Contract No. FA64WA-5223

IBM Field Engineering Maintenance Diagrams

7201-02 Computing Element

SFN-0202-1

PREFACE

This manual contains the maintenance-oriented and recall diagrams referenced in the companion 7201-02 Computing Element FETOM (Form SFN-0201) and in the 7201-02 Computing Element FEMM (Form SFN-0203).

The diagrams in this manual are arranged into six categories:

- Category 1. Diagnostic Techniques
- Category 2. Overall Data Flow
- Category 3. Data Flow by Instruction Class
- Category 4. Functional Units
- Category 5. Operations
- Category 6. Manual Controls and Maintenance Facilities

All diagrams are in numerical order. The first digit of the diagram number reflects the category; for example, Diagram 4-210 belongs to Category 4, Functional Units. A category may be further subdivided into functional groups; for example, in Category 4, the diagrams have been grouped as follows:

- Group 1. Timing and Clock Control
- Group 2. ROS
- Group 3. Data and Control Registers
- Group 4. Local Storage
- Group 5. Serial and Parallel Adders
- Group 6. Status and Control Triggers
- Group 7. SCI

Prerequisite and companion manuals are:

Prerequisite Manuals

- 9020E System Introduction, Theory of Operation Manual, Form SFN-0103
- 9020D System Introduction, Theory of Operation Manual, Form SFN-0104

Companion Manuals

- 7201-02 Computing Element, Theory of Operation Manual, Form SFN-0201
- 7201-02 Computing Element, Maintenance Manual, Form SFN-0203
- 7201-02 Computing Element, Installation Manual, Form SFN-0204
- 7201-02 Parts Catalog, Form SFN-0205
- 9020 D/E Power Controls and Distribution, Theory of Operation Manual, Form SFN-0105.

First Edition (July, 1970)

This manual has been prepared by IBM Product Publications, Kingston, N.Y.

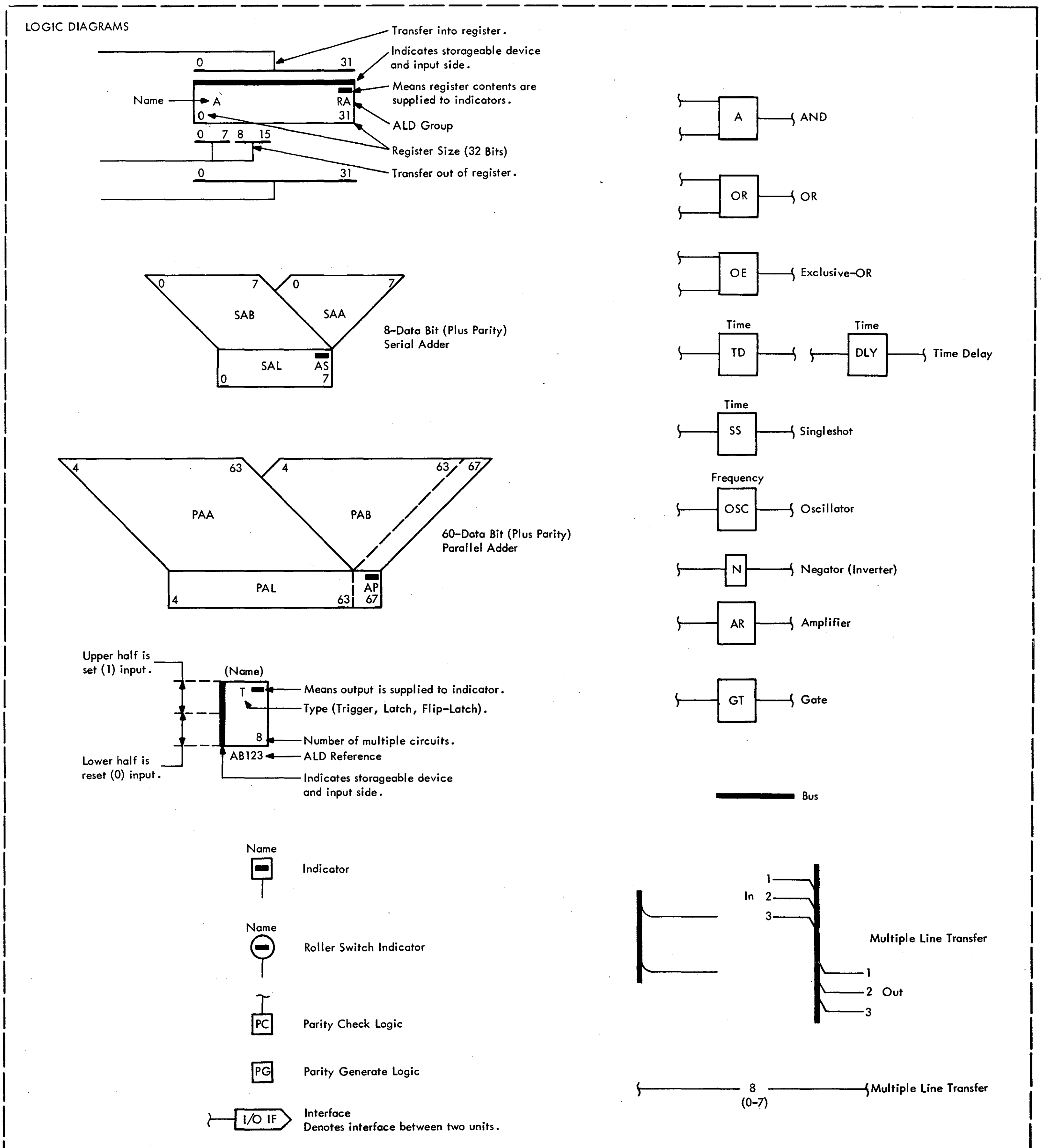
CONTENTS

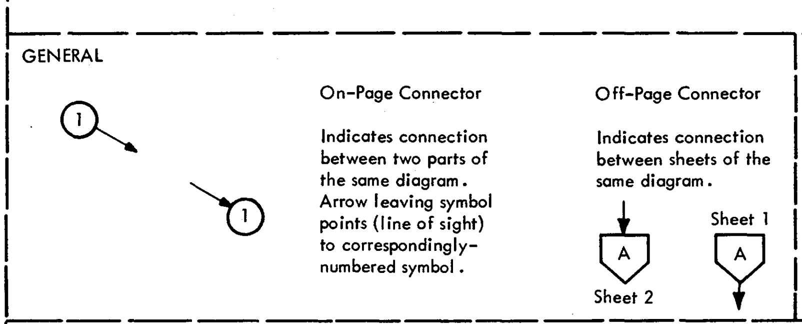
*Note: 1052 Adapter is used only with the 9020E configuration.

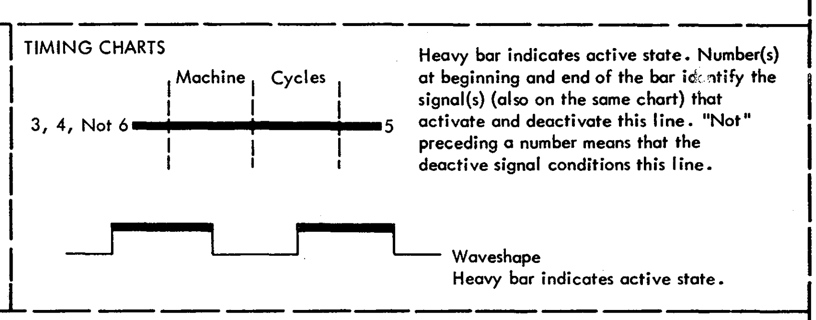
ABBREVIATIONS

ABC	AB register byte counter
ac	alternating current
ACR	Automatic Carrier Return
adr	address, addressed, addressing
ALD	automated logic diagram
ALTN	Alternate
amp	ampere
APSA	alternate preferential storage area
ASC	address store compare
ATC	air traffic control
ATN	alternate test number
ATR	address translation register
Attn	attention
Aux	Auxiliary Magnet
BCD	binary-coded decimal
BCU	bus control unit (alternate terminology for SCI)
BL	blink
BR	brightness
BSM	basic storage module
C	capacitor
CAS	control automation system
CAW	channel address word
CB	circuit breaker
CC	condition code, also Configuration Console
CCC	Central Computer Complex
CCR	configuration control register
CCW	channel command word
CE	Computing Element
Charistic	Characteristic
CLD	control automation system logic diagram
Cmd	command
CPU	Central Processing Unit (alternate terminology for CE)
CR	diode or Carrier Return
CROS	capacitive read-only storage
CSW	channel status word
CT	conditional terminate
CTC	channel-to-channel
CU	Control Unit
CVG	Character Vector Generator
DA	dash
DAR	diagnose accessible register
DARM	diagnose accessible register mask
DAU	Data Adapter Unit
dc	direct current
DCP	Display Channel Processor
DE	Display Element
dec	decimal
dec div	decimal divide
dec ovflo	decimal overflow
DG	Display Generator
Disc	disconnect
dly	delay
Dply	display
dsbl	disable
DX	first byte in a series of destination bytes
DX+1	second byte in a series of destination bytes
DX+2	third byte in a series of destination bytes
ELC	element check
end op	end operation
EOB	end of block
EOL	End-of-Line
EPO	emergency power off
ERSLT	expected result
EXC	Executive Control Program
exp ovflo	exponent overflow
exp unflo	exponent underflow
F	fuse
FEMDM	Field Engineering Maintenance Diagrams Manual
FEMI	Field Engineering Manual of Instruction
FEMM	Field Engineering Maintenance Manual
FETOM	Field Engineering Theory of Operation Manual
fix-pt ovflo	fixed-point overflow
FLT	fault locating test
flt-pt div	floating-point divide
FMTN	Format New
FMTO	Format Old
FMTW	Format Weather
FPR	Floating-point register
fract	fraction

GIS	general initialization sequence
GPR	general-purpose register
hex	hexadecimal
Hz	Hertz
IC	instruction counter
ICR	inhibit carrier return
IDES	inhibit display element stop
I-Fetch	instruction fetching
ILC	instruction length code
ILOS	inhibit logout stop
Init	initial
I/O	input/output
IOCE	Input/Output Control Element
IPL	initial program load
K	kilo; also relay
kHz	kilohertz
LAB	logical address bus
LADS	Logic Automation Documentation System
LAL	local storage address latches
LAR	local storage address register
LC	lower case
LF	line feed
LOS	logout stop
LS	local store
LSWR	local storage working register
MACH	maintenance and channel (storage)
max	maximum
MC	machine check
MCW	maintenance control word
mHz	megahertz
MMSC	maintenance mode stop clock
Mple	Multiple
MPR	multiplier
MPX	multiplex
ms	millisecond
NDT	new descriptor tables
no op	no operation
NRM	new refresh memory
NRMA	new refresh memory address
ns	nanosecond
OBS	on battery signal
ODT	old descriptor tables
op code	operation code
oper	operation
opr	operand
ORM	old refresh memory
ORMA	old refresh memory address
OTC	out of tolerance check
P	parity
PAA	parallel adder A-side
PAB	parallel adder B-side
PAL	parallel adder latch
PB	pushbutton
pf	picofarad
PK	power contactor
PP	partial product
PQ	partial quotient
priv oper	privileged operation
proc	process
prog	program
PROSAR A	previous read-only storage address register A
PROSAR B	previous read-only storage address register B
prot	protection
PS	power supply
PSA	preferential storage address
PSBA	preferential storage base address
PSBAR	preferential storage base address register
PSW	program status word
PVD	Plan View Display
R	resistor
RCU	Reconfiguration Control Unit
reg	register
RKM	Radar Keyboard Multiplexor
ROS	read-only storage
ROSAR	read-only storage address register
ROSBR	read-only storage backup register
ROSDR	read-only storage data register
RST	Reset
SAA	serial adder A-side
SAB	storage address bus, also serial adder B-side
SAL	serial adder latch
SATR	set Address Translation Register
SBA	serial adder bus A
SBB	serial adder bus B
SC	System Console
SCI	storage control interface
SCON	set Configuration Control Register
SCOPEX	scoping index
SCR	silicon-controlled rectifier
SDBI	storage data bus in
SDBO	storage data bus out
SE	Storage Element
Sel	select
Serv	service
signif	significance
SLT	solid logic technology
SMMC	system maintenance monitor console
SMS	standard modular system
SOROS	scan out read-only storage
spec	specification
SRL	Systems Reference Library
SSU	storage switching unit
STAT	status trigger
STC	ST register byte counter
stg	storage
SU	switch unit
sync	synchronizing
T	transformer
TC	time clock (interval timer)
TCU	tape control unit
T(DX)	table byte specified by DX
T(DX+1)	table byte specified by DX+1
TIC	transfer in channel
TN	test number
T/R	tilt/rotate
TU	tape unit
uc	upper case
uf	microfarad
usec	microsecond
UT	unconditional terminate
V	volt
VFL	variable-field length
VFR	visual flight rules
Xlat	translate
$\geq$	greater than or equal to
$\geqq$	greater than or equal to
$\leq$	less than or equal to
$\leqq$	less than or equal to
=	equal to
$\neq$	not equal to
&	and


LOGIC DIAGRAMS
Transfer into register.
Indicates storageable device and input side.
Means register contents are supplied to indicators.
ALD Group
Register Size (32 Bits)
Transfer out of register.
Name
A
RA
0
31
0 7 8 15
SAB
SAA
SAL
AS
8-Data Bit (Plus Parity) Serial Adder
PAA
PAB
PAL
AP
60-Data Bit (Plus Parity) Parallel Adder
Upper half is set (1) input.
Lower half is reset (0) input.
(Name)
T
8
AB123
Means output is supplied to indicator.
Type (Trigger, Latch, Flip-Latch).
Number of multiple circuits.
ALD Reference
Indicates storageable device and input side.
Name
Indicator
Name
Roller Switch Indicator
PC
Parity Check Logic
PG
Parity Generate Logic
I/O IF
Interface
Denotes interface between two units.
A
AND
OR
OR
OE
Exclusive-OR
Time
TD
Time
DLY
Time Delay
Time
SS
Singleshot
Frequency
OSC
Oscillator
N
Negator (Inverter)
AR
Amplifier
GT
Gate
Bus
In 1 2 3
Multiple Line Transfer
1 2 Out 3
8
(0-7)
Multiple Line Transfer


GENERAL
1
On-Page Connector
Indicates connection between two parts of the same diagram. Arrow leaving symbol points (line of sight) to correspondingly-numbered symbol.
Off-Page Connector
Indicates connection between sheets of the same diagram.
Sheet 1
A
Sheet 2


TIMING CHARTS
Machine Cycles
3, 4, Not 6
5
Heavy bar indicates active state. Number(s) at beginning and end of the bar identify the signal(s) (also on the same chart) that activate and deactivate this line. "Not" preceding a number means that the deactive signal conditions this line.
Waveshape
Heavy bar indicates active state.

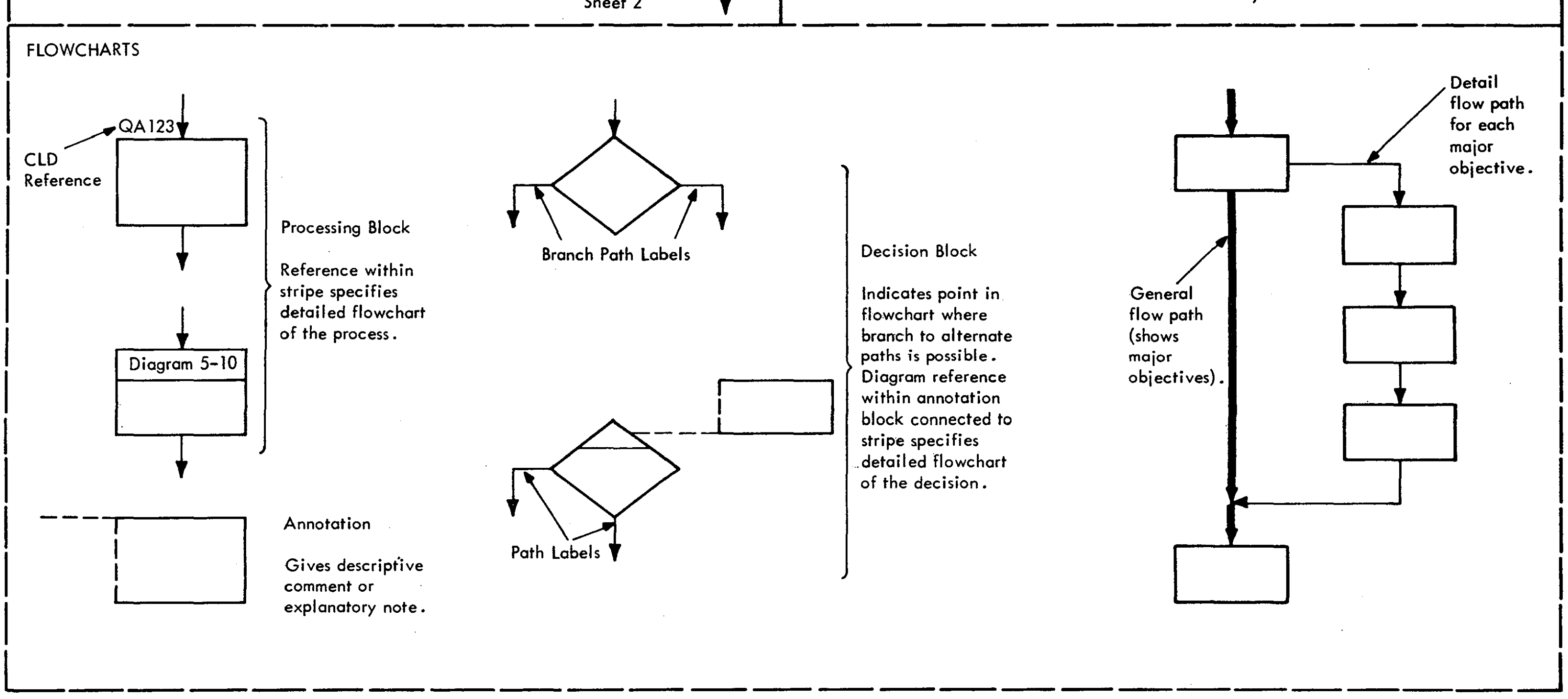

FLOWCHARTS
CLD Reference
QA123
Processing Block
Reference within stripe specifies detailed flowchart of the process.
Diagram 5-10
Annotation
Gives descriptive comment or explanatory note.
Branch Path Labels
Path Labels
Decision Block
Indicates point in flowchart where branch to alternate paths is possible. Diagram reference within annotation block connected to stripe specifies detailed flowchart of the decision.
Detail flow path for each major objective.
General flow path (shows major objectives).

Diagram 1-1. Maintenance Strategy Diagram (Sheet 1 of 2)

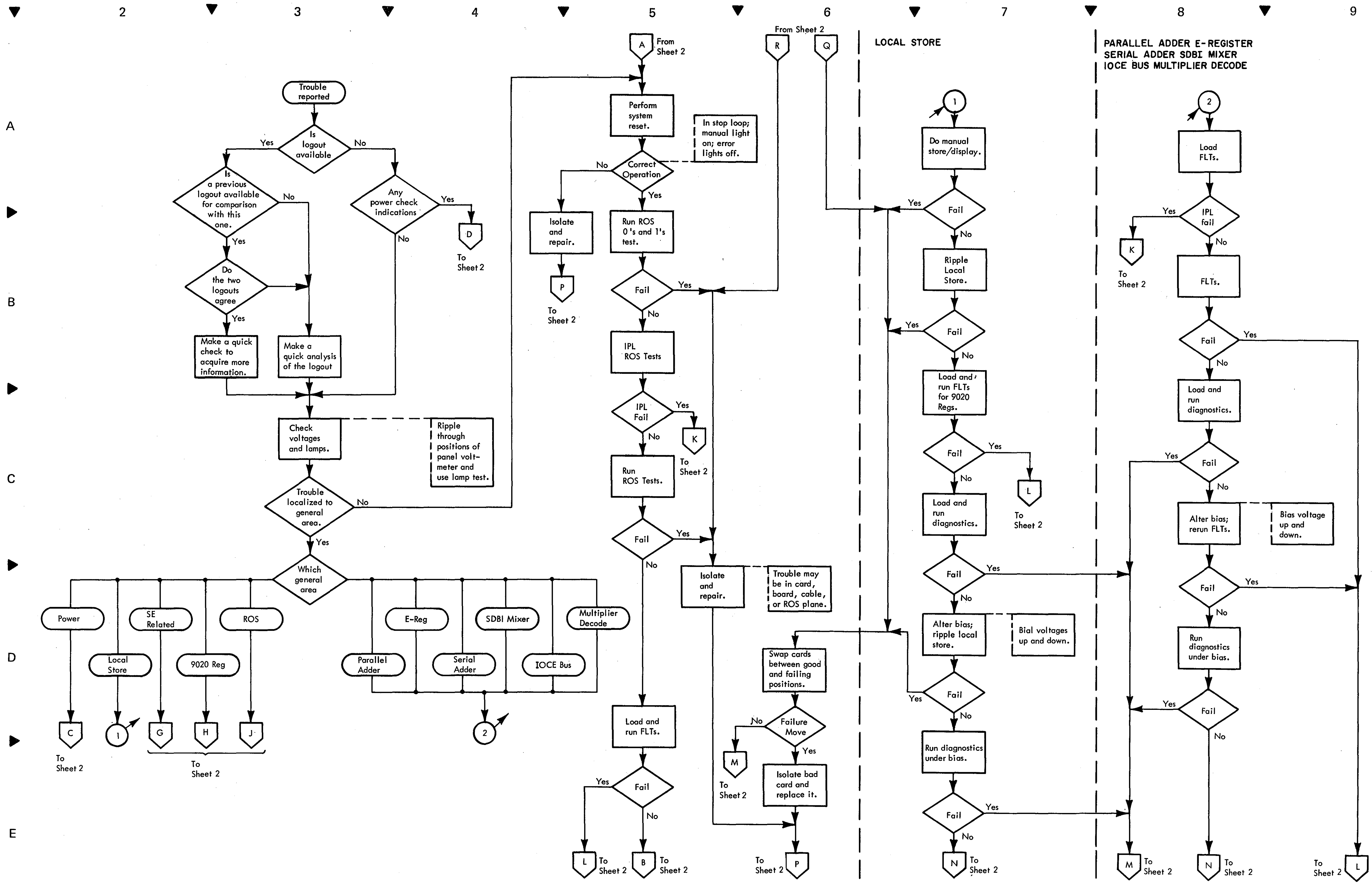
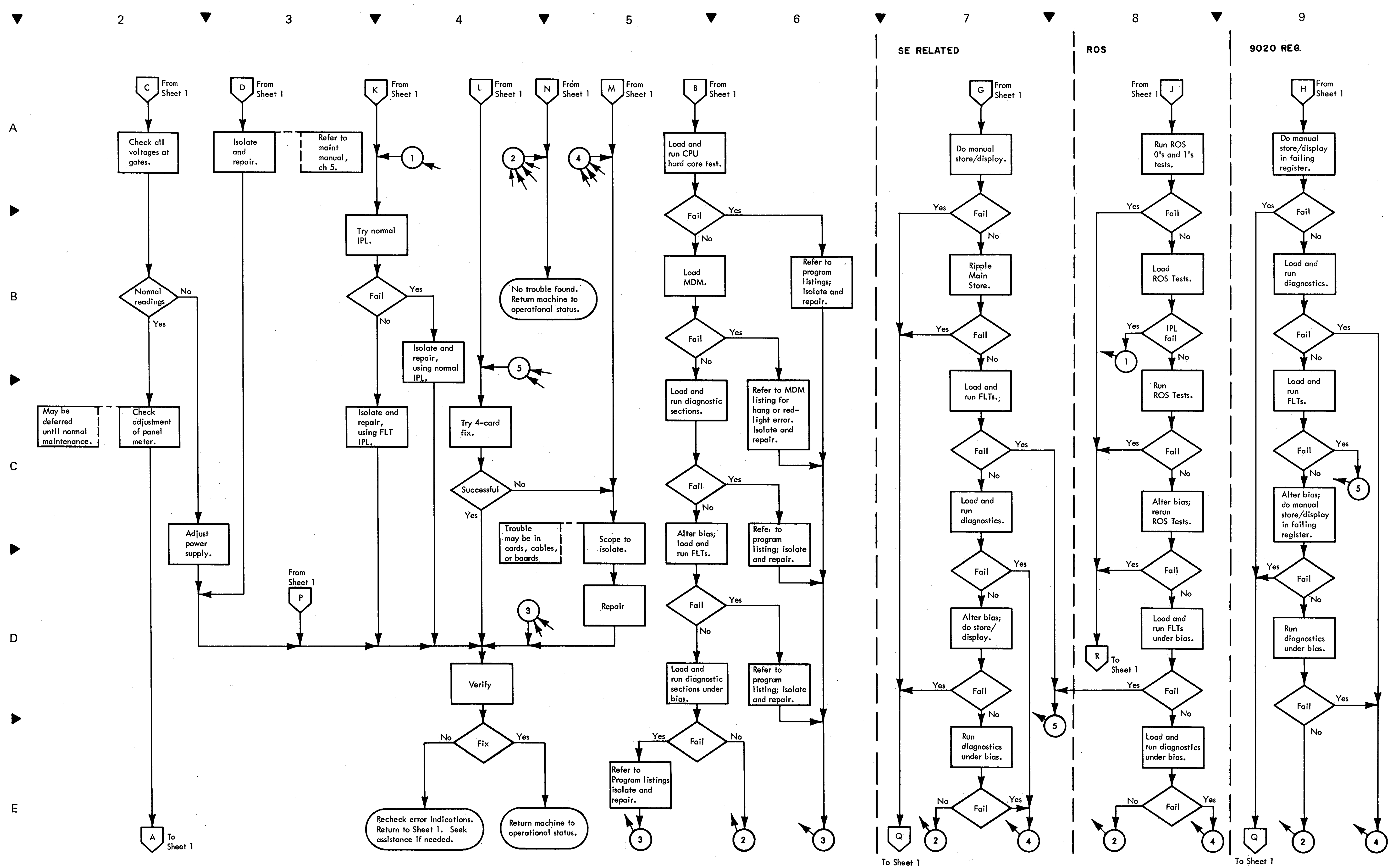


Diagram 1-1. Maintenance Strategy Diagram (Sheet 2 of 2)



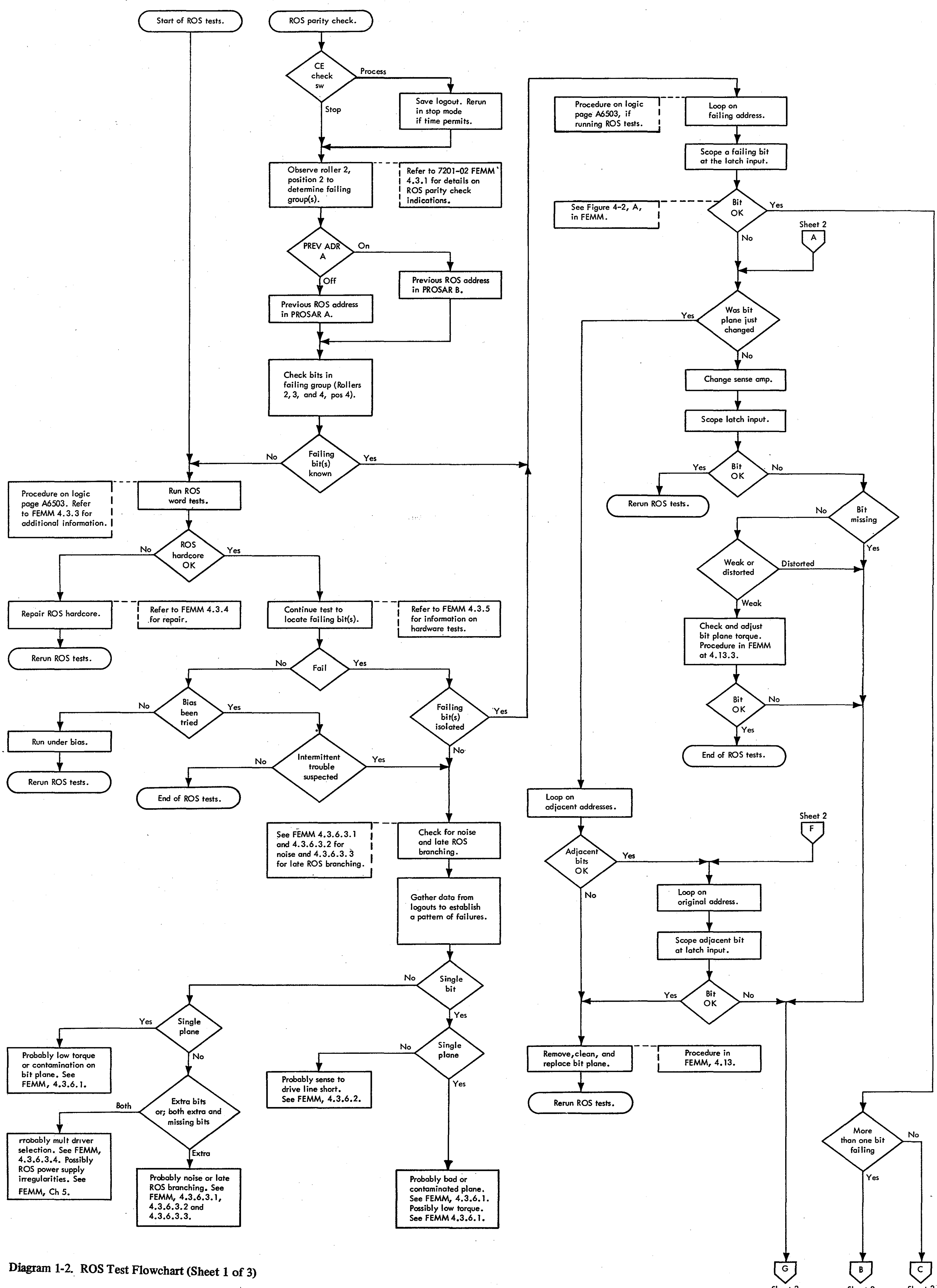


Diagram 1-2. ROS Test Flowchart (Sheet 1 of 3)

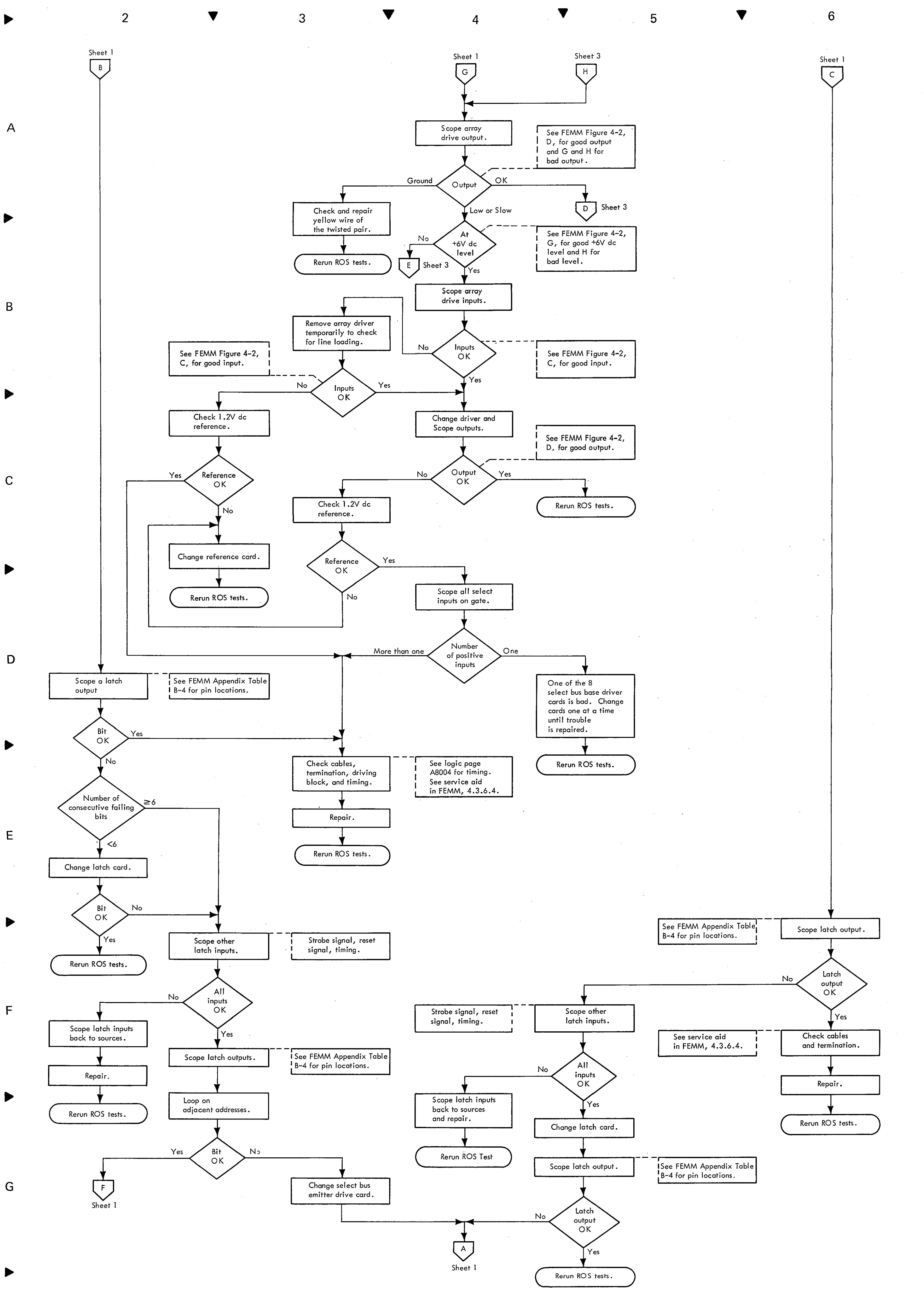
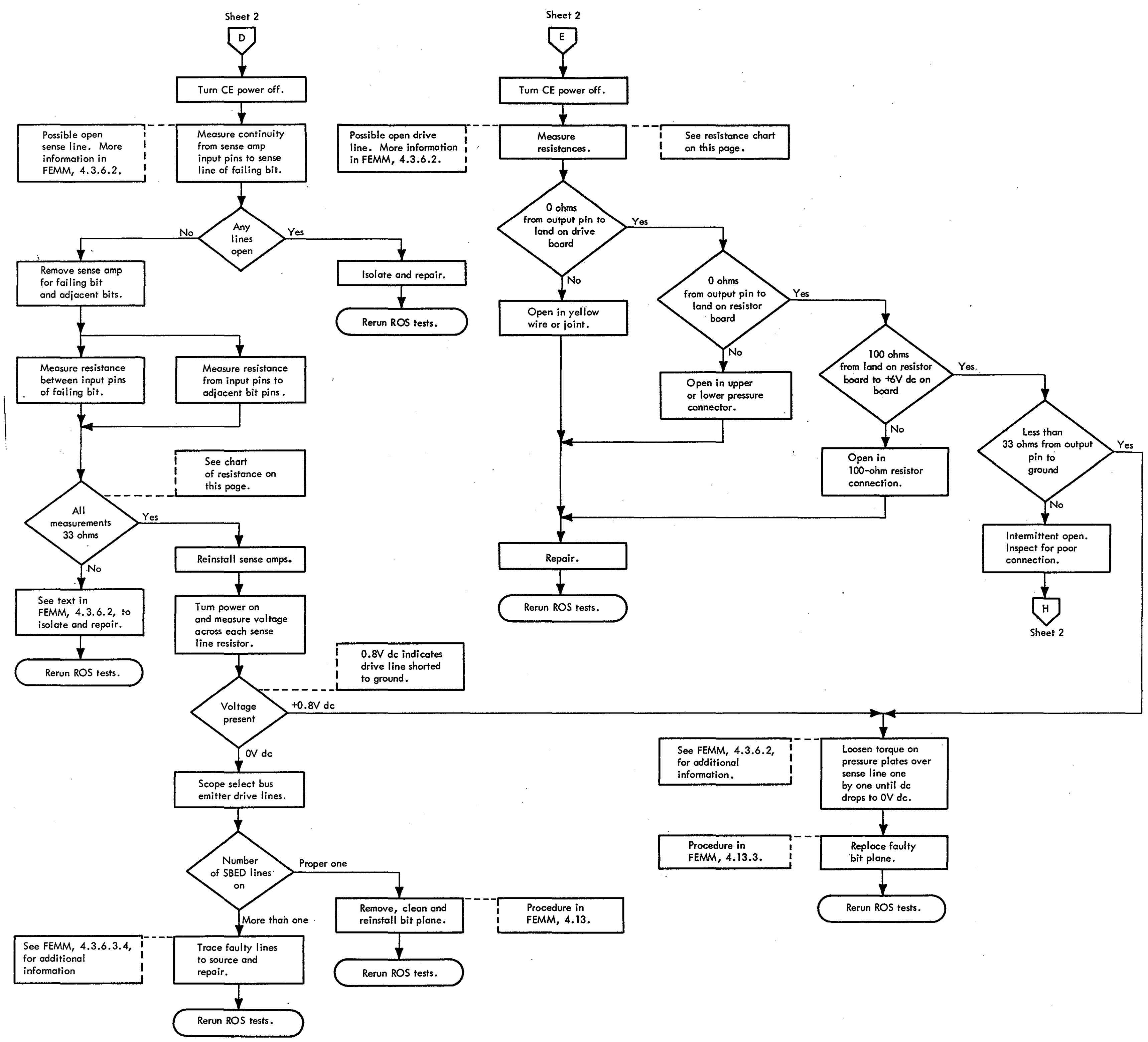


Diagram 1-2. ROS Test Flowchart (Sheet 2 of 3)



Resistance Readings

Condition	From Sense Amp Input to	Resistance
Normal	DC return. Any other sense amp input. Any drive/balance line.	16.5 ohms 33.0 ohms Open
Sense - sense line short	DC return. Input of sense amp to which it is shorted. Any other sense amp input. Any drive/balance line.	8.25 ohms 0 ohm 24.75 ohms Open
Sense - DC return short	DC return. Any other sense amp input. Any drive/balance line.	0 ohm 16.5 ohms Open
Sense - drive/balance line short	DC return. Drive/balance line to which it is shorted. Any drive/balance line in same or opposite plane, except line to which it is shorted. Any drive/balance line not in same or opposite plane.	16.5 ohms 0 ohm 200 ohms plus short resistance Open

Diagram 1-2. ROS Test Flowchart (Sheet 3 of 3)

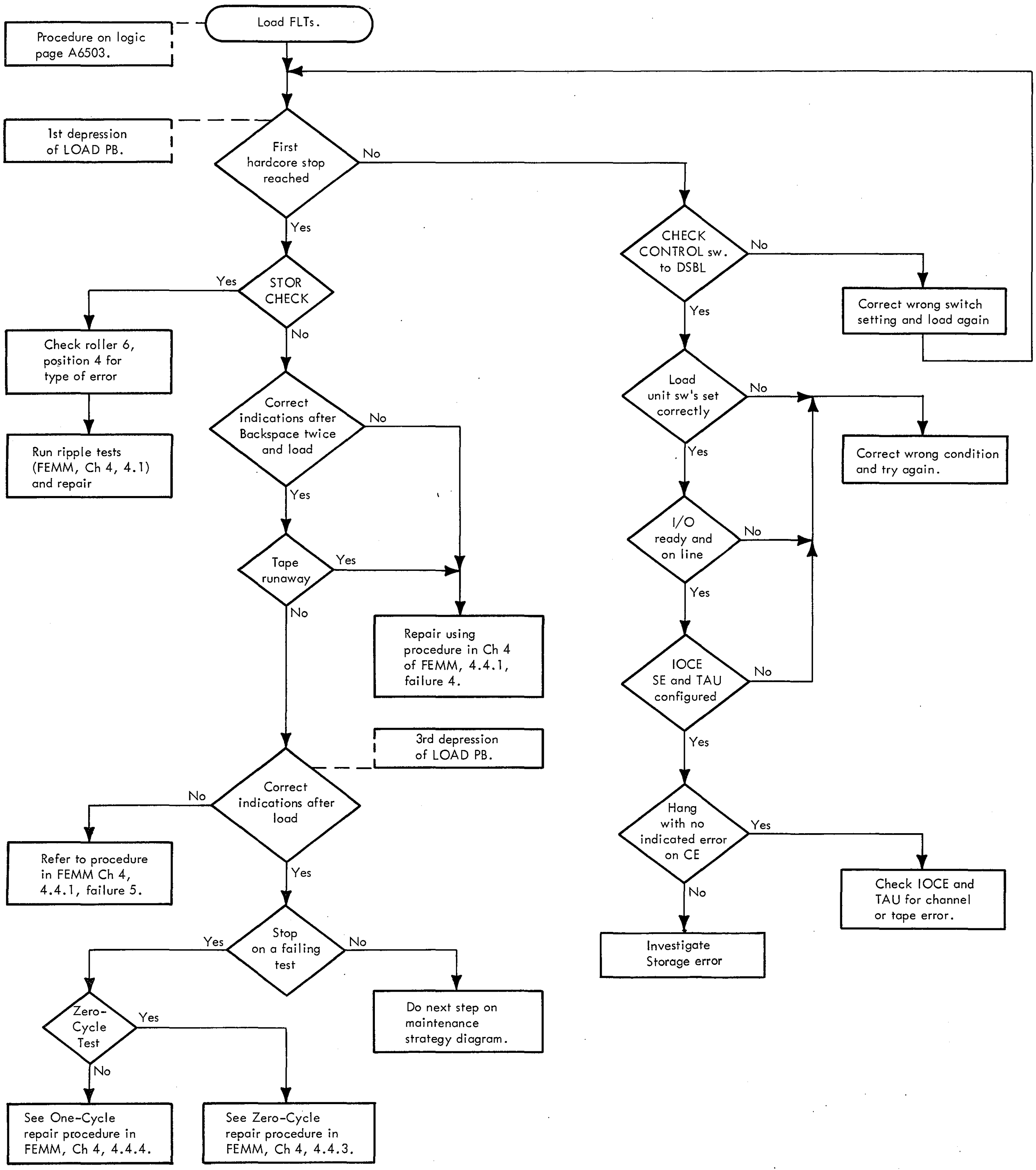


Diagram 1-3. FLT Flowchart

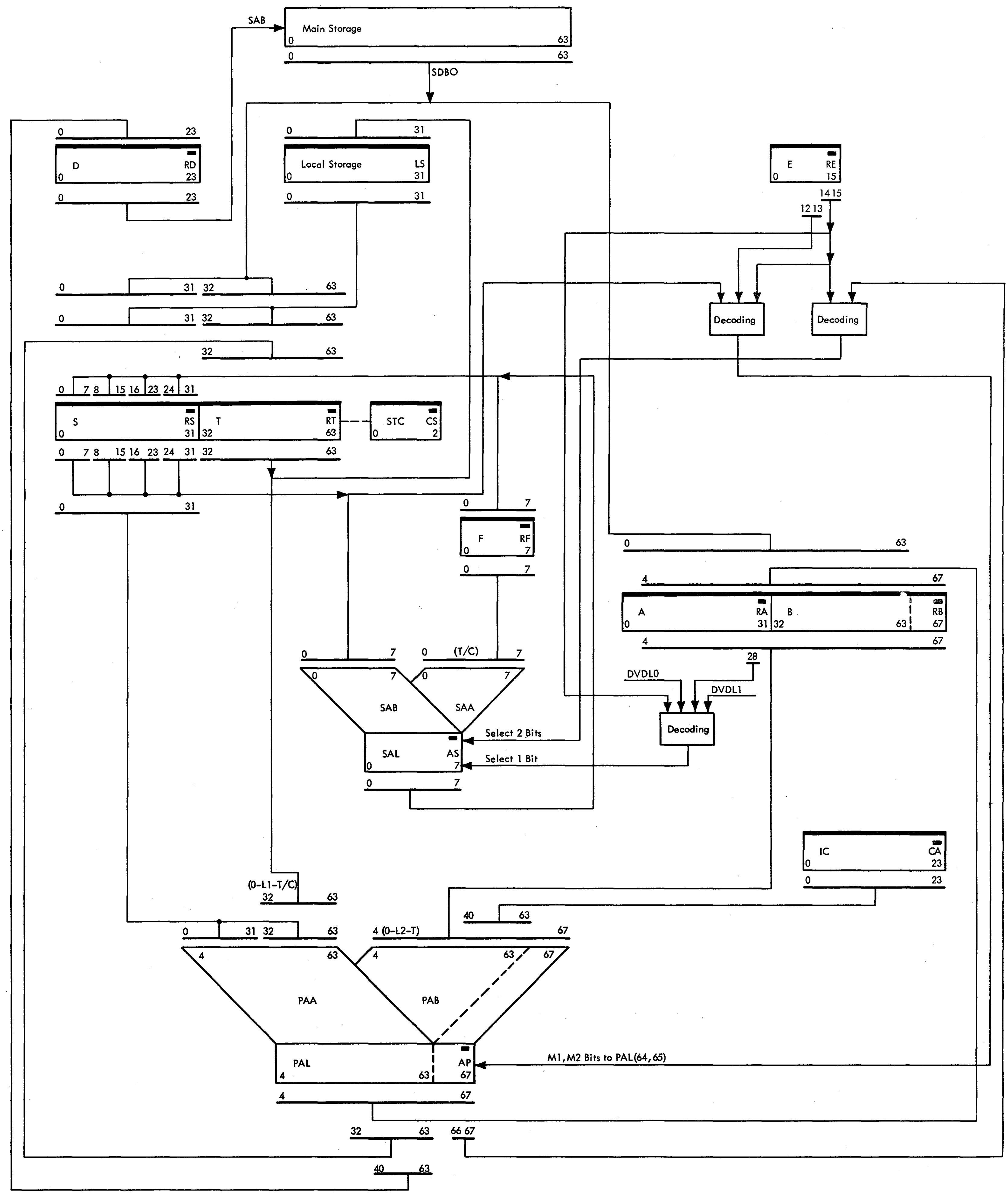


Diagram 3-1. Fixed Point Instruction Data Flow

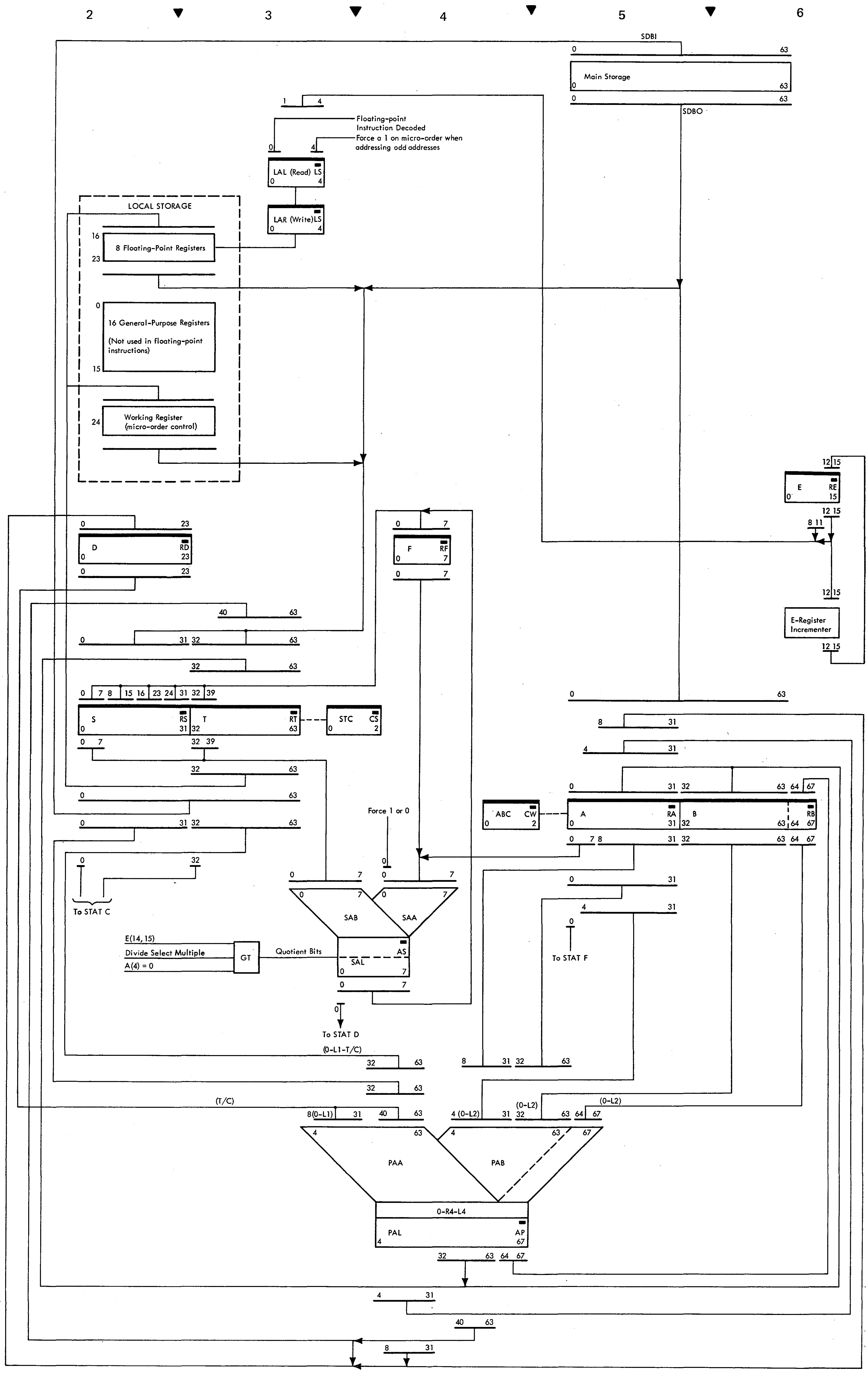


Diagram 3-2. Floating-Point Instruction Data Flow

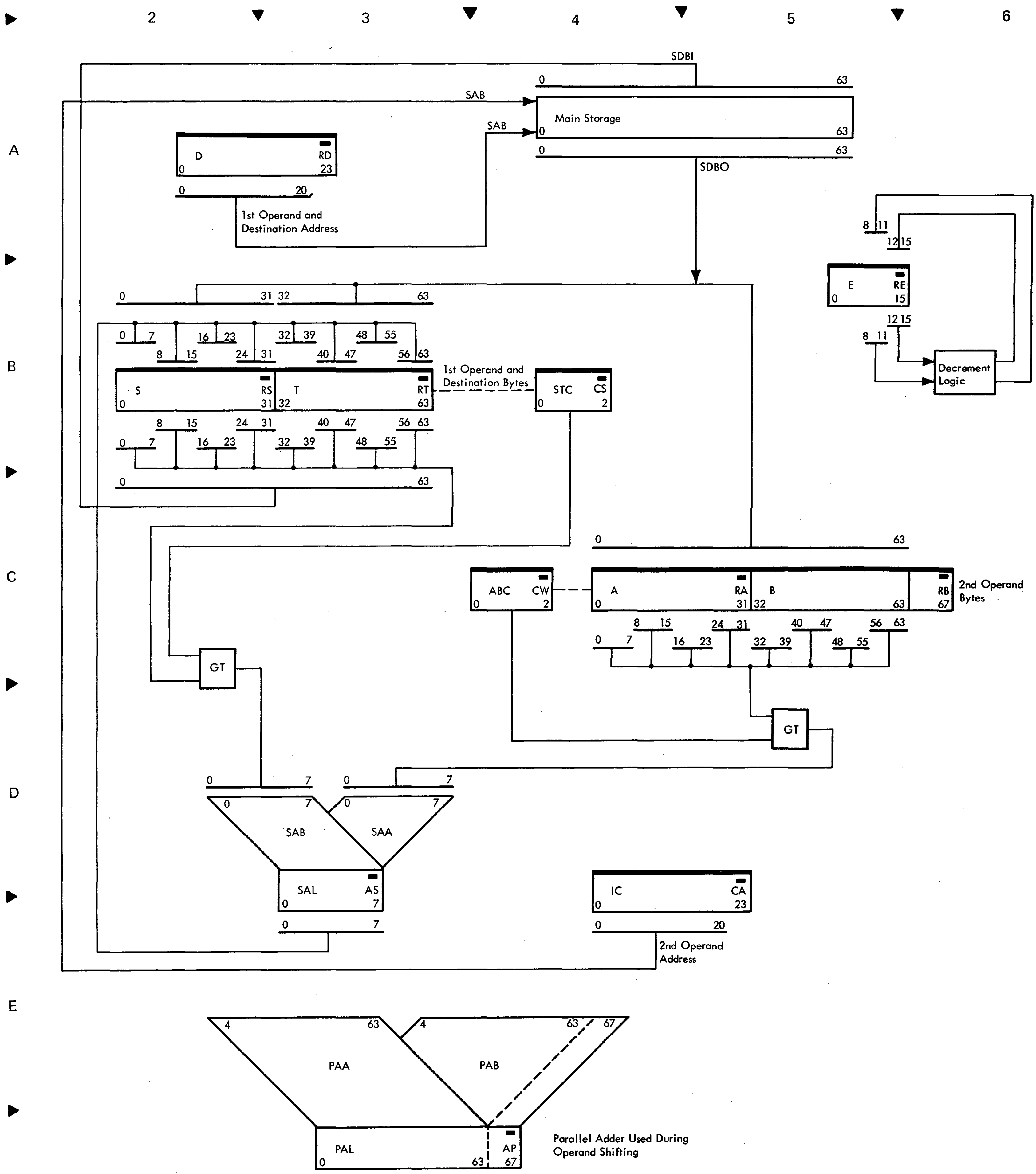


Diagram 3-3. Decimal and Logical Instruction Data Flow

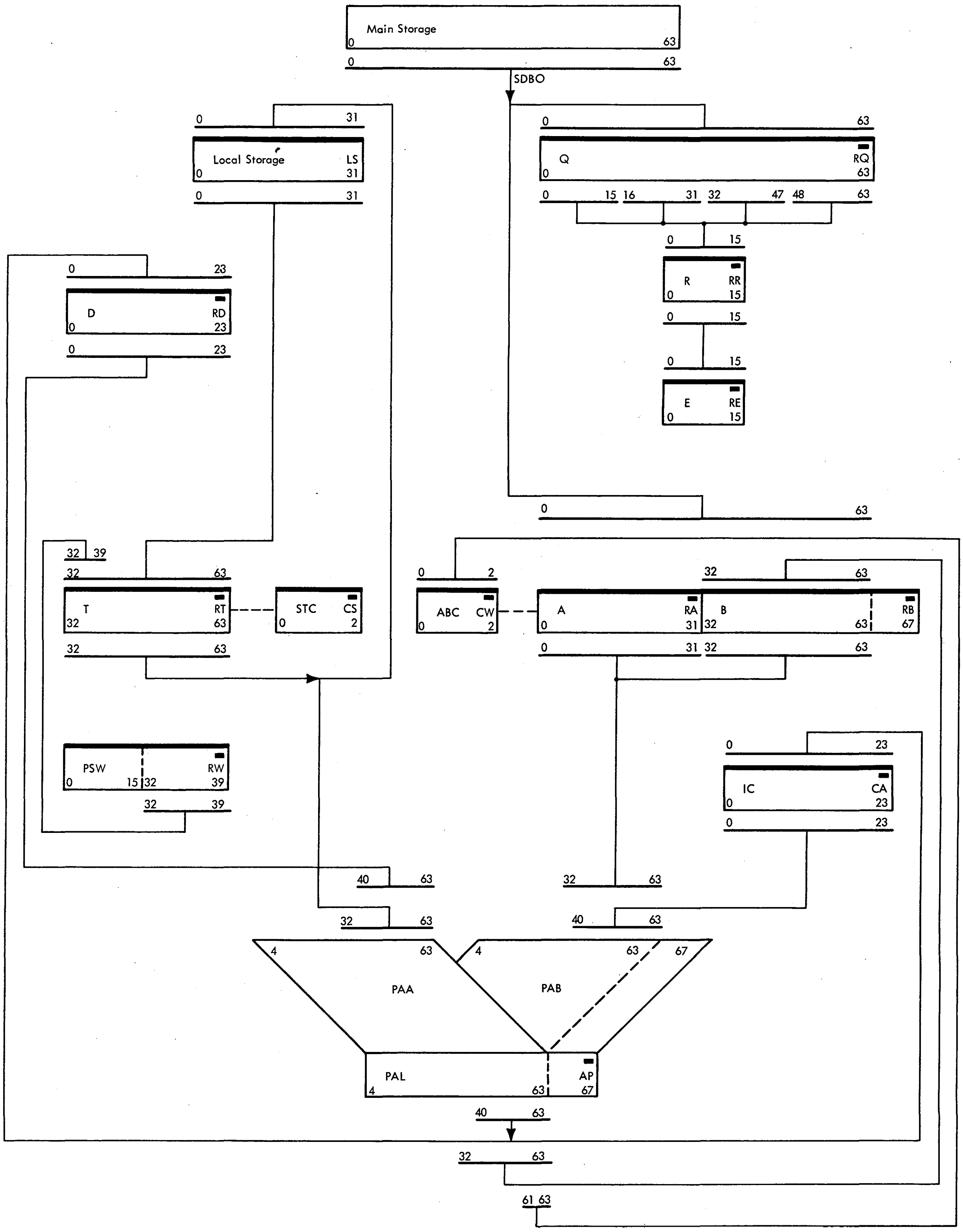


Diagram 3-4. Branching Instruction Data Flow

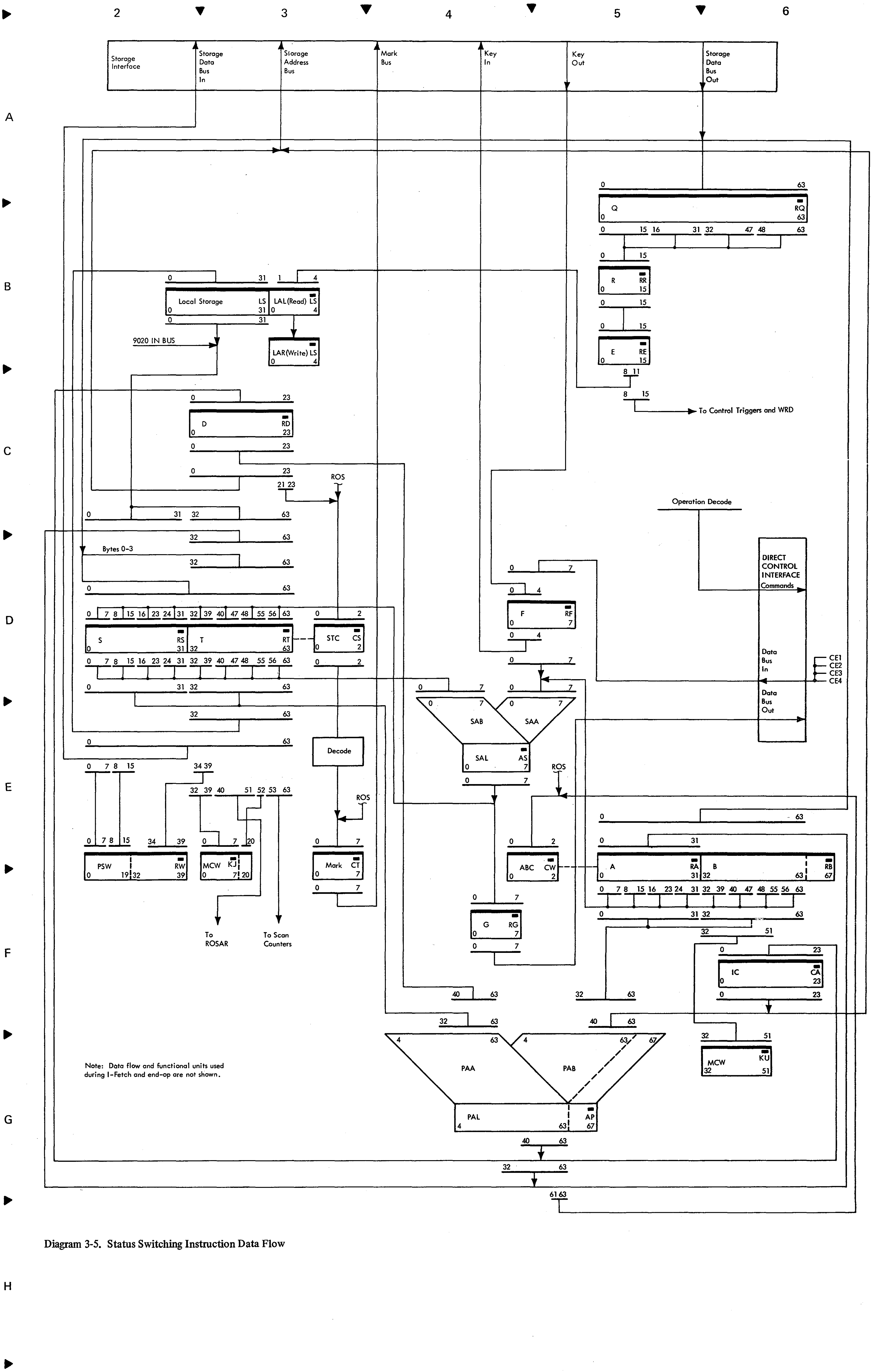


Diagram 3-5. Status Switching Instruction Data Flow

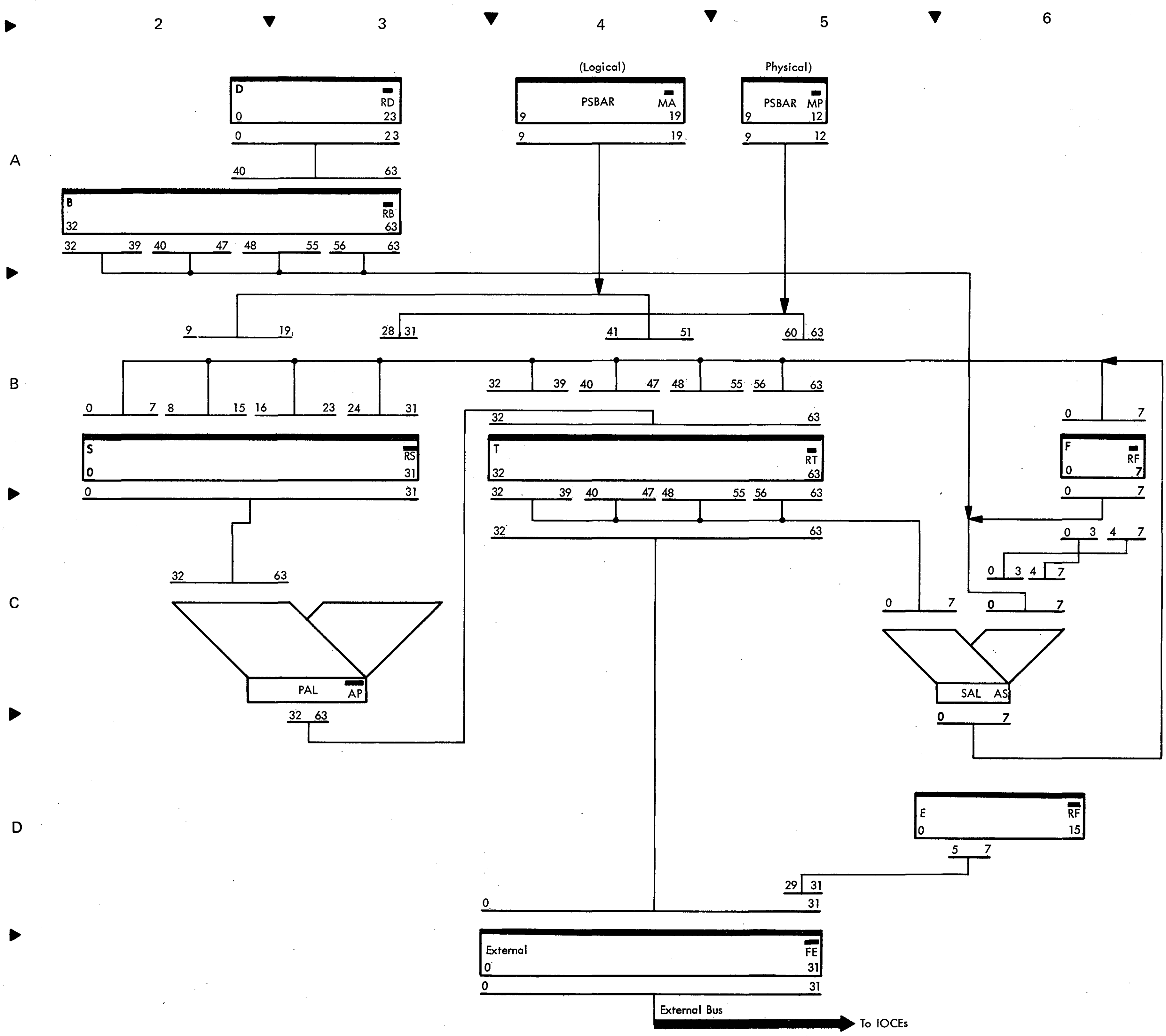


Diagram 3-6. Input/Output Instruction Data Flow

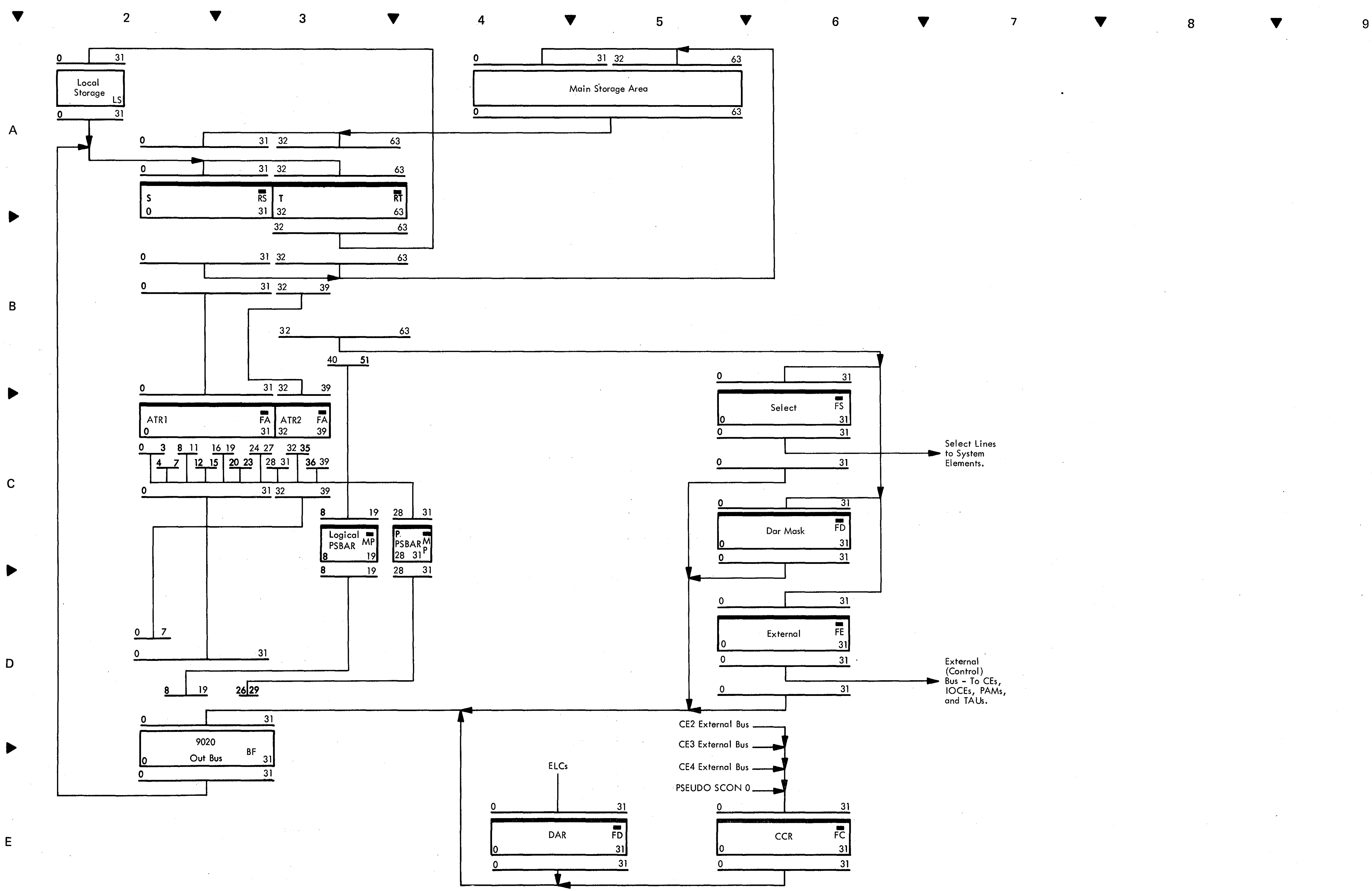


Diagram 3-7. Multiple Computing Element Instruction Data Flow

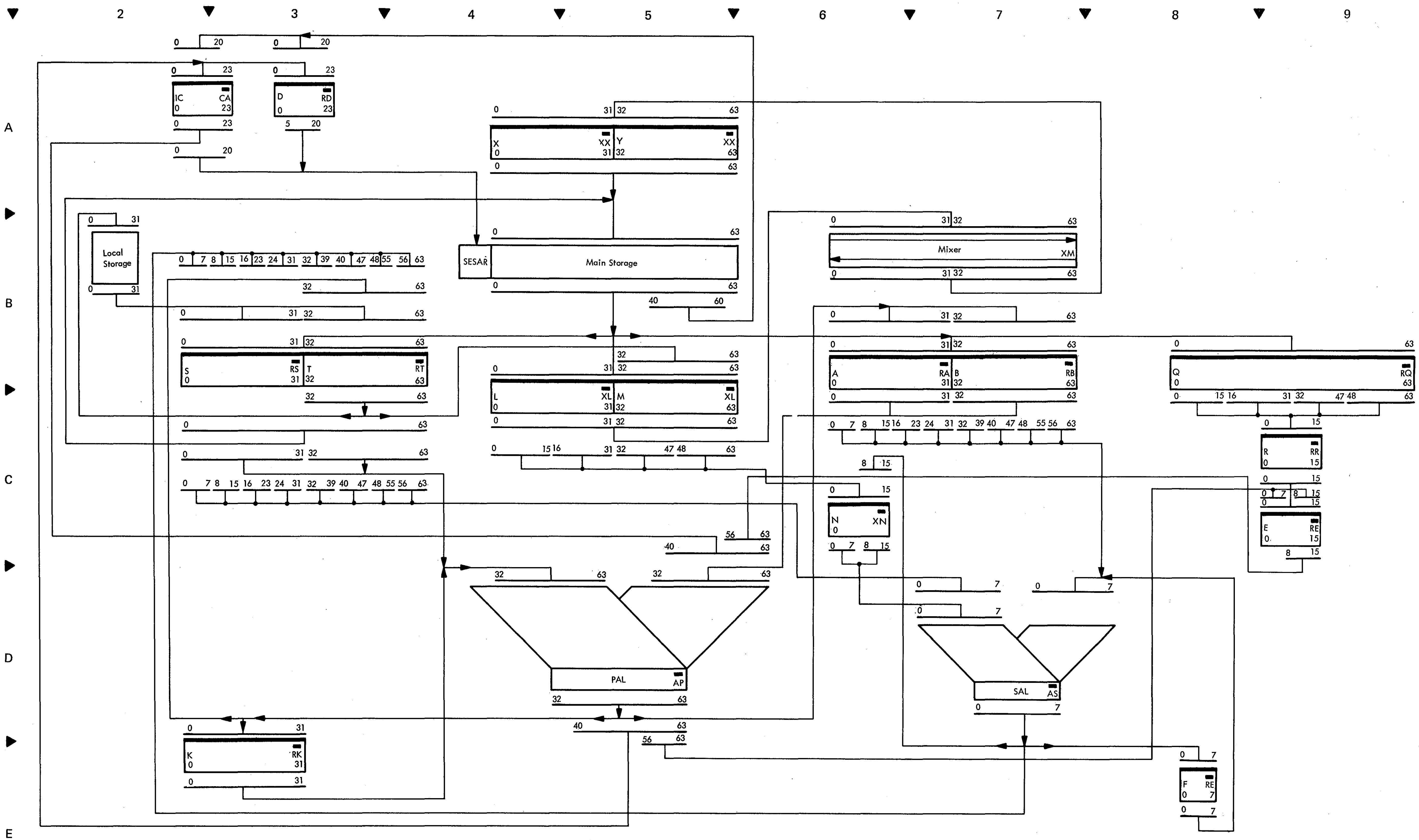


Diagram 3-8. Display Instruction Data Flow

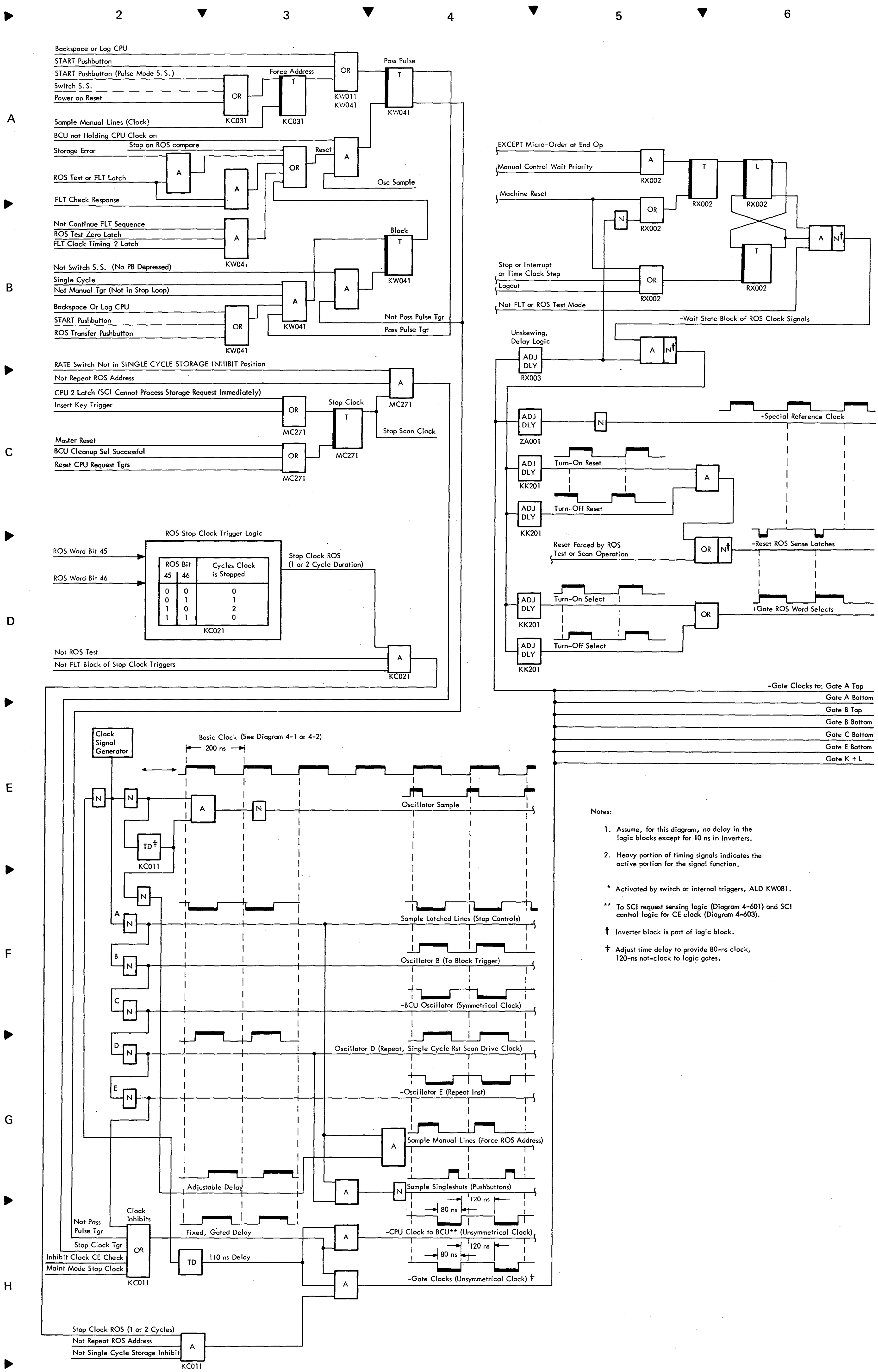
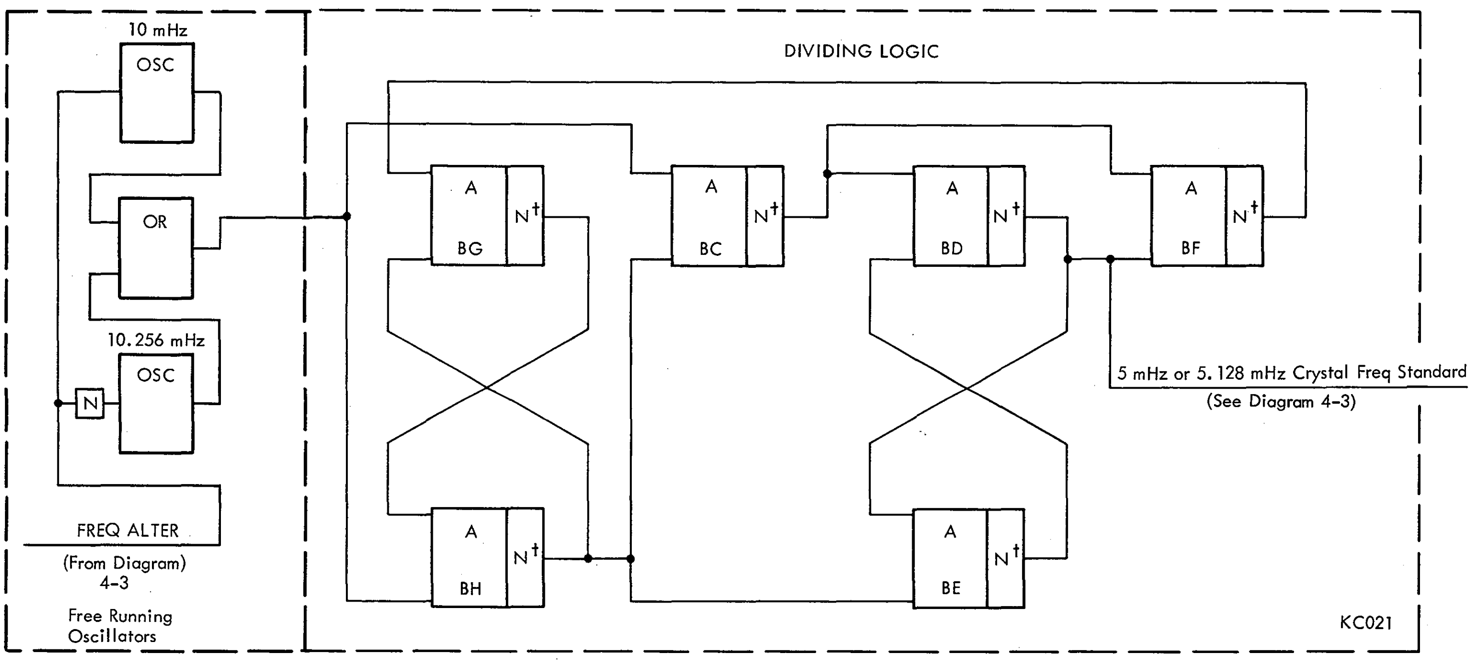
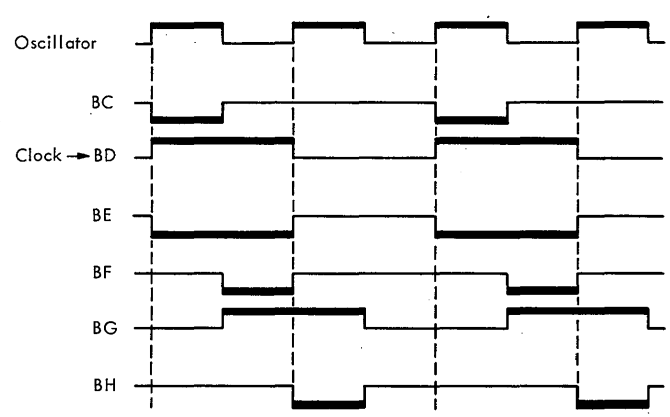


Diagram 4-1. Clock Control Logic



Ideal Waveform At Output Of Each AND After Inversion:



Notes:

1. Heavy portion of timing signals indicates the active portion for the signal function.
2. The two letter notation within the AND's is the block serial number on ALD KC021.

†Inverter block is part of logic block.

Diagram 4-2. Reference Oscillator

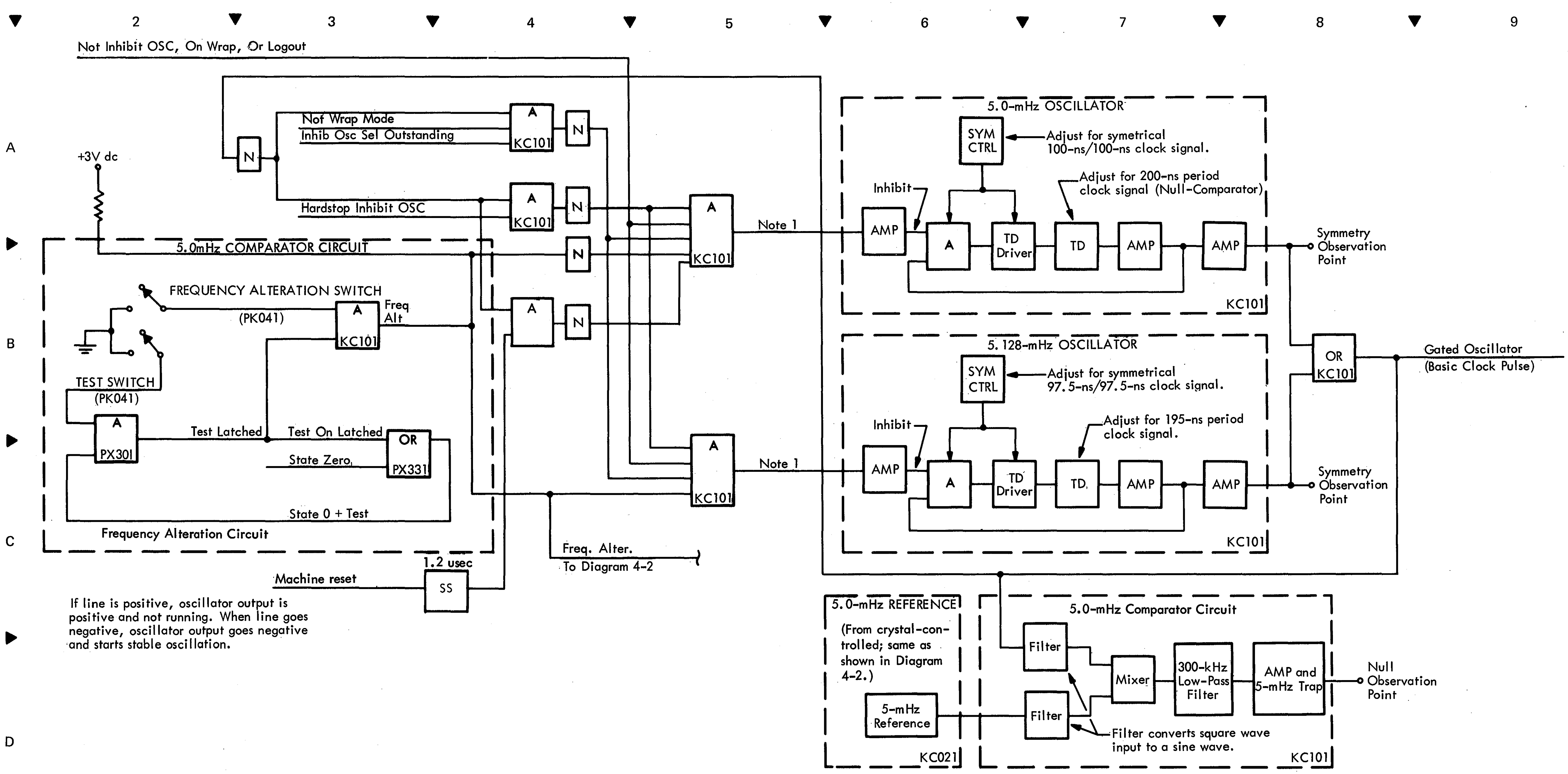


Diagram 4-3. CE Clock Signal Generator

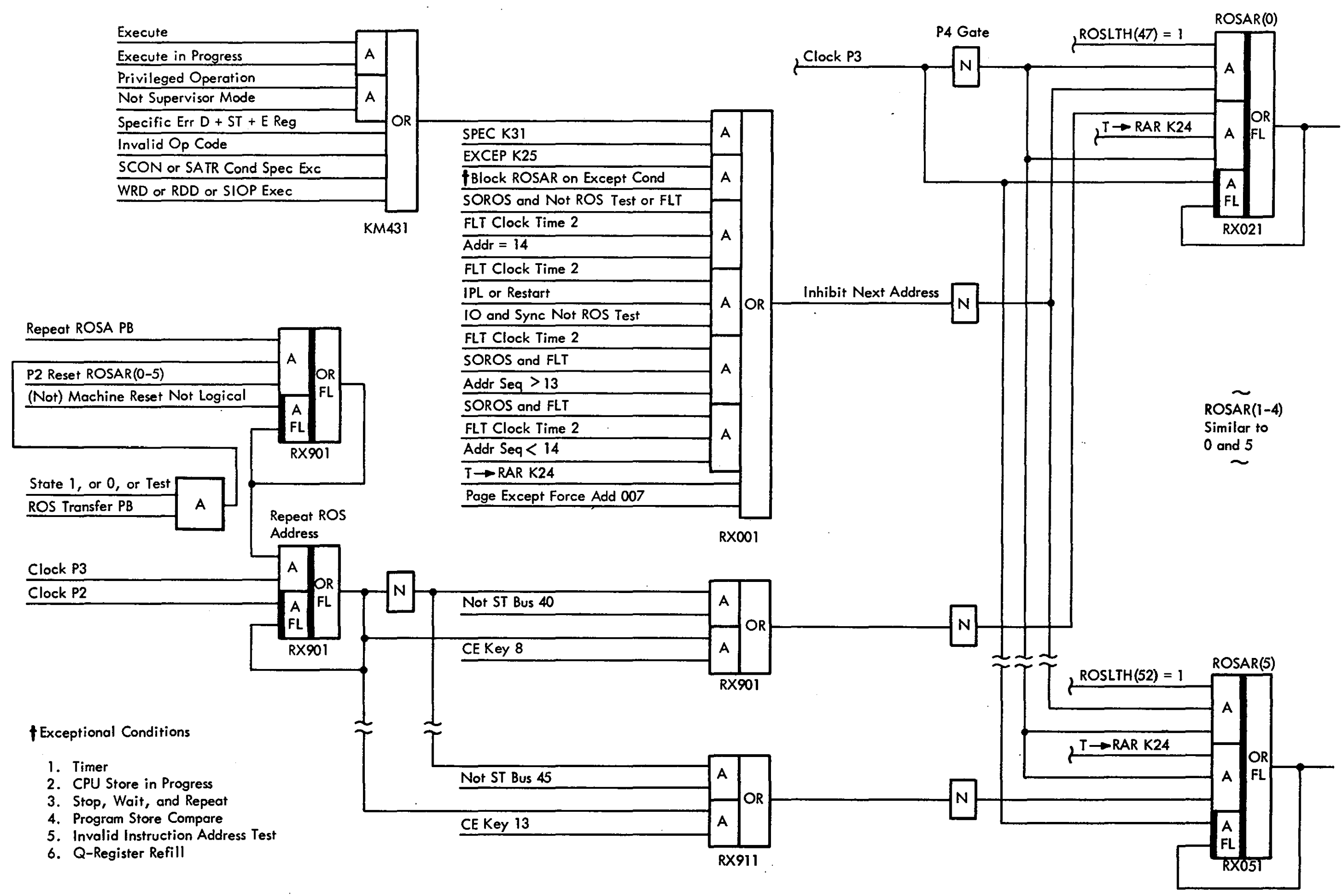


Diagram 4-101. ROSAR (0-5) Logic

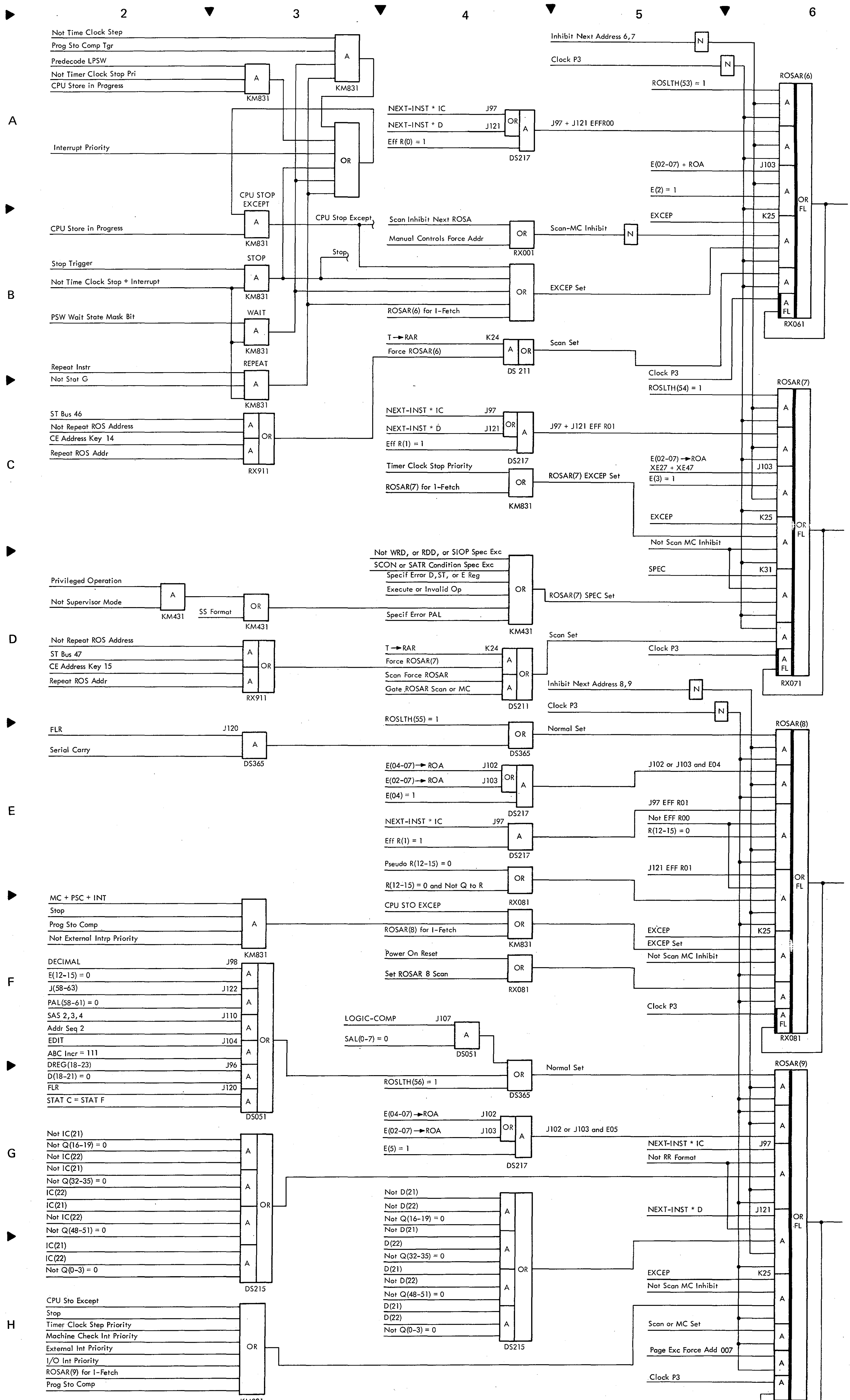


Diagram 4-102. ROSAR (6-9) Logic

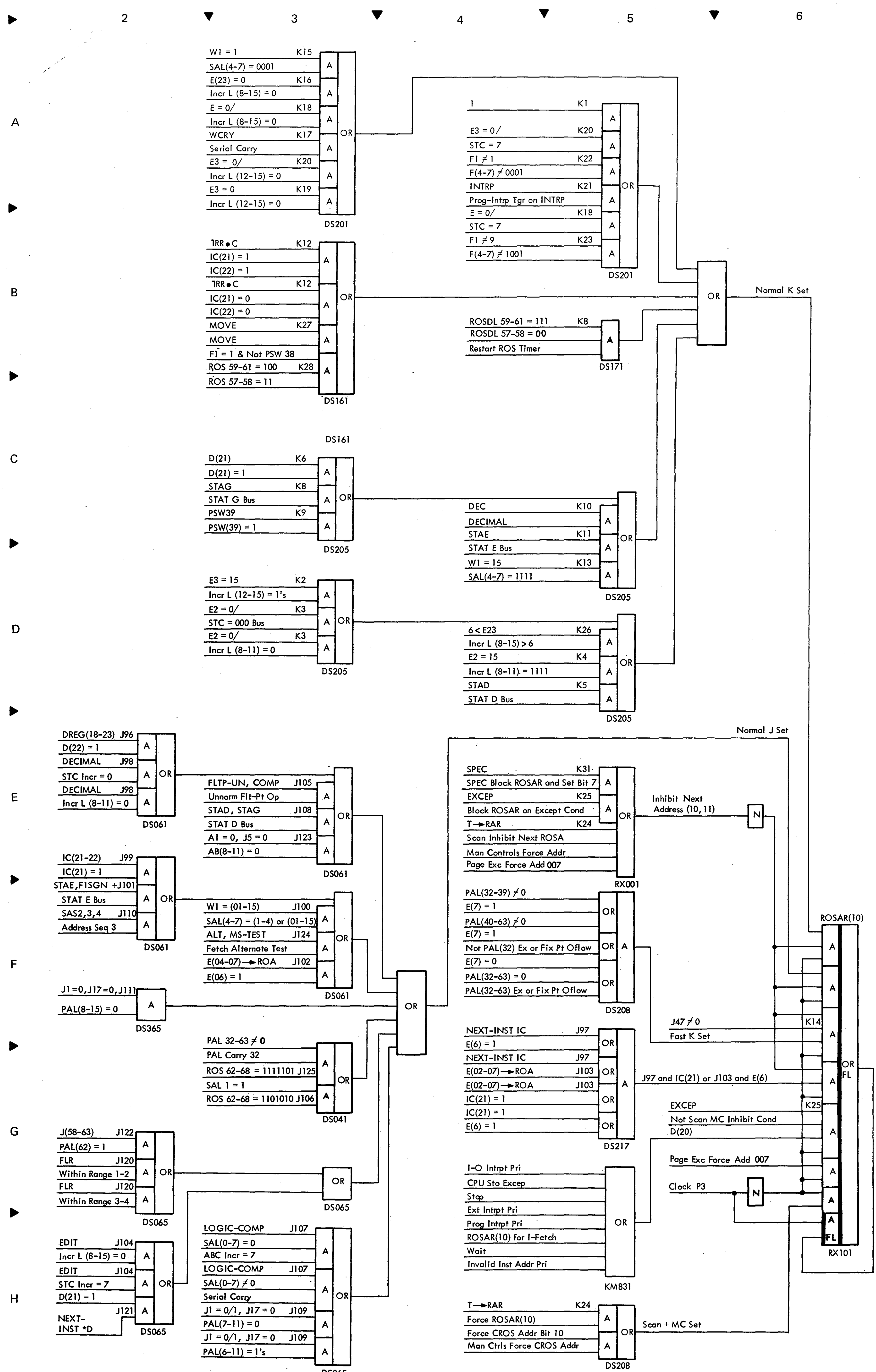


Diagram 4-103. ROSAR (10) Logic

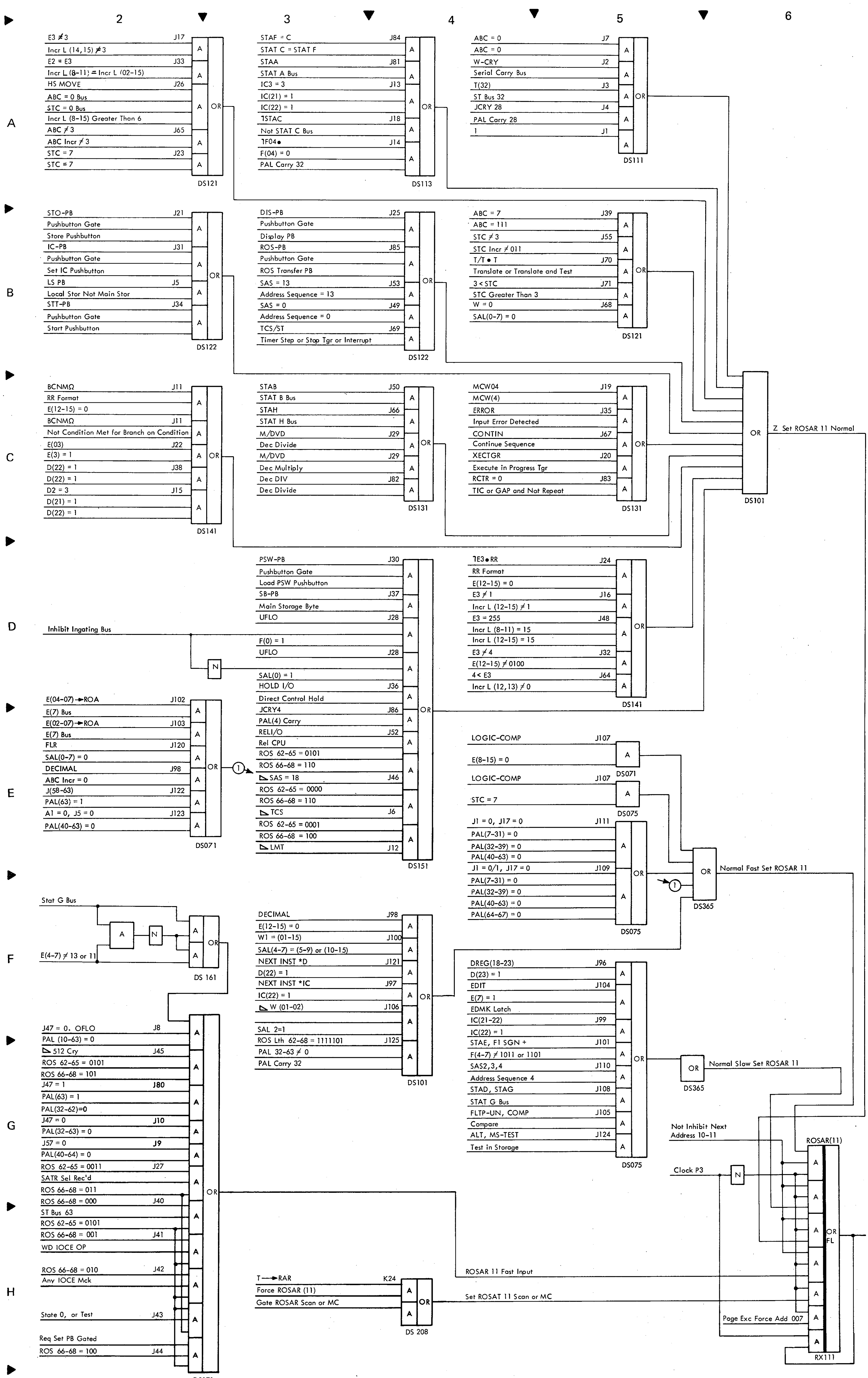


Diagram 4-104. ROSAR (11) Logic

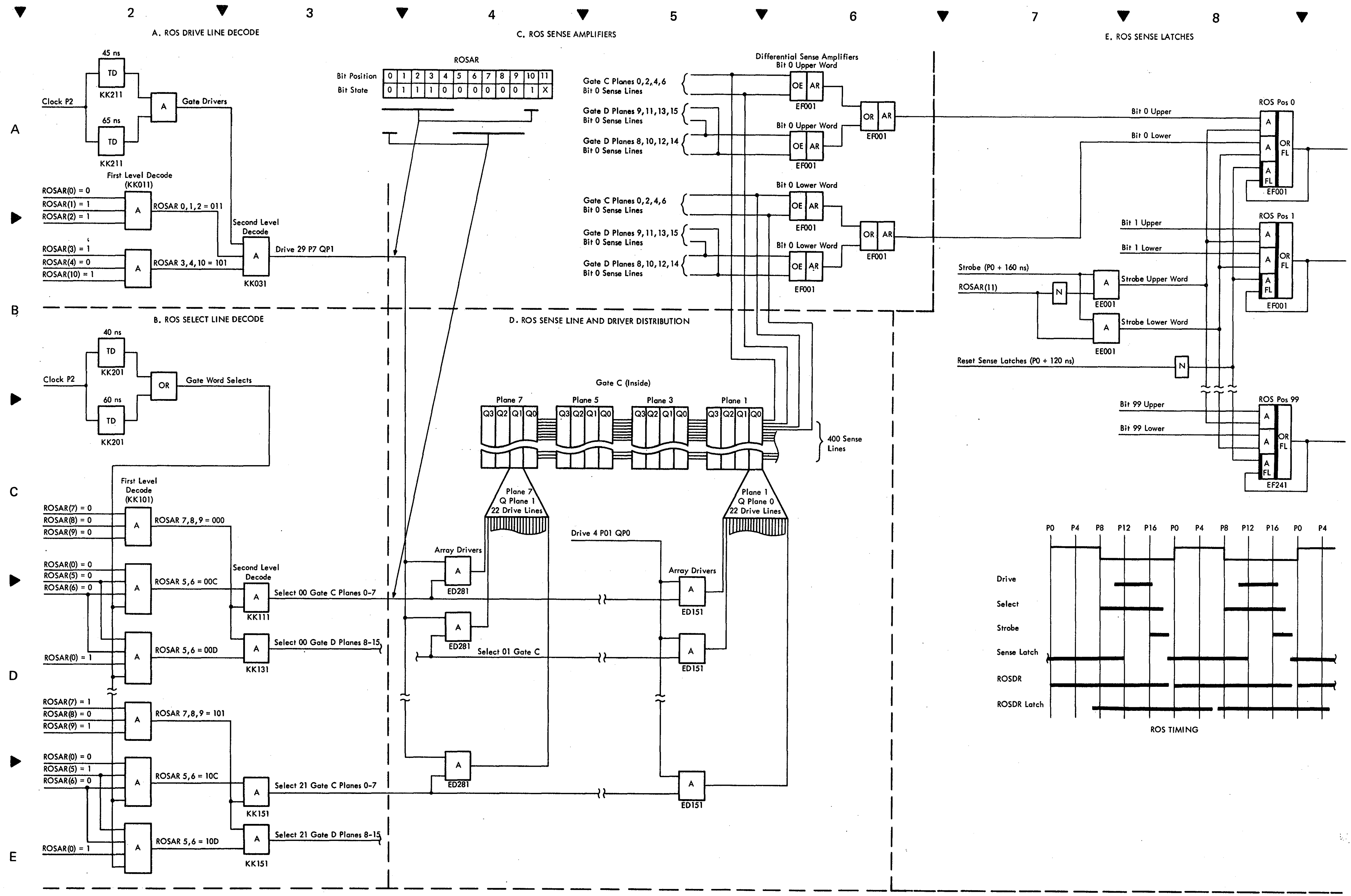


Diagram 4-105. ROS Addressing and Data Flow (Sheet 1 of 2)

2 3 4 5 6 7 8 9

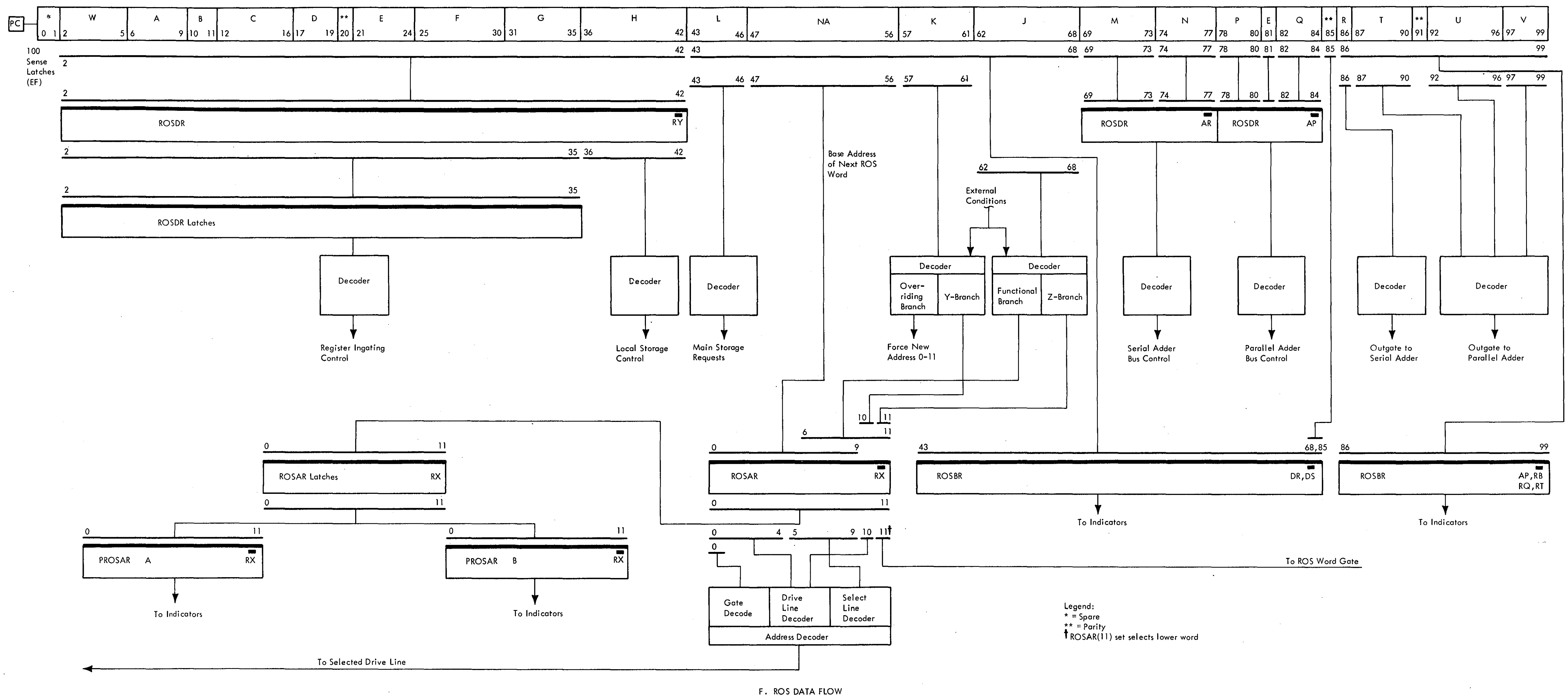
A

B

C

D

E



F. ROS DATA FLOW

Diagram 4-105. ROS Addressing and Data Flow (Sheet 2 of 2)

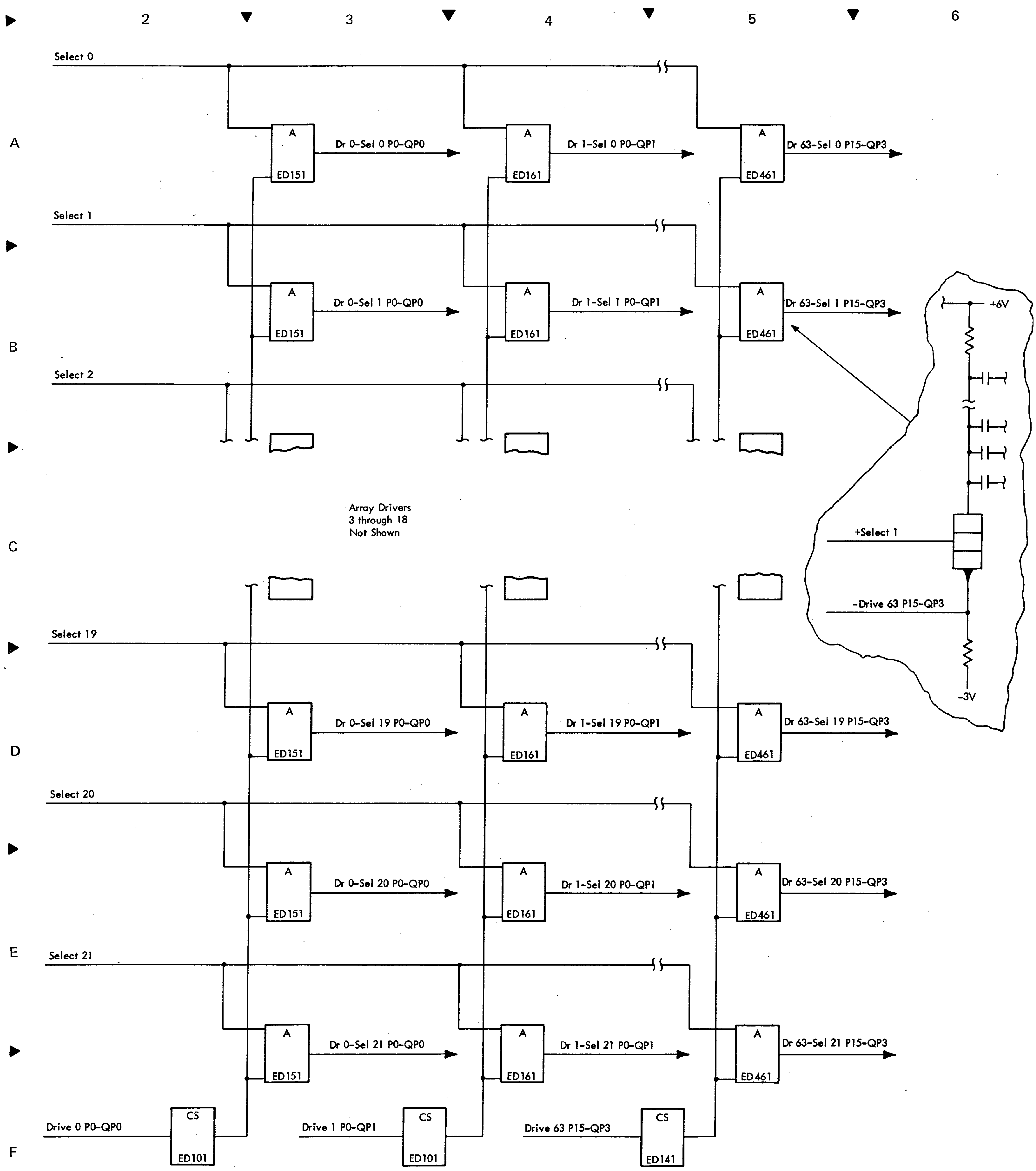
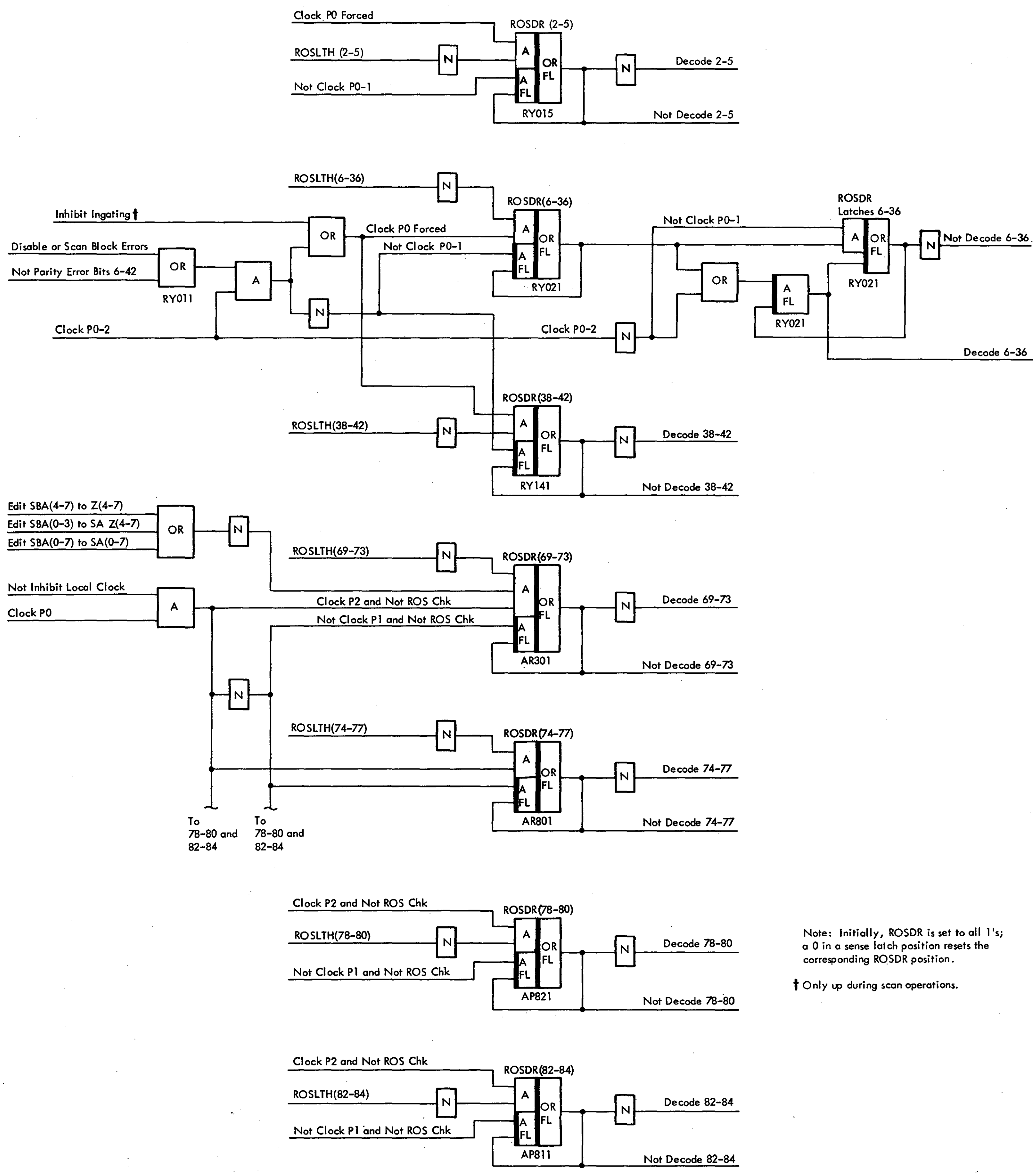


Diagram 4-106. Array Drivers



Note: Initially, ROSDR is set to all 1's; a 0 in a sense latch position resets the corresponding ROSDR position.

† Only up during scan operations.

Diagram 4-107. ROS Data Register

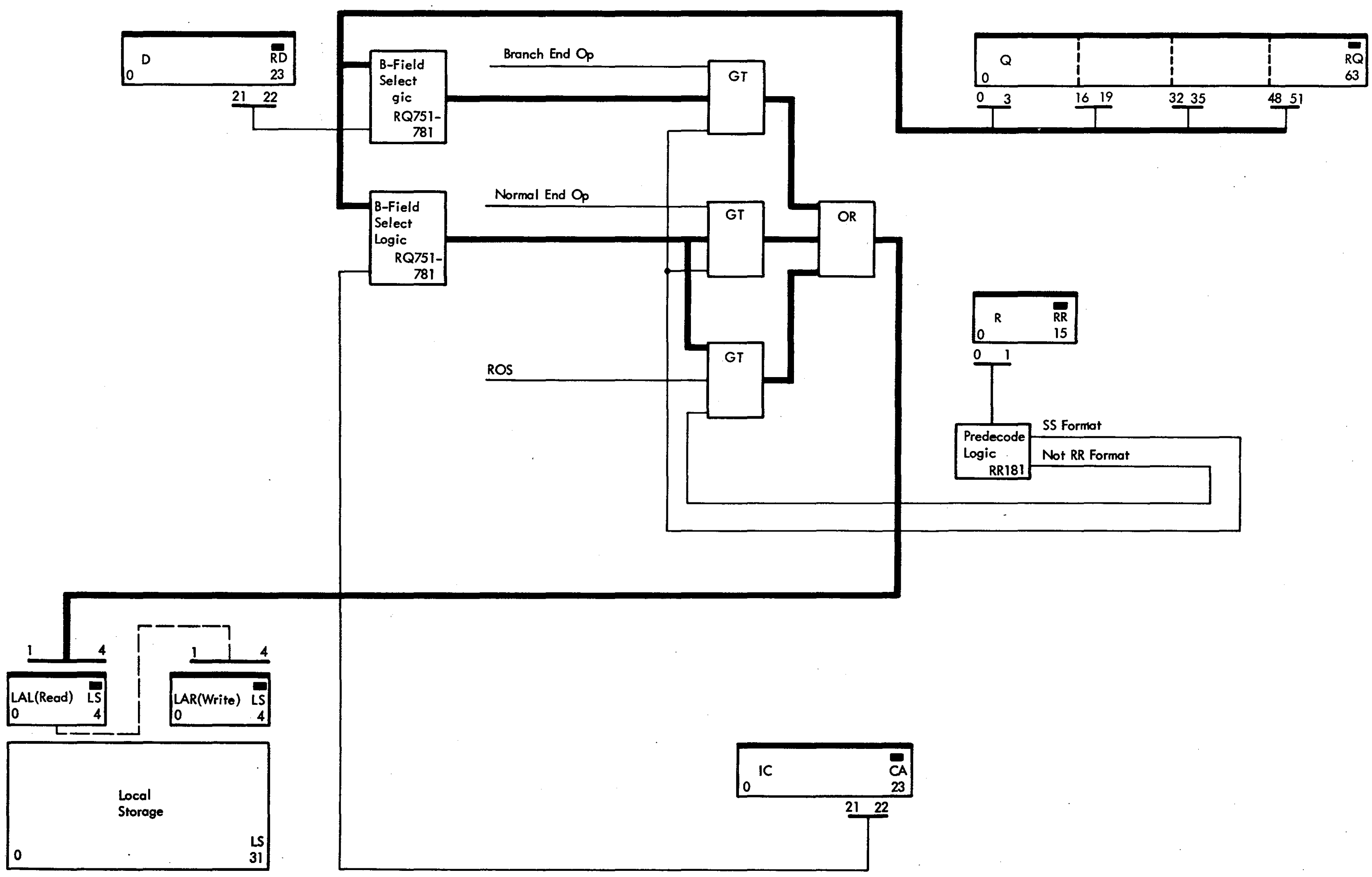


Diagram 4-201. Q-Register B-Field Transfer Controls

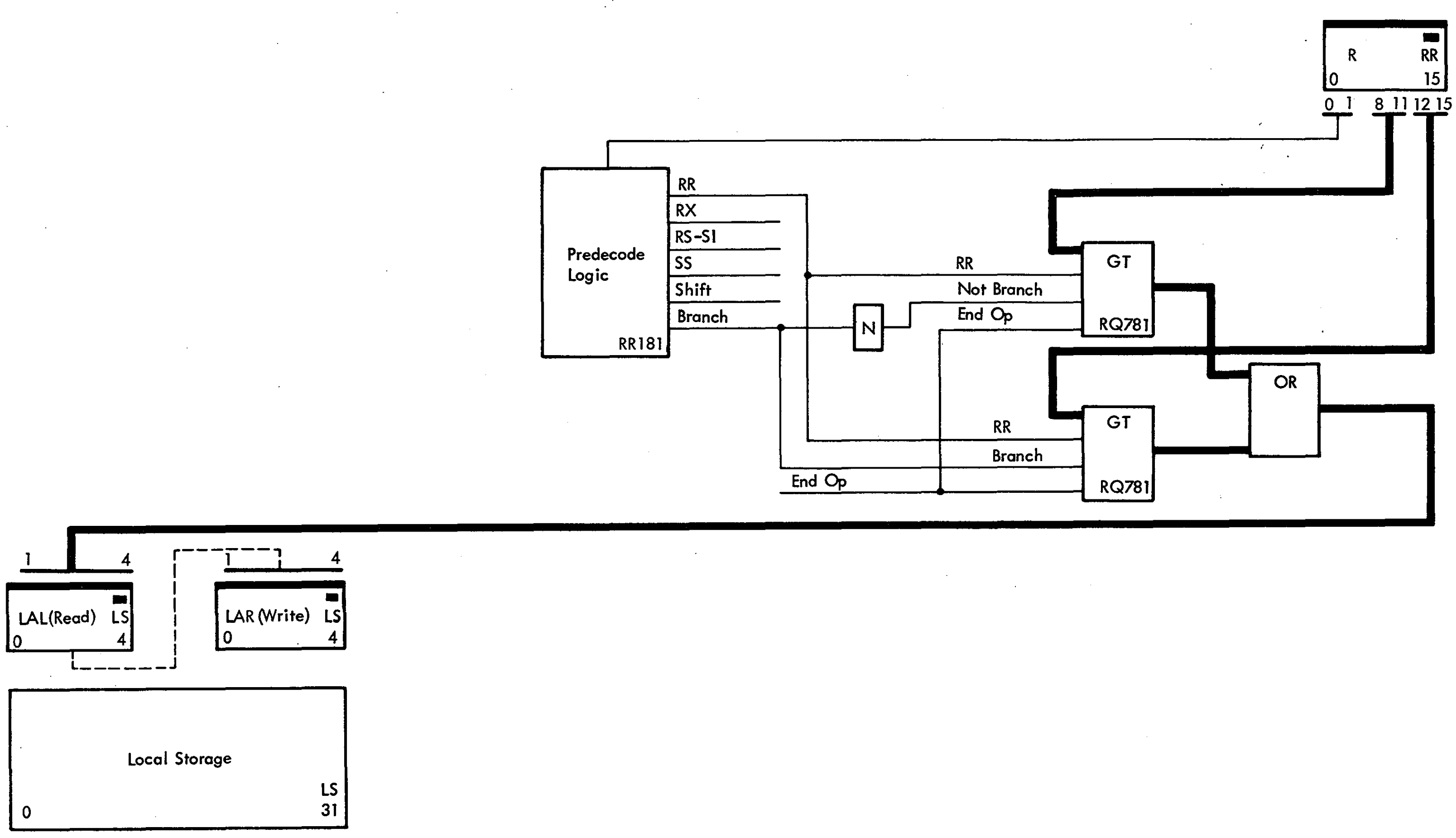


Diagram 4-202. R-Register Transfer to LAL

2 3 4 5 6

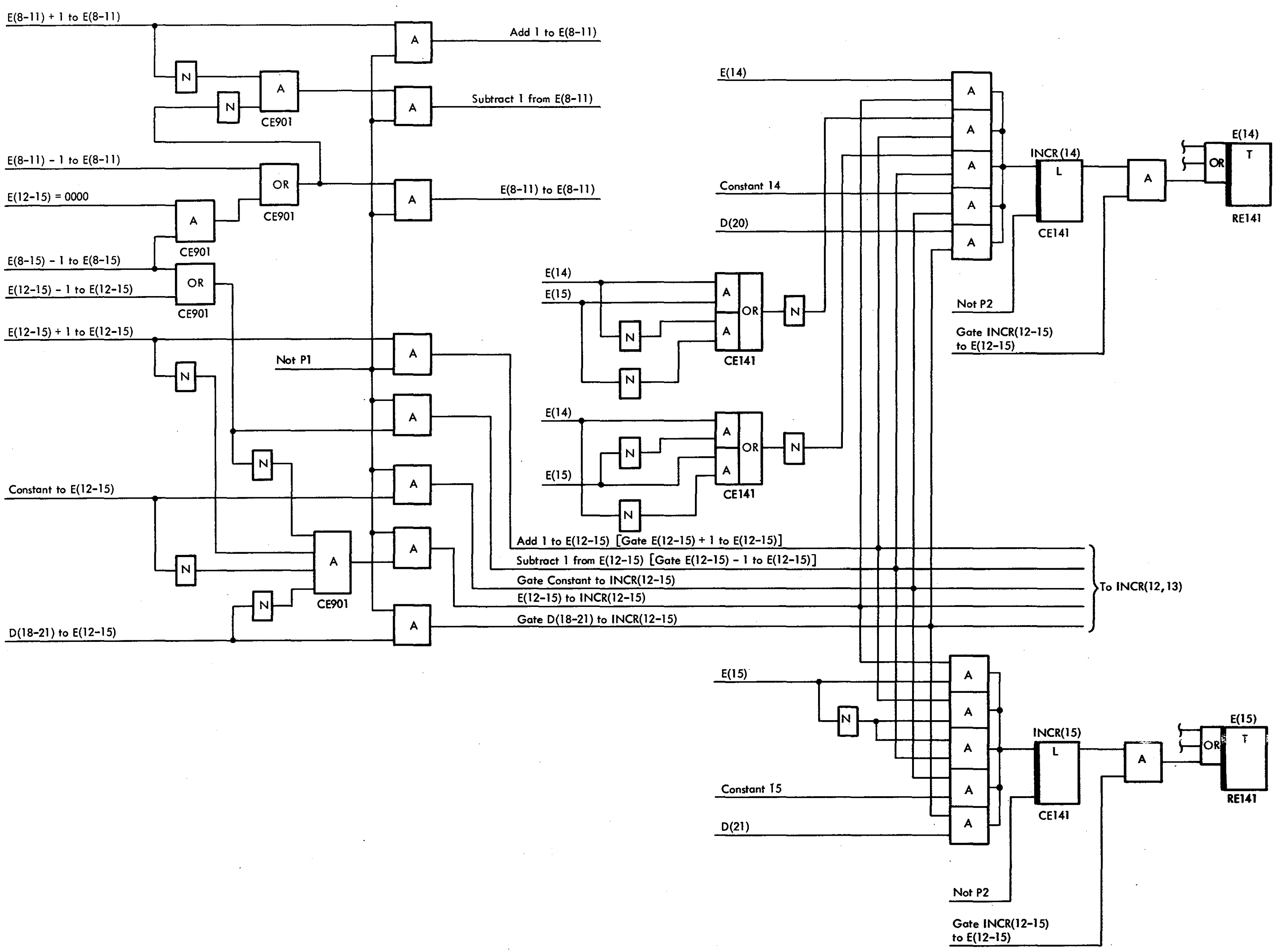


Diagram 4-203. E-Register Incrementer, Bits 14 and 15

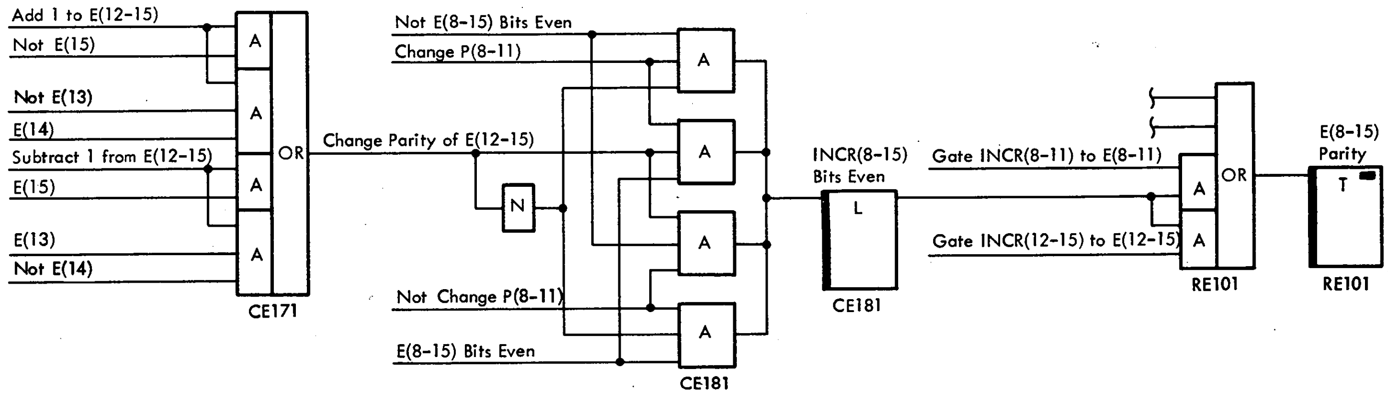


Diagram 4-204. E-Register Parity Prediction after Incrementing

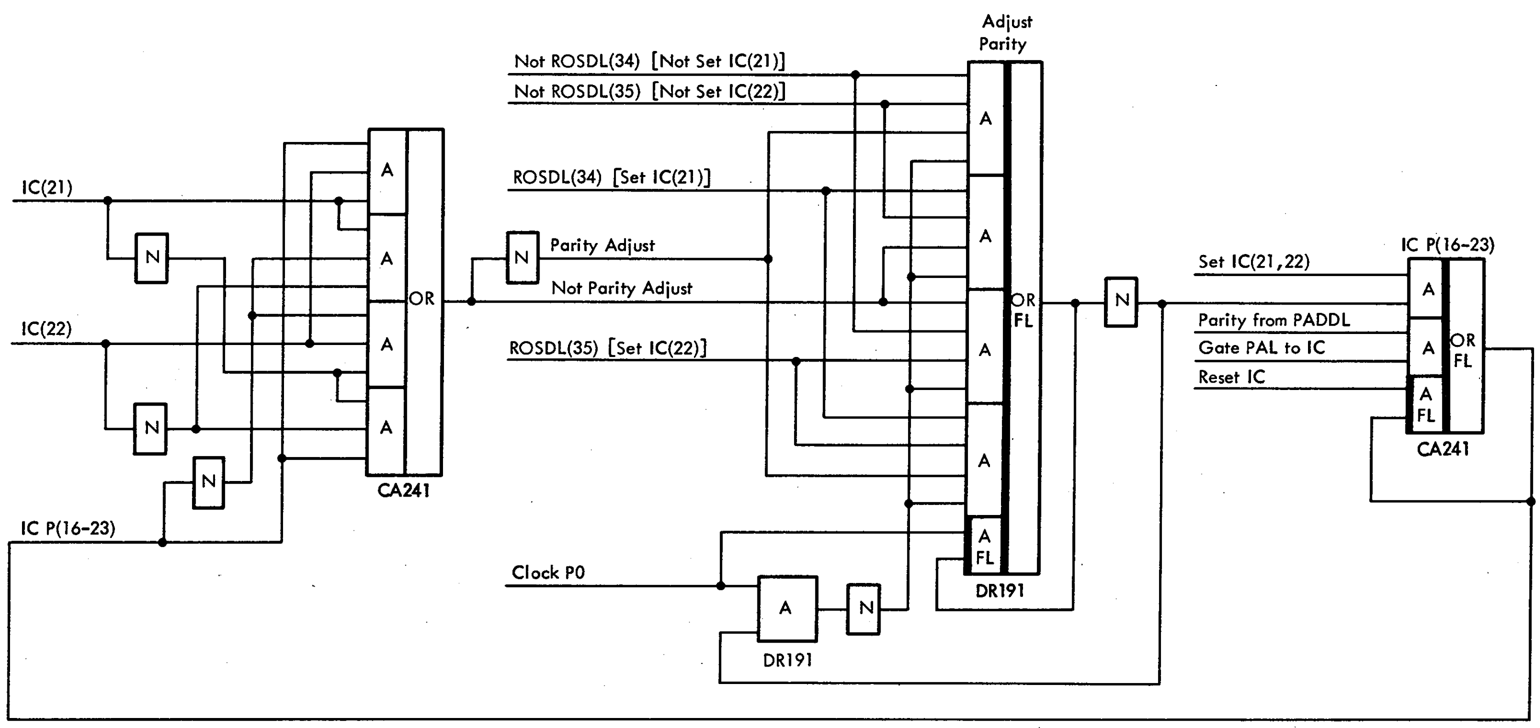


Diagram 4-205. Parity Adjustment for IC (21, 22) Stepping

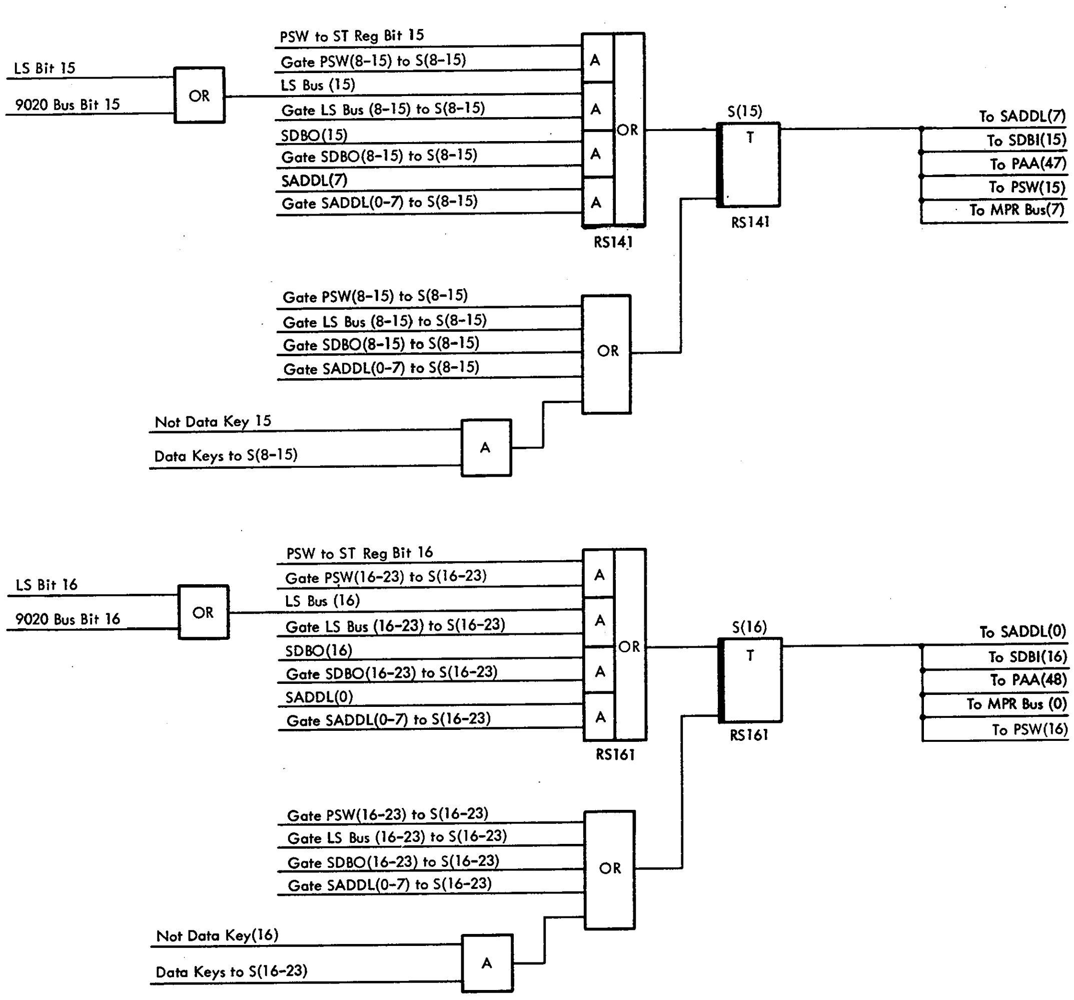


Diagram 4-206. S-Register, Bits 15 and 16

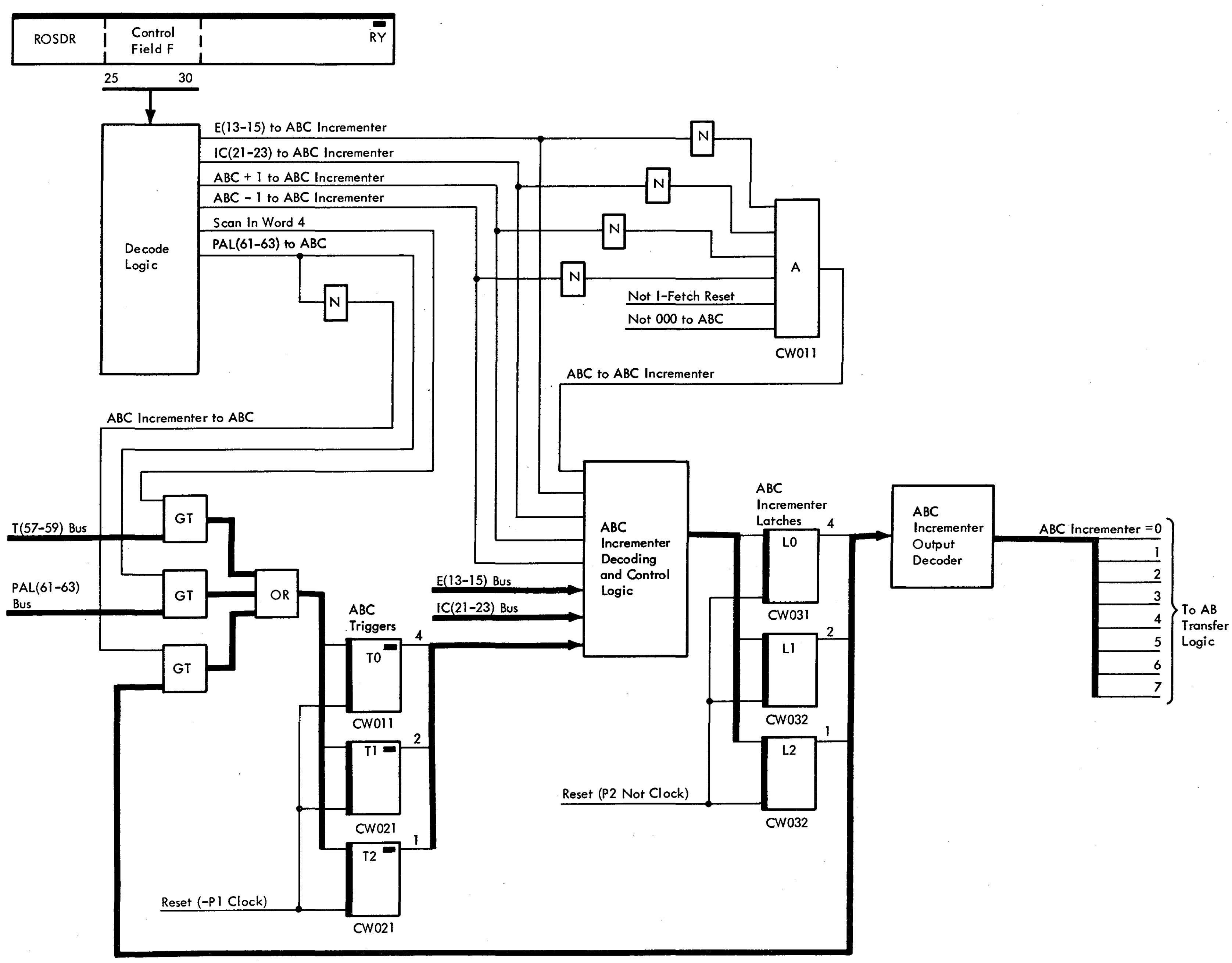


Diagram 4-207. AB Byte Counter

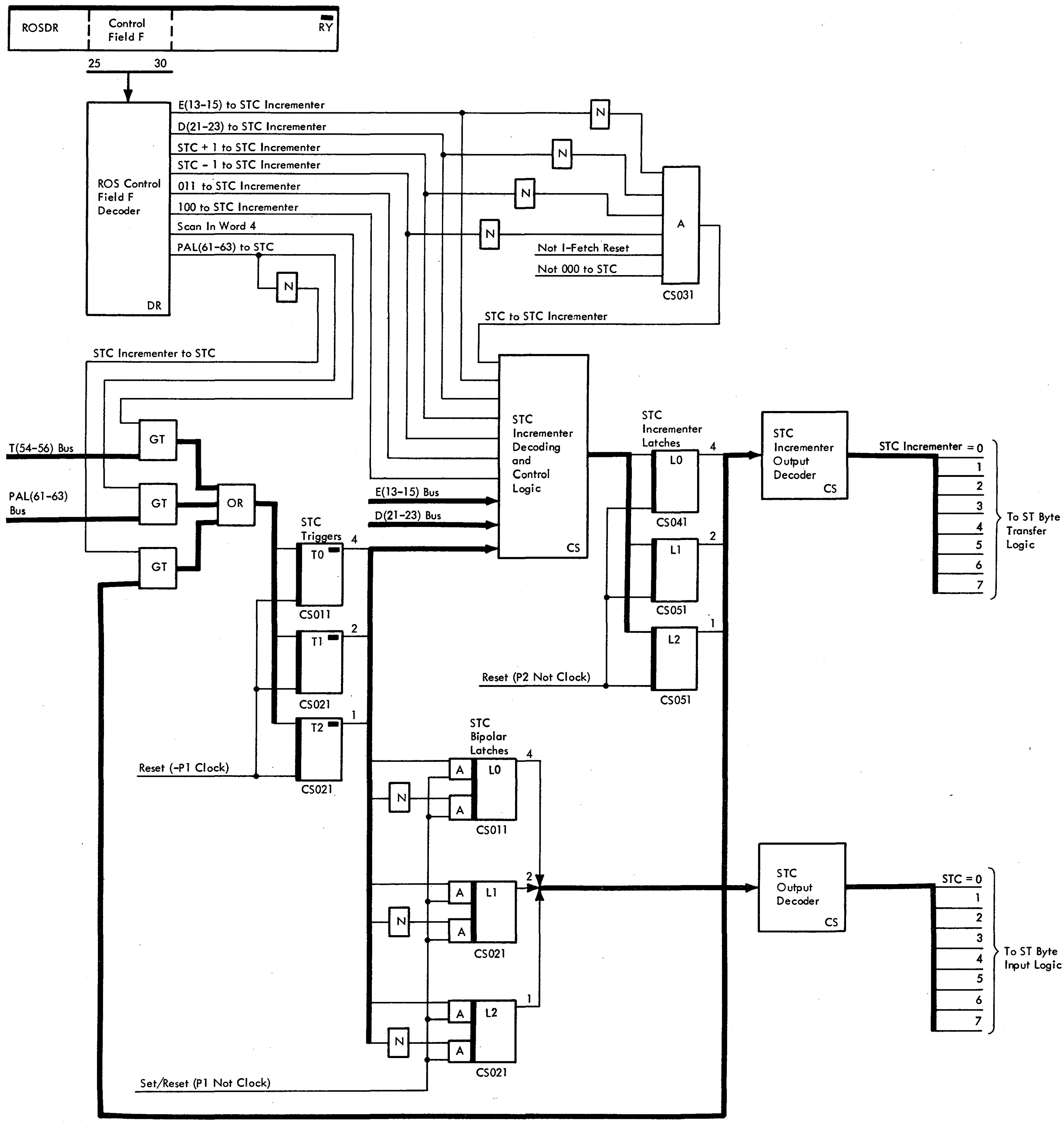


Diagram 4-208. ST Byte Counter

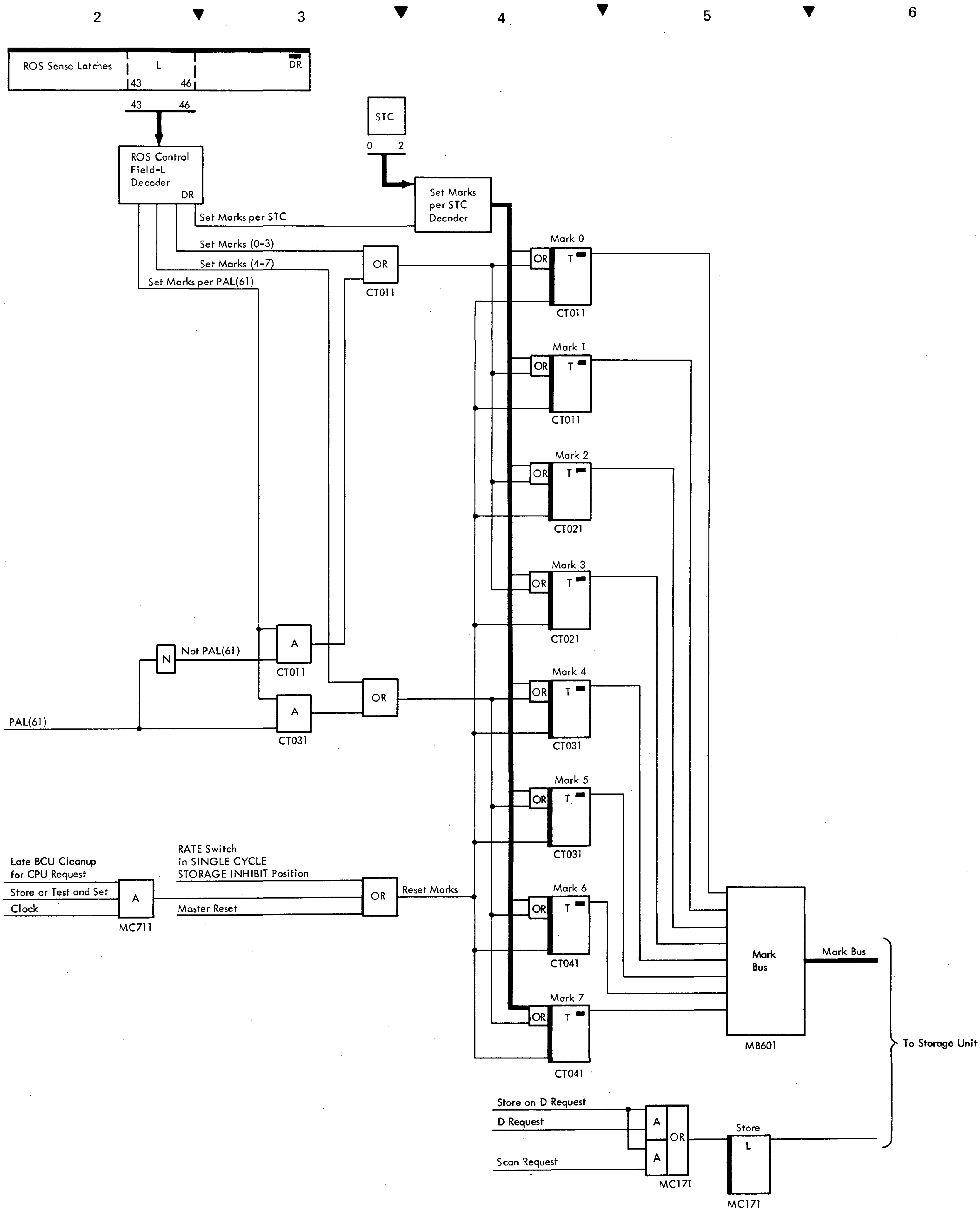


Diagram 4-209. Mark Trigger Logic

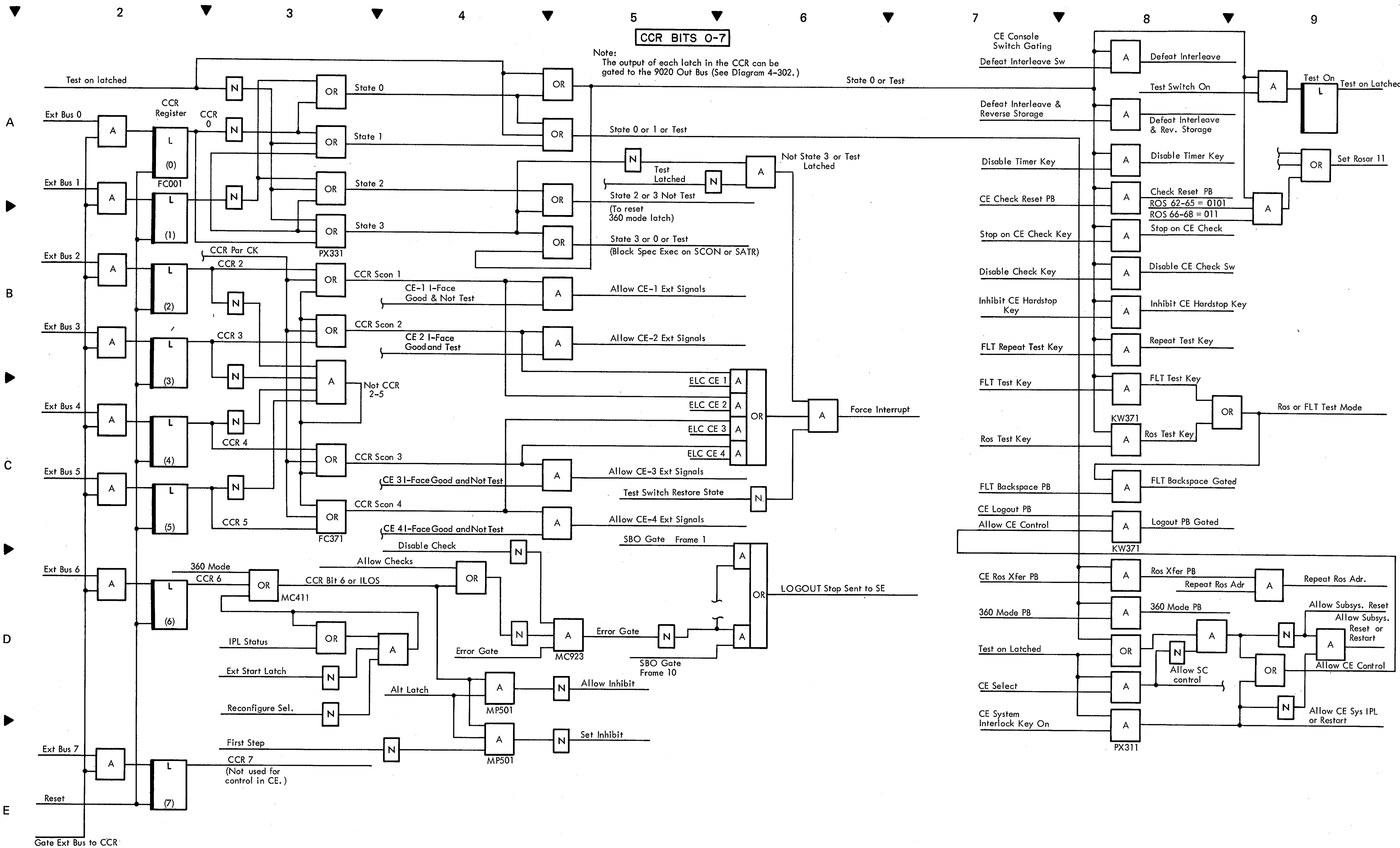


Diagram 4-210. CCR Output Logic and Control Paths (Sheet 1 of 3)

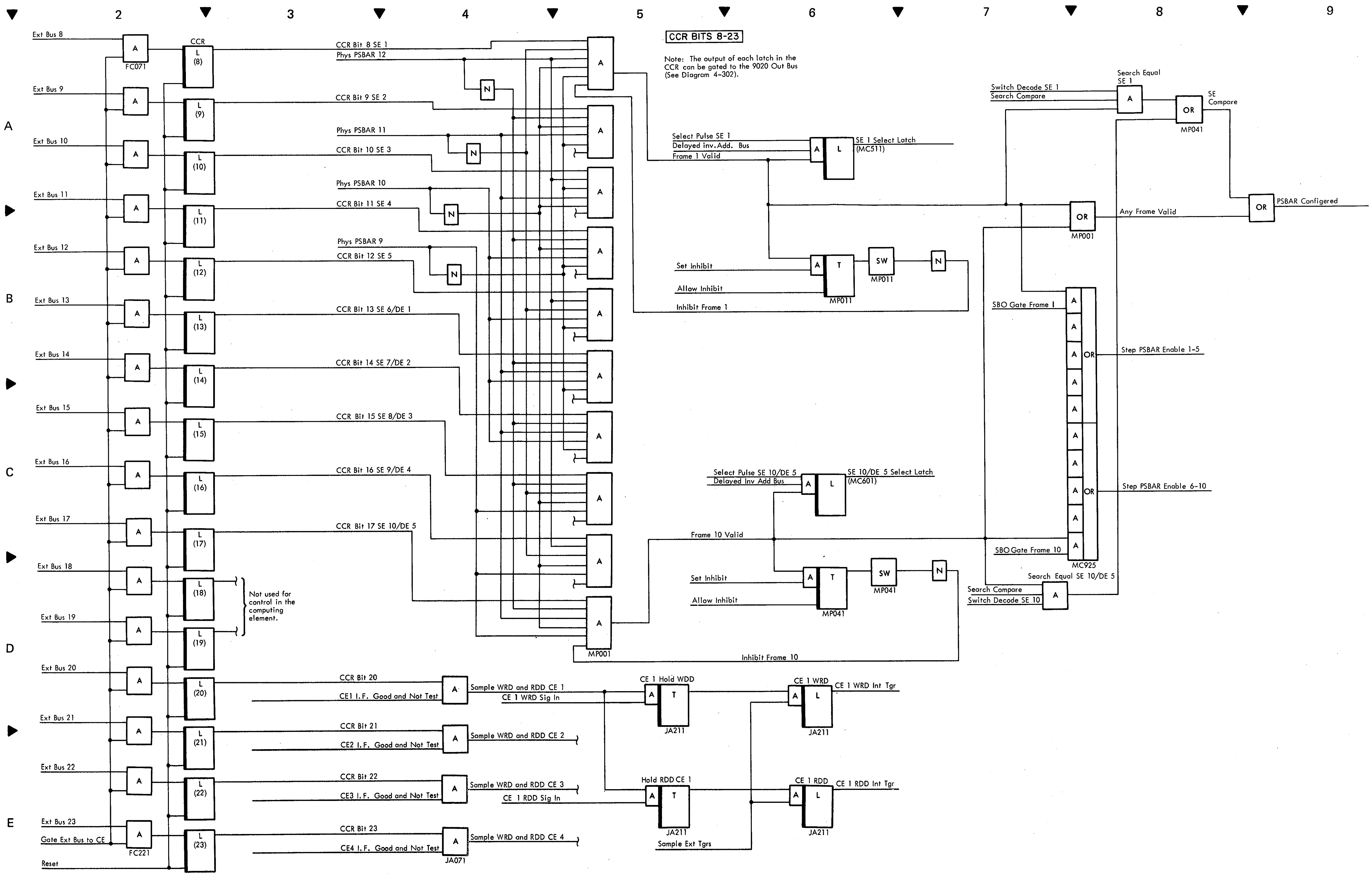


Diagram 4-210. CCR Output Logic and Control Paths (Sheet 2 of 3)

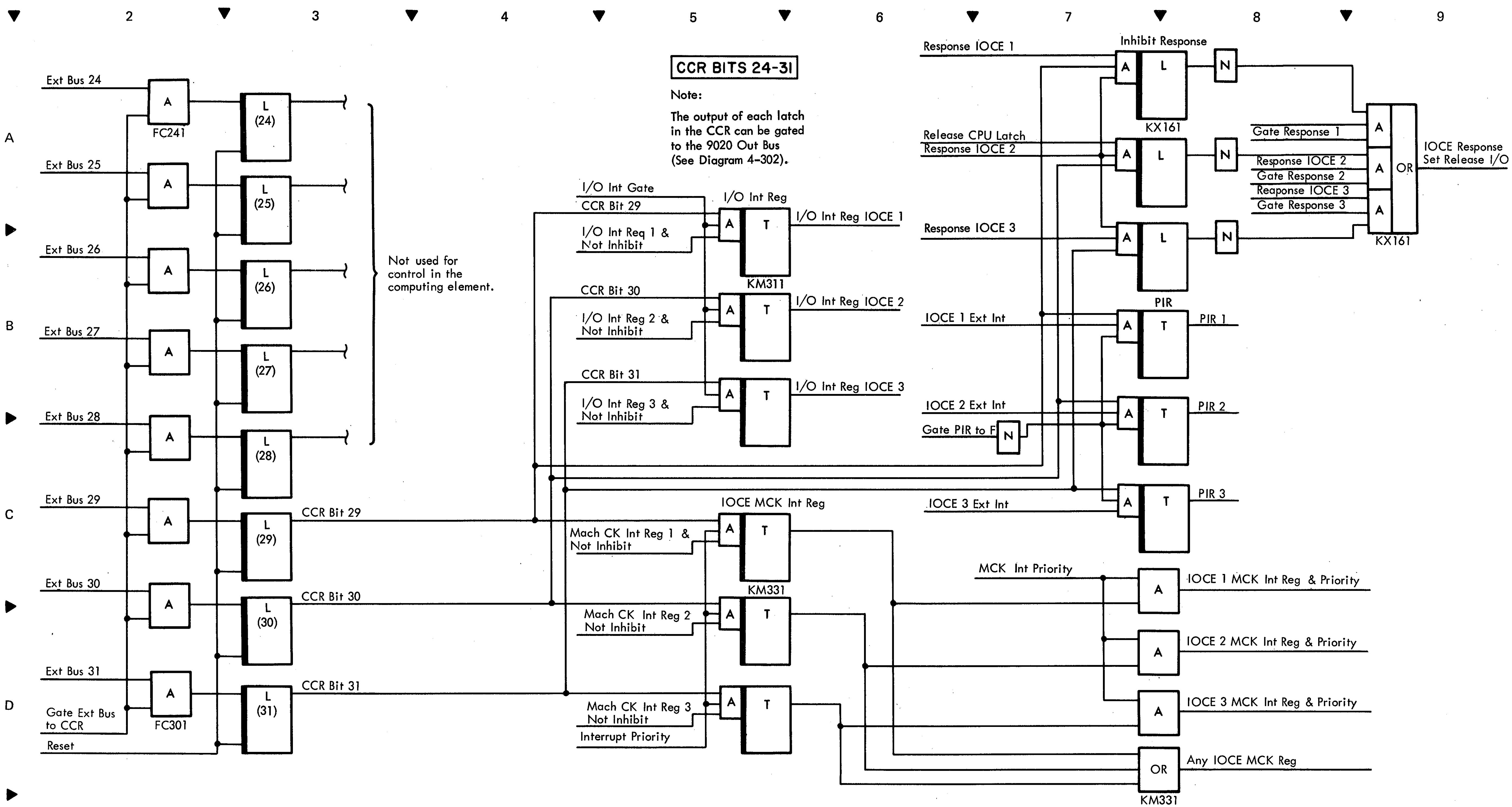
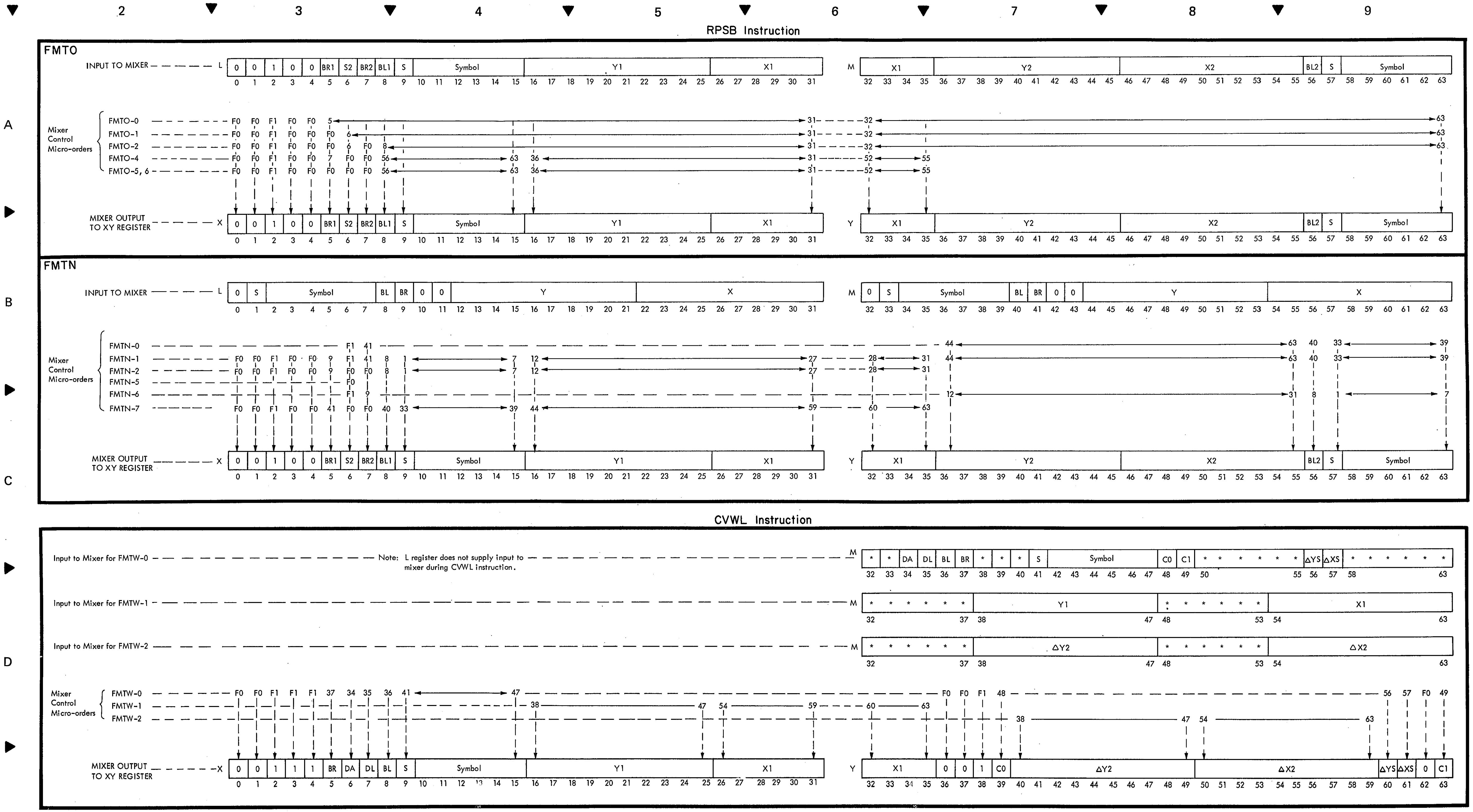


Diagram 4-210. CCR Output Logic and Control Paths (Sheet 3 of 3)



Notes:

1. To trace the origin of a mixer output bit: (1) select chart for specific instruction, RPSB or CVWL; (2) if RPSB is selected, refer to FMTO or FMTN portion of chart; (3) find bit to be traced in XY register and; (4) follow the line up to the micro-order being issued. For example, assume an RPSB instruction and an FMTN micro-order and assume bit 8 is to be traced. Refer to the RPSB instruction chart, FMTN portion, XY register. Find bit 8 in the XY register and follow the line up to the specific micro-order. If the specific micro-order is an FMTN-7, bit 8 originates from bit 40 of the LM-register. Note also that FMTN-0, 5, and 6 do not set bit 8 and that FMTN-1 and 2 set bit 8 from the input bit 8 in the LM register.
2. Parity for all XY bytes is predicted from LM data parity and LM parity bits.
3. XY register is reset on receipt of either FMTO; FMTN-1, -2, -7; or FMTW-0.

Legend:

1. F0 = Bit forced to 0
2. F1 = Bit forced to 1
3. BR = Brightness bit
   BL = Blink bit
   S = Symbol size
   DA = Dash
   DL = Dash length
   ΔYS = Sign of ΔY2
   ΔXS = Sign of ΔX2
   C0 = Character at major position
   C1 = Character at secondary position
   * = Not used

Diagram 4-211. LM to XY Reformatting via Mixer (Sheet 1 of 2)

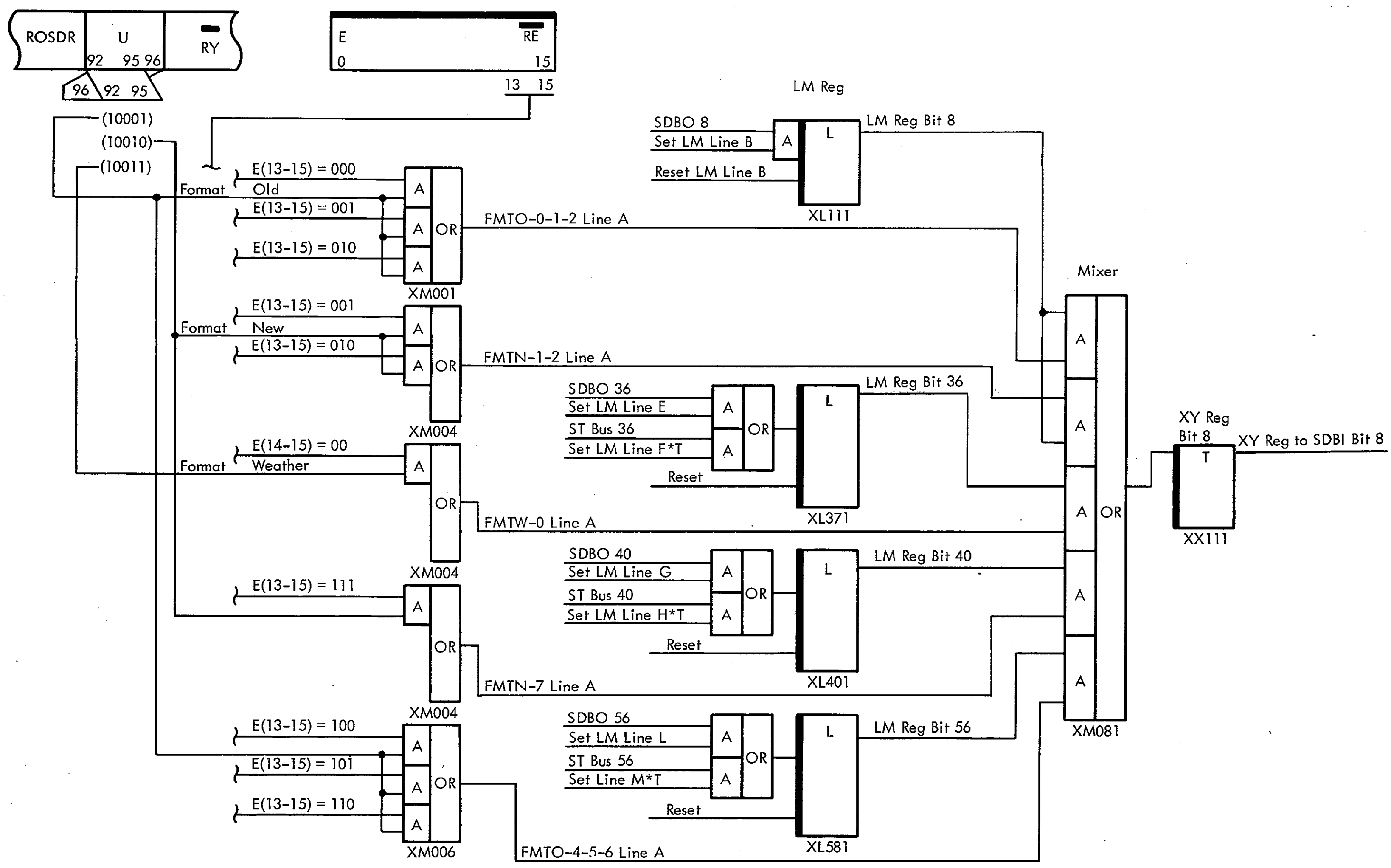


Diagram 4-211. LM to XY Reformatting via Mixer (Sheet 2 of 2)

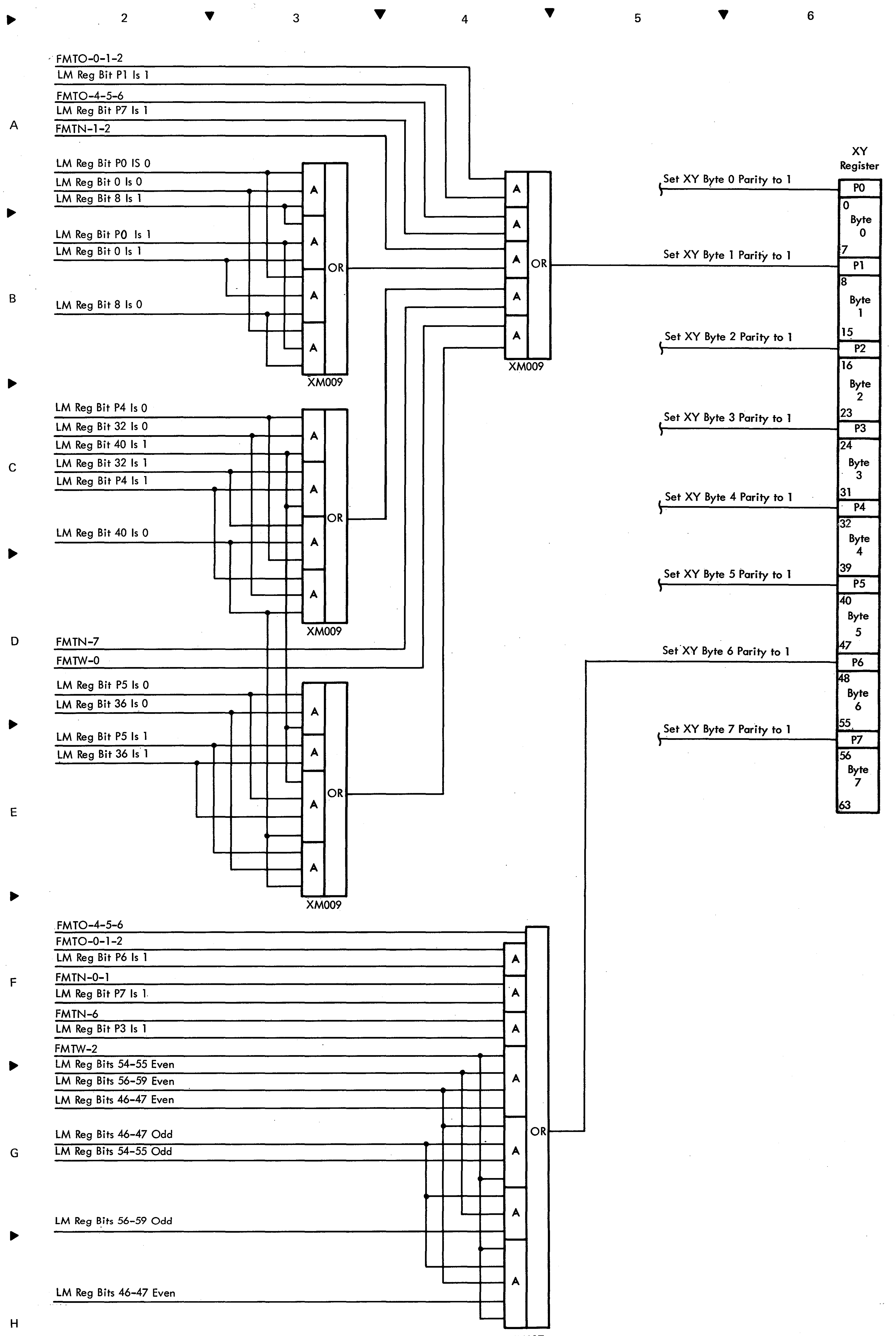


Diagram 4-212. XY Register Parity Prediction Logic

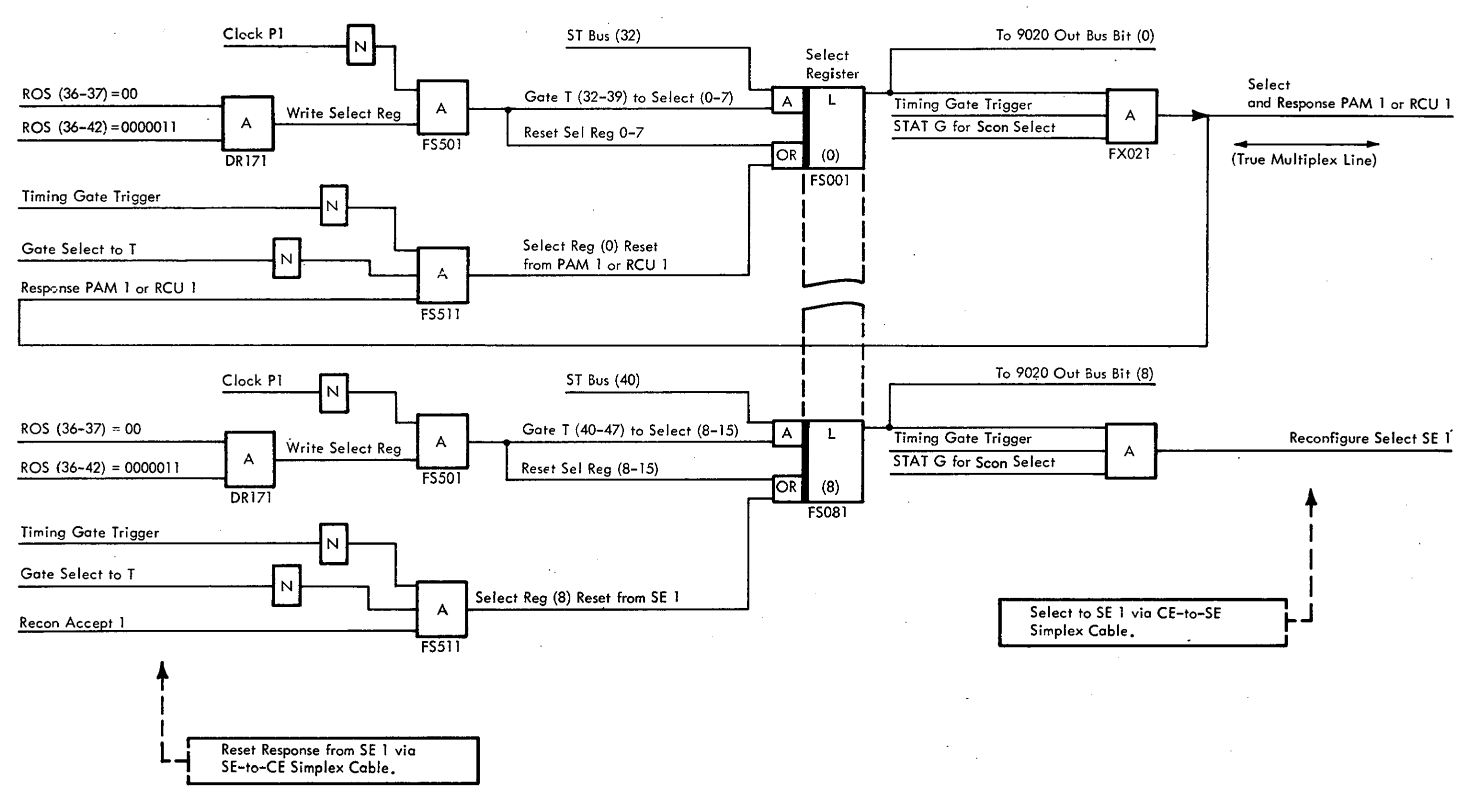


Diagram 4-213. Select Register - Select Signal Generation and Response Reset

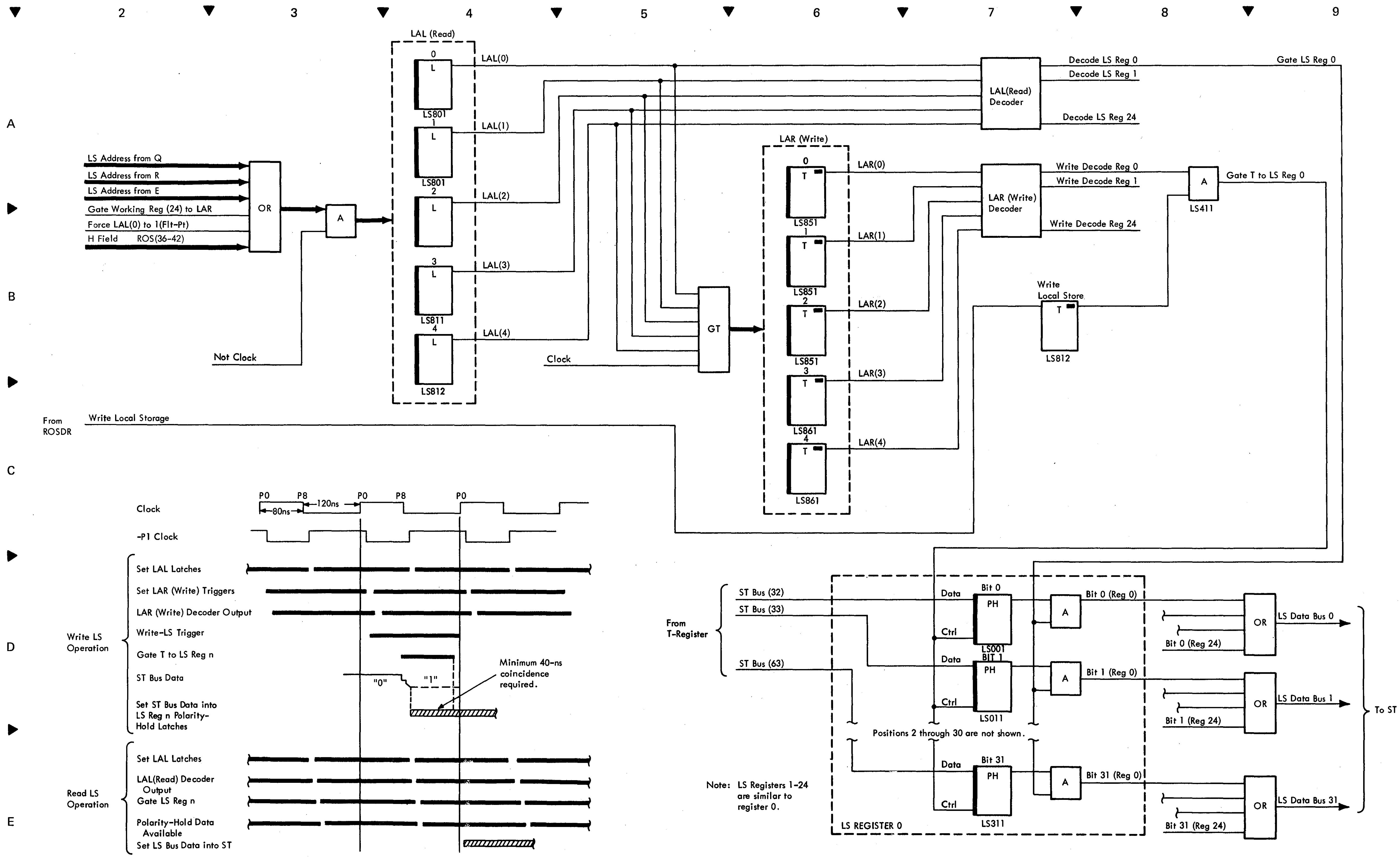


Diagram 4-301. Local Storage Read/Write Controls

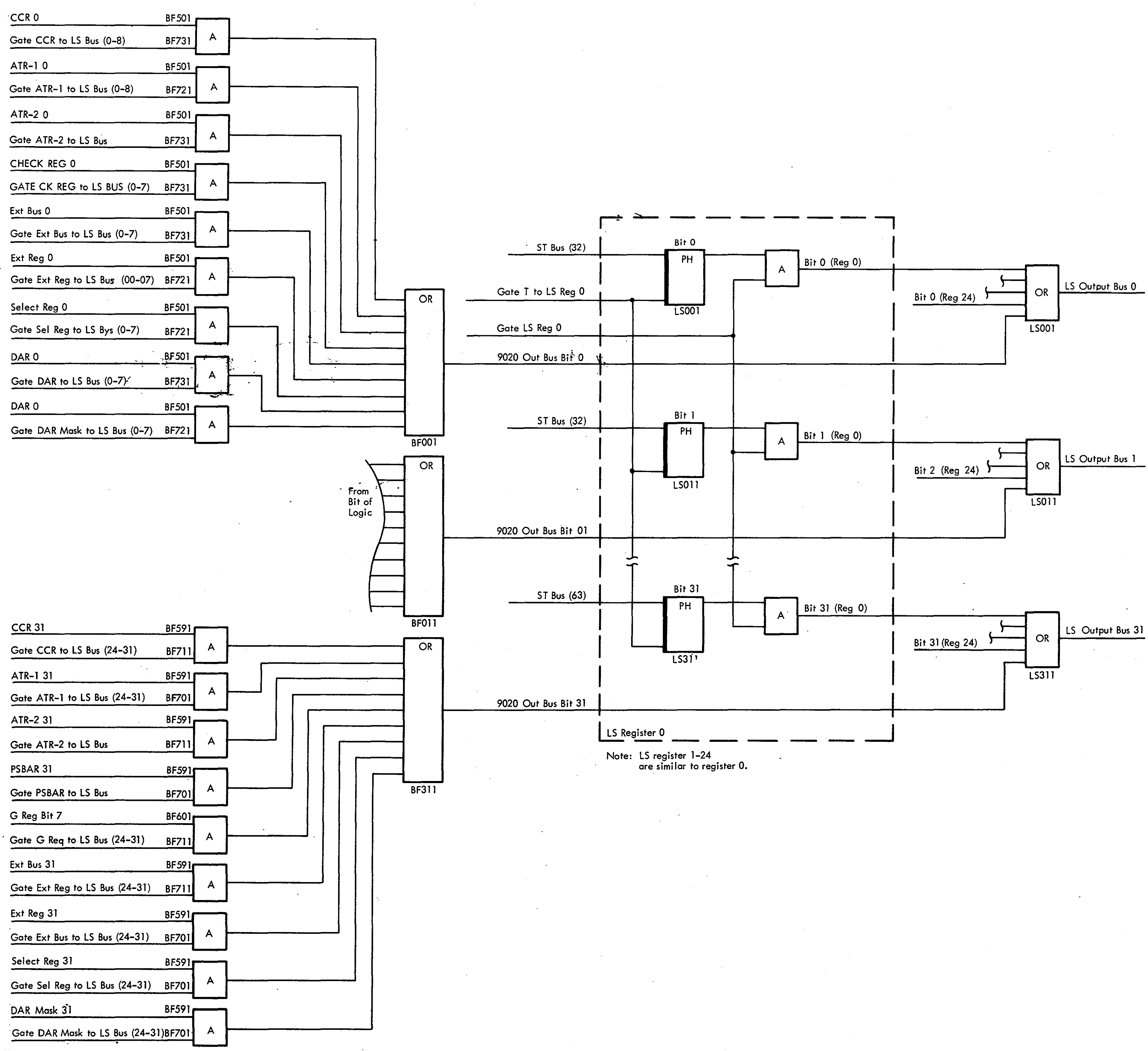


Diagram 4-302. 9020 Out Bus to LS Data Bus Gating Logic

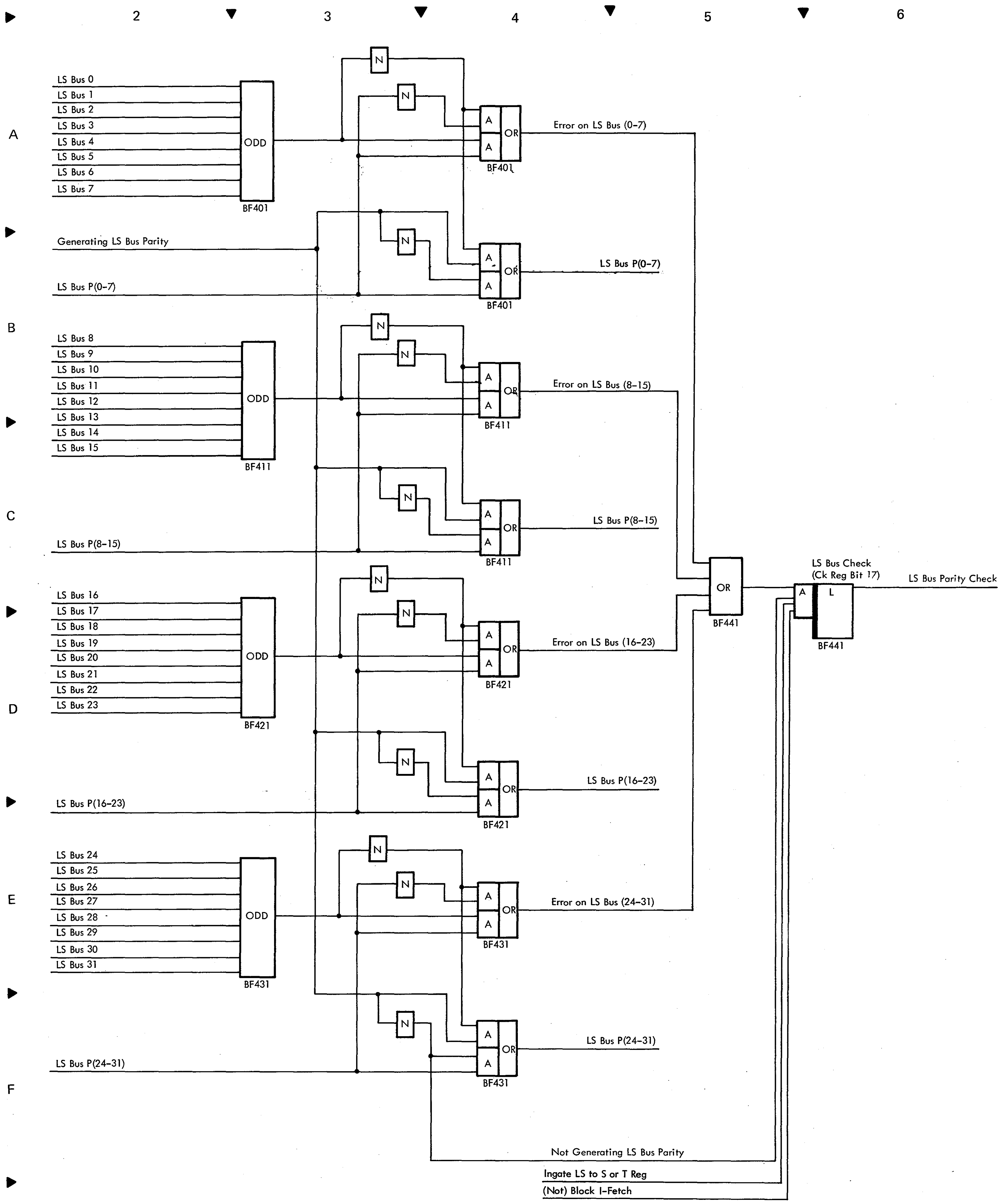


Diagram 4-303. LS Bus Parity Generation or Check

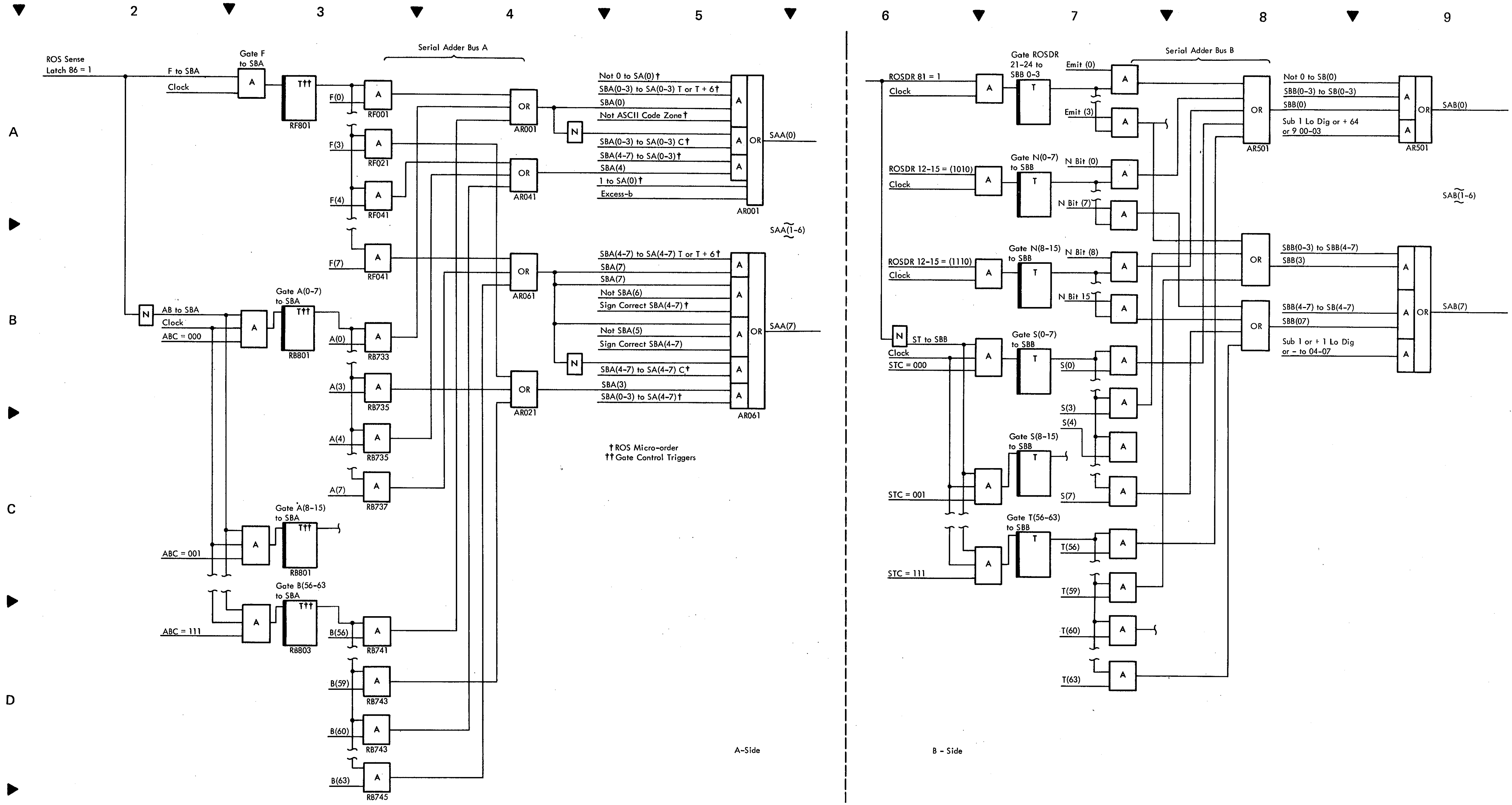


Diagram 4-401. Serial Adder Input Bus Logic

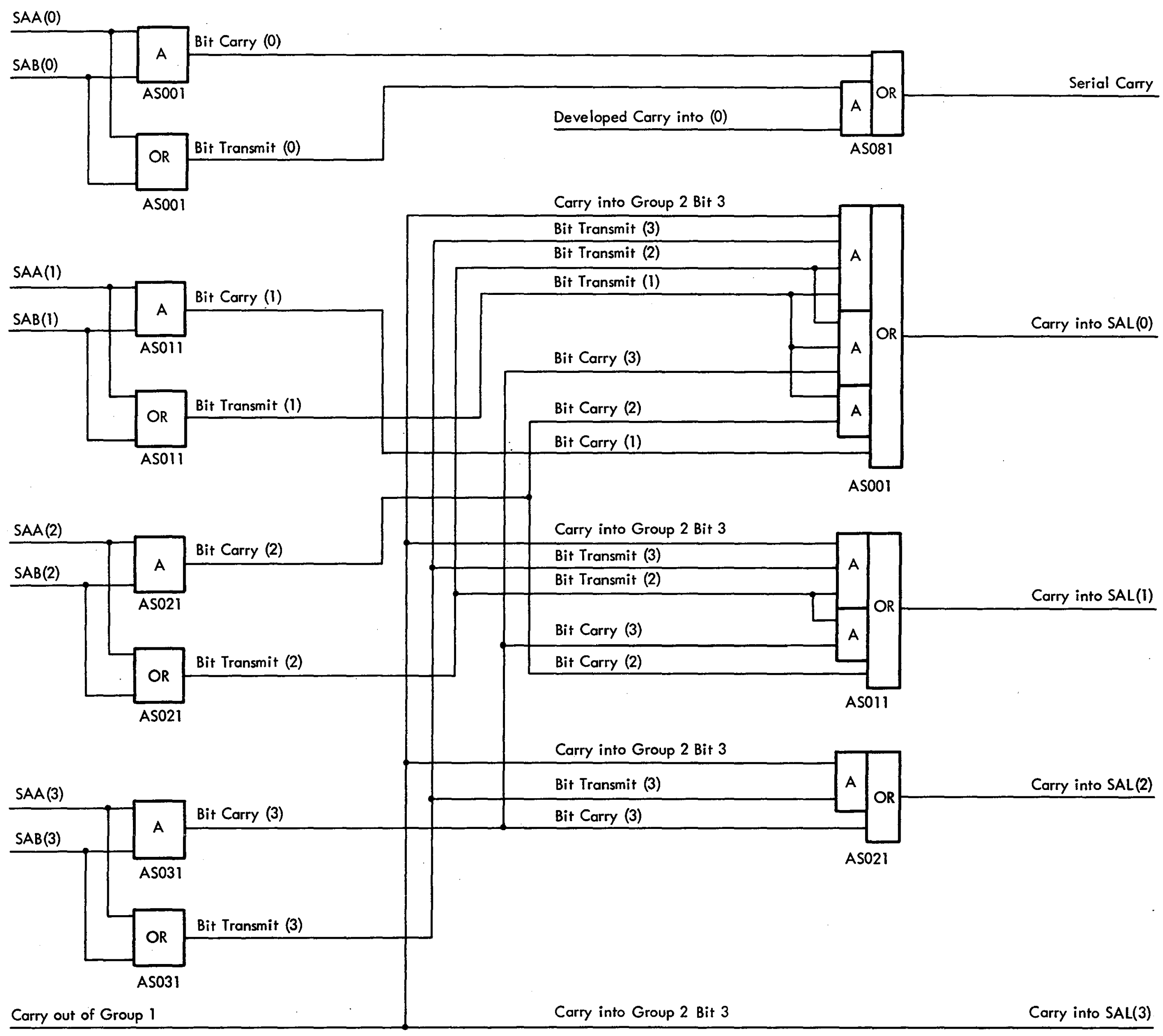
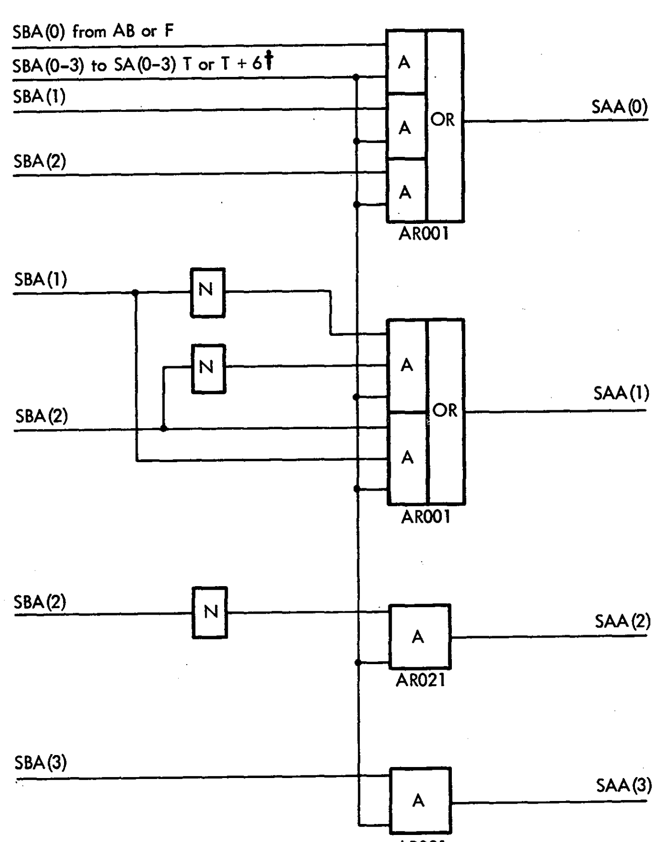


Diagram 4-402. Carry Lookahead Logic, SAL(0-3)



† ROS Micro-order

Diagram 4-403. Decimal Add 6 Logic

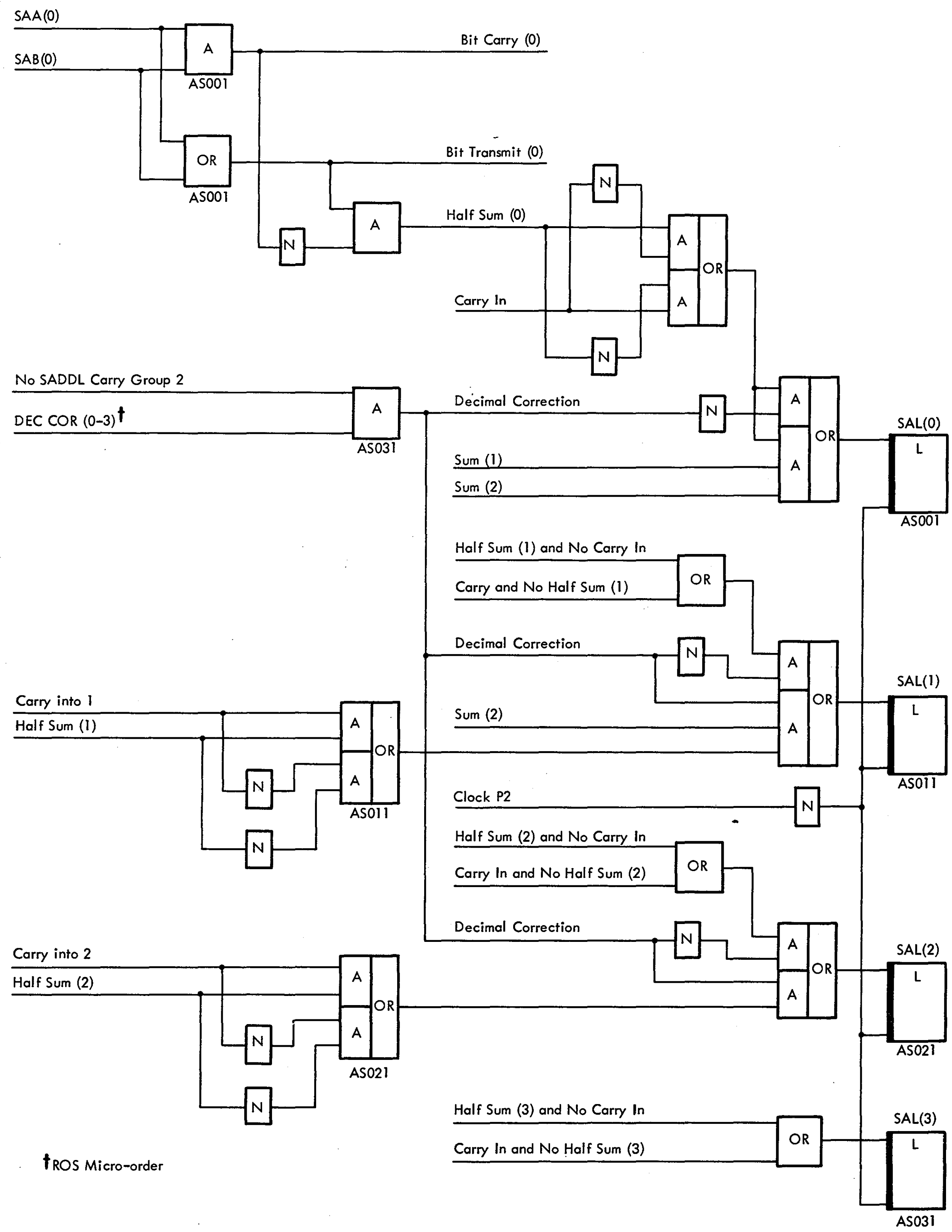


Diagram 4-404. Decimal Correction Logic For SAL (0-3)

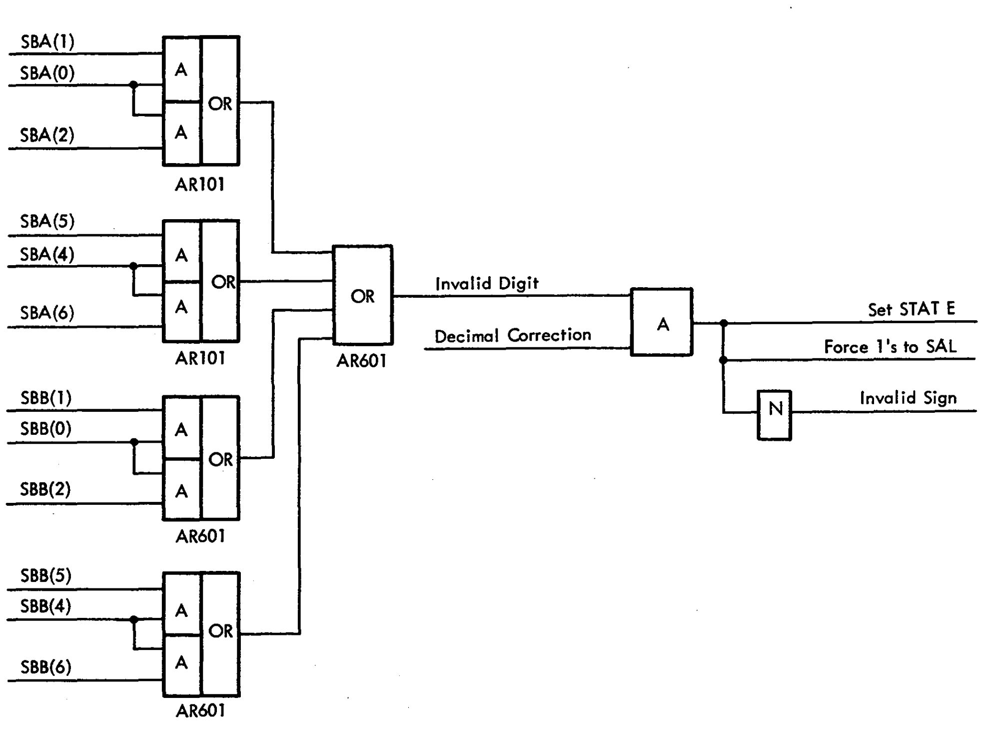


Diagram 4-405. Invalid Digit Logic

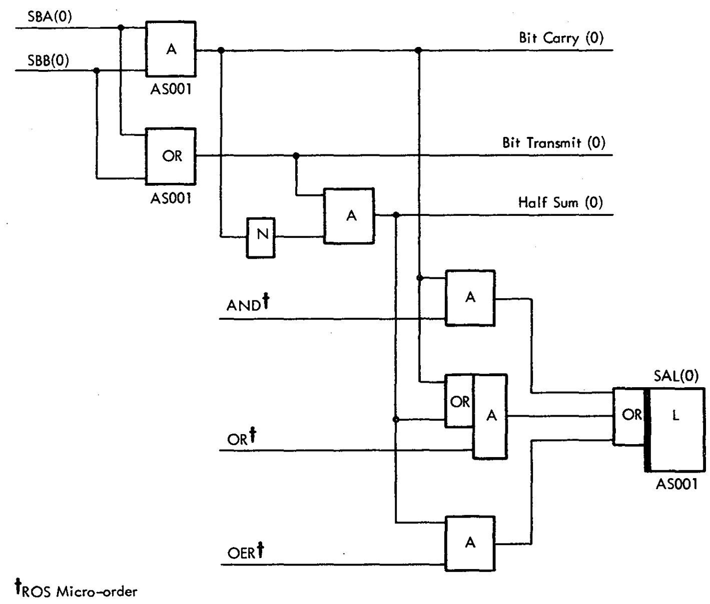


Diagram 4-406. Logical Functions, SAL (0)

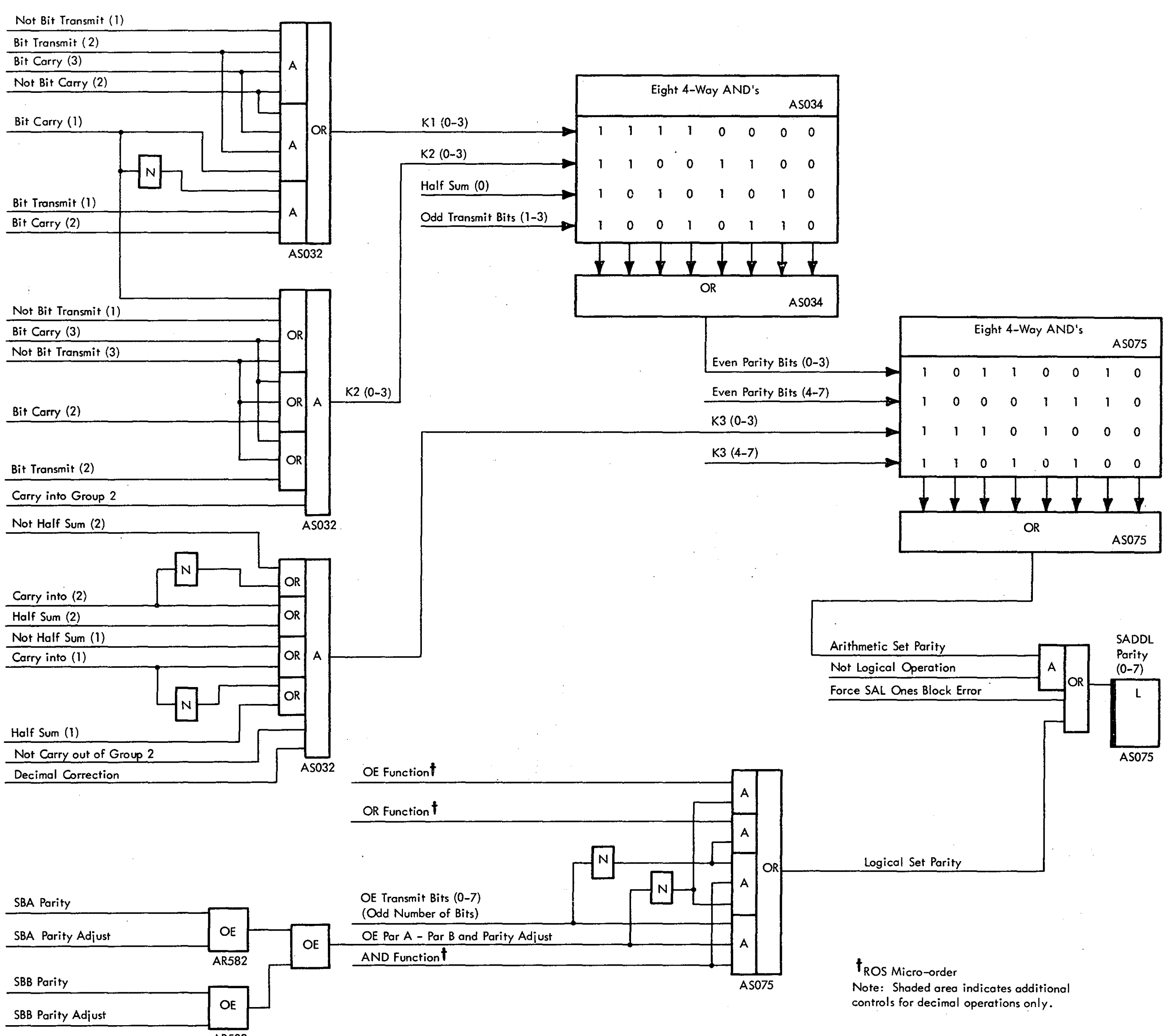


Diagram 4-407. Serial Adder Parity Predict Logic

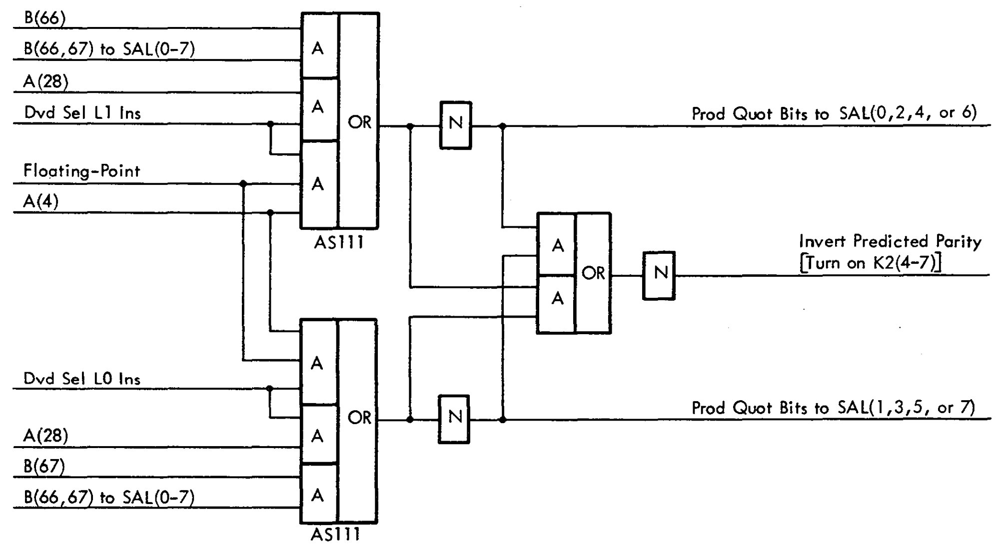


Diagram 4-408. Serial Adder Product-Quotient Bit Logic

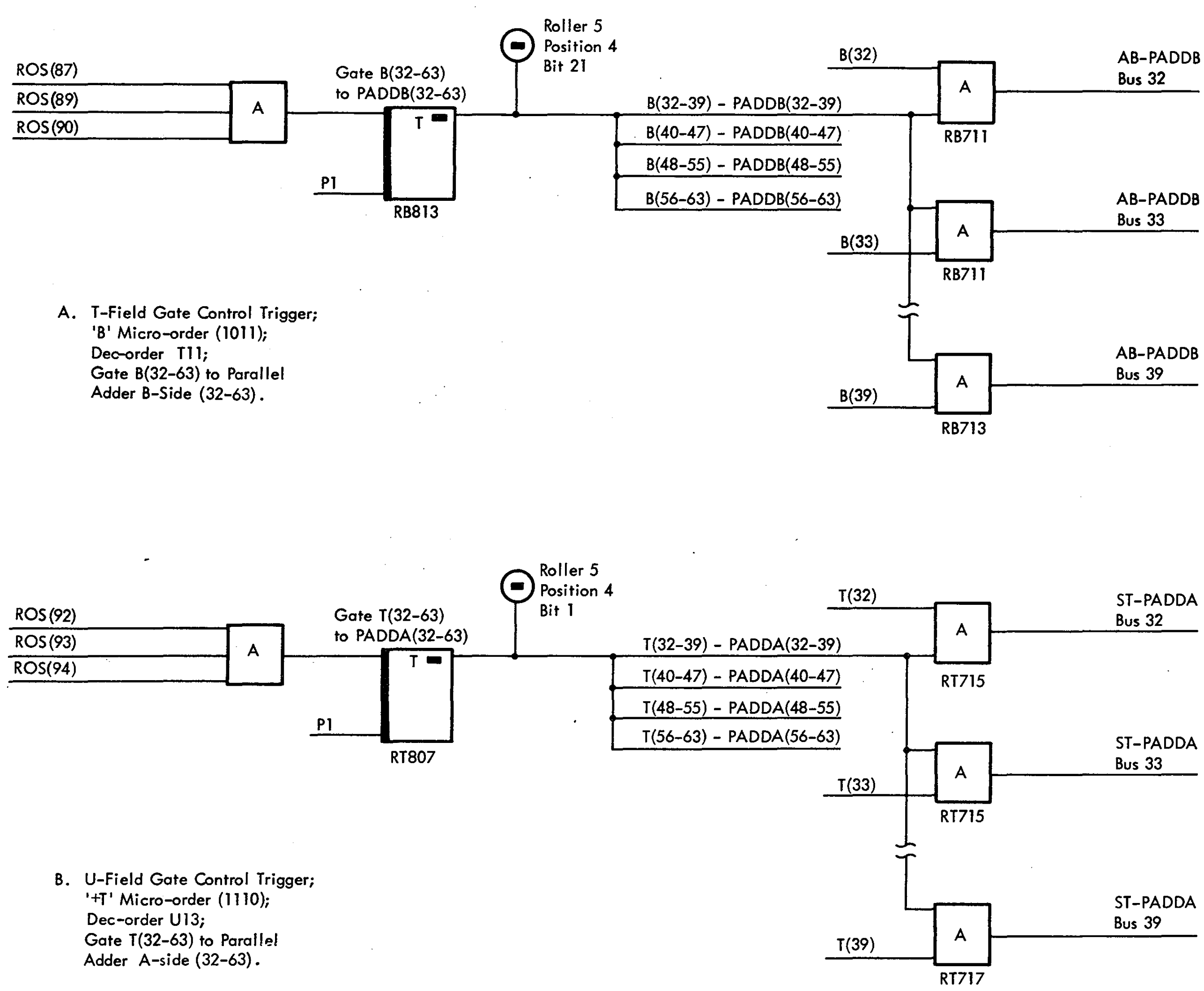
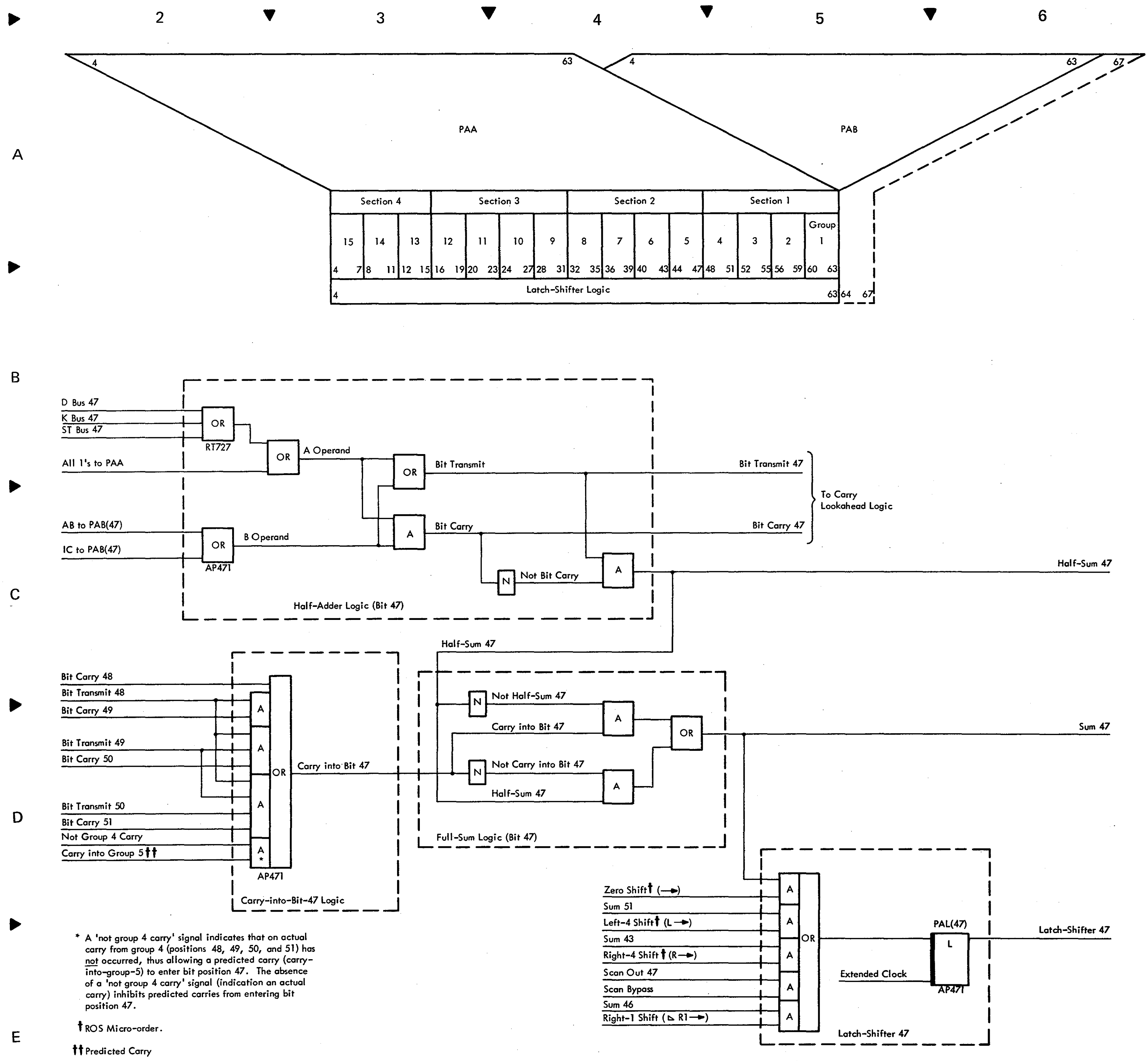


Diagram 4-409. Gate Control Triggers for 'B + T' Micro-order



* A 'not group 4 carry' signal indicates that an actual carry from group 4 (positions 48, 49, 50, and 51) has not occurred, thus allowing a predicted carry (carry-into-group-5) to enter bit position 47. The absence of a 'not group 4 carry' signal (indication an actual carry) inhibits predicted carries from entering bit position 47.

