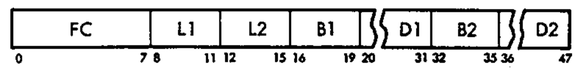
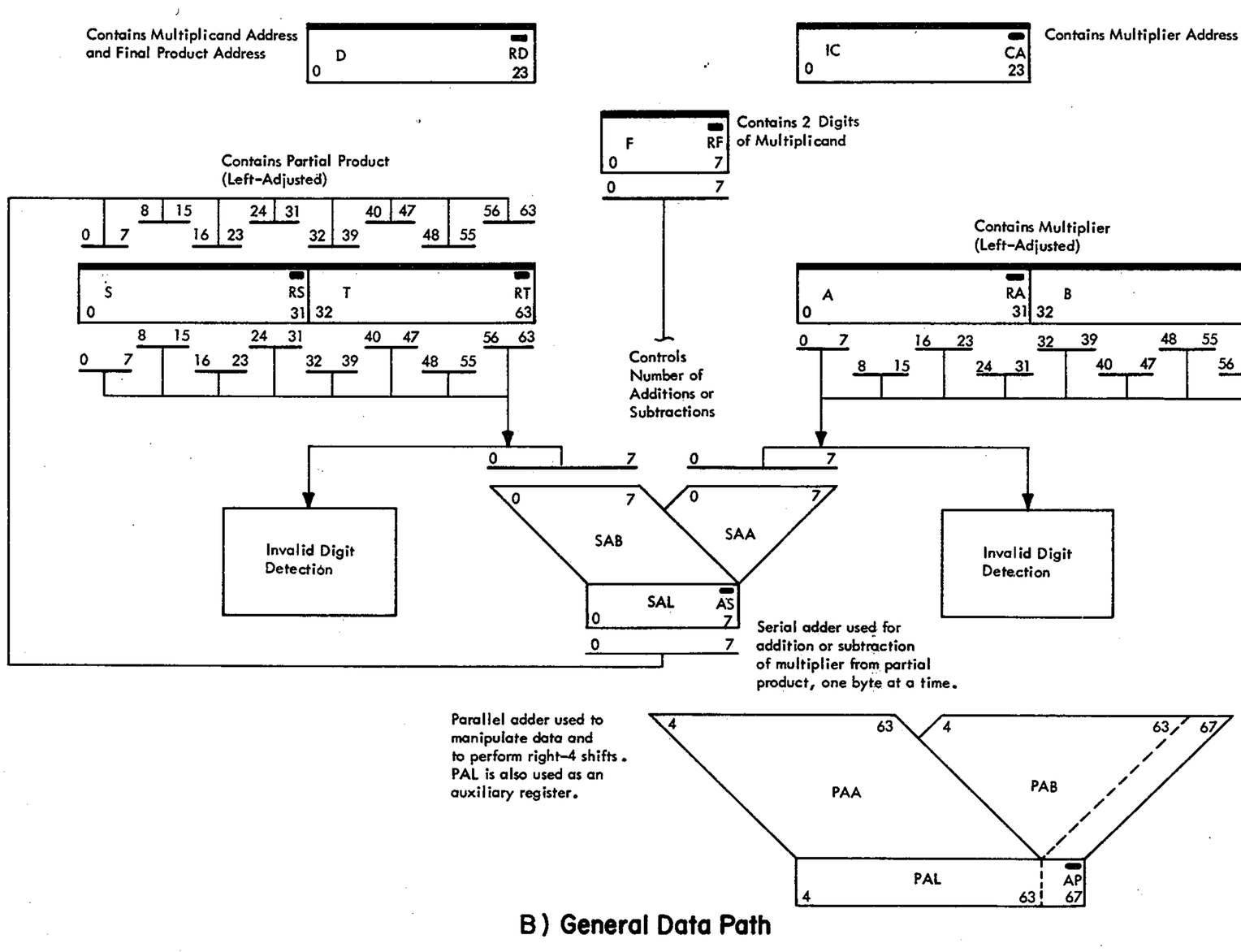
- SS Format:



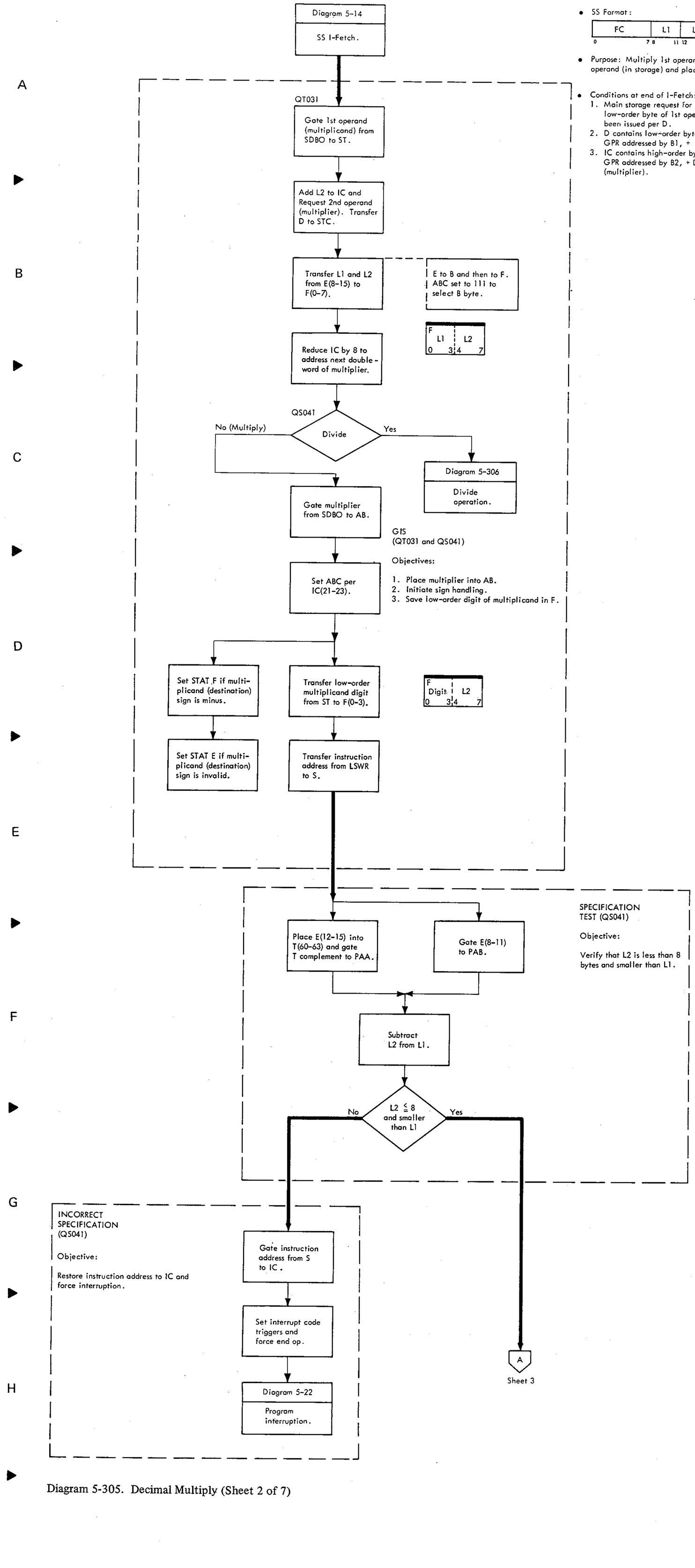
- Purpose: Multiply 1st operand (in storage) by 2nd operand (in storage) and place result into 1st operand location.

