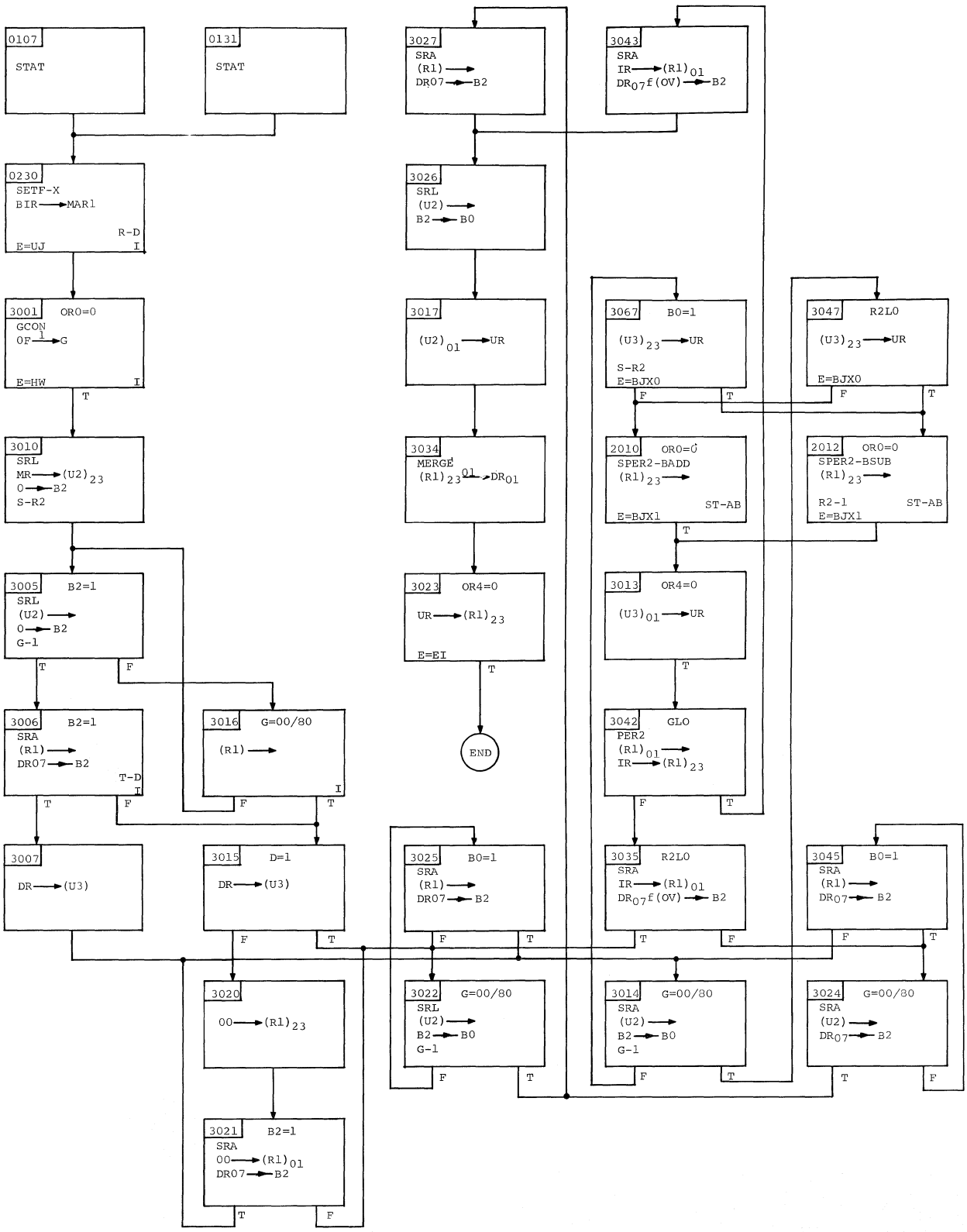
ADD HALFWORD AH RX 4A



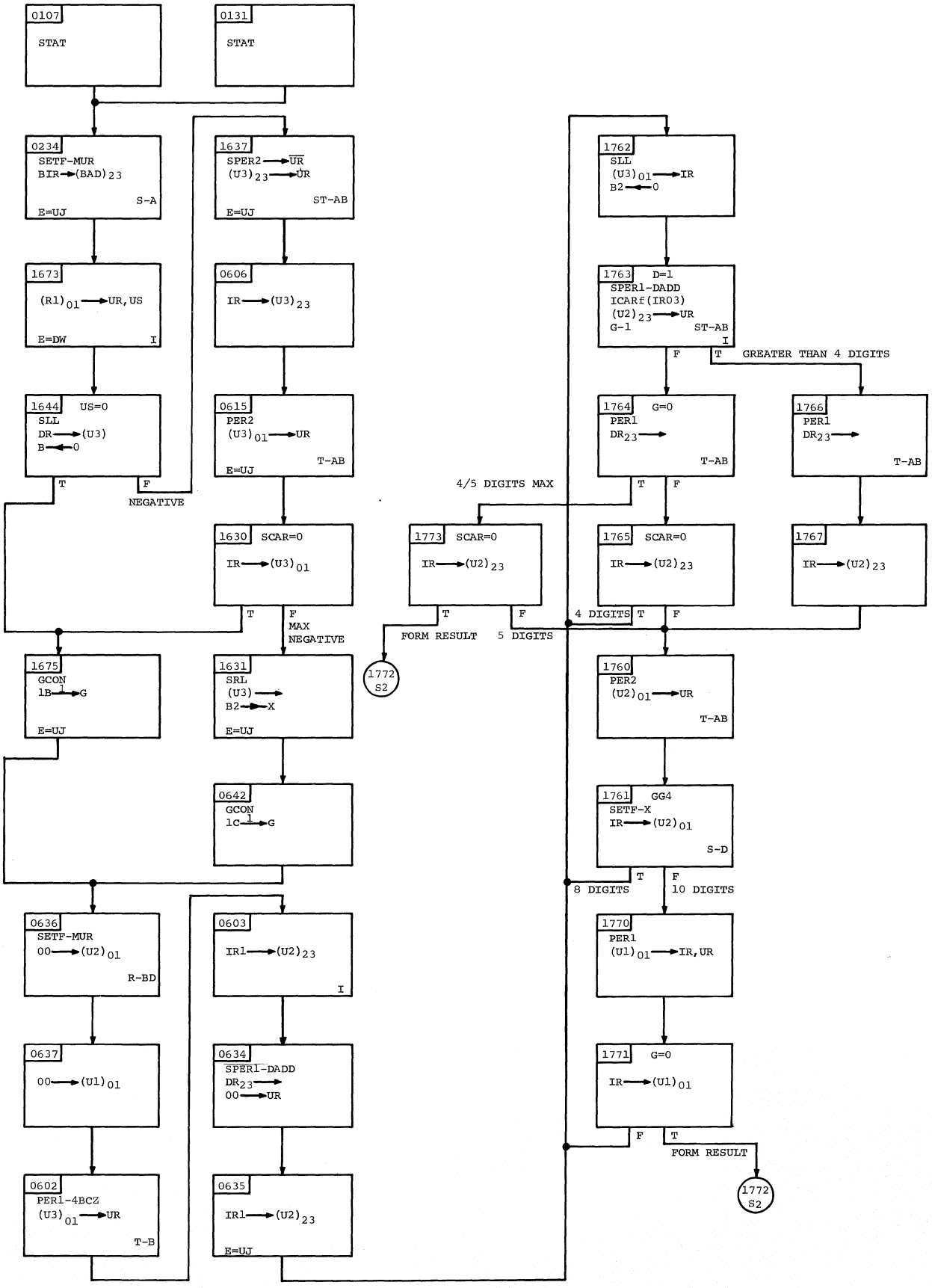
SUBTRACT HALFWORD SH RX 4B



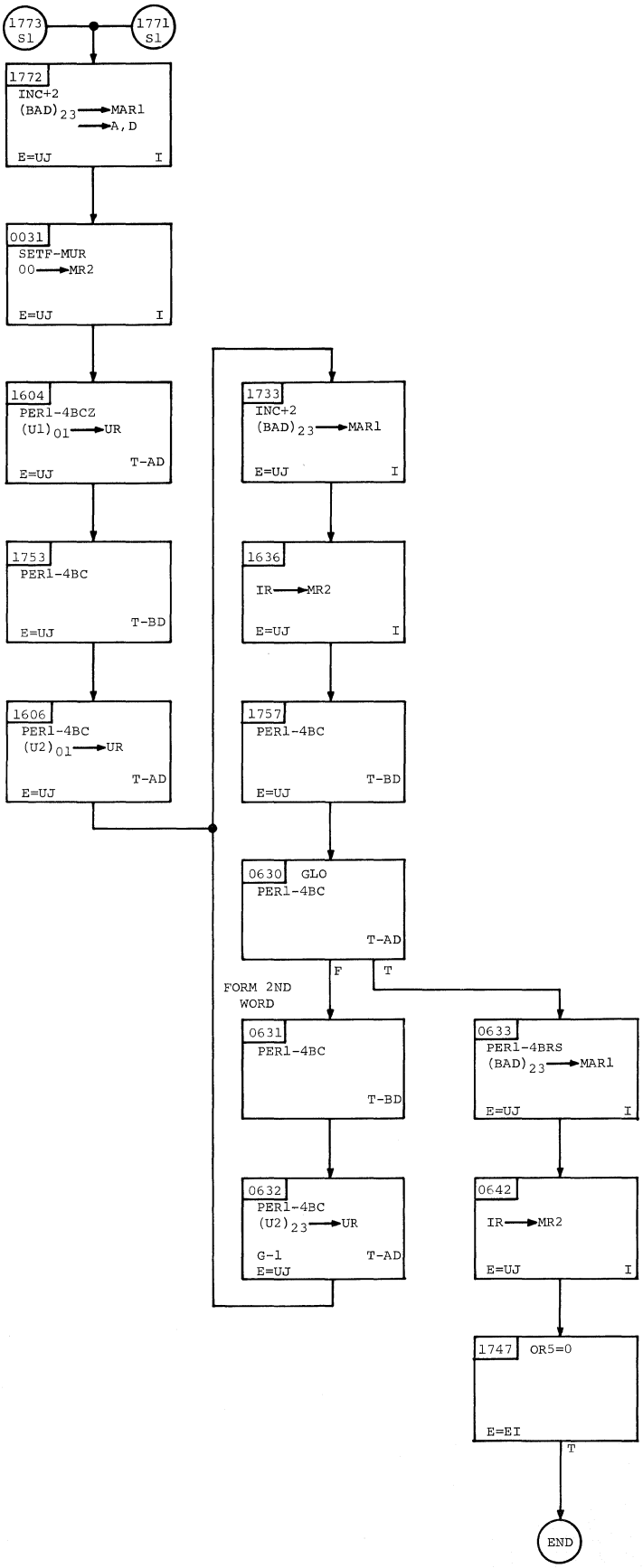
MULTIPLY HALFWORD MH RX 4C



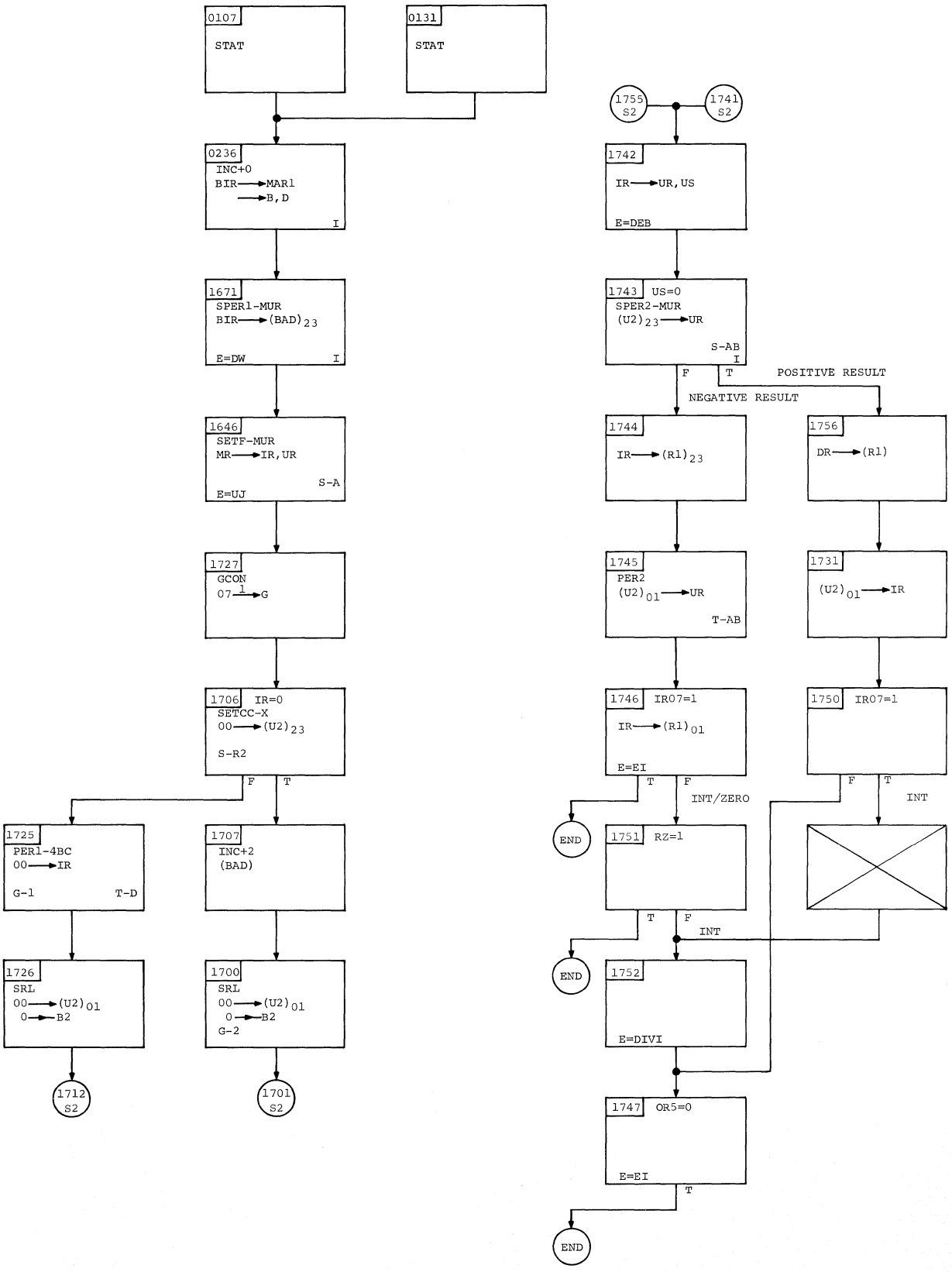
CONVERT TO DECIMAL CVD RX 4E



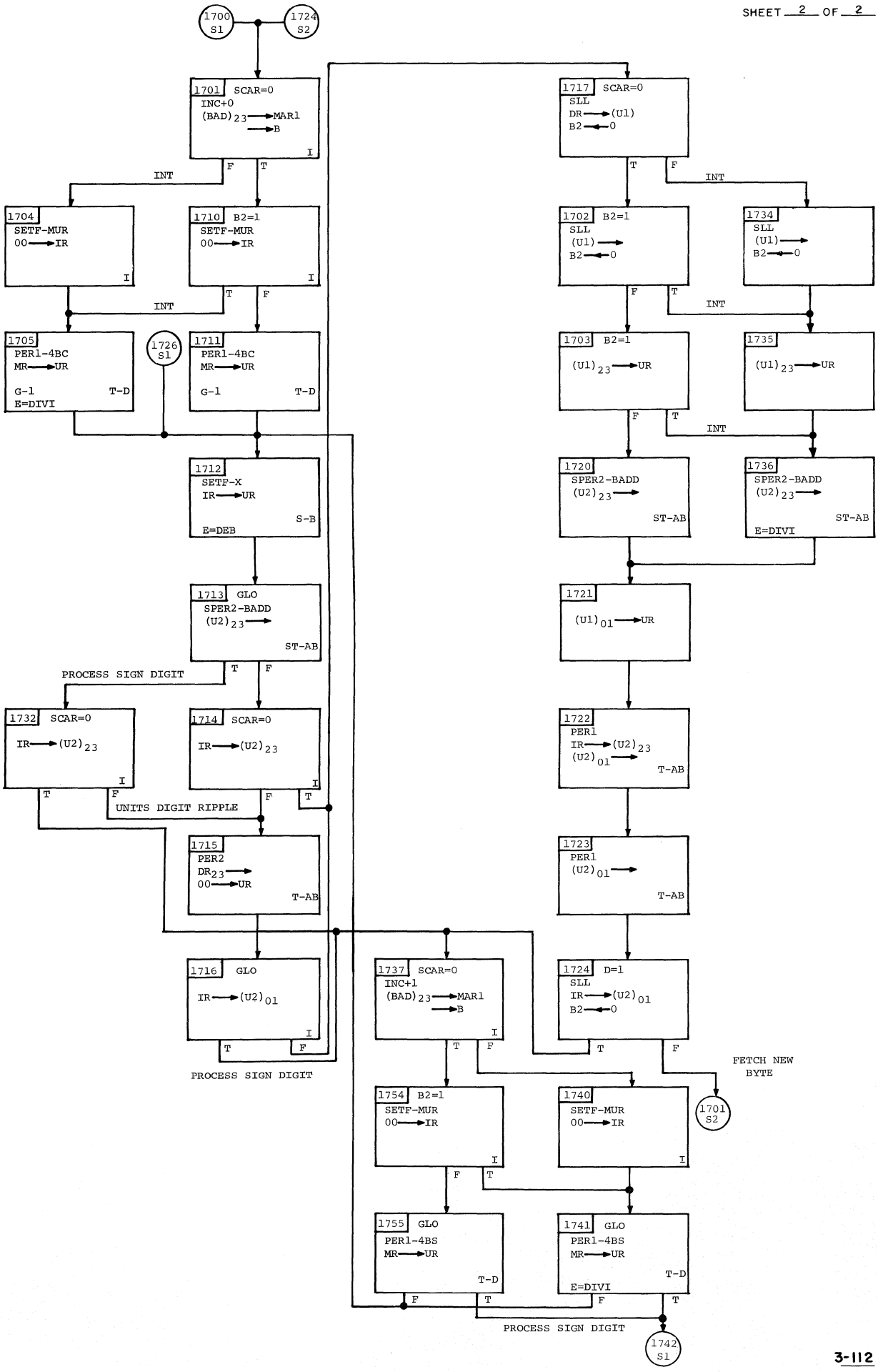
CONVERT TO DECIMAL CVD RX 4E



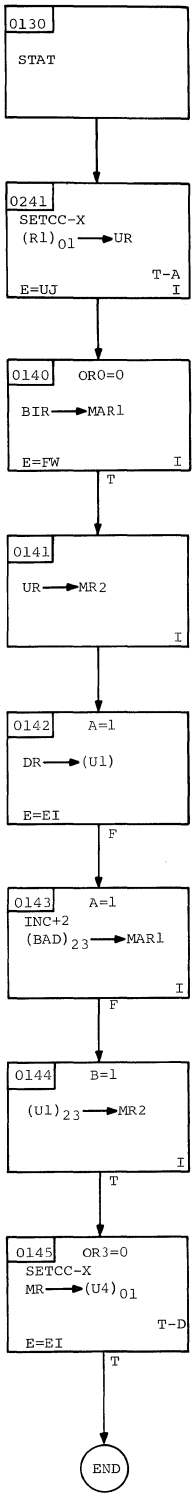
CONVERT TO BINARY CVB RX 4F



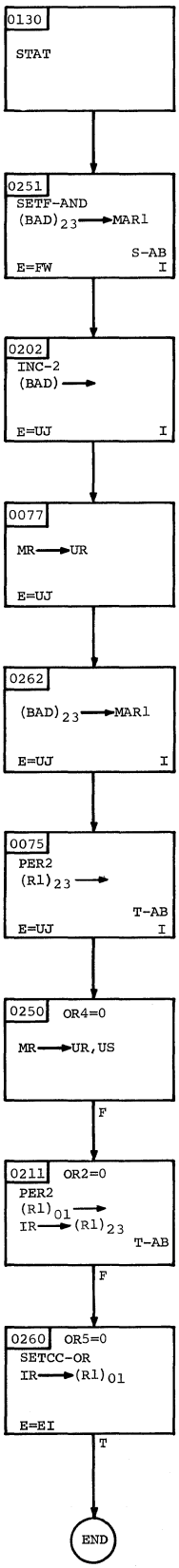
CONVERT TO BINARY CVB RX 4F



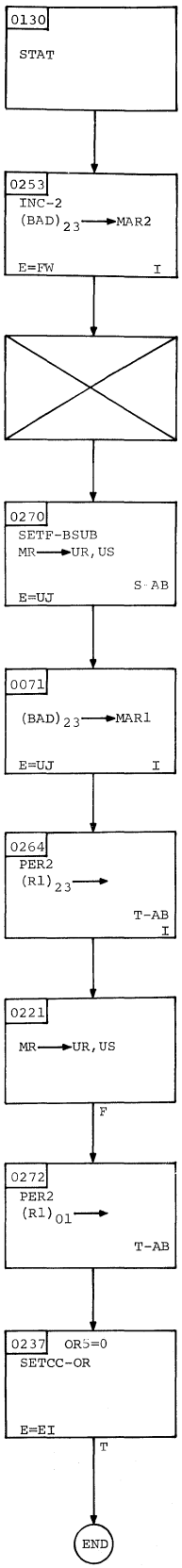
STORE ST RX 50



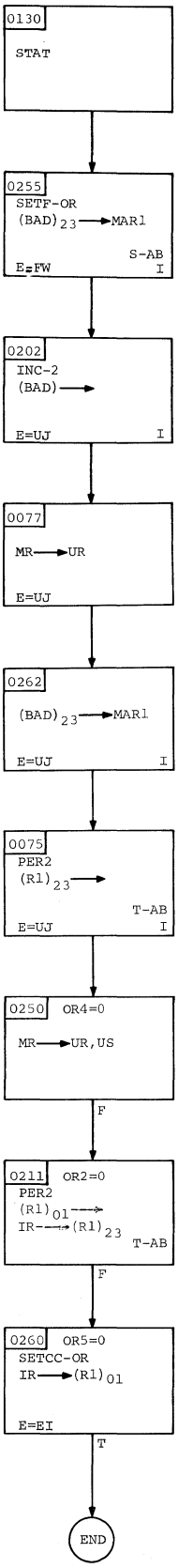
LOGICAL AND N RX 54



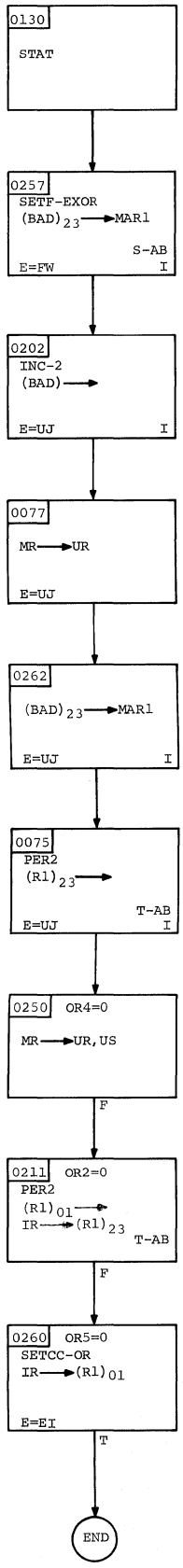
COMPARE LOGICAL CL RX 55



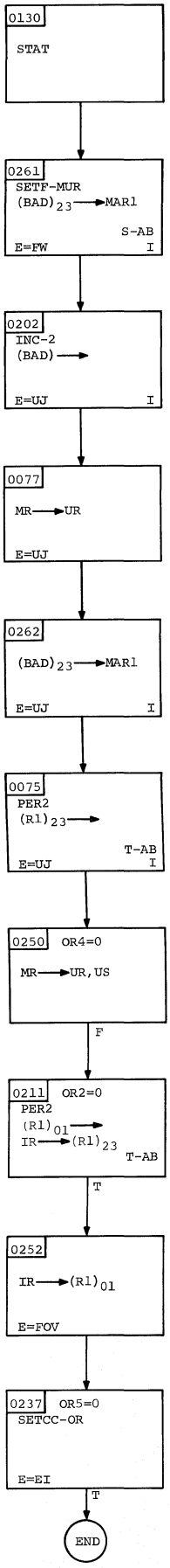
LOGICAL OR 0 RX 56

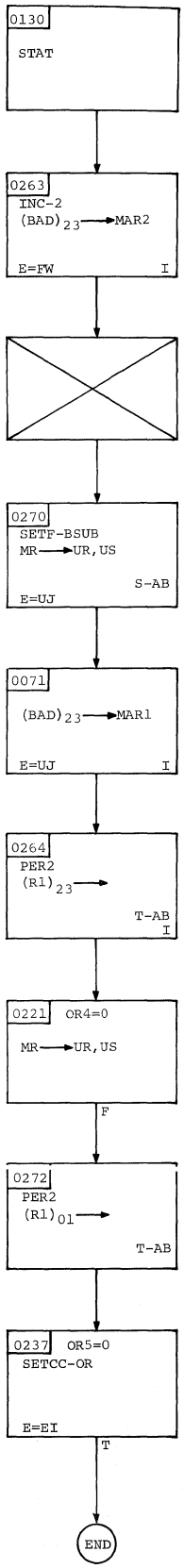


LOGICAL EXCLUSIVE OR X RX 57

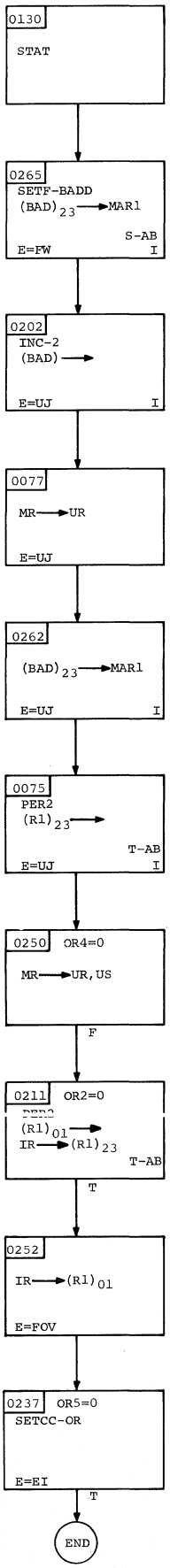


LOAD L RX 58

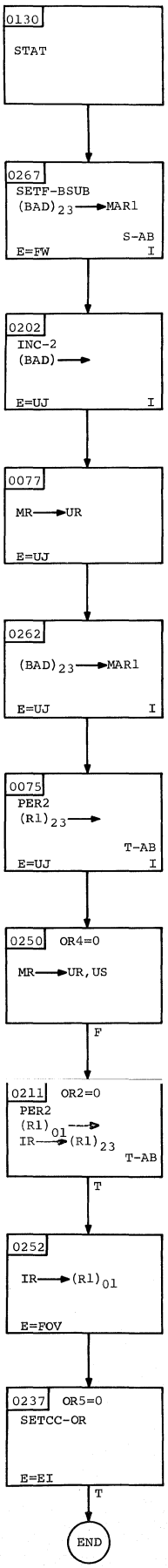

0130
STAT
0261
SETF-MUR
(BAD)23 → MAR1
S-AB
E=FW
I
0202
INC-2
(BAD) →
E=UJ
I
0077
MR → UR
E=UJ
0262
(BAD)23 → MAR1
E=UJ
I
0075
PER2
(R1)23 →
T-AB
E=UJ
I
0250
OR4=0
MR → UR,US
F
0211
OR2=0
PER2
(R1)01 →
IR → (R1)23
T-AB
T
0252
IR → (R1)01
E=FOV
0237
OR5=0
SETCC-OR
E=EI
T
END


0130
STAT
0263
INC-2
(BAD)23 → MAR2
E=FW
I
0270
SETF-BSUB
MR → UR, US
S-AB
E=UJ
0071
(BAD)23 → MAR1
E=UJ
I
0264
PER2
(R1)23 →
T-AB
I
0221
OR4=0
MR → UR, US
F
0272
PER2
(R1)01 →
T-AB
0237
OR5=0
SETCC-OR
E=EI
T
END

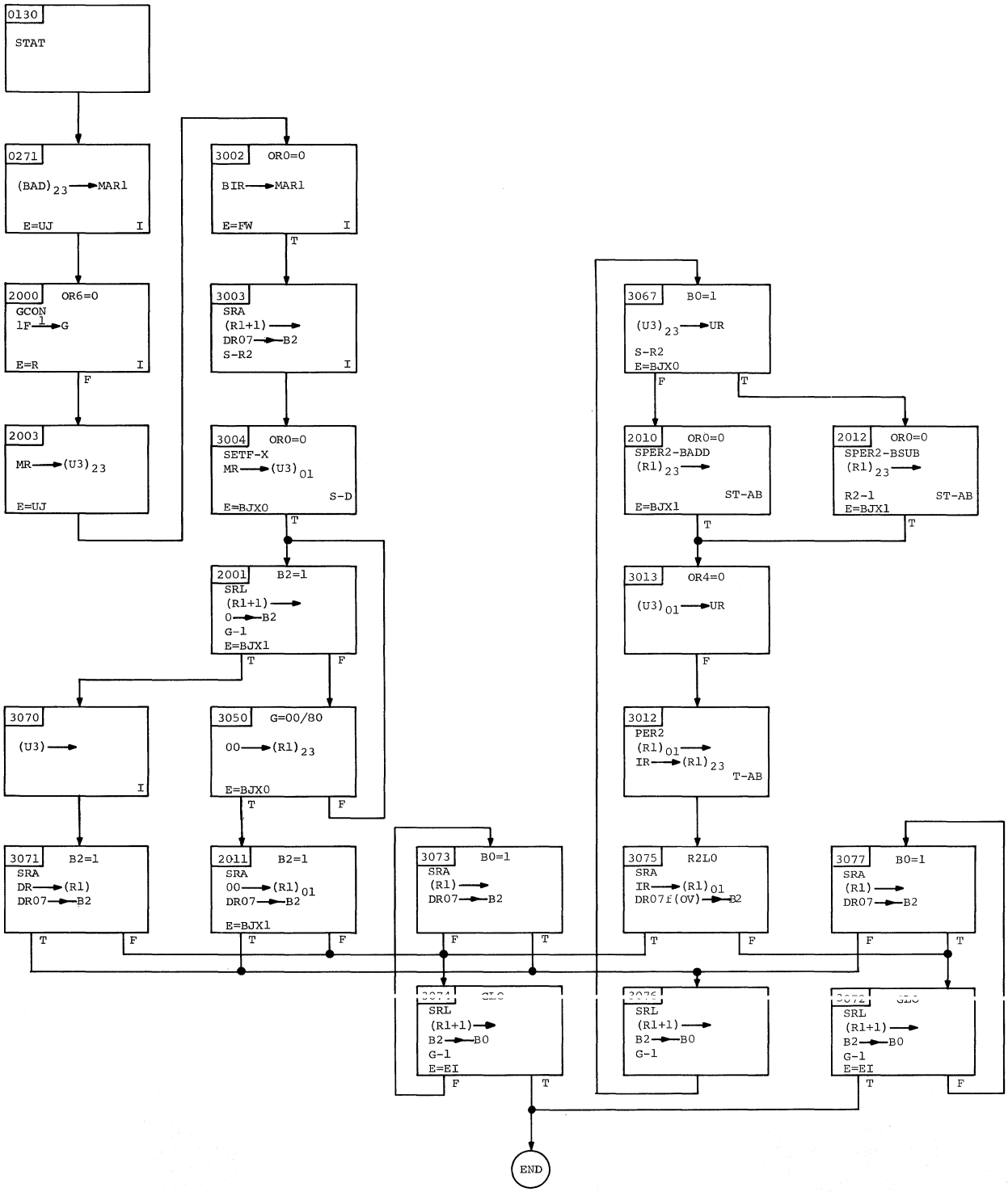
ADD A RX 5A


0130
STAT
0265
SETF-BADD
(BAD)23 → MAR1
S-AB
E=FW
I
0202
INC-2
(BAD) →
E=UJ
I
0077
MR → UR
E=UJ
0262
(BAD)23 → MAR1
E=UJ
I
0075
PER2
(R1)23 →
T-AB
E=UJ
I
0250
OR4=0
MR → UR, US
F
0211
OR2=0
(R1)01 →
IR → (R1)23
T-AB
T
0252
IR → (R1)01
E=FOV
0237
OR5=0
SETCC-OR
E=EI
T
END

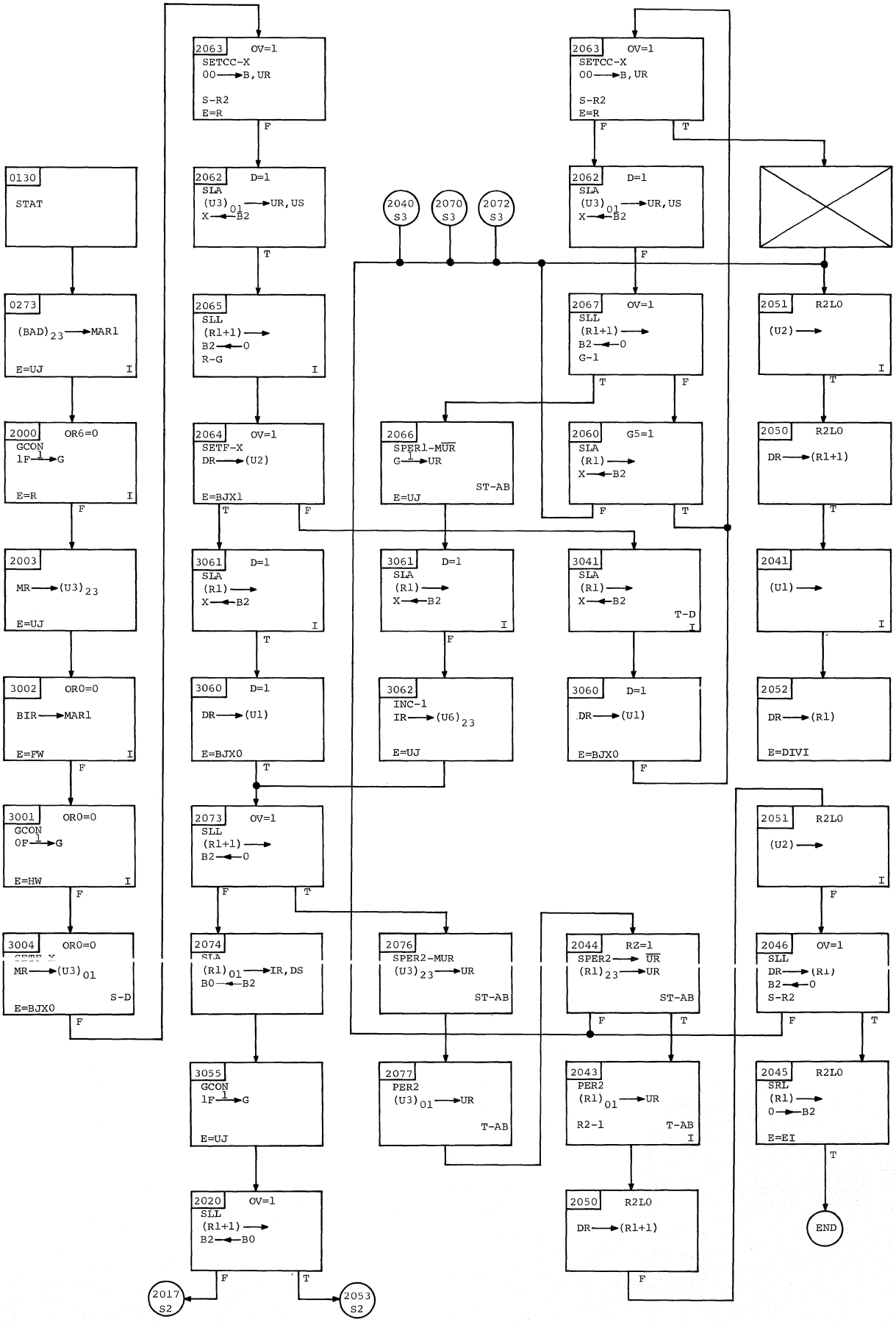
SUBTRACT S RX 5B



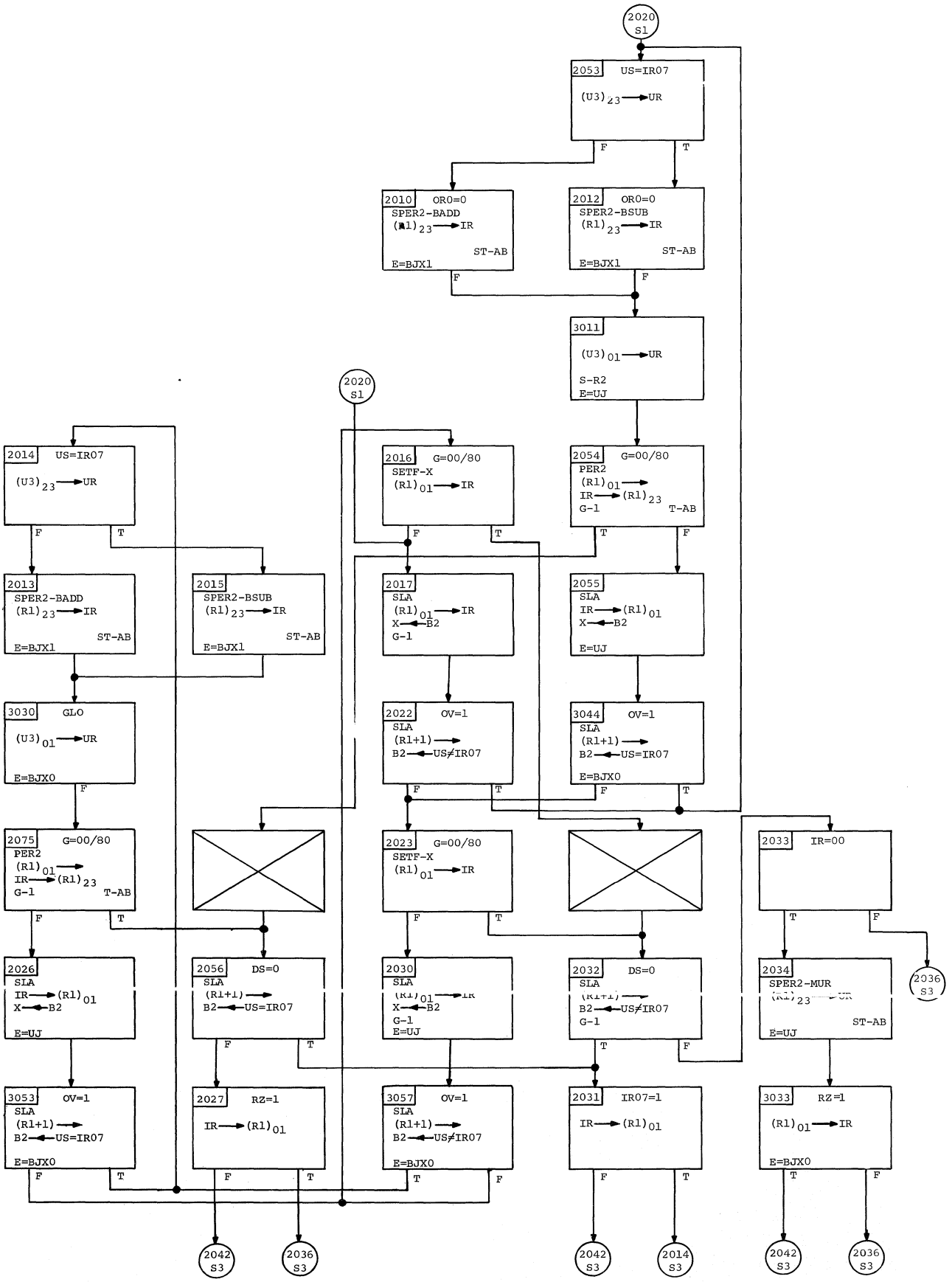
MULTIPLY M RX 5C

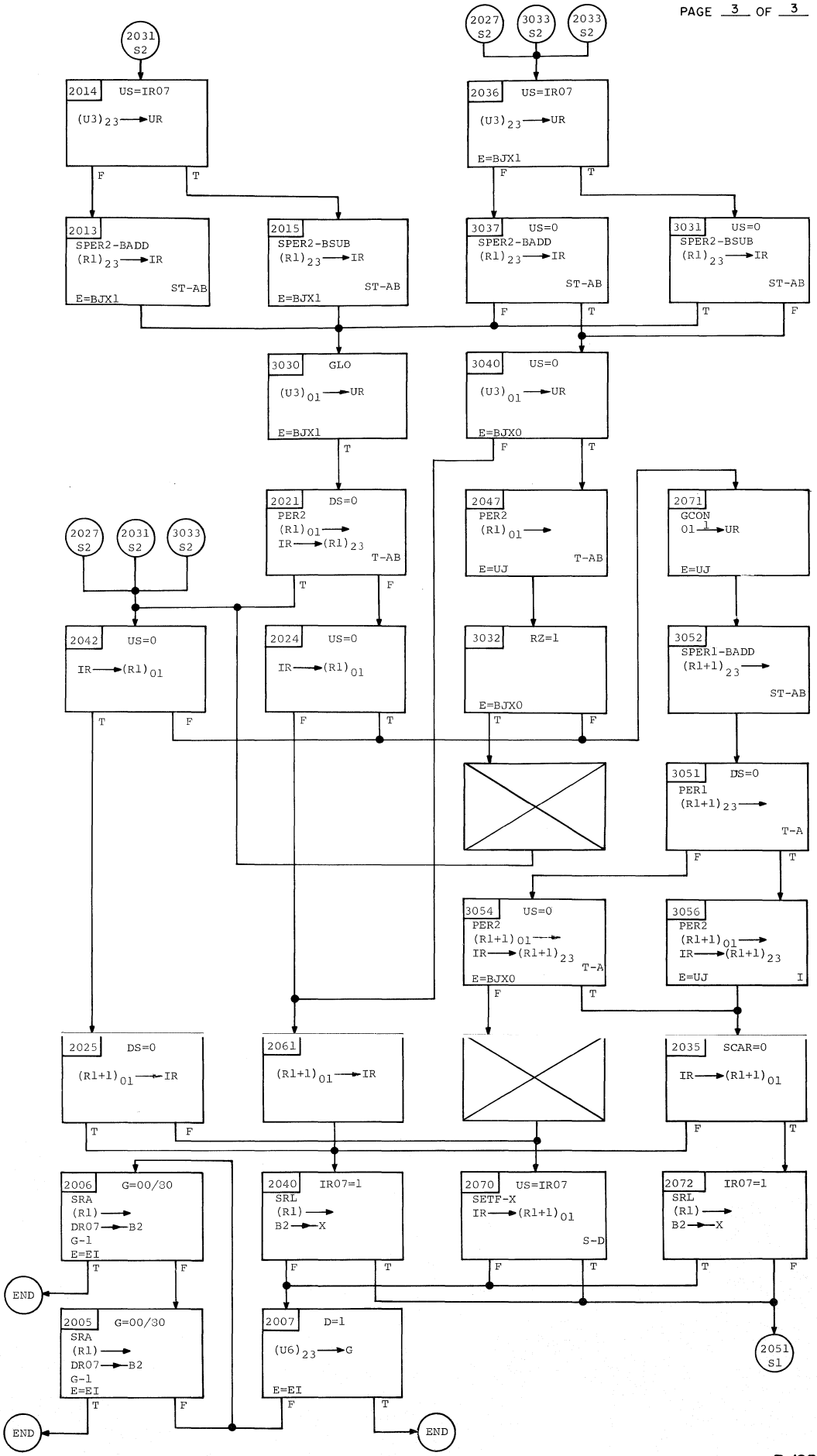


DIVIDE D RX 5D

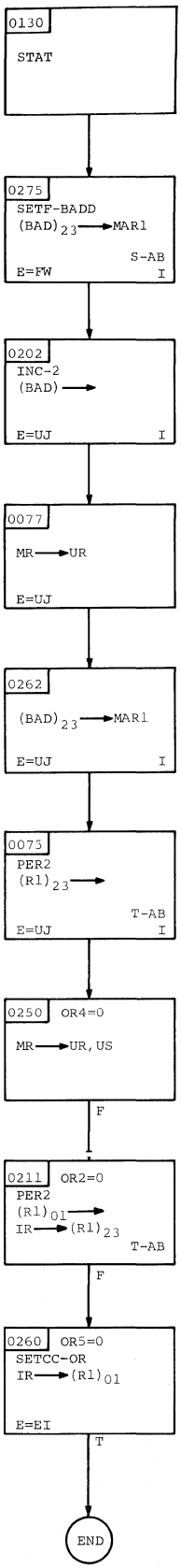


DIVIDE D RX 5D

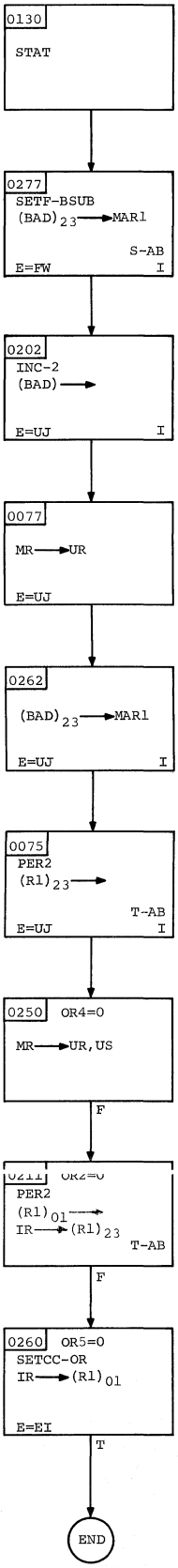



2031 S2
2027 S2
3033 S2
2033 S2
2014 US=IR07
(U3)23 → UR
2036 US=IR07
(U3)23 → UR
E=BJX1
2013 SPER2-BADD
(R1)23 → IR
ST-AB
E=BJX1
2015 SPER2-BSUB
(R1)23 → IR
ST-AB
E=BJX1
3037 US=0
SPER2-BADD
(R1)23 → IR
ST-AB
3031 US=0
SPER2-BSUB
(R1)23 → IR
ST-AB
3030 GLO
(U3)01 → UR
E=BJX1
3040 US=0
(U3)01 → UR
E=BJX0
2021 DS=0
PER2
(R1)01 →
IR → (R1)23
T-AB
2047
PER2
(R1)01 →
E=UJ
T-AB
2071
GCON
01 → UR
E=UJ
2042 US=0
IR → (R1)01
2024 US=0
IR → (R1)01
3032 RZ=1
E=BJX0
3052
SPER1-BADD
(R1+1)23 →
ST-AB
3051 DS=0
PER1
(R1+1)23 →
T-A
3054 US=0
PER2
(R1+1)01 →
IR → (R1+1)23
T-A
E=BJX0
3056
PER2
(R1+1)01 →
IR → (R1+1)23
E=UJ
I
2025 DS=0
(R1+1)01 → IR
2061
(R1+1)01 → IR
2035 SCAR=0
IR → (R1+1)01
2006 G=00/30
SRA
(R1) →
DR07 → B2
G-1
E=EI
2040 IR07=1
SRL
(R1) →
B2 → X
2070 US=IR07
SETF-X
IR → (R1+1)01
S-D
2072 IR07=1
SRL
(R1) →
B2 → X
END
2005 G=00/30
SRA
(R1) →
DR07 → B2
G-1
E=EI
2007 D=1
(U6)23 → G
E=EI
END
END
2051 S1
T
F

