

# SPECTRA 70

RADIO CORPORATION OF AMERICA • ELECTRONIC DATA PROCESSING



SYSTEM

# 7045

**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

	<u>Page</u>
<u>INTRODUCTION</u>	1-1
<u>FVC FIELDS</u>	
F = 0 (NO FUNCTION)	1-2
F = 1 (SPER)	1-3
F = 2 (PER)	1-4
F = 3(0)·V3(0) (INC/MERGE)	1-7
F = 3·V3(1) (GCON)	1-8
F = 4(SLL/SLA/SRL/SRA)	1-9
F = 5 (TEST)	1-11
F = 6 (SETCC)	1-12
F = 7 (SETF)	1-14
<u>C7,MSD FIELDS F ≠ 5 (Data Transfer)</u>	
M FIELD (F ≠ 5)	1-15
FM ADDRESS CODES	1-16
HARDWARE SOURCE REGISTER CODES	1-19
HARDWARE DESTINATION REGISTER CODES	1-21
<u>TNA FIELDS (Branch Control)</u>	
BRANCH CONTROL FIELDS	1-24
TEST CONDITION CODES	1-25
<u>E FIELD</u>	
EXCEPTION CODES	1-27
<u>I FIELD</u>	
INHIBIT I/O SERVICING	1-30
SECTION 2	
EO FLOW CHART NOTATION	2-1

TABLE OF CONTENTS (Cont'd.)

SECTION 3. EO FLOW CHARTS

Title			Page
ELAPSED TIME CLOCK	ETC		3-1
INTERRUPT			3-2
SNAPSHOT	SNS		3-4
READ HIGH SPEED MEMORY	RDM		3-5
WRITE HIGH SPEED MEMORY	WRM		3-6
MULTIPLEXOR SERVICE	MUXS		3-7
SELECTOR SERVICE	SELS		3-10
<u>RR FORMAT</u>			
STATICIZING			3-12
NO OPERATION INSTRUCTIONS			3-13
SET PROGRAM MASK	SPM	04	3-14
BRANCH AND LINK	BALR	05	3-15
BRANCH ON COUNT	BCTR	06	3-16
BRANCH ON CONDITION	BCR	07	3-17
SET STORAGE KEY	SSK	08	3-18
INSERT STORAGE KEY	ISK	09	3-19
SUPERVISOR CALL	SVC	0A	3-20
LOAD POSITIVE	LPR	10	3-21
LOAD NEGATIVE	LNR	11	3-22
LOAD AND TEST	LTR	12	3-23
LOAD COMPLEMENT	LCR	13	3-24
LOGICAL AND	NR	14	3-25
COMPARE LOGICAL	CLR	15	3-26

TABLE OF CONTENTS (Cont'd.)

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

Title			Page
LOGICAL OR	OR	16	3-27
LOGICAL EXCLUSIVE OR	XR	17	3-28
LOAD	LR	18	3-29
COMPARE	CR	19	3-30
ADD	AR	1A	3-31
SUBTRACT	SR	1B	3-32
MULTIPLY	MR	1C	3-33
DIVIDE	DR	1D	3-34
ADD LOGICAL	ALR	1E	3-37
SUBTRACT LOGICAL	SLR	1F	3-38
LOAD POSITIVE LONG	LPDR	20	3-39
LOAD NEGATIVE LONG	LNDR	21	3-40
LOAD AND TEST LONG	LTDR	22	3-41
LOAD COMPLEMENT LONG	LCDR	23	3-42
HALVE LONG	HDR	24	3-43
LOAD LONG	LDR	28	3-44
COMPARE LONG	CDR	29	3-45
ADD NORMALIZED LONG	ANDR	2A	3-47
SUBTRACT NORMALIZED LONG	SNDR	2B	3-50
MULTIPLY LONG	NMDR	2C	3-53
DIVIDE LONG	NDDR	2D	3-57
ADD UNNORMALIZED LONG	AWR	2E	3-61
SUBTRACT UNNORMALIZED LONG	SWR	2F	3-64

TABLE OF CONTENTS (Cont'd.)

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

Title			Page
LOAD POSITIVE SHORT	LPER	30	3-67
LOAD NEGATIVE SHORT	LNER	31	3-68
LOAD AND TEST SHORT	LTER	32	3-69
LOAD COMPLEMENT SHORT	LCER	33	3-70
HALVE SHORT	HER	34	3-71
LOAD SHORT	LER	38	3-72
COMPARE SHORT	CER	39	3-73
ADD NORMALIZED SHORT	ANER	3A	3-75
SUBTRACT NORMALIZED SHORT	SNER	3B	3-78
MULTIPLY SHORT	NMER	3C	3-81
DIVIDE SHORT	NDER	3D	3-84
ADD UNNORMALIZED SHORT	AUR	3E	3-87
SUBTRACT UNNORMALIZED SHORT	SUR	3F	3-90
<u>RX FORMAT</u>			
STATICIZING			3-93
NO OPERATION INSTRUCTIONS			3-94
STORE HALFWORD	STH	40	3-95
LOAD ADDRESS	LA	41	3-96
STORE CHARACTER	STC	42	3-97
INSERT CHARACTER	IC	43	3-98
EXECUTE	EX	44	3-99
BRANCH AND LINK	BAL	45	3-101
BRANCH ON COUNT	BCT	46	3-102

TABLE OF CONTENTS (Cont'd.)

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

Title			Page
BRANCH ON CONDITION	BC	47	3-103
LOAD HALFWORD	LH	48	3-104
COMPARE HALFWORD	CH	49	3-105
ADD HALFWORD	AH	4A	3-106
SUBTRACT HALFWORD	SH	4B	3-107
MULTIPLY HALFWORD	MH	4C	3-108
CONVERT TO DECIMAL	CVD	4E	3-109
CONVERT TO BINARY	CVB	4F	3-111
STORE	ST	50	3-113
LOGICAL AND	N	54	3-114
COMPARE LOGICAL	CL	55	3-115
LOGICAL OR	O	56	3-116
LOGICAL EXCLUSIVE OR	X	57	3-117
LOAD	L	58	3-118
COMPARE	C	59	3-119
ADD	A	5A	3-120
SUBTRACT	S	5B	3-121
MULTIPLY	M	5C	3-122
DIVIDE	D	5D	3-123
ADD LOGICAL	AL	5E	3-126
SUBTRACT LOGICAL	SL	5F	3-127
STORE LONG	STD	60	3-128
LOAD LONG	LD	68	3-129

TABLE OF CONTENTS (Cont'd.)

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

Title			Page
COMPARE LONG	CD	69	3-130
ADD NORMALIZED LONG	AND	6A	3-132
SUBTRACT NORMALIZED LONG	SND	6B	3-135
MULTIPLY LONG	NMD	6C	3-138
DIVIDE LONG	NDD	6D	3-142
ADD UNNORMALIZED LONG	AW	6E	3-146
SUBTRACT UNNORMALIZED LONG	SW	6F	3-149
STORE SHORT	STE	70	3-152
LOAD SHORT	LE	78	3-153
COMPARE SHORT	CE	79	3-154
ADD NORMALIZED SHORT	ANE	7A	3-156
SUBTRACT NORMALIZED SHORT	SNE	7B	3-159
MULTIPLY SHORT	NME	7C	3-162
DIVIDE SHORT	NDE	7D	3-165
ADD UNNORMALIZED SHORT	AU	7E	3-168
SUBTRACT UNNORMALIZED SHORT	SU	7F	3-171
 <u>RS/SI FORMAT</u>			
STATICIZING			3-174
NO OPERATION INSTRUCTIONS			3-175
IDLE	IDLE	80	3-176
PROGRAM CONTROL	PC	82	3-177
DIAGNOSE	DIA	83	3-178
WRITE DIRECT	WRD	84	3-179



TABLE OF CONTENTS (Cont'd.)

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

Title			Page
READ DIRECT	RDD	85	3-180
BRANCH ON INDEX HIGH	BXH	86	3-181
BRANCH ON INDEX LOW OR EQUAL	BXLE	87	3-182
SHIFT RIGHT LOGICAL	SRL	88	3-183
SHIFT LEFT LOGICAL	SLL	89	3-184
SHIFT RIGHT ALGEBRAIC	SRA	8A	3-185
SHIFT LEFT ALGEBRAIC	SLA	8B	3-186
SHIFT RIGHT DOUBLE LOGICAL	SRDL	8C	3-187
SHIFT LEFT DOUBLE LOGICAL	SLDL	8D	3-188
SHIFT RIGHT DOUBLE ALGEBRAIC	SRDA	8E	3-189
SHIFT LEFT DOUBLE ALGEBRAIC	SLDA	8F	3-190
STORE MULTIPLE	STM	90	3-191
TEST UNDER MASK	TM	91	3-192
MOVE IMMEDIATE	MVI	92	3-193
LOGICAL AND	NI	94	3-194
COMPARE LOGICAL	CLI	95	3-195
LOGICAL OR	OI	96	3-196
LOGICAL EXCLUSIVE OR	XI	97	3-197
LOAD MULTIPLE	LM	98	3-198
START DEVICE	STD	9C	3-199
TEST DEVICE	TDV	9D	3-202
HALT DEVICE	HDV	9E	3-203
TEST CHANNEL	CKC	9F	3-204

TABLE OF CONTENTS (Cont'd.)

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

Title			Page
<u>SS FORMAT</u>			
STATICIZING			3-205
NO OPERATION INSTRUCTIONS			3-206
STORE SCRATCH PAD	SSP	D0	3-207
MOVE NUMERICS	MVN	D1	3-208
MOVE CHARACTERS	MVC	D2	3-210
MOVE ZONES	MVZ	D3	3-211
LOGICAL AND	NC	D4	3-213
COMPARE LOGICAL	CLC	D5	3-214
LOGICAL OR	OC	D6	3-215
LOGICAL EXCLUSIVE OR	XC	D7	3-216
LOAD SCRATCH PAD	LSP	D8	3-217
TRANSLATE	TR	DC	3-218
TRANSLATE AND TEST	TRT	DD	3-219
EDIT	EDIT	DE	3-220
EDIT AND MARK	EDMK	DF	3-225
MOVE WITH OFFSET	MVO	F1	3-230
PACK	PACK	F2	3-231
UNPACK	UNPK	F3	3-232
ZERO AND ADD DECIMAL	ZAP	F8	3-233
COMPARE DECIMAL	CP	F9	3-237
ADD DECIMAL	AP	FA	3-239
SUBTRACT DECIMAL	SP	FB	3-243

TABLE OF CONTENTS (Cont'd.)

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

Title			Page
MULTIPLY DECIMAL	MP	FC	3-247
DIVIDE DECIMAL	DP	FD	3-257

**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

$F = (F)_8 = F(2-0)$  -- Function

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

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

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

$M = (M)_4 = M(1-0)$  -- FM Control

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

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

### TNA Fields (Branch Control)

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

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

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

### E Field

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

### I Field

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

FVC

F = 0 (NO FUNCTION)

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		Description
		FV #1	FV #2	
10	SPER1-X	70	20	Function Unchanged
18	SPER2-X	70	28	Function Unchanged
11	SPER1-DADDf(IRO3)	71	20	Decimal Add, IRO3 → ICAR
19	SPER2-DADDf(IRO3)	71	28	Decimal Add, IRO3 → 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-UR	75	20	Two's Complement of UR
1D	SPER2-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  $(\overline{V} = 0/8) \cdot BDB(1)$ : CAR3 → SCAR  
 CAR7 ≠ CAR6 → OV  
 SUM 8 → B0  
 SUM 9 → B1  
 CAR9 → B3

If  $[V = 0/V = 8/BDB(0)] \cdot [Adder(7-4) \neq 0]$ : Reset RZ

If  $[V = 0/V = 8/BDB(1)] \cdot [Adder(3-0) \neq 0]$ : 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(9-8) = COMP(0) · G(1-0)/COMP(1) ·  $\overline{G(1-0)}$   
 U(7-0) = COMP(0)BCD(0)UR(7-0)/COMP(1) ·  $\overline{UR(7-0)}$   
 U(7-4) = COMP(0)BCD(1) ·  $[6 + UR(7-4)]$   
 U(3-0) = COMP(0)BCD(1) ·  $[6 + UR(3-0)]$   
 P(9-8) = U(9-8) · DR(9-8)  
 P(7-0) =  $\overline{ELOR(1)}/U(7-0) \cdot DR(7-0)$   
 N(9-8) =  $\overline{U(9-8)} \cdot \overline{DR(9-8)}$   
 N(7-0) = ELAND(0) ·  $\overline{U(7-0)} \cdot \overline{DR(7-0)}$   
 SUM(9-1) =  $[P(9-1)/N(9-1)] \cdot CAR(8-0) / [P(9-1)/N(9-1)] \cdot \overline{CAR(8-0)}$   
 SUM0 = (P0/N0) · ICAR(1) /  $\overline{(P0/N0)}$  · ICAR(0)  
 CAR(9-4) =  $\overline{N(9-4)} \cdot CAR(8-3) / P(9-4) / SIMC(1)$   
 CAR3 =  $(\overline{V=0/8}) \cdot BDB(0) \cdot ICAR(1)$   
 CAR3 =  $[V = 0/V = 8/BDB(1)] \cdot [N3 \cdot CAR2/P3/SIMC(1)]$   
 CAR(2-1) =  $\overline{N(2-1)} \cdot CAR(1-0) / P(2-1) / SIMC(1)$   
 CAR0 =  $\overline{N0} \cdot ICAR(1) / P0 / SIMC(1)$   
 ADDER(7-4) =  $[BCD(0)/CAR7] \cdot SUM(7-4) / BCD(1) \cdot \overline{CAR7} \cdot [A+SUM(7-4)]$   
 ADDER(3-0) =  $[BCD(0)/CAR3] \cdot SUM(3-0) / BCD(1) \cdot \overline{CAR3} \cdot [A+SUM(3-0)]$



FVC FIELDS

F = 2(PER)

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

X = Unchanged

If A(0): Zone = F

If A(1): Zone = 5

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

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

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

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

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

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

<u>C(7-4)</u>	<u>Mnemonic</u>	<u>C(7-4)</u>	<u>Mnemonic</u>	<u>C(7-4)</u>	<u>Mnemonic</u>	<u>C(7-4)</u>	<u>Mnemonic</u>
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.

<u>C(3-0)</u>	<u>Mnemonic</u>	<u>OR7(0)/OR6(0)</u>	<u>Mnemonic</u>	<u>OR7(1)OR6(1)OR5(0)</u>	<u>Mnemonic</u>	<u>OR7(1)OR6(1)OR5(0)</u>	<u>Mnemonic</u>
0	None		None		None		None
4	R-G		R-G		R-G		R-G
5	G-1		G-1		G-1		G-1
6	G-2		G-2		G-2		G-2
7	G-4		G-4		G-4		G-4
8	S-R2		R1-1, S-R2		R1-1, S-R2		R1-1, S-R2
9	R2-1		R-1		R1, R2-1		R1, R2-1
A	R2-2		R-2		R1, R2-2		R1, R2-2
B	R2-4		R-4		R1, R2-4		R1, R2-4
C	S-R2, R-G		R1-1, 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		R1, R2, G-1
E	R2, G-2		R, G-2		R1, R2, G-2		R1, R2, G-2
F	R2, G-4		R, G-4		R1, R2, 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  $\longrightarrow$  WDB After Bus Transfer

M = 1 · (D = 08/0C) · C7(0)WDB(1): (S)<sub>01</sub>  $\longrightarrow$  (S)<sub>23</sub>

M = 1 · (D = 08/0C) · C7(1)WDB(0): (S)<sub>01</sub>  $\longleftarrow$  (S)<sub>23</sub>

M = 3 · WDB(1): D  $\longrightarrow$  (S)<sub>23</sub> Independent of C7

M = 3 · WDB(0): D  $\longrightarrow$  (S)<sub>01</sub> 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.

<u>FV</u>	<u>Mnemonic</u>	<u>Description</u>	<u>M</u>
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

<u>C(7-4)</u>	<u>Mnemonic</u>	<u>C(7-4)</u>	<u>Mnemonic</u>
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

<u>C(3-0)</u>	<u>Mnemonic</u>	<u>Description Standard</u>
0	INC	
1	MERGE	BUS0 → <sup>0</sup> DR0
2	MERGE	BUS1 → <sup>1</sup> DR1
3	MERGE	BUS → <sup>01</sup> DR0, DR1

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

<u>M</u>	<u>C7</u>	<u>Standard Transfer</u>	+	<u>Second Transfer When C(3-0) ≠ 0</u>
0	0	S → D		S → DR <sub>0/1</sub>
1	1	(S) <sub>23</sub> → D		(S) <sub>23</sub> → (S) <sub>0/1</sub>
3	0	<u>D = 0B</u> None		D → (S) <sub>0/1</sub>
3	0	D = 0B DR → (S) <sub>0/I</sub>		00 → (S) <sub>0/1</sub>

TNA-E-I: Standard

<u>FV</u>	<u>Mnemonic</u>	<u>Bus</u>
39/3D	V2,C(6-0) $\xrightarrow{0}$	BUS0(7-0)
3A/3E	V2,C(6-0) $\xrightarrow{1}$	BUS1(7-0)
3B/3F	V2,C(6-0) $\xrightarrow{01}$	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/B0/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.

<u>FV</u>	<u>Mnemonic</u>	<u>DR07</u> <u>Stored In</u>	<u>FM2<sup>0</sup></u> <u>Filled From</u>	
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
		<u>FM2<sup>31</sup></u> <u>Filled From</u>	<u>DR30</u> <u>Stored In</u>	
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) = DR07 if OV(0) = $\overline{DR07}$ if OV(1)

<u>C(7-4)</u>	<u>Mnemonic</u>	<u>C(7-4)</u>	<u>Mnemonic</u>	<u>C(7-4)</u>	<u>Mnemonic</u>	<u>C(7-4)</u>	<u>Mnemonic</u>
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.

<u>C(3-0)</u>	<u>OR7(0)/OR6(0)</u> <u>Mnemonic</u>	<u>OR7(1)OR6(1)OR5(0)</u> <u>Mnemonic</u>	<u>OR7(1)OR6(1)OR5(0)</u> <u>Mnemonic</u>
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

<u>T</u>	<u>D</u>	<u>S</u>	<u>Branch Address</u>	<u>ROM Address</u>	
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		Set SREG(5-4)		
		Class	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) ·P*	RZ(0) ·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  
SREG(5-4)

		9	Unchanged
		<u>1-9</u>	00
64	SETCC-0	<u>9</u>	00
		9	Unchanged
65	SETCC-1	<u>9</u>	01
		9	Set SREG4
66	SETCC-2	<u>9</u>	10
		9	Set SREG5
67	SETCC-3		11
68	SETCC-X		Unchanged

				OR	Class					OR	Class
8A	8B	8E	8F	-1		D6	D7	DC	DE	DF	-4
12	13	1A	1B	-2		1E	1F	5E	5F		-5
4A	4B	5A	5B	-2		10	90				-6
15	1D	55	5D	-3		91					-7
95	D5	DD		-3		X9	<u>89</u>				-8
11	14	16	17	1C	-4	08	18	48	58		-9
54	56	57	5C		-4	88	89	8C	89		-9
94	96	97	D4		-4						



<u>C(7-4)</u>	<u>Mnemonic</u>	<u>C(7-4)</u>	<u>Mnemonic</u>	<u>C(7-4)</u>	<u>Mnemonic</u>	<u>C(7-4)</u>	<u>Mnemonic</u>
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.

<u>C(3-0)</u>	<u>Mnemonic</u>	<u>OR7(1)OR6(1)OR5(0)</u>	<u>Mnemonic</u>	<u>OR7(1)OR6(1)OR5(0)</u>	<u>Mnemonic</u>
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)

<u>FV</u>	<u>Mnemonic</u>	<u>EDR</u>	<u>COMP</u>	<u>ICAR</u>	<u>BCD</u>	<u>SIMC</u>	<u>Eland</u>	<u>ELOR</u>	<u>SCAR</u>	<u>OV</u>	<u>RZ</u>	<u>Function</u>
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-UR	0	1	1	0	0	0	0	X	0	1	0 + 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, ICAR	1	1	0	0	0	0	0	X	0	1	DR + UR

<u>C</u>	<u>Mnemonic</u>	<u>C</u>	<u>Mnemonic</u>	<u>C</u>	<u>Mnemonic</u>
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, M=(1/3)·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

<u>M</u>	<u>C7</u>	<u>Mnemonic</u>	<u>FM Address Code</u>	<u>Hardware Source Register Code</u>	<u>Hardware Destination Register Code</u>	<u>FM Cycle</u>
0	X	S → D	None	S	D	0
1	0	(S) <sub>01</sub> → D	S	OB (DR)	D	1
1	1	(S) <sub>23</sub> → D	S	OB (DR)	D	1
3•D ≠ 0B	0	D → (S) <sub>01</sub>	S	D	08 (DR)	1
3•D ≠ 0B	1	D → (S) <sub>23</sub>	S	D	08 (DR)	1
3•D = 0B	X	DR → (S)	S	00	00	1

Inhibit  
FM Strobe

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

M	Field	Mnemonic	FM Address				
			11	10	01	00	
1	S		1	2	3	4	← PS2.PS1
3	S		1	2	3	4	← 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	Field		FM Address (I/O)							Register
	S	Mnemonic	Trunk No. T(2-0)							
			0	1	2	3	4	5	6	
3	S	Mnemonic	MUX	SEL1	SEL2	SEL3	SEL4	SEL5	SEL6	
	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 Field		11	10	01	00	PS2·PS1				
1	S	1	2	3	4	←	Prog. State			
3	S	Mnemonic	FMAR (6-4)	FMAR (6-4)	FMAR (6-4)	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	
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

C7,MSD FIELDS (F=5)

FM ADDRESS CODES

M	Field			
0	S			
3	D	<u>Mnemonic</u>	<u>Source Register</u>	
	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 DATAACK → 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 <sub>01</sub>	DR0 (7-0) → BUS0 (7-0)	If C7(0)·M≠3
		DR <sub>23</sub>	DR1 (7-0) → BUS1 (7-0) DR1 (1-0) → BUSA (1-0) DR2 (7-0) → BUS0 (7-0) DR3 (7-0) → BUS1 (7-0)	If C7(0)·M≠3 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 R1=0·R23(0)·R22(0): Set PS=01
	0E	G	G (7-0) <sup>0</sup> → BUS0 (7-0)	
	0F	G	G (7-0) <sup>1</sup> → 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	Field				
0	S				
3	D	<u>Mnemonic</u>	<u>Source Register</u>		
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	Field			
0	D			
<u>1</u>	<u>D</u>	<u>Mnemonic</u>	<u>Destination Register</u>	
00		None	None	
01		DOUT	BUS0(8-0) → DOUT(8-0) BUS1(8-0) → DOUT(8-0)	If MAR10(0). See Maint. Manual If MAR10(1).
02		CHR	BUS07 → MCDF BUS06 → MCCF BUS05 → MSLIF BUS04 → MSKF BUS03 → MPCIF BUS03 → ITSCAN BUS03 → CHR5 BUS0(2-0) → CHR(2-0)	See Maint. Manual
03		IAR	BUS0(7-0) → IAR(7-0)	See Maint. Manual
04		KCR	BUS0(7-4) → MKEY(4-1) BUS0(3-0) → COMM(3-0)	See Maint. Manual
05		FLR	BUS07 → MCDF BUS06 → MCCF BUS05 → MSLIF BUS04 → MSKF BUS03 → MPCIF BUS1(7-1) → MCHS(7-1)	See Maint. Manual
06		TR		See Maint. Manual
07		TICR		See Maint. Manual
08		DR01	BUS0(7-0) → DR0(7-0) BUS0(7-0) → DR0(7-0) BUS1(7-0) → DR1(7-0) BUS1(7-0) → DR1(7-0)	If F=2·M=1/3·WDB(0) If $\overline{C7(0)} \cdot F=2 \cdot M=1/3 \cdot F=3 \cdot V3(0) \cdot C1(1)$ If F=2·M=1/3·WDB(0) If $\overline{C7(0)} \cdot F=2 \cdot M=1/3 \cdot F=3 \cdot V3(0) \cdot C0(1)$

C7,MSD FIELDS (F≠5)

HARDWARE DESTINATION  
REGISTER CODES

M	Field			
0	D			
1	D	Mnemonic	Destination Register	
08	DR <sub>23</sub>	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)• F=2•M=1/3•F=3•V3(0)• <u>C1(1)/C0(1)</u>	
		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)• F=2•M=1/3•F=3•V3(0)• <u>C1(1)/C0(1)</u>	
		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) <u>C1(1)/C0(1)</u>	
		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) <u>C1(1)/C0(1)</u>	
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	Field			
0	D			
1	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) 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. $M=0 \cdot (S=0A/12/16) \cdot F=(1/2)$
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)  $\longrightarrow$  ROMAR(5-0). ROMAR(11-6) remains unchanged.

N: If  $F \neq 5 \cdot T\text{-False} / F = 5 \cdot T\text{-False} \cdot D\text{-True}$ : N(5-0)  $\longrightarrow$  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.

## TNA FIELDS

## TEST CONDITION CODES

<u>T</u>	<u>Mnemonic</u>	<u>Test True(T) If --</u>
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	$\overline{R1=0 \cdot R2=0 / R1=F \cdot R2=F}$
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	$\overline{OR7 (1) OR6 (1) OR5 (1) \cdot G4 (0) /OR7 (1) OR6 (1) OR5 (1) G6 (0)}$
	G6=0	$\overline{OR7 (1) OR6 (1) OR5 (1) \cdot G4 (0) /OR7 (1) OR6 (1) OR5 (1) G6 (0)}$
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 ( $\overline{A}$ )	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	$\overline{G=00/01/02/03/04}$
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

## TNA FIELDS

## TEST CONDITION CODES

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

<u>E</u>	<u>Mnemonic</u>	<u>Exception/Check</u>
00	None	No Exception/Check
01	UJ	<u>Unconditional Jump</u> -- A(5-0) $\longrightarrow$ ROMAR(11-6). If T-False: N(5-0) $\longrightarrow$ ROMAR(5-0), Next EO=AANN. If T-True: A(5-0) $\longrightarrow$ ROMAR(5-0), Next EO=AAAA. Used to Enter or Exit a Block of 64 EO's.
02	EI	<u>End Instruction</u> -- If T-False: N(5-0) $\longrightarrow$ ROMAR(5-0), ROMAR(11-6) Remains Unchanged; Next EO Address = XXNN. If T-True: A(5-0) $\longrightarrow$ ROMAR(5-0), Where AA = 00/01, and  1. If a Unit of Time has Passed on the Elapsed Time Clock, 15 $\longrightarrow$ 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 $\longrightarrow$ ROMAR(11-6); Next EO = 1300/1301, First EO of Interrupt  3. If Neither (1) nor (2), 11 $\longrightarrow$ ROMAR(11-6); Next EO = 1100/1101, First EO of Staticizing
03	ES	<u>End Staticizing</u> -- If T-False: N(5-0) $\longrightarrow$ ROMAR(5-0), ROMAR(11-6) Remains Unchanged and Next EO = XXNN. If T-True: A(5-0) $\longrightarrow$ ROMAR(5-0) Where AA = 00/01 and 0 $\longrightarrow$ ROMAR(11-9) and OR(7-0) $\longrightarrow$ ROMAR(8-1); Next EO = OXXX, First EO of Execution of an Instruction
04	ASM	<u>Address Shaded Memory</u> -- 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	<u>Instruction Address</u> -- If MAR10(1): Set INT24
06	DEB	<u>Data Error Both Digits</u> -- If $\overline{F} = (1/2)/RIUR \cdot URX(7-4) \neq 0-9$ : Set INT25 If $\overline{F} = (1/2)/RIUR \cdot RIUS \cdot URX(3-0) \neq 0-9$ : Set INT25 If RIUS URX(3-0) = 0-9: Set INT25 If $\overline{F} = (1/2)/RIDR \cdot DRX(7-4) \neq 0-9$ : Set INT25 If $\overline{F} = (1/2)/RIDR \cdot RIDS \cdot DRX(3-0) \neq 0-9$ : Set INT25 RIDS $\cdot$ DRX(3-0) = 0-9: Set INT25  Where: RIUR = F $\neq$ 5 $\cdot$ M = (0/1) $\cdot$ D = (0F/10/11/12/13) RIUS = F $\neq$ 5 $\cdot$ M = (0/1) $\cdot$ D = 11 RIDR = F $\neq$ 5 $\cdot$ $\overline{M}$ = (0/1) $\cdot$ D = (08/0C)/M = 3 $\cdot$ D = 0B RIDS = F $\neq$ 5 $\cdot$ M = (0/1) $\cdot$ D = OD

Also See Destination Register Codes #11 and #0D

<u>E</u>	<u>Mnemonic</u>	<u>Exception/Check</u>
07	DEM	<u>Data Error Most Significant Digit</u> -- If $\overline{F} = (1/2)/RIUR$ URX(7-4) $\neq$ 0-9: Set INT25 If $\overline{F} = (1/2)/RIDR$ DRX(7-4) $\neq$ 0-9: Set INT25 Where: RIUR = F $\neq$ 5 • M = (0/1) • D = (0F/10/11/12/13) RIDR = F $\neq$ 5 • $\overline{M} = (0/1) \cdot D = (08/0C)/M = 3 \cdot D \neq 0B$ Also See Destination Register Codes #11 and #0D
08	EX	<u>Execute</u> -- If OR = 44: Set INT24
09	HW	<u>Half Word Address</u> -- If IR10(1): Set INT24
0A	FW	<u>Full Word Address</u> -- If IR11(1)/IR10(1): Set INT24
0B	DW	<u>Double Word Address</u> -- If IR12(1)/IR11(1)/IR10(1): Set INT24
0C	R	<u>R Specification</u> If R10(1)/OR5(1)R13(1): Set INT24 If OR6(0)OR5(1) • $\overline{R23(1)/R20(1)}$ : Set INT24
0D	PRIV	<u>Privileged</u> -- If PRIV(1): Set INT22
0E	NOOP	<u>Illegal Instruction</u> -- Set INT23
0F	R2	<u>R2 Specification</u> -- If R23(1)/RZ(1)/SCAR(0): Set INT24
10	DE	<u>Data Error</u> -- Set INT25
14	FOV	<u>Fixed Pt. Overflow</u> -- If OV(1) SREG3(1): Set INT31 If OR = (8B/8F) SREG3(1): Set INT31
15	EOV	<u>Exponent Overflow</u> -- Set INT26
16	ETC	<u>Elapsed Time Clock</u> -- Set INT17
17	ESRV	<u>End I/O Servicing</u> -- 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	<u>Divide Error</u> -- Set INT27
19	SIGE	<u>Significance Error</u> -- If SREG0(1): Set INT28



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

<u>I</u>	<u>Mnemonic</u>	<u>Function</u>
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  $M = (0/1) \cdot D = 1C/LD$ ,  $I(1)$  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.

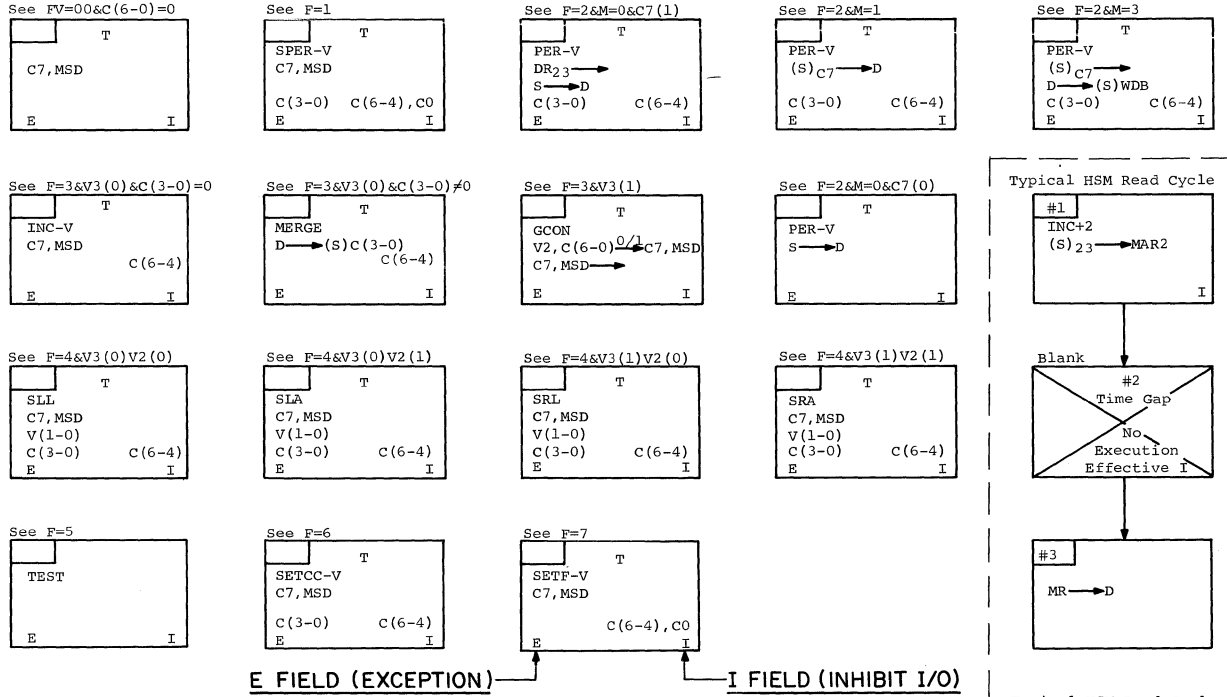
If the next EO is a blank, the I bit also applies to the blank.

In the Normal mode, the next normal EO address is preserved in REGAD and MREGAD -- ROMAR(11-0)  $\longrightarrow$  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)  $\longrightarrow$  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)  $\longrightarrow$  ROMAR(11-0) and the BPU continues MUX service. When returning from SEL/MUX service to Normal mode, REGAD(11-0)  $\longrightarrow$  ROMAR(11-0) and the BPU continues in the Normal mode.

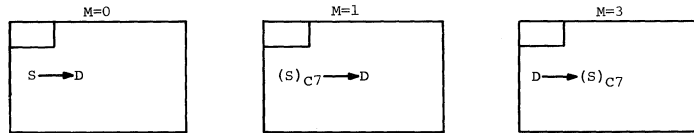
**SECTION 2**  
**EO FLOW CHART NOTATION**

FVC FIELD (FUNCTION)

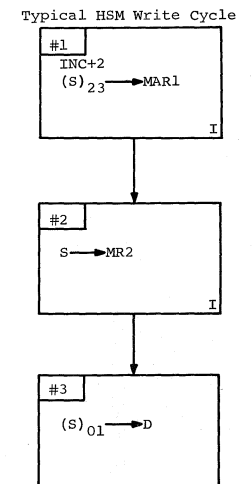
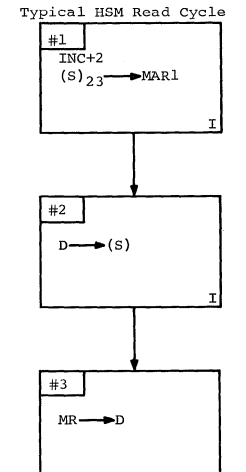
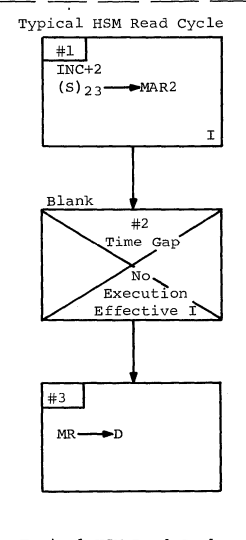
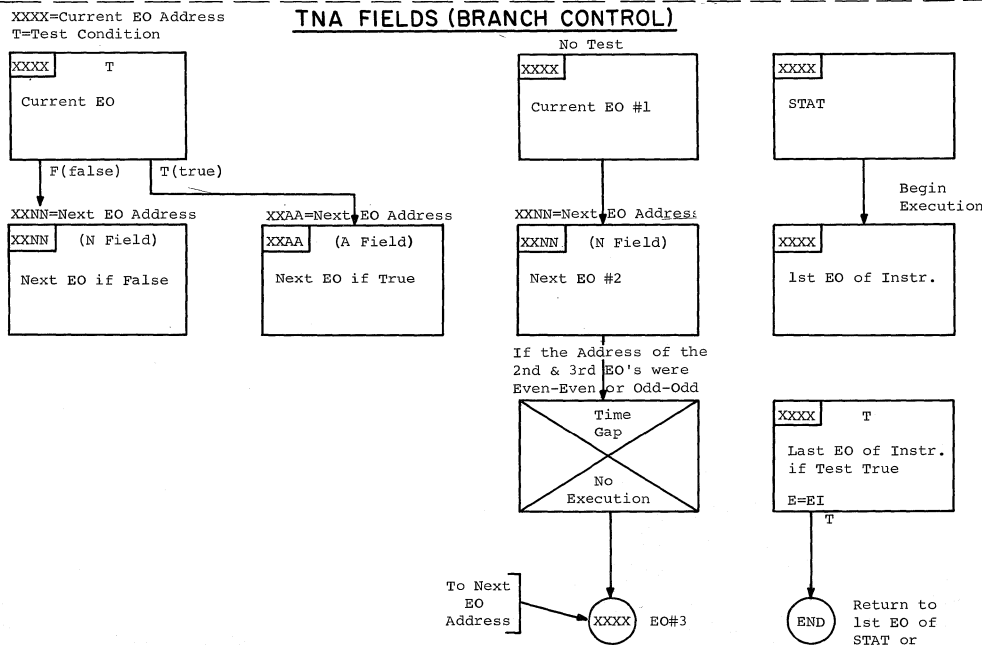


E FIELD (EXCEPTION)      I FIELD (INHIBIT I/O)

C7,MSD FIELDS (DATA TRANSFER) F≠5

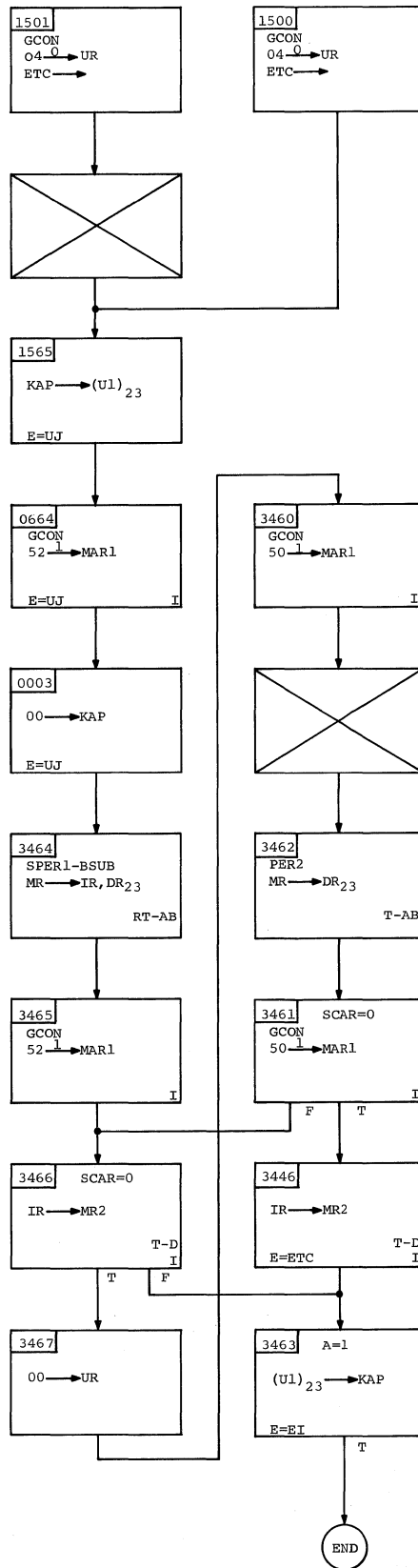


TNA FIELDS (BRANCH CONTROL)