† ROS Micro-order.

†† Predicted Carry

Diagram 4-410. Parallel Adder Bit-Position Logic (Bit 47)

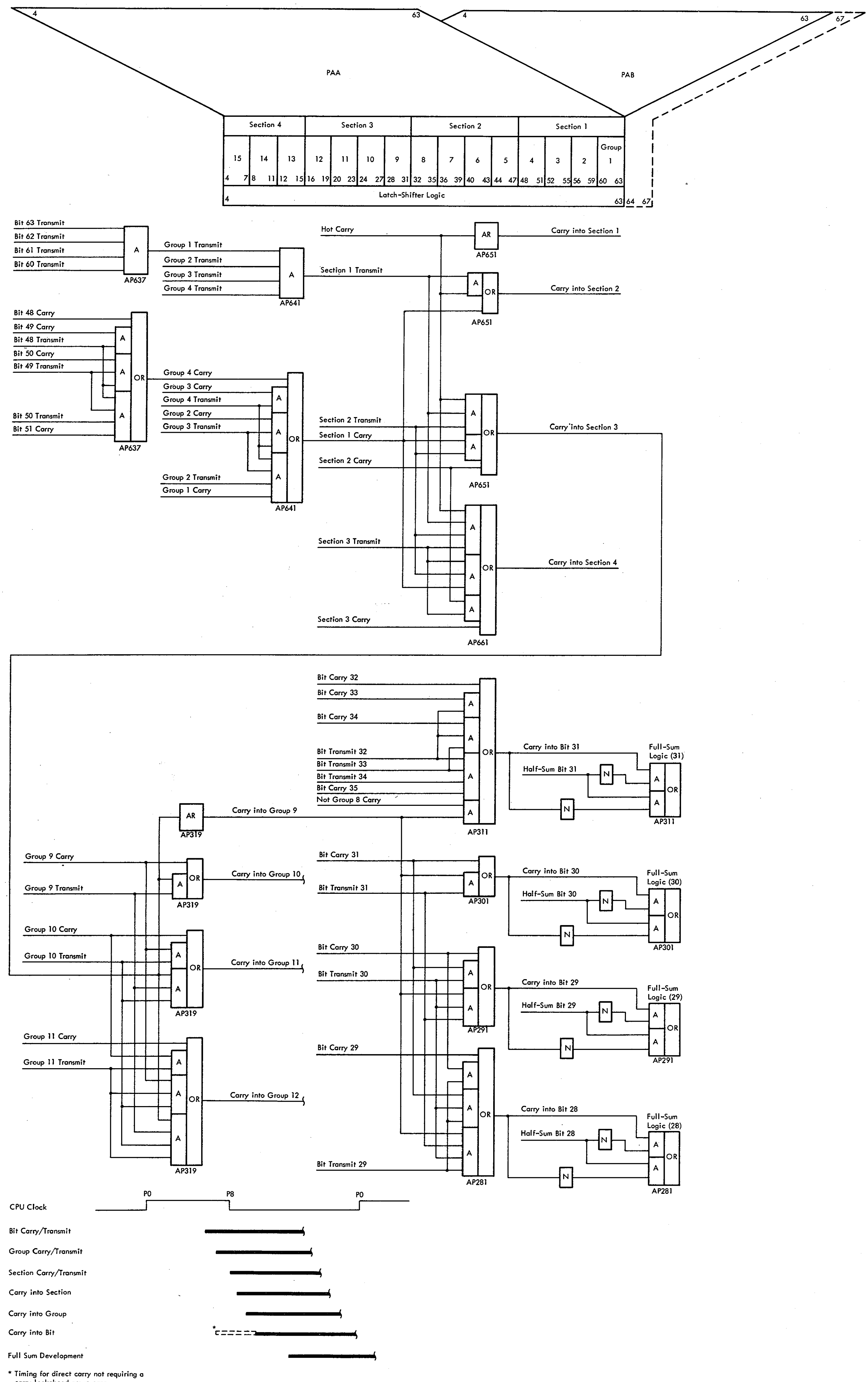


Diagram 4-411. Parallel Adder Carry Lookahead Logic

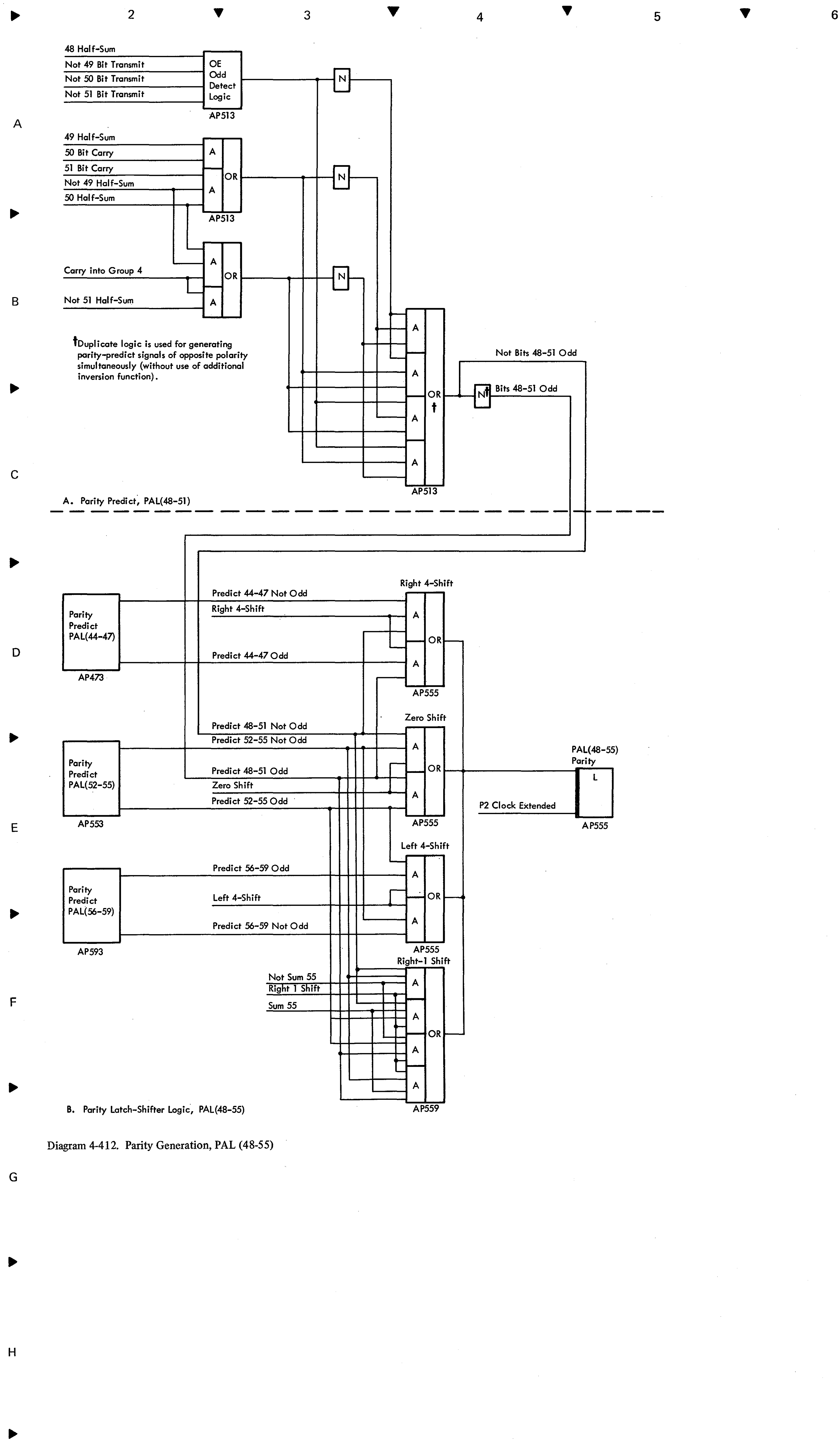


Diagram 4-412. Parity Generation, PAL (48-55)

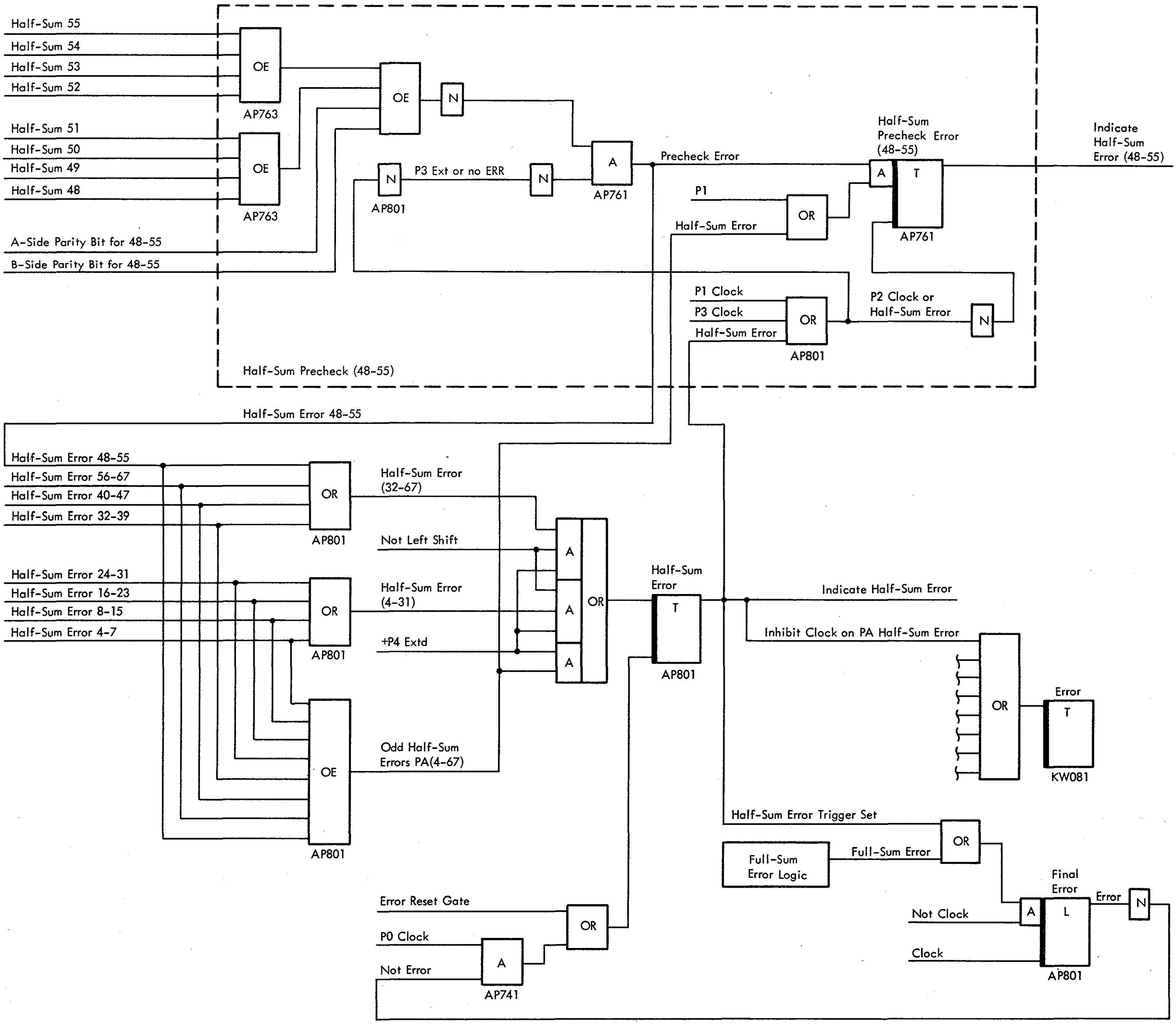


Diagram 4-413. Parallel Adder Half-Sum Checking Logic, PA (48-55)

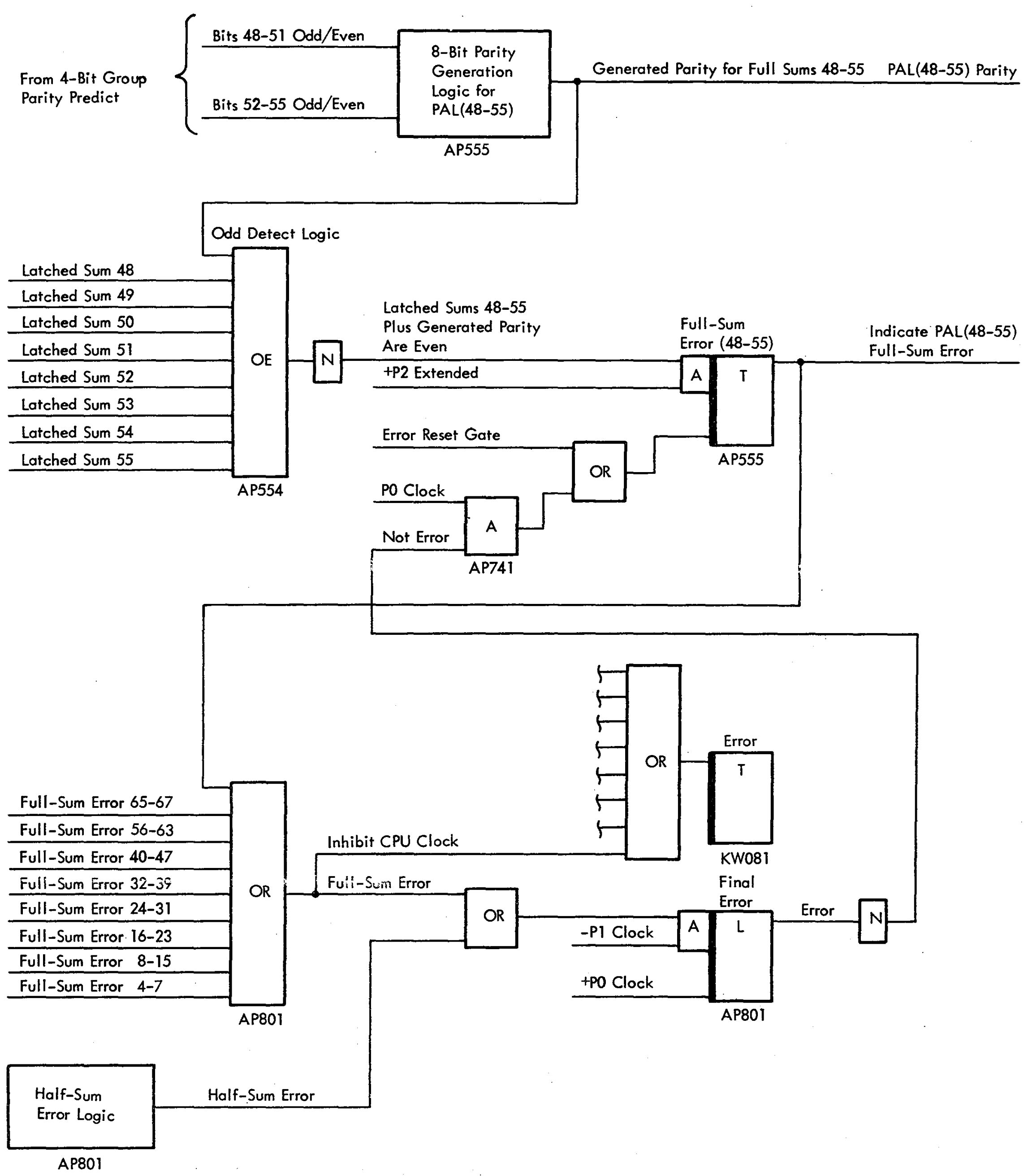


Diagram 4-414. Parallel Adder Full-Sum Checking Logic, PA(48-55)

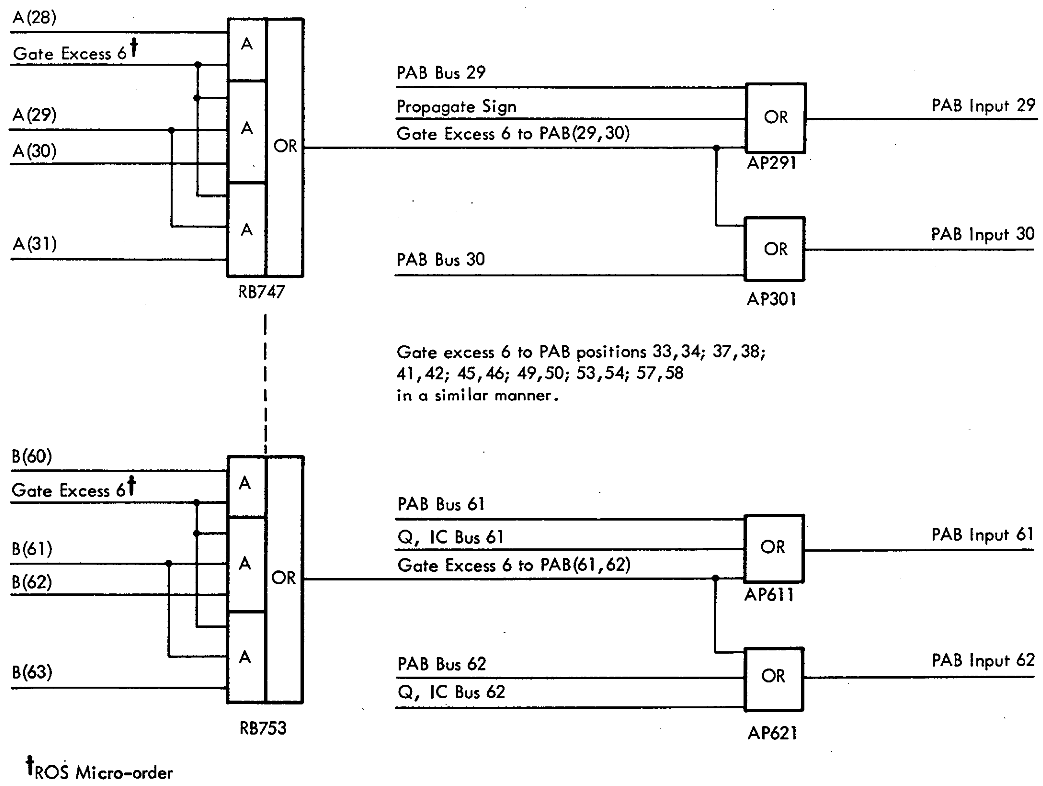


Diagram 4-415. Parallel Adder Excess 6 Logic

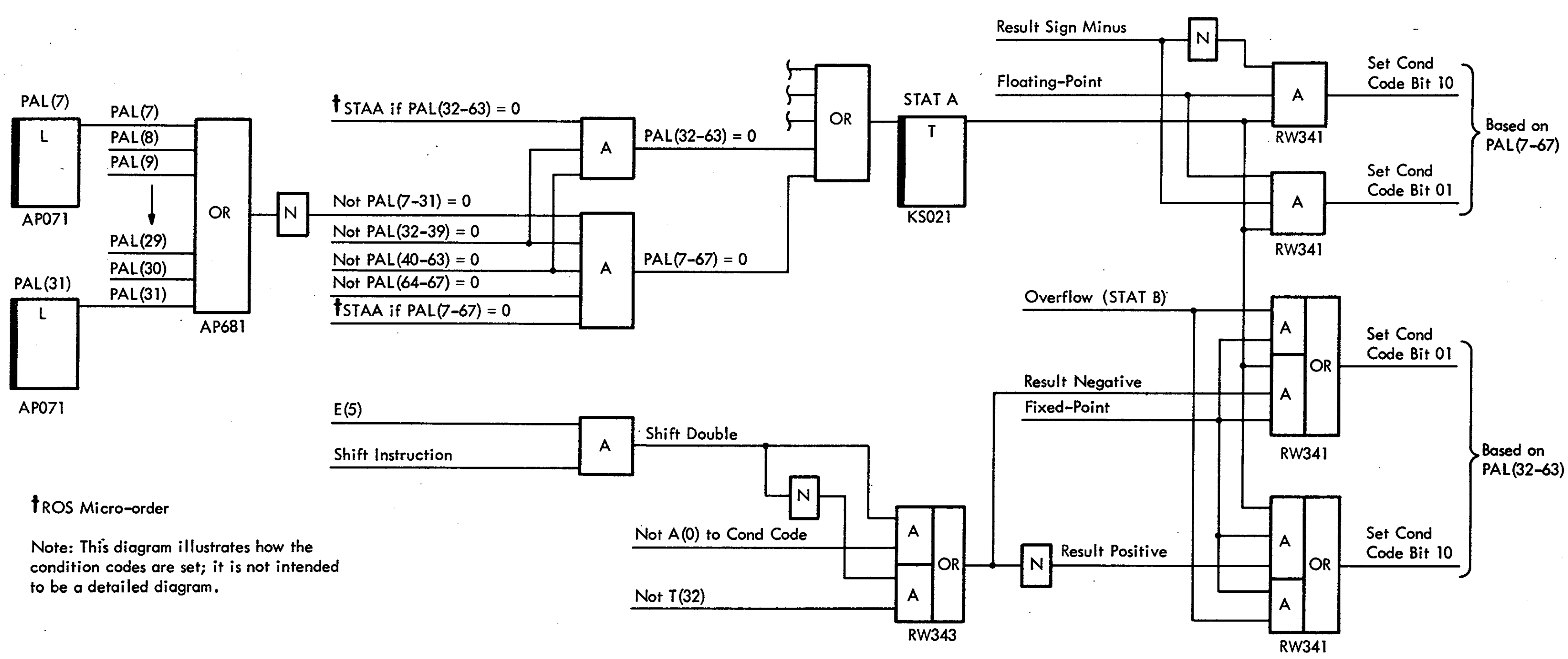


Diagram 4-416. Parallel Adder Set-Condition-Code Logic

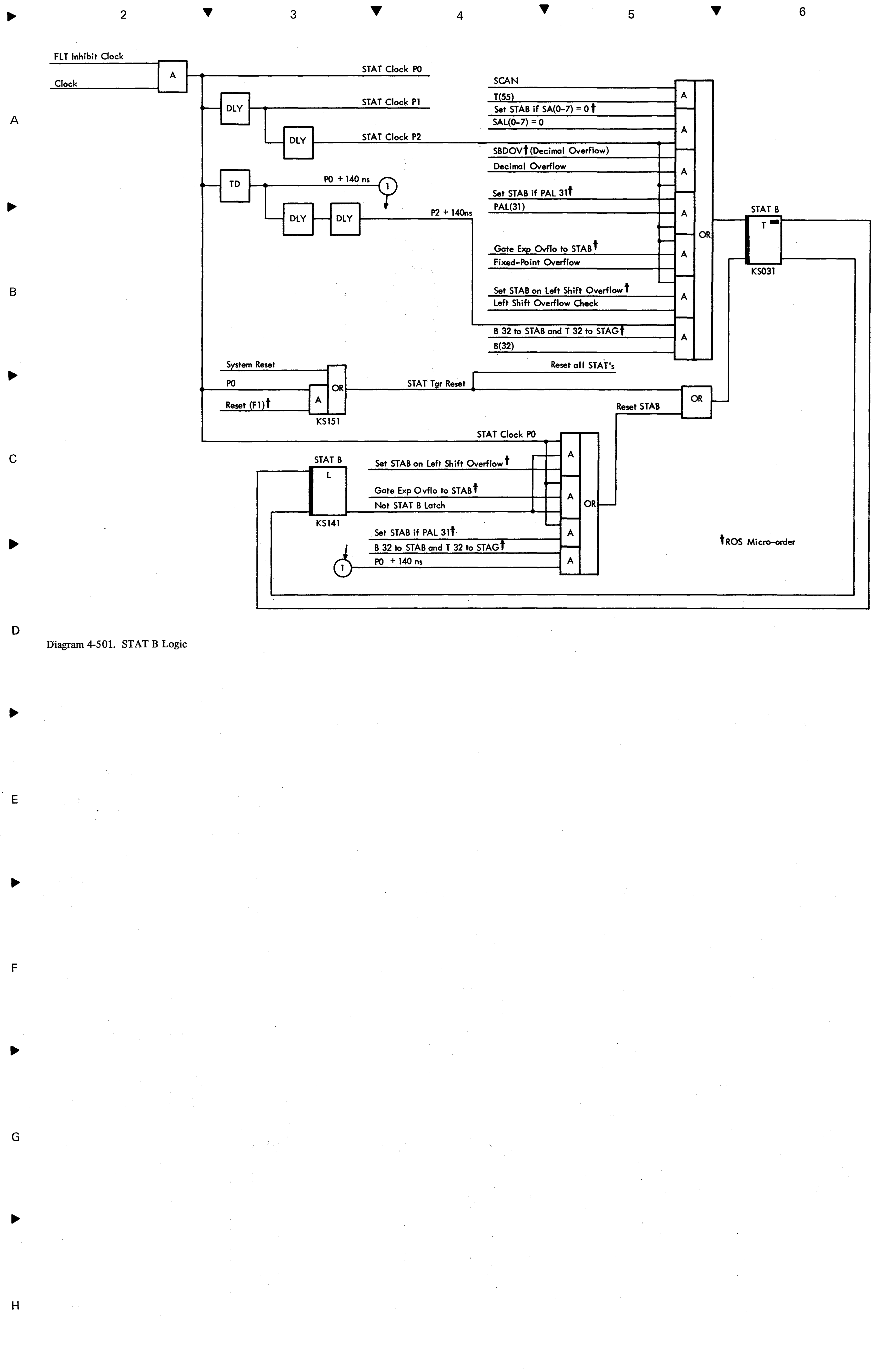


Diagram 4-501. STAT B Logic

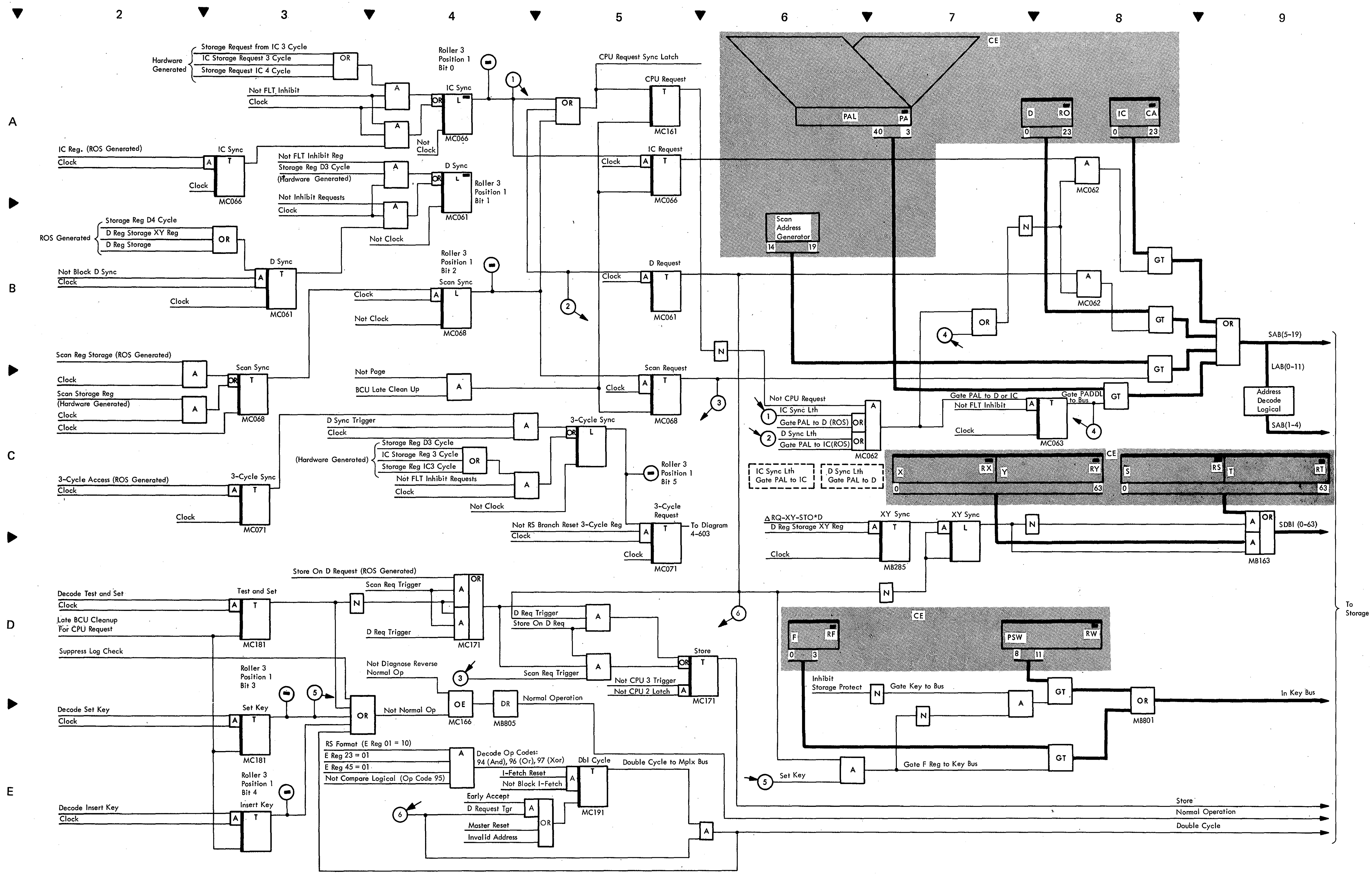


Diagram 4-601. SCI Request Sensing and Gating Logic

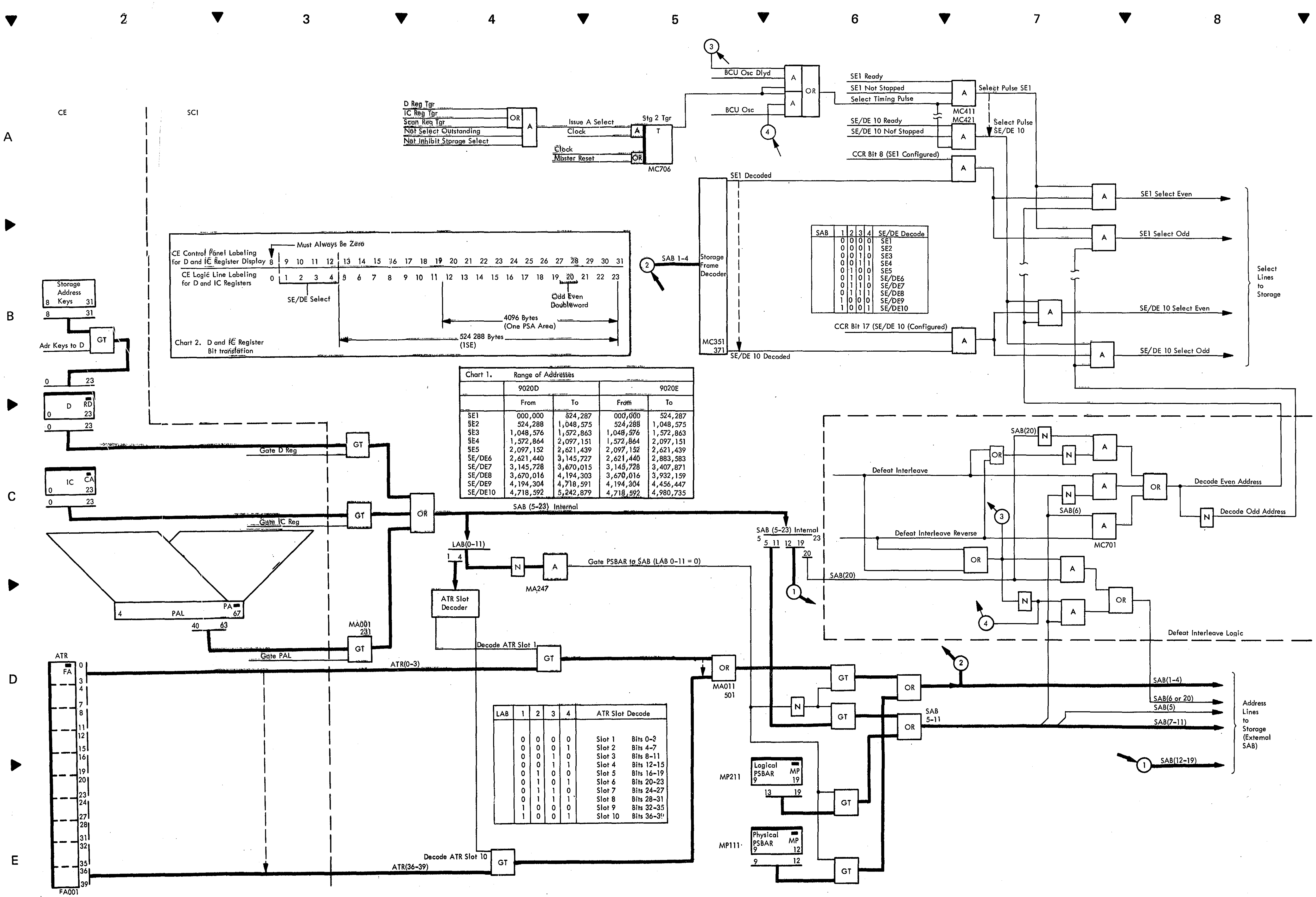


Chart 1. Range of Addresses				
	9020D		9020E	
	From	To	From	To
SE1	000,000	524,287	000,000	524,287
SE2	524,288	1,048,575	524,288	1,048,575
SE3	1,048,576	1,572,863	1,048,576	1,572,863
SE4	1,572,864	2,097,151	1,572,864	2,097,151
SE5	2,097,152	2,621,439	2,097,152	2,621,439
SE/DE6	2,621,440	3,145,727	2,621,440	2,883,583
SE/DE7	3,145,728	3,670,015	3,145,728	3,407,871
SE/DE8	3,670,016	4,194,303	3,670,016	3,932,159
SE/DE9	4,194,304	4,718,591	4,194,304	4,456,447
SE/DE10	4,718,592	5,242,879	4,718,592	4,980,735

SAB	1	2	3	4	SE/DE Decode
	0	0	0	0	SE1
	0	0	0	1	SE2
	0	0	1	0	SE3
	0	0	1	1	SE4
	0	1	0	0	SE5
	0	1	0	1	SE/DE6
	0	1	1	0	SE/DE7
	0	1	1	1	SE/DE8
	1	0	0	0	SE/DE9
	1	0	0	1	SE/DE10

LAB	1	2	3	4	ATR Slot Decode	
	0	0	0	0	Slot 1	Bits 0-3
	0	0	0	1	Slot 2	Bits 4-7
	0	0	1	0	Slot 3	Bits 8-11
	0	0	1	1	Slot 4	Bits 12-15
	0	1	0	0	Slot 5	Bits 16-19
	0	1	0	1	Slot 6	Bits 20-23
	0	1	1	0	Slot 7	Bits 24-27
	0	1	1	1	Slot 8	Bits 28-31
	1	0	0	0	Slot 9	Bits 32-35
	1	0	0	1	Slot 10	Bits 36-39

Diagram 4-602, Address Decode and Gating Logic

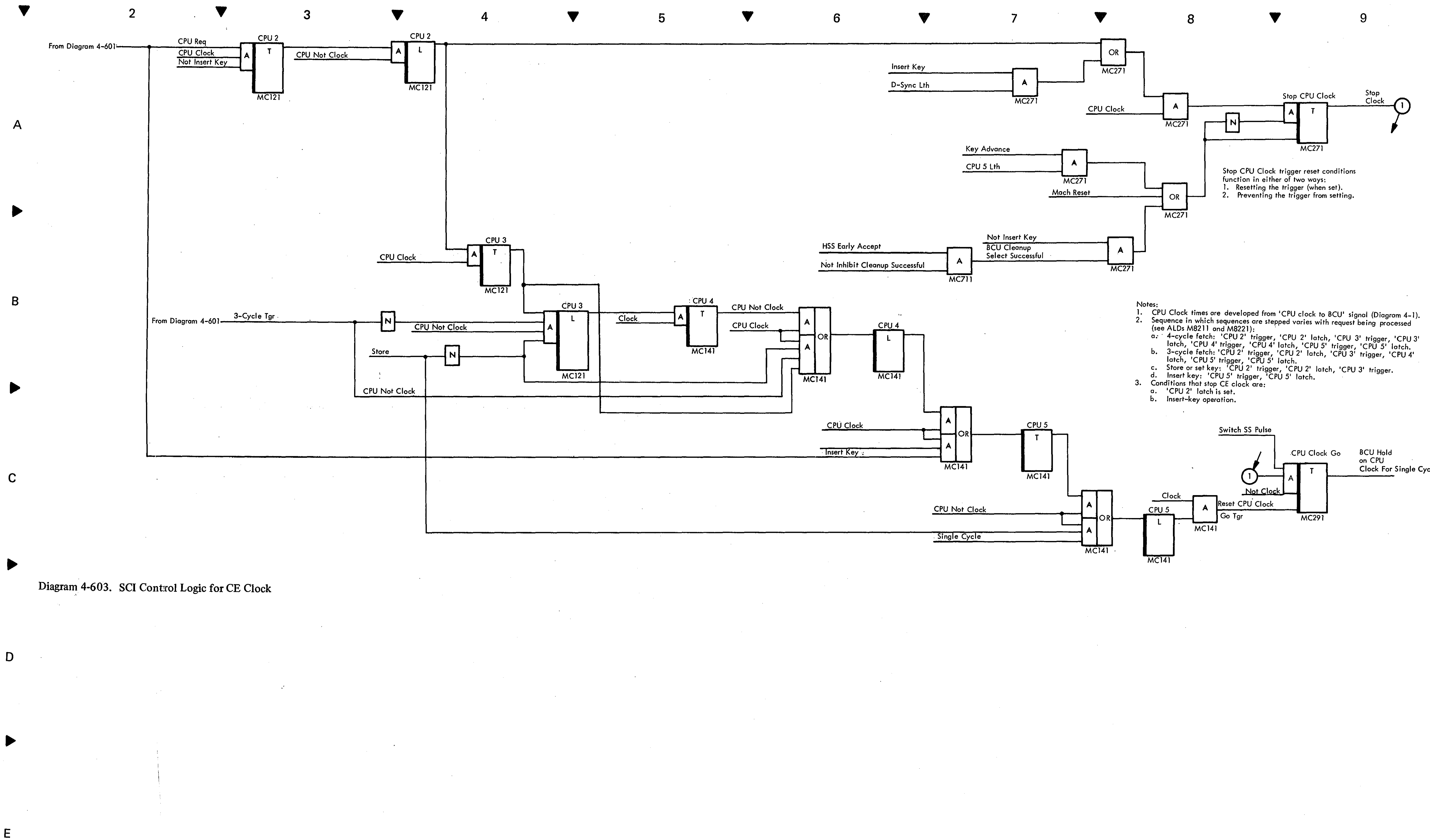


Diagram 4-603. SCI Control Logic for CE Clock

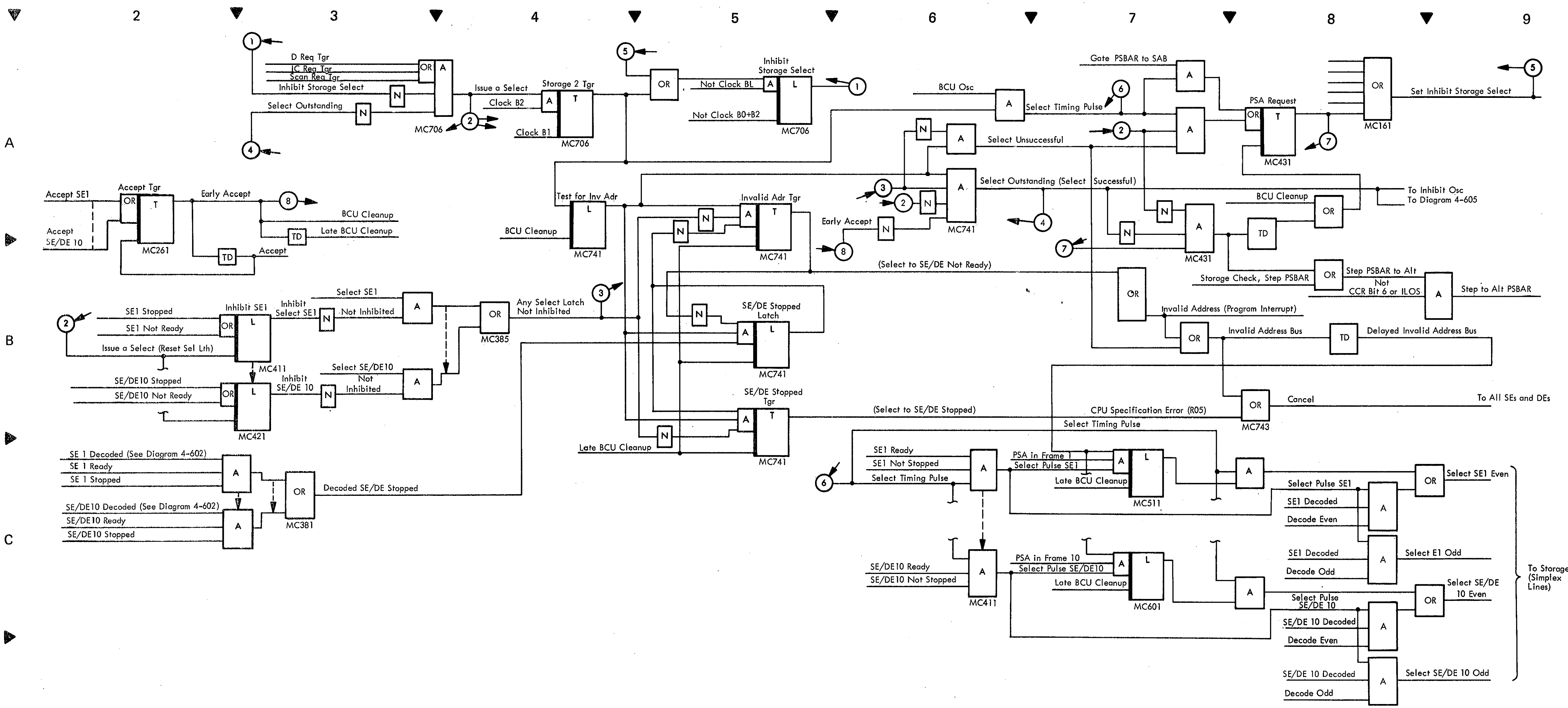


Diagram 4-604. Invalid Address and Frame Stopped Logic (Sheet 1 of 2)

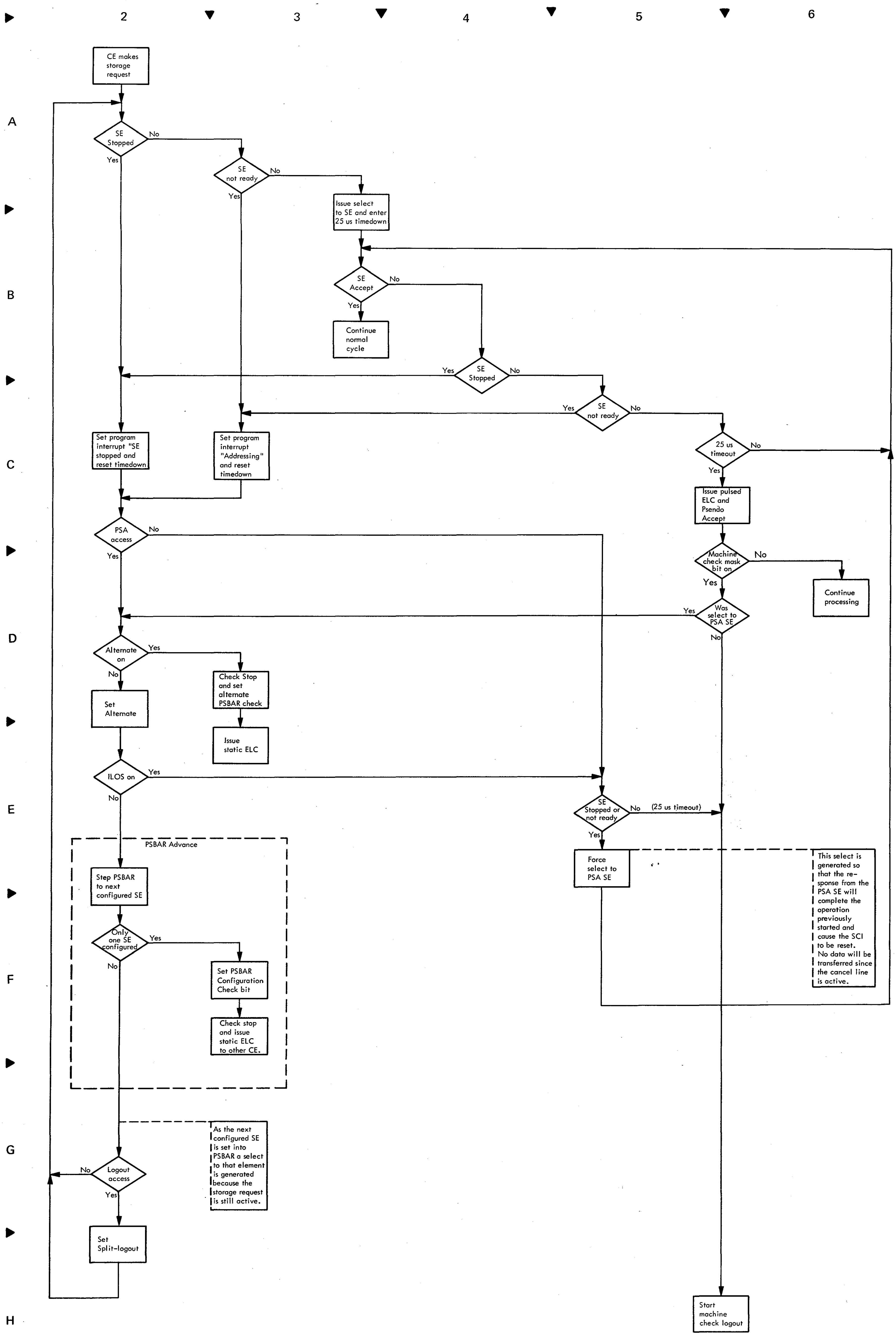
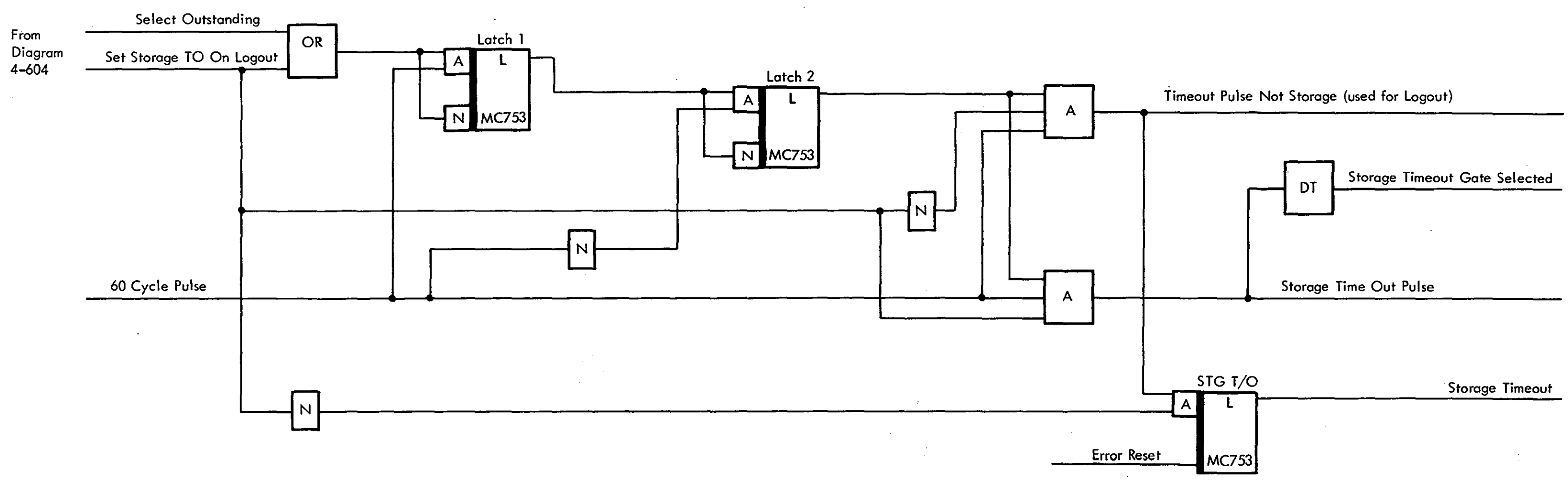
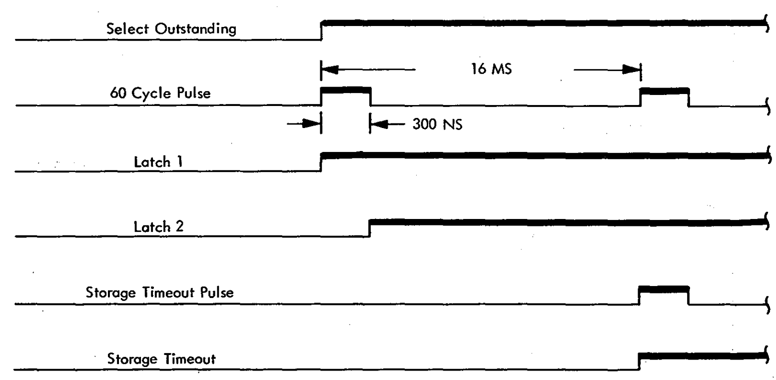


Diagram 4-604. Invalid Address and Frame Stopped Logic (Sheet 2 of 2)





Notes:
1. 'Storage timeout' is activated if 'select outstanding' remains active through two '60-cycle pulses'.
2. 'Select outstanding' may come at any time; the example shown is for the least amount of time necessary to set 'storage timeout'.
3. 'Select outstanding' is deactivated by 'accept' from storage.

Diagram 4-605. Storage Timeout Logic

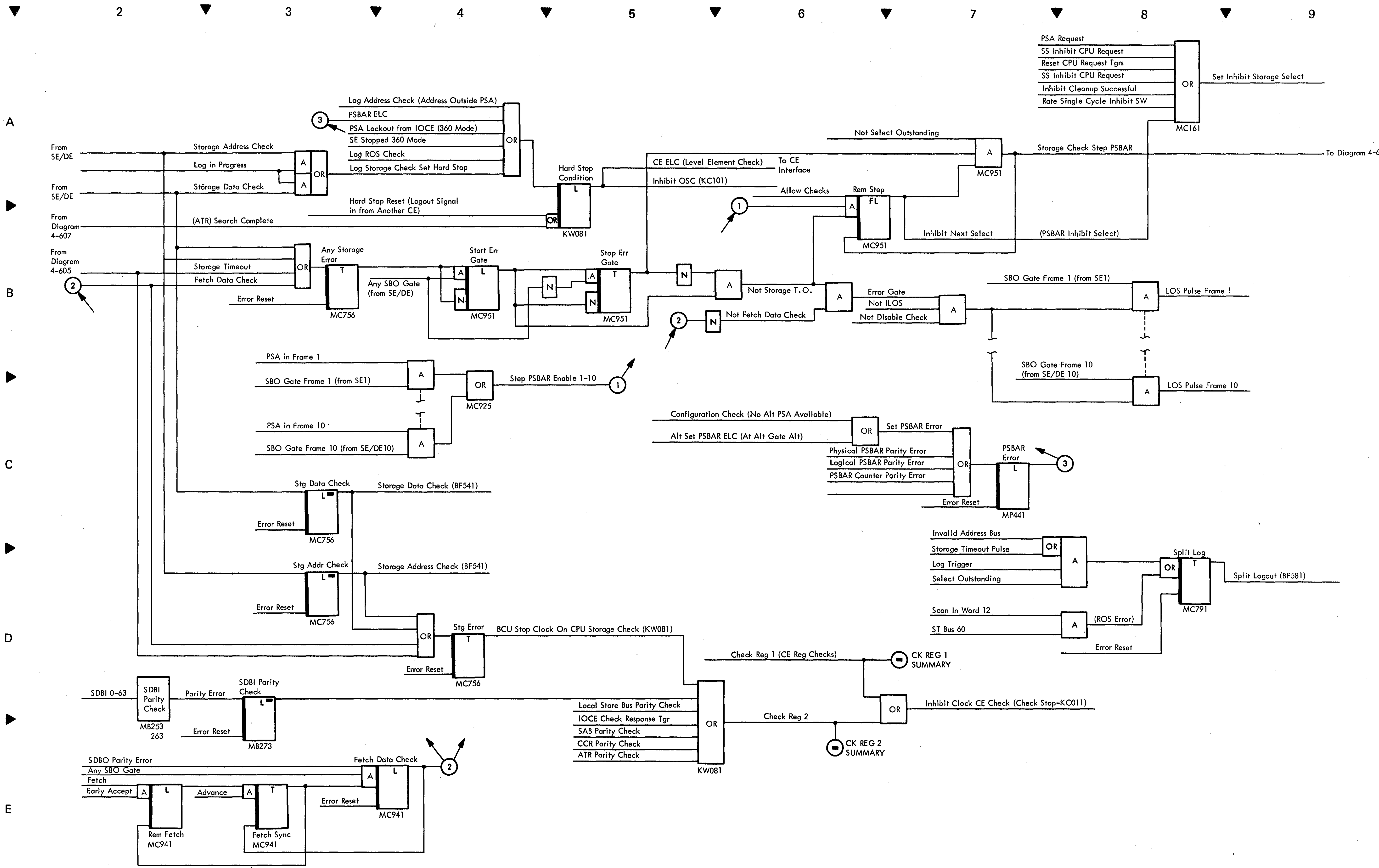
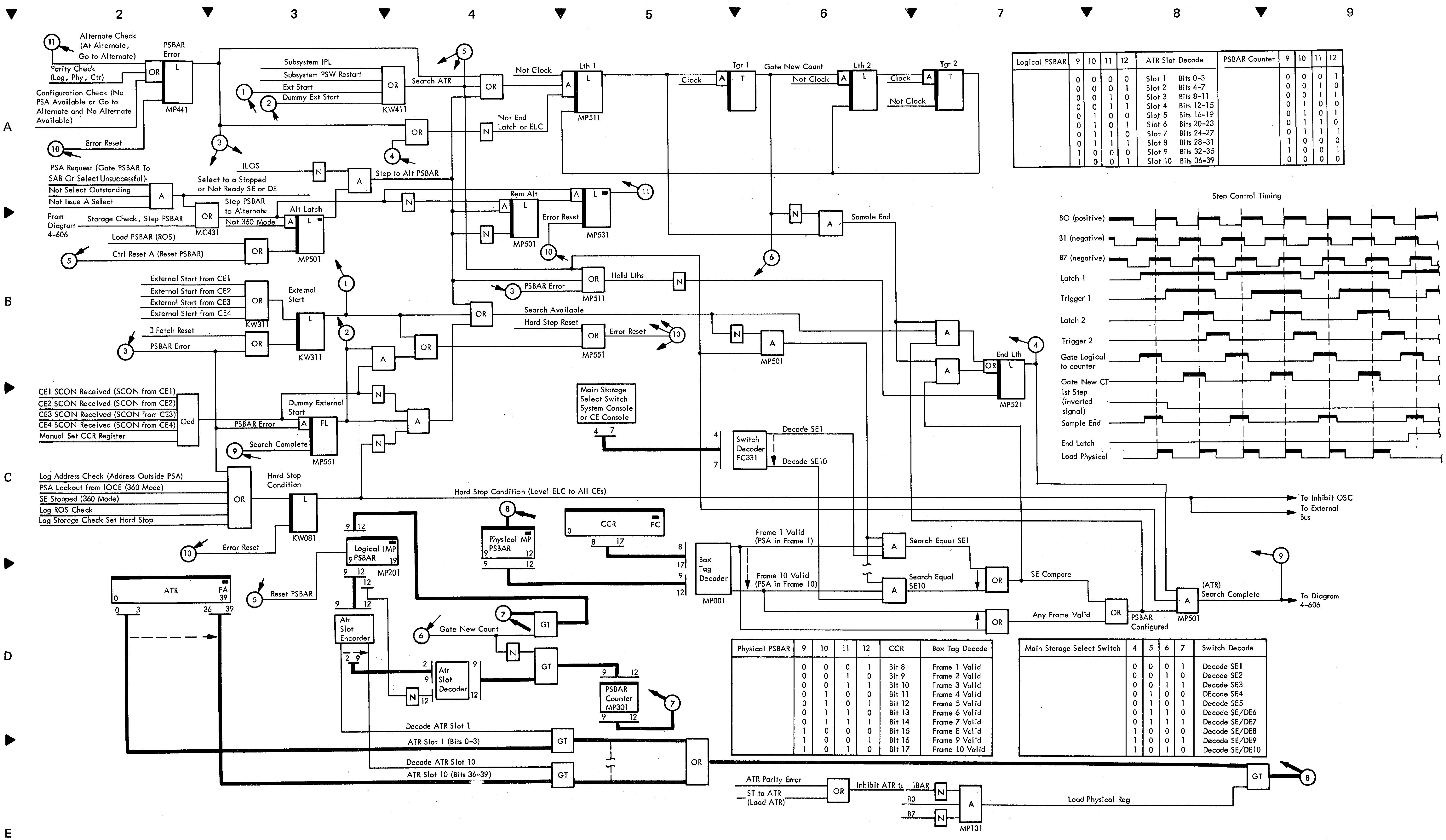


Diagram 4-606. Error Handling Logic



Logical PSBAR	9	10	11	12	ATR Slot Decode	PSBAR Counter	9	10	11	12
	0	0	0	0	Slot 1 Bits 0-3		0	0	0	1
	0	0	0	1	Slot 2 Bits 4-7		0	0	1	0
	0	0	1	0	Slot 3 Bits 8-11		0	0	1	1
	0	0	1	1	Slot 4 Bits 12-15		0	1	0	0
	0	1	0	0	Slot 5 Bits 16-19		0	1	0	1
	0	1	0	1	Slot 6 Bits 20-23		0	1	1	0
	0	1	1	0	Slot 7 Bits 24-27		0	1	1	1
	0	1	1	1	Slot 8 Bits 28-31		1	0	0	0
	1	0	0	0	Slot 9 Bits 32-35		1	0	0	1
	1	0	0	1	Slot 10 Bits 36-39		0	0	0	0

Physical PSBAR	9	10	11	12	CCR	Box Tag Decode
	0	0	0	1	Bit 8	Frame 1 Valid
	0	0	1	0	Bit 9	Frame 2 Valid
	0	0	1	1	Bit 10	Frame 3 Valid
	0	1	0	0	Bit 11	Frame 4 Valid
	0	1	0	1	Bit 12	Frame 5 Valid
	0	1	1	0	Bit 13	Frame 6 Valid
	0	1	1	1	Bit 14	Frame 7 Valid
	1	0	0	0	Bit 15	Frame 8 Valid
	1	0	0	1	Bit 16	Frame 9 Valid
	1	0	1	0	Bit 17	Frame 10 Valid

Main Storage Select Switch	4	5	6	7	Switch Decode
	0	0	0	1	Decode SE1
	0	0	1	0	Decode SE2
	0	0	1	1	Decode SE3
	0	1	0	0	DEcode SE4
	0	1	0	1	Decode SE5
	0	1	1	0	Decode SE/DE6
	0	1	1	1	Decode SE/DE7
	1	0	0	0	Decode SE/DE8
	1	0	0	1	Decode SE/DE9
	1	0	1	0	Decode SE/DE10

Diagram 4-607. PSBAR Step Control Logic

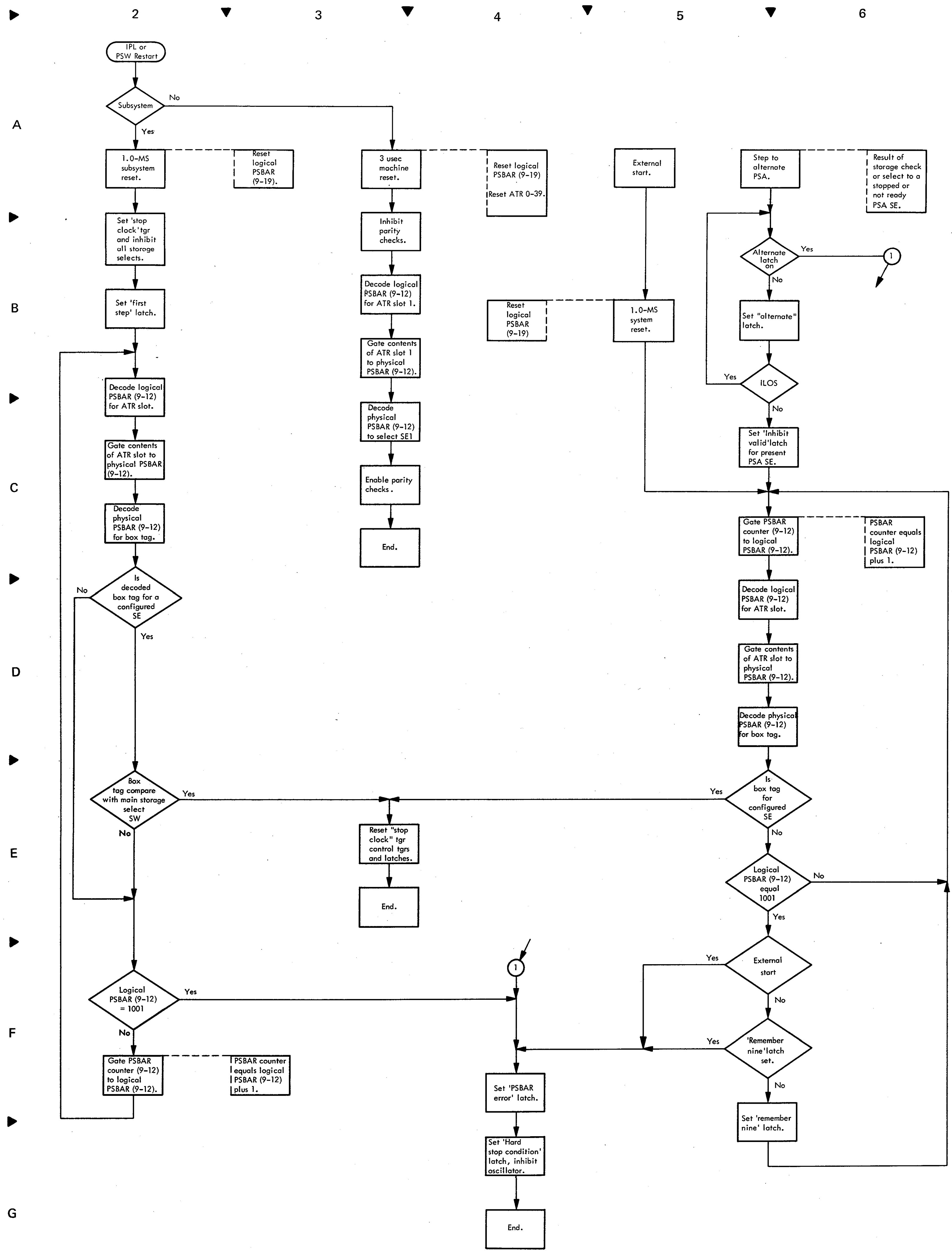


Diagram 4-608. PSBAR Operations

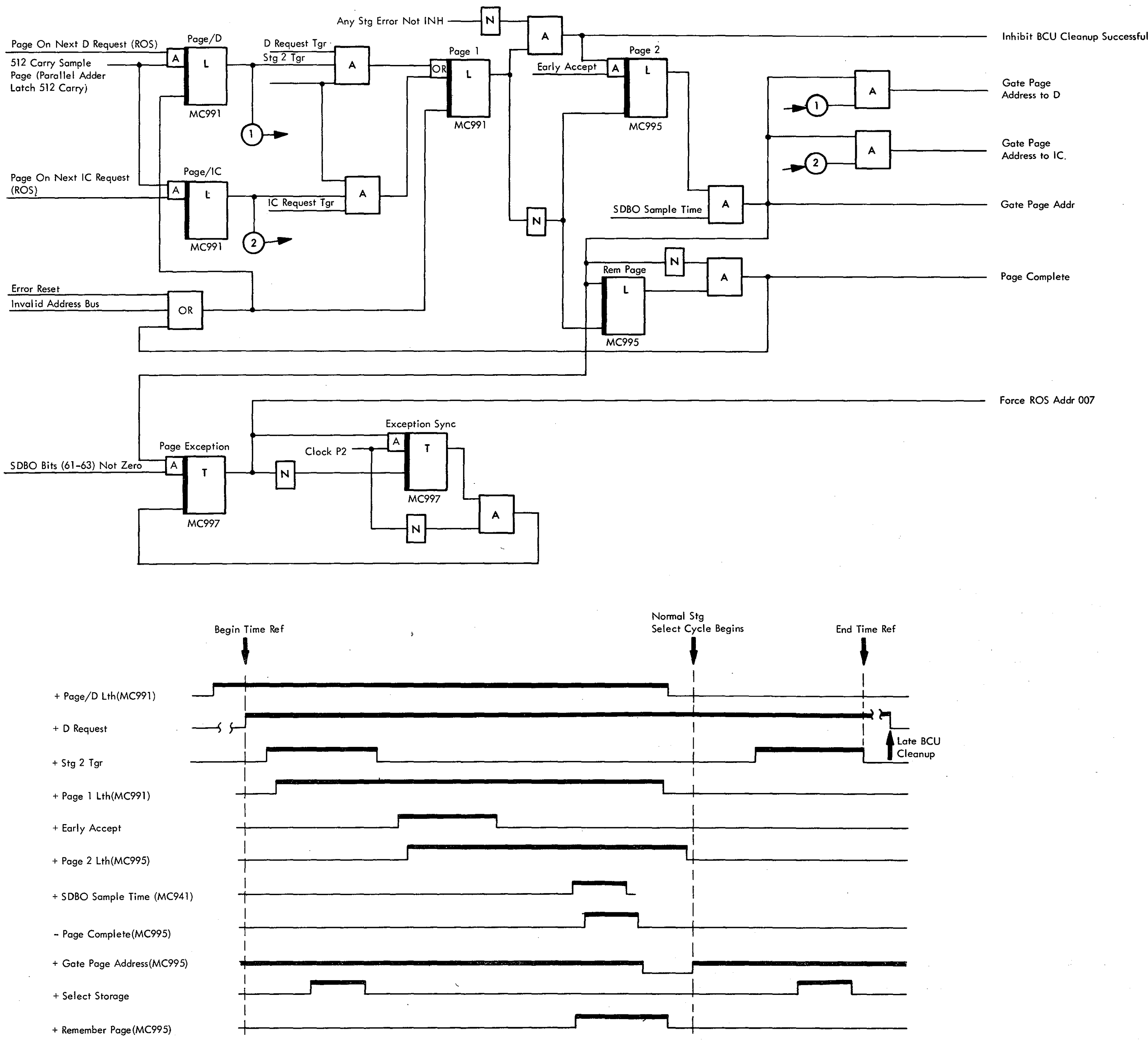


Diagram 4-609. Page Control Logic and Timing

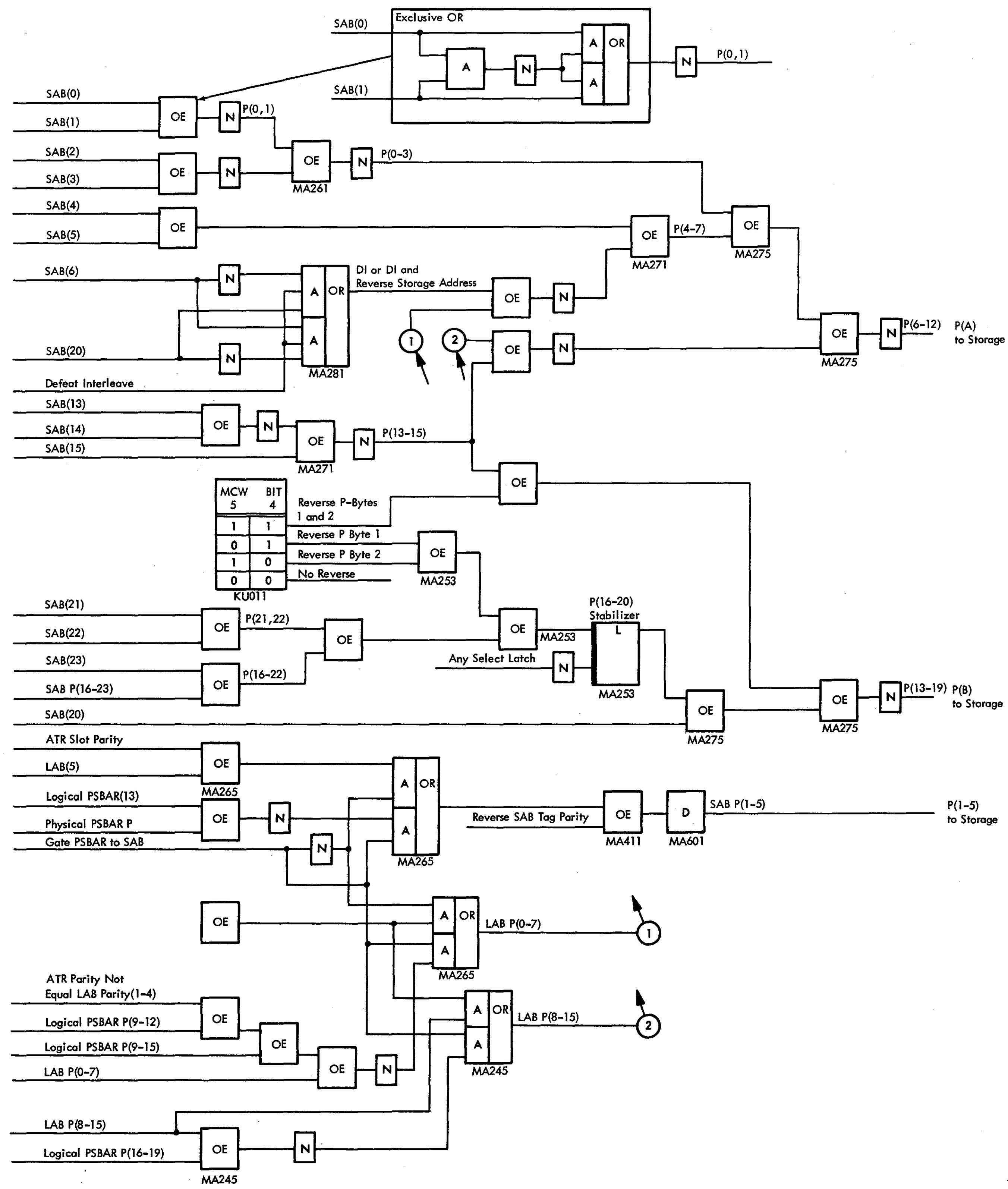


Diagram 4-610. SAB Parity Conversion Logic

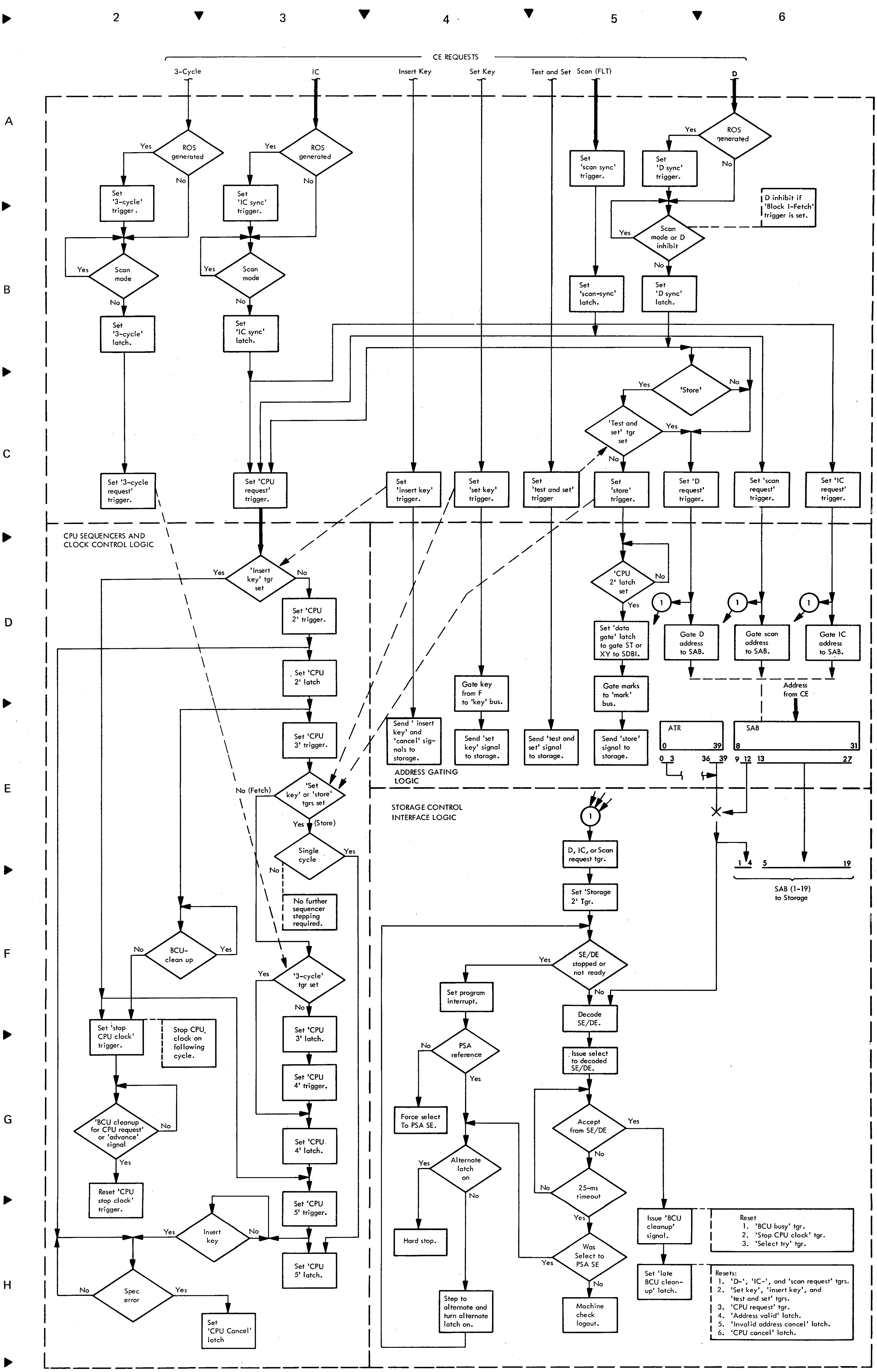


Diagram 4-611. Detailed SCI Functional Sequence (Sheet 1 of 2)

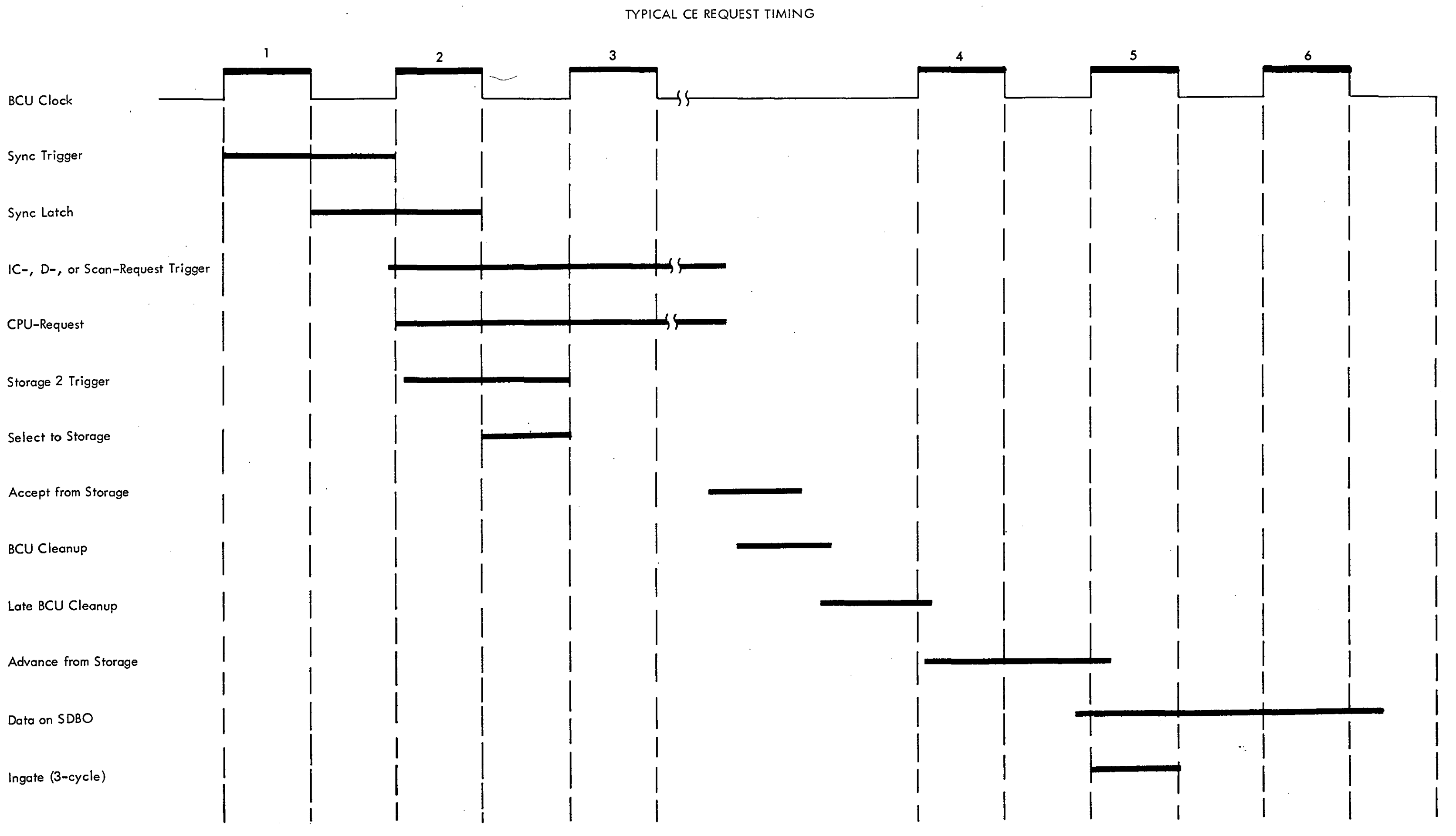


Diagram 4-611. Servicing of Storage Requests in Single-Cycle Mode (Sheet 2 of 2)

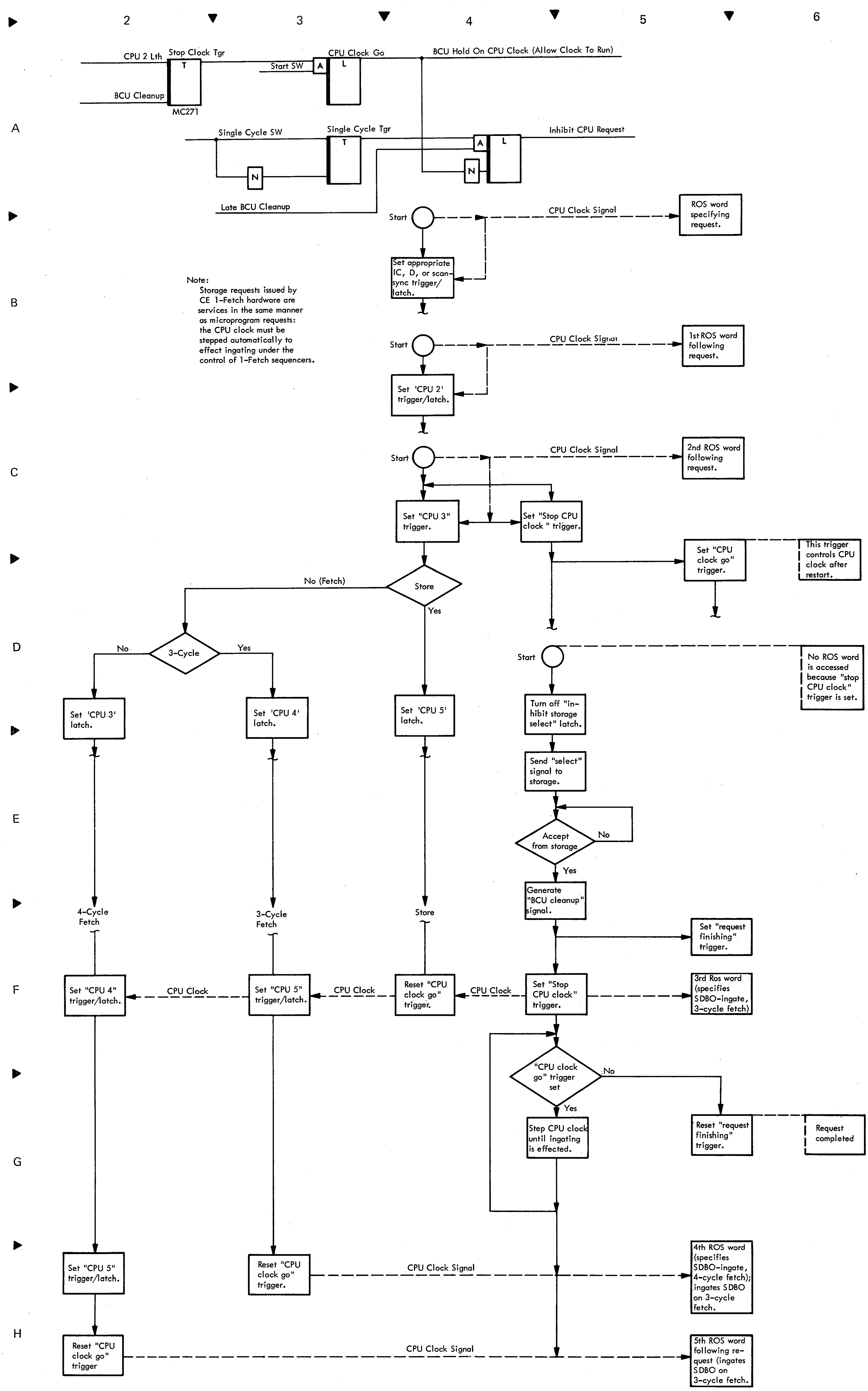


Diagram 4-612. Servicing of Storage Requests in Single-Cycle Mode

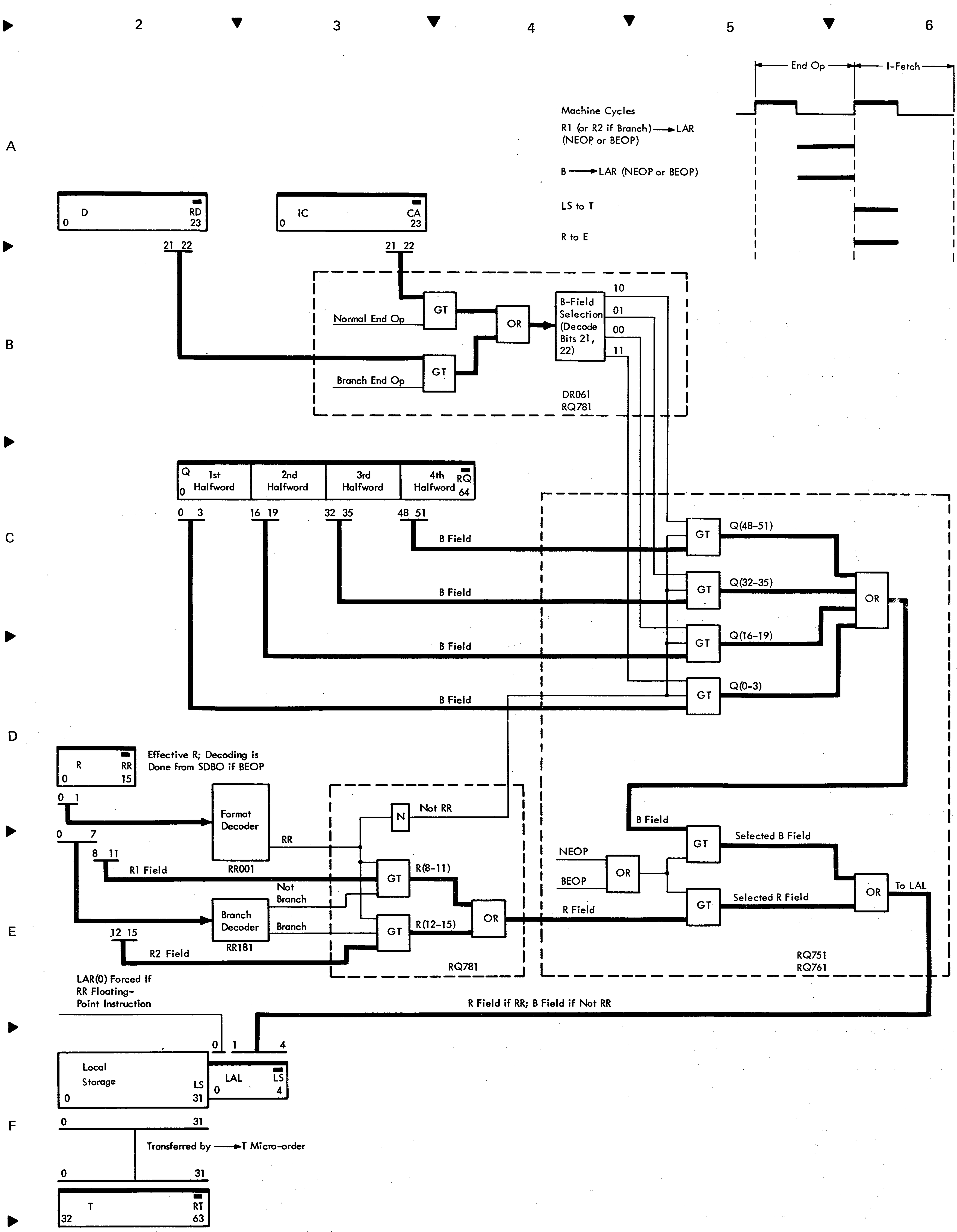


Diagram 5-1. Operand Prefetching During End Op

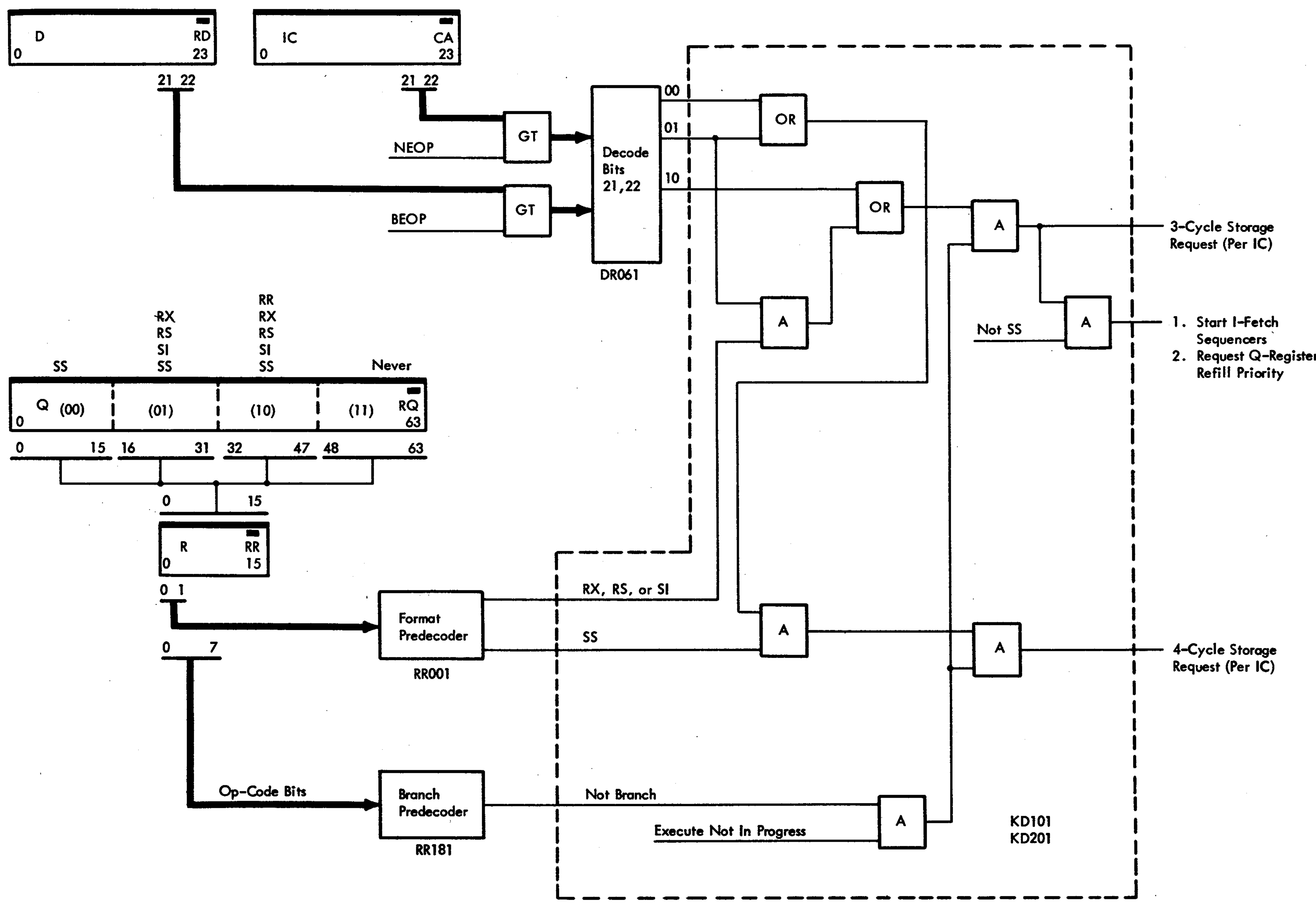


Diagram 5-2. Instruction Requests During End Op

DIAGRAM 5-2. INSTRUCTION REQUESTS DURING END OP

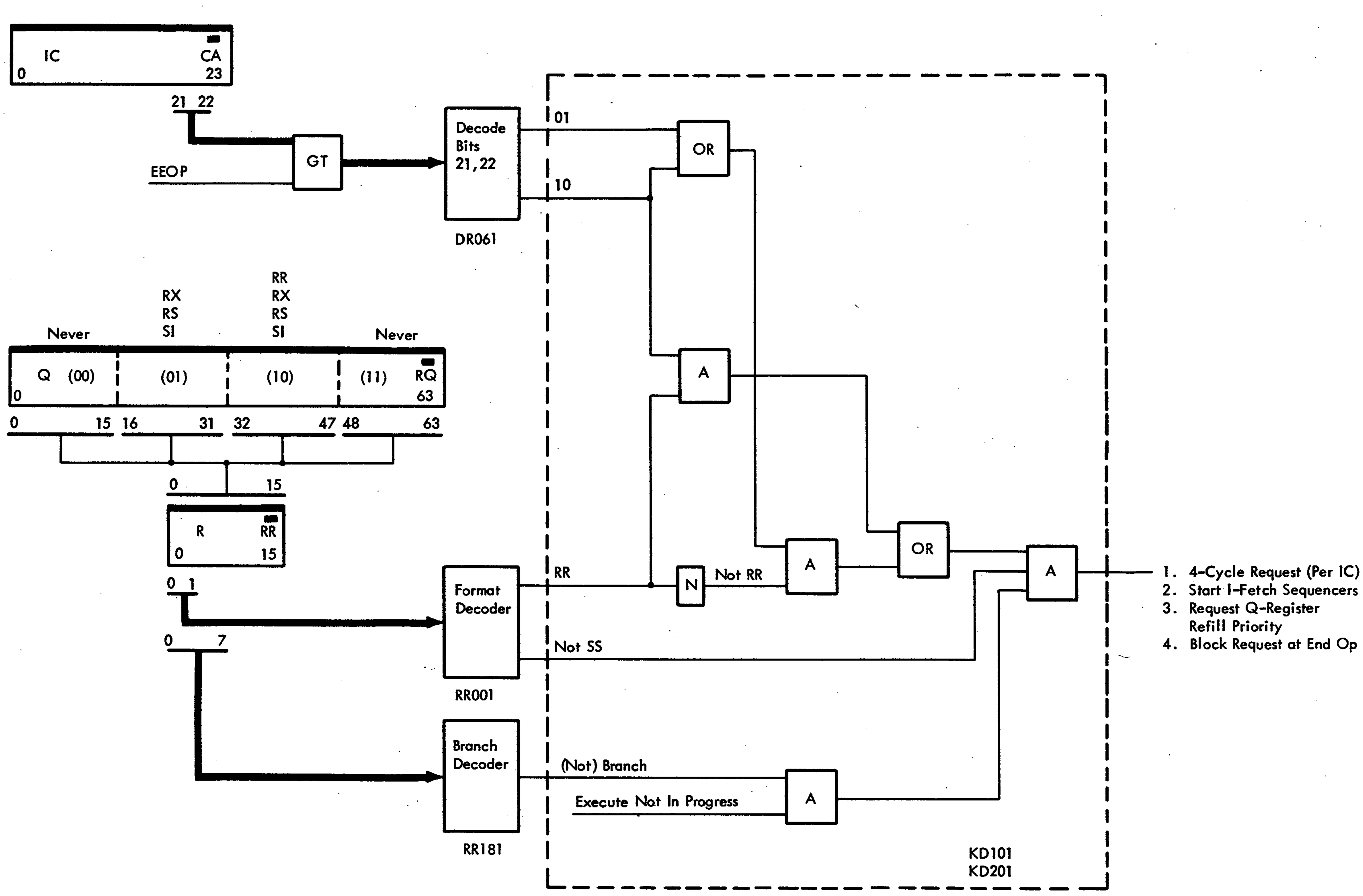


Diagram 5-3. Instruction Requests During Early End Op

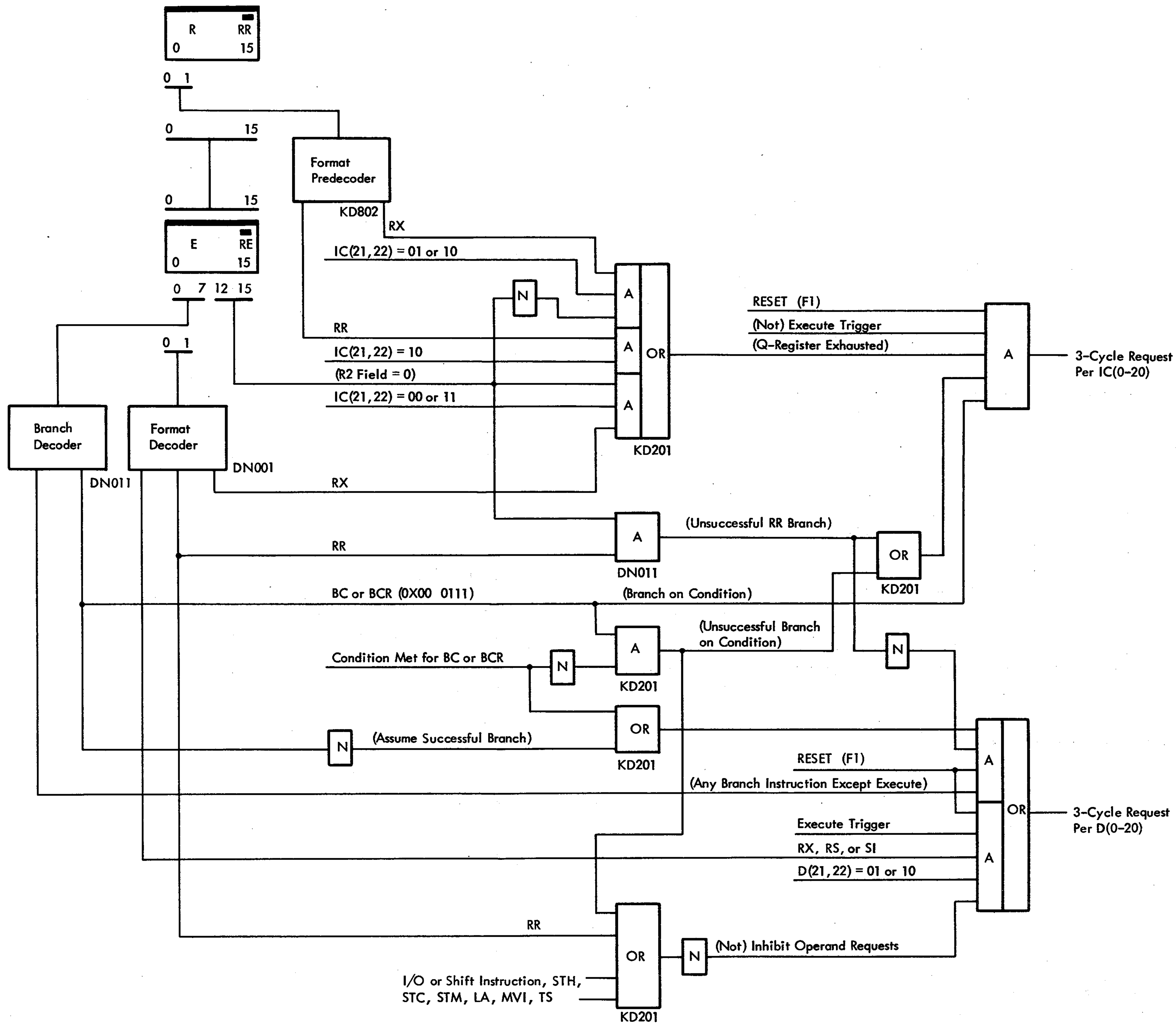


Diagram 5-4. Branch Requests

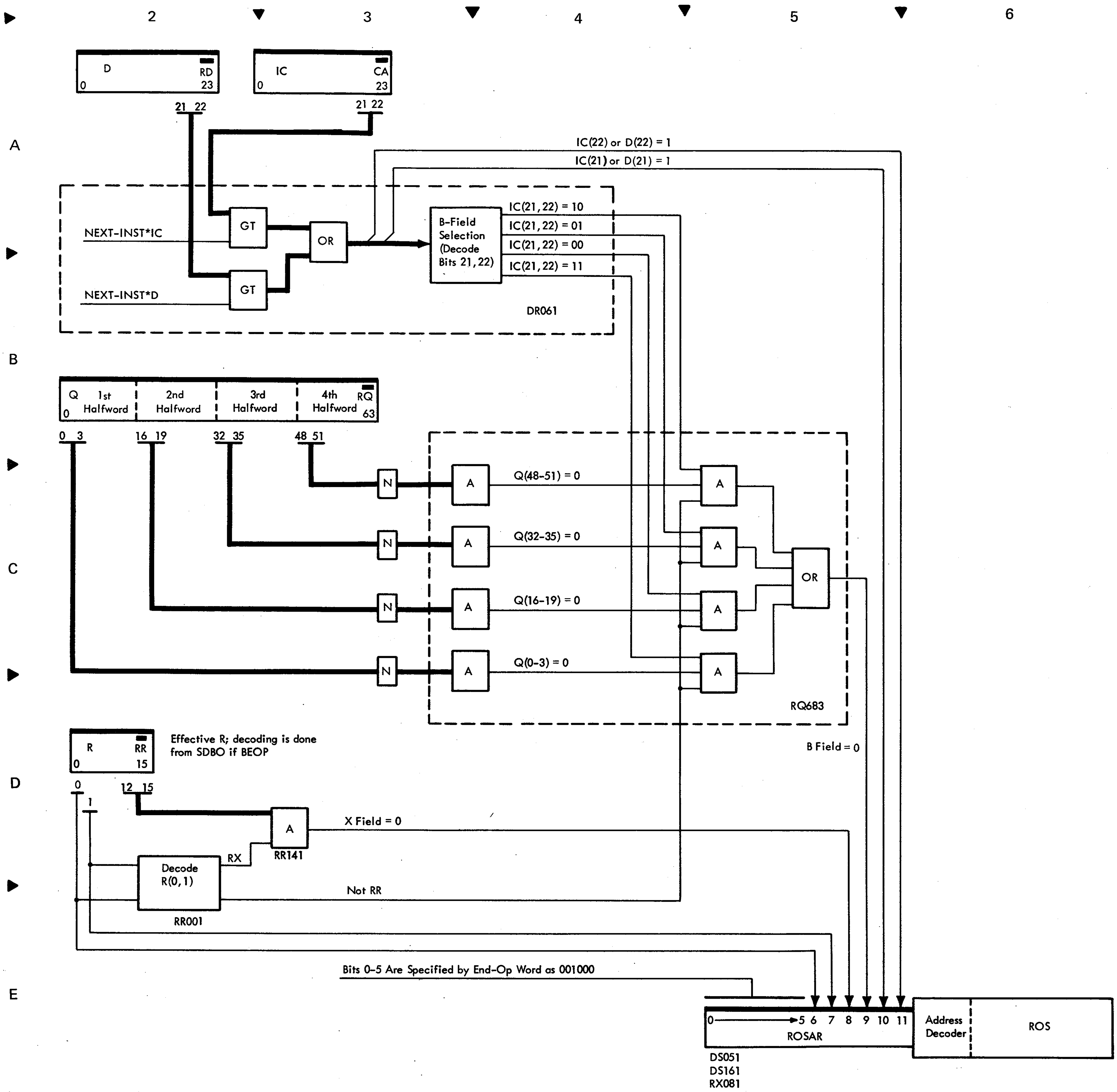


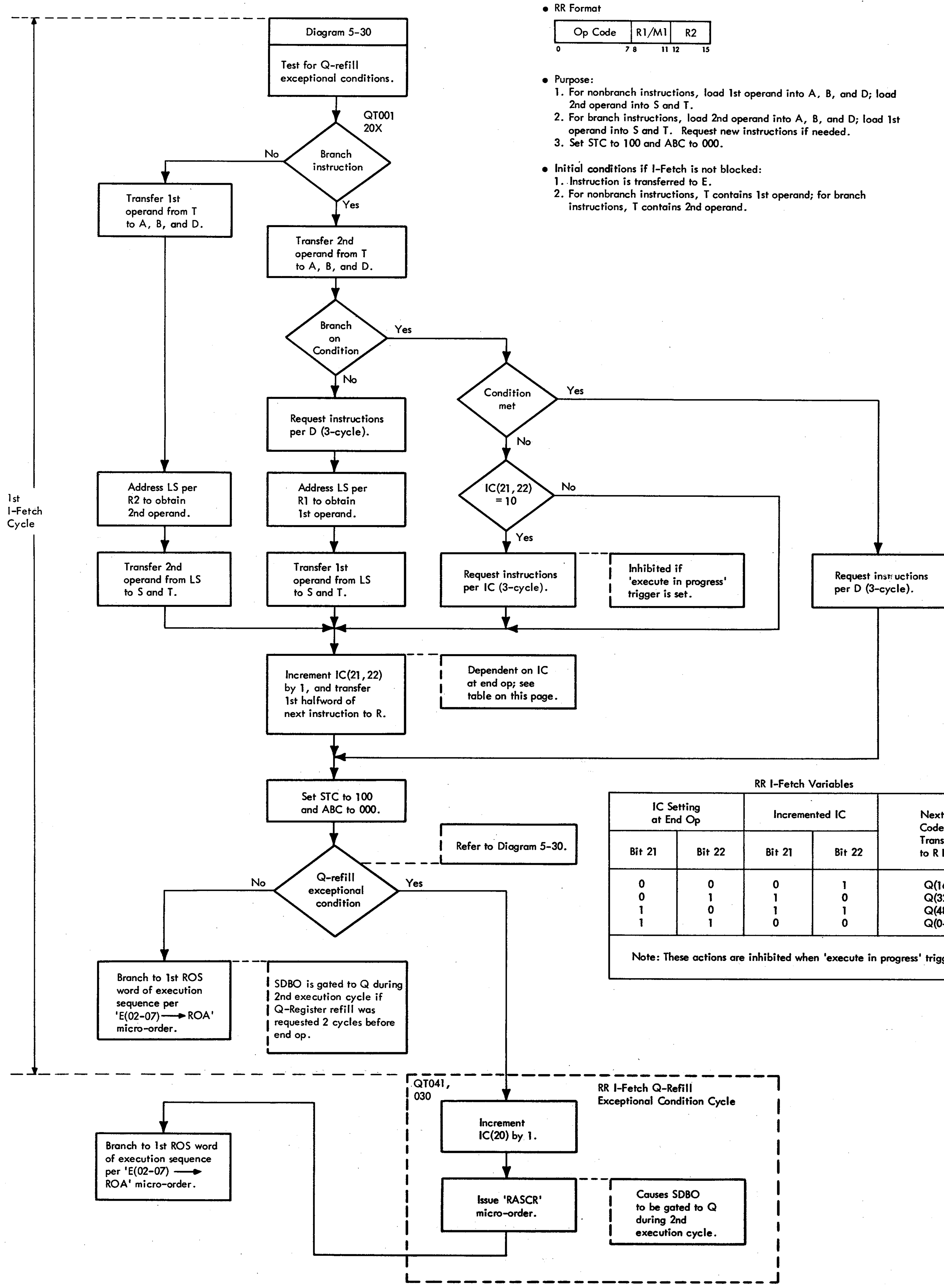
Diagram 5-5. Selection of I-Fetch Sequence

- RR Format

Op Code	R1/M1	R2
0–7	8–11	12–15

- Purpose:
  1. For nonbranch instructions, load 1st operand into A, B, and D; load 2nd operand into S and T.
  2. For branch instructions, load 2nd operand into A, B, and D; load 1st operand into S and T. Request new instructions if needed.
  3. Set STC to 100 and ABC to 000.

- Initial conditions if I-Fetch is not blocked:
  1. Instruction is transferred to E.
  2. For nonbranch instructions, T contains 1st operand; for branch instructions, T contains 2nd operand.



RR I-Fetch Variables

IC Setting at End Op		Incremented IC		Next Op-Code Word Transferred to R From:
Bit 21	Bit 22	Bit 21	Bit 22	
0	0	0	1	Q(16-31)
0	1	1	0	Q(32-47)
1	0	1	1	Q(48-63)
1	1	0	0	Q(0-15)

Note: These actions are inhibited when 'execute in progress' trigger is set.

Diagram 5-6. RR I-Fetch

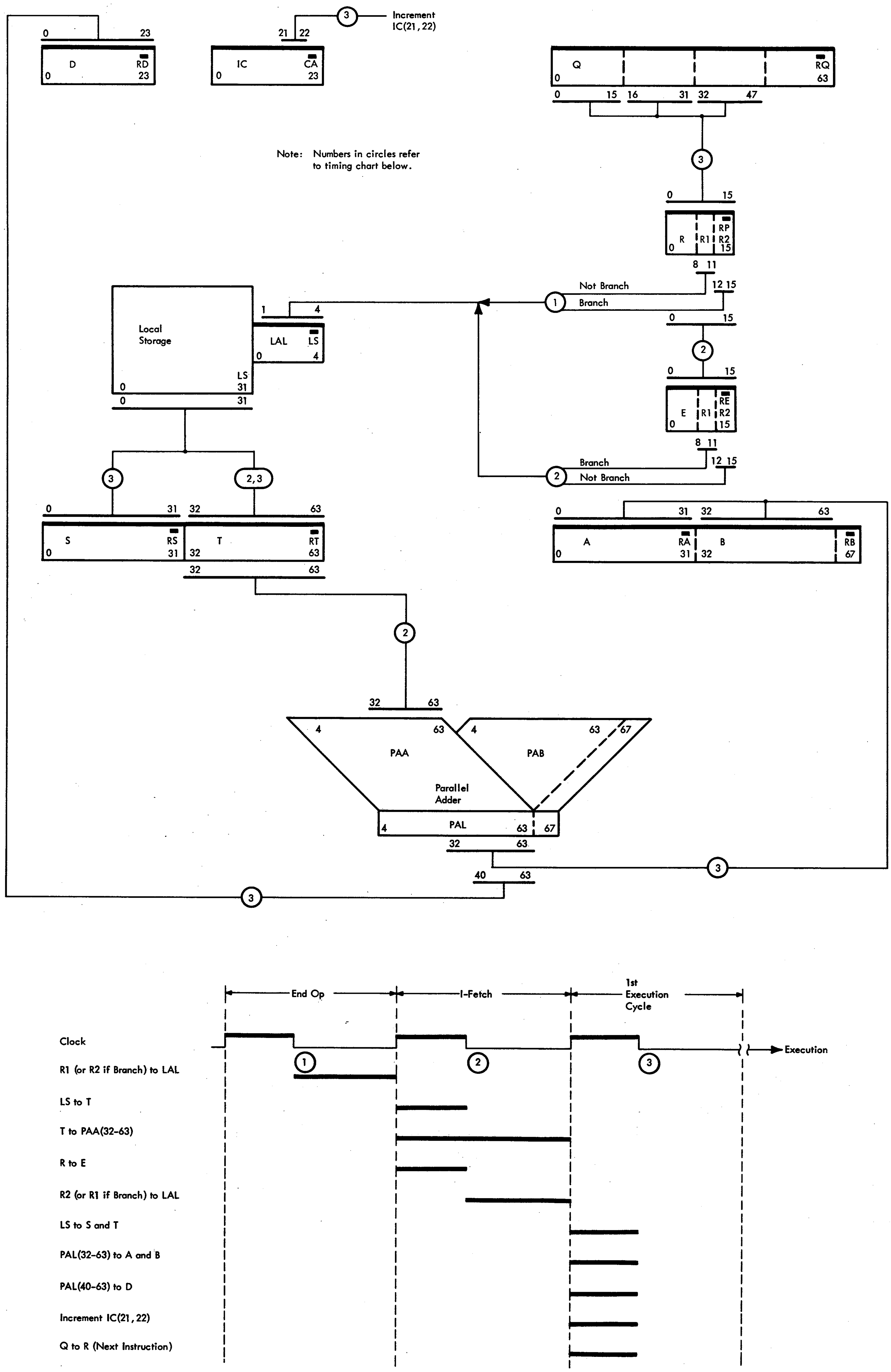


Diagram 5-7. One-Cycle RR I-Fetch

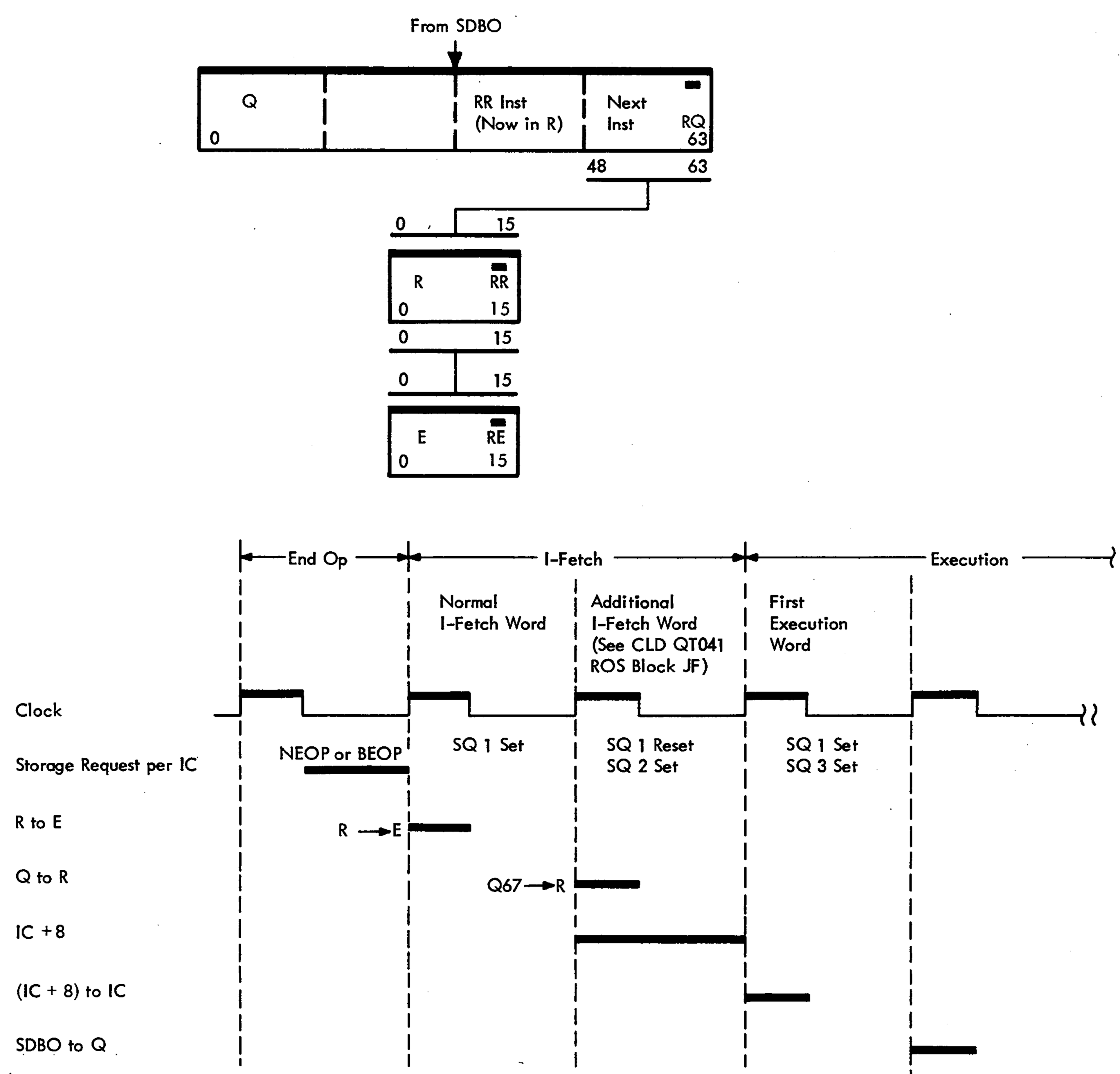


Diagram 5-8. Two-Cycle RR I-Fetch

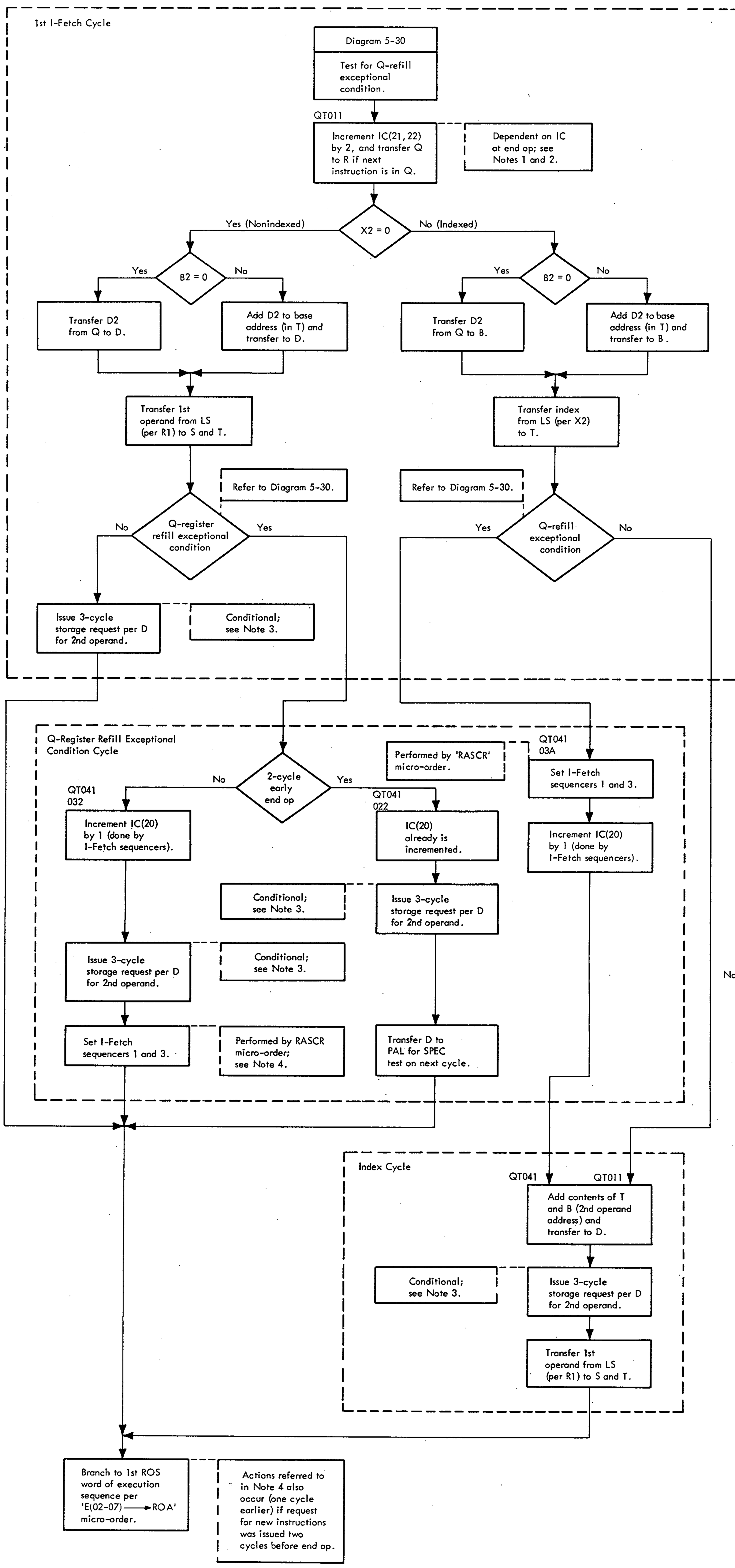


Diagram 5-9. RX I-Fetch

- RX Format

Op Code	R1	X2	B2	D2
0–7	8–11	12–15	16–19	20–31

- Purpose:
  1. Transfer 1st operand to S and T.
  2. Compute address of 2nd operand and transfer to D.
  3. Request 2nd operand from main storage (see Note 3).

- Initial conditions if I-Fetch is not blocked:
  1. 1st halfword of instruction is transferred to E; 2nd halfword is in Q.
  2. Contents of LS register specified by B2 are transferred to T.

Notes:

1. IC Incrementing and Q to R Transfer:

Conditions at End Op: IC(21,22)	Conditions at End Op: Position in Q	Incremented IC(21,22)	Next Op-Code Word Transferred to R From:
00		10	Q(32-47)
01	†	11	Q(48-63)
10	†	00	Not Yet in Q
11		01	Q(16-31)

† Q-register refill is requested by end-op word; Q-register refill priority does not result if RX is indexed and request is issued two cycles early.

2. Incrementing of IC, transfer of Q to R, and the refilling of Q are inhibited if 'execute in progress' trigger is set.

3. 'RESET' micro-order:
   a. Inhibits setting of 'D sync' latch if LA, STC, STH, or unsuccessful BC.
   b. Sets 'IC sync' latch if unsuccessful BC.
   c. Resets STAT's, Edit controls, STC, and ABC.
   d. Sets LAR(0) if floating-point instruction.

4. I-Fetch sequencers 1 and 3 gate SDBO to Q during 2nd execution cycle, and transfer Q(0-15) to R if IC(21,22) = 10 during previous end op.

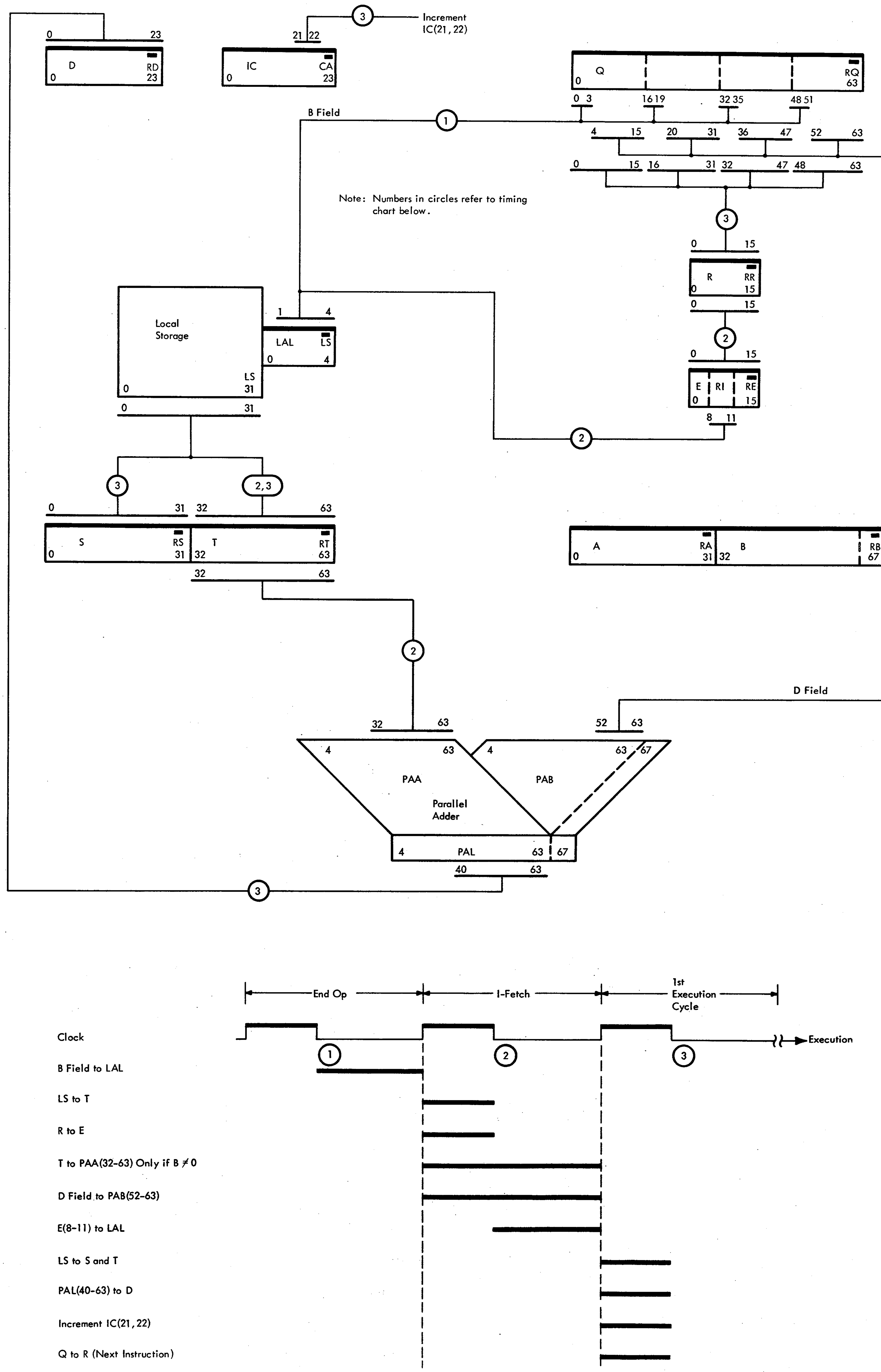


Diagram 5-10. One-Cycle RX, RS, and SI I-Fetch

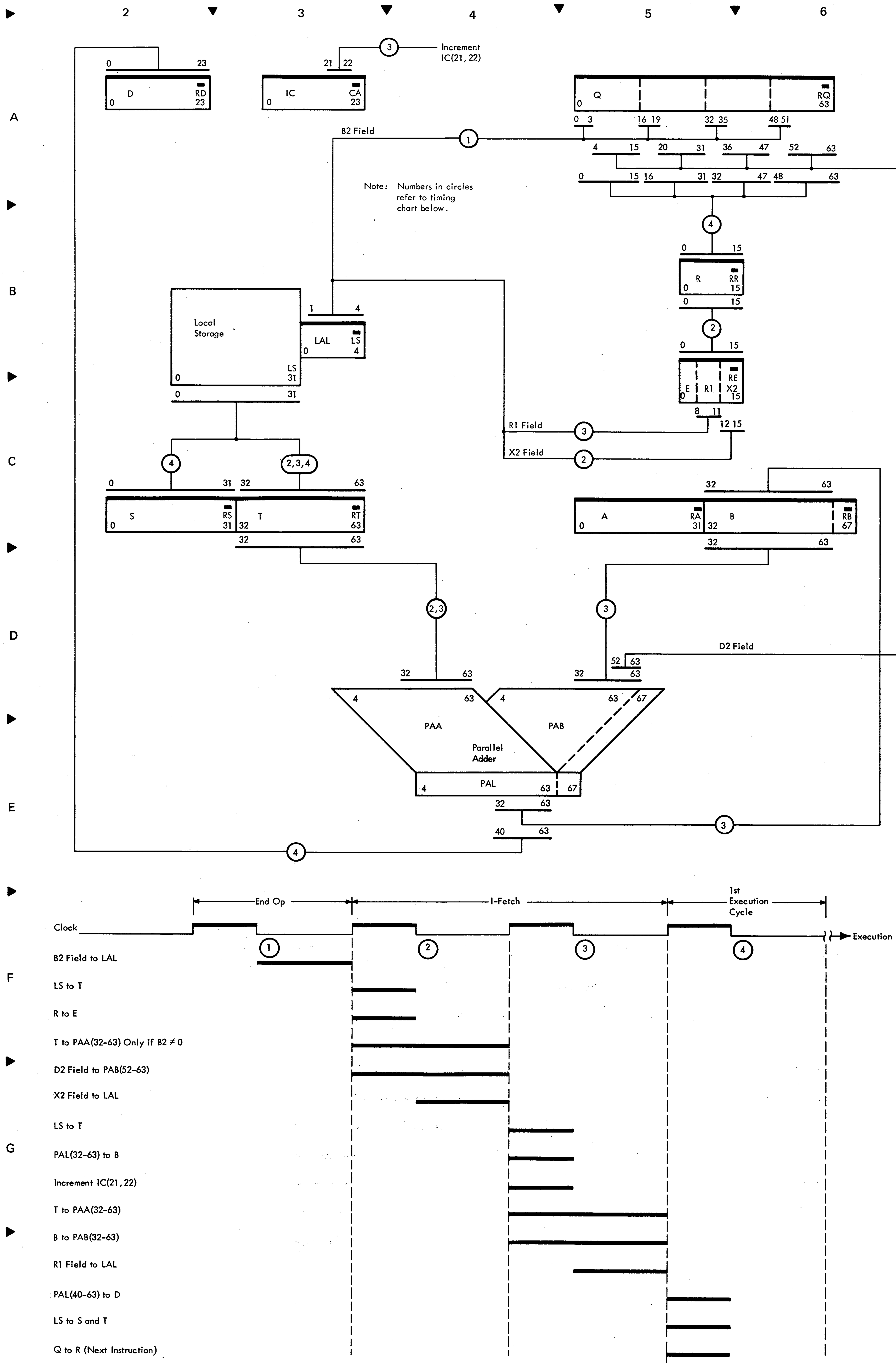


Diagram 5-11. Two-Cycle Indexed RX I-Fetch

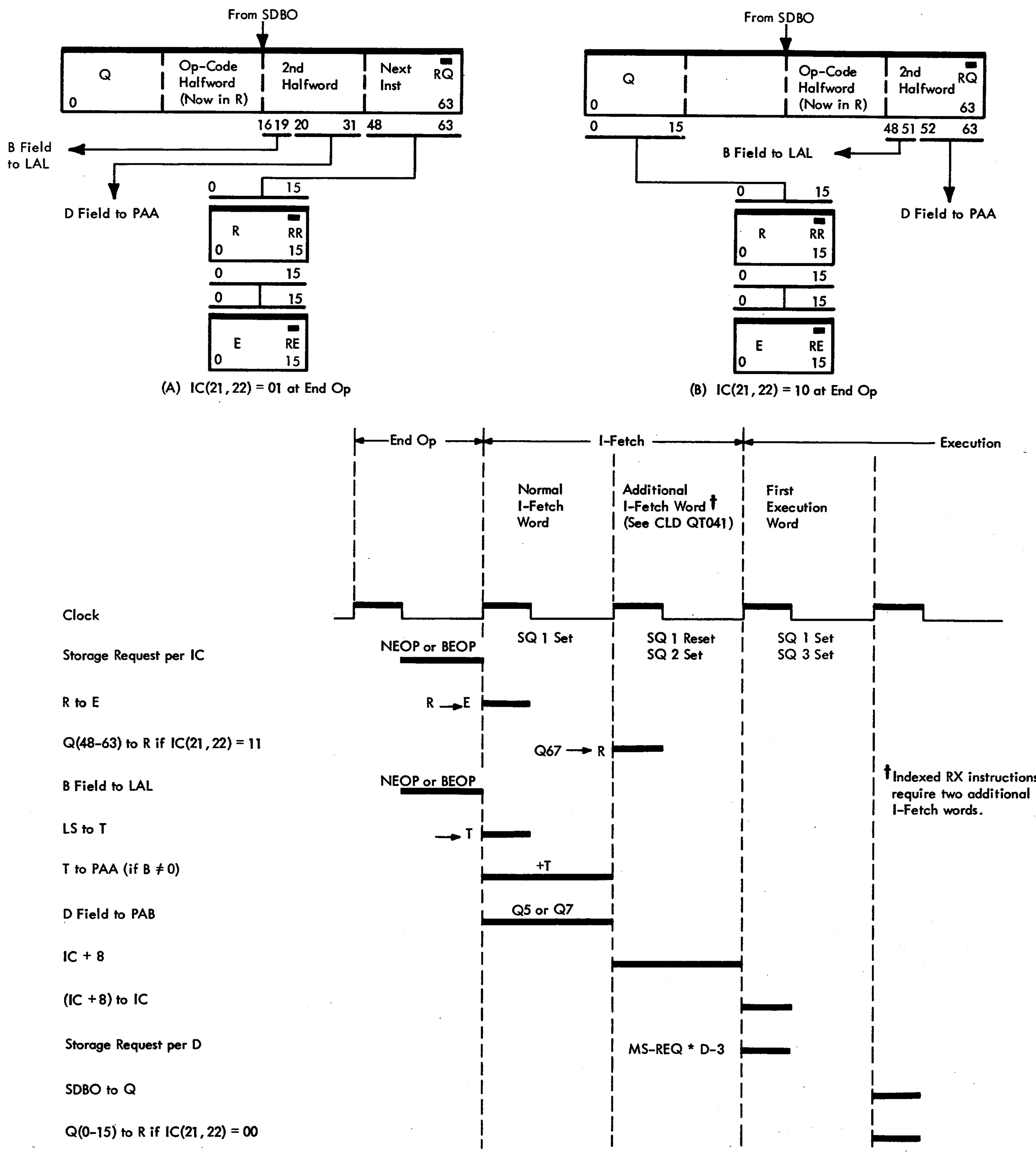


Diagram 5-12. Two-Cycle Non-Indexed RX, RS, and SI I-Fetch

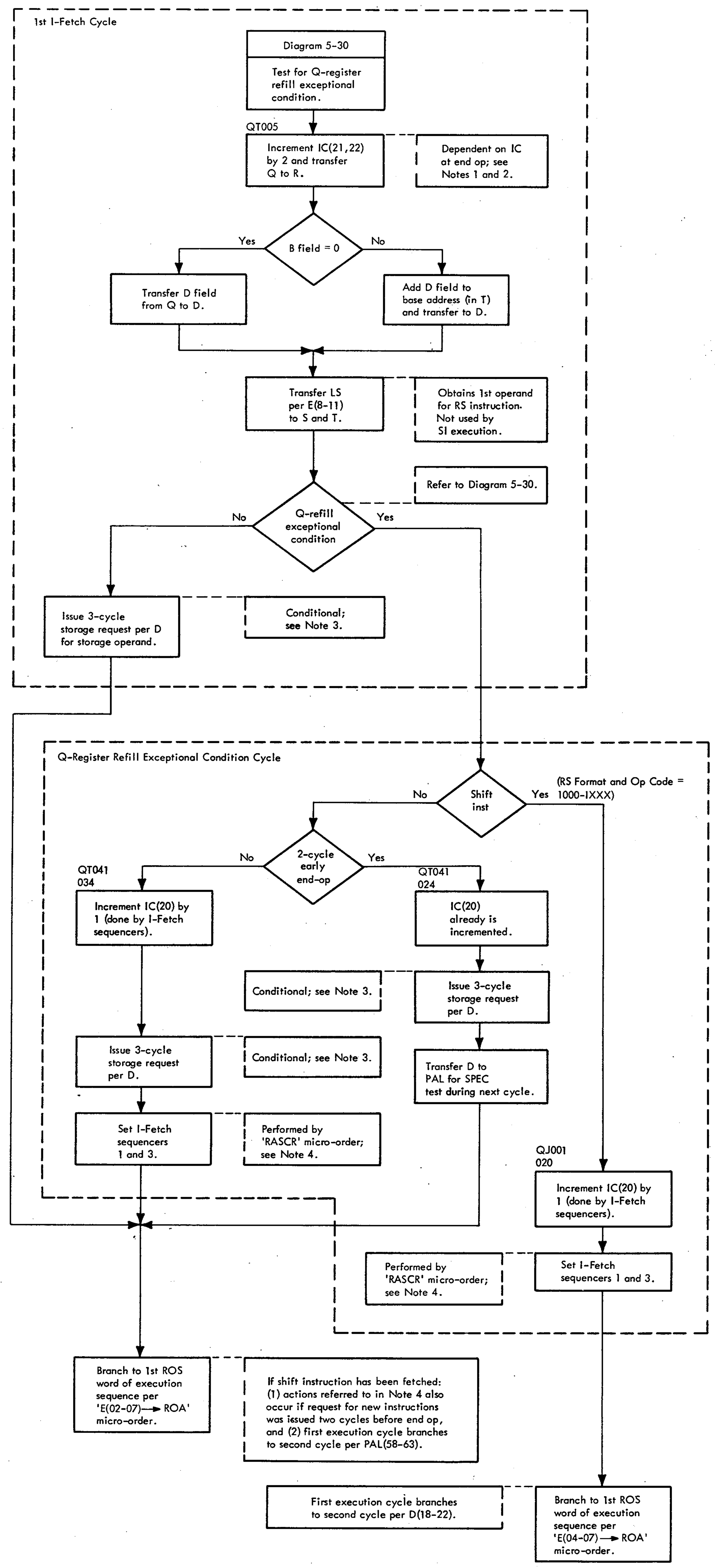


Diagram 5-13. RS and SI I-Fetch

- RS Format:

Op Code	R1	R3	B2	D2
0 7	8 11	12 15	16 19	20 31

- SI Format:

Op Code	I2	B1	D1
0 7	8 15	16 19	20 31

- Purpose:
  1. Transfer contents of LS register specified by E(8-11) to S and T. (This is 1st operand for RS instructions only.)
  2. Compute address of storage operand and transfer to D. (For RS instructions, this is address of 2nd operand.)
  3. Request operand from main storage (see Note 3).

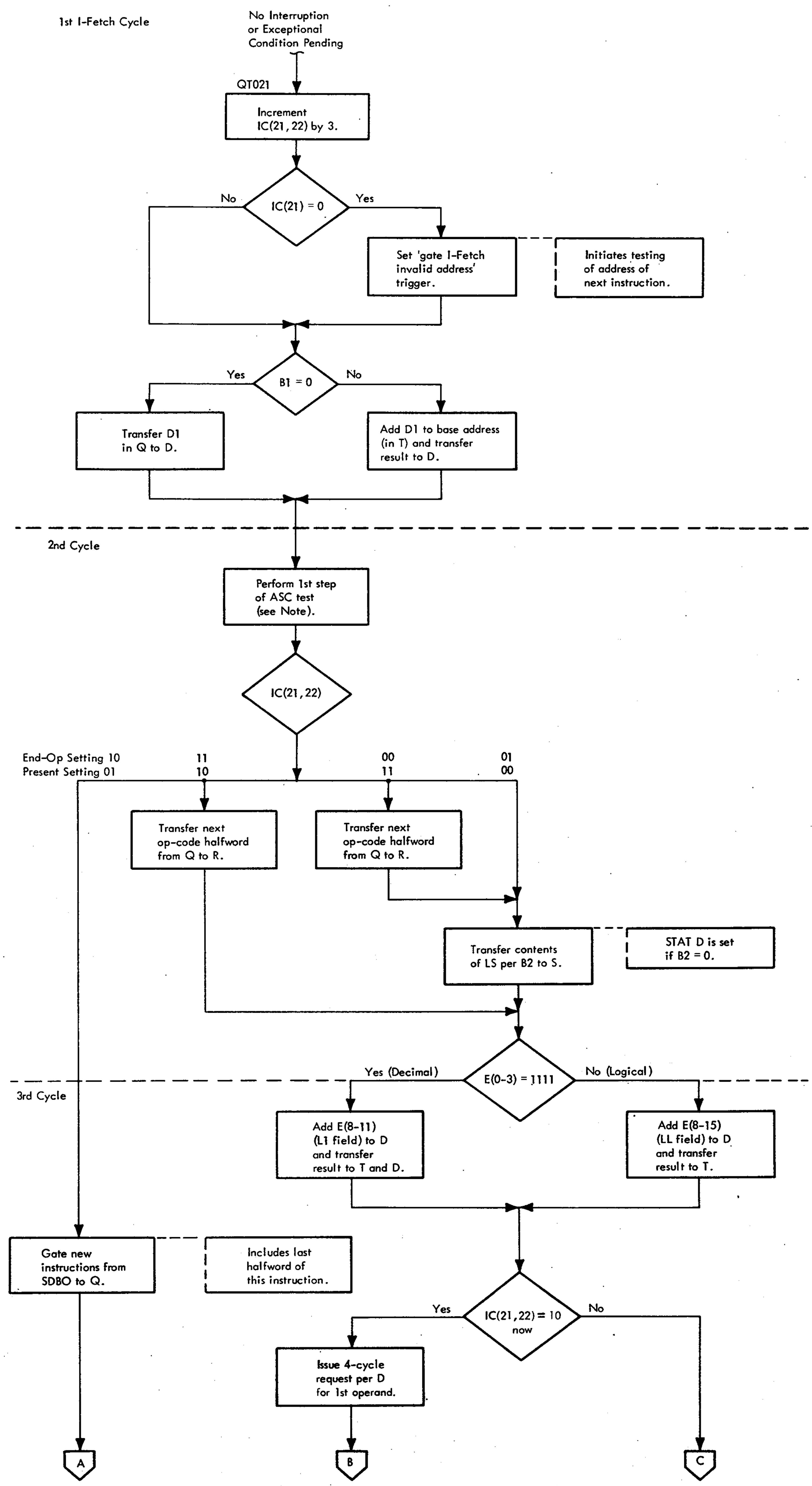
- Initial conditions if I-Fetch is not blocked:
  1. First halfword of instruction is transferred to E; 2nd halfword is in Q.
  2. Contents of LS register specified by B field are transferred to T.

Notes : 1. IC Incrementing and Q to R Transfer:

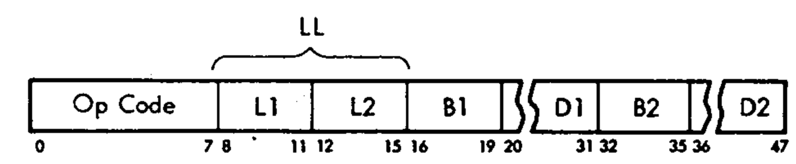
Conditions at End Op		Incremented IC(21,22)	Next Op-Code Word Transferred to R From:
IC(21,22)	Position in Q		
00		10	Q(32-47)
01	†	11	Q(48-63)
10	†	00	Not Yet in Q
11		01	Q(16-31)

†Q-register refill requested by end-op word; Q-register refill priority does not result if shift instruction is being fetched and request is issued two cycles early.

2. Incrementing of IC, transferring of Q to R, and setting of I-Fetch sequencers are inhibited if 'execute in progress' trigger is set.
3. 'RESET' micro-order:
   a. Inhibits setting of 'D sync' trigger if fetching MVI, STM, TS, shift, or I/O instruction.
   b. Resets '3-cycle request' trigger (causing 4-cycle request) if fetching BXH or BXLE instruction.
   c. Resets STAT's, Edit controls, STC, and ABC.
4. I-Fetch sequencers 1 and 3 gate SDBO to Q during 2nd execution cycle, and transfer Q(0-15) to R if IC(21,22) = 10 during previous end op.

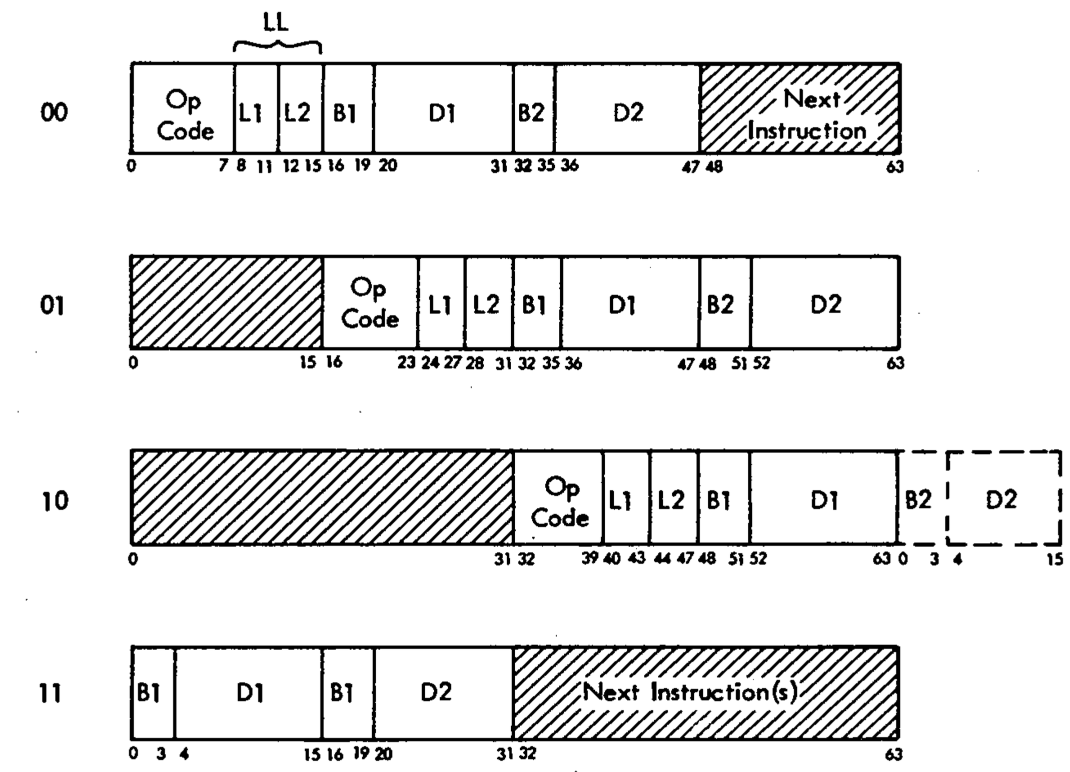


- SS Format:



- Purpose:
  1. Transfer op-code halfword of next instruction to R; update IC and place into LSWR.
  2. Transfer computed address of 1st operand (destination) per instruction class to D; request destination operand from main storage (gated into CPU during 2nd execution cycle):
     a. Lowest destination address for logical instructions = base address (per B1) + D1.
     b. Highest destination address for decimal instructions = base address (per B1) + D1 + L1.
  3. Transfer computed address of 2nd operand (source) to IC and T. Lowest source address = base address (per B2) + D2.
  4. Perform ASC test (and invalid instruction address test if complete instruction is in Q).

- Initial conditions if I-Fetch is not blocked:
  1. Contents of Q per setting of IC(21, 22) during preceding end of cycle:



  2. First halfword of instruction is transferring to E.
  3. Base address (per B1) is in T.
  4. Main storage request for more instructions was generated per IC during preceding end op if 'execute in progress' trigger is not set and IC(21) = 0 (4-cycle) or if IC(21, 22) = 10 (3-cycle). (Branch end op tests D instead of IC.)

Note: Details of 1st step of ASC test:

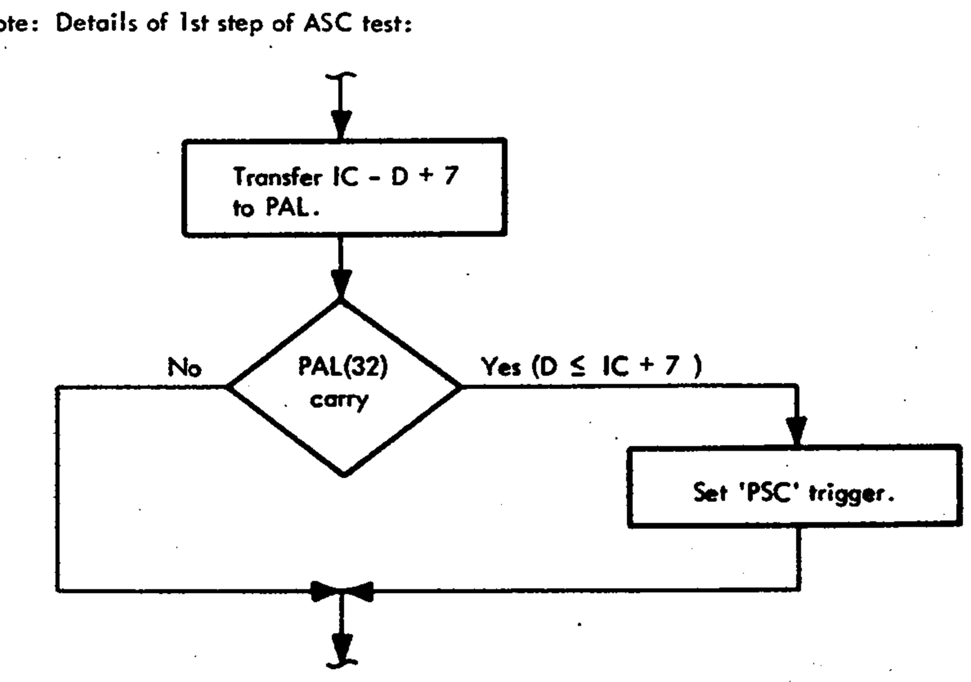


Diagram 5-14. SS I-Fetch (Sheet 1 of 2)

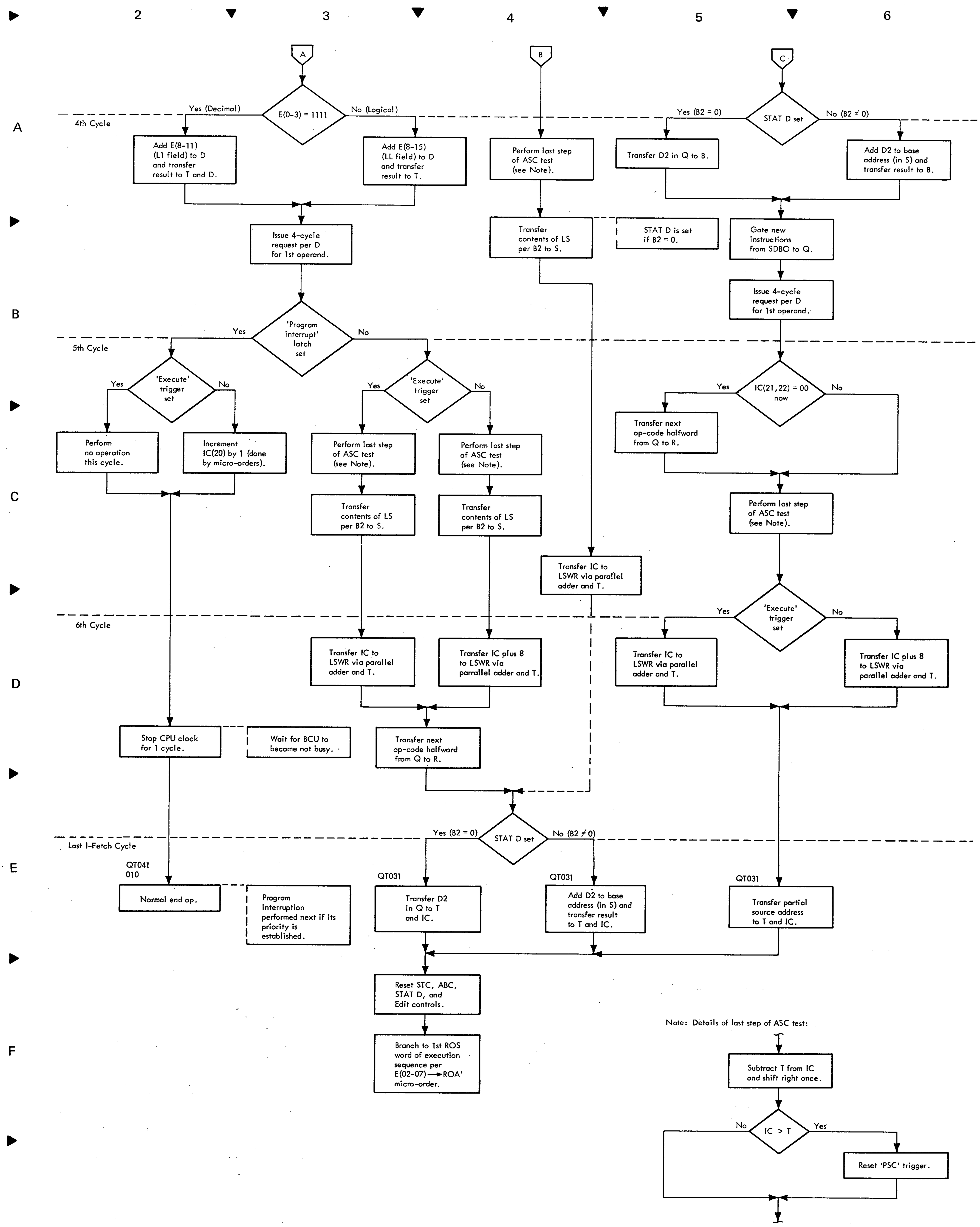


Diagram 5-14. SS I-Fetch (Sheet 2 of 2)

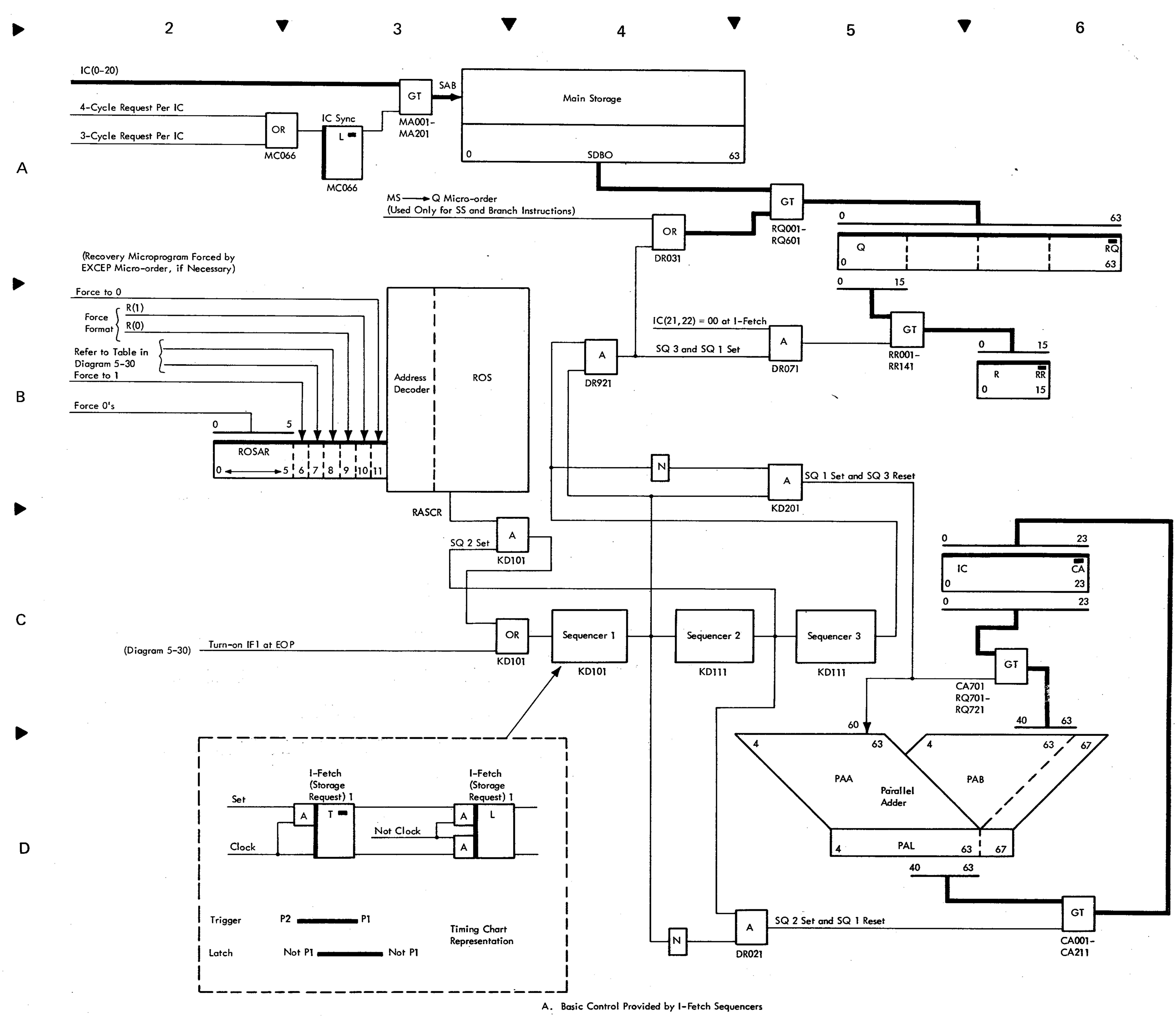
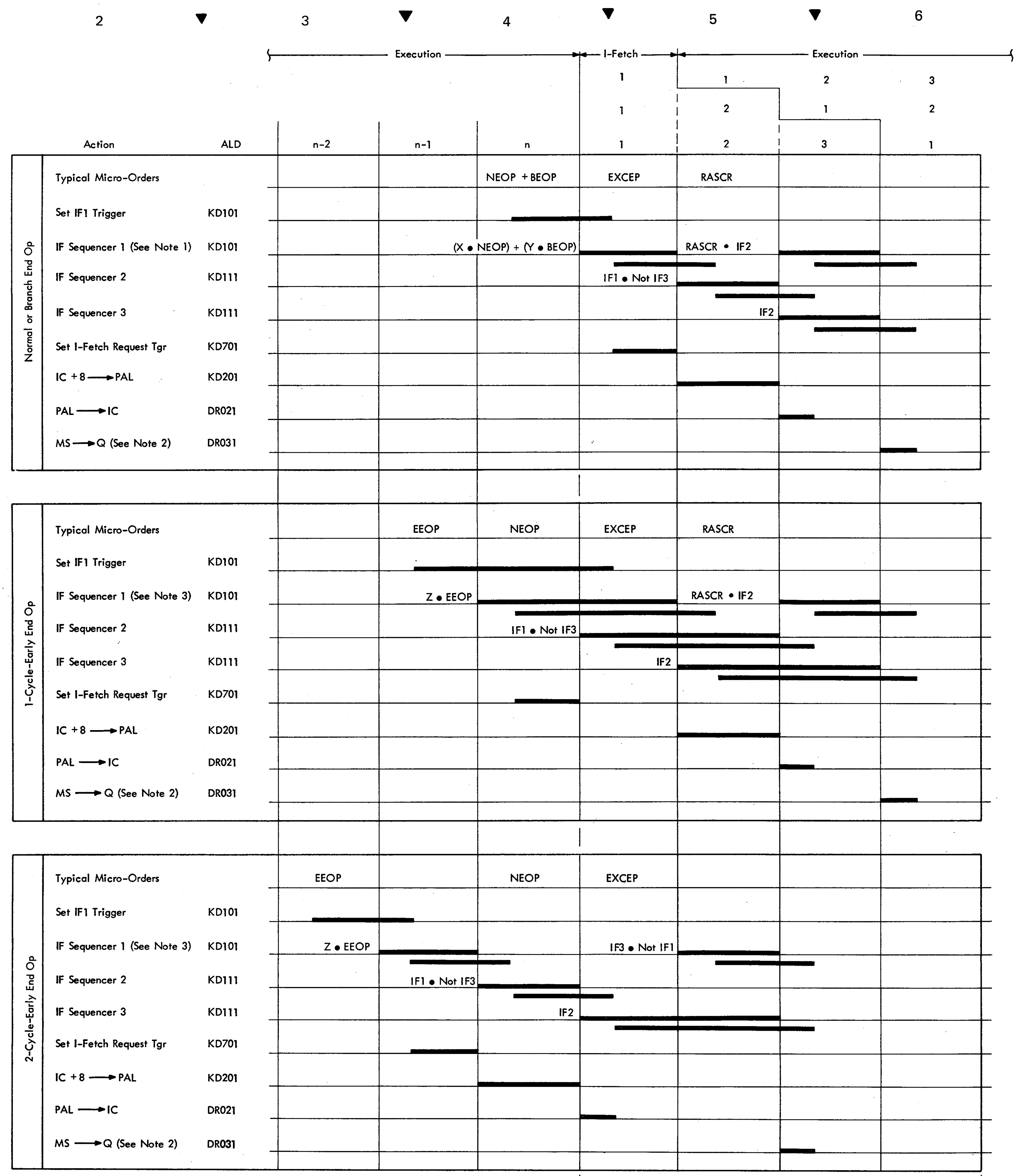


Diagram 5-15. I-Fetch Sequences (Sheet 1 of 2)



These actions are inhibited by 'block I-Fetch' trigger.

Notes: 1. X = Z • Not IF2, where Z is defined in Note 3.
Y = [Not (Predecode branch + Predecode SS + Execute)] • [D21,22 = 10 + (D21,22 = 01 • Predecode not RR)].

2. Also, gate Q(0-15) to R(0-15) if IC(21,22) = 00 (ALD DR071).

3. Z = [Not (Predecode branch + Predecode SS + Execute)] • [IC21,22 = 10 + (IC 21,22 = 01 • Predecode not RR).

B. I-Fetch Sequencer Timing Chart

Diagram 5-15. I-Fetch Sequencers (Sheet 2 of 2)

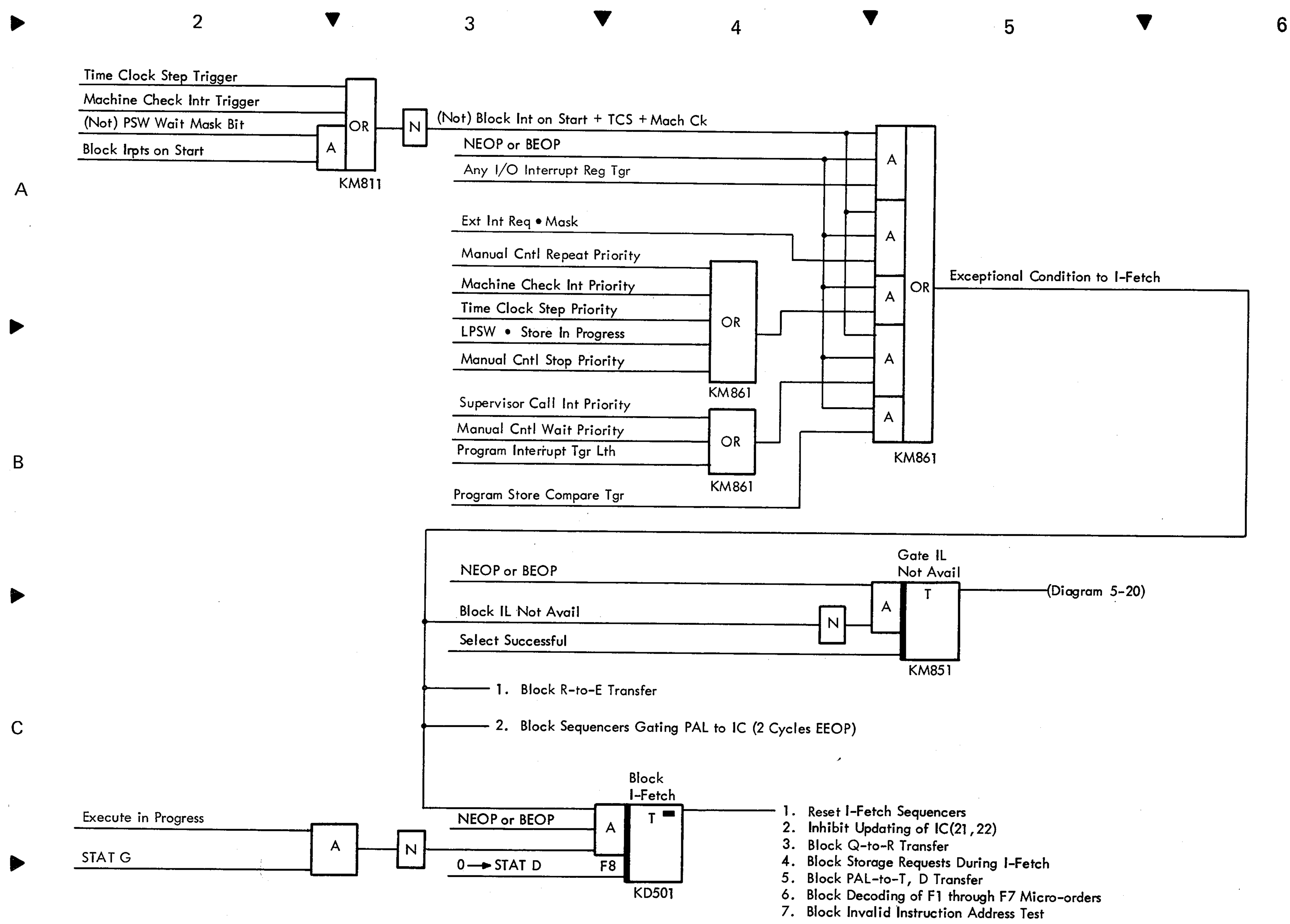


Diagram 5-16. Block I-Fetch Trigger

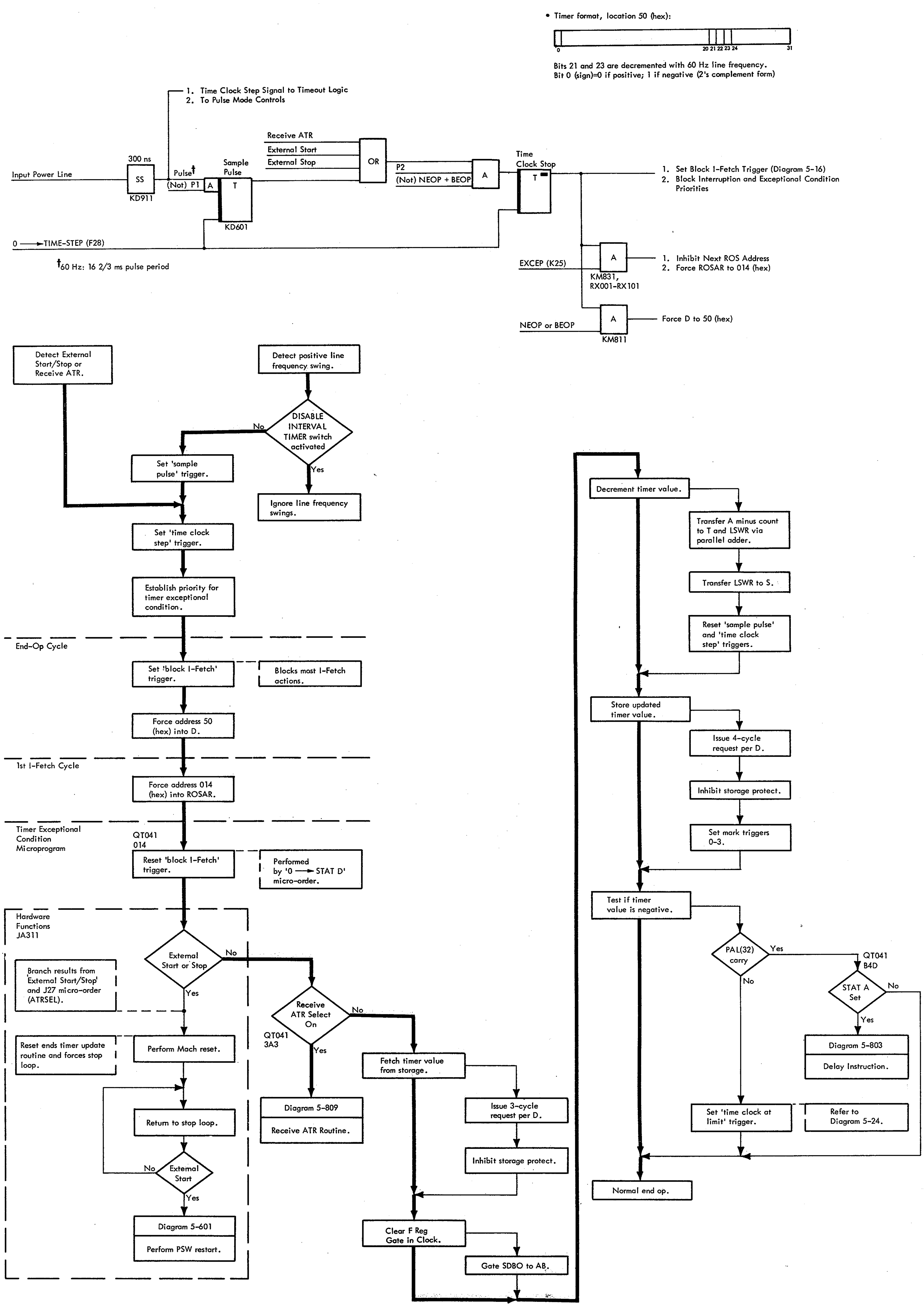


Diagram 5-17. Timer Exceptional Condition

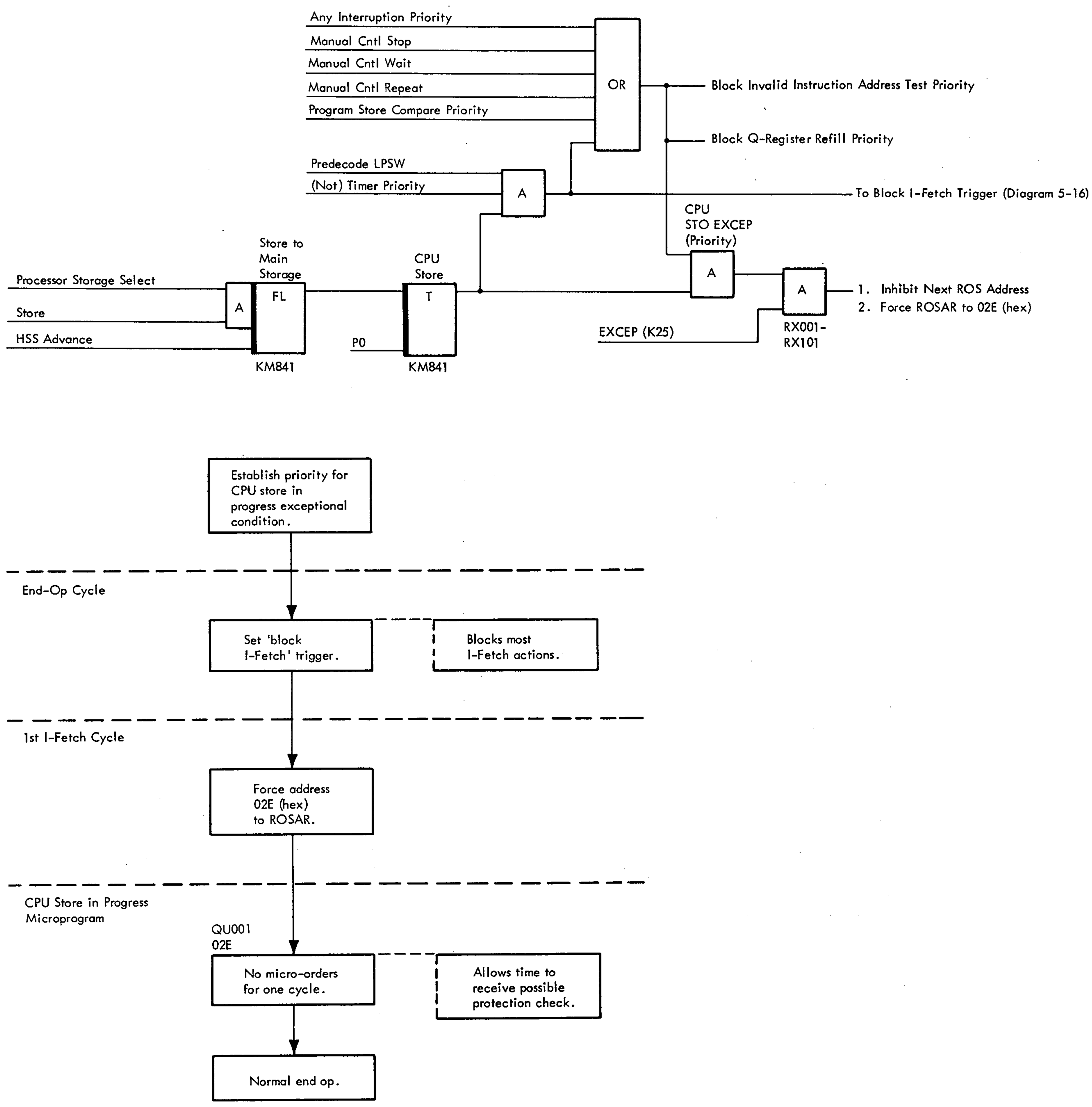


Diagram 5-18. CPU Store In Progress Exceptional Condition

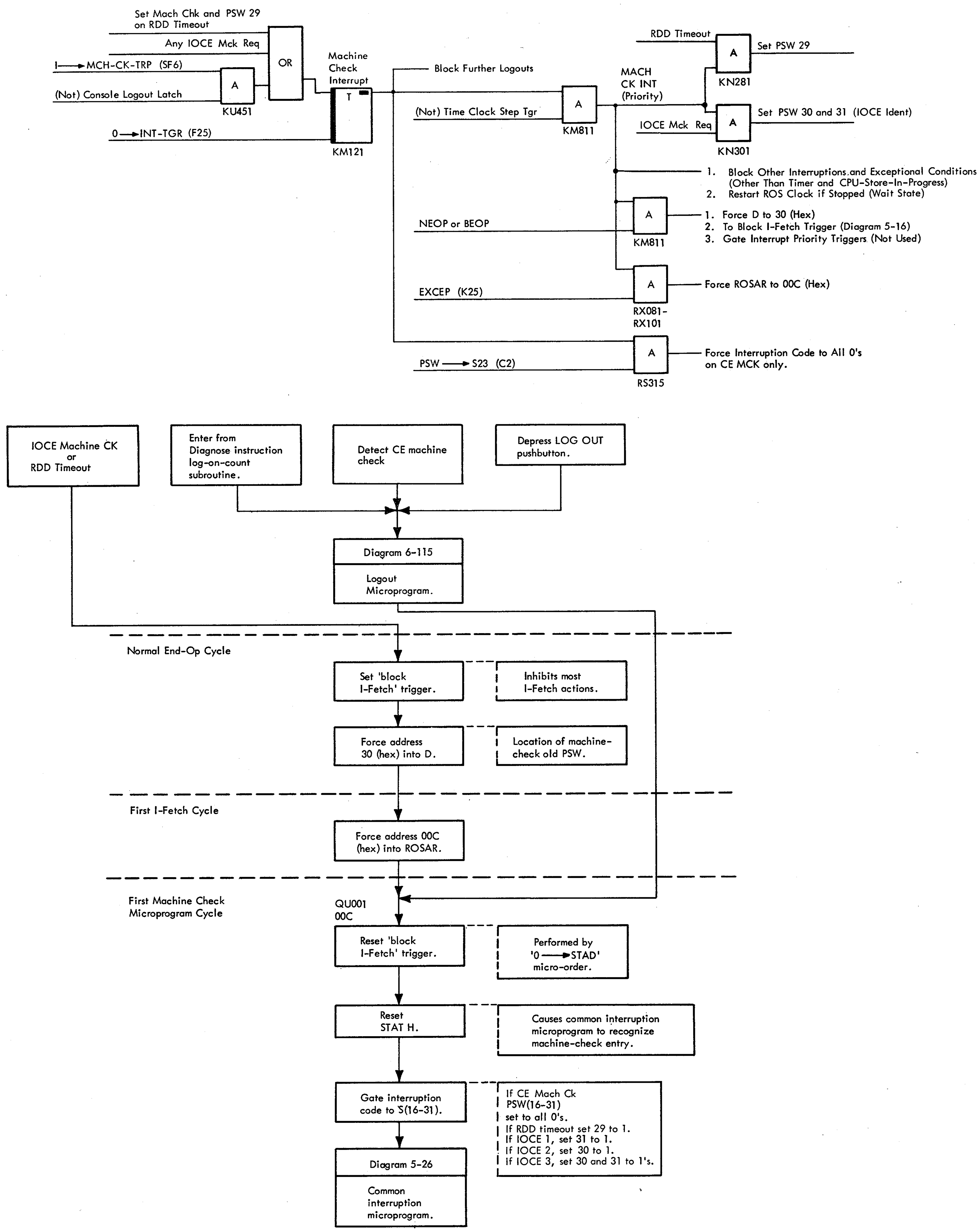
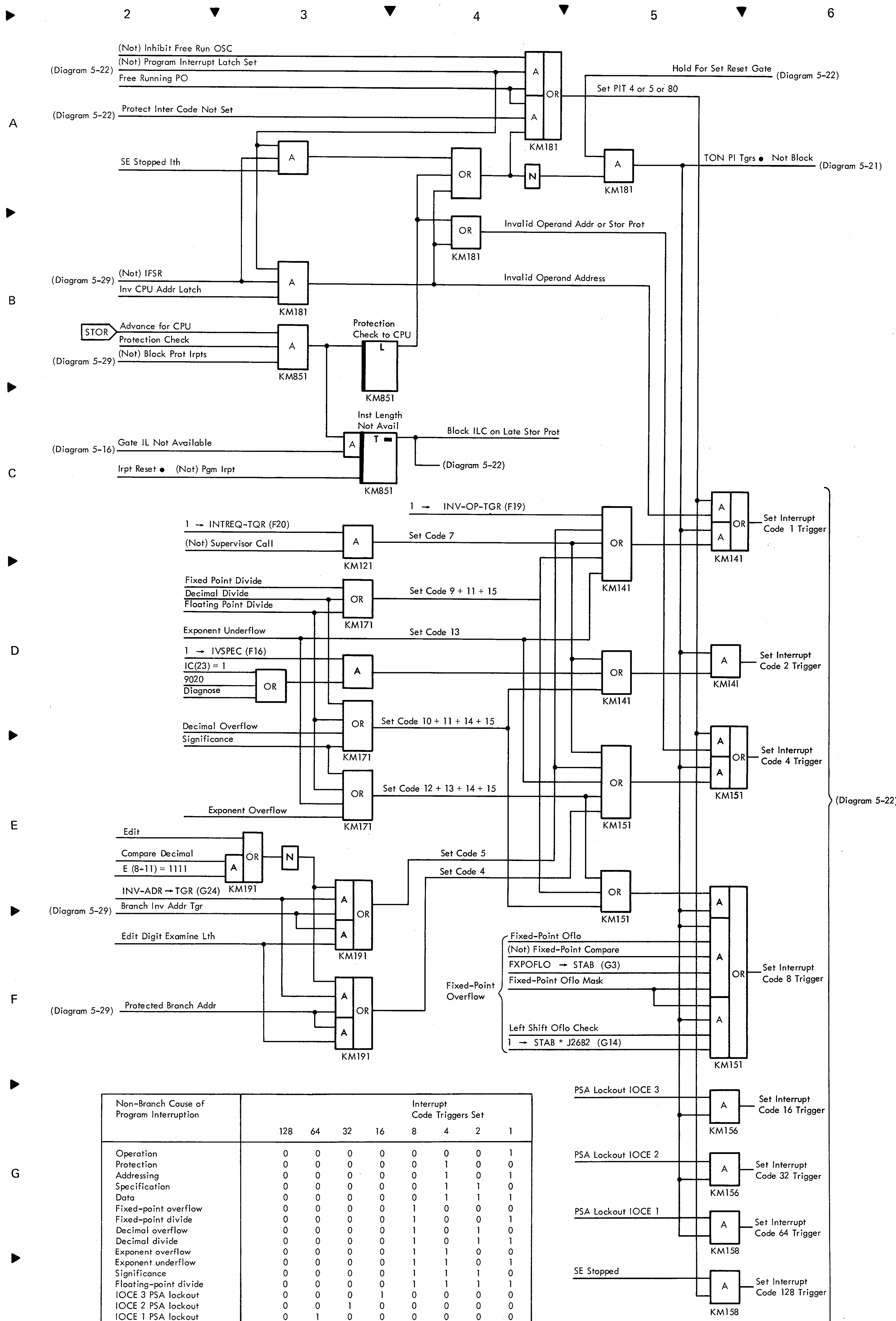
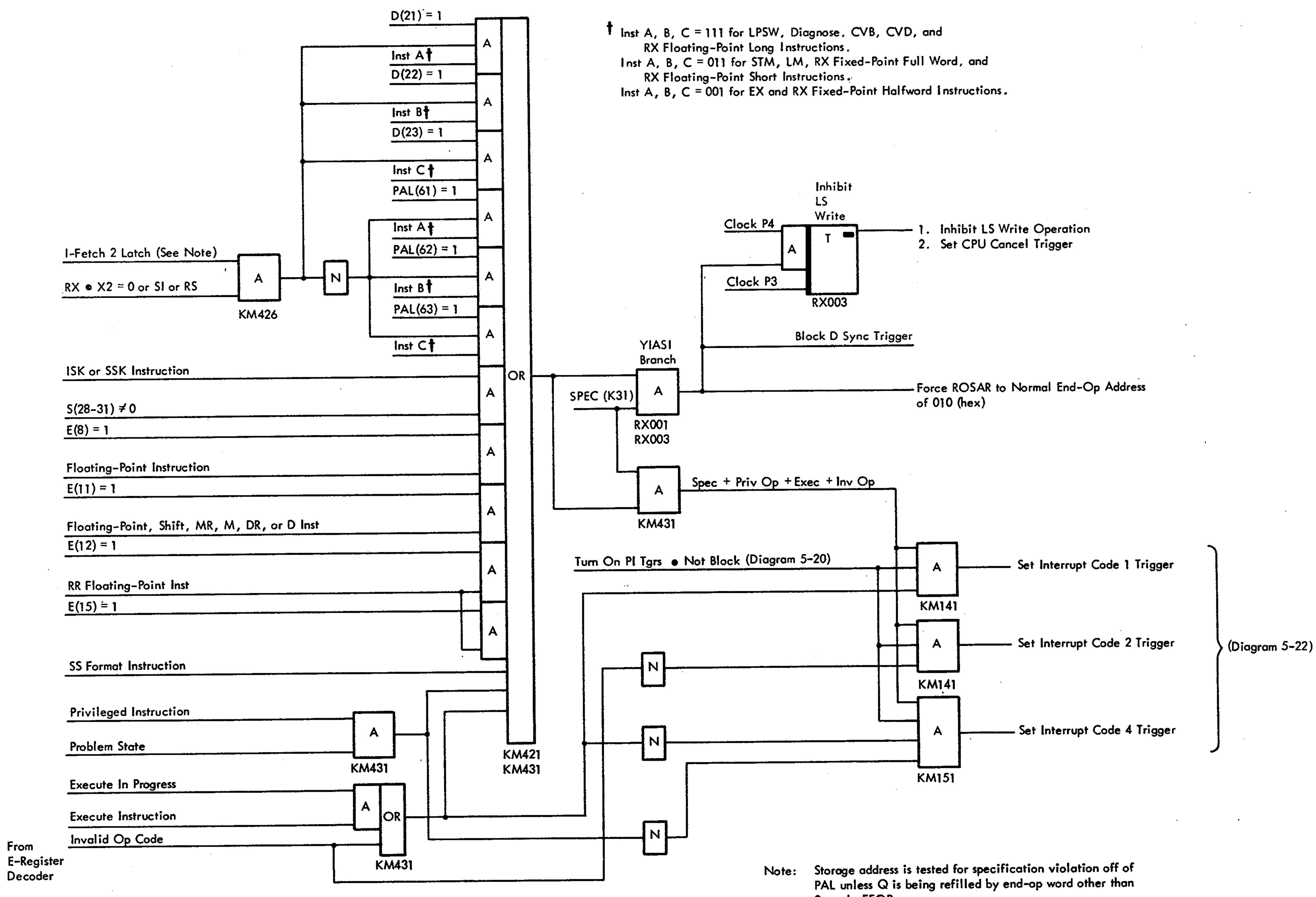


Diagram 5-19. Machine Check Interruption



Non-Branch Cause of Program Interruption	Interrupt Code Triggers Set 128	64	32	16	8	4	2	1
Operation	0	0	0	0	0	0	0	1
Protection	0	0	0	0	0	1	0	0
Addressing	0	0	0	0	0	1	0	1
Specification	0	0	0	0	0	1	1	0
Data	0	0	0	0	0	1	1	1
Fixed-point overflow	0	0	0	0	1	0	0	0
Fixed-point divide	0	0	0	0	1	0	0	1
Decimal overflow	0	0	0	0	1	0	1	0
Decimal divide	0	0	0	0	1	0	1	1
Exponent overflow	0	0	0	0	1	1	0	0
Exponent underflow	0	0	0	0	1	1	0	1
Significance	0	0	0	0	1	1	1	0
Floating-point divide	0	0	0	0	1	1	1	1
IOCE 3 PSA lockout	0	0	0	1	0	0	0	0
IOCE 2 PSA lockout	0	0	1	0	0	0	0	0
IOCE 1 PSA lockout	0	1	0	0	0	0	0	0
Logout stop	1	0	0	0	0	0	0	0

Diagram 5-20. Non-Branch Setting of Interrupt Code Triggers

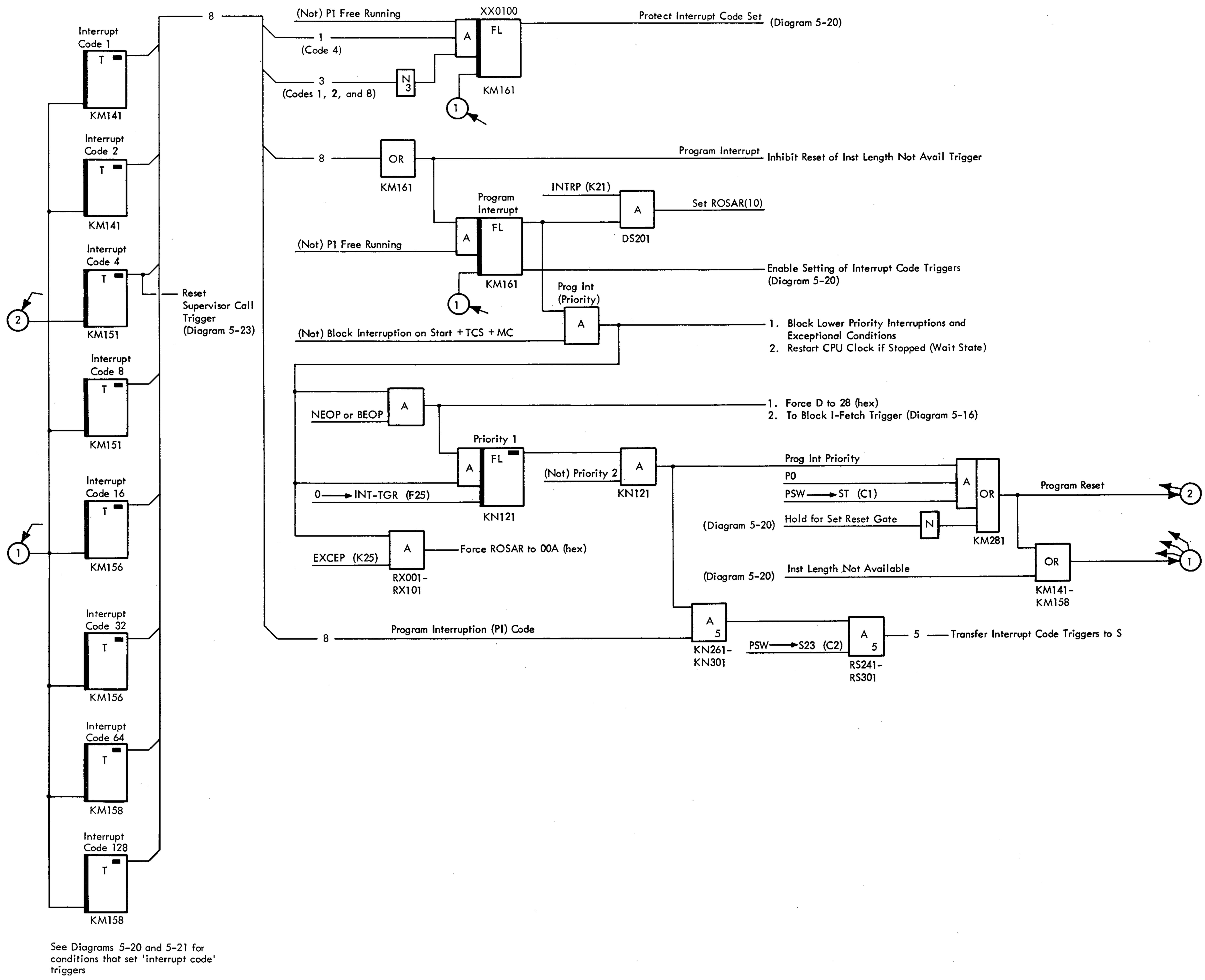


† Inst A, B, C = 111 for LPSW, Diagnose, CVB, CVD, and RX Floating-Point Long Instructions.
Inst A, B, C = 011 for STM, LM, RX Fixed-Point Full Word, and RX Floating-Point Short Instructions.
Inst A, B, C = 001 for EX and RX Fixed-Point Halfword Instructions.

Note: Storage address is tested for specification violation off of PAL unless Q is being refilled by end-op word other than 2-cycle EEOP.

SPEC Y-Branch Cause	Interrupt Code Triggers Set 16	8	4	2	1
Operation	0	0	0	0	1
Privileged Operation	0	0	0	1	0
Execute	0	0	0	1	1
Specification	0	0	1	1	0

Diagram 5-21. SPEC Y-Branch Setting of Interrupt Code Triggers



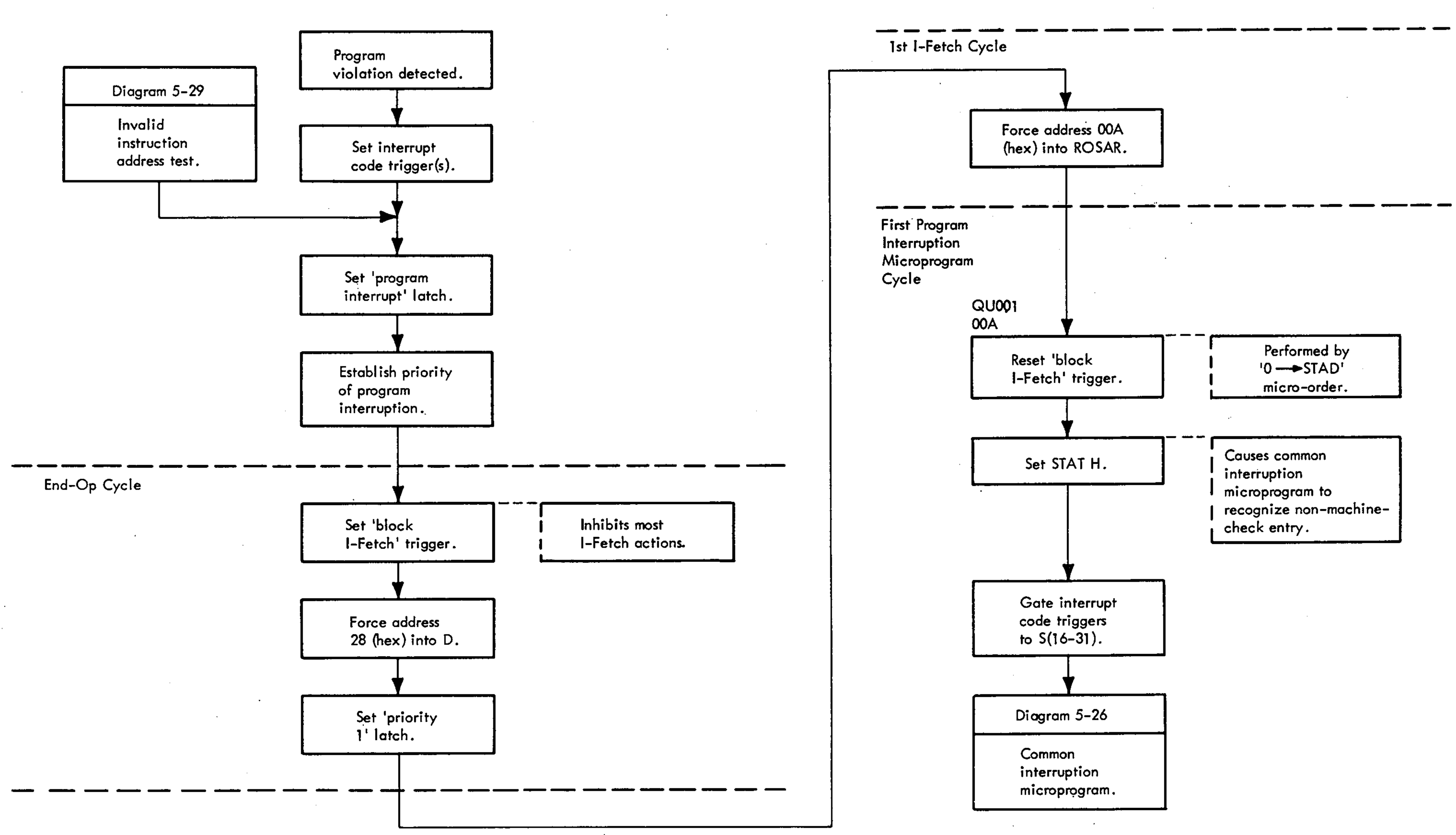
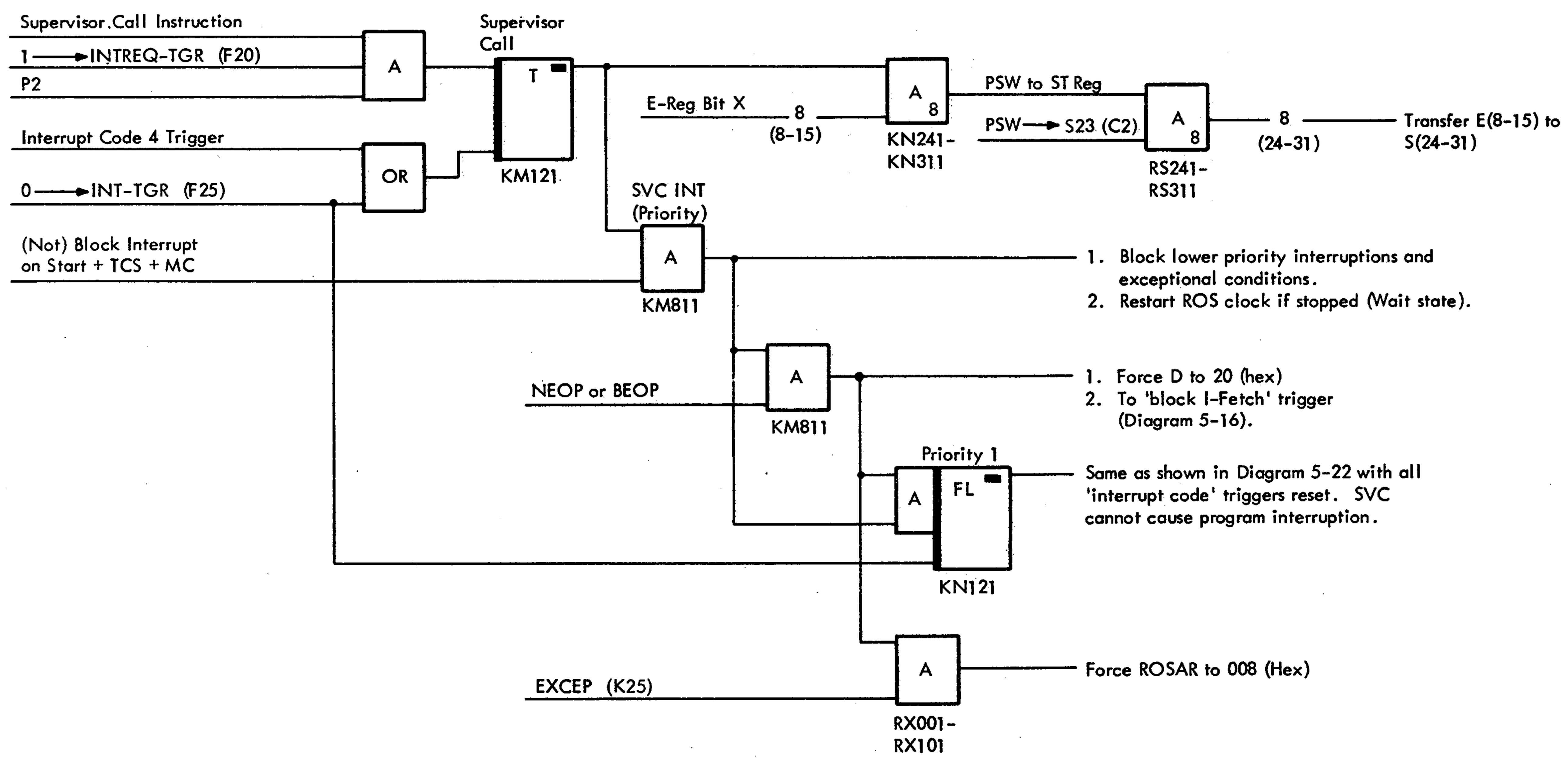


Diagram 5-22. Program Interruption



- RR Format:

0A	I
0–7	8–15

- Conditions at start of execution:
  1. Instruction is in E.
  2. 1st operand (not used) is in A, B, and D.
  3. 2nd operand (not used) is in S and T.