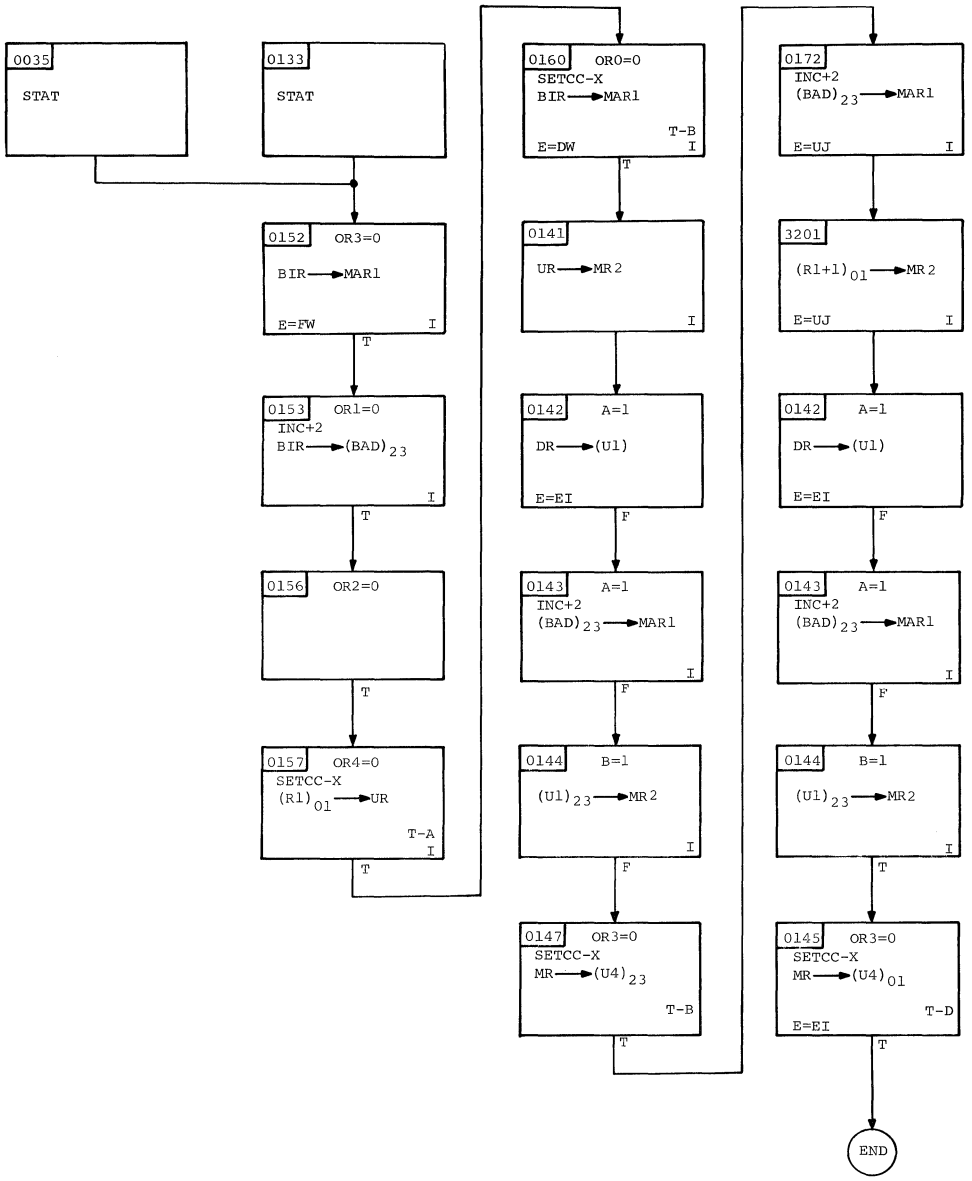
ADD LOGICAL AL RX 5E



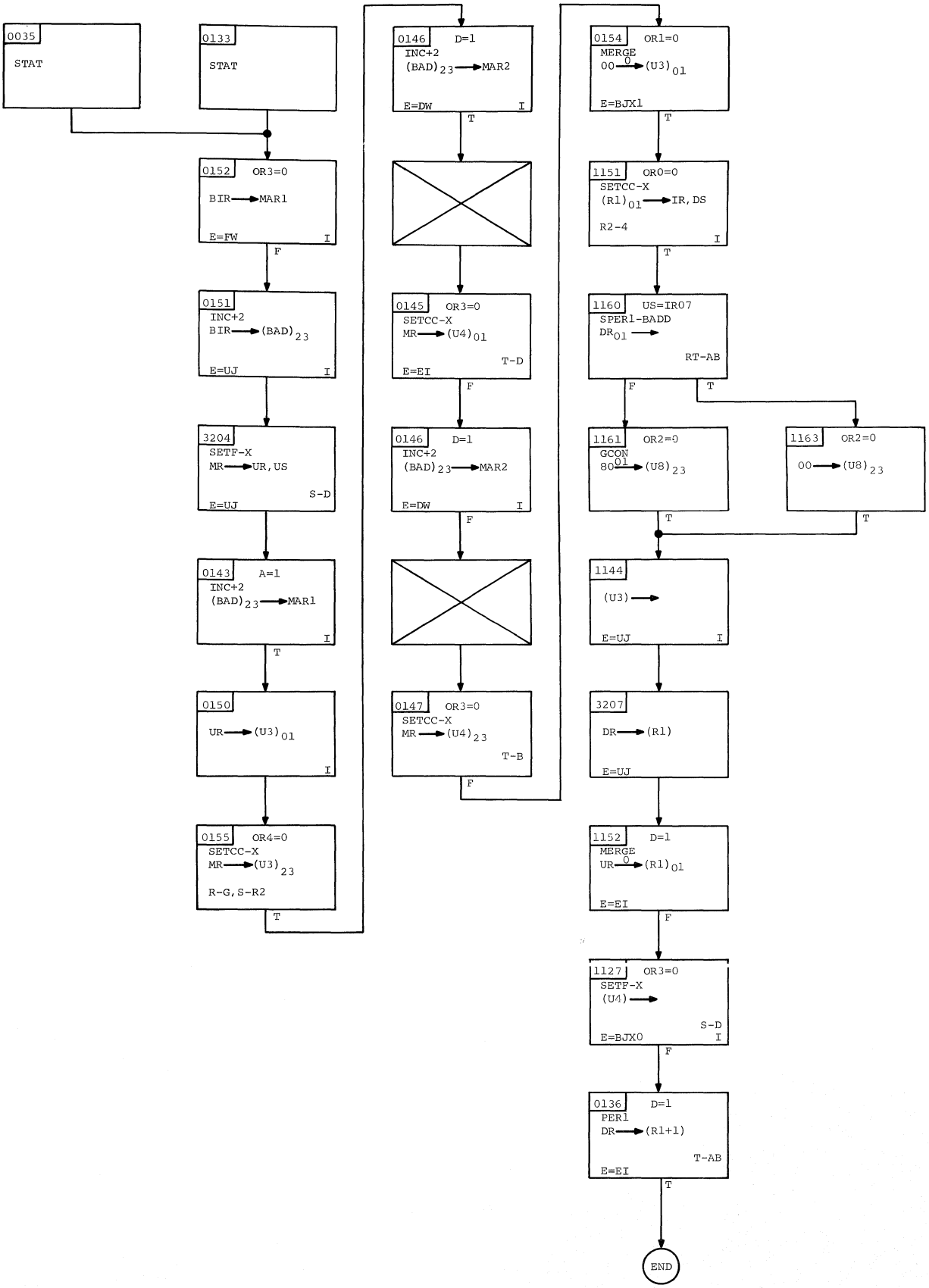
SUBTRACT LOGICAL SL RX 5F



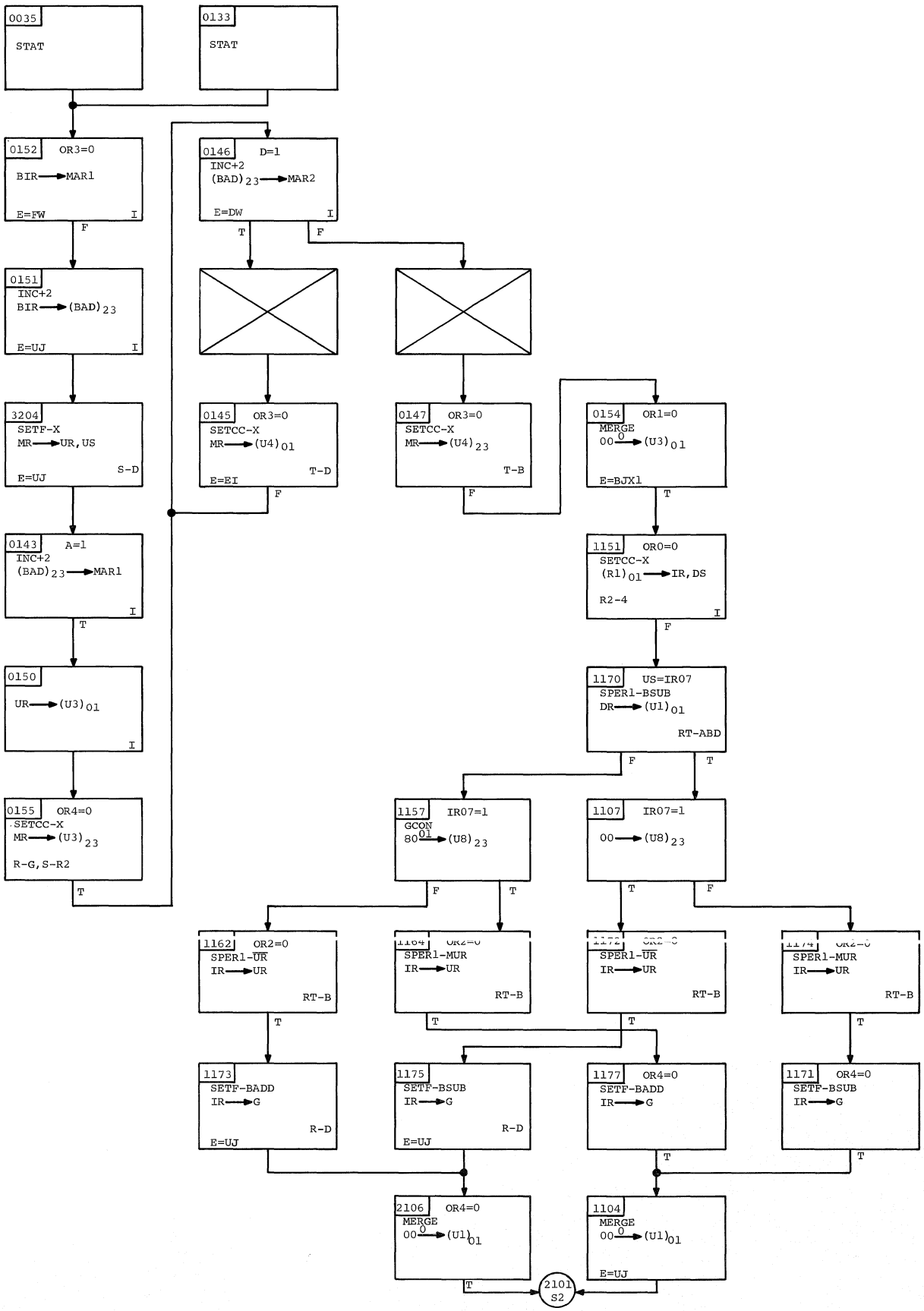
STORE LONG STD RX 60



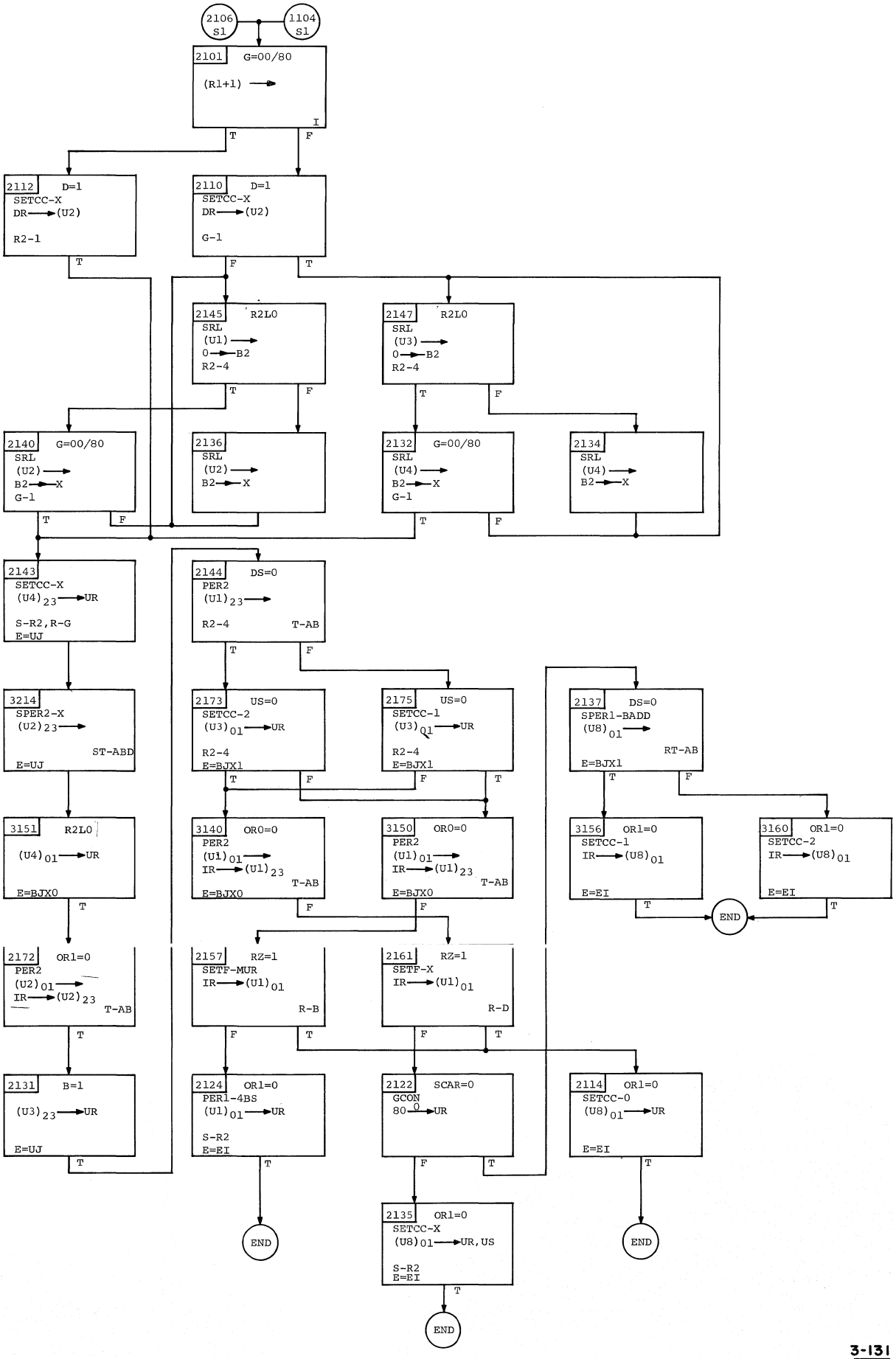
LOAD LONG LD RX 68



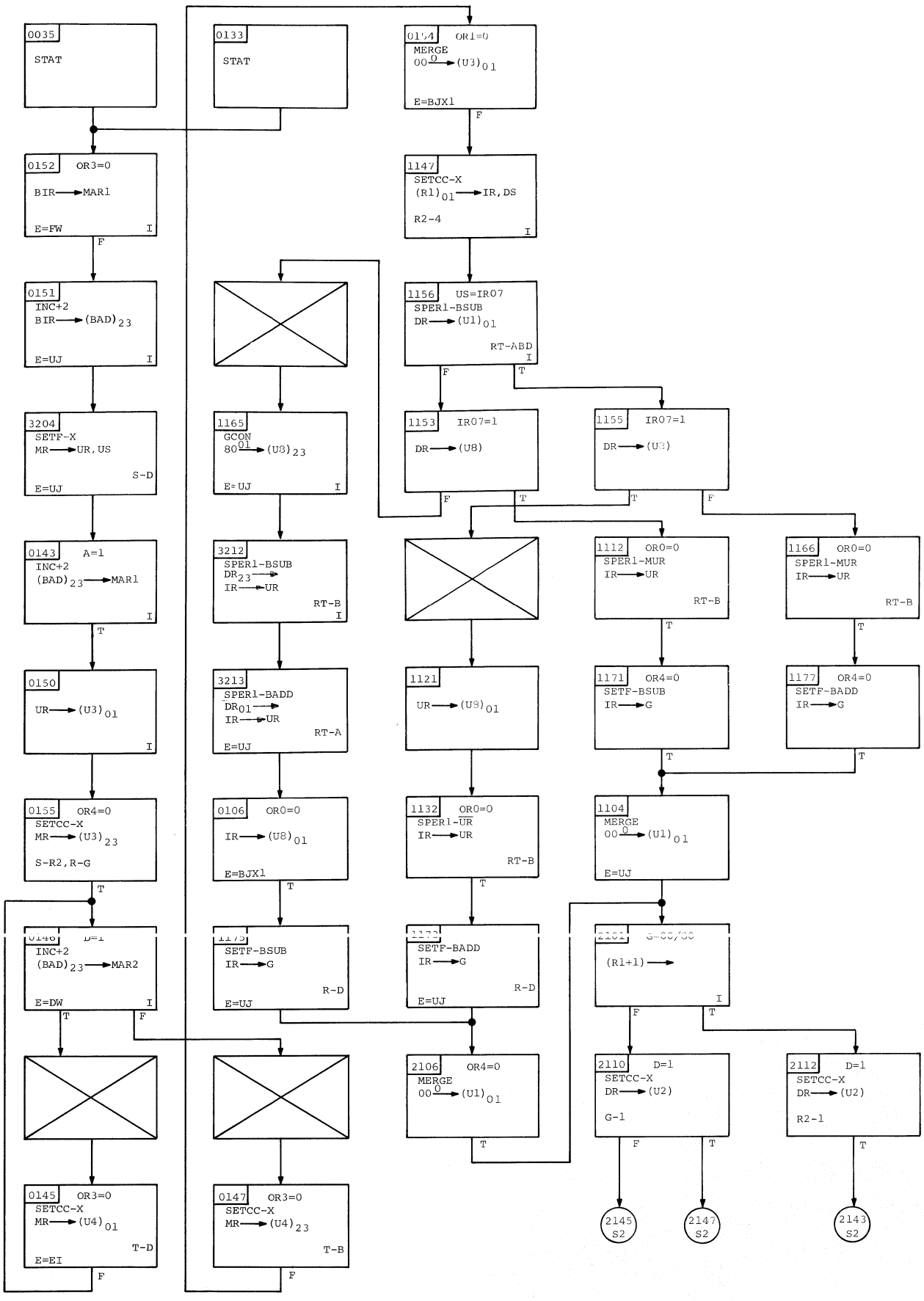
COMPARE LONG CD RX 69



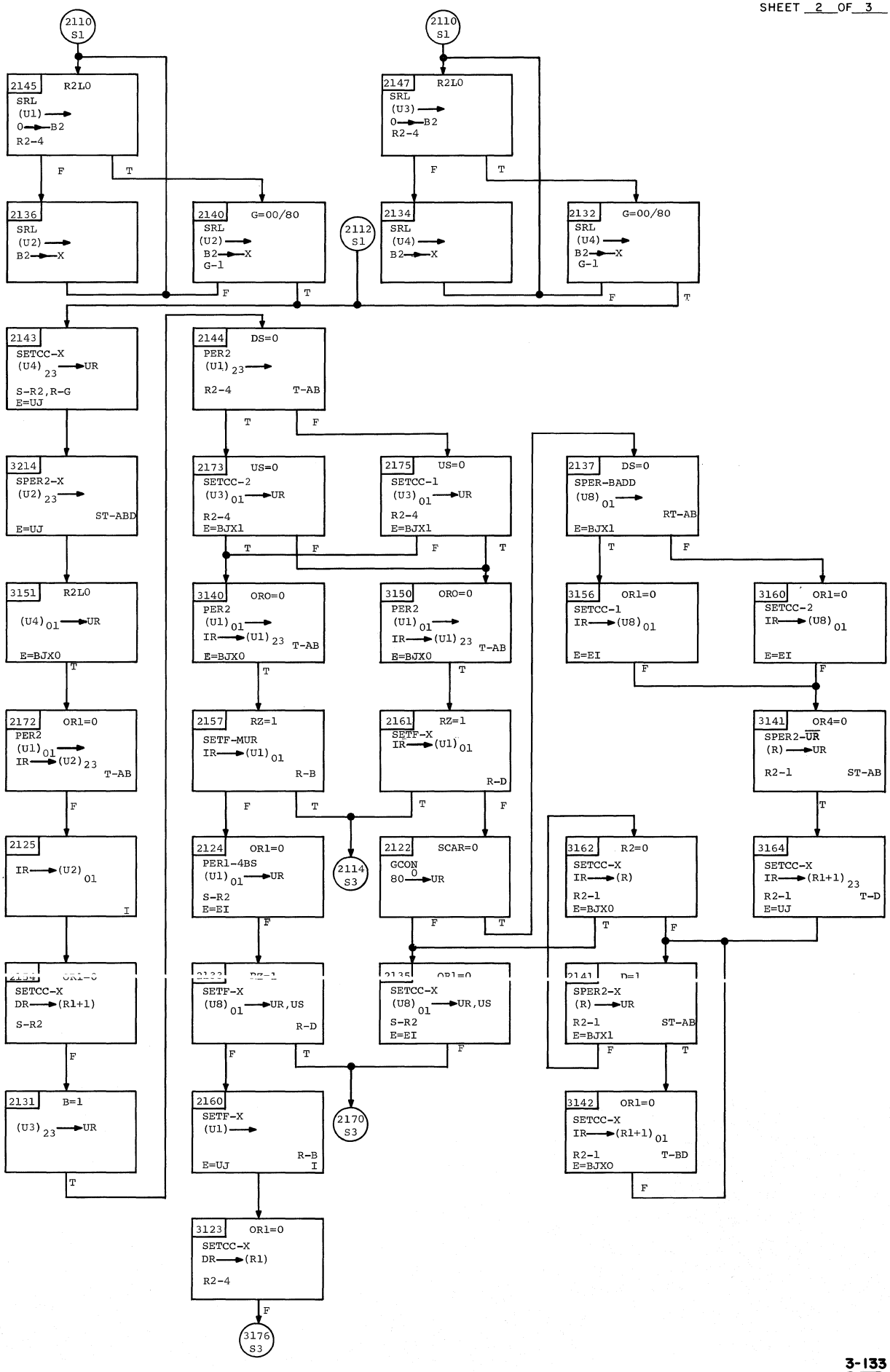
COMPARE LONG CD RX 69



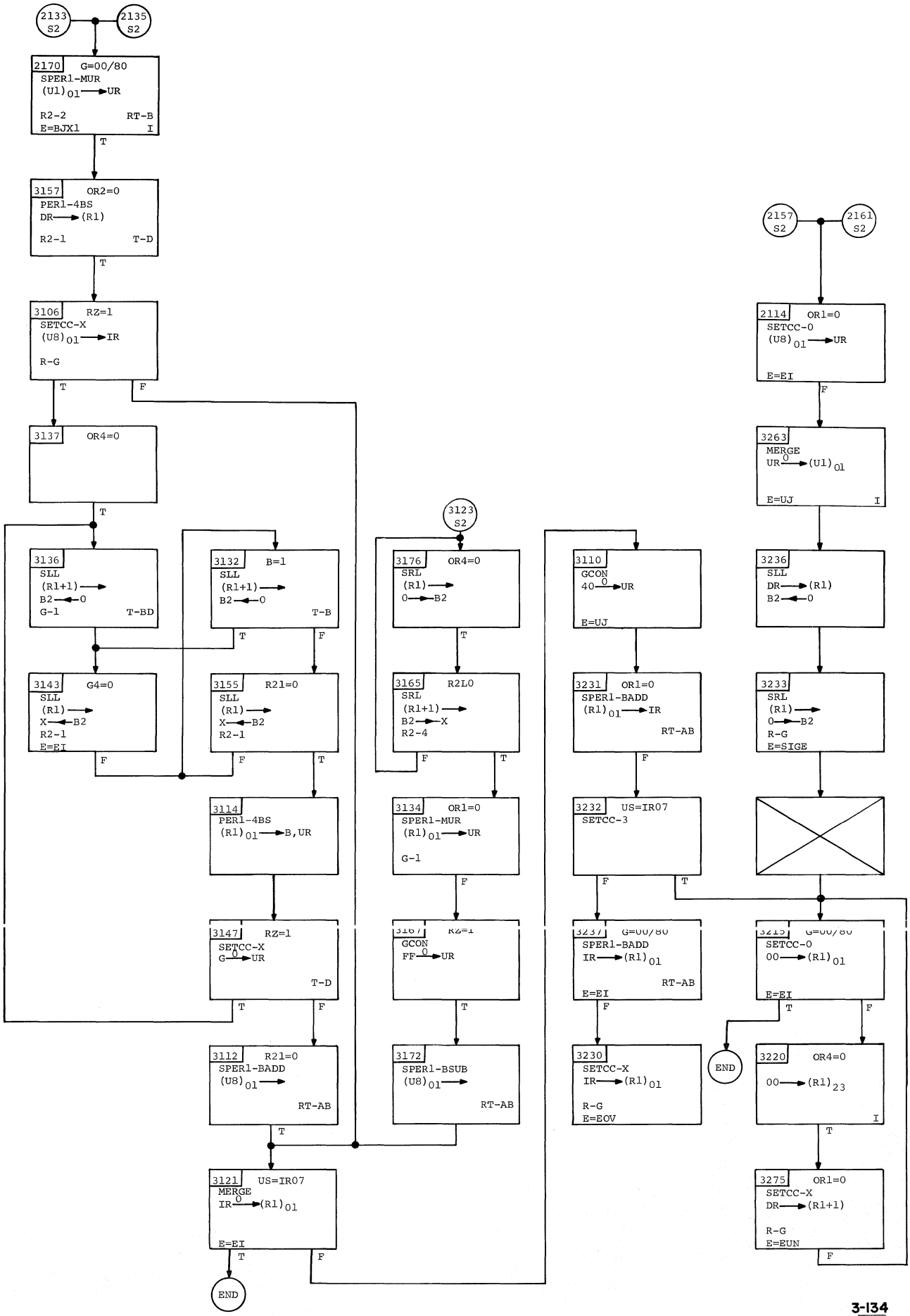
ADD NORMALIZED LONG AND RX 6A



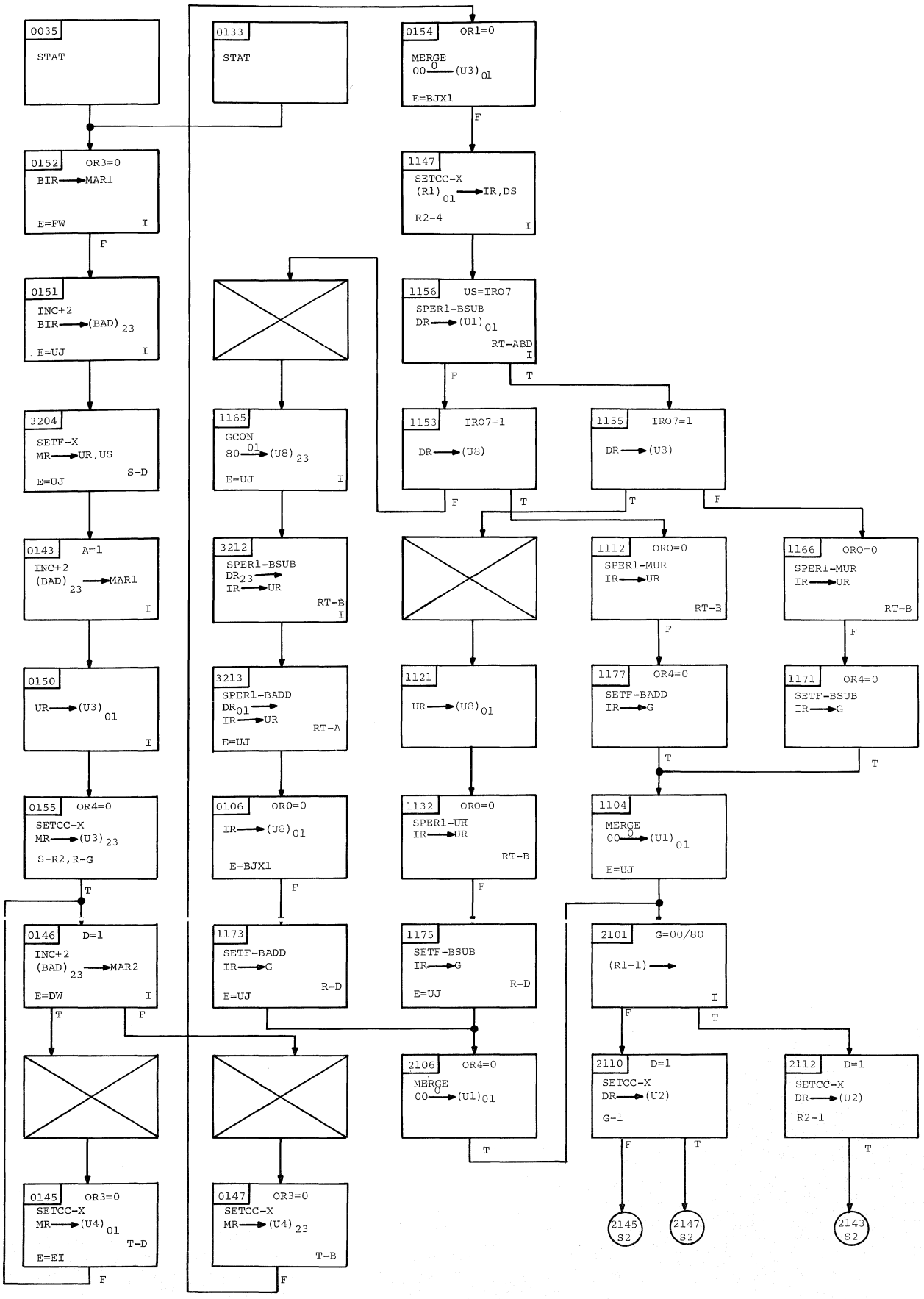
ADD NORMALIZED LONG AND RX 6A



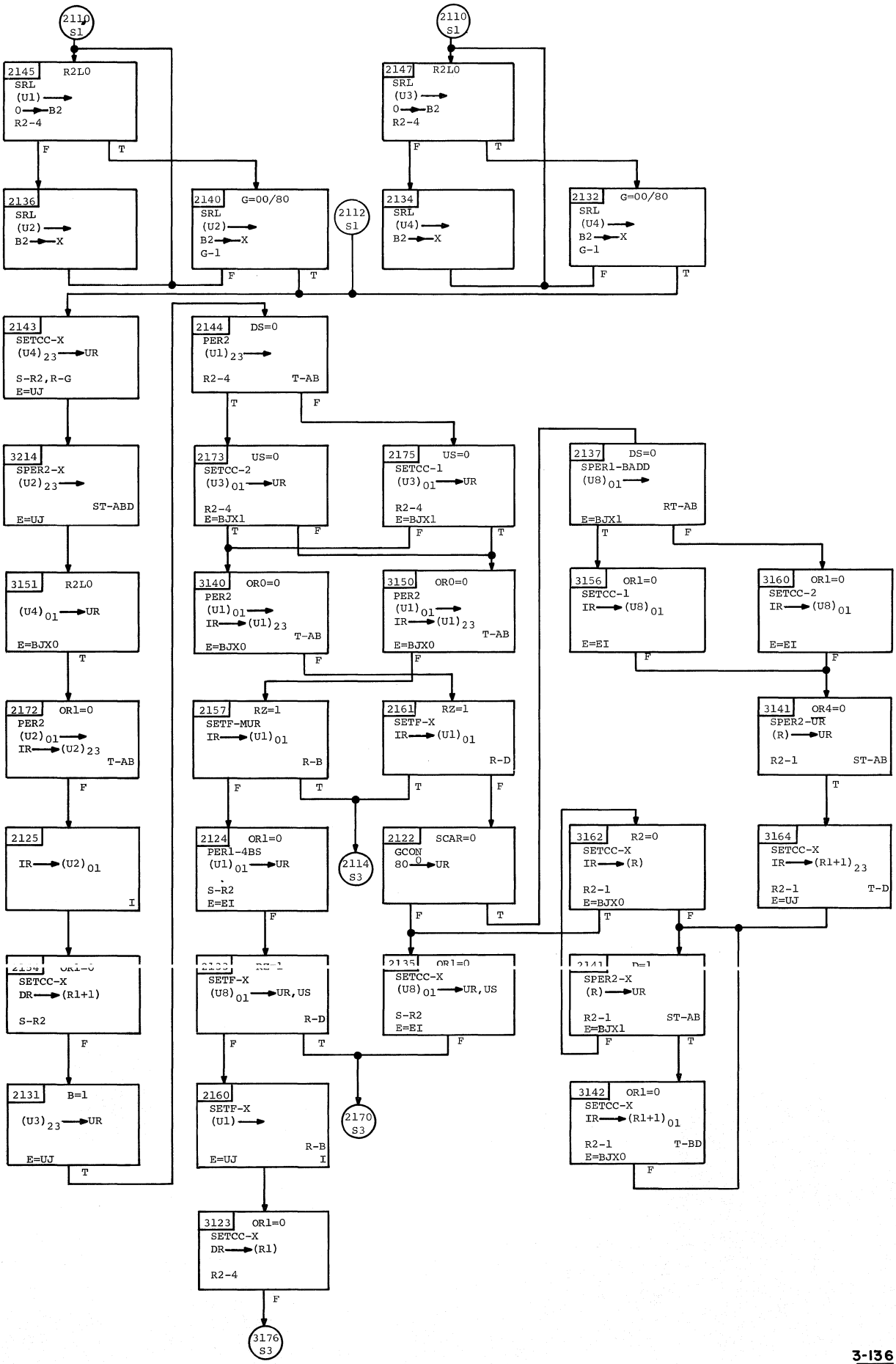
ADD NORMALIZED LONG AND RX 6A



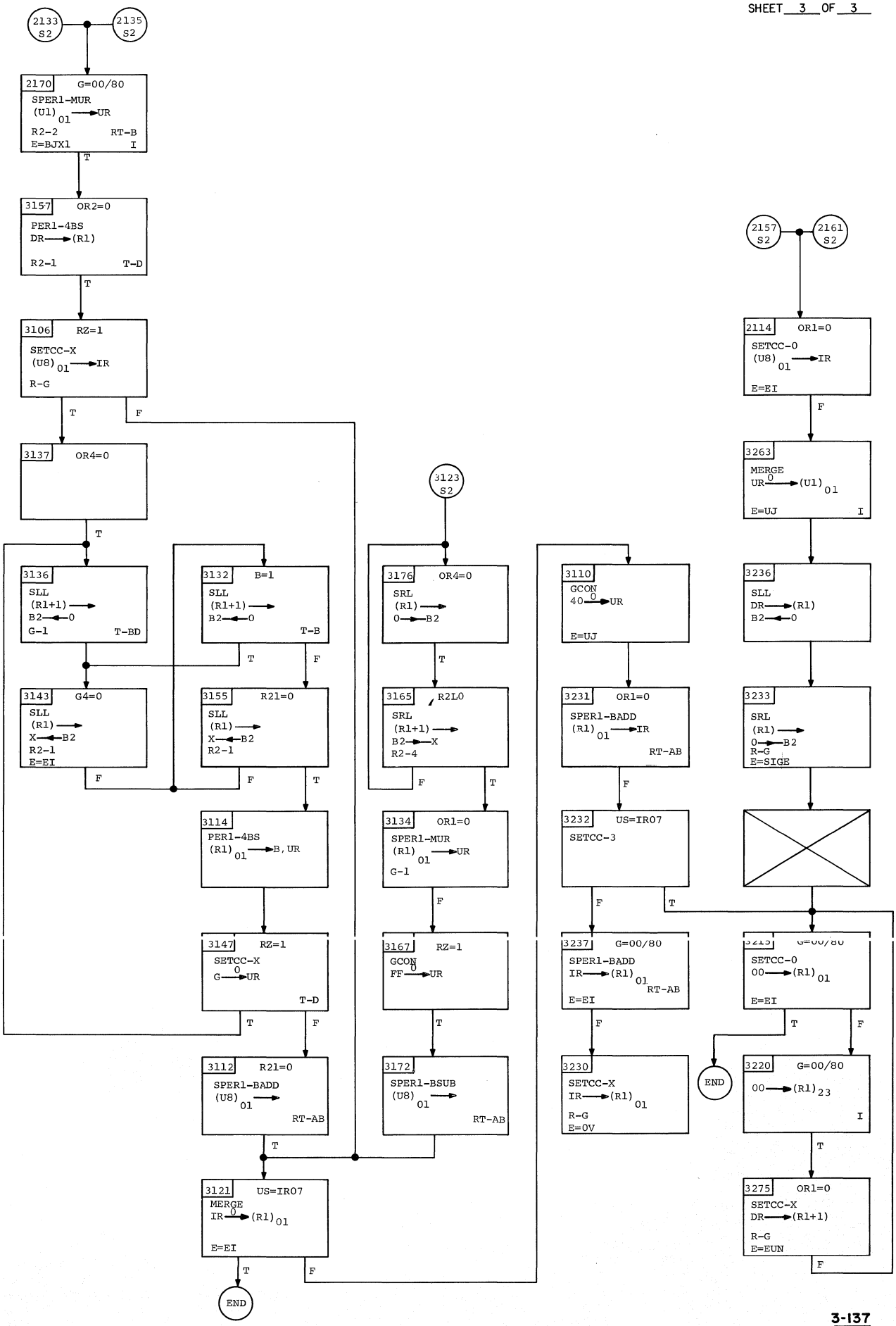
SUBTRACT NORMALIZED LONG SND RX 6B



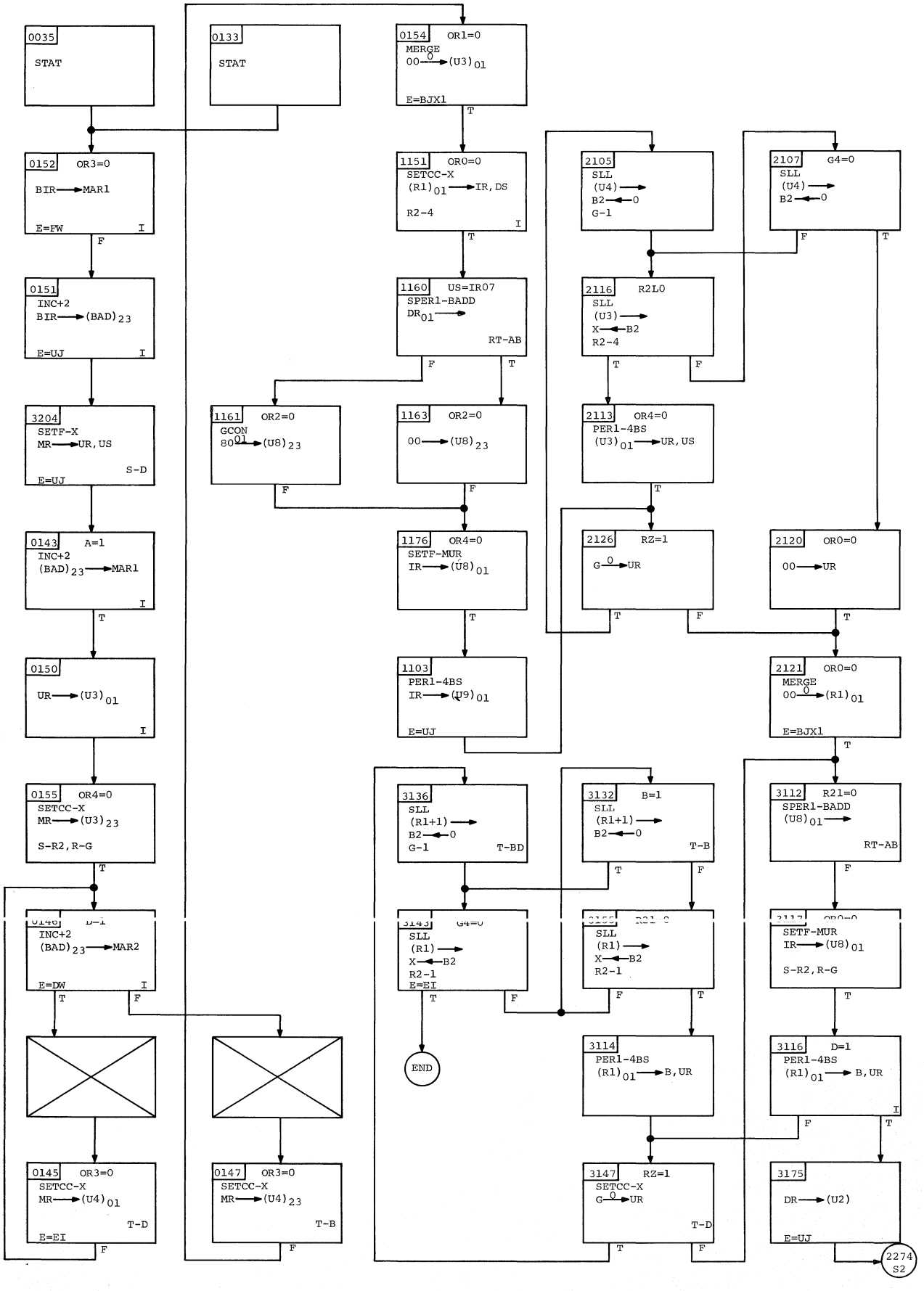
SUBTRACT NORMALIZED LONG SND RX 6B



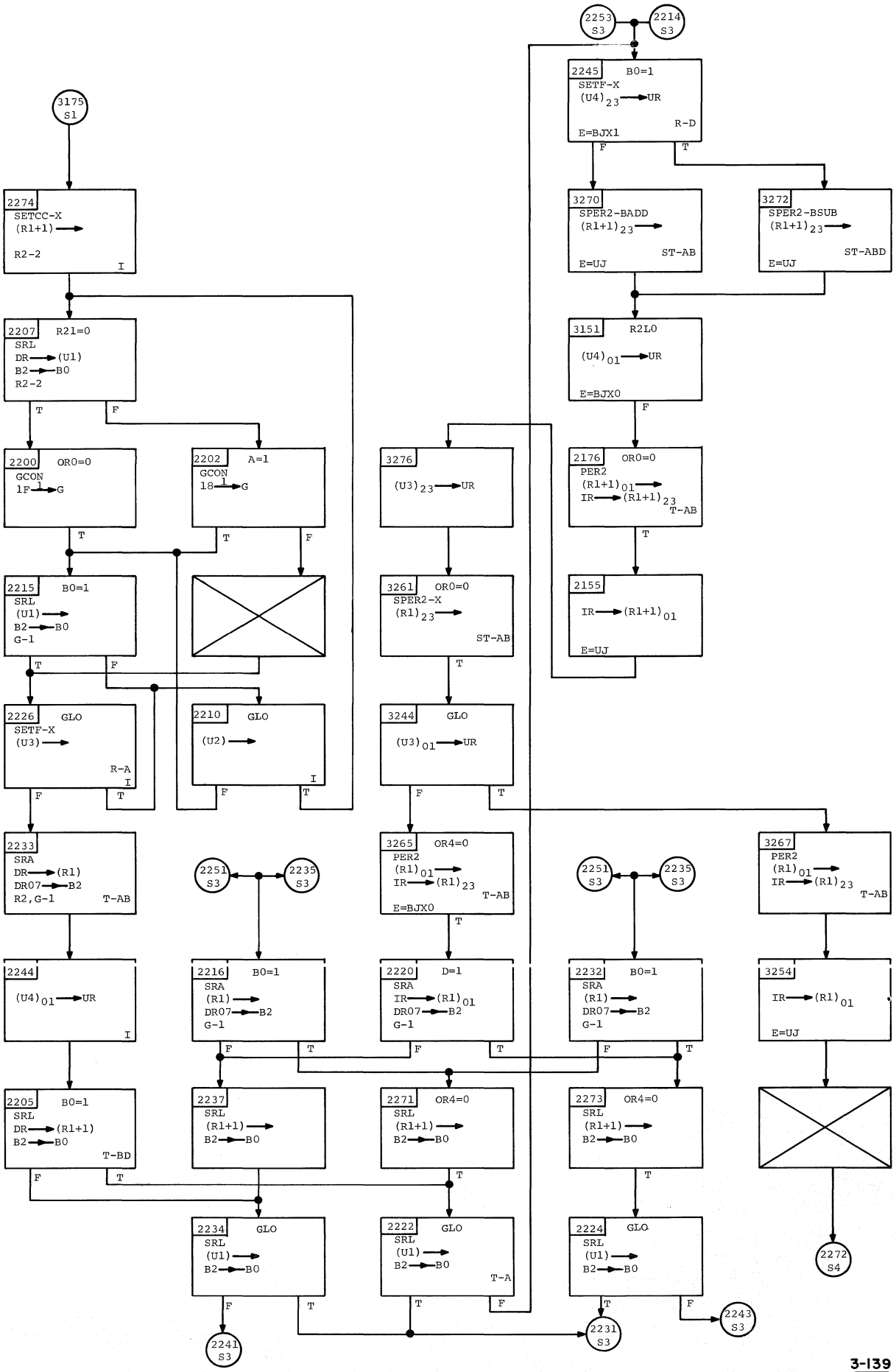
SUBTRACT NORMALIZED LONG SND RX 6B



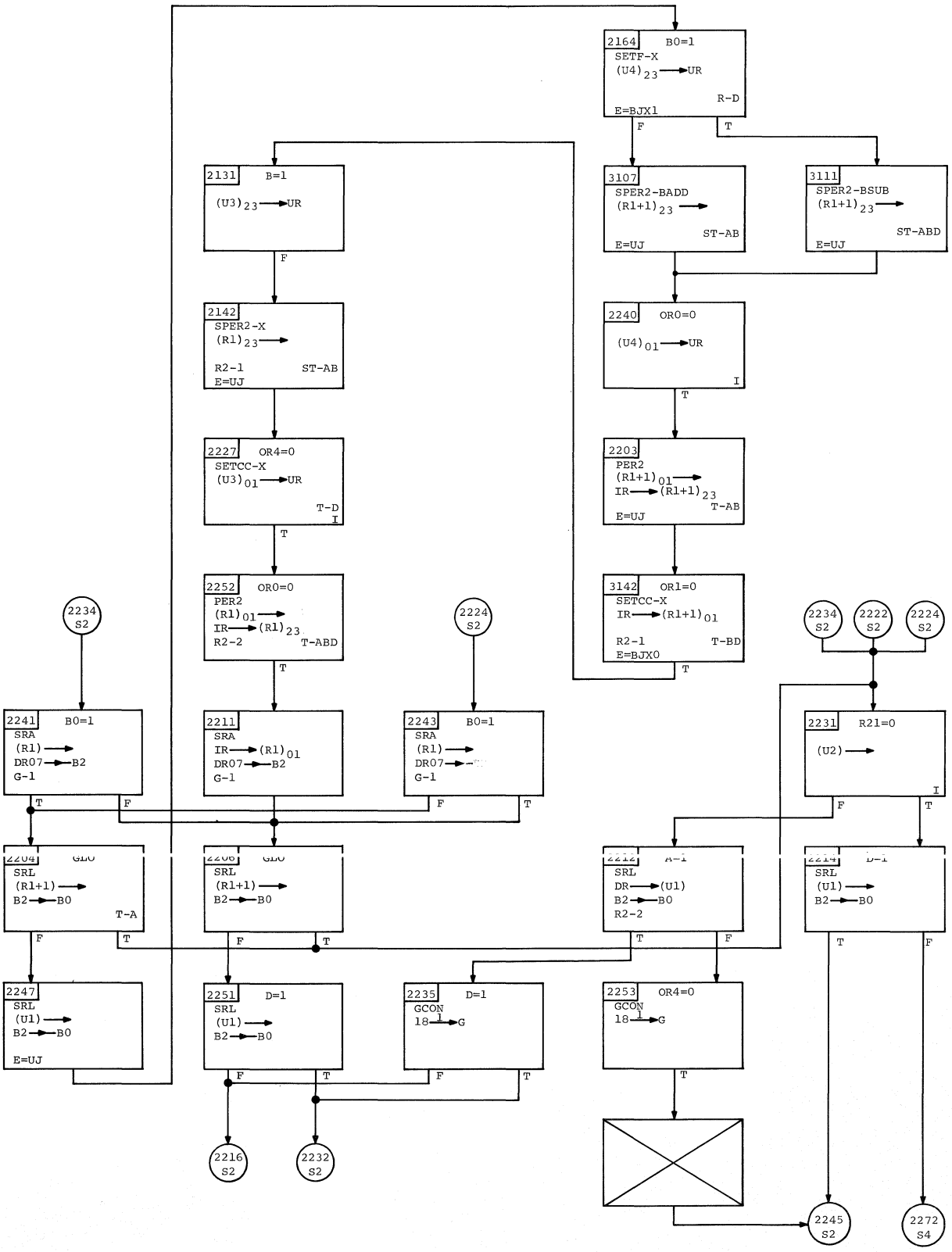
MULTIPLY LONG MD RX 6C



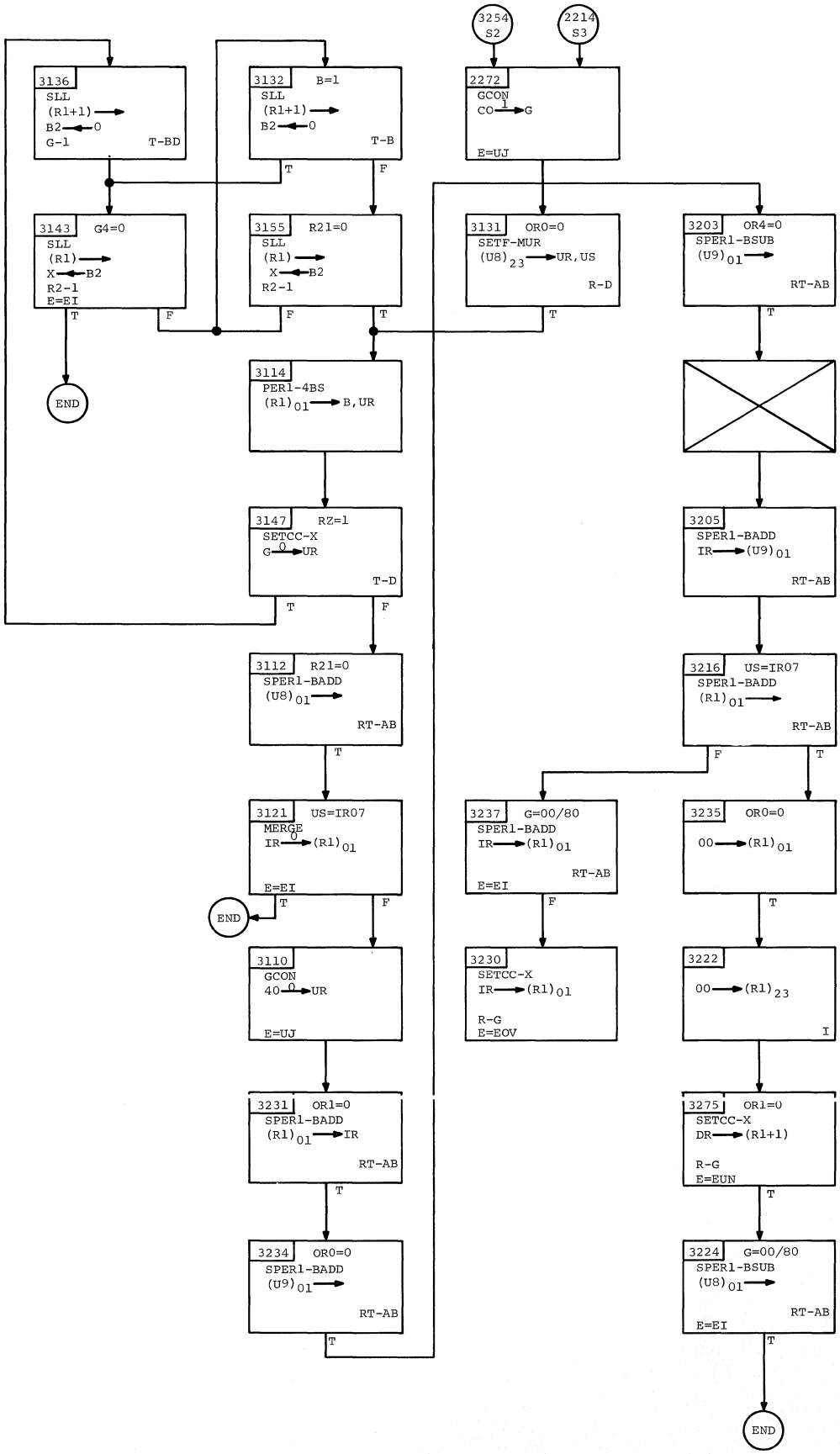
MULTIPLY LONG MD RX 6C



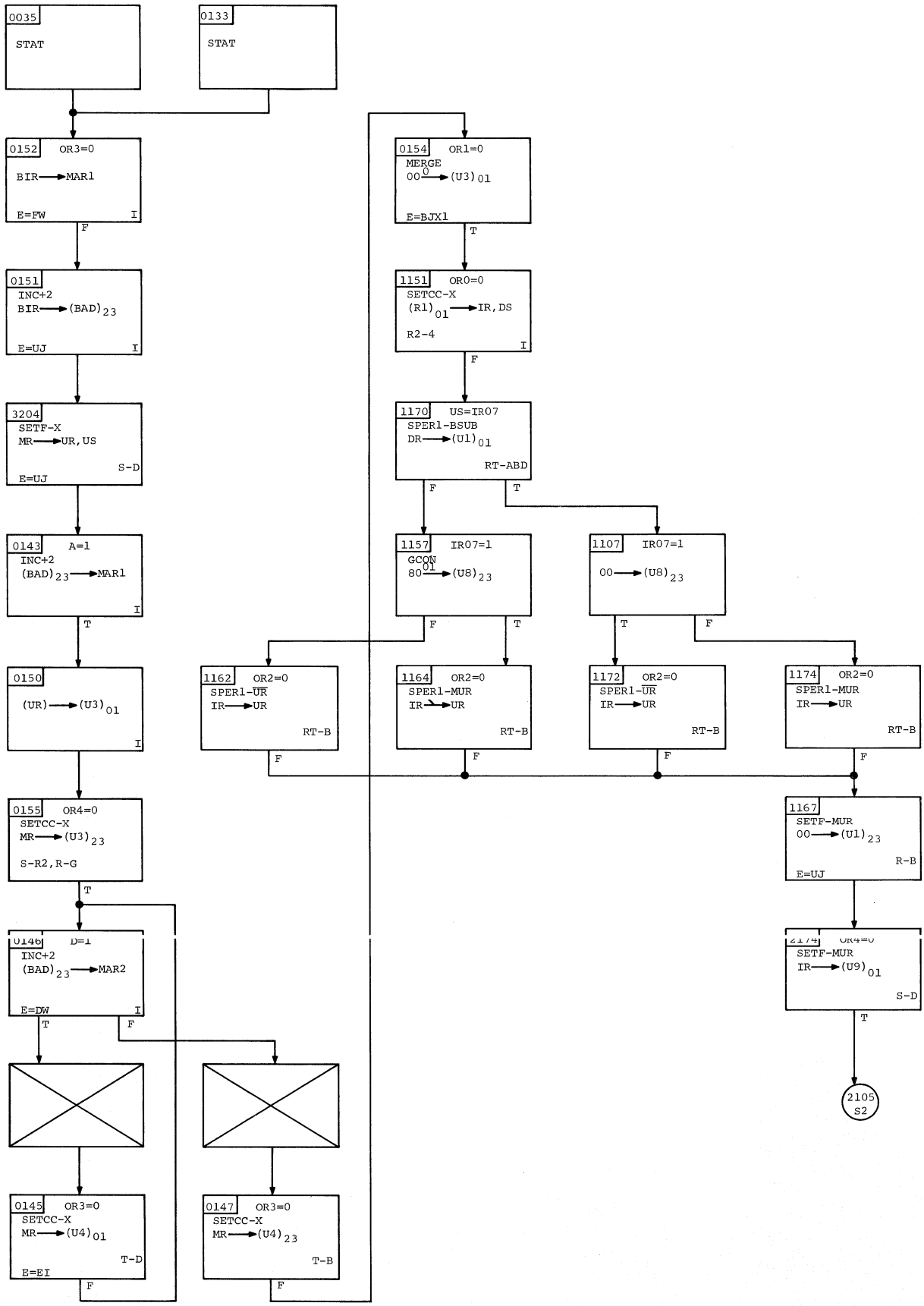
MULTIPLY LONG MD RX 6C



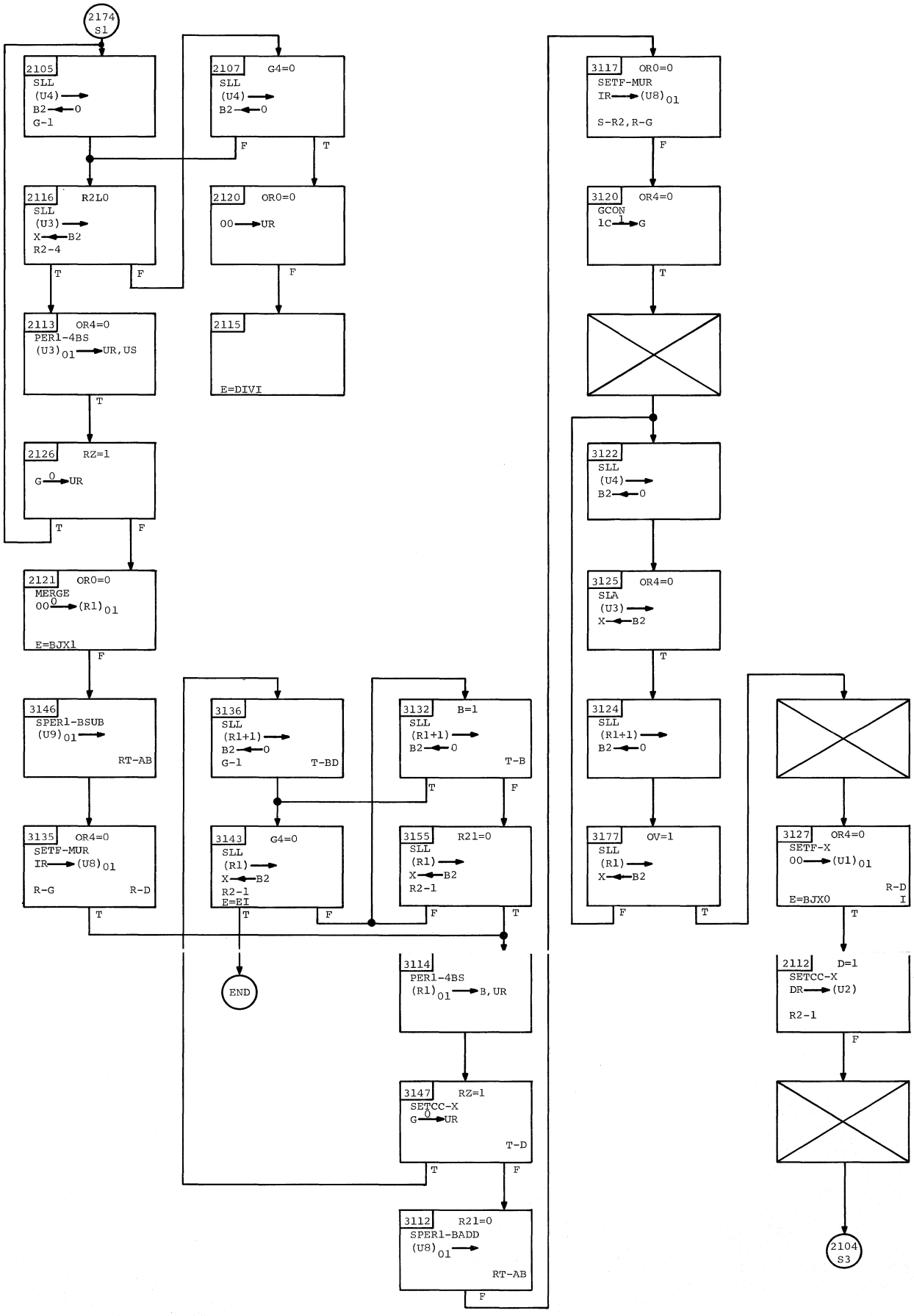
MULTIPLY LONG MD RX 6C



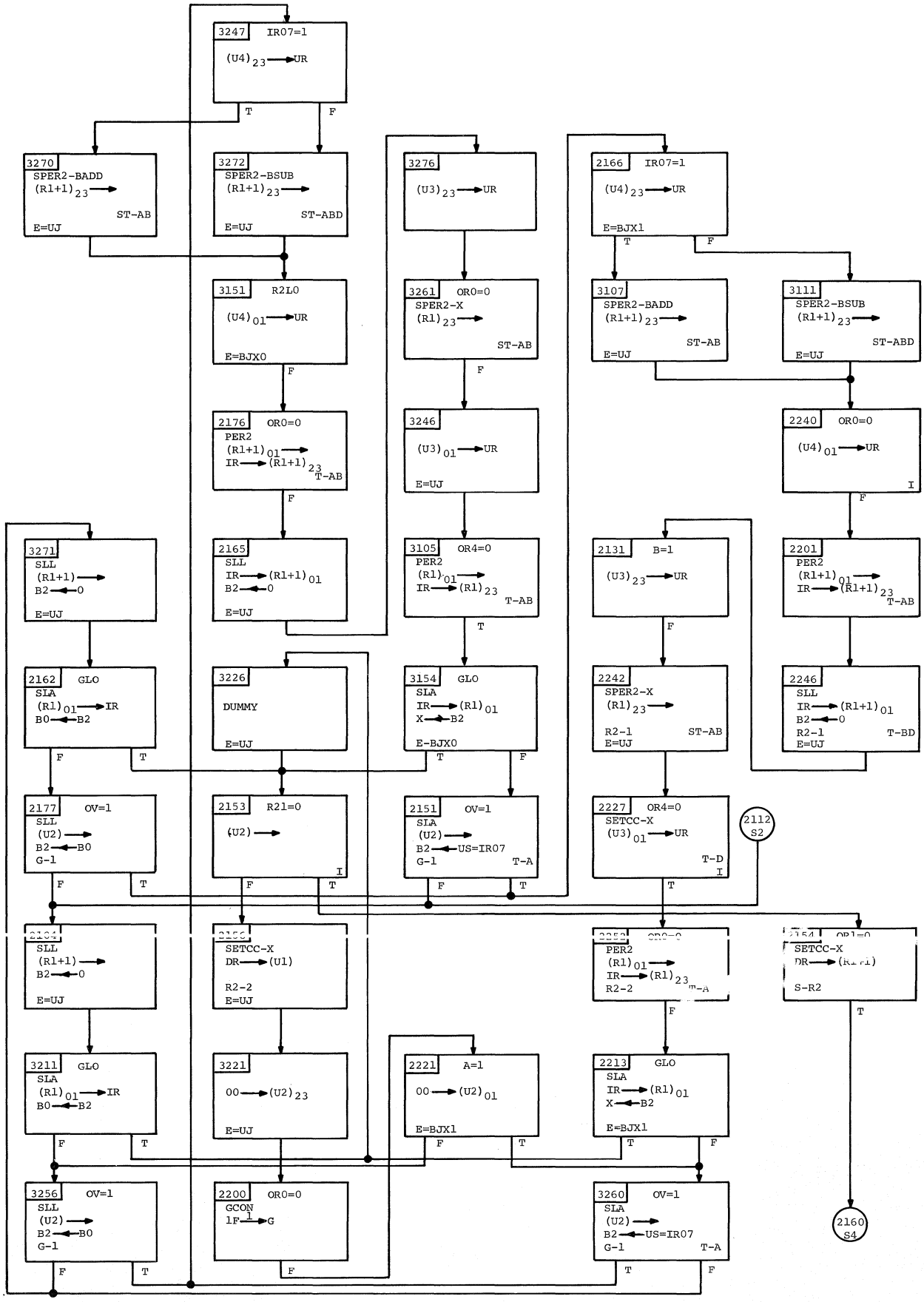
DIVIDE LONG DD RX 6D



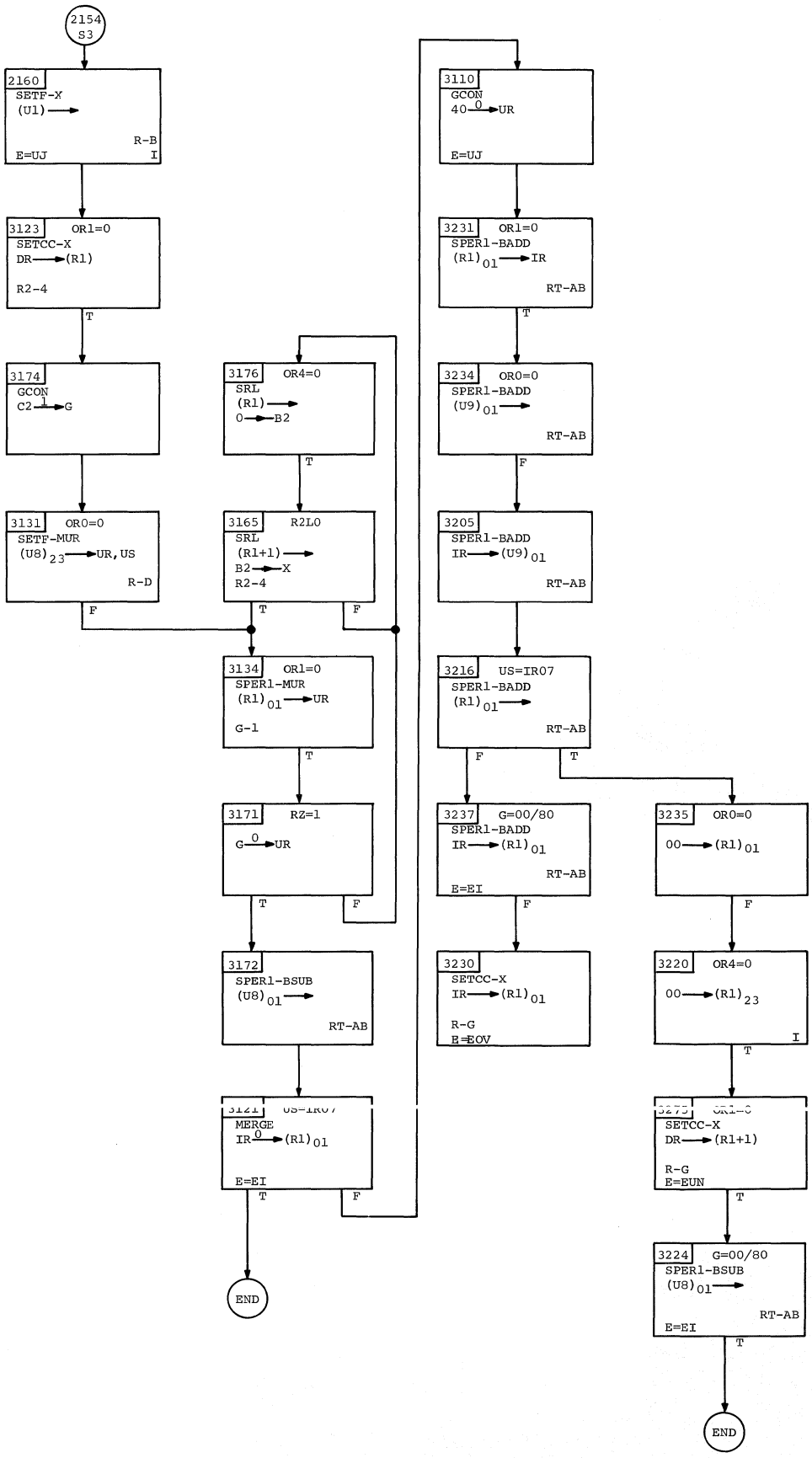
DIVIDE LONG DD RX 6D



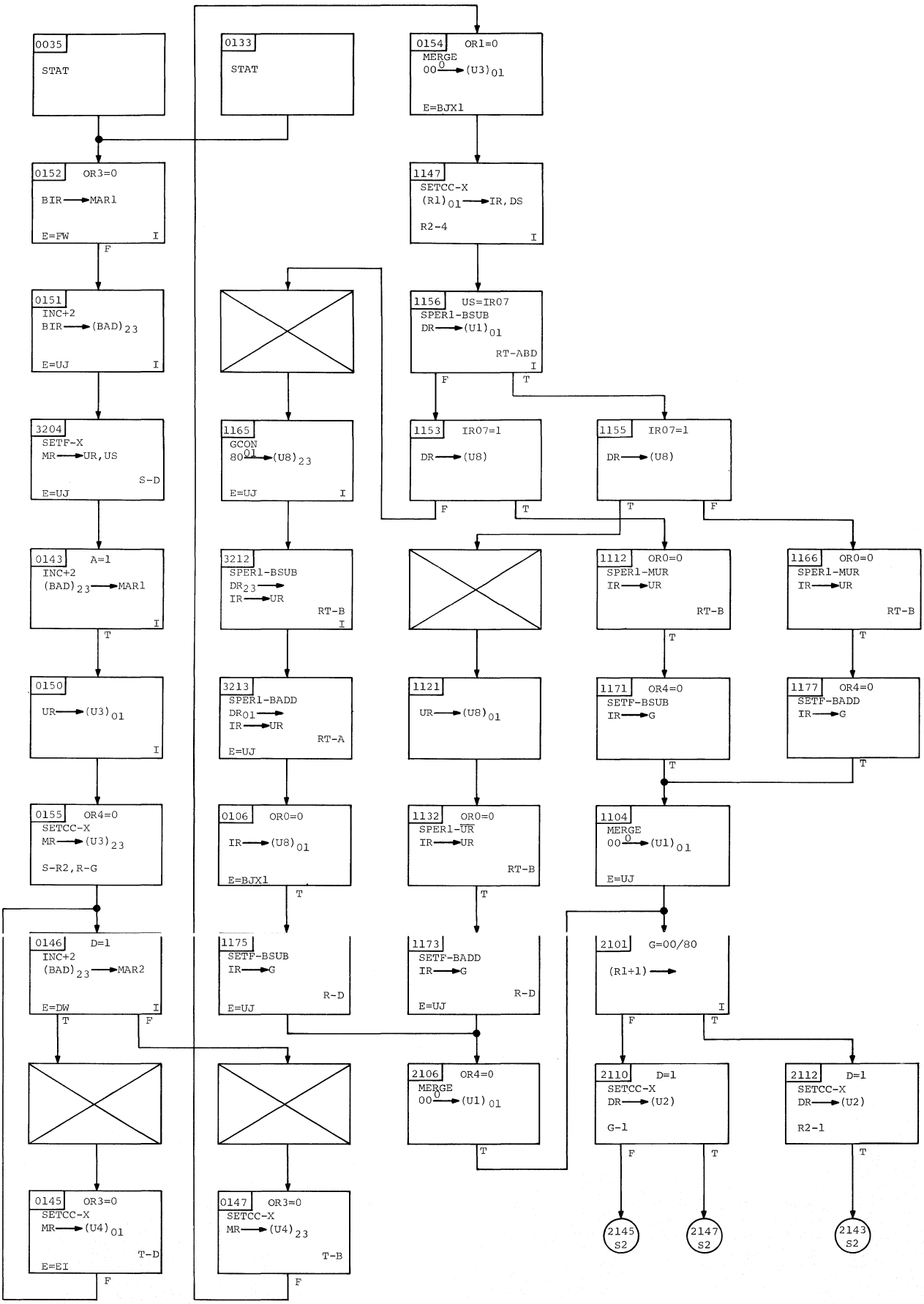
DIVIDE LONG DD RX 6D



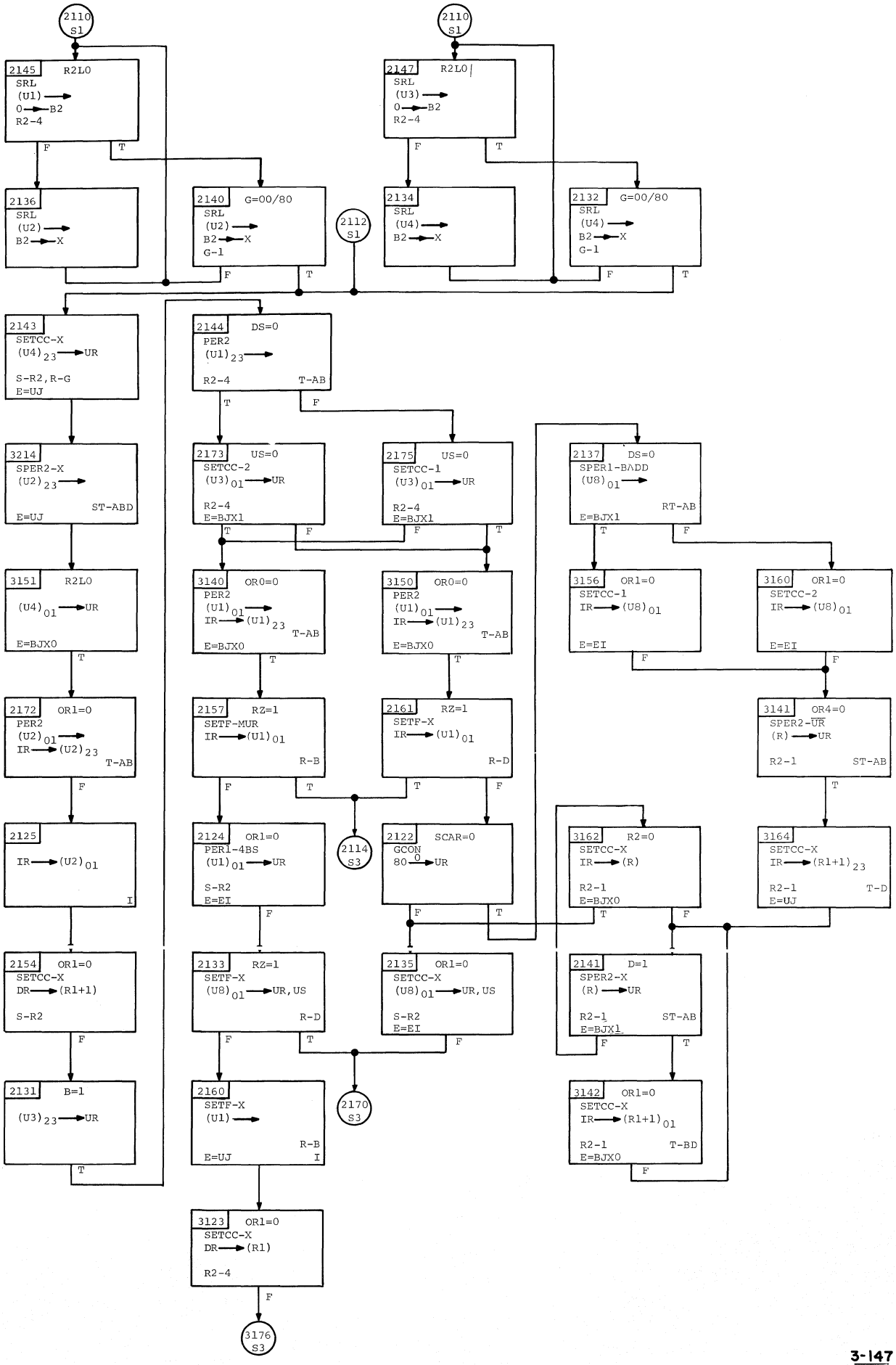
DIVIDE LONG DD RX 6D



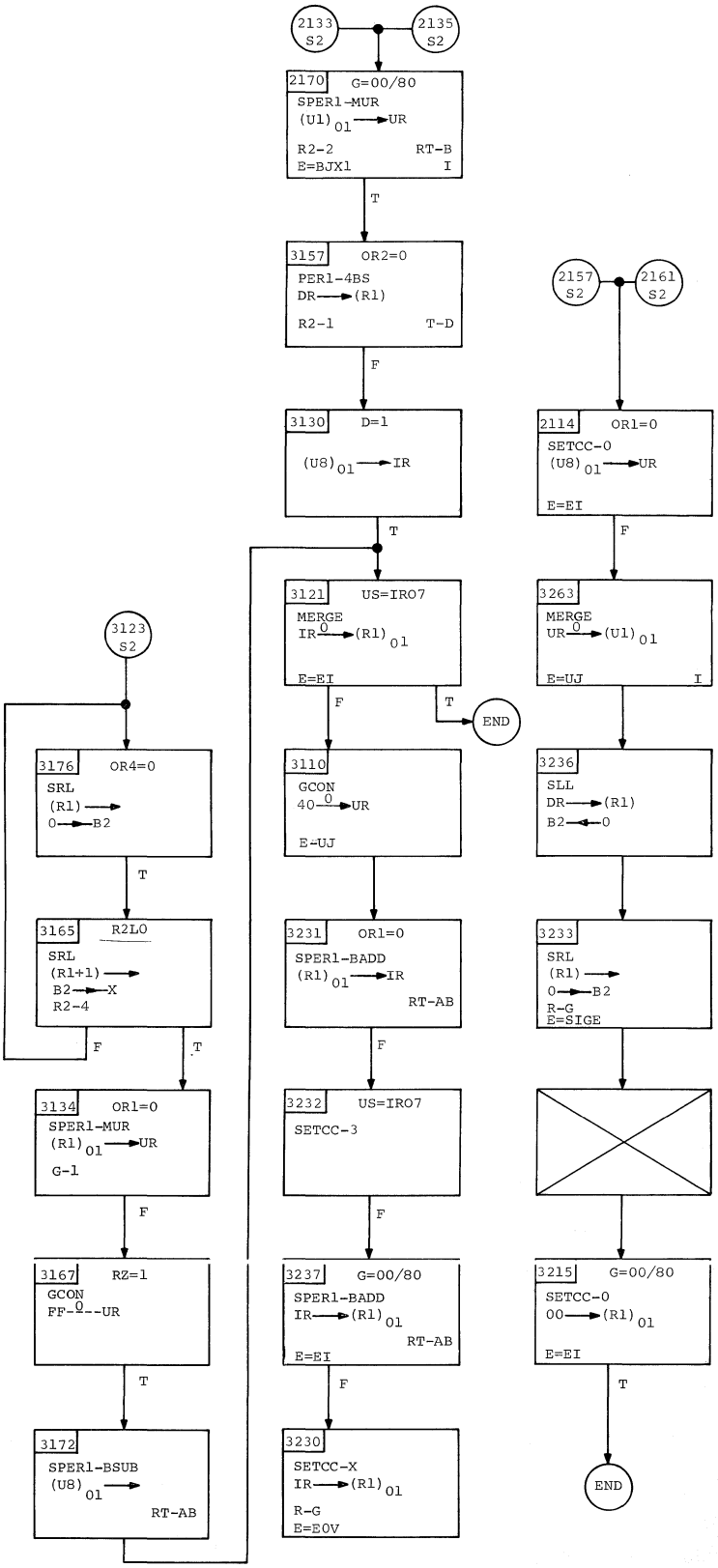
ADD UNNORMALIZED LONG AW RX 6E



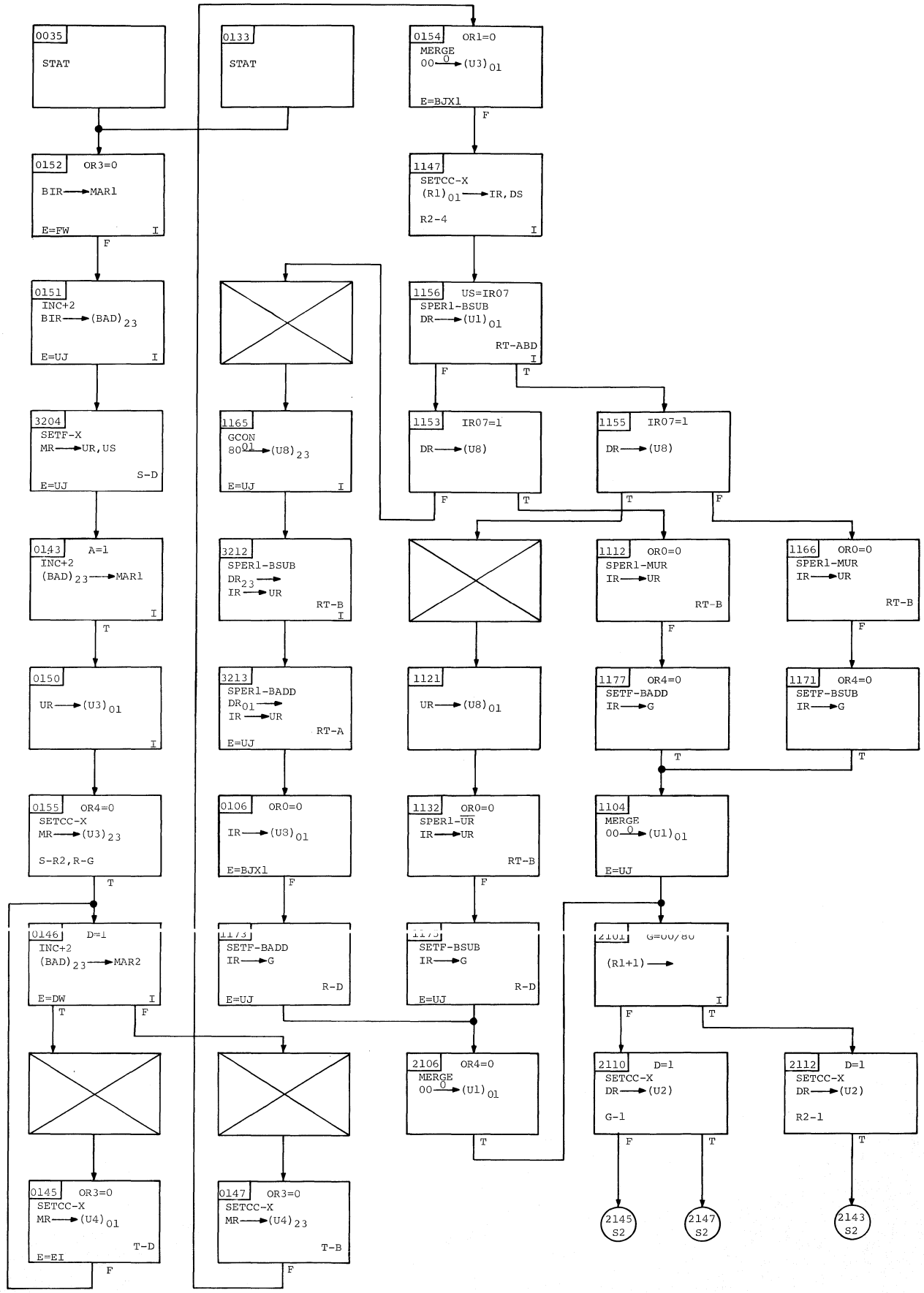
ADD UNNORMALIZED LONG AW RX 6E



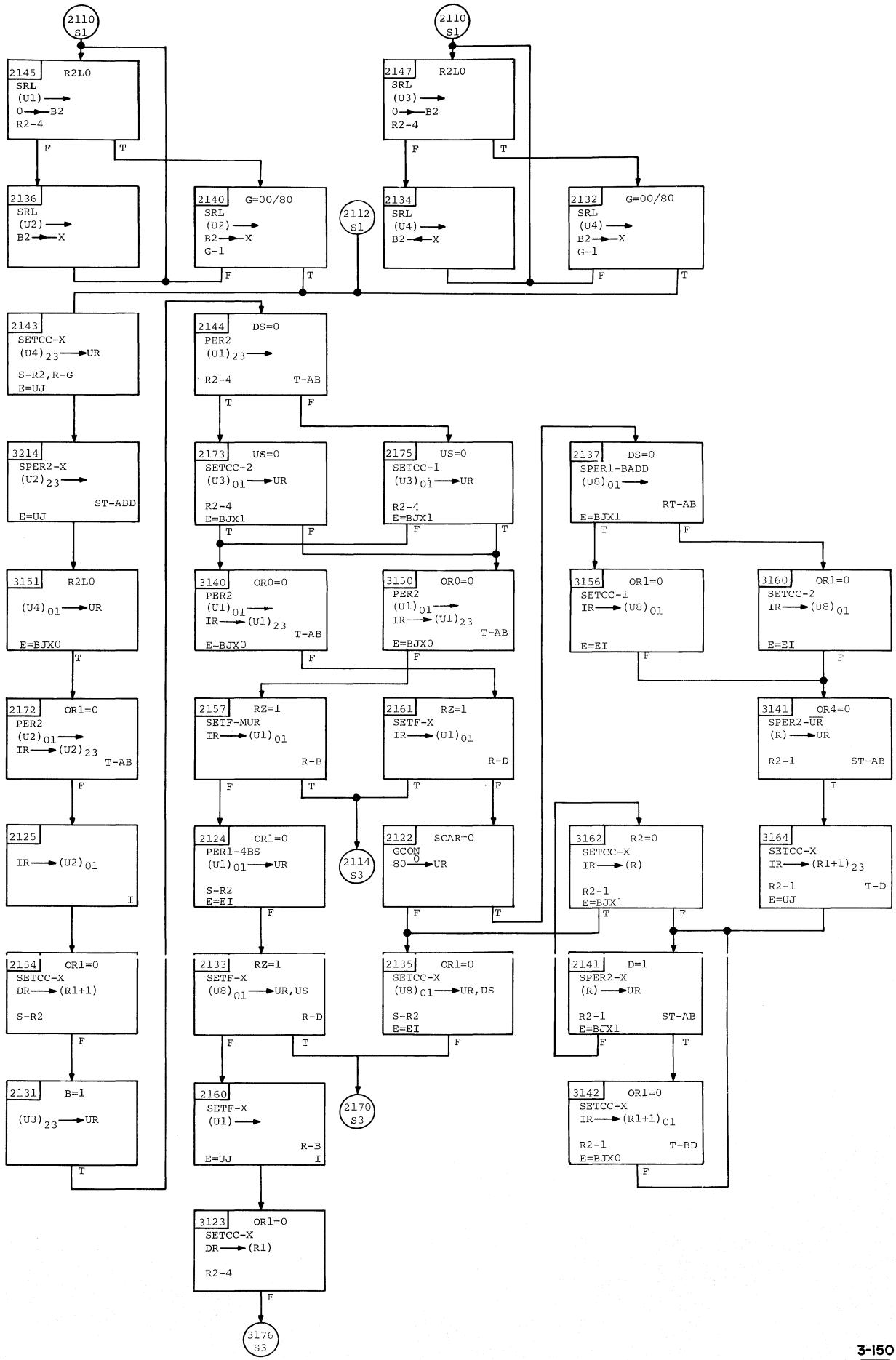
ADD UNNORMALIZED LONG AW RX 6E



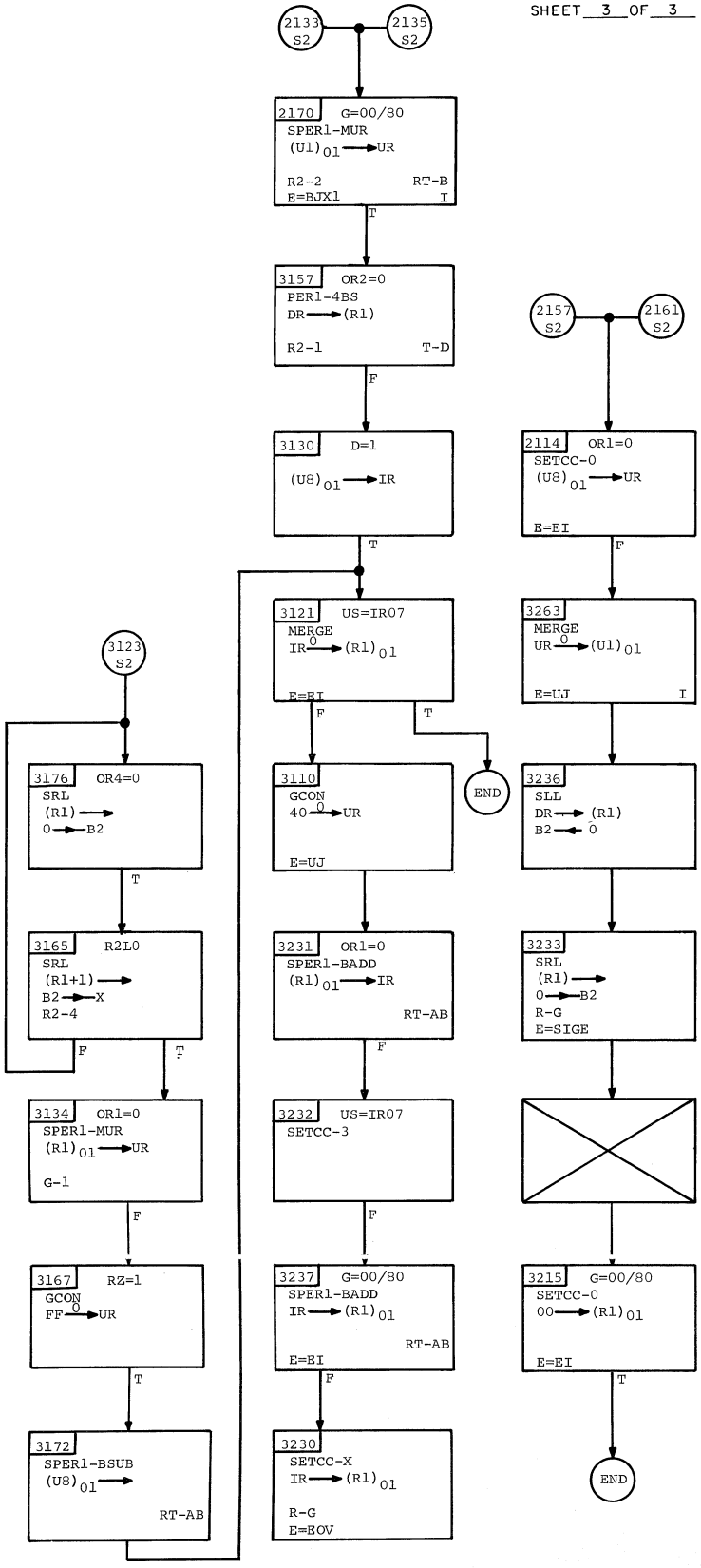
SUBTRACT UNNORMALIZED LONG SW RX 6F



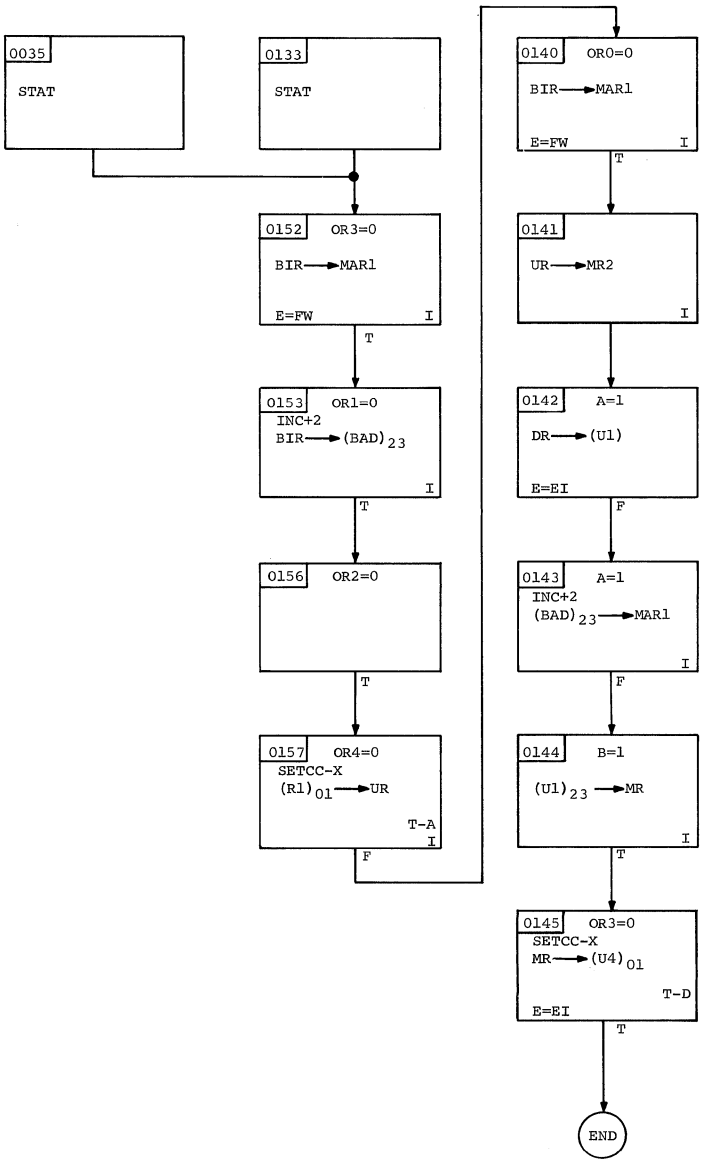