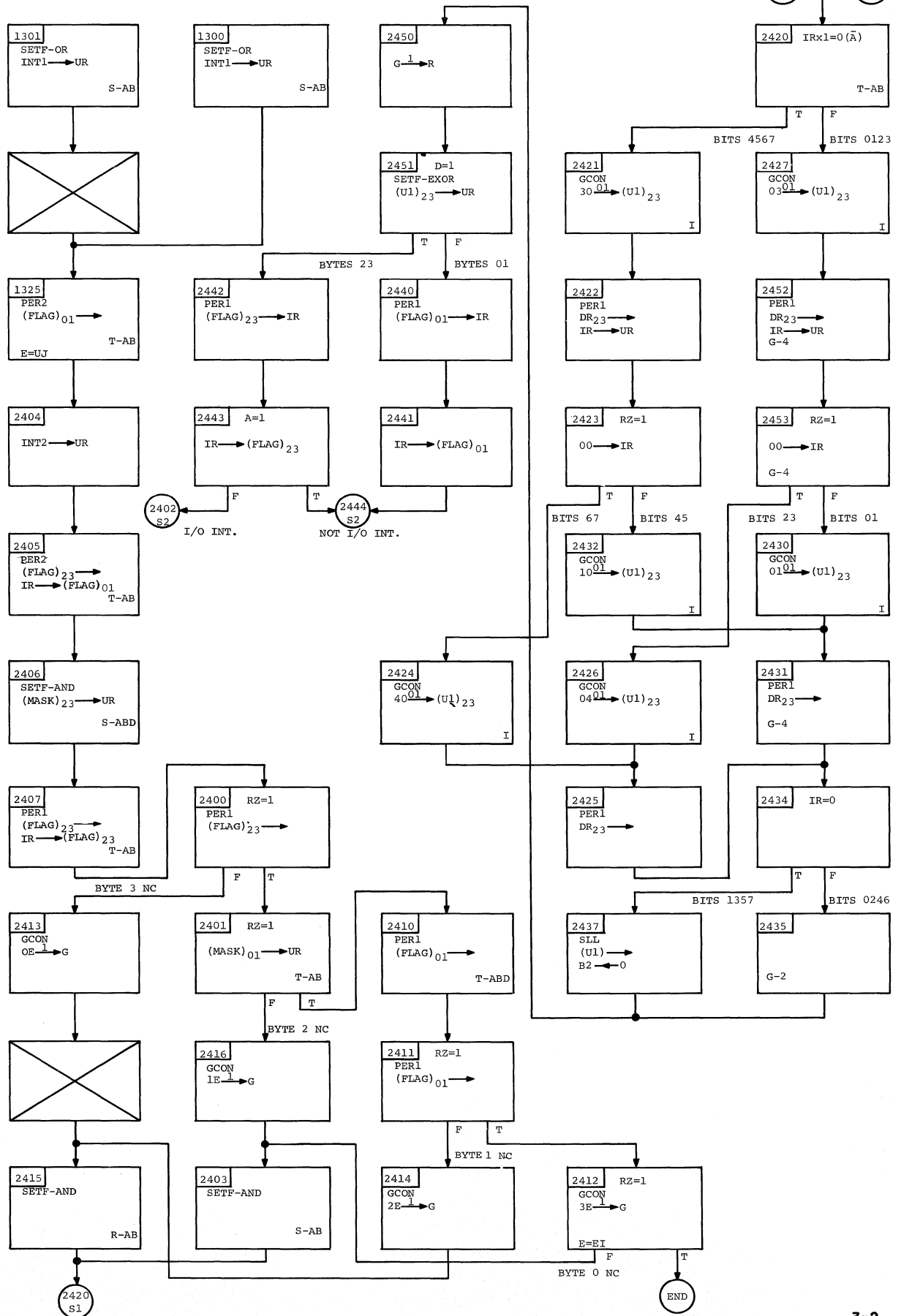


**SECTION 3**  
**EO FLOW CHARTS**

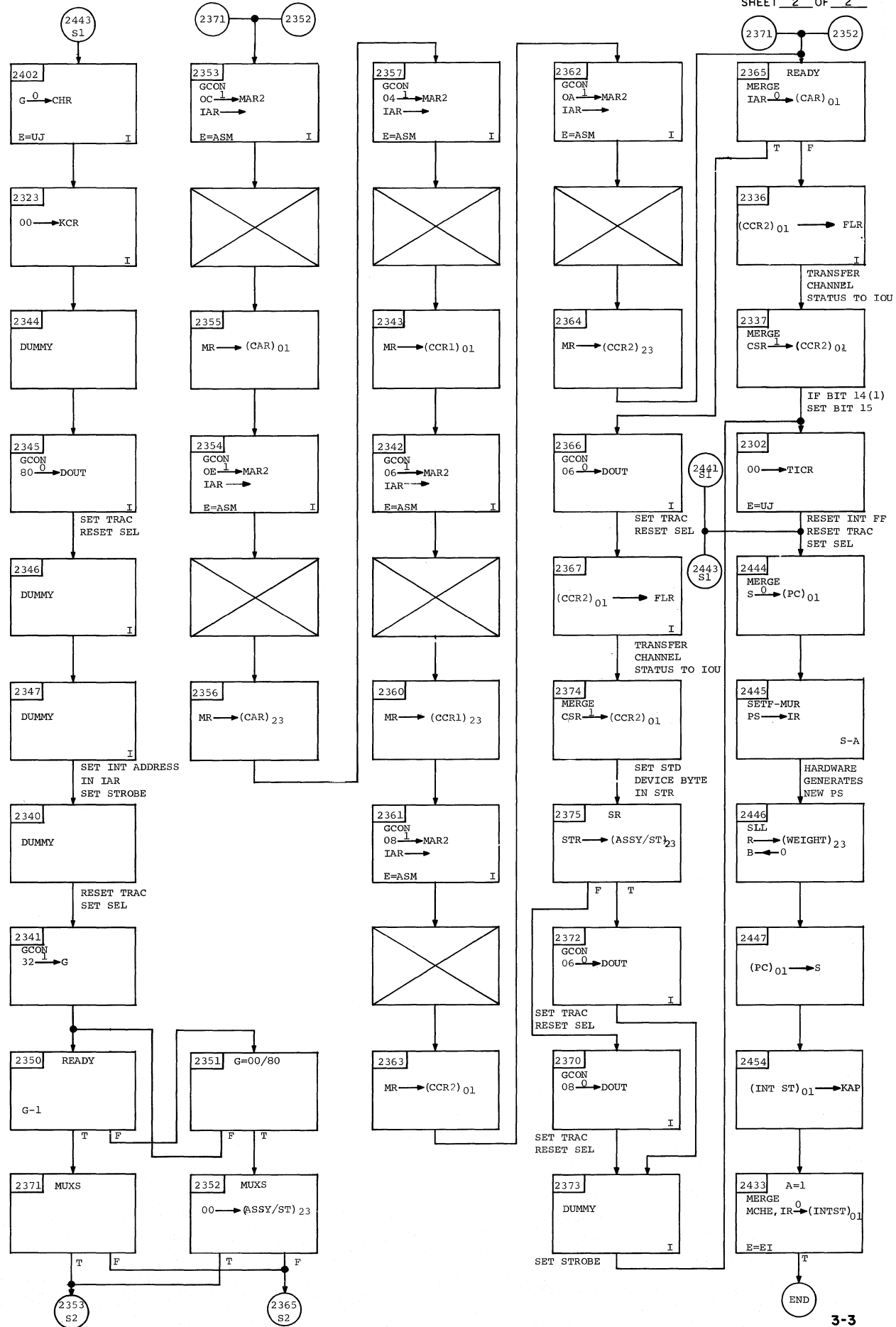
ELAPSED TIME CLOCK ETC



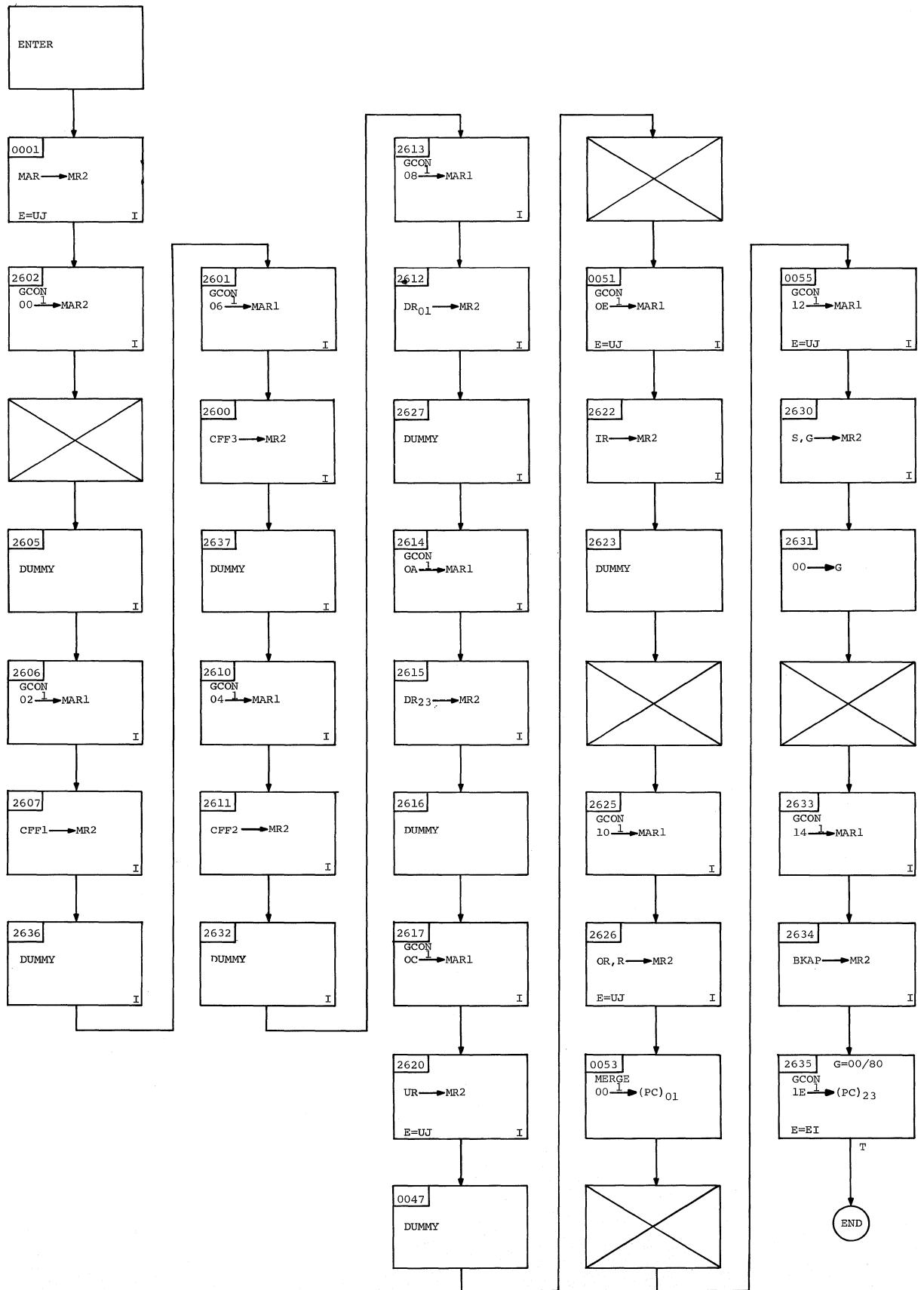
# INTERRUPT



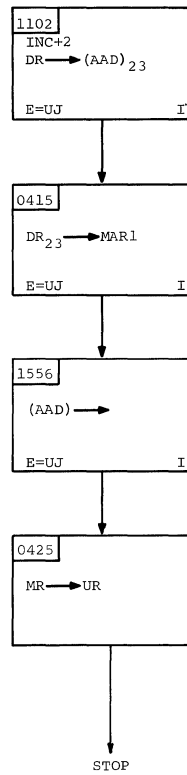
# INTERRUPT



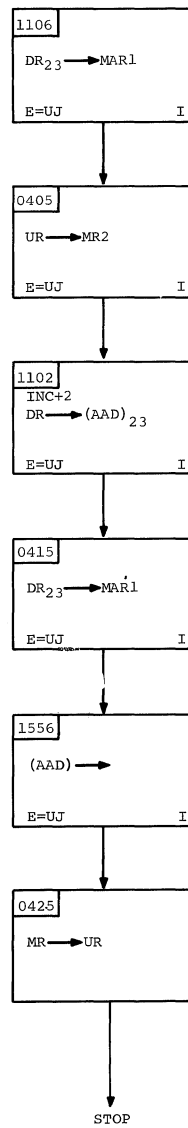




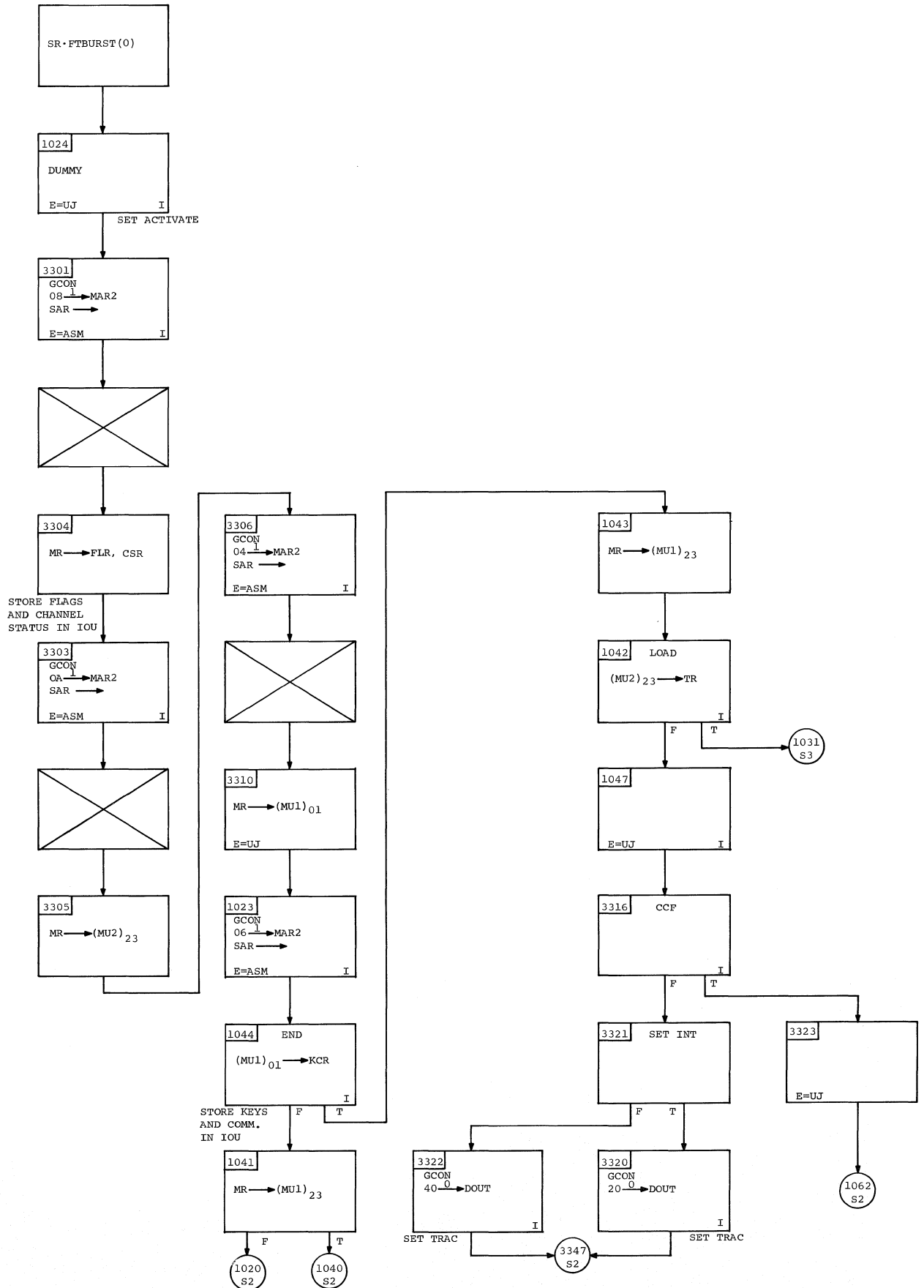
READ HIGH SPEED MEMORY RDM



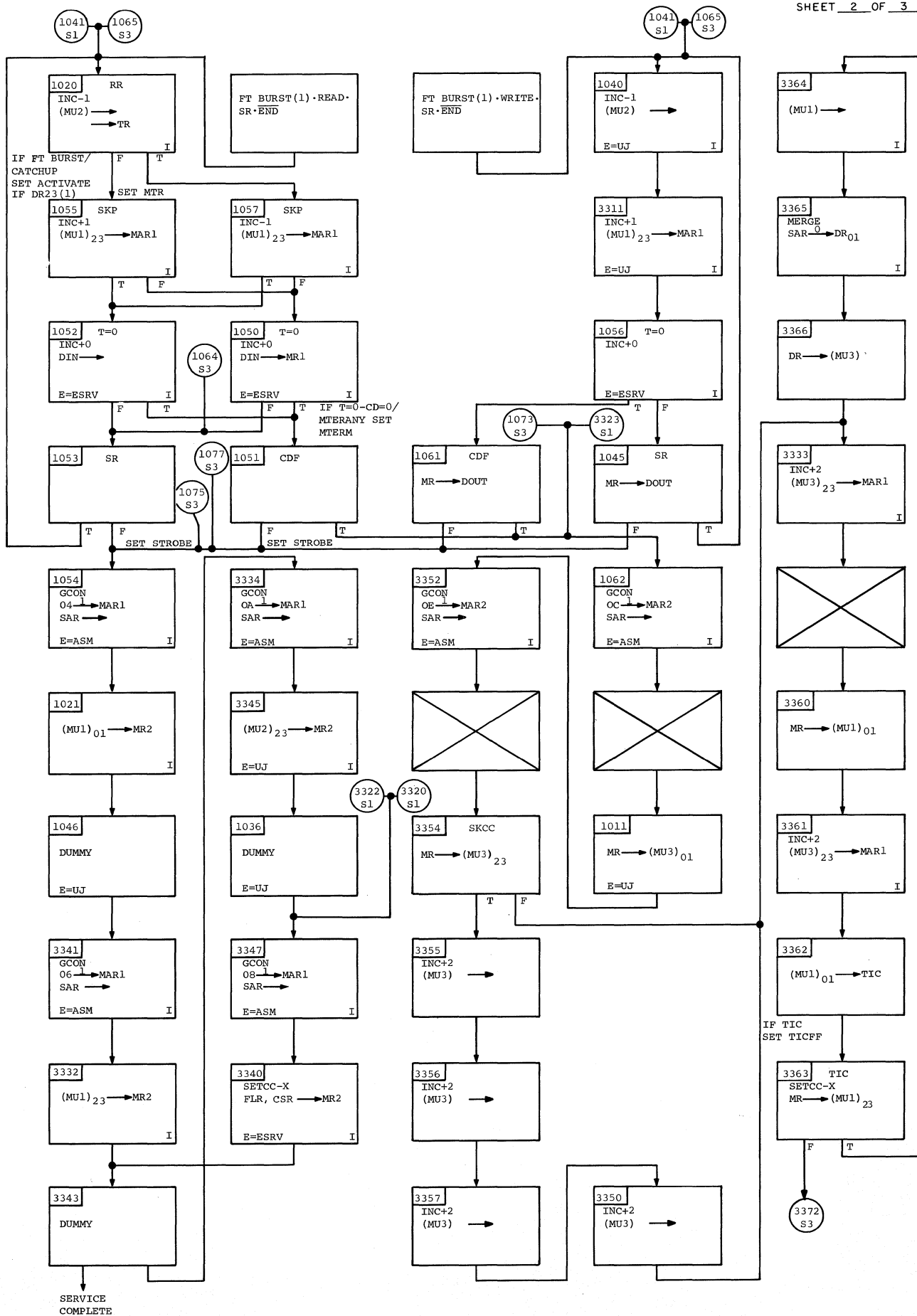
# WRITE HIGH SPEED MEMORY WRM



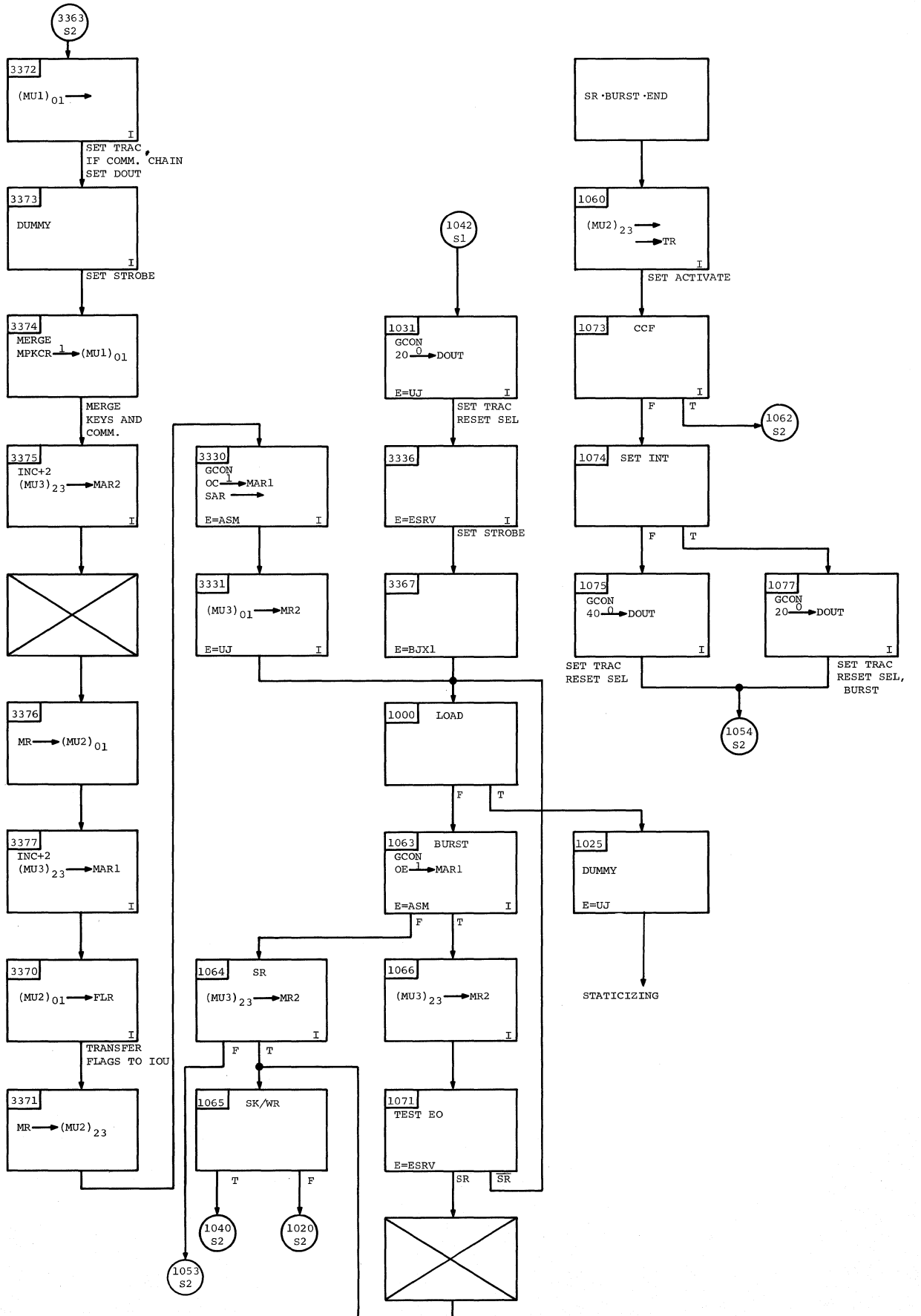
# MULTIPLEXOR SERVICE MUXS



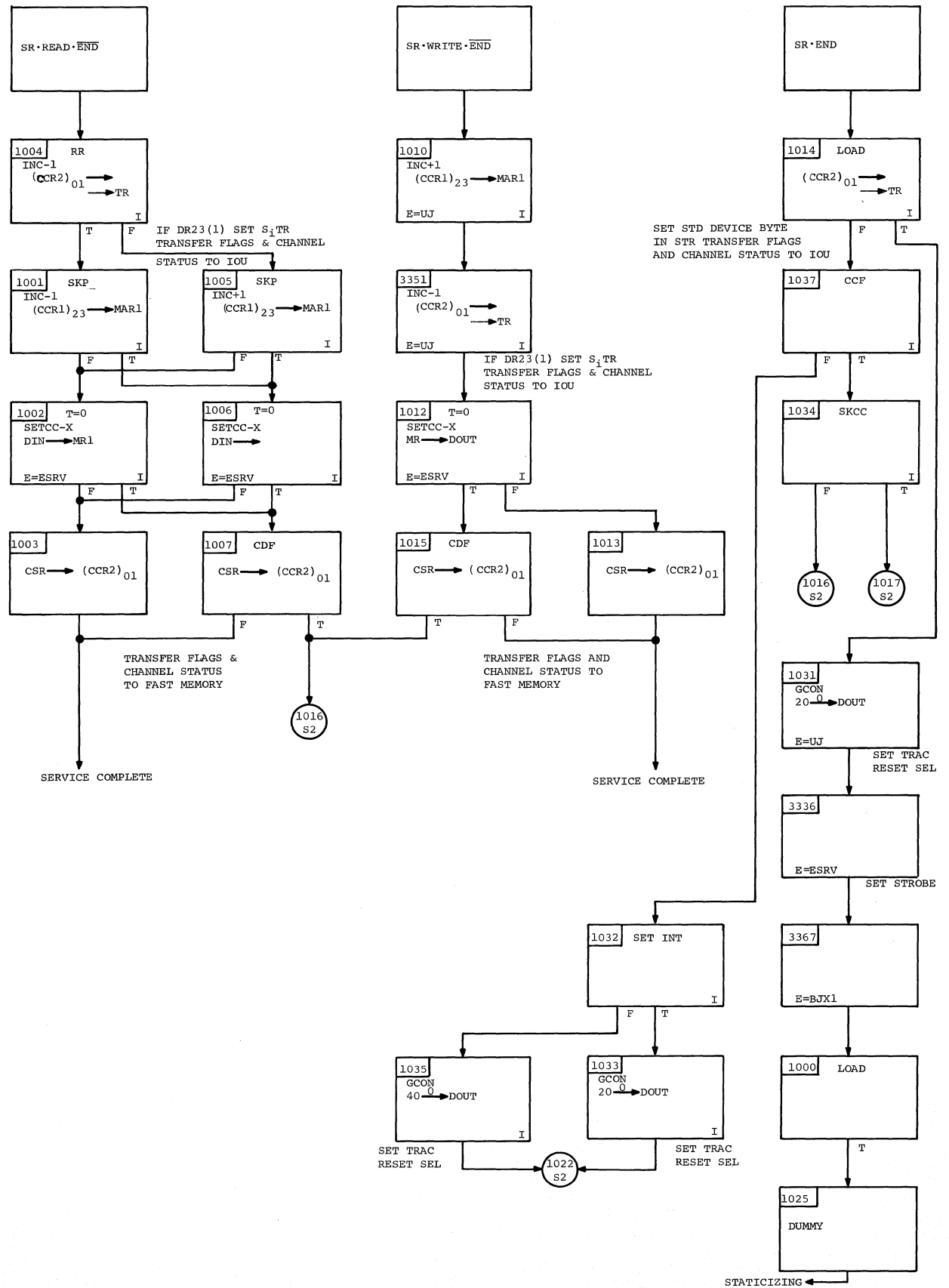
# MULTIPLEXOR SERVICE MUXS



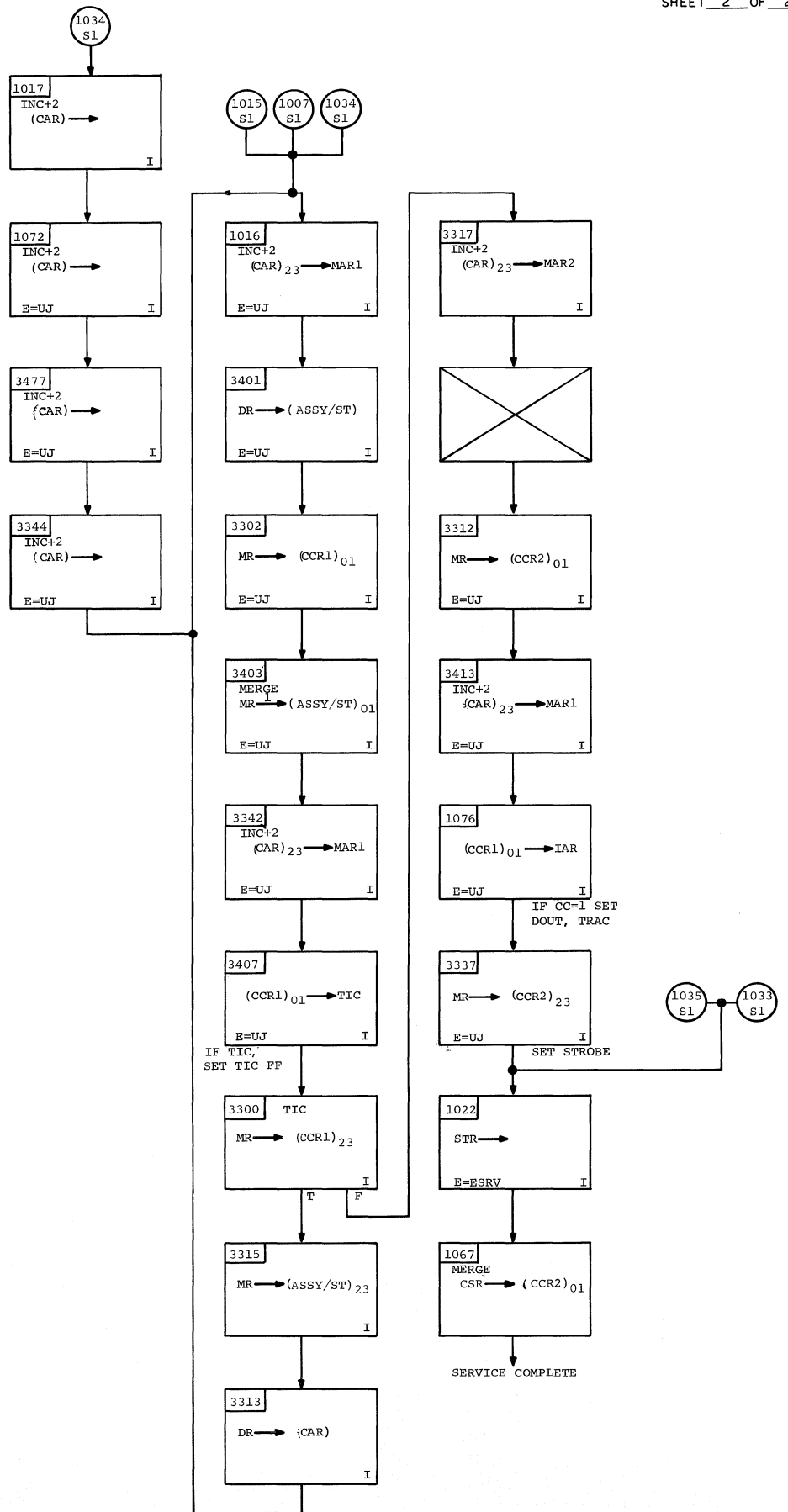
MULTIPLEXOR SERVICE MUXS



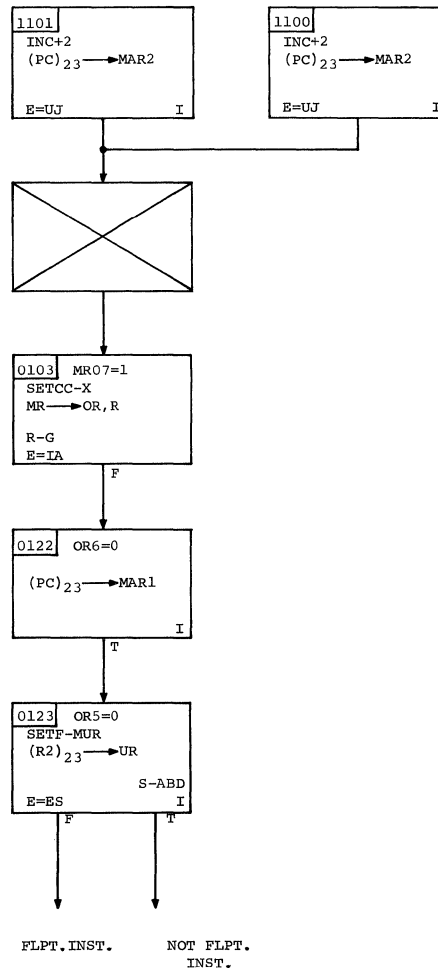
# SELECTOR SERVICE SELS



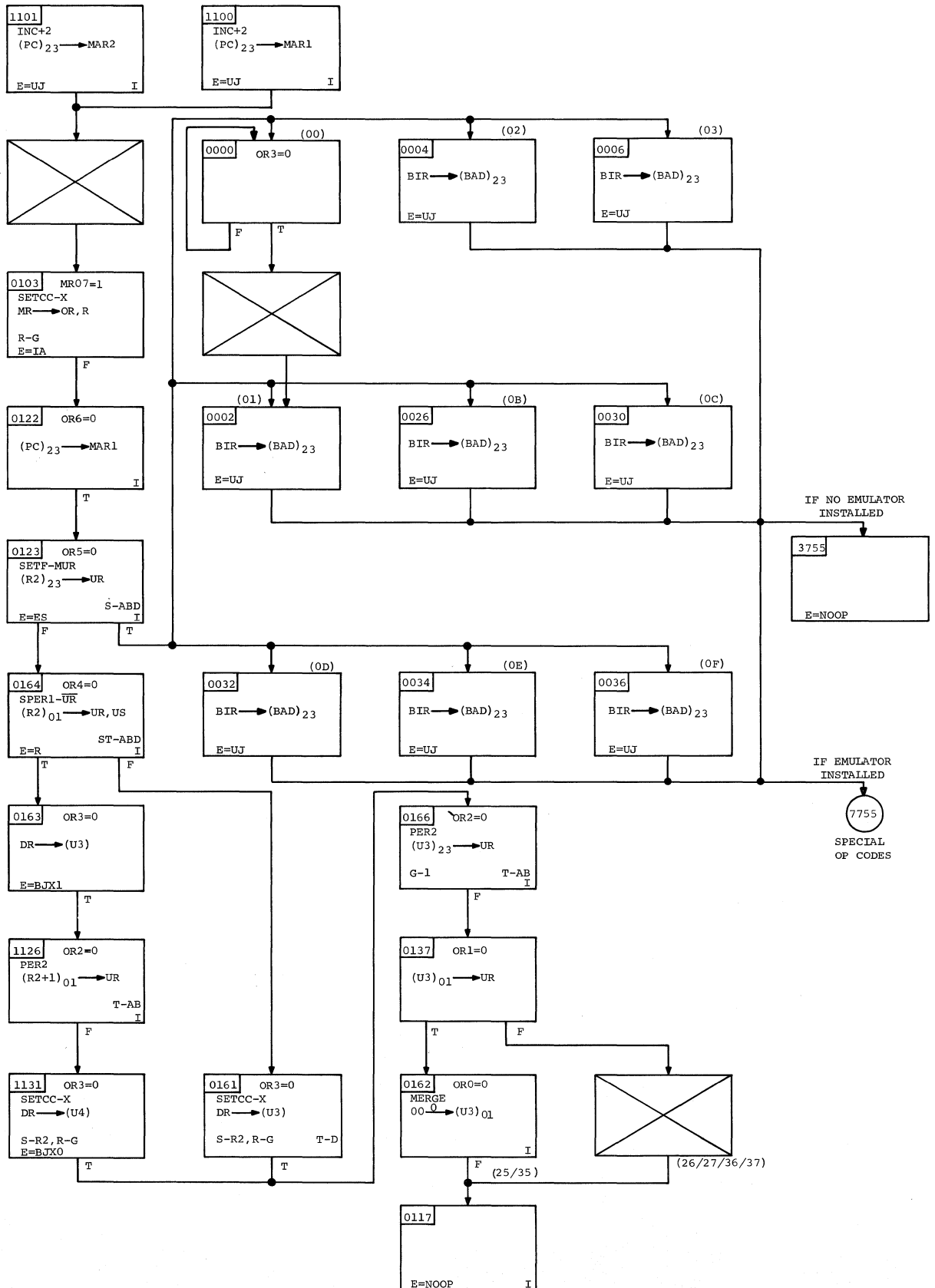
# SELECTOR SERVICE SELS

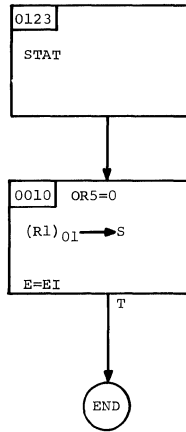


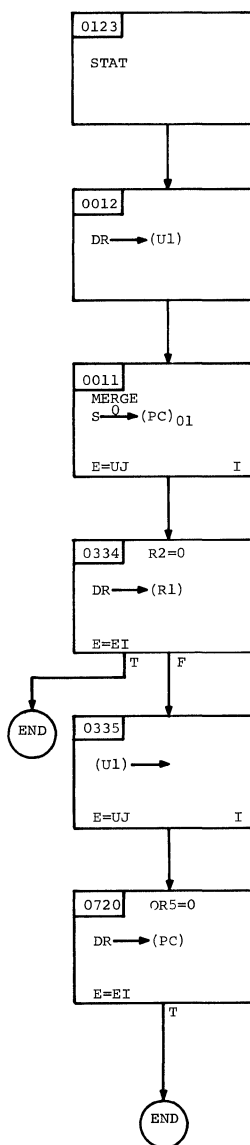


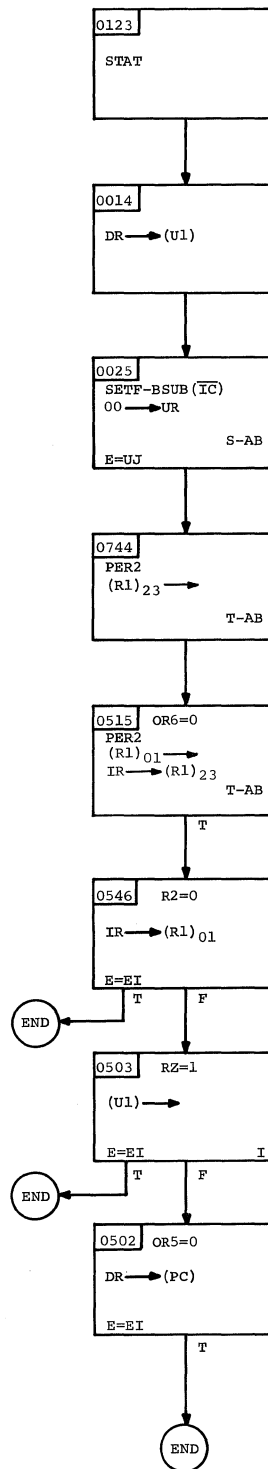


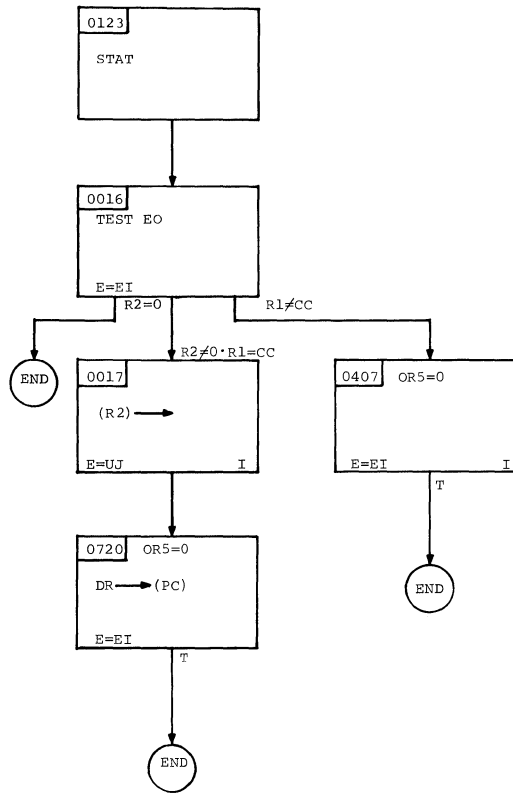
# NO OPERATION INSTRUCTIONS RR FORMAT

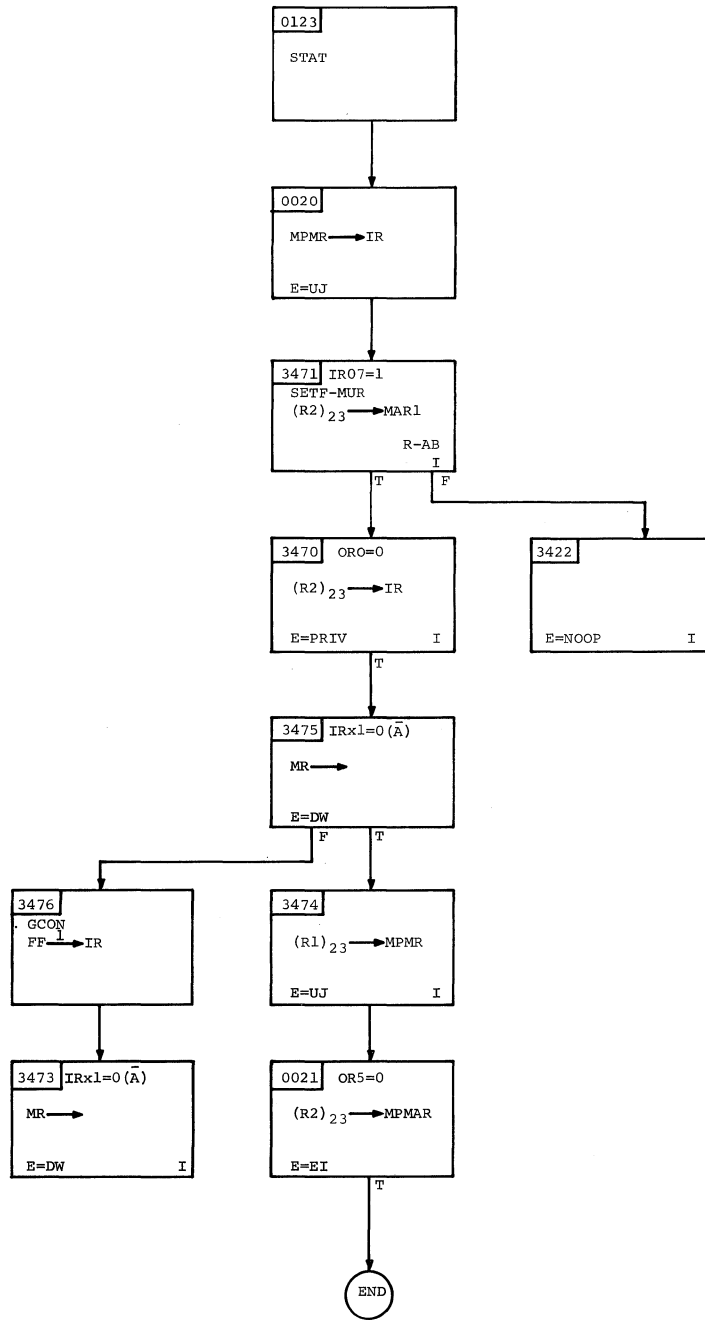


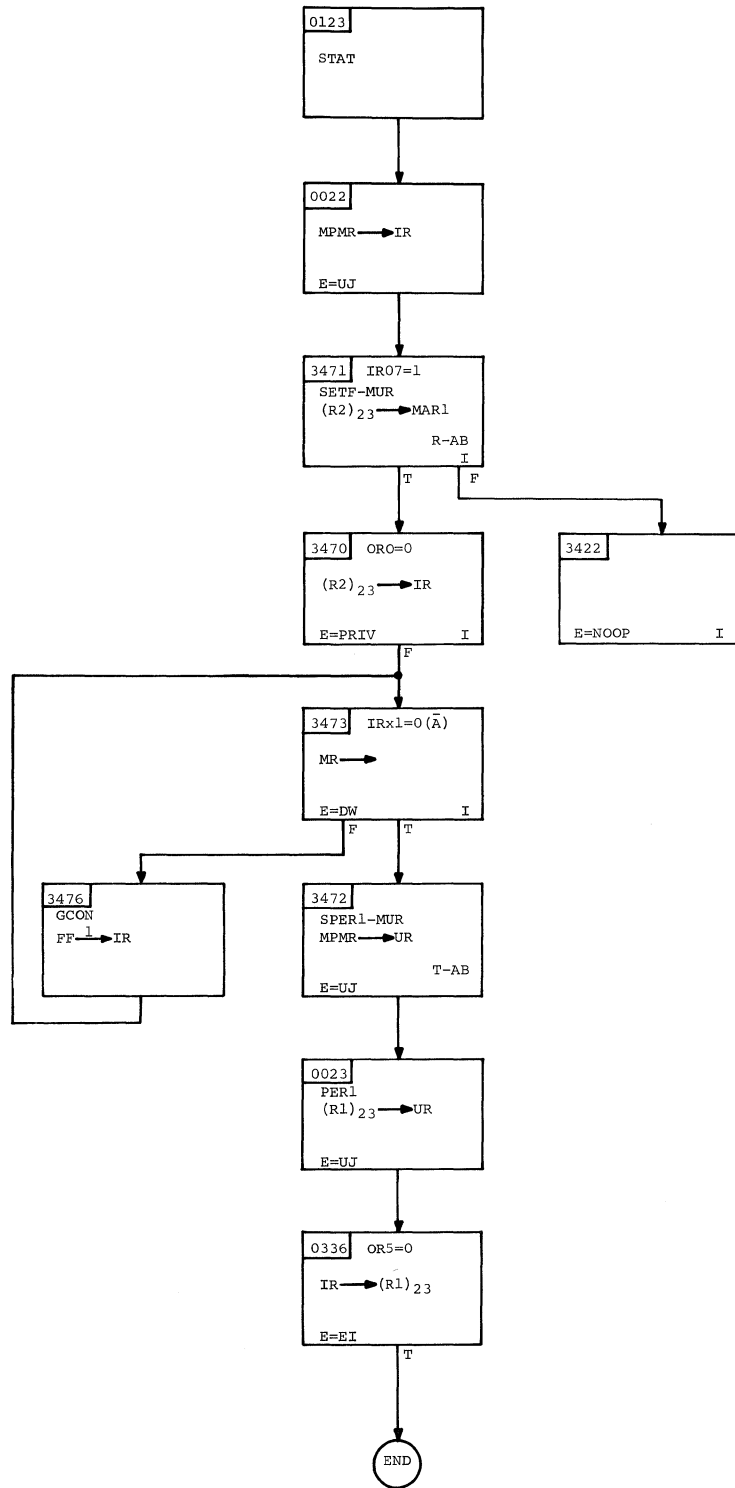




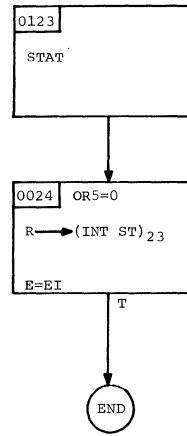


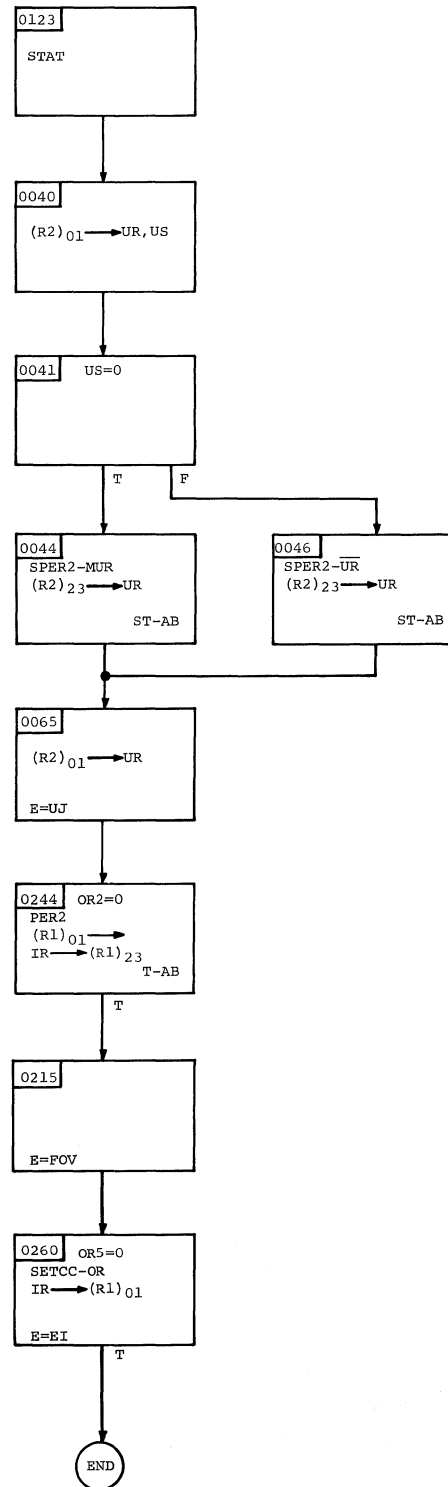


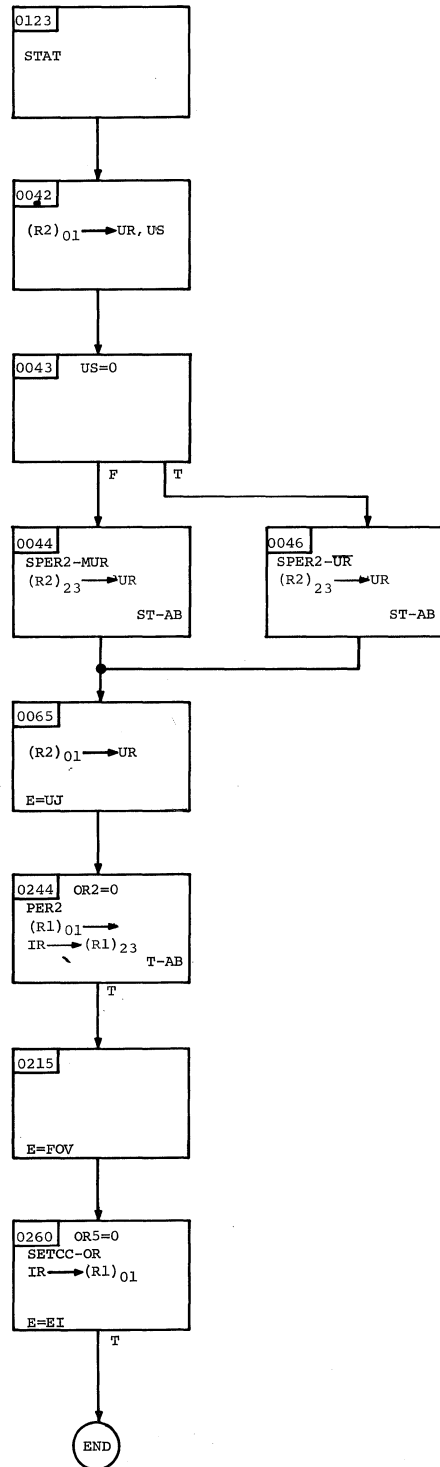


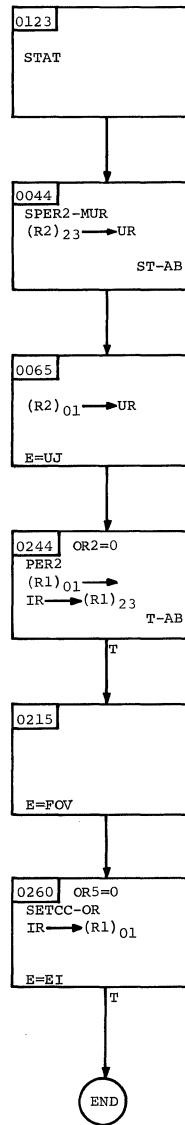


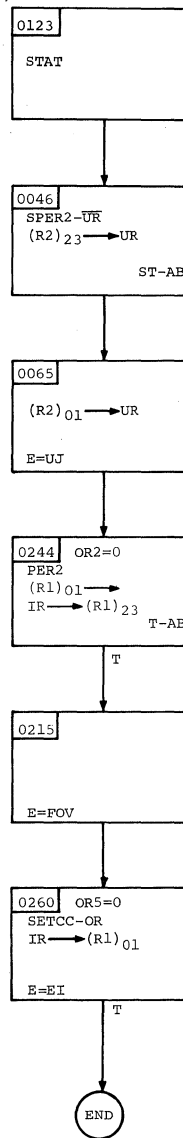


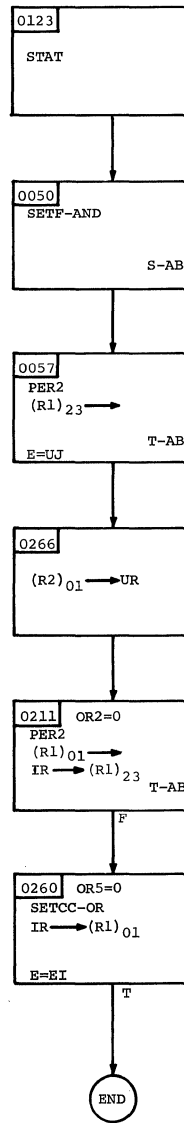


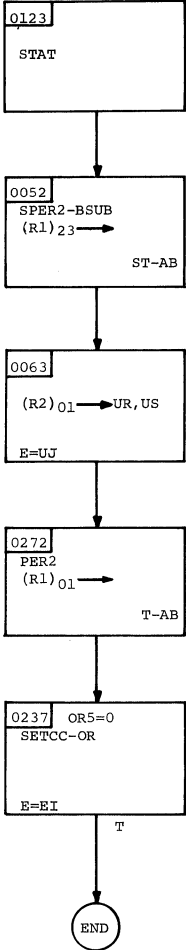


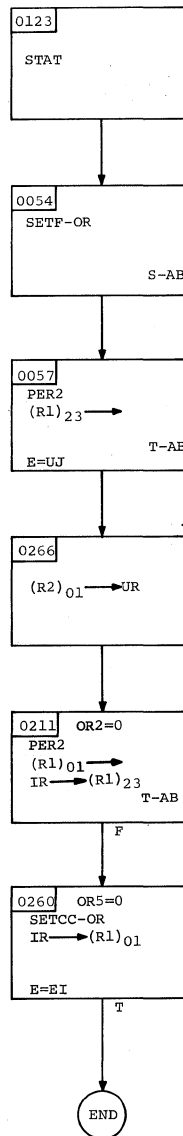




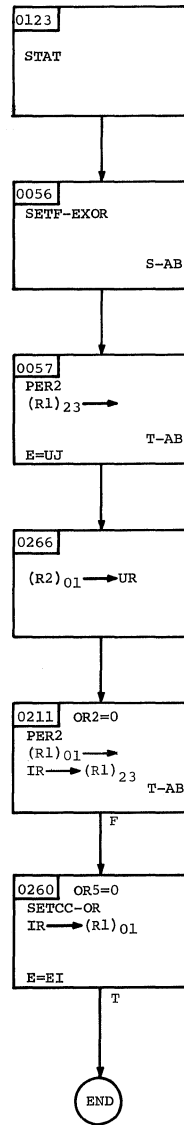


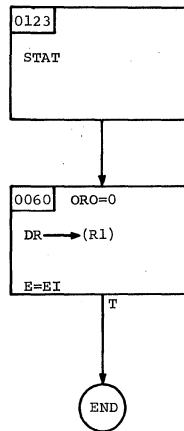


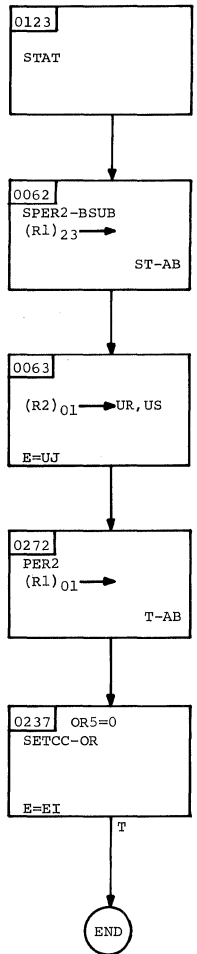


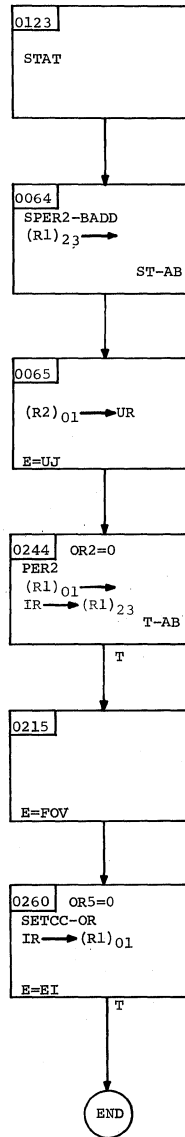


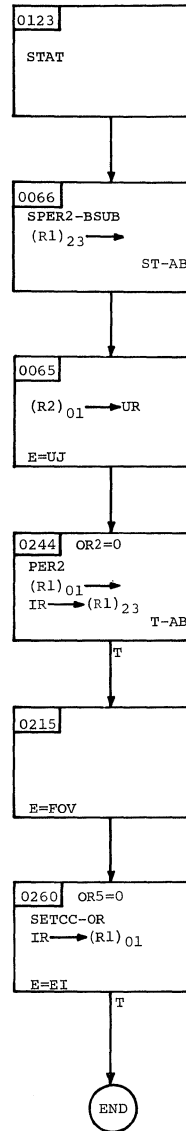


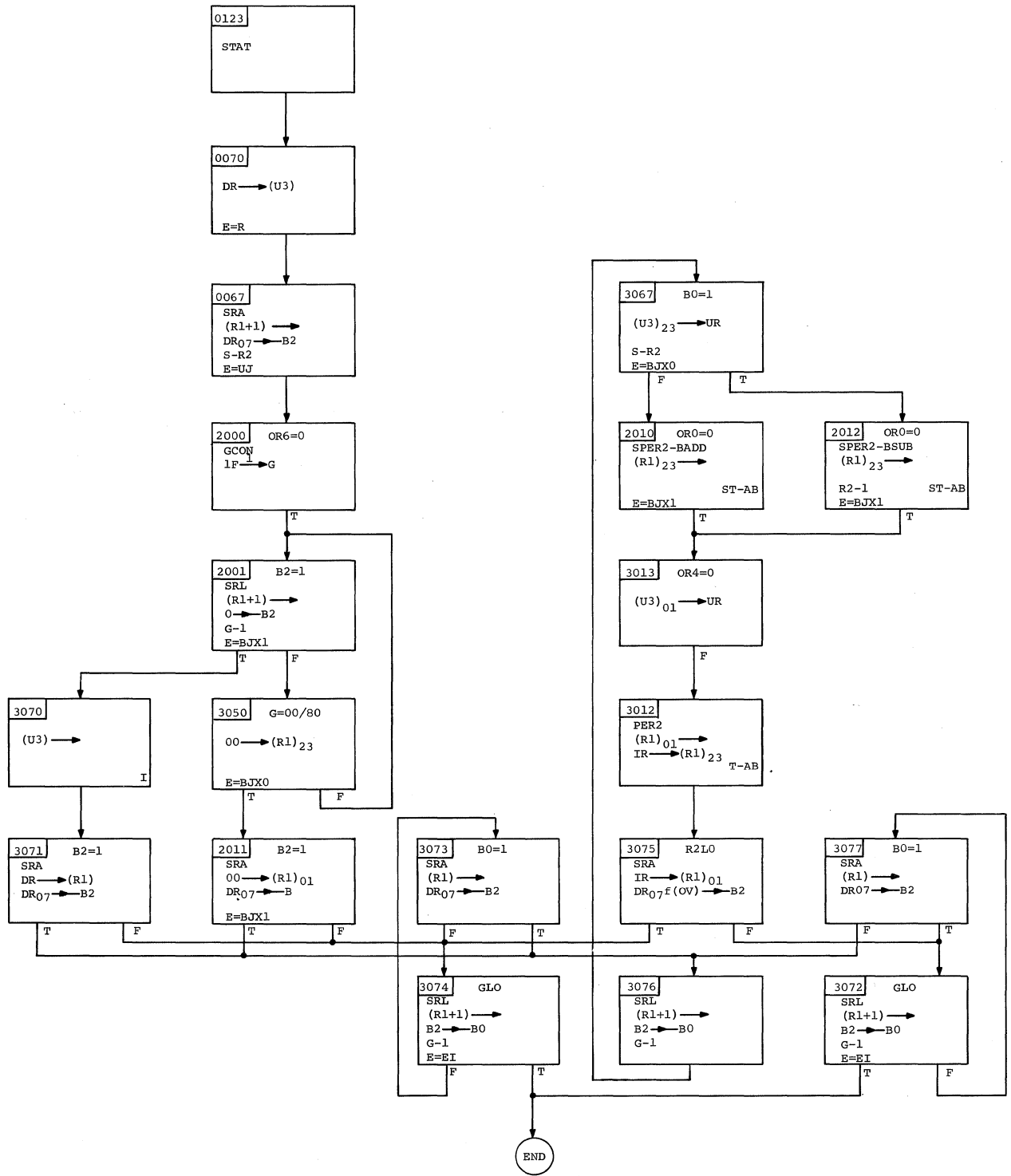


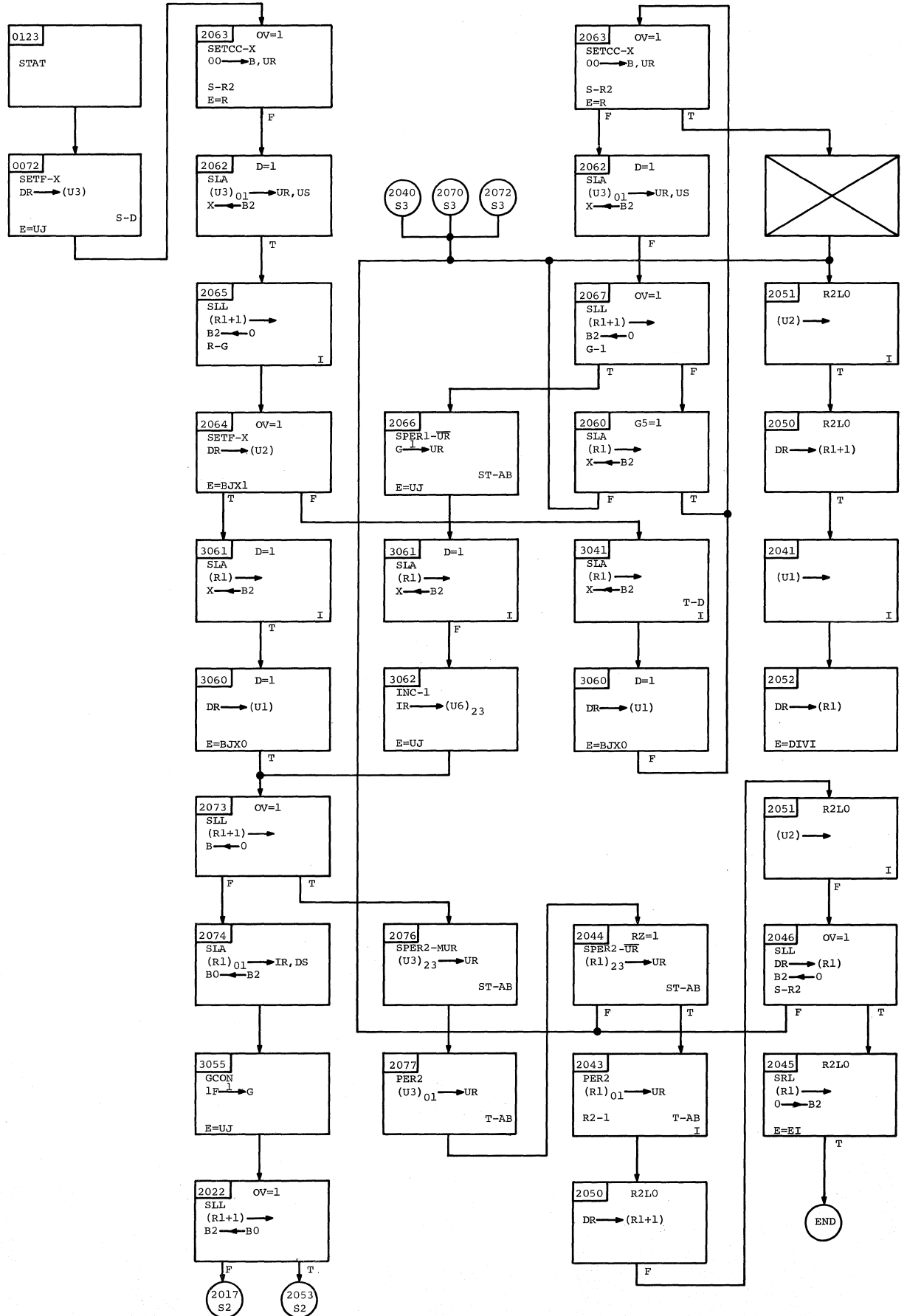


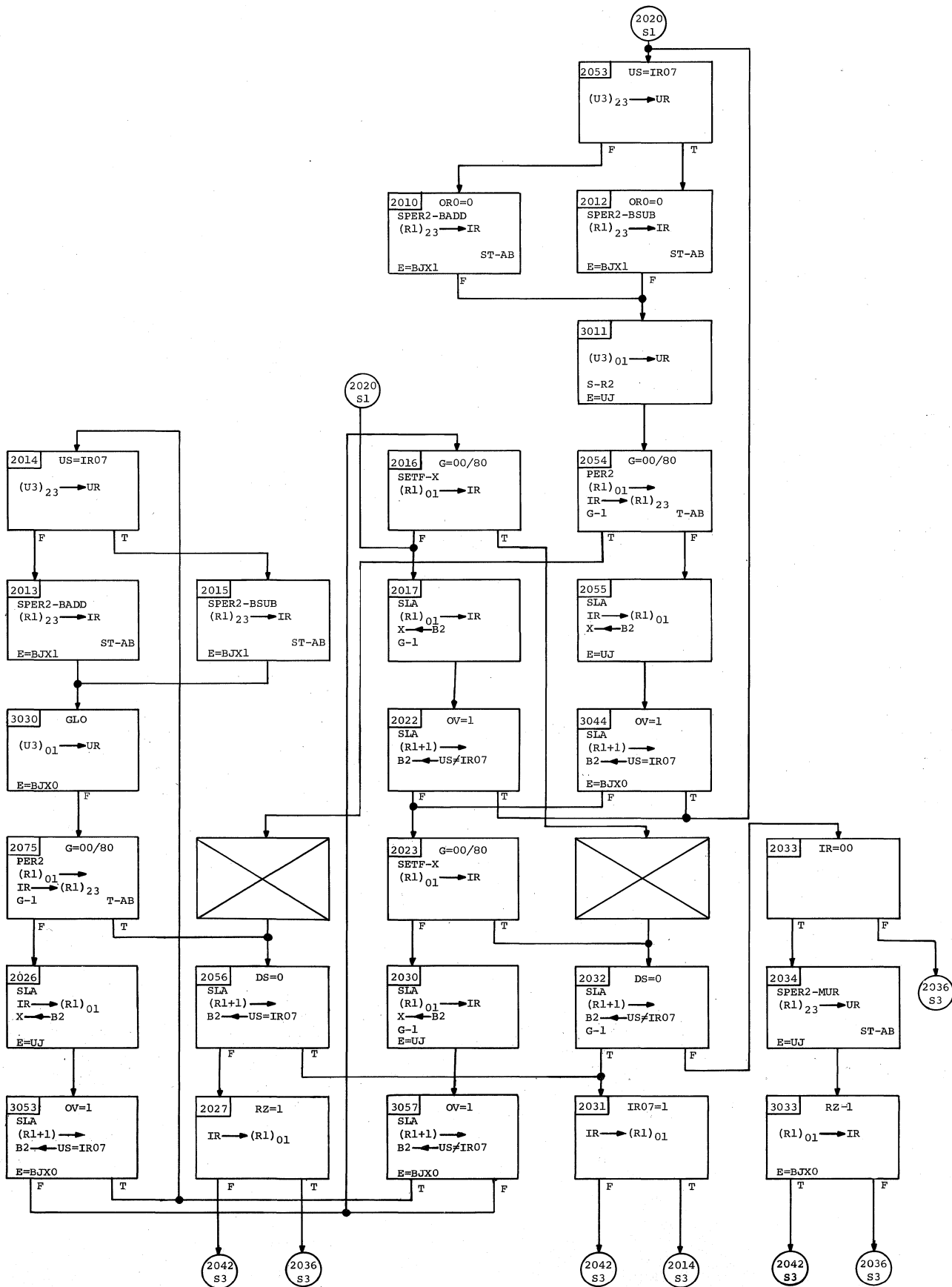




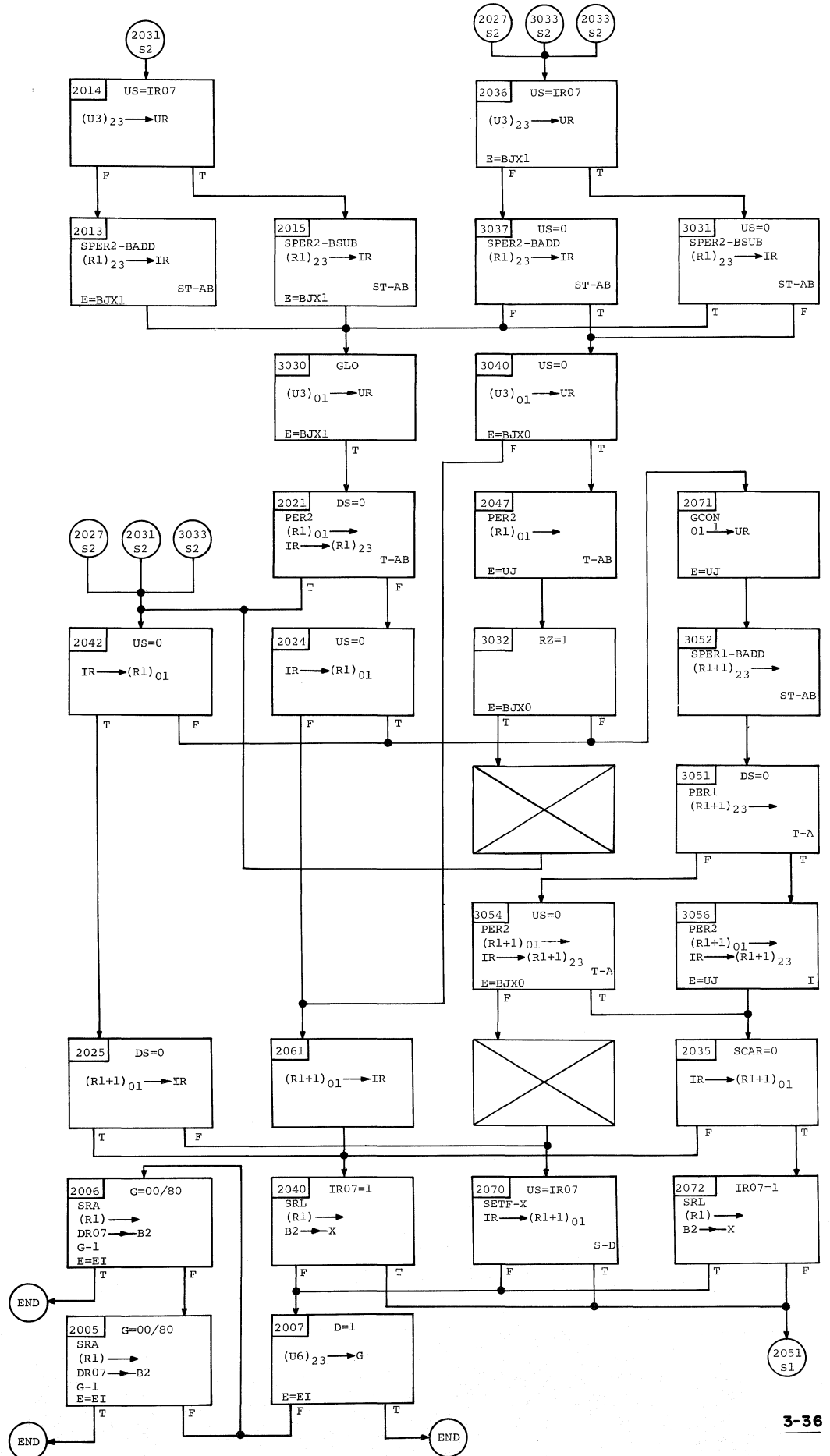


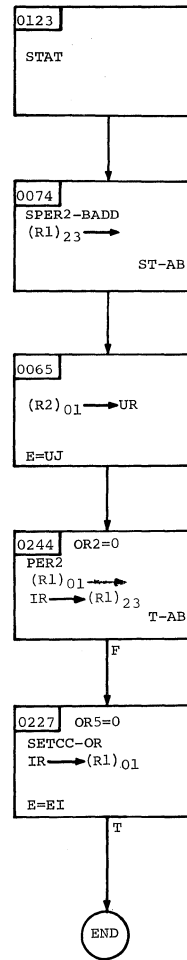