A) Overall Flow Chart

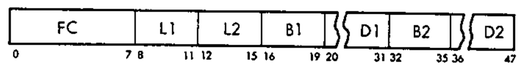


B) General Data Path

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



- SS Format:



- 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)

Sheet 2

A

MULTIPLIER LEFT-ADJUST SEQUENCE (QS041)

Objective:

Fetch full multiplier from storage and transfer to ST. High-order multiplier digit must occupy leftmost byte (byte 0) in ST.

Transfer instruction address from S to PAL, and hold.

During left-adjust sequence, instruction address is held in PAL by 'HOLD' micro-order.

Gate L2 from E to STC.

STC is set to where the right-most byte of multiplier will be placed after left adjustment.

Move multiplier sign byte from AB to ST (per STC).

Transfer performed via serial adder.

Set STAT C if multiplier sign is minus. Set STAT E if multiplier sign or units digit is invalid.

L2 = 0

No

Yes

ABC = 0

No

Yes

Reduce L2, ABC, and STC by 1 count.

Move next multiplier byte from AB to ST.

Set STAT A if multiplier digit(s) is not zero.

Set STAT E if multiplier digit(s) is invalid.

Issue IC request for additional multiplier bytes.

Stop CPU clock for two cycles, then gate multiplier from SDBO to AB.

Reduce L2, ABC, and STC by 1 count.

Gate instruction address from PAL to IC.

Full multiplier has been fetched. Instruction address is restored to IC.

Add 1 to E(12-15).

L2 in E(12-15) was reduced 1 below zero. It is now restored to zero.

Set STC to 111.

To perform correct gating of L2 from F to S.

Gate F(4-7) to S(28-31) per STC.

L2 RESTORATION (QS041)

Objective:

L2 count must be restored to E(12-15) before entering the multiply add or subtract loop.

Add S(28-31) to E(12-15) (in parallel adder).

S(28-31) contains L2. E(12-15) is zero.

Gate L2 from PAL to E(12-15).

MULTIPLIER RIGHT-4 SHIFT TO DROP SIGN (QS041)

Objective:

Shift multiplier right 4 to eliminate sign and transfer to AB.

Gate high-order multiplier bytes from S to PAA and shift right 4.

Hold PAL to save last high-order multiplier digit.

Transfer shifted result from PAL to A.

A(4-31) contains high-order multiplier digits excluding last digit.

Transfer last high-order multiplier digit from PAL (64-67) to B(64-67).

B(64-67) serves as temporary storage during right-4 shift.

Gate low-order multiplier bytes from T to PAA and shift right 4.

Gate B(64-67) to PAB(28-31) and shift right 4.

Transfer shifted result from PAL to B.

B

Sheet 4

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

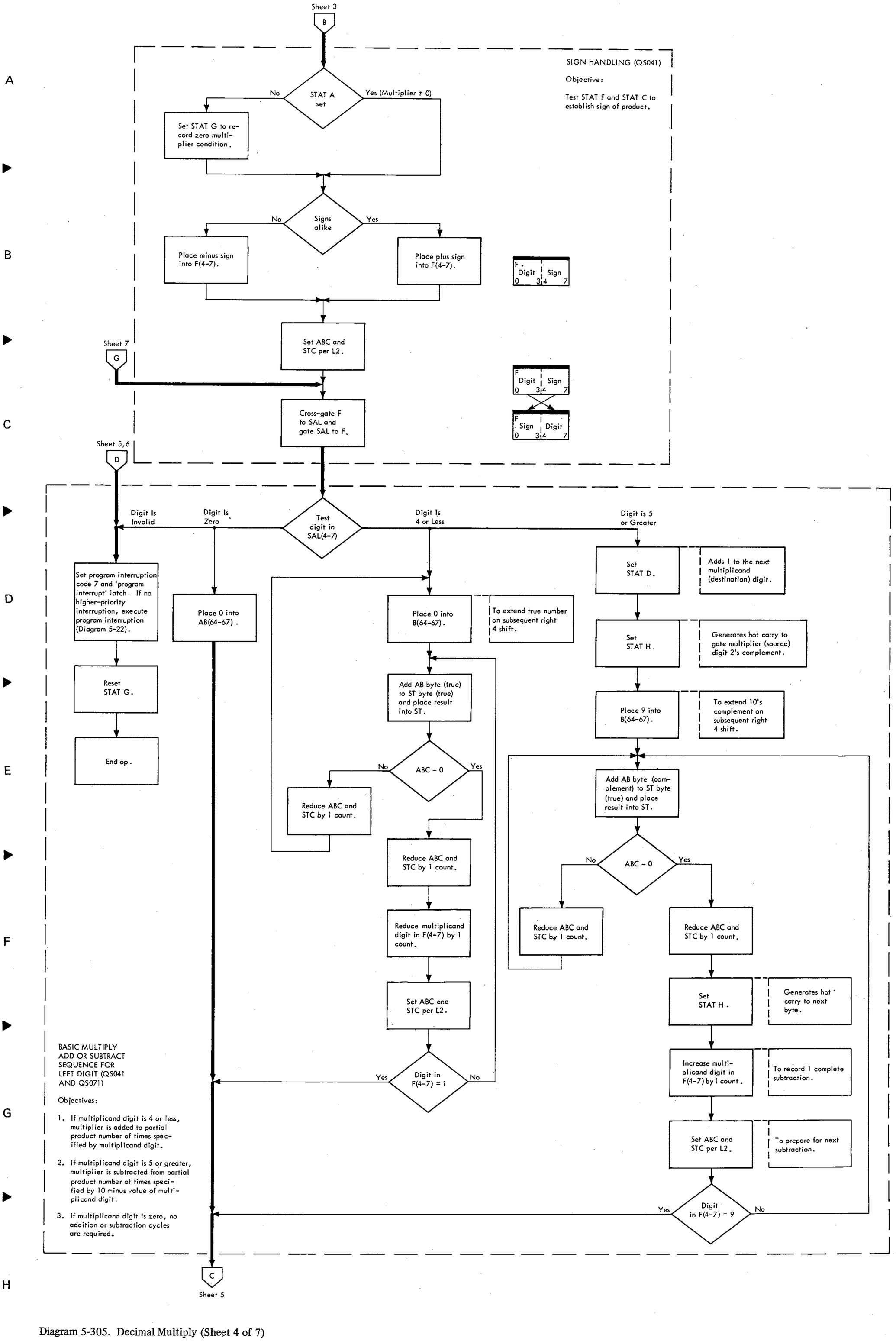


Diagram 5-305. Decimal Multiply (Sheet 4 of 7)

