IBM Customer Engineering

Instruction-Maintenance

7094 II Data Processing System—Volume 3

IBM®

Customer Engineering

Instruction-Maintenance

7094 II Data Processing System—Volume 3

Preface

Volume 3 of the 7094 II Instruction-Maintenance Manuals contains Non-Arithmetic Instructions, Overlap, Trapping, Channel Instructions, IBM 7151 Model 2 Console, and Compatibility.

The material in this manual is written at engineering change level 253401; however, future engineering changes may change the logic and machine operations from the presentation in this manual. This volume obsoletes Form Z22-2723.

Two other CEIM manuals for the 7094 II are available:

7094 II Data Processing System—Volume 1, Introduction, Component Circuits, System and Functional Components, Timing, Form 223-2721.

7094 II Data Processing System—Volume 2, Arithmetic Instructions, Form 223-2722.

Other manuals for the 7094 II are: *7302 Model 3 Core Storage, CEIM,* Form 223-2724; *7607 Data Channel, CEIR,* Form 223-6910; and *7909 Data Channel, CEIR,* Form 223-2551.

Copies of this and other IBM publications can be obtained through IBM Branch Offices. Address comments concerning the contents of this publication to:
IBM Corporation, CE Manuals, Dept. B96, P.O. Box 390, Poughkeepsie, N.Y. 12602

Contents

Illustrations

FIGURE TITLE PAGE

Word Transmission Instructions

Transfer Instructions

Index Instructions

Shifting Instructions

Sense Indicator Instructions

AND and OR Instructions

Skip Instructions

Convert Instructions

Control Instructions

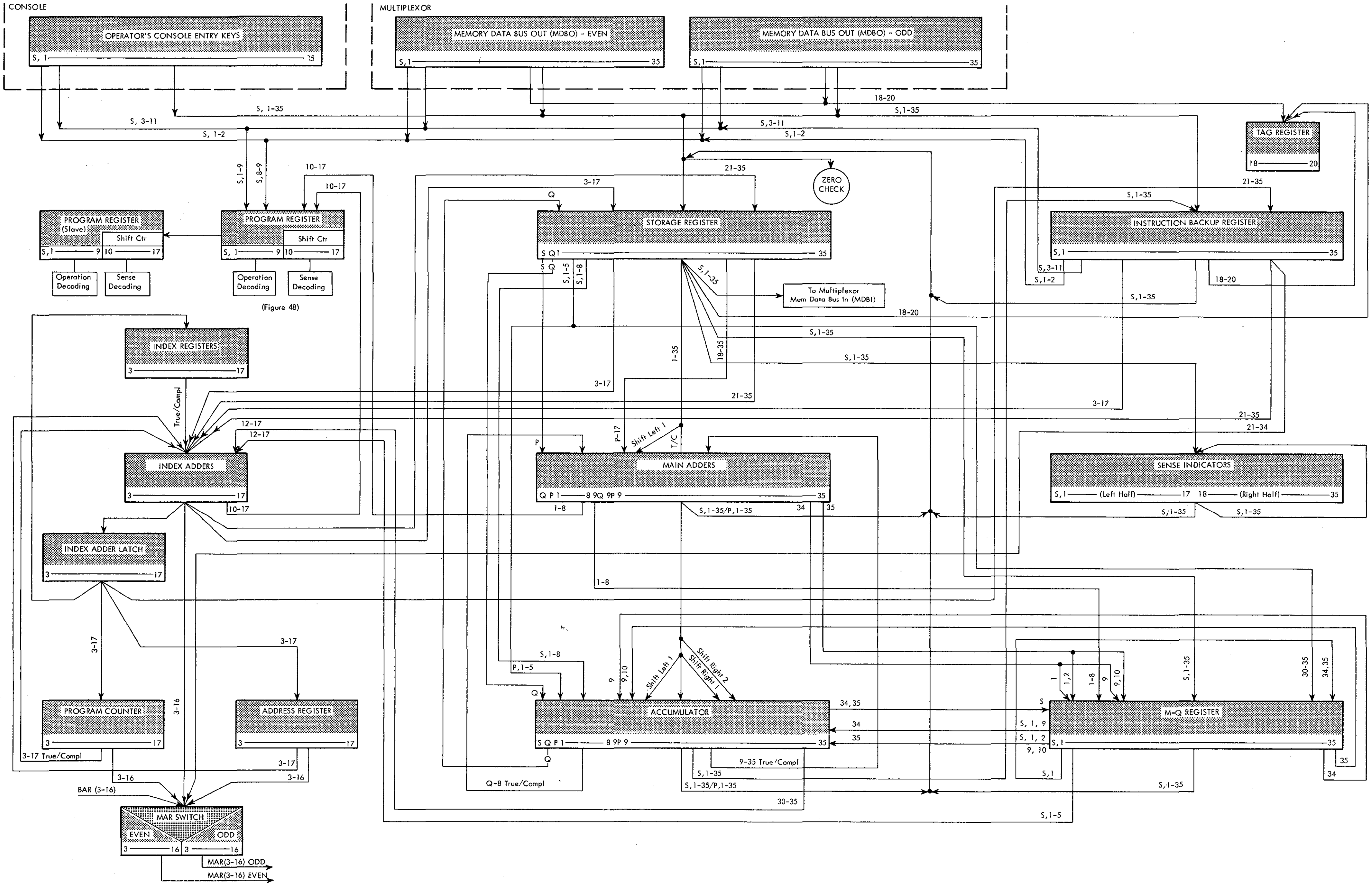
Overlap

Trapping

Channel Instructions

7151 Model 2 Console Control Unit

Compatibility



7094 II CPU Data Flow

Word transmission instructions move words or parts of words from registers in the computer to storage and to other registers. Entry to multiplexor storage bus OR'ing from the central processing unit is through the storage register. Information from the channels enters multiplexor storage bus OR'ing from the channel storage bus switches.

When storing a complete word, MF store prefix, MF store decrement, MF store tag, and MF store address are activated (Systems 02.09.03.1). These lines, acting simultaneously, inhibit strobing of the entire word, and gate their portion of the word from the storage bus to the memory data register. If an instruction calls for storing part of a word, the MF store line for that portion only is activated. The remainder of the word in the specified storage location is strobed and not altered.

Store Zero STZ + 0600 (I, E)

This instruction stores zeros in all positions of storage location X. The operation is similar to store, except that the storage register is not gated to the storage bus, thereby storing zeros. See Figure 1.

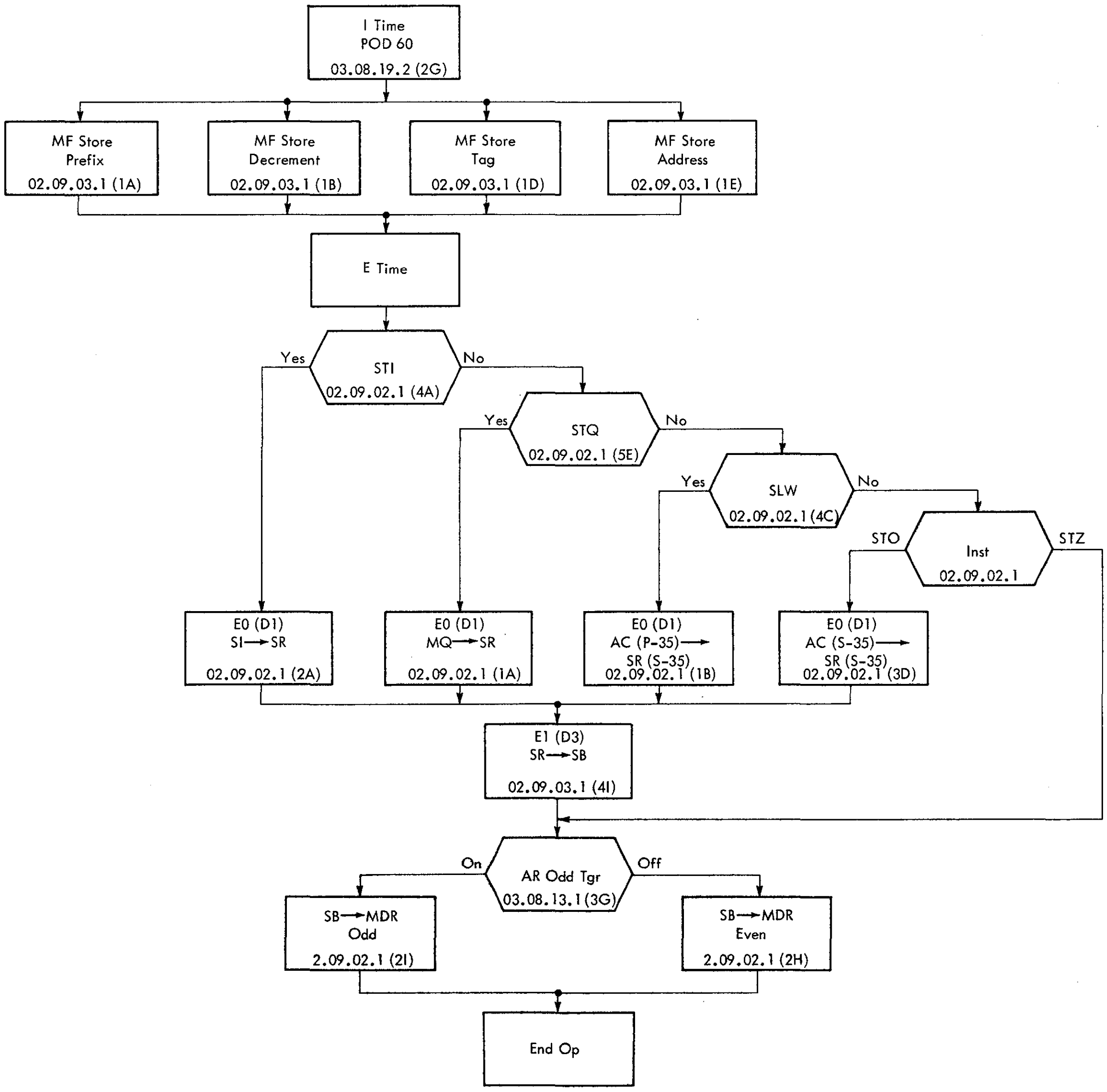

I Time
POD 60
03.08.19.2 (2G)
MF Store Prefix 02.09.03.1 (1A)
MF Store Decrement 02.09.03.1 (1B)
MF Store Tag 02.09.03.1 (1D)
MF Store Address 02.09.03.1 (1E)
E Time
Yes
STI 02.09.02.1 (4A)
No
Yes
STQ 02.09.02.1 (5E)
No
Yes
SLW 02.09.02.1 (4C)
No
STO
Inst 02.09.02.1
STZ
E0 (D1) SI→SR 02.09.02.1 (2A)
E0 (D1) MQ→SR 02.09.02.1 (1A)
E0 (D1) AC (P-35)→ SR (S-35) 02.09.02.1 (1B)
E0 (D1) AC (S-35)→ SR (S-35) 02.09.02.1 (3D)
E1 (D3) SR→SB 02.09.03.1 (4I)
On
AR Odd Tgr 03.08.13.1 (3G)
Off
SB→MDR Odd 2.09.02.1 (2I)
SB→MDR Even 2.09.02.1 (2H)
End Op

Figure 1. Store Zero (STZ); Store MQ(STQ); Store (STO); Store Logical Word (SLW); Store Indicators (STI)

Store MQ STQ – 0600 (I, E)

The contents of storage location X are replaced by the contents of the MQ register (S, 1-35). The contents of the MQ are unchanged. See Figure 1.

Store STO + 0601 (I, E)

This instruction replaces the contents of storage location X with the contents of the AC (S, 1-35). Store prefix, decrement, tag, and address control lines are activated so that the entire word may be stored. The AC is unchanged. See Figure 1.

Store Logical Word SLW + 0602 (I, E)

This instruction replaces the contents of storage location X with the contents of the AC (P, 1-35). The AC is unchanged. Operation is similar to store except for the gating of AC(P) to SR(S). See Figure 1.

Store Left Half MQ SLQ – 0620 (I, E)

The contents of the MQ (S, 1-17) replace the contents of storage location X (S, 1-17). Storage location X contents (18-35) and the contents of the MQ are unchanged. MF store prefix and MF store decrement are activated (Systems 02.09.03.1). See Figure 2.

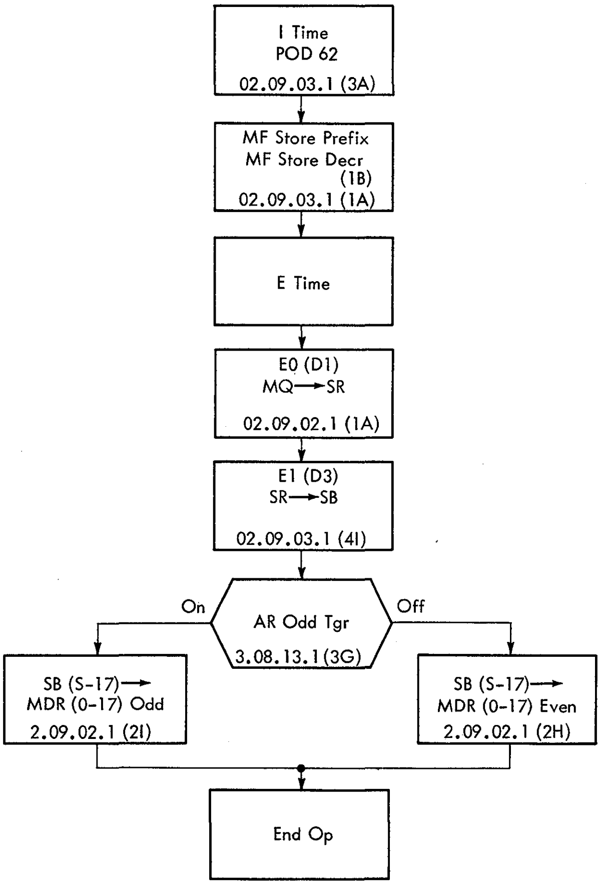


Figure 2. Store Left Half MQ (SLQ)

Store Address STA+0621 (I, E)

The contents of storage location X (21-35) are replaced by the contents of the AC (21-35). Storage location X (S, 1-20) and the AC remain unchanged. MF store address is activated (Systems 02.09.03.1). See Figure 3.

Store Decrement STD+0622 (I, E)

This instruction places the contents of the AC (3-17) into storage location X (3-17). Storage location X (S, 1-2, 18-35) and the AC are unchanged. MF store decrement is activated (Systems 02.09.03.1). See Figure 3.

Store Tag STT+0625 (I, E)

The contents of storage location X (18-20) are replaced by the contents of the AC (18-20). The contents of storage location X (S, 1-17, 21-35) and the AC are unchanged. MF store tag is activated (Systems 02.09.03.1). See Figure 3.

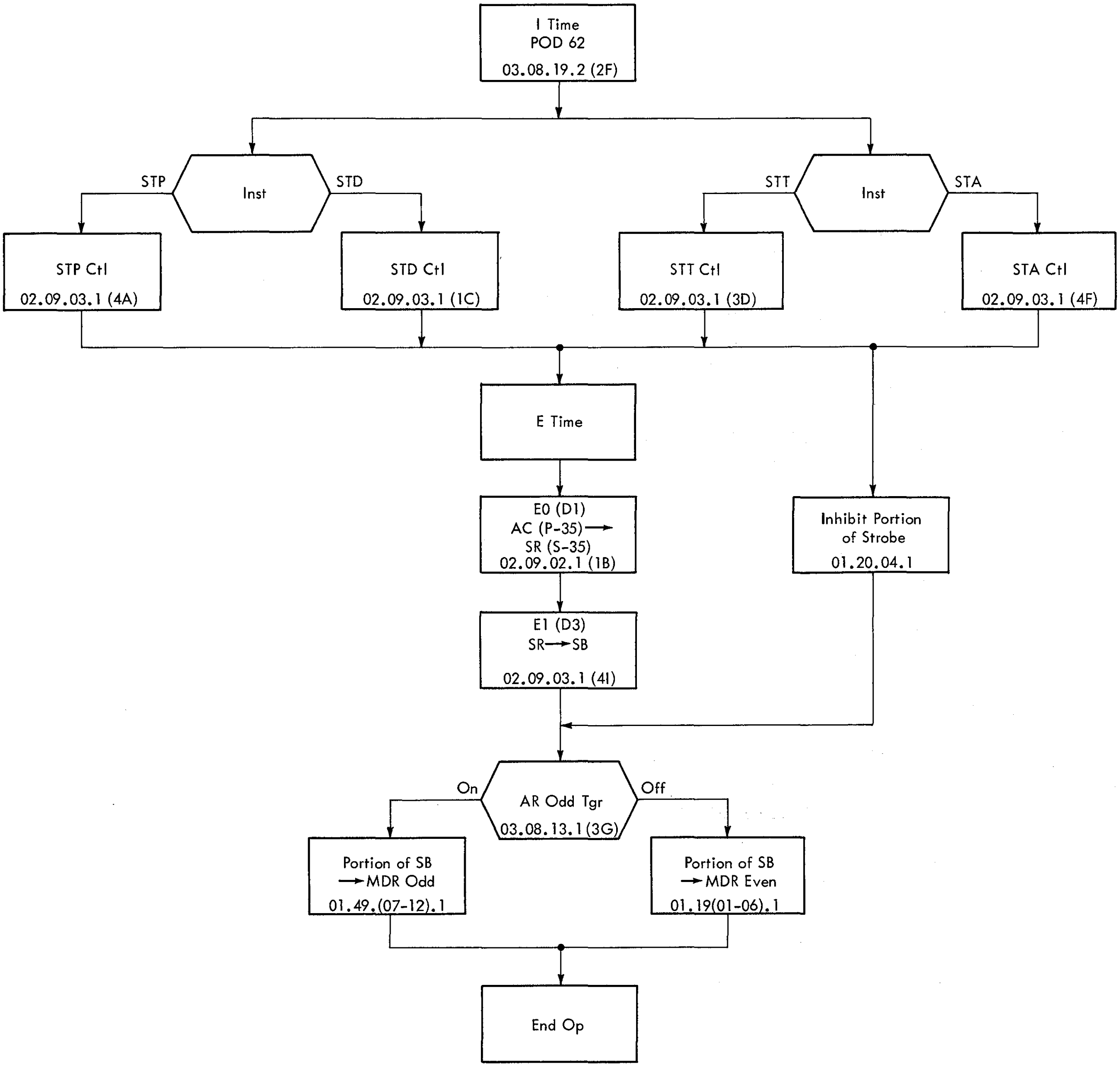


Figure 3. Store Prefix (STP); Store Decrement (STD); Store Tag (STT); Store Address (STA)

Store Instruction Location Counter STL – 0625 (I, E)

This instruction stores the PC in storage location X (21-35). The contents of storage location X (S, 1-20) and the PC are unchanged. MF store address is activated (Systems 02.09.03.1). The PC is gated to the SR via the XAD (Systems 02.09.03.1). See Figure 4.

Store Prefix STP + 0630 (I, E)

The contents of storage location X (S, 1-2) are replaced by the contents of the AC (P, 1-2). Storage location X contents (3-35) and the contents of the AC remain unchanged. See Figure 3.

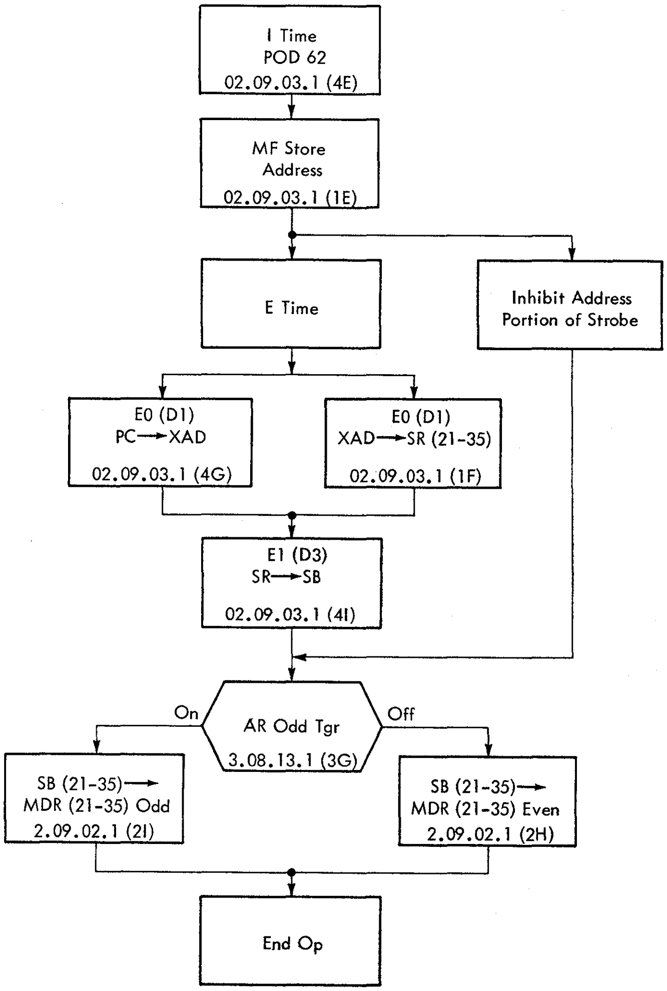


Figure 4. Store Instruction Location Counter (STL)

Load MQ LDQ + 0560 (I, E)

The contents of the MQ are replaced by storage location X. The MDBO is gated to the SR, and the SR is gated to the MQ. See Figure 5.

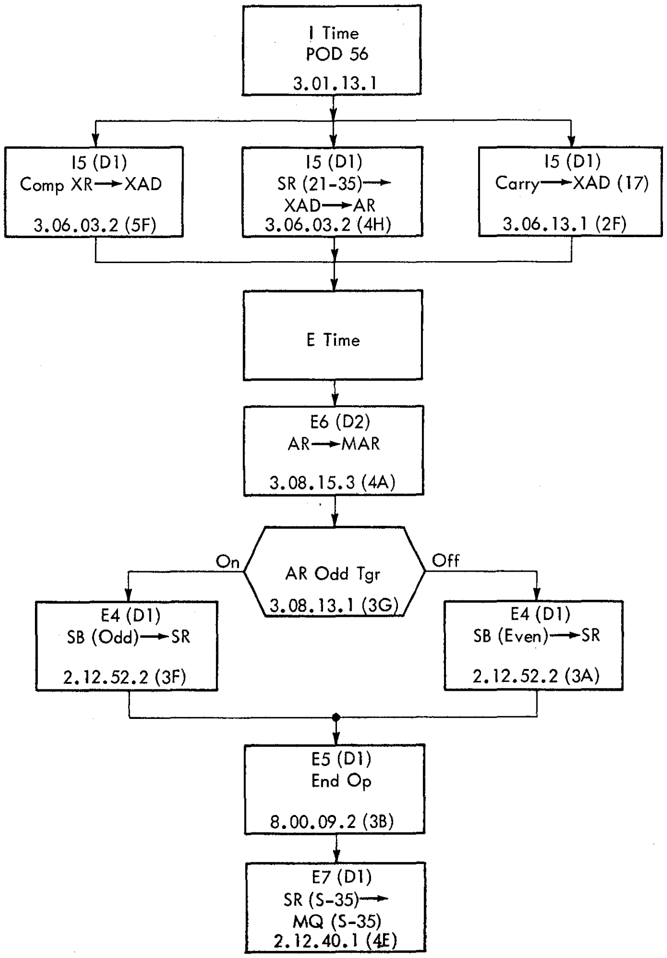


Figure 5. Load MQ (LDQ)

Exchange Logical AC and MQ XCL−0130 (I)

The AC (P, 1-35) and the MQ (S, 1-35) are interchanged. AC S and Q are reset. The exchange is made through the SR: the MQ is gated first to the SR, and then to the AD and AC. At the same time the AD are gated to the AC, the AC is sent to the SR, and then to the MQ. See Figure 6.

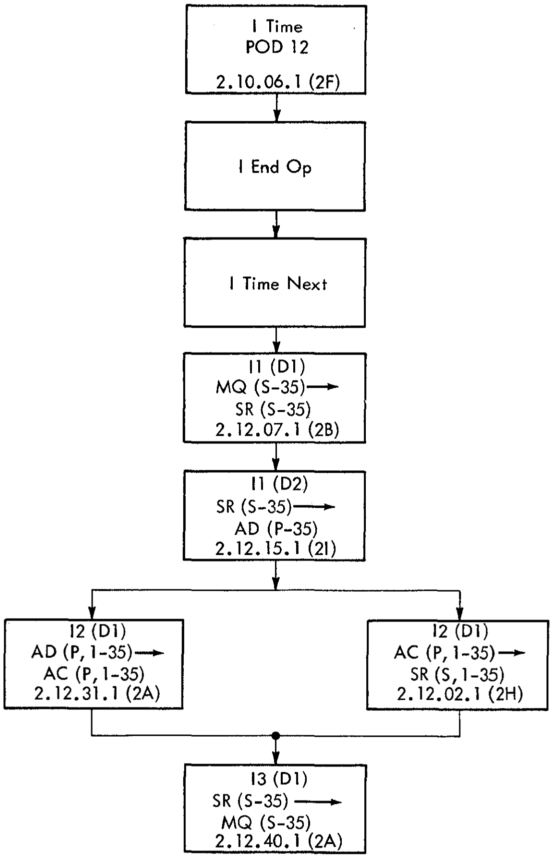


Figure 6. Exchange Logical AC and MQ (XCL)

Exchange AC and MQ XCA+0131 (I)

This instruction exchanges the AC (S, 1-35) with the MQ (S, 1-35). Gating is done through the SR in the same manner as in XCL. See Figure 7.

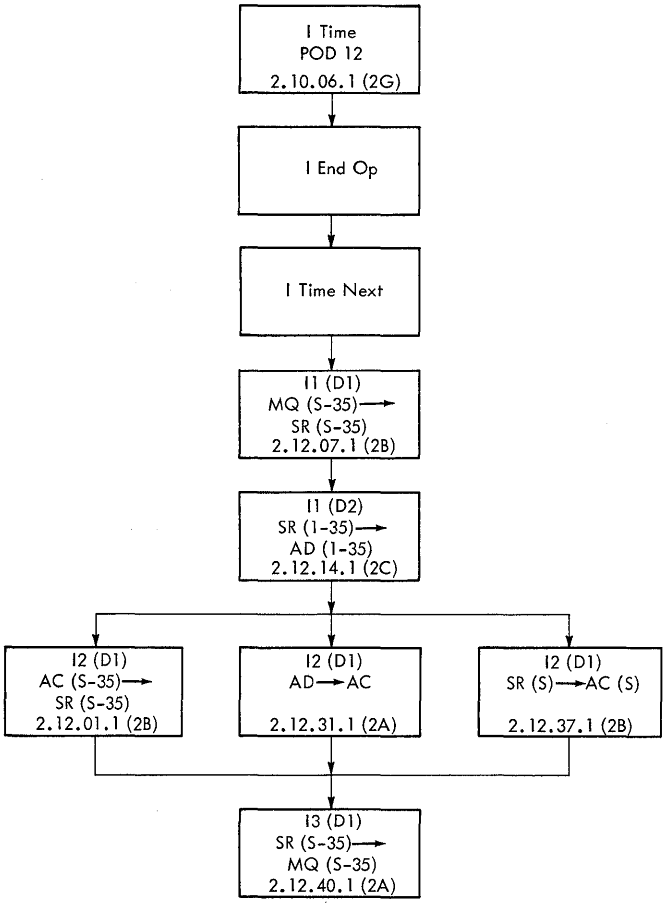


Figure 7. Exchange AC and MQ (XCA)

Double Store DST — 0603 (I, E, E)

This instruction stores the contents of the accumulator (S-35) in location X, and the contents of the MQ (S-35) in location X + 1. See Figure 8.

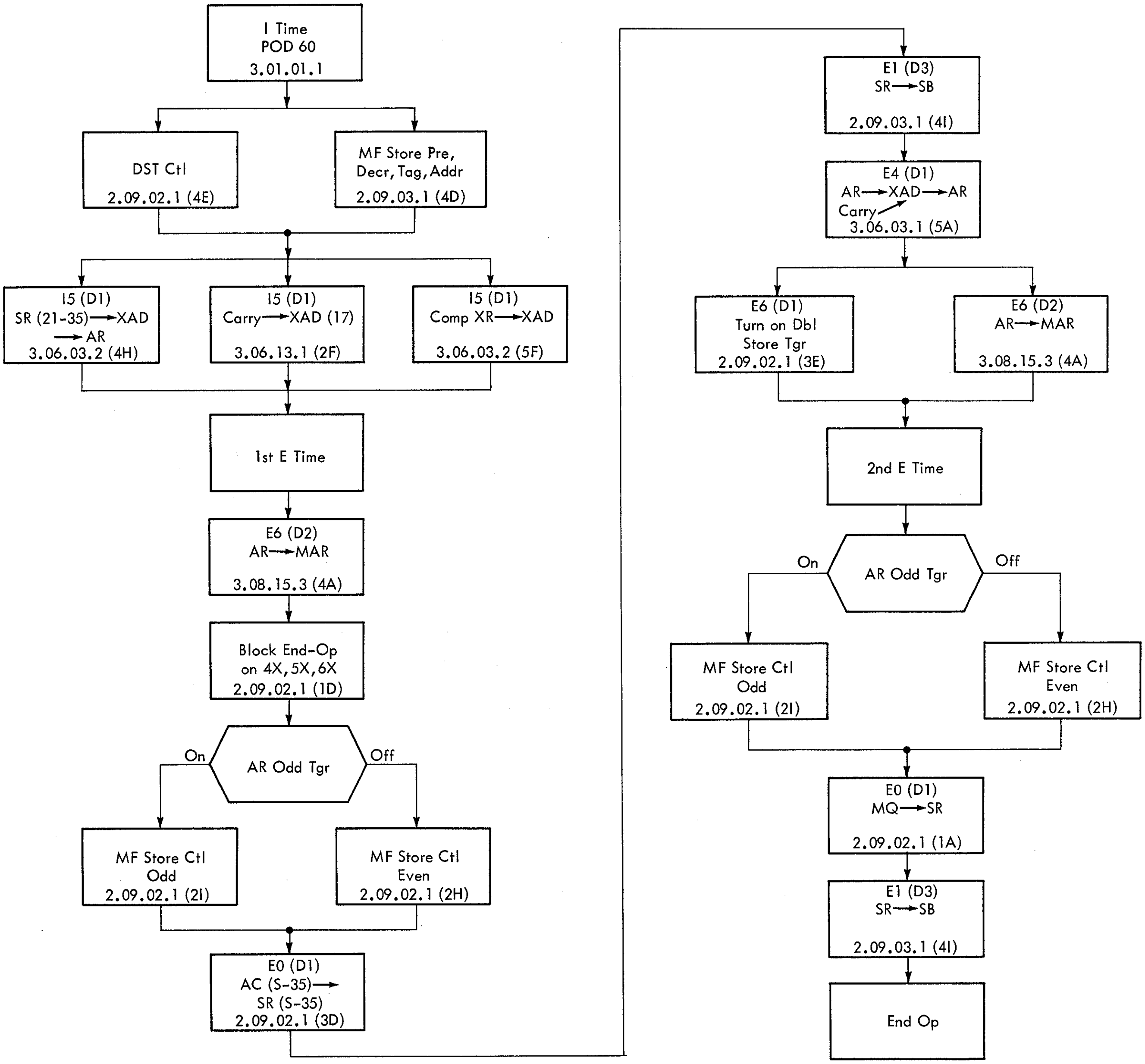


Figure 8. Double Store (DST)

Double Load   DLD + 0443 (I, E)

This instruction places the contents of location X into the accumulator and X + 1 into the MQ. See Figure 9.

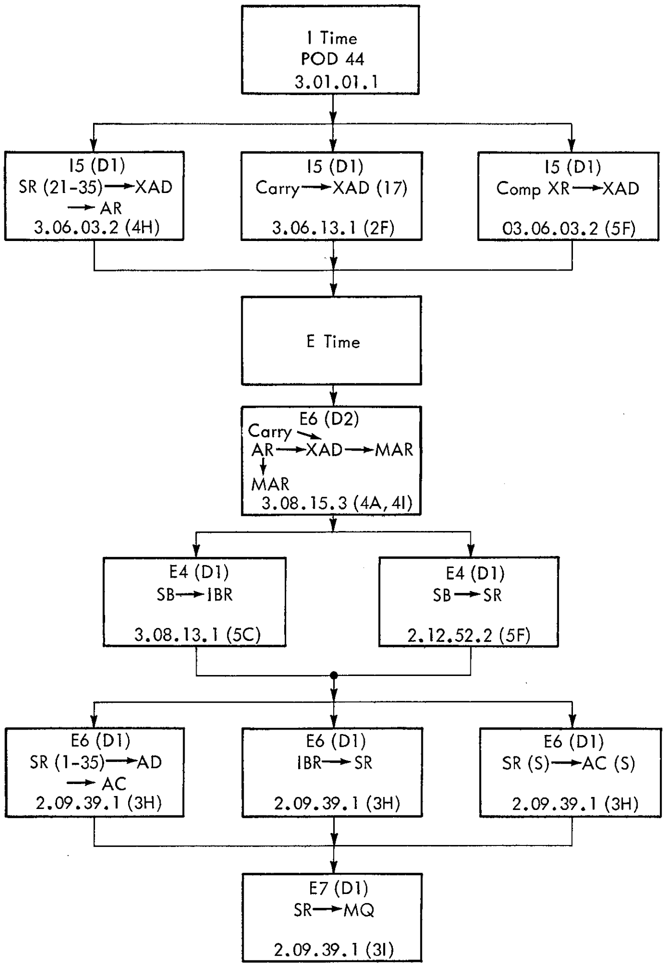


Figure 9. Double Load (DLD)

Transfer Instructions

Transfer instructions can alter a program sequence either conditionally or unconditionally. This ability reduces the number of steps necessary to complete a program.

Transfer TRA + 0200 (I)

The computer takes its next instruction from location X. The address portion of the instruction is gated to the address register via the index adders. This address is then set into the program counter during I time of the next instruction. See Figure 10.

Transfer on MQ Plus TQP + 0162 (I)

The computer transfers to location X if the MQ sign is plus. If the MQ sign is minus, the computer proceeds to the next sequential instruction. See Figure 11.

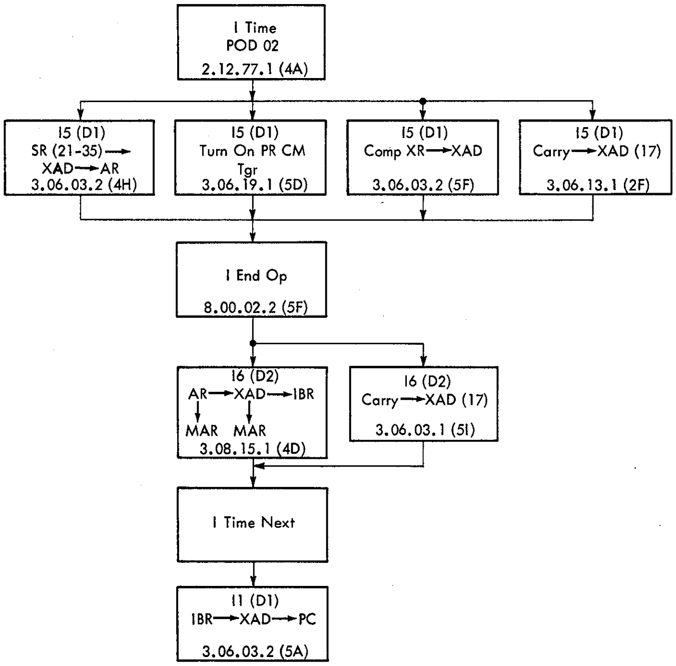


Figure 10. Transfer (TRA)

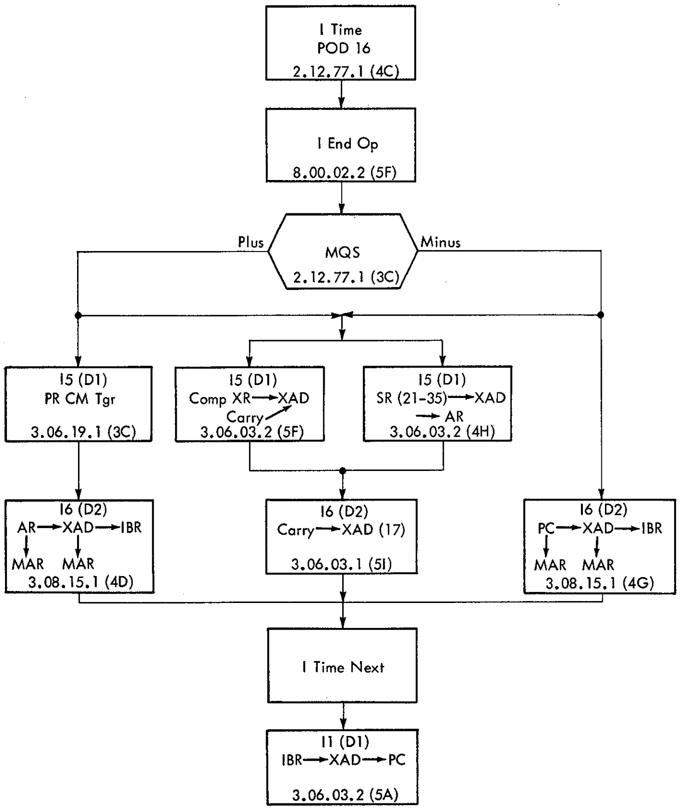


Figure 11. Transfer on MQ Plus (TQP)

Transfer on Plus TPL + 0120 (I)

The computer transfers to location X if the accumulator sign is plus. If the AC sign is minus, the computer proceeds to the next sequential instruction. This operation is similar to Transfer. See Figure 12.

Transfer on Minus TMI − 0120 (I)

This instruction is performed the same as TPL except that the transfer occurs if the accumulator sign is minus. See Figure 12.

Transfer on Overflow TOV + 0140 (I)

If the AC overflow trigger is on, a transfer is taken to storage location X and the overflow trigger is turned off. If the overflow trigger is off, the computer proceeds to the next instruction in sequence. See Figure 13.

Transfer on No Overflow TNO − 0140 (I)

If the AC overflow trigger is off, the next instruction is taken from storage location X. If the overflow trigger is on, no transfer is taken and the overflow trigger is turned off. Execution is the same as in TOV. See Figure 13.

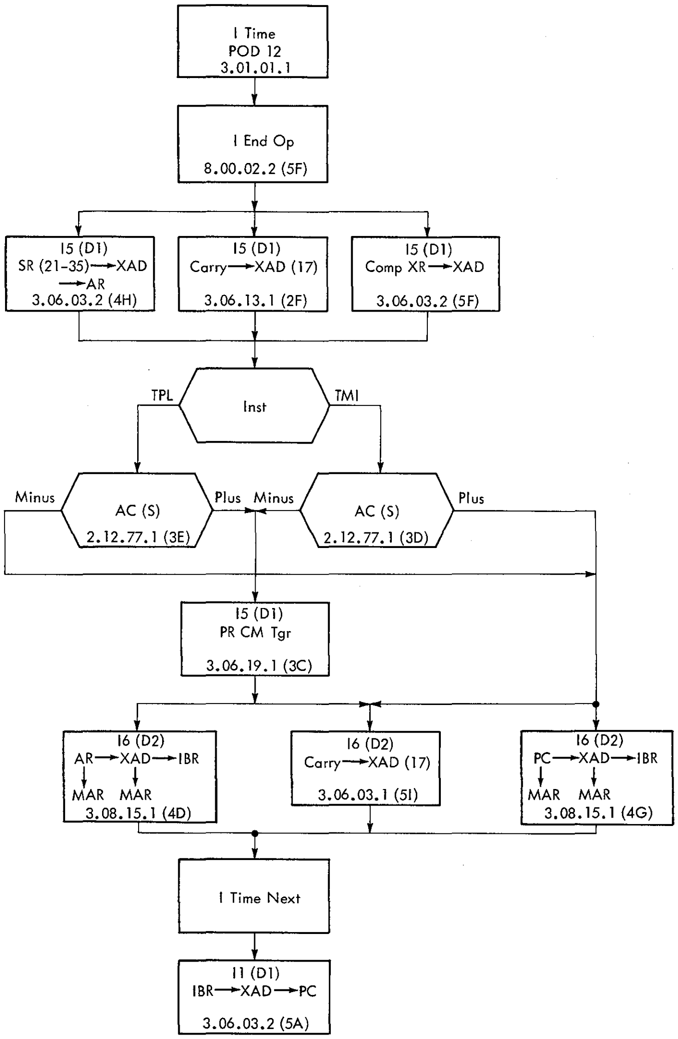


Figure 12. Transfer on Plus (TPL); Transfer on Minus (TMI)

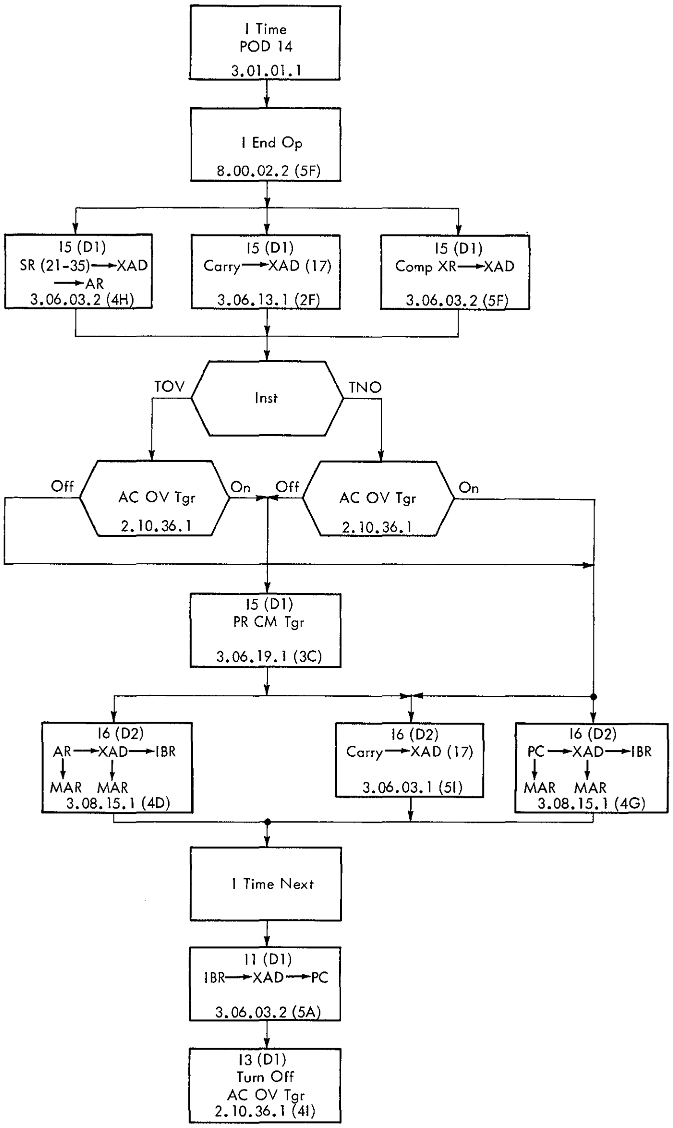


Figure 13. Transfer on Overflow (TOV); Transfer on No Overflow (TNO)

Transfer on Zero TZE + 0100 (I)

If the contents of the AC (Q, P, 1-35) are zero, the computer transfers to location X. If the contents are not zero, the next instruction in sequence is taken. In either case the contents of the AC are not changed. To test for a zero condition the AC(1-35) is gated to the SR, but not set in. The SR inputs are then checked along with AC (P and Q). See Figure 14.

Transfer on No Zero TNZ − 0100 (I)

If the contents of the AC (Q, P, 1-35) are not zero, a transfer is taken to location X. If the contents are zero, the computer proceeds to the next sequential instruction. AC contents are not altered. This operation is similar to TZE. See Figure 14.