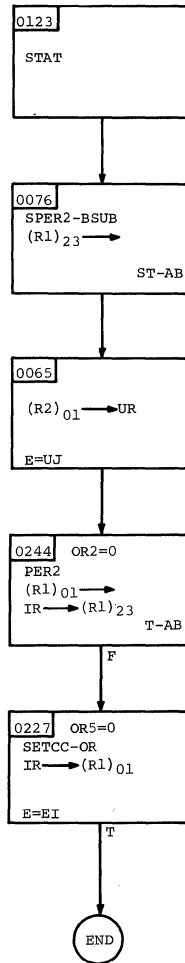


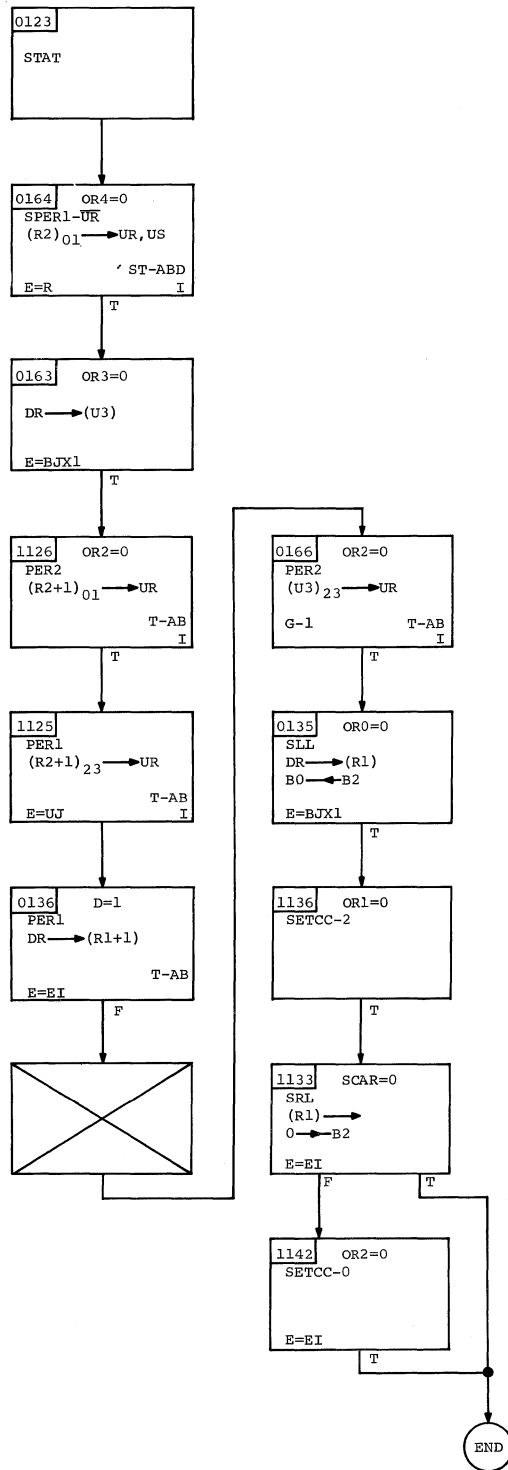


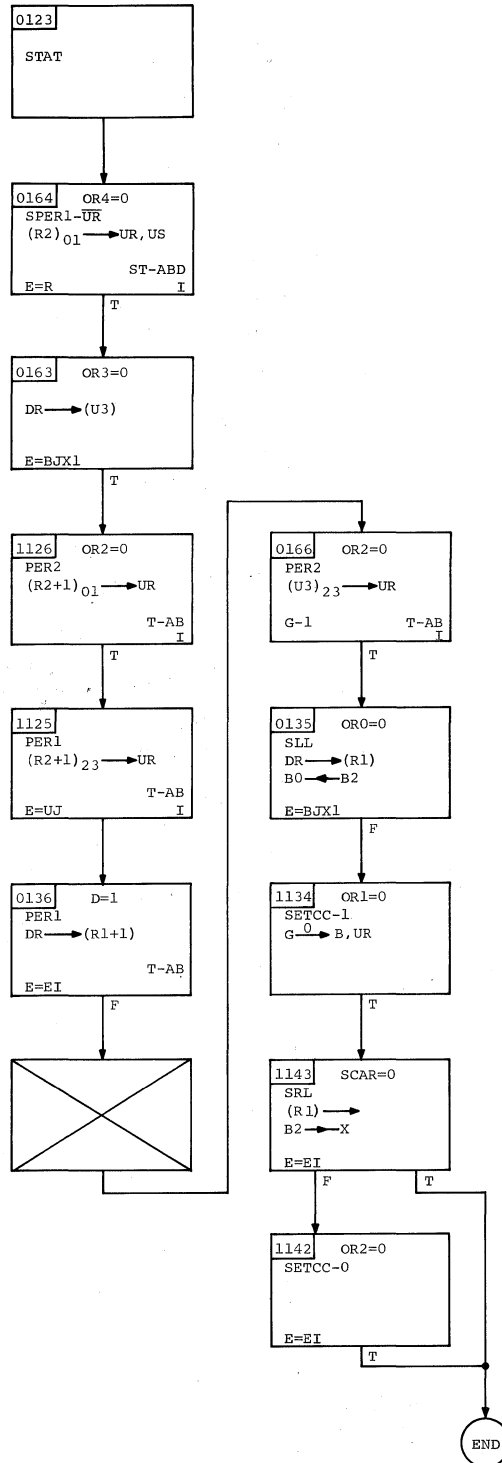


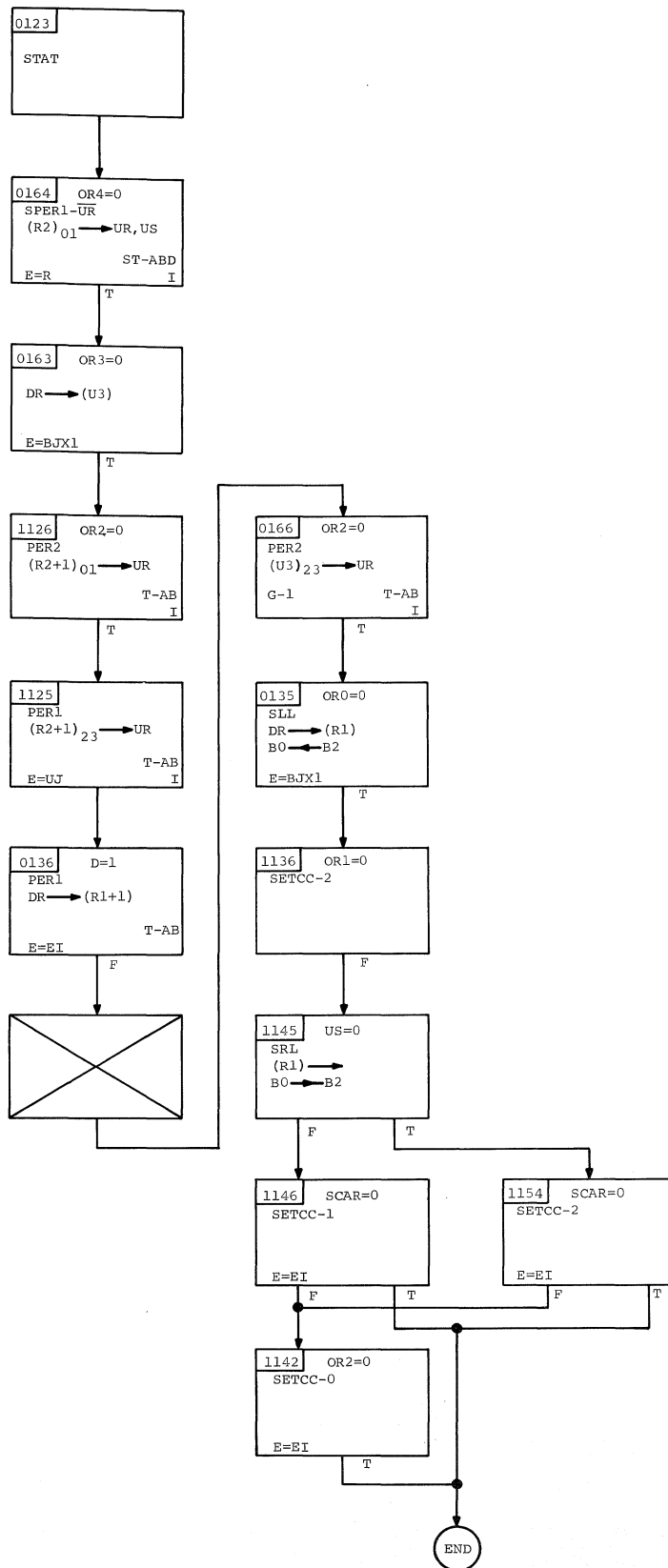


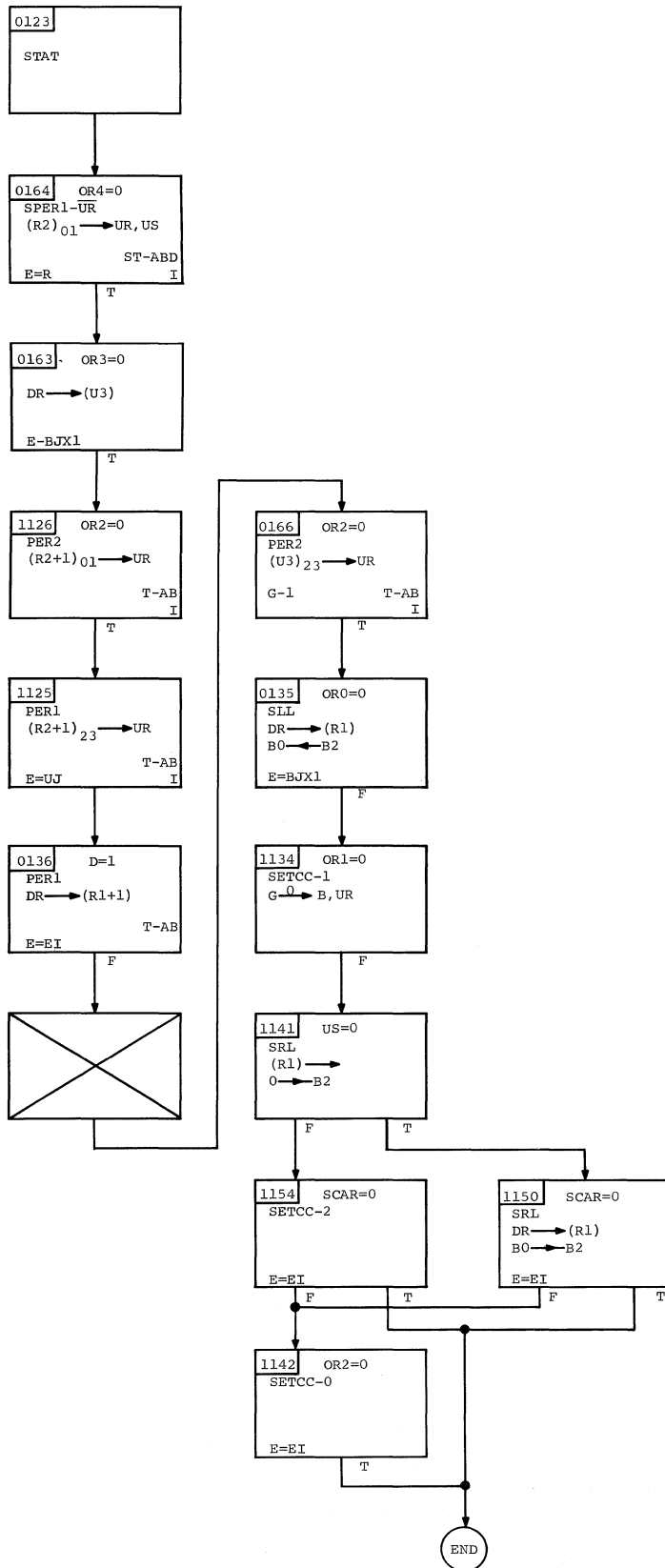


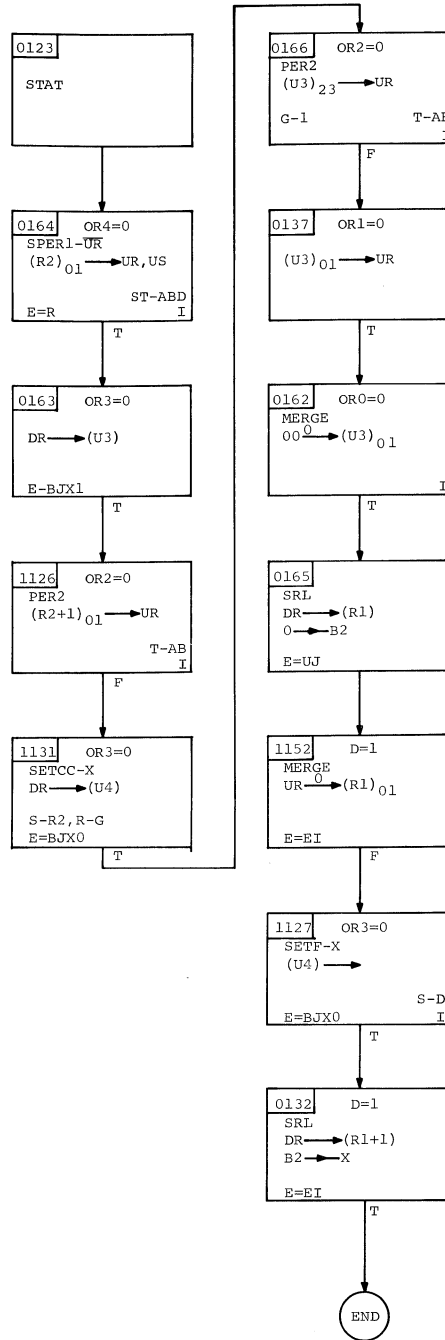




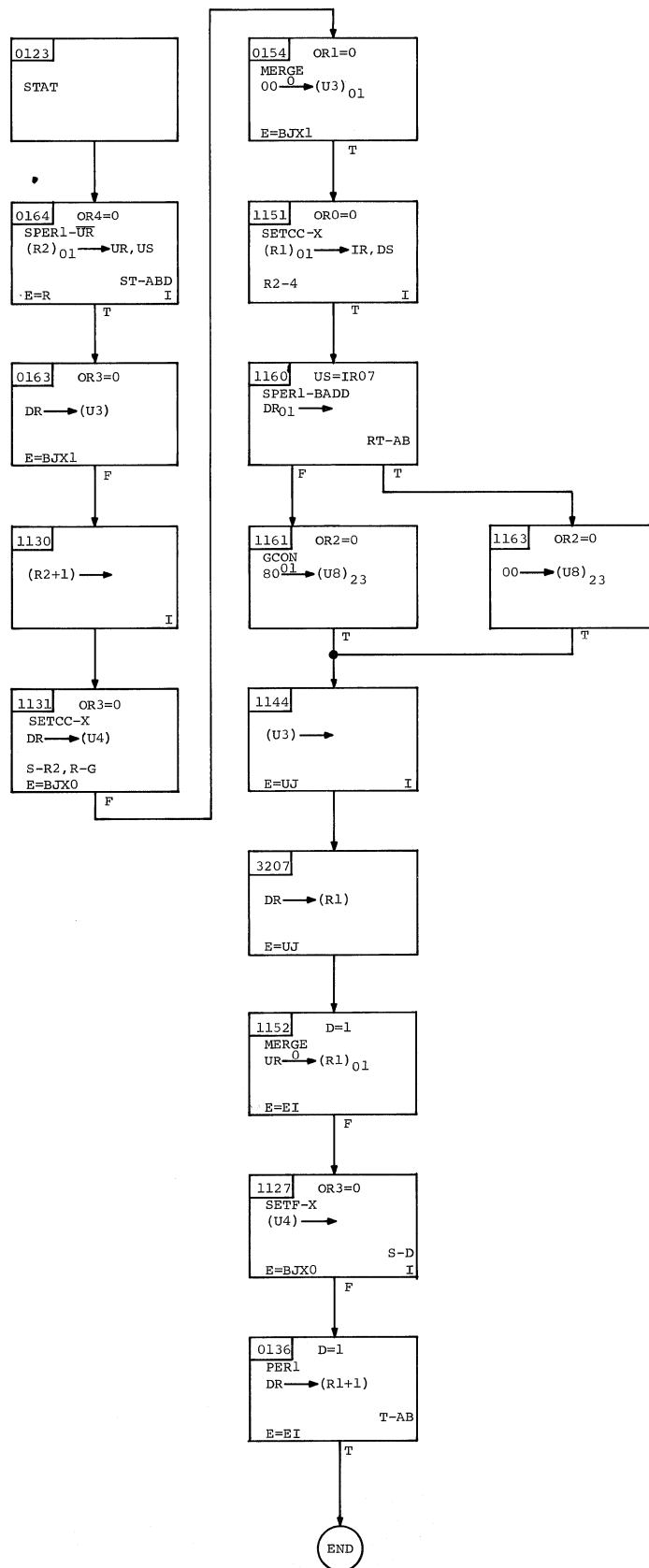


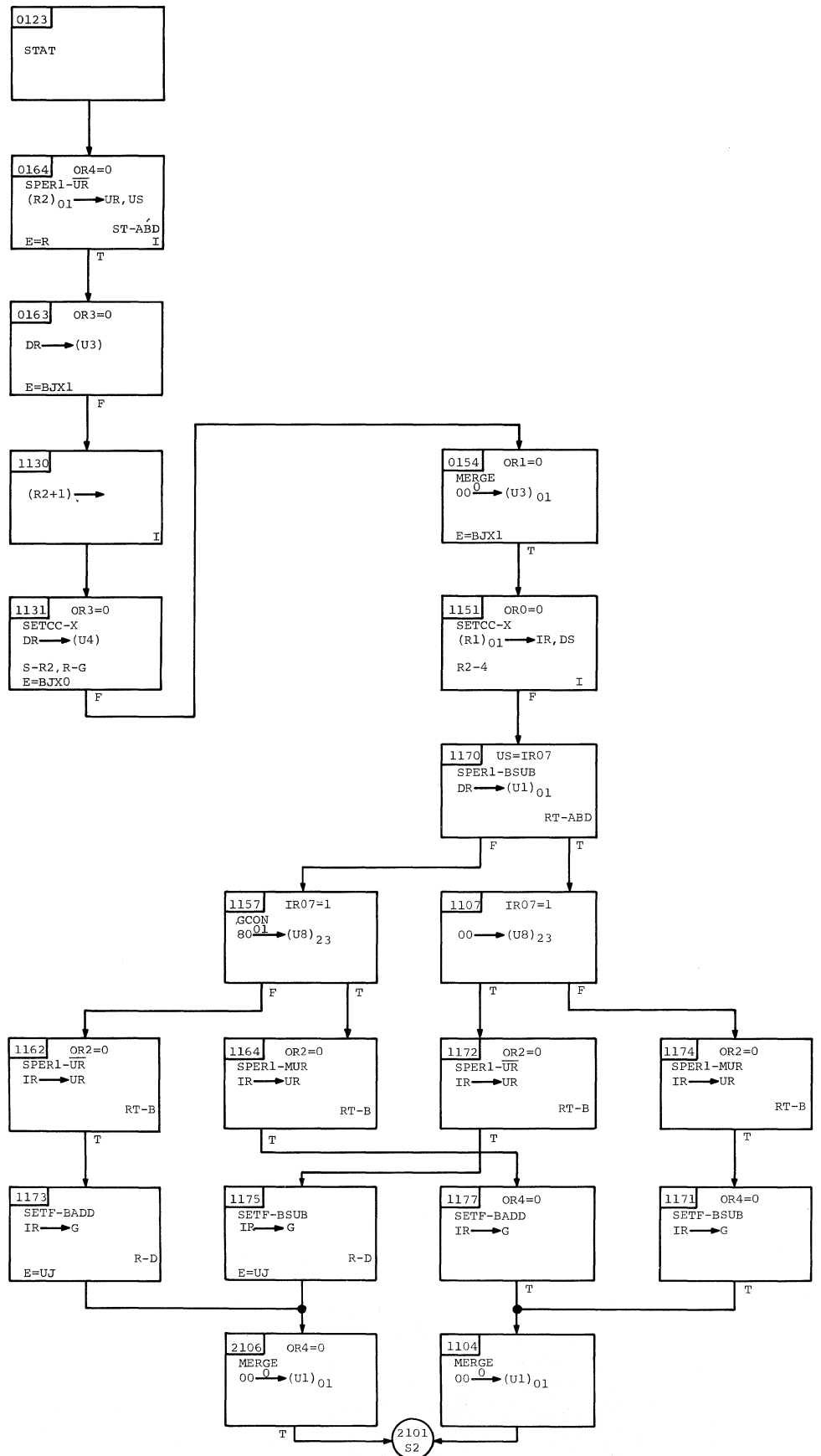


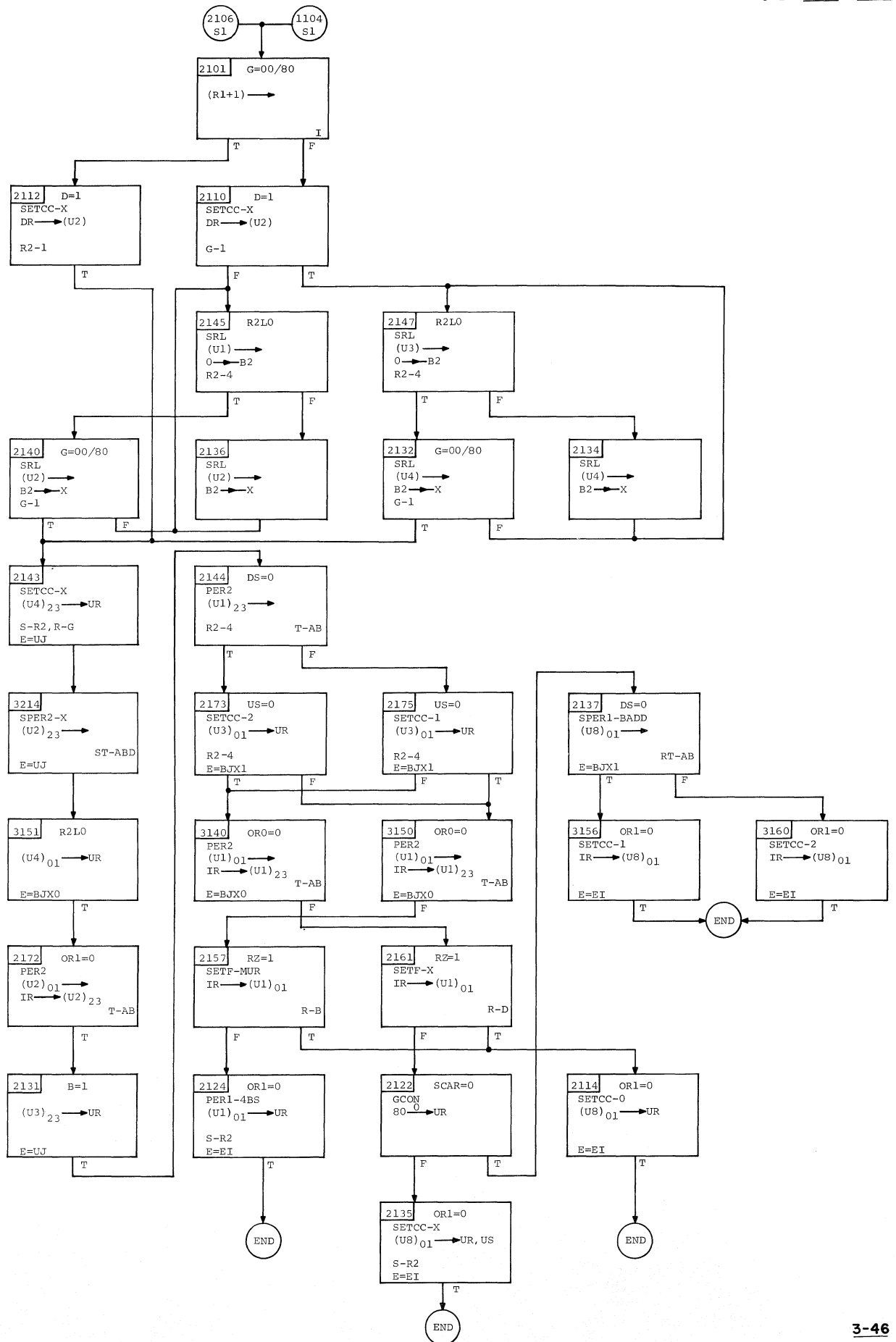


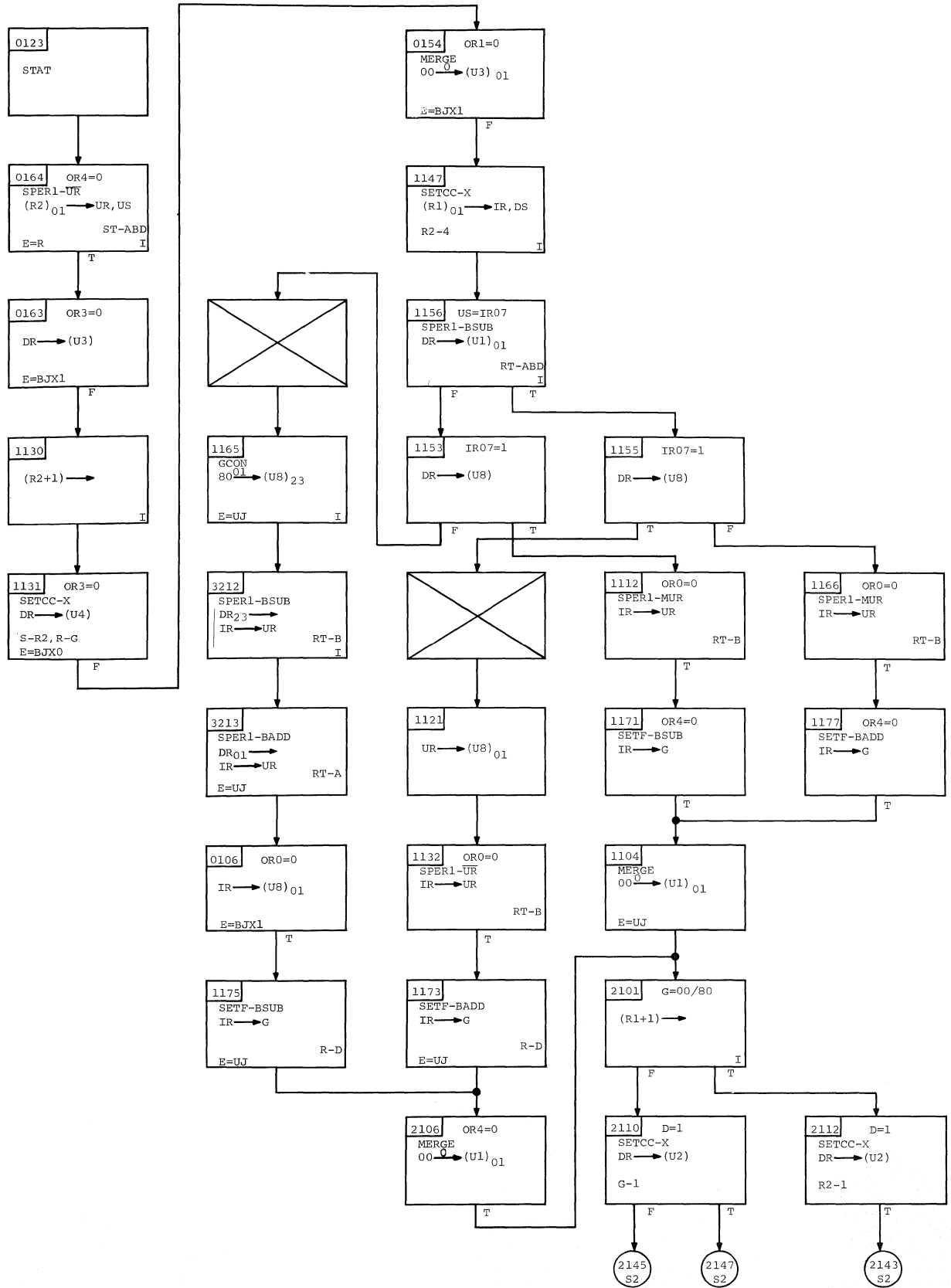


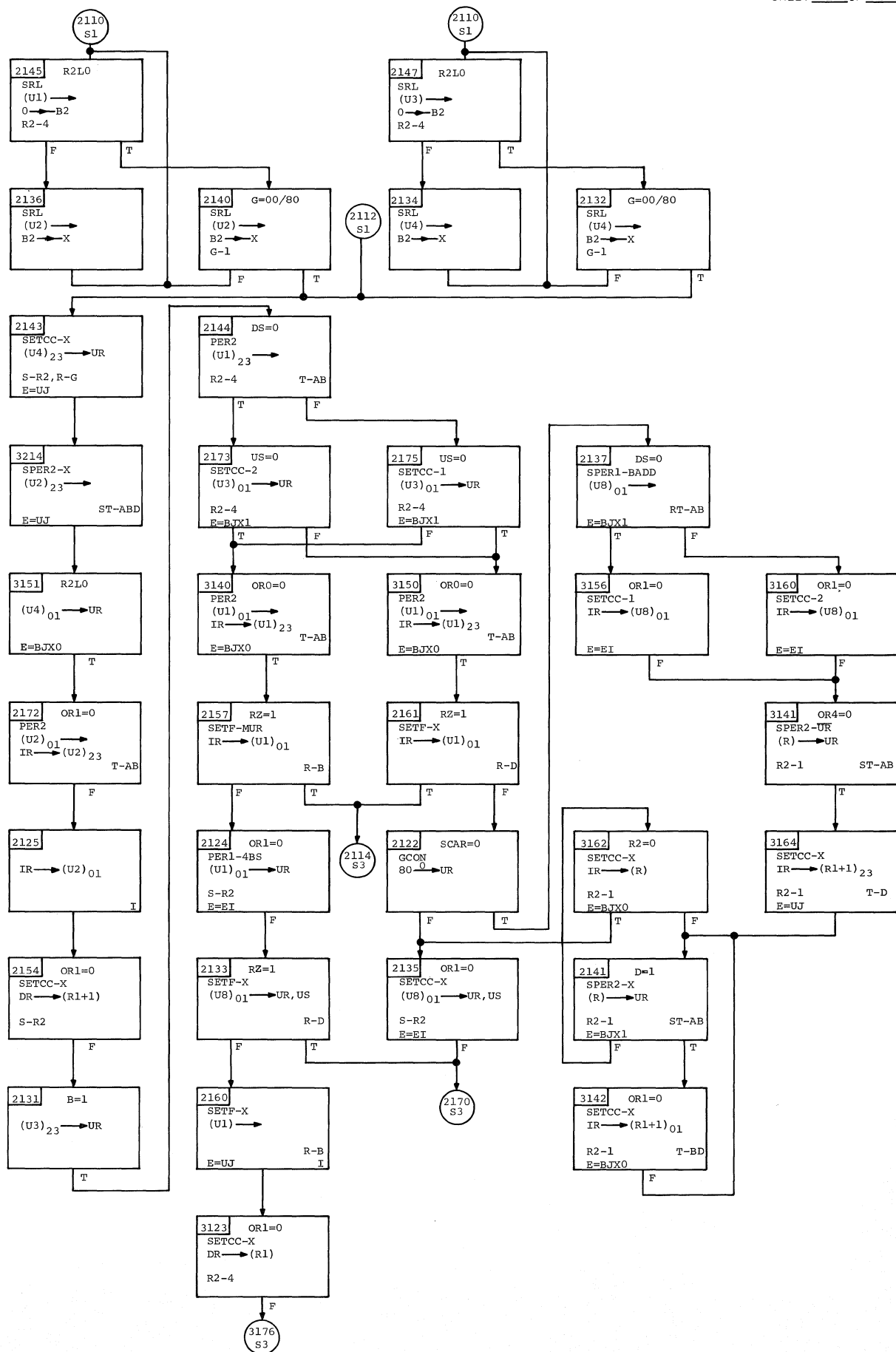




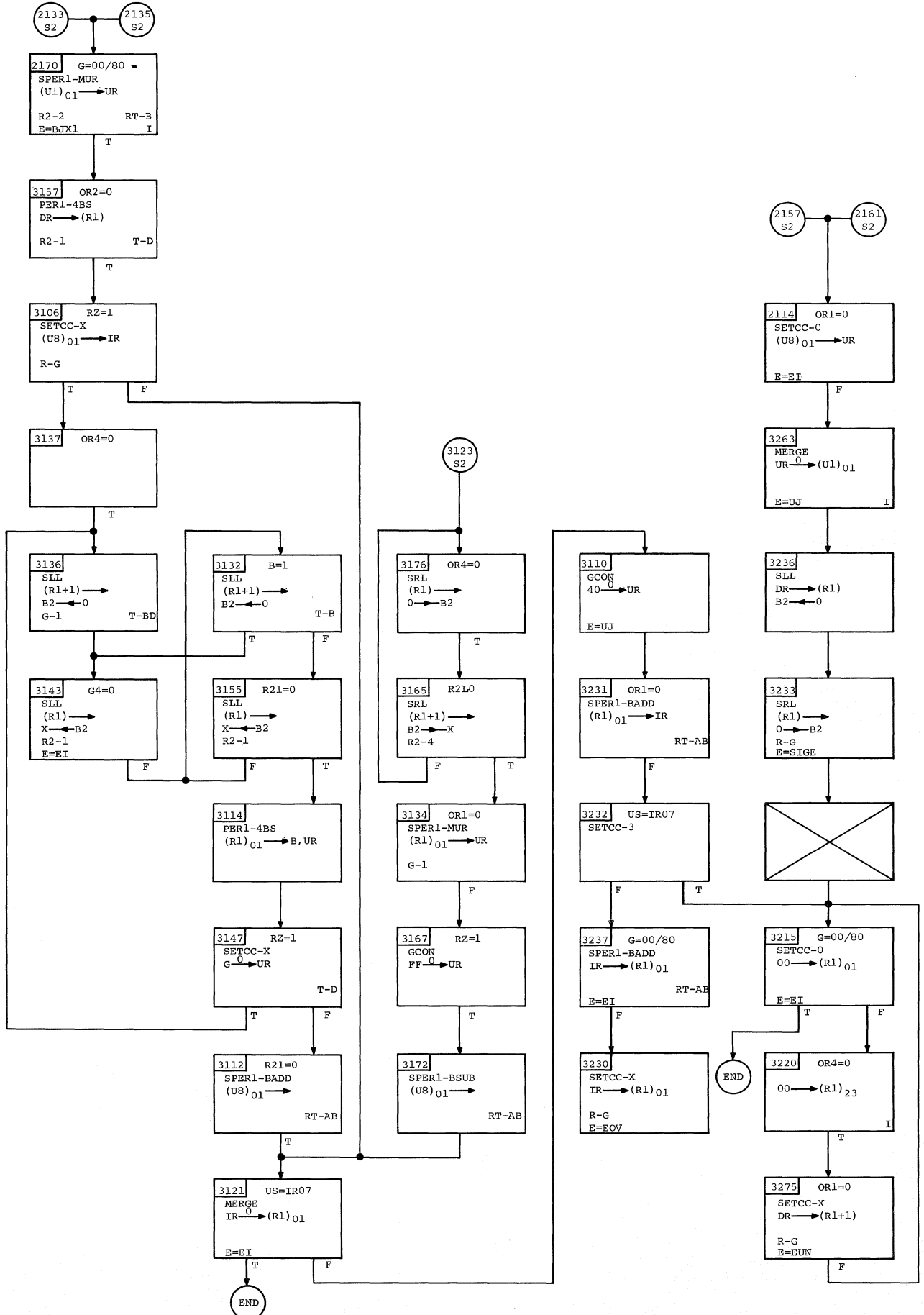


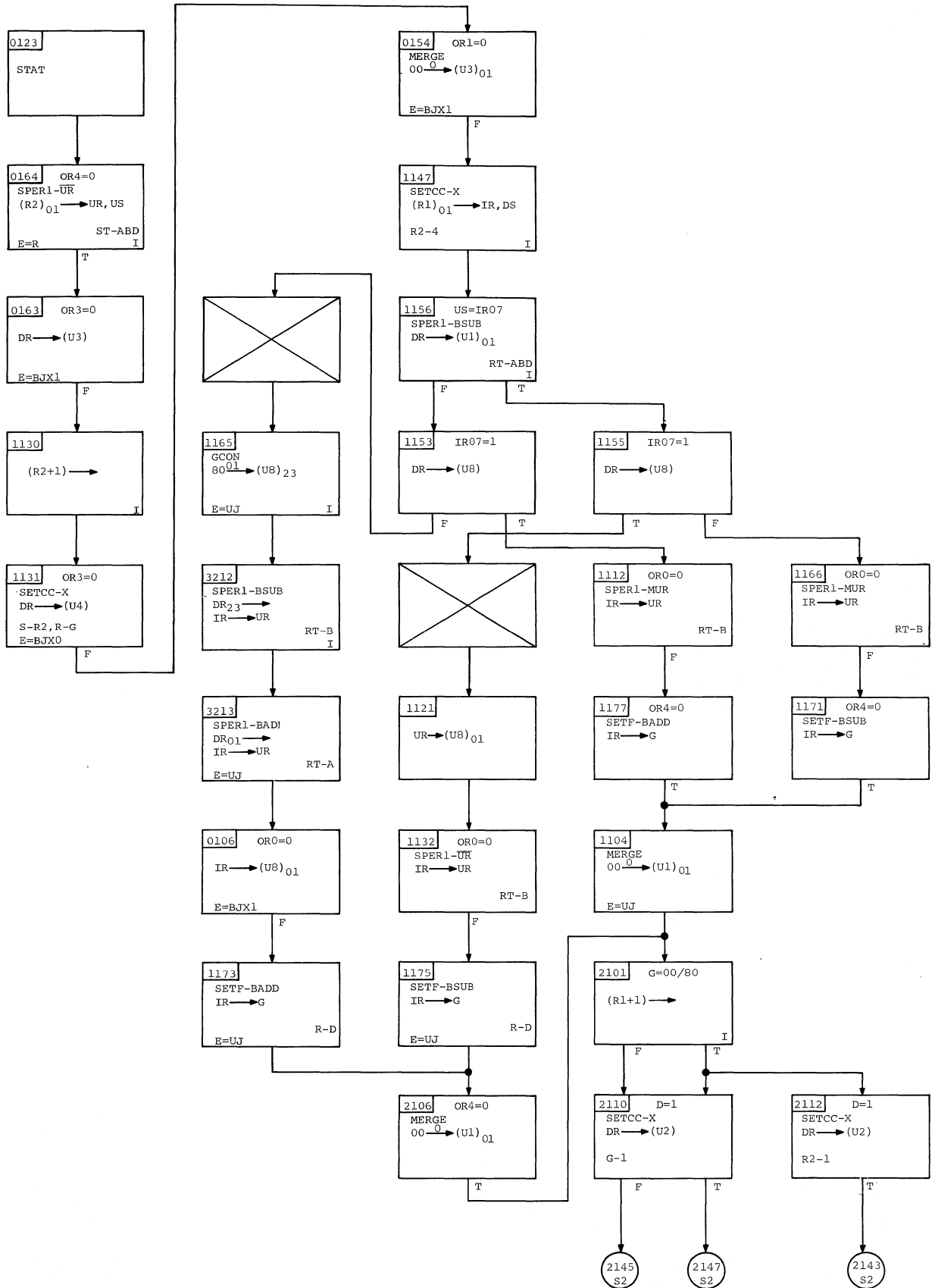




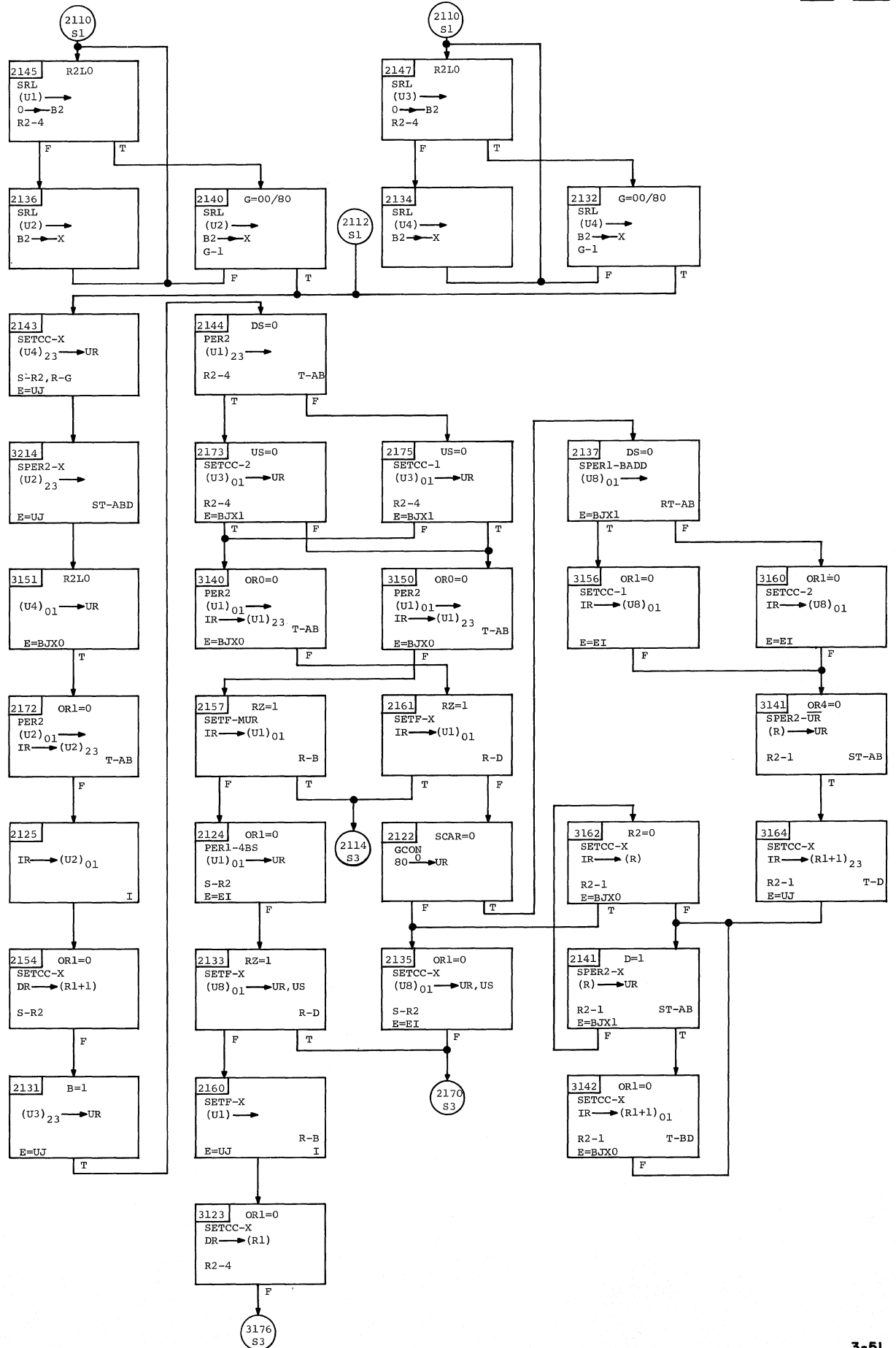


ADD NORMALIZED LONG ANDR RR 2A

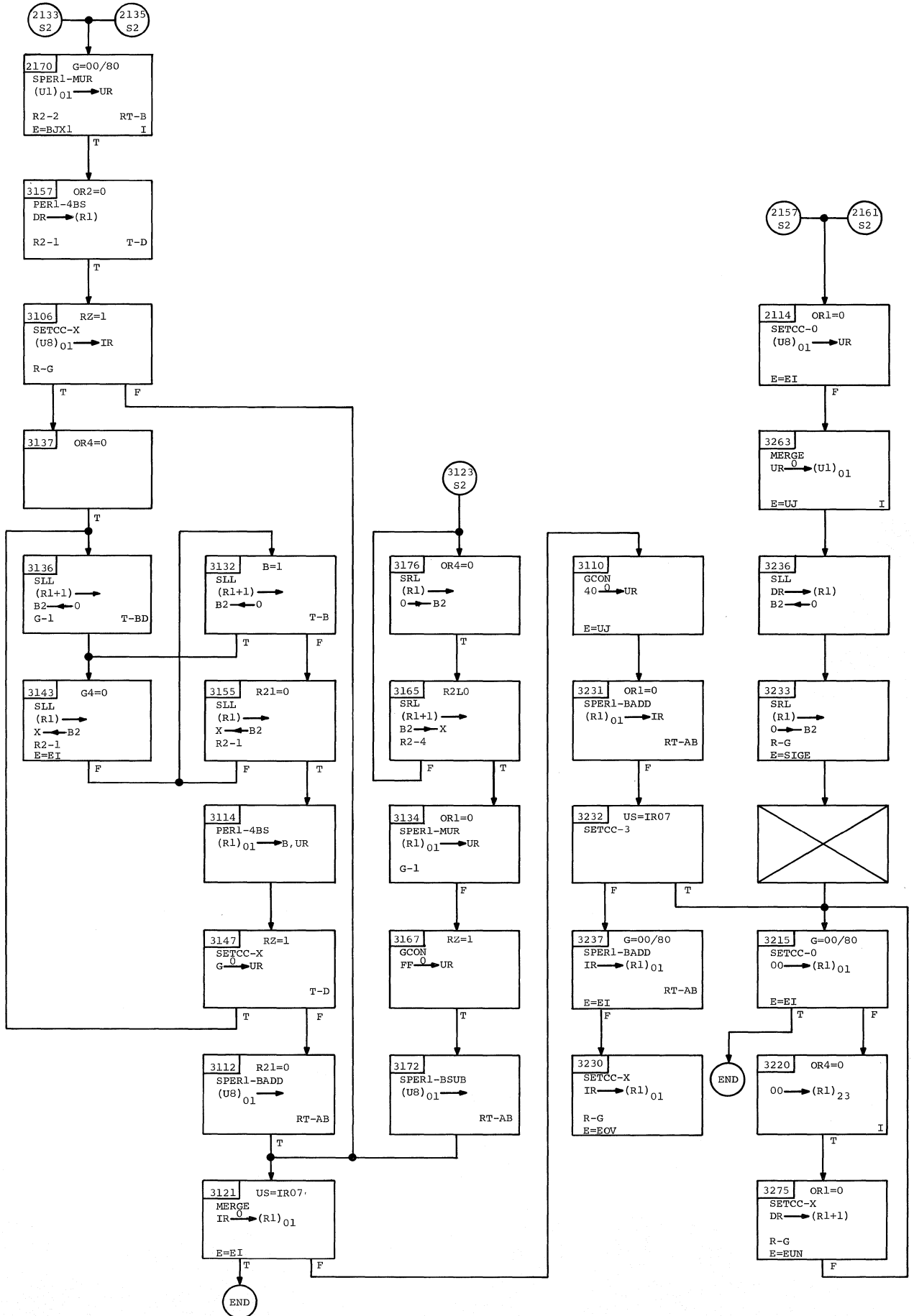


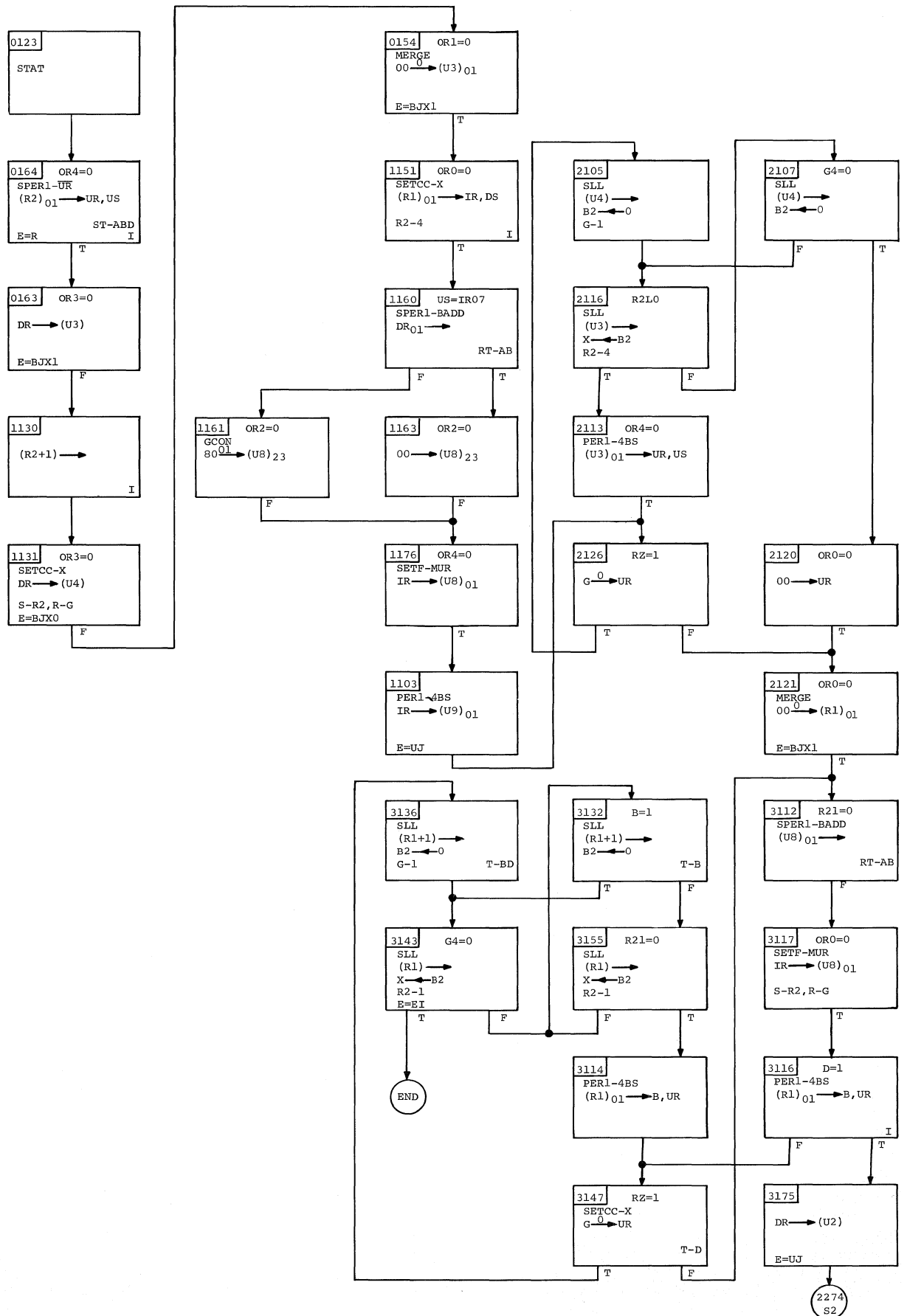


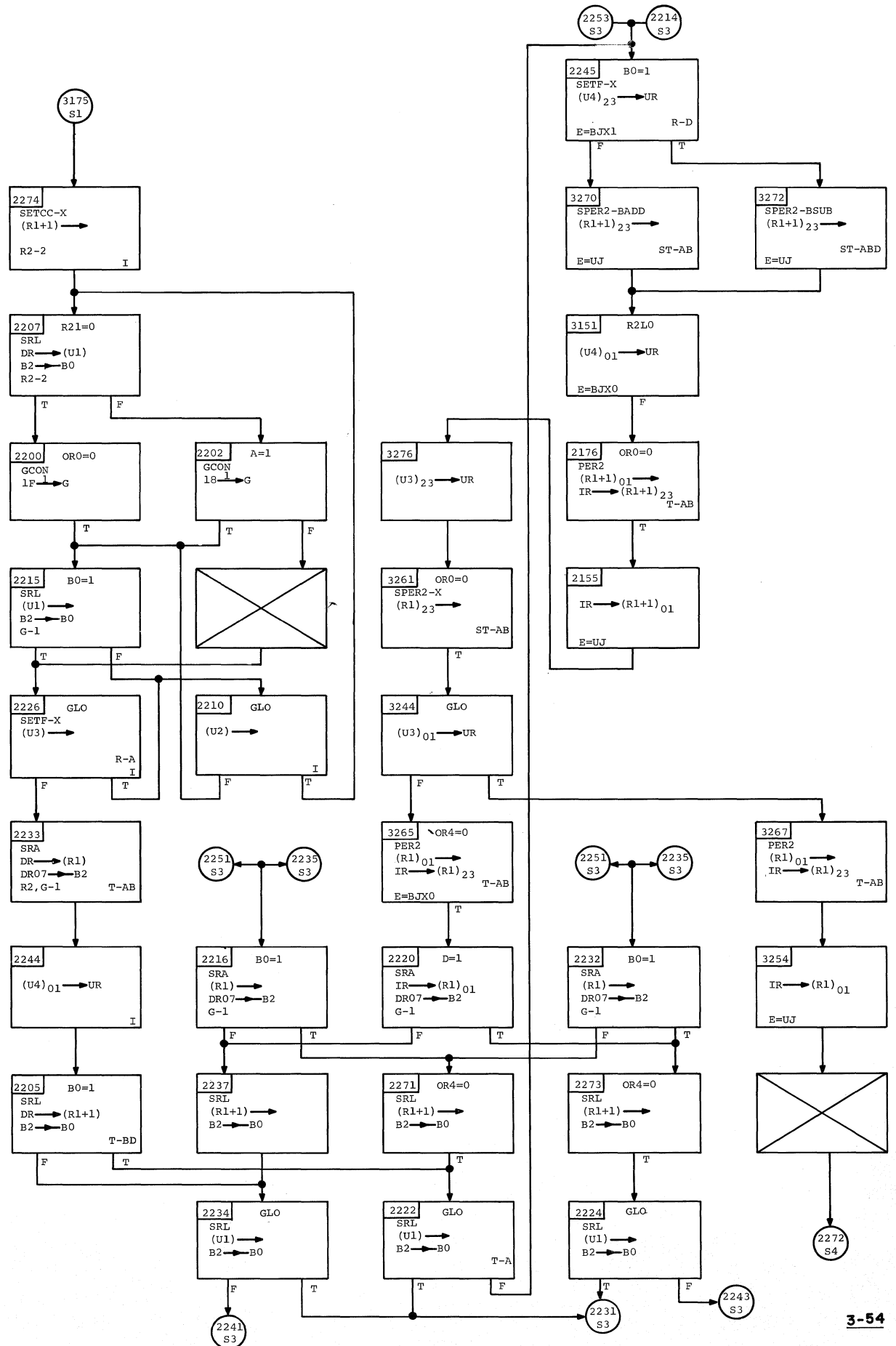
SUBTRACT NORMALIZED LONG SNDR RR 2B

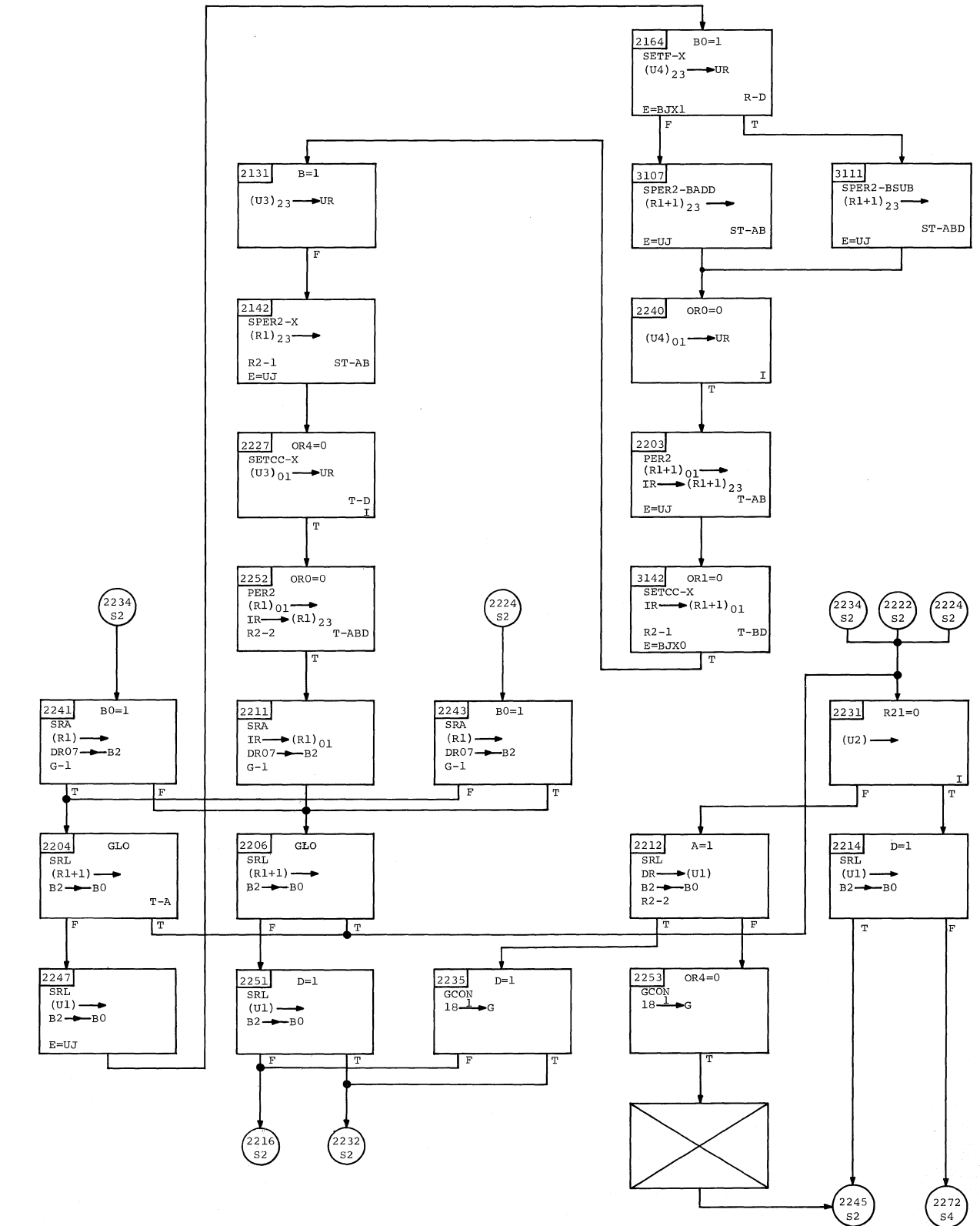


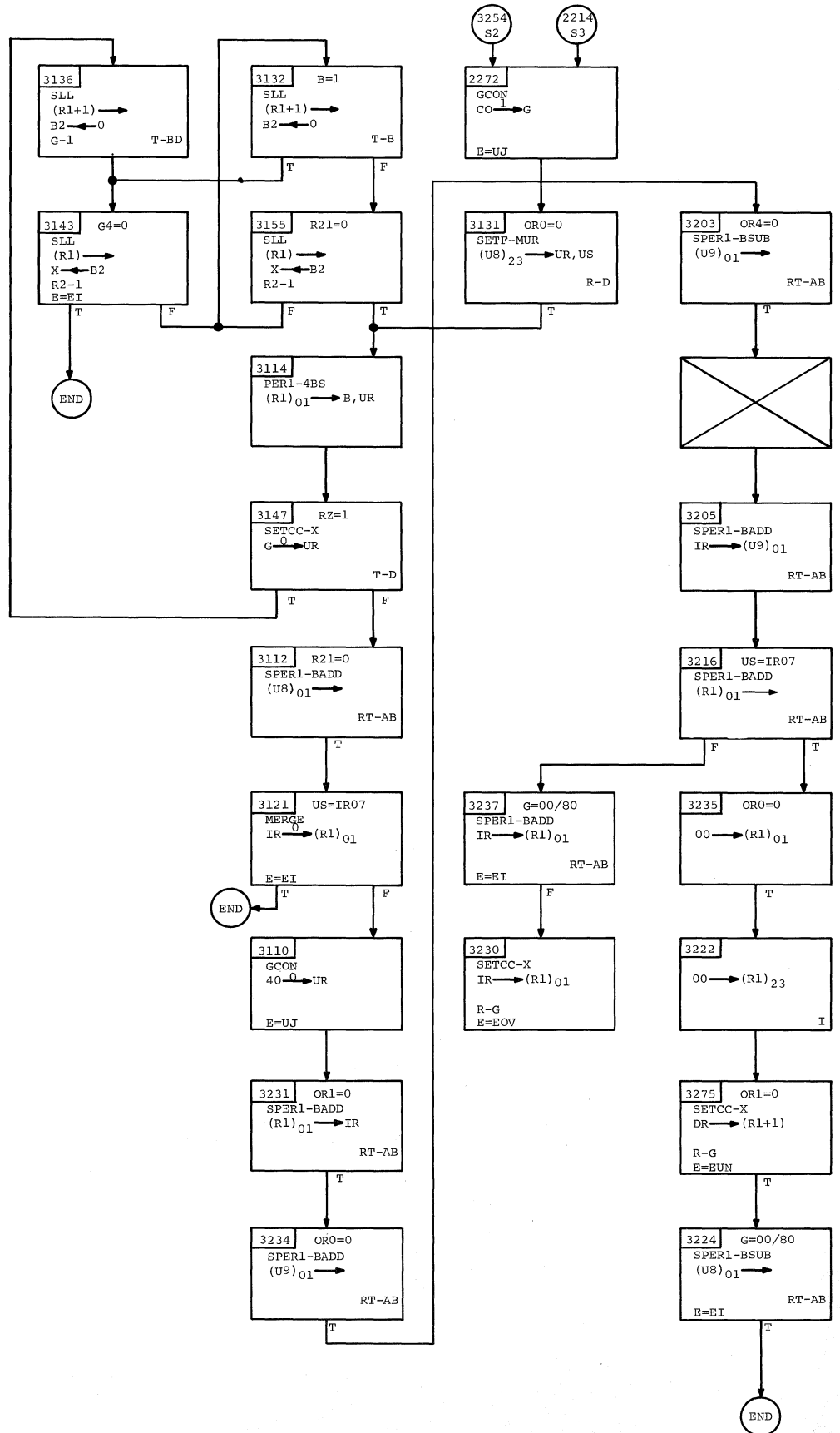


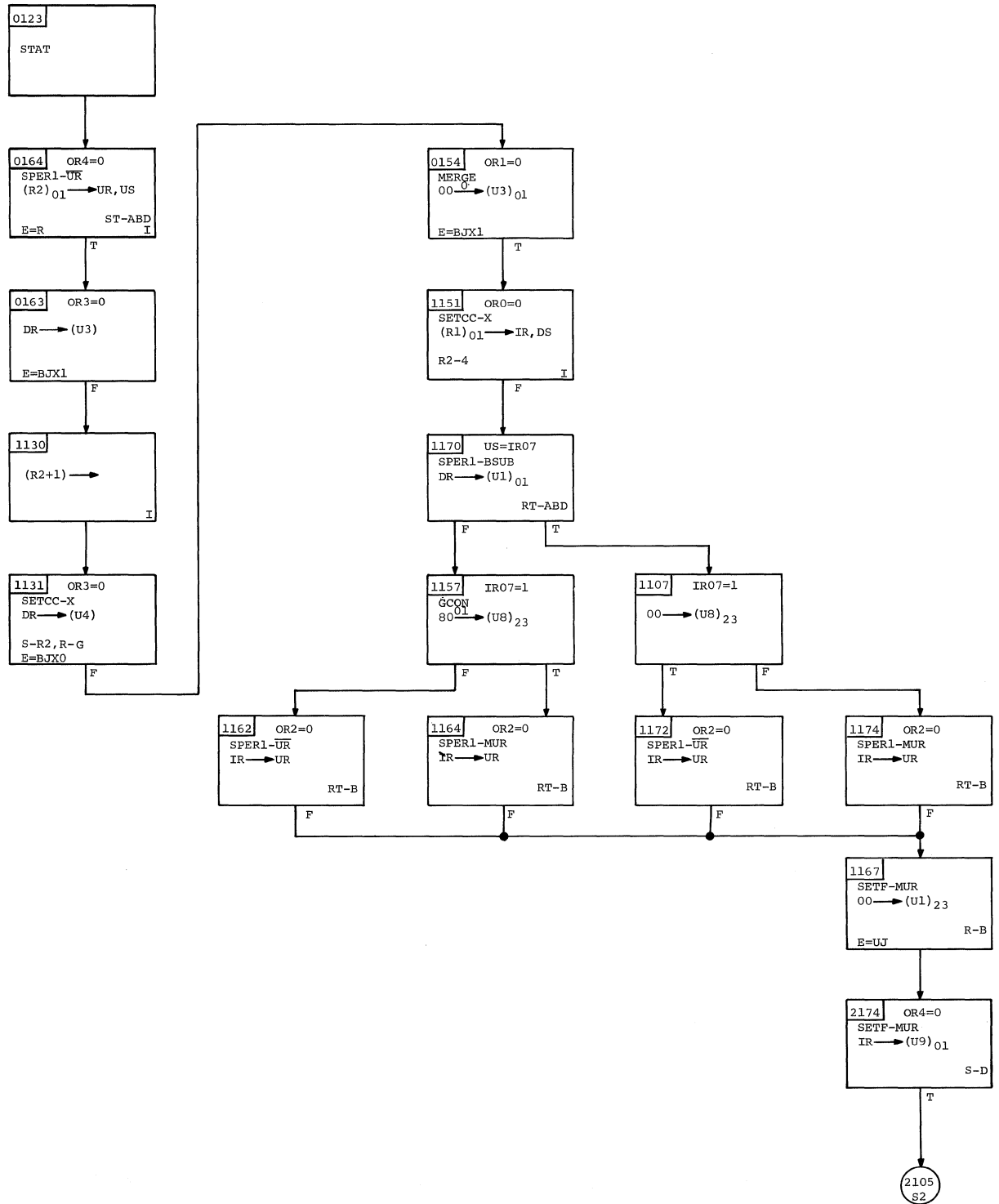


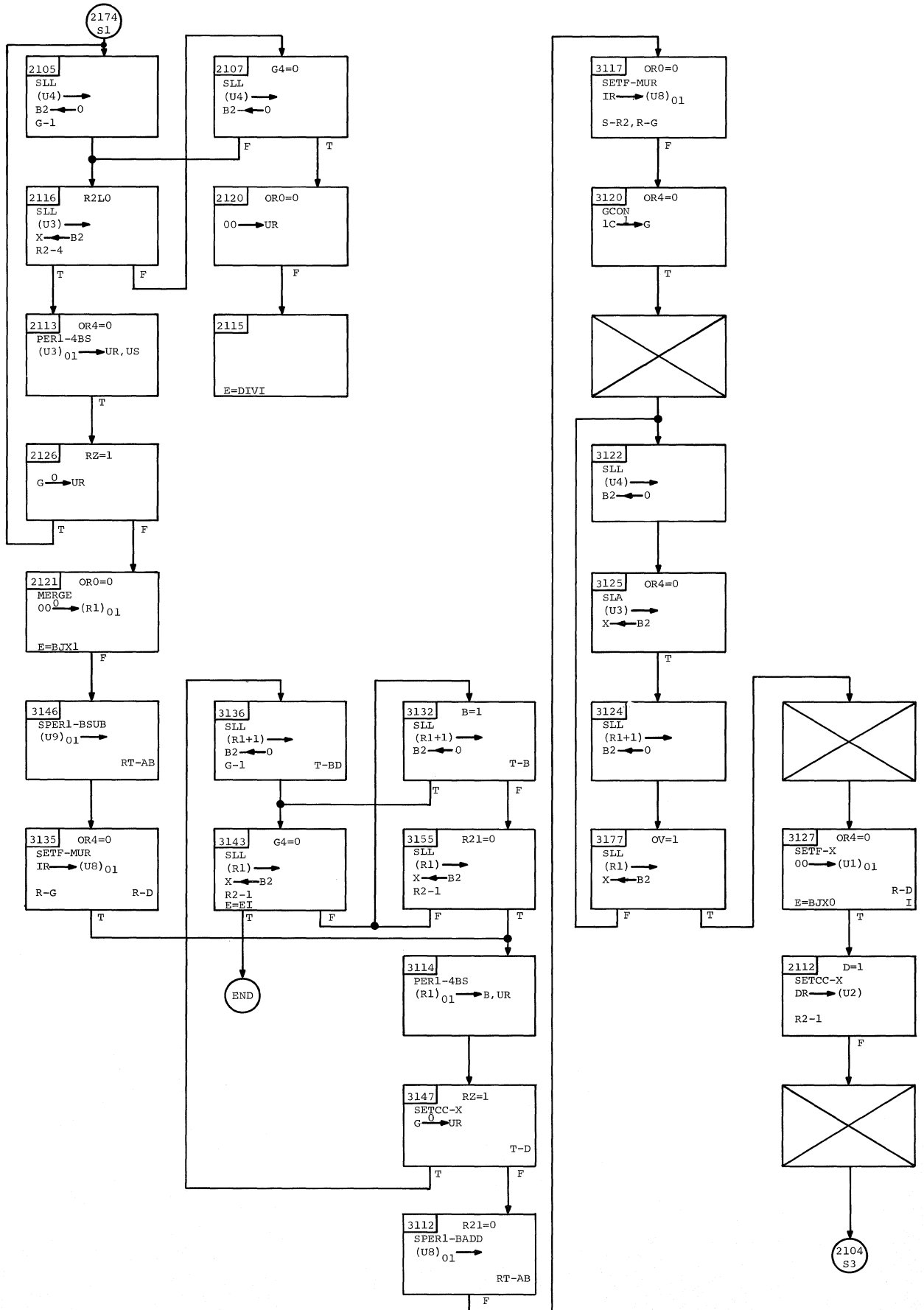


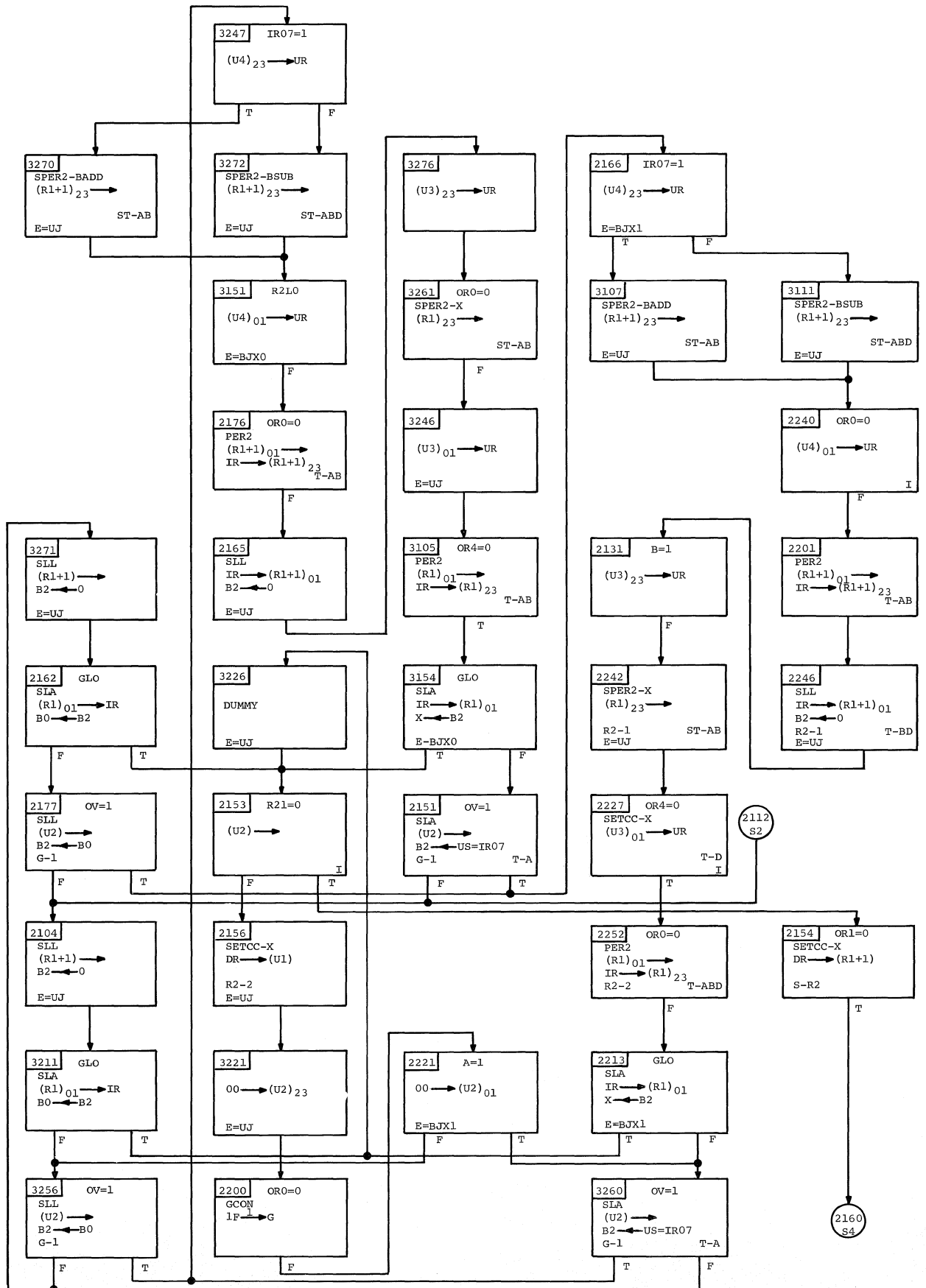




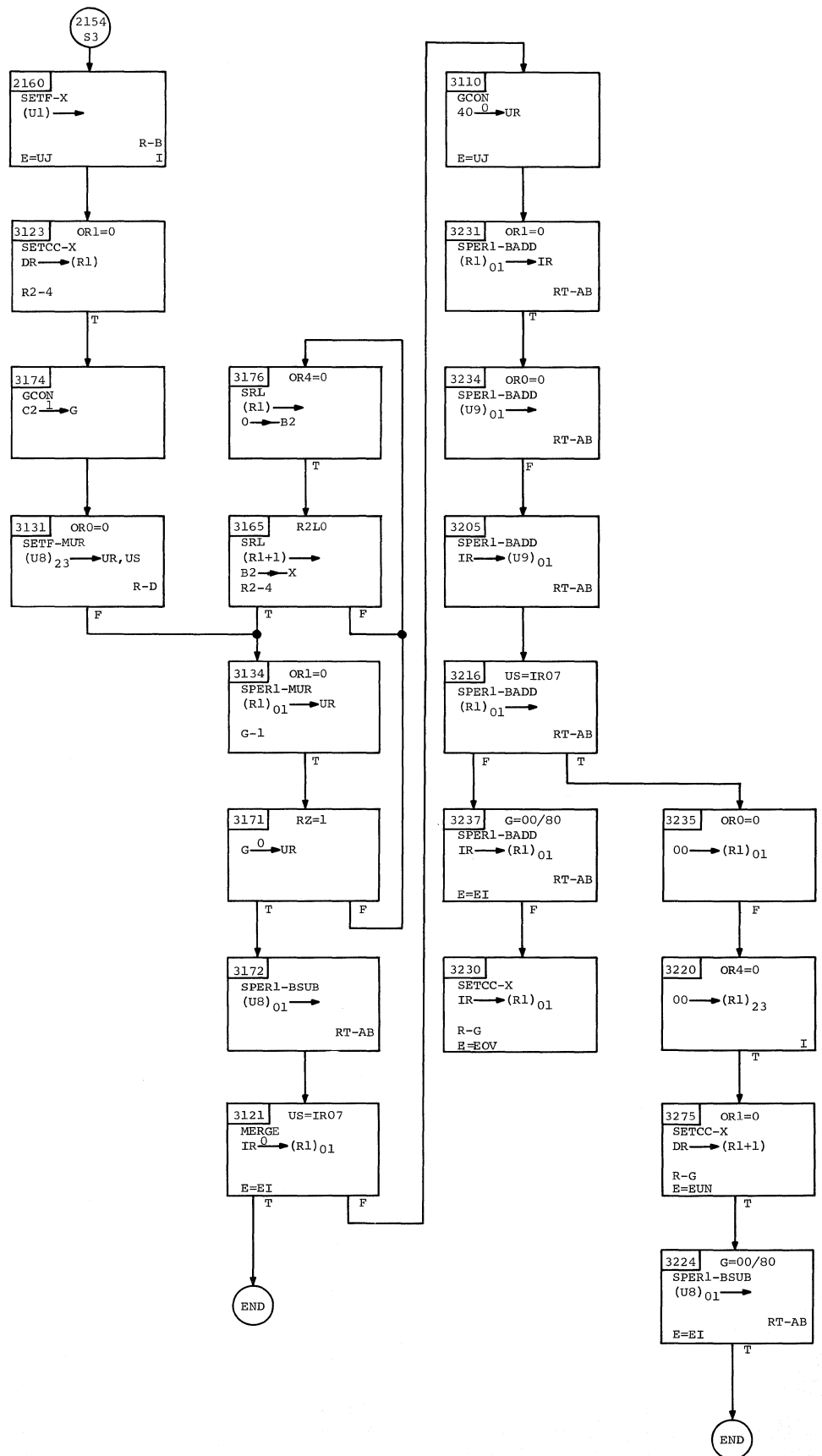


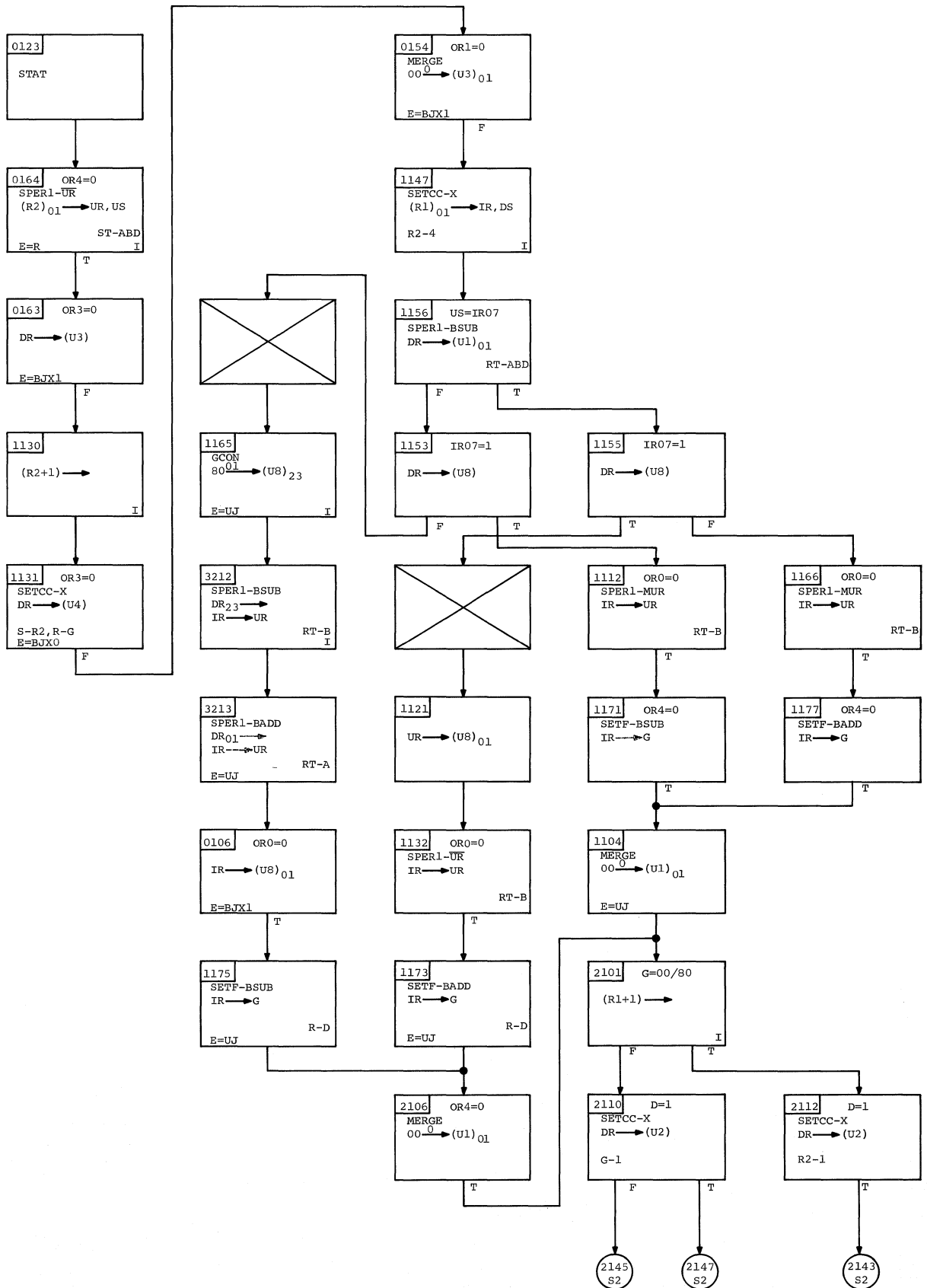


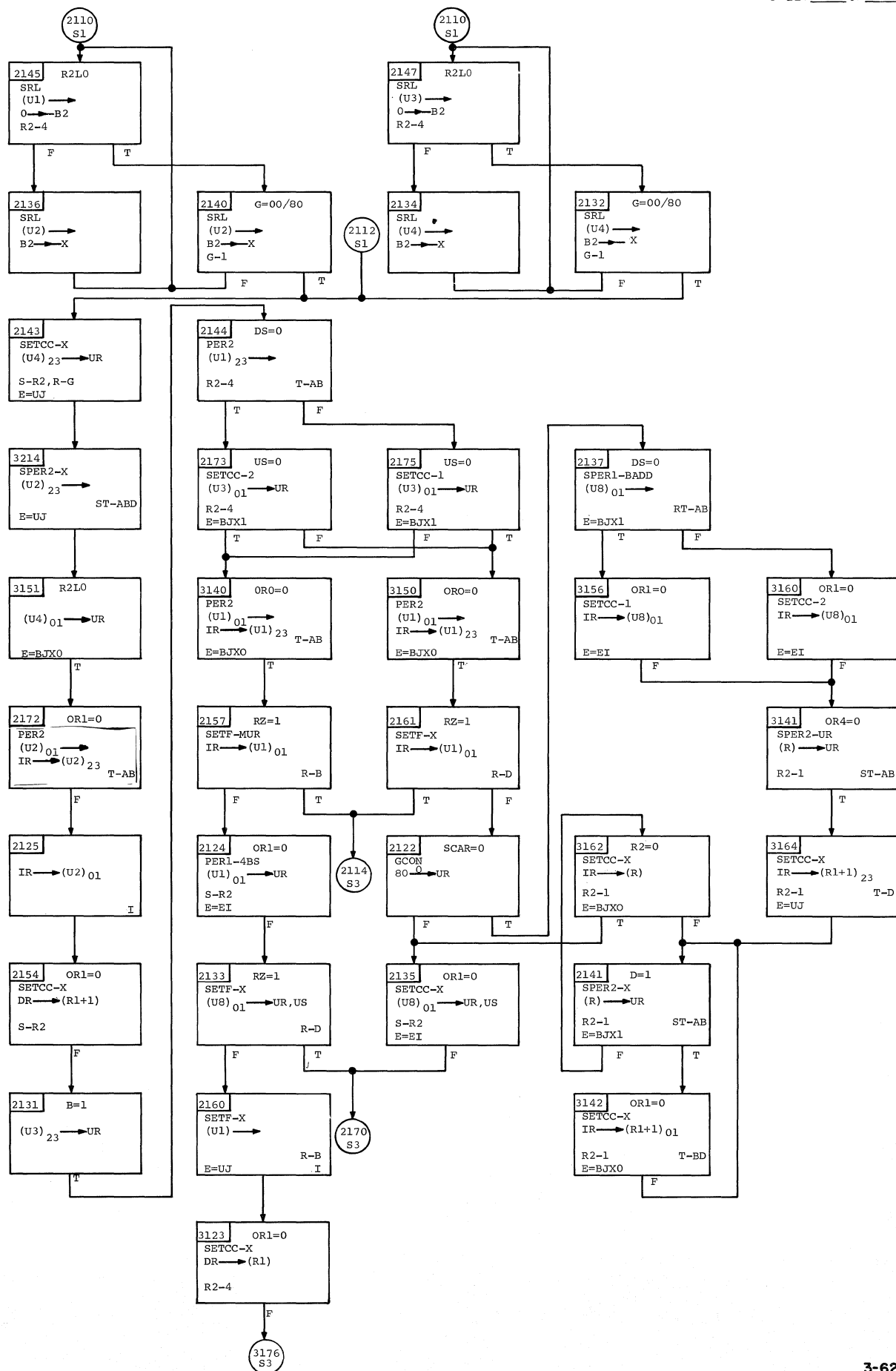


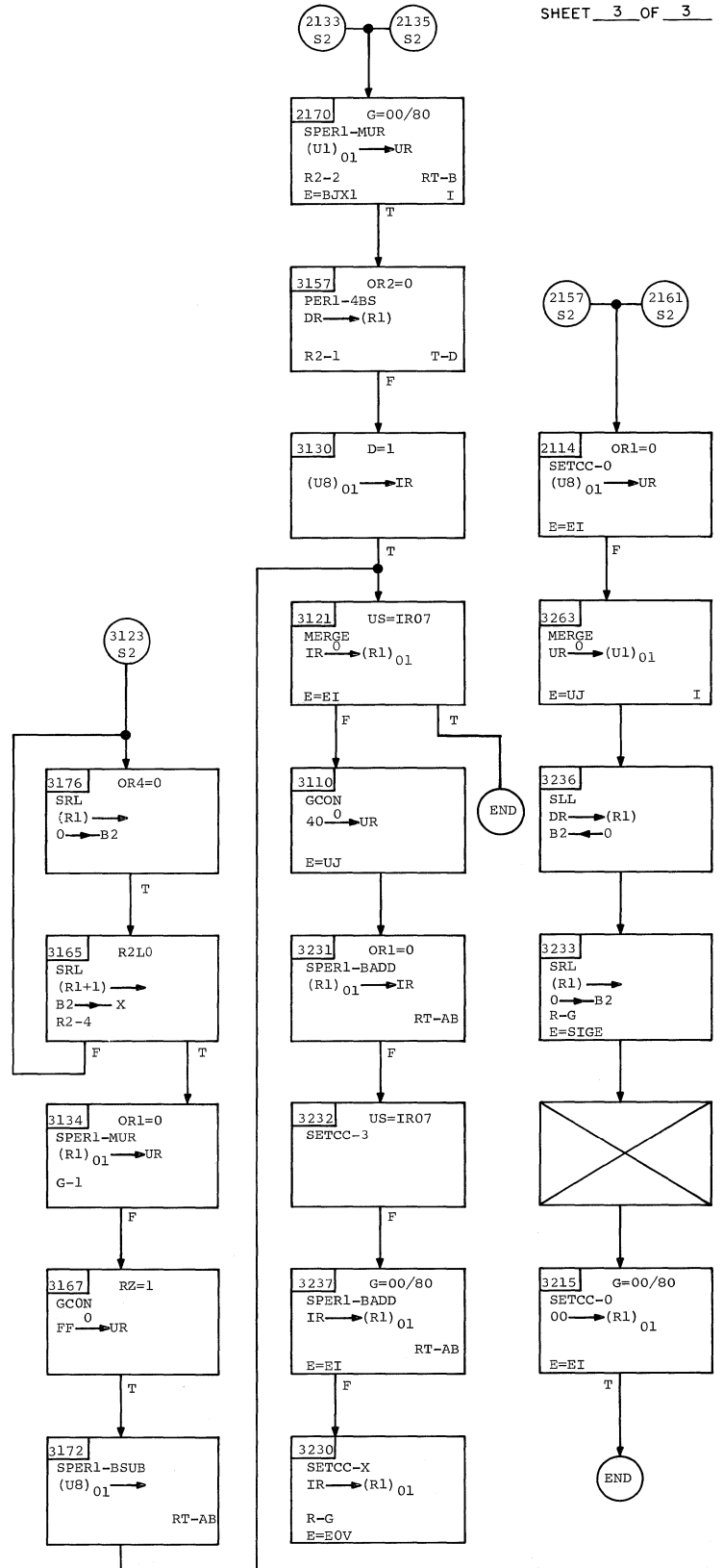


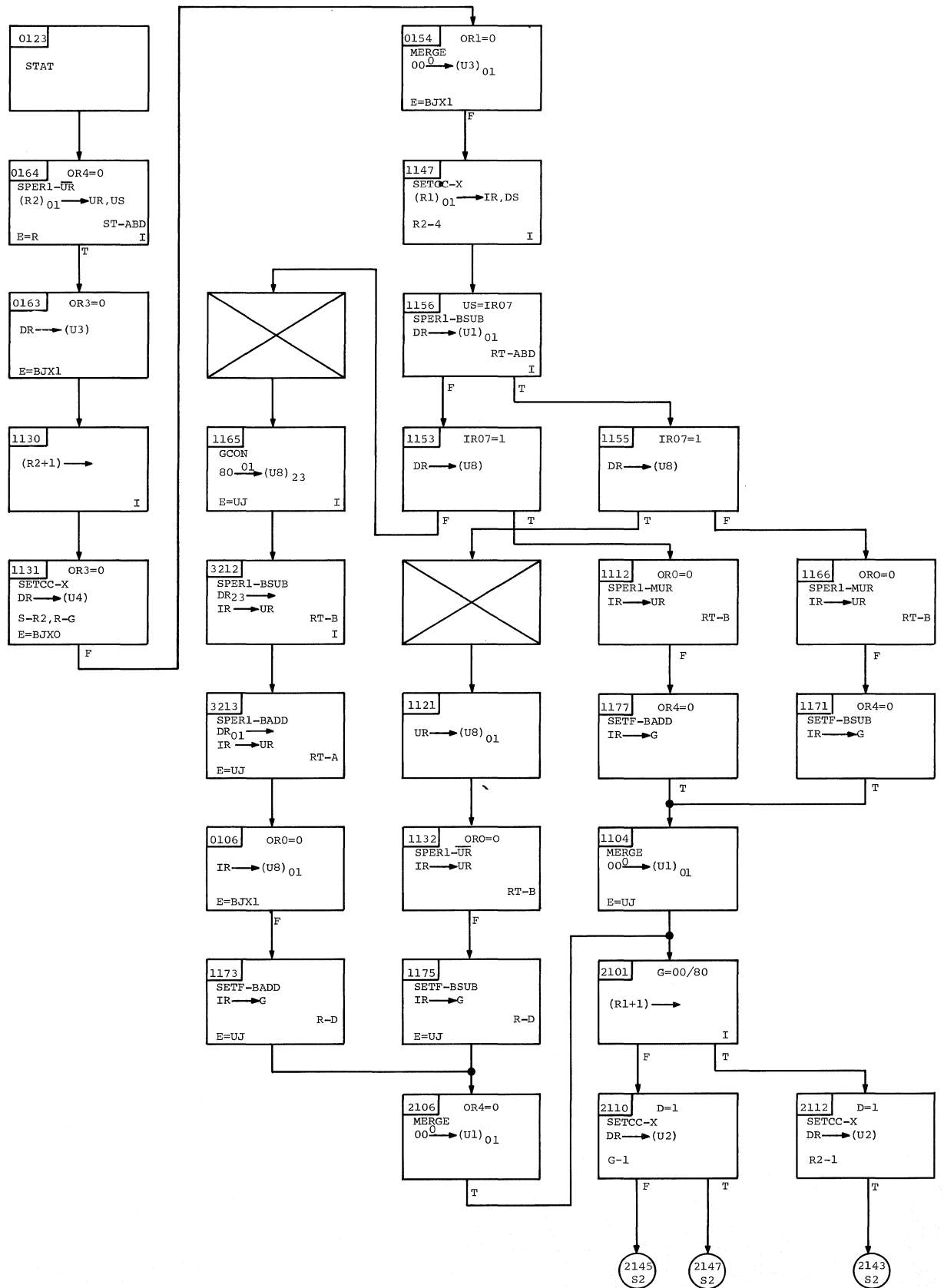




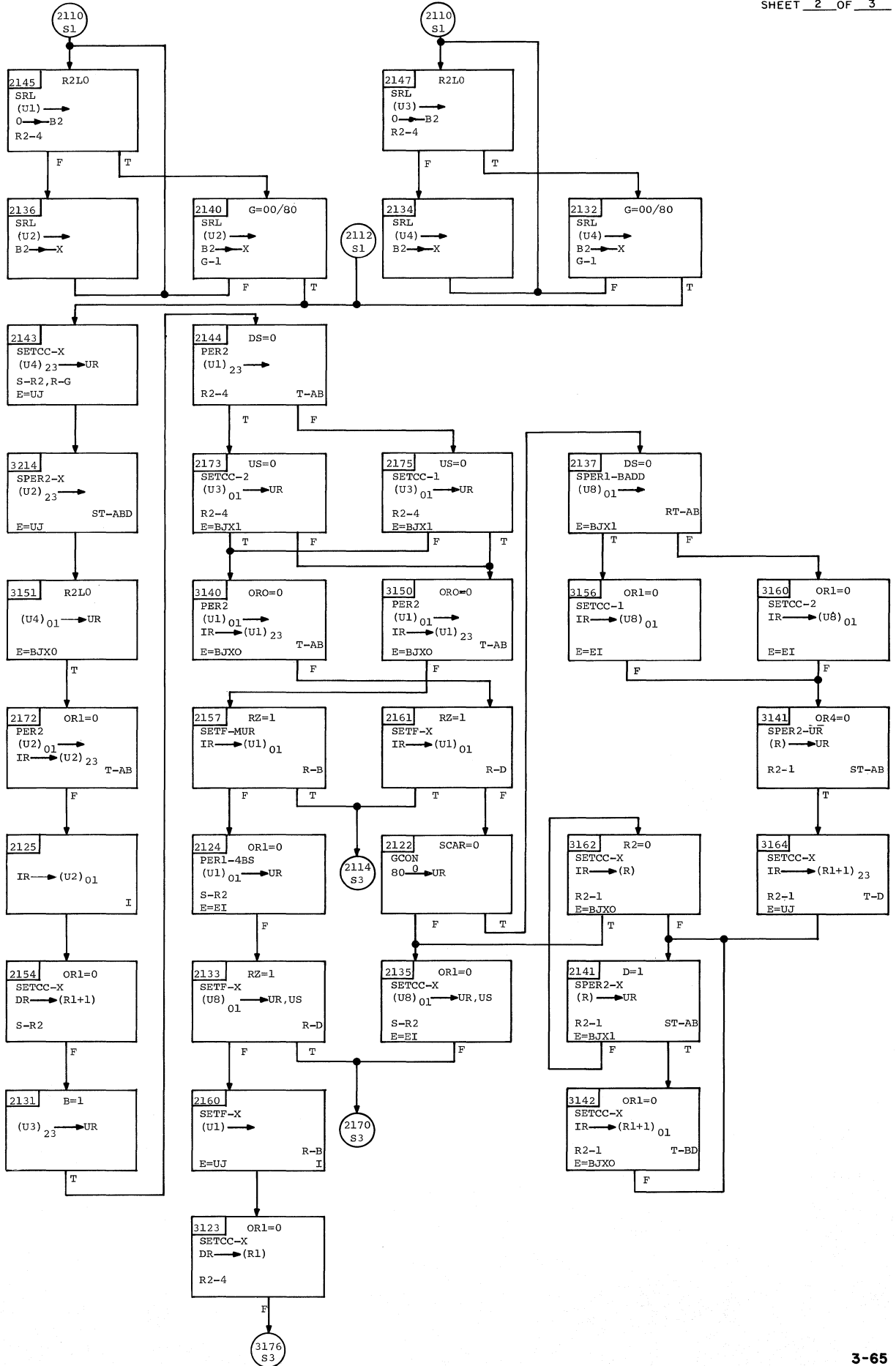




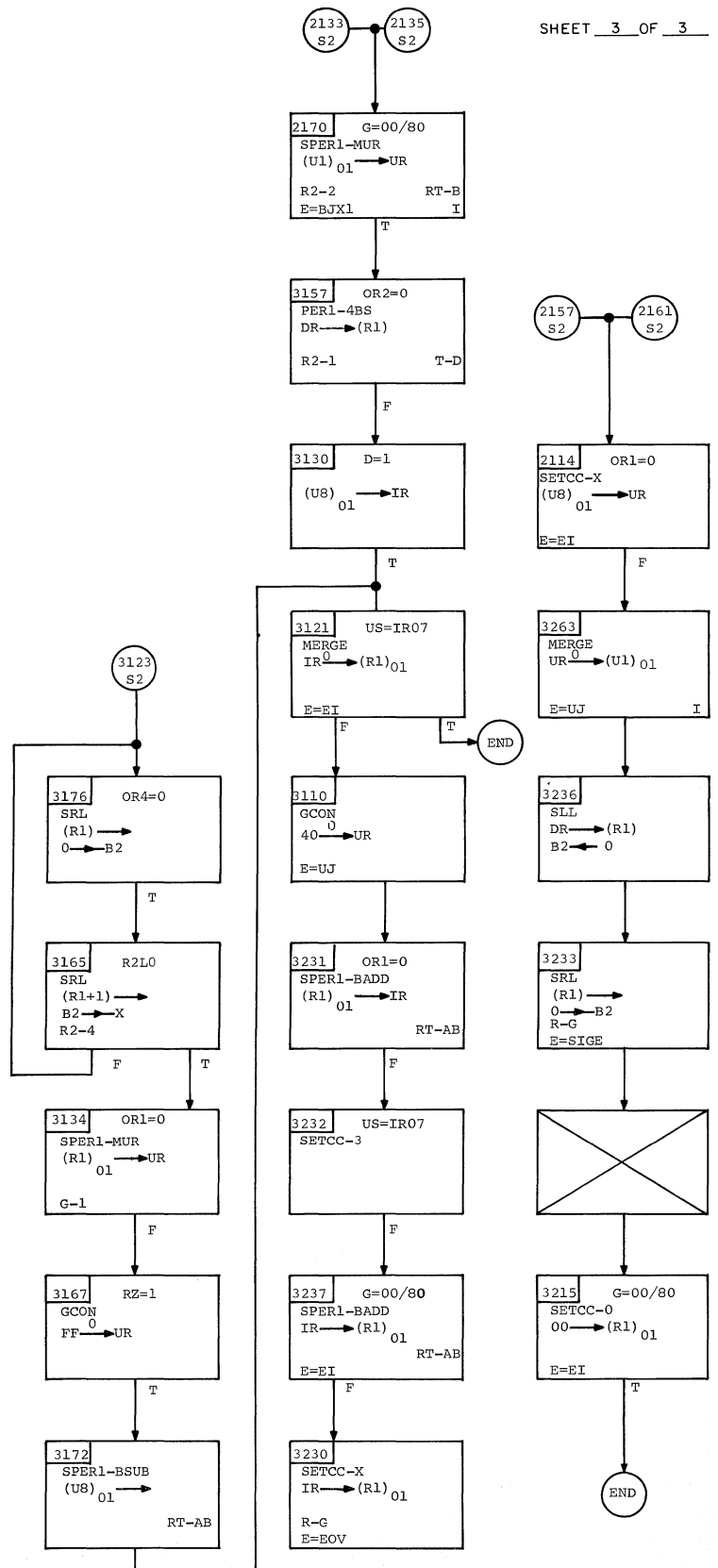


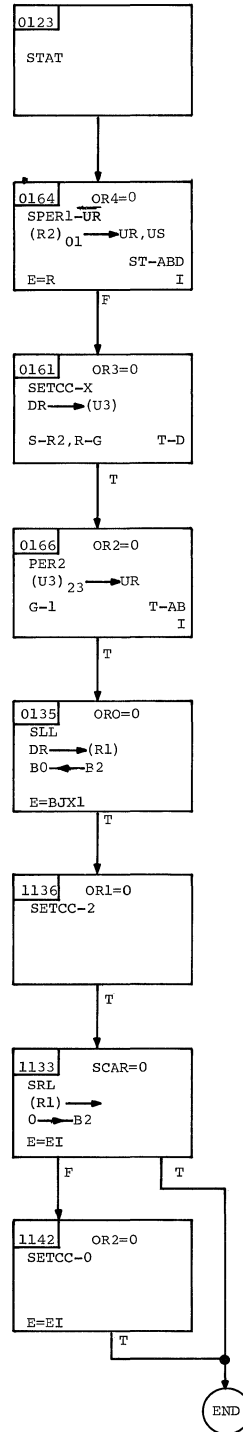


SUBTRACT UNNORMALIZED LONG SWR RR 2F

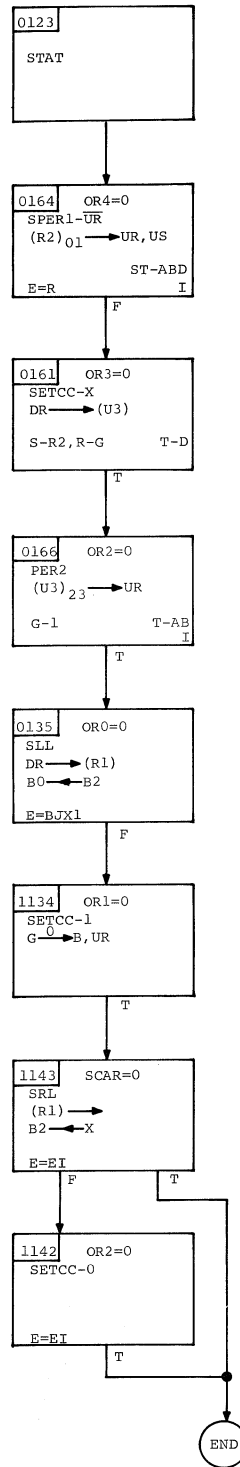


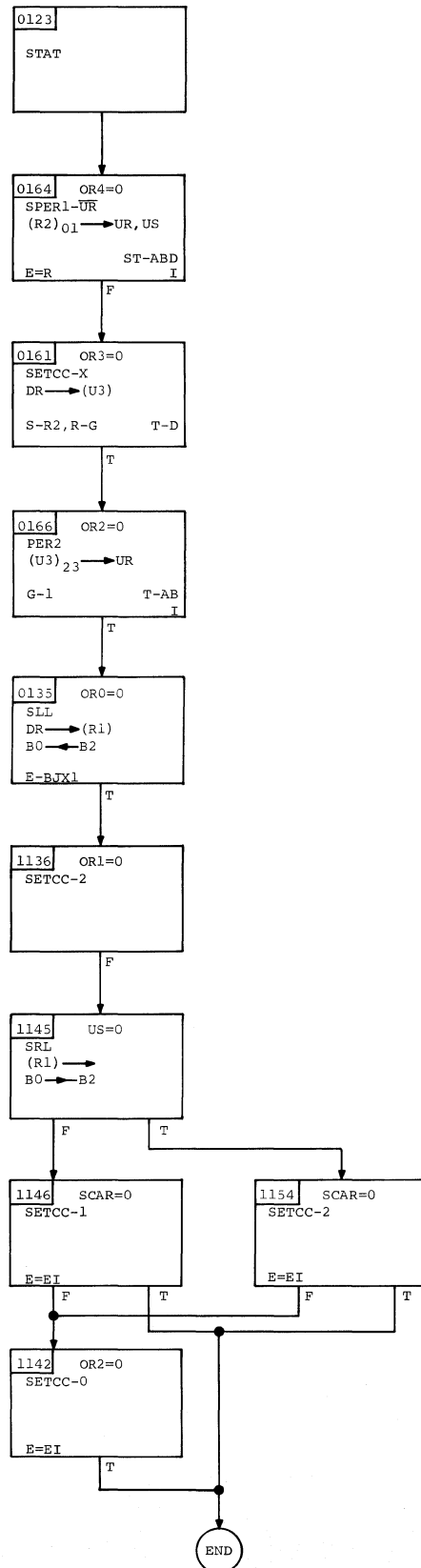
SUBTRACT UNNORMALIZED LONG SWR RR 2F

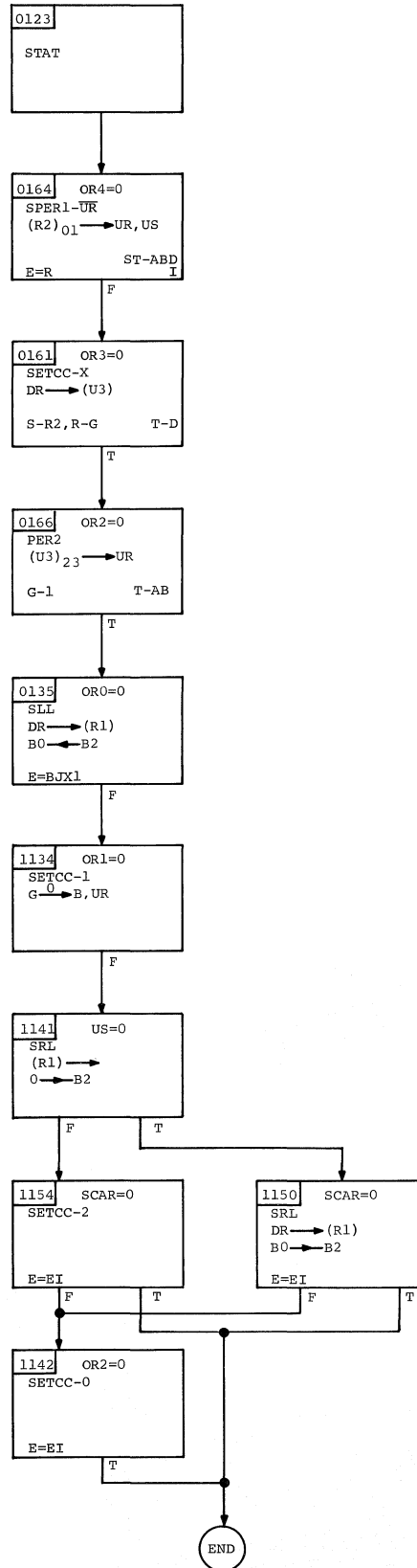


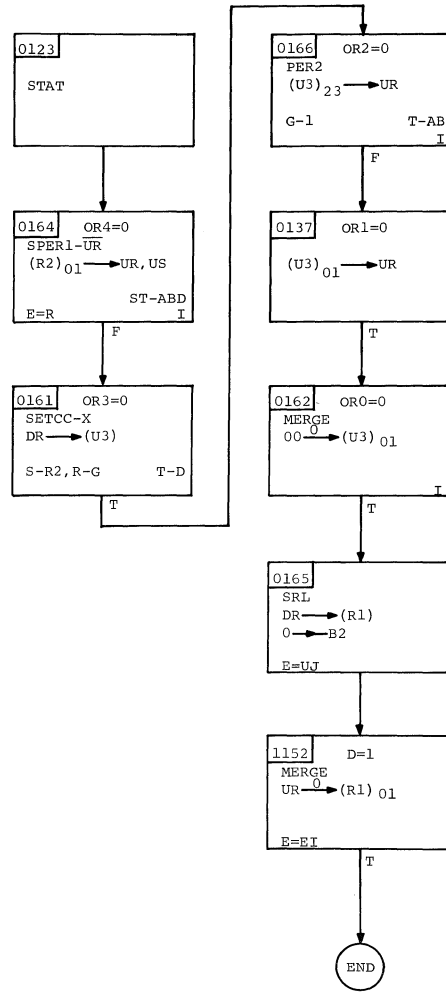


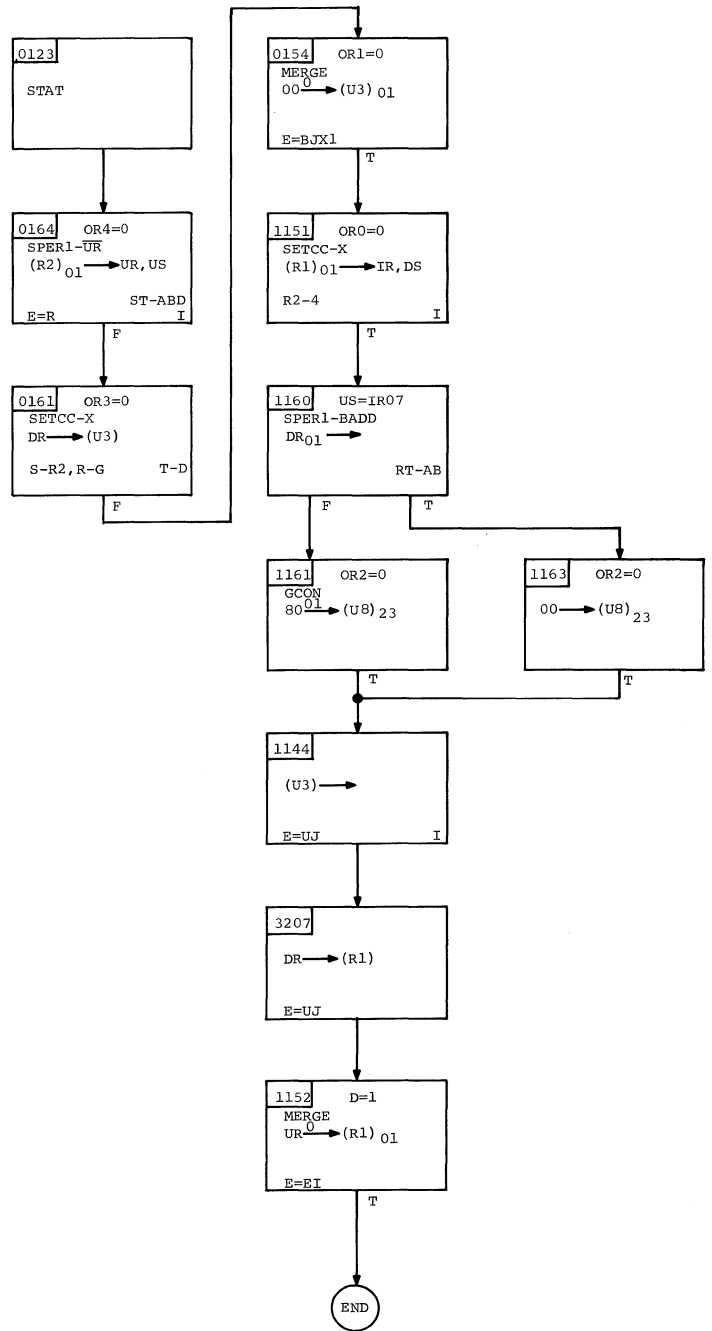


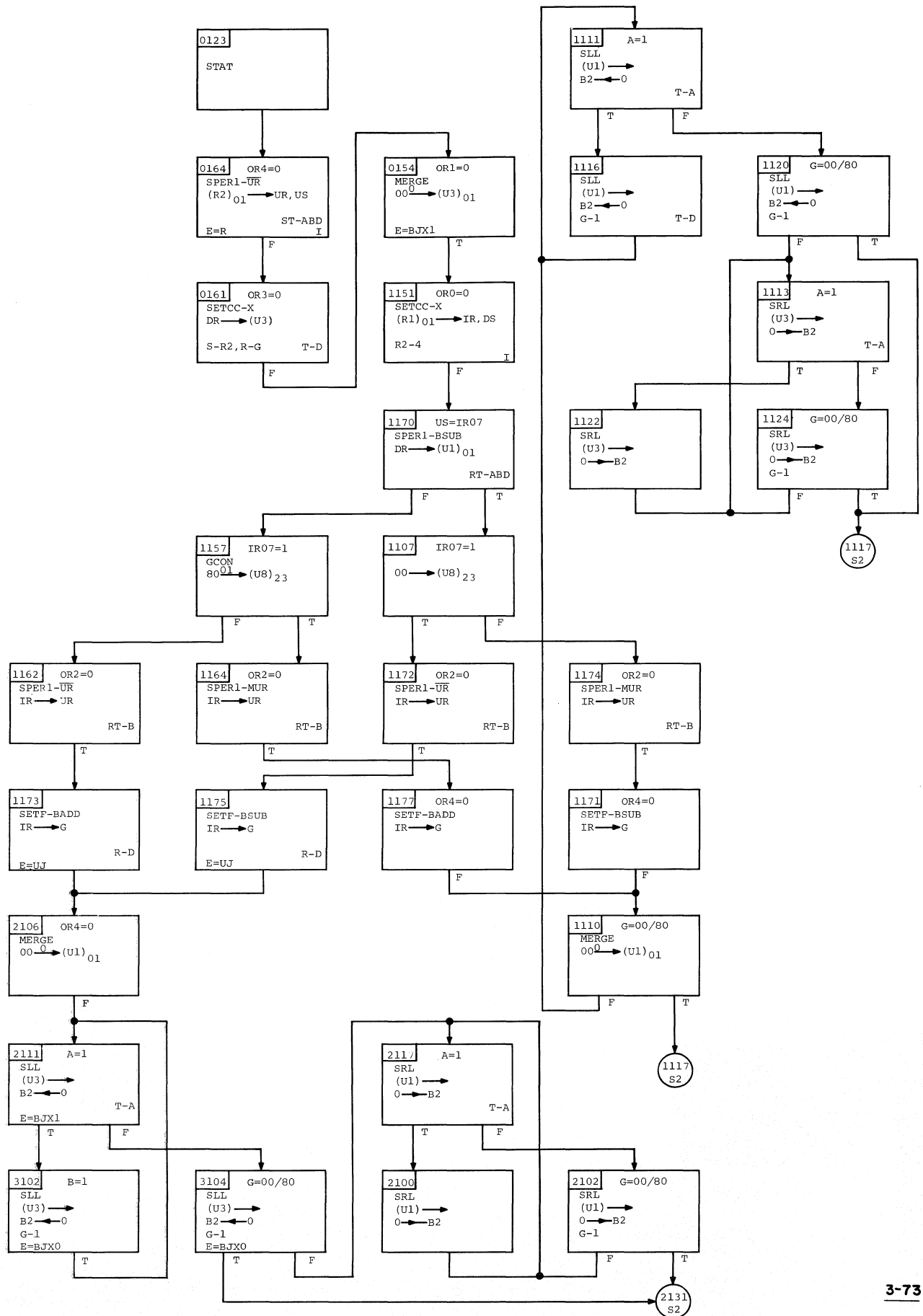


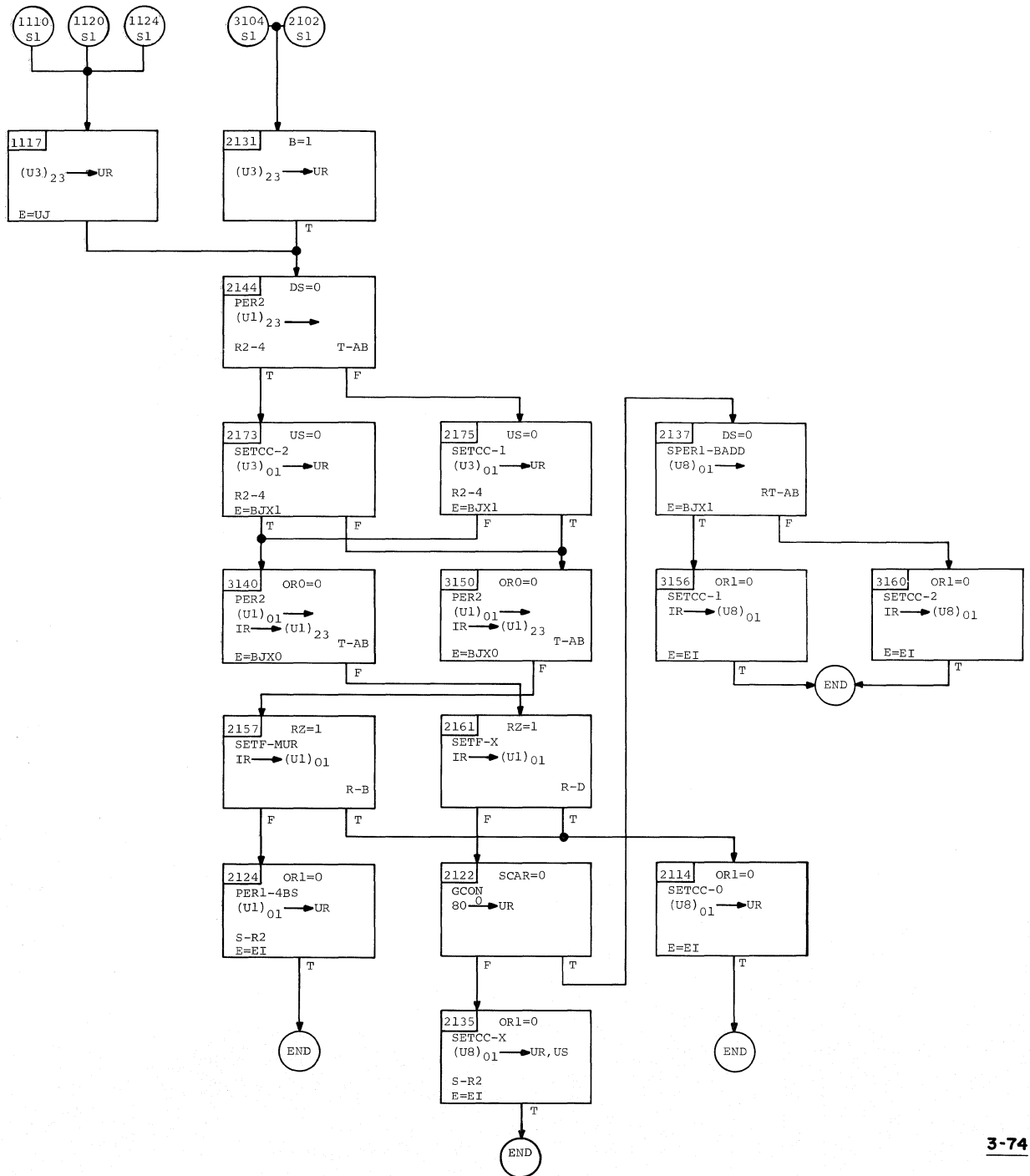


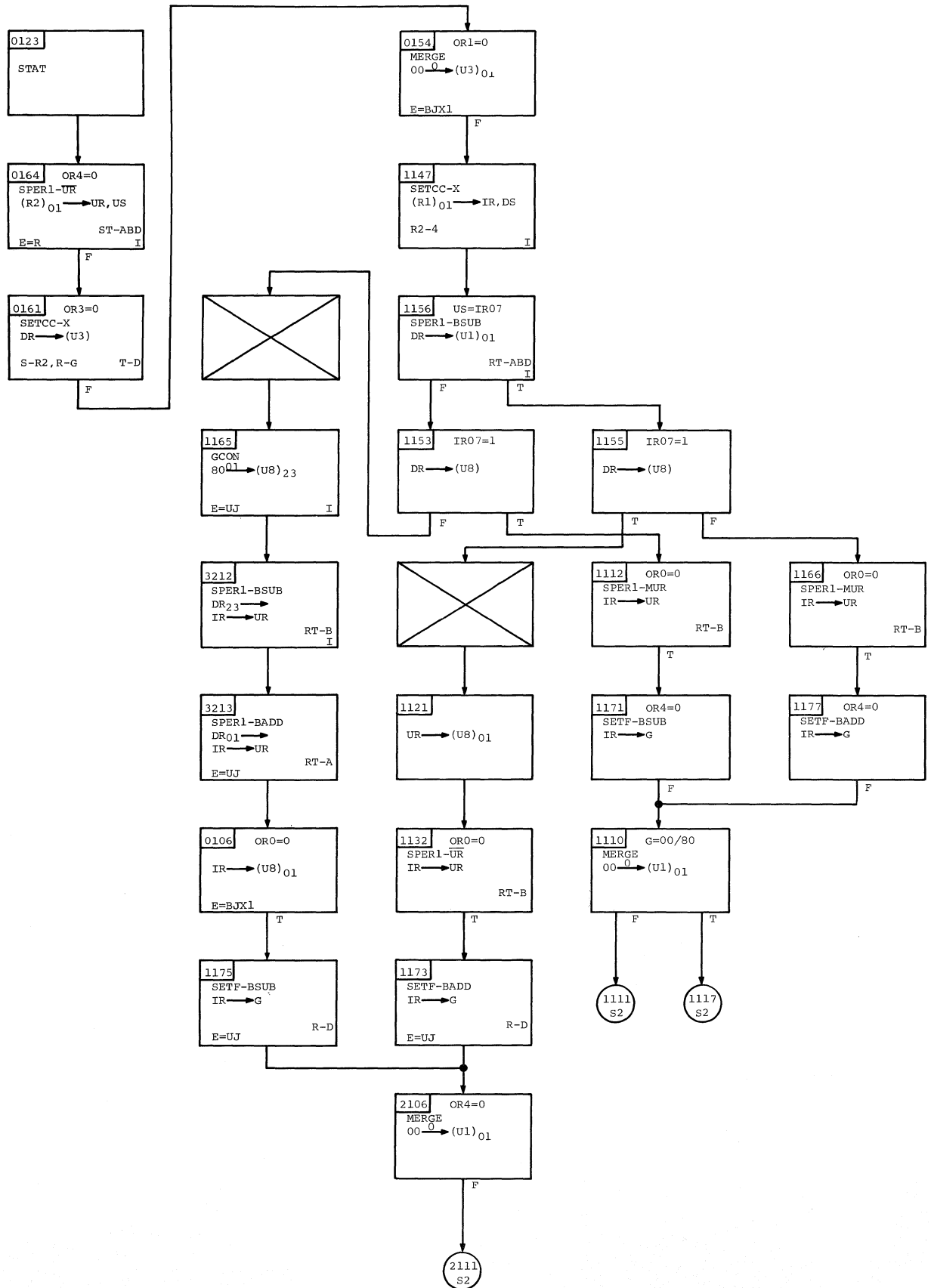




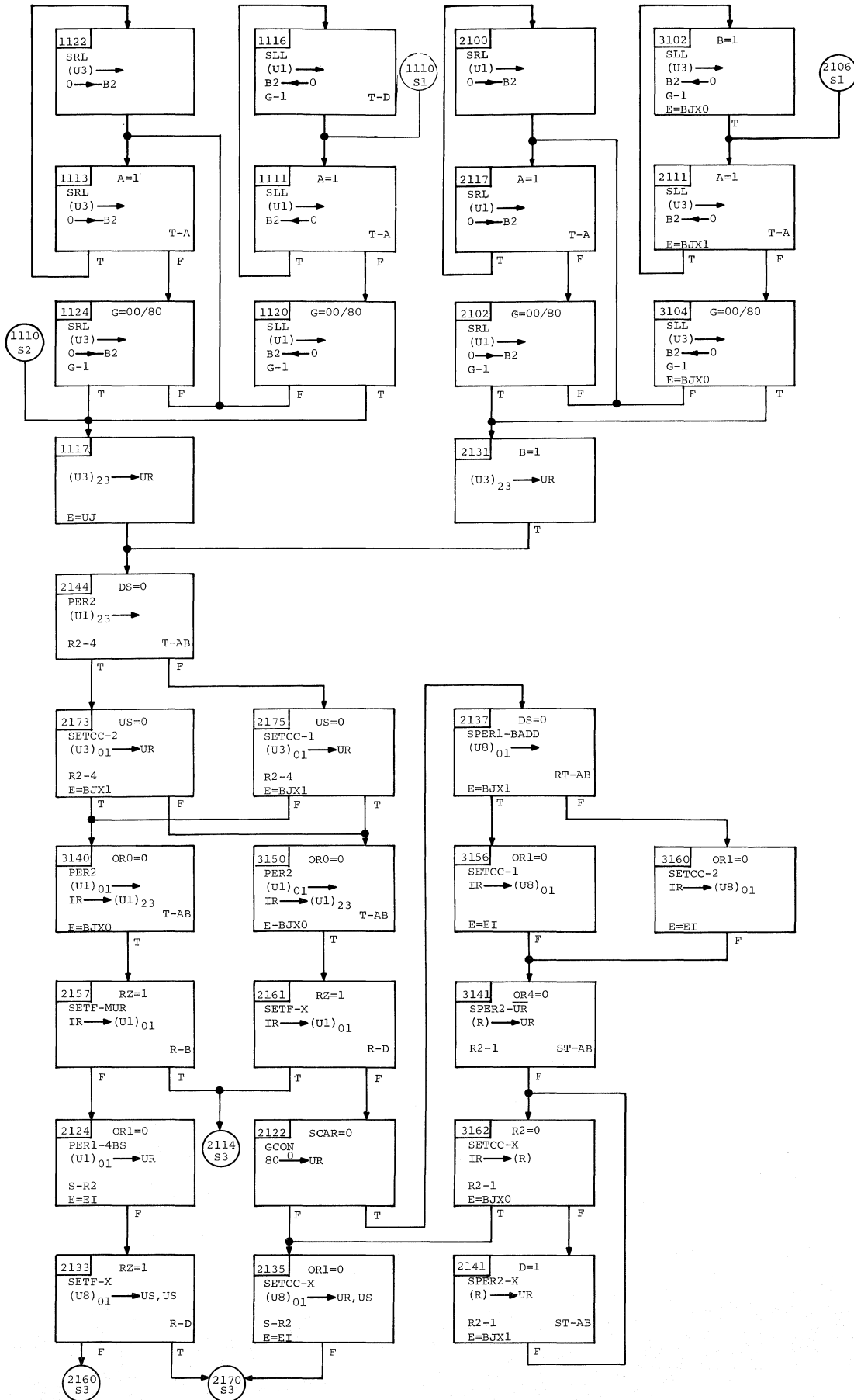


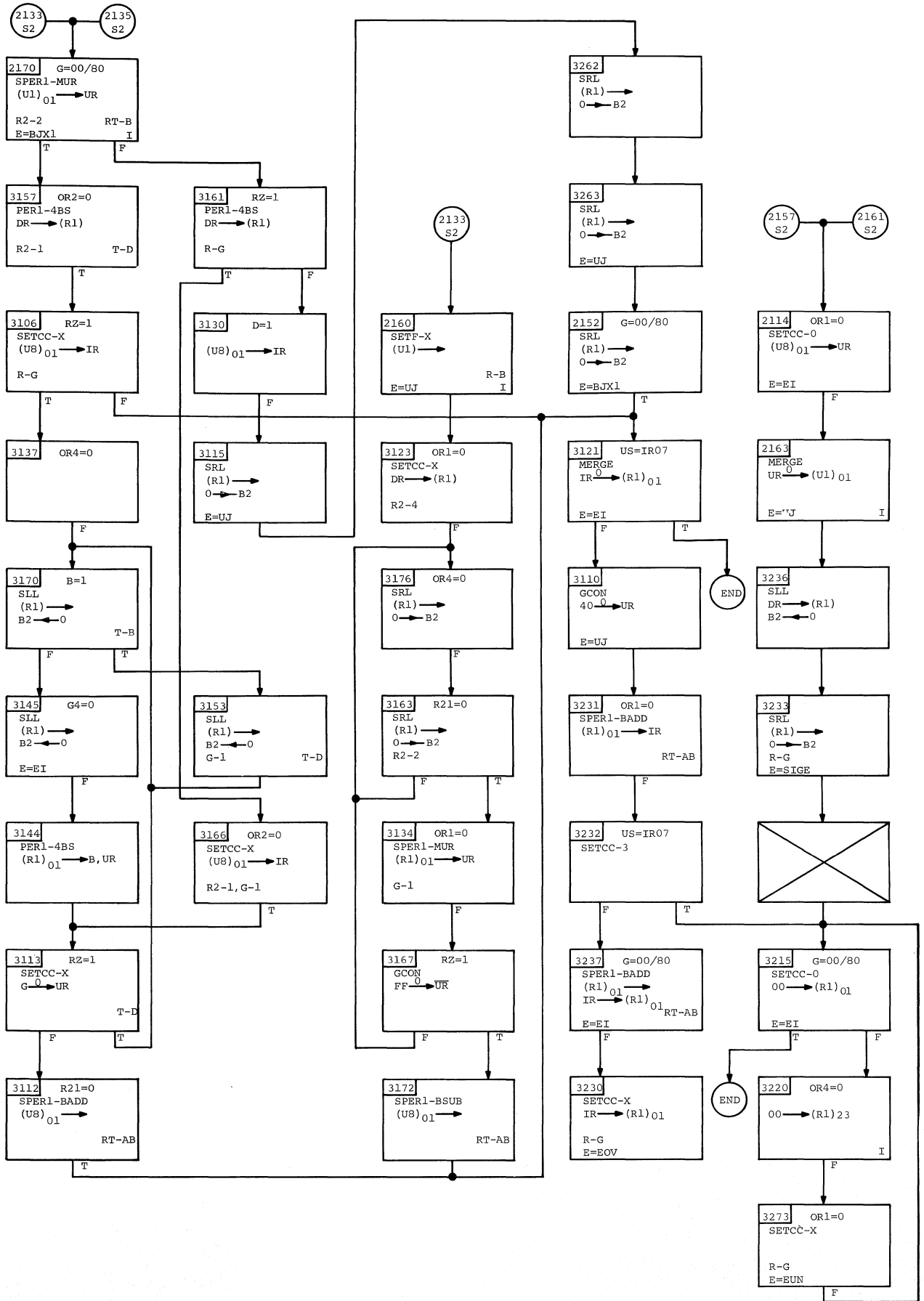