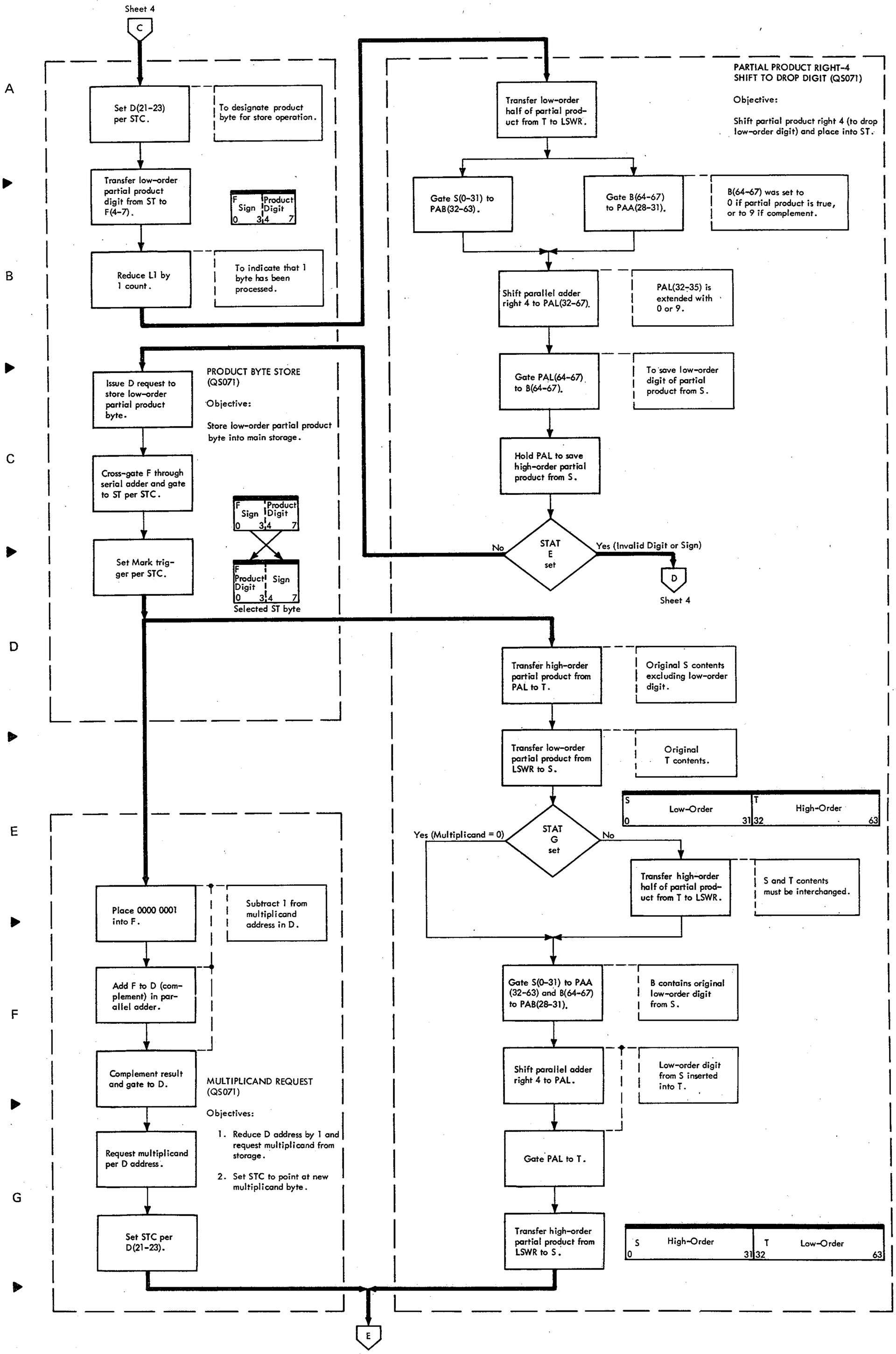


Diagram 5-305. Decimal Multiply (Sheet 5 of 7)

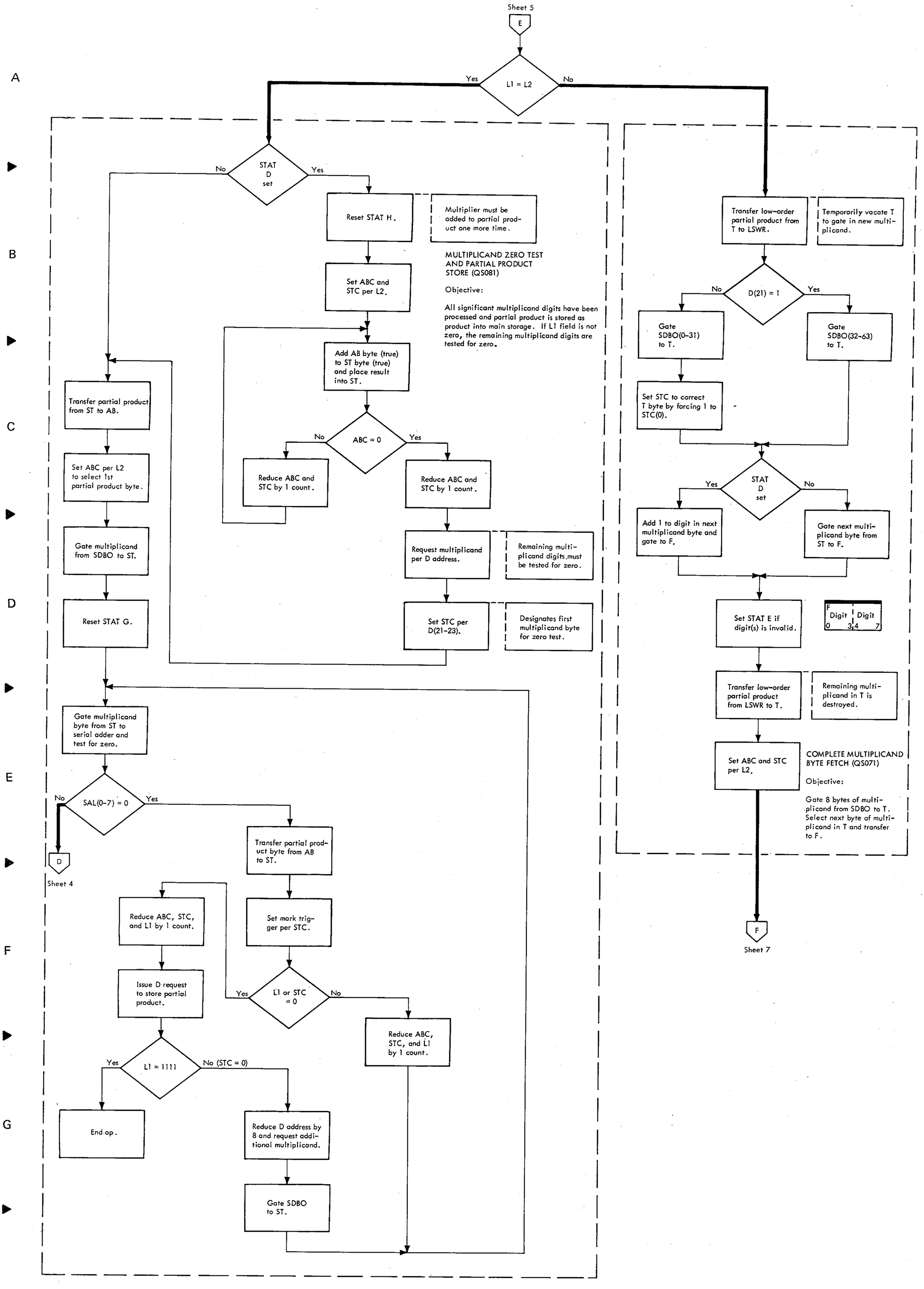
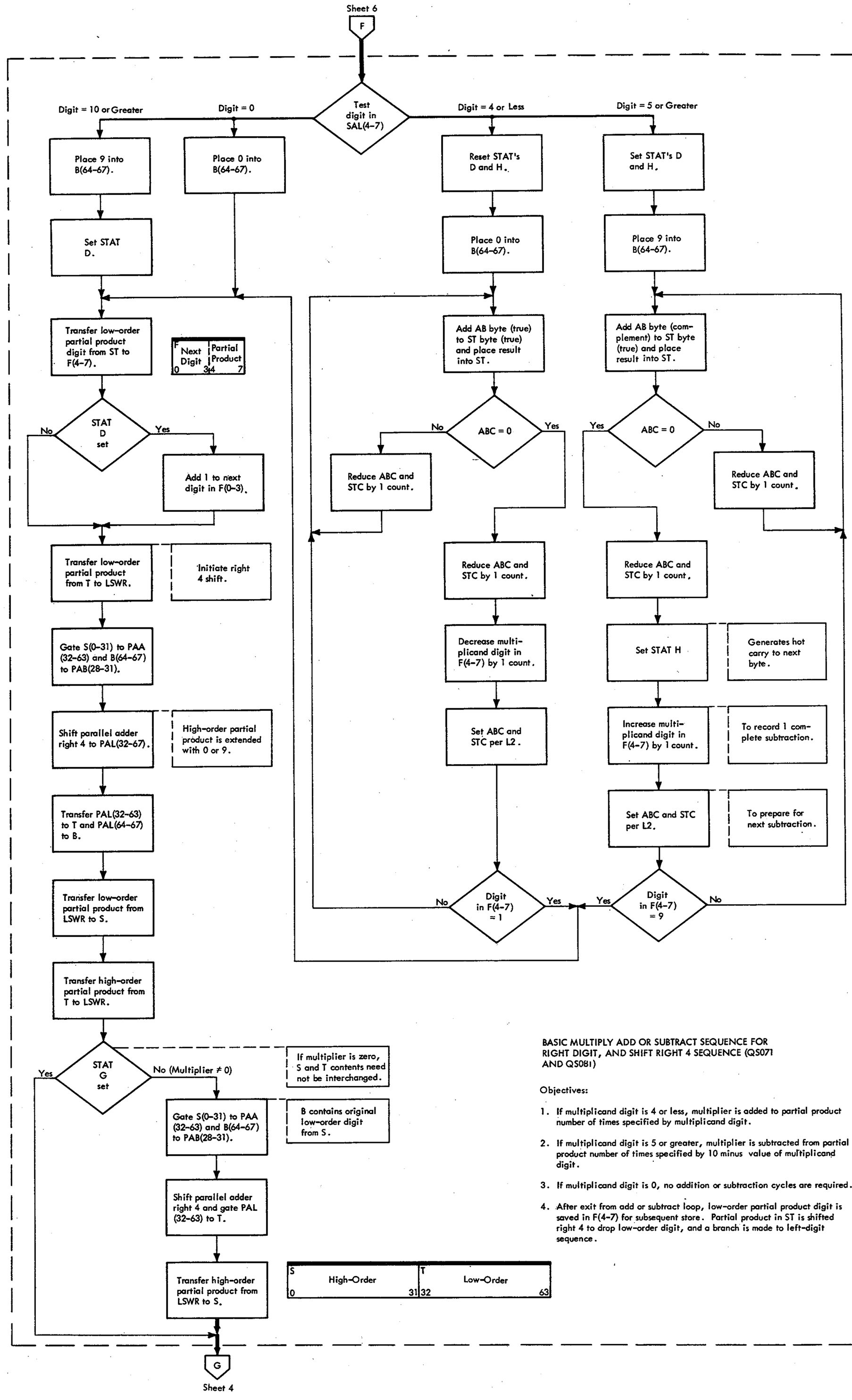