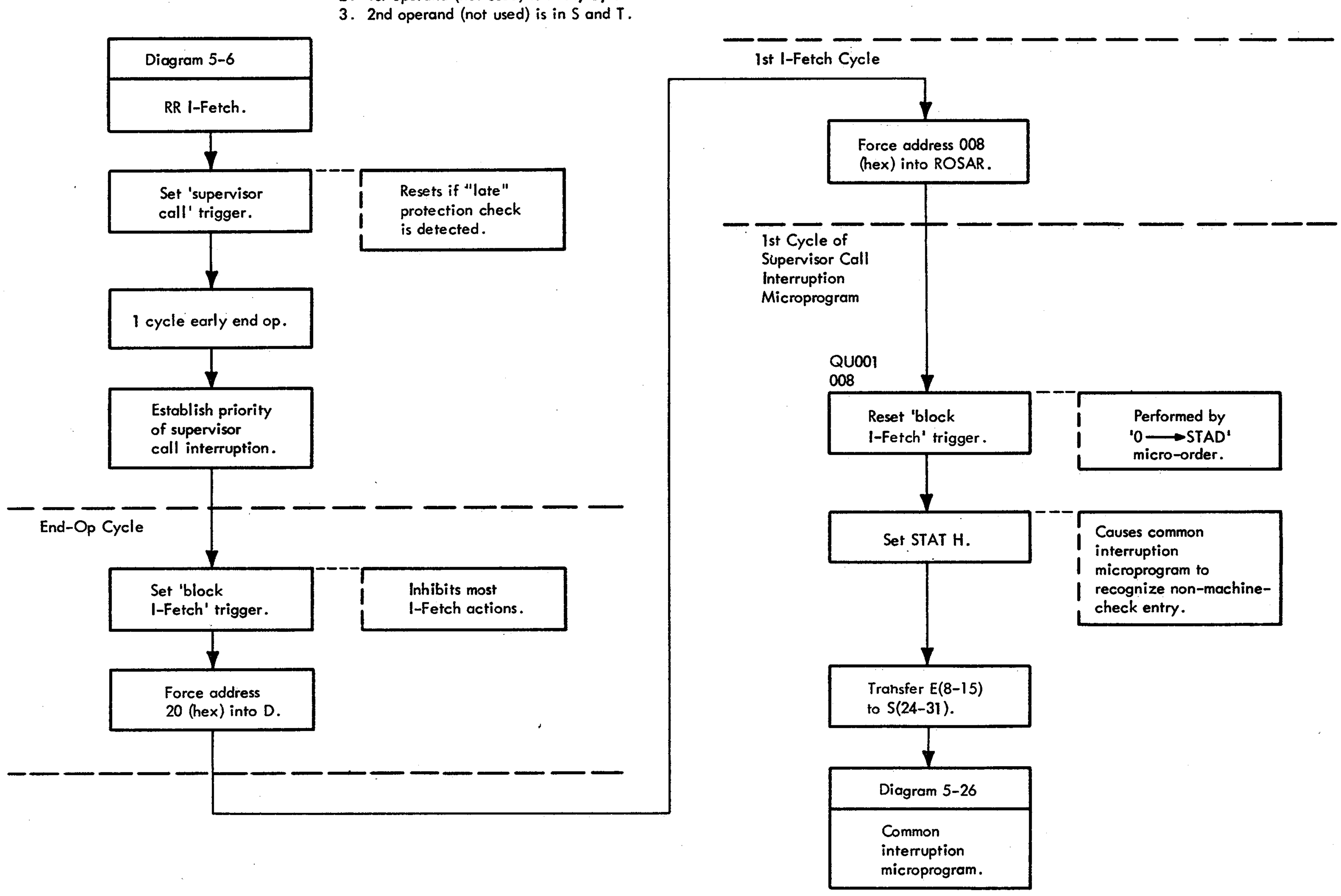


Diagram 5-23. Supervisor Call Interruption

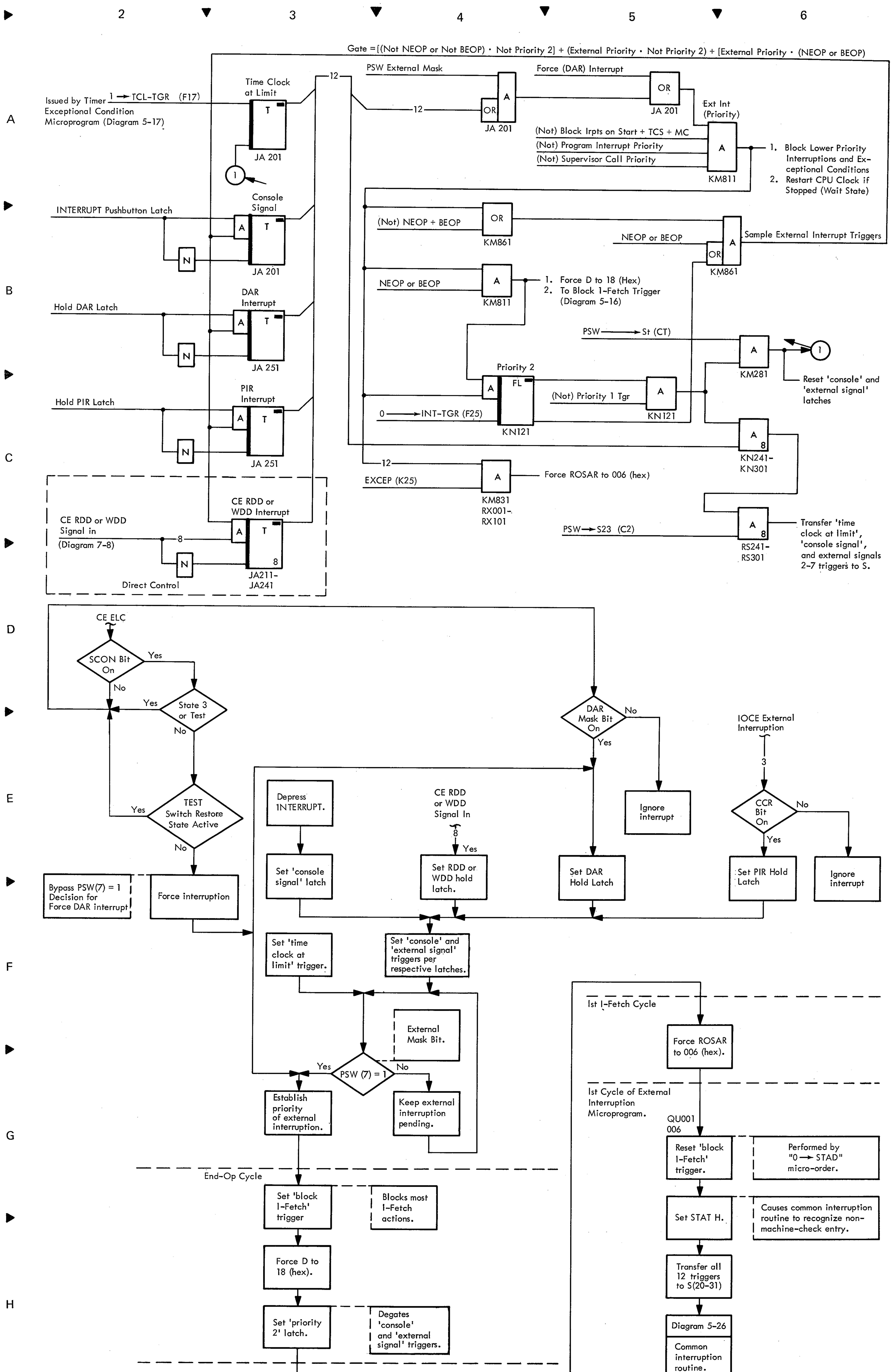


Diagram 5-24. External Interruption

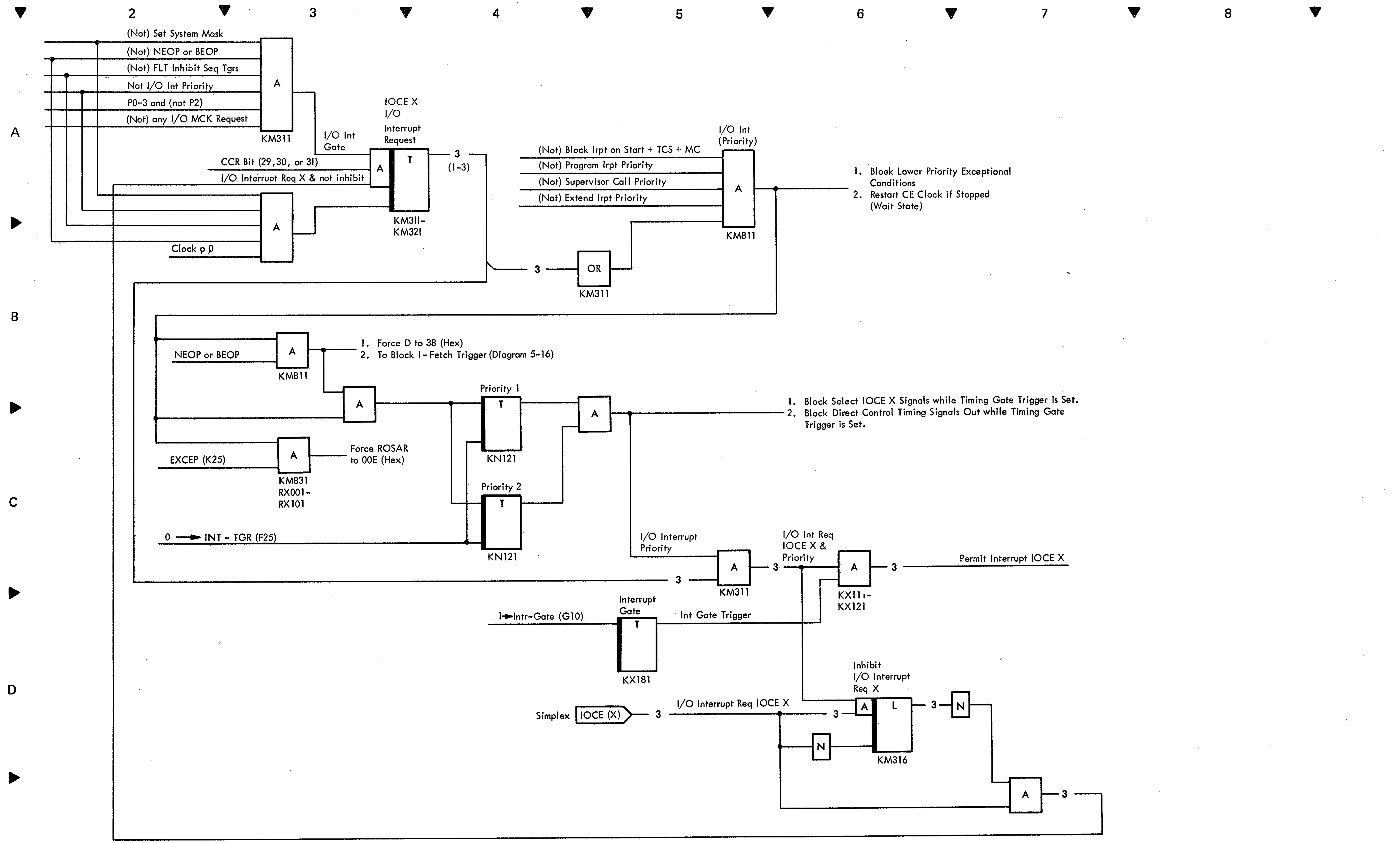


Diagram 5-25. I/O Interruption (Sheet 1 of 2)

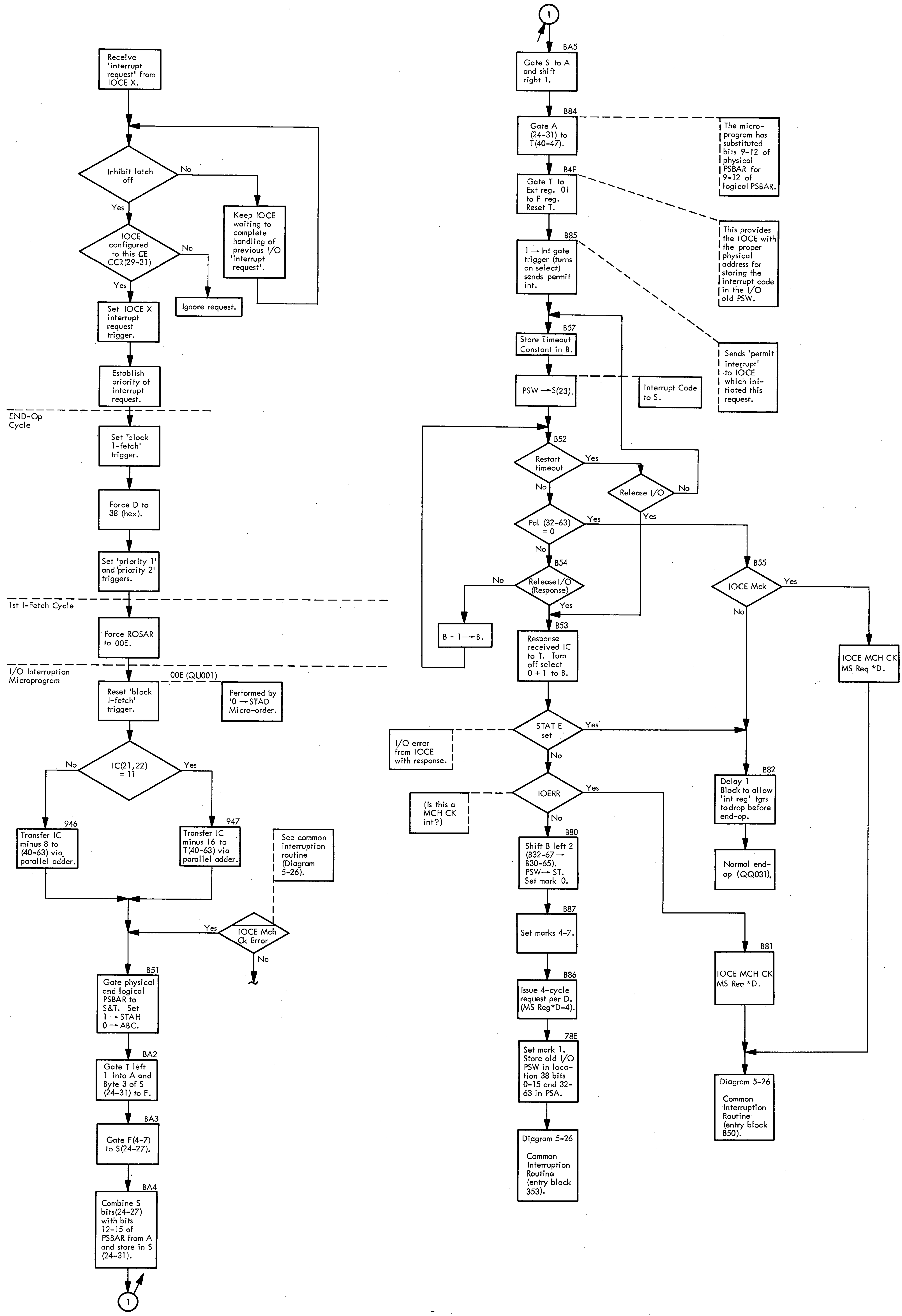


Diagram 5-25. I/O Interruption (Sheet 2 of 2)

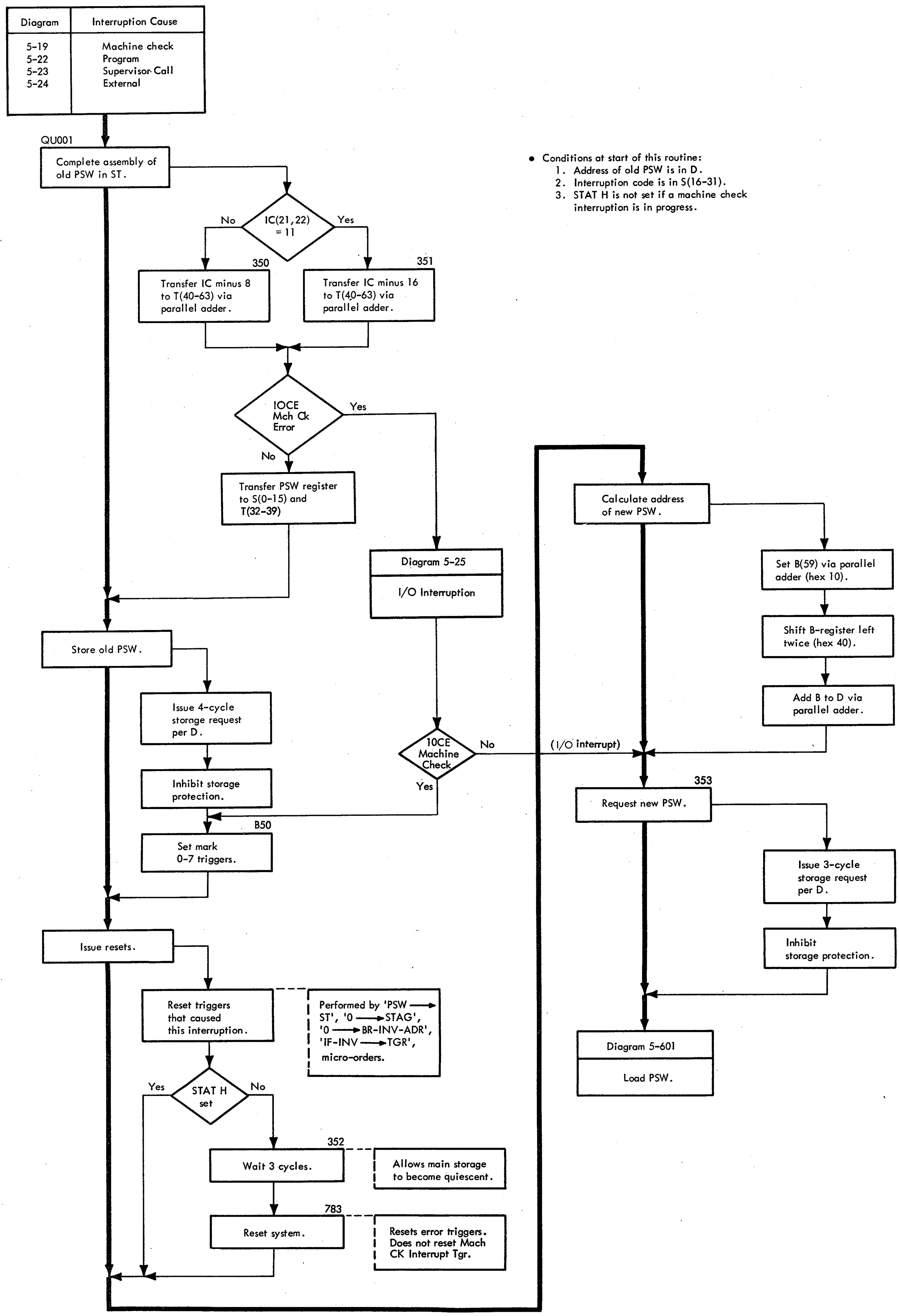


Diagram 5-26. Common Interruption Routine

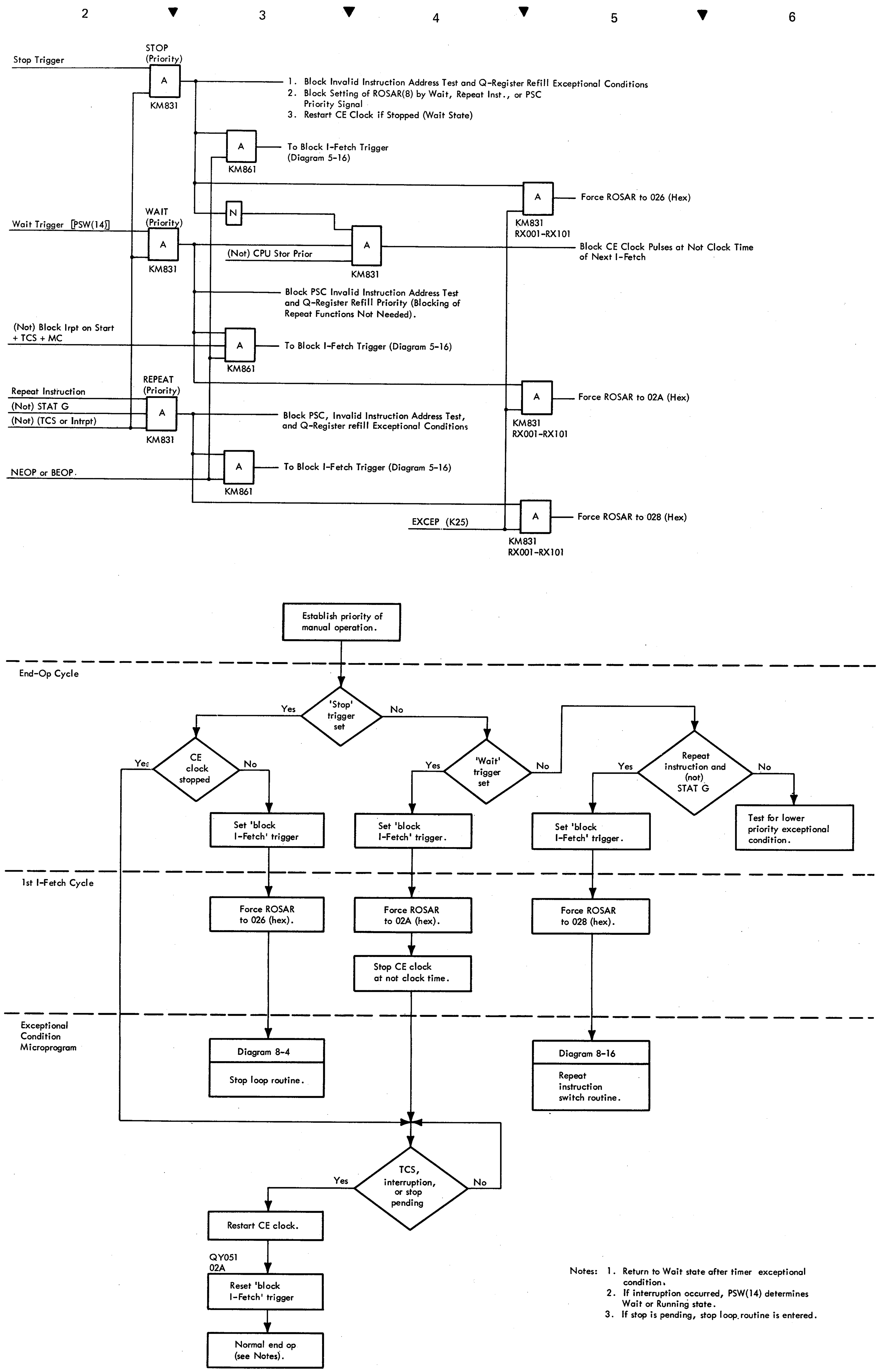


Diagram 5-27. Manual Control Exceptional Conditions

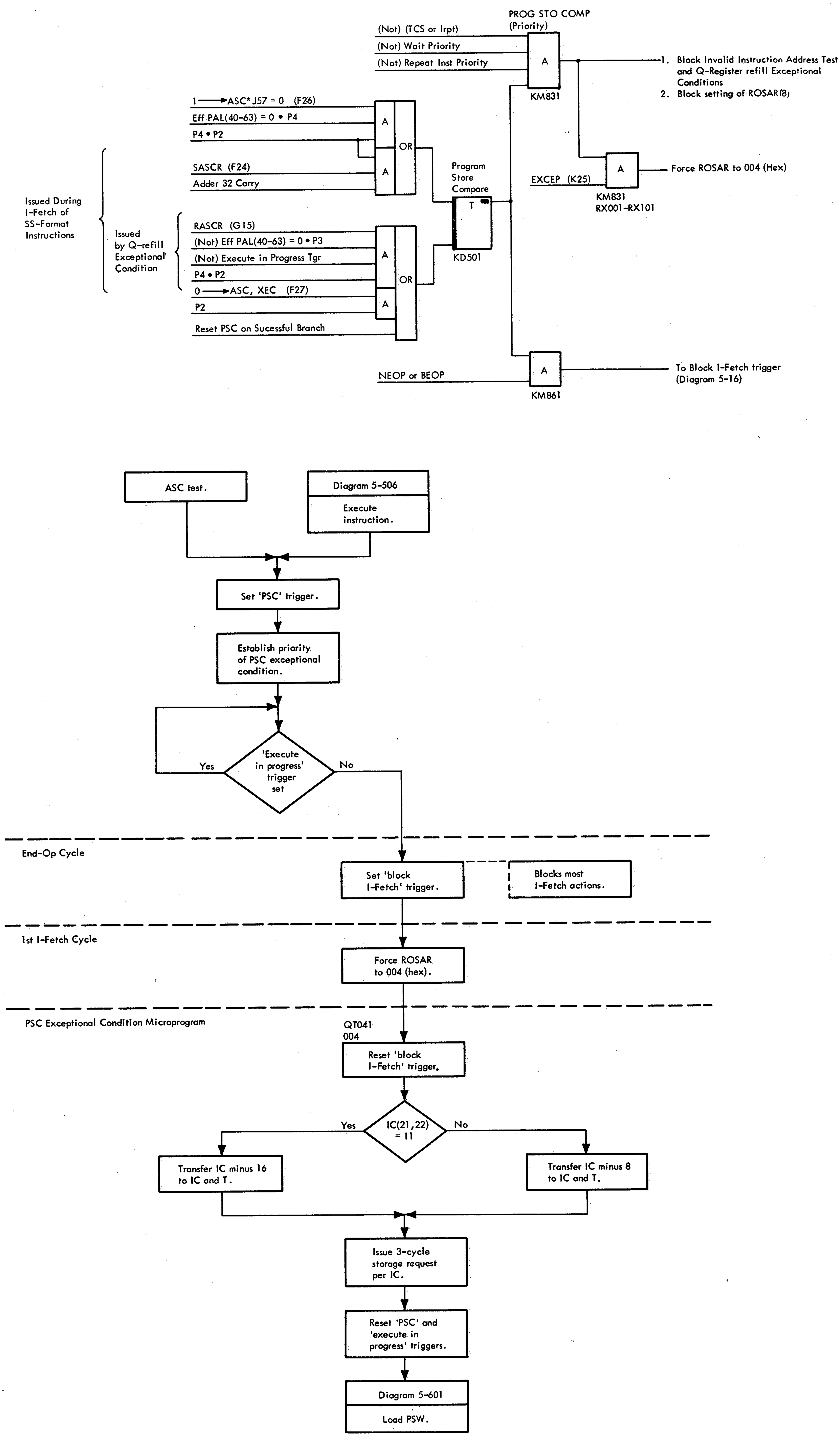


Diagram 5-28. Program Store Compare Exceptional Condition

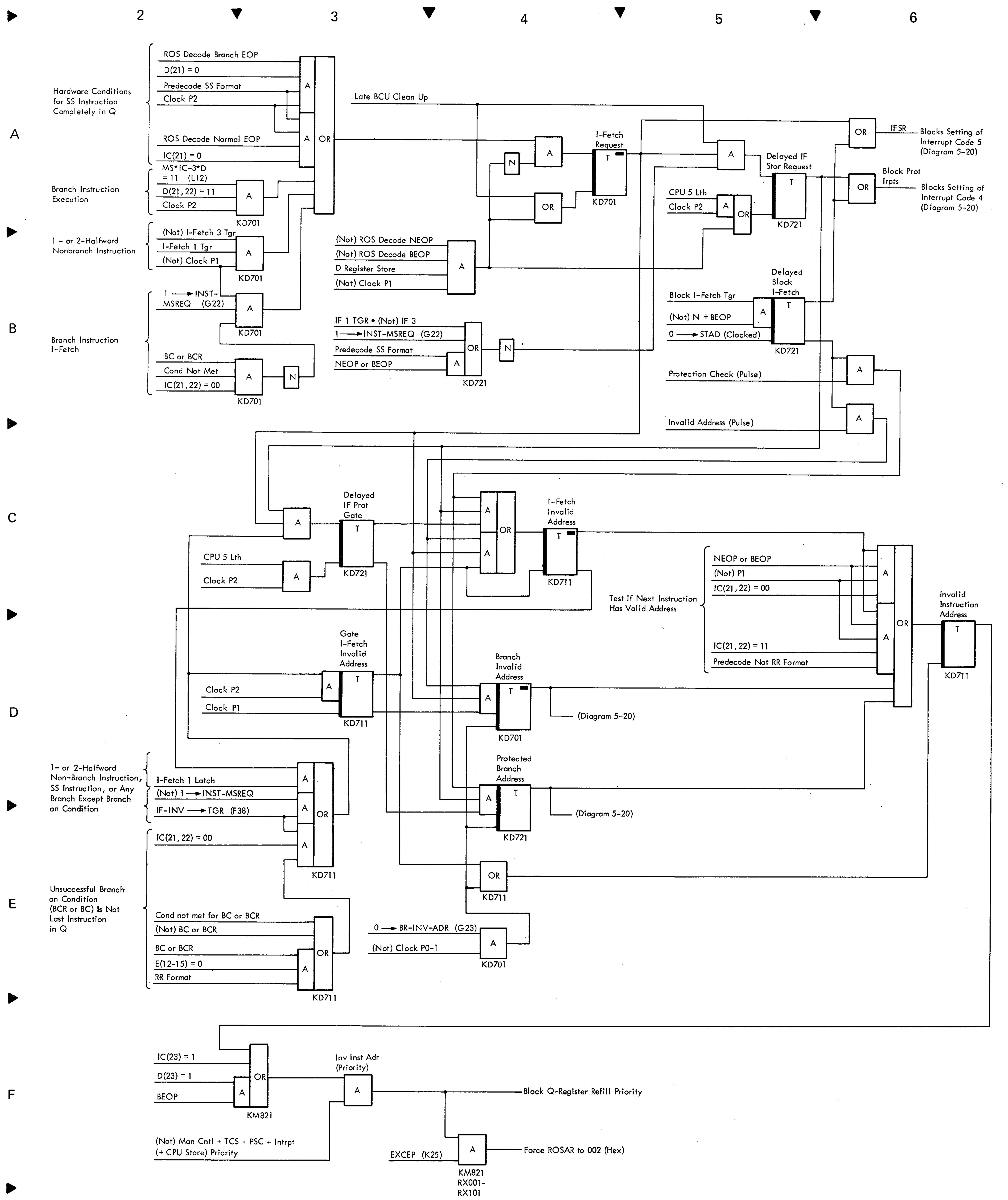


Diagram 5-29. Invalid Instruction Address Test Exceptional Condition (Sheet 1 of 2)

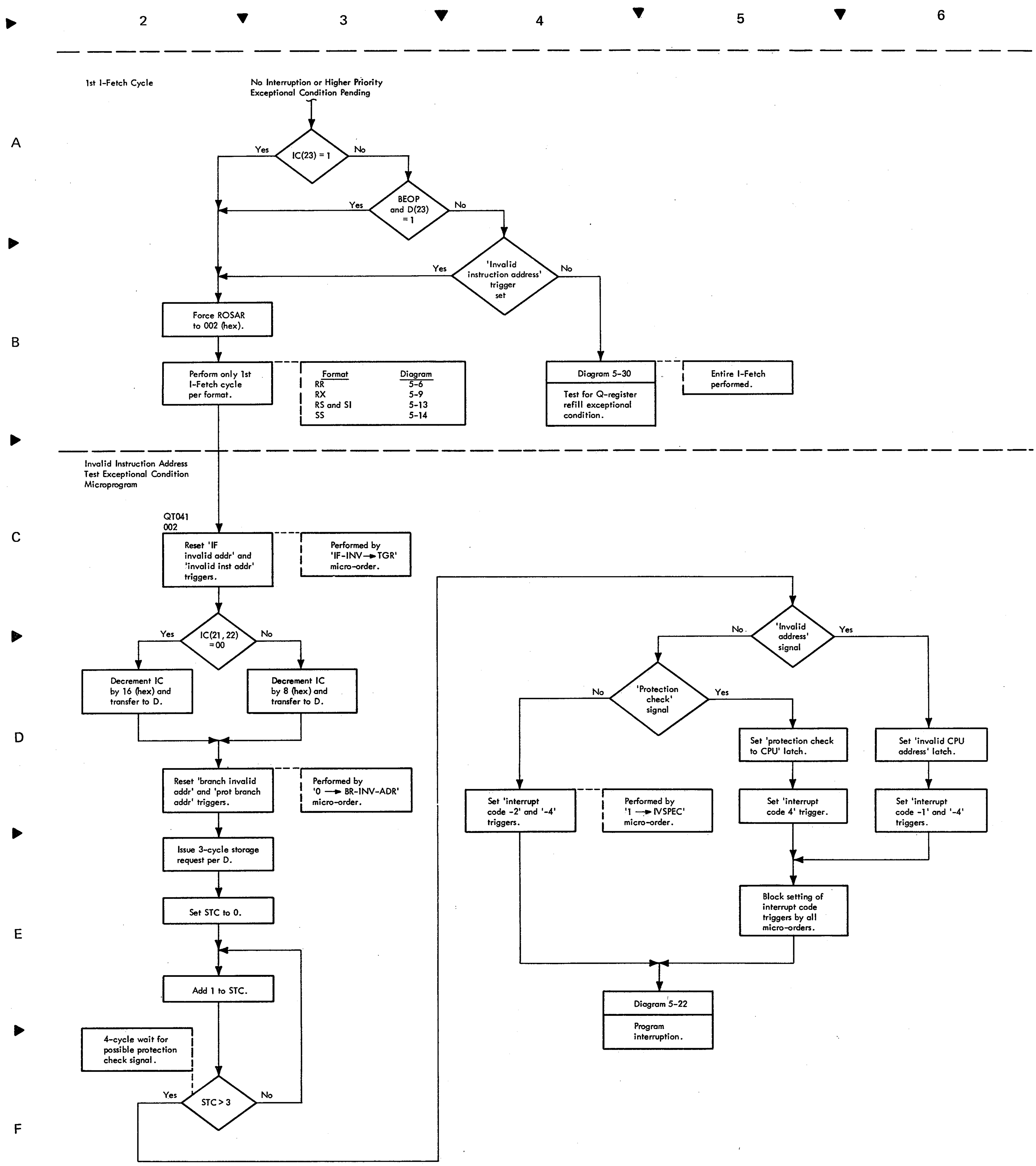
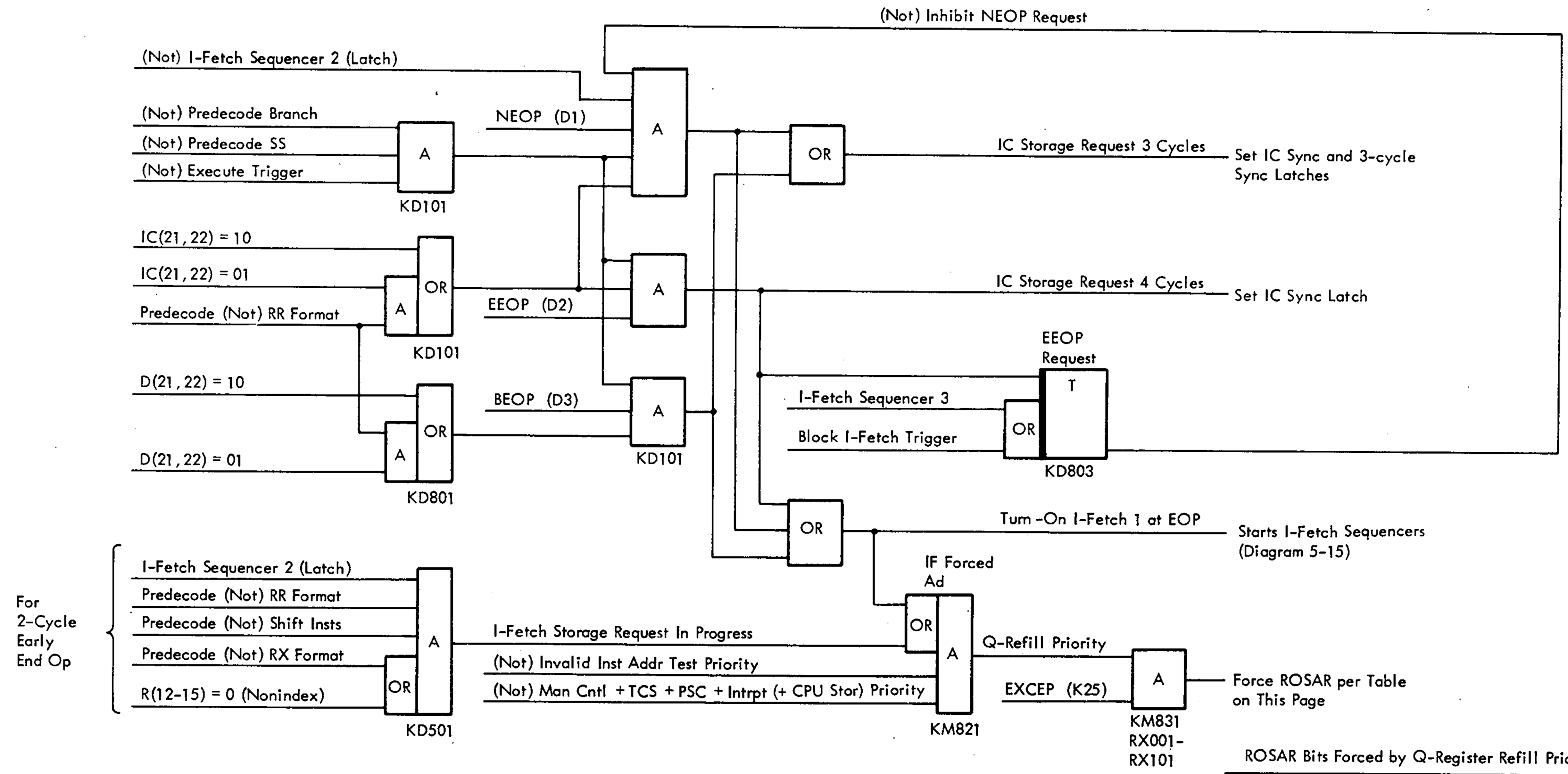


Diagram 5-29. Invalid Instruction Address Test Exceptional Condition (Sheet 2 of 2)

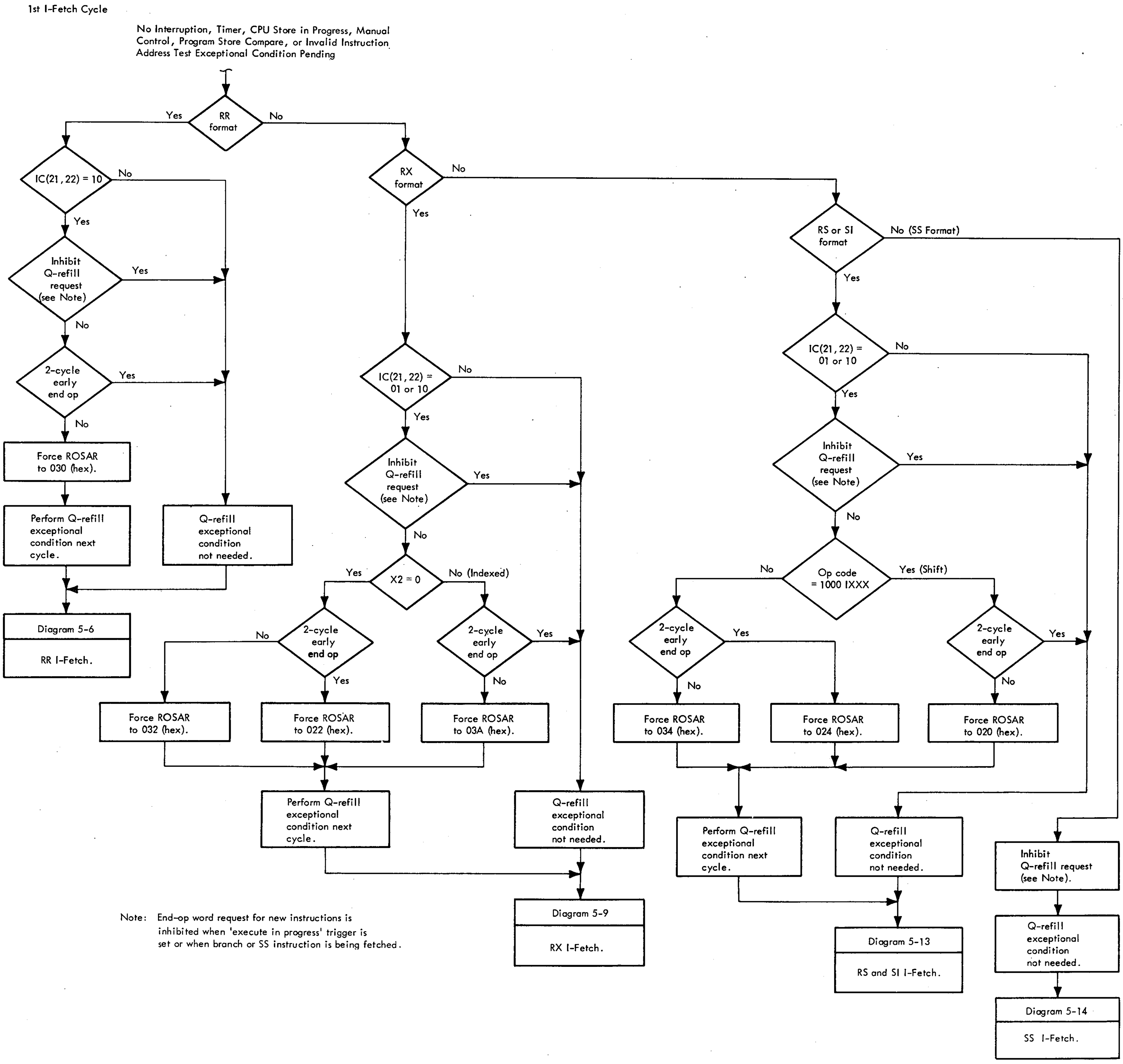
2 3 4 5 6



ROSAR Bits Forced by Q-Register Refill Priority and EXCEP Micro-Order

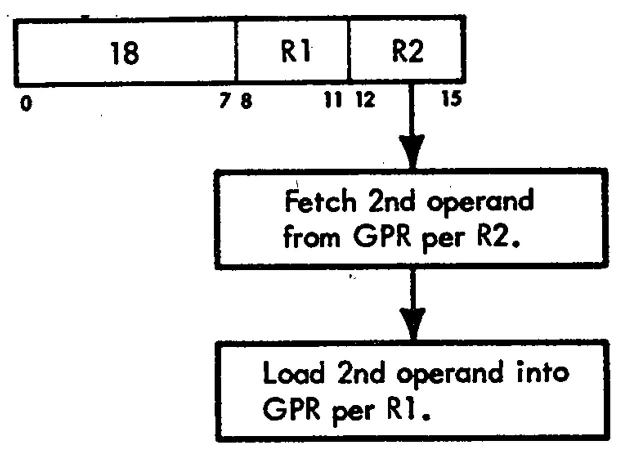
Bit	Setting Condition
0-5	0's (unconditional)
6	1 (unconditional)
7	Not shift • Not 2-cycle early end op
8	Not shift • Not 2-cycle early end op • indexed RX format
9	Not shift • R(0) = 1
10	Not shift • R(1) = 1
11	0 (unconditional)

1st I-Fetch Cycle



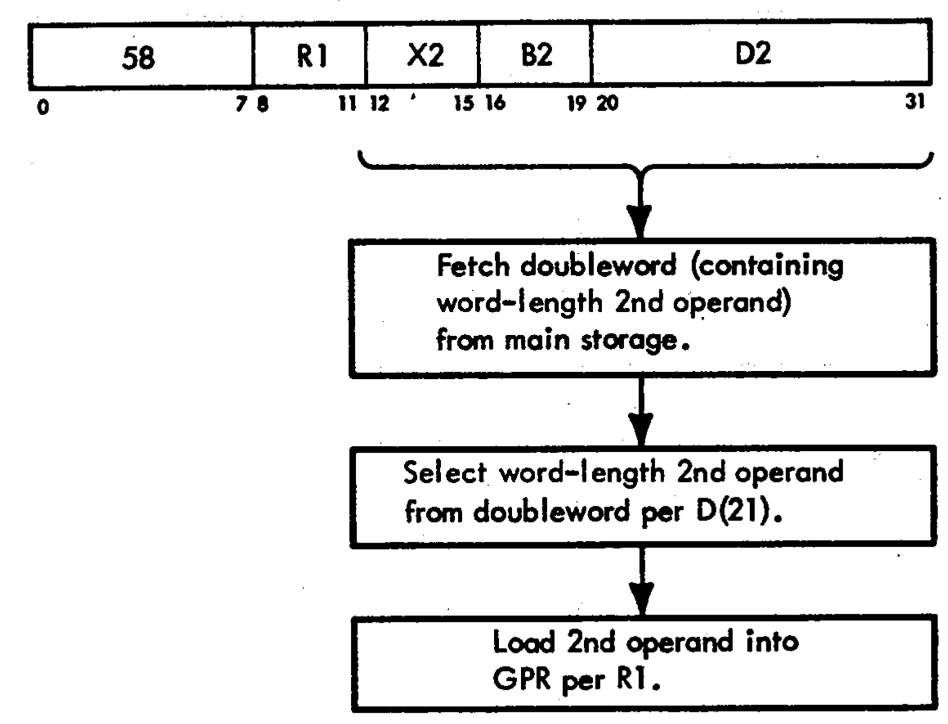
Note: End-op word request for new instructions is inhibited when 'execute in progress' trigger is set or when branch or SS instruction is being fetched.

Diagram 5-30. Test for Q-Register Refill Exceptional Condition



A. Load, LR (18)

- RR format.
- Purpose: Load 2nd operand (in GPR, per R2) into 1st operand location (in GPR, per R1)



B. Load, L (58)

- RX format.
- Purpose: Load 2nd operand (in storage) into 1st operand location (in GPR, per R1).

Diagram 5-101. Load, LR (18); Load, L (58)

- RX format:

48	R1	X2	B2	D2
0	7 8	11 12	15 16	19 20 31

- Purpose: Load halfword 2nd operand (in storage) into 1st operand location (in GPR, per R1).
- Conditions at start of execution:
  1. 1st 16 bits of instruction are in E.
  2. 1st operand is in S and T (not used).
  3. 2nd operand address is in D.
  4. Main storage request for 2nd operand has been issued per D.

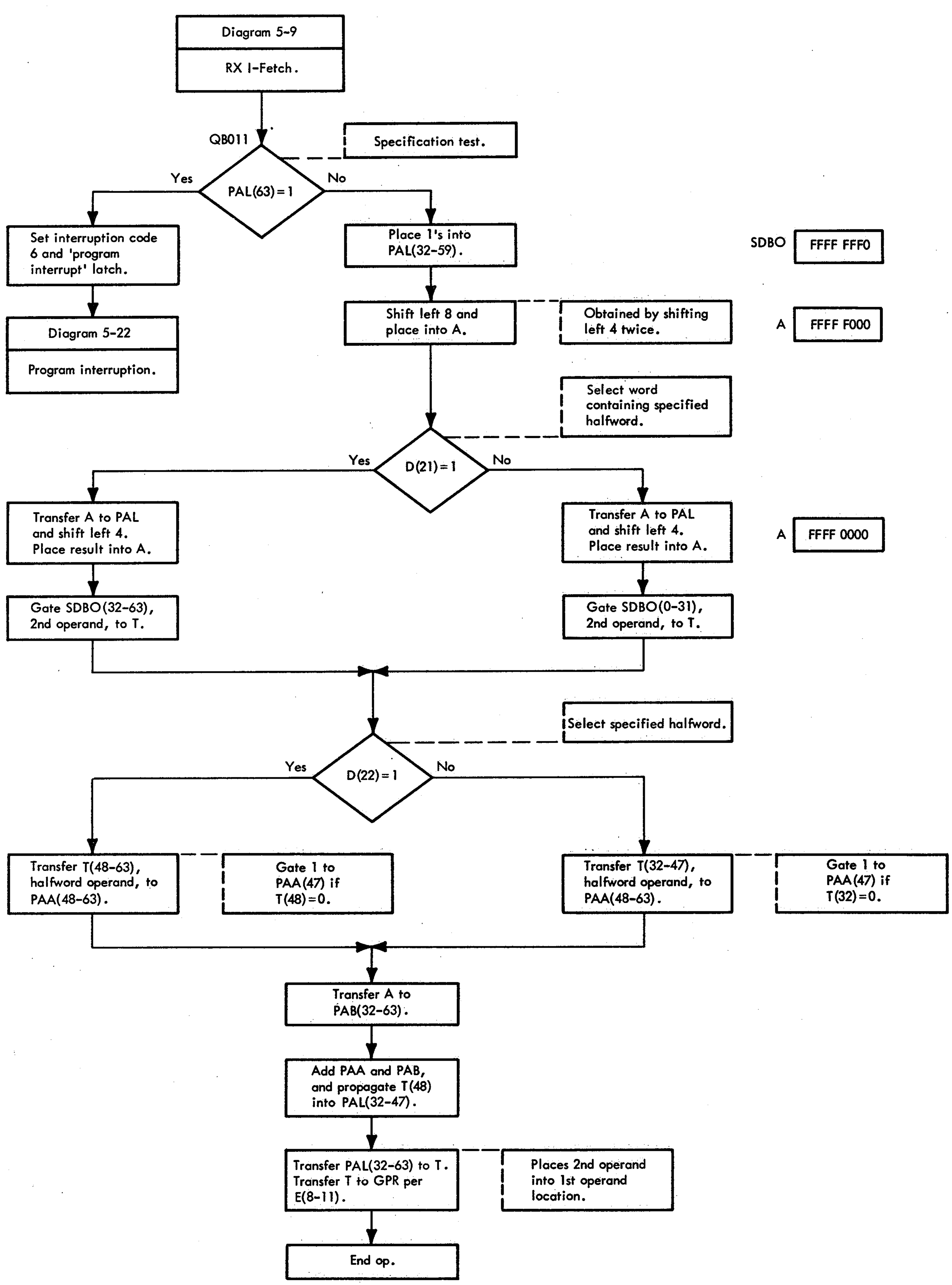
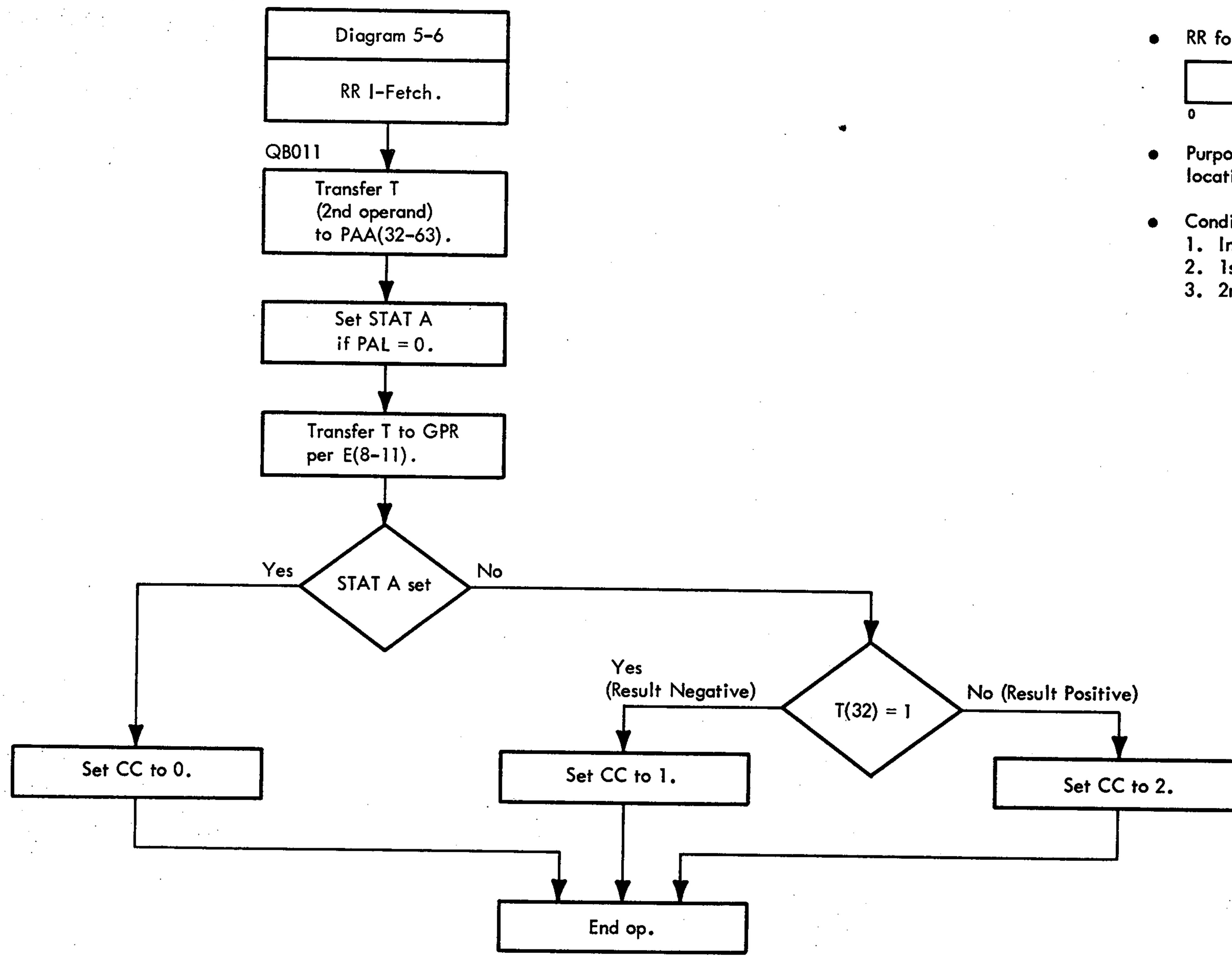


Diagram 5-102. Load Halfword, LH (48)

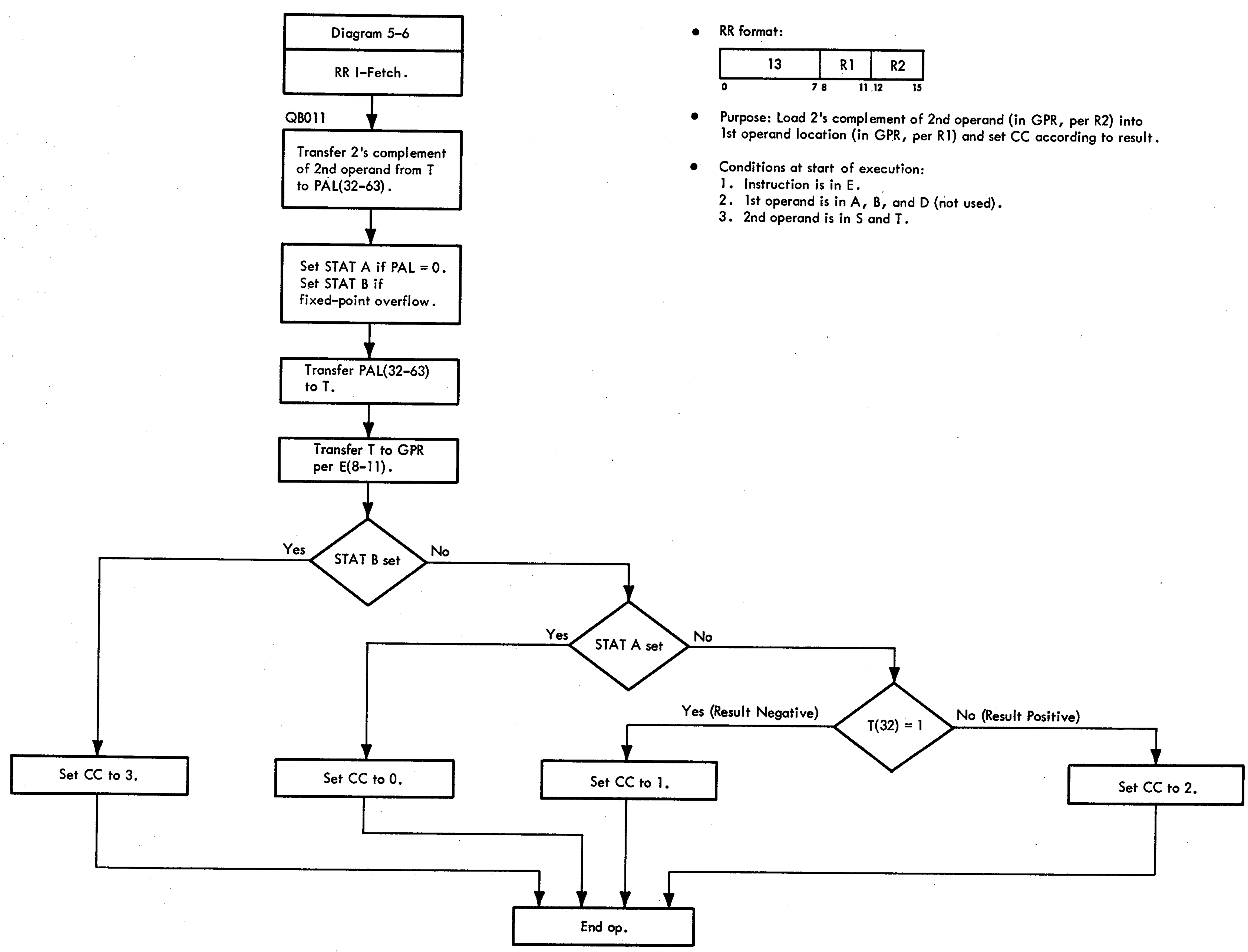


- RR format:

12	R1	R2
0 7	8 11	12 15

- Purpose: Load 2nd operand (in GPR, per R2) into 1st operand location (in GPR, per R1) and set CC according to result.
- Conditions at start of execution:
  1. Instruction is in E.
  2. 1st operand is in A, B, and D (not used).
  3. 2nd operand is in S and T.

Diagram 5-103. Load and Test, LTR (12)



- RR format:

13	R1	R2
0 7	8 11	12 15

- Purpose: Load 2's complement of 2nd operand (in GPR, per R2) into 1st operand location (in GPR, per R1) and set CC according to result.
- Conditions at start of execution:
  1. Instruction is in E.
  2. 1st operand is in A, B, and D (not used).
  3. 2nd operand is in S and T.

Diagram 5-104. Load Complement, LCR (13)

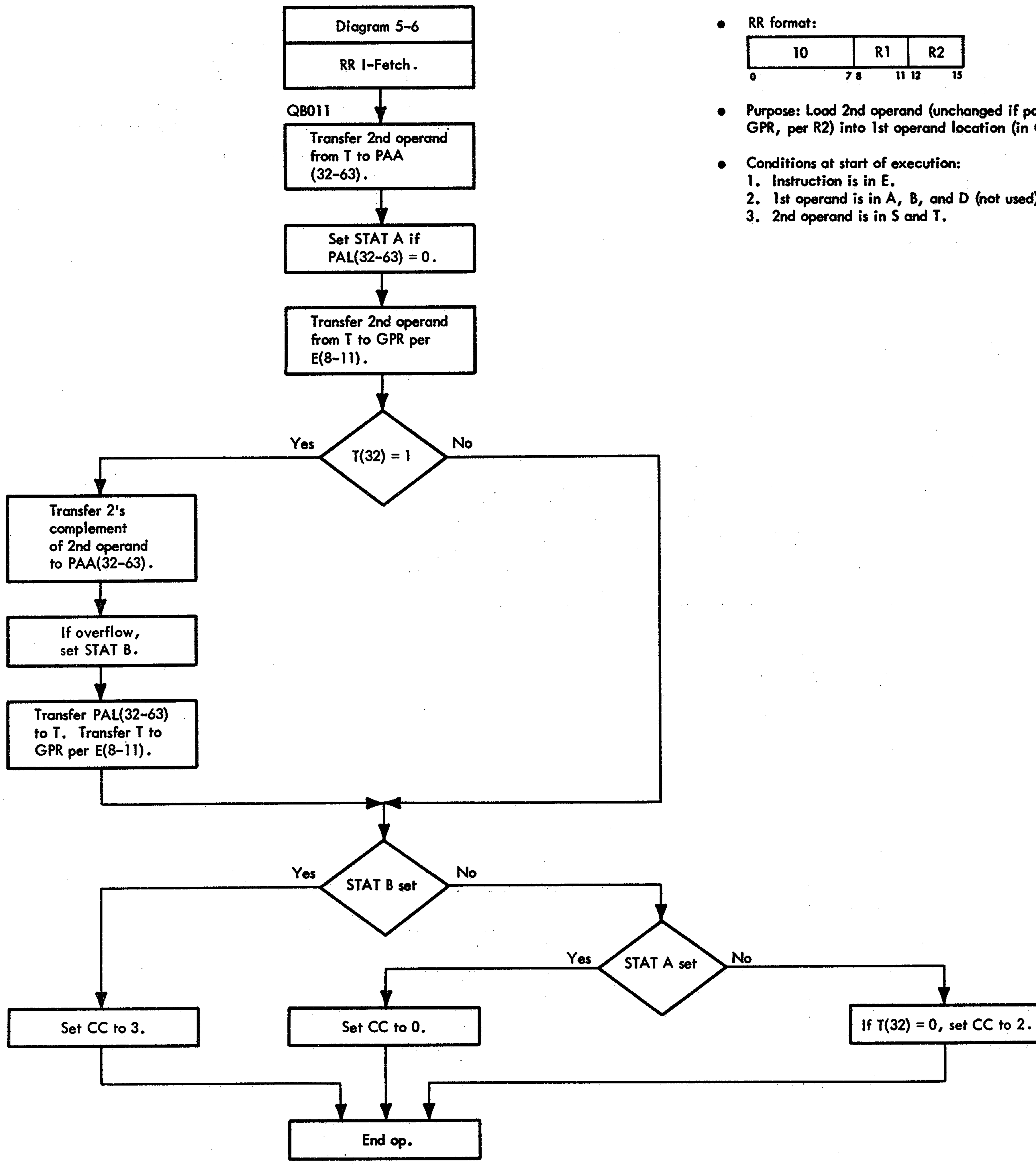


Diagram 5-105. Load Positive, LPR (10)

- RR format:

10	R1	R2
0–7	8–11	12–15

- Purpose: Load 2nd operand (unchanged if positive, 2's complemented if negative; in GPR, per R2) into 1st operand location (in GPR, per R1).

- Conditions at start of execution:
  1. Instruction is in E.
  2. 1st operand is in A, B, and D (not used).
  3. 2nd operand is in S and T.

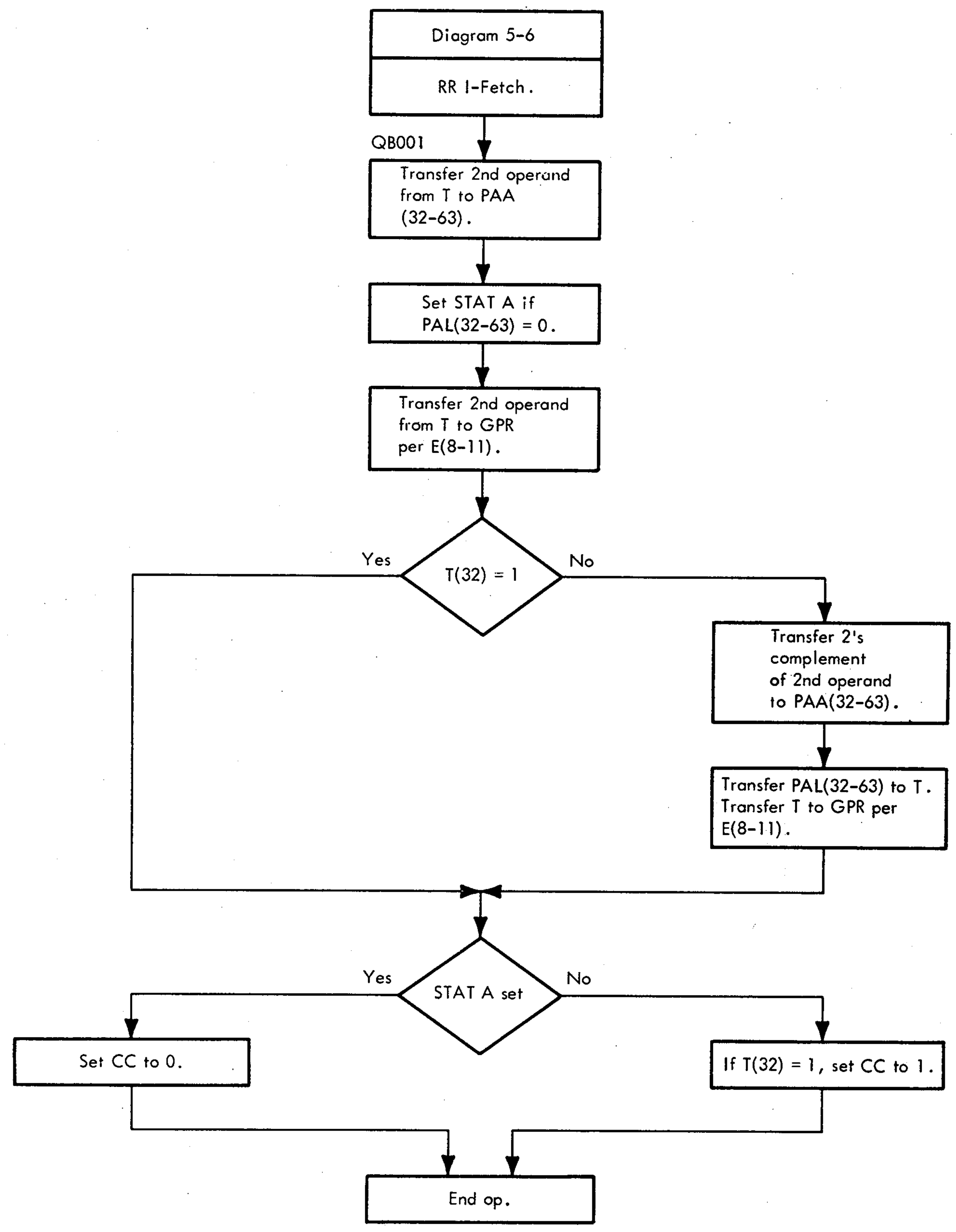


Diagram 5-106. Load Negative, LNR (11)

- RR format:

11	R1	R2
0–7	8–11	12–15

- Purpose: Load 2nd operand (unchanged if negative, 2's complemented if positive; in GPR per R2) into 1st operand location (in GPR, per R1).

- Conditions at start of execution:
  1. Instruction is in E.
  2. 1st operand is in A, B, and D (not used).
  3. 2nd operand is in S and T.

- RS format:

98	R1	R3	B2	D2
0–7	8–11	12–15	16–19	20–31

- Purpose: Load 2nd operand (as many words as required; in storage) into GPR's, in ascending order, starting with 1st operand location (per R1) and ending with 3rd operand location (per R3).
- Conditions at start of execution:
  1. 1st 16 bits of instruction are in E.
  2. 1st operand is in S and T (not used).
  3. 2nd operand address is in D.
  4. Main storage request for 2nd operand has been issued per D.

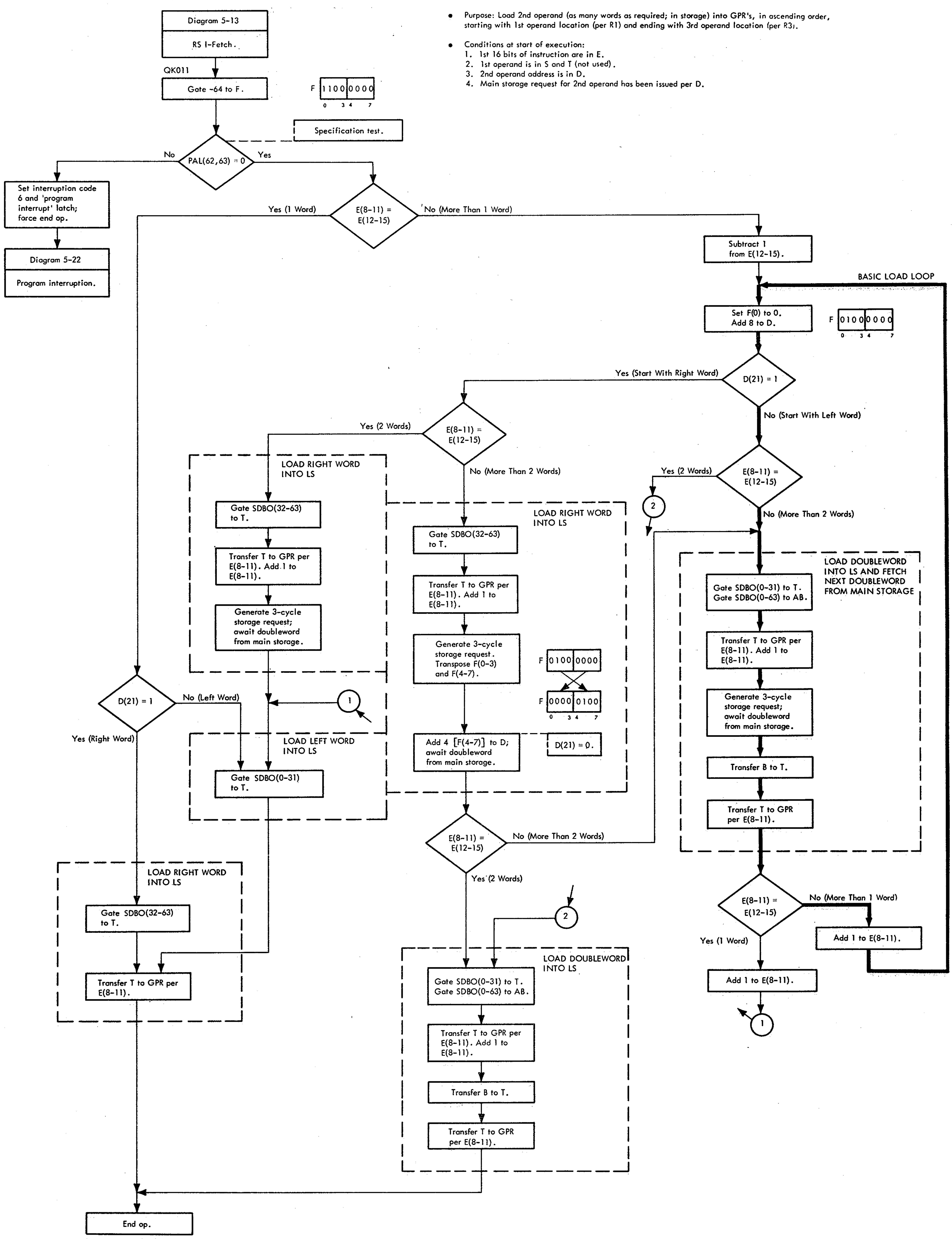


Diagram 5-107. Load Multiple, LM (98)

2 3 4 5 6

A B C D E F G H

- RR format: AR, ALR, SR, SLR, CR.

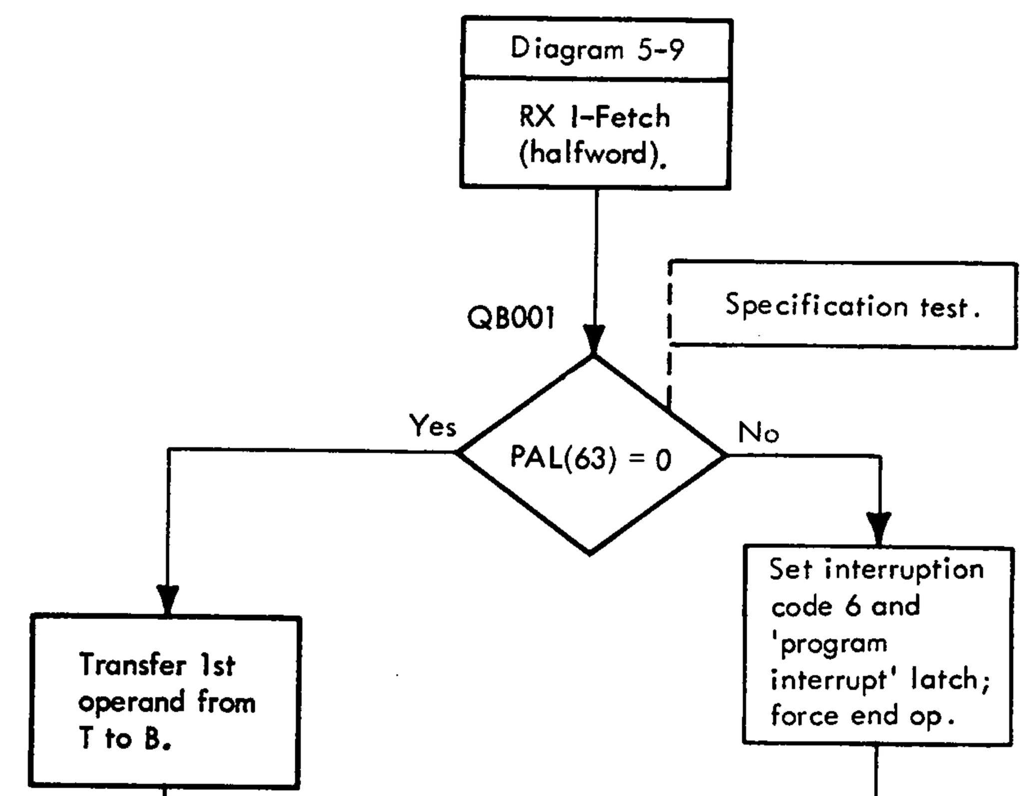
Op Code	R1	R2
0–7	8–11	12–15

- Purpose:
  1. AR, ALR, SR, SLR - Algebraically add (subtract) 2nd operand (in GPR, per R2) to (from) 1st operand (in GPR, per R1) and place result into 1st operand location.
  2. CR - Algebraically compare 1st operand (in GPR, per R1) with 2nd operand (in GPR, per R2) and set CC according to result.
- Conditions at start of execution:
  1. Instruction is in E.
  2. 1st operand is in A, B, and D.
  3. 2nd operand is in S and T.
- Op codes:
  1. Add, AR = 1A.
  2. Add Logical, ALR = 1E.
  3. Subtract, SR = 1B.
  4. Subtract Logical, SLR = 1F.
  5. Compare, CR = 19.

- RX format: A, AH, AL, S, SH, SL, C, CH.

Op Code	R1	X2	B2	D2
0–7	8–11	12–15	16–19	20–31

- Purpose:
  1. A, AL, S, SL - Algebraically add (subtract) 2nd operand (in storage) to (from) 1st operand (in GPR, per R1) and place result into 1st operand location.
  2. AH, SH - Algebraically add (subtract) halfword 2nd operand (in storage) to (from) 1st operand (in GPR, per R1) and place result into 1st operand location.
  3. C - Algebraically compare 1st operand (in GPR, per R1) with 2nd operand (in storage) and set CC according to result.
  4. CH - Algebraically compare 1st operand (in GPR, per R1) with halfword 2nd operand (in storage) and set CC according to result.
- Conditions at start of execution:
  1. 1st 16 bits of instruction are in E.
  2. 1st operand is in S and T.
  3. 2nd operand address is in D.
  4. Main storage request for 2nd operand has been issued per D.
- Op codes:
  1. Add, A = 5A.
  2. Add Halfword, AH = 4A.
  3. Add Logical, AL = 5E.
  4. Subtract, S = 5B.
  5. Subtract Halfword, SH = 4B.
  6. Subtract Logical, SL = 5F.
  7. Compare, C = 59.
  8. Compare Halfword, CH = 49.



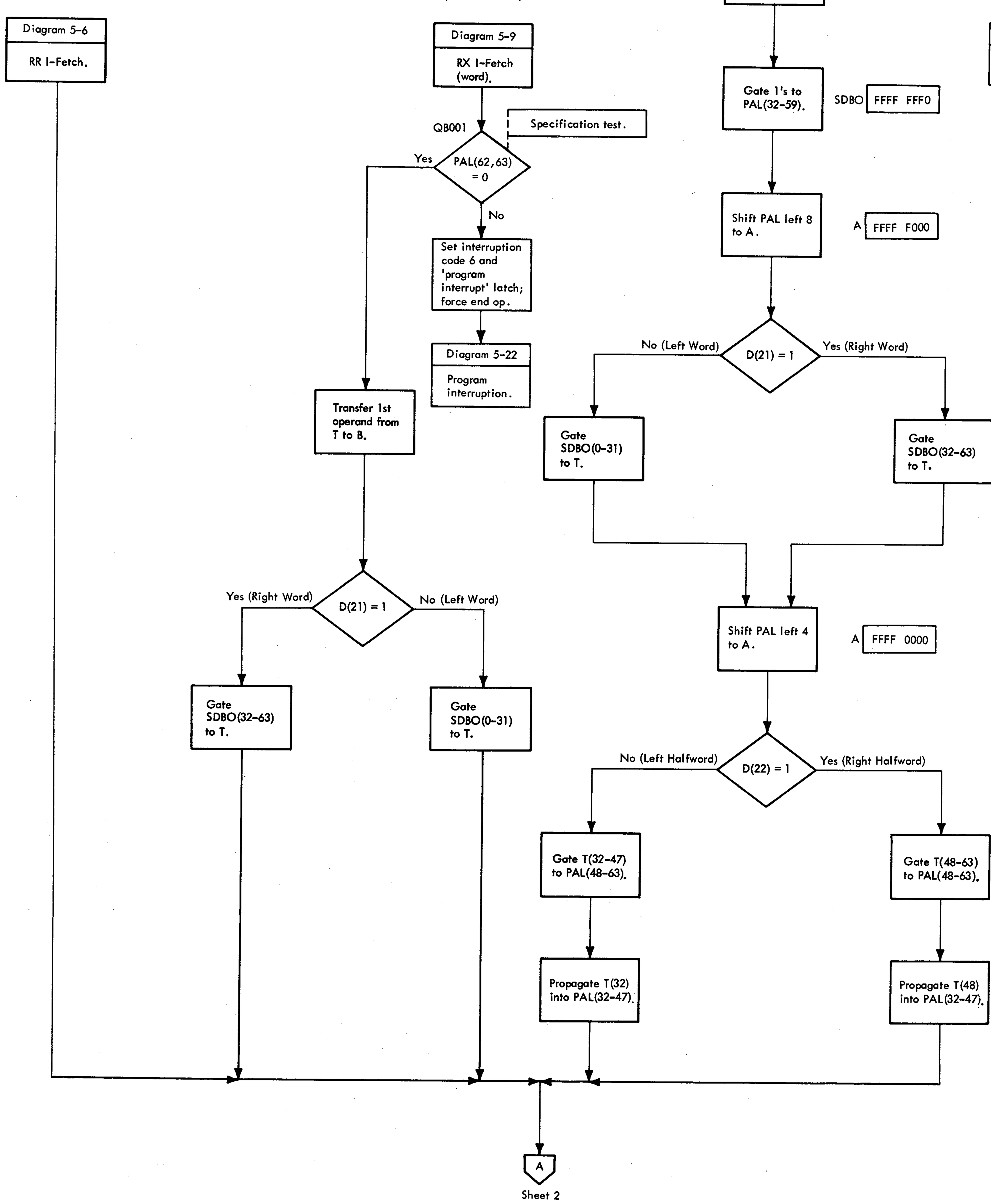
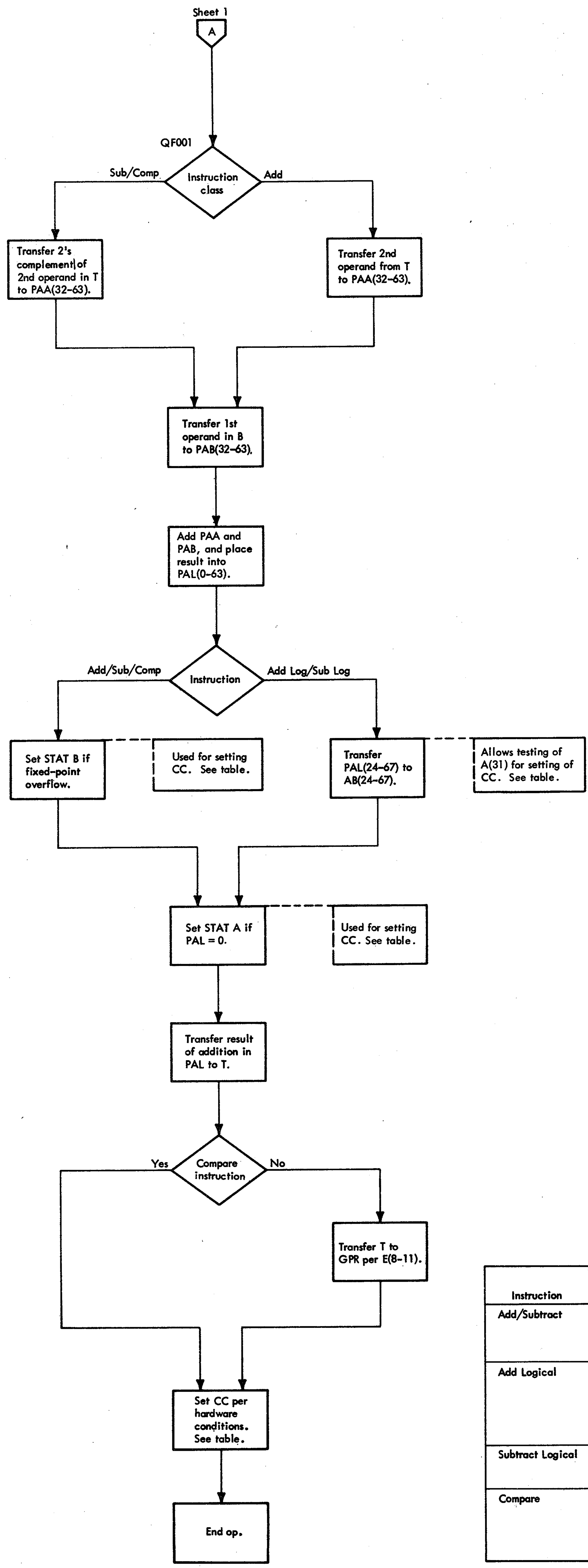


Diagram 5-108. Fixed-Point Add-Type Instructions (Sheet 1 of 2)



Instruction	STAT A	STAT B	State of Bit Tested	CC	Comments
Add/Subtract	1	0	T(32) = 0	0	PAL equals zero.
	0	0	T(32) = 1	1	PAL is less than zero.
	0	0	T(32) = 0	2	PAL is greater than zero.
	0	1	T(32) = 0	3	Overflow.
Add Logical	1	0	A(31) = 0	0	PAL equals zero and no carry from PAL(32).
	0	0	A(31) = 0	1	PAL does not equal zero and no carry from PAL(32).
	1	0	A(31) = 1	2	PAL equals zero and carry from PAL(32).
	0	0	A(31) = 1	3	PAL does not equal zero and carry from PAL(32).
Subtract Logical	0	0	A(31) = 0	1	Same as Add Logical.
	1	0	A(31) = 1	2	Same as Add Logical.
	0	0	A(31) = 1	3	Same as Add Logical.
Compare	1	0	T(32) = 0	0	Operands are equal.
	0	1	T(32) = 0	1	1st operand is less than 2nd operand.
	0	0	T(32) = 1	1	1st operand is less than 2nd operand.
	0	0	T(32) = 0	2	1st operand is greater than 2nd operand.
	0	1	T(32) = 1	2	1st operand is greater than 2nd operand.

Diagram 5-108. Fixed-Point Add-Type Instructions (Sheet 2 of 2)

Multiply (MR) 1C

- RR format:

1C	R1	R2
0 7	8 11	12 15

- Purpose: Multiply 1st operand (in GPR, per R1 + 1) by 2nd operand (in GPR, per R2) and place 64-bit result into 1st operand location (in GPR, per R1 and R1 + 1).

- Conditions at start of execution:
  1. Instruction is in E.
  2. Contents of even-address GPR specified by R1 is in A, B, and D (not used).
  3. Multiplicand (1st operand) is in odd-address GPR specified by R1 + 1.
  4. Multiplier (2nd operand) is in S and T.

Multiply (M), 5C and Multiply Halfword (MH), 4C

- RX format:

5C, 4C	R1	X2	B2	D2
0 7	8 11	12 15	16 19	20 31

- Purpose:
  1. M - Multiply 1st operand (in GPR, per R1 + 1) and 2nd operand (in storage) and place 64-bit result into 1st operand location (in GPR, per R1 and R1 + 1).
  2. MH - Multiply 1st operand (in GPR, per R1) and halfword 2nd operand (in storage) and place low-order 32 bits of result into 1st operand location.

- Conditions at start of execution:
  1. 1st 16 bits of instruction are in E.
  2. Contents of even-address GPR specified by R1 is in S and T for M instruction (not used).
  3. 1st operand for M instruction is in odd-address GPR specified by R1 + 1.
  4. 1st operand for MH instruction is in S and T.
  5. 2nd operand address is in D.
  6. Main storage request for 2nd operand has been issued per D.

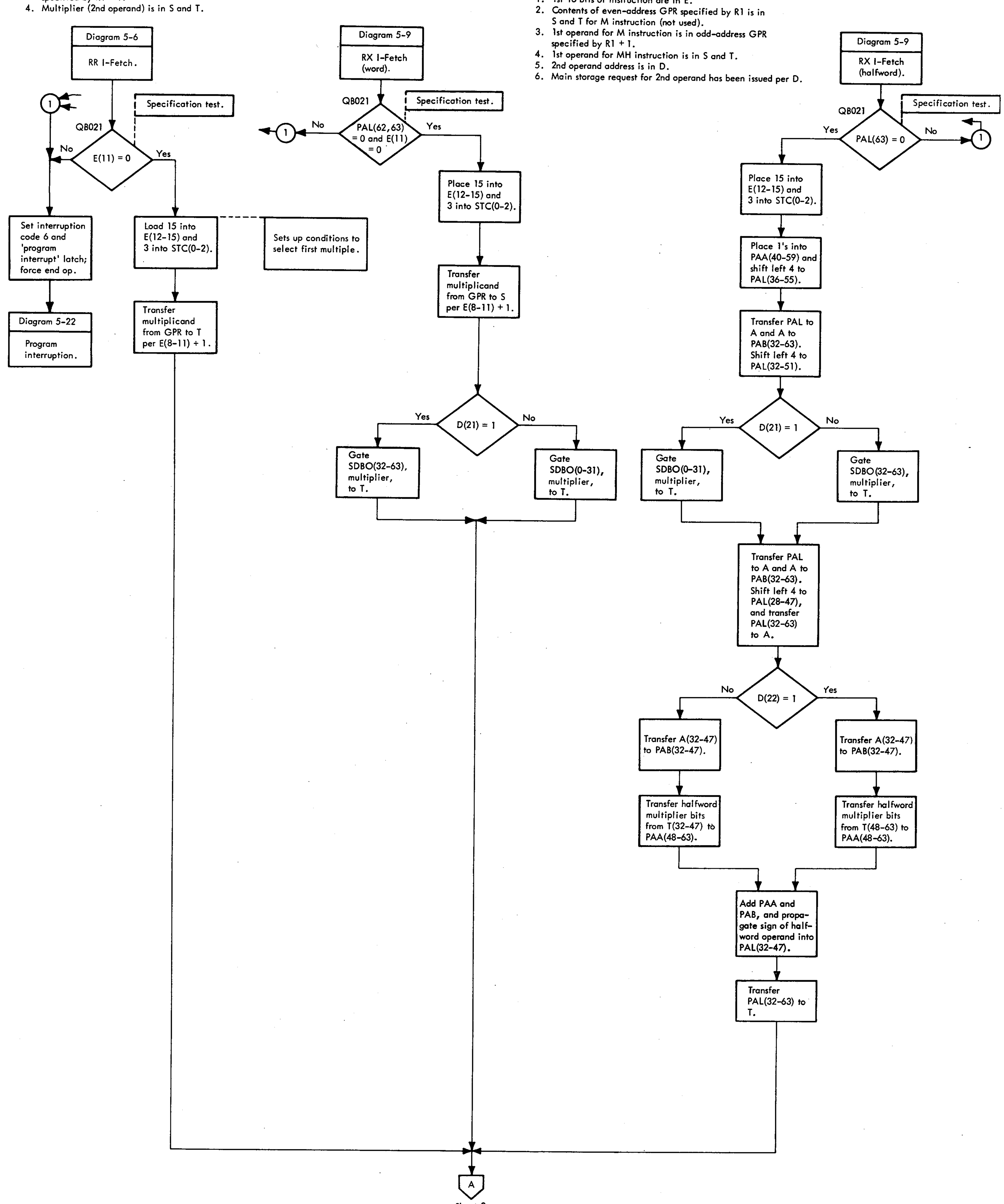


Diagram 5-109. Fixed-Point Multiply (Sheet 1 of 3)

Objectives:

1. Select multiple (of T).
   a. Select M1,M2 bits from S per E(12-15) (A of Sheet 3).
   b. Select multiple (of T) per M1,M2 bits and 'TX' trigger (Table 1, Sheet 3).
2. Develop partial product (PP) bit-pair.
   a. Add multiple to PP in AB (shifted right 2) (B of Sheet 3).
   b. Gate new PP to AB.
   c. Gate PP bit-pair [B(66,67)] to SAL per E(14,15) (C of Sheet 3).
3. Develop PP byte.
   Add SAL to F.
4. Develop low-order PP in S.
   a. When PP byte is complete, gate SAL to S per STC.
   b. When S is full (4 bytes of PP has been loaded), low-order product is in S and high-order product is in PAL.
5. Store product.
   a. Store high-order product (in PAL) into GPR per R1 (even register).
   b. Store low-order product (in S) into GPR per R1 + 1 (odd register) (per R1 for MH instruction).

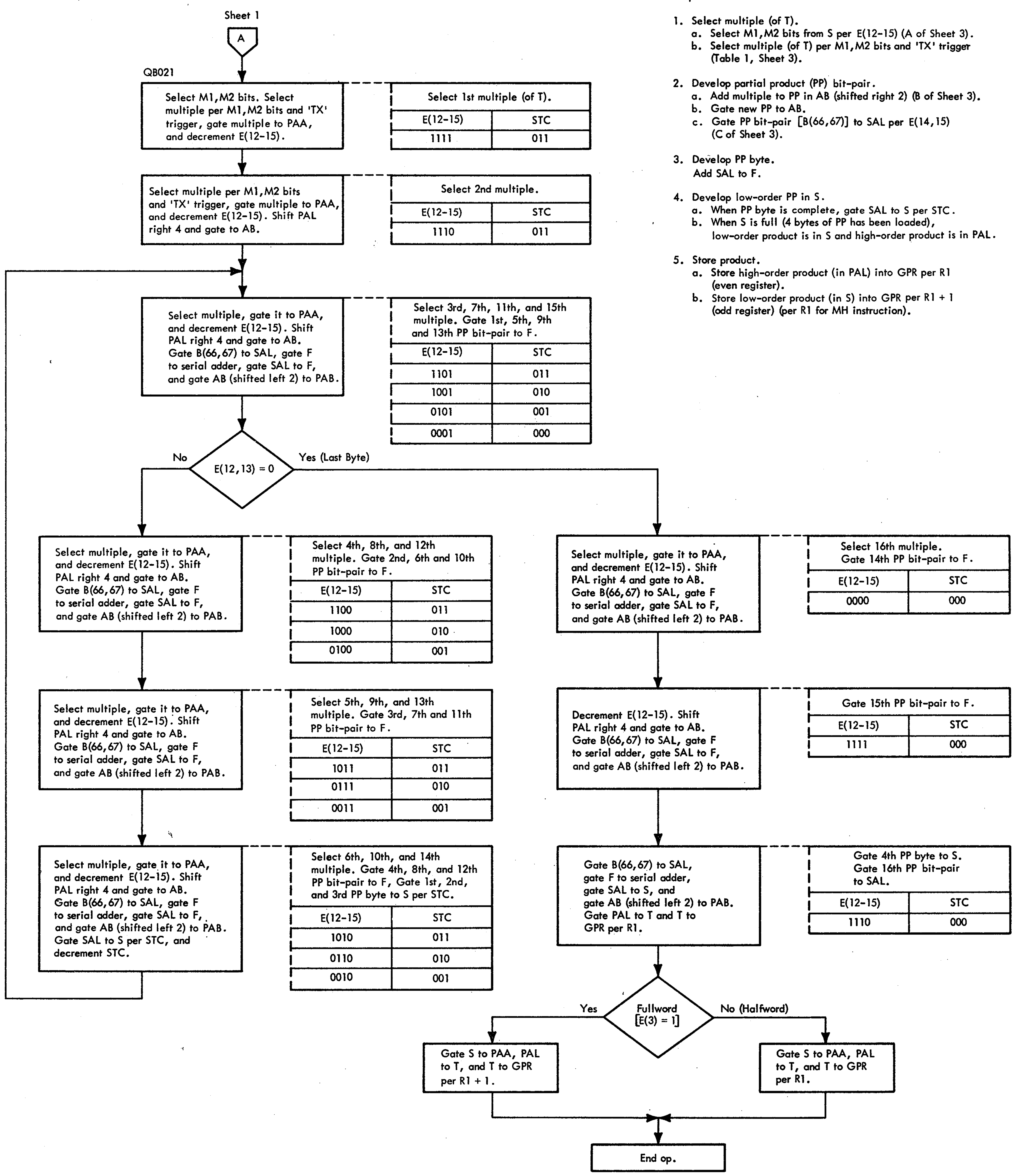


Diagram 5-109. Fixed-Point Multiply (Sheet 2 of 3)

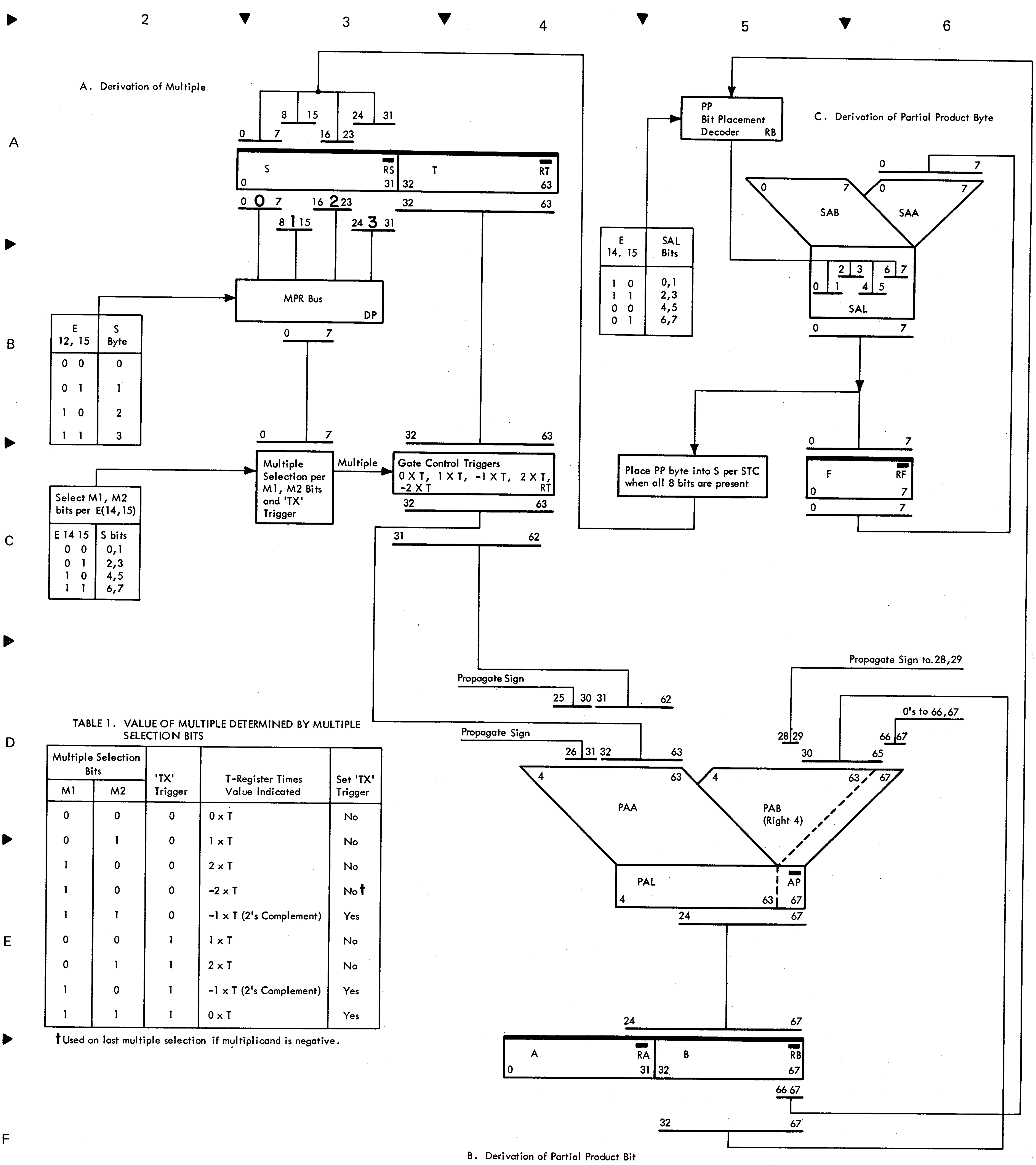


TABLE 1. VALUE OF MULTIPLE DETERMINED BY MULTIPLE SELECTION BITS

Multiple Selection Bits		'TX' Trigger	T-Register Times Value Indicated	Set 'TX' Trigger
M1	M2			
0	0	0	0 x T	No
0	1	0	1 x T	No
1	0	0	2 x T	No
1	0	0	-2 x T	No†
1	1	0	-1 x T (2's Complement)	Yes
0	0	1	1 x T	No
0	1	1	2 x T	No
1	0	1	-1 x T (2's Complement)	Yes
1	1	1	0 x T	Yes

†Used on last multiple selection if multiplicand is negative.

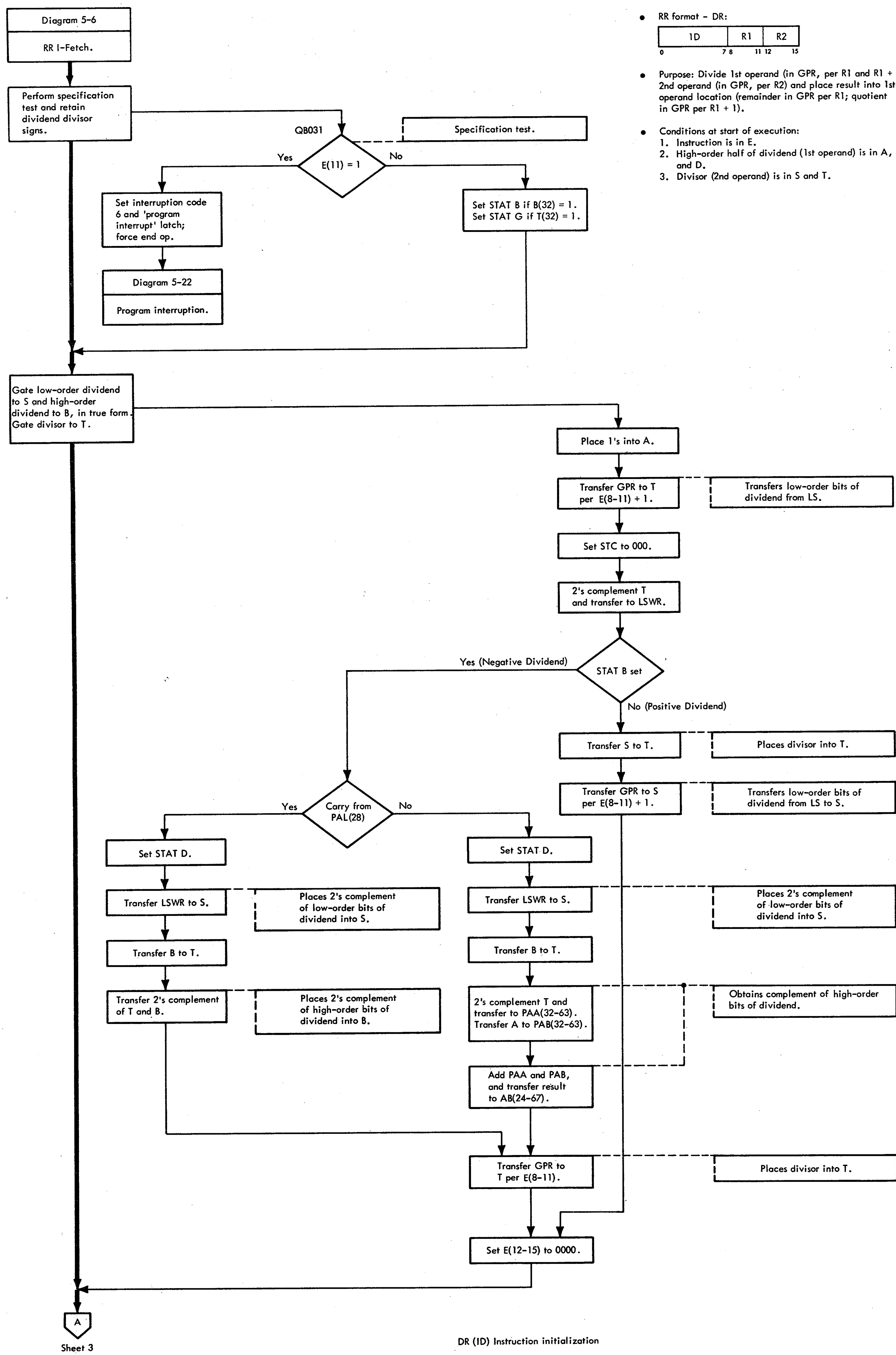
Diagram 5-109. Fixed-Point Multiply (Sheet 3 of 3)

- RR format - DR:

1D	R1	R2
0 7	8 11	12 15

- Purpose: Divide 1st operand (in GPR, per R1 and R1 + 1) by 2nd operand (in GPR, per R2) and place result into 1st operand location (remainder in GPR per R1; quotient in GPR per R1 + 1).

- Conditions at start of execution:
  1. Instruction is in E.
  2. High-order half of dividend (1st operand) is in A, B, and D.
  3. Divisor (2nd operand) is in S and T.



DR (ID) Instruction initialization

Diagram 5-110. Fixed-Point Divide (Sheet 1 of 6)

- RX format – D:

5D	R1	X2	B2	D2
0 7	8 11	12 15	16 19	20 31

- Purpose: Divide 1st operand (in GPR, per R1 and R1 + 1) by 2nd operand (in storage) and place result into 1st operand location (remainder in GPR per R1; quotient in GPR per R1 + 1).
- Conditions at start of execution:
  1. 1st 16 bits of instruction are in E.
  2. High-order half of dividend (1st operand) is in S and T.
  3. Divisor (2nd operand) address is in D.
  4. Main storage request for divisor has been issued per D.

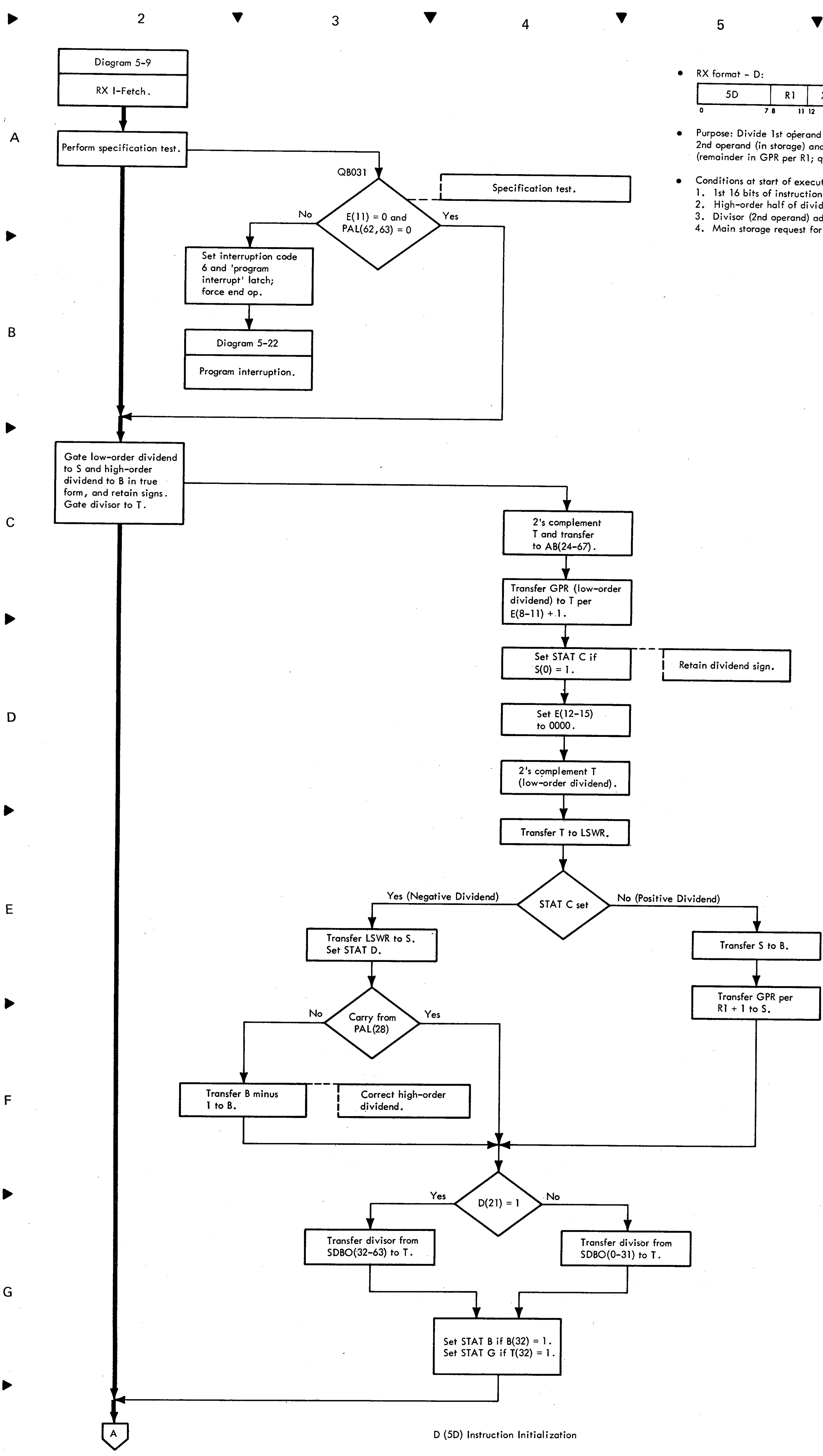


Diagram 5-110. Fixed-Point Divide (Sheet 2 of 6)

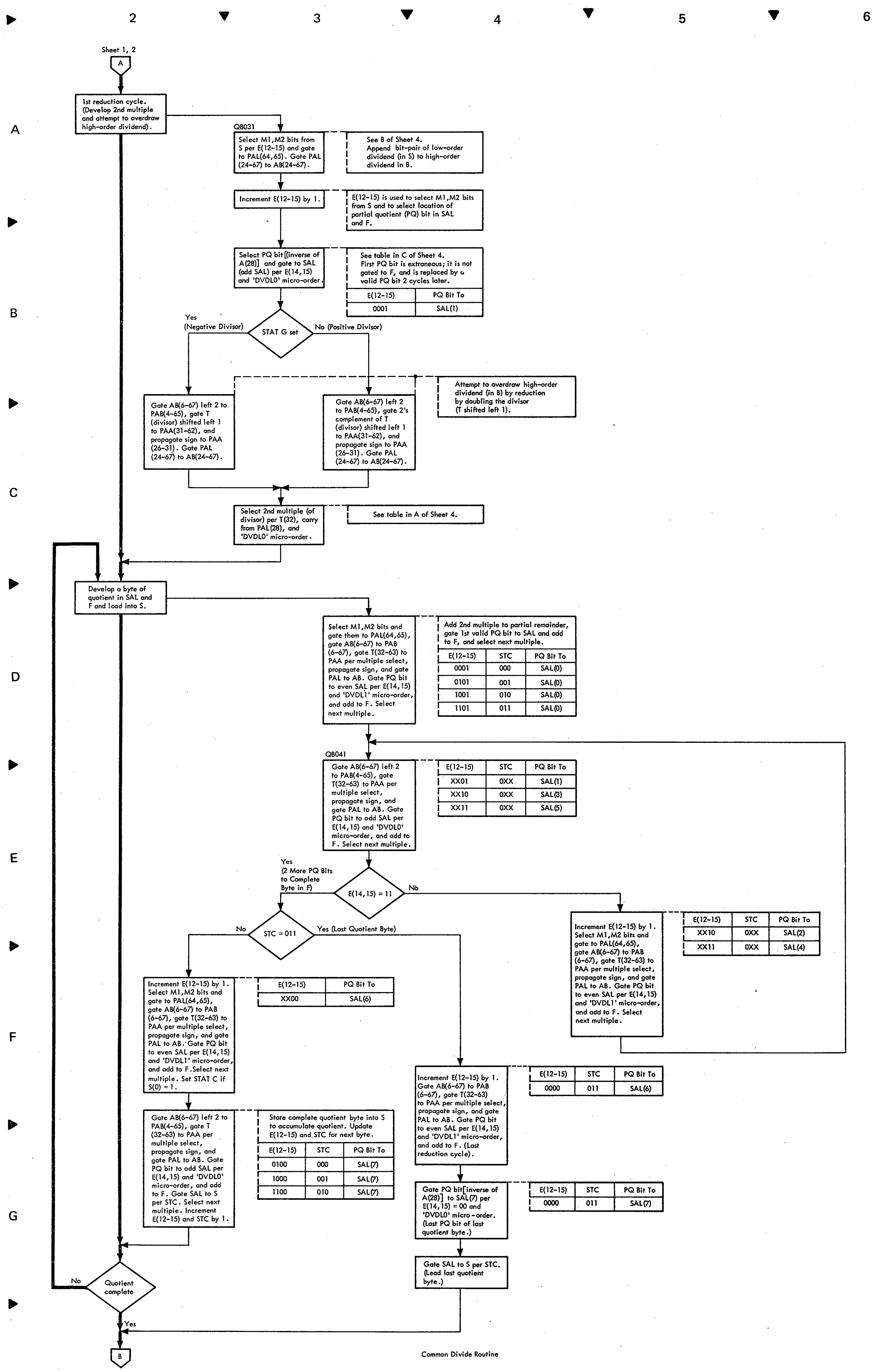


Diagram 5-110. Fixed-Point Divide (Sheet 3 of 6)

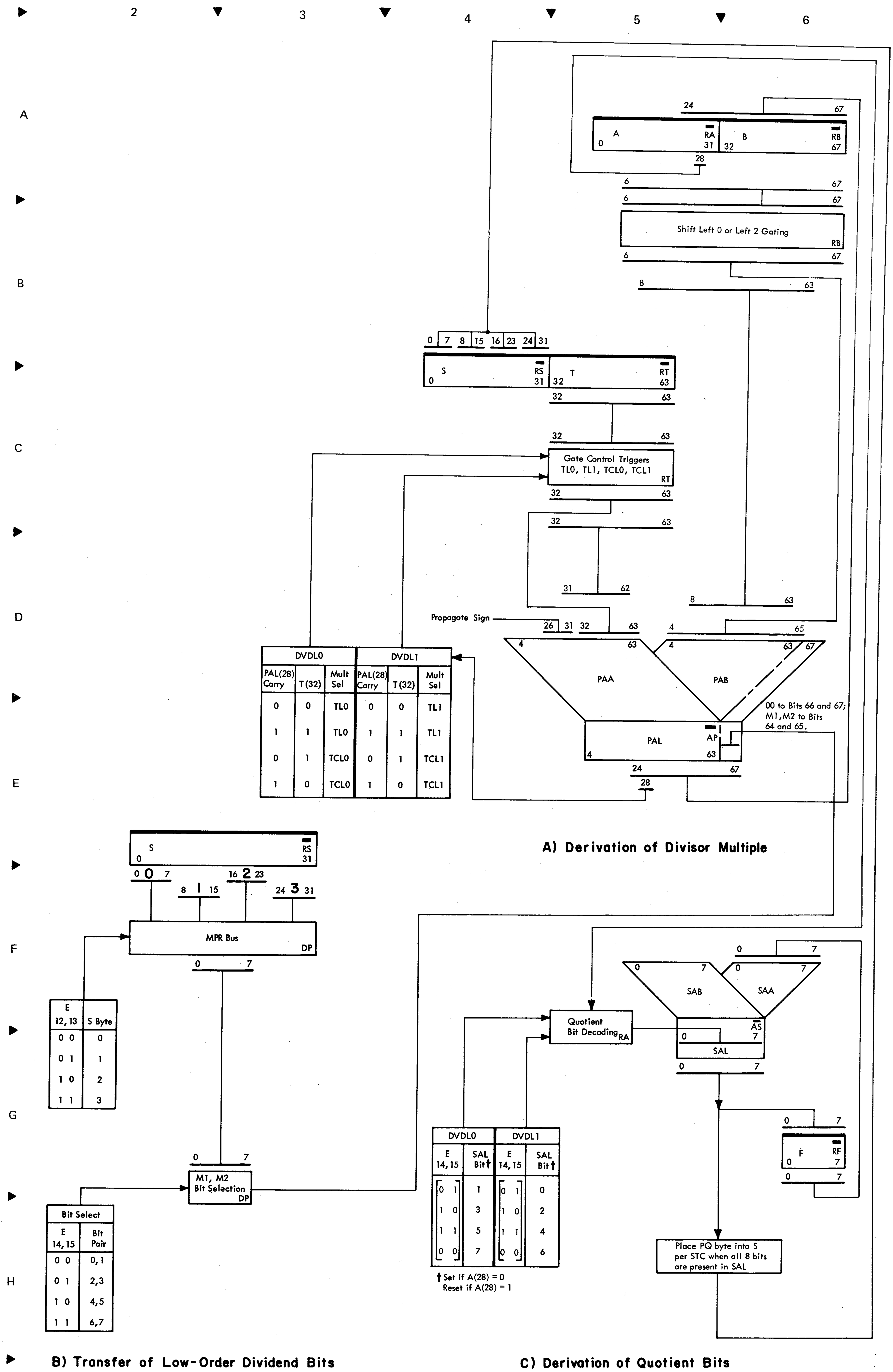


Diagram 5-110. Fixed-Point Divide (Sheet 4 of 6)

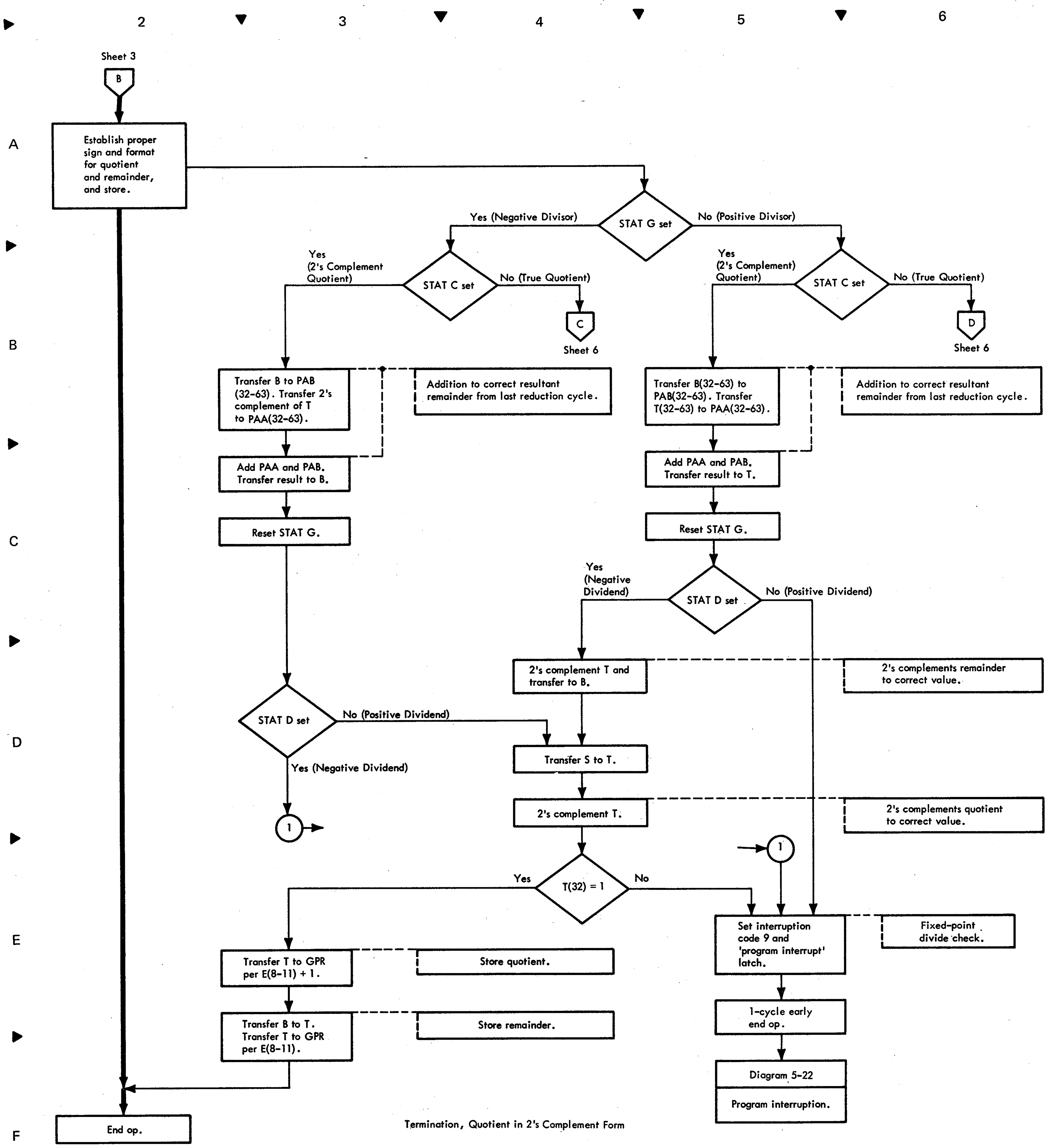
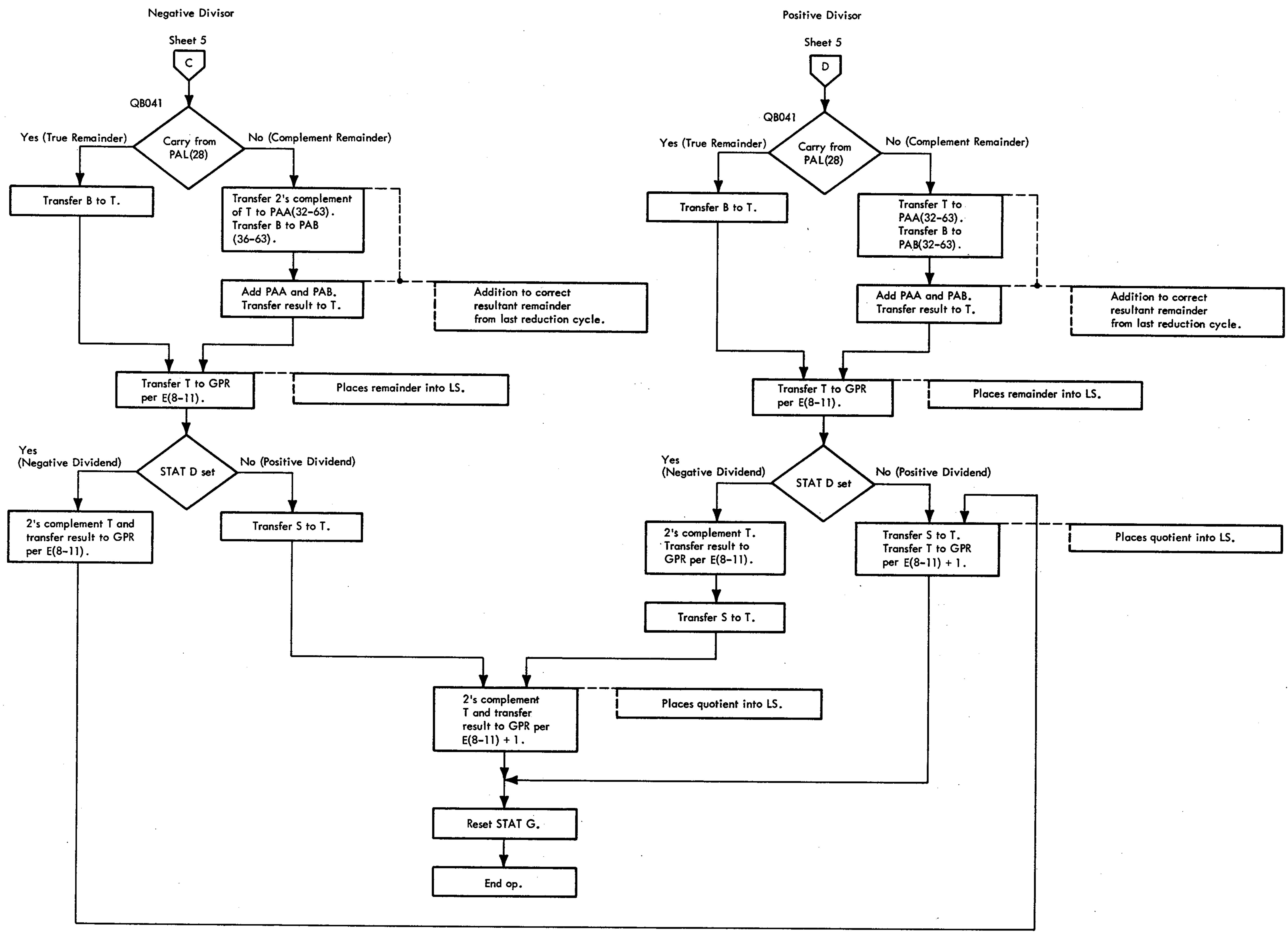


Diagram 5-110. Fixed-Point Divide (Sheet 5 of 6)

2 3 4 5 6



Termination, Quotient in True Form

Diagram 5-110. Fixed-Point Divide (Sheet 6 of 6)

A B C D E F G H

- RX format:

4F	R1	X2	B2	D2
0 7	8 11	12 15	16 19	20 31

- Purpose: Convert radix of 2nd operand (in storage) from decimal to binary and place result into 1st operand location (in GPR, per R1).
- Conditions at start of execution:
  1. 1st 16 bits of instruction are in E.
  2. 1st operand is in S and T (not used).
  3. 2nd operand address is in D.
  4. Main storage request for 2nd operand has been issued per D.

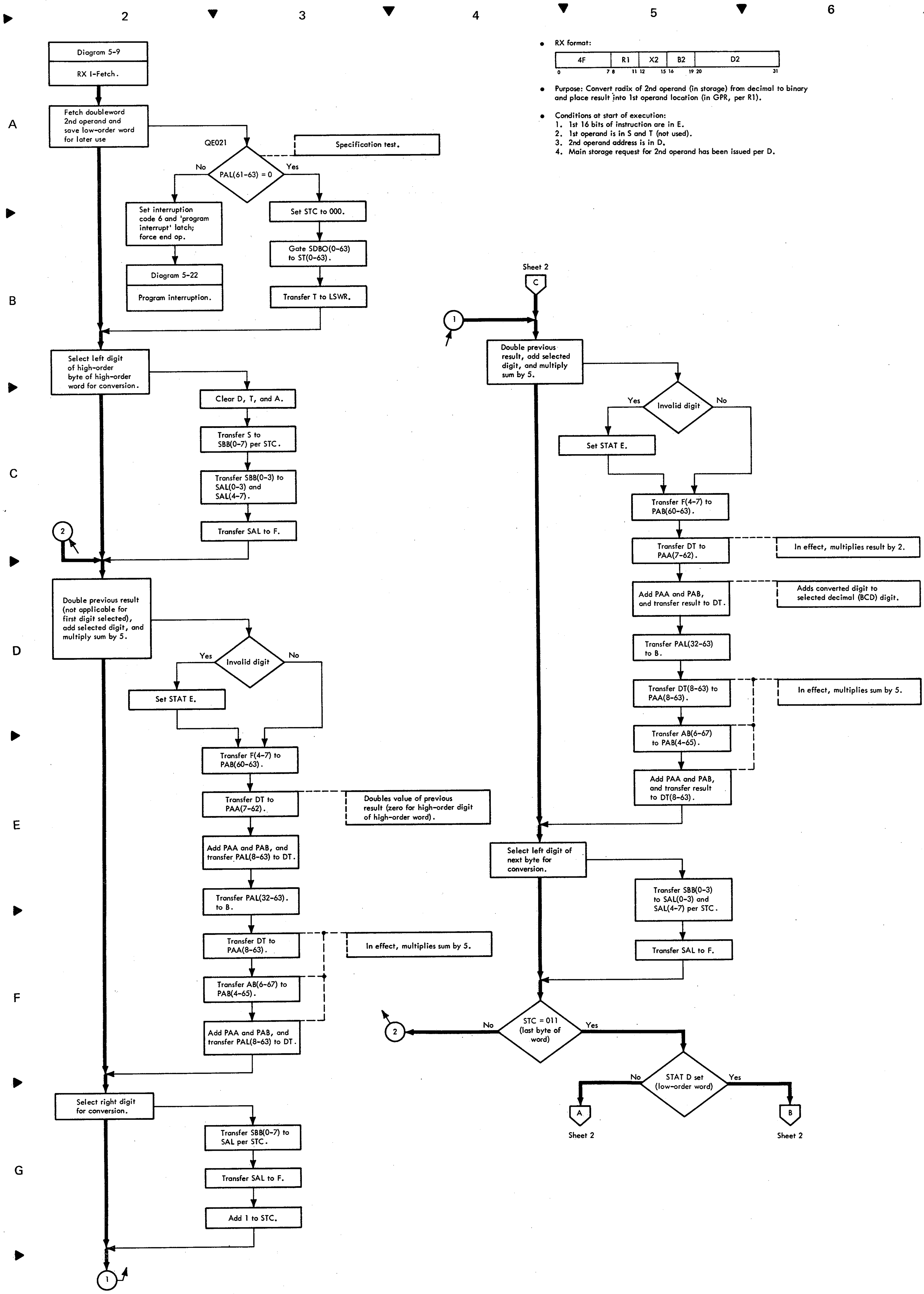


Diagram 5-111. Convert to Binary, CVB (4F) (Sheet 1 of 2)

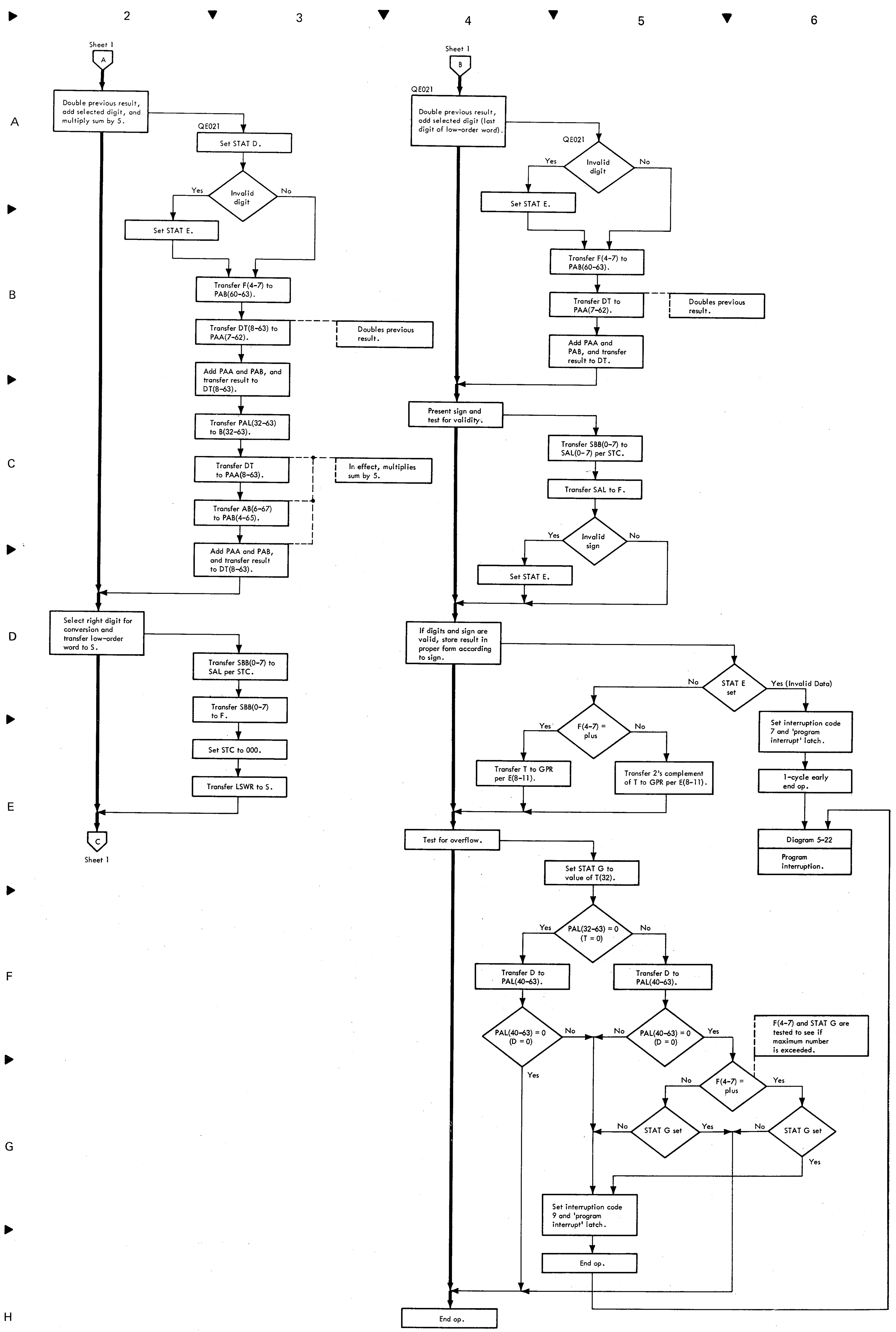


Diagram 5-111. Convert to Binary, CVG (4F) (Sheet 2 of 2)

- RX format:

4E	R1	X2	B2	D2
0 7	8 11	12 15	16 19	20 31

- Convert radix of 1st operand (in GPR, per R1) from binary to decimal and place result into 2nd operand location (in storage).
- Conditions at start of execution:
  1. 1st 16 bits of instruction are in E.
  2. 1st operand is in S and T.
  3. 2nd operand address is in D.
  4. Main storage request for 2nd operand has been issued per D (not used).

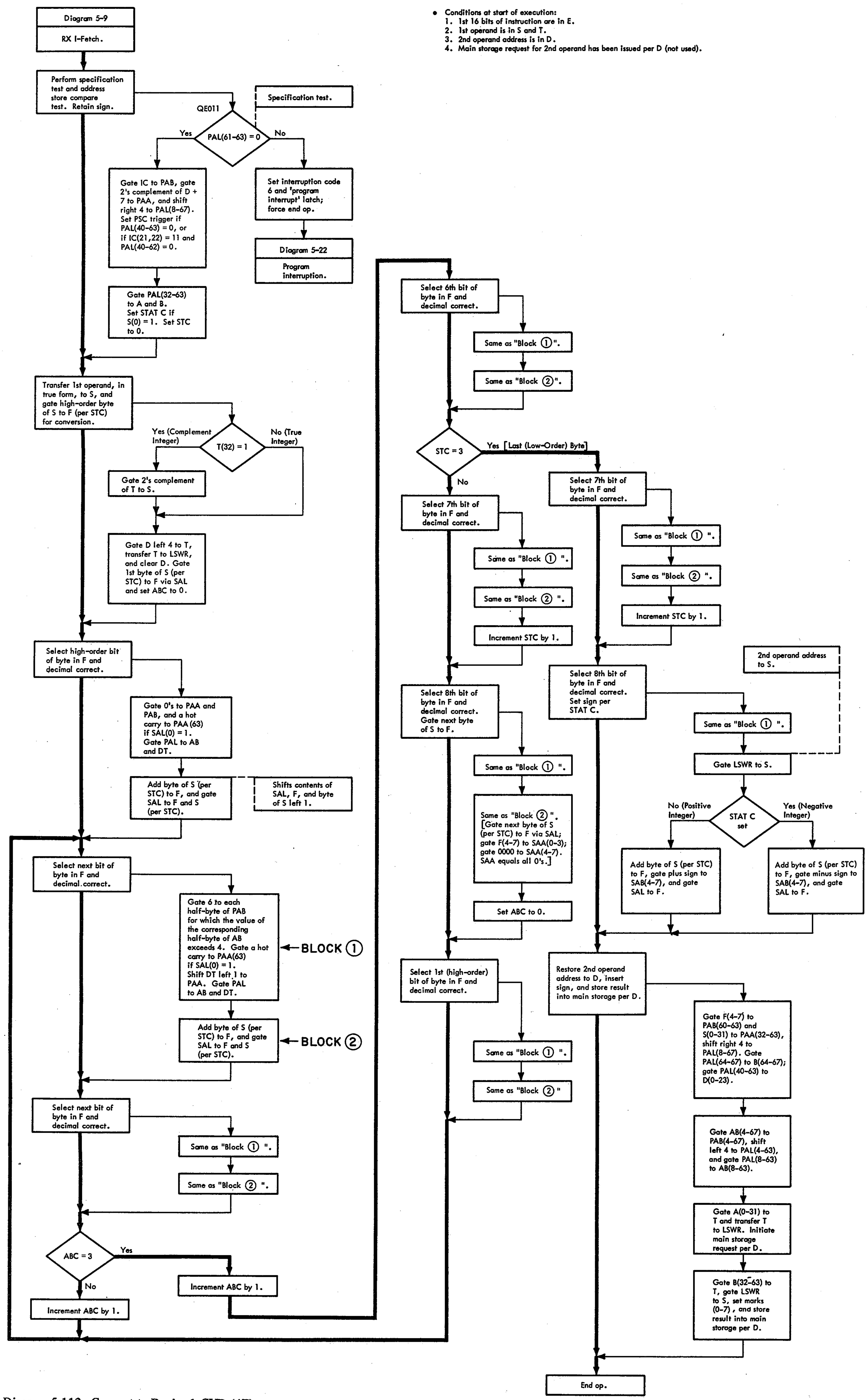


Diagram 5-112. Convert to Decimal, CVD (4E)

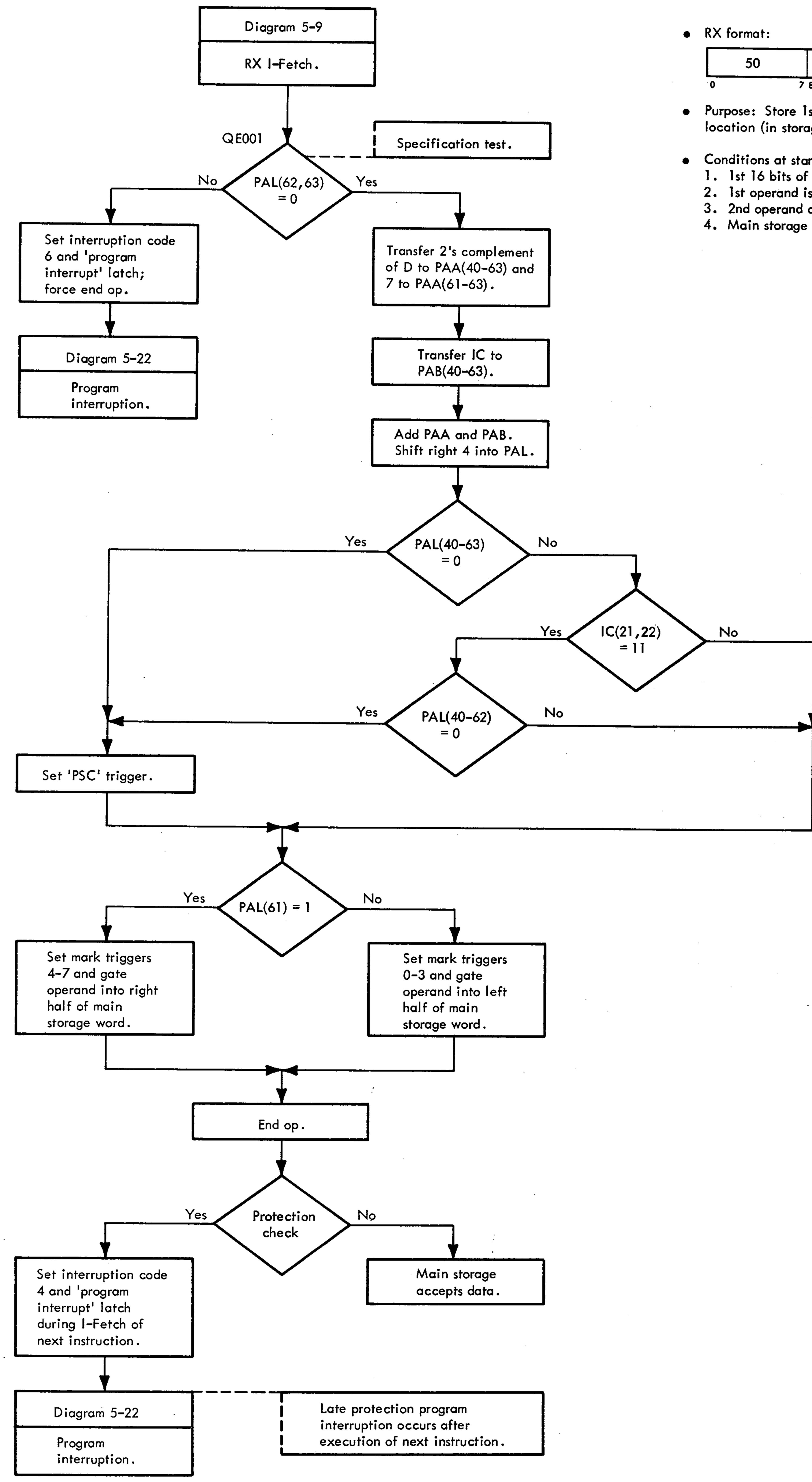


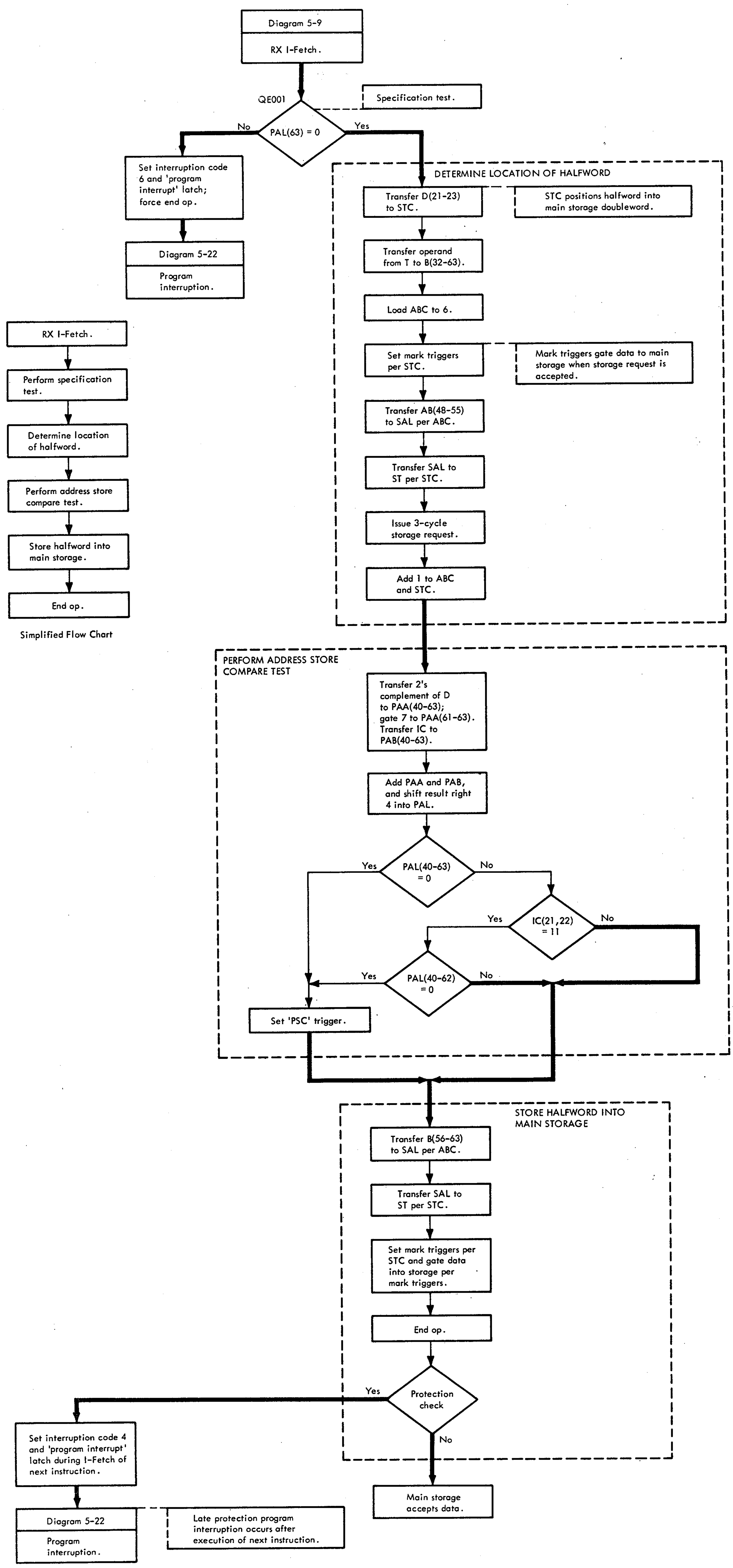
Diagram 5-113. Store, ST (50)

- RX format:

50	R1	X2	B2	D2
0 – 7	8 – 11	12 – 15	16 – 19	20 – 31

- Purpose: Store 1st operand (in GPR, per R1) into 2nd operand location (in storage).

- Conditions at start of execution:
  1. 1st 16 bits of instruction are in E.
  2. 1st operand is in S and T.
  3. 2nd operand address is in D.
  4. Main storage request for 2nd operand has been issued per D (not used).



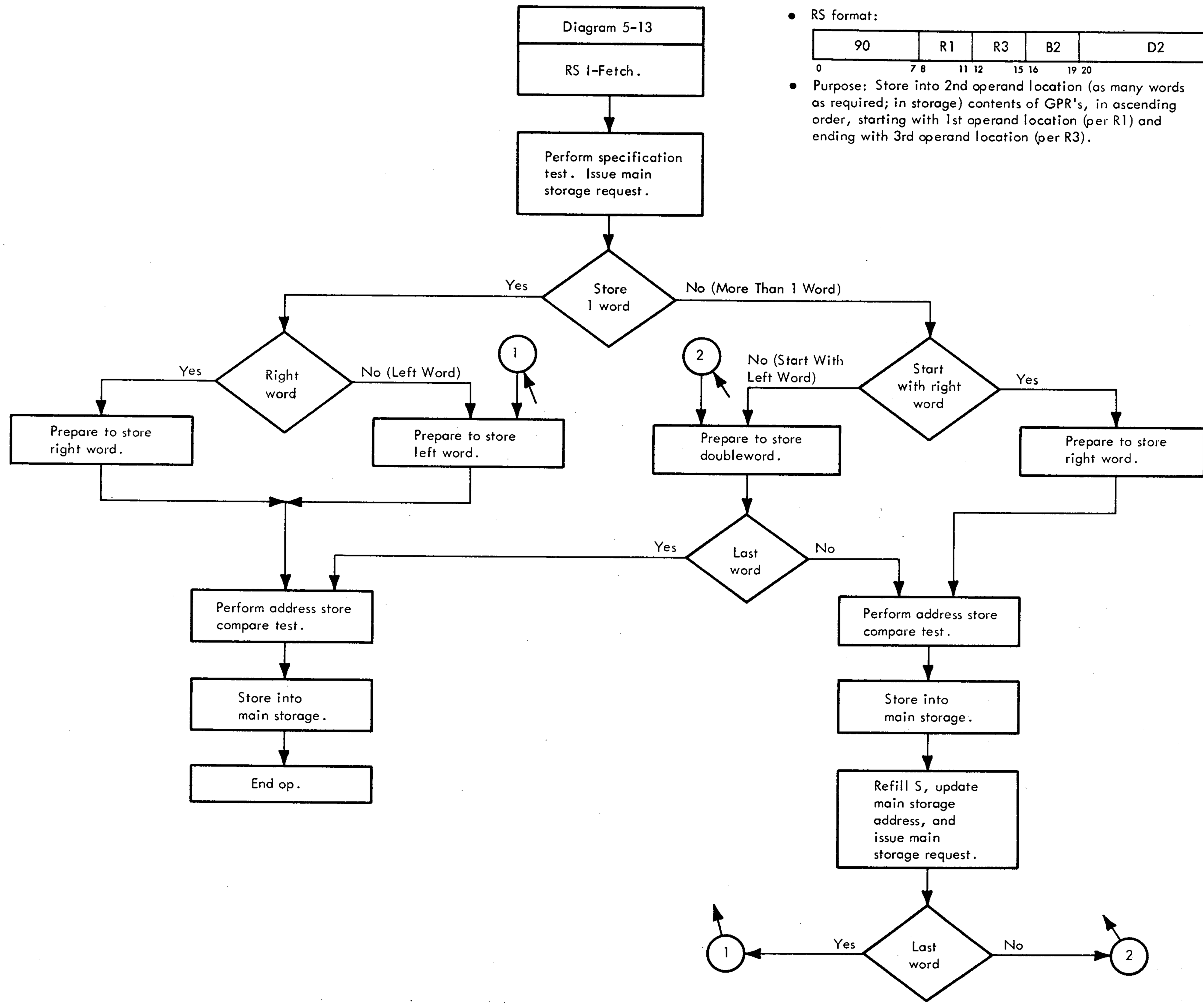
- RX format:

40	R1	X2	B2	D2
0–7	8–11	12–15	16–19	20–31

- Purpose: Store halfword 1st operand (in GPR, per R1) into 2nd operand location (in storage).

- Conditions at start of execution:
  1. 1st 16 bits of instruction are in E.
  2. 1st operand is in S and T.
  3. 2nd operand address is in D.
  4. Main storage request for 2nd operand has been issued per D (not used).

Diagram 5-114. Store Halfword, STH (40)



- RS format:

90	R1	R3	B2	D2
0 7	8 11	12 15	16 19	20 31

- Purpose: Store into 2nd operand location (as many words as required; in storage) contents of GPR's, in ascending order, starting with 1st operand location (per R1) and ending with 3rd operand location (per R3).

Diagram 5-115. Store Multiple, STM (90) (Sheet 1 of 2)

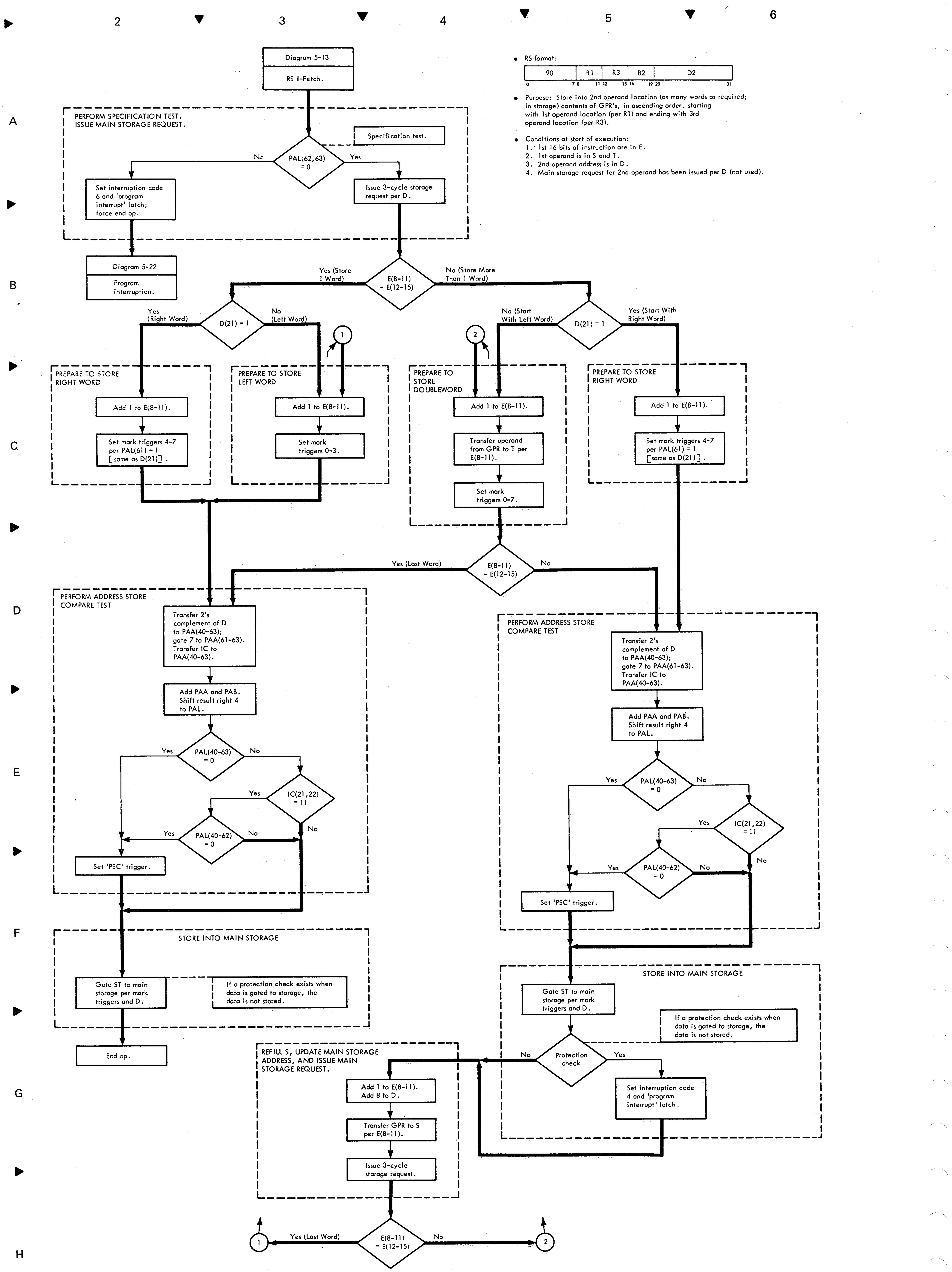
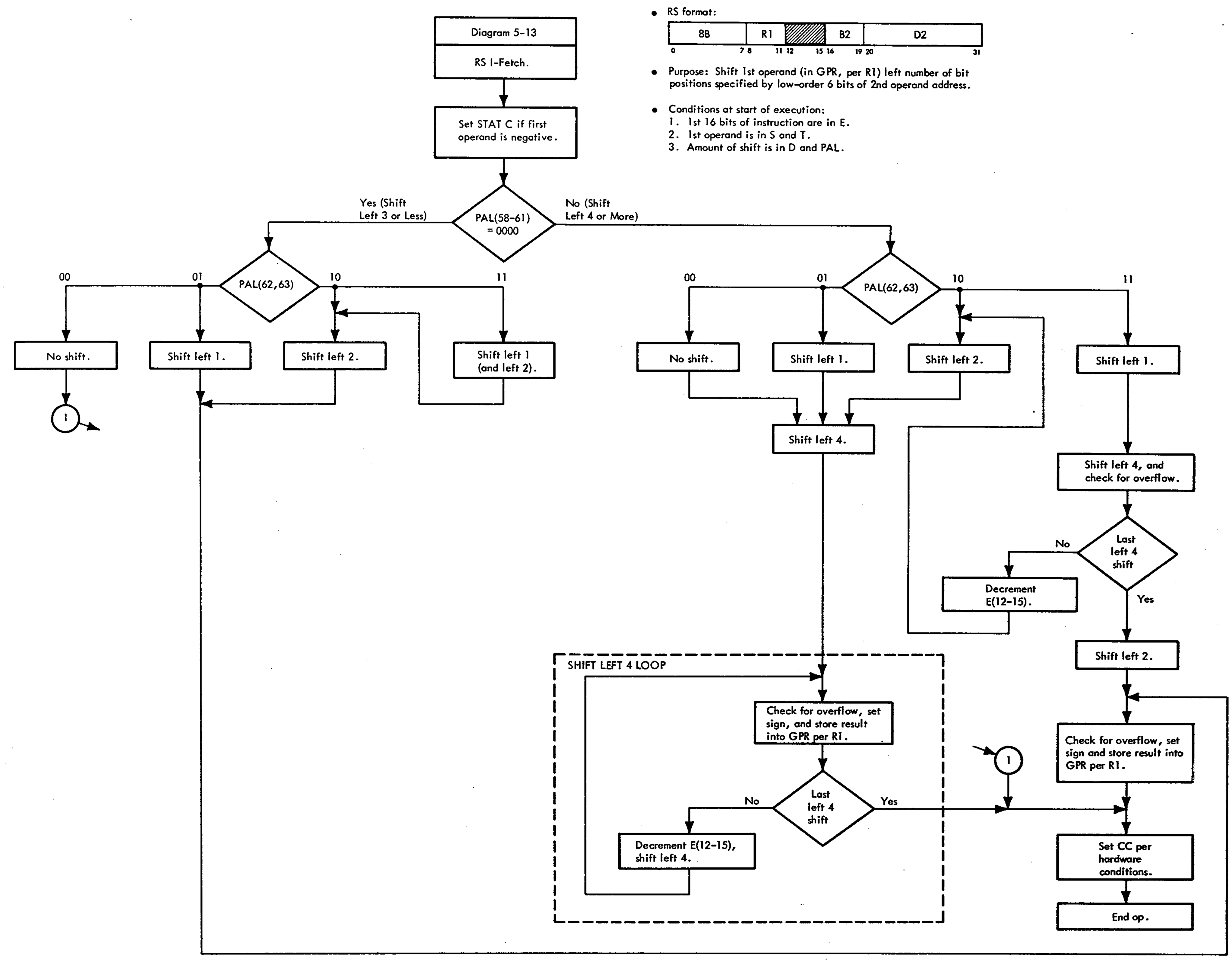
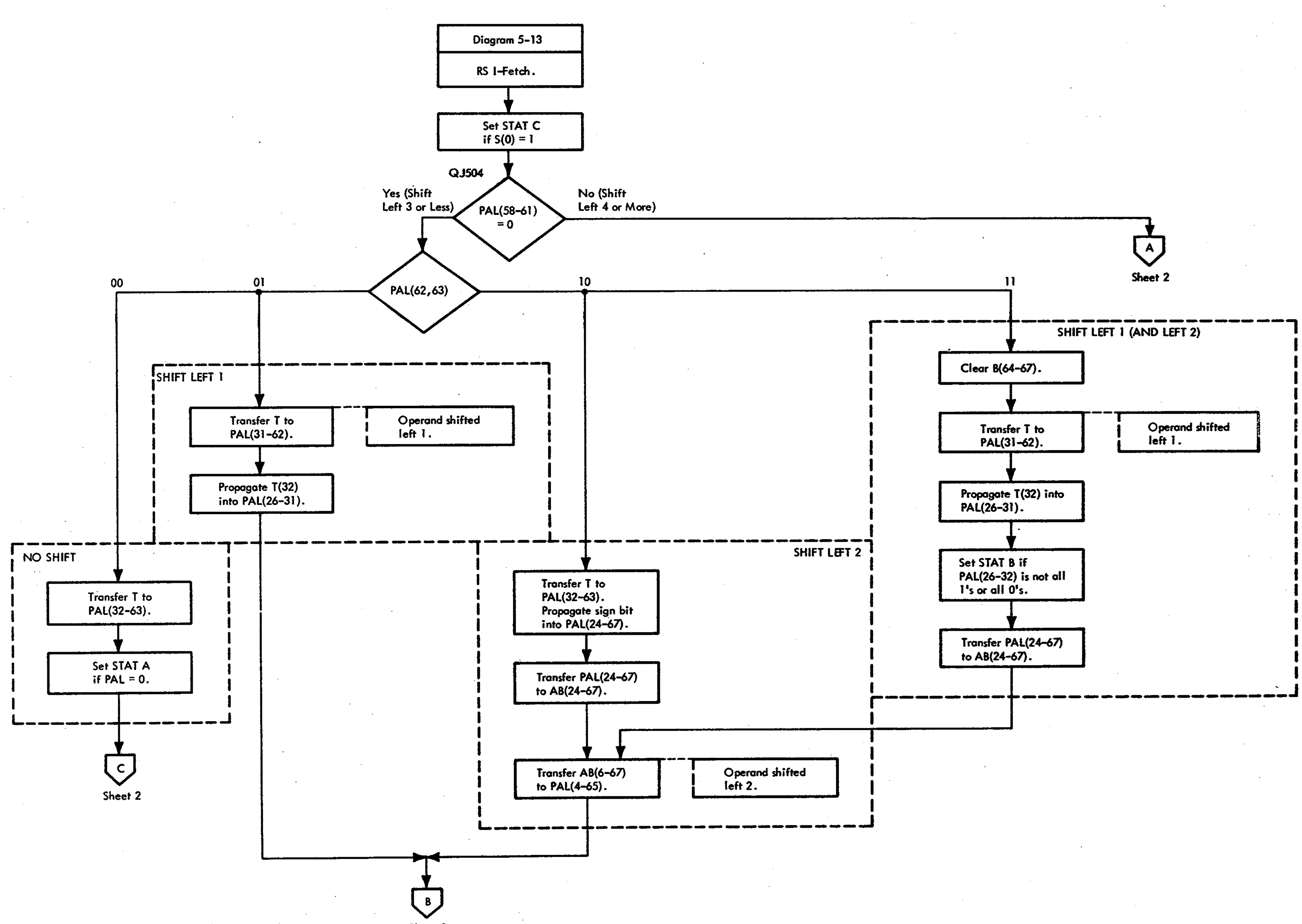


Diagram 5-115. Store Multiple, STM (90) (Sheet 2 of 2)



A) Simplified Flow Chart



B) Detailed Flow Chart (Continued on Sheet 2)

Diagram 5-116. Shift Left Single, SLA (8B) (Sheet 1 of 2)

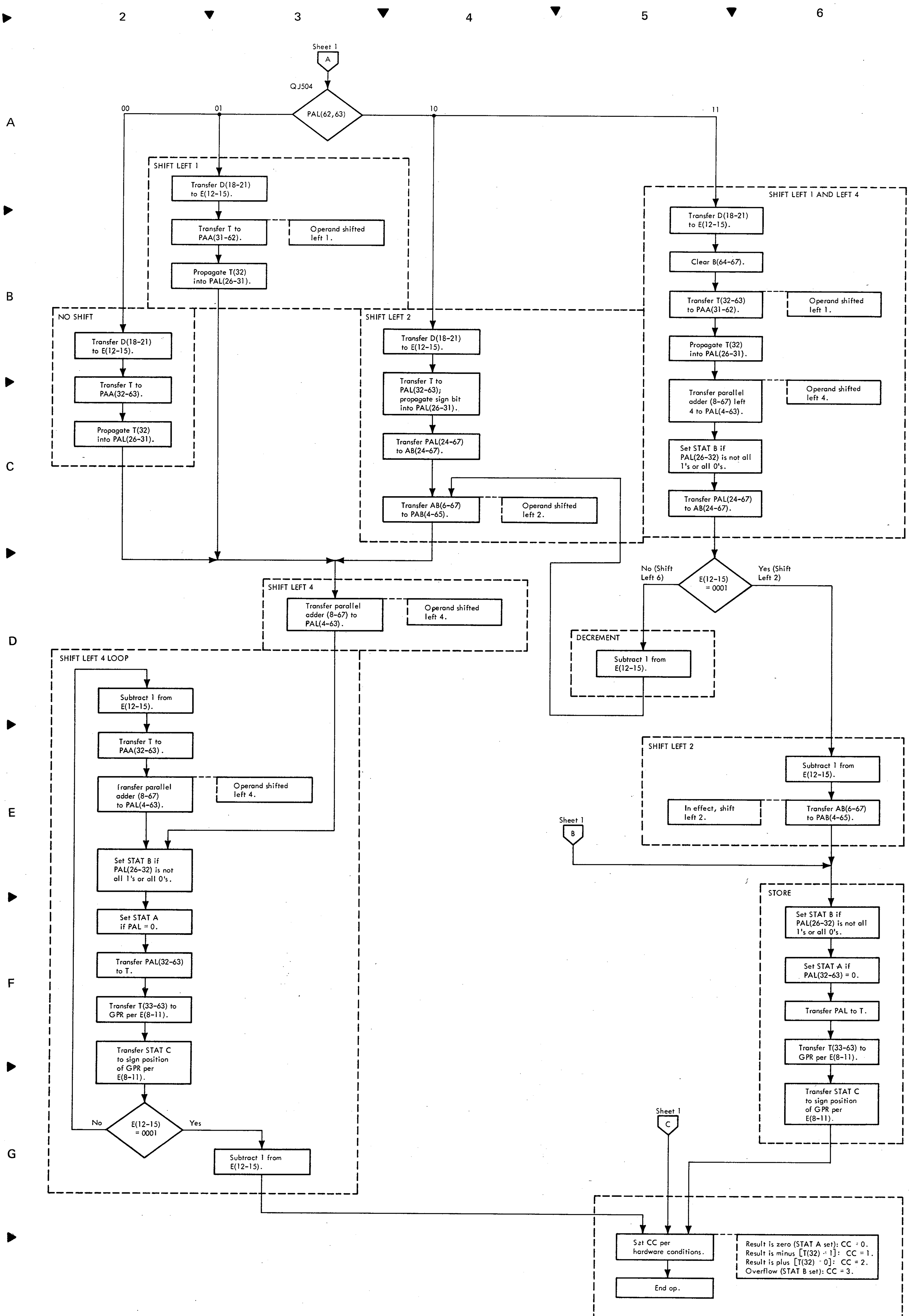


Diagram 5-116. Shift Left Single, SLA (8B) (Sheet 2 of 2)

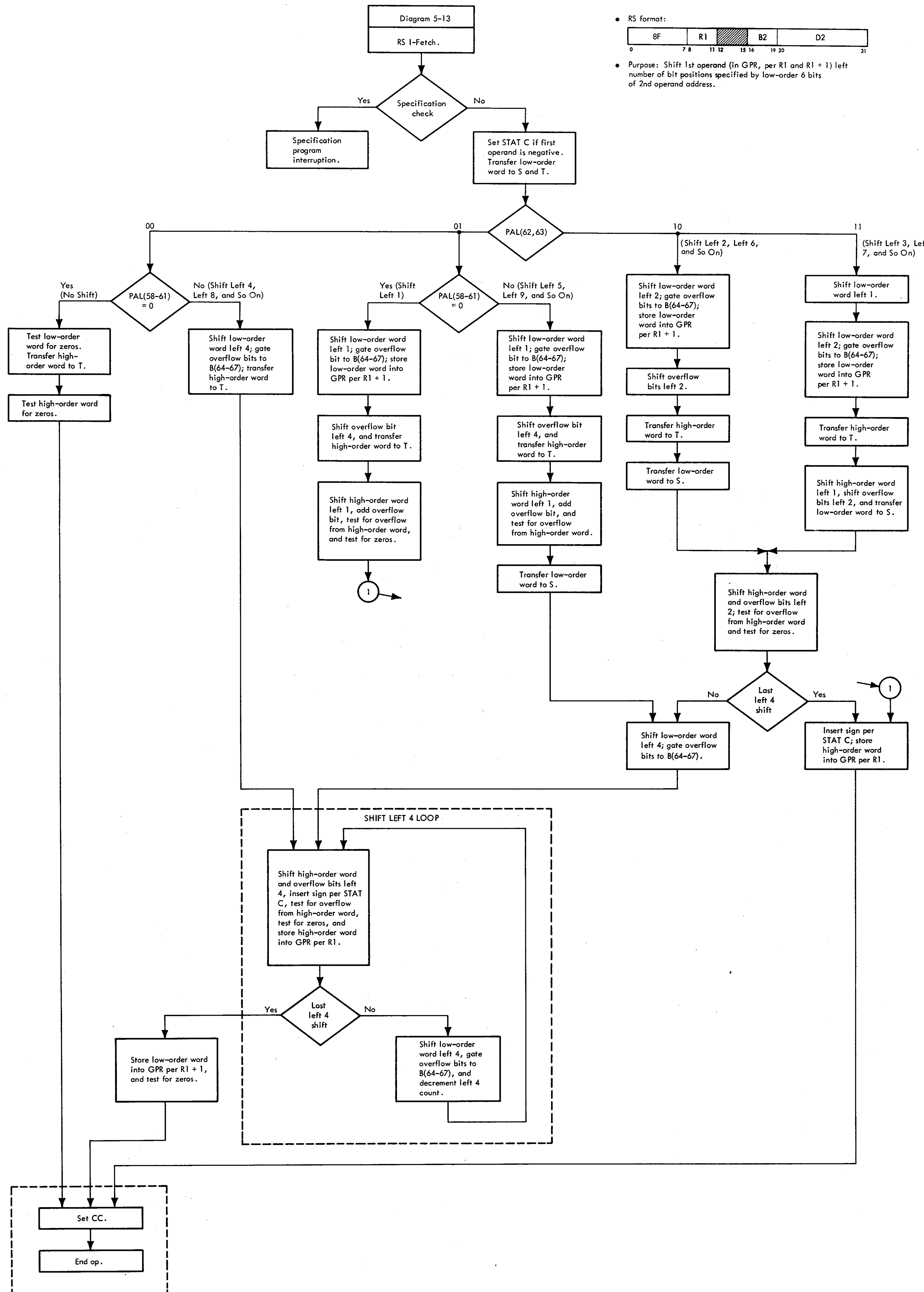


Diagram 5-117. Shift Left Double, SLDA (8F) (Sheet 1 of 4)

- RS format:

8F	R1		B2	D2
0–7	8–11	12–15	16–19	20–31

- Purpose: Shift 1st operand (in GPR, per R1 and R1 + 1) left number of bit positions specified by low-order 6 bits of 2nd operand address.
- Conditions at start of execution:
  1. 1st 16 bits of instruction are in E.
  2. High-order word of 1st operand is in S and T.
  3. Amount of shift is in D and PAL.

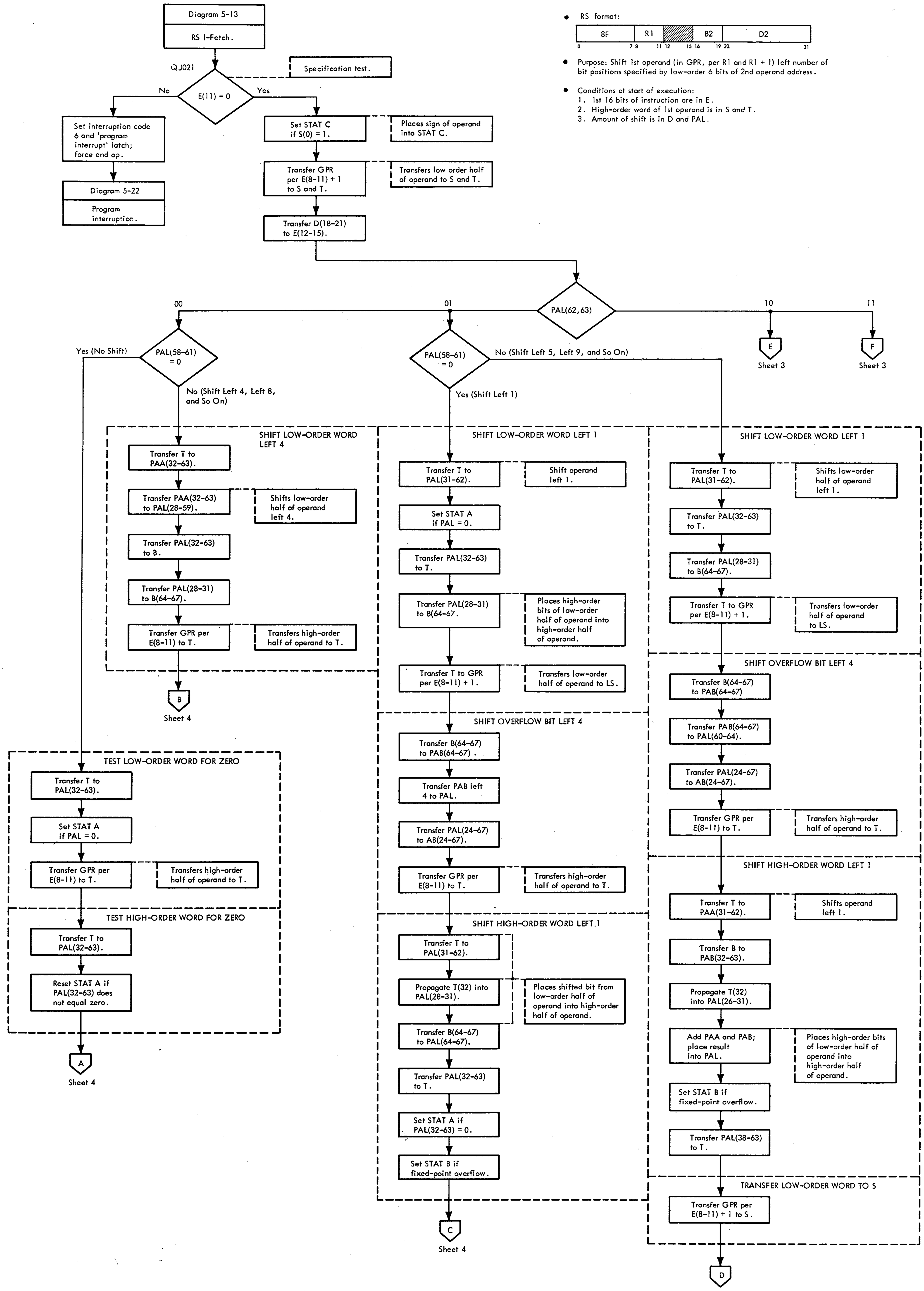


Diagram 5-117. Shift Left Double, SLDA (8F) (Sheet 2 of 4)

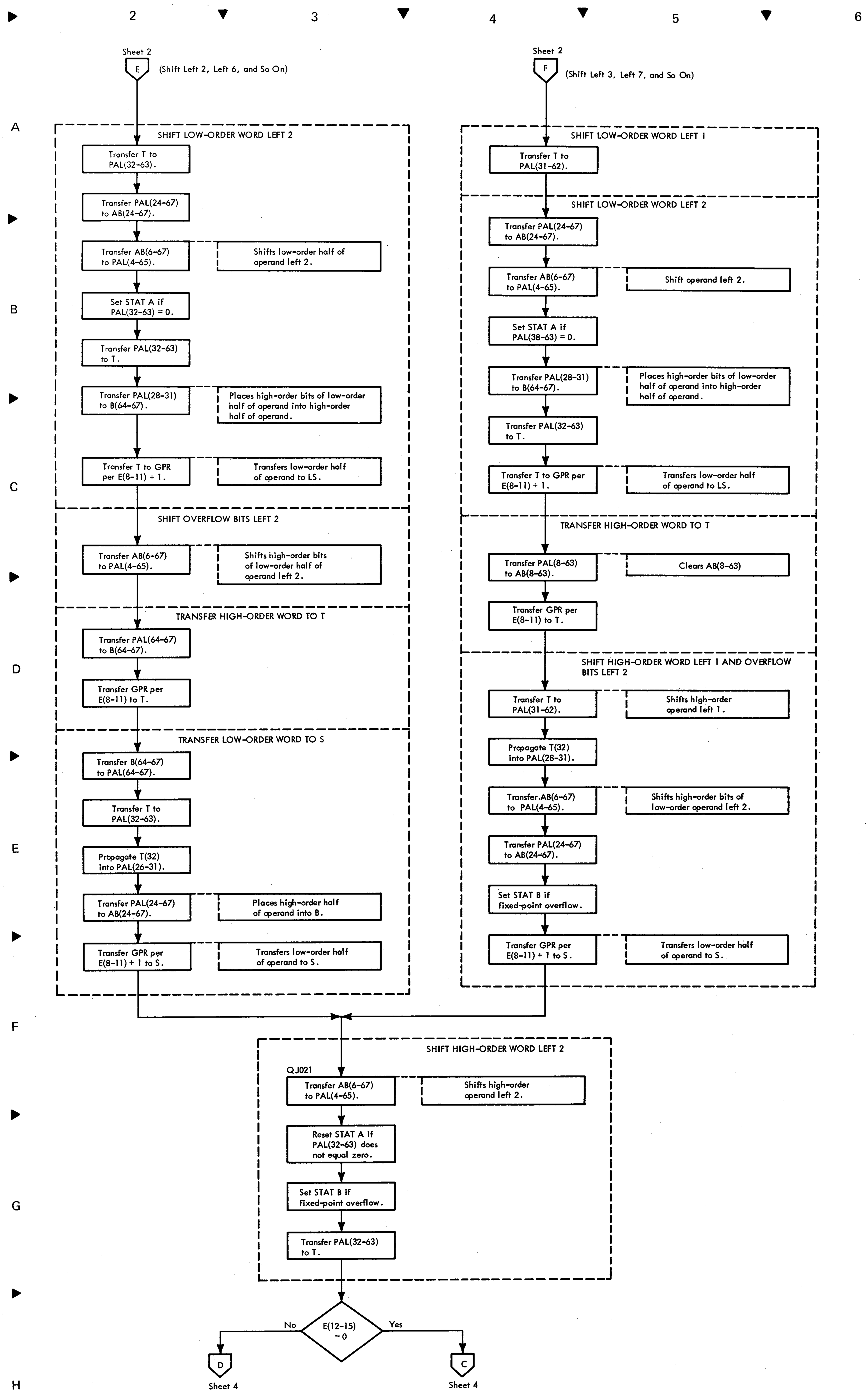


Diagram 5-117. Shift Left Double, SLDA (8F) (Sheet 3 of 4)

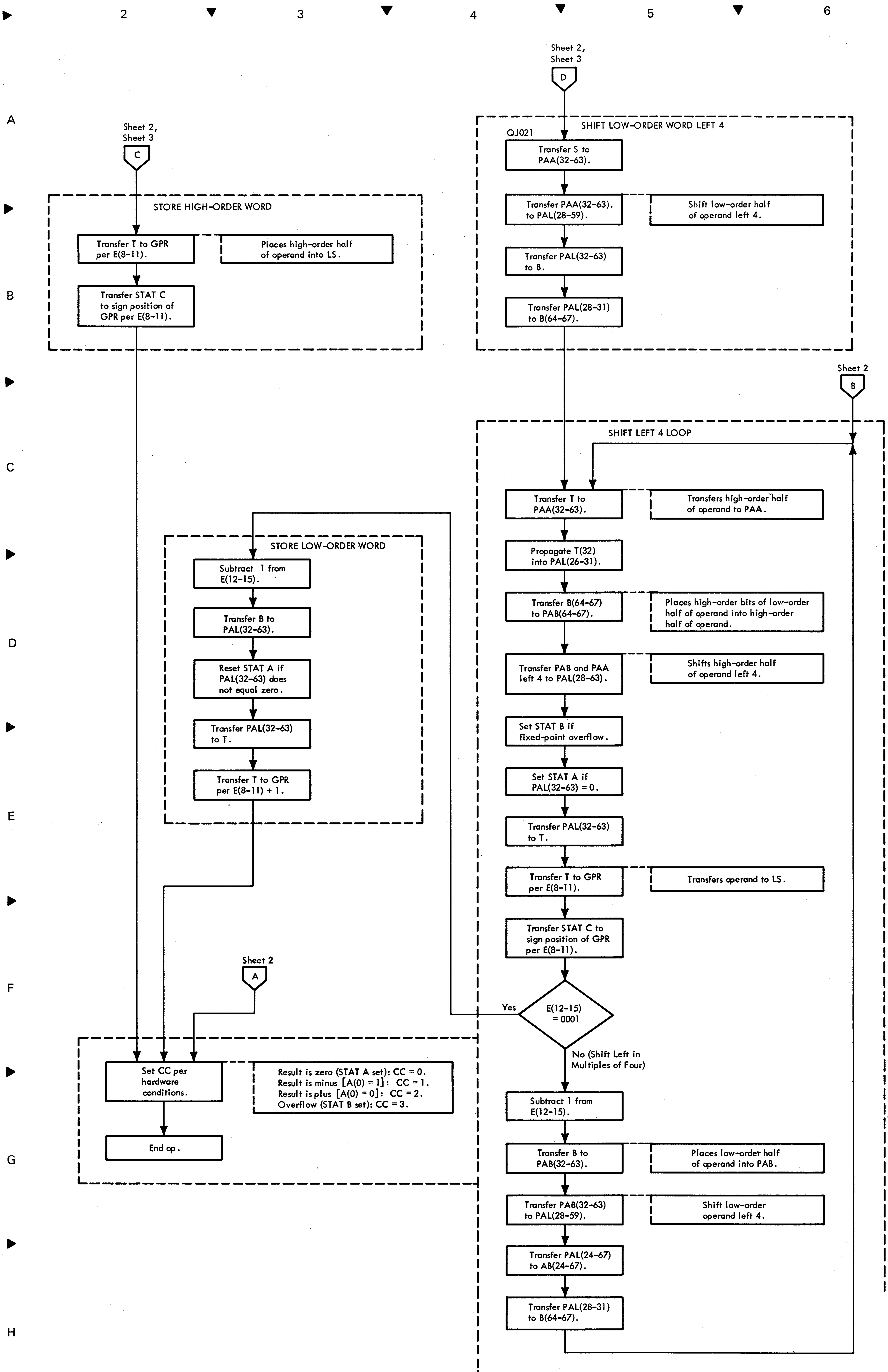


Diagram 5-117. Shift Left Double, SLDA (8F) (Sheet 4 of 4)

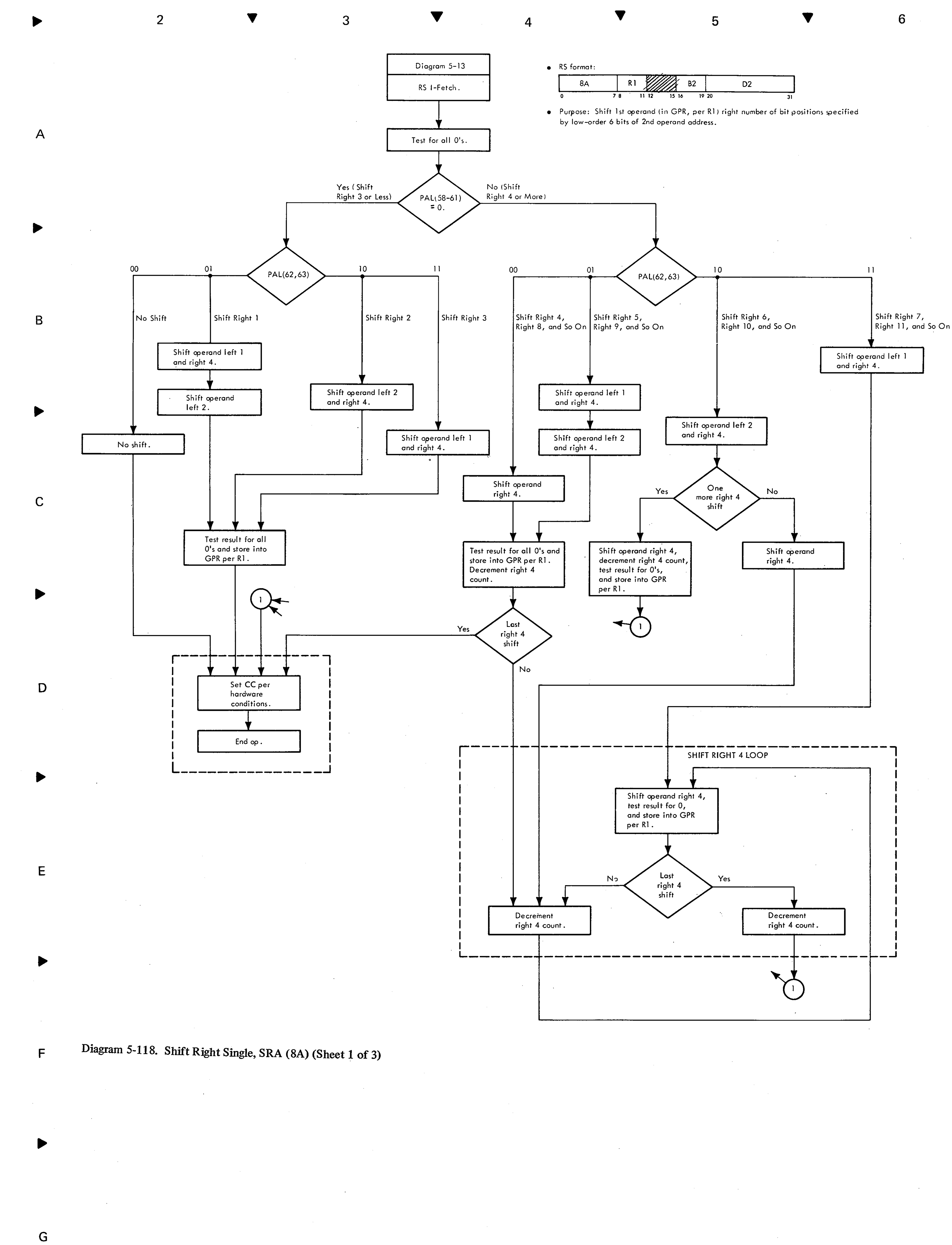


Diagram 5-118. Shift Right Single, SRA (8A) (Sheet 1 of 3)

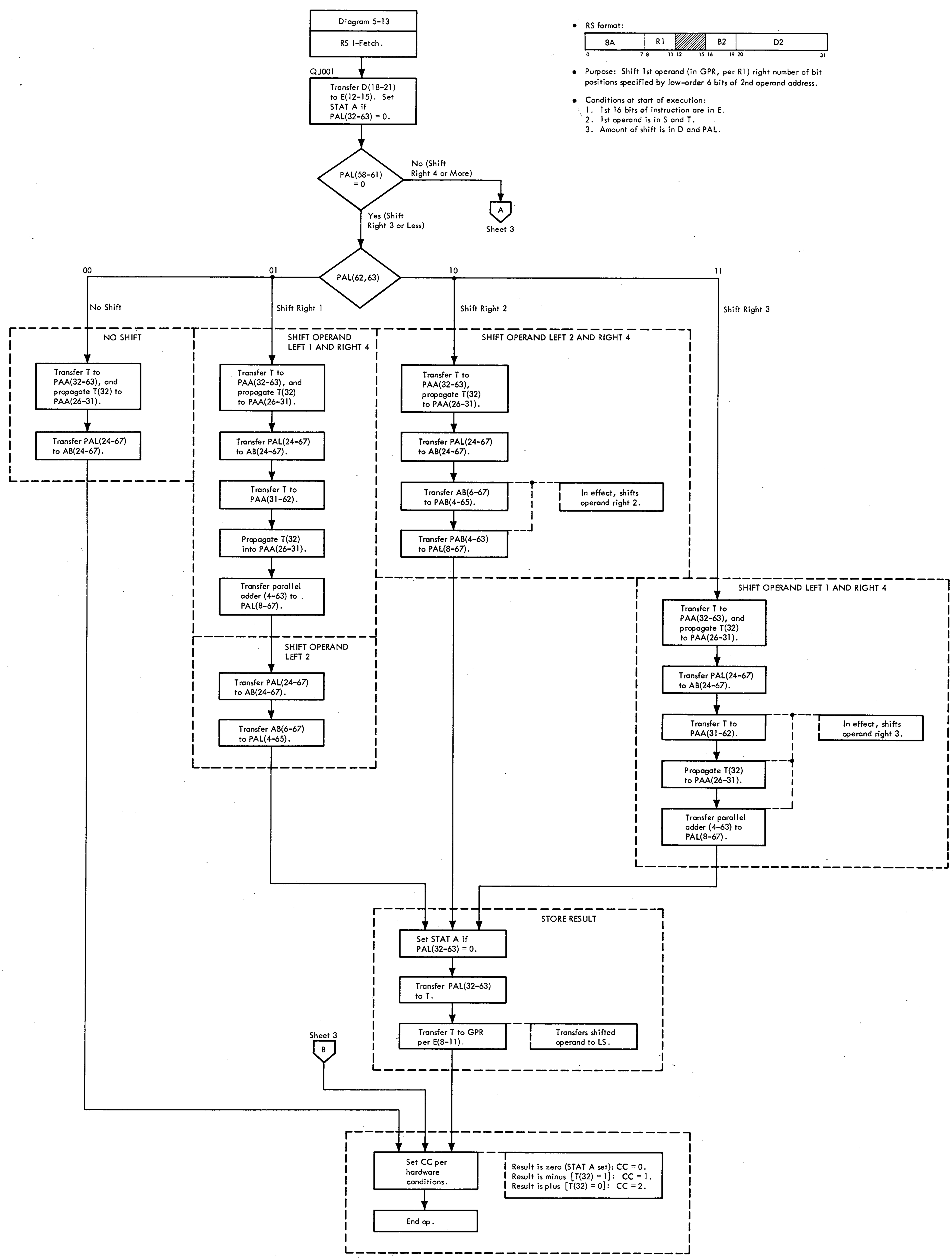


Diagram 5-118. Shift Right Single, SRA (8A) (Sheet 2 of 3)

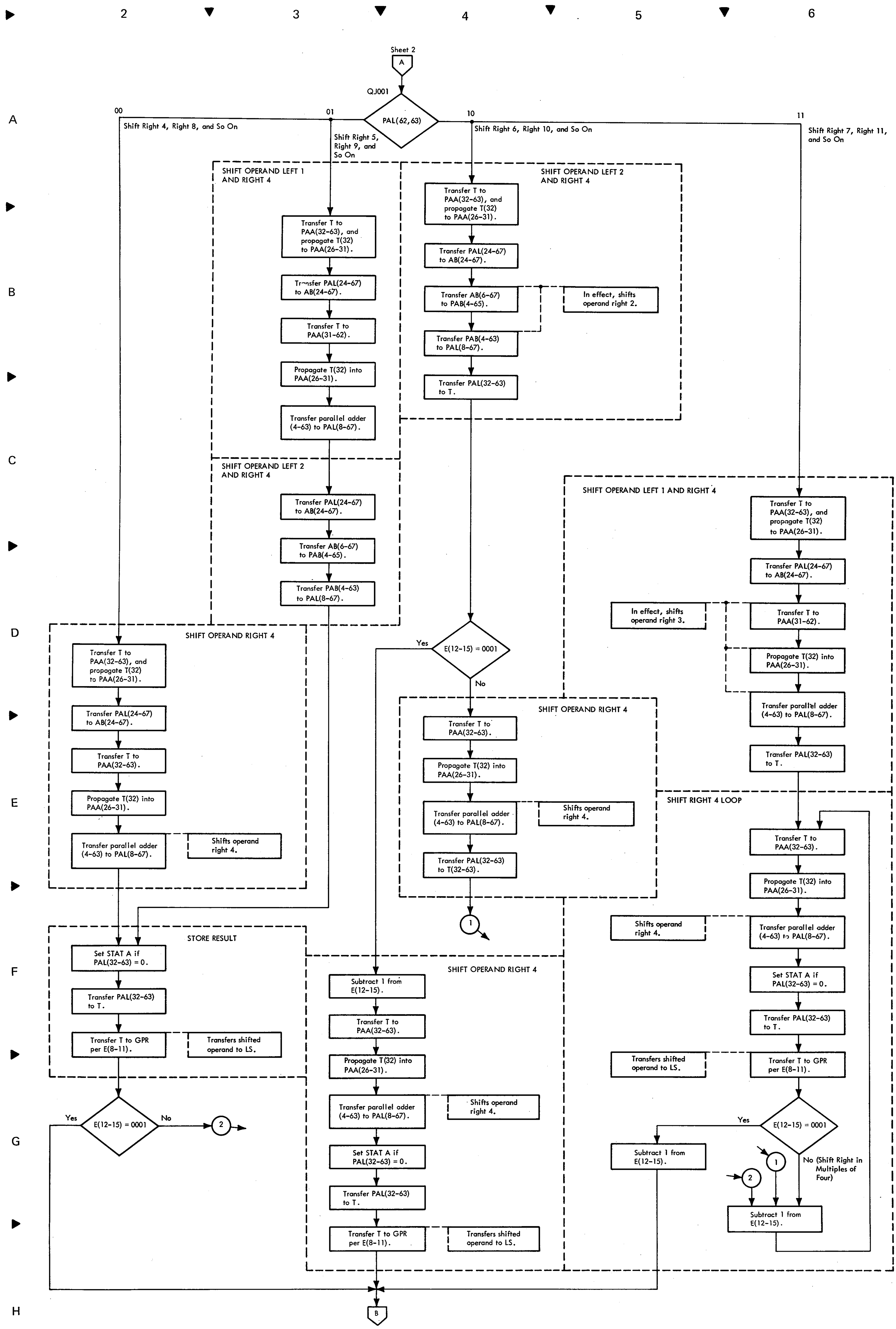


Diagram 5-118. Shift Right Single, SRA (8A) (Sheet 3 of 3)

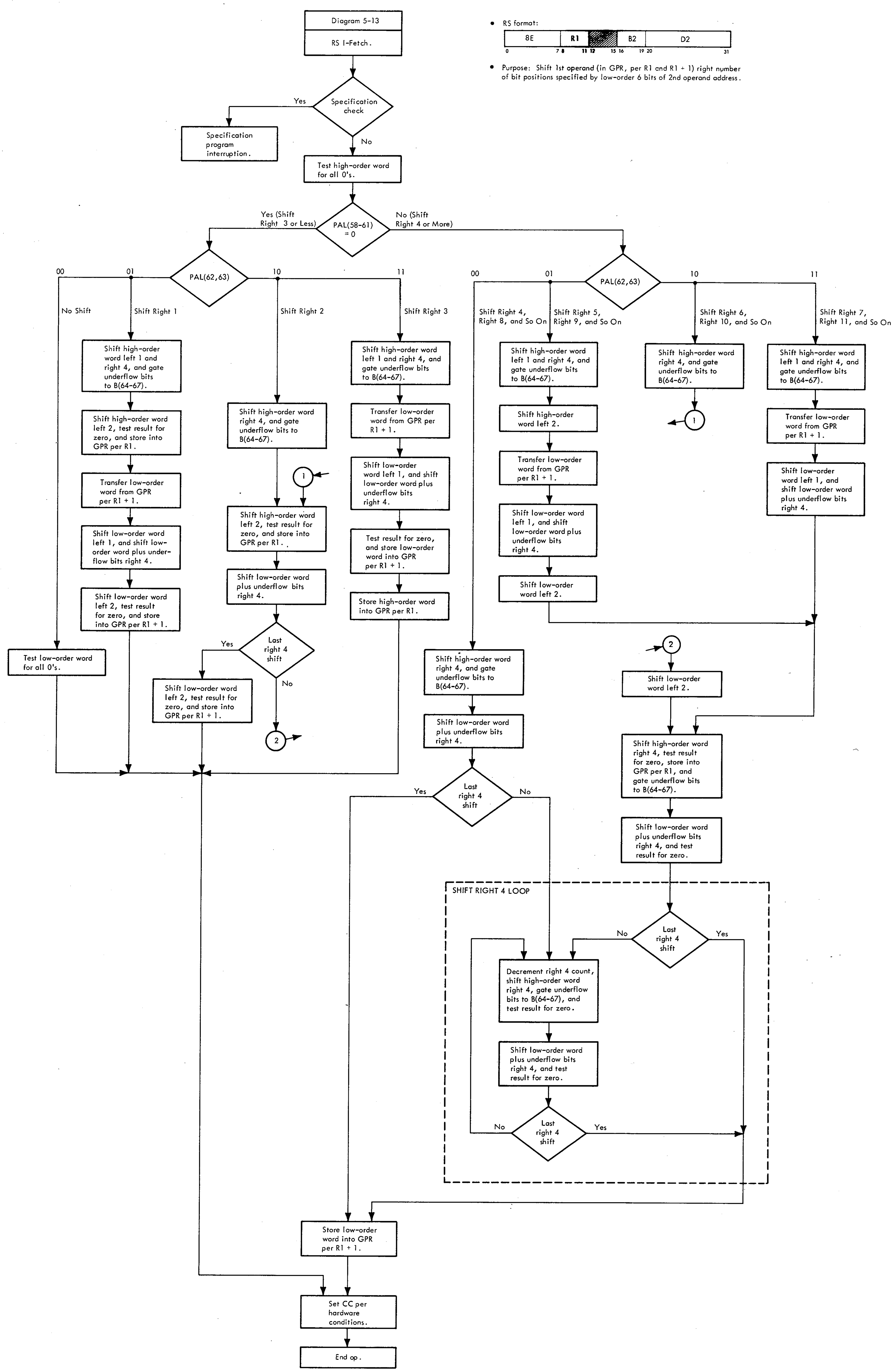


Diagram 5-119. Shift Right Double, SRDA (8E) (Sheet 1 of 4)

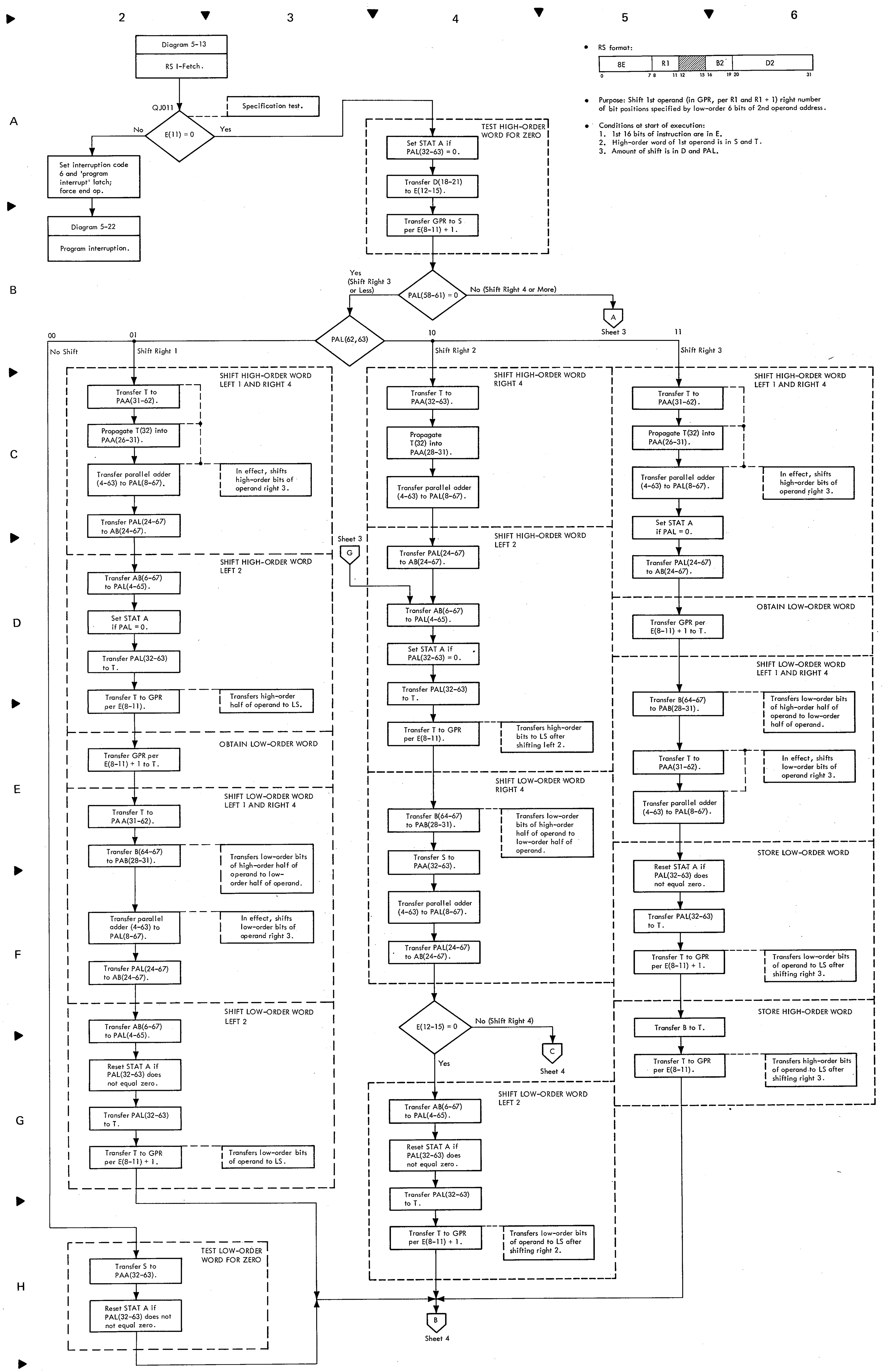


Diagram 5-119. Shift Right Double, SRDA (8E) (Sheet 2 of 4)

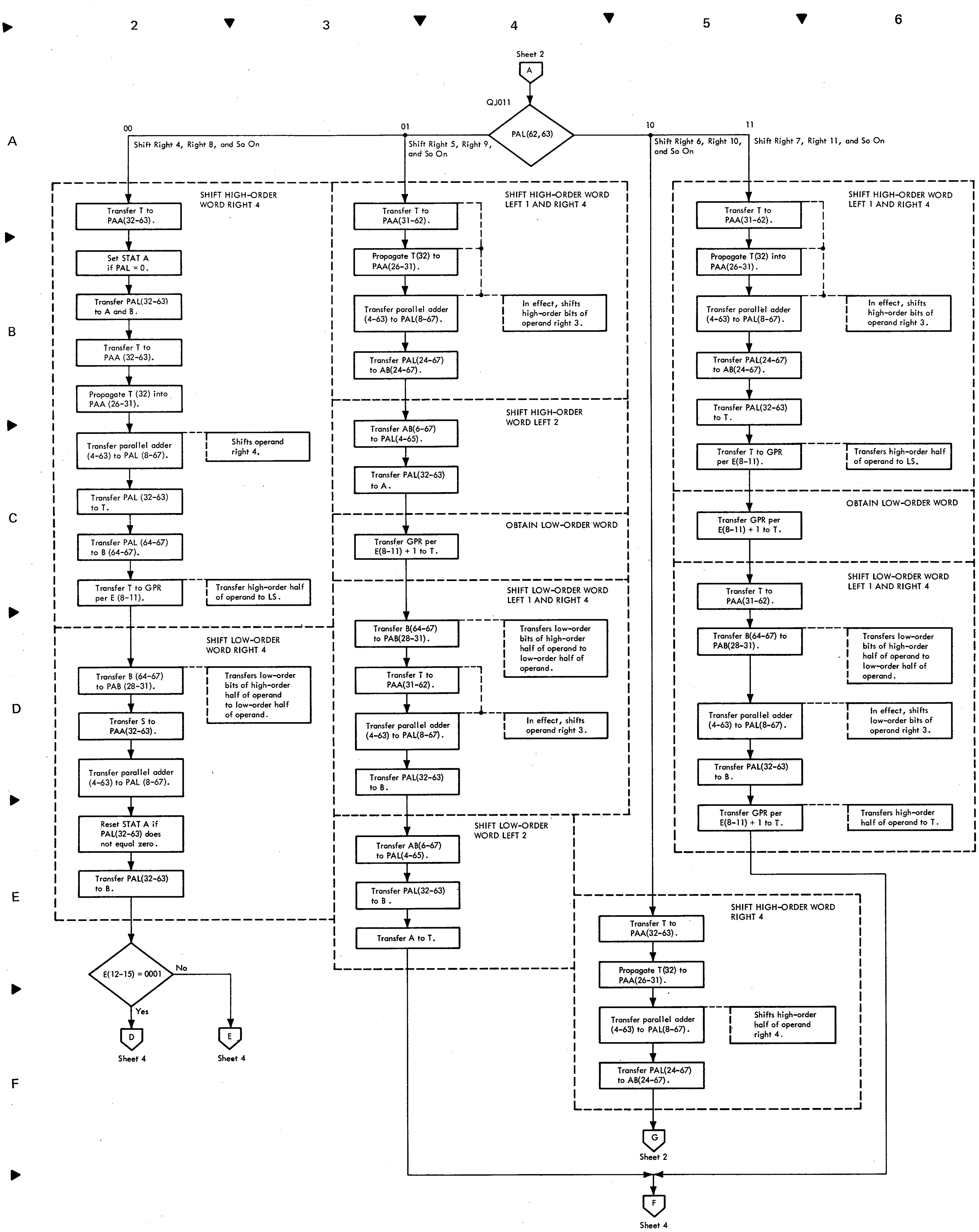


Diagram 5-119. Shift Right Double, SRDA (8E) (Sheet 3 of 4)

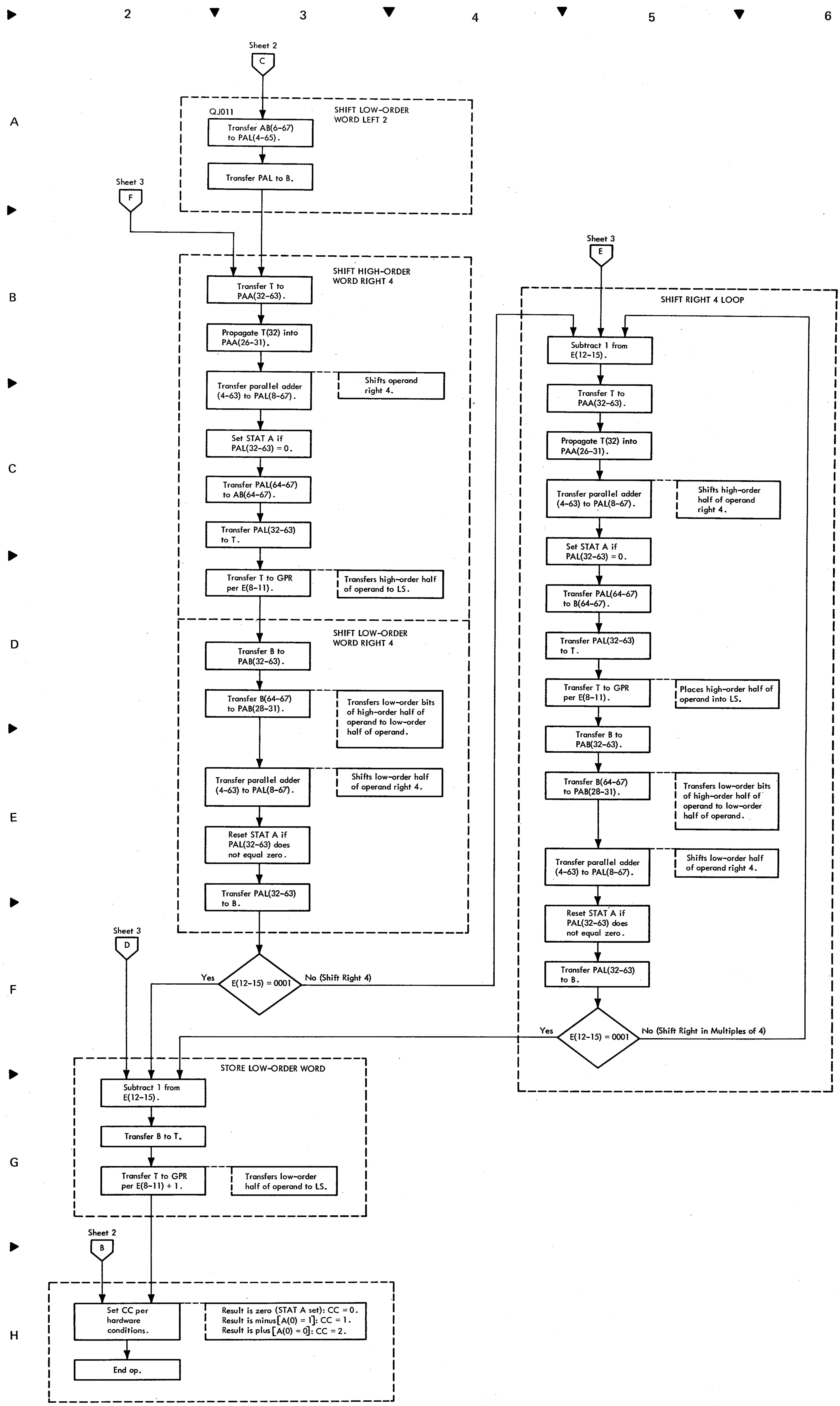


Diagram 5-119. Shift Right Double, SRDA (8E) (Sheet 4 of 4)

DIAGRAM 5-119. SHIFT RIGHT DOUBLE, SRDA (8E) (SHEET 4 OF 4)

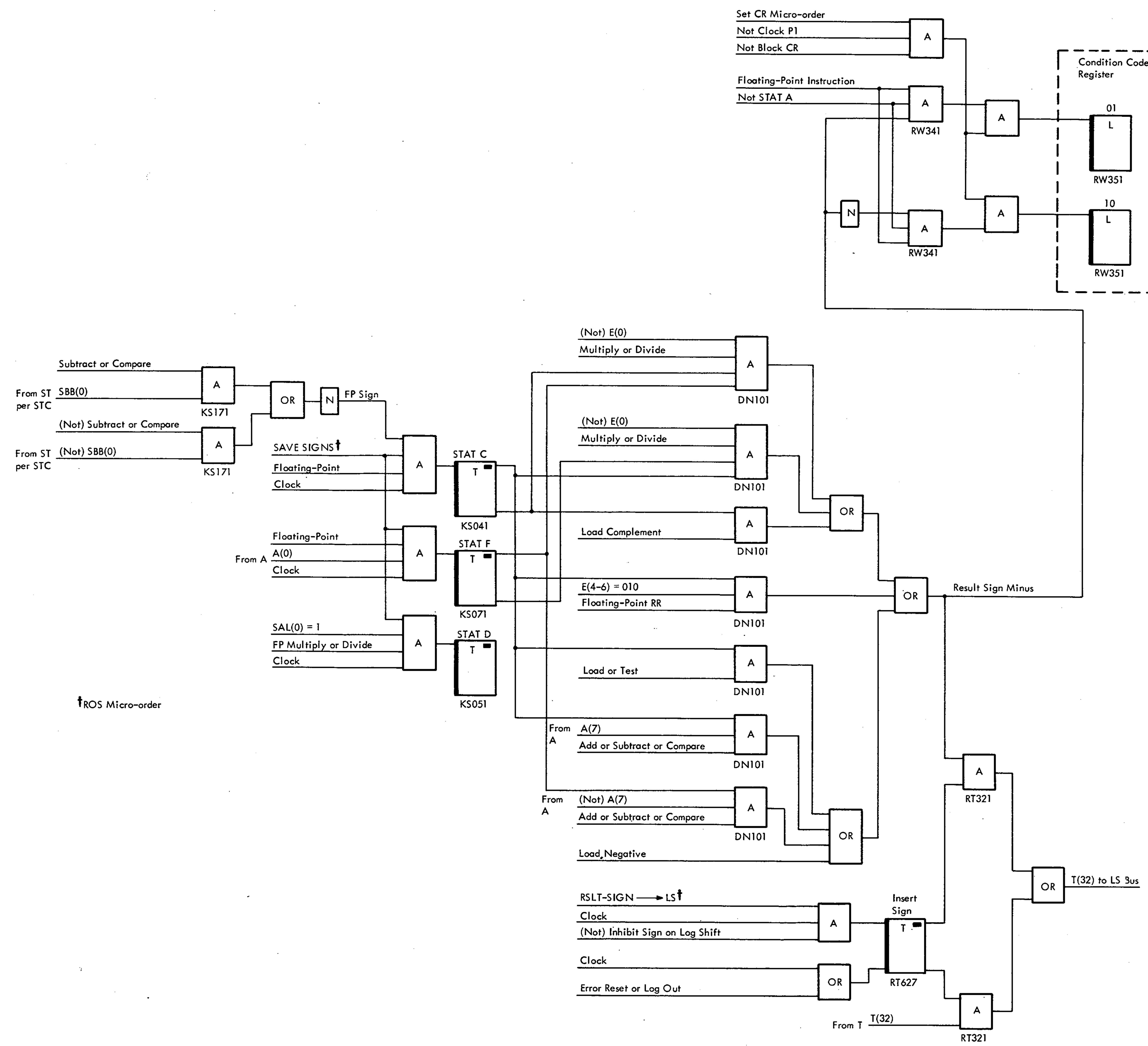
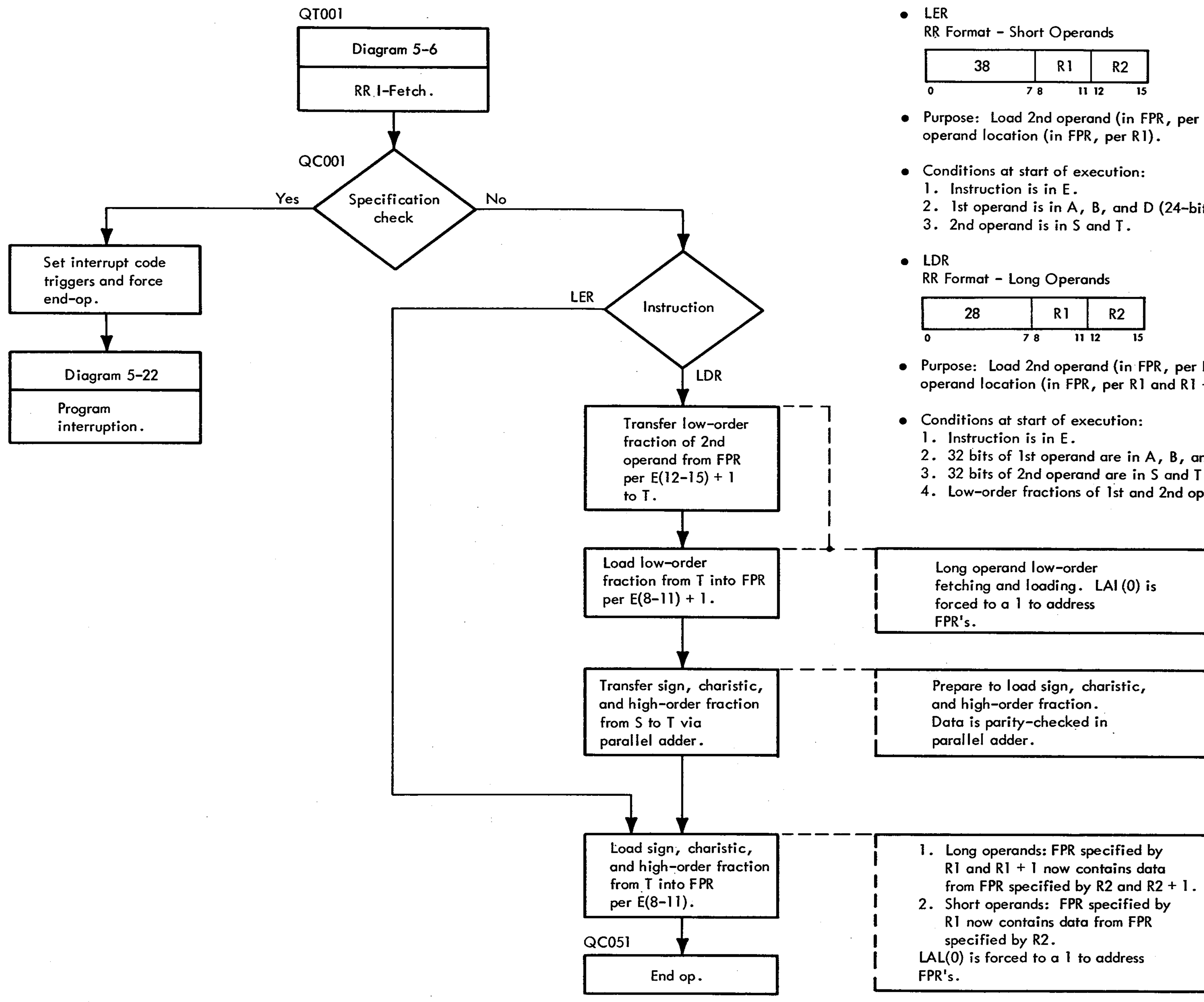


Diagram 5-201. Save Signs and Insert Sign Functions, and CC Setting

2 3 4 5 6



- LER
  RR Format - Short Operands

38	R1	R2
0 7	8 11	12 15

- Purpose: Load 2nd operand (in FPR, per R2) into 1st operand location (in FPR, per R1).
- Conditions at start of execution:
  1. Instruction is in E.
  2. 1st operand is in A, B, and D (24-bit fraction only).
  3. 2nd operand is in S and T.
- LDR
  RR Format - Long Operands

28	R1	R2
0 7	8 11	12 15

- Purpose: Load 2nd operand (in FPR, per R2 and R2 + 1) into 1st operand location (in FPR, per R1 and R1 + 1).
- Conditions at start of execution:
  1. Instruction is in E.
  2. 32 bits of 1st operand are in A, B, and D (24-bit fraction only)
  3. 32 bits of 2nd operand are in S and T.
  4. Low-order fractions of 1st and 2nd operands are in LS.

Diagram 5-202. Load, LER (38) - Short Operands; Load, LDR (28) - Long Operands

A B C D E F G H

- LE
  RX Format - Short Operands

78	R1	X2	B2	D2
0 7	8 11	12 15	16 19	20 31

- Purpose: Load 2nd operand (in storage) into 1st operand location (in FPR, per R1).
- Conditions at start of execution:
  1. 1st 16 bits of instruction are in E.
  2. 1st operand is in S and T.
  3. Main storage request for 2nd operand has been issued per D.

- LD
  RX Format - Long Operands

68	R1	X2	B2	D2
0 7	8 11	12 15	16 19	20 31

- Purpose: Load 2nd operand (in storage) into 1st operand location (in FPR, per R1 and R1 + 1).
- Conditions at start of execution:
  1. 1st 16 bits of instruction are in E.
  2. 32 bits of 1st operand are in S and T.
  3. Low-order fraction of 1st operand is in LS.
  4. Main storage request for 2nd operand has been issued per D.

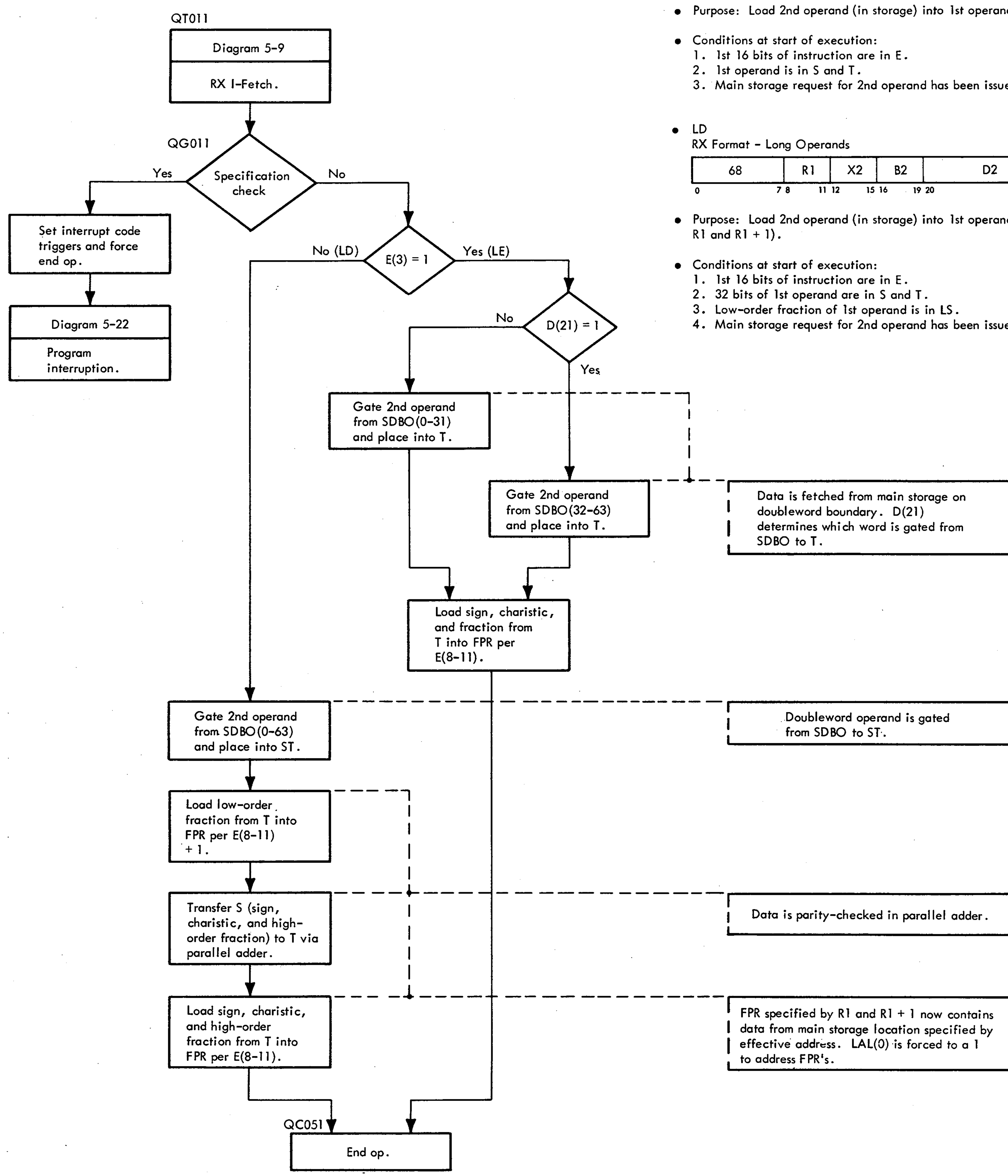


Diagram 5-203. Load, LE (78) - Short Operands; Load, LD (68) - Long Operands

- RR Format - Short Operands

Op Code	R1	R2
0 – 7	8 – 11	12 – 15

- Purpose: Load 2nd operand (in FPR, per R2) into 1st operand location (in FPR, per R1): LTER, with original sign; LCER, with sign complemented; LNER, with sign made minus; LPER, with sign made plus.