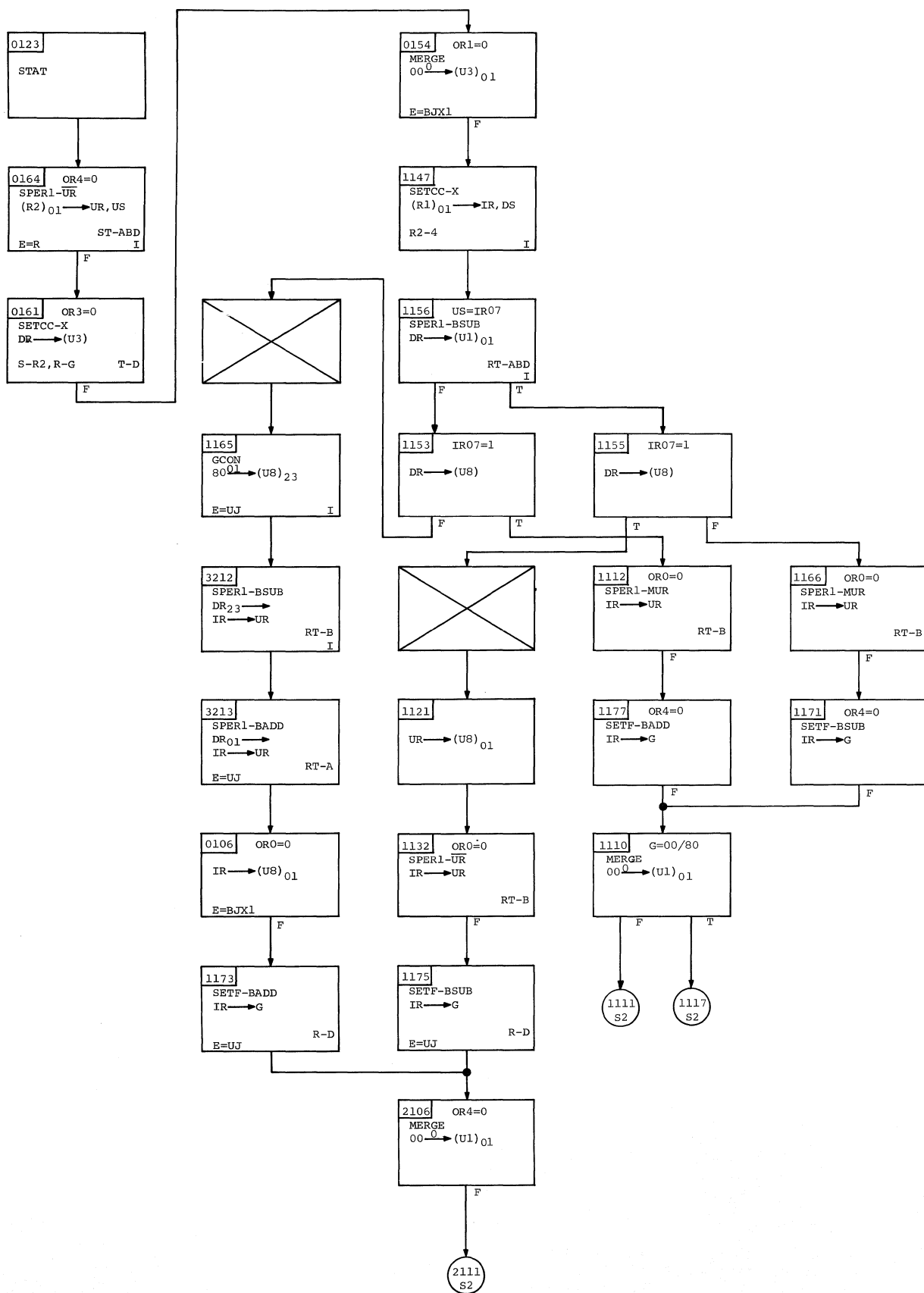




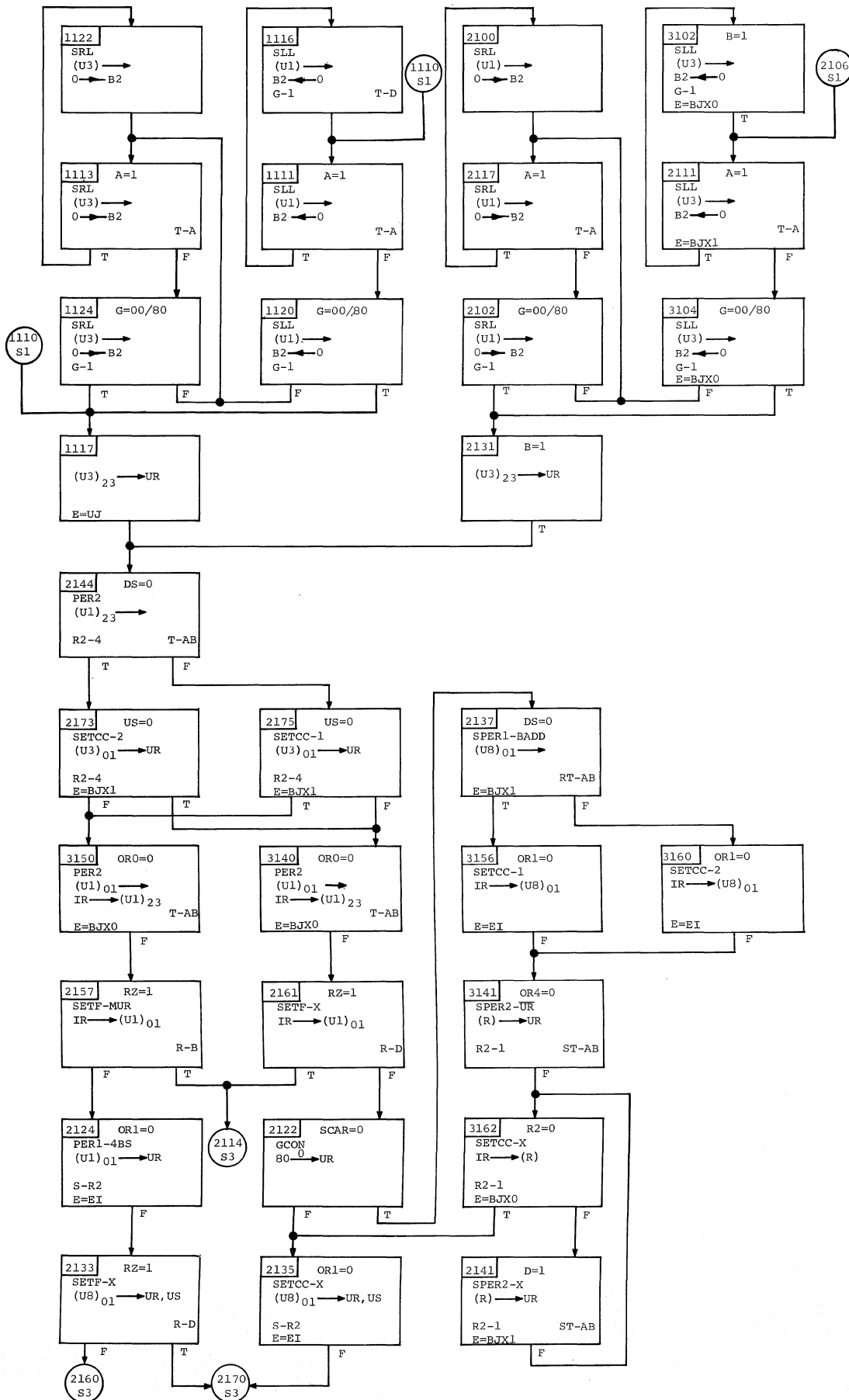




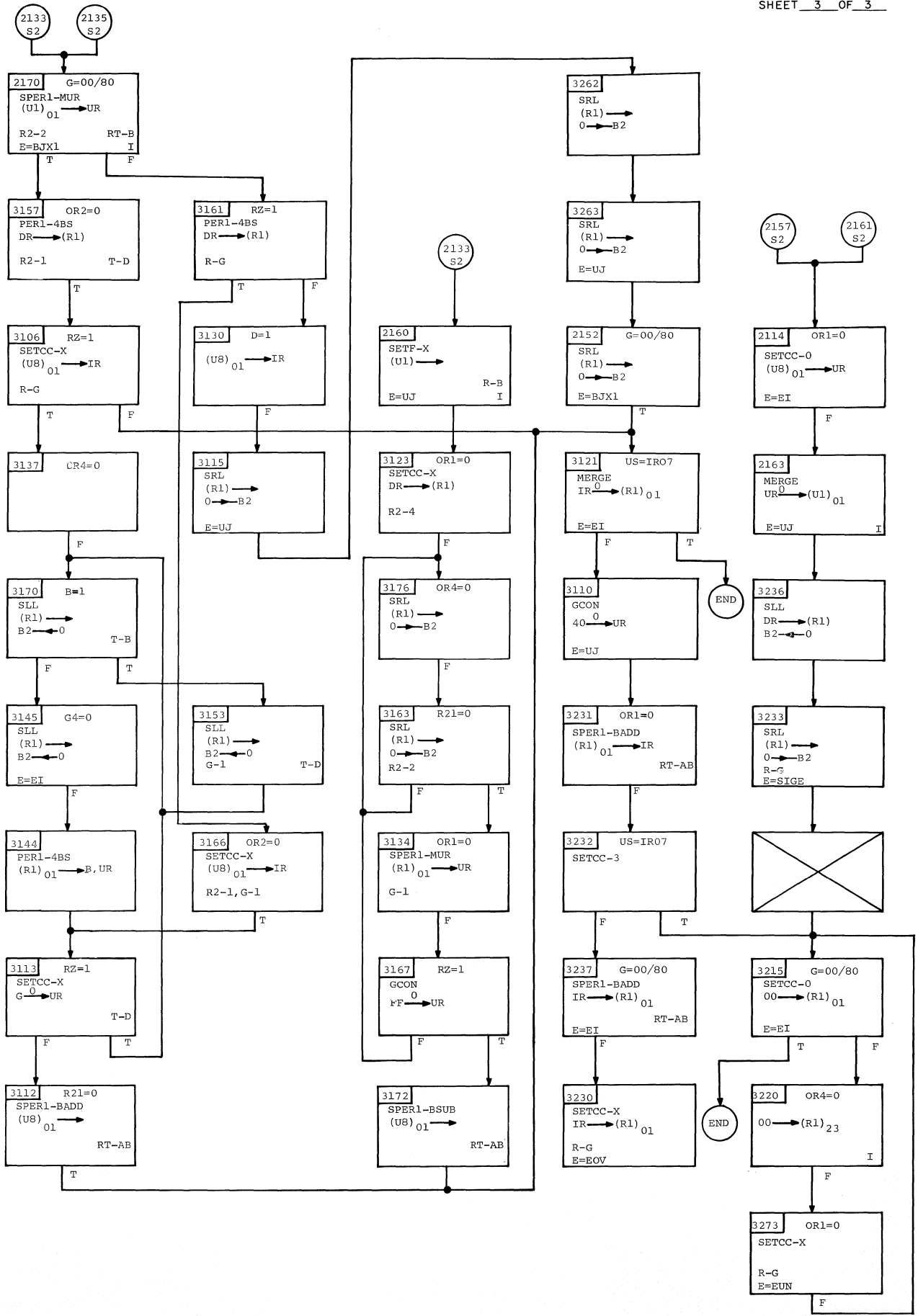


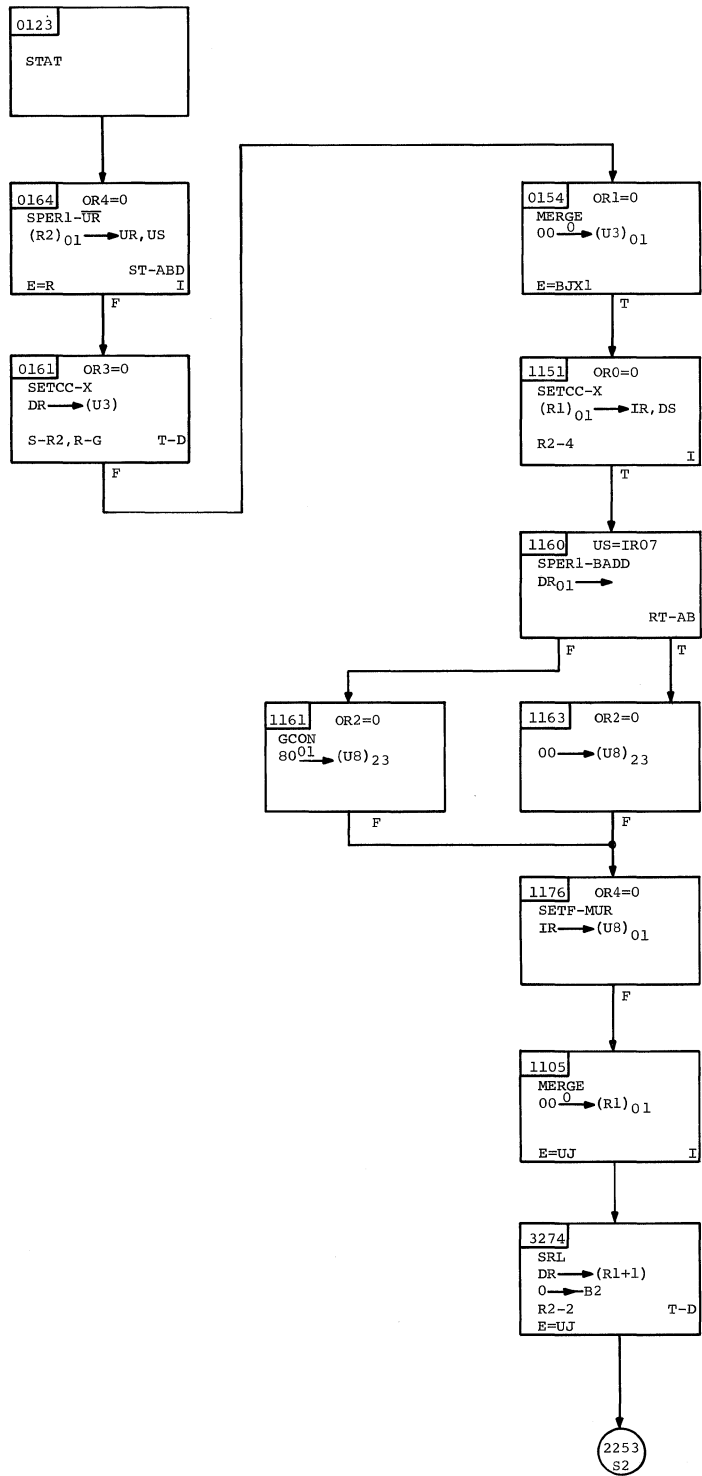


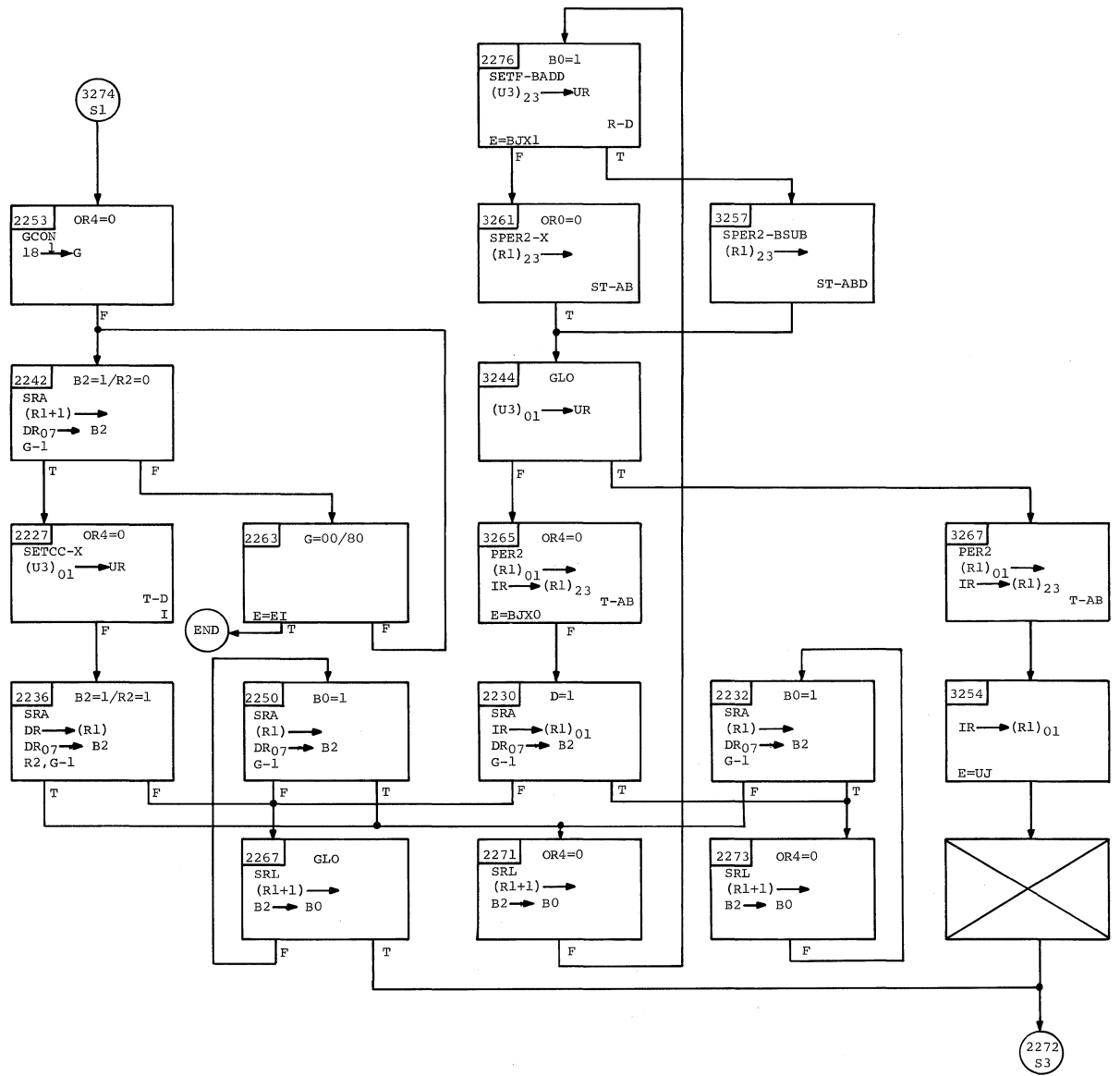
SUBTRACT NORMALIZED SHORT SNER RR 3B

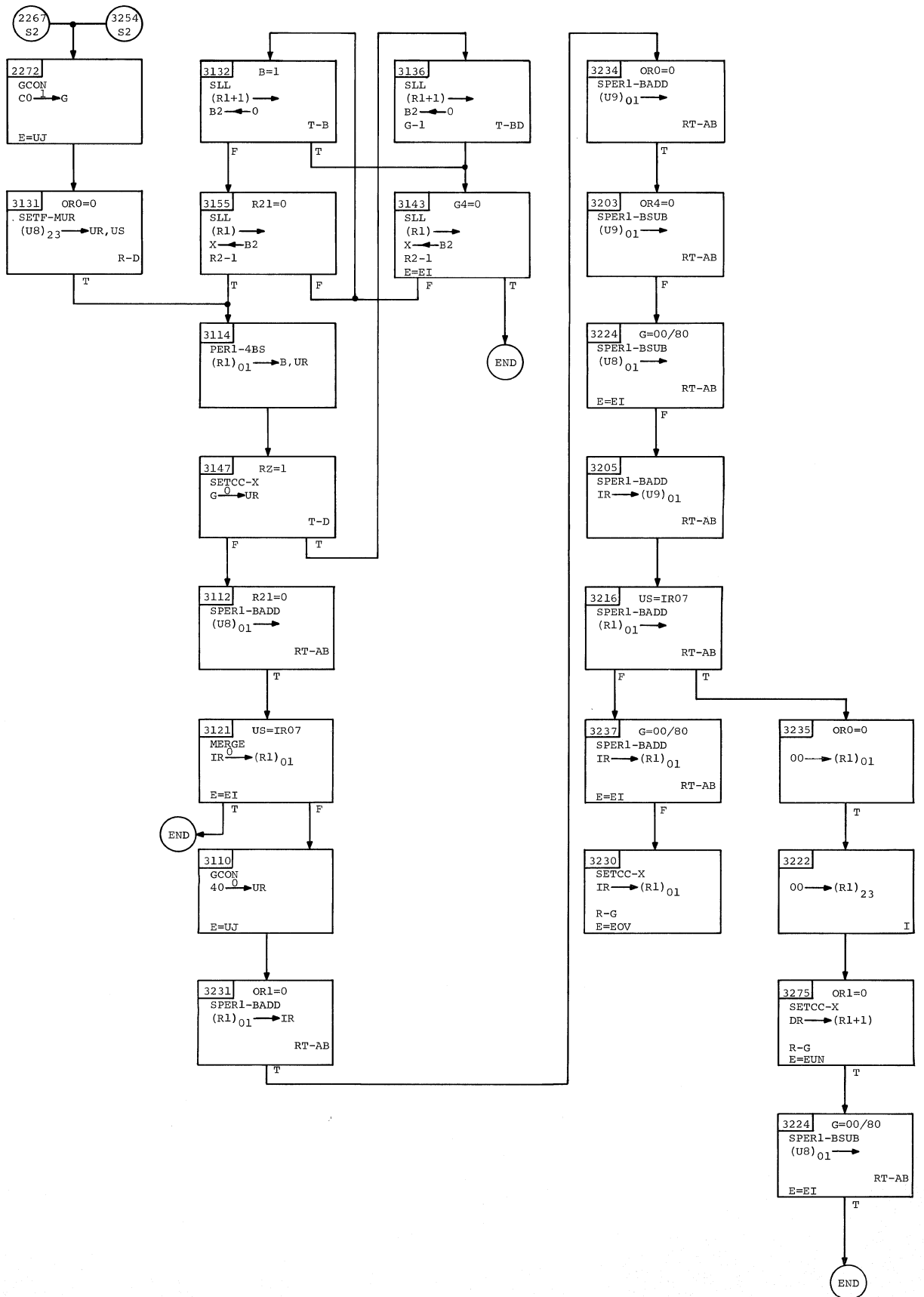


SUBTRACT NORMALIZED SHORT SNER RR 3B

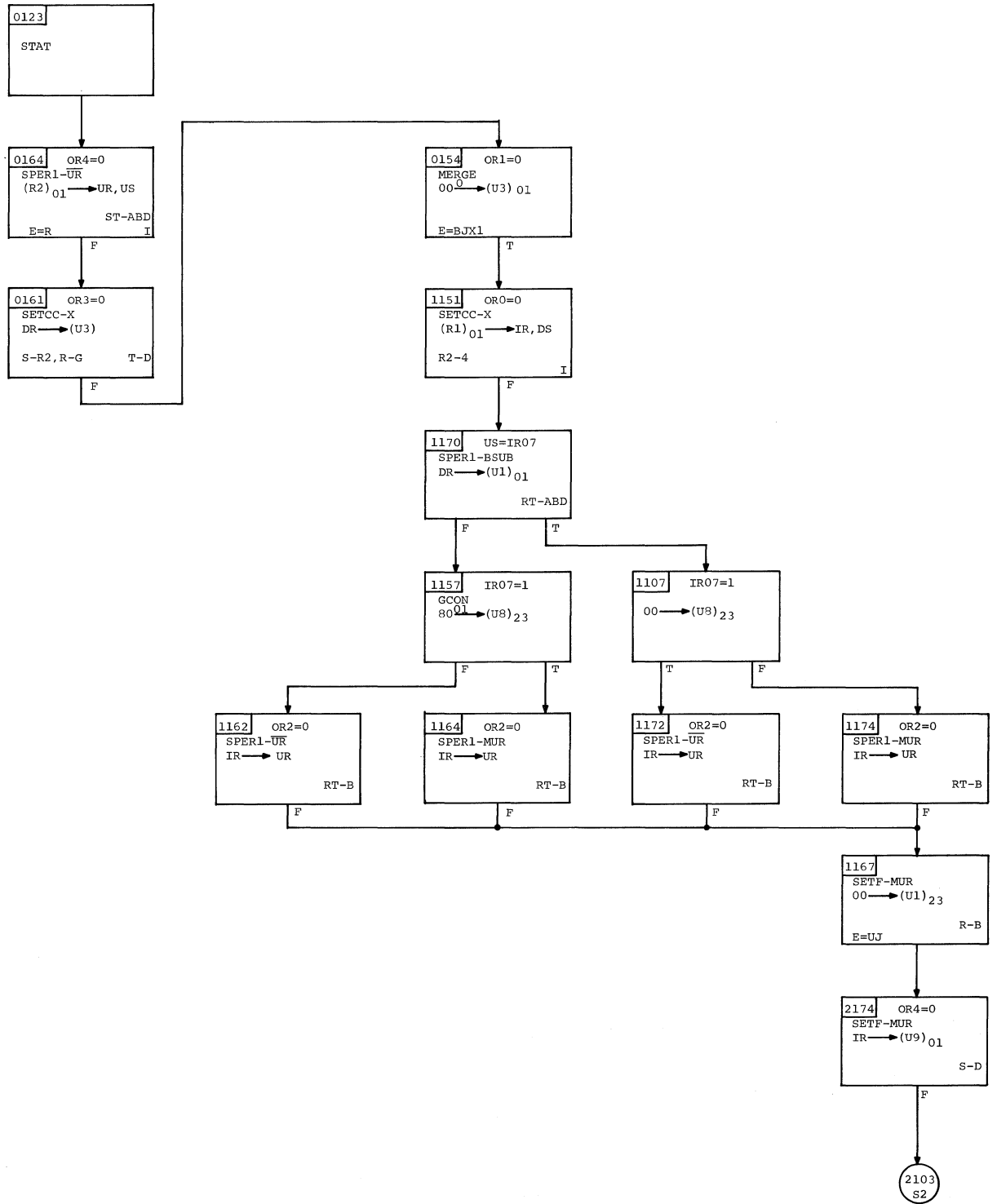


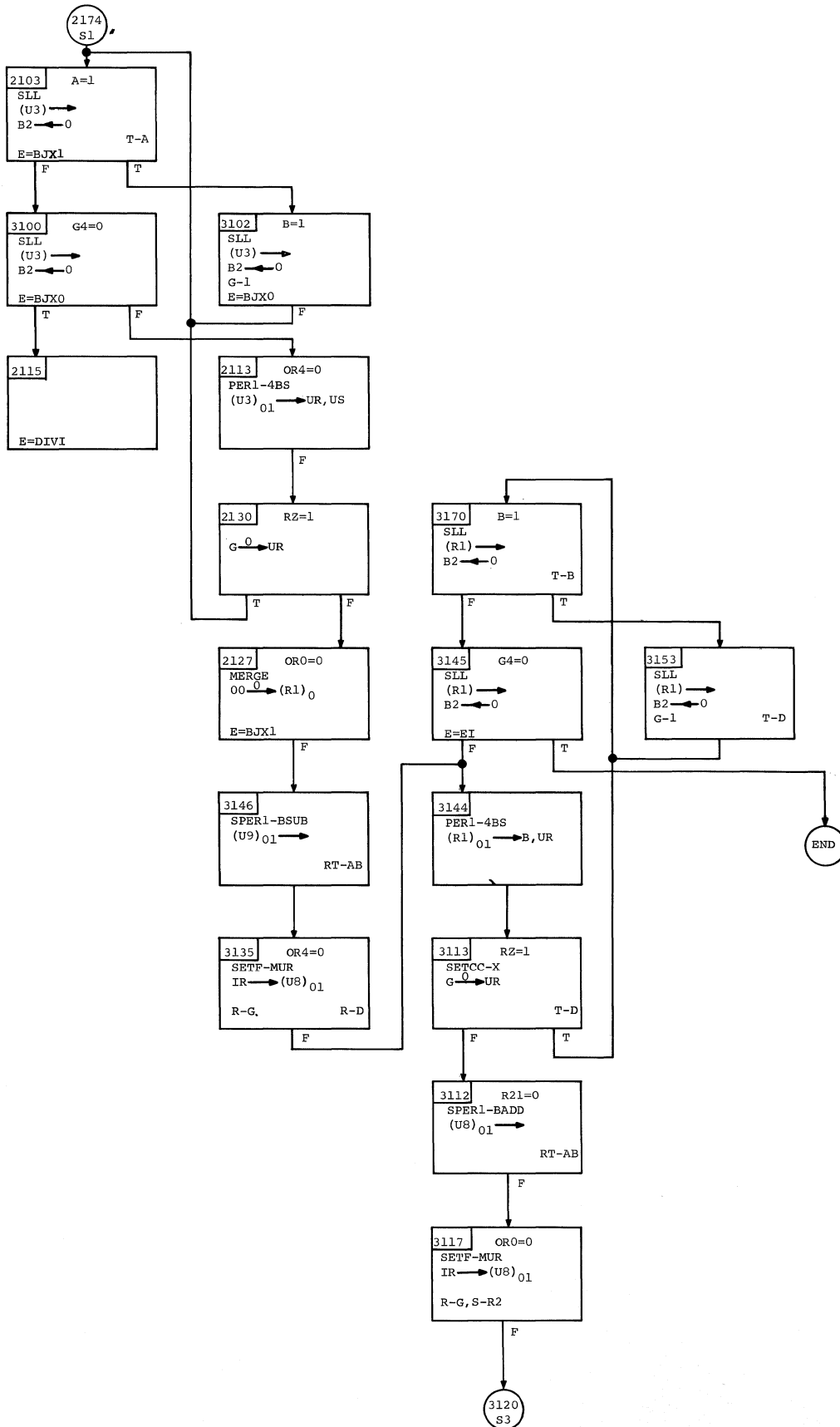


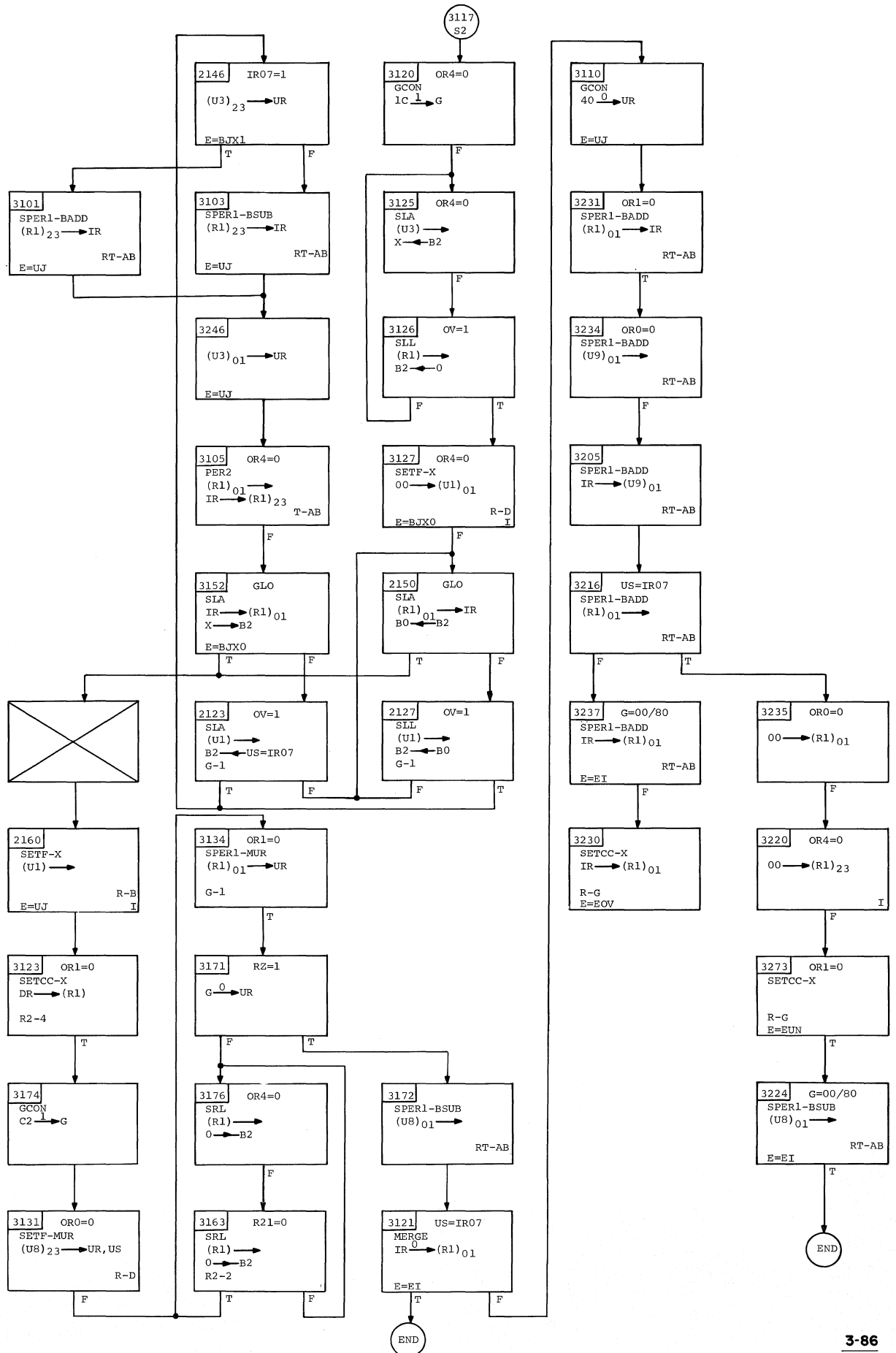


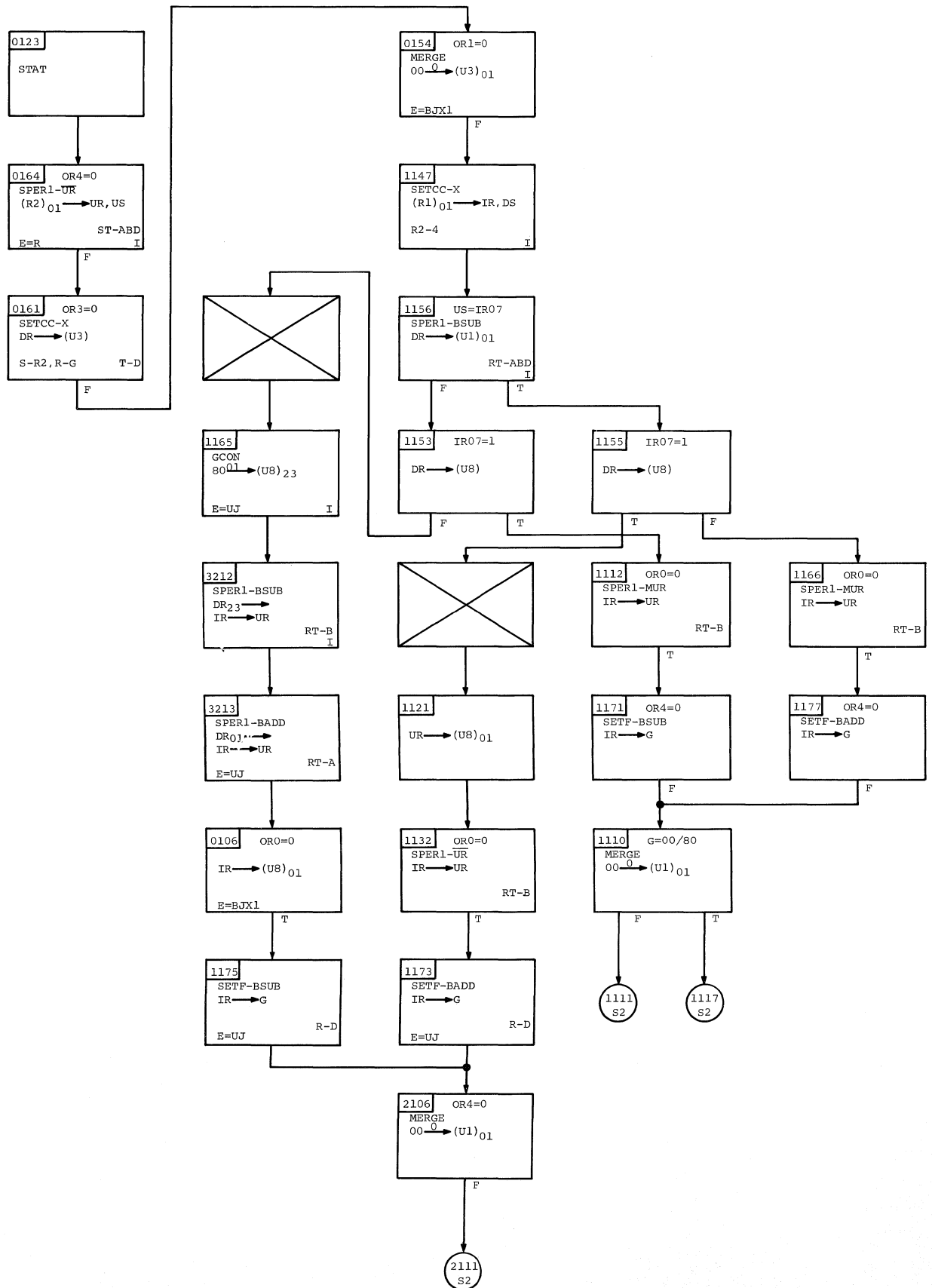


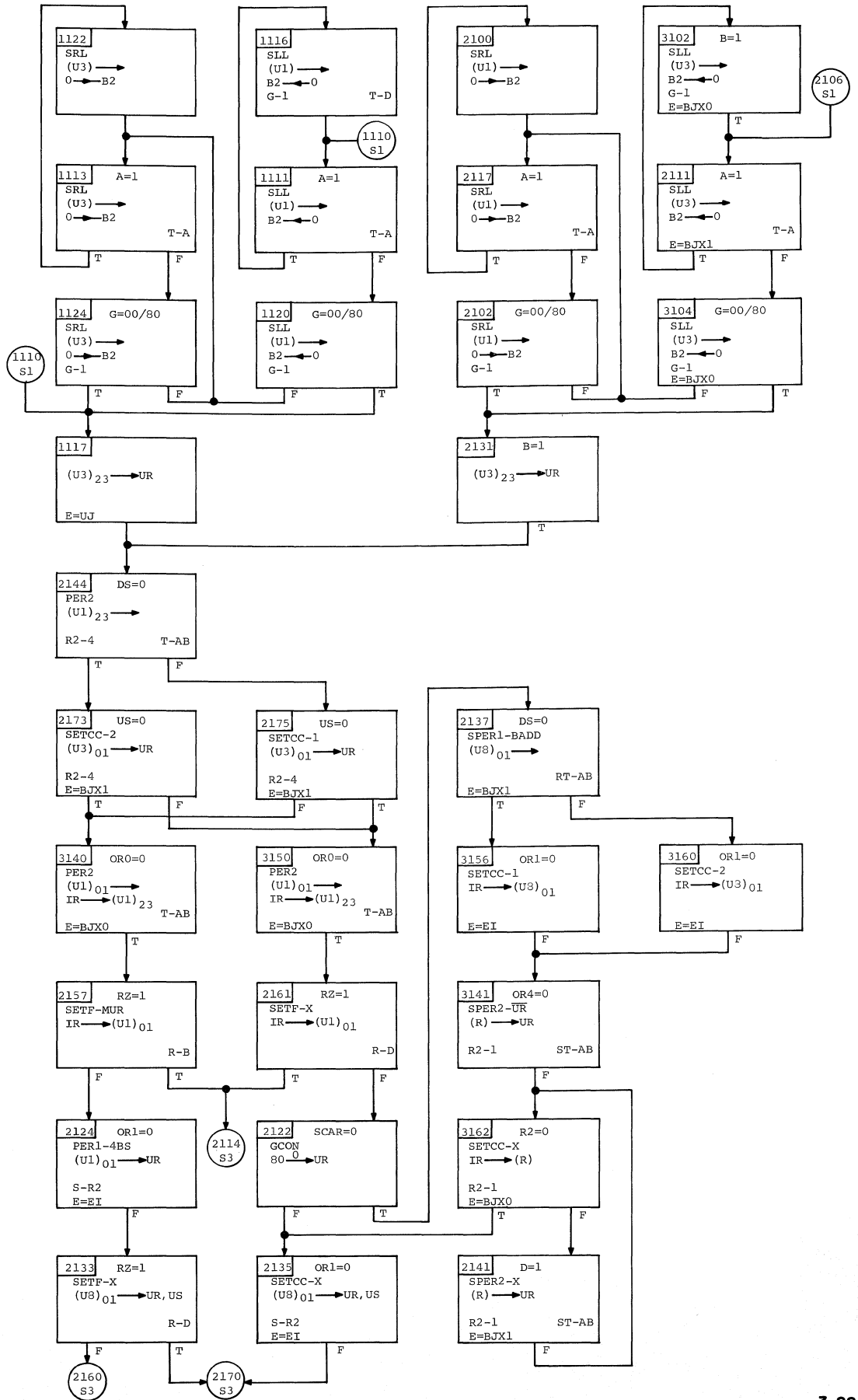


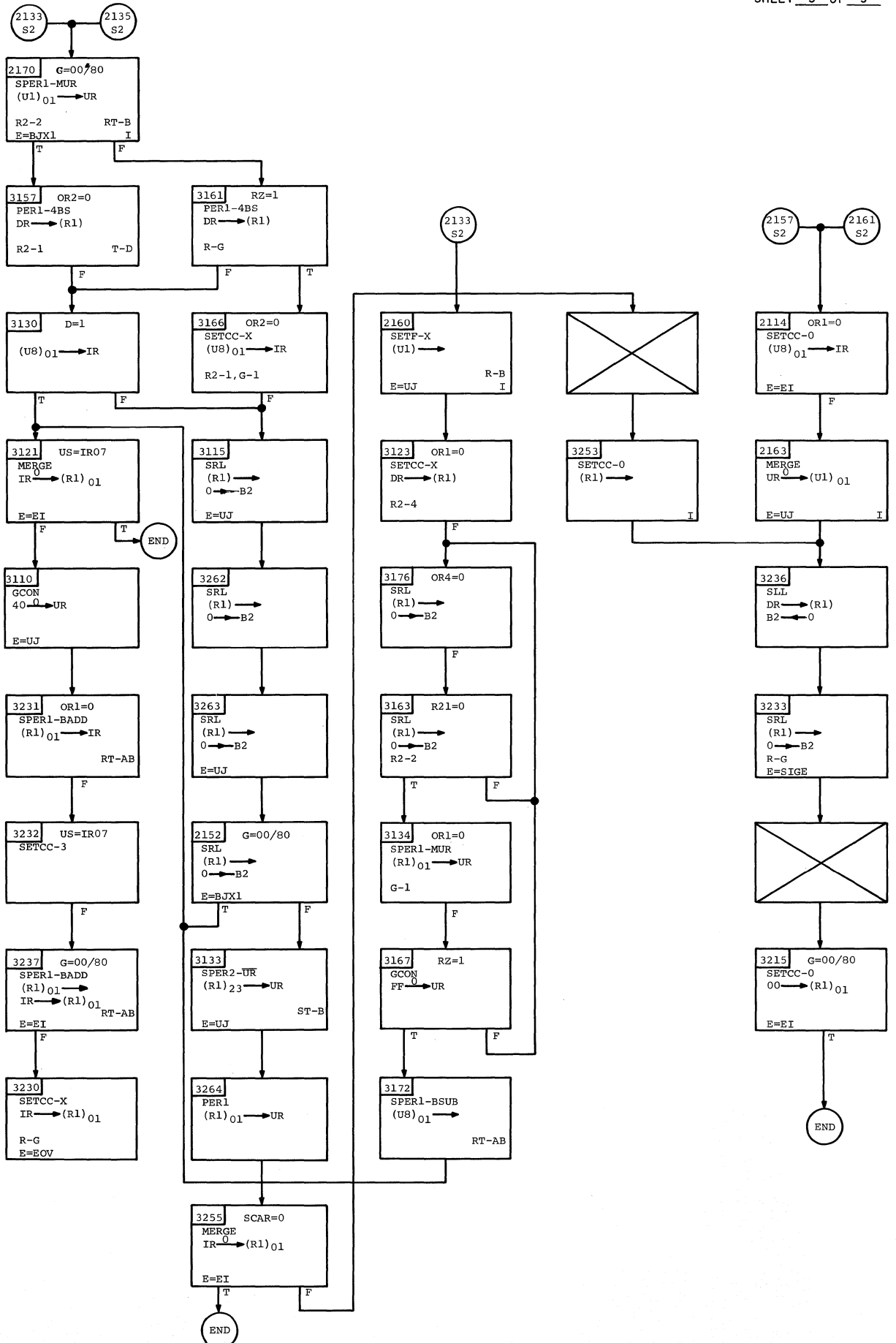


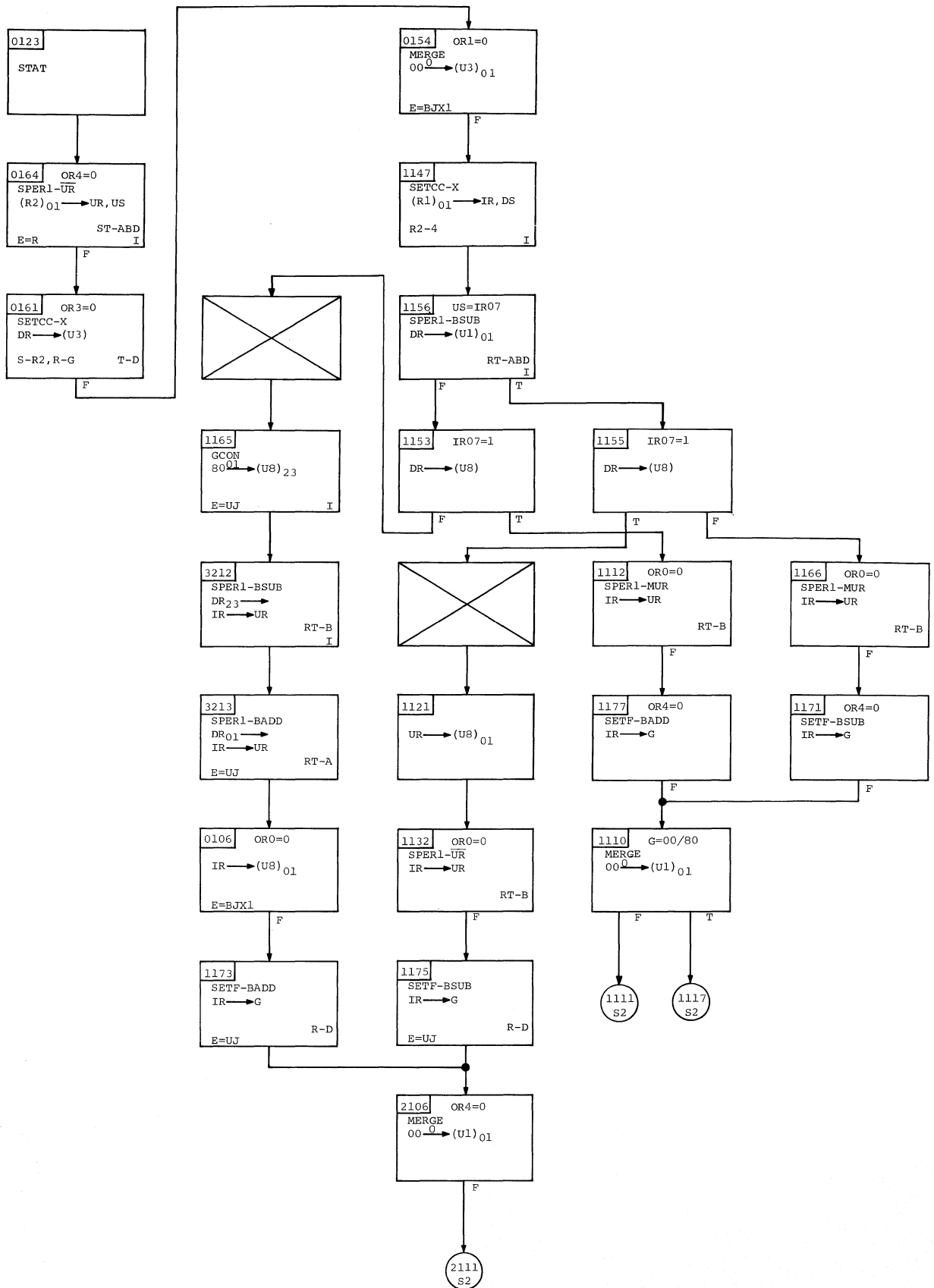




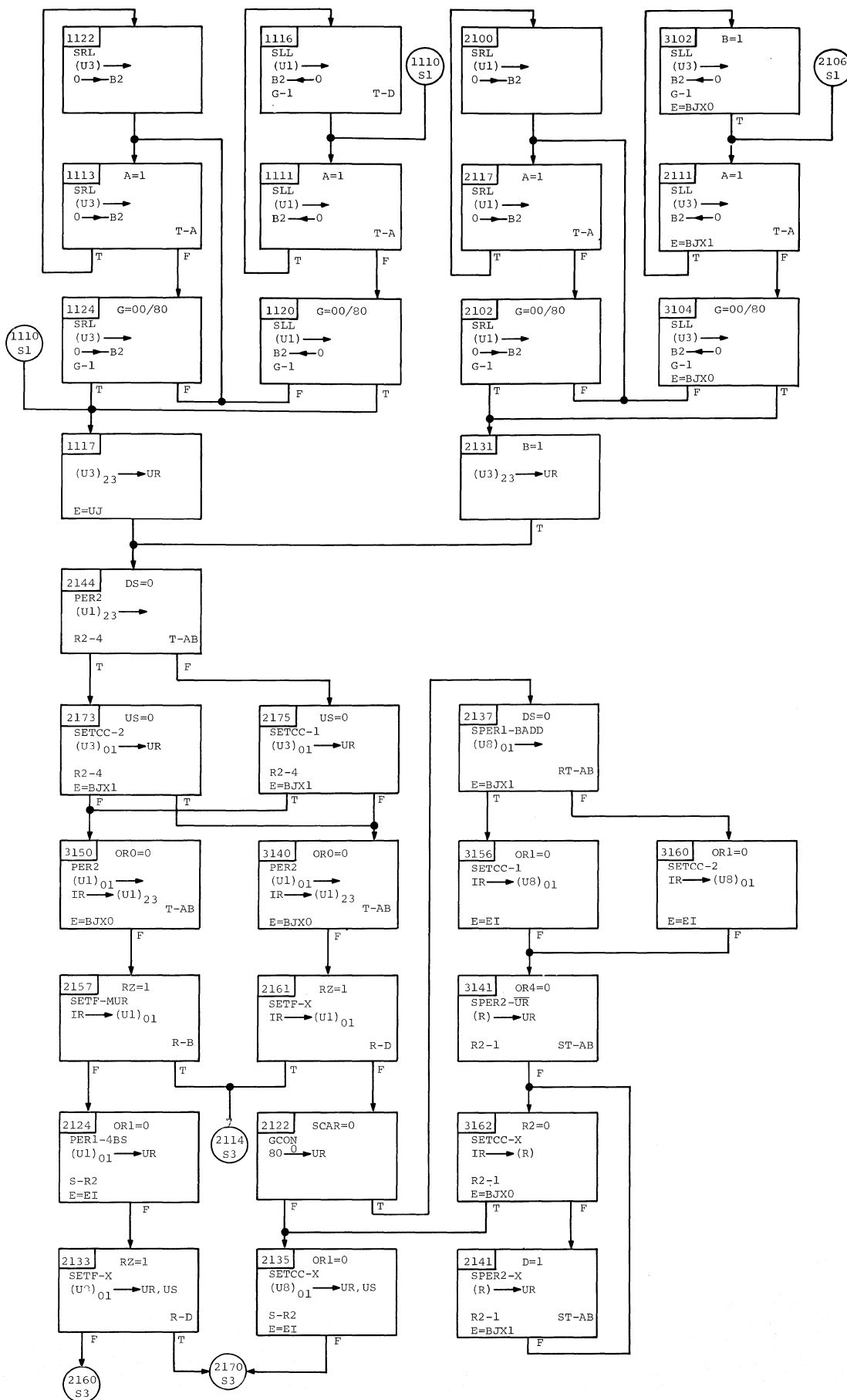




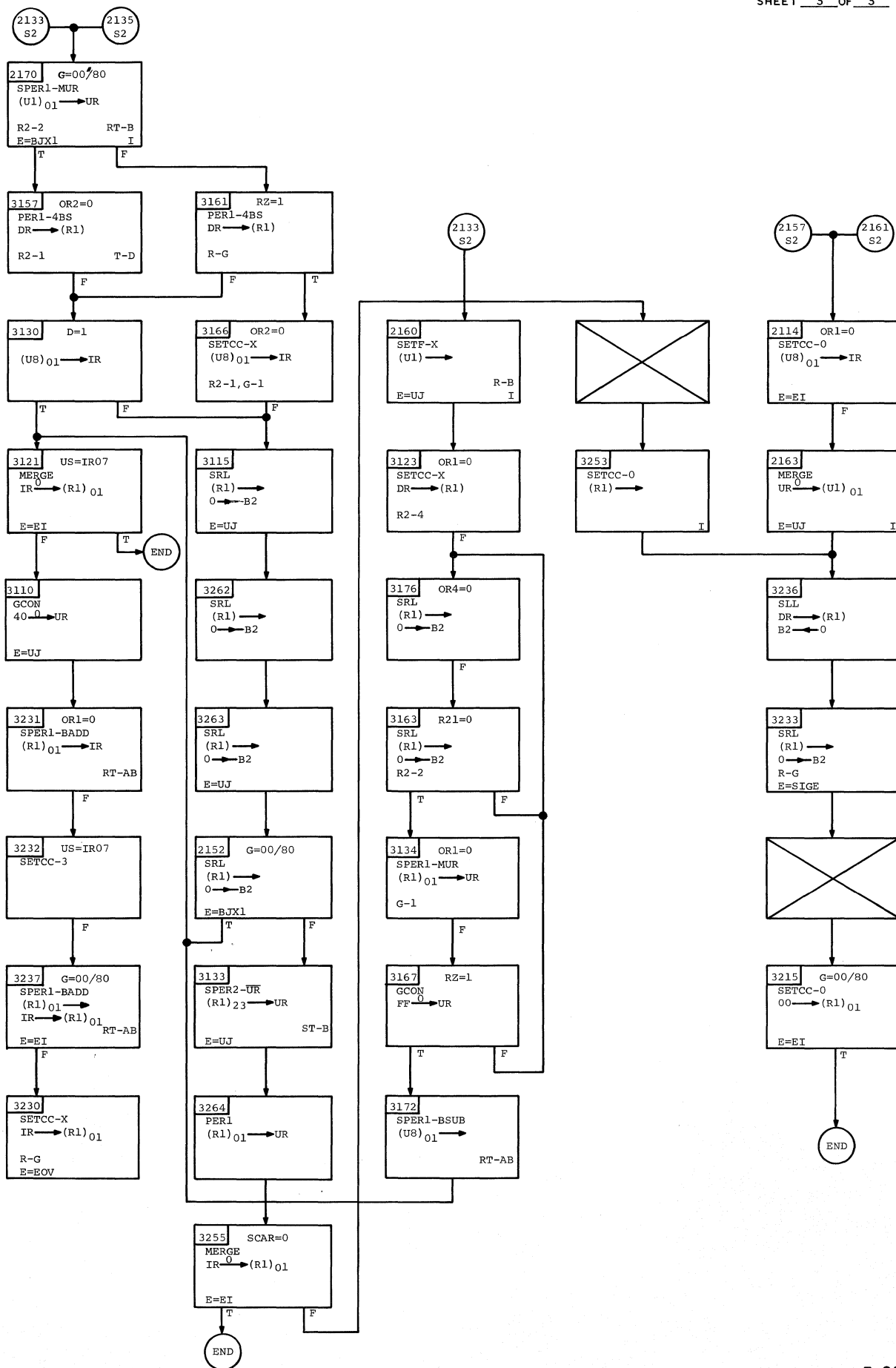


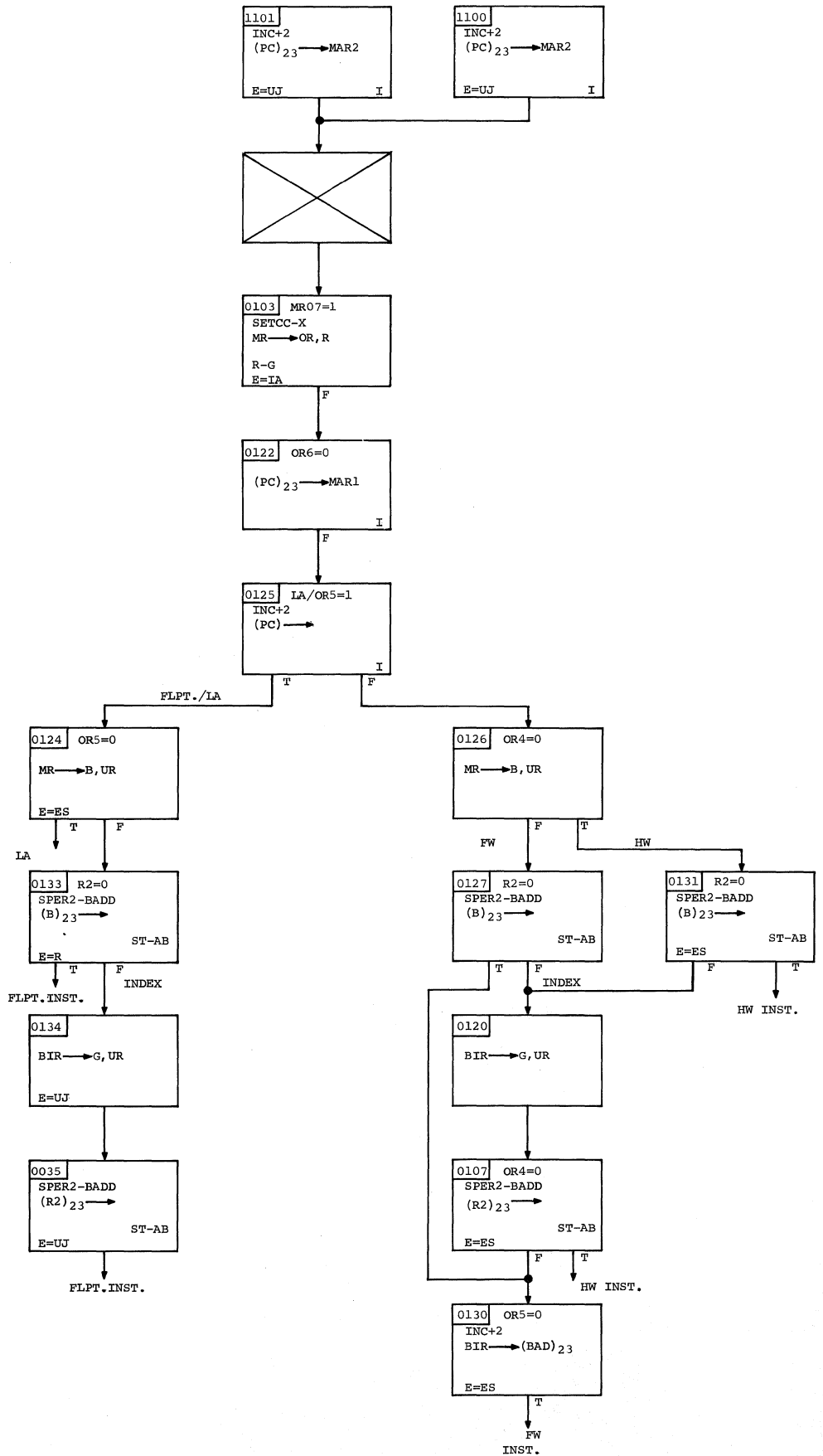


SUBTRACT UNNORMALIZED SHORT SUR RR 3F

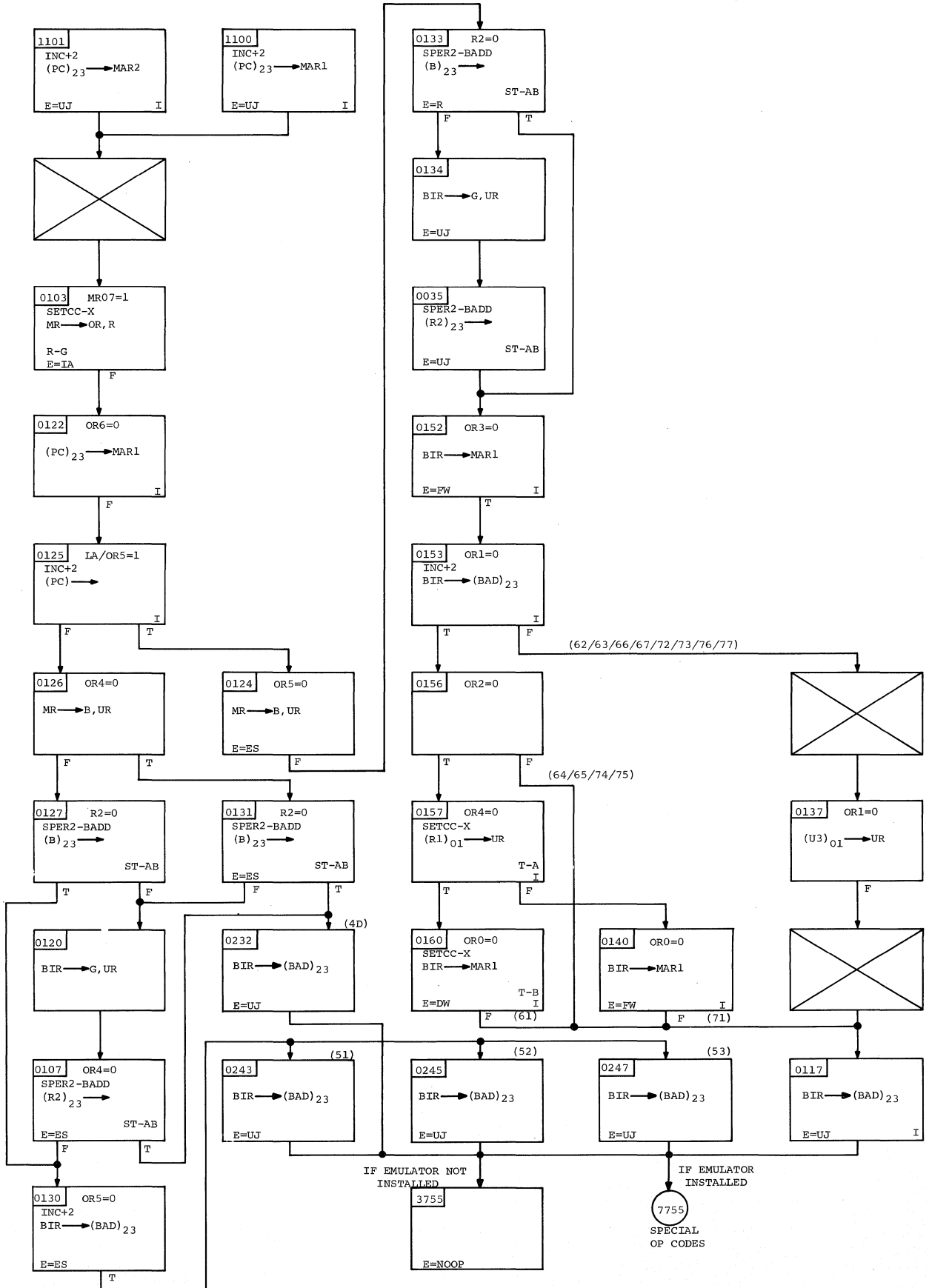




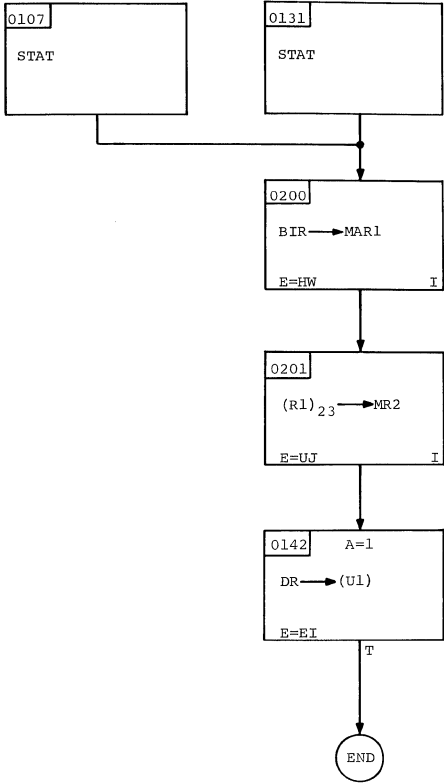


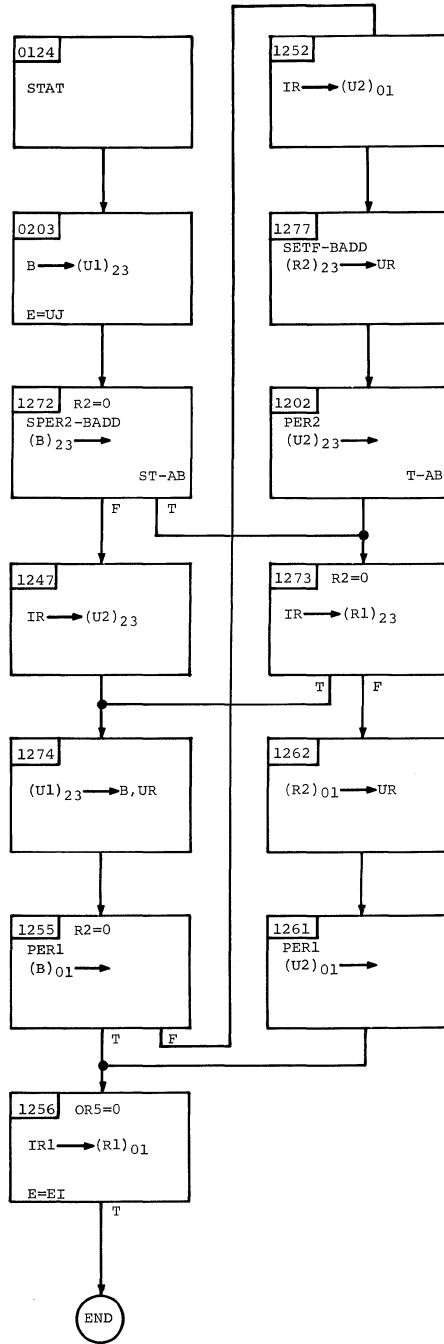


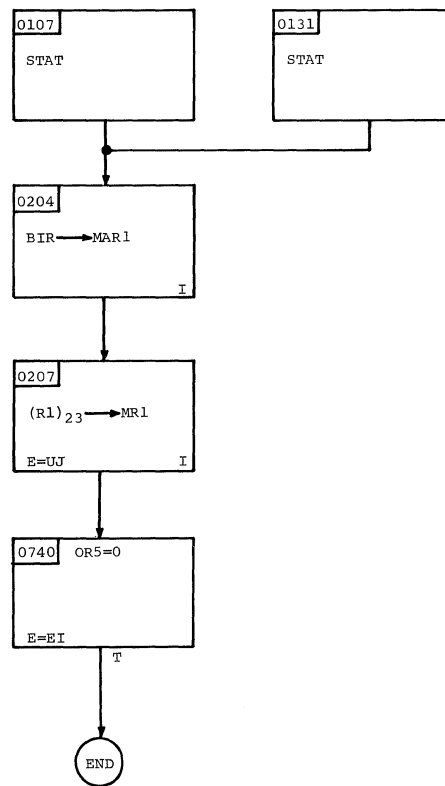
# NO OPERATION INSTRUCTIONS RX FORMAT

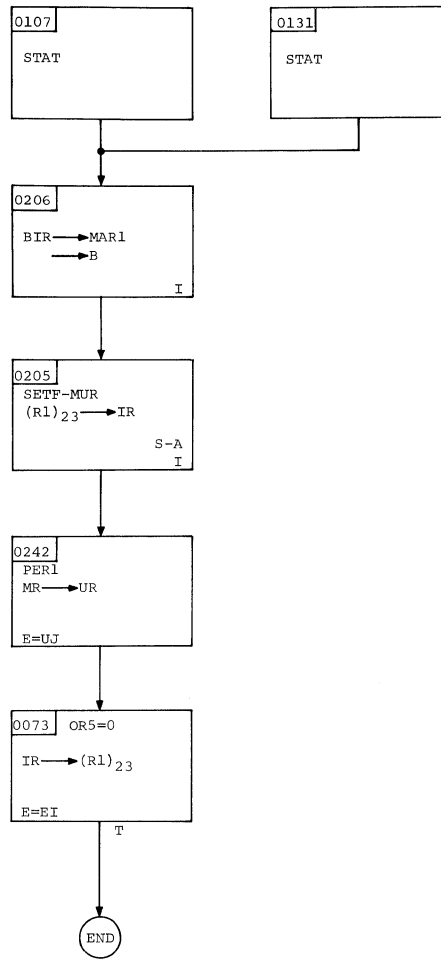


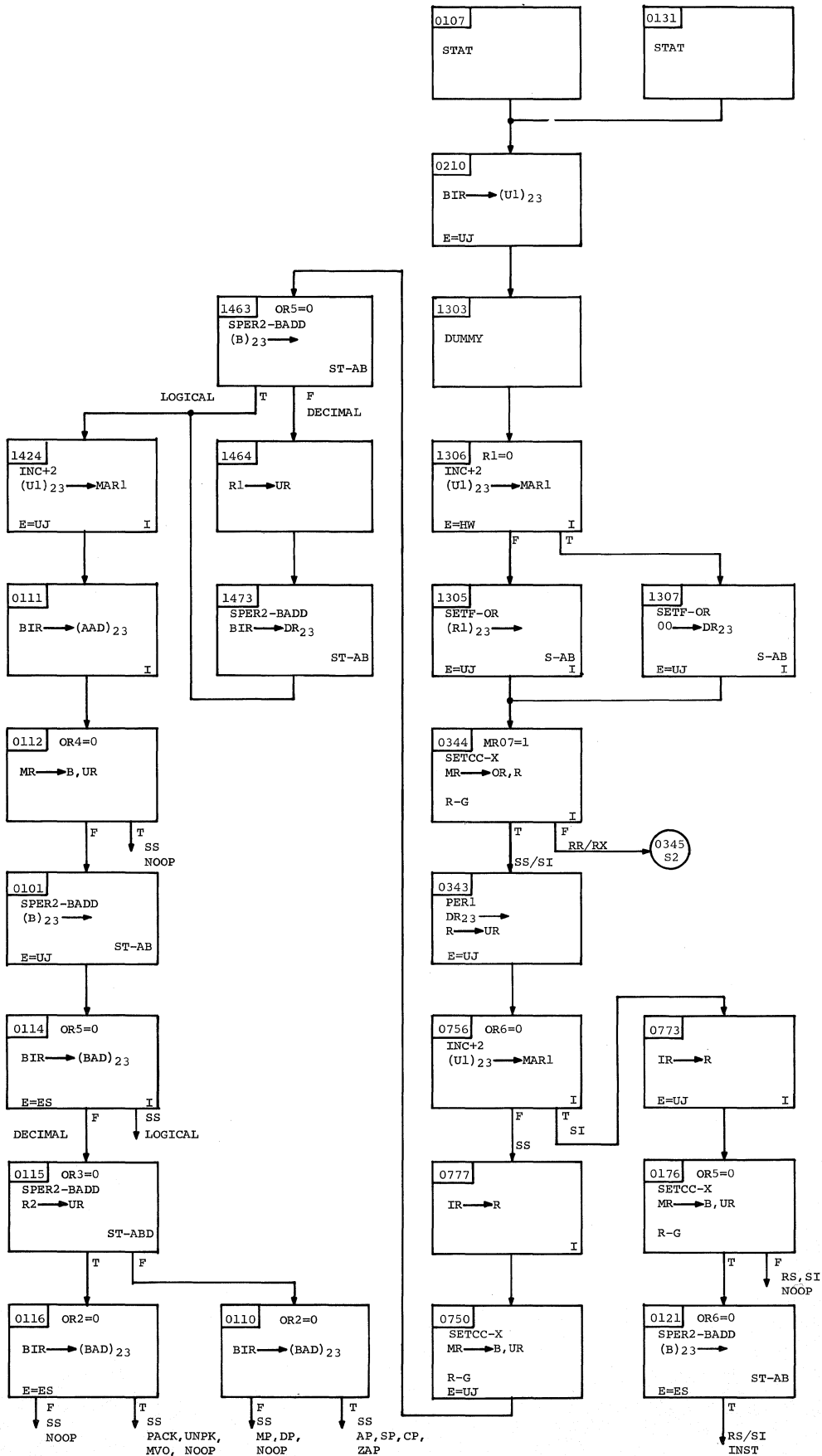
STORE HALFWORD STH RX 40



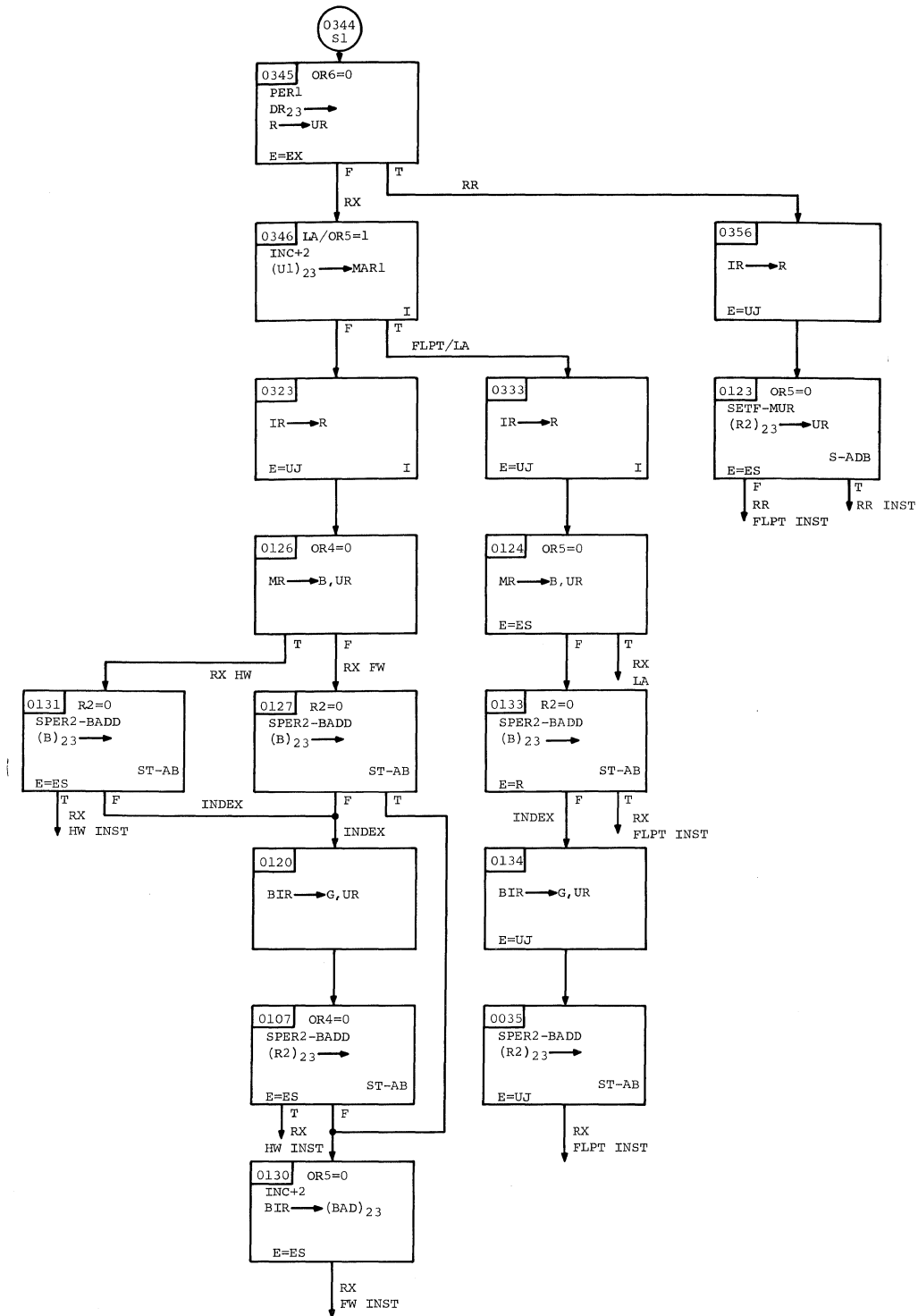


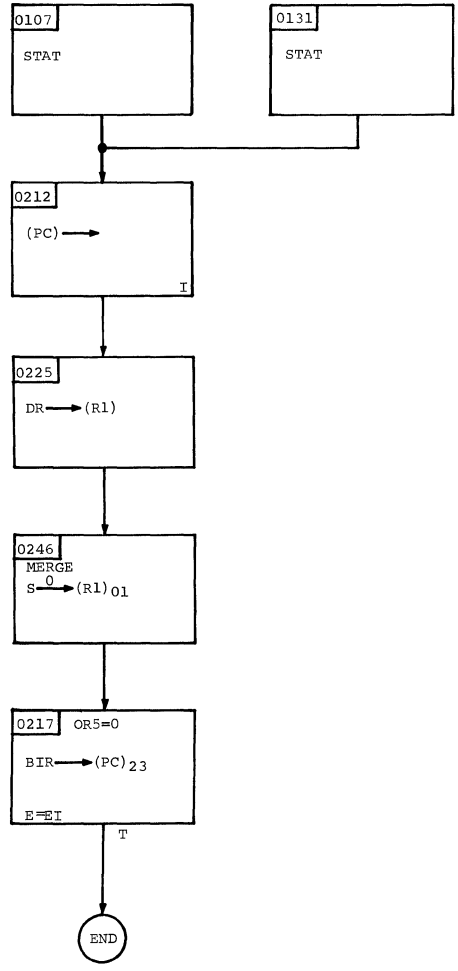


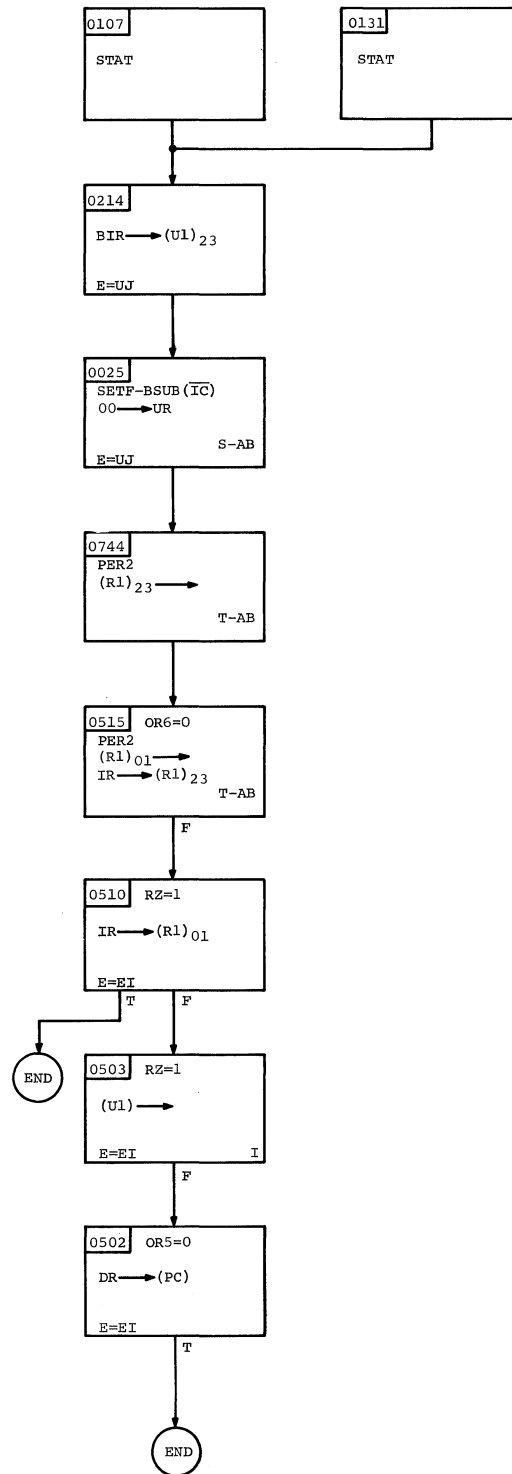


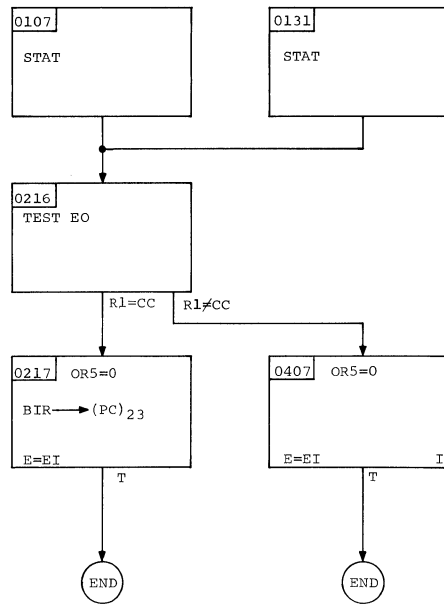


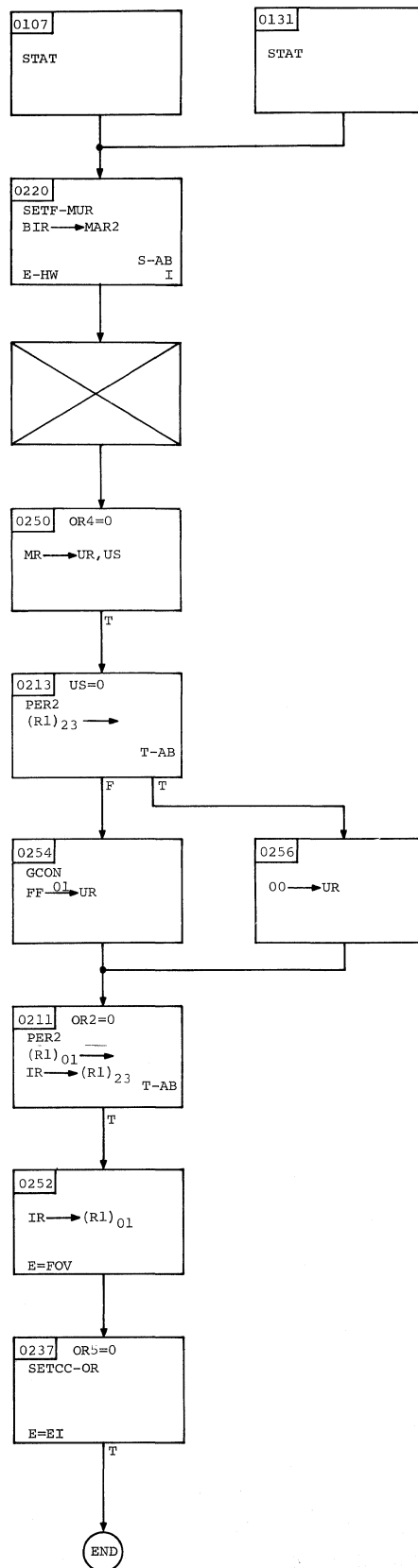


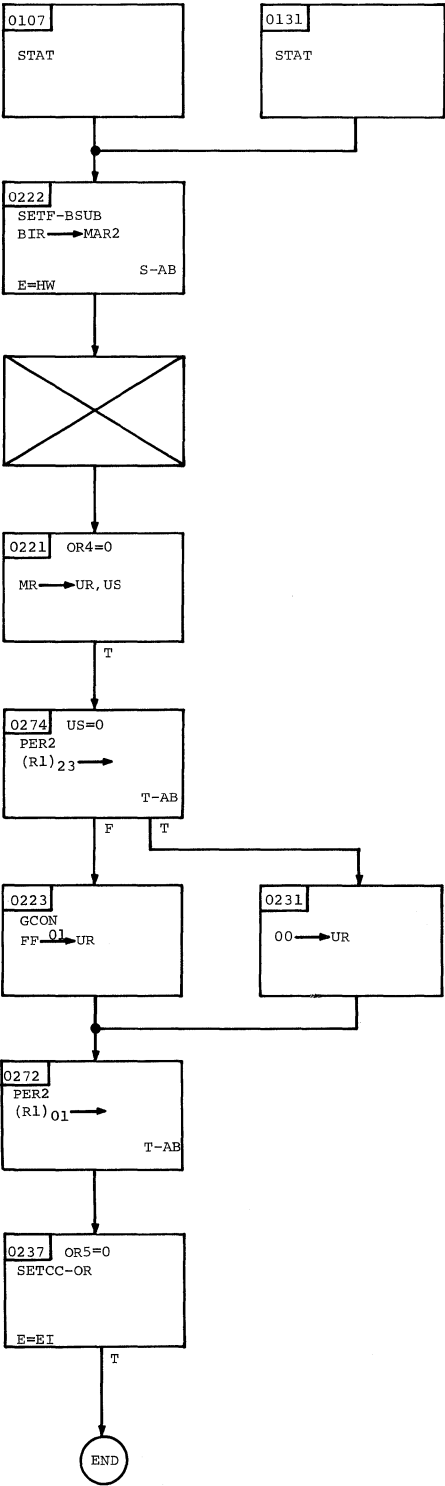


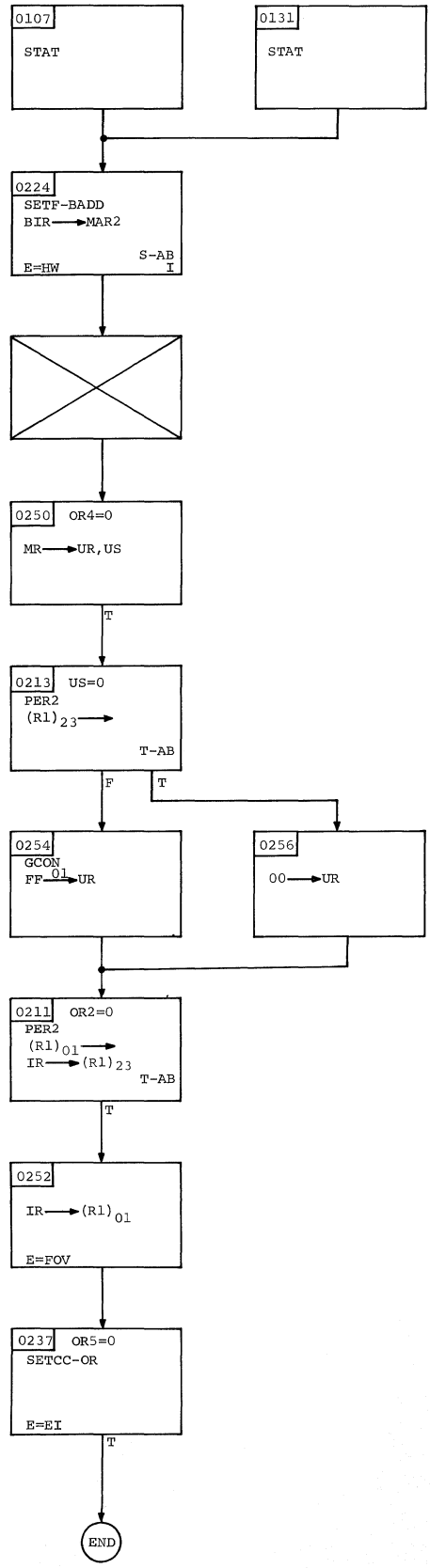


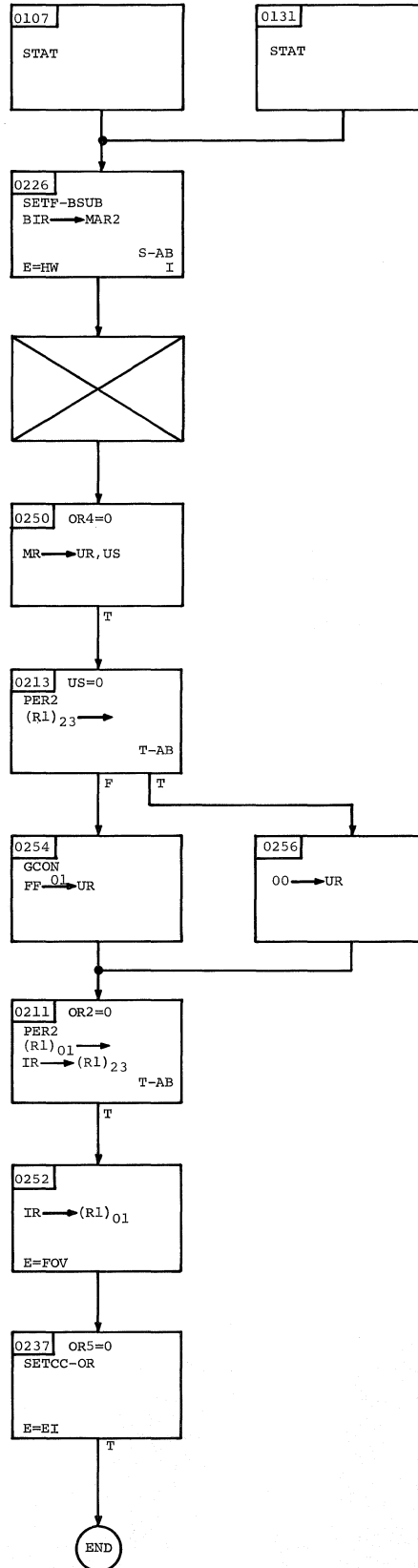




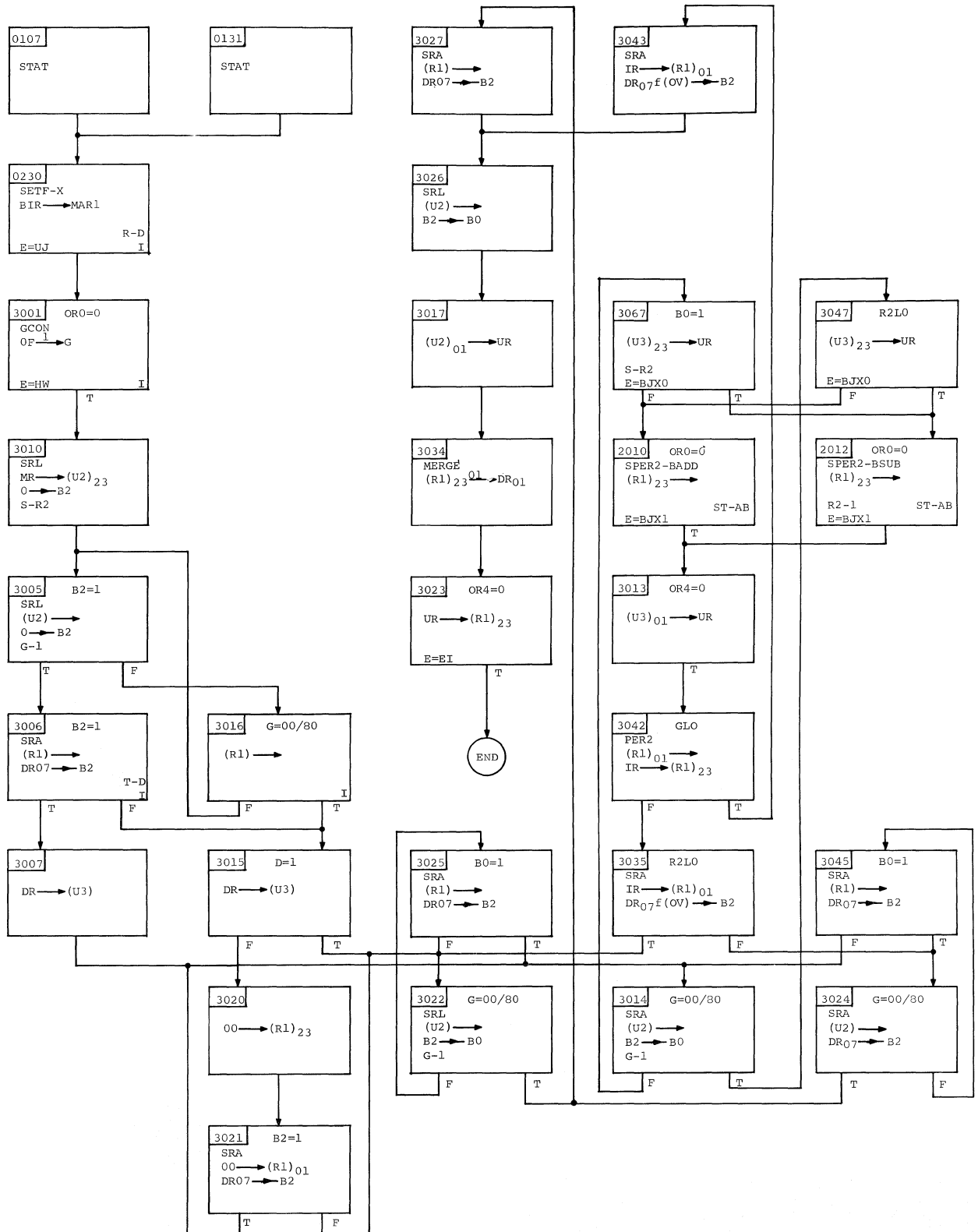


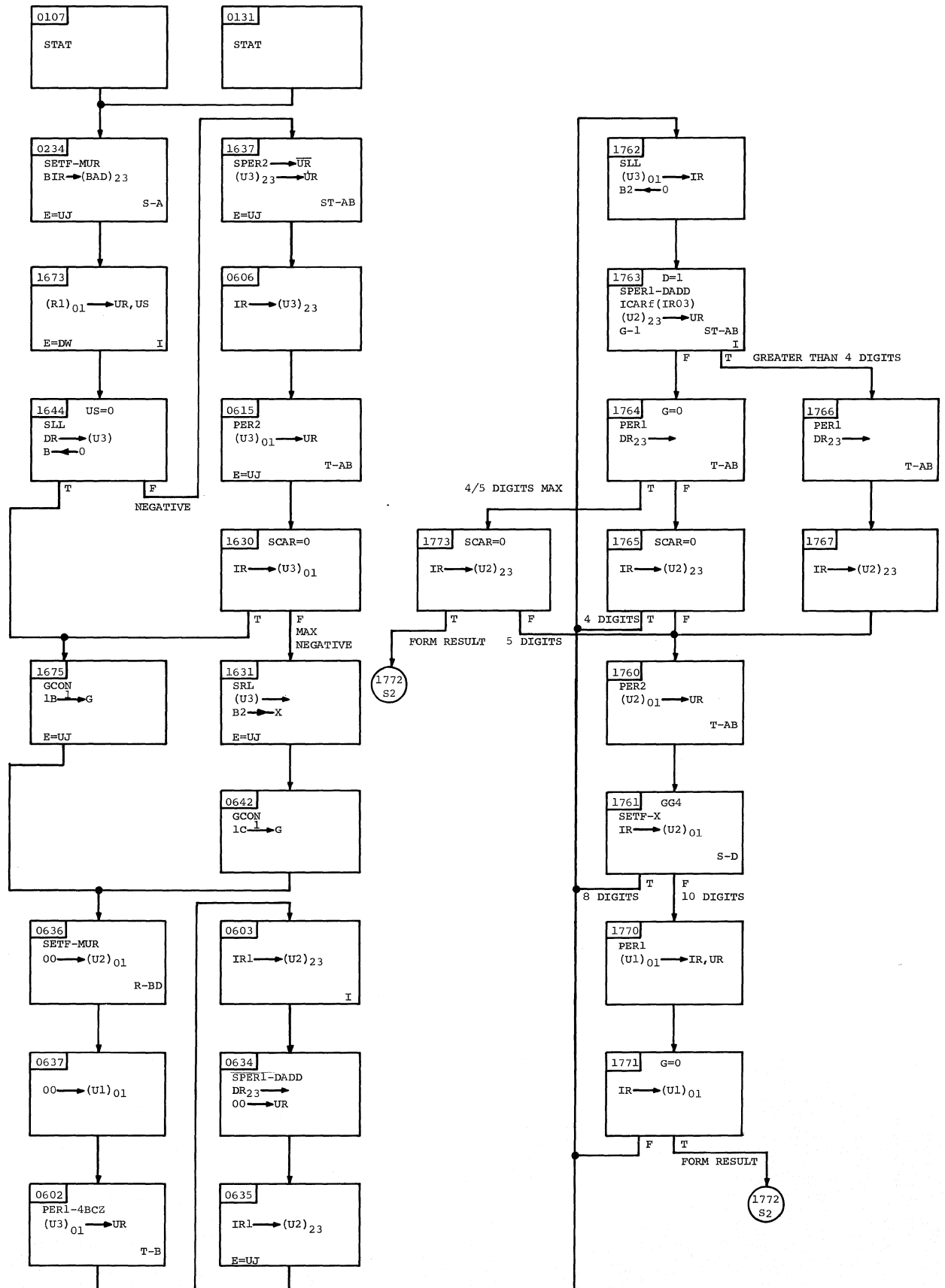




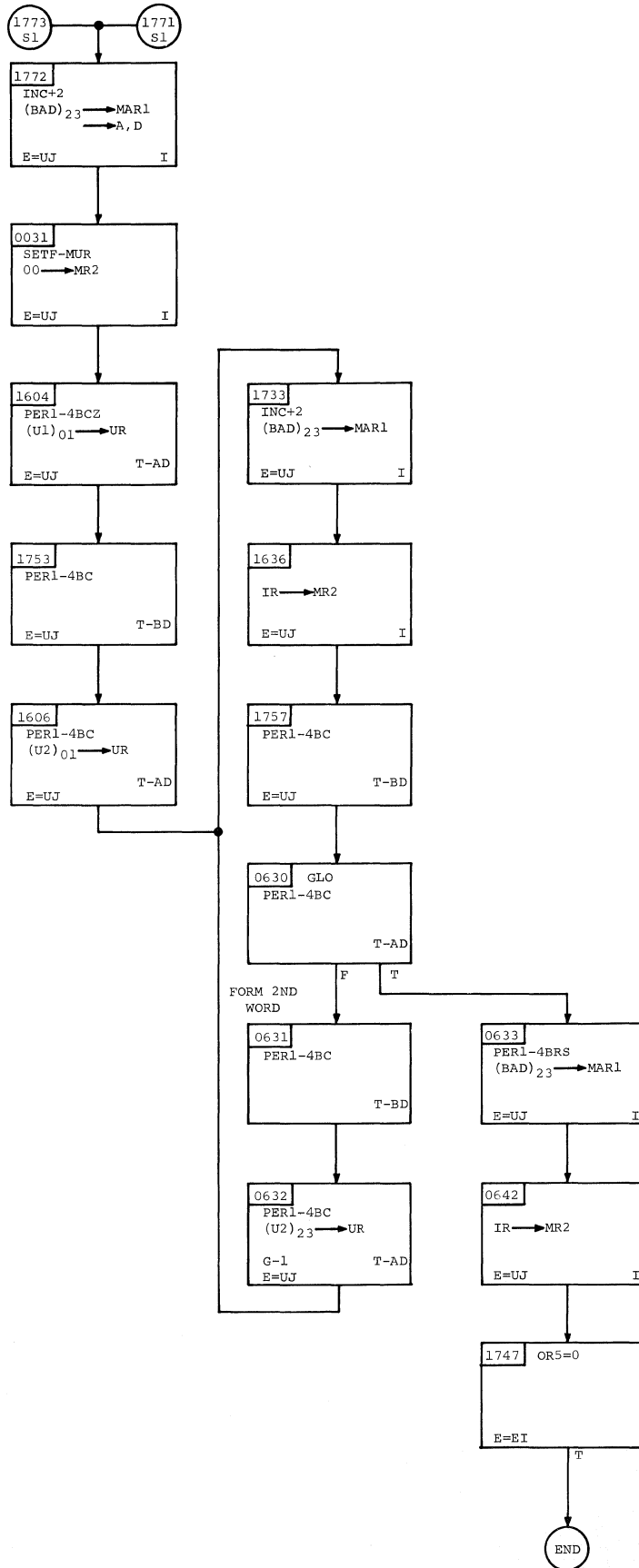


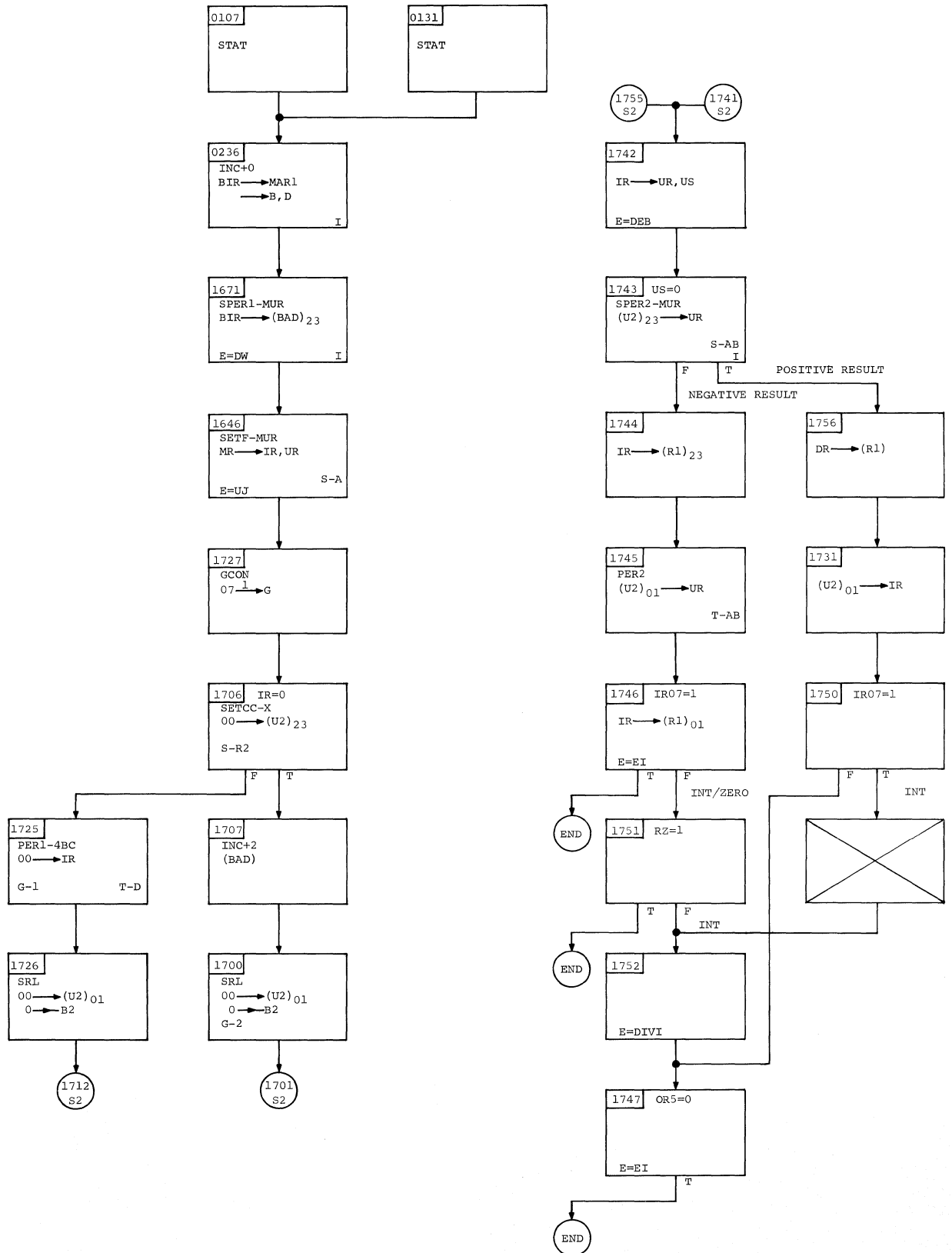


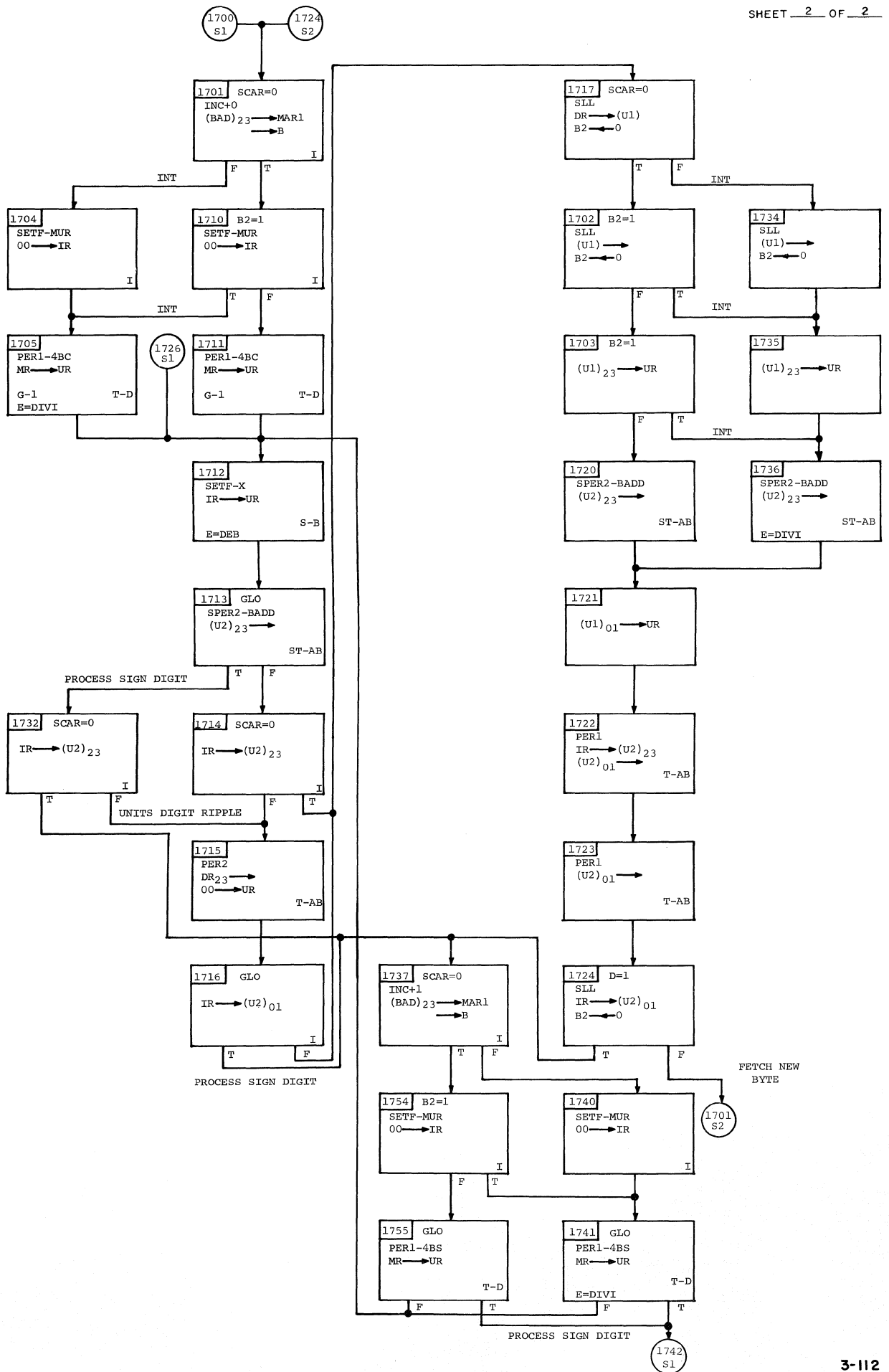


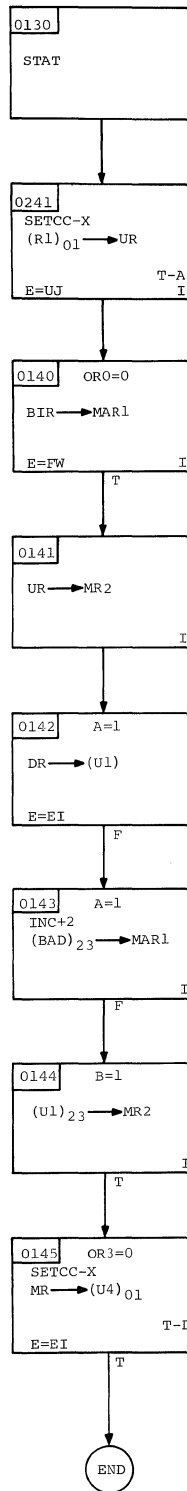


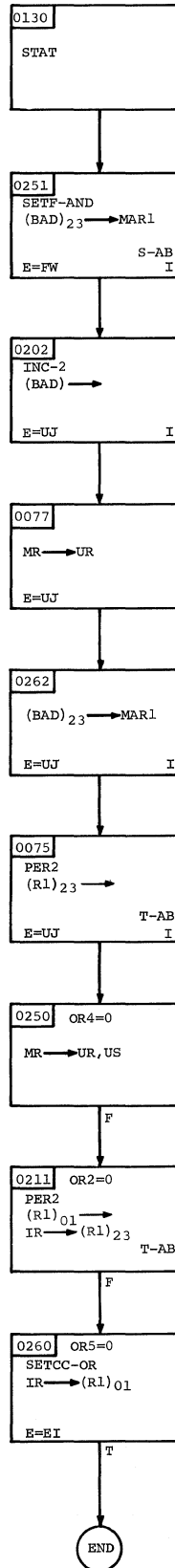
CONVERT TO DECIMAL CVD RX 4E

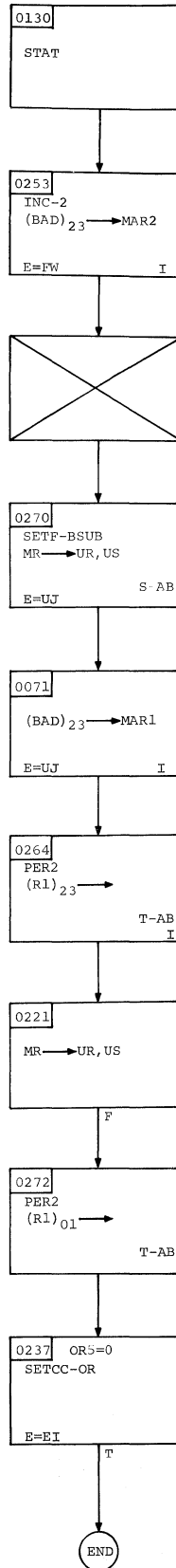