SUBTRACT UNNORMALIZED LONG SW RX 6F



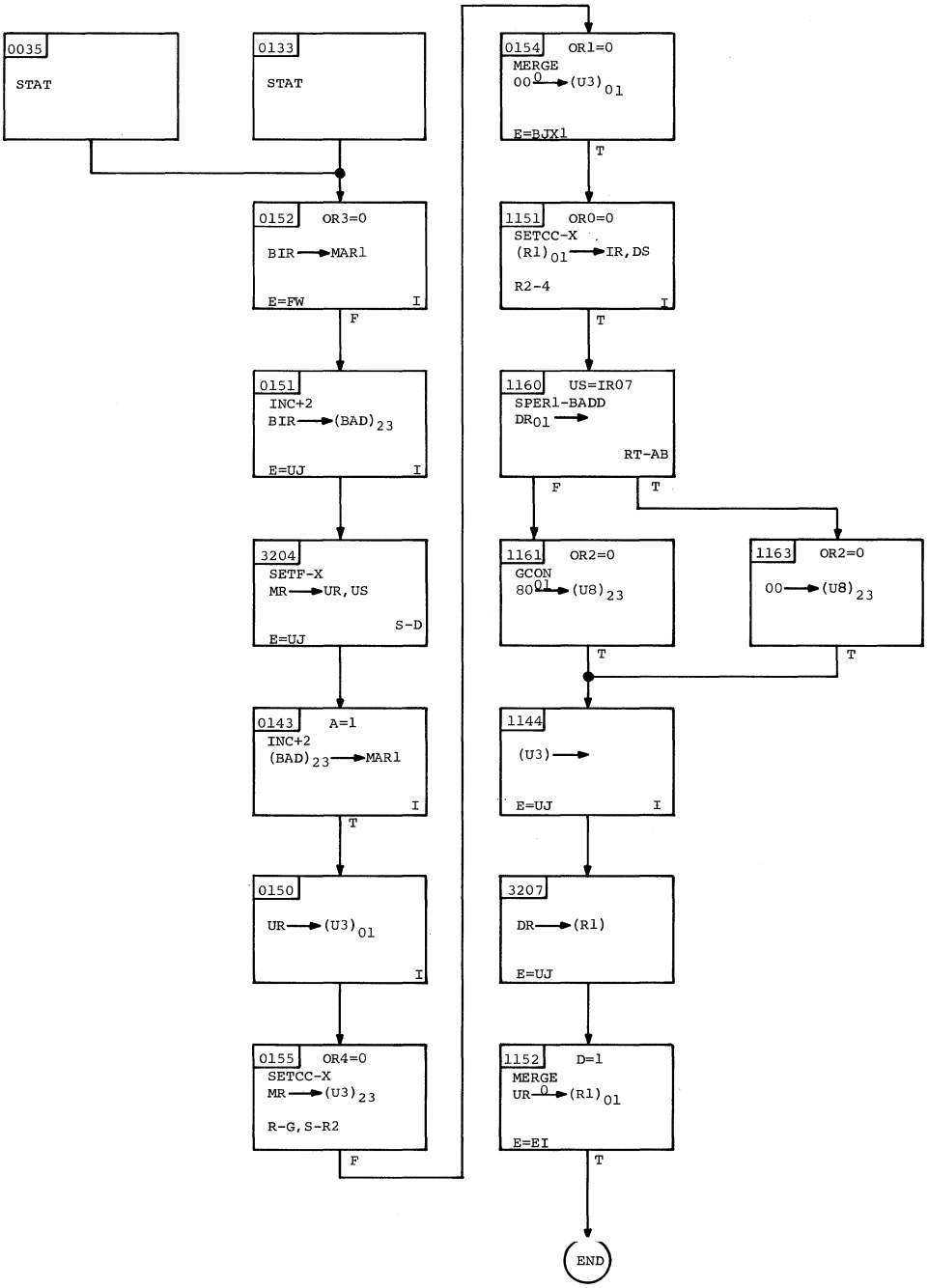
SUBTRACT UNNORMALIZED LONG SW RX 6F



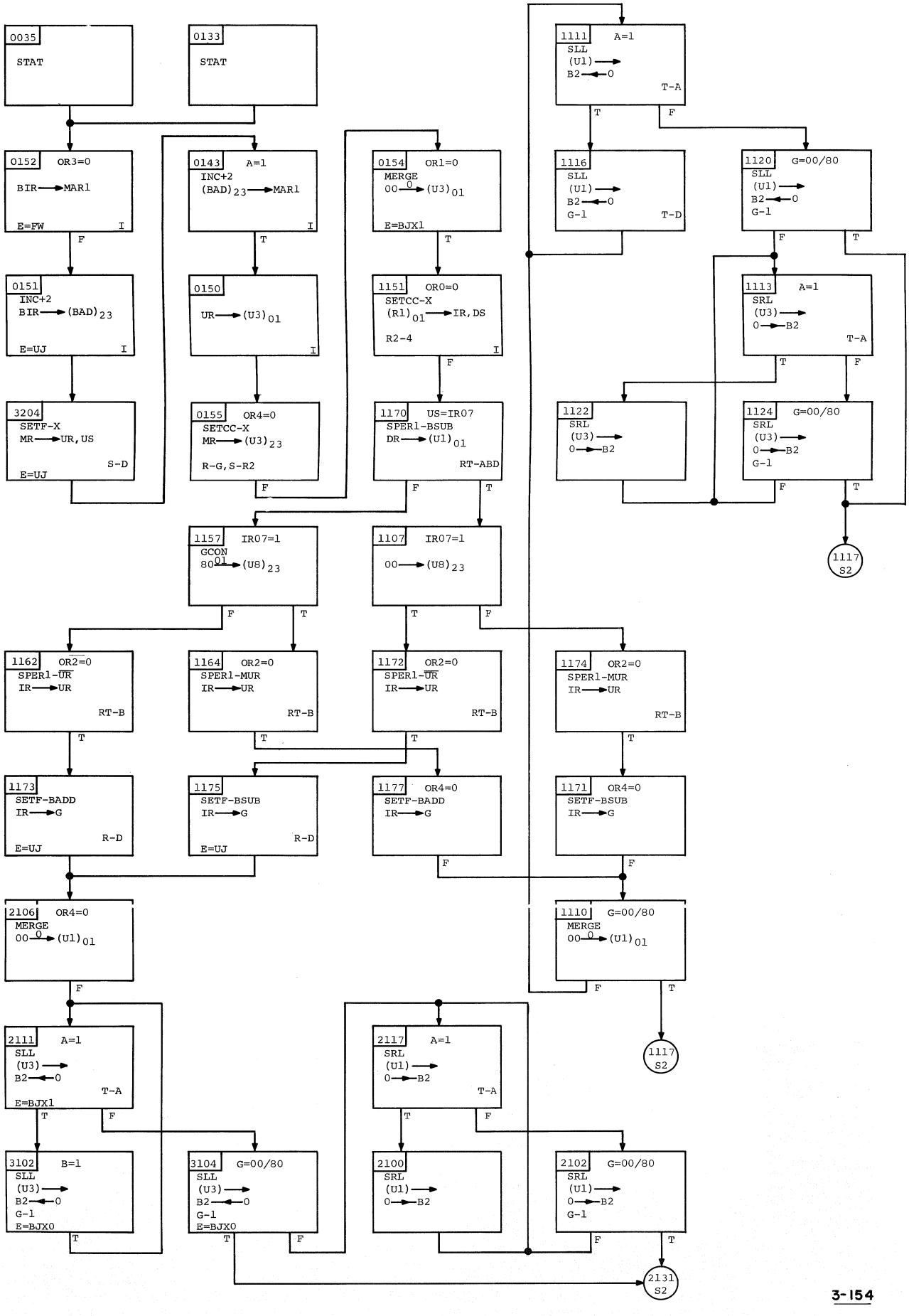
STORE SHORT STE RX 70



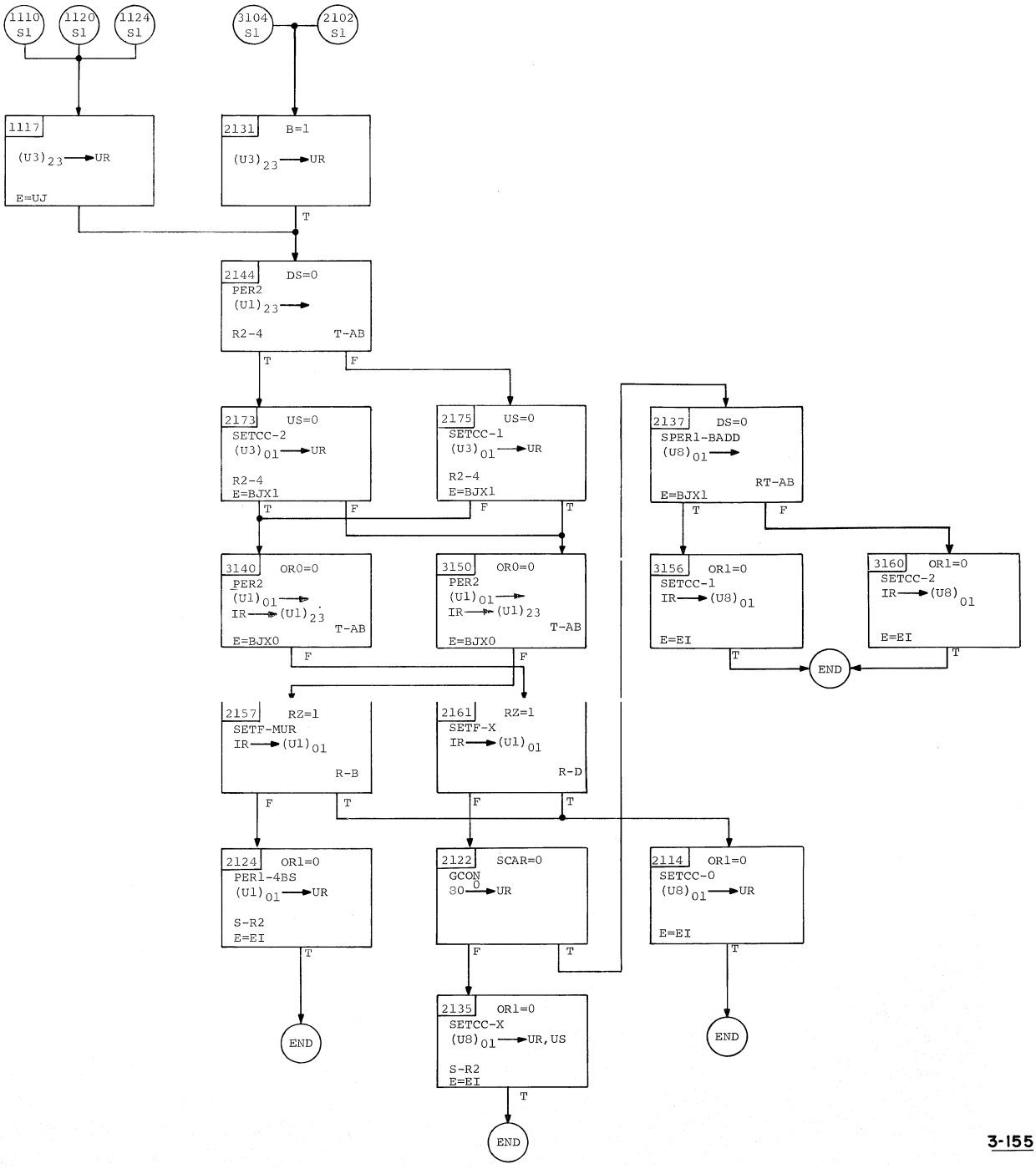
LOAD SHORT LE RX 78



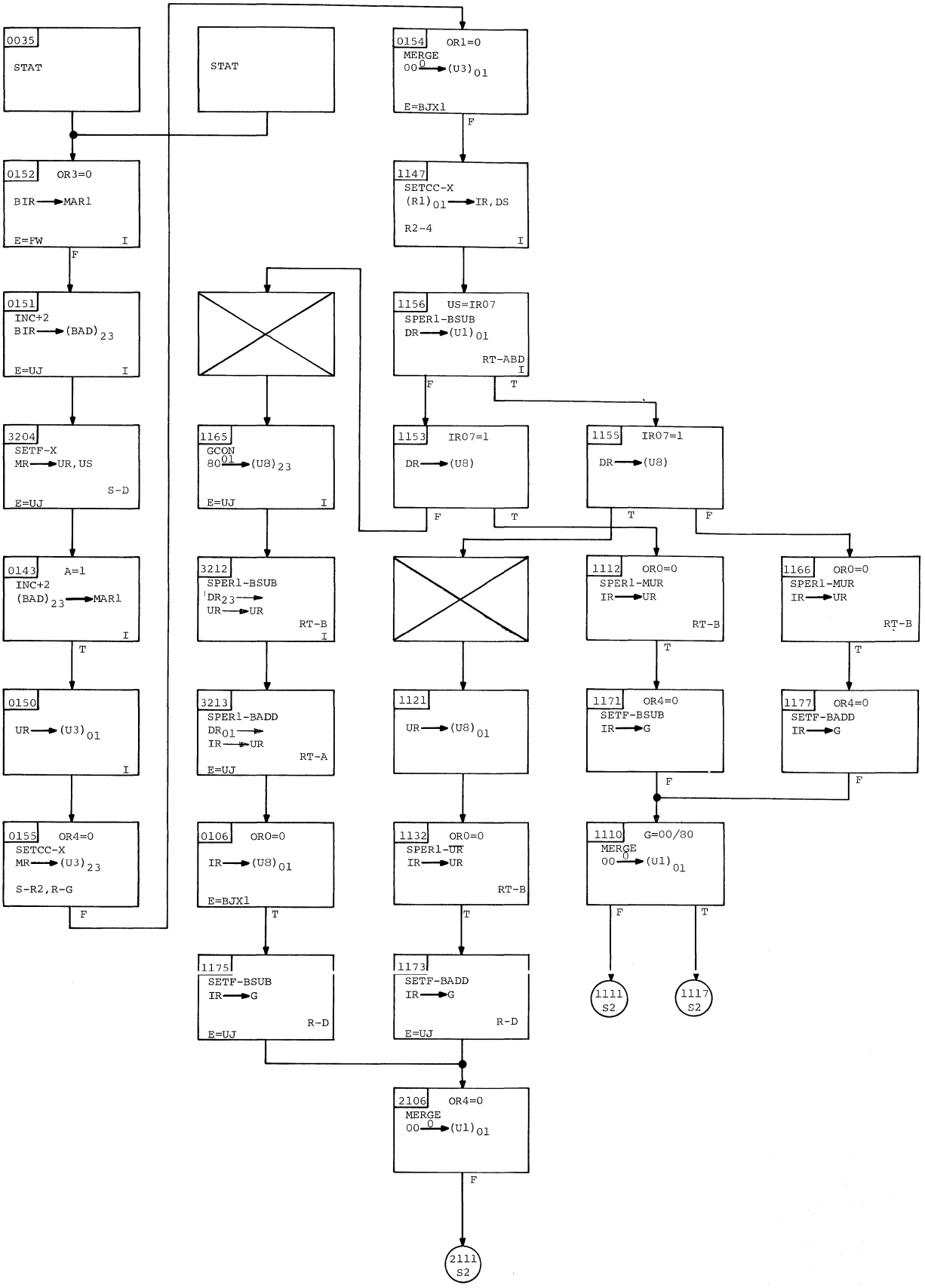
COMPARE SHORT CE RX 79



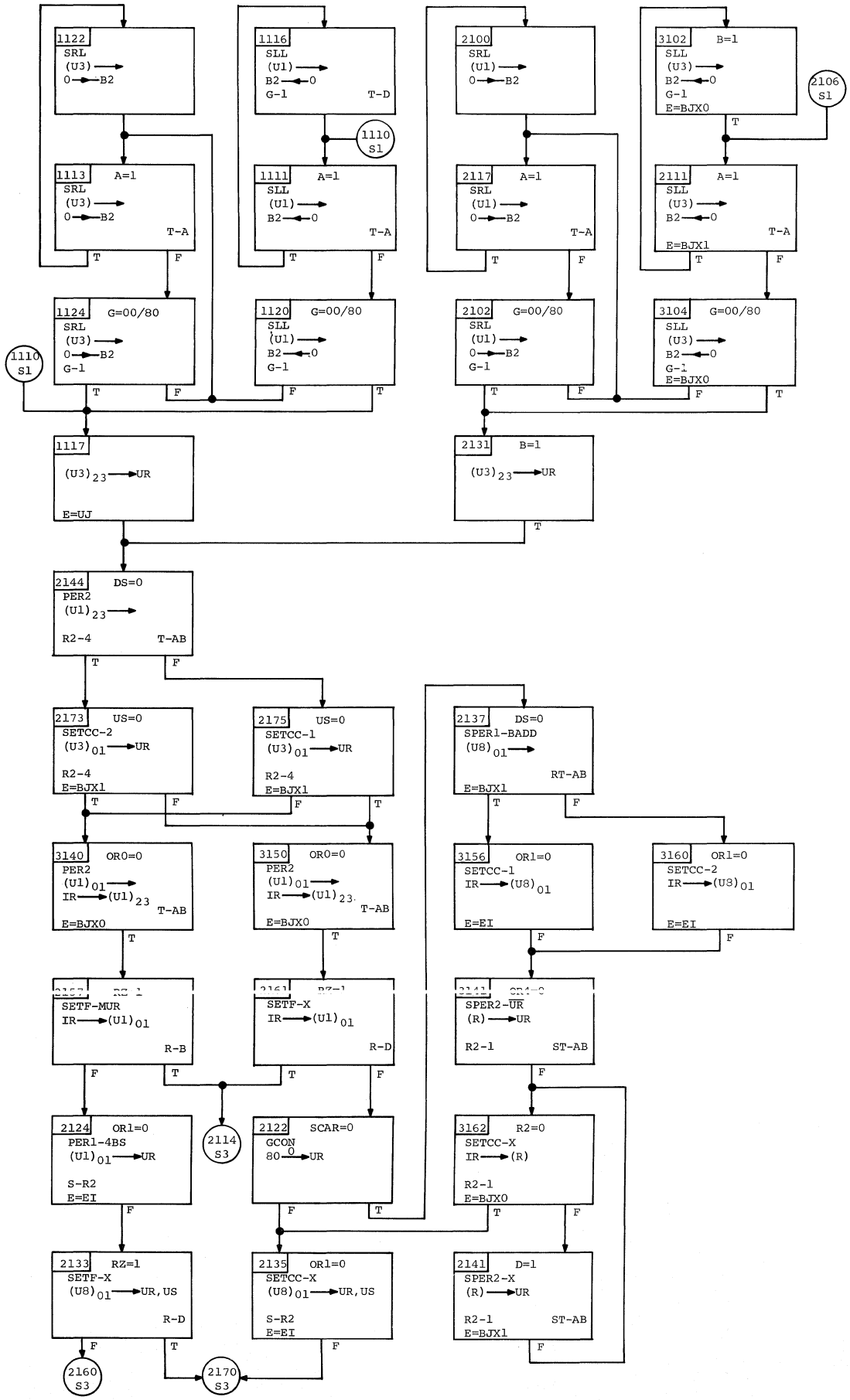
COMPARE SHORT CE RX 79



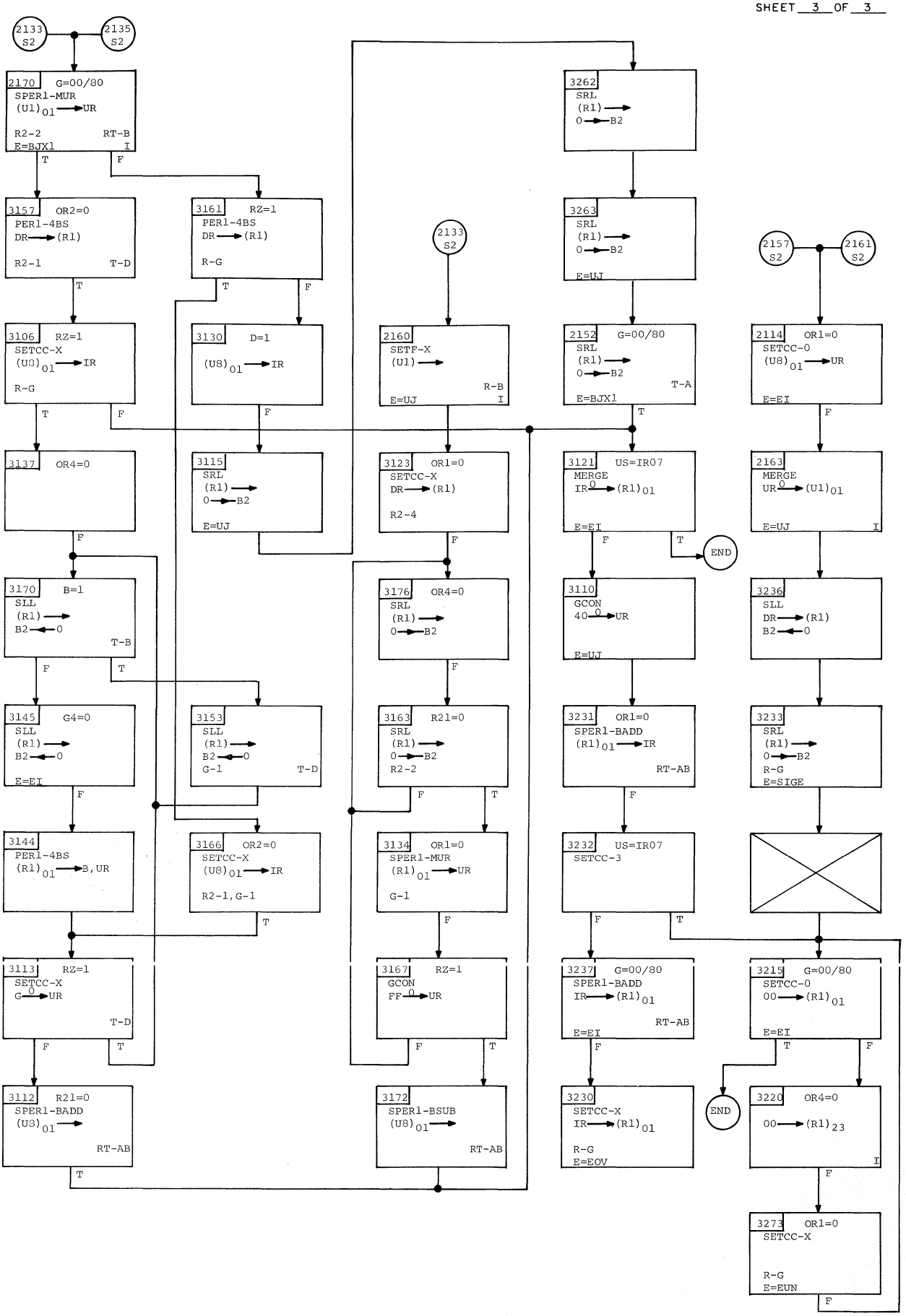
ADD NORMALIZED SHORT ANE RX 7A



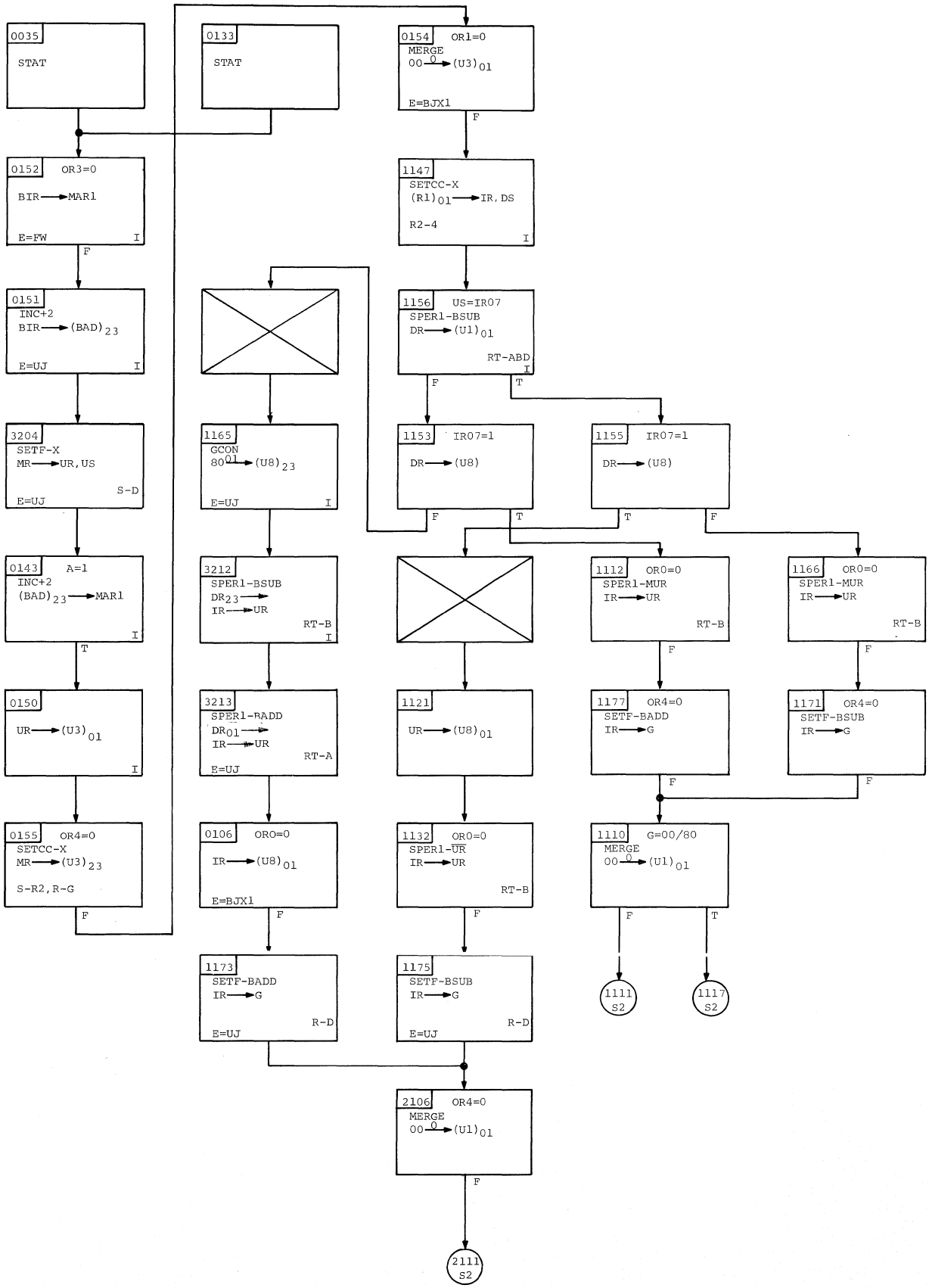
ADD NORMALIZED SHORT ANE RX 7A



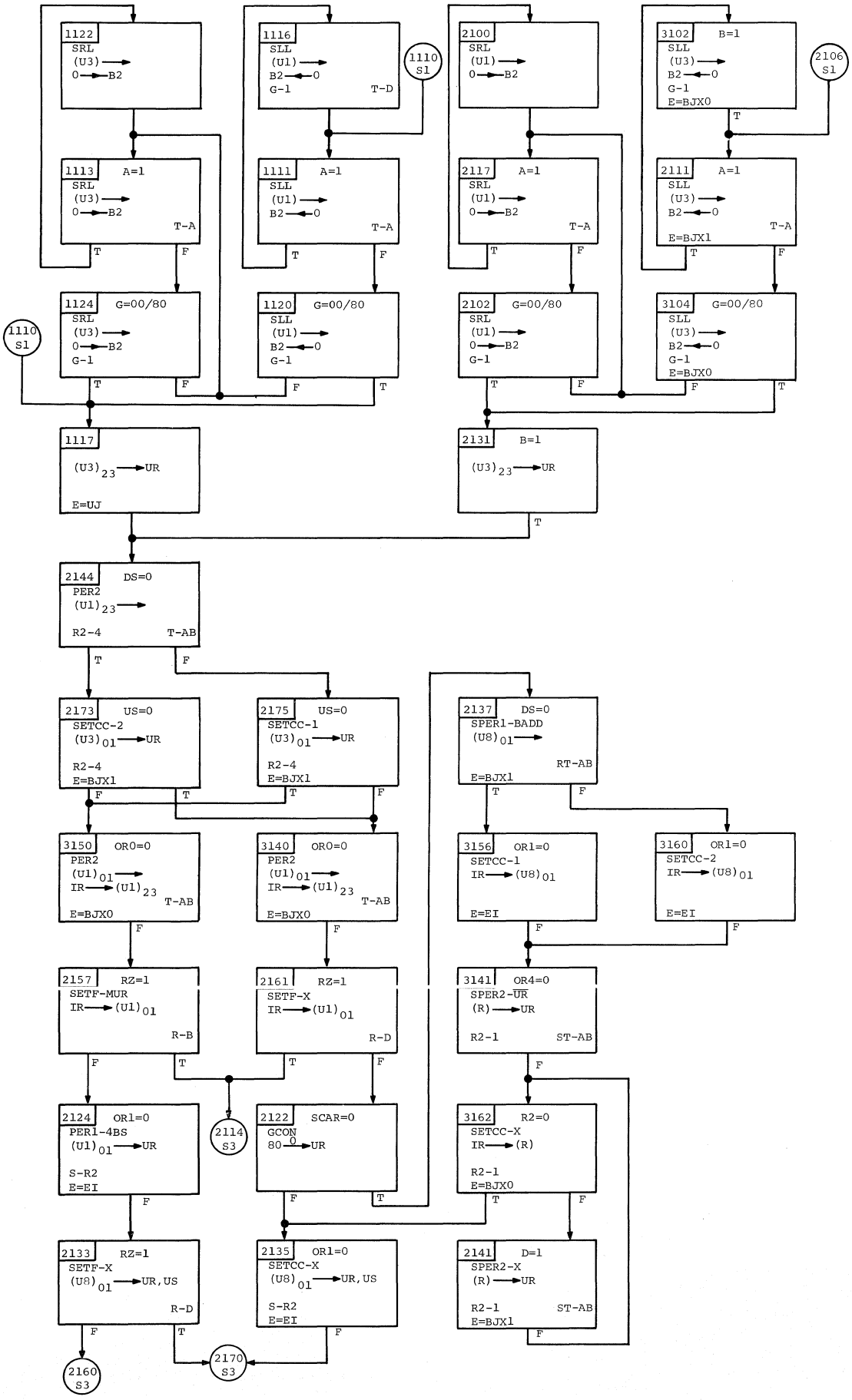
ADD NORMALIZED SHORT ANE RX 7A



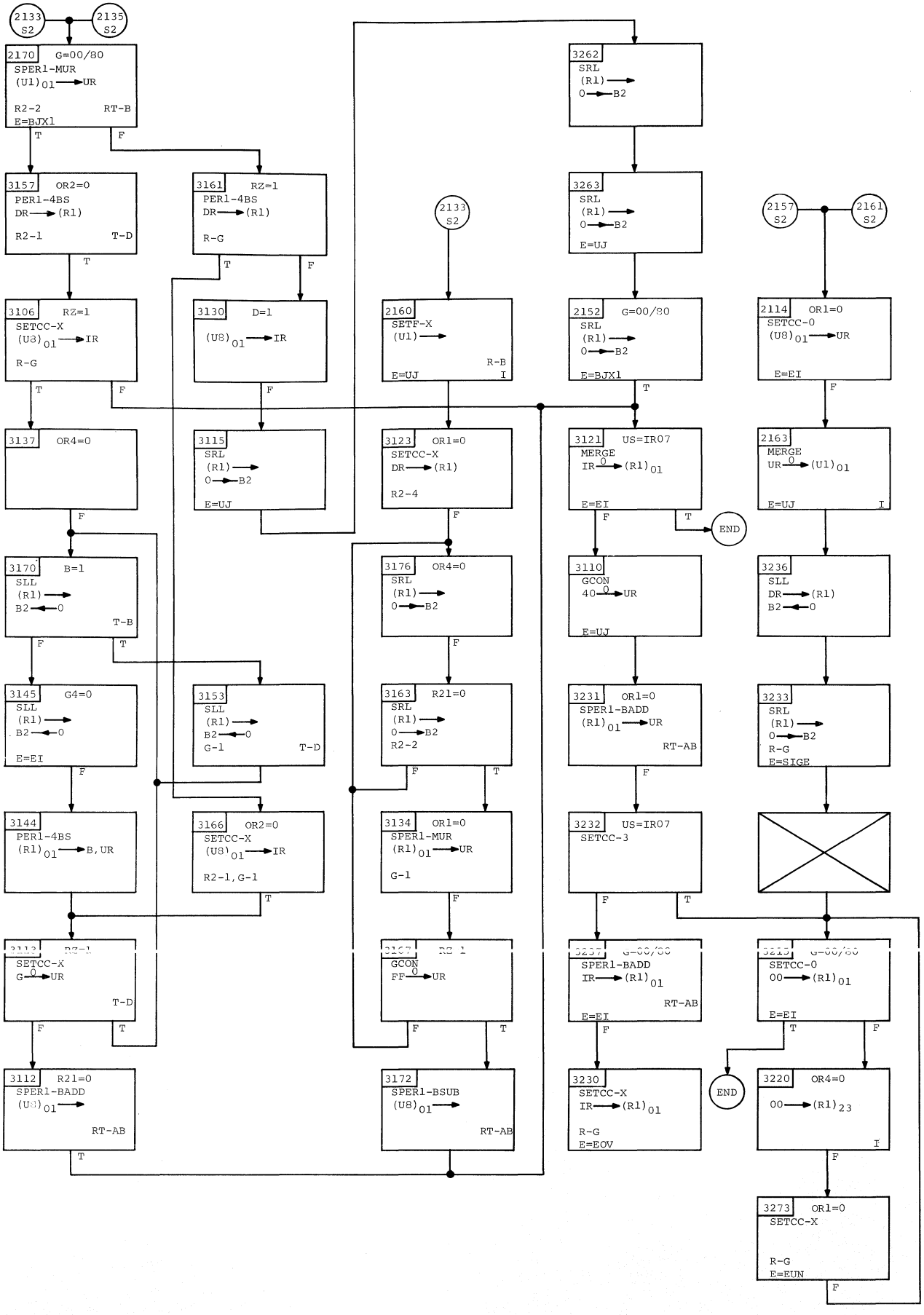
SUBTRACT NORMALIZED SHORT SNE RX 7B



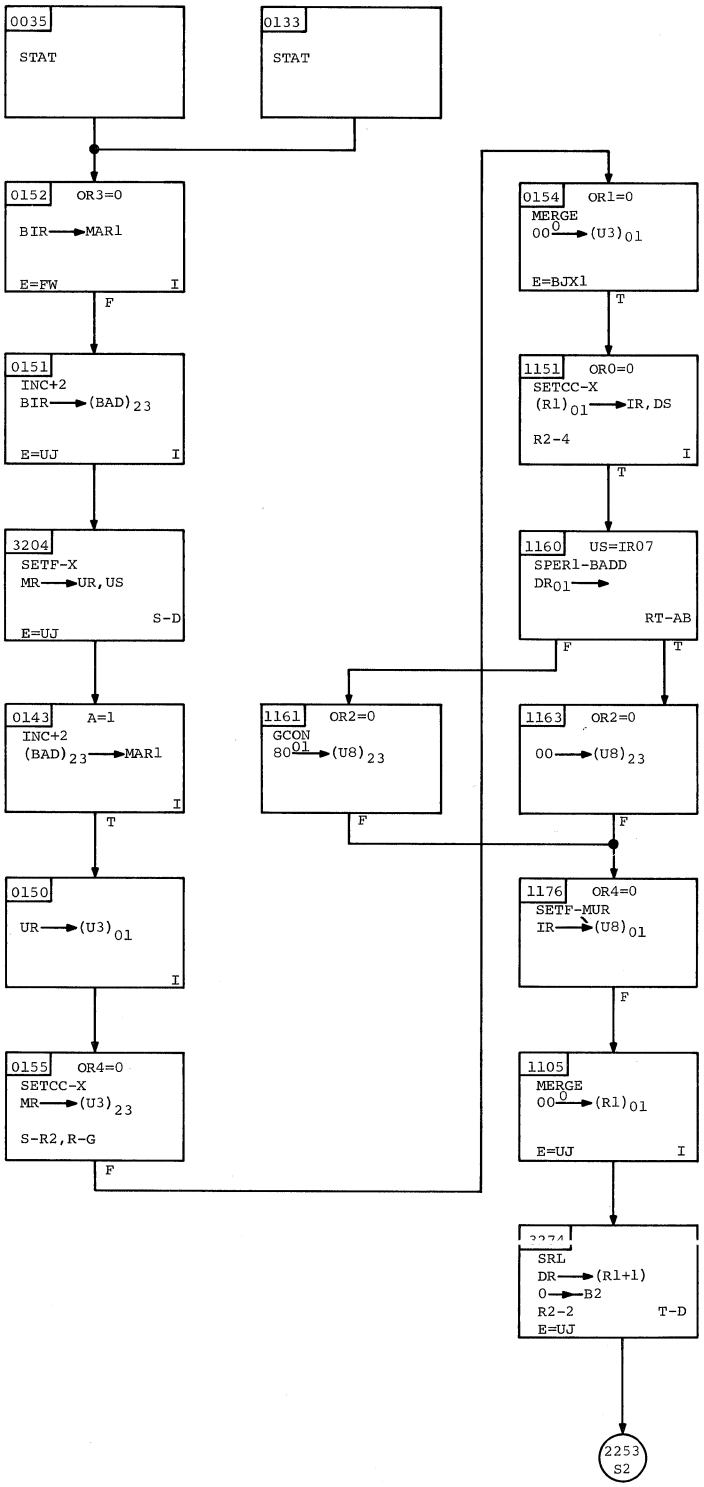
SUBTRACT NORMALIZED SHORT SNE RX 7B



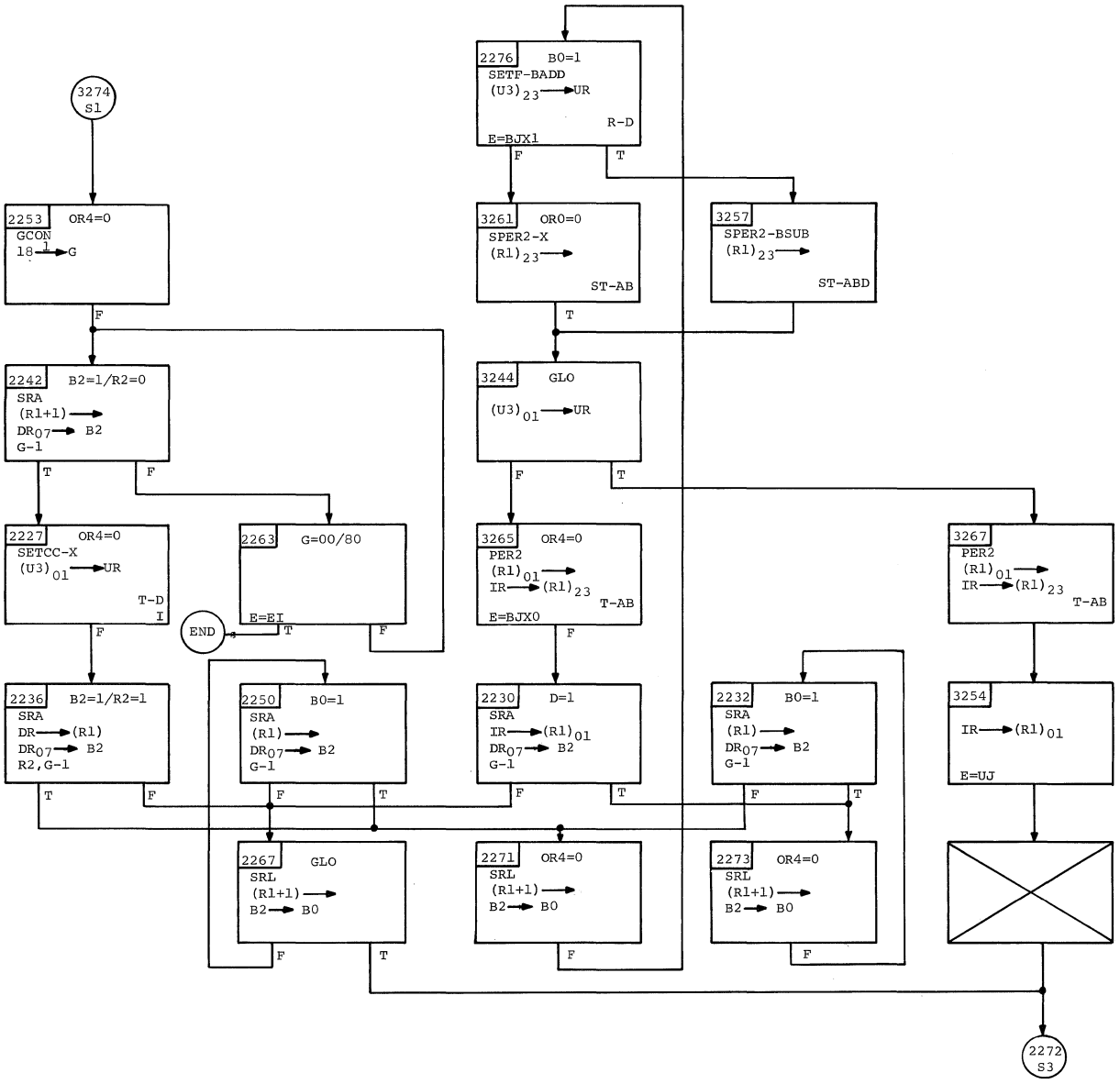
SUBTRACT NORMALIZED SHORT SNE RX 7B



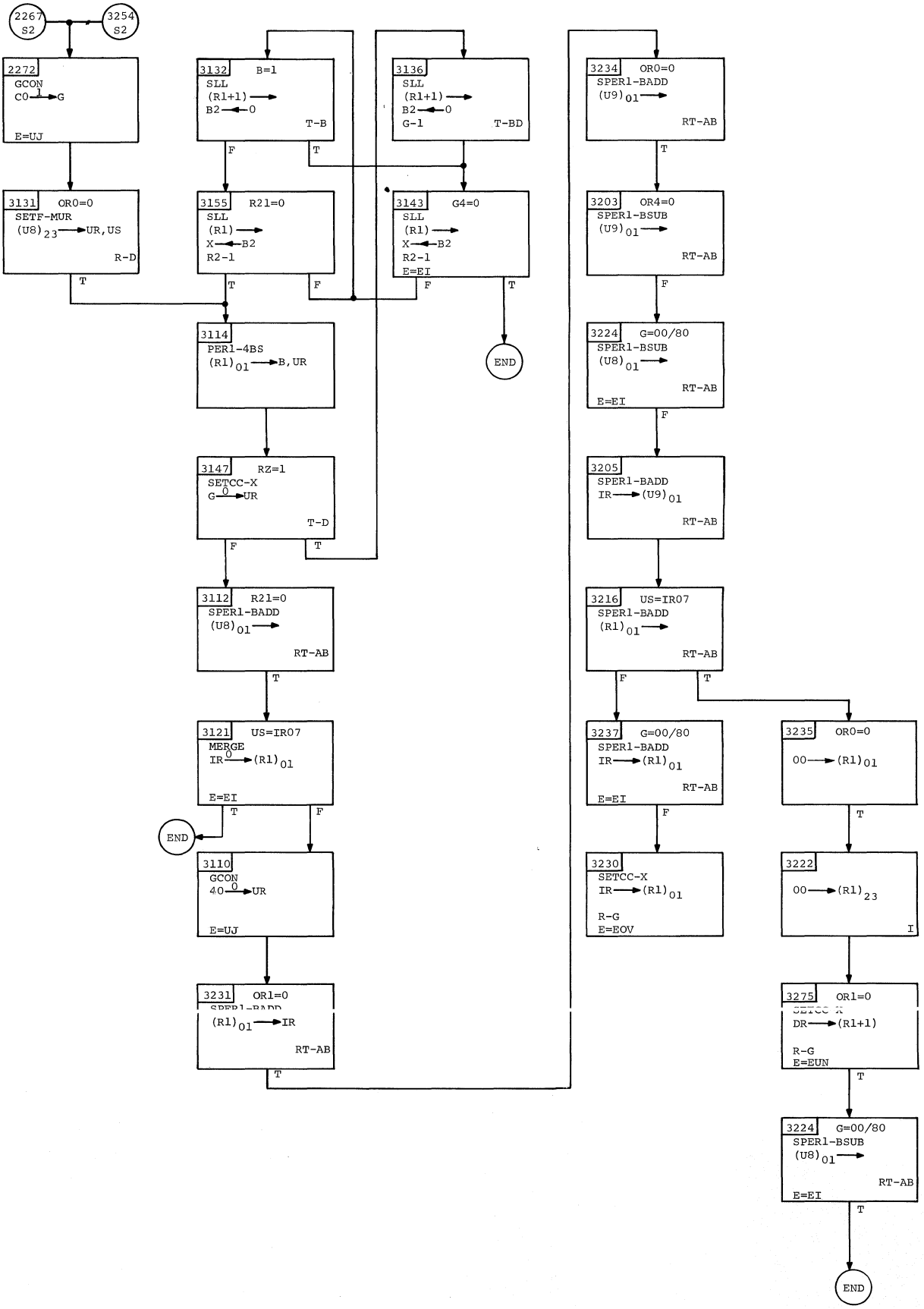
MULTIPLY SHORT ME RX 7C



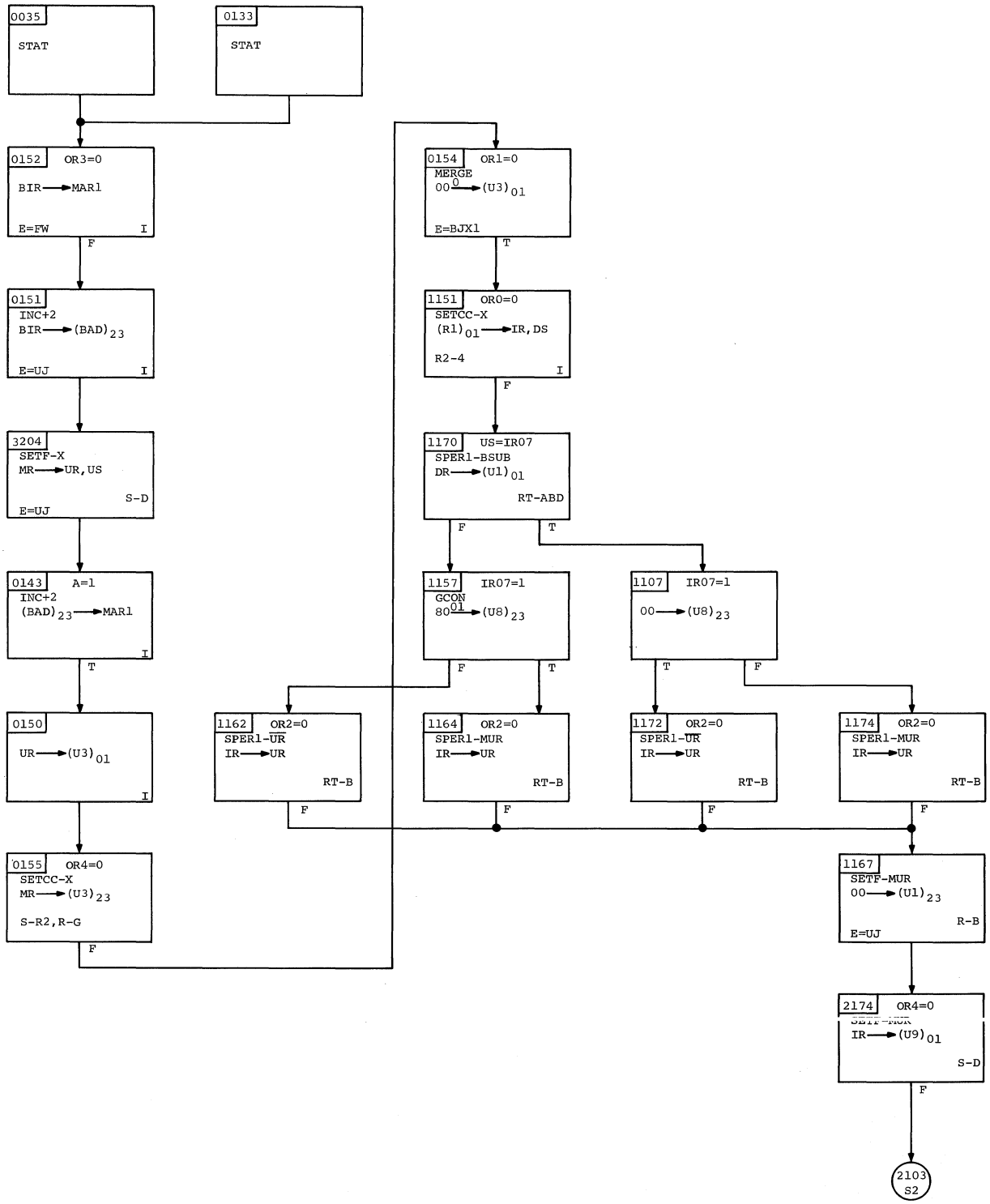
MULTIPLY SHORT ME RX 7C

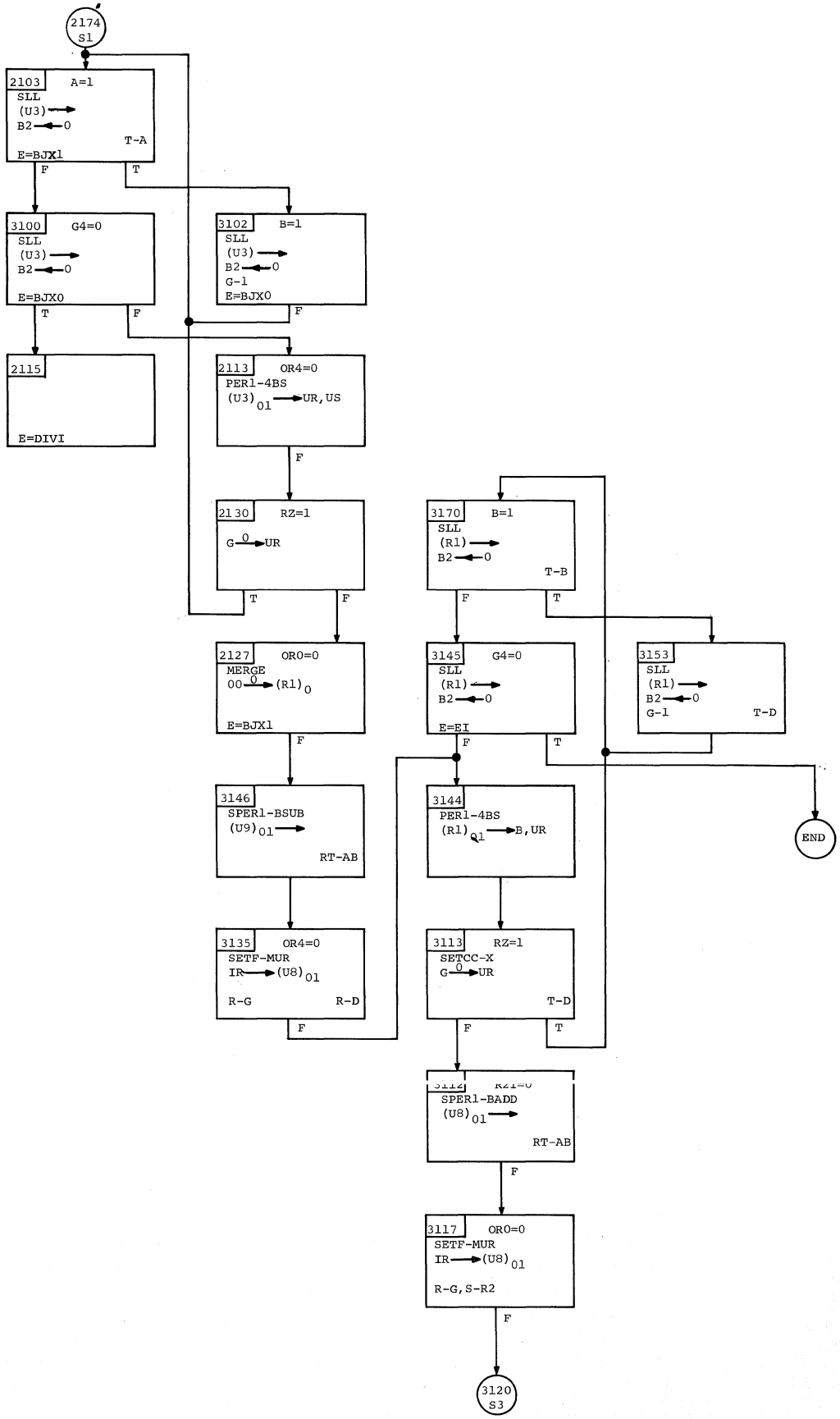


MULTIPLY SHORT ME RX 7C

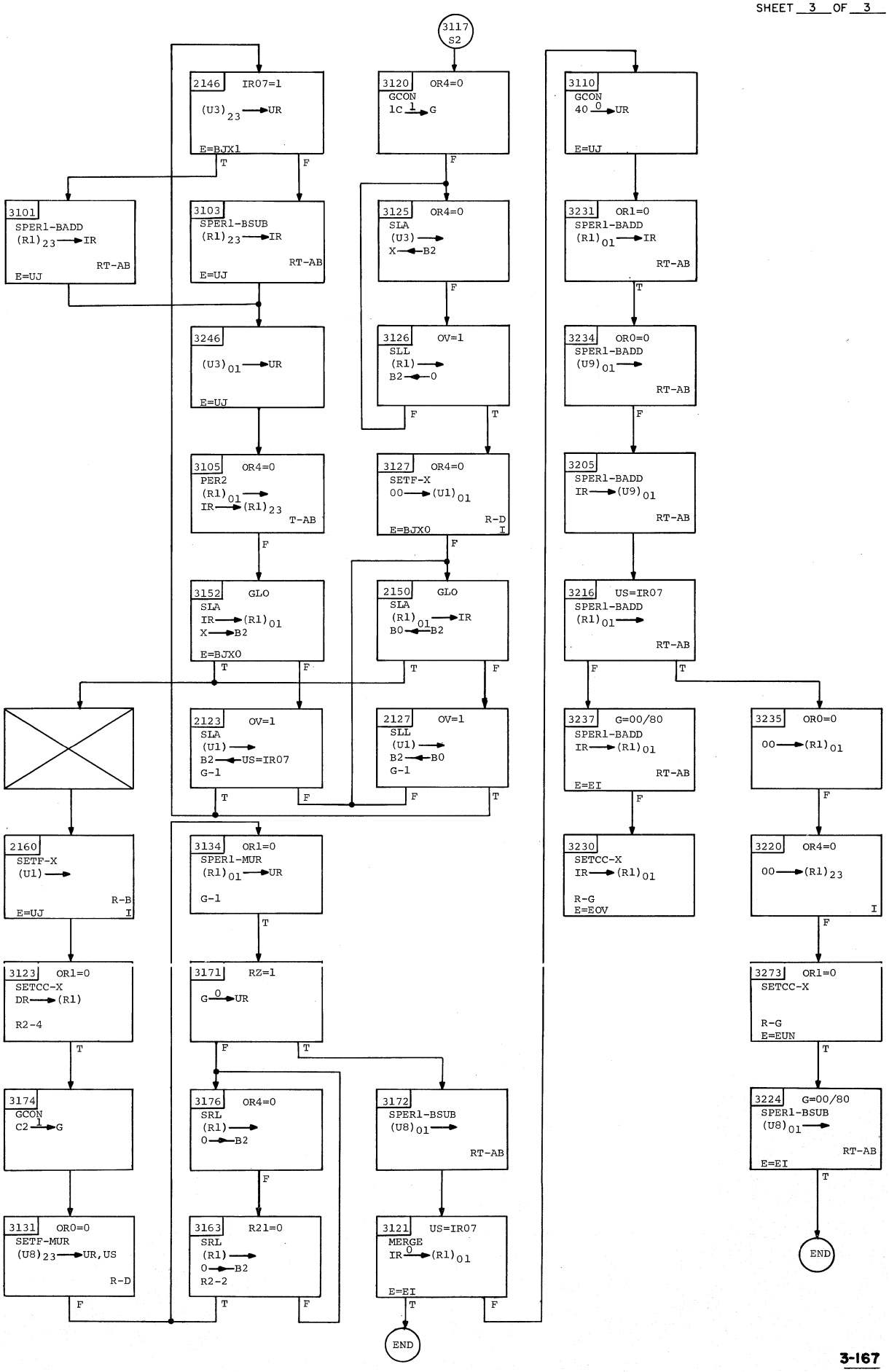


DIVIDE SHORT DE RX 7D

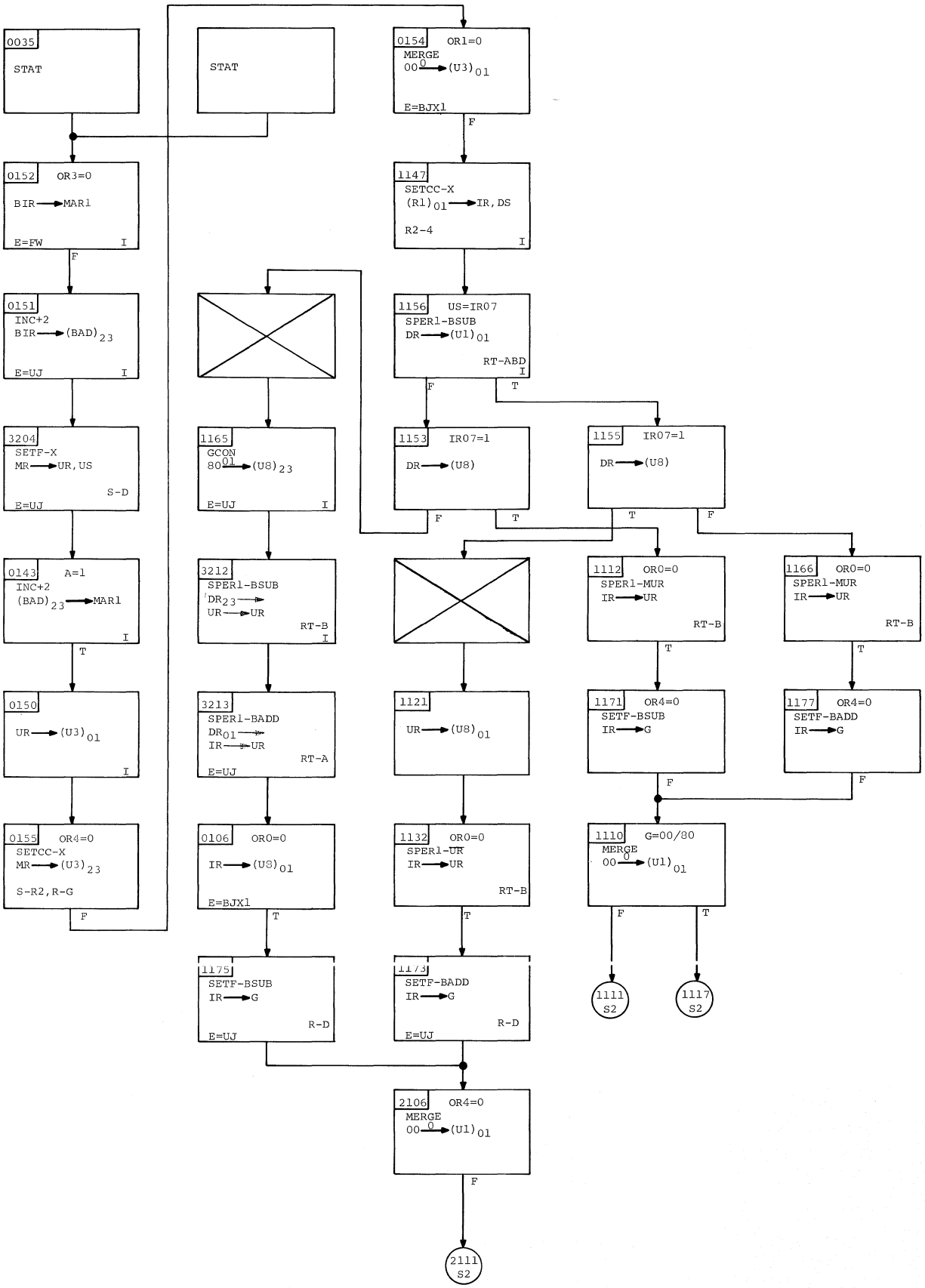



2174
S1
2103 A=1
SLL
(U3)
B2 0
T-A
E=BJX1
F
T
3100 G4=0
SLL
(U3)
B2 0
E=BJX0
T
F
3102 B=1
SLL
(U3)
B2 0
G-1
E=BJX0
F
2115
E=DIVI
2113 OR4=0
PER1-4BS
(U3) 01 UR,US
F
2130 RZ=1
G 0 UR
T
F
2127 OR0=0
MERGE
00 0 (R1) 0
E=BJX1
F
3146
SPER1-BSUB
(U9) 01
RT-AB
3135 OR4=0
SETF-MUR
IR (U8) 01
R-G R-D
F
3170 B=1
SLL
(R1)
B2 0
T-B
F
T
3145 G4=0
SLL
(R1)
B2 0
E=EI
F
T
3153
SLL
(R1)
B2 0
G-1 T-D
3144
PER1-4BS
(R1) 01 B,UR
END
3113 RZ=1
SETCC-X
G 0 UR
T-D
F
T
SPER1-BADD
(U8) 01
RT-AB
F
3117 OR0=0
SETF-MUR
IR (U8) 01
R-G,S-R2
F
3120
S3