- Conditions at start of execution:
  1. Instruction is in E.
  2. 1st operand is in A, B, and D (24-bit fraction only).
  3. 2nd operand is in S and T.

- Op codes:
  1. LCER = 33.
  2. LNER = 31.
  3. LPER = 30.
  4. LTER = 32.

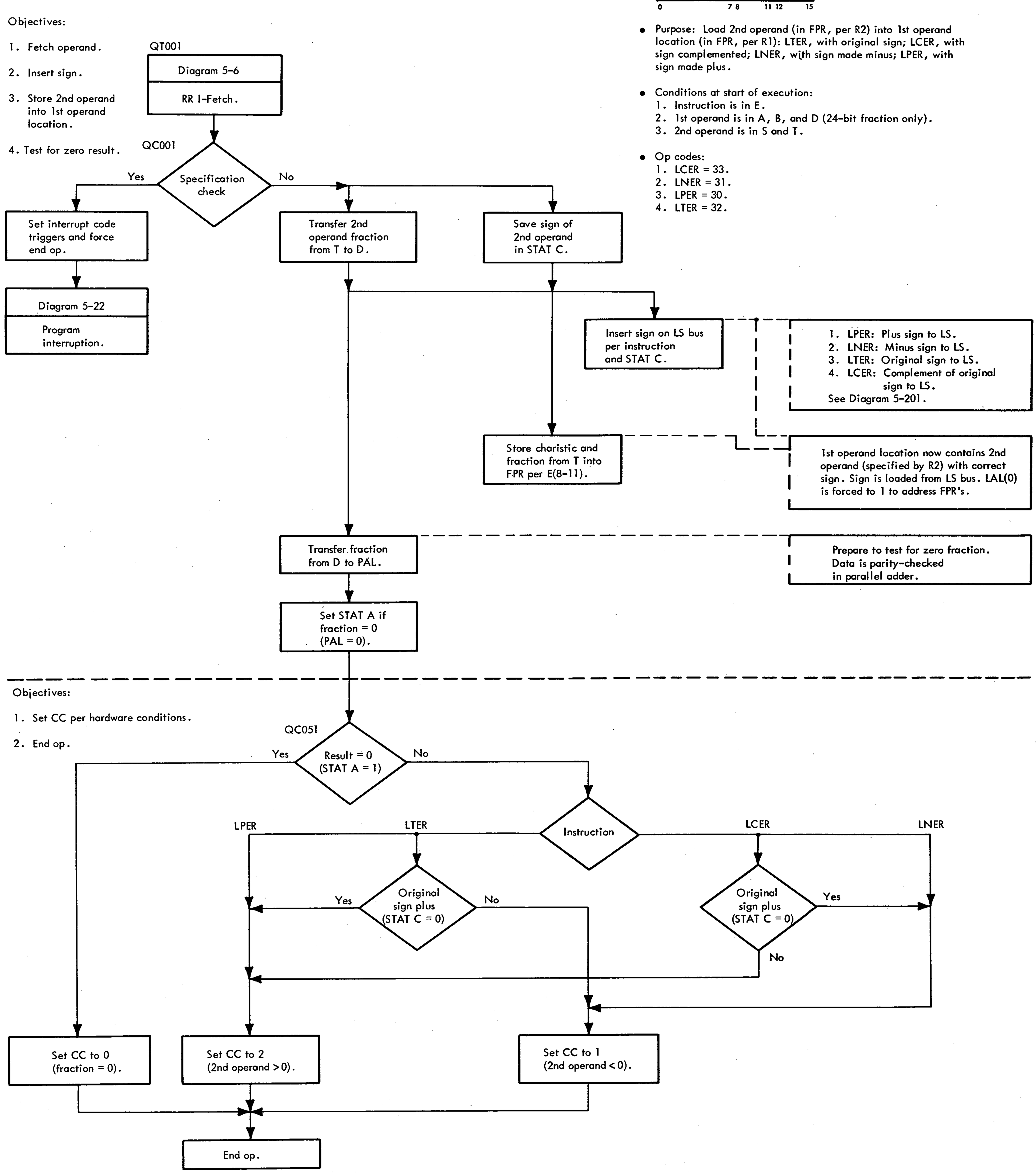


Diagram 5-204. Load Positive, LPER (30); Load Negative, LNER (31); Load and Test, LTER (32); Load Complement, LCER (33) - Short Operand

- RR Format - Long Operands

Op Code	R1	R2
0 7	8 11	12 15

- Purpose: Load 2nd operand (in FPR, per R2 and R2 + 1) into 1st operand location (in FPR, per R1 and R1 + 1); LTDR, with original sign; LCDR, with sign complemented; LNDR, with sign made minus; LPDR, with sign made plus.

- Conditions at start of execution:
  1. Instruction is in E.
  2. 32 bits of 1st operand are in A, B, and D (24-bit fraction only).
  3. 32 bits of 2nd operand are in S and T.
  4. Low-order fractions of 1st and 2nd operands are in LS.

- Op codes:
  1. LCDR = 23.
  2. LNDR = 21.
  3. LPDR = 20.
  4. LTDR = 22.

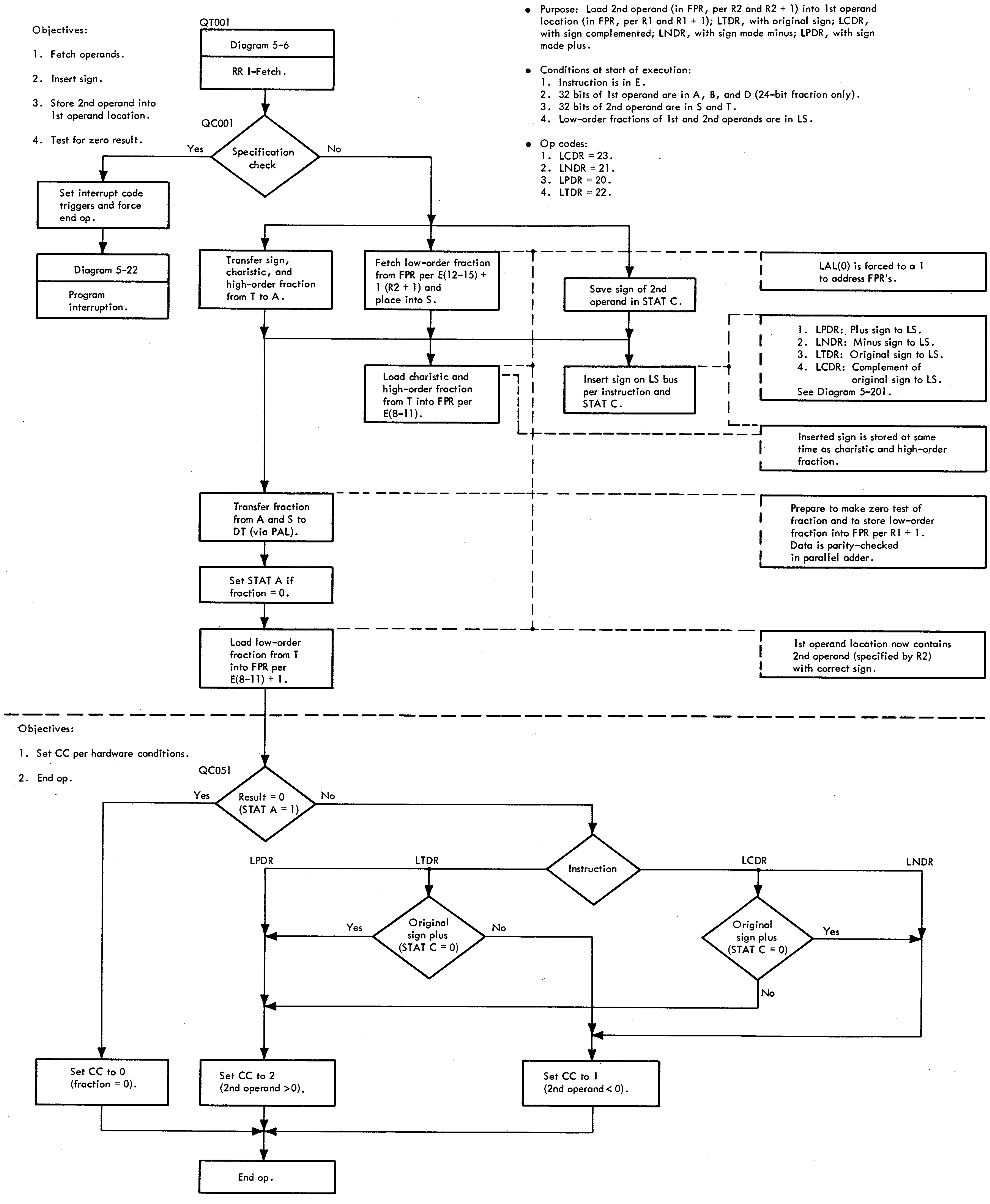


Diagram 5-205. Load Positive, LPDR (20); Load Negative, LNDR (21); Load and Test, LTDR (22); Load Complement, LCDR (23) - Long Operands

- RR instructions: AER, AUR, SER, SUR, CER.

Op Code	R1	R2
0–7	8–11	12–15

- Purpose:
  1. AER, AUR, SER, SUR - Algebraically add (subtract) 2nd operand (in FPR, per R2) to (from) 1st operand (in FPR, per R1) and place normalized result (unnormalized result for AUR and SUR) into 1st operand location.
  2. CER - Algebraically compare 1st operand (in FPR, per R1) with 2nd operand (in FPR, per R2); CC indicates results.

- RX instructions: AE, AU, SE, SU, CE.

Op Code	R1	X2	B2	D2
0–7	8–11	12–15	16–19	20–31

- Purpose:
  1. AE, AU, SE, SU - Algebraically add (subtract) 2nd operand (in storage) to (from) 1st operand (in FPR, per R1) and place normalized result (unnormalized result for AU and SU) into 1st operand location.
  2. CE - Algebraically compare 1st operand (in FPR, per R1) with 2nd operand (in storage); CC indicates result.

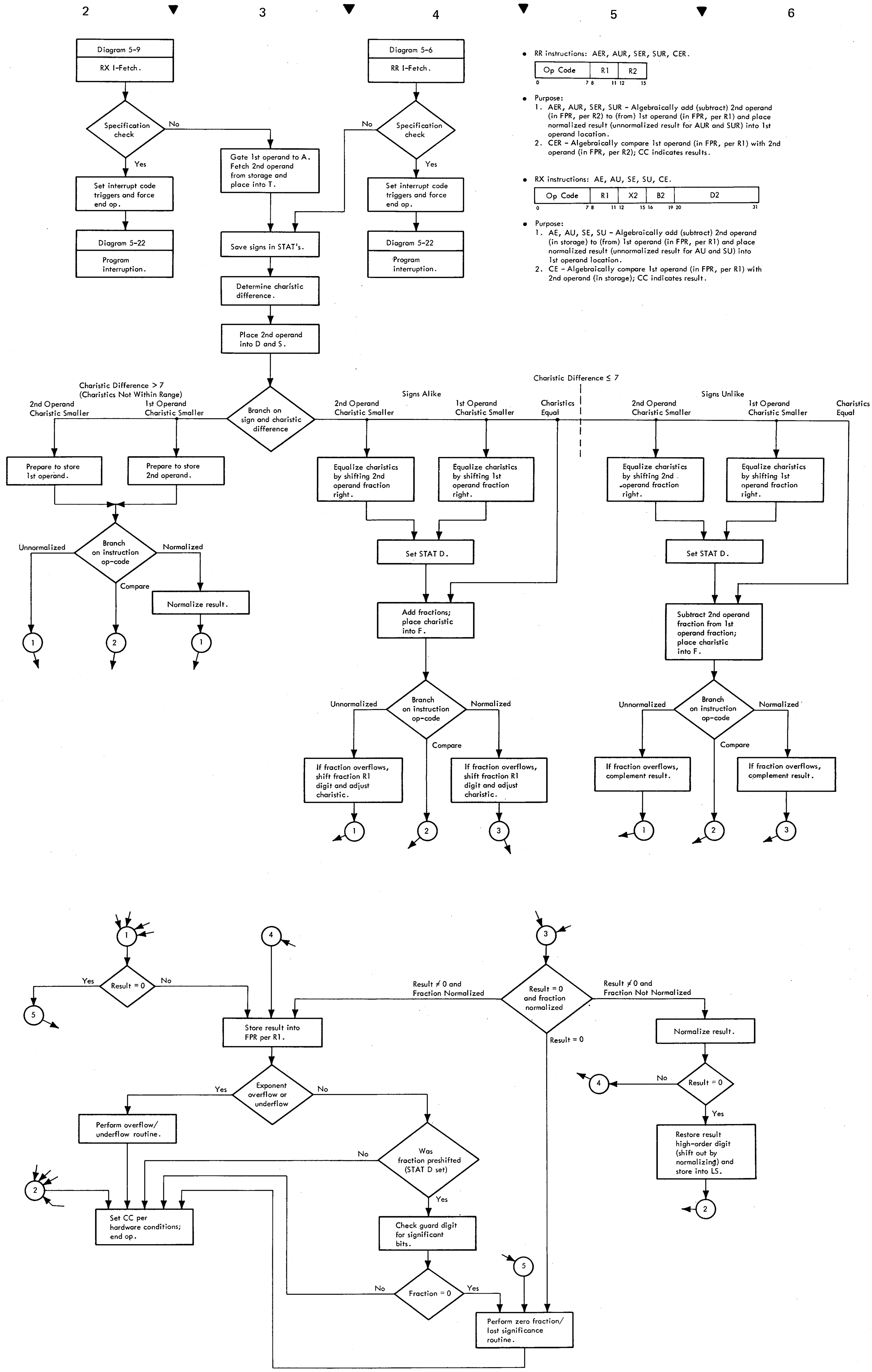
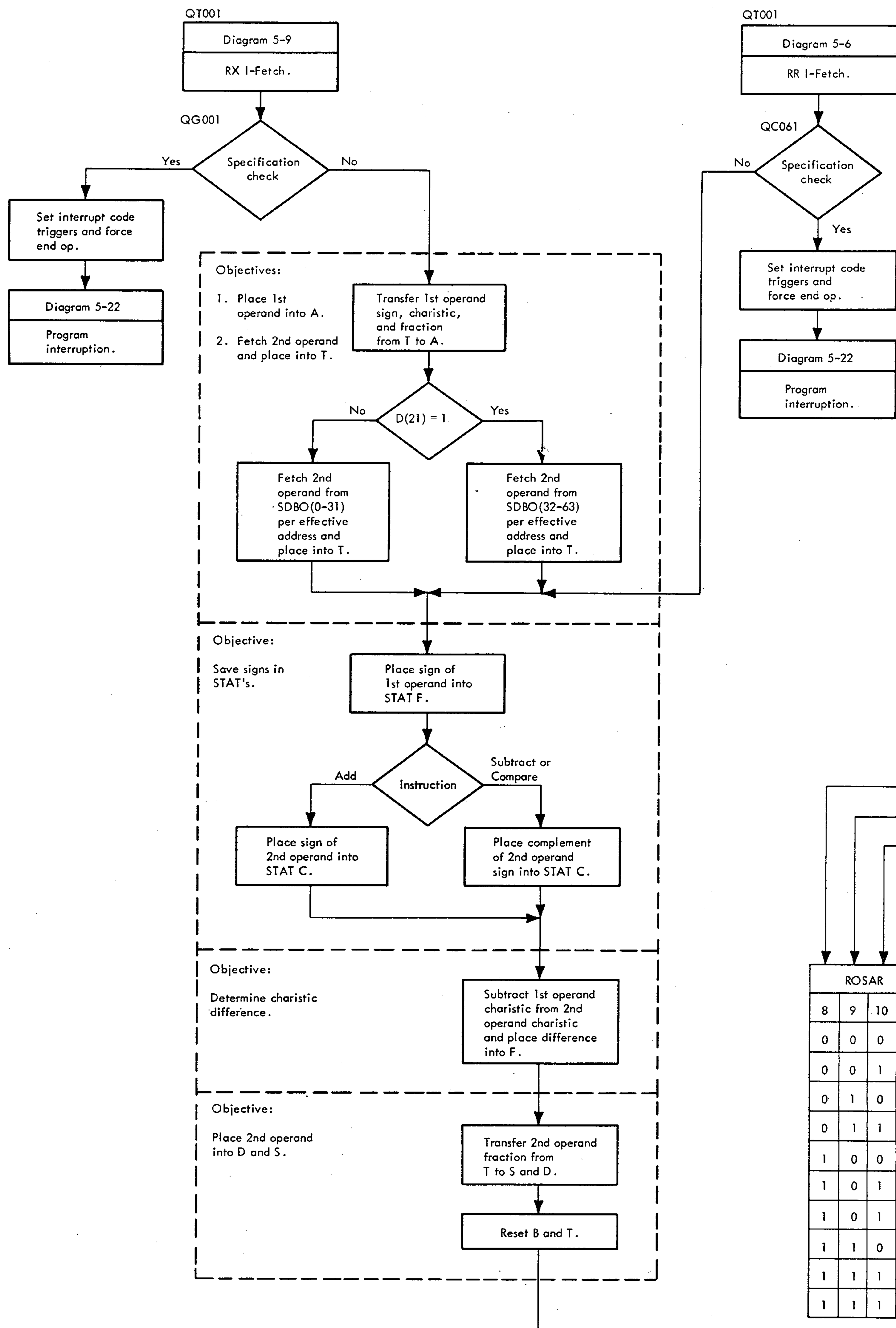


Diagram 5-206. Floating-Point Add, Subtract, and Compare - Short Operands (Sheet 1 of 5)



- RR Format - Short Operands

Op Code (0-7)	R1 (8-11)	R2 (12-15)

- Conditions at start of execution:
  1. Instruction is in E.
  2. 1st operand is in A, B, and D (24-bit fraction only).
  3. 2nd operand is in S and T.
  4. STC = 4 ('RESET' micro-order during I-Fetch).
- Op Code:
  1. AER = 3A.
  2. AUR = 3E.
  3. SER = 3B.
  4. SUR = 3F.
  5. CER = 39.

- RX Format - Short Operands

Op Code (0-7)	R1 (8-11)	X2 (12-15)	B2 (16-19)	D2 (20-31)

- Conditions at start of execution:
  1. 1st 16 bits of instruction are in E.
  2. 1st operand is in S and T.
  3. Effective address of 2nd operand is in D.
  4. 2nd operand is in main storage.
  5. STC = 4 ('RESET' micro-order during I-Fetch).
- Op Codes:
  1. AE = 7A.
  2. AU = 7E.
  3. SE = 7B.
  4. SU = 7F.
  5. CE = 79.

ROSAR bit 8 = 1 if a serial adder carry occurred (2nd operand charistic ≥ 1st operand charistic).
ROSAR bit 9 = 1 if signs are alike (add fractions).
ROSAR bit 10 = 1 if charistics are within range (charistic difference ≤ 7):
a. Serial adder carry and SAL(0-4) = 0.
b. No serial adder carry and SAL(0-4) = F (hex).
ROSAR bit 11 = 1 if SAL = 0 (charistics equal).

ROSAR 8	9	10	11	Operation Performed
0	0	0	0	Store 1st operand.
0	0	1	0	Preshift 2nd operand and subtract 2nd operand from 1st operand.
0	1	0	0	Store 1st operand.
0	1	1	0	Preshift 2nd operand and add 2nd operand to 1st operand.
1	0	0	0	Store 2nd operand.
1	0	1	0	Preshift 1st operand and subtract 2nd operand from 1st operand.
1	0	1	1	Subtract 2nd operand from 1st operand (charistics equal).
1	1	0	0	Store 2nd operand.
1	1	1	0	Preshift 1st operand and add 2nd operand to 1st operand.
1	1	1	1	Add 2nd operand to 1st operand (charistics equal).

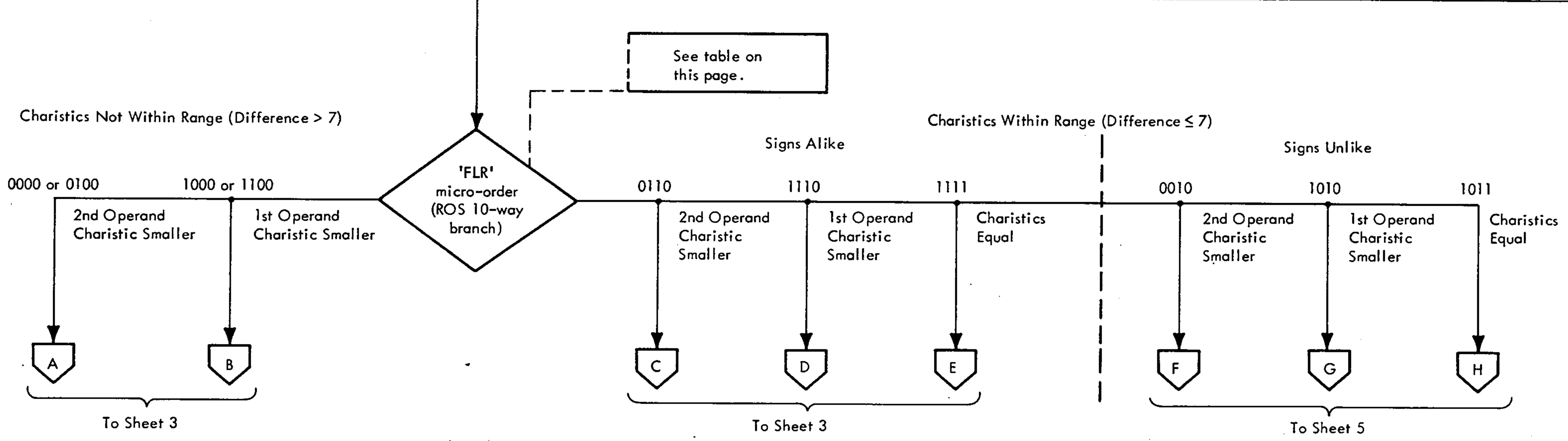


Diagram 5-206. Floating-Point Add, Subtract, and Compare - Short Operands (Sheet 2 of 5)

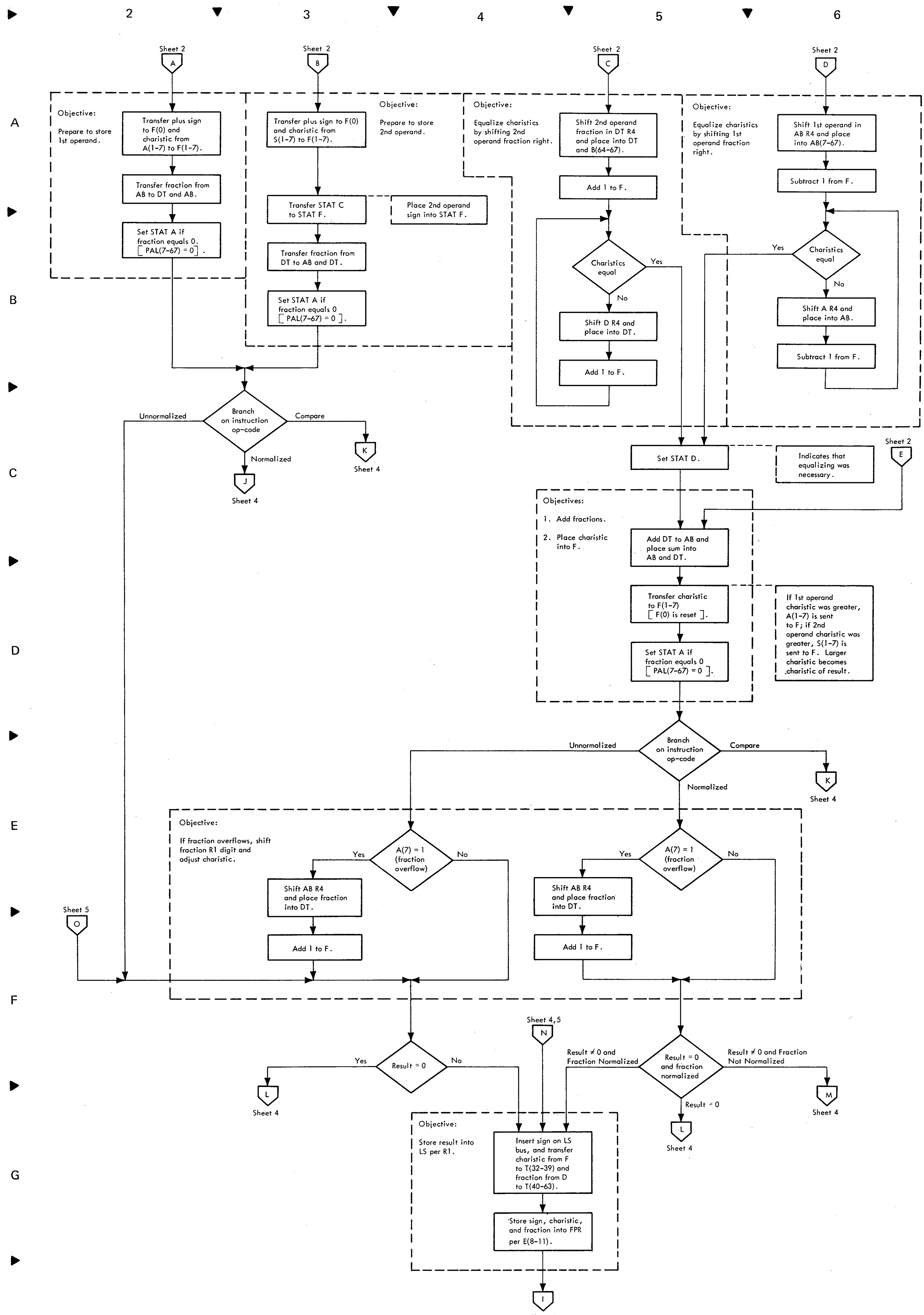


Diagram 5-206. Floating-Point Add, Subtract, and Compare - Short Operands (Sheet 3 of 5)

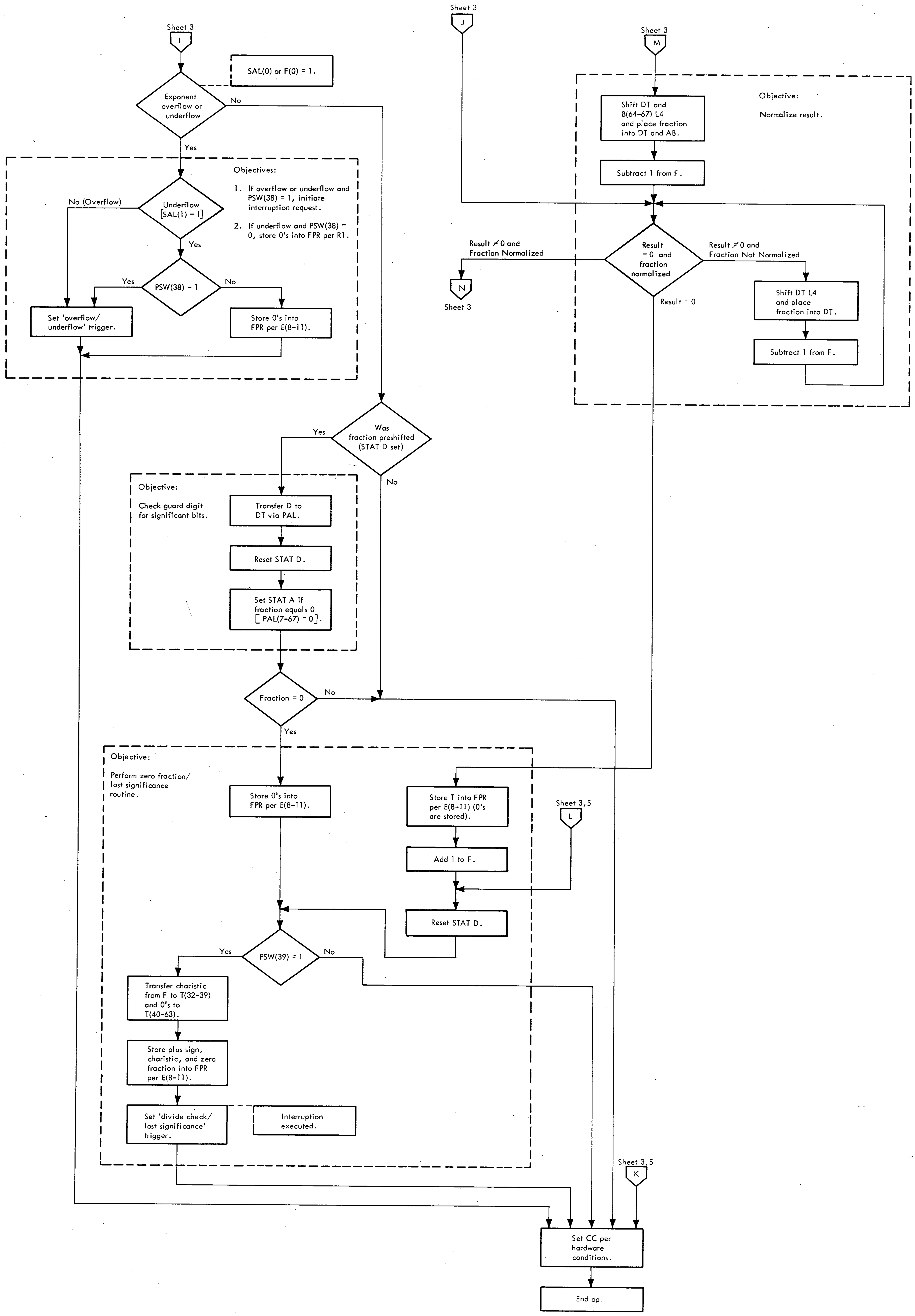


Diagram 5-206. Floating-Point Add, Subtract, and Compare - Short Operands (Sheet 4 of 5)

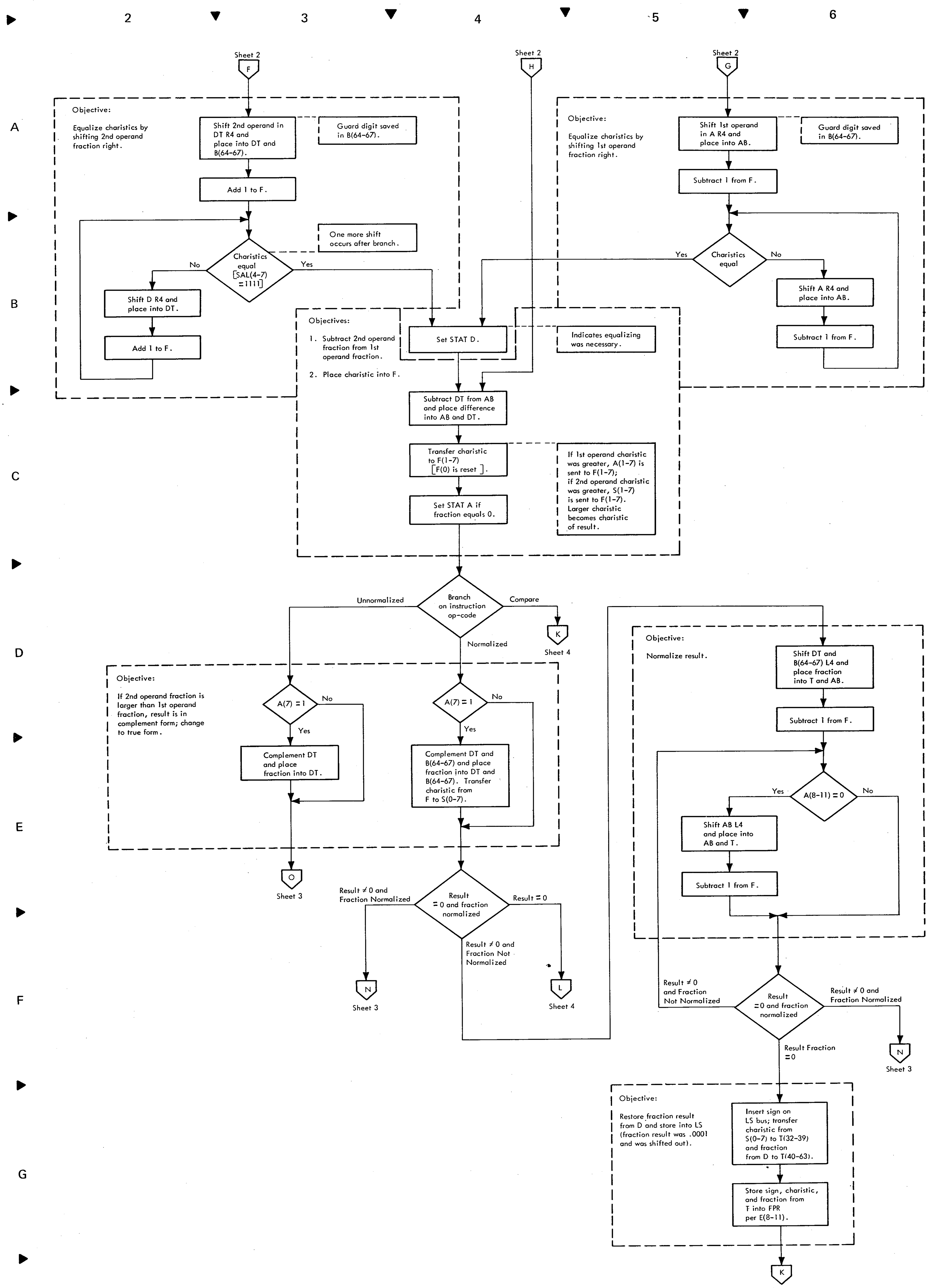


Diagram 5-206. Floating-Point Add, Subtract, and Compare - Short Operands (Sheet 5 of 5)

- RR instructions: ADR, AWR, SDR, SWR, CDR

Op Code	R1	R2
0 7	8 11	12 15

- Purpose:
  1. ADR, AWR, SDR, SWR - Algebraically add (subtract) 2nd operand (in FPR, per R2 and R2 + 1) to (from) 1st operand (in FPR, per R1 and R1 + 1) and place normalized result (unnormalized result for AWR and SWR) into 1st operand location.
  2. CDR - Algebraically compare 1st operand (in FPR, per R1 and R1 + 1) with 2nd operand (in FPR, per R2 and R2 + 1); CC indicates result.

- RX instructions: AD, AW, SD, SW, CD

Op Code	R1	X2	B2	D2
0 7	8 11	12 15	16 19	20 31

- Purpose:
  1. AD, AW, SD, SW - Algebraically add (subtract) 2nd operand (in storage) to (from) 1st operand (in FPR, per R1 and R1 + 1) and place normalized result (unnormalized result for AW and SW) into 1st operand location.
  2. CD - Algebraically compare 1st operand (in FPR, per R1 and R1 + 1) with 2nd operand (in storage); CC indicates result.

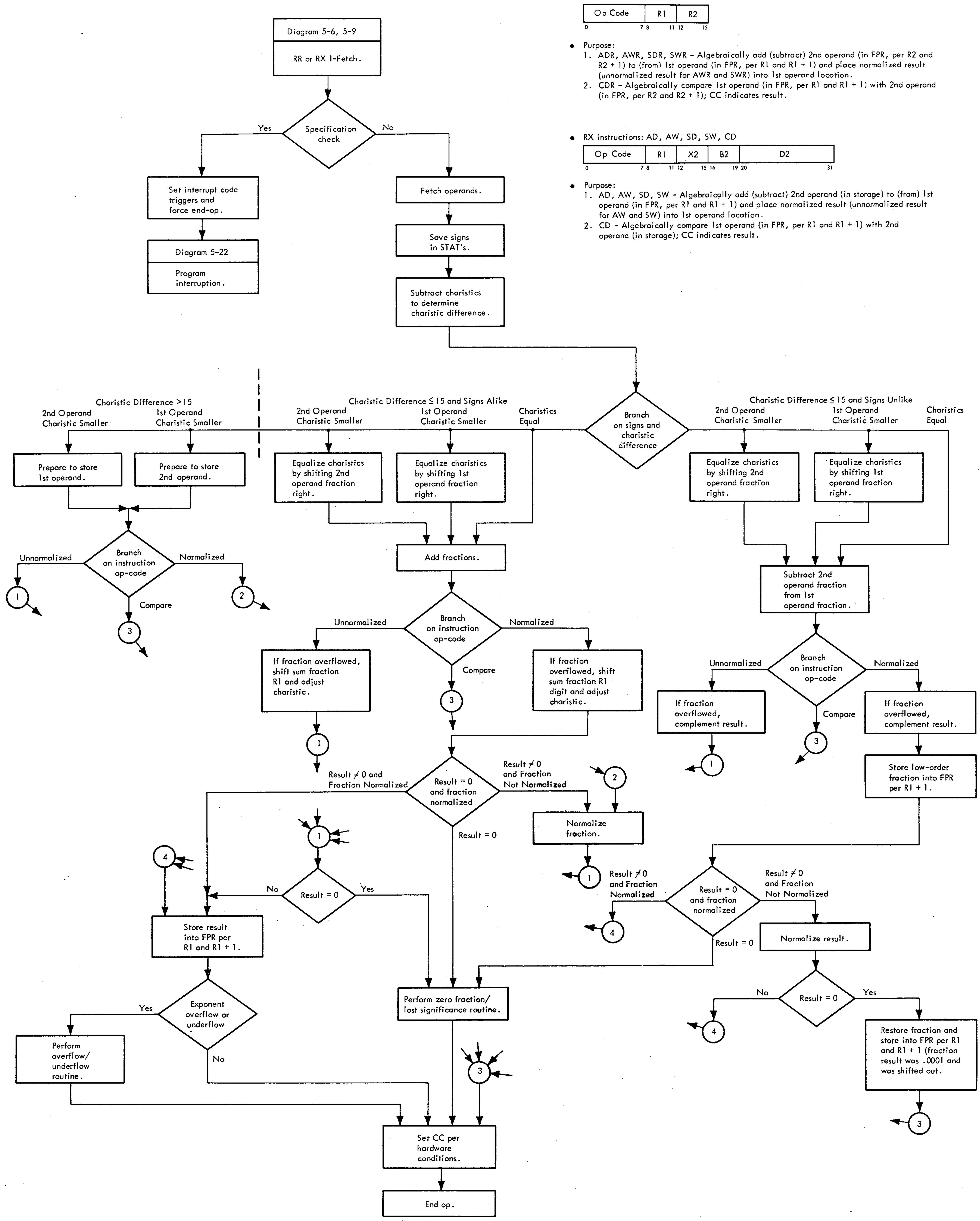
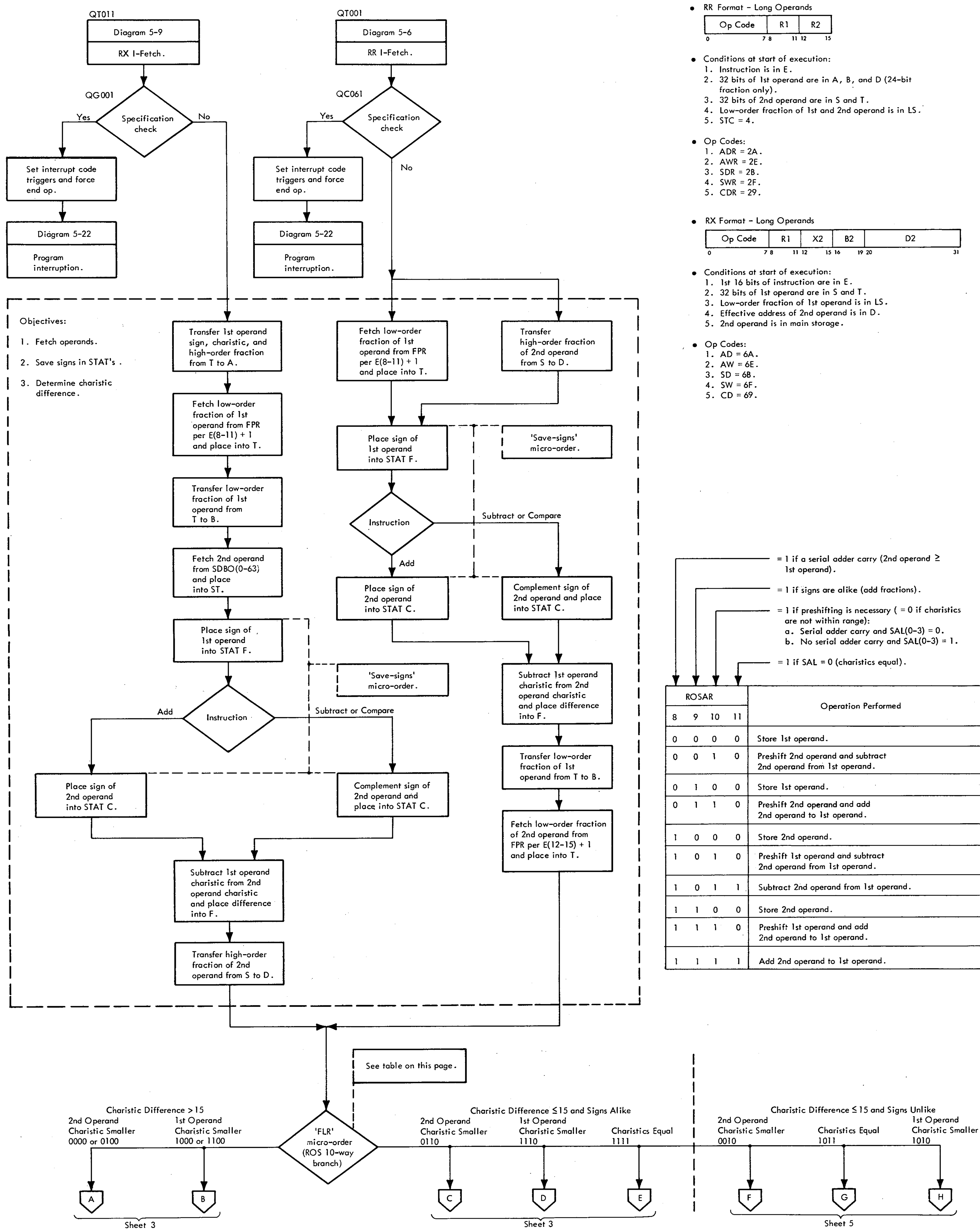


Diagram 5-207. Floating-Point Add, Subtract, and Compare - Long Operands (Sheet 1 of 5)



- RR Format - Long Operands

Op Code	R1	R2
0–7	8–11	12–15

- Conditions at start of execution:
  1. Instruction is in E.
  2. 32 bits of 1st operand are in A, B, and D (24-bit fraction only).
  3. 32 bits of 2nd operand are in S and T.
  4. Low-order fraction of 1st and 2nd operand is in LS.
  5. STC = 4.

- Op Codes:
  1. ADR = 2A.
  2. AWR = 2E.
  3. SDR = 2B.
  4. SWR = 2F.
  5. CDR = 29.

- RX Format - Long Operands

Op Code	R1	X2	B2	D2
0–7	8–11	12–15	16–19	20–31

- Conditions at start of execution:
  1. 1st 16 bits of instruction are in E.
  2. 32 bits of 1st operand are in S and T.
  3. Low-order fraction of 1st operand is in LS.
  4. Effective address of 2nd operand is in D.
  5. 2nd operand is in main storage.

- Op Codes:
  1. AD = 6A.
  2. AW = 6E.
  3. SD = 6B.
  4. SW = 6F.
  5. CD = 69.

ROSAR 8	9	10	11	Operation Performed
0	0	0	0	Store 1st operand.
0	0	1	0	Preshift 2nd operand and subtract 2nd operand from 1st operand.
0	1	0	0	Store 1st operand.
0	1	1	0	Preshift 2nd operand and add 2nd operand to 1st operand.
1	0	0	0	Store 2nd operand.
1	0	1	0	Preshift 1st operand and subtract 2nd operand from 1st operand.
1	0	1	1	Subtract 2nd operand from 1st operand.
1	1	0	0	Store 2nd operand.
1	1	1	0	Preshift 1st operand and add 2nd operand to 1st operand.
1	1	1	1	Add 2nd operand to 1st operand.

Diagram 5-207. Floating-Point Add, Subtract, and Compare - Long Operands (Sheet 2 of 5)

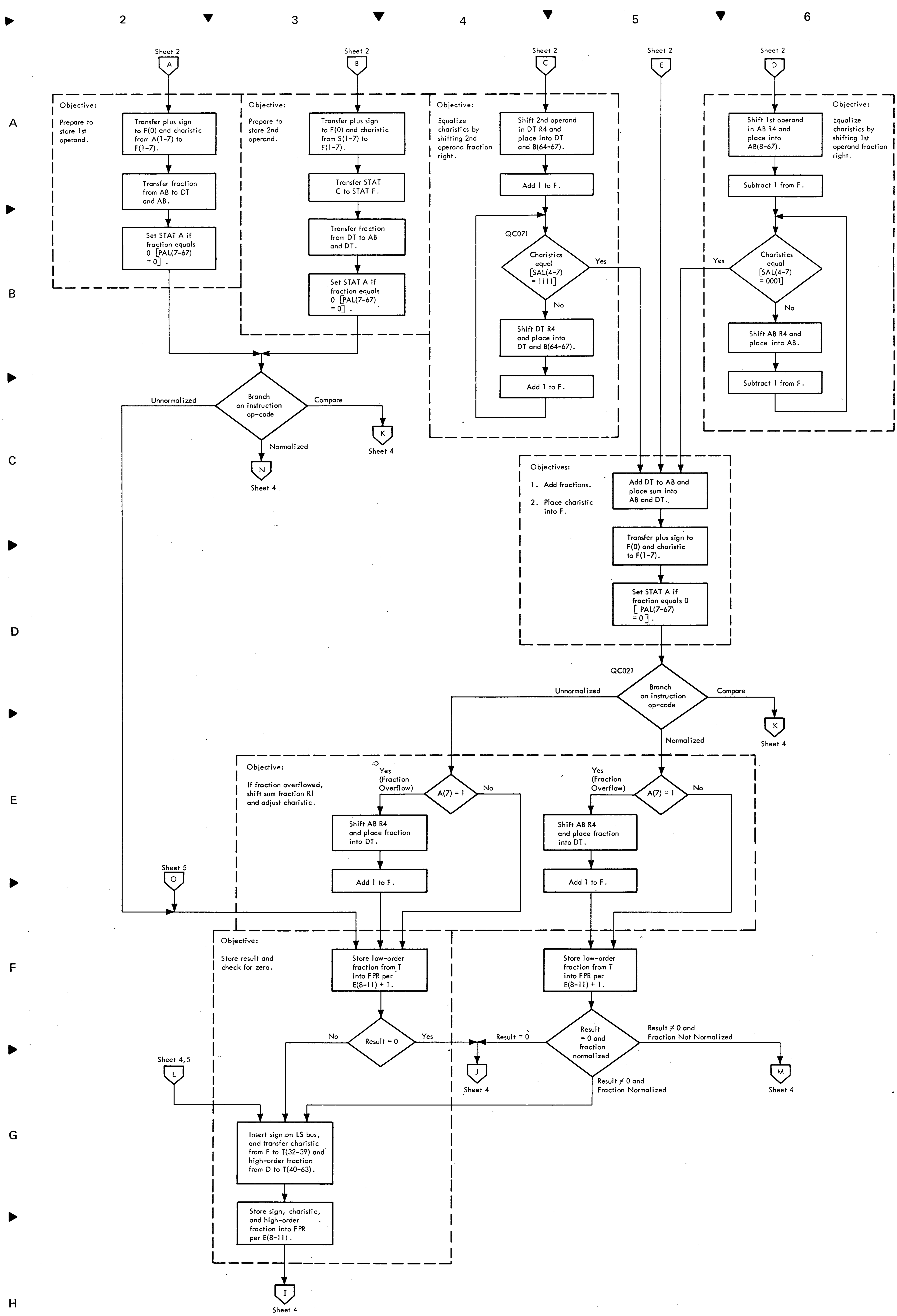


Diagram 5-207. Floating-Point Add, Subtract, and Compare - Long Operands (Sheet 3 of 5)

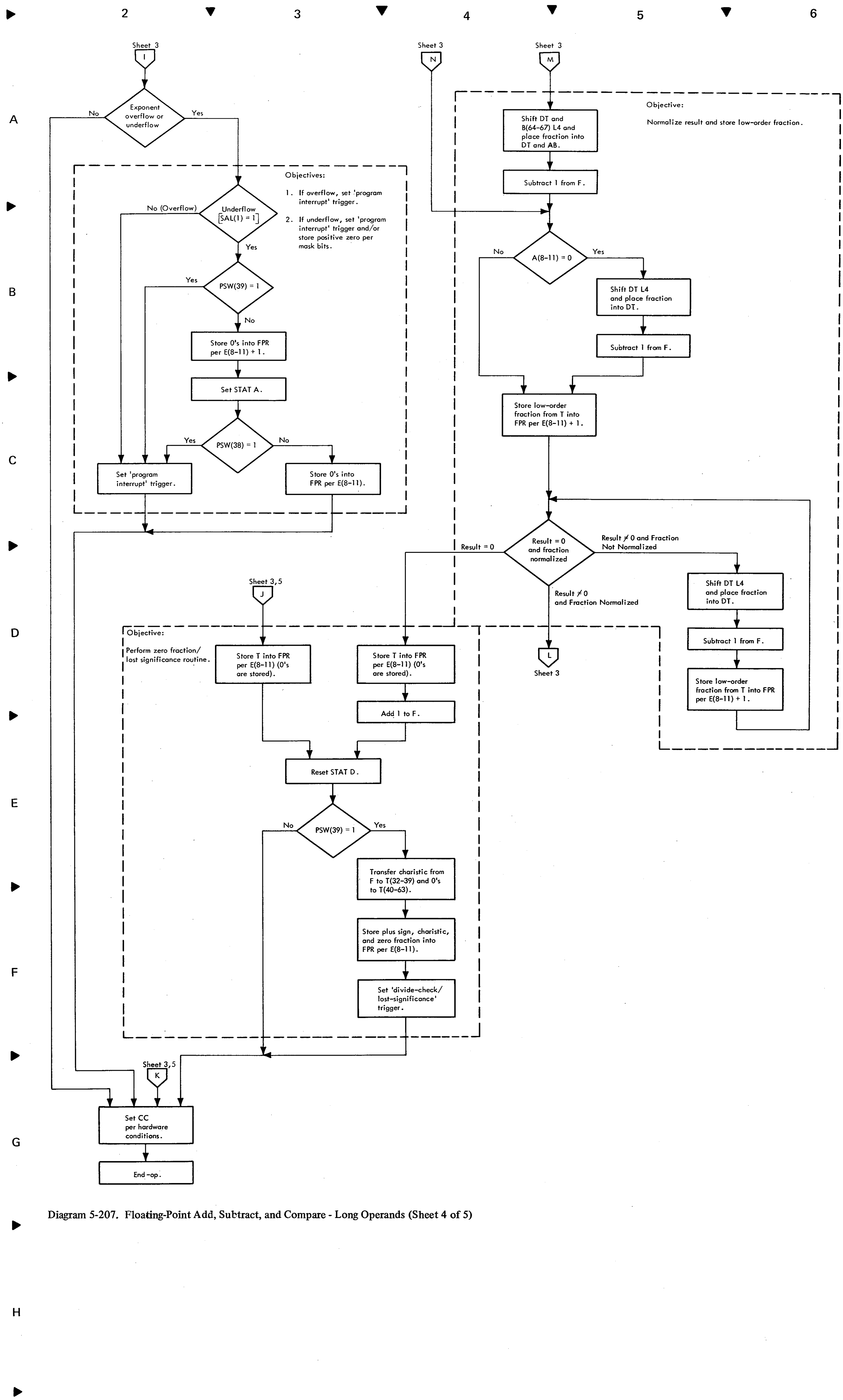


Diagram 5-207. Floating-Point Add, Subtract, and Compare - Long Operands (Sheet 4 of 5)

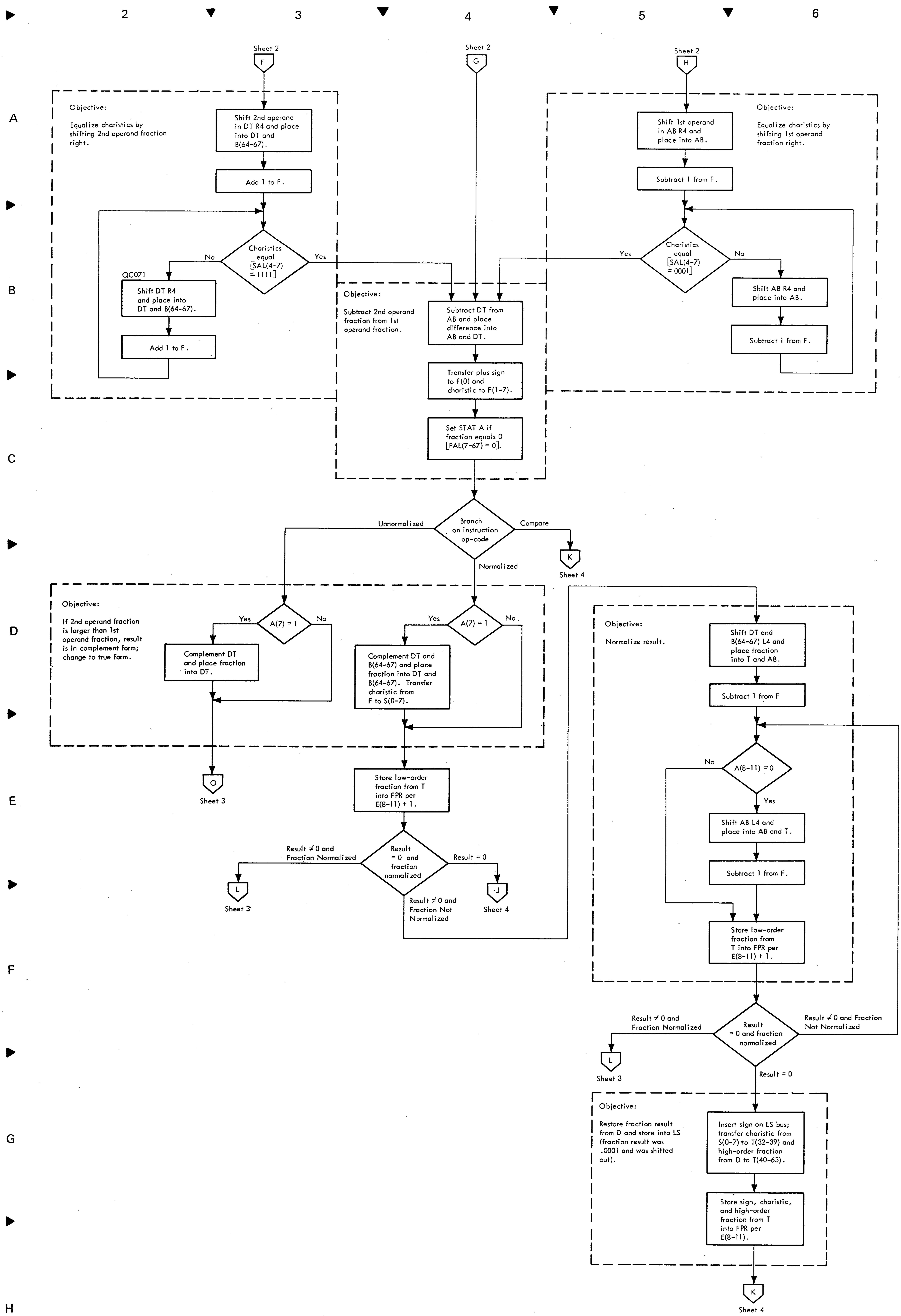


Diagram 5-207. Floating-Point Add, Subtract, and Compare - Long Operands (Sheet 5 of 5)

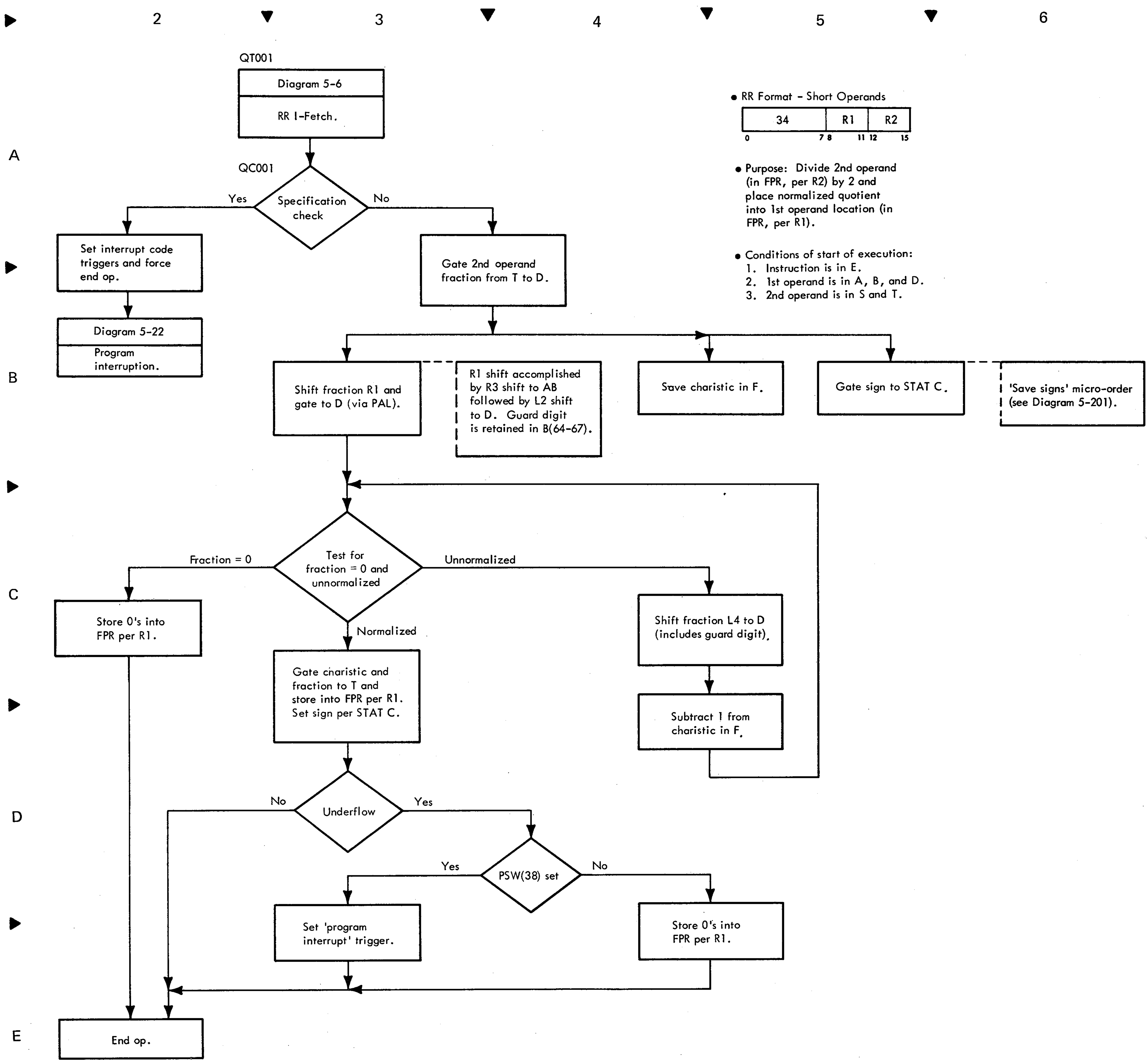


Diagram 5-208. Halve, HER (34) - Short Operands

- RR Format - Long Operands

24	R1	R2
0 — 7	8 — 11	12 — 15

- Purpose: Divide 2nd operand (in FPR, per R2 and R2 + 1) by 2 and place normalized quotient into 1st operand location (in FPR, per R1 and R1 + 1).

- Conditions of start of execution:
  Instruction is in E.
  32 bits of 1st operand are in A, B, and D.
  32 bits of 2nd operand are in S and T.

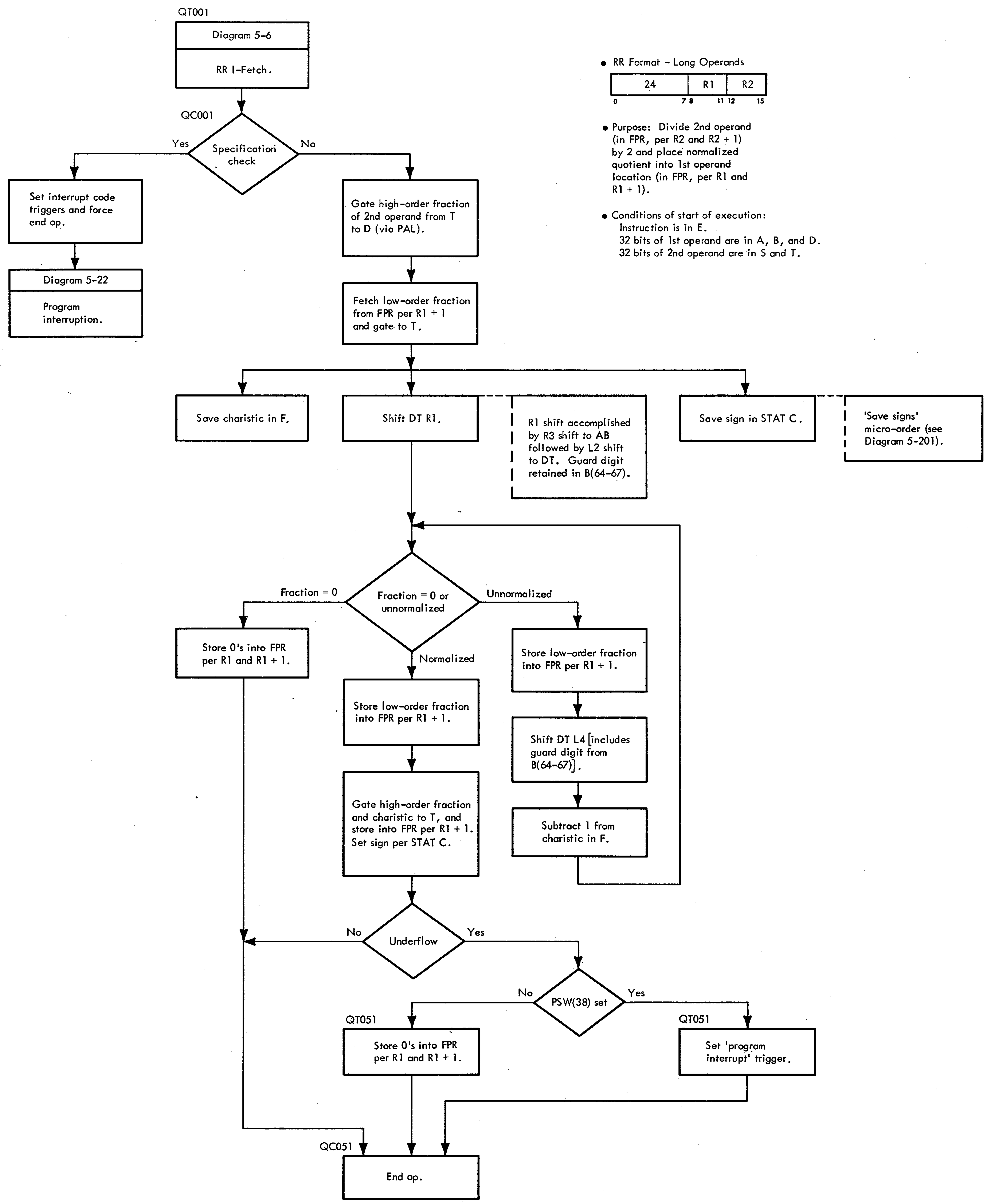
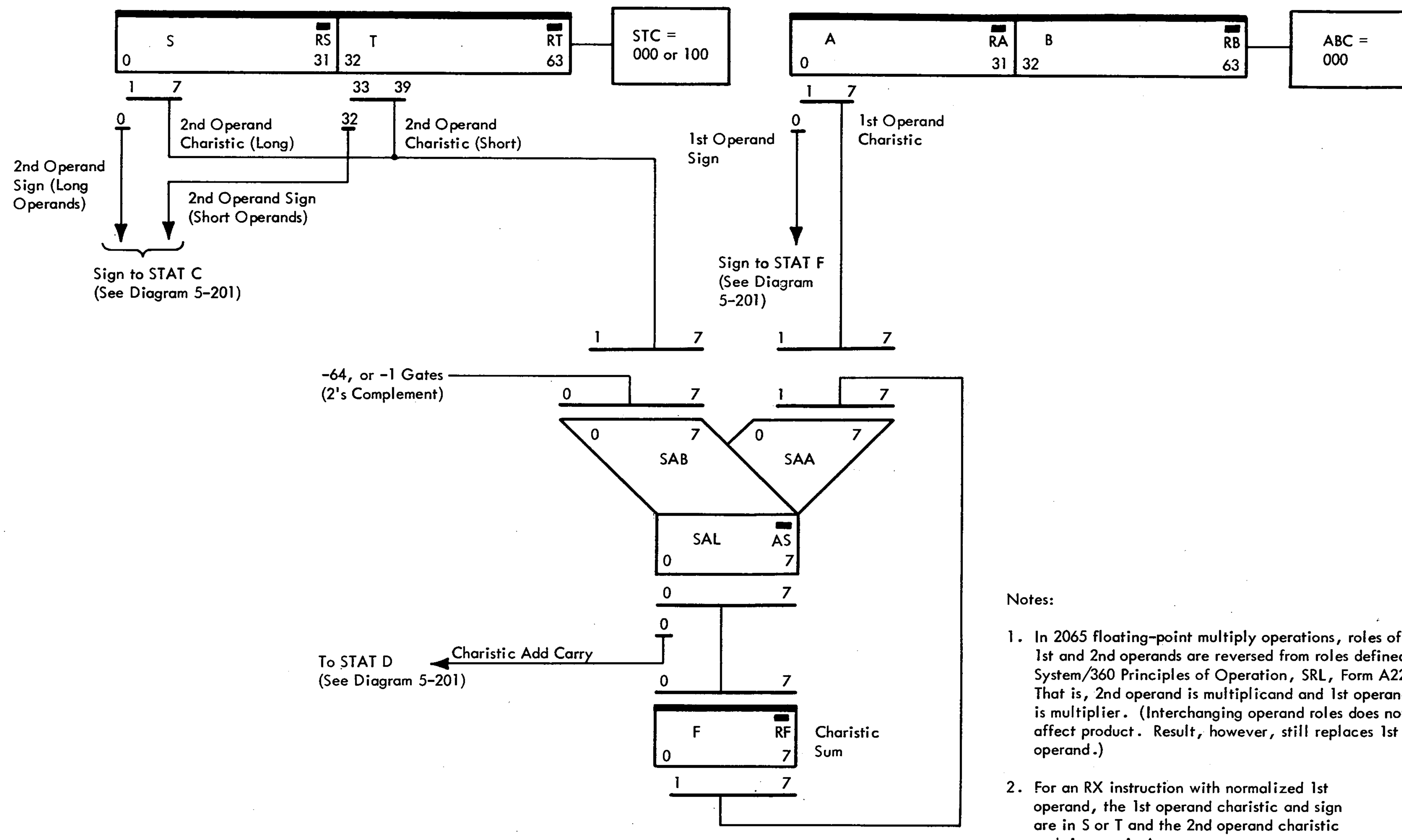


Diagram 5-209. Halve, HDR (24) - Long Operands

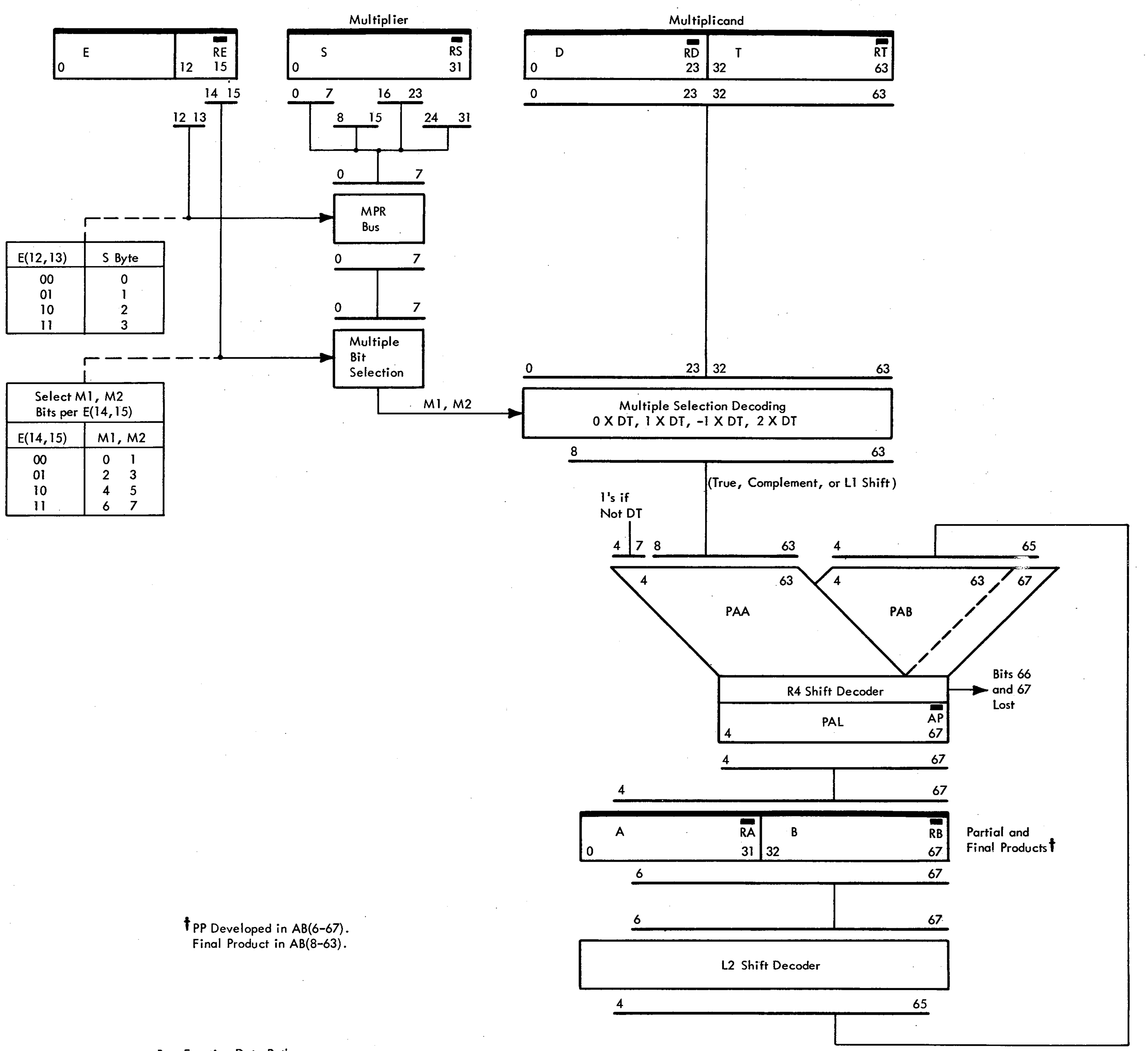


A. Sign and Characteristic Data Paths.

Notes:

1. In 2065 floating-point multiply operations, roles of 1st and 2nd operands are reversed from roles defined in System/360 Principles of Operation, SRL, Form A22-6821-6 That is, 2nd operand is multiplicand and 1st operand is multiplier. (Interchanging operand roles does not affect product. Result, however, still replaces 1st operand.)

2. For an RX instruction with normalized 1st operand, the 1st operand charistic and sign are in S or T and the 2nd operand charistic and sign are in A.



E(12,13)	S Byte
00	0
01	1
10	2
11	3

Select M1, M2 Bits per E(14,15)

E(14,15)	M1, M2
00	0 1
01	2 3
10	4 5
11	6 7

†PP Developed in AB(6-67).
Final Product in AB(8-63).

B. Fraction Data Path.

Diagram 5-210. Floating-Point Multiply Data Paths

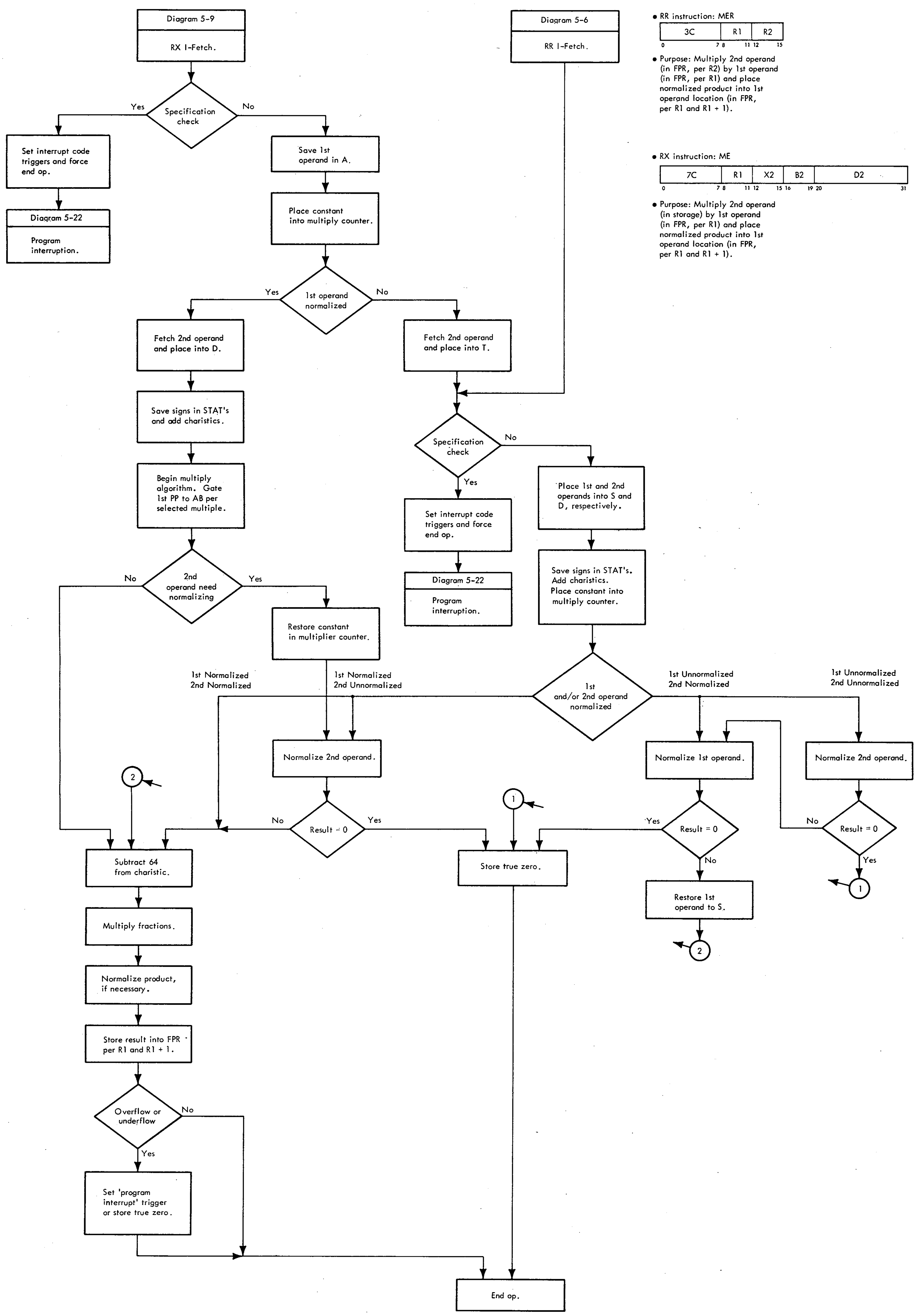


Diagram 5-211. Floating-Point Multiply, Short Operands (Sheet 1 of 4)

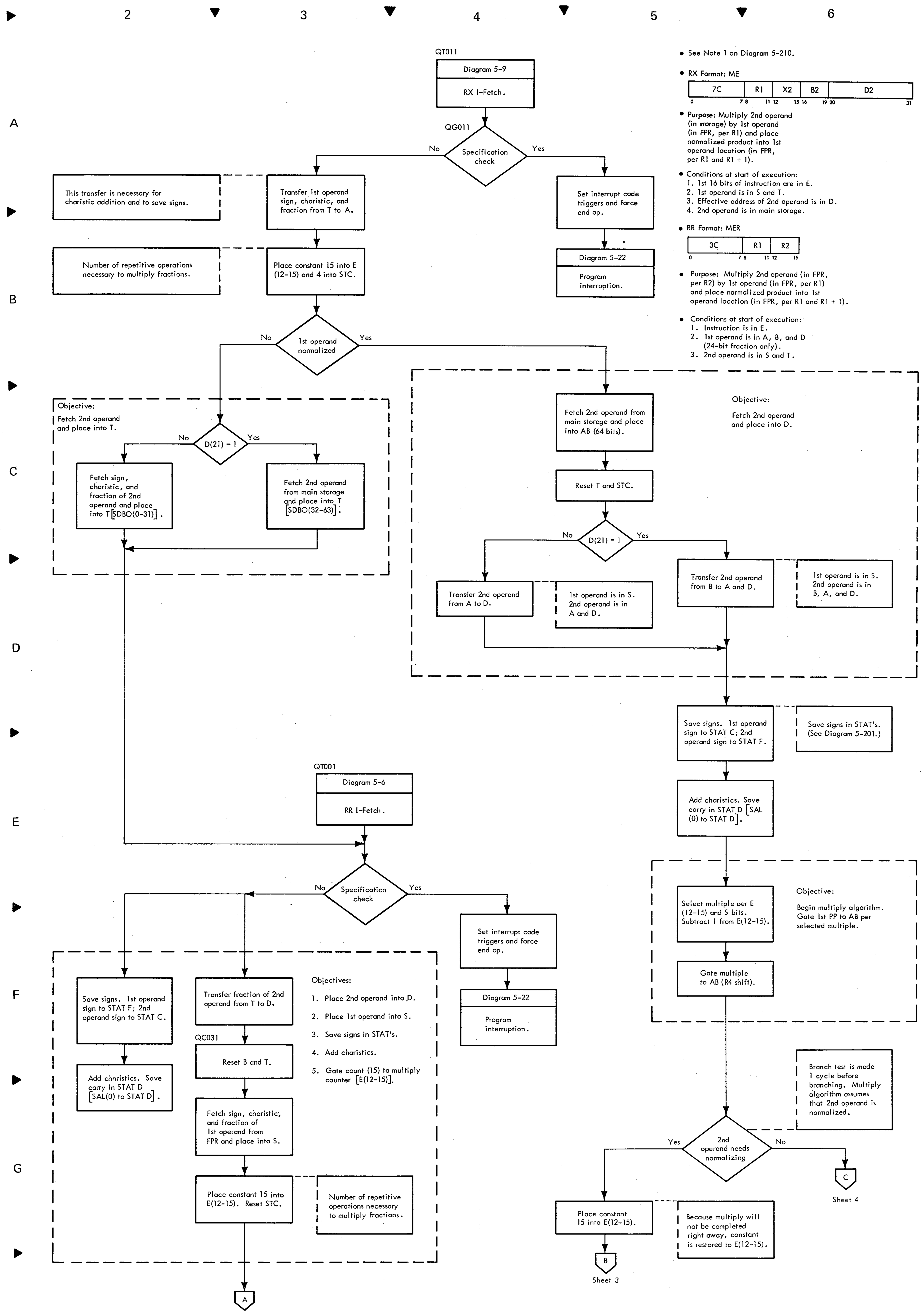


Diagram 5-211. Floating-Point Multiply, Short Operands (Sheet 2 of 4)

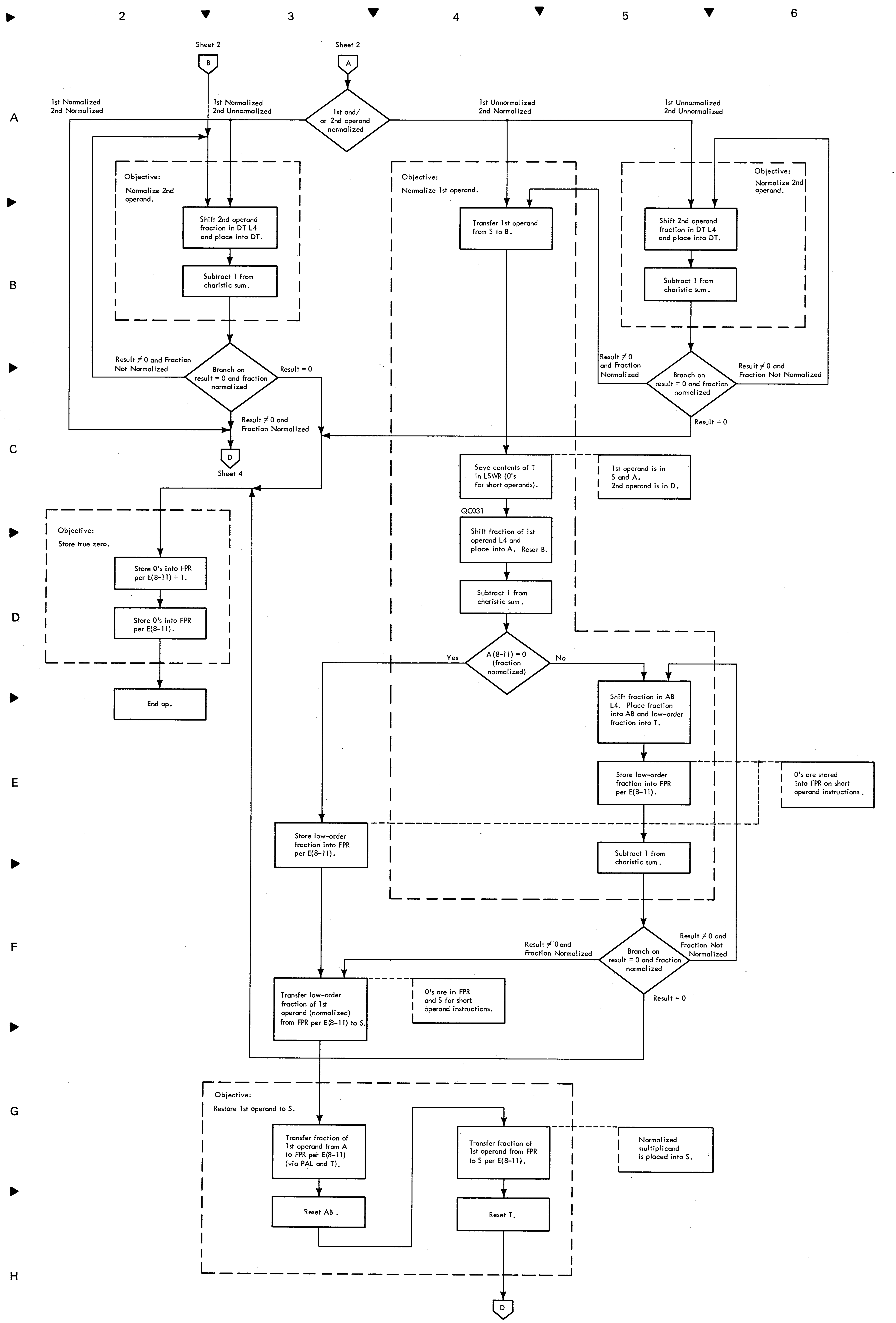
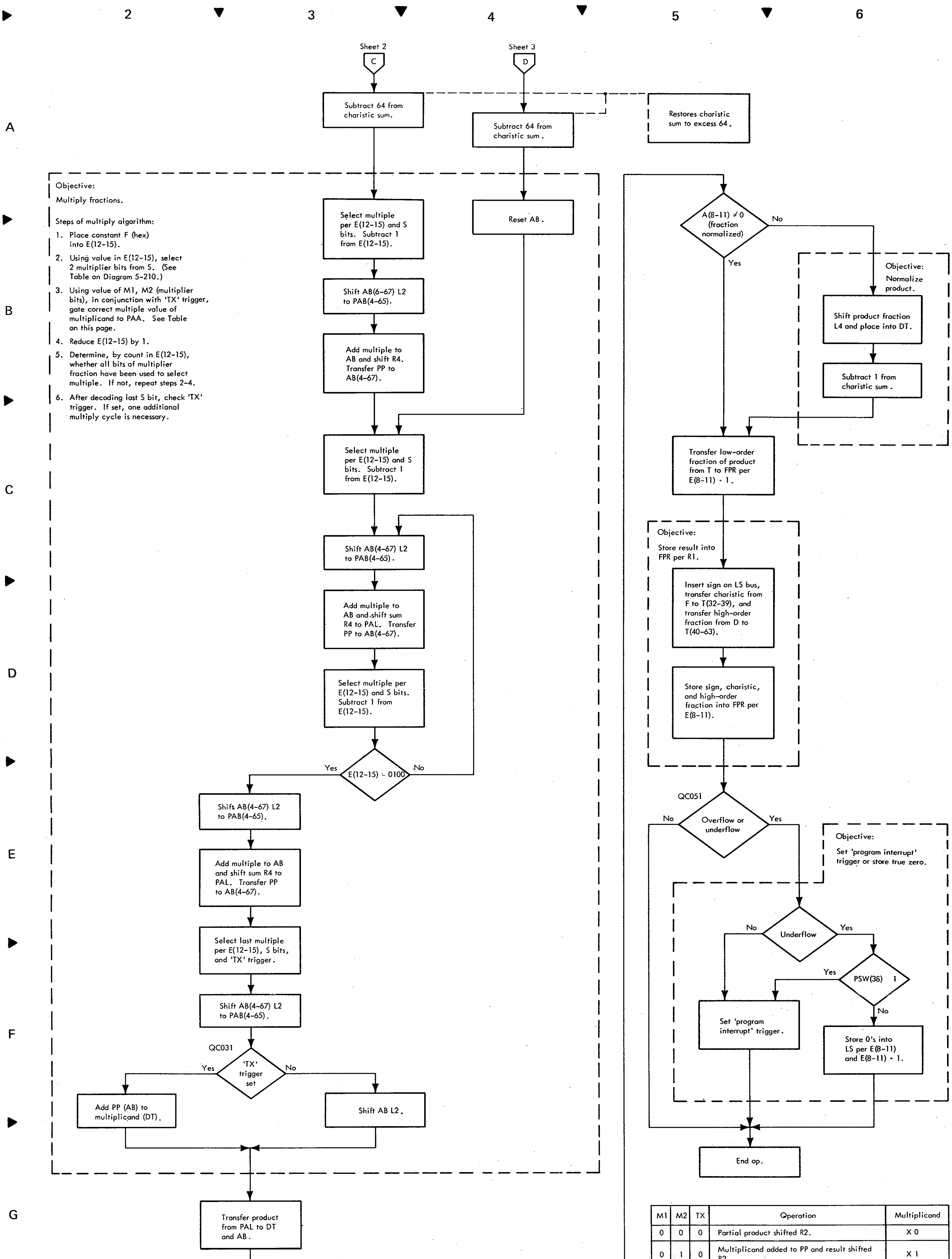


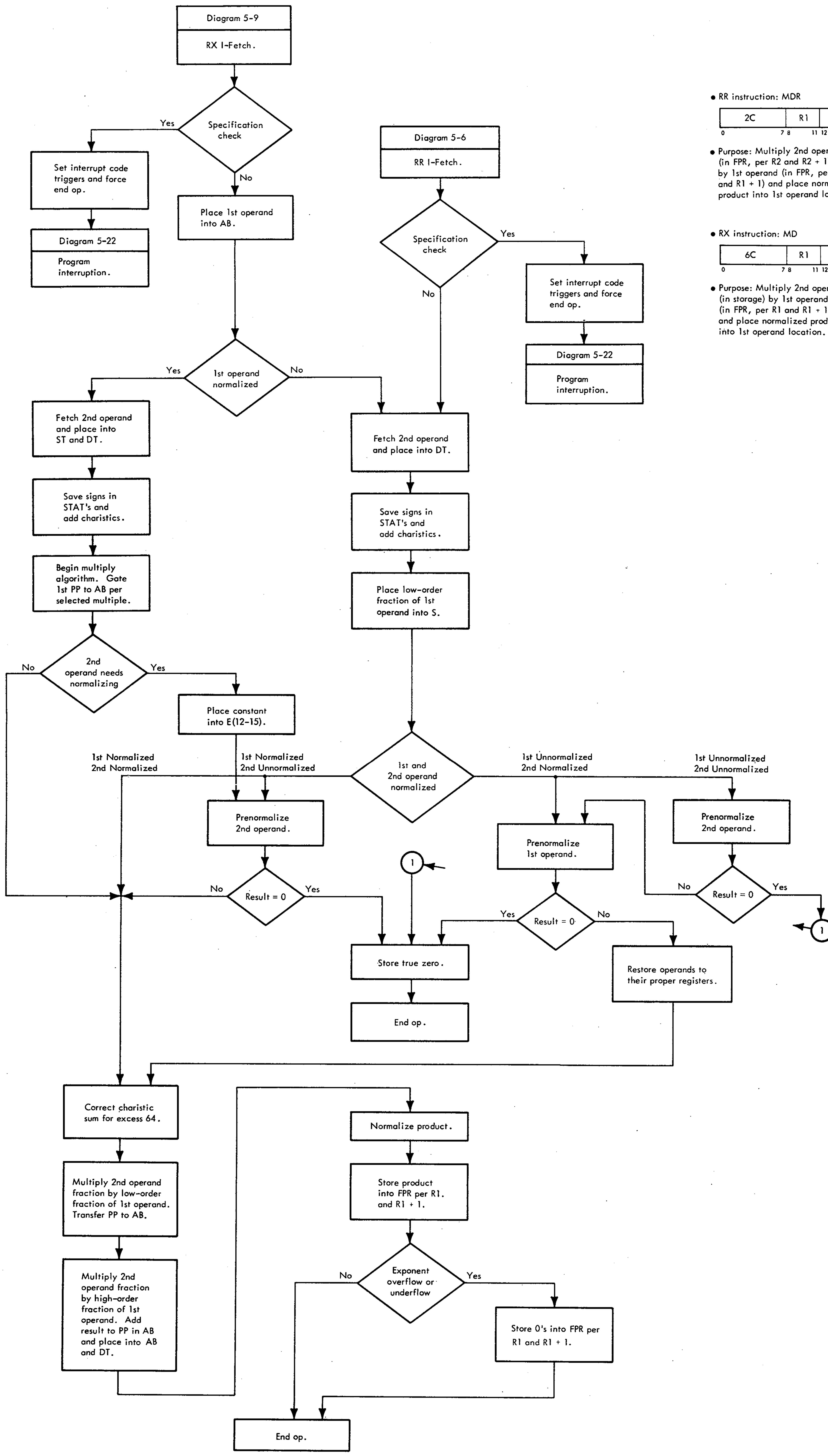
Diagram 5-211. Floating-Point, Short Operands (Sheet 3 of 4)



M1	M2	TX	Operation	Multiplicand
0	0	0	Partial product shifted R2.	X 0
0	1	0	Multiplicand added to PP and result shifted R2.	X 1
1	0	0	Multiplicand shifted L2 and added to PP; result shifted R2.	X 2
1	1	0	2's complement of multiplicand added to partial product, 'TX' trigger set, and result shifted R2.	X 3 (X -1 and 'TX')
0	0	1	Multiplicand added to PP and result shifted R2.	X 1
0	1	1	Multiplicand shifted L2 and added to PP; result shifted R2.	X 2
1	0	1	2's complement of multiplicand added to PP, 'TX' trigger set, and result shifted R2.	X 3 (X -1 and 'TX')
1	1	1	Partial product shifted R2 and 'TX' trigger set	X 0 and 'TX'

Note: See Note 1 on Diagram 5-210.

Diagram 5-211. Floating-Point Multiply, Short Operands (Sheet 4 of 4)



- RR instruction: MDR

2C	R1	R2
0–7	8–11	12–15

- Purpose: Multiply 2nd operand (in FPR, per R2 and R2 + 1) by 1st operand (in FPR, per R1 and R1 + 1) and place normalized product into 1st operand location.

- RX instruction: MD

6C	R1	R3	B2	D2
0–7	8–11	12–15	16–19	20–31

- Purpose: Multiply 2nd operand (in storage) by 1st operand (in FPR, per R1 and R1 + 1) and place normalized product into 1st operand location.

Diagram 5-212. Floating-Point Multiply, Long Operands (Sheet 1 of 4)

- See Note 1 on Diagram 5-210.
- RX Format: MD

6C	R1	X2	B2	D2
0–7	8–11	12–15	16–19	20–31

- Purpose: Multiply 2nd operand (in storage) by 1st operand (in FPR, per R1 and R1 + 1) and place normalized product into 1st operand location.
- Conditions at start of execution:
  1. 1st 16 bits of instruction are in E.
  2. 32 bits of 1st operand are in S and T.
  3. Low-order fraction of 1st operand is in LS.
  4. Effective address of 2nd operand is in D.
  5. 2nd operand is in main storage.

- RR Format: MDR

2C	R1	R2
0–7	8–11	12–15

- Purpose: Multiply 2nd operand (in FPR, per R2 and R2 + 1) by 1st operand (in FPR, per R1 and R1 + 1) and place normalized product into 1st operand location.
- Conditions at start of execution:
  1. Instruction is in E.
  2. 32 bits of 1st operand are in A, B and D (24-bit fraction only).
  3. 32 bits of 2nd operand are in S and T.
  4. Low-order fractions of 1st and 2nd operands are in LS.
  5. STC = 4.

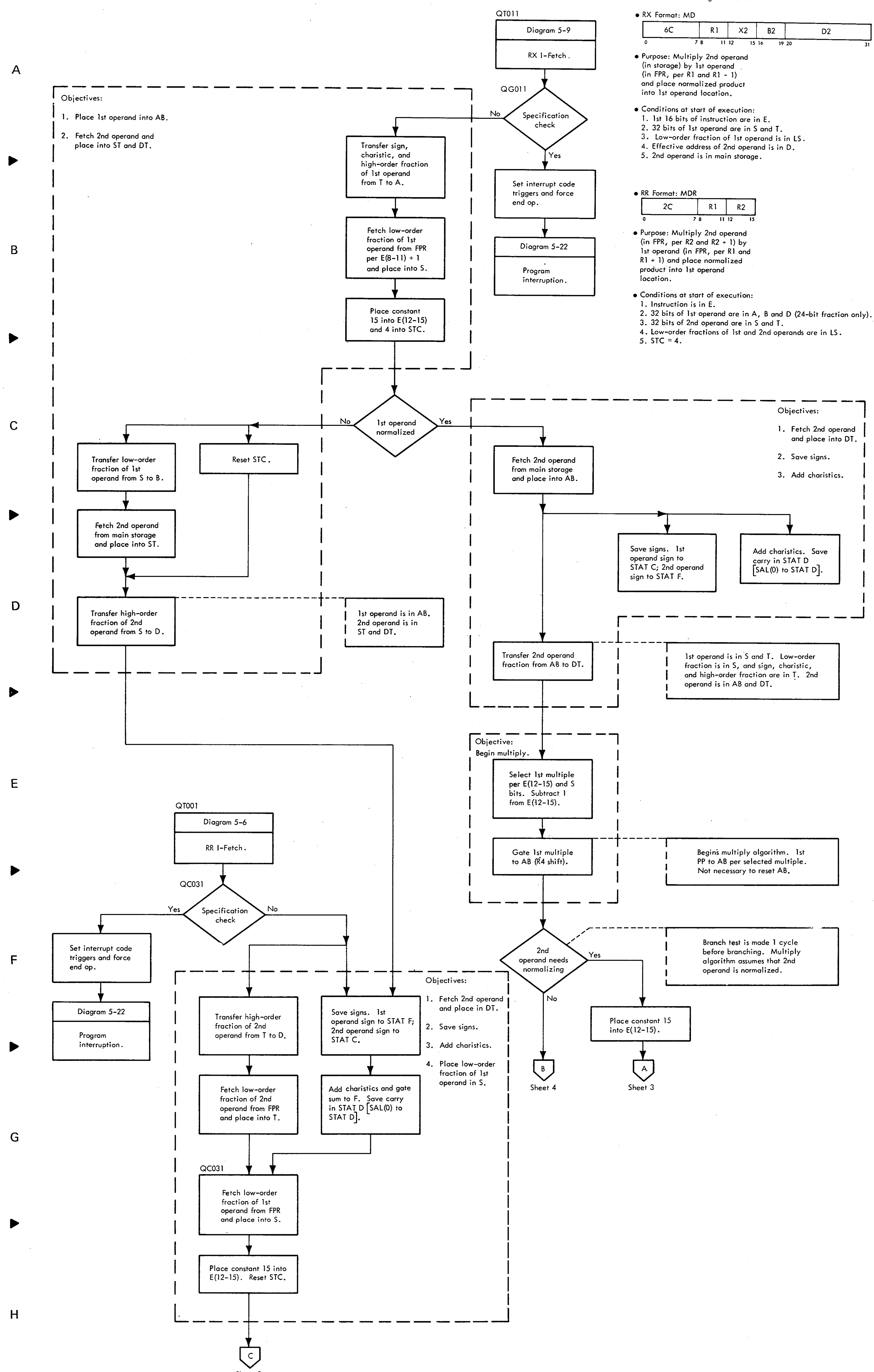


Diagram 5-212. Floating-Point Multiply, Long Operands (Sheet 2 of 4)

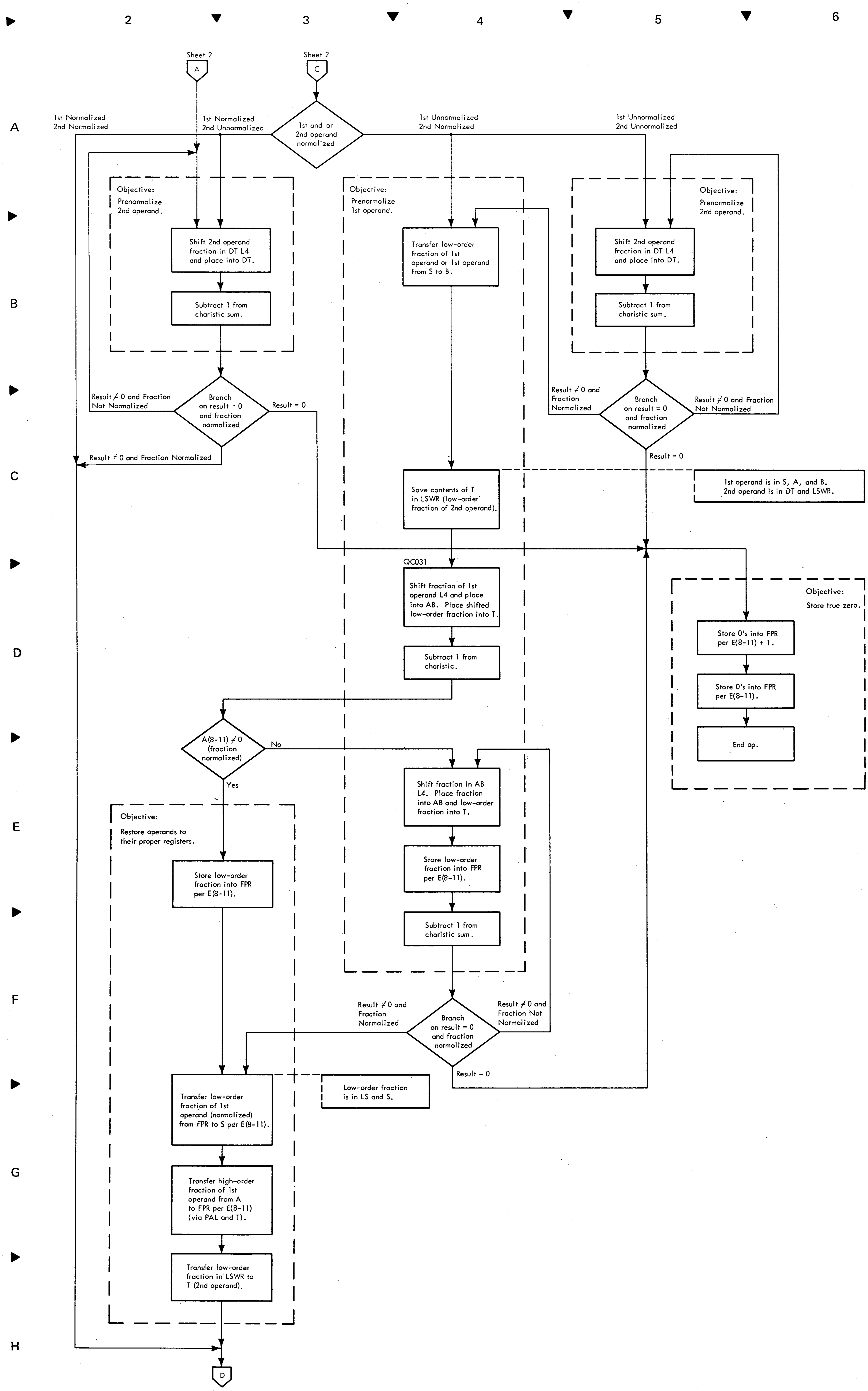
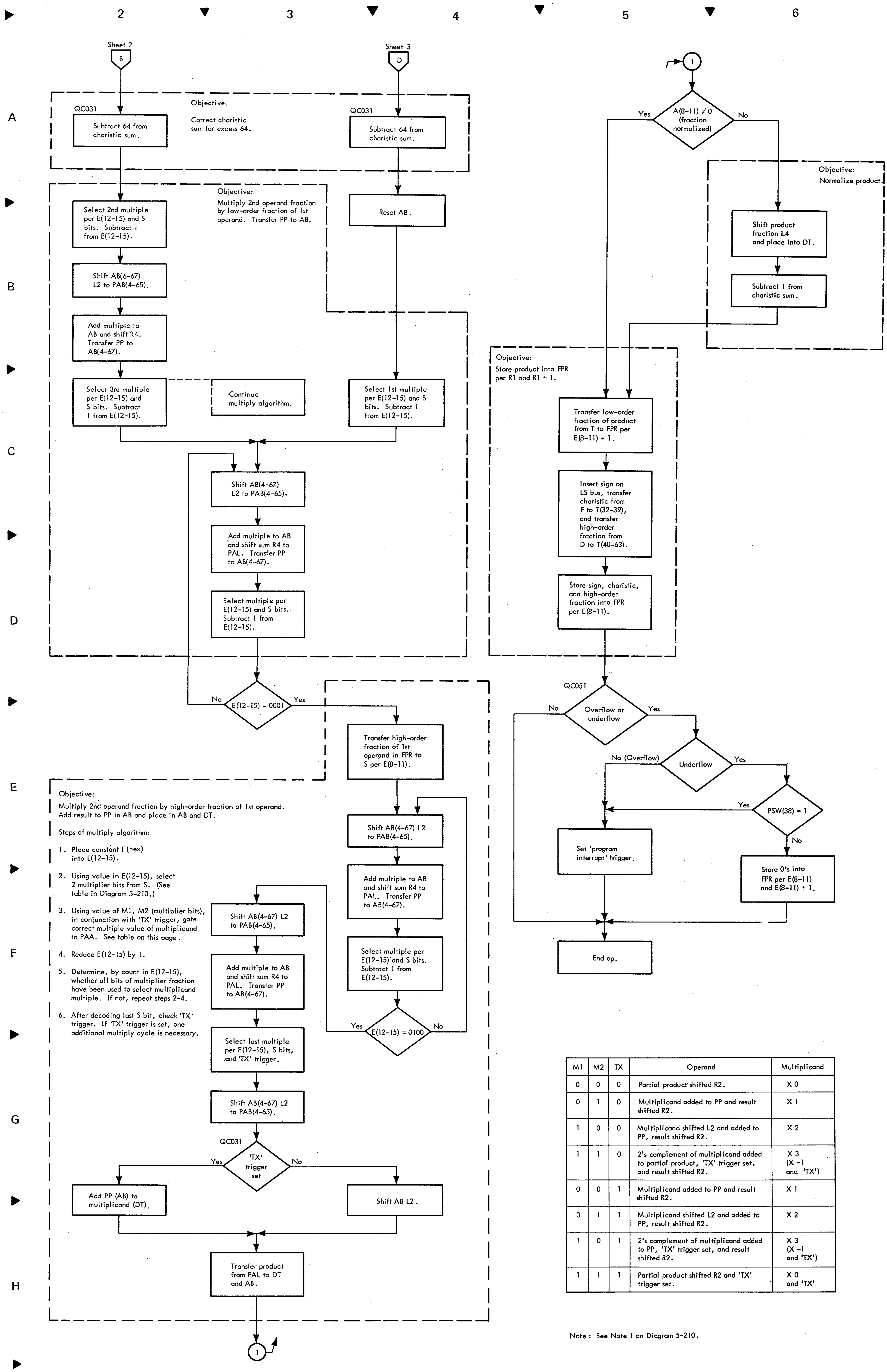


Diagram 5-212. Floating-Point Multiply, Long Operands (Sheet 3 of 4)



M1	M2	TX	Operand	Multiplicand
0	0	0	Partial product shifted R2.	X 0
0	1	0	Multiplicand added to PP and result shifted R2.	X 1
1	0	0	Multiplicand shifted L2 and added to PP, result shifted R2.	X 2
1	1	0	2's complement of multiplicand added to partial product, 'TX' trigger set, and result shifted R2.	X 3 (X -1 and 'TX')
0	0	1	Multiplicand added to PP and result shifted R2.	X 1
0	1	1	Multiplicand shifted L2 and added to PP, result shifted R2.	X 2
1	0	1	2's complement of multiplicand added to PP, 'TX' trigger set, and result shifted R2.	X 3 (X -1 and 'TX')
1	1	1	Partial product shifted R2 and 'TX' trigger set.	X 0 and 'TX'

Note: See Note 1 on Diagram 5-210.

Diagram 5-212. Floating-Point Multiply, Long Opernads (Sheet 4 of 4)

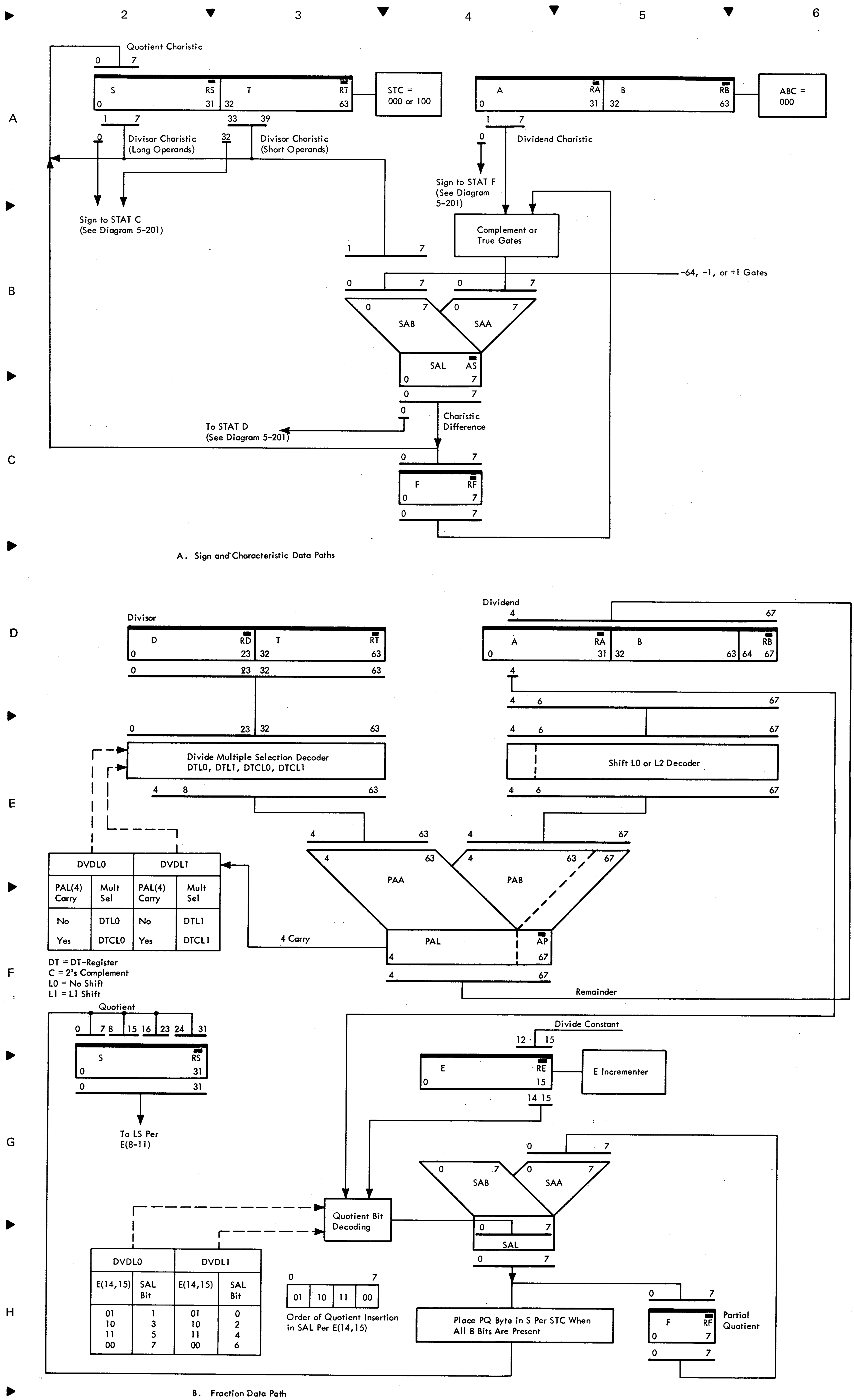
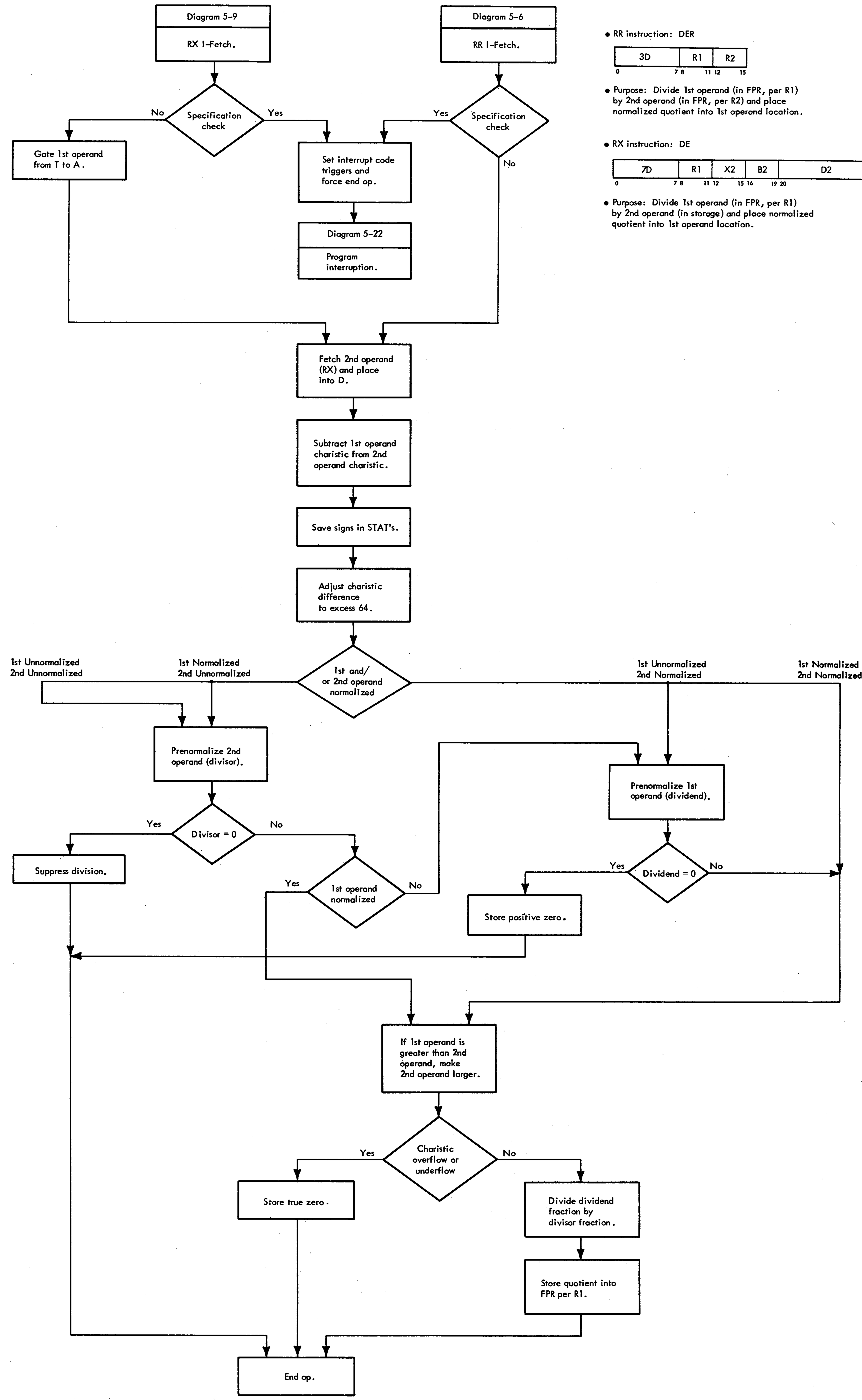


Diagram 5-213. Floating-Point Divide Data Paths

2 3 4 5 6



- RR instruction: DER

3D	R1	R2
0 7	8 11	12 15

- Purpose: Divide 1st operand (in FPR, per R1) by 2nd operand (in FPR, per R2) and place normalized quotient into 1st operand location.

- RX instruction: DE

7D	R1	X2	B2	D2
0 7	8 11	12 15	16 19	20 31

- Purpose: Divide 1st operand (in FPR, per R1) by 2nd operand (in storage) and place normalized quotient into 1st operand location.

Diagram 5-214. Floating-Point Divide, Short Operands (Sheet 1 of 4)

- RX Format: DE

7D	R1	X2	B2	D2
0–7	8–11	12–15	16–19	20–31

- Purpose: Divide 1st operand (in FPR, per R1) by 2nd operand (in storage) and place normalized quotient into 1st operand location.
- Conditions at start of execution:
  1. 1st 16 bits of instruction are in E.
  2. 1st operand is in S and T.
  3. Effective address of 2nd operand is in D.
  4. 2nd operand is in main storage.

- RR Format: DER

3D	R1	R2
0–7	8–11	12–15

- Purpose: Divide 1st operand (in FPR, per R1) by 2nd operand (in FPR, per R2) and place normalized quotient into 1st operand location.
- Conditions at start of execution:
  1. Instruction is in E.
  2. 1st operand is in A, B, and D (24-bit fraction only).
  3. 2nd operand is in S and T.
  4. STC = 4.

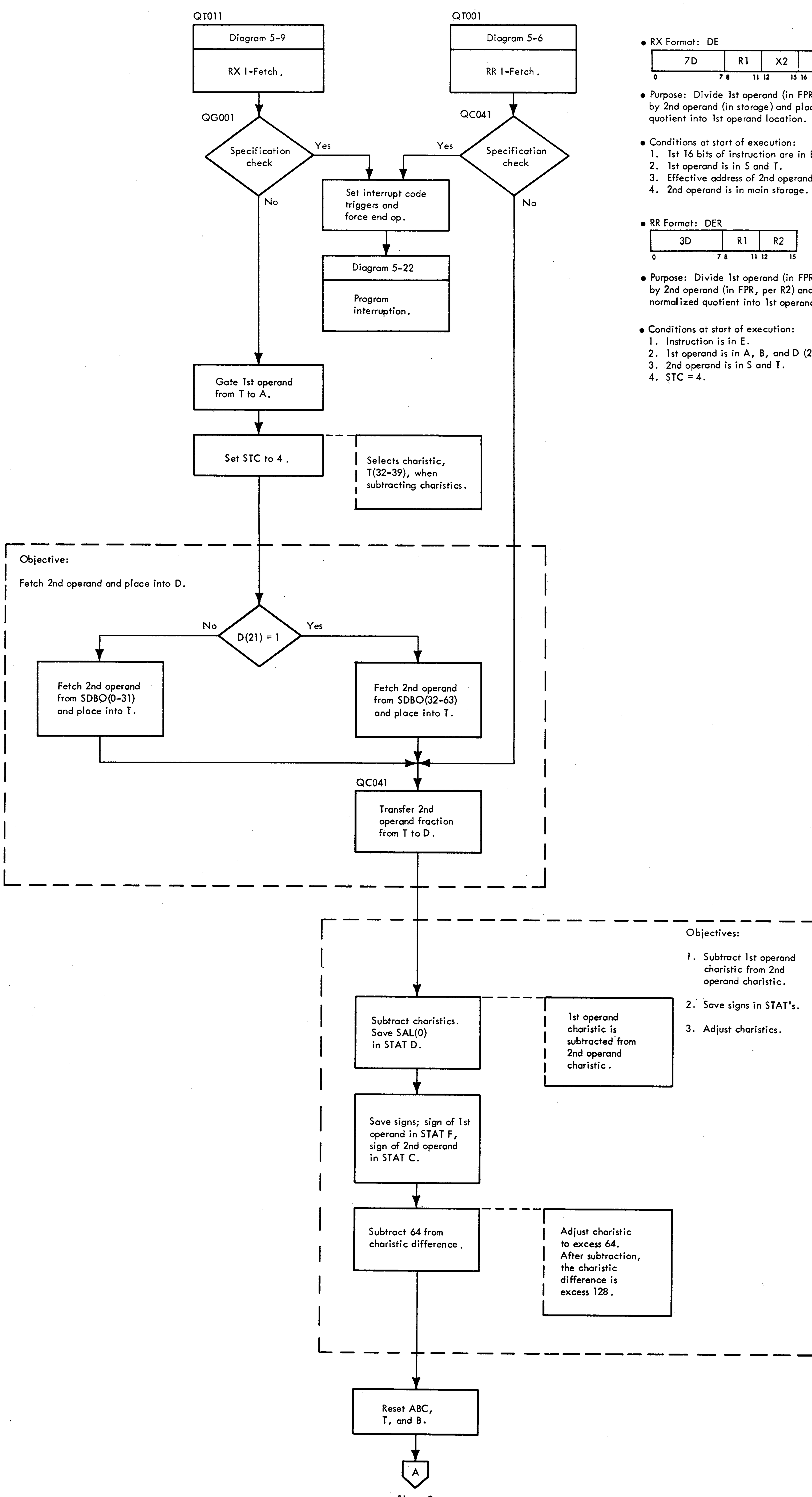


Figure 5-214. Floating-Point Divide, Short Operands (Sheet 2 of 4)

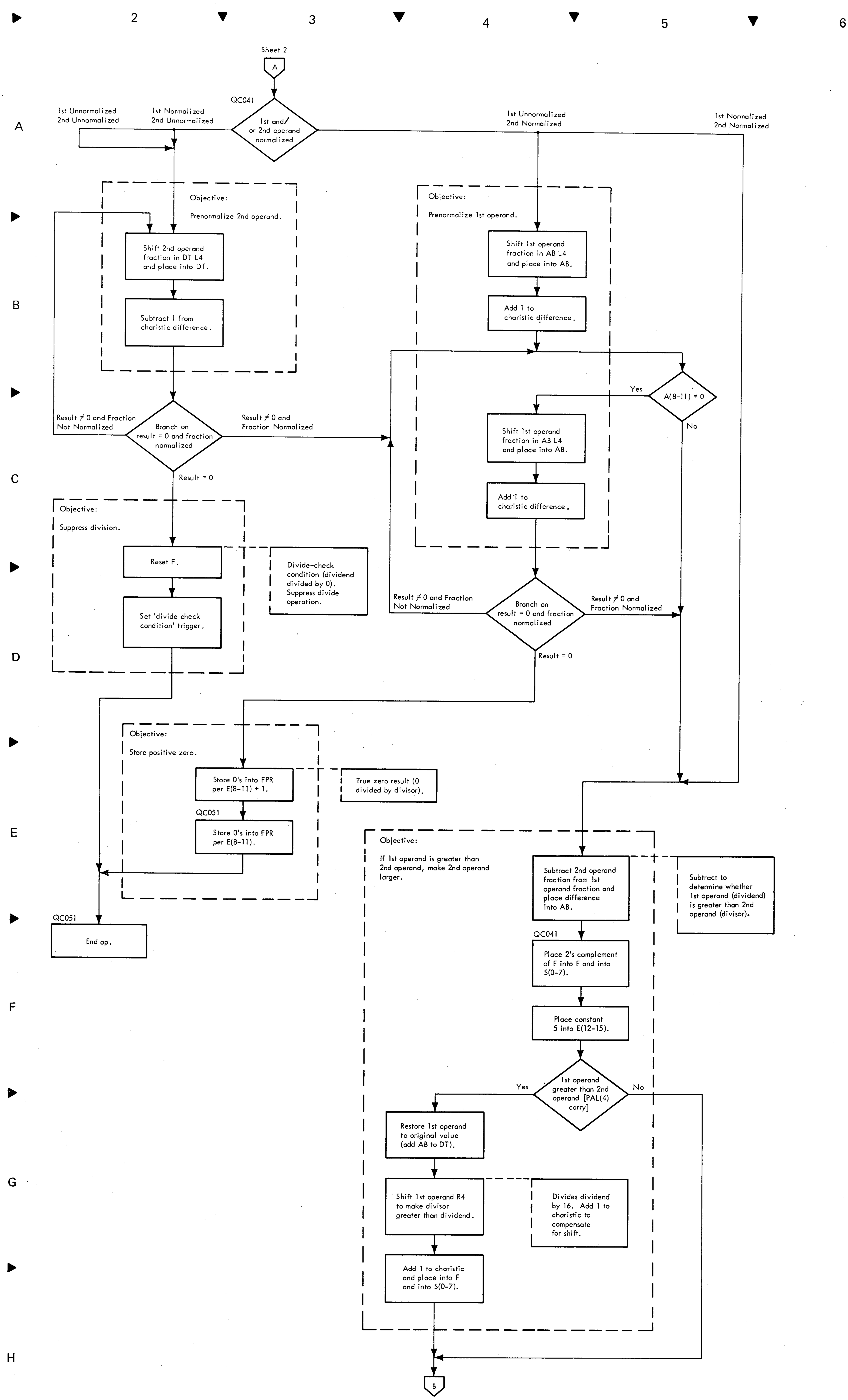


Diagram 5-214. Floating-Point Divide, Short Operands (Sheet 3 of 4)

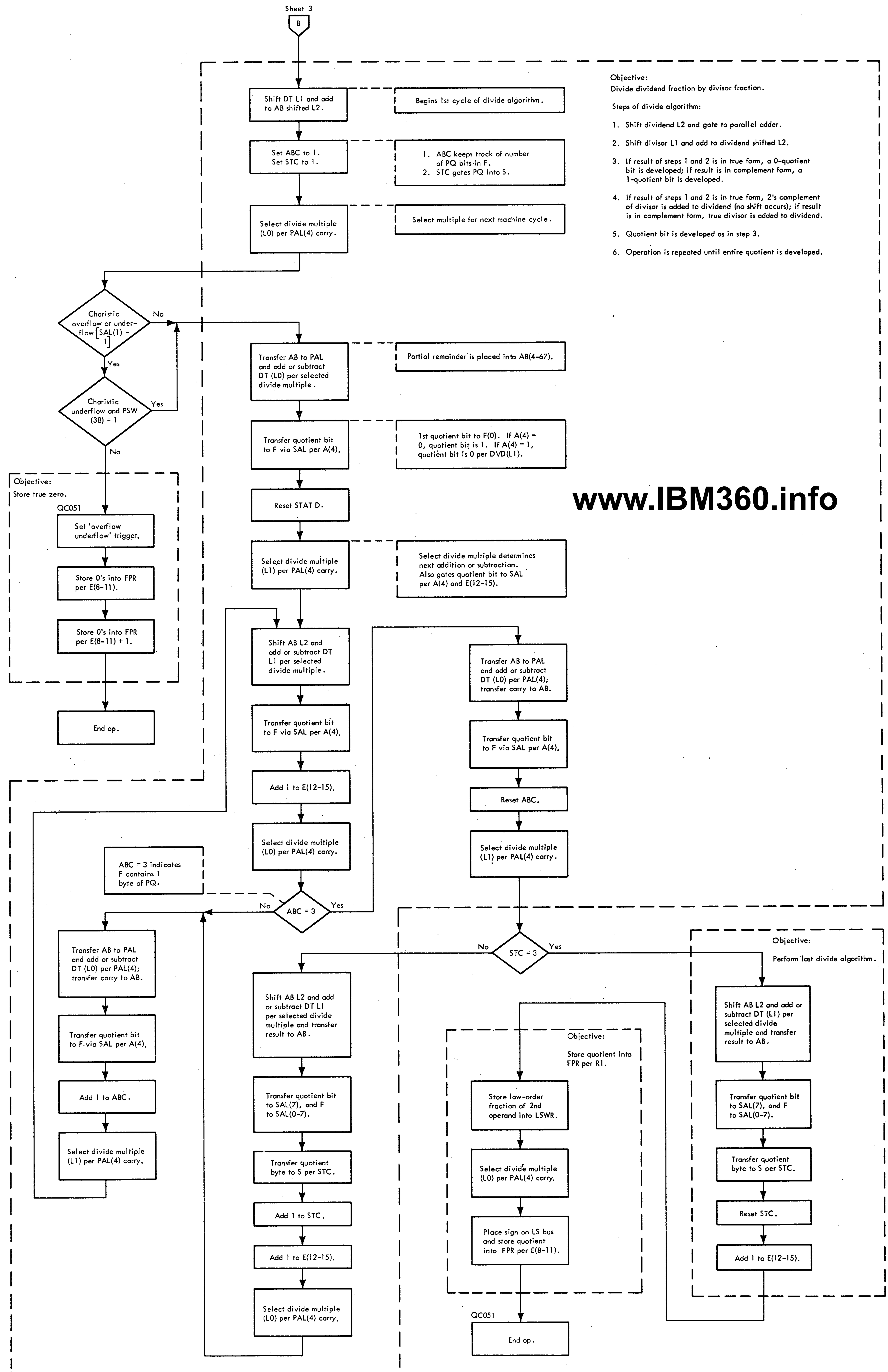


Diagram 5-214. Floating-Point Divide, Short Operands (Sheet 4 of 4)

- RR instruction: DDR

2D	R1	R2
0–7	8–11	12–15

- Purpose: Divide 1st operand (in FPR, per R1 and R1 + 1) by 2nd operand (in FPR, per R2 and R2 + 1) and place normalized quotient into 1st operand location.
- RX instruction: DD

6D	R1	X2	B2	D2
0–7	8–11	12–15	16–19	20–31

- Purpose: Divide 1st operand (in FPR, per R1 and R1 + 1) by 2nd operand (in storage) and place normalized quotient into 1st operand location.

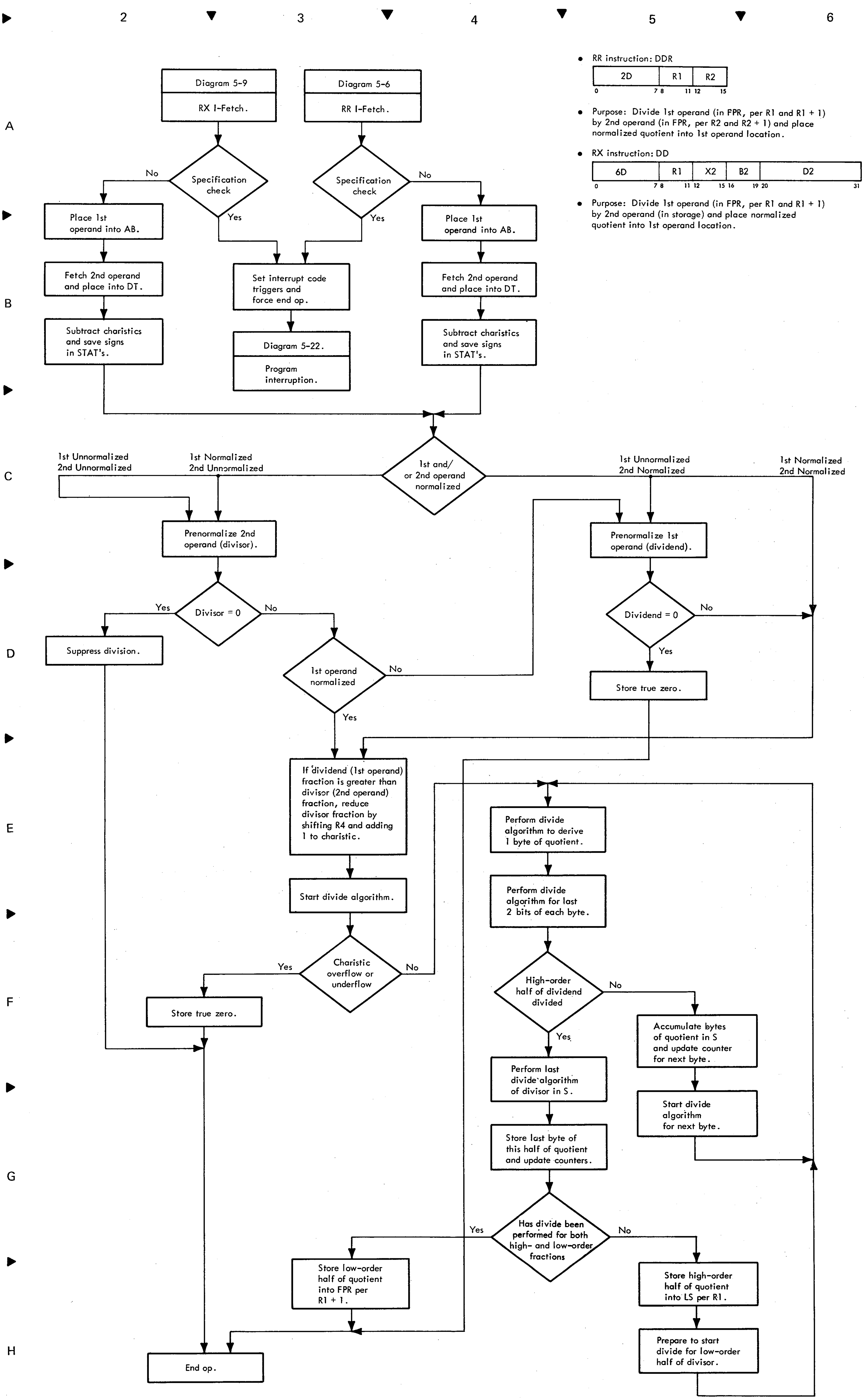


Diagram 5-215. Floating-Point Divide, Long Operands (Sheet 1 of 5)

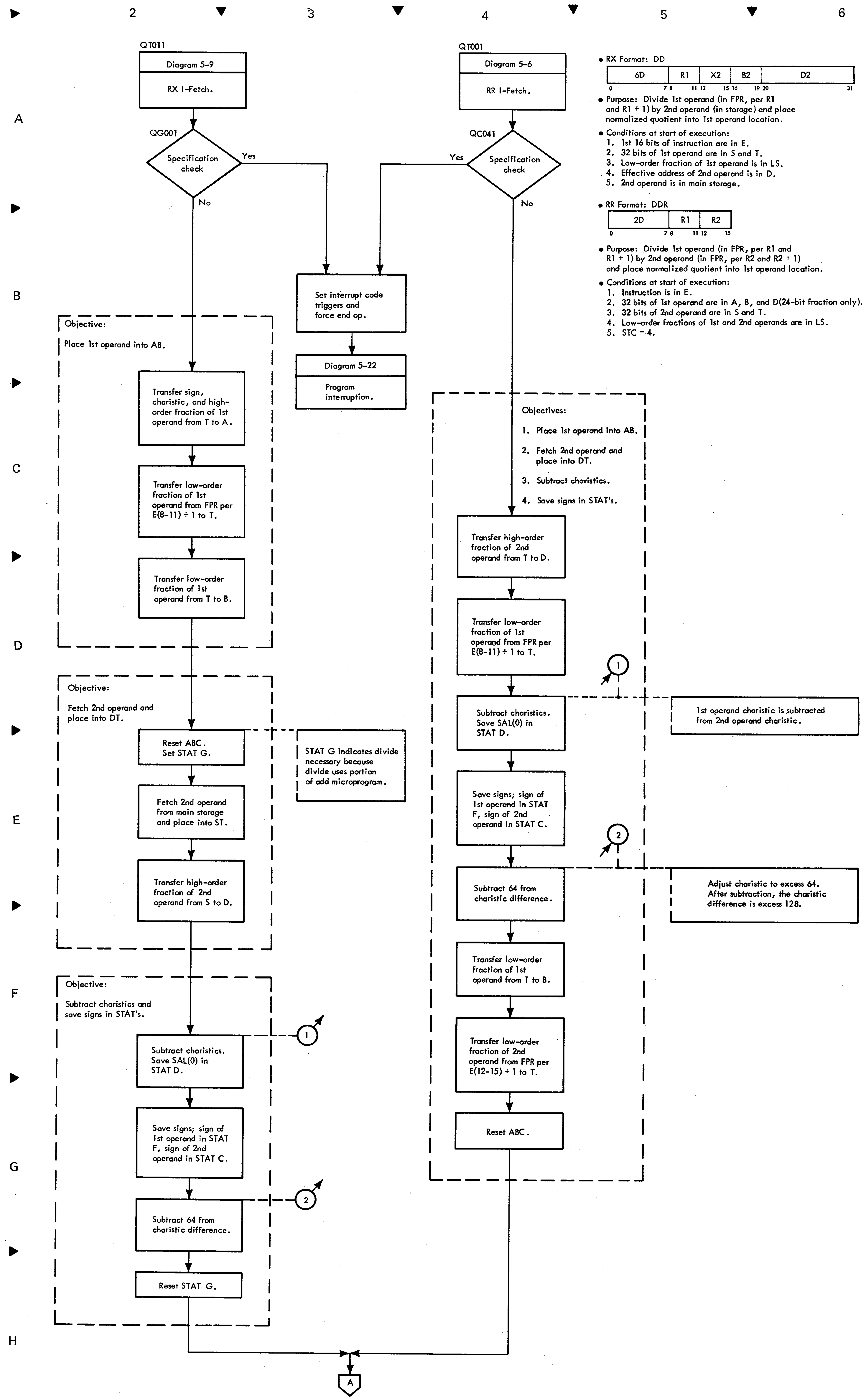


Diagram 5-215. Floating-Point Divide, Long Operands (Sheet 2 of 5)

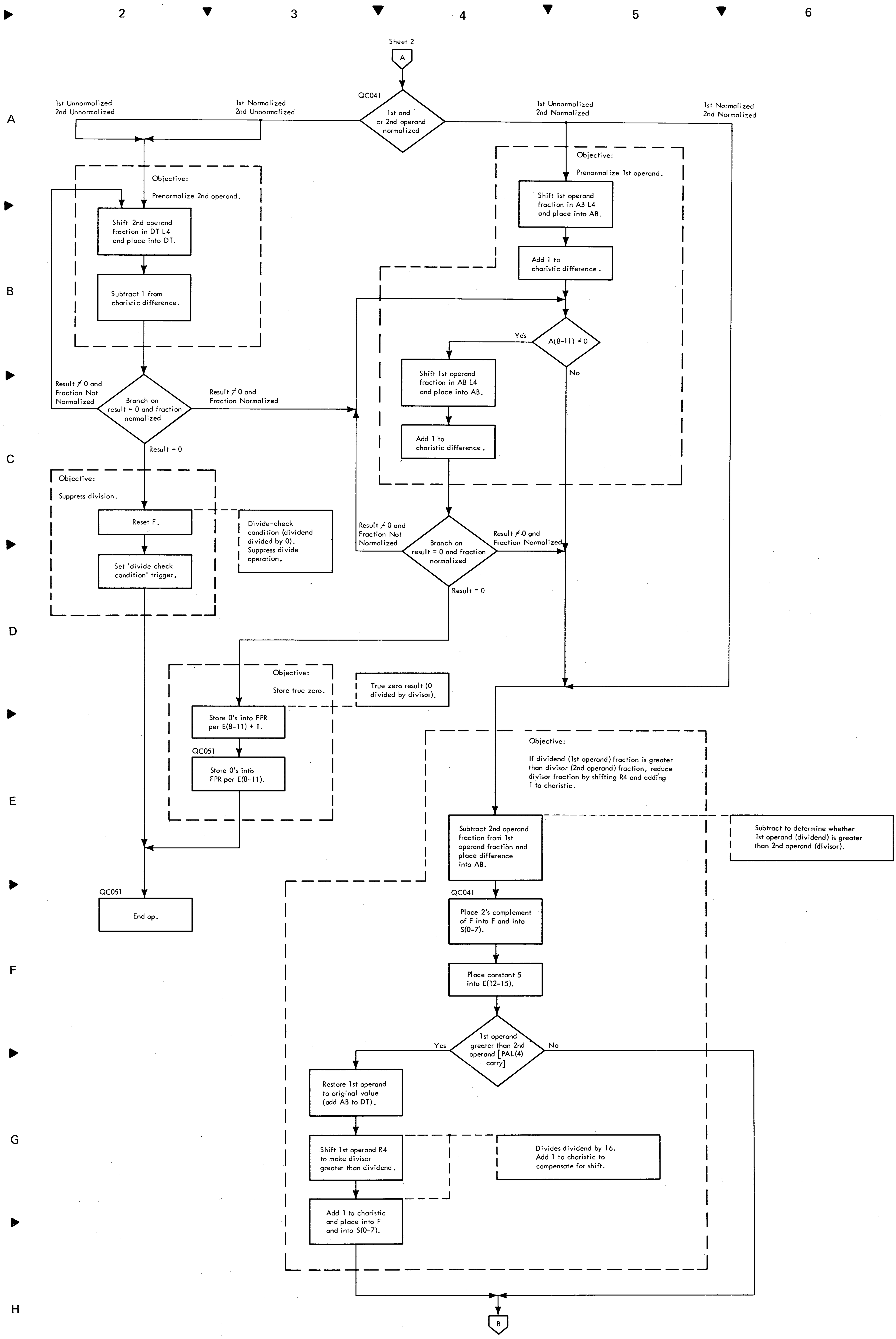


Diagram 5-215. Floating-Point Divide, Long Operands (Sheet 3 of 5)

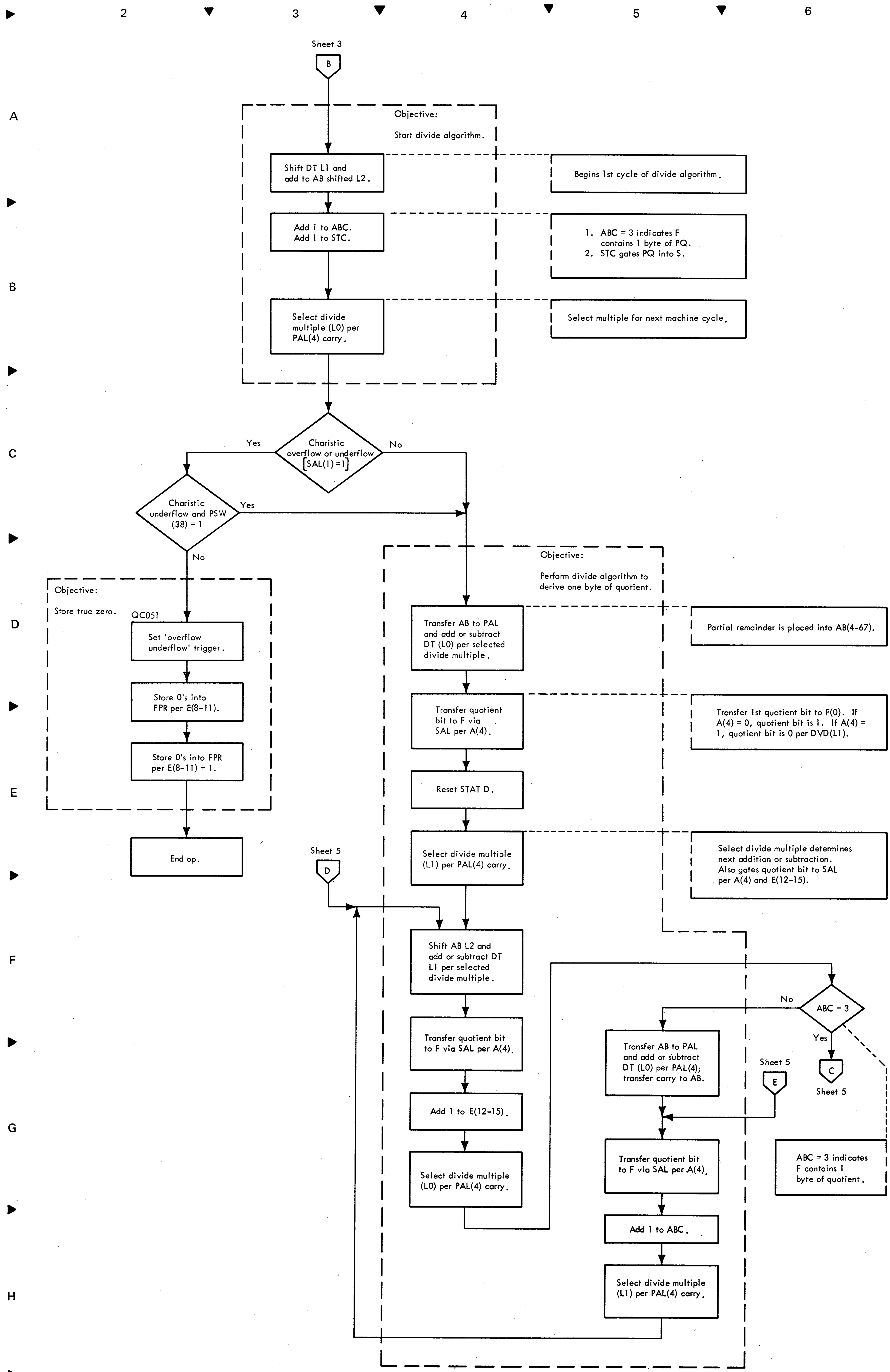


Diagram 5-215. Floating-Point Divide, Long Operands (Sheet 4 of 5)

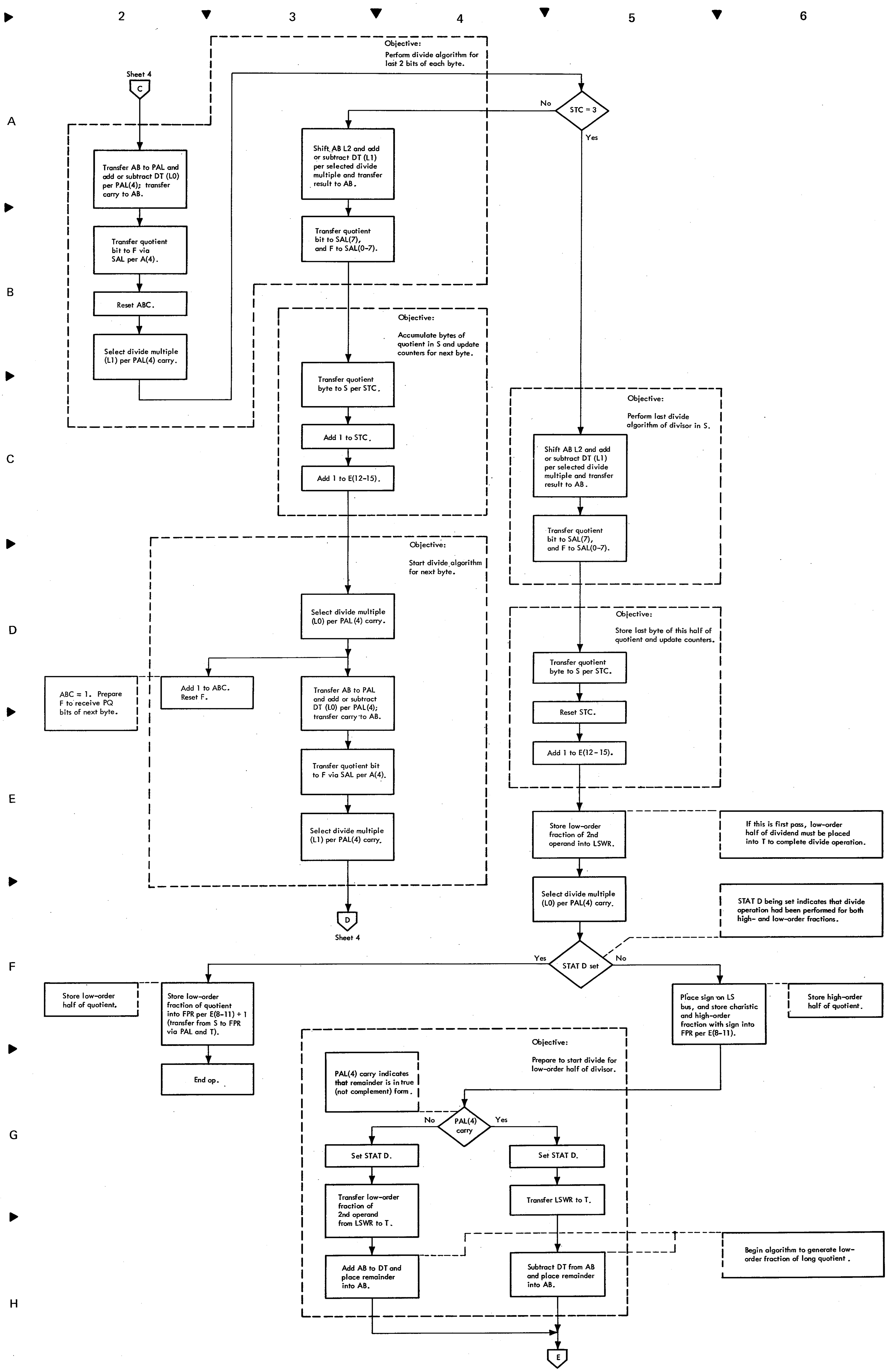


Diagram 5-215. Floating-Point Divide, Long Operands (Sheet 5 of 5)

- RX Format – Long Operands: STD

60	R1	X2	B2	D2
0 7	8 11	12 15	16 19	20 31

- Purpose: Store 1st operand (in FPR, per R1 and R1 + 1) into 2nd operand location (in storage).
- Condition at start of execution:
  1. 1st 16 bits of instruction are in E.
  2. High-order 32 bits of 1st operand are in S and T.
  3. Low-order 32 bits of 1st operand are in LS.
  4. Storage request has been issued for 2nd operand per effective address in D.
- RX Format – Short Operands: STE

70	R1	X2	B2	D2
0 7	8 11	12 15	16 19	20 31

- Purpose: Store 1st operand (in FPR, per R1) into 2nd operand location (in storage).
- Conditions at start of execution:
  1. 1st 16 bits of instruction are in E.
  2. 1st operand is in S and T.
  3. Storage request has been issued for 2nd operand per effective address in D.

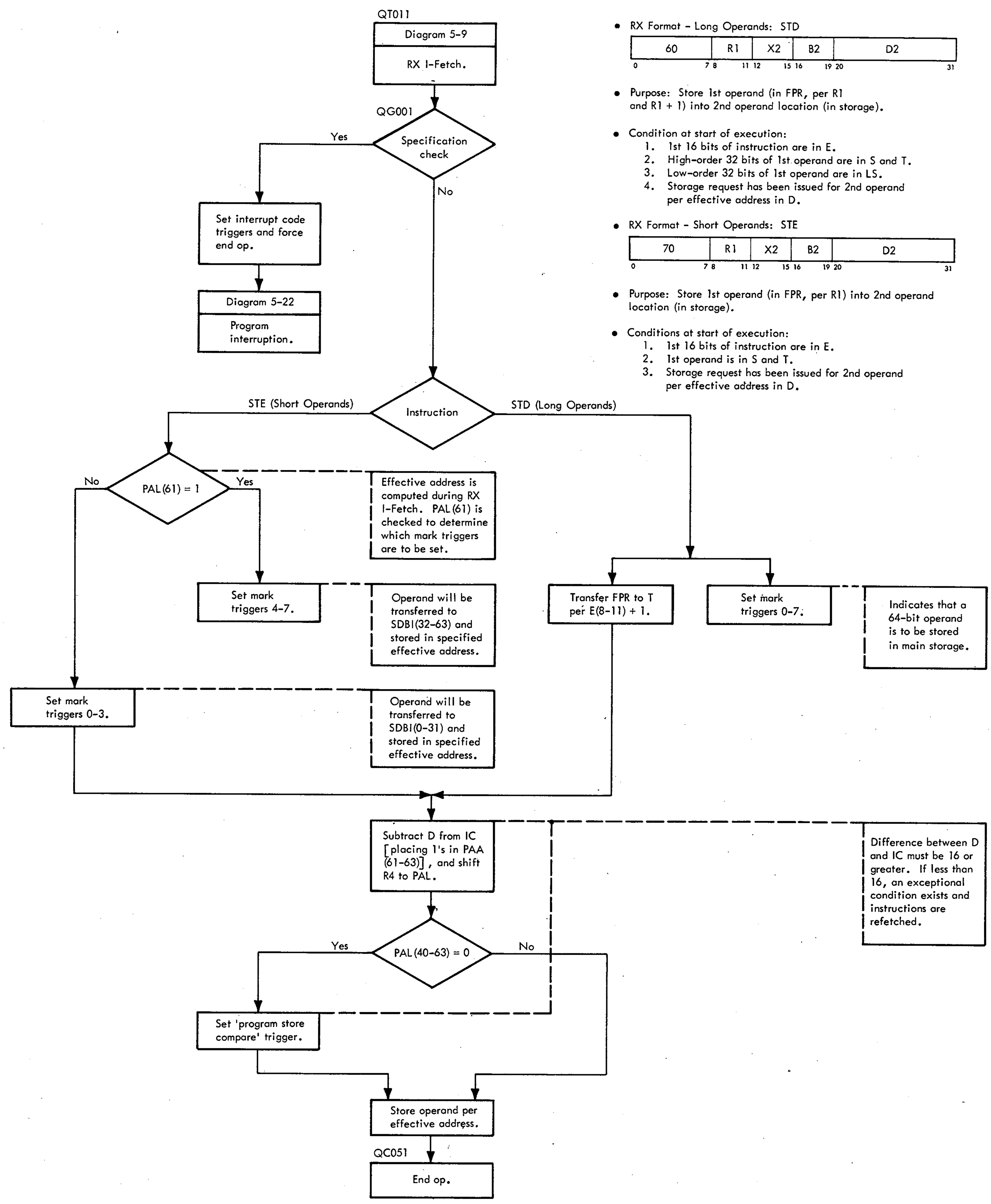
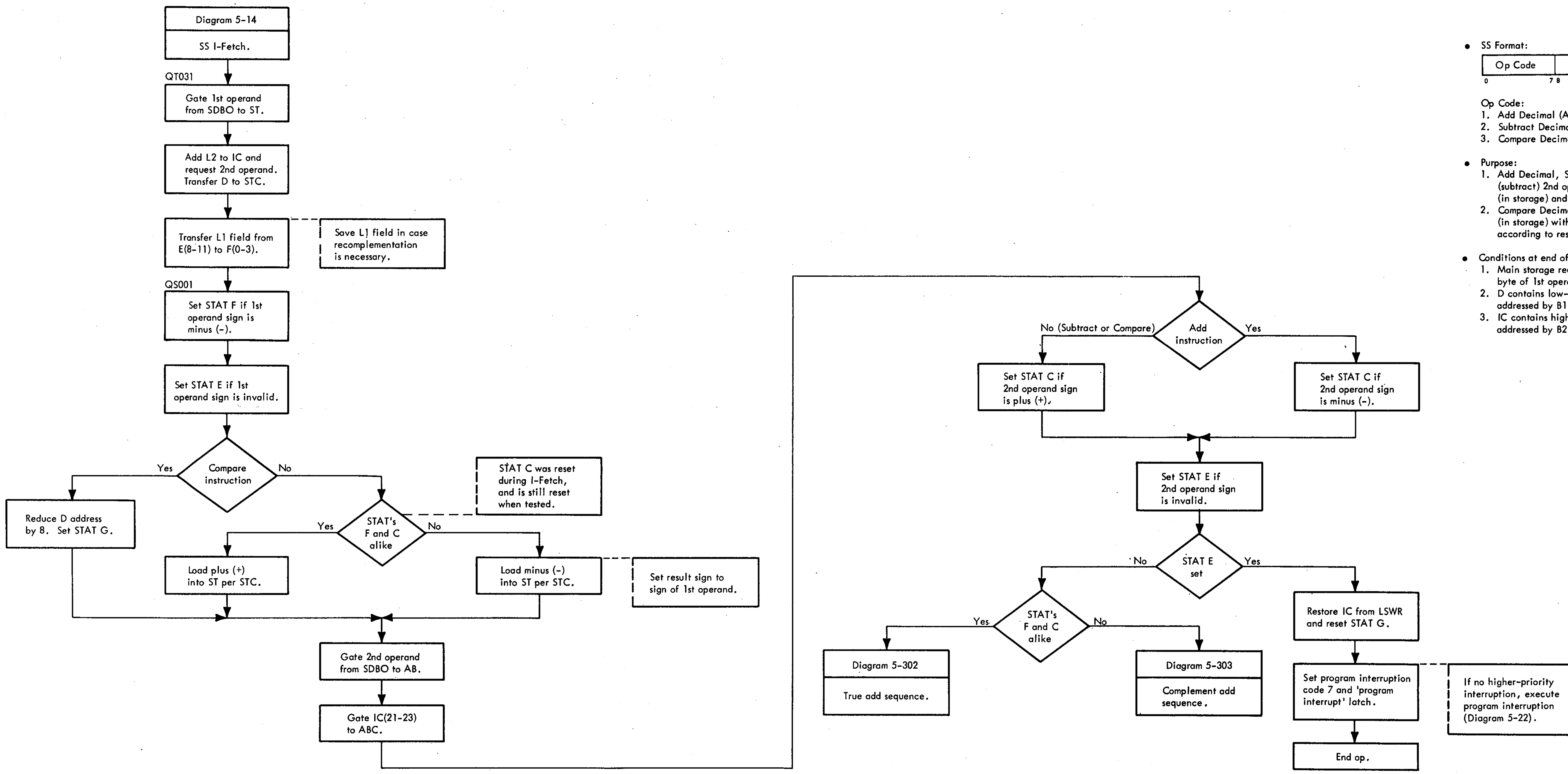


Diagram 5-216. Store, STE (70) – Short Operands; Store, STD (60) – Long Operands



- SS Format:

Op Code	L1	L2	B1	D1	B2	D2
0 7	8 11	12 15	16 19	20 31	32 35	36 47

  Op Code:
  1. Add Decimal (AP) - FA.
  2. Subtract Decimal (SP) - FB.
  3. Compare Decimal (CP) - F9.

- Purpose:
  1. Add Decimal, Subtract Decimal - Algebraically add (subtract) 2nd operand (in storage) to (from) 1st operand (in storage) and place result into 1st operand location.
  2. Compare Decimal - Algebraically compare 1st operand (in storage) with 2nd operand (in storage) and set CC according to result.

- Conditions at end of I-Fetch:
  1. Main storage request for doubleword containing low-order byte of 1st operand (destination) has been issued per D.
  2. D contains low-order byte address (contents of GPR addressed by B1, + D1 + L1) of 1st operand.
  3. IC contains high-order byte address (contents of GPR addressed by B2, + D2) of 2nd operand.

Decimal Add Instruction

Sign of 1st Operand	Sign of 2nd Operand	STAT F	STAT C at 1st Test	STAT F Equals STAT C	Set Sign of Result	STAT C at 2nd Test	STAT F Equals STAT C	Form of 1st Operand to SAB	Form of 2nd Operand to SAA	Form of Result† 1st Operand ≧ 2nd Operand	Form of Result 1st Operand < 2nd Operand
+	+	0	0	Yes	+	0	Yes	True	True +6	True	True
+	-	0	0	Yes	+	1	No	True	2's complement	True	10's complement*
-	+	1	0	No	-	0	No	True	2's complement	True	10's complement*
-	-	1	0	No	-	1	Yes	True	True +6	True	True

Decimal Subtract Instruction

Sign of 1st Operand	Sign of 2nd Operand	STAT F	STAT C at 1st Test	STAT F Equals STAT C	Set Sign of Result	STAT C at 2nd Test	STAT F Equals STAT C	Form of 1st Operand to SAB	Form of 2nd Operand to SAA	Form of Result† 1st Operand ≧ 2nd Operand	Form of Result 1st Operand < 2nd Operand
+	+	0	0	Yes	+	1	No	True	2's complement	True	10's complement*
+	-	0	0	Yes	+	0	Yes	True	True +6	True	True
-	+	1	0	No	-	1	Yes	True	True +6	True	True
-	-	1	0	No	-	0	No	True	2's complement	True	10's complement*

†Result sign set to plus for zero result.

*Result recomplemented and sign inverted.

Diagram 5-301. GIS for Decimal Add, Subtract, and Compare

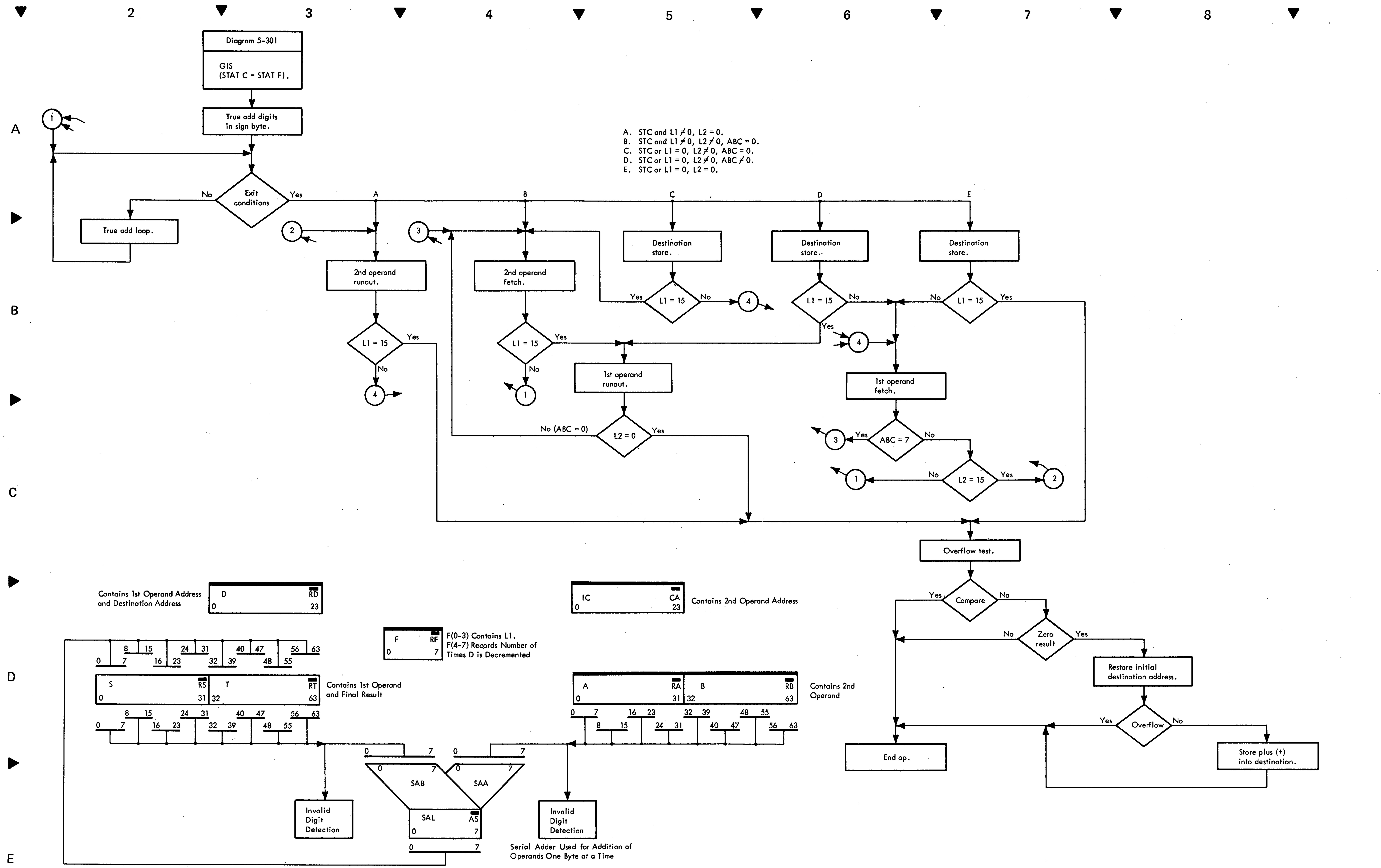


Diagram 5-302. True Add Sequence for Decimal Add, Subtract, and Compare (Sheet 1 of 3)

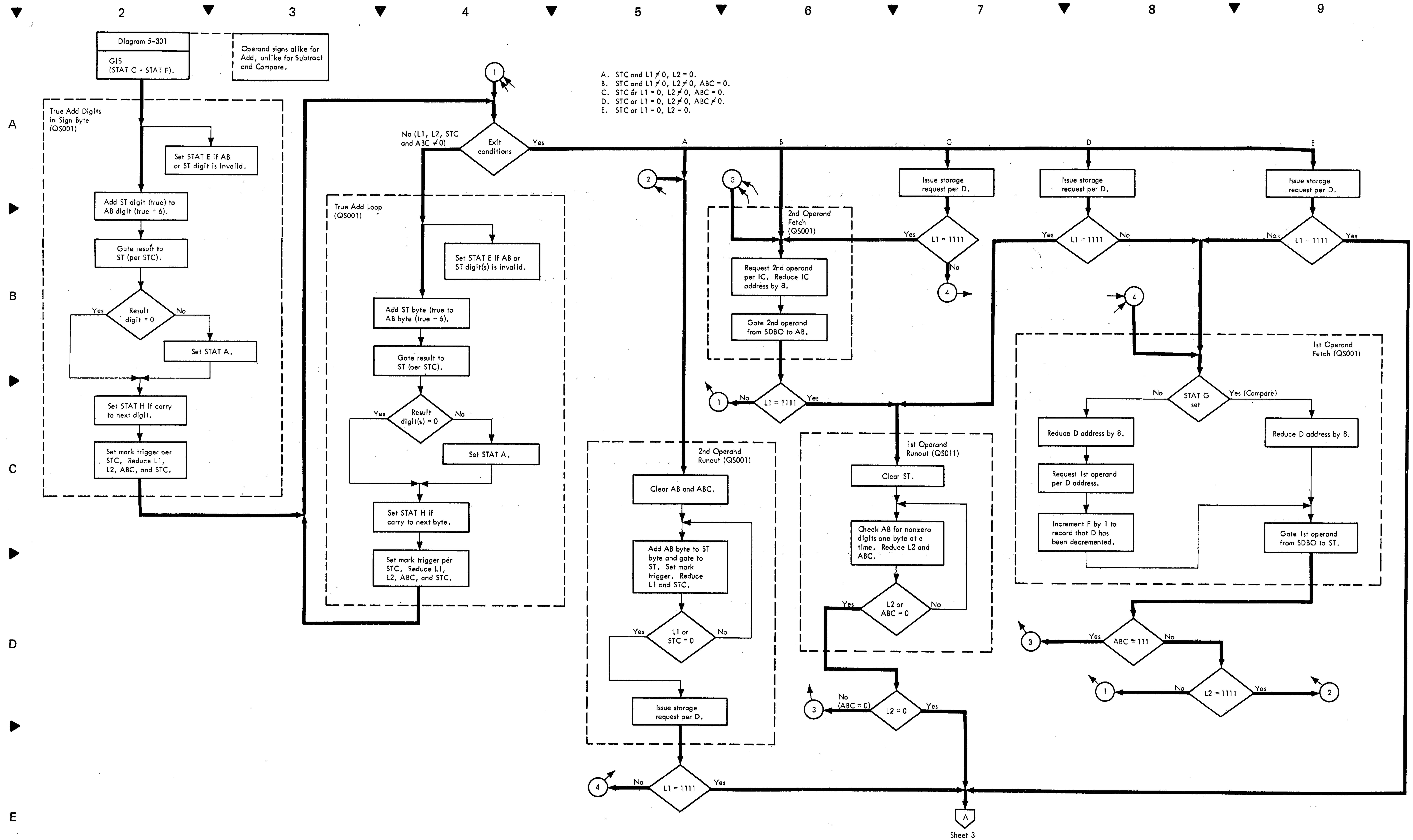
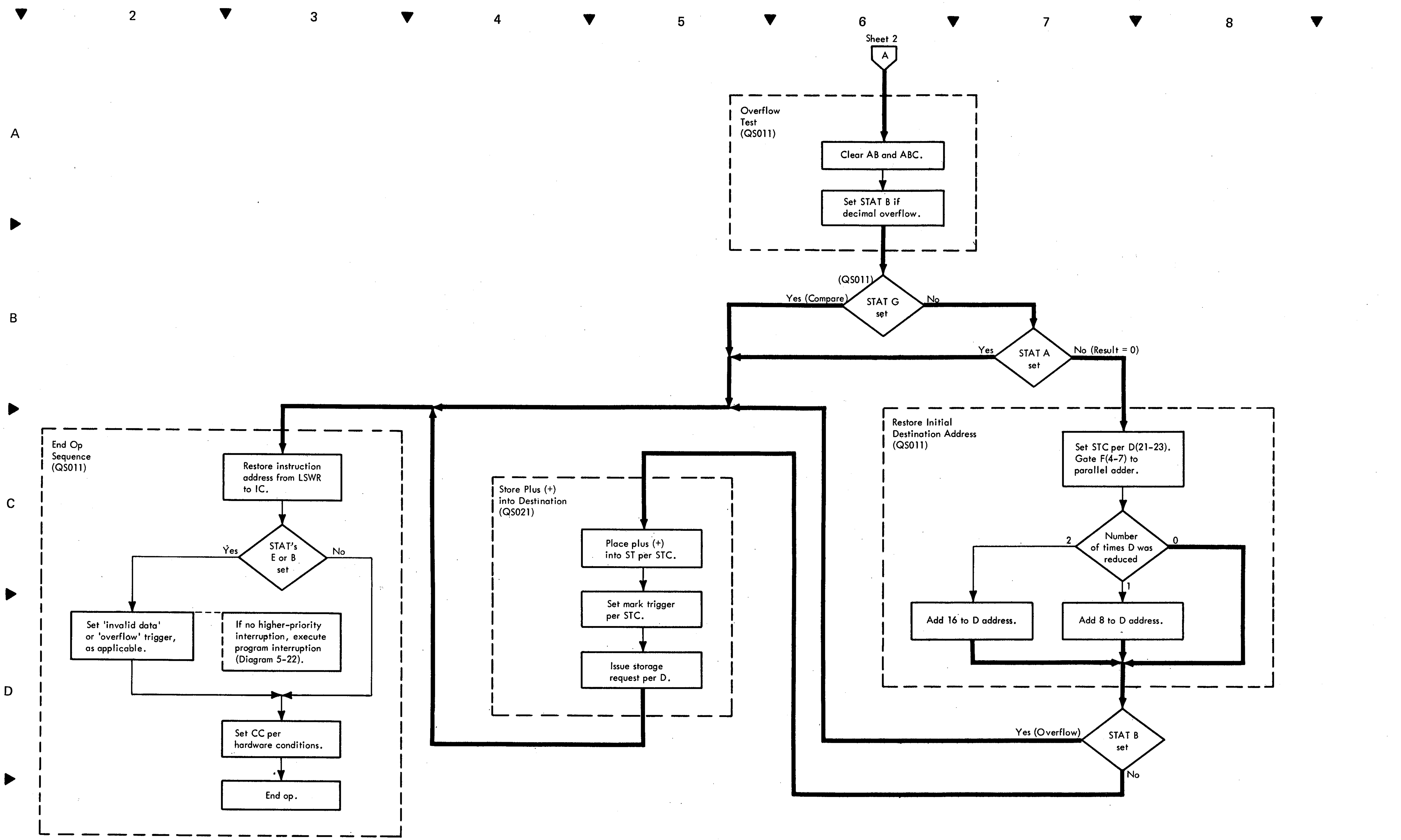


Diagram 5-302. True Add Sequence for Decimal Add, Subtract, and Compare (Sheet 2 of 3)



E

Diagram 5-302. True Add Sequence for Decimal Add, Subtract, and Compare (Sheet 3 of 3)

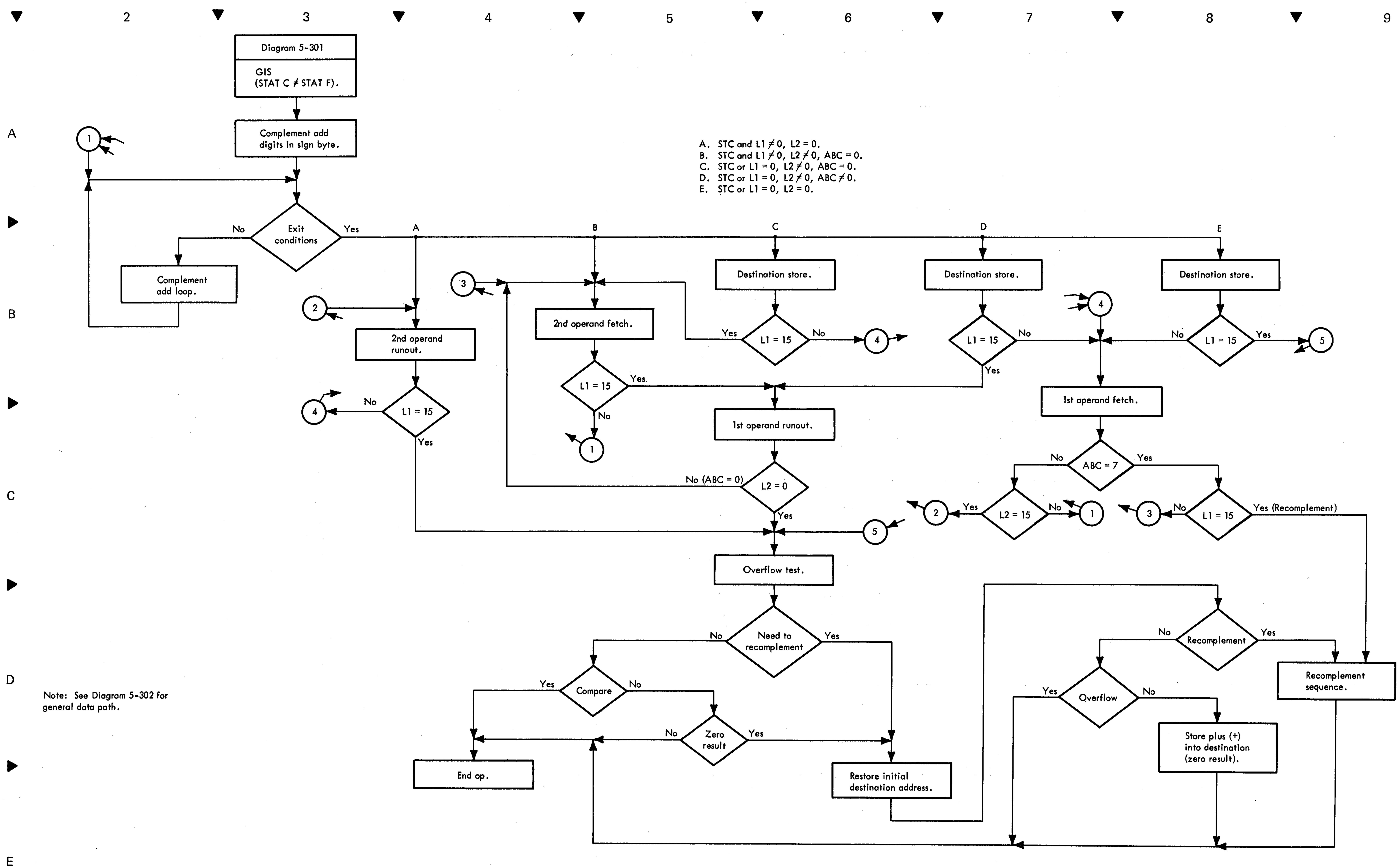


Diagram 5-303. Complement Add Sequence for Decimal Add, Subtract, and Compare (Sheet 1 of 3)

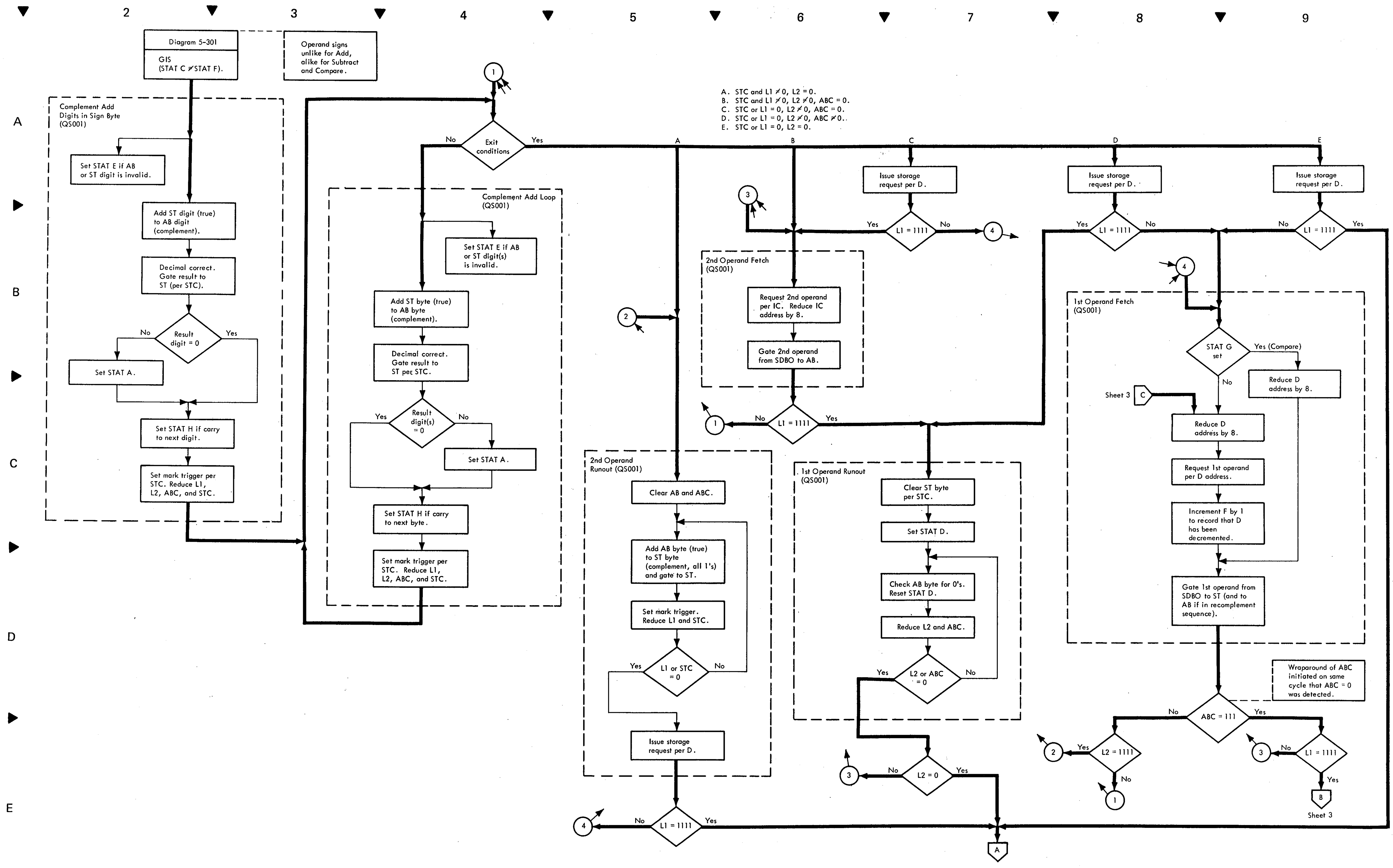


Diagram 5-303. Complement Add Sequence for Decimal Add, Subtract, and Compare (Sheet 2 of 3)

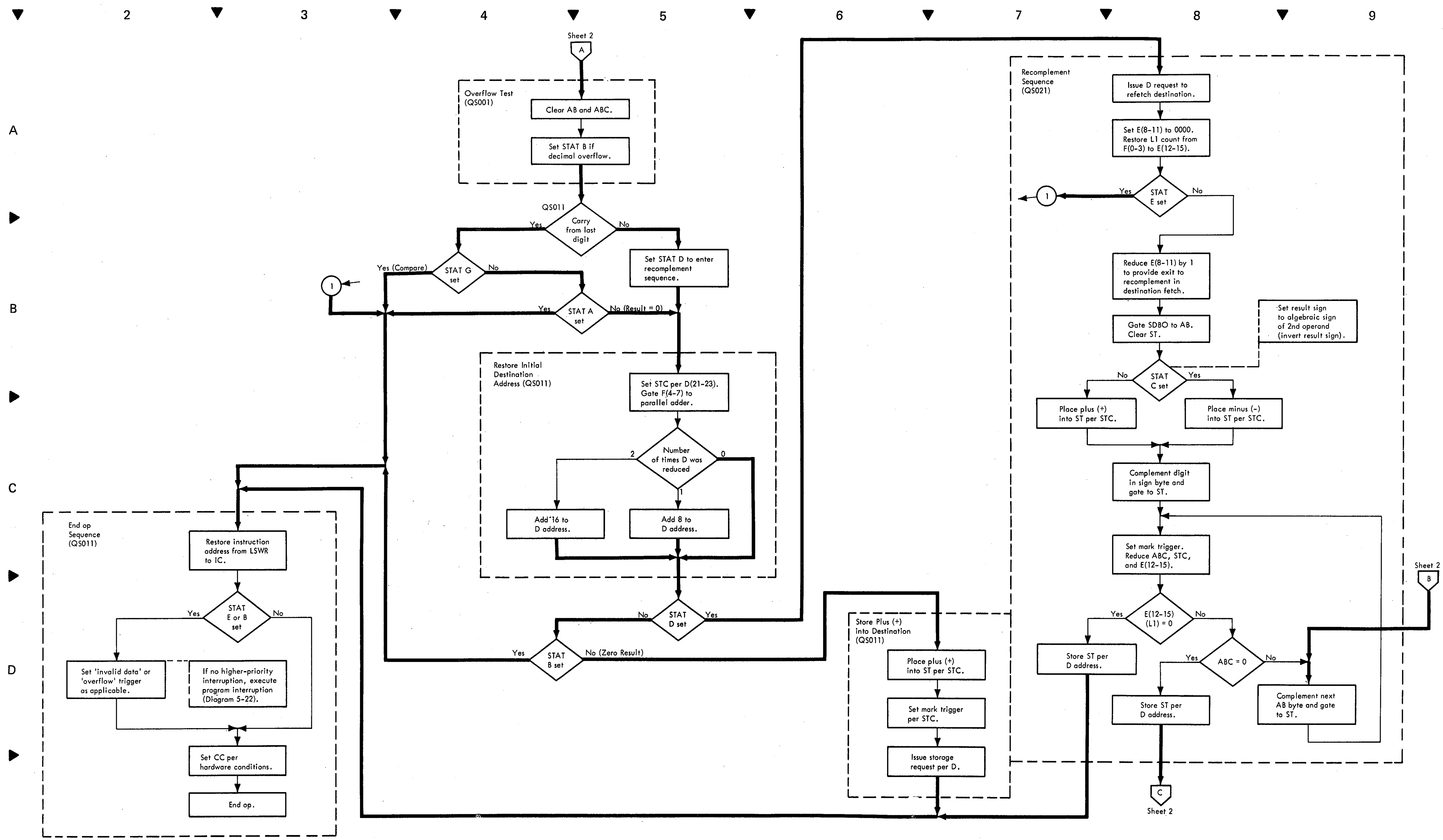


Diagram 5-303. Complement Add Sequence for Decimal Add, Subtract, and Compare (Sheet 3 of 3)

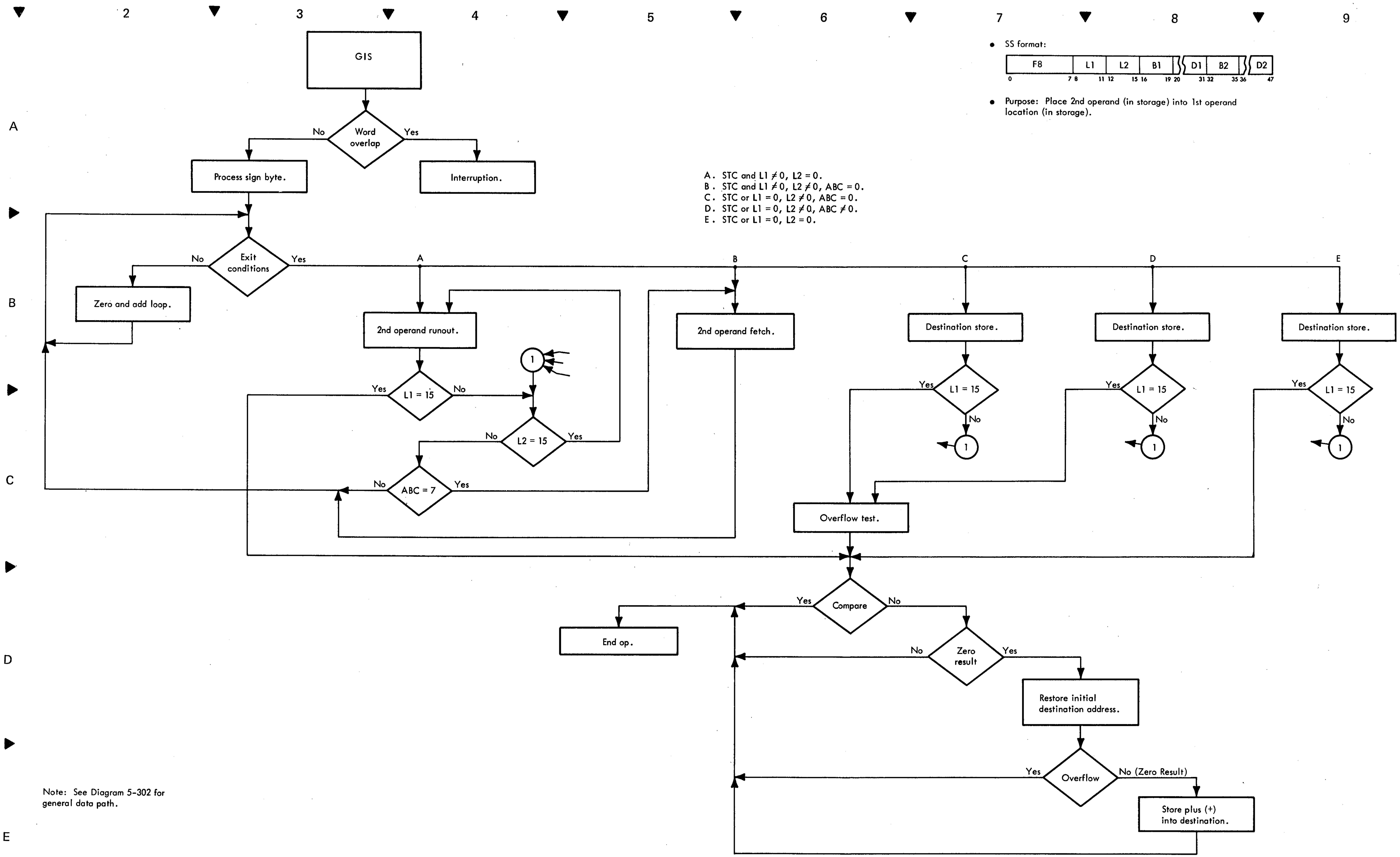
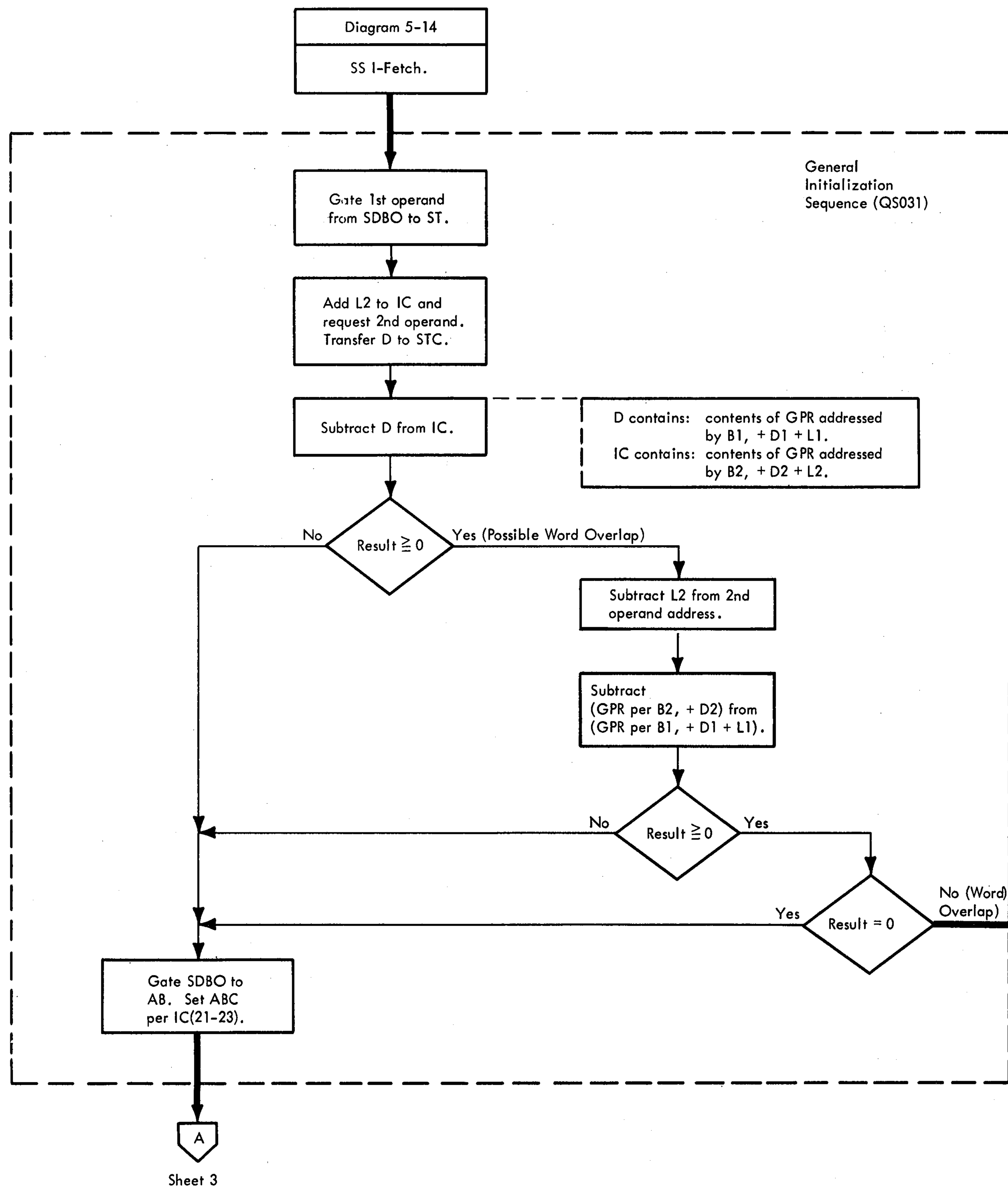
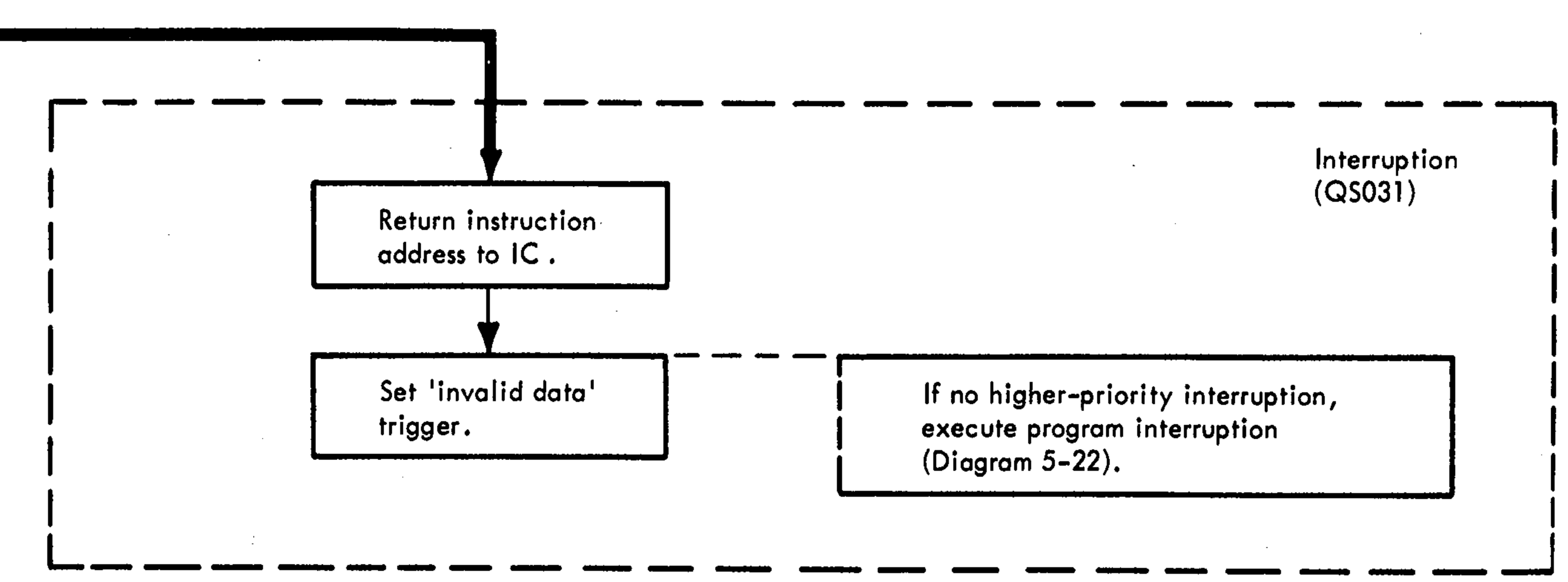


Diagram 5-304. Zero and Add (Sheet 1 of 4)





- SS Format

F8	L1	L2	B1	D1	B2	D2
0 7	8 11	12 15	16 19	20 31	32 35	36 47

- Purpose: Place 2nd operand (in storage) into 1st operand location (in storage).
- Conditions at end of I-Fetch:
  1. Main storage request for doubleword containing low-order byte of 1st operand (destination) has been issued per D.
  2. D contains low-order byte address (contents of GPR addressed by B1, + D1 + L1) of 1st operand.
  3. IC contains high-order byte address (contents of GPR addressed by B2, + D2) of 2nd operand.

Diagram 5-304. Zero and Add (Sheet 2 of 4)

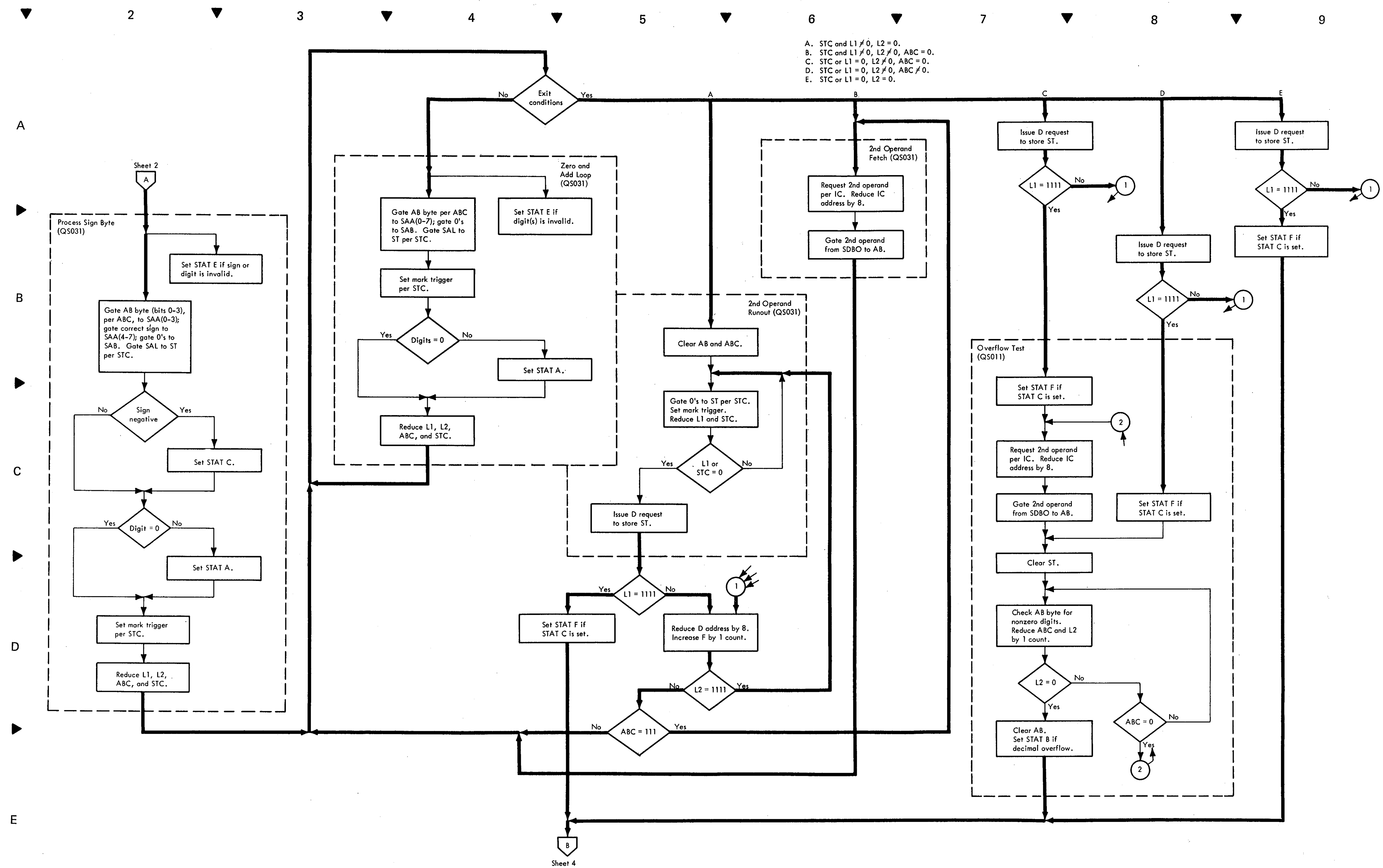


Diagram 5-304. Zero and Add (Sheet 3 of 4)

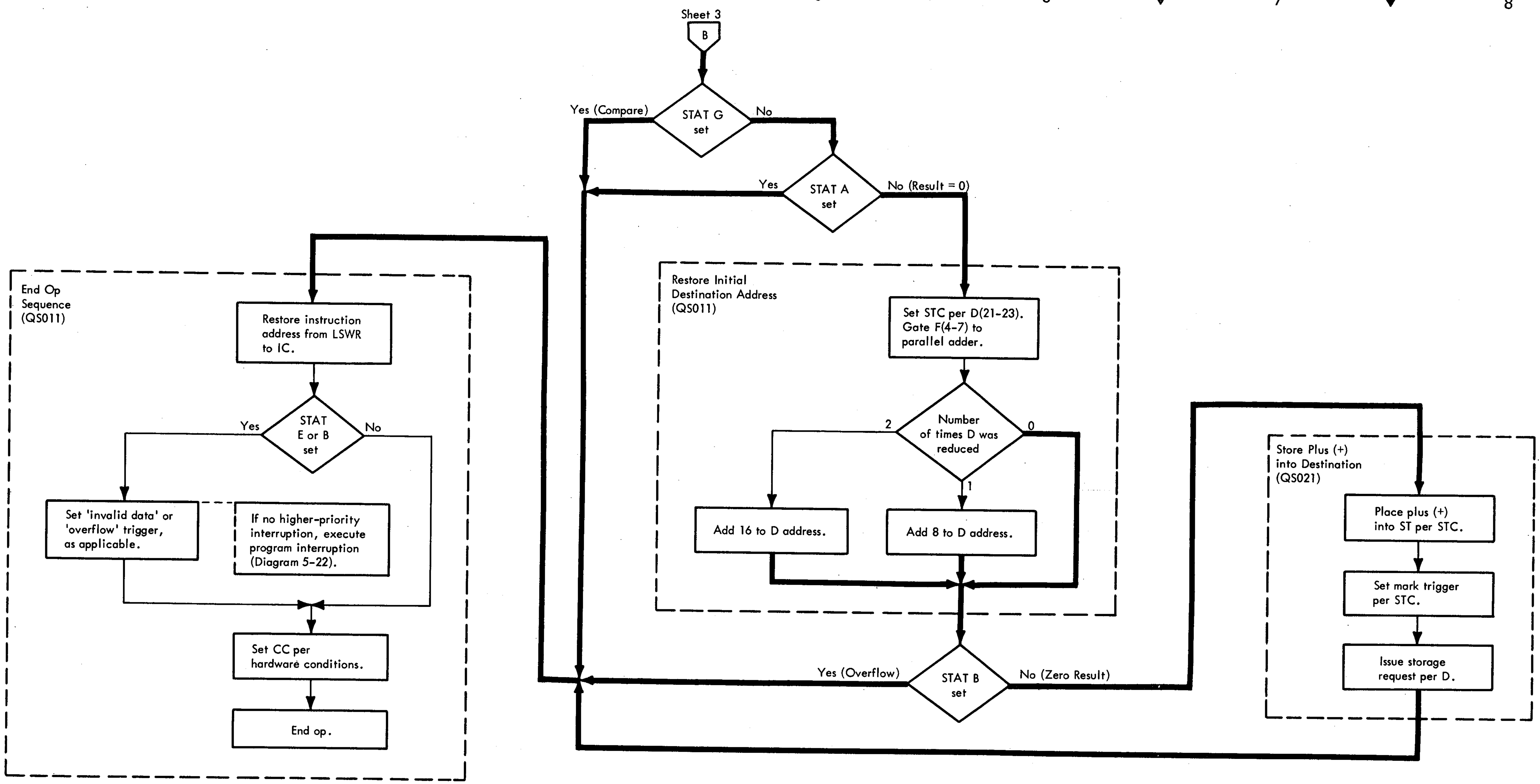
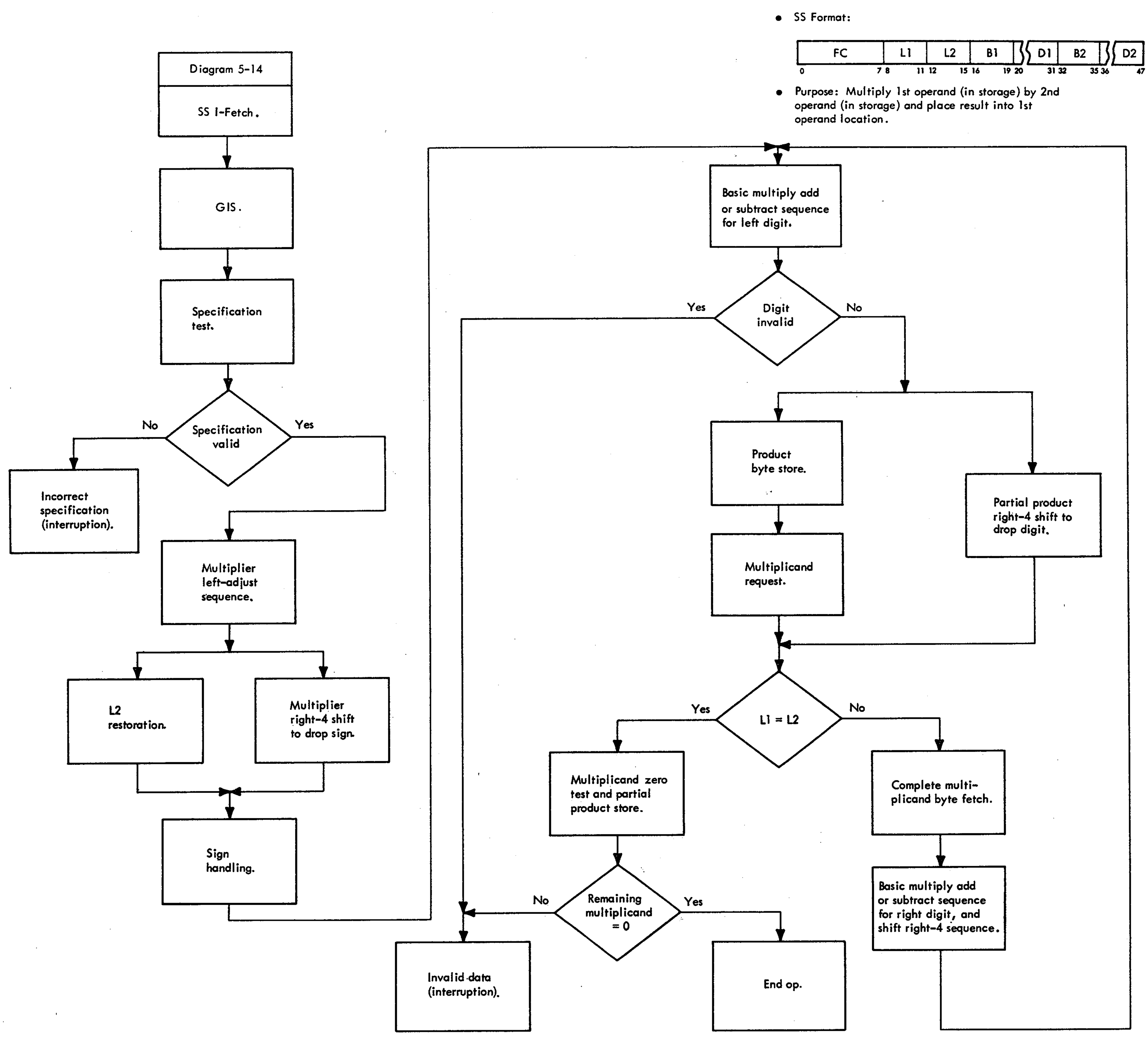
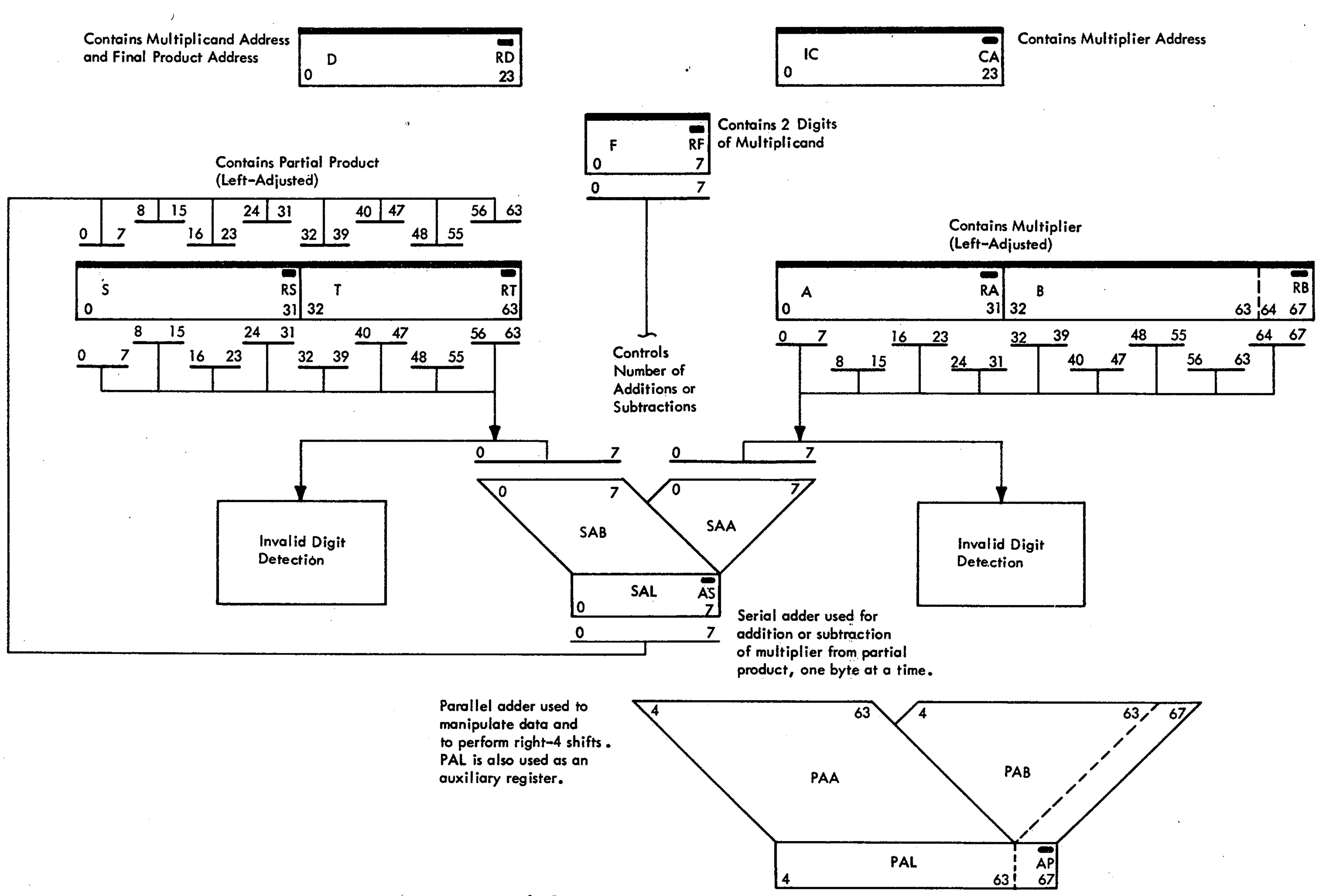


Diagram 5-304. Zero and Add (Sheet 4 of 4)



A) Overall Flow Chart



B) General Data Path

Diagram 5-305. Decimal Multiply (Sheet 1 of 7)

- SS Format:

FC	L1	L2	B1	D1	B2	D2
0–7	8–11	12–15	16–19	20–31	32–35	36–47

- Purpose: Multiply 1st operand (in storage) by 2nd operand (in storage) and place result into 1st operand location.

- Conditions at end of I-Fetch:
  1. Main storage request for doubleword containing low-order byte of 1st operand (multiplicand) has been issued per D.
  2. D contains low-order byte address (contents of GPR addressed by B1, + D1 + L1) of 1st operand.
  3. IC contains high-order byte address (contents of GPR addressed by B2, + D2) of 2nd operand (multiplier).