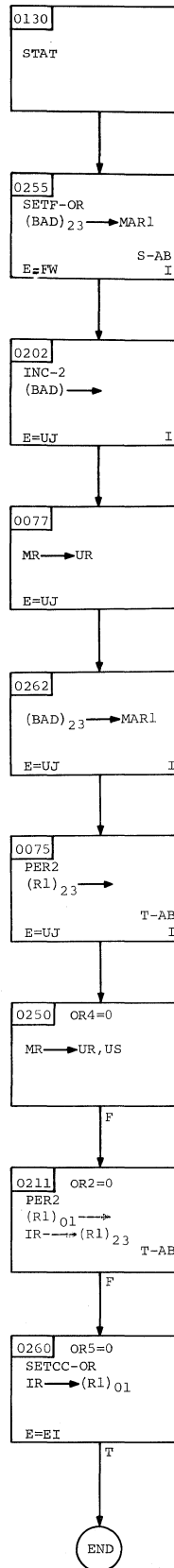


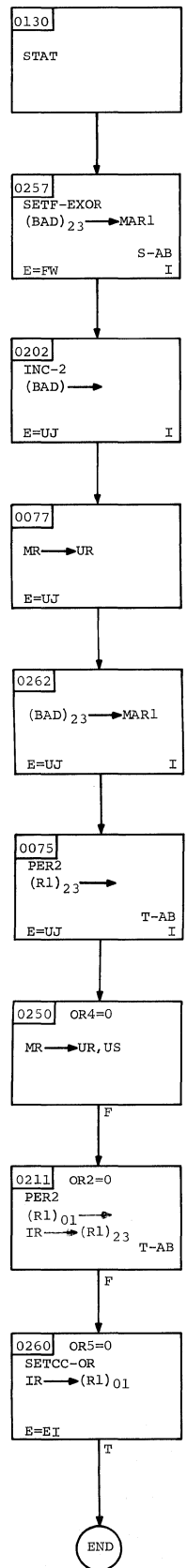


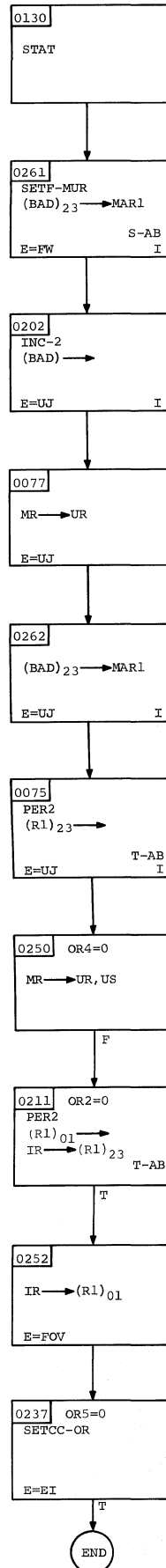


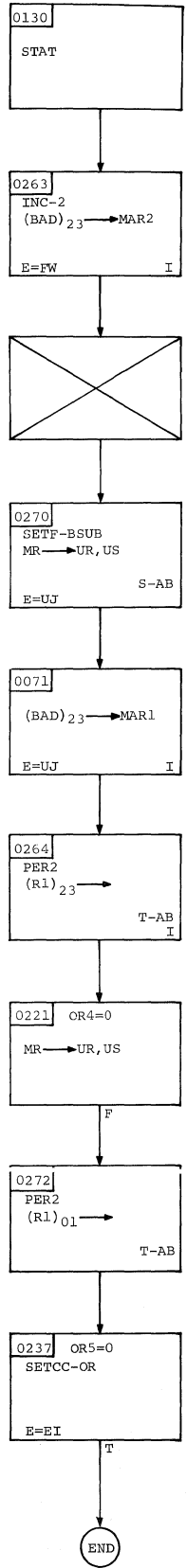


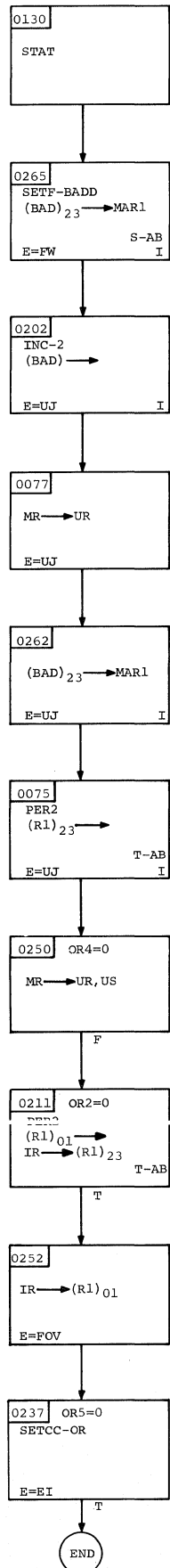


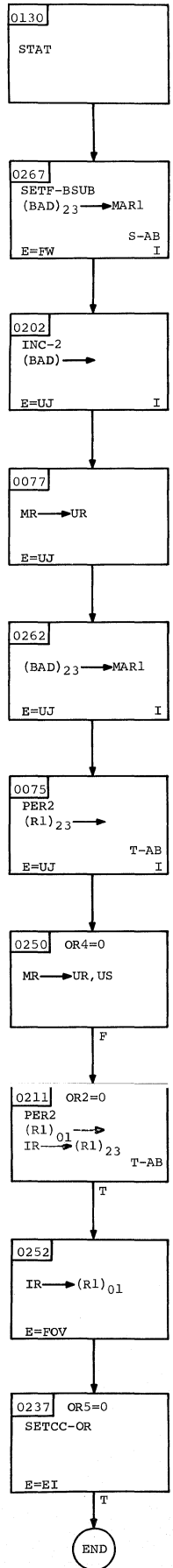


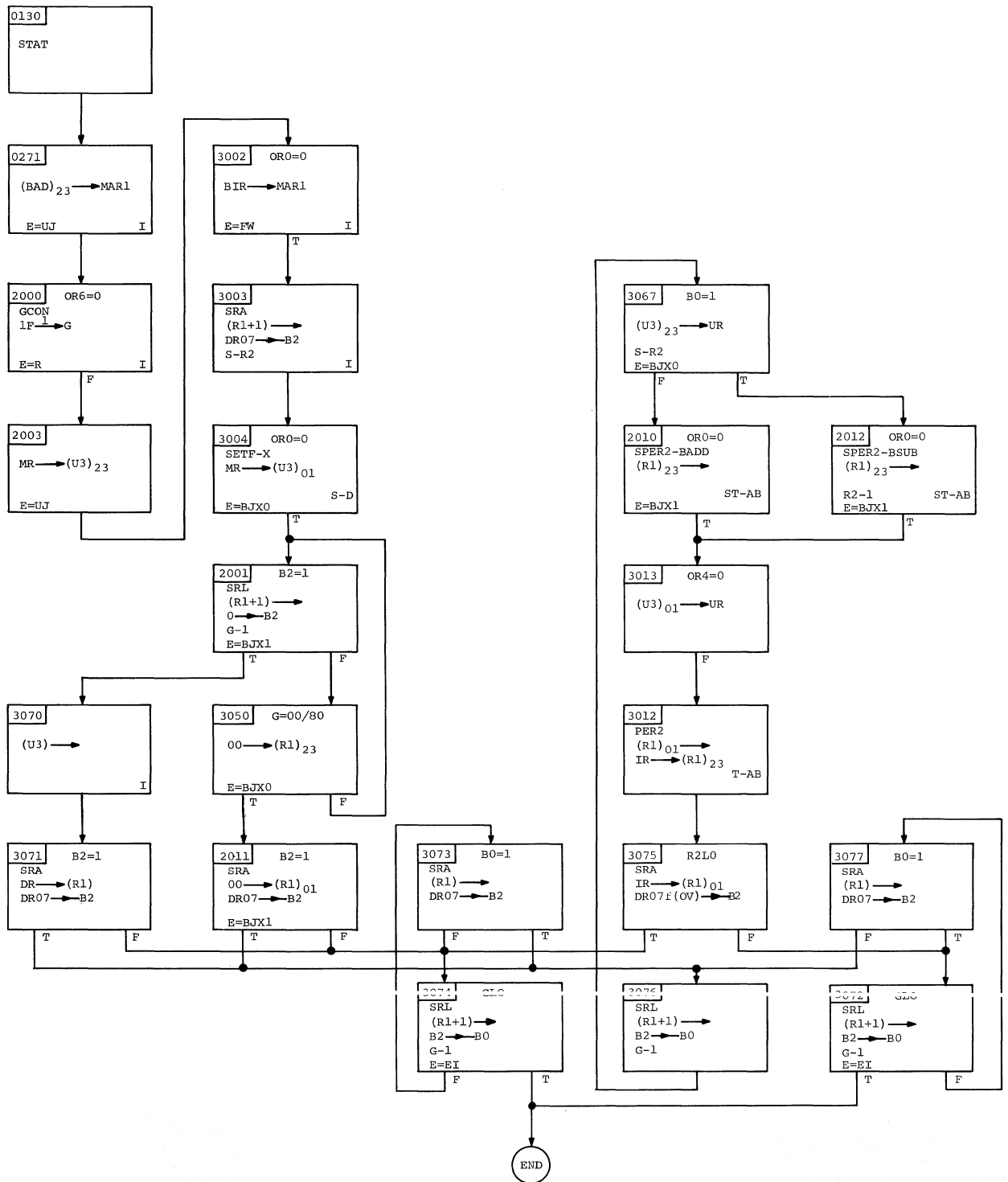


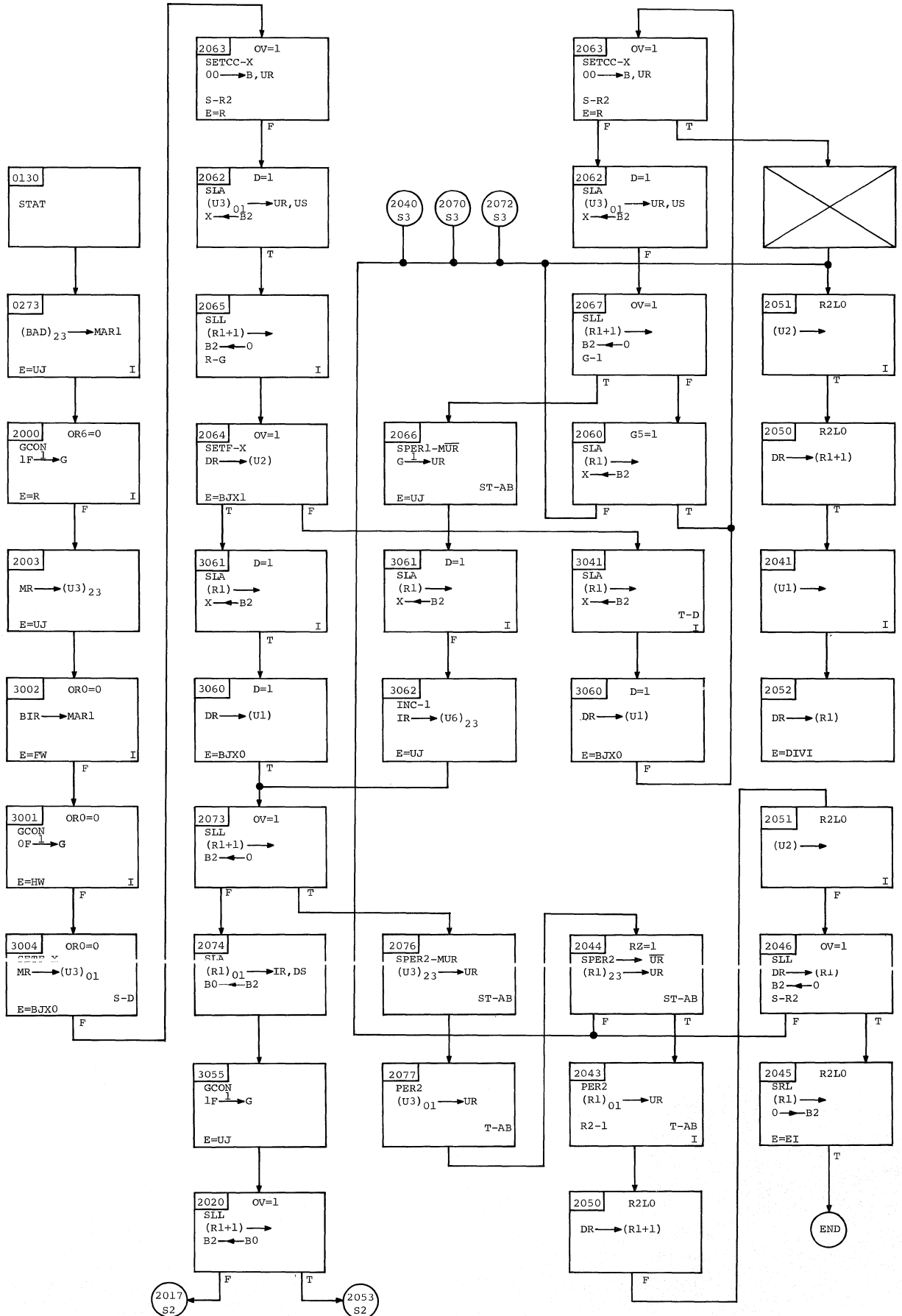




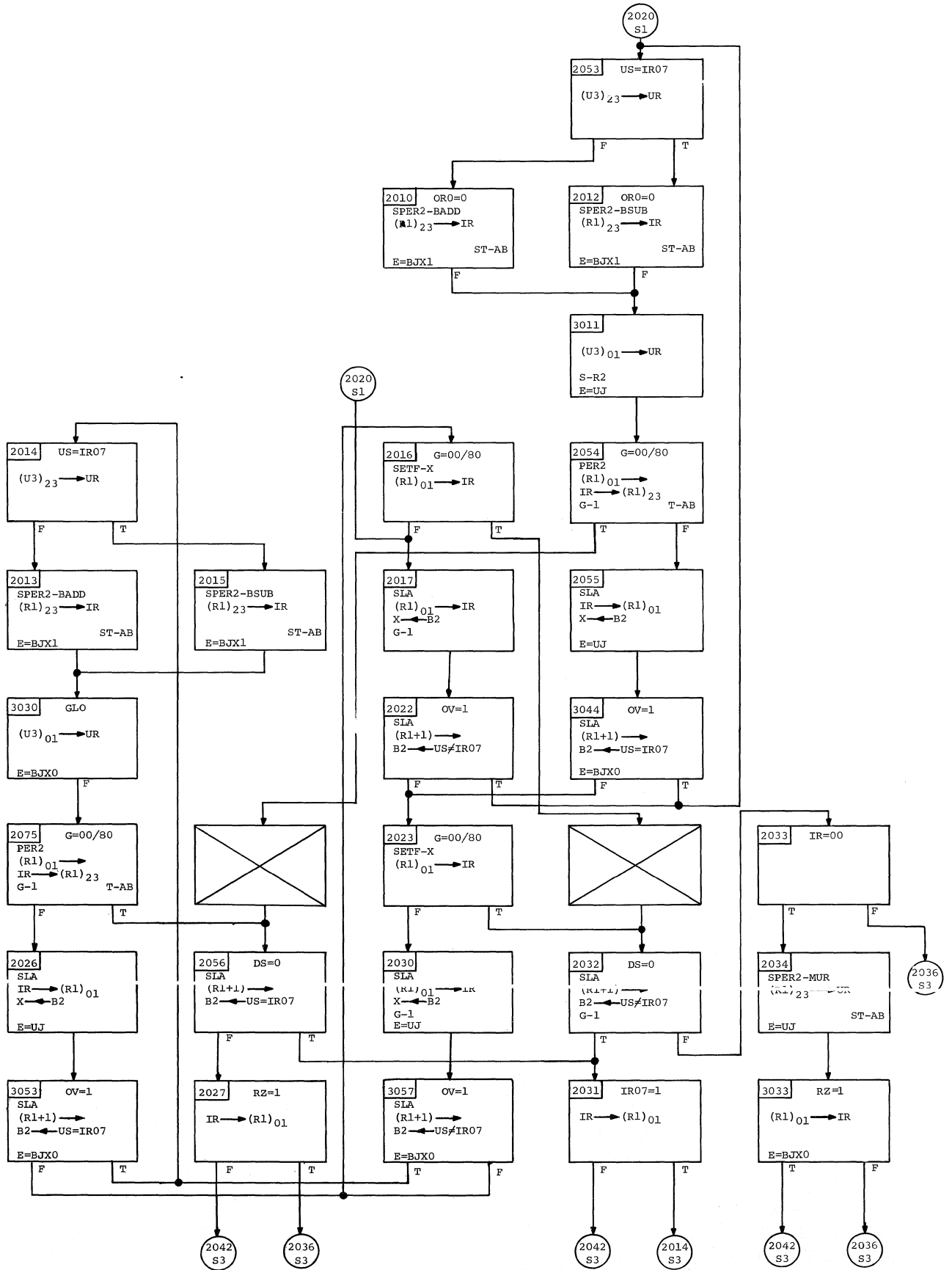


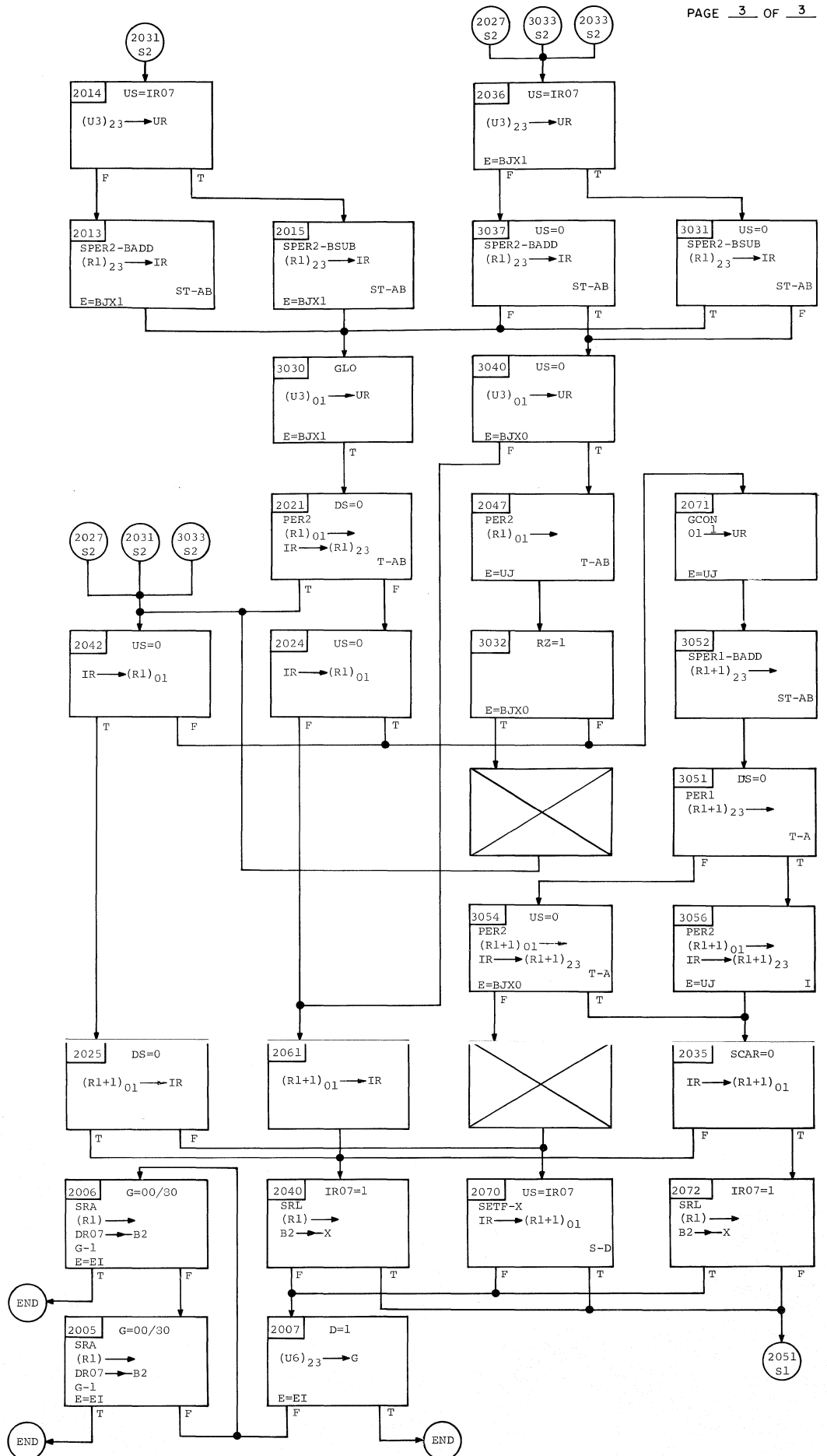


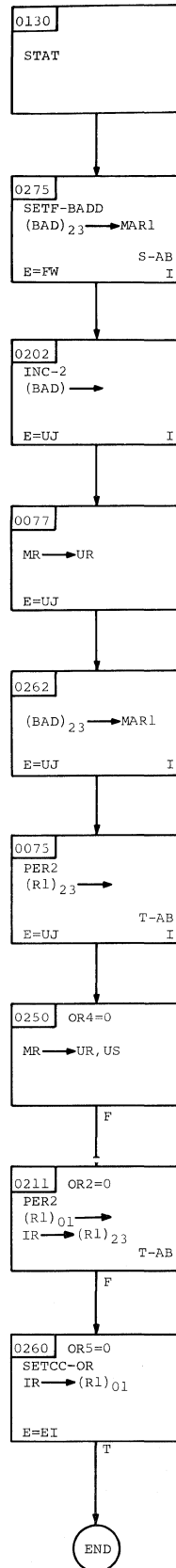


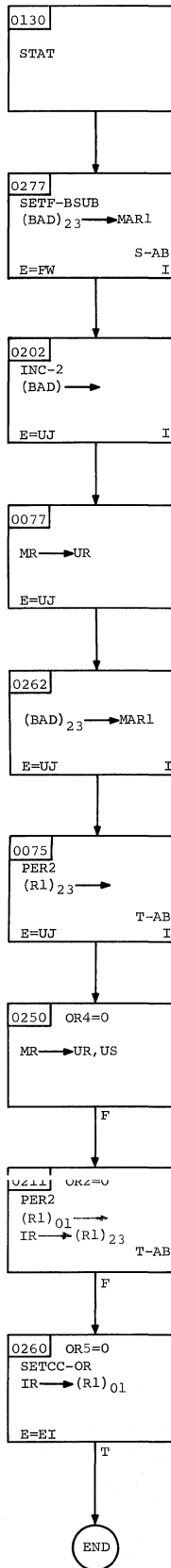


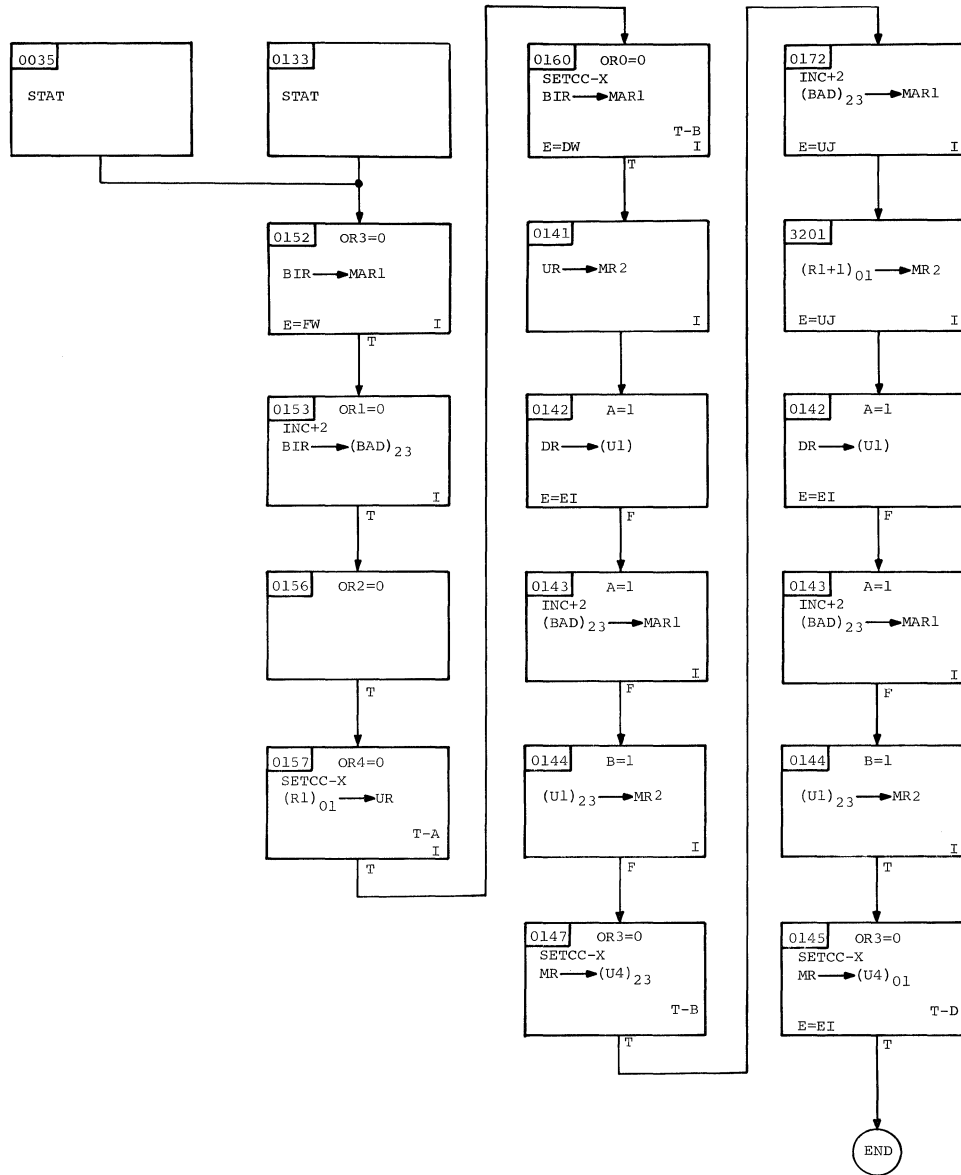


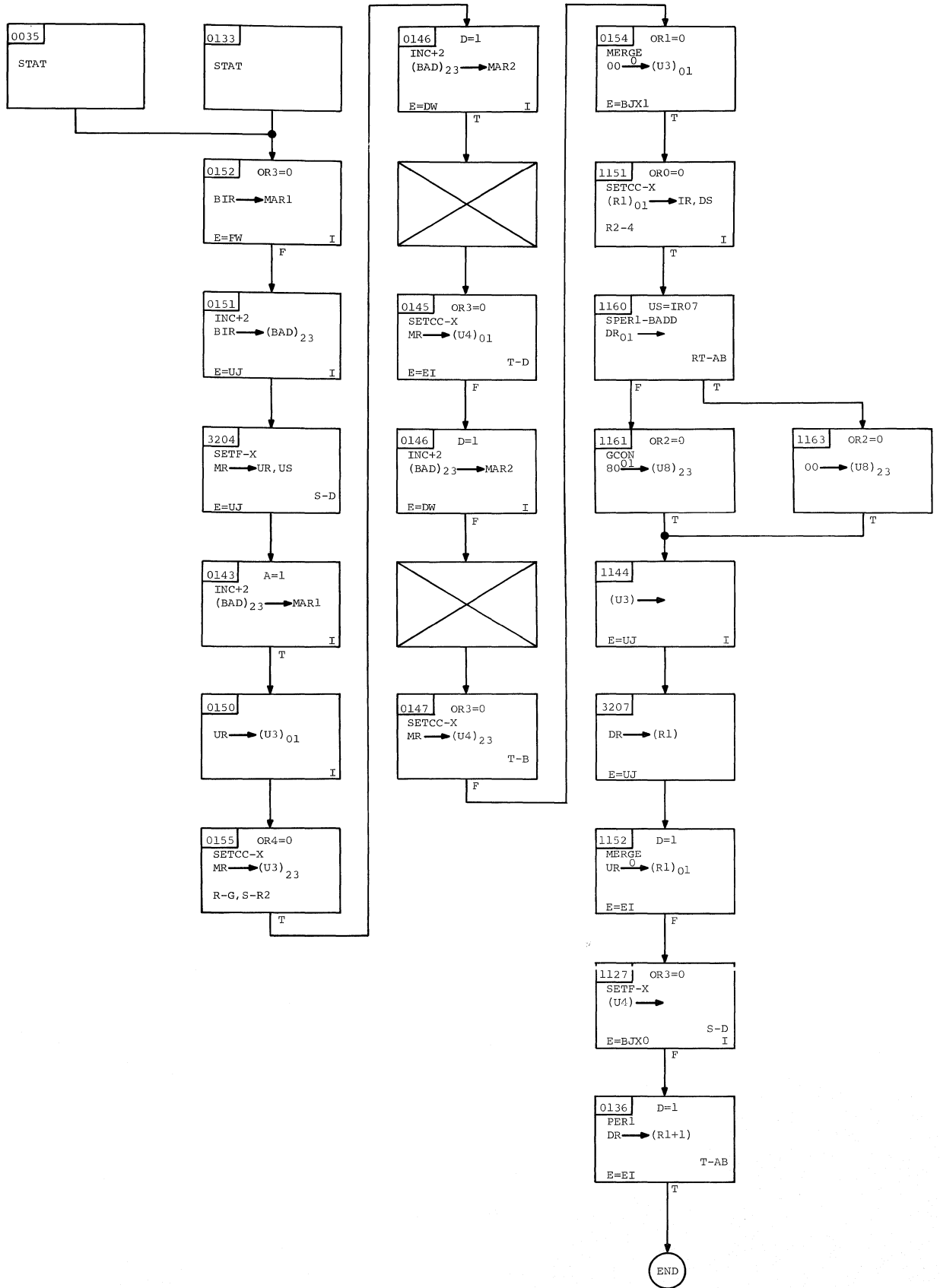


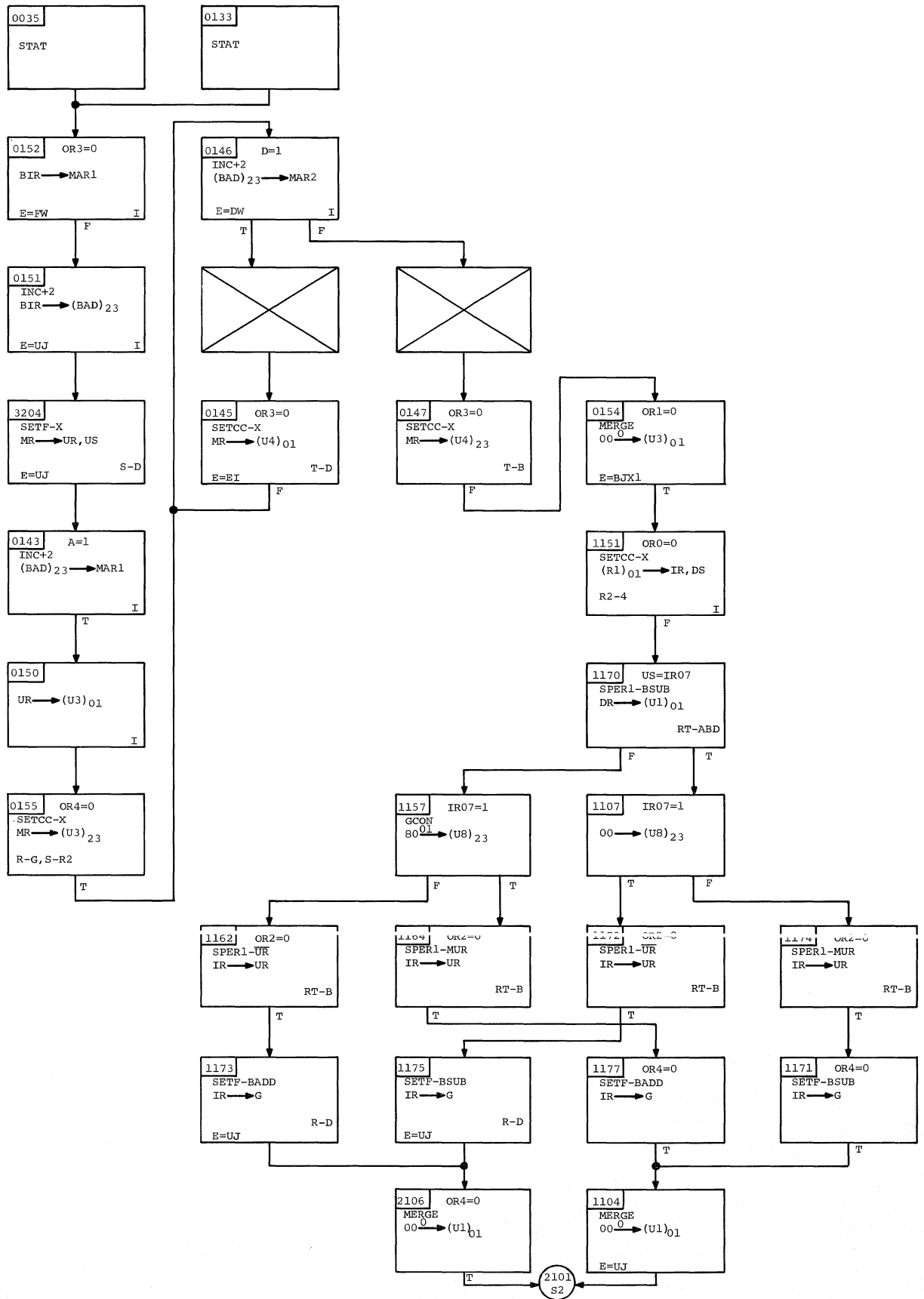


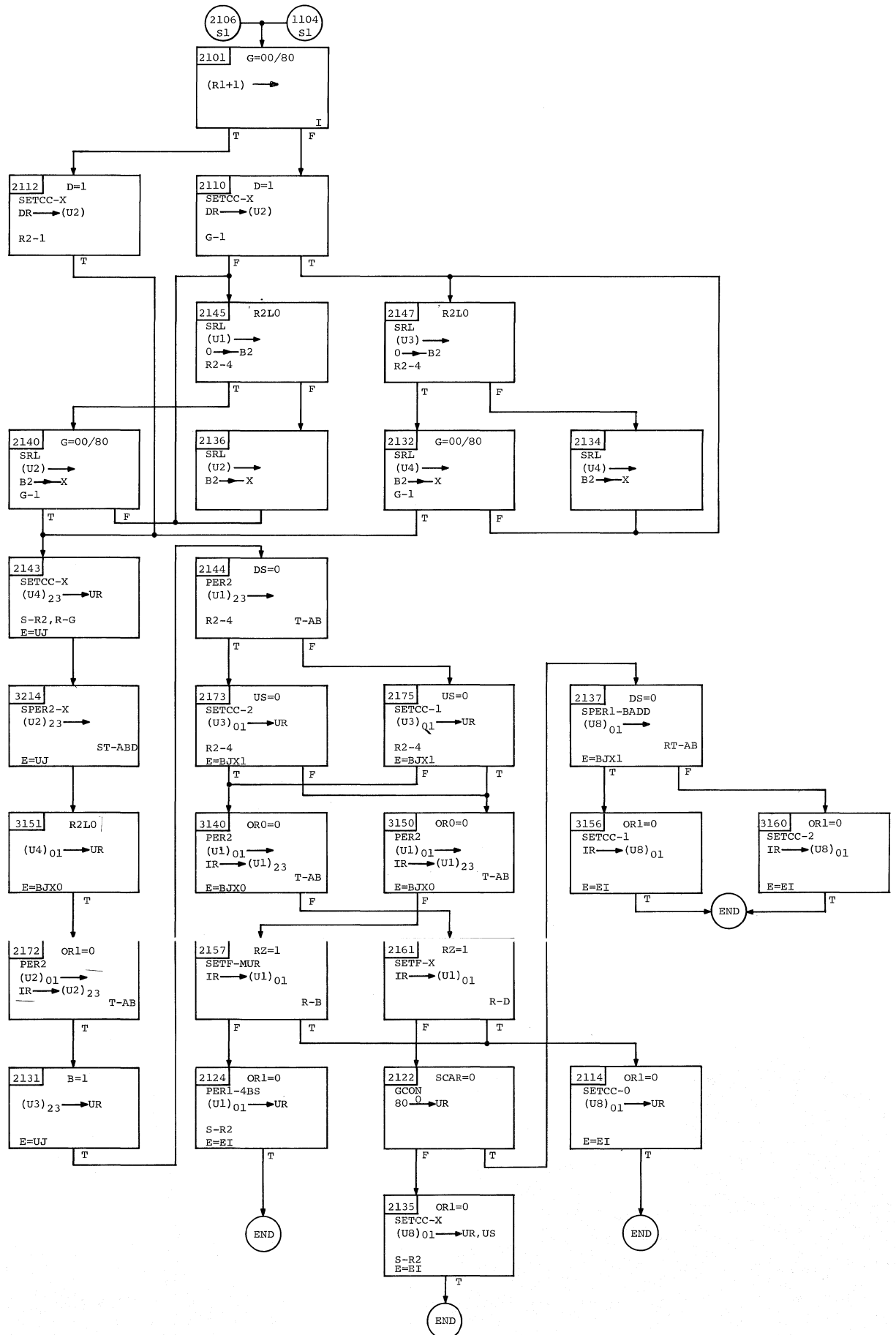




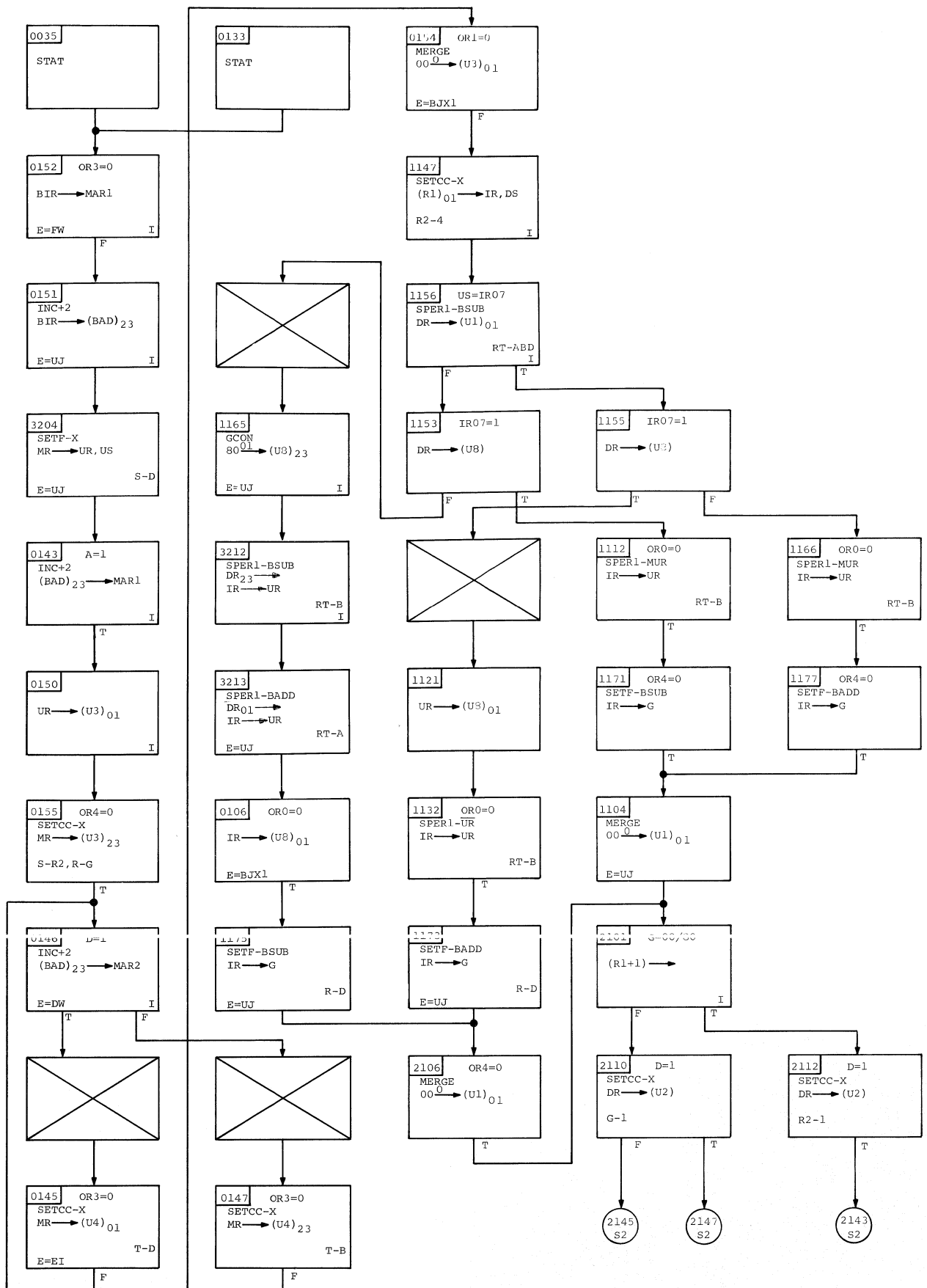




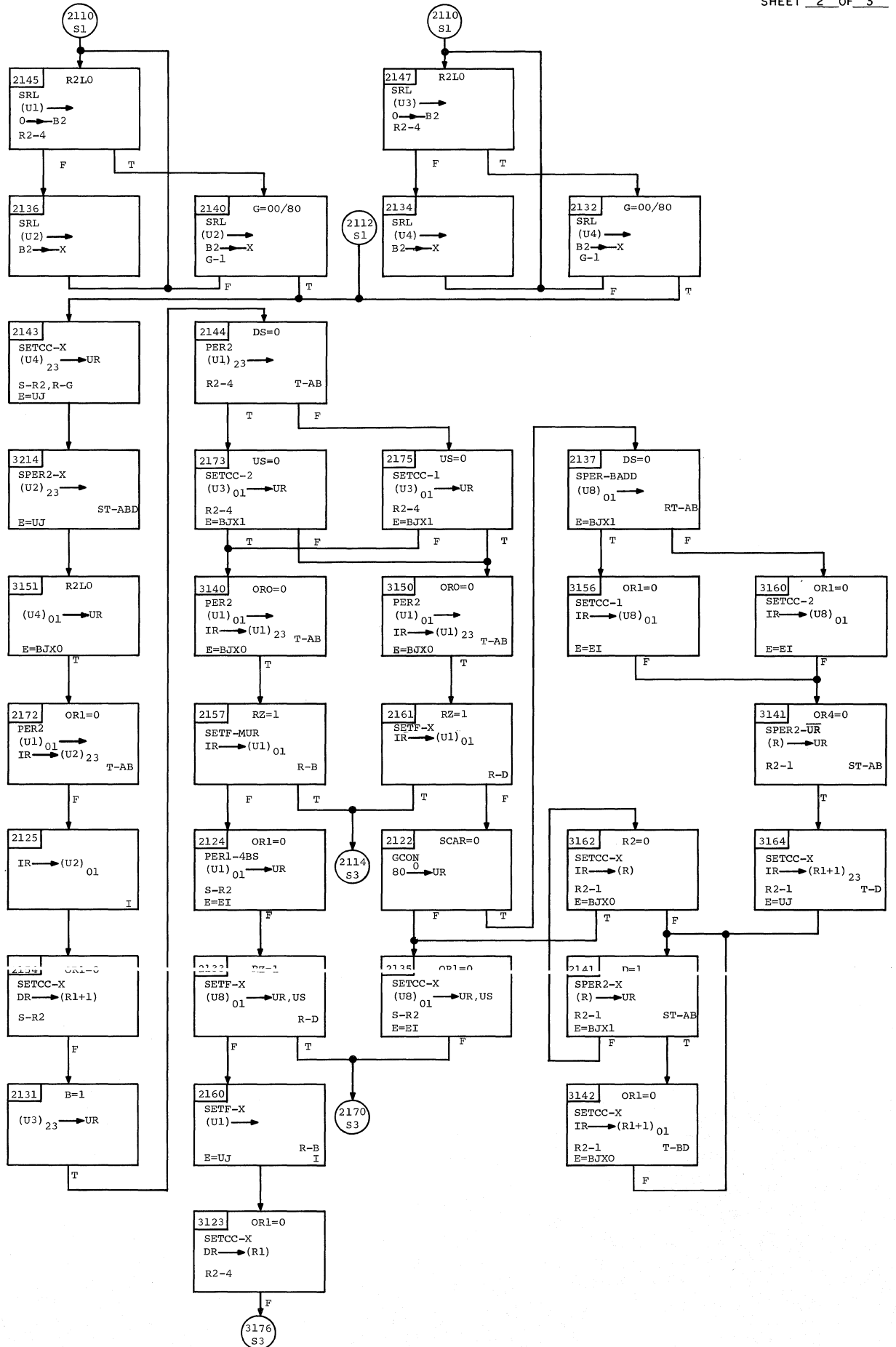


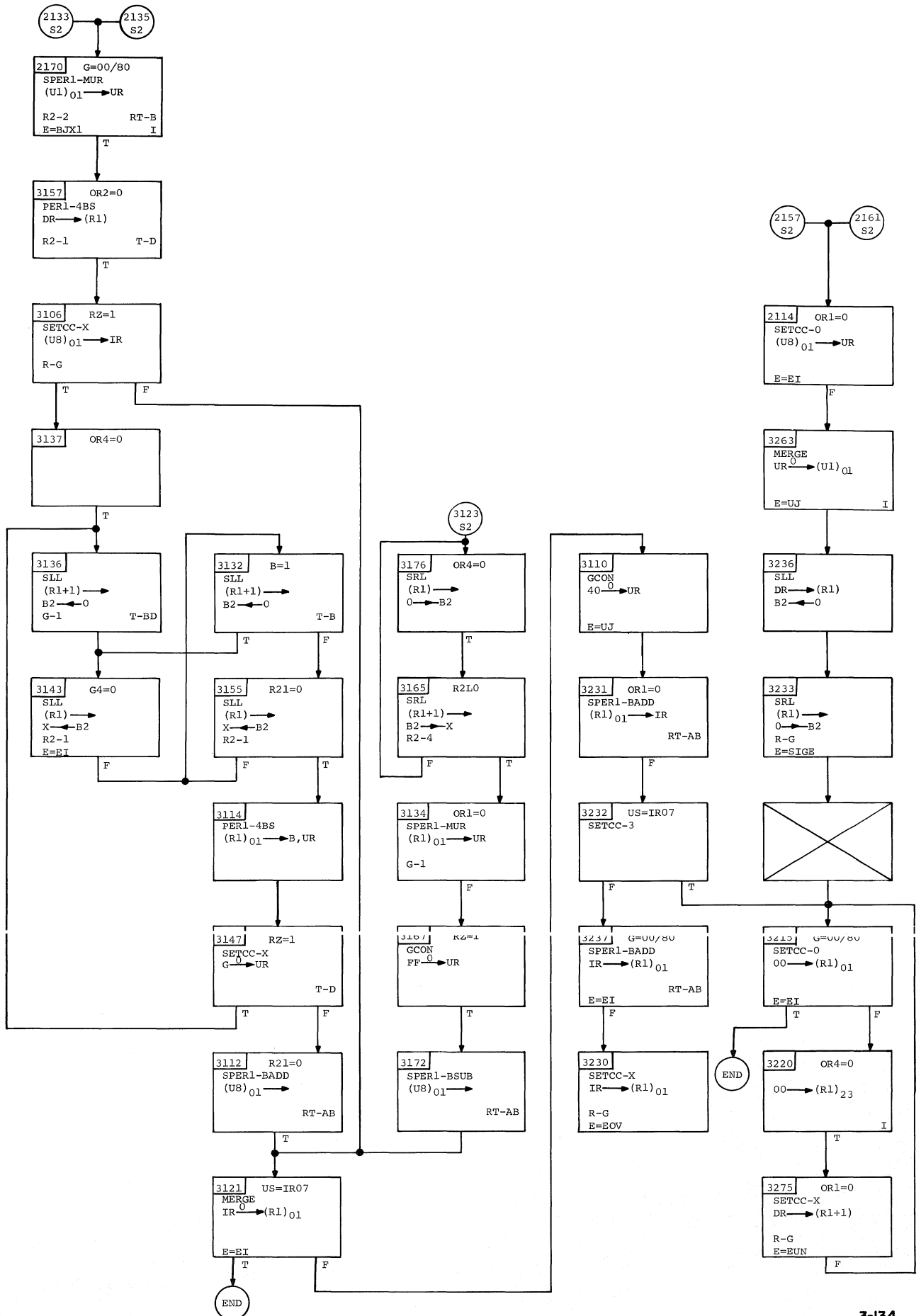


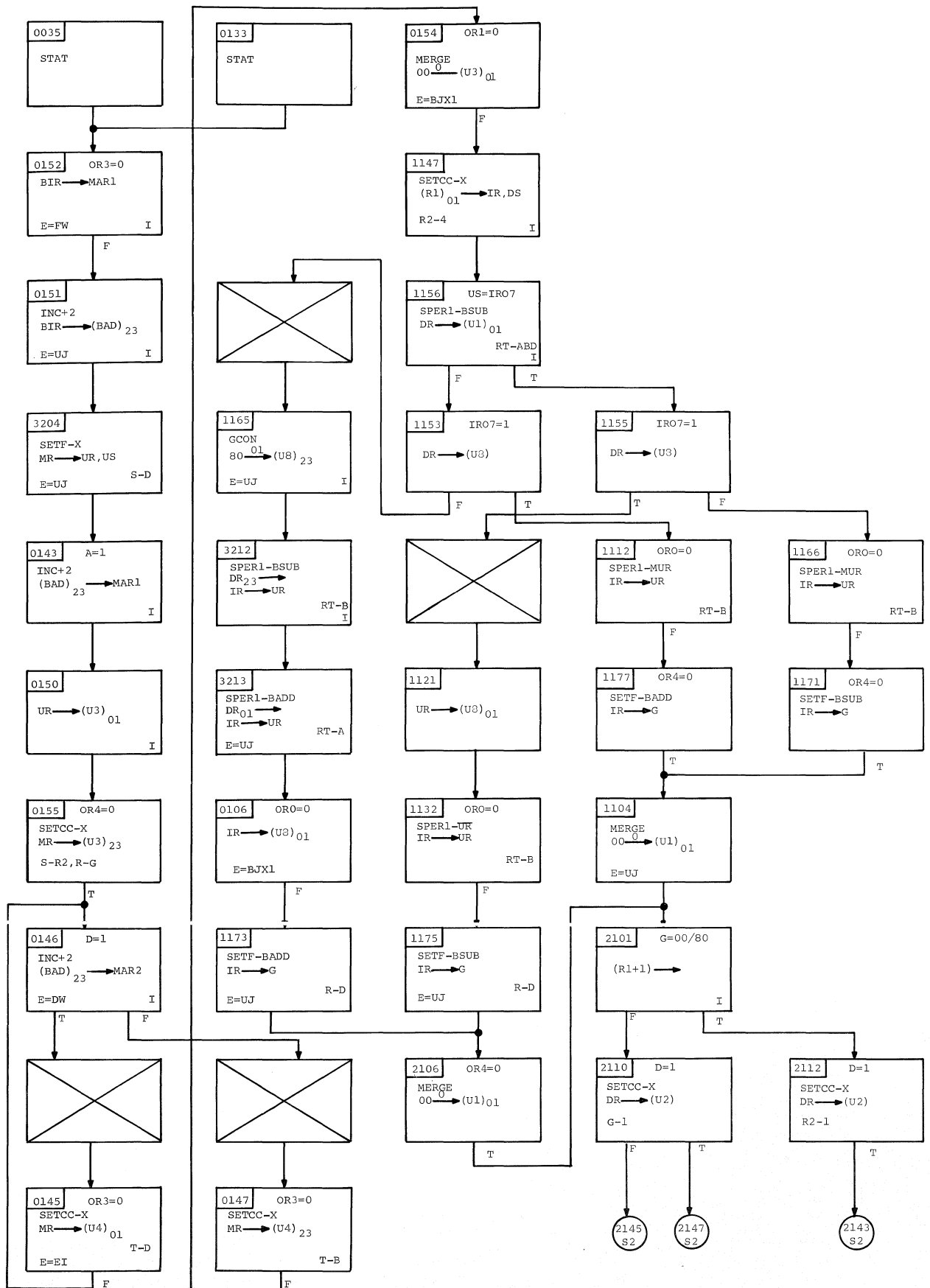




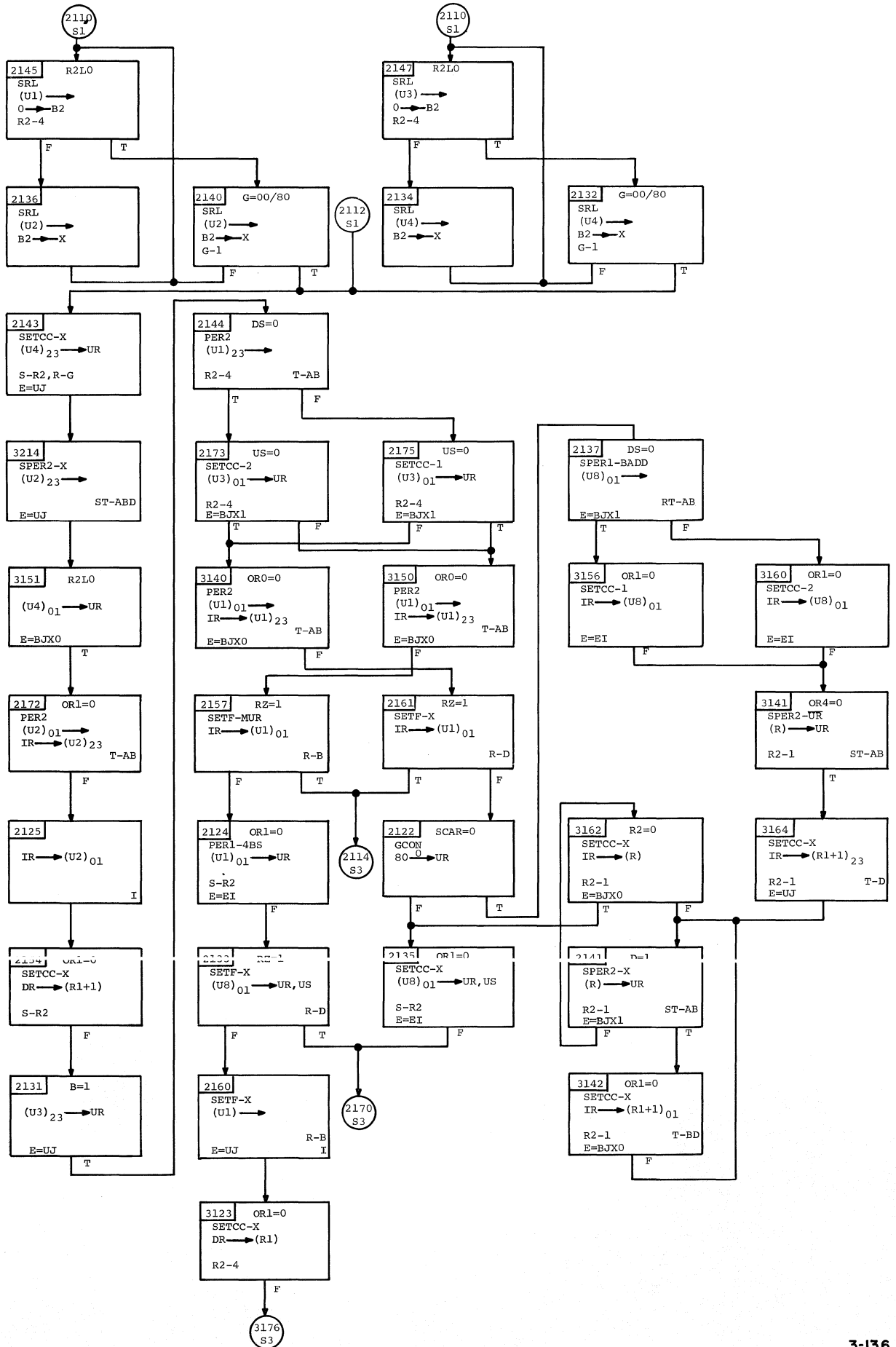
# ADD NORMALIZED LONG AND RX 6A

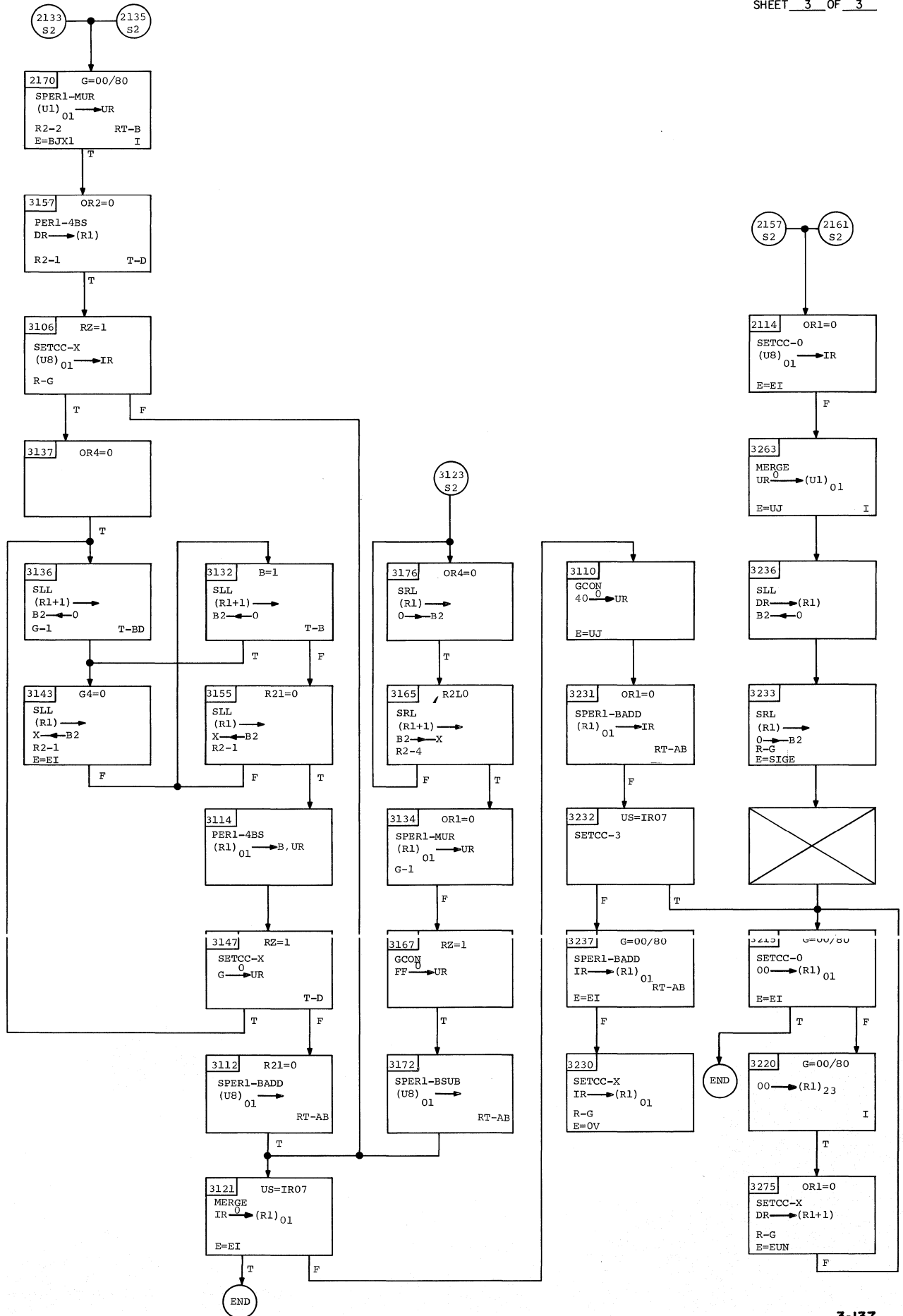


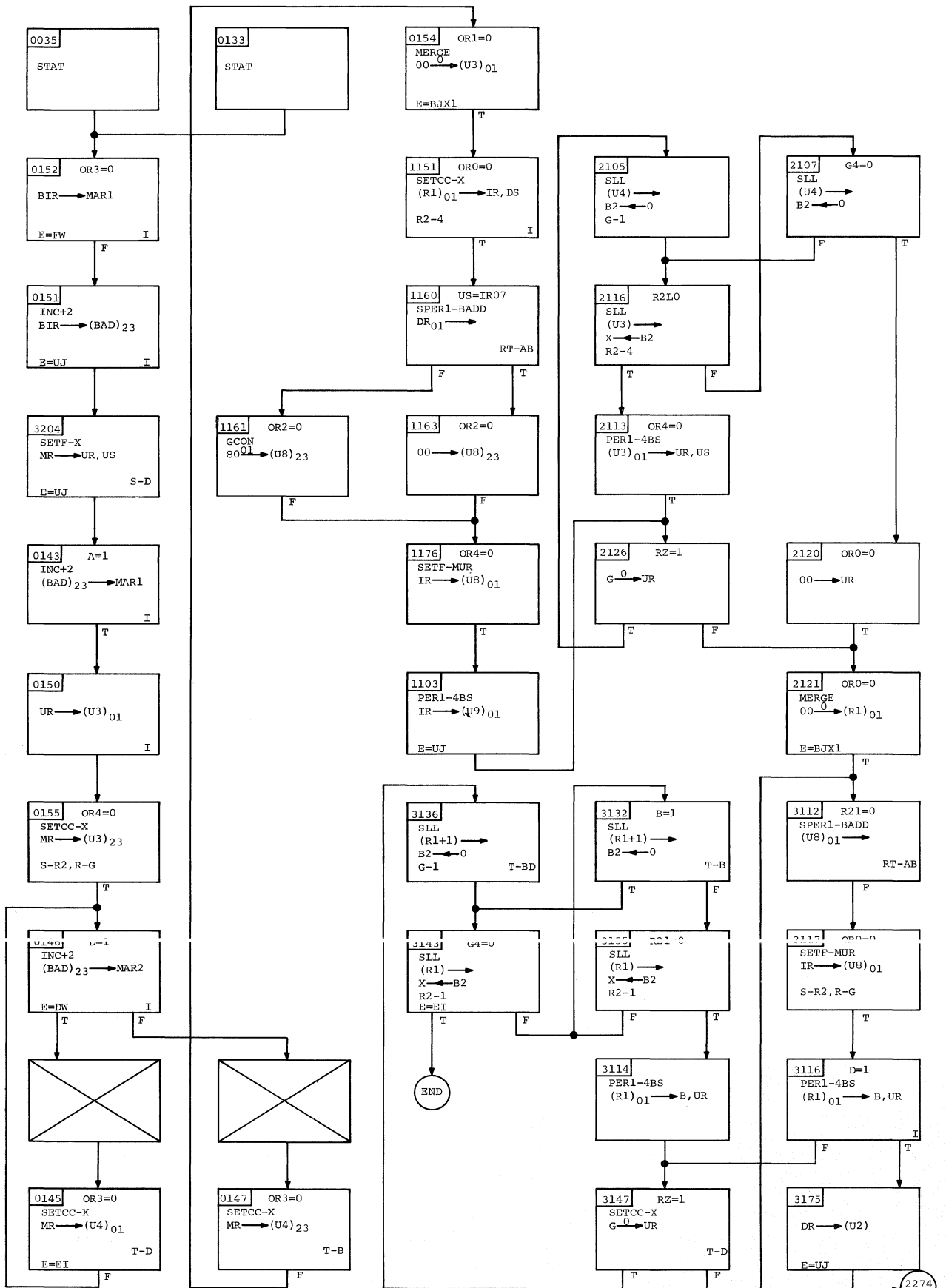


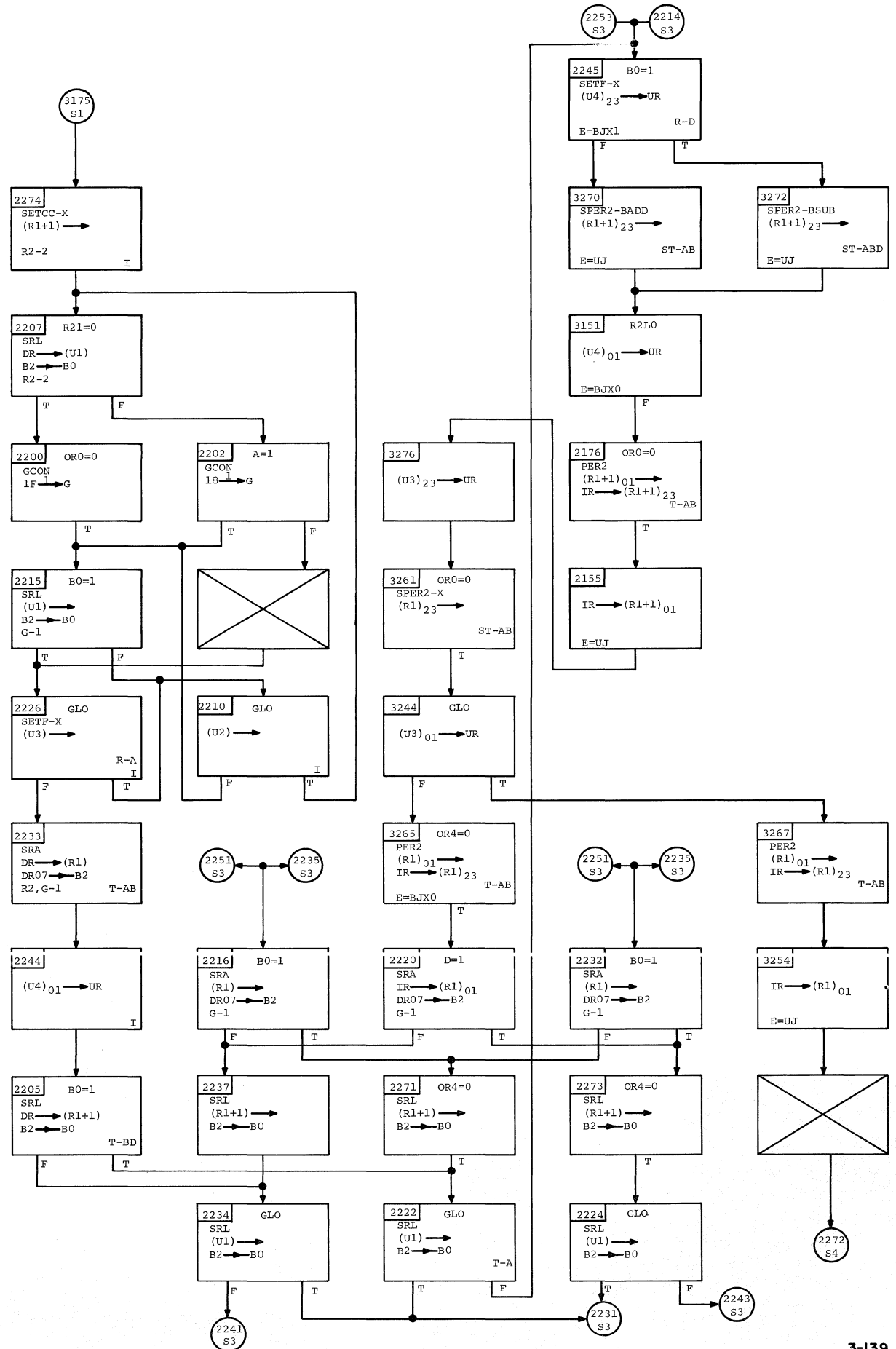


SUBTRACT NORMALIZED LONG SND RX 6B

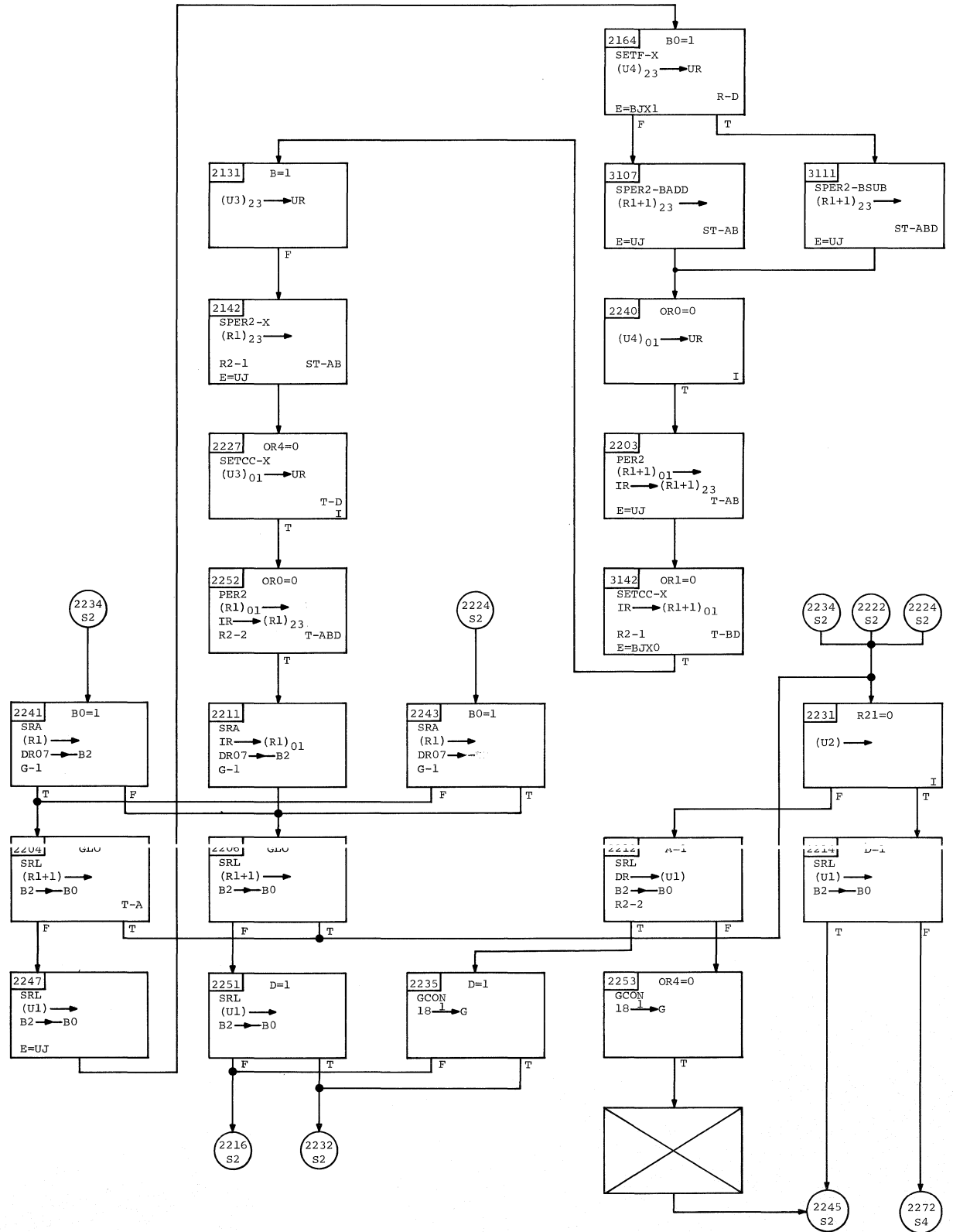


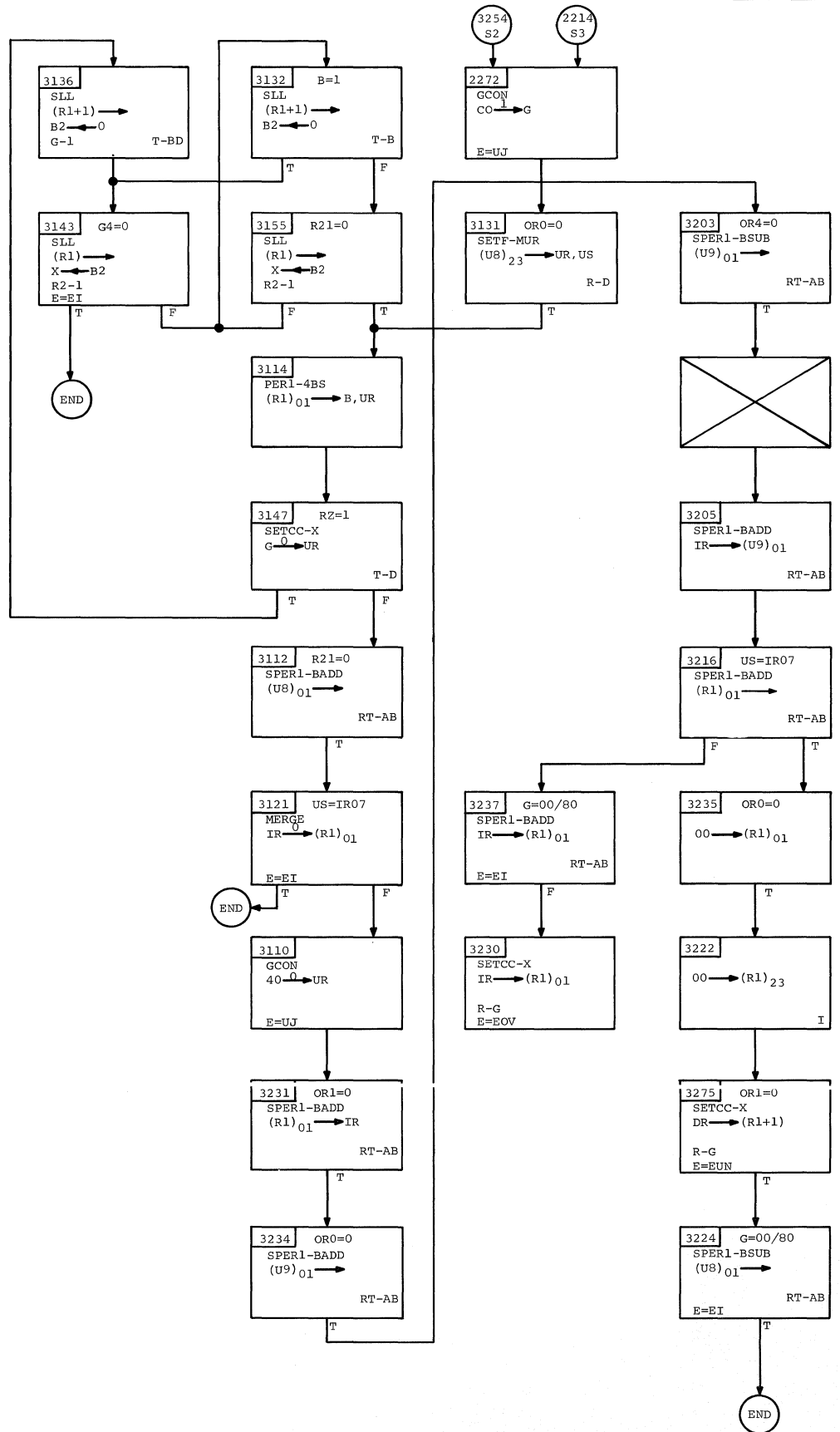


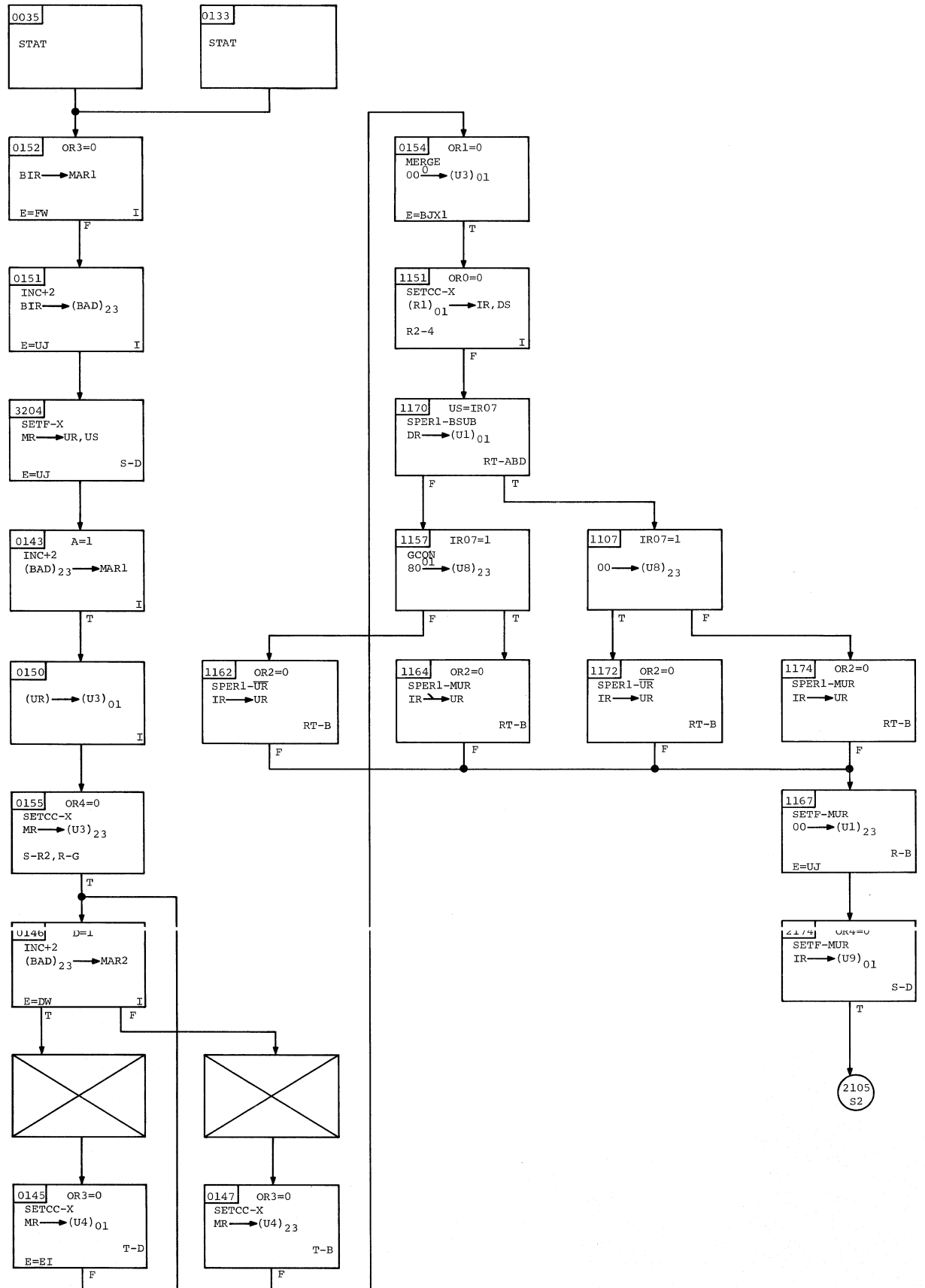


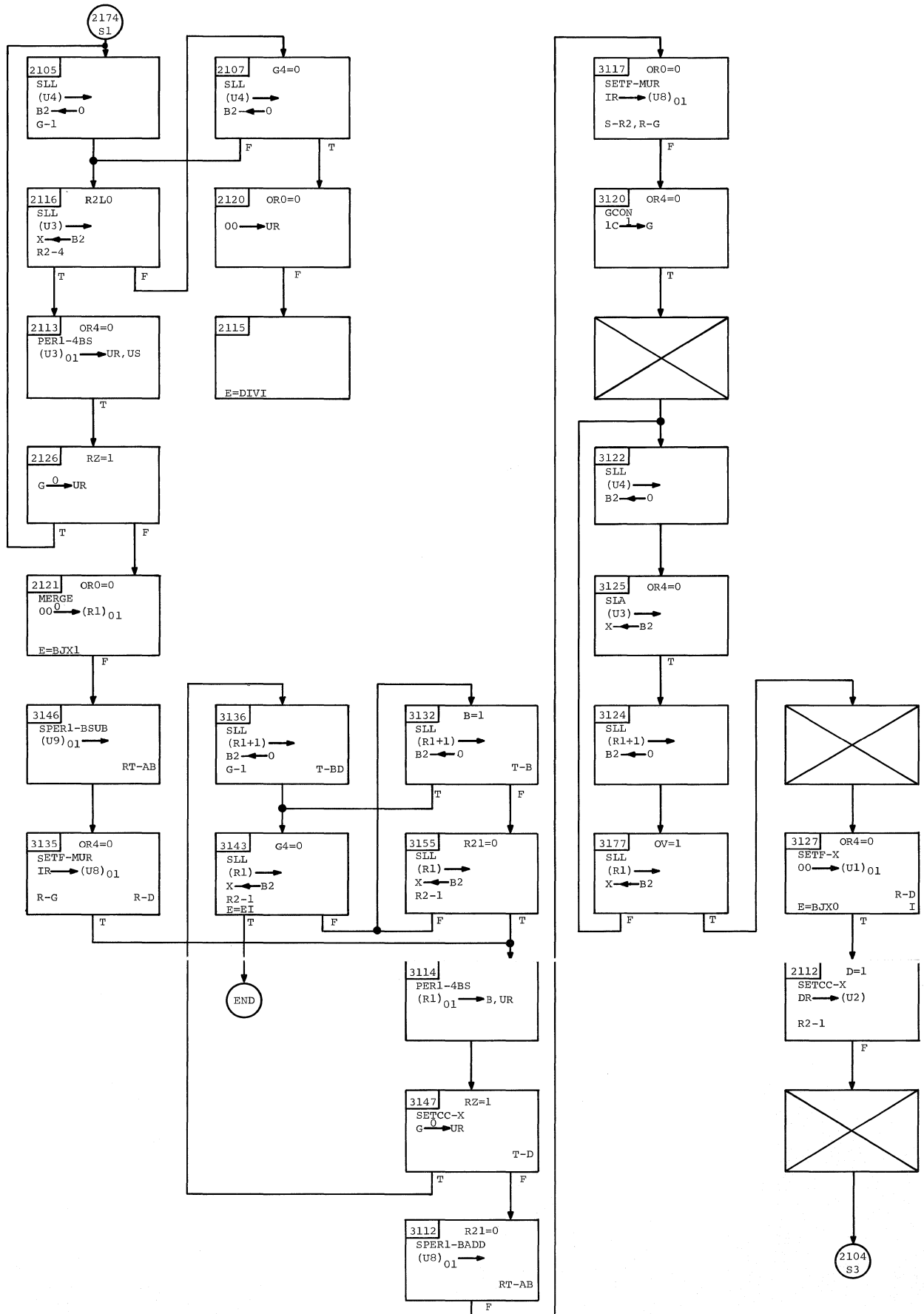


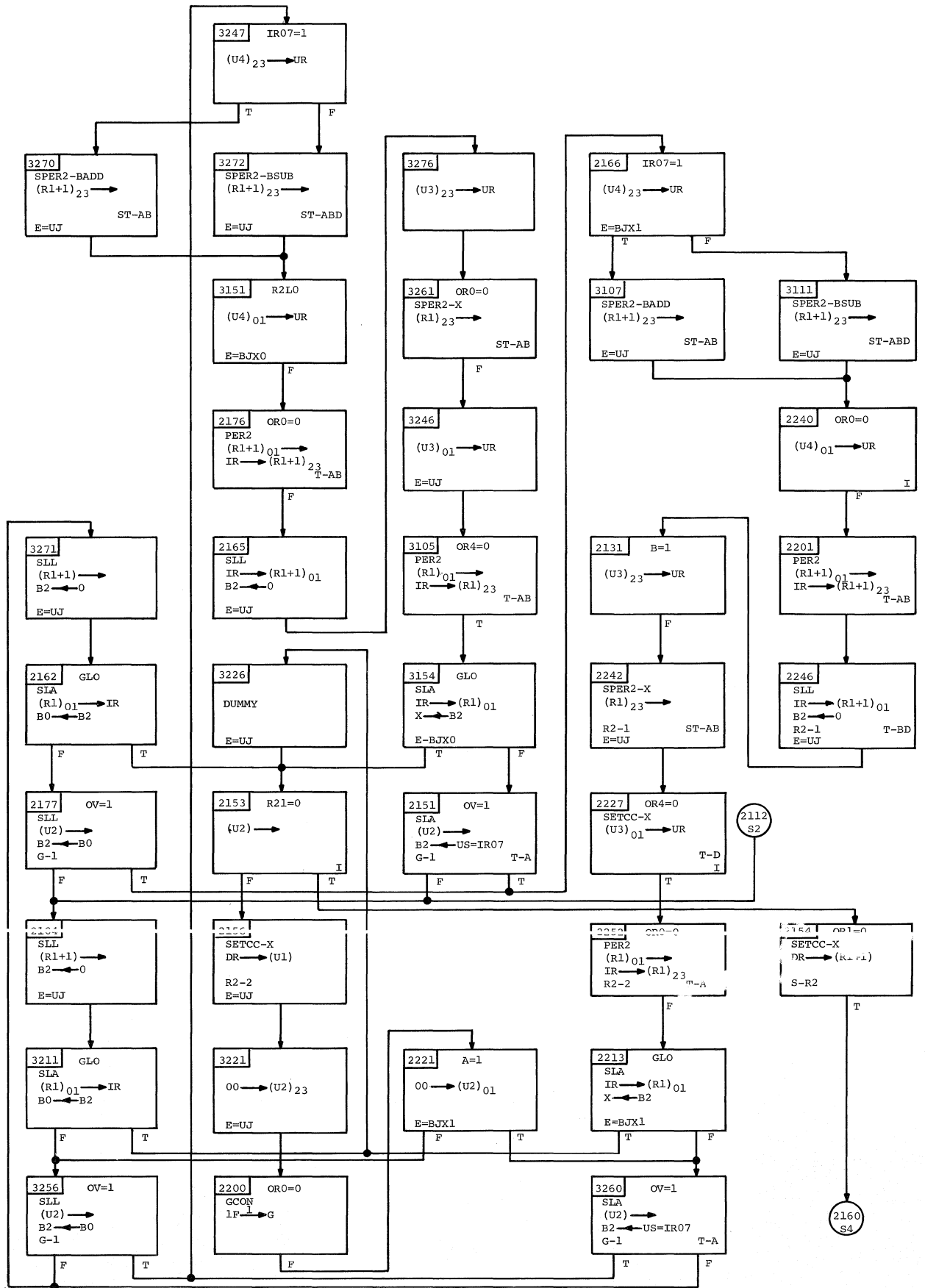


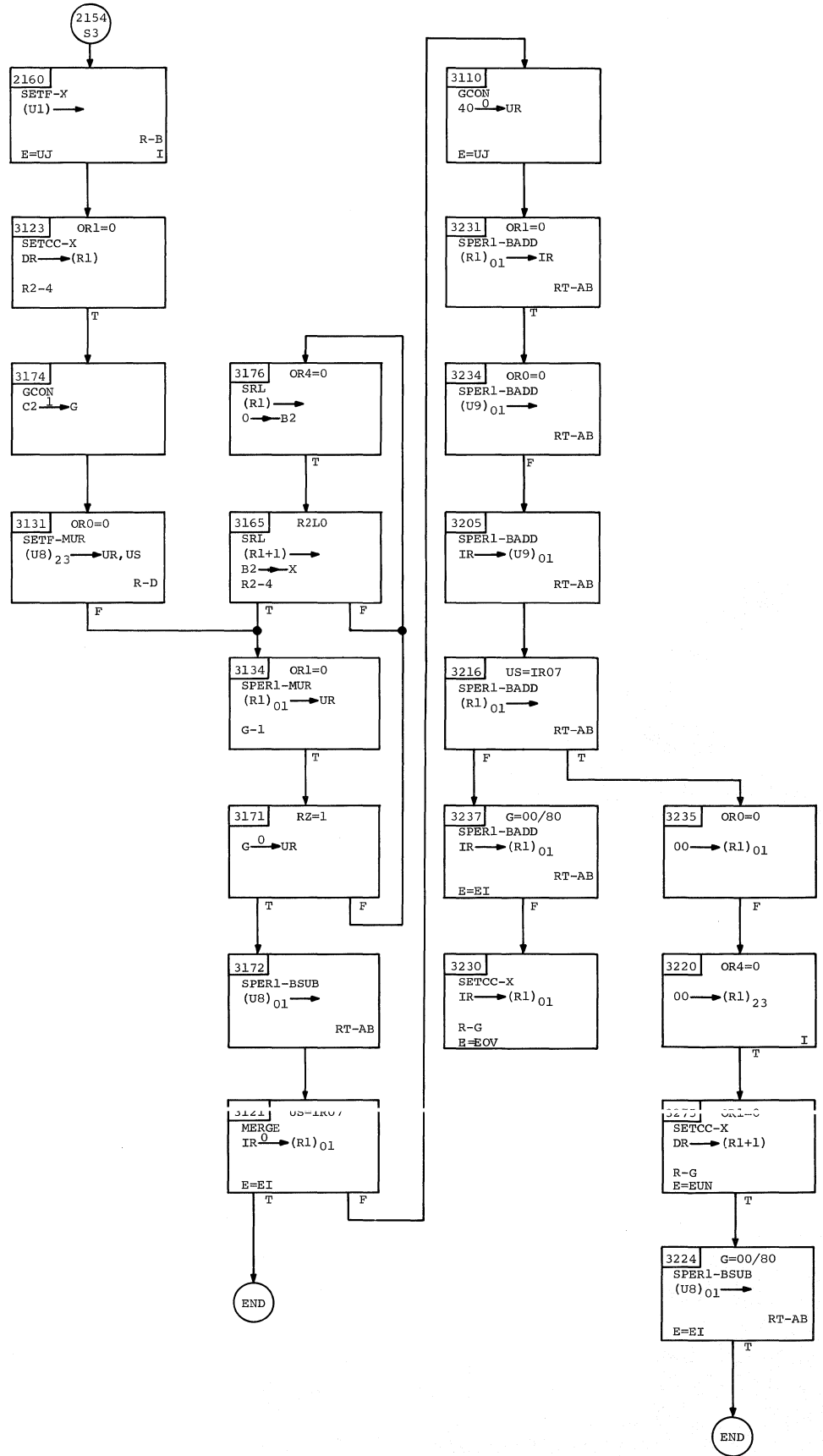


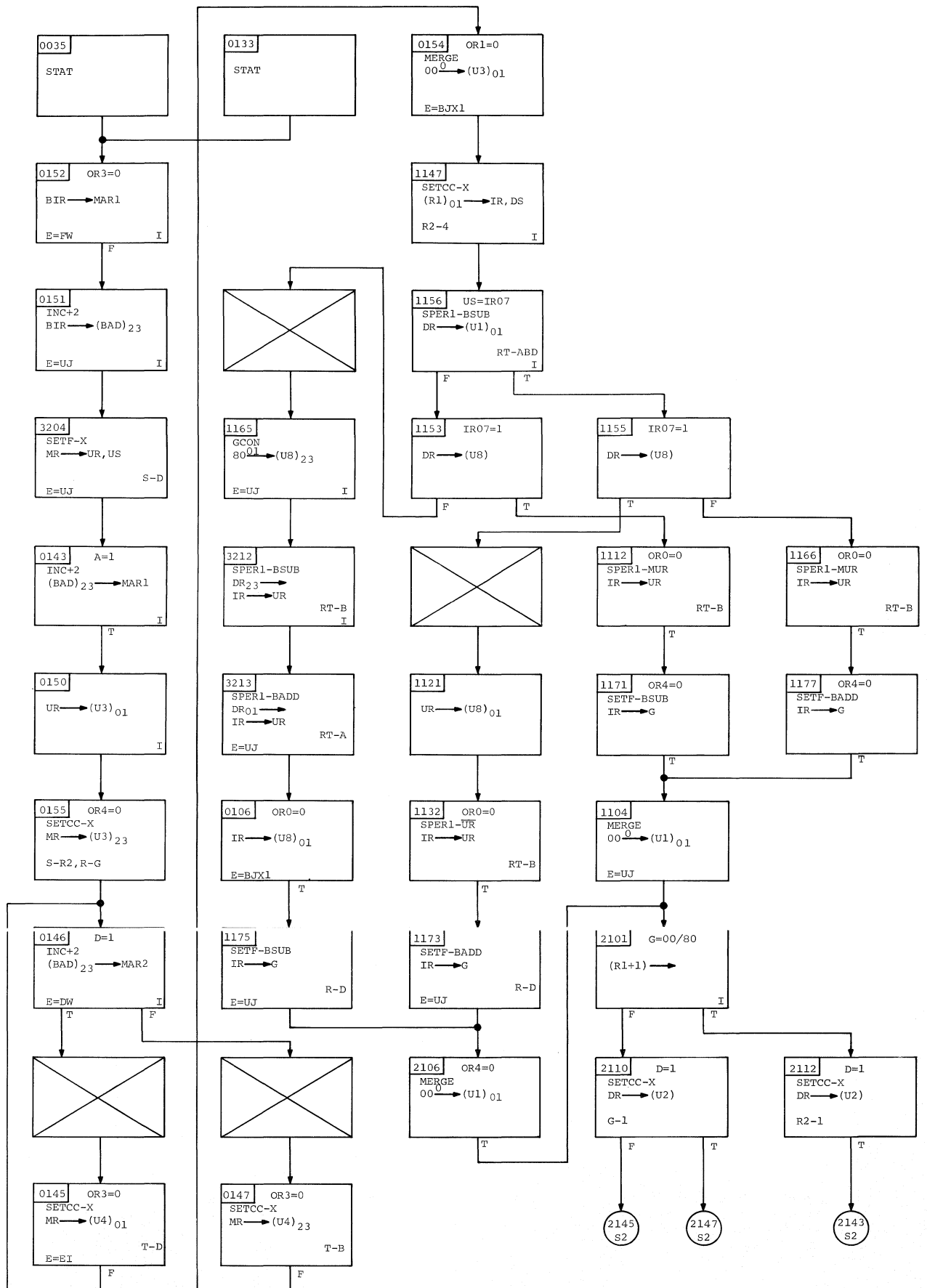




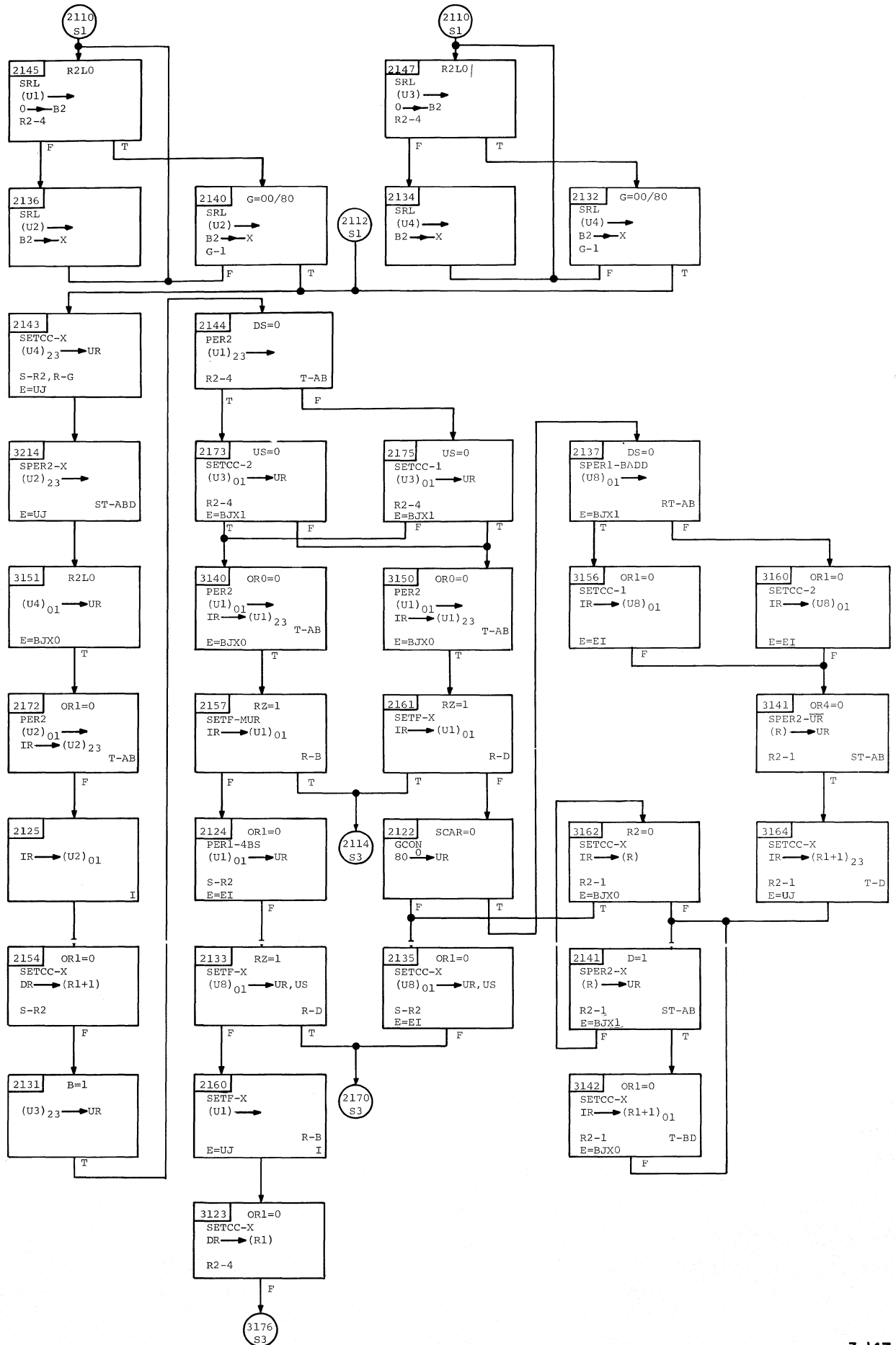




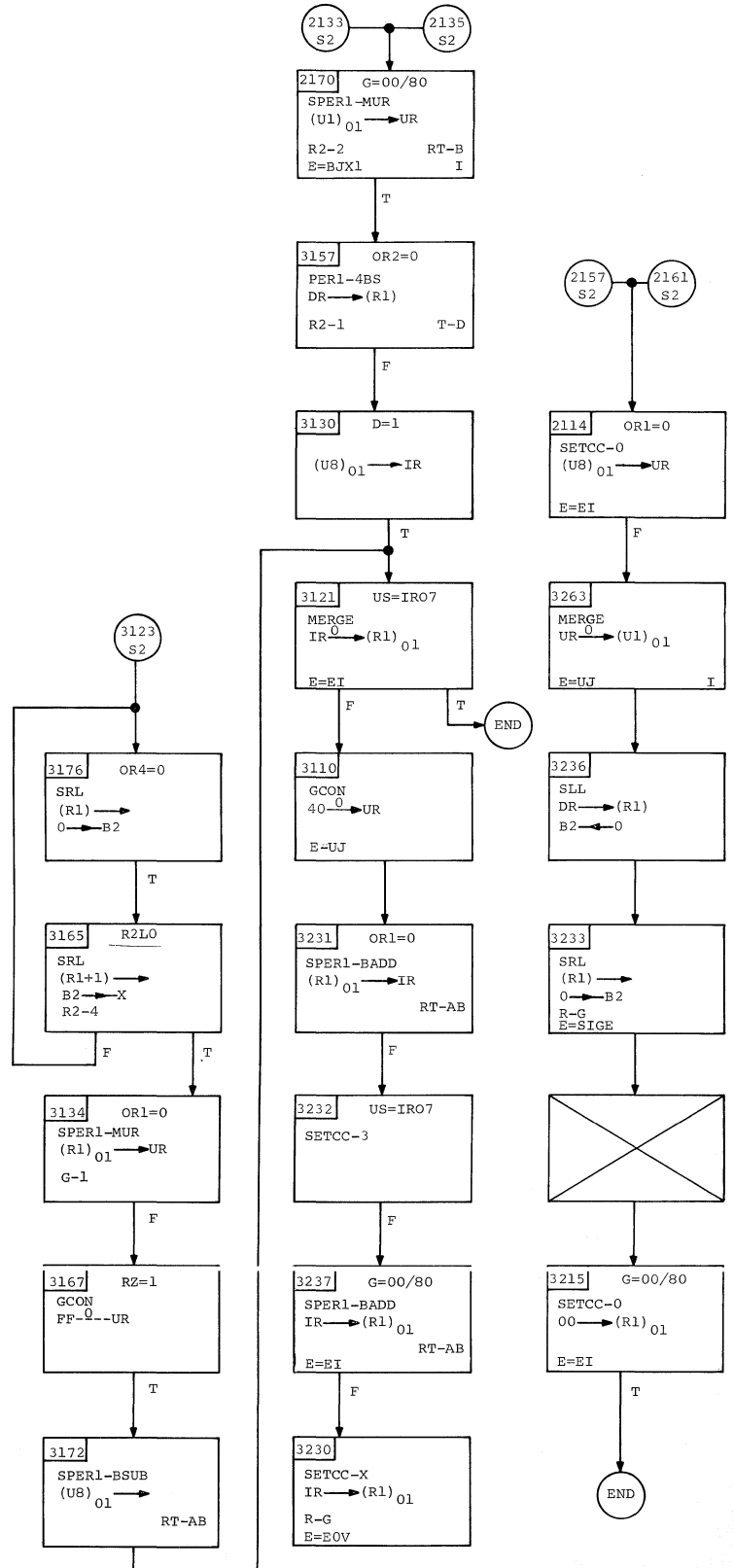


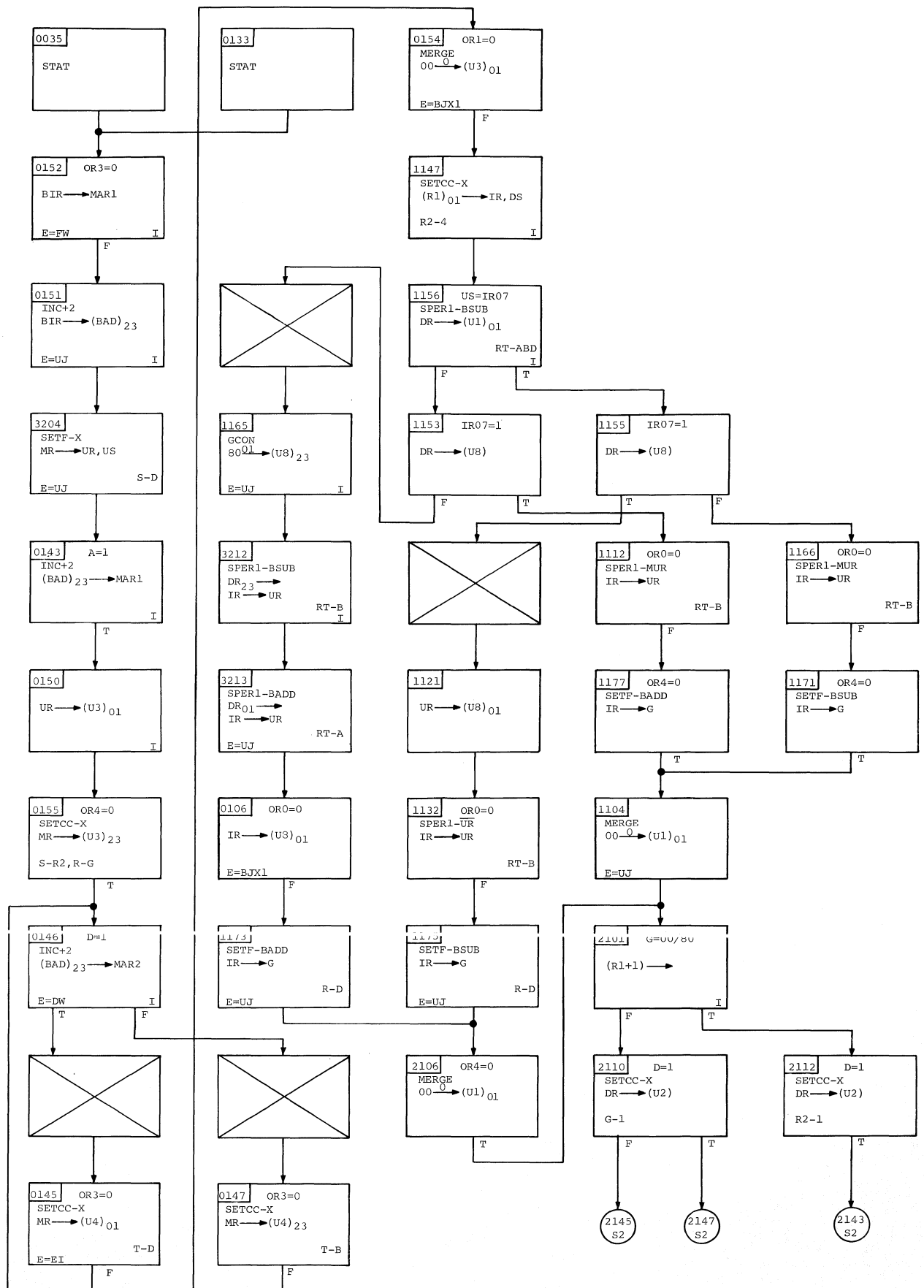


# ADD UNNORMALIZED LONG AW RX 6E

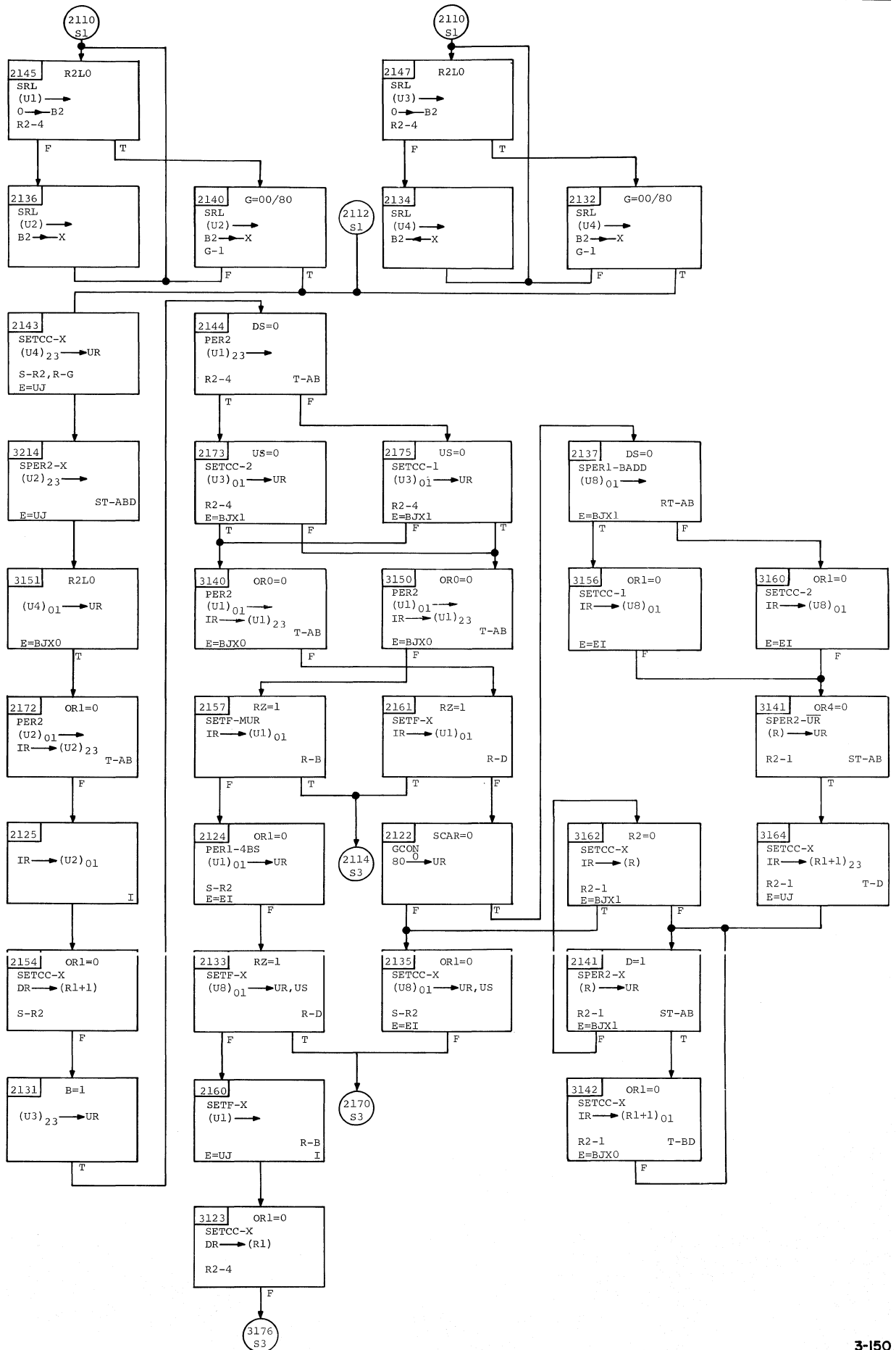


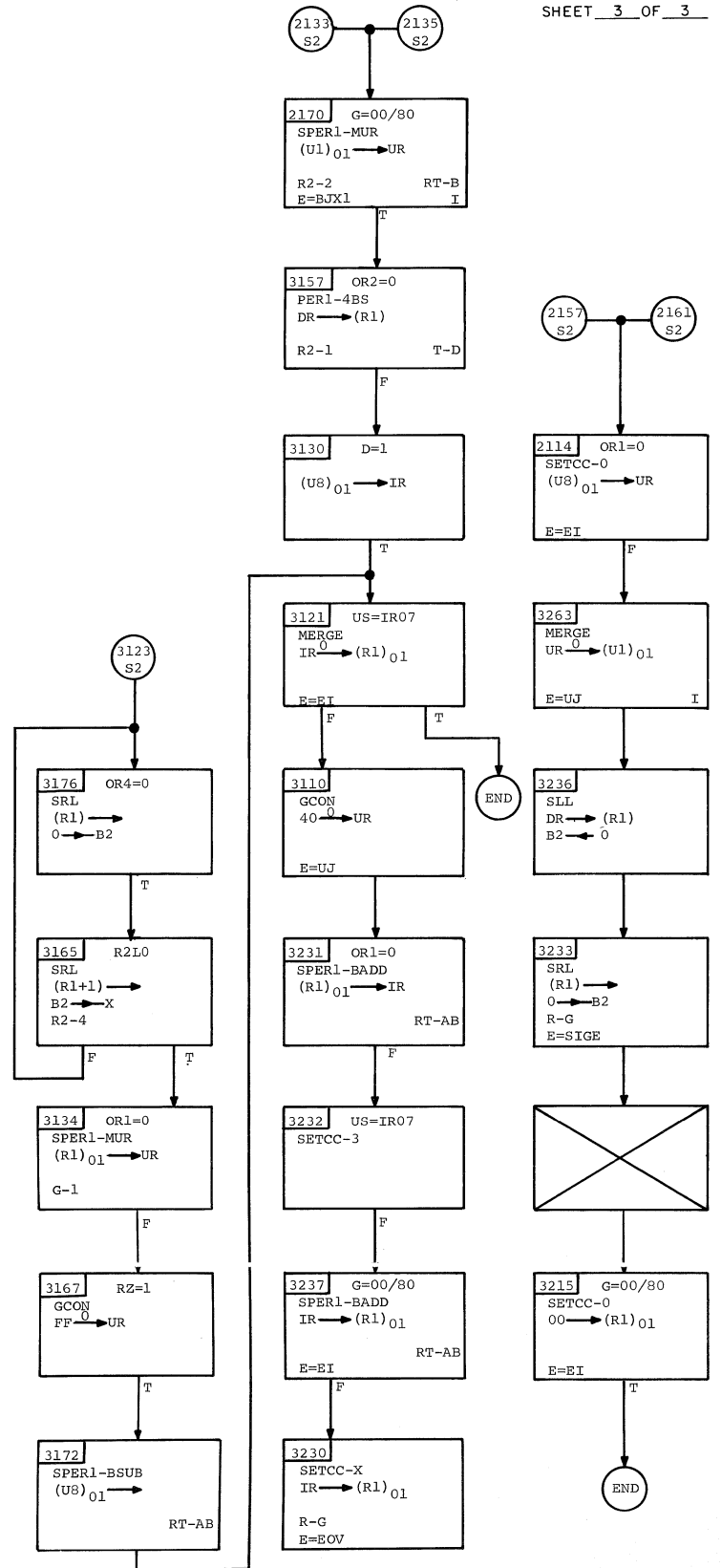


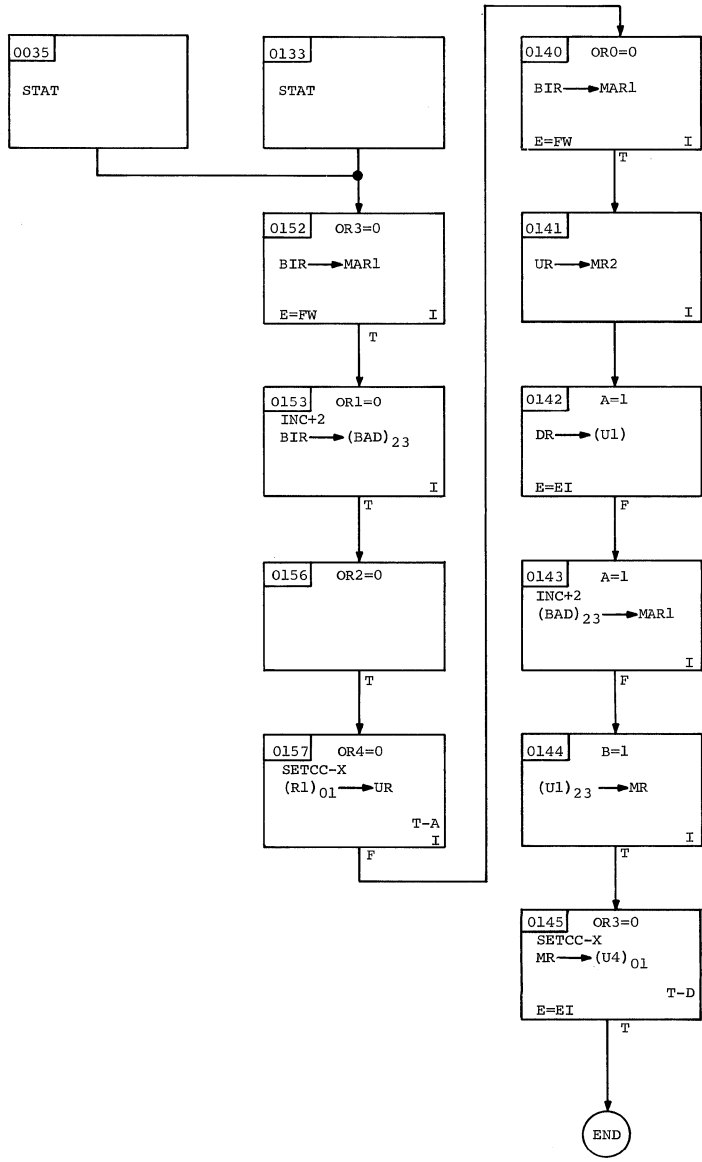


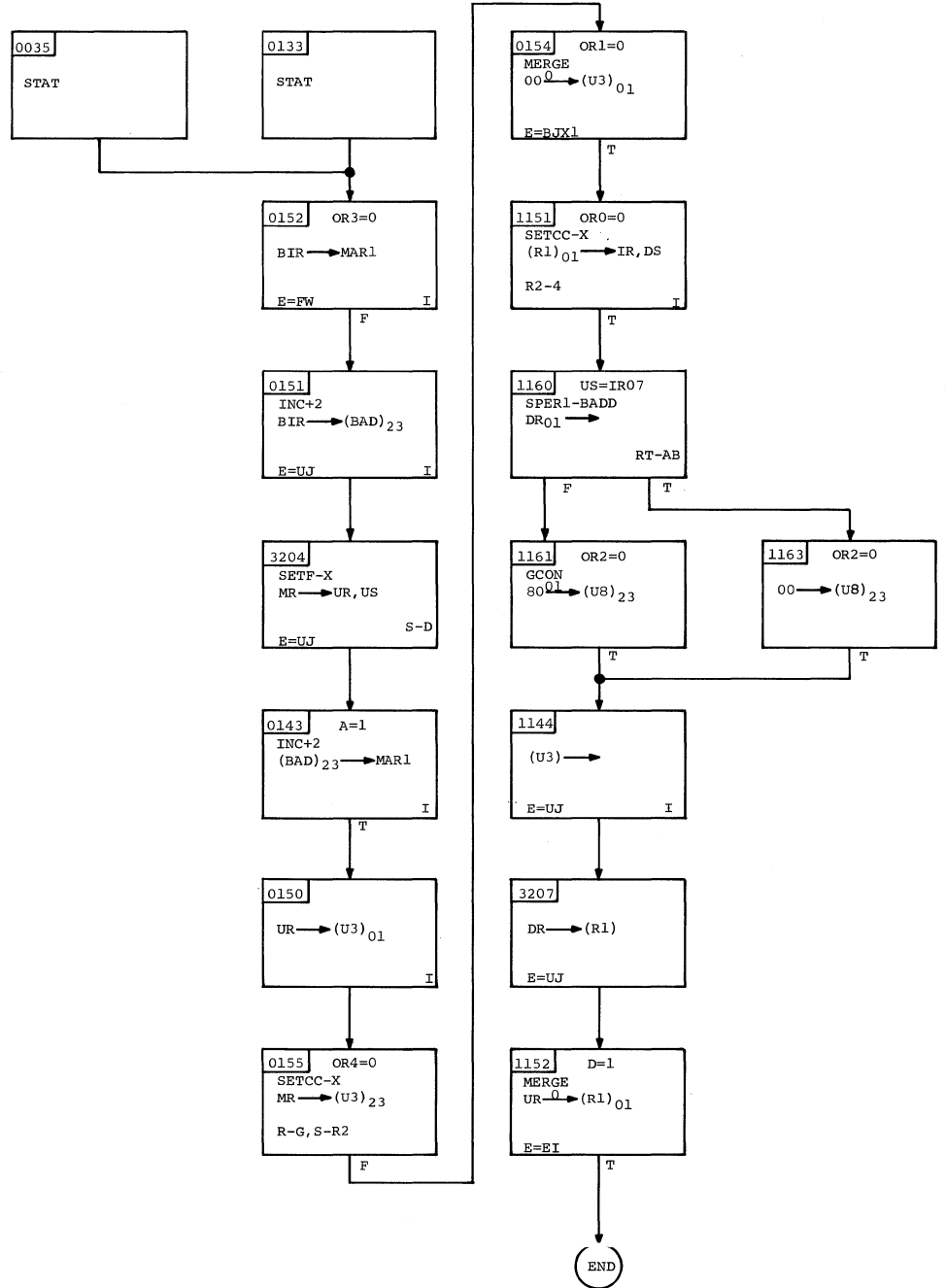


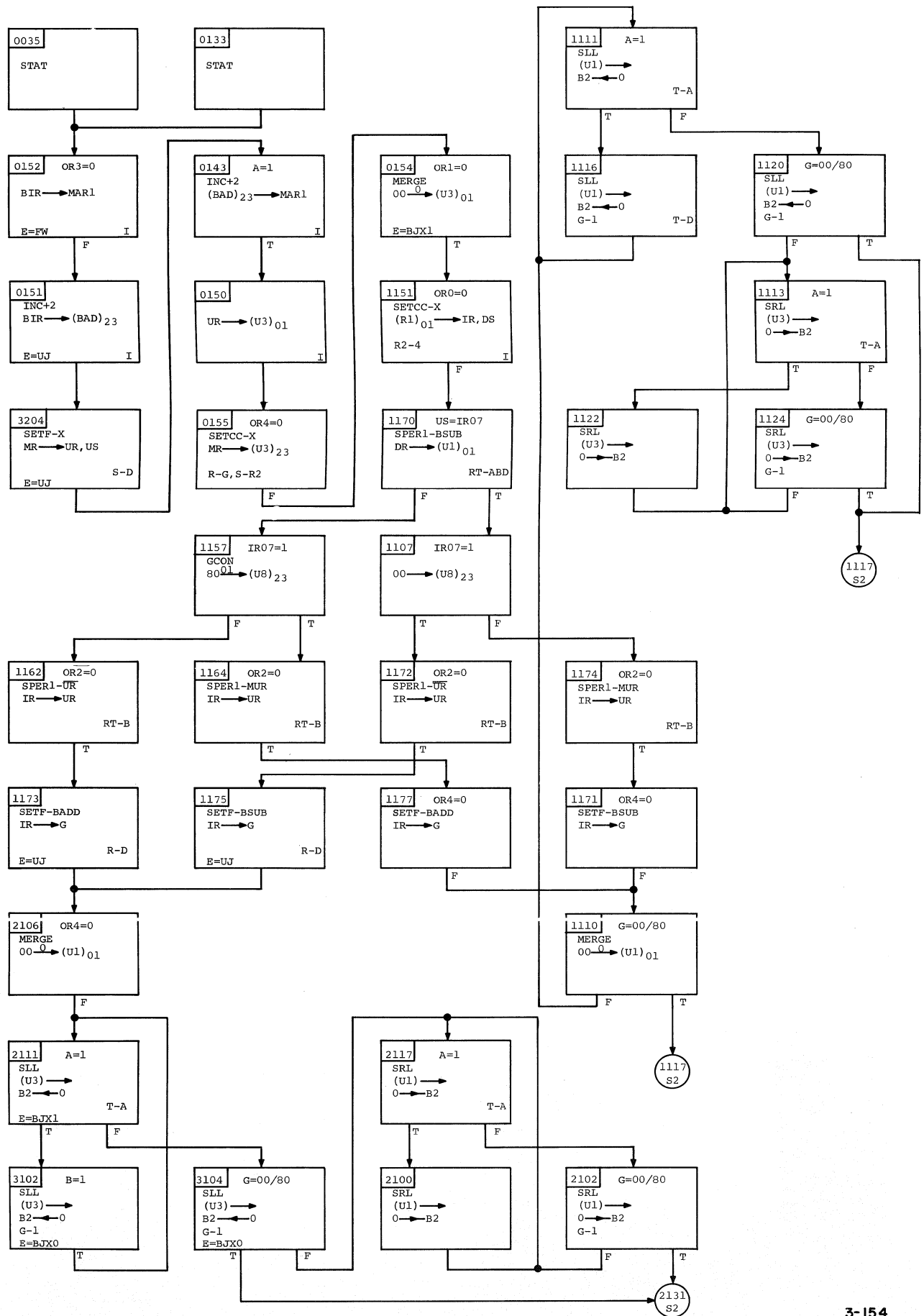
SUBTRACT UNNORMALIZED LONG SW RX 6F

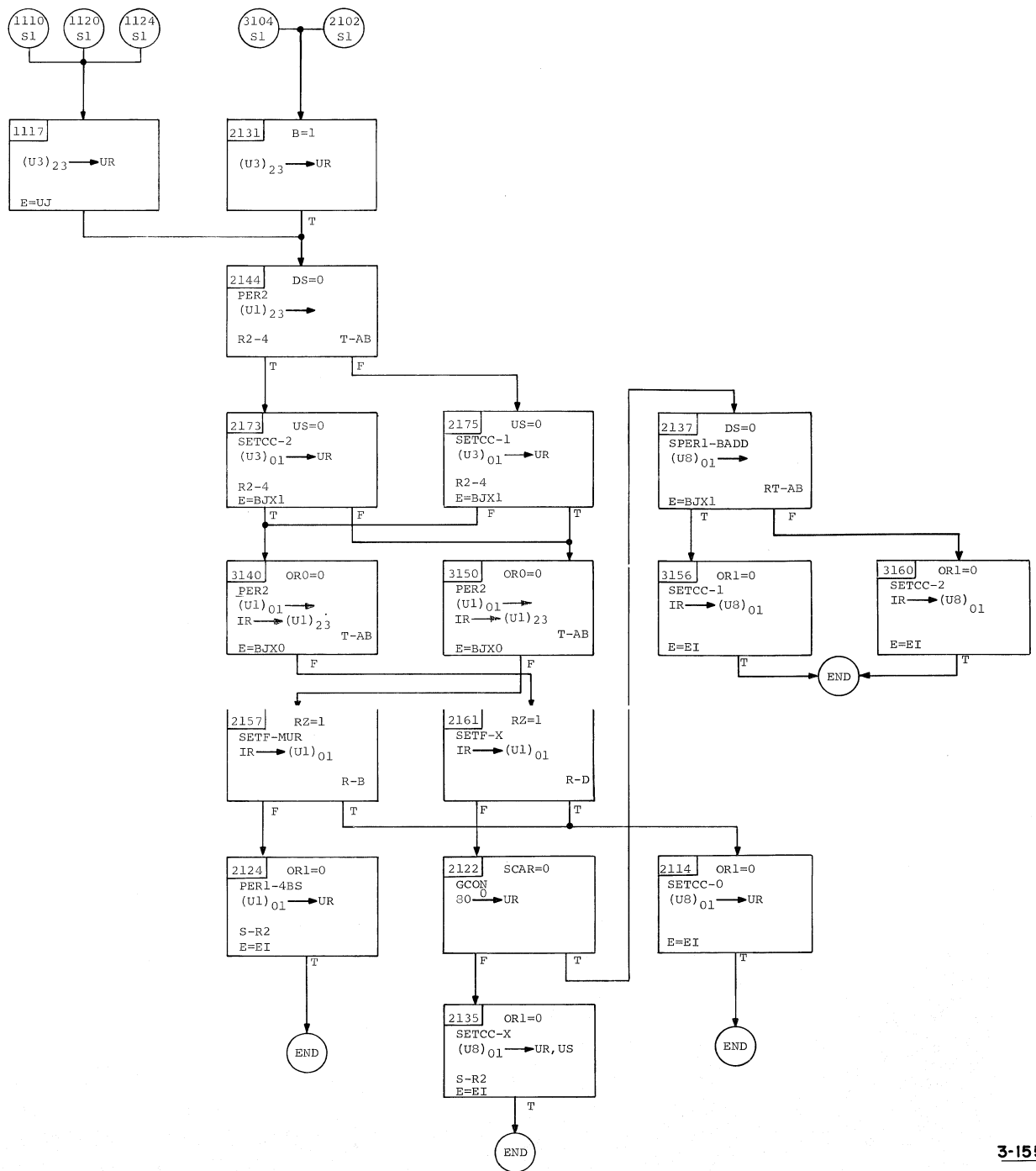