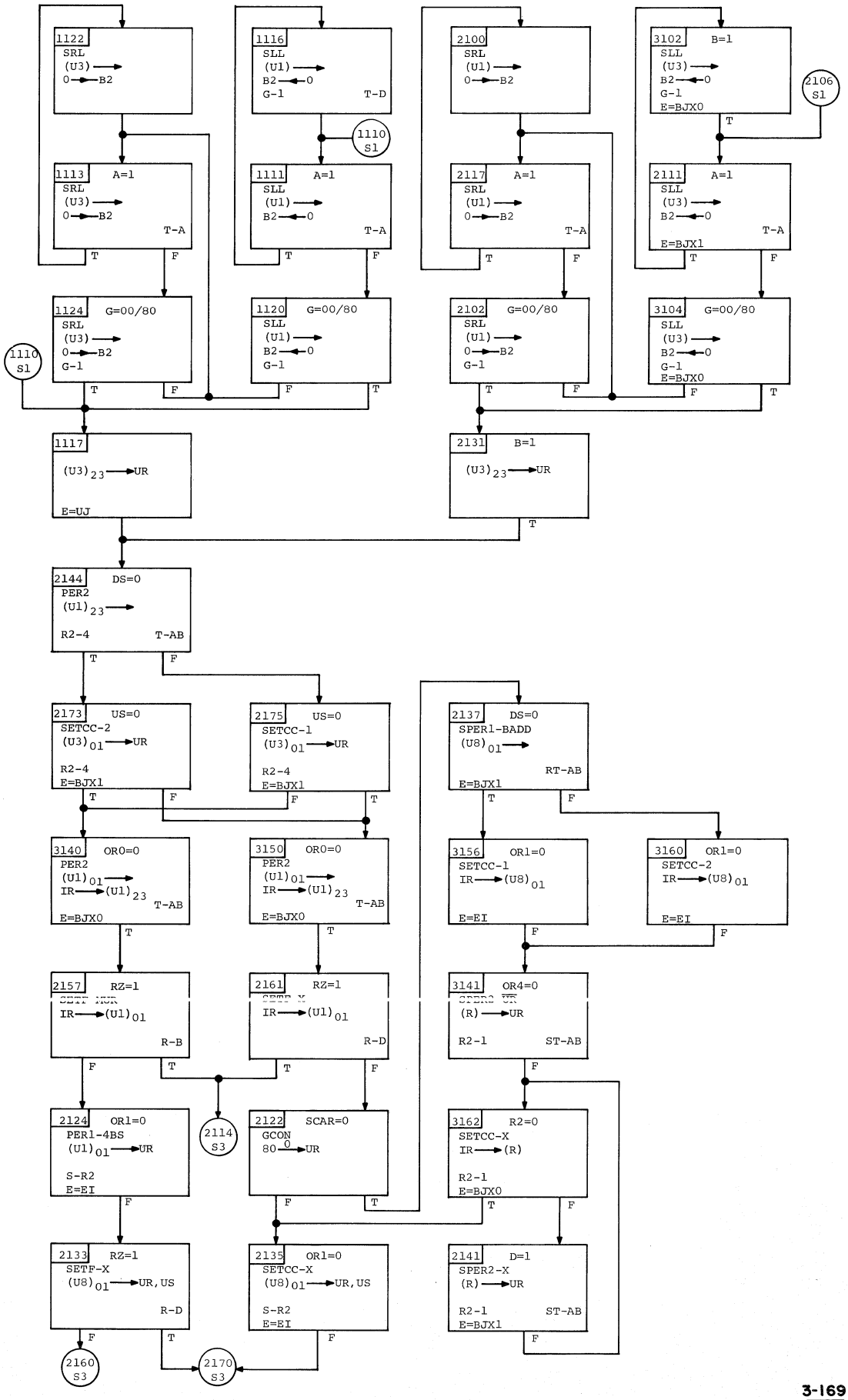
DIVIDE SHORT DE RX 7D



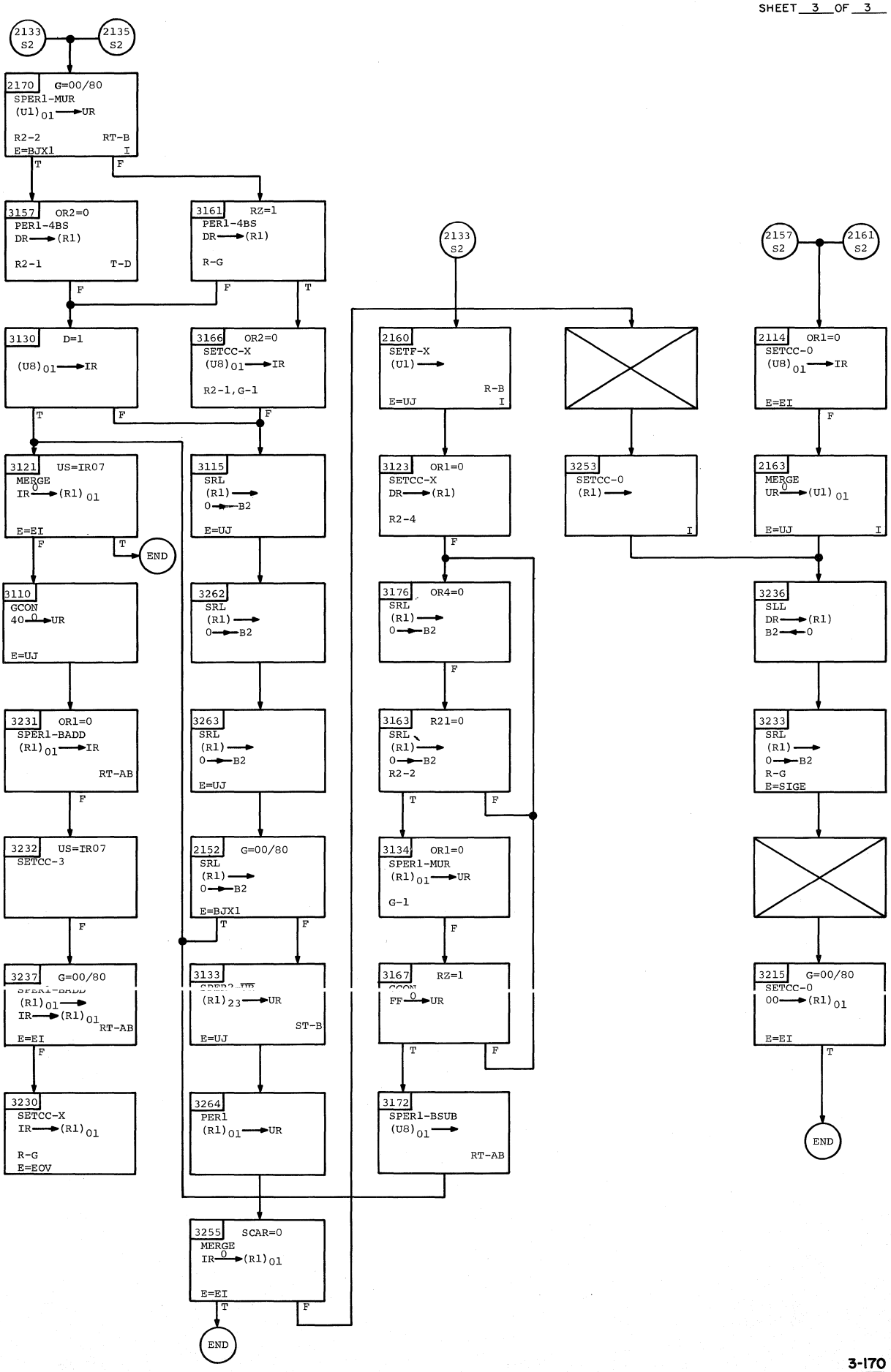
ADD UNNORMALIZED SHORT AU RX 7E



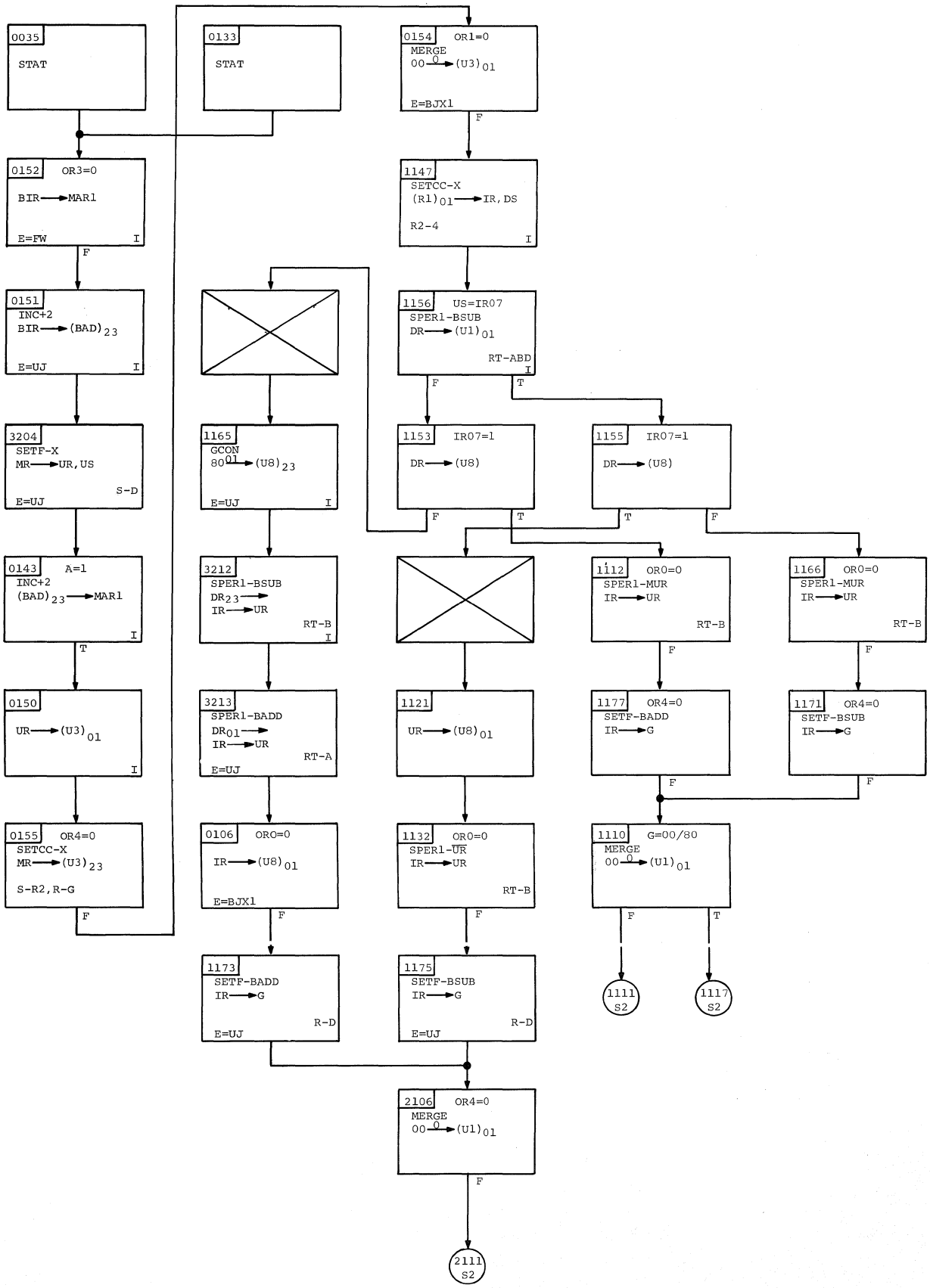
ADD UNNORMALIZED SHORT AU RX 7E



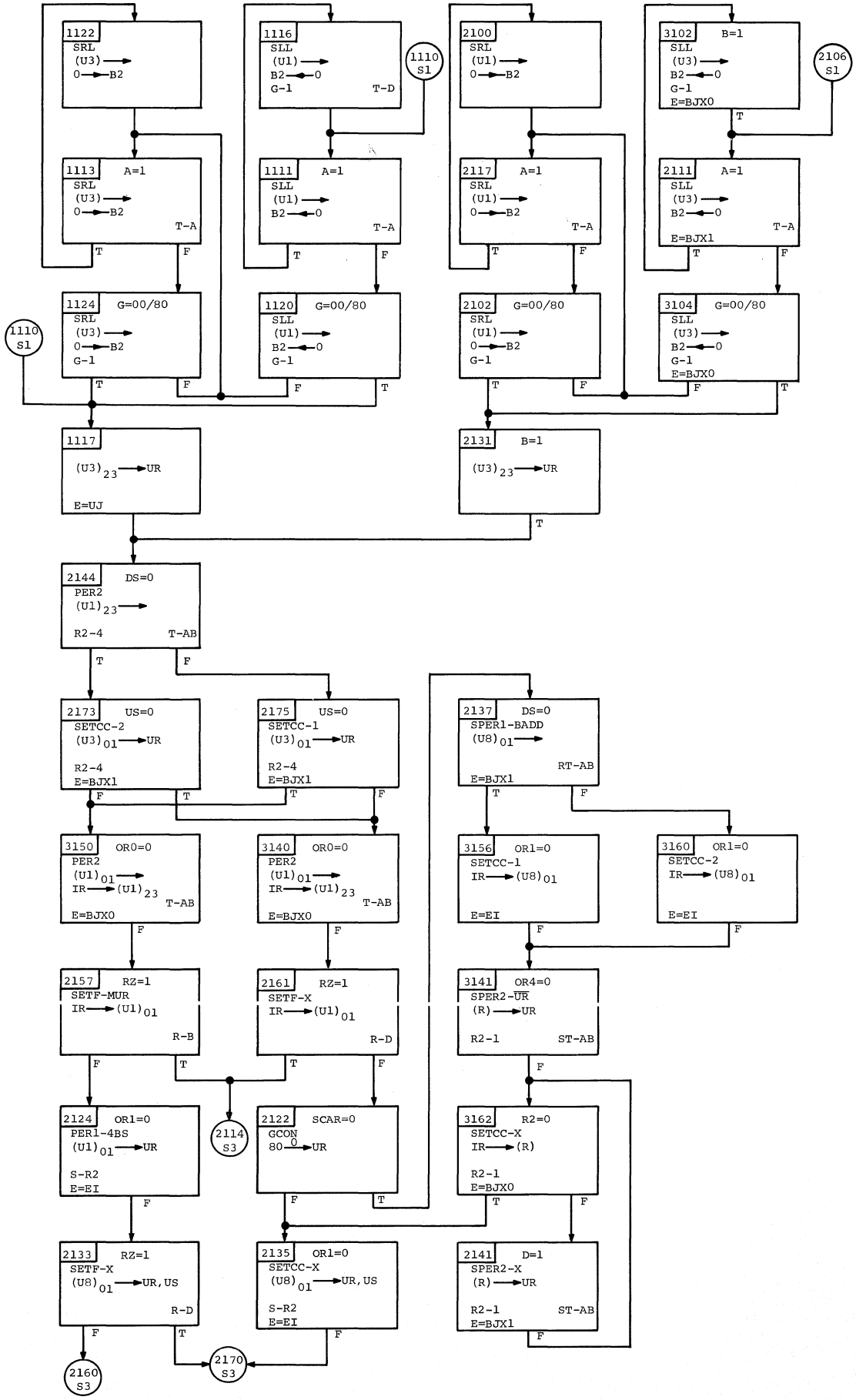
ADD UNNORMALIZED SHORT AU RX 7E



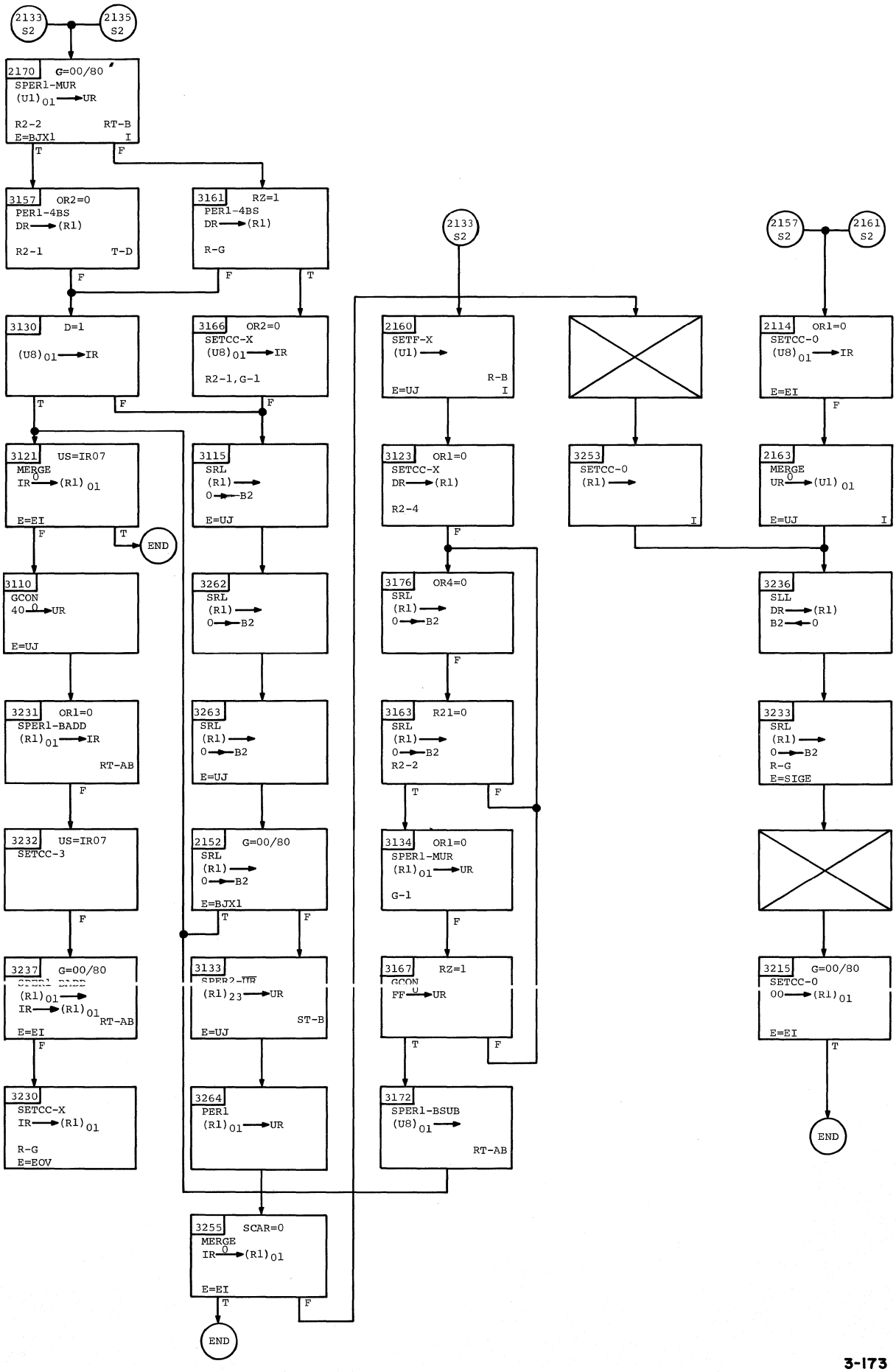
SUBTRACT UNNORMALIZED SHORT SU RX 7F



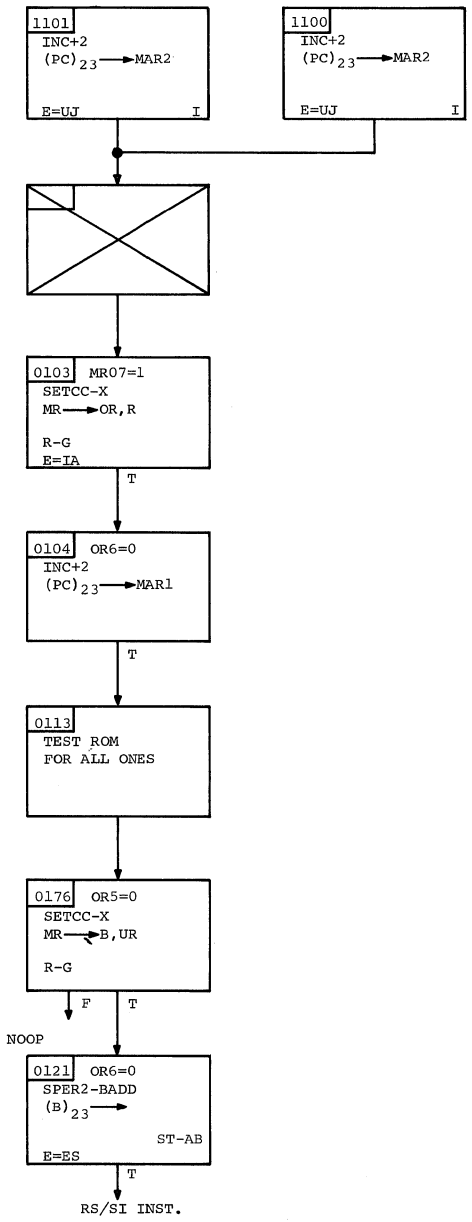
SUBTRACT UNNORMALIZED SHORT SU RX 7F



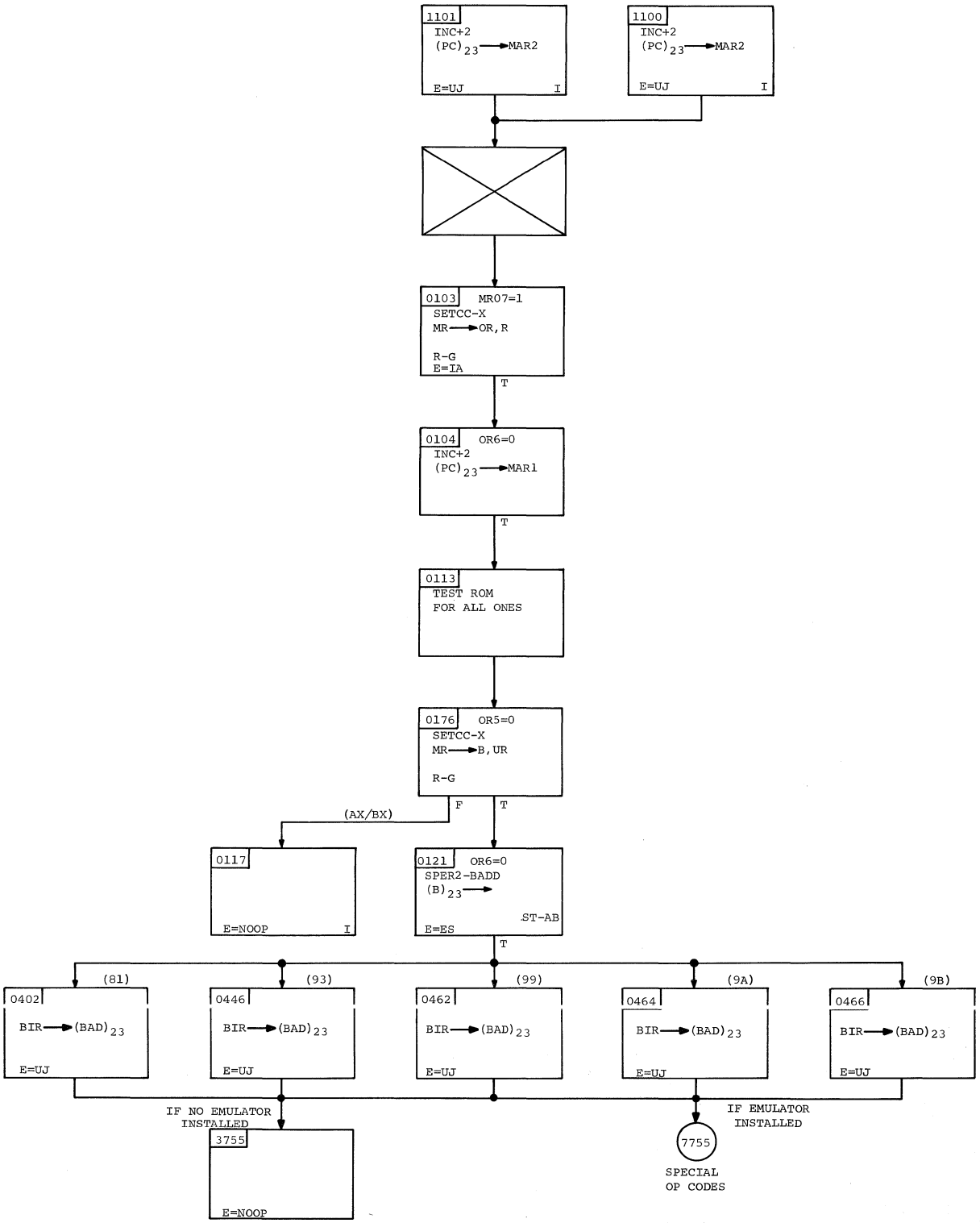
SUBTRACT UNNORMALIZED SHORT SU RX 7F

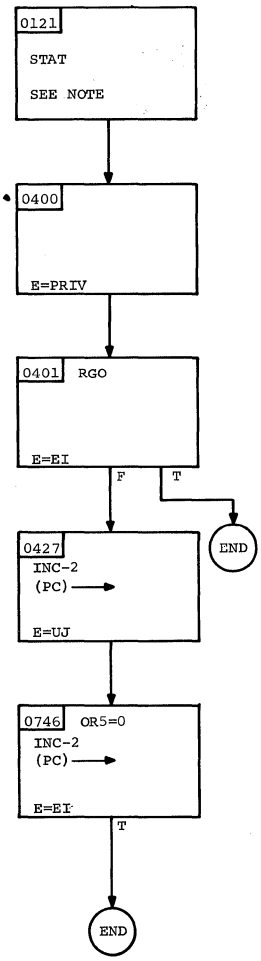


STATICIZING RS/SI FORMAT



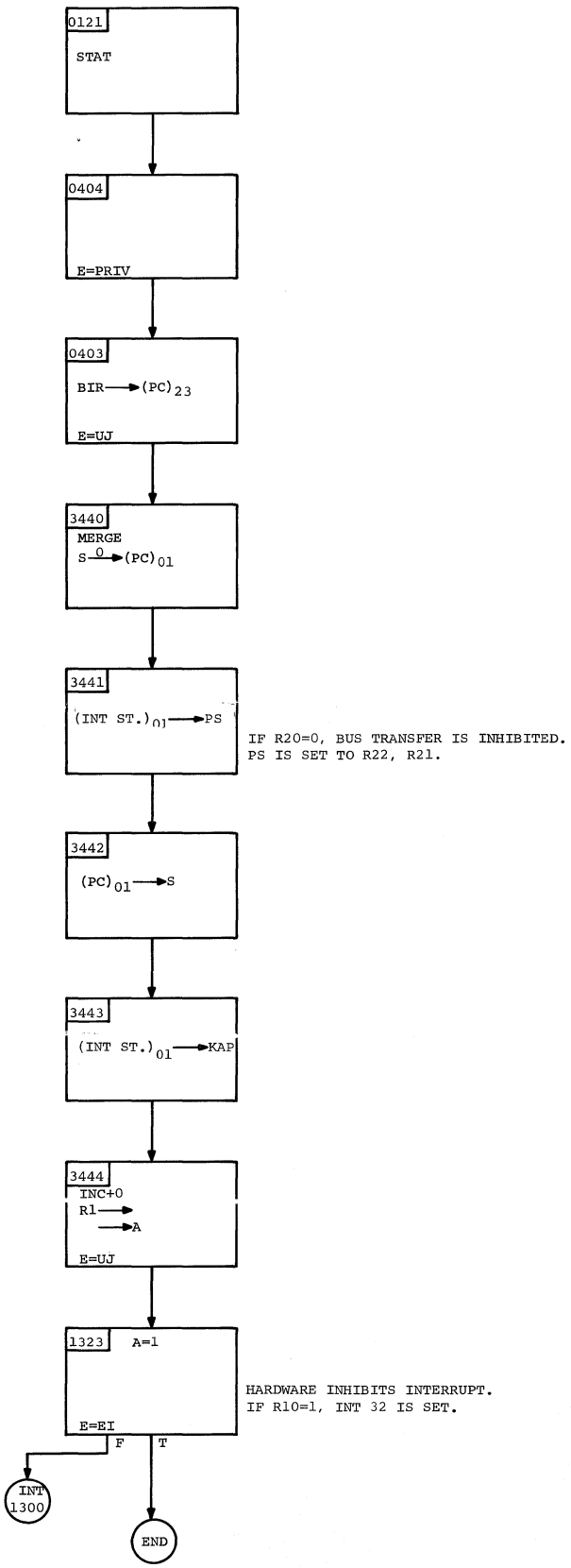
NO OPERATIONS INSTRUCTIONS SI FORMAT

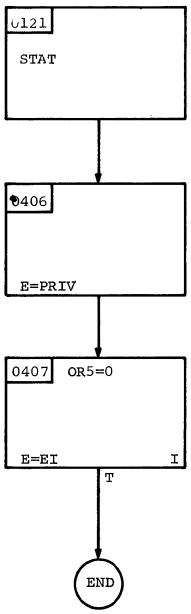




NOTE: DURING STATICIZING, HARDWARE WILL SET OR RESET ALI OR ISIM AS PER R REGISTER, IF IN PRIVILEGED MODE.

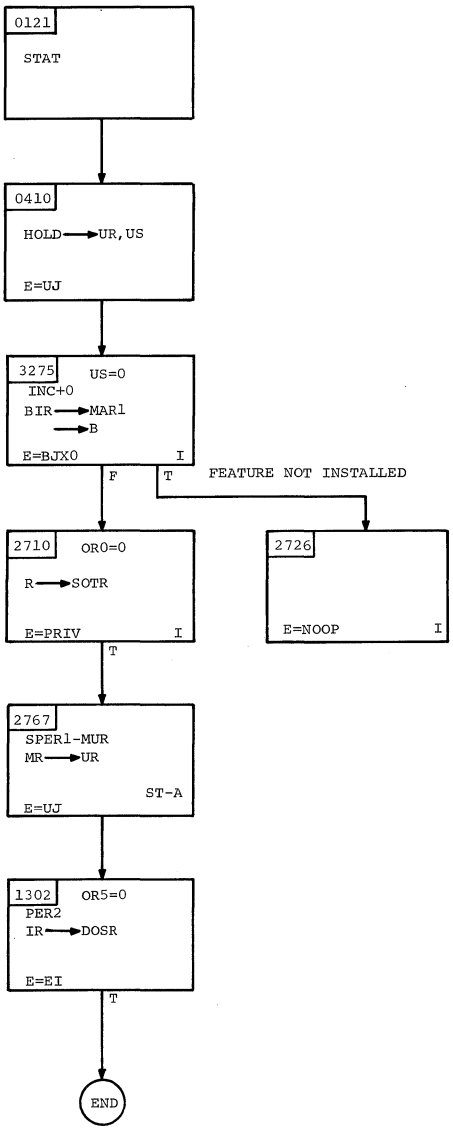
PROGRAM CONTROL PC SI 82

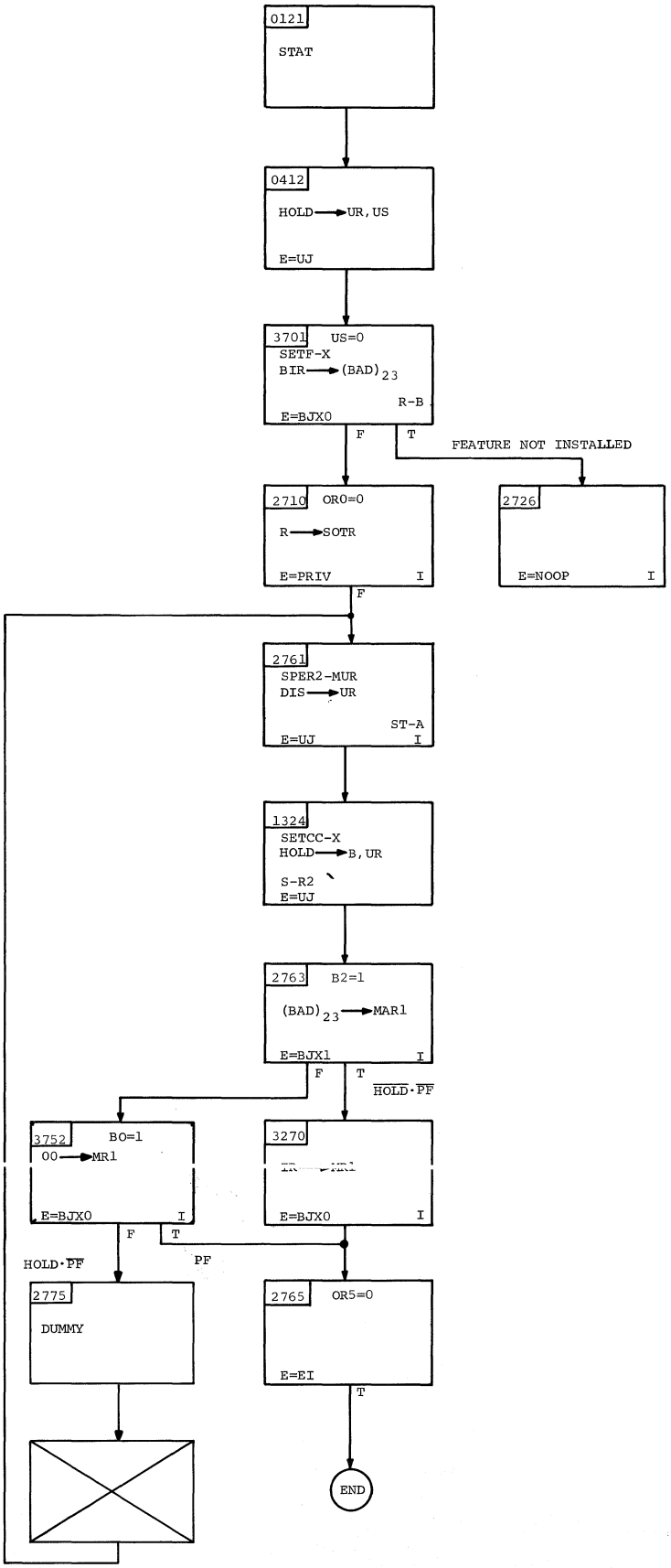




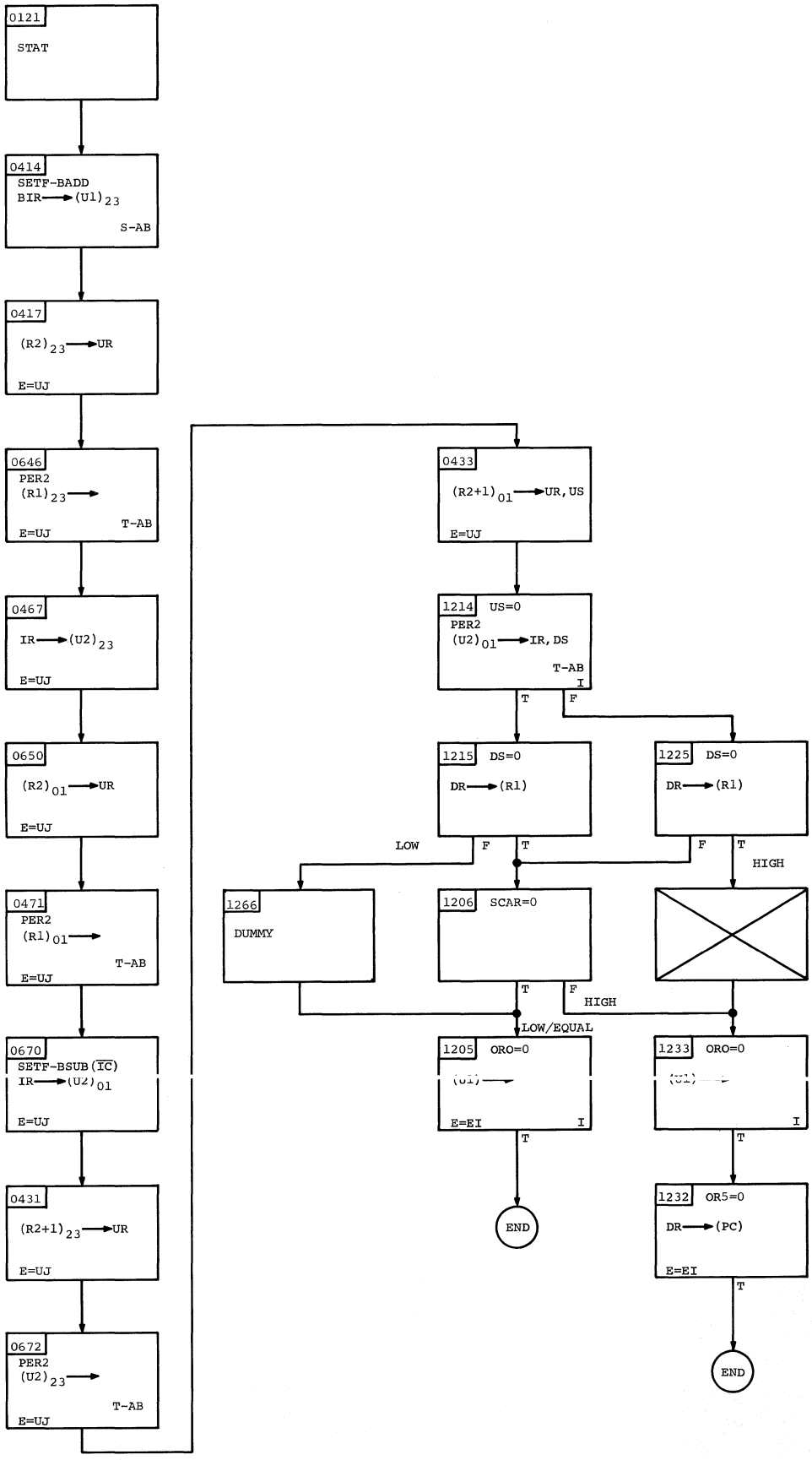
COUNTER IS SET TO R AND COUNTING BEGINS.
HARDWARE INHIBITS INTERRUPT.

WRITE DIRECT WRD SI 84

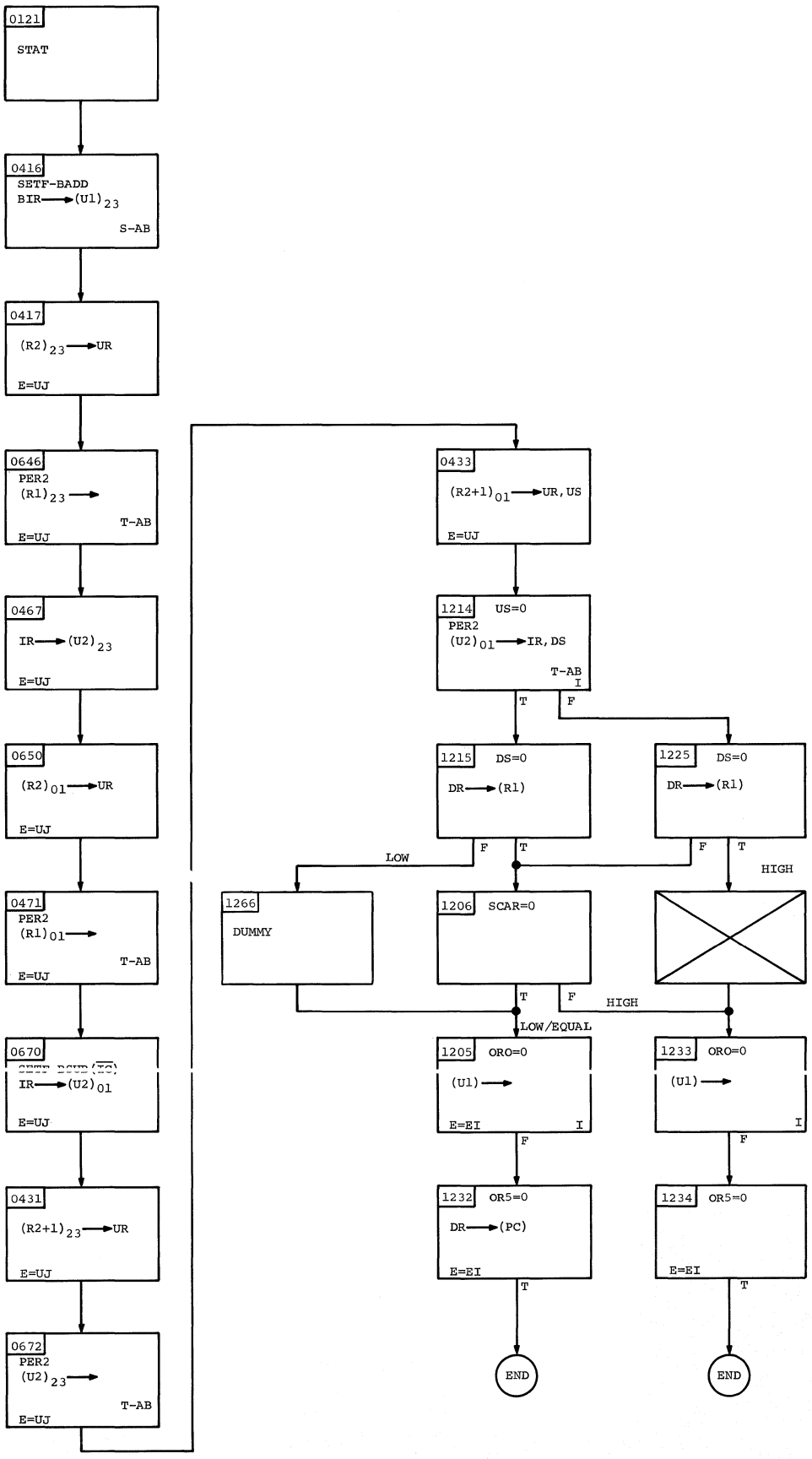



0121
STAT
0412
HOLD → UR, US
E=UJ
3701 US=0
SETF-X
BIR → (BAD)23
R-B
E=BJX0
F
T
FEATURE NOT INSTALLED
2710 OR0=0
R → SOTR
E=PRIV I
2726
E=NOOP I
F
2761
SPER2-MUR
DIS → UR
ST-A
E=UJ I
1324
SETCC-X
HOLD → B, UR
S-R2
E=UJ
2763 B2=1
(BAD)23 → MAR1
E=BJX1 I
F
T
HOLD·PF
3752 BO=1
00 → MR1
E=BJX0 I
3270
E=BJX0 I
F
T
HOLD·PF
PF
2775
DUMMY
2765 OR5=0
E=EI
T
END