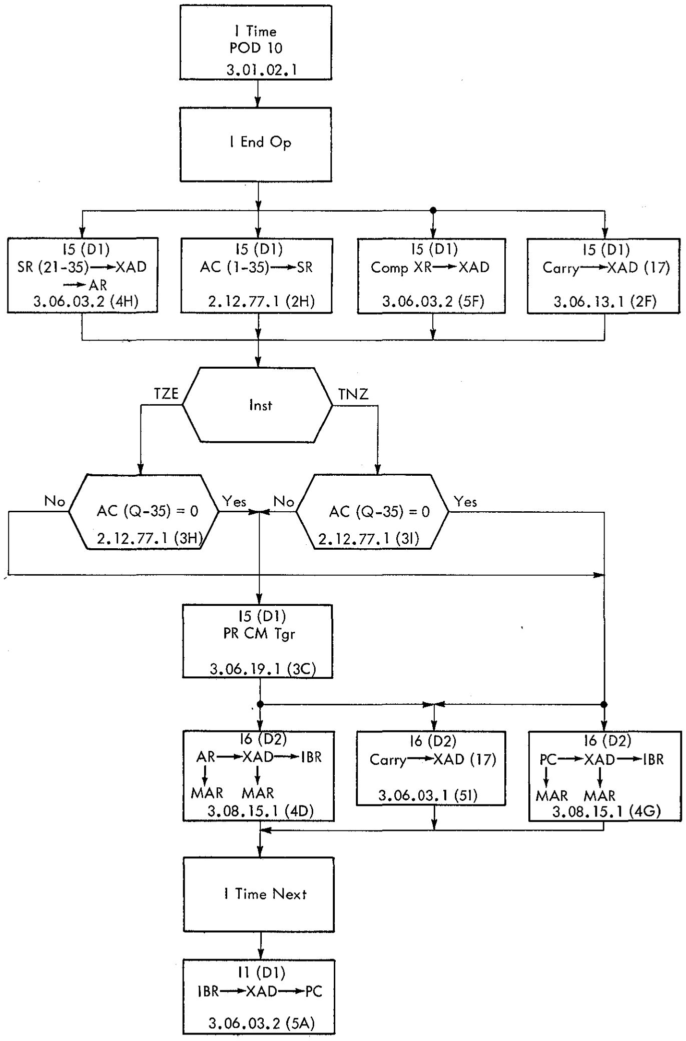


Figure 14. Transfer on Zero (TZE); Transfer on No Zero (TNZ)

Transfer on MQ Overflow TQO + 0161 (I)

If the MQ overflow trigger is on, and the FPT mode trigger is off, the computer transfers to location X. If the MQ overflow trigger is off, the next sequential instruction is taken. This is a 704 compatibility instruction. If the FPT mode trigger is on, the instruction acts as a NOP. See Figure 15.

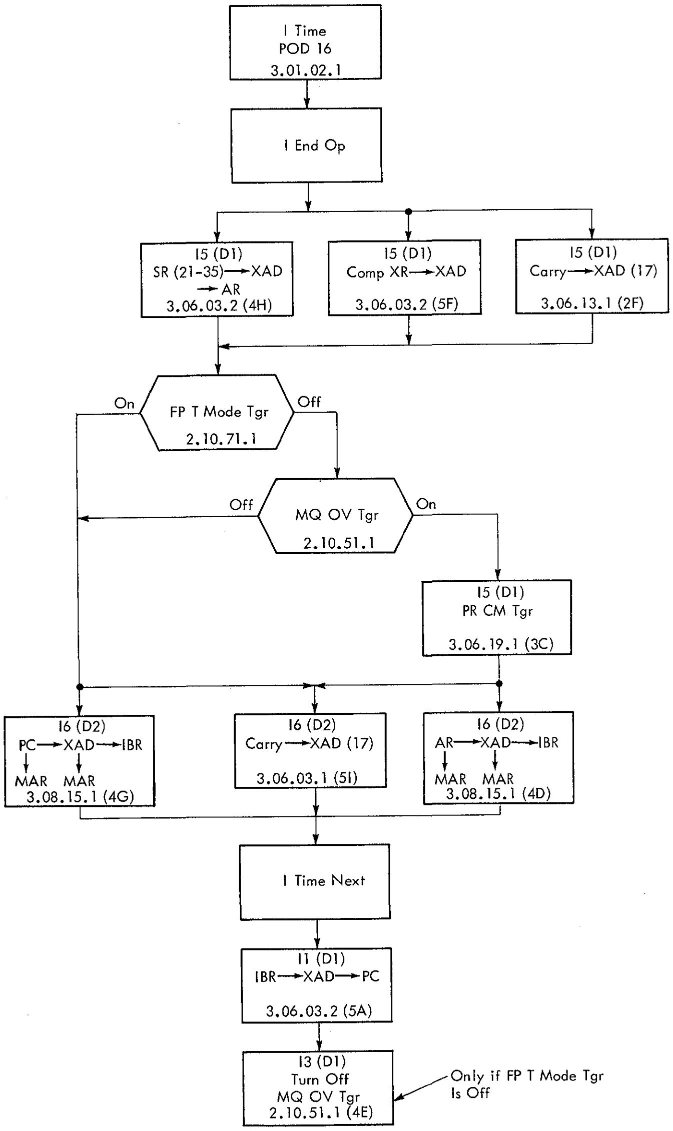


Figure 15. Transfer on MQ Overflow (TQO)

Transfer on Low MQ TLQ + 0040 (I, L)

If the contents of the MQ are algebraically less than the contents of the AC, the computer transfers to location X. If the MQ is equal to or greater than the AC, no transfer is taken. Signs are considered, and the contents of both registers are left unchanged. See Figure 16.

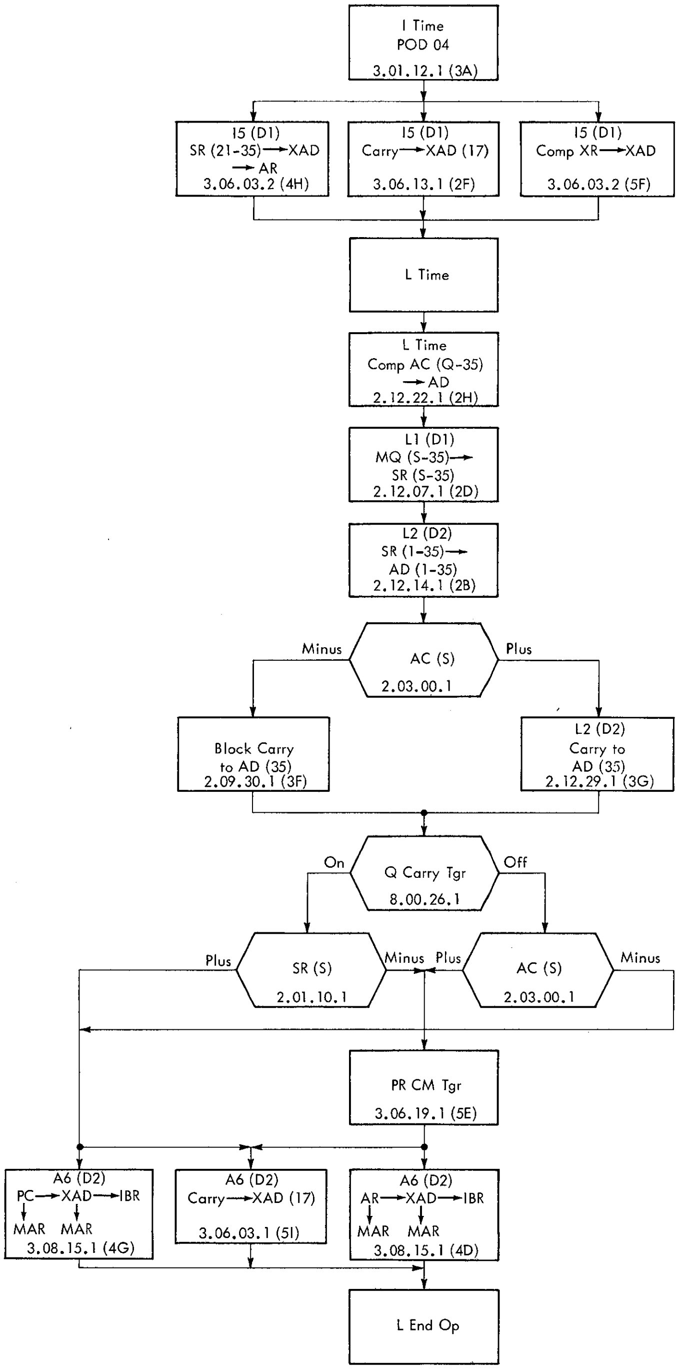


Figure 16. Transfer on Low MQ (TLQ)

Index Instructions

The 7094 II contains seven 15-position index registers. The computer can operate in either one of two indexing modes:

1. Multiple tag mode (three index register mode)
2. Seven index register mode

Multiple tag mode, set by the EMTM instruction, which is also the normal mode, causes the computer to use only three index registers and provides compatibility with 7090 programs. Instruction positions 18, 19, and 20 indicate any one or combination of the three index registers. Multiple tagging in these positions causes OR'ing of the index registers.

The seven index register mode is entered by using the instruction Leave Multiple Tag Mode (LMTM). Instruction positions 18, 19, and 20 specify a specific index register. OR'ing is not possible.

A three-position tag register is fed from the storage bus or IBR depending upon overlap conditions. The output decoding of this register depends on the index mode of operation.

For addressing, the index registers are gated to the index adders in complement form with a carry to index adder (17). In effect, this subtracts the value of the specified index register.

Transfer and Set Index TSX + 0074 (I)

The TSX instruction places the two's complement of the program counter (location of TSX instruction) into the specified index register and transfers to the location specified in positions 21-35 of the TSX instruction. At I0 of the next instruction, one is subtracted from the value of the PC so that it will equal the location of the TSX instruction. Then, the PC is placed into the XR. At I3 the complement of the XR is gated to the XAD with a carry into XAD(17); the XAD is gated back to the XR, placing the two's complement of the PC into the XR. The transfer is similar to a TRA instruction without address modification. See Figure 17.

Transfer on Index TIX + 2000 (I)

This instruction compares its decrement with the contents of the specified index register. If the number in the index register is greater than the decrement, the number in the index register is reduced by the amount of the decrement; the computer transfers to location X. If the number in the index register is equal to or less than the decrement, the computer takes the next sequential instruction. See Figure 18.

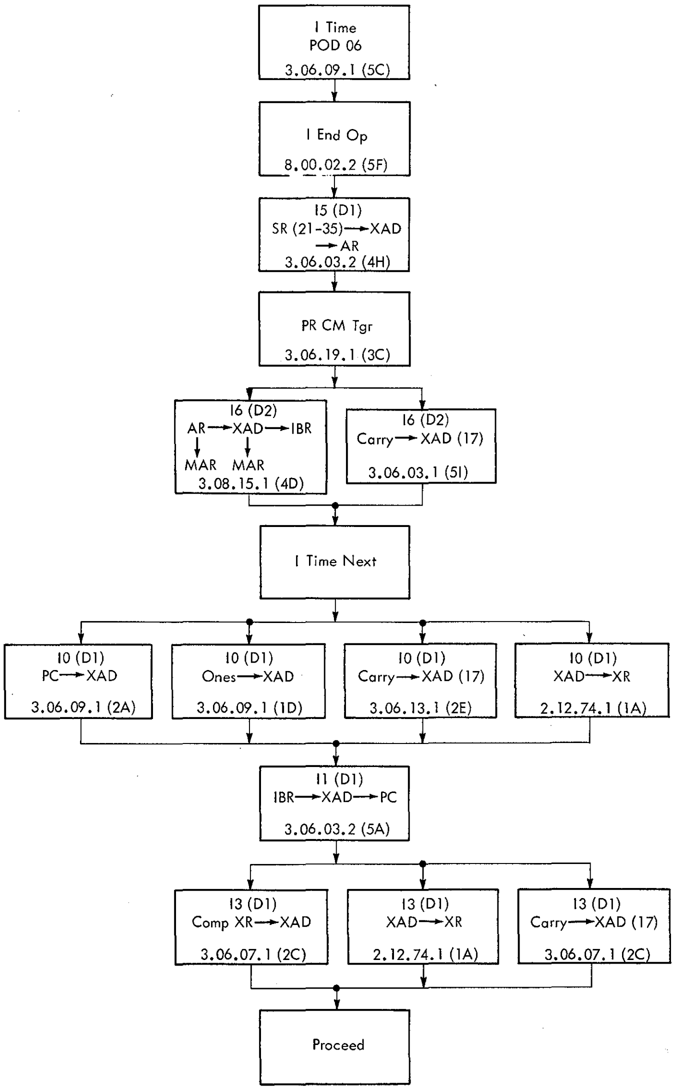
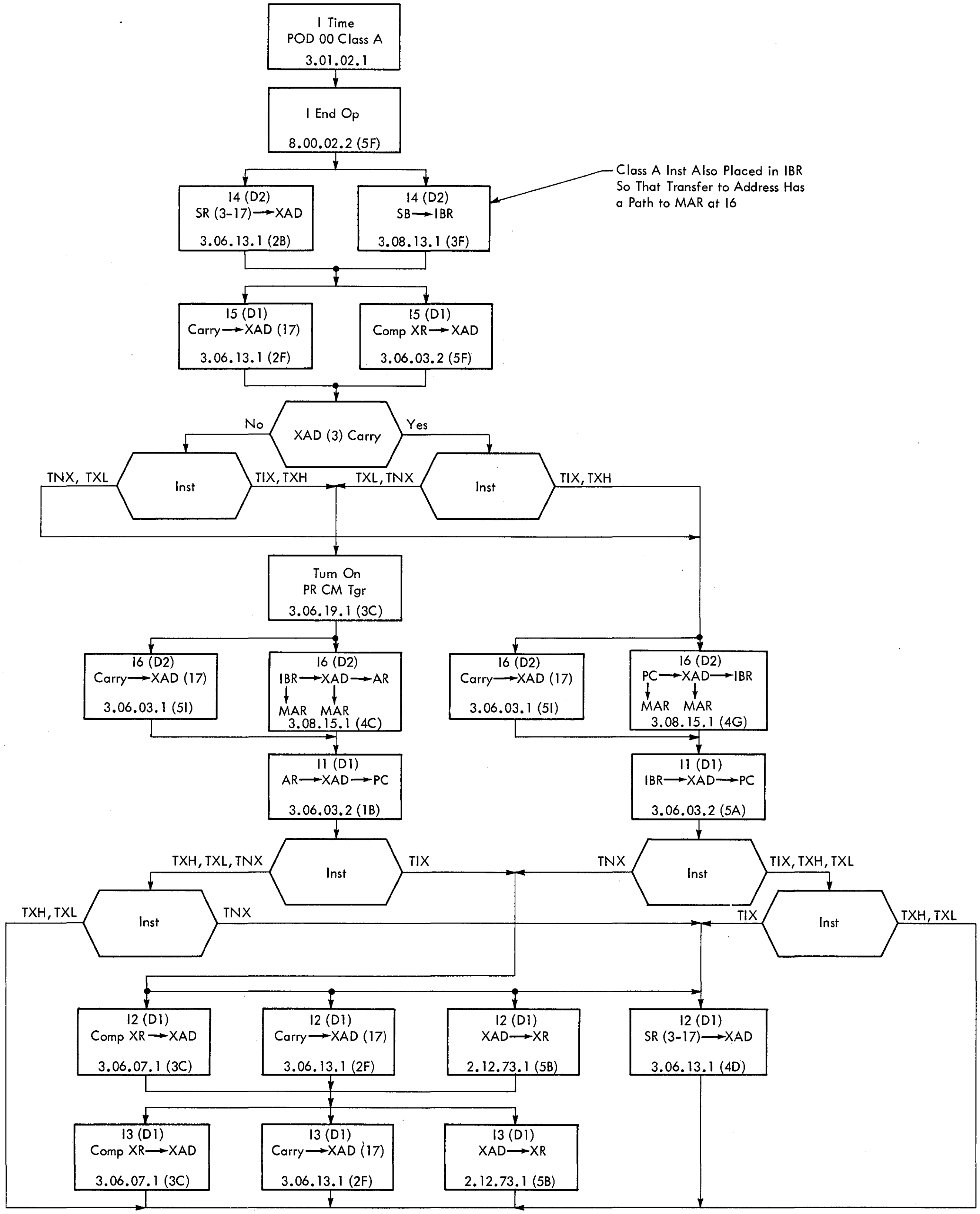


Figure 17. Transfer and Set Index (TSX)



Class A Inst Also Placed in IBR So That Transfer to Address Has a Path to MAR at I6

Figure 18. Transfer on Index (TIX); Transfer on No Index (TNX); Transfer on Index High (TXH); Transfer on Index Low or Equal (TXL)

At I4, the storage bus is gated to the storage register, program register, tag register, and IBR. At I4 and I5 the storage register (3-17) and the complement of the index register are gated to the index adders along with a carry to index adder 17. No carry from index adder 3 indicates a larger number in the index register. For this condition, the IBR (21-35) is gated to MAR at I6 to fetch the transfer-to instruction. At the same time the IBR is sent to the index adder with a carry to set up the transfer-to-address plus one, which is gated to MAR and the address register. At I1 the address register is gated to the PC via the index adders. At I2 and I3 the index register is reduced by the amount in the decrement and placed back in the index register.

If the index register was less than or equal to the decrement, an index adder 3 carry was generated at I5. For this condition, the program counter is gated to MAR to fetch the next sequential instruction. The program counter is also sent to the index adders with a carry, and then to MAR and the IBR. At I1 the IBR is gated to the program counter via the index adders. The number in the index register is not altered.

Transfer on No Index TNX − 2000 (I)

The TNX instruction compares its decrement with the value of the specified index register. If the value of the index register is greater than the decrement, the index register is reduced by the amount of the decrement and the computer takes the next sequential instruction.

If the index register is equal to or less than the decrement, the number in the index register is not changed, and the computer transfers to location X. The operation is similar to TIX, except that the reduction of the index register takes place when conditions for the transfer are not met. See Figure 18.

Transfer on Index High TXH + 3000 (I)

This instruction compares its decrement with the value of the specified index register. If the number in the index register is greater than the decrement, the computer transfers to location X. If the index register is equal to or less than the decrement, the computer takes the next sequential instruction. In either case, the number in the index register is not changed. See Figure 18.

Transfer on Index Low or Equal TXL − 3000 (I)

The TXL instruction compares its decrement with the specified index register. If the index register is equal to or less than the decrement, the computer transfers to location X. If the index register is greater than the decrement, the computer takes the next sequential instruction. In either case, the value of the index register is unchanged. See Figure 18.

Transfer with Index Incremented TXI + 1000 (I)

This instruction adds its decrement to the specified index register and transfers to location X. This transfer always occurs.

The complement of the index register to the index adders at I5 of the TXI cycle has no effect on the operation. The addition takes place at I3 of the following I cycle. See Figure 19.

Load Index From Address LXA + 0534 (I, E)

The contents of the specified index register are replaced by the contents of storage location X (21-35). Storage location X is unchanged. The storage register (21-35) is gated through the index adders to the specified index register at I2 of the following I cycle. See Figure 20.

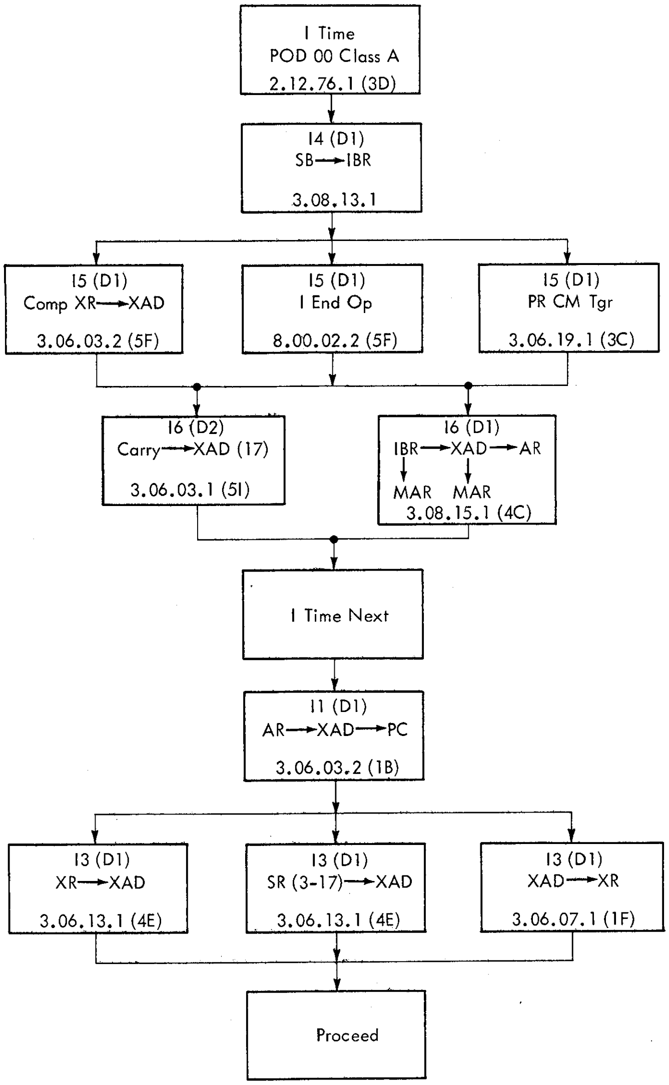


Figure 19. Transfer with Index Incremented (TXI)

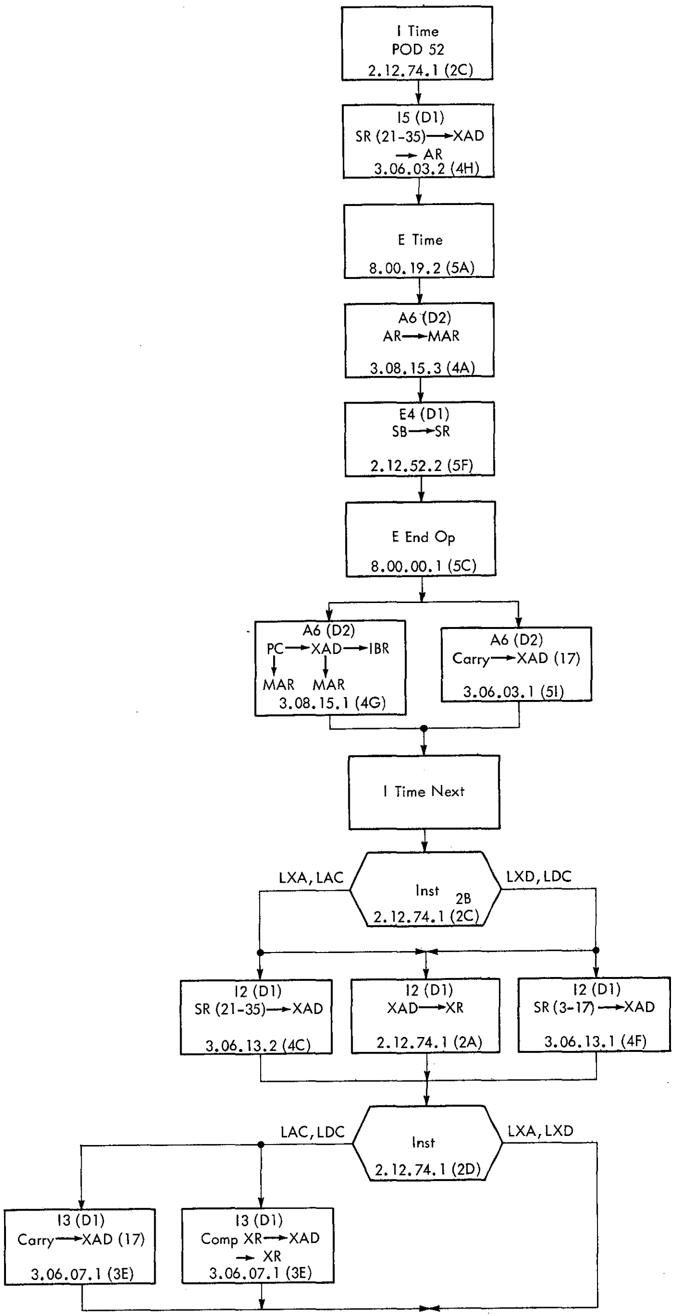


Figure 20. Load Index from Address (LXA); Load Index from Decrement (LXD); Load Complement of Address in Index Register (LAC); Load Complement of Decrement in Index Register (LDC)

Load Index From Decrement LXD — 0534 (I, E)

The contents of the specified index register are replaced by the contents of storage location X (3-17). Storage location X is unchanged. The storage register (3-17) is gated through the index adders to the specified index register at I2 of the following I cycle. See Figure 20.

Load Complement of Address in Index Register LAC + 0535 (I, E)

The contents of the specified index register are replaced by the two's complement of storage location X (21-35). Storage location X is unchanged. At I2 of the I cycle following LAC, the storage register (21-35) is gated through the index adders to the index register. At I3, the complement of the index register is routed through the index adders with a carry to index adder 17, back to the index register. See Figure 20.

Load Complement of Decrement in Index Register LDC — 0535 (I, E)

The contents of the specified index register are replaced by the two's complement of storage location X (3-17). Storage location X is unchanged. At I2 of the I cycle following LDC, the storage register (3-17) is gated through the index adders to the index register. At I3, the complement of the index register is routed through the index adders with a carry to index adder 17, back to the index register. See Figure 20.

Place Address in Index PAX + 0734 (I)

The contents of the accumulator (21-35) replace the contents of the specified index register. The accumulator contents are unchanged.

At I1 of the I cycle following PAX, the accumulator (1-35) is gated to the storage register. At I2, the storage register (21-35) is routed through the index adders to the specified index register. See Figure 21.

Place Complement of Address in Index PAC + 0737 (I)

The two's complement of the contents of the accumulator (21-35) replaces the contents of the specified index register. The accumulator contents are unchanged.

This instruction is performed in the same way as PAX except for complementing: at I3 of the I cycle following PAC, the complement of the index register is routed through the index adders with a carry to index adder 17, and back to the index register. See Figure 21.

Place Decrement in Index PDX – 0734 (I)

The contents of the accumulator (3-17) replace the contents of the specified index register. The accumulator contents are unchanged.

At I1 of the I cycle following PDX, the accumulator (1-35) is gated to the storage register. At I2, the storage register (3-17) is routed through the index adders to the specified index register. See Figure 21.

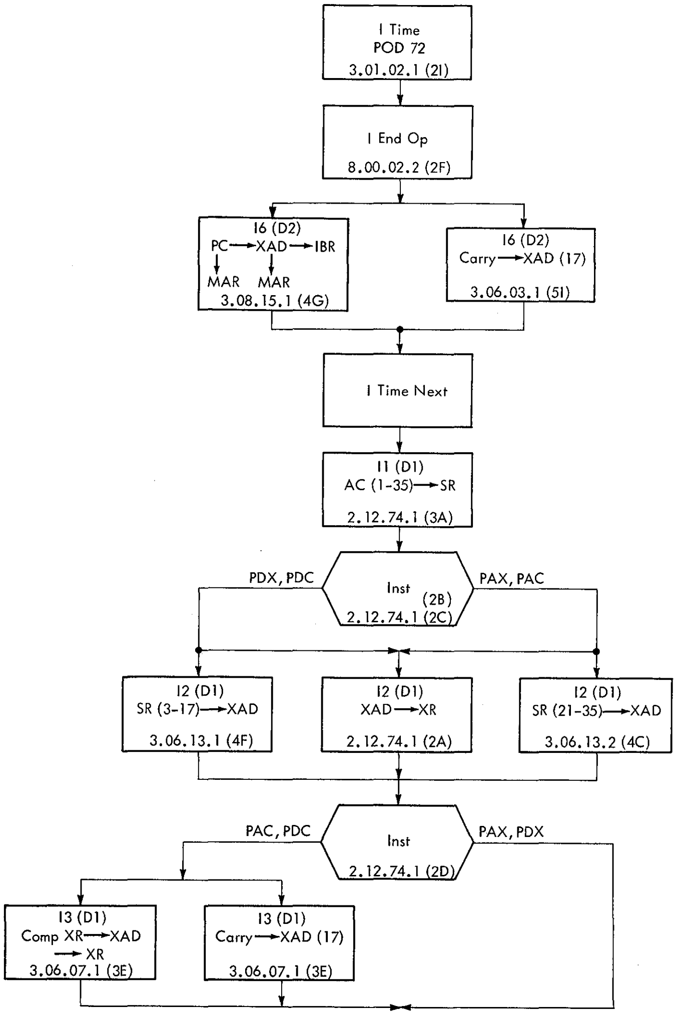


Figure 21. Place Address in Index (PAX); Place Complement of Address in Index (PAC); Place Decrement in Index (PDX); Place Complement of Decrement in Index (PDC)

Place Complement of Decrement in Index PDC – 0737 (I)

The two's complement of the contents of the accumulator (3-17) replaces the contents of the specified index register. The accumulator contents are unchanged.

This instruction is performed the same as is PDX except for complementing: at I3 of the I cycle following PDC, the complement of the index register is routed through the index adders with a carry to index adder (17), and back to the index register. See Figure 21.

Place Index in Address PXA + 0754 (I)

The contents of the specified index register replace the contents of the accumulator (21-35). The remainder of the accumulator is cleared. No tag clears the accumulator. In multiple tag mode, a multiple tag results in the OR of the specified index registers being placed in the accumulator (21-35), and in the index registers.

At I2 of the I cycle following PXA, the index register is routed through the index adders to the storage register (21-35), and to the index register. At I3, the storage register is gated to the adders, and from the adders to the accumulator. See Figure 22.

Place Index in Decrement PXD — 0754 (I)

The contents of the specified index register replace the contents of the accumulator (3-17). The remainder of the accumulator is cleared. A tag of zero clears the accumulator. In multiple tag mode, a multiple tag results in the OR of the specified index registers being placed in the accumulator (3-17), and in the index registers.

At I2 of the I cycle following PXD, the index register is routed through the index adders to the storage register (3-17); the storage register is gated to the adders, and from the adders to the accumulator. See Figure 22.

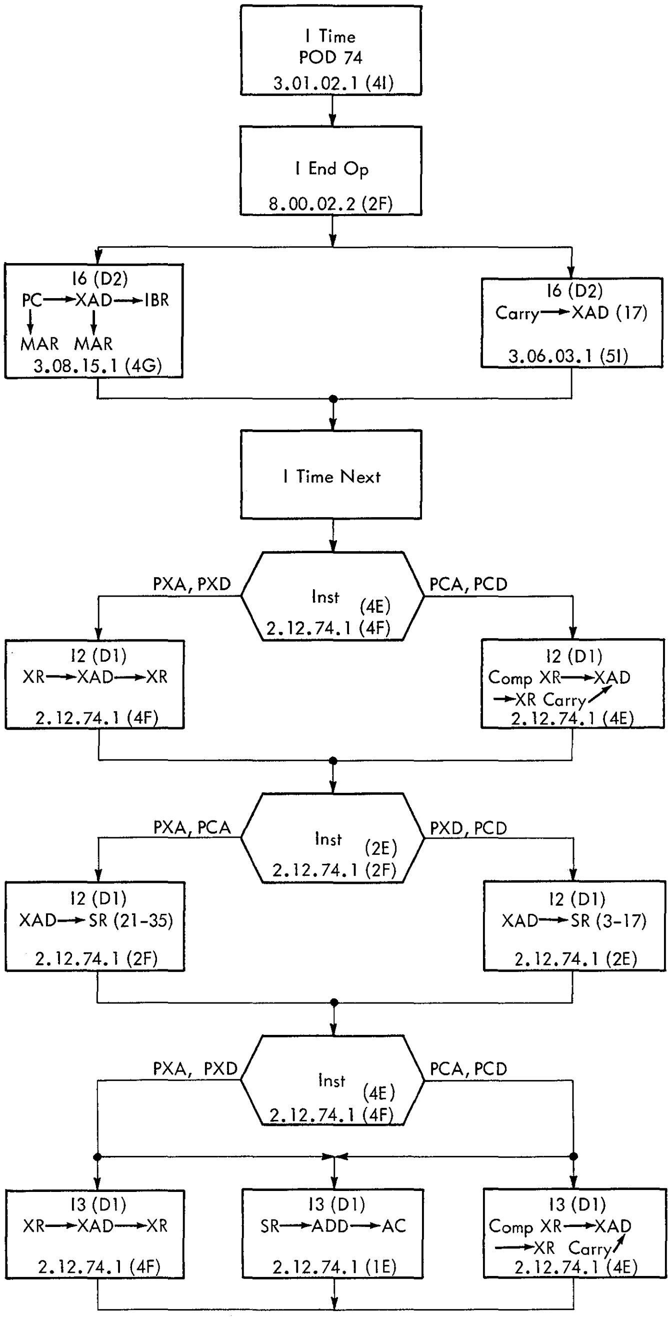


Figure 22. Place Index in Address (PXA); Place Index in Decrement (PXD); Place Complement of Index in Address (PCA); Place Complement of Index in Decrement (PCD)

Place Complement of Index in Address PCA + 0756 (I)

The two's complement of the specified index register replaces the contents of the accumulator (21-35). The remainder of the accumulator is cleared. No tag clears the accumulator. In multiple tag mode, a multiple tag results in the two's complement OR of the specified index registers being placed in the accumulator (21-35), and the OR in the index registers.

At I2 of the I cycle following PCA, the complement of the index register is routed through the index adders with a carry to index adder (17) to the storage register (21-35), and to the index register. At I3 the storage register is gated to the adders, and from the adders to the accumulator. Also at I3, the complement of the index register is routed through the index adders with a carry to index adder (17) to the index register. This recomplementing restores the index register to its original value. See Figure 22.

Place Complement of Index in Decrement PCD — 0756 (I)

The two's complement of the specified index register replaces the contents of the accumulator (3-17). The remainder of the accumulator is cleared. No tag clears the accumulator. In multiple tag mode, a multiple tag results in the two's complement OR of the specified index registers being placed in the accumulator (3-17), and the OR in the index registers.

At I2 of the I cycle following PCD, the complement of the index register is routed through the index adders with a carry to index adder (17) to the storage register (3-17), and the index register. At I3 the storage register is gated to the adders, and from the adders to the accumulator. Also at I3, the complement of the index register is routed through the index adders with a carry to index adder 17 to the index register. This recomplementing restores the index register to its original value. See Figure 22.

Store Index in Address SXA + 0634 (I, E)

The contents of the specified index register replace the contents of storage location X (21-35). Storage location X (S-20) is not changed. A tag of zero will store zeros in storage location X (21-35). The index register is unaltered unless a multiple tag is used when the computer is in multiple tag mode. Here, the OR of the specified index registers replaces the contents of these index registers as well as the contents of storage location X (21-35).

At E0 the index register is gated to the index adders. At this time, the index adders are gated to the storage register (21-35), and to the index register. At E1 the storage register is gated to the storage bus. See Figure 23.

Store Index in Decrement SXD — 0634 (I, E)

The contents of the specified index register replace the contents of storage location X (3-17). Storage location X (S-2, 18-35) is not changed. A tag of zero will store zeros in storage location X (3-17). The index register is unaltered unless a multiple tag is used when the computer is in multiple tag mode. Here, the OR of the spec-

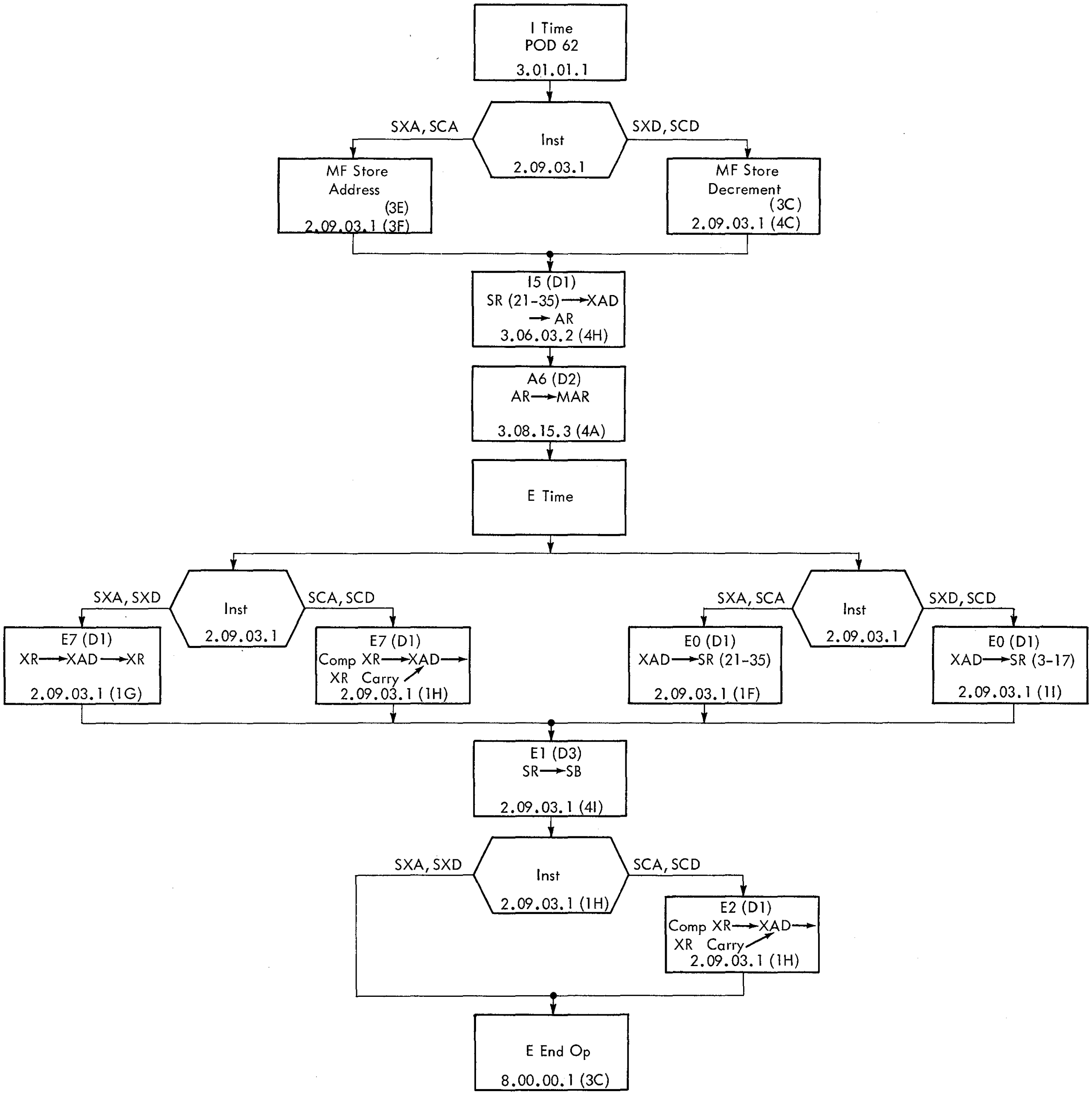


Figure 23. Store Index in Address (SXA); Store Index in Decrement (SXD); Store Complement of Index in Address (SCA); Store Complement of Index in Decrement (SCD)

ified index registers replaces the contents of these index registers as well as the contents of storage location X (3-17).

At E0 the index register is gated to the index adders. At this time, the index adders are gated to the storage register (3-17), and to the index register. At E1 the storage register is gated to the storage bus. See Figure 23.

Store Complement of Index in Address SCA + 0636 (I, E)

The two's complement of the specified index register replaces the contents of storage location X (21-35). Storage location X (S-20) is not changed. A tag of zero will store zeros in storage location X (21-35). The index register is unaltered unless a multiple tag is used when the computer is in multiple tag mode. Here, the two's complement OR of the specified index registers replaces the contents of storage location X (21-35) and the OR replaces the contents of the specified index registers.

At E0 the index register complement is gated to the index adders with a carry to index adder 17. At this time, the index adders are gated to the storage register (21-35), and to the index register. At E1 the storage register is gated to the storage bus. At E2 the index register complement is once again gated to the index adders with a carry to index adder 17 and to the index register. This restores the index register to its original value. See Figure 23.

Store Complement of Index in Decrement SCD − 0636 (I, E)

The two's complement of the specified index register replaces the contents of storage location X (3-17). Storage location X (S-2, 18-35) is not changed. A tag of zero will store zeros in storage location X (3-17). The index register is unaltered unless a multiple tag is used when the computer is in multiple tag mode. Here, the two's complement OR of the specified index registers replaces the contents of storage location X (21-35). and the OR replaces the contents of the specified index registers.

At E0 the index register complement is gated to the index adders with a carry to index adder 17. At this time, the index adders are gated to the storage register (3-17), and to the index register. At E1 the storage register is gated to the storage bus. At E2 the index register complement is once again gated to the index adders with a carry to index adder 17, and to the index register. This restores the index register to its original value. See Figure 23.

Address to Index True AXT + 0774 (I)

Positions 21-35 of this instruction replace the contents of the specified index register. The instruction remains unchanged.

At I2 of the I cycle following AXT, the storage register (21-35) is gated through the index adders to the specified index register. See Figure 24.

Address to Index Complemented AXC − 0774

The two's complement of positions 21-35 of this instruction replaces the contents of the specified index register. The instruction remains unchanged.

At I2 of the I cycle following AXC, the storage register (21-35) is gated through the index adders to the specified index register. At I3 the index register complement is gated to the index adders with a carry to index adder 17 back to the index register. See Figure 24.

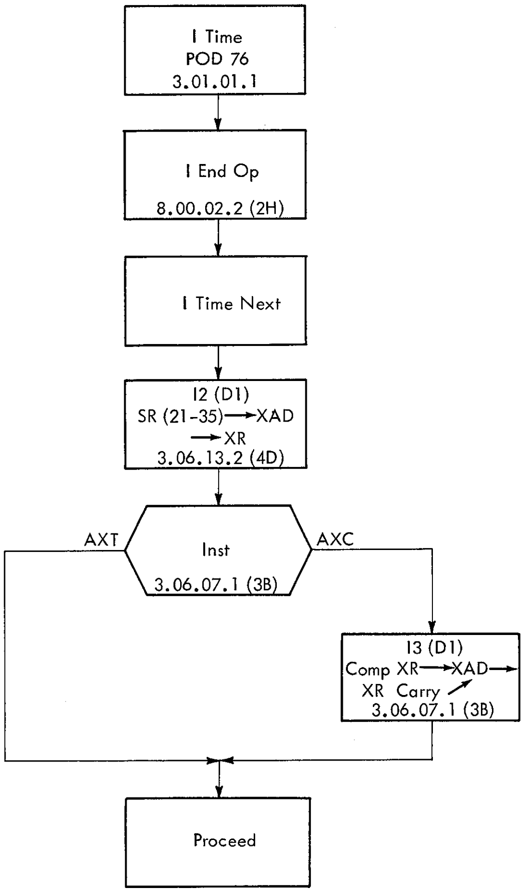


Figure 24. Address to Index True (AXT); Address to Index Complemented (AXC)

Shifting Instructions

Shift instructions align words or perform fast multiplication or division by powers of two. Because only eight positions are available in the shift counter, the maximum number of shifts possible is 377_8 or 255_{10}.

The shift count is gated through the index adders to the shift counter at I5. If the shift count is zero, at least one L cycle is still necessary to end operation, because there is not enough time available to determine a zero shift count and set the end-operation trigger at I5. When the shift count is six or less before L5, the instruction ends operation in that L cycle. Shifting starts at L1 of the first L cycle and shifts two places each clock pulse until the shift counter equals one or zero. Fourteen shifts may be performed during the first L cycle, and 16 shifts during subsequent L cycles. Right shifting is performed in the accumulator by gating the accumulator to the adders and the adders right one or two places back to the accumulator, making use of circuits available to arithmetic operations. A left shift gates directly from register-position to register-position.

Rotate Quotient Left RQL — 0773 (I, L . . . L)

The contents of the MQ (S, 1-35) are shifted left the number of places specified in positions 28-35 of this instruction. Bits leaving MQ(s) enter MQ 35. No bits are lost. See Figure 25.

Long Left Shift LLS + 0763 (I, L . . . . L)

The contents of the accumulator (Q, P, 1-35) and the contents of the MQ (1-35) are shifted left the number of places specified in positions 28-35 of this instruction. The accumulator sign is made to agree with the MQ sign. A bit entering accumulator P from accumulator 1 will turn on the accumulator overflow trigger. Bits shifted past accumulator Q are lost. Bits shifted past MQ1 enter accumulator 35. Vacated positions are filled with zeros. See Figure 26.

Logical Left Shift LGL — 0763 (I, L . . . L)

The contents of the accumulator (Q, P, 1-35) and the contents of the MQ (S, 1-35) are shifted left the number of places specified in positions 28-35 of this instruction. The accumulator sign is unchanged. A bit entering accumulator P from accumulator 1 turns on the accumulator overflow trigger. Bits shifted past accumulator Q are lost. Bits shifted past MQ(s) enter accumulator 35. Vacated positions are filled with zeros. See Figure 26.