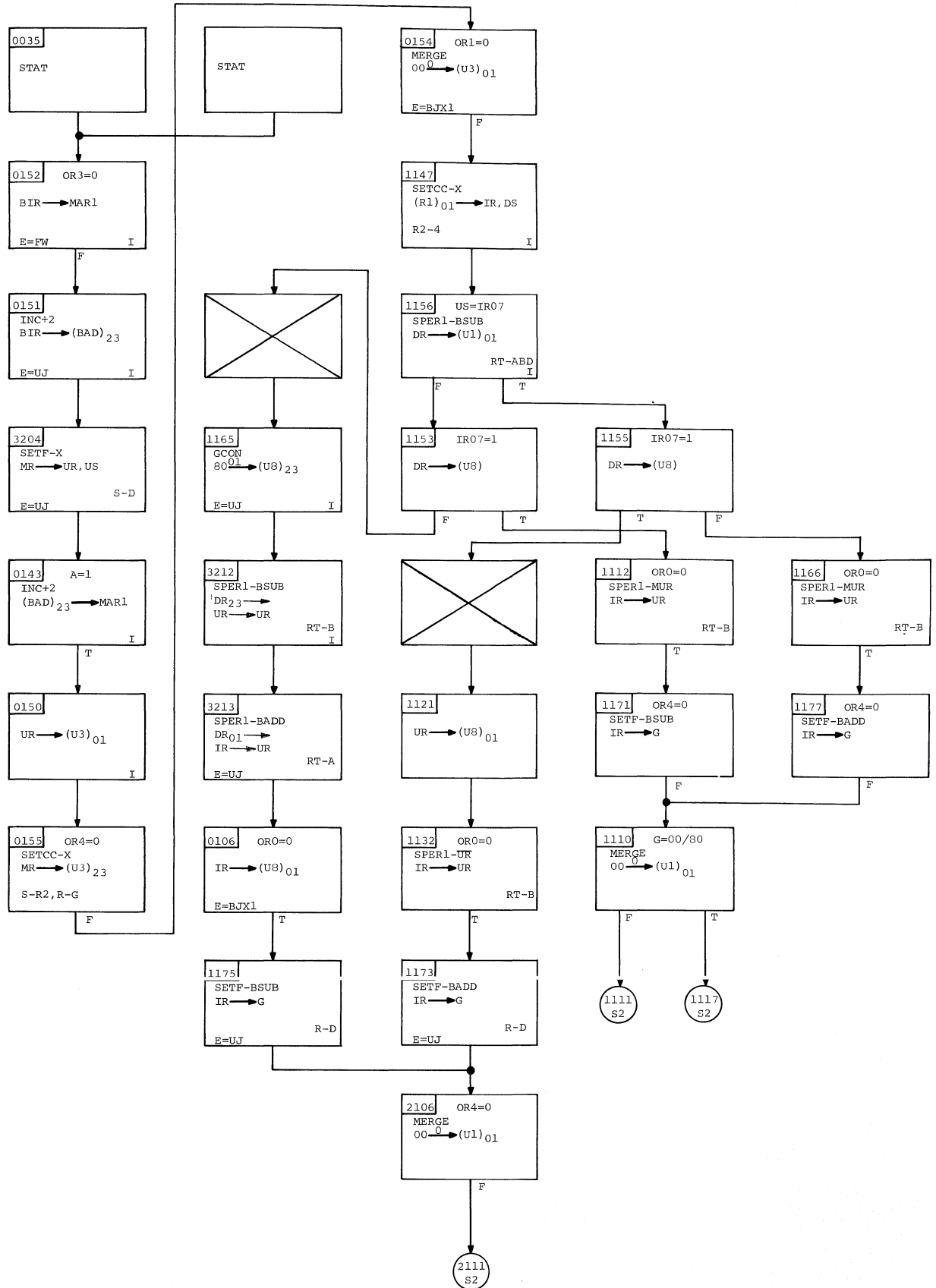


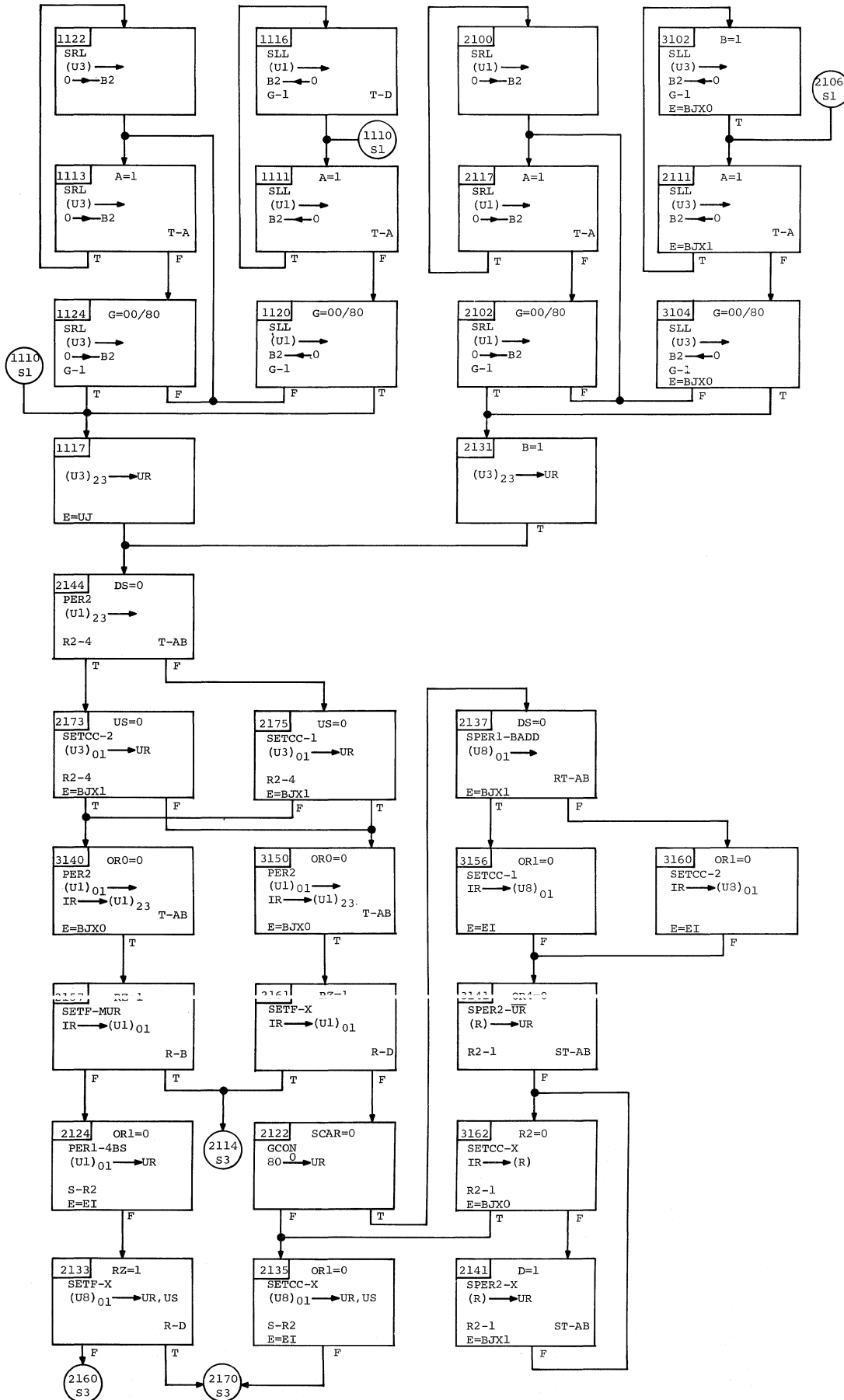


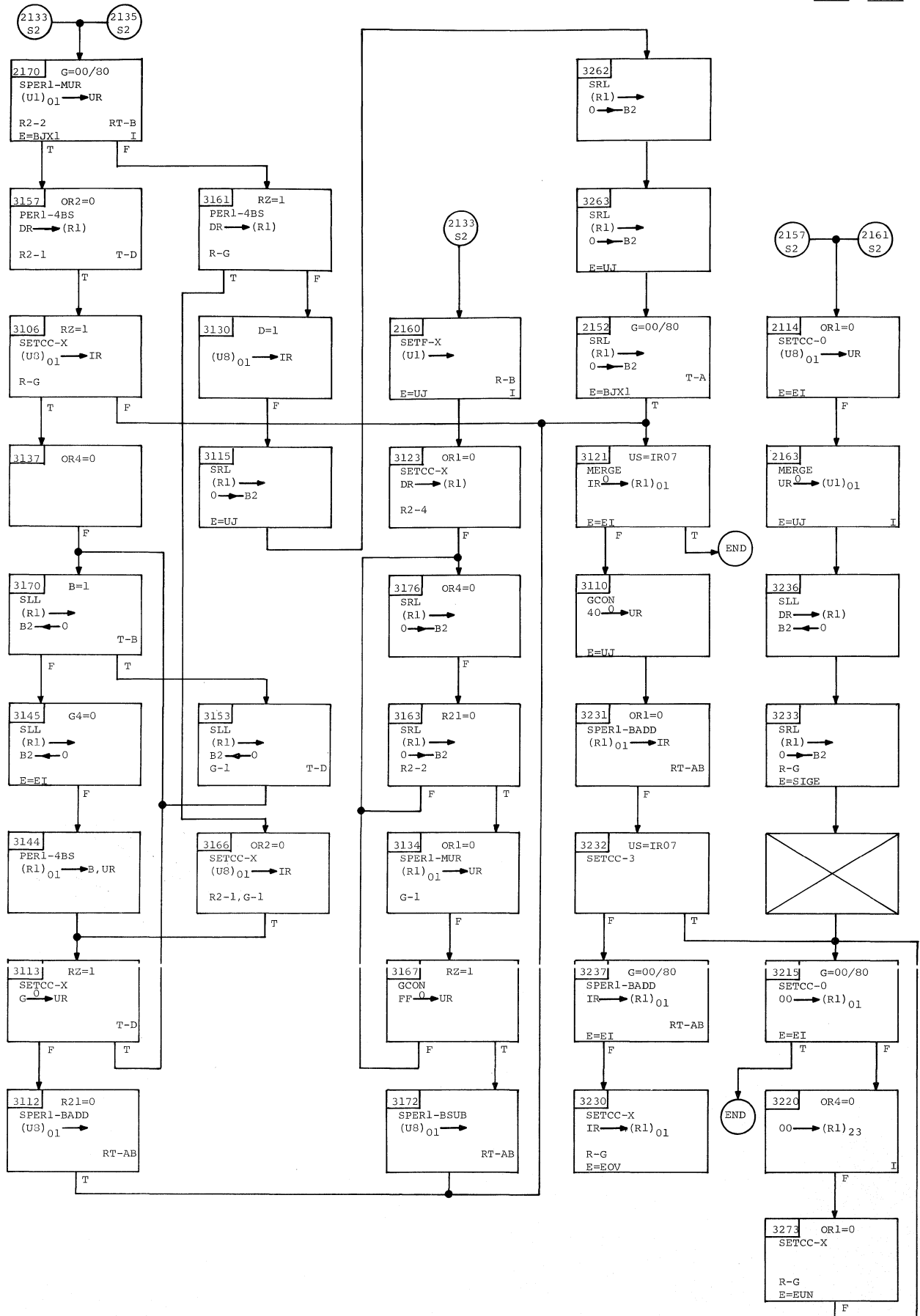




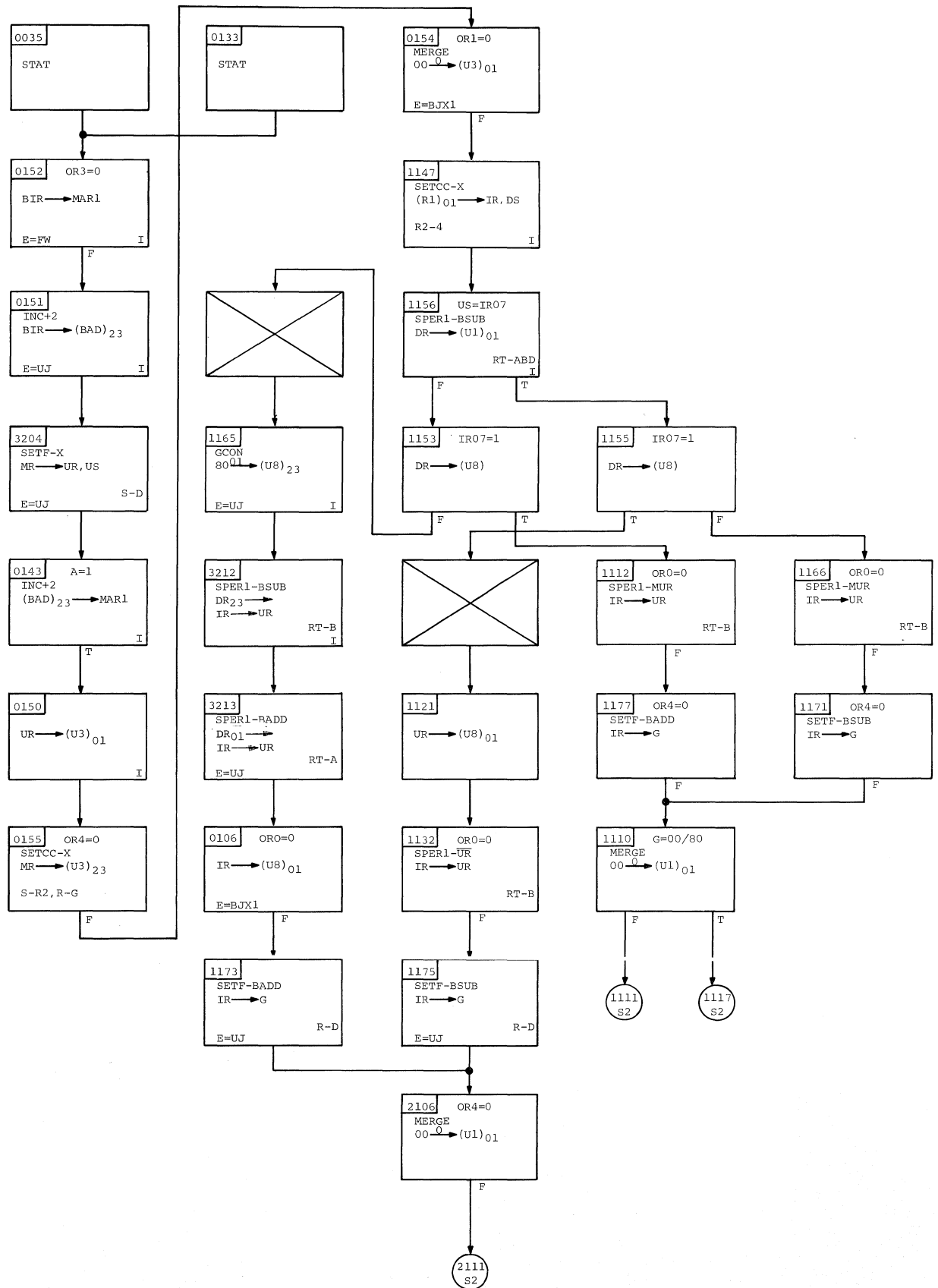




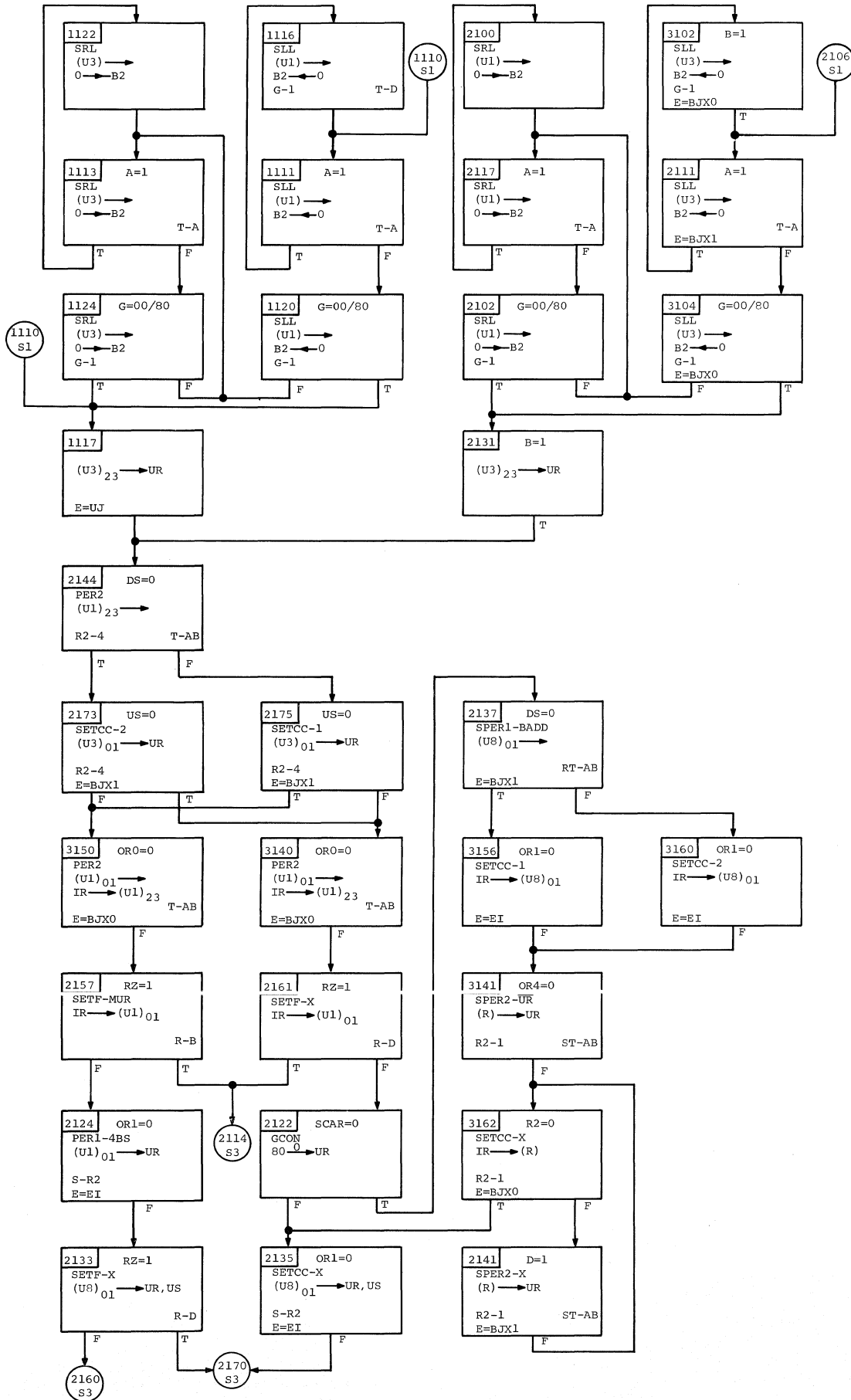




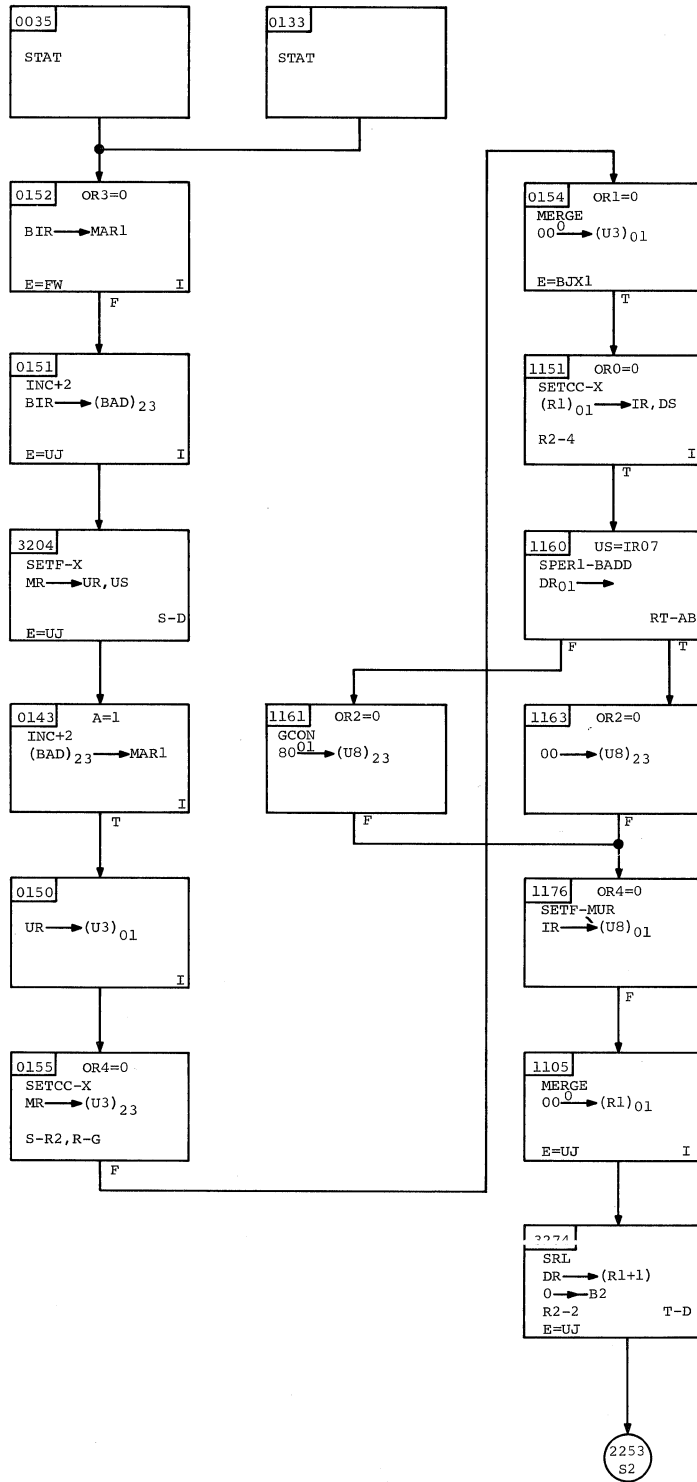
SUBTRACT NORMALIZED SHORT SNE RX 7B



# SUBTRACT NORMALIZED SHORT SNE RX 7B

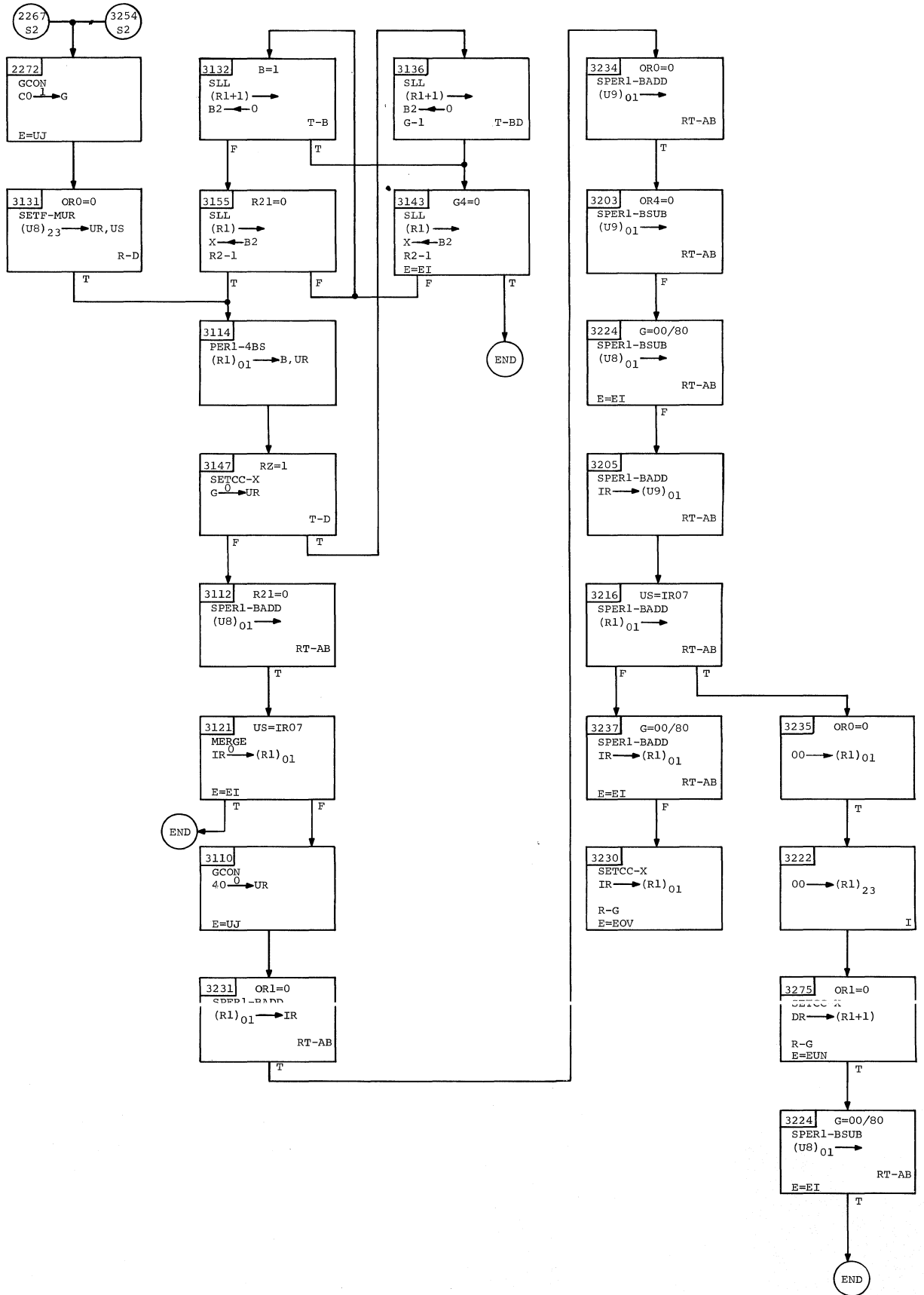


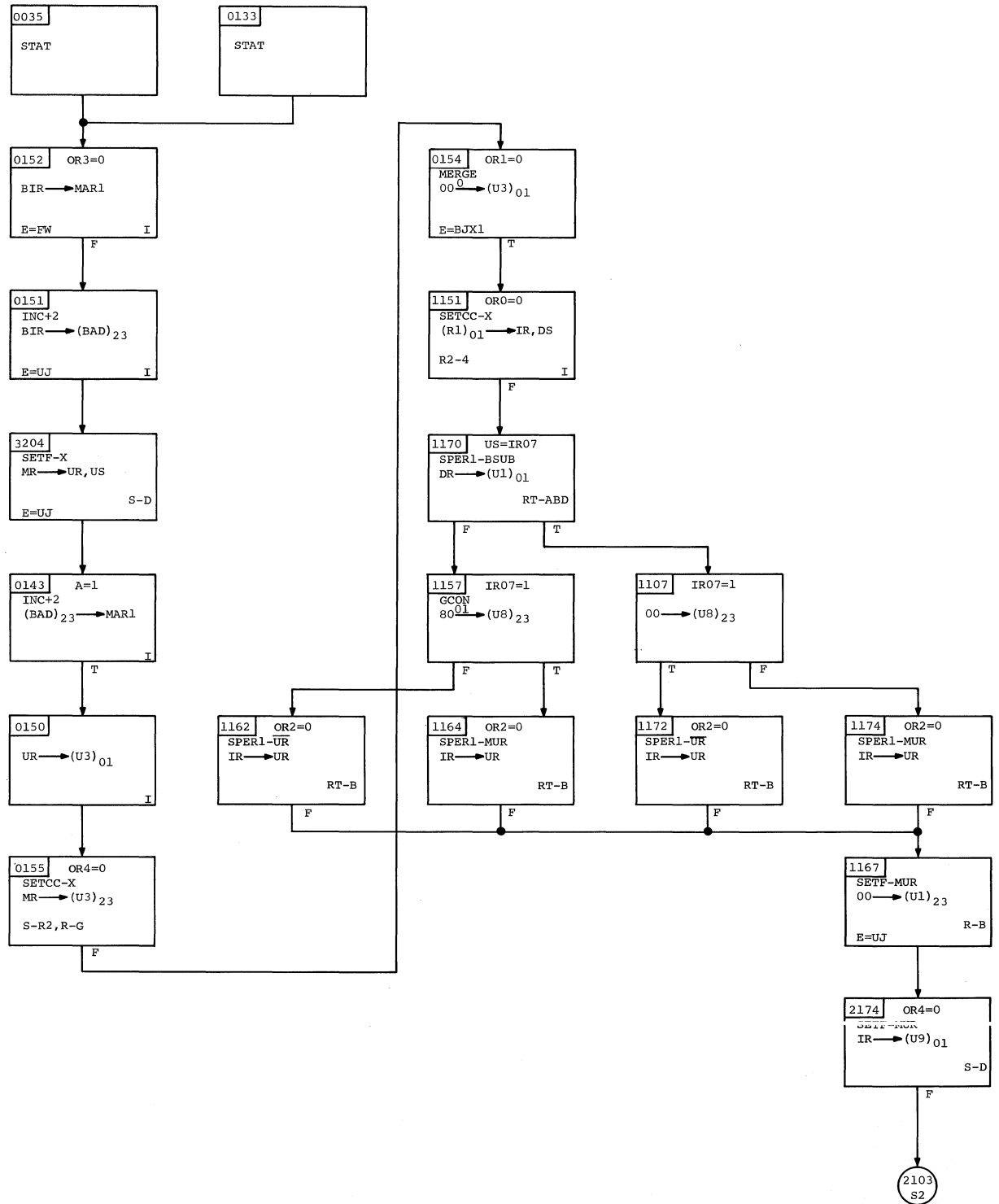


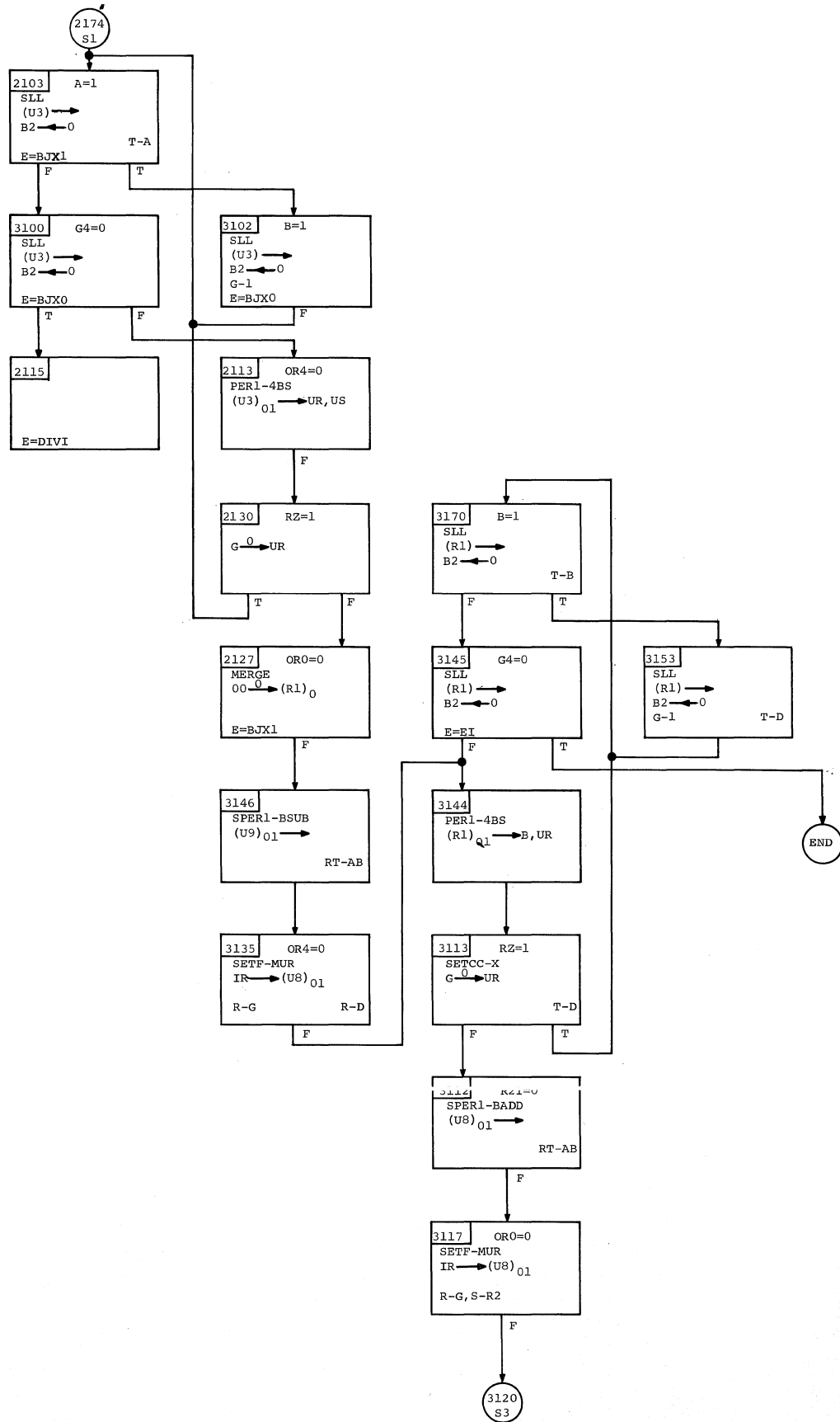


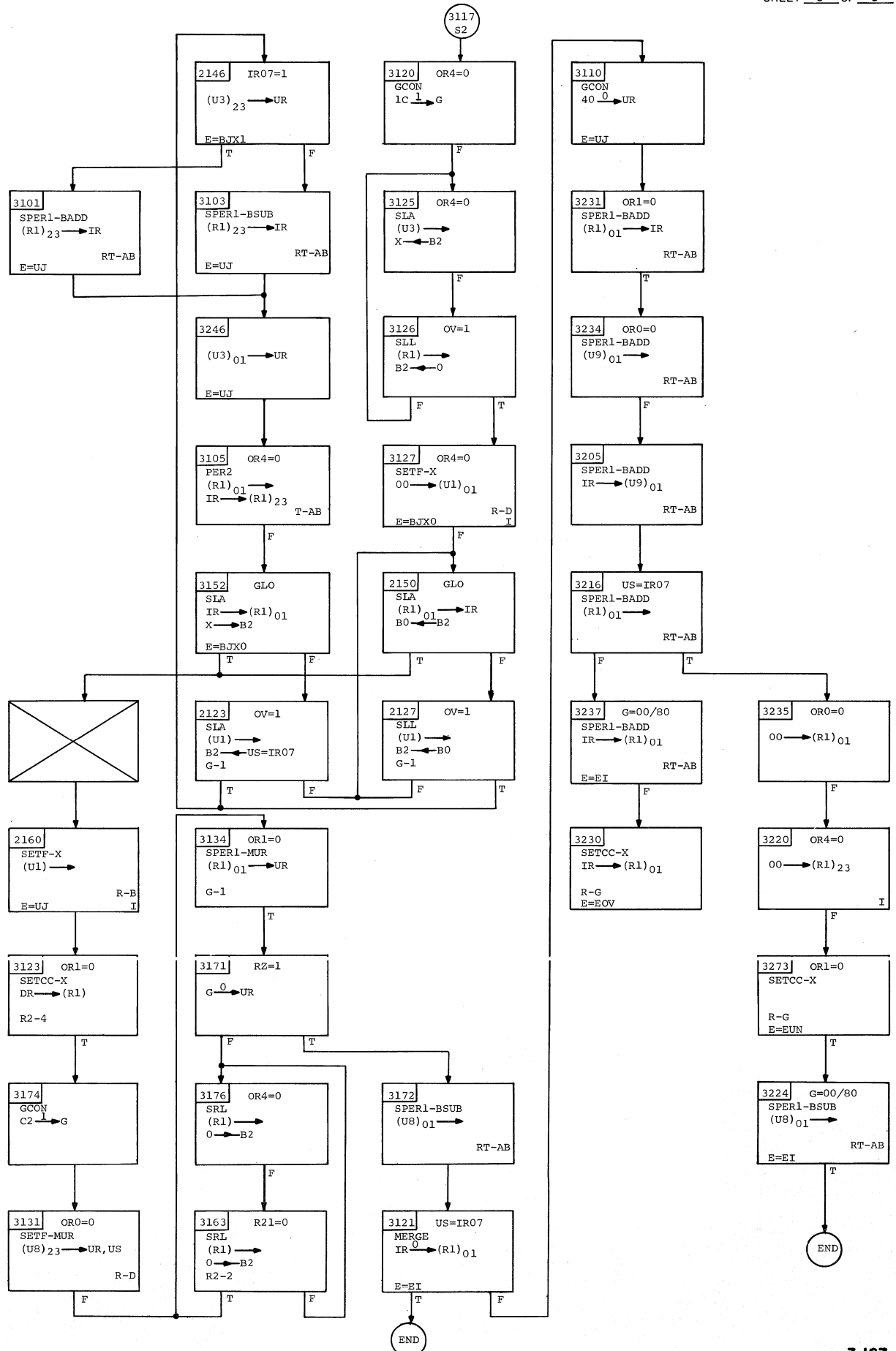


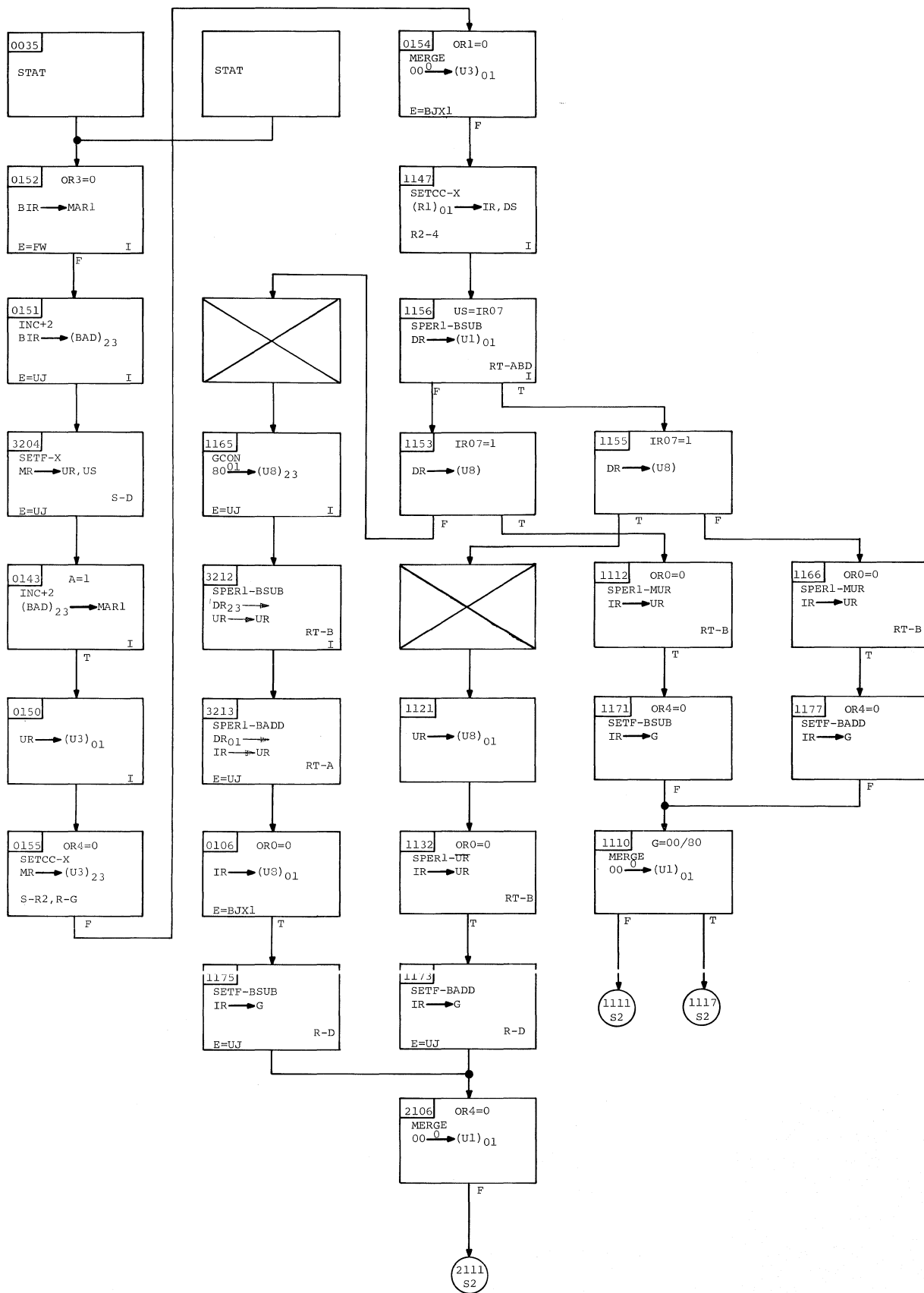


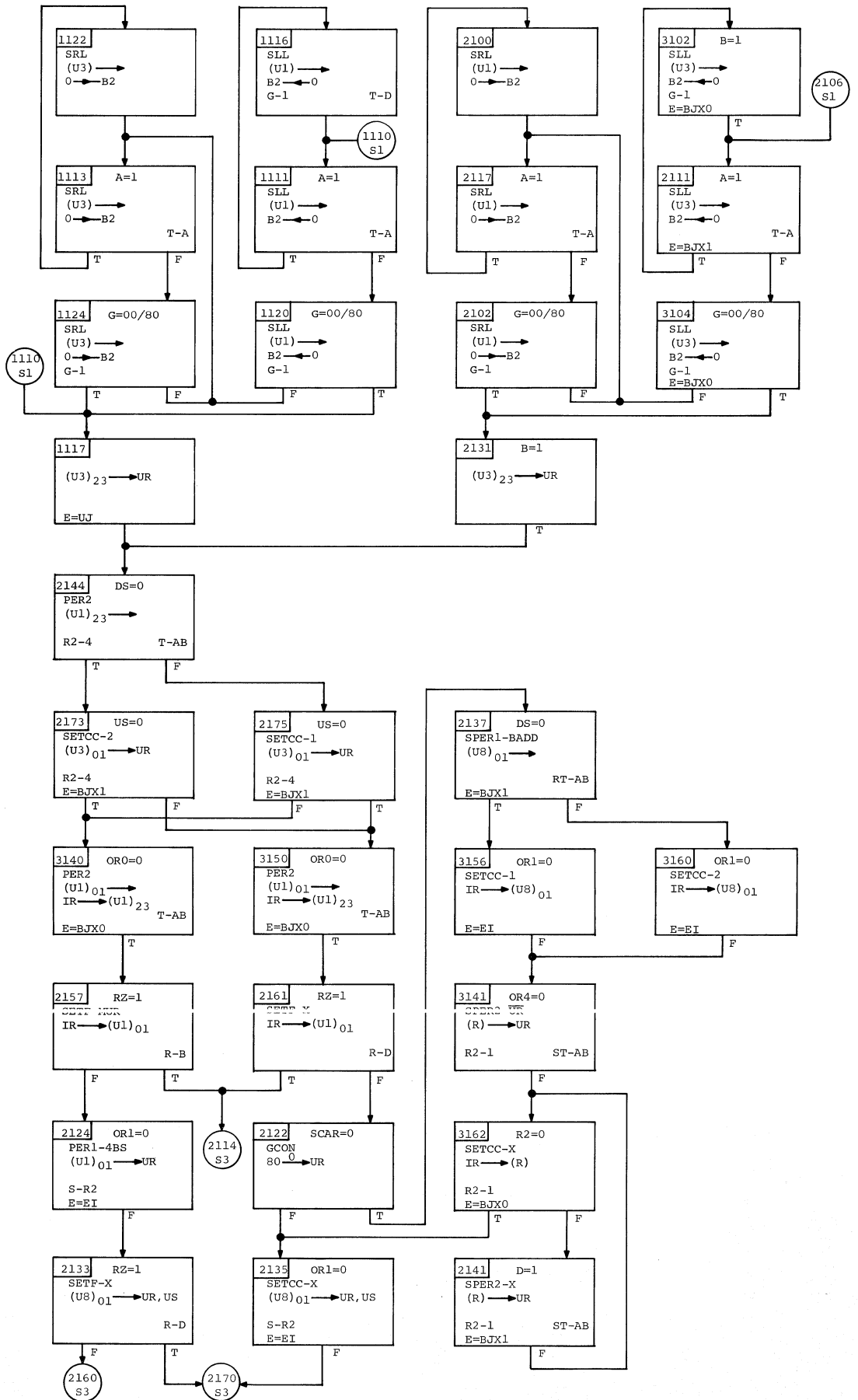




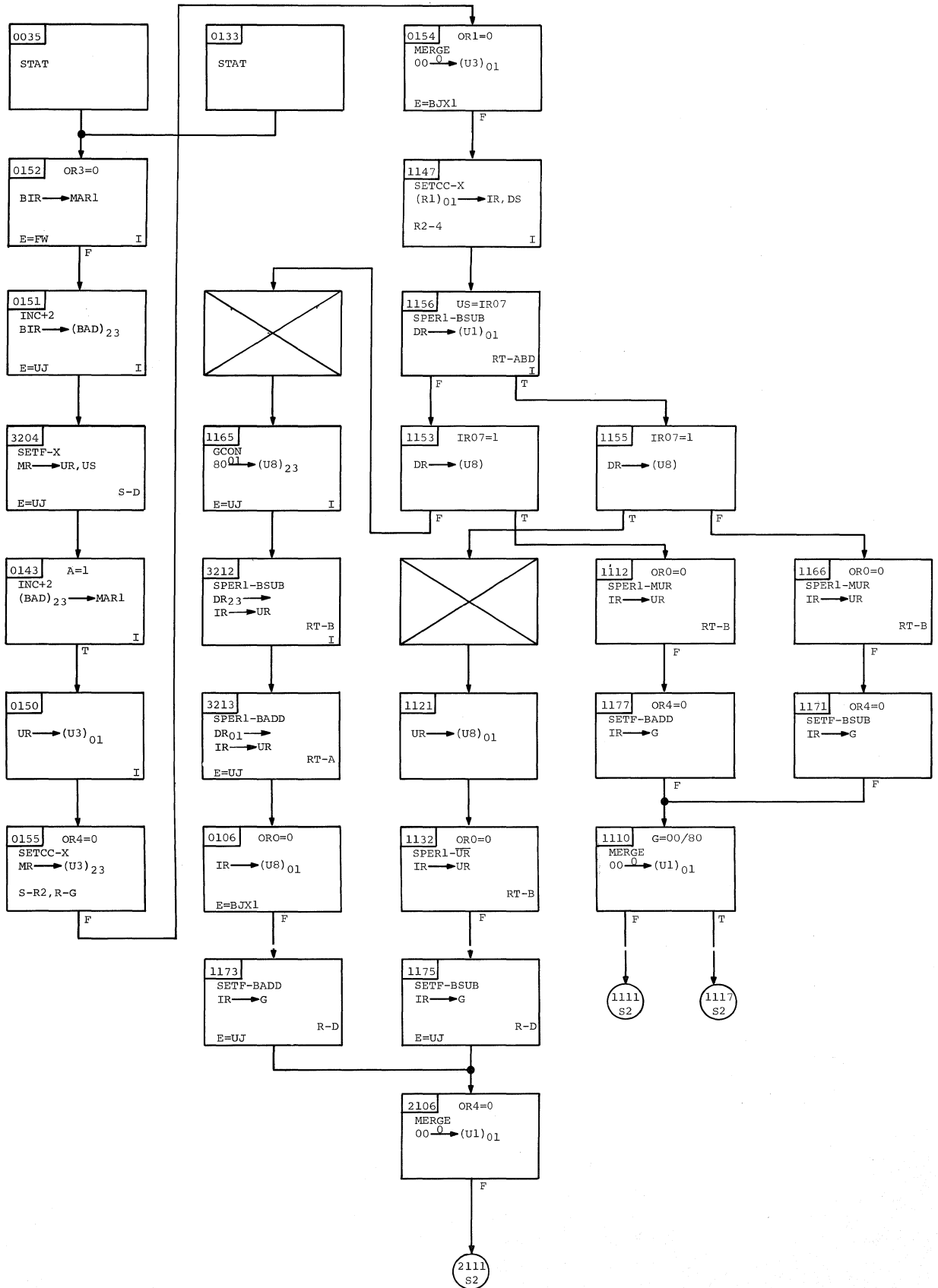






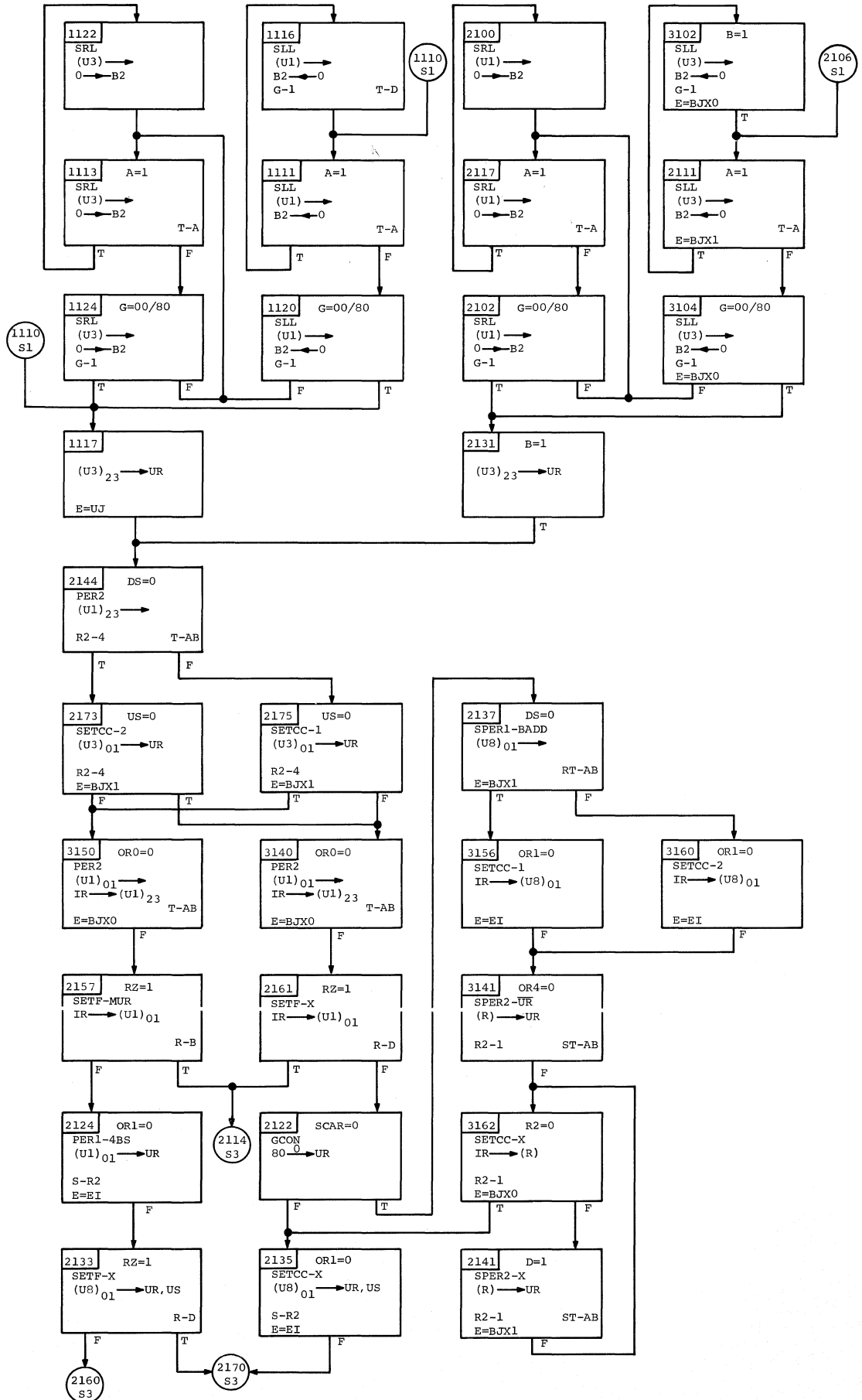


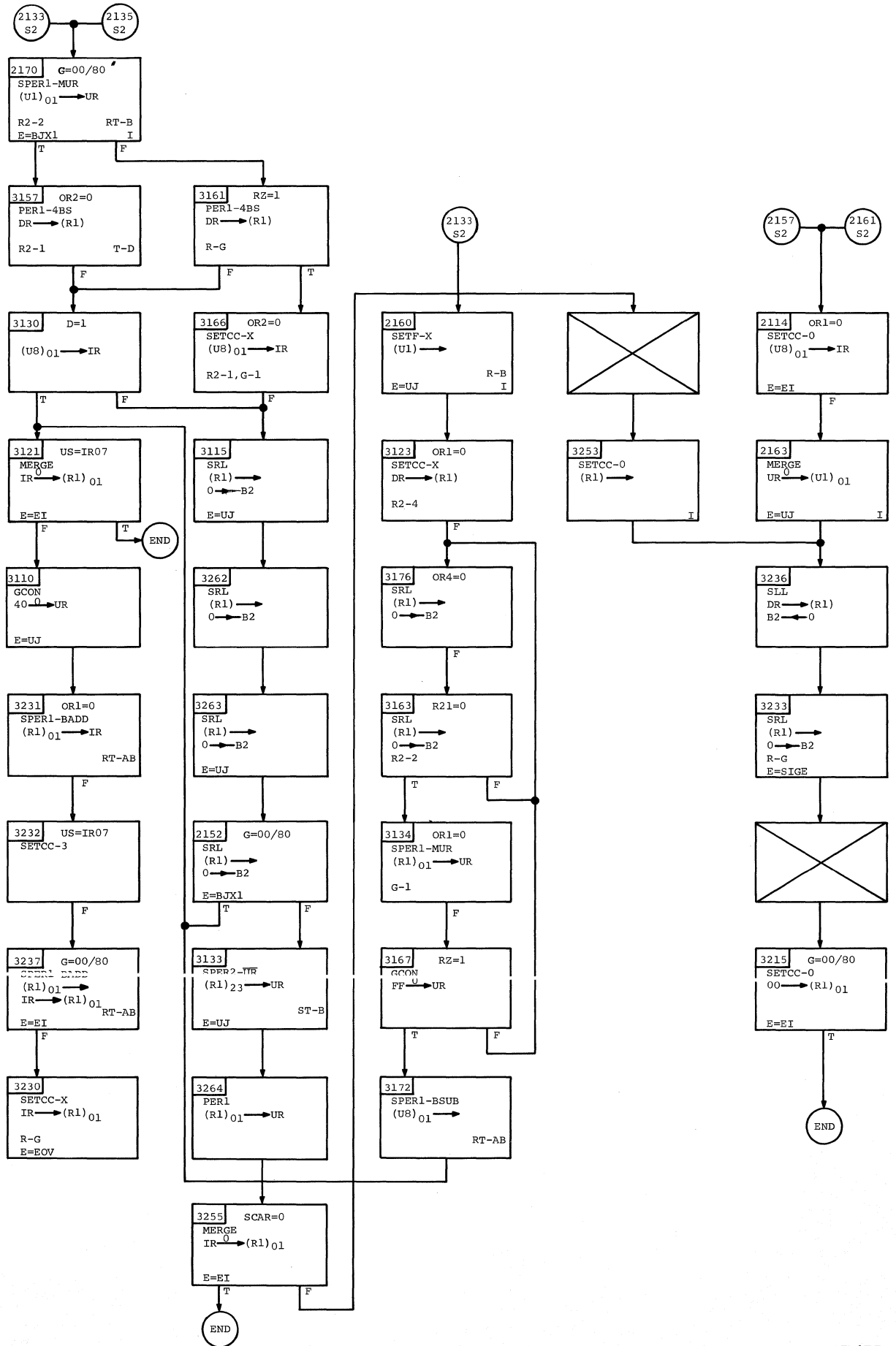


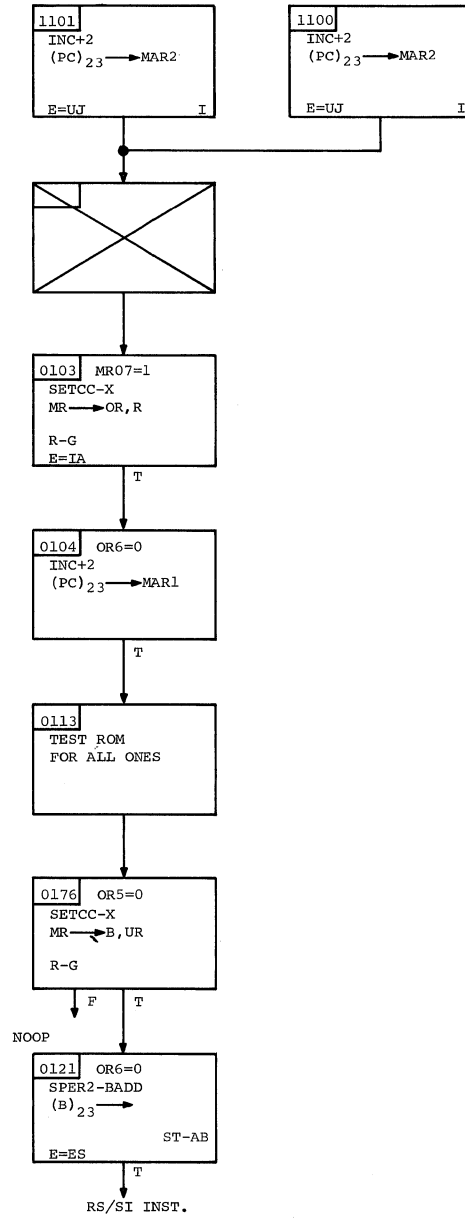


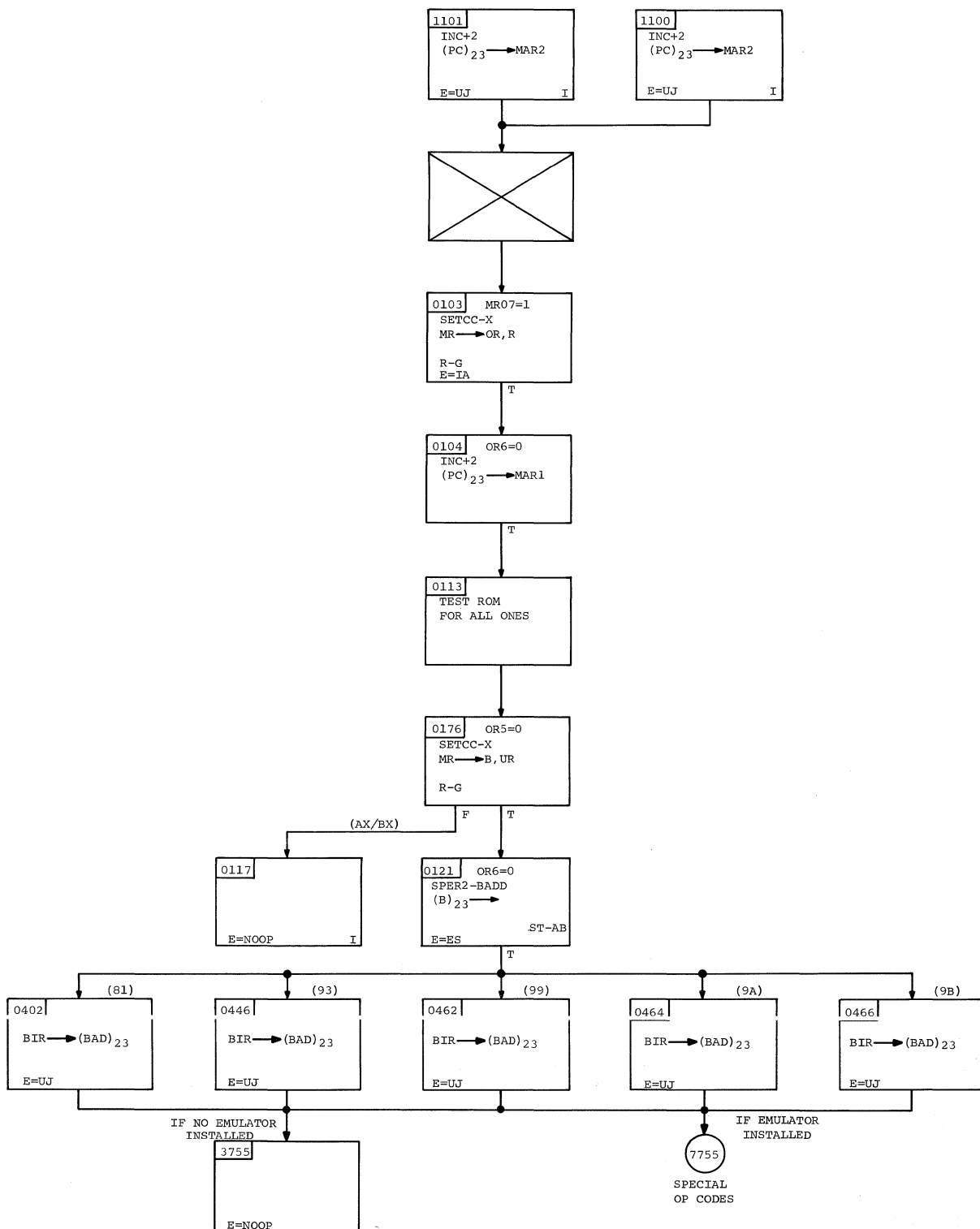


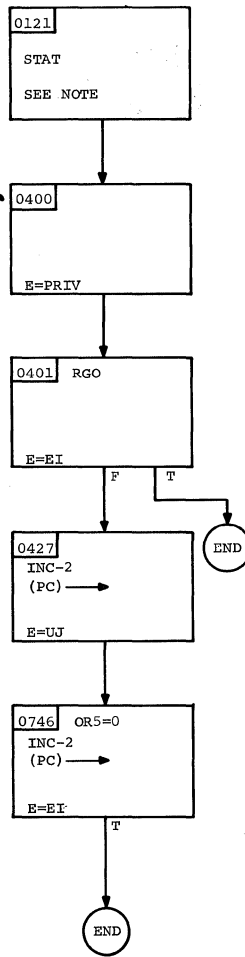
SUBTRACT UNNORMALIZED SHORT SU RX 7F



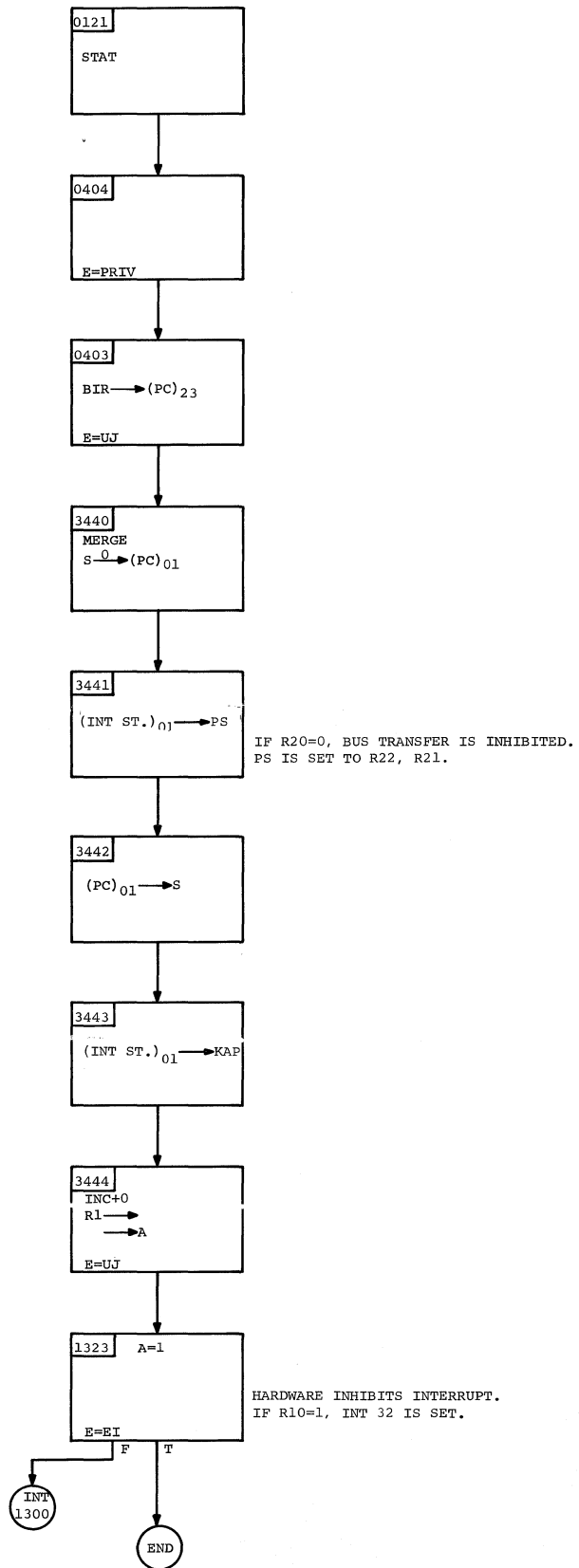


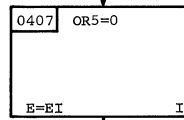
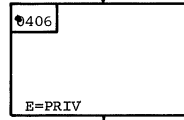
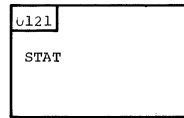






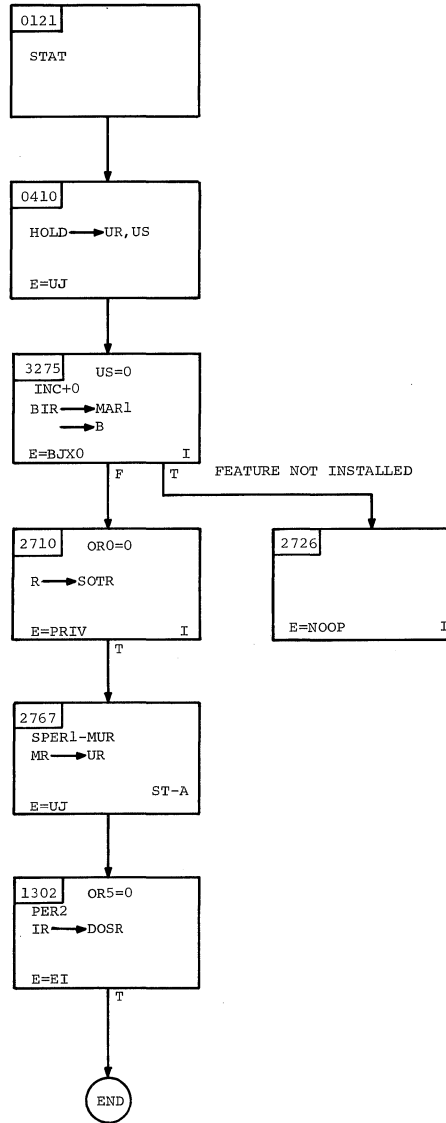
NOTE: DURING STATICIZING, HARDWARE WILL SET OR RESET ALI OR ISIM AS PER R REGISTER, IF IN PRIVILEGED MODE.



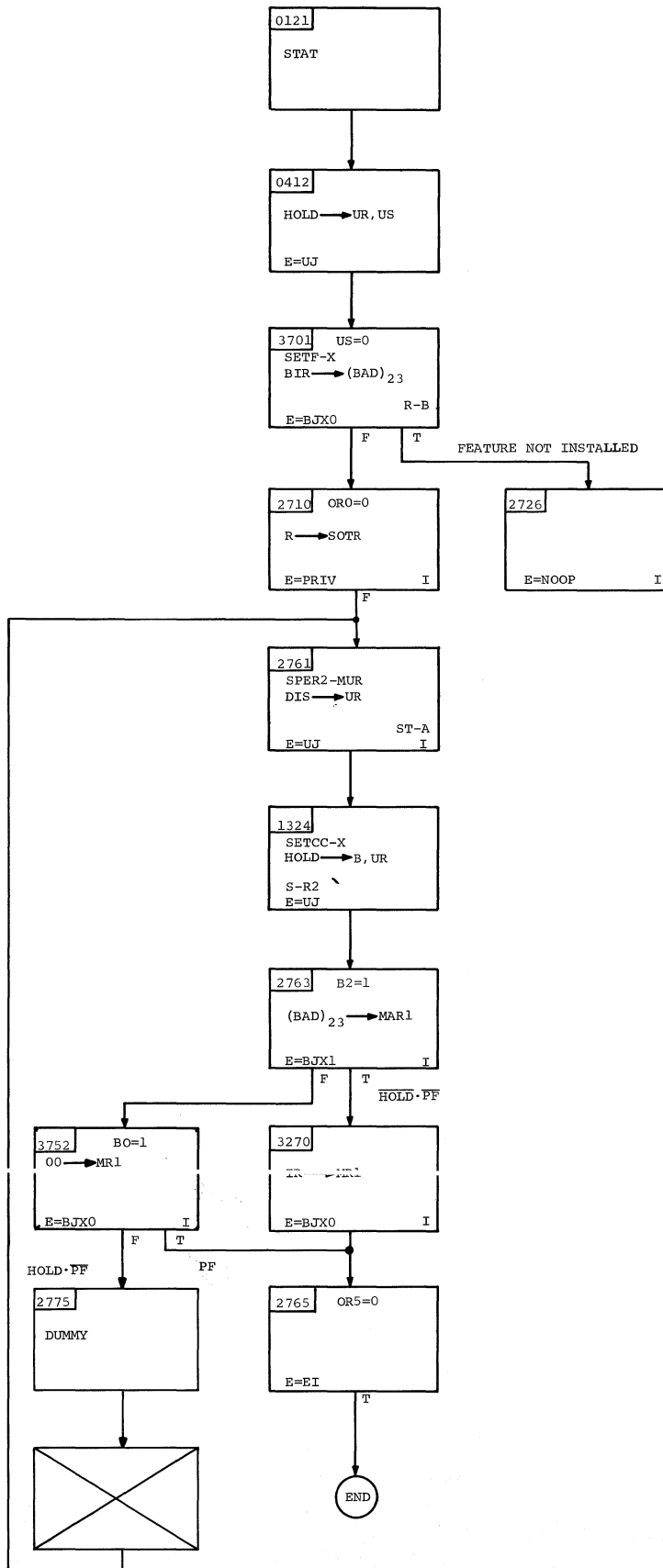


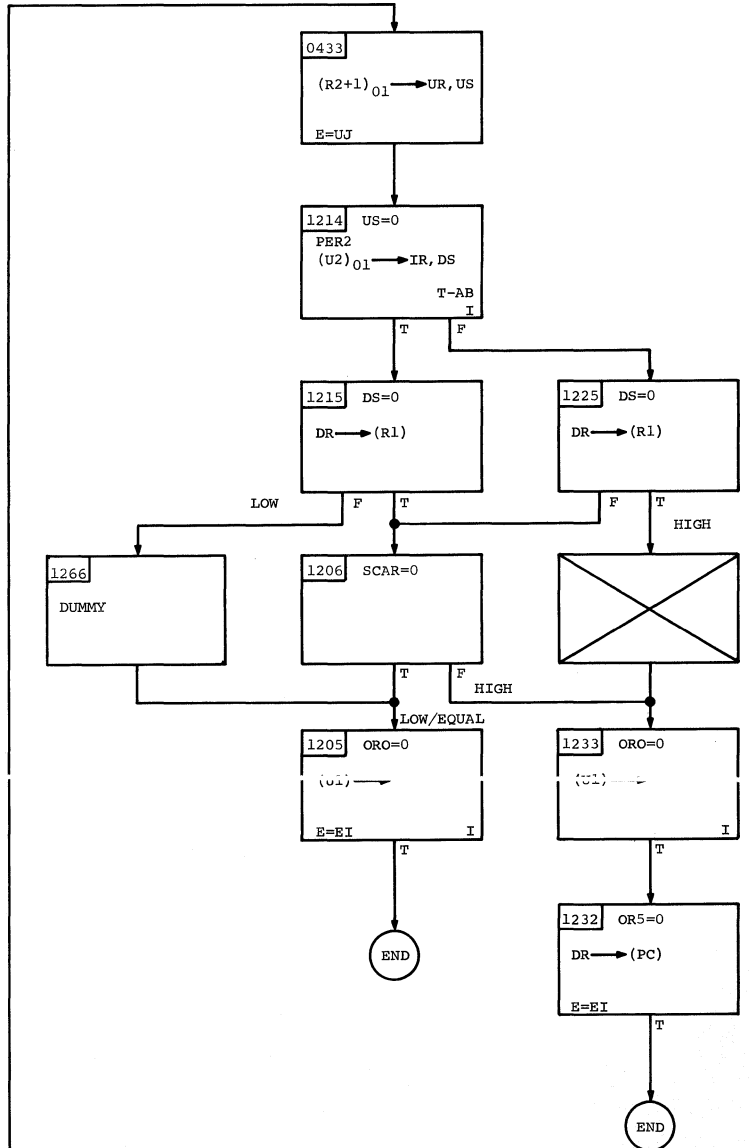
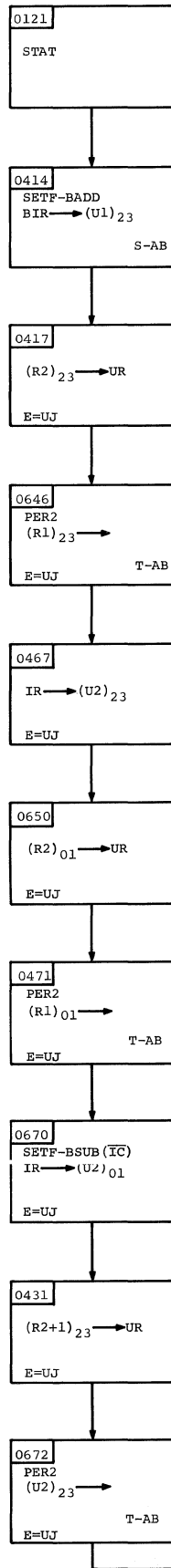
COUNTER IS SET TO R AND COUNTING BEGINS.  
HARDWARE INHIBITS INTERRUPT.

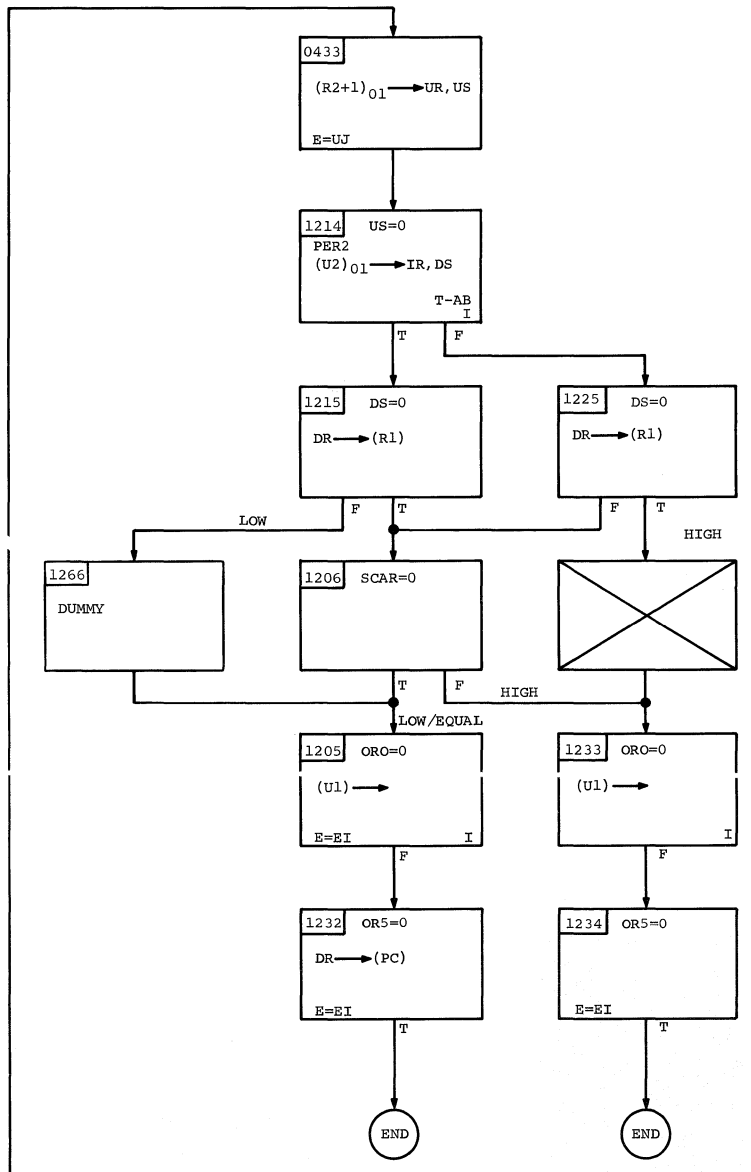
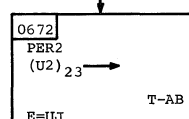
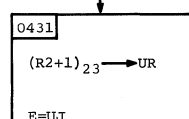
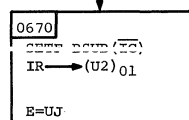
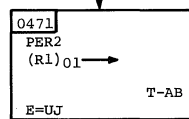
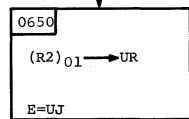
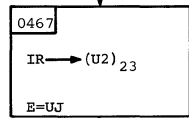
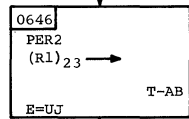
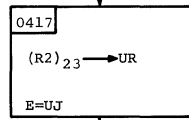
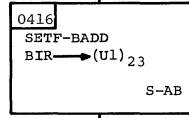
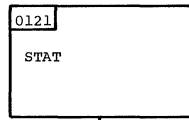


