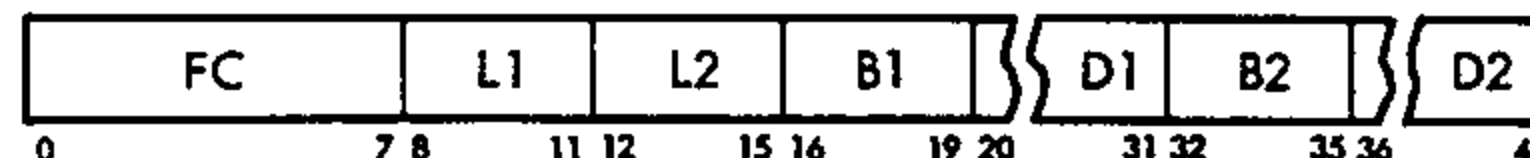


• SS Format:



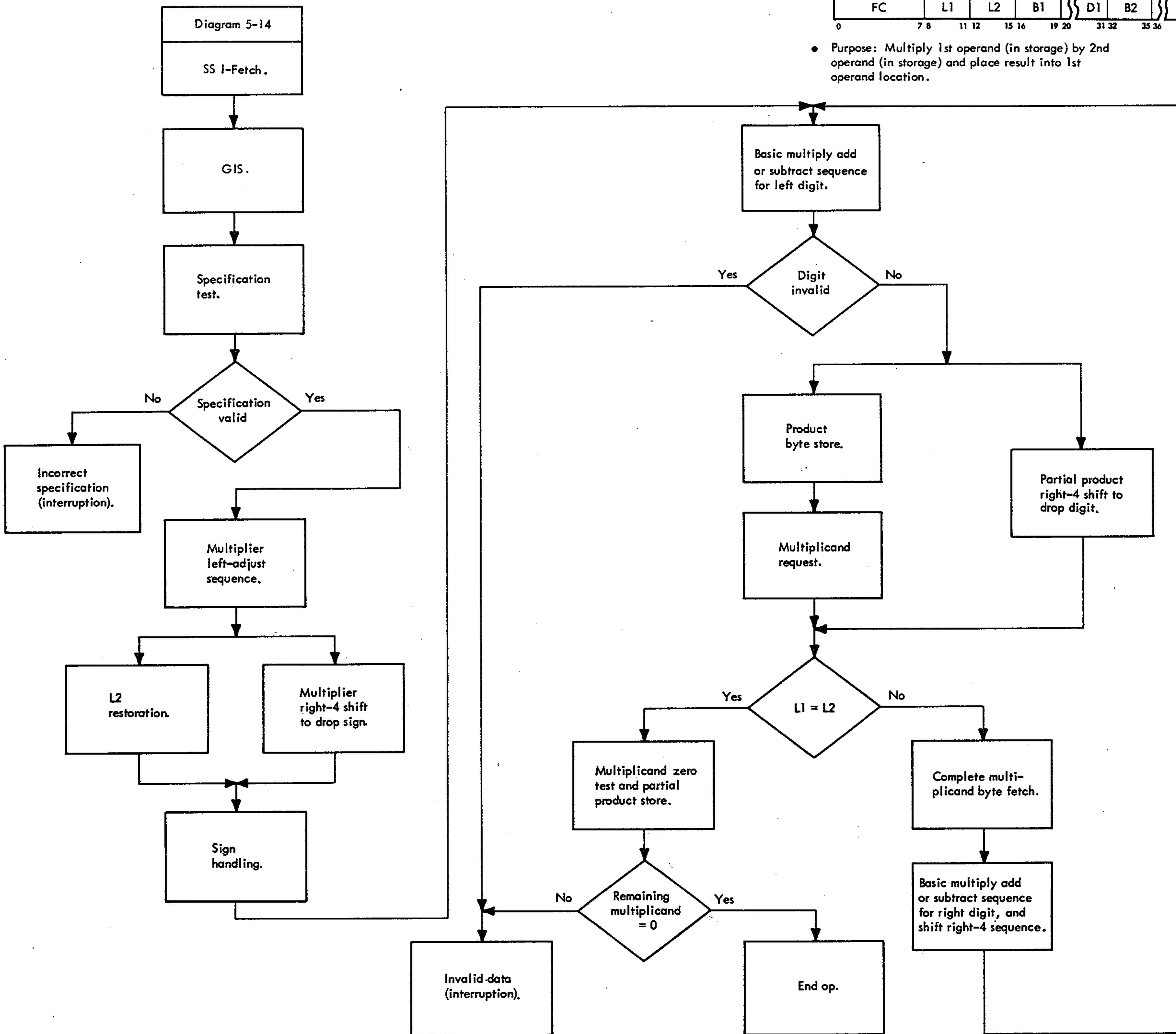
• Purpose: Multiply 1st operand (in storage) by 2nd operand (in storage) and place result into 1st operand location.