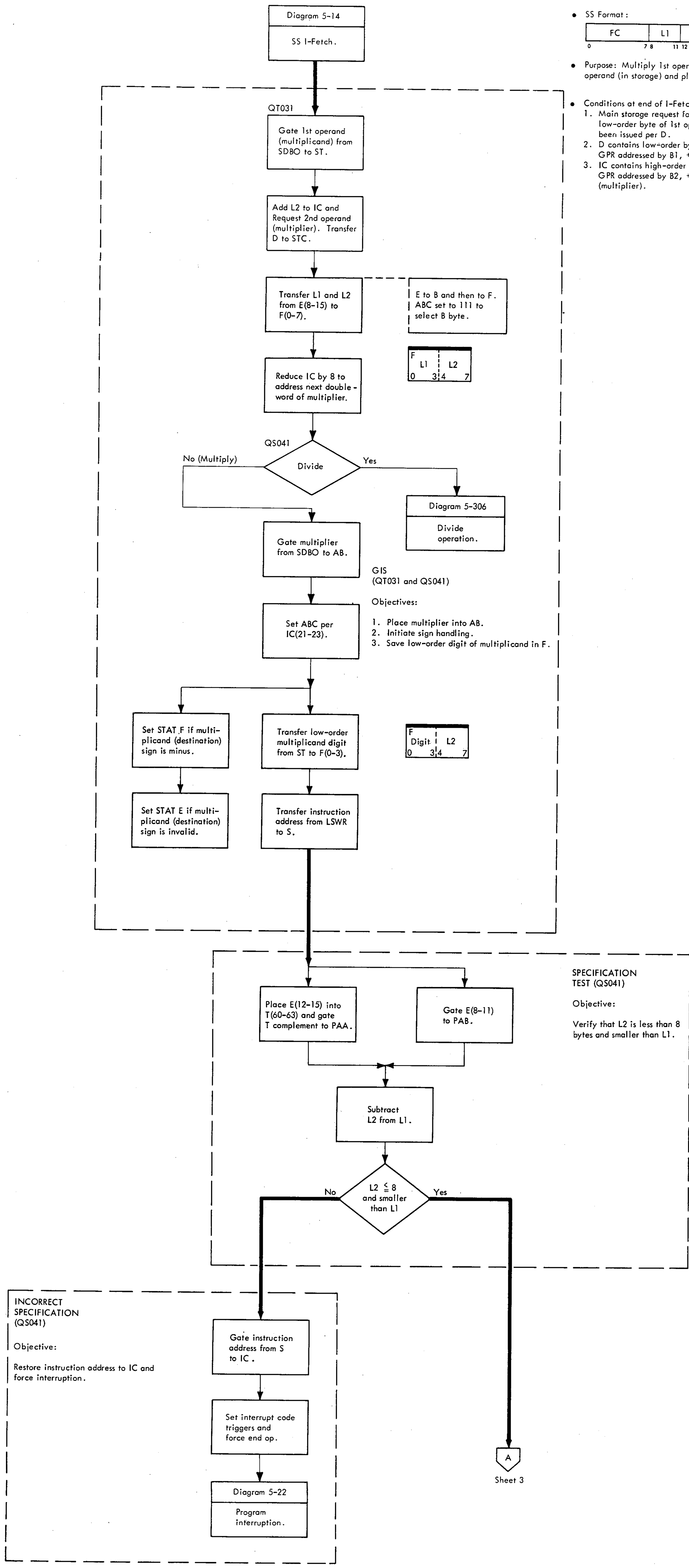


Diagram 5-305. Decimal Multiply (Sheet 2 of 7)

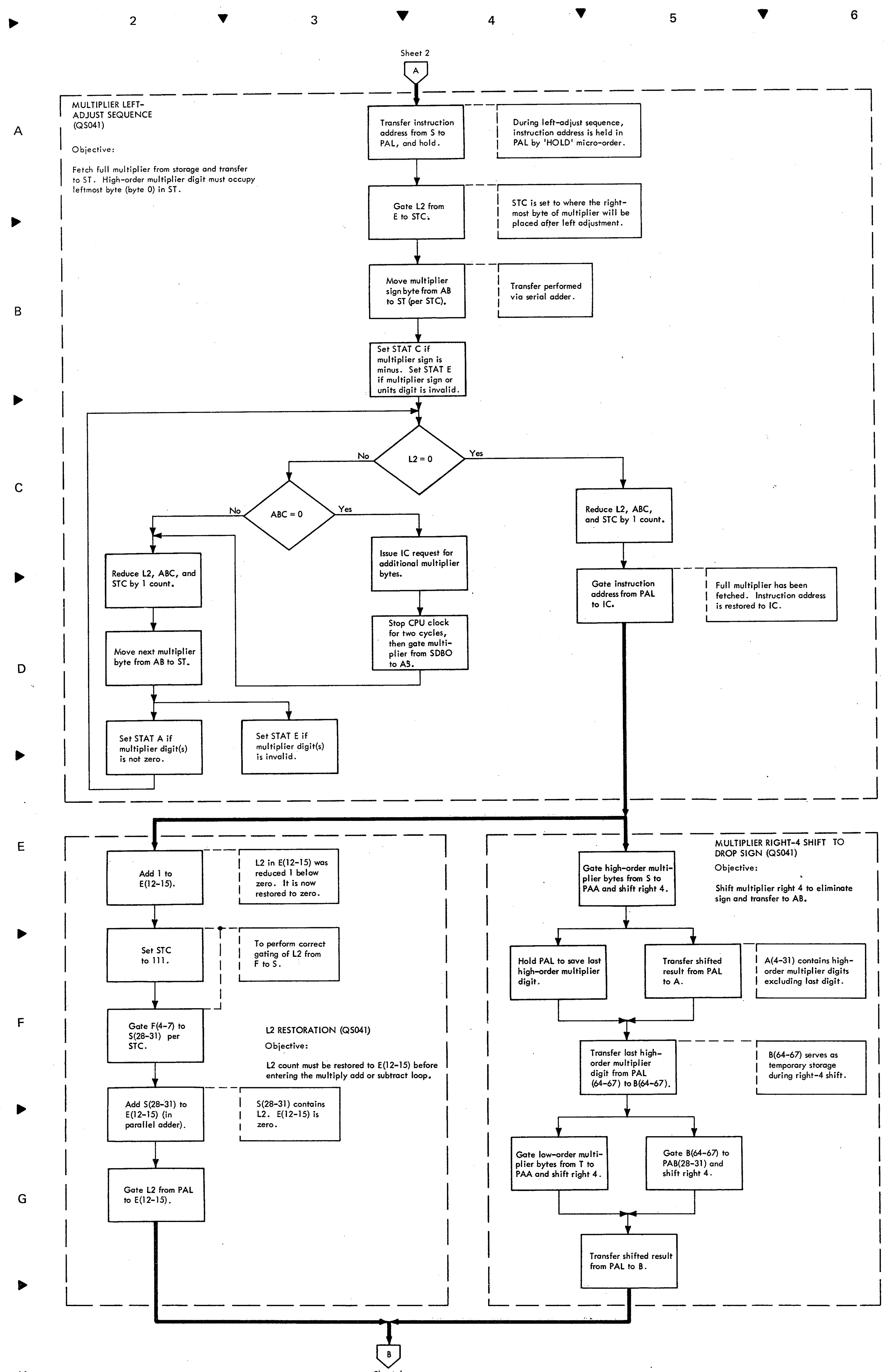


Diagram 5-305. Decimal Multiply (Sheet 3 of 7)

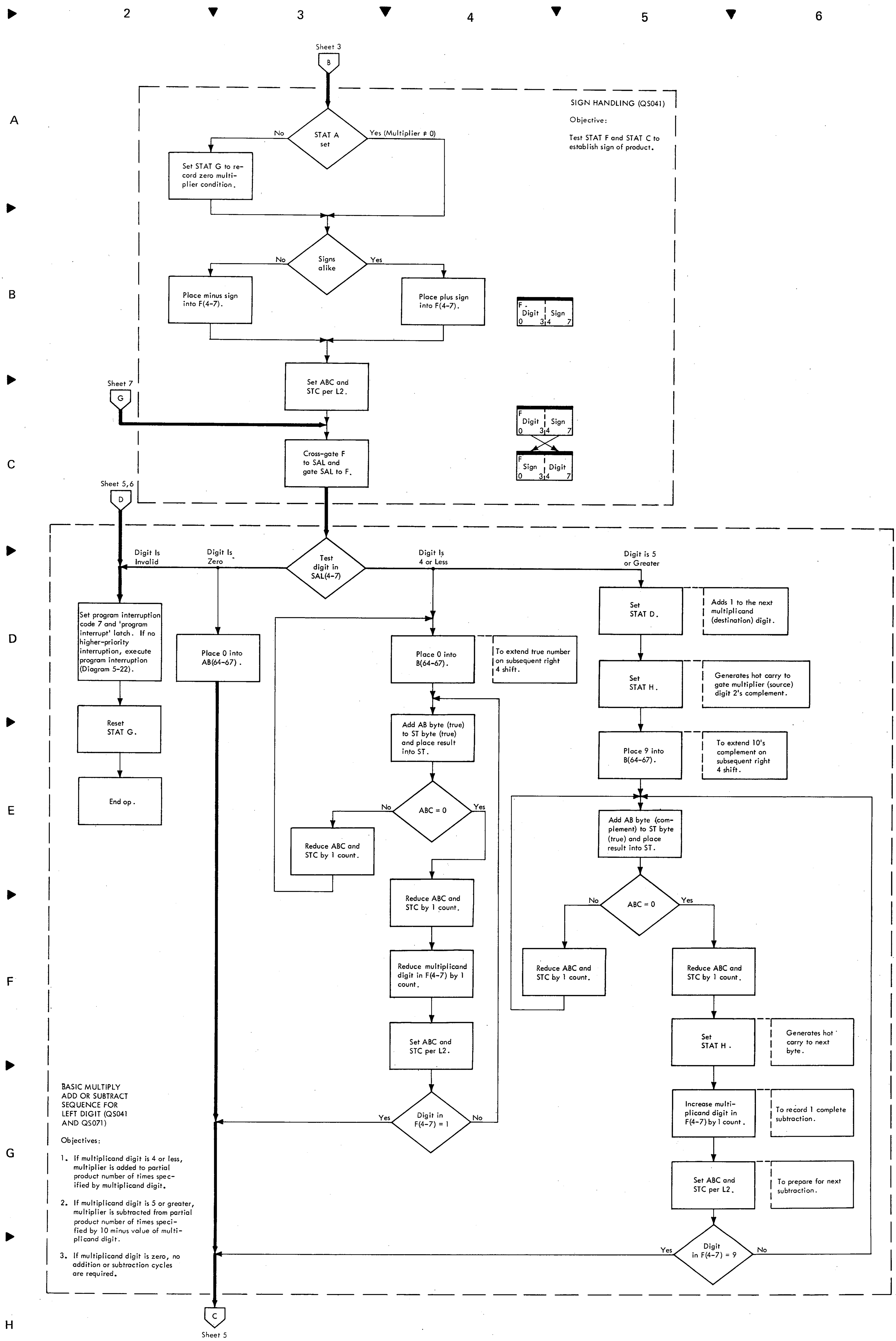


Diagram 5-305. Decimal Multiply (Sheet 4 of 7)

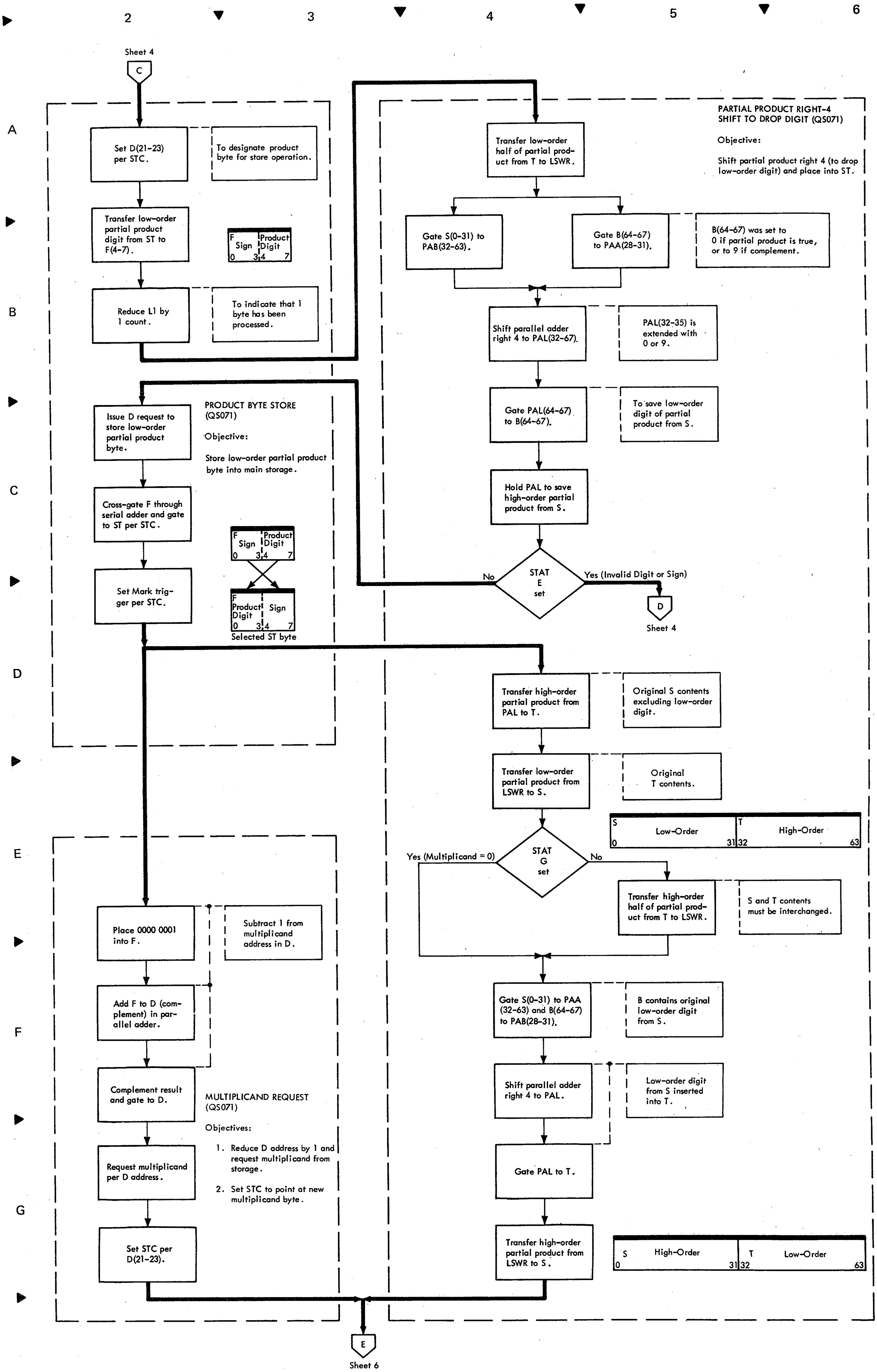


Diagram 5-305. Decimal Multiply (Sheet 5 of 7)

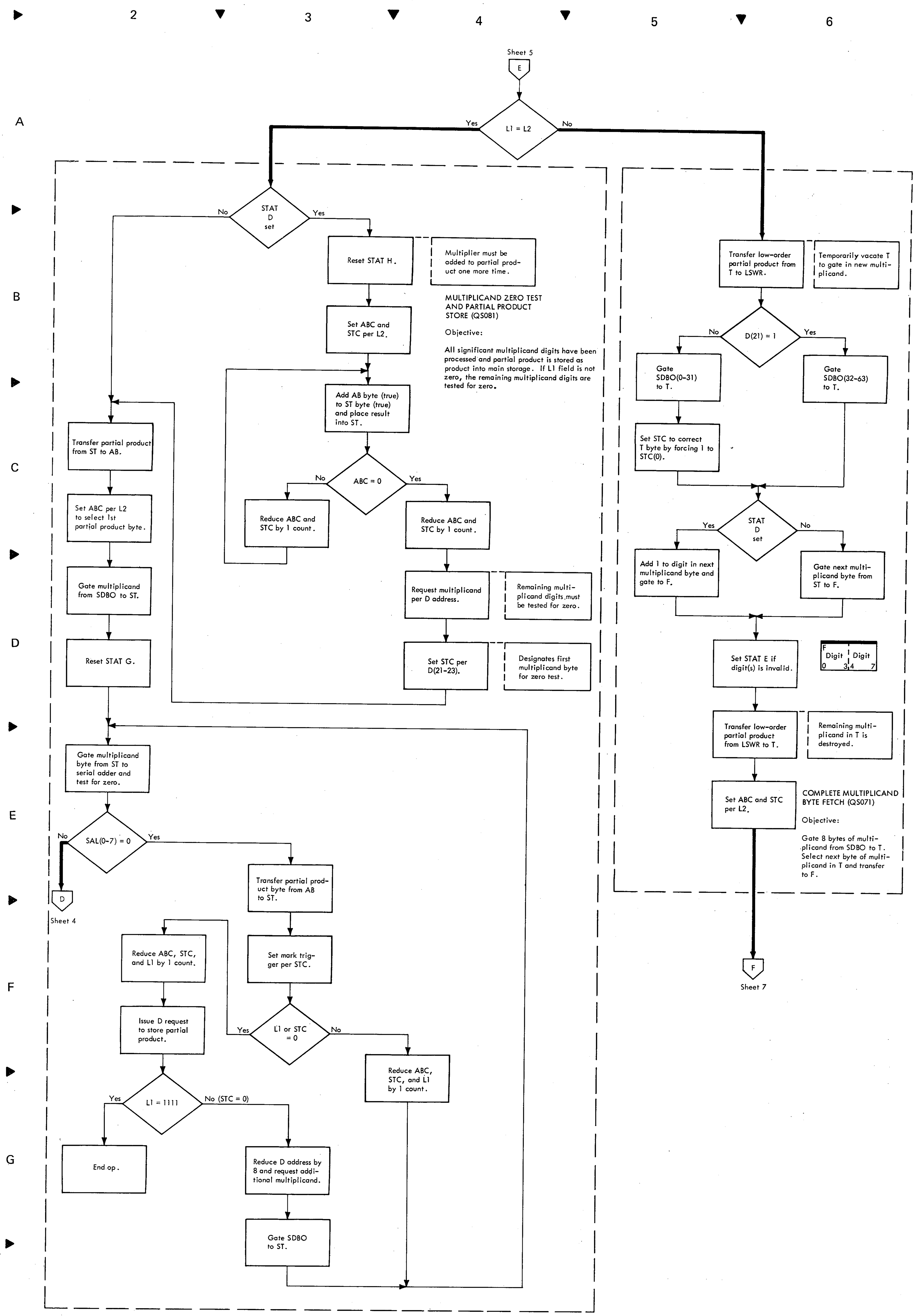
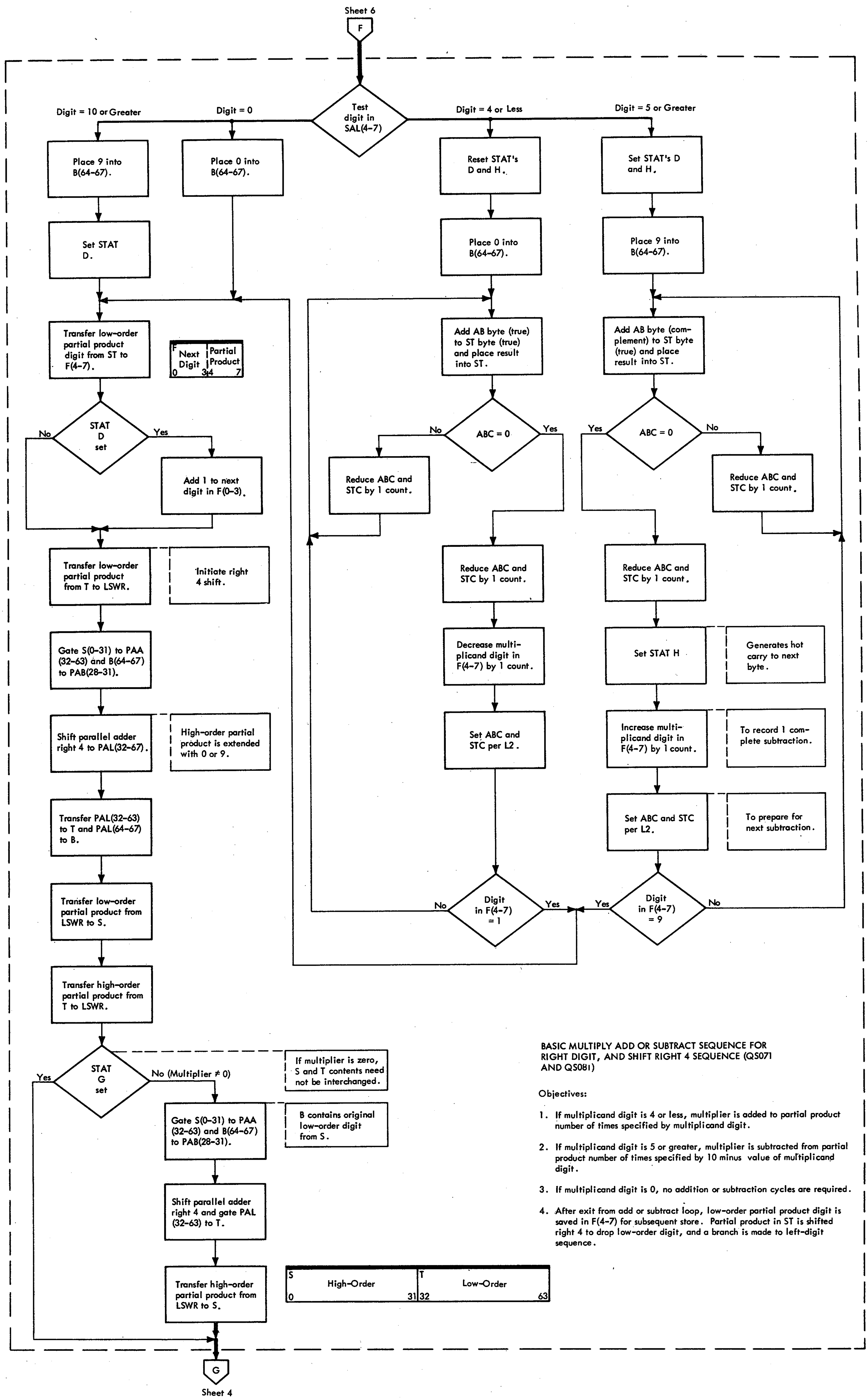


Diagram 5-305. Decimal Multiply (Sheet 6 of 7)

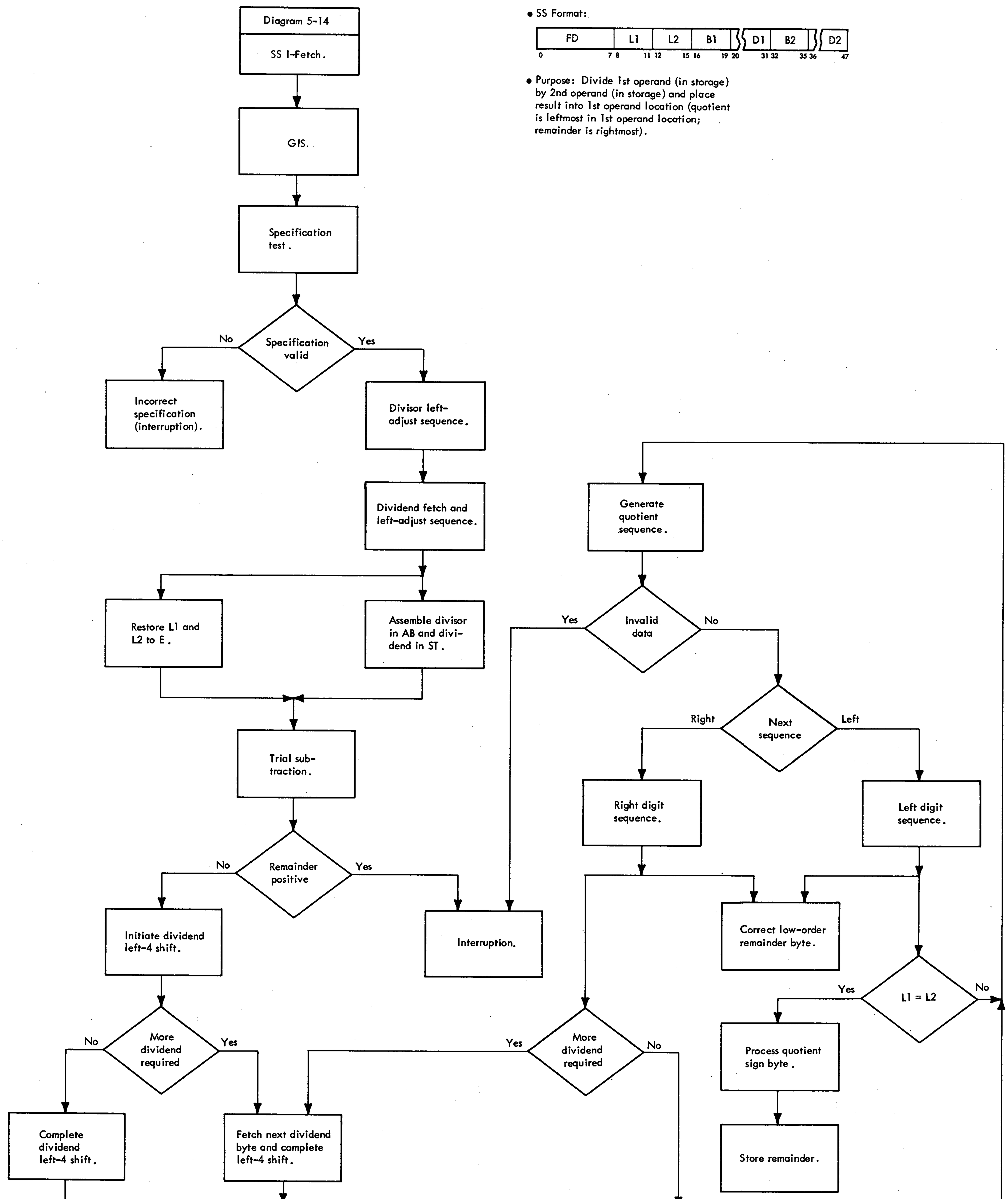


BASIC MULTIPLY ADD OR SUBTRACT SEQUENCE FOR RIGHT DIGIT, AND SHIFT RIGHT 4 SEQUENCE (QS071 AND QS081)

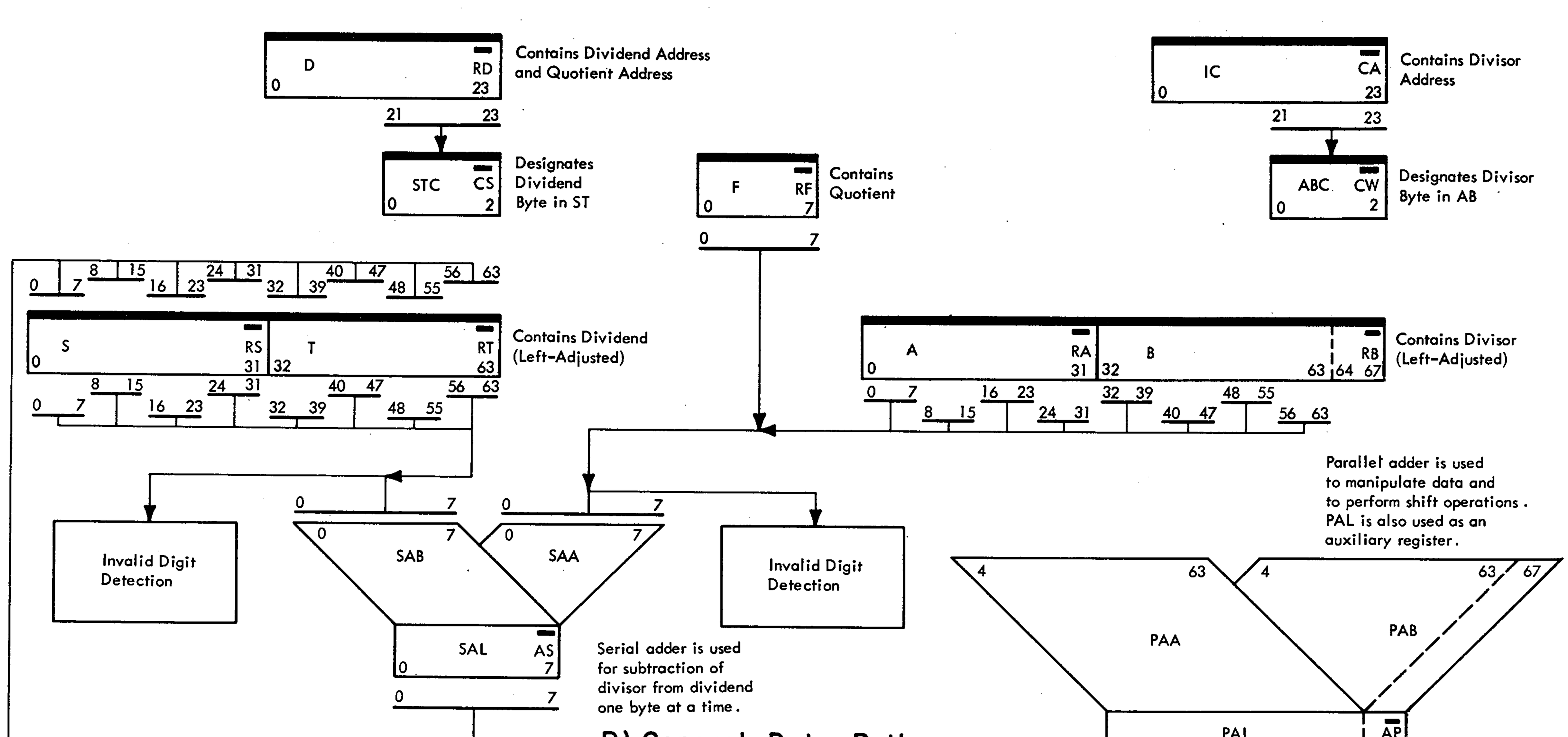
Objectives:

1. If multiplicand digit is 4 or less, multiplier is added to partial product number of times specified by multiplicand digit.
2. If multiplicand digit is 5 or greater, multiplier is subtracted from partial product number of times specified by 10 minus value of multiplicand digit.
3. If multiplicand digit is 0, no addition or subtraction cycles are required.
4. After exit from add or subtract loop, low-order partial product digit is saved in F(4-7) for subsequent store. Partial product in ST is shifted right 4 to drop low-order digit, and a branch is made to left-digit sequence.

Diagram 5-305. Decimal Multiply (Sheet 7 of 7)



A) Overall Flow Chart



B) General Data Path

Diagram 5-306. Decimal Divide (Sheet 1 of 9)

- SS Format:

FD	L1	L2	B1	D1	B2	D2
0–7	8–11	12–15	16–19	20–31	32–35	36–47

- Purpose: Divide 1st operand (in storage) by 2nd operand (in storage) and place result into 1st operand location (quotient is leftmost in 1st operand location; remainder is rightmost).

- Conditions at end of I-Fetch:
  1. Main storage request for doubleword containing low-order byte of 1st operand (dividend) has been issued per D.
  2. D contains low-order byte address (contents of GPR addressed by B1, + D1 + L1) of 1st operand.
  3. IC contains high-order byte address (contents of GPR addressed by B2, + D2) of 2nd operand (divisor).

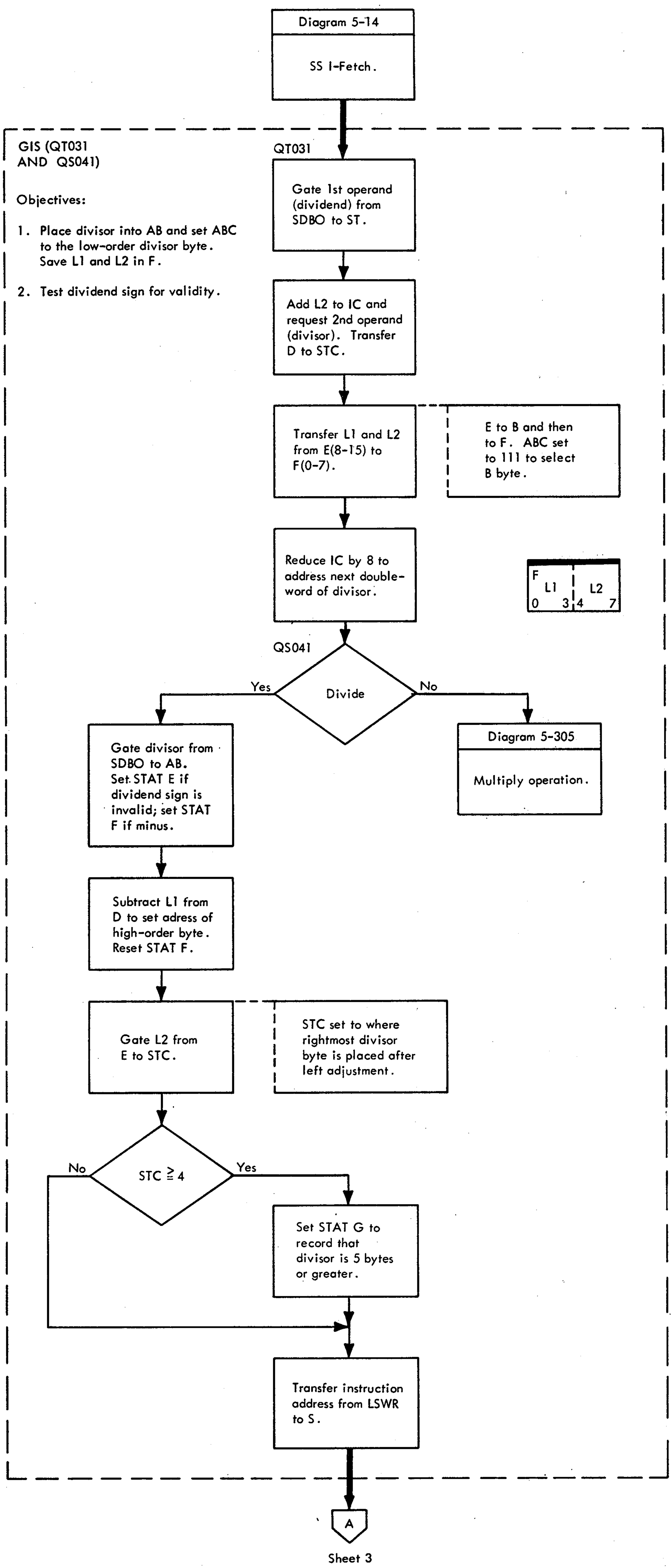


Diagram 5-306. Decimal Divide (Sheet 2 of 9)

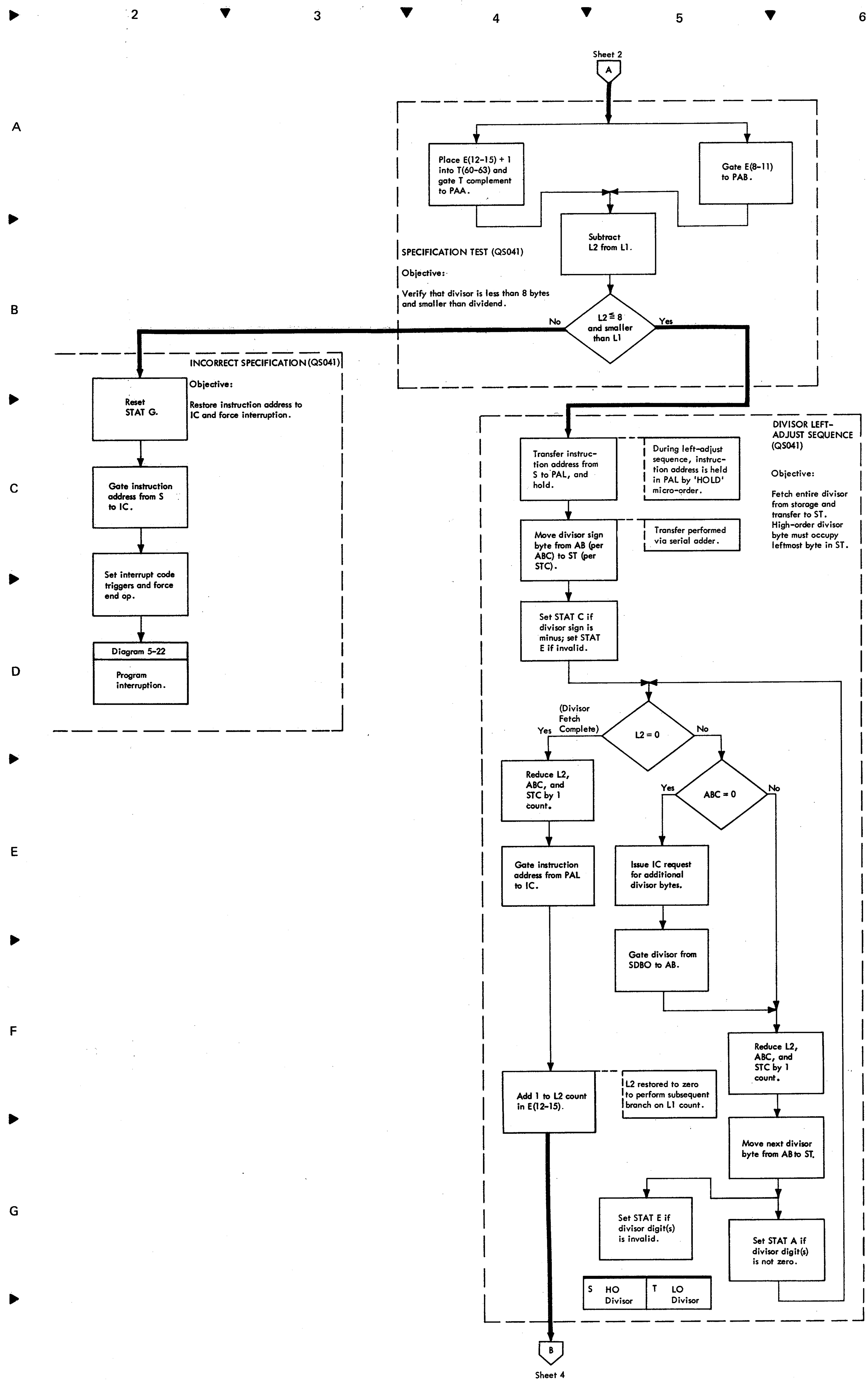


Diagram 5-306. Decimal Divide (Sheet 3 of 9)

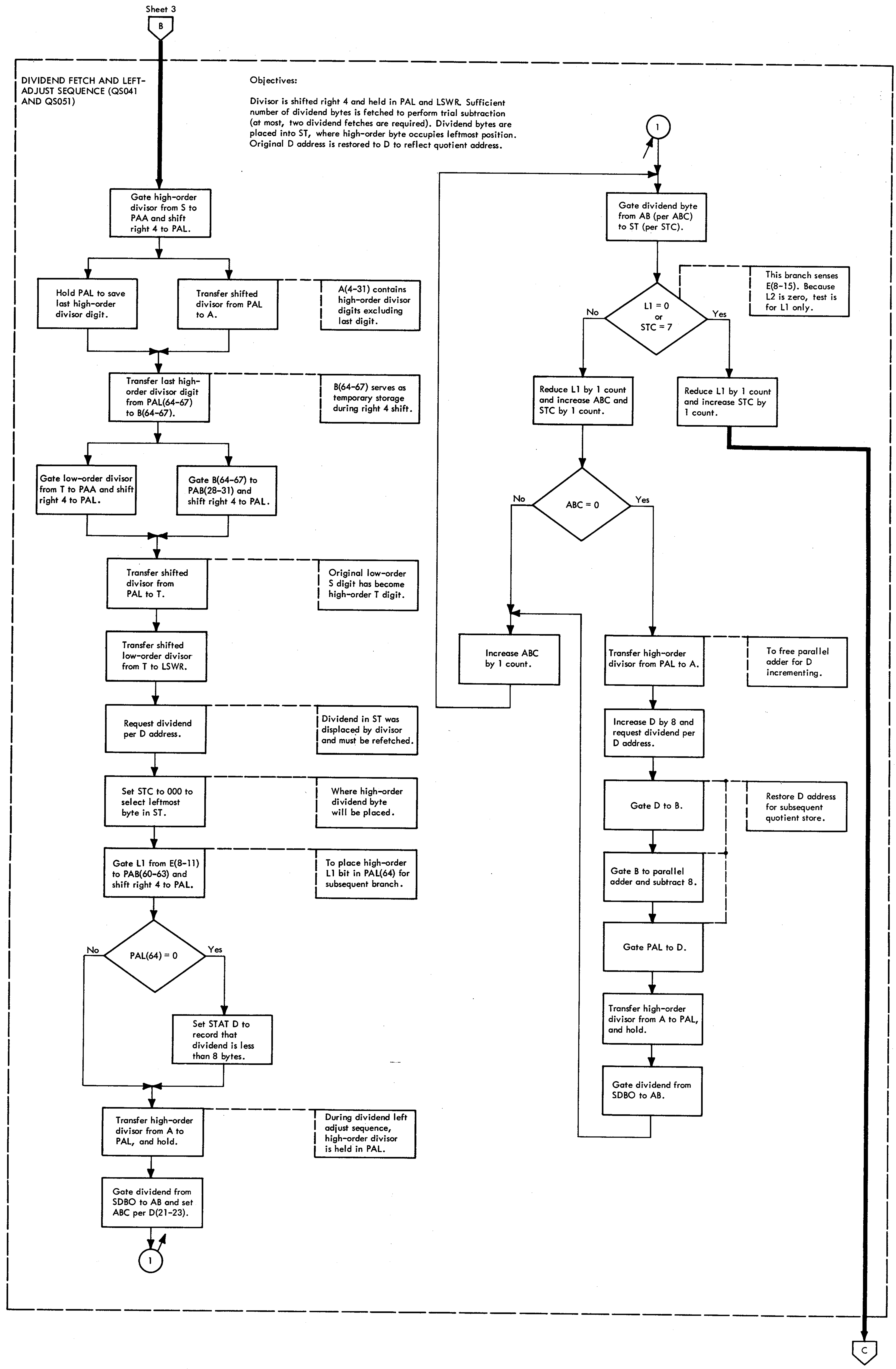


Diagram 5-306. Decimal Divide (Sheet 4 of 9)

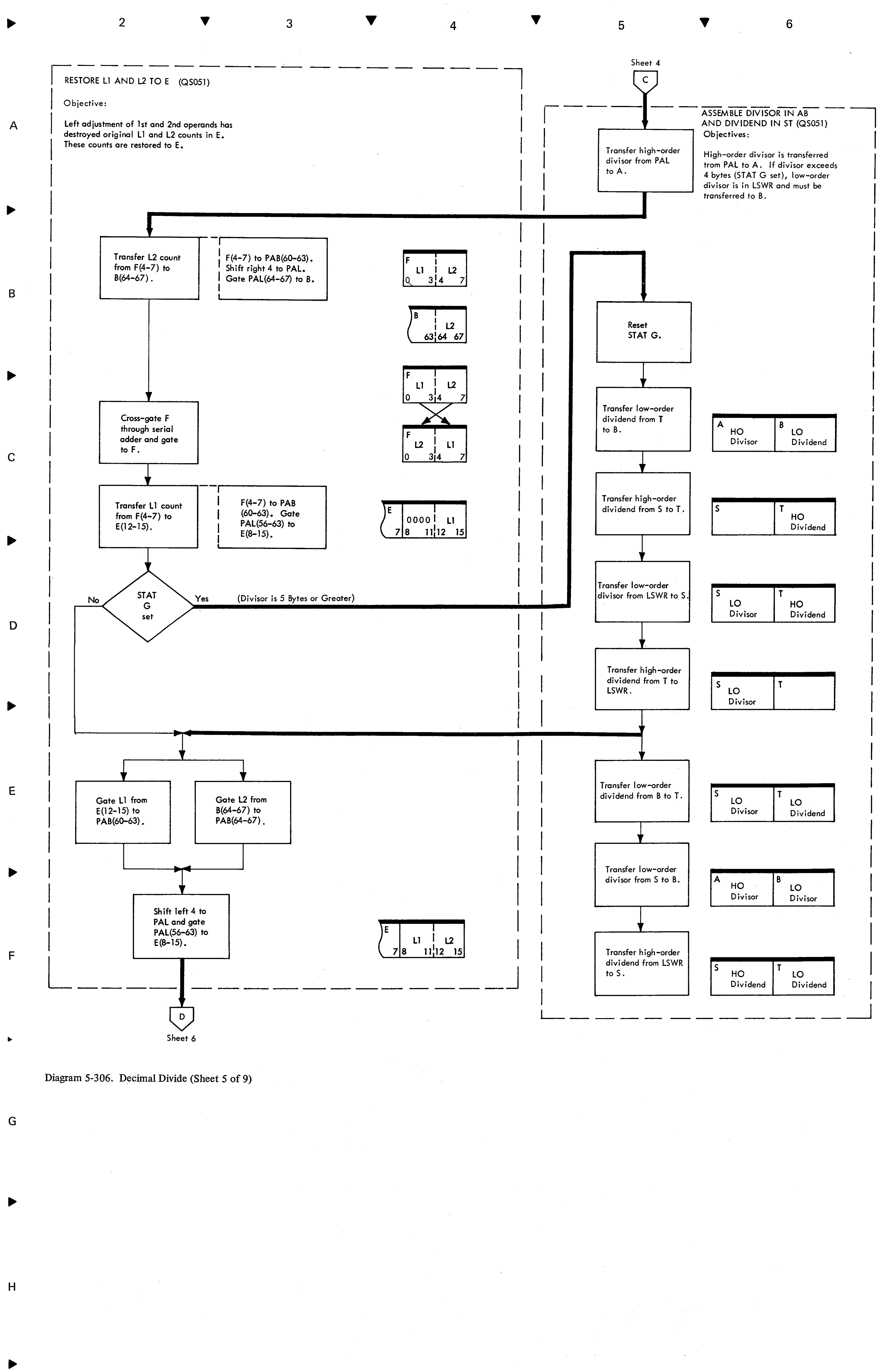


Diagram 5-306. Decimal Divide (Sheet 5 of 9)

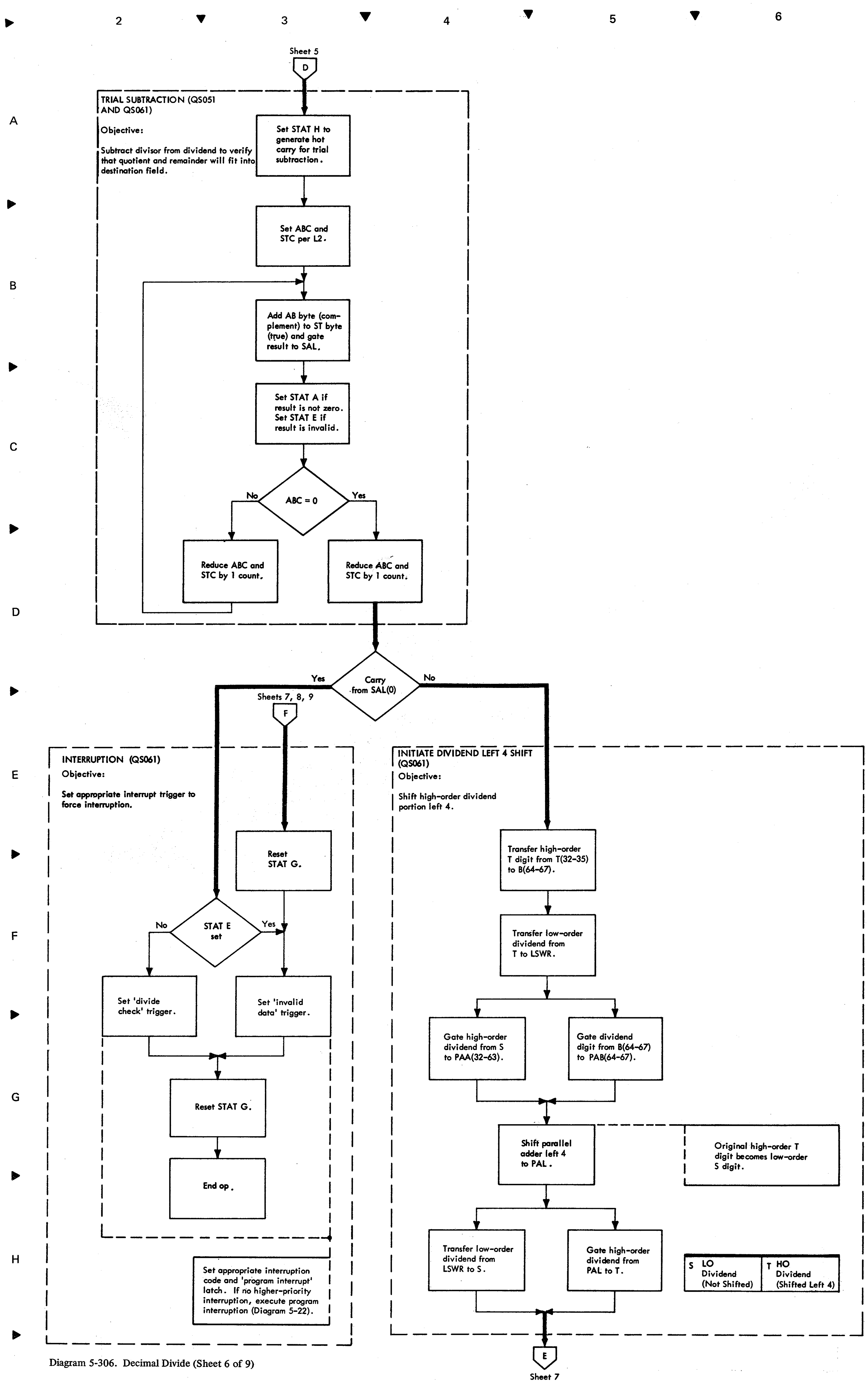


Diagram 5-306. Decimal Divide (Sheet 6 of 9)

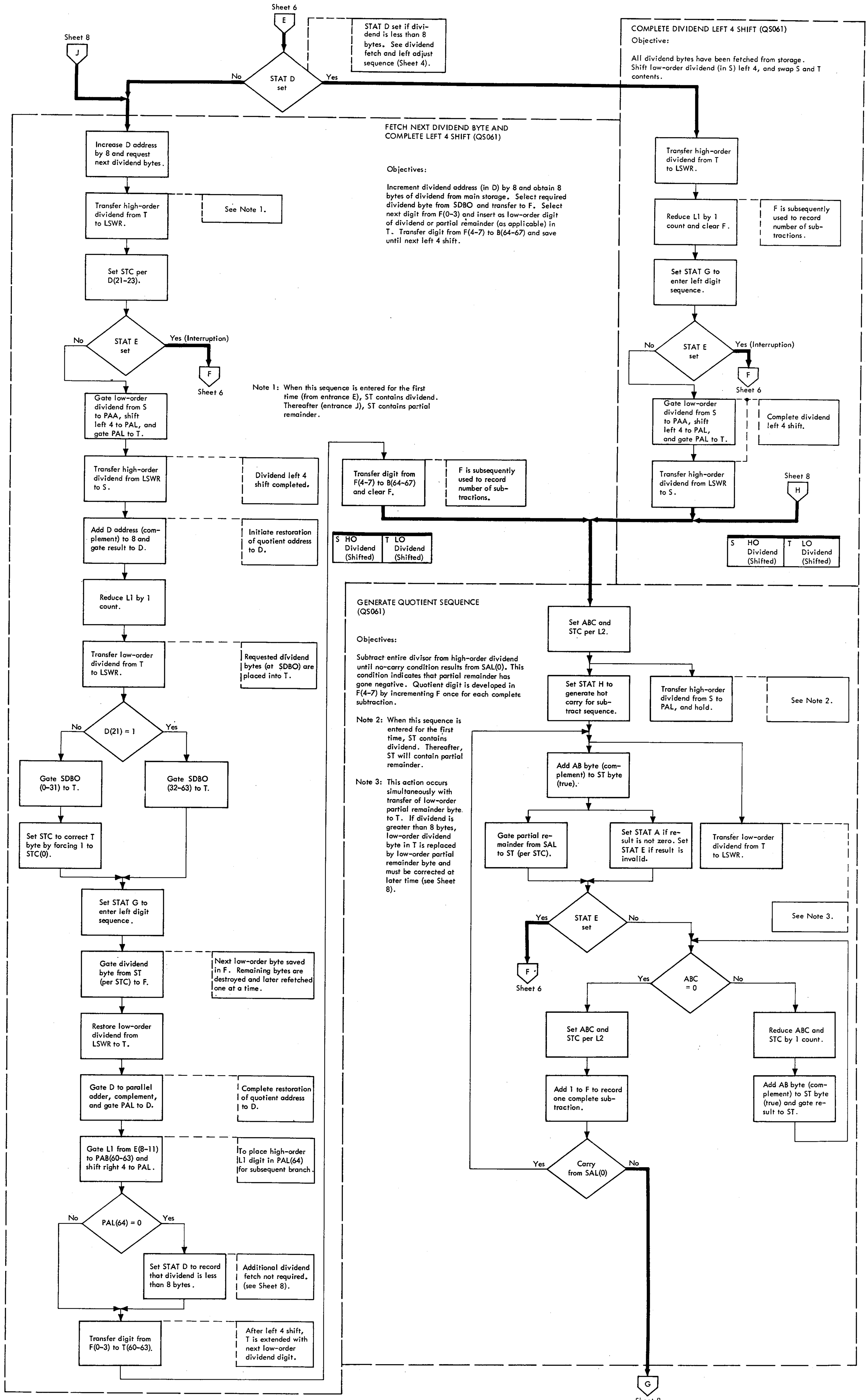


Diagram 5-306. Decimal Divide (Sheet 7 of 9)

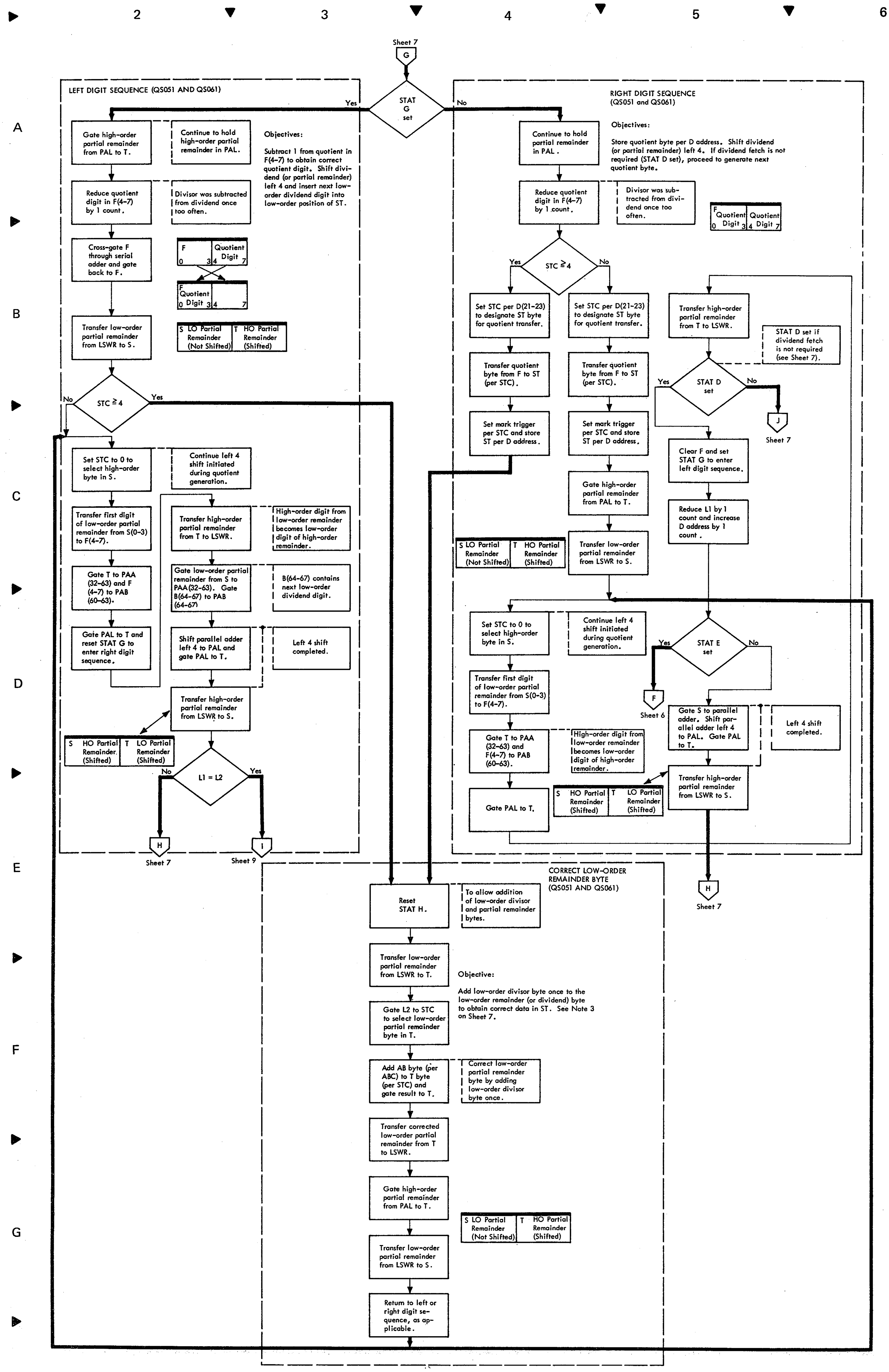


Diagram 5-306. Decimal Divide (Sheet 8 of 9)

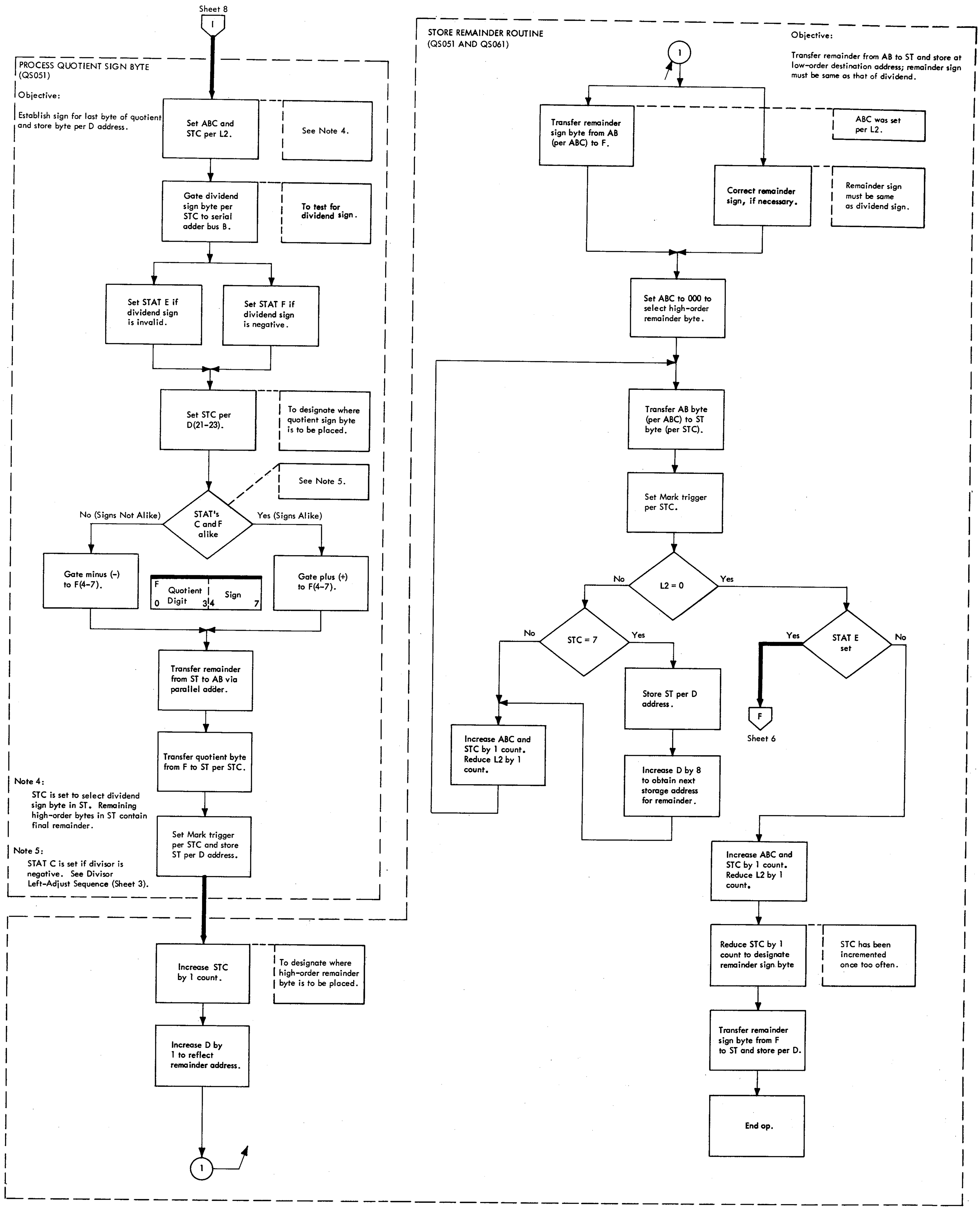


Diagram 5-306. Decimal Divide (Sheet 9 of 9)

- SS Format:

Op Code	L1	L2	B1	D1	B2	D2
0 – 7	8 – 11	12 – 15	16 – 19	20 – 31	32 – 35	36 – 47

- Op Code:
  1. Pack (PACK) - F2.
  2. Unpack (UNPK) - F3.
  3. Move With Offset (MVO) - F1.

- Purpose:
  1. Pack - Convert format of 2nd operand (in storage) from zoned to packed and place result into 1st operand location (in storage).
  2. Unpack - Convert format of 2nd operand (in storage) from packed to zoned and place result into 1st operand location (in storage).
  3. Move with Offset - Store 2nd operand (in storage) to left of and adjacent to low-order 4 bits of 1st operand (in storage).

- Conditions at end of I-Fetch:
  1. Main storage request for doubleword containing low-order byte of 1st operand (destination) has been issued per D.
  2. D contains low-order byte address (contents of GPR addressed by B1, + D1 + L1) of 1st operand.
  3. IC contains high-order byte address (contents of GPR addressed by B2, + D2) of 2nd operand.

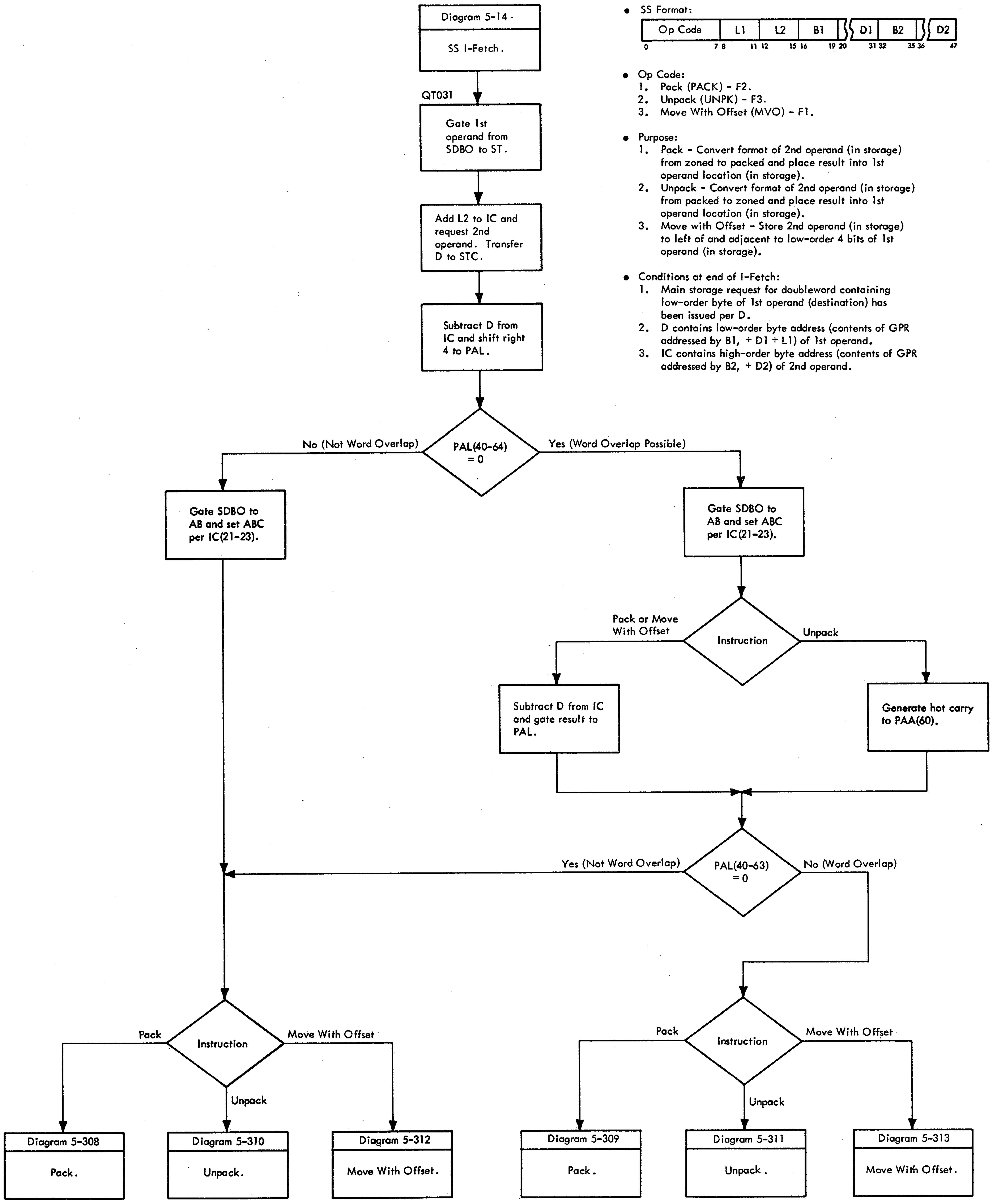


Diagram 5-307. GIS for Pack, Unpack, and Move With Offset

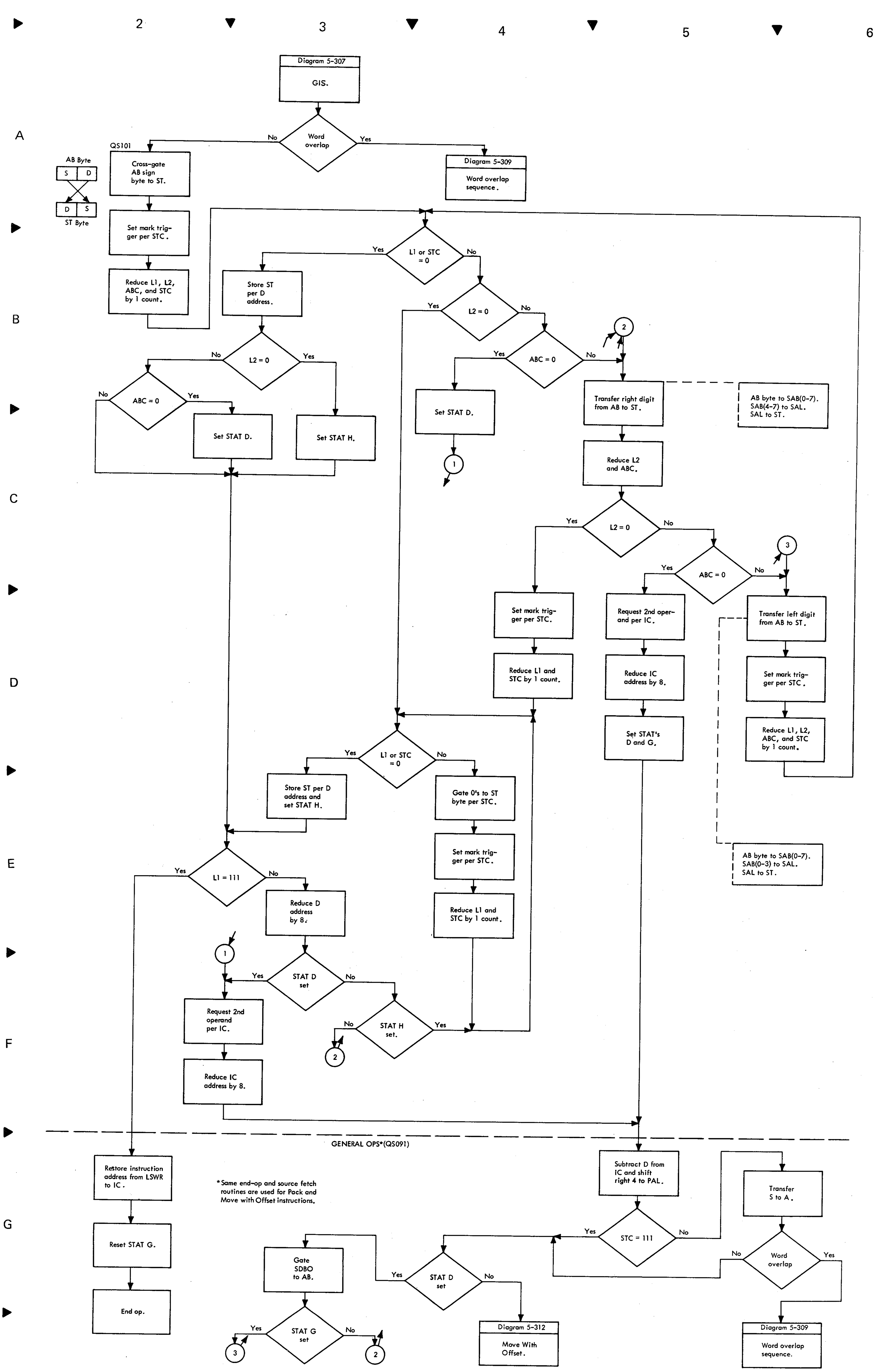


Diagram 5-308. Pack, Not Word Overlap Sequence

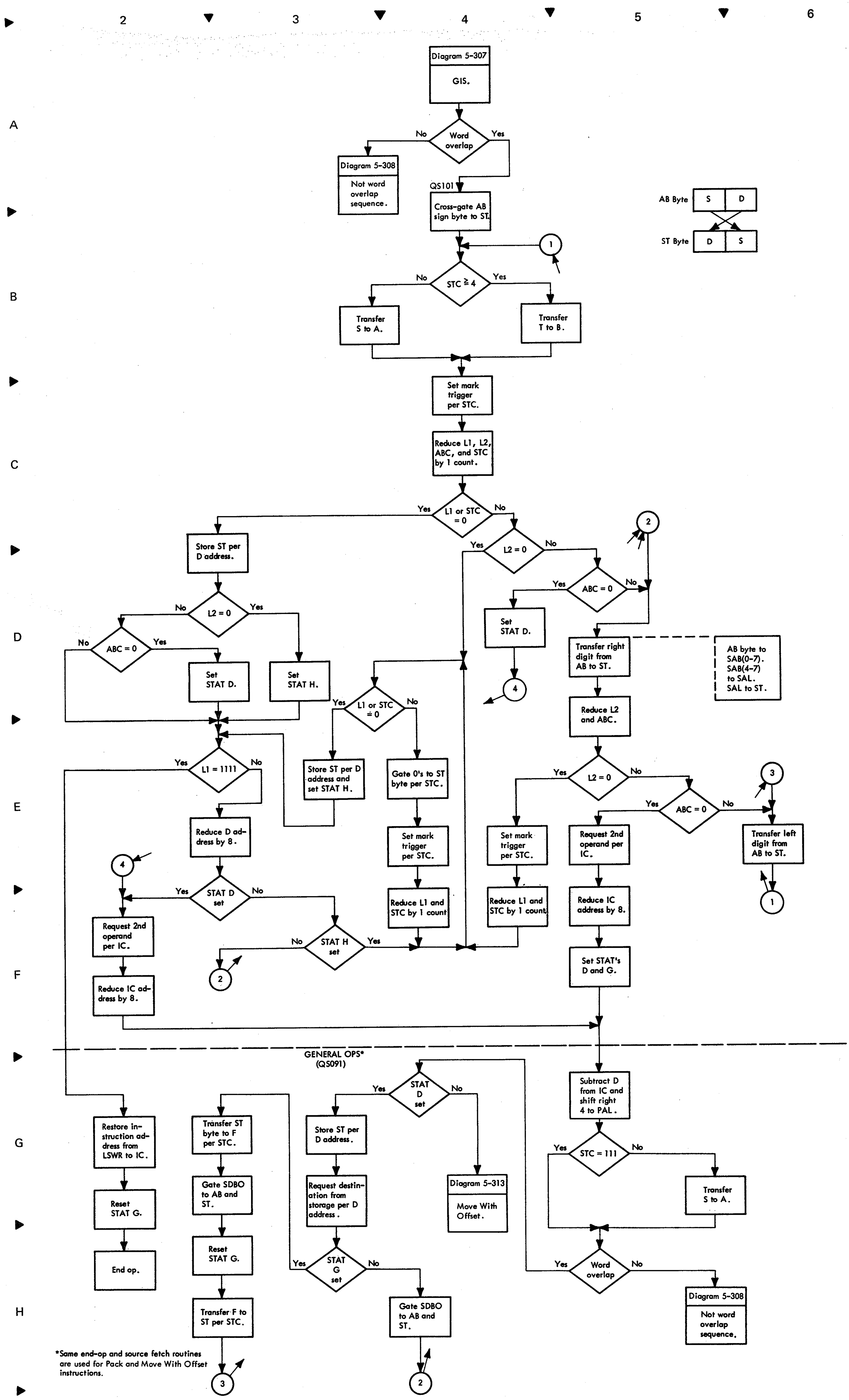


Diagram 5-309. Pack, Word Overlap Sequence

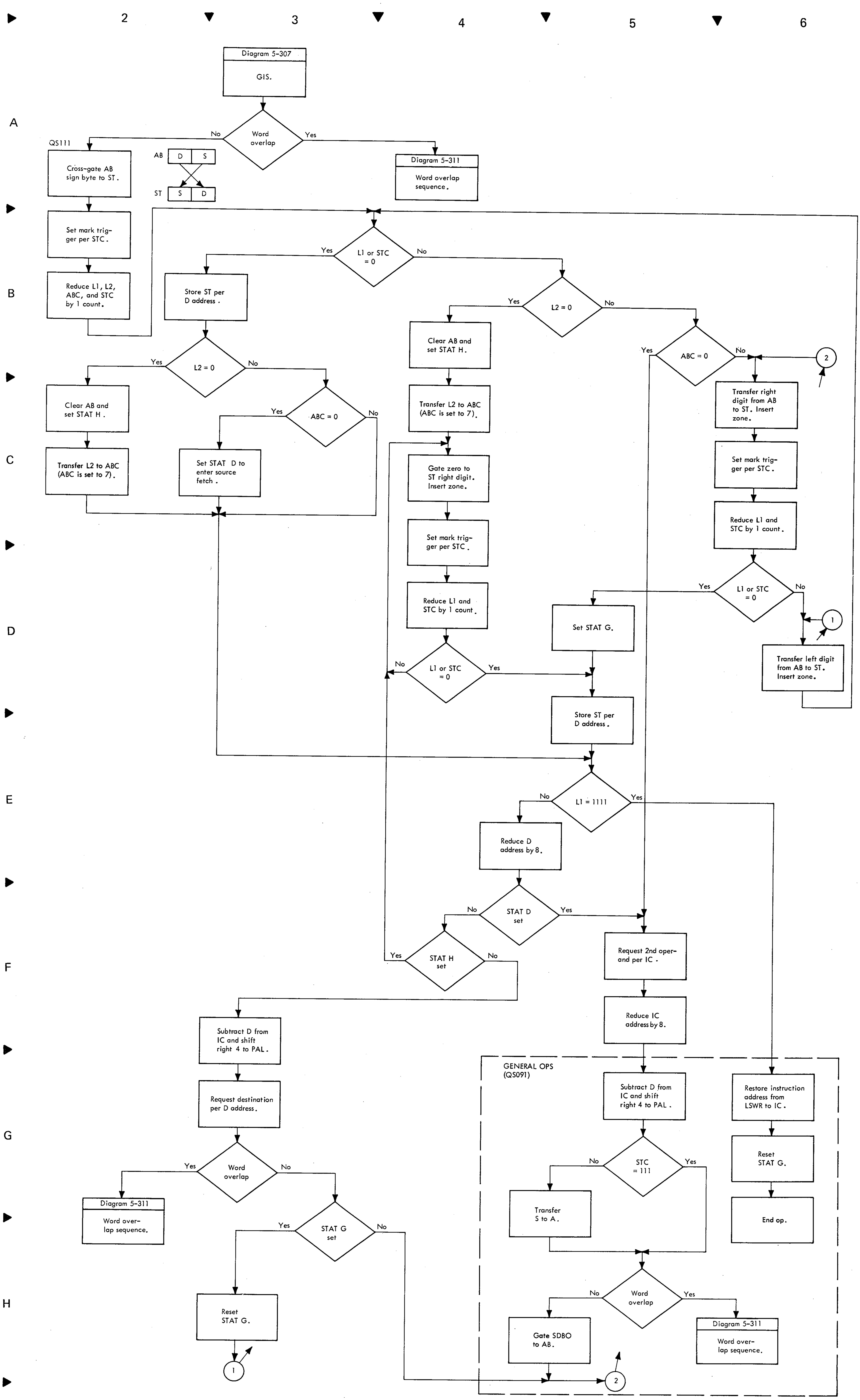


Diagram 5-310. Unpack, Not Word Overlap Sequence

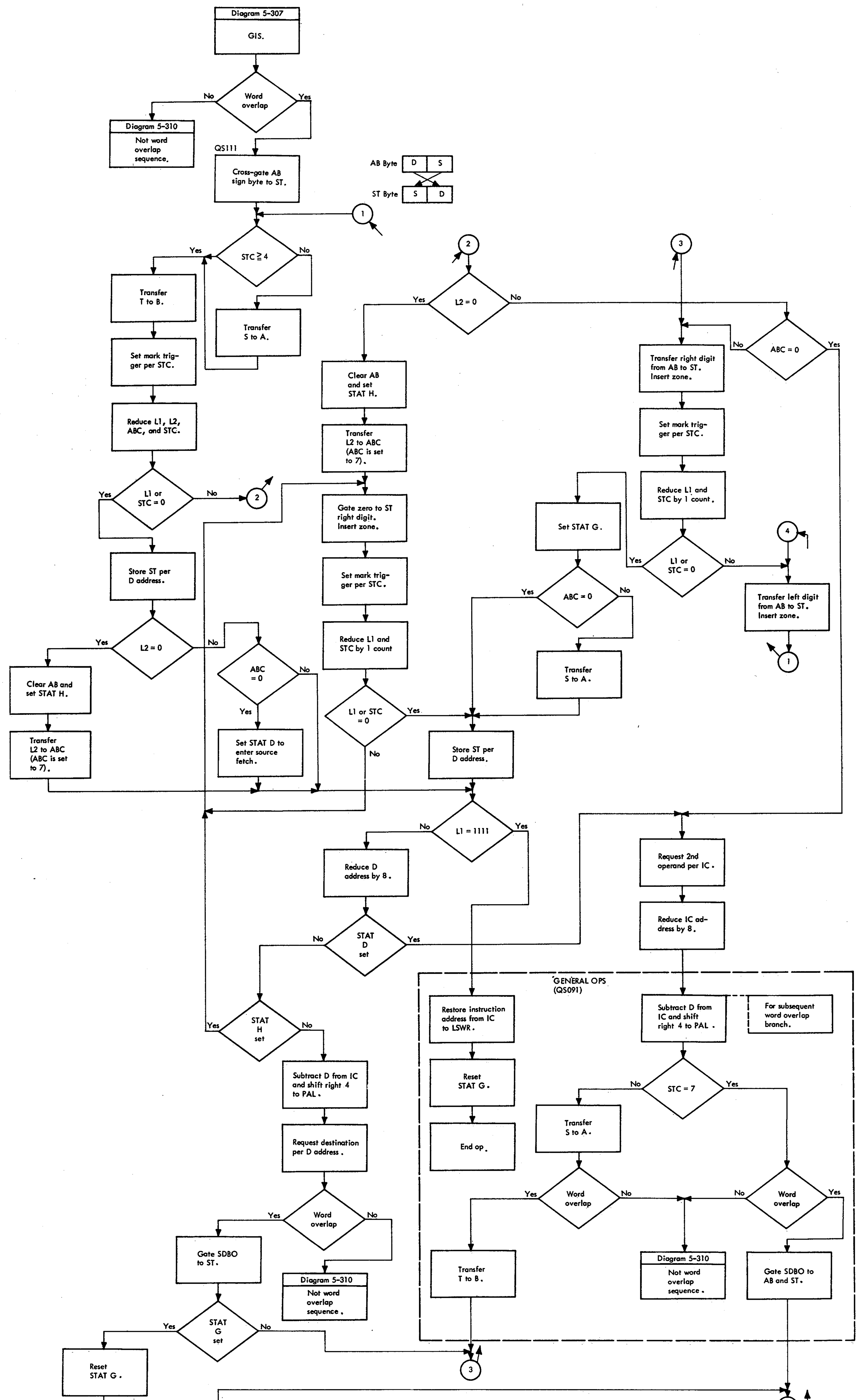


Diagram 5-311. Unpack, Word Overlap Sequence

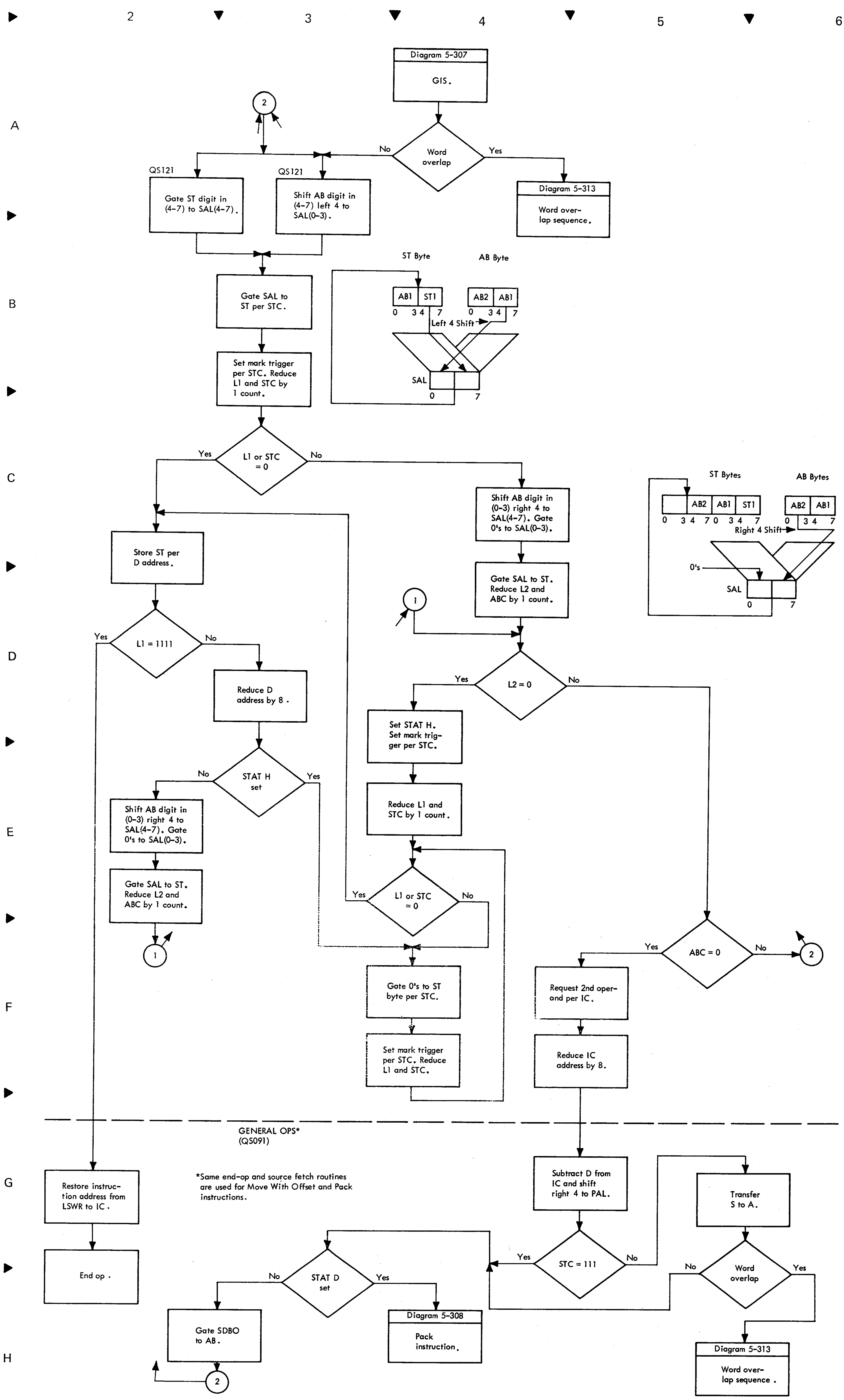


Diagram 5-312. Move With Offset, Not Word Overlap Sequence

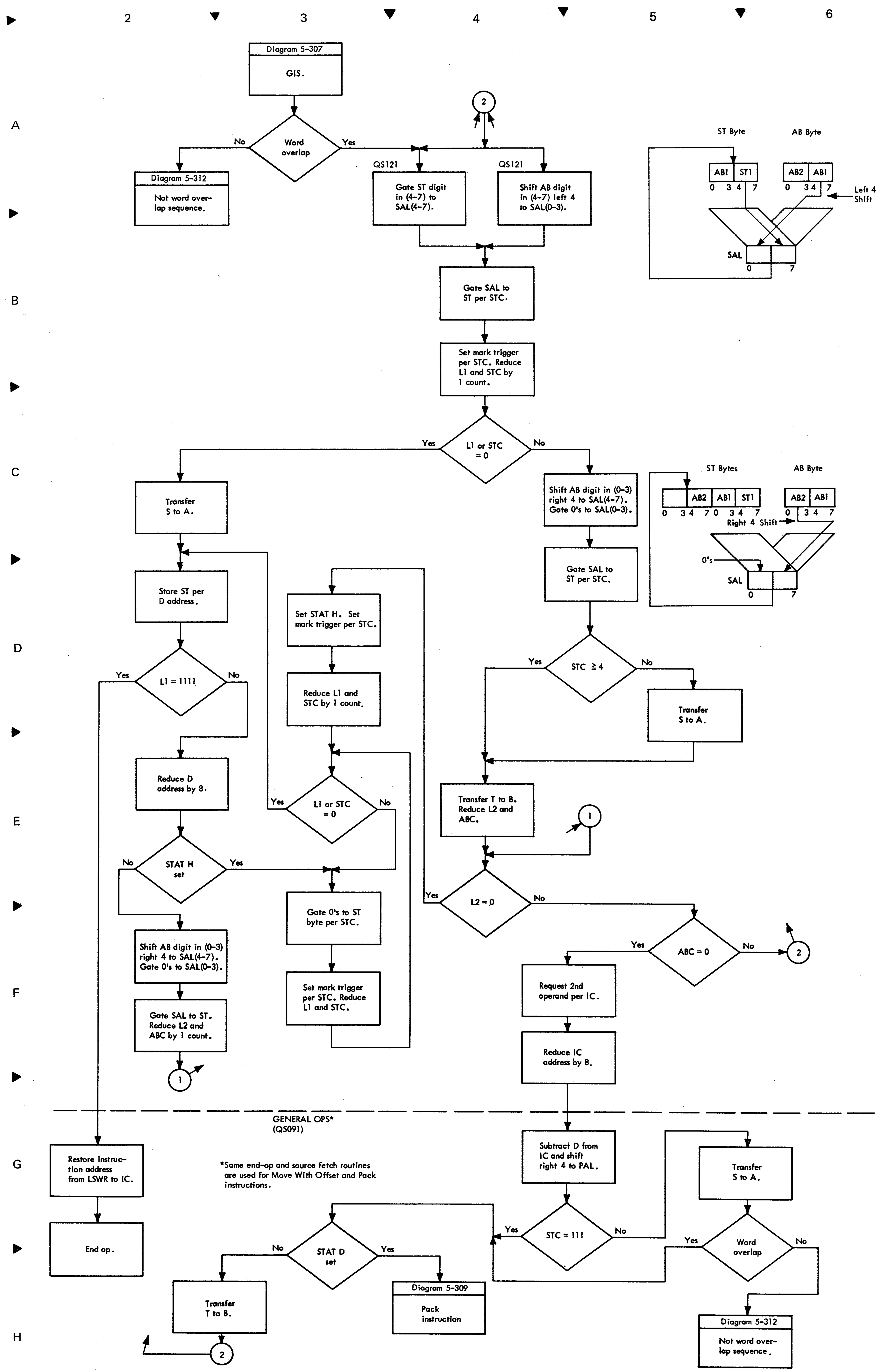


Diagram 5-313. Move With Offset, Word Overlap Sequence

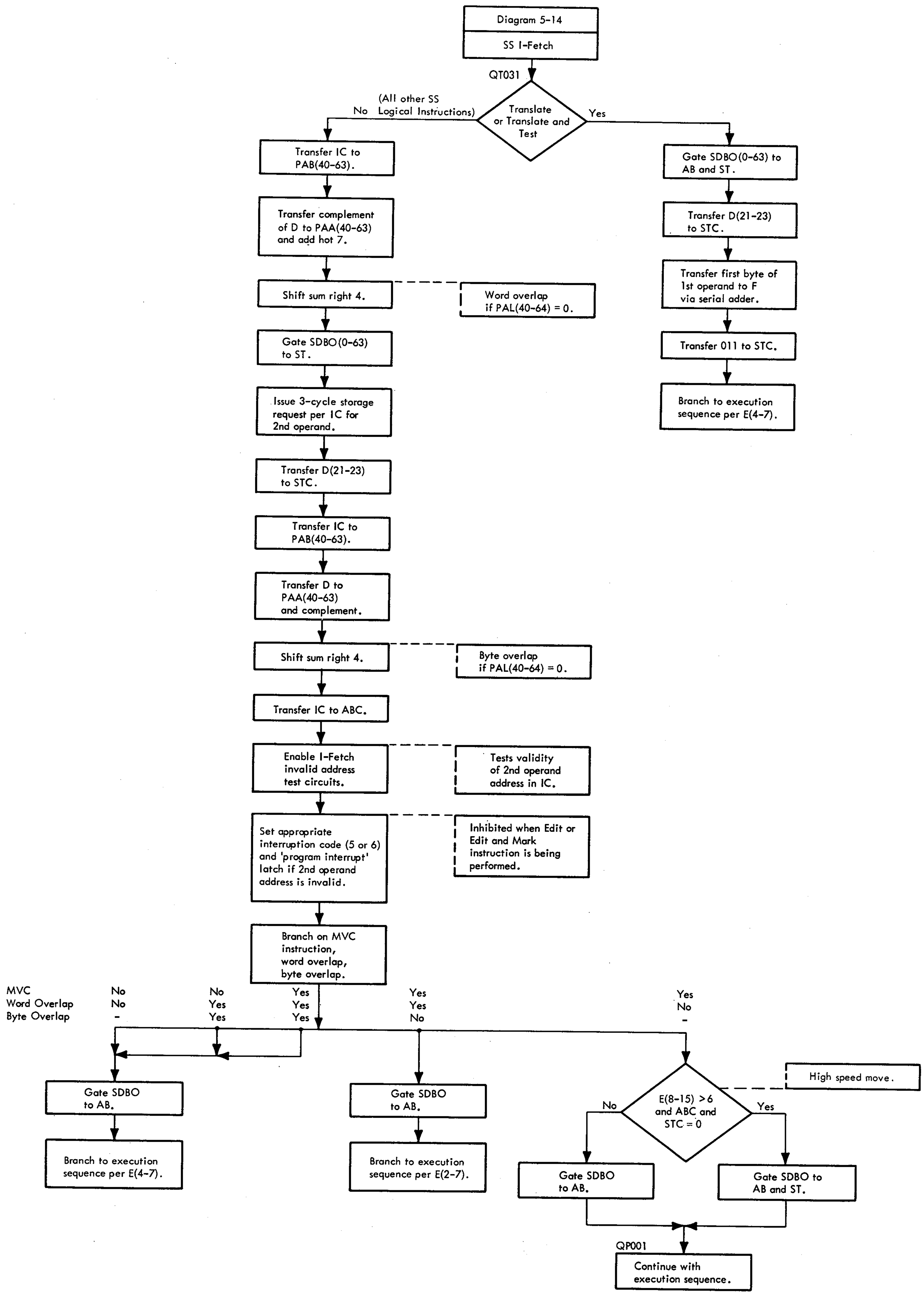
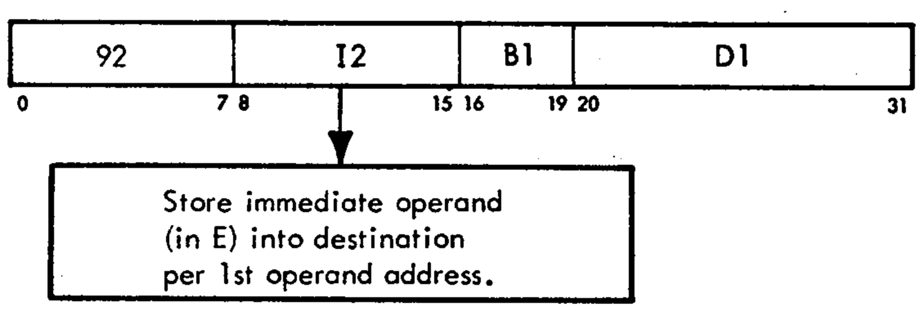
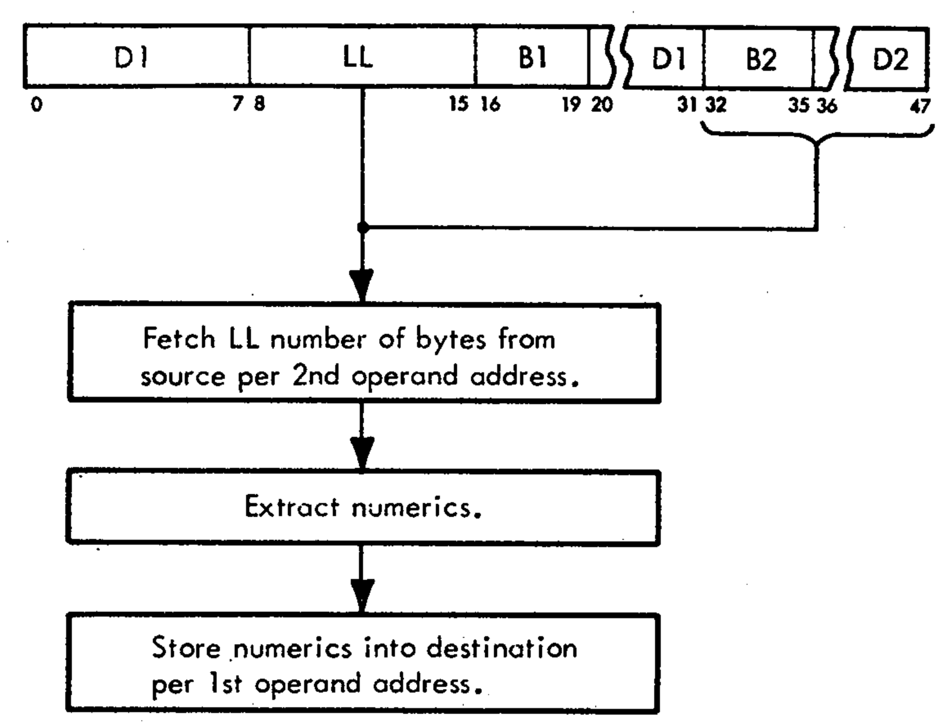


Diagram 5-401. GIS for Logical Instructions



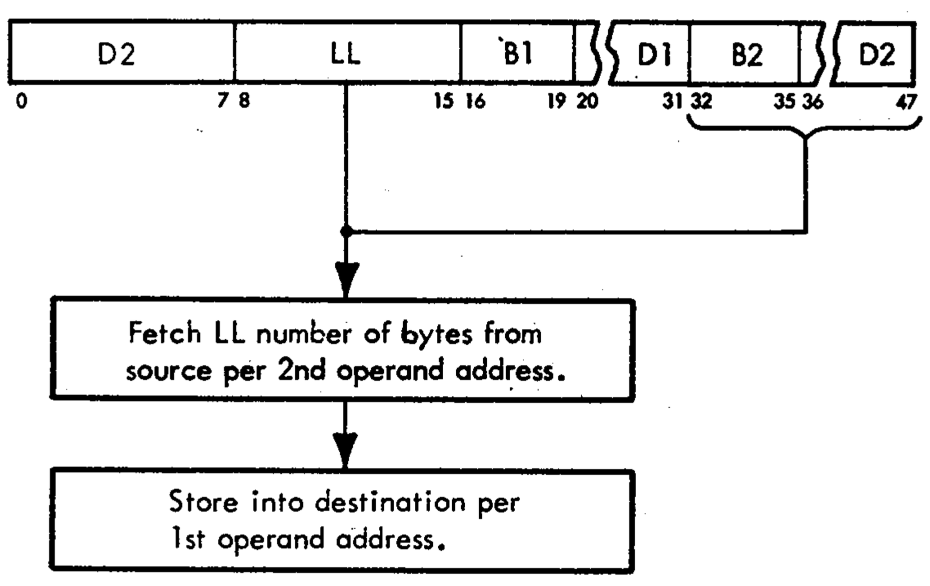
A. Move, MVI (92)

- SI format.
- Purpose: Place immediate operand (I2 of instruction) into 1st operand location (in storage).



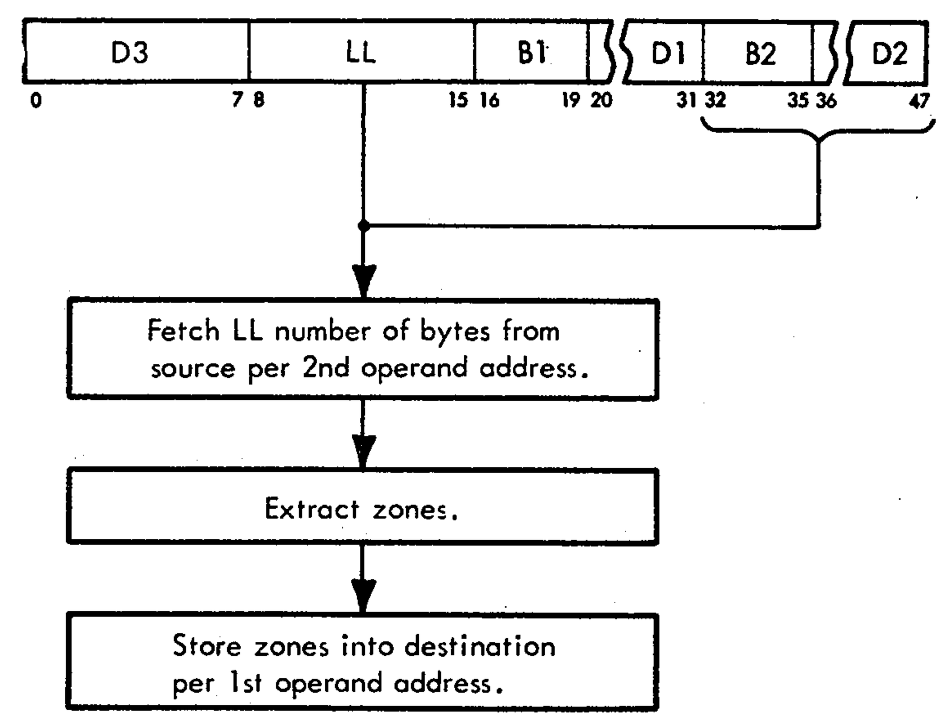
C. Move Numerics, MVN (D1)

- SS format.
- Purpose: Place numeric portion (low-order 4 bits) of each byte of 2nd operand (in storage) into low-order 4 bits of corresponding byte of 1st operand (in storage).



B. Move, MVC (D2)

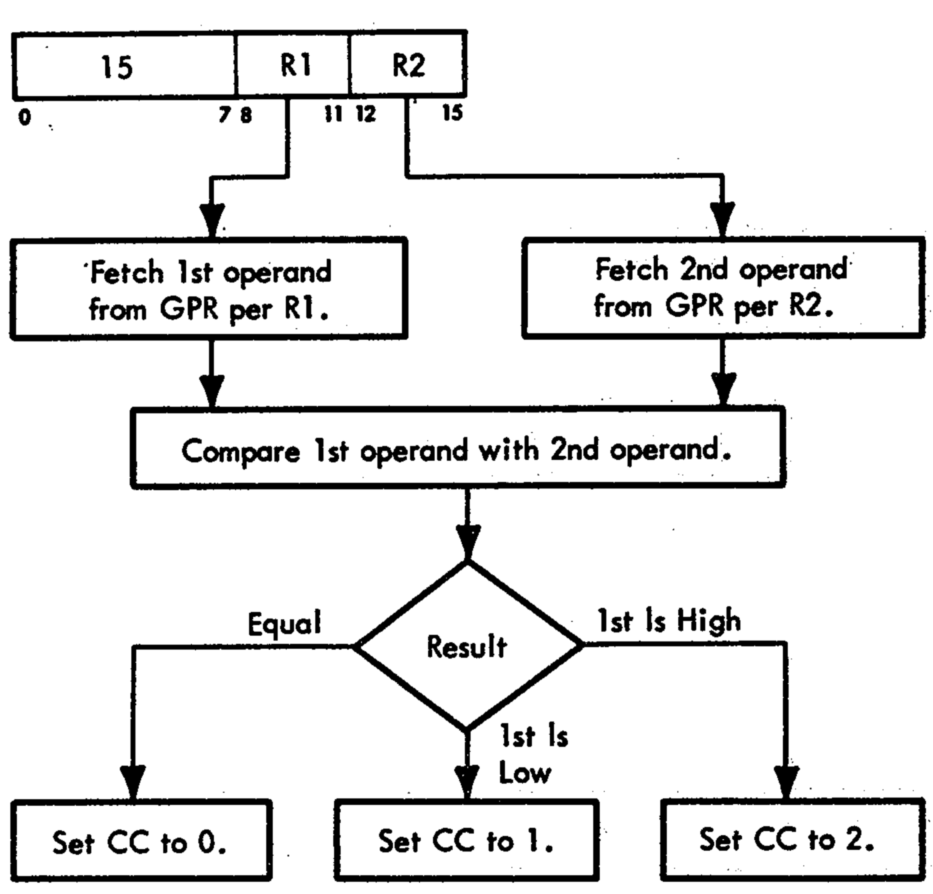
- SS format.
- Purpose: Place 2nd operand (in storage) into 1st operand location (in storage).



D. Move Zones, MVZ (D3)

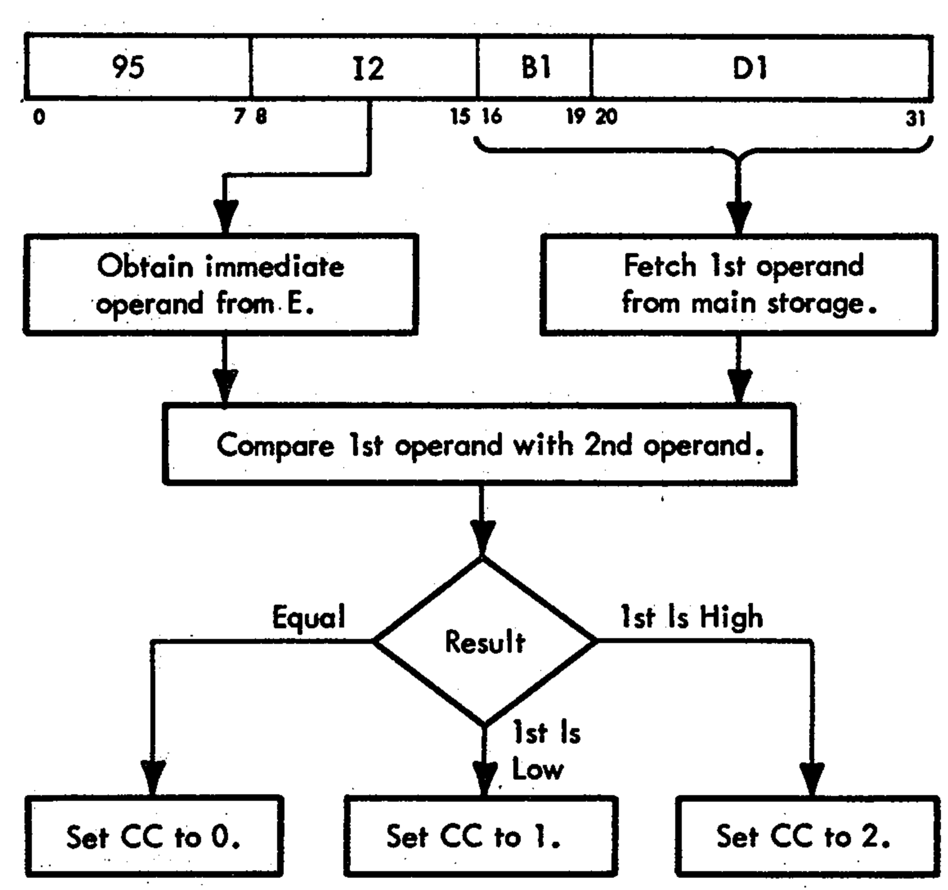
- SS format.
- Purpose: Place zone portion (high-order 4 bits) of each byte of 2nd operand (in storage) into high-order 4 bits of corresponding byte of 1st operand (in storage).

Diagram 5-402. Logical Move Instructions



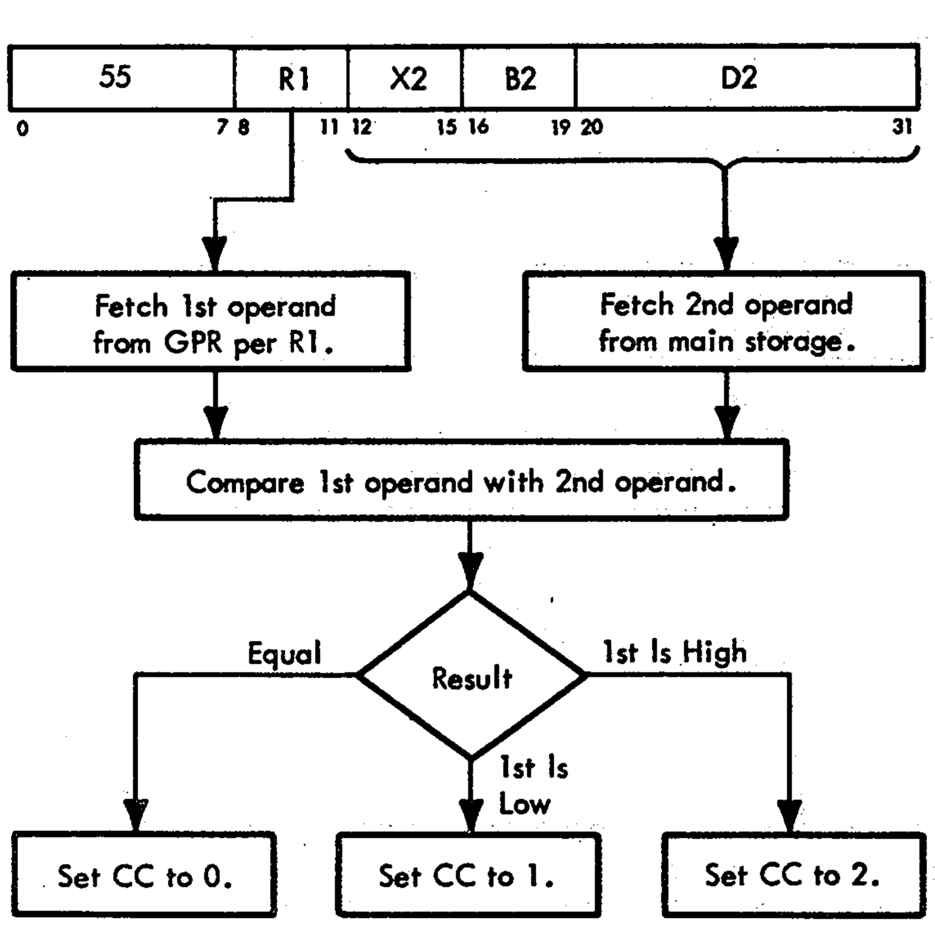
A. Compare Logical, CLR (15)

- RR format.
- Purpose: Binarily compare 1st operand (in GPR, per R1) with 2nd operand (in GPR, per R2) and set CC according to result.



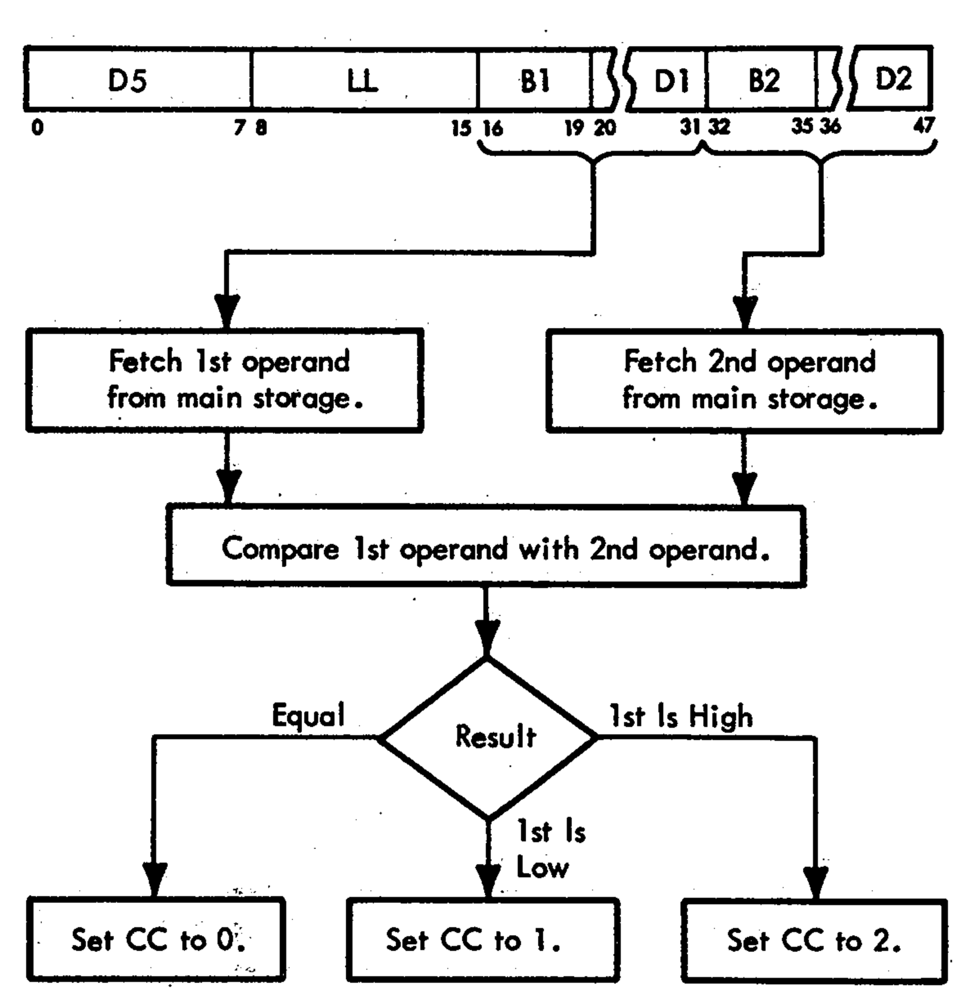
C. Compare Logical, CLI (95)

- SI format.
- Purpose: Binarily compare 1st operand (in storage) with immediate operand (I2 of instruction) and set CC according to result.



B. Compare Logical, CL (55)

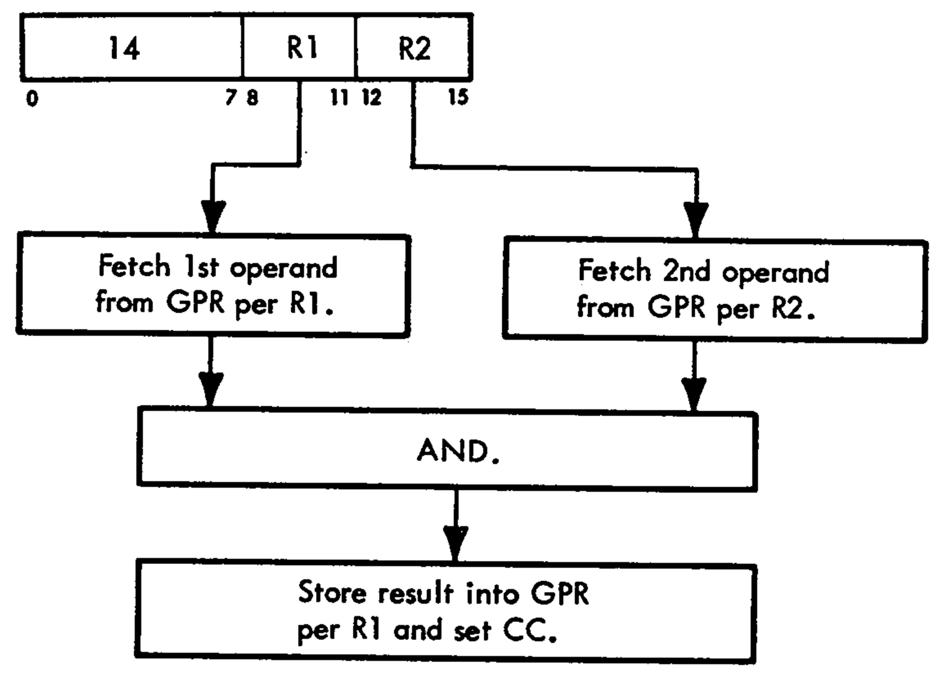
- RX format.
- Purpose: Binarily compare 1st operand (in GPR, per R1) with 2nd operand (in storage) and set CC according to result.



D. Compare Logical, CLC (D5)

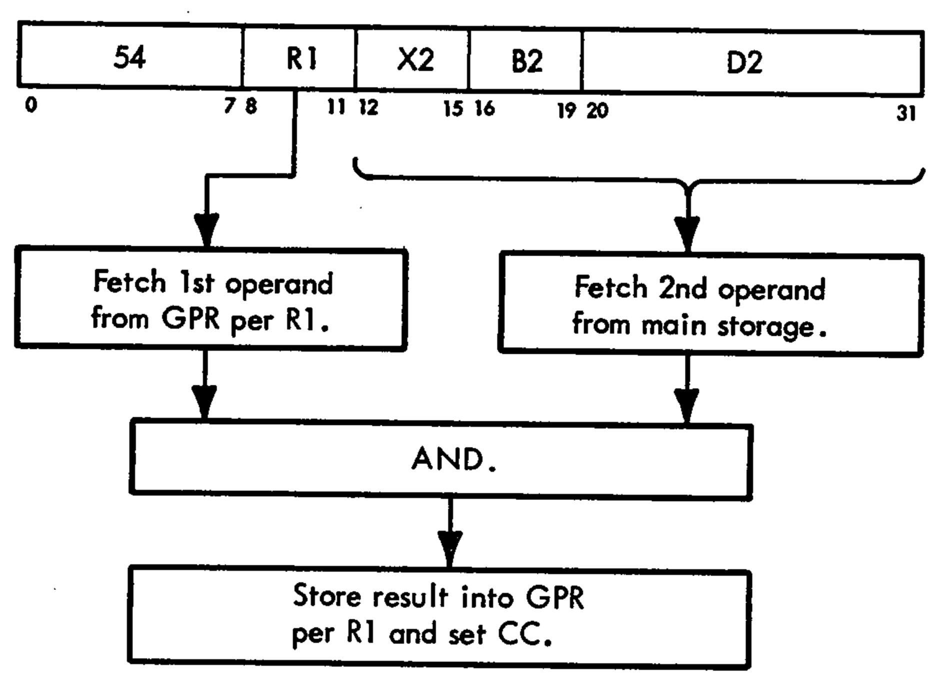
- SS format.
- Purpose: Binarily compare 1st operand (in storage) with 2nd operand (in storage) and set CC according to result.

Diagram 5-403. Logical Compare Instructions



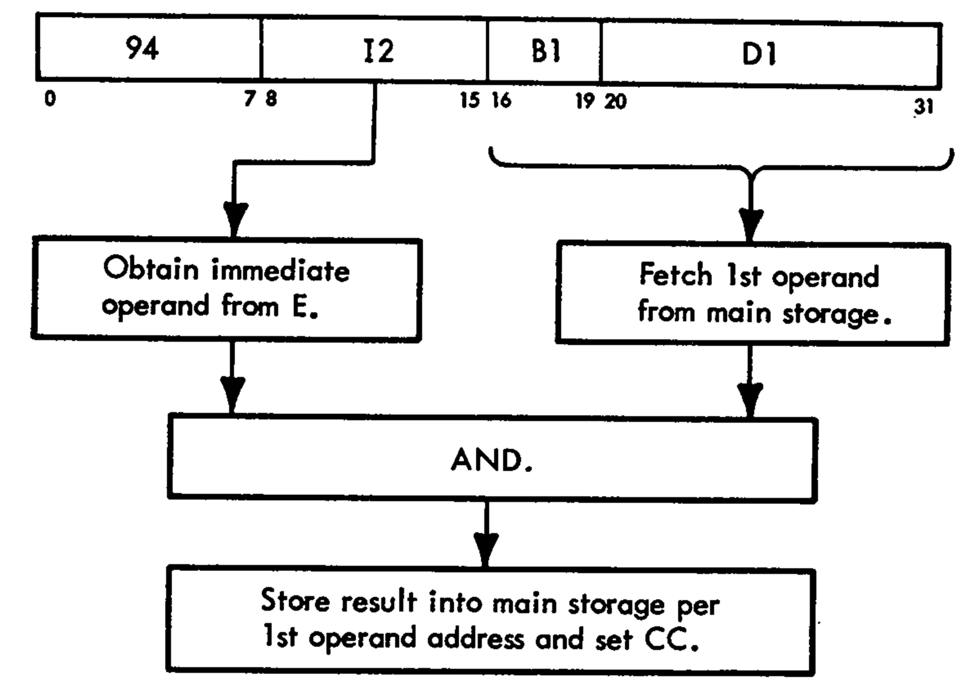
A. AND, NR (14)

- RR format.
- Purpose: AND 1st operand (in GPR, per R1) with 2nd operand (in GPR, per R2) and place result into 1st operand location.



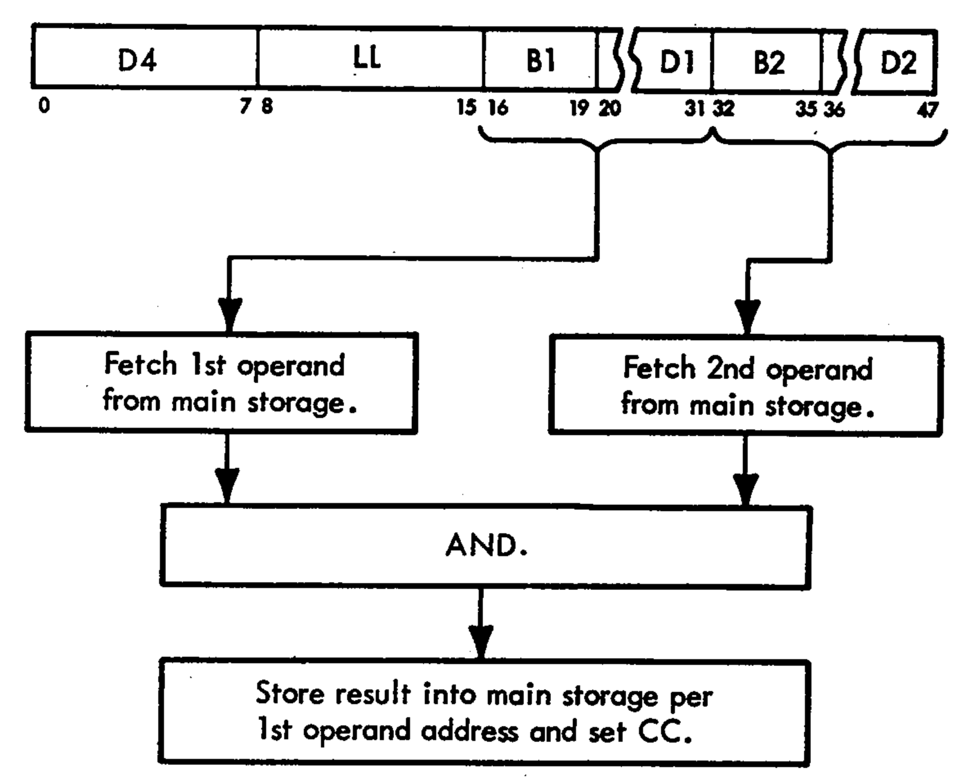
B. AND, N (54)

- RX format.
- Purpose: AND 1st operand (in GPR, per R1) with 2nd operand (in storage) and place result into 1st operand location.



C. AND, NI (94)

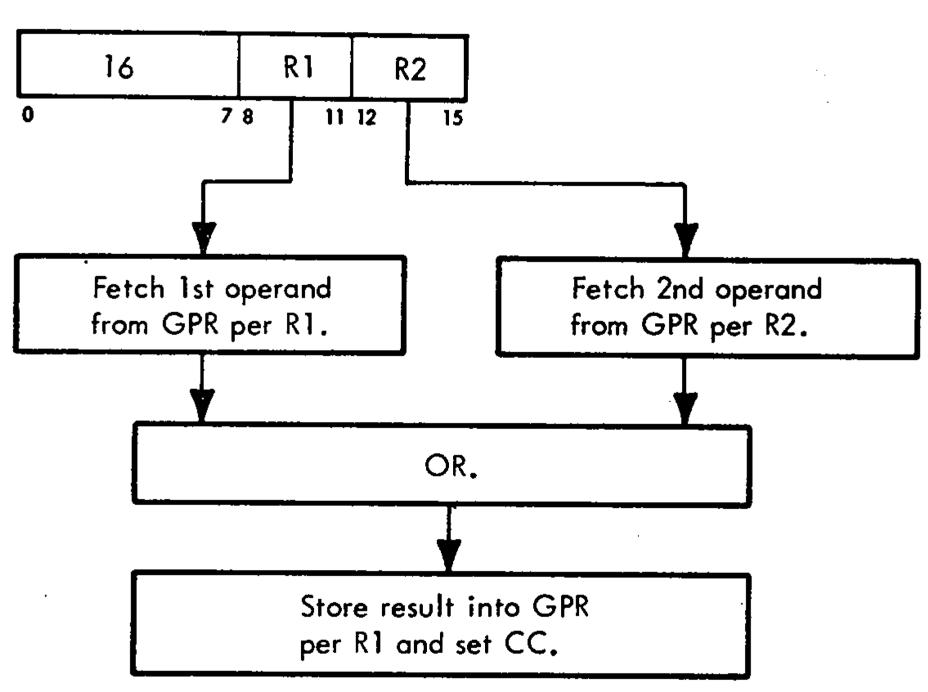
- SI format.
- Purpose: AND immediate operand (I2 of instruction) with 1st operand (in storage) and place result into 1st operand location.



D. AND, NC (D4)

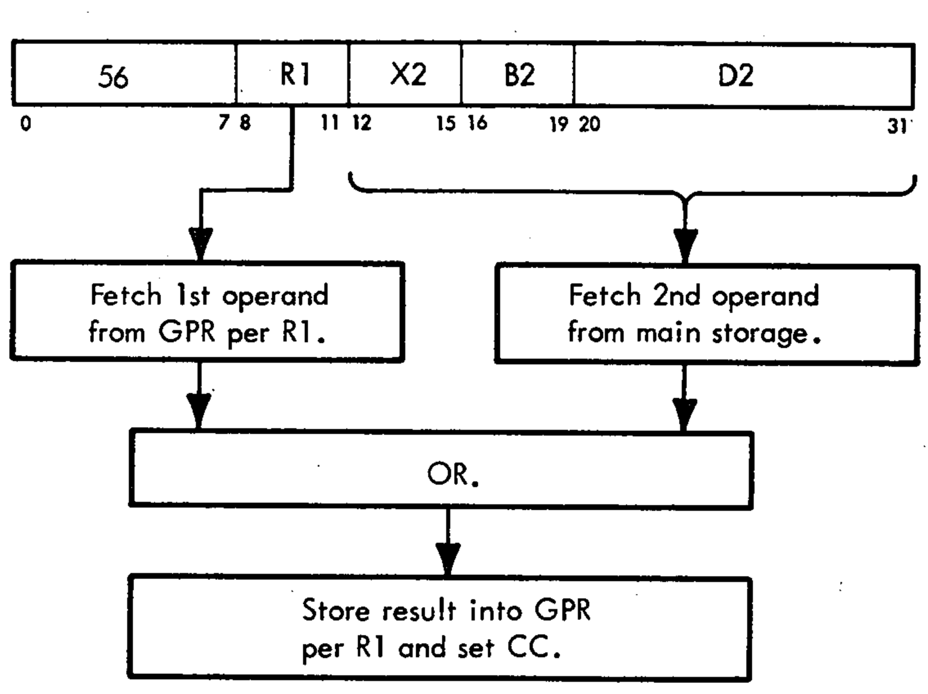
- SS format.
- Purpose: AND 1st operand (in storage) with 2nd operand (in storage) and place result into 1st operand location.

Diagram 5-404. Logical AND Instructions



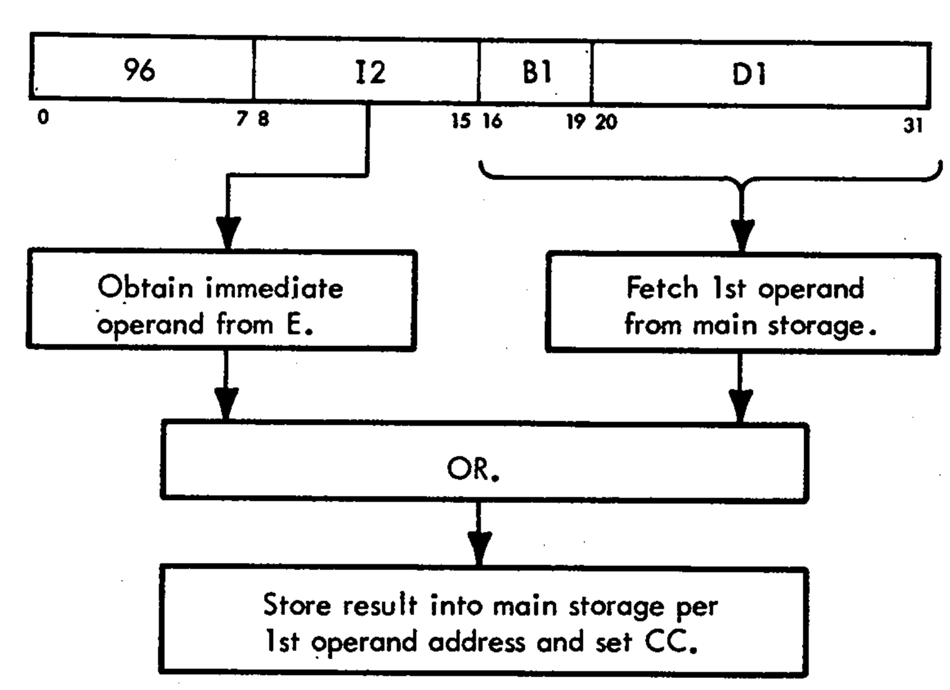
A. OR, OR (16)

- RR format.
- Purpose: OR 1st operand (in GPR, per R1) with 2nd operand (in GPR, per R2) and place result into 1st operand location.



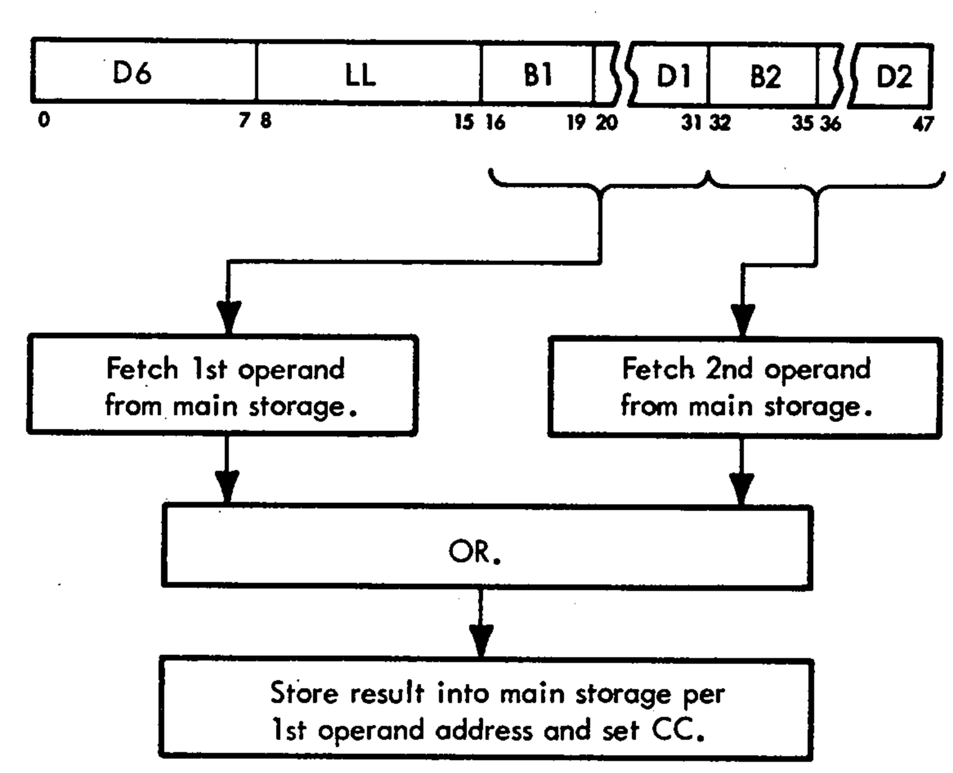
B. OR, O (56)

- RX format.
- Purpose: OR 1st operand (in GPR, per R1) with 2nd operand (in storage) and place result into 1st operand location.



C. OR, OI (96)

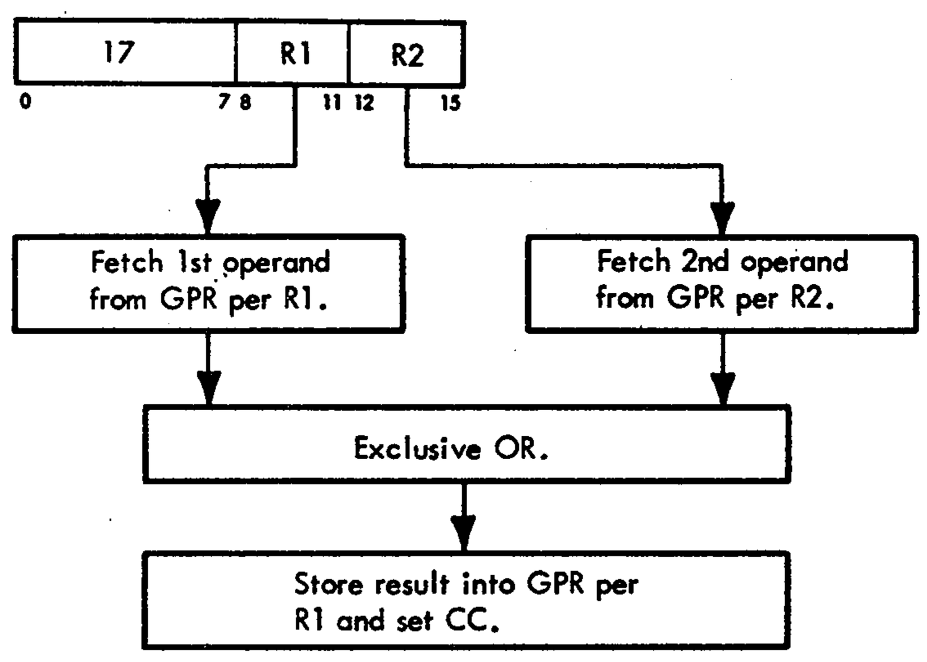
- SI format.
- Purpose: OR immediate operand (I2 of instruction) with 1st operand (in storage) and place result into 1st operand location.



D. OR, OC (D6)

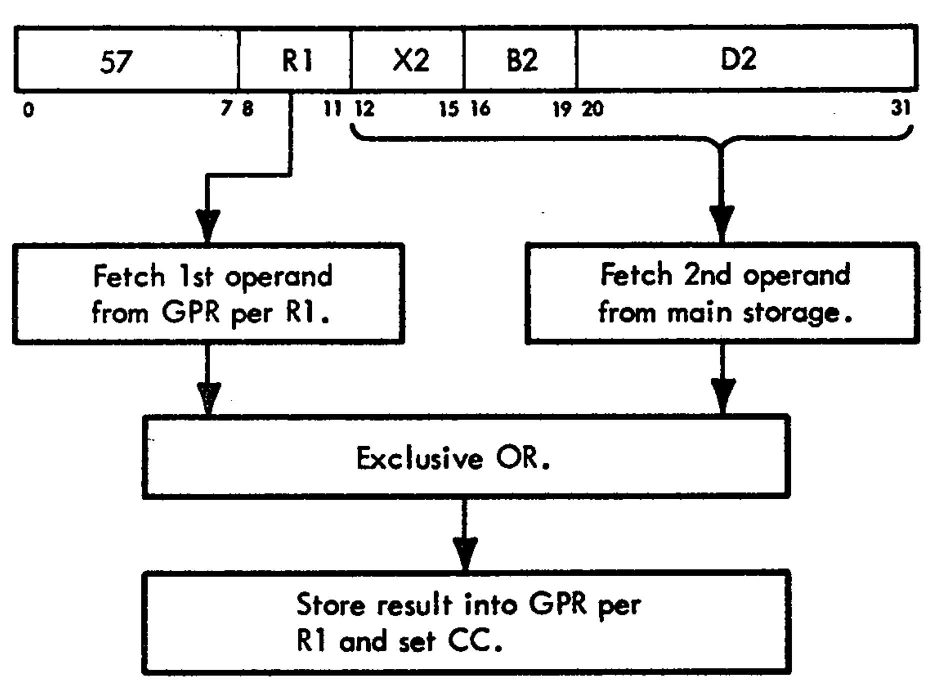
- SS format.
- Purpose: OR 1st operand (in storage) with 2nd operand (in storage) and place result into 1st operand location.

Diagram 5-405. Logical OR Instructions



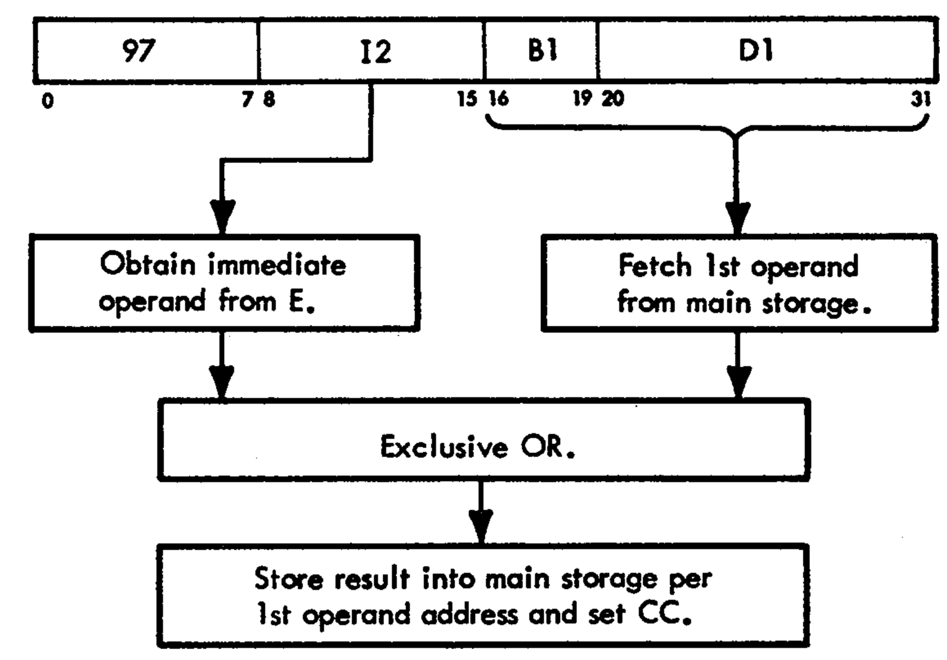
- RR format.
- Purpose: Exclusive-OR 1st operand (in GPR, per R1) with 2nd operand (in GPR, per R2) and place result into 1st operand location.

A. Exclusive-OR, XR (17)



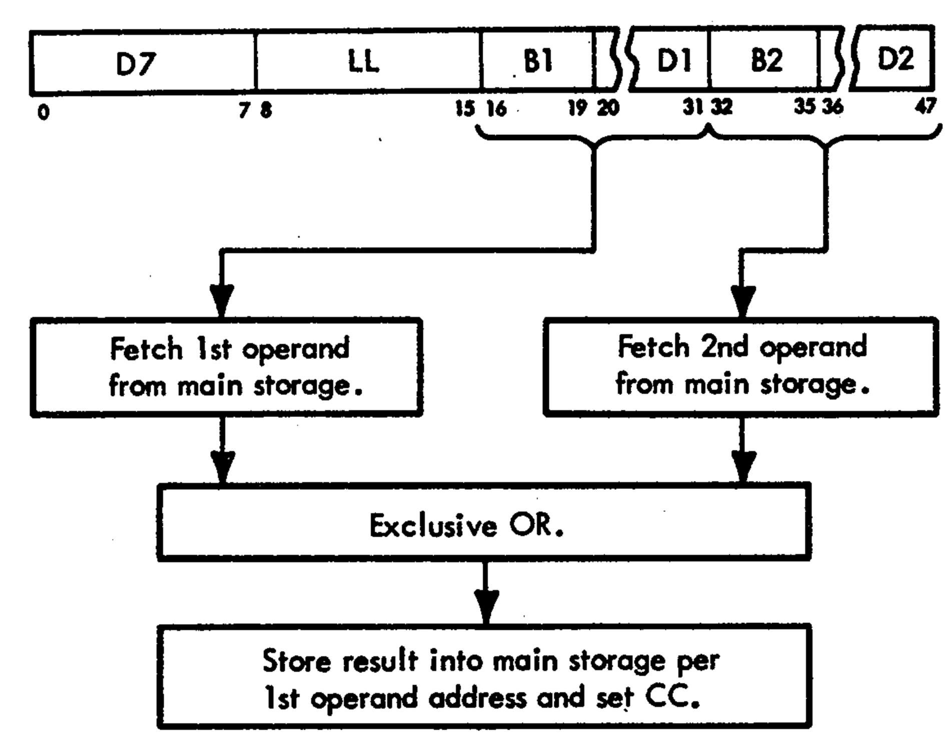
- RX format.
- Purpose: Exclusive-OR 1st operand (in GPR, per R1) with 2nd operand (in storage) and place result into 1st operand location.

B. Exclusive-OR, X (57)



- SI format.
- Purpose: Exclusive-OR immediate operand (I2 of instruction) with 1st operand (in storage) and place result into 1st operand location.

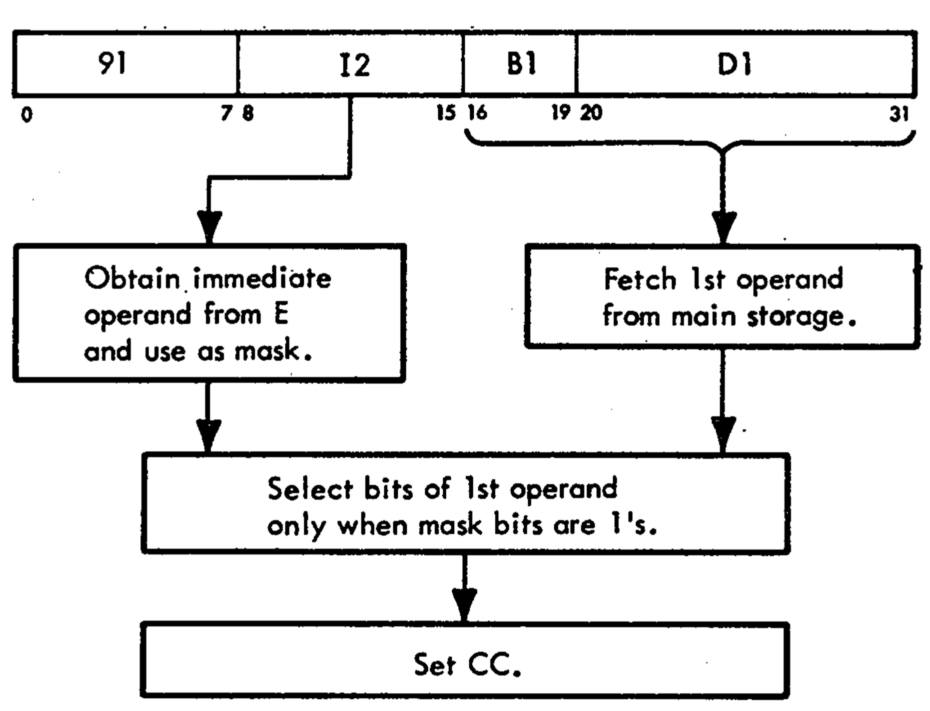
C. Exclusive-OR, XI (97)



- SS format.
- Purpose: Exclusive-OR 1st operand (in storage) with 2nd operand (in storage) and place result into 1st operand location.

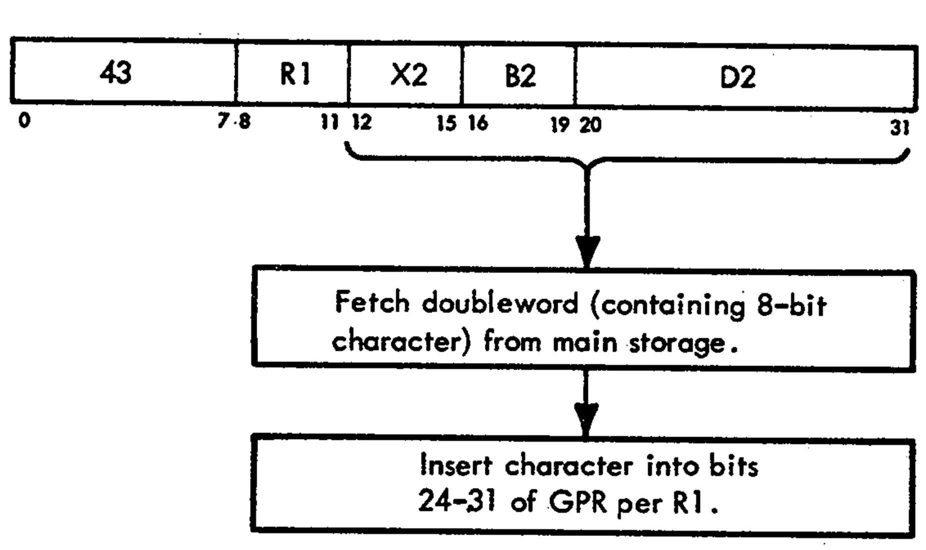
D. Exclusive-OR, XC (D7)

Diagram 5-406. Logical Exclusive-OR Instructions



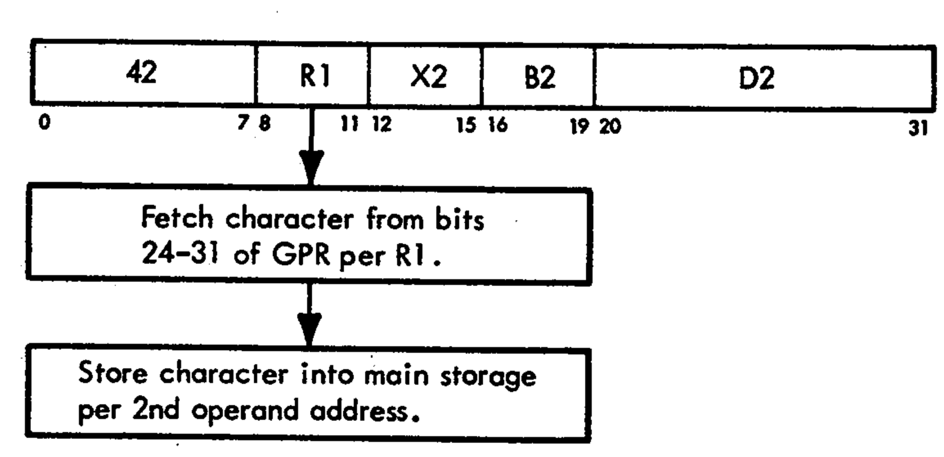
- SI format.
- Purpose: Set CC according to state of 1st operand bits (in storage) selected by mask bits (I2 of instruction).

Diagram 5-407. Test Under Mask, TM (91)



- RX format.
- Purpose: Insert 2nd operand (byte; in storage) into bits 24-31 of 1st operand location (in GPR, per R1).

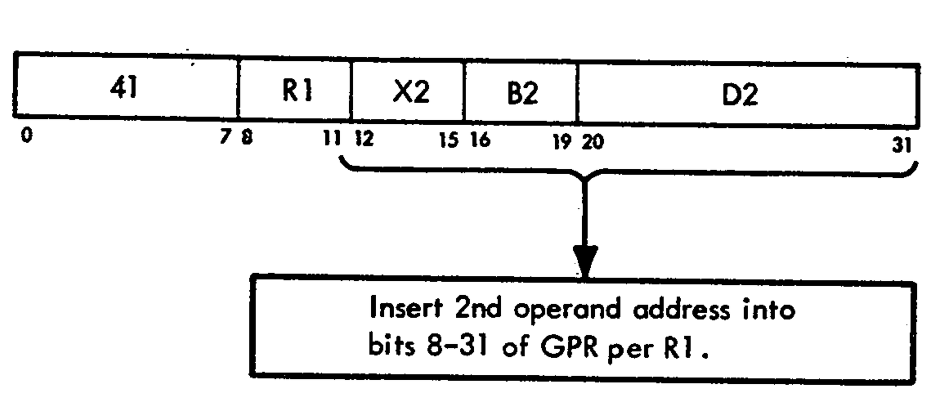
A. Insert Character, IC (43)



- RX format.
- Purpose: Store bits 24-31 of 1st operand (in GPR, per R1) into 2nd operand location (in storage).

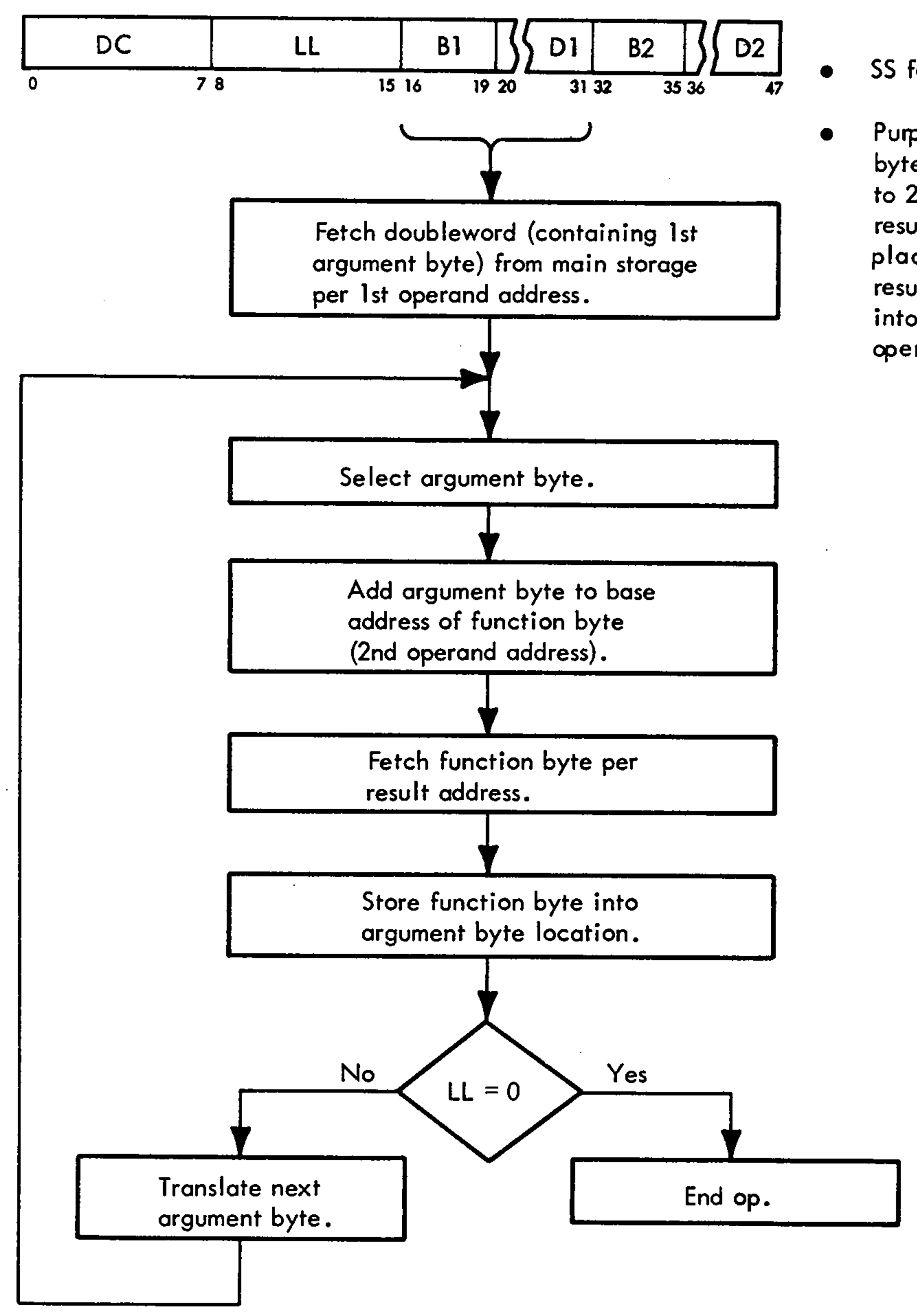
B. Store Character, STC (42)

Diagram 5-408. Insert Character, IC (43); Store Character, STC (42)



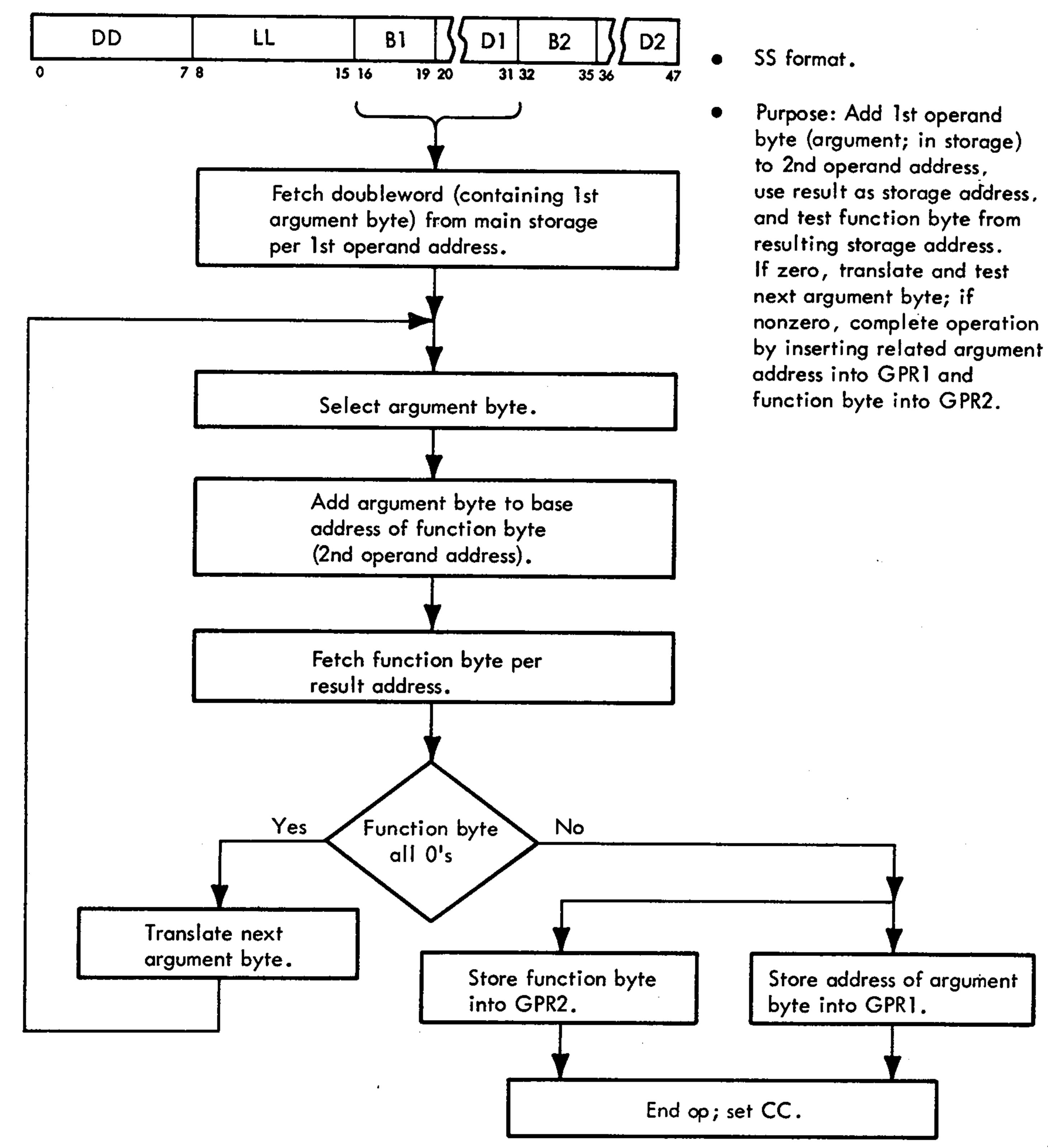
- RX format.
- Purpose: Insert 2nd operand address into bits 8-31 of GPR specified by R1.

Diagram 5-409. Load Address, LA (41)



- SS format.
- Purpose: Add 1st operand byte (argument; in storage) to 2nd operand address, use result as storage address, and place function byte from resulting storage address into corresponding 1st operand byte location.

A. Translate, TR (DC)



- SS format.
- Purpose: Add 1st operand byte (argument; in storage) to 2nd operand address, use result as storage address, and test function byte from resulting storage address. If zero, translate and test next argument byte; if nonzero, complete operation by inserting related argument address into GPR1 and function byte into GPR2.

B. Translate and Test, TRT (DD)

Diagram 5-410. Translate, TR (DC); Translate and Test, TRT (DD)

- SS format:

Op Code †	LL	B1	D1	B2	D2
0–7	8–15	16–19	20–31	32–35	36–47

† DE for Edit
DF for Edit and Mark

- Purpose:

1. Edit: change format of source (2nd operand; in storage) from packed to zoned, edit source under control of pattern (1st operand; in storage), and place result into 1st operand location
2. Edit and Mark: same as Edit, but, in addition, place location of each 1st significant digit into GPR1.

- Conditions at end of GIS:

1. Pattern (destination operand) is in ST.
2. Pattern address is in D.
3. Source address (contents of GPR per B2, + D2) is in IC.
4. 1st 16 bits of instruction are in E.

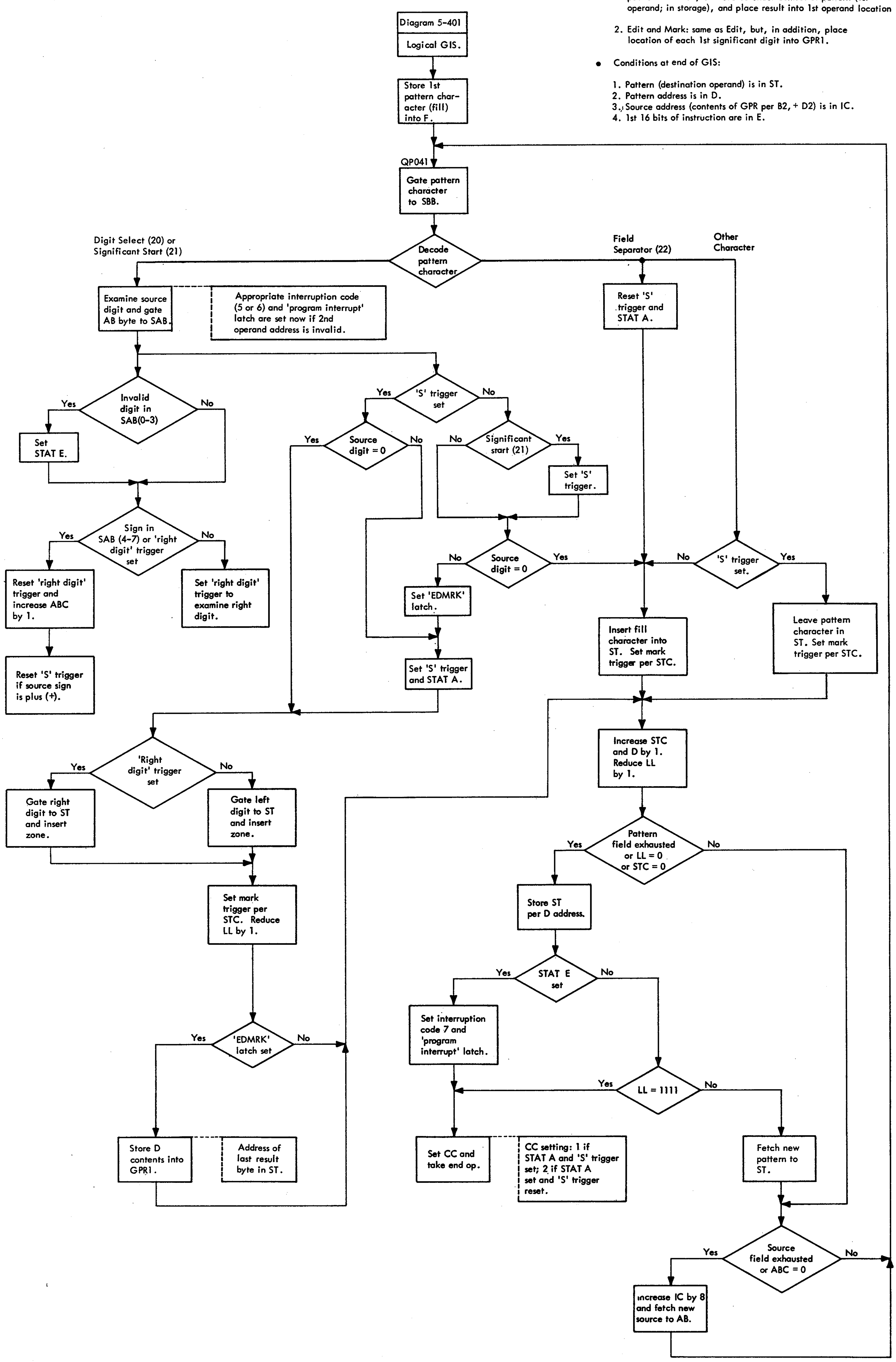
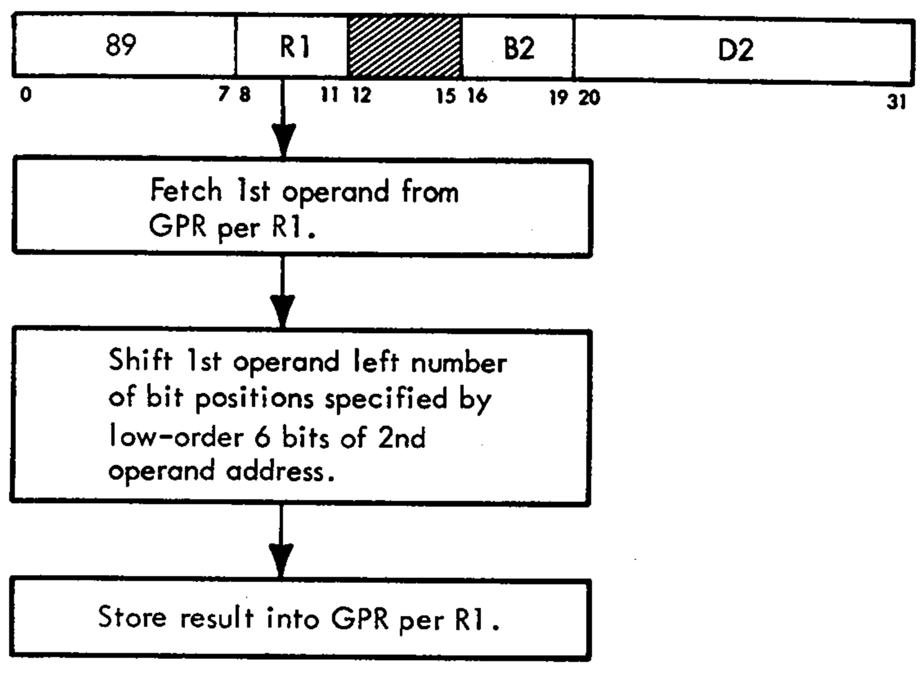
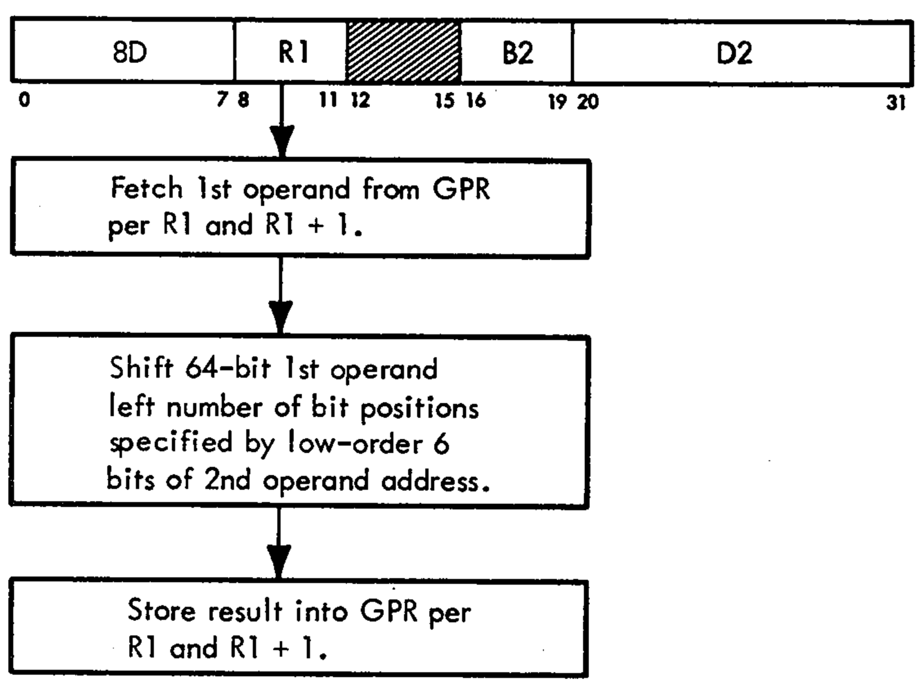


Diagram 5-411. Edit, ED (DE); Edit and Mark, EDMK (DF)



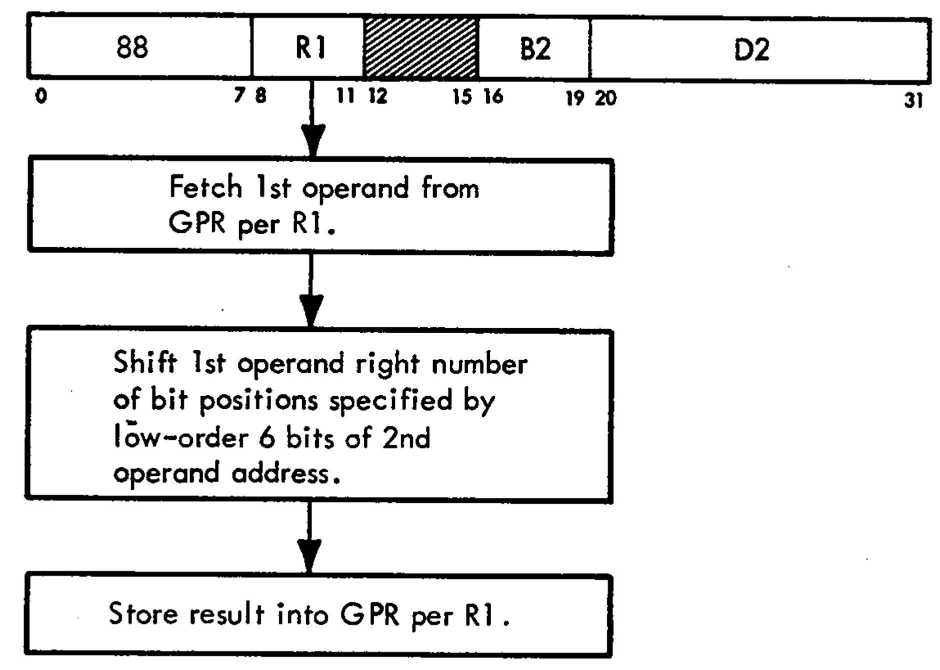
A. Shift Left Single, SLL (89)

- RS format.
- Purpose: Shift 1st operand (in GPR, per R1) left number of bit positions specified by low-order 6 bits of 2nd operand address.



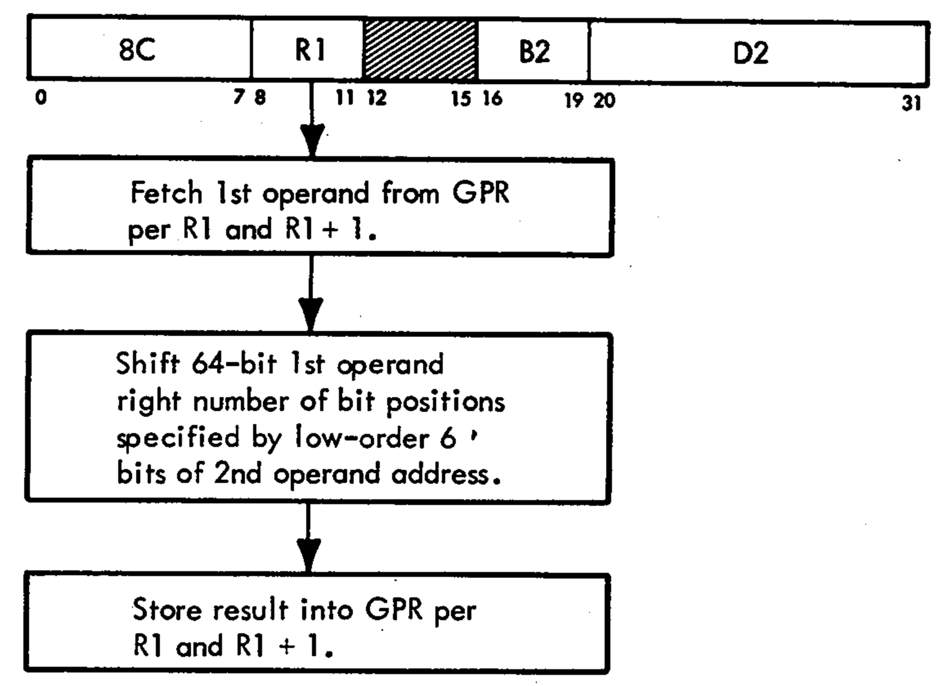
B. Shift Left Double, SLDL (8D)

- RS format.
- Purpose: Shift 1st operand (in GPR, per R1 and R1 + 1) left number of bit positions specified by low-order 6 bits of 2nd operand address.



C. Shift Right Single, SRL (88)

- RS format.
- Purpose: Shift 1st operand (in GPR, per R1) right number of bit positions specified by low-order 6 bits of 2nd operand address.



D. Shift Right Double, SRDL (8C)

- RS format.
- Purpose: Shift 1st operand (in GPR, per R1 and R1 + 1) right number of bit positions specified by low-order 6 bits of 2nd operand address.

Diagram 5-412. Logical Shift Instructions

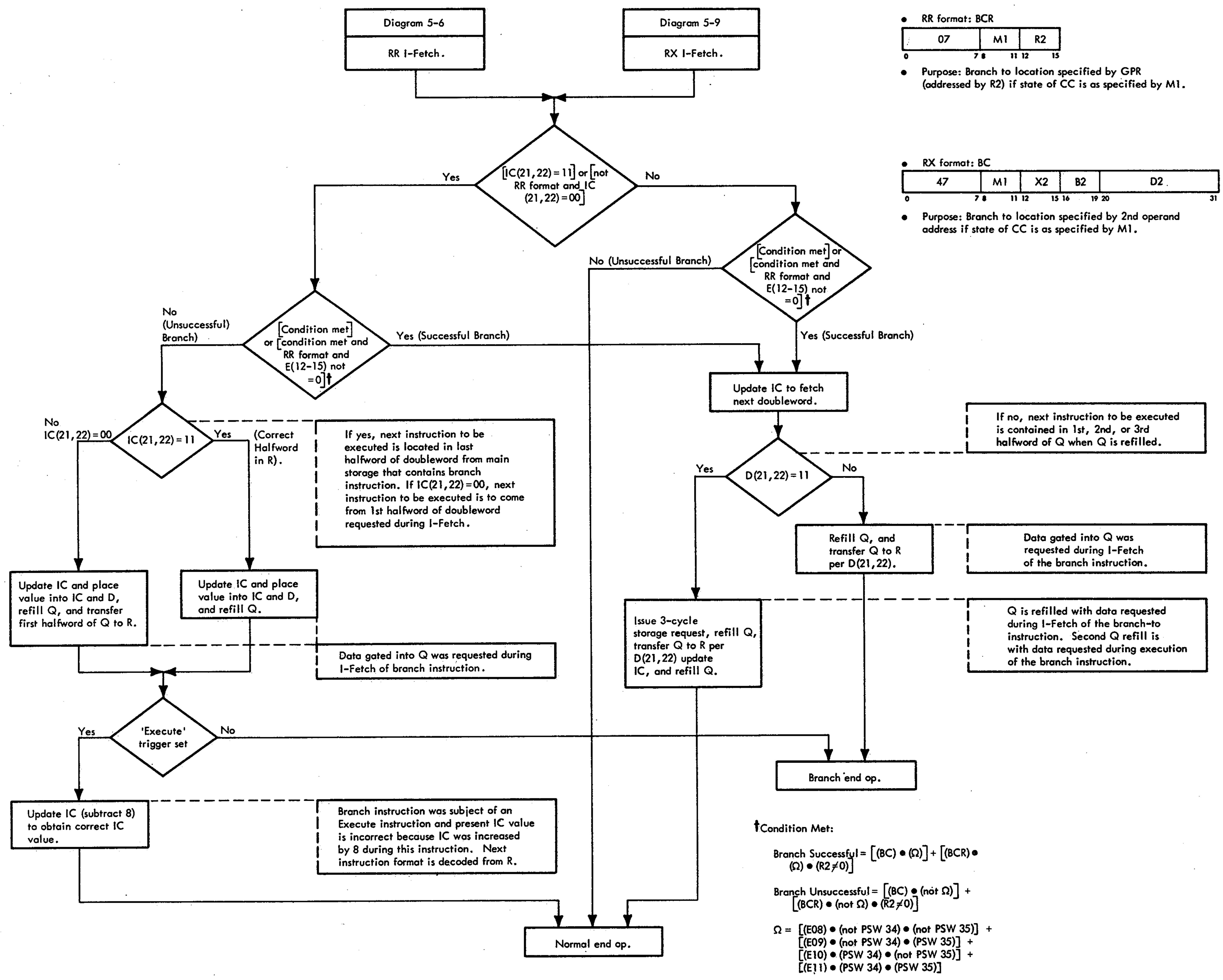
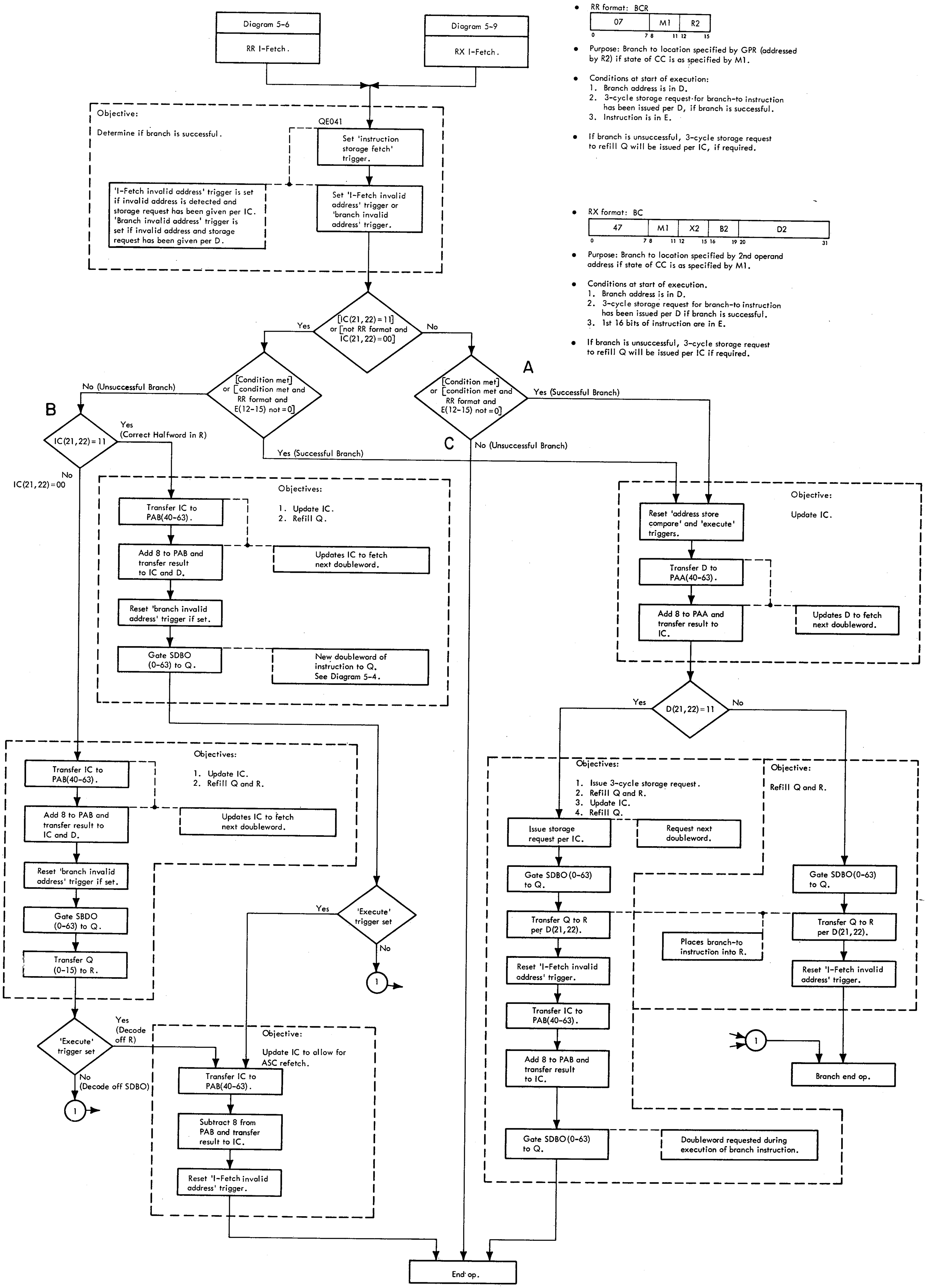


Diagram 5-501. Branch On Condition, BCR (07); BC (47) (Sheet 1 of 2)



- RR format: BCR

07	M1	R2
0–7	8–11	12–15

- Purpose: Branch to location specified by GPR (addressed by R2) if state of CC is as specified by M1.
- Conditions at start of execution:
  1. Branch address is in D.
  2. 3-cycle storage request for branch-to instruction has been issued per D, if branch is successful.
  3. Instruction is in E.
- If branch is unsuccessful, 3-cycle storage request to refill Q will be issued per IC, if required.

- RX format: BC

47	M1	X2	B2	D2
0–7	8–11	12–15	16–19	20–31

- Purpose: Branch to location specified by 2nd operand address if state of CC is as specified by M1.
- Conditions at start of execution.
  1. Branch address is in D.
  2. 3-cycle storage request for branch-to instruction has been issued per D if branch is successful.
  3. 1st 16 bits of instruction are in E.
- If branch is unsuccessful, 3-cycle storage request to refill Q will be issued per IC if required.

Diagram 5-501. Branch On Condition, BCR (07); BC (47) (Sheet 2 of 2)

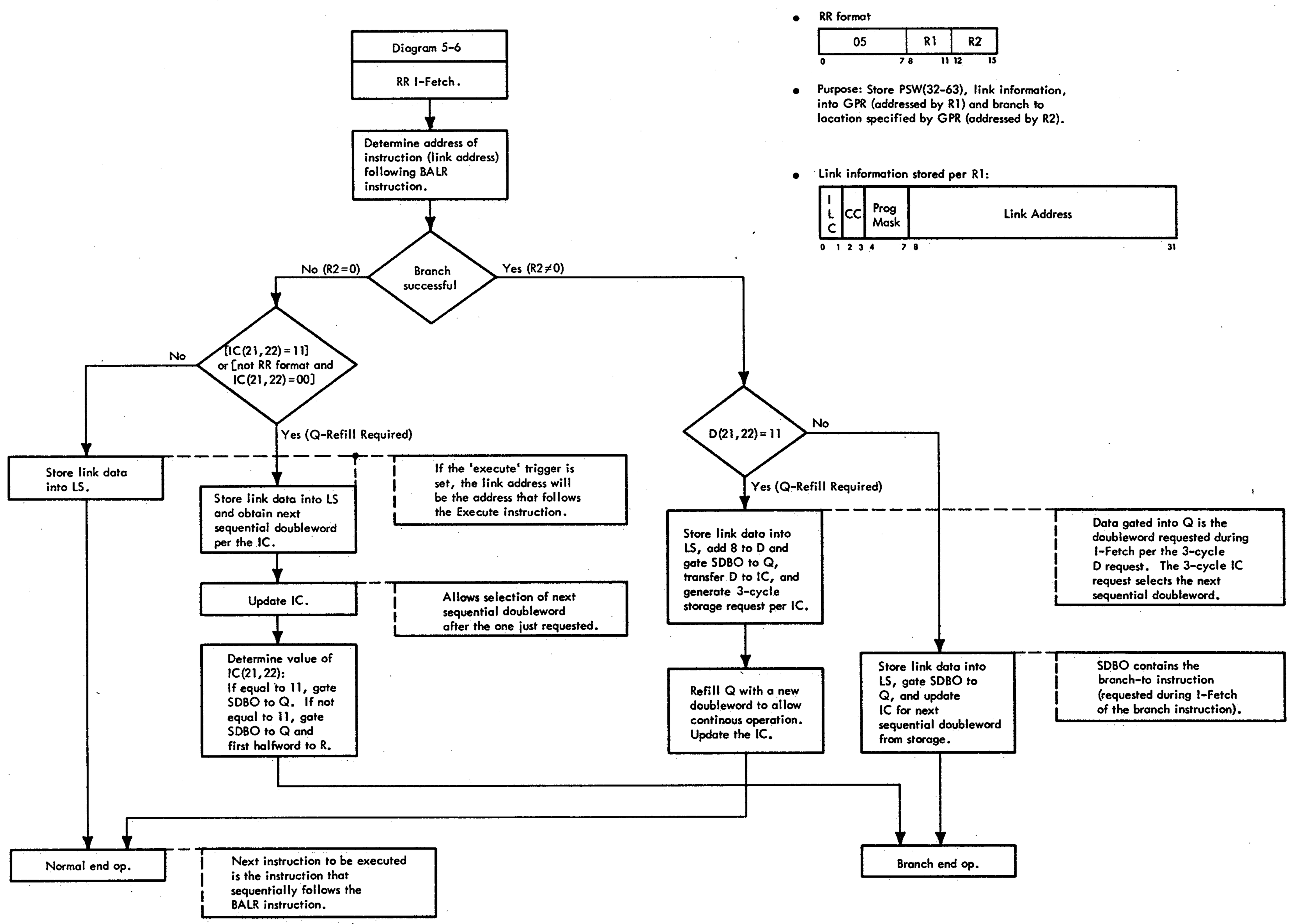


Diagram 5-502. Branch and Link, BALR (05) (Sheet 1 of 2)

- RR format:

05	R1	R2
0 – 7	8 – 11	12 – 15

- Purpose: Store PSW(32–63), link information, into GPR (addressed by R1) and branch to location specified by GPR (addressed by R2).
- Conditions at start of execution:
  1. 2nd operand is in A, B, and D.
  2. 3-cycle storage request has been issued per D for branch-to instruction.
  3. Instruction is in E.

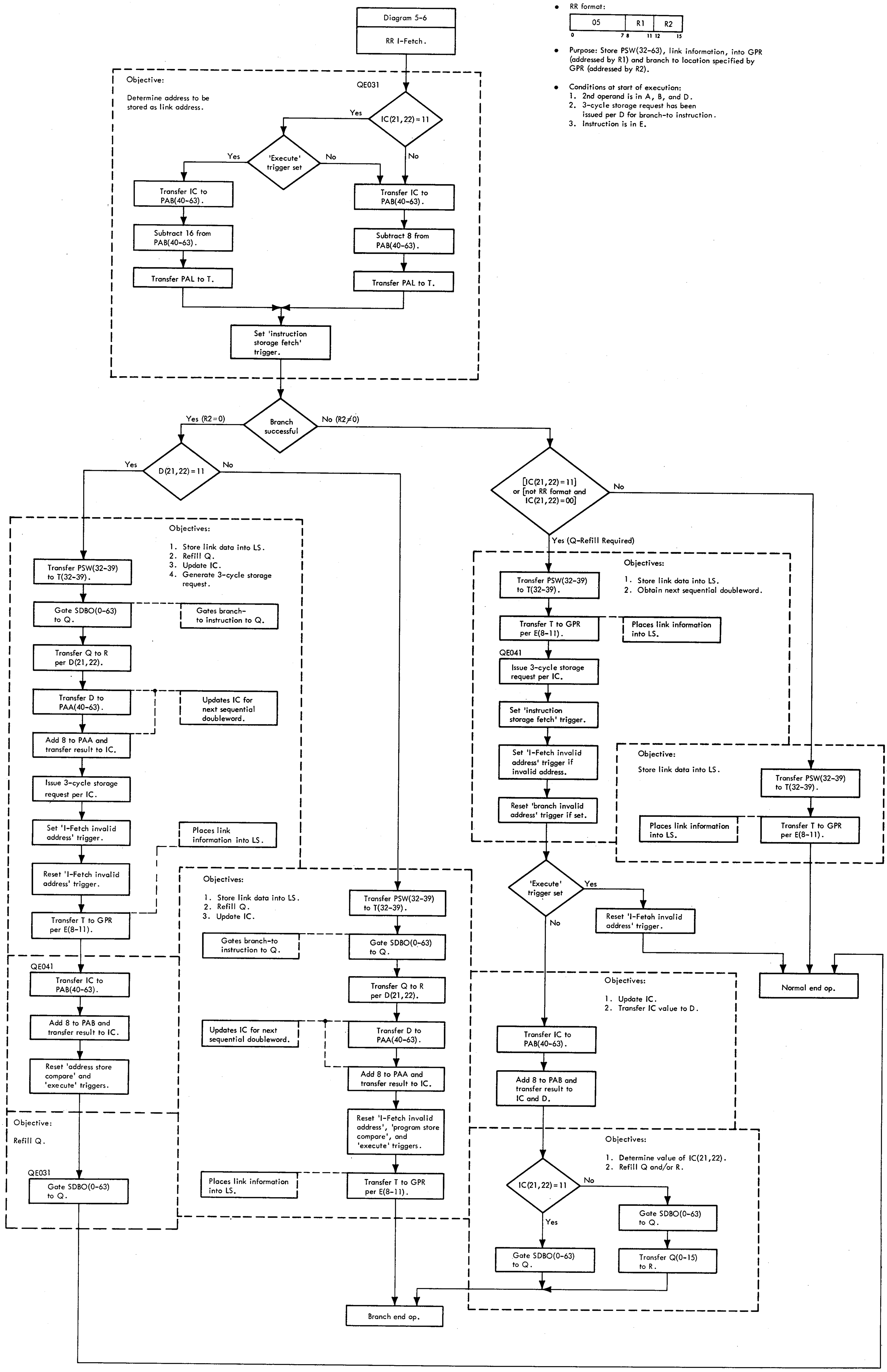


Diagram 5-502. Branch and Link BALR (05) (Sheet 2 of 2)

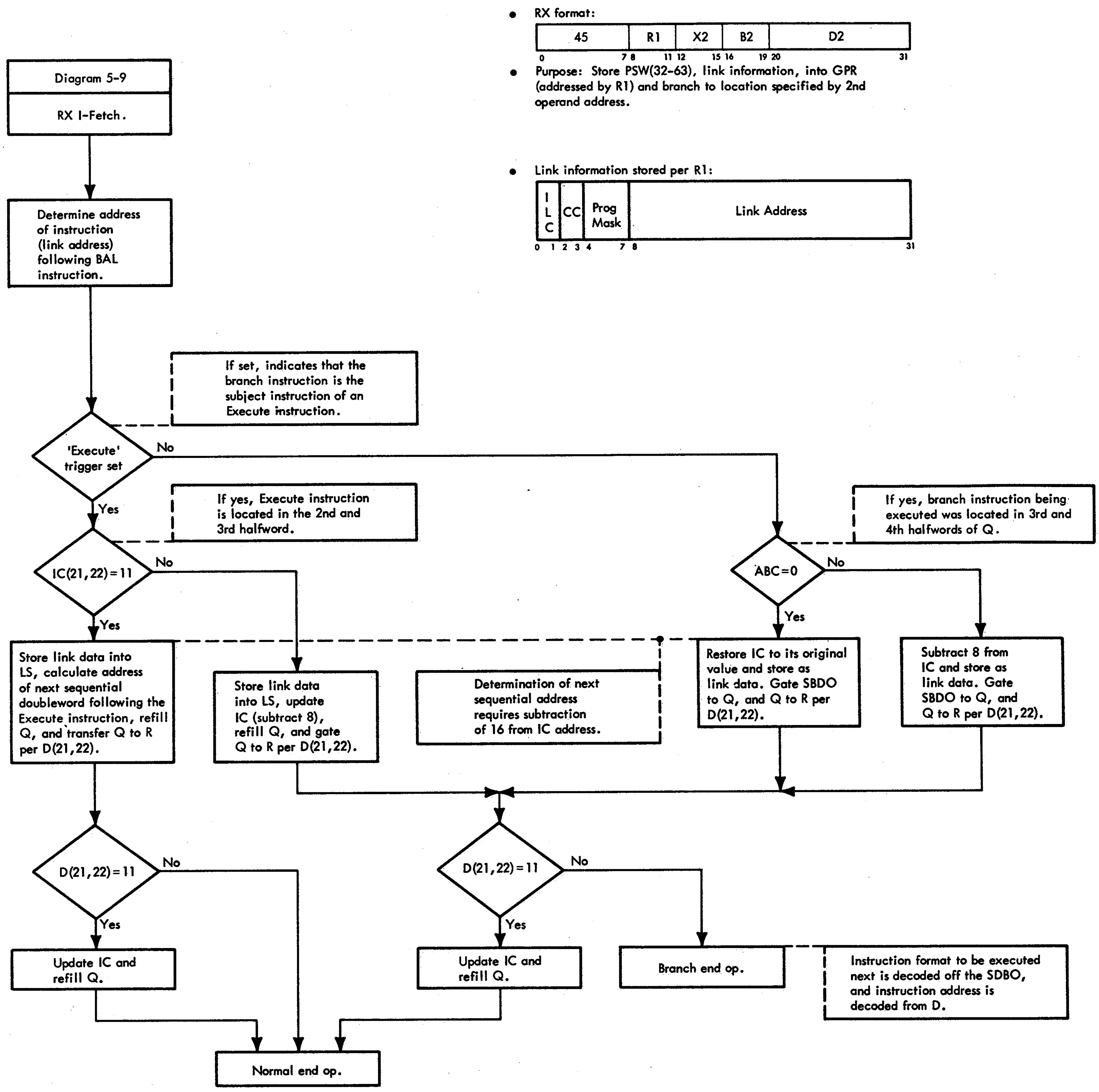


Diagram 5-503. Branch and Link, BAL (45) (Sheet 1 of 2)

- RX format:

45	R1	X2	B2	D2
0–7	8–11	12–15	16–19	20–31

- Purpose: Store PSW(32-63), link information, into GPR (addressed by R1) and branch to location specified by 2nd operand address.
- Conditions at start of execution:
  1. Branch address is in D.
  2. 3-cycle storage request has been issued per D for branch-to instruction.
  3. 1st 16 bits of instruction are in E.

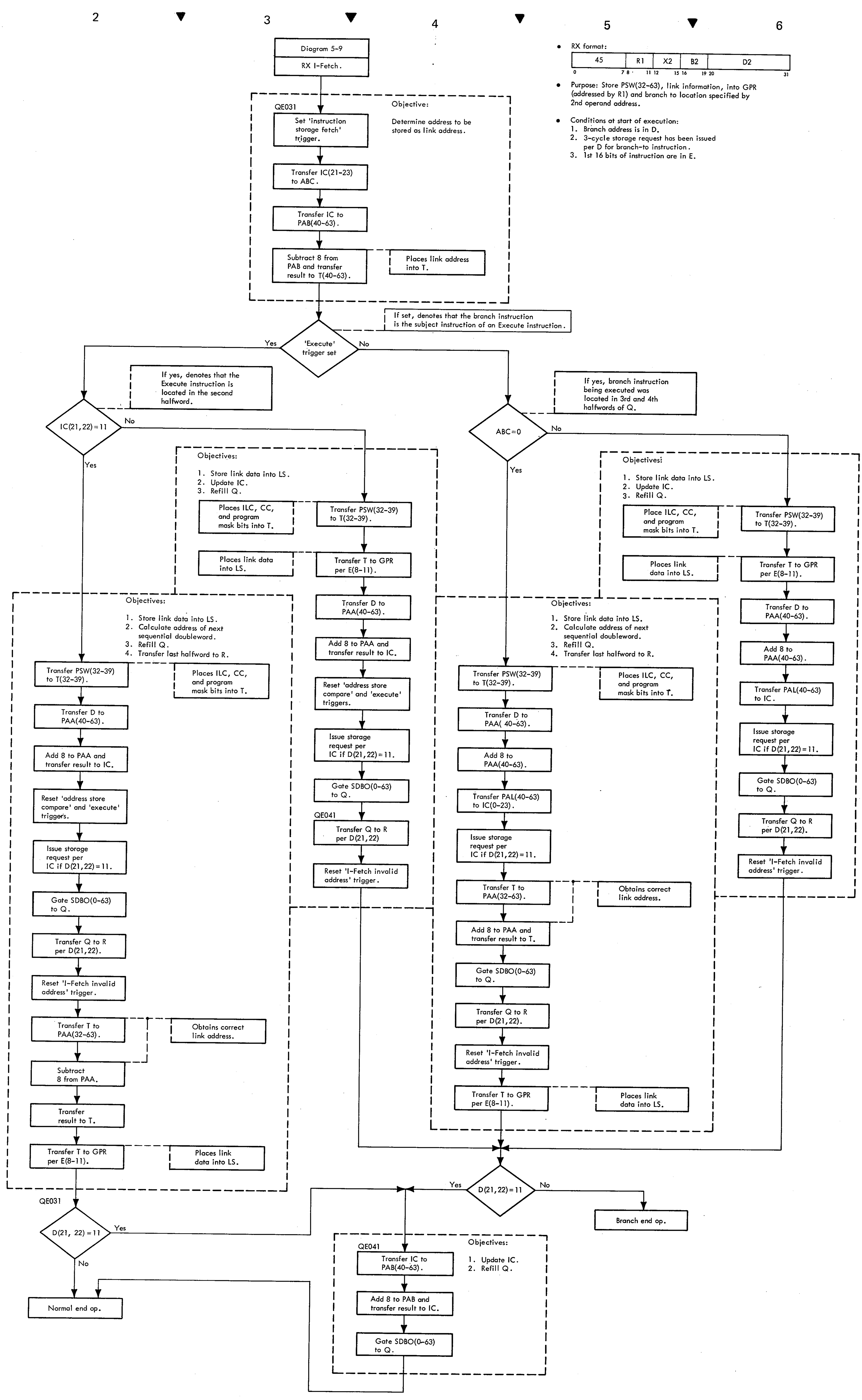


Diagram 5-503. Branch and Link, BAL (45) (Sheet 2 of 2)

- RR format: BCTR

06	R1	R2
0–7	8–11	12–15

- Purpose: Subtract 1 from 1st operand (in GPR, per R1) and, if result ≠ 0, branch to location specified by GPR (addressed by R2).

- RX format: BCT

46	R1	X2	B2	D2
0–7	8–11	12–15	16–19	20–31

- Purpose: Subtract 1 from 1st operand (in GPR, per R1) and, if result ≠ 0, branch to location specified by 2nd operand address.

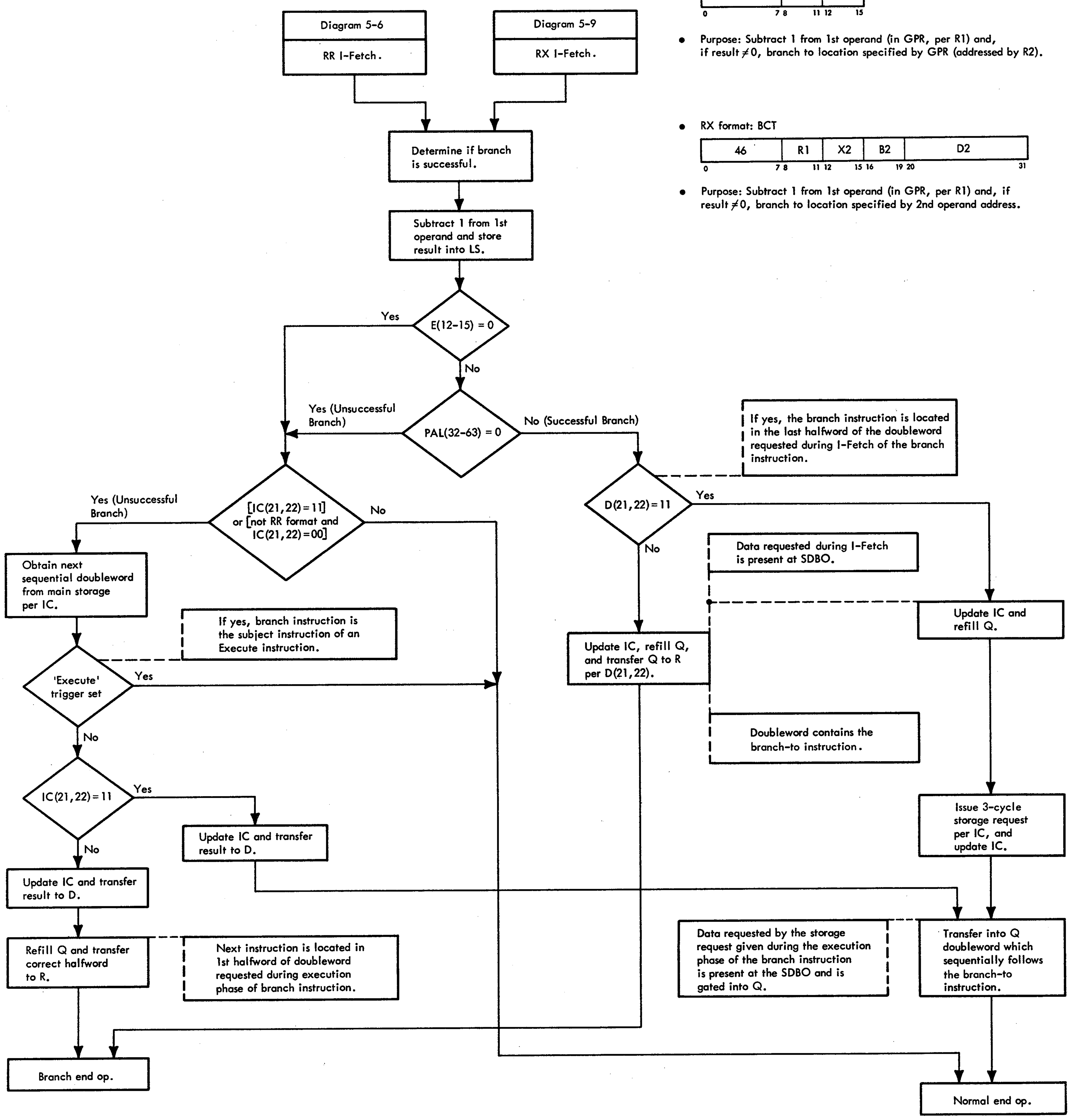
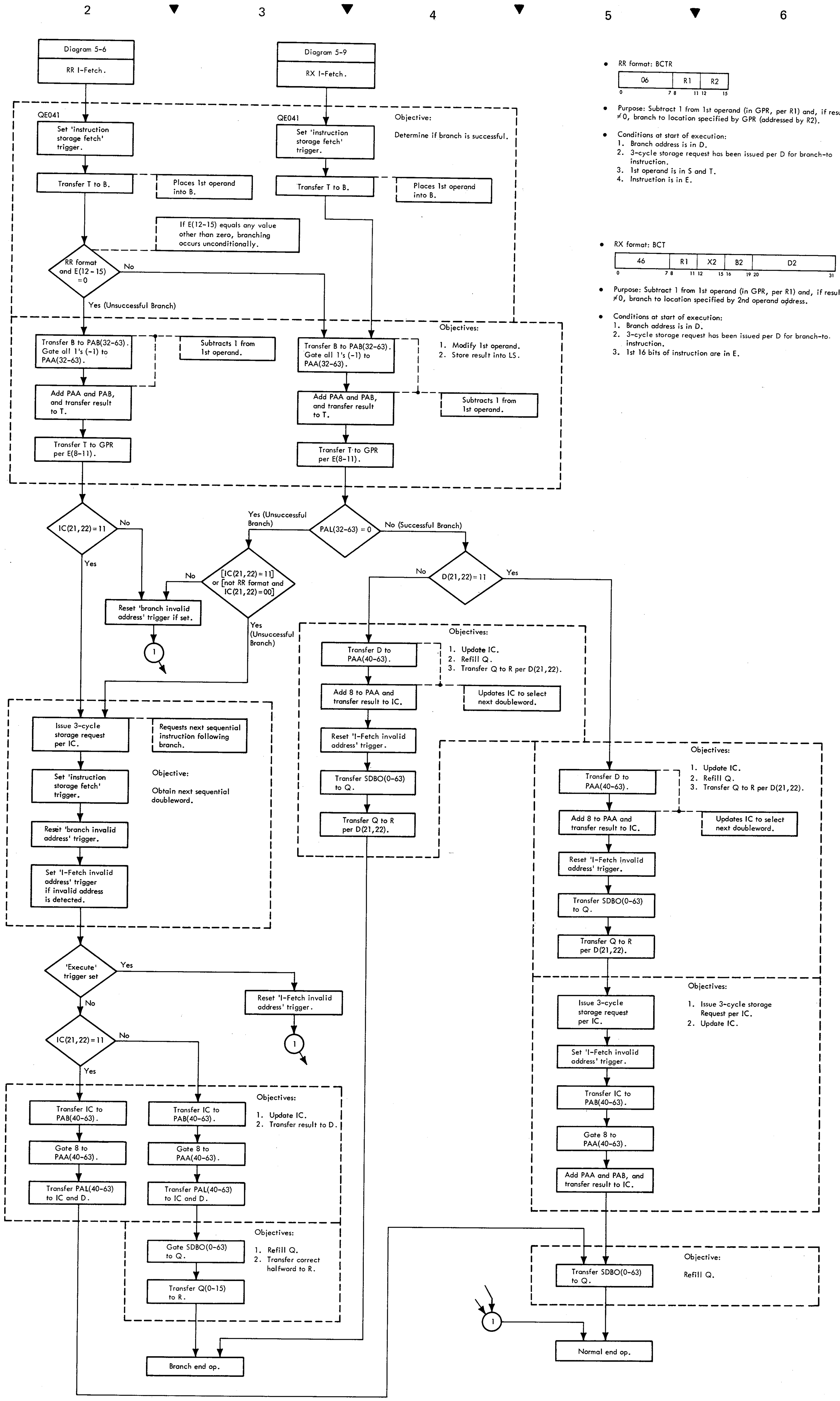


Diagram 5-504. Branch On Count, BCTR (06); BCT (46) (Sheet 1 of 2)



- RR format: BCTR

06	R1	R2
0 7	8 11	12 15

- Purpose: Subtract 1 from 1st operand (in GPR, per R1) and, if result ≠0, branch to location specified by GPR (addressed by R2).
- Conditions at start of execution:
  1. Branch address is in D.
  2. 3-cycle storage request has been issued per D for branch-to instruction.
  3. 1st operand is in S and T.
  4. Instruction is in E.

- RX format: BCT

46	R1	X2	B2	D2
0 7	8 11	12 15	16 19	20 31

- Purpose: Subtract 1 from 1st operand (in GPR, per R1) and, if result ≠0, branch to location specified by 2nd operand address.
- Conditions at start of execution:
  1. Branch address is in D.
  2. 3-cycle storage request has been issued per D for branch-to instruction.
  3. 1st 16 bits of instruction are in E.

Diagram 5-504. Branch On Count, BCTR (06); BCT (46) (Sheet 2 of 2)

- RS format:

86,87	R1	R3	B2	D2
0 7	8 11	12 15	16 19	20 31

- Purpose:

1. BXH - Add increment (3rd operand; in GPR, per R3) to 1st operand (in GPR, per R1), algebraically compare result (index) with comparand (in odd-address GPR specified by R3 or R3 + 1), and, if index is greater than comparand, branch to location specified by 2nd operand address.

2. BXLE - Add increment (3rd operand; in GPR, per R3) to 1st operand (in GPR, per R1), algebraically compare result (index) with comparand (in odd-address GPR specified by R3 or R3 + 1), and, if index is equal to or less than comparand, branch to location specified by 2nd operand address.

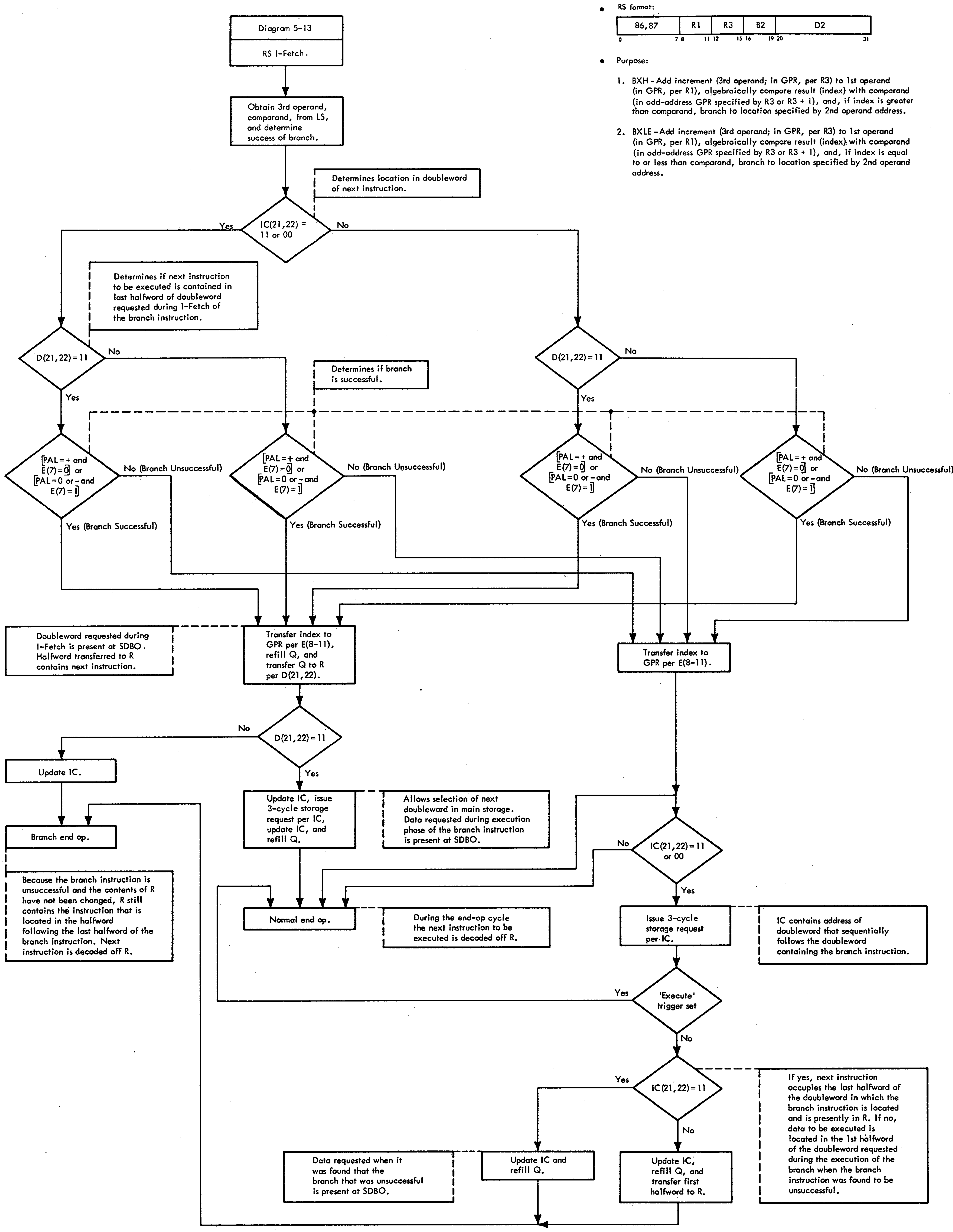


Diagram 5-505. Branch on Index High, BXH (86); Branch on Index Low or Equal, BXLE (87) (Sheet 1 of 3)

- RS format:

86,87	R1	R3	B2	D2
0 7	8 11	12 15	16 19	20 31

- Purpose:

1. BXH - Add increment (3rd operand; in GPR, per R3) to 1st operand (in GPR, per R1), algebraically compare result (index) with comparand (in odd-address GPR specified by R3 or R3 + 1), and, if index is greater than comparand, branch to location specified by 2nd operand address.
2. BXLE - Add increment (3rd operand; in GPR, per R3) to 1st operand (in GPR, per R1), algebraically compare result (index) with comparand (in odd-address GPR specified by R3 or R3 + 1), and, if index is equal to or less than comparand, branch to location specified by 2nd operand address.

- Conditions at start of execution:

1. 1st operand is in S and T.
2. Branch address is in D.
3. 3-cycle storage request has been issued per D for the branch-to instruction.
4. 1st 16 bits of instruction are in E.