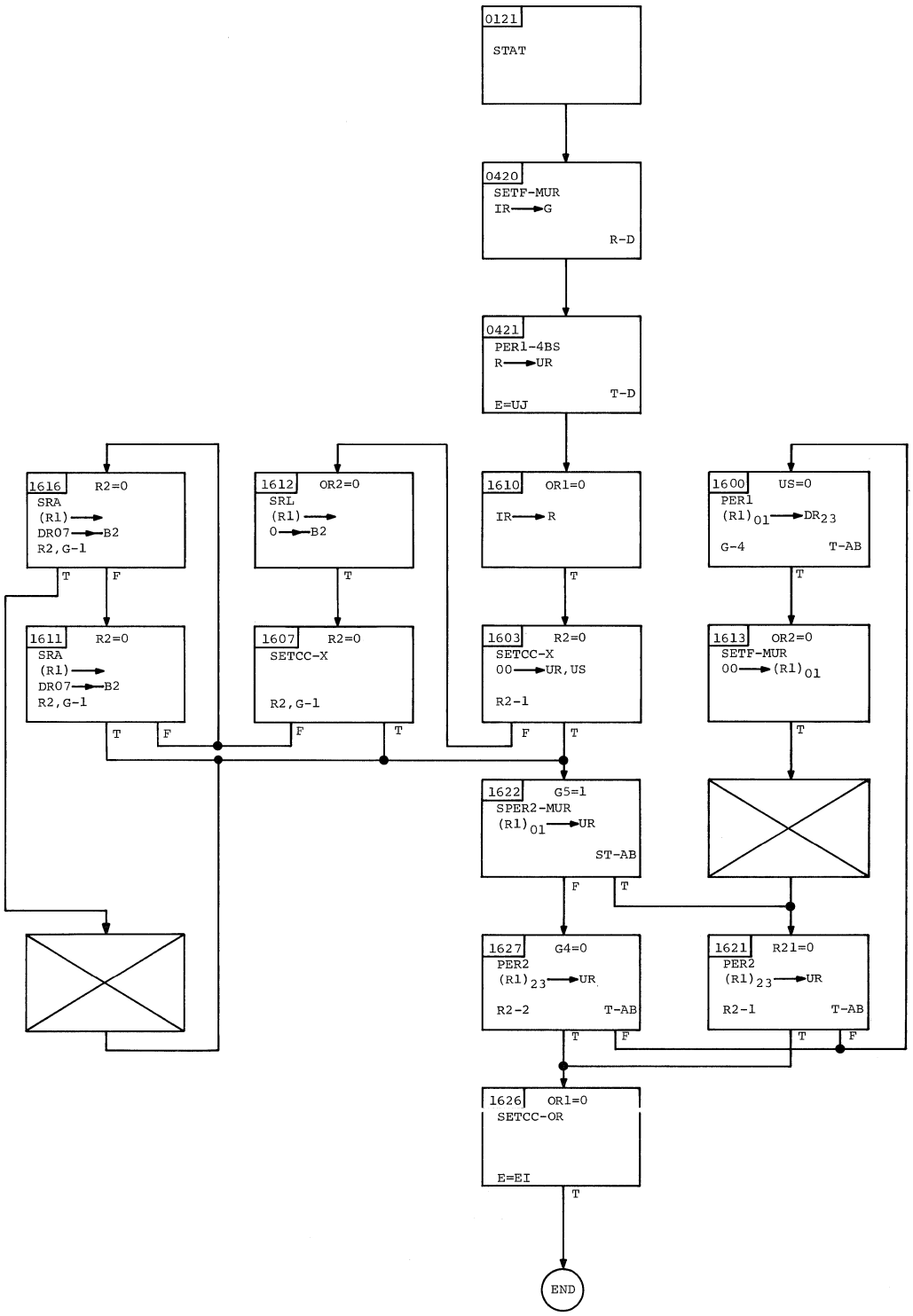
BRANCH ON INDEX HIGH BXH RS 86



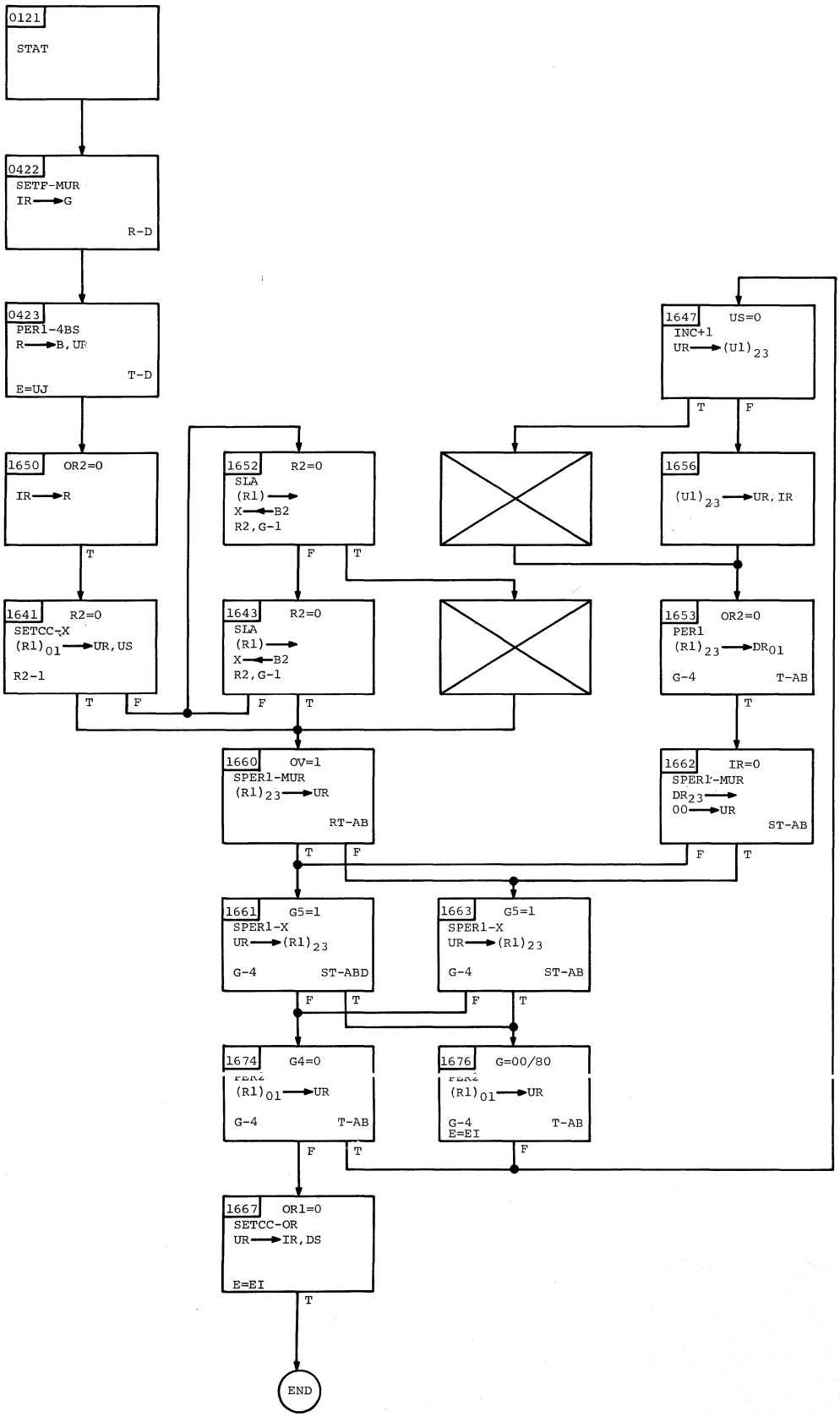
BRANCH ON INDEX LOW OR EQUAL BXLE RS 87



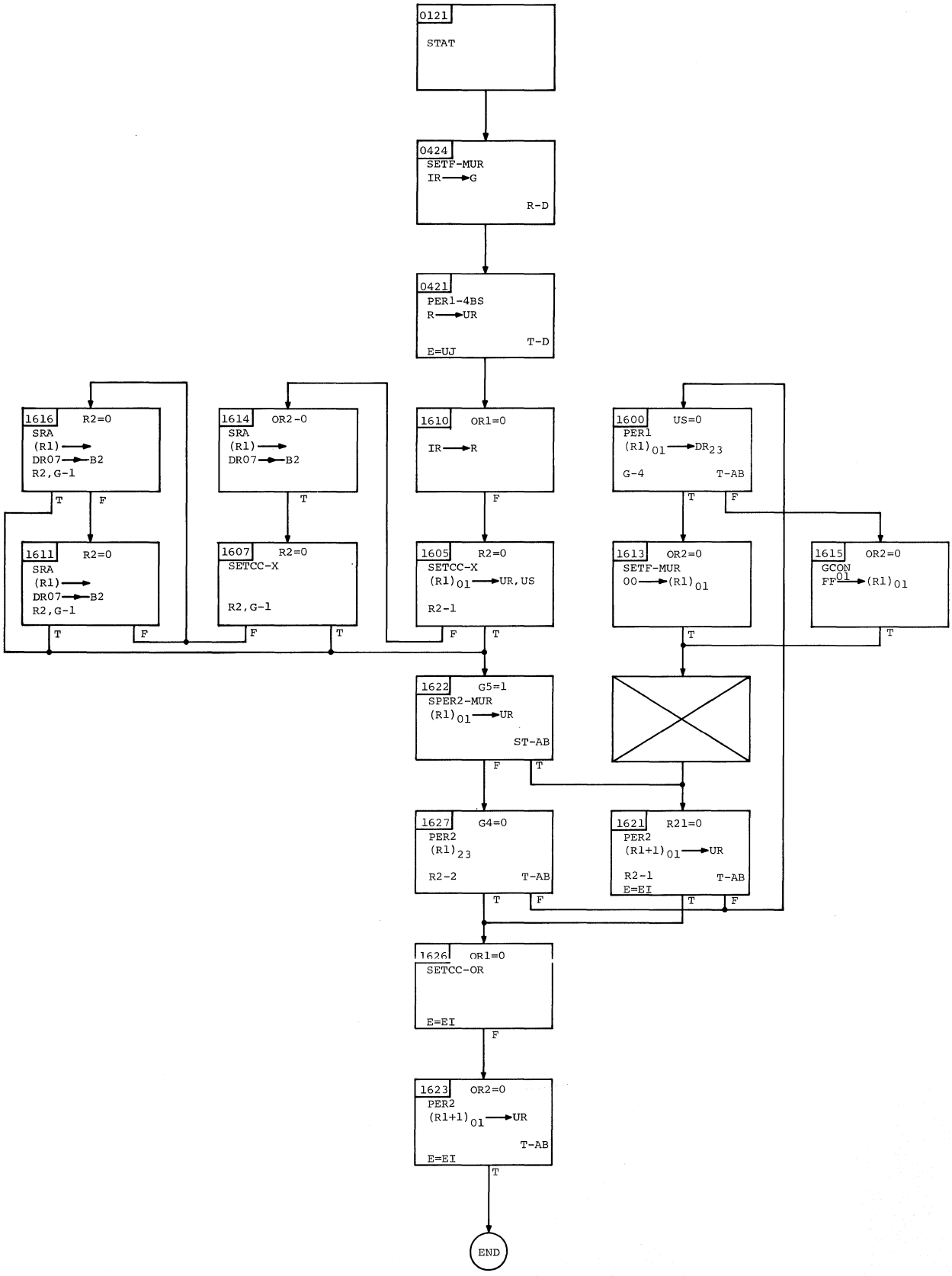
SHIFT RIGHT LOGICAL SRL RS 88



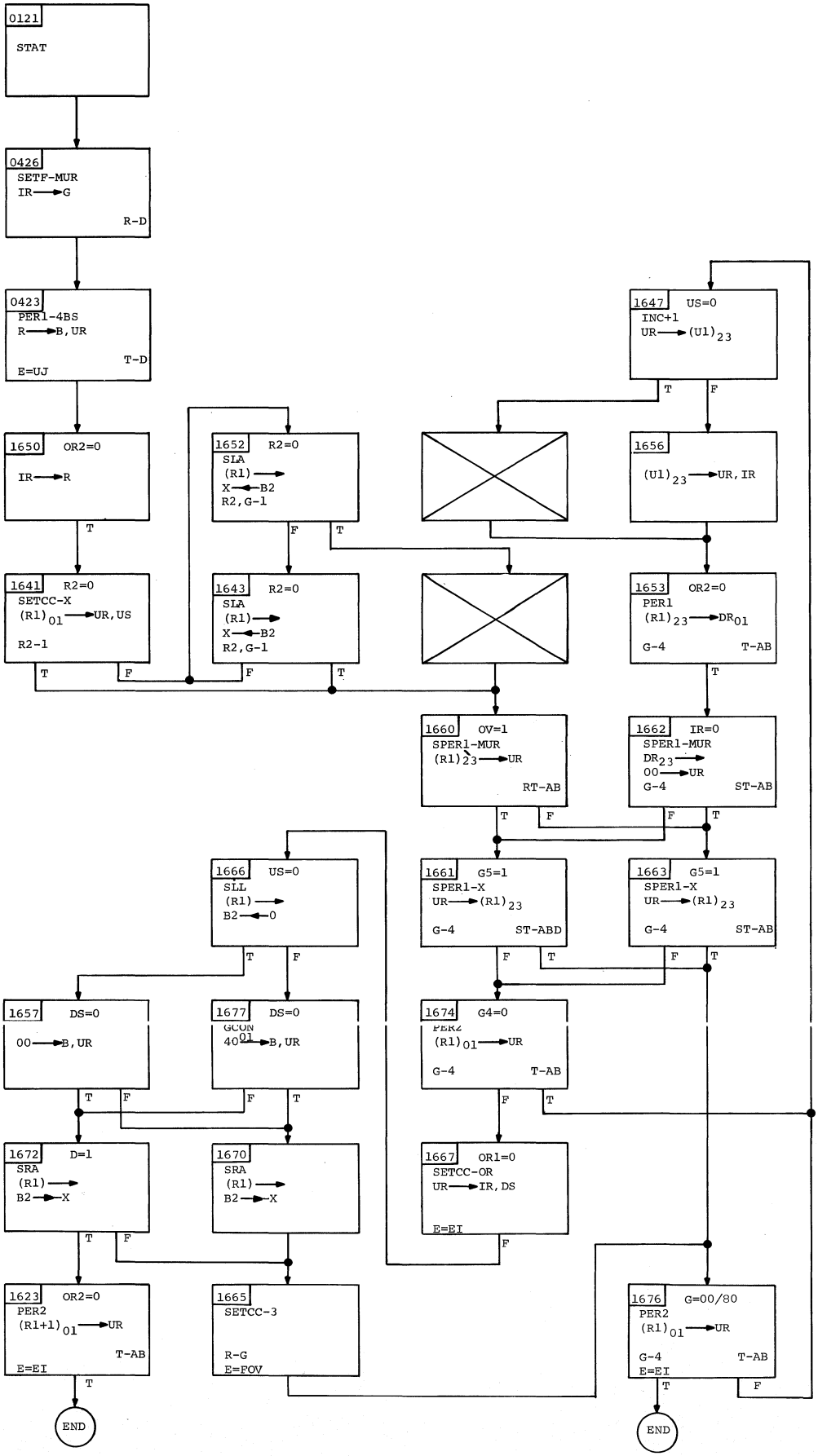
SHIFT LEFT LOGICAL SLL RS 89



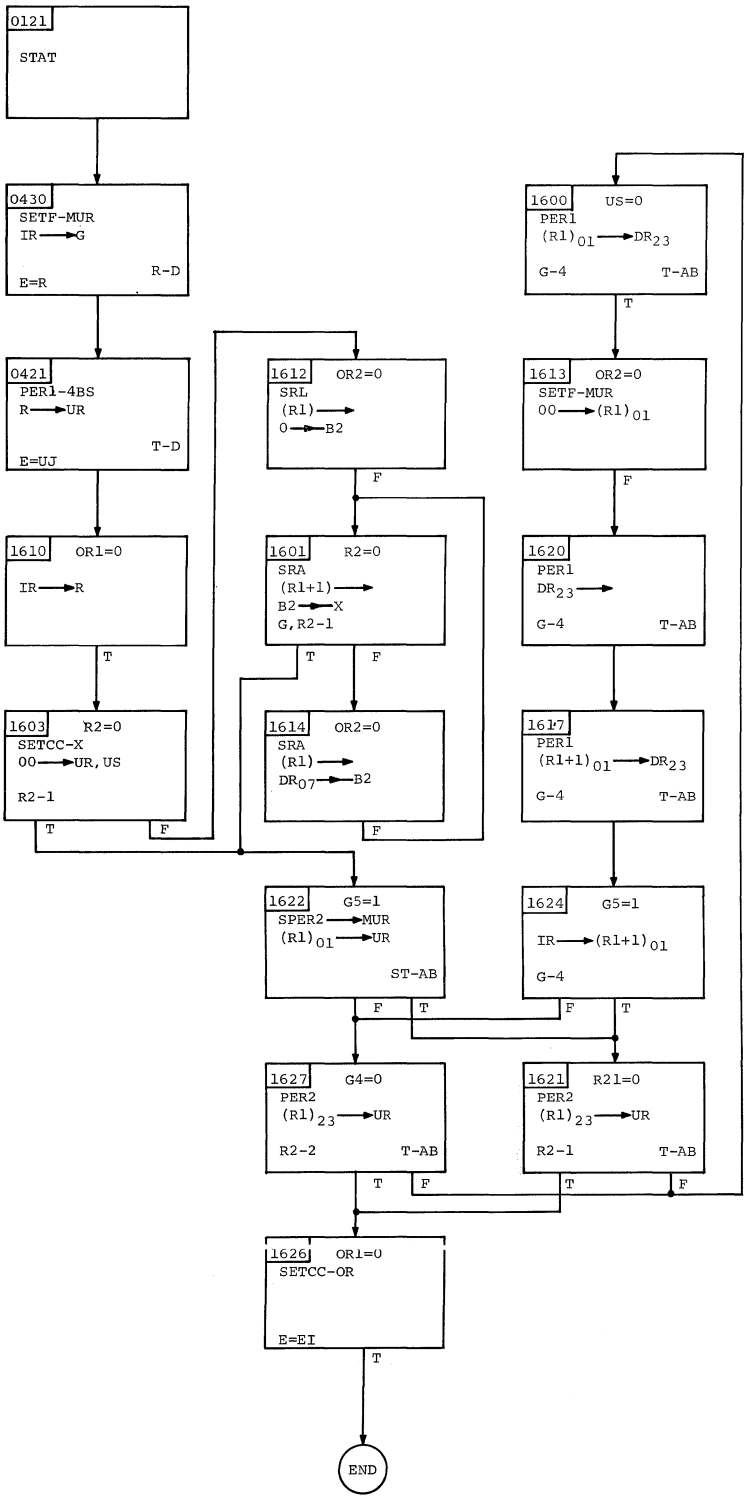
SHIFT RIGHT ALGEBRAIC SRA RS 8A



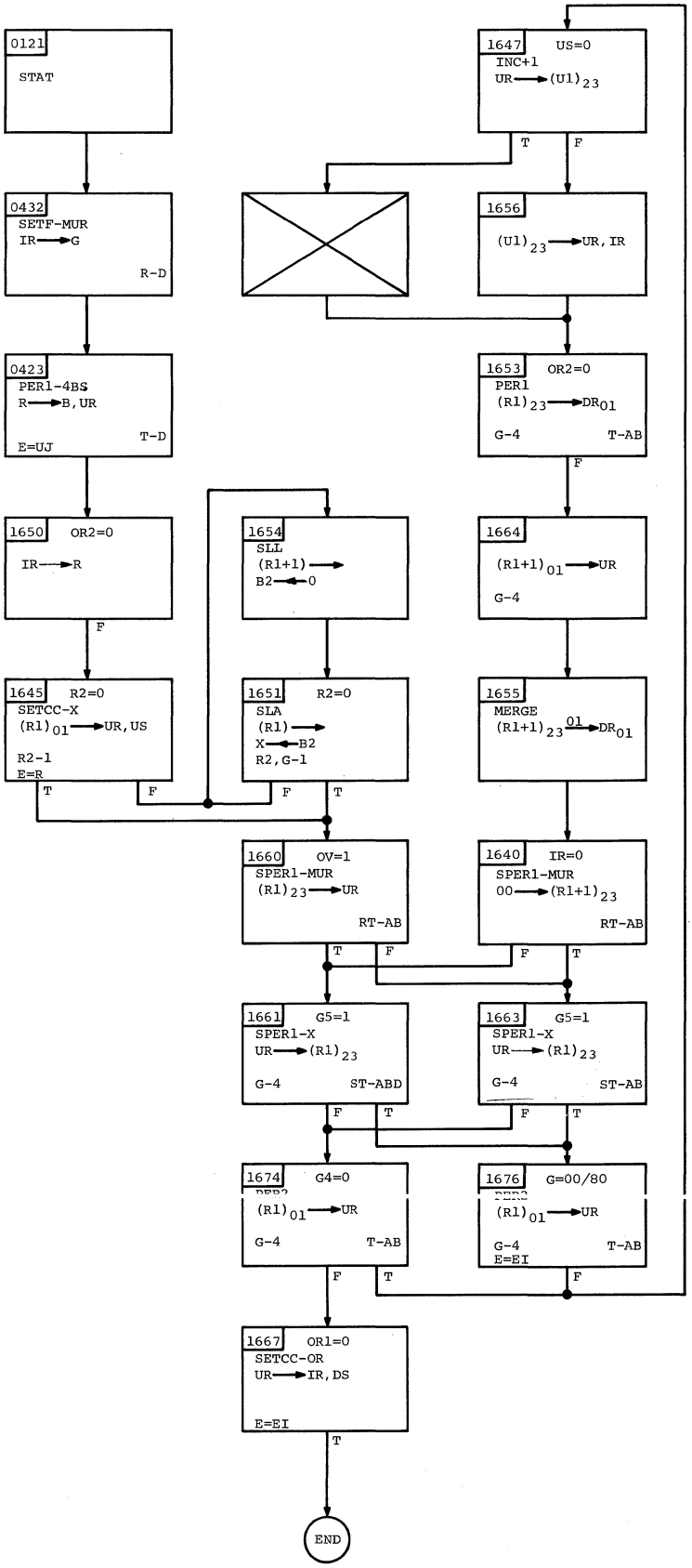
SHIFT LEFT ALGEBRAIC SLA RS 8B



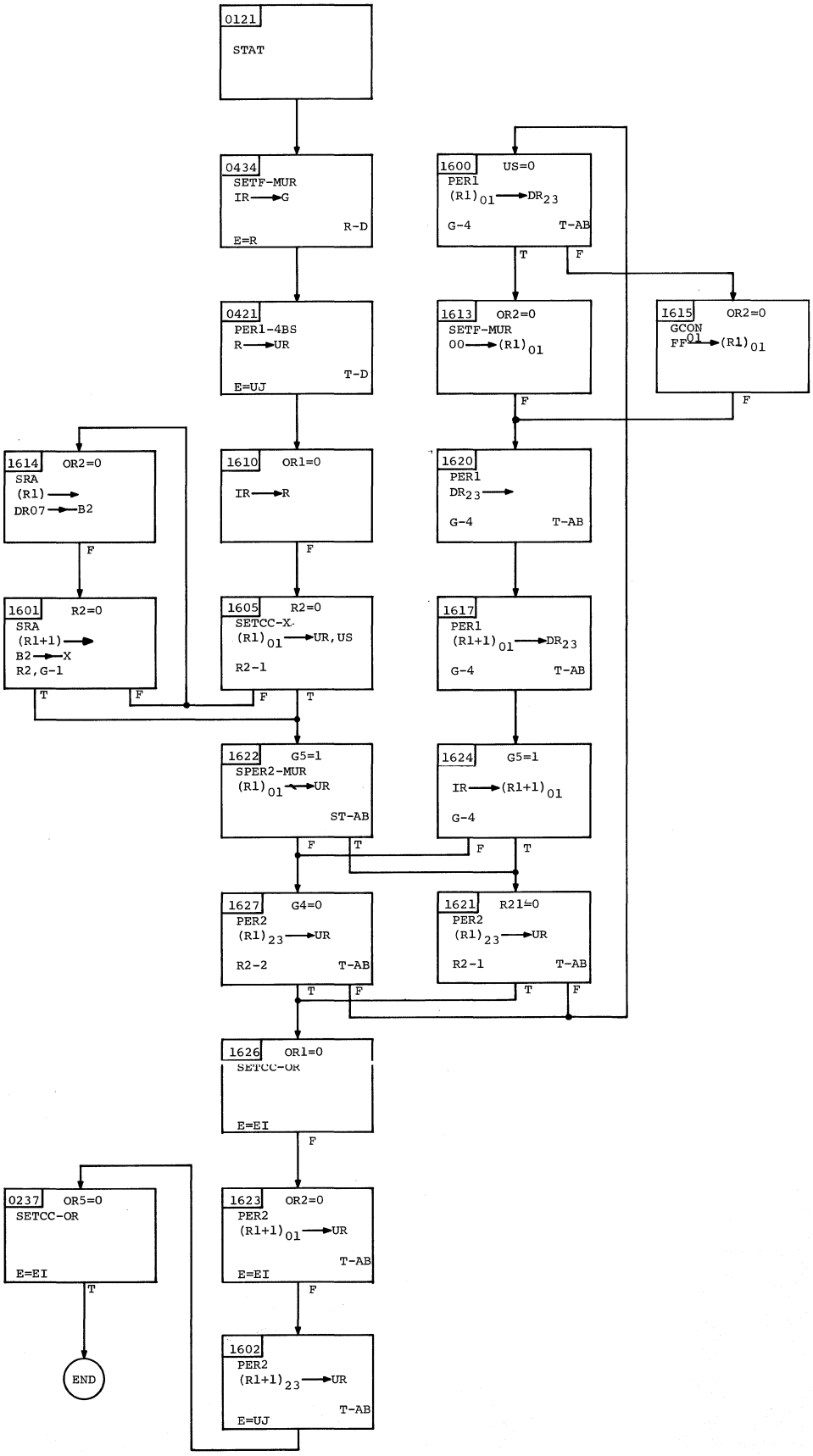
SHIFT RIGHT DOUBLE LOGICAL SRDL RS 8C



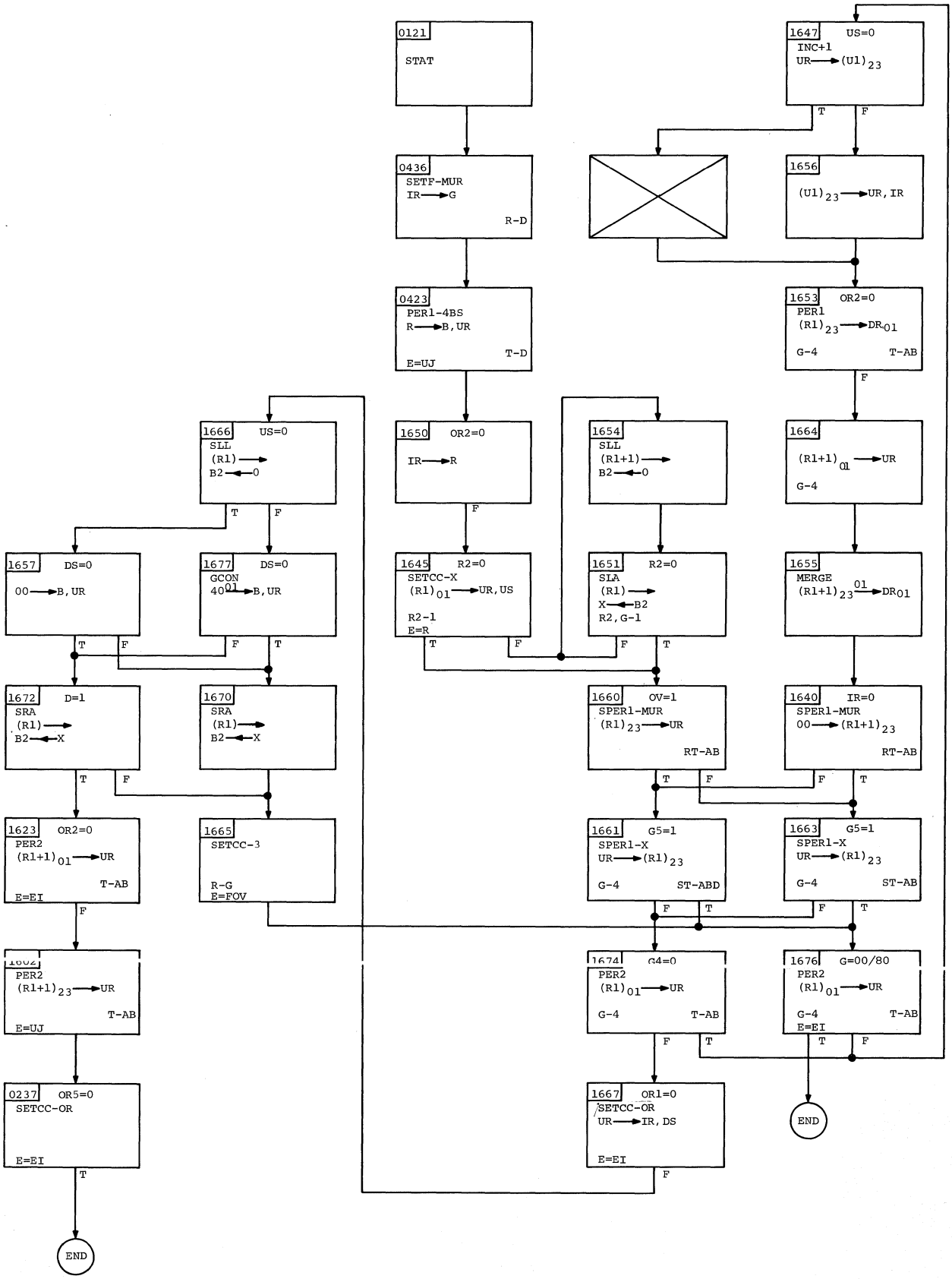
SHIFT LEFT DOUBLE LOGICAL SLDL RS 8D



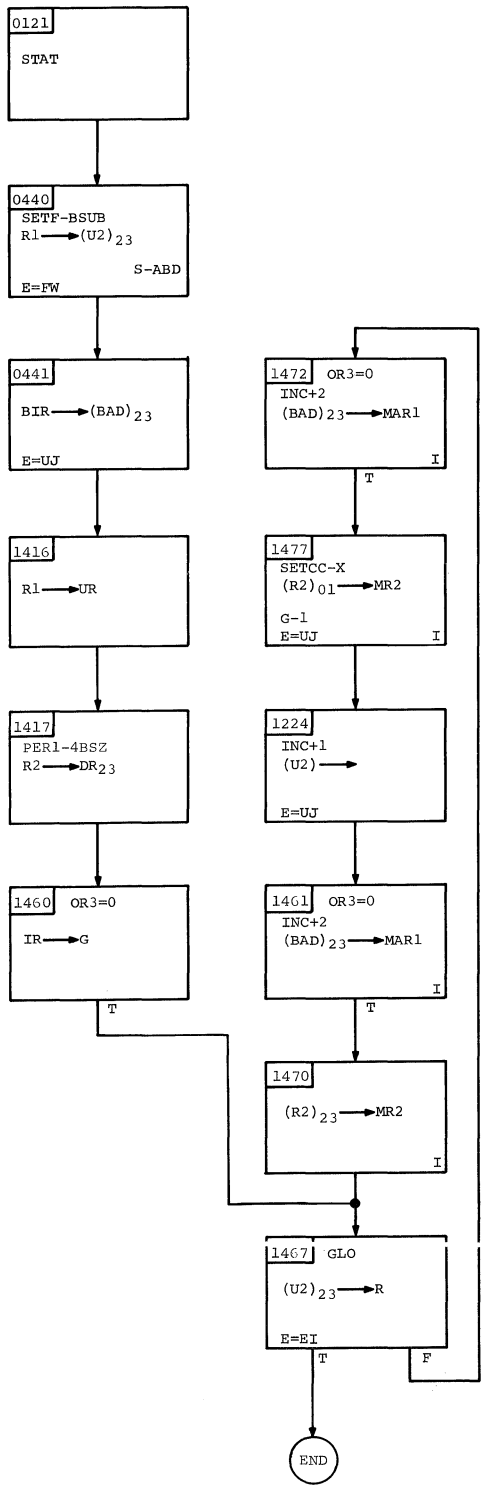
SHIFT RIGHT DOUBLE ALGEBRAIC SRDA RS 8E



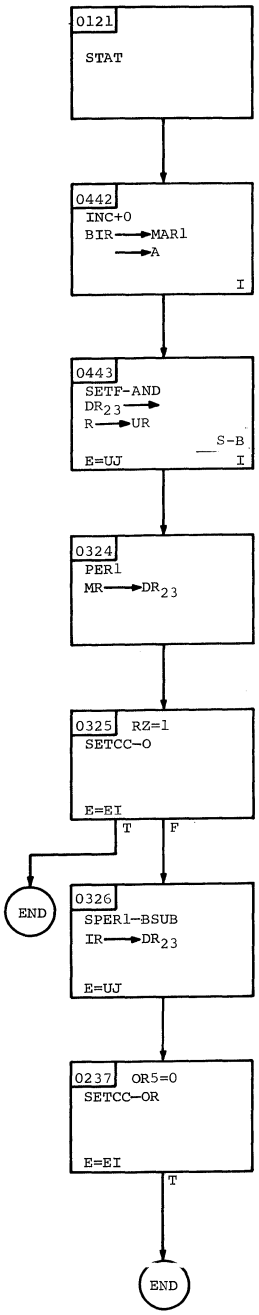
SHIFT LEFT DOUBLE ALGEBRAIC SLDA RS 8F



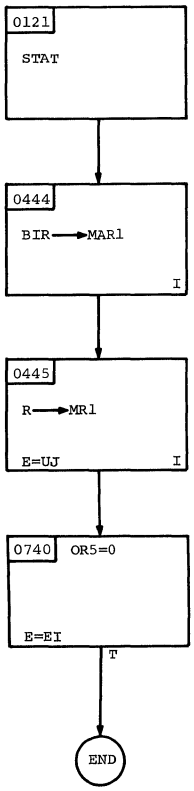
STORE MULTIPLE STM RS 90



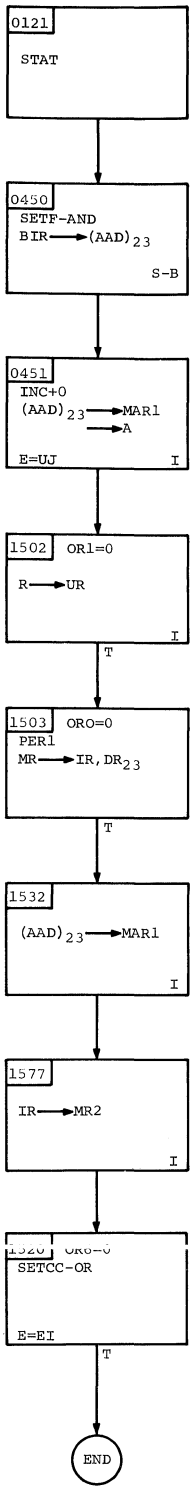
TEST UNDER MASK TM SI 91



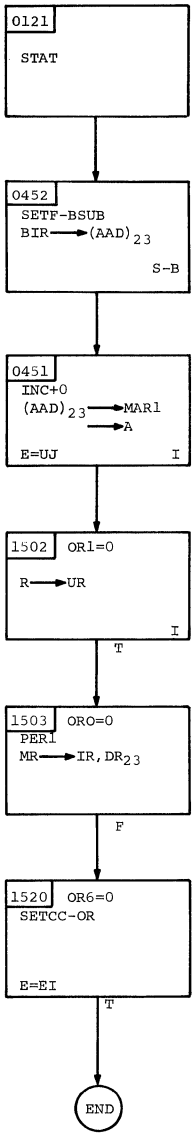
MOVE IMMEDIATE MVI SI 92



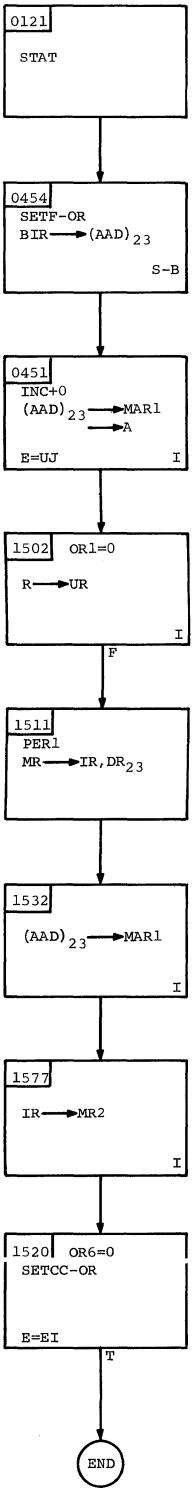
LOGICAL AND NI SI 94



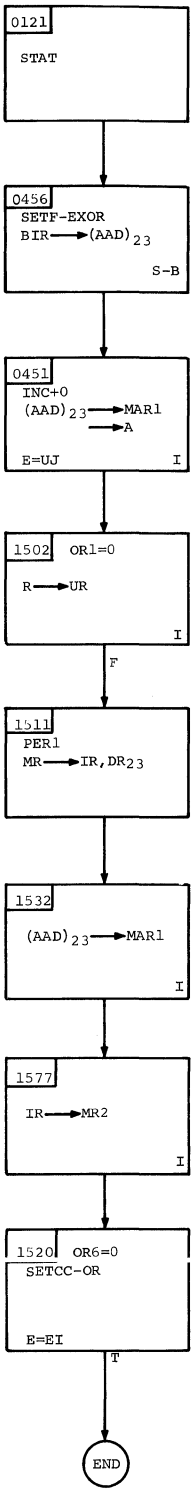
COMPARE LOGICAL CLI SI 95



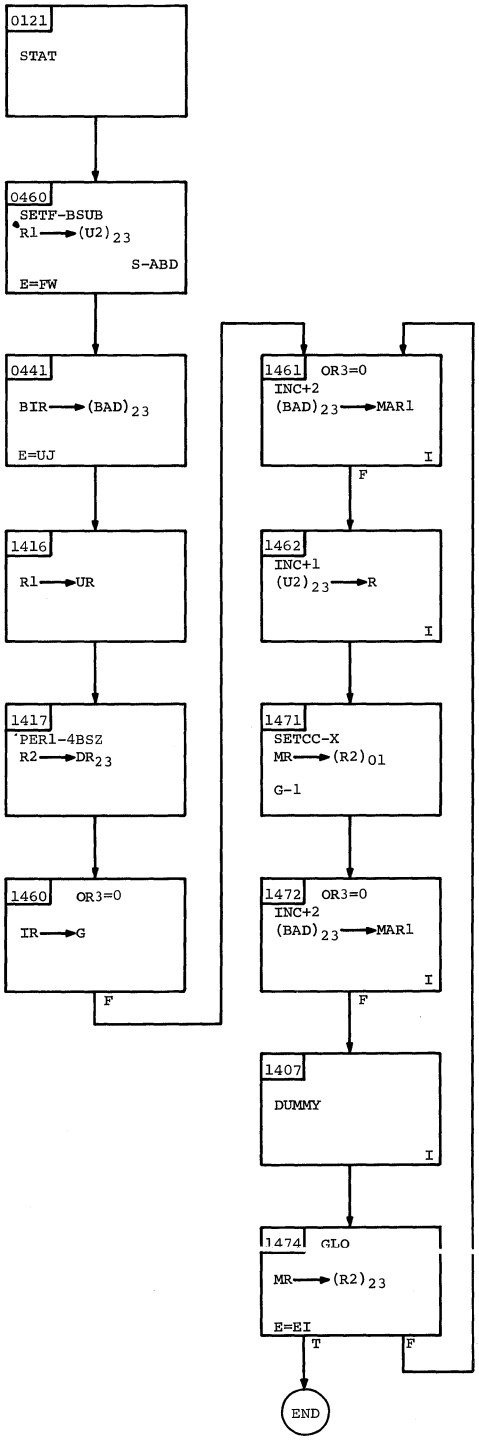
LOGICAL OR 01 SI 96



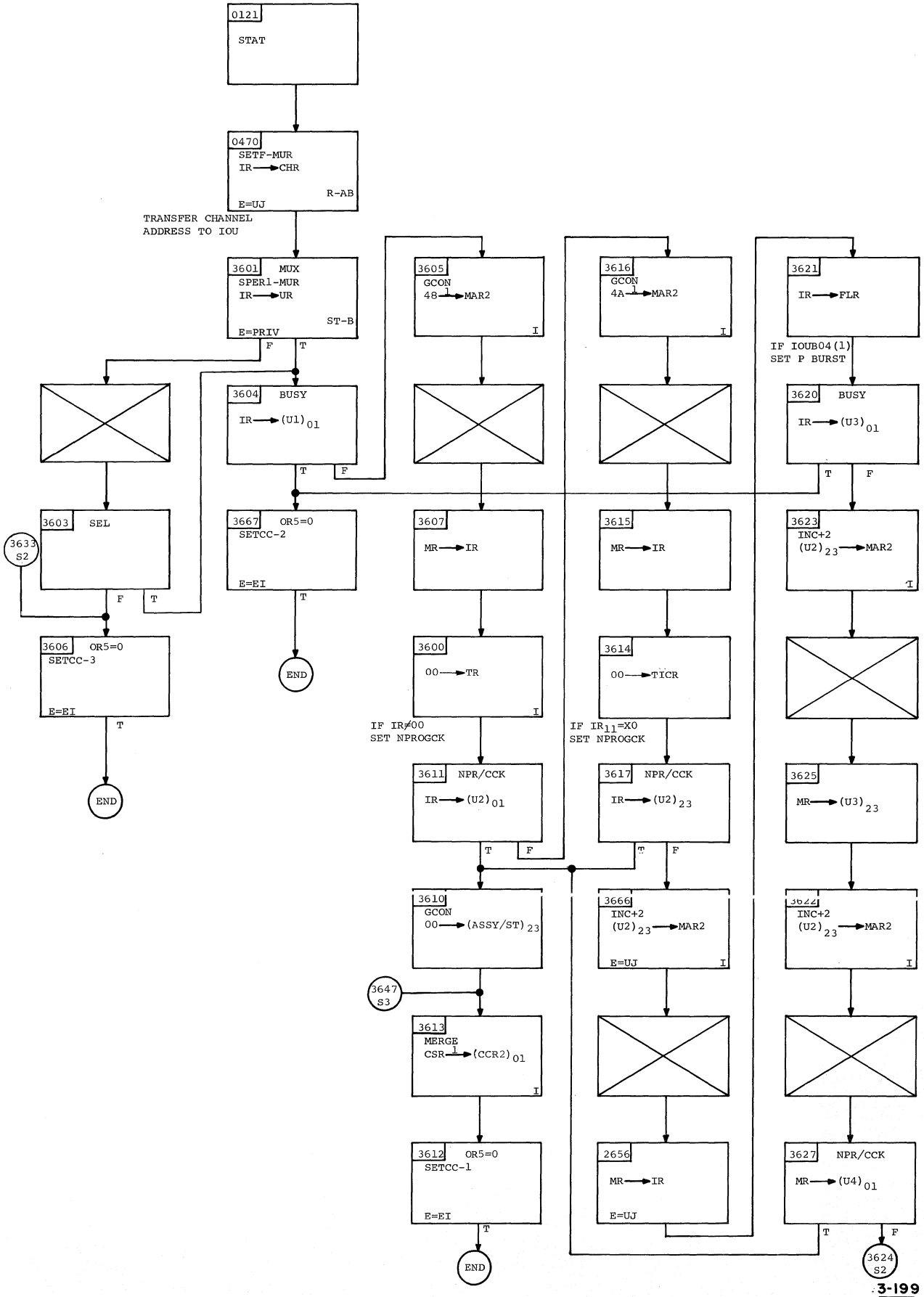
LOGICAL EXCLUSIVE OR XI ST 97



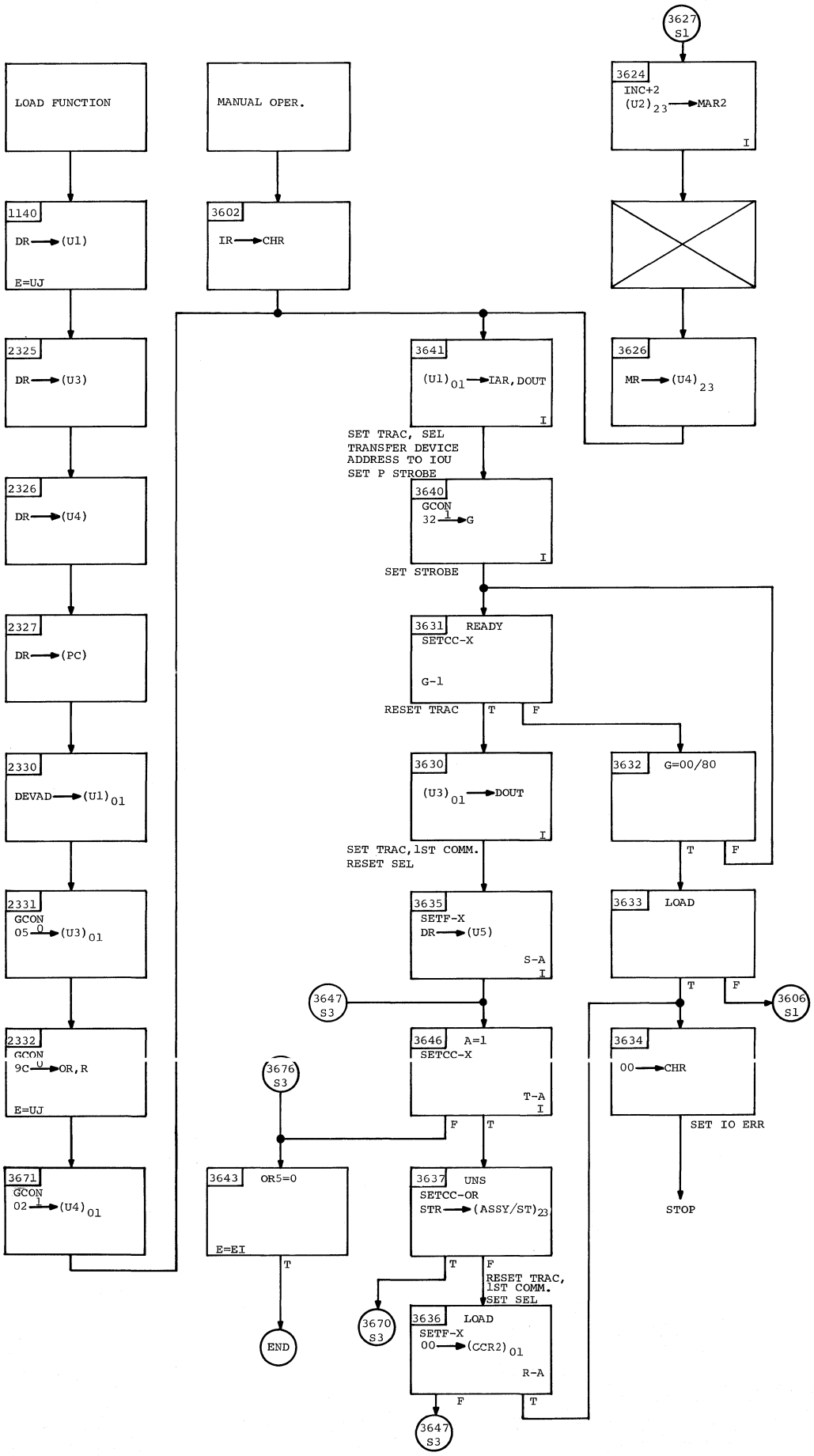
LOAD MULTIPLE LM RS 98

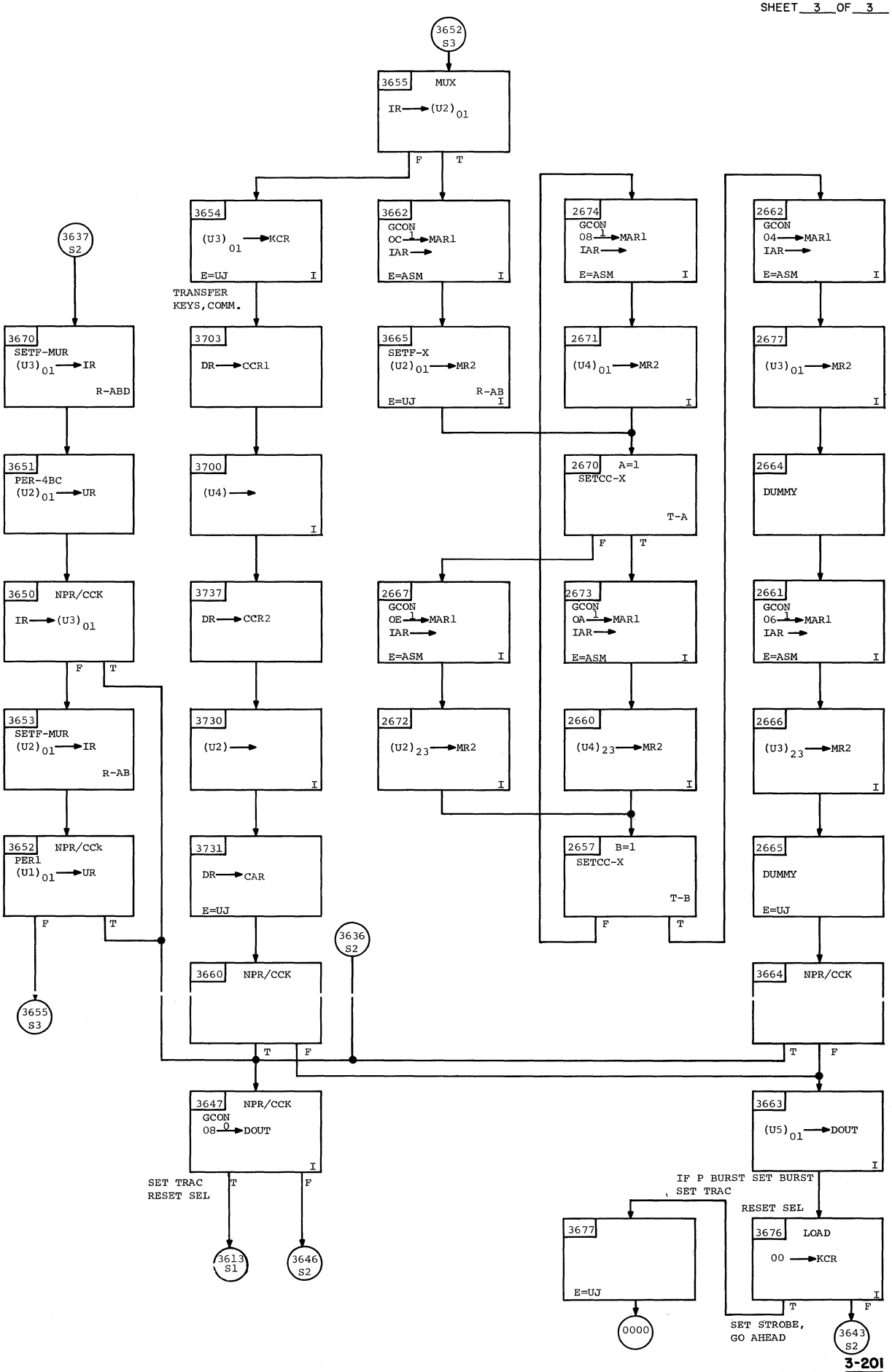


START DEVICE SDV SI 9C

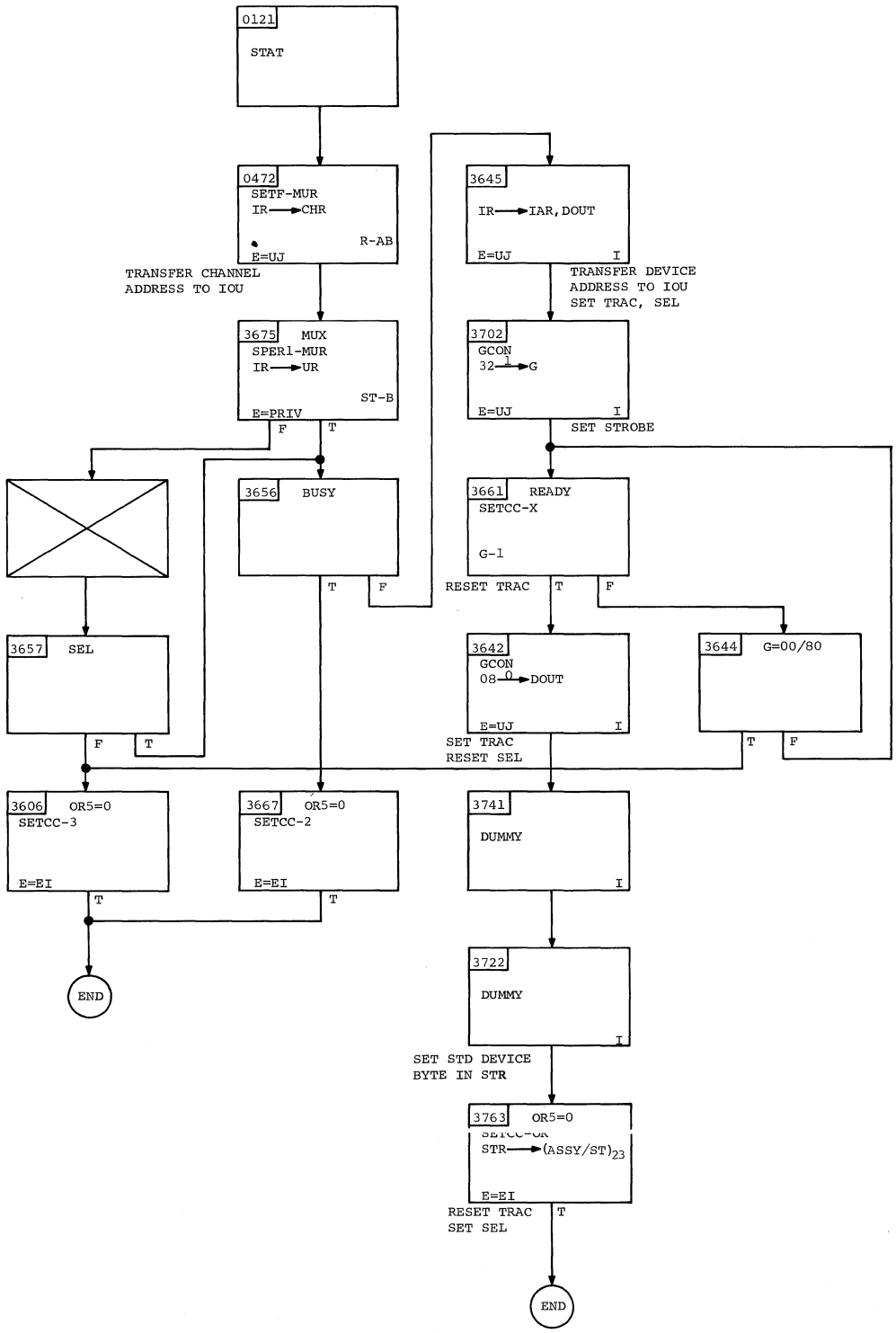


START DEVICE SDV SI 9C

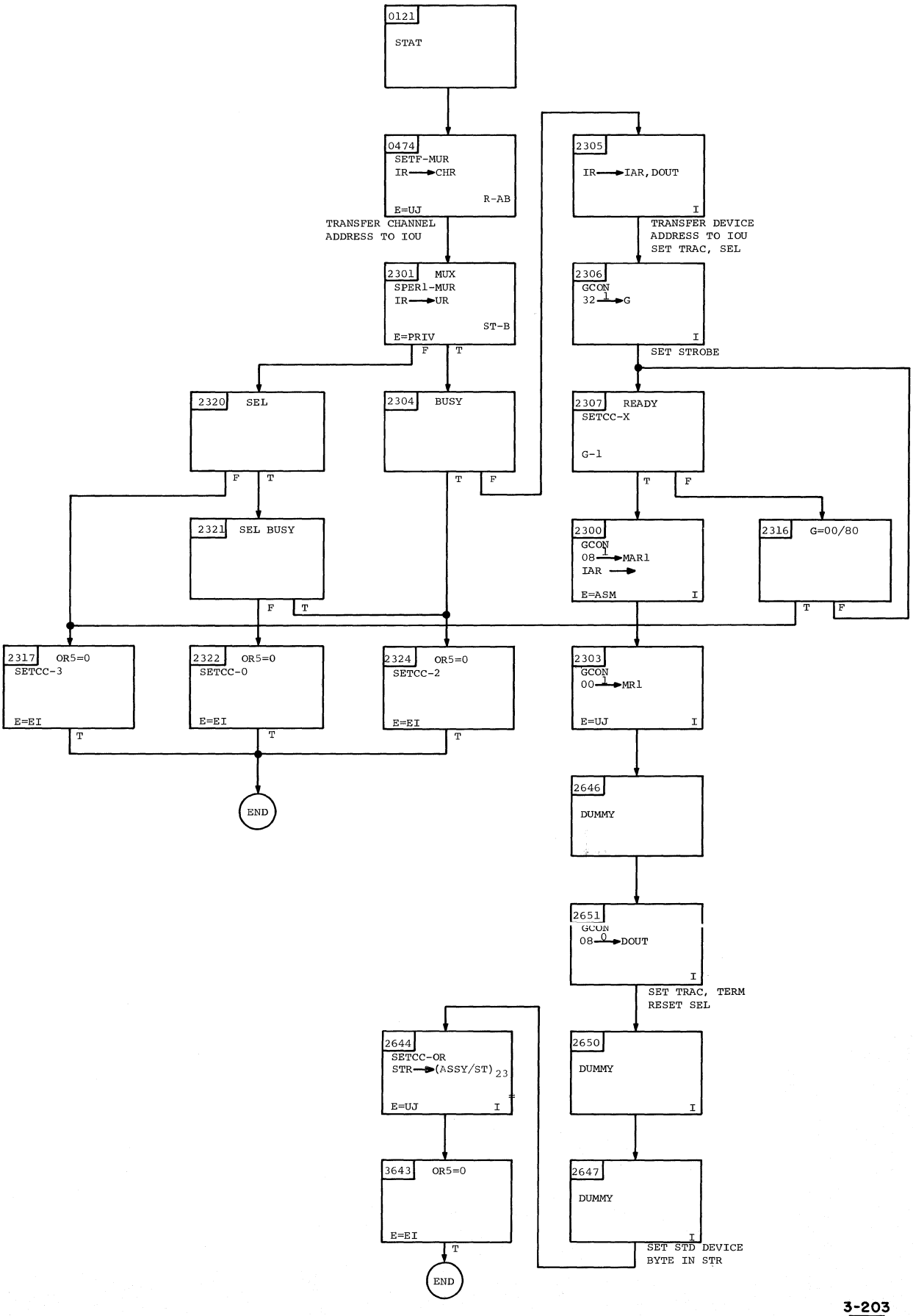



3652 S3
3655 MUX
IR → (U2) 01
F
T
3654
(U3) 01 → KCR
E=UJ
I
TRANSFER KEYS, COMM.
3662
GCON
OC 1 → MAR1
IAR →
E=ASM
I
2674
GCON
08 1 → MAR1
IAR →
E=ASM
I
2662
GCON
04 → MAR1
IAR →
E=ASM
I
3637 S2
3670
SETF-MUR
(U3) 01 → IR
R-ABD
3703
DR → CCR1
3665
SETF-X
(U2) 01 → MR2
R-AB
E=UJ
I
2671
(U4) 01 → MR2
I
2677
(U3) 01 → MR2
I
3651
PER-4BC
(U2) 01 → UR
3700
(U4) →
I
2670 A=1
SETCC-X
T-A
2664
DUMMY
3650 NPR/CCK
IR → (U3) 01
F
T
3737
DR → CCR2
2667
GCON
OE 1 → MAR1
IAR →
E=ASM
I
2673
GCON
OA 1 → MAR1
IAR →
E=ASM
I
2661
GCON
06 1 → MAR1
IAR →
E=ASM
I
3653
SETF-MUR
(U2) 01 → IR
R-AB
3730
(U2) →
I
2672
(U2) 23 → MR2
I
2660
(U4) 23 → MR2
I
2666
(U3) 23 → MR2
I
3652 NPR/CCk
PER1
(U1) 01 → UR
F
T
3731
DR → CAR
E=UJ
2657 B=1
SETCC-X
T-B
2665
DUMMY
E=UJ
3636 S2
3660 NPR/CCK
3664 NPR/CCK
3655 S3
T
F
3647 NPR/CCK
GCON
08 0 → DOUT
I
3663
(U5) 01 → DOUT
I
SET TRAC
RESET SEL
IF P BURST SET BURST
SET TRAC
RESET SEL
3677
E=UJ
3676 LOAD
00 → KCR
I
3613 S1
3646 S2
0000
SET STROBE, GO AHEAD
3643 S2