A

B

C

D



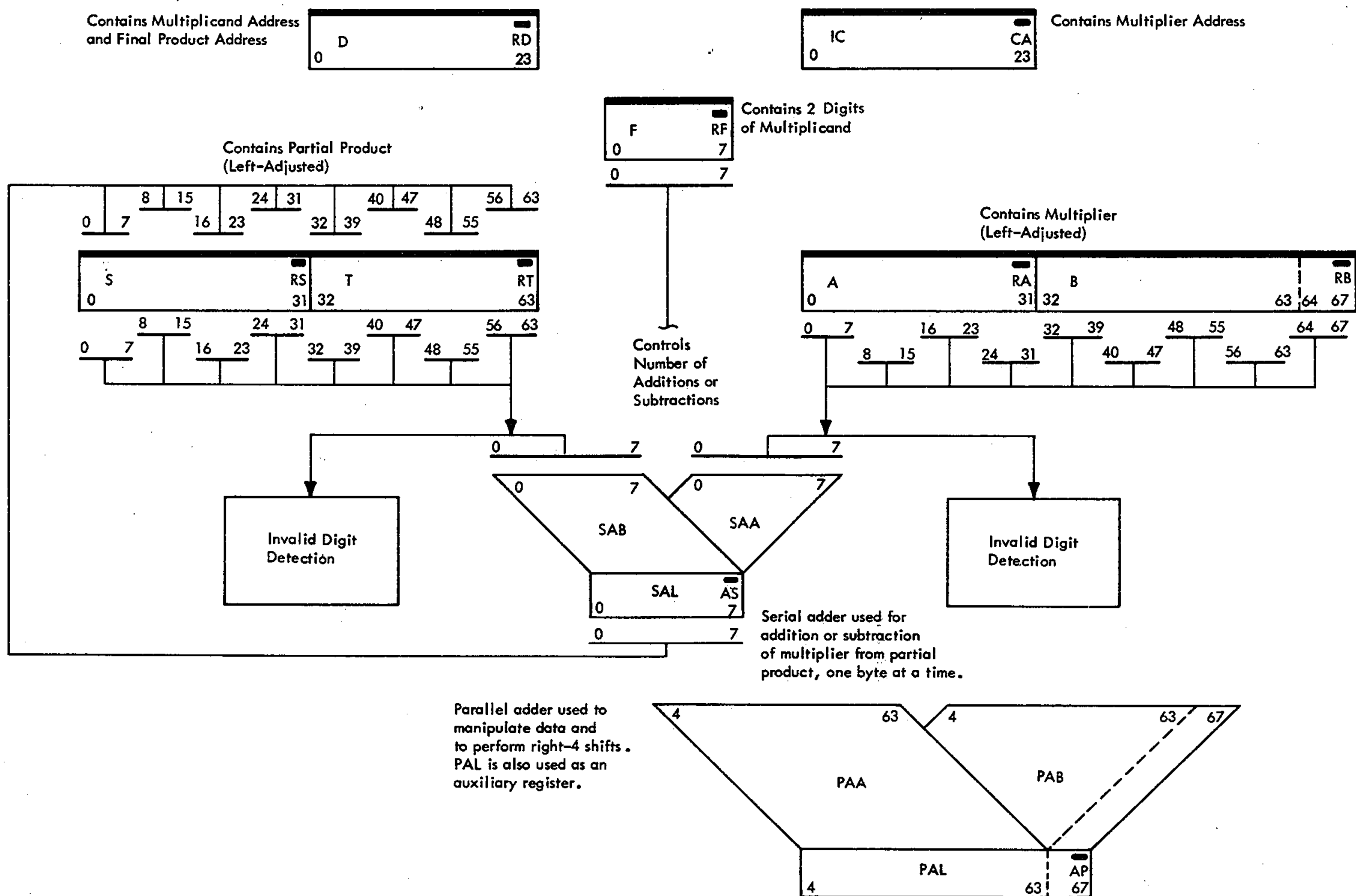
A) Overall Flow Chart

E

F

G

H



B) General Data Path

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

A

B

C

D

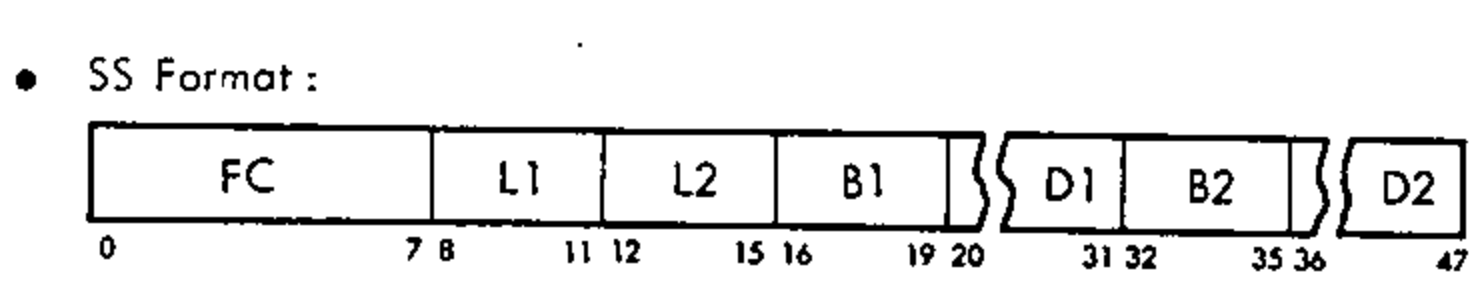
E

F

G

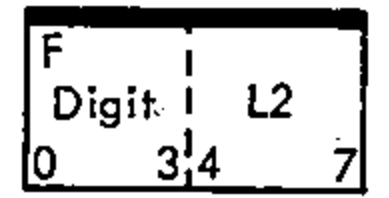
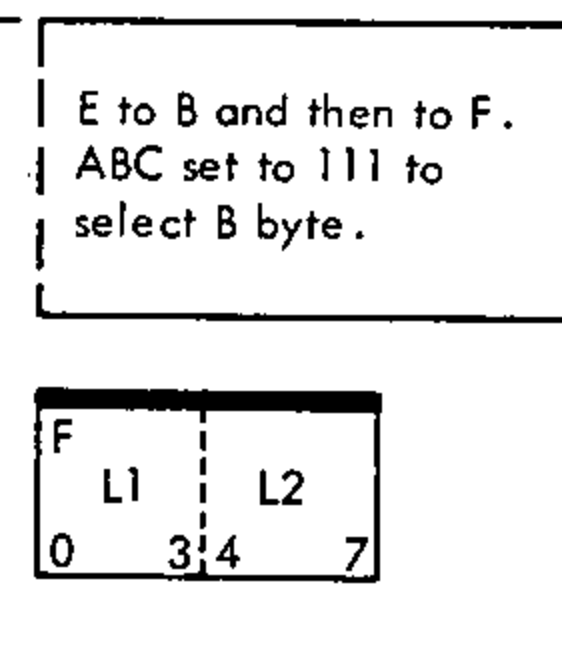
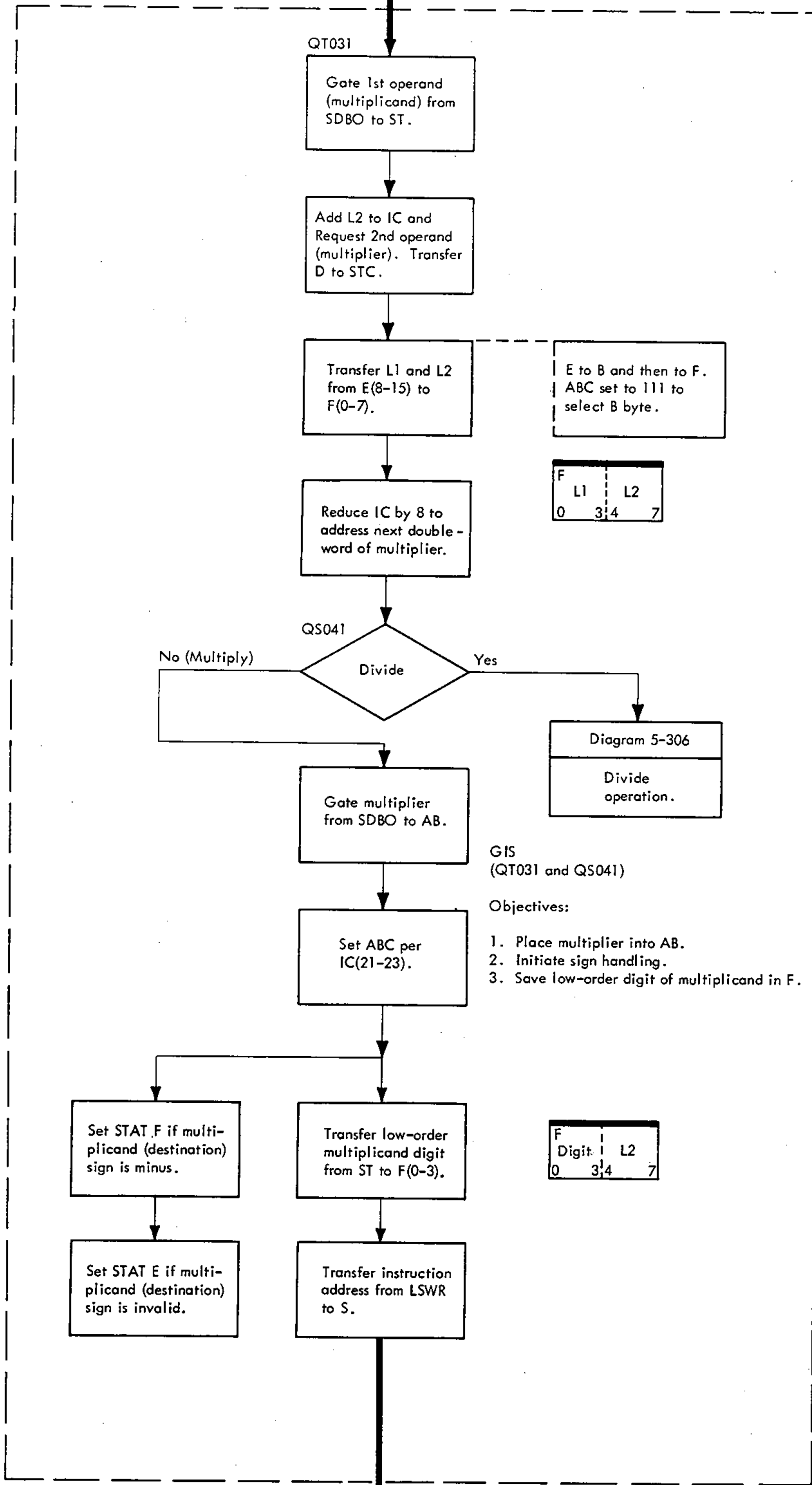
H

Diagram 5-14  
SS I-Fetch.



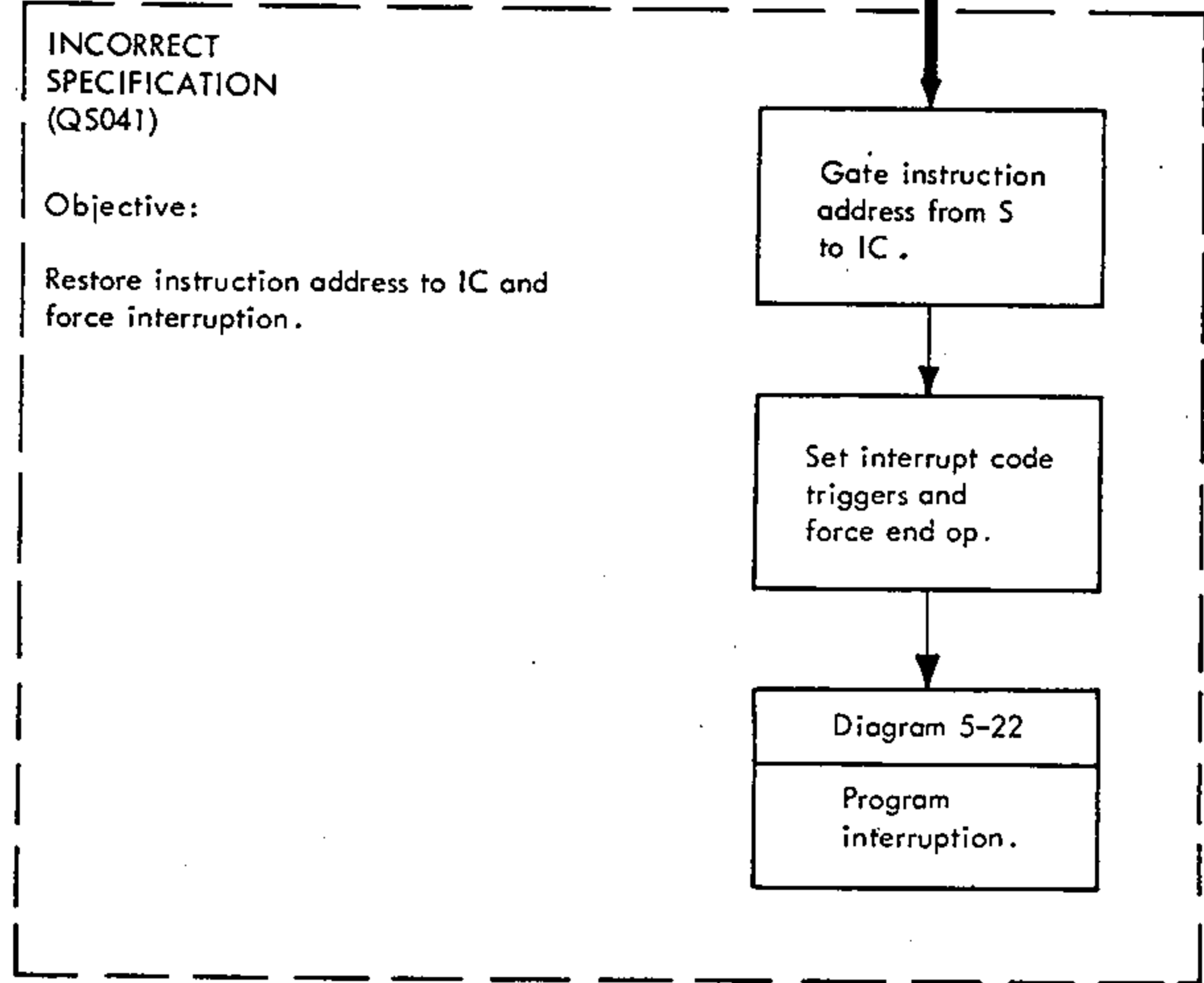
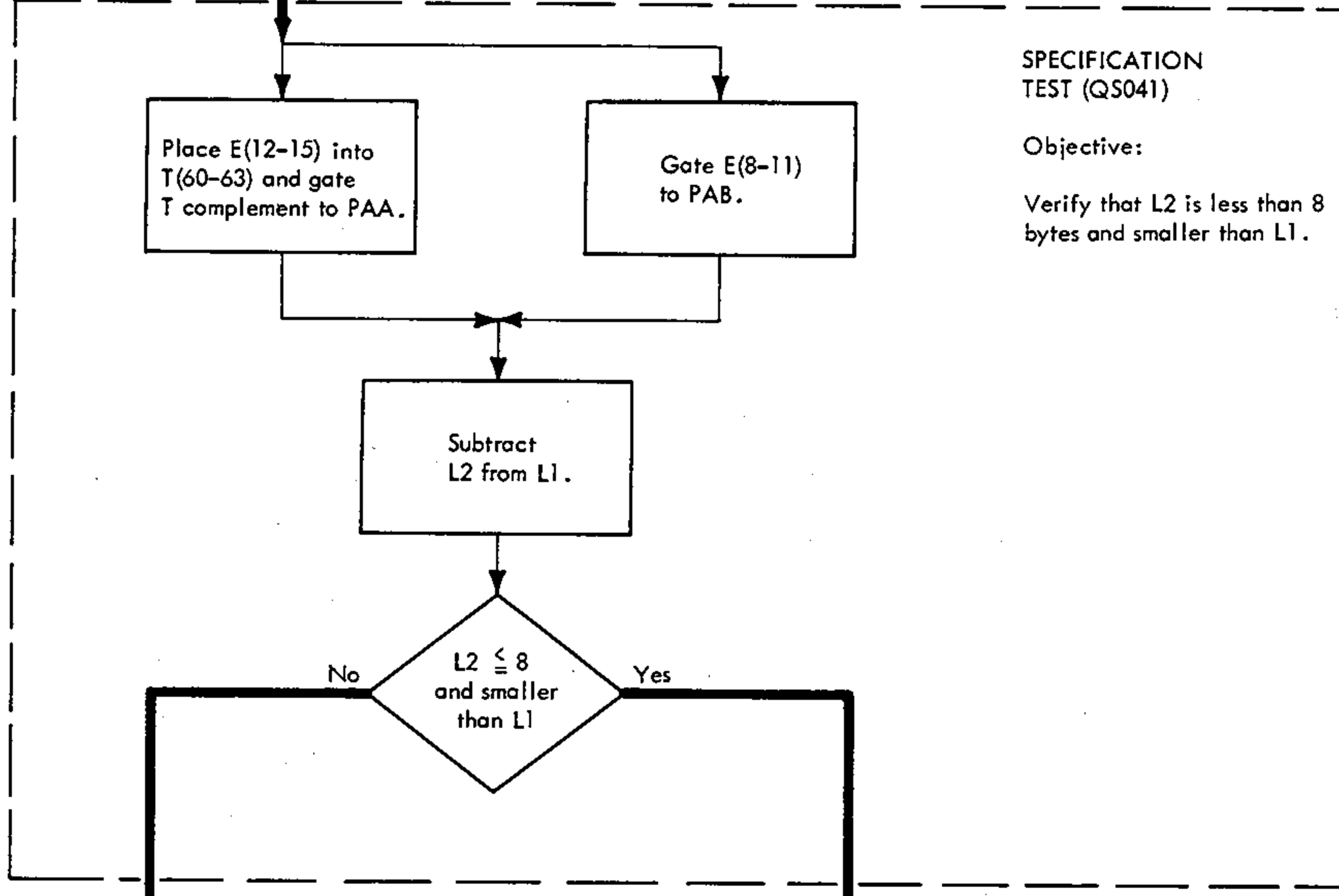
• Purpose: Multiply 1st operand (in storage) by 2nd operand (in storage) and place result into 1st operand location.

- 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).



GIS (QT031 and Q5041)  
Objectives:  
1. Place multiplier into AB.  
2. Initiate sign handling.  
3. Save low-order digit of multiplicand in F.

SPECIFICATION TEST (Q5041)  
Objective:  
Verify that L2 is less than 8 bytes and smaller than L1.