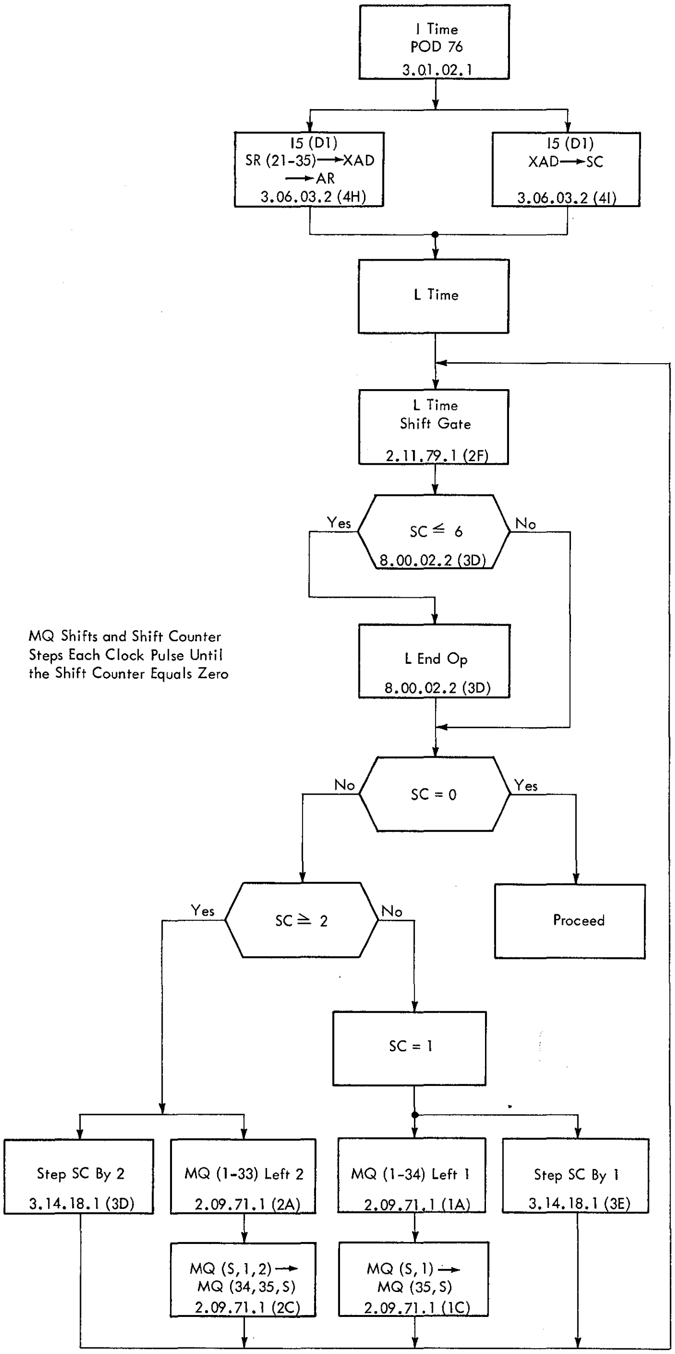


Figure 25. Rotate Quotient Left (RQL)

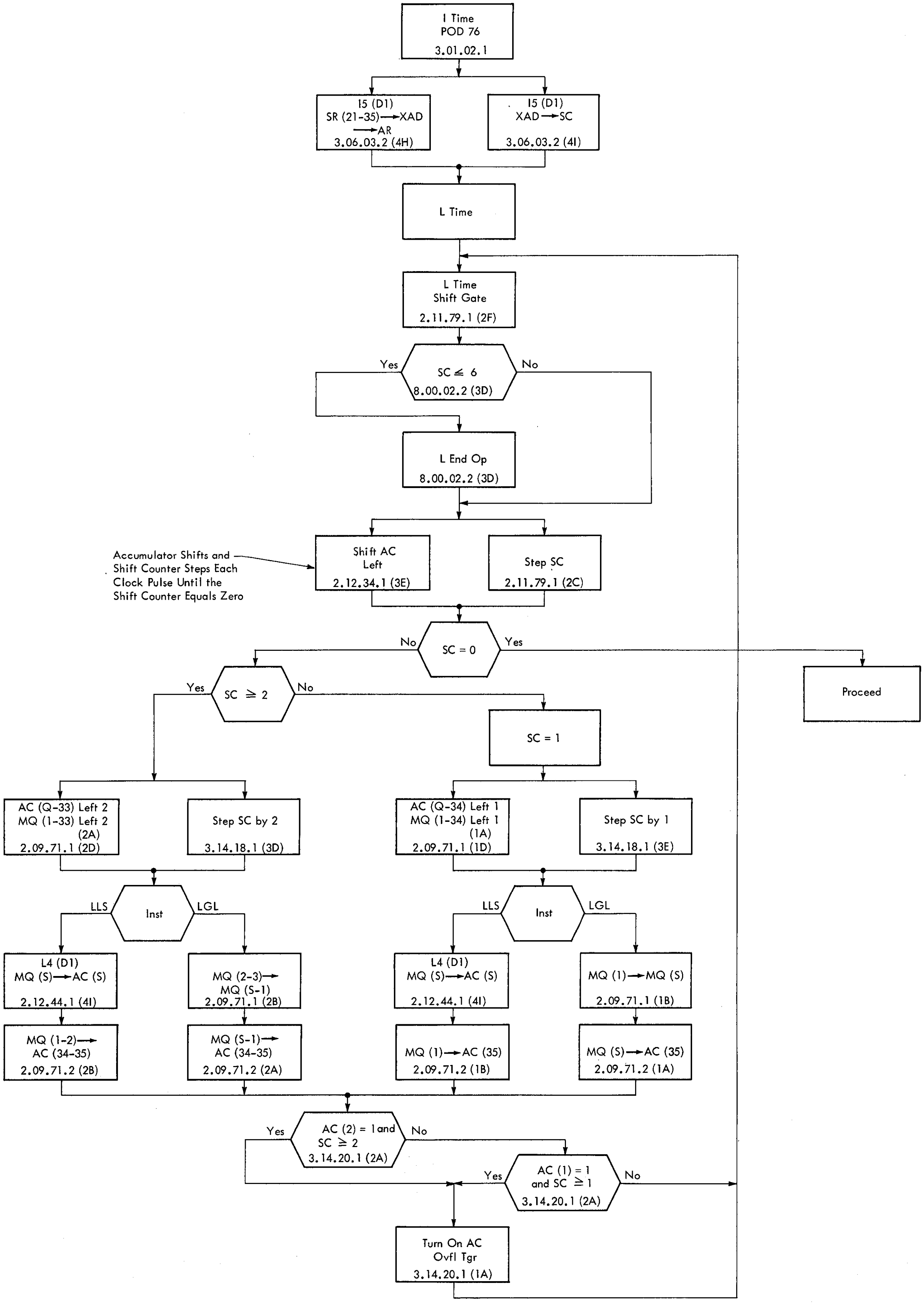


Figure 26. Long Left Shift (LLS); Logical Left Shift (LGL)

Accumulator Left Shift ALS + 0767 (I, L . . . L)

The contents of the accumulator (Q, P, 1-35) are shifted left the number of places specified in positions 28-35 of this instruction. The sign position is not changed. A bit entering accumulator P from accumulator 1 turns on the accumulator overflow trigger. Bits shifted past accumulator Q are lost. Vacated positions are filled with zeros. See Figure 27.

Accumulator Right Shift ARS + 0771 (I, L . . . L)

The contents of the accumulator (Q, P, 1-35) are shifted right the number of places specified in positions 28-35 of this instruction. The sign position is not changed. Bits shifted past accumulator 35 are lost. Vacated positions are filled with zeros. See Figure 27.

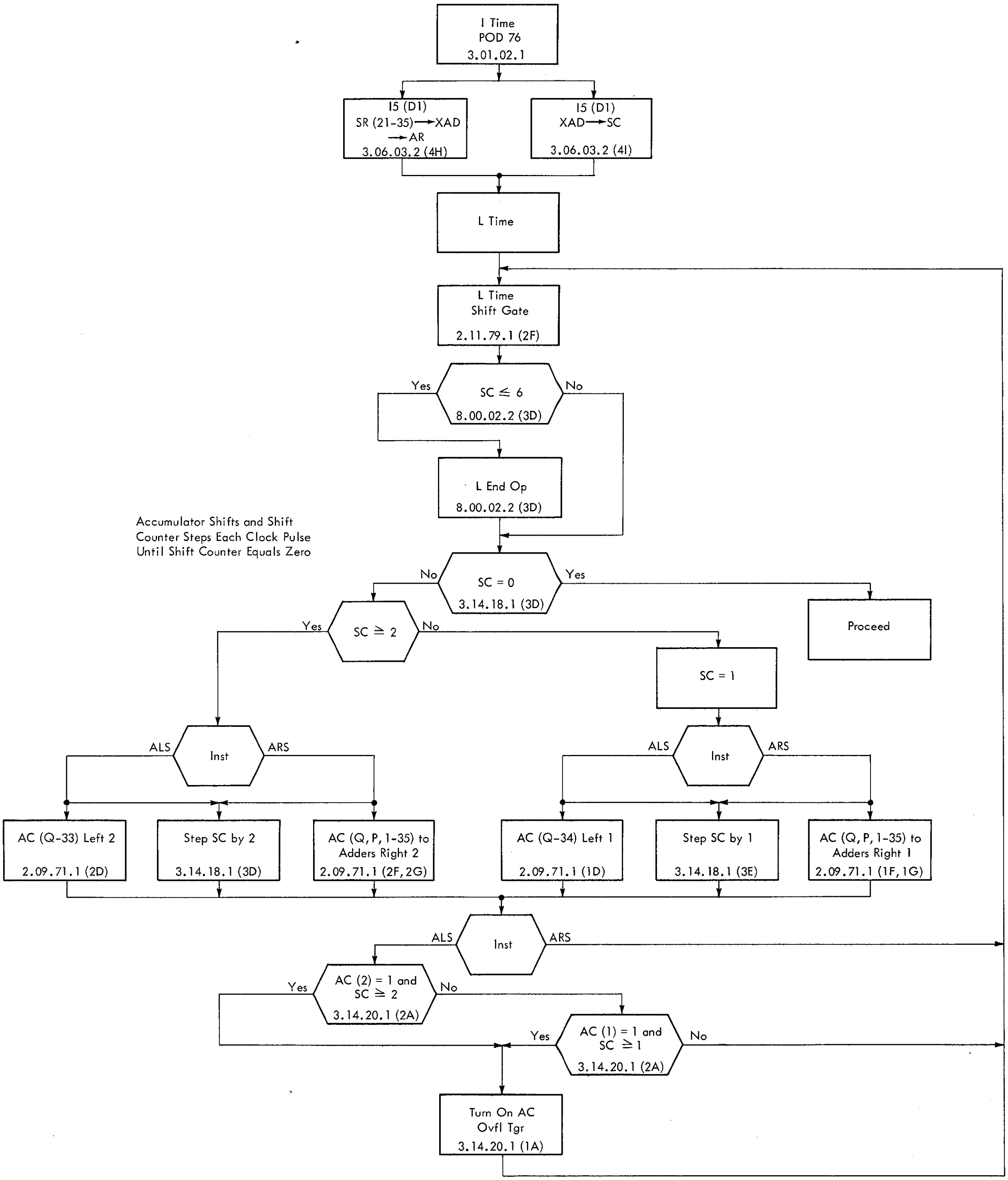


Figure 27. Accumulator Left Shift (ALS); Accumulator Right Shift (ARS)

Long Right Shift LRS + 0765 (I, L . . . L)

The contents of the accumulator (Q, P, 1-35) and the contents of the MQ (1-35) are shifted right the number of places specified in positions 28-35 of this instruction. The MQ sign is made to agree with the accumulator sign. Bits shifted past accumulator 35 enter MQ1. Bits shifted past MQ35 are lost. Vacated positions are filled with zeros. See Figure 28.

Logical Right Shift LGR − 0765 (I, L . . L)

The contents of the accumulator (Q, P, 1-35) and the contents of the MQ (S, 1-35) are shifted right the number of places specified in positions 28-35 of this instruction. The sign of the accumulator is not changed. Bits shifted past accumulator 35 enter MQ(S). Bits shifted past MQ35 are lost. Vacated positions are filled with zeros. See Figure 28.

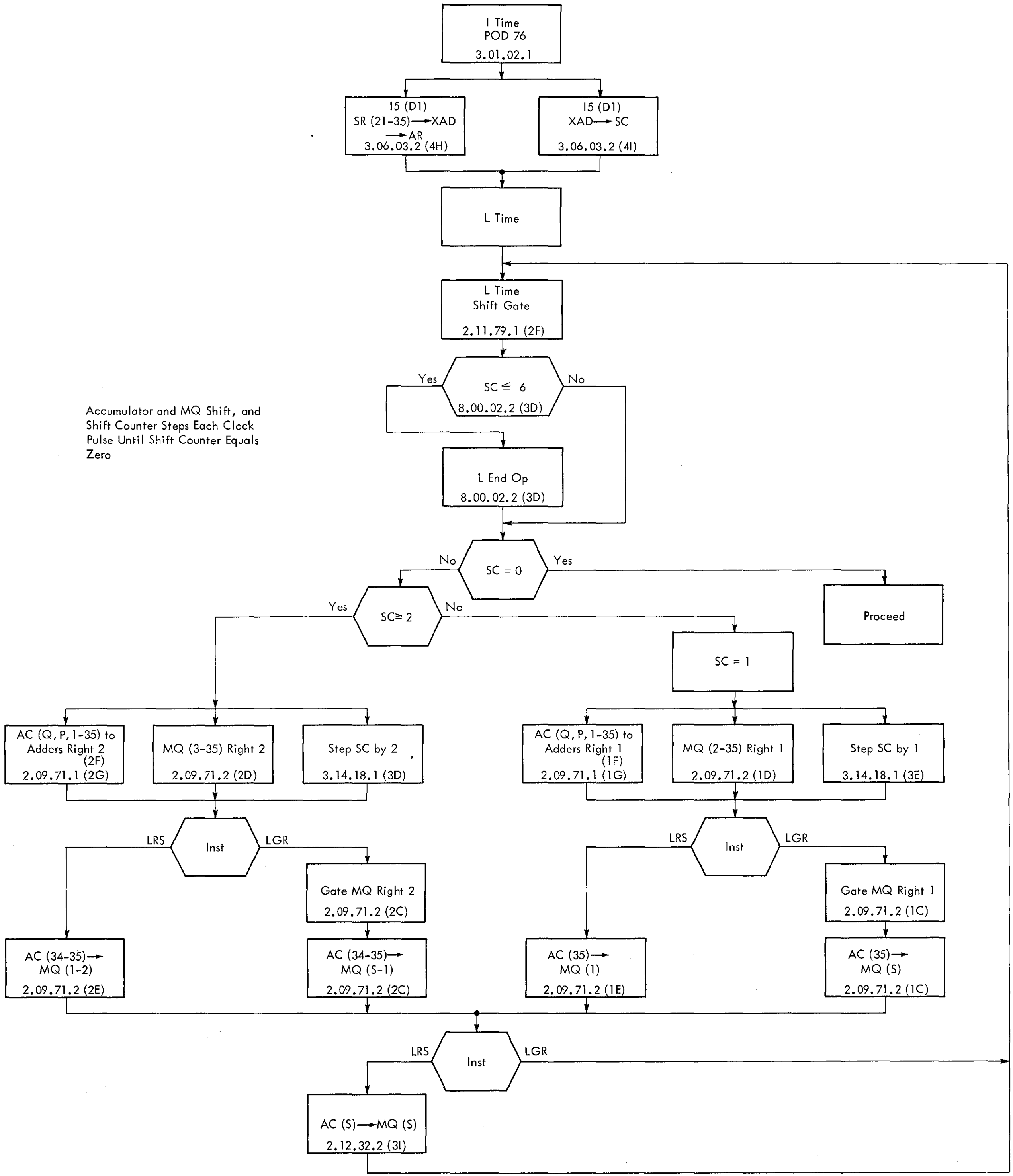


Figure 28. Long Right Shift (LRS); Logical Right Shift (LGR)

Sense Indicator Instructions

The 24 sense indicator instructions enable the computer to test, turn on or off, and invert the sense indicators under program control. These instructions operate on the sense indicator register.

The sense indicators are conditioned by the storage register contents, which are always present at the sense indicator inputs. Along with the storage register contents are the sense indicator control lines which effectively turn the sense indicators on or off. These control lines are:

1. "Load SI," which turns a sense indicator on if its equivalent storage register is on. This line cannot turn a sense indicator off under any condition.
2. "Invert or reset SI" will reset any sense indicator that is on, off if its equivalent storage register is on. This line cannot turn on a sense indicator under any condition.
3. "Invert or set SI" will turn on any sense indicator that is off, if its equivalent storage register is on. This line cannot turn a sense indicator off under any condition.

Any of these three control lines will affect a sense indicator only if its equivalent storage register position is on.

To invert a sense indicator, two of these control lines must be present at the same time. "Invert or reset SI" must be active to turn a sense indicator off; "invert or set SI" must be active to turn a sense indicator on.

Store Indicators STI + 0604 (I, E)

The contents of storage location X (S, 1-35) are replaced by the contents of sense indicators (0-35). The contents of the sense indicators are not changed. See Figure 1.

Load Indicators LDI + 0441 (I, E)

The contents of storage location X (S-35) replace the contents of sense indicators (0-35). Storage location X is not changed. See Figure 29.

OR Storage to Indicators OSI + 0442 (I, E)

If the corresponding position of storage location X or the sense indicator (or both) contain a one, a one is placed in the same position in the sense indicators. If the corresponding positions of location X and the sense indicators contains a zero, a zero is placed in that position of the sense indicators. Storage location X is not changed. See Figure 29.

Invert Indicators From Storage IIS + 0440 (I, E)

Each bit of core storage location X (S, 1-35) is matched with the corresponding position of sense indicators (0-35). When the position of location X contains a one, the corresponding position of the sense indicator is complemented. When the position of location X contains a zero, the corresponding position of the sense

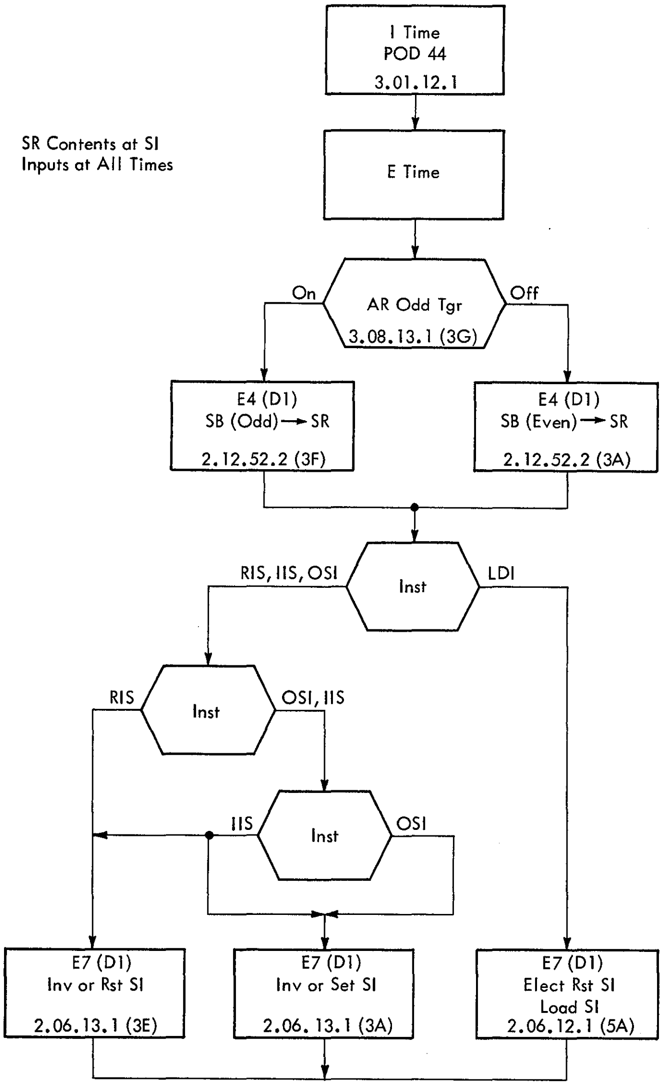


Figure 29. Load Indicators (LDI); OR Storage to Indicators (OSI); Invert Indicators from Storage (IIS); Reset Indicators from Storage (RIS)

indicator is not changed. Storage location X is not changed. See Figure 29.

Reset Indicators From Storage RIS + 0445 (I, E)

Each bit of storage location X (S-35) resets the corresponding position of the sense indicators (0-35). Storage location X is not changed. See Figure 29.

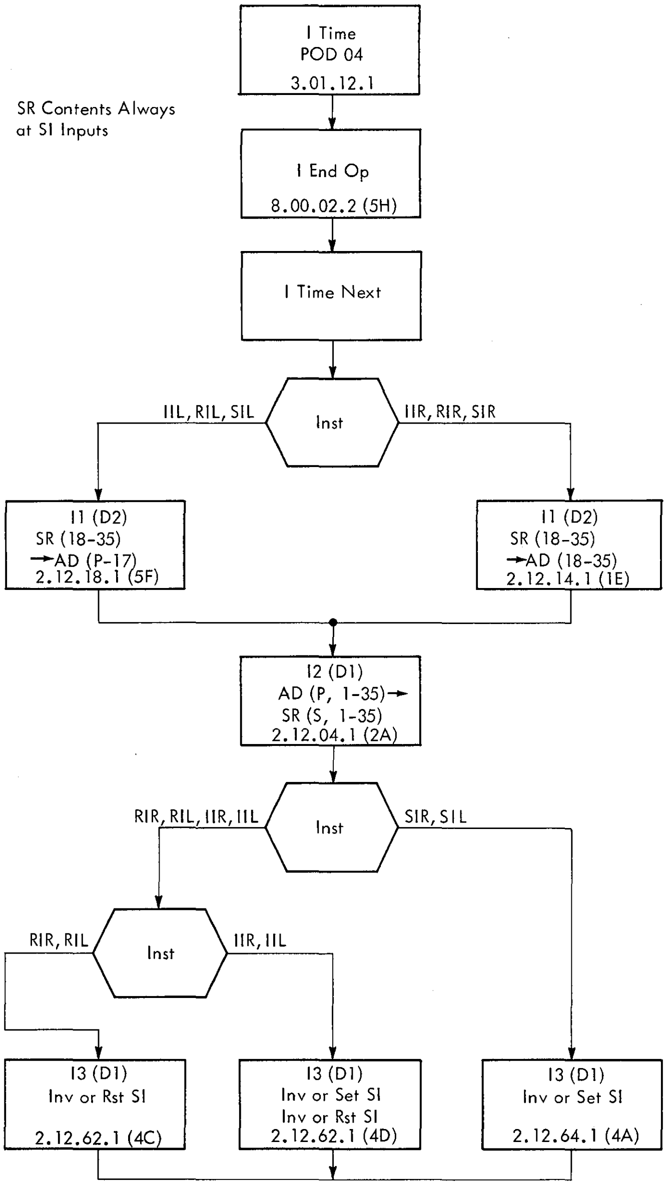


Figure 30. Invert Indicators of Right Half (IIR); Invert Indicators of Left Half (IIL); Set Indicators of Right Half (SIR); Set Indicators of Left Half (SIL); Reset Indicators of Right Half (RIR); Reset Indicators of Left Half (RIL)

Invert Indicators of Right Half IIR + 0051 (I)

Positions 18-35 of this instruction are matched with the sense indicators (18-35). If the position in the instruction contains a bit, the corresponding position of the sense indicators is complemented. The sense indicators (0-17) and this instruction are not changed. See Figure 30.

Invert Indicators of Left Half ILL − 0051 (I)

Positions 18-35 of this instruction are matched with sense indicators (0-17). If the position in the instruction contains a bit, the corresponding position of the sense indicators is complemented. Sense indicators (18-35) and this instruction are not changed. See Figure 30.

Set Indicators of Right Half SIR + 0055 (I)

Positions 18-35 of this instruction are matched with sense indicators (18-35). If the position in the instruction or the corresponding position of the sense indicator (or both) contain a bit, a one is placed in the same position of the sense indicators. If the corresponding positions are both zero, a zero is placed in that position of the sense indicators. Sense indicators (0-17) and this instruction are not changed. See Figure 30.

Set Indicators of Left Half SIL − 0055 (I)

Positions 18-35 of this instruction are matched with sense indicators (0-17). If the position in the instruction or the corresponding position of the sense indicator (or both) contain a bit, a one is placed in that position of the sense indicators. If the corresponding positions are both zero, a zero is placed in that position of the sense indicators. Sense indicators (18-35) and this instruction are not changed. See Figure 30.

Reset Indicators of Right Half RIR + 0057 (I)

Each bit in positions 18-35 of this instruction resets the corresponding position of sense indicators (18-35) to zero. Sense indicators (0-17) and this instruction are not changed. See Figure 30.

Reset Indicators of Left Half RIL − 0057 (I)

Each bit in positions 18-35 of this instruction resets the corresponding position of sense indicators (0-17) to zero. Sense indicators (18-35) and this instruction are not changed. See Figure 30.

Invert Indicators from Accumulator IIA + 0041 (I)

Accumulator (P-35) is matched with sense indicators (0-35). If the accumulator position contains a bit, the corresponding position of the sense indicators is complemented. If the accumulator position contains a zero, the corresponding sense indicator position is not changed. The accumulator is not changed. See Figure 31.

Reset Indicators From Accumulator RIA – 0042 (I)

Accumulator (P-35) is matched with the sense indicators (0-35). If the accumulator position contains a bit, the corresponding position of the sense indicators is reset to zero. If the accumulator position contains a zero, the corresponding sense indicator position is not changed. The accumulator is not changed. See Figure 31.

OR Accumulator to Indicators OAI + 0043 (I)

Accumulator (P-35) is matched with sense indicators (0-35). If the corresponding positions of the accumulator or sense indicator (or both) contain a one, a one is placed in that position of the sense indicator. If the corresponding positions of the accumulator and sense indicators contain zeros, a zero is placed in that position of the sense indicators. The accumulator is not changed. See Figure 31.

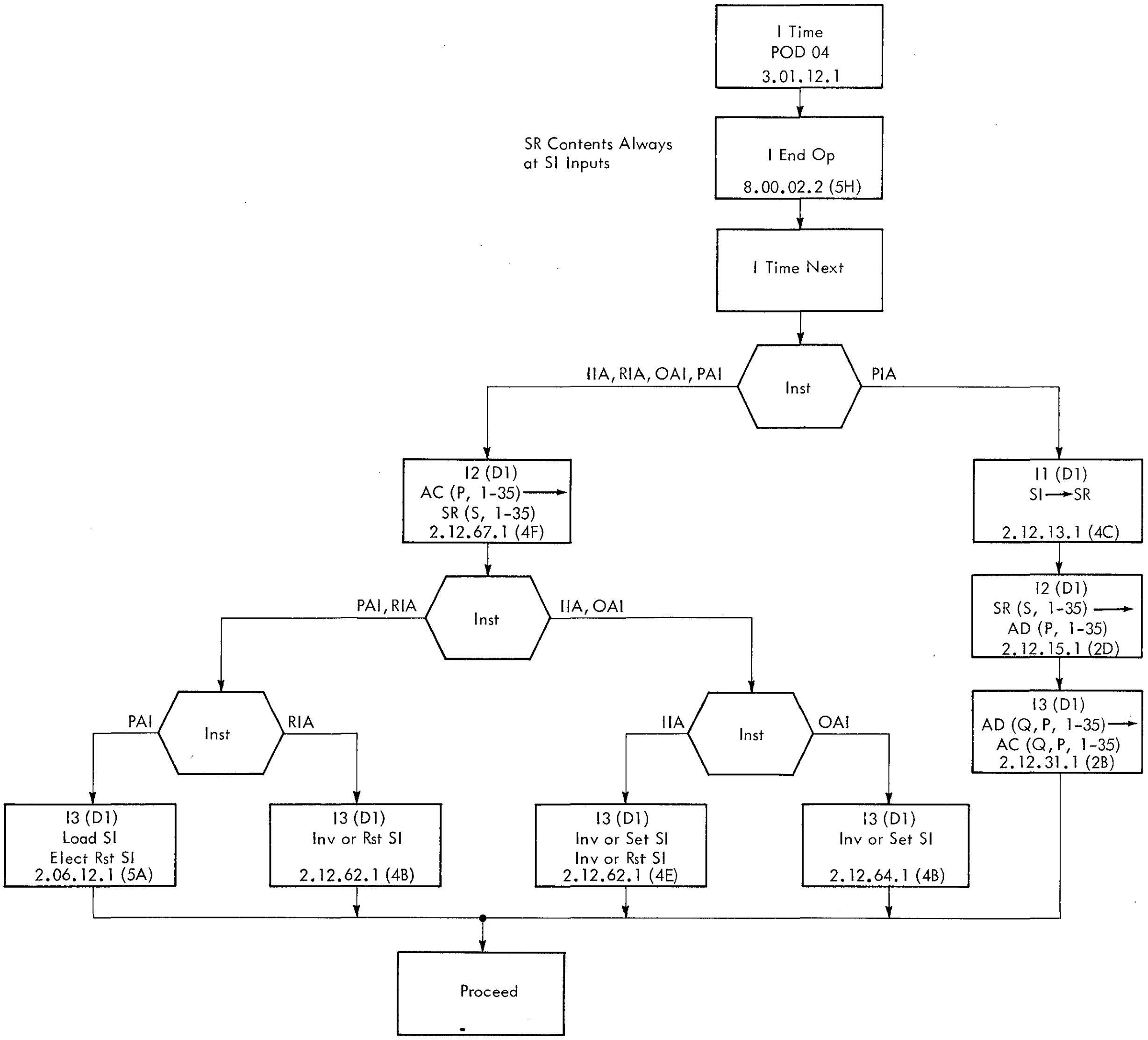


Figure 31. Invert Indicators from Accumulator (IIA); Reset Indicators from Accumulator (RIA); OR Accumulator to Indicators (OAI); Place Accumulator in Indicators (PAI); Place Indicators in Accumulator (PIA)

Place Accumulator in Indicators PAI + 0044 (I)

The contents of the accumulator (P-35) replace the contents of sense indicators (0-35). The accumulator is not changed. See Figure 31.

Place Indicators in Accumulator PIA − 0046 (I)

The contents of sense indicators (0-35) replace the contents of the accumulator (P-35). The sense indicators are not changed. Accumulator Q is reset. See Figure 31.

On Test for Indicators ONT + 0446 (I, E, L, L)

The contents of storage location X (S-35) are matched with the contents of the sense indicators (0-35). If each bit in storage location X is matched by a bit in the corresponding position of the sense indicators, the computer skips the next instruction. If the ones are not matched, the computer takes the next instruction in sequence. Storage location X and the sense indicators are not changed.

The test is accomplished by complementing the storage location, OR'ing this value with the sense indicators and testing for an adder (P) carry when the OR is gated to the adders with a carry to adder (35). An adder (P) carry forces a skip. See Figure 32.

Off Test for Indicators OFT + 0444 (I, E, L, L)

The contents of storage location X (S-35) are matched with the contents of sense indicators (0-35). If each bit in storage location X is matched by a zero in the corresponding position of the sense indicators, the computer skips the next instruction. If the ones are not matched by zeros, the computer takes the next instruction in sequence. Storage location X and the sense indicators are not changed.

The test is accomplished by complementing the contents of the sense indicators and the storage location, OR'ing the two values, and gating the OR to the adders with a carry to adder (35). An adder (P) carry forces a skip. See Figure 32.

Left Half Indicators Off Test LFT − 0054 (I, L, L)

Positions 18-35 of this instruction are compared with sense indicators (0-17). If each bit in positions 18-35 of this instruction is matched by a zero in the corresponding position of the sense indicators, the computer skips the next instruction. If the ones are not matched by zeros, the computer takes the next instruction in sequence. This instruction and the sense indicators are not changed.

The test is performed by complementing the sense indicators and the contents of this instruction (18-35), OR'ing the two values, then testing for an adder (P) carry when the OR is gated to the adders with a carry to adder (35). See Figure 33.

Right Half Indicators Off Test RFT + 0054 (I, L, L)

Positions 18-35 of this instruction are compared with sense indicators (18-35). If each bit in positions 18-35 of this instruction is matched by a zero in the corresponding position of the sense indicators, the computer skips the next instruction. If the ones are not matched by zeros, the computer takes the next instruction in sequence. This instruction and the sense indicators are not changed.

The test is performed by complementing the sense indicators and the contents of this instruction (18-35), OR'ing the two values, then testing for an adder (P) carry when the OR is gated to the adders with a carry to adder (35). See Figure 33.

Left Half Indicators On Test LNT − 0056 (I, L, L)

Positions 18-35 of this instruction are compared with sense indicators (0-17). If each bit in positions 18-35 of this instruction is matched by a bit in the corresponding position of the sense indicators, the computer skips the next instruction. If the bits are not matched, the computer takes the next instruction in sequence. This instruction and the sense indicators are not changed.

The test is performed by complementing the contents of this instruction (18-35), OR'ing this value with the sense indicators (0-17), then testing for an adder (P) carry when the OR is gated to the adders with a carry to adder (35). See Figure 33.

Right Half Indicators On Test RNT + 0056 (I, L, L)

Positions 18-35 of this instruction are compared with sense indicators (18-35). If each bit in positions 18-35 of this instruction is matched by a bit in the corresponding position of the sense indicators, the computer skips the next instruction. If the bits are not matched, the computer takes the next instruction in sequence. This instruction and the sense indicators are not changed.

The test is performed by complementing the contents of this instruction (18-35), OR'ing this value with sense indicators (18-35), then testing for an adder (P) carry when the OR is gated to the adders with a carry to adder (35). See Figure 33.

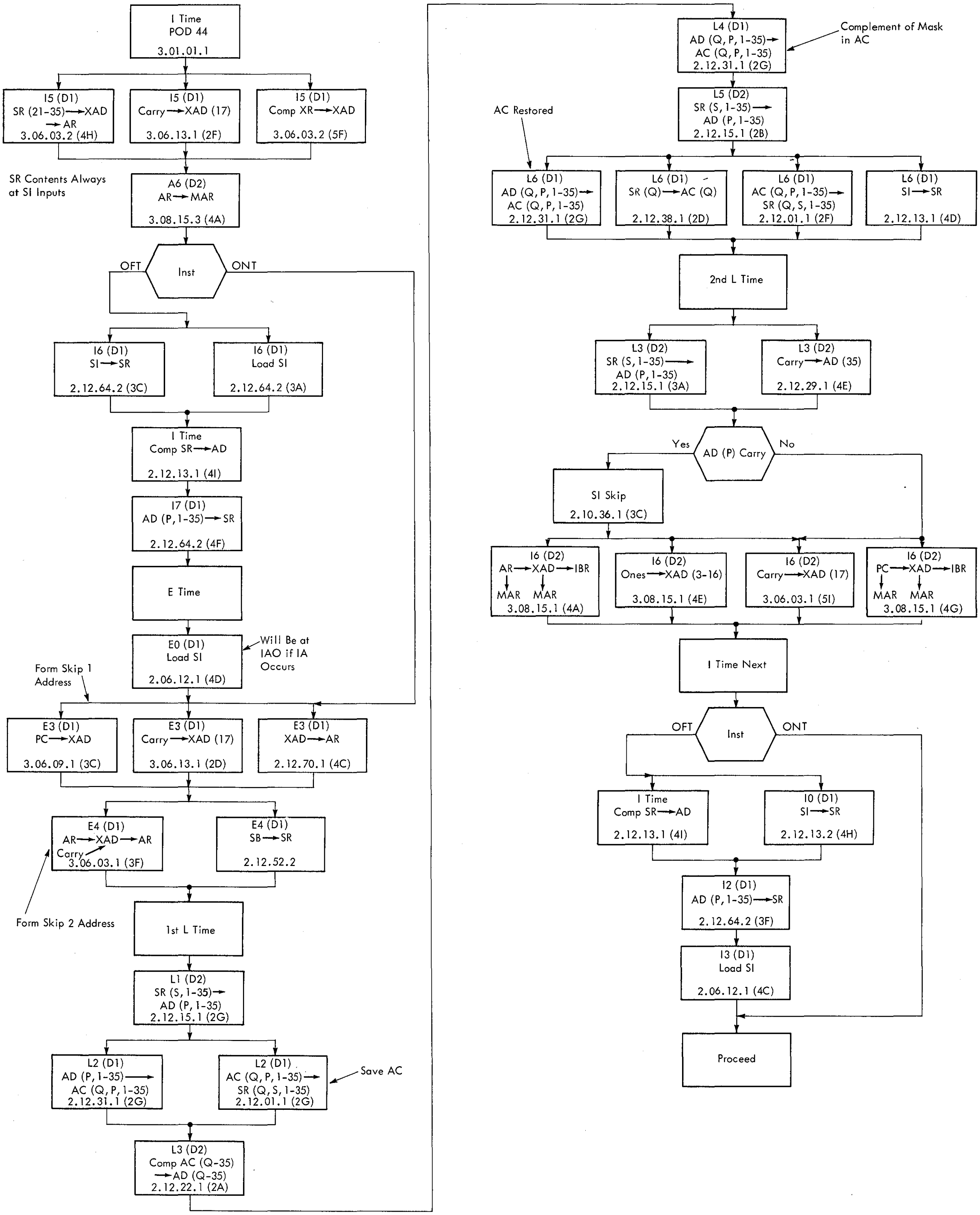


Figure 32. On Test for Indicators (ONT); Off Test for Indicators (OFT)

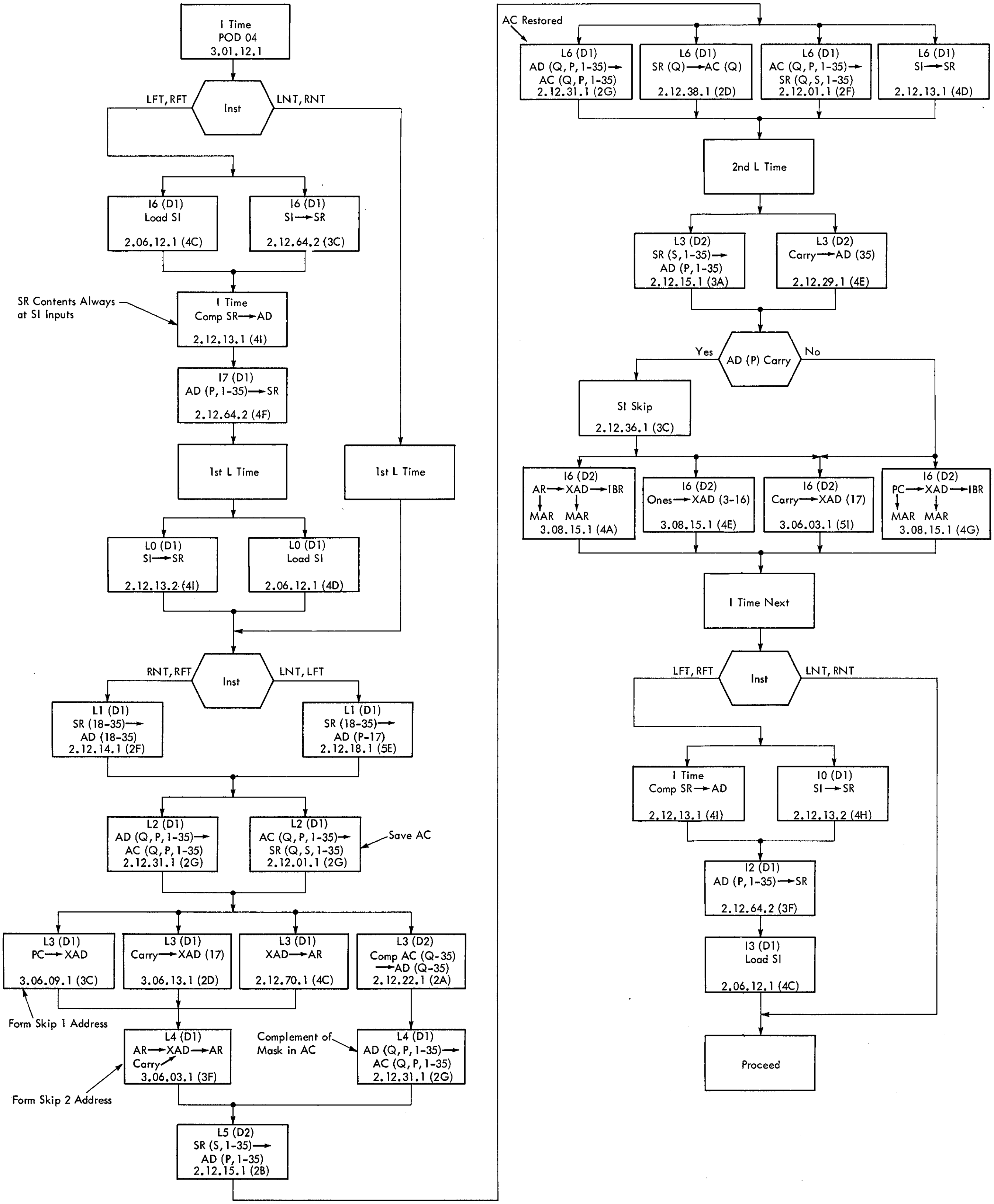


Figure 33. Left Half Indicators Off Test (LFT); Right Half Indicators Off Test (RFT); Left Half Indicators On Test (LNT); Right Half Indicators On Test (RNT)

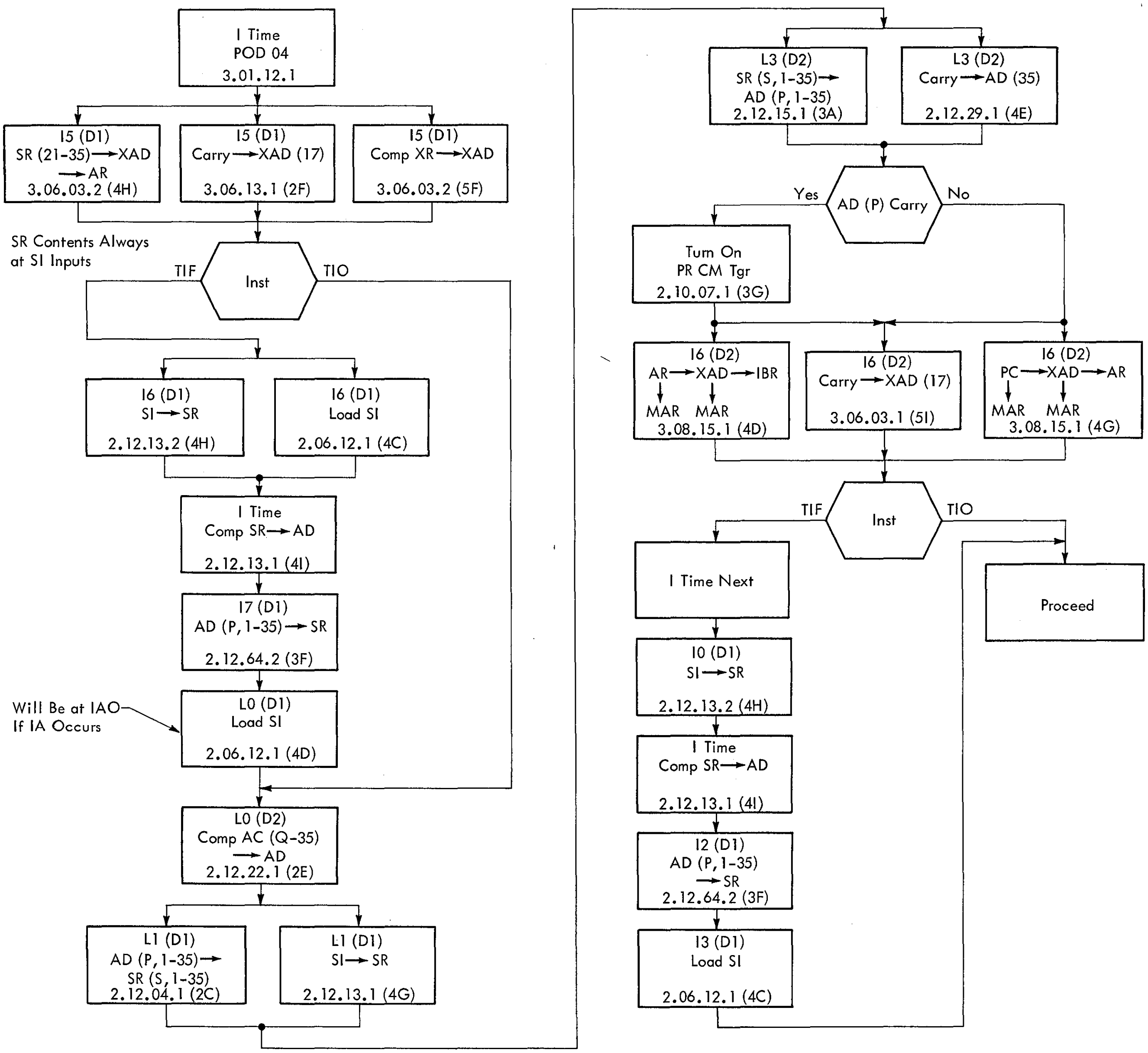


Figure 34. Transfer When Indicators On (TIO); Transfer When Indicators Off (TIF)

Transfer When Indicators On TIO + 0042 (I, L)

The contents of the accumulator (P-35) are matched with sense indicators (0-35). If the bits in the accumulator are matched by bits in the corresponding positions of the sense indicators, the computer transfers to location X. If the bits are not matched, the computer takes the next sequential instruction.