Diagram 5-305. Decimal Multiply (Sheet 6 of 7)



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)

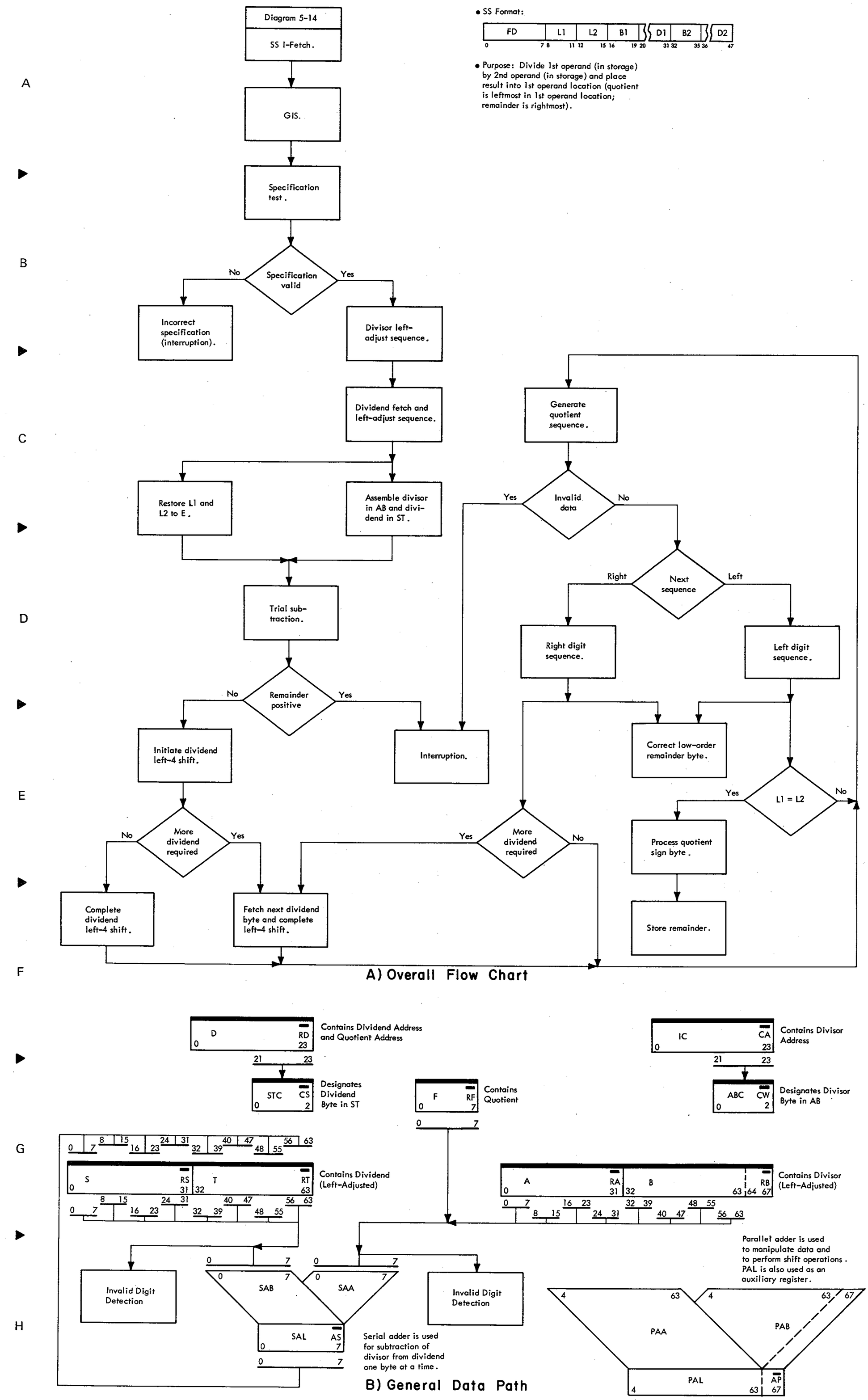
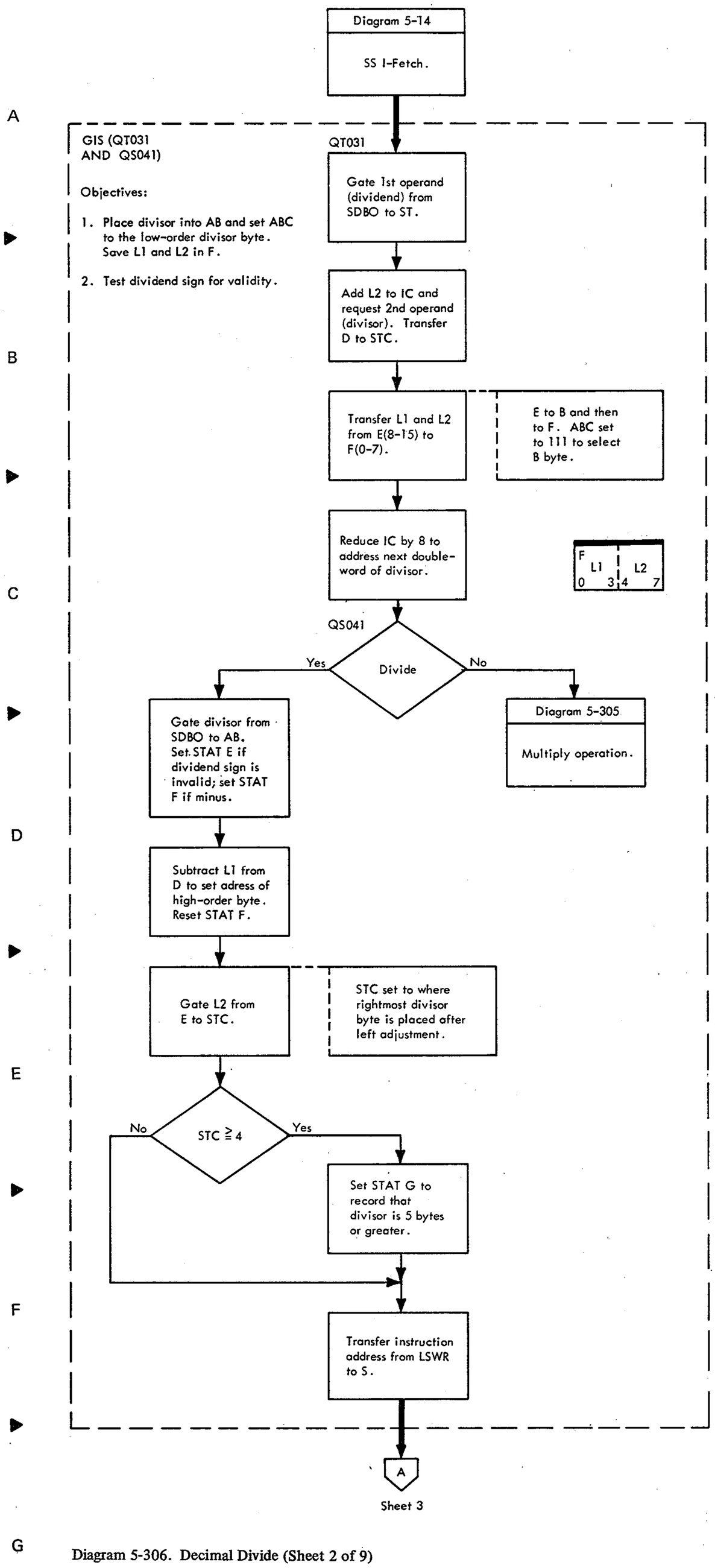
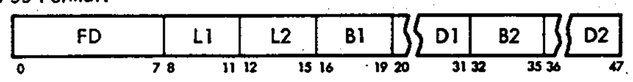