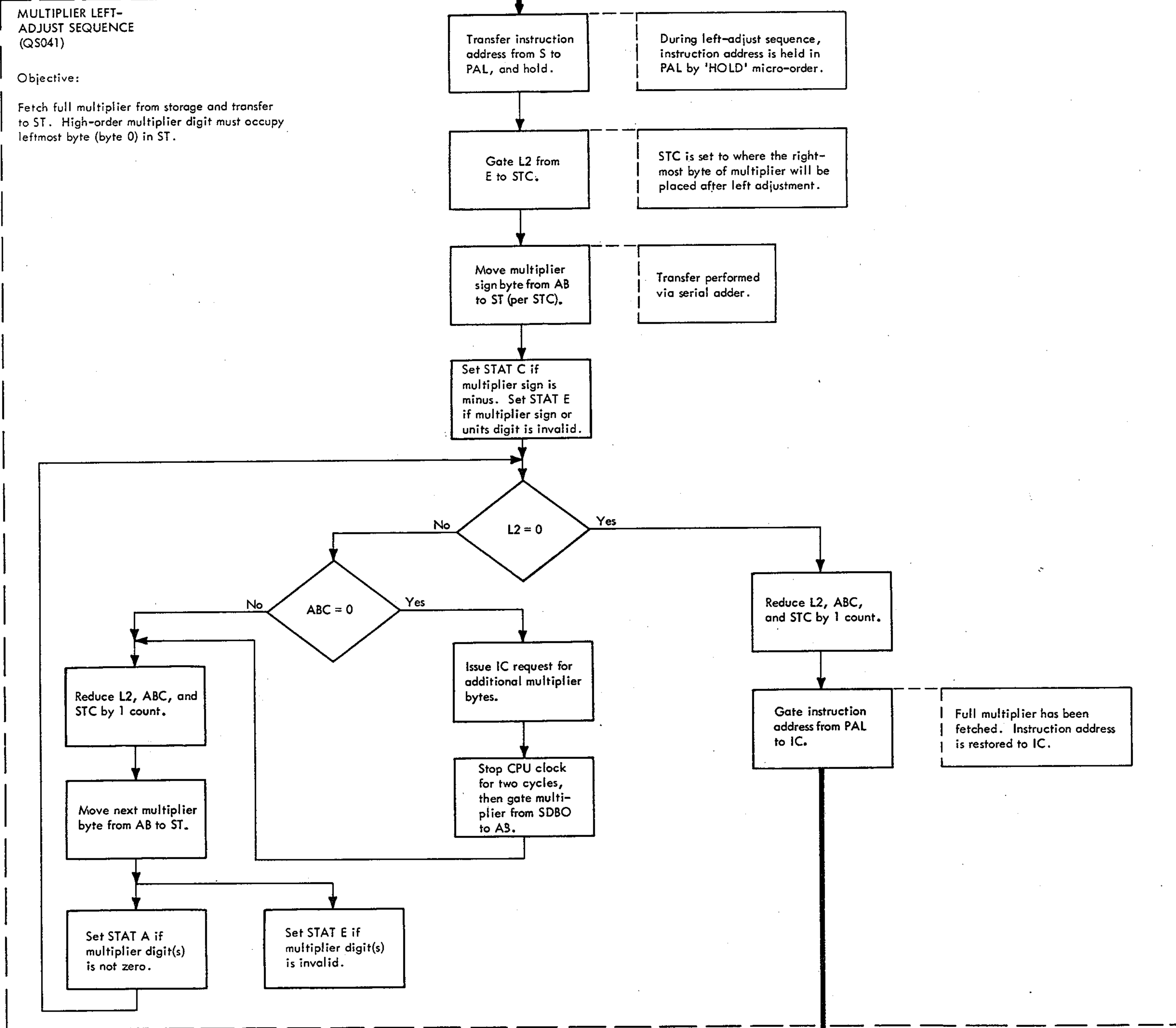


Sheet 3

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

Sheet 2

A



Sheet 4

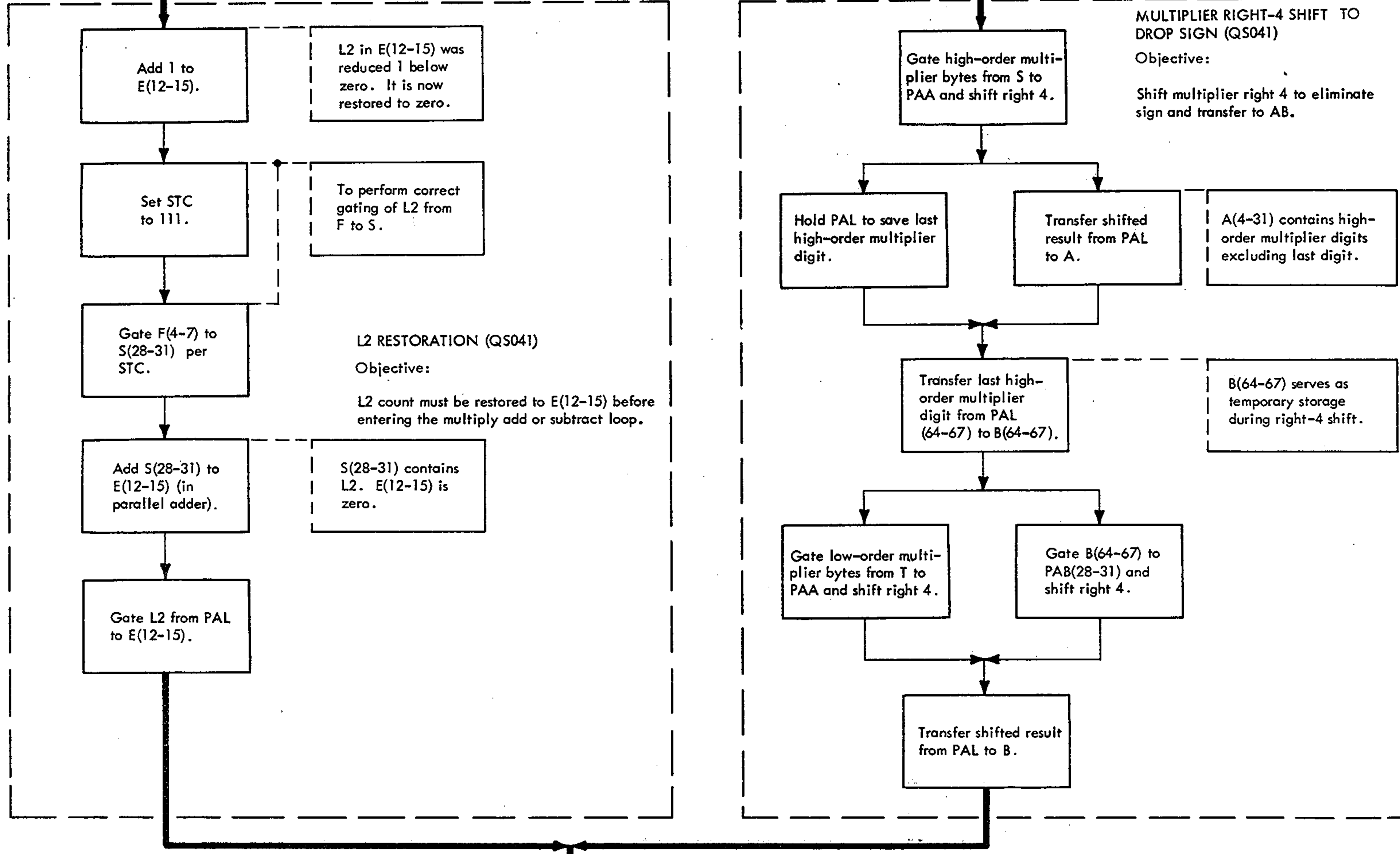


Diagram 5-305. Decimal Multiply (Sheet 3 of 7)

A

B

C

D

E

F

G

H

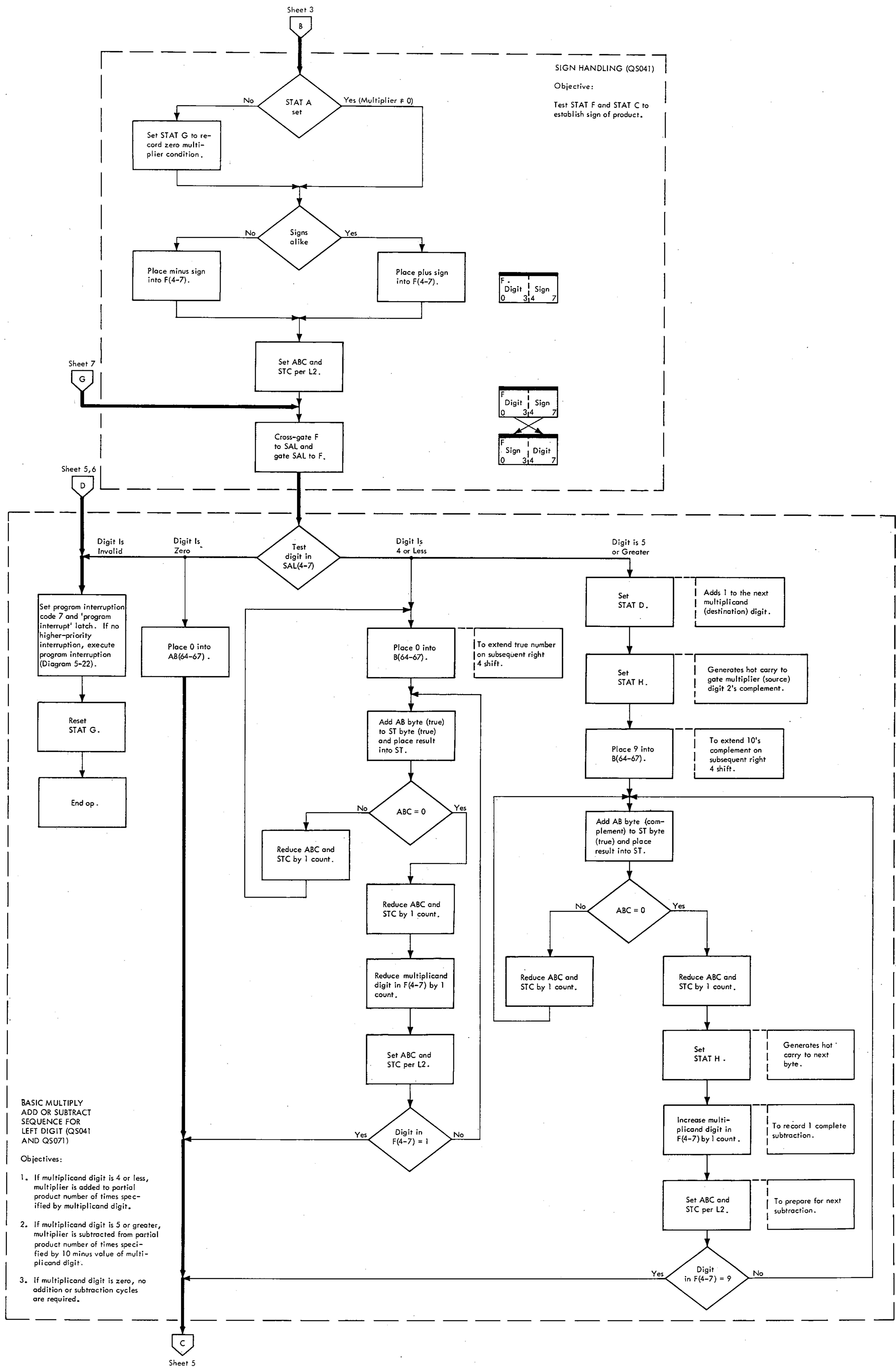


Diagram 5-305. Decimal Multiply (Sheet 4 of 7)

Sheet 4

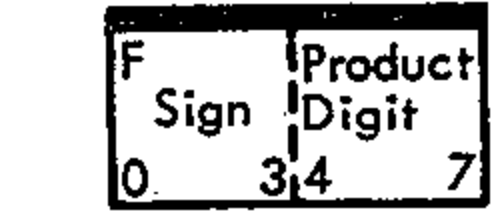
C

A

Set D(21-23) per STC.

To designate product byte for store operation.

Transfer low-order partial product digit from ST to F(4-7).



Reduce LI by 1 count.

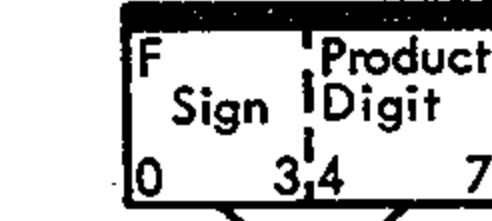
To indicate that 1 byte has been processed.

B

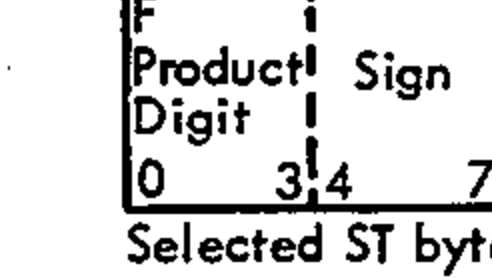
Issue D request to store low-order partial product byte.