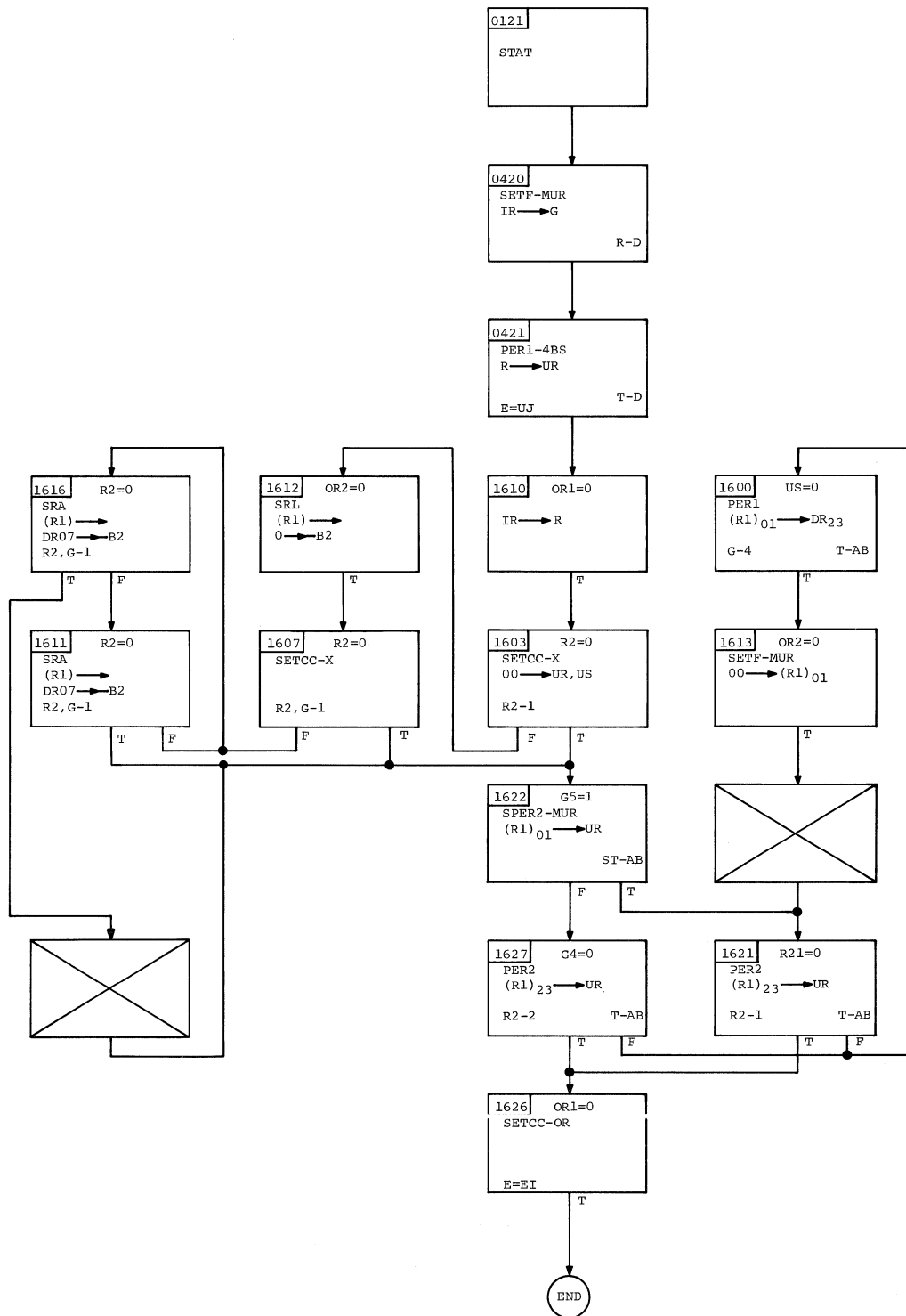


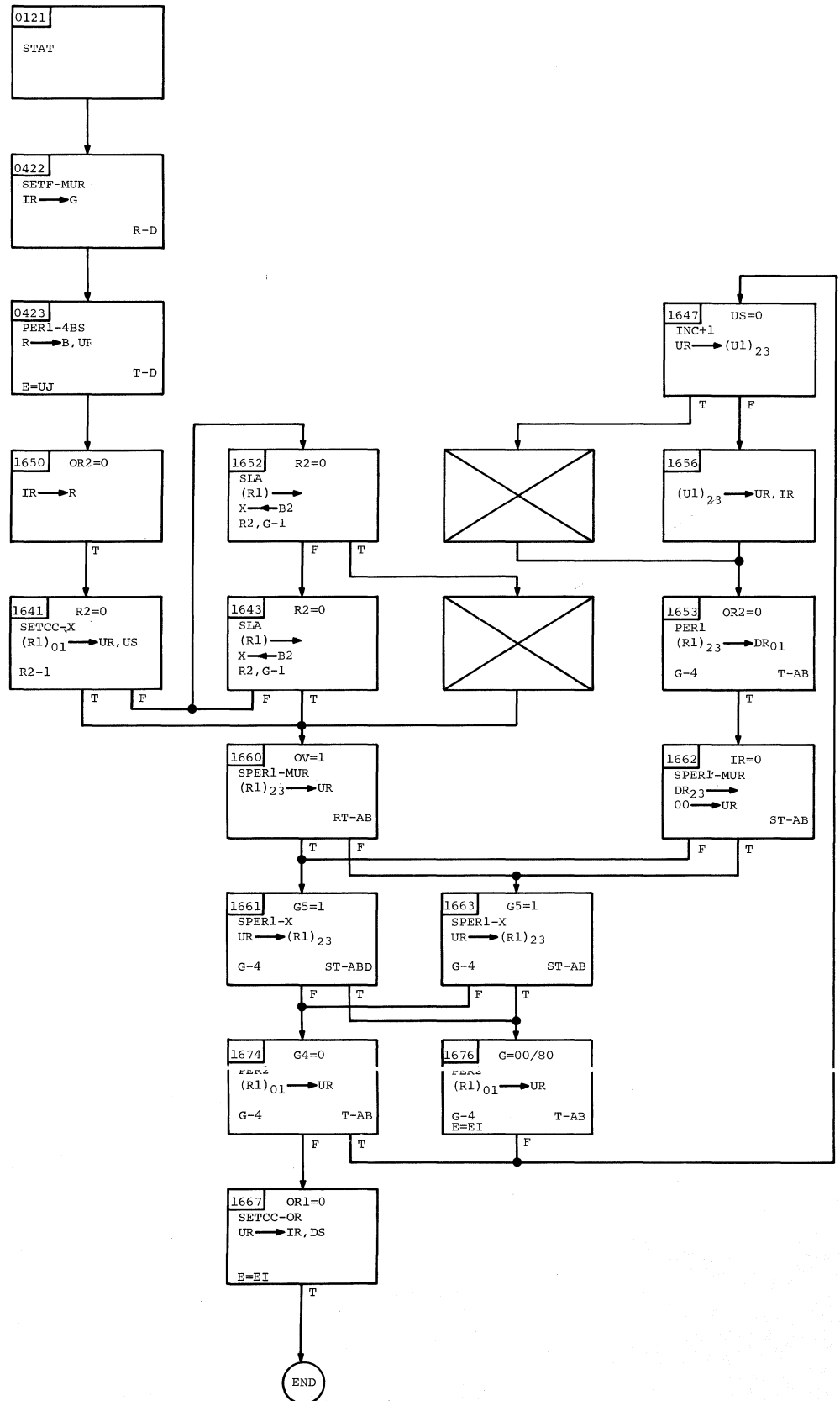


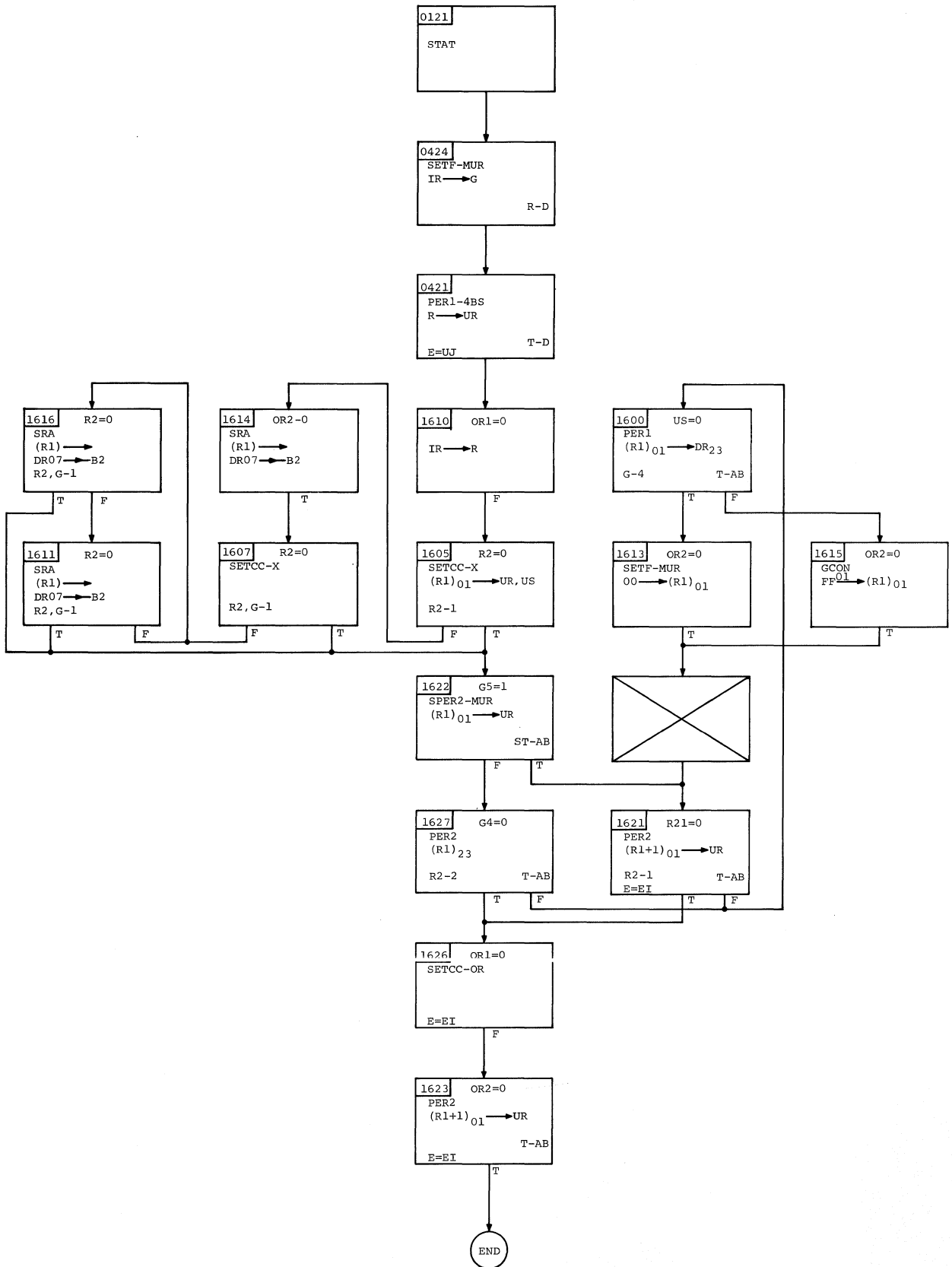


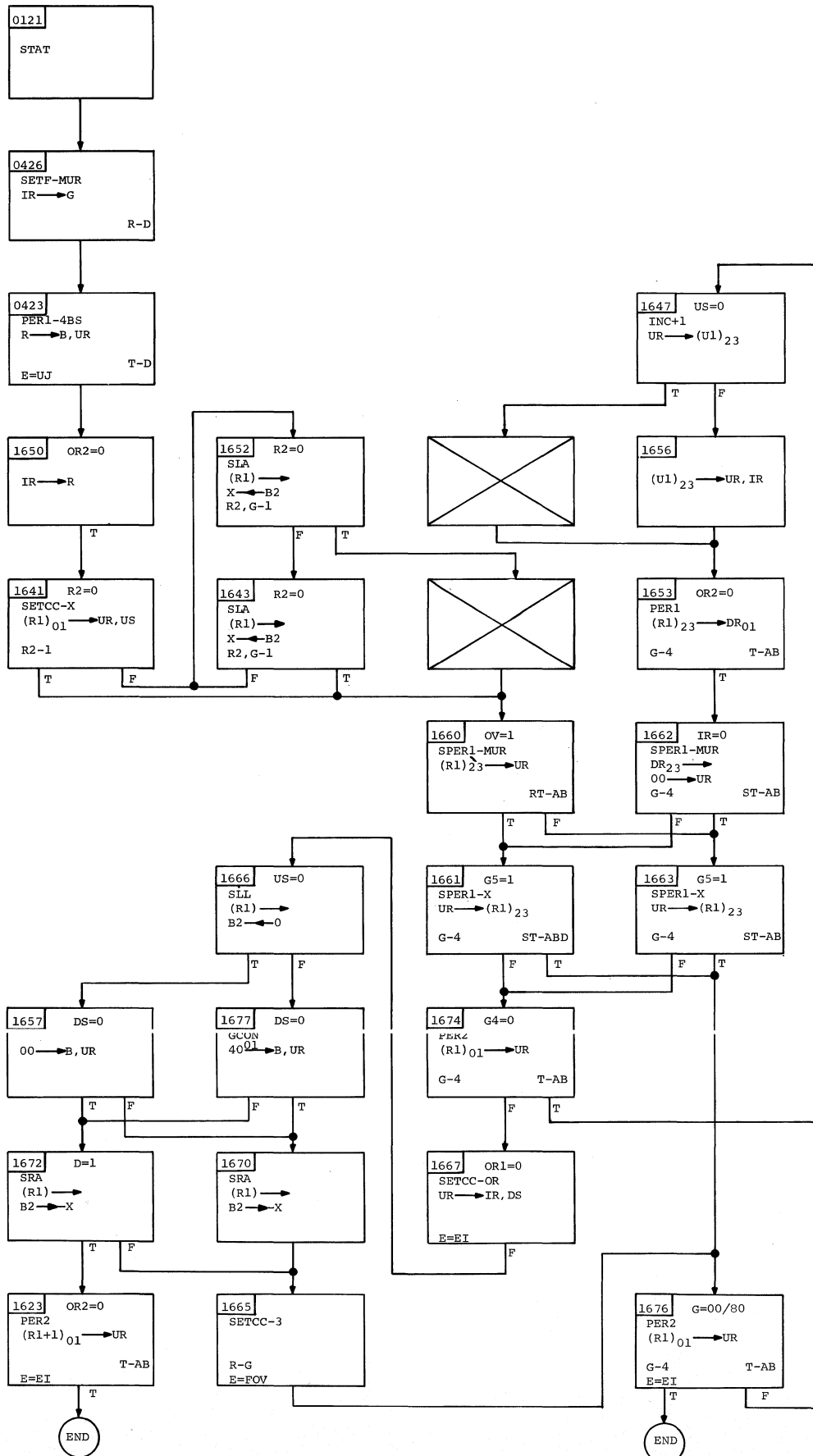


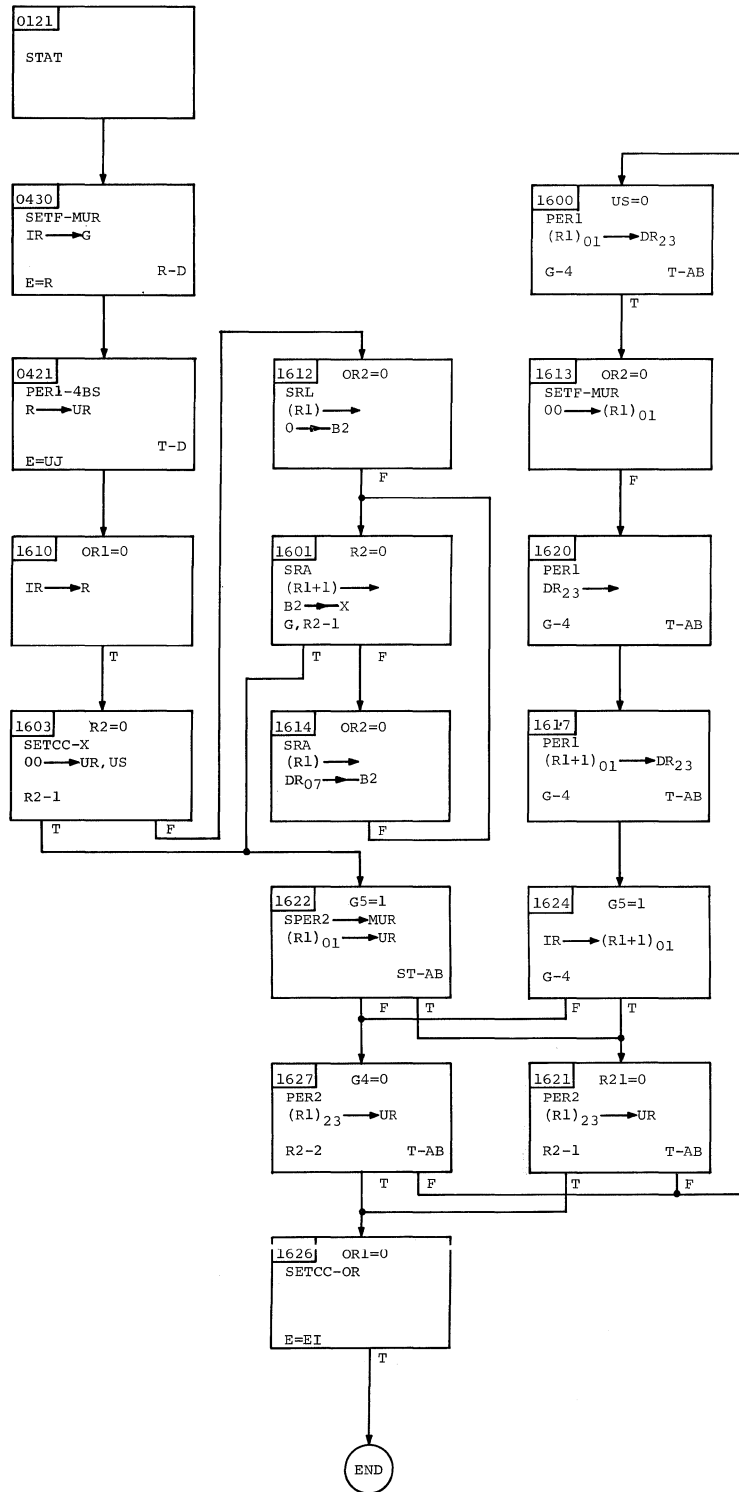




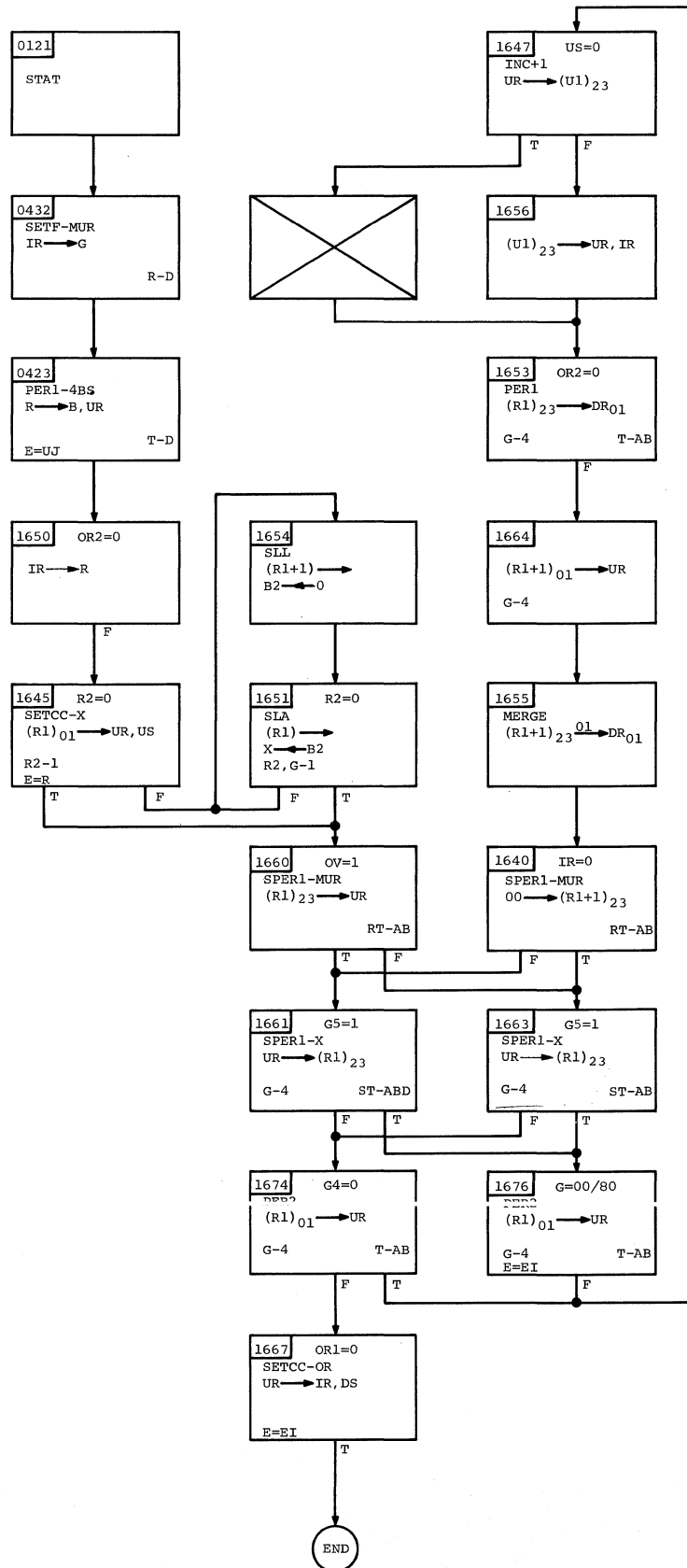


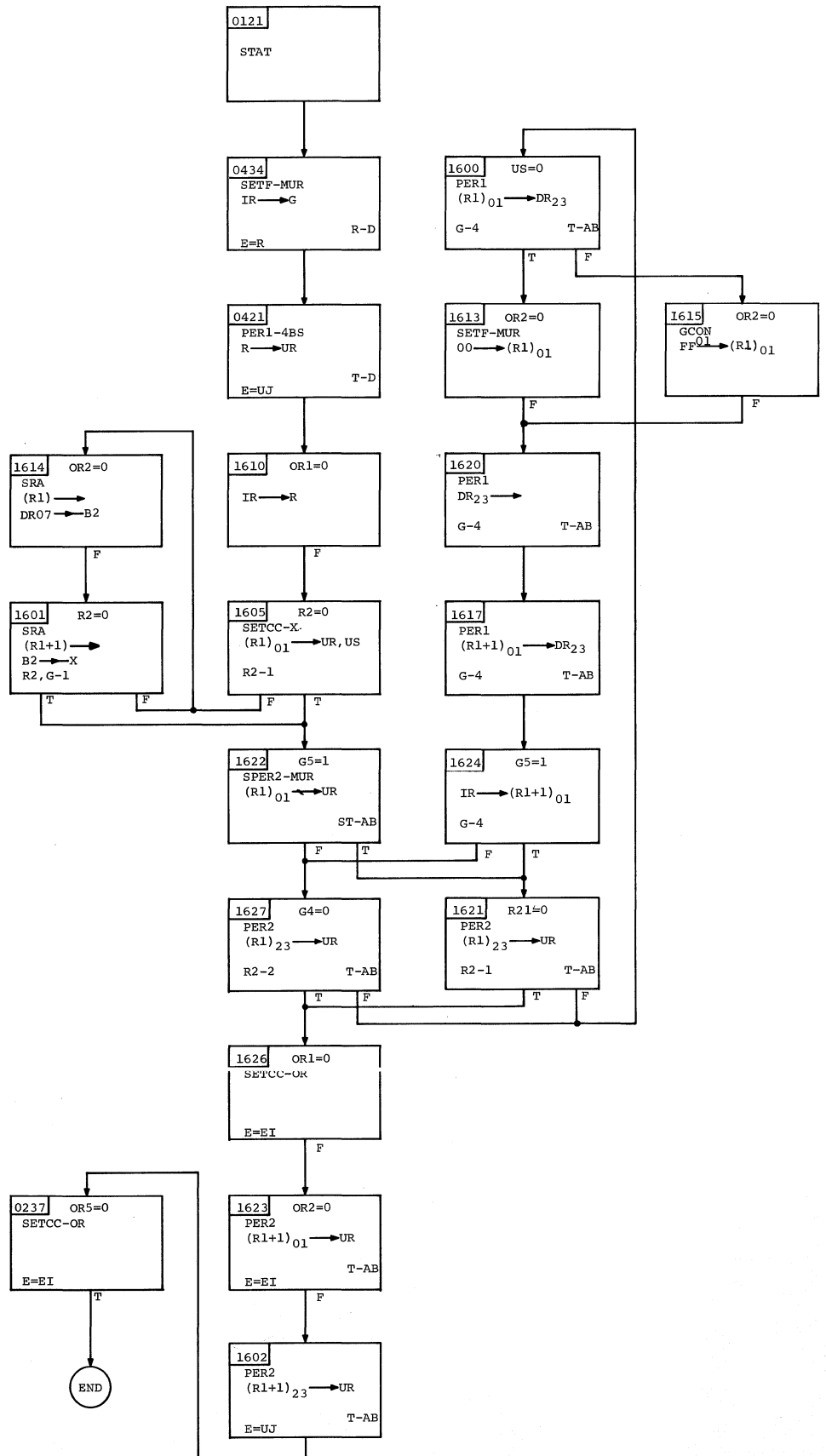


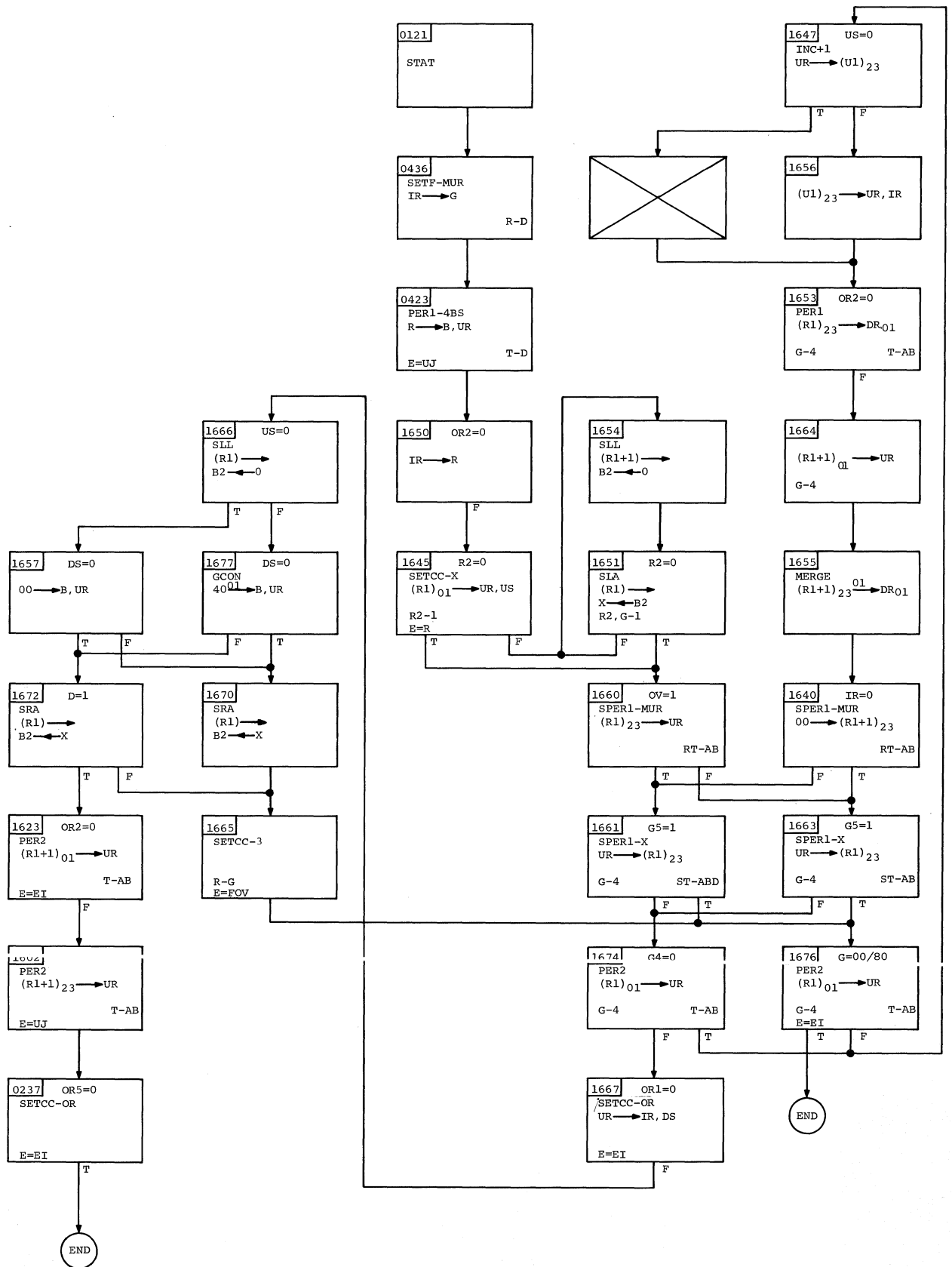


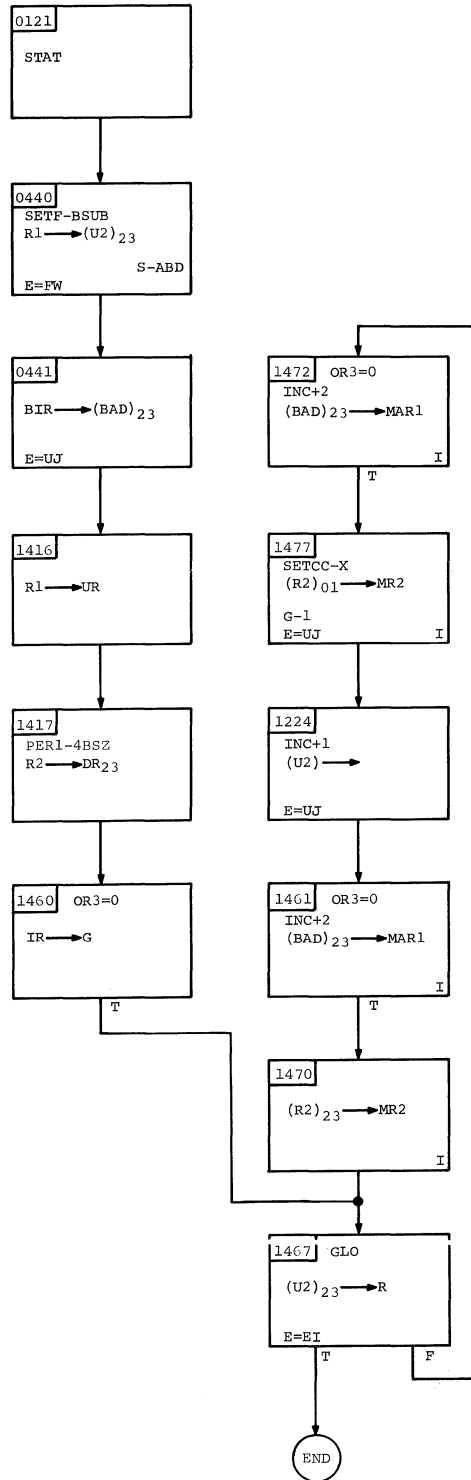


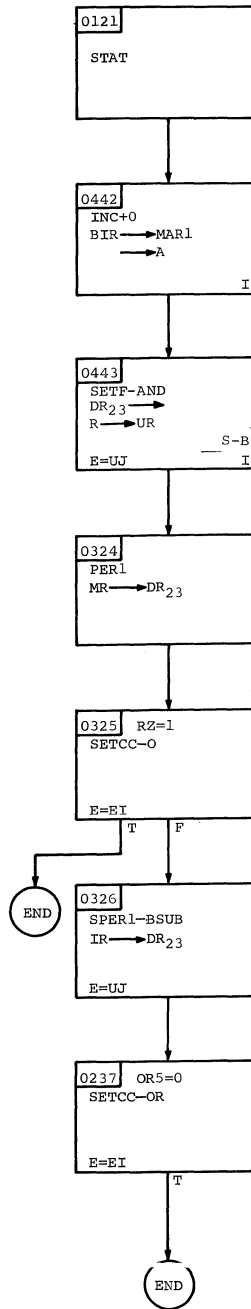


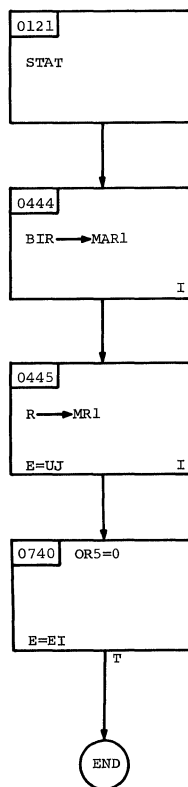


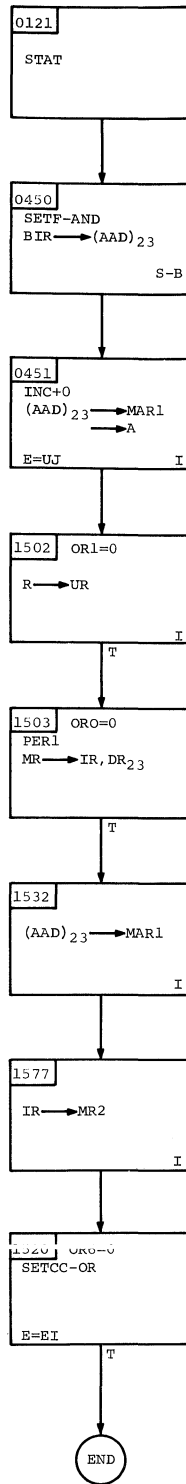


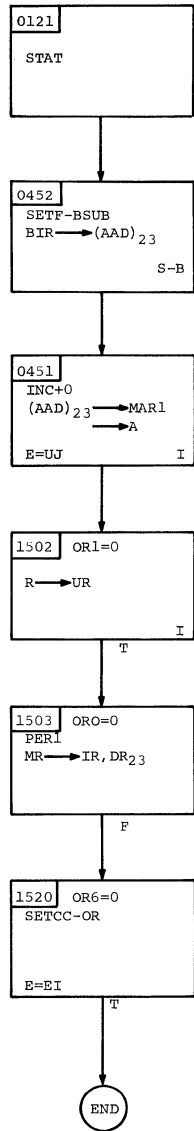




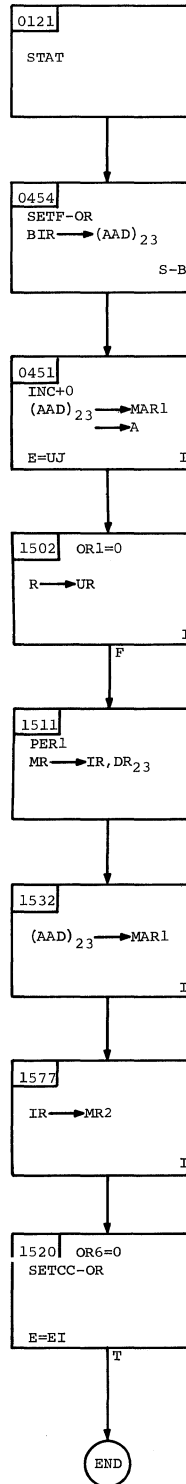


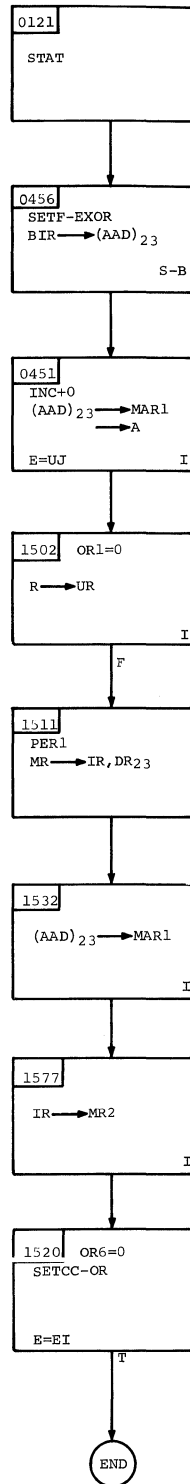


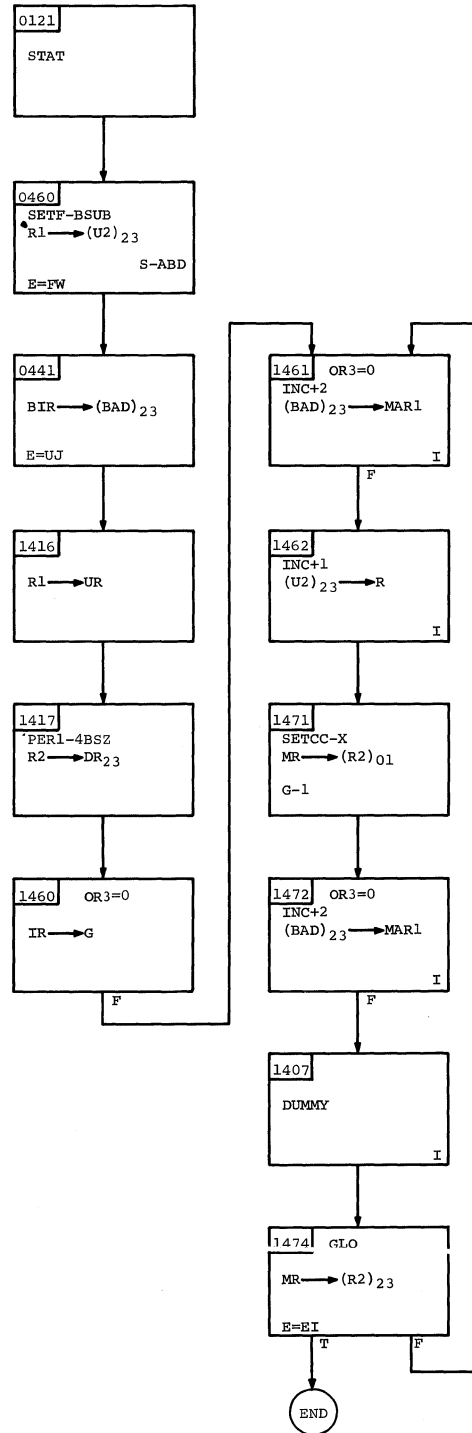


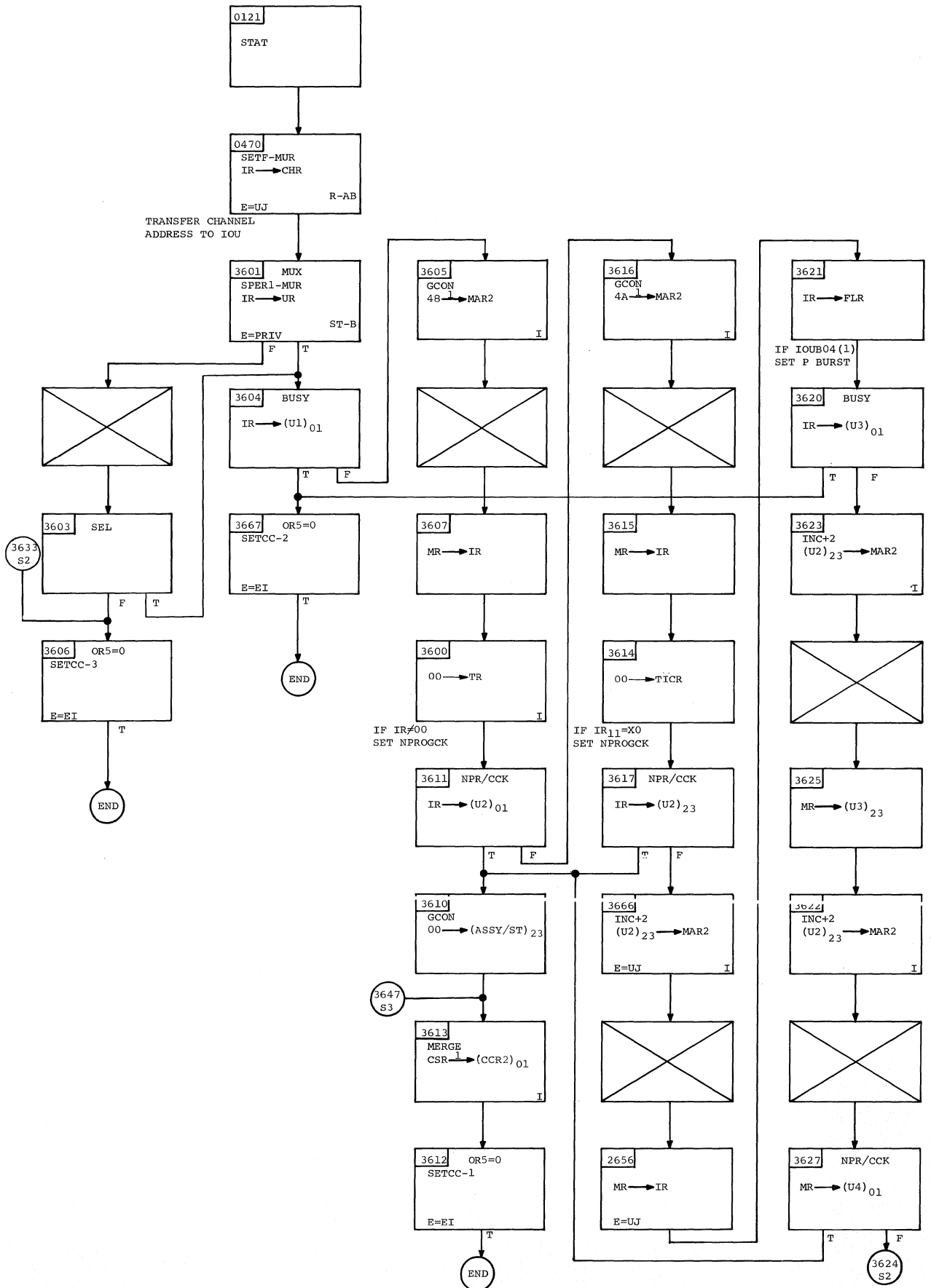


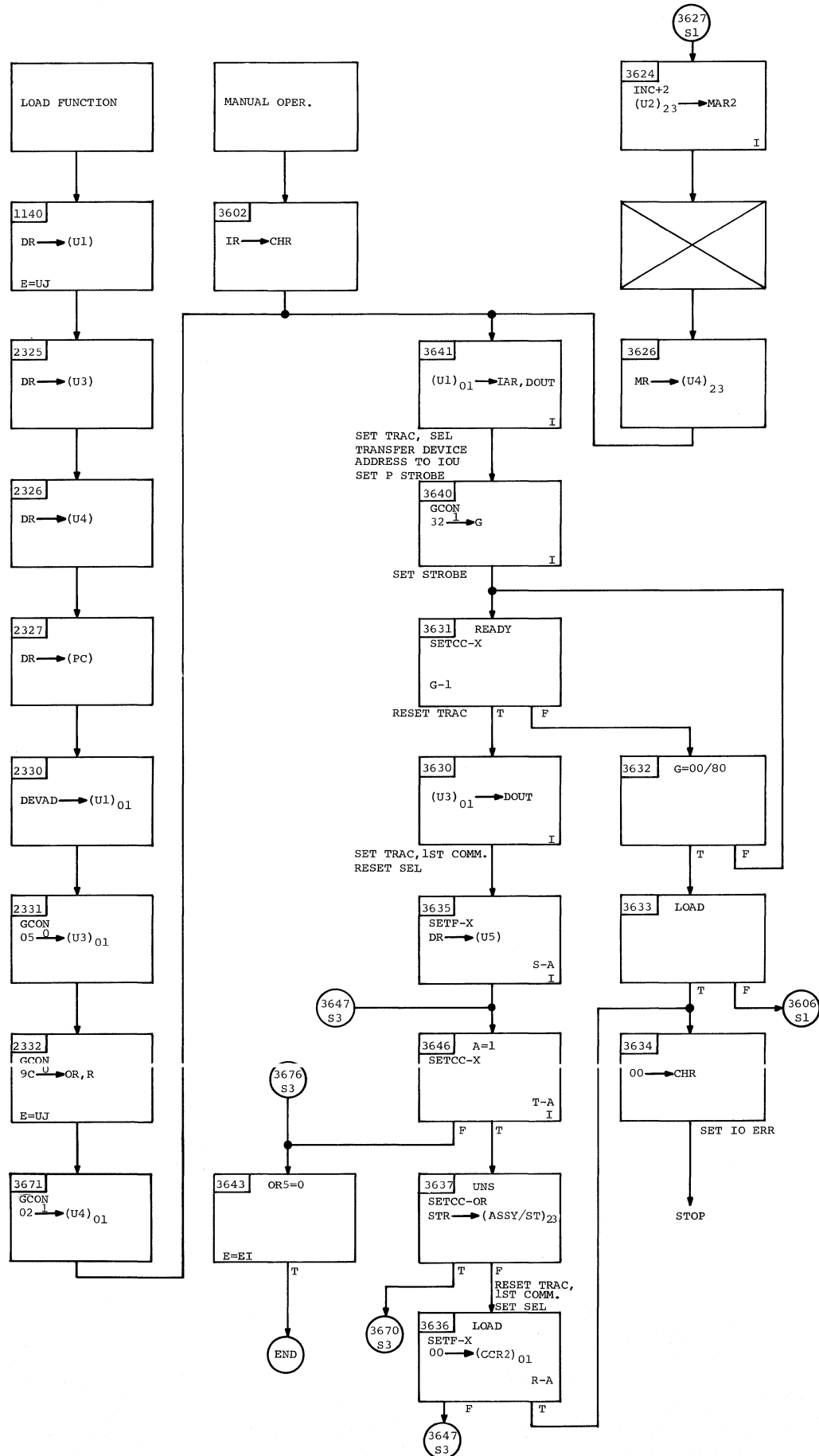


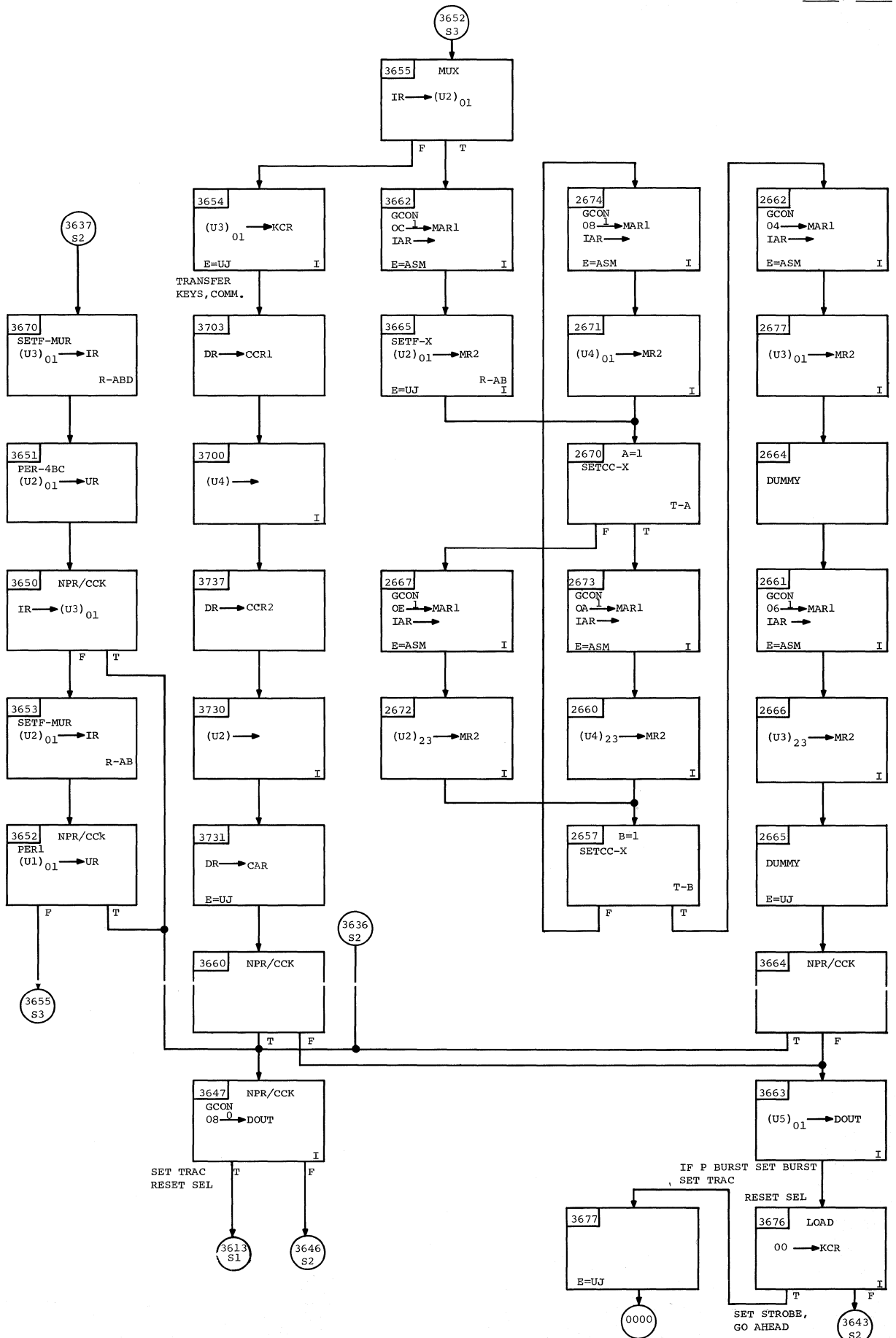


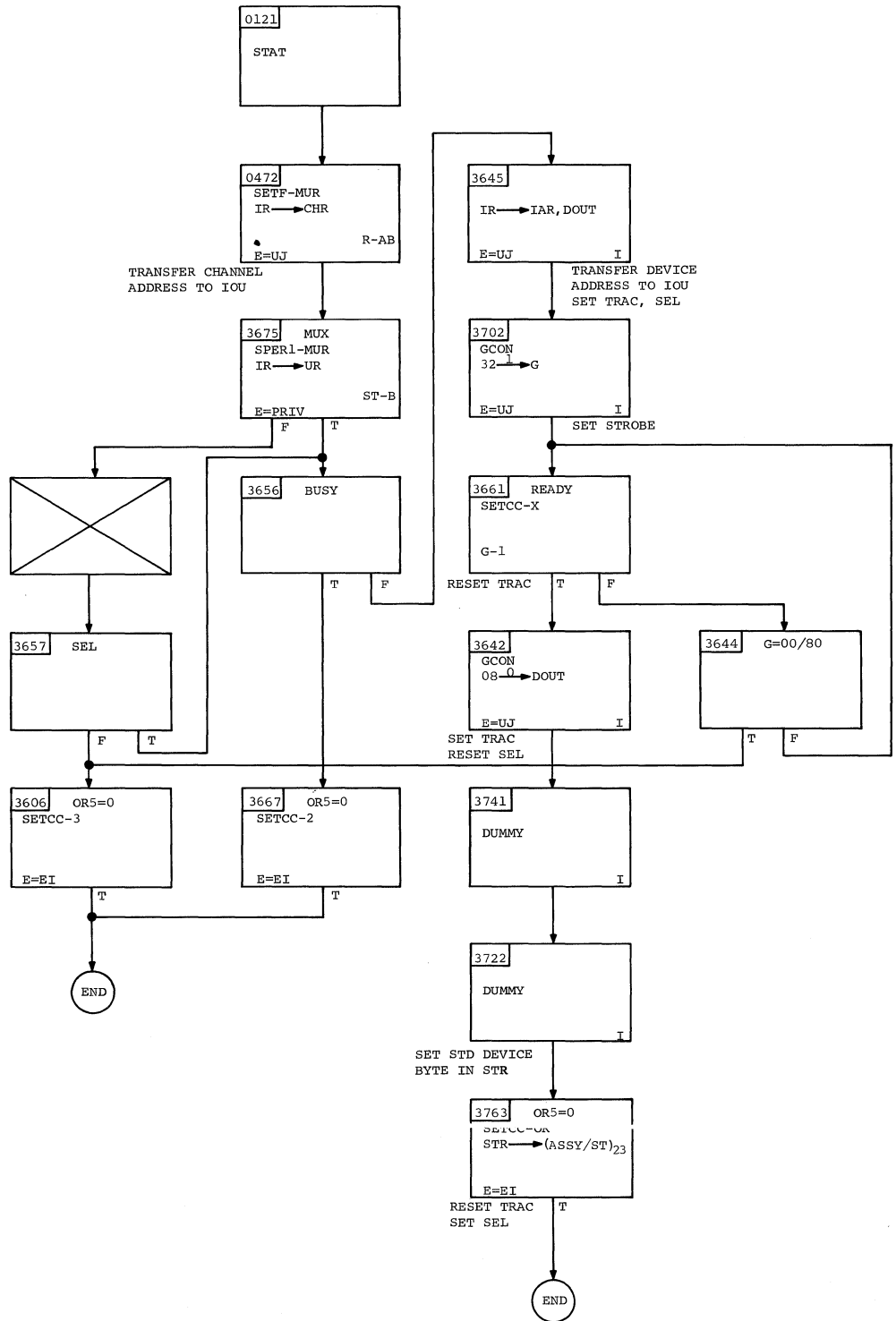


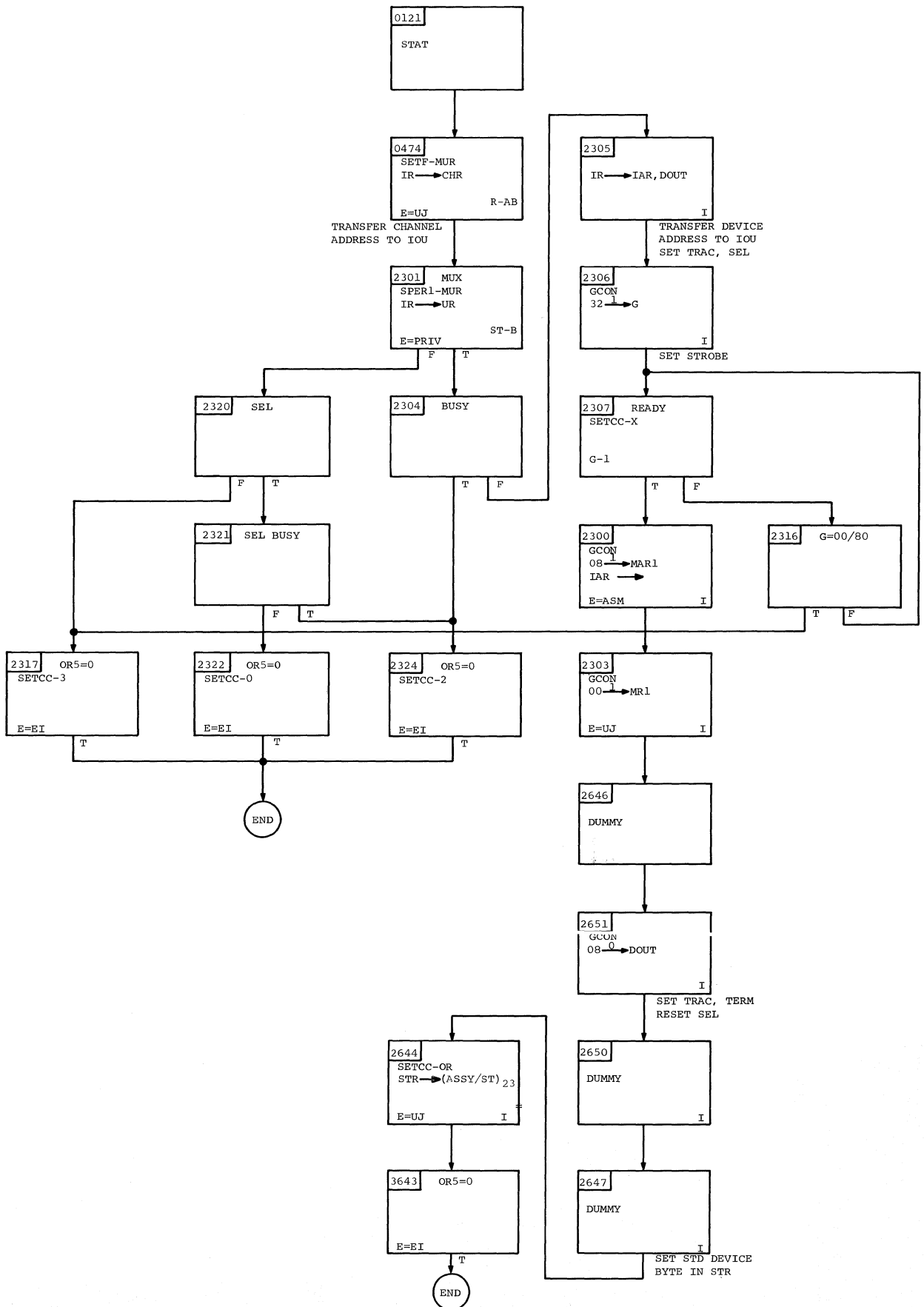




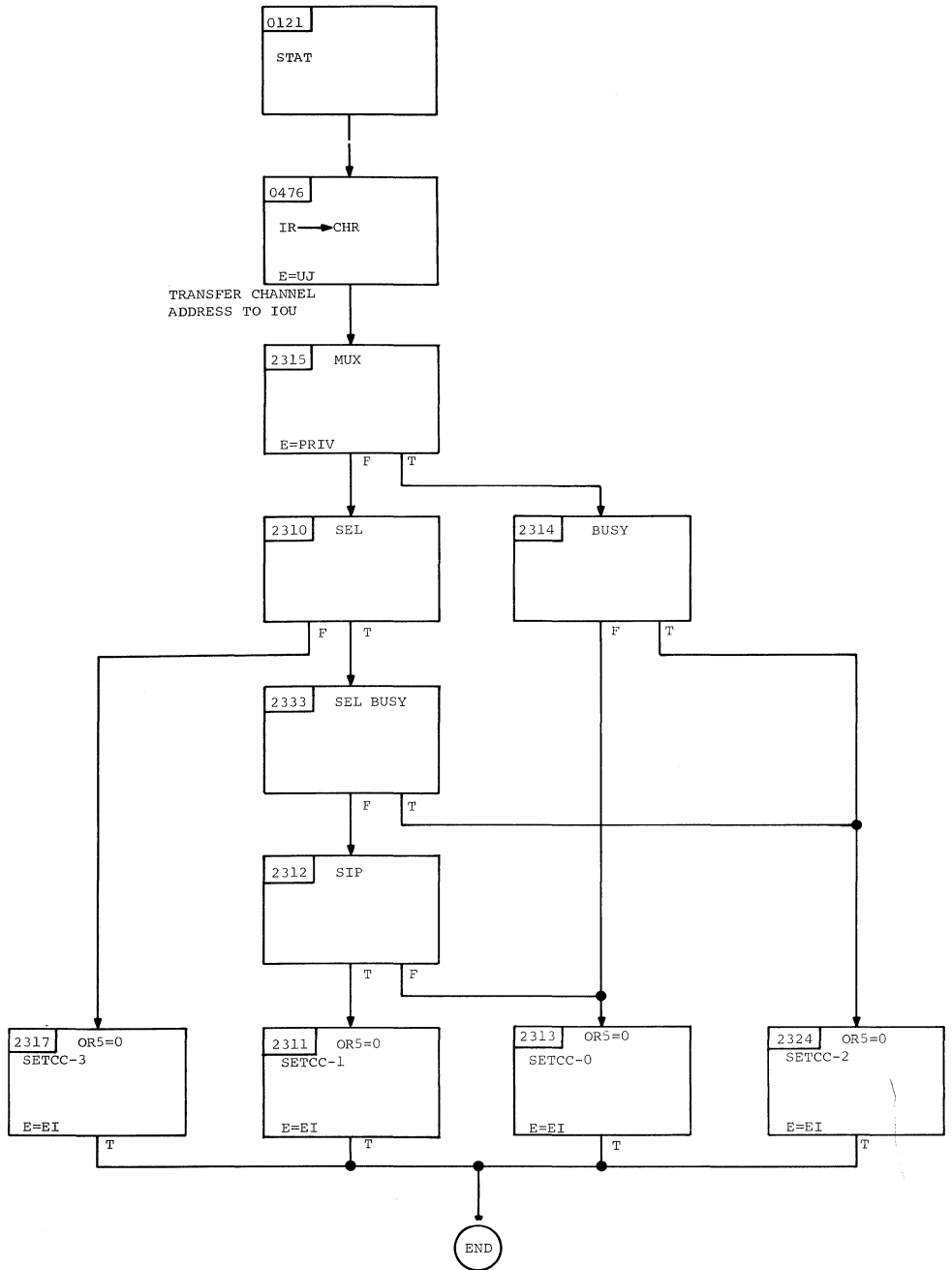


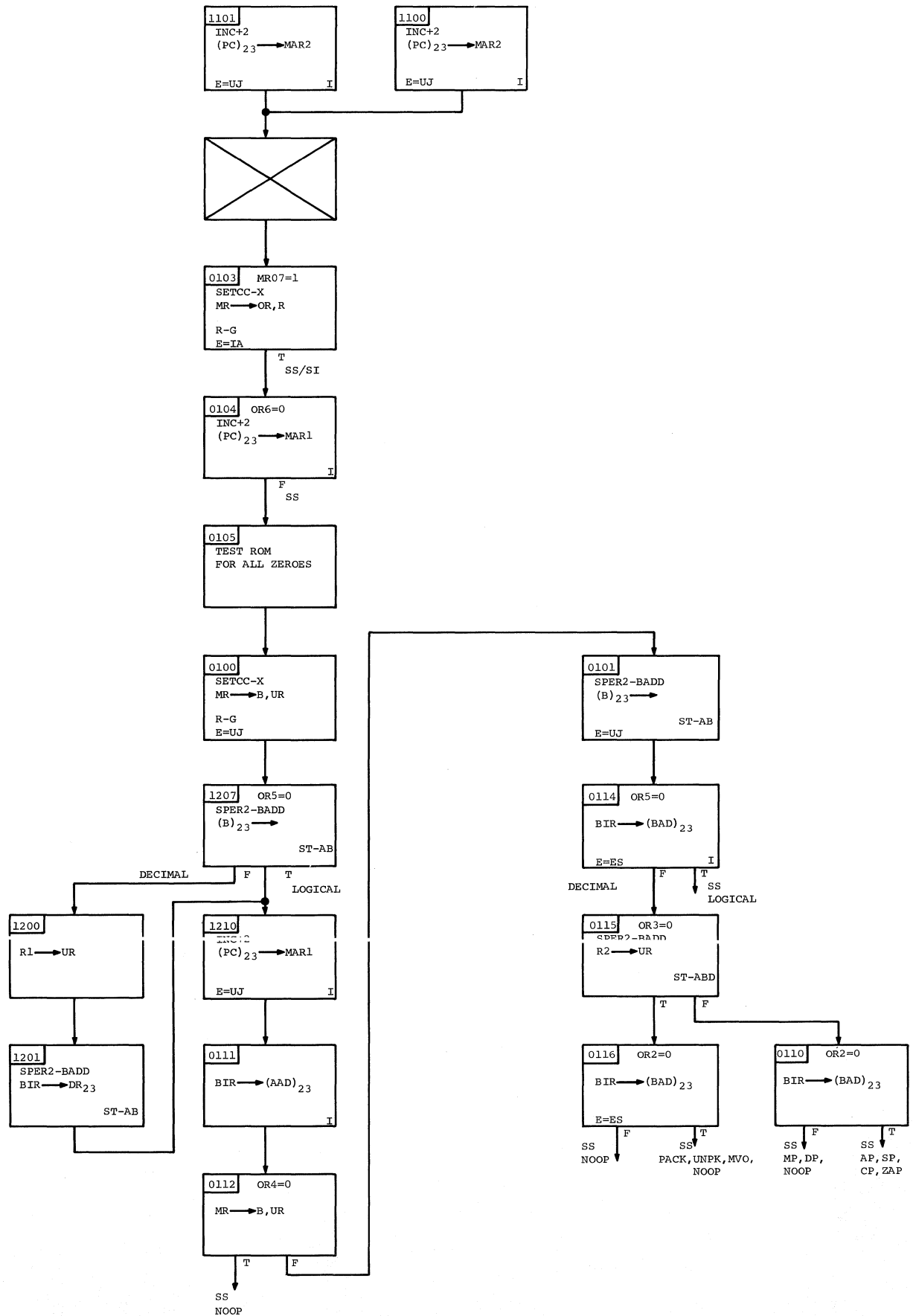




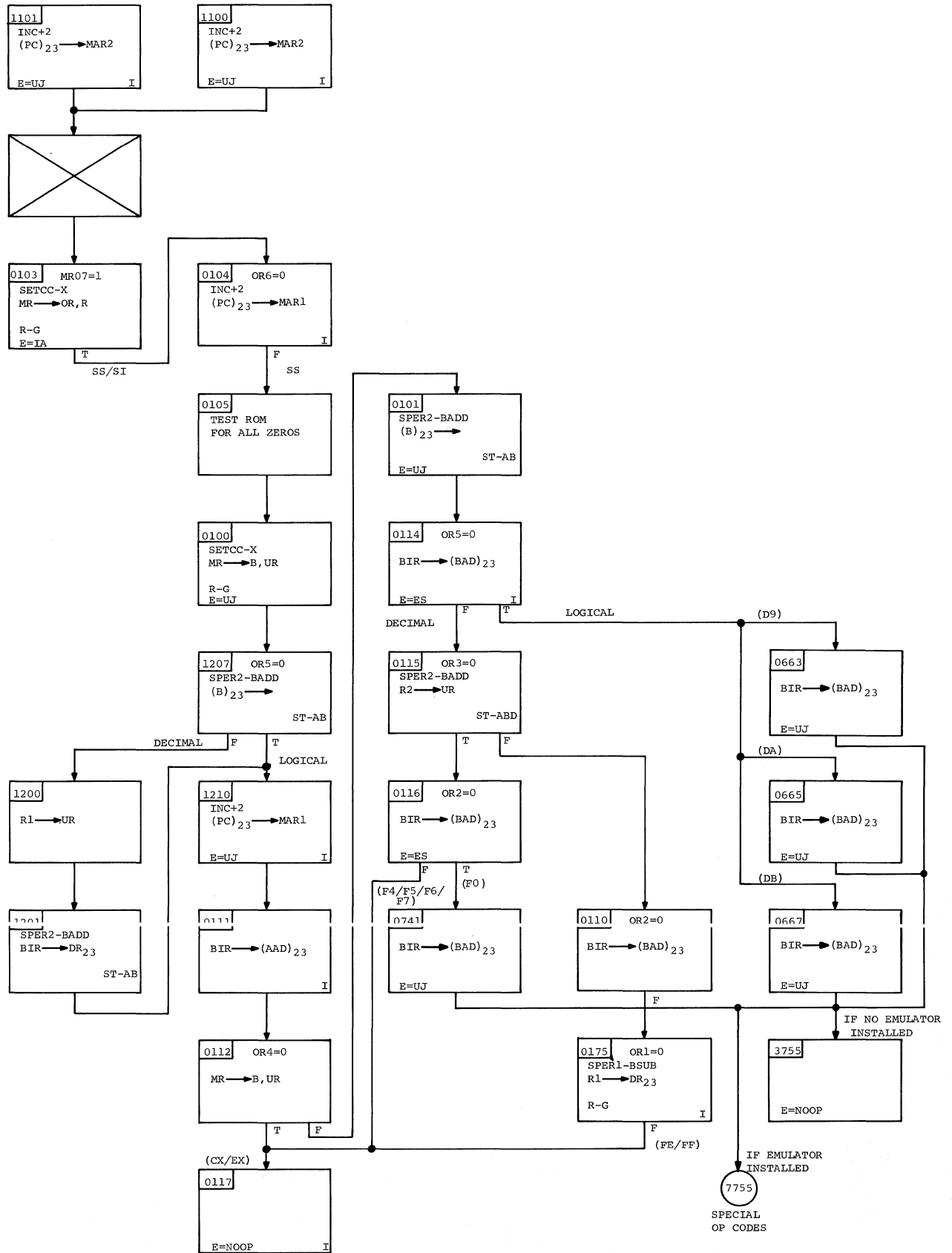


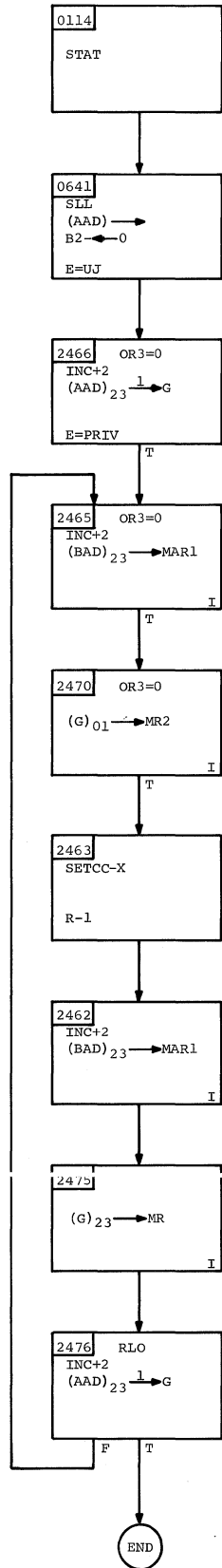


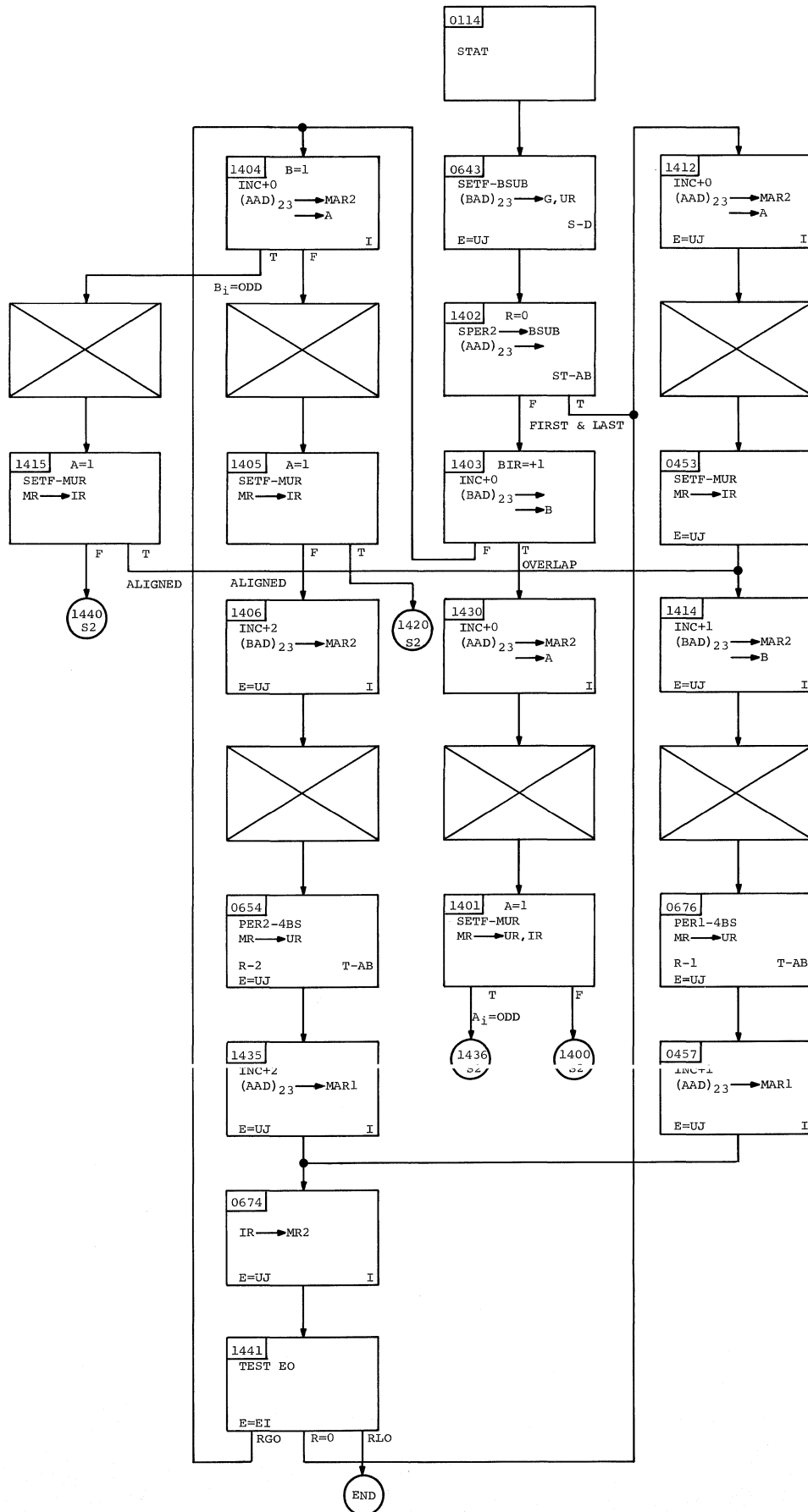




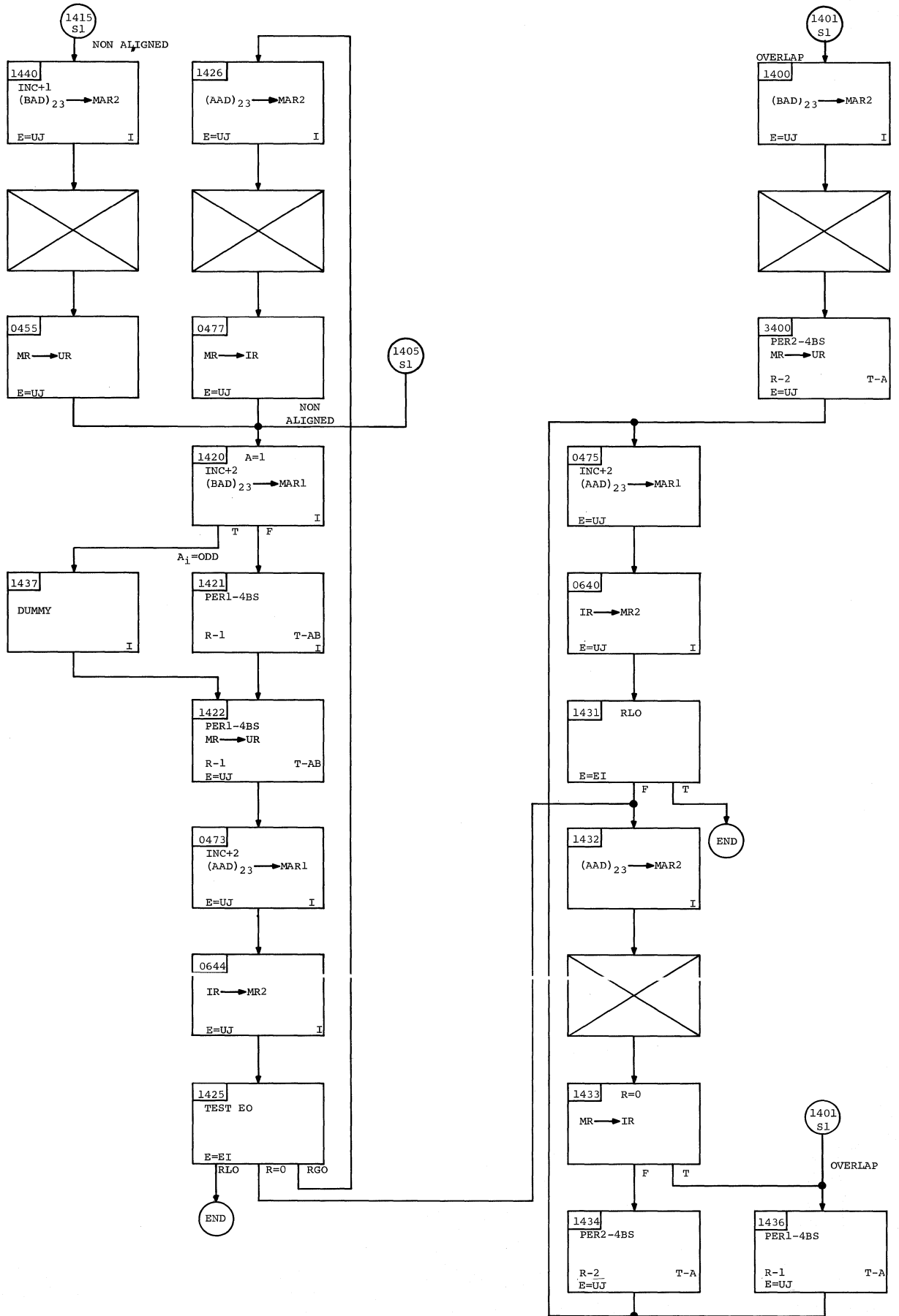
NO OPERATION INSTRUCTIONS SS FORMAT

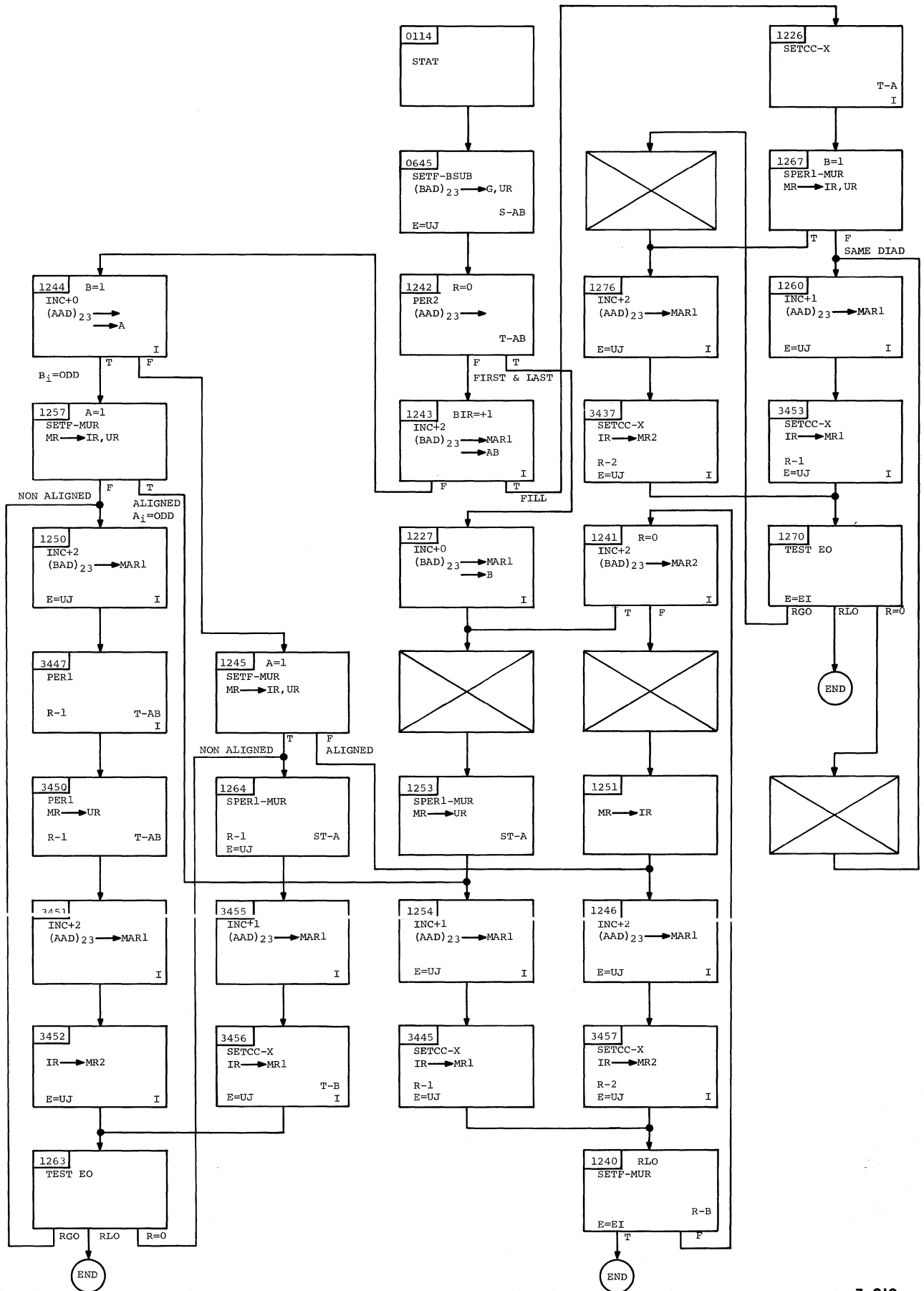


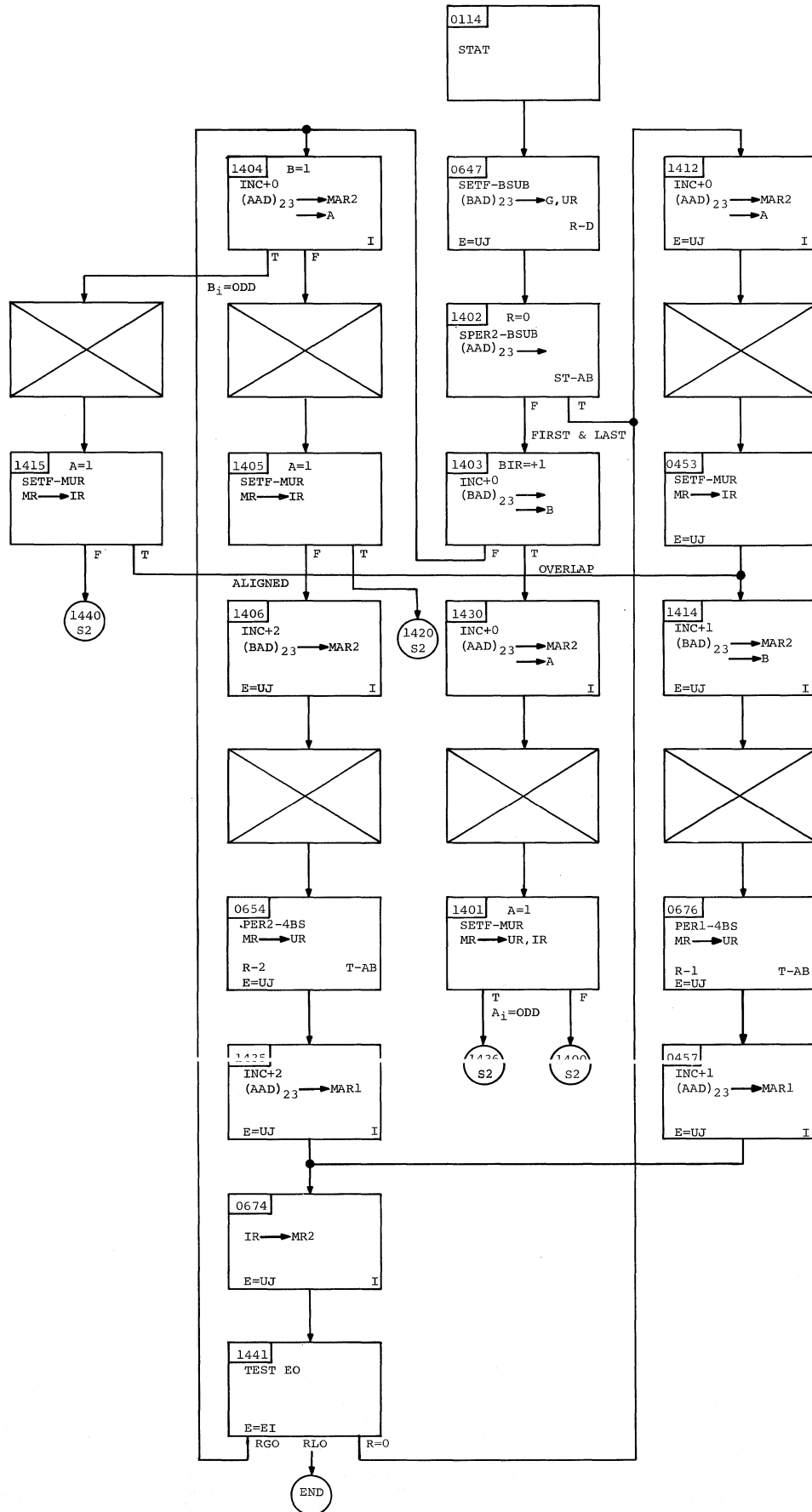




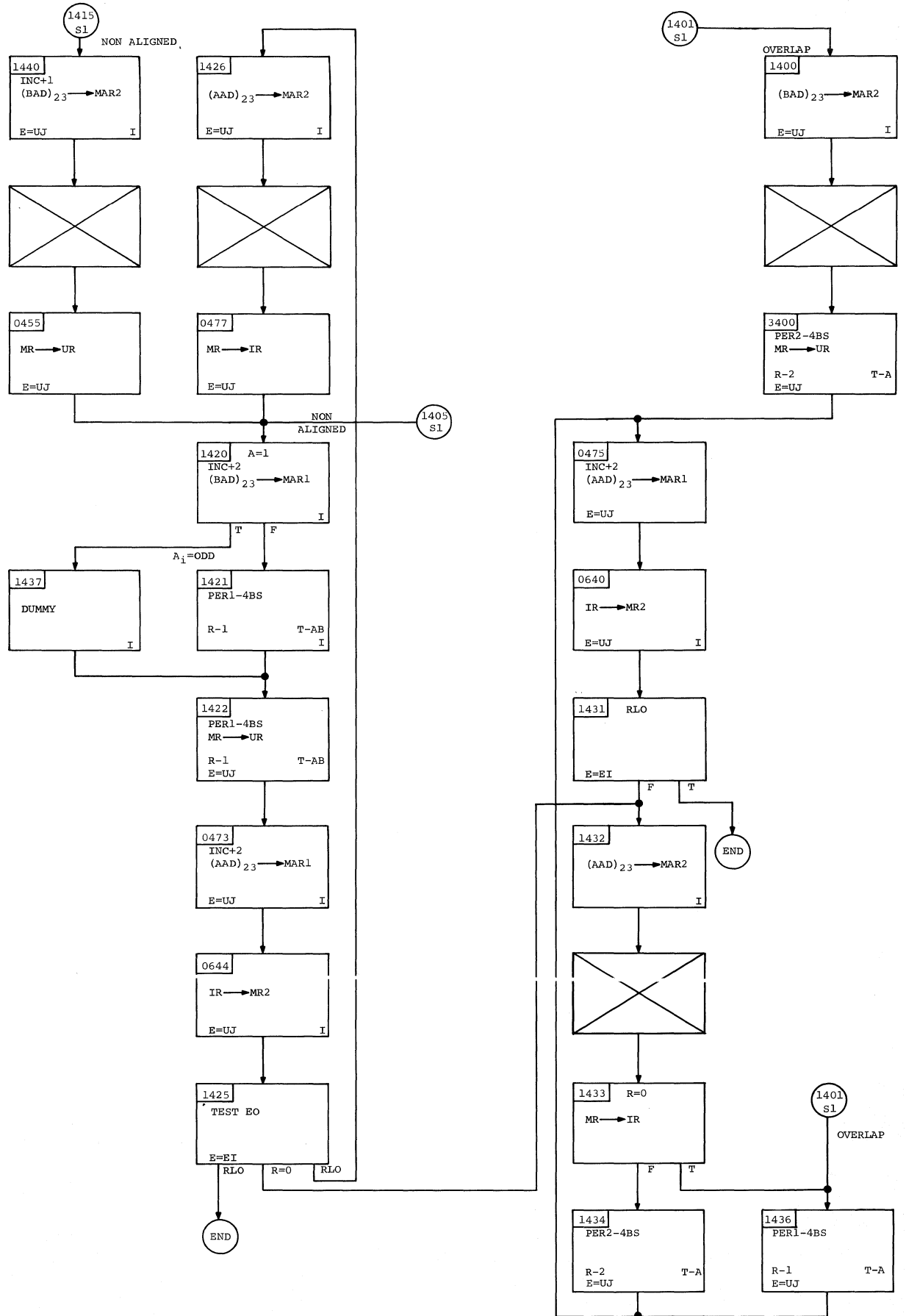
MOVE NUMERICS MVN SS DI

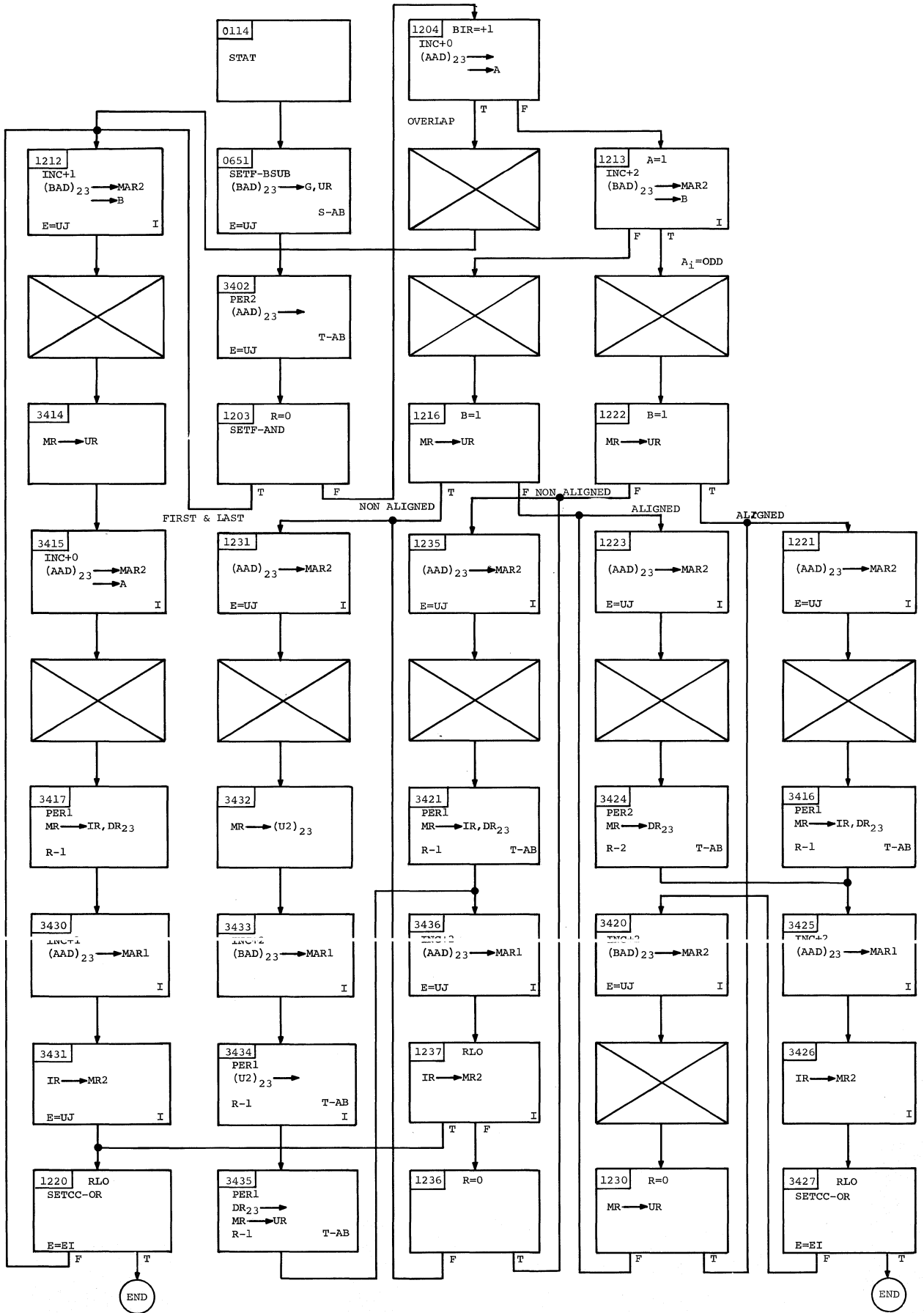


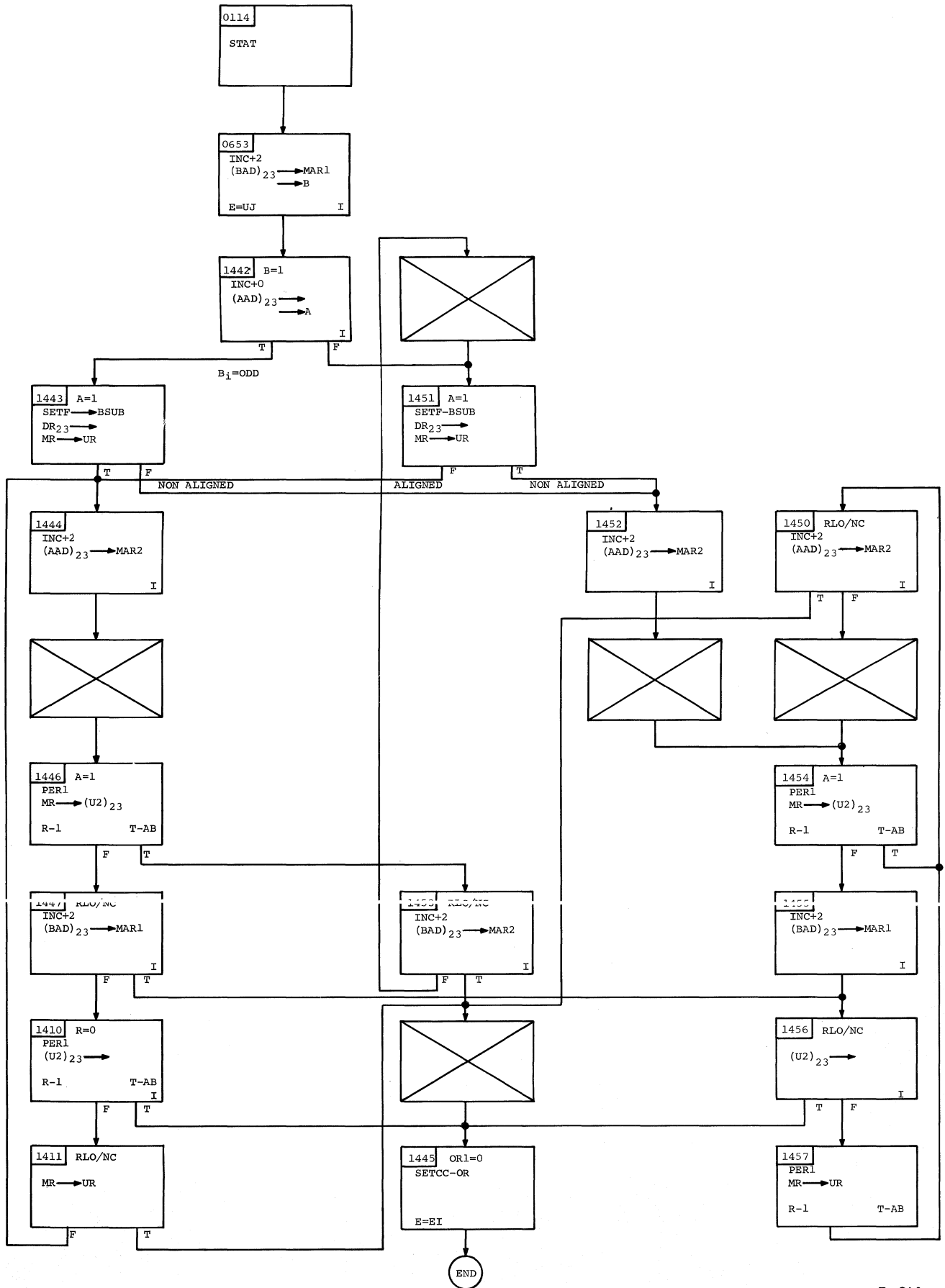


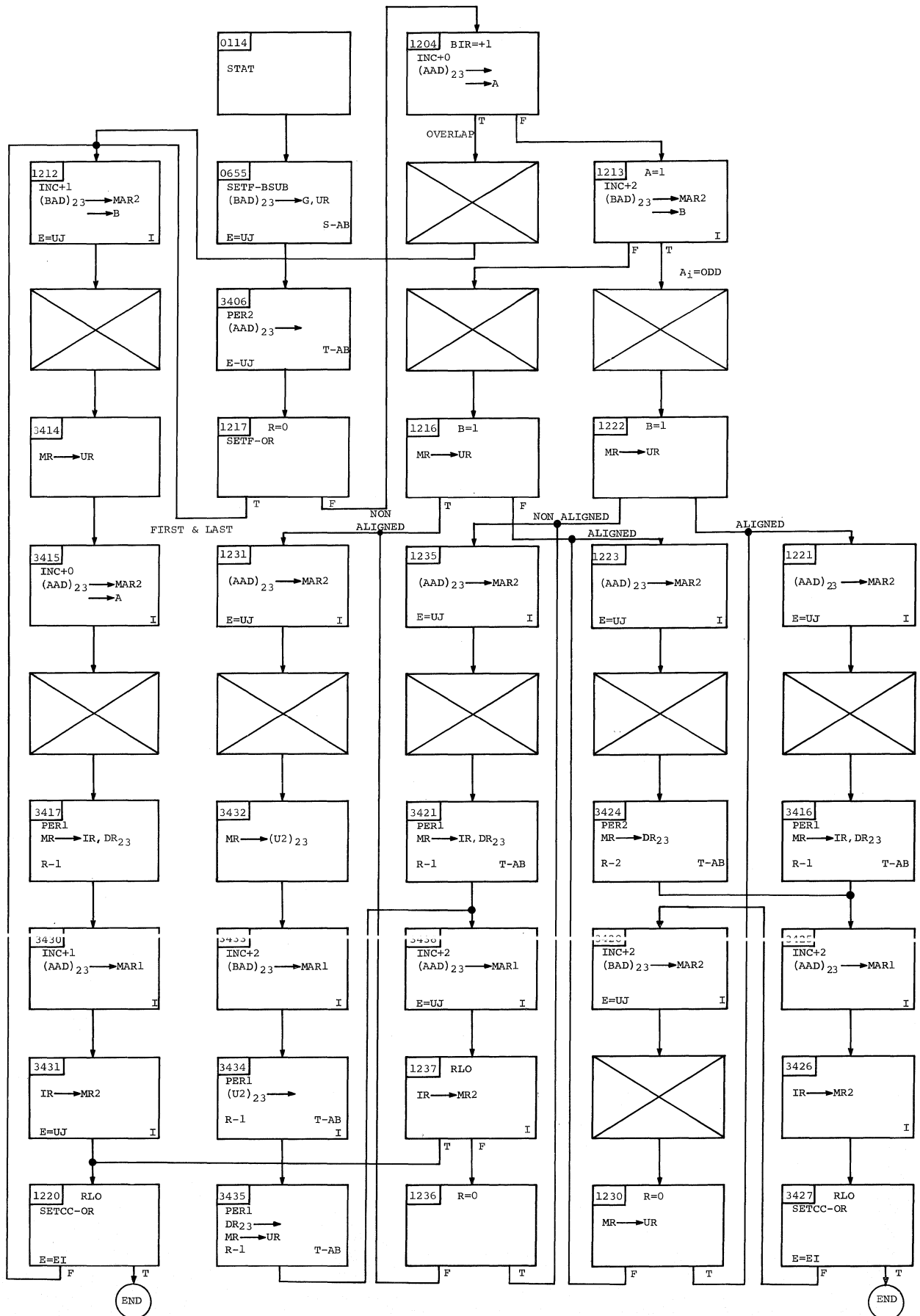


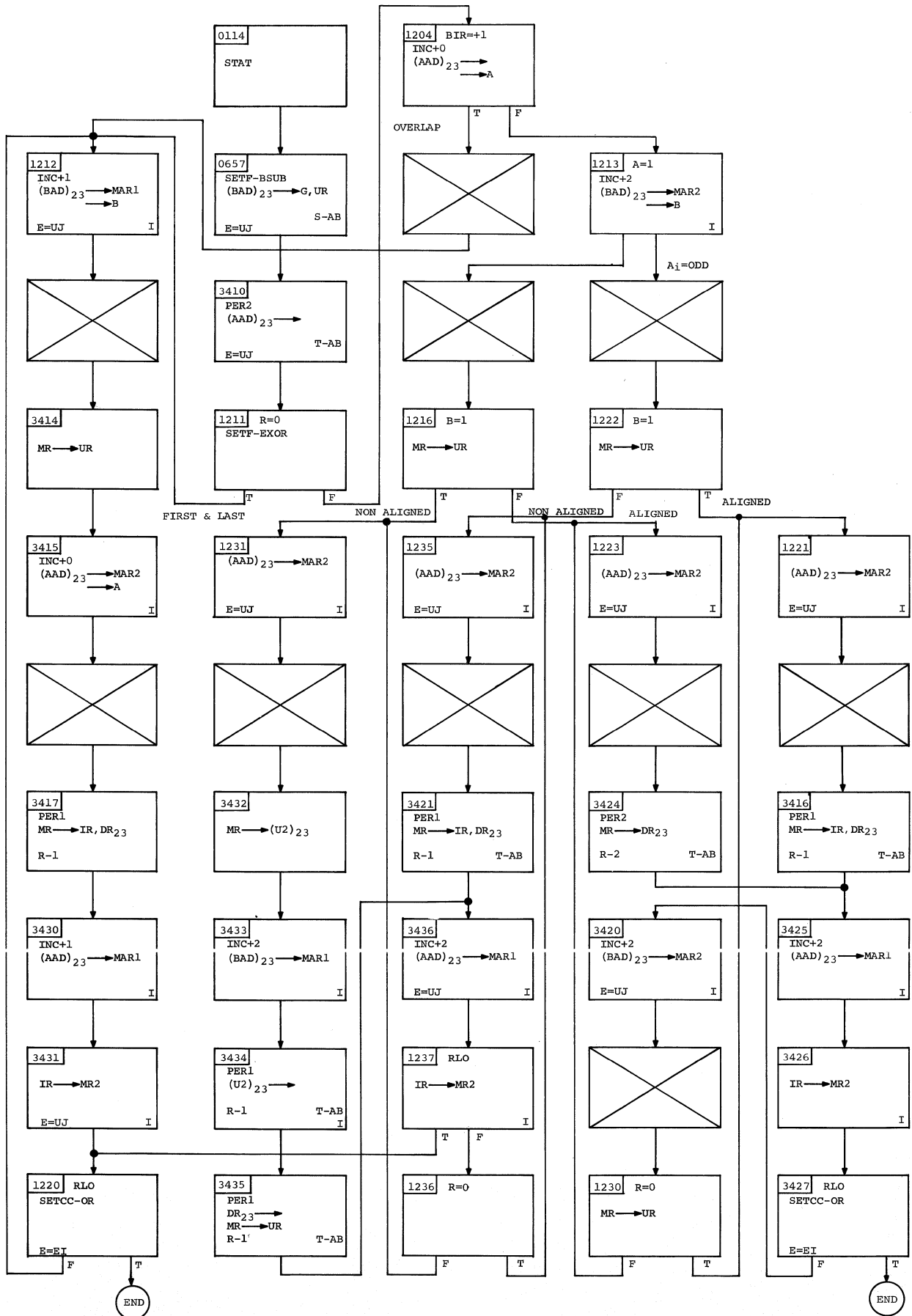


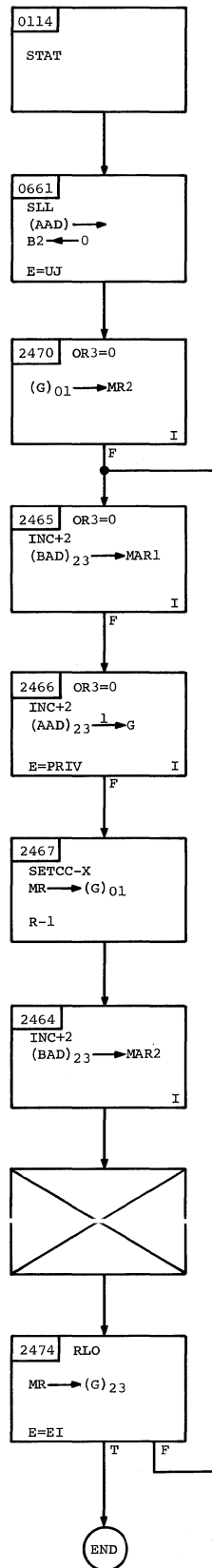


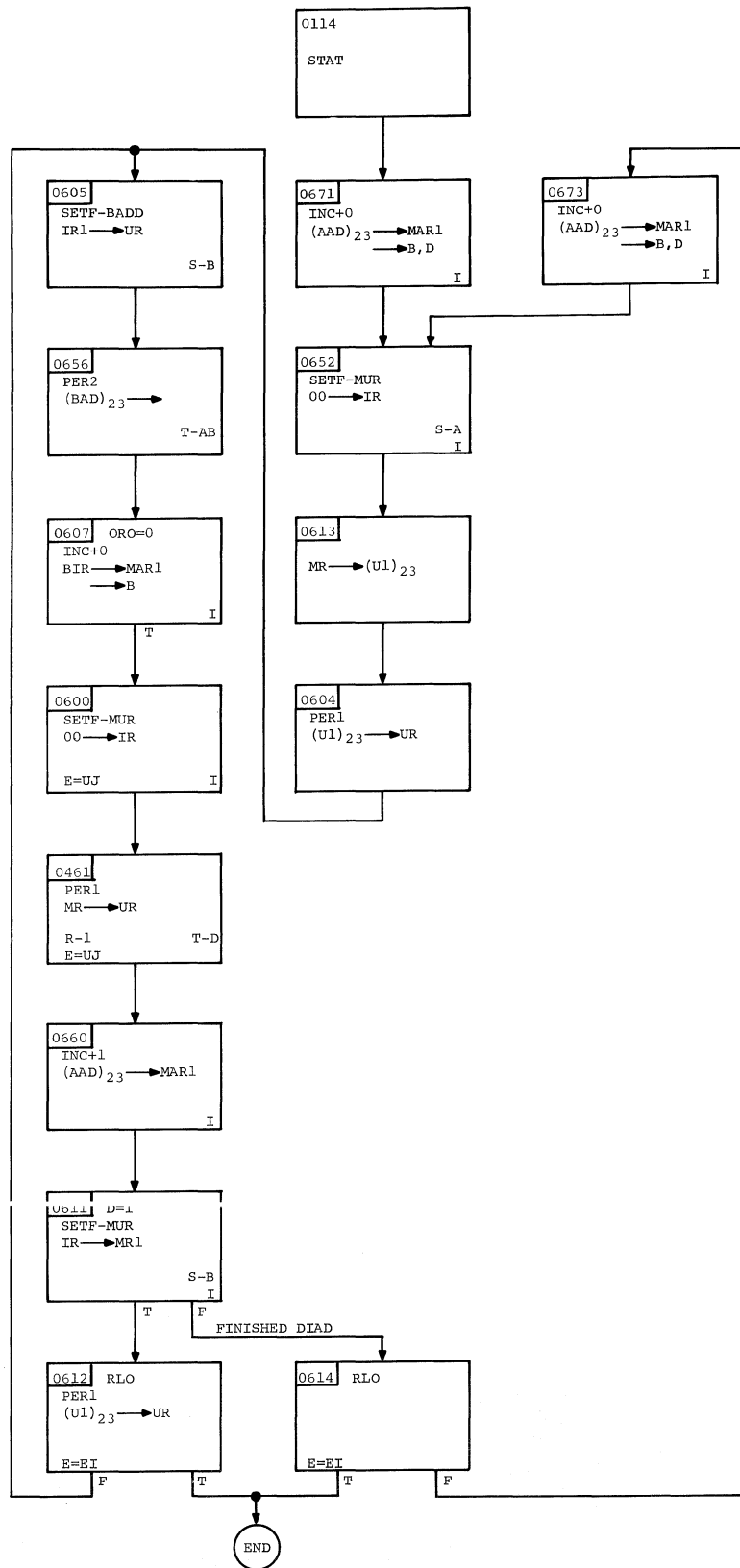


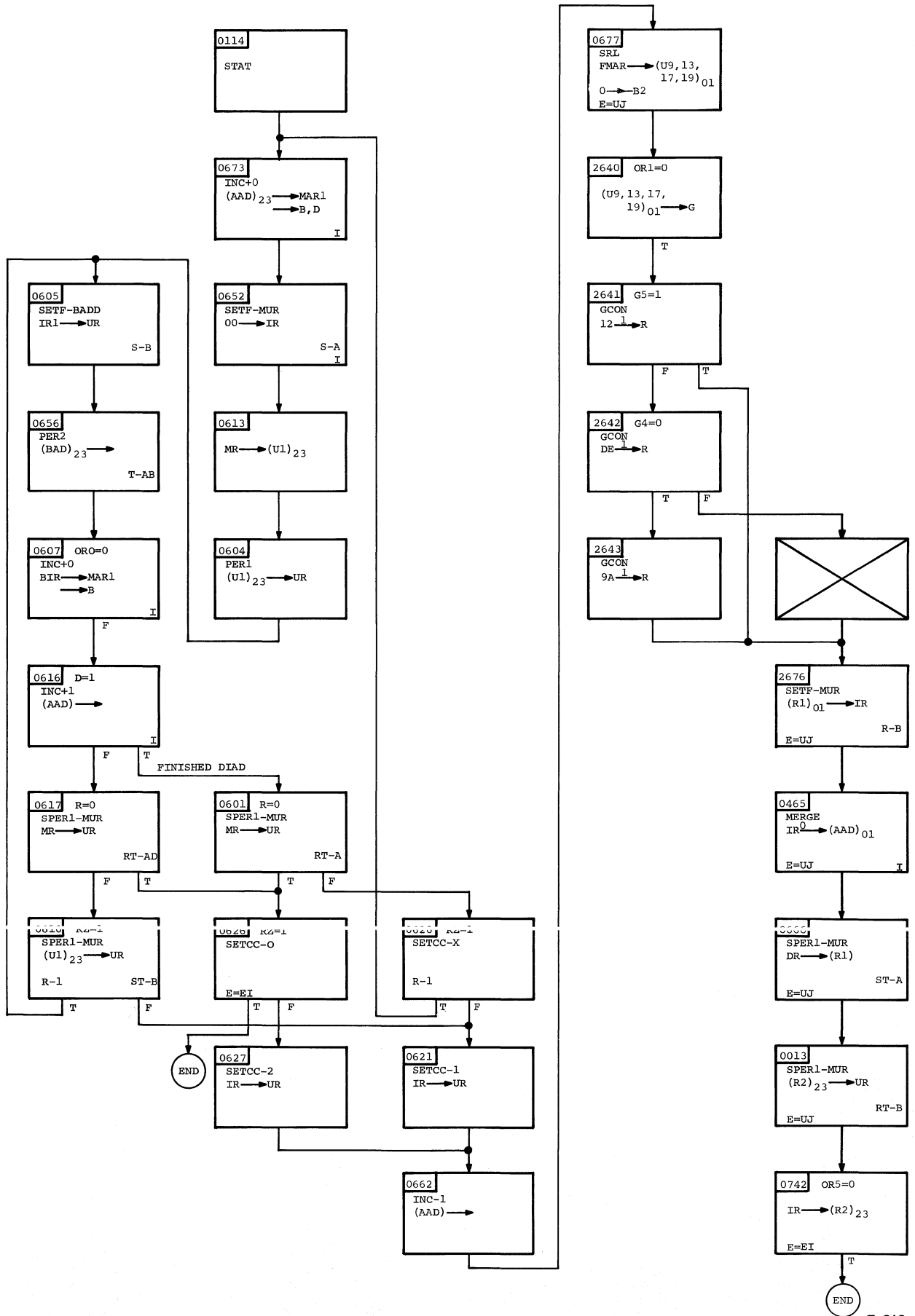




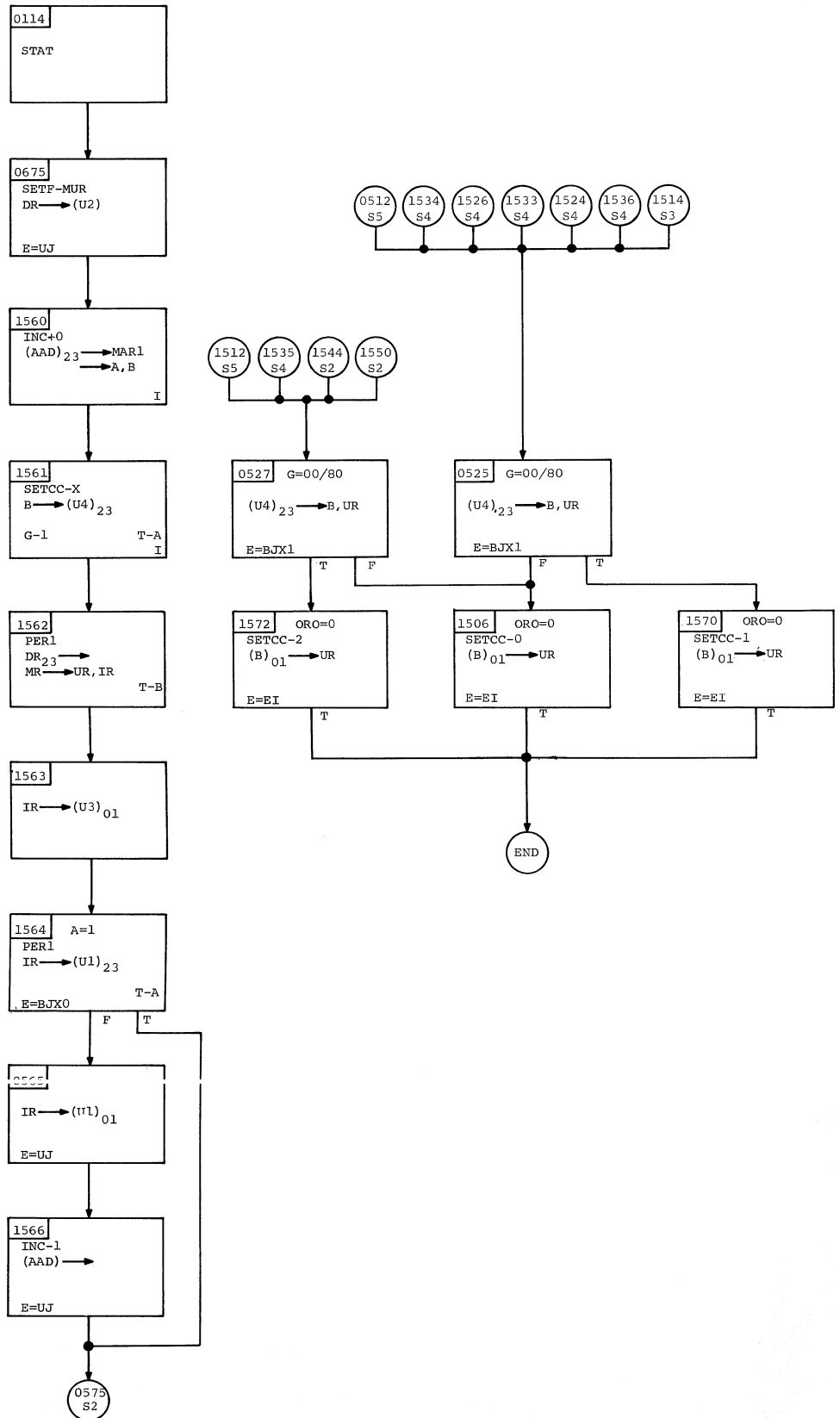


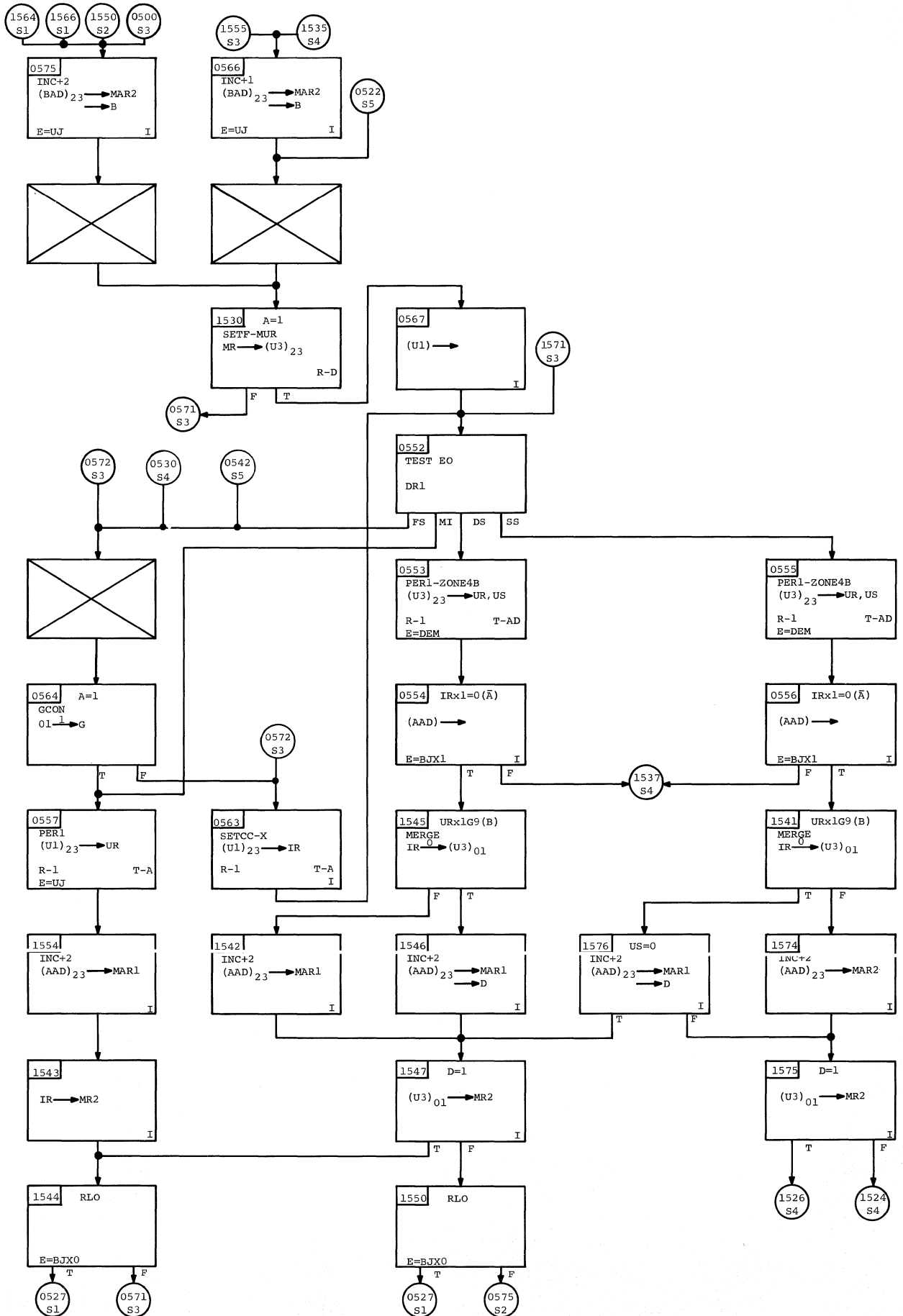


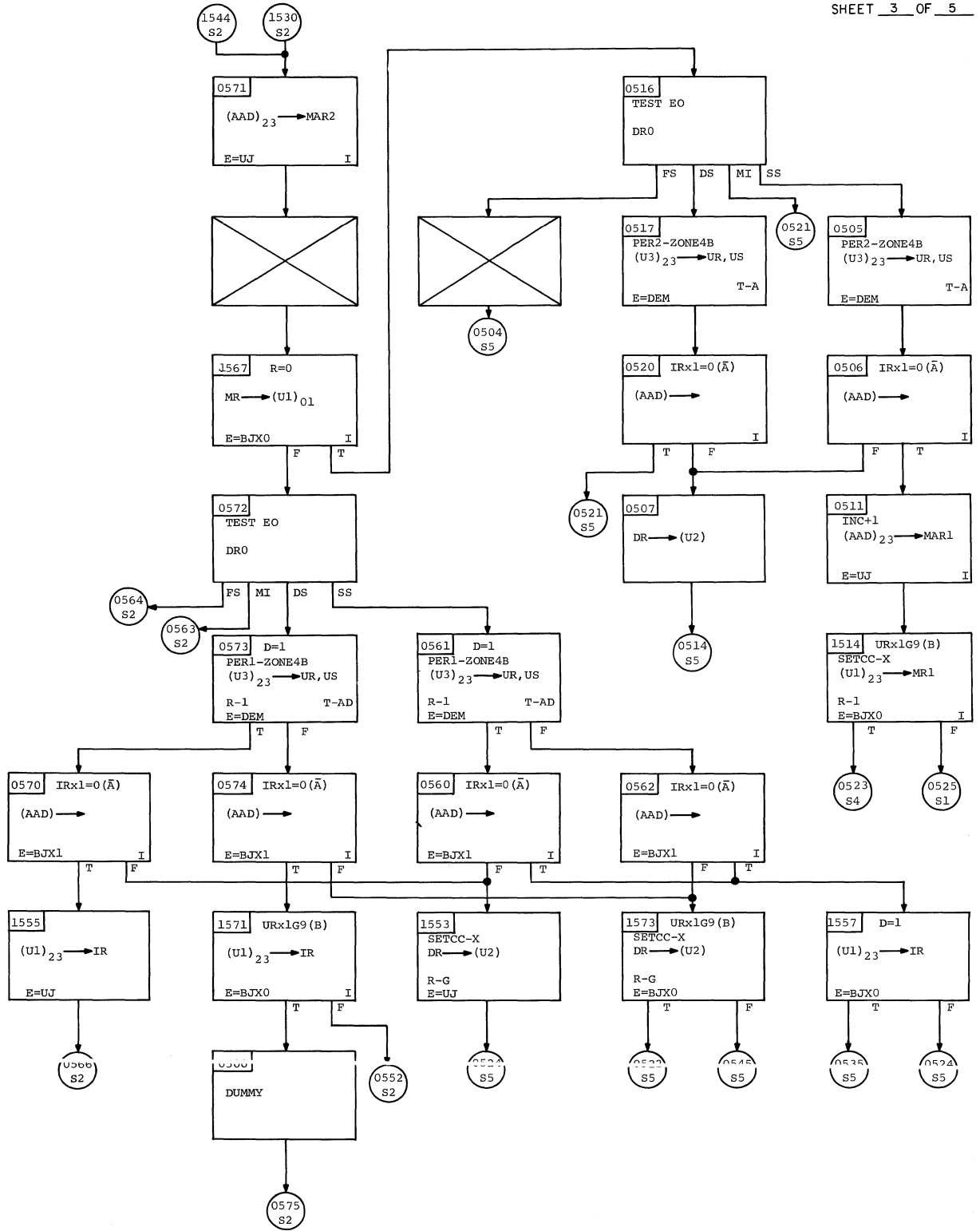


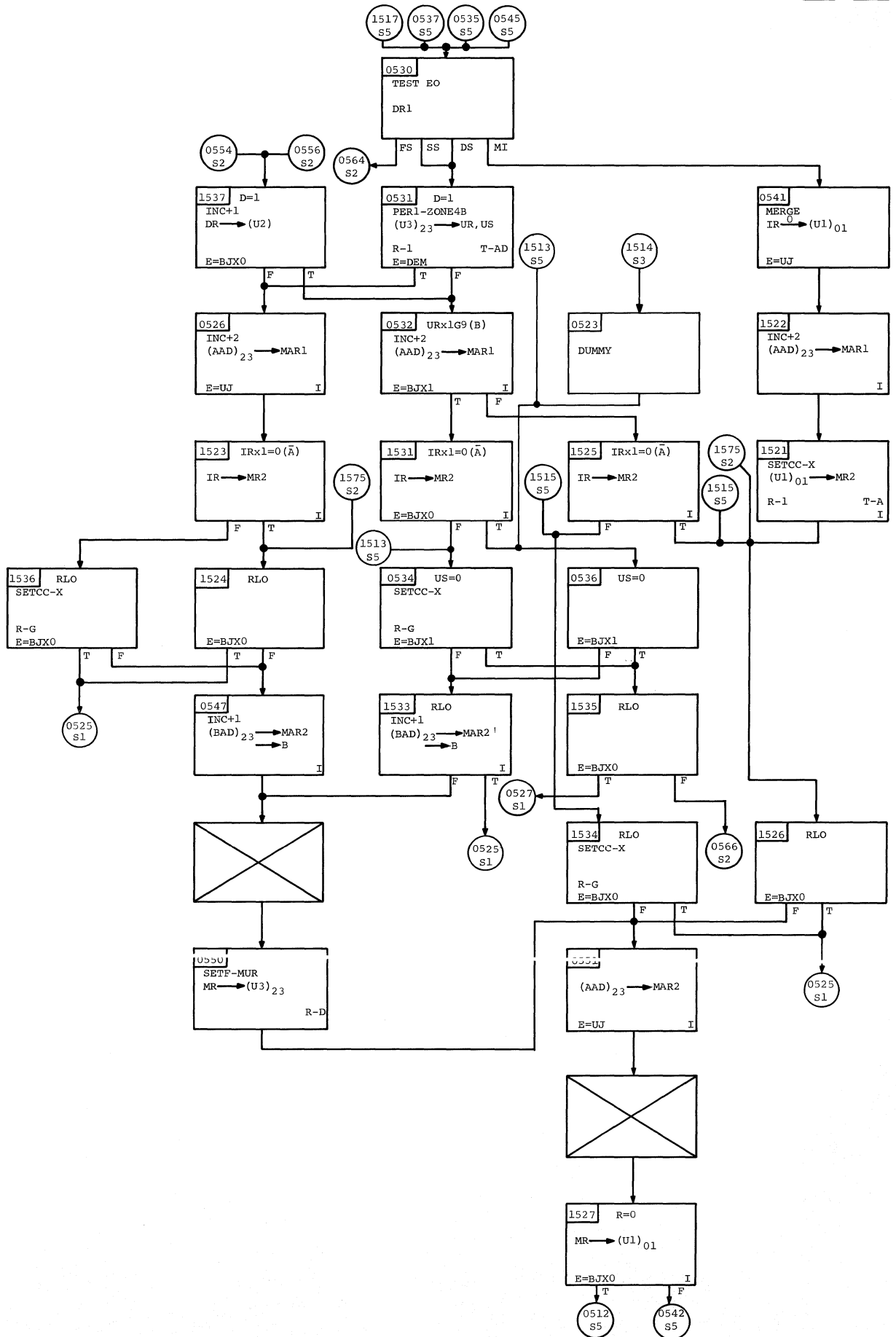


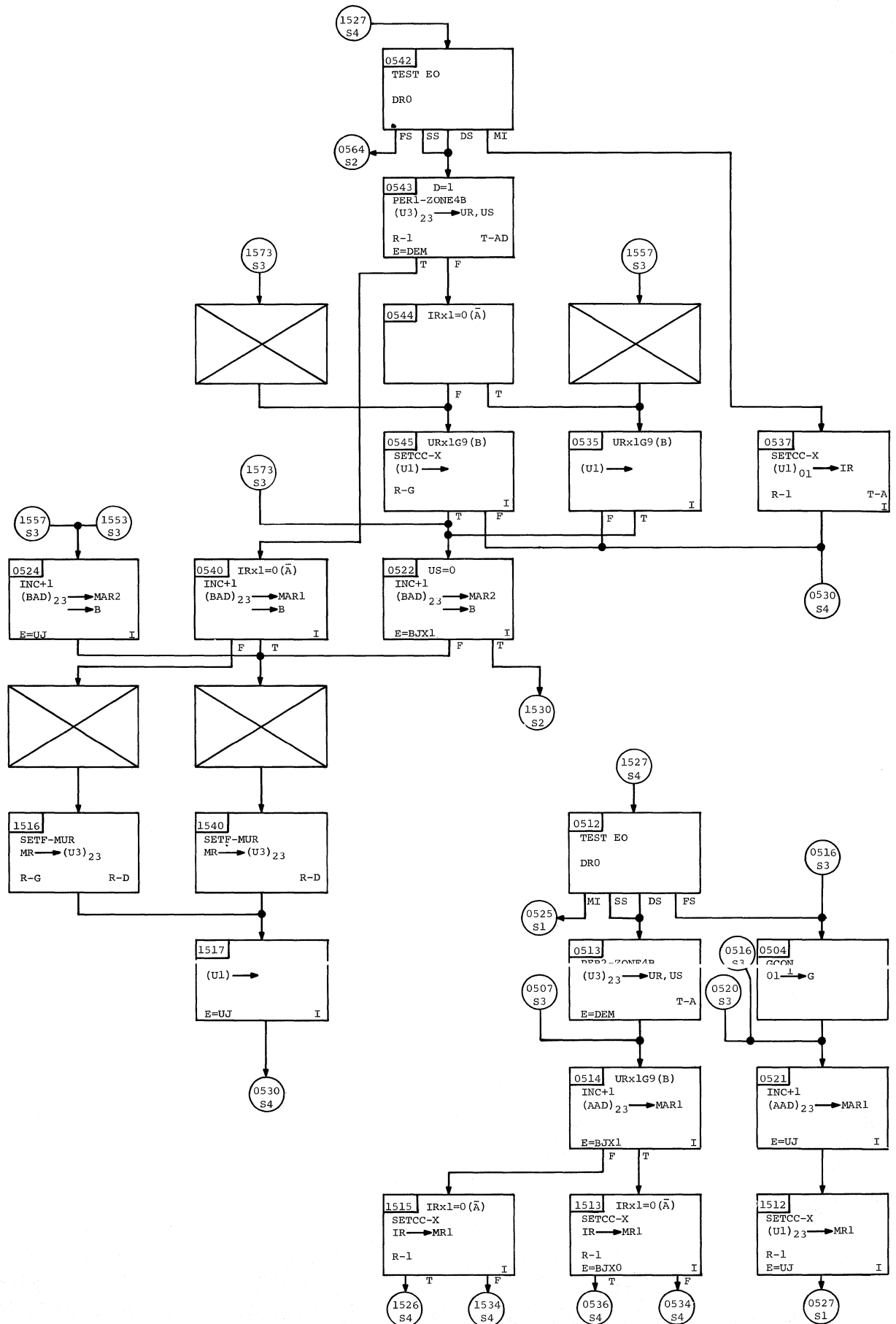


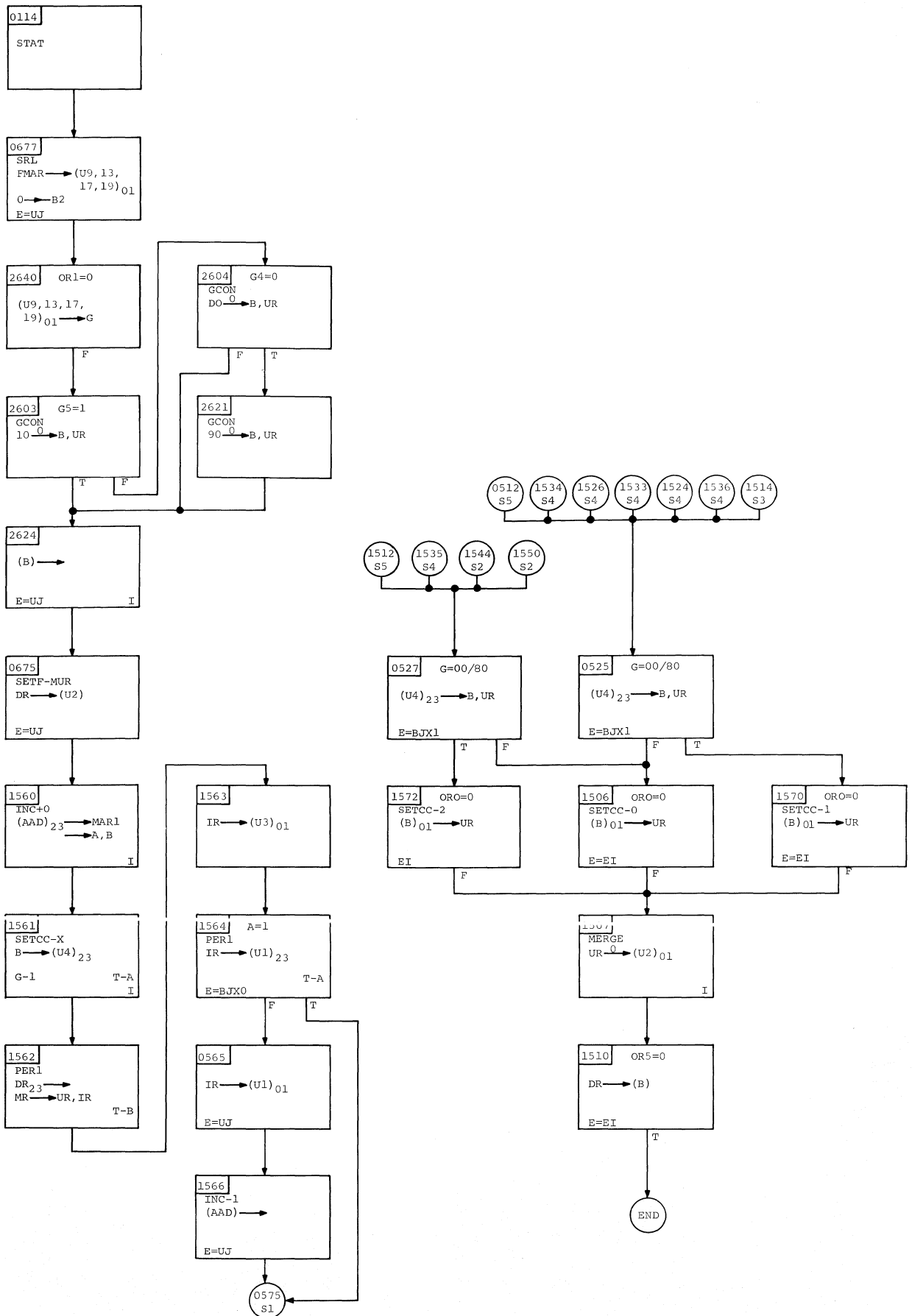