The complement of the accumulator is OR'd with the contents of the sense indicators. The OR is gated to the adders with a carry to adder (35), and a test is made for an adder (P) carry. See Figure 34.

Transfer When Indicators Off TIF + 0046 (I, L)

The contents of the accumulator (P-35) are matched with sense indicators (0-35). If the bits in the accumulator are matched by zeros in the corresponding positions of the sense indicators, the computer transfers to location X. If the bits are not matched by zeros, the computer takes the next sequential instruction.

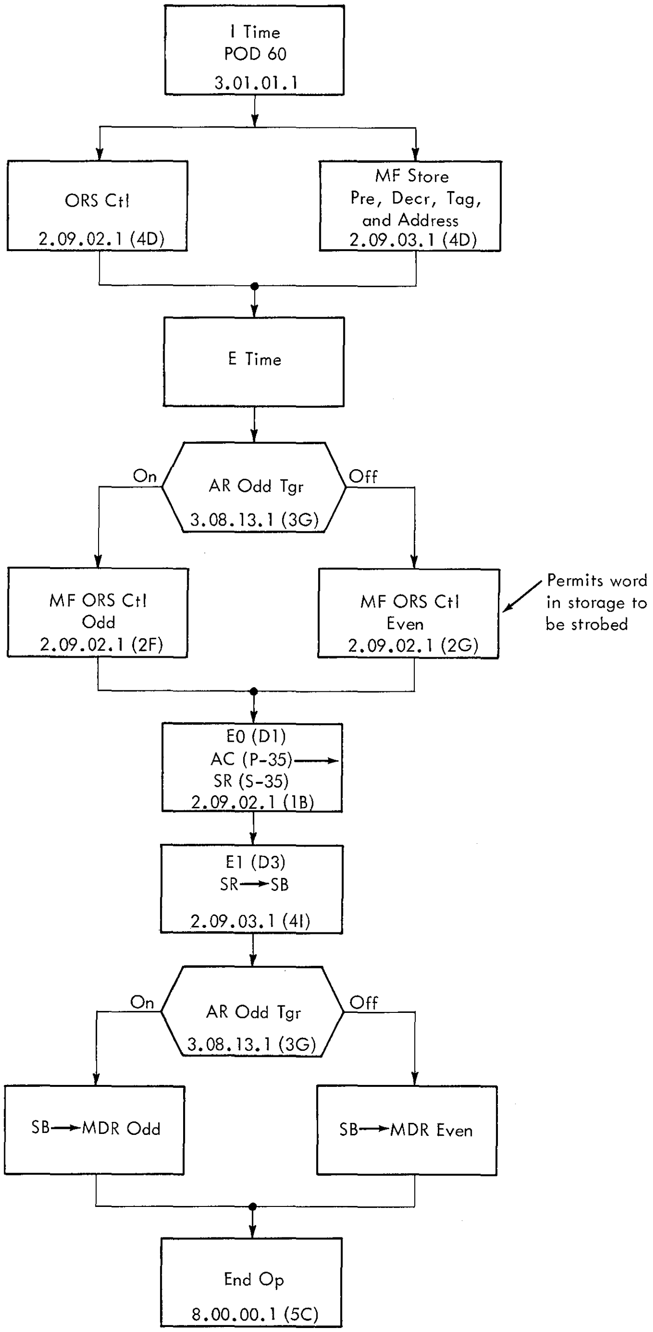
The complement of the accumulator is OR'd with the complement of the sense indicators. The OR is gated to the adders with a carry to adder (35), and a test is made for an adder (P) carry. See Figure 34.

The AND and OR instructions are used to produce logical combinations of bits, useful for masking or matching words.

OR to Storage ORS – 0602 (I, E)

The contents of the accumulator (P-35) are matched with the contents of storage location X (S-35). If the corresponding position of the accumulator or storage location X (or both) contains a one, a one is placed in the same position of storage location X. If the corresponding positions of the accumulator and location X both contains zeros, a zero is placed in that position of location X. The accumulator is not changed.

The OR is developed in the memory data register by gating the accumulator and location X to the MDR during the E cycle. See Figure 35.



Permits word in storage to be strobed

Figure 35. OR to Storage (ORS)

OR to Accumulator ORA – 0501 (I, E)

The contents of storage location X (S-35) are matched with the contents of the accumulator (P-35). If the corresponding position of the accumulator or location X (or both) contains a one, a one is placed in the same position of the accumulator. If the corresponding positions of the accumulator and location X both contain zeros, a zero is placed in that position of the accumulator. Storage location X and positions S and Q of the accumulator are not changed.

The OR is developed by gating the SB and the accumulator to the storage register at the same time. See Figure 36.

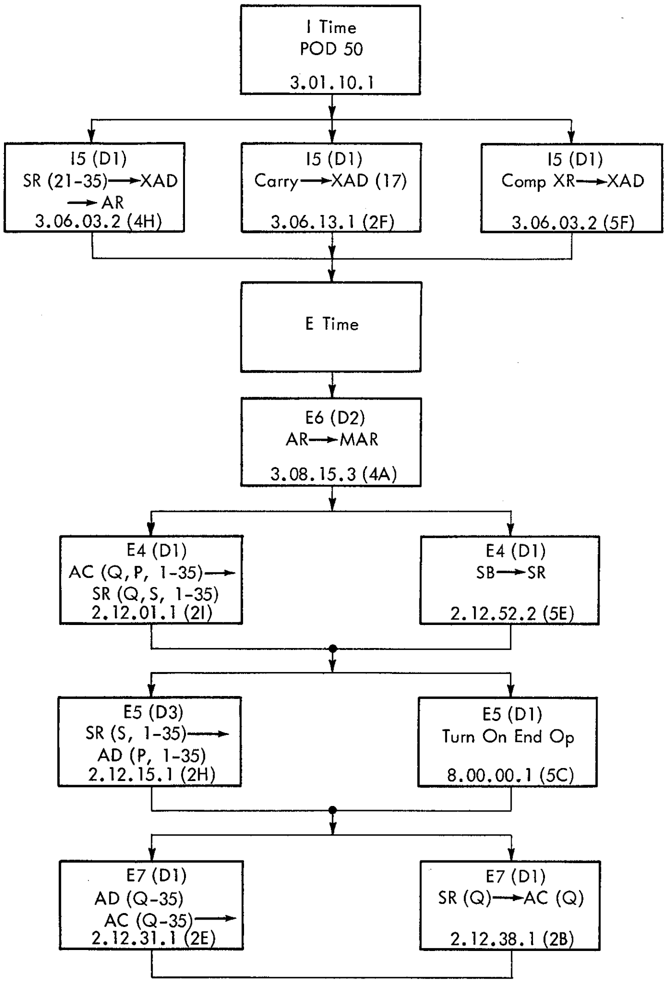


Figure 36. OR to Accumulator (ORA)

AND to Accumulator ANA – 0320 (I, E, L)

The contents of location X (S-35) are matched with the contents of the accumulator (P-35). If the corresponding positions of both the accumulator and location X are ones, a one is placed in the same position of the accumulator. If the corresponding position of either the accumulator or location X is a zero, a zero is placed in that position of the accumulator. Storage location X is not changed. Positions S and Q of the accumulator are cleared.

The AND is obtained by OR'ing the complement of both words, then complementing this value. See Figure 37.

AND to Storage ANS + 0320 (I, E, L, E)

The contents of storage location X (S-35) are matched with the contents of the accumulator (P-35). If the corresponding positions of both the accumulator and location X are ones, a one is placed in the same position of location X. If the corresponding positions of either the accumulator or location X is a zero, a zero is placed in that position of location X. The accumulator is not changed.

The AND is obtained by OR'ing the complement of both words and then complementing this value. See Figure 37.

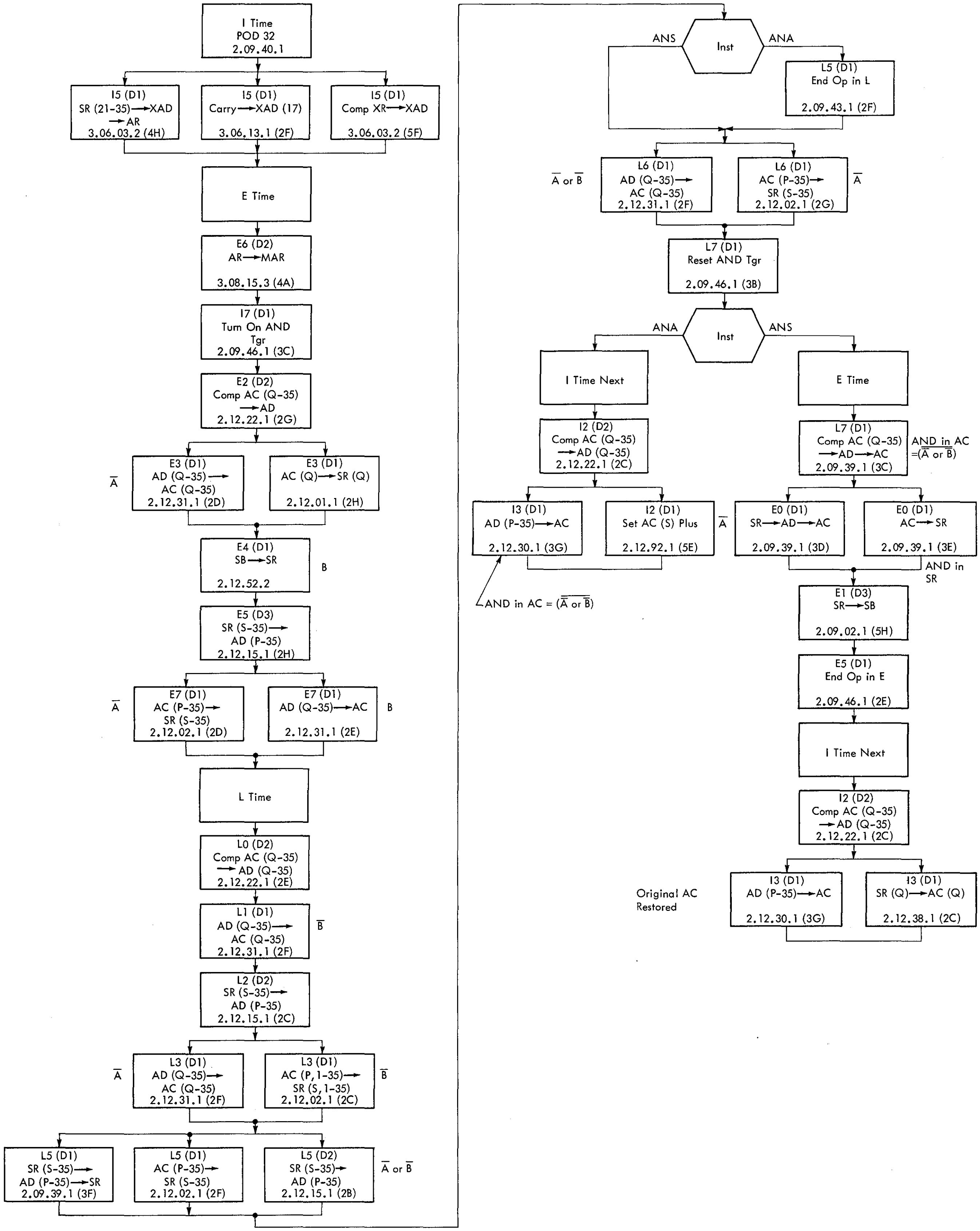


Figure 37. AND to Accumulator (ANA); AND to Storage (ANS)

Exclusive OR to Accumulator ERA + 0322 (I, E, L)

The contents of the accumulator (P-35) are matched with the contents of storage location X (S-35). If the corresponding position of the accumulator matches the position in location X, a zero is placed in the same position of the accumulator. If the corresponding position of the accumulator does not match the position in location X, a one is placed in that position of the accumulator. Storage location X is not changed. Positions S and Q of the accumulator are cleared. See Figure 38.

The Exclusive OR is obtained by use of the equation: Ex OR = 2 (A or B) − (A+B). The derivation of the preceding equation can be shown with the adder table:

Factor A	00011
Factor B	00101
A + B (carry not blocked)	01000
Carry to be blocked	00010
A + B (carry blocked)	00110
Ex OR (A + B carry blocked)	00110
A or B	00111
2 (A + B)	10000
2 (A or B)	01110

From this table one can see that the Exclusive OR equals the sum output of the adders with all carries blocked. The carries cannot be blocked, but the Ex OR can be obtained by subtracting an amount equal to the blocked carries from the sum of the two words:

1. Ex OR = (A + B) − blocked carries. The blocked carries can be simulated by subtracting twice the OR from twice the sum of two words.

2. Blocked carries = 2 (A + B) − 2 (A or B) = 10000 − 01110 = 00010. Substituting equation 2 in equation 1:

3. Ex OR = (A + B) − [2 (A + B) − 2 (A or B)] = 01000 − 00010 = 00110 and simplifying equation 3:

4. Ex OR = 2 (A or B) − (A + B) = 01110 − 01000 = 00110.

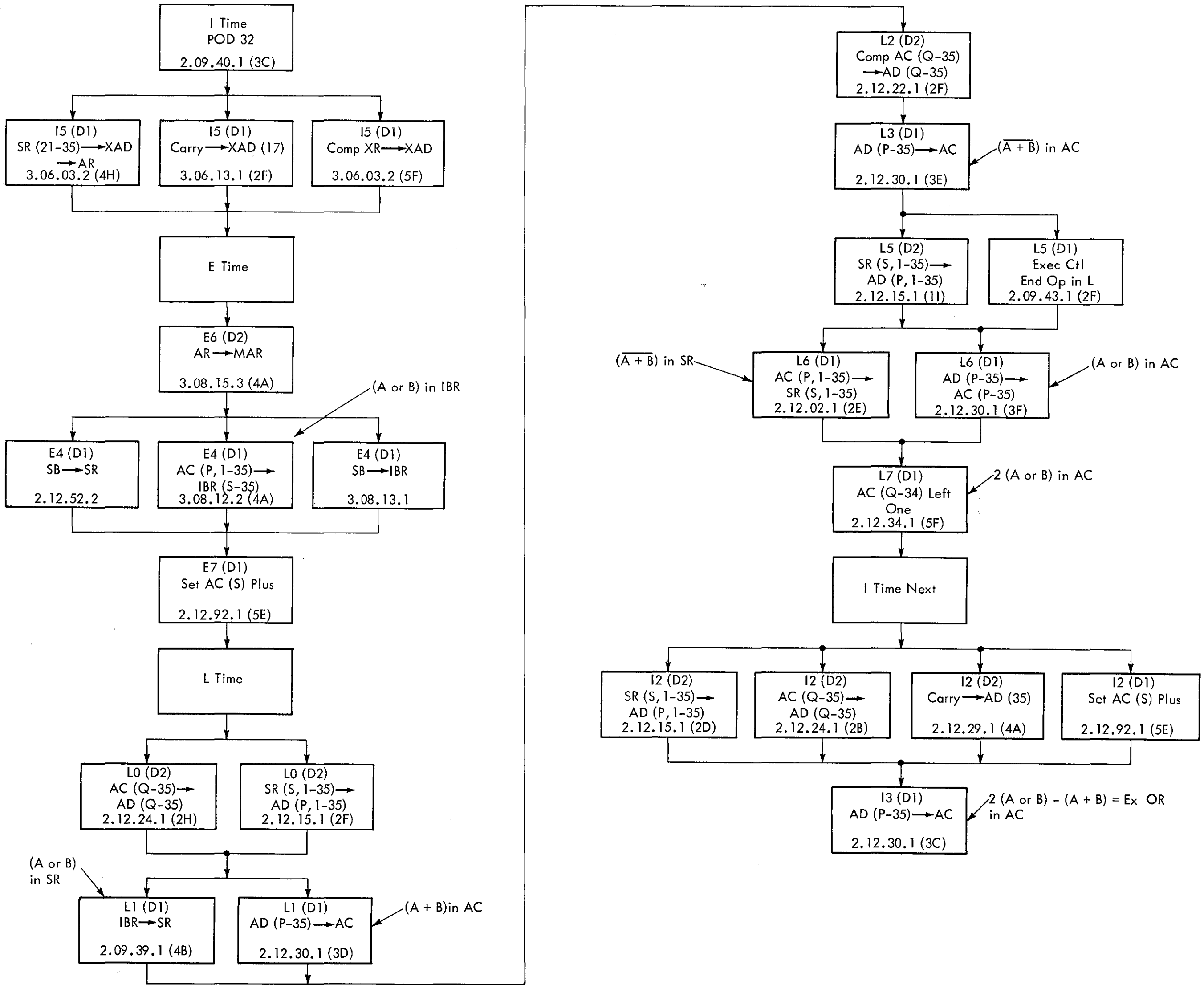


Figure 38. Exclusive OR to Accumulator (ERA)

Skip instructions allow the programmer to branch into subroutines under certain conditions without stopping the computer. These instructions, if skip conditions are met, cause one or two instructions to be skipped. Most skip instructions cause the computer to skip when the condition being tested is met. The exceptions are the error testing and I/O testing instructions, which cause skipping when the condition being tested is not met. All skip instructions except CAS and LAS can cause a skip of one. CAS and LAS cause skips of two, one, or none, depending on conditions.

The skip addresses are formed during E or L3 and 4 time as shown on Systems 02.09.41.1 and 03.06.03.1 (CAS, LAS Flow Diagram, Sheet 1). Except for certain inhibiting conditions, the skip one address is formed by gating PC to XAD to AR with a carry at E or L3. If the overlap trigger is on, the previous address would be the skip two address. If the overlap trigger is not on, the skip two address is formed by gating the AR to XAD to AR with a carry at E or L4. If skip conditions are met, the skip trigger and the skip one or skip two triggers are turned on. Turn on of the skip one or skip two trigger is dependent on the number of skips to be taken.

Storage Zero Test ZET + 0520 (I, E)

If the contents of storage location X (1-35) are zero, the computer skips the next instruction. If the contents are not zero, the computer takes the next sequential instruction. The contents of location X are not changed. The storage bus is tested for zero. See Figure 39.

Storage Non-Zero Test NZT − 0520 (I, E)

If the contents of storage location X (1-35) are not zero, the computer skips the next instruction. If the contents are zero, the computer takes the next sequential instruction. The contents of location X are not changed. See Figure 39.

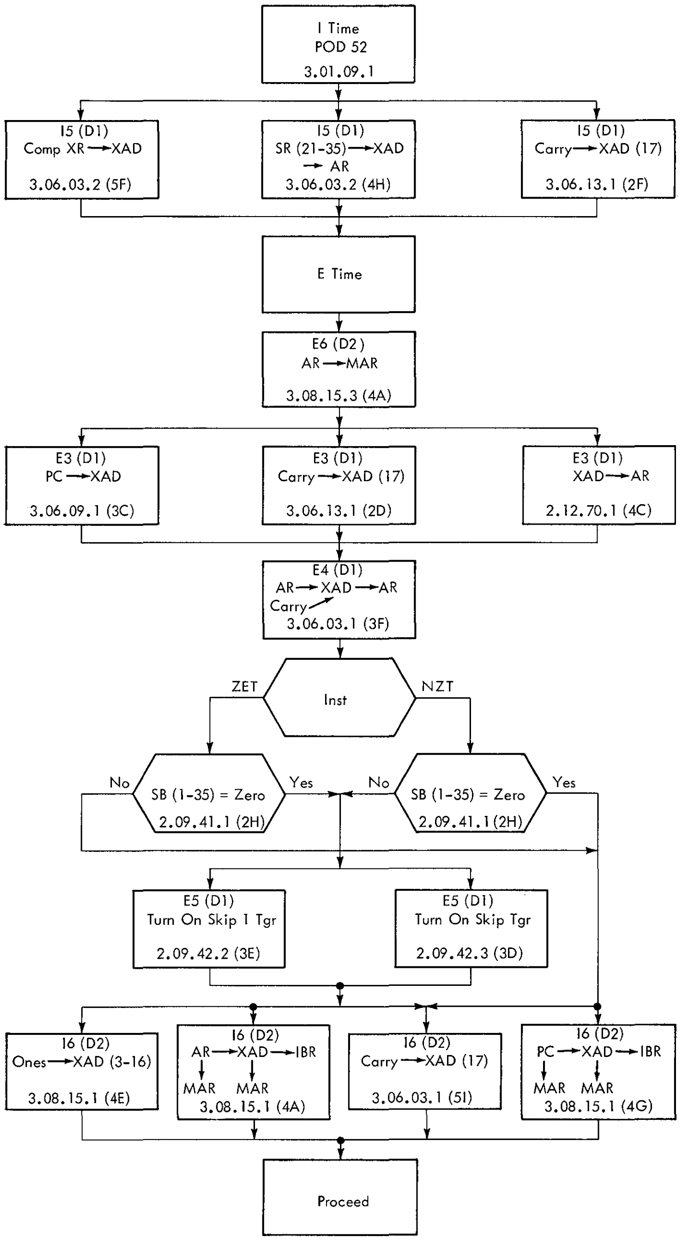


Figure 39. Storage Zero Test (ZET); Storage Non-Zero Test (NZT)

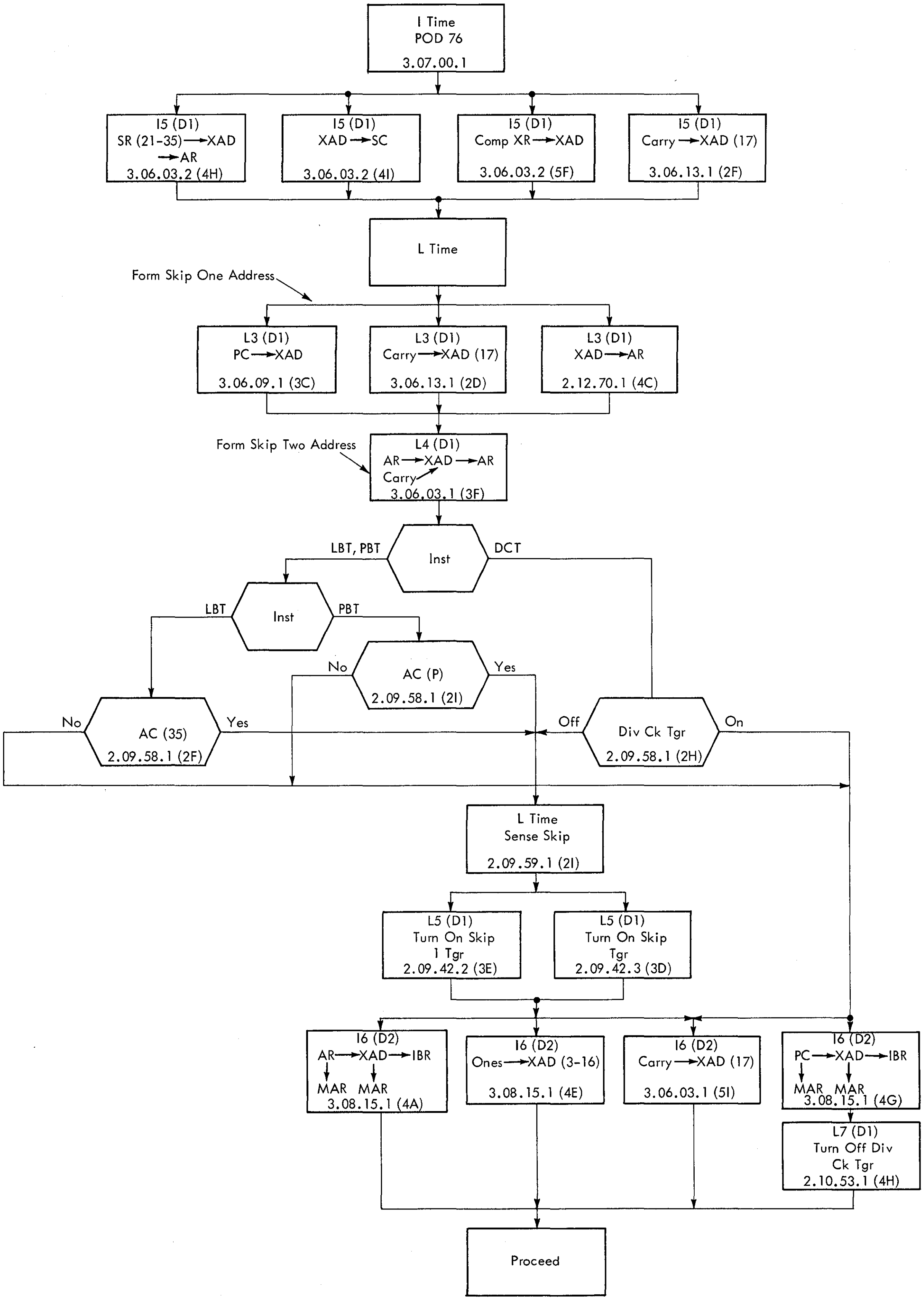


Figure 40. Low-Order Bit Test (LBT); P Bit Test (PBT); Divide Check Test (DCT)

Low-Order Bit Test LBT+0760...0001 (I, L)

If accumulator position 35 contains a one, the computer skips the next instruction. If 35 is a zero, the computer takes the next instruction in sequence. Address modification may result in changing the instruction. See Figure 40.

P Bit Test PBT−0760...0001 (I, L)

If accumulator position P contains a one, the computer skips the next instruction. If P is a zero, the computer takes the next instruction in sequence. Address modification may result in changing the instruction. See Figure 40.

Divide Check Test DCT+0760...0012 (I, L)

If the divide check trigger is on, it is turned off and the computer takes the next sequential instruction. If the divide check trigger is off, the computer skips the next instruction. Address modification may result in changing the instruction. See Figure 40.

Sense Switch Test SWT+0760... 161-166 (I, L)

If the specified sense switch is on (down), the computer skips the next instruction. If the sense switch is off (up), the computer takes the next sequential instruction. Address 161-166 selects sense switch 1-6 respectively. Address modification may change this instruction. See Figure 41.

Compare Accumulator with Storage CAS+0340 (I E,)

The accumulator (Q-35) is compared with the word at storage location X (1-35). If the contents of the accumulator are algebraically greater than the contents of location X, the computer takes the next sequential in-

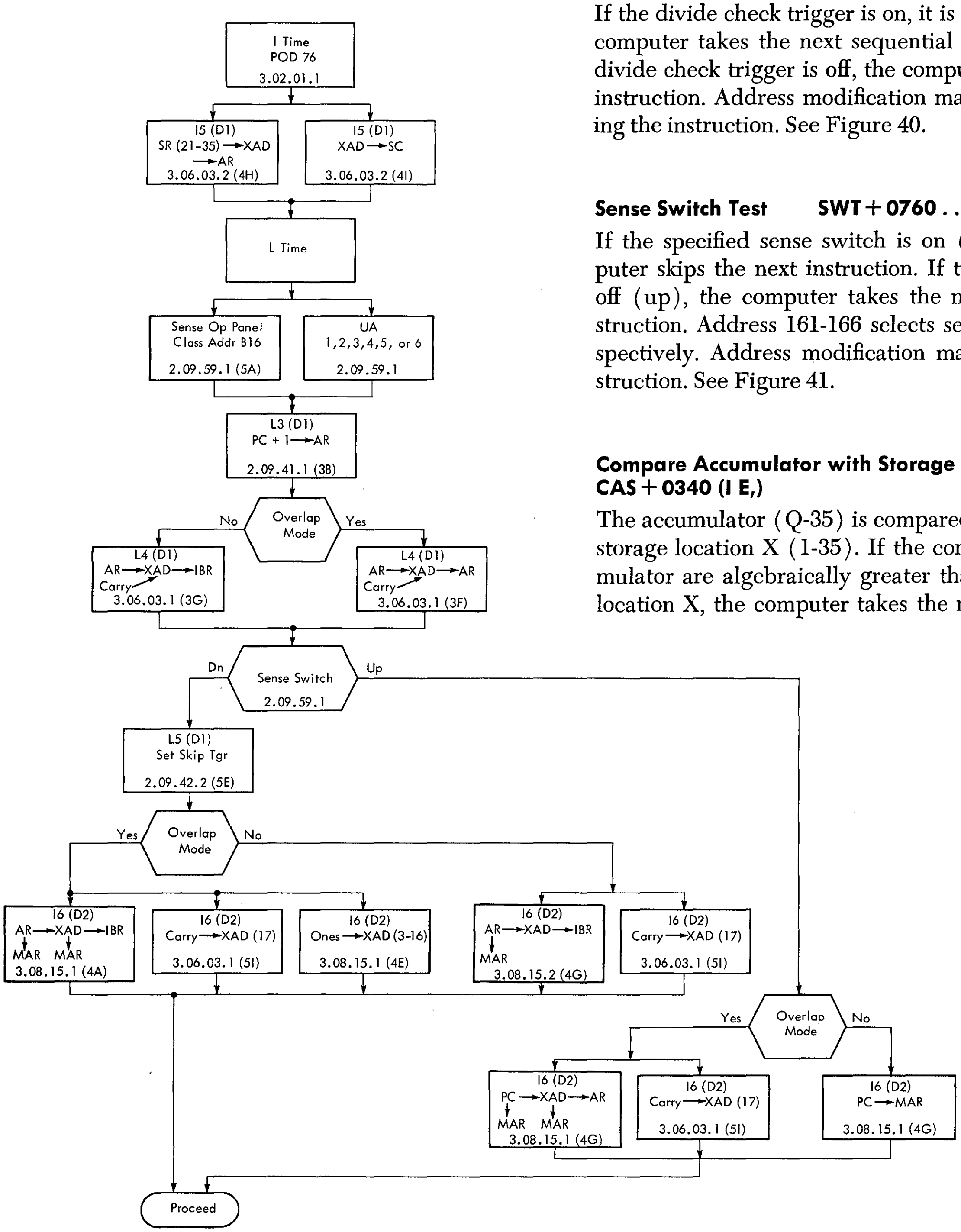


Figure 41. Sense Switch Test (SWT)

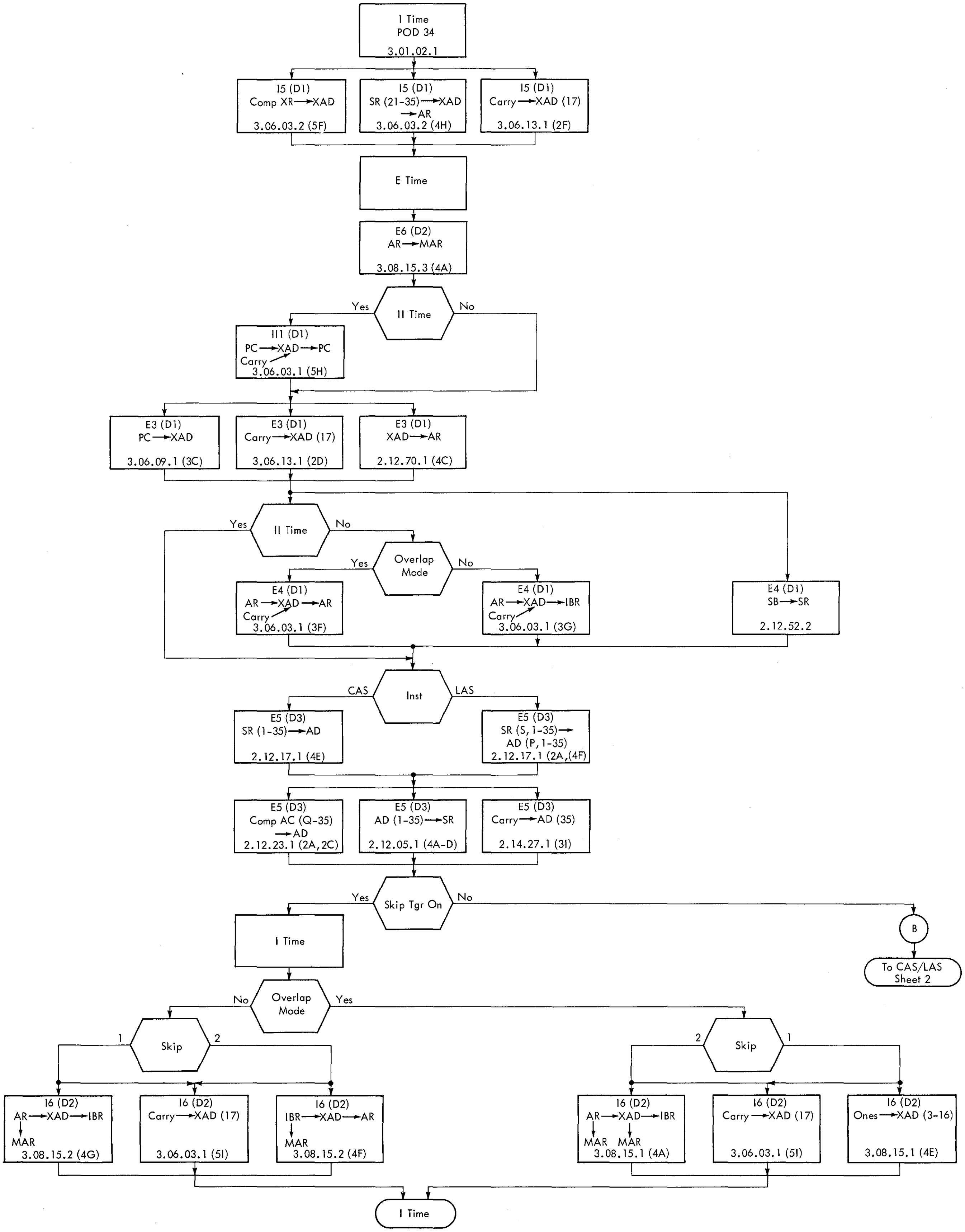


Figure 42, Sheet 1 of 2. Compare Accumulator with Storage (CAS); Logical Compare Accumulator with Storage (LAS) struction. If the contents of the accumulator are algebraically equal to the contents of location X, the computer skips the next instruction and proceeds. If the contents of the accumulator are algebraically less than the contents of location X, the computer skips the next two instructions and proceeds. See Figure 42.

At E4 of this instruction, the word at storage location X enters the storage register. The storage register (1-35) and the complement of the accumulator (Q-35) are gated to the adders at E5. The resulting conditions are checked (Systems 02.09.42.2). CAS and LAS skip conditions are:

INSTR	SR(S)	AC(S)	Q CARRY	AD SUM	SKIP	SYSTEMS
CAS	+	−			2	02.09.42.2 (2G)
CAS	+	+	Yes	Not Zero	2	02.09.42.2 (IE)
CAS	±	−	No		2	02.09.42.2 (2F)
CAS	±	±	Yes	Zero	1	02.09.42.2 (3B)
LAS			Yes	Not Zero	2	02.09.42.2 (2H)
LAS			Yes	Zero	1	02.09.42.2 (3C)

Logical Compare Accumulator with Storage LAS — 0340 (I, E)

The accumulator (Q-35) is compared with the word at storage location X (S-35). Signs are not considered. If the contents of the accumulator are greater than the contents of location X, the computer takes the next sequential instruction. If the contents of the accumulator are equal to the contents of location X, the computer skips the next instruction and proceeds. If the contents of the accumulator are less than the contents of location X, the computer skips the next two instructions and proceeds.

At E4 of this instruction, the word at storage location X enters the storage register. The storage register (S-35) and the complement of the accumulator (Q-35) are gated to the adders at E5. The resulting conditions are checked (Systems 02.09.42.2). Refer to CAS for the skip chart. See Figure 42.

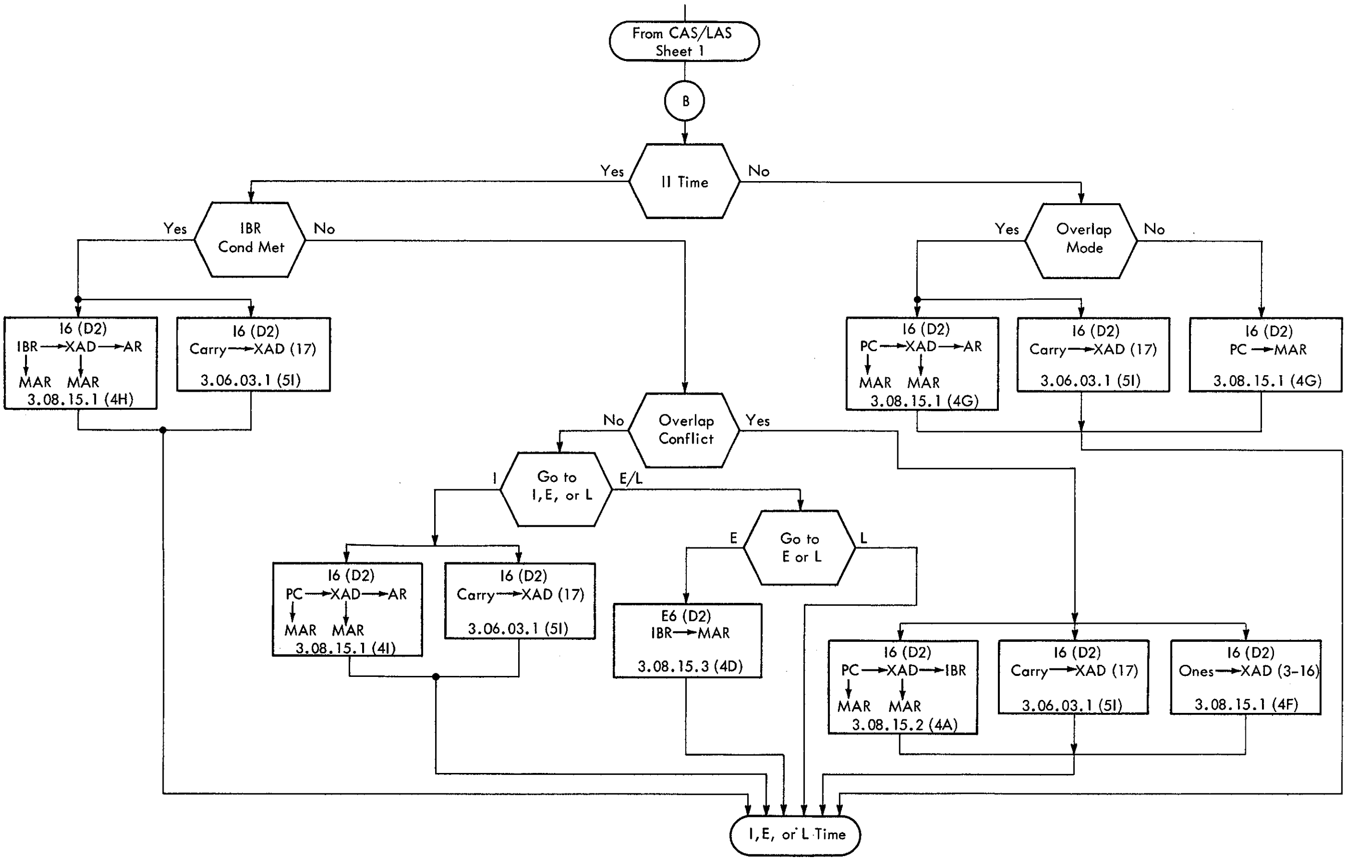


Figure 42, Sheet 2 of 2. Compare Accumulator with Storage (CAS); Logical Compare Accumulator with Storage (LAS)

Convert Instructions

The three convert instructions can materially reduce the time required for many "housekeeping" and table-look-up routines. They can be used for number conversions, for preparing print fields, and even for adding numbers in systems other than binary.

The convert instructions, like variable-length instructions, include a count field as well as the operation code, address and tag bit. The convert instructions cause a series of references (usually six) and the address specifies the starting location of the first storage table. The register (accumulator or MQ) from which the reference is controlled is considered to be made up of six 6-bit groups. The first of these groups is added to the instruction address to give the location of the first storage reference. The word stored at this location must contain, in addition to its conversion information, the starting location of the next storage table. The convert-by-replacement instructions shift the controlling register six places, clearing the six places on the opposite end of the register. Positions S-5 (this text uses S-5 to represent S, 1-5) of the table word are entered into the six cleared positions, and the next group of six bits is in position to add to the starting location of the next table. The process continues until up to six references have been made. The controlling register is gradually replaced by six-bit entries from the storage tables. When the number of references specified by the count has been made, the conversion is complete.

The instruction tag has a unique function for the convert instructions. Positions 18 and 19 are not used, but a bit in 20 will cause the storage table starting location contained in the last reference word to be stored in index register A.

Convert by Replacement from Accumulator
CVR + 0114 (Min I, L), (Max I, L, 6E)

This instruction treats the contents of the AC (P, 1-35) as six 6-bit representations. The instruction replaces a number of these representations, equal to the count, with positions S-5 of a like number of words from storage. These words are found by adding AC (30-35) (the first six-bit group) to SR (21-35) (initially the instruction address) and directing storage to this modified address. The word thus found is brought to the SR; the AC is shifted right six places; SR (S-5) replaces AC (P, 1-5). SR (21-35) adds to the next six-bit group in AC (30-35) to locate the next word in storage. The process is then repeated. After the required number of replacements, the address portion of the last storage word can be stored in index register A by including a "tag" bit in position 20 of the instruction. See Figure 43.

The following example illustrates the use and operation of CVR.