PRODUCT BYTE STORE (QS071)  
Objective:  
Store low-order partial product byte into main storage.

Cross-gate F through serial adder and gate to ST per STC.



Set Mark trigger per STC.



C

D

Place 0000 0001 into F.

Subtract 1 from multiplicand address in D.

Add F to D (complement) in parallel adder.

Complement result and gate to D.

MULTIPLICAND REQUEST (QS071)  
Objectives:  
1. Reduce D address by 1 and request multiplicand from storage.  
2. Set STC to point at new multiplicand byte.

Request multiplicand per D address.

Set STC per D(21-23).

E

F

G

Sheet 6

E

Transfer low-order half of partial product from T to LSWR.

Gate S(0-31) to PAB(32-63).

Gate B(64-67) to PAA(28-31).

B(64-67) was set to 0 if partial product is true, or to 9 if complement.

Shift parallel adder right 4 to PAL(32-67).

PAL(32-35) is extended with 0 or 9.

Gate PAL(64-67) to B(64-67).

To save low-order digit of partial product from S.

Hold PAL to save high-order partial product from S.

STAT E set

Yes (Invalid Digit or Sign)

Transfer high-order partial product from PAL to T.

Original S contents excluding low-order digit.

Transfer low-order partial product from LSWR to S.

Original T contents.

STAT G set

Yes (Multiplicand = 0)

Transfer high-order half of partial product from T to LSWR.

S and T contents must be interchanged.

Gate S(0-31) to PAA(32-63) and B(64-67) to PAB(28-31).

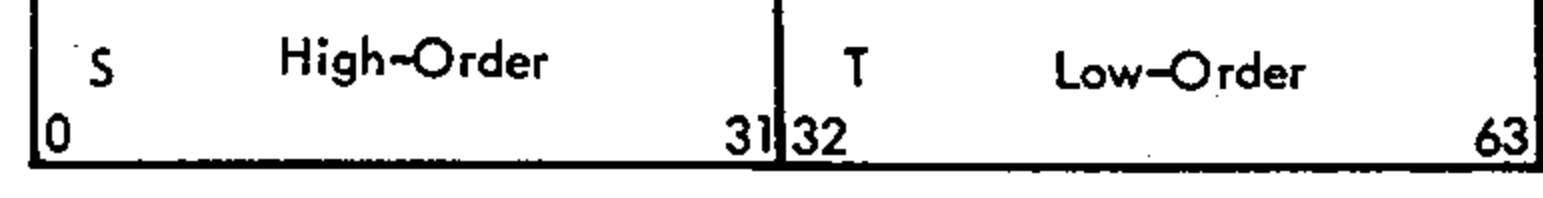
B contains original low-order digit from S.

Shift parallel adder right 4 to PAL.

Low-order digit from S inserted into T.

Gate PAL to T.

Transfer high-order partial product from LSWR to S.



PARTIAL PRODUCT RIGHT-4 SHIFT TO DROP DIGIT (QS071)

Objective:

Shift partial product right 4 (to drop low-order digit) and place into ST.

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

Sheet 5

A  
B  
C  
D  
E  
F  
G  
H

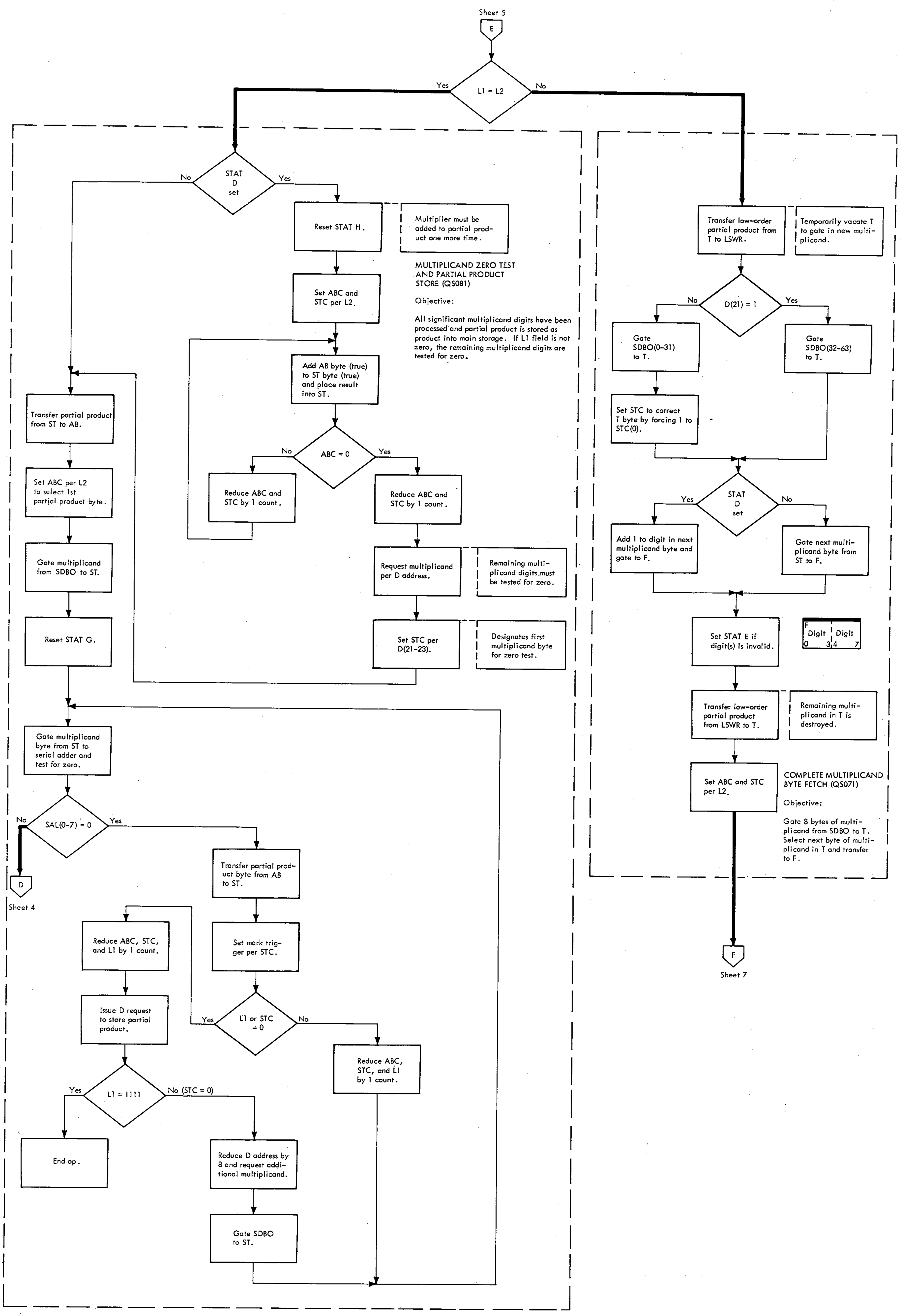
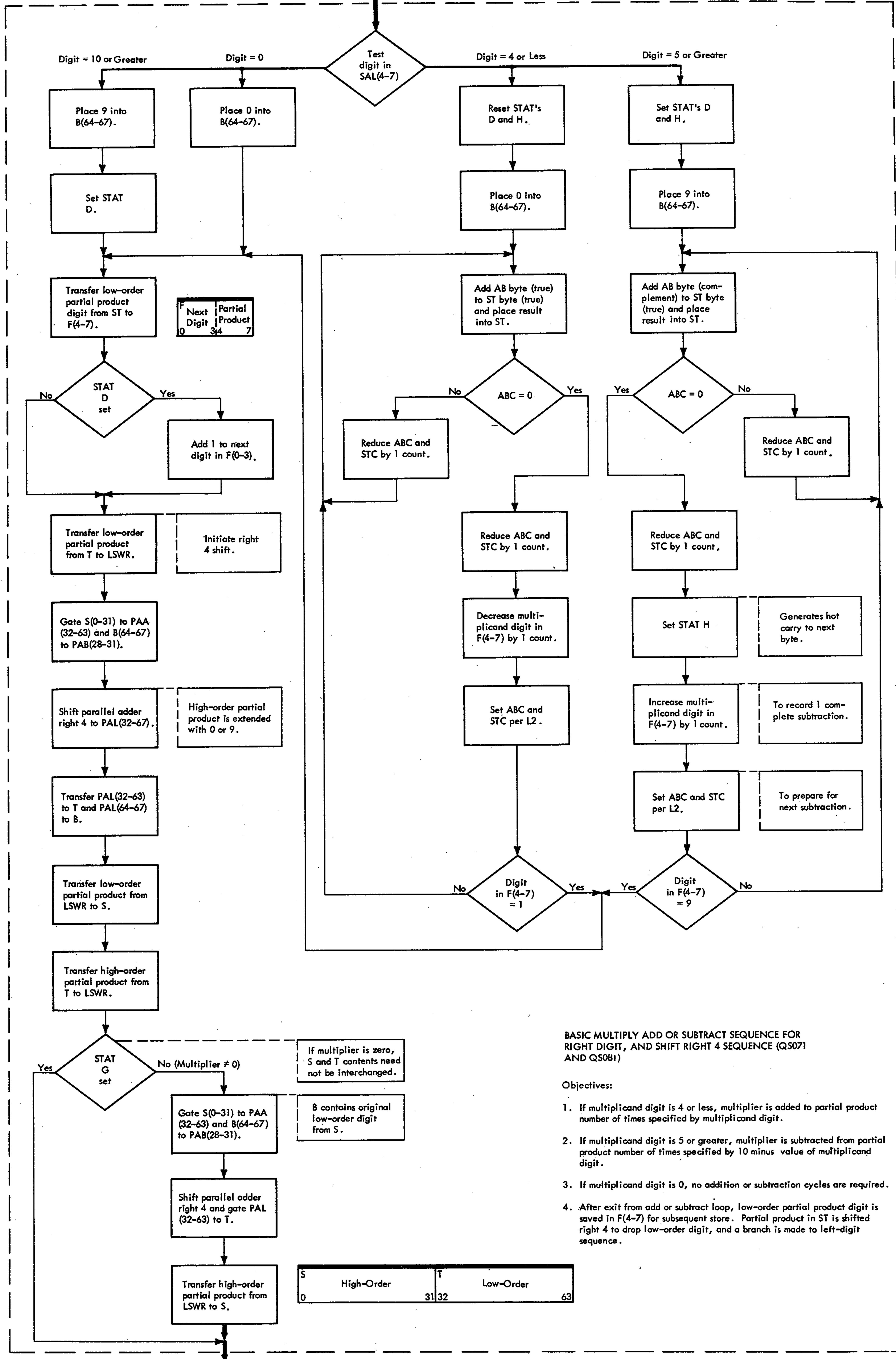


Diagram 5-305. Decimal Multiply (Sheet 6 of 7)

Sheet 6  
F

A  
B  
C  
D  
E  
F  
G  
H



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

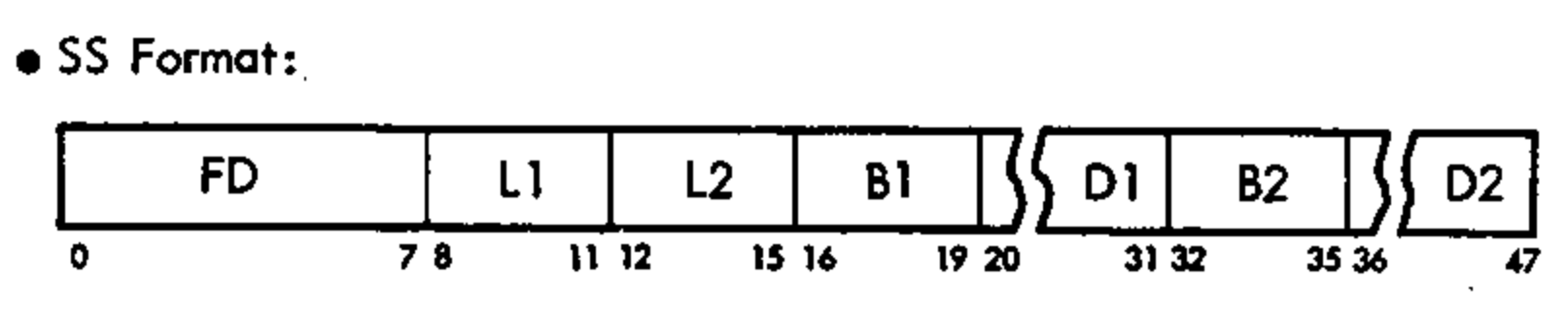
**Objectives:**

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.