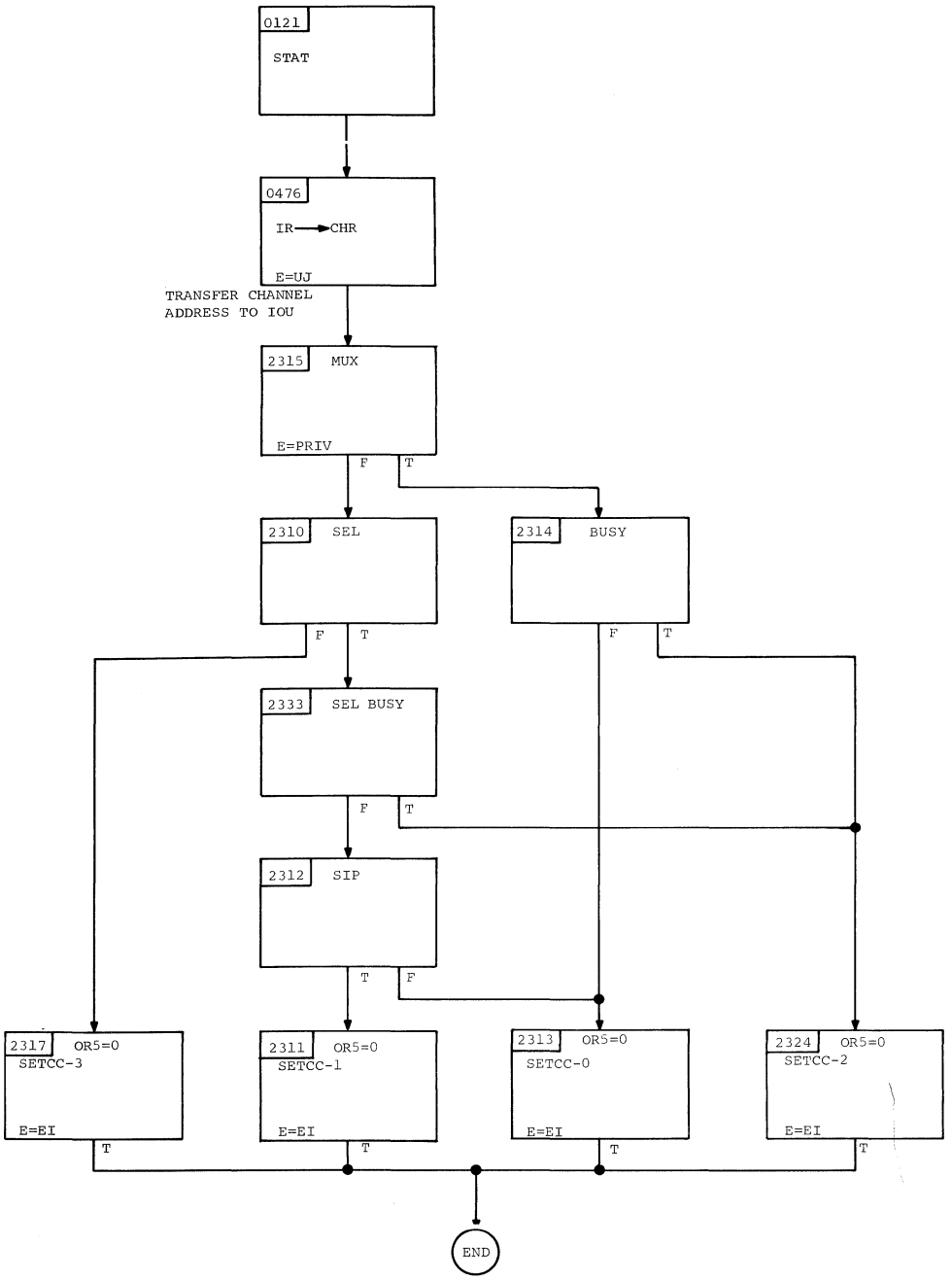
TEST DEVICE TDV SI 9D



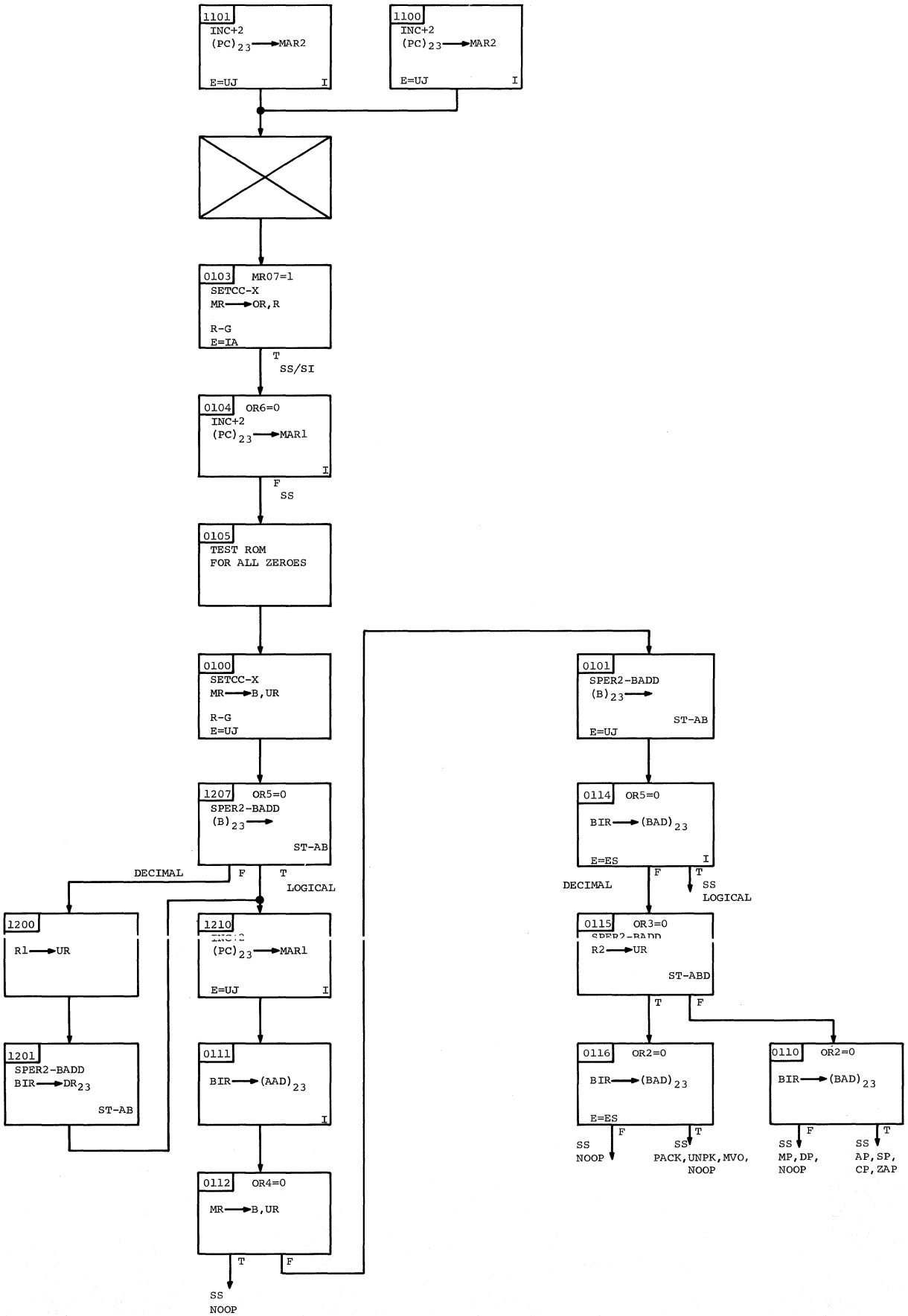
HALT DEVICE HDV SI 9E



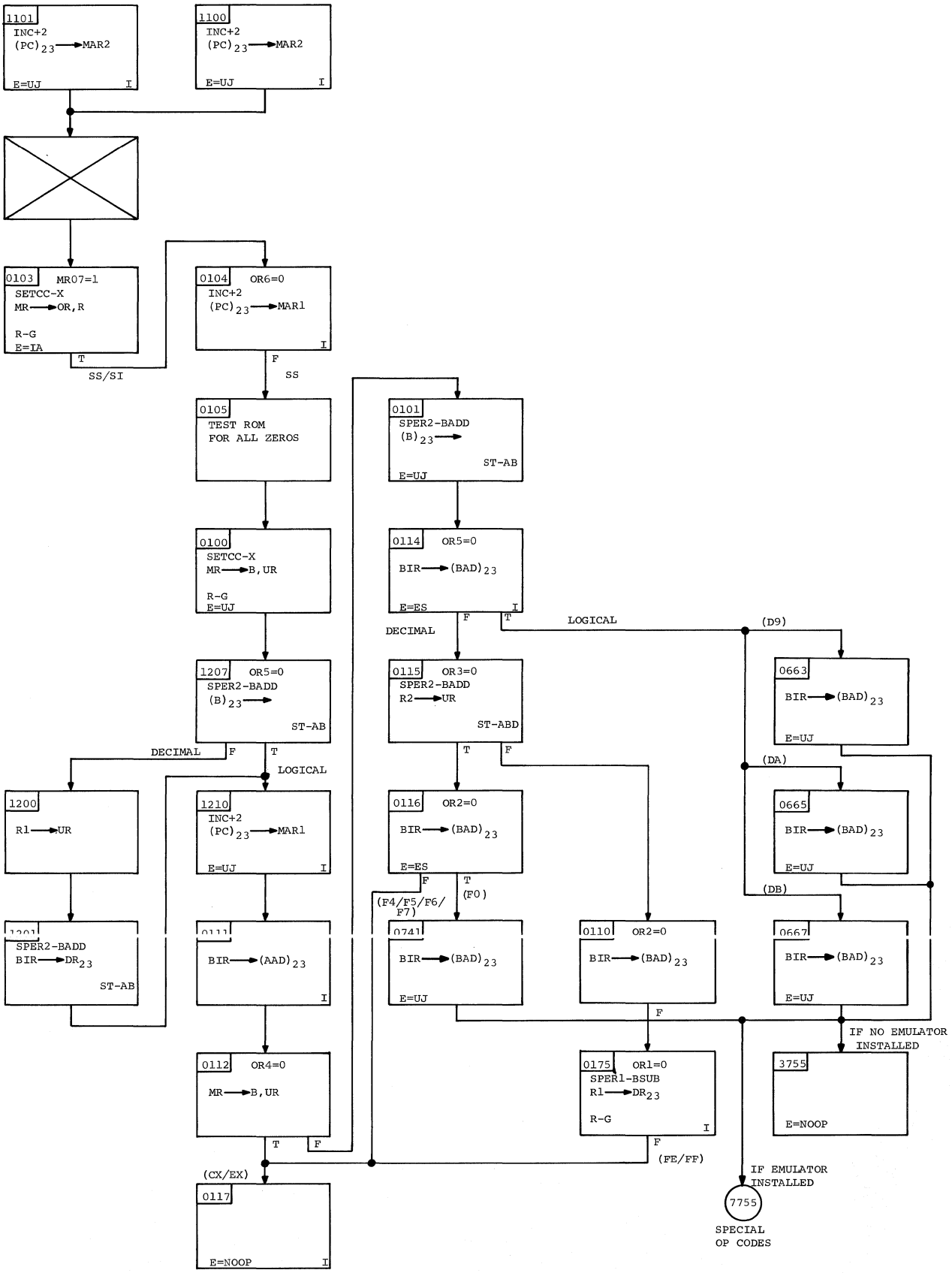
CHECK CHANNEL CKC SI 9F



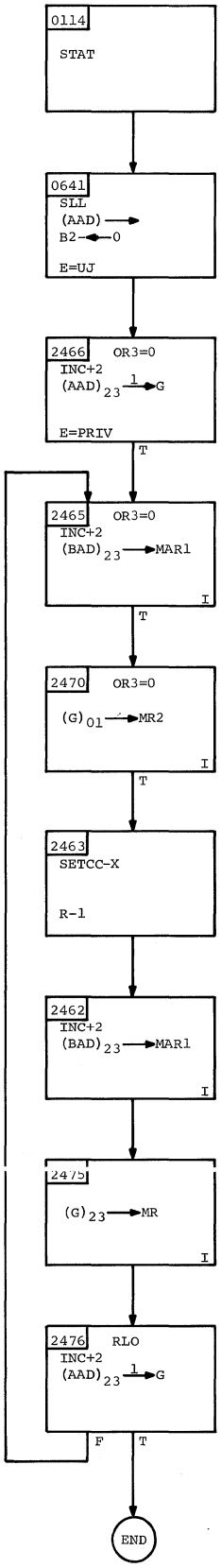
STATICIZING SS FORMAT



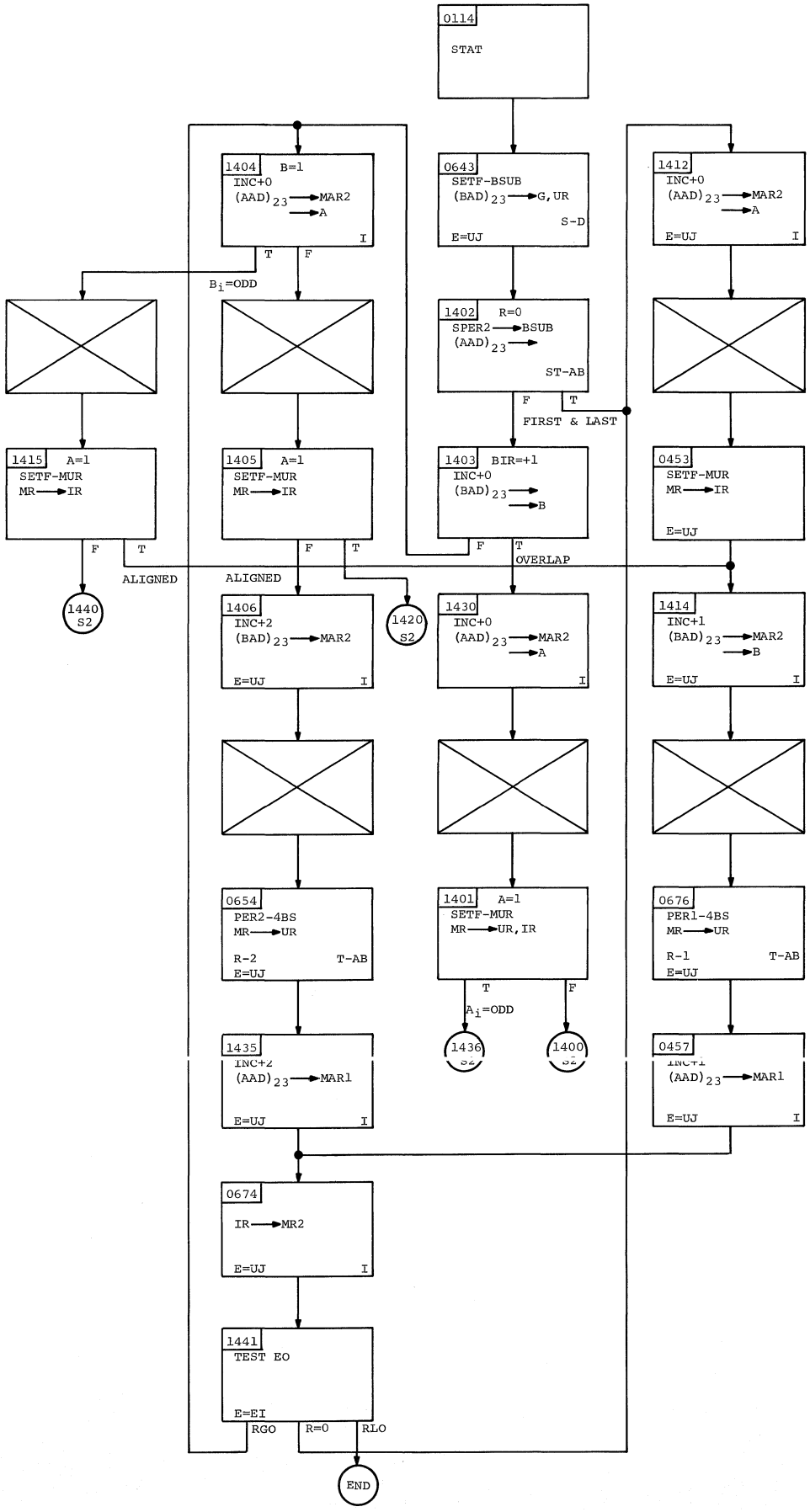
NO OPERATION INSTRUCTIONS SS FORMAT



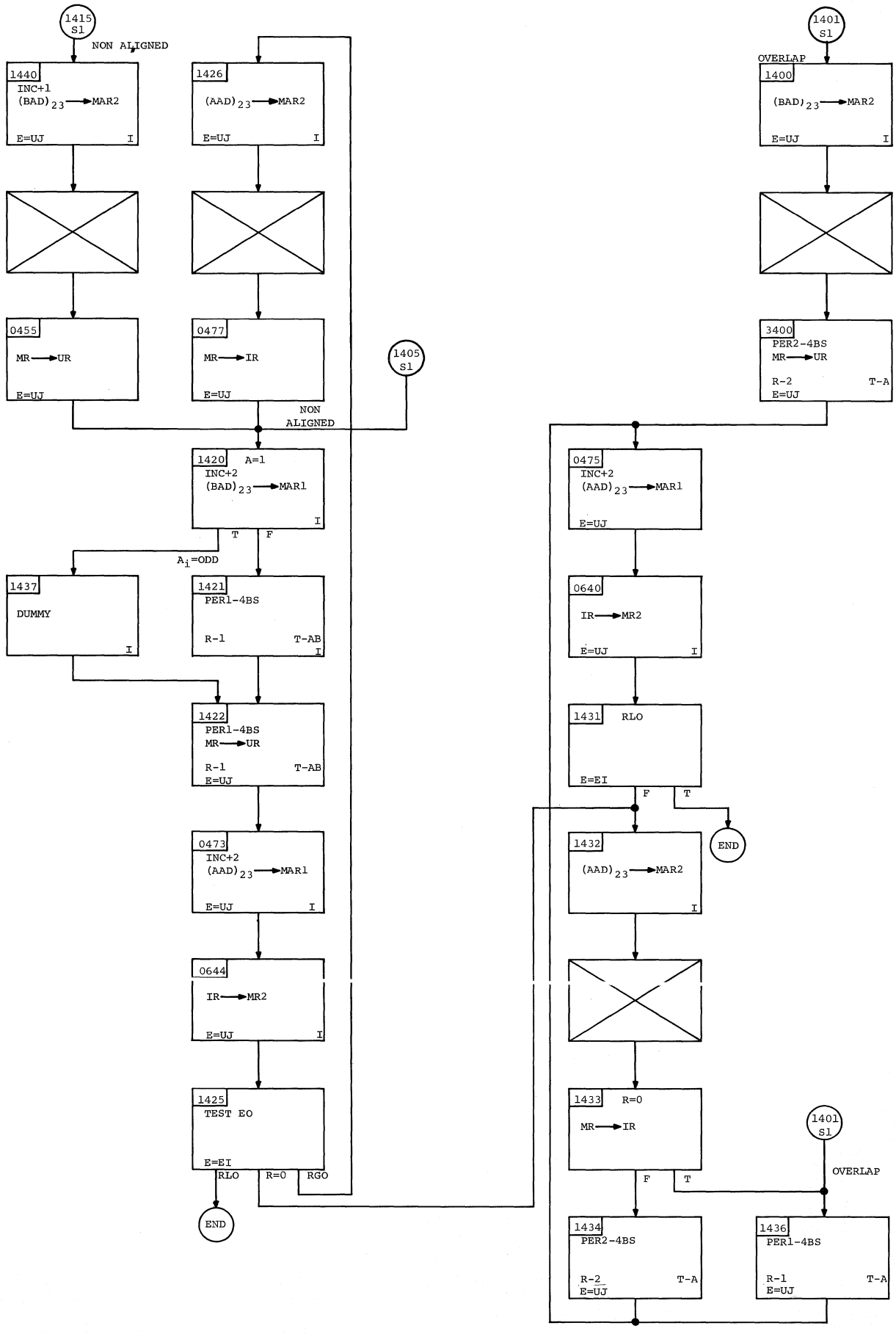
STORE SCRATCH PAD SSP SS DO



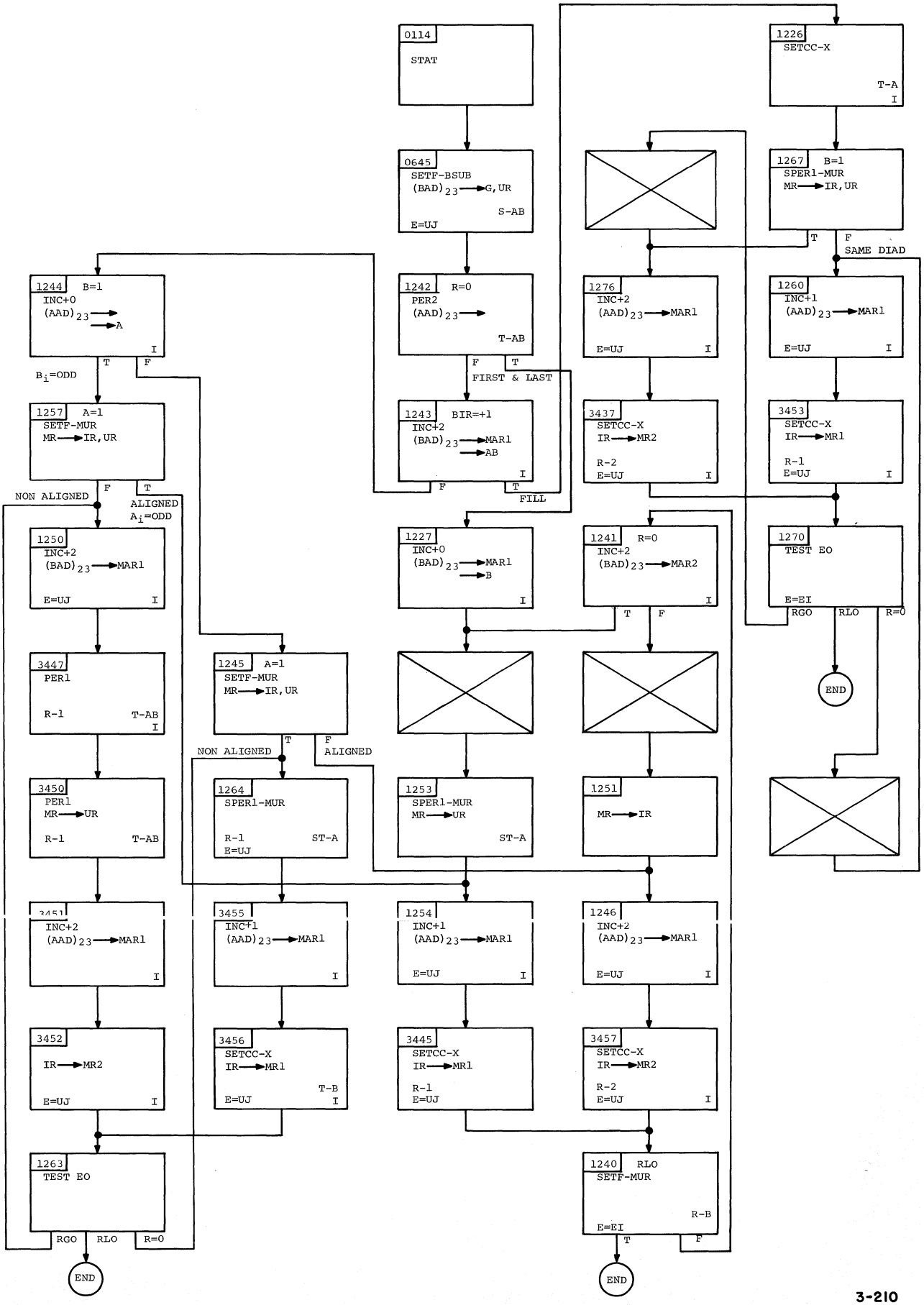
MOVE NUMERICS MVN SS DI



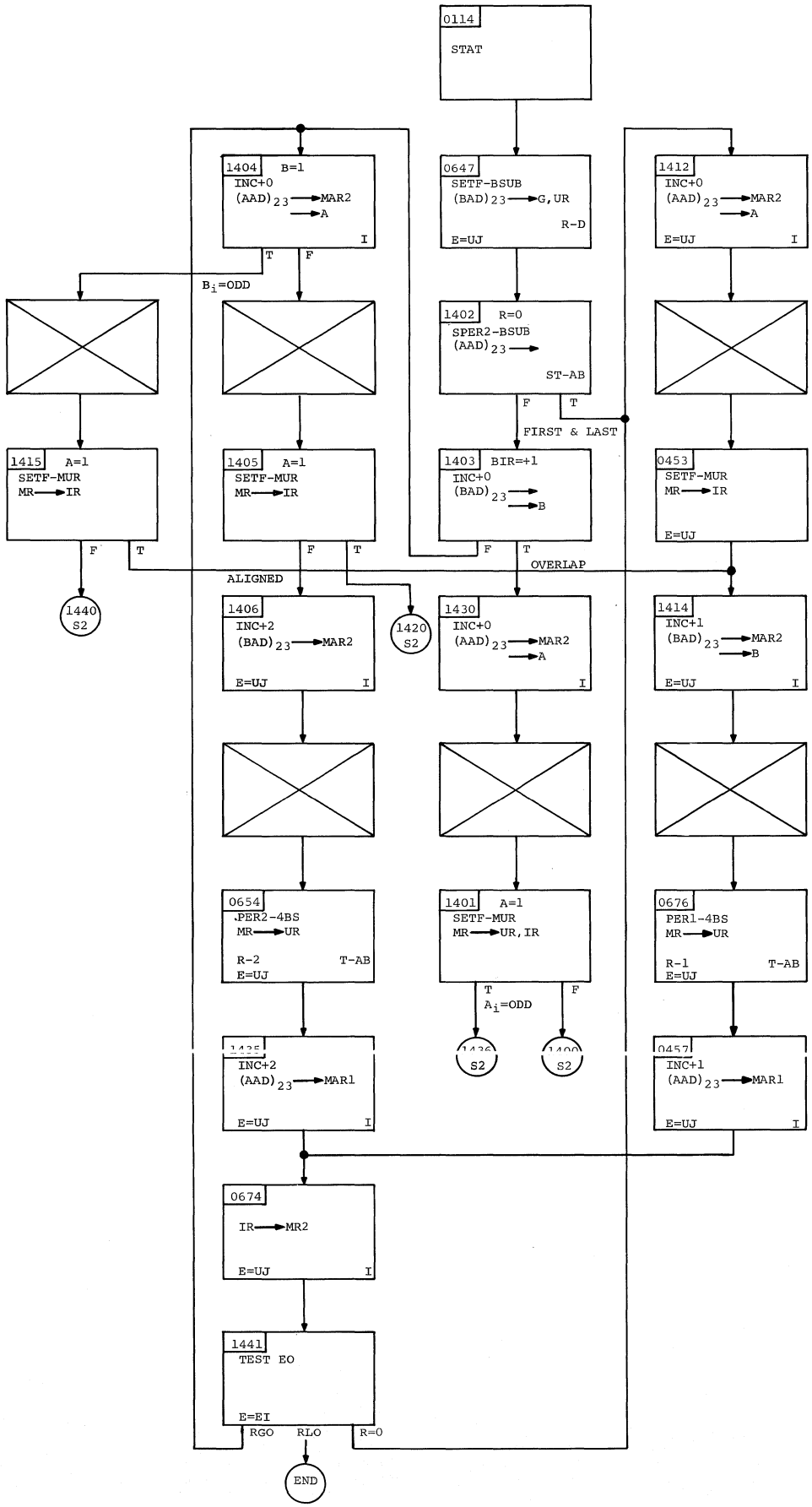
MOVE NUMERICS MVN SS DI



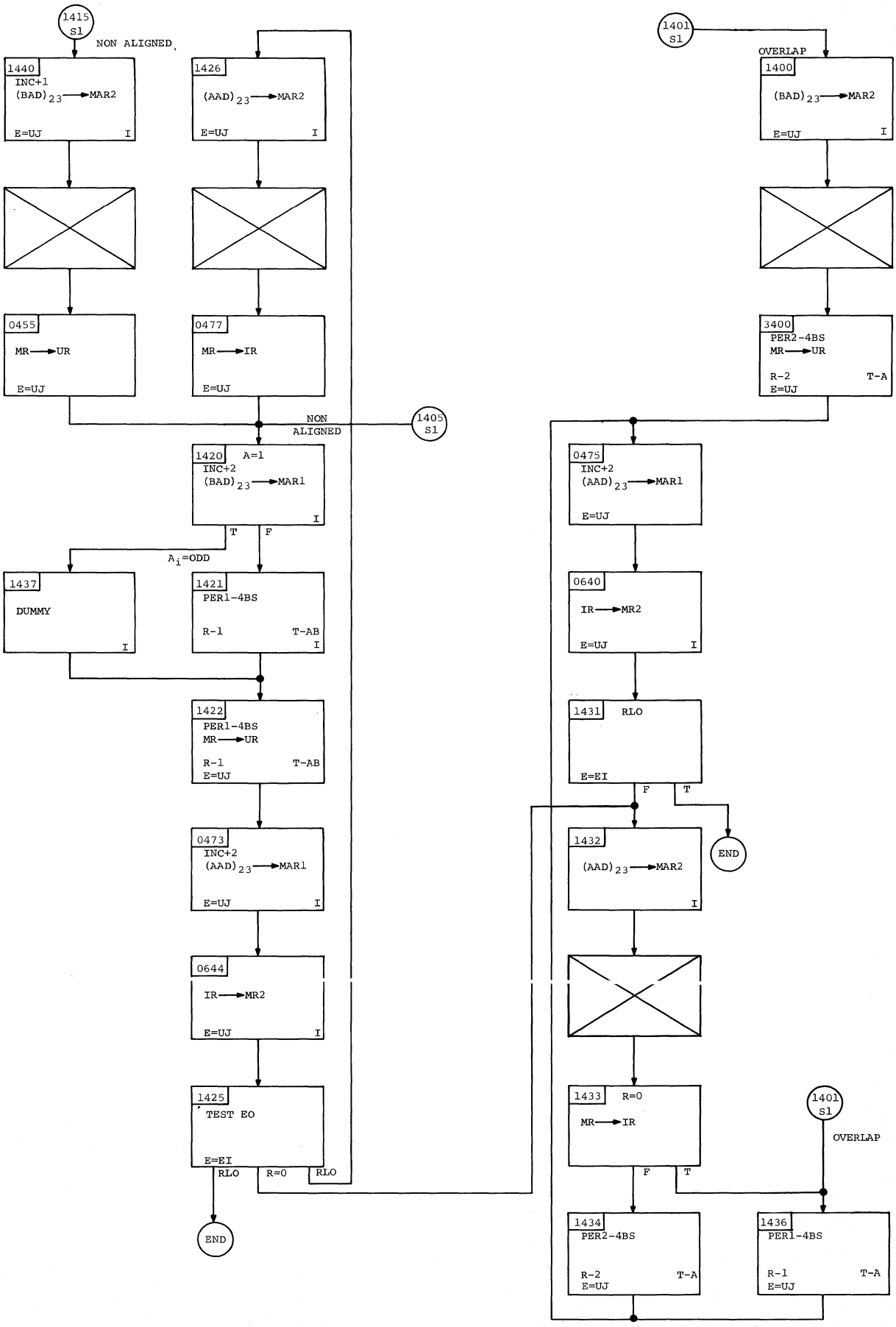
MOVE CHARACTERS MVC SS D2



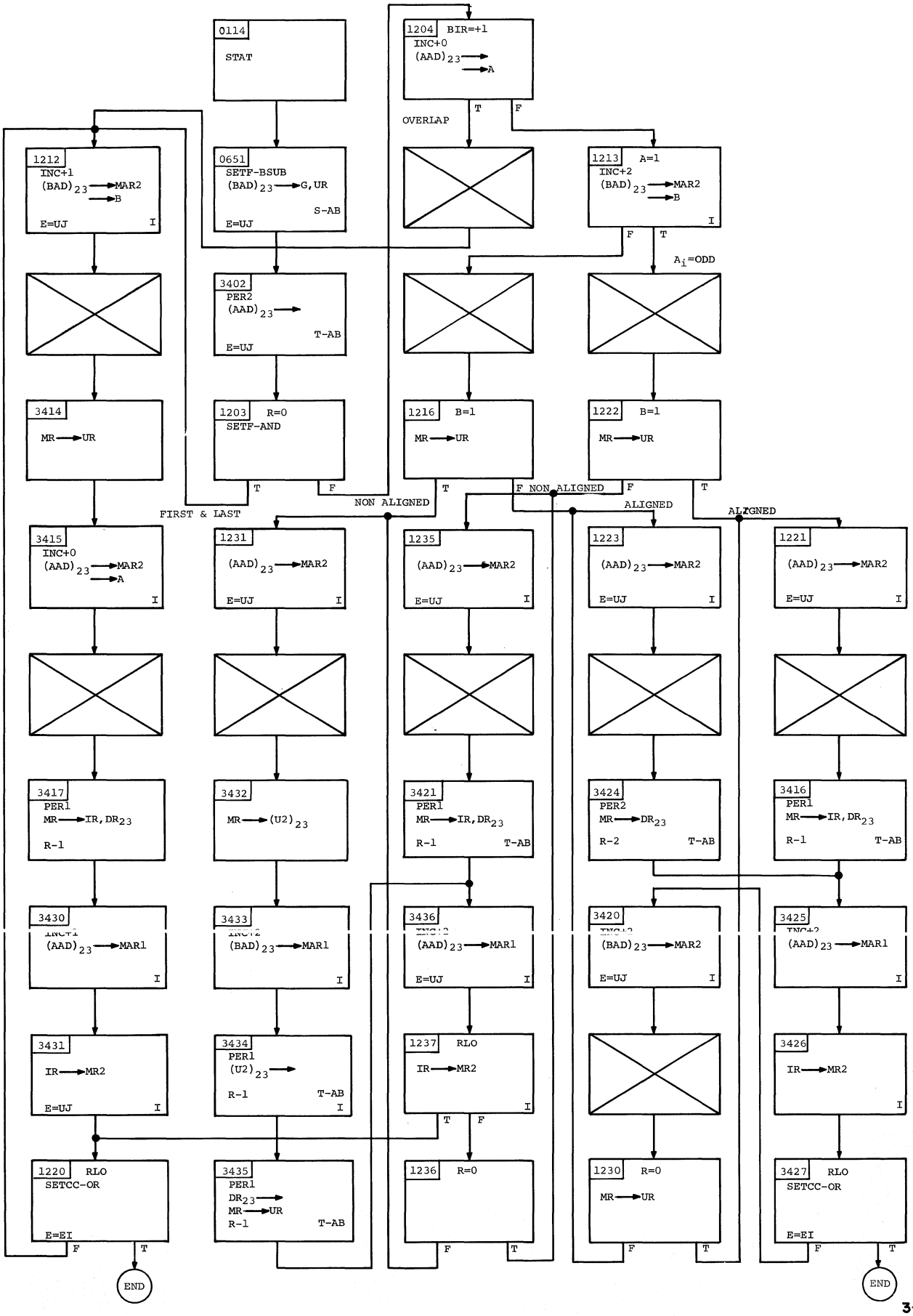
MOVE ZONES MVZ SS D3



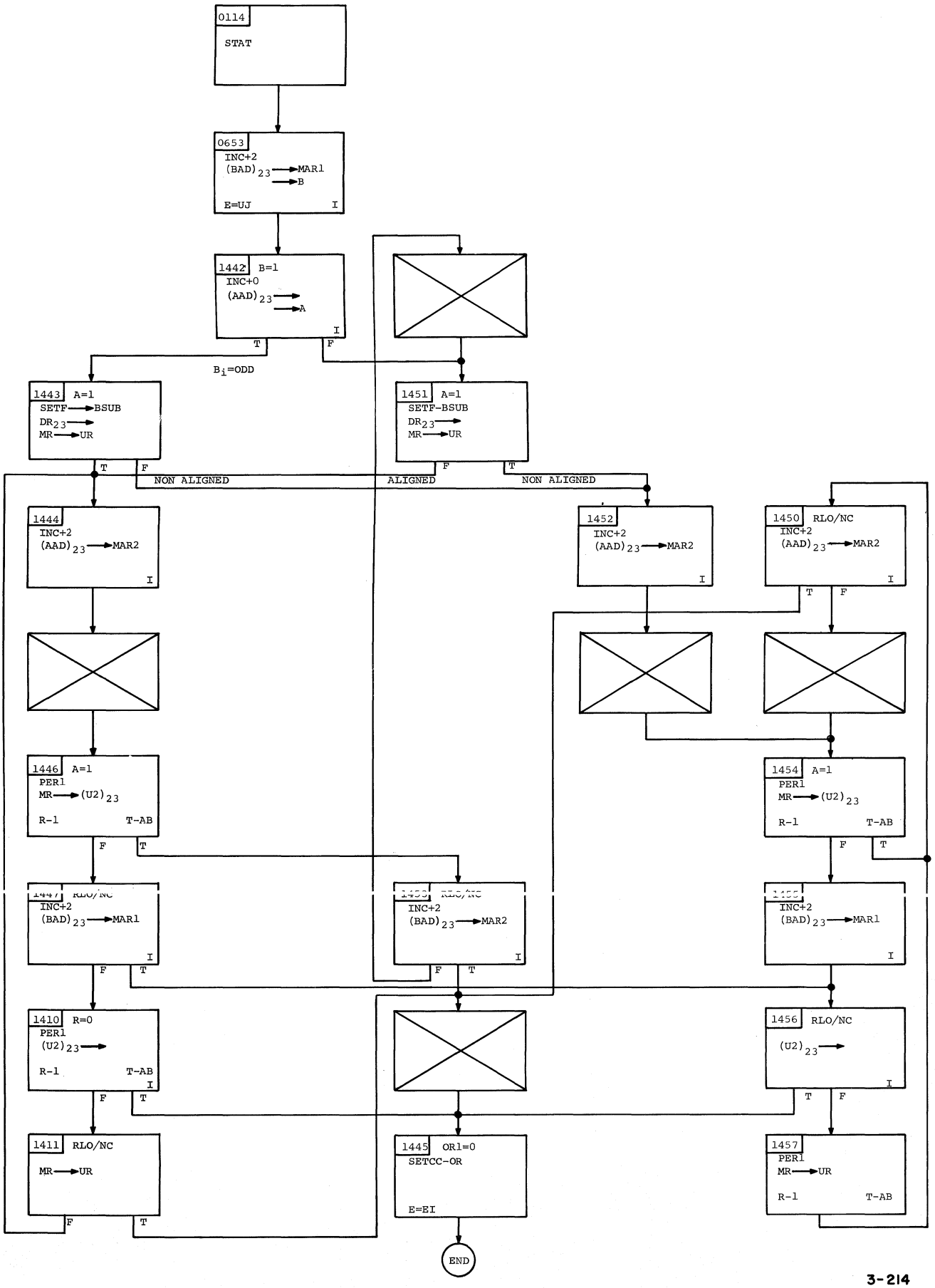
MOVE ZONES MVZ SS D3



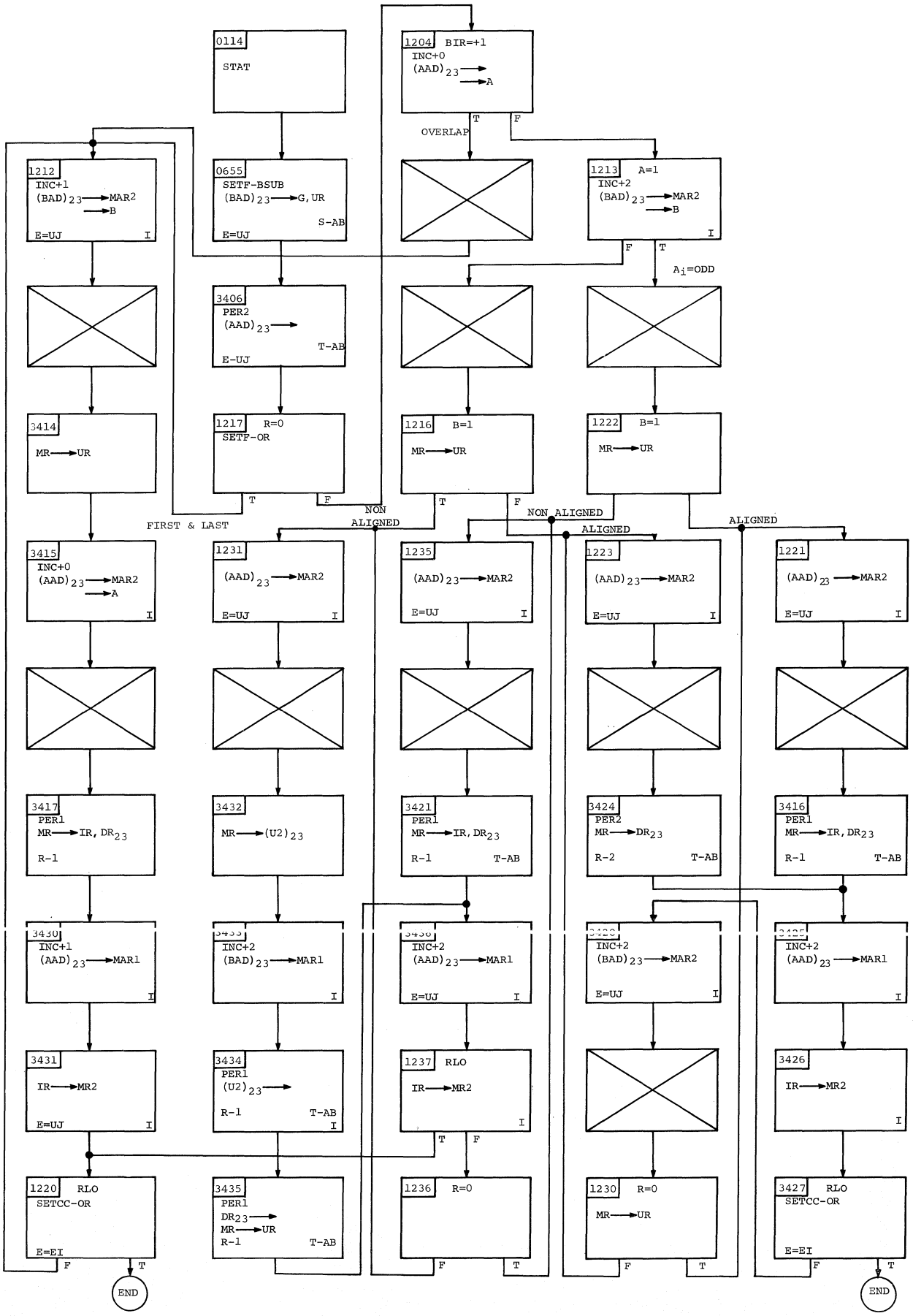
LOGICAL AND NC SS D4



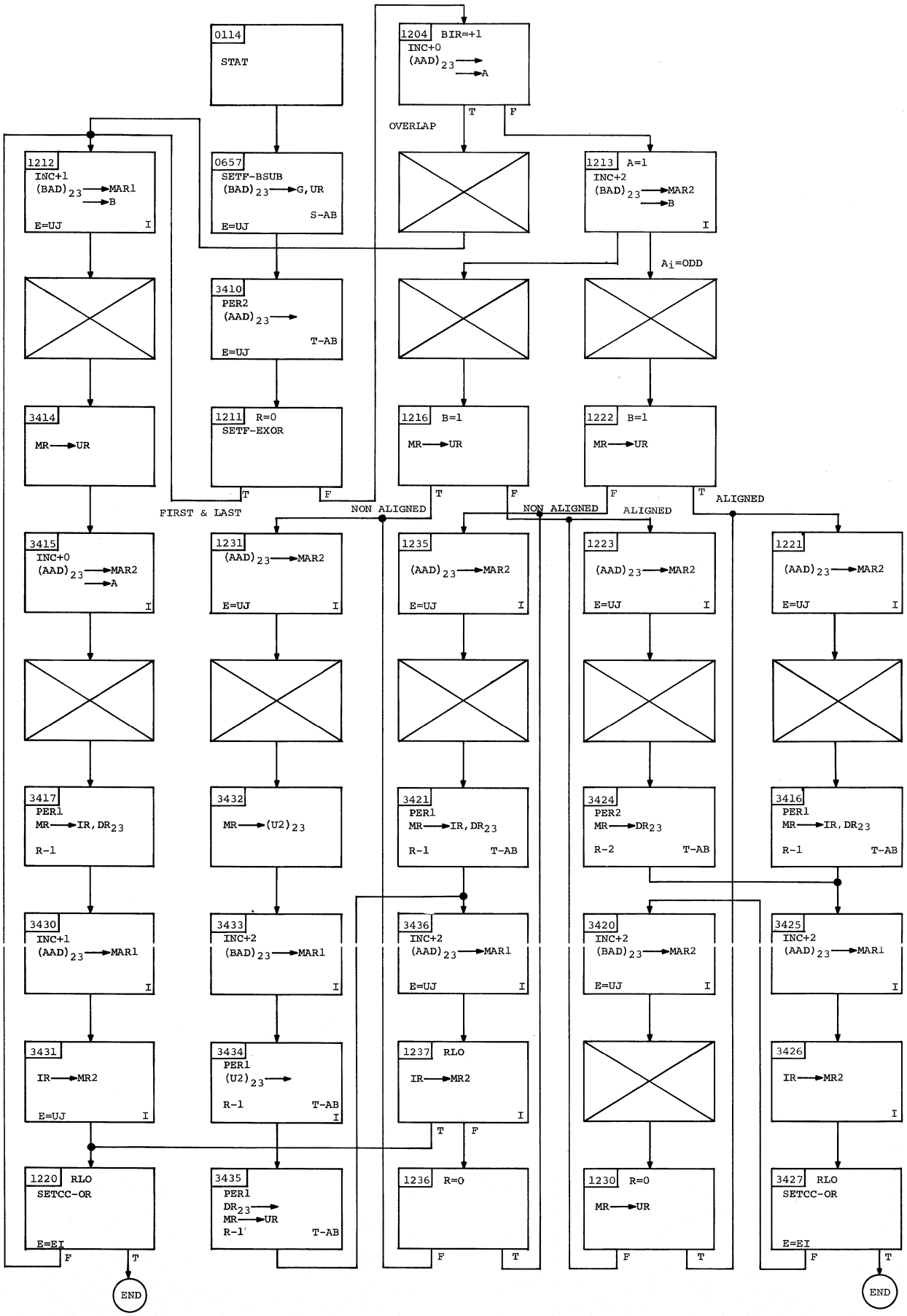
COMPARE LOGICAL CLC SS D5



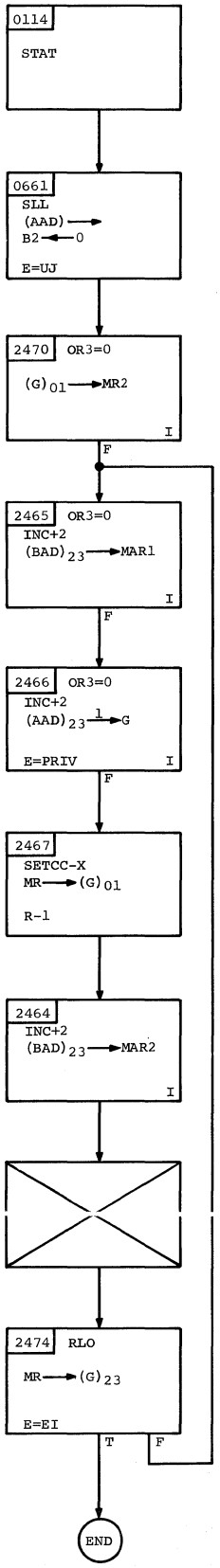
LOGICAL OR OC SS D6



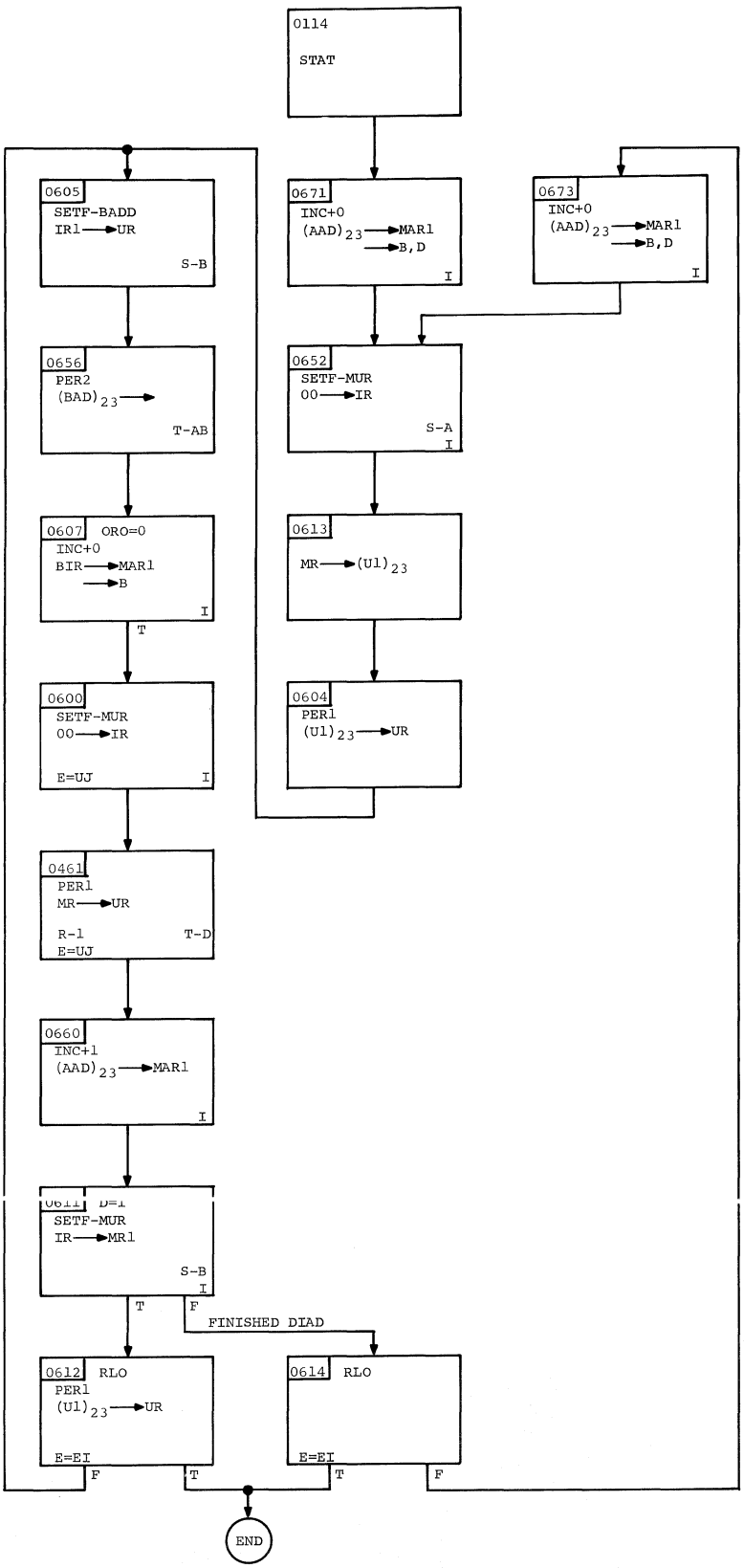
LOGICAL EXCLUSIVE OR XC SS D7



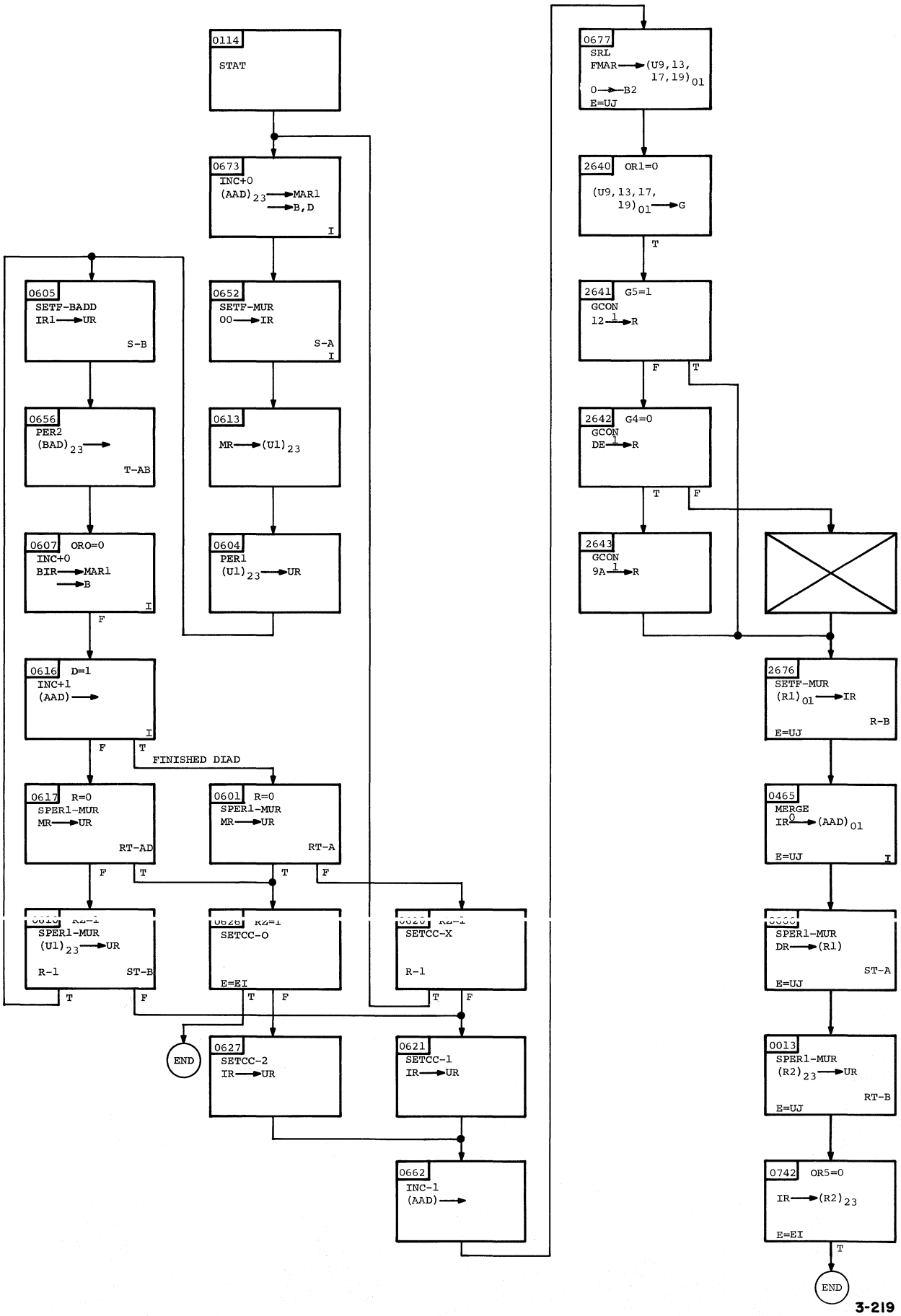
LOAD SCRATCH PAD LSP SS D8

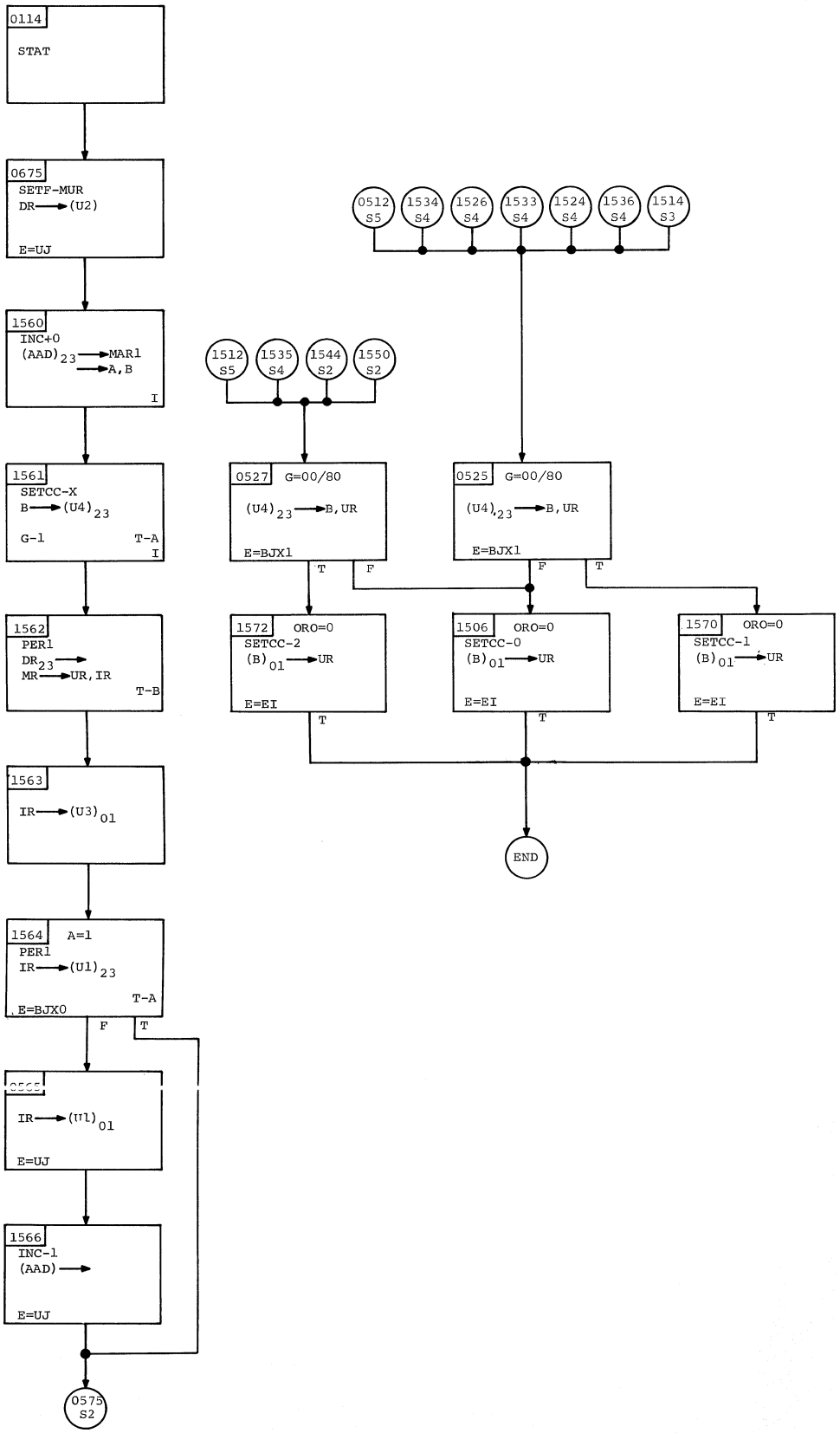


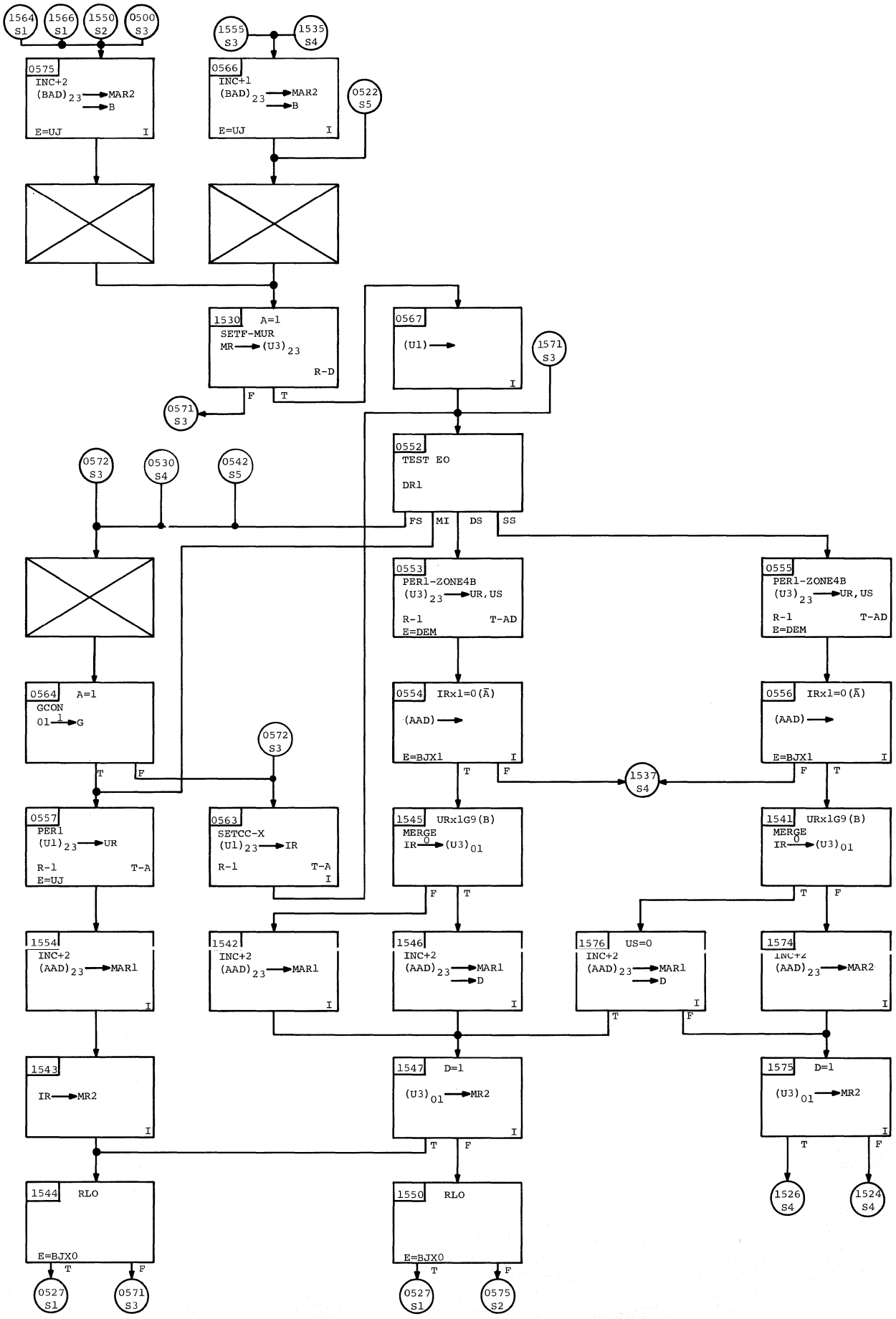
TRANSLATE TR SS DC

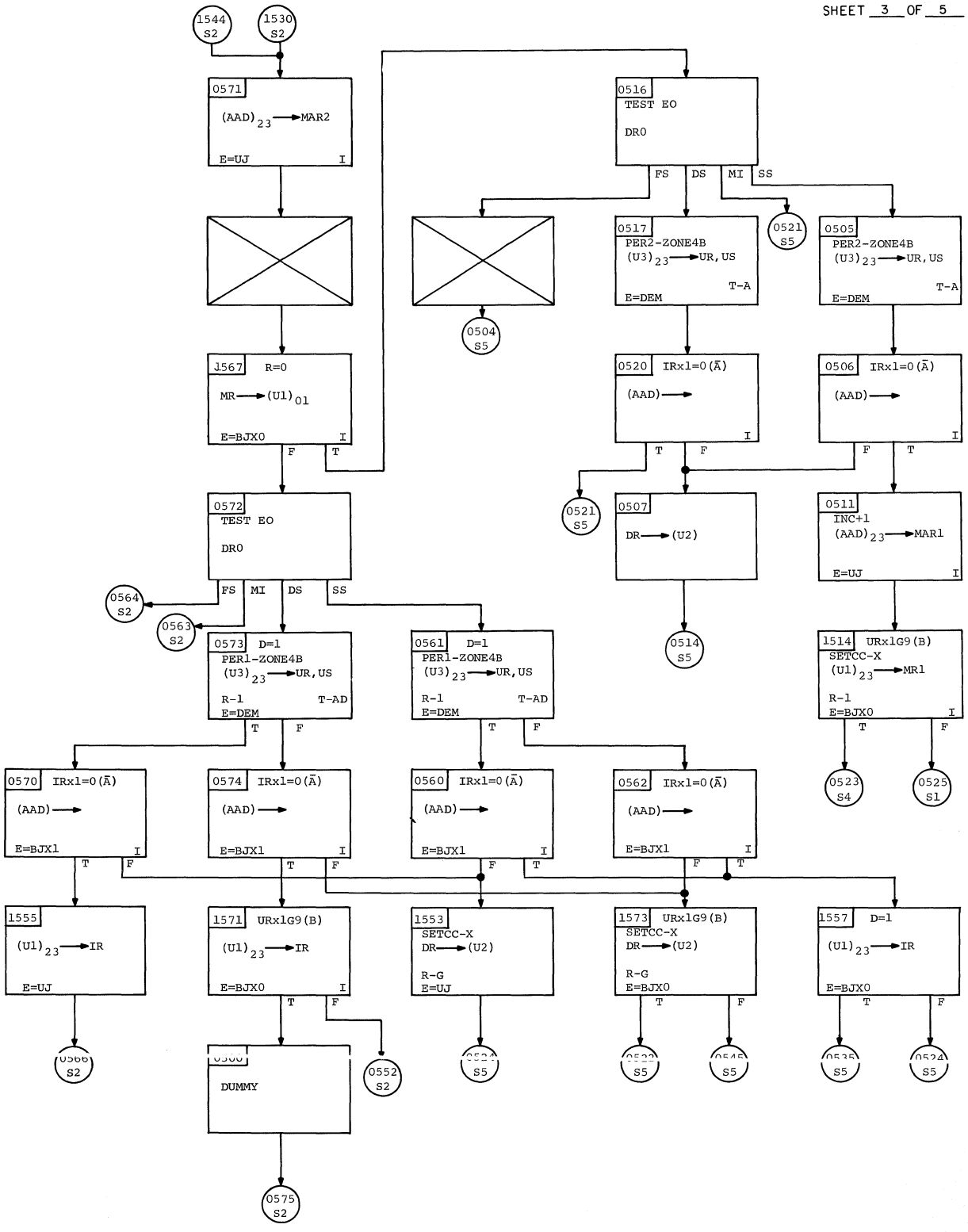


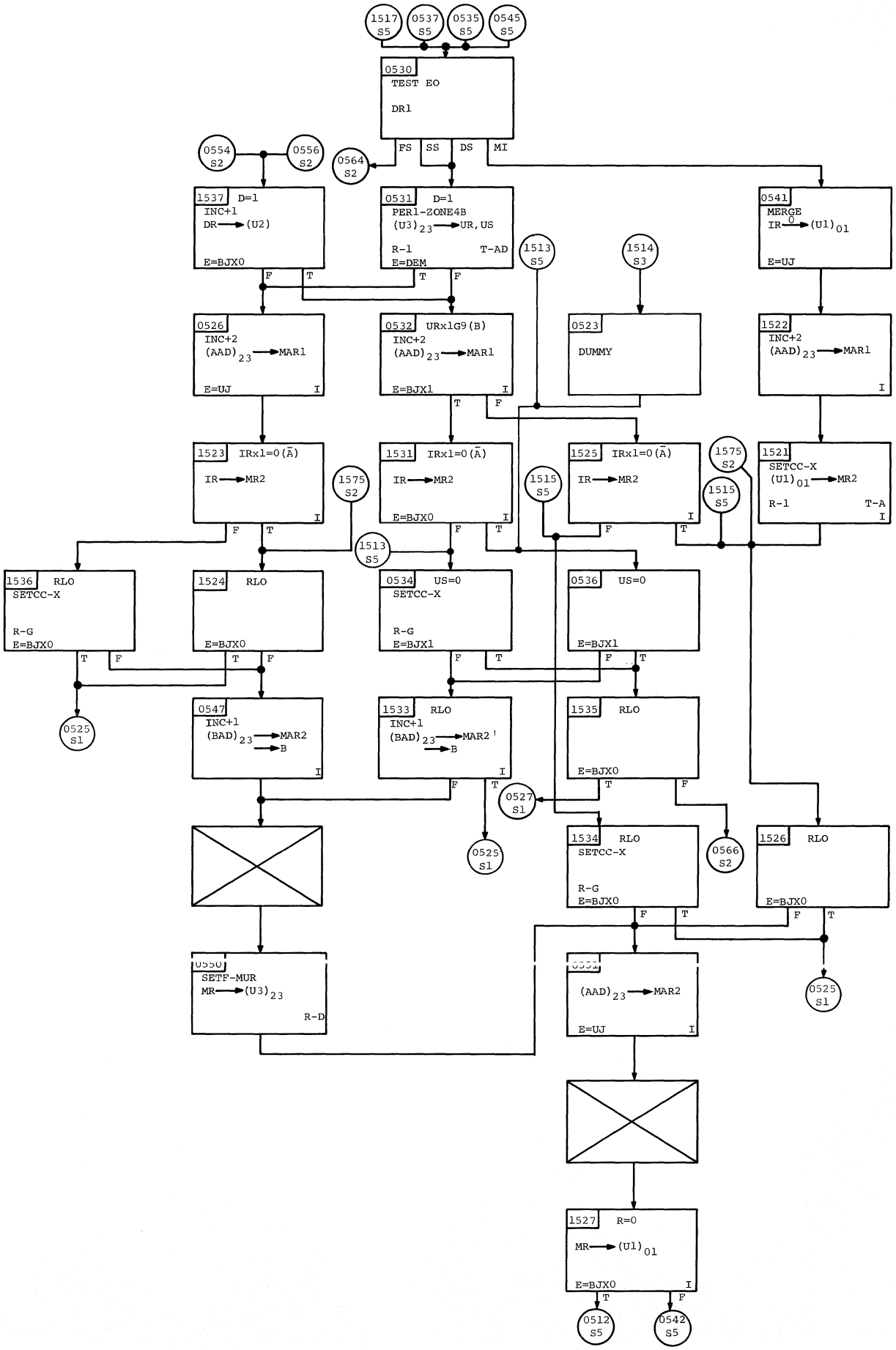
TRANSLATE AND TEST TRT SS DD

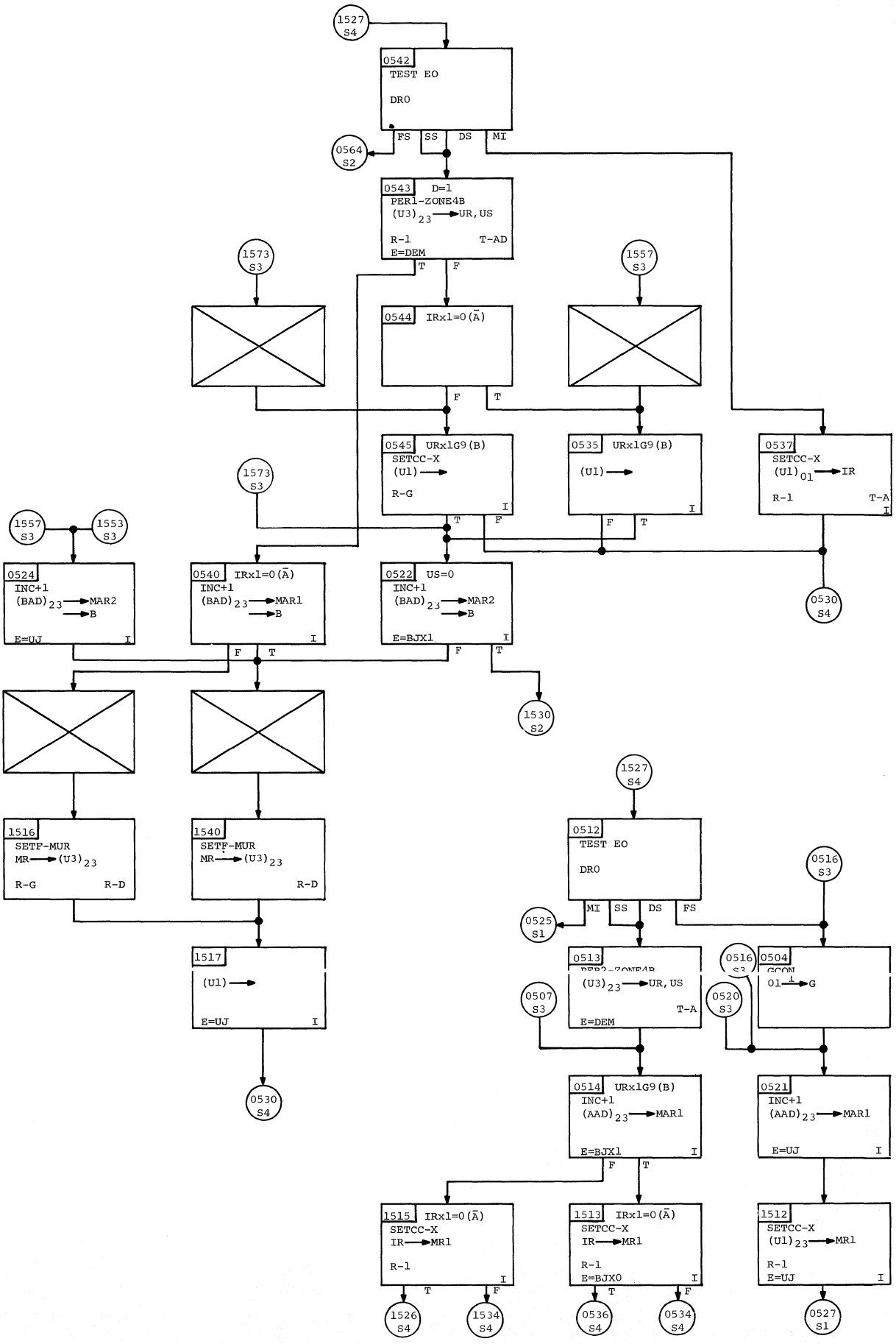



0114
STAT
0675
SETF-MUR
DR → (U2)
E=UJ
1560
INC+0
(AAD)23 → MAR1
→ A,B
I
1561
SETCC-X
B → (U4)23
G-1
T-A
I
1562
PER1
DR23 →
MR → UR,IR
T-B
1563
IR → (U3)01
1564
A=1
PER1
IR → (U1)23
T-A
E=BJX0
F
T
IR → (U1)01
E=UJ
1566
INC-1
(AAD) →
E=UJ
0575
S2
0512
S5
1534
S4
1526
S4
1533
S4
1524
S4
1536
S4
1514
S3
1512
S5
1535
S4
1544
S2
1550
S2
0527
G=00/80
(U4)23 → B,UR
E=BJX1
0525
G=00/80
(U4)23 → B,UR
E=BJX1
1572
ORO=0
SETCC-2
(B)01 → UR
E=EI
1506
ORO=0
SETCC-0
(B)01 → UR
E=EI
1570
ORO=0
SETCC-1
(B)01 → UR
E=EI
END

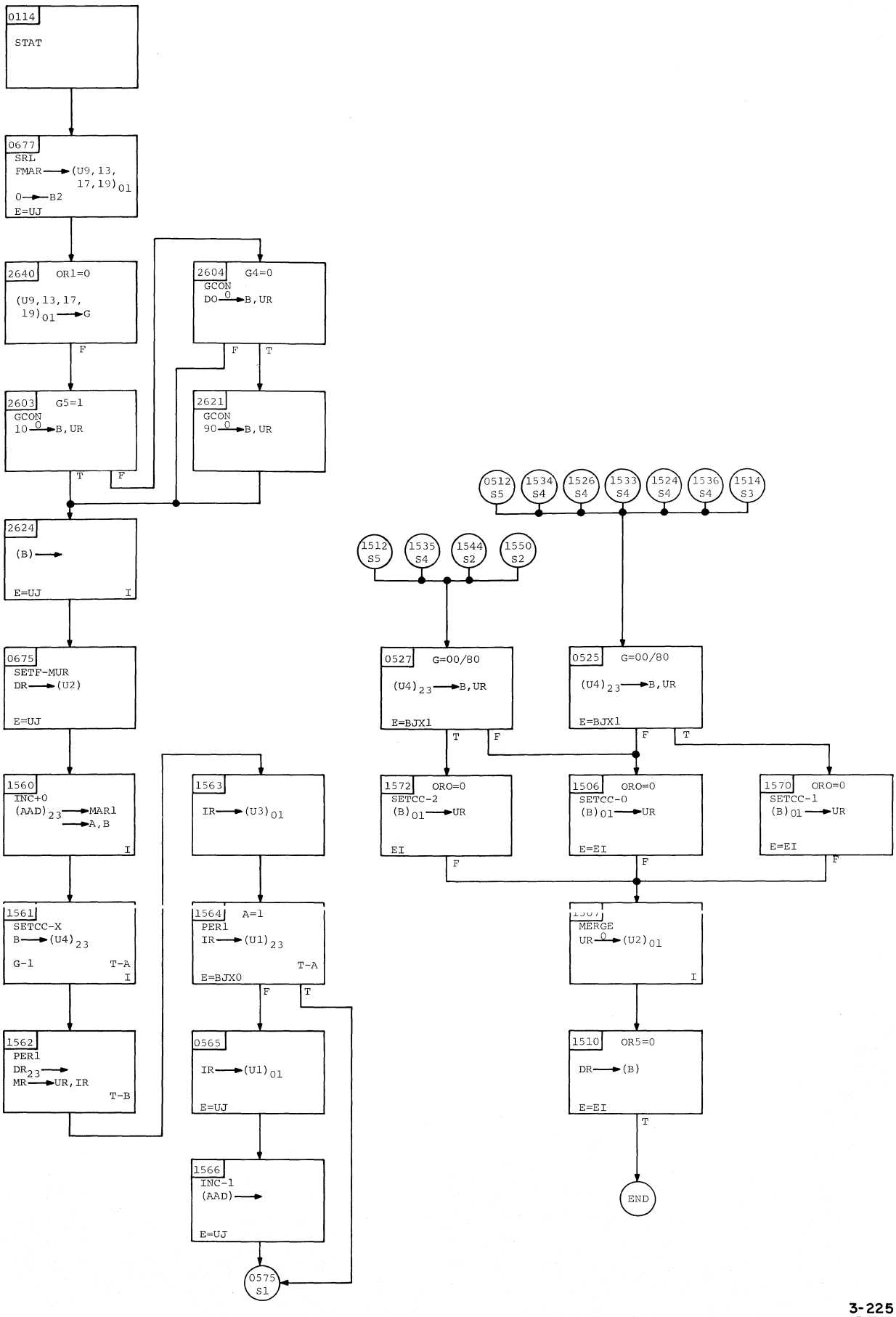

1564 S1
1566 S1
1550 S2
0500 S3
0575
INC+2
(BAD)23 → MAR2
→ B
E=UJ
I
1555 S3
1535 S4
0566
INC+1
(BAD)23 → MAR2
→ B
E=UJ
I
0522 S5
1530 A=1
SETF-MUR
MR → (U3)23
R-D
F
T
0571 S3
0567
(U1) →
I
1571 S3
0572 S3
0530 S4
0542 S5
0552
TEST EO
DR1
FS
MI
DS
SS
0553
PER1-ZONE4B
(U3)23 → UR, US
R-1
T-AD
E=DEM
0555
PER1-ZONE4B
(U3)23 → UR, US
R-1
T-AD
E=DEM
0564 A=1
GCON
01 → G
T
F
0572 S3
0554 IRxl=0(Ā)
(AAD) →
E=BJX1
I
T
F
1537 S4
0556 IRxl=0(Ā)
(AAD) →
E=BJX1
I
F
T
0557
PER1
(U1)23 → UR
R-1
T-A
E=UJ
0563
SETCC-X
(U1)23 → IR
R-1
T-A
I
1545 URxlG9(B)
MERGE
IR → (U3)01
F
T
1541 URxlG9(B)
MERGE
IR → (U3)01
T
F
1554
INC+2
(AAD)23 → MAR1
I
1542
INC+2
(AAD)23 → MAR1
I
1546
INC+2
(AAD)23 → MAR1
→ D
I
1576 US=0
INC+2
(AAD)23 → MAR1
→ D
I
T
F
1574
INC+2
(AAD)23 → MAR2
I
1543
IR → MR2
I
1547 D=1
(U3)01 → MR2
I
T
F
1575 D=1
(U3)01 → MR2
I
T
F
1544 RLO
E=BJX0
T
F
1550 RLO
E=BJX0
T
F
1526 S4
1524 S4
0527 S1
0571 S3
0527 S1
0575 S2


1544 S2
1530 S2
0571
(AAD)23 → MAR2
E=UJ I
0516
TEST EO
DR0
FS DS MI SS
0517
PER2-ZONE4B
(U3)23 → UR, US
T-A
E=DEM
0521 S5
0505
PER2-ZONE4B
(U3)23 → UR, US
T-A
E=DEM
0504 S5
1567 R=0
MR → (U1)01
E=BJX0 I
F T
0520 IRx1=0(Ā)
(AAD) →
I
0506 IRx1=0(Ā)
(AAD) →
I
T F
F T
0521 S5
0572
TEST EO
DR0
0507
DR → (U2)
0511
INC+1
(AAD)23 → MAR1
E=UJ I
0564 S2
0563 S2
FS MI DS SS
0573 D=1
PER1-ZONE4B
(U3)23 → UR, US
R-1 T-AD
E=DEM
0561 D=1
PER1-ZONE4B
(U3)23 → UR, US
R-1 T-AD
E=DEM
0514 S5
1514 URx1G9(B)
SETCC-X
(U1)23 → MR1
R-1
E=BJX0 I
T F
0523 S4
0525 S1
0570 IRx1=0(Ā)
(AAD) →
E=BJX1 I
0574 IRx1=0(Ā)
(AAD) →
E=BJX1 I
0560 IRx1=0(Ā)
(AAD) →
E=BJX1 I
0562 IRx1=0(Ā)
(AAD) →
E=BJX1 I
T F
1555
(U1)23 → IR
E=UJ
1571 URx1G9(B)
(U1)23 → IR
E=BJX0 I
1553
SETCC-X
DR → (U2)
R-G
E=UJ
1573 URx1G9(B)
SETCC-X
DR → (U2)
R-G
E=BJX0
1557 D=1
(U1)23 → IR
E=BJX0
0566 S2
DUMMY
0552 S2
0575 S2
S5
S5
S5
S5
S5

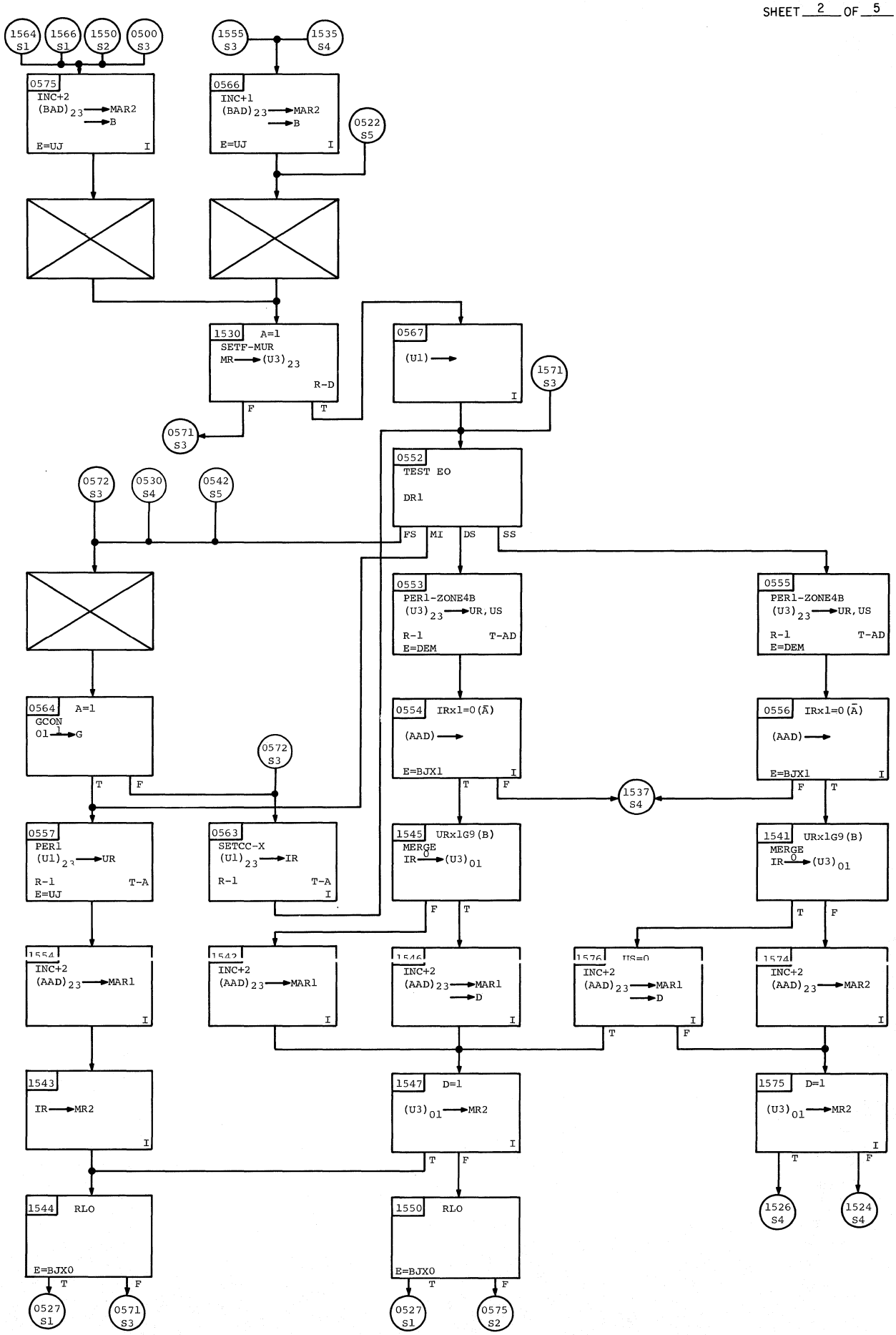

1517 S5
0537 S5
0535 S5
0545 S5
0530 TEST EO DR1
FS SS DS MI
0554 S2
0556 S2
0564 S2
1537 D=1 INC+1 DR→(U2) E=BJX0
0531 D=1 PER1-ZONE4B (U3)23→UR,US R-1 T-AD E=DEM
1513 S5
1514 S3
0541 MERGE IR→(U1)01 E=UJ
0526 INC+2 (AAD)23→MAR1 E=UJ I
0532 URx1G9(B) INC+2 (AAD)23→MAR1 E=BJX1 I
0523 DUMMY
1522 INC+2 (AAD)23→MAR1 I
1523 IRx1=0(Ā) IR→MR2 I
1531 IRx1=0(Ā) IR→MR2 E=BJX0 I
1525 IRx1=0(Ā) IR→MR2 I
1575 S2
1515 S5
1521 SETCC-X (U1)01→MR2 R-1 T-A I
1536 RLO SETCC-X R-G E=BJX0
1524 RLO E=BJX0
0534 US=0 SETCC-X R-G E=BJX1
0536 US=0 E=BJX1
0525 S1
0547 INC+1 (BAD)23→MAR2 →B I
1533 RLO INC+1 (BAD)23→MAR2 →B I
1535 RLO E=BJX0
0527 S1
1534 RLO SETCC-X R-G E=BJX0
0566 S2
1526 RLO E=BJX0
0550 SETF-MUR MR→(U3)23 R-D
(AAD)23→MAR2 E=UJ I
1527 R=0 MR→(U1)01 E=BJX0 I
0512 S5
0542 S5


1527 S4
0542 TEST EO DR0
FS SS DS MI
0564 S2
0543 D=1 PER1-ZONE4B (U3)23 → UR, US R-1 T-AD E=DEM
T F
1573 S3
1557 S3
0544 IRx1=0(Ā)
F T
0545 URx1G9(B) SETCC-X (U1) → R-G I
0535 URx1G9(B) (U1) → I
0537 SETCC-X (U1)01 → IR R-1 T-A I
1573 S3
1557 S3
1553 S3
0524 INC+1 (BAD)23 → MAR2 → B E=UJ I
0540 IRx1=0(Ā) INC+1 (BAD)23 → MAR1 → B I
0522 US=0 INC+1 (BAD)23 → MAR2 → B E=BJX1 I
0530 S4
1530 S2
1516 SETF-MUR MR → (U3)23 R-G R-D
1540 SETF-MUR MR → (U3)23 R-D
1517 (U1) → E=UJ I
0530 S4
1527 S4
0512 TEST EO DR0
MI SS DS FS
0516 S3
0525 S1
0513 (U3)23 → UR, US T-A E=DEM
0516 S3
0504 G00N 01 → G
0507 S3
0520 S3
0514 URx1G9(B) INC+1 (AAD)23 → MAR1 E=BJX1 I
0521 INC+1 (AAD)23 → MAR1 E=UJ I
1515 IRx1=0(Ā) SETCC-X IR → MR1 R-1 I
1513 IRx1=0(Ā) SETCC-X IR → MR1 R-1 E=BJX0 I
1512 SETCC-X (U1)23 → MR1 R-1 E=UJ I
T F
1526 S4
1534 S4
0536 S4
0534 S4
0527 S1