Diagram 5-306. Decimal Divide (Sheet 1 of 9)



Sheet 3

- 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).

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

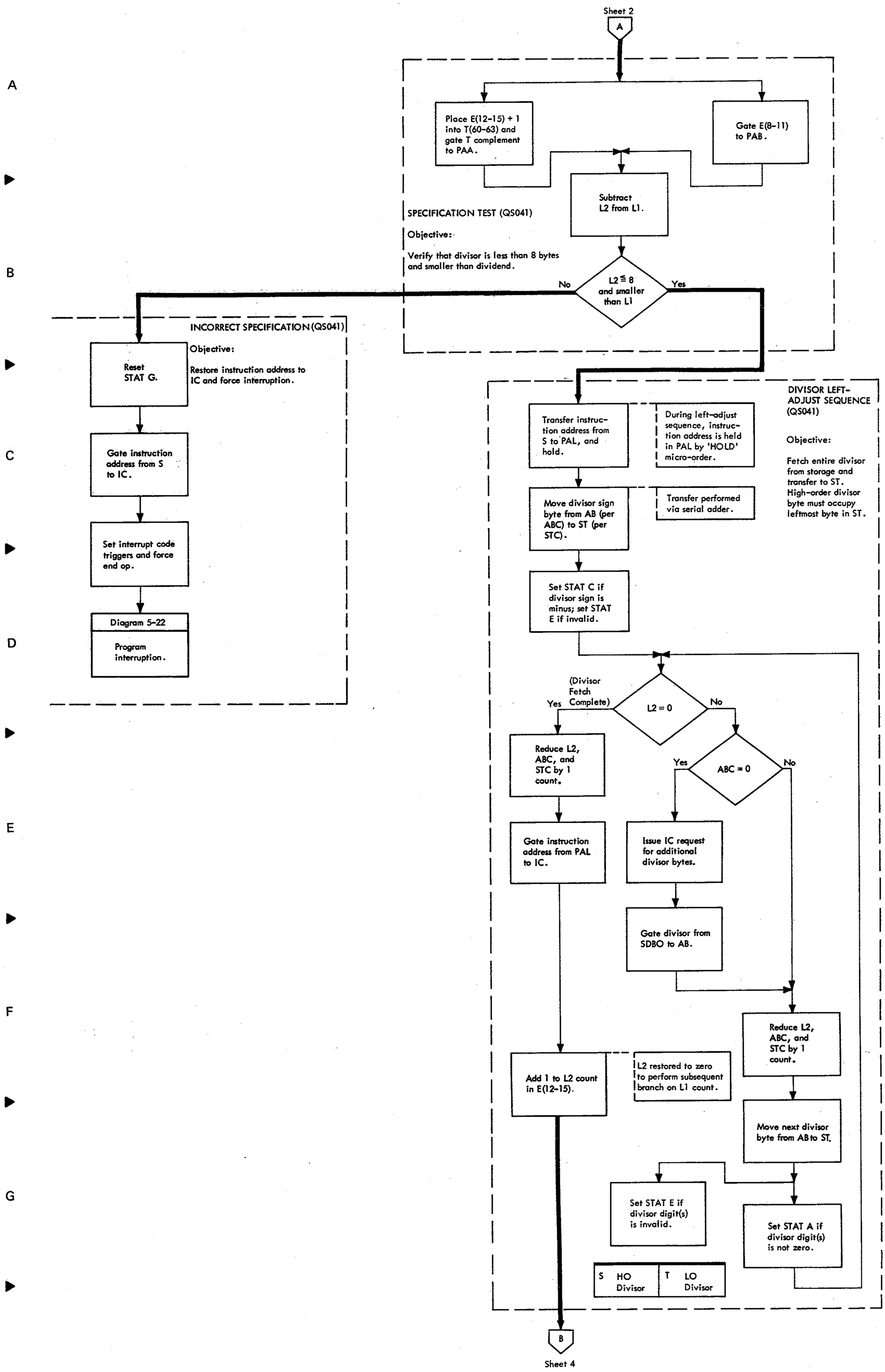


Diagram 5-306. Decimal Divide (Sheet 3 of 9)