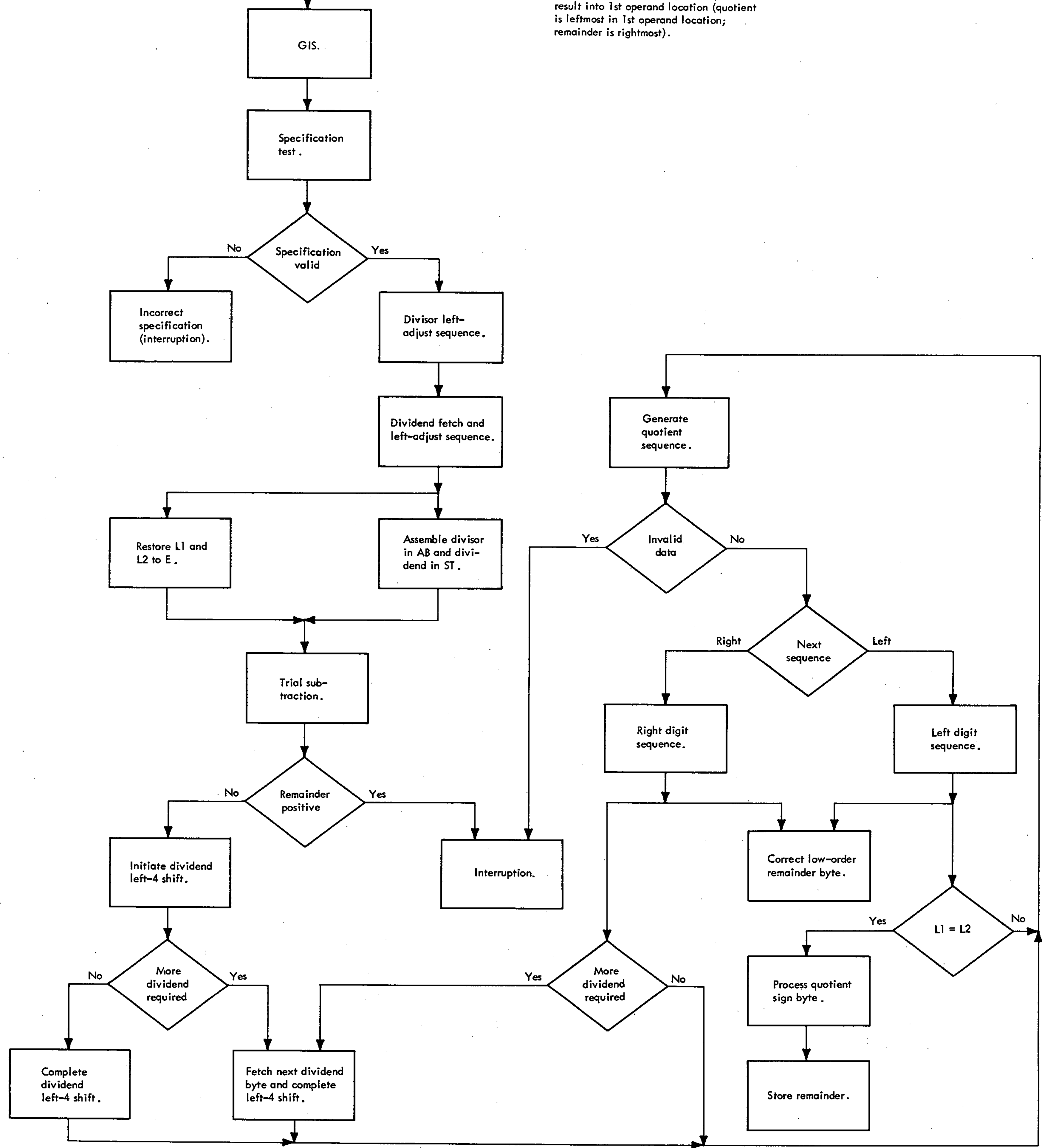
Diagram 5-305. Decimal Multiply (Sheet 7 of 7)

A  
B  
C  
D  
E  
F  
G  
H

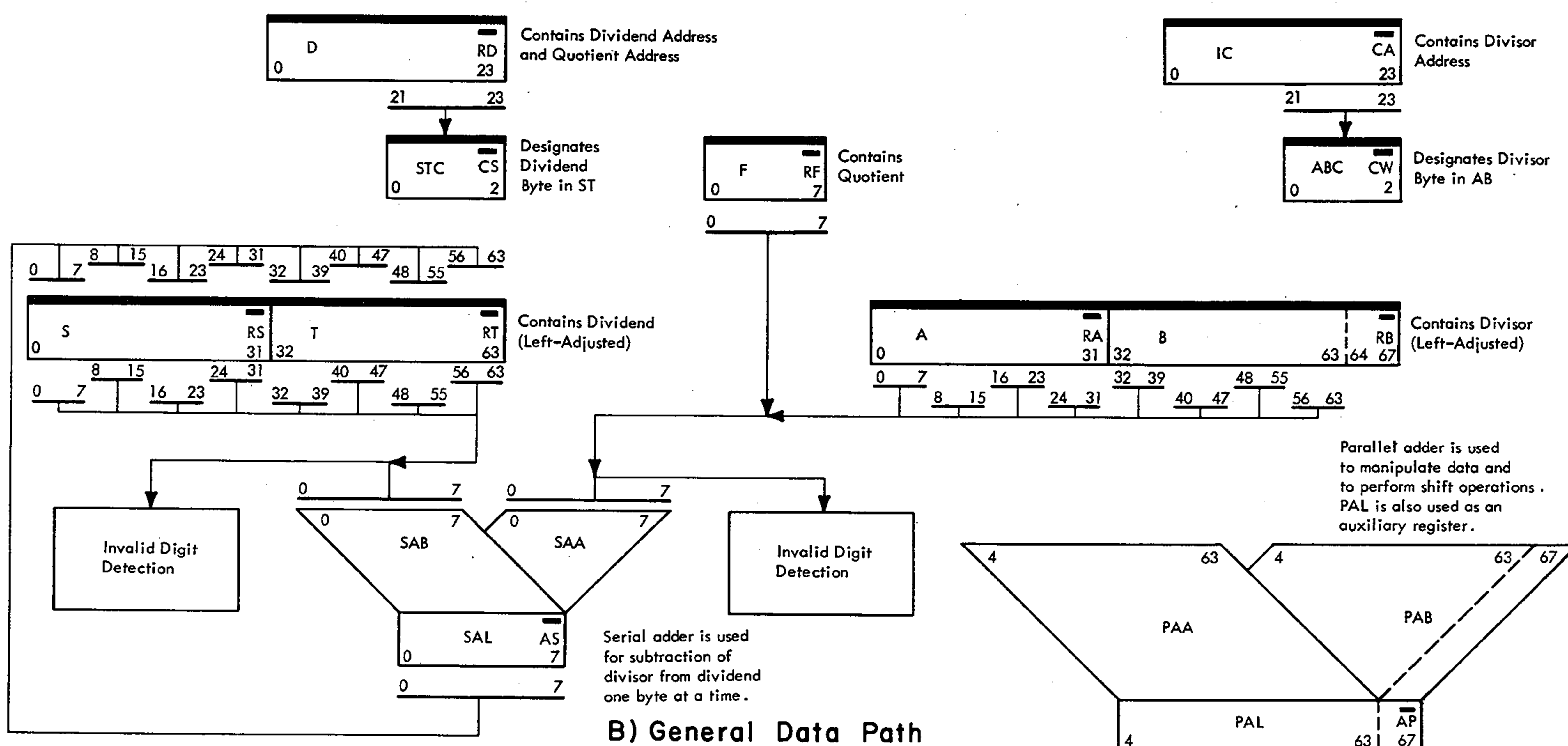
Diagram 5-14  
SS I-Fetch.



• 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).



A) Overall Flow Chart

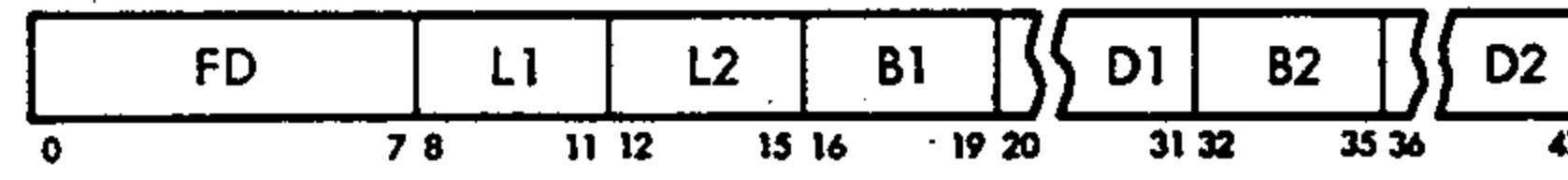


B) General Data Path



Diagram 5-14  
SS I-Fetch.

• SS Format:



• 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).

A

GIS (QT031 AND QS041)

Objectives:

1. Place divisor into AB and set ABC to the low-order divisor byte. Save L1 and L2 in F.
2. Test dividend sign for validity.

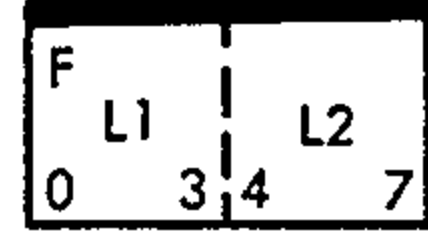
QT031  
Gate 1st operand (dividend) from SDBO to ST.

Add L2 to IC and request 2nd operand (divisor). Transfer D to STC.

Transfer L1 and L2 from E(8-15) to F(0-7).

E to B and then to F. ABC set to 111 to select B byte.

Reduce IC by 8 to address next doubleword of divisor.



QS041  
Divide

Yes  
Gate divisor from SDBO to AB. Set STAT E if dividend sign is invalid; set STAT F if minus.

No  
Diagram 5-305  
Multiply operation.

Subtract L1 from D to set address of high-order byte. Reset STAT F.

Gate L2 from E to STC.

STC set to where rightmost divisor byte is placed after left adjustment.

No  
STC ≥ 4

Yes  
Set STAT G to record that divisor is 5 bytes or greater.

Transfer instruction address from LSWR to S.