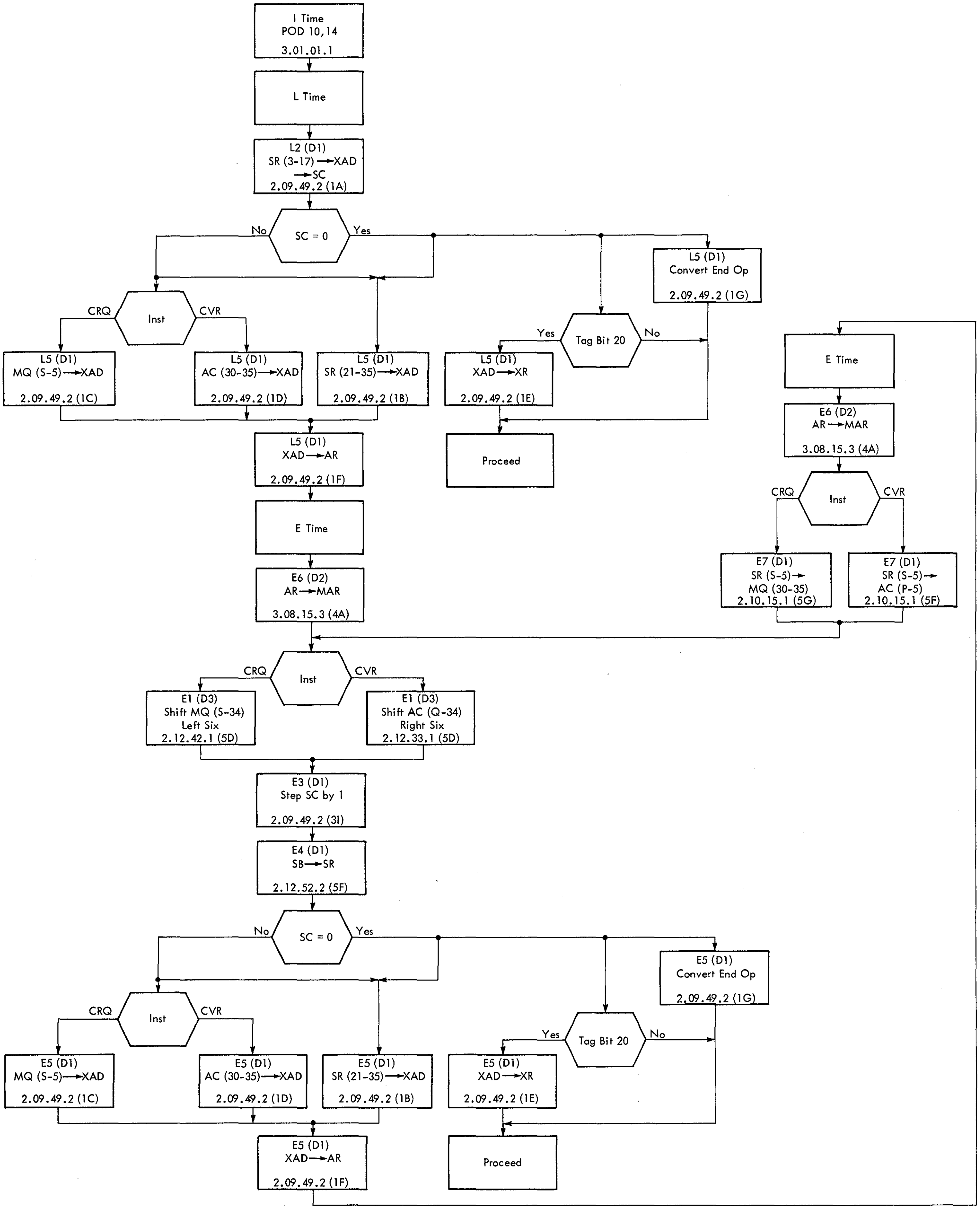


Figure 43. Convert by Replacement from Accumulator (CVR); Convert by Replacement from MQ (CRQ)

Direct Addition of BCD Numbers:

A	+	B	=	C
134589	+	691593	=	826182

Table required in storage for this example:

STORAGE LOCATION	CONTENTS (S-5)	CONTENTS (21-35)
1000	0	1000
1001	1	1000
1002	2	1000
1003	3	1000
1004	4	1000
1005	5	1000
1006	6	1000
1007	7	1000
1008	8	1000
1009	9	1000

STORAGE LOCATION	CONTENTS (S-5)	CONTENTS (21-35)
1010	0	1001
1011	1	1001
1012	2	1001
1013	3	1001
1014	4	1001
1015	5	1001
1016	6	1001
1017	7	1001
1018	8	1001
1019	9	1001

Instructions required for operation:

CAL A	(First BCD word)
ADD B	(Second BCD word)
CVR 6, 0, 1000	(Convert sum to BCD word)

Development of Program:

					CONTENTS OF ACCUMULATOR						
					P1-5	6-11	12-17	-23	-29	-35	
CAL A(Clear AC, add logical word A) ——→					1	3	4	5	8	9	1st
Add B					+6	9	1	5	9	3	Char
(Unconverted binary sum)					=7	12	5	10	17	12	Conv
CVR 1000, 0, 6											ARS 6
Series of Steps within CVR											
C(AC) 30-35	Start Loc	Table Refer	Table C(—) 1-5	Next Start Loc C(—) 21-35							
12	+1000	−1012	2	1001	2	7	12	5	10	17	Count = 5
17	+1001	−1018	8	1001 →	8	2	7	12	5	10	= 4
10	+1001	−1011	1	1001 →	1	8	2	7	12	5	= 3
5	+1001	−1006	6	1000 →	6	1	8	2	7	12	= 2
12	+1000	−1012	2	1001 →	2	6	1	8	2	7	= 1
7	+1001	−1008	8	1000 →	8	2	6	1	8	2	= 0

Count = zero: if Tag = 1, 1000 XRA

SLW C (Store converted BCD sum as logical word)

The execution of CVR requires one L cycle to set the count in the shift counter and to calculate the initial table location. The instruction is completed in as many E cycles as specified by the count.

Convert by Replacement from MQ
CRQ — 0154 (Min I, L), (Max I, L, 6E)

This instruction operates on the MQ, considered to be composed of six 6-bit representations. The instruction replaces a number of these representations equal to its count with positions (S-5) of a like number of words from storage. The location of the first of these storage words is found by adding the contents of MQ (S-5) to SR (21-35) (initially the instruction address portion). The word stored at this modified location is brought to the SR, and the MQ is shifted left six places. Positions (S-5) of the stored word are placed in MQ (30-35), and the location of the next storage word is computed by again adding MQ (S-5) to SR (21-35) which is now the address portion of the previous storage word. The process continues until the required number of replacements have been made. At this time the presence of a tag bit in position 20 of the instruction will cause the address portion of the final storage word to be stored in index register A. See Figure 43.

The following example illustrates the use and operation of CRQ.

Prepare a BCD number for printing by replacing leading zeros with blanks:
Convert BCD number 007109 to BL BL 7109
Instructions required for this operation:

LDQ A	(BCD number in storage)
CRQ 6, 0, 2000	(Convert from MQ)
STQ B	(Store converted number)

Storage table required for CRQ (in decimal):

STORAGE	CONTENTS	
LOCATION	(S-5)	(21-35)
2000	BL	2000
2001	1	2010
2002	2	2010
2003	3	2010
2004	4	2010
2005	5	2010
2006	6	2010
2007	7	2010
2008	8	2010
2009	9	2010

STORAGE	CONTENTS	
LOCATION	(S-5)	(21-35)
2010	0	2010
2011	1	2010
2012	2	2010
2013	3	2010
2014	4	2010
2015	5	2010
2016	6	2010
2017	7	2010
2018	8	2010
2019	9	2010

Development of Program:

LDQ A

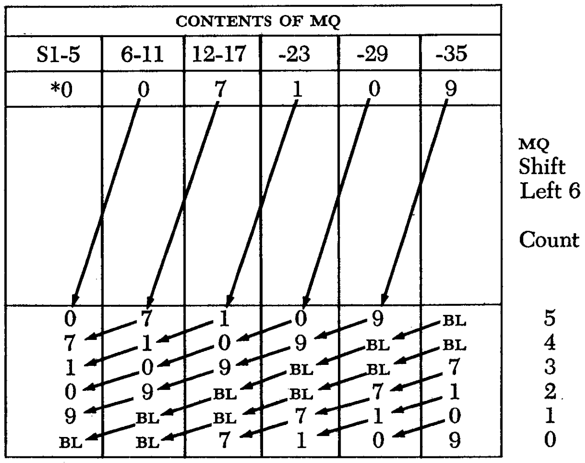
CRQ 2000, 0, 6

1st Char Conv

Series of Steps within CRQ

C(MQ)s 1-5	Start Loc	Table Refer	Table C(—)s 1-5	Next Starting Loc C(—) 21-35
*0	+2000 =	2000	BL	2000
0	+2000 =	2000	BL	2000
7	+2000 =	2007	7	2010
1	+2010 =	2011	1	2010
0	+2010 =	2010	0	2010
9	+2010 =	2019	9	2010

Count = 0; if Tag = 1, 2010 → XRA



CONTENTS OF MQ						
S1-5	6-11	12-17	-23	-29	-35	Count
*0	0	7	1	0	9	
0	7	1	0	9	BL	5
7	1	0	9	BL	BL	4
1	0	9	BL	BL	7	3
0	9	BL	BL	7	1	2
9	BL	BL	7	1	0	1
BL	BL	7	1	0	9	0

The execution of CRQ is accomplished in one L cycle and a number of E cycles equal to the count.

Convert by Addition from MQ CAQ — 0114 (Min I, L), (Max I, L, 6E)

This instruction treats the MQ as six 6-bit representations, as does CRQ. This instruction does not replace any of these representations, but uses them to locate a number of storage words equal to the count of the instruction. The storage words are located at the address developed by adding MQ (S-5) to SR (21-35) (initially the instruction address) and are brought to the SR. The storage words are then added to the contents of the accumulator. The MQ is not replaced, but is merely rotated left six places to allow the next six-bit representation to be added to the address portion of the previous storage word. The process continues until a number of additions equal to the count have been made to the AC. The operation is then complete. The address portion of the last storage word used can be stored in index registers for future reference by adding a tag bit in position 20 of the instruction. See Figure 44.

To illustrate the operation and use of CAQ, a program which will convert BCD to binary follows.

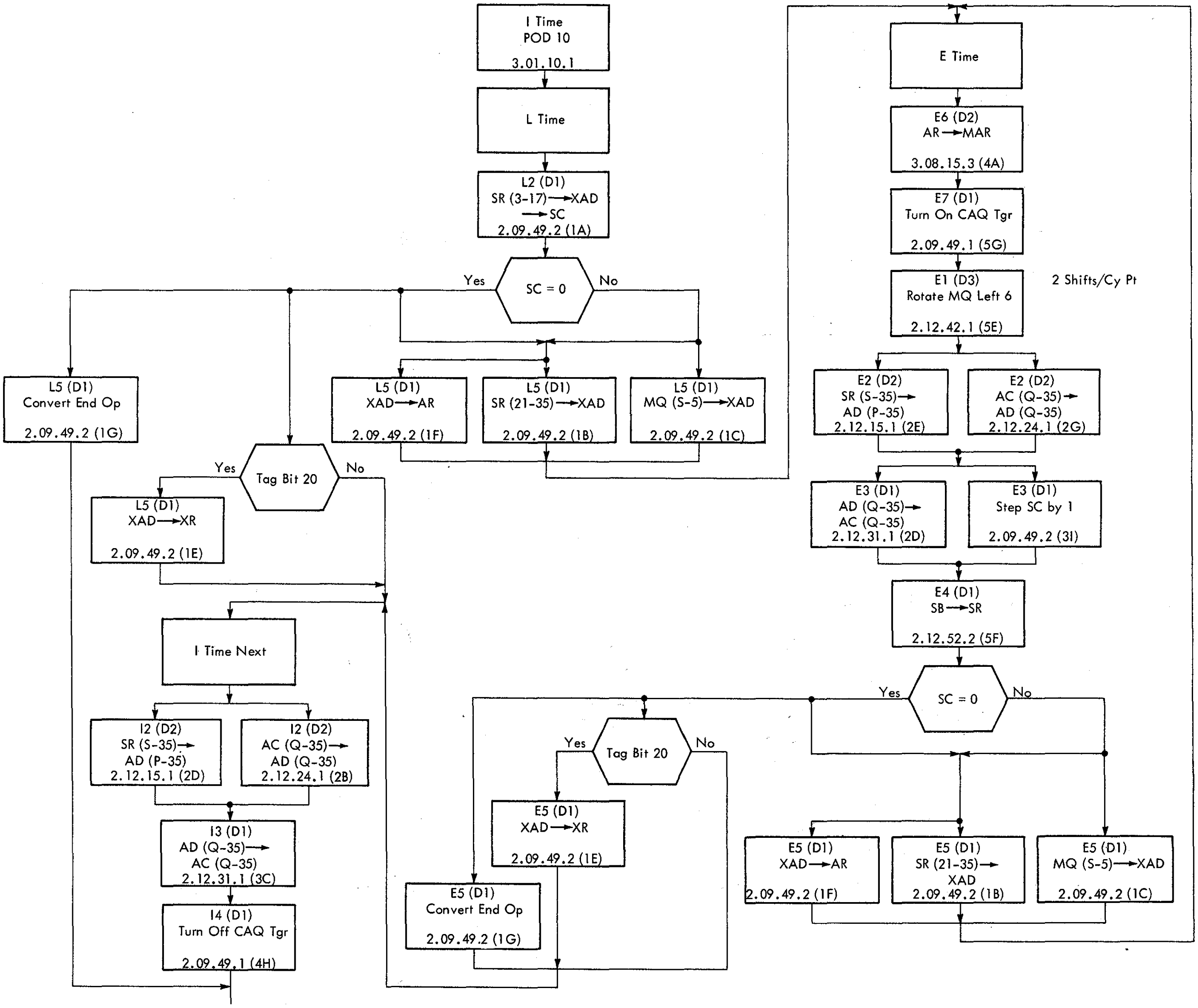


Figure 44. Convert by Addition from MQ (CAQ)

Convert BCD Word to Binary
709542 (BCD) converted to 2,551,646 (Octal)
Instructions required for this operation:

```
LDQ          A      (BCD word)
CLM                 (Clear AC)              ARS 16₁₀      (Position result in AC)
CAQ 6, 0, 3000      (Convert to binary)     SLW      B    (Store result)
```

Storage table required for CAQ:

STORAGE	CONTENTS		STORAGE	CONTENTS	
LOCATION	(S-19)	(21-35)	LOCATION	(S-19)	(21-35)
DECIMAL	OCTAL	DECIMAL	DECIMAL	OCTAL	DECIMAL
3000	0	3100	3300	0	3400
3001	303,240	3100	3301	144	3400
3002	606,500	3100	3302	310	3400
3003	1,111,740	3100	3303	454	3400
3004	1,415,200	3100	3304	620	3400
3005	1,720,440	3100	3305	764	3400
3006	2,223,700	3100	3306	1,130	3400
3007	2,527,140	3100	3307	1,274	3400
3008	3,032,400	3100	3308	1,440	3400
3009	3,335,640	3100	3309	1,604	3400
3100	0	3200	3400	0	3500
3101	23,420	3200	3401	12	3500
3102	47,040	3200	3402	24	3500
3103	72,460	3200	3403	36	3500
3104	116,100	3200	3404	50	3500
3105	141,520	3200	3405	62	3500
3106	165,140	3200	3406	74	3500
3107	210,560	3200	3407	106	3500
3108	234,200	3200	3408	120	3500
3109	257,620	3200	3409	132	3500
3200	0	3300	3500	0	
3201	1,750	3300	3501	1	
3202	3,720	3300	3502	2	
3203	5,670	3300	3503	3	
3204	7,640	3300	3504	4	
3205	11,610	3300	3505	5	
3206	13,560	3300	3506	6	
3207	15,530	3300	3507	7	
3208	17,500	3300	3508	10	
3209	21,450	3300	3509	11	

Development of Program

						Binary Equivalent			BCD Word					
						CONTENTS OF ACCUMULATOR			CONTENTS OF MQ					
						P, 1-19	21-35		S1-5	6-11	-17	-23	-29	-35
LDQ BCD WORD						XX	XX		*7	0	9	5	4	2
CLM						00	00							
CAQ 3000, 0, 6								RQL 6						
Series of steps within CAQ														
C(MQ) S, 1-5	Start Loc	Table Refer	Table C(—)S, 1-19	Next Start Loc C(—) 21-35				Count						
7	+3000 =	3007	2,527,140	3100	+	2,527,140	3100	5	0	9	5	4	2	7
0	+3100 =	3100	0	3200	+	0	3200	4	9	5	4	2	7	0
					=	2,527,140	6300							
9	+3200 =	3209	21,450	3300	+	21,450	3300	3	5	4	2	7	0	9
					=	2,550,610	9600							
5	+3300 =	3305	764	3400	+	764	3400	2	4	2	7	0	9	5
					=	2,551,574	13000							
4	+3400 =	3404	50	3500	+	50	3500	1	2	7	0	9	5	4
					=	2,551,644	16500							
2	+3500 =	3502	2	—	+	2	—	0	7	0	9	5	4	2
					=	2,551,646	16500							
ARS 16 (Binary Equivalent) →						2,551,646								
SLW														

Control Instructions

Control instructions enable the programmer to change problem conditions and program flow, alter card machine output, and automatically enter information or commands from the operator's panel.

Sense Light Test SLT — 0760 . . . 0141-0144 (I, L)

If the sense light corresponding to the address portion of the instruction is on, the light is turned off and the computer skips the next instruction. If the sense light is off, the computer takes the next sequential instruction. See Figure 45.

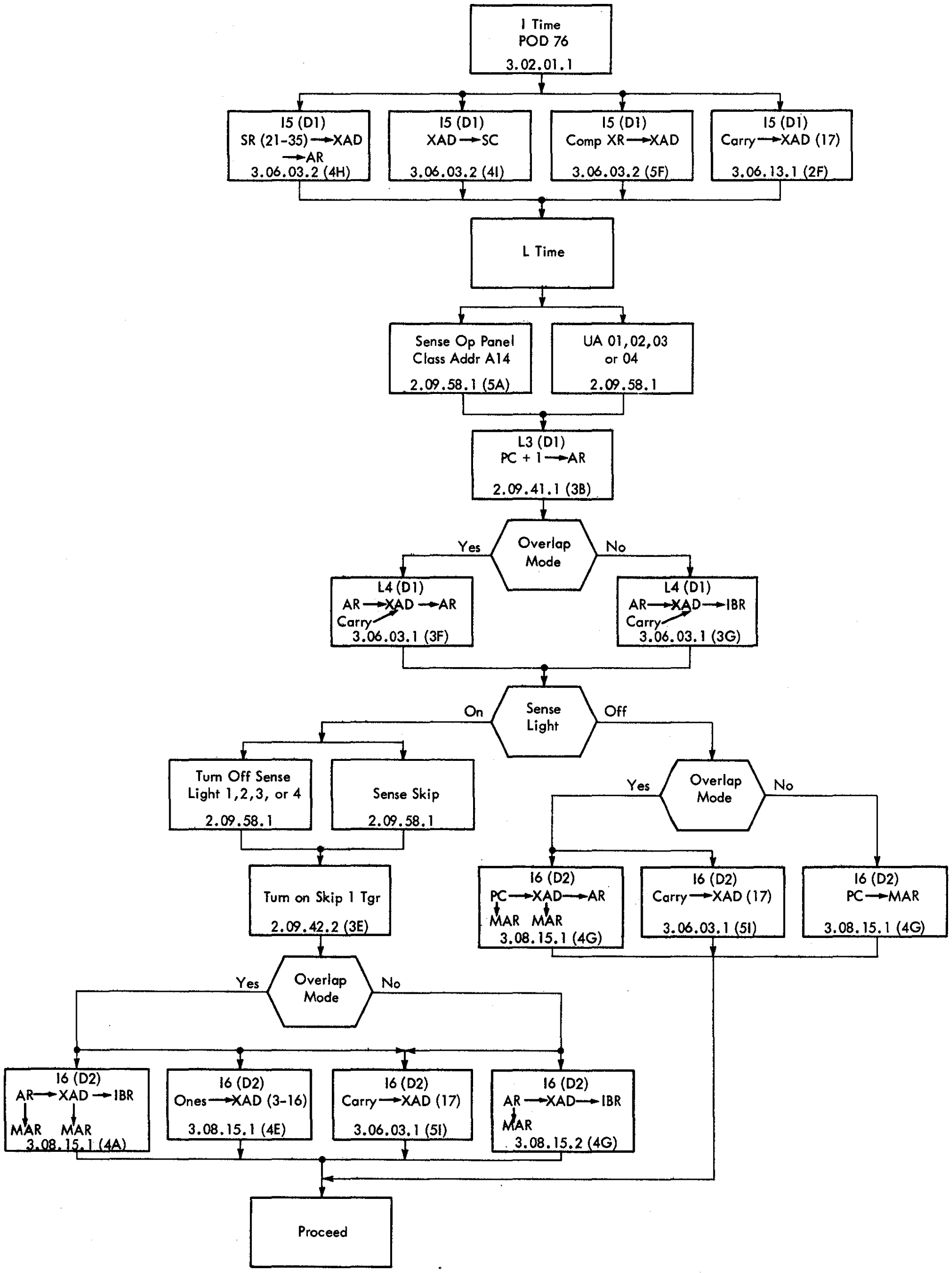


Figure 45. Sense Light Test (SLT)

Input/Output Check Test
IOT + 0760 . . . 0005 (I, L)

If the I/O check trigger is on, the trigger is turned off and the computer takes the next instruction in sequence. If the I/O check trigger is off, the computer skips the next instruction. See Figure 46.

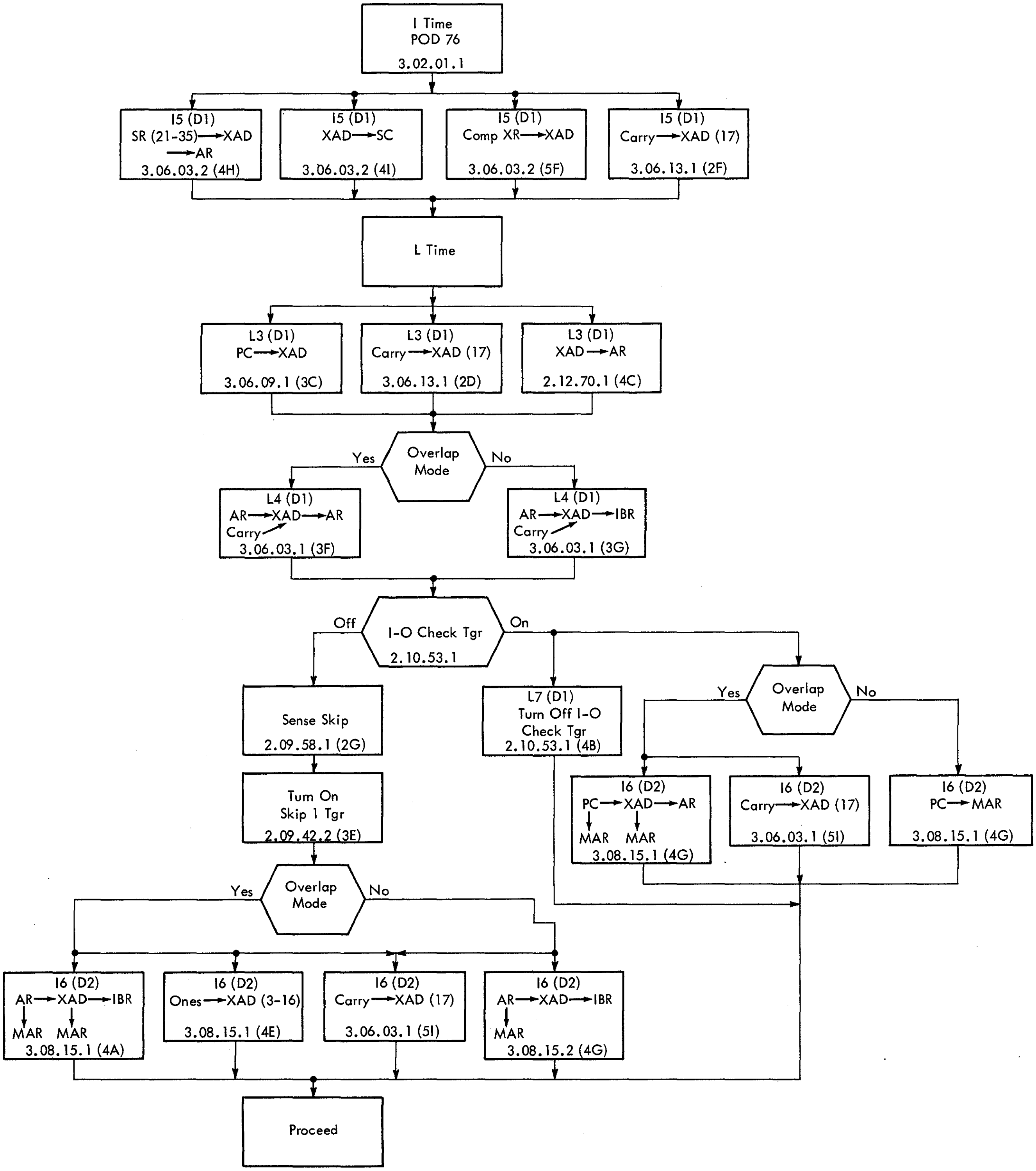


Figure 46. Input-Output Check Test (IOT)

Enter Overlap Mode and Transfer ELAP + 0047 (I)

If the overlap switch is up (off), this instruction turns on the enter overlap trigger, placing the computer in overlap mode, and the computer transfers to location X. If the overlap switch is down (on), this instruction acts like a NOP. With the switch down the computer is in overlap mode. See the section, "Overlap." See Figure 47.

Leave Overlap Mode and Transfer LLAP — 0047 (I)

If the overlap switch is up (off), this instruction turns off the enter overlap trigger, taking the computer out of overlap mode, and transfers to location X. If the overlap switch is down (on), this instruction acts like a NOP. With the switch down, the computer is in overlap mode. See the section, "Overlap." See Figure 47.

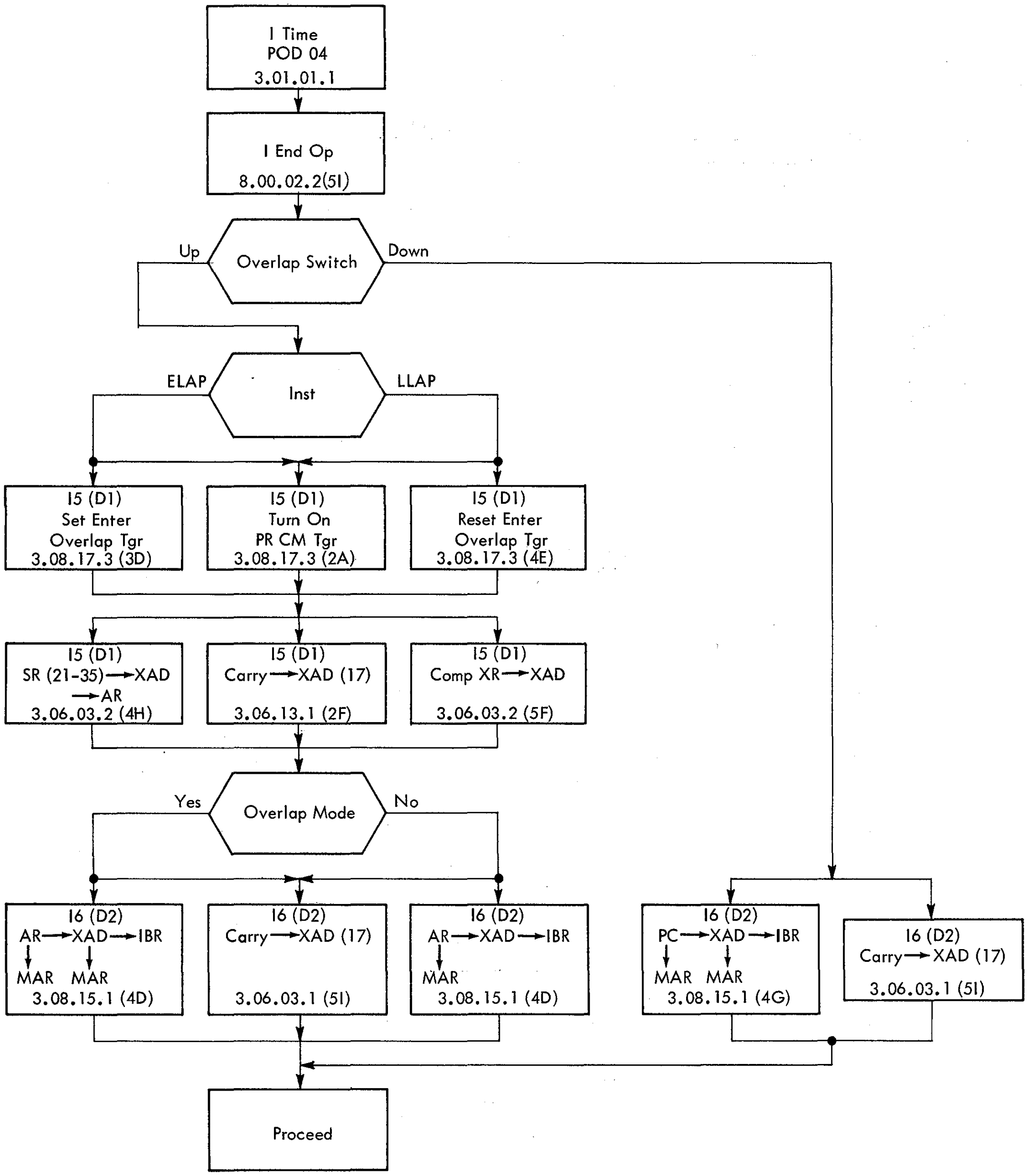


Figure 47. Enter Overlap Mode and Transfer (ELAP); Leave Overlap Mode and Transfer (LLAP)

Enter Diagnostic Mode and Transfer EDAT + 0045 (I)

If the diagnostic mode switch is up (on), this instruction will turn on the diagnostic mode trigger, placing the computer in diagnostic mode, and the computer transfers to location X. If the diagnostic mode switch is down (off), this instruction acts like a NOP. Throwing the switch down takes the computer out of diagnostic mode.

When the computer is in diagnostic mode, address position three instead of 17 determines which memory is addressed. Diagnostic mode also blocks XAD to MAR gating. Consequently, only one instruction is fetched from memory during an I cycle. The IBR loaded trigger is not turned on, and the II cycle following an I cycle must also fetch its instruction from memory. See Figure 48.

Leave Diagnostic Mode and Transfer LDAT − 0045 (I)

If the diagnostic mode switch is up (on), this instruction will turn off the diagnostic mode trigger, taking the computer out of diagnostic mode, and the computer transfers to location X. If the diagnostic mode switch is down (off), this instruction acts like a NOP. Throwing the switch down takes the computer out of diagnostic mode. See Figure 48.

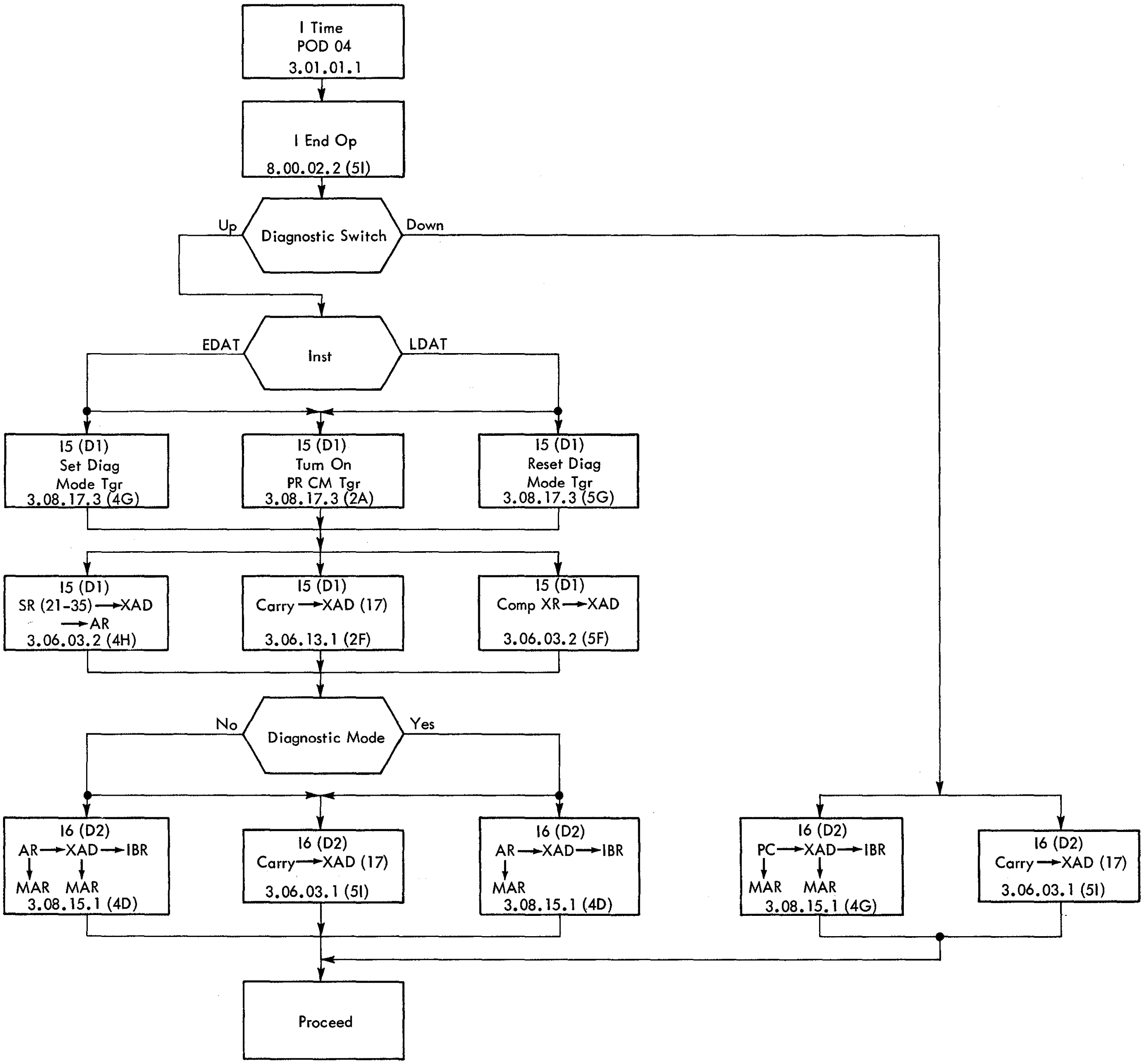


Figure 48. Enter Diagnostic Mode and Transfer (EDAT); Leave Diagnostic Mode and Transfer (LDAT)

Halt and Transfer HTR + 0000 (I, L)

This instruction stops the computer in L time by turning on the master stop trigger at I6. The computer will gate out L time and transfer to location X when the start key is depressed. When halted, the program counter contains the address of the HTR instruction. See Figure 49.

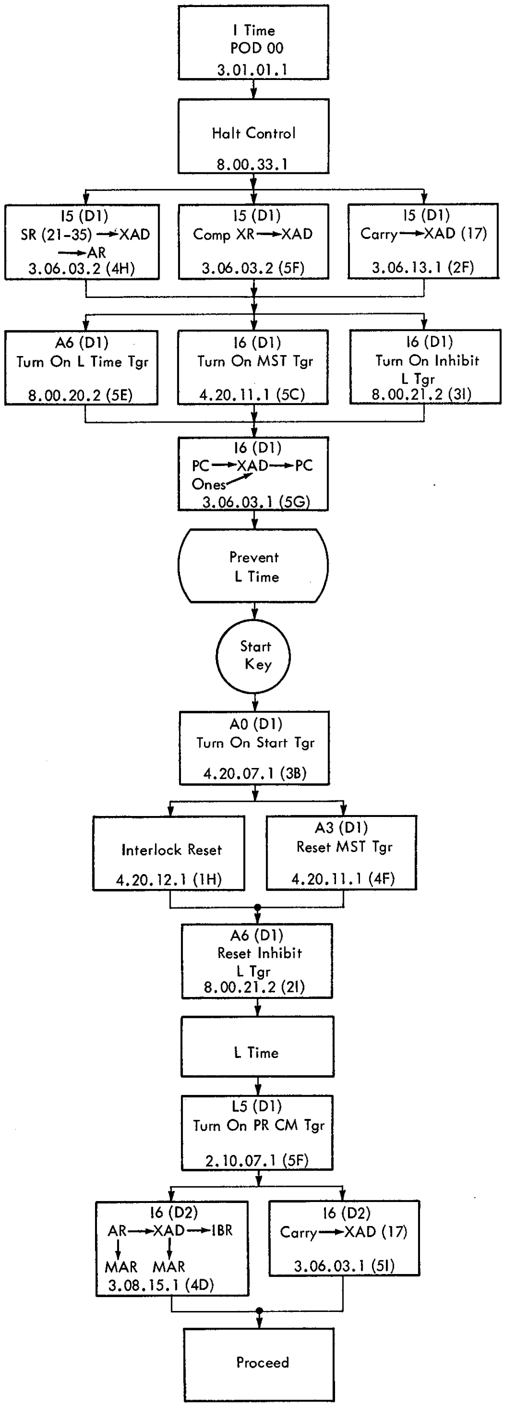


Figure 49. Halt and Transfer (HTR)

Halt and Proceed HPR + 0420 (I, L)

This instruction stops the computer in L time by turning on the master stop trigger at I6. The computer will gate out L time and proceed to the next sequential instruction upon depression of the start key. When halted, the program counter contains the address of the HPR plus one. See Figure 50.

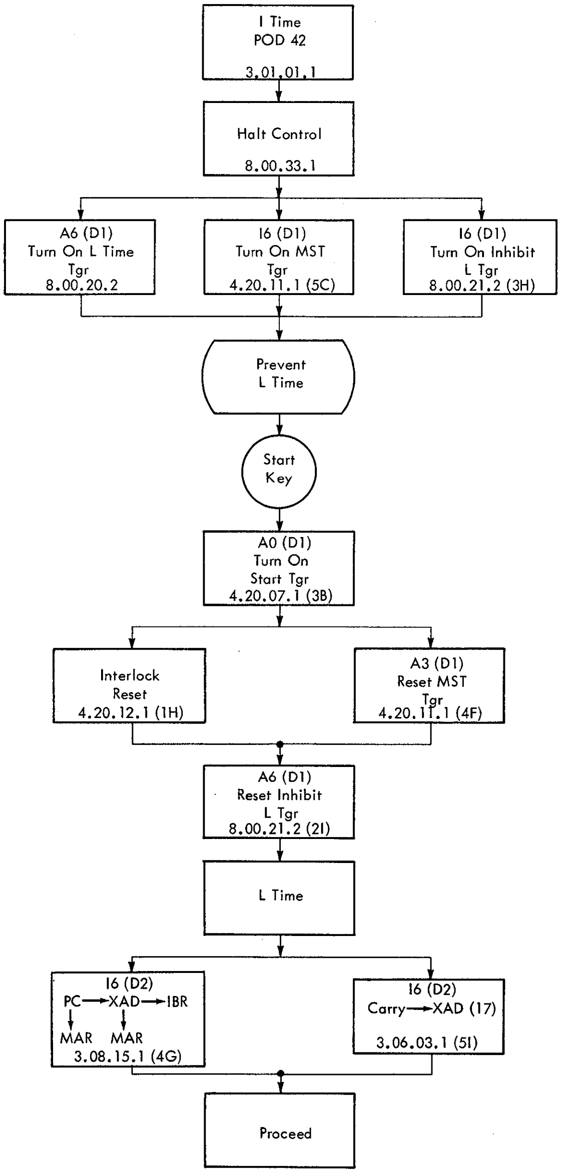


Figure 50. Halt and Proceed (HPR)

Clear Magnitude CLM + 0760 . . . 0000 (I, L)

The contents of the accumulator (Q, P, 1-35) are cleared. Accumulator S is not changed. Address modification may change the instruction. See Figure 51.

Complement Magnitude COM + 0760 . . . 0006 (I, L)

The contents of the accumulator (Q, P, 1-35) are complemented. Accumulator S is not changed. Address modification may change the instruction. See Figure 51.

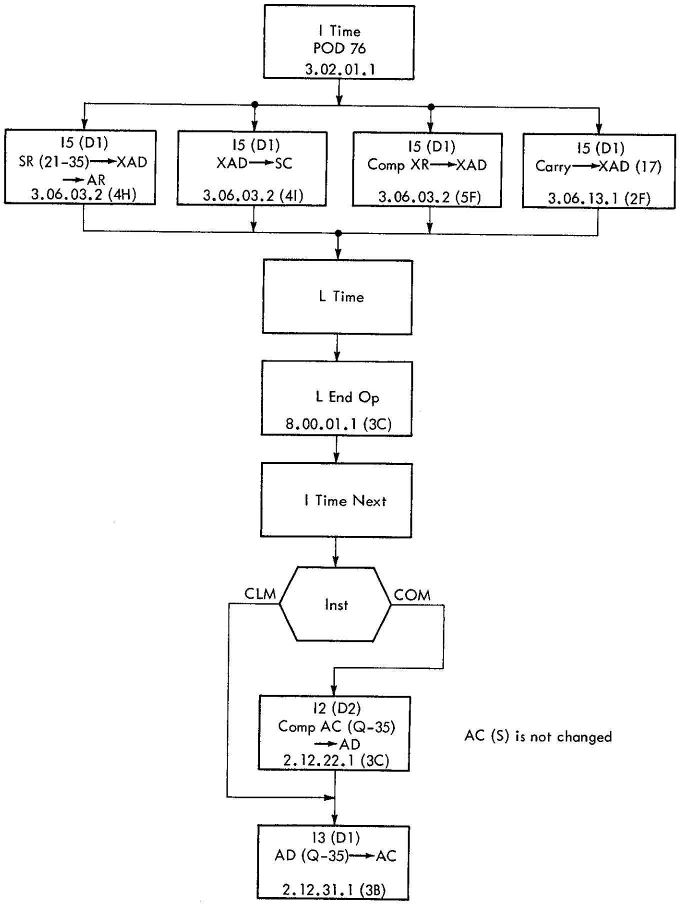


Figure 51. Clear Magnitude (CLM); Complement Magnitude (COM)

Enter Keys ENK + 0760 . . . 0004 (I, L)

The contents of the MQ (S-35) are replaced by the word set up in the console keys (S-35). The keys are gated to the storage register and then to the MQ. See Figure 52.

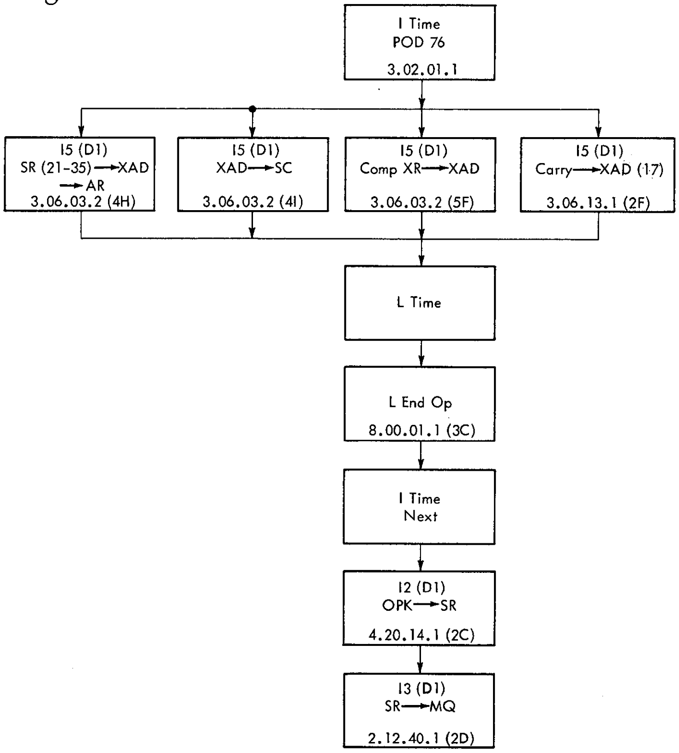


Figure 52. Enter Keys (ENK)

No Operation NOP + 0761 (I, L)

The NOP instruction does nothing but reserve space for other instructions. Because this instruction has a POD of 76, an I and an L cycle are required. The only function of this instruction is to turn on the end operation trigger to allow the computer to proceed. SOD 01, as shown on Systems 8.00.01.1, brings up L END OP.

Execute XEC + 0522 (I)

This instruction causes the computer to perform the instruction at location X. The program counter is not altered. Therefore, after the instruction at location X has been executed, the computer proceeds to the next sequential instruction after XEC. XEC prevents updating the PC as shown on Systems 03.06.03.2 2A. If the instruction at location X is a successful transfer, the computer proceeds to the transfer-to address. If the instruction at location X is a skip type instruction, the skip, if successful, will be relative to the XEC instruction. See Figure 53.

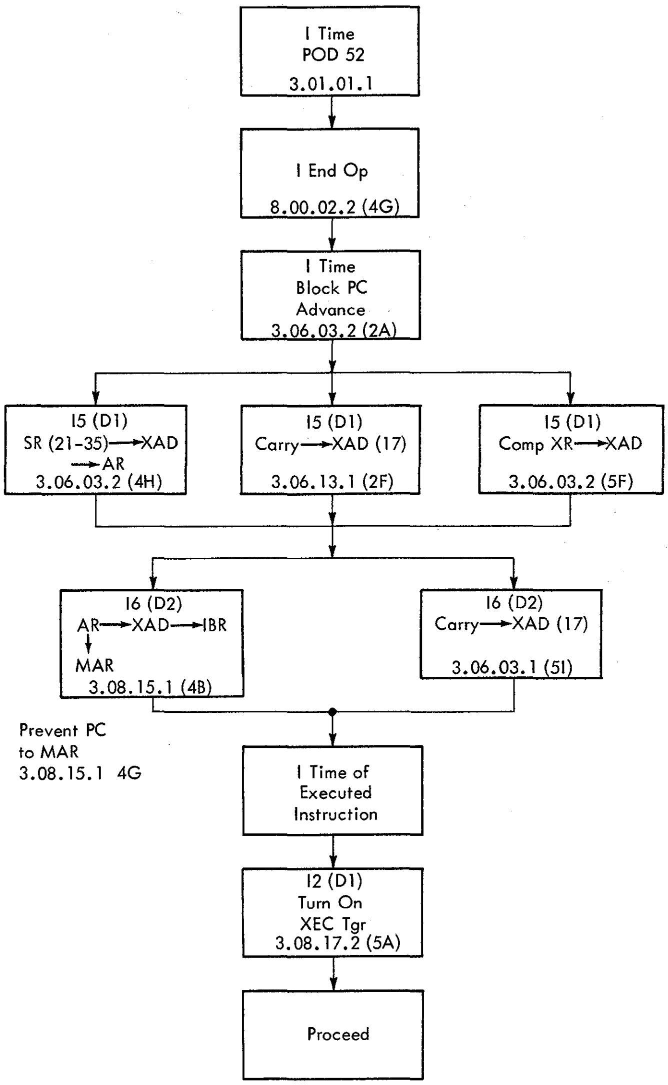


Figure 53. Execute (XEC)

Set Sign Plus SSP + 0760 . . . 0003 (I, L)

This instruction places a zero (a plus) in the accumulator sign position. Accumulator (Q-35) is not changed. See Figure 54.