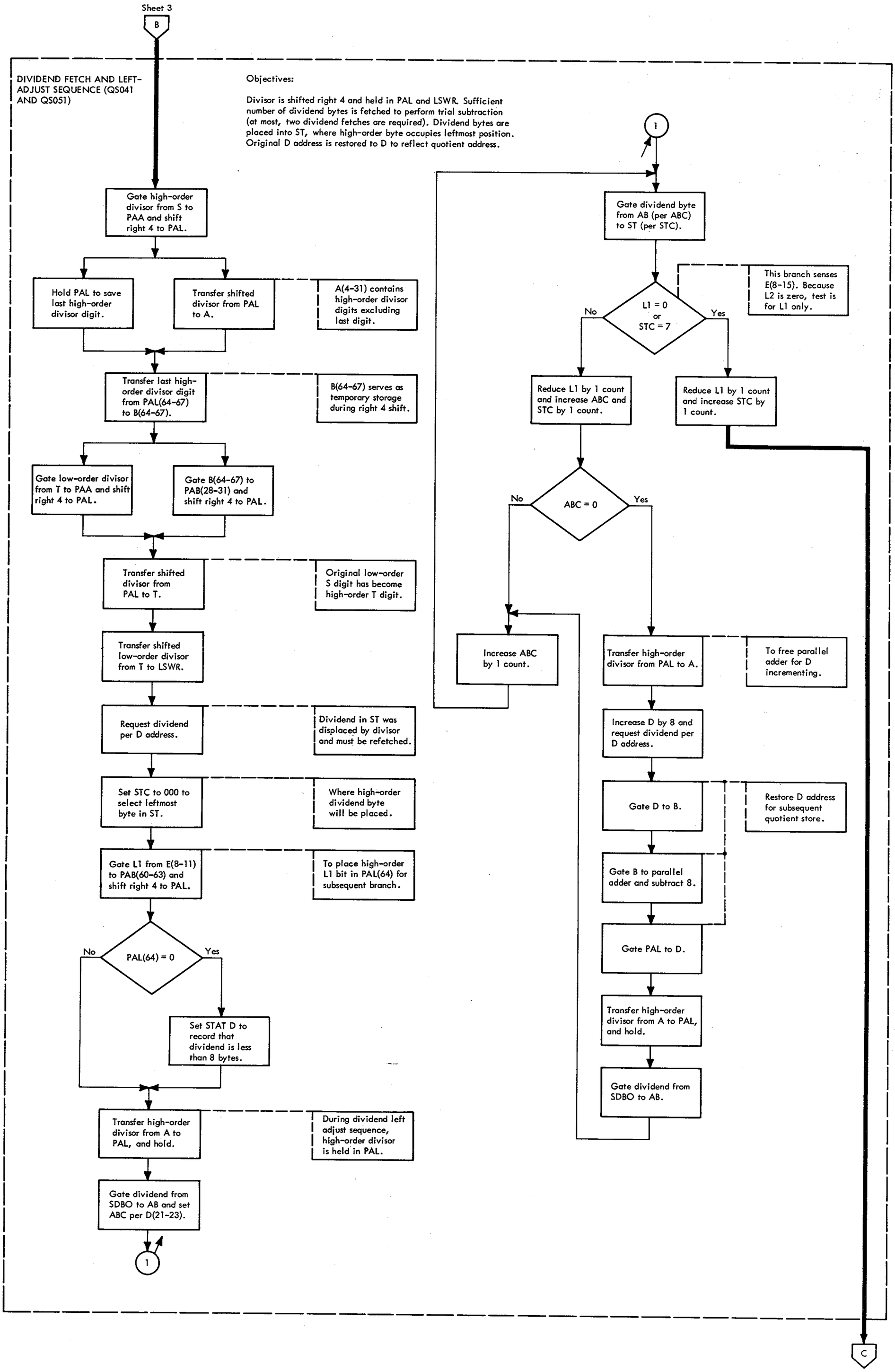


Diagram 5-306. Decimal Divide (Sheet 4 of 9)

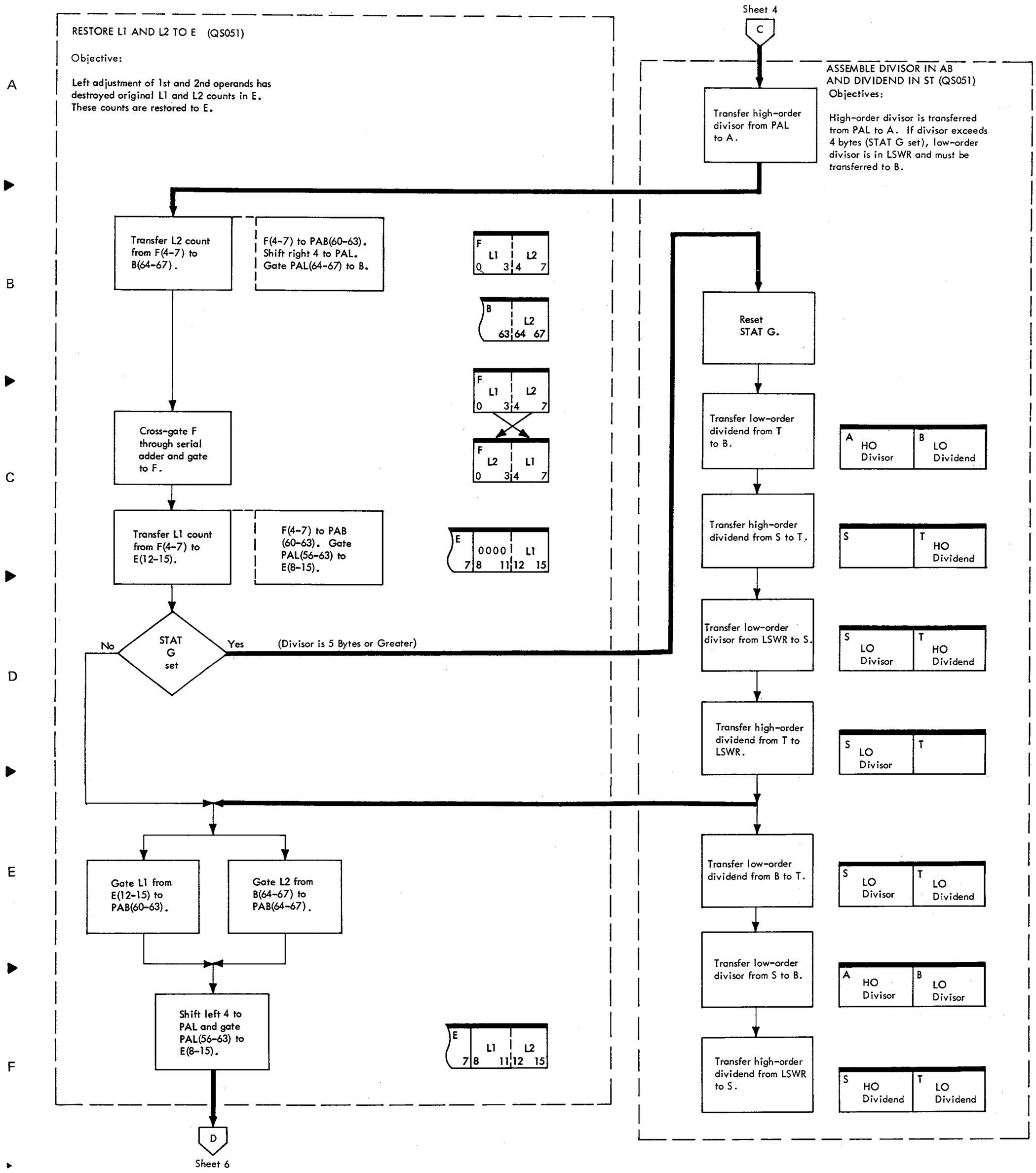


Diagram 5-306. Decimal Divide (Sheet 5 of 9)