B

C

D

E

F

Sheet 3

G

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

H

A

B

C

D

E

F

G

H

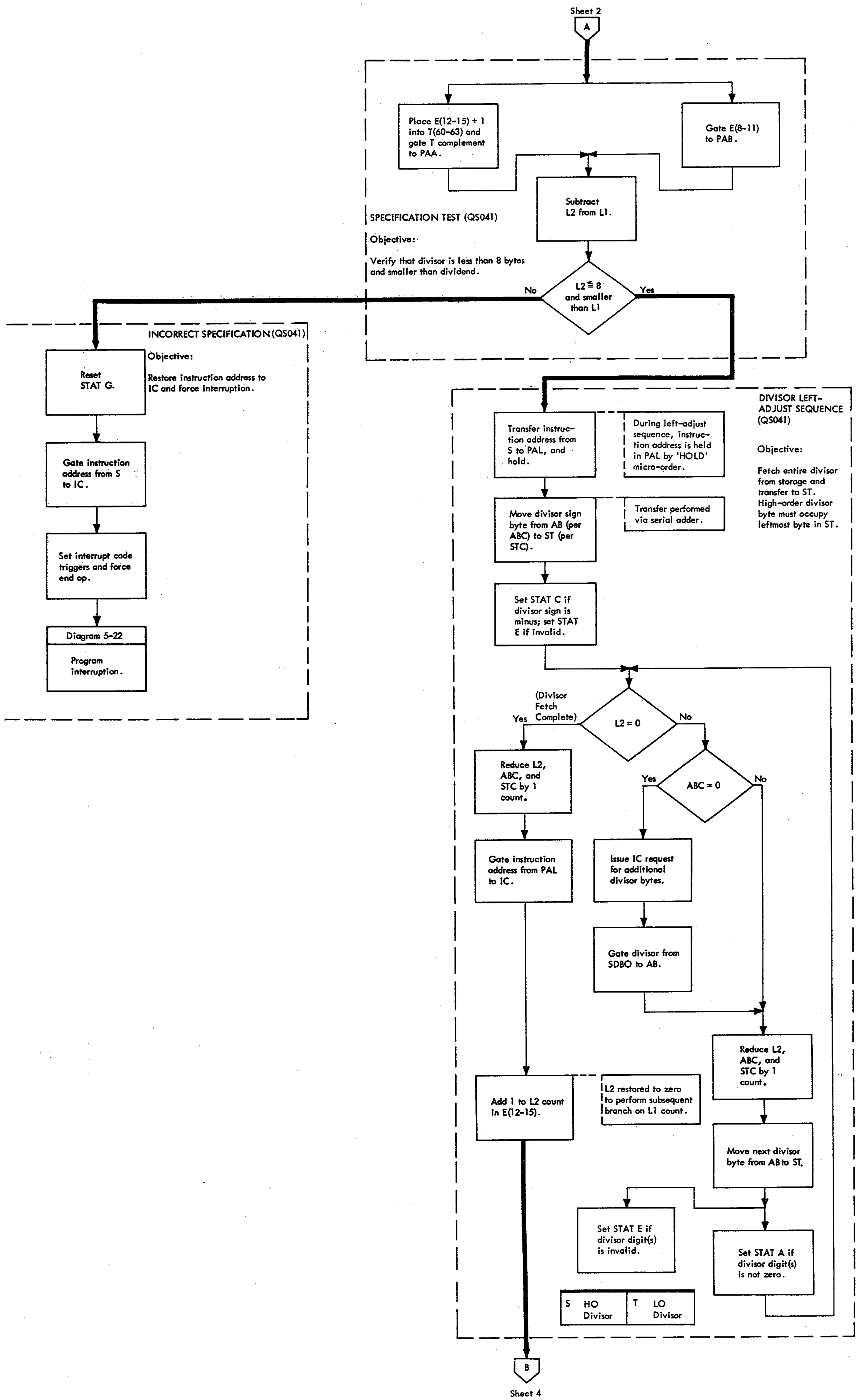


Diagram 5-306. Decimal Divide (Sheet 3 of 9)

Sheet 3

DIVIDEND FETCH AND LEFT-ADJUST SEQUENCE (QS041 AND QS051)

Objectives:

Divisor is shifted right 4 and held in PAL and LSWR. Sufficient number of dividend bytes is fetched to perform trial subtraction (at most, two dividend fetches are required). Dividend bytes are placed into ST, where high-order byte occupies leftmost position. Original D address is restored to D to reflect quotient address.

A

B

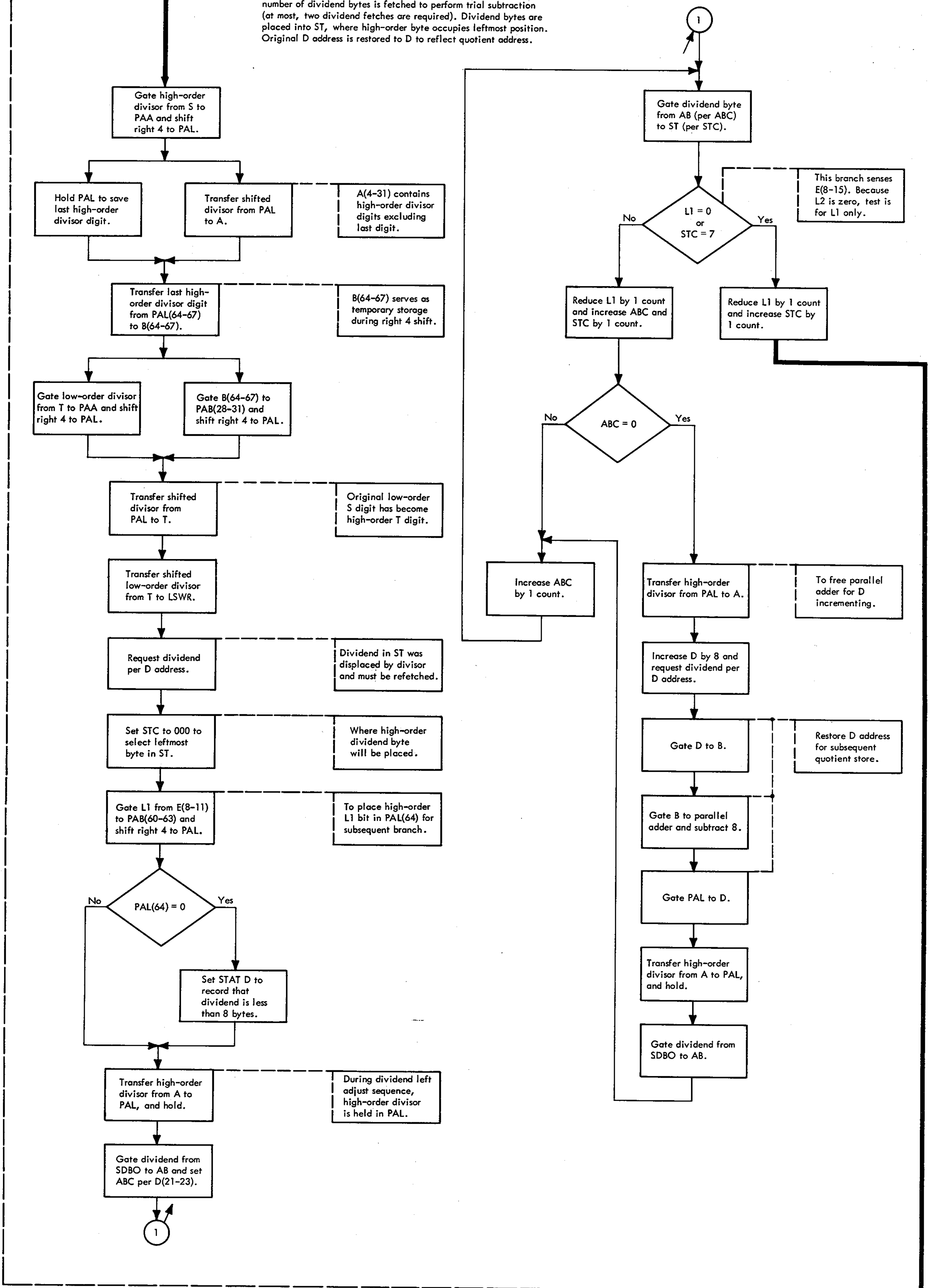
C

D

E

F

G



Sheet 5

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

A

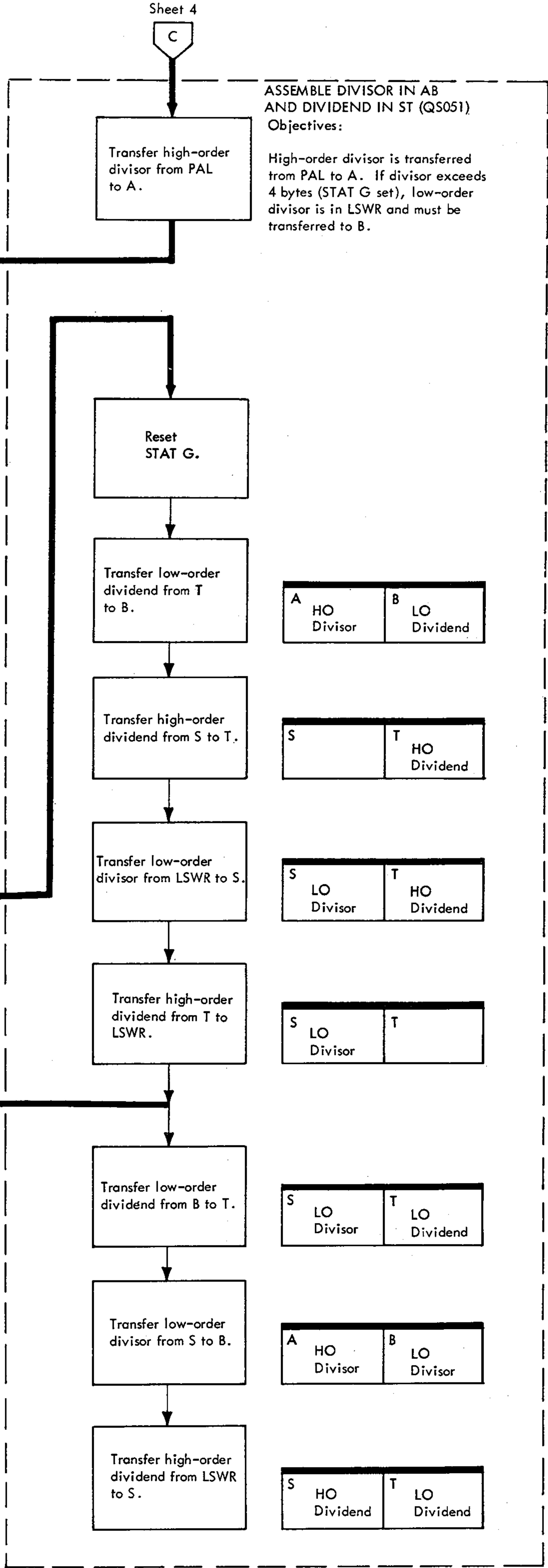
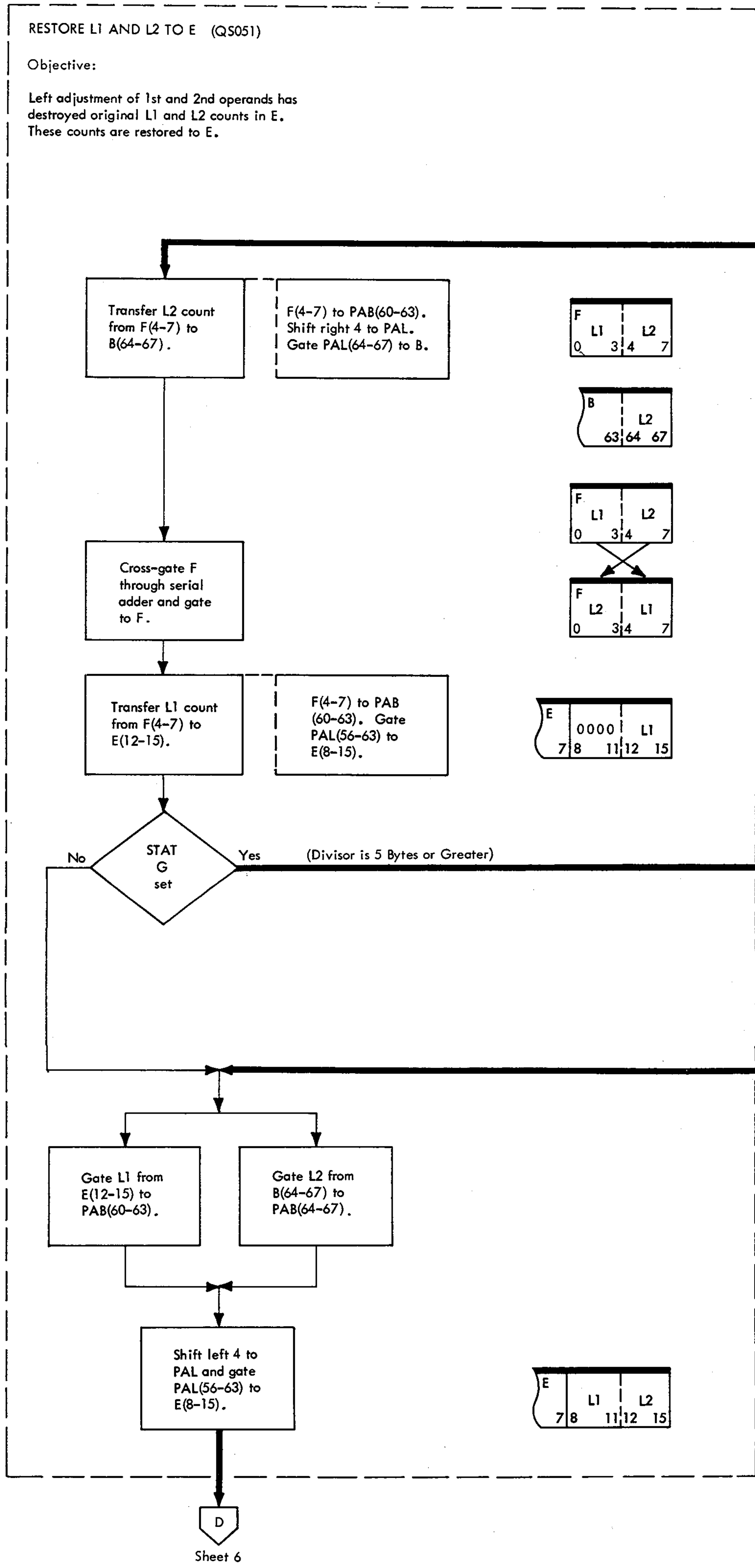
B