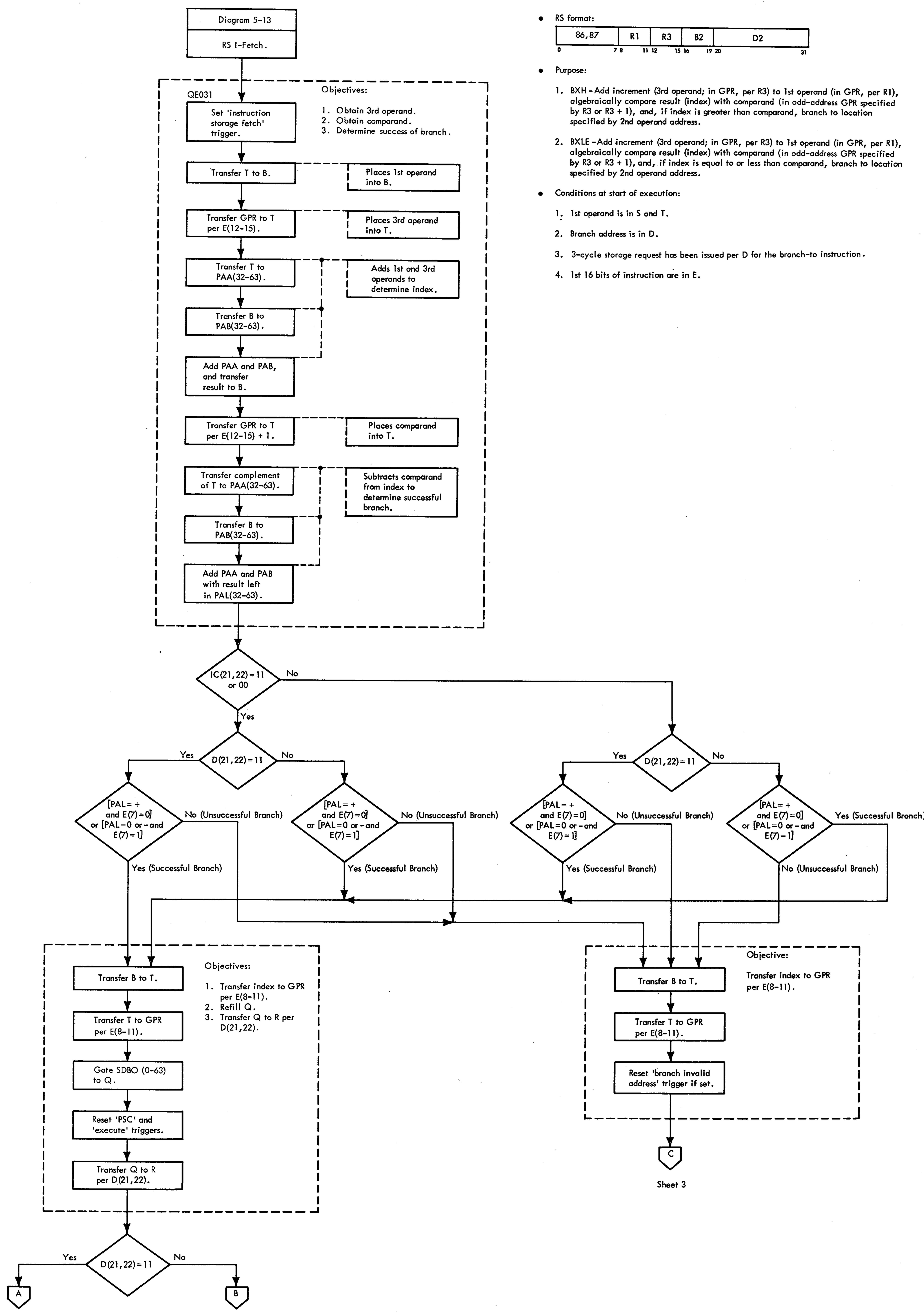


Diagram 5-505. Branch on Index High, BXH (86); Branch on Index Low or Equal, BXLE (87) (Sheet 2 of 3)

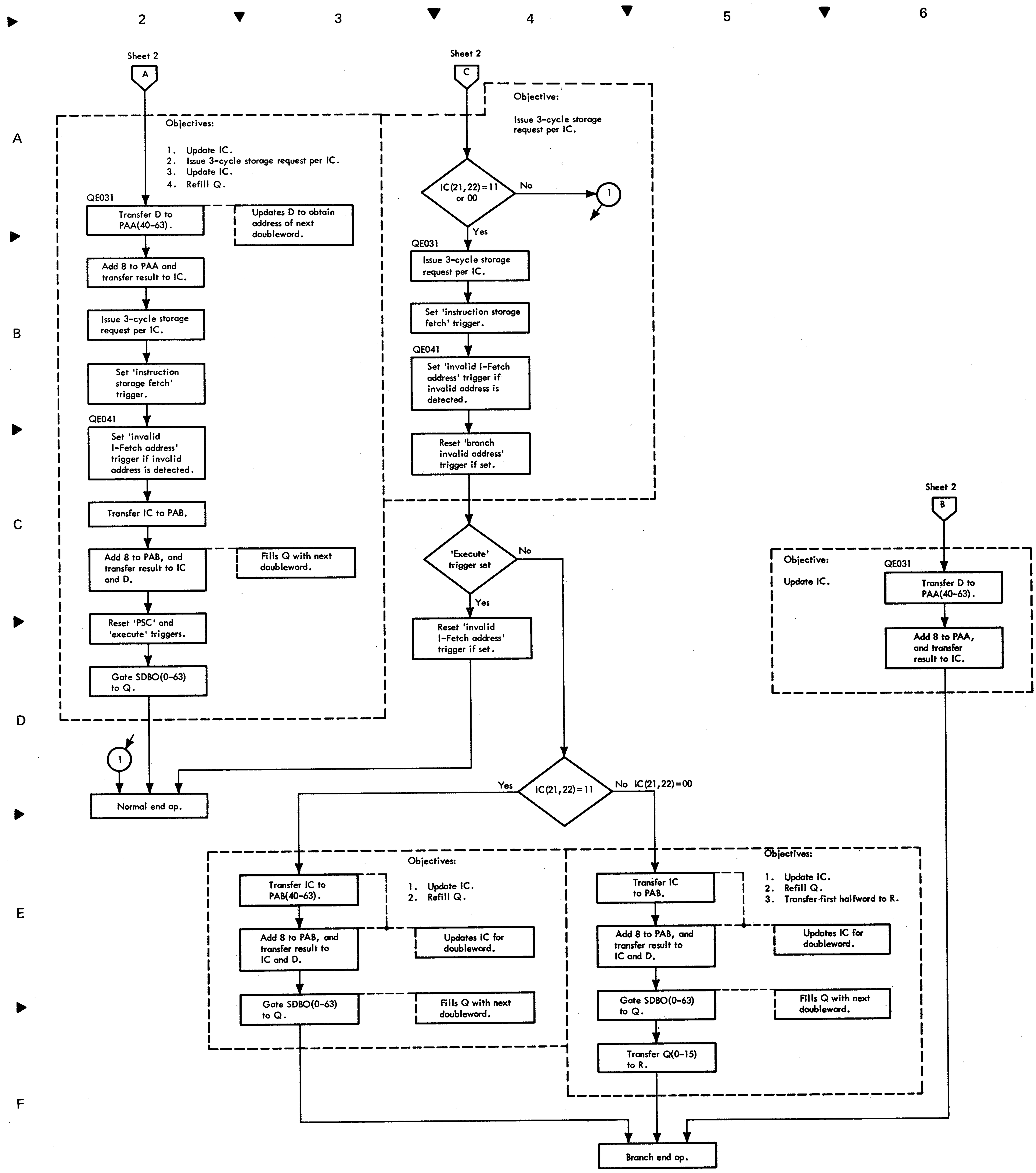


Diagram 5-505. Branch on Index High, BXH (86); Branch on Index Low or Equal, BXLE (87) (Sheet 3 of 3)

- RX format:

44	R1	X2	B2	D2
0–7	8–11	12–15	16–19	20–31

- Purpose: Execute subject instruction at location specified by 2nd operand address. Subject instruction may be modified by 1st operand (in GPR, per R1) if E(8-11) is not equal to zero.

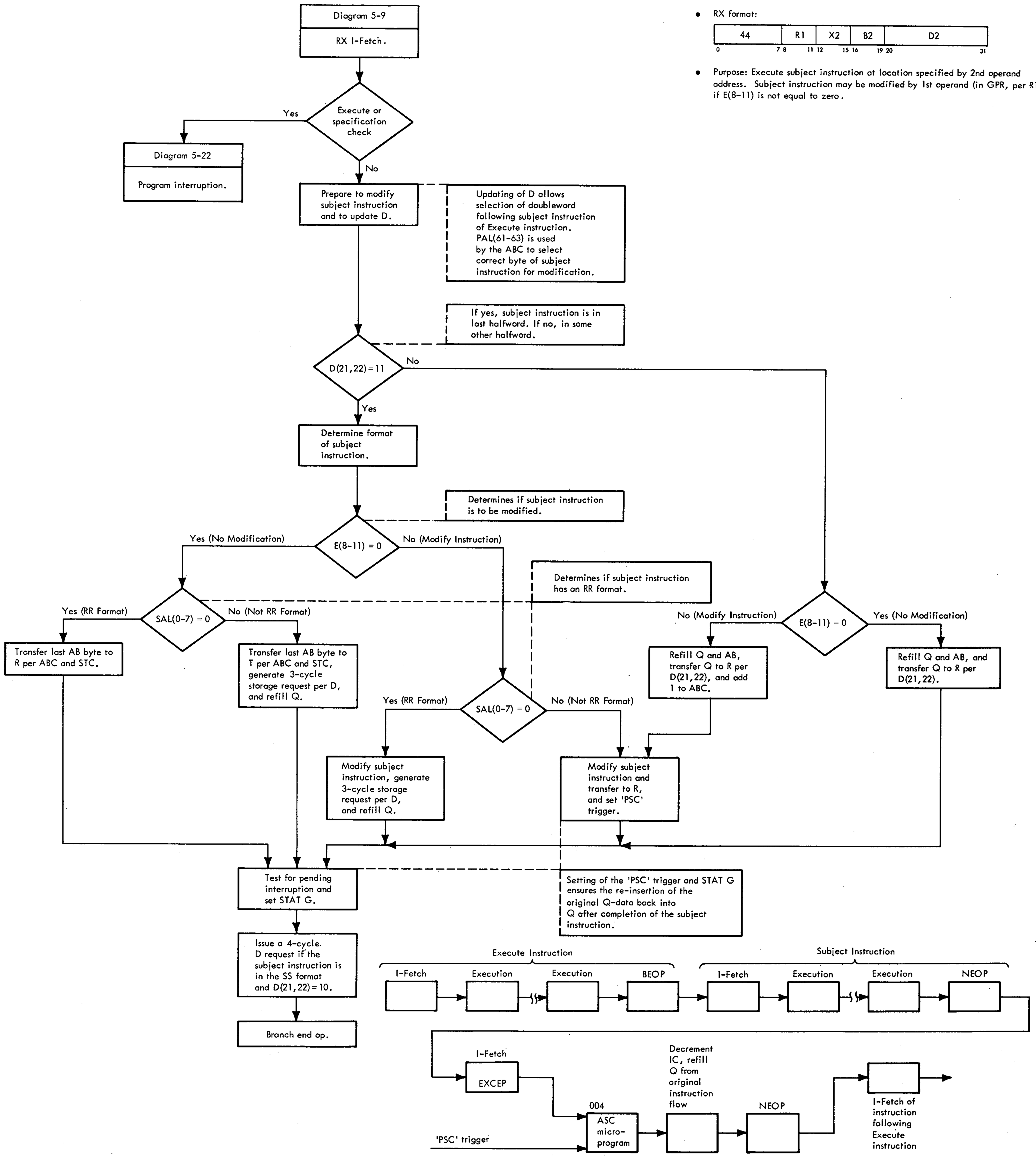


Diagram 5-506. Execute, EX (44) (Sheet 1 of 2)

- RX format:

44	R1	X2	B2	D2
0 7	8 11	12 15	16 19	20 31

- Purpose: Execute subject instruction at location specified by 2nd operand address. Subject instruction may be modified by 1st operand (in GPR, per R1) if E(8-11) is not equal to zero.
- Conditions at start of execution:
  1. 1st operand is in S and T.
  2. Address of subject instruction is in D.
  3. 3-cycle storage request for subject instruction has been issued per D.
  4. 1st 16 bits of instruction are in E.

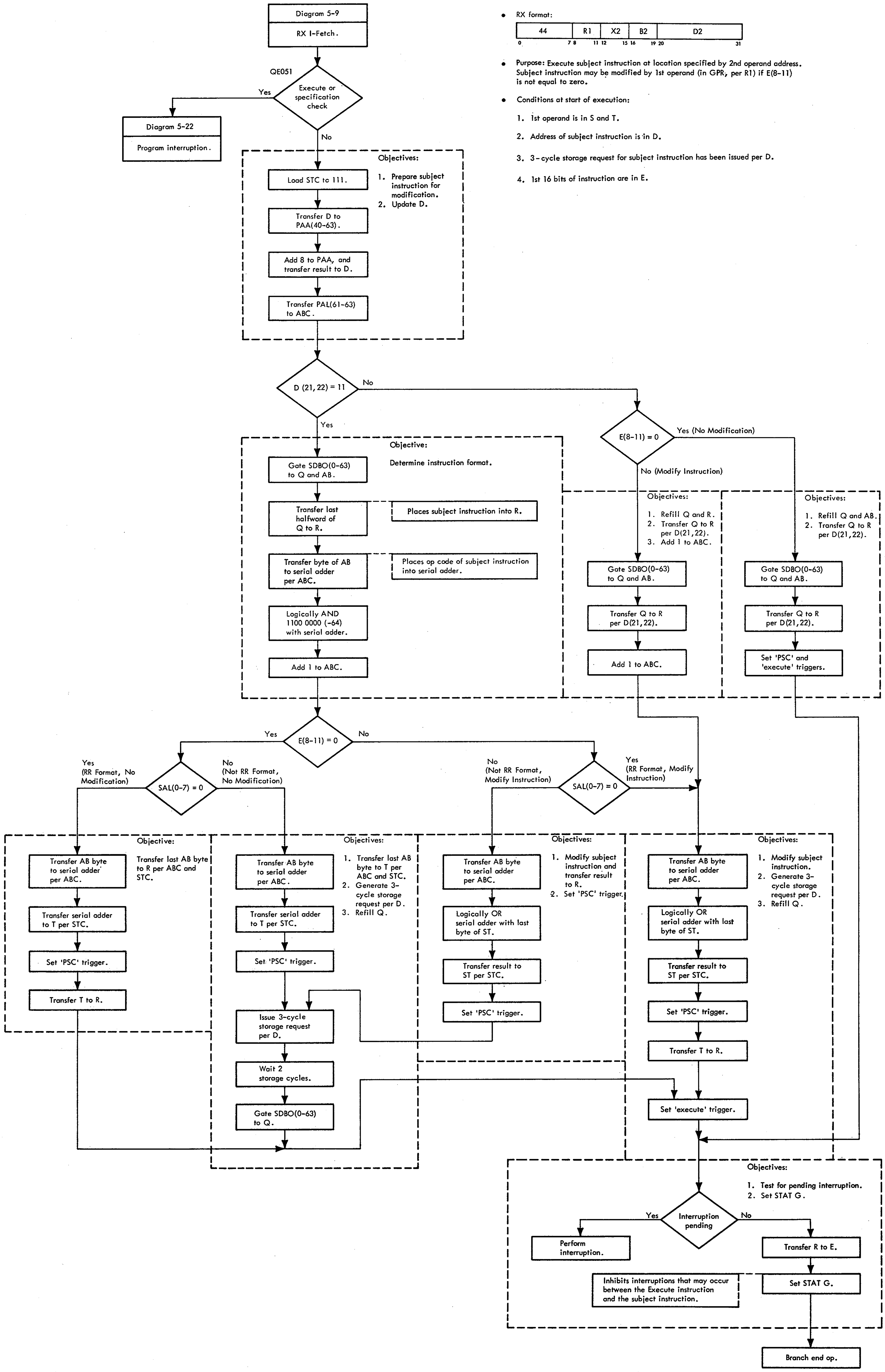


Diagram 5-506. Execute, EX (44) (Sheet 2 of 2)

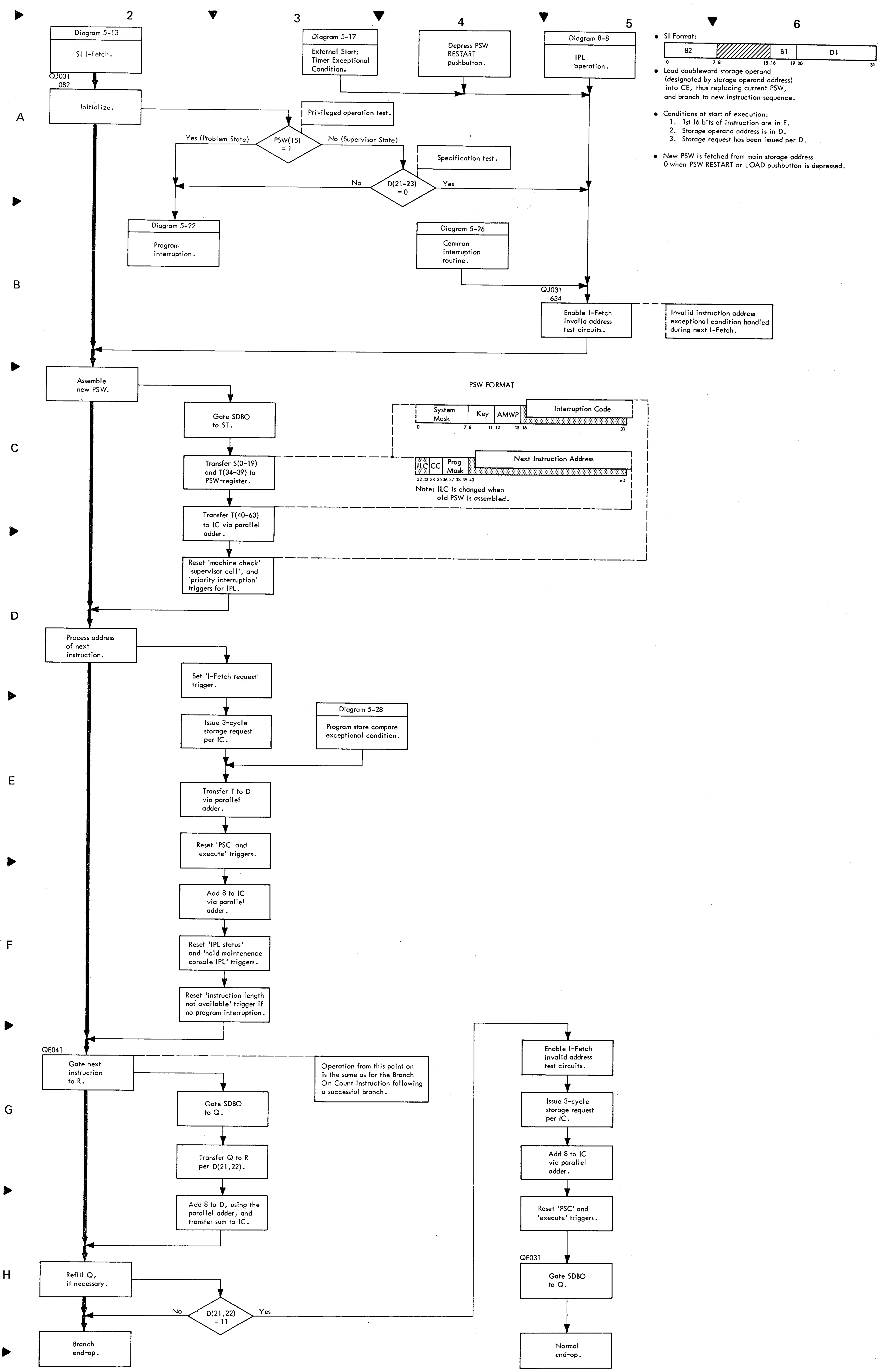


Diagram 5-601. Load PSW, LPSW (82)

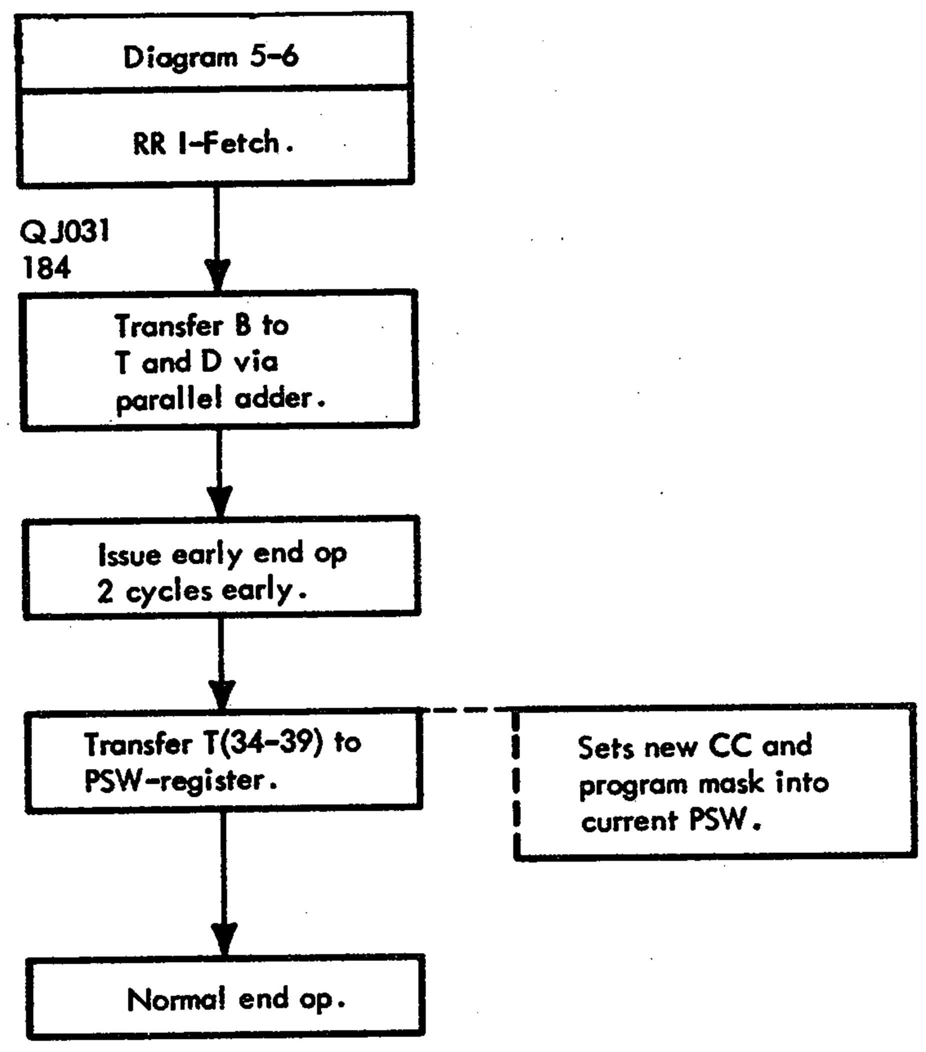
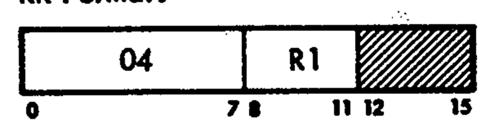


Diagram 5-602. Set Program Mask, SPM (04)

- RR Format:



- Replace CC and program mask (bits 34-39) of current PSW with bits 2-7 of 1st operand (in GPR, per R1).
- Conditions at start of execution:
  1. Instruction is in E.
  2. 1st operand is in A, B, and D.
  3. 2nd operand is not used.
- Bits 2-7 of 1st operand may have been loaded from PSW-register by a previous Branch and Link instruction.
- Program mask format (set mask bit permits interruption):
  Bit 36 - Fixed-point overflow mask.
  Bit 37 - Decimal overflow mask.
  Bit 38 - Exponent underflow (floating-point) mask.
  Bit 39 - Significance (floating-point) mask.

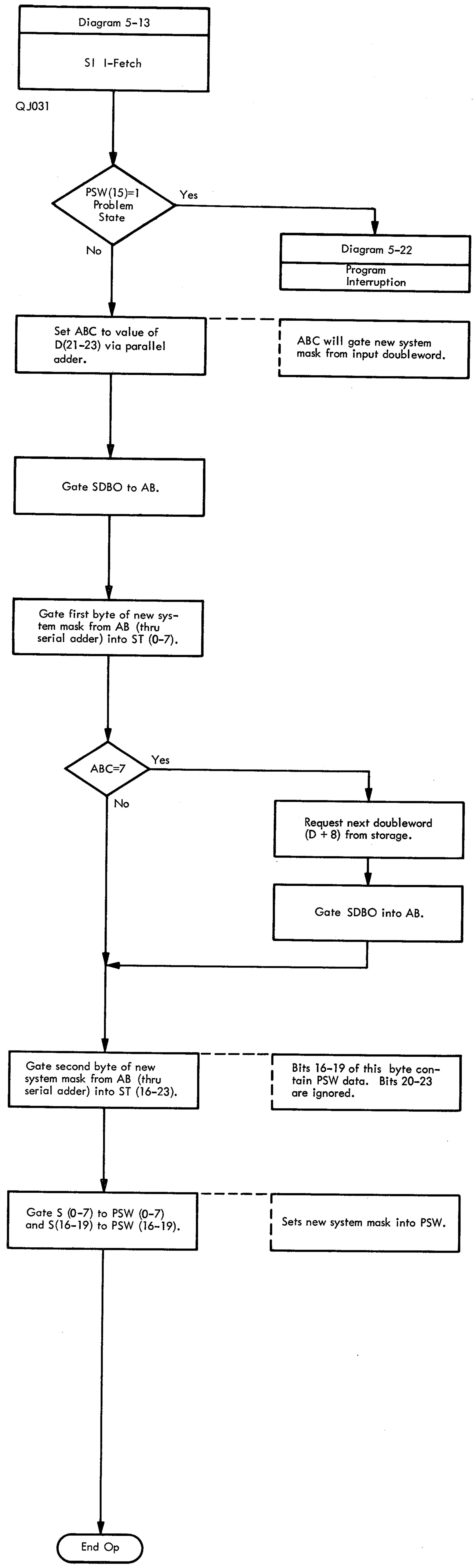


Diagram 5-603. Set System Mask, SSM (80)

- SI Format

80 (0–7)	(8–15)	B1 (16–19)	D1 (20–31)

- Purpose: Replace system mask (bits 0-7 and 16-19) of current PSW with byte designated by storage operand address and bits 0-3 of the following byte.

- Conditions at start of execution:
  1. First 16 bits of instruction are in E.
  2. Storage operand address is in D.
  3. Storage request has been issued per D.
- System mask format:

System mask (0–7)	Key (8–15)	AMWP (12–15)	System mask (16–20)	(20–63)

0 - Multiplexer channel A mask bit
1 - Selector channel 1A mask bit
2 - Selector channel 2A mask bit
3 - Selector channel 3A mask bit
4 - Multiplexer channel B mask bit
5 - Selector channel 1B mask bit
6 - Selector channel 2B mask bit
7 - External mask bit
16 - Selector channel 3B mask bit
17 - Multiplexer channel C mask bit
18 - Selector channel 1C mask bit
19 - Selector channel 2C mask bit

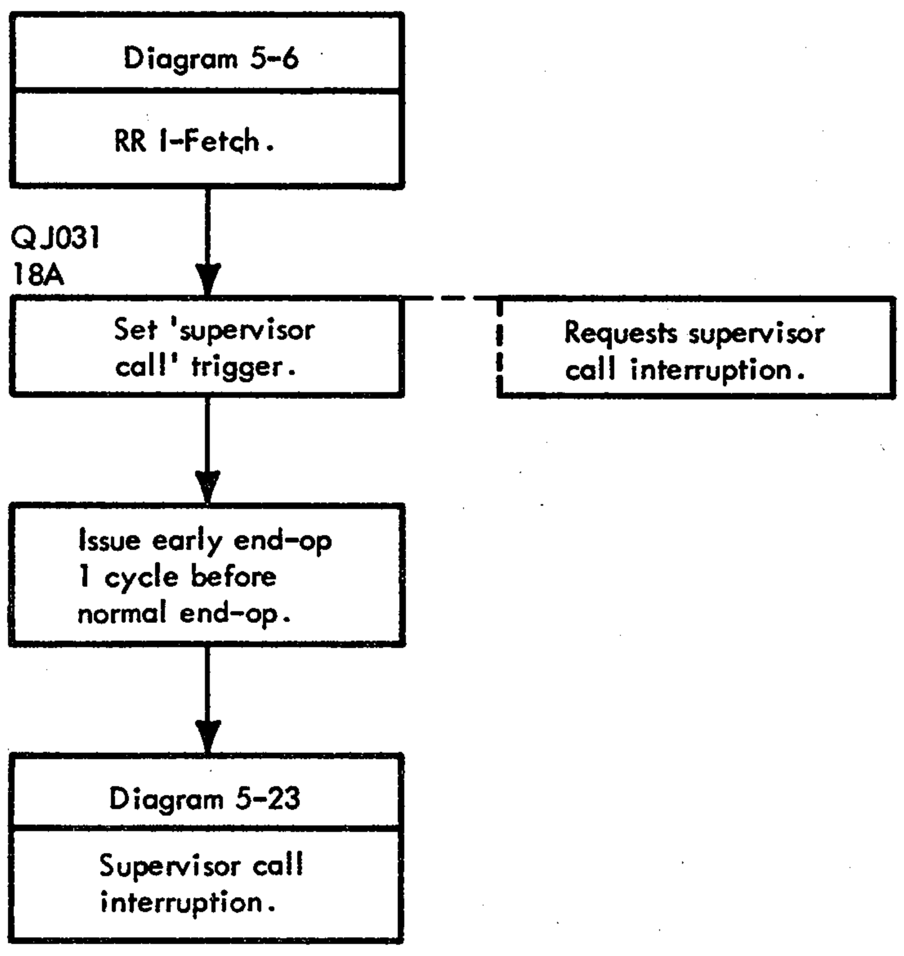
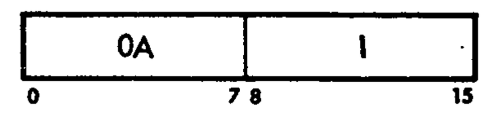


Diagram 5-604. Supervisor Call, SVC (0A)

- RR Format:



- Cause supervisor call interruption; replace old-PSW(24-31) with I-field (bits 8-15) of instruction, providing interruption code.
- Conditions at start of execution:
  1. Instruction is in E.
  2. E(8-15) contains interruption code.

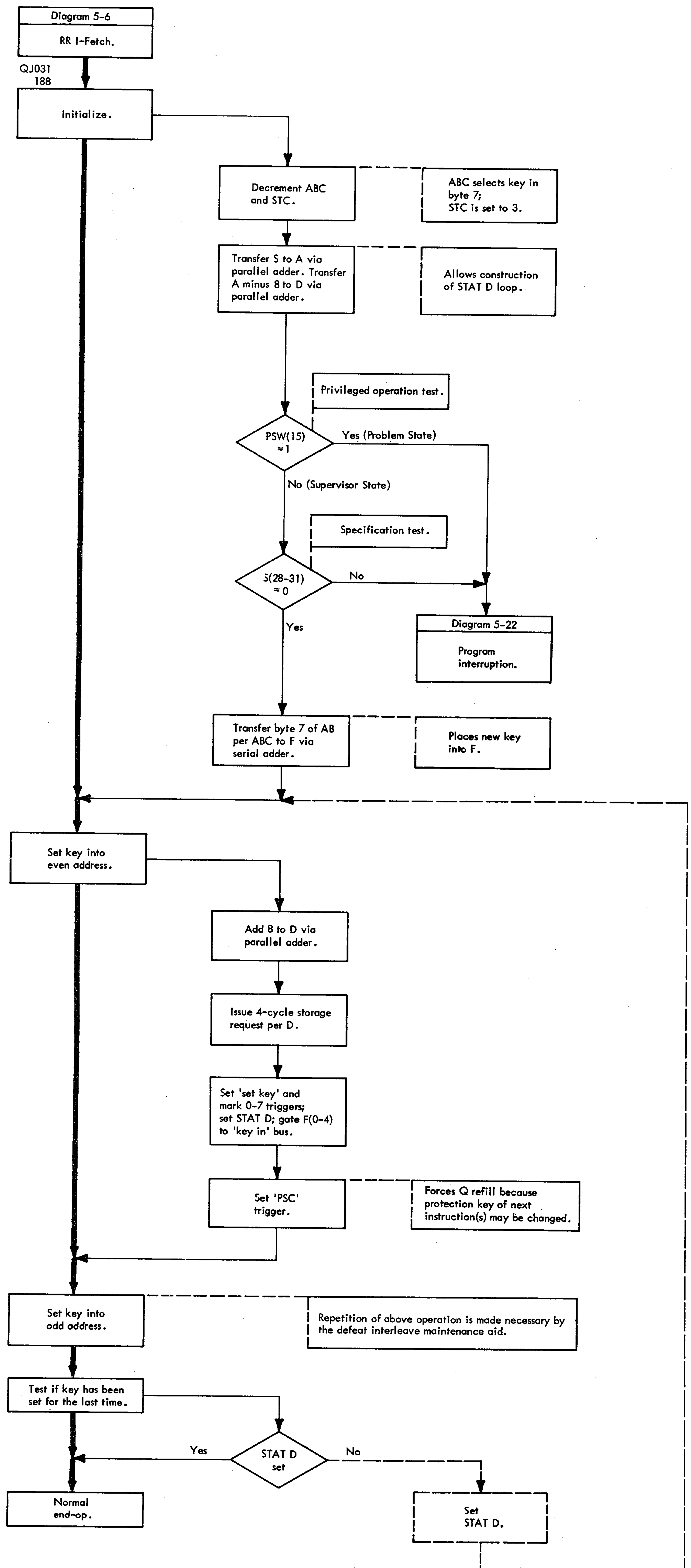


Diagram 5-605. Set Storage Key, SSK (08)

- RR Format:

08	R1	R2
0–7	8–11	12–15

- Set storage key (bits 24-28 of 1st GPR, per R1) for 2048-byte storage block (addressed by bits 8-20 of 2nd operand, in GPR per R2) into storage protection logic in main storage.
- Conditions at start of execution:
  1. Instruction is in E.
  2. 1st operand is in A, B, and D.
  3. 2nd operand is in S and T.
  4. STC = 4 and ABC = 0.
- Format of key in 1st operand:

Match Bits (24–27) | F (28) | 29–31 — F = 'fetch protect' bit

- Bits 8-20 of 2nd operand designate which block of 2048 bytes in main storage is to have its key set.
- 'Set key' trigger: ALD MC181; STAT D: ALD KS051.

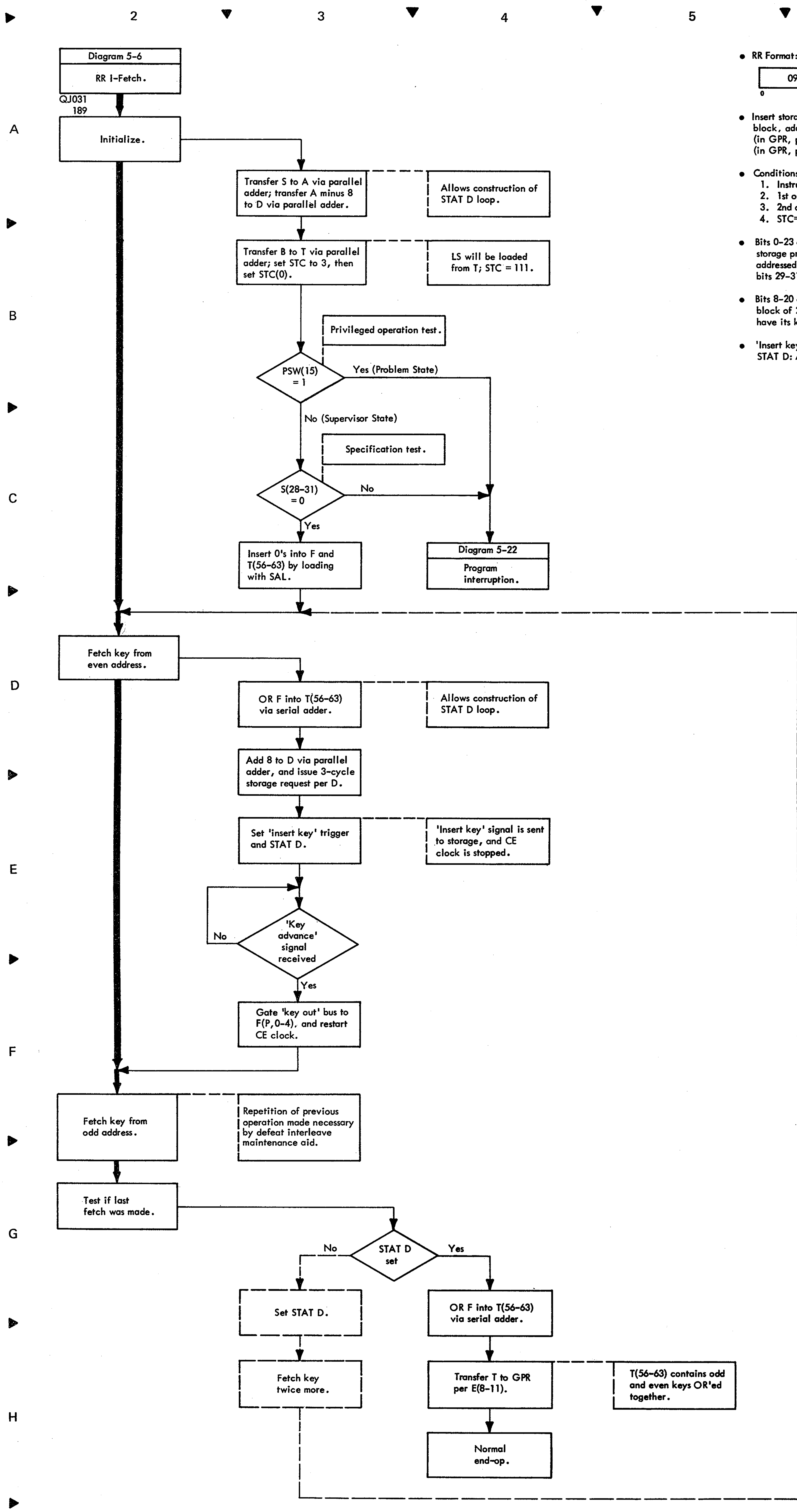


Diagram 5-606. Insert Storage Key, ISK (09)

- RR Format:

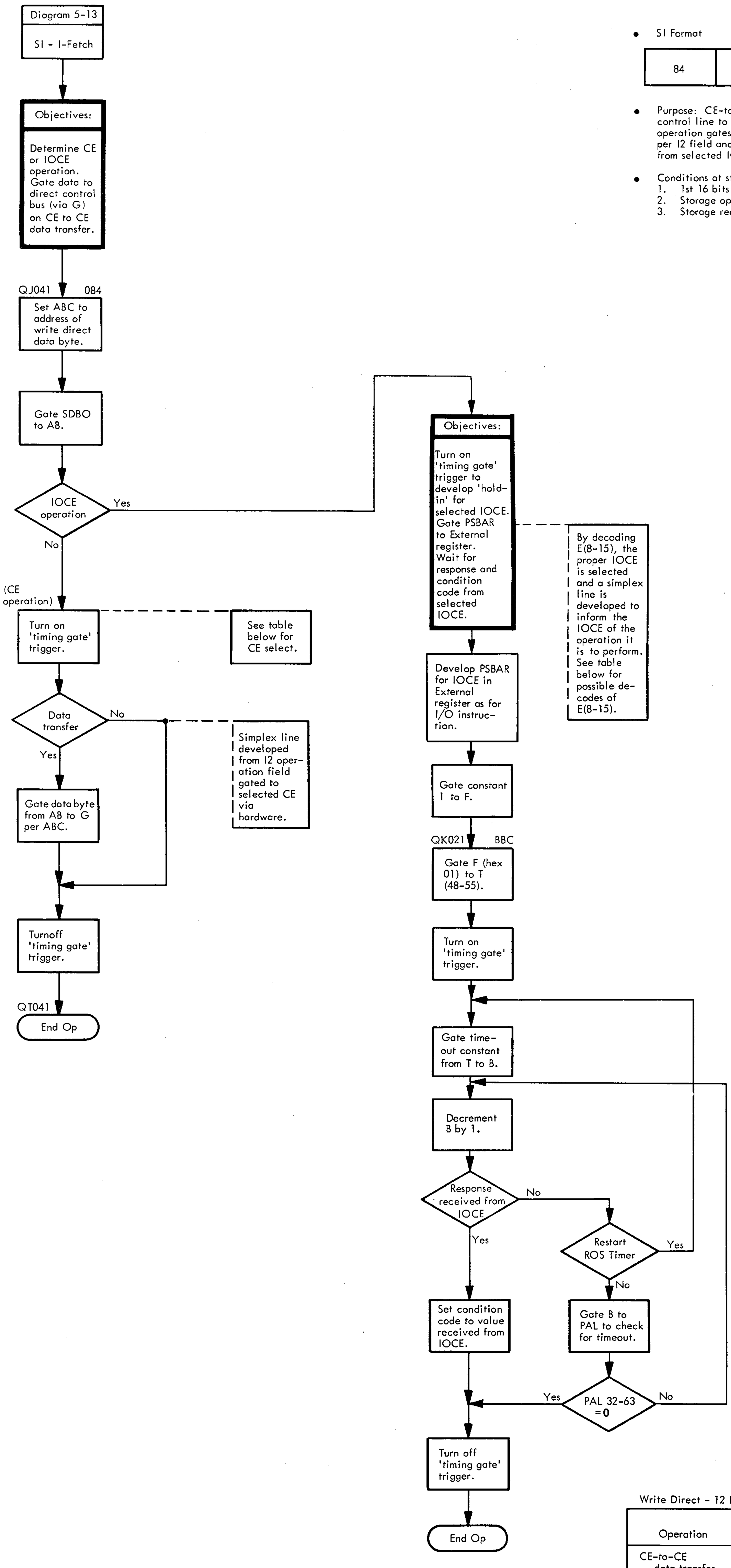
09	R1	R2
0 … 7	8 … 11	12 … 15

- Insert storage protection key for 2048-byte storage block, addressed by bits 8-20 of 2nd operand (in GPR, per R2), into bits 24-28 of 1st operand (in GPR, per R1).
- Conditions at start of execution:
  1. Instruction is in E.
  2. 1st operand is in A, B, and D.
  3. 2nd operand is in S and T.
  4. STC= 4 and ABC = 0.
- Bits 0-23 of 1st operand are not changed; storage protection key from block of addressed storage is inserted into bits 24-28; bits 29-31 are set to 0's.
- Bits 8-20 of 2nd operand designate which block of 2048 bytes in main storage is to have its key inserted into 1st operand.
- 'Insert key' trigger: ALD MC 181; STAT D: ALD KS051.

- SI Format

84	I2	B1	D1

- Purpose: CE-to-CE operation gates data byte or simplex control line to CE selected per I2 field. CE-to-IOCE operation gates simplex control line to IOCE selected per I2 field and sets condition code to value received from selected IOCE.
- Conditions at start of execution:
  1. 1st 16 bits of instruction are in E.
  2. Storage operand address in D.
  3. Storage request has been issued per D.



Write Direct - I2 Field - E(8-15) Decode

Operation	Operation				Element Selected			
	8	9	10	11	12	13	14	15
CE-to-CE data transfer	0	0	0	0	CE 1	CE 2	CE 3	CE 4
CE external start	0	0	0	1	CE 1	CE 2	CE 3	CE 4
CE logout	0	0	1	0	CE 1	CE 2	CE 3	CE 4
IOCE logout	0	0	1	1	IOCE 1	IOCE 2	IOCE 3	0
CE external stop	0	1	0	0	CE 1	CE 2	CE 3	CE 4
IOCE processor start	0	1	0	1	IOCE 1	IOCE 2	IOCE 3	0
IOCE processor stop	0	1	1	0	IOCE 1	IOCE 2	IOCE 3	0
IOCE processor interrupt	0	1	1	1	IOCE 1	IOCE 2	IOCE 3	0

Note:
I2 field decoding is done by hardware, and the select to the proper element is also hardware-developed.

Diagram 5-607. Write Direct, WRD (84)

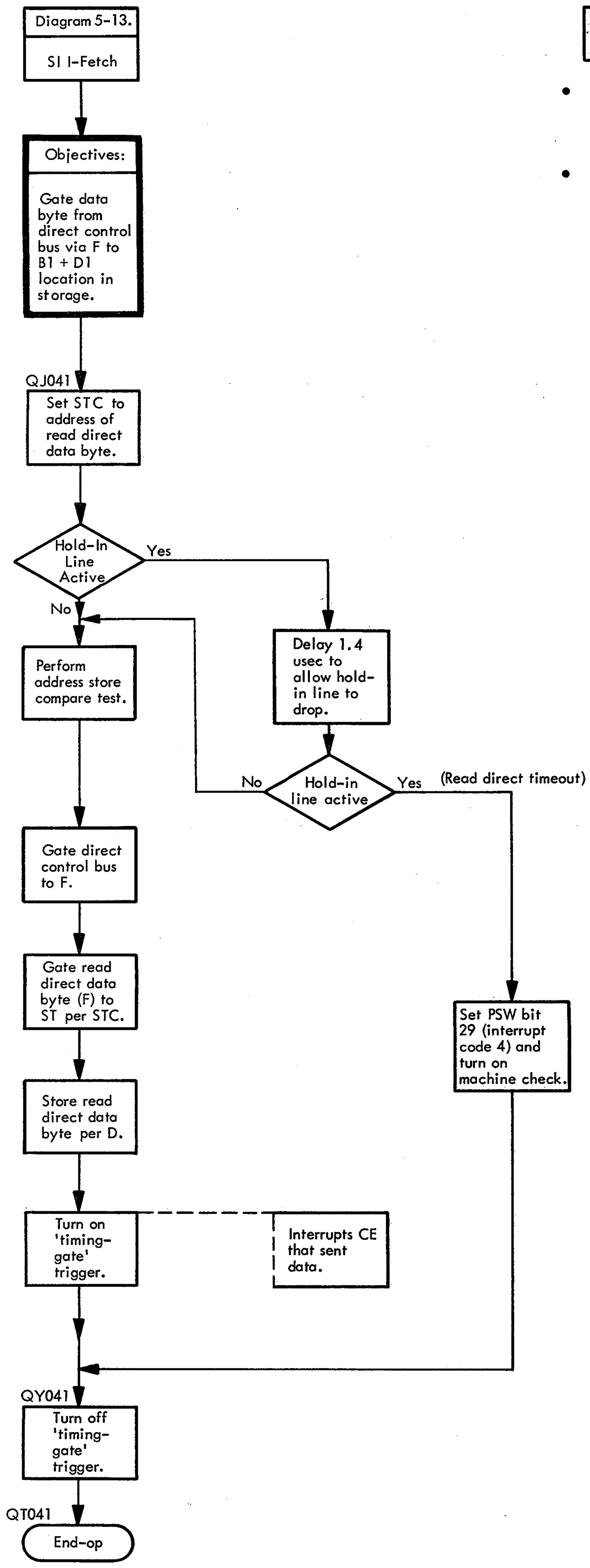


Diagram 5-608. Read Direct, RDD (85)

- SI Format

85	I2	B1	D1

- Purpose: Gates a data byte from the control bus to the byte location (specified by the B1, D1 fields) in storage.

- Conditions at start of execution:
  1. 1st 16 bits of instruction are in E.
  2. Storage operand address is in D.
  3. Storage request has been issued per D.

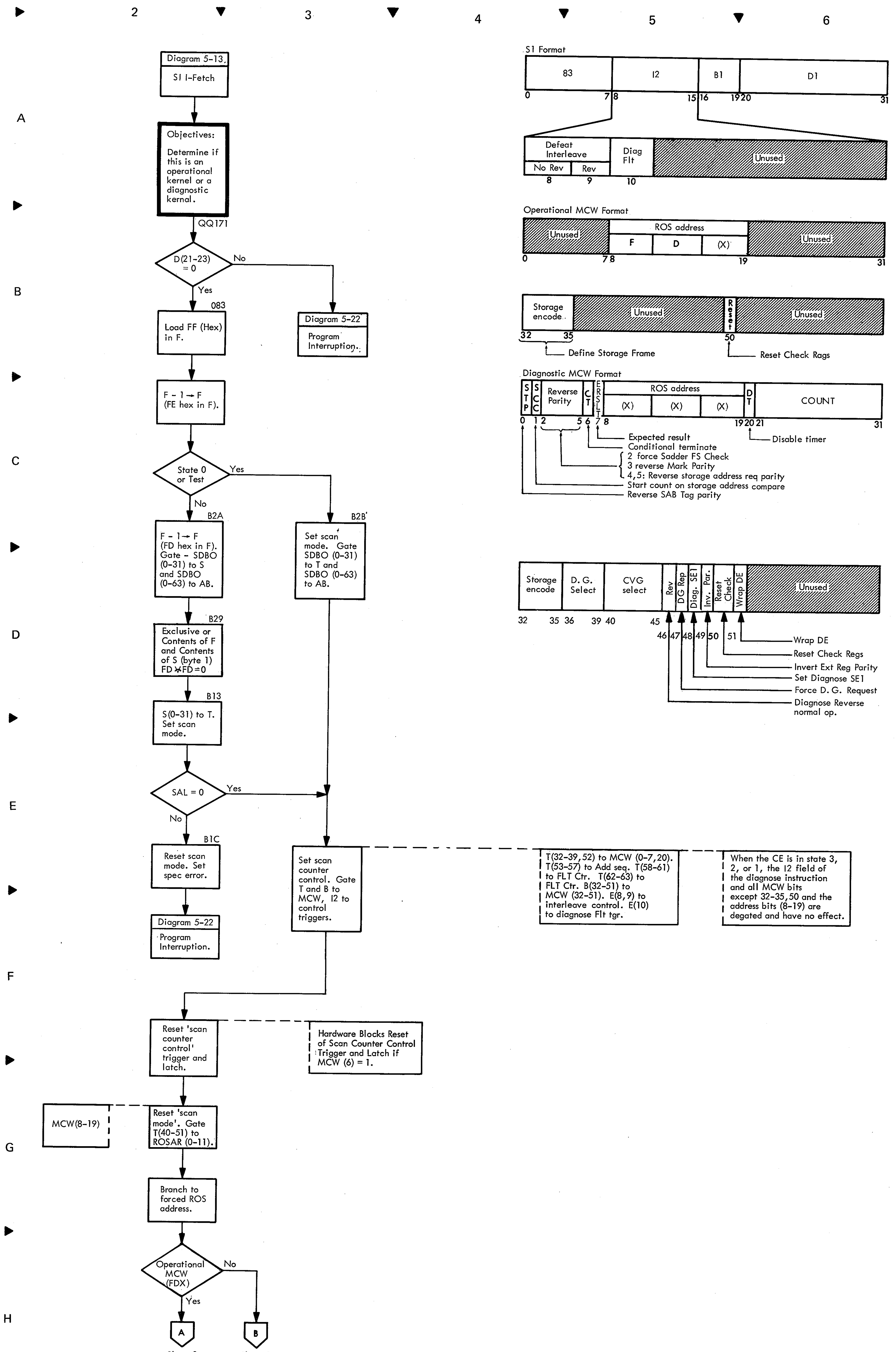


Diagram 5-609. Diagnose (83) (Sheet 1 of 3)

Table 1

E Req 11-15	STAT H	STAT D	STAT G	Register Logged Out
16-23	Off	Off	Off	FLT Point
00-15	Off	Off	Off	Gen Purpose
01	On	Off	Off	CCR
01	On	Off	Off	DAR Mask
01	On	Off	On	Ext Req
01	On	On	Off	Check Req
01	On	On	On	ATR 1
00	On	Off	Off	ATR 2
00	On	Off	On	DAR

E3 = 0 and STAT H off mean logout complete.

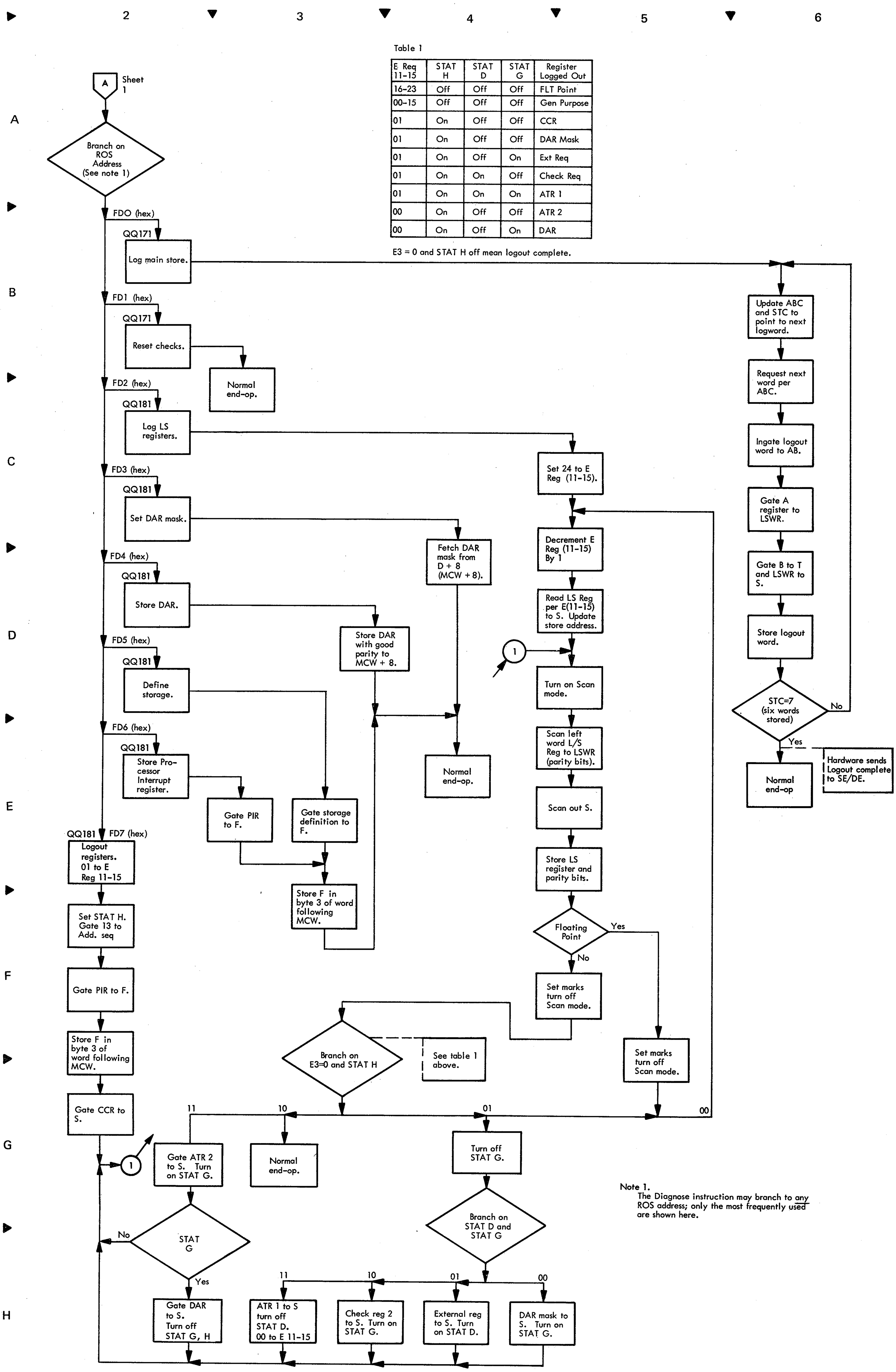


Diagram 5-609. Diagnose (83) (Sheet 2 of 3)

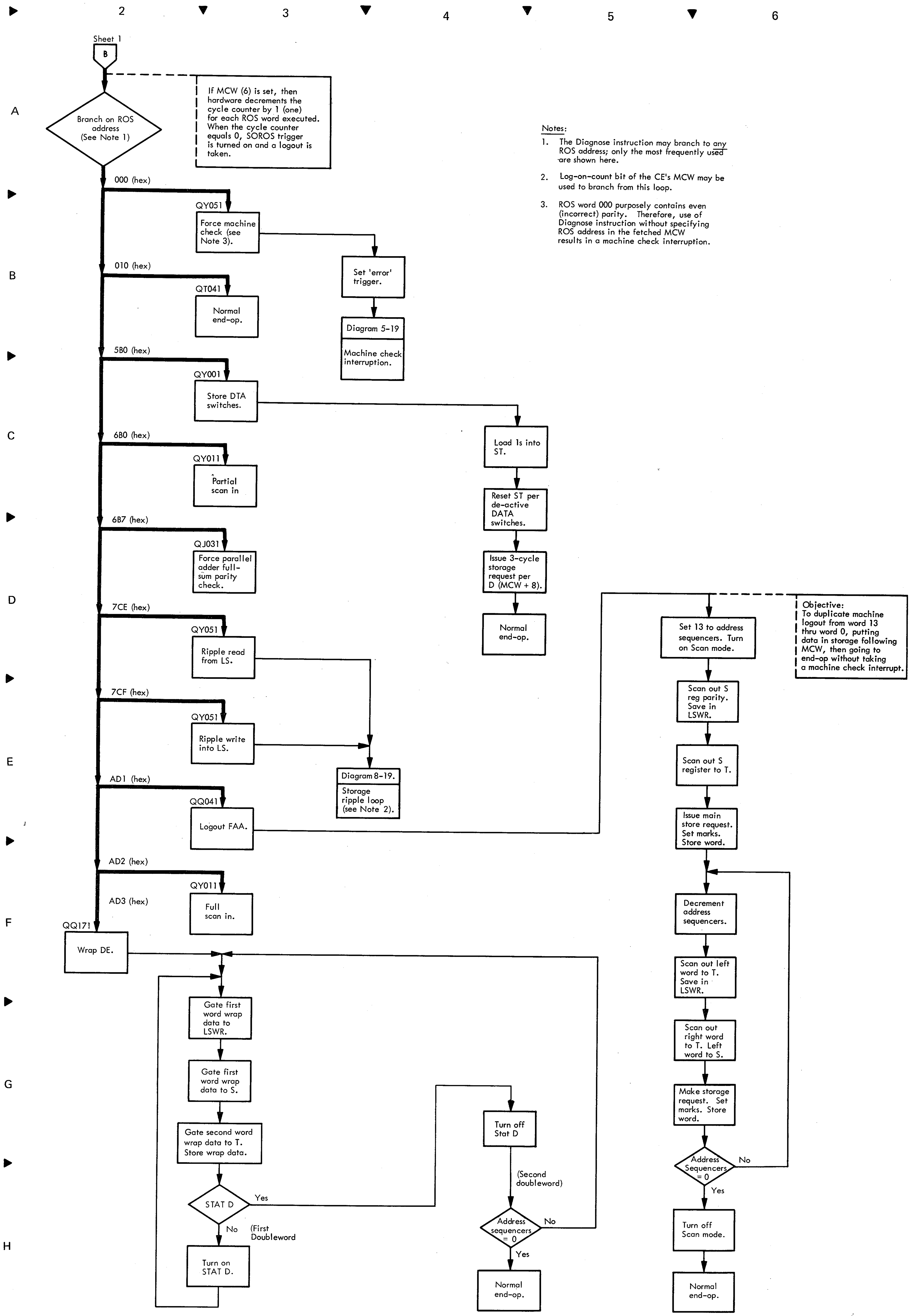
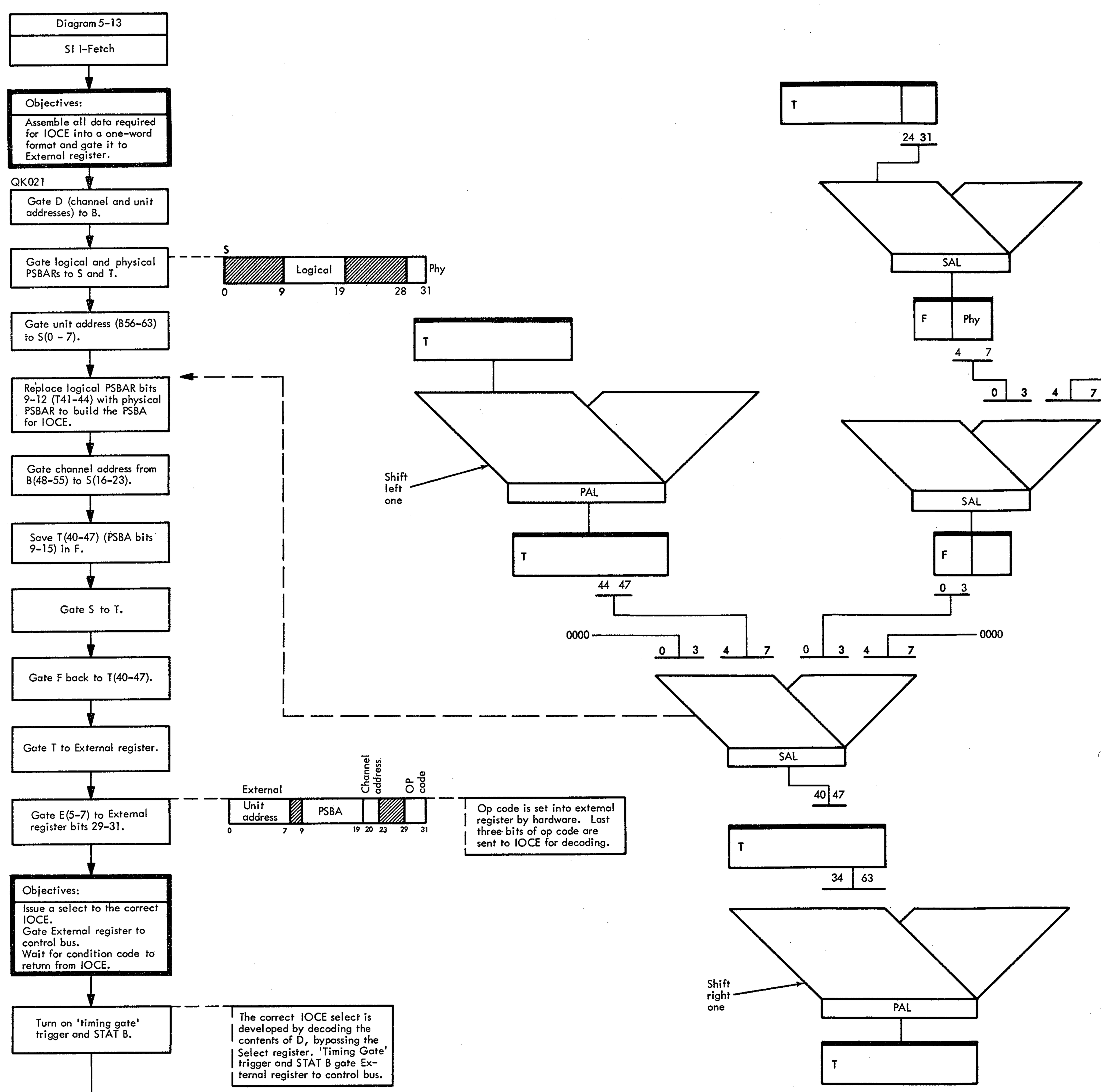
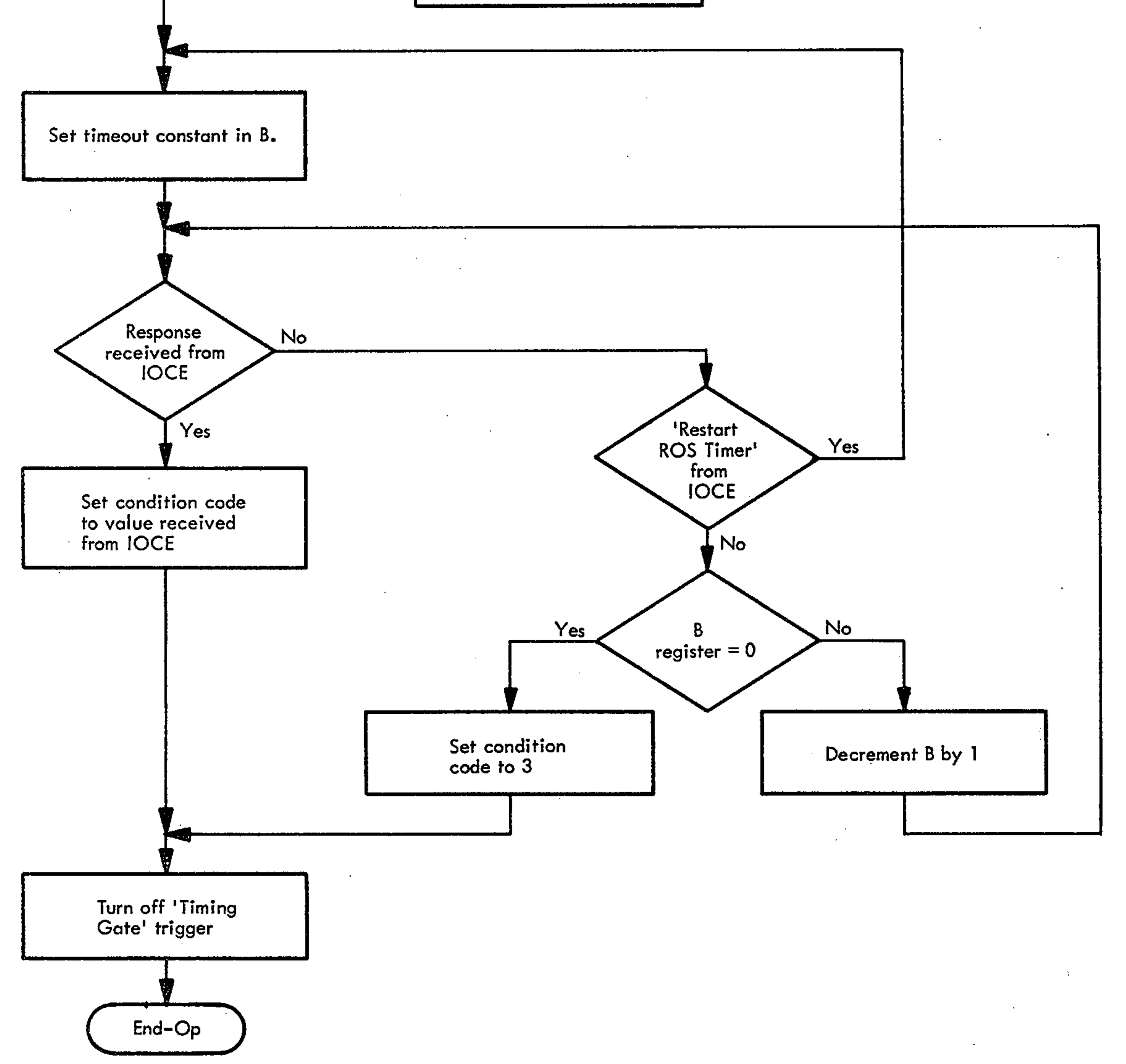


Diagram 5-609. Diagnose (83) (Sheet 3 of 3)





- SI Format

Op Code		B1	D1

- Purpose: Assembles data required for IOCE to execute the I/O instructions into a one-word format and gates it to the proper IOCE via the control bus.

- Instructions

Op Code	Mnemonic	Condition Code: Zero	One	Two	Three
9C	SIO	Operation initiated	CSW stored	Channel busy	*
9D	TIO	Available	CSW stored	Channel busy	*
9E	HIO	Interrupt pending	CSW Stored	Burst operation terminated	*
9F	TCH	Available	Interrupt pending	Channel in burst mode	*
9B	SPCI	Not working	CSW stored	PCI flag set	Invalid I/O address format

* - Not operational or invalid I/O address format

- Conditions at start of execution:
  1. 1st 16 bits of instruction are in E.
  2. Operand address (address of channels and I/O unit) is in D.

Diagram 5-701. I/O Instructions

- RR Format: L1

0C	R1	R2
0 – 7	8 – 11	12 – 15

- Purpose: Loads identity of CE executing instruction into bits 28-31 of GPR specified by R1 field.

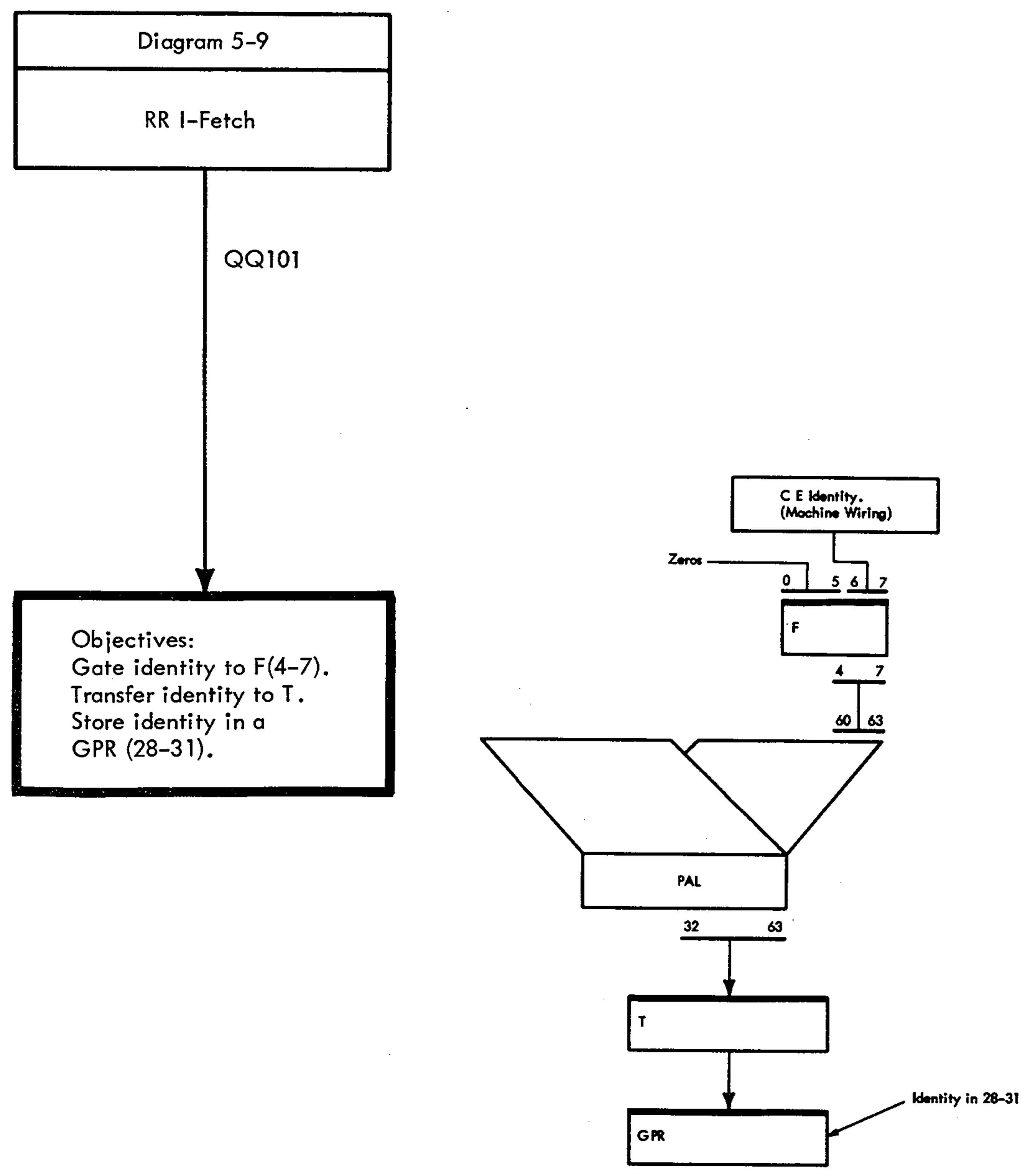


Diagram 5-801. Load Identity, LI (0C)

- RR Format

0E	R1	R2
0 … 7	8 … 11	12 … 15

- Purpose: Places contents of ATR1 and ATR2 into two GPR's specified by R1 and R2 fields.

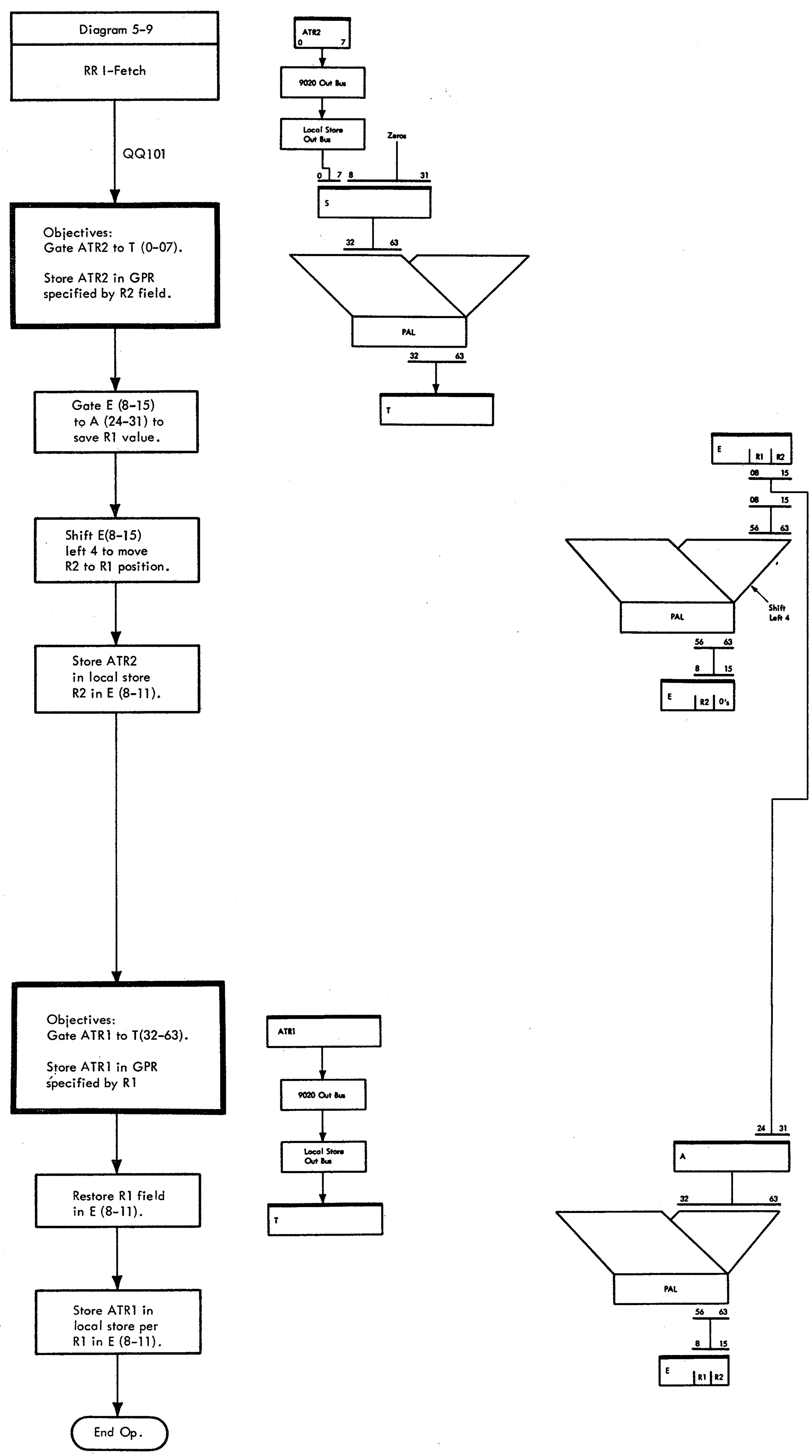


Diagram 5-802. Insert ATR, IATR (0E)

- RR Format

0B	N
0–7	8–15

- Purpose: Provides a variable delay (256 usec x N) dependent on the value of N.
- Conditions at the beginning of execution: Instruction and N are in E.

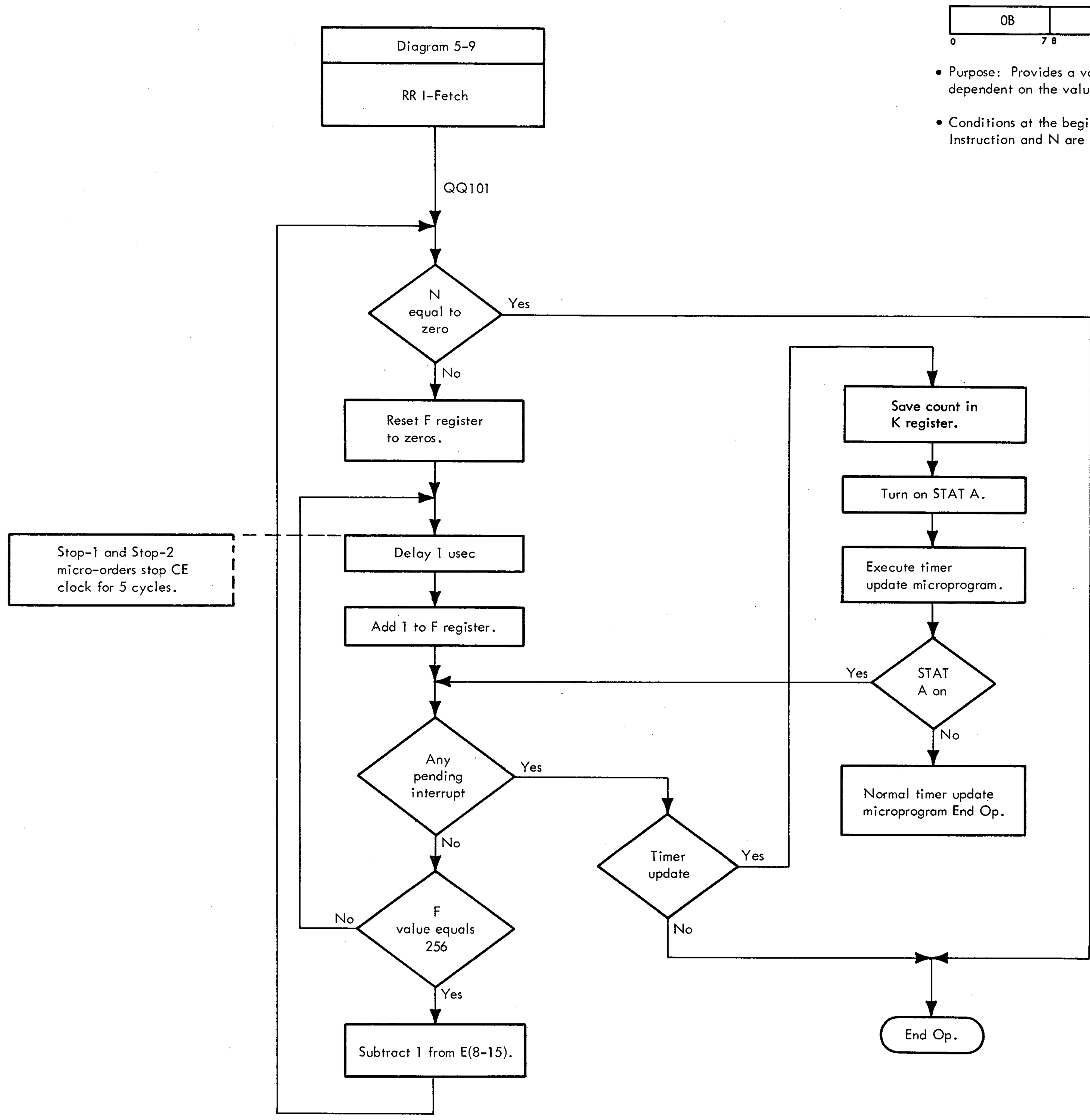


Diagram 5-803. Delay, DLY (0B)

- SI Format

A0	(unused, bits 8–15)	B1	D1
0–7	8–15	16–19	20–31

- Purpose: Store contents of logical and physical PSBAR in bit positions 9 - 19 and 28 - 31, respectively, at the word location in main storage (specified by 1st operand address).
- Conditions at the beginning of execution:
  1. 1st 16 bits of instruction are in E.
  2. Operand address (address to store PSBA) is in D.

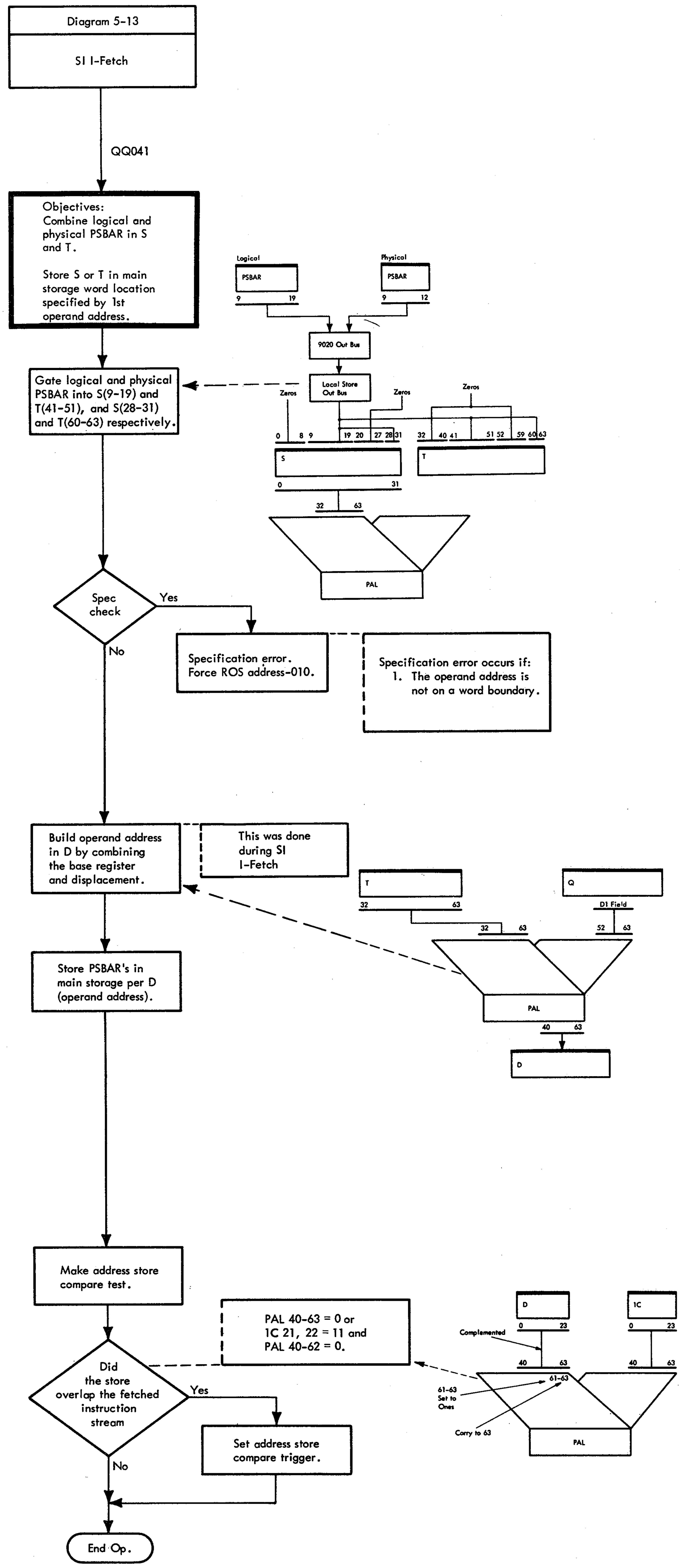


Diagram 5-804. Store PSBAR, SPSB (A0)

- SI Format

A1	(hatched)	B1	D1
0–7	8–15	16–19	20–31

- Purpose: Load logical PSBAR from a storage location pointed to by the operand. Load physical PSBAR from an ATR slot (selected by bits 9-12 of the newly fetched logical PSBA).
- Conditions at beginning of execution:
  1. 1st 16 bits of instruction are in E.
  2. Operand address (address of new PSBA) is in D.

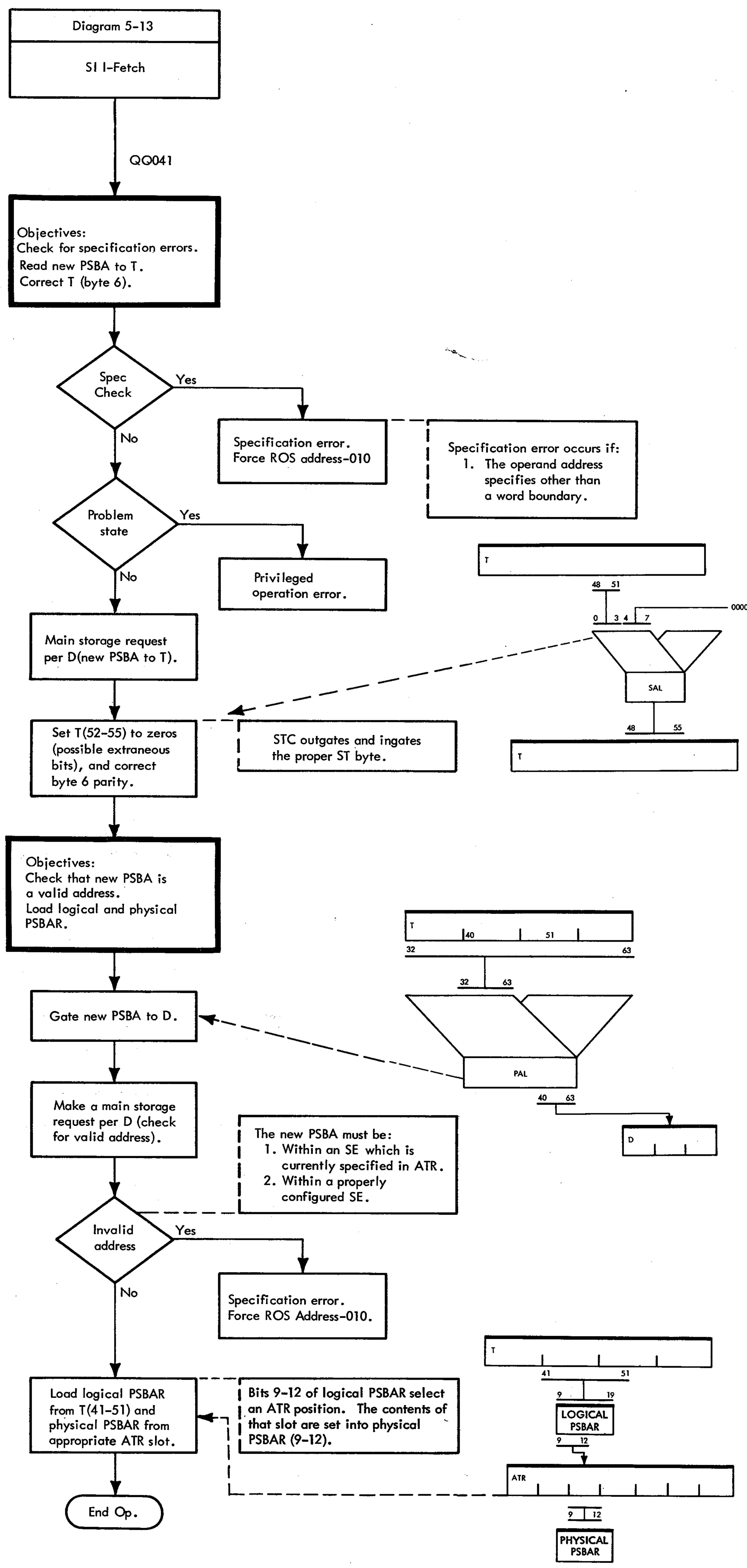


Diagram 5-805. Load PSBAR, LPSB (A1)

- SS Format

D8	L	B1	D1	B2	D2
0–7	8–15	16–19	20–31	32–35	36–47

- Purpose: Moves up to 256 words from one storage location (B2, D2) to another storage location (B1, D1). The number of words to be moved is specified by the L-field.
- Conditions at the beginning of execution:
  1. Source address is in IC and T.
  2. Destination address is in D.
  3. Updated IC is in LSWR.

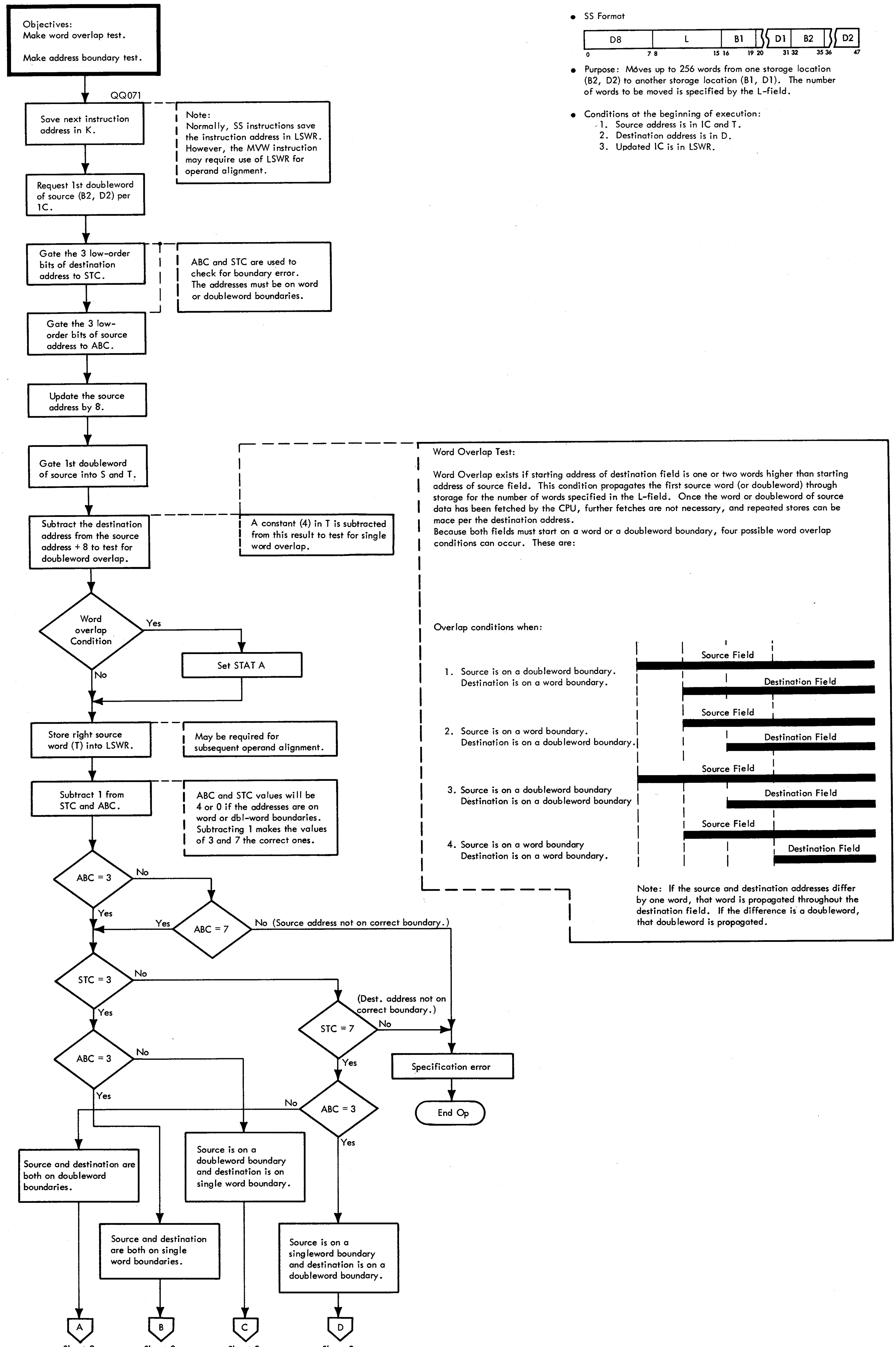


Diagram 5-806. Move Word, MVW (D8) (Sheet 1 of 3)

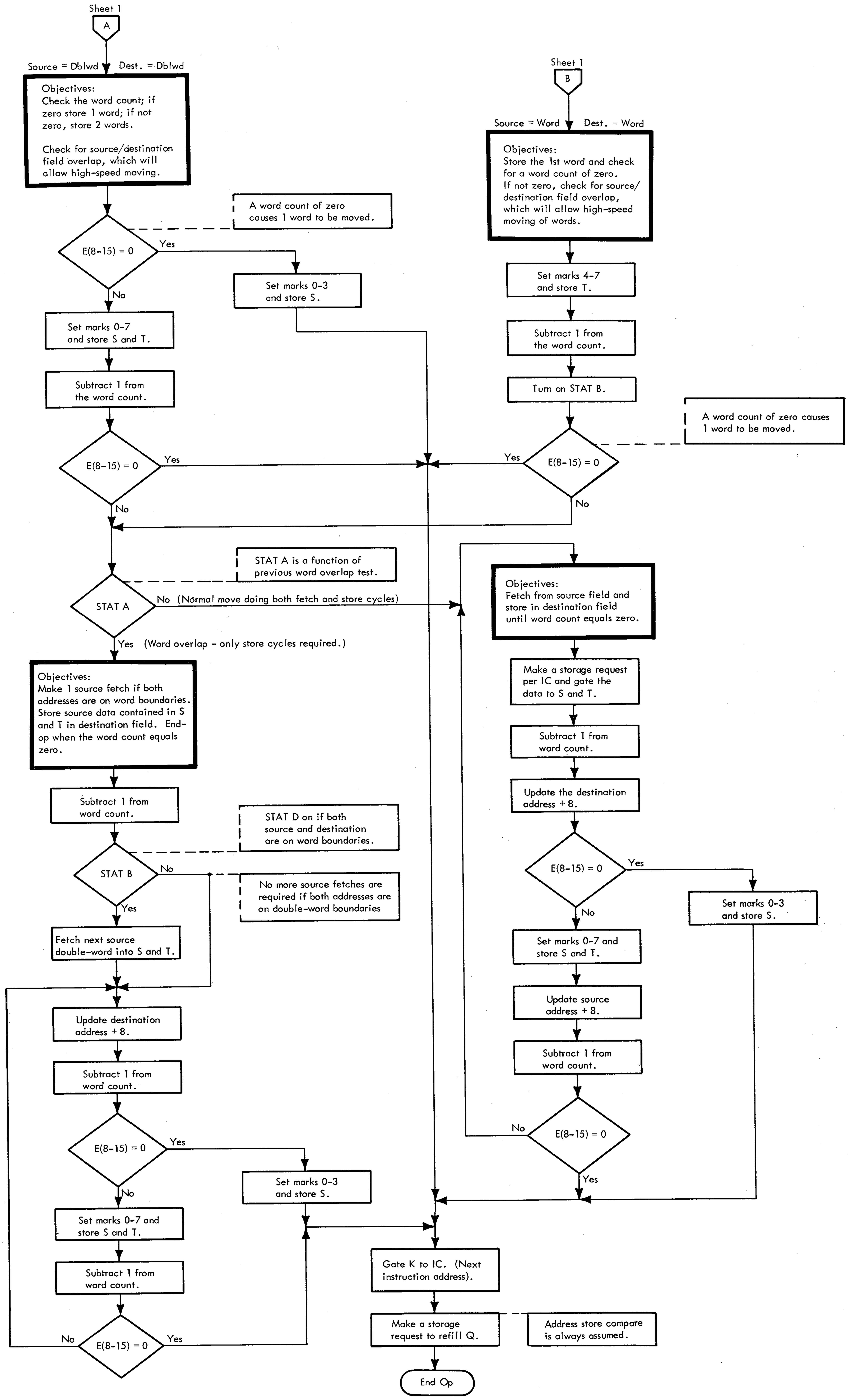


Diagram 5-806. Move Word, MVW (D8) (Sheet 2 of 3)

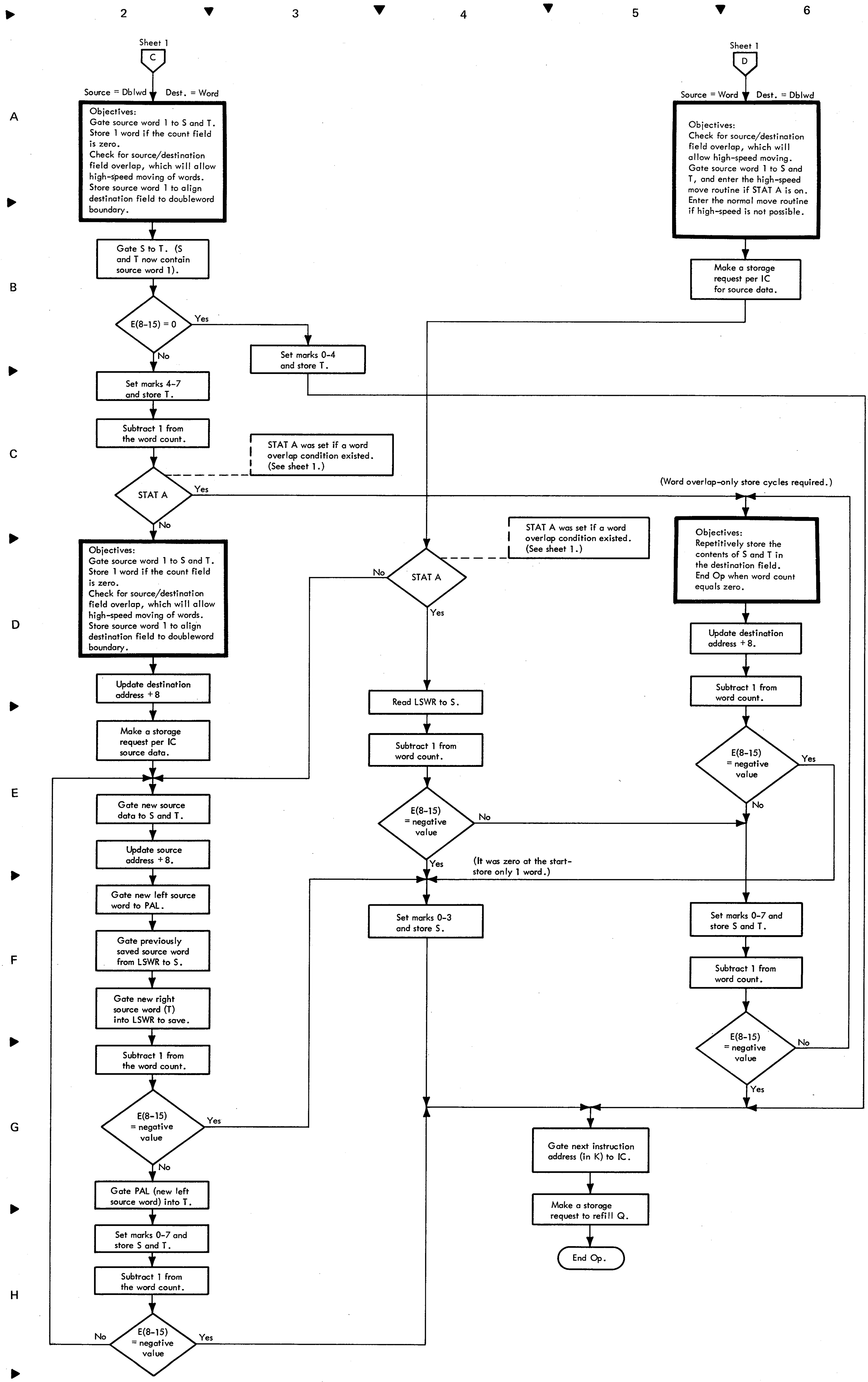


Diagram 5-806. Move Word, MVW (D8) (Sheet 3 of 3)

- SI Format:

9A	Key / I2 Sel bits	B1	D1
0–7	8–15	16–19	20–31

- Purpose: Transfers storage key and PSW address to selected IOCE. The IOCE fetches the new PSW and proceeds with the operation designated by the PSW.

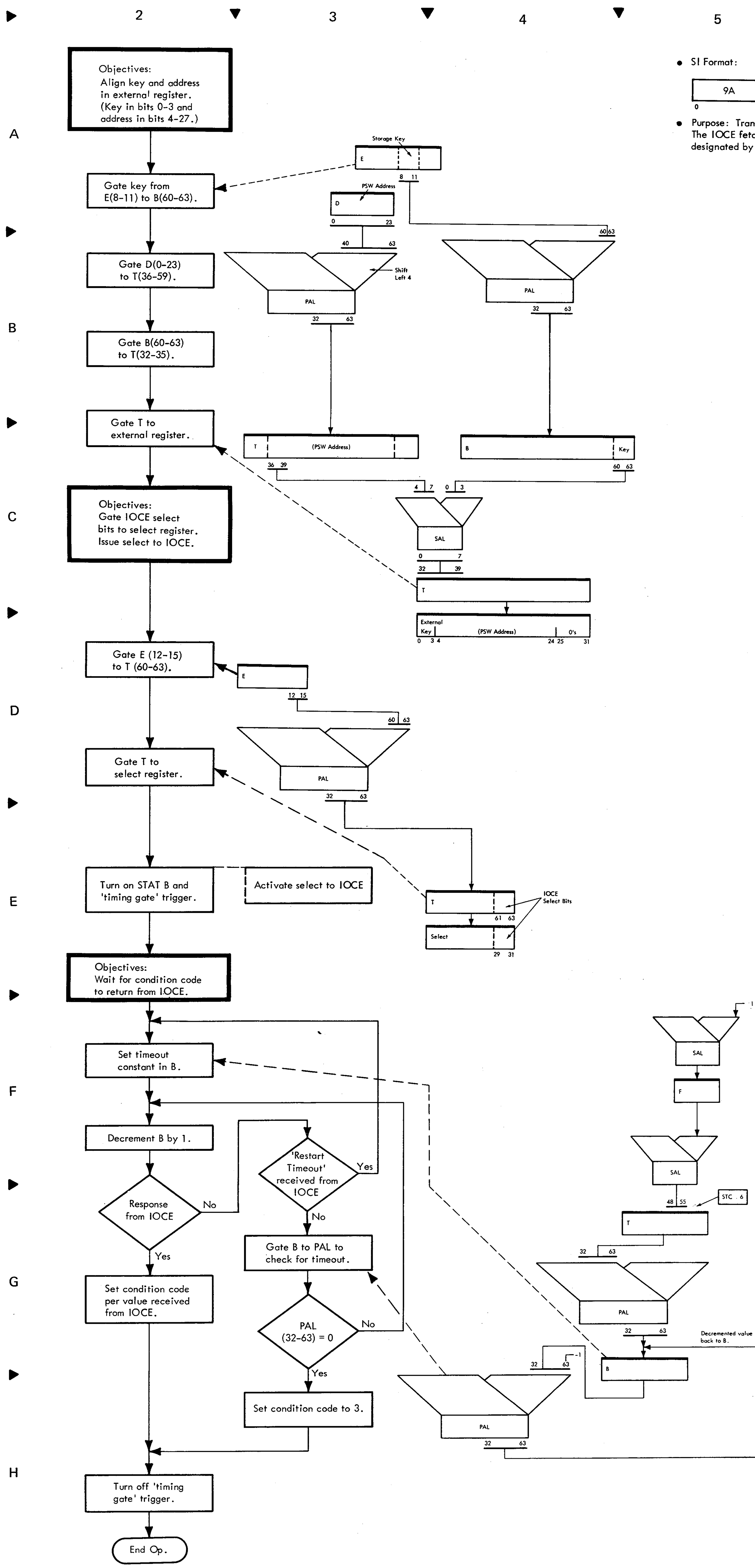
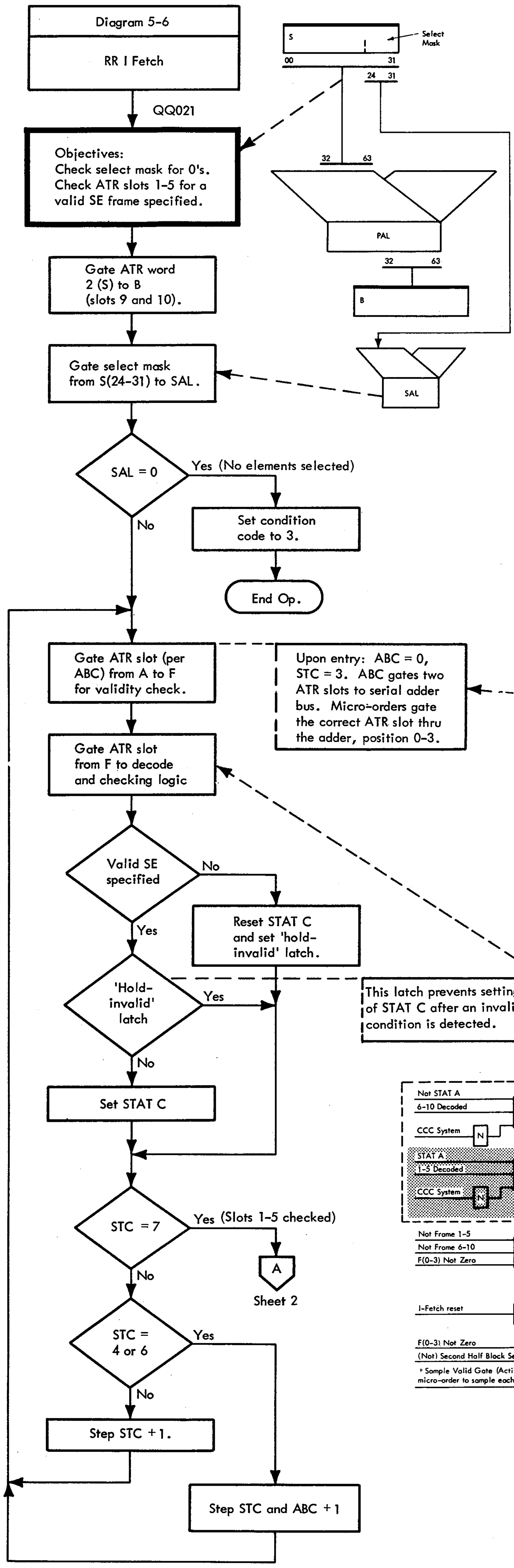
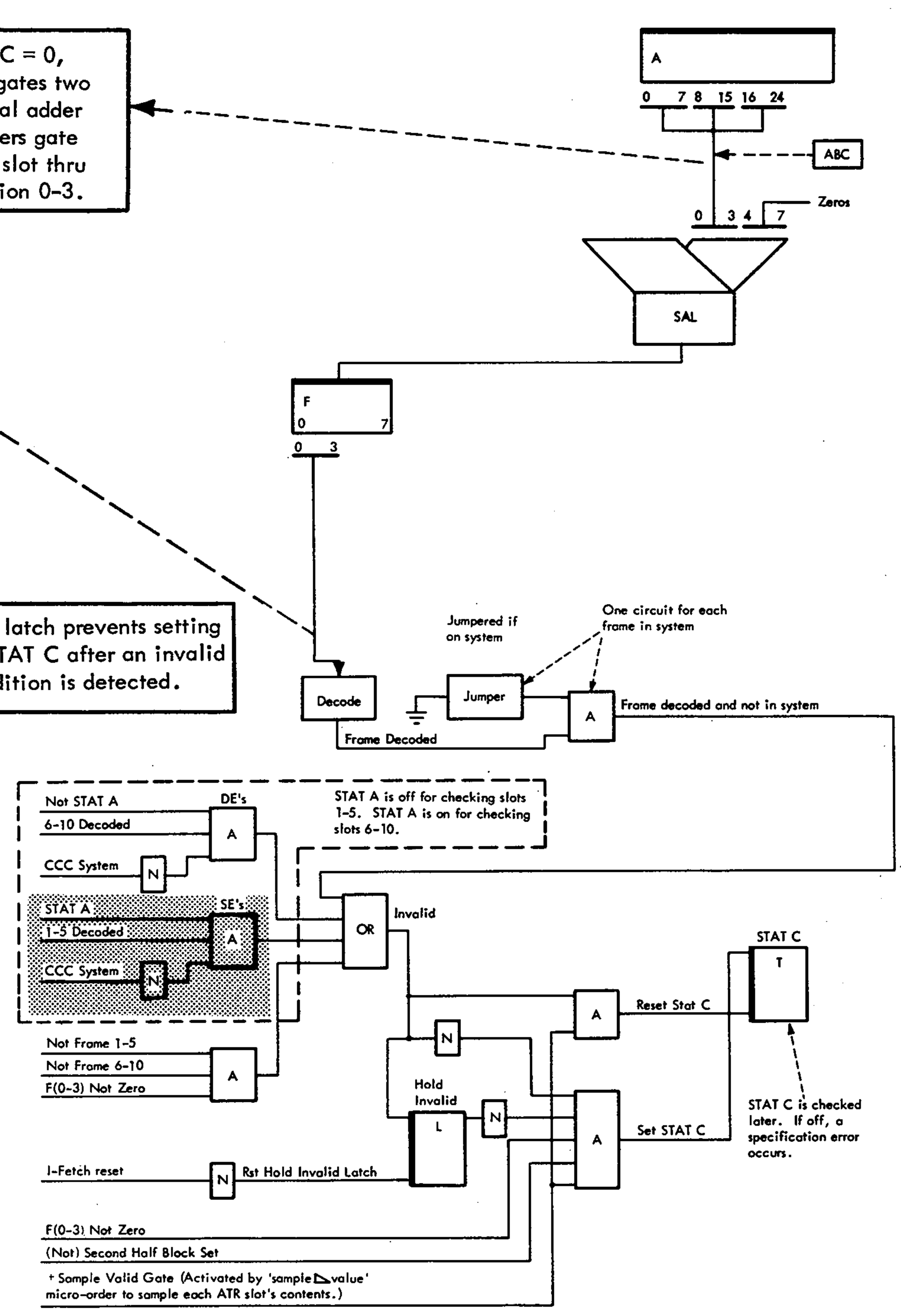


Diagram 5-807. Start I/O Processor, SIOP (9A)





- RR Format

0	D	R1	R2

- Purpose: Loads the address translation register: (0-31) from the GPR specified by R1, and (32-39) from byte 0 of the GPR specified by R2. Byte 3 of the GPR specified by R2 is the selection mask, and the response bits are set into byte 2.

- Format of GPR specified by:
1. R1

SE	SE	SE	SE	SE	SE or DE *	SE or DE *	SE or DE *

2. R2

SE or DE *	SE or DE *	Ignored	Response Bits **	Selection Mask **

* = 9020 E only

** =

Response Bits	Selection Mask	CE or IOCE Designated
16	24	CE 1
17	25	CE 2
18	26	CE 3
19	27	CE 4
20	28	not used
21	29	IOCE 1
22	30	IOCE 2
23	31	IOCE 3

- Conditions at the start of execution:
1. Instruction is in E.
2. ATR word 1 is in A and B
3. ATR word 2 is in S and T.

Diagram 5-808. Set Address Translator, SATR (0D), Execution in Issuing CE (Sheet 1 of 6)

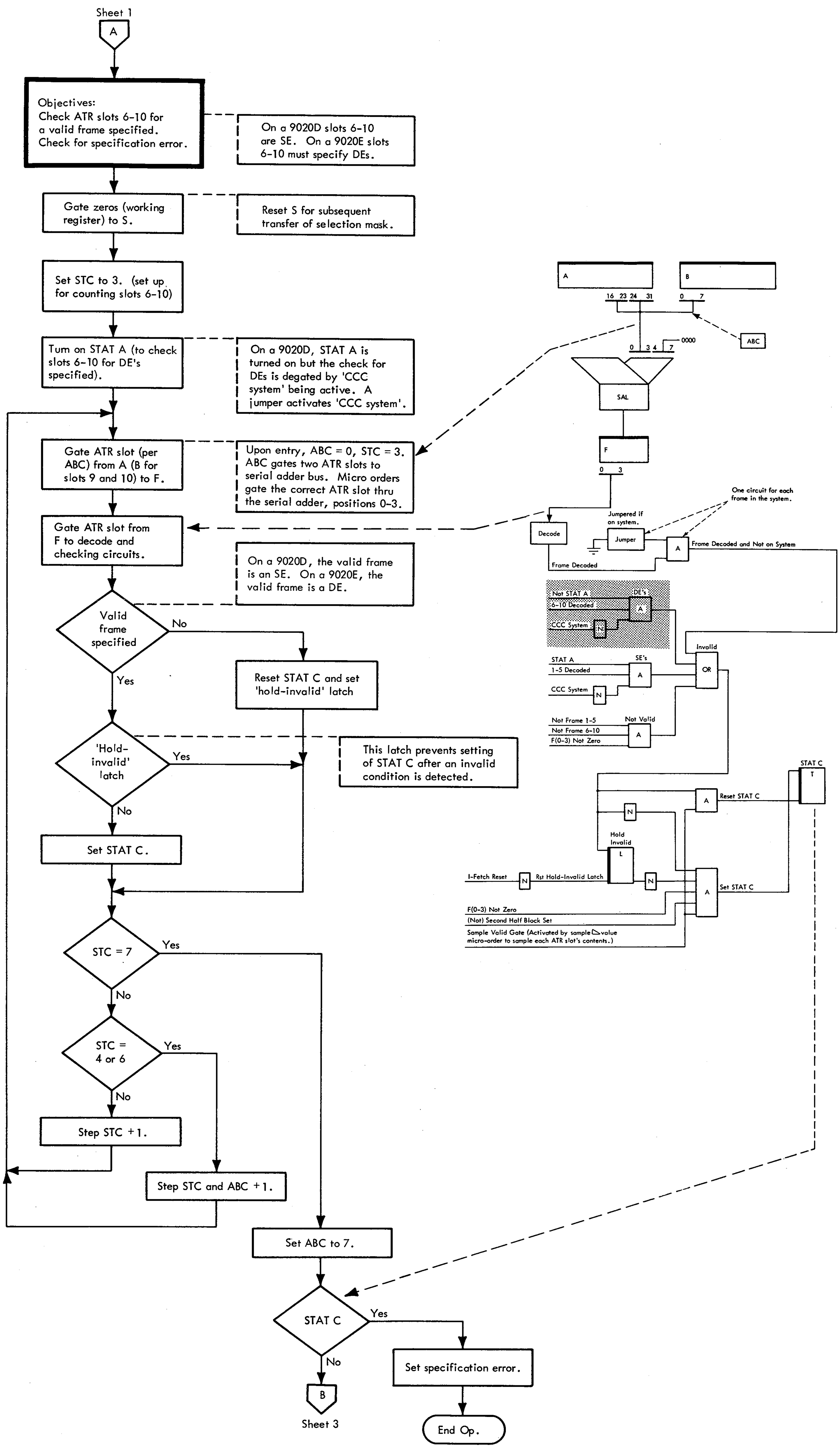


Diagram 5-808. Set Address Translator, SATR (0D), Execution in Issuing CE (Sheet 2 of 6)

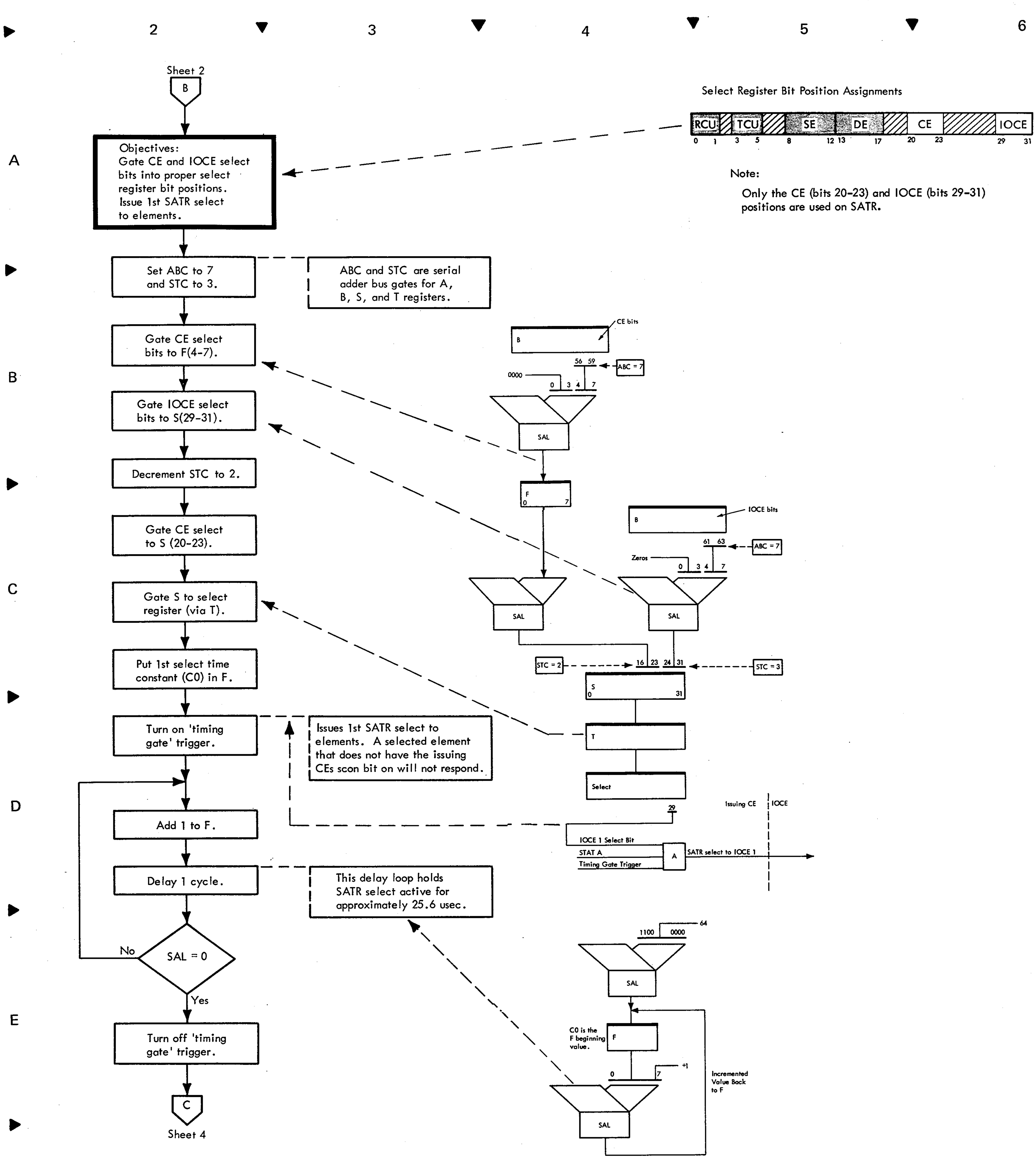


Diagram 5-808. Set Address Translator, SATR (0D), Execution in Issuing CE (Sheet 3 of 6)

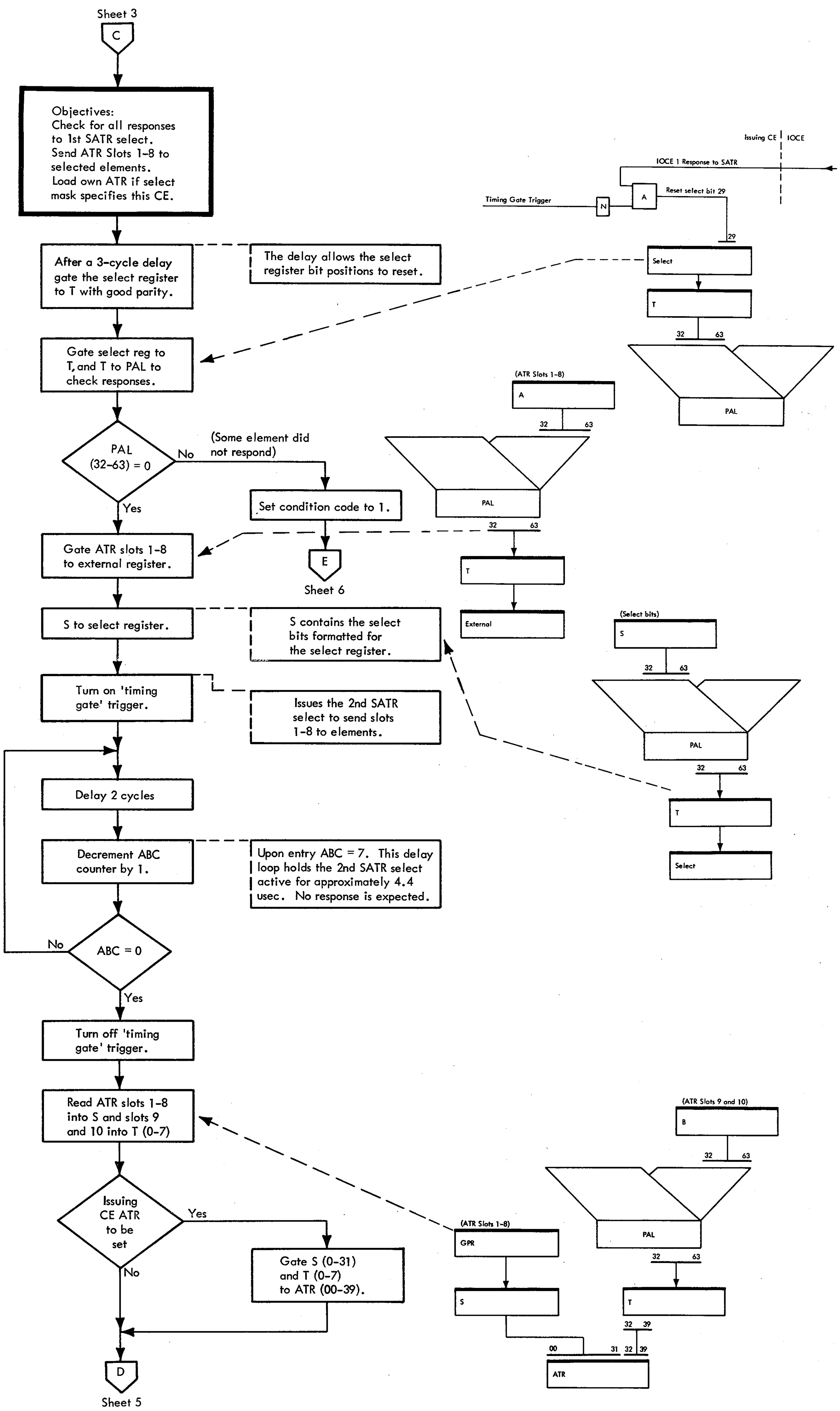


Diagram 5-808. Set Address Translator, SATR (0D), Execution in Issuing CE (Sheet 4 of 6)

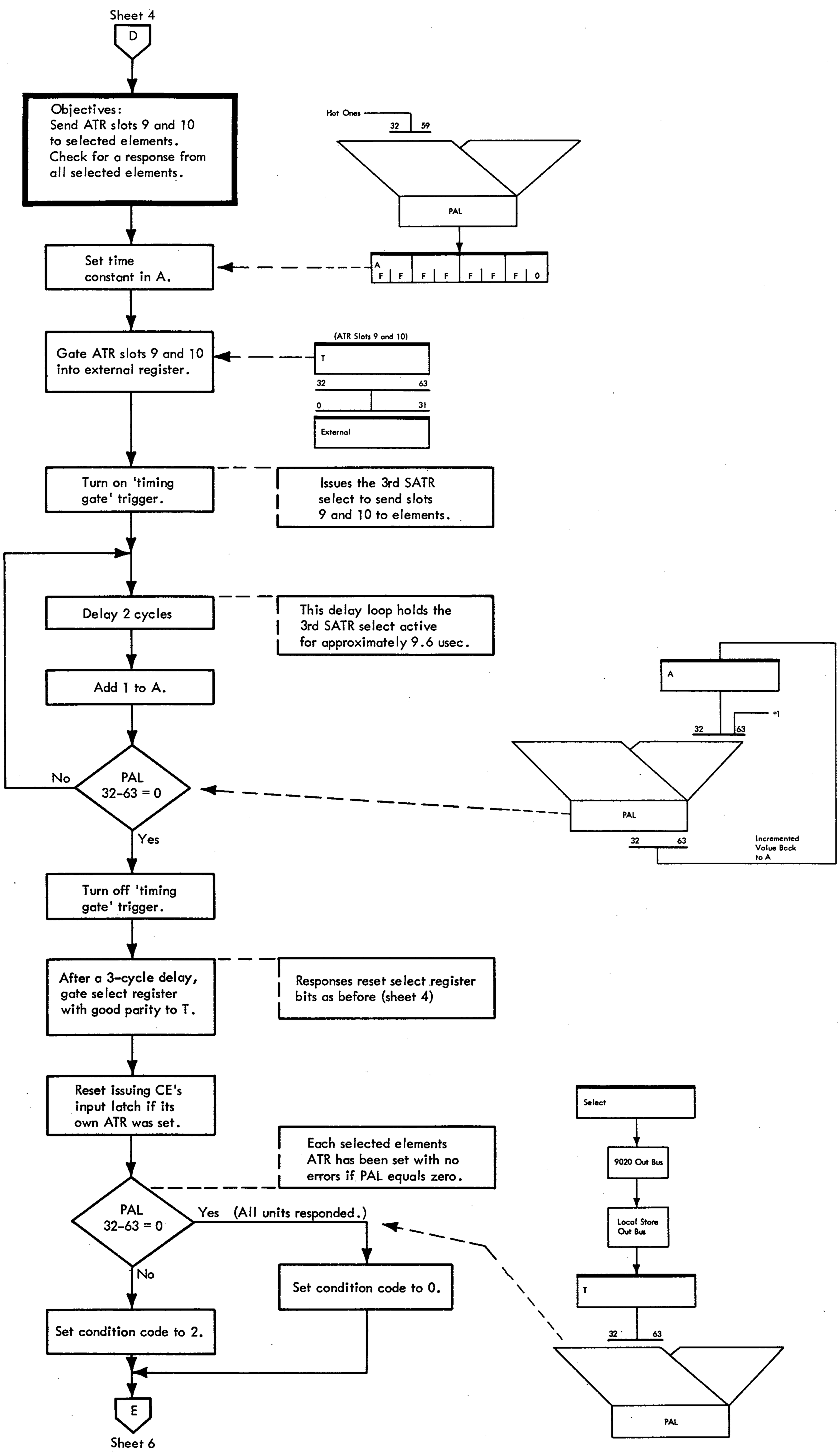
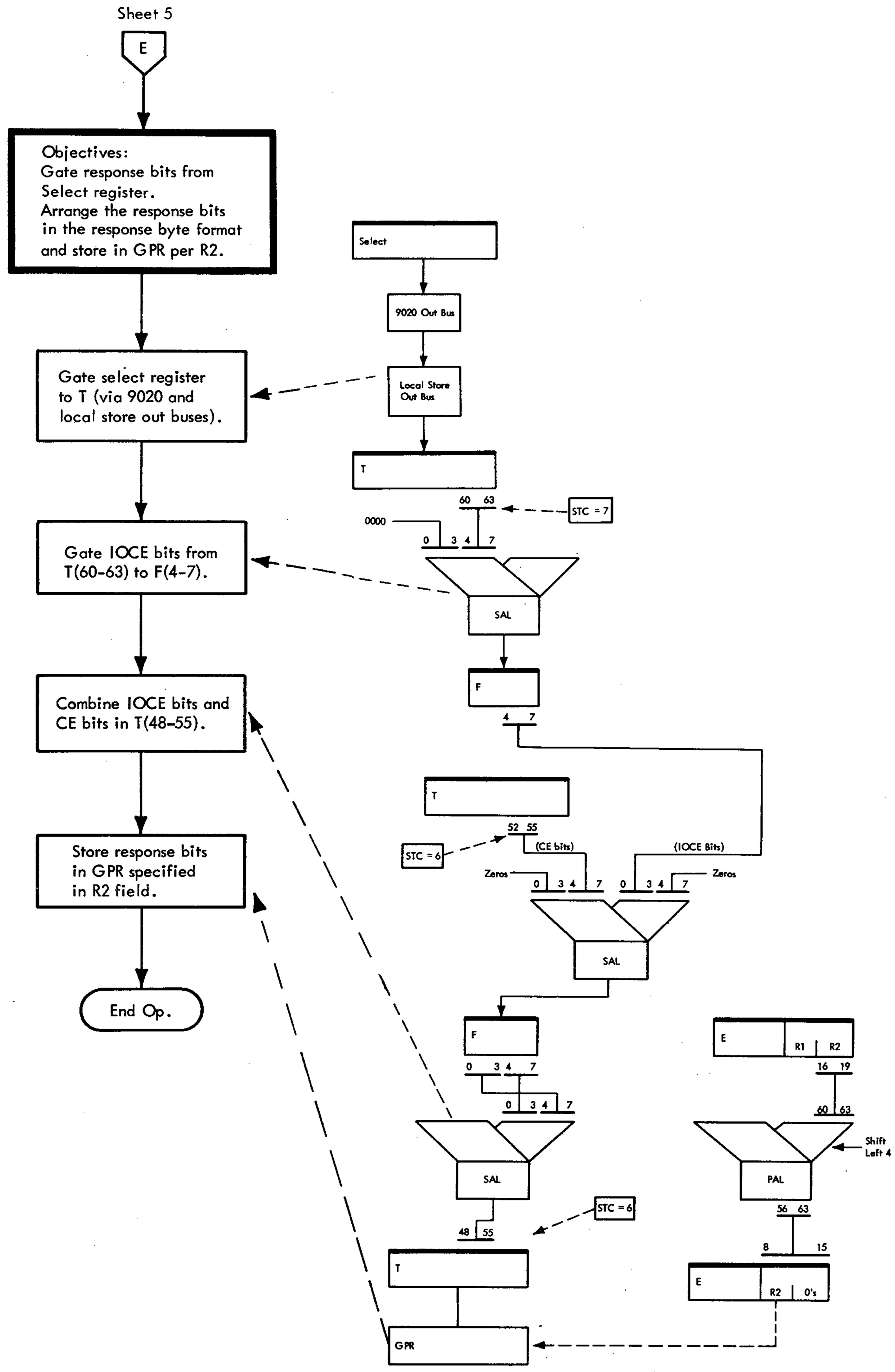


Diagram 5-808. Set Address Translator, SATR (0D), Execution in Issuing CE (Sheet 5 of 6)

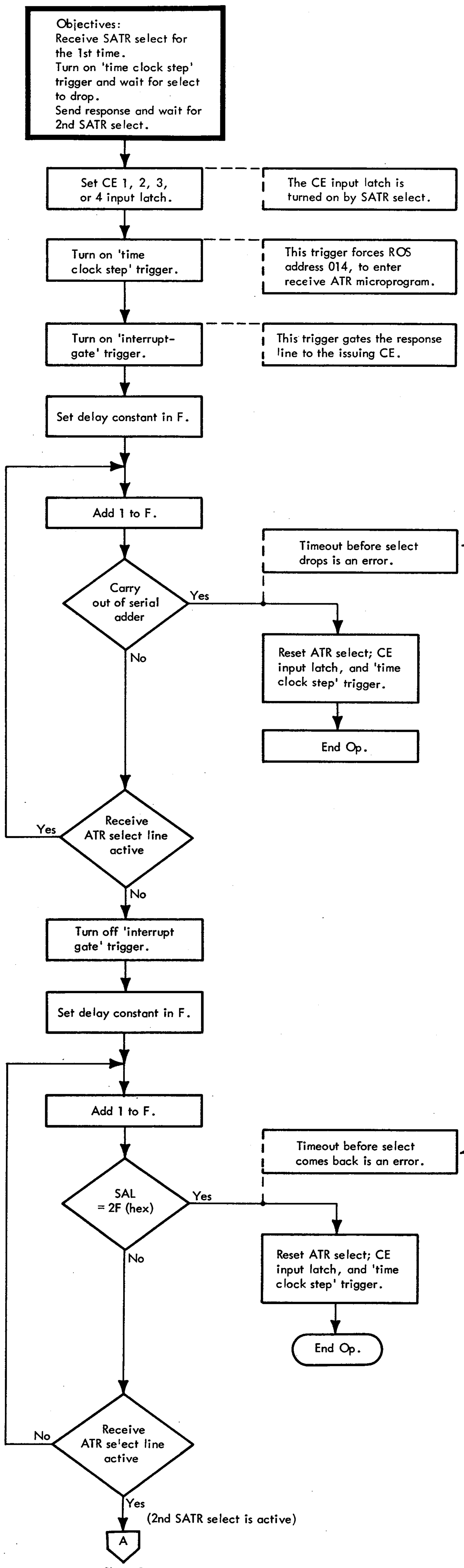


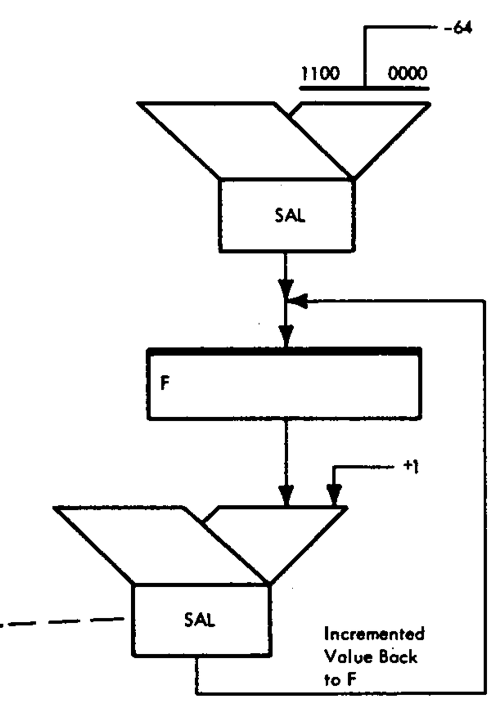
Note:
A CE receiving a SATR select signal from another CE must execute a microprogram to ingate the new ATR information. Refer to Diagram 9, MDM for explanation of the SATR instruction in the receiving CE.

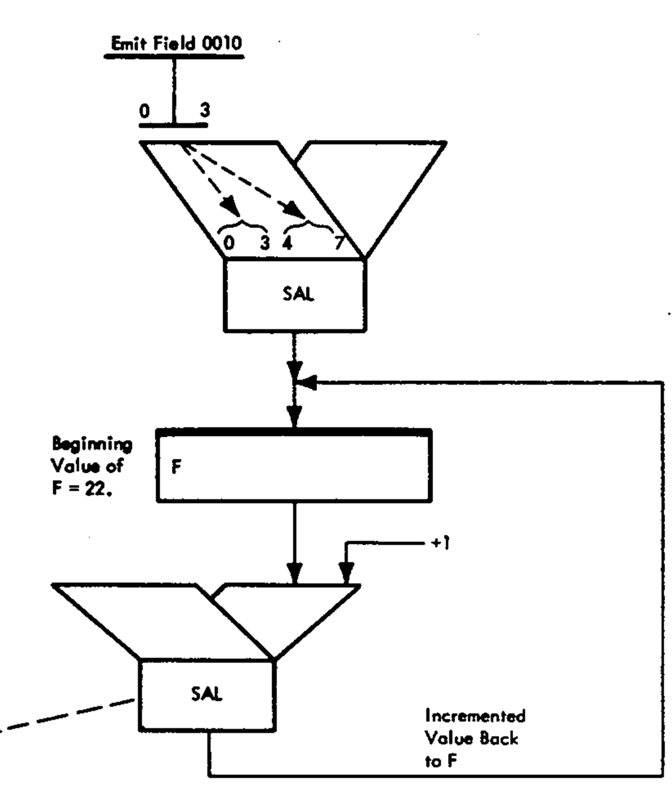
Diagram 5-808. Set Address Translator, SATR (0D), Execution in Issuing CE (Sheet 6 of 6)

2 3 4 5 6

Note:
The CE receiving a SATR select signal responds to the issuing CE and executes a microprogram to ingate the new ATR information from the external bus. This diagram shows the functions performed by the receiving CE.







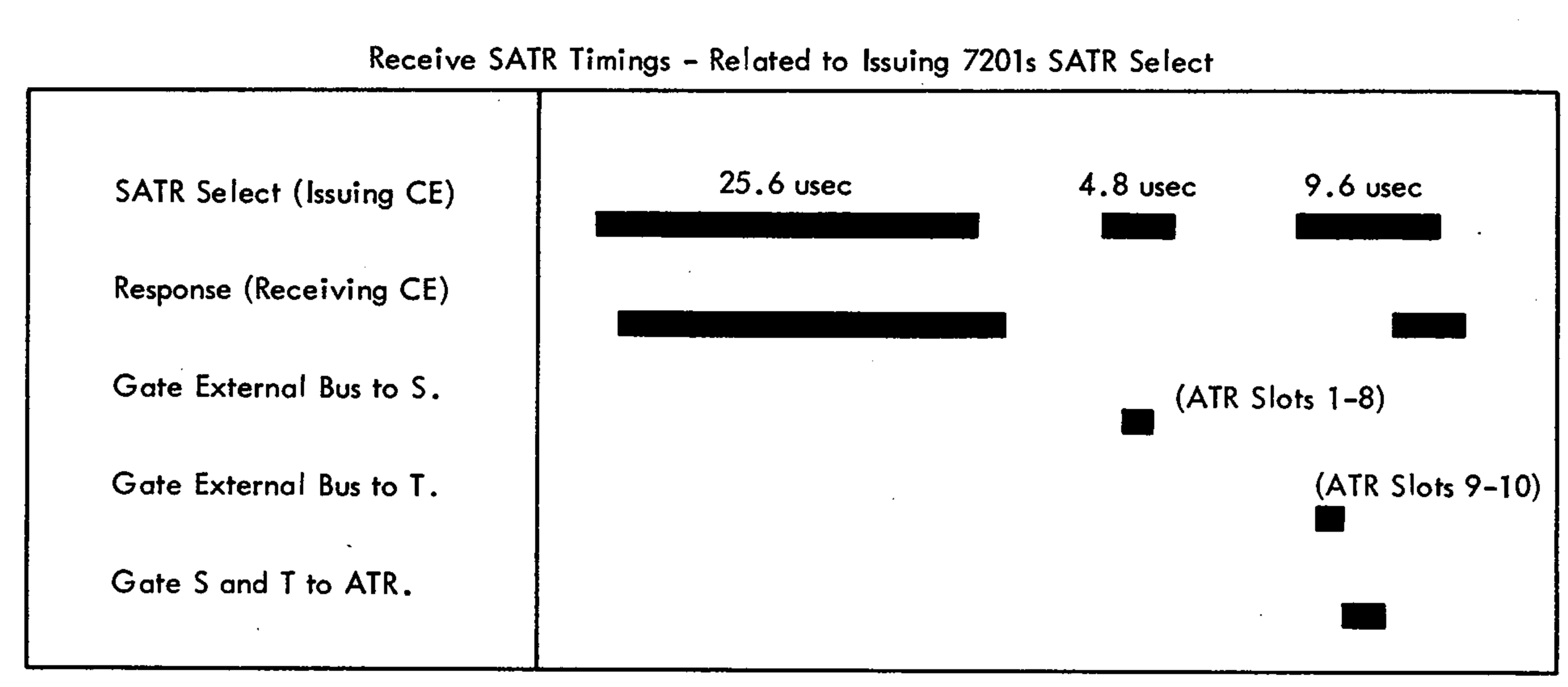


Diagram 5-809. Set Address Translator, SATR (0D), Execution in Receiving CE (Sheet 1 of 3)

2 3 4 5 6

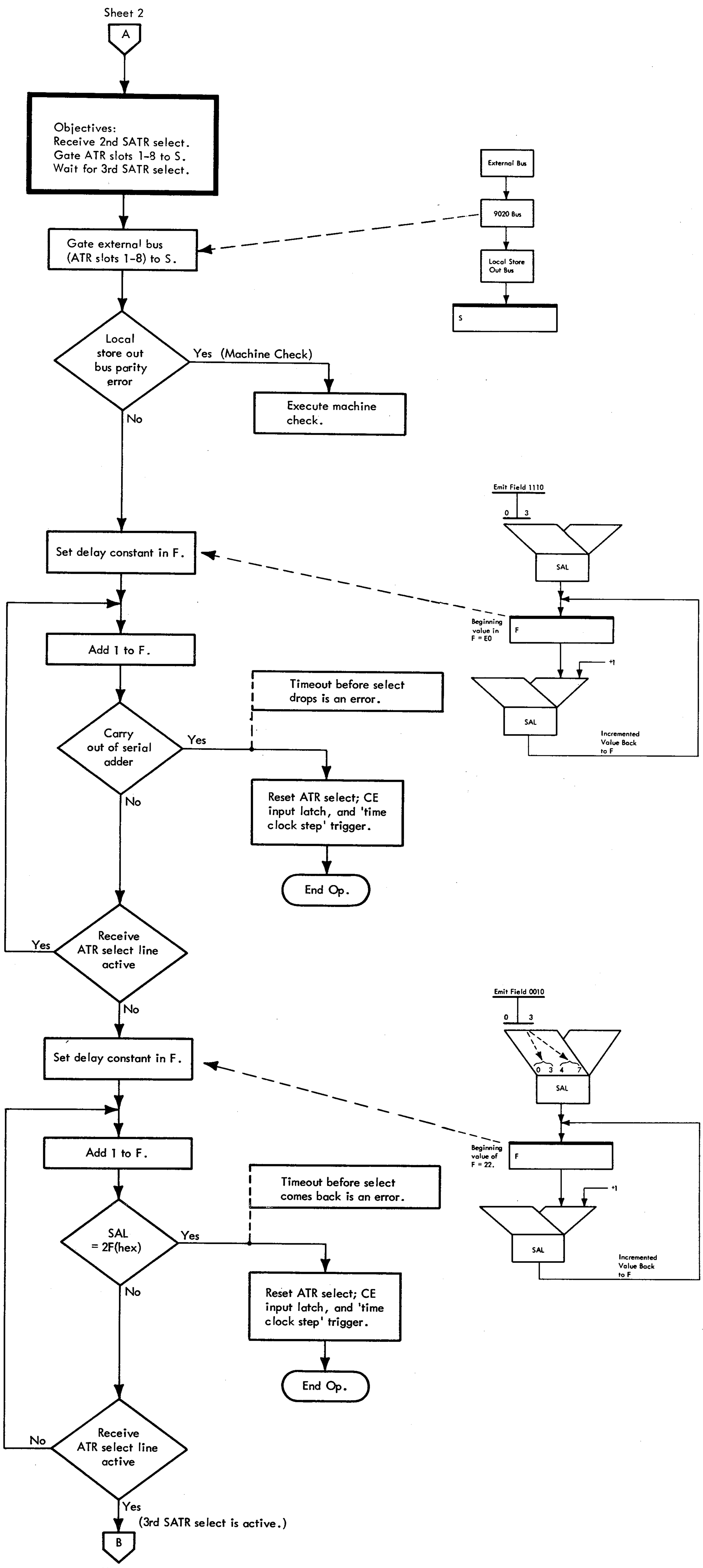


Diagram 5-809. Set Address Translator, SATR (0D), Execution in Receiving CE (Sheet 2 of 3)

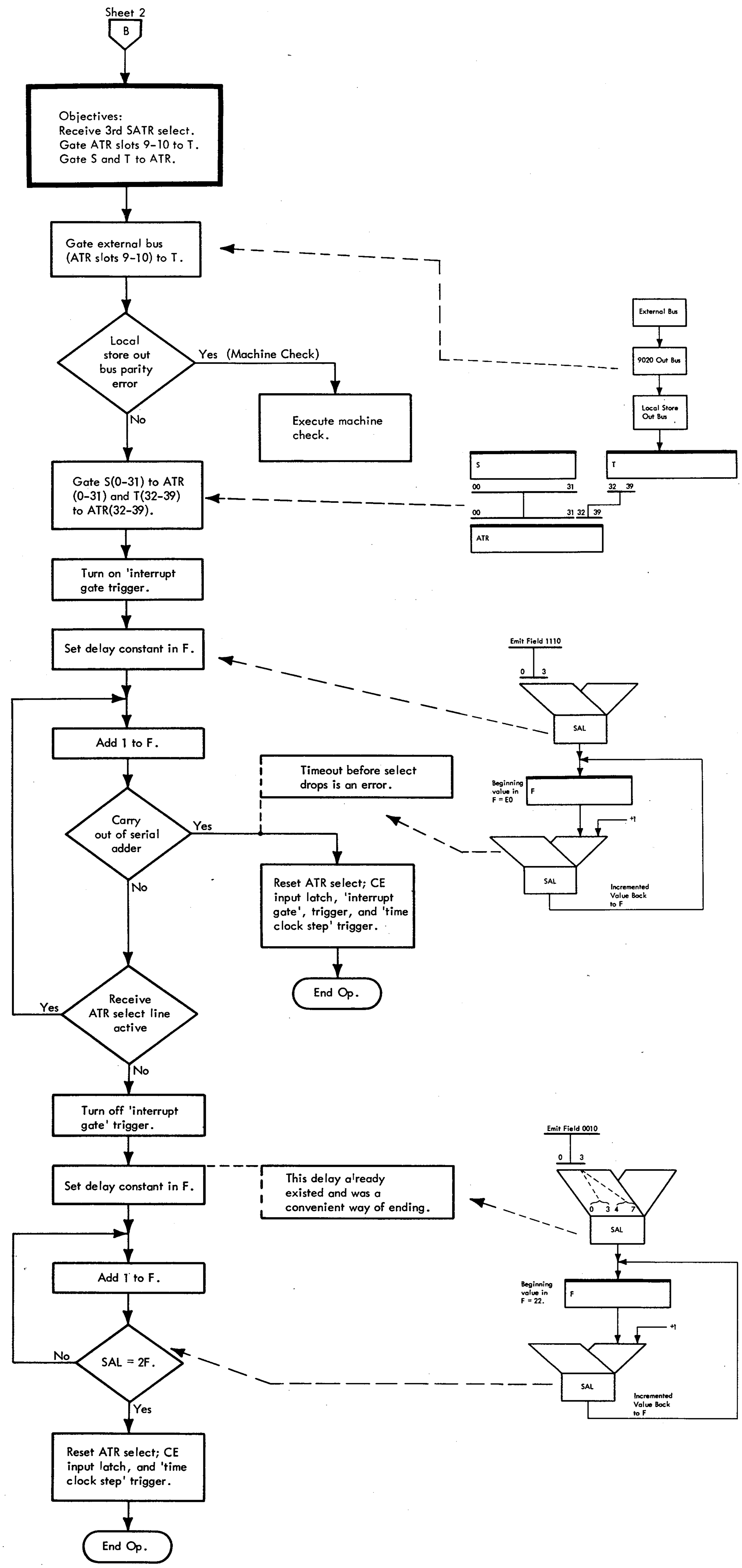


Diagram 5-809. Set Address Translator, SATR (0D), Execution in Receiving CE (Sheet 3 of 3)

- RR Format

01	R1	R2
0–7	8–11	12–15

- Purpose: To transfer configuration mask to CCR of element(s) specified by selection mask.
- Conditions at the end of I-Fetch:
  1. A-reg has 1st word of configuration mask (contents of GPR specified by R1).
  2. T-reg has selection mask (contents of GPR specified by R2).
- Configuration and selection mask formats:
  1. 9020D System

Configuration Mask — 1st Word

State	Scon		SE		CE		IOCE
S1 S2	1 2 3 -	‡ *	1 2 3 4 5 6 7 8 - -	* *	1 2 3 -	* * * * *	1 2 3
0 1	2 5	6 7	8 17	18 19	20 23	24 28	29 31

2nd Word

(Not Used)
0 31

Selection Mask

PAM	TCU		SE		CE		IOCE
1 2 3	1 2 3	* *	1 2 3 4 5 6 7 8 - -	* *	1 2 3 -	* * * * *	1 2 3
0 2	3 5	6 7	8 17	18 19	20 23	24 28	29 31

  2. 9020E System

Configuration Mask — 1st Word

State	Scon		SE	DE		CE		IOCE
S1 S2	1 2 3 -	‡ ǂ	1 2 3 4 -	1 2 3 4 -	* *	1 2 3 -	* * * * *	1 2 3
0 1	2 5	6 7	8 12	13 17	18 19	20 23	24 28	29 31

2nd Word

DG	(Not Used)
1A 1B 2B 2C 3C 3D 4D 4A 5A 5B 6B 6C 7C 7D 8D 8A	
0 15	16 31

Selection Mask

RCU		TCU		SE	DE		CE		IOCE
1 2	*	1 2 -	* *	1 2 3 4 -	1 2 3 4 -	* *	1 2 3 -	* * * * *	1 2 -
0 1	2	3 5	6 7	8 12	13 17	18 19	20 23	24 28	29 31

Legend:
- * Spare Bit
- \- Reserved Bit
- ‡ Inhibit Logout Stop Bit
- ǂ Inhibit DE Stop Bit

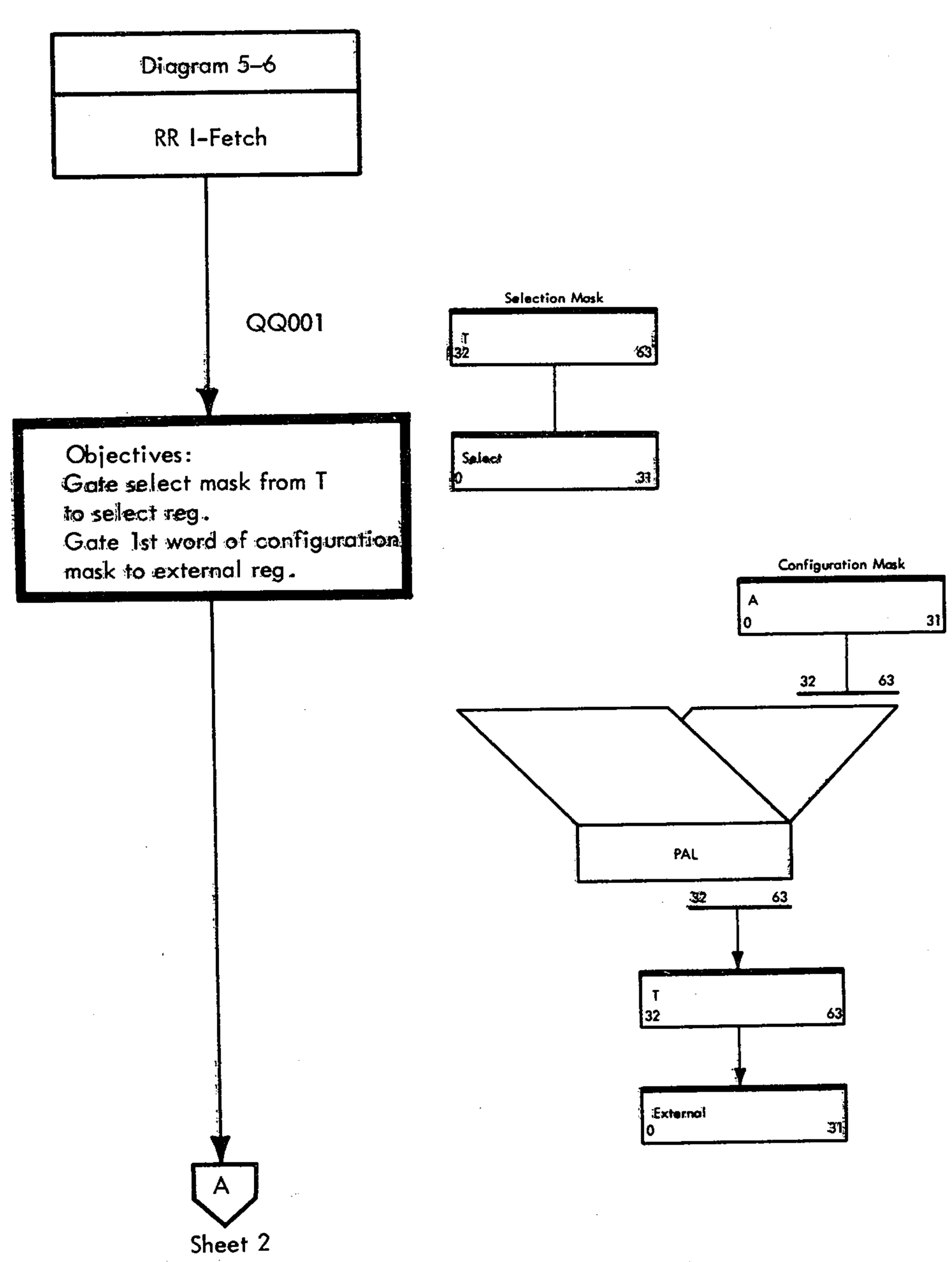


Diagram 5-810. Set Configuration, SCON (01) (Sheet 1 of 6)

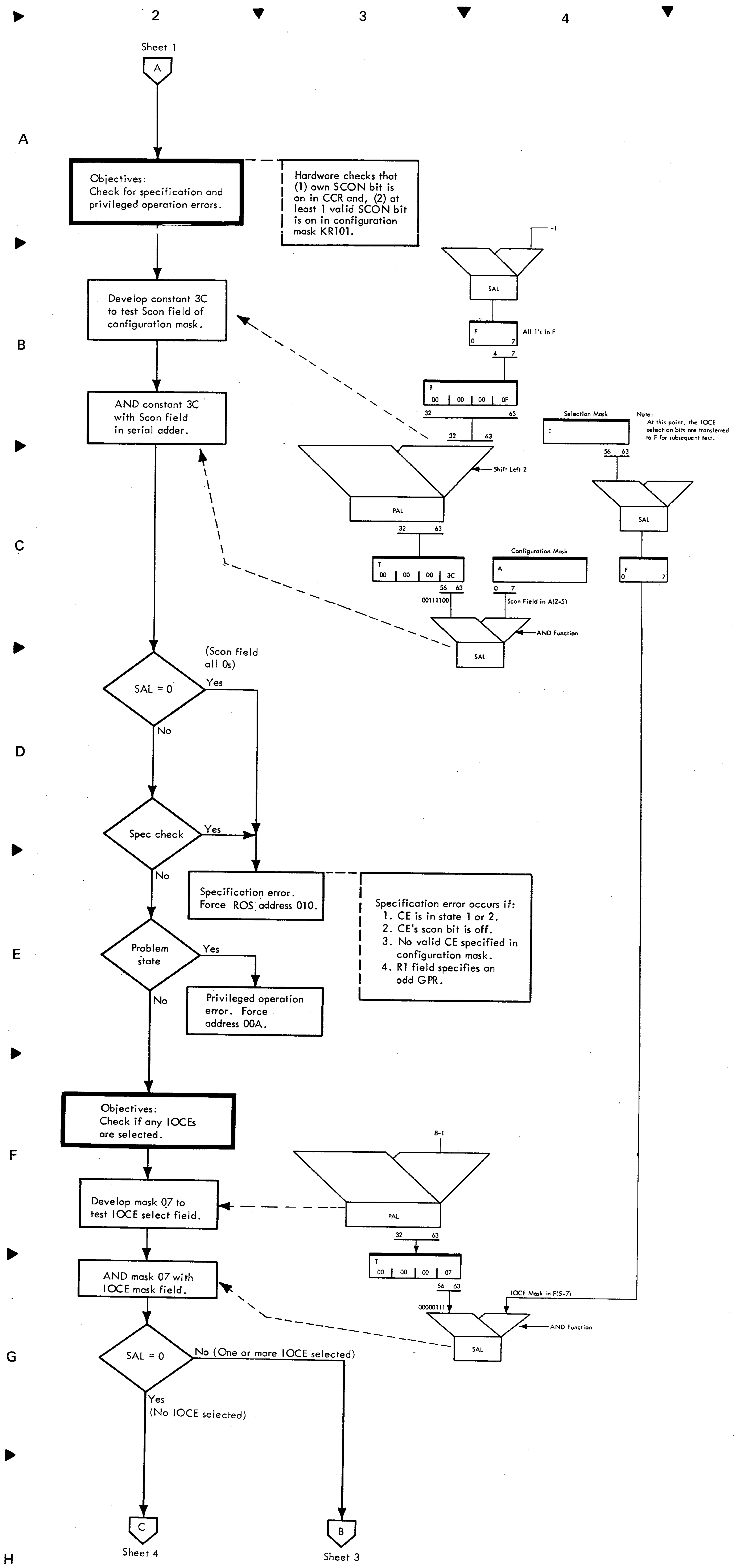


Diagram 5-810. Set Configuration, SCON (01) (Sheet 2 of 6)

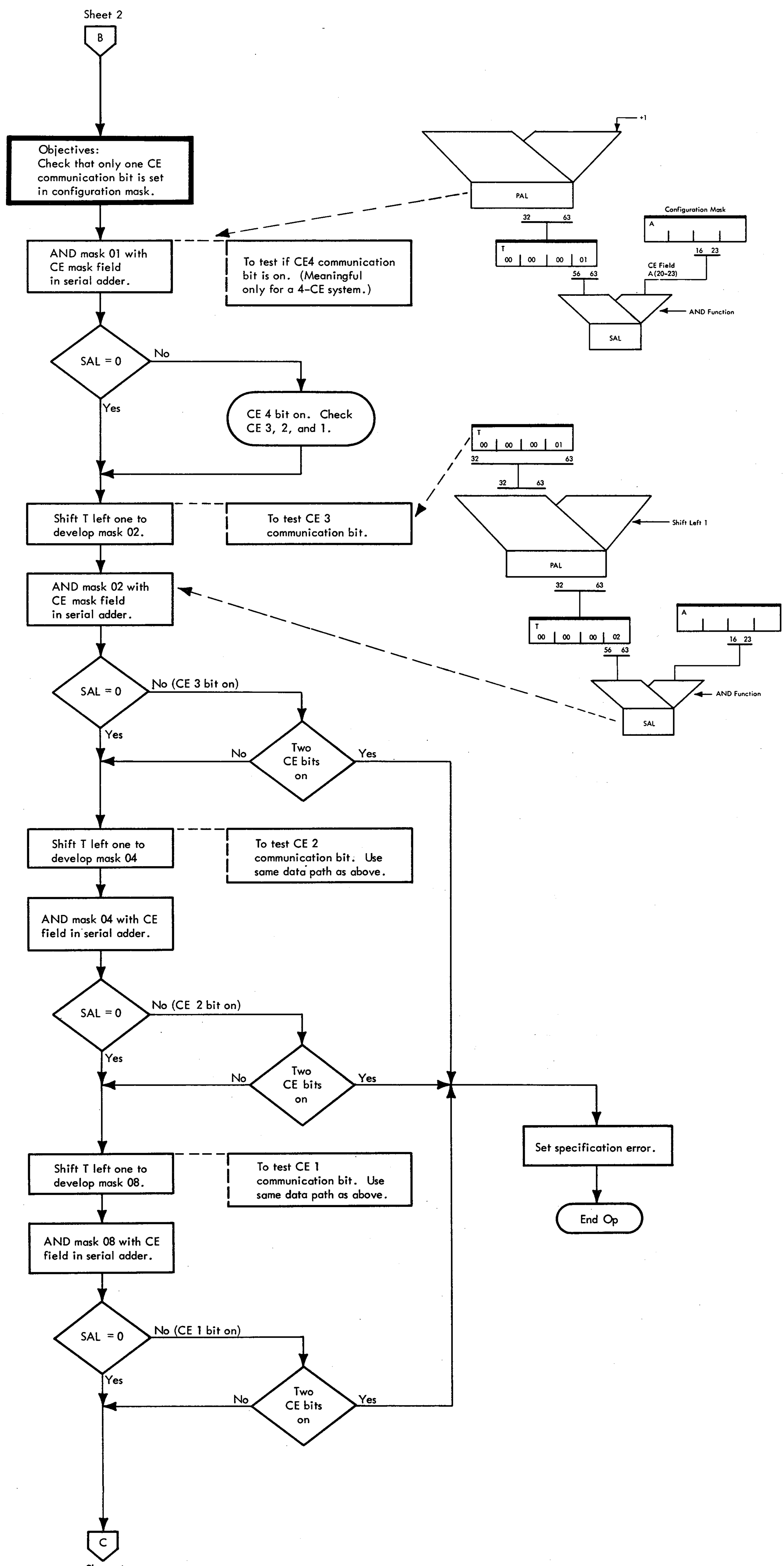


Diagram 5-810. Set Configuration, SCON (01) (Sheet 3 of 6)

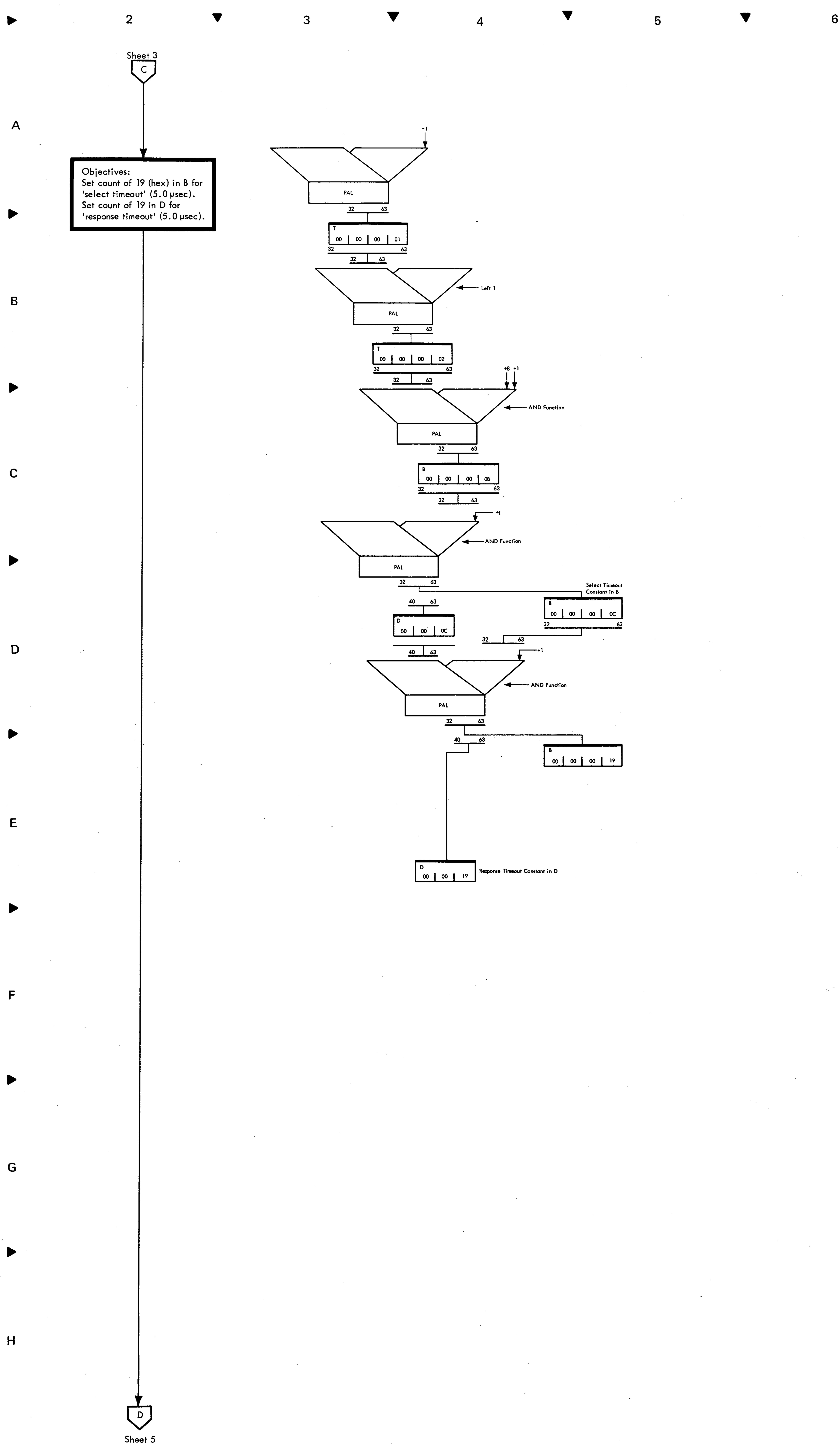


Diagram 5-810. Set Configuration, SCON (01) (Sheet 4 of 6)

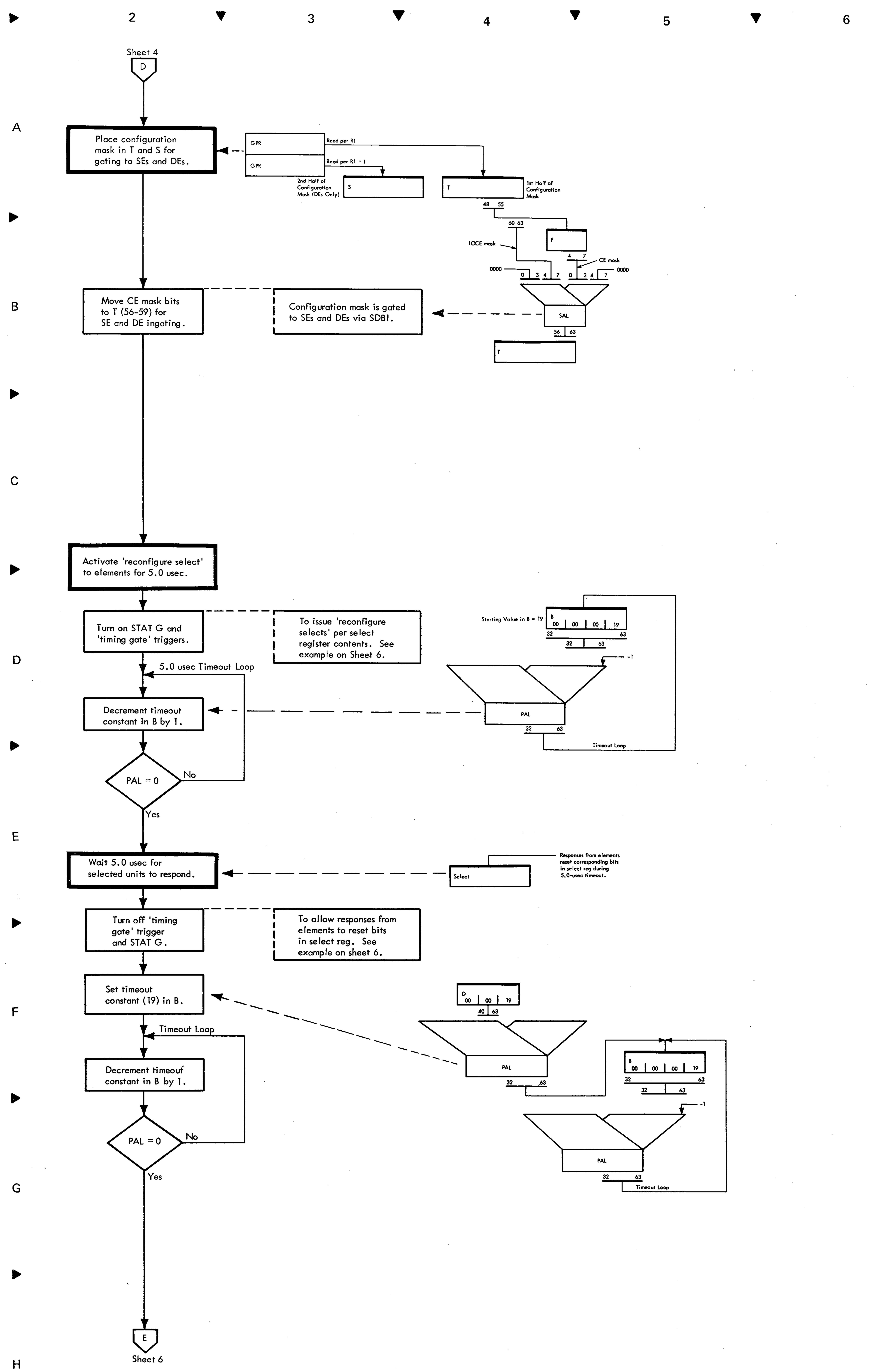
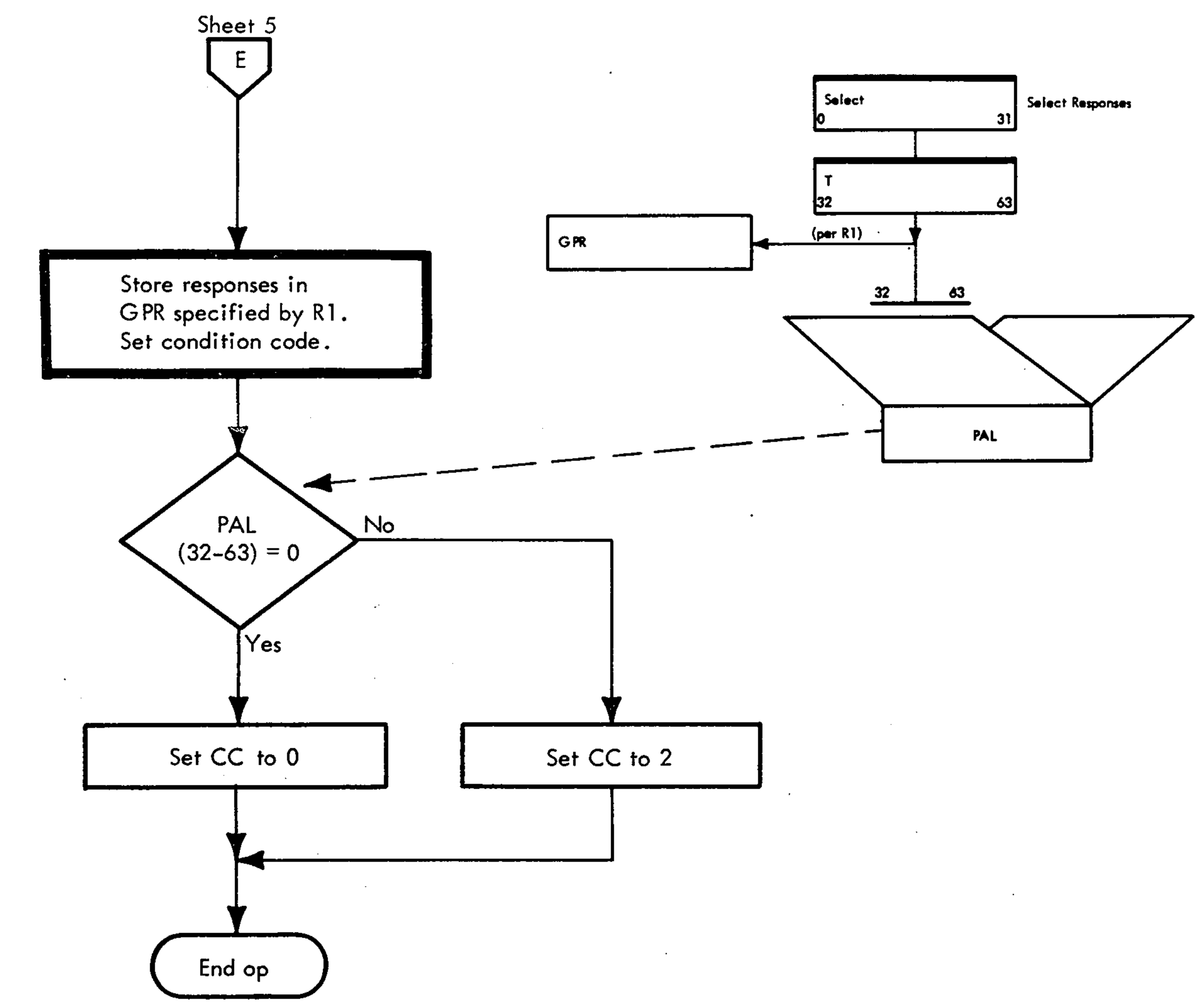


Diagram 5-810. Set Configuration, SCON (01) (Sheet 5 of 6)



The example below shows the basic logic active during the select and response timeouts. CE1 is executing the SCON instruction, and CE2 is receiving the 'scon select'. Similar select and response logic is used for all elements in the system.

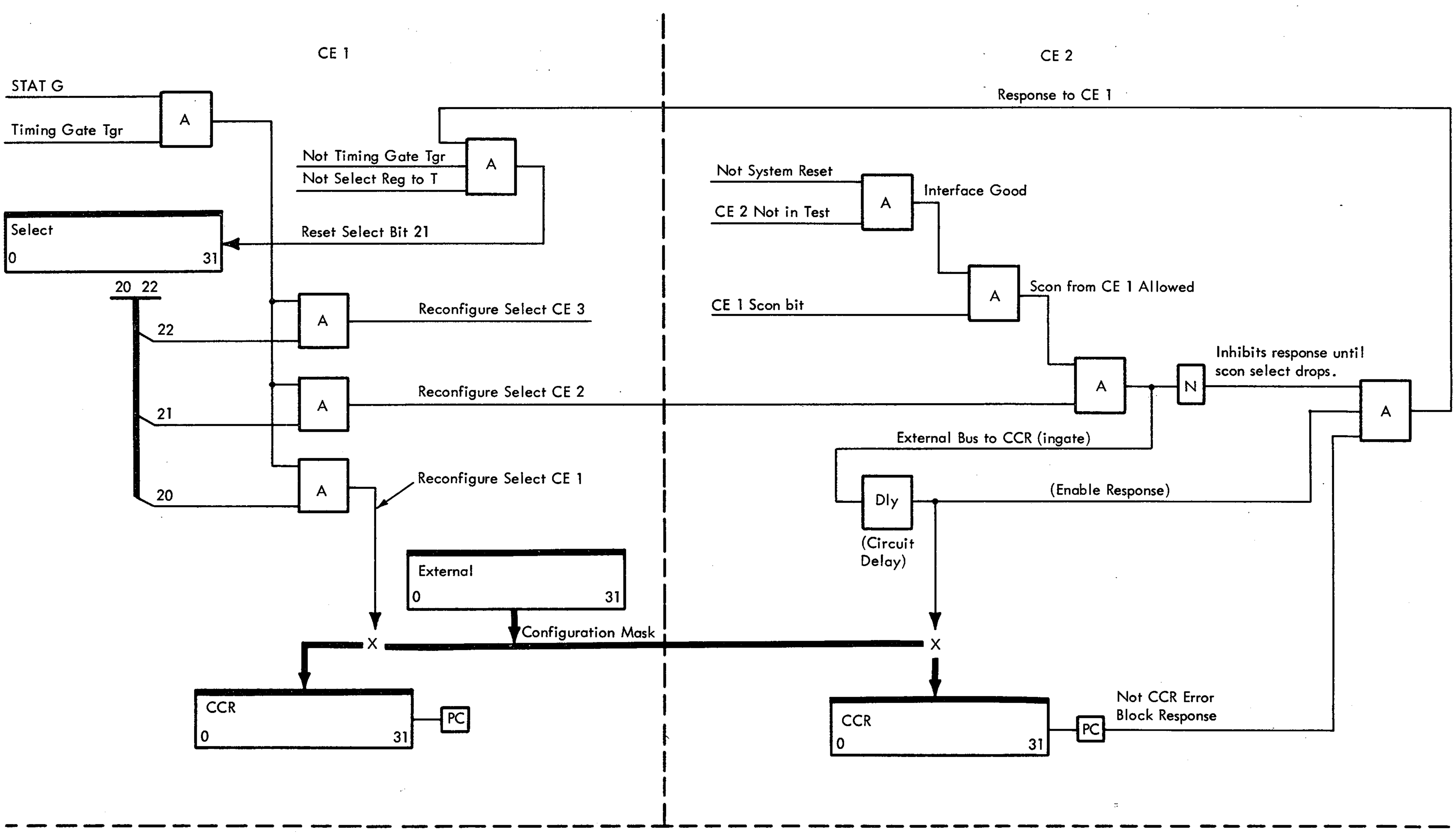
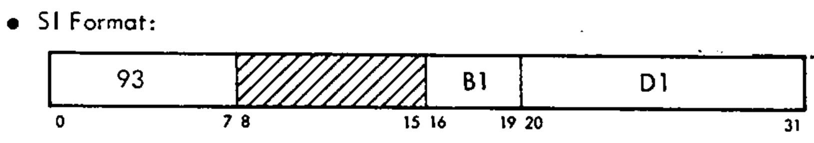


Diagram 5-810. Set Configuration, SCON (01) (Sheet 6 of 6)

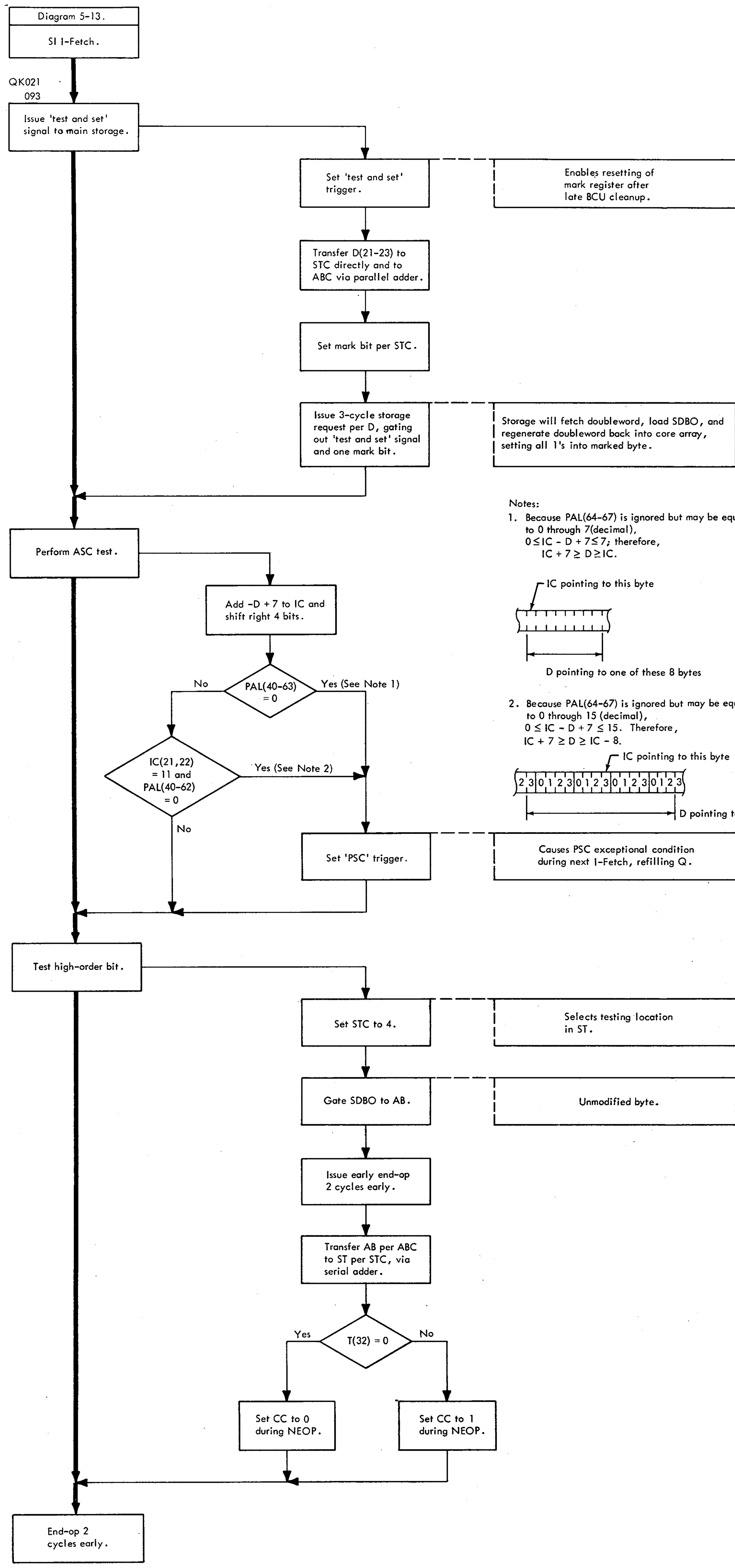
- SI Format:



- Test high-order bit (bit 0) of storage operand byte (in storage), set CC according to state of tested bit, and set addressed byte back into storage as all 1's.

- Conditions at start of execution:
  1. 1st 16 bits of instruction are in E.
  2. Storage operand address is in D.
  3. Storage request per D is blocked during I-Fetch ('D sync' latch blocked) because data byte could change before executing this instruction.

- 'Test and set' trigger: ALD MC181.



Notes:

1. Because PAL(64-67) is ignored but may be equal to 0 through 7(decimal), $0 \leq IC - D + 7 \leq 7$; therefore, $IC + 7 \geq D \geq IC$.

IC pointing to this byte

D pointing to one of these 8 bytes

2. Because PAL(64-67) is ignored but may be equal to 0 through 15 (decimal), $0 \leq IC - D + 7 \leq 15$. Therefore, $IC + 7 \geq D \geq IC - 8$.

IC pointing to this byte

2 3 0 1 2 3 0 1 2 3 0 1 2 3 0 1 2 3

D pointing to one of these 16 bytes

Diagram 5-811. Test and Set, TS (93)

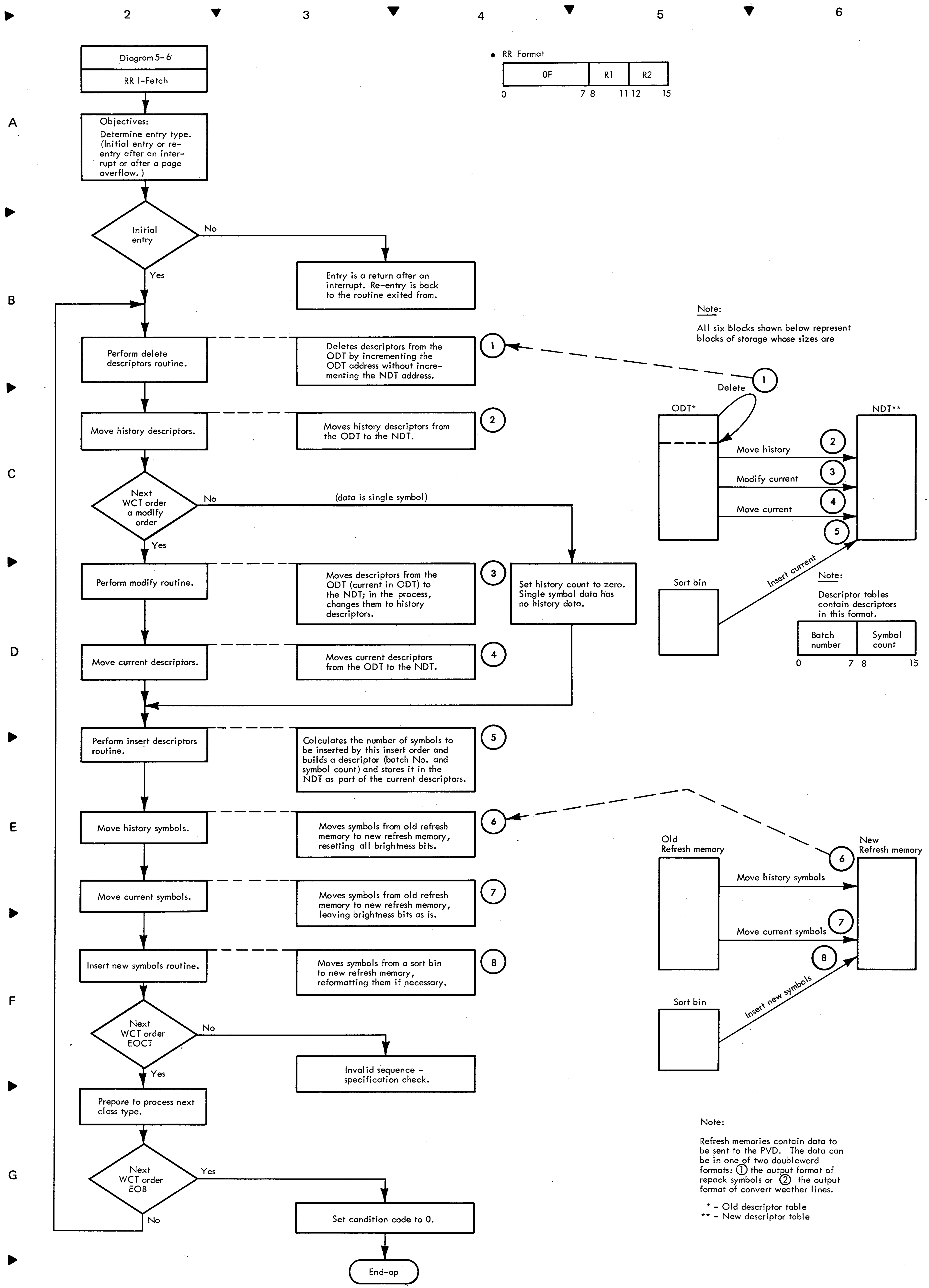


Diagram 5-901. Repack Symbols, Simplified Flow Chart

2 3 4 5 6

Diagram 5-6

RR I - Fetch

Objectives:

Determine entry type whether initial entry or re-entry after an interrupt.

Prepare to enter the delete descriptors routine.

* EIE Sheet 21 (Re-entry from interrupt routine.)

QQ401 D8F

Gate next WCT address (GPR 7) to A (via S).

Gate INTRPT ID (bits 4-7) and bin number (GPR 8) to S.

Save 360 IC by gating to LSWR (via T).

Add the WCTA (A) and (2X bin no.) and gate result to IC.

Each bin displacement value is two bytes long. Adding two times the bin number increments the WCTA to the correct bin displacement value.

Gate INTRPT ID to F.

Gate DP bits (GPR 11) to T.

Make storage request for bin displacement address from WCT.

Gate F(4-7) (INTRPT ID) to PAL for test.

Rearrange INTRPT ID - bits 4-7 to F(0-3) and bits 0-3 to F(4-7).

Clear K register.

Used to accumulate symbol count throughout RPSB execution.

Gate next WCT (GPR 7) address to S.

Initial Entry

PAL = 0 — No: (INTRPT ID (4-7) not zero indicates a re-entry after an interrupt.) → CFE Sheet 2

Yes

DP bit (T32) a one — Yes: (This is re-entry after a page overflow.) → F87 Reset DP bits in GPR 11. → * CGE Sheet 3

No

Reset DP bits in GPR 11.

F88

Gate first WCT order address to T and D and GPR 7.

Adding 32 to the first WCTA increments it past the halfword bin displacements to the first order.

Gate SDBO (bin disp address) to AB.

Clear T register.

Gate old descriptor table (ODT) address to S.

F8D

Make memory request per IC for ODT doubleword.

Store bin displacement address (T) in GPR 9.

Gate SDBO (ODT doubleword) to L, M and Q.

Compute sort bin address and store it in GPR 6.

This address will be used in calculating the symbol count when a new descriptor is constructed in the insert descriptors routine.

A Sheet 4

- RR Format

0F	R1	R2
0 – 7	8 – 11	12 – 15

- Purpose: Assembles a new updated display image (refresh memory) for 1/16 of a PVD's area. It accomplishes this by deleting, from the old display image, all information which is no longer to be displayed, by moving (and modifying, if necessary) the data remaining in old refresh memory (ORM) to new refresh memory (NRM), and by adding new data to be displayed by inserting data from a sort bin (created by CSS execution) into NRM.

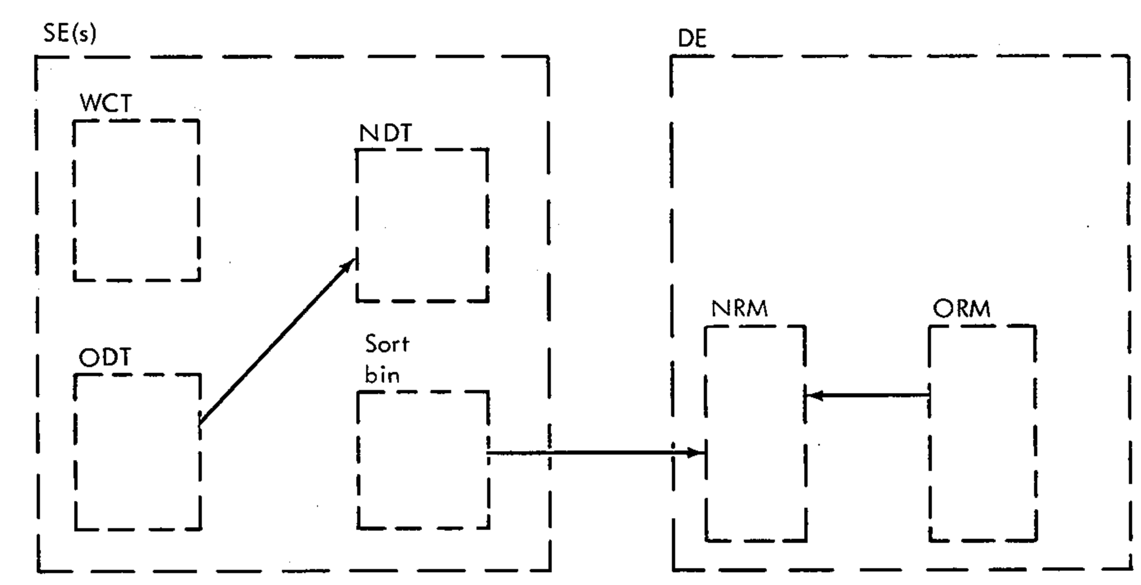
- RPSB moves two types of data, descriptors

Batch Number	Symbol Count

and display words,

0	S	Symbol	BL	BR	0	0	Y	X
0	1	2 – 7	8	9		12	– 21	22 – 31

and is logically divided into two sections. The first builds a new descriptor table by using the applicable previous descriptors (from the ODT) and by generating new descriptors for the new input data (from CSS execution(s)) located in a sort bin. This section operates only in SEs. The second section builds a new display image for a sort bin (called new refresh memory, NRM), using the symbol counts accumulated while building the new descriptor table (NDT) which actually describes NRM. This section operates in both SEs and DEs as shown below.



* 3-letter designations identify actual line edge connectors on CAS page.