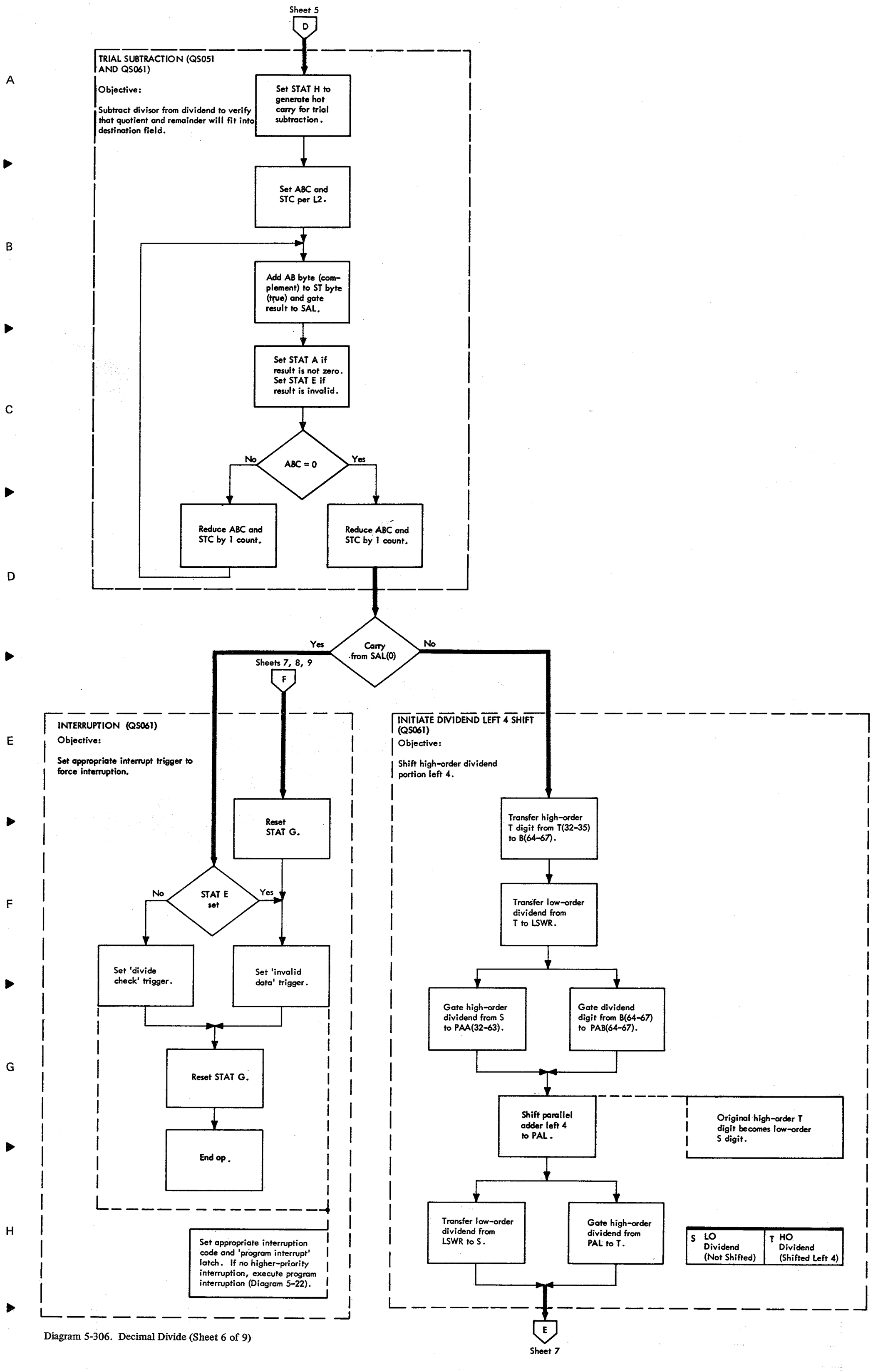


Diagram 5-306. Decimal Divide (Sheet 6 of 9)

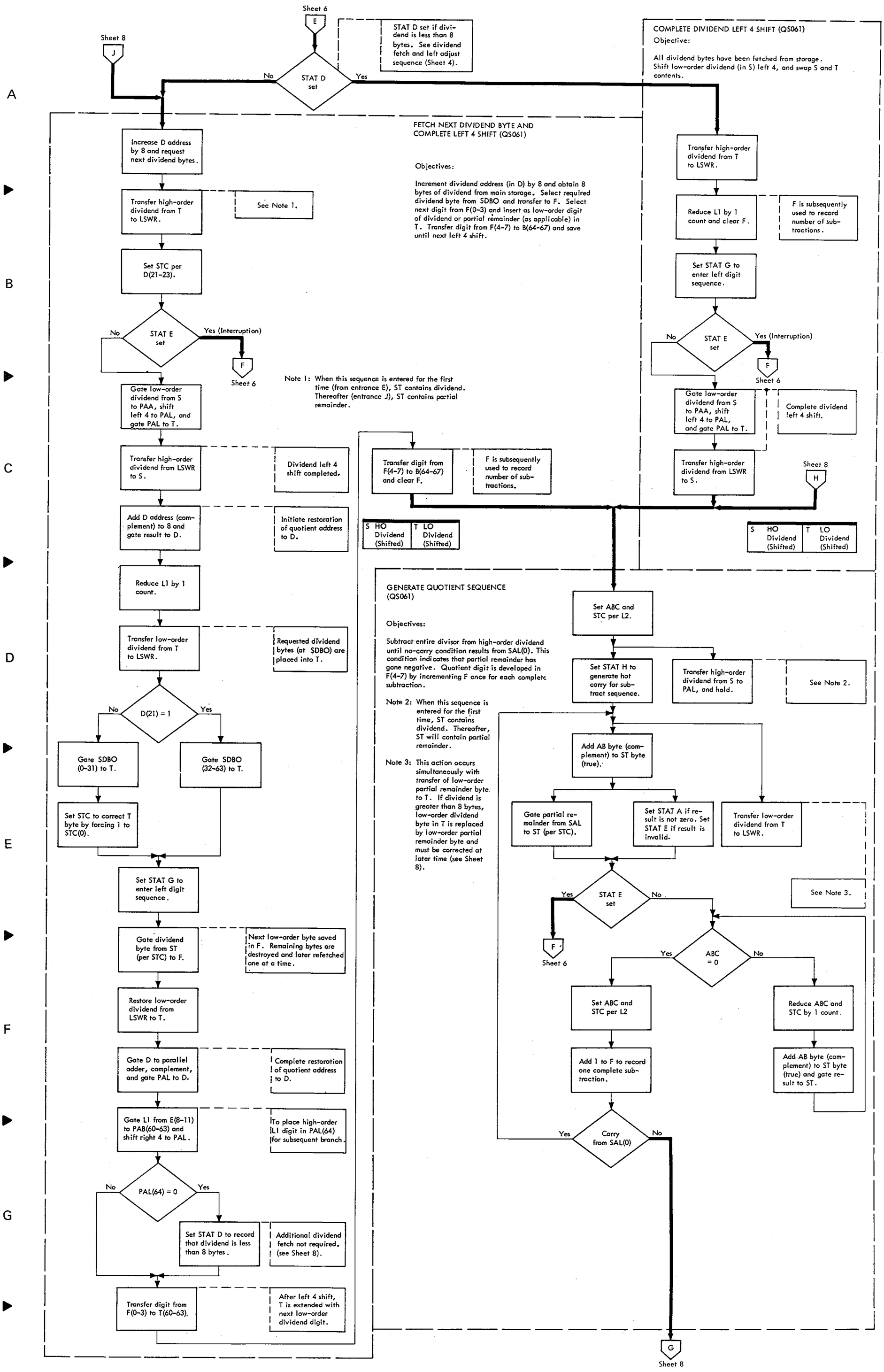


Diagram 5-306. Decimal Divide (Sheet 7 of 9)