C

D

E

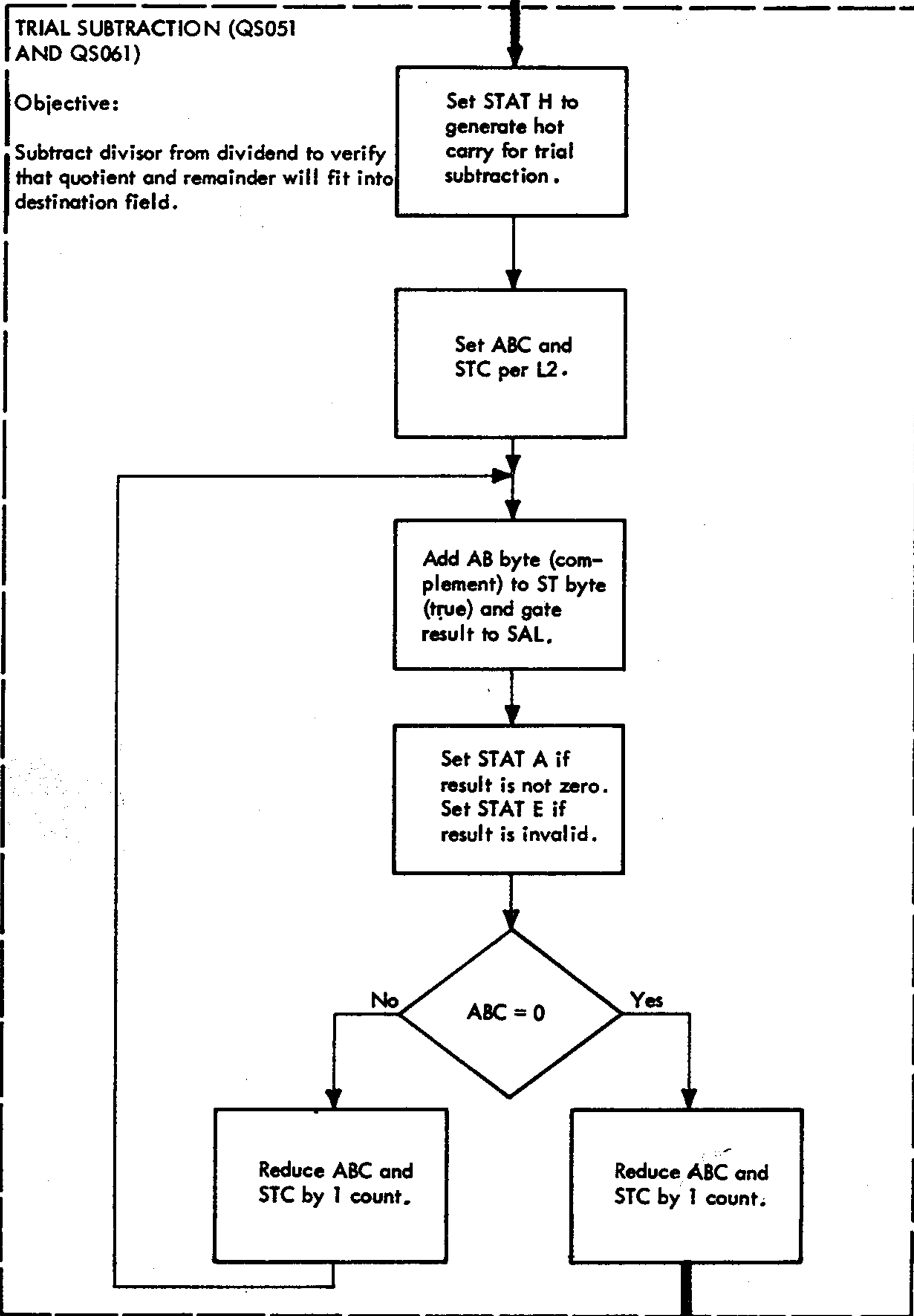
F



Sheet 6

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

Sheet 5



Yes  
 Carry from SAL(0)  
 No

Sheets 7, 8, 9

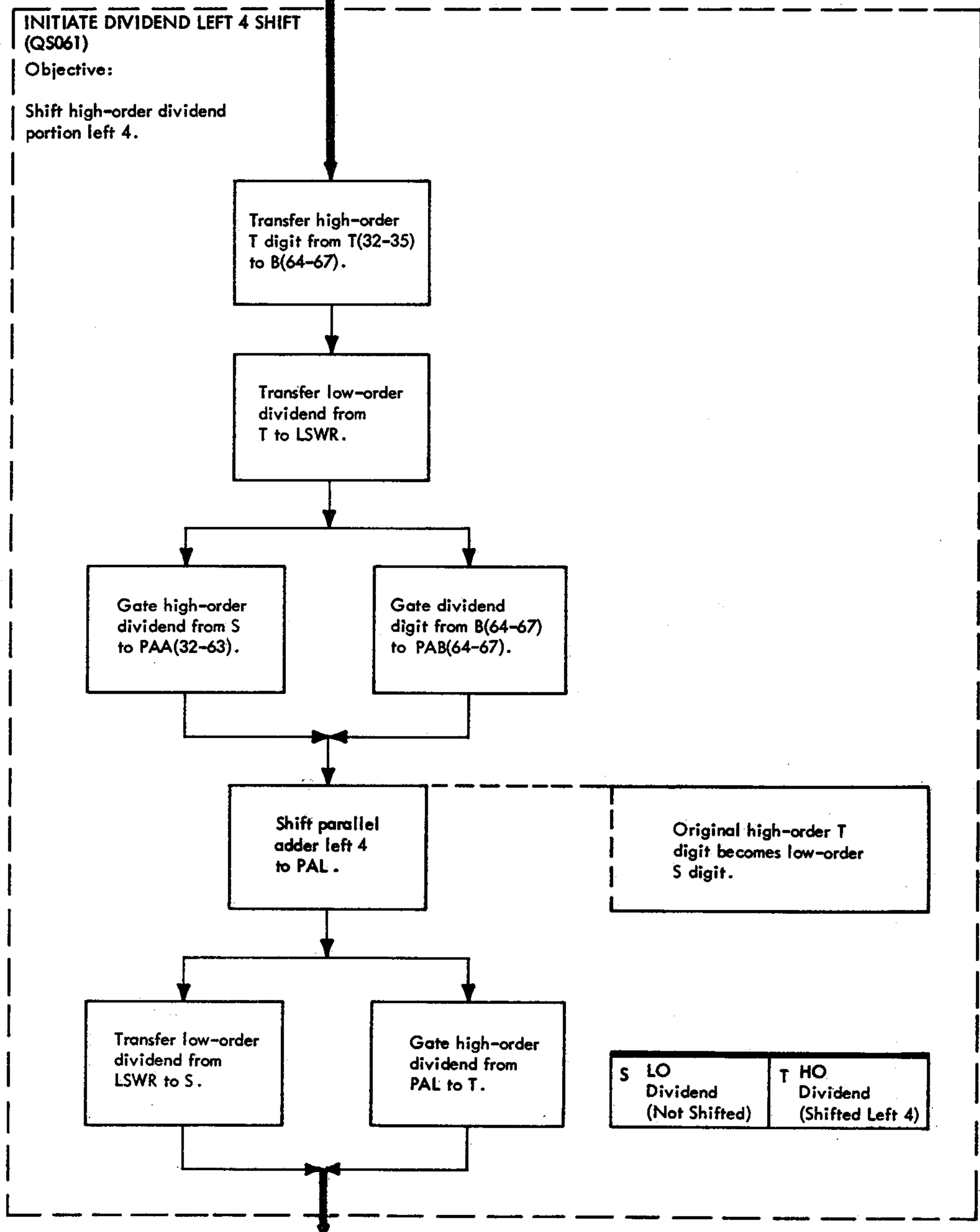
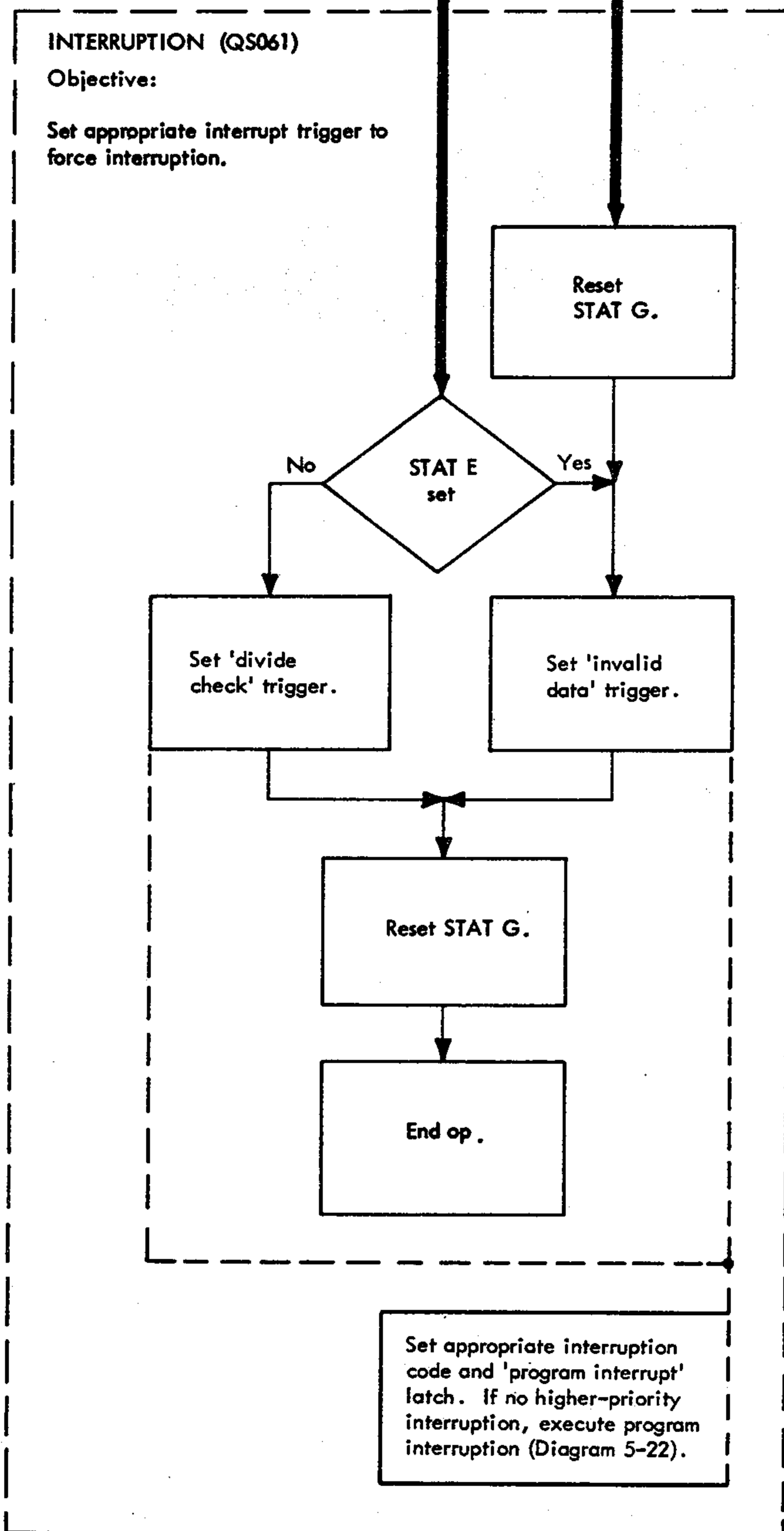