Set Sign Minus SSM − 0760 . . . 0003 (I, L)

This instruction places a bit (a minus) in the accumulator sign position. Accumulator (Q-35) is not changed. See Figure 54.

Change Sign CHS + 0760 . . . 0002 (I, L)

This instruction complements the sign of the accumulator. A one is changed to a zero and a zero is changed to a one. Accumulator (Q-35) is not changed. See Figure 54.

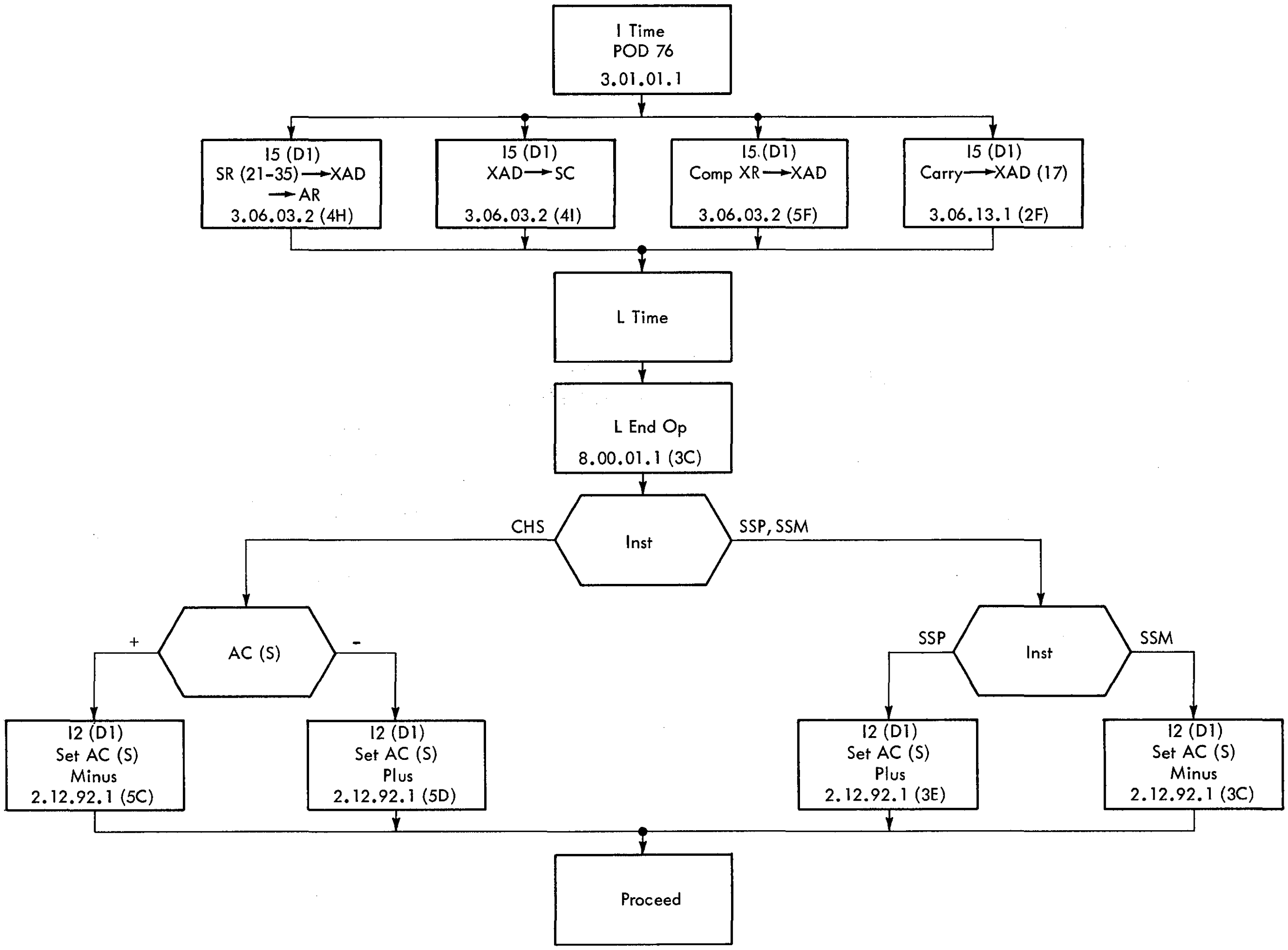


Figure 54. Set Sign Plus (SSP); Set Sign Minus (SSM); Change Sign (CHS)

Sense Lights On SLN + 0760 . . . 141-144 (I, L)

This instruction turns on the specified sense light on the operator's console. Address 141-144 selects sense lights 1-4 respectively. Address modification may change the instruction. See Figure 55.

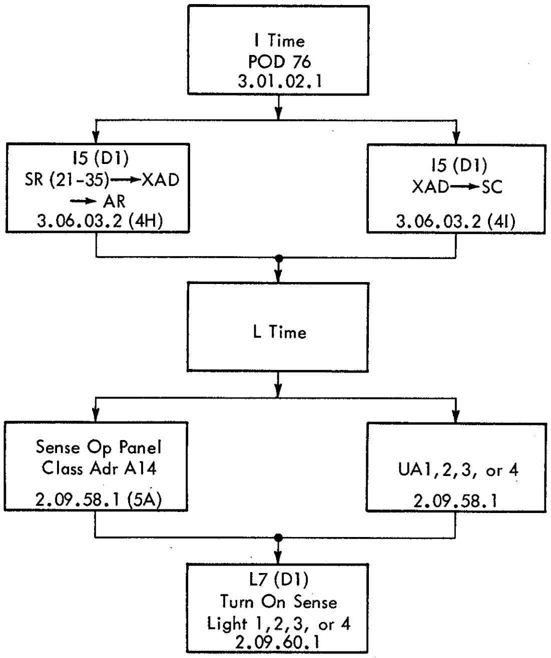


Figure 55. Sense Lights On (SLN)

Sense Lights Off SLF + 0760 . . . 140 (I, L)

This instruction turns off all four sense lights on the operator's console. Address modification may change the instruction. See Figure 56.

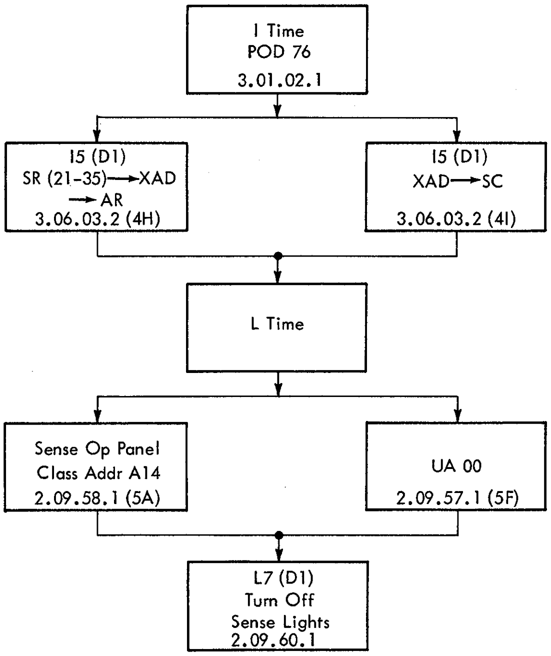


Figure 56. Sense Lights Off (SLF)

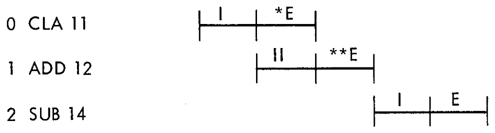
Increased overlap is possible on the 7094 II because of two 16K core storages that can be addressed independently. Two locations, one from the odd array and one from the even array, can be addressed simultaneously. Two words can be obtained, or one word obtained while another is stored. Simultaneous storing is not possible because there is only one MDBI fed by the storage register. Continuous overlapping can occur, resulting in the elimination of all I cycles except the I cycle of the first instruction in the sequence. This is accomplished by overlapping the I cycle with the E or L cycle of the preceding instruction. These overlapped I cycles are called II cycles (IBR I cycles). See Figure 57.

II Time

II time performs the functions of I time for instruction N+1 while instruction N is being processed in E or L time. The instruction received in II time is placed in the IBR. Consequently, decoding circuits used for indexing, indirect addressing, and I, E, or L time call allow the IBR to interpret instructions and to perform I time functions. However, the computer must be in overlap mode for these operations to occur. Also, the instruction in the PR must be one that allows its E/L cycle to overlap, and the instruction in the IBR must allow its I cycle to overlap. No overlap is possible if the instruction in the PR requires processing of data located in the same memory as the next instruction (memory conflict), unless the II cycle is preceded by an I cycle. The I cycle fetches two instructions from storage.

One II cycle occurs for any instruction that permits overlap; it must occur simultaneously with an E or L cycle of the preceding instruction. An II cycle cannot exist alone. Generally, an II cycle will occur during the first E or L cycle of the preceding instruction, allowing for B time or an (IA) E cycle. If the E cycle of the preceding instruction causes a memory conflict, the II cycle occurs during the first L cycle, if the L cycle is required by the preceding instruction, and if both instructions can be overlapped.

The II cycle that occurs immediately after an I cycle does not bring an instruction into the IBR because the instruction is already there. The I cycle fetches two instructions, placing one into the SR and PR, and the other into the IBR. Therefore, the II cycle following an I cycle cannot be inhibited because of a memory conflict, since the instruction that would normally be placed in the IBR during the II cycle has been placed in the IBR during the I cycle (Figure 58). The IBR loaded trigger prevents the instruction fetch during this particular II time. The trigger is set at I5 and is reset at II5.



* Can be overlapped because CLA and ADD were fetched during I Time
** Cannot be overlapped becuase ADD data and SUB are in the same memory

Figure 58. Memory Conflict

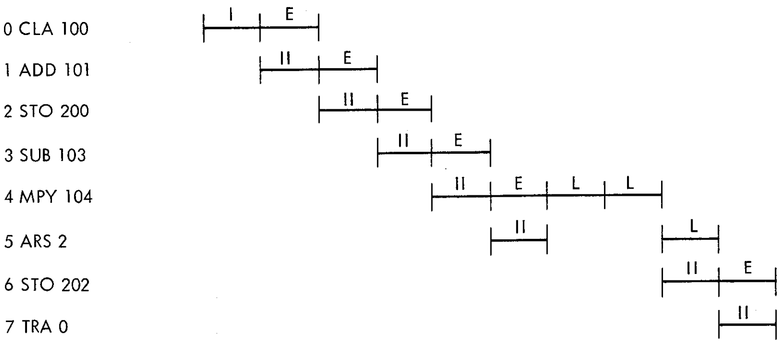


Figure 57. Continuous Overlapping

Setting II Time

Any one of five groups of conditions must be satisfied to set the II time trigger (Figure 59 and II time flow chart). Figure 59 presents the five methods. These conflicts are examples of instructions that cannot be overlapped in either their I or L time.

1. Permits II time after an I cycle, allowing for B time or an IA cycle.

IBR loaded
I time late or IA trigger
Not transfer in trap mode
Not end-op
A6(D1)
Not execute trigger
Overlap mode
Not B time

2. Permits II time during the first L cycle following an E cycle whose address is in memory-select conflict. The L cycle must be required by the same instruction that used the E cycle.

E time late
Not overlap trigger
Not I or E time early
Not end-op
Not pods to inhibit II
Not PR doubel precision
A6(D1)
Not execute trigger
Overlap mode
Not B time

3. Permits II time coincident with the L cycle of a shift instruction decoded in the IBR.

IBR shift decode
End-op
Not skip
Overlap trigger
Not overlap conflict trigger
A6(D1)
Not execute trigger
Overlap mode
Not B time

4. Permits II time after an IA cycle. The IA cycle is preceded by an II cycle.

IA trigger
Not IBR loaded
Not pods to inhibit II
E time trigger
Not IBR-PC select conflict
A6(D1)
CP set A
Not execute trigger
Overlap mode
Not B time
Not IBR Dbl Prec, ANA, ANS, ERA
Not Pre-IIA trigger

5. Permits II time coincident with the E cycle of an instruction decoded in the IBR.

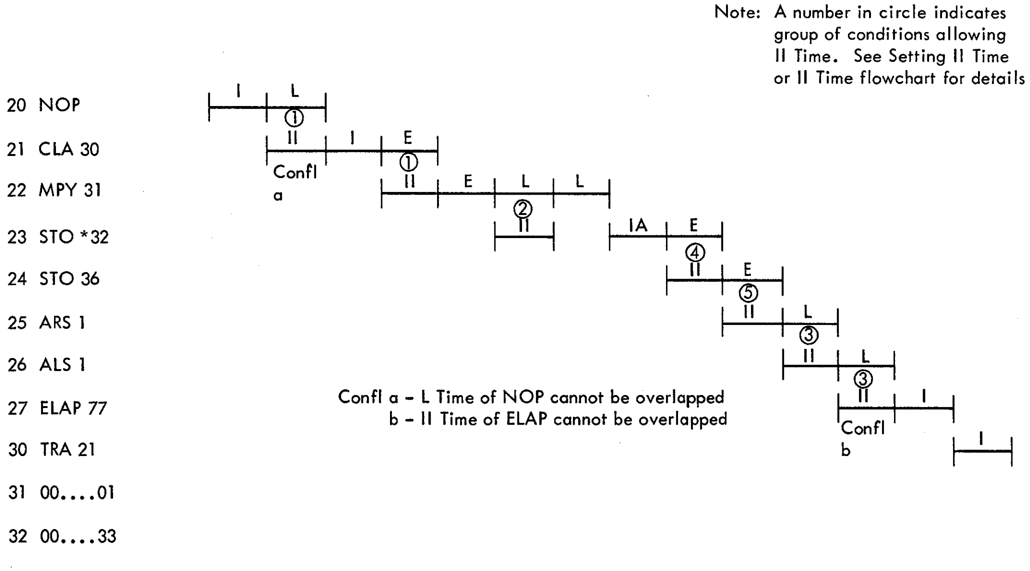
End-op trigger
E time trigger
Not IBR-PC select conflict
A6(D1)
CP set A
Not execute trigger
Overlap mode
Not B time
Not IBR Dbl Prec, ANA, ANS, ERA
Not Pre-IIA trigger

II Time Functions

If conditions are met and overlap is possible, II time performs these functions:

1. Addresses memory from PC at II(6) early (IBR loaded trigger off).
2. Increments PC at II(1).
3. Sets overlap trigger at II(3).
4. Loads IBR from MDBO at II(4) (IBR loaded trigger off).
5. Indexes the IBR (21-35) at II5 or tests IBR (3-17) for class A transfers at II5.
6. Sets the pre-IIA trigger, if needed, at II5.

The remaining II time functions are performed during the II cycle if end-op is on for the previous instruction. If end-op is not on, the remaining functions are performed by the overlap trigger during the E or L cycle, in which the previous instruction ends operation.



Note: A number in circle indicates group of conditions allowing II Time. See Setting II Time or II Time flowchart for details

Confl a - L Time of NOP cannot be overlapped
b - II Time of ELAP cannot be overlapped

Figure 59. Conditions Permitting II Time

The overlap trigger is turned on at II(3) and is reset at I, E, or L(2) with the end-op trigger on for the previous instruction.

7. Set E time at A(5) or L time at A(6) if required.
8. Gate IBR op code to PR at A(7).
9. Gate IBR (28-35) to XAD to SC at A(7) CP set.
10. Gate IBR (21-35) to XAD to AR at A(7) (for double store in IBR).
11. Gate IBR to SR at A(0) (inhibited by IBR OD 6X).

II Time Inhibited

II time is inhibited by either one of two methods:

1. The II time trigger is not turned on.
2. The overlap conflict trigger is turned on to block the functions of II time. The first method is the result of decoding in the PR or IBR, the instructions that cannot have an overlapping II cycle. ERA, CVR, and double precision instructions are examples. ERA uses the IBR, CVR uses the tag register, and double precision uses the IBR (Figures 60 and 61).

When a double precision arithmetic instruction is placed into the IBR during an II cycle, use IBR decoding, shown on Systems 03.08.20.2, to inhibit setting the II time trigger again. For example, if an II cycle were allowed to overlap the E cycle of DFMP, the II cycle would place the next instruction into the IBR at the same time factor D was entering the IBR. Therefore, POD26 is used to inhibit setting II time after A7.

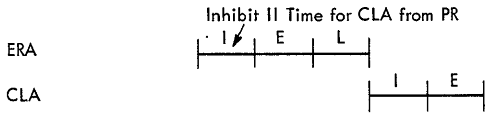


Figure 60. II Cycle Inhibited from Program Register

Likewise, an E_N cycle that attempts to use the memory containing the next instruction will inhibit setting the II time trigger if the IBR is not loaded (II to II). However, the overlap may occur during the L_N cycle if the instruction allowed overlap and required an L cycle.

The second method of inhibiting overlap allows II time to be set, but nullifies its effect by setting the overlap conflict trigger. The overlap conflict trigger is set from the IBR for instructions that cannot have overlapped I cycles, and from the program register for instructions that cannot have overlapped E or L cycles (Figure 62).

The trigger is set at A5 and inhibits all II time functions past A5. There can be no further II cycles until an I cycle has occurred.

If a store instruction, starting in I time, references either of the next two sequential locations, the overlapped II cycle must be inhibited to prevent use of that instruction before it is modified by the store. At II(3) the AR and the complement of the PC are routed to the XAD's with a carry to XAD(17). If XAD (3-16) contains all one's, the overlap conflict trigger is turned on (Figure 67). Figures 63-67 are detailed examples of the conditions previously mentioned, while Figure 68 shows 7094 II time.

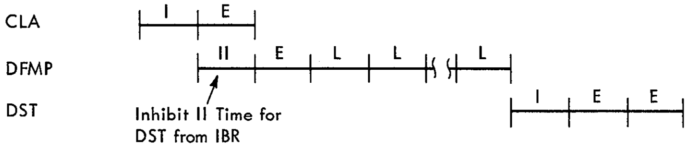


Figure 61. II Cycle Inhibited from IBR

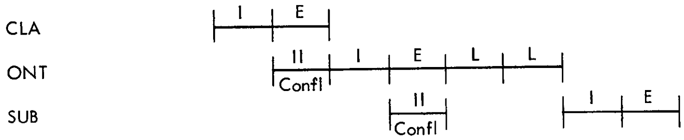


Figure 62. II Cycle Nullified by Overlap Conflict Trigger

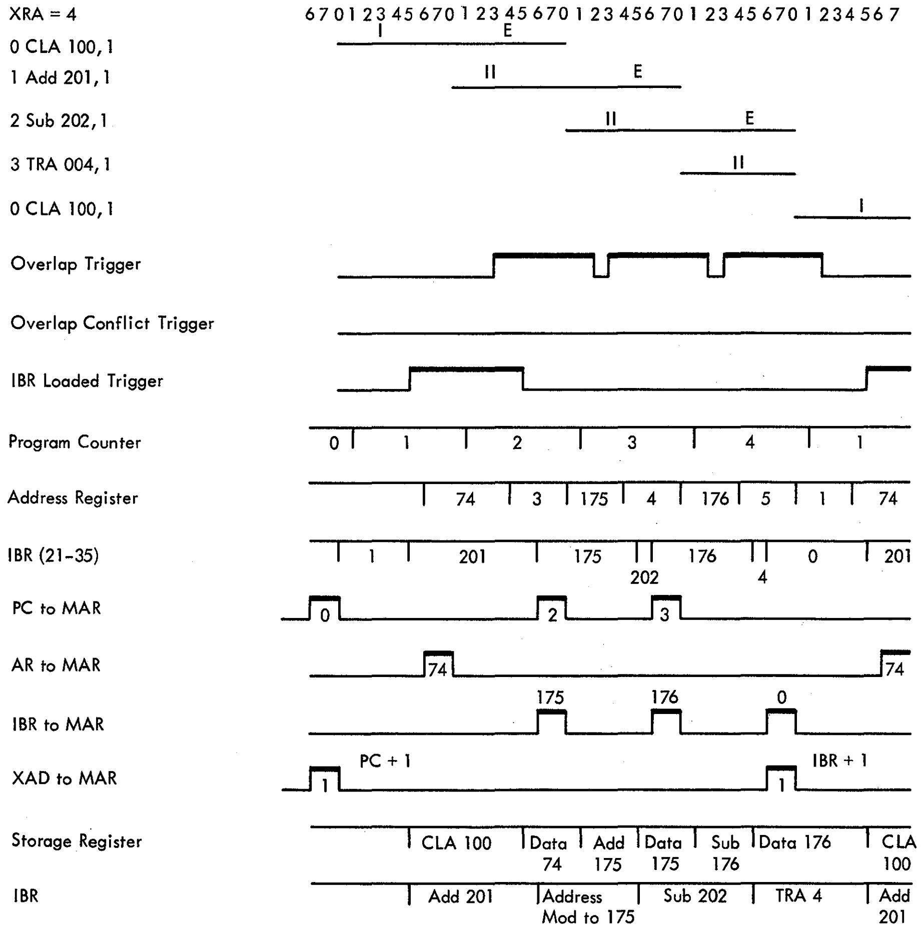


Figure 63. Overlap Example (No Conflicts)

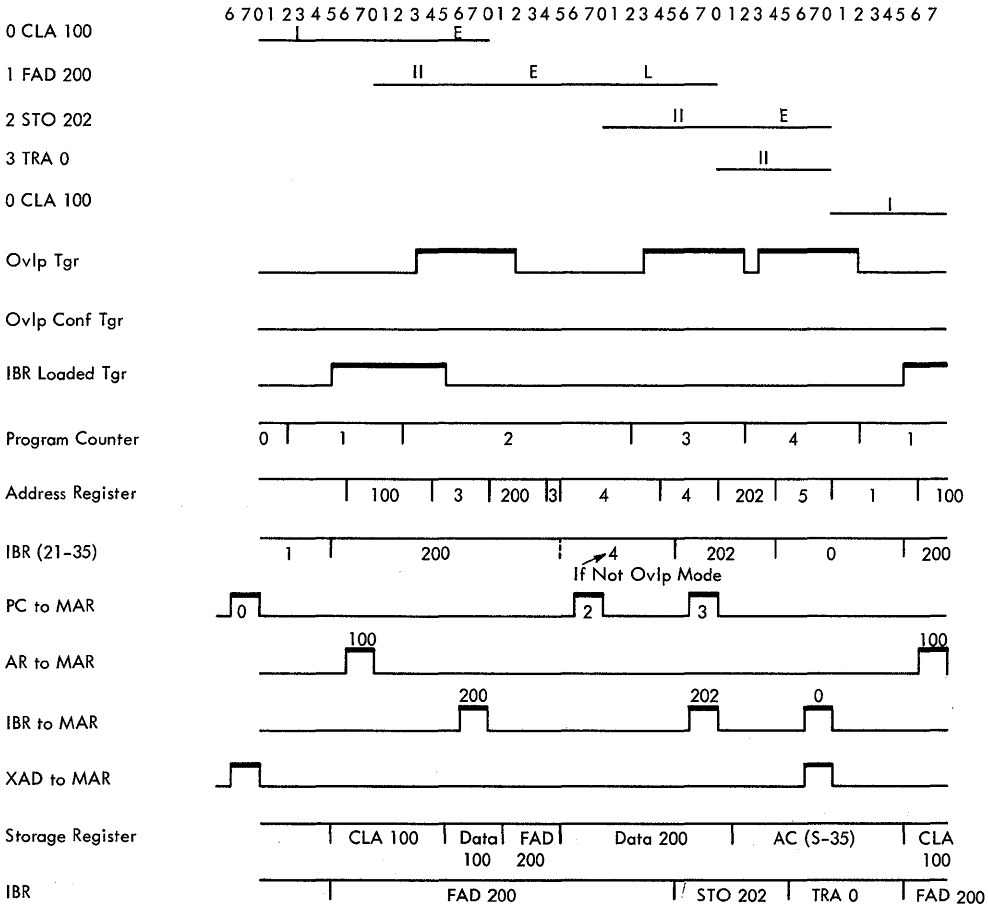


Figure 64. Overlap Example (E-Time Conflict, Overlap L Time)

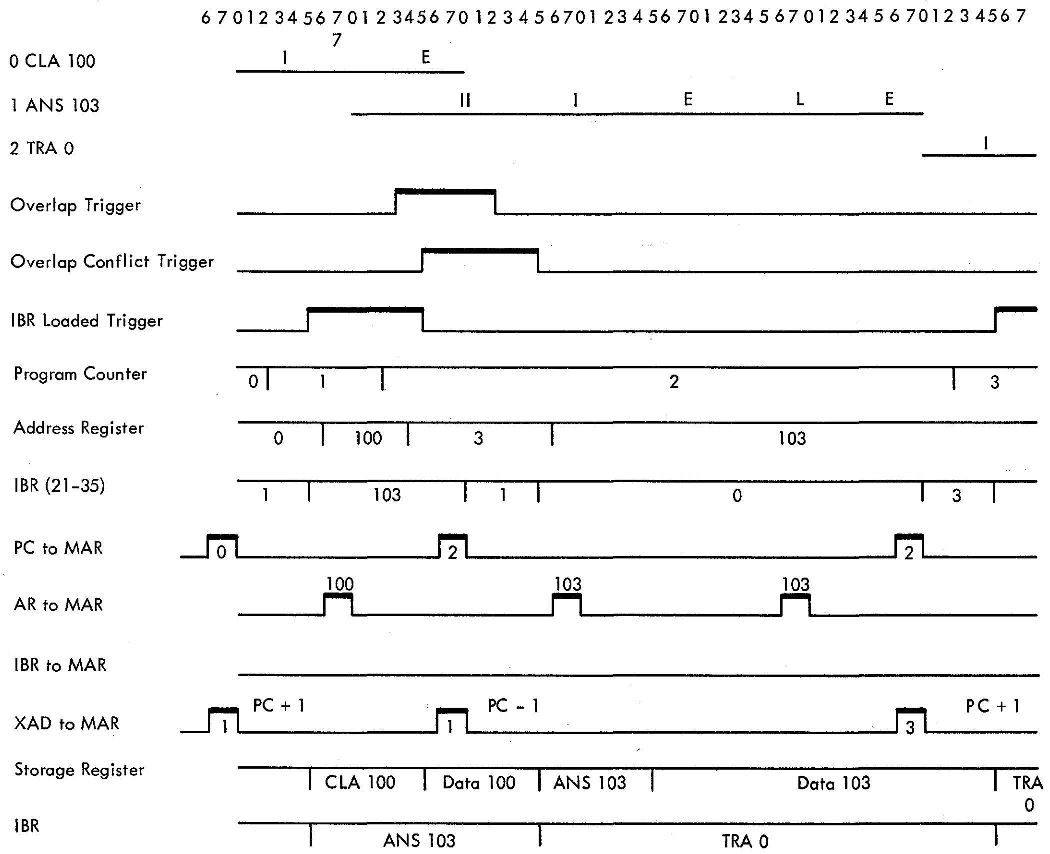


Figure 65. Overlap Example (No Overlap Allowed)

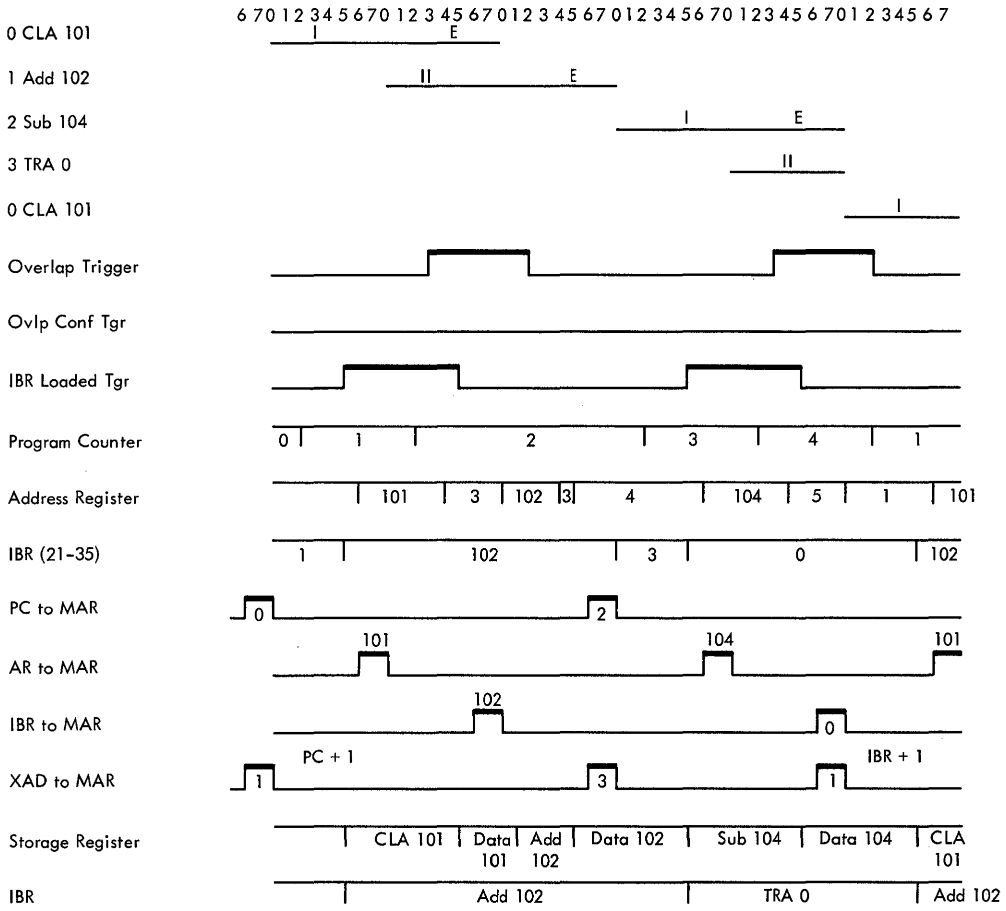


Figure 66. Overlap Example (Address Conflict)

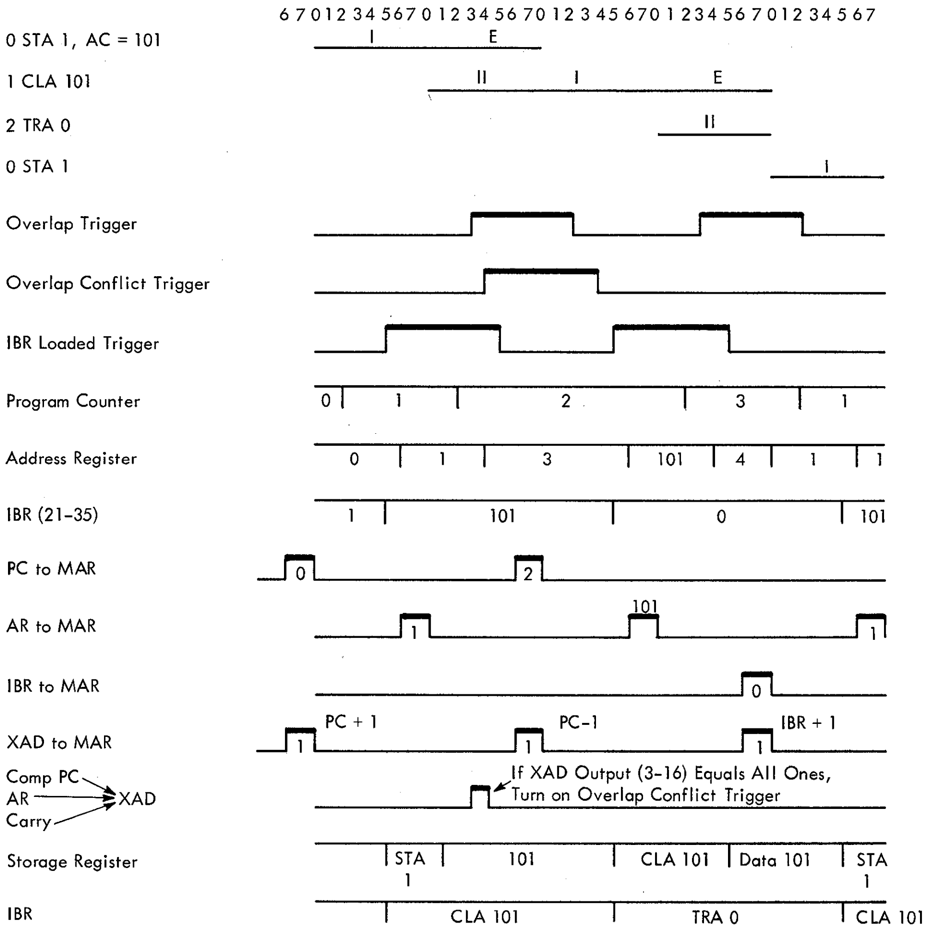


Figure 67. Overlap Example (Store Conflict)

(1) Permits II time after an I cycle, allowing for B time or an IA cycle:

a. IBR loaded
b. I time late or IA Tgr
c. Not trap transfer
d. Not end-op
e. Not PODS to inhibit II

(2) Permits II time during the first L cycle following an E cycle whose address was in memory-select conflict. The L cycle must be required by the same instruction that used the E cycle:

a. E time late
b. Not overlap trigger
c. Not I or E time early
d. Not end-op
e. Not PODS to inhibit II
f. Not PR DBL-PREC

(3) Permits II time coincident with the L cycle of a shift instruction decoded in the IBR:

a. IBR shift decode
b. End-op
c. Not skip
d. Overlap trigger
e. Not overlap conflict trigger

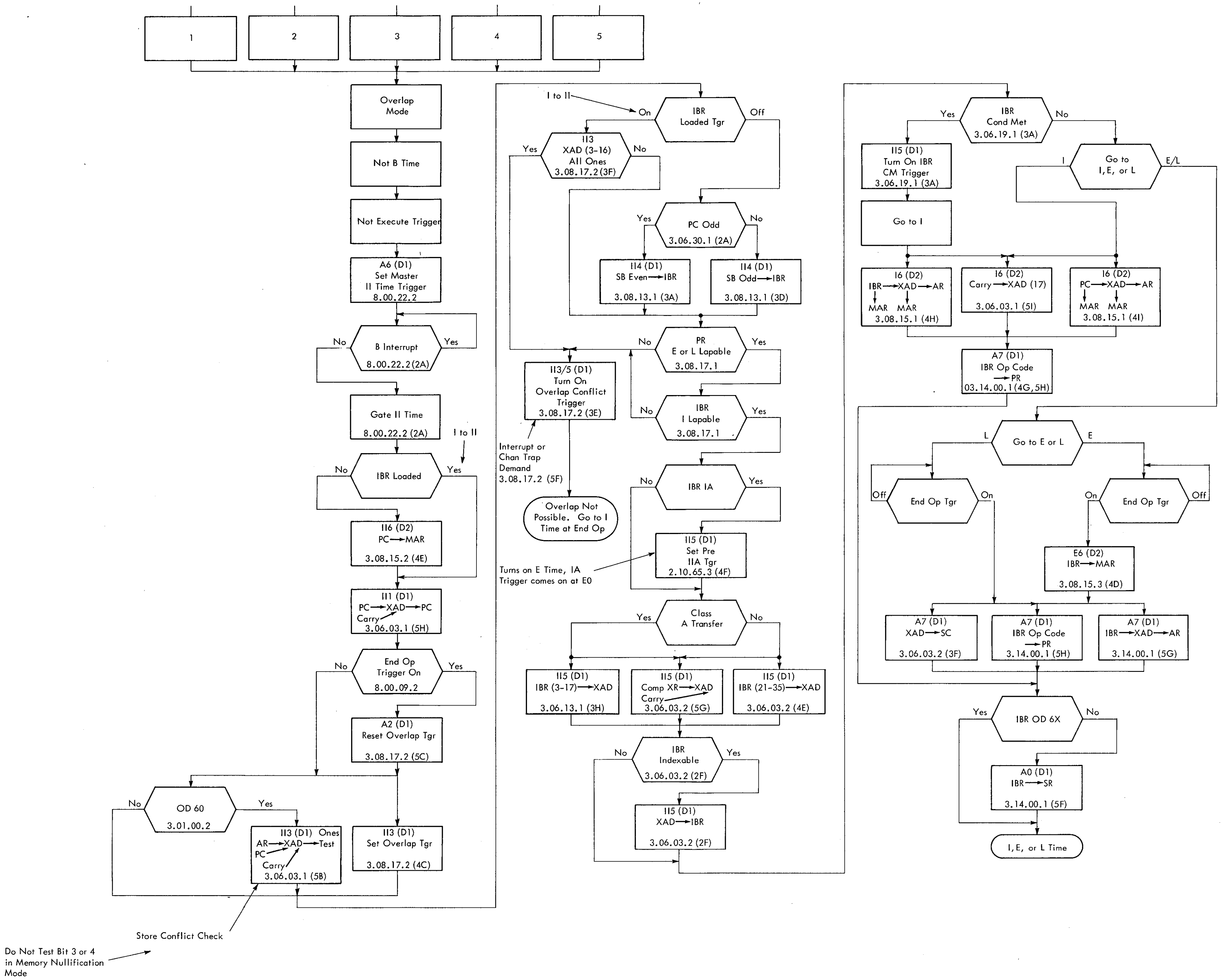
(4) Permits II time after an IA cycle. The IA cycle was preceded by an II cycle:

a. IA trigger
b. IBR not loaded
c. Not PODS to inhibit II
d. E time trigger
e. Not IBR-PC select conflict
f. Not pre IIA trigger
g. Not IBR DBL-PREC, ANA, ANS, or ERA

(5) Permits II time coincident with the E cycle of an instruction decoded in the IBR:

a. End-op trigger
b. E time trigger
c. Not IBR-PC select conflict
d. Not IBR-DBL-PREC, ANA, ANS, or ERA
e. Not pre-IIA trigger

○ Refer to Figure 59



1
2
3
4
5
Overlap Mode
Not B Time
Not Execute Trigger
A6 (D1) Set Master II Time Trigger 8.00.22.2
No
B Interrupt 8.00.22.2 (2A)
Yes
Gate II Time 8.00.22.2 (2A)
I to II
No
IBR Loaded
Yes
II6 (D2) PC→MAR 3.08.15.2 (4E)
II1 (D1) PC→XAD→PC Carry 3.06.03.1 (5H)
No
End Op Trigger On 8.00.09.2
Yes
A2 (D1) Reset Overlap Tgr 3.08.17.2 (5C)
No
OD 60 3.01.00.2
Yes
II3 (D1) Ones AR→XAD→Test PC Carry 3.06.03.1 (5B)
II3 (D1) Set Overlap Tgr 3.08.17.2 (4C)
Store Conflict Check
Do Not Test Bit 3 or 4 in Memory Nullification Mode
I to II
On
IBR Loaded Tgr
Off
Yes
II3 XAD (3-16) All Ones 3.08.17.2 (3F)
No
Yes
PC Odd 3.06.30.1 (2A)
No
II4 (D1) SB Even→IBR 3.08.13.1 (3A)
II4 (D1) SB Odd→IBR 3.08.13.1 (3D)
No
PR E or L Lapable 3.08.17.1
Yes
II3/5 (D1) Turn On Overlap Conflict Trigger 3.08.17.2 (3E)
Interrupt or Chan Trap Demand 3.08.17.2 (5F)
Overlap Not Possible. Go to I Time at End Op
No
IBR I Lapable 3.08.17.1
Yes
No
IBR IA
Yes
II5 (D1) Set Pre IIA Tgr 2.10.65.3 (4F)
Turns on E Time, IA Trigger comes on at E0
Yes
Class A Transfer
No
II5 (D1) IBR (3-17)→XAD 3.06.13.1 (3H)
II5 (D1) Comp XR→XAD Carry 3.06.03.2 (5G)
II5 (D1) IBR (21-35)→XAD 3.06.03.2 (4E)
No
IBR Indexable 3.06.03.2 (2F)
Yes
II5 (D1) XAD→IBR 3.06.03.2 (2F)
Yes
IBR Cond Met 3.06.19.1 (3A)
No
II5 (D1) Turn On IBR CM Trigger 3.06.19.1 (3A)
I
Go to I, E, or L
E/L
Go to I
I6 (D2) IBR→XAD→AR MAR MAR 3.08.15.1 (4H)
I6 (D2) Carry→XAD (17) 3.06.03.1 (5I)
I6 (D2) PC→XAD→AR MAR MAR 3.08.15.1 (4I)
A7 (D1) IBR Op Code →PR 03.14.00.1 (4G, 5H)
L
Go to E or L
E
Off
End Op Tgr
On
On
End Op Tgr
Off
E6 (D2) IBR→MAR 3.08.15.3 (4D)
A7 (D1) XAD→SC 3.06.03.2 (3F)
A7 (D1) IBR Op Code →PR 3.14.00.1 (5H)
A7 (D1) IBR→XAD→AR 3.14.00.1 (5G)
Yes
IBR OD 6X
No
A0 (D1) IBR→SR 3.14.00.1 (5F)
I, E, or L Time

Figure 68. 7094 II Time

Trapping

Special computer conditions can be indicated in various ways; for example, console lights, triggers, tone signals, program stops, and printouts. Some conditions can be tested with instructions, but this is time consuming. Others stop the computer and require operator intervention; this is not only time consuming but also subject to human error. The answer lies in letting the computer monitor these special conditions and take the necessary action immediately, at computer speed.

Trapping occurs when certain indications are detected by circuitry within the computer. When these indications are detected, the computer takes control of the program and causes an unconditional transfer to a specific location. As the computer is trapped, the program counter contents and identification bits are stored; using these, programming subroutines can test and determine the cause of the trap and exactly where it occurred in the computer program. With this information, it is possible to reconstruct the cause, take appropriate action, and continue again with the main program. Figure 69 is a summary of the various trap locations, the conditions causing the trap, and the in-

Location	Condition	Address	Decrement
00000	Trap Mode	Location of Transfer Instruction	Not Used
	FP Trap	Location +1 of FP Inst	Bit code of trap conditions
	Divide Check	Location +1 of DIV Inst	Bit indication (Pos 13)
	STR	Location +1 of STR Inst	Not Used
	Significant Arithmetic	Location +1 of FP Inst	Not Used
00001	Transfer instruction for transfer trap mode.		
00002	Transfer instruction of STR.		
00003	Direct Data	PC +1	Channel indication (A-H)
	7281 Data Communication Channel	PC +1	Channel indication (A-H)
	Interrupt	PC +1	Not Used
00004	Transfer instruction for direct data, interrupt, or 7281 Data Communications Channel		
00005	Location of Storage cell clock and interval timer		
00006	SCCT Trap	PC +1	Not Used
00007	Transfer instruction of storage cell clock trap.		
00010	Transfer instruction for floating-point or divide check traps.		
00011	Transfer instruction for significant arithmetic trap.		
00012	Channel A	PC +1	Trap bit code
00013	Transfer instruction for channel A trap		
00014	Channel B	PC +1	Trap bit code
00015	Transfer instruction for channel B trap		
00016	Channel C	PC +1	Trap bit code
00017	Transfer instruction for channel C trap		
00020	Channel D	PC +1	Trap bit code
00021	Transfer instruction for channel D trap		
00022	Channel E	PC +1	Trap bit code
00023	Transfer instruction for channel E trap		
00024	Channel F	PC +1	Trap bit code
00025	Transfer instruction for channel F trap		
00026	Channel G	PC +1	Trap bit code
00027	Transfer instruction for channel G trap		
00030	Channel H	PC +1	Trap bit code
00031	Transfer instruction for channel H trap		
00032	Addressable Memory Protect	PC +1	Violation code
	Memory Protect with Relocation	PC +1	Not Used
00033	Transfer instruction for addressable memory protect or Memory Protect with Relocation traps.		
00034	Not Used		
00035	Not Used		
00036	Interval Timer Reset	PC +1	Not Used
00037	Transfer instruction for interval timer Reset trap		
00040	Storage Parity (7040/7044)	PC +1	Error address
00041	Transfer instruction for storage parity trap		
00042	Channel A 7909 Int.	Command Counter	Address Counter
00043	Transfer command for 7909 channel A interrupt		
00044	Channel B 7909 Int	Command Counter	Address Counter
00045	Transfer command for 7909 channel B interrupt		
00046	Channel C 7909 Int.	Command Counter	Address Counter
00047	Transfer command for 7909 channel C interrupt		
00050	Channel D 7909 Int.	Command Counter	Address Counter
00051	Transfer command for 7909 channel D interrupt		
00052	Channel E 7909 Int.	Command Counter	Address Counter
00053	Transfer command for 7909 channel E interrupt		
00054	Channel F 7909 Int.	Command Counter	Address Counter
00055	Transfer command for 7909 channel F interrupt		
00056	Channel G 7909 Int.	Command Counter	Address Counter
00057	Transfer command for 7909 channel G interrupt		
00060	Channel H 7909 Int.	Command Counter	Address Counter
00061	Transfer command for 7909 channel H interrupt		
40,000	Select or Copy Trap	Location +1 of trap instruction	Not Used
40,001	Transfer instruction for select trap		
40,002	Transfer instruction for copy trap		

Figure 69. Trap Locations

formation automatically stored in the address and decrement portions. Note that this is a complete list of trap locations; some locations apply only to special features or other systems (i.e., location 0040, storage parity, pertains to 7040/7044).