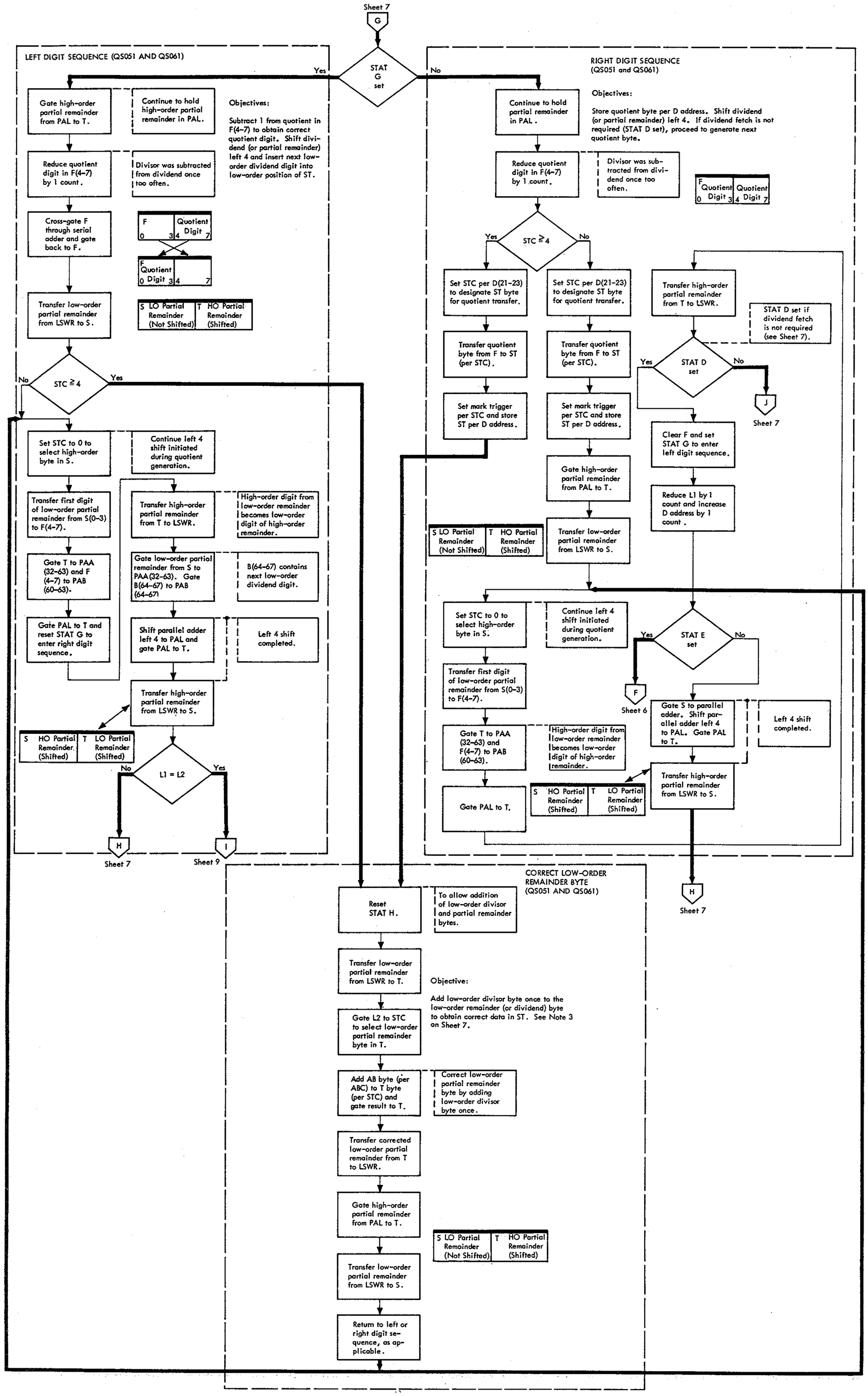


Diagram 5-306. Decimal Divide (Sheet 8 of 9)

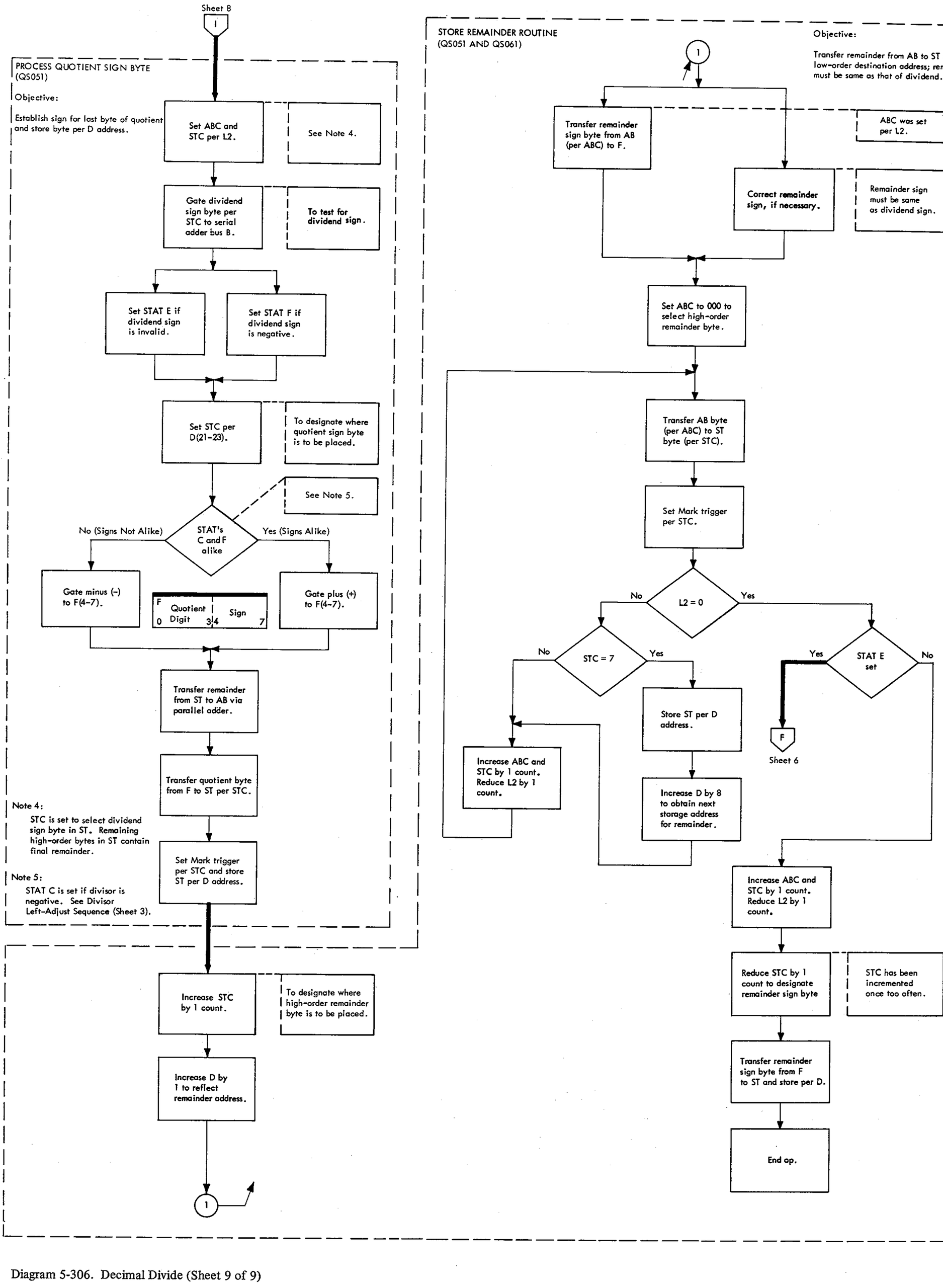


Diagram 5-306. Decimal Divide (Sheet 9 of 9)