Diagram 5-902. Repack Symbols, RPSB (0F) (Sheet 1 of 21)

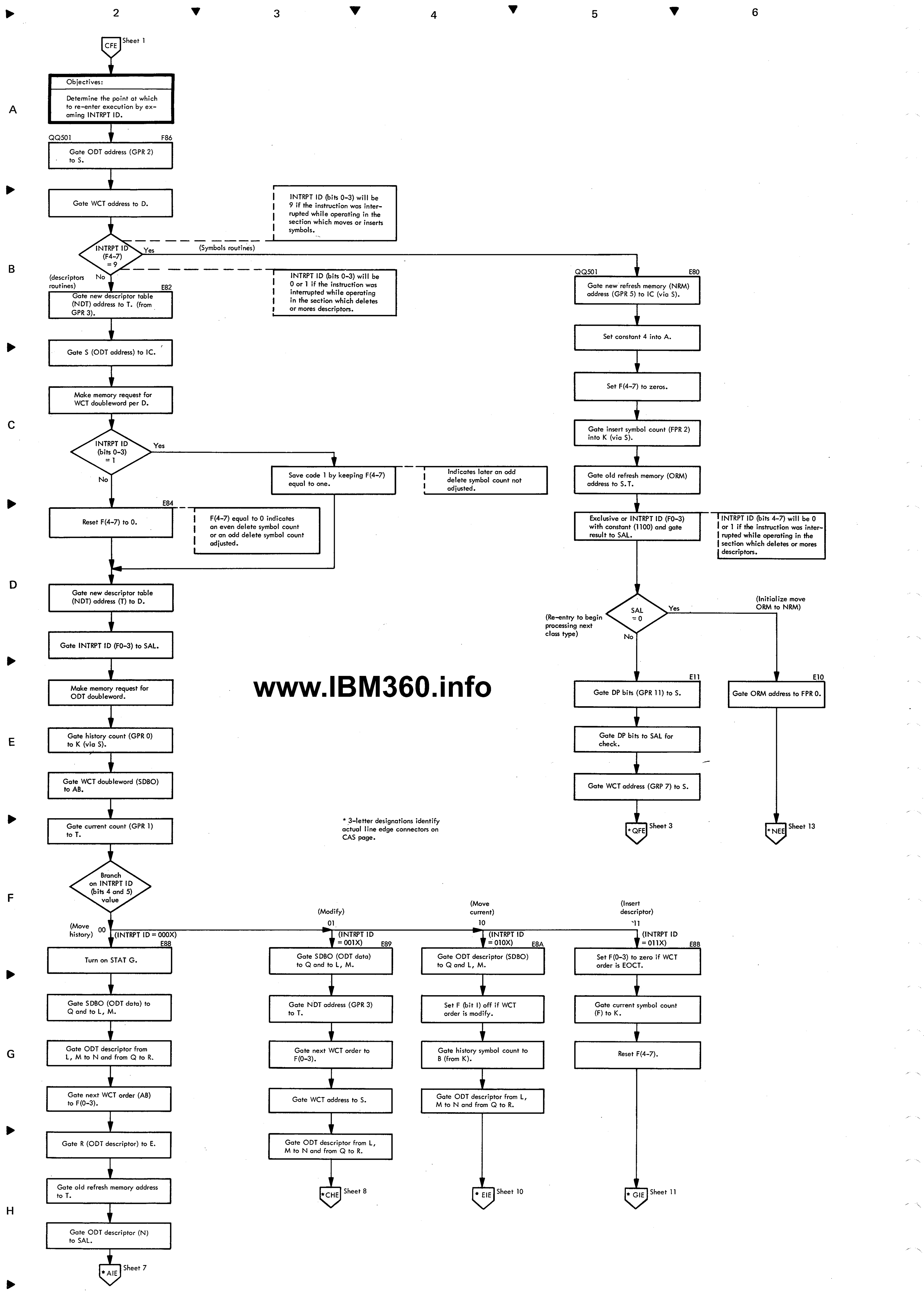


Diagram 5-902. Repack Symbols, RPSB (0F) (Sheet 2 of 21)

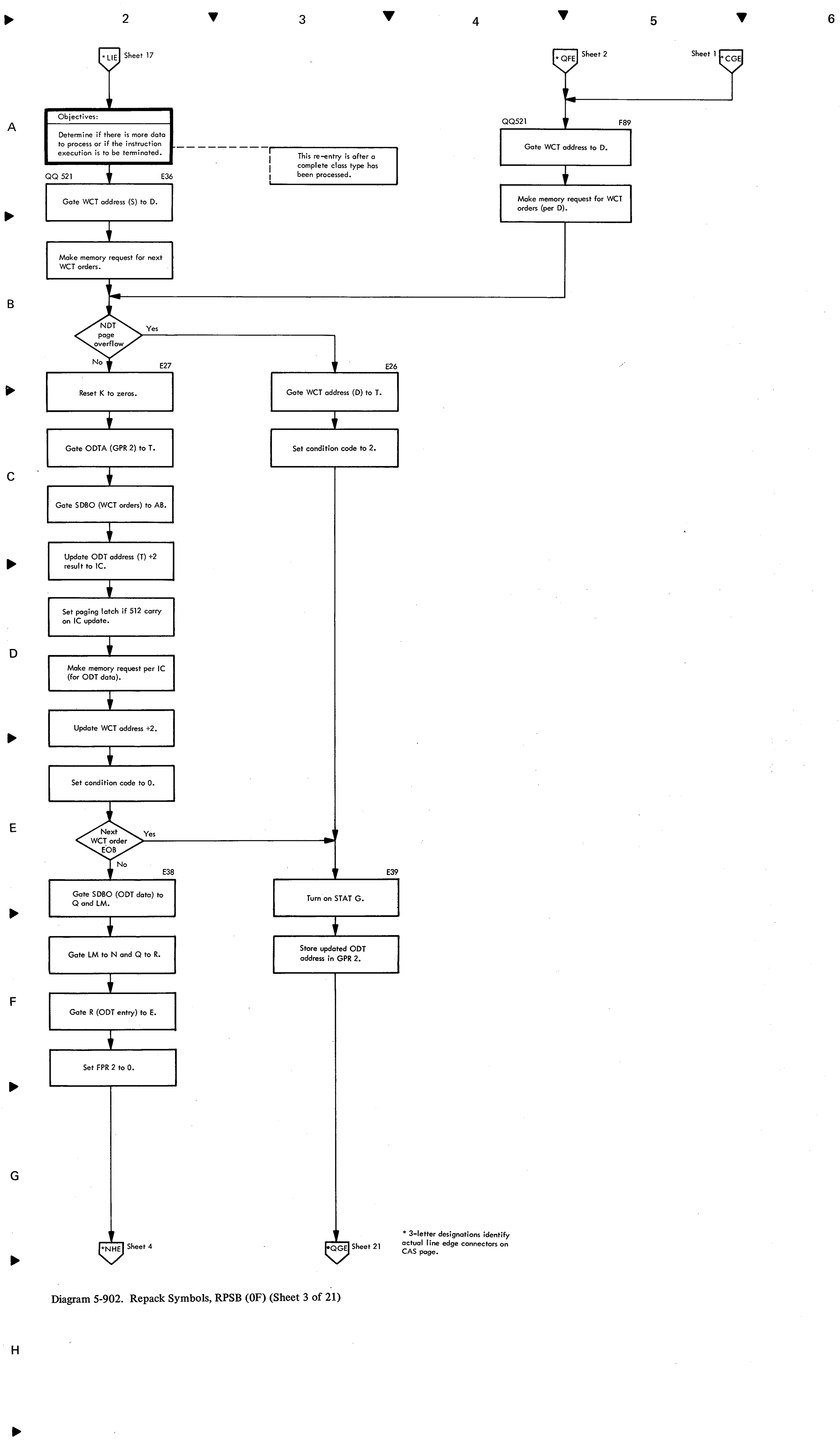


Diagram 5-902. Repack Symbols, RPSB (0F) (Sheet 3 of 21)

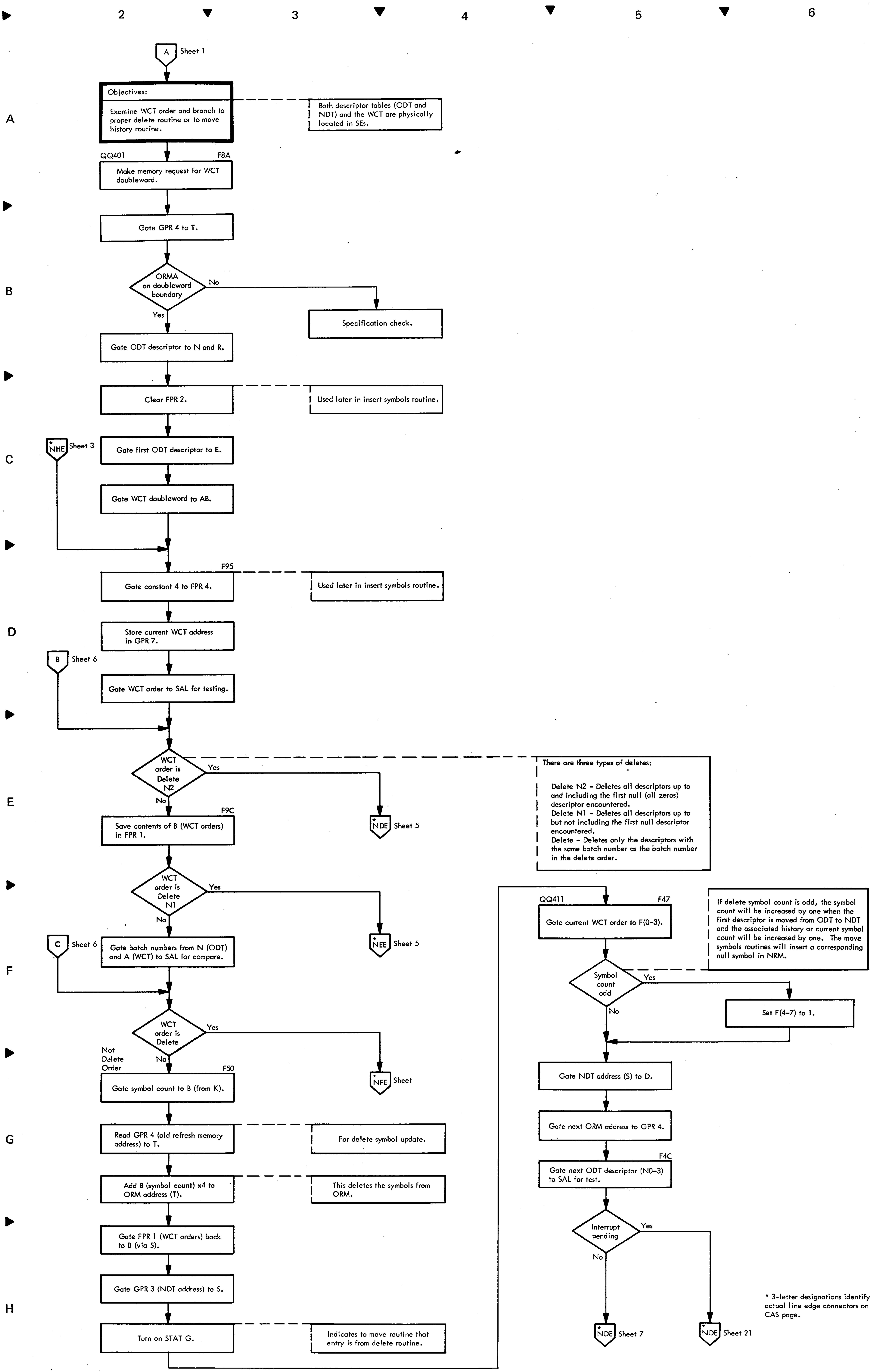


Diagram 5-902. Repack Symbols, RPSB (0F) (Sheet 4 of 21)

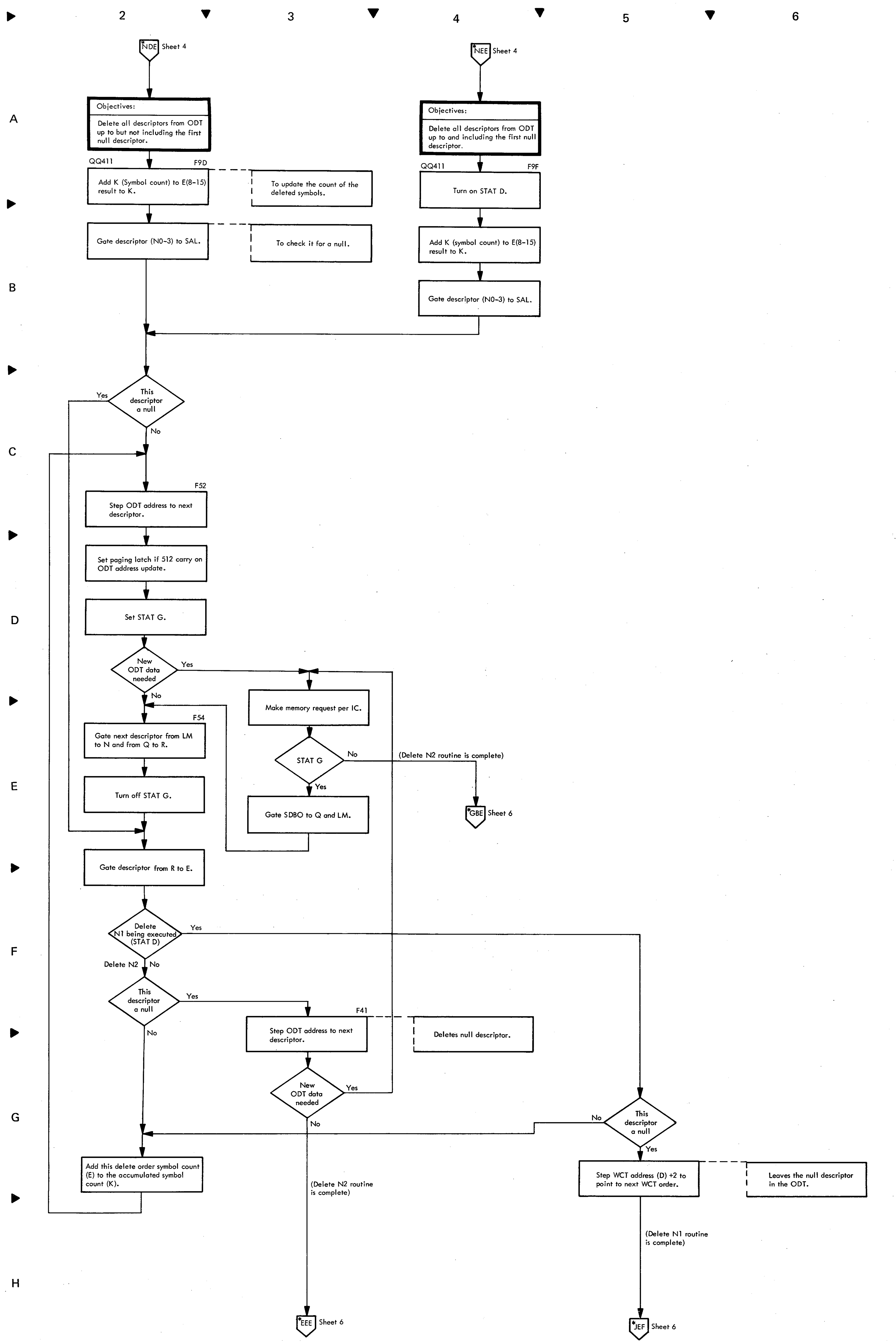


Diagram 5-902. Repack Symbols, RPSB (0F) (Sheet 5 of 21)

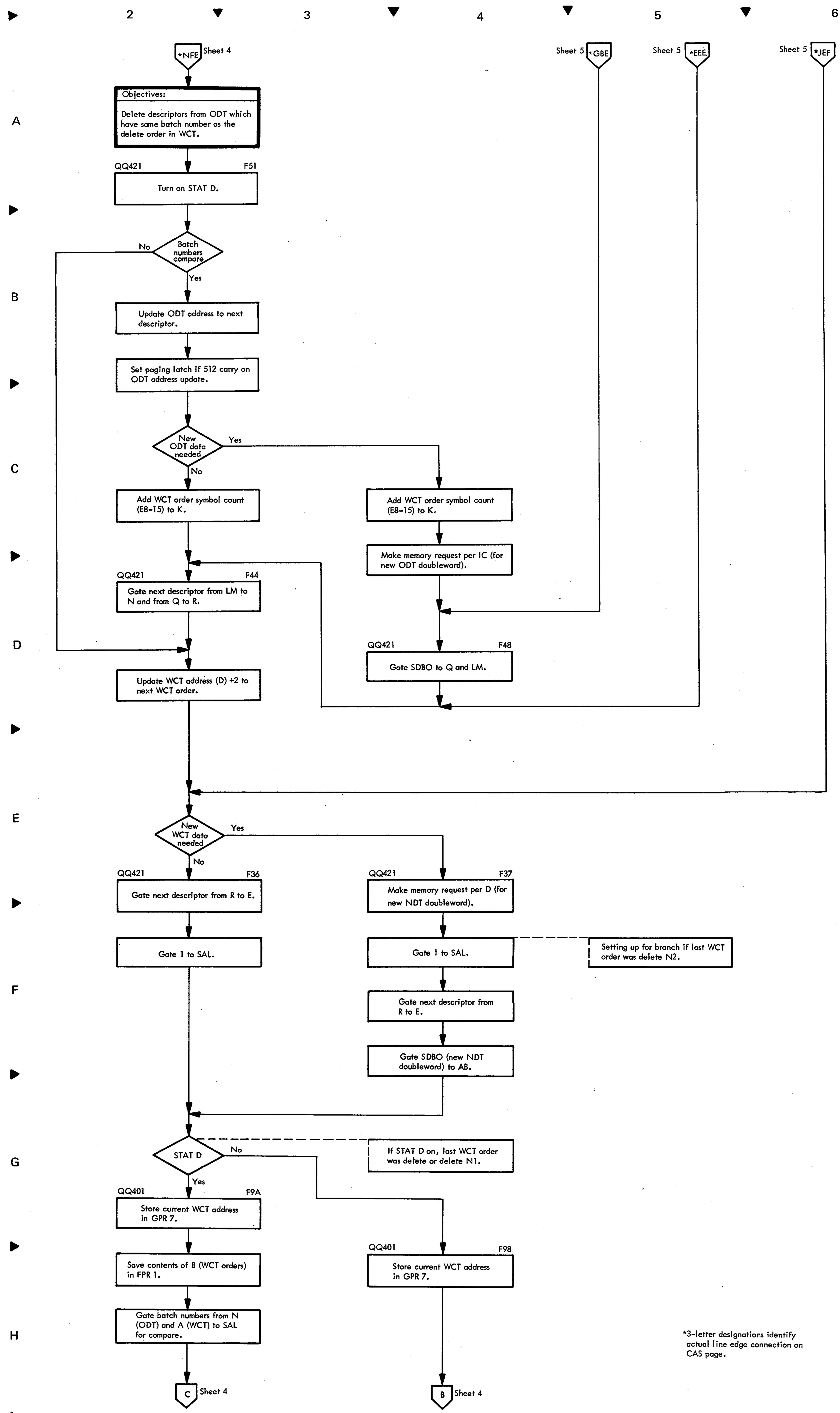


Diagram 5-902. Repack Symbols, RPSB (0F) (Sheet 6 of 21)

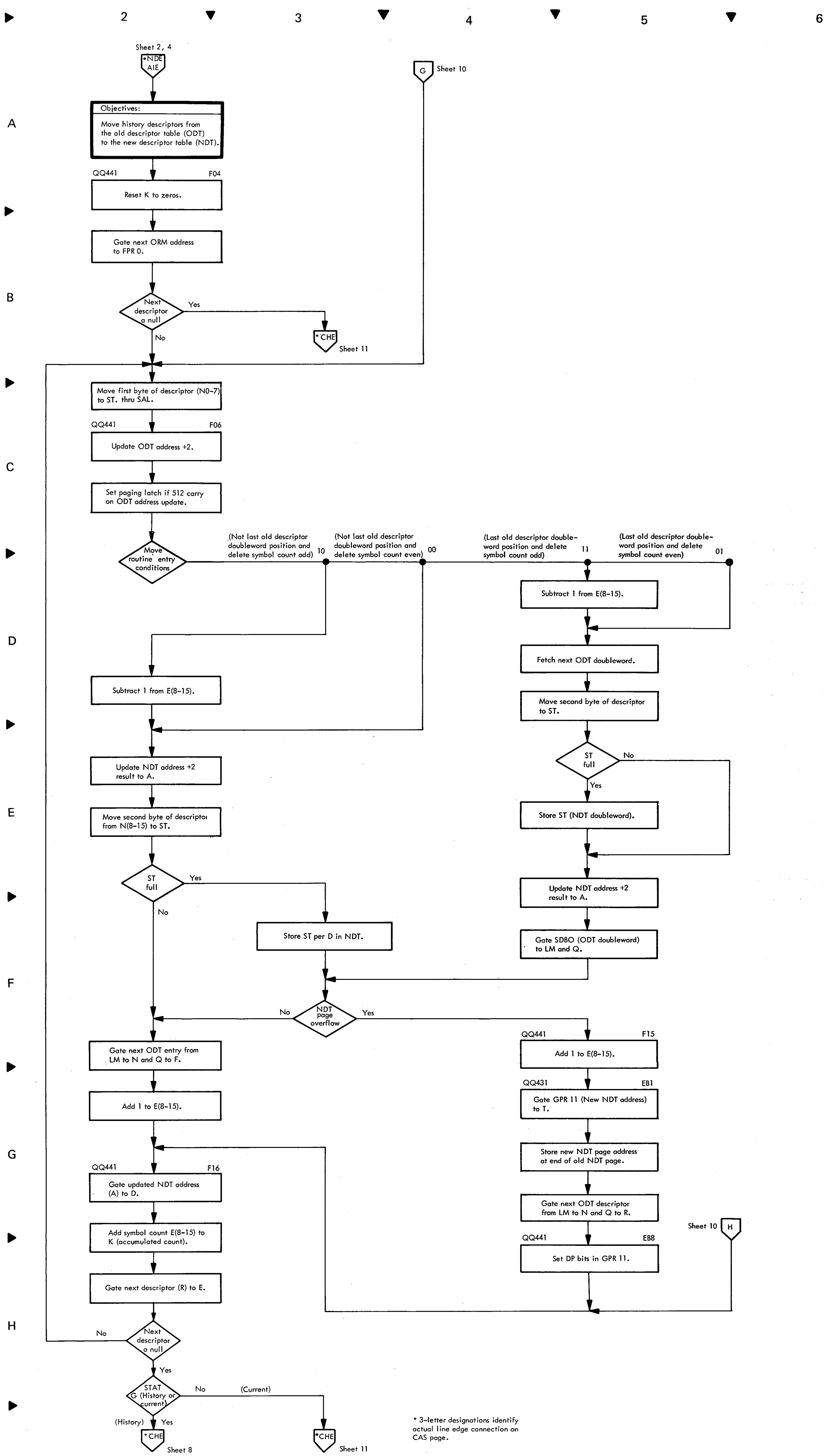


Diagram 5-902. Repack Symbols, RPSB (0F) (Sheet 7 of 21)

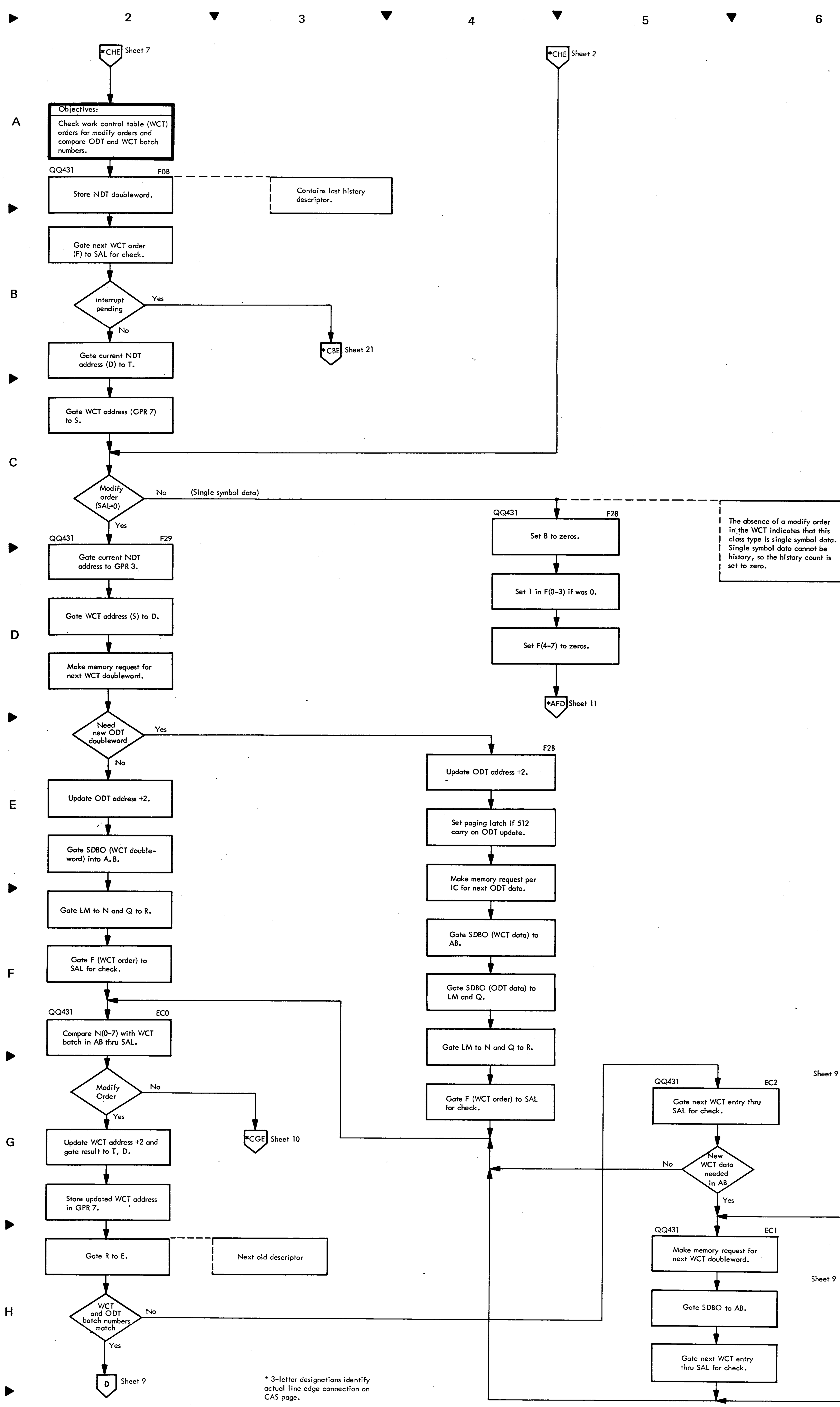


Diagram 5-902. Repack Symbols, RPSB (0F) (Sheet 8 of 21)

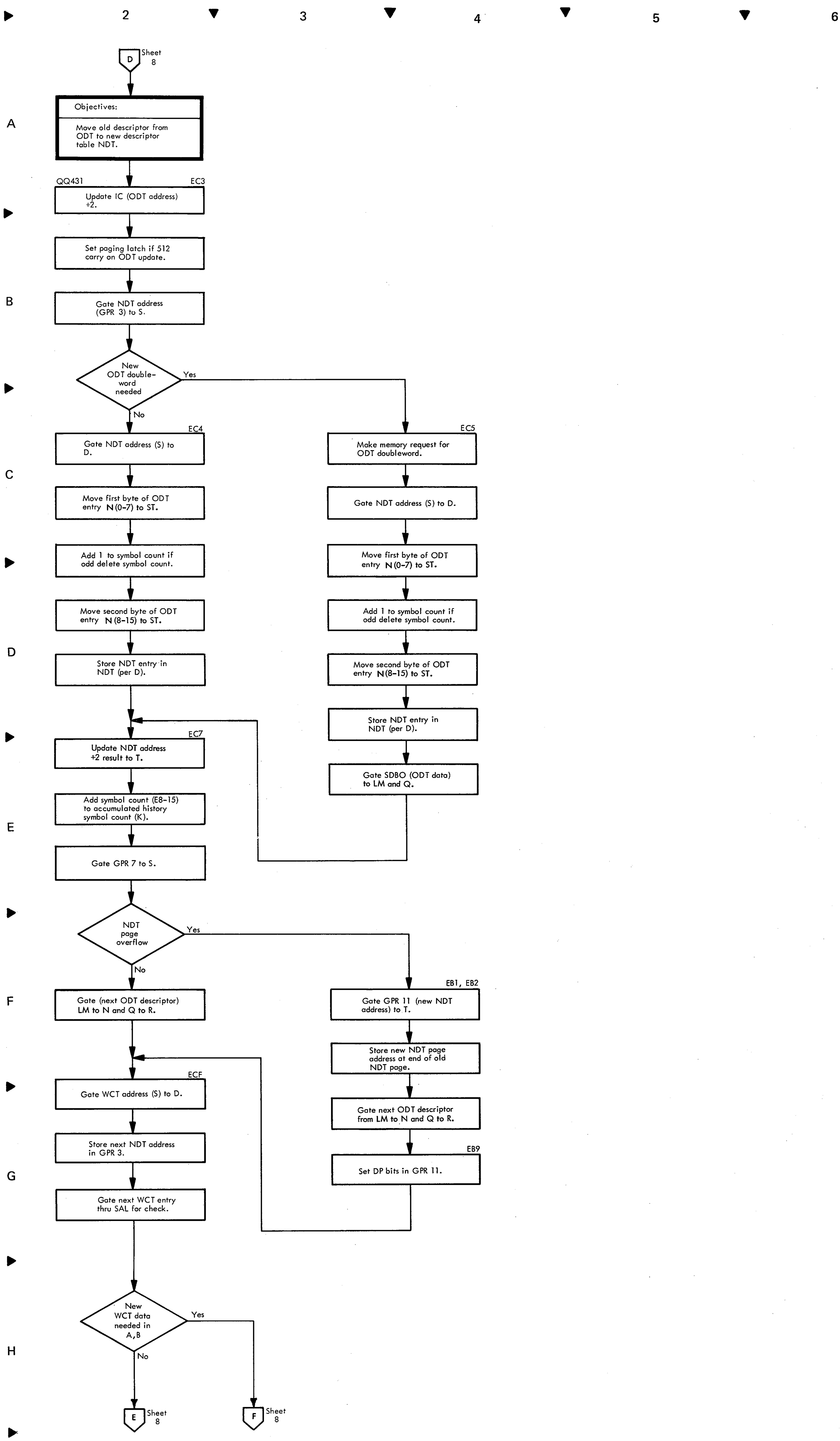


Diagram 5-902. Repack Symbols, RPSB (0F) (Sheet 9 of 21)

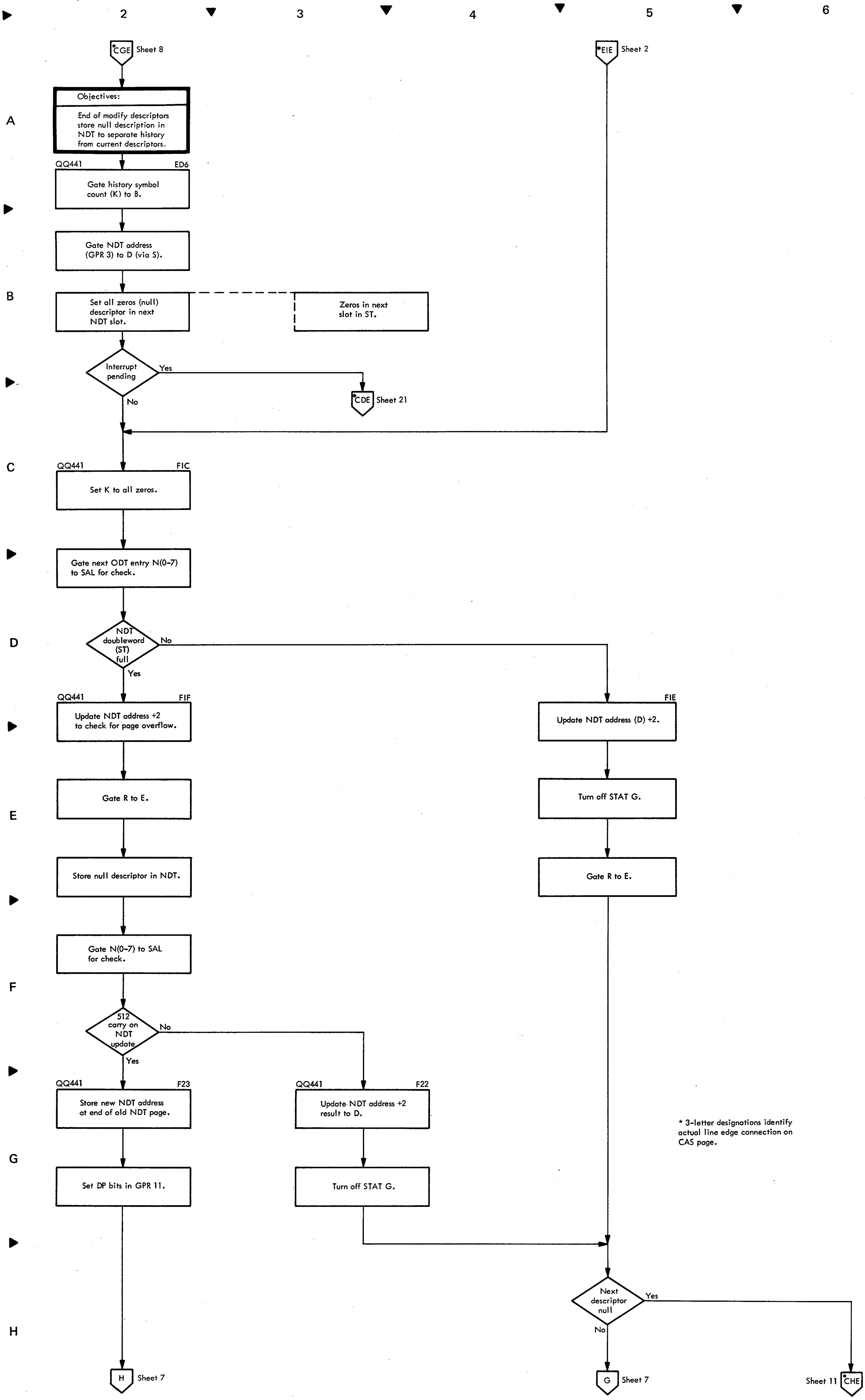


Diagram 5-902. Repack Symbols, RPSB (0F) (Sheet 10 of 21)

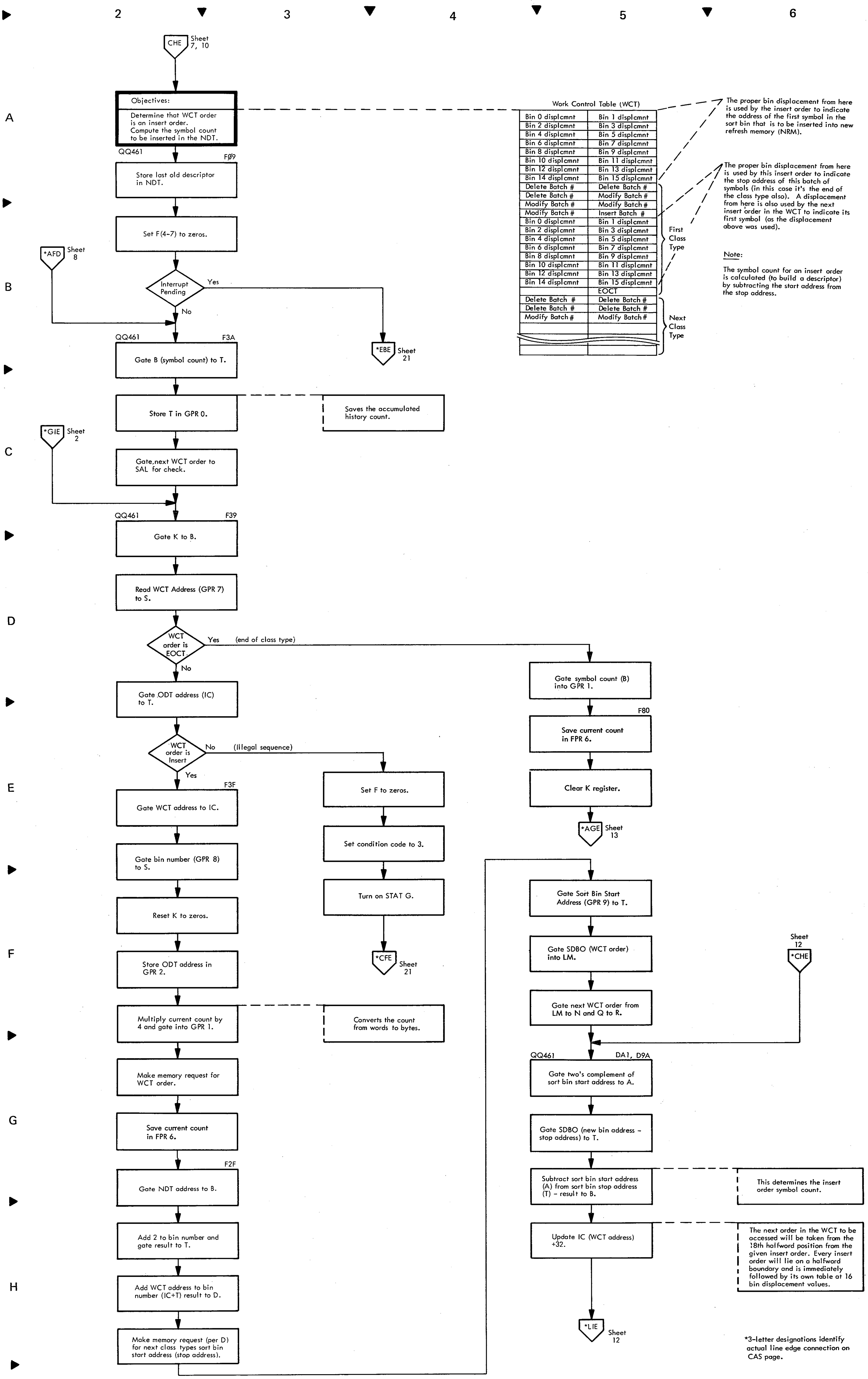


Diagram 5-902. Repack Symbols, RPSB (0F) (Sheet 11 of 21)

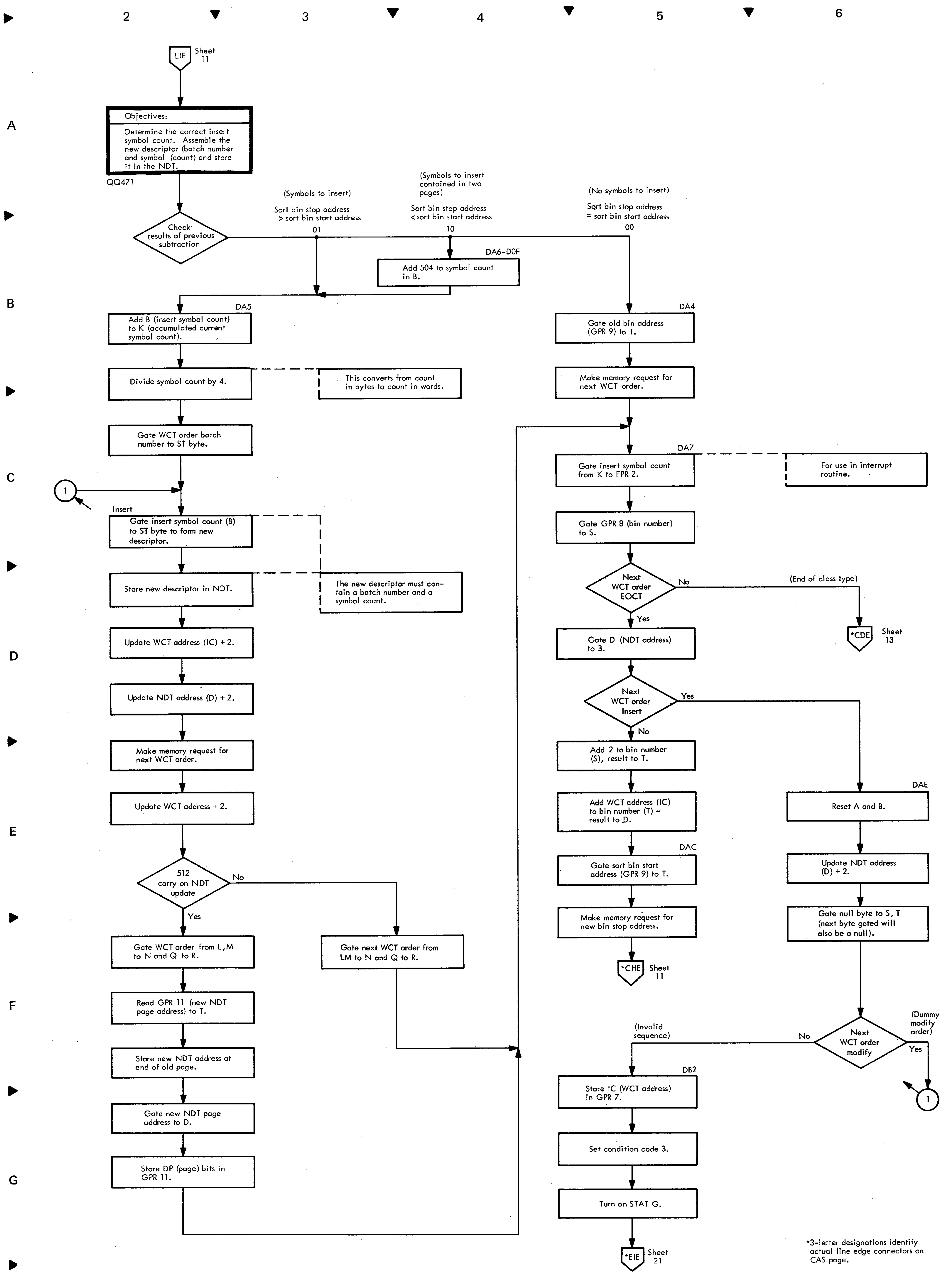


Diagram 5-902. Repack Symbols, RPSB (0F) (Sheet 12 of 21)

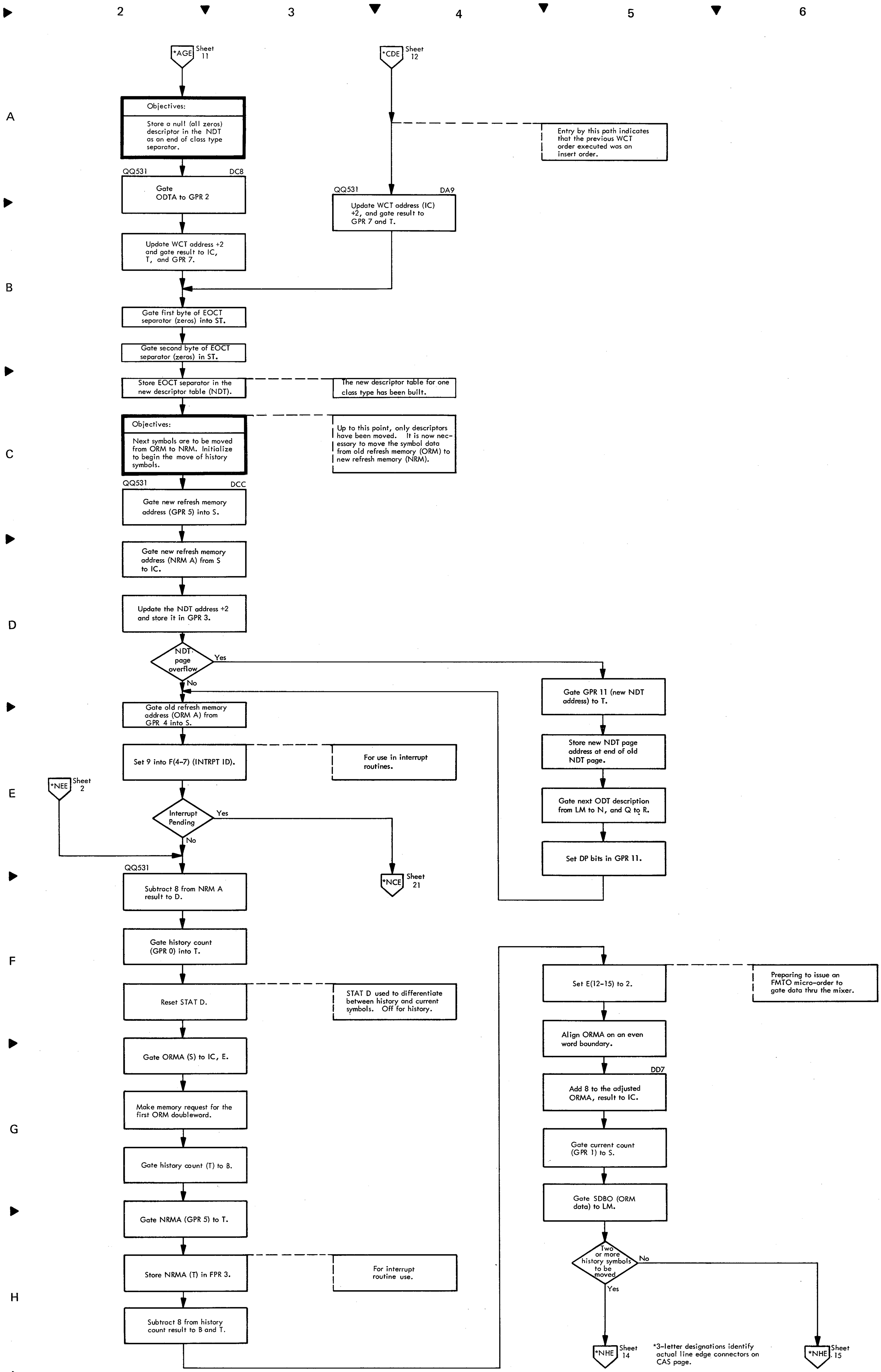


Diagram 5-902. Repack Symbols, RPSB (0F) (Sheet 13 of 21)

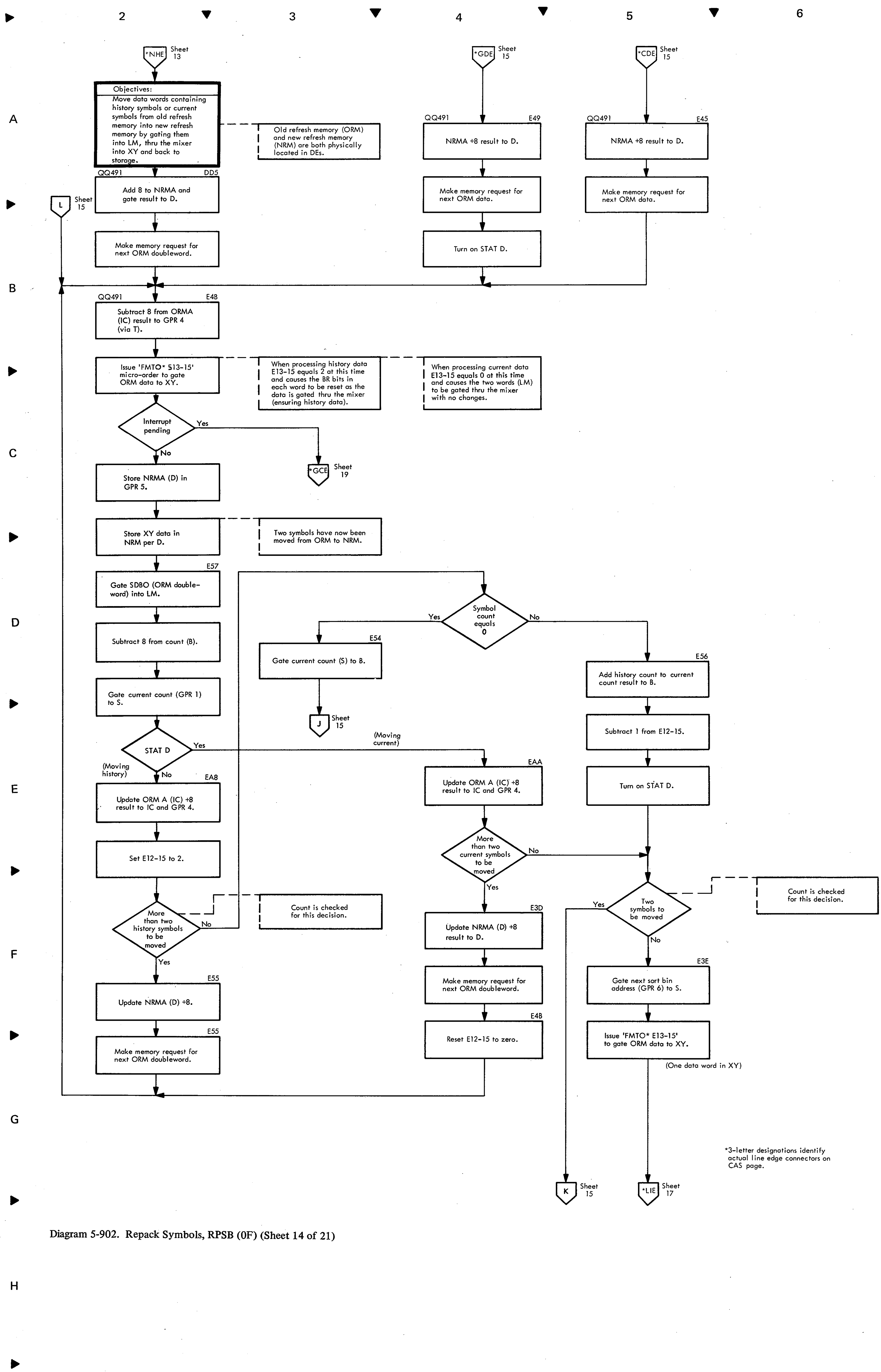


Diagram 5-902. Repack Symbols, RPSB (0F) (Sheet 14 of 21)

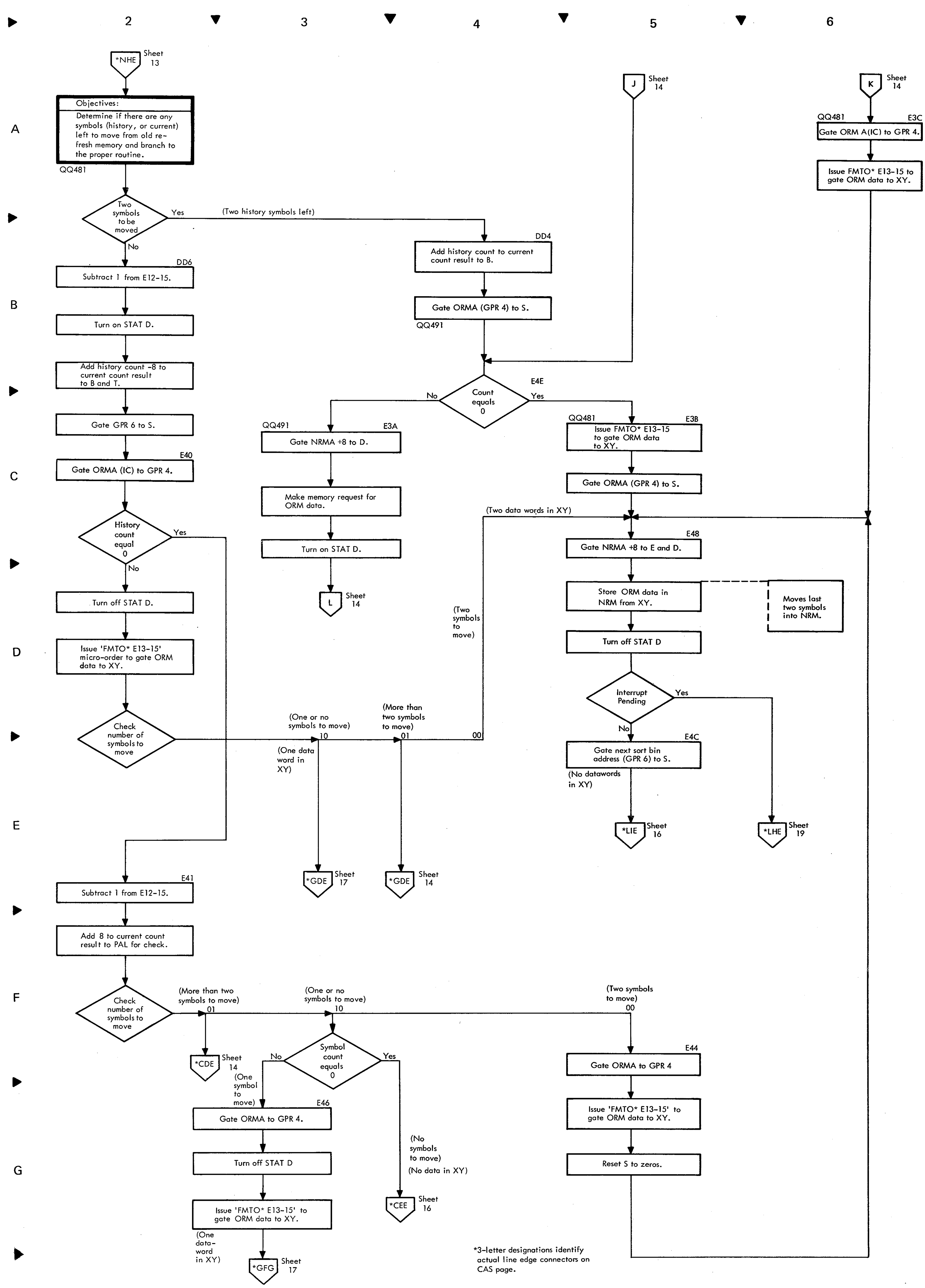


Diagram 5-902. Repack Symbols, RPSB (0F) (Sheet 15 of 21)

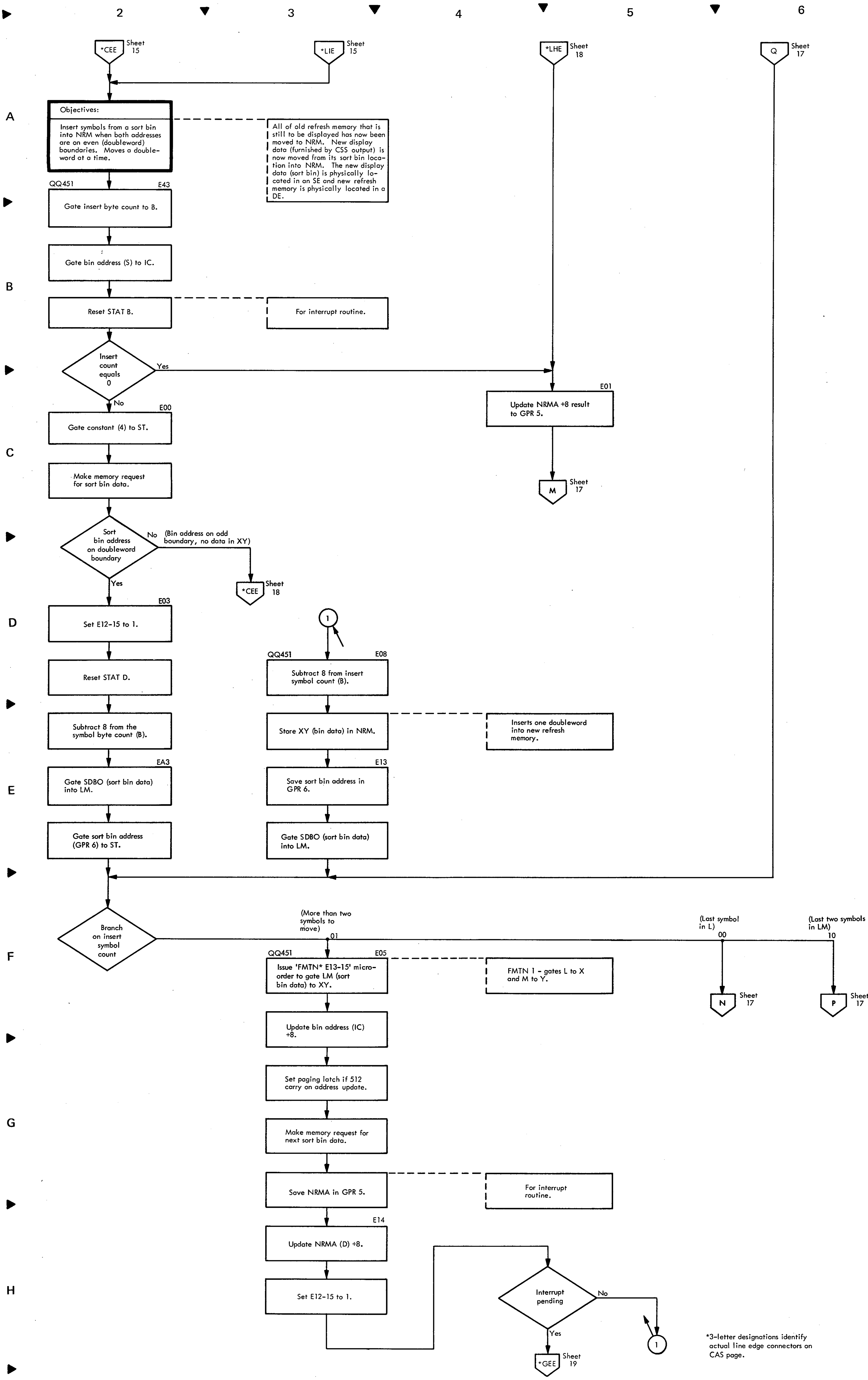


Diagram 5-902. Repack Symbols, RPSB (0F) (Sheet 16 of 21)

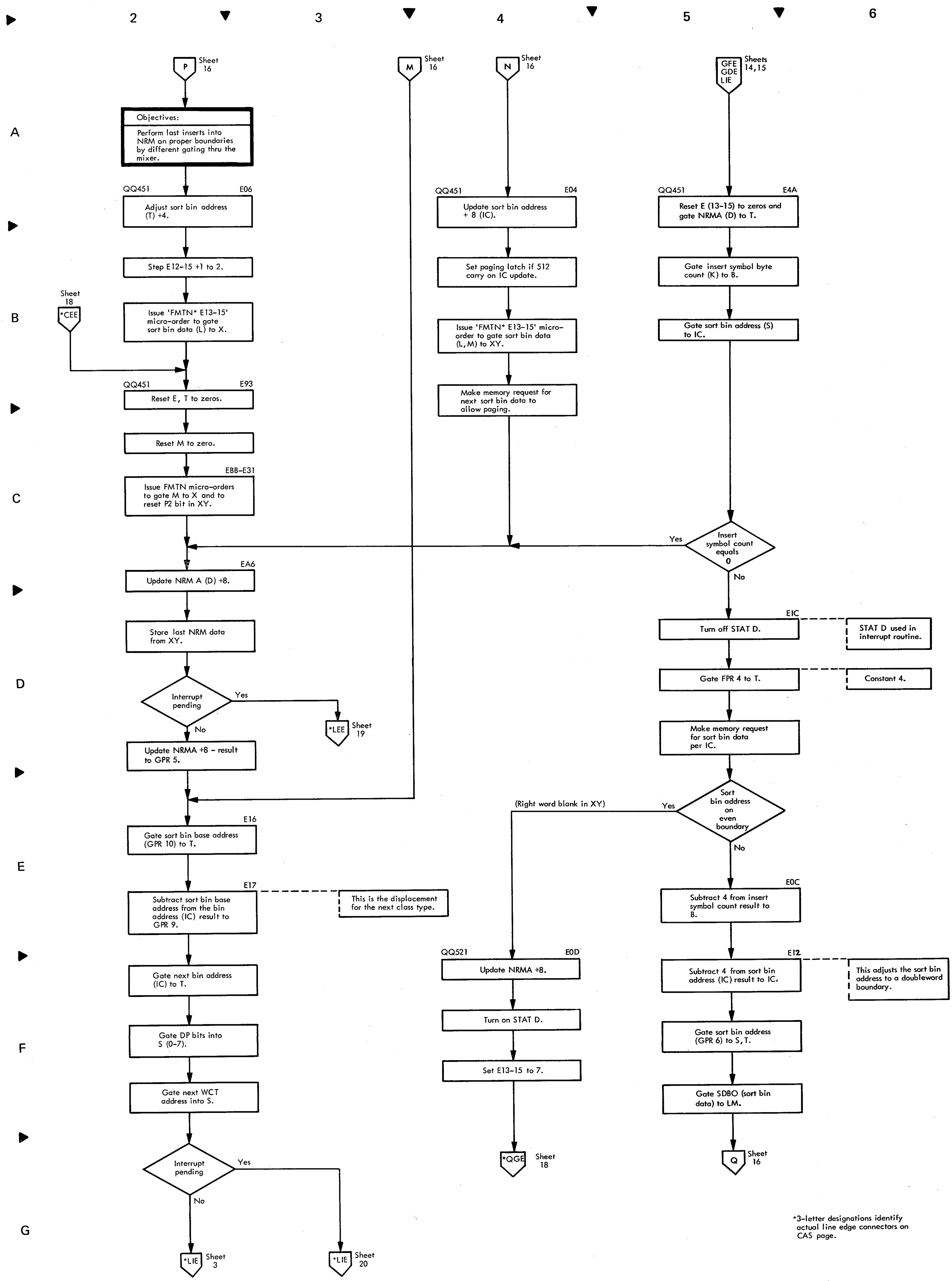


Diagram 5-902. Repack Symbols, RPSB (0F) (Sheet 17 of 21)

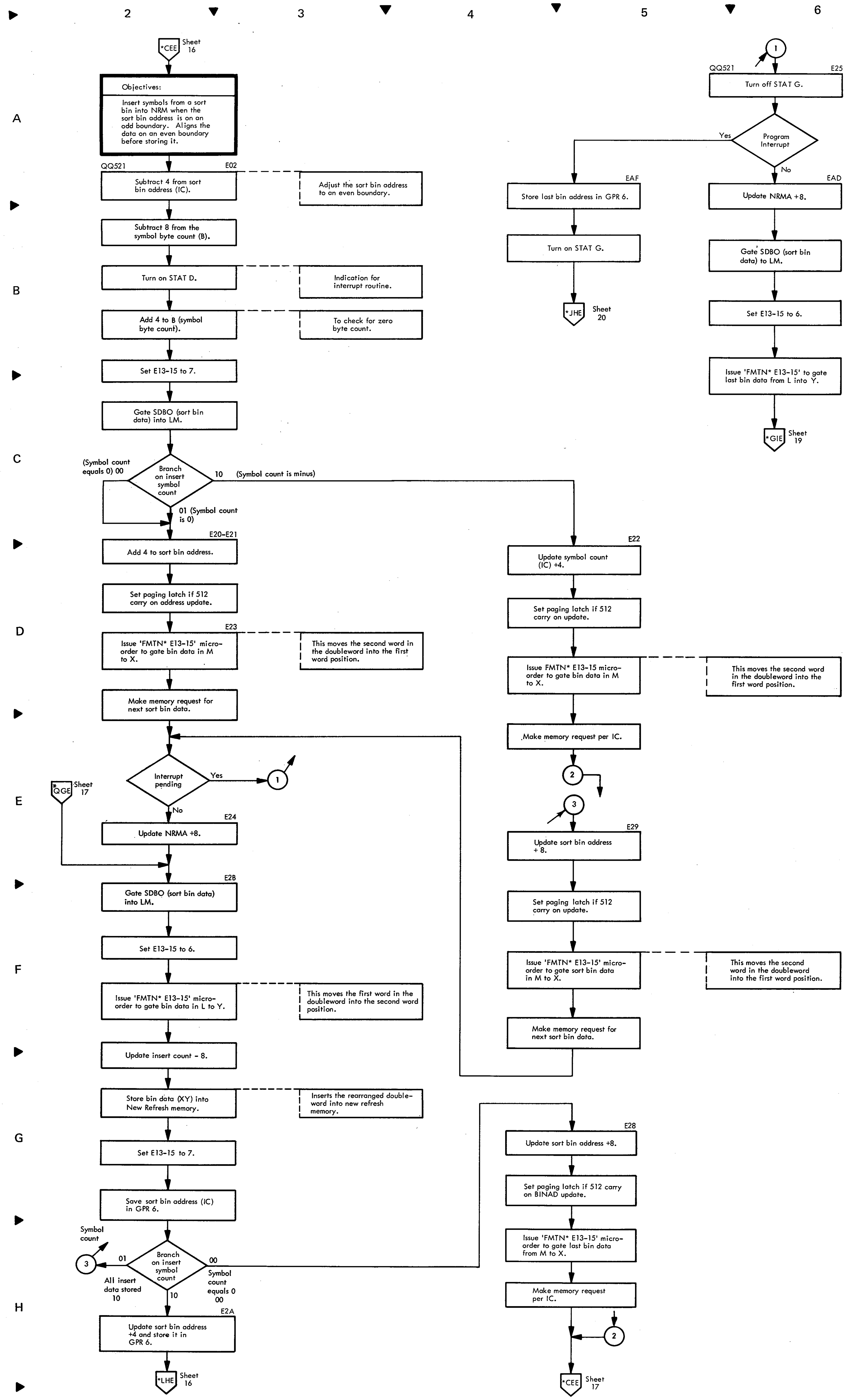


Diagram 5-902. Repack Symbols, RPSB (0F) (Sheet 18 of 21)

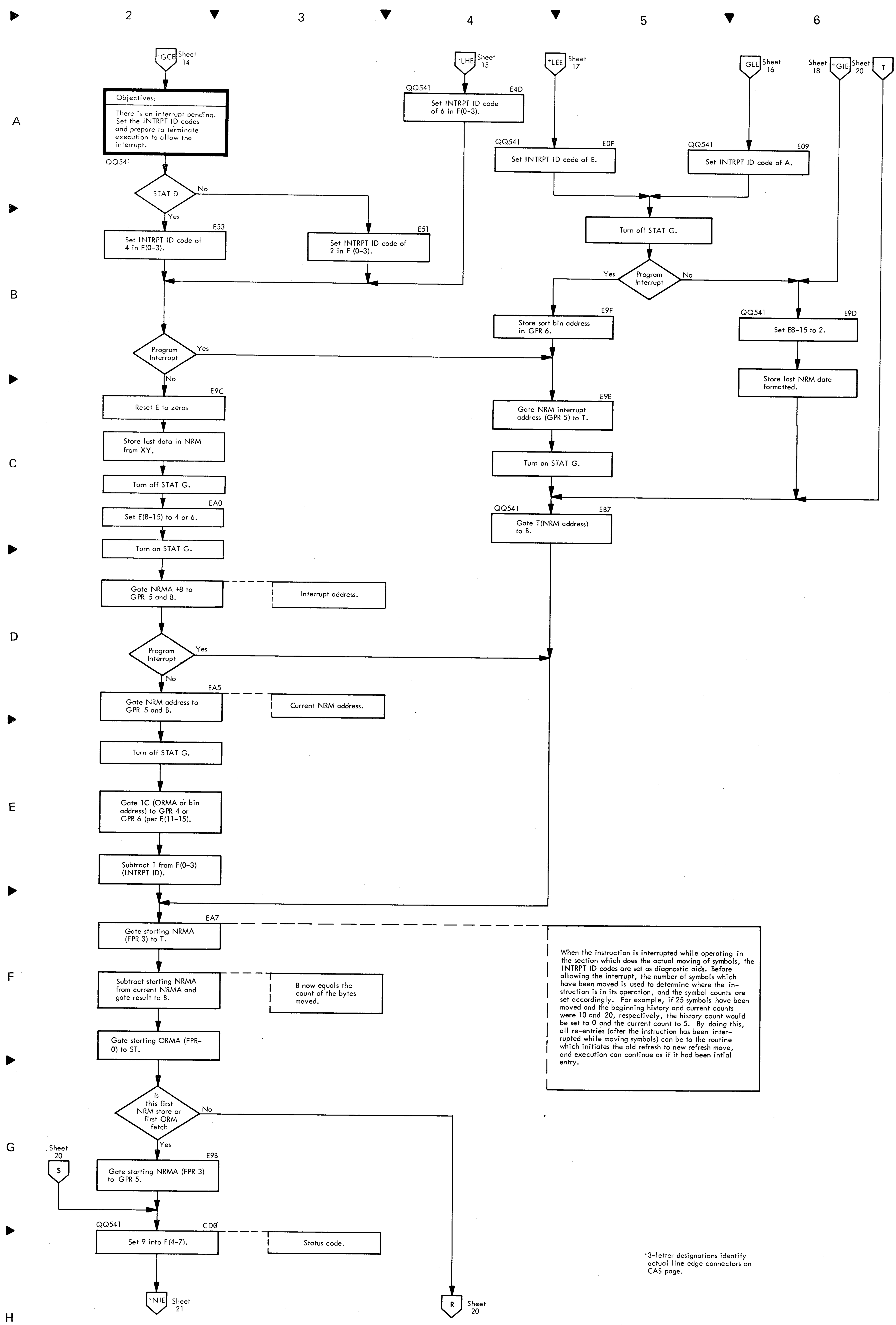


Diagram 5-902. Repack Symbols, RPSB (0F) (Sheet 19 of 21)

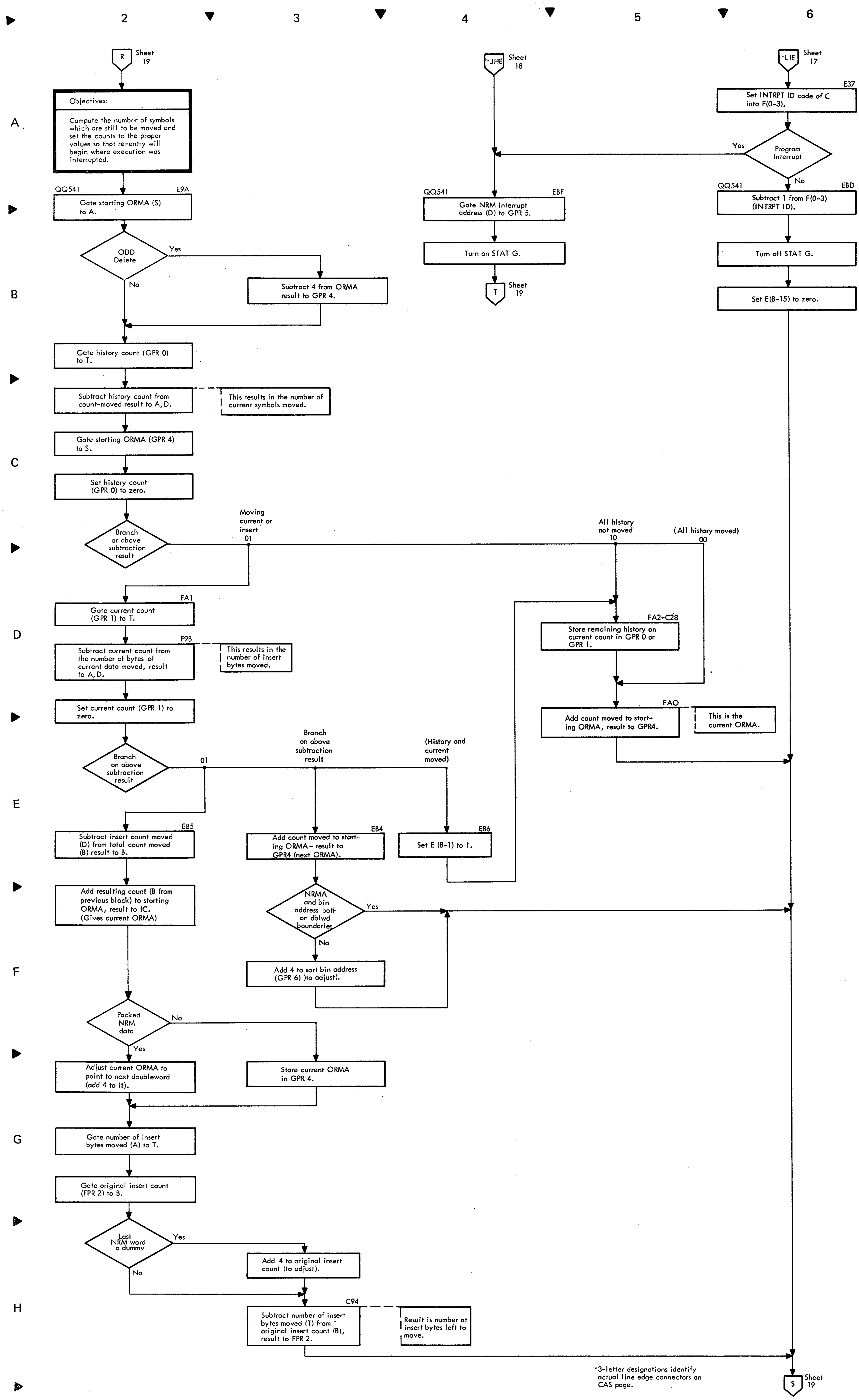


Diagram 5-902. Repack Symbols, RPSB (0F) (Sheet 20 of 21)

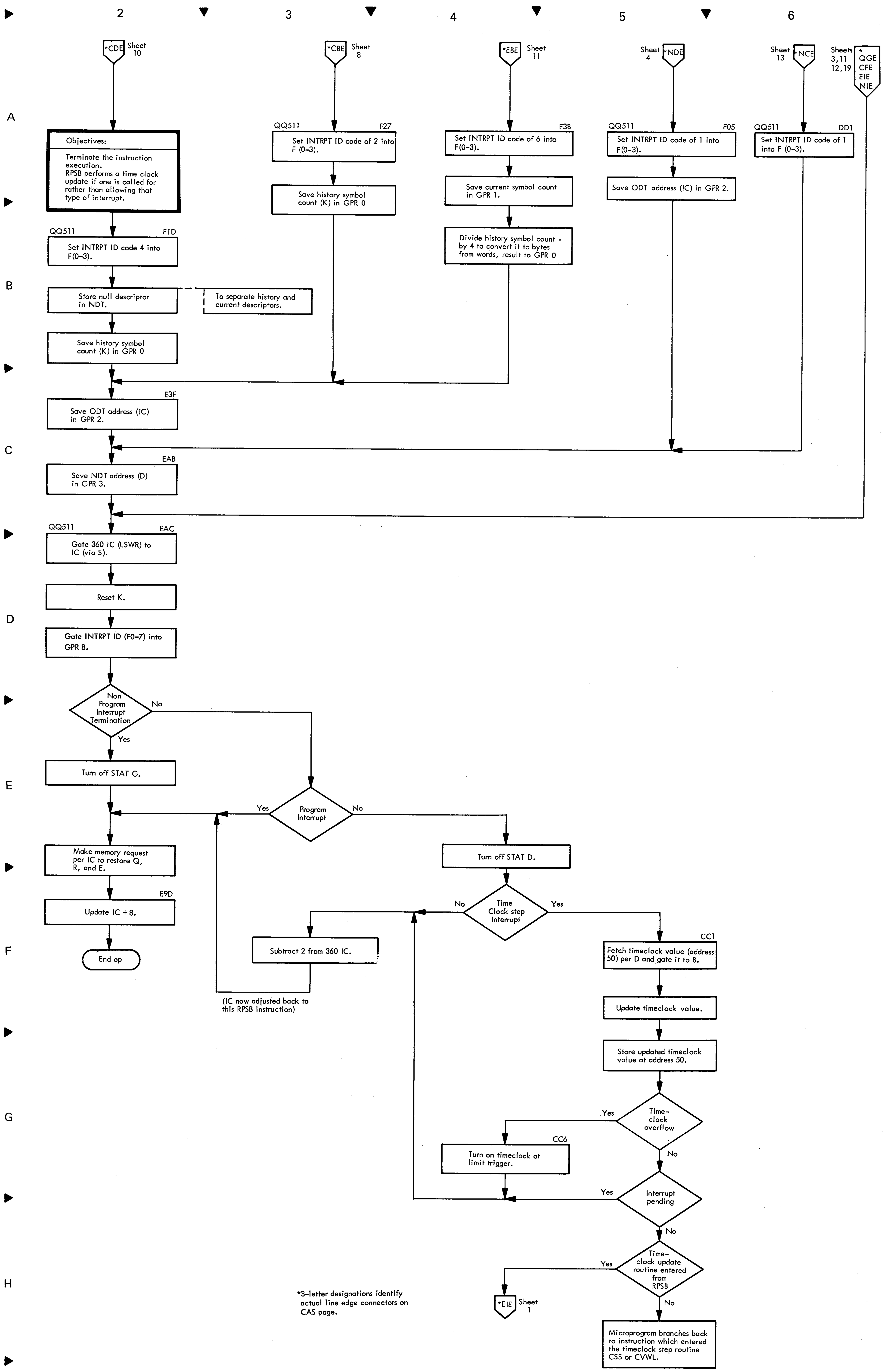
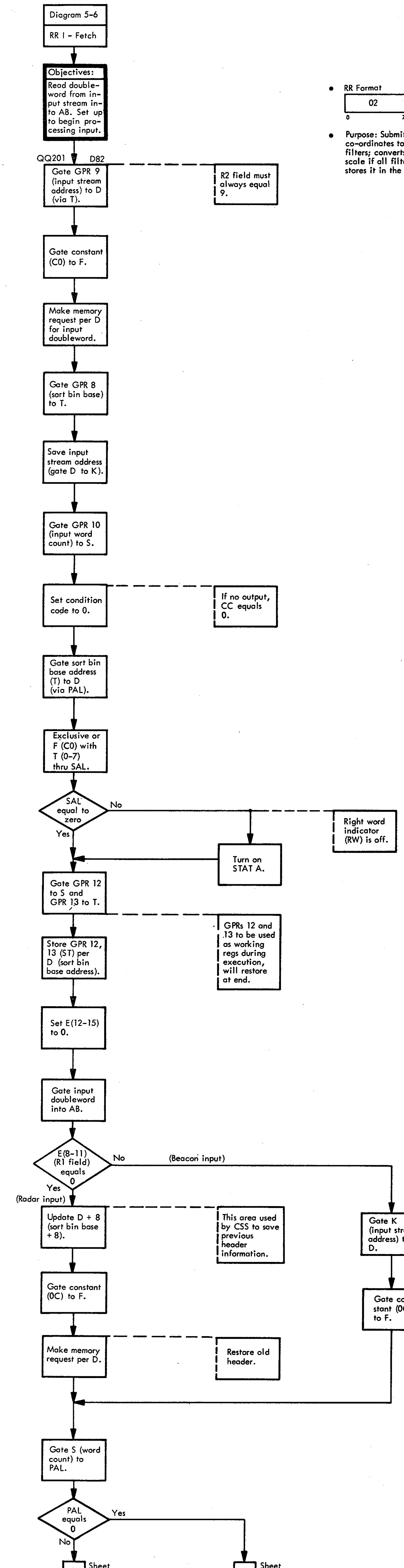


Diagram 5-902. Repack Symbols, RPSB (0F) (Sheet 21 of 21)

2 3 4 5 6

A B C D E F G H



- RR Format

02	R1	R2
0 7	8 11	12 15

- Purpose: Submits an input stream of either Radar or Beacon co-ordinates to one PVD's geographic and sterile area filters; converts the co-ordinates from system scale to PVD scale if all filters are passed; assembles an output word and stores it in the proper sort bin location.

Diagram 5-903. Convert and Sort Symbols, CSS (02) (Sheet 1 of 10)

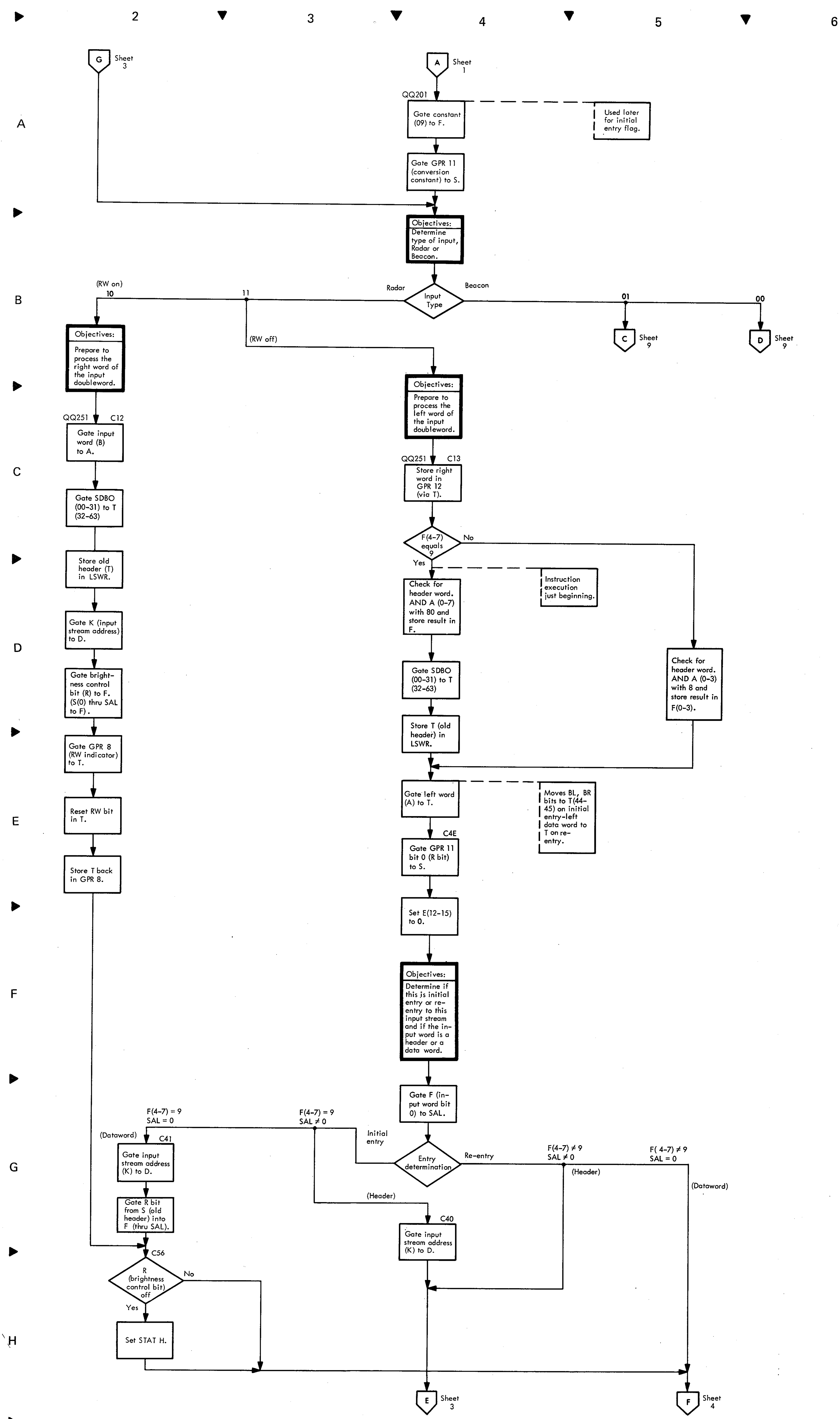


Diagram 5-903. Convert and Sort Symbols, CSS (02) (Sheet 2 of 10)

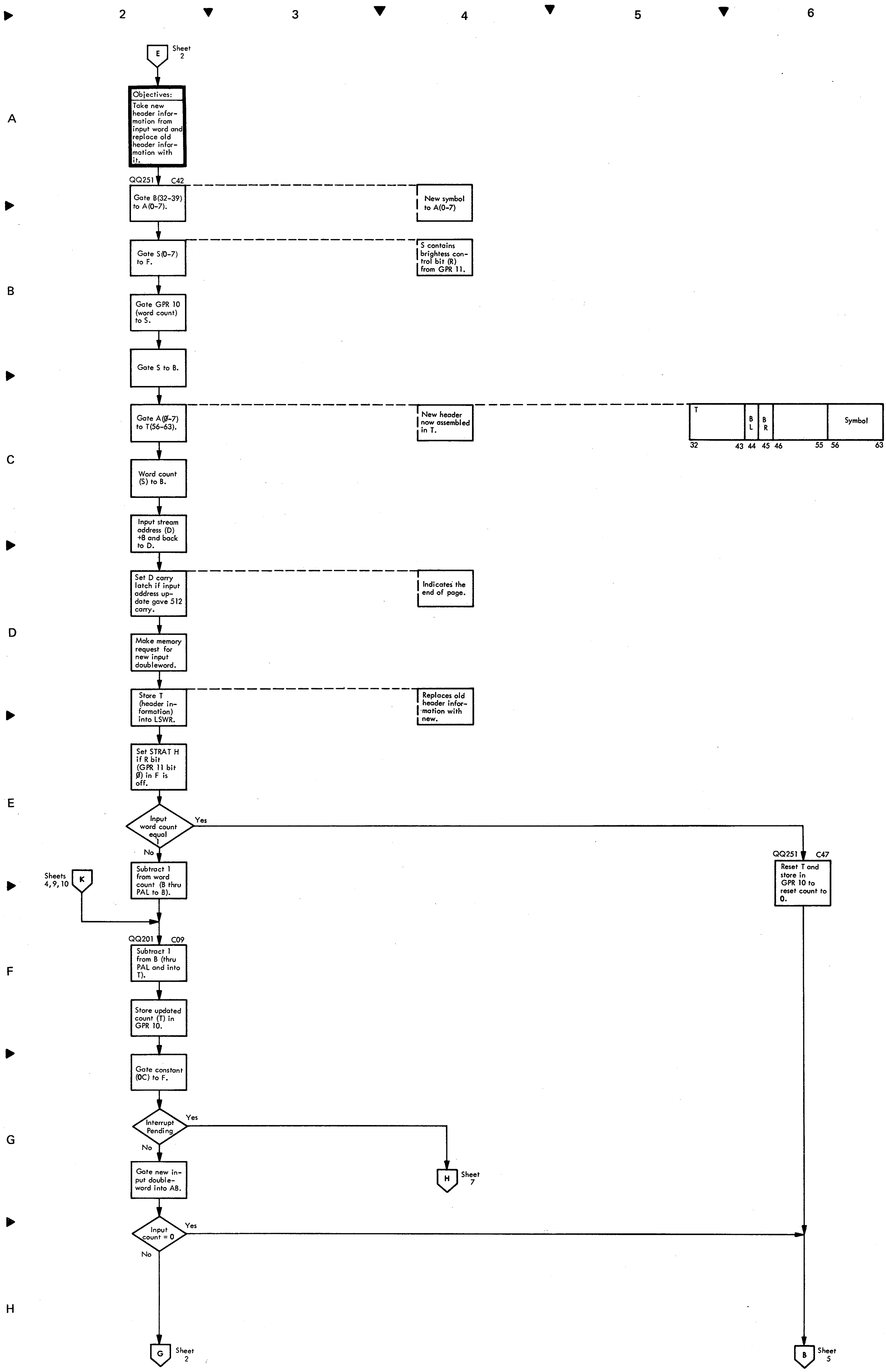


Diagram 5-903. Convert and Sort Symbols, CSS (02) (Sheet 3 of 10)

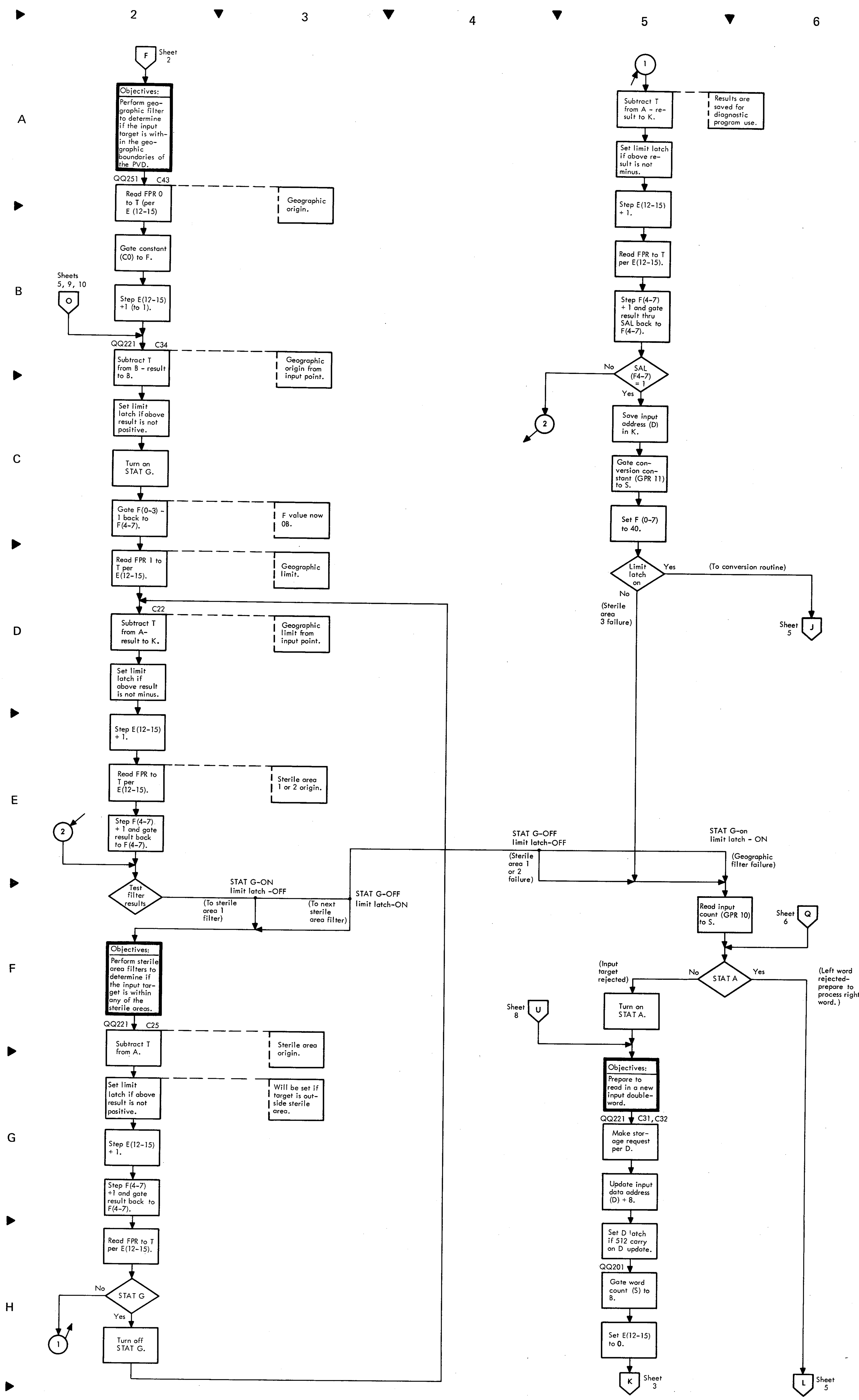


Diagram 5-903. Convert and Sort Symbols, CSS (02) (Sheet 4 of 10)

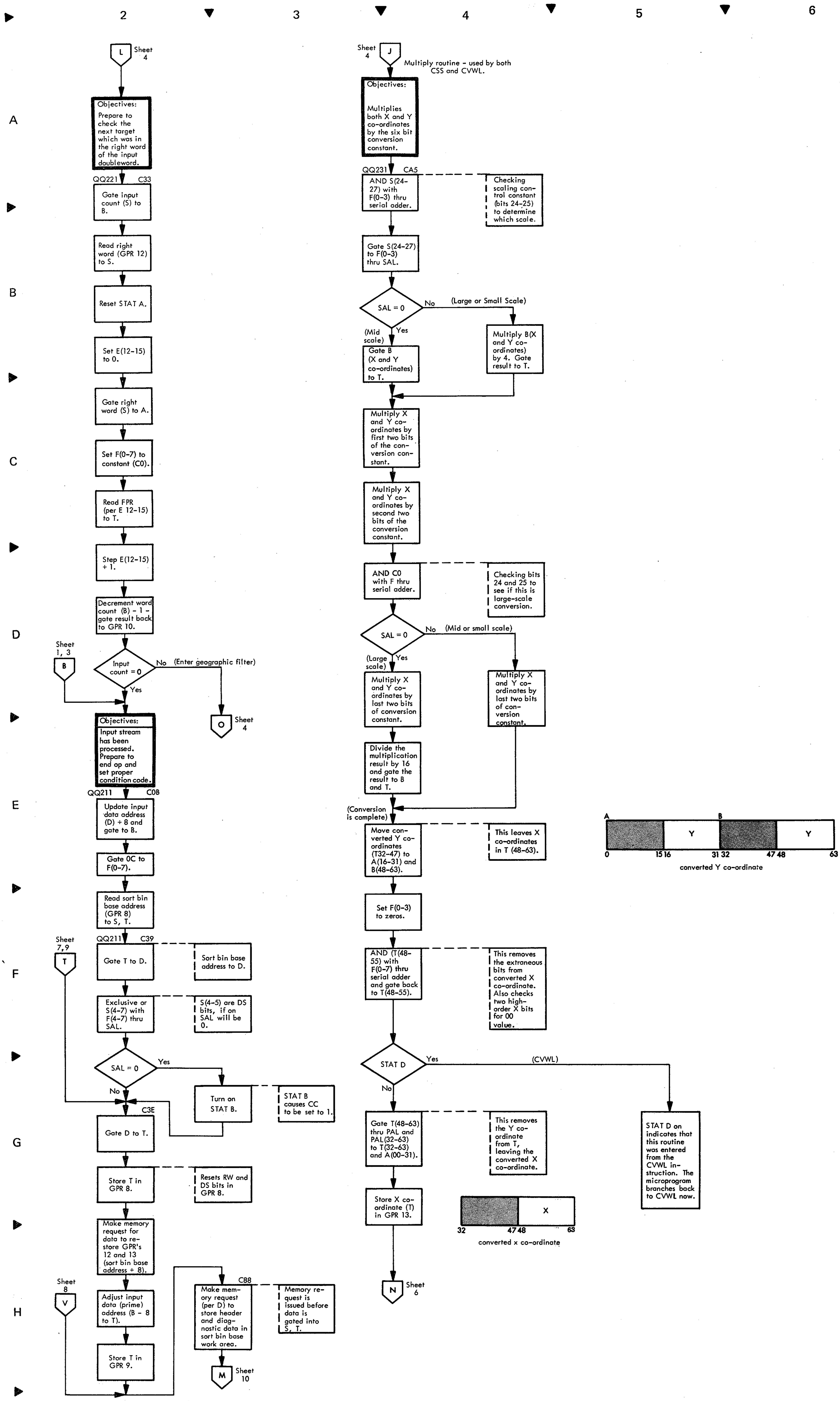


Diagram 5-903. Convert and Sort Symbols, CSS (02) (Sheet 5 of 10)

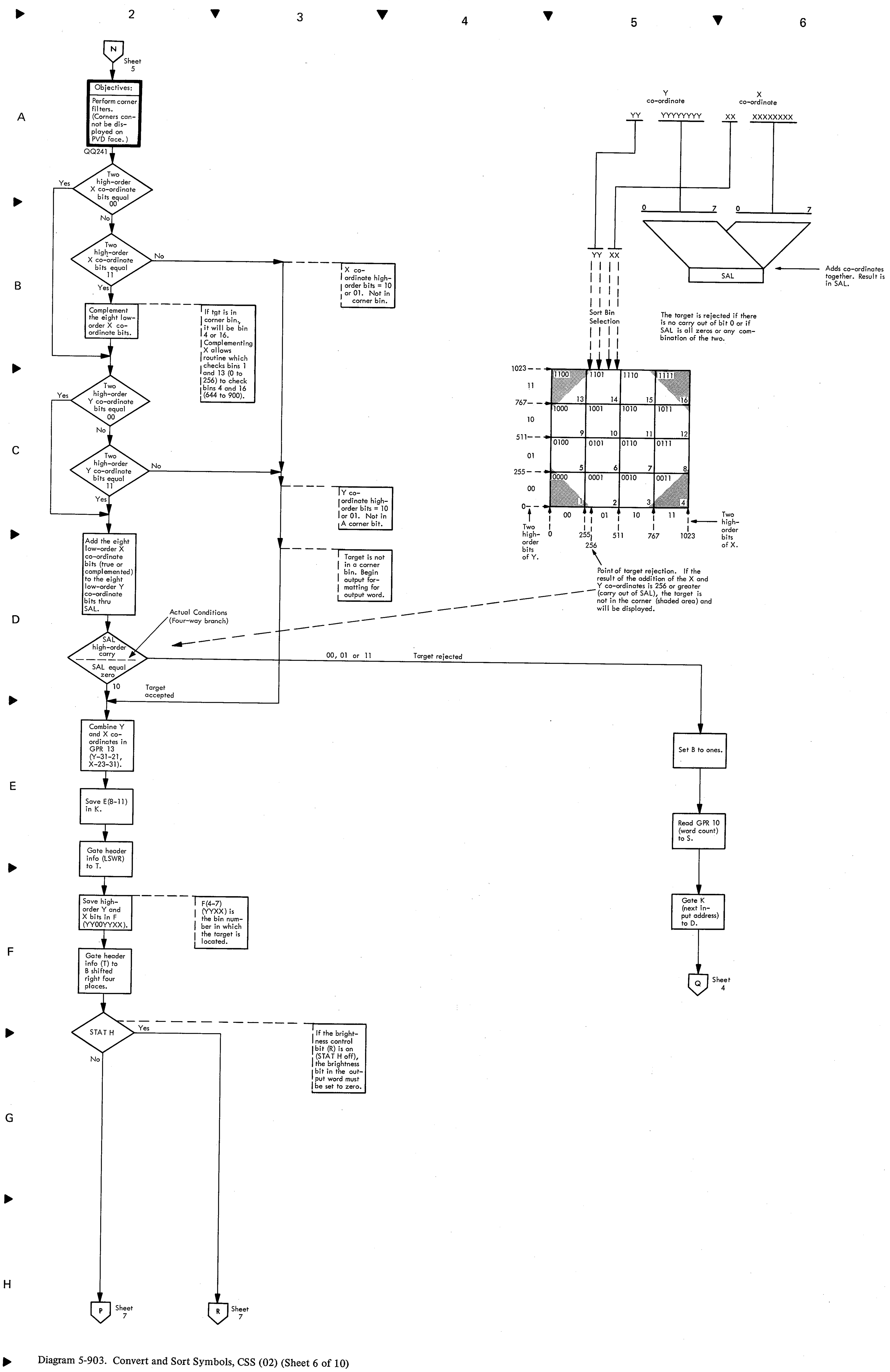


Diagram 5-903. Convert and Sort Symbols, CSS (02) (Sheet 6 of 10)

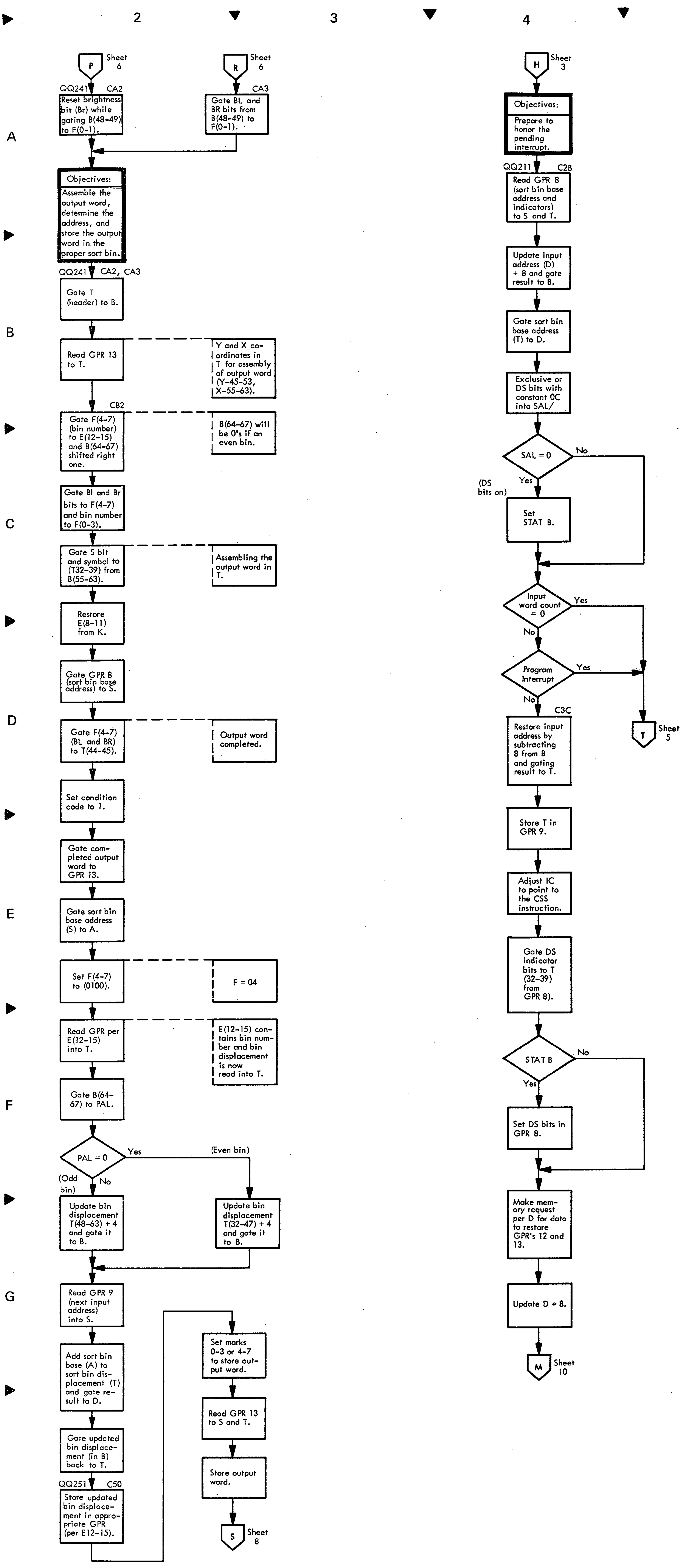


Diagram 5-903. Convert and Sort Symbols, CSS (02) (Sheet 7 of 10)

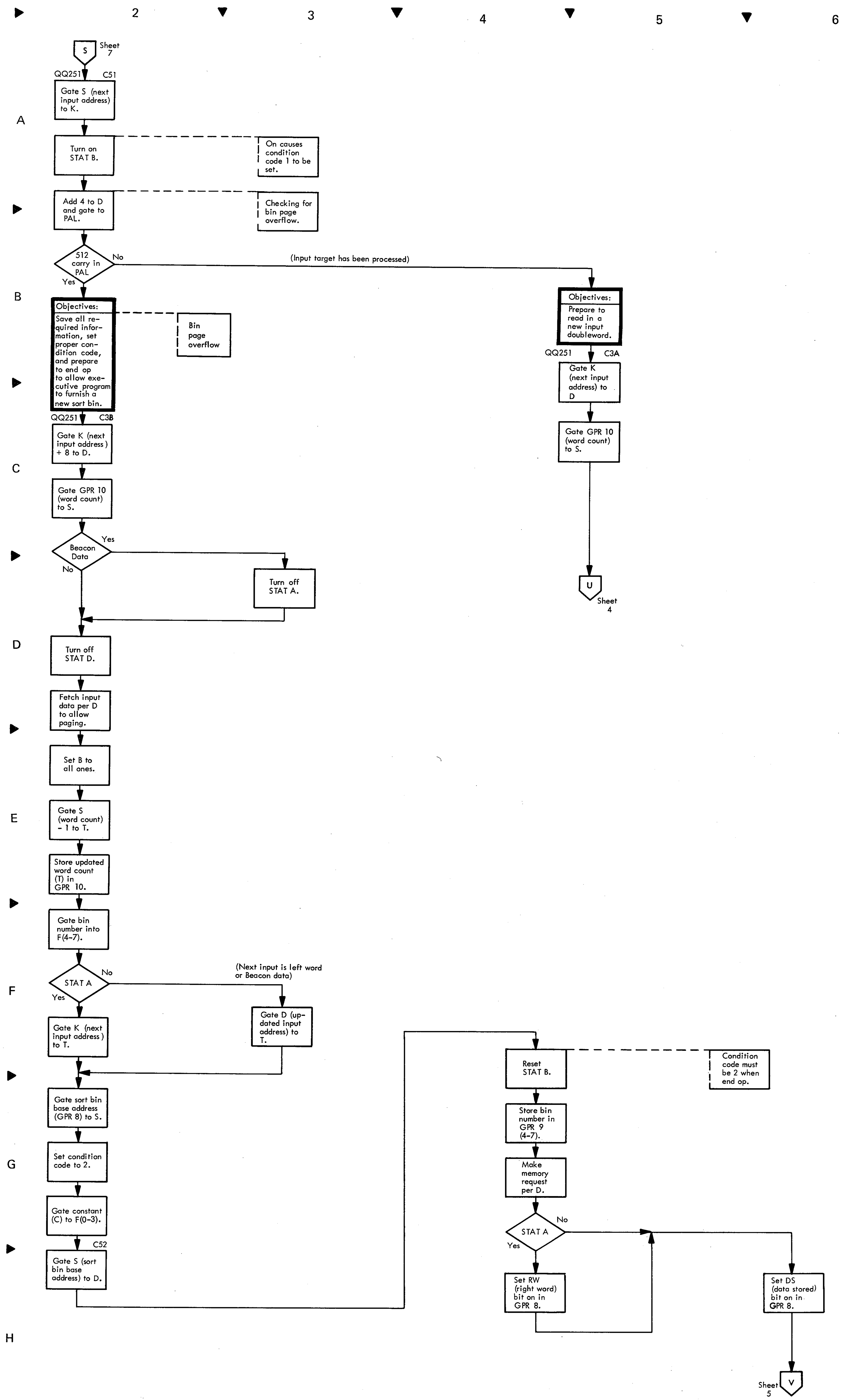


Diagram 5-903. Convert and Sort Symbols, CSS (02) (Sheet 8 of 10)

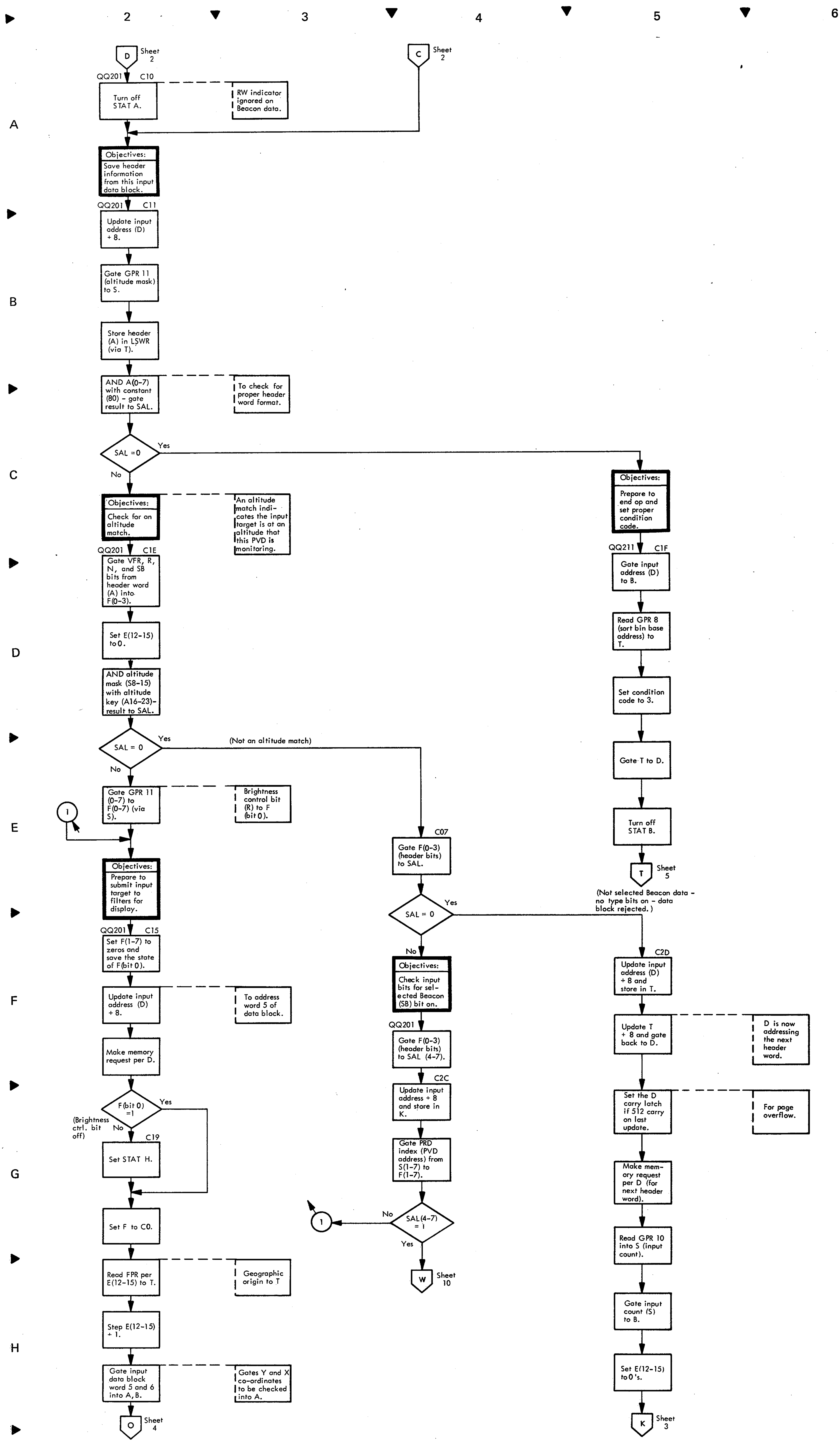


Diagram 5-903. Convert and Sort Symbols, CSS (02) (Sheet 9 of 10)

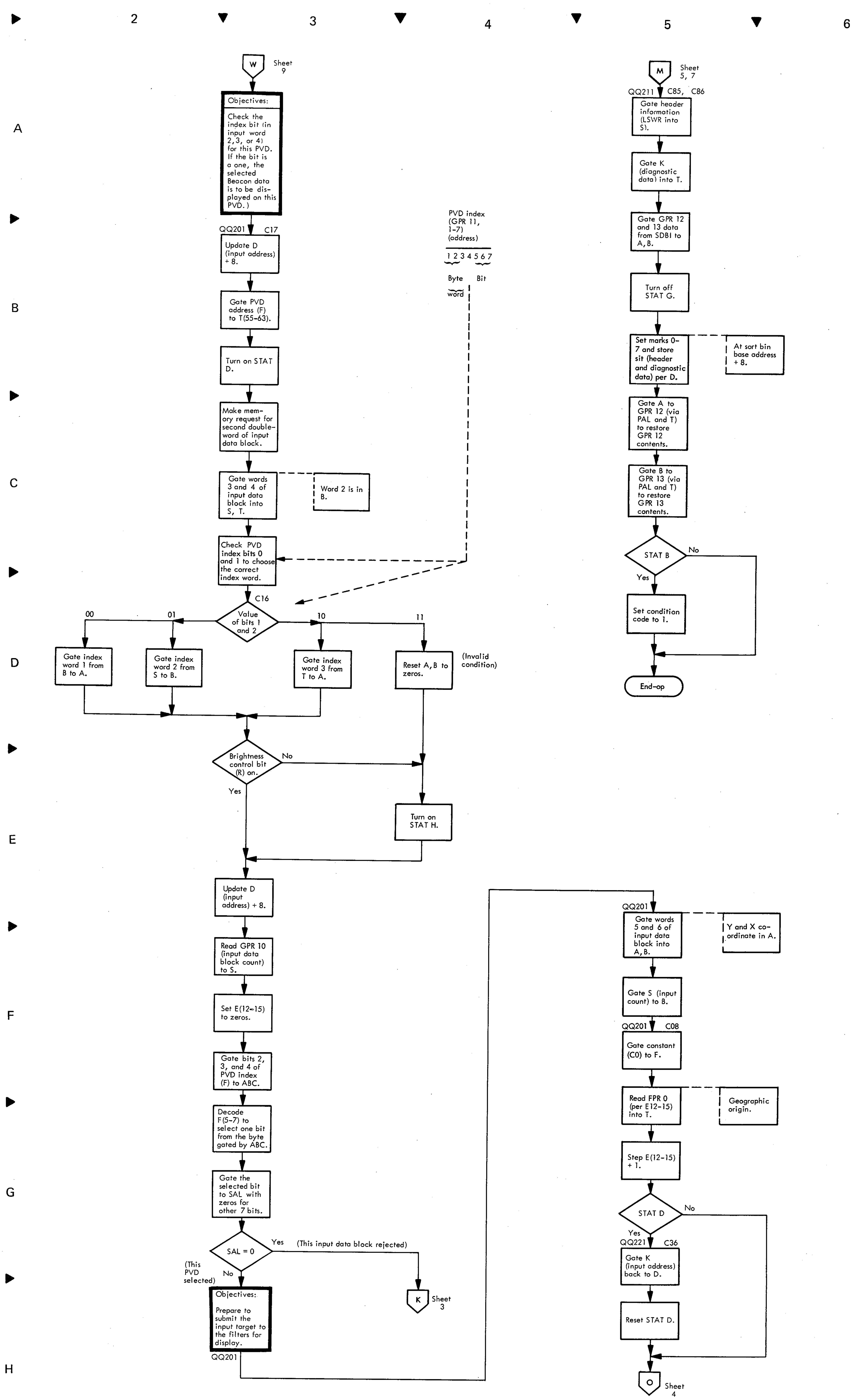


Diagram 5-903. Convert and Sort Symbols, CSS (02) (Sheet 10 of 10)

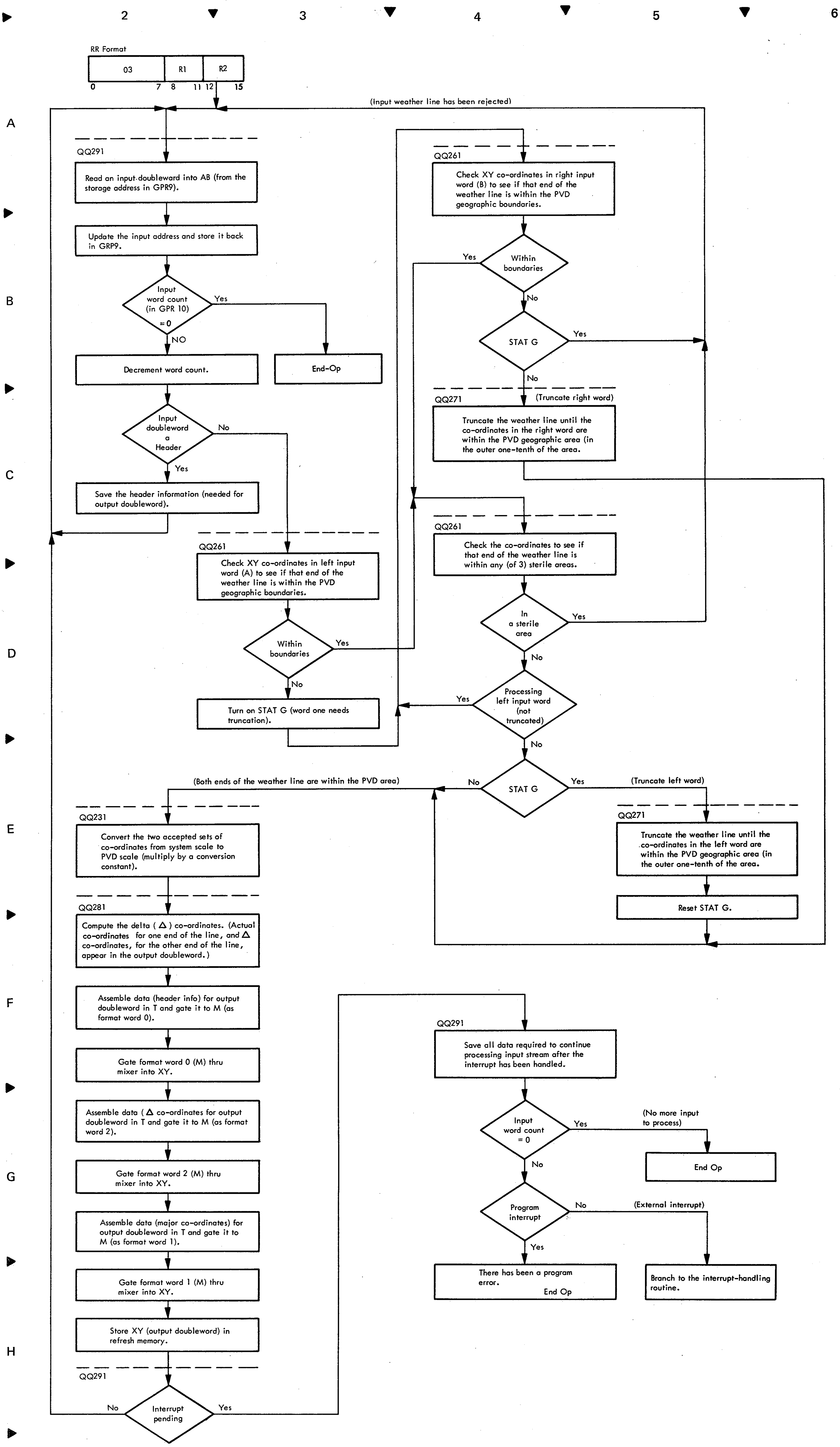
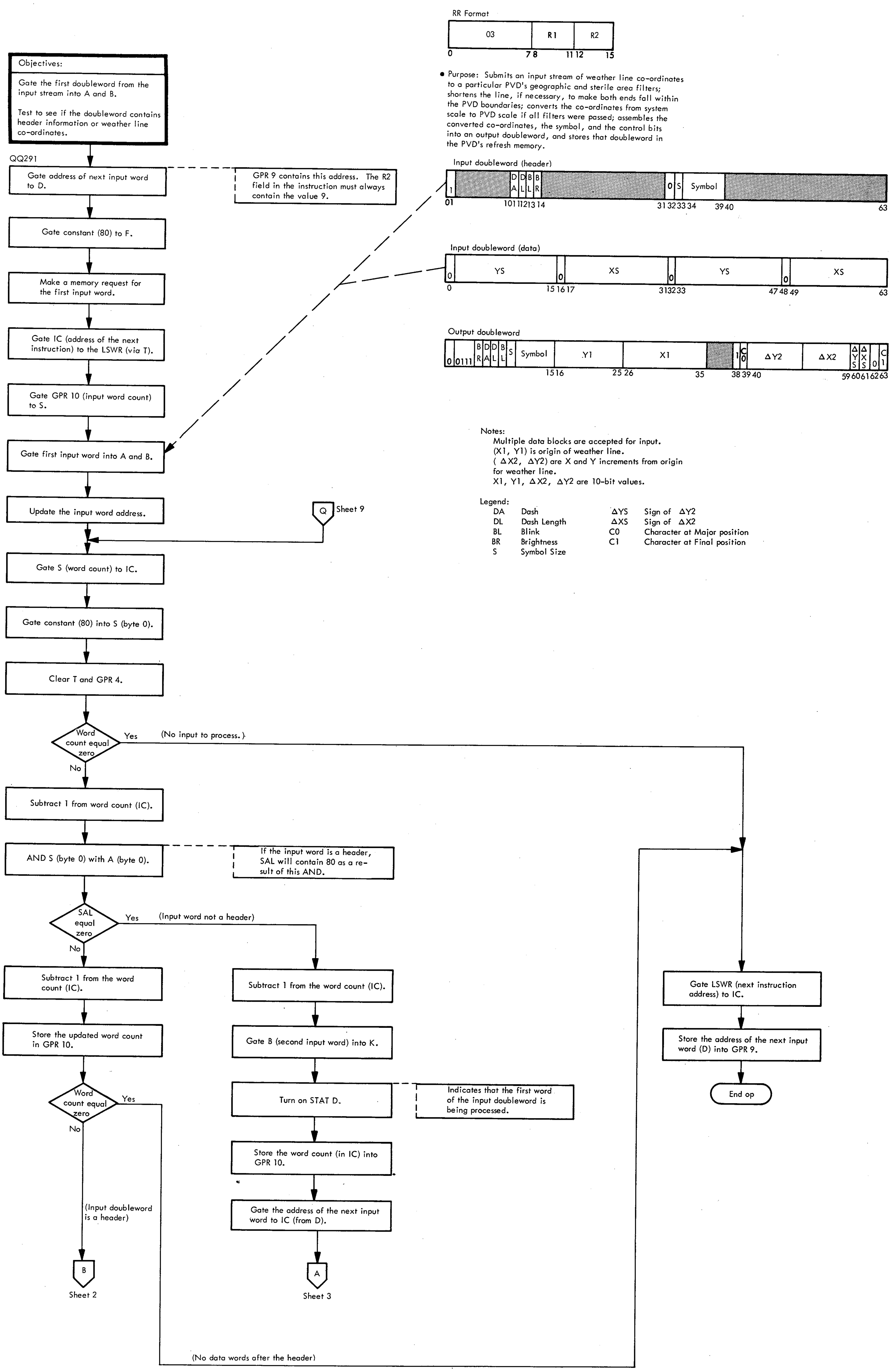


Diagram 5-904. Convert Weather Lines, Simplified Flowchart

2 3 4 5 6



- Purpose: Submits an input stream of weather line co-ordinates to a particular PVD's geographic and sterile area filters; shortens the line, if necessary, to make both ends fall within the PVD boundaries; converts the co-ordinates from system scale to PVD scale if all filters were passed; assembles the converted co-ordinates, the symbol, and the control bits into an output doubleword, and stores that doubleword in the PVD's refresh memory.

Notes:
Multiple data blocks are accepted for input.
(X1, Y1) is origin of weather line.
(ΔX2, ΔY2) are X and Y increments from origin for weather line.
X1, Y1, ΔX2, ΔY2 are 10-bit values.

Legend:

DA	Dash	ΔYS	Sign of ΔY2
DL	Dash Length	ΔXS	Sign of ΔX2
BL	Blink	C0	Character at Major position
BR	Brightness	C1	Character at Final position
S	Symbol Size		

A B C D E F G H

Diagram 5-905. Convert Weather Lines, CVWL (03) (Sheet 1 of 9)

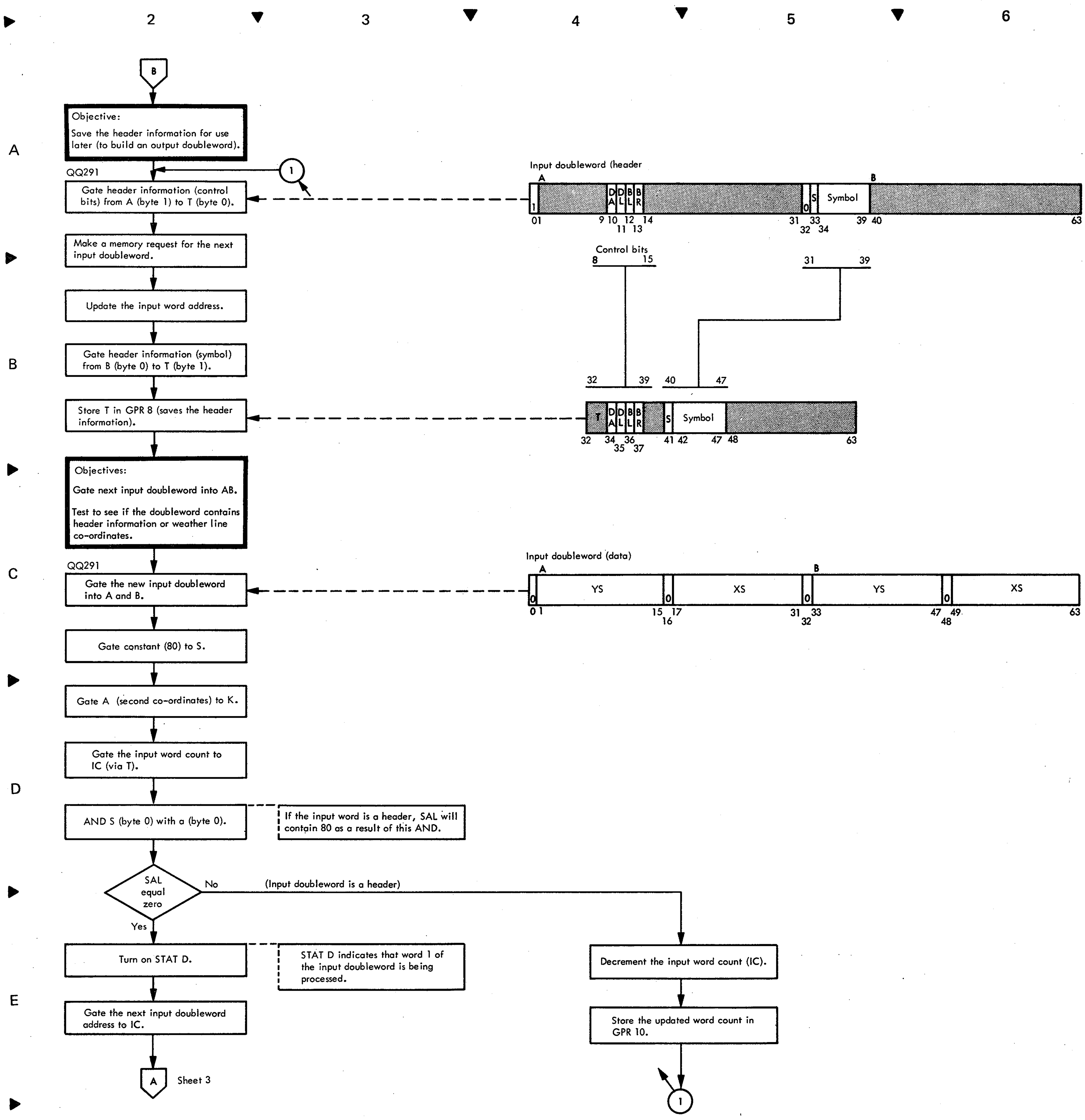


Diagram 5-905. Convert Weather Lines, CVWL (03) (Sheet 2 of 9)

2 3 4 5 6

A

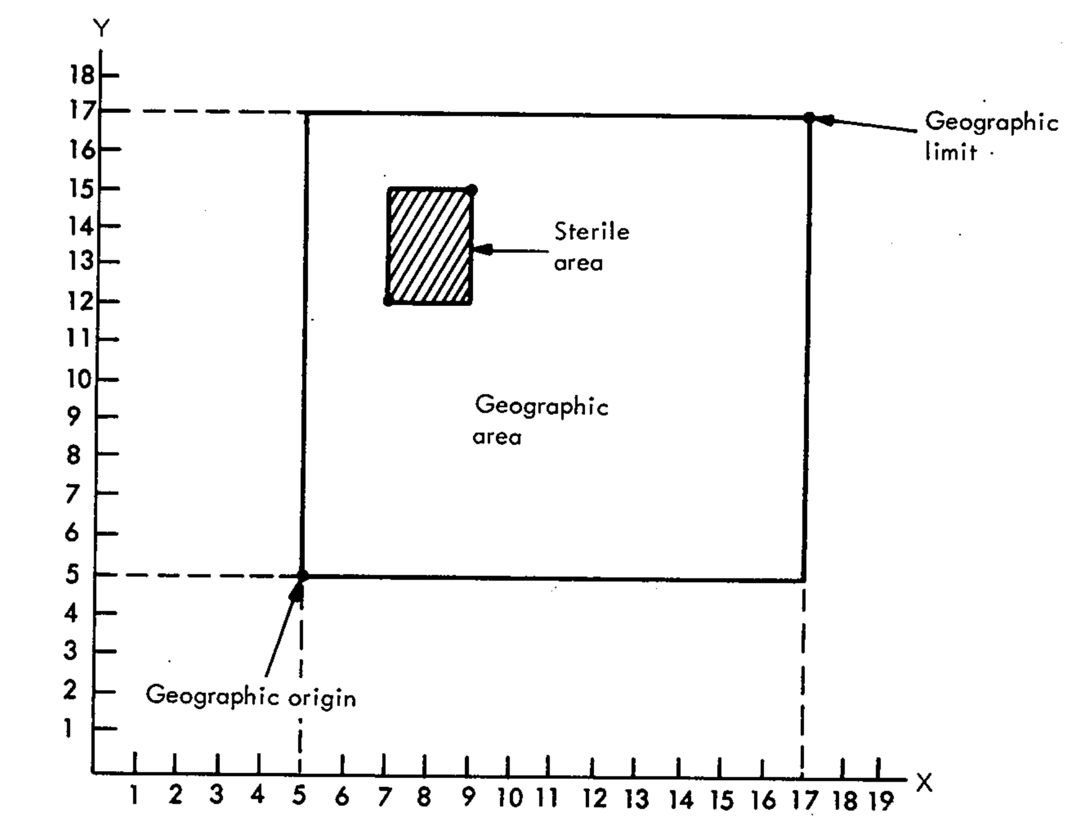
1

Objectives:
Perform the geographic filter (determine if the input point is within the PVD geographic area).
Perform the first and second sterile area filters (determine if the input point is within a sterile area on the PVD screen).

QQ261

Gate the PVD geographic origin co-ordinates (in FPR0) to T.

Two sets of co-ordinates are used to describe the geographic boundaries of a PVD. Two sets are also used to describe each sterile area within a PVD's geographic boundaries. The set of co-ordinates which describe the lowest left point of an area will be referred to as the geographic origin. The other set, describing the uppermost right point of an area, will be called the geographic limit.



Subtract T (origin) from the first input co-ordinates (in A) and put the result in B.

If the co-ordinates do not require truncation, this result becomes the major or secondary position.

Set the Limit Latch if the subtraction result is not positive.

Gate the PVD geographic Limit (FPR1) to T.

Subtract the geographic Limit (in T) from the first co-ordinates (in A).

Set the Limit Latch if the subtraction result is not minus.

Limit latch on — Yes (This point failed geographic filter) — No (Sterile area filter)

Gate the first sterile area origin (FPR 2) to T.

If either end of a weather line (original or truncated point) is within a sterile area, the line is rejected.

Subtract T (origin) from the first input co-ordinates (in A).

Set the limit latch if the subtraction result is not positive.

Gate the first sterile area limit (FPR 3) to T.

Subtract T (Limit) from the first input co-ordinates (in A).

Set the Limit latch if the subtraction result is not minus.

Limit latch on — Yes (This point failed the first sterile area filter.) — No (Sterile area filter)

E Sheet 4

Gate the second sterile area origin (FPR 4) to T.

Subtract T (origin) from the first input co-ordinates (in A).

Set the Limit latch if the subtraction result is not positive.

Gate the second sterile area Limit (FPR 5) to T.

Subtract T (Limit) from the input co-ordinates (in A).

Set the Limit latch if the subtraction result is not minus.

F Sheet 4

Set 2 into GPR 4.

Indicates to later routine that truncation has been performed.

STATD — No (Word two is being processed and has failed the geographic filter.) — Yes (Input word 1 is being processed.)

D Sheet 4

Objectives:
Save the first word values (for possible truncation later).
Set up to gate the second word co-ordinates thru the filters.

QQ261

STAT G indicates that input word 1 failed the geographic filter and requires truncation.

Reset STAT D and turn on STAT G.

Store first input co-ordinates in GRP 0.

Store the first subtraction result (in B) in GPR 1.

Gate K (second word of input doubleword) to A.

1

A B C D E F G H

Diagram 5-905. Convert Weather Lines, CVWL (03) (Sheet 3 of 9)

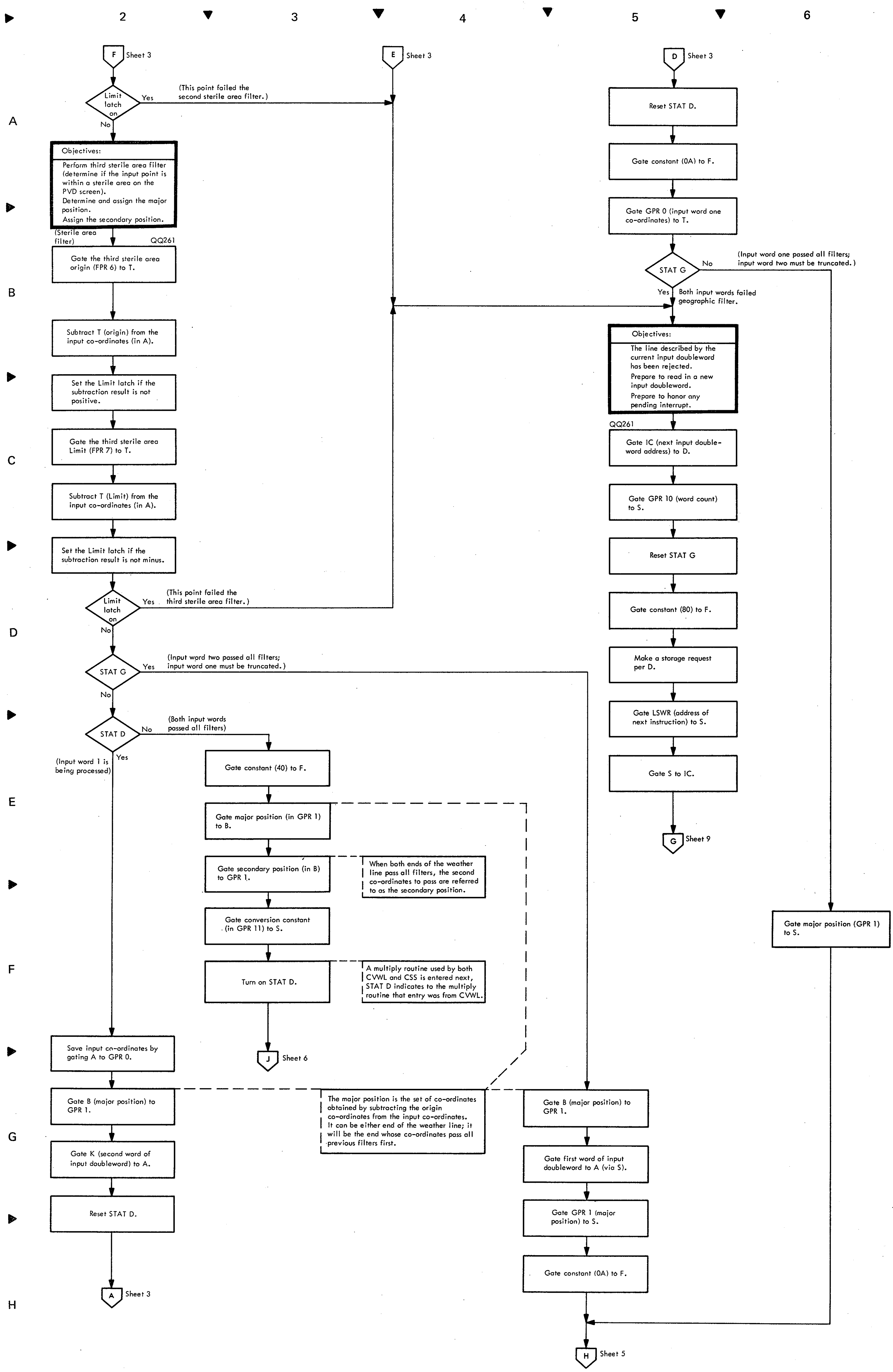


Diagram 5-905. Convert Weather Lines, CVWL (03) (Sheet 4 of 9)

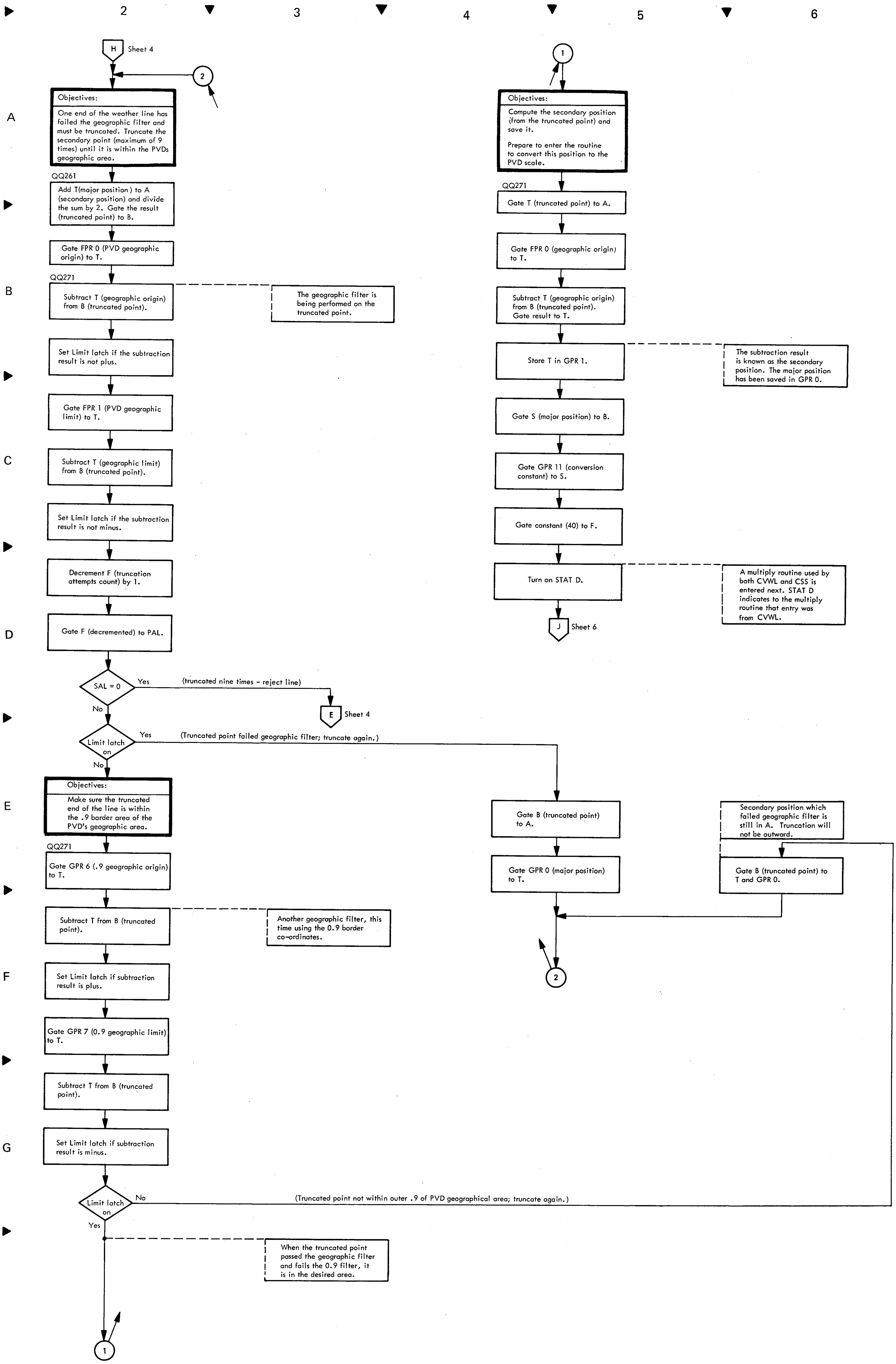


Diagram 5-905. Convert Weather Lines, CVWL (03) (Sheet 5 of 9)

2 3 4 5 6

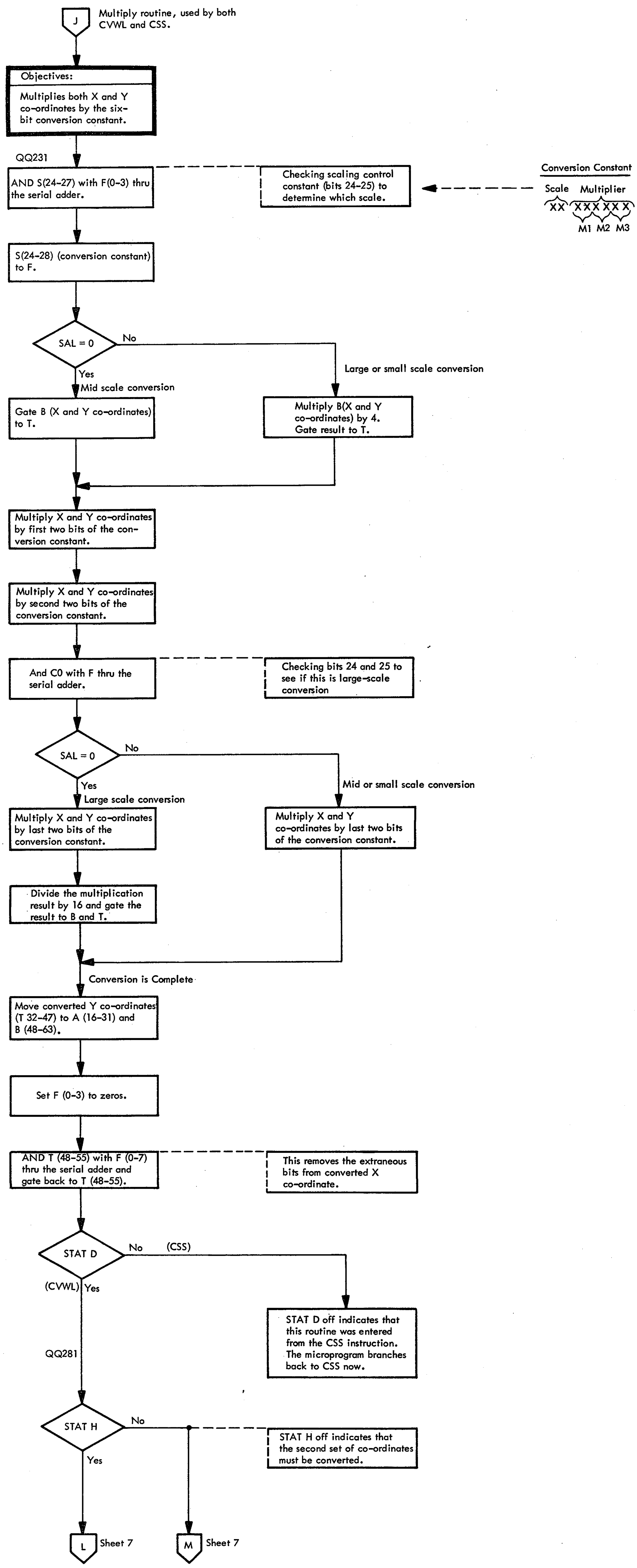


Diagram 5-905. Convert Weather Lines, CVWL (03) (Sheet 6 of 9)

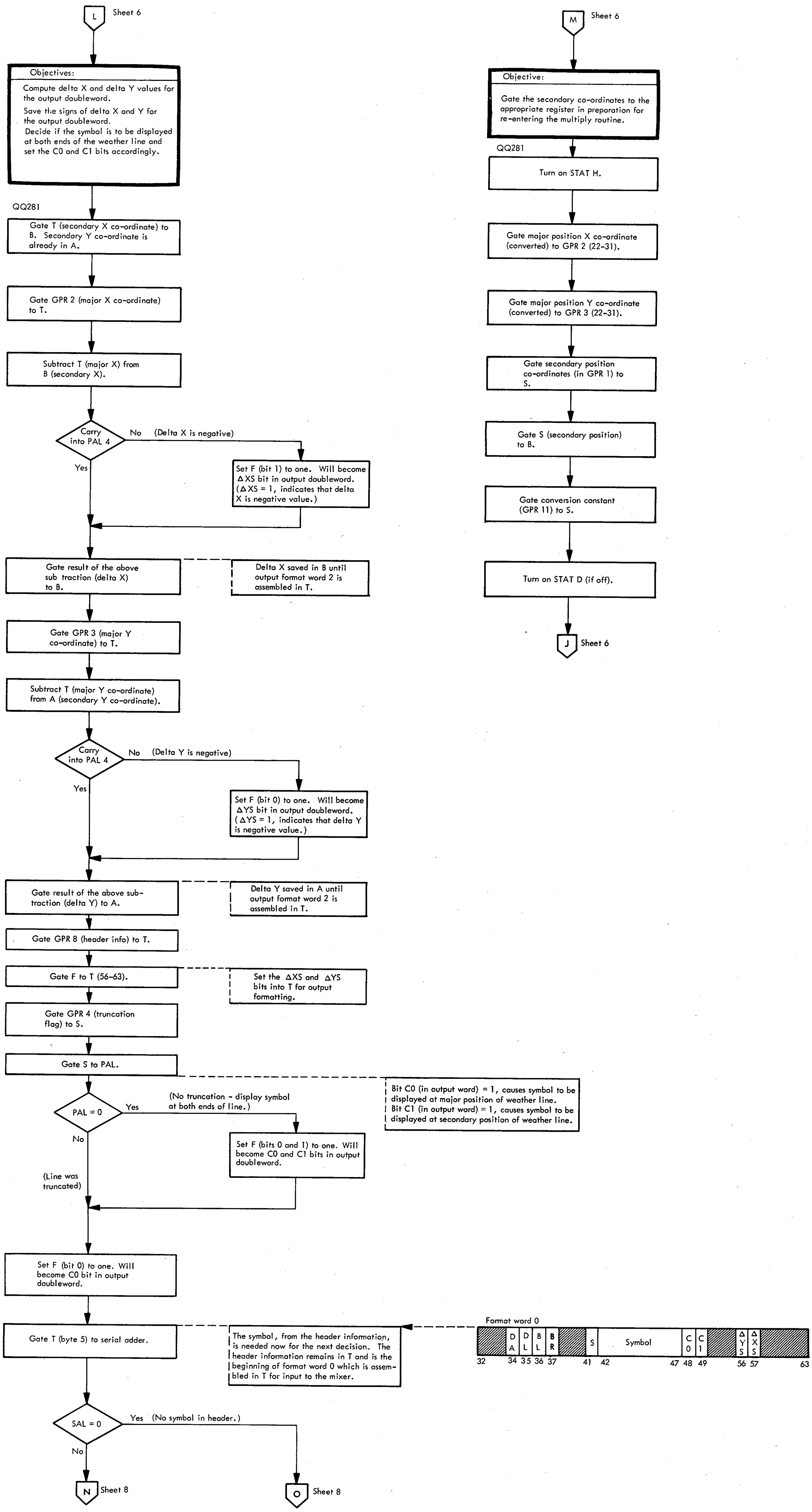


Diagram 5-905. Convert Weather Lines, CVWL (03) (Sheet 7 of 9)

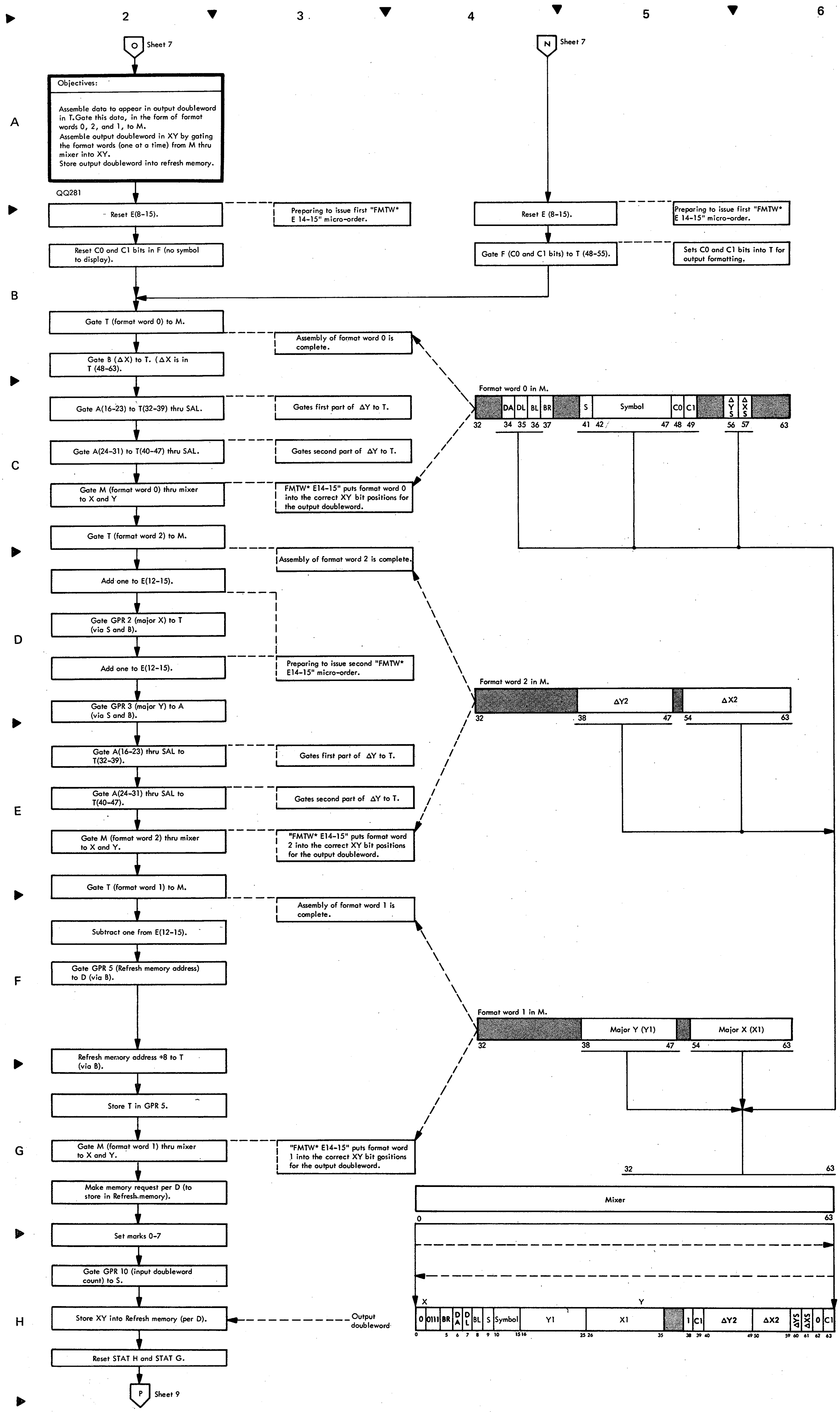


Diagram 5-905. Convert Weather Lines, CVWL (03) (Sheet 8 of 9)

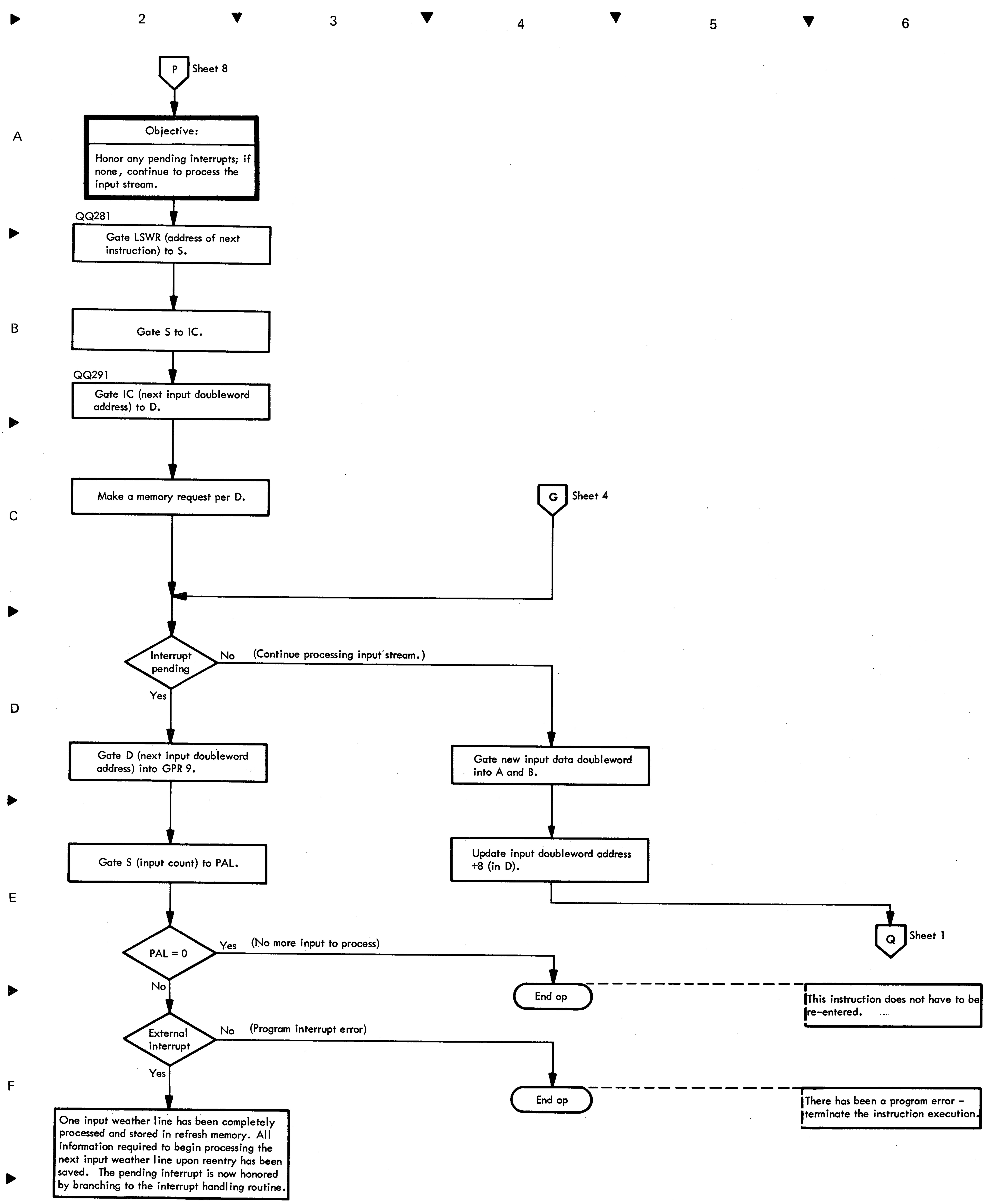


Diagram 5-905. Convert Weather Lines, CVWL (03) (Sheet 9 of 9)

- RX Format

52	R1	X2	B2	D2
0 7	8 11	12 15	16 19	20 31

- Purpose: Moves first operand to third operand location and second operand to first operand location.
- Conditions at beginning of execution:
  1. First operand in S and T.
  2. Second operand address in D.
  3. Request for second operand issued per D.

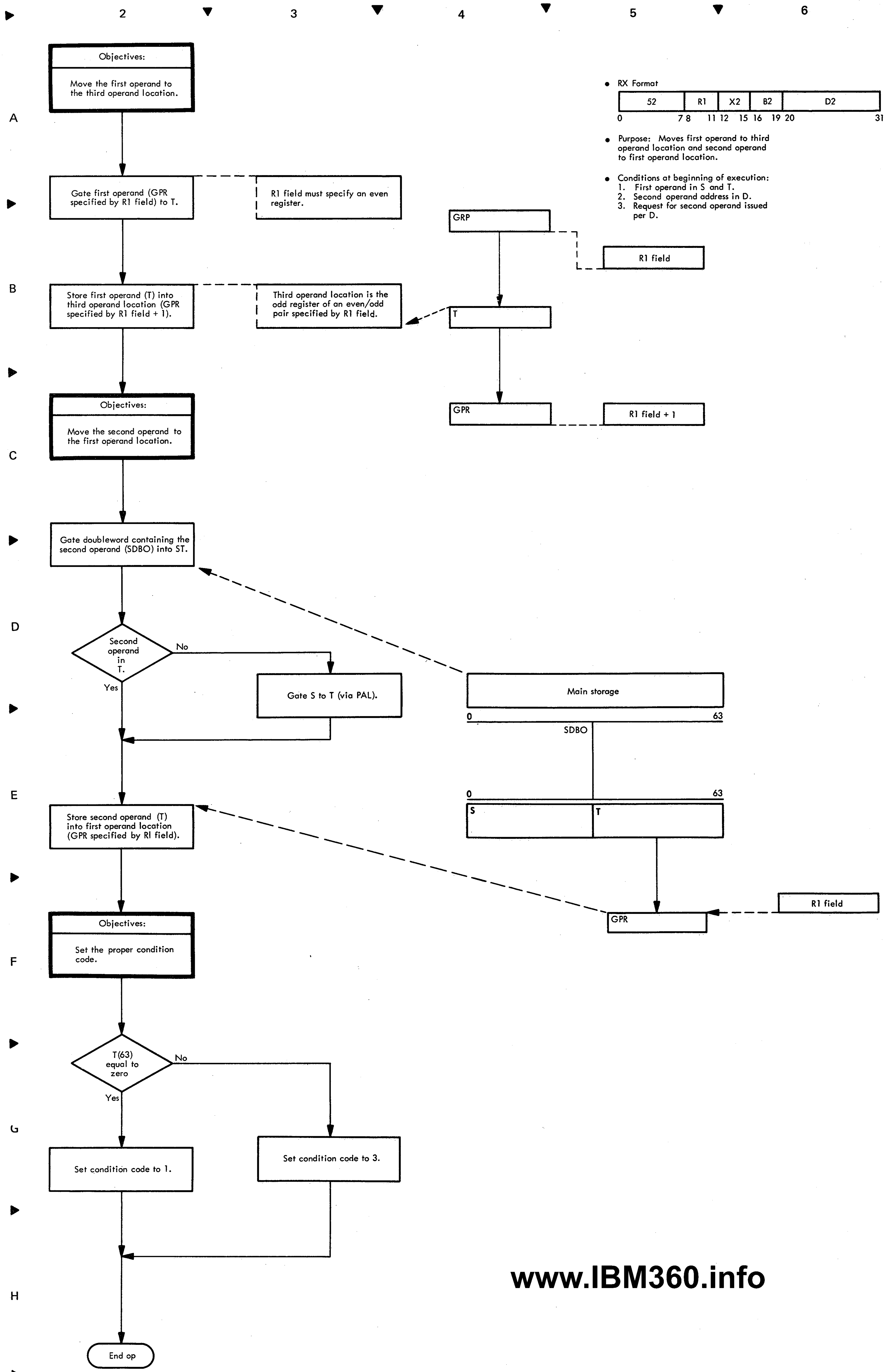


Diagram 5-906. Load Chain, LC (52)

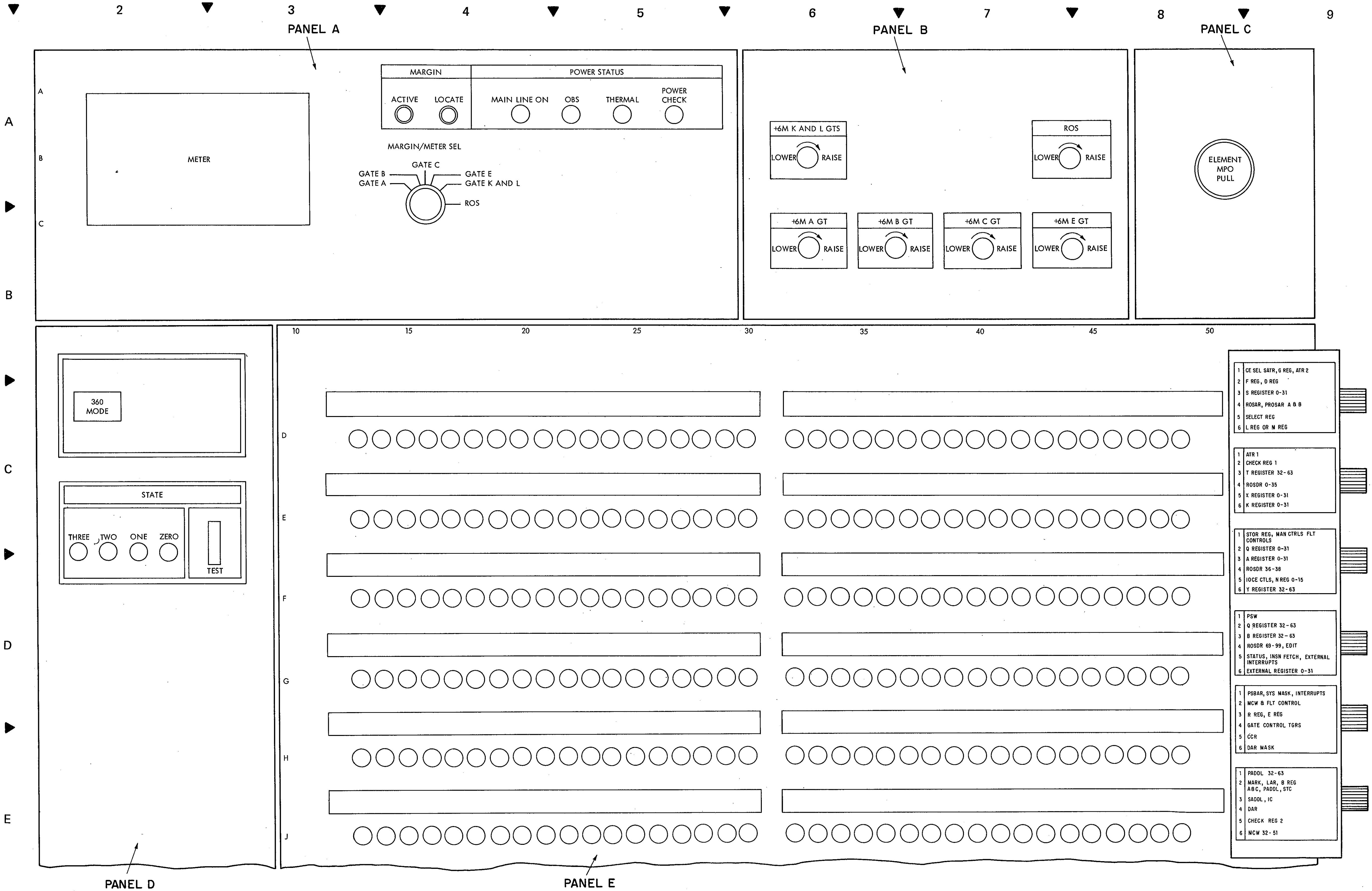


Diagram 6-1. CE Control Panel (Sheet 1 of 2)

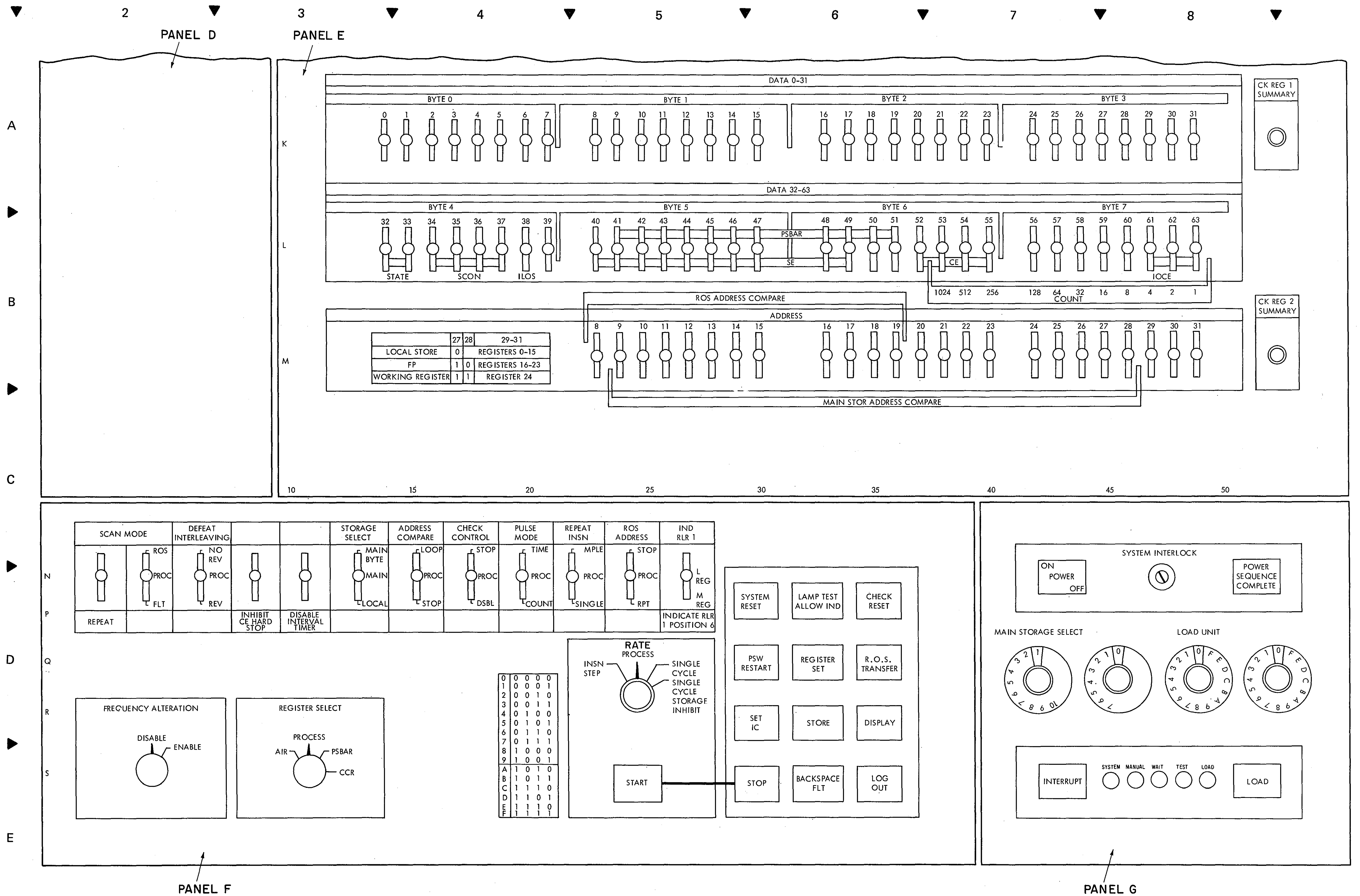


Diagram 6-1. CE Control Panel (Sheet 2 of 2)

DISPLAY NO. 1

1	CE SEL ATR: 1, 2, 3, 4 · T C STEP · G REGISTER (WRITE DIRECT): P 0-7, 0, 1, 2, 3, 4, 5, 6, 7 · ATR-2: P 32-39, 32, 33, 34, 35, 36, 37, 38, 39
2	F REGISTER (READ DIRECT): P 0-7, 0, 1, 2, 3, 4, 5, 6, 7, P 8-15, 8, 9, 10, 11, 12, 13, 14, 15 · D REGISTER: P 16-23, 16, 17, 18, 19, 20, 21, 22, 23, P 24-31, 24, 25, 26, 27, 28, 29, 30, 31
3	S REGISTER: P 0-7, 0, 1, 2, 3, 4, 5, 6, 7, P 8-15, 8, 9, 10, 11, 12, 13, 14, 15, P 16-23, 16, 17, 18, 19, 20, 21, 22, 23, P 24-31, 24, 25, 26, 27, 28, 29, 30, 31
4	READ ONLY STORAGE ADDRESS REGISTER: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 · ROS PREVIOUS ADDRESS REGISTER A: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 · ROS PREVIOUS ADDRESS REGISTER B: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11
5	SELECT REGISTER: P 0-7 · PAM/RCU: 1, 2, 3 · TCU: 1, 2, 3 · P 8-15 · SE: 1, 2, 3, 4, 5, 6, 7 · SE/DE 8 · P 16-23, 9, 10 · CE: 1, 2, 3, 4 · P 24-31 · IOCE: 1, 2, 3
6*	L REGISTER: P 0-7, 0, 1, 2, 3, 4, 5, 6, 7, P 8-15, 8, 9, 10, 11, 12, 13, 14, 15, P 16-23, 16, 17, 18, 19, 20, 21, 22, 23, P 24-31, 24, 25, 26, 27, 28, 29, 30, 31
6*	M REGISTER: P 32-39, 32, 33, 34, 35, 36, 37, 38, 39, P 40-47, 40, 41, 42, 43, 44, 45, 46, 47, P 48-55, 48, 49, 50, 51, 52, 53, 54, 55, P 56-63, 56, 57, 58, 59, 60, 61, 62, 63

DISPLAY 1

1	CE SEL SATR, G REG, ATR-2
2	F REG, D REG
3	S REG (0-31)
4	ROSAR, PROSAR A AND B
5	SELECT REG
6	L REG*
6	M REG*

*Selected via INDICATE ON ROLLER 1 POSITION 6 switch.

DISPLAY NO. 2

1	ADDRESS TRANSLATION REGISTER 1: P 0-7, 0, 1, 2, 3, 4, 5, 6, 7, P 8-15, 8, 9, 10, 11, 12, 13, 14, 15, P 16-23, 16, 17, 18, 19, 20, 21, 22, 23, P 24-31, 24, 25, 26, 27, 28, 29, 30, 31
2	E REG PTY · PADD FULL SUM: 4-7, 8-15, 16-23, 24-31, 32-39, 40-47, 48-55, 56-63, 64-67 · MLPR DECODE PTY · PADD HALF-SUM: 4-7, 8-15, 16-23, 24-31, 32-39, 40-47, 48-55, 56-67 · ERROR TGR · ROS PTY: 2-42, 43-68, 69-99 · SADD: HALF SUM, FULL SUM
3	T REGISTER: P 32-39, 32, 33, 34, 35, 36, 37, 38, 39, P 40-47, 40, 41, 42, 43, 44, 45, 46, 47, P 48-55, 48, 49, 50, 51, 52, 53, 54, 55, P 56-63, 56, 57, 58, 59, 60, 61, 62, 63
4	0, 1 · W FAA CONTROLS: 2, 3, 4, 5 · A AB, IC INGATES: 6, 7, 8, 9 · B LS-T 10, LS-S 11 · C ST, D, Q, G, PSW INGATES: 12, 13, 14, 15, 16 · D FLT 17 · F INGT AND EOP: A FIELD 18, SD 19, P 6-42 · E EMIT E, R INGATES: 21, 22, 23, 24 · F MISC CONTROL PART 2 FLT B FIELD: 25, 26, 27, 28, 29, SF 30 · G MISC CONTROL PART 1 FLT C FIELD: 31, 32, 33, 34, SG 35
5	X REGISTER: P 0-7, 0, 1, 2, 3, 4, 5, 6, 7, P 8-15, 8, 9, 10, 11, 12, 13, 14, 15, P 16-23, 16, 17, 18, 19, 20, 21, 22, 23, P 24-31, 24, 25, 26, 27, 28, 29, 30, 31
6	K REGISTER: P 0-7, 0, 1, 2, 3, 4, 5, 6, 7, P 8-15, 8, 9, 10, 11, 12, 13, 14, 15, P 16-23, 16, 17, 18, 19, 20, 21, 22, 23, P 24-31, 24, 25, 26, 27, 28, 29, 30, 31

DISPLAY 2

1	ATR 1
2	CHECK REG 1
3	T REG (32-63)
4	ROSDR (0-35)
5	X REG (0-31)
6	K REG (0-31)

DISPLAY NO. 3

1	STORAGE REQUEST: IC, D, FLT, SET KEY, INSERT KEY, 3 CYCLE · PAGE 1 LTH · MANUAL CONTROLS: RPT INSTR ADJ, INIT; PULSE MODE ADJ, INIT; STOP, BLOCK, PASS PULSE, BLOCK INTRPT, FORCE ADDR, INSTR STEP, SINGLE CYCLE, CHK SUMM, INHIB CLOCK · FLT CONTROLS: RPT TEST, FLT TEST, ROS TEST, SCC, SYNC, DEFEAT INTLV NO REV, REV, BACK SPACE, CNSL LOG, SOROS, DIAG, DIAG RELEASE, TIC, GAP
2	Q REGISTER: P 0-7, 0, 1, 2, 3, 4, 5, 6, 7, P 8-15, 8, 9, 10, 11, 12, 13, 14, 15, P 16-23, 16, 17, 18, 19, 20, 21, 22, 23, P 24-31, 24, 25, 26, 27, 28, 29, 30, 31
3	A REGISTER: P 0-7, 0, 1, 2, 3, 4, 5, 6, 7, P 8-15, 8, 9, 10, 11, 12, 13, 14, 15, P 16-23, 16, 17, 18, 19, 20, 21, 22, 23, P 24-31, 24, 25, 26, 27, 28, 29, 30, 31
4	HW EXTD: 36, 37 · H LS LAR CONTROL: 38, 39, 40, 41, 42 · L STOR REQ SET MARK: 43, 44, 45, 46 · NA, AD NEXT ROS BASE ADDRESS: 47, 48, 49, 50, 51, 52, 53, 54, 55, 56 · K Y BRANCH: 57, 58, 59, 60, 61 · J X OR Z BRANCH: 62, 63, 64, 65, 66, 67, 68 · PREV ADR A · FLT MODE
5	PERMIT IOCE: 1, 2, 3 · IOCE MACH CHK: 1, 2, 3 · SEL IOCE: 1, 2, 3 · IOCE: INTRPT REQ, MC REQ · INTRPT GATE TGR · N REGISTER: P 0-7, 0, 1, 2, 3, 4, 5, 6, 7, P 8-15, 8, 9, 10, 11, 12, 13, 14, 15
6	Y REGISTER: P 32-39, 32, 33, 34, 35, 36, 37, 38, 39, P 40-47, 40, 41, 42, 43, 44, 45, 46, 47, P 48-55, 48, 49, 50, 51, 52, 53, 54, 55, P 56-63, 56, 57, 58, 59, 60, 61, 62, 63

DISPLAY 3

1	STOR REQ, MAN CTLS, FLT CTLS
2	Q REG (0-31)
3	A REG (0-31)
4	ROSDR (36-68)
5	IOCE CTLS, N REG (0-15)
6	Y REG (32-63)

Diagram 6-2. CE Roller Switch Indicators (Sheet 1 of 2)

DISPLAY 4

1	PSW
2	Q REG (32-63)
3	B REG (32-63)
4	ROSDR (69-99), EDIT
5	STATUS, INSN FETCH, EXTERNAL INTERRUPTS
6	EXTERNAL REG 0-31

DISPLAY 5

1	PSBAR, SYS MASK, INTRPTS
2	MCW AND FLT CONTROL
3	R REG, E REG
4	GATE CONTROL TGRS
5	CCR
6	DAR MASK

DISPLAY 6

1	PADDL (32-63)
2	MARK, LAR, B REG, ABC, PADDL, STC
3	SADDL, IC
4	DAR
5	CHECK REG 2
6	MCW (32-51)

Diagram 6-2. CE Roller Switch Indicators (Sheet 2 of 2)

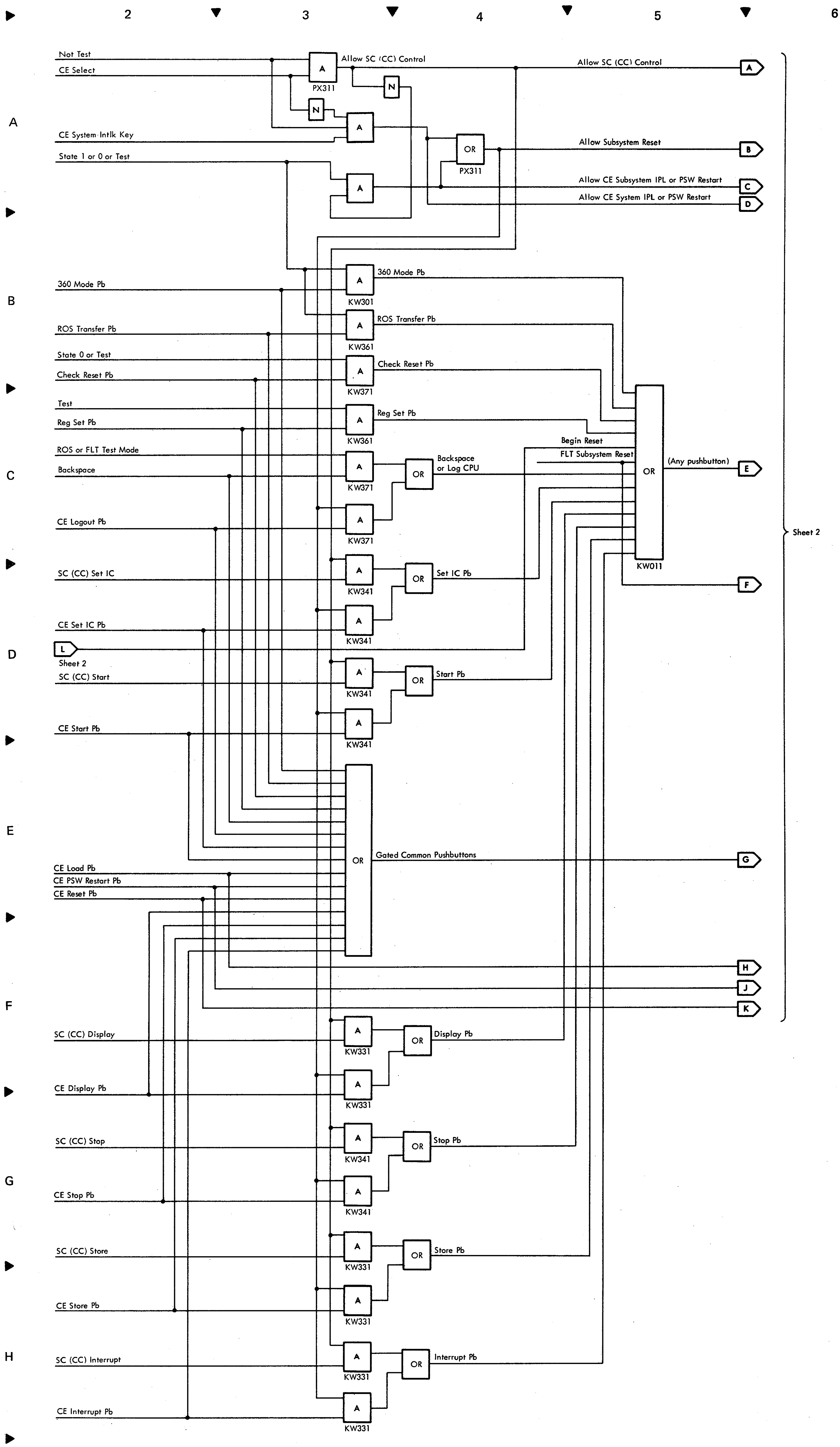


Diagram 6-3. Pushbutton Signal Generation (Sheet 1 of 2)

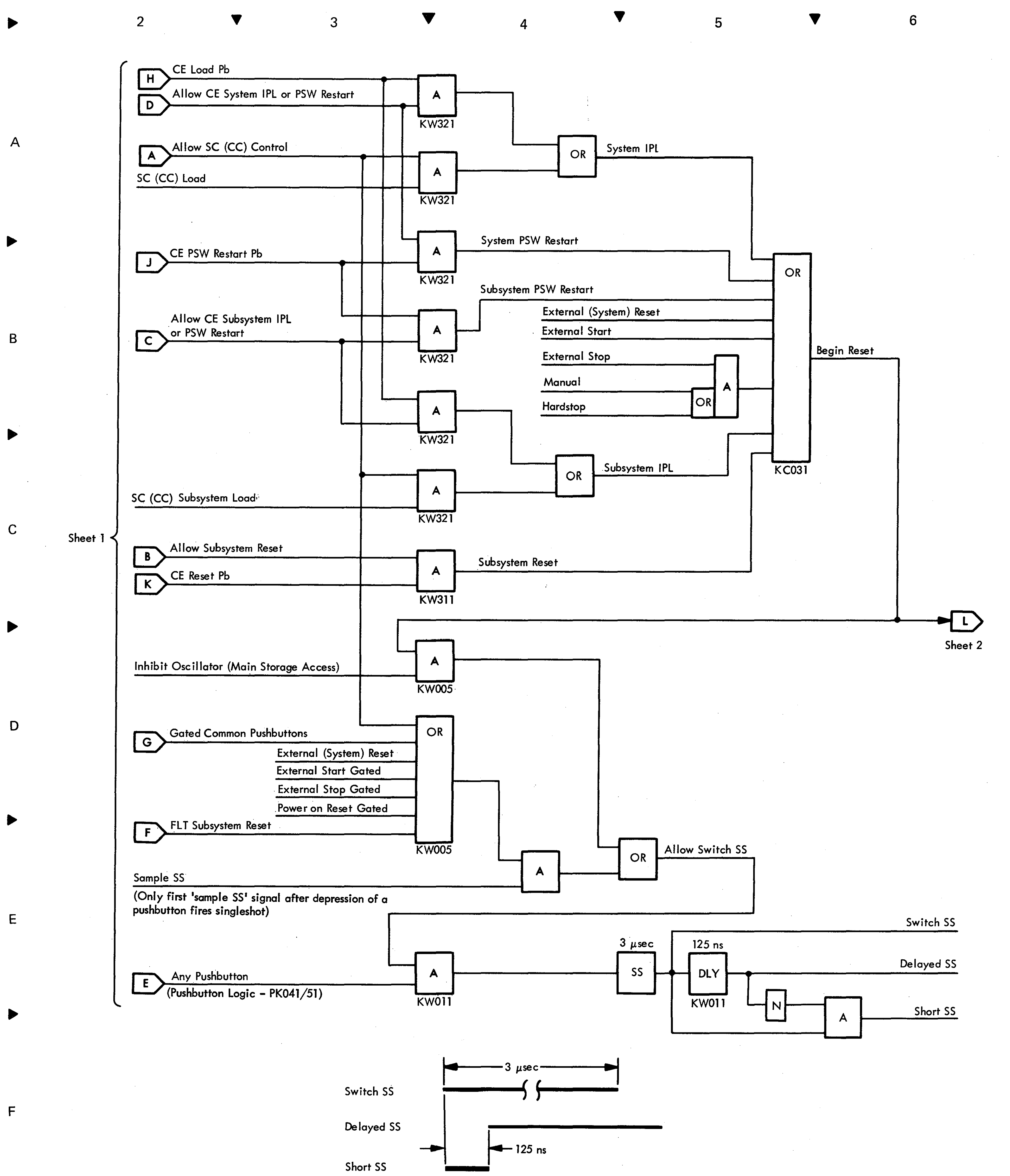


Diagram 6-3. Pushbutton Signal Generation (Sheet 2 of 2)

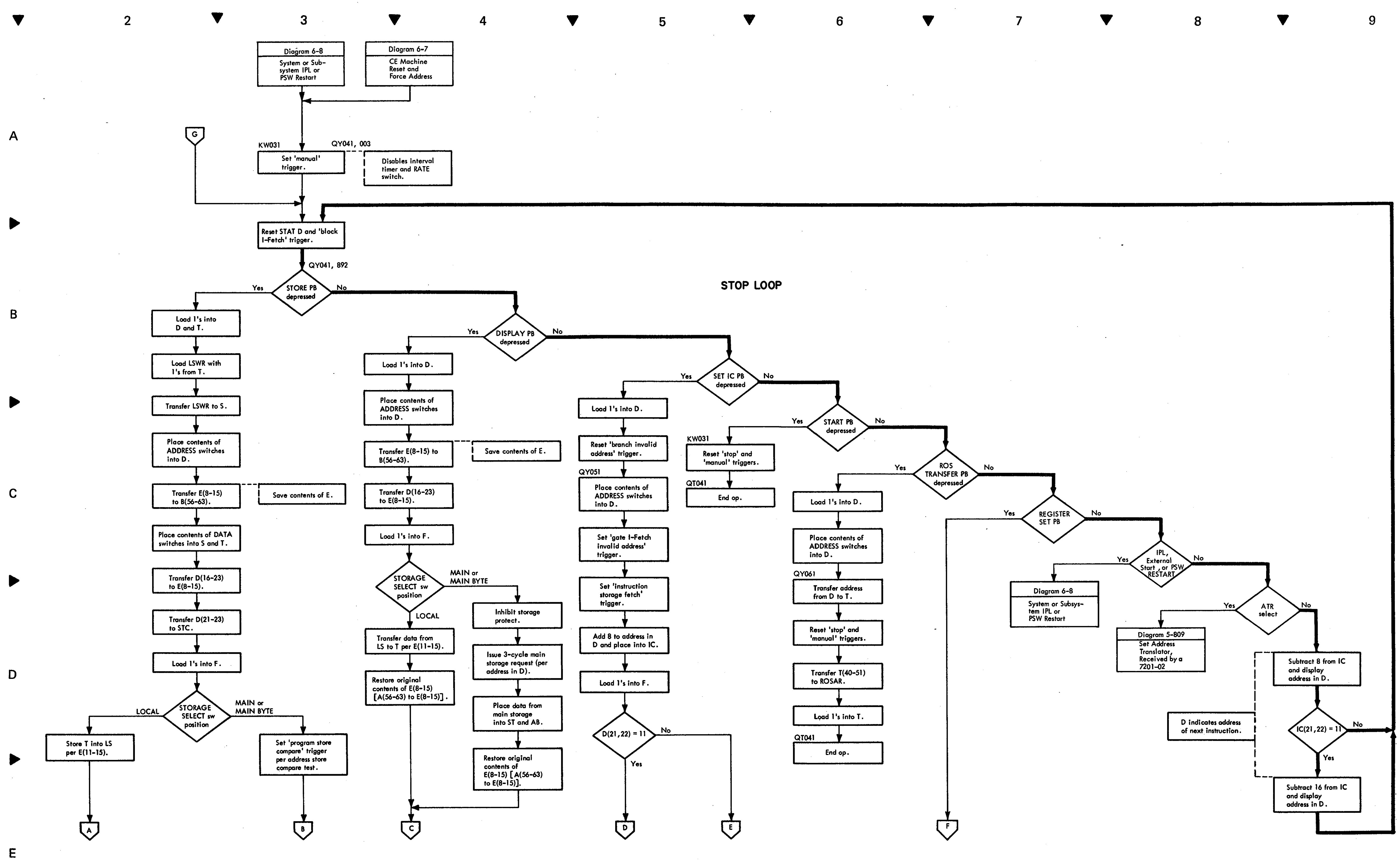


Diagram 6-4. Stop Loop Routine (Sheet 1 of 2)

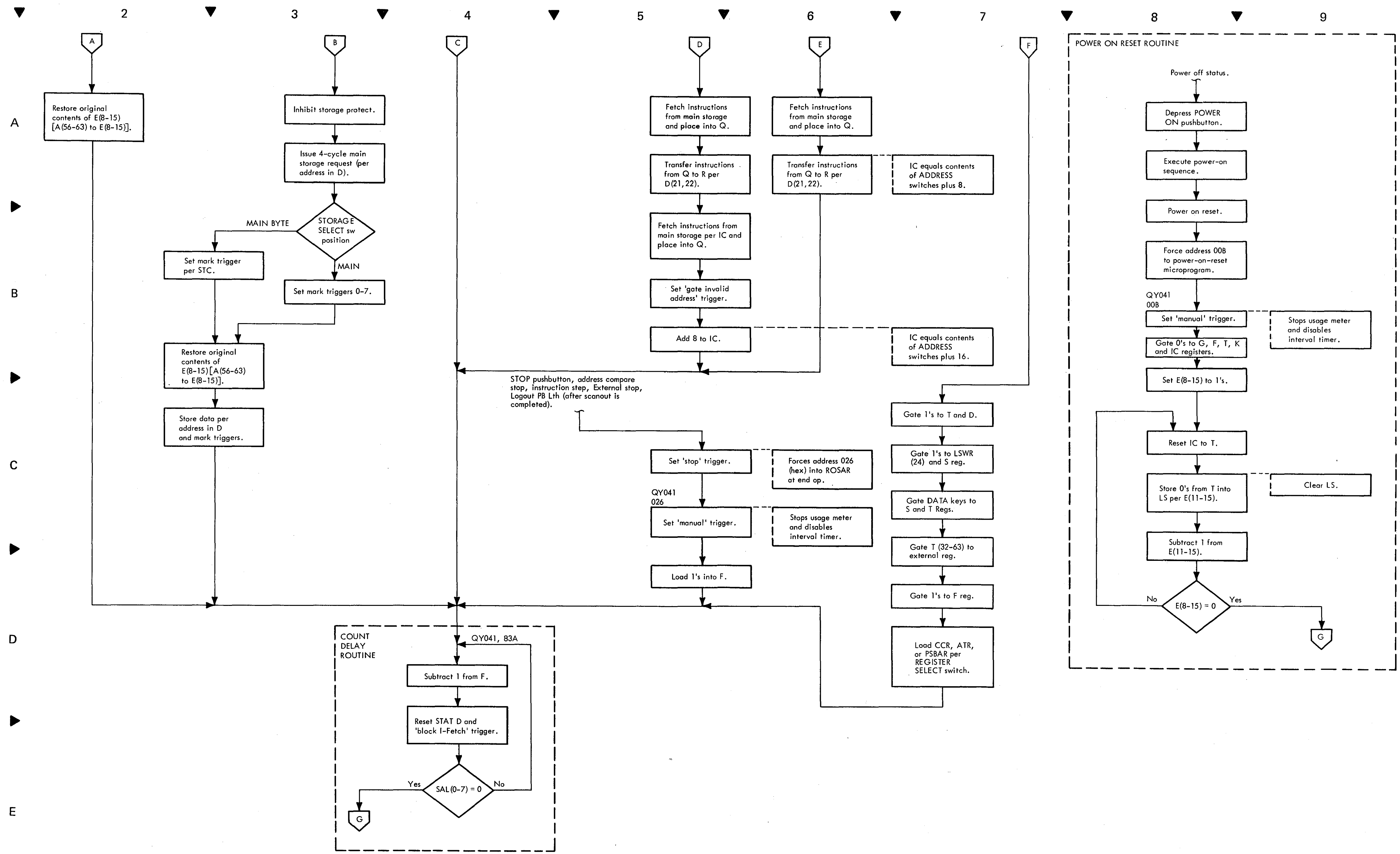


Diagram 6-4. Stop Loop Routine (Sheet 2 of 2)

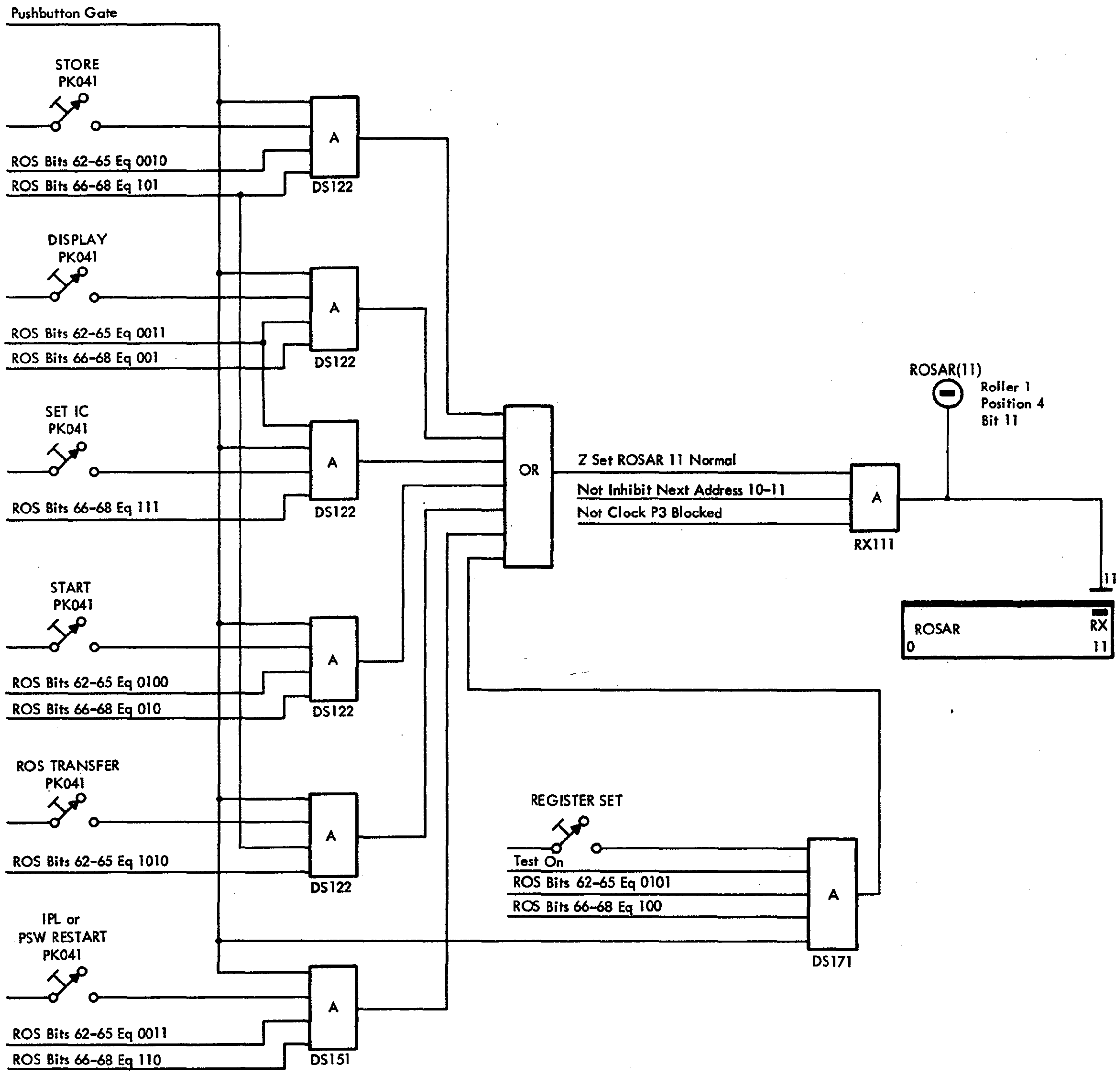


Diagram 6-5. Stop Loop Monitored Pushbutton Gating

2 3 4 5 6

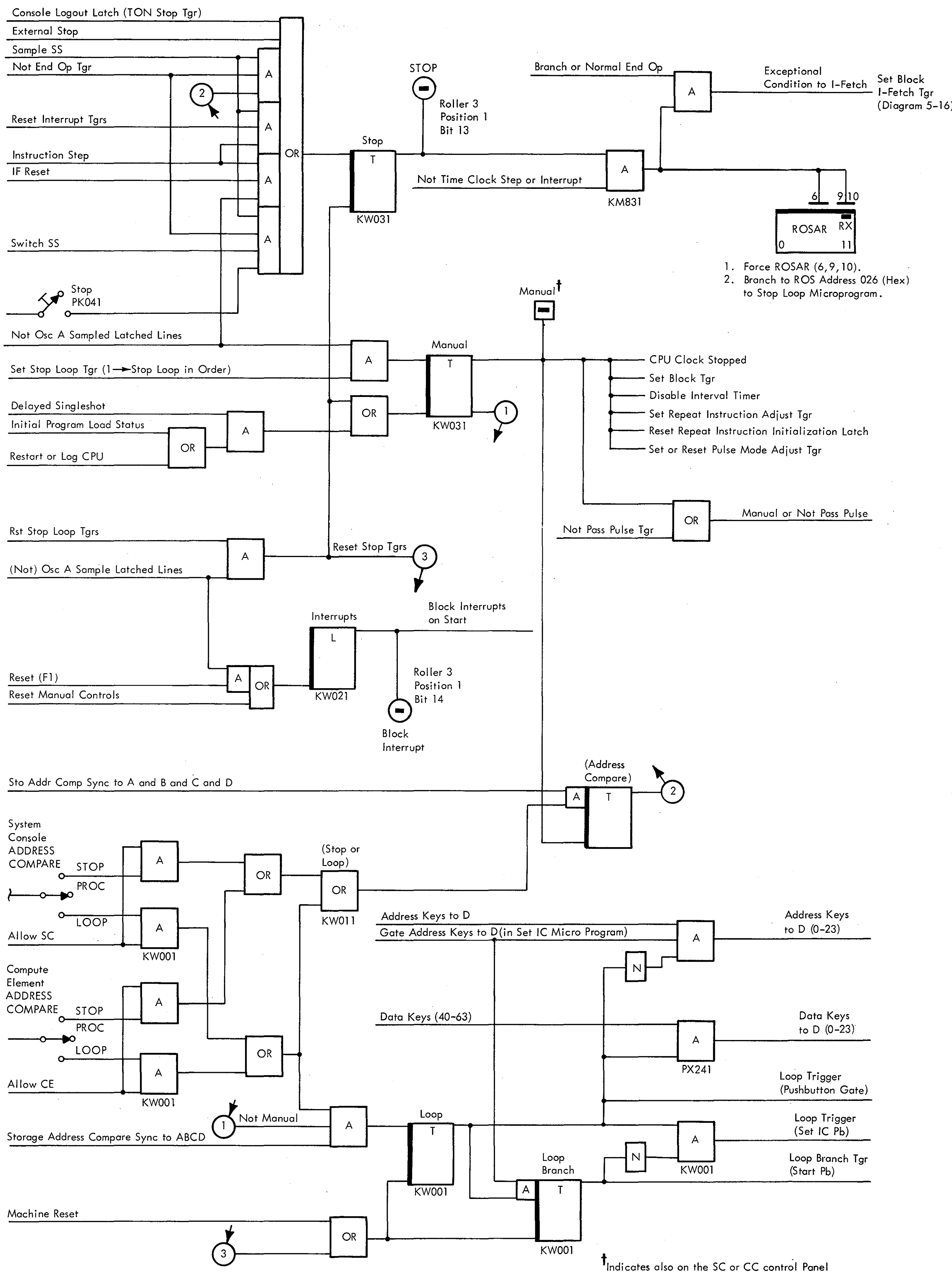


Diagram 6-6. Stop, Manual, Address Compare Triggers, and Block Interrupt Latch (Sheet 1 of 2)

A B C D E F G H

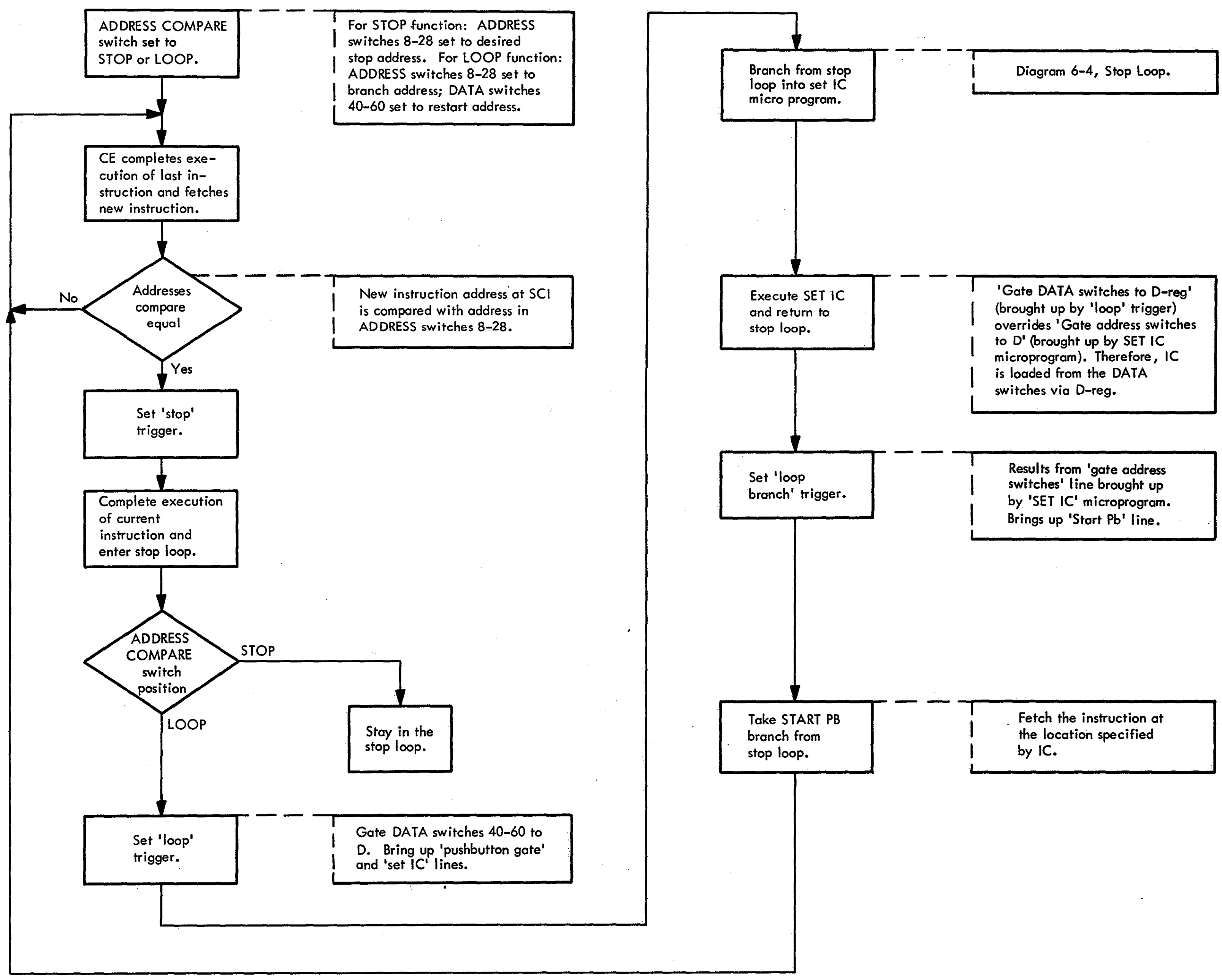


Diagram 6-6. Stop, Manual, Address Compare Triggers, and Block Interrupt Latch (Sheet 2 of 2)

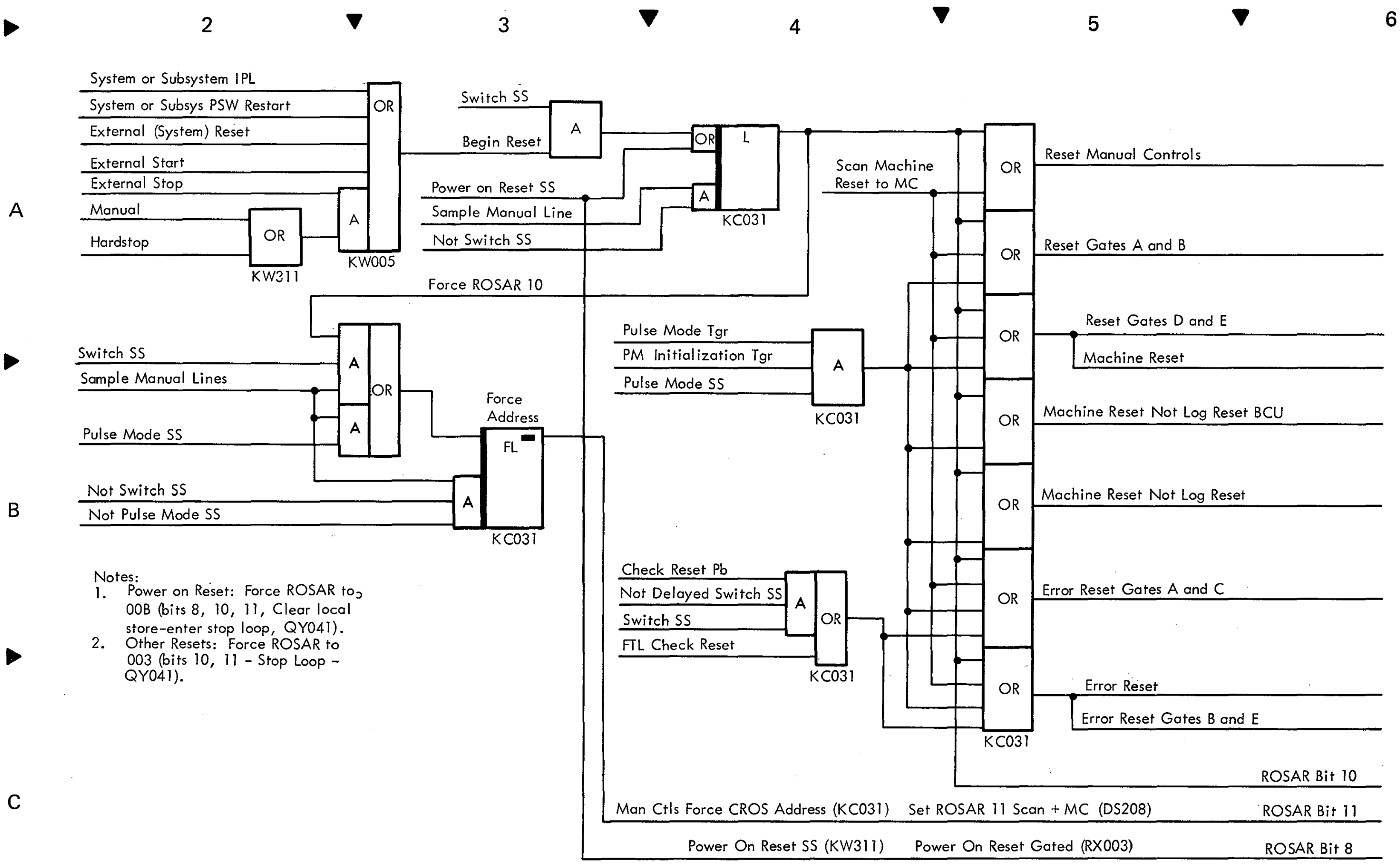
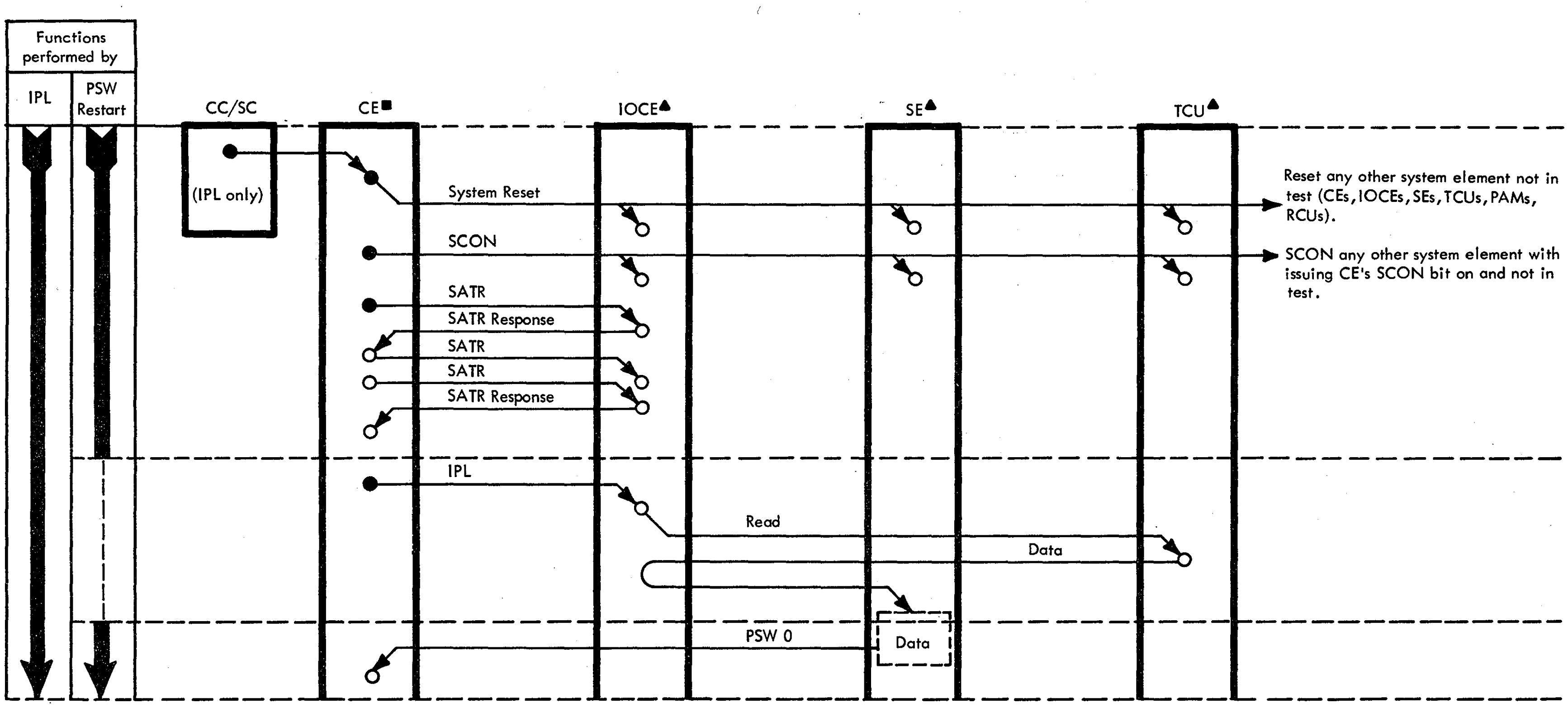


Diagram 6-7. CE Machine Reset and Force Address



Summary of System Operations: IPL or PSW Restart

Legend:
■ CE selected by CC/SC if IPL is initiated at CC/SC. CE at which IPL or PSW Restart is initiated.
▲ IOCE and TCU selected by SC/CC or CE LOAD UNIT switches. SE selected by SC/CC or CE MAIN STORAGE SELECT switch.
● Signal origin
○ Signal destination

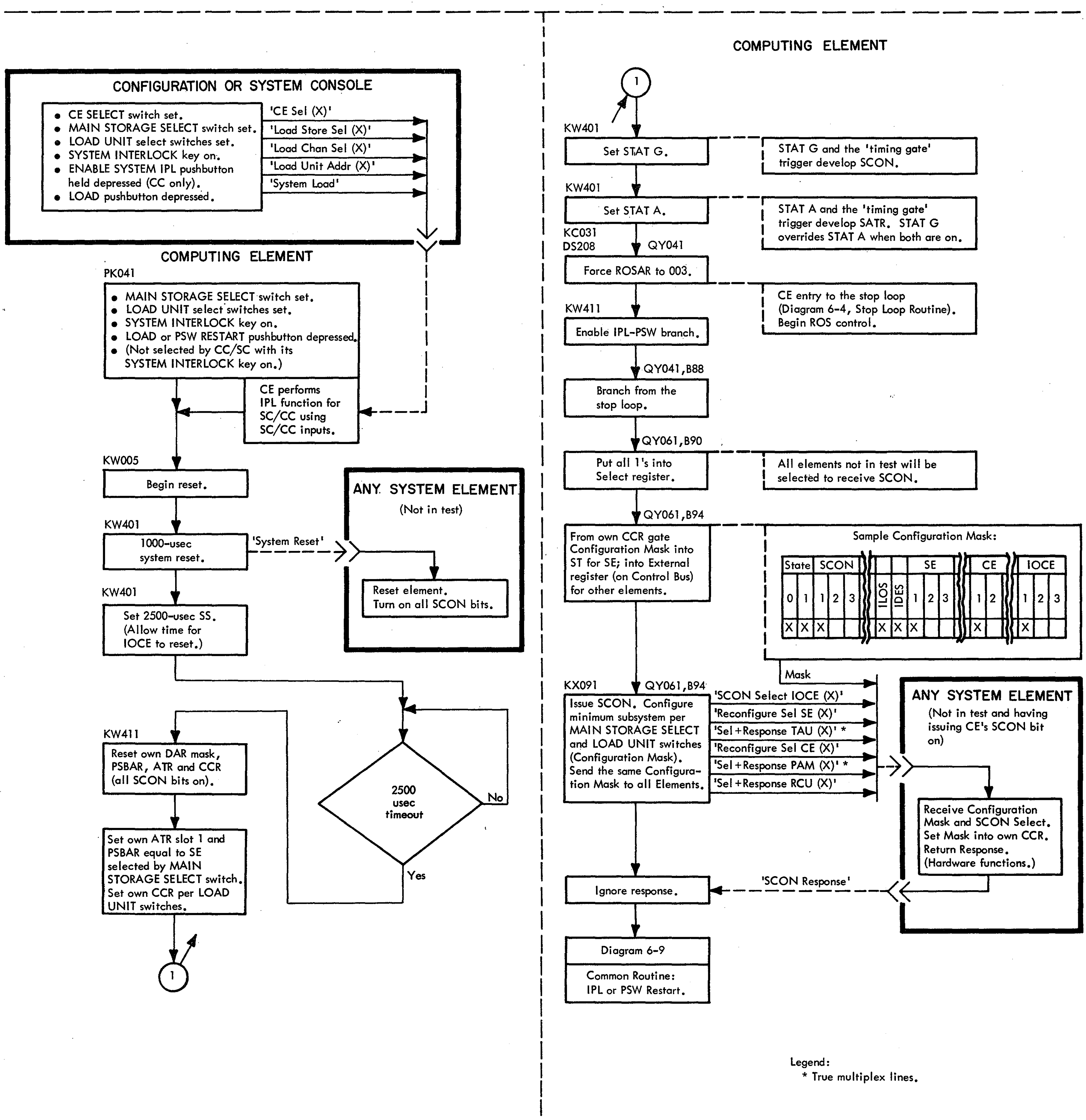
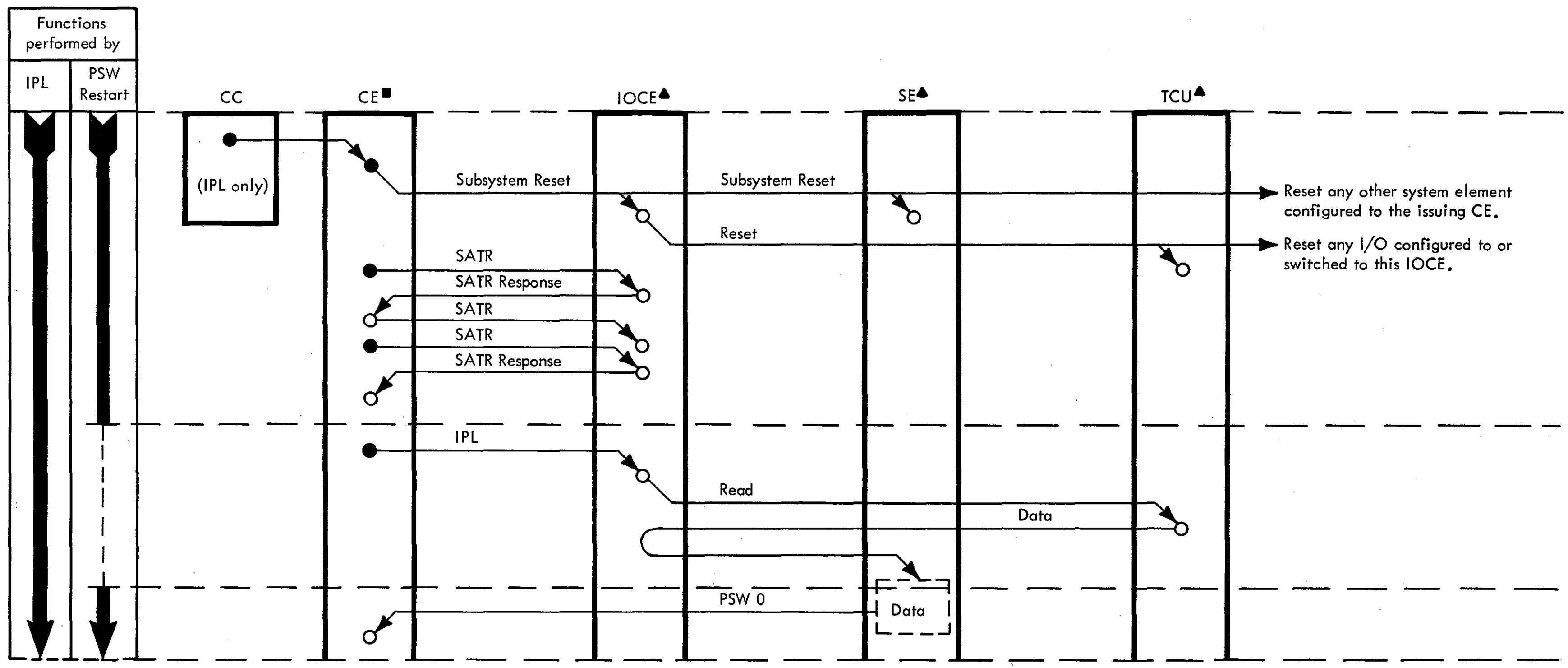


Diagram 6-8A. System Operation: IPL or PSW Restart



Summary of Subsystem Operations: IPL or PSW Restart.

Legend:
- ■ CE which is selected by CC if IPL is initiated at the CC. CE at which IPL or PSW Restart is initiated.
- ▲ IOCE and TCU selected by CC or CE LOAD UNIT switches and SE selected by CC or CE MAIN STORAGE SELECT switch.
- ● Signal origin
- ○ Signal destination

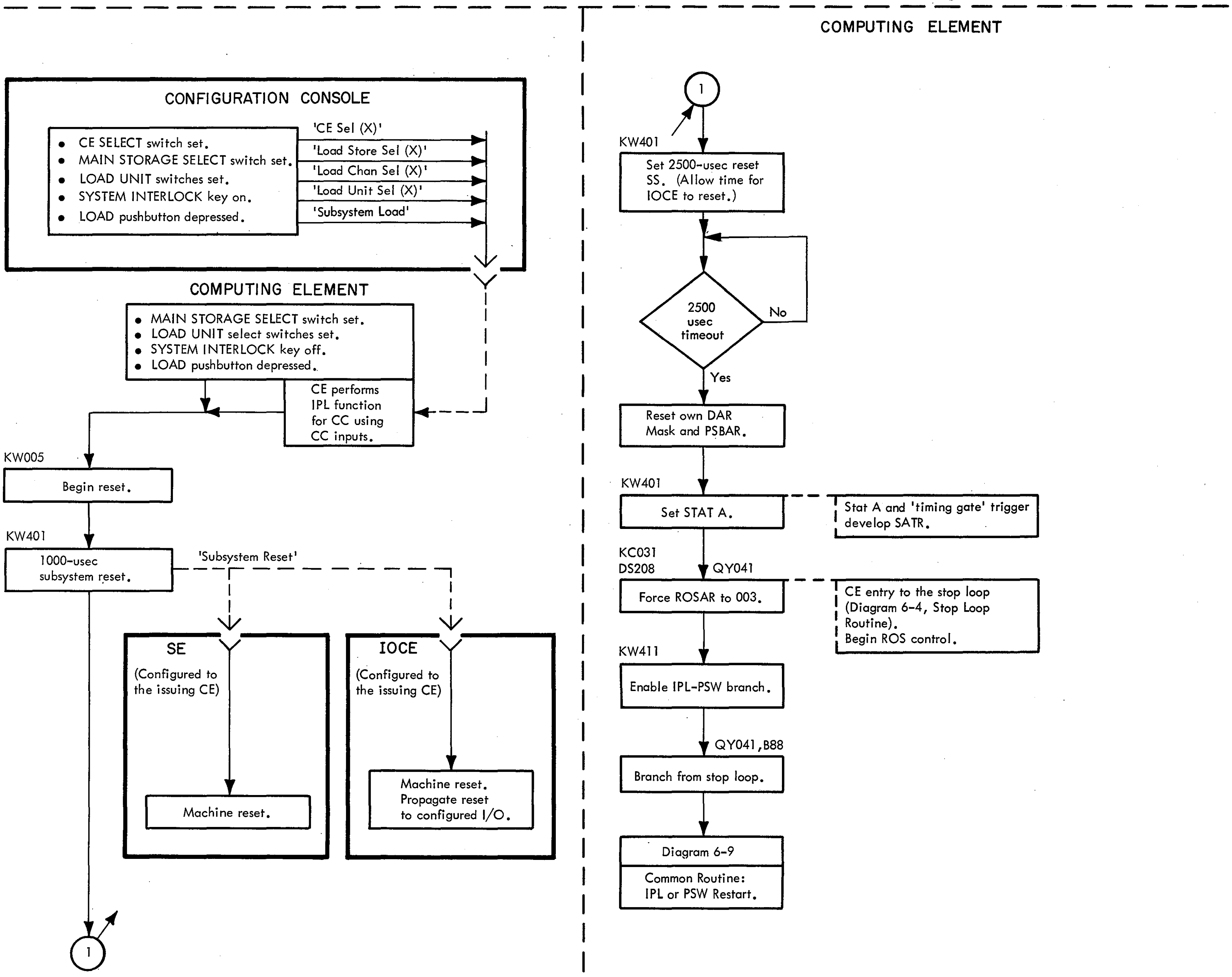


Diagram 6-8B. Subsystem Operation: IPL or PSW Restart

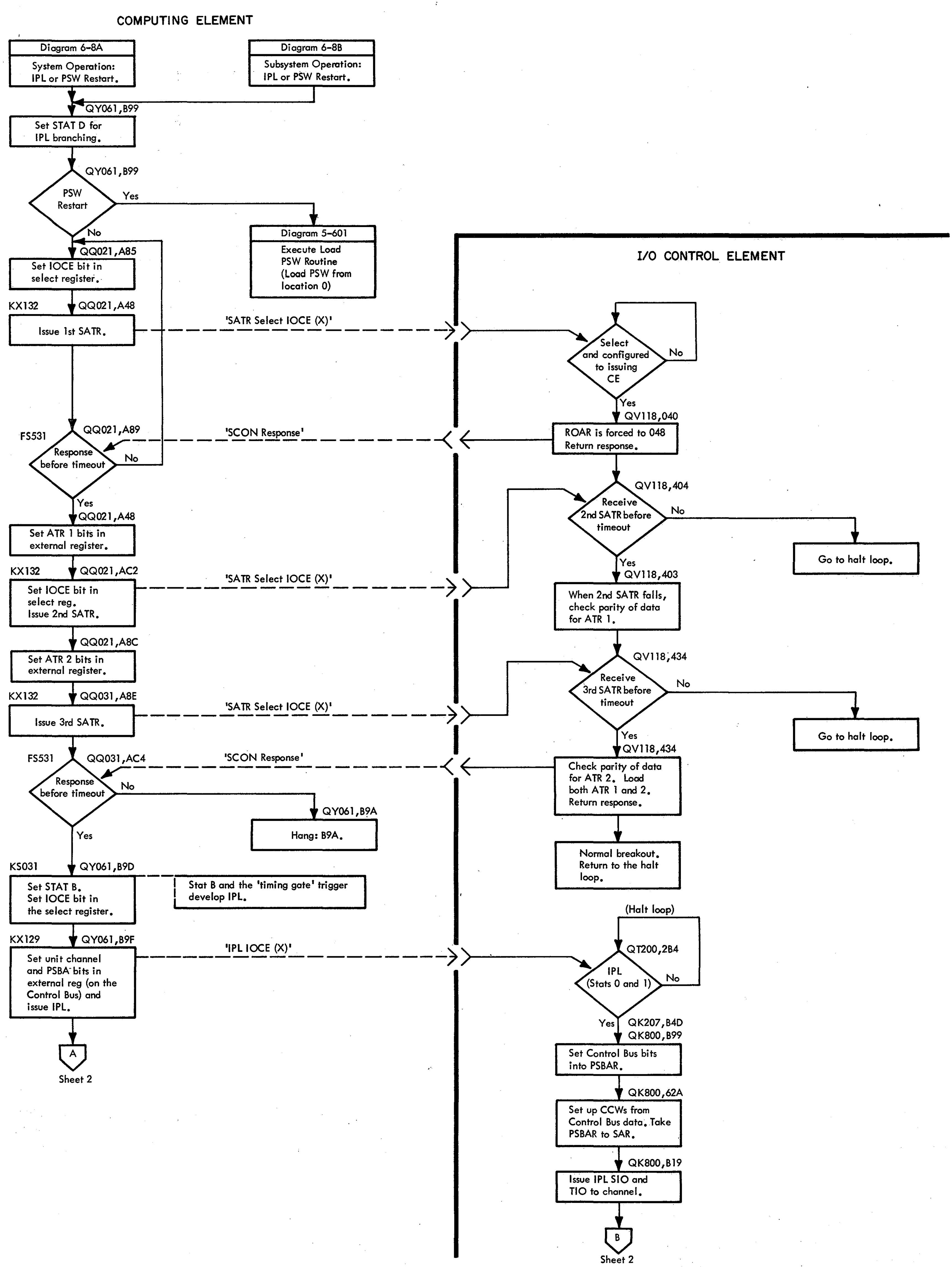
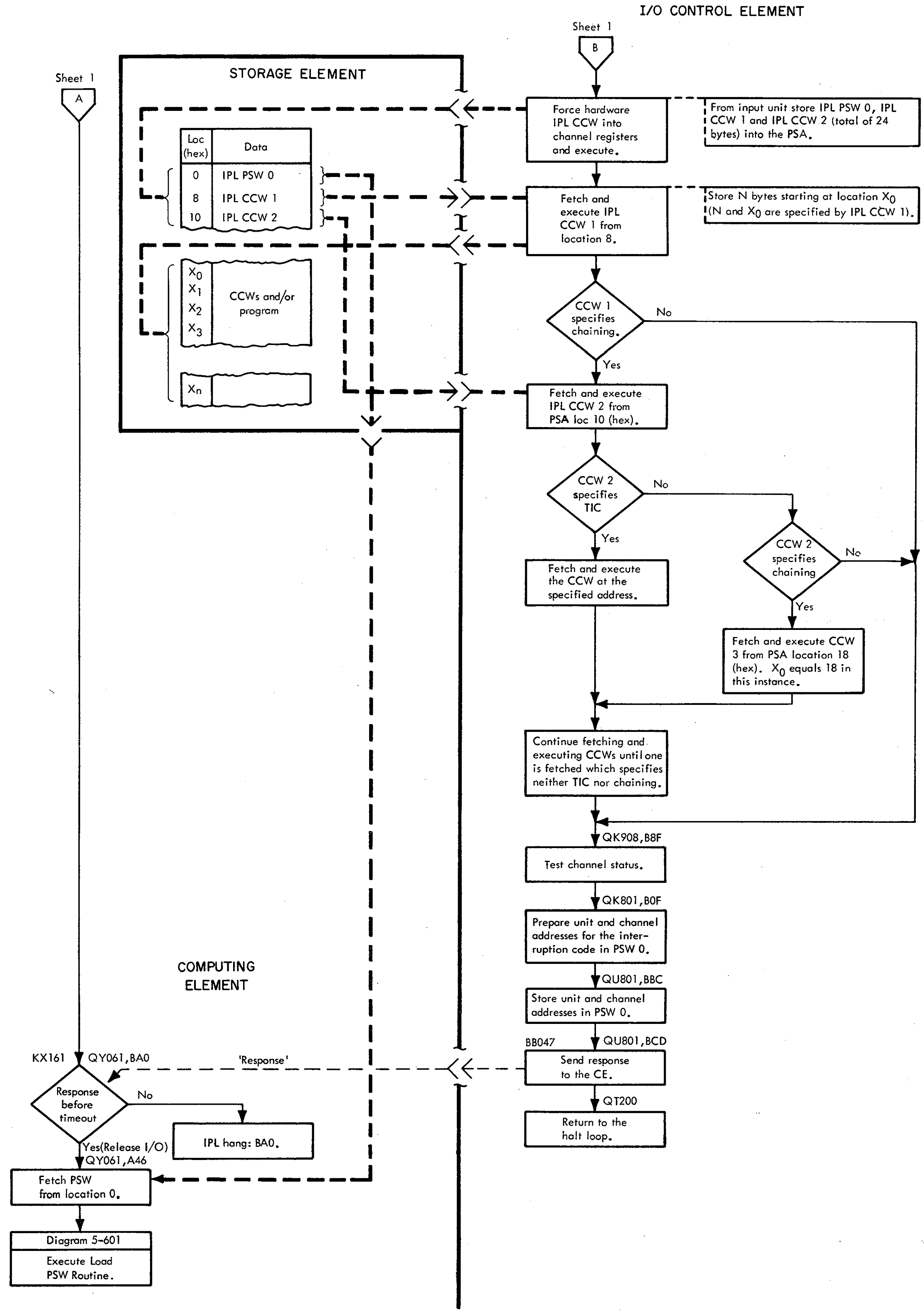


Diagram 6-9. Common Routine: IPL or PSW Restart (Sheet 1 of 2)



Legend: Heavy dashed lines indicate data flow.