There are five major types of traps:

Transfer traps
Arithmetic traps
I/O traps
Compatibility traps
Special traps

Store Location and Trap STR — 1000 (I, E)

This instruction stores the value of the program counter (location of the STR instruction plus one) in the address portion of location 000 and traps to location 002. STR is not affected by trap mode and does not place the computer in trap mode. Location zero (S-20) is not changed. See Figure 70.

The address register is reset at I6 so that a zero address is available to MAR for the E cycle. The program counter is routed to the storage register (21-35) via the index adders. Storage register (S-20) is cleared. At E5, the address register is set to 002, and at I6 the address register and index adders are gated to MAR.

Transfer Trap Mode

In program debugging, it is helpful to prevent successful transfer instructions from transferring. In this way, "wild" transfers are caught before they damage the rest of the program or data located in core storage.

In the trap mode of operation, the location of all transfer instructions encountered (with two exceptions) is automatically stored in the address portion of location 00000_8. The normal operating functions of the instruction are completed, and, if the transfer conditions are successfully met, the actual transferring is blocked and a trap is initiated to location 00001_8. In 00001_8 there is normally an unconditional trap transfer (TTR) to a system program subroutine which analyzes the transfer instruction causing the trap. At this point, a table of all successful transfers can be accumulated or the transfer instruction can be tested further to determine if the transfer conditions are valid. If valid, the subroutine may allow the transfer to be executed; if invalid, the subroutine may cause some form of machine stop or printout to the console operator.

Indirect addressing is blocked for transfer instructions in the trap mode; it could perform no logic because of the trapping on successful transfer conditions. Indirect addressing, if it is called for in the instruction by bits in positions 12 and 13, must be tested and determined by the appropriate subroutine.

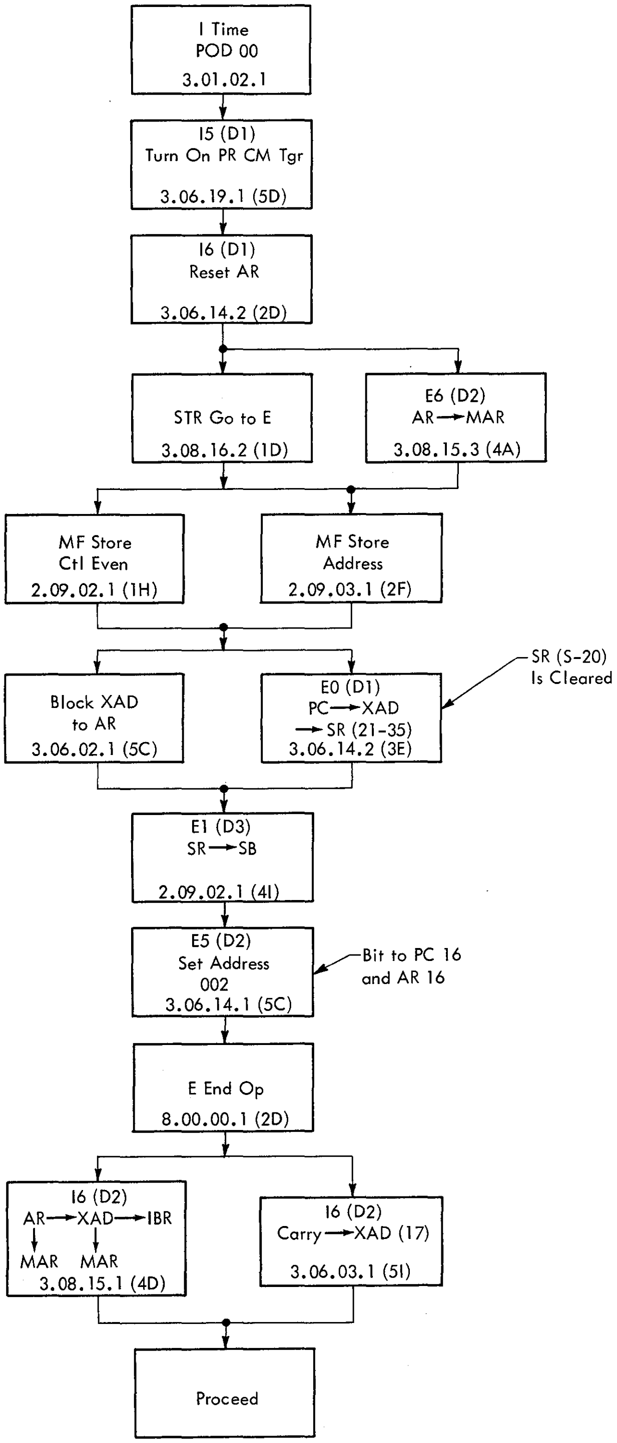


Figure 70. Store Location and Trap (STR)

The computer is placed in trap mode by execution of the enter trapping mode (ETM) instruction. Exit from the trap mode is accomplished by execution of the leave trapping mode (LTM) instruction or by pressing either the clear or reset keys on the operator's console. Two instructions are immune to the trap mode: trap transfer (TTR), and enter storage nullification and transfer (ESNT).

Enter Trapping Mode ETM + 0760 . . . 0007 (I, L)

This instruction places the computer in trapping mode by turning on the trap mode trigger as shown in Systems 02.10.53.1. In this mode of operation, all successful transfers except TTR and ESNT store their storage locations in the address portion of location zero and trap to location 0001.

The computer remains in the transfer trapping mode until execution of the leave trapping mode (LTM) instruction or until the clear or reset buttons are depressed at the operator's console. See Figure 71.

Leave Trapping Mode LTM − 0760 . . . 0007 (I, L)

This instruction takes the computer out of transfer trapping mode by resetting the trap mode trigger as shown in Systems 02.10.53.1.

If the computer is not in transfer trap mode, this instruction has the effect of a NOP. See Figure 71.

Trap Transfer TTR + 0021 (I)

This instruction is an unconditional transfer and will transfer whether the computer is in transfer trap mode or not.

If the computer is in transfer trap mode, TTR will prevent the −A block 4B on Systems 03.06.14.2 from being activated; in effect, nullifying trap mode for the duration of the instruction and causing TTR to act like a normal transfer. See Figure 72.

Trapping Operation TOV/TNO

The objectives of the operation are (Figure 73):

1. To store the location of the transfer instruction in the address portion of core storage location zero.

2. To trap the program to location one if a successful transfer is indicated, or to proceed to the next sequential instruction if the transfer is not successful.

3. To perform the functions called for by the particular instruction.

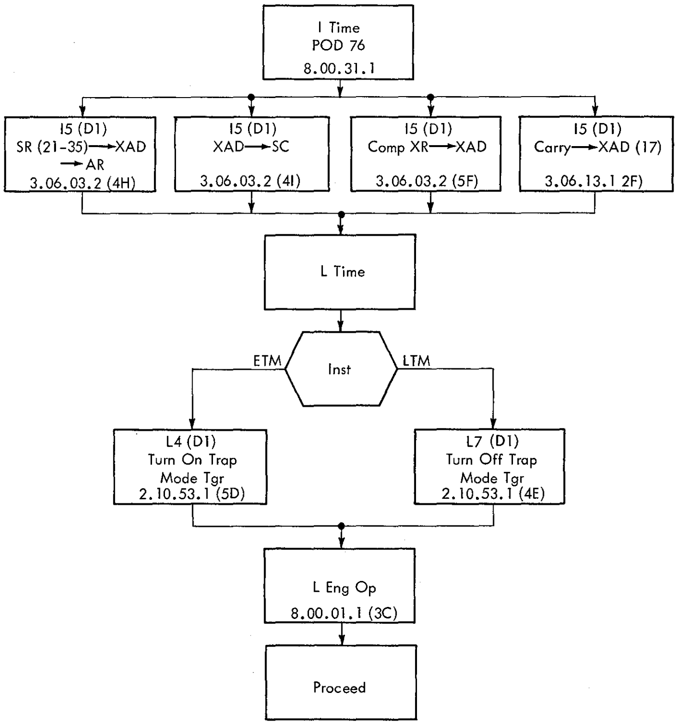


Figure 71. Enter Trapping Mode (ETM); Leave Trapping Mode (LTM)

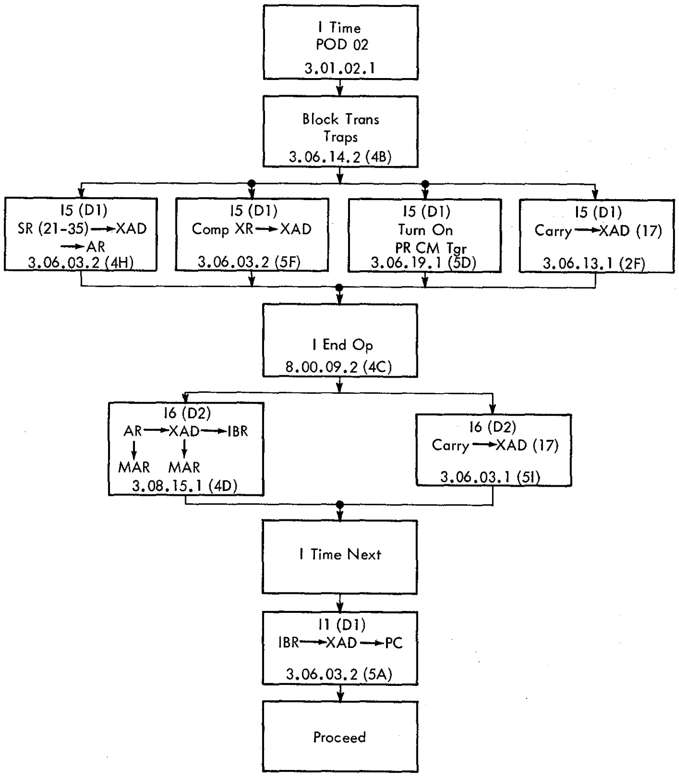


Figure 72. Trap Transfer (TTR)

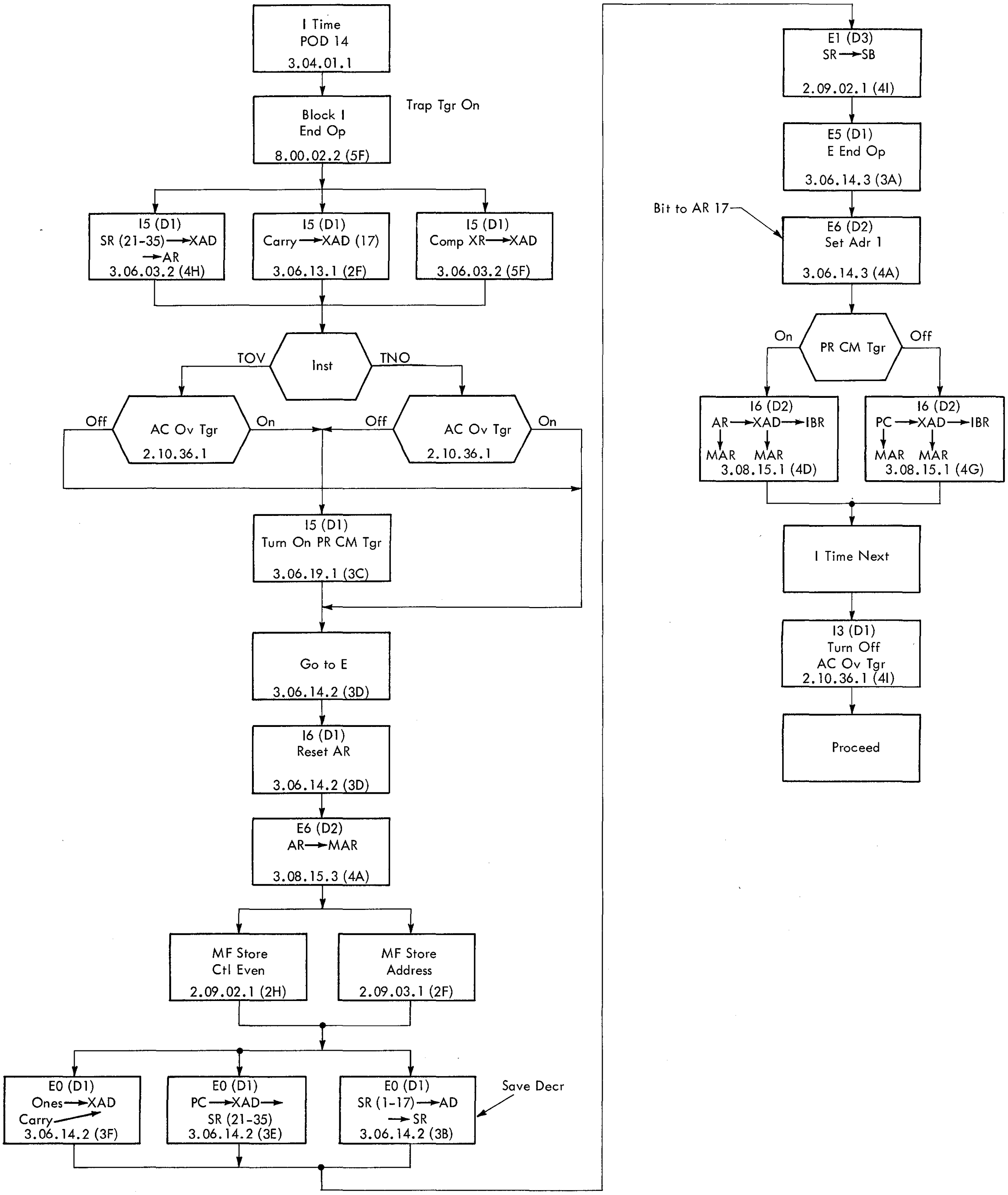


Figure 73. TOV, TNO Trap Mode

The first objective is realized by forcing an E cycle and storing the contents of the program counter minus one in location zero. The decrement portion of the storage register is saved by routing the storage register to the adders, and back to the storage register. This is done because the decrement may be needed to modify an index register later, for example, on a TIX instruction.

The address register is reset at I6 to obtain the zero address. "MF store address" causes the address portion of the storage register, containing the program counter value minus one to be stored. The rest of location zero is not changed.

The second objective is realized by testing the accumulator overflow trigger during the first I cycle. If conditions are met, the PR-CM-Tgr is turned on. At the end of the E cycle this trigger is tested. If it is on, the address register which was set to one at E6 is gated to MAR for the trap to location one; if it is not on, the program counter is gated to MAR, causing the computer to proceed to the next sequential instruction after the TOV/TNO.

The third objective, that of performing the function called for by the instruction, is performed at the beginning of the I cycle following TOV/TNO. The accumulator overflow trigger is turned off at I3. All functions of a transfer type instruction occurring in trap mode are performed, except for the transfer.

Floating-Point Trapping

During floating-point operations (or any mathematical operation), the answer must be constantly tested to determine if it has exceeded the register capacity. To eliminate this continual, time consuming job of testing with actual instructions, 7094 II circuitry automatically assumes the responsibility of continuously monitoring for overflow and underflow conditions. If these conditions arise in either the AC or MQ during a floating-point operation, the computer stops the sequential execution of instructions and, instead, stores the program counter (location of the FP instruction +1) in the address portion of location 00000_8, and traps to location 00010_8.

To further pinpoint the operation causing the trap, an identifying bit code is stored in the decrement of location 00000_8. The decrement bit codes and identifications are:

Position 14—Indicates that the overflow or underflow occurred during a floating-point divide operation.

Position 15—Indicates an overflow in either the AC or MQ.

Position 16—Indicates an overflow or underflow in the AC.

Position 17—Indicates an overflow or underflow in the MQ.

The identifying bit codes can be further associated with the various floating-point operations as follows:

FP OPERATION (WITH POSSIBLE TYPES)	TYPE	AC	MQ	DECREMENT POSITION			
				14	15	16	17
Floating Round (C)	A	—	Unfl	0	0	0	1
Single and Double-Precision: Add or Subtract (A, B, C);	B	Unfl	Unfl	0	0	1	1
Multiply (A, B, C, D)	C	Ovfl	—	0	1	1	0
Double-Precision Divide (A, B, C)	D	Ovfl	Ovfl	0	1	1	1
Single-Precision Divide (E, F, G, H)	E	—	Unfl	1	0	0	1
	F	Unfl	—	1	0	1	0
	G	Unfl	Unfl	1	0	1	1
	H	—	Ovfl	1	1	0	1

Overflow and underflow in the AC and MQ is detected and identified as follows:

1. Overflow or underflow of the floating-point characteristics is detected by examining positions P and Q of the main adder and AC.

2. MQ overflow and underflow must be recognized as the MQ characteristic is computed in the adders.

3. AC overflow or underflow may be recognized at any time after the final characteristic has been assigned.

NOTE: In each of the above cases, overflow is recognized by a bit in AD(P) only; and underflow is recognized by bits in AD (P and Q).

Floating-Point Trap Operation

Objectives of the floating-point trap operation are (Figure 74):

1. To take priority over other traps.

2. To store the location of the floating point instruction plus one in the address portion of location zero.

3. To store identification bits in the decrement portion of location zero.

4. To trap the computer to location 00010_8.

Conditions causing a floating point trap were listed in the preceding section. The floating point trap trigger (02.10.51.1) is turned on at I2 of the I cycle following the FP instruction that caused the trap. Because the floating point trap trigger is set at I2, the first objective of taking priority over other traps is realized. If several types of trapping situations occur simultaneously, the floating point trap assumes priority and is serviced first.

At I4, the incoming instruction is blocked from entering the program register (03.14.00.1-3A), and the computer is forced into an STR operation (03.14.00.1-2A).

The address register is reset at I6 and gated to MAR to provide the zero address. At E0, the program counter is gated to the storage register (21-35) via the index adders. All ones are also gated to the index adders; in effect, subtracting one from the value of the program

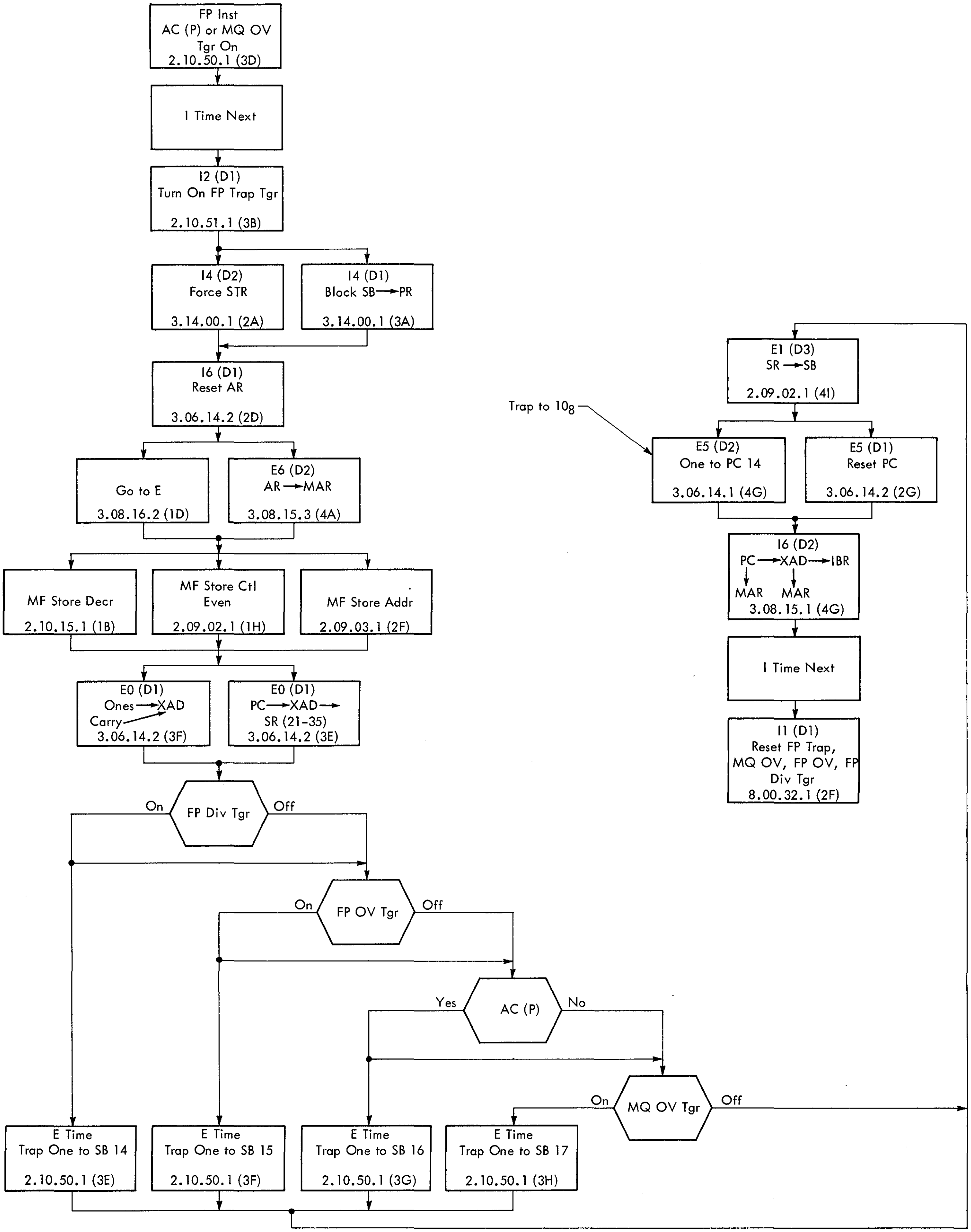


Figure 74. Floating-Point Trap Operation

counter and providing the location of the floating-point instruction plus one. The storage register (S-20) is cleared.

As in Systems 02.10.50.1, identifying bits are placed directly on the storage bus (14-17) according to conditions causing the trap. MF store decrement and MF store address have been activated so that only the prefix and tag bits of location zero are not changed. The second and third objectives, of storing the location of the floating point instruction plus one, and storing identifying bits in the address and decrement portions, respectively, of location zero, have been realized.

At E5 (D1) the program counter is reset (03.06.14.2-2G) and at E5 (D2) (03.06.14.1-4G) program counter position 14 is set to one, providing an address of 00010_8. The program counter and index adders are gated to MAR at I6, and the fourth objective of trapping to location 0010_8 is realized.

The FP trap, MQ overflow, FP overflow, and FP divide triggers are reset at I1 of the next I cycle (08.00.32.1-2F).

Data Channel Trapping

Without the data channel trapping feature, instruction time is wasted to test channel conditions or to synchronize the operation with the main computer. Trapping allows the channels to monitor their own special conditions and to notify the computer if and when these special conditions do occur (Figure 75).

The three types of channel traps and their meanings are:

Control Word Trap (CWT): This occurs whenever an IOCT, IORT, or IOST command is completed in the 7607 Data Channel, and no load channel instruction (LCH) is waiting in the main program. The CWT trap can also occur with the 7909 Data Channel because of a trap and wait (TWT) command.

Tape Check Trap (TCT): Recognition of a redundancy check condition in the 7607 Data Channel.

End-Of-File Trap (EOF): Recognition of an end-of-file in the 7607 Data Channel.

Whenever a trap occurs, the program counter is stored in the address portion of a fixed location and the next instruction is executed:

CHANNEL	PROGRAM COUNTER STORED AT	NEXT INSTRUCTION EXECUTED FROM
A	00012	00013
B	00014	00015
C	00016	00017
D	00020	00021
E	00022	00023
F	00024	00025
G	00026	00027
H	00030	00031

In addition, indication bits are stored in the decrement:

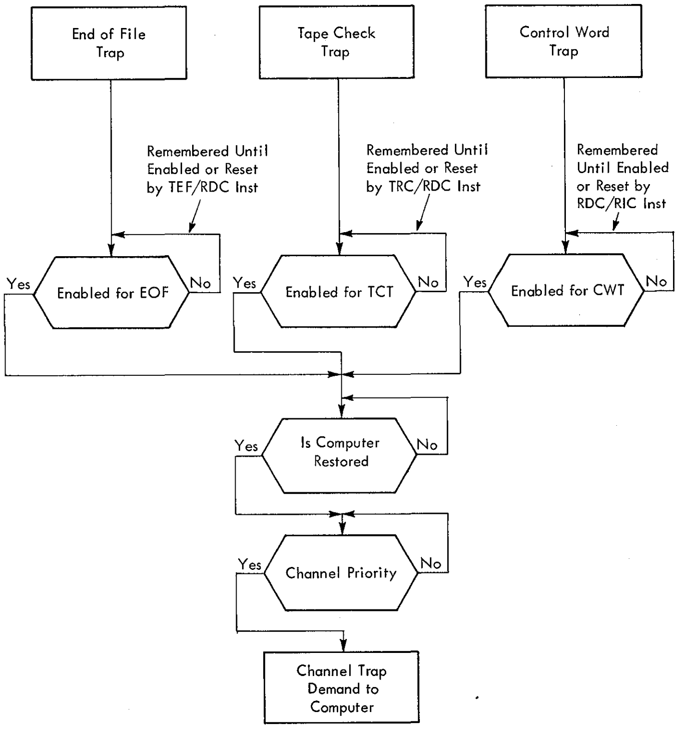
DECREMENT POSITION	TRAP CONDITION
17	Control word trap
16	Tape check trap
15	End-of-file trap

Before the channel can send a trapping request, it must be properly enabled by an enable (ENB) instruction. One or any combination of the trap conditions can be enabled by this instruction. A trap condition that arises when the channel is not enabled, is remembered until a proper ENB is executed for the channel, or until the trap condition is reset (Figure 75).

Channel traps are low on the order of servicing priority in the computer. This causes no concern because delay of a channel trap does not interfere with I/O operation; every 7607 Data Channel trap is accompanied by a disconnect condition within the channel.

Channel traps are normally serviced in the I time following their receipt. Special conditions (excluding interrupts—a special feature) can delay trapping in the computer.

1. If the computer is executing a floating-point instruction which requires a FP trap, the channel trap is



End of File Trap
Tape Check Trap
Control Word Trap
Remembered Until Enabled or Reset by TEF/RDC Inst
Remembered Until Enabled or Reset by TRC/RDC Inst
Remembered Until Enabled or Reset by RDC/RIC Inst
Yes
Enabled for EOF
No
Yes
Enabled for TCT
No
Yes
Enabled for CWT
No
Yes
Is Computer Restored
No
Yes
Channel Priority
No
Channel Trap Demand to Computer

Figure 75. Channel Trap Sequence

delayed until after the FP trap has stored the program counter and trapped to location 00010_8.

2. Any trap demand during the execution of a read select, write select, ENB, XEC, or RCT instruction is delayed one instruction. For the read or write select, this delay prevents a possible I/O check by allowing the execution of a reset and load channel (RCH) instruction before the trap subroutine. For the ENB or RCT, the delay allows a transfer to be executed back to the main program.

After the channel trap is initiated, all subsequent trap requests are blocked until another ENB or RCT instruction is executed.

If the instruction in the trap-to address is not a transfer-type instruction, the program will continue from the point at which the trap occurred.

Enable ENB + 0564 (I, Ec)

This instruction conditions the data channels to allow certain traps. Bit combinations in the storage location referred to by the enable instruction determine the allowable channels and traps (Figure 76):

DATA	DECREMENT POSITION	ADDRESS POSITION
CHANNEL	(TCT TRAP)	(CWT AND EOF TRAPS)
A	17	35
B	16	34
C	15	33
D	14	32
E	13	31
F	12	30
G	11	29
H	10	28

The address register is set to the storage reference at 15. CPU E time is blocked (08.00.19.2-4A) and channel E time is set with a channel A11 pulse. CPU goes to L time, sets the restore trigger at L1 (02.10.56.1-2D), resets the MF go trigger at L4 or channel E0, and sets the L end-op sync trigger (08.00.02.2-5C).

All channel enable triggers are reset at channel E0 (E0c) (c = channel time). Thus, an enable instruction cancels the effect of any previous enable.

At E2c the address register is gated to MAR. By E8c, the storage word is on the MDBO and is used to set the specified enable triggers. At E8c, the channel L-E end trigger and the MF go trigger are also set. If, by E8c, a trap condition exists and the channel has priority, a trap demand is sent to CPU. However, the trap demand will not be serviced by CPU. This demand will be delayed until after execution of the instruction following ENB (02.10.59.1-4F). This gives the programmer an opportunity to transfer back to his main program before confronting another trap. The channel L-E end trigger is reset at I3 of the I cycle following ENB.

When another trap is serviced, the restore trigger is turned off and further trapping is inhibited until execution of a subsequent ENB or RCT instruction.

Restore Channel Traps RCT + 0760 . . . 00014 (I, L)

When a channel trap is serviced, all further trapping is inhibited until either an ENB or RCT instruction is executed. The RCT instruction removes inhibiting and allows subsequent trapping. Execution of the RCT instruction does not change the enabling status in the

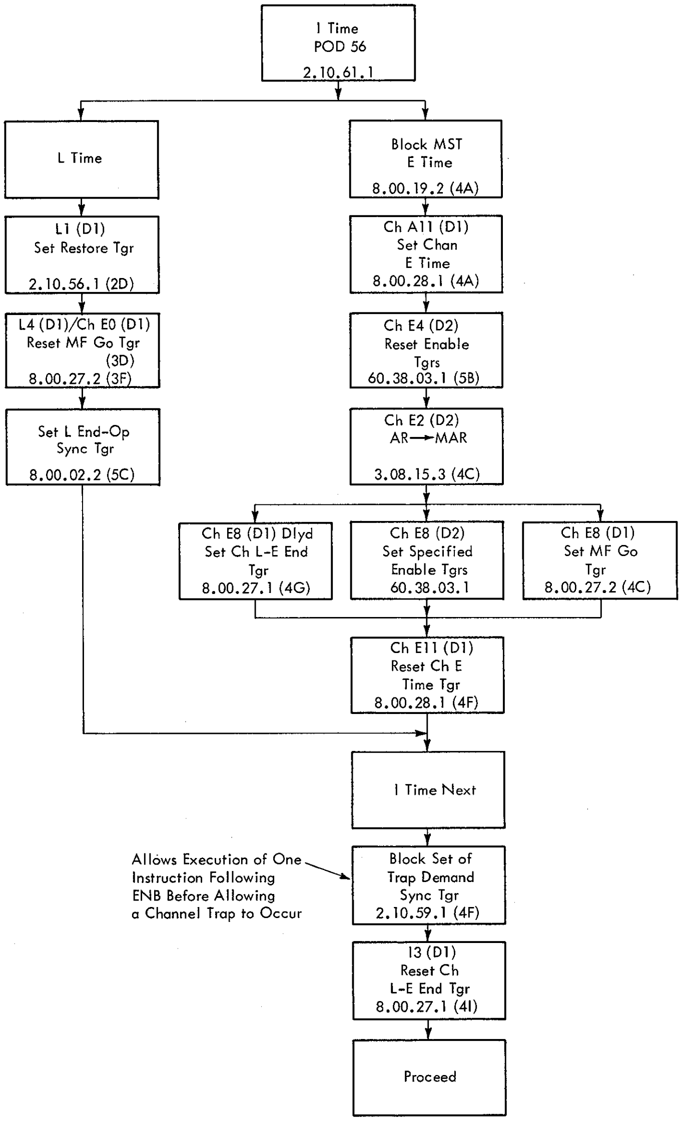


Figure 76. Enable (ENB)

channels originally set up by an ENB instruction. See Figure 77.

The address portion of the storage register is gated to the shift counter via the index adders at I5. Restore gate (03.03.15.1) is activated by shift counter decoding and sets the restore trigger at L1 (02.10.56.1). The restore trigger ON allows the channel to request priority if a trap is waiting to be serviced. Restore decoding inhibits setting the channel trap trigger during the instruction following RCT (02.10.56.1-3D). This gives the programmer time to transfer back into the main program before another trap is serviced by the computer.

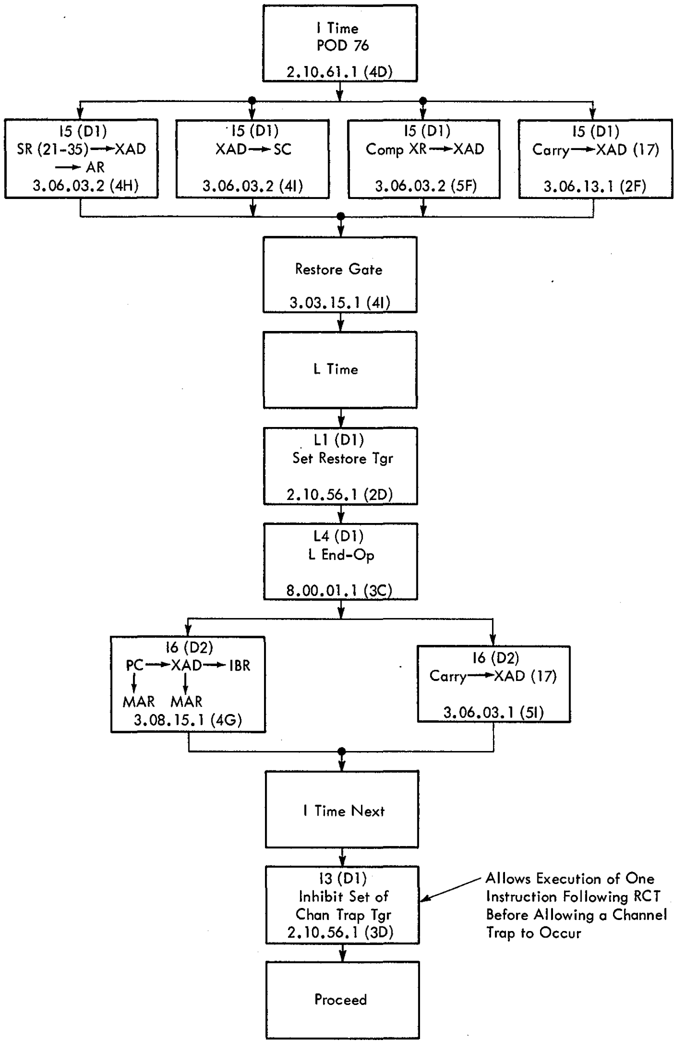


Figure 77. Restore Channel Traps (RCT)

Data Channel Trap Operation

A channel must be enabled and the restore trigger must be on to allow the channel to request priority to send a trap demand to CPU. See Figure 78.

A channel trap operation has three objectives:

1. To store the location of the next instruction to be executed (PC-1) in the address portion of the specified channel trap even address (0012_8 for channel A).

2. To store identification bits in the decrement portion of the channel trap even address.

3. Trap the computer to the specified channel trap odd address (0013_8 for channel A).

If a floating point trap or interrupt does not have priority and the instruction in the program register is not a read or write select, execute, enable or restore, the channel trap trigger is set at I3 (02.10.56.1-3A).

The normal storage bus to program register gating is blocked (03.14.00.1-3A); instead, the computer is forced into a STR operation (03.14.00.1-2A). CPU and channel are both sent to E time.

Channel E time is needed to set the buffer address register to the correct channel-trap even-address and to reset the trap priority for the channel.

At E6, the address register is reset; AR to MAR gating is blocked (03.08.15.3-4A). BAR is gated to MAR at E6 (03.08.15.3-4E) to supply memory with the channel-trap even-address. MF store address and MF store decrement are activated and the program counter is gated at E0 to the index adders with all ones. This subtracts one from the value of the program counter and supplies the location of the next instruction to be executed in the main program. Then, the index adders are gated to the storage register (21-35), and back to the program counter. The identifying bits for the decrement portion of the word to be stored are placed on the storage bus early in E time (02.10.59.1).

At E1, the storage register is gated to the storage bus. The location of the next instruction to be executed has been stored in the address portion of the channel-trap even address, and the trap identifying bits in the decrement portion (objectives one and two). The prefix and tag positions of the trap even address are not changed. The restore trigger is reset at E3.

Trapping to the channel trap odd address occurs by gating BAR to MAR with a hot bit to BAR bus 17 (03.08.15.3-3F) (objective three). Program counter and and index adder gating to MAR is blocked.

During the following I time, the channel trap trigger is reset at I2. The program counter is not updated and set of the IBR loaded trigger is blocked. If the instruction in the channel trap odd address is not a transfer, the program resumes where it was when the channel trap was initiated.

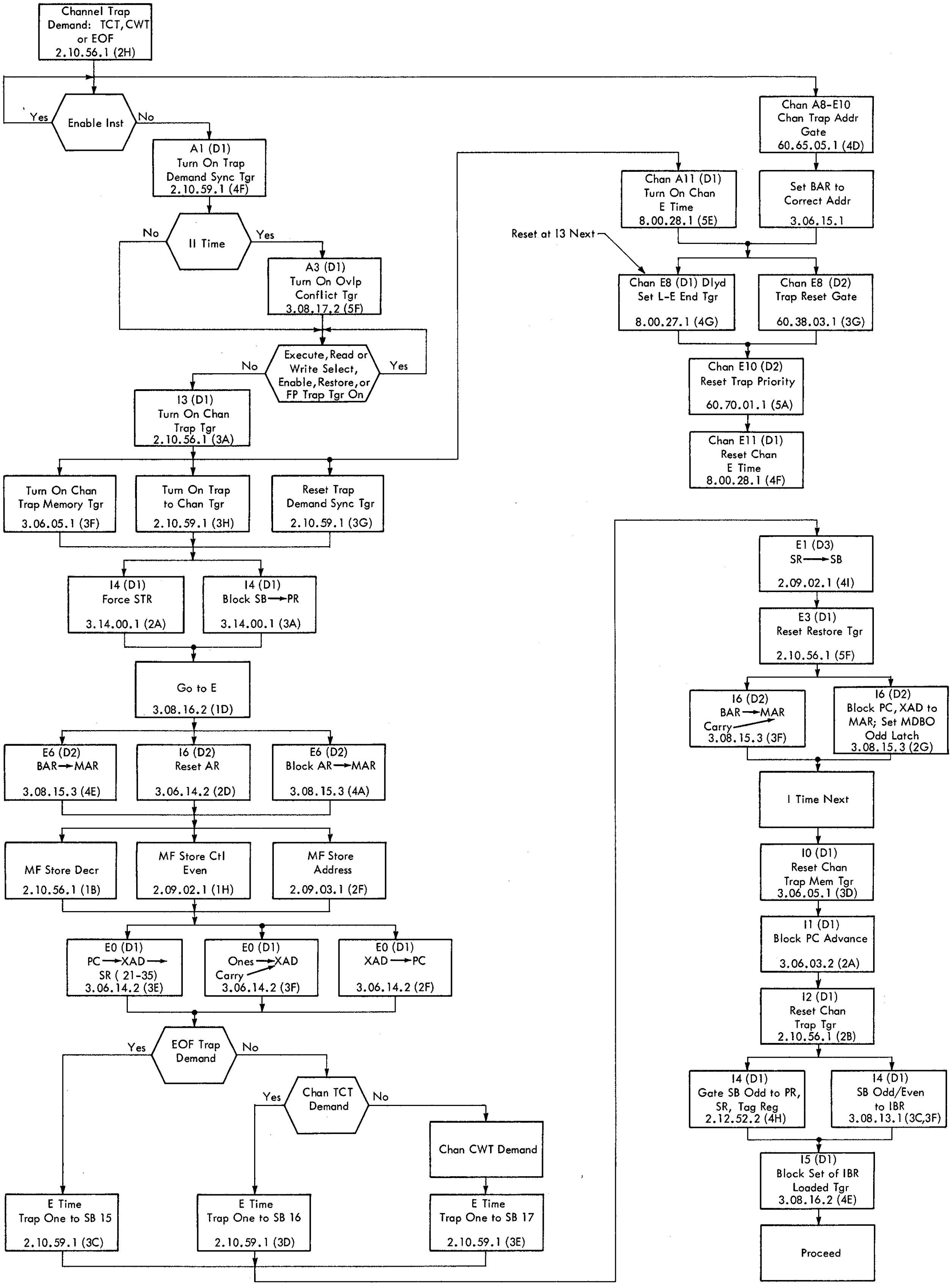

Channel Trap Demand: TCT, CWT or EOF 2.10.56.1 (2H)
Yes
Enable Inst
No
A1 (D1) Turn On Trap Demand Sync Tgr 2.10.59.1 (4F)
No
I1 Time
Yes
A3 (D1) Turn On Ovlp Conflict Tgr 3.08.17.2 (5F)
No
Execute, Read or Write Select, Enable, Restore, or FP Trap Tgr On
Yes
I3 (D1) Turn On Chan Trap Tgr 2.10.56.1 (3A)
Turn On Chan Trap Memory Tgr 3.06.05.1 (3F)
Turn On Trap to Chan Tgr 2.10.59.1 (3H)
Reset Trap Demand Sync Tgr 2.10.59.1 (3G)
I4 (D1) Force STR 3.14.00.1 (2A)
I4 (D1) Block SB→PR 3.14.00.1 (3A)
Go to E 3.08.16.2 (1D)
E6 (D2) BAR→MAR 3.08.15.3 (4E)
I6 (D2) Reset AR 3.06.14.2 (2D)
E6 (D2) Block AR→MAR 3.08.15.3 (4A)
MF Store Decr 2.10.56.1 (1B)
MF Store Ctl Even 2.09.02.1 (1H)
MF Store Address 2.09.03.1 (2F)
E0 (D1) PC→XAD→ SR (21-35) 3.06.14.2 (3E)
E0 (D1) Ones→XAD Carry 3.06.14.2 (3F)
E0 (D1) XAD→PC 3.06.14.2 (2F)
Yes
EOF Trap Demand
No
Yes
Chan TCT Demand
No
Chan CWT Demand
E Time Trap One to SB 15 2.10.59.1 (3C)
E Time Trap One to SB 16 2.10.59.1 (3D)
E Time Trap One to SB 17 2.10.59.1 (3E)
Chan A8-E10 Chan Trap Addr Gate 60.65.05.1 (4D)
Chan A11 (D1) Turn On Chan E Time 8.00.28.1 (5E)
Set BAR to Correct Addr 3.06.15.1
Reset at I3 Next
Chan E8 (D1) Dlyd Set L-E End Tgr 8.00.27.1 (4G)
Chan E8 (D2) Trap Reset Gate 60.38.03.1 (3G)
Chan E10 (D2) Reset Trap Priority 60.70.01.1 (5A)
Chan E11 (D1) Reset Chan E Time 8.00.28.1 (4F)
E1 (D3) SR→SB 2.09.02.1 (4I)
E3 (D1) Reset Restore Tgr 2.10.56.1 (5F)
I6 (D2) BAR→MAR Carry 3.08.15.3 (3F)
I6 (D2) Block PC, XAD to MAR; Set MDBO Odd Latch 3.08.15.3 (2G)
I Time Next
I0 (D1) Reset Chan Trap Mem Tgr 3.06.05.1 (3D)
I1 (D1) Block PC Advance 3.06.03.2 (2A)
I2 (D1) Reset Chan Trap Tgr 2.10.56.1 (2B)
I4 (D1) Gate SB Odd to PR, SR, Tag Reg 2.12.52.2 (4H)
I4 (D1) SB Odd/Even to IBR 3.08.13.1 (3C, 3F)
I5 (D1) Block Set of IBR Loaded Tgr 3.08.16.2 (4E)
Proceed

Figure 78. Data Channel Trap Operation

Channel Instructions

I/O operations on the 7094 II, except for minor changes, are the same as for the 7094. In order to avoid major changes to the 7607 Data Channel (as it could not operate at 1.4 μs without extensive rework), the 12-cycle point clock is retained for the channels. Use of two clocks, one for CPU and one for channel, requires that the two clocks be synchronized and that some method be provided to synchronize channel and CPU cycles.

The two clocks are separate, but under control of the same oscillator and logic circuits. They are started at the same time: on a power on reset, or on depression of the clear key. Two channel cycles occur for three CPU cycles, so that both clocks are at zero time simultaneously every third CPU cycle (Figure 79).