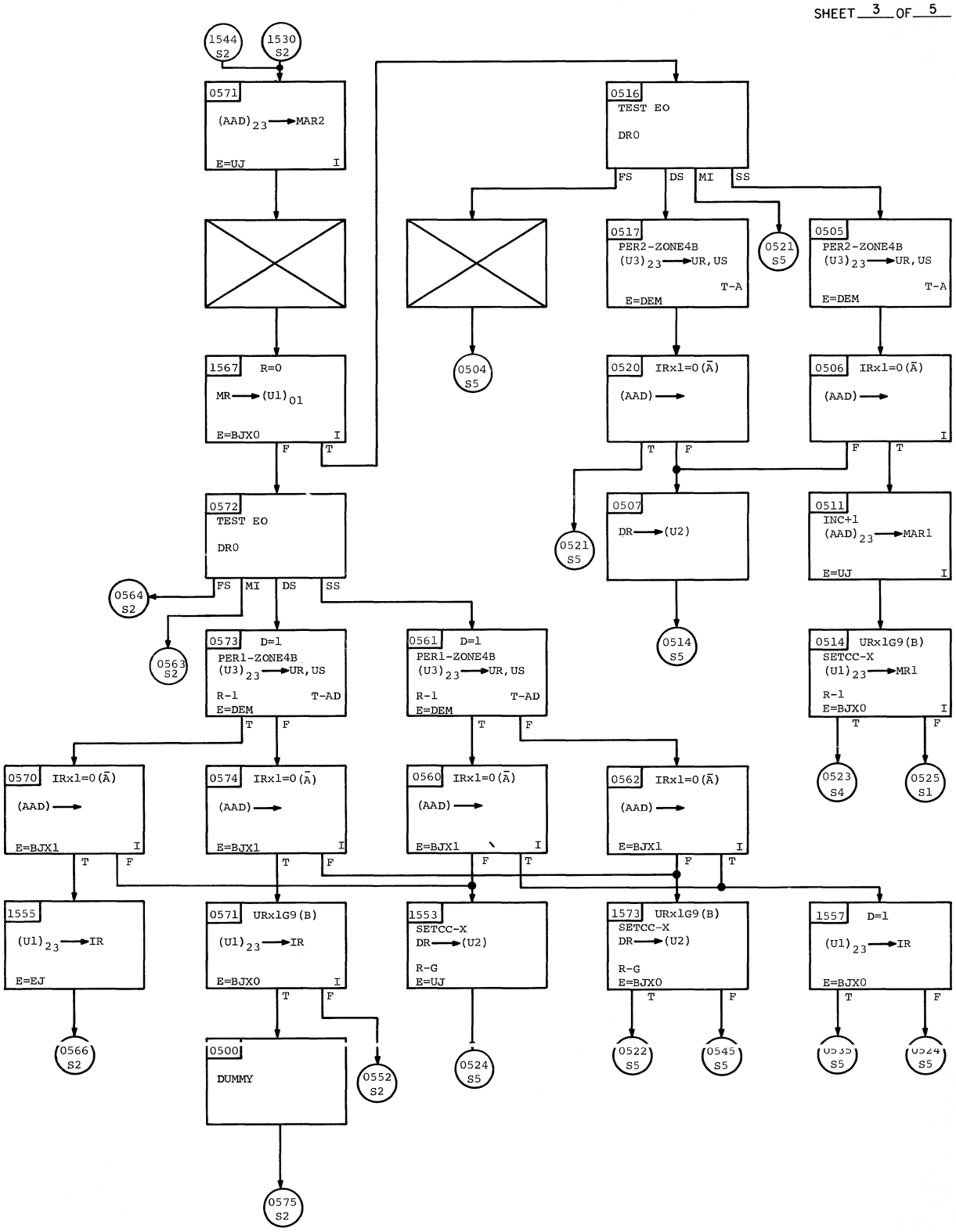
EDIT AND MARK EDMK SS DF



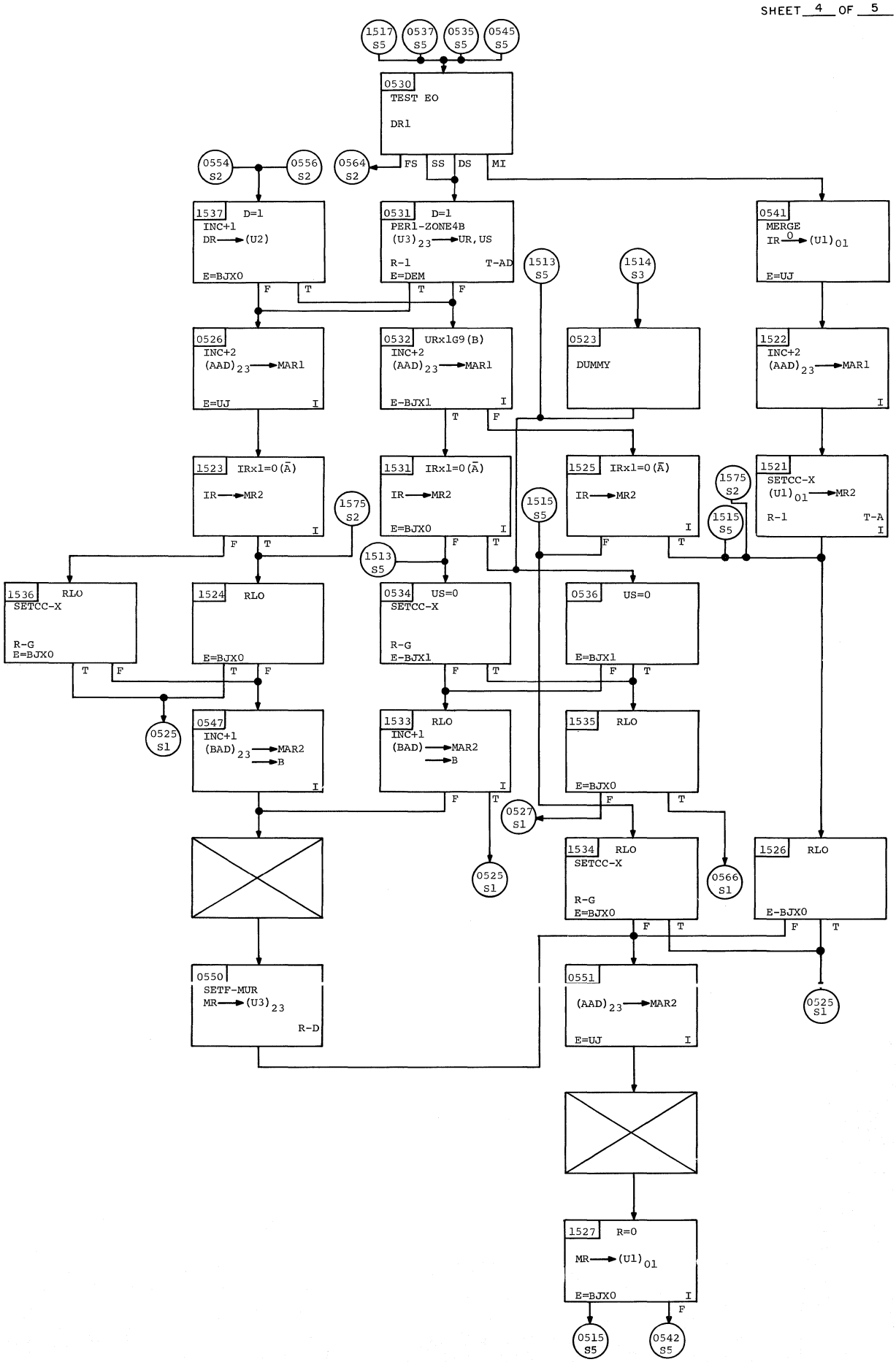
EDIT AND MARK EDMK SS DF



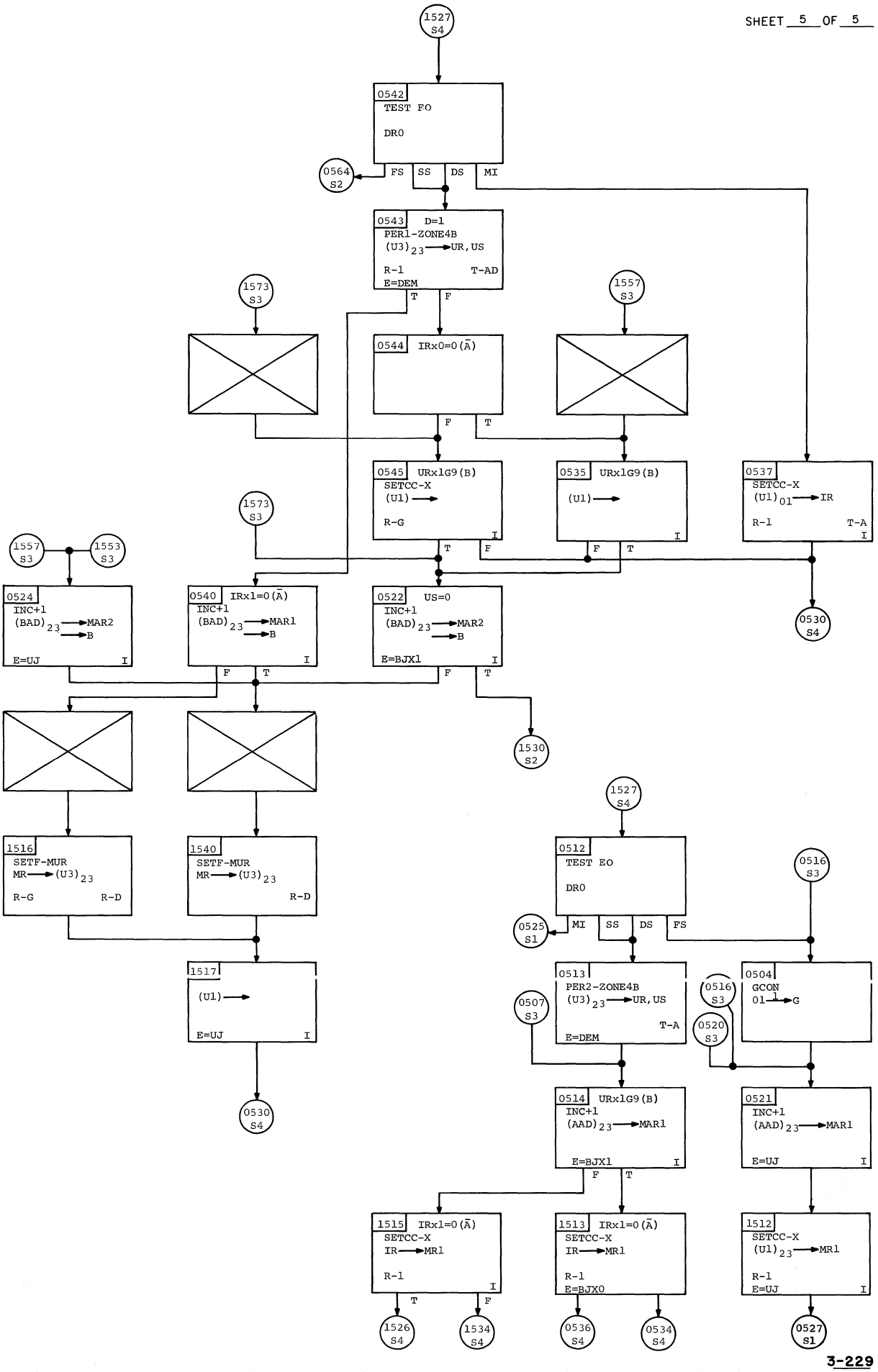
EDIT AND MARK EDMK SS DF



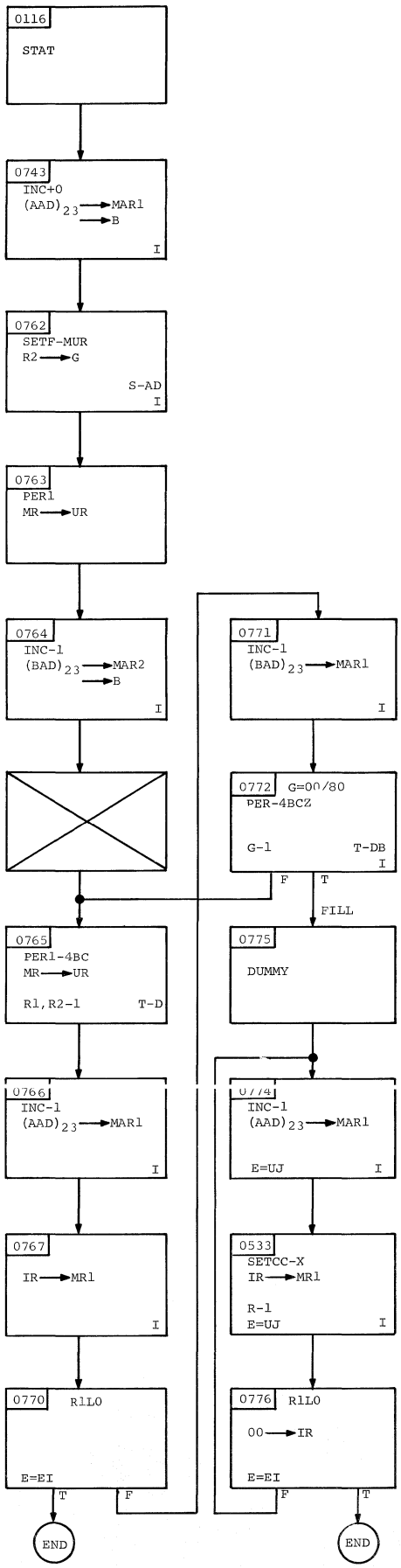
EDIT AND MARK EDMK SS DF



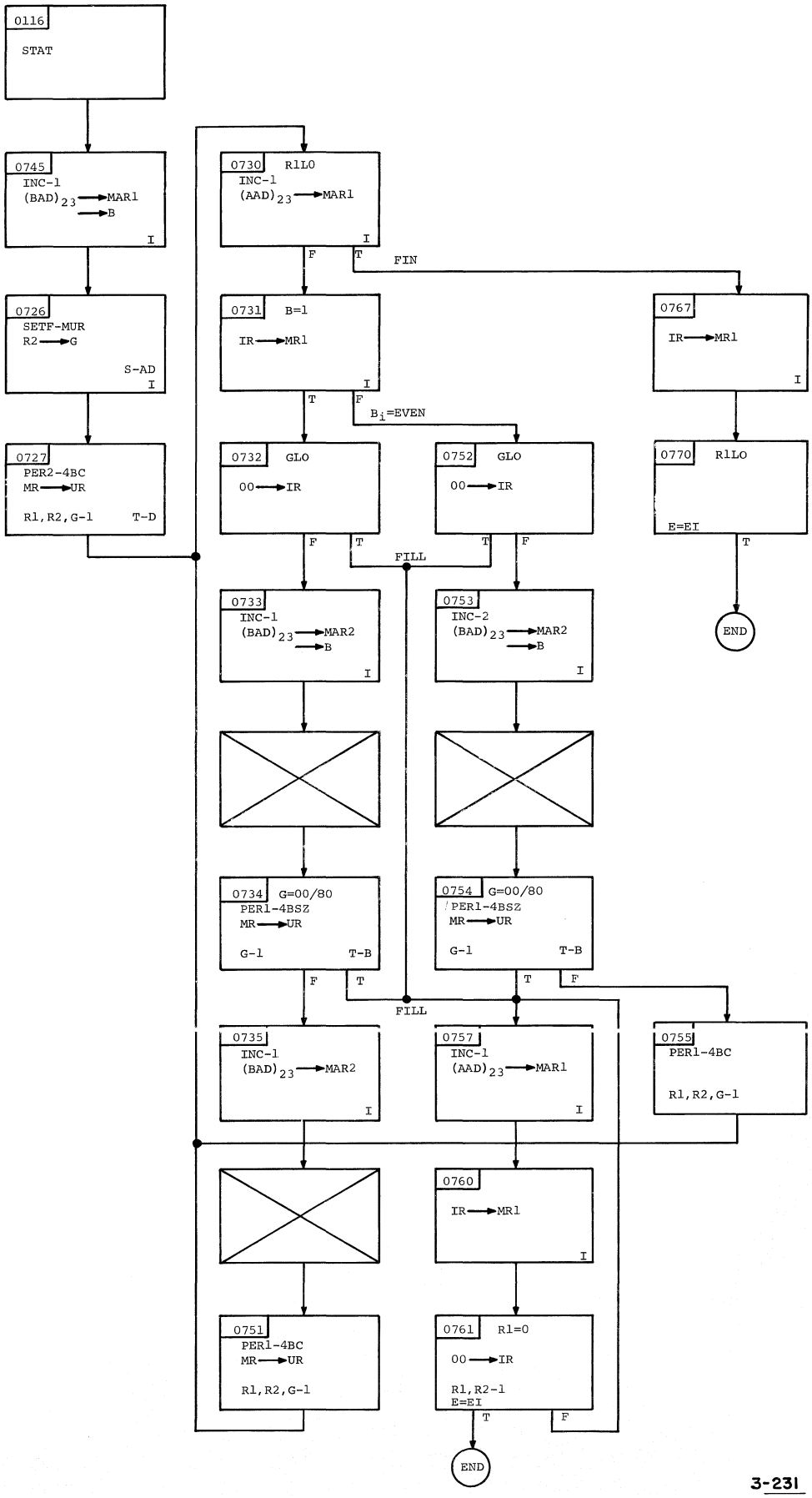
EDIT AND MARK EDMK SS DF



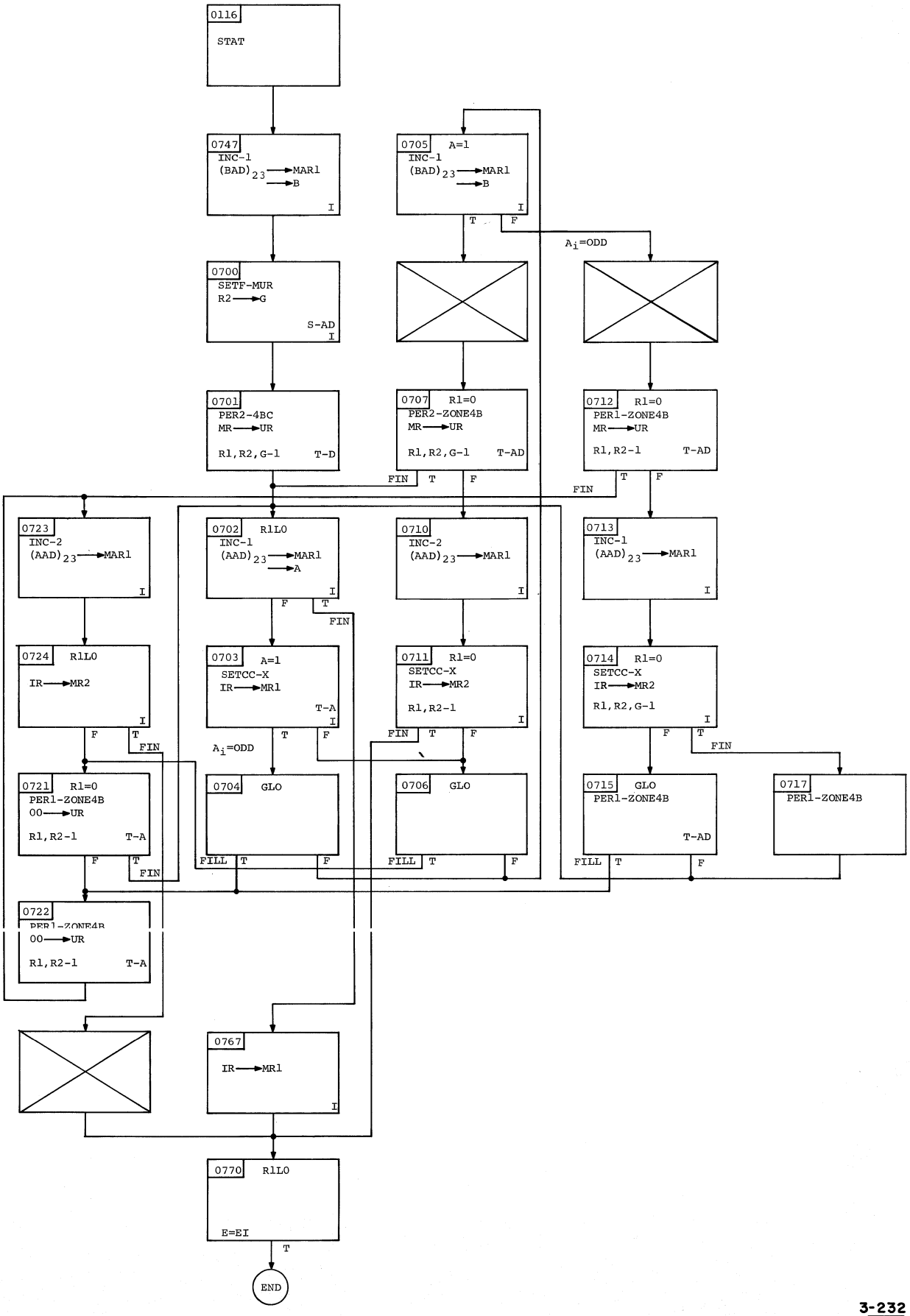
MOVE WITH OFFSET MVO SS FI



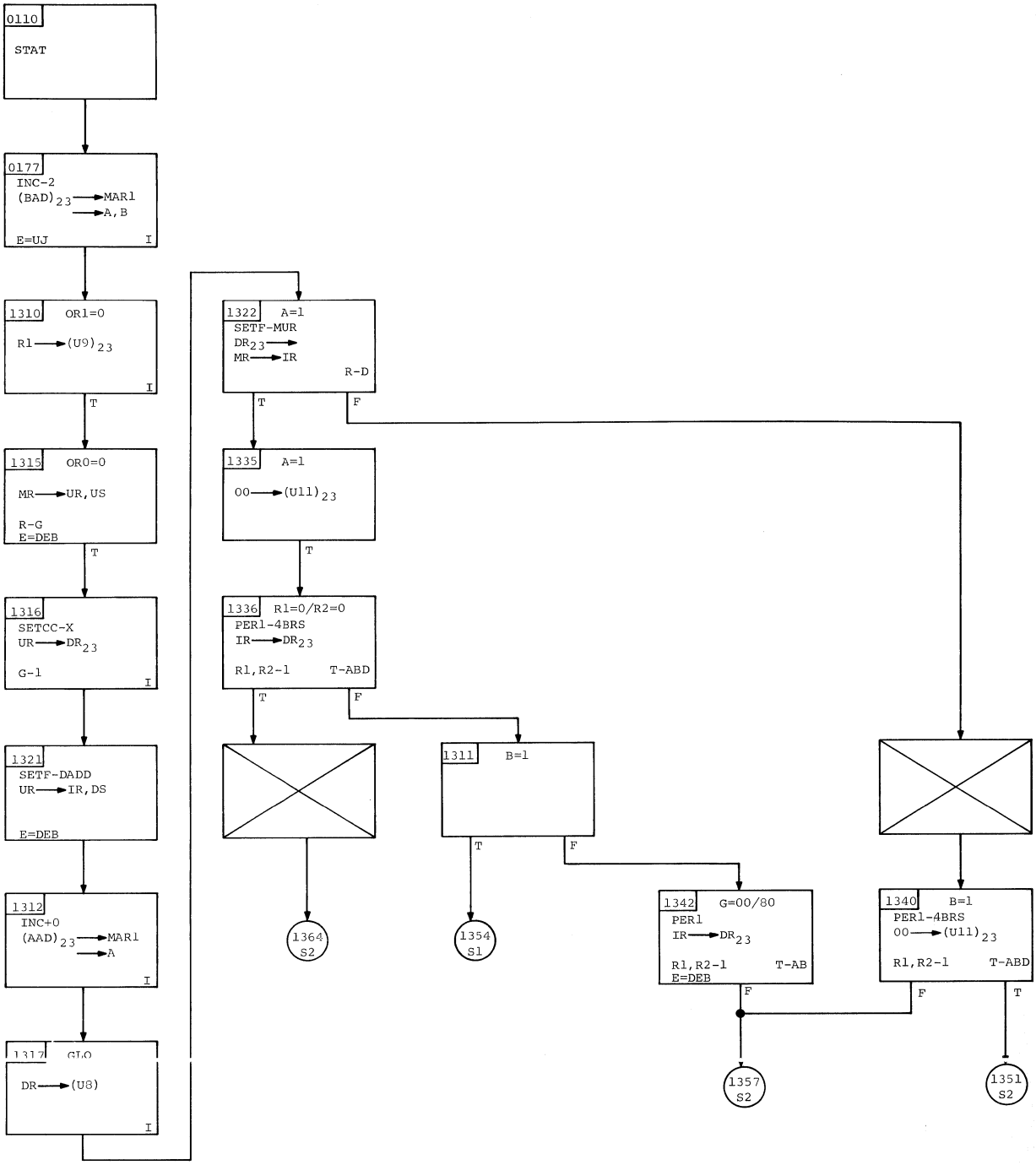
PACK PACK SS F2



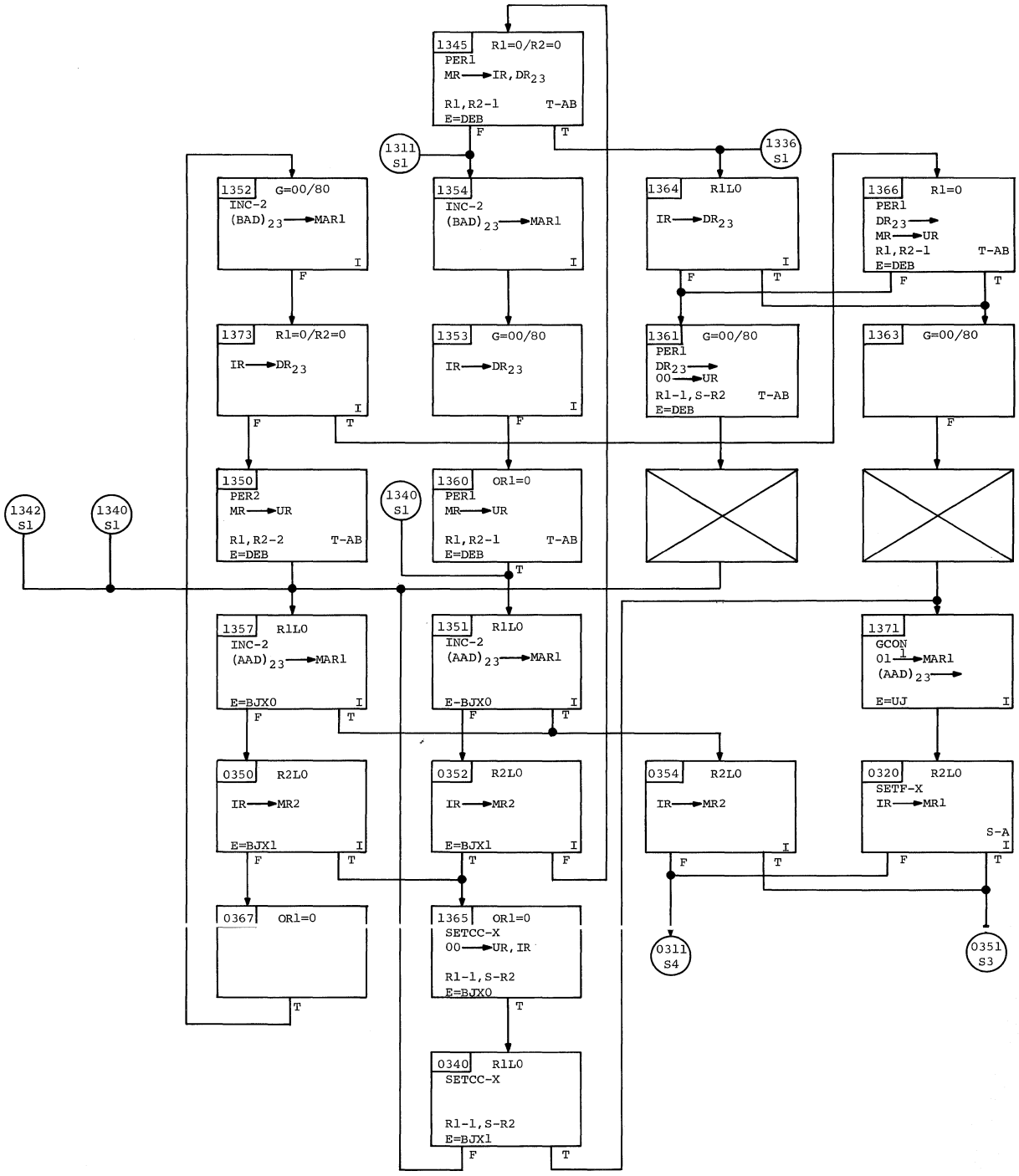
UNPACK UNPK SS F3



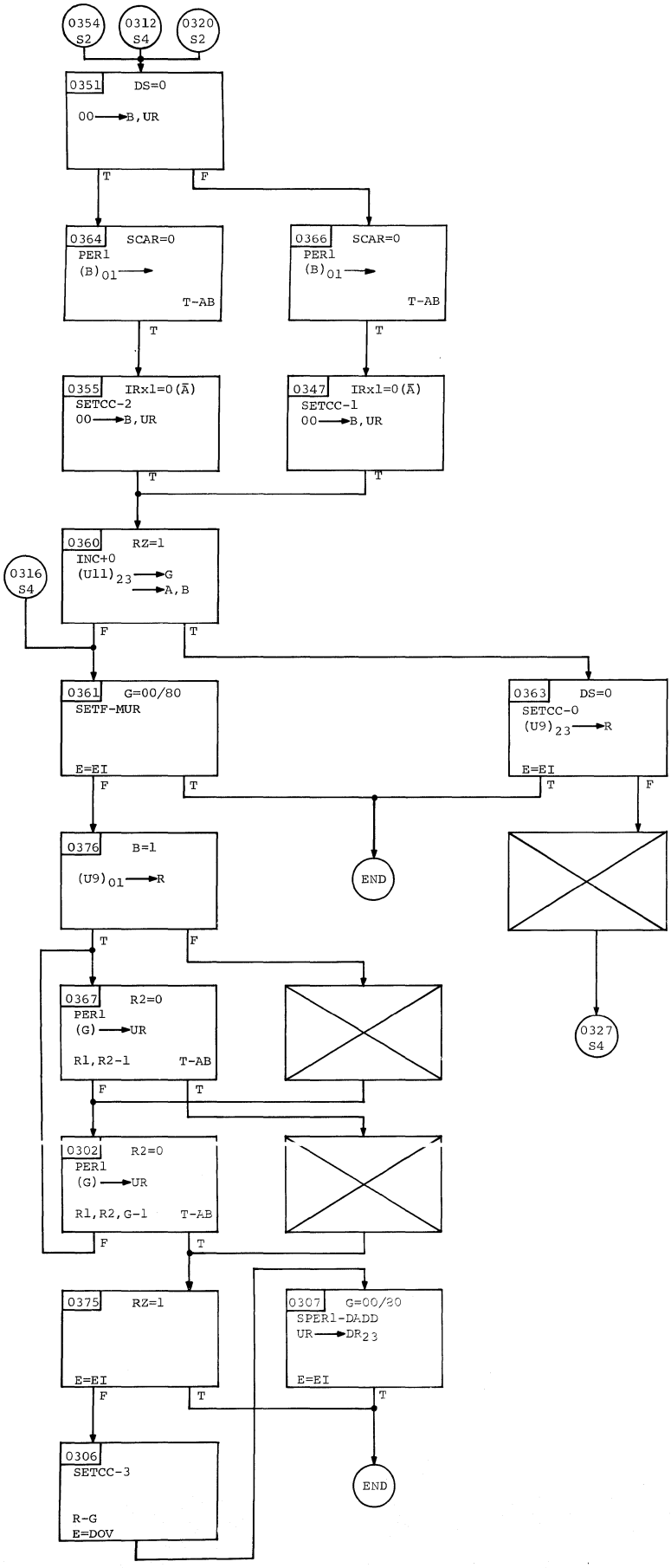
ZERO AND ADD DECIMAL ZAP SS F8



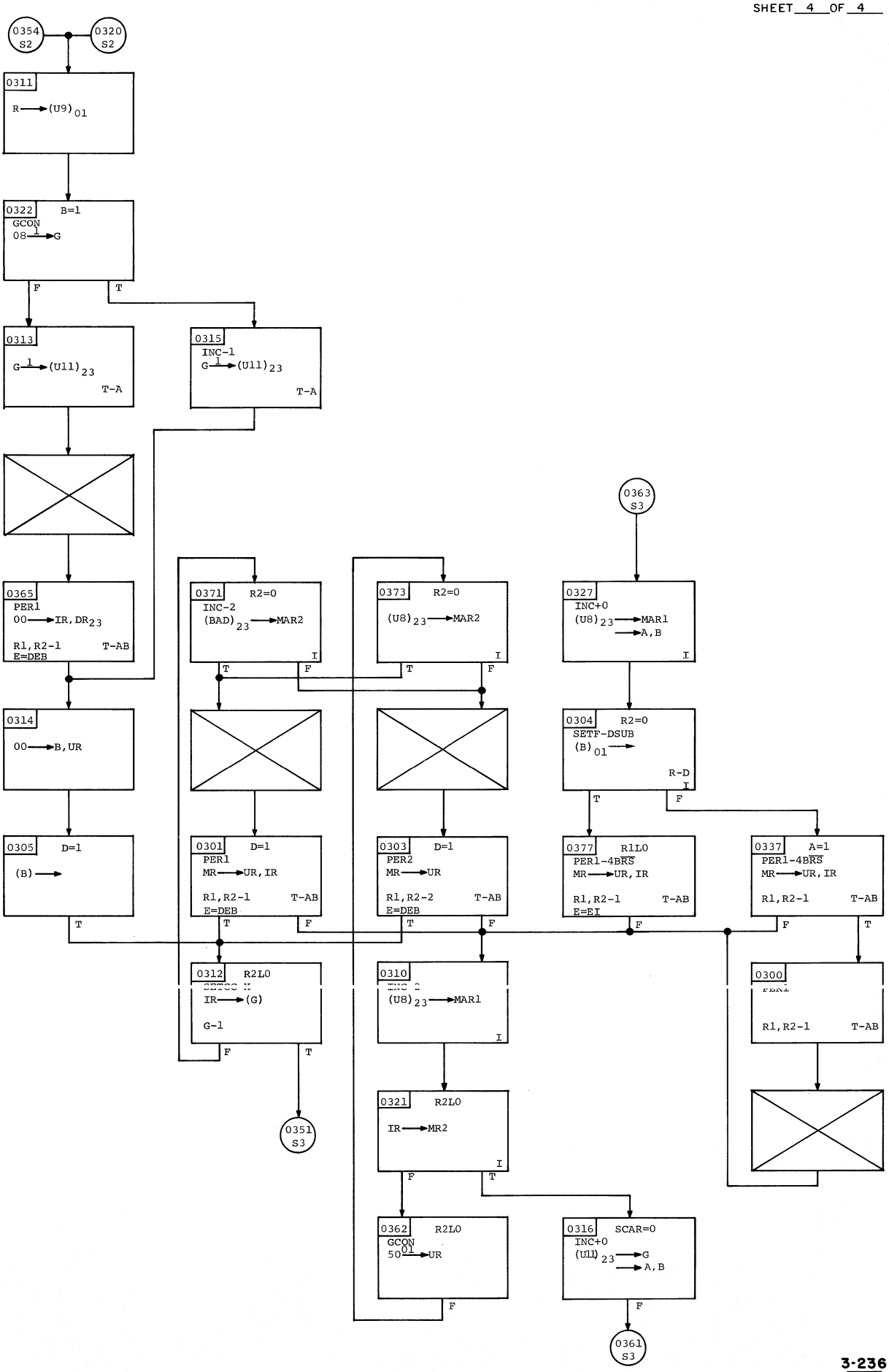
ZERO AND ADD DECIMAL ZAP SS F8



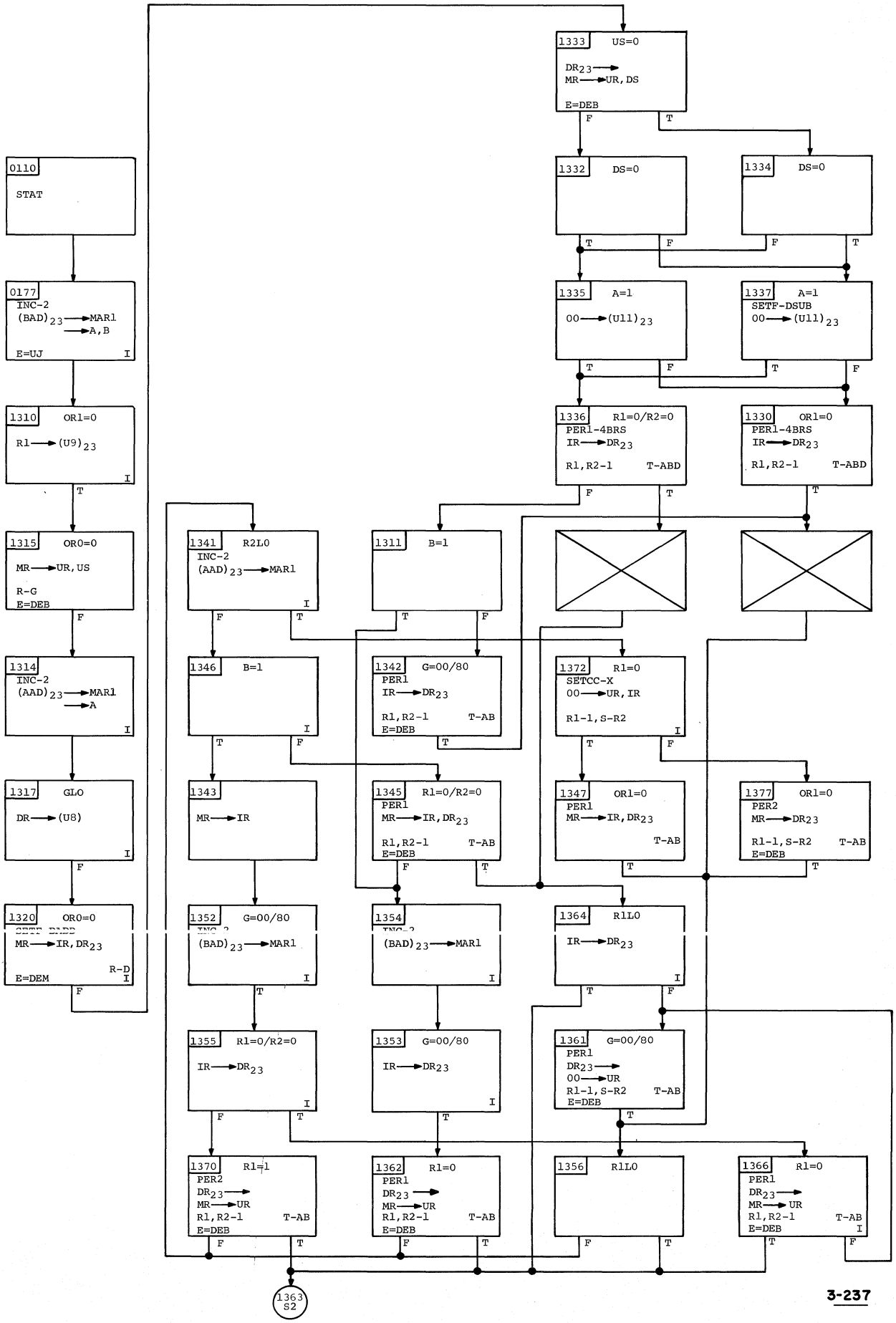
ZERO AND ADD DECIMAL ZAP SS F8



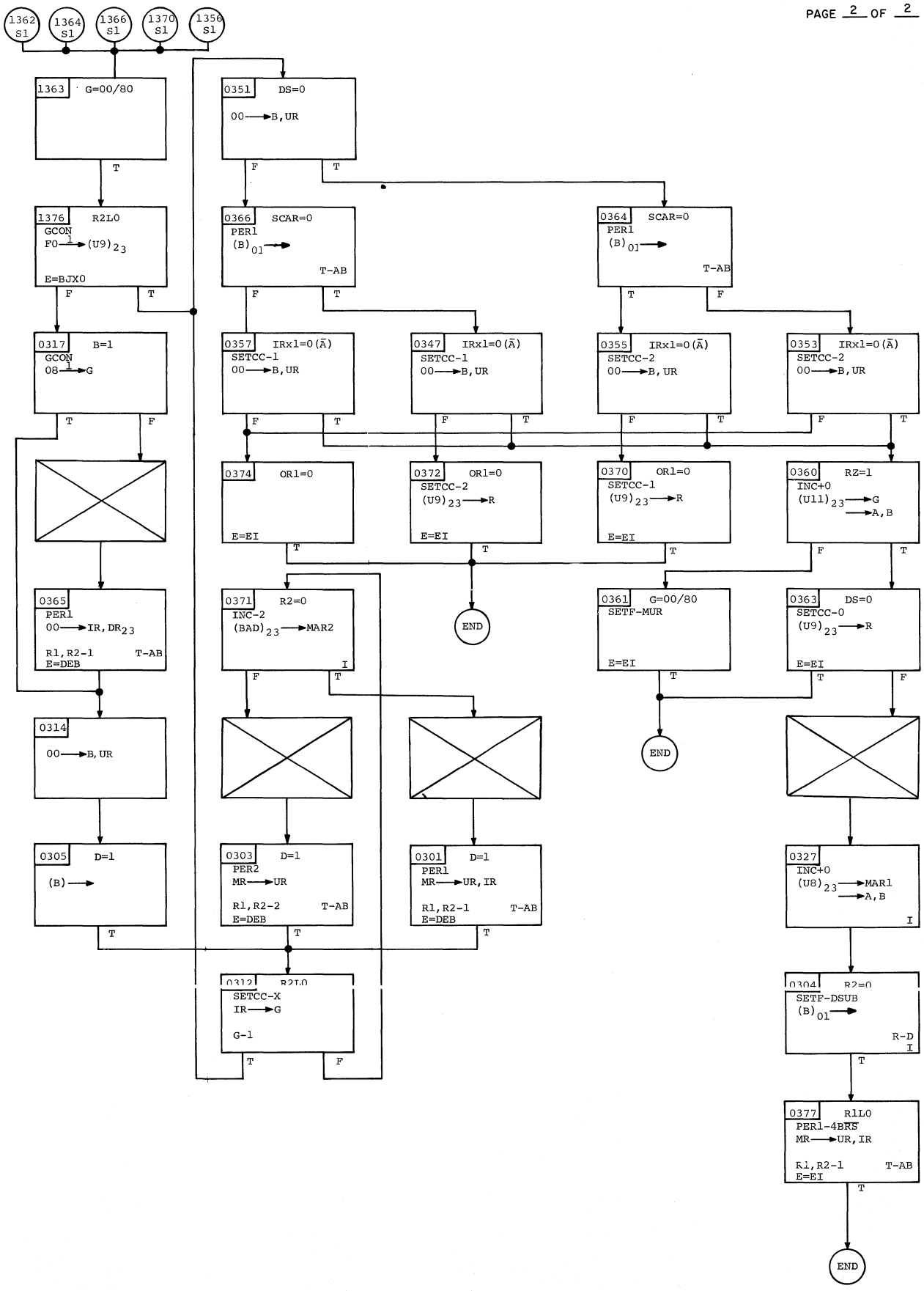
ZERO AND ADD DECIMAL ZAP SS F8



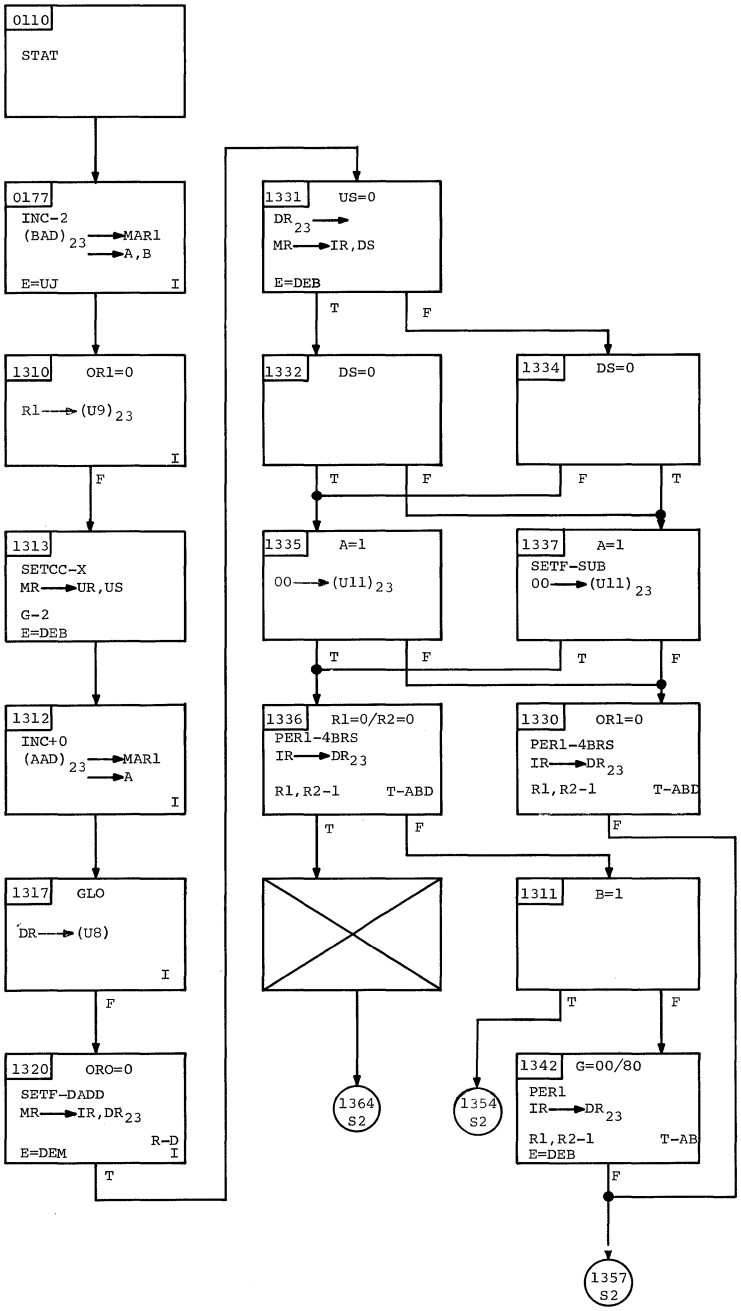
COMPARE DECIMAL CP SS F9

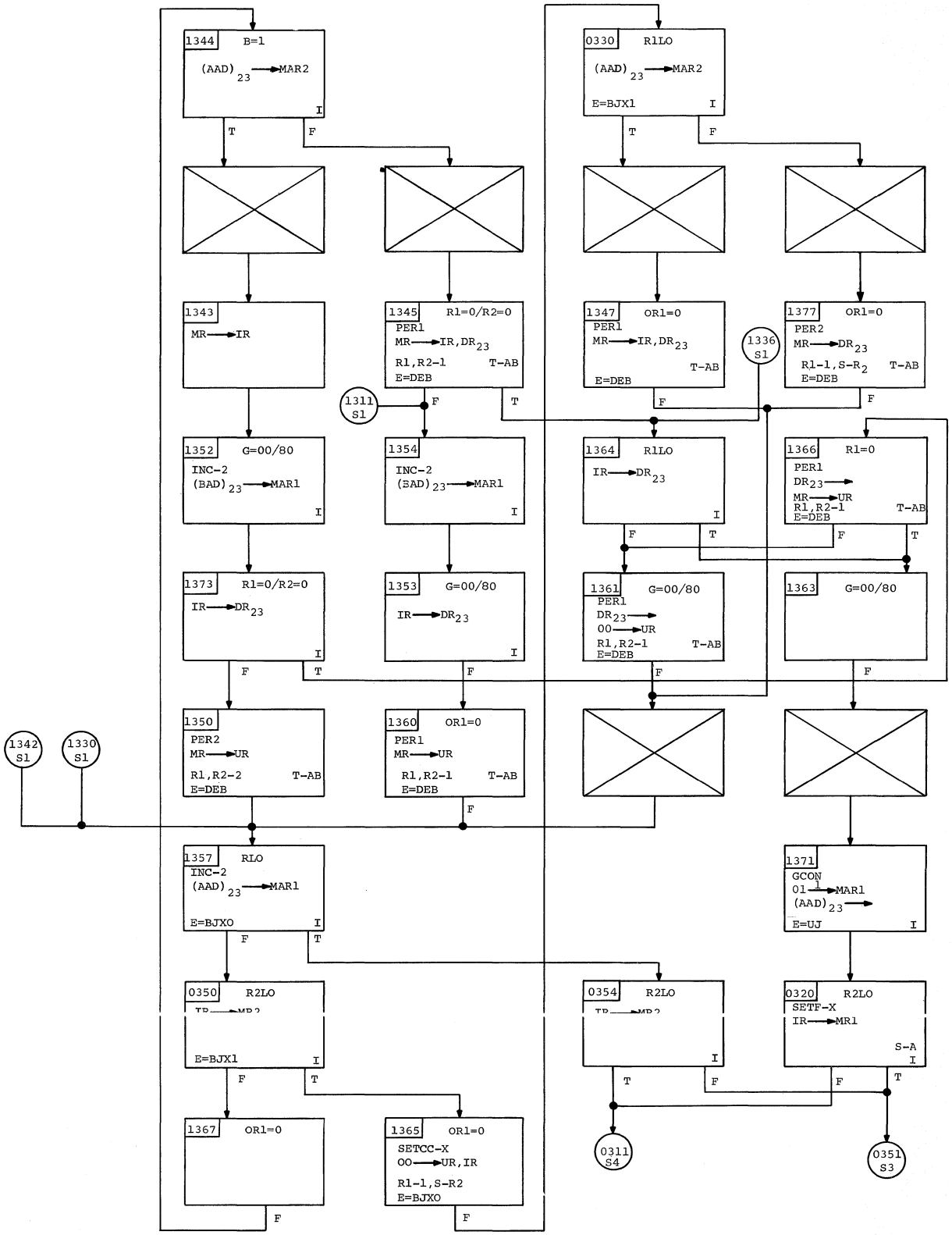


COMPARE DECIMAL CP SS F9

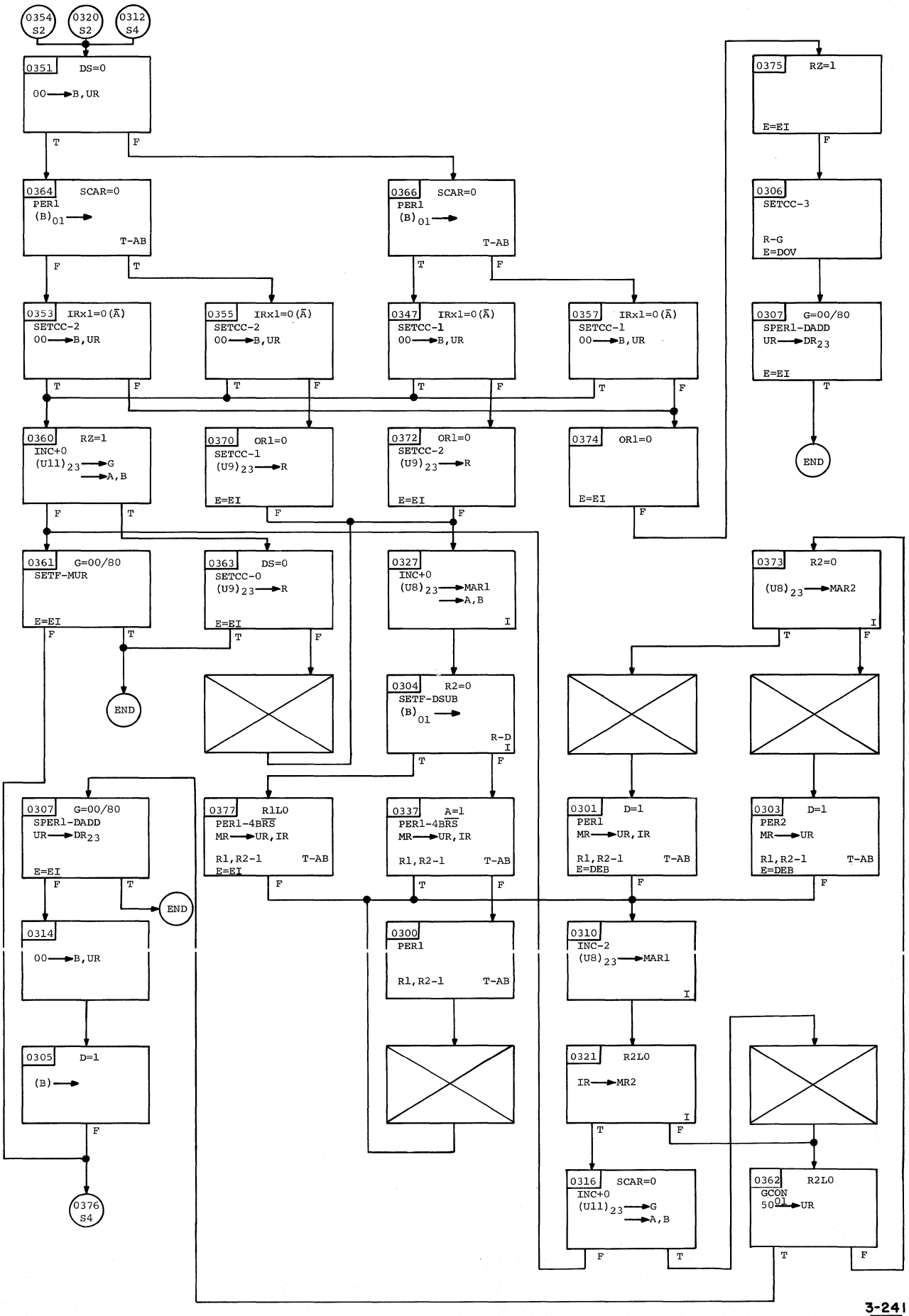


ADD DECIMAL AP SS FA

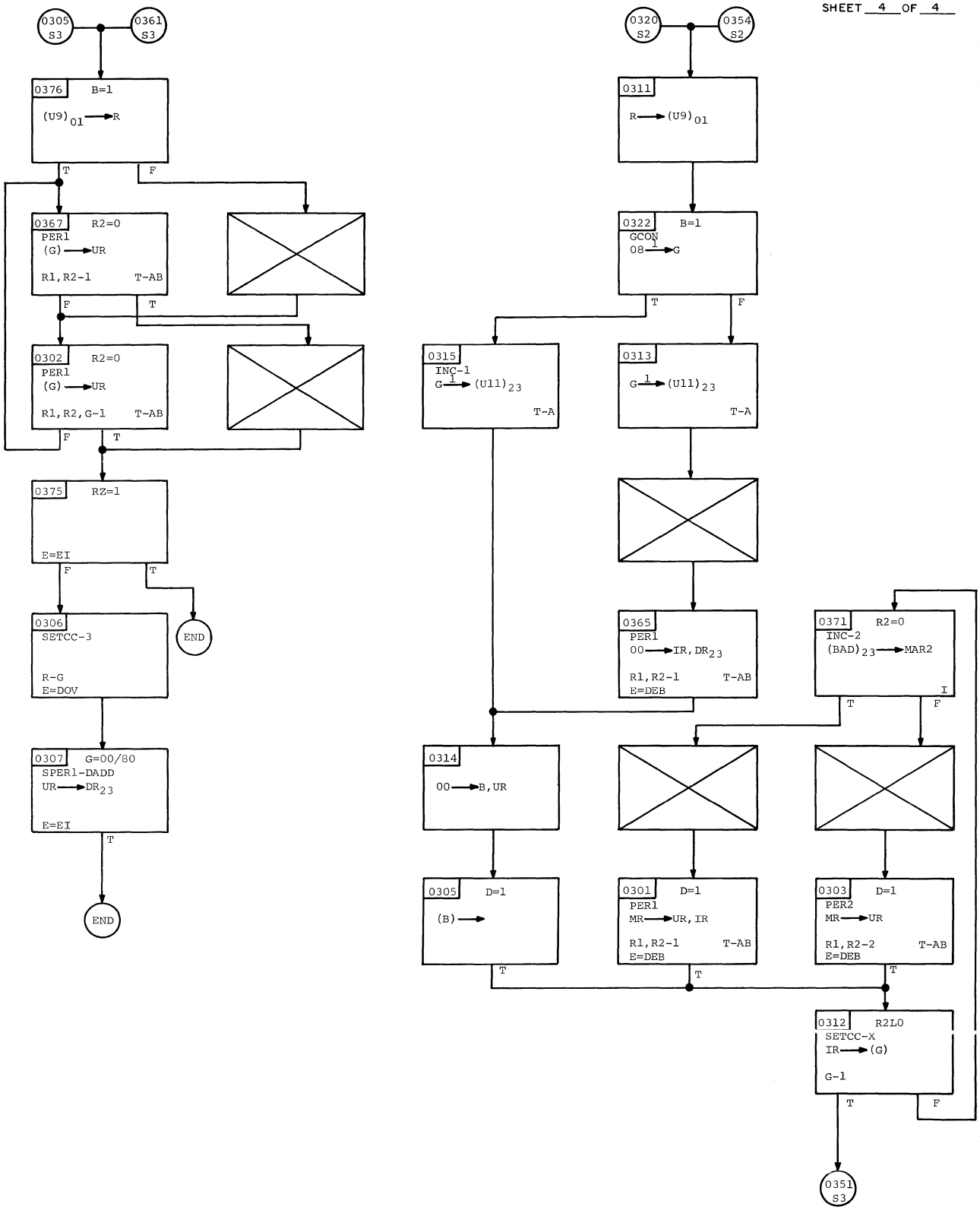



1344 B=1
(AAD)23 → MAR2
I
T
F
1343
MR → IR
1345 R1=0/R2=0
PER1
MR → IR, DR23
R1,R2-1 T-AB
E=DEB
1311 S1
1352 G=00/80
INC-2
(BAD)23 → MAR1
I
1354
INC-2
(BAD)23 → MAR1
I
1373 R1=0/R2=0
IR → DR23
I
1353 G=00/80
IR → DR23
I
1350
PER2
MR → UR
R1,R2-2 T-AB
E=DEB
1360 OR1=0
PER1
MR → UR
R1,R2-1 T-AB
E=DEB
1342 S1
1330 S1
1357 RLO
INC-2
(AAD)23 → MAR1
E=BJXO
I
0350 R2LO
E=BJX1
I
1367 OR1=0
1365 OR1=0
SETCC-X
00 → UR, IR
R1-1, S-R2
E=BJXO
0330 R1LO
(AAD)23 → MAR2
E=BJX1
I
1347 OR1=0
PER1
MR → IR, DR23
T-AB
E=DEB
1336 S1
1364 R1LO
IR → DR23
I
1361 G=00/80
PER1
DR23 →
00 → UR
R1,R2-1 T-AB
E=DEB
0354 R2LO
I
0311 S4
1377 OR1=0
PER2
MR → DR23
R1-1, S-R2 T-AB
E=DEB
1366 R1=0
PER1
DR23 →
MR → UR
R1,R2-1 T-AB
E=DEB
1363 G=00/80
1371
GCON 1 → MAR1
(AAD)23 →
E=UJ
I
0320 R2LO
SETF-X
IR → MR1
S-A
I
0351 S3

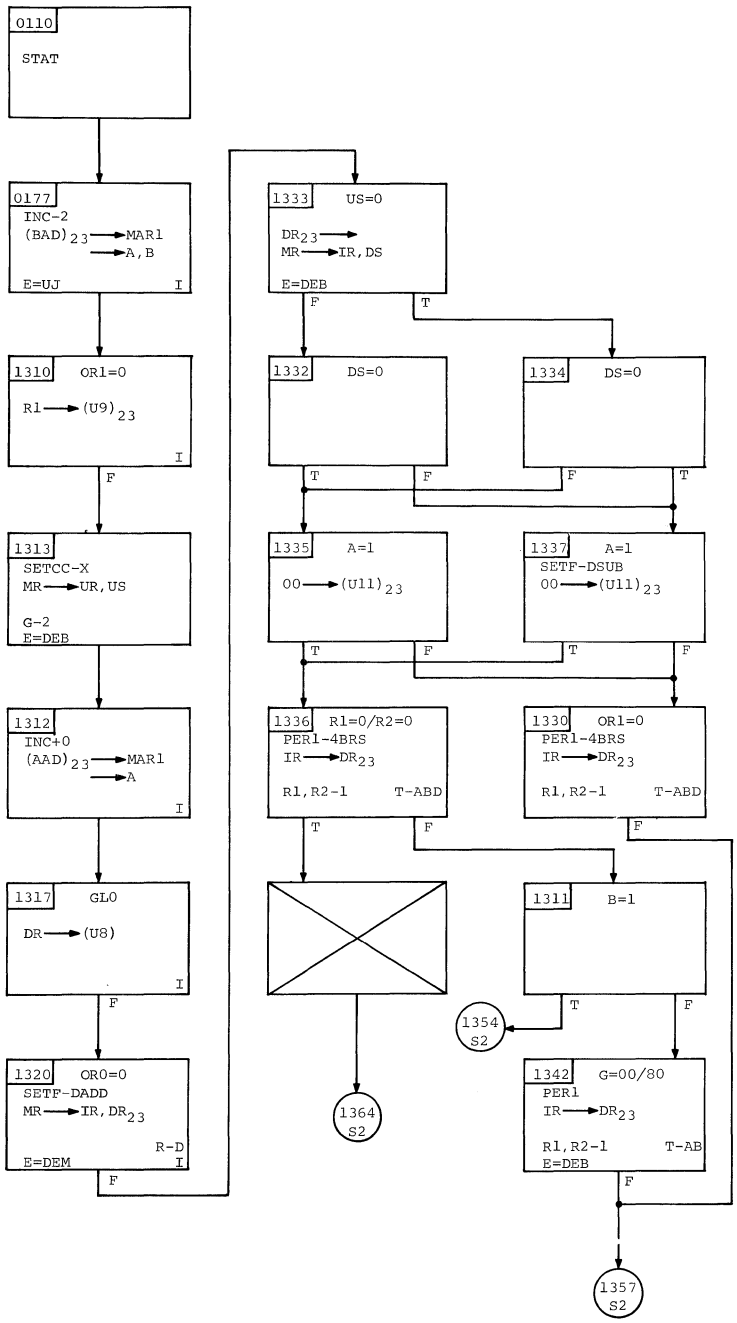
ADD DECIMAL AP SS FA

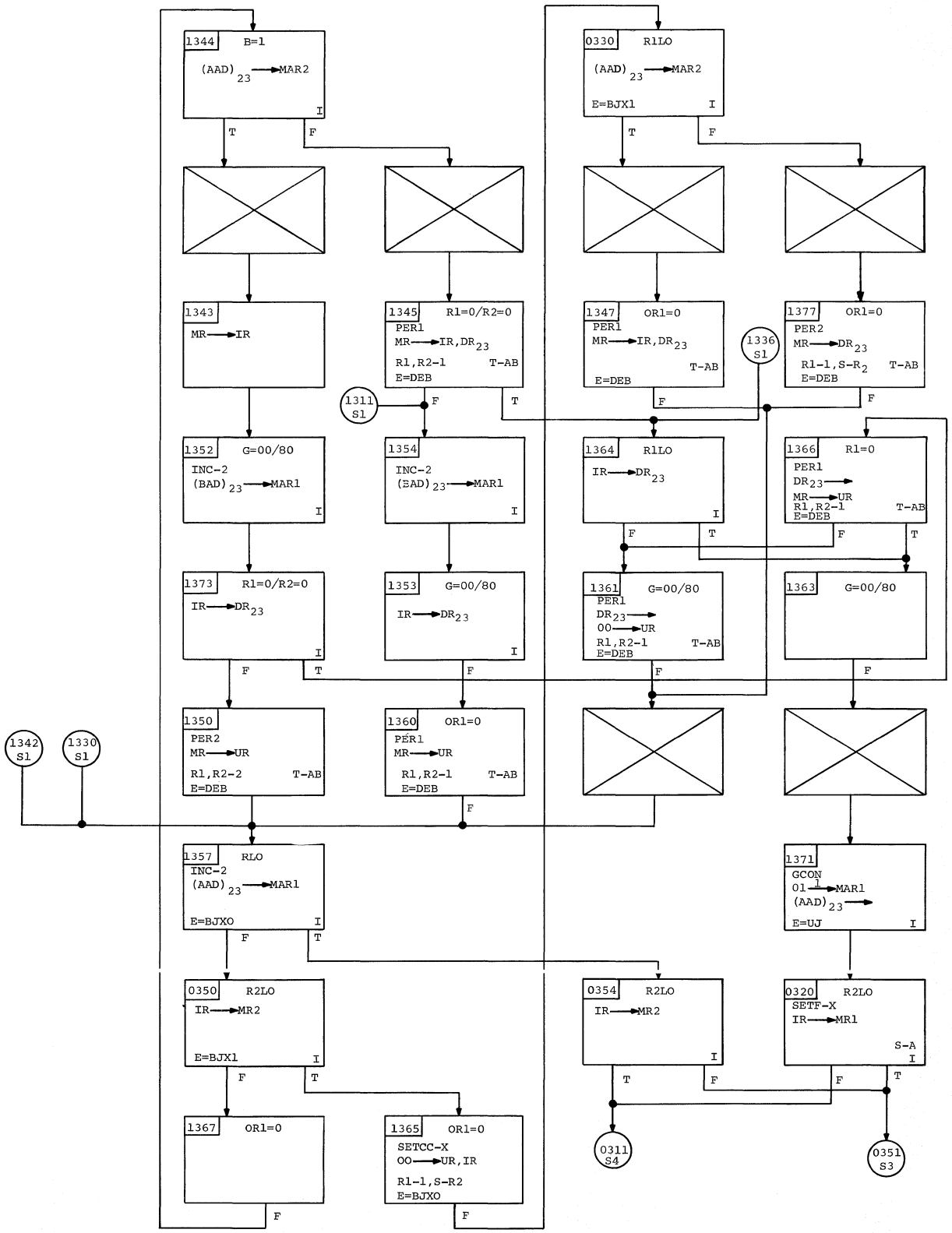


ADD DECIMAL AP SS FA

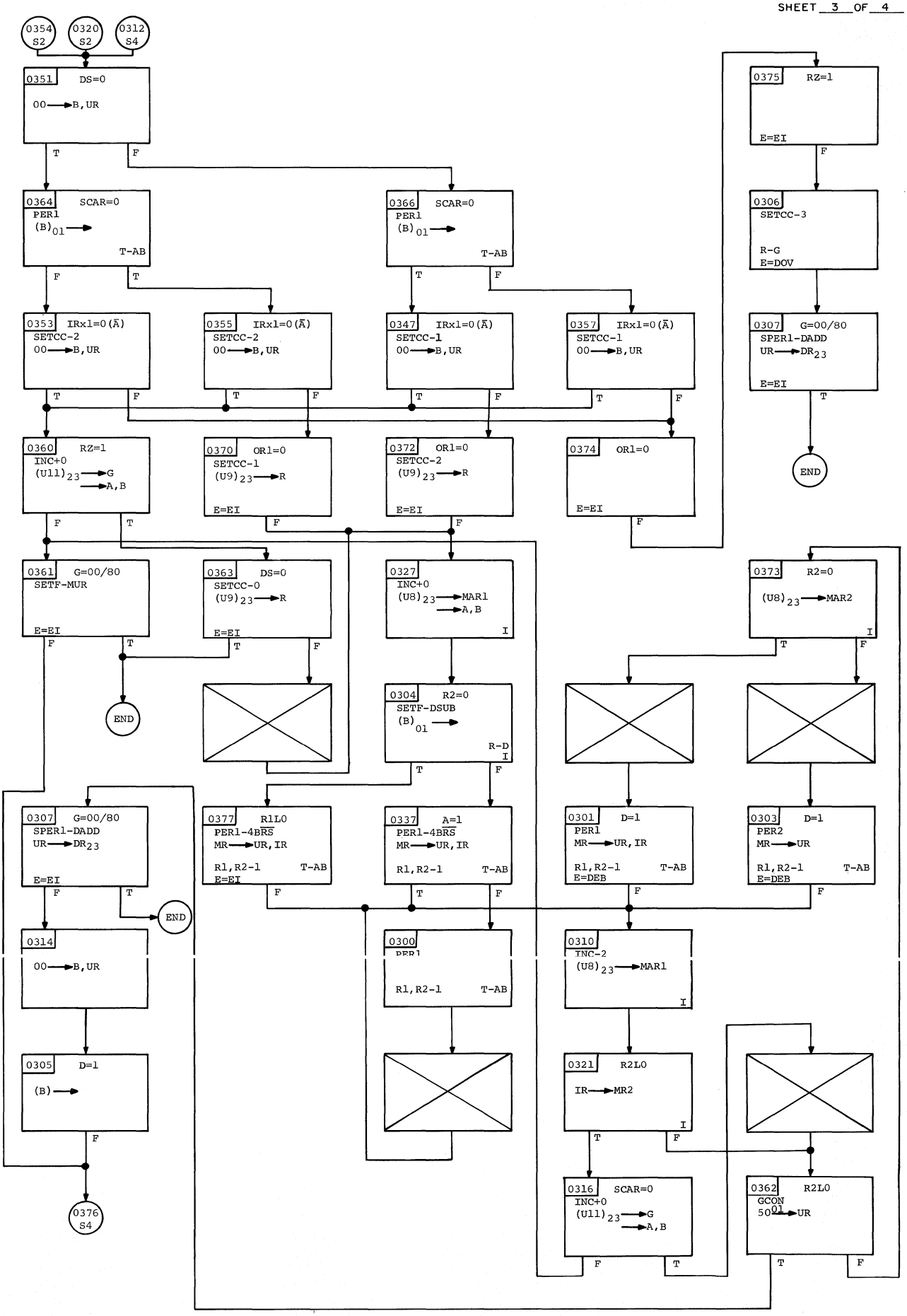


SUBTRACT DECIMAL SP SS FB

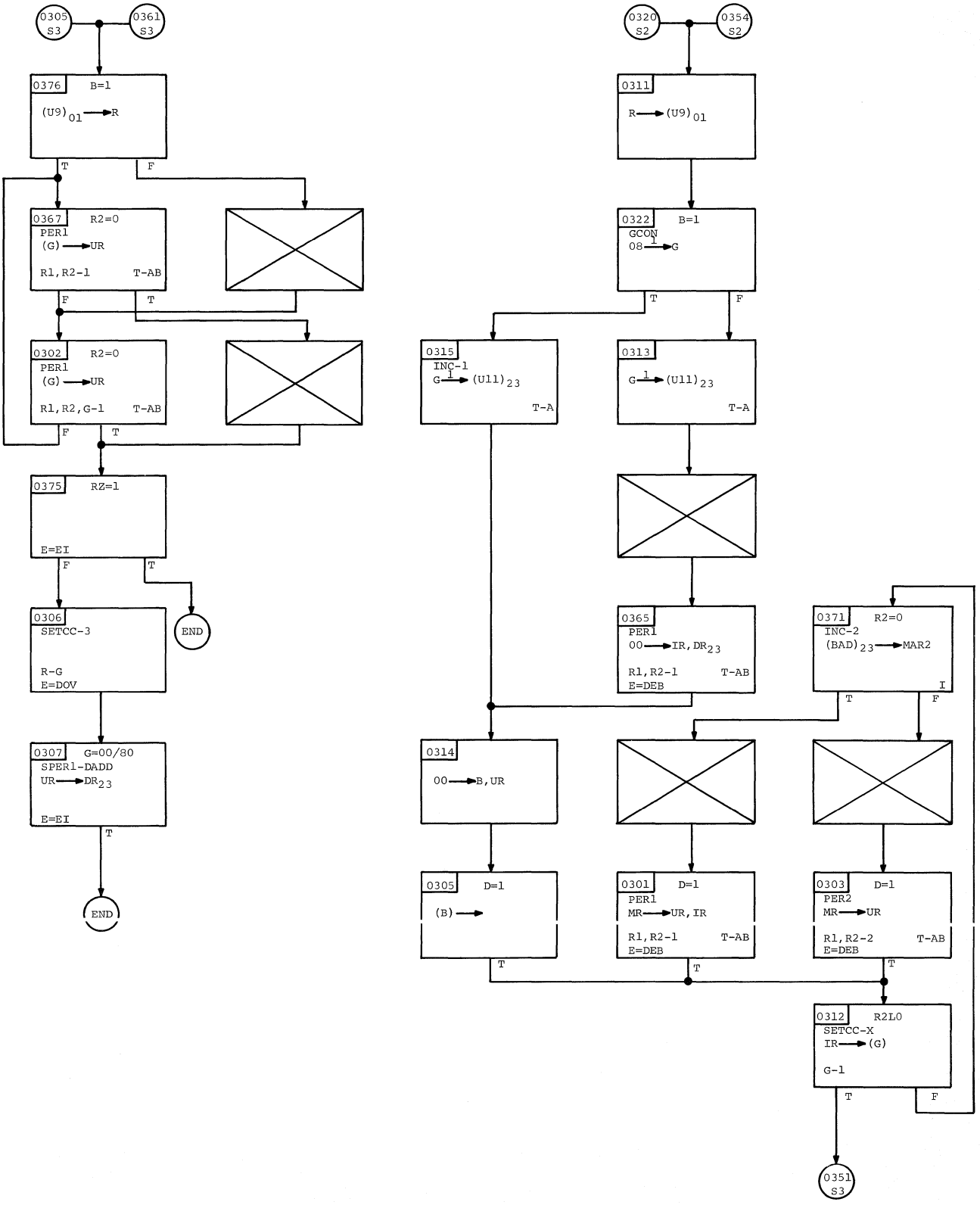



1344 B=1
(AAD)23 → MAR2
I
T
F
1343
MR → IR
1345 R1=0/R2=0
PER1
MR → IR,DR23
R1,R2-1 T-AB
E=DEB
1311 S1
1352 G=00/80
INC-2
(BAD)23 → MAR1
I
1354
INC-2
(BAD)23 → MAR1
I
1373 R1=0/R2=0
IR → DR23
I
1353 G=00/80
IR → DR23
I
1350
PER2
MR → UR
R1,R2-2 T-AB
E=DEB
1360 OR1=0
PER1
MR → UR
R1,R2-1 T-AB
E=DEB
1342 S1
1330 S1
1357 RLO
INC-2
(AAD)23 → MAR1
E=BJXO
I
0350 R2LO
IR → MR2
E=BJXI
I
1367 OR1=0
1365 OR1=0
SETCC-X
00 → UR,IR
R1-1,S-R2
E=BJXO
0330 R1LO
(AAD)23 → MAR2
E=BJX1
I
1347 OR1=0
PER1
MR → IR,DR23
T-AB
E=DEB
1336 S1
1377 OR1=0
PER2
MR → DR23
R1-1,S-R2 T-AB
E=DEB
1364 R1LO
IR → DR23
I
1366 R1=0
PER1
DR23 →
MR → UR
R1,R2-1 T-AB
E=DEB
1361 G=00/80
PER1
DR23 →
00 → UR
R1,R2-1 T-AB
E=DEB
1363 G=00/80
1371
GCON
01 → MAR1
(AAD)23 →
E=UJ
I
0354 R2LO
IR → MR2
I
0320 R2LO
SETF-X
IR → MR1
S-A
I
0311 S4
0351 S3