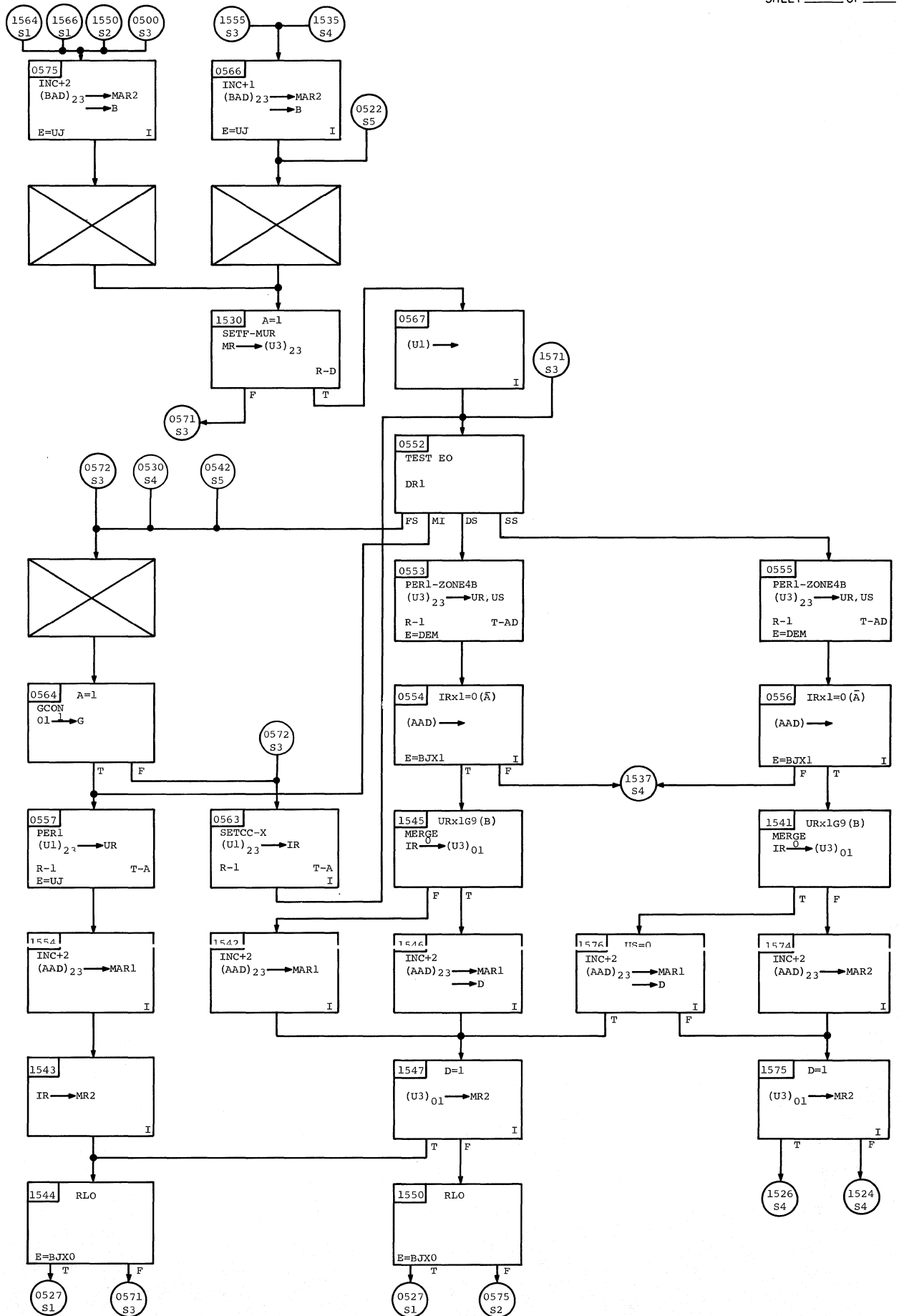


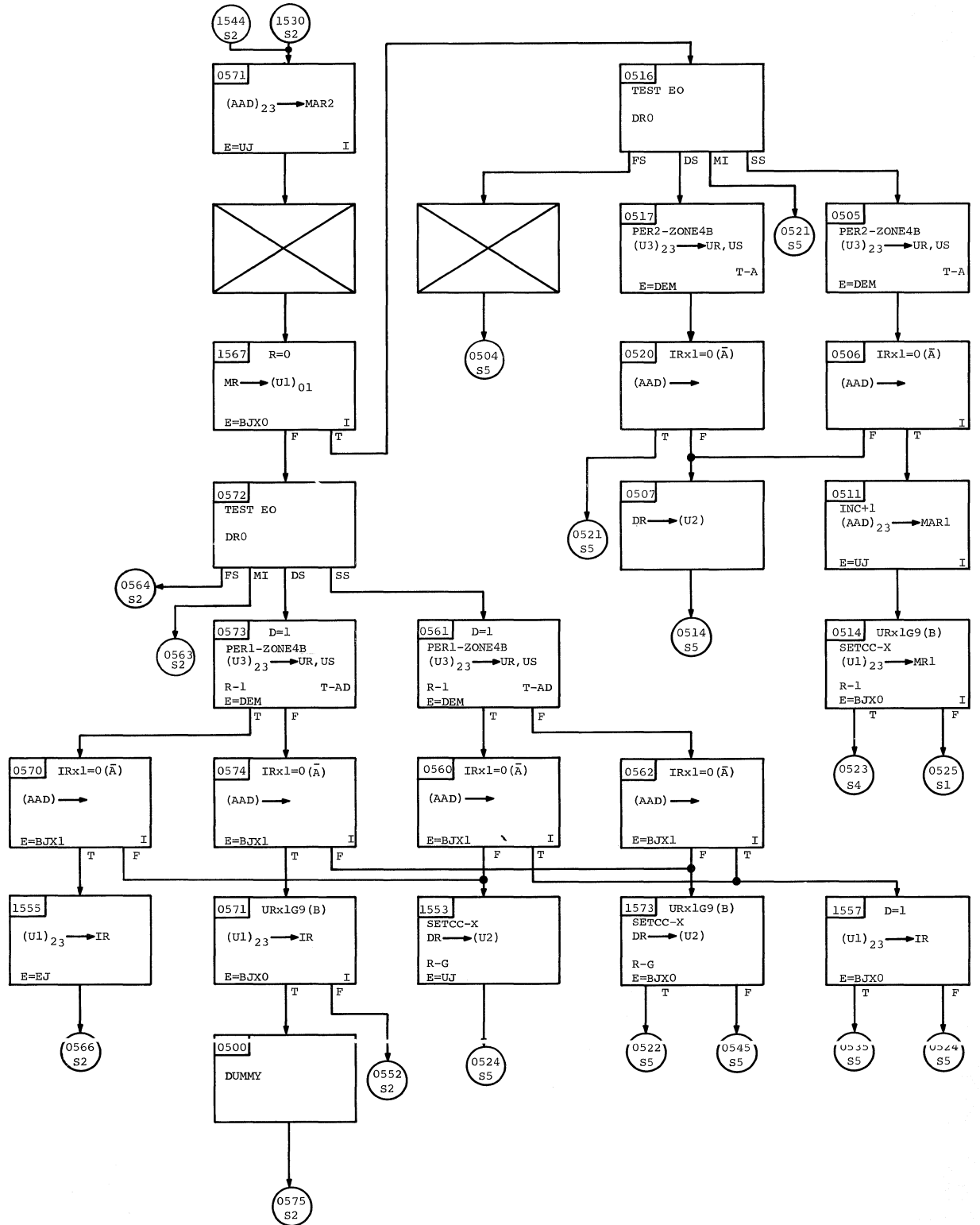




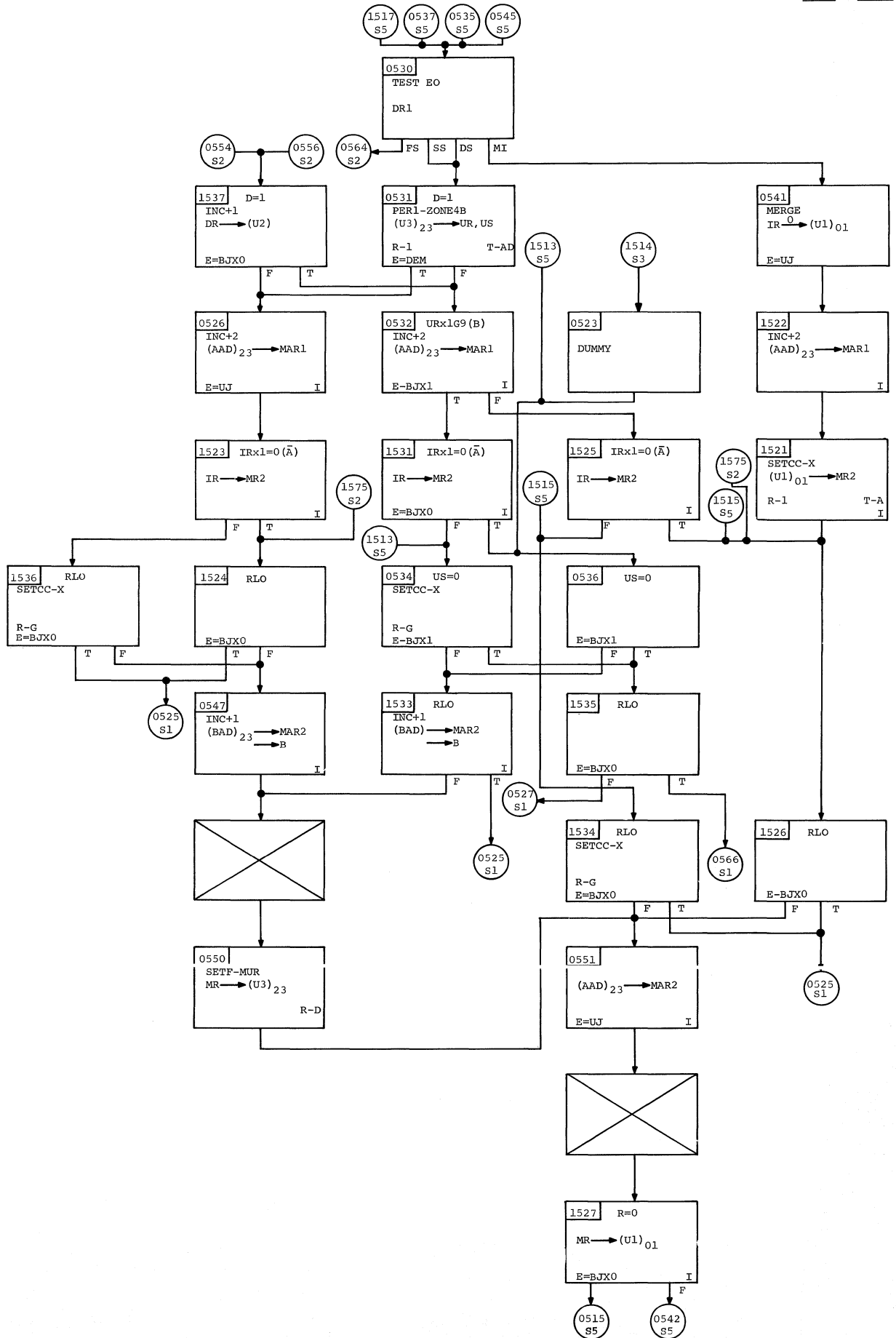


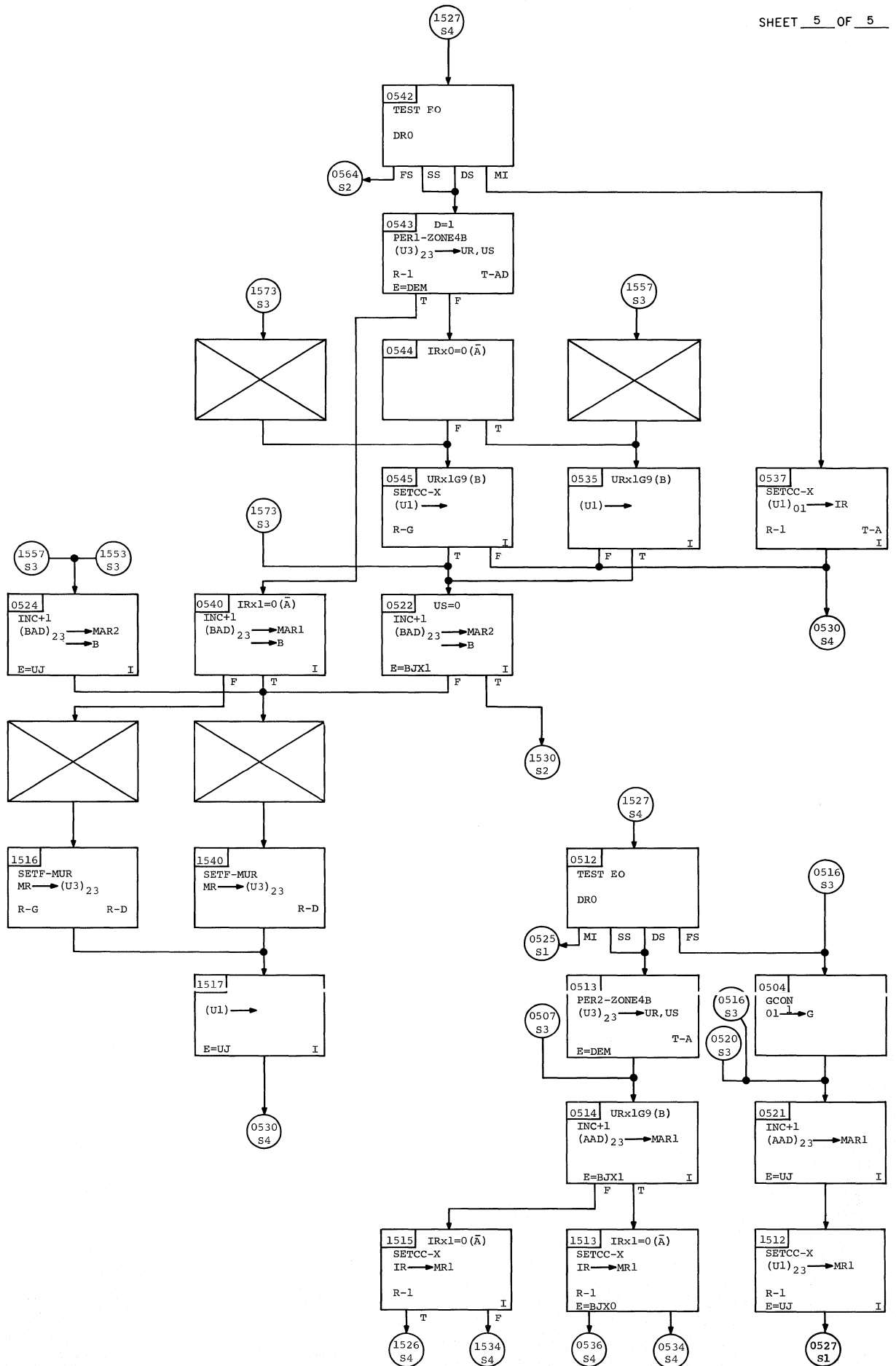




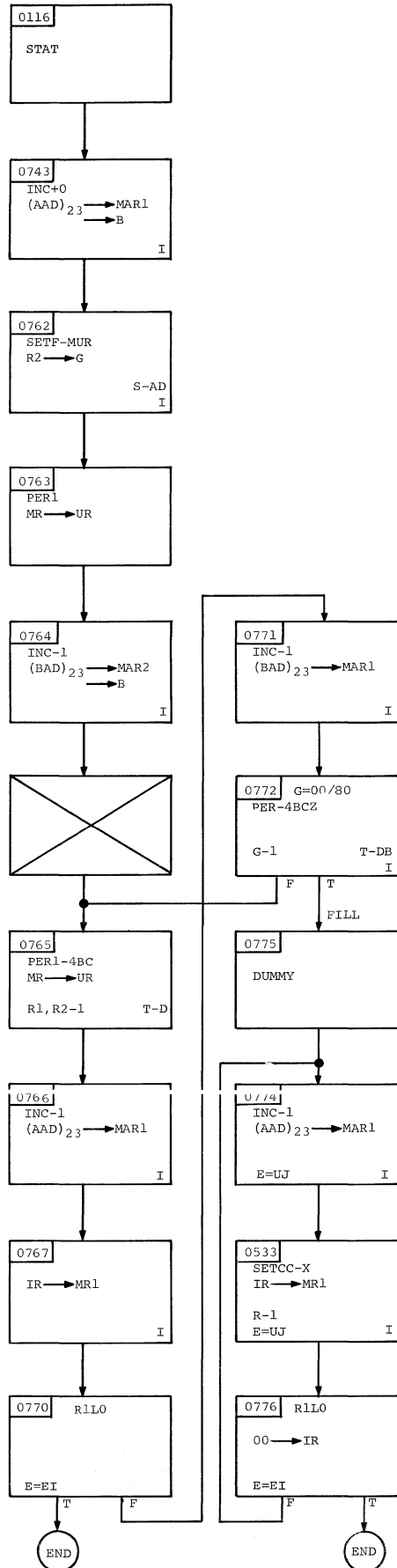


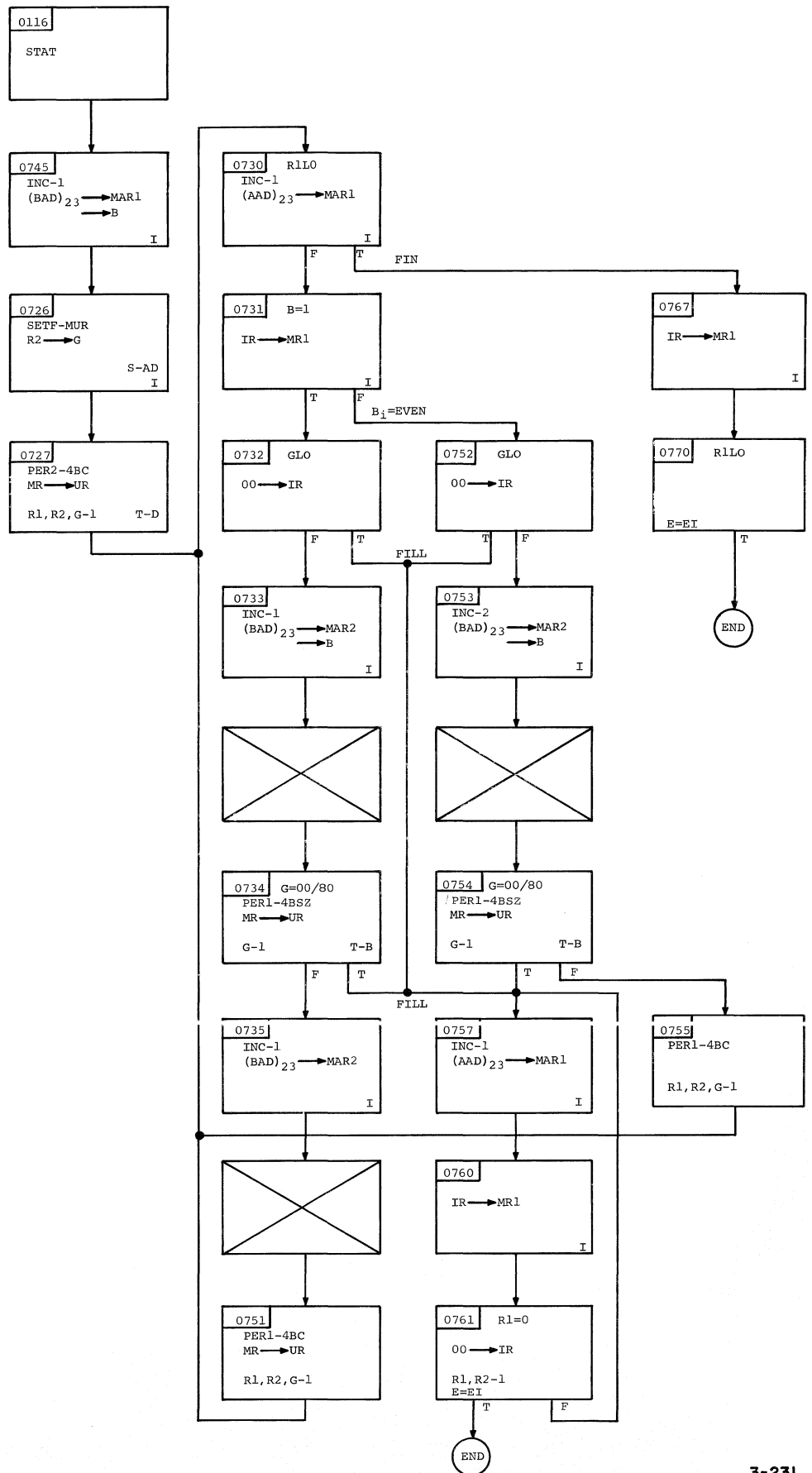


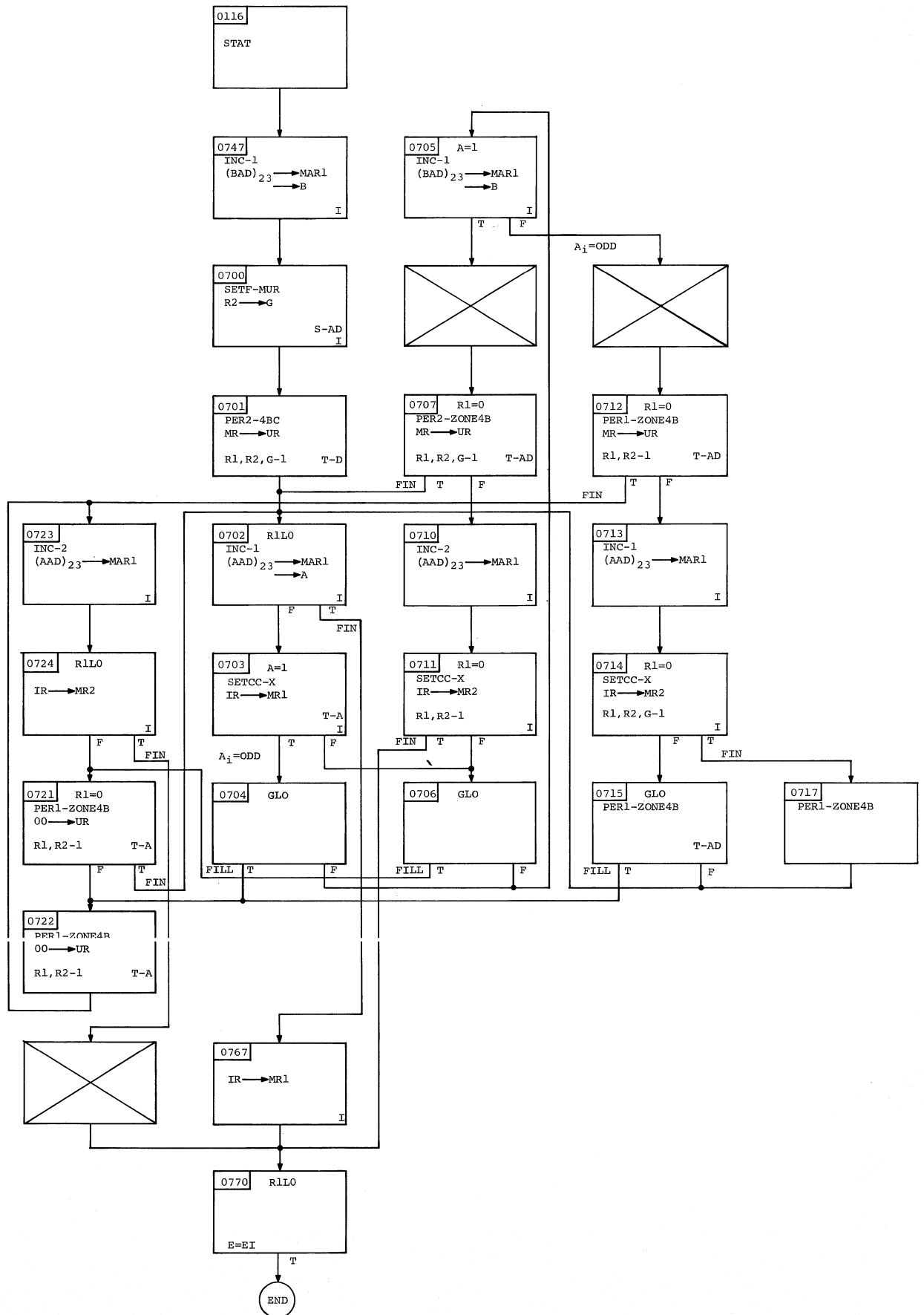


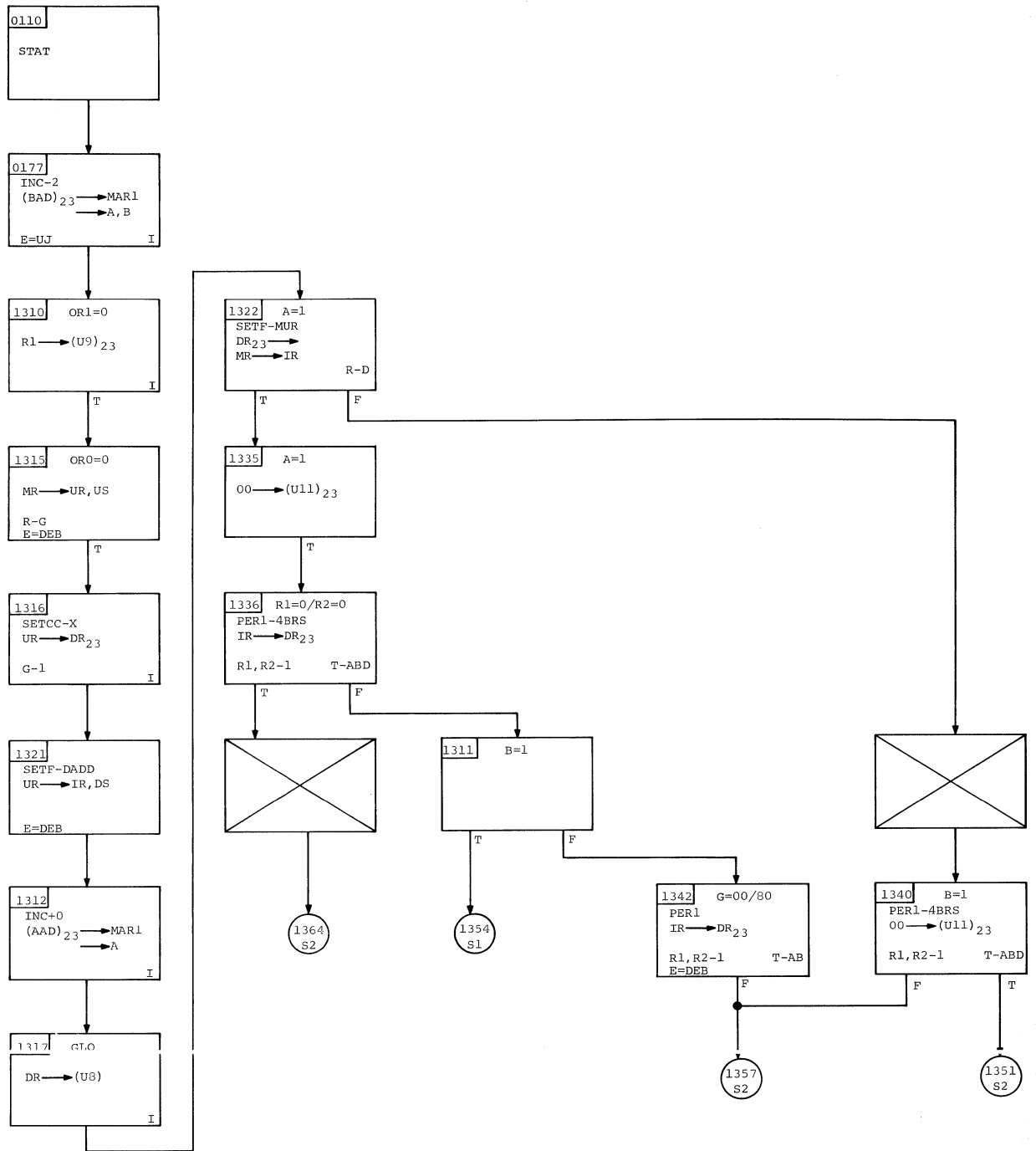


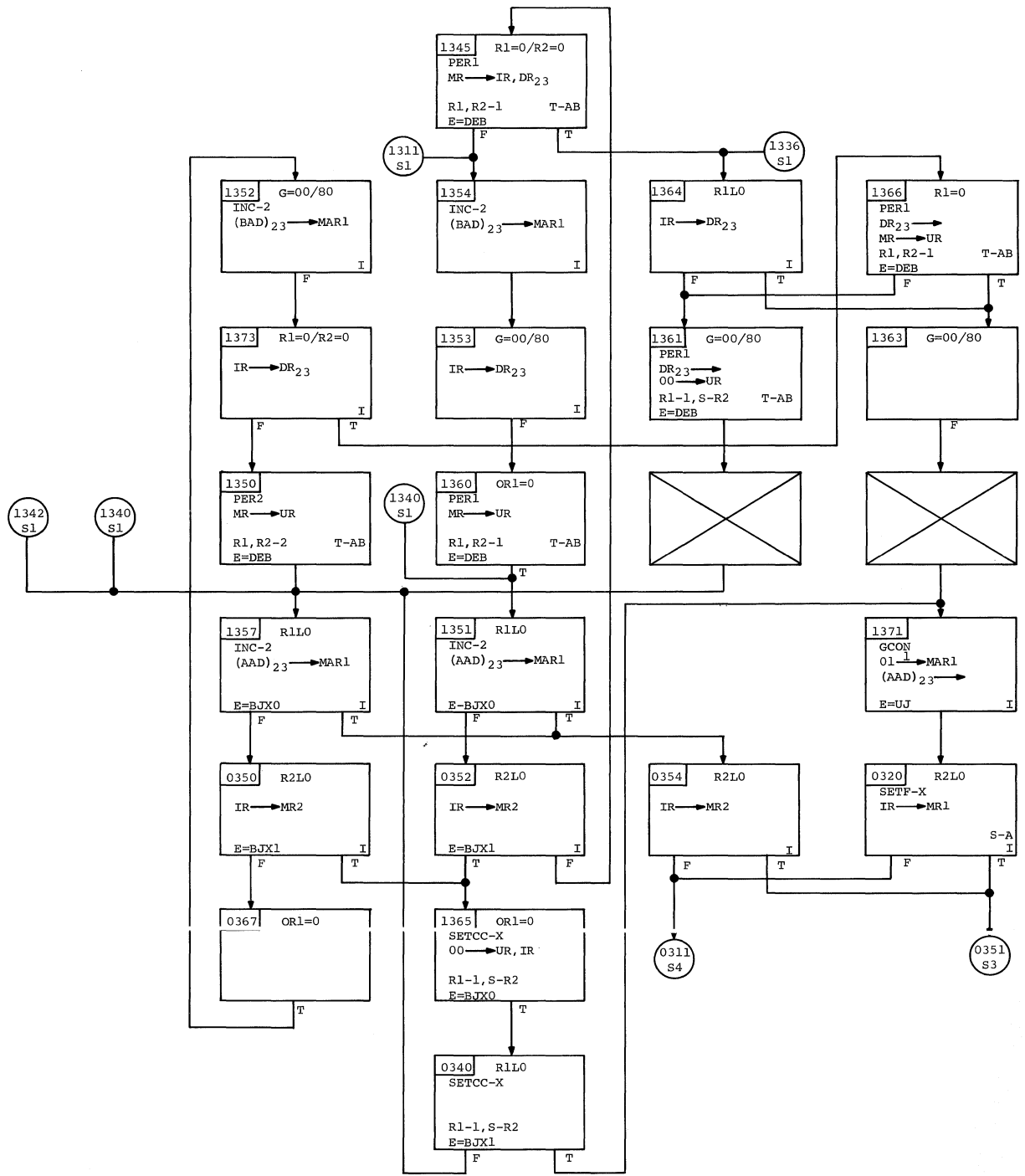
MOVE WITH OFFSET MVO SS FI

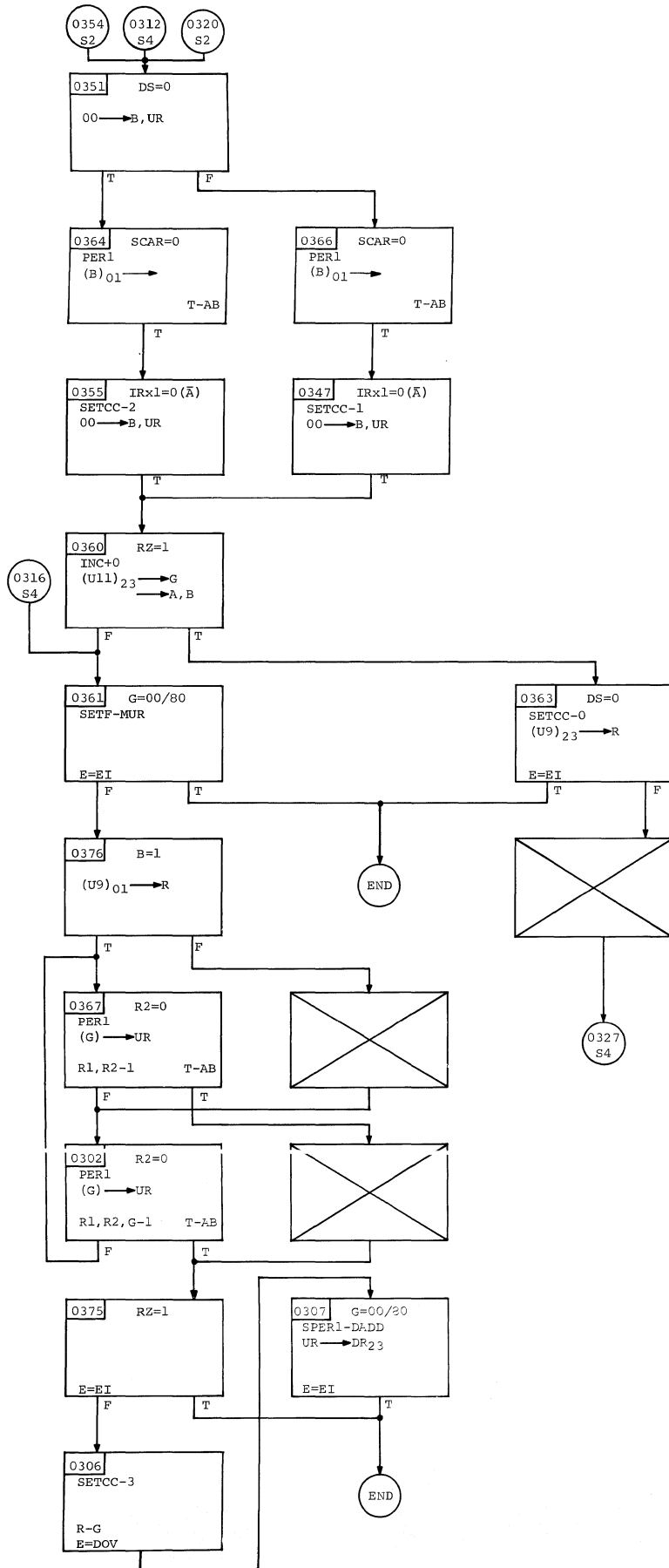




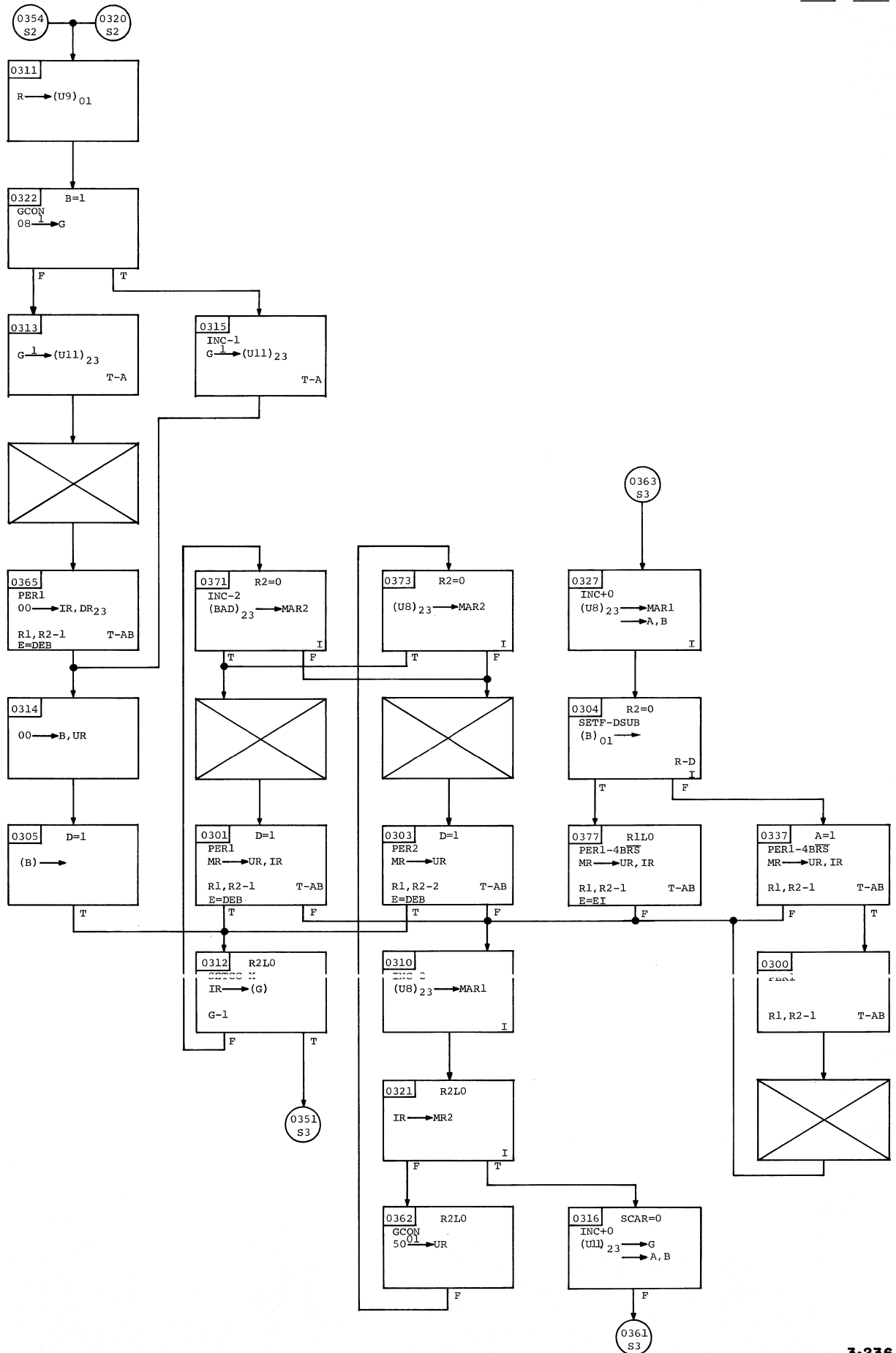


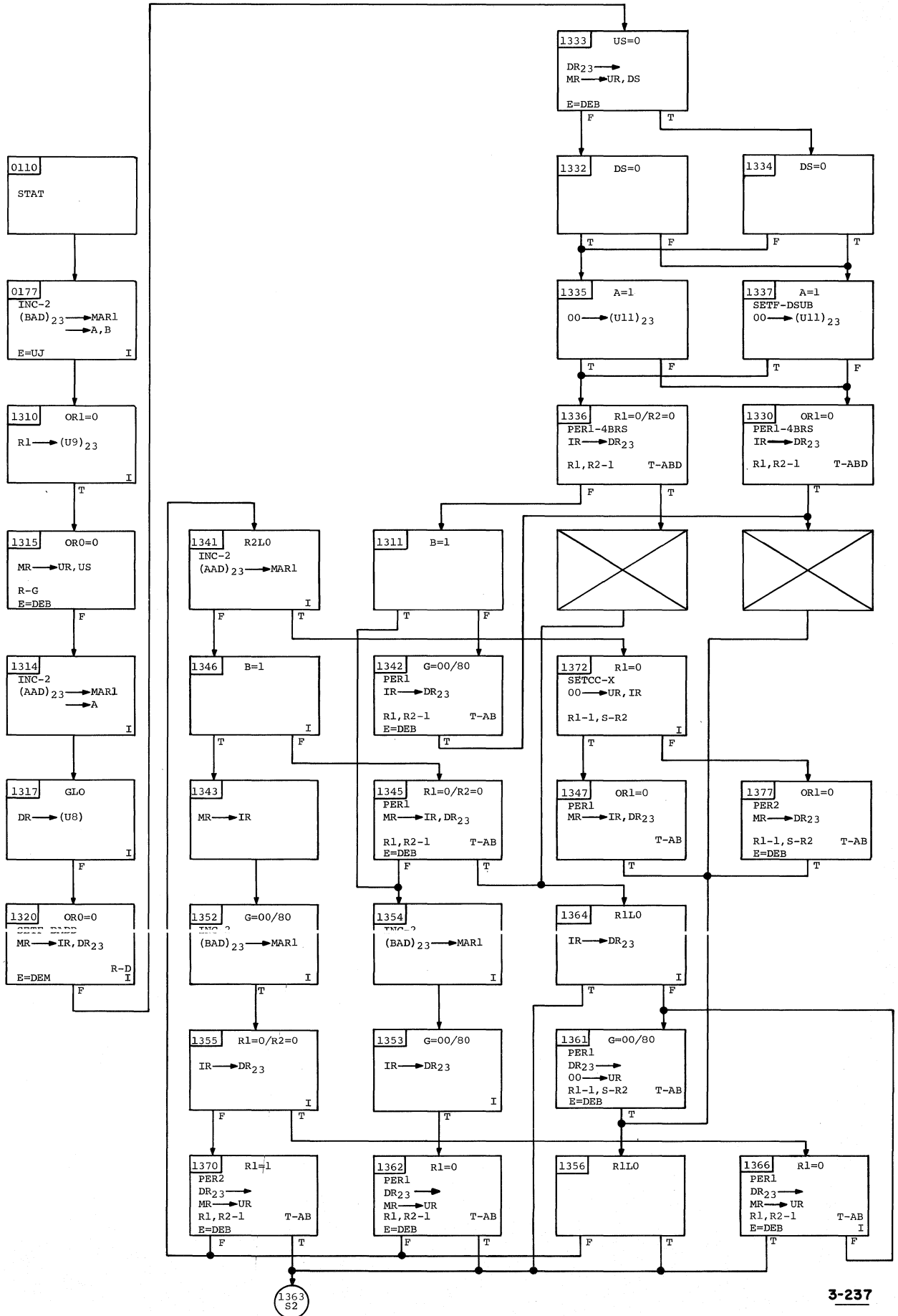


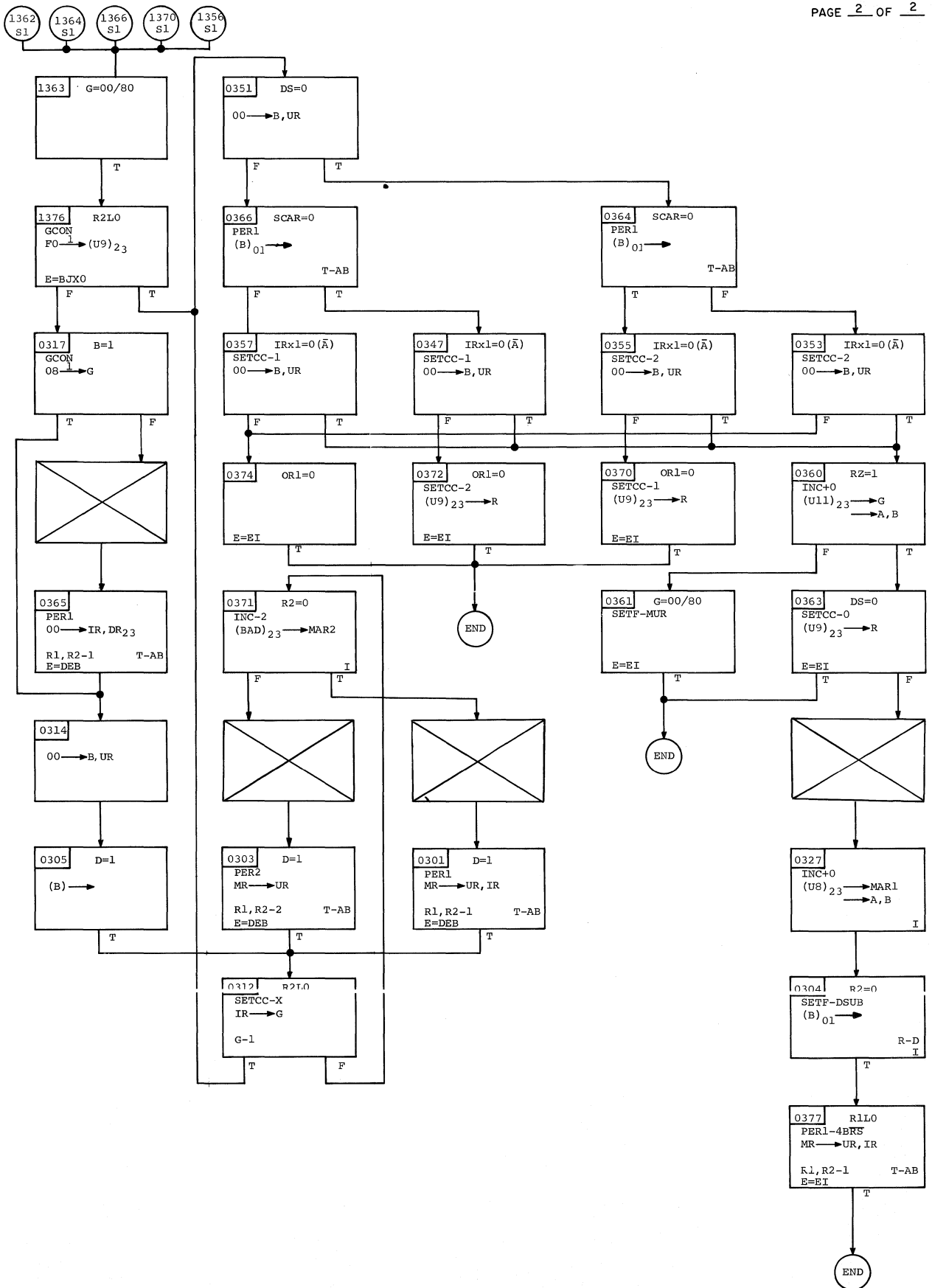


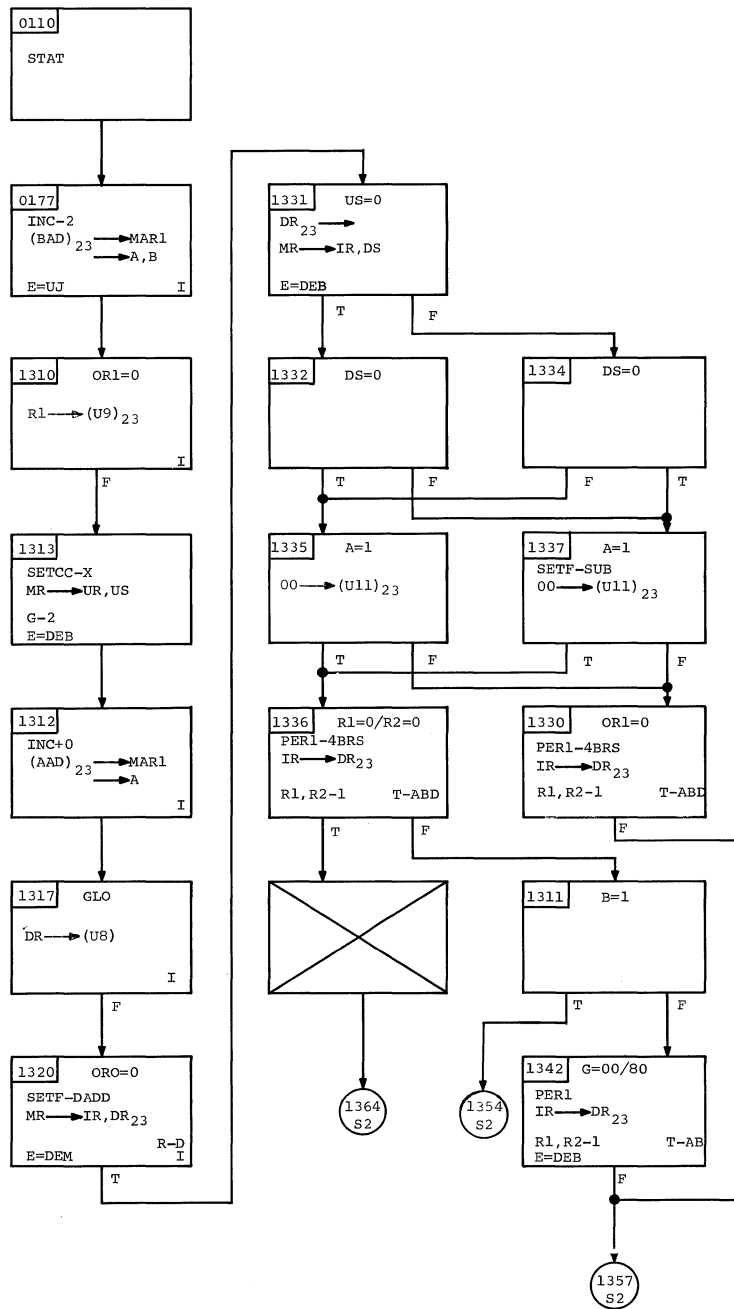


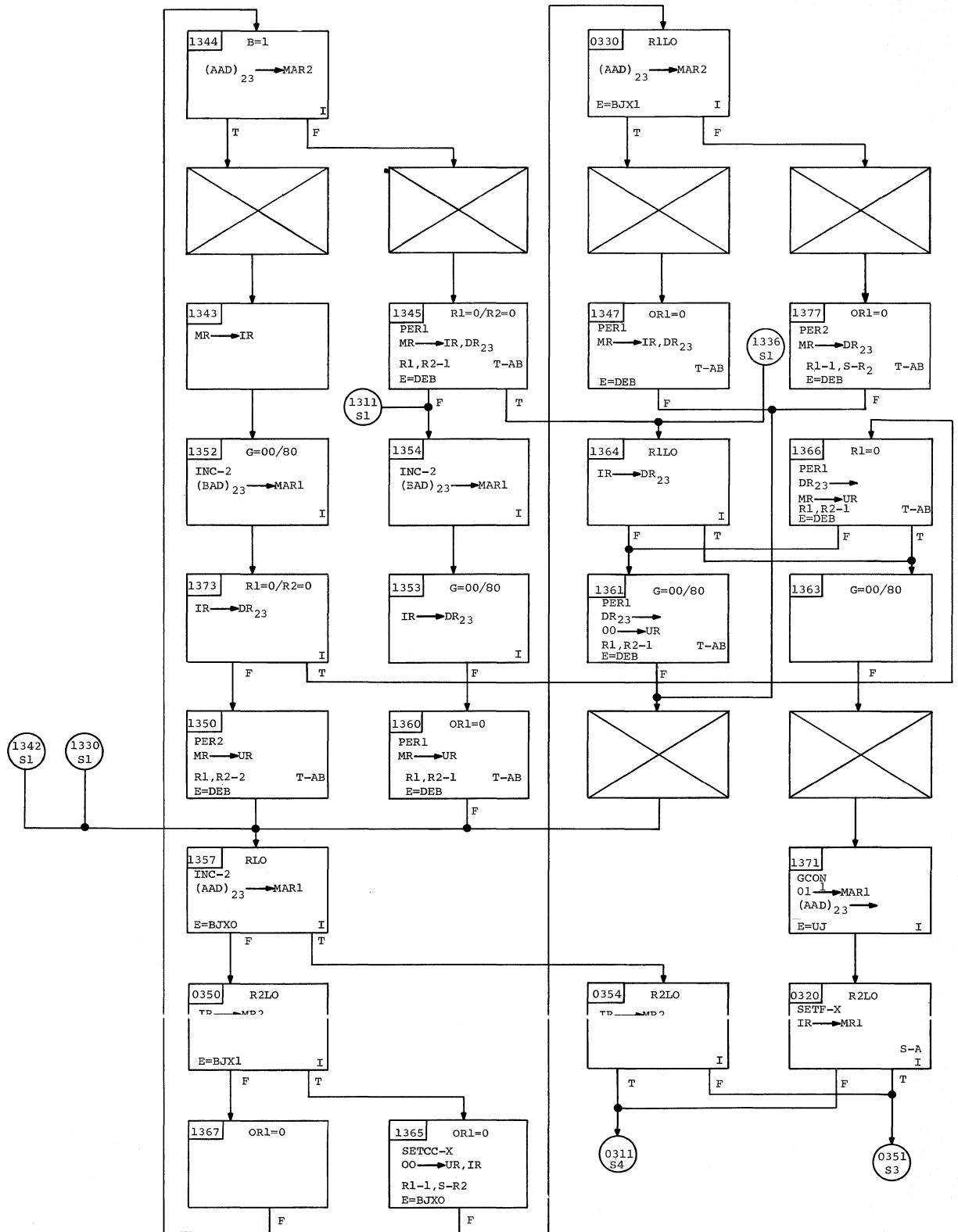


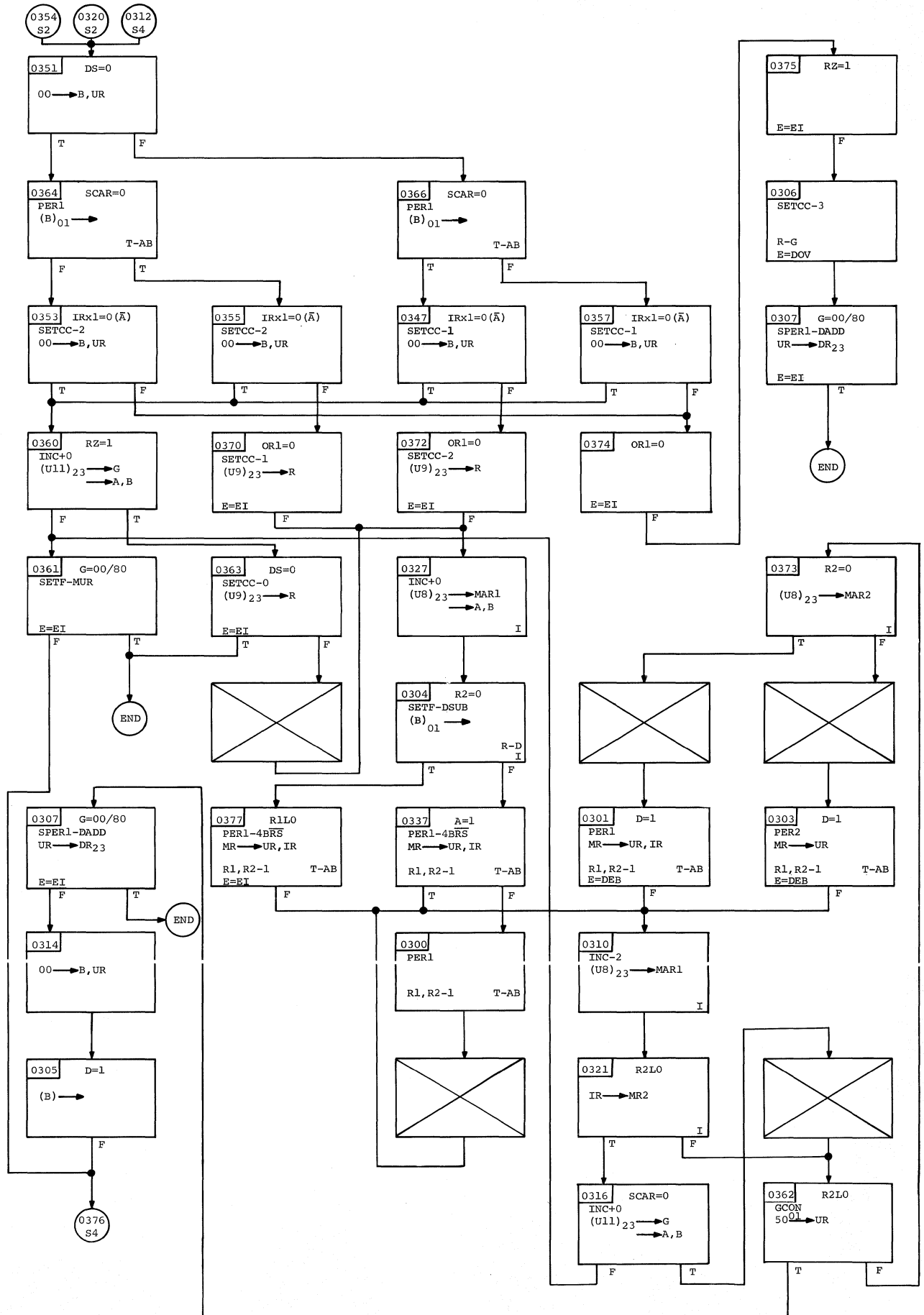




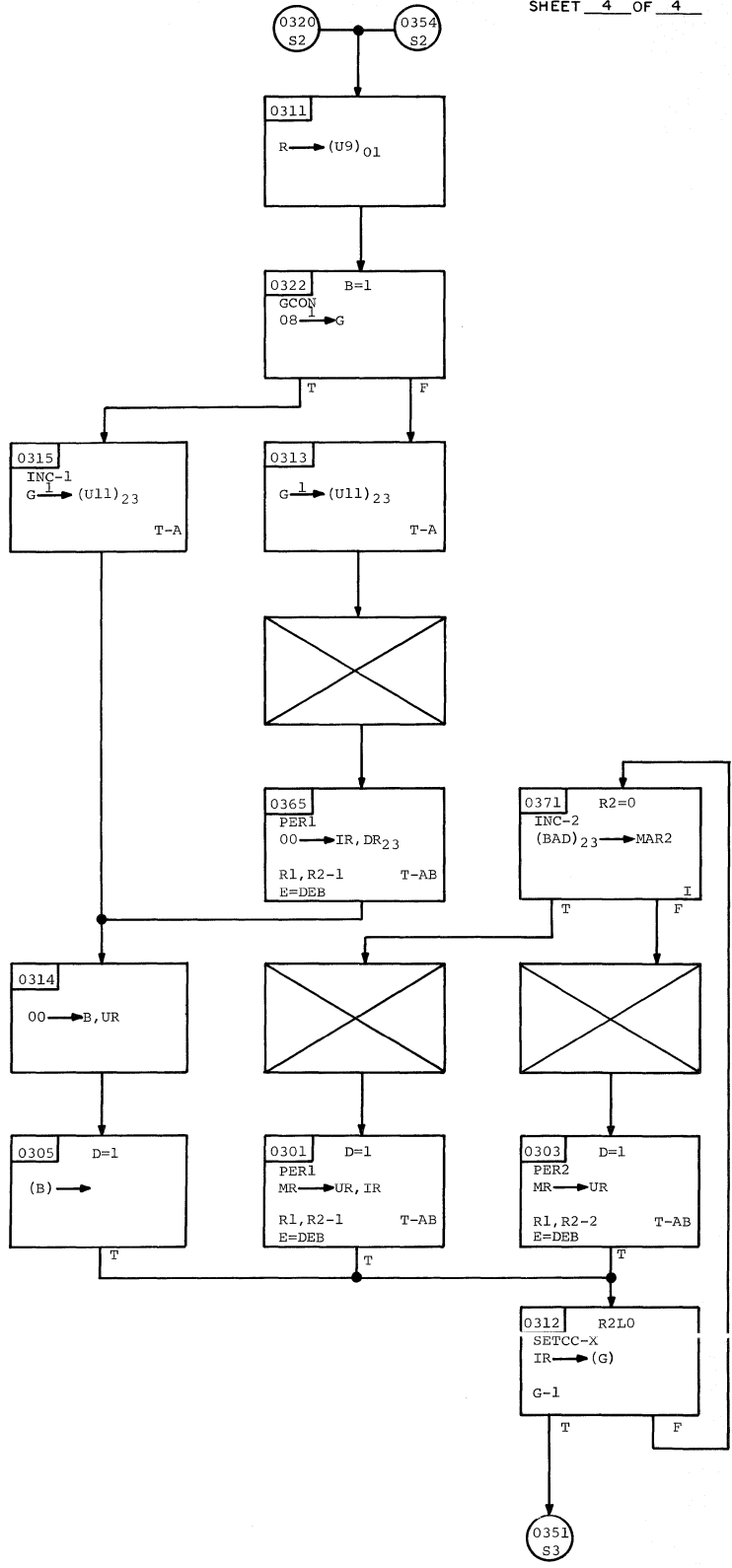
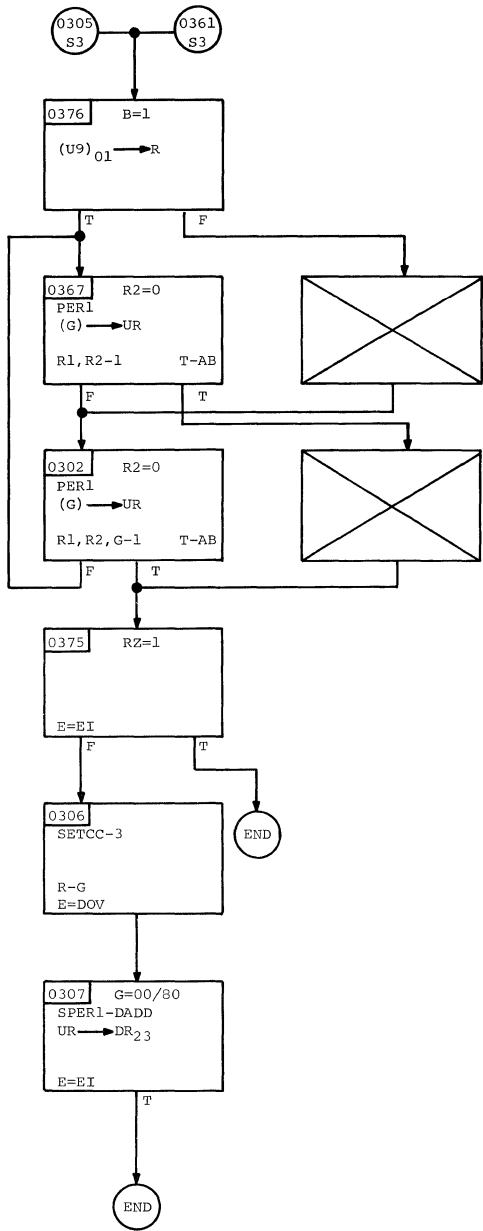


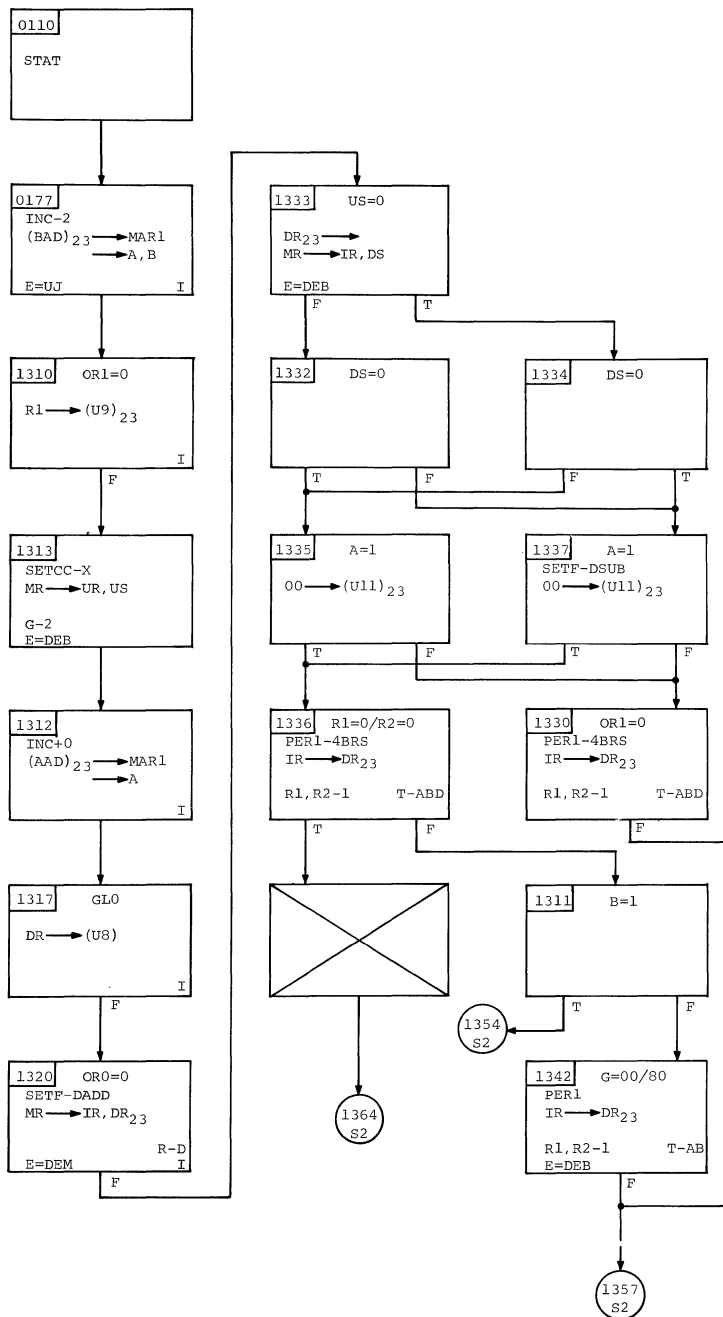




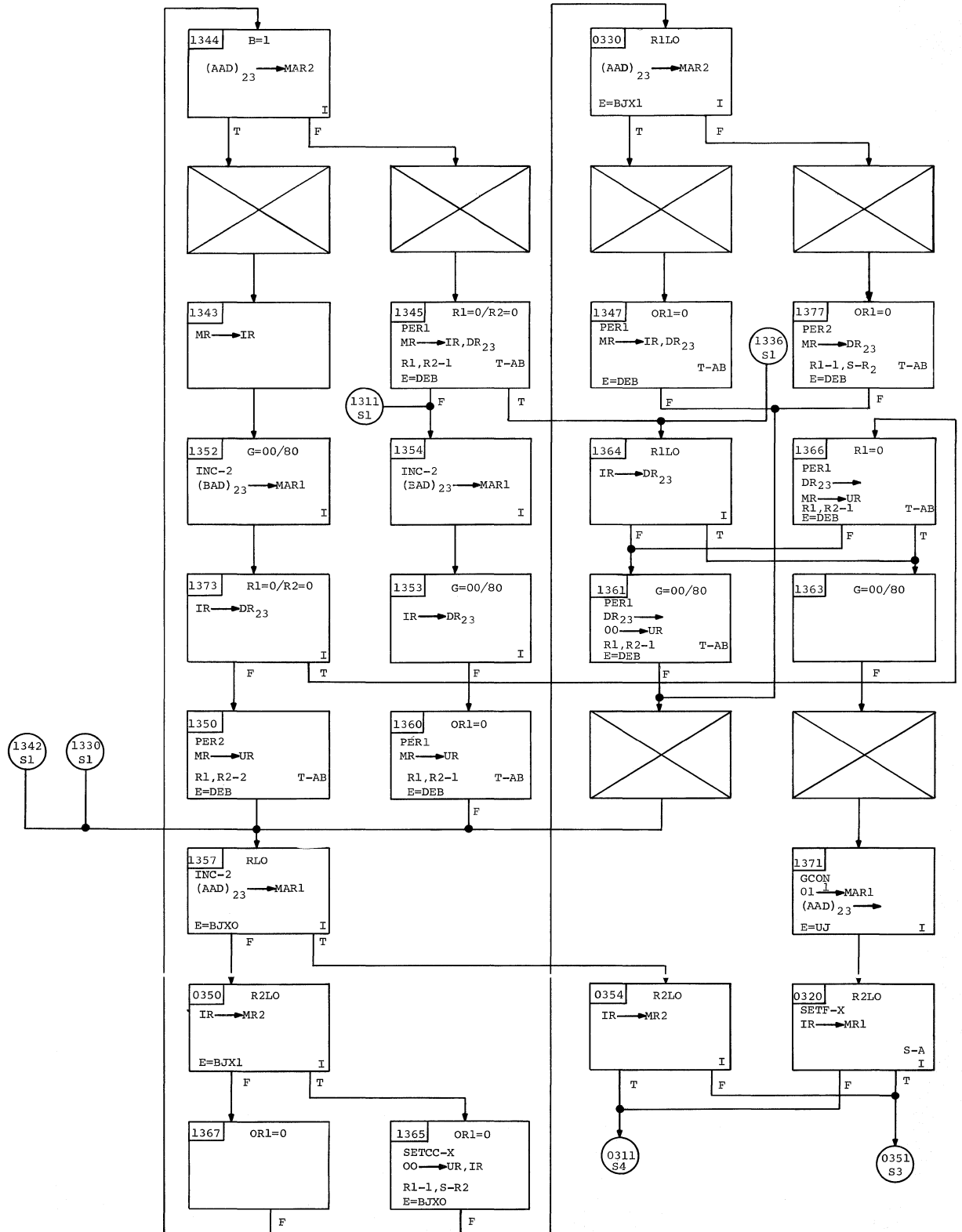


ADD DECIMAL AP SS FA

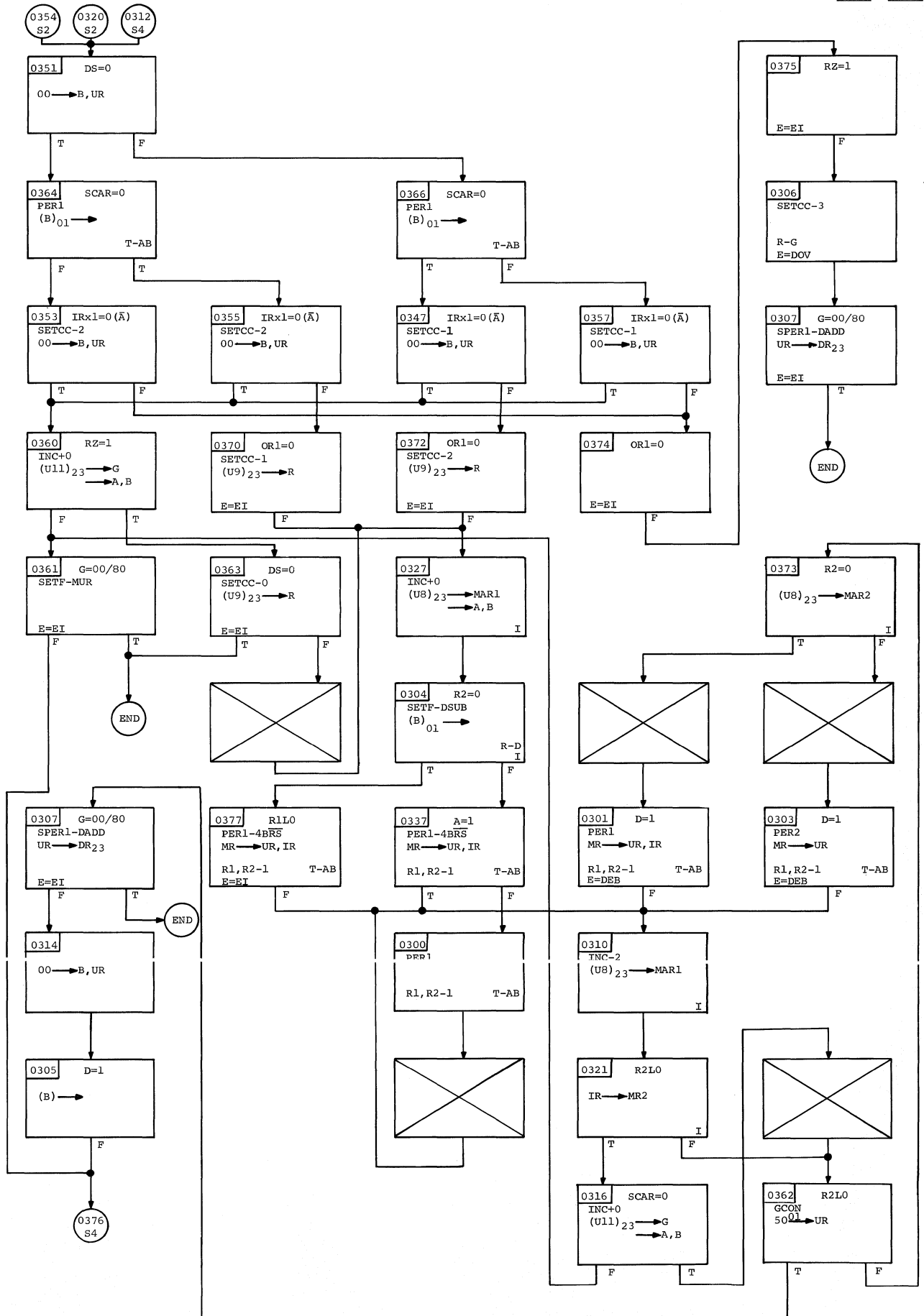


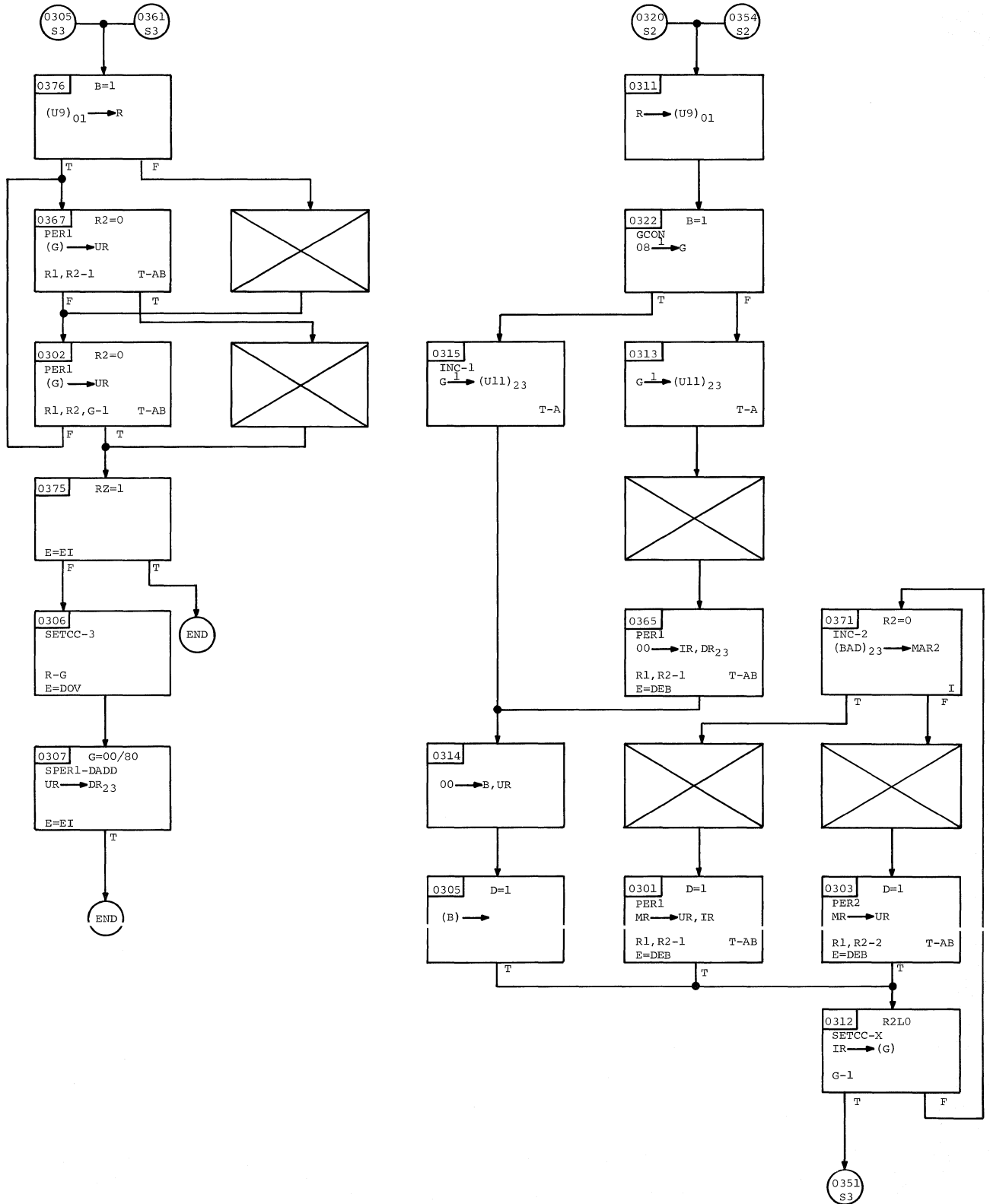


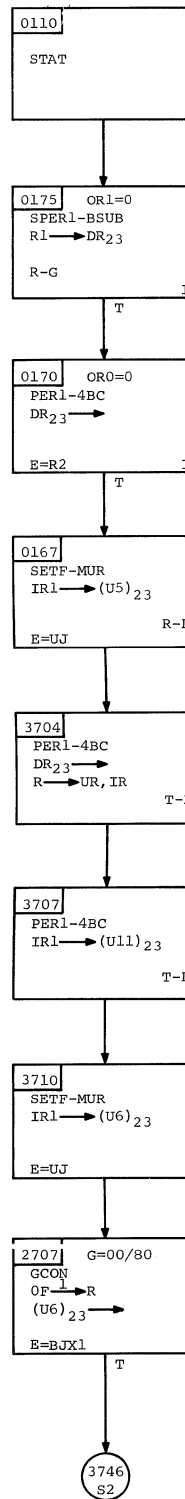


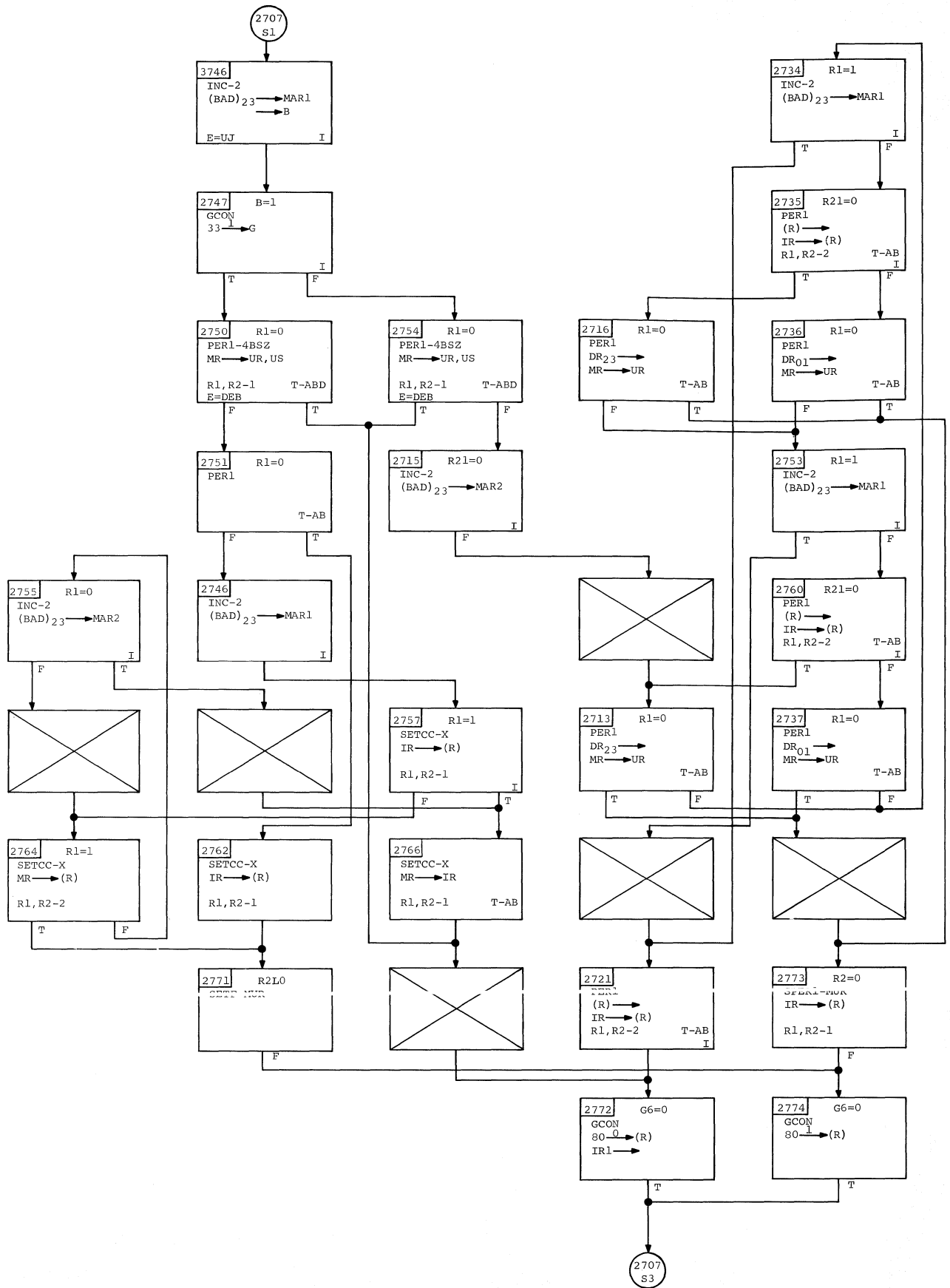


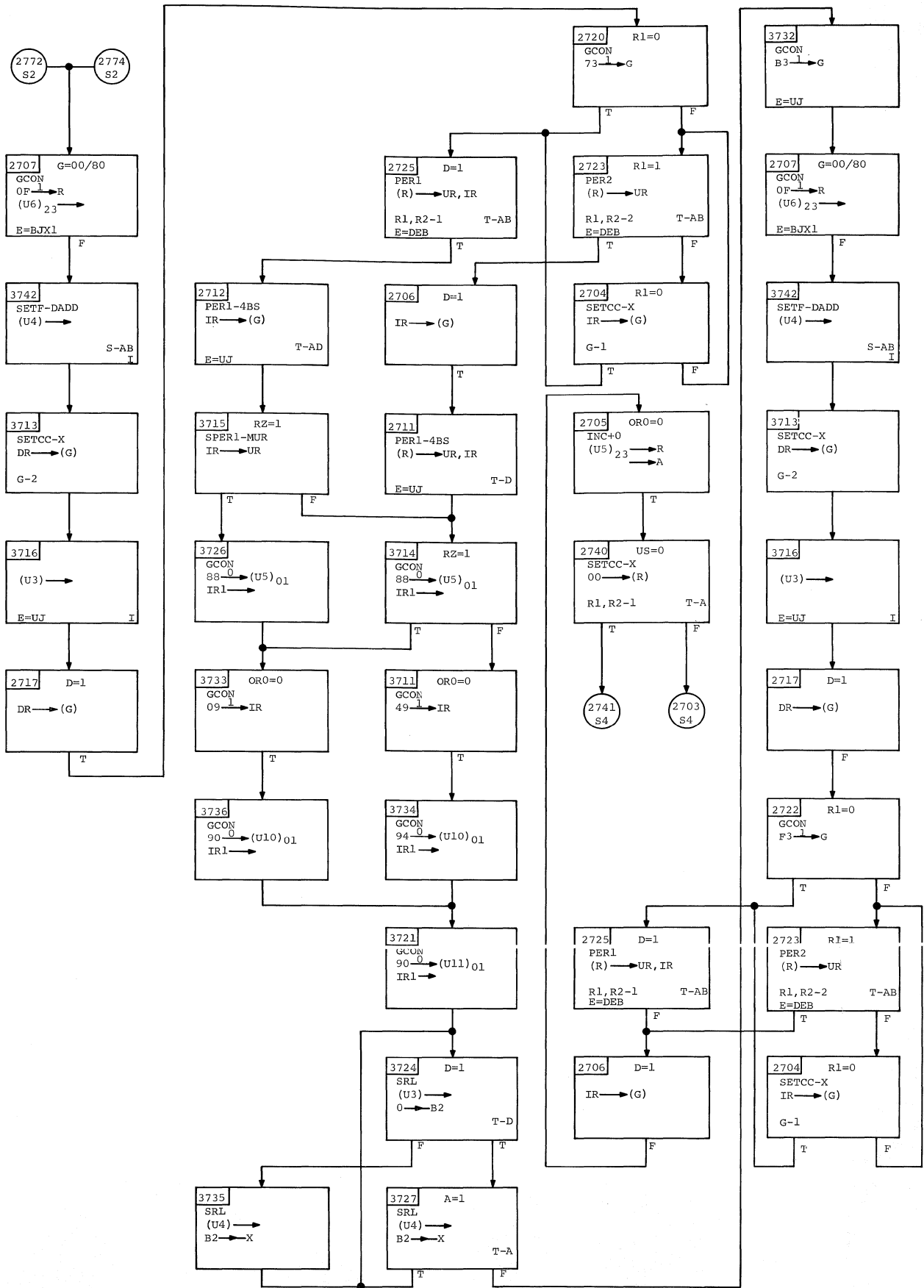
SUBTRACT DECIMAL SP SS FB

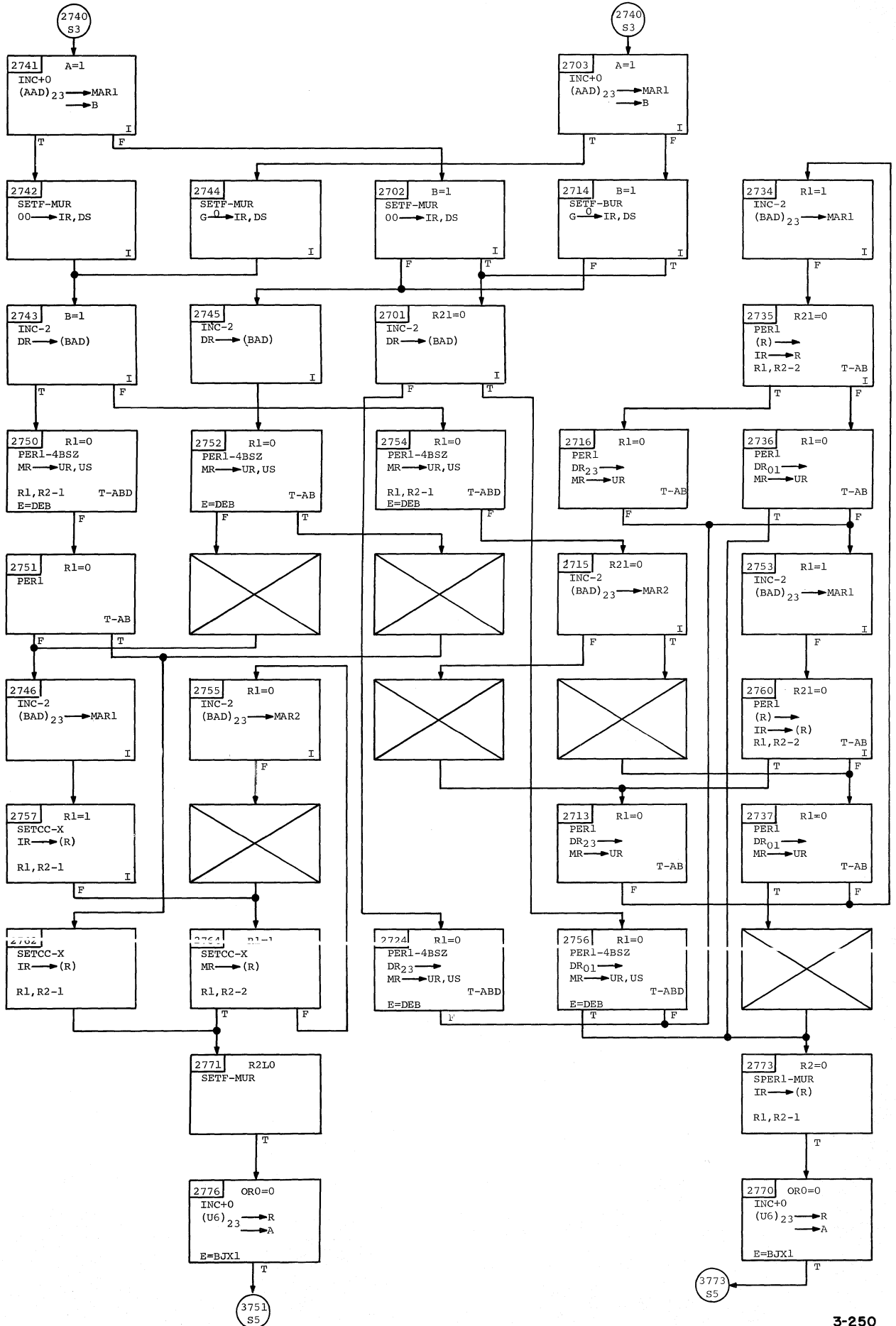


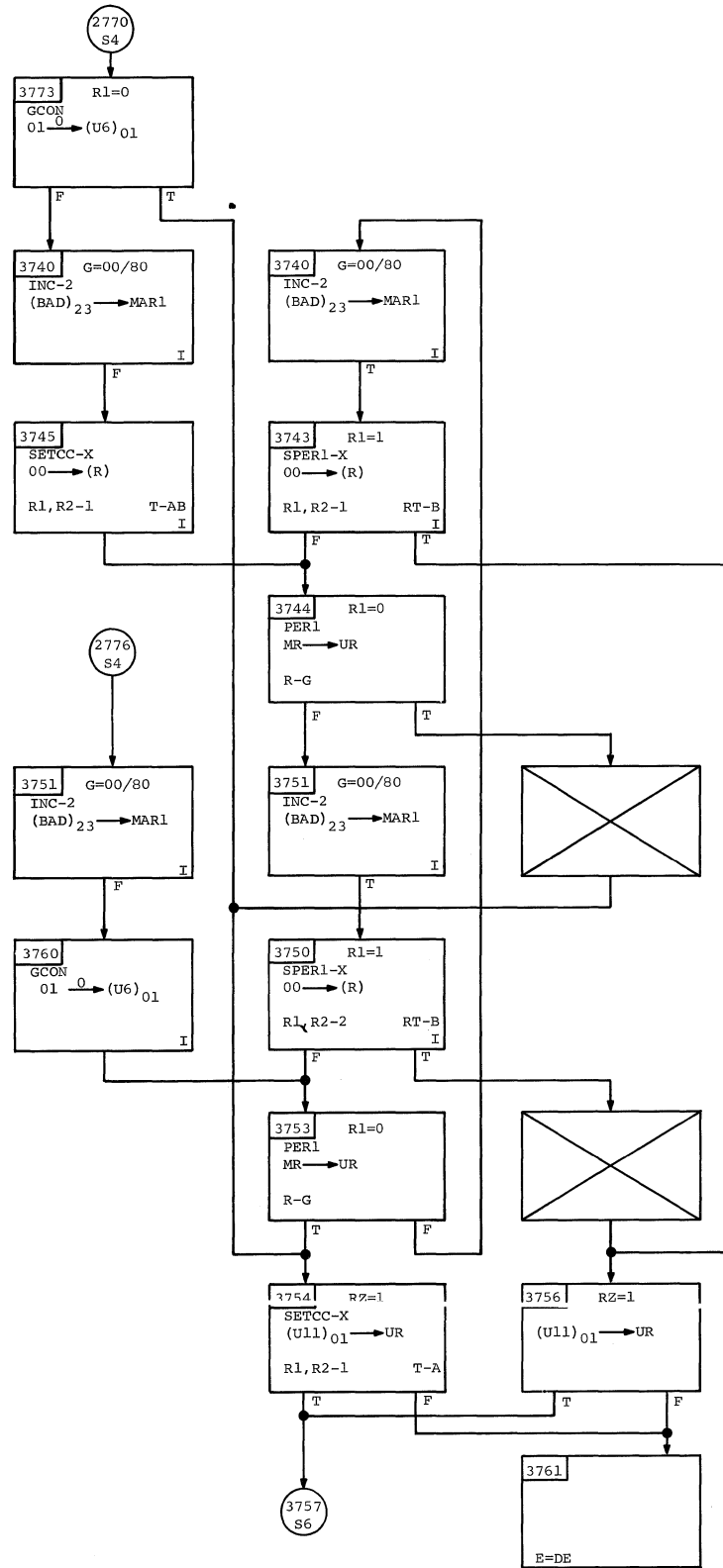




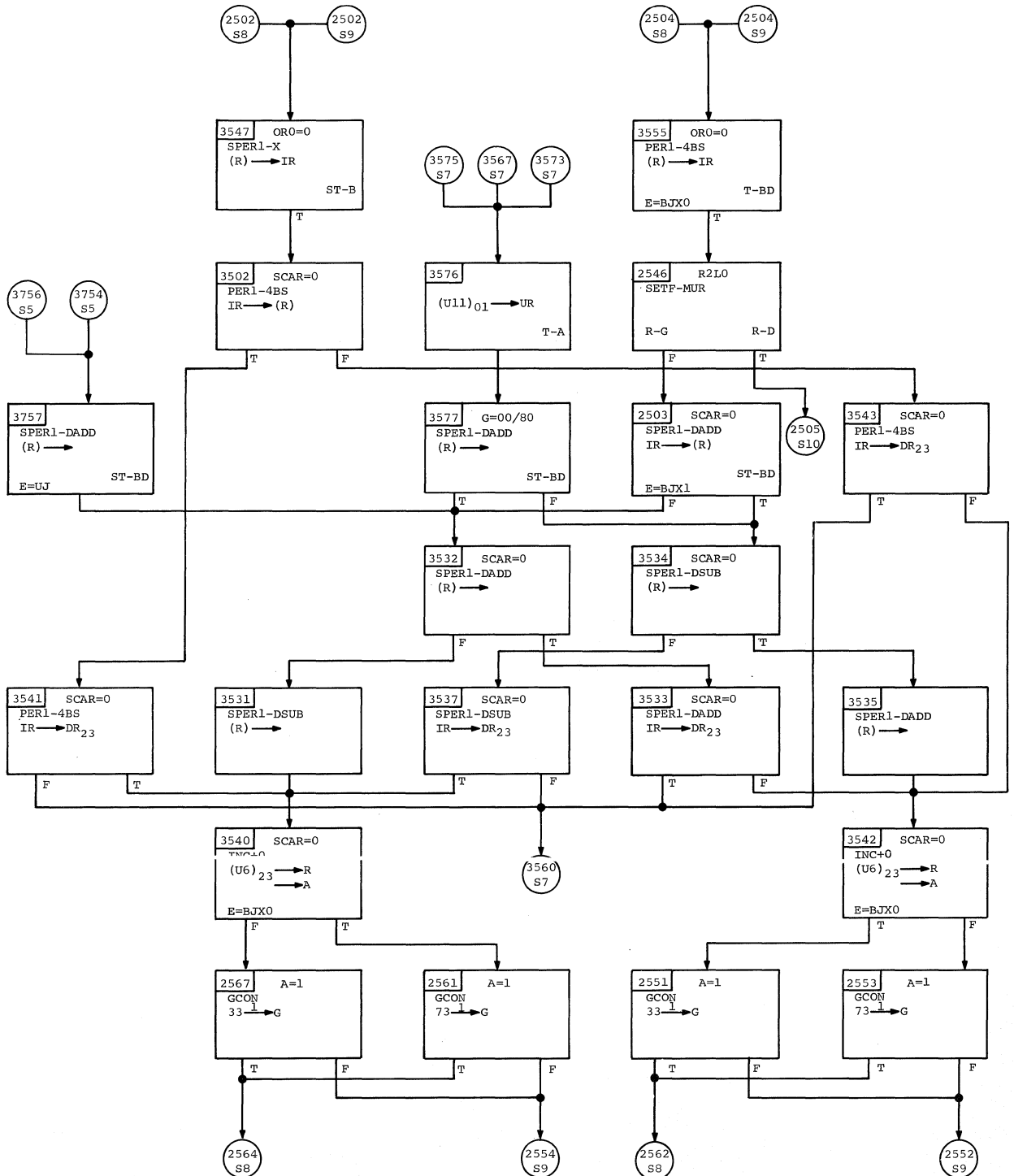


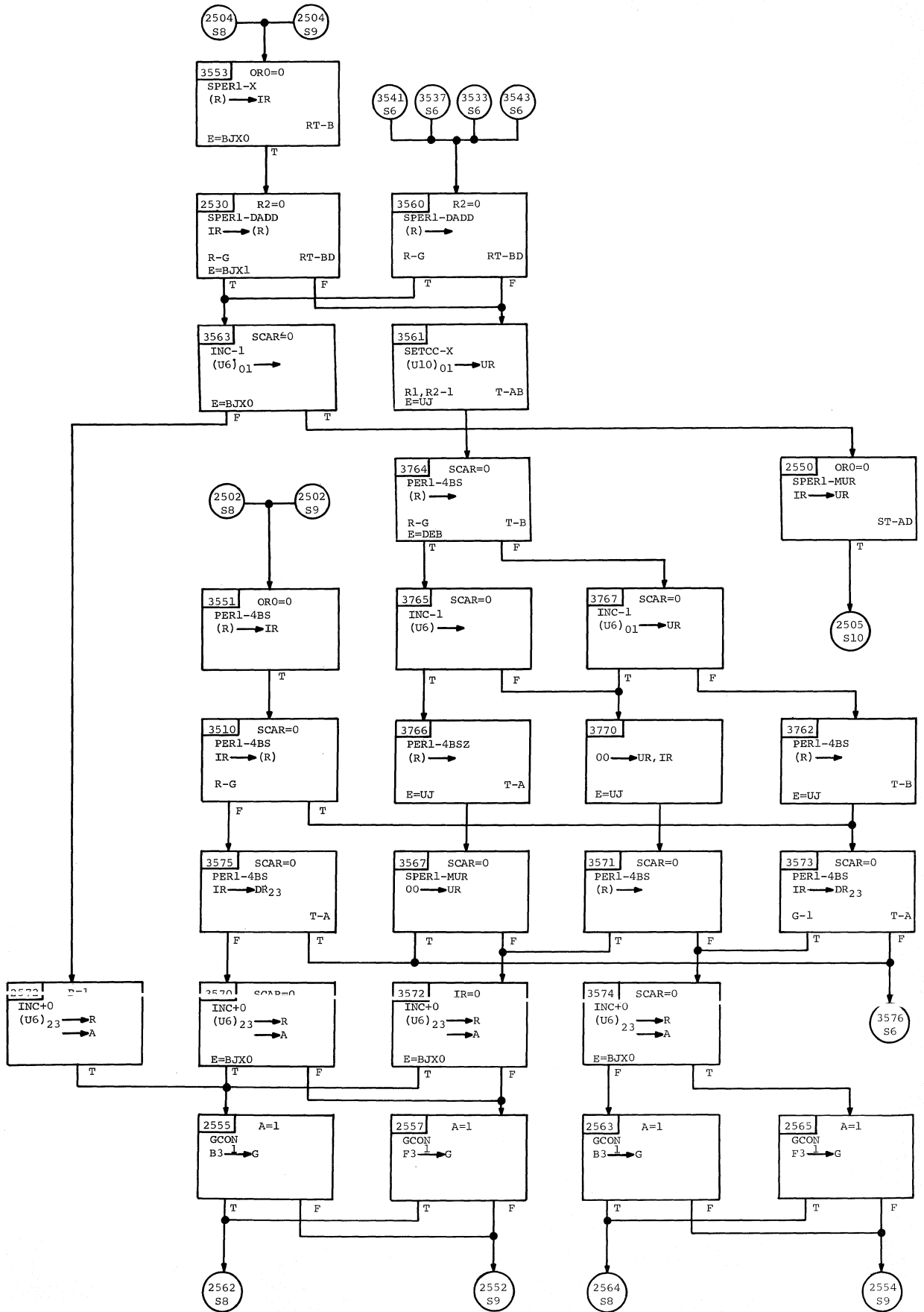


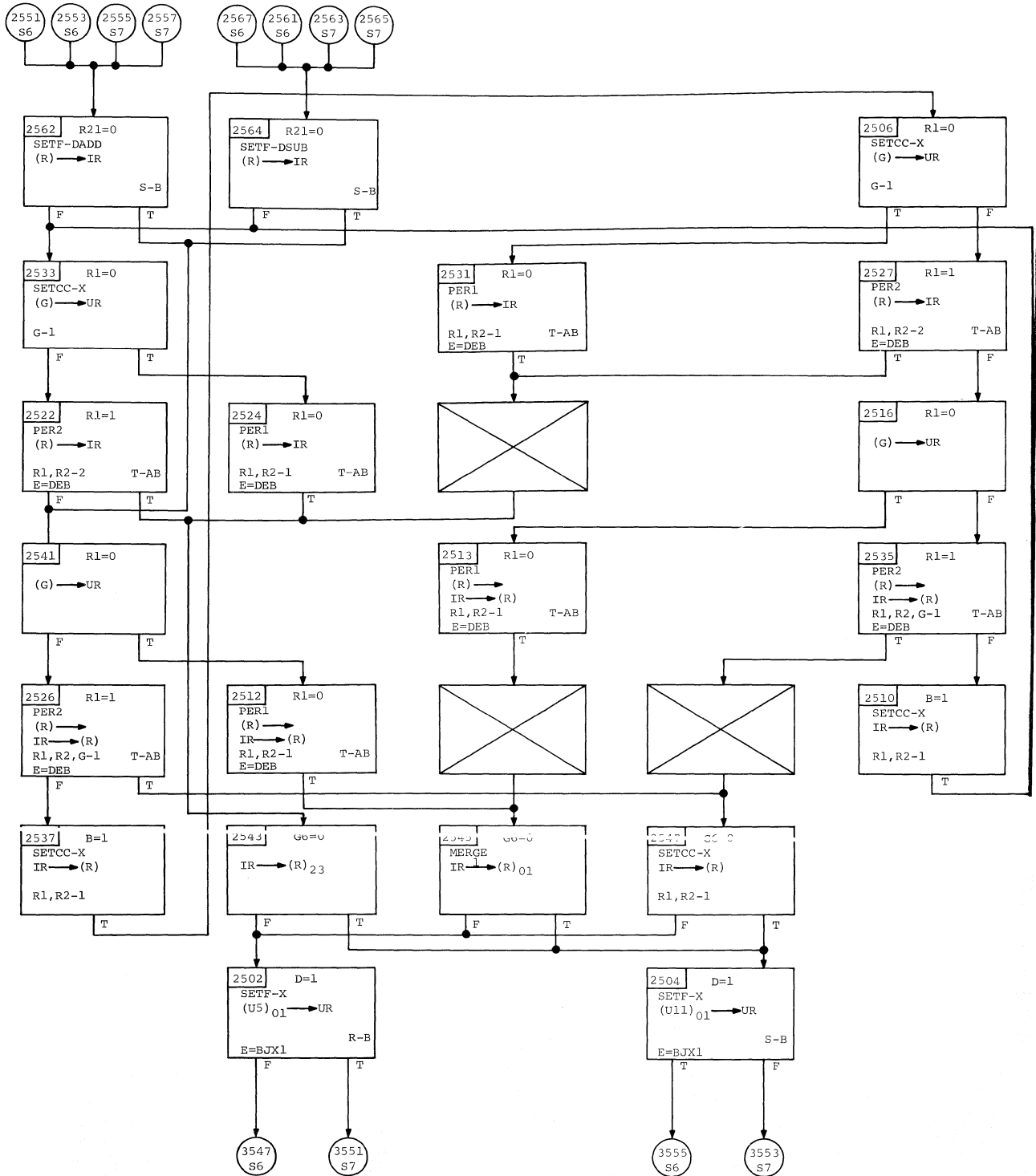


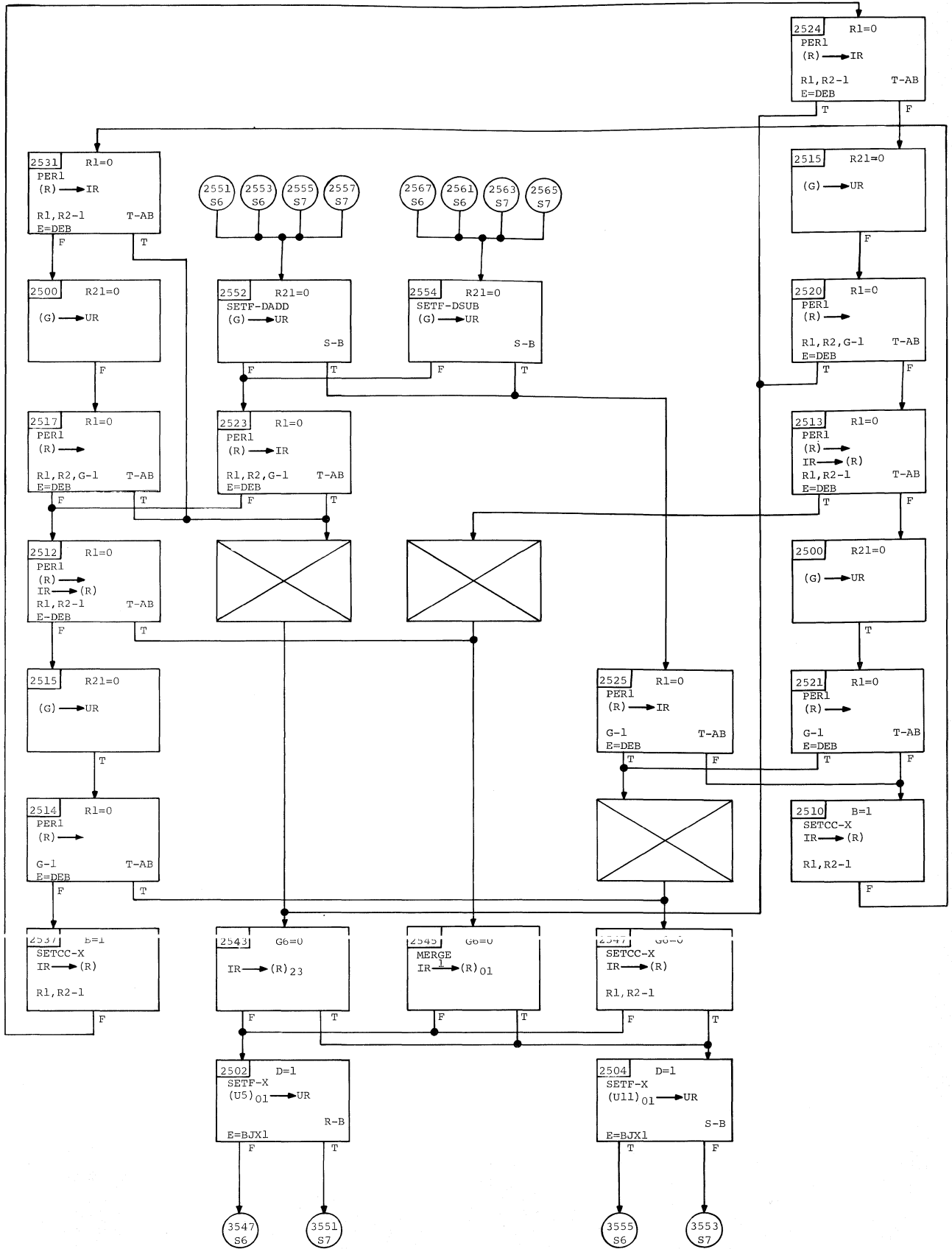


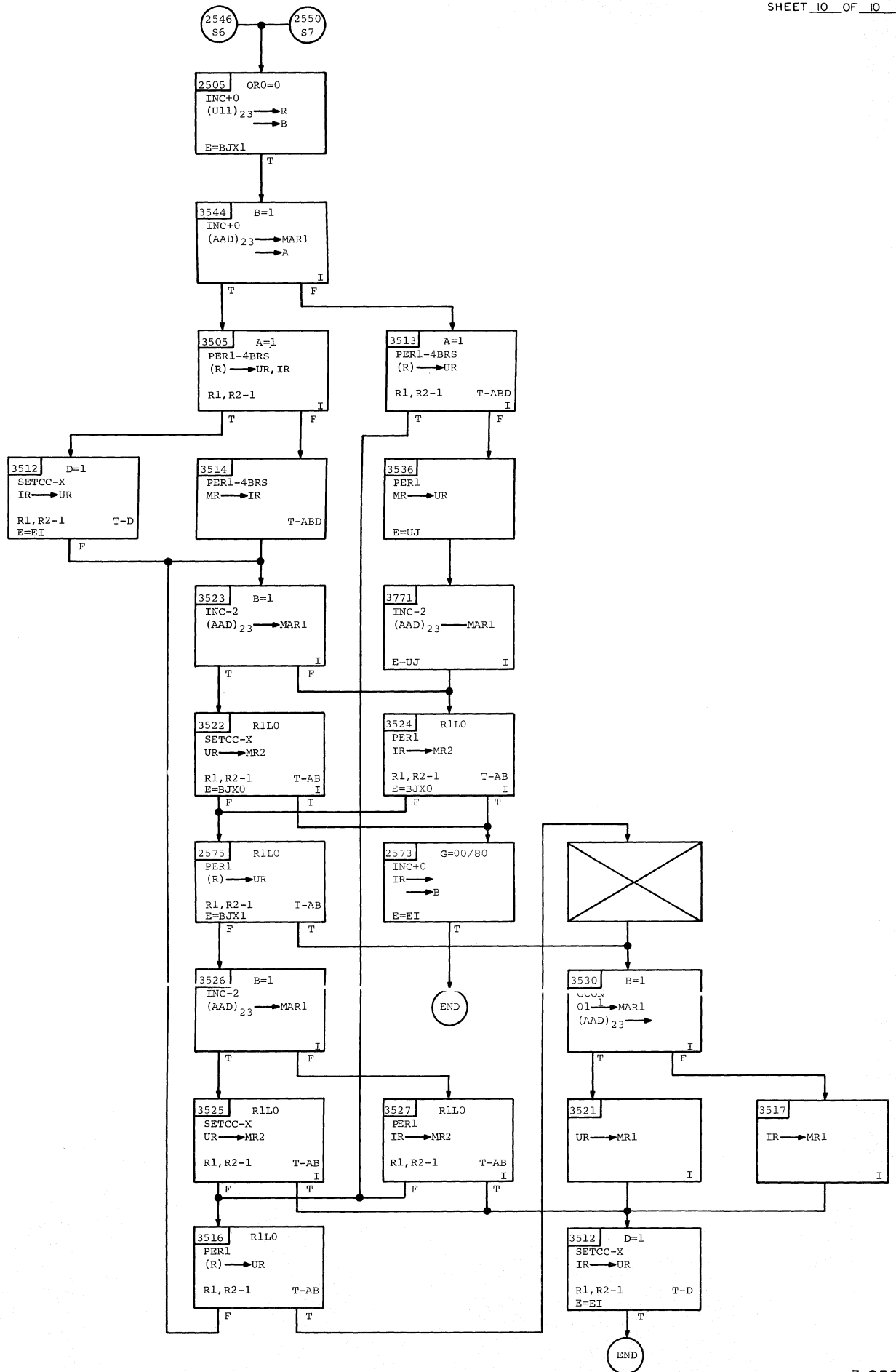


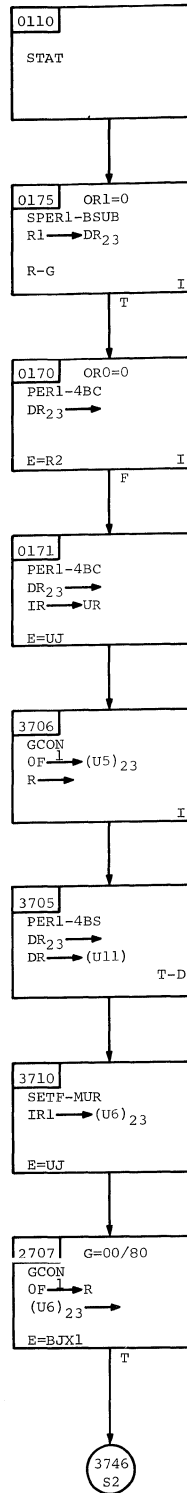


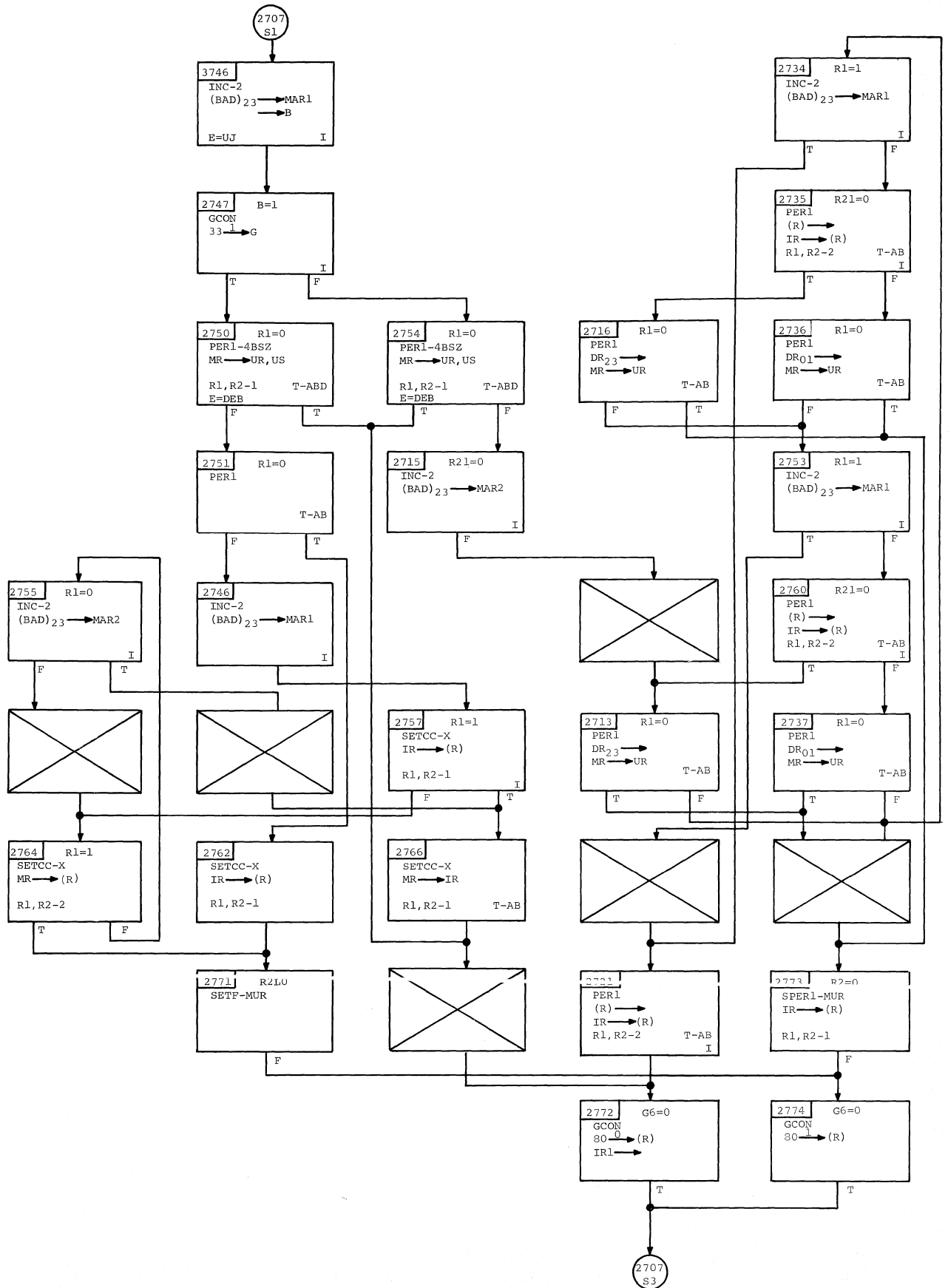


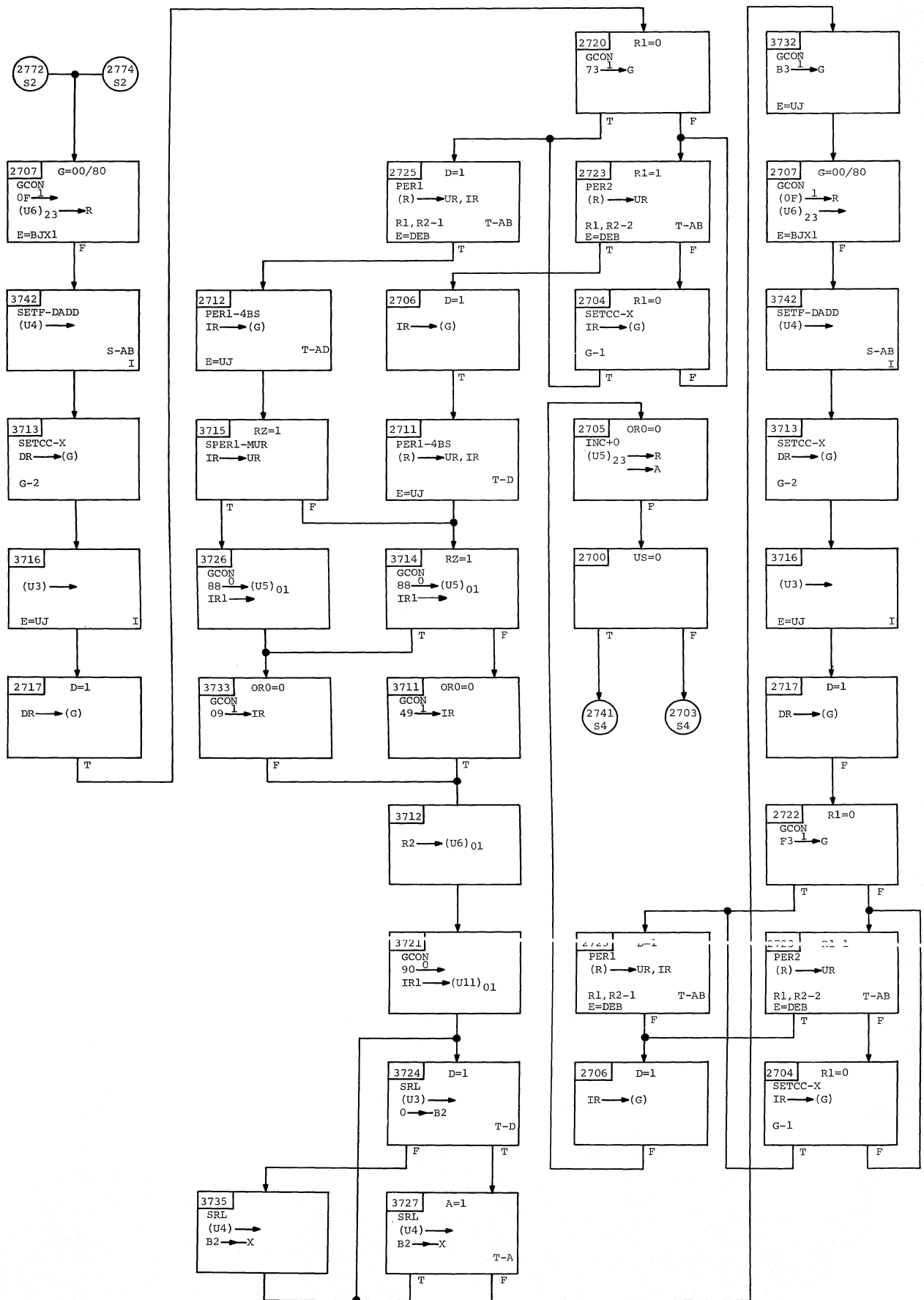




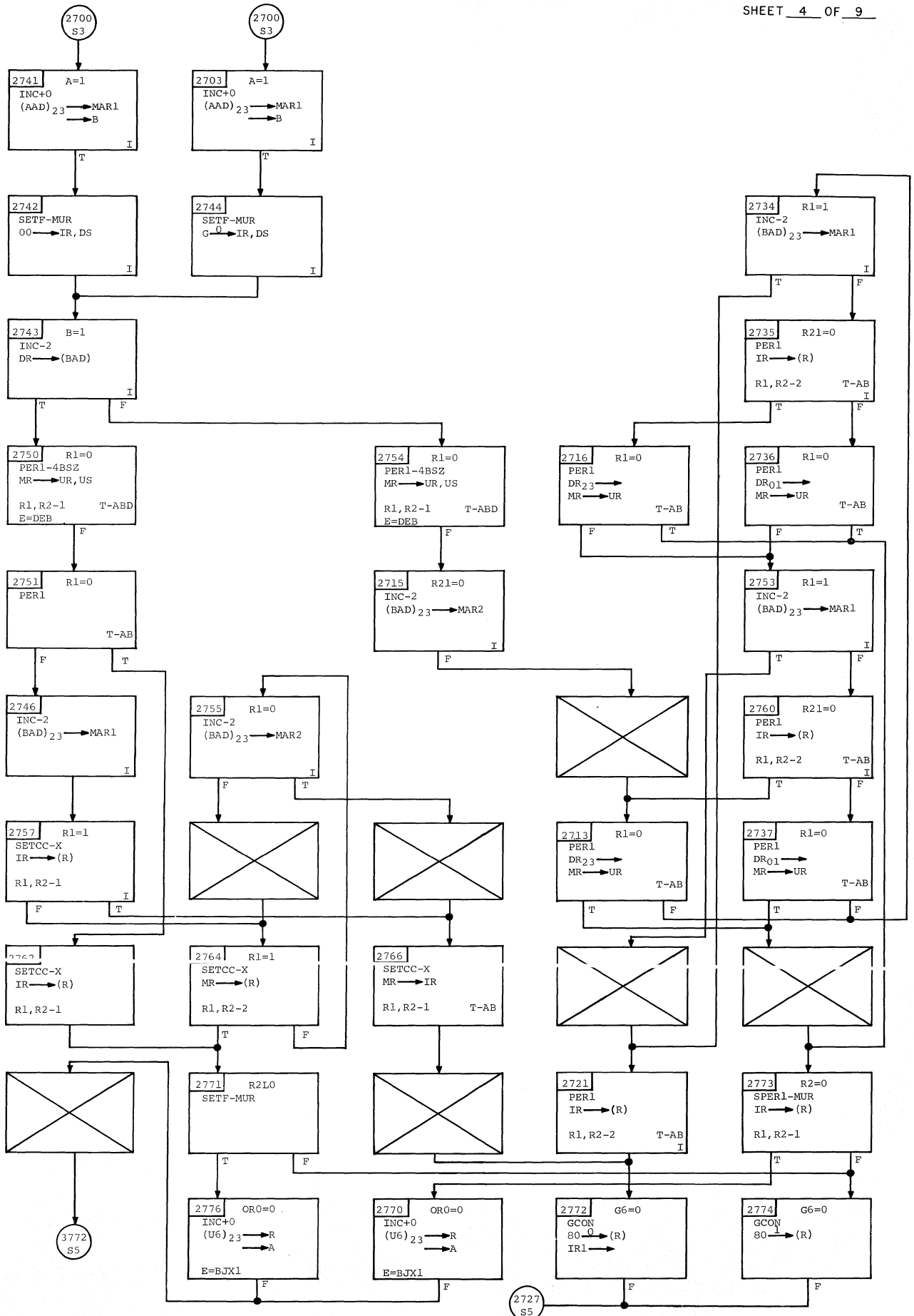


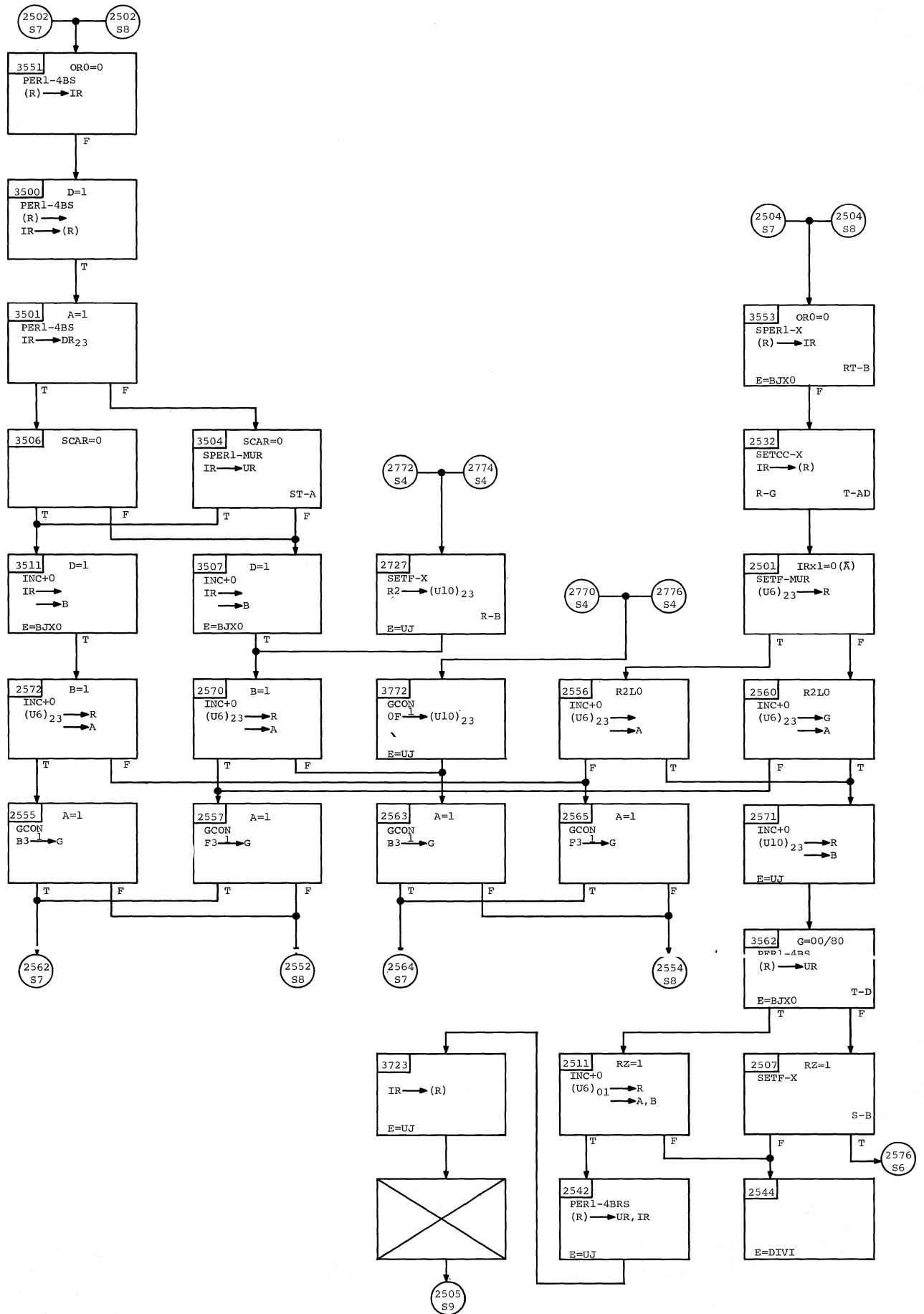


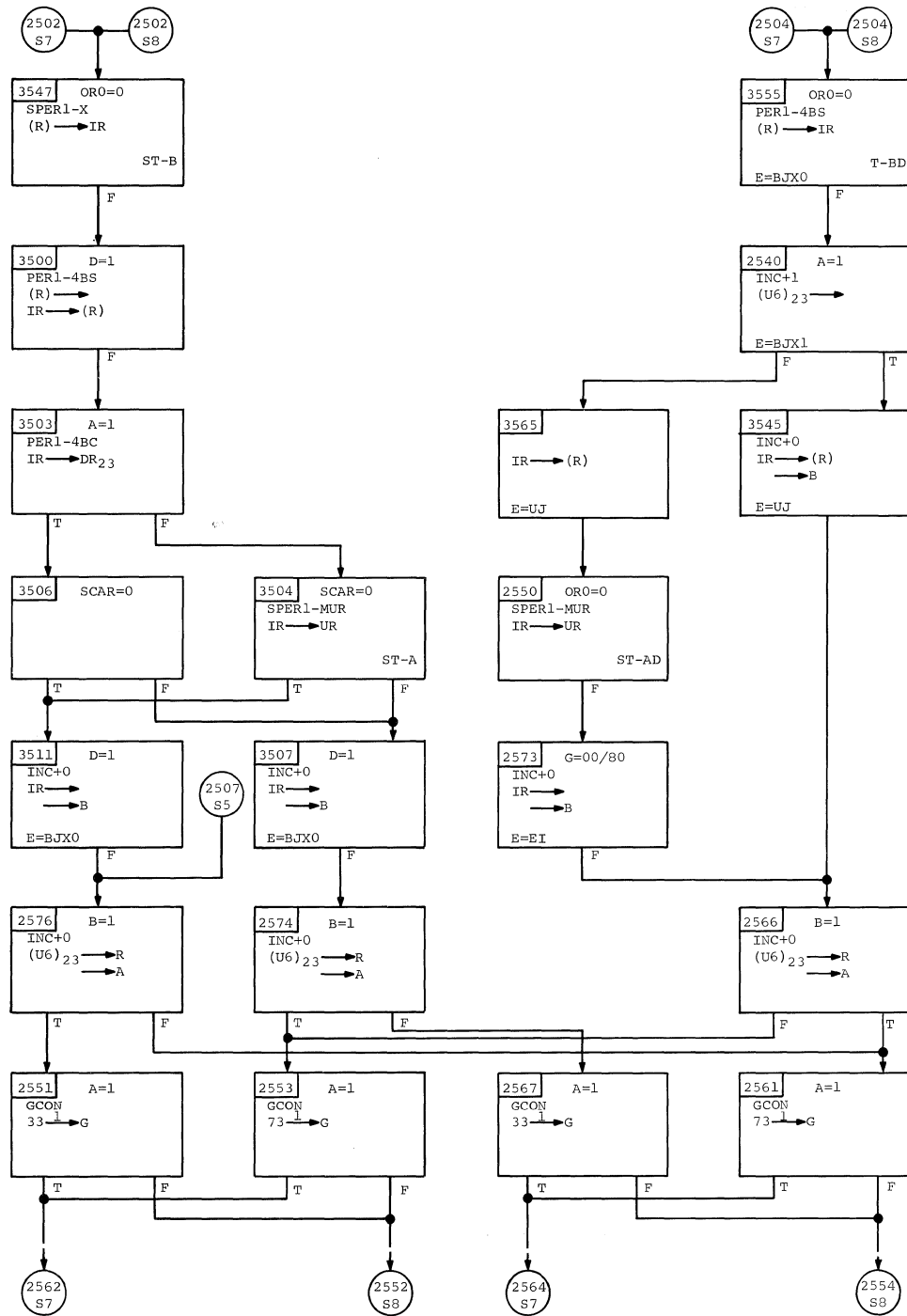


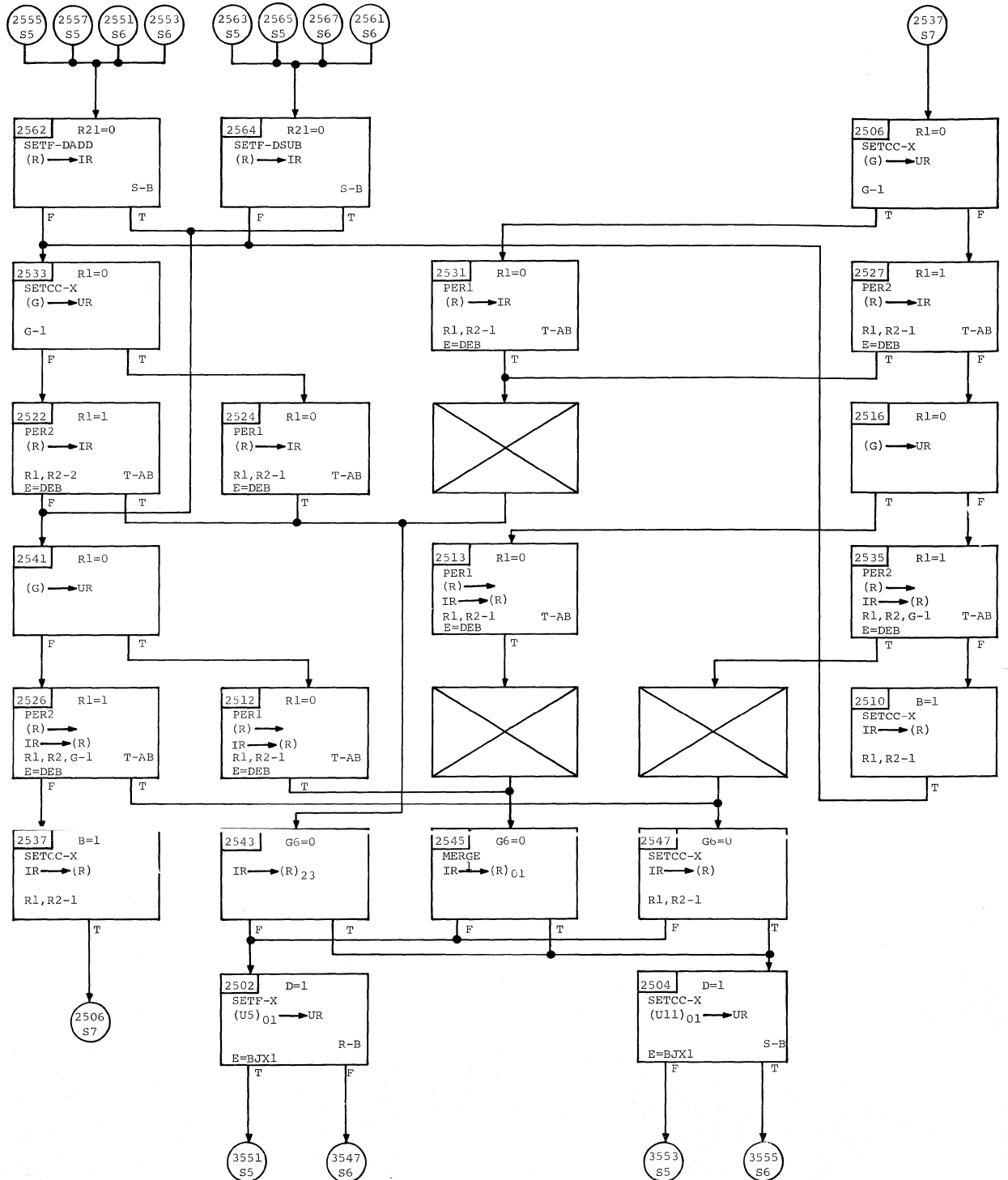




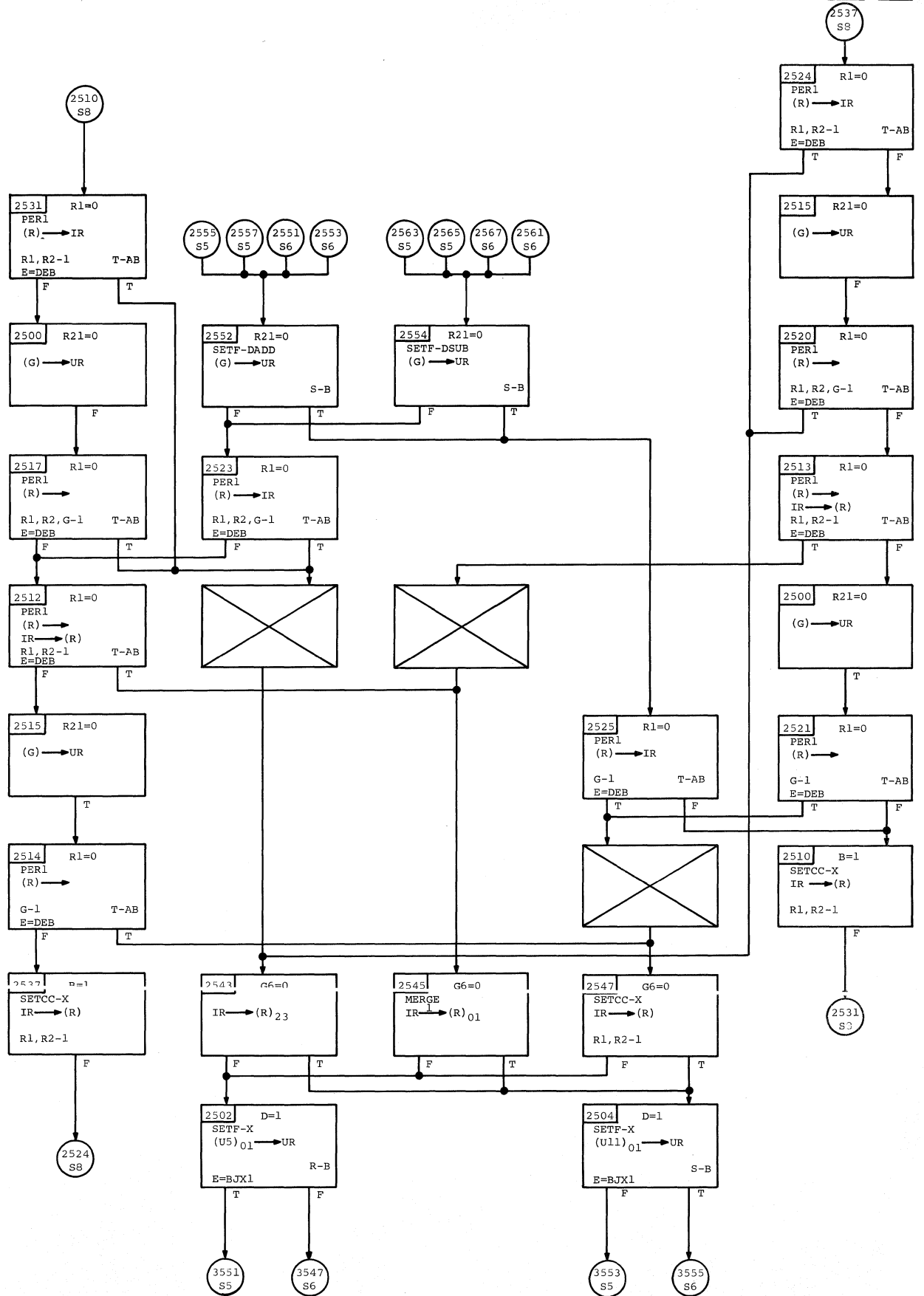


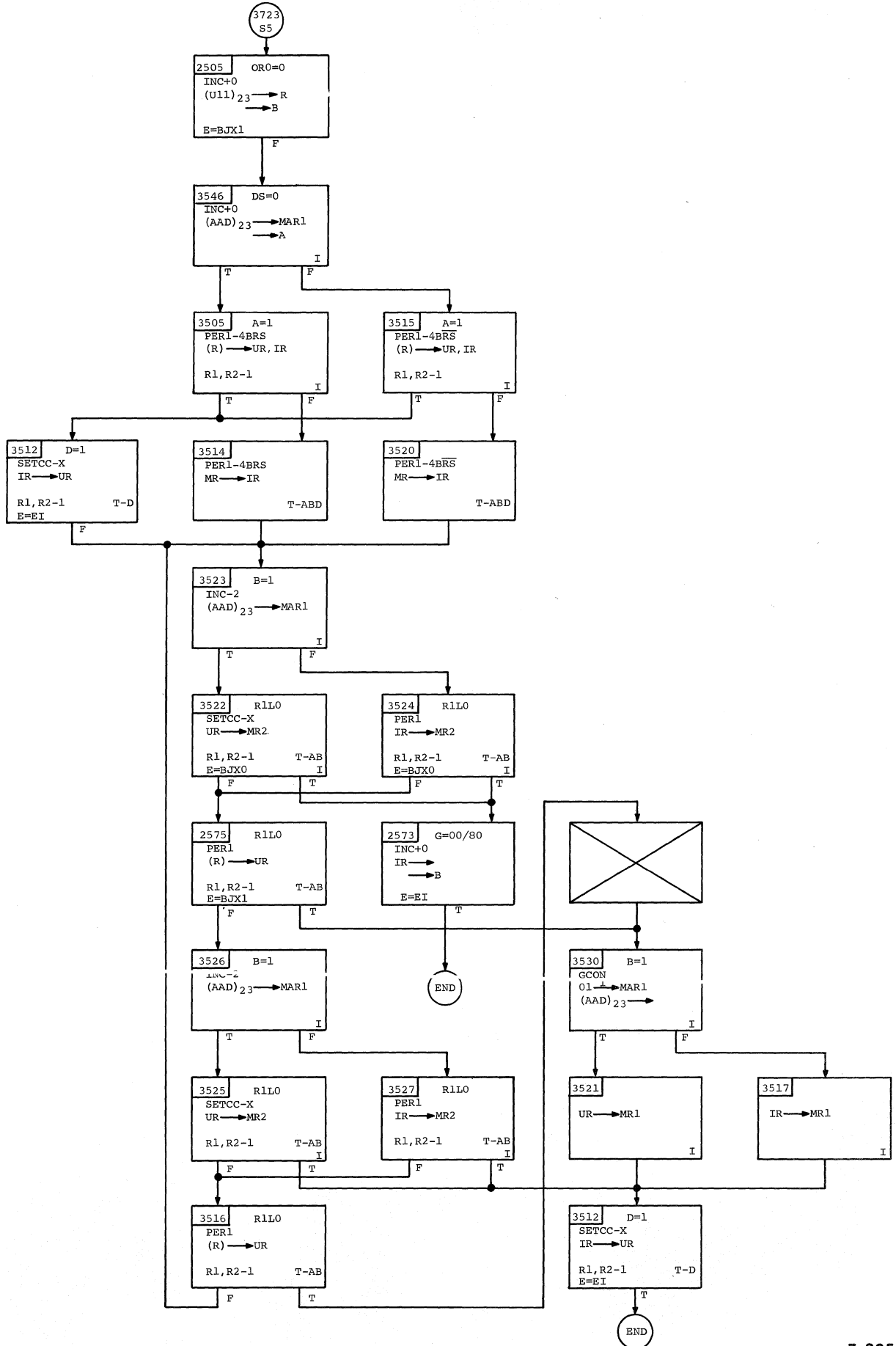






# DIVIDE DECIMAL DP SS FD







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