Diagram 5-306. Decimal Divide (Sheet 6 of 9)

Sheet 7

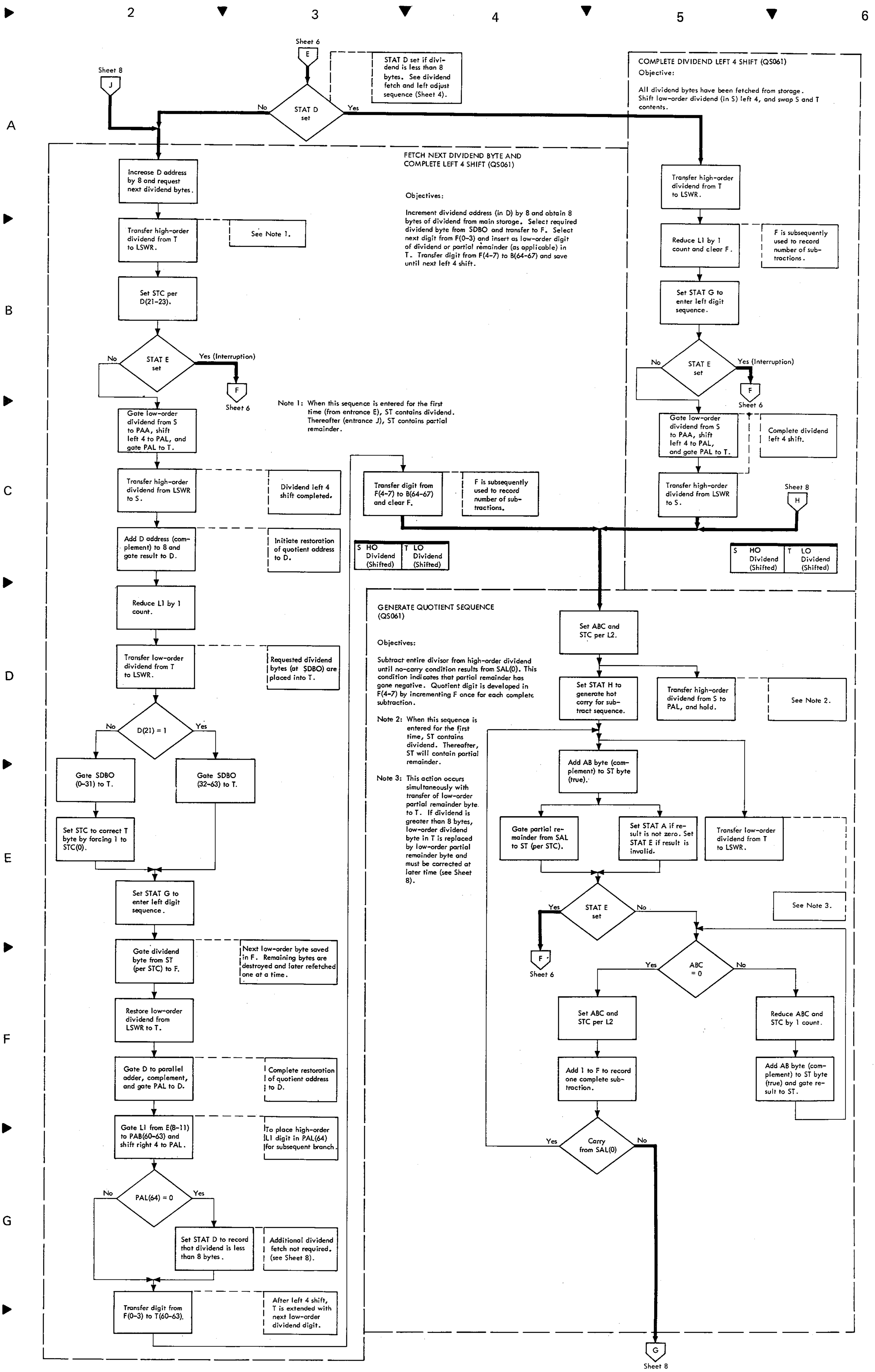


Diagram 5-306. Decimal Divide (Sheet 7 of 9)

Sheet 7  
G

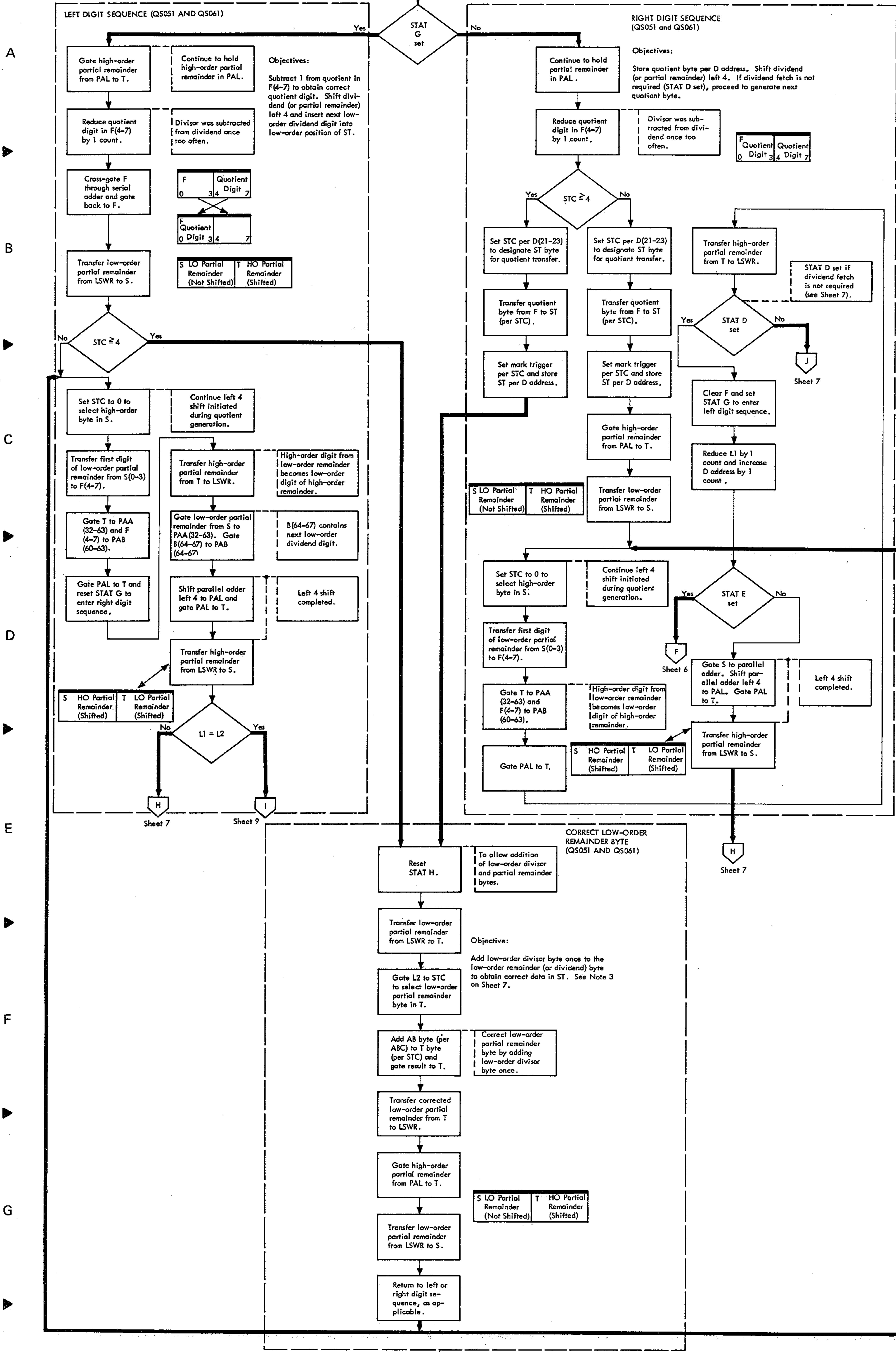


Diagram 5-306. Decimal Divide (Sheet 8 of 9)

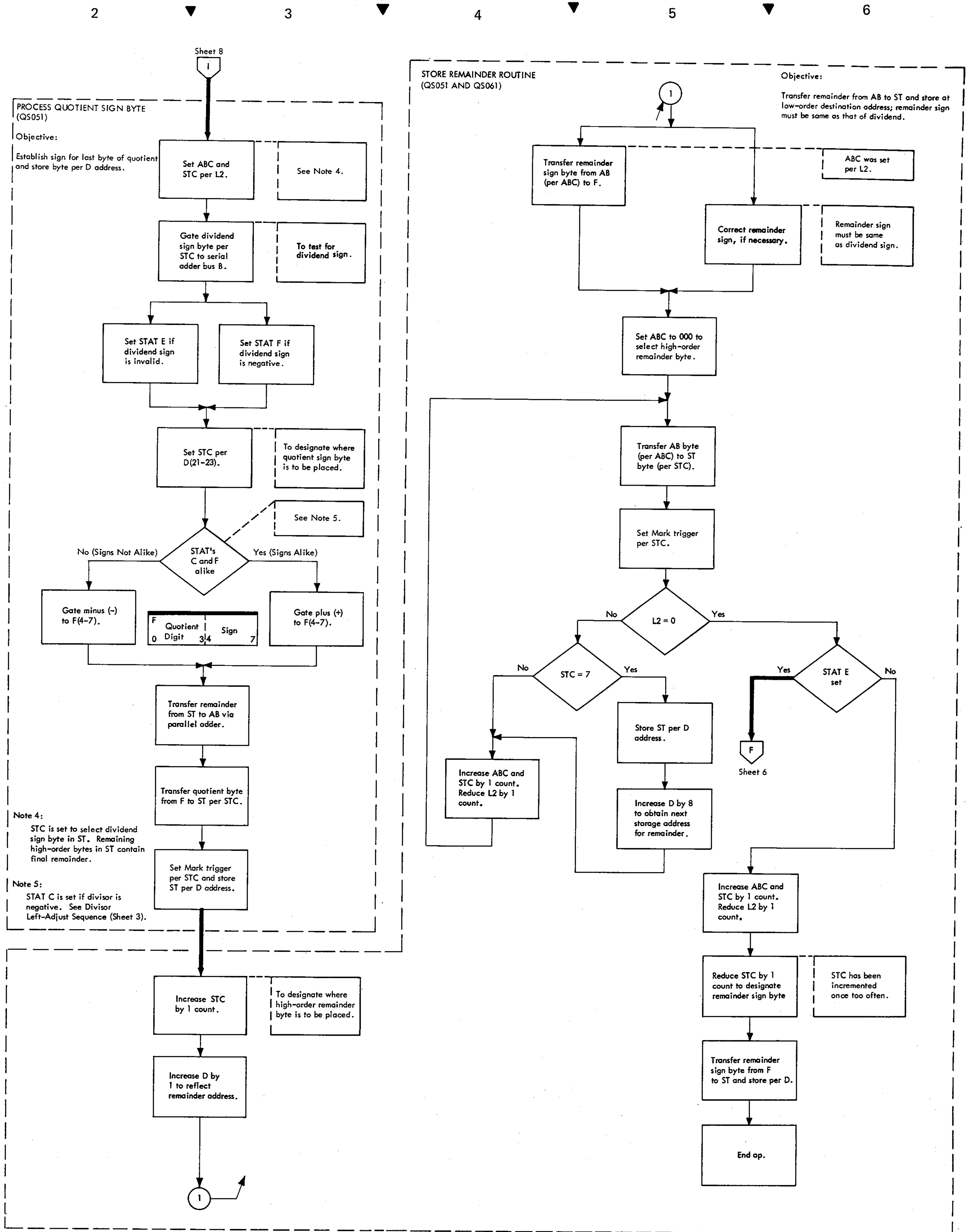


Diagram 5-306. Decimal Divide (Sheet 9 of 9)