Diagram 6-9. Common Routine: IPL or PSW Restart (Sheet 2 of 2)

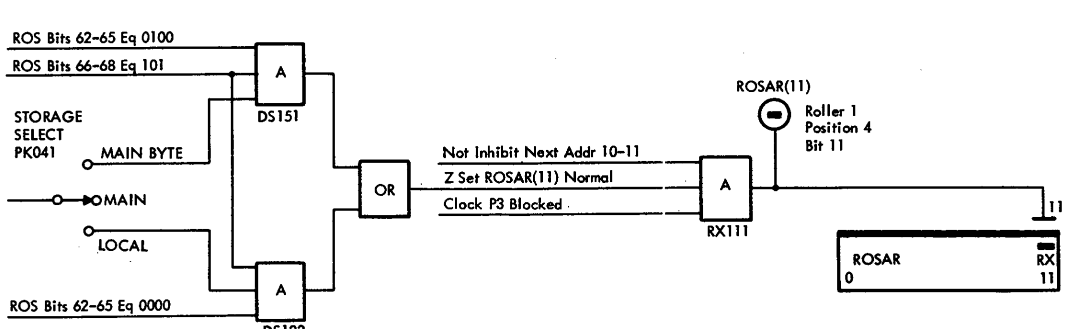


Diagram 6-10. STORAGE SELECT Switch Gating

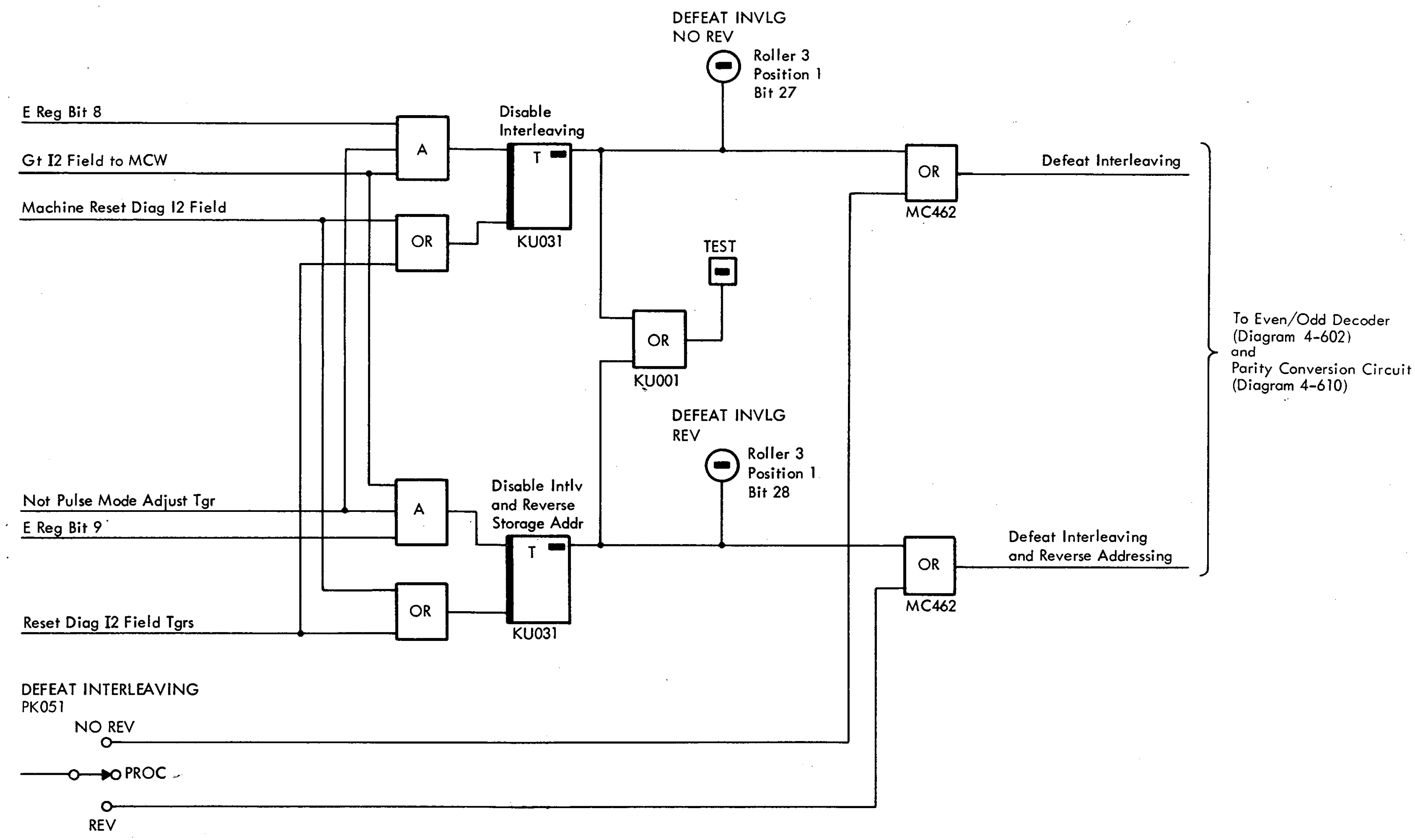


Diagram 6-11 DEFEAT INTERLEAVING Switch Gating

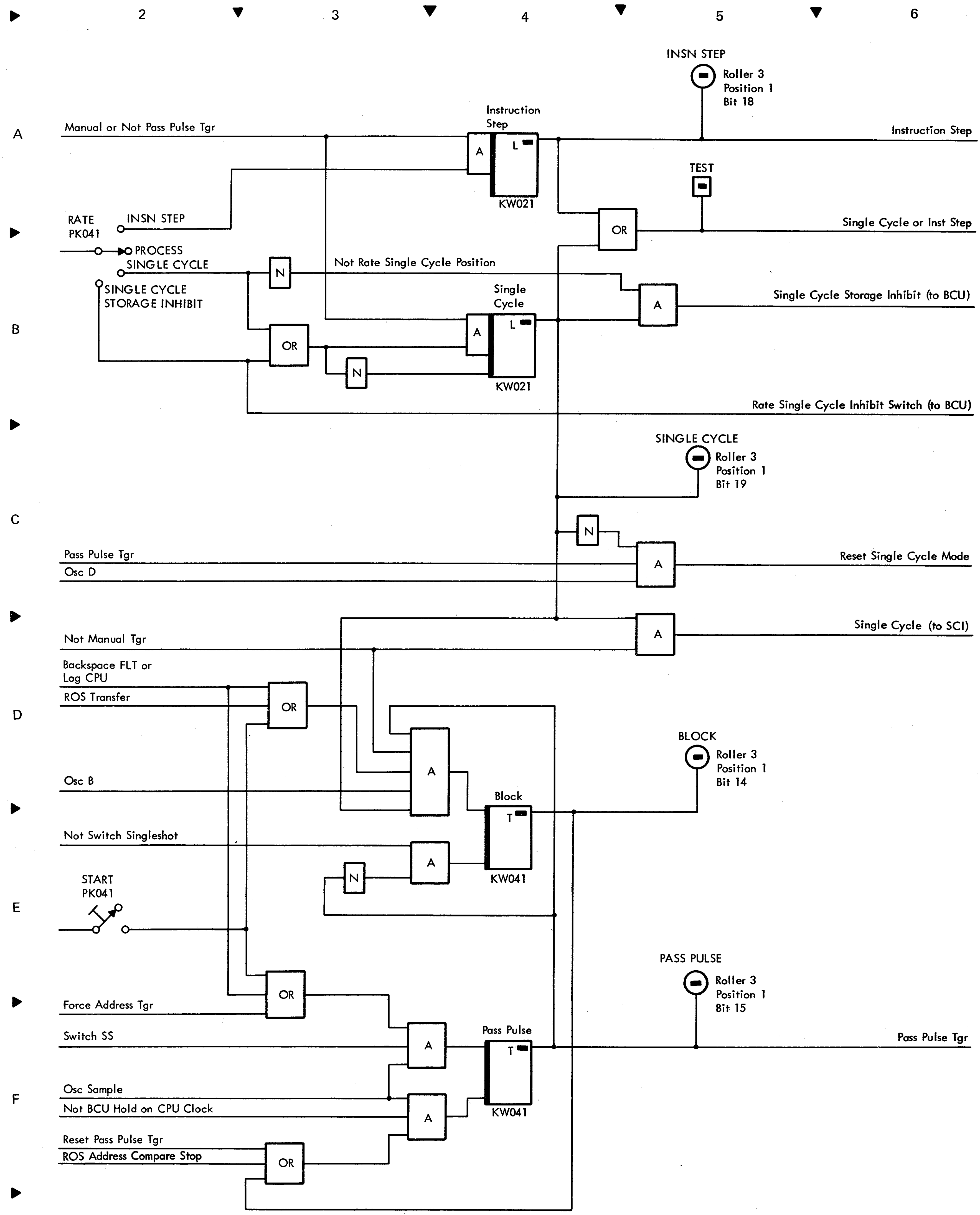


Diagram 6-12. RATE Switch Logic

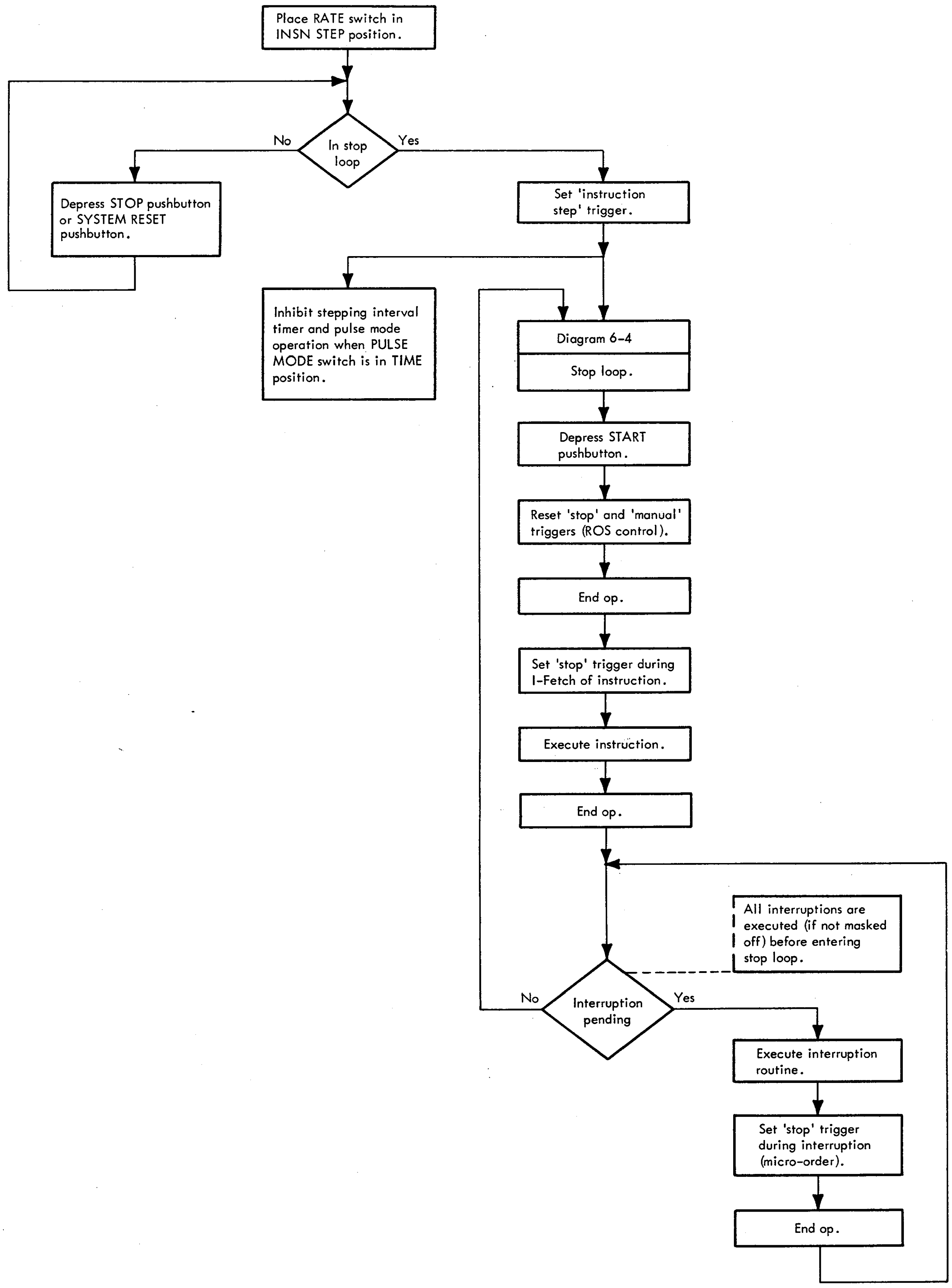


Diagram 6-13. Instruction Step Routine

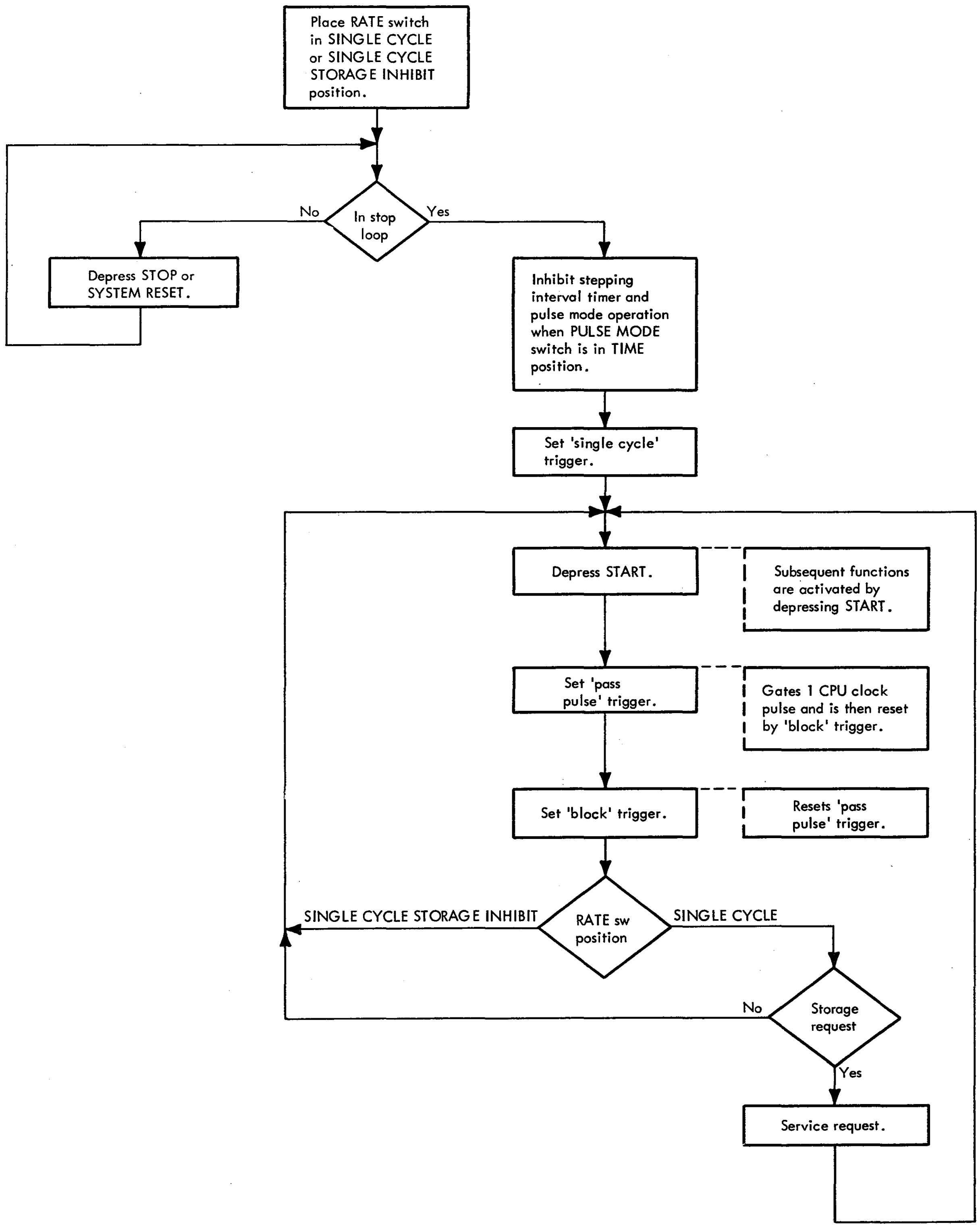


Diagram 6-14. Single-Cycle and Single-Cycle-Inhibit Routine

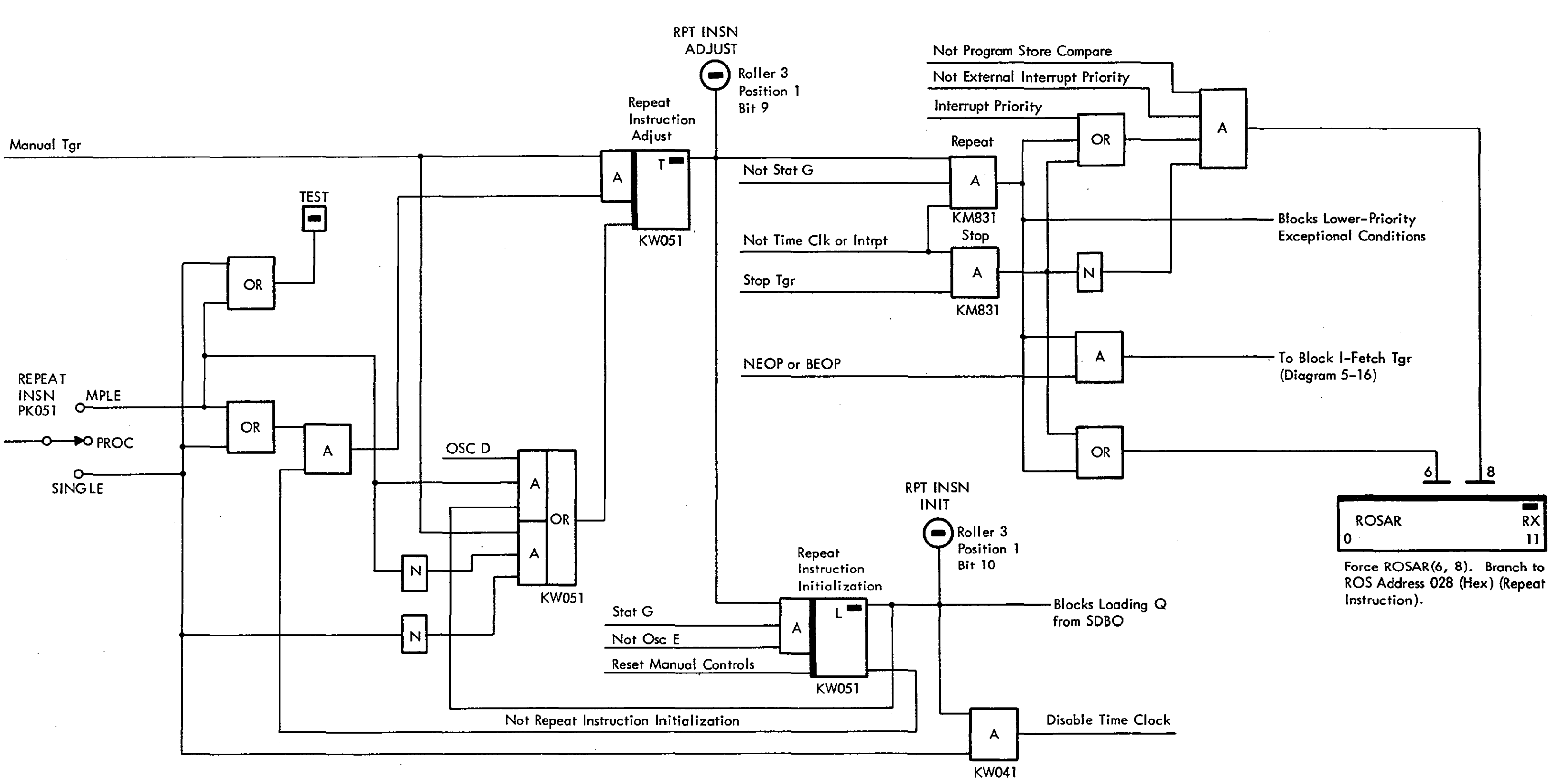


Diagram 6-15. Repeat Instruction Switch Logic

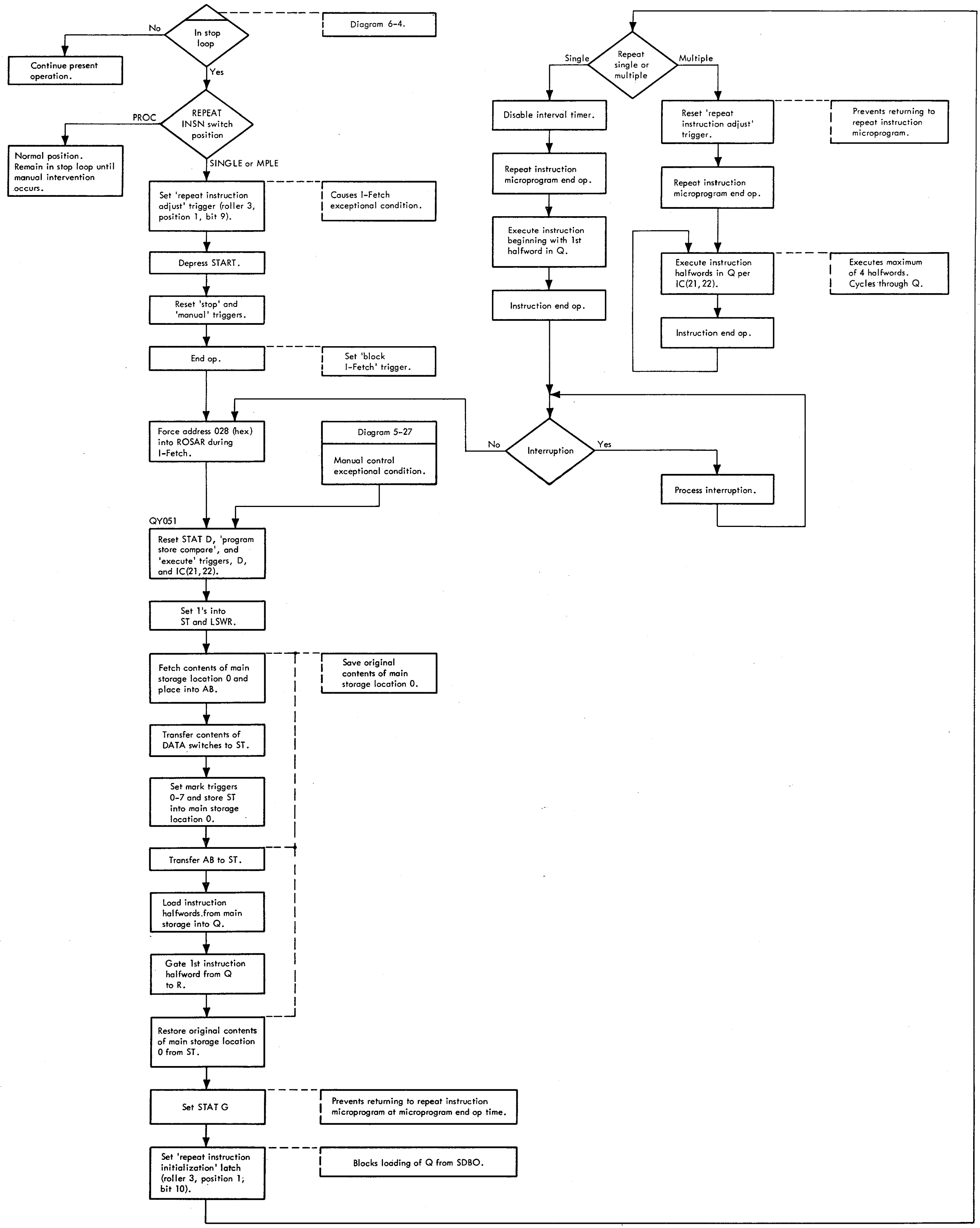


Diagram 6-16. Repeat Instruction Switch Routine

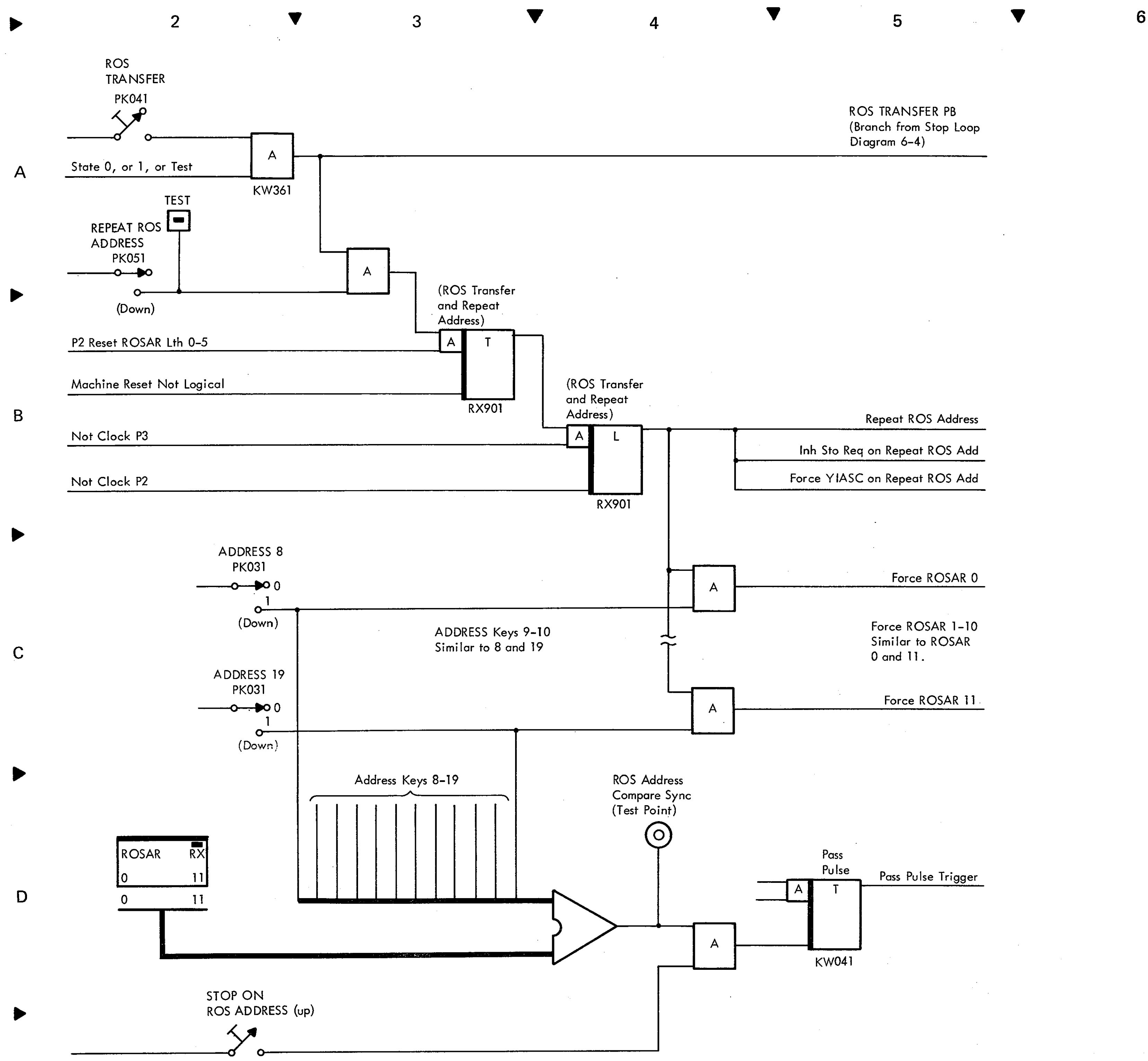
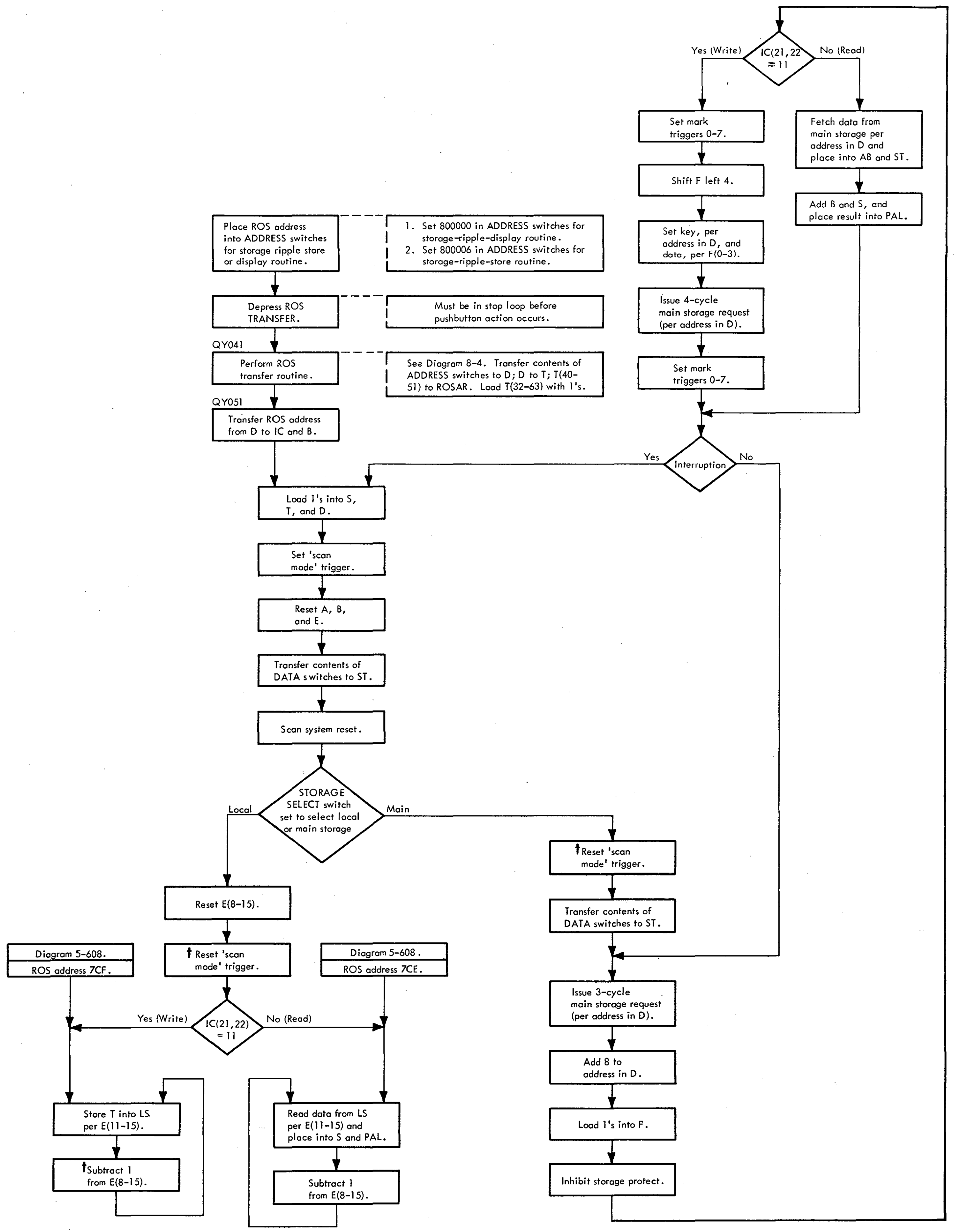


Diagram 6-17. ROS TRANSFER and REPEAT ROS ADDRESS Switch Gating



Note: Manual intervention (e.g., system reset or load) is required to get out of storage ripple loop.

† Following this block, an FLT-2, -3, or -4 microprogram is executed. These microprograms are inserted at this point to exercise circuits; they do not affect the storage ripple test.

Diagram 6-18. Storage Ripple Loop (Store and Display) Routine

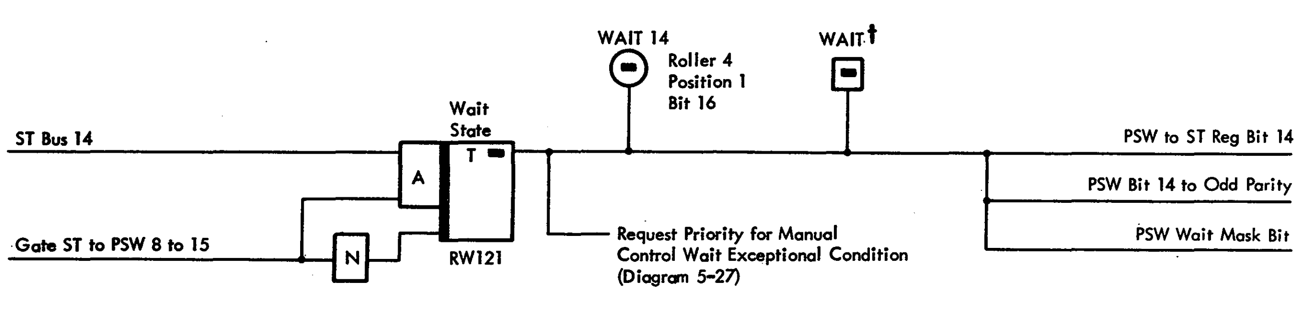


Diagram 6-19. Wait State Gating

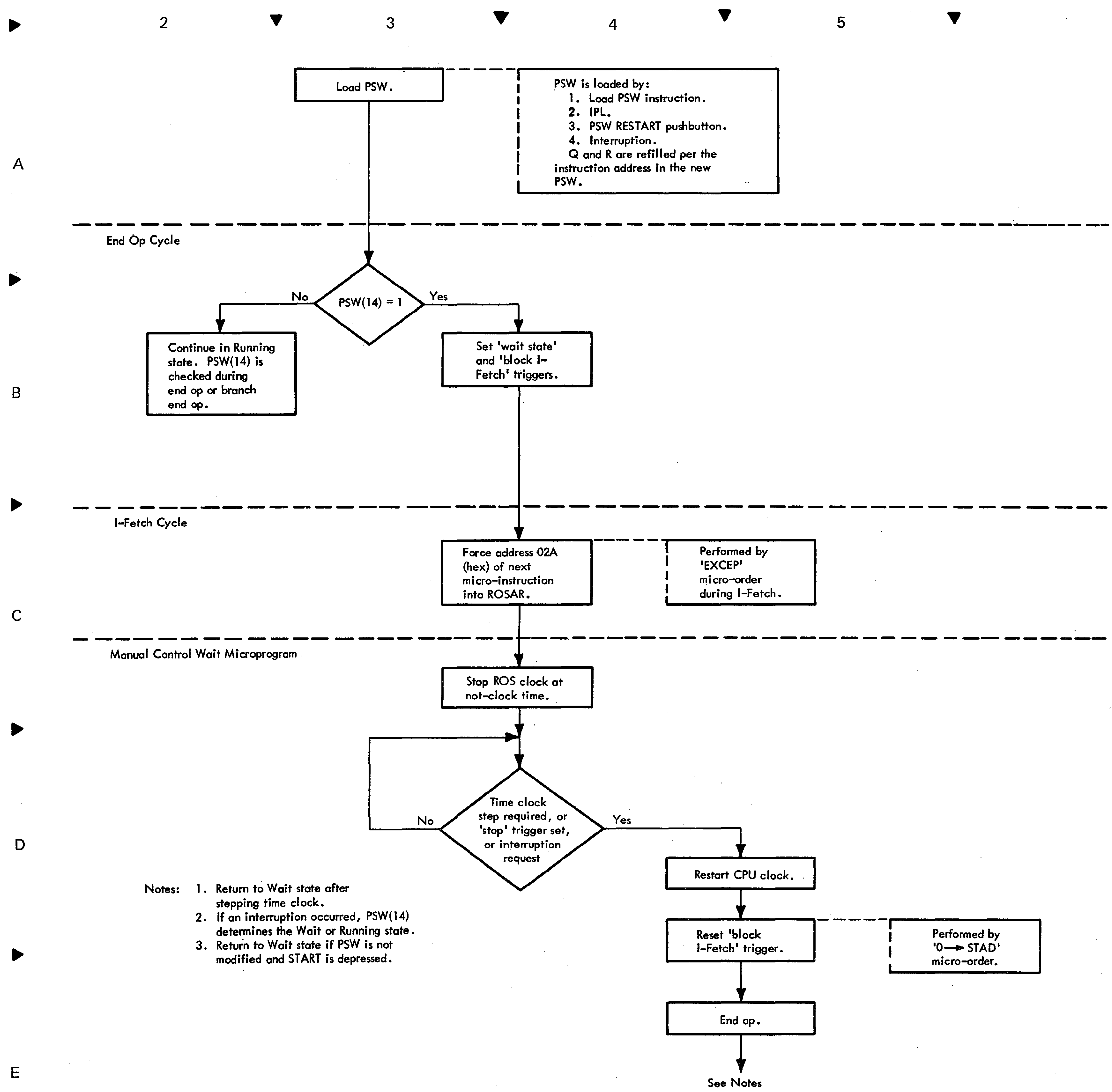


Diagram 6-20. Wait State Microprogram Routine

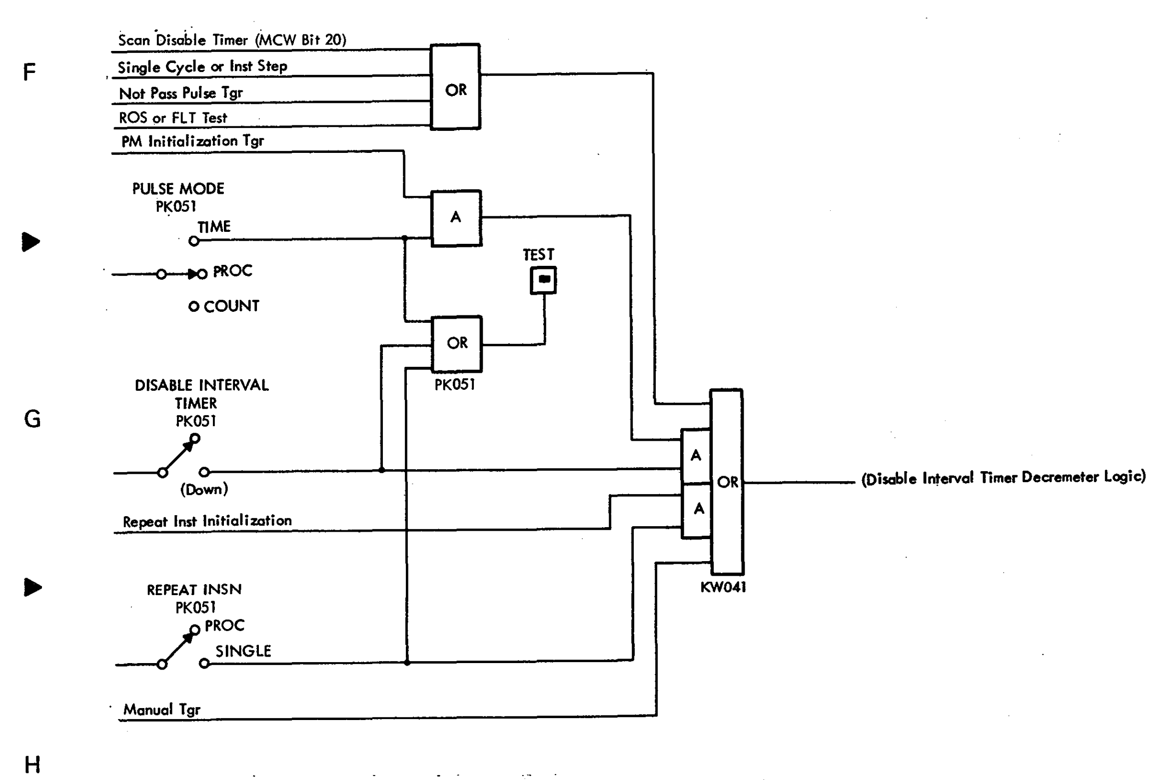


Diagram 6-21. Disable Interval Timer Logic

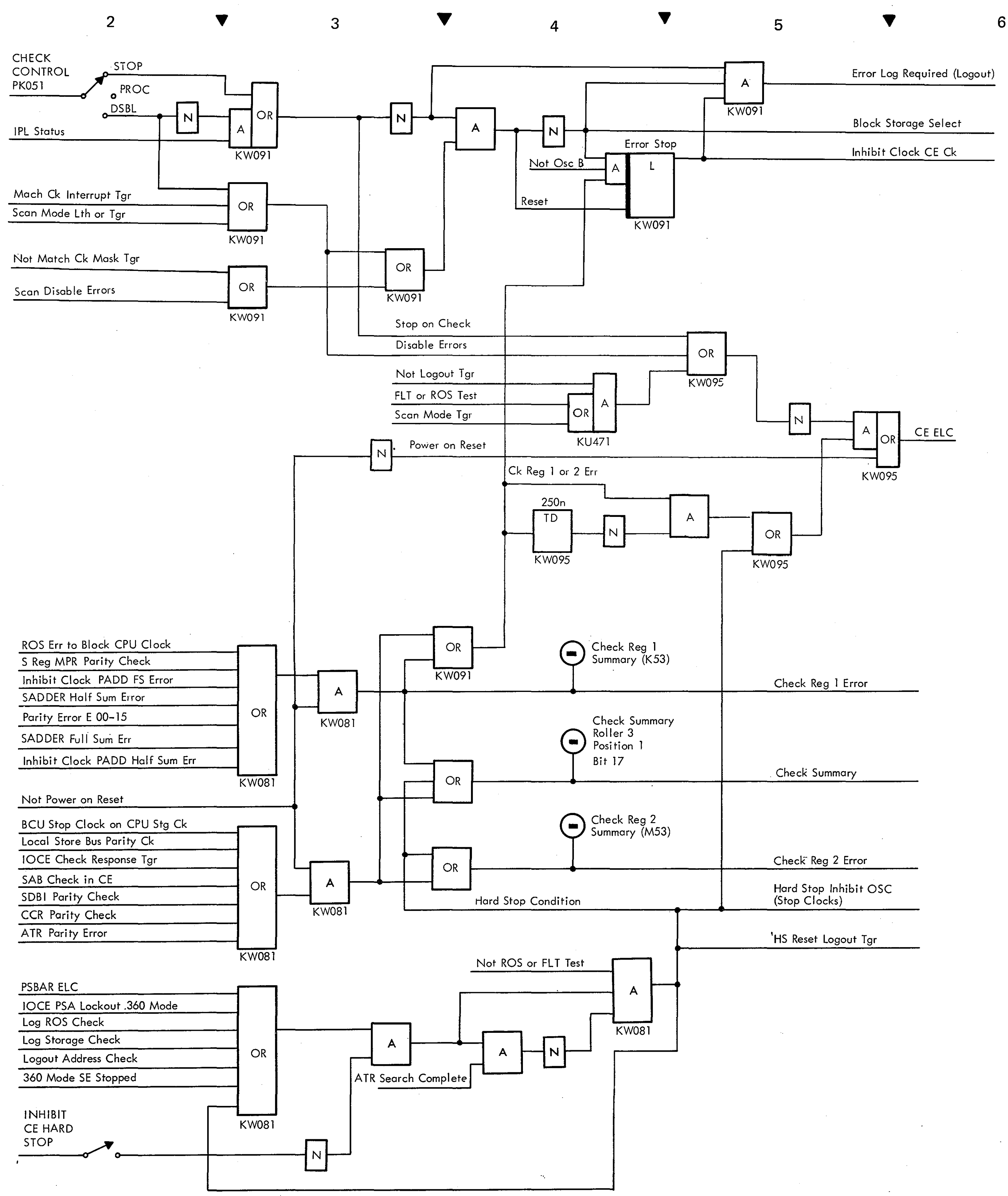


Diagram 6-22. CE Check Control and Inhibit CE Hardstop Switches, Logic and Error Controls

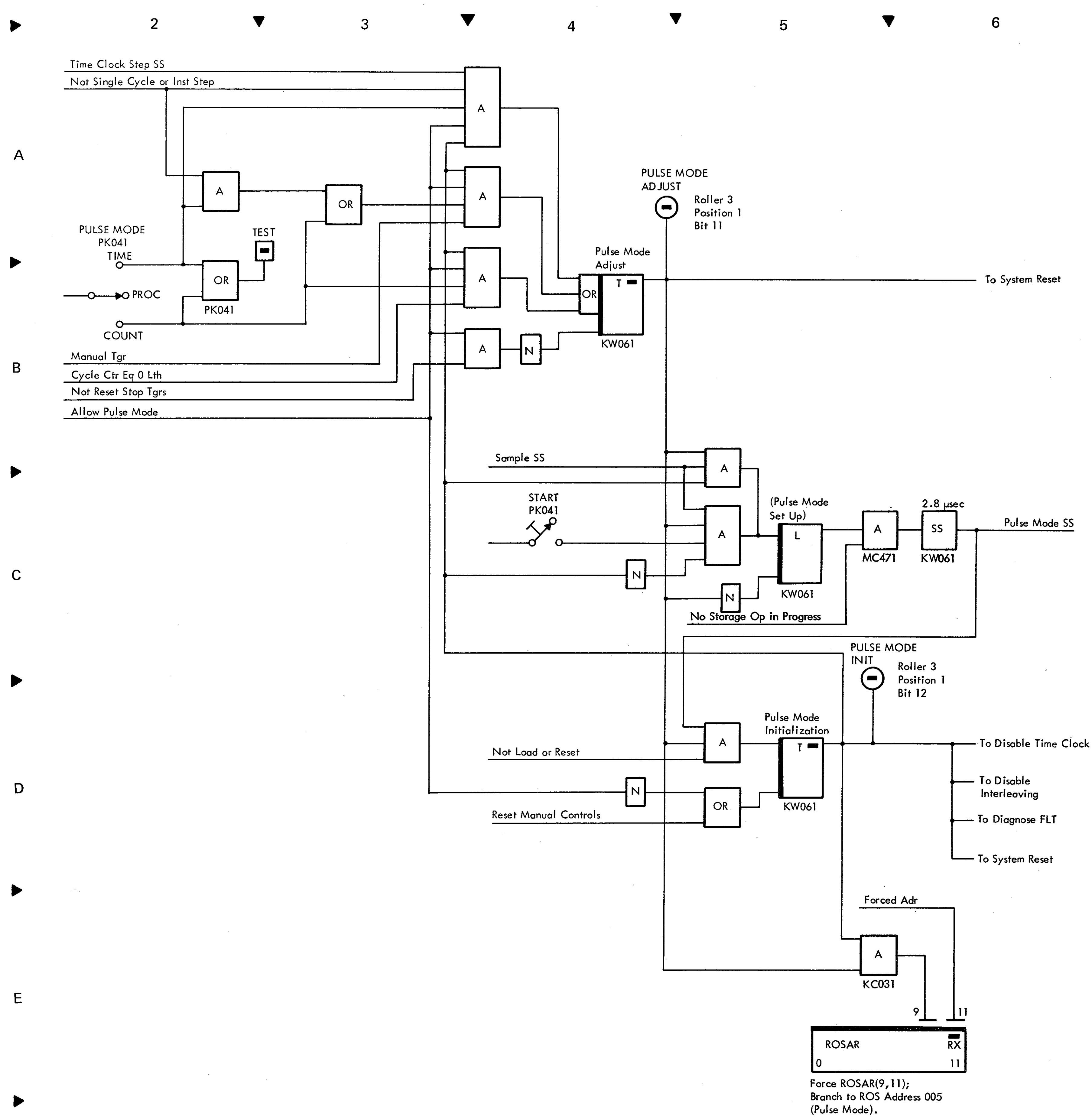


Diagram 6-23. Pulse Mode Controls

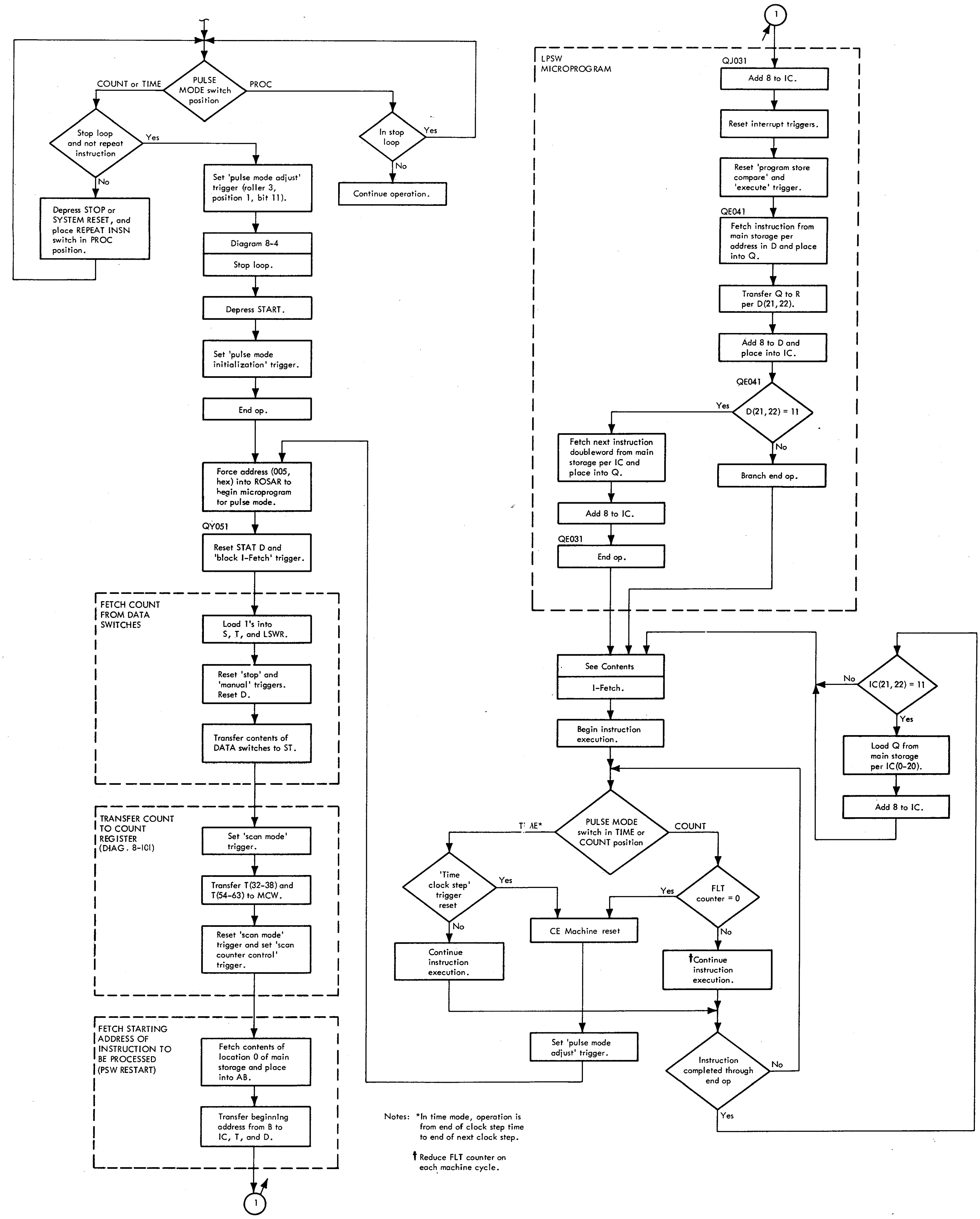


Diagram 6-24. Pulse Mode Operation

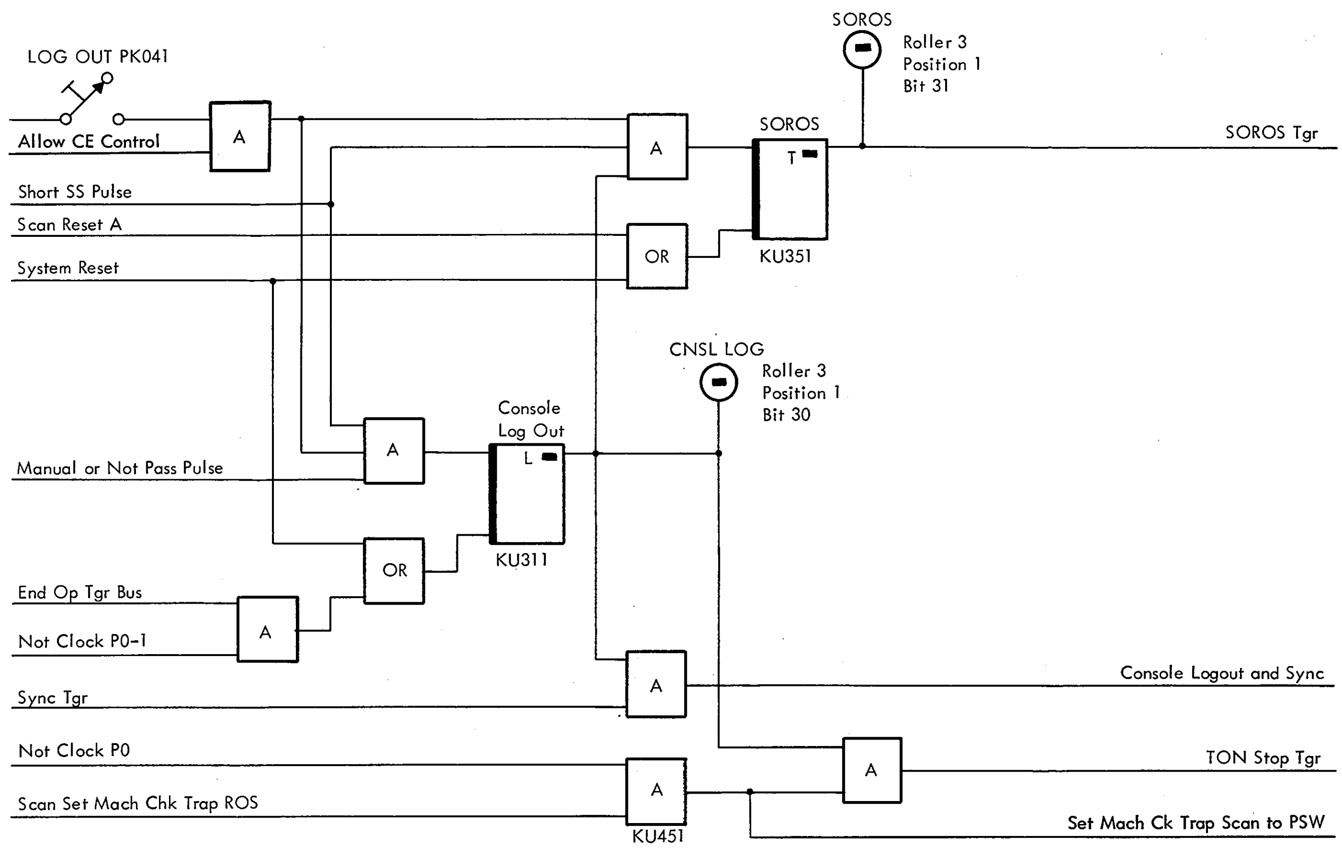


Diagram 6-25. LOG OUT Pushbutton Logic

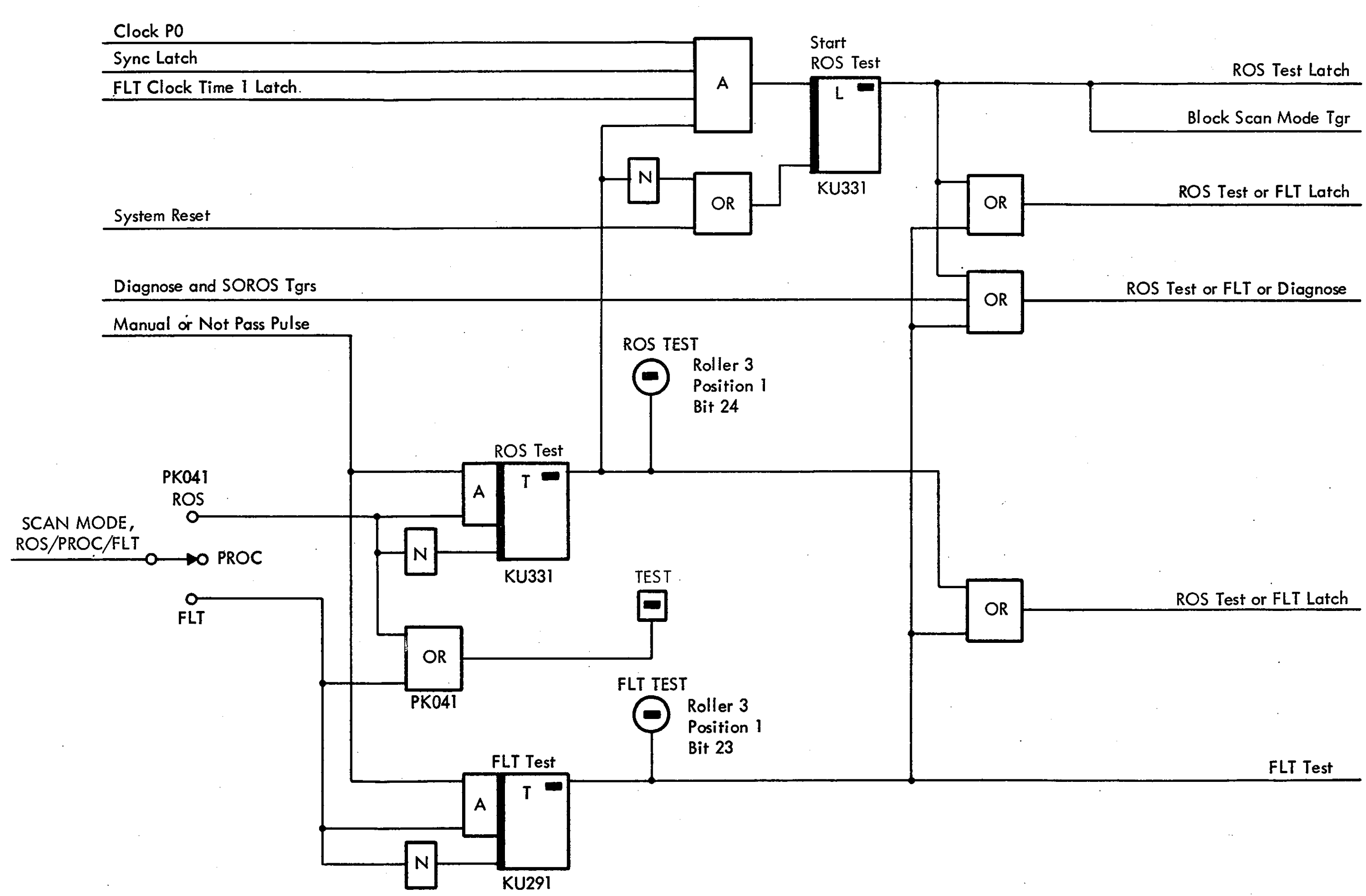


Diagram 6-26. SCAN MODE, ROS/PROC/FLT Switch Logic

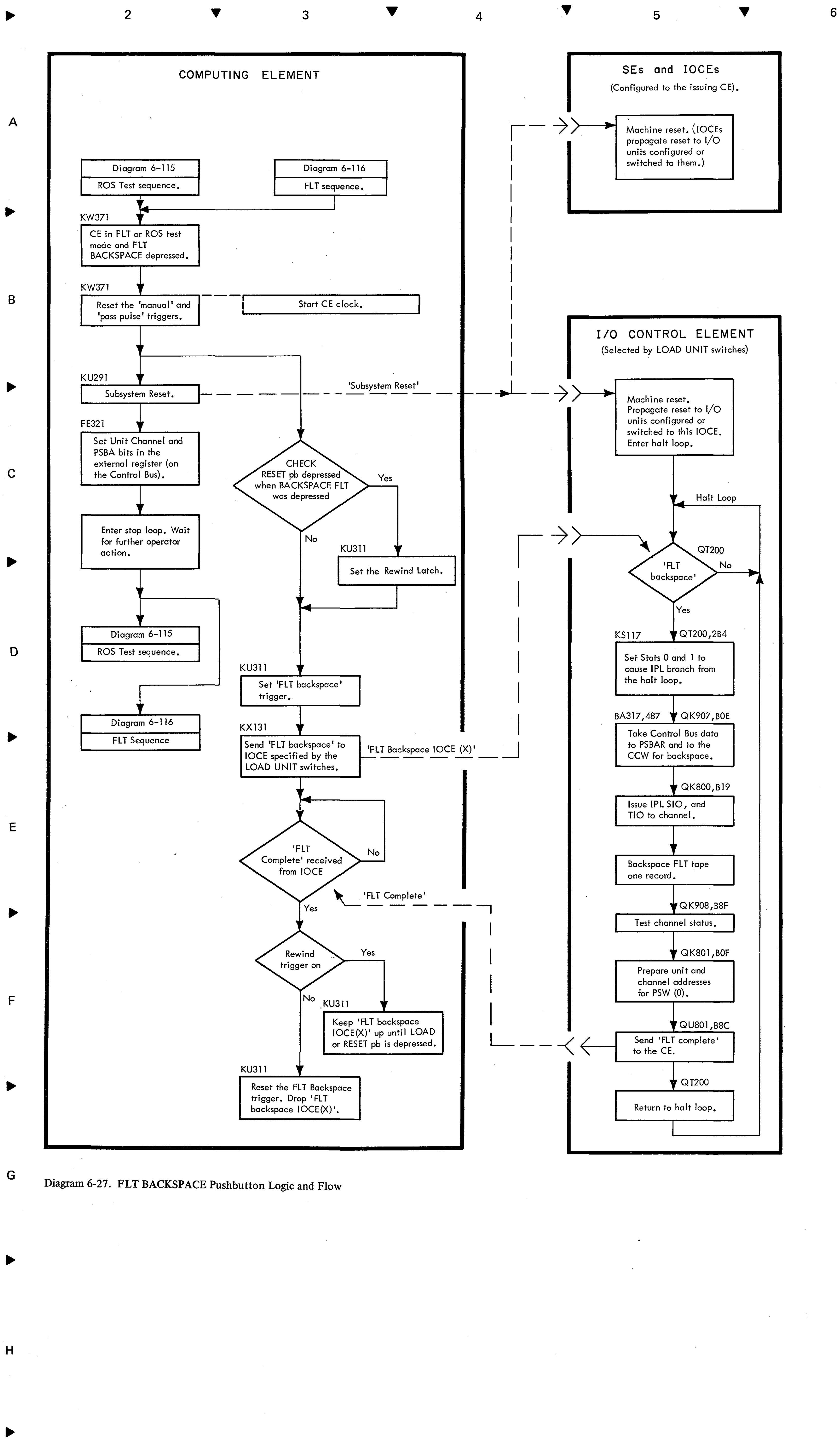


Diagram 6-27. FLT BACKSPACE Pushbutton Logic and Flow

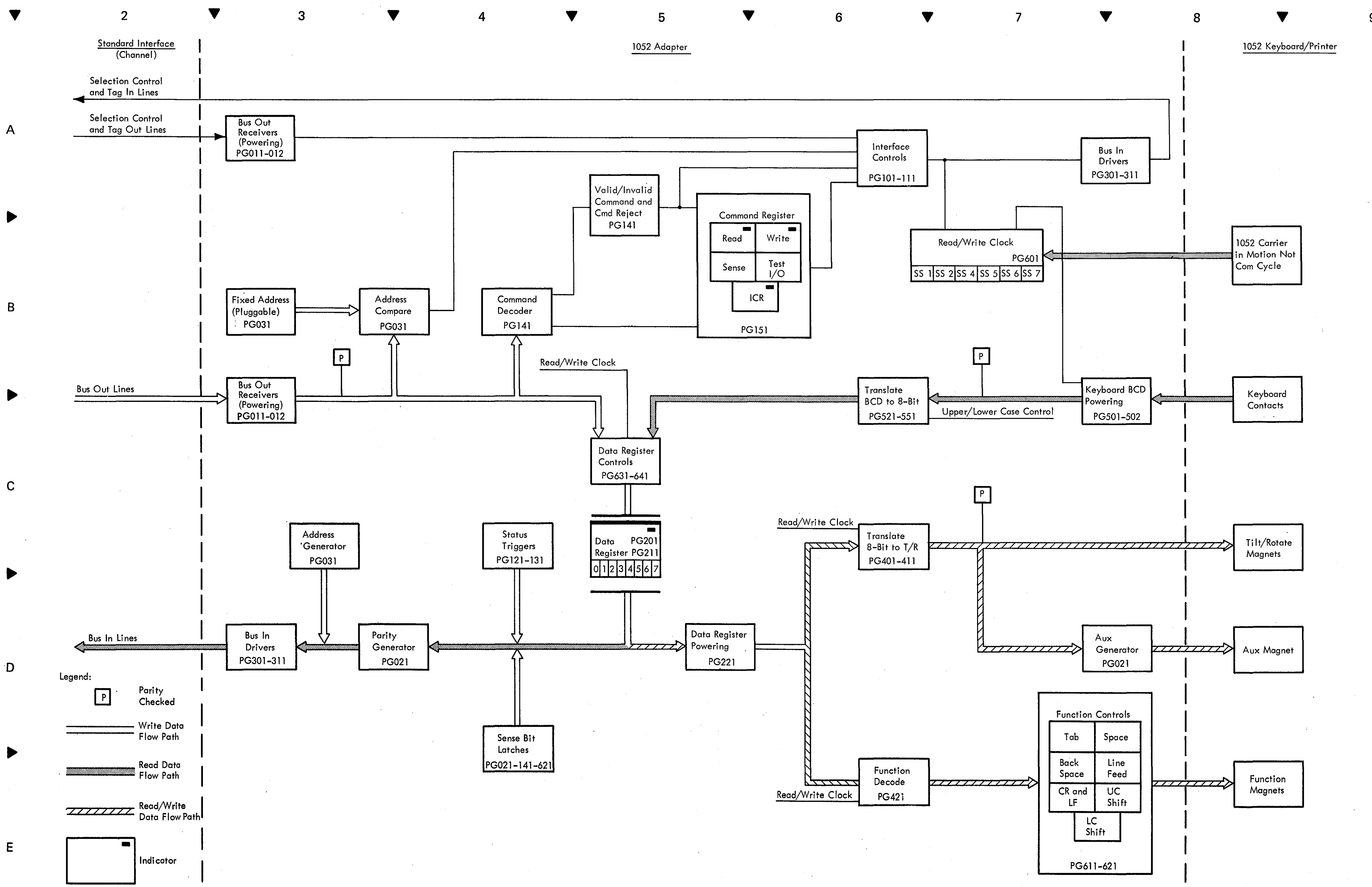
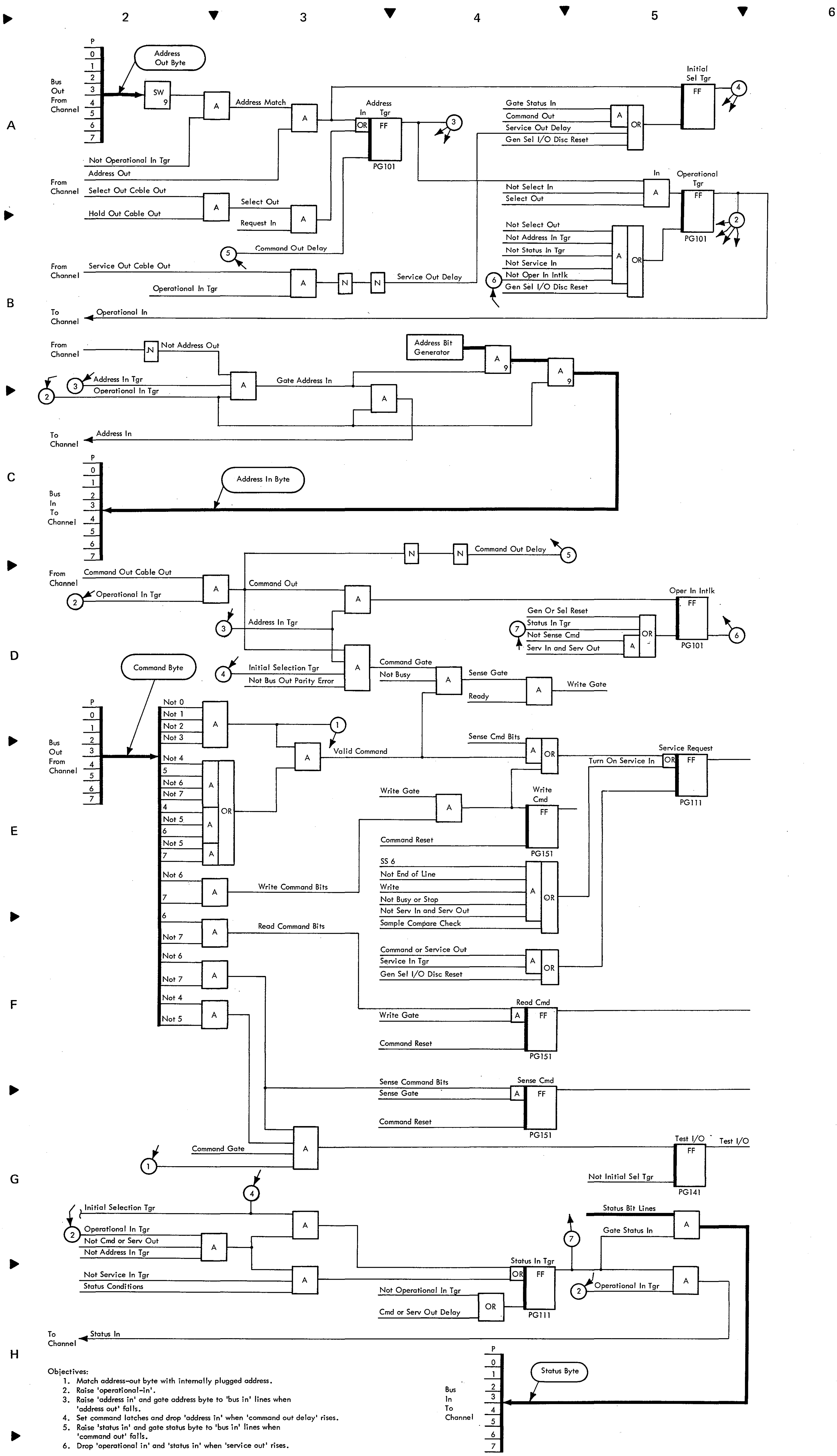


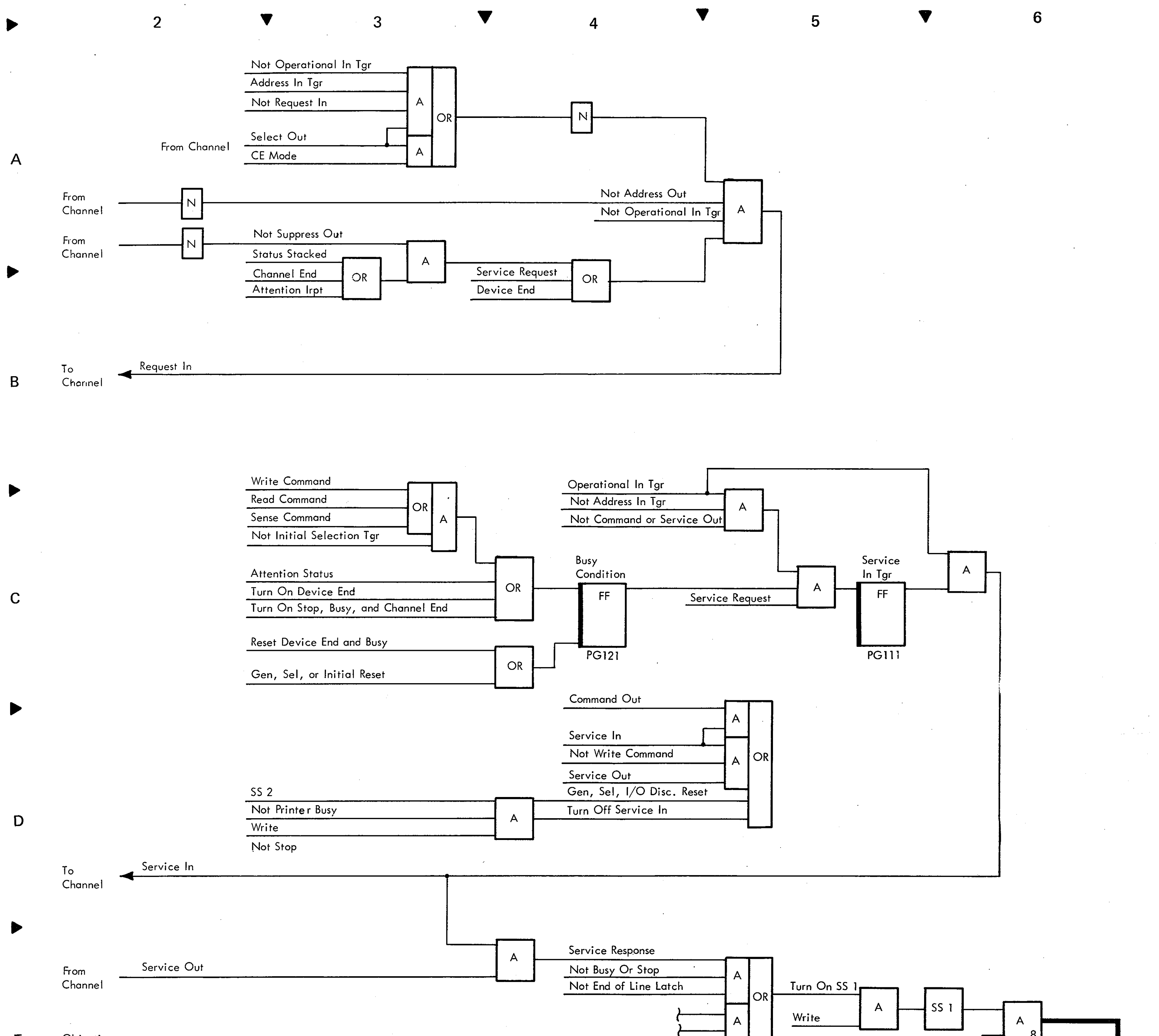
Diagram 6-28. 1052 Adapter Unit

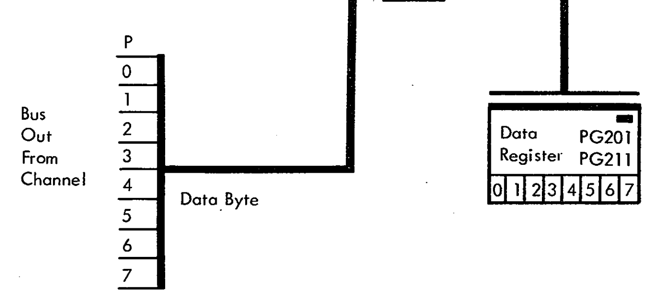


Objectives:

1. Match address-out byte with internally plugged address.
2. Raise 'operational-in'.
3. Raise 'address in' and gate address byte to 'bus in' lines when 'address out' falls.
4. Set command latches and drop 'address in' when 'command out delay' rises.
5. Raise 'status in' and gate status byte to 'bus in' lines when 'command out' falls.
6. Drop 'operational in' and 'status in' when 'service out' rises.

Diagram 6-29. 1052 Adapter Initial Selection – Read, Write, Sense





Objectives:

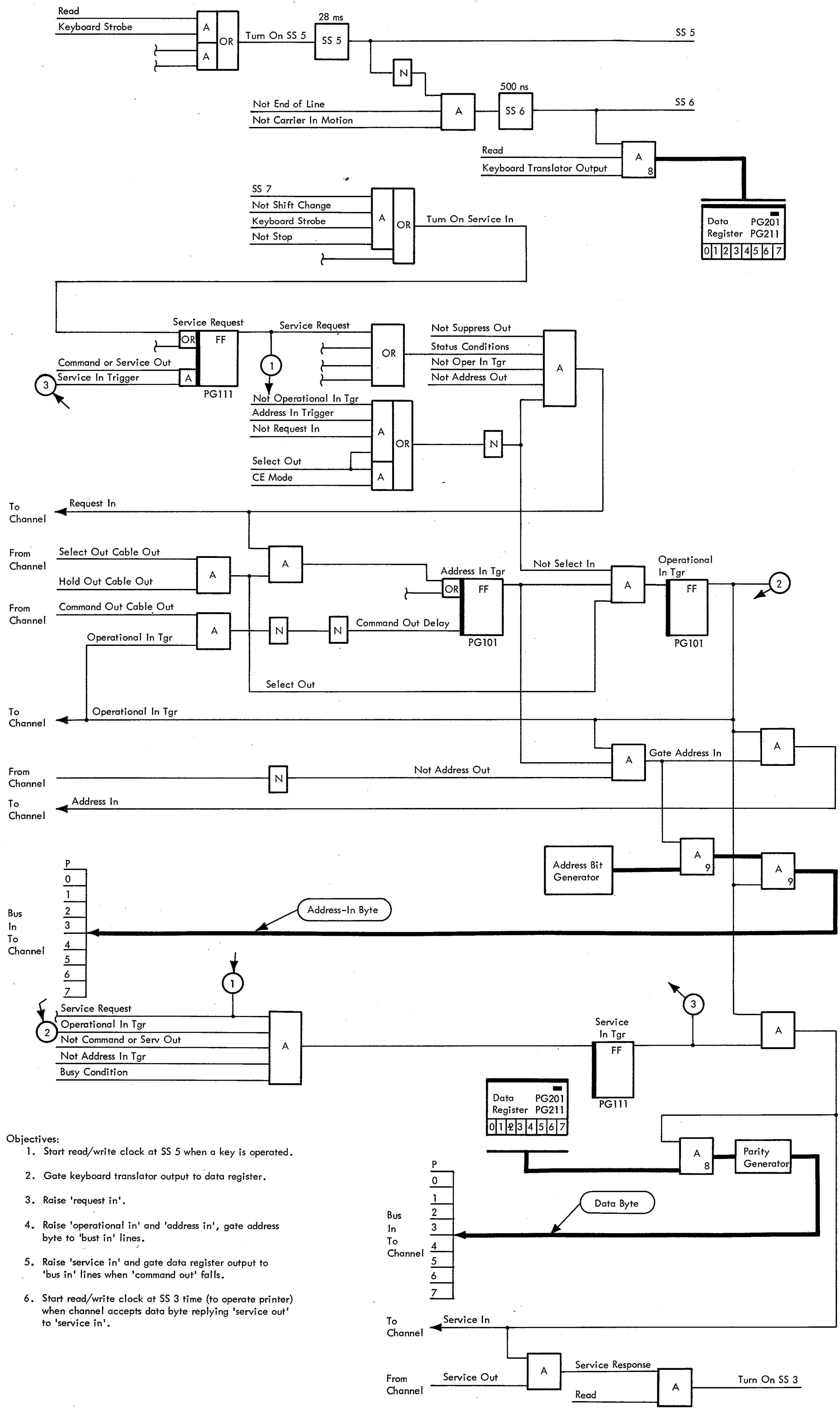
1. Raise request-in:
   A. Immediately after initial selection ('service request' latch, Figure 6-29)
   B. When the printer has finished printing the previous character, or when the printer has finished performing the function defined by the previous character ('service request' latch, Figure 6-29).
2. Raise 'service-in' when 'command out' line indicates "proceed".
3. Start read/write clock and gate 'bus out' lines to data register when 'service out' rises.
4. Repeat, starting with 1B, and continue until 'command out' line indicates "stop"

F

G

H

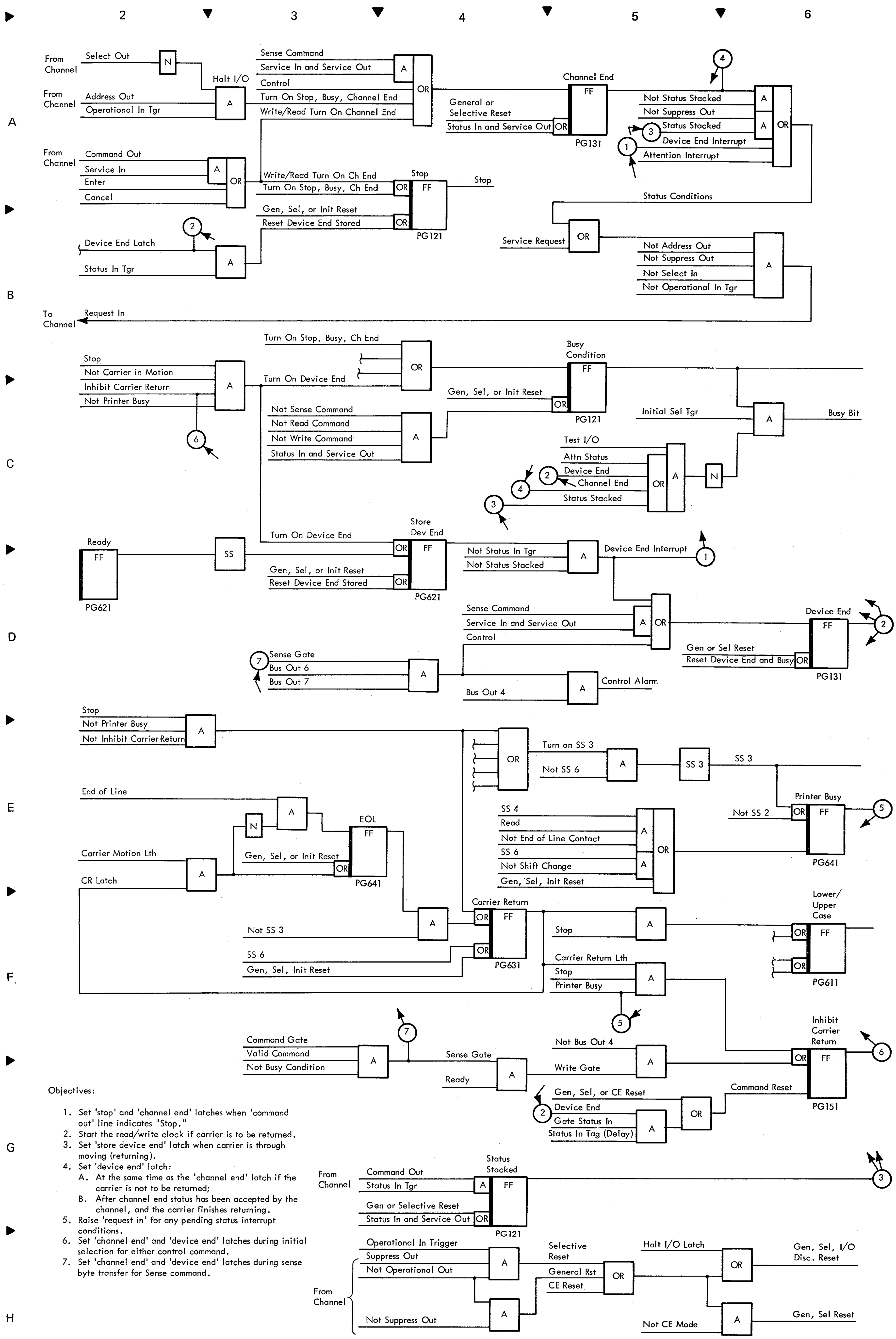
Diagram 6-30. 1052 Adapter Data Transfer – Write



Objectives:

1. Start read/write clock at SS 5 when a key is operated.
2. Gate keyboard translator output to data register.
3. Raise 'request in'.
4. Raise 'operational in' and 'address in', gate address byte to 'bust in' lines.
5. Raise 'service in' and gate data register output to 'bus in' lines when 'command out' falls.
6. Start read/write clock at SS 3 time (to operate printer) when channel accepts data byte replying 'service out' to 'service in'.

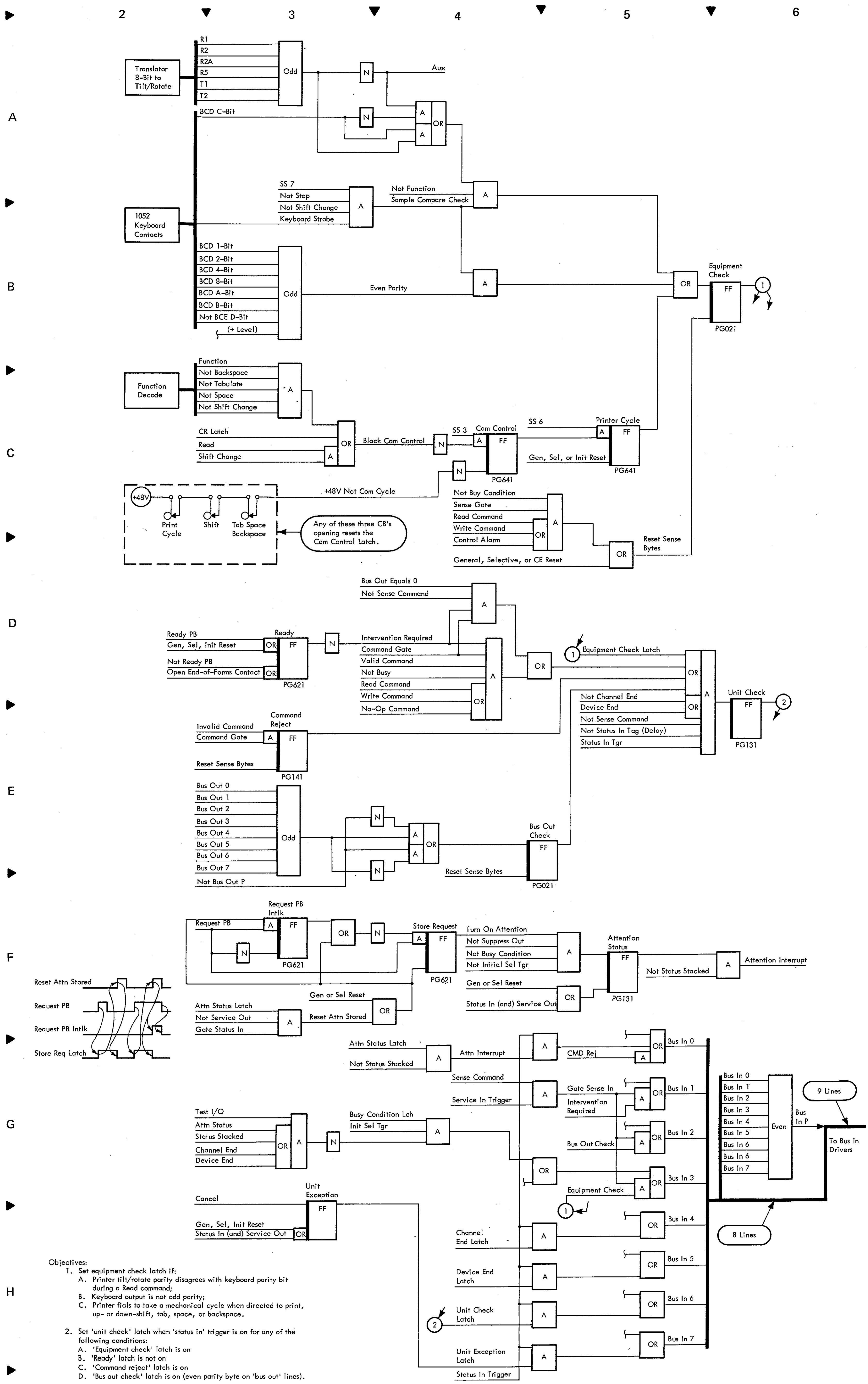
Diagram 6-31. 1052 Adapter Data Transfer – Read



Objectives:

1. Set 'stop' and 'channel end' latches when 'command out' line indicates "Stop."
2. Start the read/write clock if carrier is to be returned.
3. Set 'store device end' latch when carrier is through moving (returning).
4. Set 'device end' latch:
   - A. At the same time as the 'channel end' latch if the carrier is not to be returned;
   - B. After channel end status has been accepted by the channel, and the carrier finishes returning.
5. Raise 'request in' for any pending status interrupt conditions.
6. Set 'channel end' and 'device end' latches during initial selection for either control command.
7. Set 'channel end' and 'device end' latches during sense byte transfer for Sense command.

Diagram 6-32. 1052 Adapter Ending Sequence

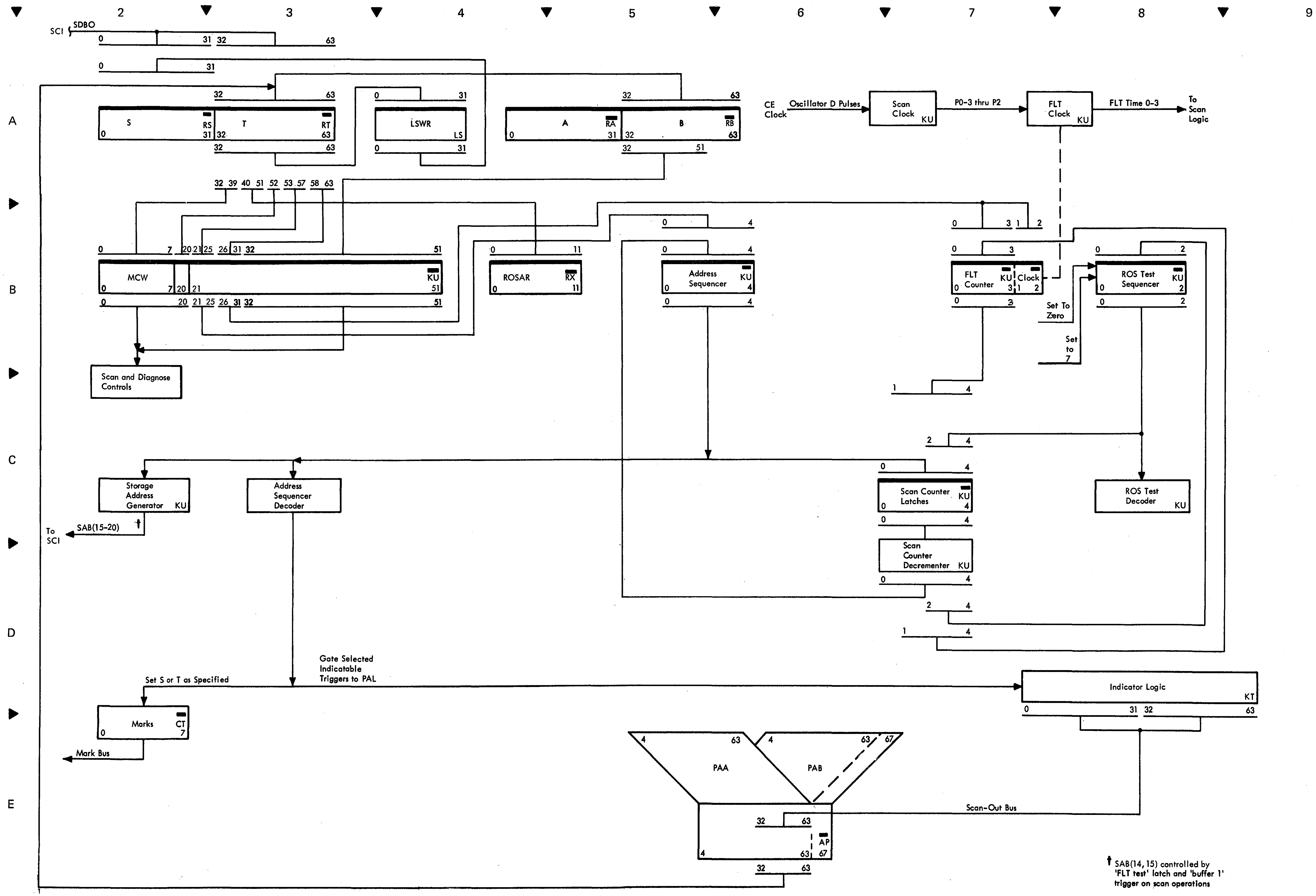


Objectives:

1. Set equipment check latch if:
   - A. Printer tilt/rotate parity disagrees with keyboard parity bit during a Read command;
   - B. Keyboard output is not odd parity;
   - C. Printer fials to take a mechanical cycle when directed to print, up- or down-shift, tab, space, or backspace.
2. Set 'unit check' latch when 'status in' trigger is on for any of the following conditions:
   - A. 'Equipment check' latch is on
   - B. 'Ready' latch is not on
   - C. 'Command reject' latch is on
   - D. 'Bus out check' latch is on (even parity byte on 'bus out' lines).

Diagram 6-33. 1052 Adapter Sense and Status Bytes

Diagram 6-101. Scan Data and Control



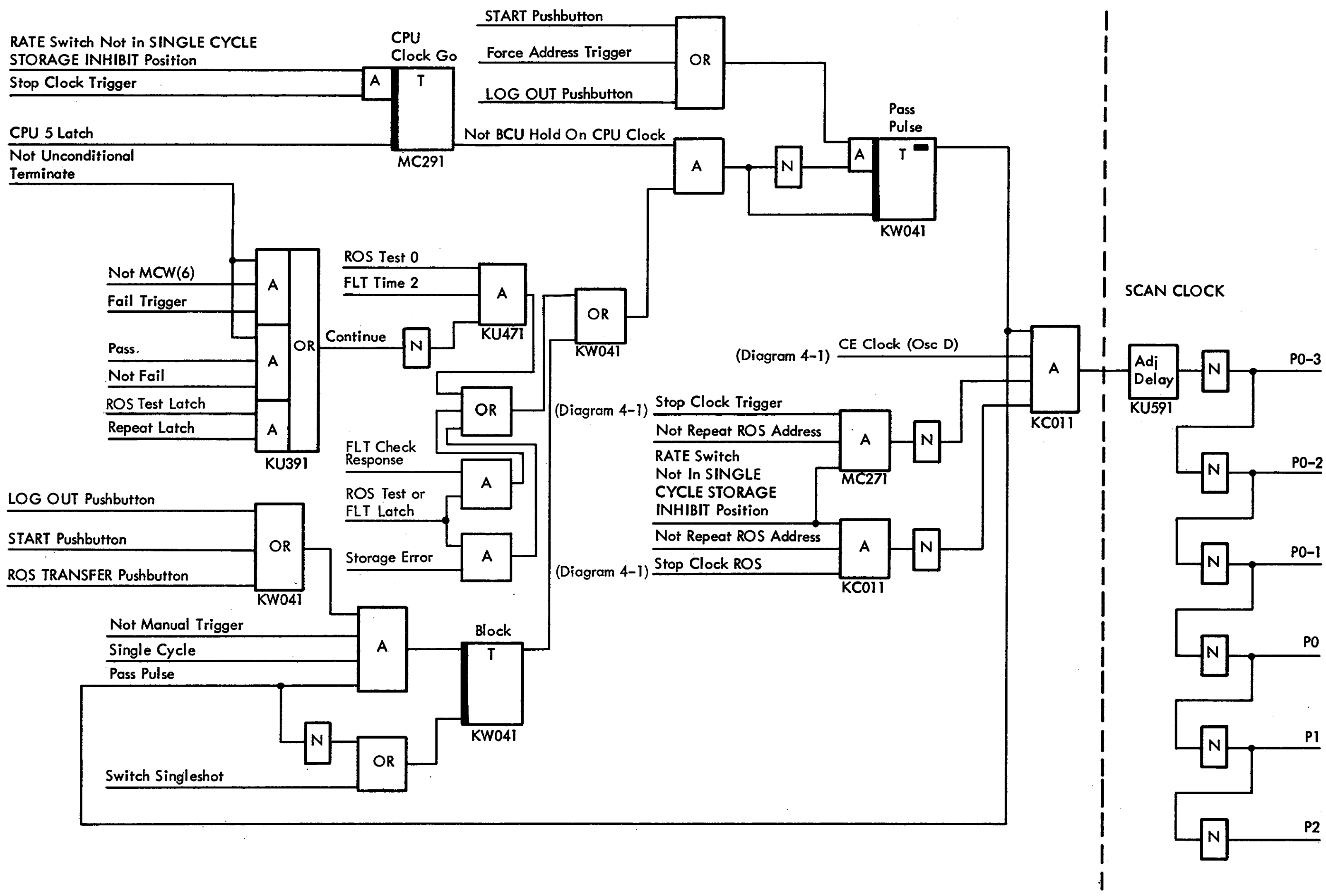
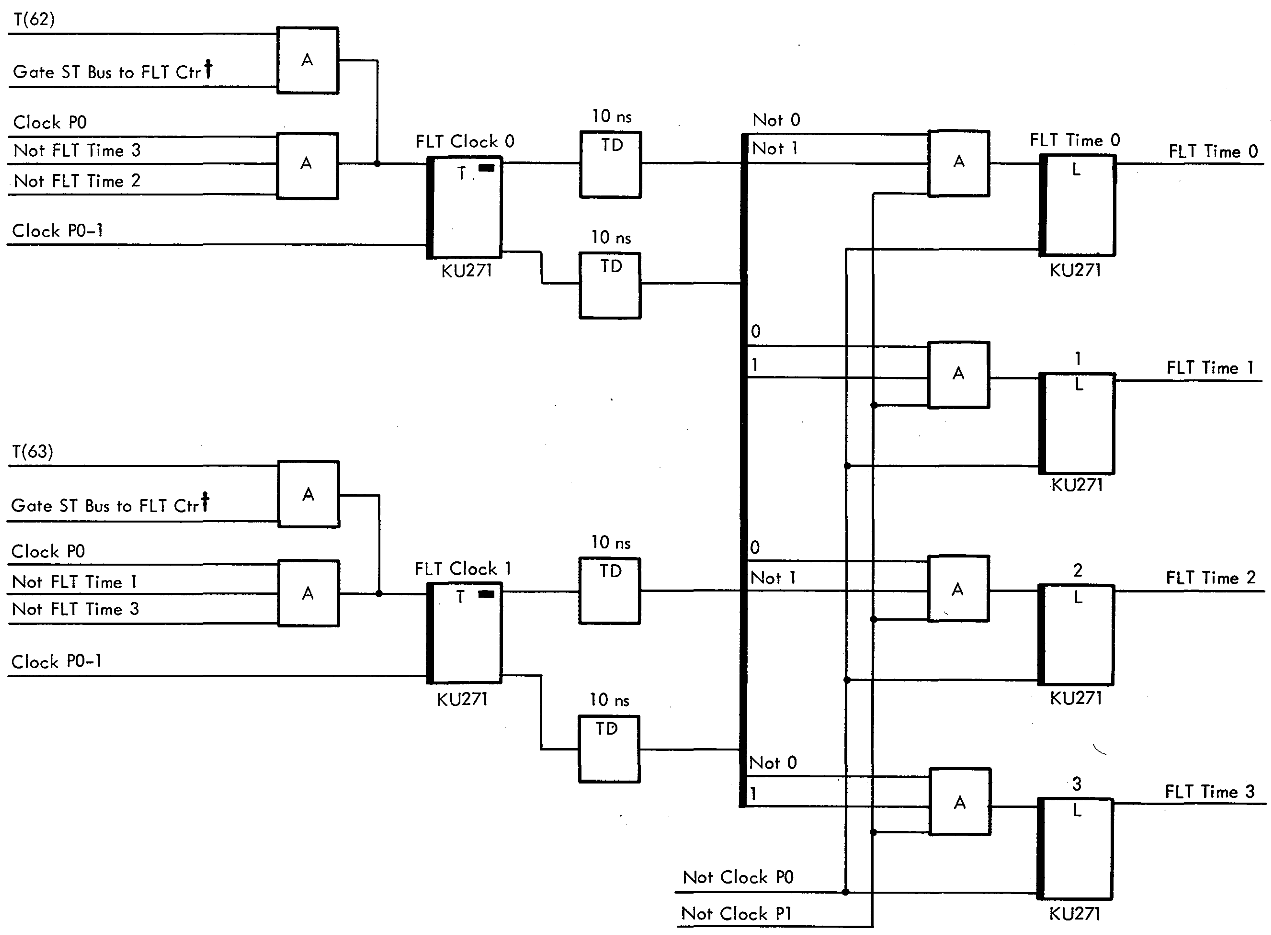
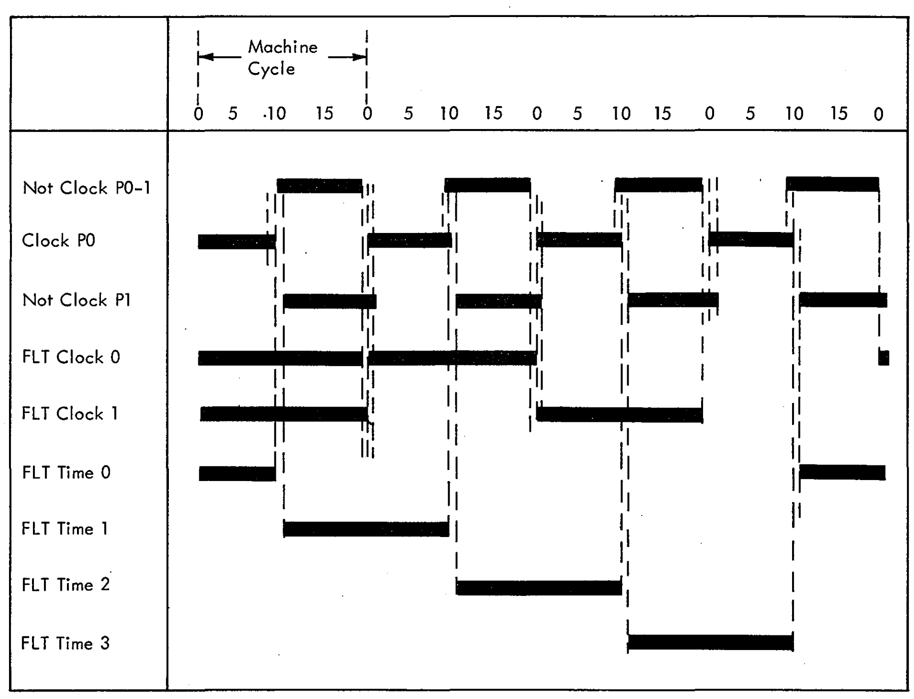


Diagram 6-102. Scan Clock





Note: All clock and not-clock signals to FLT clock are from scan clock.

† In the ALD, this signal is 'gate ST bus to scan ctr'.

Diagram 6-103. FLT Clock

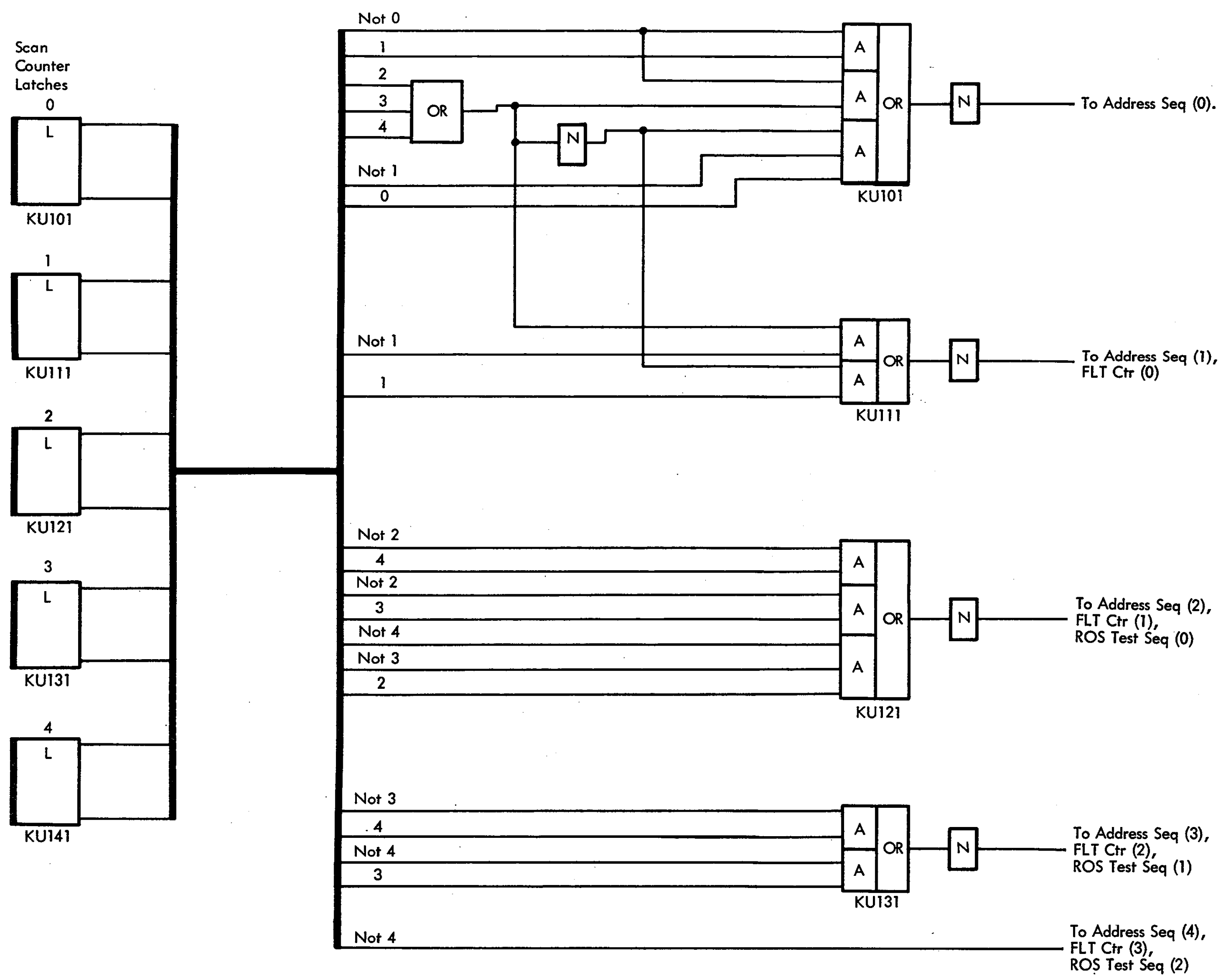
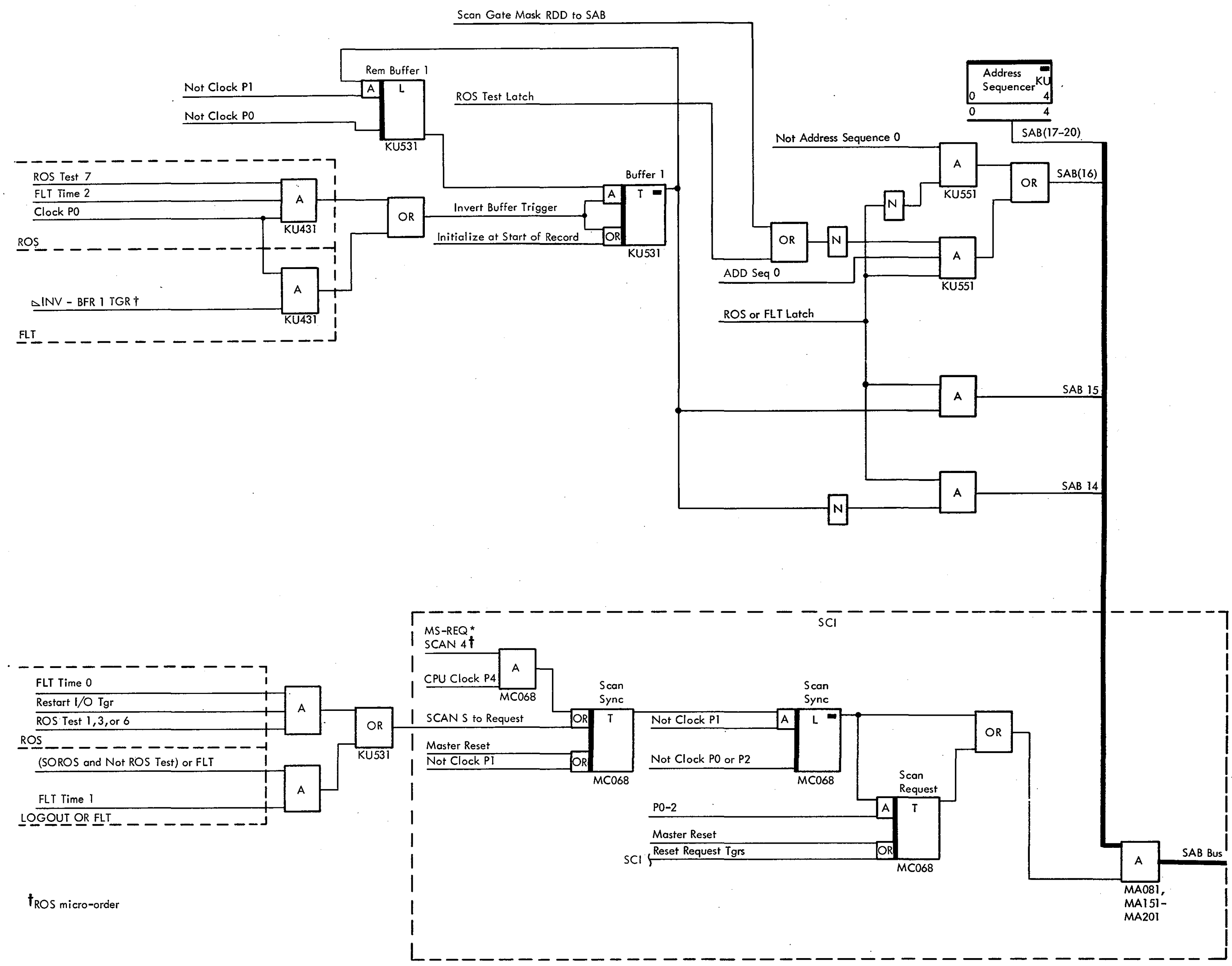


Diagram 6-104. Scan Counter Latches and Decrementer



	SAB																							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Logout 80	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
Logout F8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0
Logout 100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
Logout 138	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0	0	0
																Not Address Sequence 0		Address Sequencer						
Buffer 1:																								
FLT's 100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
FLT's 180	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
Buffer 2:																								
FLT's 200	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
FLT's 280	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
															OFF	ON	Address Sequencer							
															Buffer 1 Trigger									

Note: Scan-generated main storage addresses start with the highest value required and are decremented to the lowest value required.

Diagram 6-105. Scan Storage Address Generator

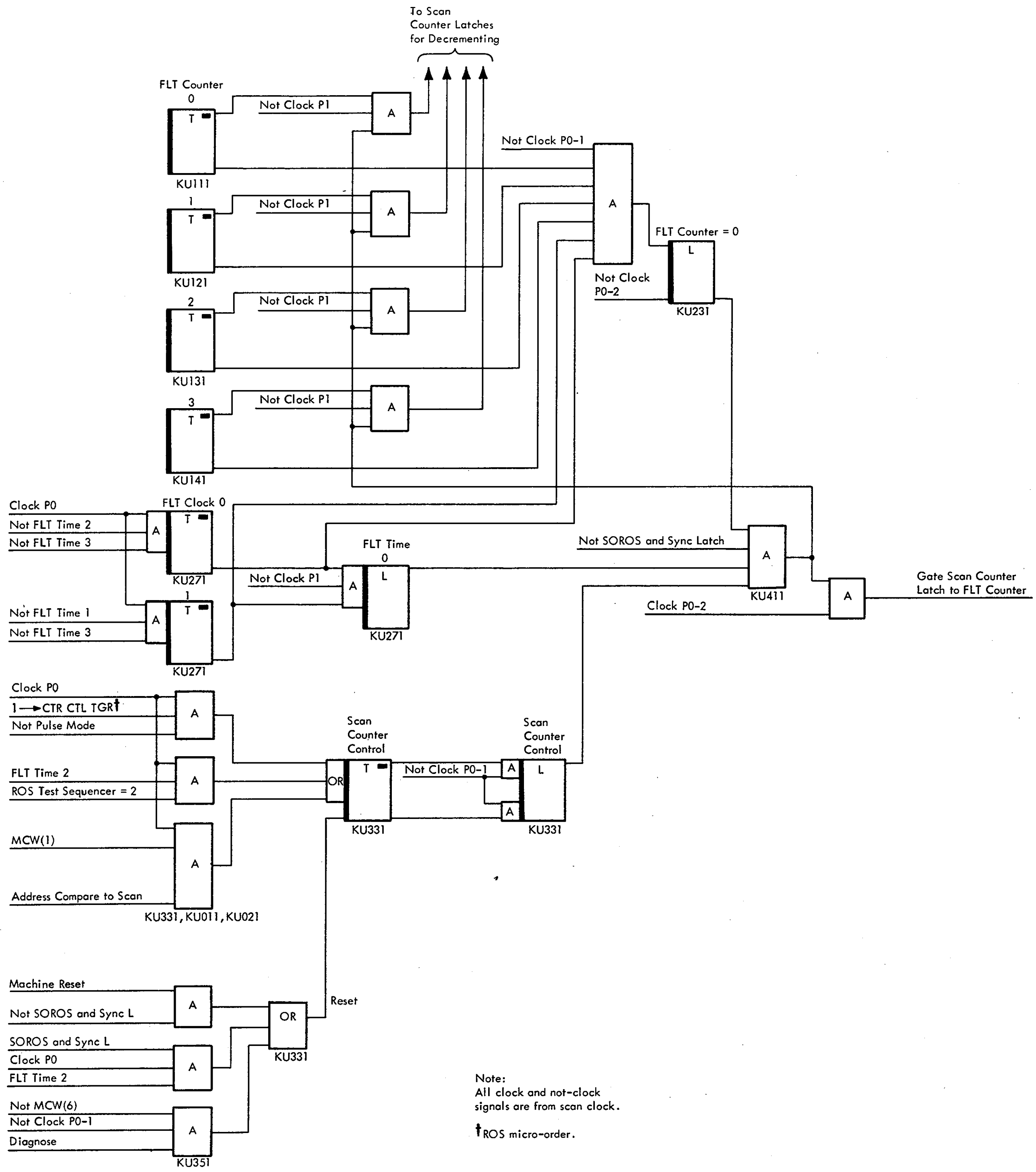


Diagram 6-106. FLT Counter Decrementing

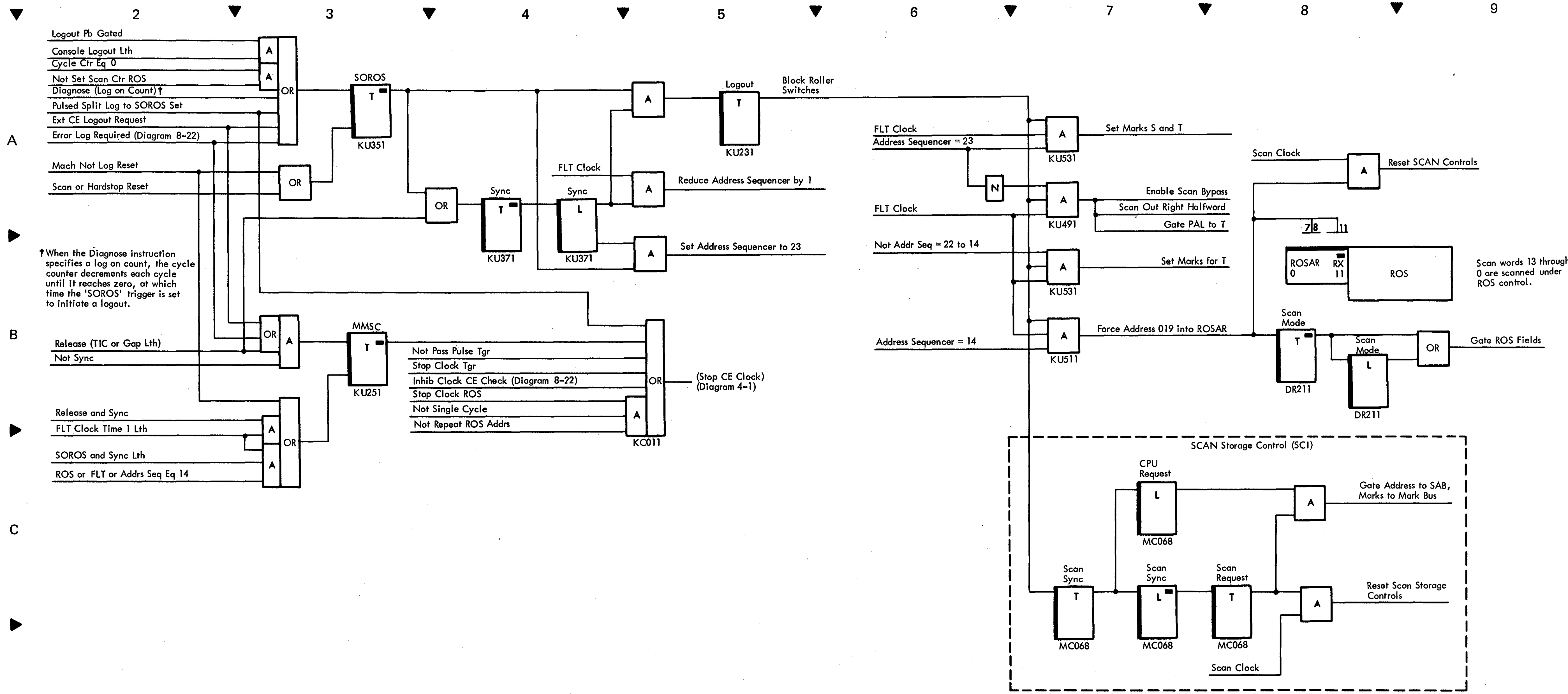
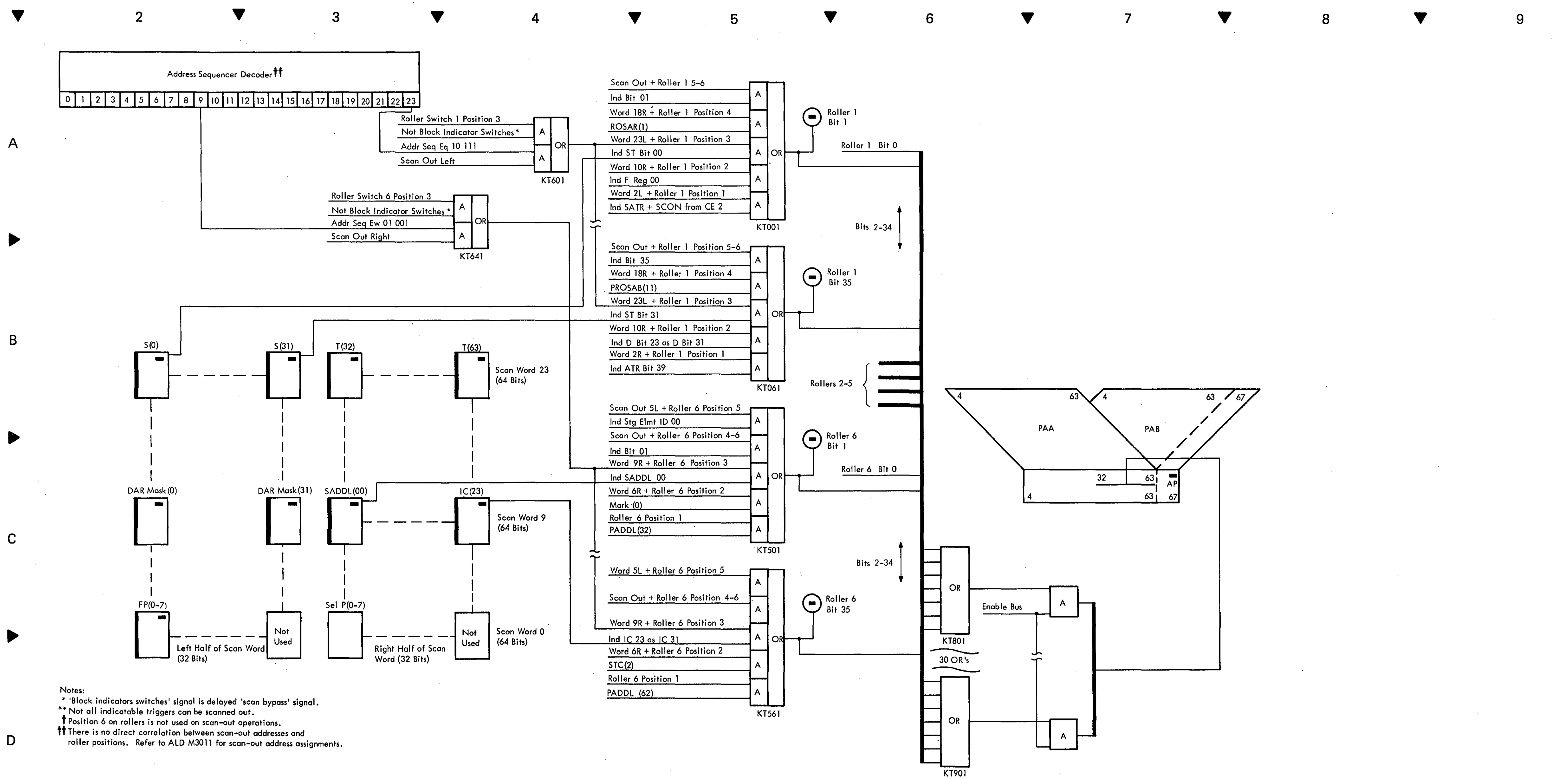
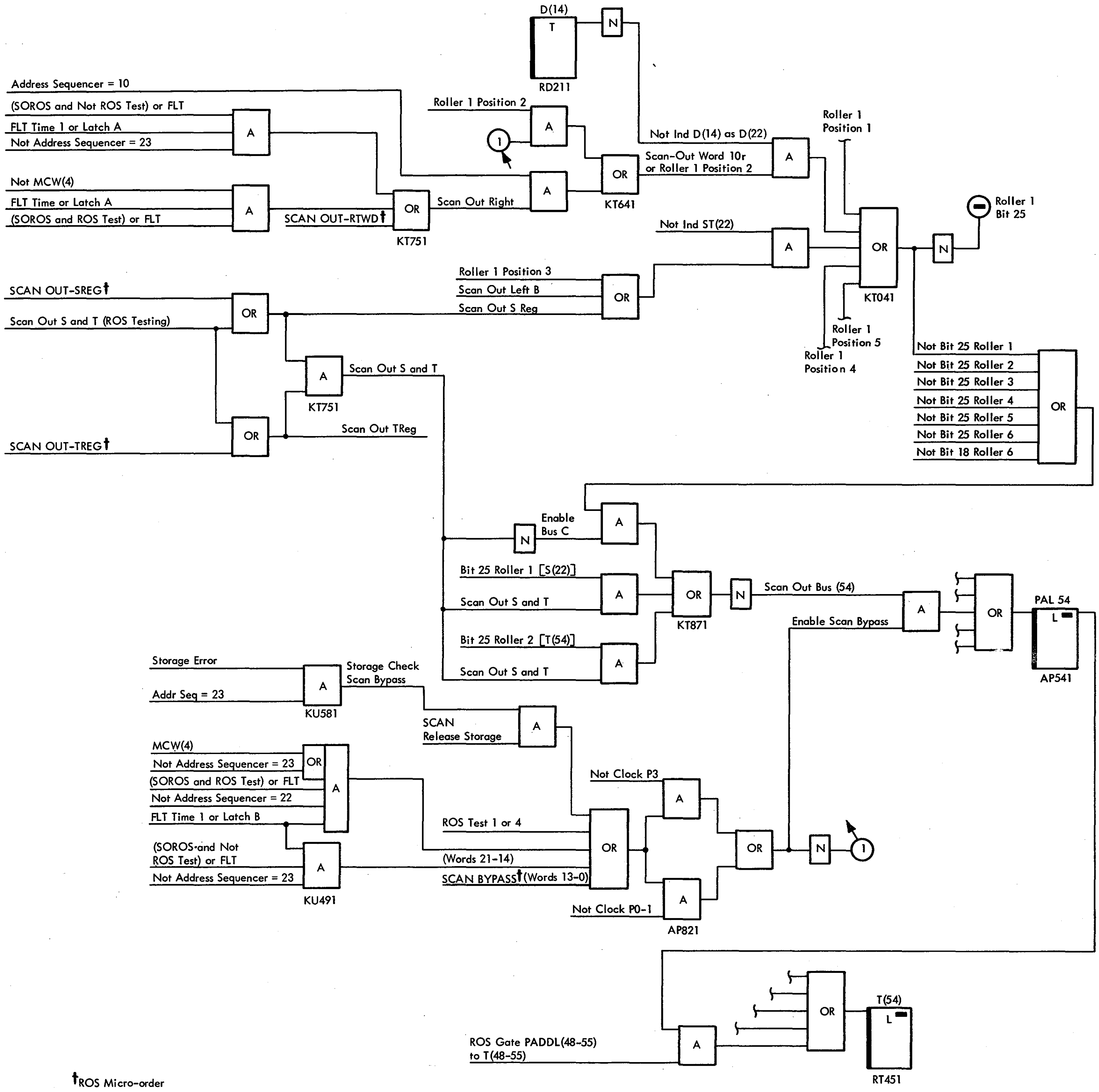


Diagram 6-108. Logout Control Logic



Notes:
* 'Block indicators switches' signal is delayed 'scan bypass' signal.
** Not all indicatable triggers can be scanned out.
† Position 6 on rollers is not used on scan-out operations.
†† There is no direct correlation between scan-out addresses and roller positions. Refer to ALD M3011 for scan-out address assignments.

Diagram 6-107. Scan-Out Bus Data Flow



†ROS Micro-order

Diagram 6-109. Scan-Out Path For One Bit

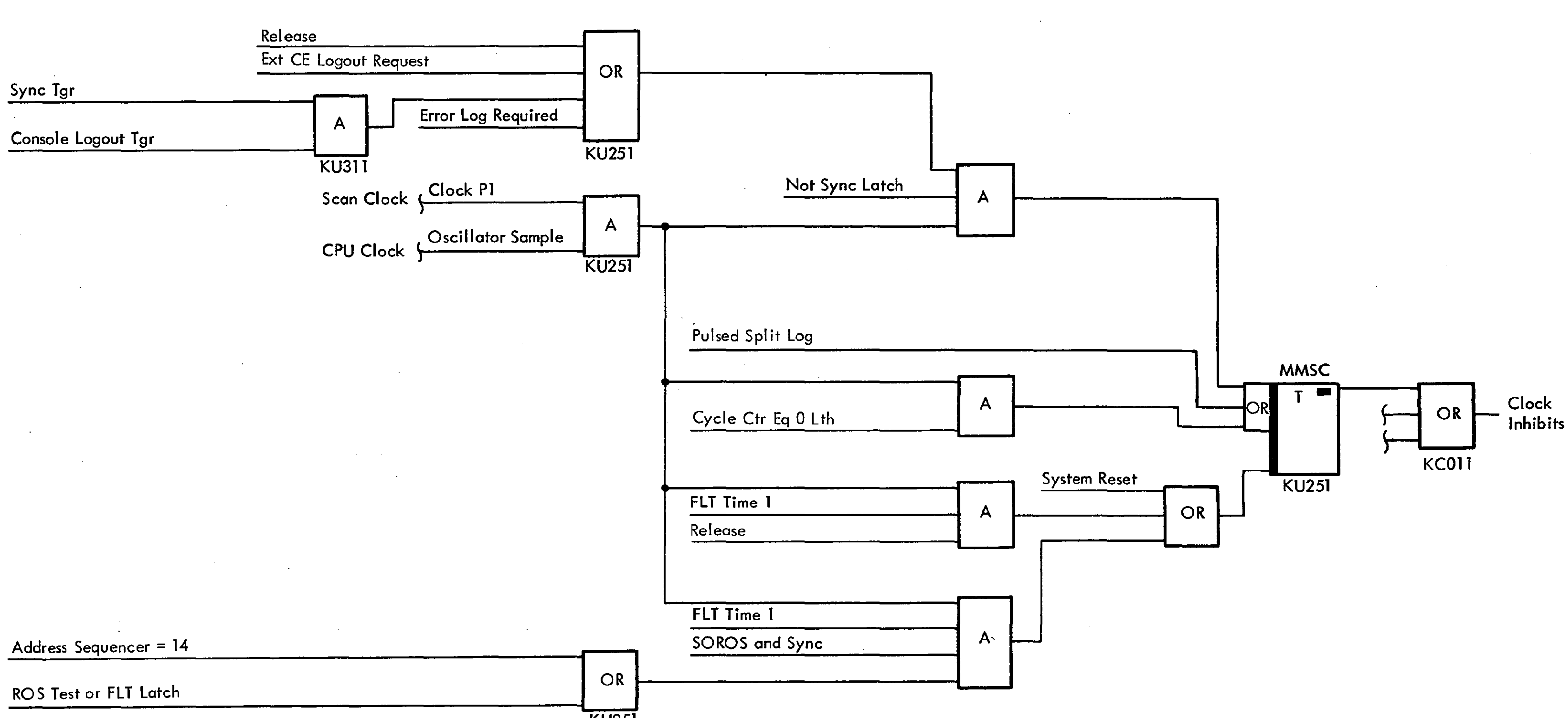
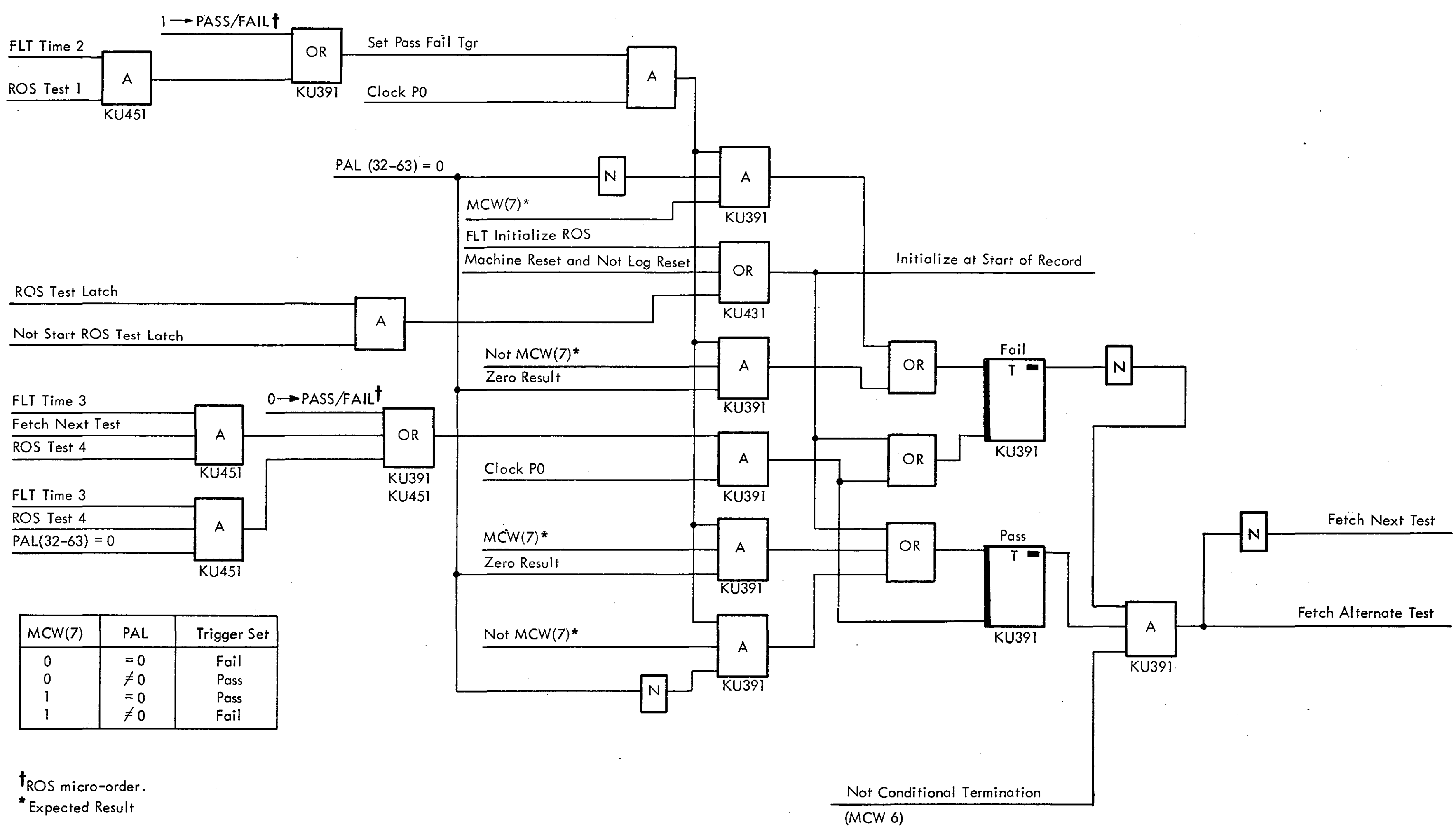


Diagram 6-110. Maintenance Mode Stop Clock Logic



MCW(7)	PAL	Trigger Set
0	=0	Fail
0	≠0	Pass
1	=0	Pass
1	≠0	Fail

†ROS micro-order.
*Expected Result

Diagram 6-111. Scan Control Triggers

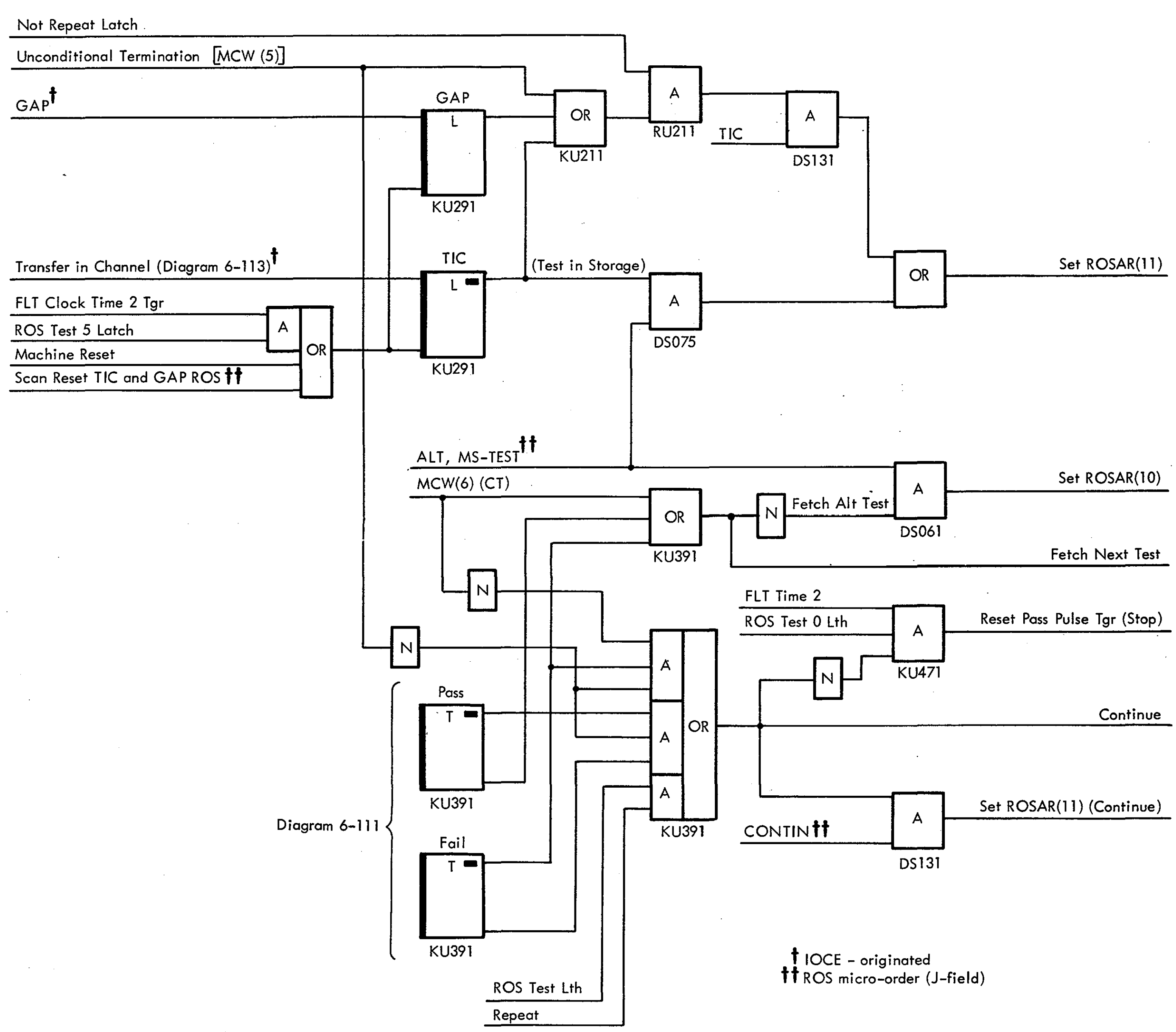


Diagram 6-112. Scan Control of ROS Microbranching

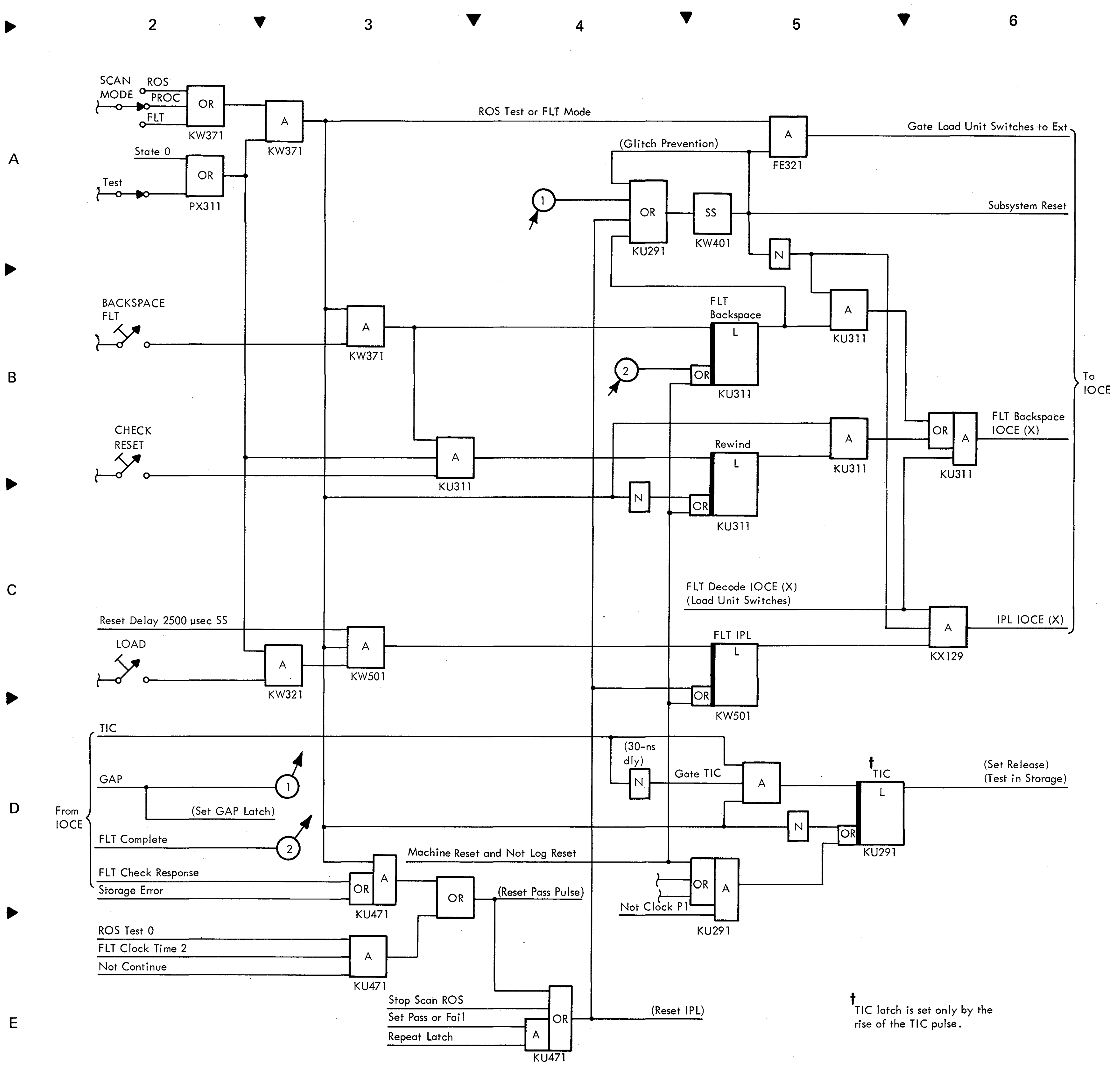


Diagram 6-113. CE Scan/IOCE Interface

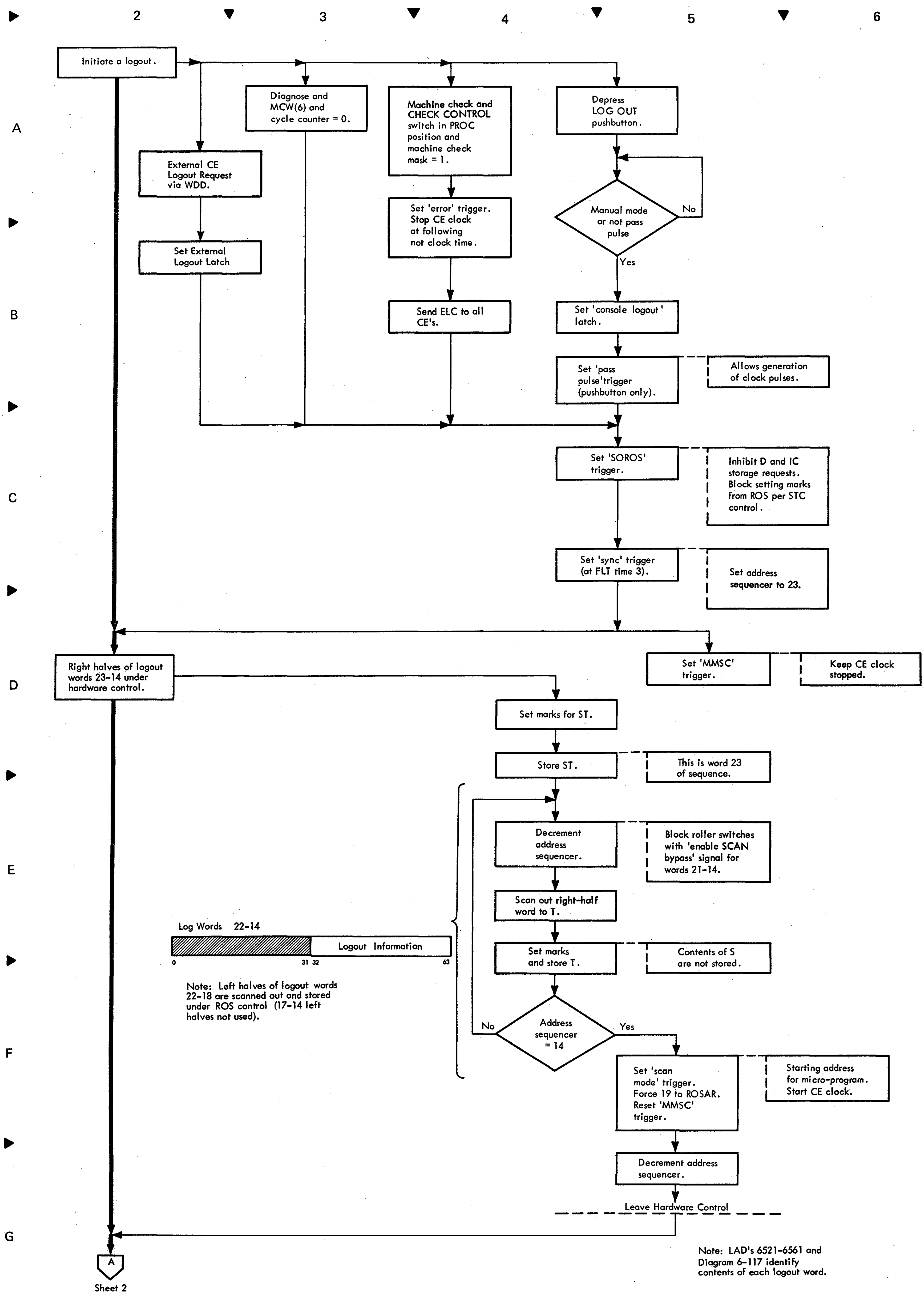


Diagram 6-114. Logout Sequence (Sheet 1 of 2)

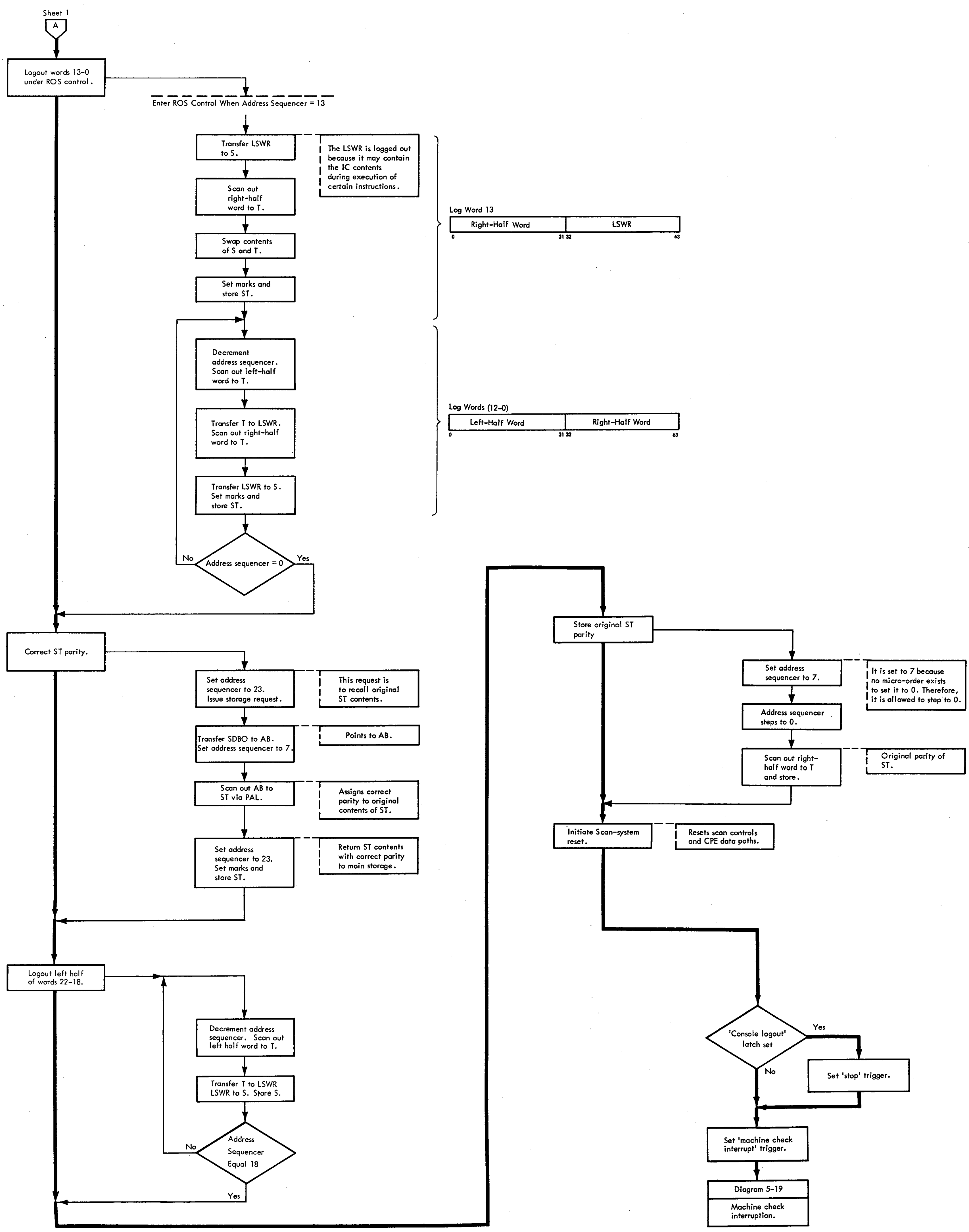


Diagram 6-114. Logout Sequence (Sheet 2 of 2)

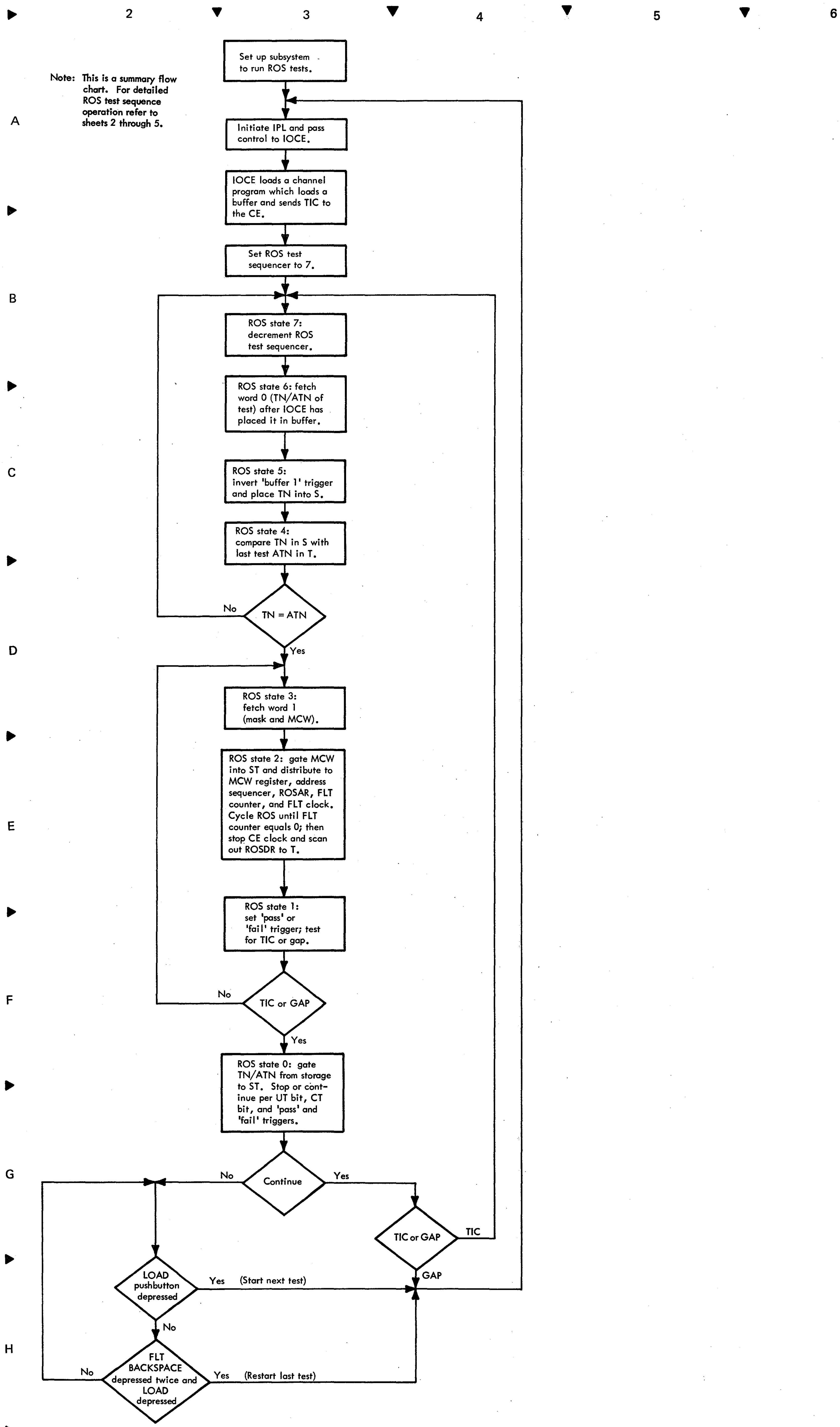


Diagram 6-115. ROS Test Sequence (Sheet 1 of 5)

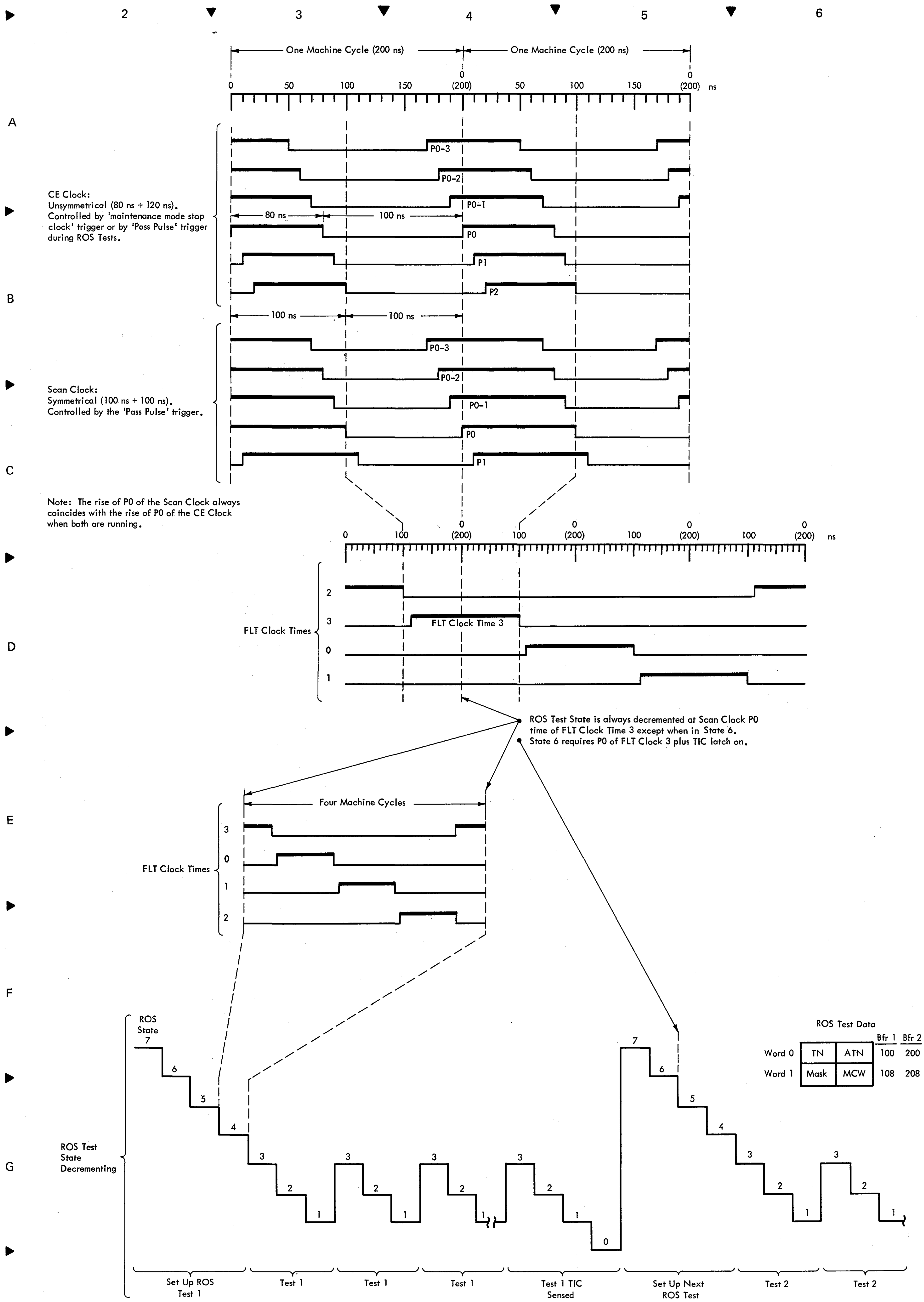


Diagram 6-115. ROS Test Sequence (Sheet 2 of 5)

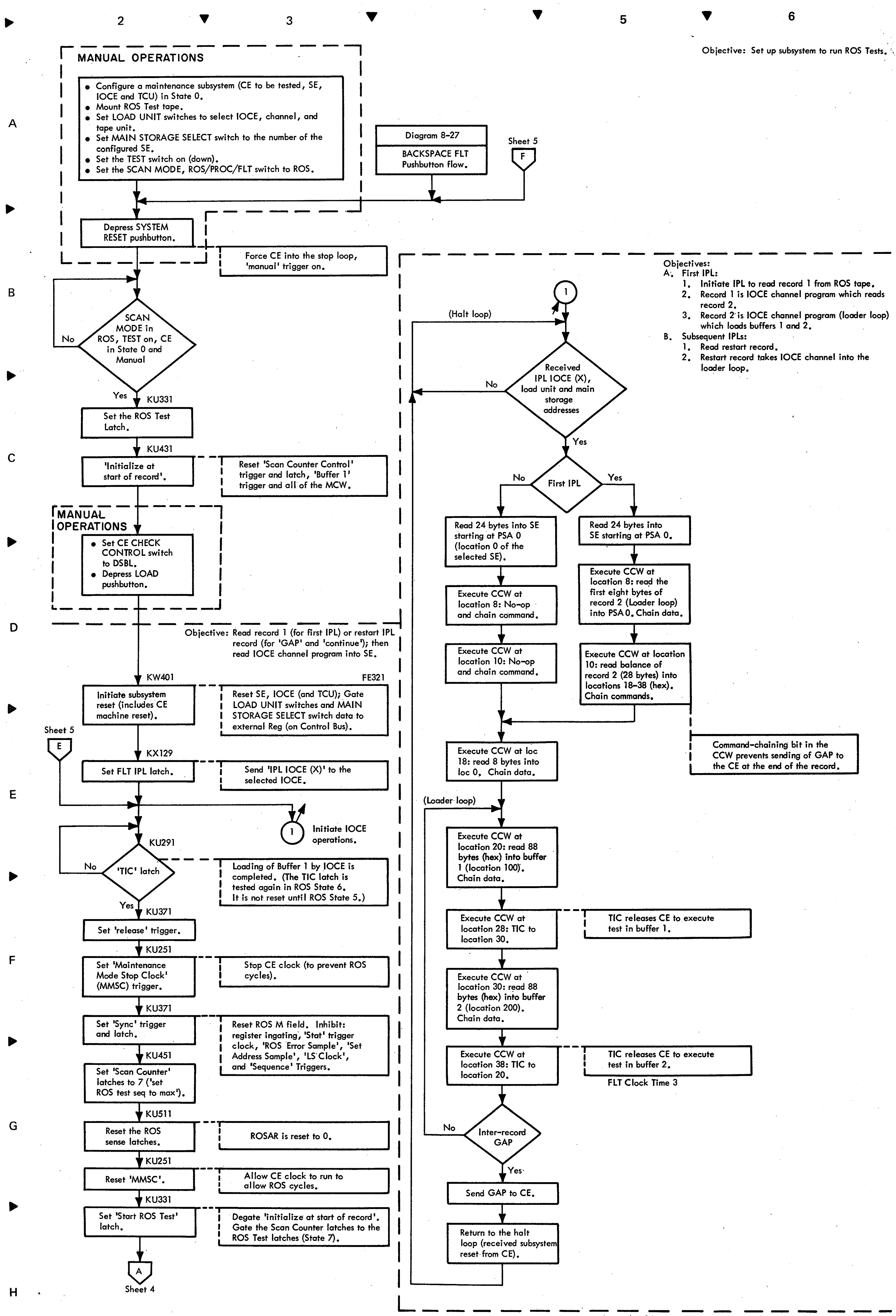


Diagram 6-115. ROS Test Sequence (Sheet 3 of 5)

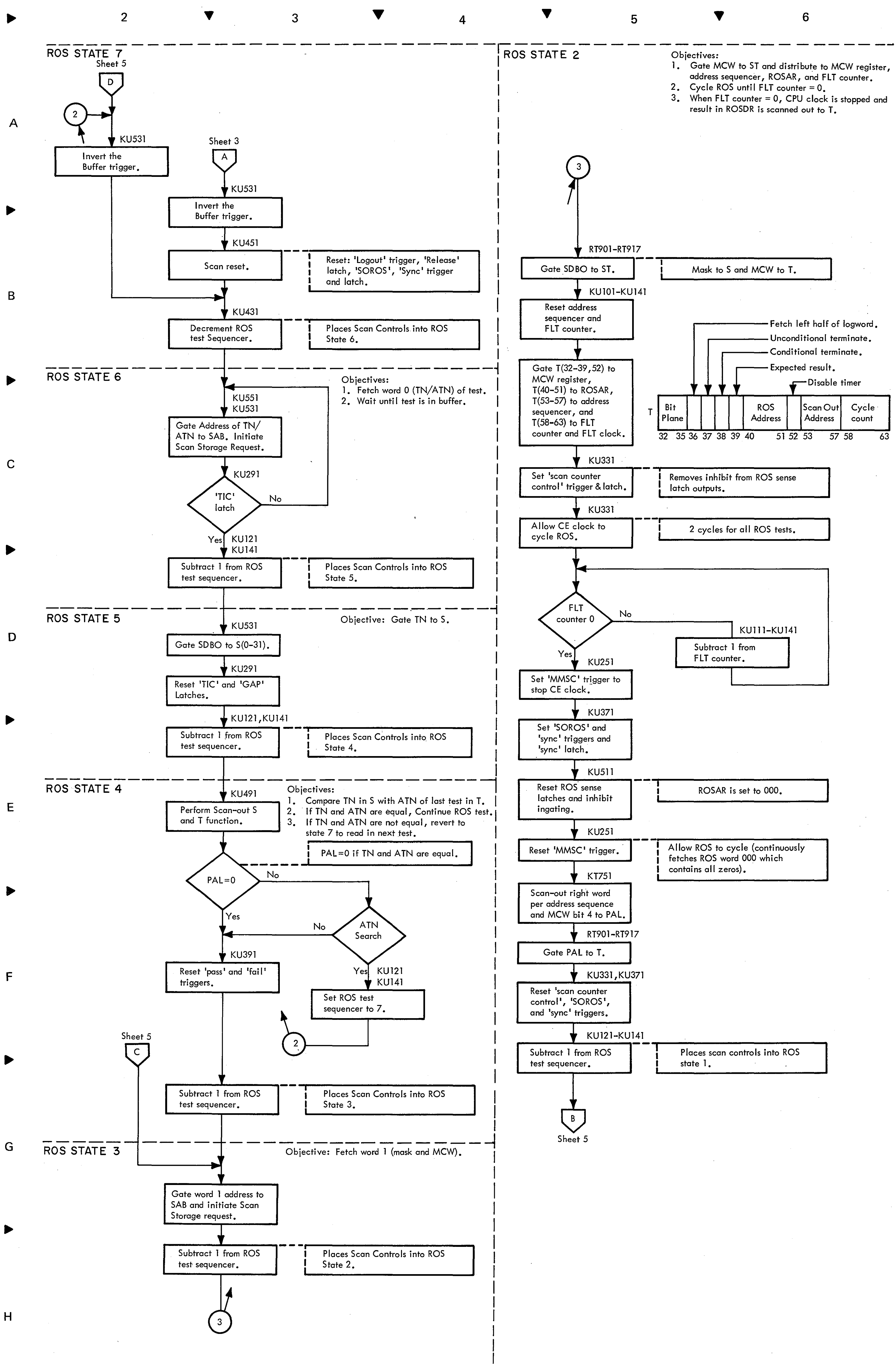


Diagram 6-115. ROS Test Sequence (Sheet 4 of 5)

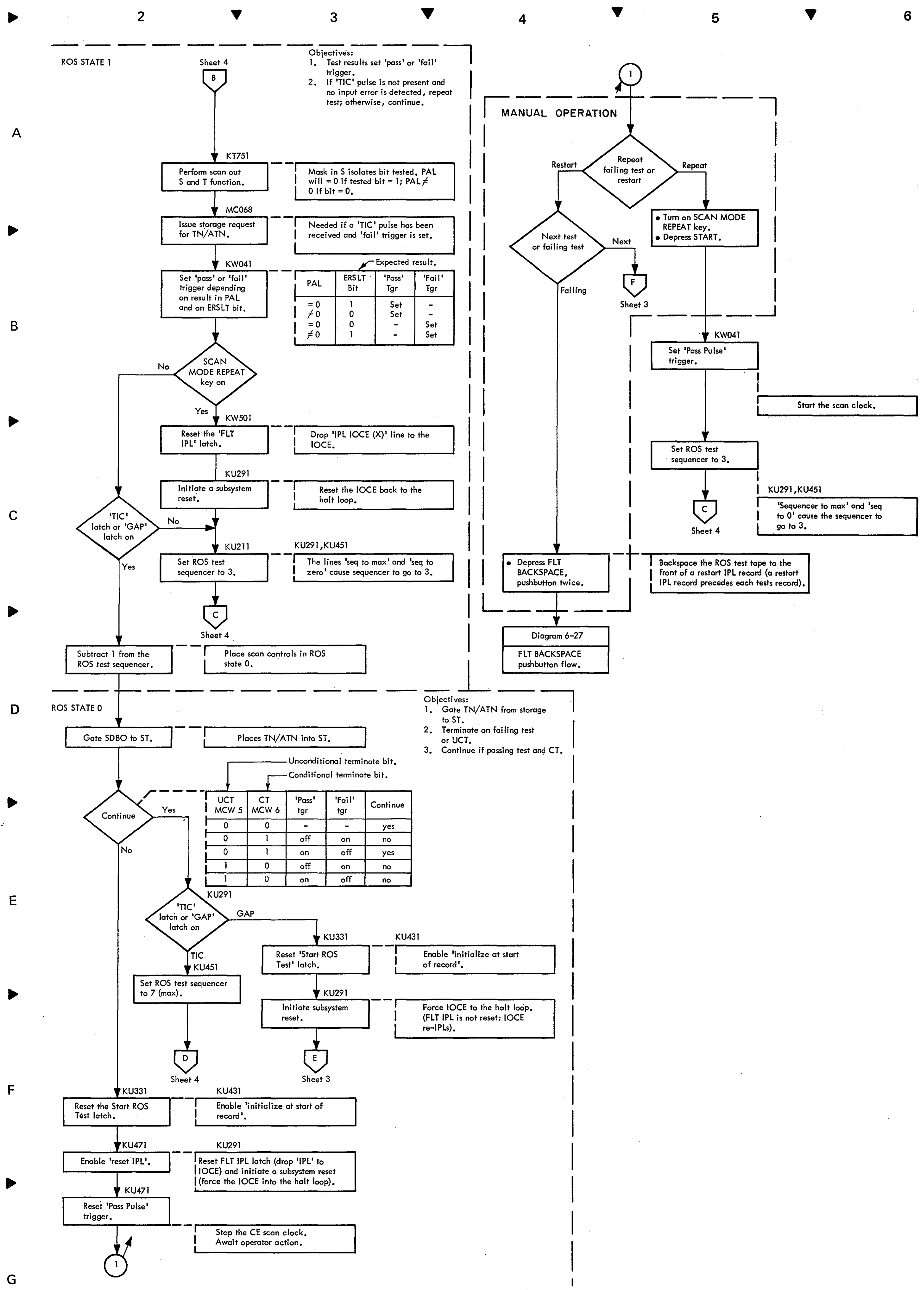


Diagram 6-115. ROS Test Sequence (Sheet 5 of 5)

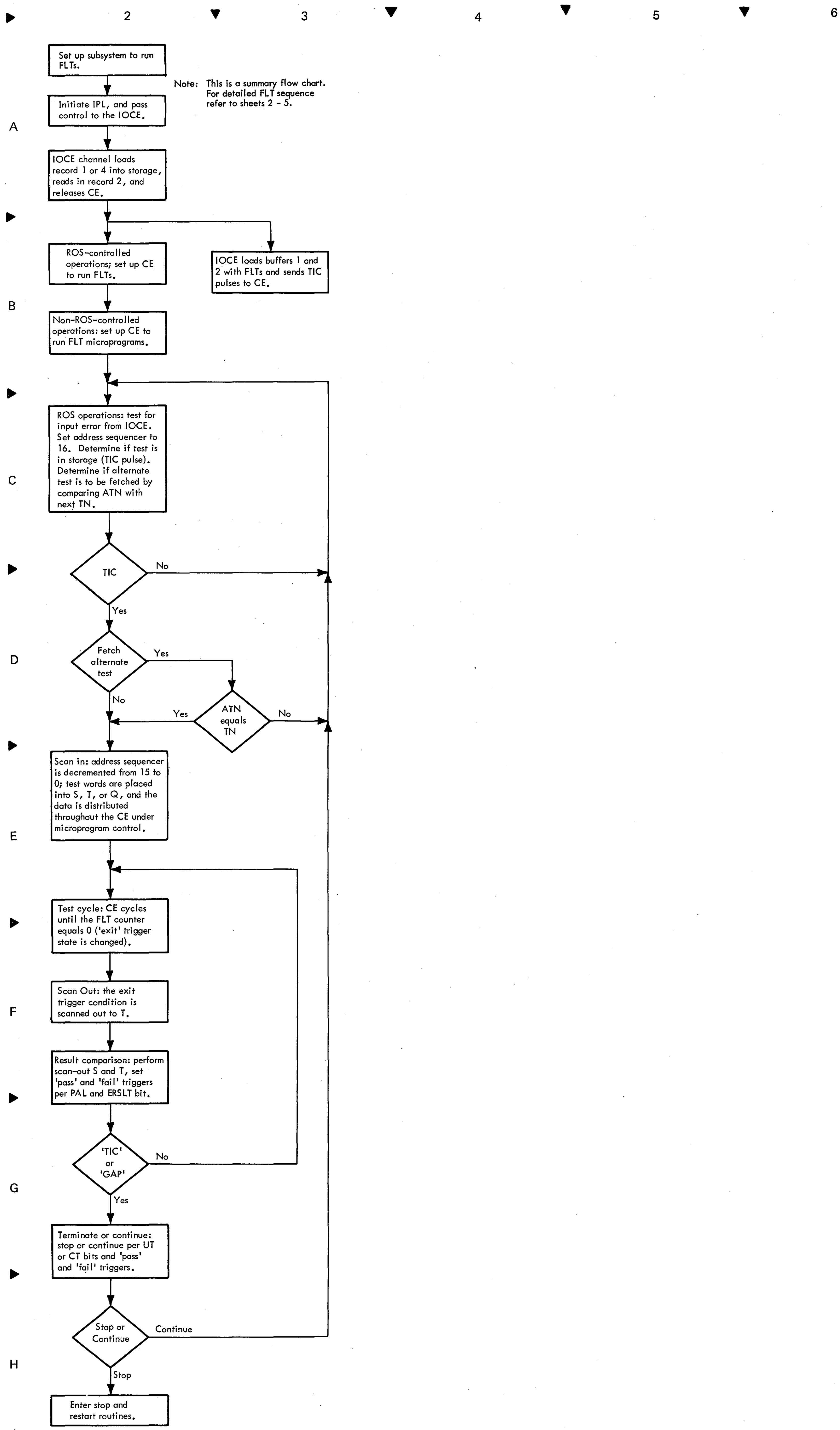


Diagram 6-116. FLT Sequence (Sheet 1 of 5)

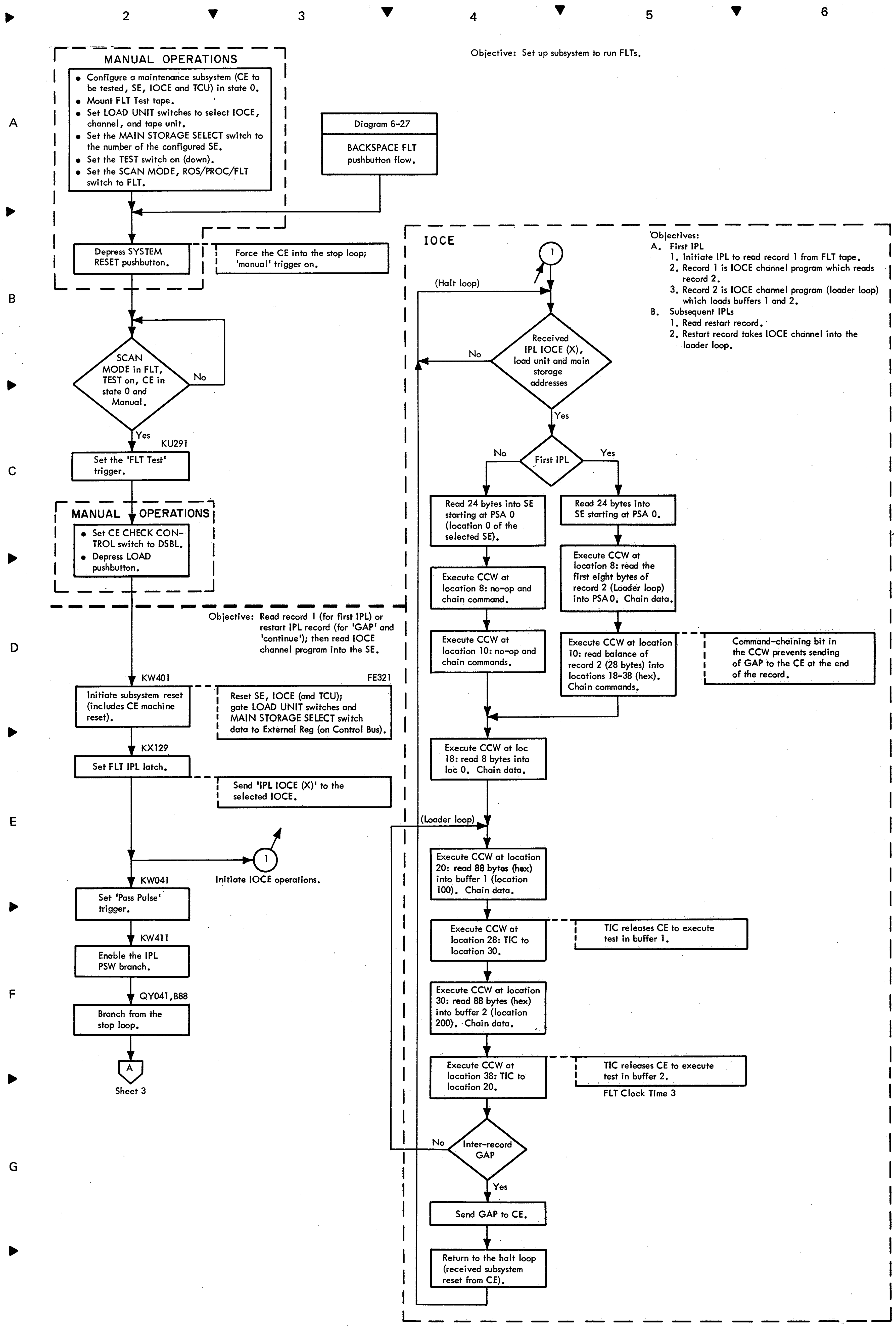


Diagram 6-116. FLT Sequence (Sheet 2 of 5)

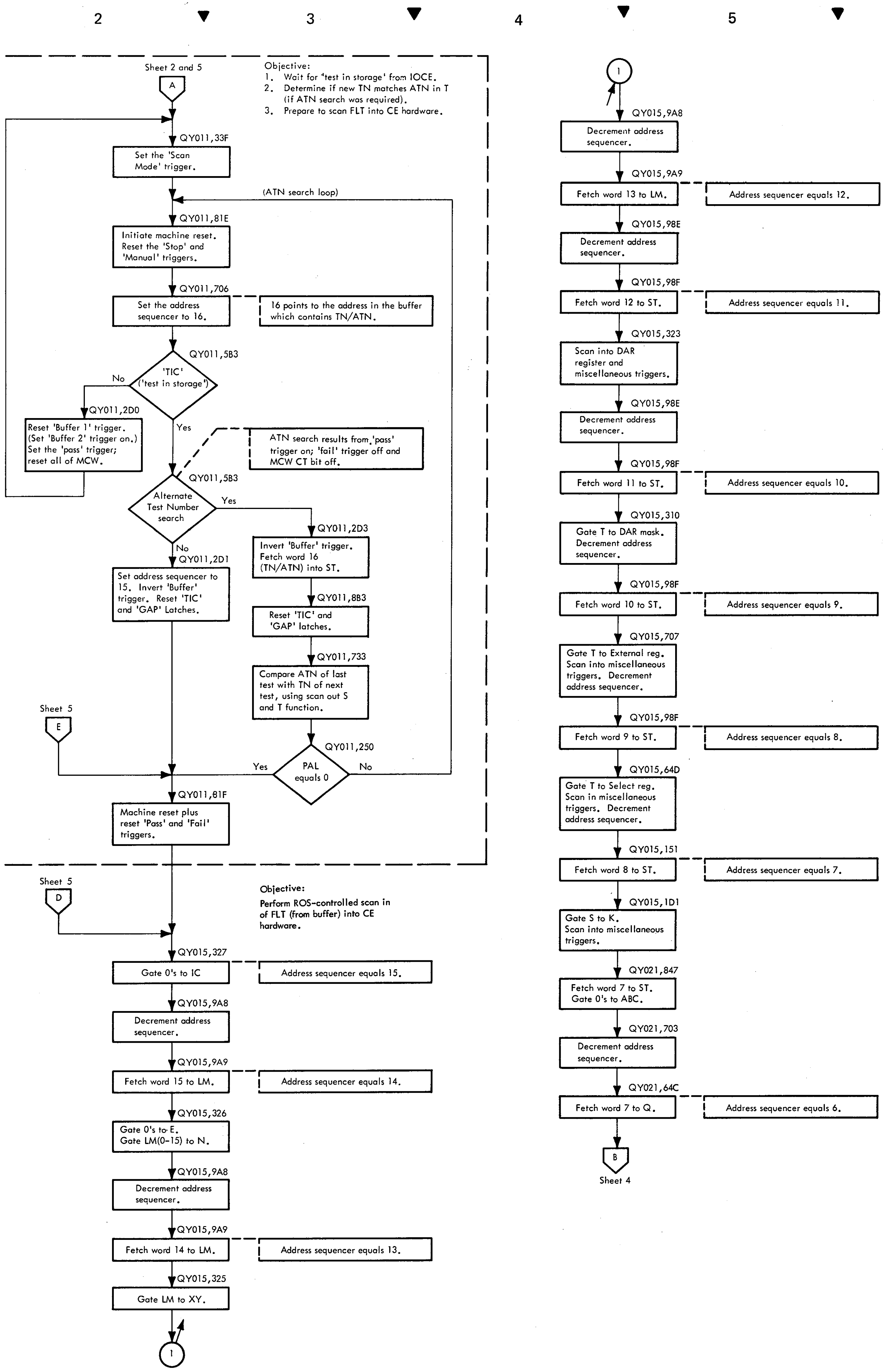


Diagram 6-116. FLT Sequence (Sheet 3 of 5)

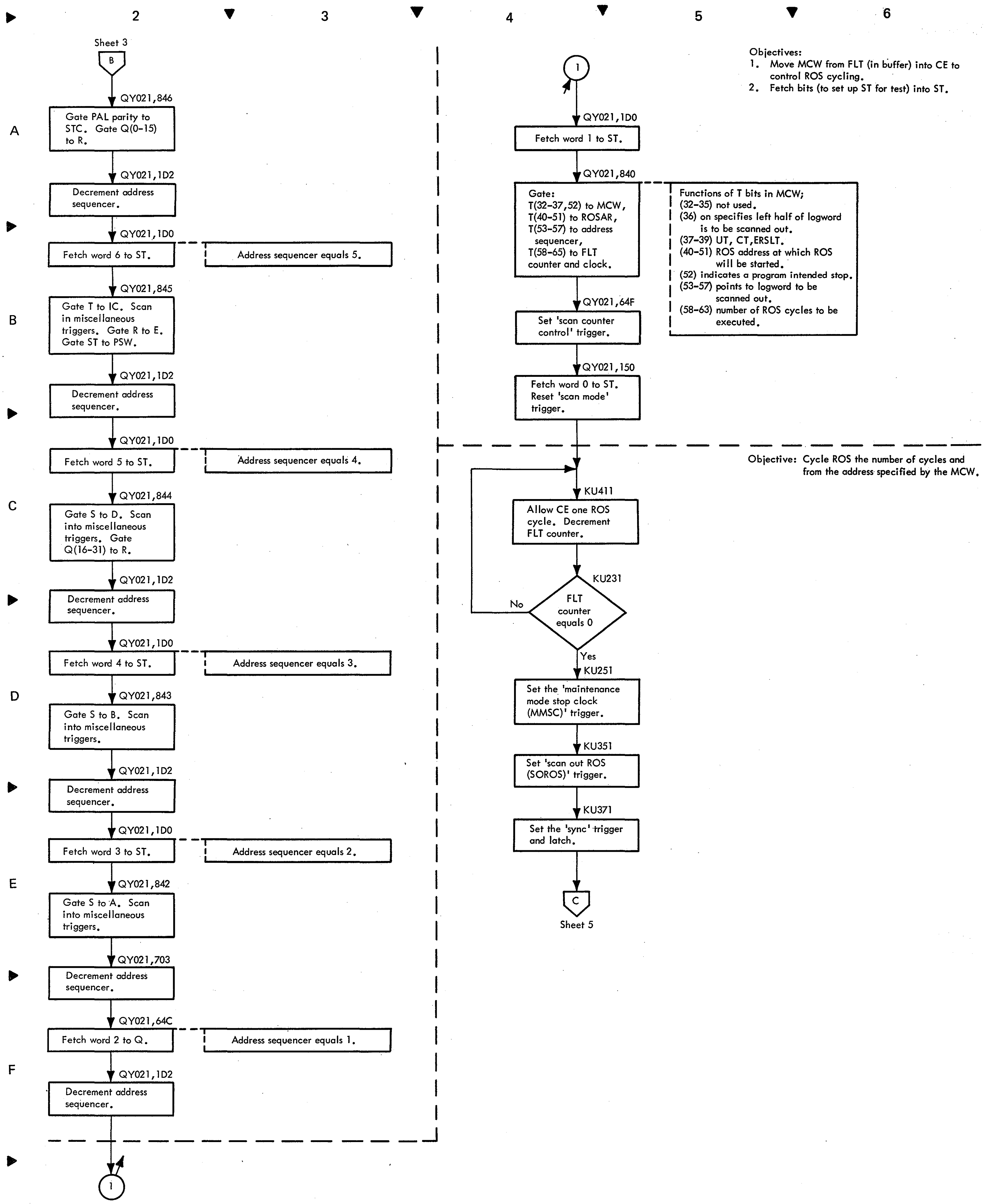
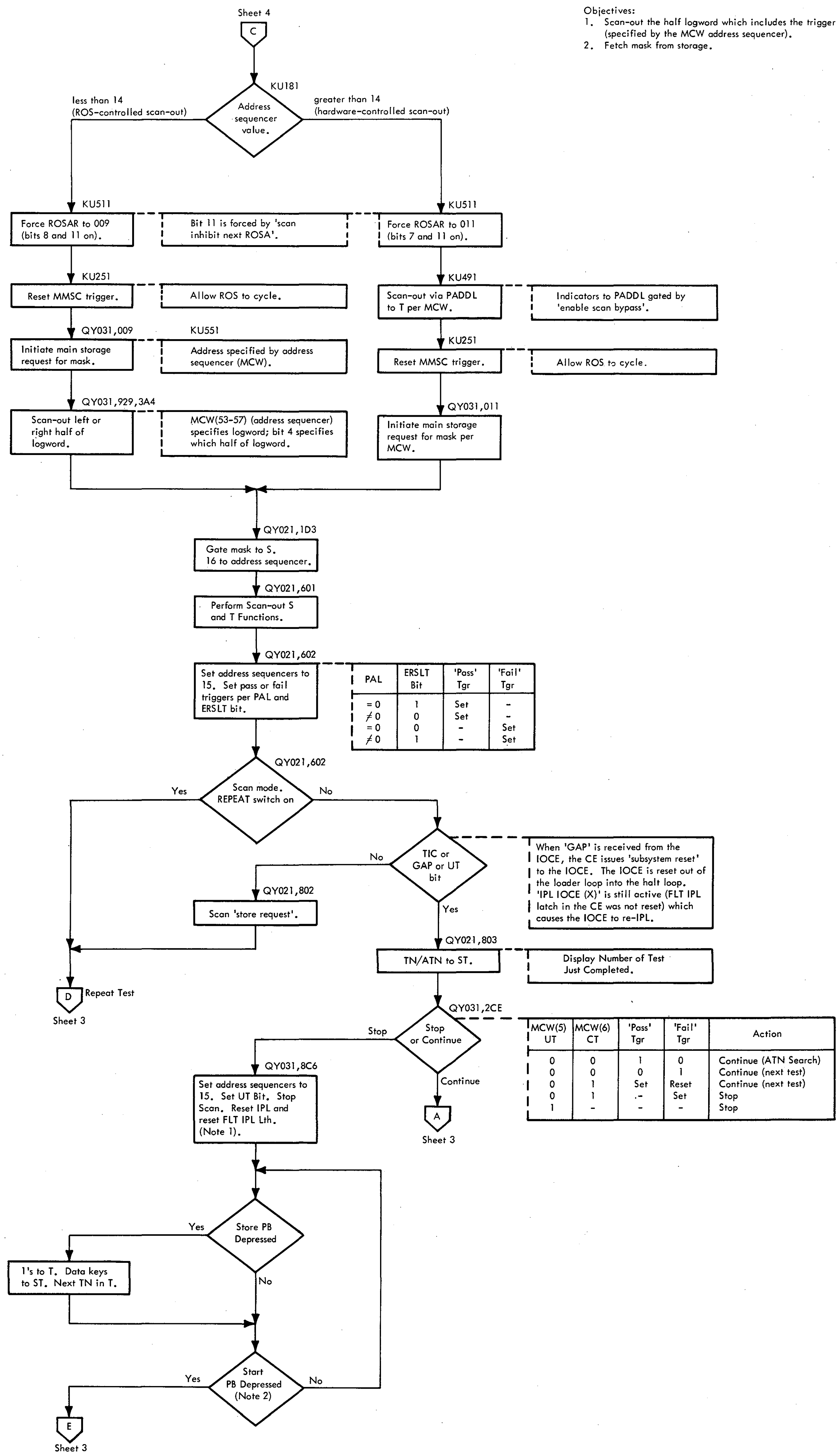


Diagram 6-116. FLT Sequence (Sheet 4 of 5)

2 ▼ 3 ▼ 4 ▼ 5 ▼ 6

Objectives:
1. Scan-out the half logword which includes the trigger being tested (specified by the MCW address sequencer).
2. Fetch mask from storage.



PAL	ERSLT Bit	'Pass' Tgr	'Fail' Tgr
= 0	1	Set	-
≠ 0	0	Set	-
= 0	0	-	Set
≠ 0	1	-	Set

MCW(5) UT	MCW(6) CT	'Pass' Tgr	'Fail' Tgr	Action
0	0	1	0	Continue (ATN Search)
0	0	0	1	Continue (next test)
0	1	Set	Reset	Continue (next test)
0	1	-	Set	Stop
1	-	-	-	Stop

Notes:
1. To restart after a stop, depress FLT BACKSPACE twice, then the LOAD Pushbutton once. This will bring in the next test and run it (Diagram 6-27).
2. To loop on failing test, operator must place SCAN MODE REPEAT Switch down before depressing START pushbutton.

Diagram 6-116. FLT Sequence (Sheet 5 of 5)

BIT POSITIONS IN MAIN STORAGE

LOG WORD NO.	PSA LOCATION	PSA WORD NO.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	COMMENTS
0	128 (80)	32	F REG P 0-7	D REG P 8-15	P 16-23	P 24-31					Q REGISTER P 0-7	P 8-15	P 16-23	P 24-31	P 32-39	P 40-47	P 48-55	P 56-63	DAR MASK P 0-7	P 8-15	P 16-23	P 24-31													PARITY FOR F, D, Q AND DAR MASK
	132	33	SELECT REG P 0-7	P 8-15	P 16-23	P 24-31					ST REGISTER P 0-7	P 8-15	P 16-23	P 24-31	P 32-39	P 40-47	P 48-55	P 56-63	CCR P 0-7	P 8-15	P 16-23	P 24-31													PARITY FOR SR, K, ST AND CCR
1	136 (88)	34	CE SEL SATR 1		G REG P 0-7	P 32-39	ATR P 0-7	P 8-15	P 16-23	P 24-31	IC REQ	RPT INSTR ADJ	INSTR STEP TGR	DISABLE INTER-LEAVE	STAT TGR A	I FETCH 1 TGR	IL NOT AVAIL	EXT INTRPT CE4 WDD 28	MCW RTP 0	ADDR SEQ 0	FLT CLOCK 0		MARK P 0-7	ST REG P 32-39	AB REG P 64-67	PADDL P 64-67									
	140	35	EXTERNAL TRIGGERS PERMIT IOCE 2	3	LOGOUT IOCE 1	2	3				AB REGISTER P 0-7	P 8-15	P 16-23	P 24-31	P 32-39	P 40-47	P 48-55	P 56-63	R REG P 0-7	P 8-15	E REG P 0-7	P 8-15													
2	144 (90)	36	CE SEL SATR 2	3	4	TIME CLOCK STEP		SELECT IOCE 1	2	3	IOCE MC REQ	INTRPT GATE TGR			EXTERNAL REG P 0-7	P 8-15	P 16-23	P 24-31	G REG (WRITE DIRECT) 0	1	2	3	4	5	6	7	ATR-2 32	33	34	35	36	37	38	39	G REG, ATR-2
	148	37	SELECT REGISTER PAM/RCU 1/1	2/2	3/	1	2	3	6	7	SE/DE 1/	2/	3/	4/	5/	6/1	7/2	8/3	9/4	10/5	18	19	CE 1	2	3	4	24	25	26	27	28	IOCE 1	2	3	SELECT REG
3	152 (98)	38	ADDRESS TRANSLATION REGISTER (ATR-1) 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	ATR-1
	156	39	STAT TRIGGERS B	C	D	E	F	G	H	BLOCK 1 FETCH	I FETCH 2 TGR	I FETCH 3 TGR	EXEC IN PROG TGR		PROG STORE COMP	I FETCH REQ TGR	BR INV ADDR TGR	I FETCH INV ADDR TGR	EXTERNAL INTERRUPT REGISTER CE1 WDD 20	CE1 RDD 21	CE2 WDD 22	CE2 RDD 23	TIMER 24	PB 25	CE3 WDD 26	CE3 RDD 27	CE4 RDD 29	PIR 30	DAR 31	PIR 1	2	3		TIMING GT TGR	
4	160 (A0)	40	PROGRAM STATUS WORD SYSTEM MASK 0	1	2	3	4	5	6	7	KEY 8	9	10	11	A 12	M 13	W 14	P 15	IC IN LSWR	PROGRAM STATUS WORD COND CODE 34	35	PROGRAM MASK 36	37	38	39	MC INTRPT			PROGRAM INTERRUPT CODE 8	4	2	1	INTERRUPT PRIORITY 1	2	PSW
	164	41	D REQ	SCAN REQ	SET KEY	INSERT KEY	3 CYCLE REQ	PAGE 1 LTH			RPT INSTR INIT	PULSE MODE ADJ	PULSE MODE INIT	STOP TGR	BLOCK TGR	PASS PULSE TGR	BLOCK INTRPT	FORCE ADDR TGR	SINGLE CYCLE TGR	CHECK SUM	CLOCK INHIBIT	RPT TEST	FLT TEST	ROS TEST TGR	SCC TGR	SYNC TGR	DISABLE INTER-LEAVE REV STOR ADDR	FLT BKSP	CONSOLE LOG OUT	SOROS	DIAG TGR	FLT REL	TIC	GAP	
5	168 (A8)	42	STORAGE CHECK ADDRESS REGISTER 0	1	2	3	360 MODE	TEST MODE	SE STPD IN 360 MODE		SAB CHK	SDBI CHK	SE/DE T/O	SE/DE ADDR CHK	CHECK REGISTER 2 SE/DE DATA CHK	FETCH CHK	LOS TO SE/DE		IOCE BUS CHK	LOCAL STORE BUS CHK	CCR PTY CHK	ATR PTY CHK	PSBAR PTY CHK	PSBAR NOT CONF	PSBAR ALT CHK	SPLIT LOG OUT	LOG ROS CHK	LOG ADDR CHK	CE LOG REQ	RDD T/O					CR2
	172	43	LACMP 1	RSA FSP 2	RMP 3	MCW REGISTER RSARP 4	5	LOC 6	7	DT 20	SCAN ADDRESS SEQUENCER 1	2	3	4	SCAN CYCLE COUNTER 0	1	2	3	FLT CLOCK 1		ROS TEST SEQUENCER 0	1	2					FLT PASS	FLT FAIL		FLT STG ERR		MAINT MODE STOP CLOCK	SCAN BUFFER 1	
6	176 (B0)	44	EXTERNAL REGISTER 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	EXTERNAL REG
	180	45	MARKS 0	1	2	3	4	5	6	7								TX TGR	AB REG 64	65	66	67		ABC 0	1	2	PADDL 64	65	66	67	INSERT LOCAL STORE SIGN	STC 0	1	2	
7	184 (B8)	46	AB REGISTER 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	AB REG (0-31)
	188	47	AB REGISTER 32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	AB REG (32-63)

Diagram 6-117. CE Logword Formats (Sheet 1 of 3)

2 3 4 5 6 7 8 9

BIT POSITIONS IN MAIN STORAGE

LOG WORD NO.	PSA LOCATION	PSA WORD NO.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	COMMENTS
8	192 (C0)	48	PHYSICAL PSBAR				PSBAR COUNTER				LOGICAL PSBAR																PROGRAM STATUS WORD SYSTEM MASK			D	SE STPD	PSA LOCKOUT 1	2	3	PSBAR
			9	10	11	12	P 9-12	9	10	11	P 9-15	9	10	11	12	13	14	15	16	17	18	19	ALT PSBAR				16	17	18	19	24	25	26	27	
	196	49	STATE		SCON				ILOS		SE/DE						CONFIGURATION CONTROL REGISTER (CCR)						CE									IOCE			CCR
			S_0	S_1	1	2	3	4	6		1/	2/	3/	4/	5/	6/1	7/2	8/3	9/4	10/5	18	19	1	2	3	4	24	25	26	27	28	1	2	3	
	200 (C8)	50	IOCE						SE/DE								DIAGNOSE ACCESSIBLE REGISTER MASK				PAM/RCU			TCU			CE OWN			CE					DAR MASK
			1	SPARE	2	SPARE	3	SPARE	1/	2/	3/	4/	5/	6/1	7/2	8/3	9/4	10/5	SPARE 16	SPARE 17	1/1	2/2	3/	1	2	3	OTC	OBS	SPARE 26	1	2	3	4	SPARE 31	
	204	51	SERIAL ADDER LATCHES (SADDL)								INSTRUCTION COUNTER (IC)																								SADDL, IC
			0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
10	208 (D0)	52	R REGISTER																E REGISTER																R REG, E REG
			0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
	212	53	F REGISTER (READ DIRECT)								D REGISTER																								D REG, F REG
			0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
11	216 (D8)	54	Q REGISTER																																Q REG
			0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
	220	55	Q REGISTER																																Q REG
			32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	
12	224 (E0)	56	IOCE1		IOCE2		IOCE3		SE/DE ELC								DIAGNOSE ACCESSIBLE REGISTER (DAR)				PAM/RCU ELC			TCU ELC			CE OWN			CE ELC					DAR
			A	B	A	B	A	B	1/	2/	3/	4/	5/	6/1	7/2	8/3	9/4	10/5	SPARE 16	SPARE 17	1/1	2/2	3/	1	2	3	OTC	OBS	SPARE 26	1	2	3	4	SPARE 31	
	228	57	PARALLEL ADDER FULL SUM-CHECK															CHECK REGISTER 1	PARALLEL ADDER HALF-SUM CHECK											ROS PARITY CHECK					CR1
			4-7	8-15	16-23	24-31	32-39	40-47	48-55	56-63									4-7	8-15	16-23	24-31	32-39	40-47	48-55	56-67				2-42	43-68	69-99	SADD HALF SUM CHK	SADD FULL SUM CHK	
13	232 (E8)	58	LSWR					CHECK REG 1			EXT TGRS		N REG		PSW REG										IC REG										
			P 0-7	P 8-15	P 16-23	P 24-31	E REG PTY 0-7	PADD FULL SUM 64-67	MPR DECODE PTY CHK	PADD HALF SUM CHK	PERMIT IOCE 1	IOCE INTRPT REQ	P 0-7	P 8-15	P 0-7	P 8-15	NO RETRY	SVC INTRPT	PHYS PSBAR PTY 9-12	PSBAR CNTR 12	LOGICAL PSBAR PTY 16-19		SADDL P 0-7	P 8-15	P 16-23	P 24-31									
	236	59	LOCAL STORAGE WORKING REGISTER (LSWR)																																LSWR
			0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
14	240 (F0)	60																																	
	244	61	T32-63 TO PADDA								D8-31 TO PADDA																								
			32-63	32-63 COMP	31-62 L1	31-62 C-L1	T 32-47 TO PADDA 48-63	T 48-63 TO PADDA 48-63	K 0-31 TO PADDA 32-63	8-31	7-30 L1	7-30 C-L1	40-63	40-63 COMP	FMTO	FMTM	FMTW	IC 8-31 TO PADDB 40-63	AB 4-7 TO PADDB 4-7	AB 8-31 TO PADDB 8-31	AB 32-63 TO PADDB 32-63	AB 64-67 TO PADDB 64-67	AB 6-31 TO PADDB 4-29L2	AB 32-67 TO PADDB 30-65L2	AB 64-67 TO PADDB 28-31	EX 6 TO PADDB 28-63	HOT ONE TO PADDB 60	E 8-11 TO PADDB 56-59	E 12-15 TO PADDB 60-63	E 8-11 TO PADDB 60-63	Q 4-15 TO PADDB 52-63	Q 20-31 TO PADDB 52-63	Q 36-47 TO PADDB 52-63	Q 52-63 TO PADDB 52-63	
15	248 (F8)	62																																	
	252	63															READ ONLY STORAGE DATA REGISTER (ROSDR)																		ROSDR
			70	71	72	73	74	75	76	77	79	80	81	82	83	84	P 43-68	86	88	89	90	P 69-99	92	93	94	95	97	98	99		RIGHT DIGIT TGR	S TGR	LEAVE TGR	STEP ABC TGR	

Diagram 6-117. CE Logword Formats (Sheet 2 of 3)

LOG WORD NO.	PSA LOC-ATION	PSA WORD NO.	BIT POSITIONS IN MAIN STORAGE																																COMMENTS
			0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
16	256 (100)	64																																	
16	260	65	READ ONLY STORAGE DATA REGISTER (ROSDR) 37	38	39	40	41	42	43	44	46	47	48	49	50	51	52	53	55	56	57	58	59	60	61	62	64	65	66	67	68	INHIBIT LS WRITE	PROSA A LAST	FLT MODE	ROSDR
17	264 (108)	66																																	
17	268	67		READ ONLY STORAGE DATA REGISTER (ROSDR) 2	3	4	5	6	7	8	10	11	12	13	14	15	16	17	19	P 2-42	21	22	23	24	25	26	28	29	30	31	32	33	34	35	ROSDR
18	272 (110)	68	L REG P 0-7	P 8-15	P 16-23	P 24-31	K REG P 0-7	P 8-15	P 16-23	P 24-31																									
18	276	69	READ ONLY STORAGE ADDRESS REGISTER (ROSAR) 1	2	3	4	5	6	7	8	10	11	PREVIOUS ROS ADDRESS (A) 0	1	2	3	4	5	7	8	9	10	11	PREVIOUS ROS ADDRESS (B) 0	1	2	4	5	6	7	8	9	10	11	ROSAR, PROSA, PROSB
19	280 (118)	70	M REG P 32-39	P 40-47	P 48-55	P 56-63	X REG P 0-7	P 8-15	P 16-23	P 24-31	Y REG P 32-39	P 40-47	P 48-55	P 56-63					N REGISTER 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	N REG
19	284	71	ROSAR 0	9	PROSA A6	B3	READ ONLY STORAGE DATA REGISTER (ROSDR) 0	9	18	27	36	45	54	63	69	78	87	96	S 0-31 TO PADDA 32-63	D 8-31 TO PADDA 8-31 C	AB 0-31 TO PADDB 32-63	F 4-7 TO PADDB 60-63					LOCAL STORAGE ADDRESS REG 0	1	2	3	4	WR LOCAL STORE			ROSAR, LSAR
20	288 (120)	72	K REGISTER 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	K REG
20	292	73	MAINTENANCE CONTROL WORD (MCW) DE/SE 32	33	SELECT 34	35	DG SELECT 36	37	38	39	CVG SELECT 40	41	42	43	44	45	RNOP 46	FDGR 47	DSE 1 48	RERP 49	RC 50	WRAP DE 51													MCW
21	296 (128)	74	LM REGISTER 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	LM REG
21	300	75	LM REGISTER 32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	LM REG
22	304 (130)	76	XY REGISTER 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	XY REG
22	308	77	XY REGISTER 32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	XY REG
23	312 (138)	78	ST REGISTER 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	ST REG
23	316	79	ST REGISTER 32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	ST REG

Diagram 6-117. CE Logword Formats (Sheet 3 of 3)

WORD NO.	BIT POSITIONS IN MAIN STORAGE																																COMMENTS
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
0	SESDR (EVEN) (or IOCE SBO)																																SESDR (EVEN)
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
1	SESDR (EVEN)																																SESDR (EVEN)
	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	
2	MARKS (EVEN)									IOCE SAB (EVEN) / TAG (EVEN)				HI/LO	SESAR 1-14 (EVEN)														IOCE SAB				MARKS, SESAR (EVEN)
	0-7	8-15	16-23	24-31	32-39	40-47	48-53	54-63		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
3	CCR(0-7) / CE					IOCE			REQUEST (EVEN) / CE				IOCE				RESPONSE (EVEN) / CE				IOCE				CCR (8-15) / STATE		SCON						CCR, REQUEST AND RESPONSE LATCHES (EVEN)
	1	2	3	4		1	2	3	1	2	3	4	1	2	3		1	2	3	4	1	2	3		S_0	S_1	1	2	3	4	14	15	
4										SE ID							OPERATION		OPERATION (EVEN)				BLK SEL LATCH		SCON BACK UP LATCHES					ADDR PARITY			TYPE OF OP (EVEN)
										1	2	3	4				ISK	SSK	T & S	STORE	FETCH	SUPP LOG CHK	EVEN	ODD	1	2	3	4	MARK PARITY (EVEN)	1-5 0-7	6-12 8-15	13-19 16-23	
5	CHECK CONDITION (EVEN)								CHECK CONDITIONS (COMMON)														CCR PARITY		SESDR PARITY (EVEN) / SBO (IOCE)								CHECKS (EVEN AND COMMON), SESDR (EVEN)
	MARK PARITY CHK	ADDR PARITY CHK	DATA PARITY CHK	SP PARITY CHK	KEY	NORMAL OP CHK			CCR	TAG PARITY CHK	TAG	SAB (IOCE)	SBO (IOCE)				OBS	OTC THERMAL					0-7	8-15	0-7	8-15	16-23	24-31	32-39	40-47	48-55	56-63	
6	SESDR (ODD)																																SESDR (ODD)
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
7	SESDR (ODD)																																SESDR (ODD)
	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	
8	MARKS (ODD)									IOCE SAB (ODD) / TAG (ODD)				HI/LO	SESAR 1-14 (ODD)																		MARKS, SESAR (ODD)
	0-7	8-15	16-23	24-31	32-39	40-47	48-55	56-63		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19					
9									REQUEST (ODD) / CE				IOCE				RESPONSE (ODD) / CE				IOCE												REQUEST AND RESPONSE LATCHES (ODD)
									1	2	3	4	1	2	3		1	2	3	4	1	2	3										
10	SPIKR								SPOKR										OPERATION (ODD)													ADDR PARITY			SPIKR, SPOKR, TYPE OF OP (ODD)
	P 0-4	0	1	2	3	4			P 0-4	0	1	2	3	4					T & S	STORE	FETCH	SUPP LOG CHK							MARK PARITY (ODD)	1-5 0-7	6-12 8-15	13-19 16-23	
11	CHECK CONDITION (ODD)						SPAR																		SESDR PARITY (ODD)								CHECKS (ODD), SPAR, SESDR (ODD)
	MARK PARITY CHK	ADDR PARITY CHK	DATA PARITY CHK	SP PARITY CHK	KEY	NORMAL OP CHK	P	X	Y	2	3	4	5	6	7										0-7	8-15	16-23	24-31	32-39	40-47	48-55	56-63	

Diagram 6-118. SE Logword Formats

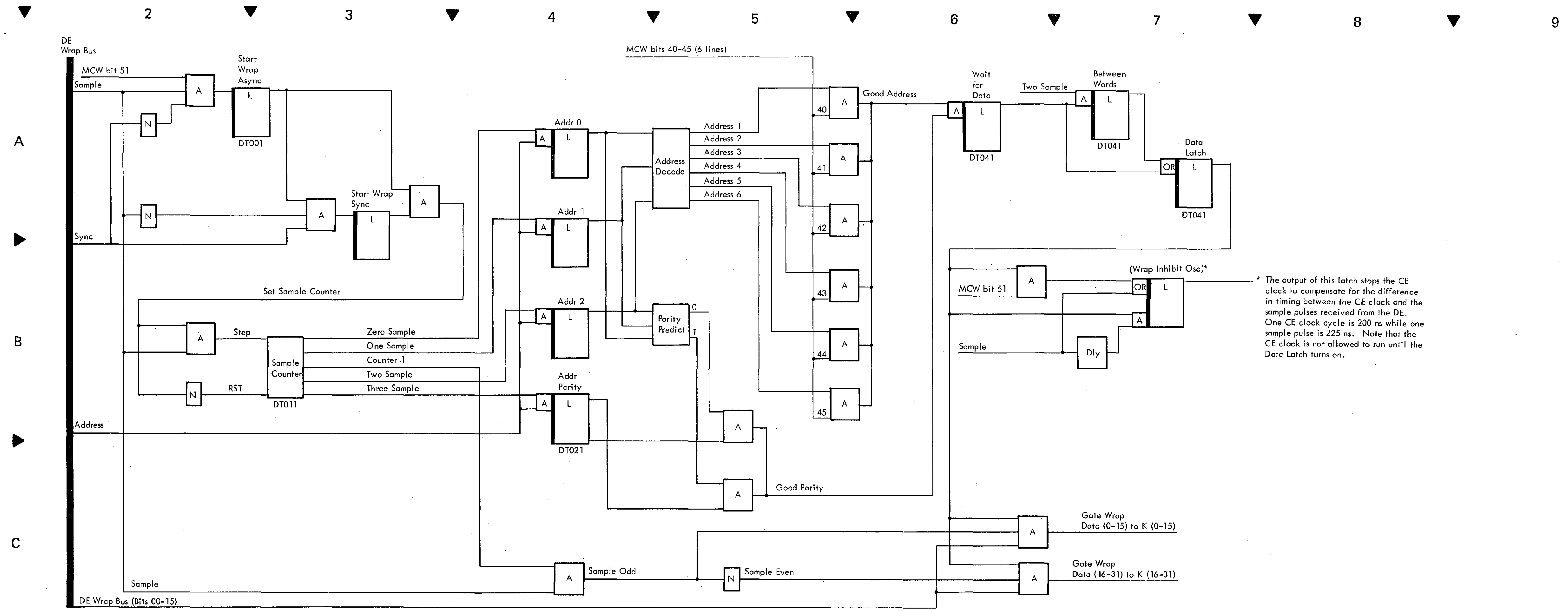
2 3 4 5 6 7 8 9

A B C D E

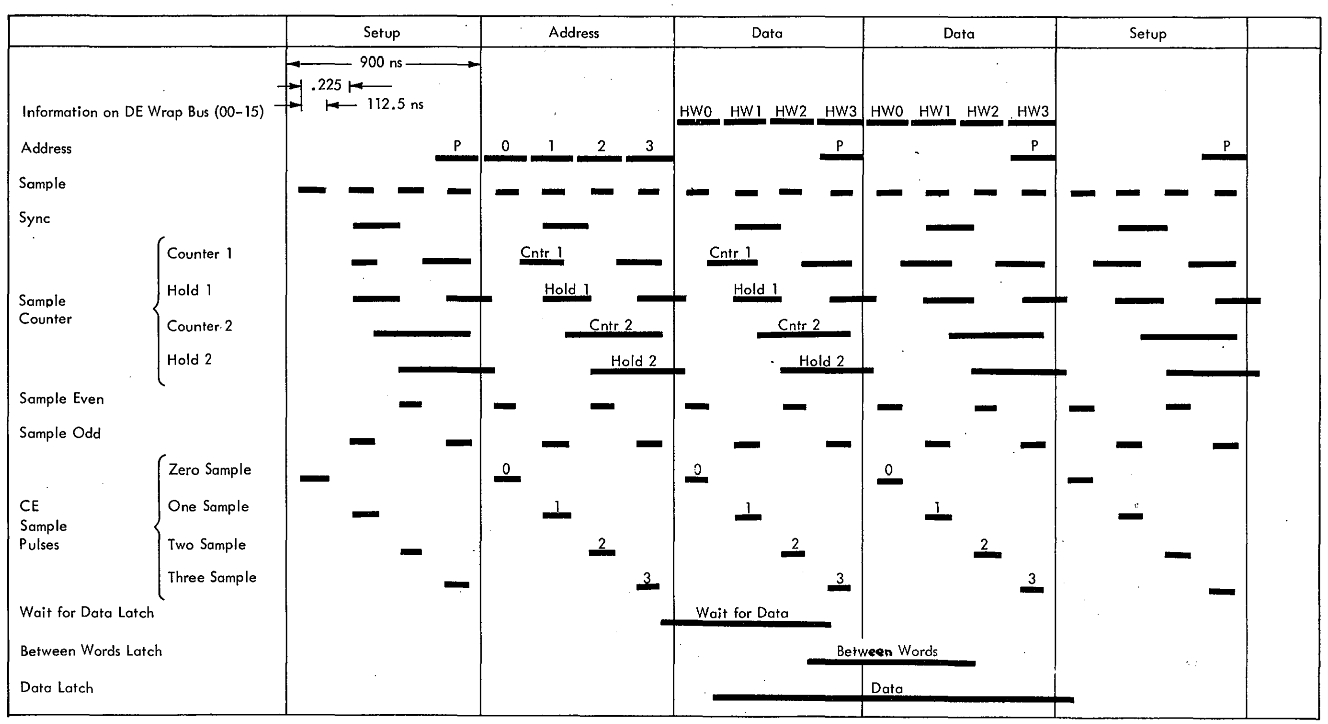
WORD NO.	BIT POSITIONS IN MAIN STORAGE																																COMMENTS
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
0	DESDR (EVEN)																																DESDR (EVEN)
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
1	DESDR (EVEN)																																DESDR (EVEN)
	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	
2	MARKS (EVEN)								DESDR PARITY (EVEN)								DESAR 1-14 (EVEN)																MARKS DESDR, DESAR (EVEN)
	0-7	8-15	16-23	24-31	32-39	40-47	48-55	56-63	0-7	8-15	16-23	24-31	32-39	40-47	48-55	56-63	P 6-12	6	7	8	9	10	11	12	P 13-19	13	14	15	16	17	18	19	
3											SPIKR						SPOKR						OPERATION		CHECK CONDITION (COMMON)								CHECKS (EVEN AND COMMON), SPIKE, SPOKE
	MARK PARITY	T & S	STORE	FETCH	DATA CHK	ADDR CHK	MARK PARITY CHK	CANCEL LATCH			P 0-4	0	1	2	3	4	P 0-4	0	1	2	3	4	ISK	SK	SP ADDR CHK	SP KEY CHK	KEY MIS-MATCH	OBS	OTC				
4	DESDR (ODD)																																DESDR (ODD)
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
5	DESDR (ODD)																																DESDR (EVEN)
	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	
6	MARKS (ODD)								DESDR PARITY (ODD)								DESAR 1-14 (ODD)																MARKS DESDR, DESAR (ODD)
	0-7	8-15	16-23	24-31	32-39	40-47	48-55	56-63	0-7	8-15	16-23	24-31	32-39	40-47	48-55	56-63	P 6-12	6	7	8	9	10	11	12	P 13-19	13	14	15	16	17	18	19	
7	MARK PARITY	T & S	STORE	FETCH	DATA CHK	ADDR CHK	MARK PARITY CHK	CANCEL LATCH																									CHECKS (ODD)
8	CCR																																CCR
	DG																			STATE		SCON						CE					
	P 0-7	1A	1B	2B	2C	3C	3D	4D	4A	P 8-15	5A	5B	6B	6C	7C	7D	8D	8A	P 16-23	S_0	S_1	1	2	3	4	IDES		1	2	3	4	CCR INVALID	
9	RESPONSE				CVG																								ADDRESS REG				RESPONSE LATCHES, DG ADDR REG ID
	CE				A						B						C						D										
	1	2	3	4	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	A	B	C	D	
10	DG REGISTER CHECK								CHECK CONDITIONS (COMMON)								DG ADDRESS REGISTER																CHECKS DG ADDR REG
	1	2	3	4	5	6	7	8	TAG PARITY CHK	TAG	CCR PARITY CHK	MULTI ACCEPT	LS PARITY CHECK	NORMAL OP	REFRESH PARITY CHECK		P 1-7	1	2	3	4	5	6	7	P 8-14	8	9	10	11	12	13	14	
11	TAG						DE ID				HALFWORD COUNTER				DG DATA REGISTER																		DG DATA REG
	P 1-5	1	2	3	4	5	1	2	3	4	8	4	2	1	P 0-7	0	1	2	3	4	5	6	7	P 8-15	8	9	10	11	12	13	14	15	

Diagram 6-119. DE Logword Formats

Diagram 6-201. DE Wrap Bus Controls



* The output of this latch stops the CE clock to compensate for the difference in timing between the CE clock and the sample pulses received from the DE. One CE clock cycle is 200 ns while one sample pulse is 225 ns. Note that the CE clock is not allowed to run until the Data Latch turns on.



Note:
The chart at the left shows the timing for transferring one quadword of data to the CE via the DE Wrap Bus. The Diagnose instruction controls the total number transferred by setting the quadword count in the MCW.