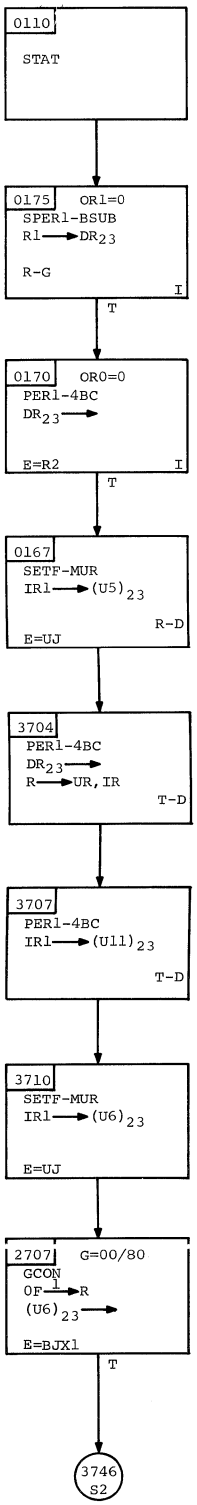
SUBTRACT DECIMAL SP SS FB

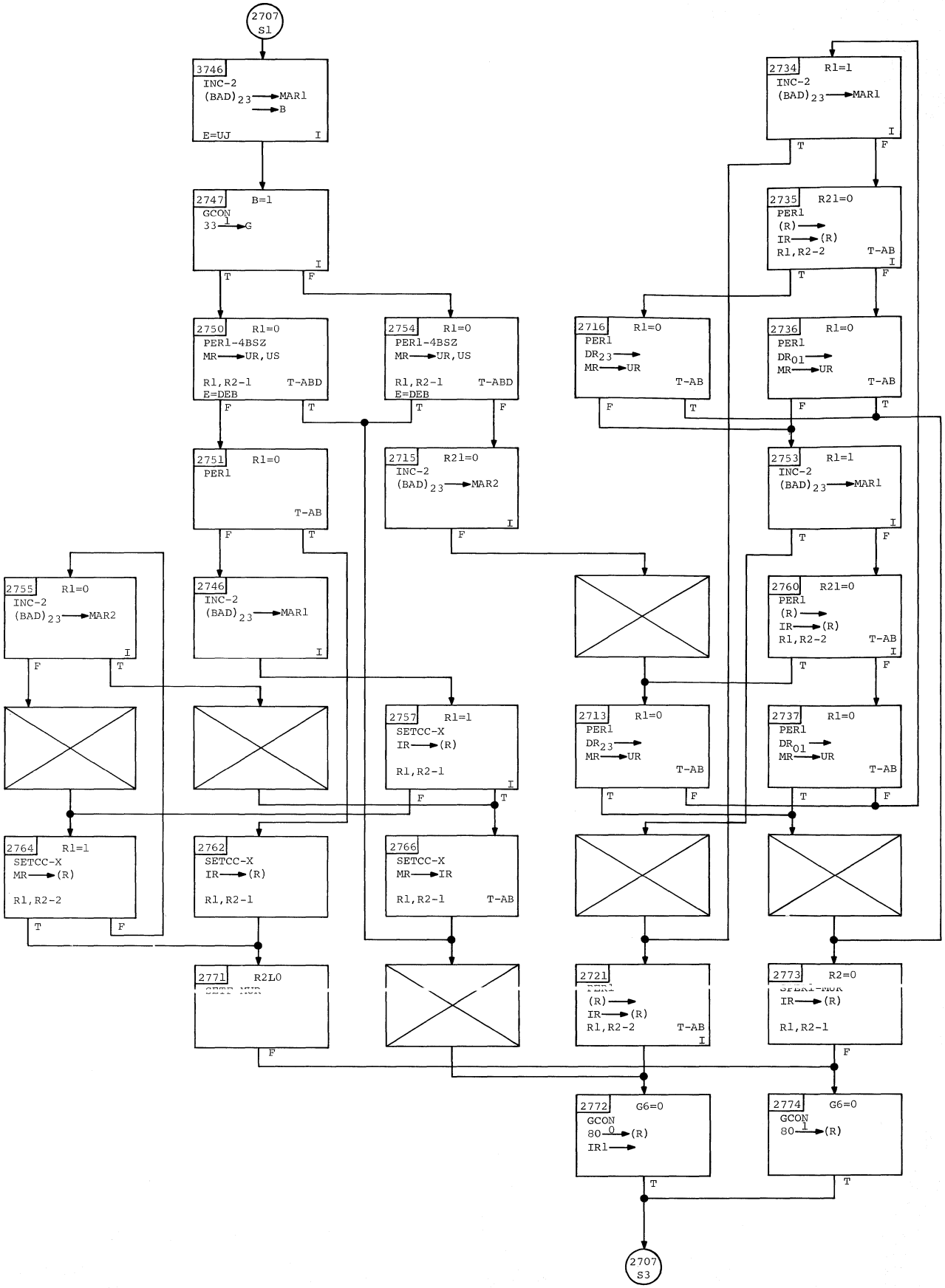


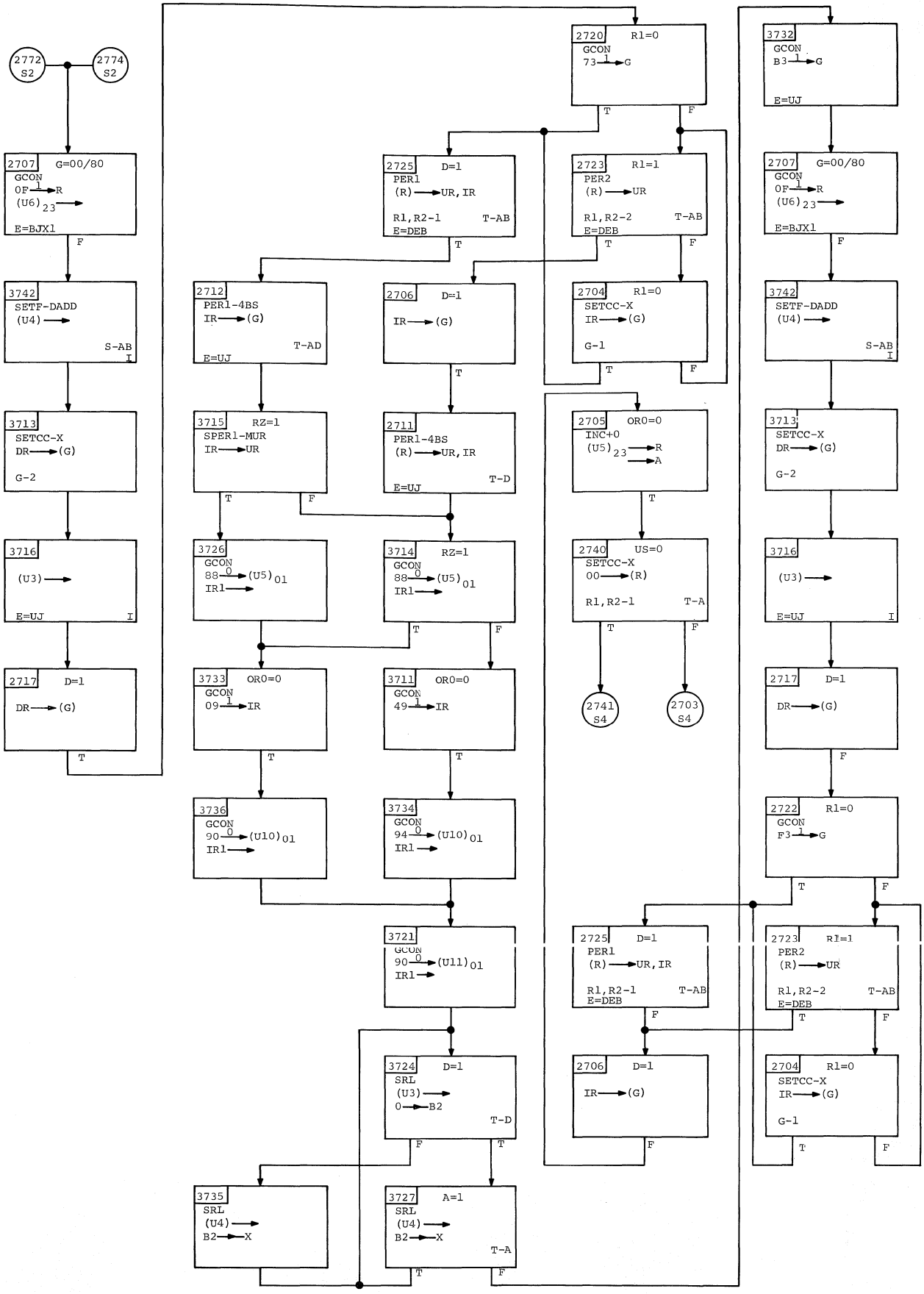
SUBTRACT DECIMAL SP SS FB



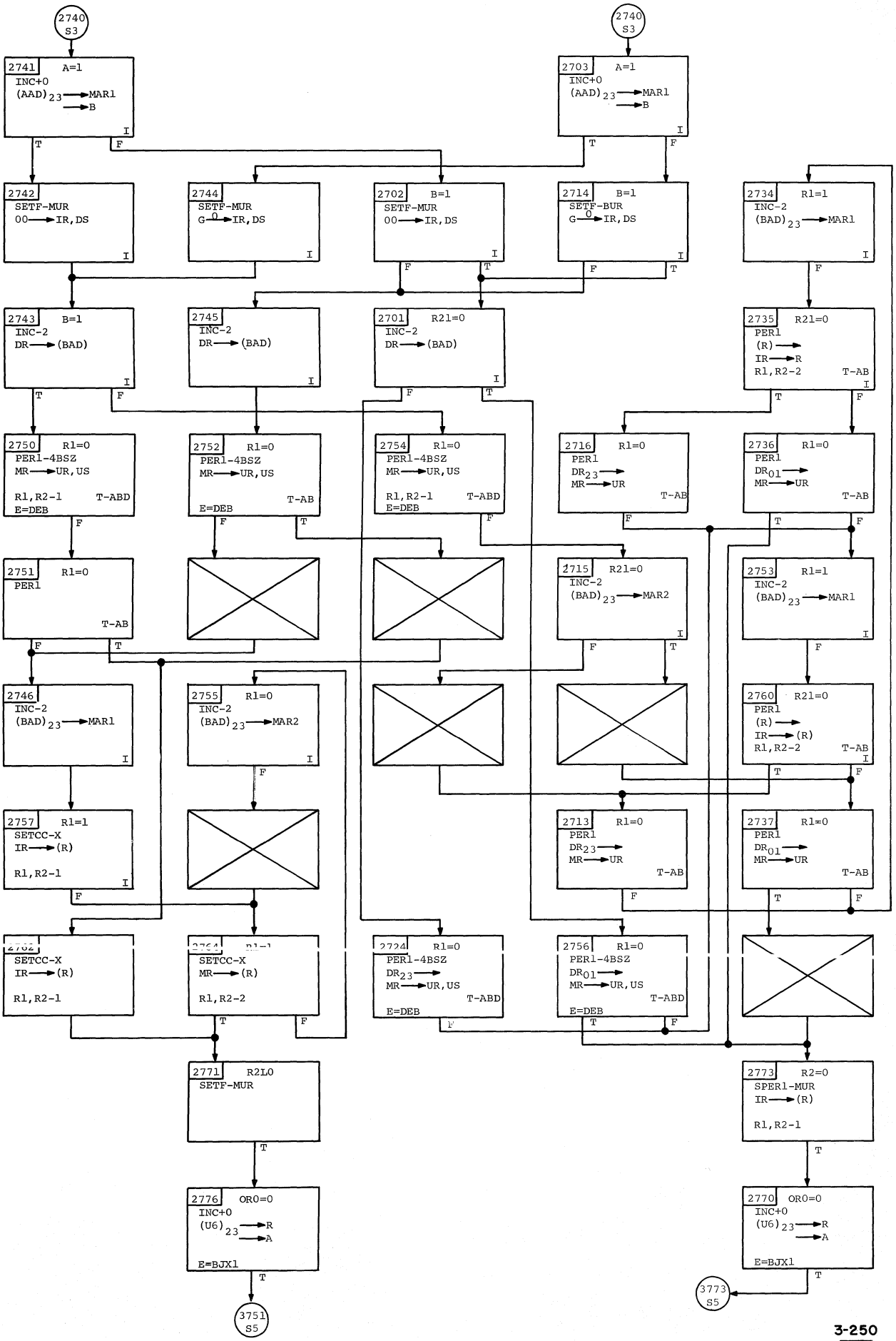
MULTIPLY DECIMAL MP SS FC

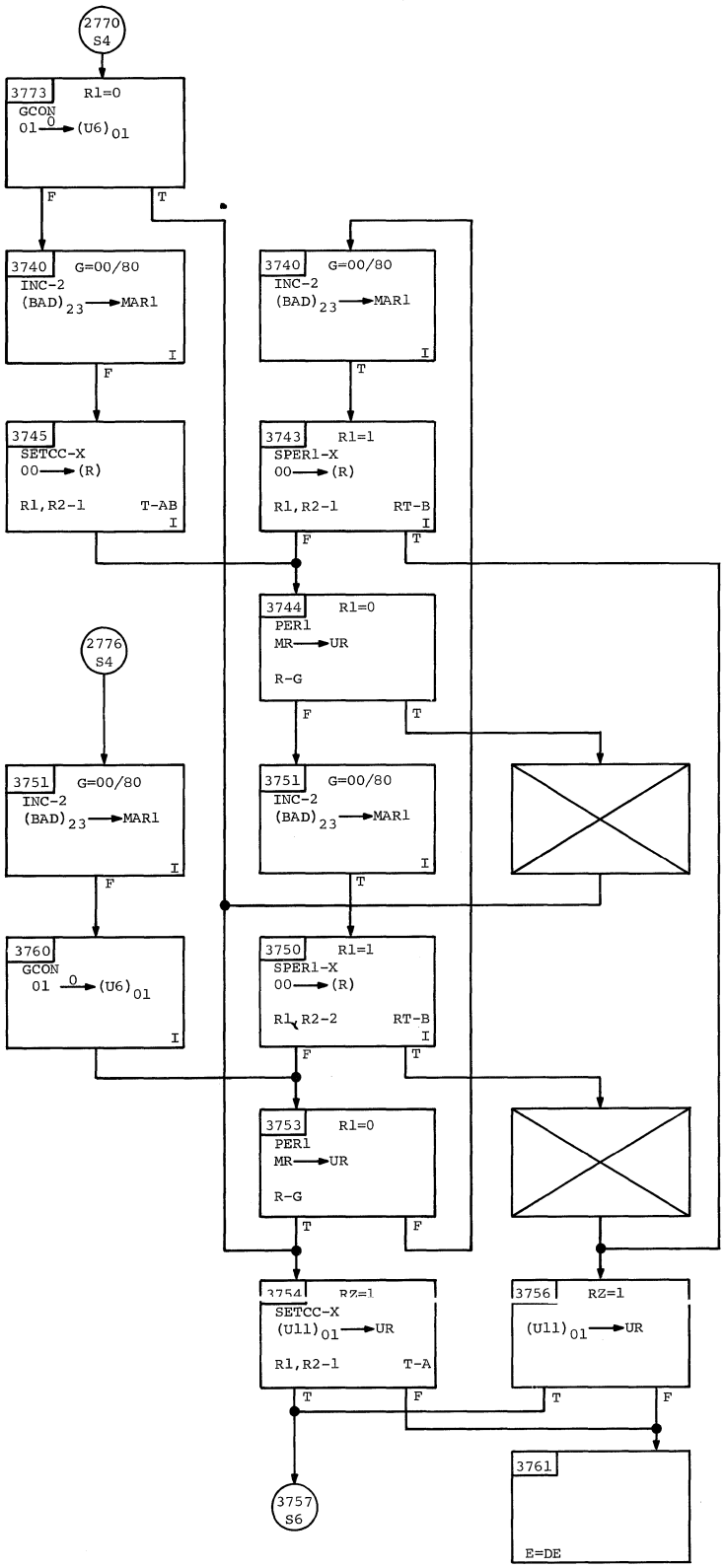


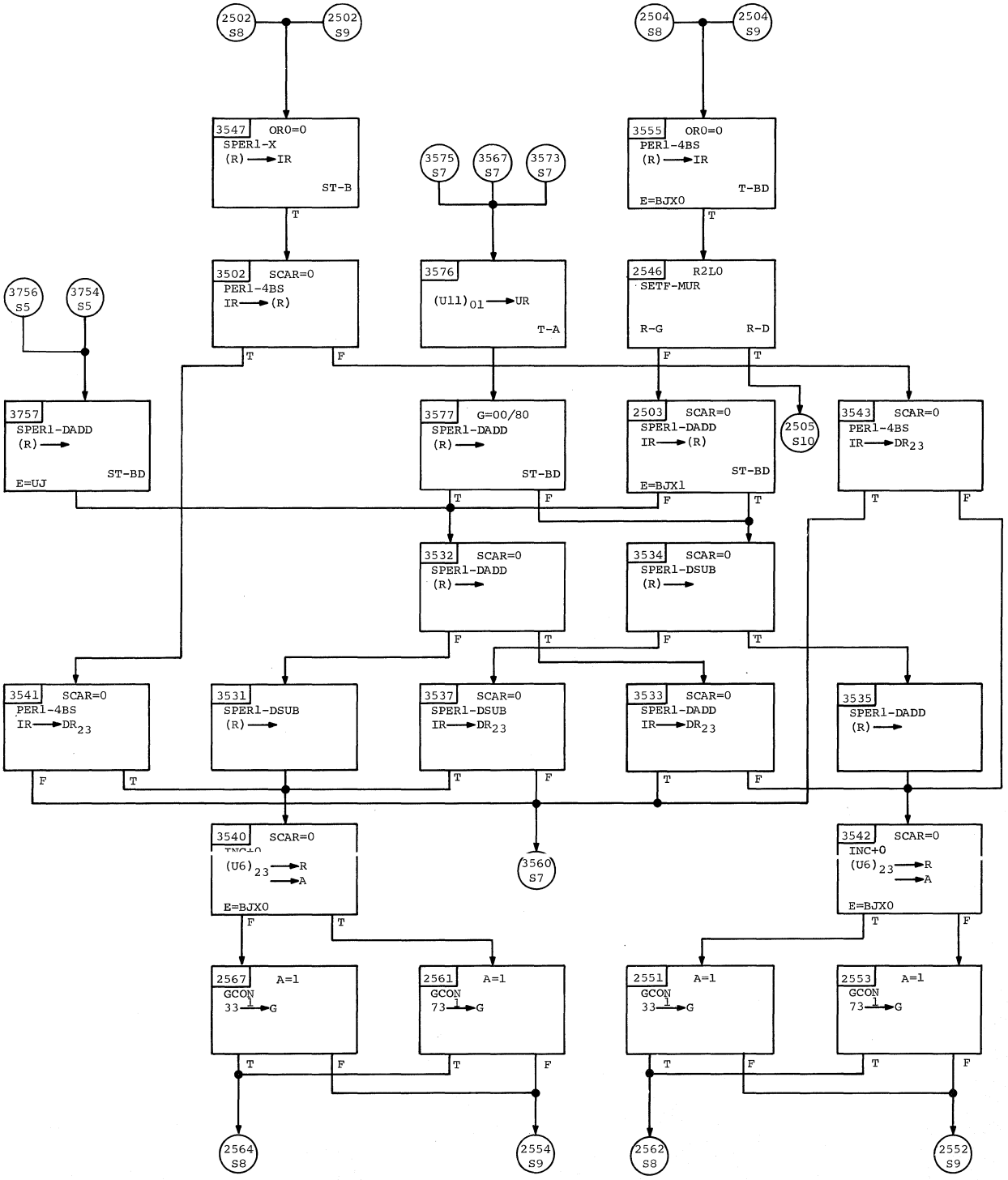

2707 S1
3746 INC-2 (BAD)23 → MAR1 → B E=UJ I
2747 B=1 GCON 33 1 → G I
2750 R1=0 PER1-4BSZ MR → UR,US R1,R2-1 T-ABD E=DEB
2754 R1=0 PER1-4BSZ MR → UR,US R1,R2-1 T-ABD E=DEB
2716 R1=0 PER1 DR23 → MR → UR T-AB
2734 R1=1 INC-2 (BAD)23 → MAR1 I
2735 R21=0 PER1 (R) → IR → (R) R1,R2-2 T-AB I
2736 R1=0 PER1 DR01 → MR → UR T-AB
2751 R1=0 PER1 T-AB
2715 R21=0 INC-2 (BAD)23 → MAR2 I
2753 R1=1 INC-2 (BAD)23 → MAR1 I
2755 R1=0 INC-2 (BAD)23 → MAR2 I
2746 INC-2 (BAD)23 → MAR1 I
2760 R21=0 PER1 (R) → IR → (R) R1,R2-2 T-AB I
2757 R1=1 SETCC-X IR → (R) R1,R2-1 I
2713 R1=0 PER1 DR23 → MR → UR T-AB
2737 R1=0 PER1 DR01 → MR → UR T-AB
2764 R1=1 SETCC-X MR → (R) R1,R2-2
2762 SETCC-X IR → (R) R1,R2-1
2766 SETCC-X MR → IR R1,R2-1 T-AB
2771 R2L0
2721 (R) → IR → (R) R1,R2-2 T-AB I
2773 R2=0 IR → (R) R1,R2-1
2772 G6=0 GCON 80 0 → (R) IR1 →
2774 G6=0 GCON 80 1 → (R)
2707 S3


2772 S2
2774 S2
2707 G=00/80
GCON 0F 1 R
(U6) 23
E=BJX1
F
3742 SETF-DADD
(U4)
S-AB I
3713 SETCC-X
DR (G)
G-2
3716
(U3)
E=UJ I
2717 D=1
DR (G)
T
2720 R1=0
GCON 73 1 G
T F
2725 D=1
PER1
(R) UR, IR
R1, R2-1 T-AB
E=DEB
T
2723 R1=1
PER2
(R) UR
R1, R2-2 T-AB
E=DEB
T F
2712 PER1-4BS
IR (G)
T-AD
E=UJ
2706 D=1
IR (G)
T
2704 R1=0
SETCC-X
IR (G)
G-1
T F
3715 RZ=1
SPER1-MUR
IR UR
T F
2711 PER1-4BS
(R) UR, IR
T-D
E=UJ
2705 OR0=0
INC+0
(U5) 23 R
A
T
3726
GCON 88 0 (U5) 01
IR1
3714 RZ=1
GCON 88 0 (U5) 01
IR1
2740 US=0
SETCC-X
00 (R)
R1, R2-1 T-A
T F
2741 S4
2703 S4
T F
3733 OR0=0
GCON 09 1 IR
3711 OR0=0
GCON 49 1 IR
T
3736
GCON 90 0 (U10) 01
IR1
3734
GCON 94 0 (U10) 01
IR1
3721
GCON 90 0 (U11) 01
IR1
3724 D=1
SRL
(U3)
0 B2
T-D
F T
3735
SRL
(U4)
B2 X
3727 A=1
SRL
(U4)
B2 X
T-A
T F
3732
GCON B3 1 G
E=UJ
2707 G=00/80
GCON 0F 1 R
(U6) 23
E=BJX1
F
3742 SETF-DADD
(U4)
S-AB I
3713 SETCC-X
DR (G)
G-2
3716
(U3)
E=UJ I
2717 D=1
DR (G)
F
2722 R1=0
GCON F3 1 G
T F
2725 D=1
PER1
(R) UR, IR
R1, R2-1 T-AB
E=DEB
F
2706 D=1
IR (G)
F
2723 R1=1
PER2
(R) UR
R1, R2-2 T-AB
E=DEB
T F
2704 R1=0
SETCC-X
IR (G)
G-1
T F

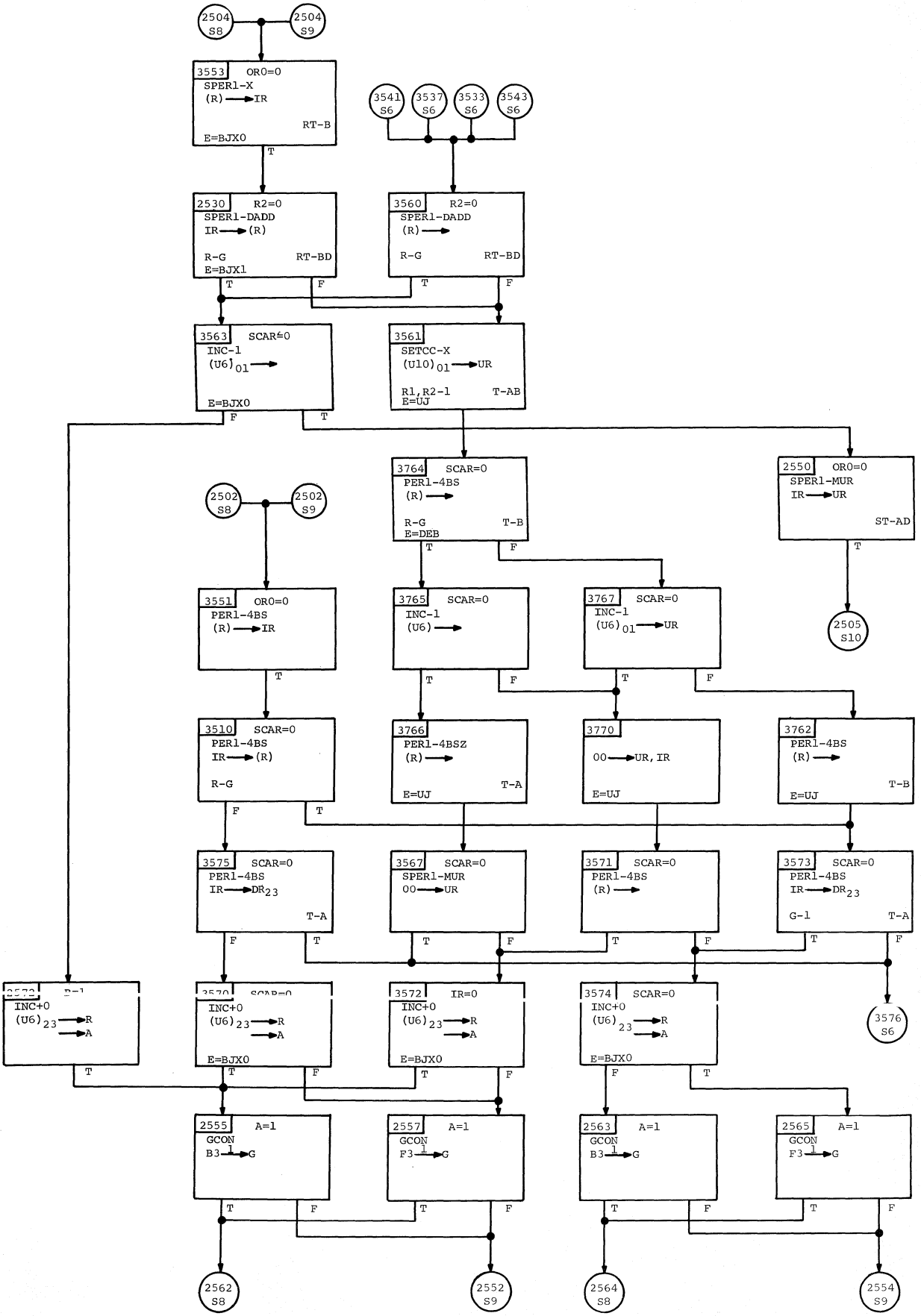
MULTIPLY DECIMAL MP SS FC

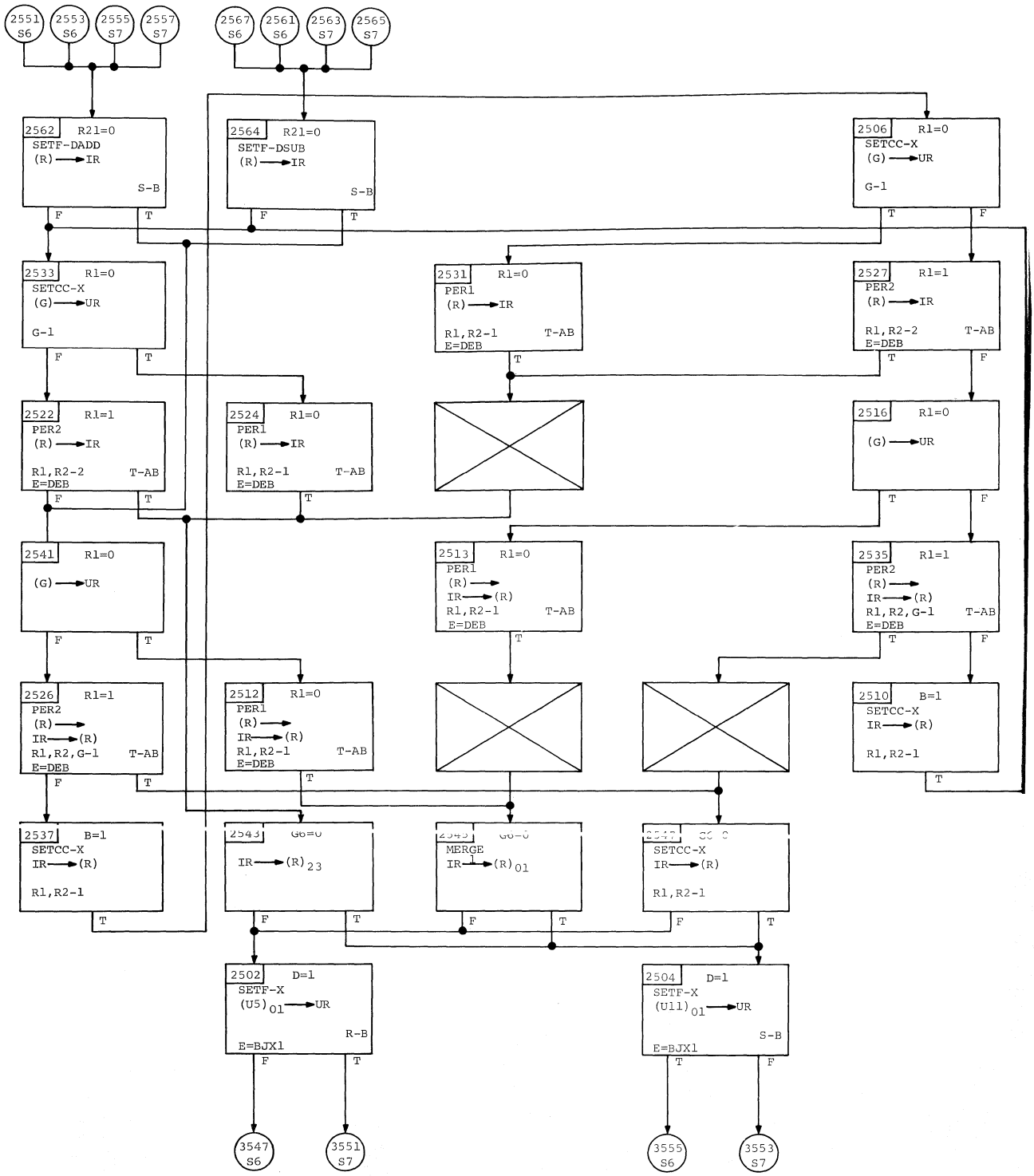


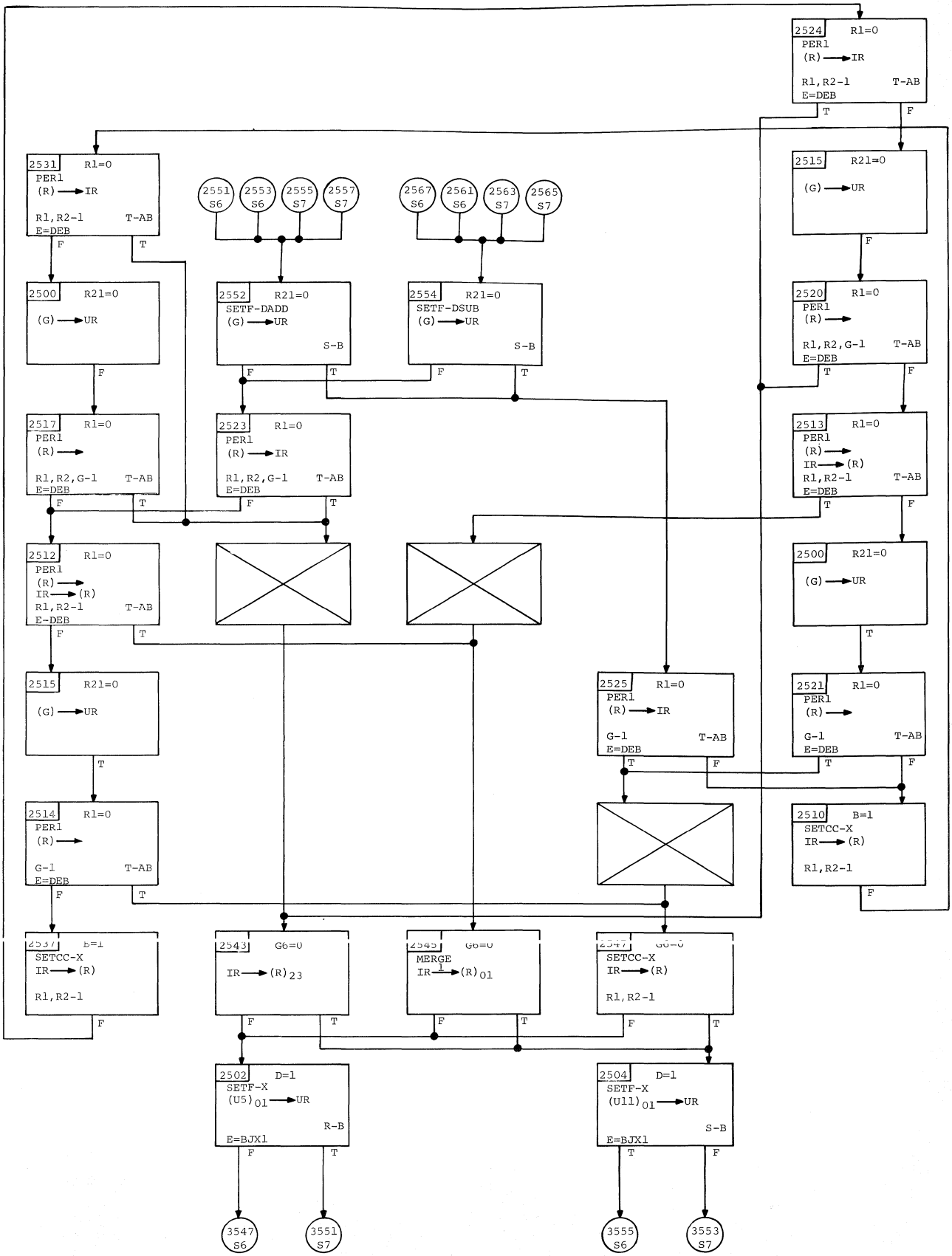

2770 S4
3773 R1=0
GCON 0 1 → (U6) 01
F
T
3740 G=00/80
INC-2
(BAD) 23 → MAR1
I
F
3745
SETCC-X
00 → (R)
R1, R2-1 T-AB
I
3740 G=00/80
INC-2
(BAD) 23 → MAR1
I
T
3743 R1=1
SPER1-X
00 → (R)
R1, R2-1 RT-B
I
F
T
3744 R1=0
PER1
MR → UR
R-G
F
T
2776 S4
3751 G=00/80
INC-2
(BAD) 23 → MAR1
I
F
3760
GCON
01 0 → (U6) 01
I
3751 G=00/80
INC-2
(BAD) 23 → MAR1
I
T
3750 R1=1
SPER1-X
00 → (R)
R1, R2-2 RT-B
I
F
T
3753 R1=0
PER1
MR → UR
R-G
T
F
3754 RZ=1
SETCC-X
(U11) 01 → UR
R1, R2-1 T-A
T
F
3756 RZ=1
(U11) 01 → UR
T
F
3757 S6
3761
E=DE


2502 S8
2502 S9
2504 S8
2504 S9
3547 OR0=0 SPER1-X (R)→IR ST-B
3555 OR0=0 PER1-4BS (R)→IR T-BD E=BJX0
3575 S7
3567 S7
3573 S7
3756 S5
3754 S5
3502 SCAR=0 PER1-4BS IR→(R)
3576 (U11)01→UR T-A
2546 R2L0 SETF-MUR R-G R-D
3757 SPER1-DADD (R)→ ST-BD E=UJ
3577 G=00/80 SPER1-DADD (R)→ ST-BD
2503 SCAR=0 SPER1-DADD IR→(R) ST-BD E=BJX1
2505 S10
3543 SCAR=0 PER1-4BS IR→DR23
3532 SCAR=0 SPER1-DADD (R)→
3534 SCAR=0 SPER1-DSUB (R)→
3541 SCAR=0 PER1-4BS IR→DR23
3531 SPER1-DSUB (R)→
3537 SCAR=0 SPER1-DSUB IR→DR23
3533 SCAR=0 SPER1-DADD IR→DR23
3535 SPER1-DADD (R)→
3540 SCAR=0 (U6)23→R →A E=BJX0
3560 S7
3542 SCAR=0 INC+0 (U6)23→R →A E=BJX0
2567 A=1 GCON 33 1→G
2561 A=1 GCON 73 1→G
2551 A=1 GCON 33 1→G
2553 A=1 GCON 73 1→G
2564 S8
2554 S9
2562 S8
2552 S9

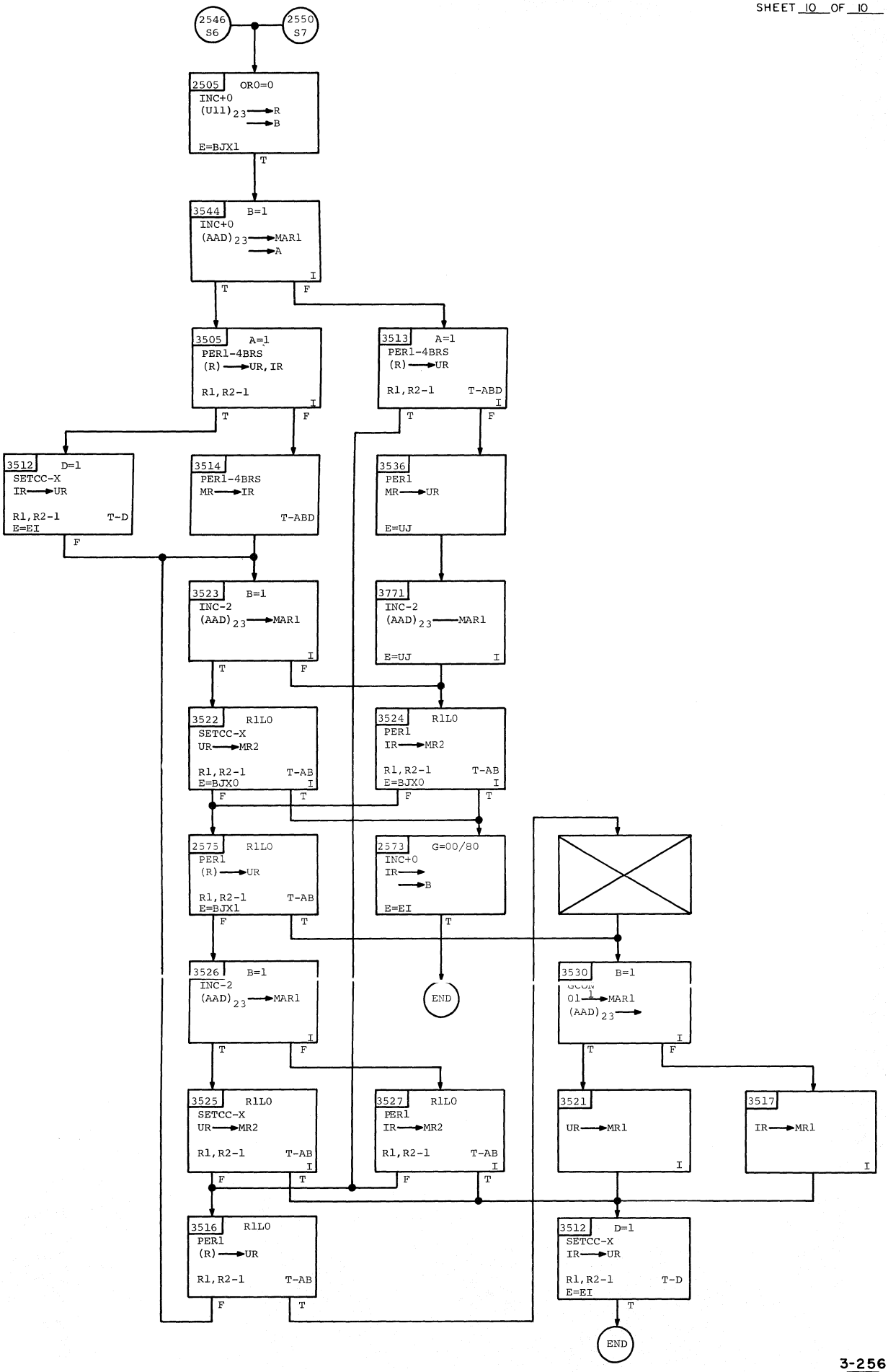
MULTIPLY DECIMAL MP SS FC



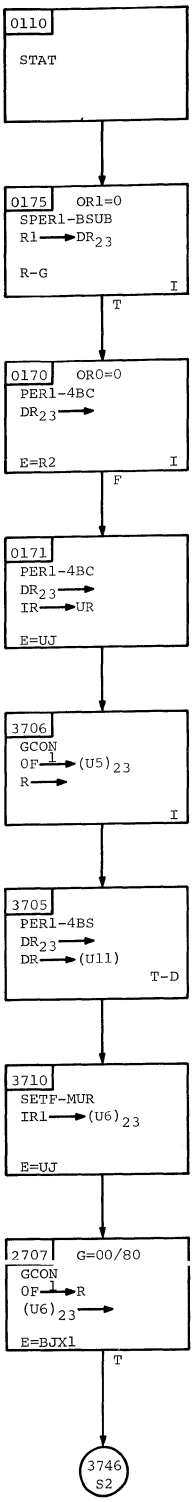

2551 S6
2553 S6
2555 S7
2557 S7
2567 S6
2561 S6
2563 S7
2565 S7
2562 R2l=0 SETF-DADD (R)→IR S-B
2564 R2l=0 SETF-DSUB (R)→IR S-B
2506 Rl=0 SETCC-X (G)→UR G-l
2533 Rl=0 SETCC-X (G)→UR G-l
2531 Rl=0 PER1 (R)→IR Rl,R2-l T-AB E=DEB
2527 Rl=1 PER2 (R)→IR Rl,R2-2 T-AB E=DEB
2522 Rl=1 PER2 (R)→IR Rl,R2-2 T-AB E=DEB
2524 Rl=0 PER1 (R)→IR Rl,R2-l T-AB E=DEB
2516 Rl=0 (G)→UR
2541 Rl=0 (G)→UR
2513 Rl=0 PER1 (R)→ IR→(R) Rl,R2-l T-AB E=DEB
2535 Rl=1 PER2 (R)→ IR→(R) Rl,R2,G-l T-AB E=DEB
2526 Rl=1 PER2 (R)→ IR→(R) Rl,R2,G-l T-AB E=DEB
2512 Rl=0 PER1 (R)→ IR→(R) Rl,R2-l T-AB E=DEB
2510 B=1 SETCC-X IR→(R) Rl,R2-l
2537 B=1 SETCC-X IR→(R) Rl,R2-l
2543 G0=0 IR→(R)23
MERGE IR→(R)01
SETCC-X IR→(R) Rl,R2-l
2502 D=1 SETF-X (U5)01→UR R-B E=BJX1
2504 D=1 SETF-X (Ull)01→UR S-B E=BJX1
3547 S6
3551 S7
3555 S6
3553 S7


2524 R1=0
PER1
(R) → IR
R1,R2-1 T-AB
E=DEB
2531 R1=0
PER1
(R) → IR
R1,R2-1 T-AB
E=DEB
2551 S6
2553 S6
2555 S7
2557 S7
2567 S6
2561 S6
2563 S7
2565 S7
2515 R2l=0
(G) → UR
2500 R2l=0
(G) → UR
2552 R2l=0
SETF-DADD
(G) → UR
S-B
2554 R2l=0
SETF-DSUB
(G) → UR
S-B
2520 R1=0
PER1
(R) →
R1,R2,G-1 T-AB
E=DEB
2517 R1=0
PER1
(R) →
R1,R2,G-1 T-AB
E=DEB
2523 R1=0
PER1
(R) → IR
R1,R2,G-1 T-AB
E=DEB
2513 R1=0
PER1
(R) →
IR → (R)
R1,R2-1 T-AB
E=DEB
2512 R1=0
PER1
(R) →
IR → (R)
R1,R2-1 T-AB
E-DEB
2500 R2l=0
(G) → UR
2515 R2l=0
(G) → UR
2525 R1=0
PER1
(R) → IR
G-1 T-AB
E=DEB
2521 R1=0
PER1
(R) →
G-1 T-AB
E=DEB
2514 R1=0
PER1
(R) →
G-1 T-AB
E=DEB
2510 B=1
SETCC-X
IR → (R)
R1,R2-1
2537 B=1
SETCC-X
IR → (R)
R1,R2-1
2543 G6=0
IR → (R)23
2545 G6=0
MERGE
IR → (R)01
2547 G6=0
SETCC-X
IR → (R)
R1,R2-1
2502 D=1
SETF-X
(U5)01 → UR
R-B
E=BJX1
2504 D=1
SETF-X
(U11)01 → UR
S-B
E=BJX1
3547 S6
3551 S7
3555 S6
3553 S7

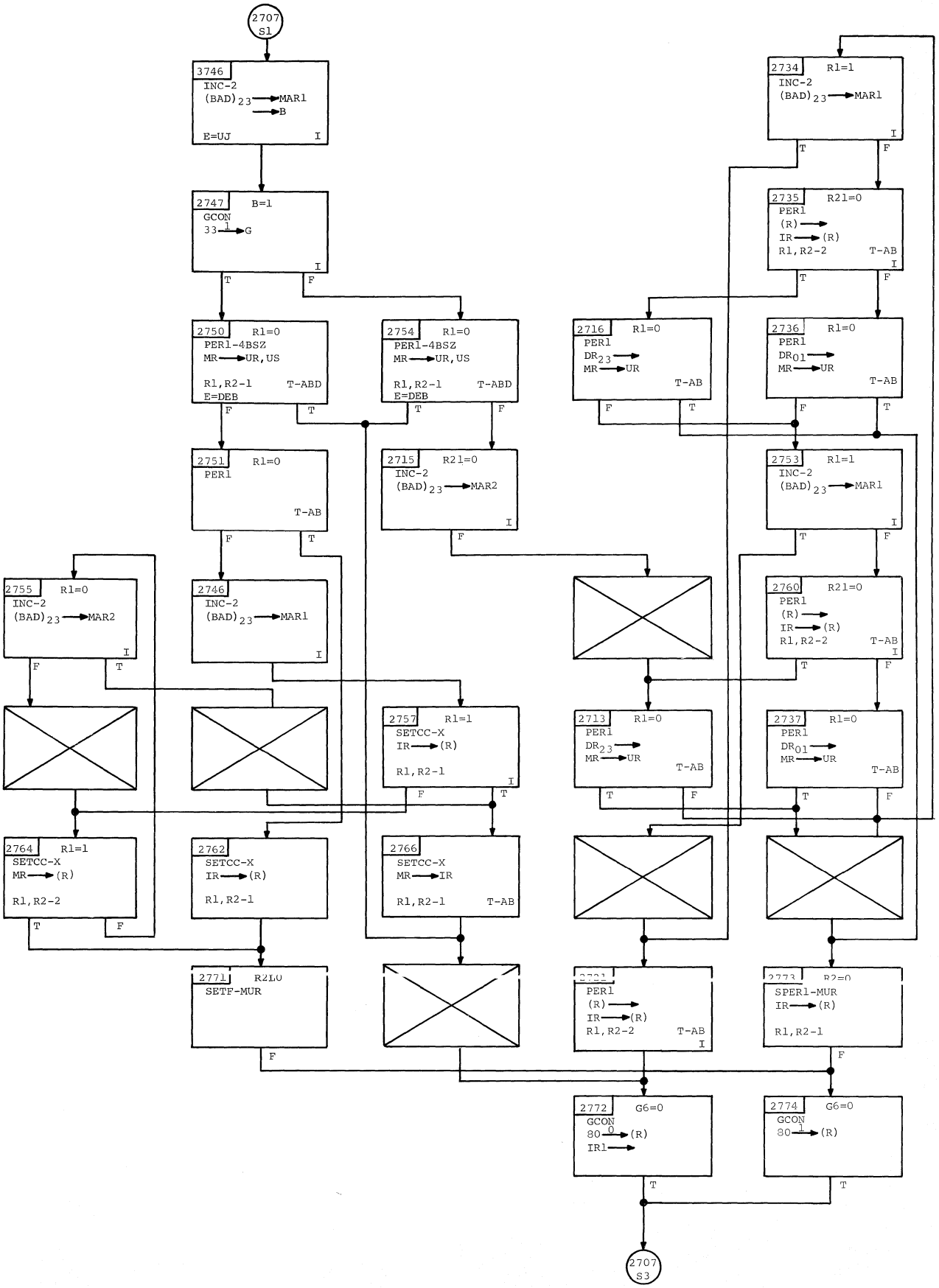
MULTIPLY DECIMAL MP SS FC



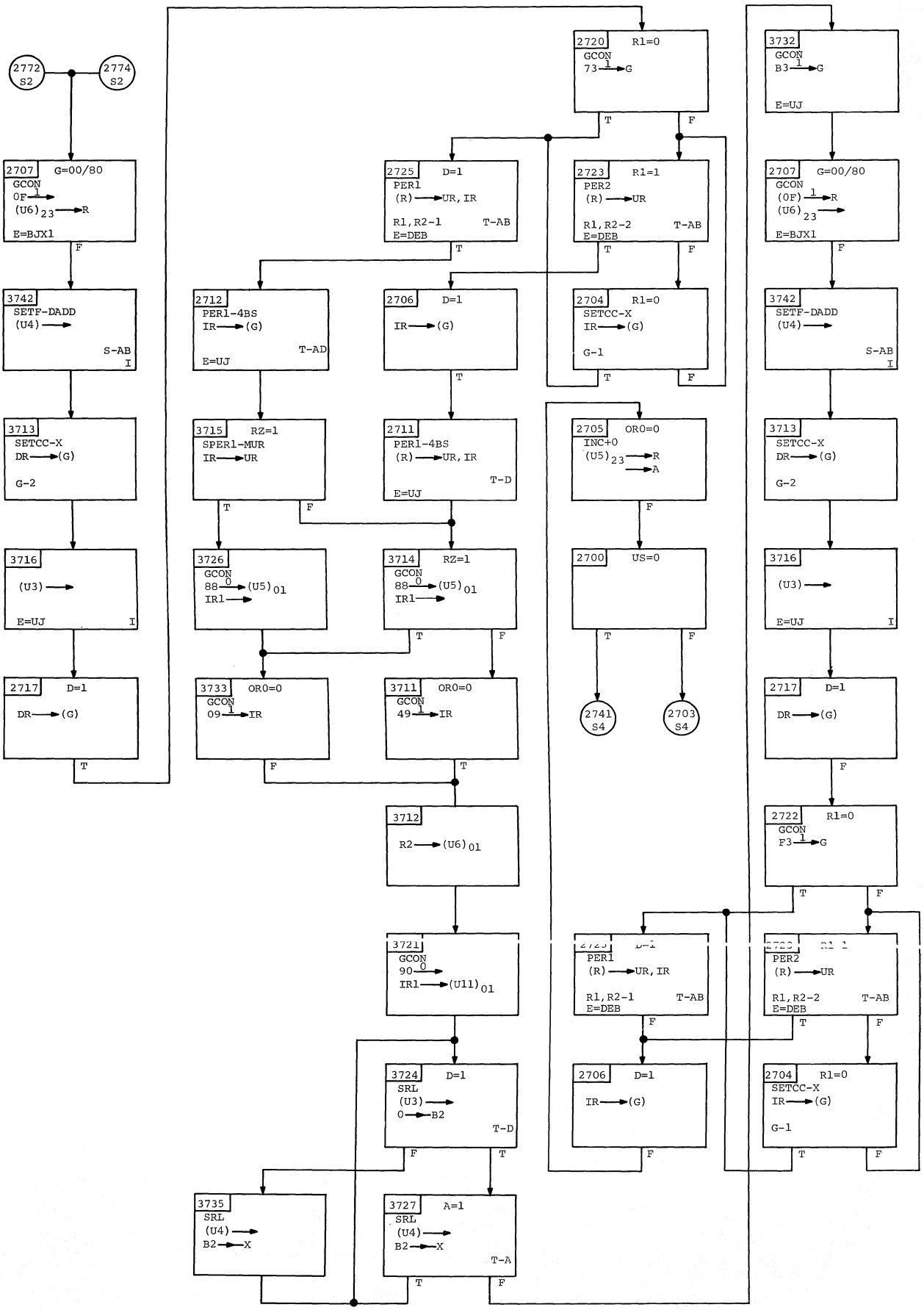
DIVIDE DECIMAL DP SS FD



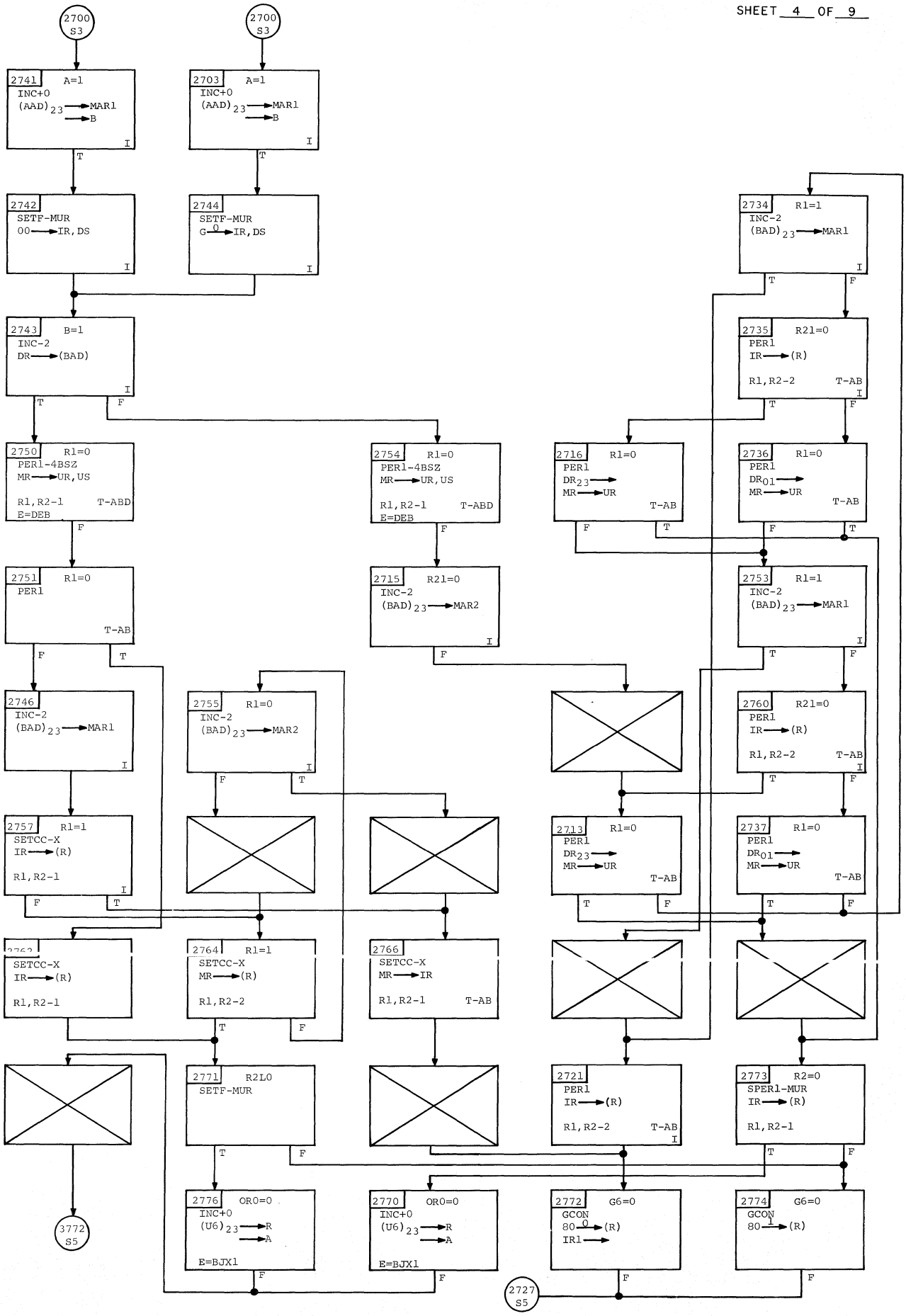
DIVIDE DECIMAL DP SS FD



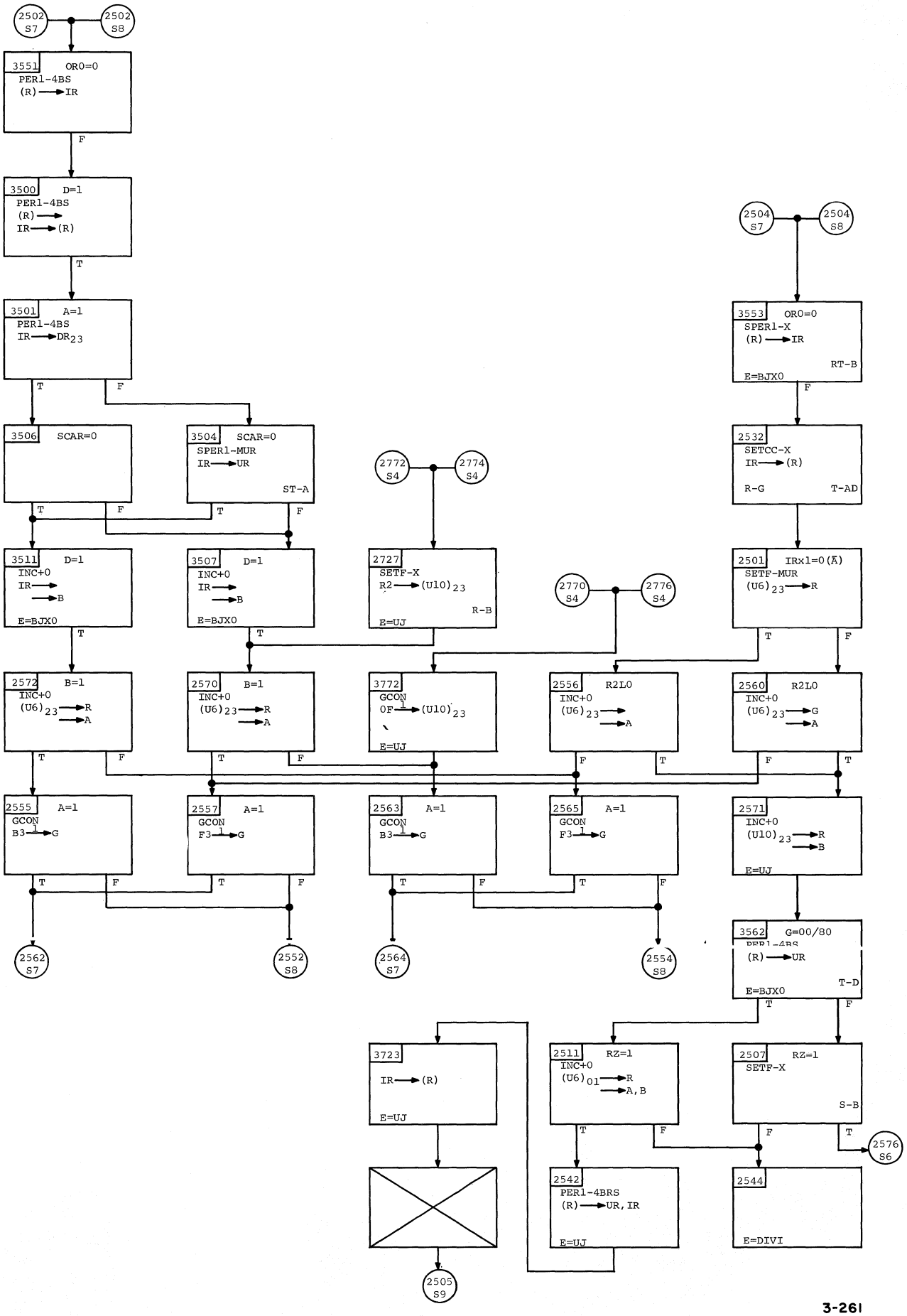
DIVIDE DECIMAL DP SS FD



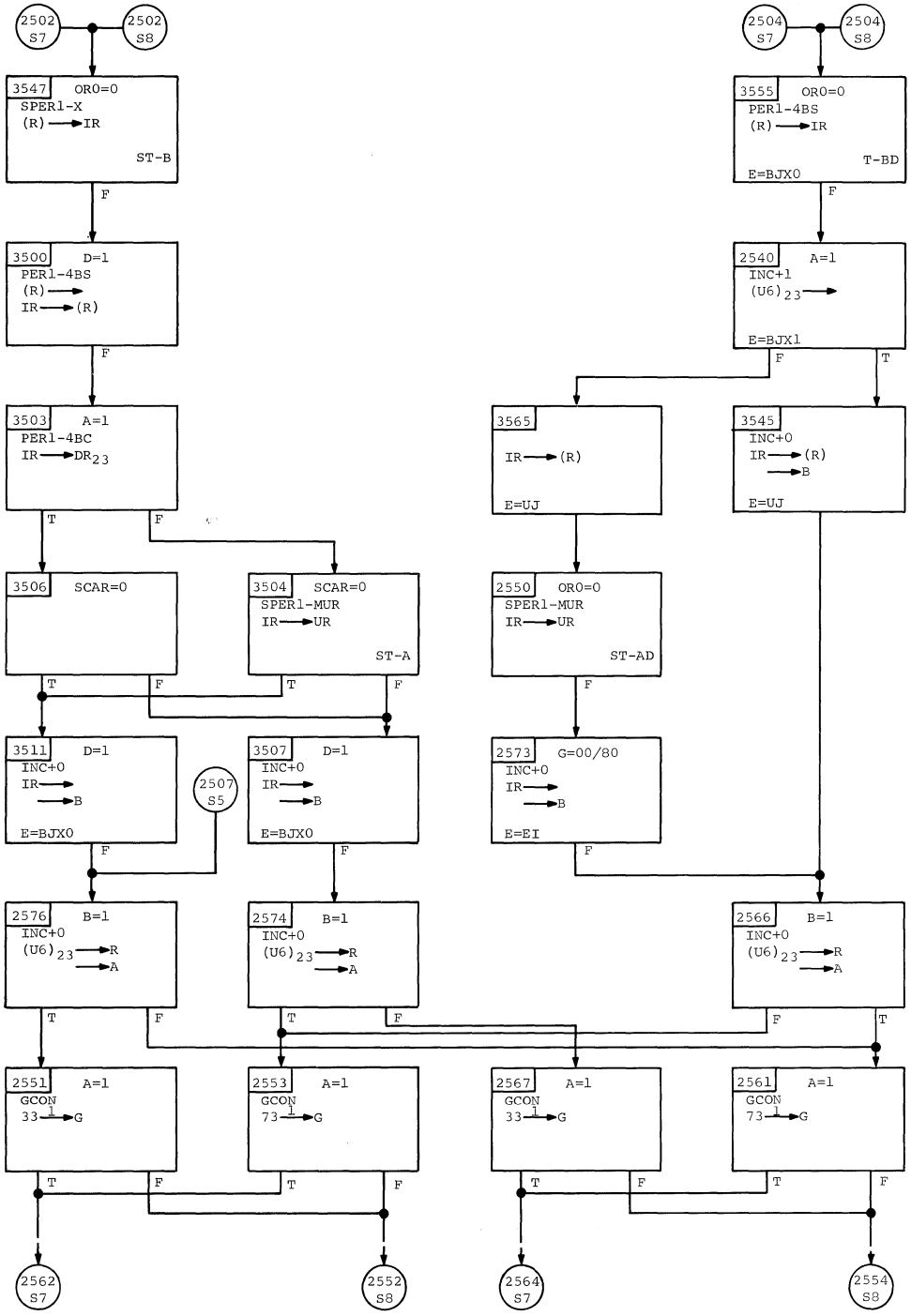
DIVIDE DECIMAL DP SS FD



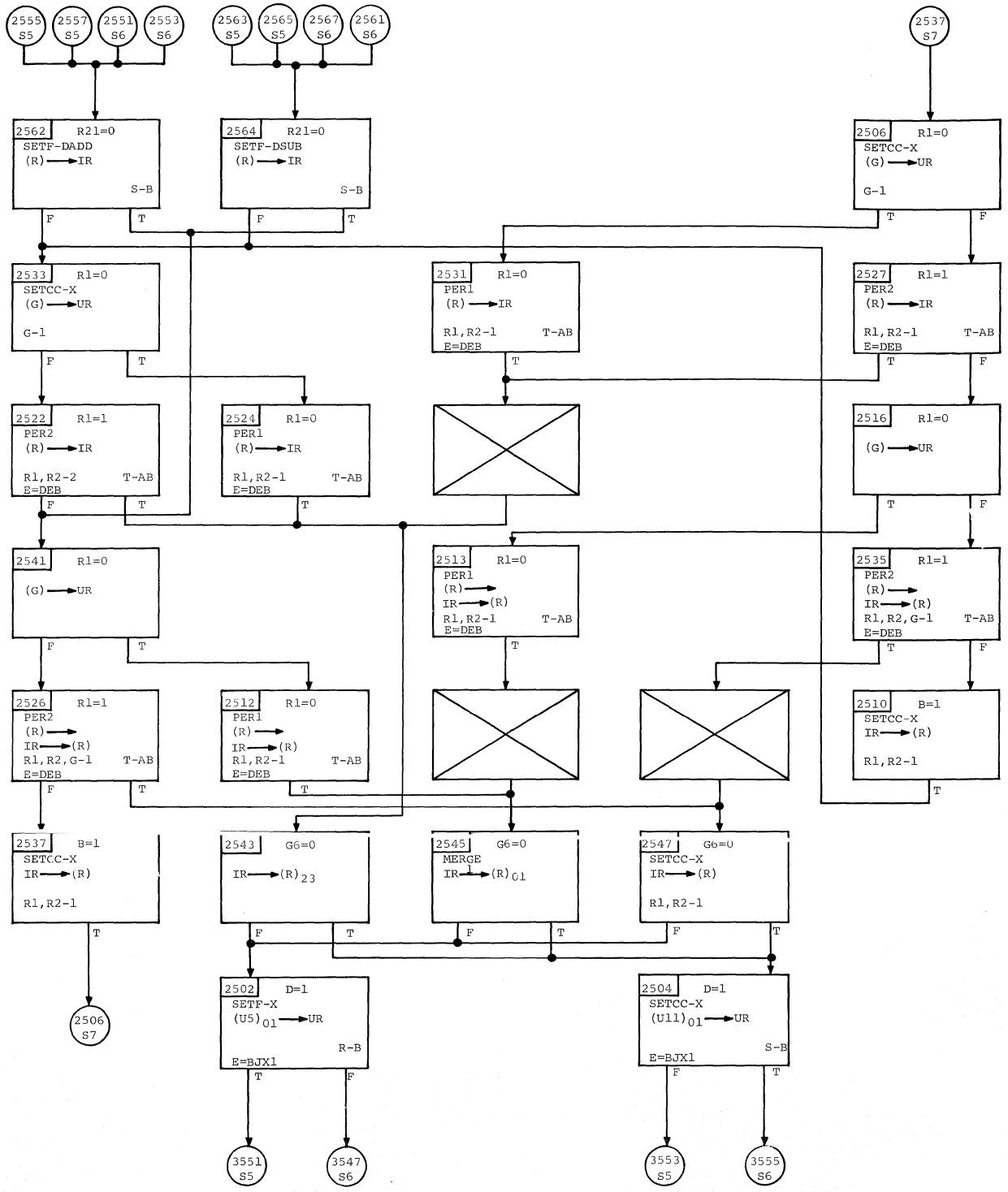
DIVIDE DECIMAL DP SS FD



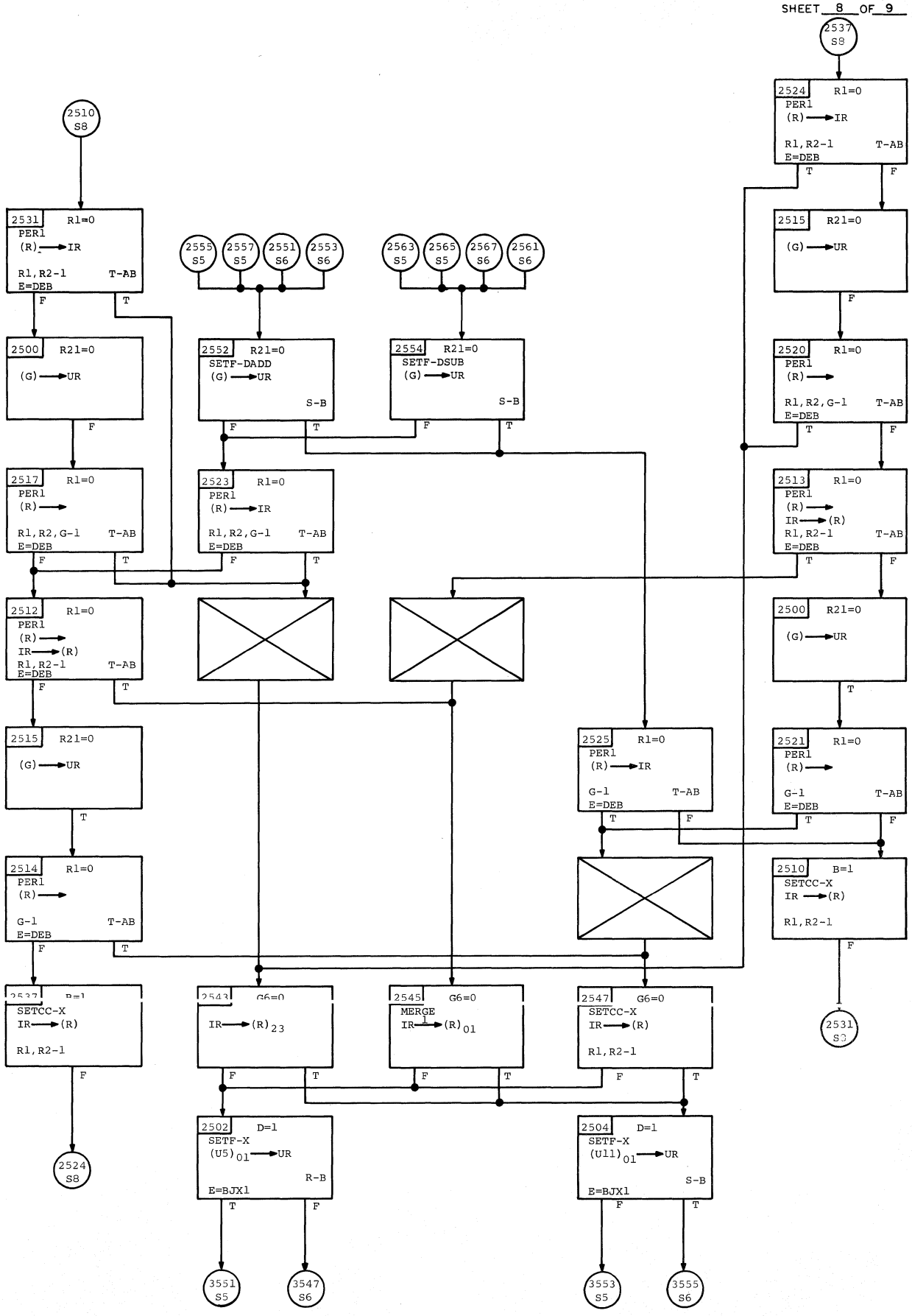
DIVIDE DECIMAL DP SS FD



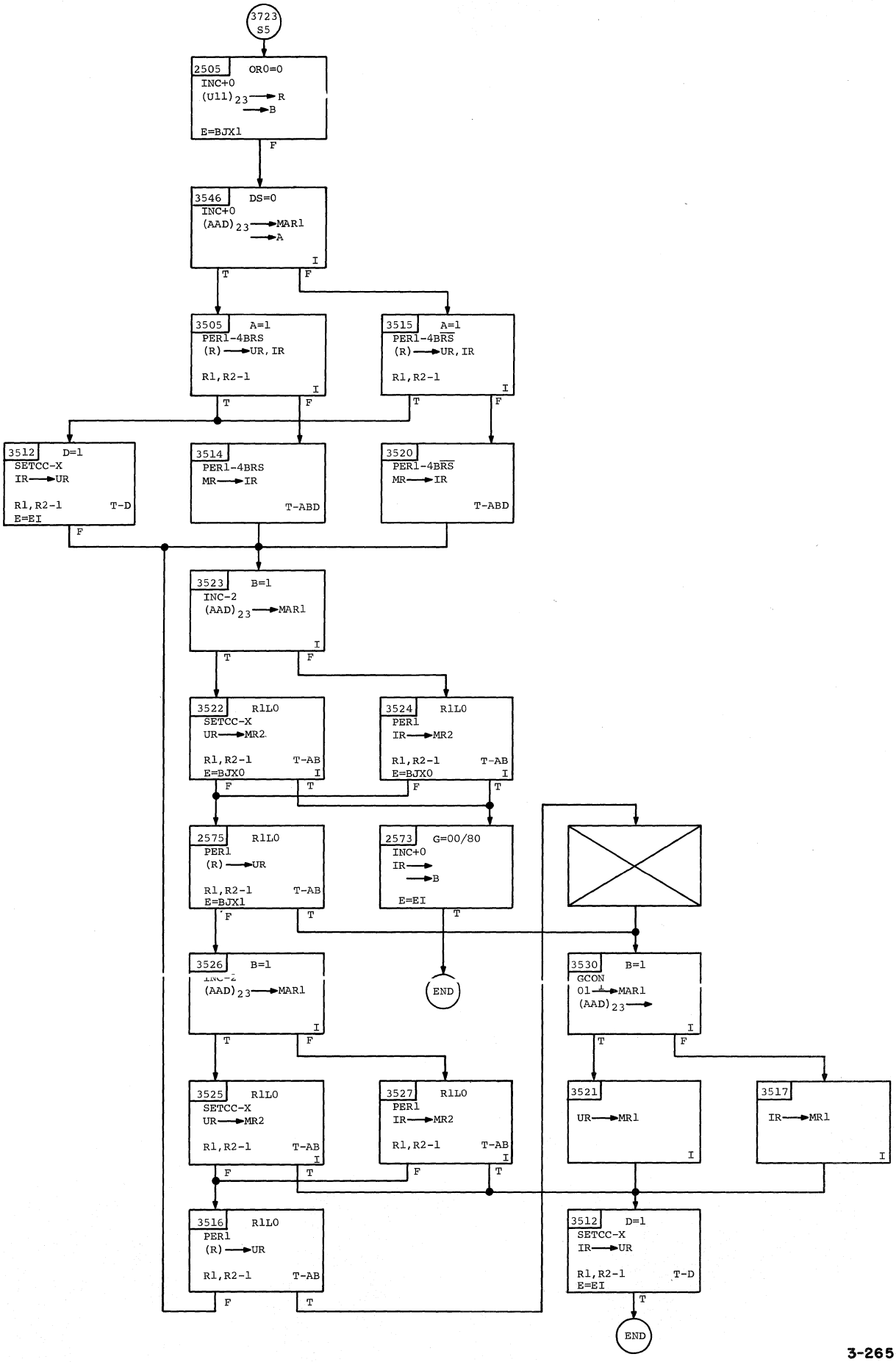
DIVIDE DECIMAL DP SS FD



DIVIDE DECIMAL DP SS FD



DIVIDE DECIMAL DP SS FD



RADIO CORPORATION OF AMERICA
ELECTRONIC DATA PROCESSING
CAMDEN, NEW JERSEY 08101