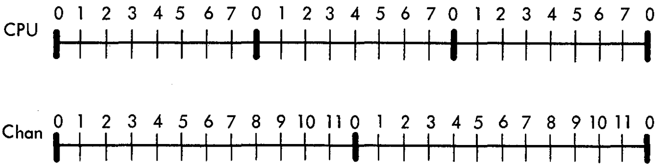


Figure 79. CPU and Channel Clock Synchronization

If, because of some malfunction, the clocks lose synchronization, the multi-time error trigger turns on. A check to see that a Ch A9 pulse lines up with a CPU A1 or A5 pulse (Systems 08.00.22.2) will determine correct operation.

Generally, when an instruction requires a Ch E or Ch L cycle, or both, CPU is forced to hang up in L time. The exception to this occurs during a channel trap operation when a Ch E cycle is taken without stopping CPU. The MF go trigger and the Ch L-E end trigger are the principle synchronizing factors in the control of channel and CPU cycles. The MF go trigger is reset at L4 of the instruction requiring a channel cycle (may be Ch E0 for an SCH instruction), and is set after the Ch L-E end trigger has been turned on. See individual flow charts for exact timings. CPU will not end operation until the MF go trigger is set.

The Ch L-E end trigger is set by the channel end-op ctl tgr for RDS, WRS, SPU, SPR and all tape instructions at Ch L10. It is set at Ch E8 for RCH, LCH, and SCH, and at Ch L8 for BTT, ETT, TEF, and TRC.

A few instructions require channel I-cycles. These I cycles are used almost exclusively to perform resets and are therefore short I cycles. The channel I-time trigger is reset at Ch A6 (08.00.27.1).

The following flow charts and descriptions show primarily the synchronization between channel and CPU cycle times. For detailed information regarding the effect of these instructions on the channel, refer to *7607 Data Channel, CEIR,* Form 223-6910.

Tape Data and Non-Data Selects

Read Select RDS + 0762 I, L − L (Ch L). See Figure 80.

Write Select WRS + 0766 I, L − L (Ch L). See Figure 80.

Backspace File BSF − 0764 I, L − L (Ch L). See Figure 80.

Backspace Record BSR + 0764 I, L − L (Ch L). See Figure 80.

Rewind REW + 0772 I, L − L (Ch L). See Figure 80.

Rewind and Unload RUN − 0772 I, L − L (Ch L). See Figure 80.

Set Density SDN + 0776 I, L – L (Ch L). See Figure 80.

Write End of File WEF + 0770 I, L – L (Ch L). See Figure 80.

The preceding instructions select a tape unit in the specified channel to perform their required operation. Storage register (23-26) determines the channel to be selected. Shift counter (10-13) indicates class address and mode of operation, and shift counter (14-17) determines the unit address.

At L4, the MF go trigger is reset, which sets the L-end-op sync trigger (08.00.02.2-5C). The first Ch A11 pulse not occurring during I time sets the channel L-time trigger. A Ch L3 pulse sets the end-op ctl trigger, and at Ch L10 the Ch L-E end trigger is set. The Ch L-E end trigger on sets the MF go trigger which allows CPU to end operation.

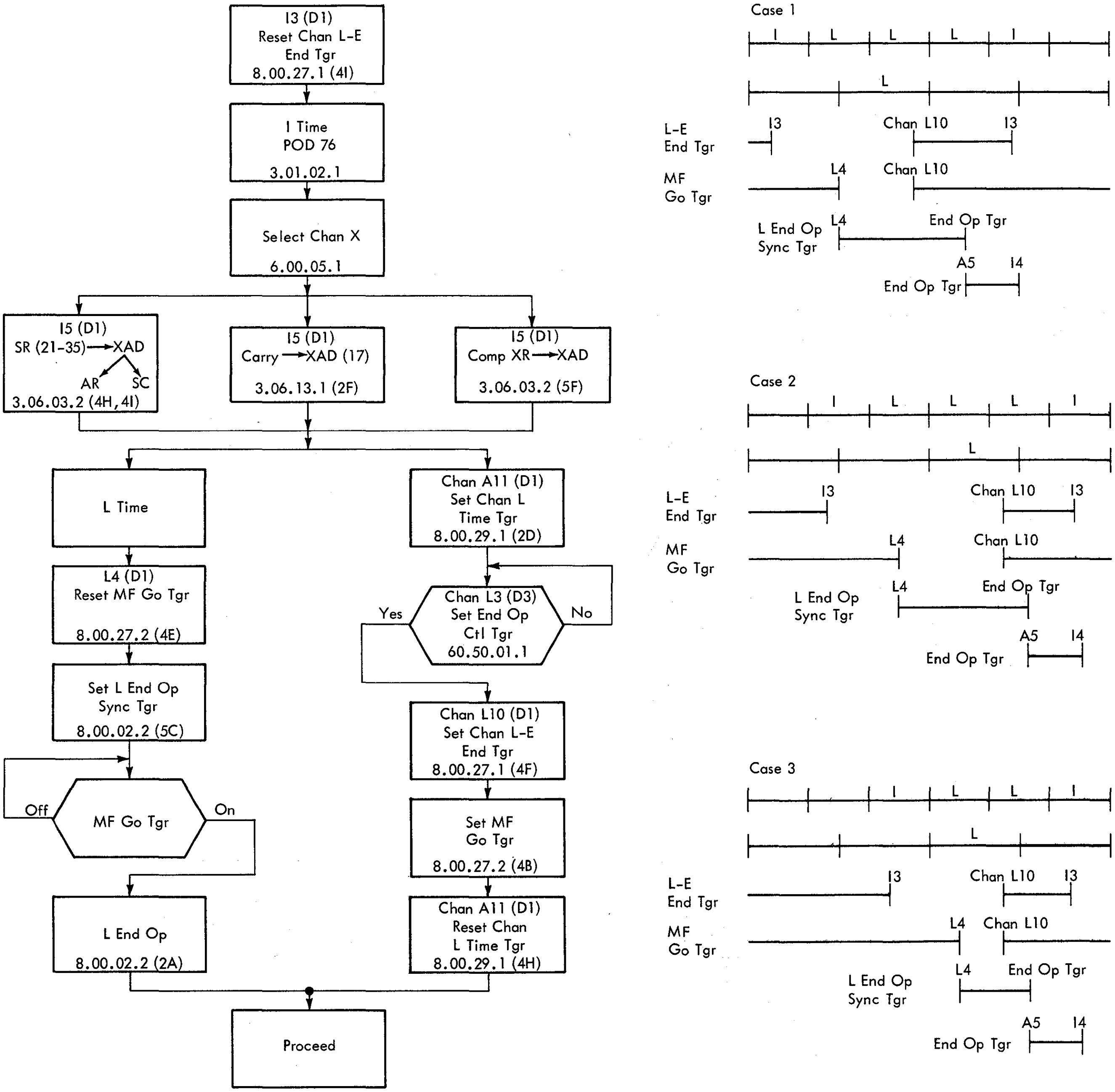


Figure 80. Read Select (RDS); Write Select (WRS); Backspace File (BSF); Backspace Record (BSR); Rewind (REW); Rewind and Unload (RUN); Set Density (SDN); Write End of File (WEF)

Card Machine Select

Read Select RDS + 0762 I, L-L (Ch L, I)

See Figure 81.

Write Select WRS + 0766 I, L-L (Ch L, I)

The preceding instructions select the reader, punch, or printer in the specified channel to perform the required operation. Storage register (23-26) determines the channel to be selected. Shift counter (10-13) indicates class address, and shift counter (14-17) determines the unit address and the mode of operation for the printer. See Figure 81.

At L4 the MF go trigger is reset; this sets the L-end-op sync trigger (08.00.02.2-5C). The first Ch A11 pulse not occurring during I time sets the channel L time trigger. A Ch L3 pulse sets the end-op ctl trigger, and at Ch L10 the Ch L-E end trigger is set. Channel I-time

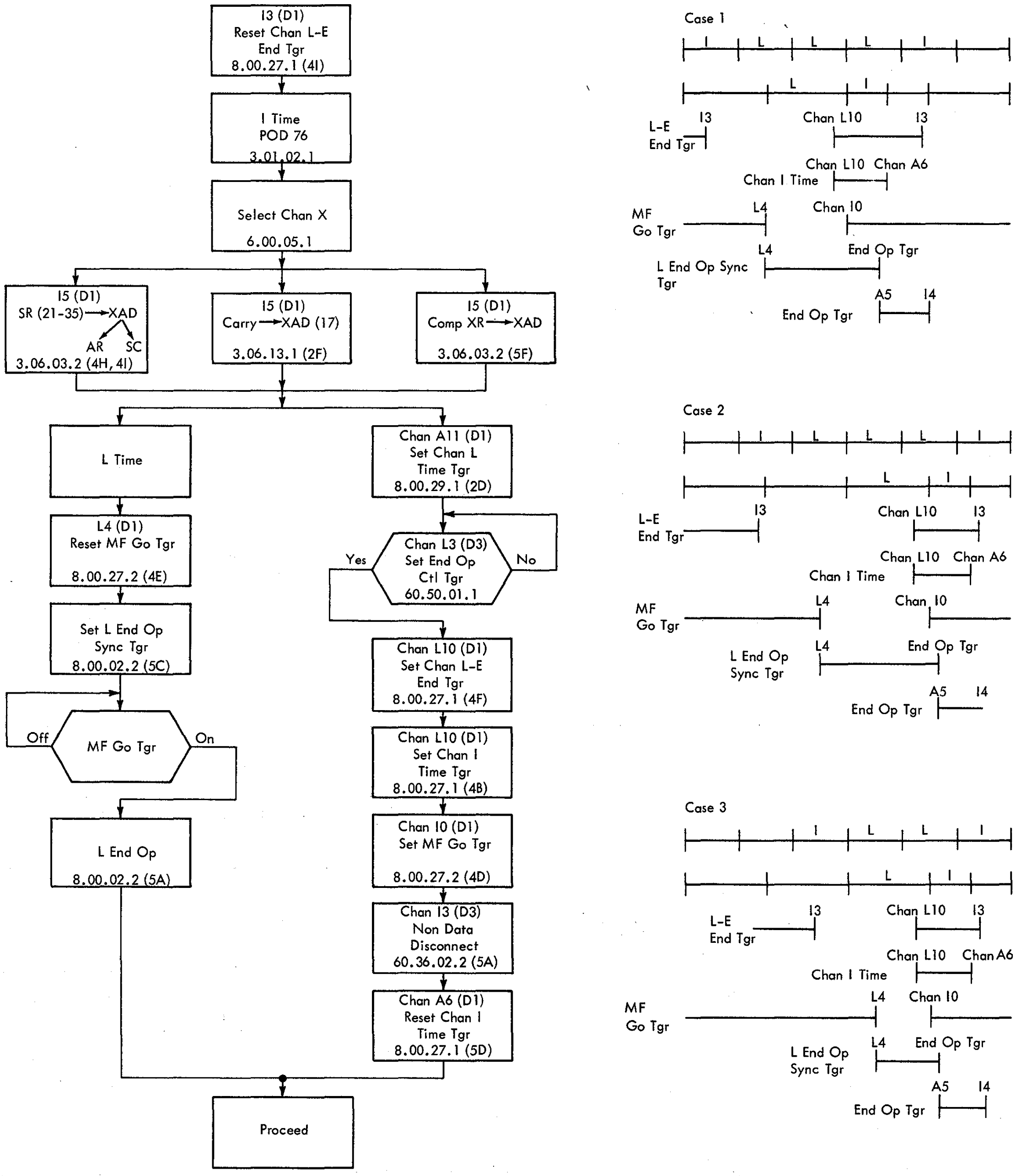


Figure 81. Read Select/Write Select Cards

provides a nondata disconnect, which frees the class and unit triggers for a sense instruction. The MF go trigger is set at Ch I0, allowing CPU to end operation.

Store Channel SCHX± I, L-L (Ch E)

This instruction stores the contents of the channel indicators, location counter, and address counter in the specified storage location. The indicators are stored in positions S, 1, 2, 19, location counter in positions 3-17, and address counter in positions 21-35 of the storage location. See Figure 82.

The MF go trigger is reset at L4 or at Ch E0, whichever comes first. Resetting the MF go trigger sets the L-end-op sync trigger. At Ch A11, the channel E-time

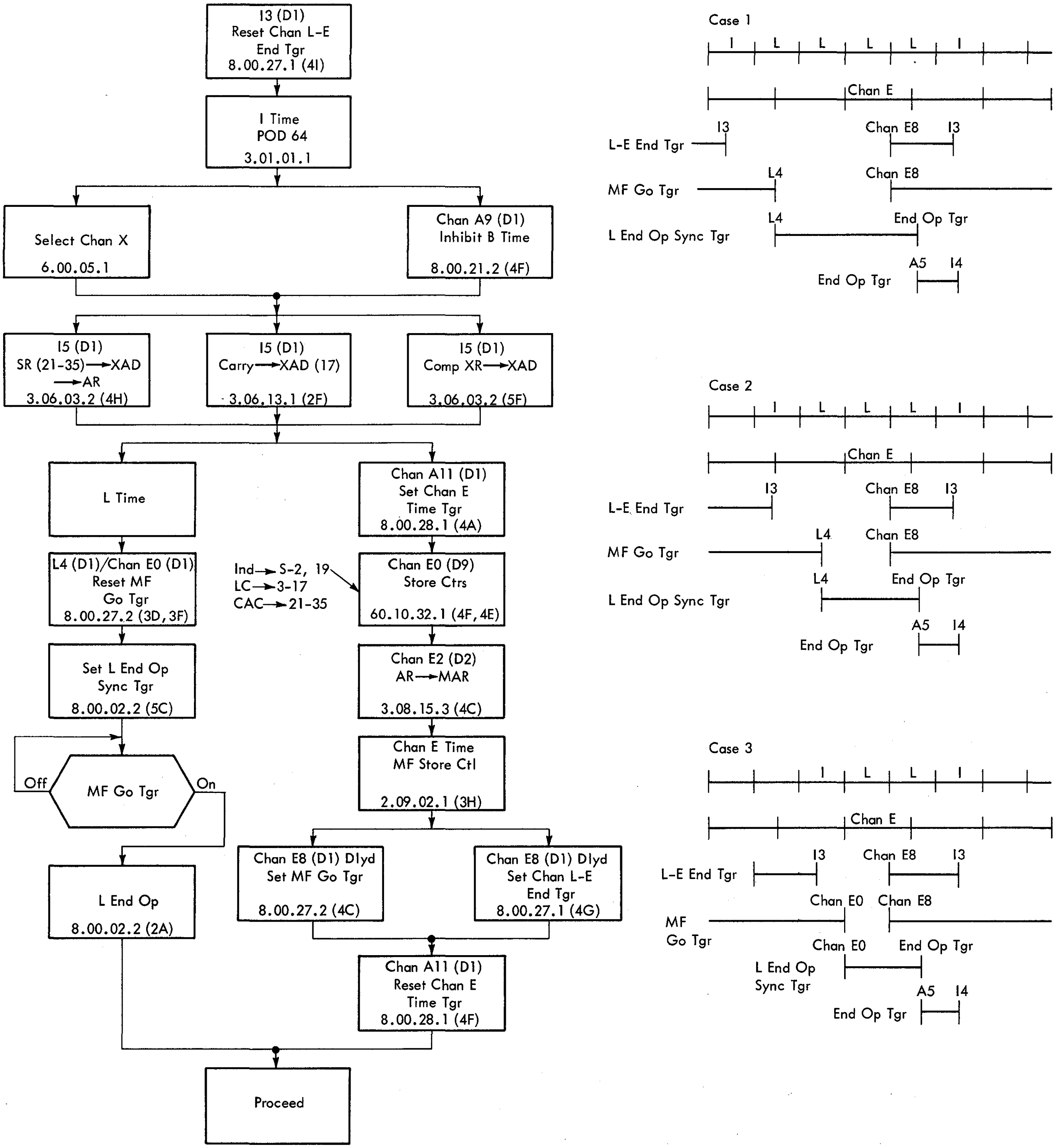


Figure 82. Store Channel (SCHX)

trigger is set. "Store counters" is energized at Ch E0 (60.10.32.1), placing the contents of the counters on the channel storage bus. At Ch E2, the address register is gated to MAR, and at Ch E8 the Ch L-E end trigger and the MF go trigger are set. The MF go trigger ON allows CPU to end operation.

Transfer on Data Channel End-of-File TEFX ± 003X I, L-L (Ch L, I)

See Figure 83.

Transfer on Data Channel Redundancy Check TRCX ± 002X I, L-L (Ch L, I)

The preceding instructions test the EOF trigger and the rdn chk trigger respectively. If the tested trigger is on, the computer transfers to the location specified in positions 21-35 of the instruction. If the trigger is off, the computer takes the next sequential instruction. See Figure 83.

The MF go trigger and L-end-op sync trigger are set at L4. The channel L-time trigger is set at Ch A11. At the start of the channel L-cycle the specified trigger, EOF or Rdn Chk, is tested. If the trigger is on, the ind sync trigger is set at Ch L1 (60.32.03.1), and at L5 the PR-CM-tgr is set. The Ch L-E end trigger is set at Ch L8, and at Ch A10, the I-time trigger is set. During channel I-time the MF go trigger is set and the ind sync trigger and tested trigger are reset. Setting the MF go trigger allows CPU to end operation.

Beginning of Tape Test BTT + 0760 I, L-L (Ch L, I)

See Figure 84.

End of Tape Test ETT − 0760 I, L-L (Ch L, I)

The preceding instructions test the BOT trigger and the EOT trigger respectively. If the tested trigger is off, the computer skips the next sequential instruction. If the tested trigger is on the computer takes the next sequential instruction. See Figure 84.

At L3 and L4 the skip addresses are formed (02.09.-41.1) and placed in the address register. The MF go trigger is reset, and L-end-op sync trigger set at L4. At the start of the channel L-cycle the specified trigger, BOT or EOT, is tested. If the trigger is on, the ind sync trigger is set and inhibits "I/O sense skip," (02.10.80.1-3G). If the BOT or EOT triggers are not on, "I/O sense skip" is energized. The Ch L-E end trigger is set at Ch L8, and at Ch L10 the channel I-time trigger is set. During channel I time, the MF go trigger is set, BOT or EOT trigger and the ind sync trigger reset. Setting the MF go trigger allows CPU to end operation.

Reset and Load Channel RCH ± 054X I, L-L (Ch L, E)

See Figure 85.

Load Channel LCH ± 054X I, L-L (Ch L, E)

The preceding instructions provide a storage reference location and force channel L and E cycles in order to provide channel with a command word. Positions 21-35 of this instruction contain the address of the command word. This address is placed in the channel location counter. S, 1, 2 and 19 of the command word are placed in the channel indicators. Positions 21-35 of the command word are placed in the channel address counter and 3-17 in the channel word counter. See Figure 85.

During the I cycle (I5) the location of the command word is routed to the address register. At L4, the MF go trigger is reset; this sets the L-end-op sync trigger (08.00.02.2-5C). A Ch A11 pulse sets the channel L-time trigger. The address register is gated through the index adders to the storage register and then to the storage bus. The storage bus is gated to the BAR output, and the BAR gated to the channel location counter at Ch E0. This places the location of the command word in the channel location counter. The proceed-to-E trigger is set at Ch L3 (60.80.04.1-3E) and the channel E-time trigger at Ch A11. At Ch E2, the address register is gated to MAR and at Ch E7, the command word is on the storage bus. The positions of the command word are routed to the channel registers as previously indicated. The Ch L-E end trigger and the MF go trigger are set at Ch E8. Setting the MF go trigger allows CPU to end operation.

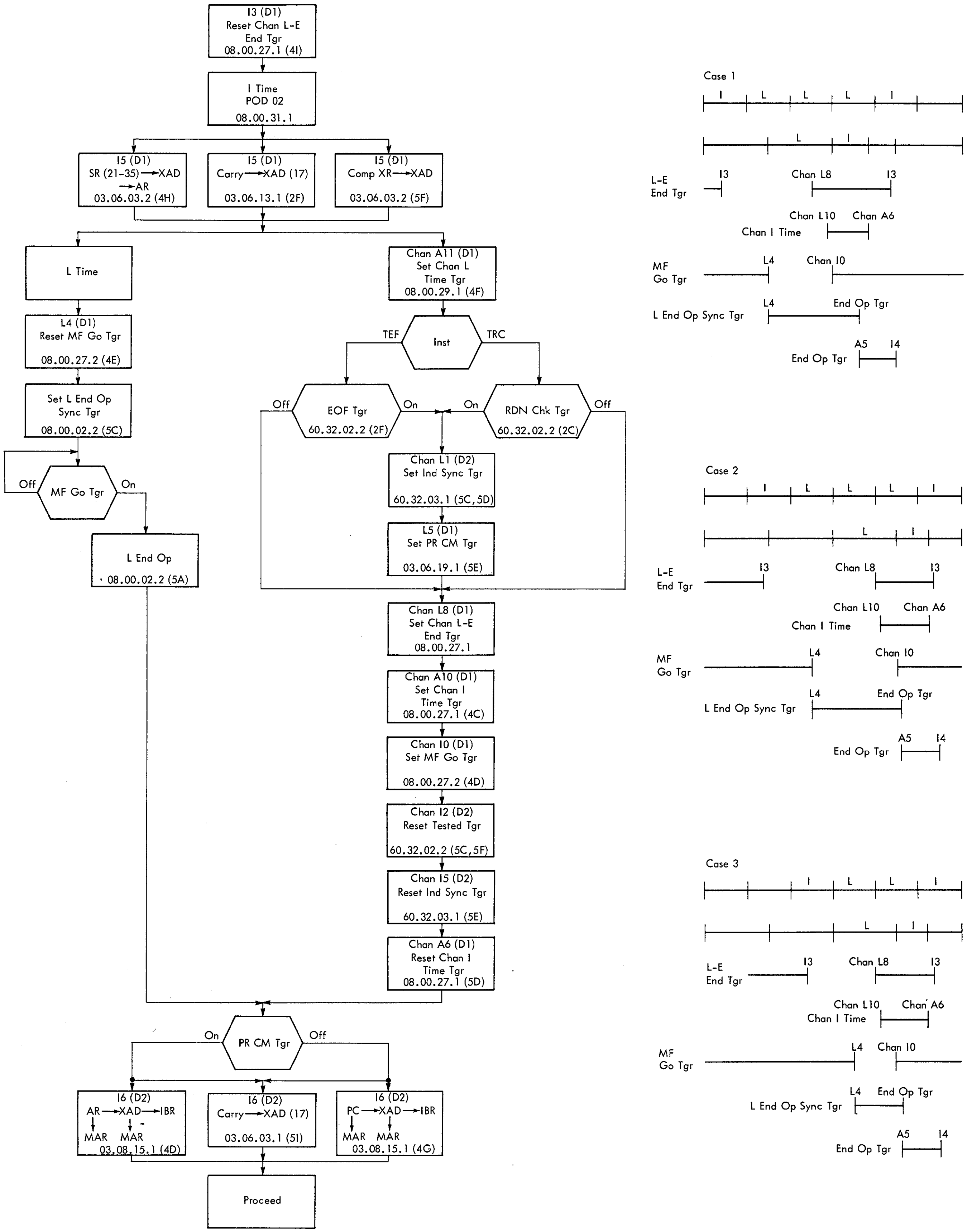
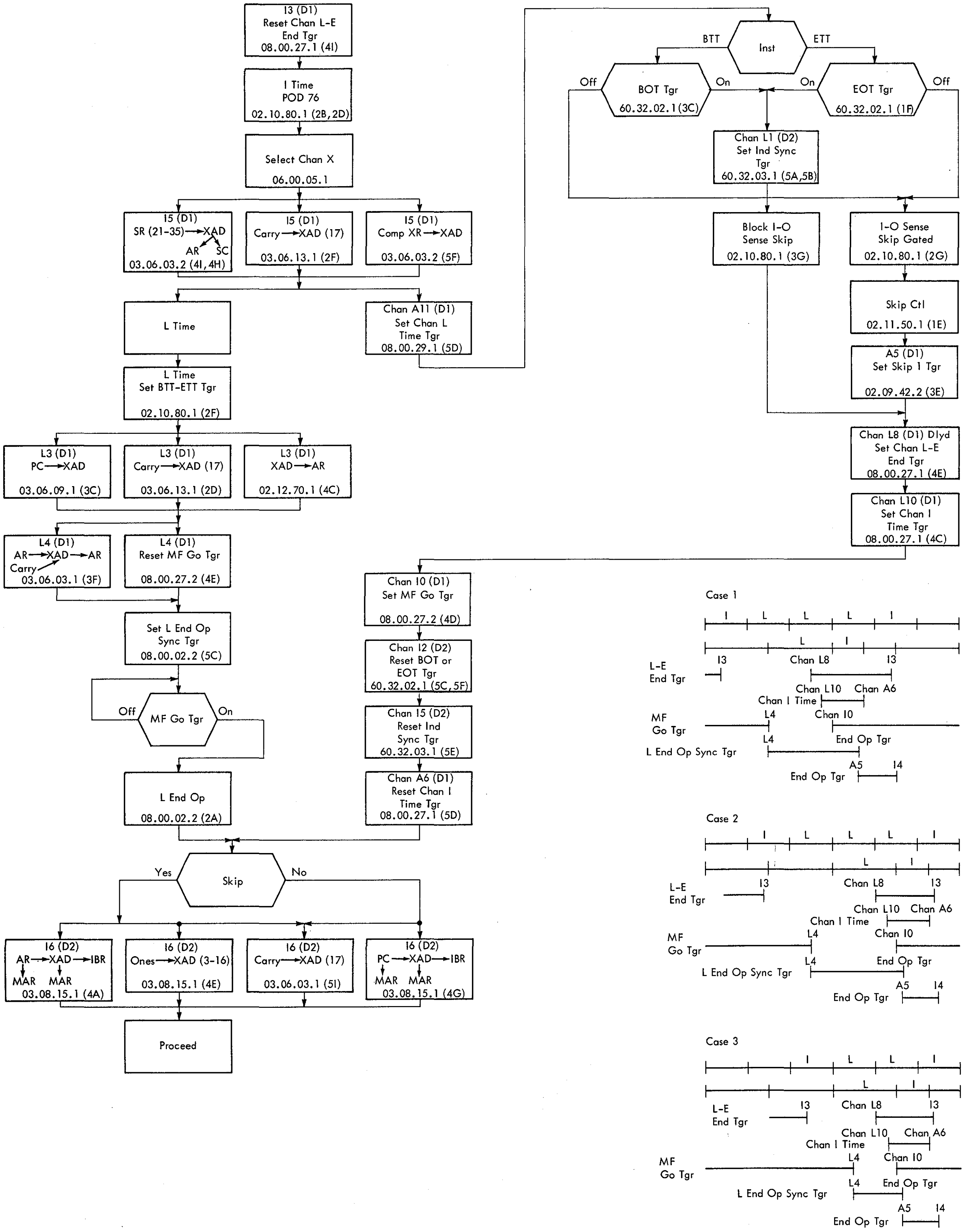

I3 (D1) Reset Chan L-E End Tgr 08.00.27.1 (4I)
I Time POD 02 08.00.31.1
I5 (D1) SR (21-35)→XAD →AR 03.06.03.2 (4H)
I5 (D1) Carry→XAD (17) 03.06.13.1 (2F)
I5 (D1) Comp XR→XAD 03.06.03.2 (5F)
L Time
Chan A11 (D1) Set Chan L Time Tgr 08.00.29.1 (4F)
L4 (D1) Reset MF Go Tgr 08.00.27.2 (4E)
TEF
Inst
TRC
Set L End Op Sync Tgr 08.00.02.2 (5C)
Off
EOF Tgr 60.32.02.2 (2F)
On
On
RDN Chk Tgr 60.32.02.2 (2C)
Off
Off
MF Go Tgr
On
Chan L1 (D2) Set Ind Sync Tgr 60.32.03.1 (5C,5D)
L5 (D1) Set PR CM Tgr 03.06.19.1 (5E)
L End Op 08.00.02.2 (5A)
Chan L8 (D1) Set Chan L-E End Tgr 08.00.27.1
Chan A10 (D1) Set Chan I Time Tgr 08.00.27.1 (4C)
Chan I0 (D1) Set MF Go Tgr 08.00.27.2 (4D)
Chan I2 (D2) Reset Tested Tgr 60.32.02.2 (5C,5F)
Chan I5 (D2) Reset Ind Sync Tgr 60.32.03.1 (5E)
Chan A6 (D1) Reset Chan I Time Tgr 08.00.27.1 (5D)
On
PR CM Tgr
Off
I6 (D2) AR→XAD→IBR MAR MAR 03.08.15.1 (4D)
I6 (D2) Carry→XAD (17) 03.06.03.1 (5I)
I6 (D2) PC→XAD→IBR MAR MAR 03.08.15.1 (4G)
Proceed
Case 1
I L L L I
L I
L-E End Tgr
I3
Chan L8
I3
Chan L10
Chan A6
Chan I Time
MF Go Tgr
L4
Chan I0
L End Op Sync Tgr
L4
End Op Tgr
A5
I4
End Op Tgr
Case 2
I L L L I
L I
L-E End Tgr
I3
Chan L8
I3
Chan L10
Chan A6
Chan I Time
MF Go Tgr
L4
Chan I0
L4
End Op Tgr
L End Op Sync Tgr
A5
I4
End Op Tgr
Case 3
I L L I
L I
L-E End Tgr
I3
Chan L8
I3
Chan L10
Chan A6
Chan I Time
MF Go Tgr
L4
Chan I0
L4
End Op Tgr
L End Op Sync Tgr
A5
I4
End Op Tgr

Figure 83. Transfer on Data Channel End of File (TEFX); Transfer on Data Channel Redundancy Check (TRCX)



I3 (D1) Reset Chan L-E End Tgr 08.00.27.1 (4I)
I Time POD 76 02.10.80.1 (2B,2D)
Select Chan X 06.00.05.1
I5 (D1) SR (21-35)→XAD AR SC 03.06.03.2 (4I,4H)
I5 (D1) Carry→XAD (17) 03.06.13.1 (2F)
I5 (D1) Comp XR→XAD 03.06.03.2 (5F)
L Time
Chan A11 (D1) Set Chan L Time Tgr 08.00.29.1 (5D)
L Time Set BTT-ETT Tgr 02.10.80.1 (2F)
L3 (D1) PC→XAD 03.06.09.1 (3C)
L3 (D1) Carry→XAD (17) 03.06.13.1 (2D)
L3 (D1) XAD→AR 02.12.70.1 (4C)
L4 (D1) AR→XAD→AR Carry 03.06.03.1 (3F)
L4 (D1) Reset MF Go Tgr 08.00.27.2 (4E)
Set L End Op Sync Tgr 08.00.02.2 (5C)
Off
MF Go Tgr
On
L End Op 08.00.02.2 (2A)
Yes
Skip
No
I6 (D2) AR→XAD→IBR MAR MAR 03.08.15.1 (4A)
I6 (D2) Ones→XAD (3-16) 03.08.15.1 (4E)
I6 (D2) Carry→XAD (17) 03.06.03.1 (5I)
I6 (D2) PC→XAD→IBR MAR MAR 03.08.15.1 (4G)
Proceed
BTT
Inst
ETT
Off
BOT Tgr 60.32.02.1 (3C)
On
On
EOT Tgr 60.32.02.1 (1F)
Off
Chan L1 (D2) Set Ind Sync Tgr 60.32.03.1 (5A,5B)
Block I-O Sense Skip 02.10.80.1 (3G)
I-O Sense Skip Gated 02.10.80.1 (2G)
Skip Ctl 02.11.50.1 (1E)
A5 (D1) Set Skip 1 Tgr 02.09.42.2 (3E)
Chan L8 (D1) Dlyd Set Chan L-E End Tgr 08.00.27.1 (4E)
Chan L10 (D1) Set Chan I Time Tgr 08.00.27.1 (4C)
Chan I0 (D1) Set MF Go Tgr 08.00.27.2 (4D)
Chan I2 (D2) Reset BOT or EOT Tgr 60.32.02.1 (5C,5F)
Chan I5 (D2) Reset Ind Sync Tgr 60.32.03.1 (5E)
Chan A6 (D1) Reset Chan I Time Tgr 08.00.27.1 (5D)
Case 1
I L L L I
L I
L-E End Tgr
I3
Chan L8
I3
Chan L10
Chan A6
Chan I Time
MF Go Tgr
L4
Chan I0
L4
End Op Tgr
L End Op Sync Tgr
A5
I4
End Op Tgr
Case 2
I L L L I
L I
L-E End Tgr
I3
Chan L8
I3
Chan L10
Chan A6
Chan I Time
MF Go Tgr
L4
Chan I0
L4
End Op Tgr
L End Op Sync Tgr
A5
I4
End Op Tgr
Case 3
I L L I
L I
L-E End Tgr
I3
Chan L8
I3
Chan L10
Chan A6
Chan I Time
MF Go Tgr
L4
Chan I0
L4
End Op Tgr
L End Op Sync Tgr
A5
I4
End Op Tgr

Figure 84. Beginning of Tape Test (BTT); End of Tape Test (ETT)

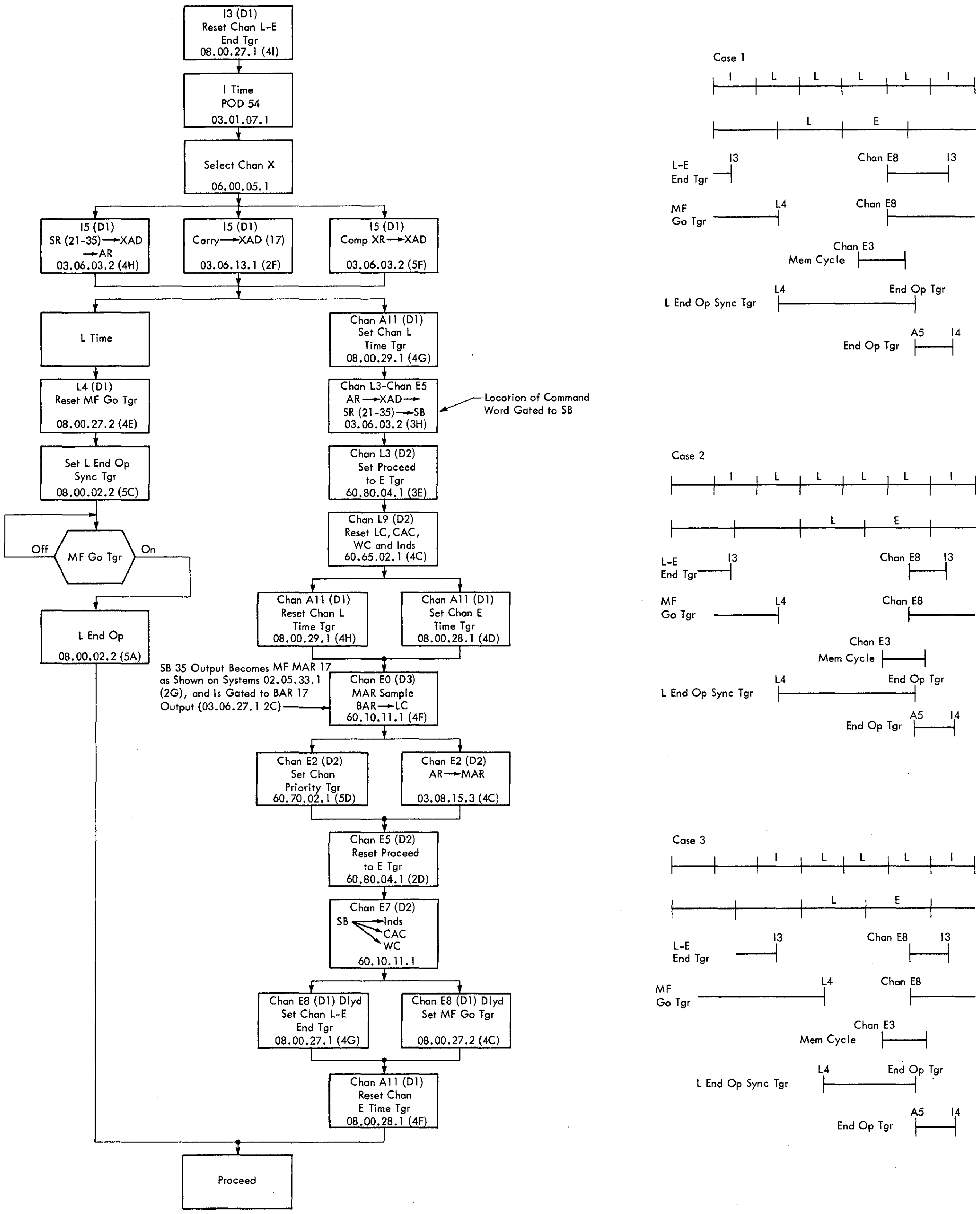

I3 (D1) Reset Chan L-E End Tgr 08.00.27.1 (4I)
I Time POD 54 03.01.07.1
Select Chan X 06.00.05.1
I5 (D1) SR (21-35)→XAD →AR 03.06.03.2 (4H)
I5 (D1) Carry→XAD (17) 03.06.13.1 (2F)
I5 (D1) Comp XR→XAD 03.06.03.2 (5F)
L Time
Chan A11 (D1) Set Chan L Time Tgr 08.00.29.1 (4G)
L4 (D1) Reset MF Go Tgr 08.00.27.2 (4E)
Chan L3-Chan E5 AR→XAD→ SR (21-35)→SB 03.06.03.2 (3H)
Location of Command Word Gated to SB
Set L End Op Sync Tgr 08.00.02.2 (5C)
Chan L3 (D2) Set Proceed to E Tgr 60.80.04.1 (3E)
Off
MF Go Tgr
On
Chan L9 (D2) Reset LC,CAC, WC and Inds 60.65.02.1 (4C)
L End Op 08.00.02.2 (5A)
Chan A11 (D1) Reset Chan L Time Tgr 08.00.29.1 (4H)
Chan A11 (D1) Set Chan E Time Tgr 08.00.28.1 (4D)
SB 35 Output Becomes MF MAR 17 as Shown on Systems 02.05.33.1 (2G), and Is Gated to BAR 17 Output (03.06.27.1 2C)
Chan E0 (D3) MAR Sample BAR→LC 60.10.11.1 (4F)
Chan E2 (D2) Set Chan Priority Tgr 60.70.02.1 (5D)
Chan E2 (D2) AR→MAR 03.08.15.3 (4C)
Chan E5 (D2) Reset Proceed to E Tgr 60.80.04.1 (2D)
Chan E7 (D2) SB→Inds →CAC →WC 60.10.11.1
Chan E8 (D1) Dlyd Set Chan L-E End Tgr 08.00.27.1 (4G)
Chan E8 (D1) Dlyd Set MF Go Tgr 08.00.27.2 (4C)
Chan A11 (D1) Reset Chan E Time Tgr 08.00.28.1 (4F)
Proceed
Case 1
I L L L L I
L E
L-E End Tgr
I3
Chan E8
I3
MF Go Tgr
L4
Chan E8
Chan E3 Mem Cycle
L End Op Sync Tgr
L4
End Op Tgr
End Op Tgr
A5
I4
Case 2
I L L L L I
L E
L-E End Tgr
I3
Chan E8
I3
MF Go Tgr
L4
Chan E8
Chan E3 Mem Cycle
L End Op Sync Tgr
L4
End Op Tgr
End Op Tgr
A5
I4
Case 3
I L L L I
L E
L-E End Tgr
I3
Chan E8
I3
MF Go Tgr
L4
Chan E8
Chan E3 Mem Cycle
L End Op Sync Tgr
L4
End Op Tgr
End Op Tgr
A5
I4

Figure 85. Reset and Load Channel (RCH); Load Channel (LCH)

Sense Printer **SPR + 0760 . . . XXXX I, L-L (Ch L, I)**

See Figure 86.

Sense Punch **SPU + 0760 . . . XXXX I, L-L (Ch L, I)**

The preceding instructions cause an impulse to appear at the specified sense-exit hub of their respective card machine control panel attached to the specified data channel. For the printer, addresses 361-372 specify sense-exits 1-10 and 1XXX-10XXX specify channels A-H. For the punch, addresses 341 and 342 specify sense-exits one and two respectively and 1XXX-10XXX specify channels A-H. The printer or punch must be selected for the preceding instructions to operate. The instructions will act like a NOP if their respective card machine is not selected. See Figure 86.

The MF go trigger is reset at L4, setting the L-end-op sync trigger. If the correct card machine is selected, the end op ctl trigger (60.50.01.1) is set in channel and the impulse gated to the specified sense-exit hub (80.50.06.1). The channel L-time trigger is set at Ch A11, and at Ch L8 the Ch L-E end trigger is set. Channel I-time is set at Ch L10, and the MF go trigger is set at Ch I0. The MF go trigger ON allows CPU to end operation. Channel I time is reset at Ch A6.

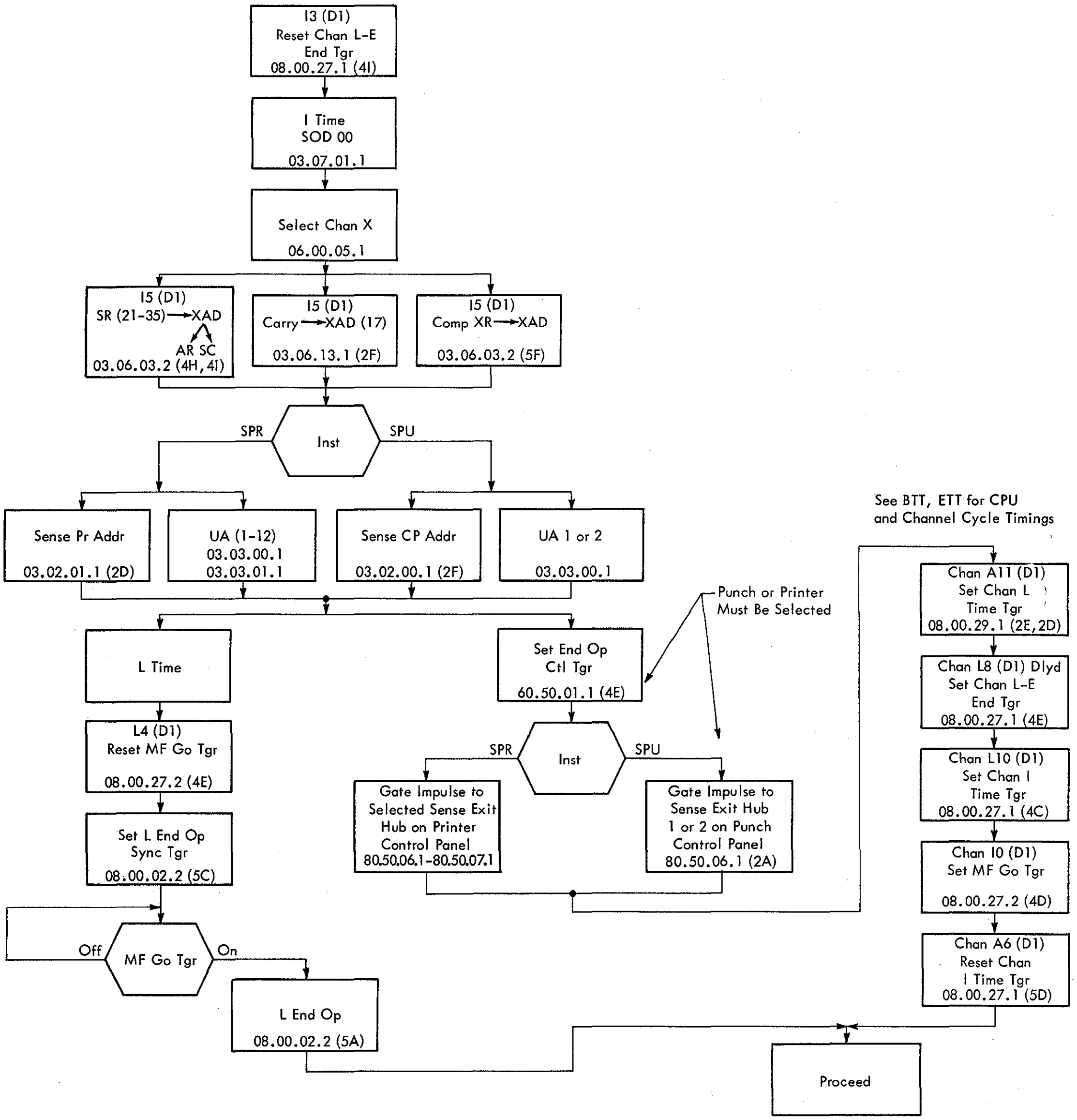
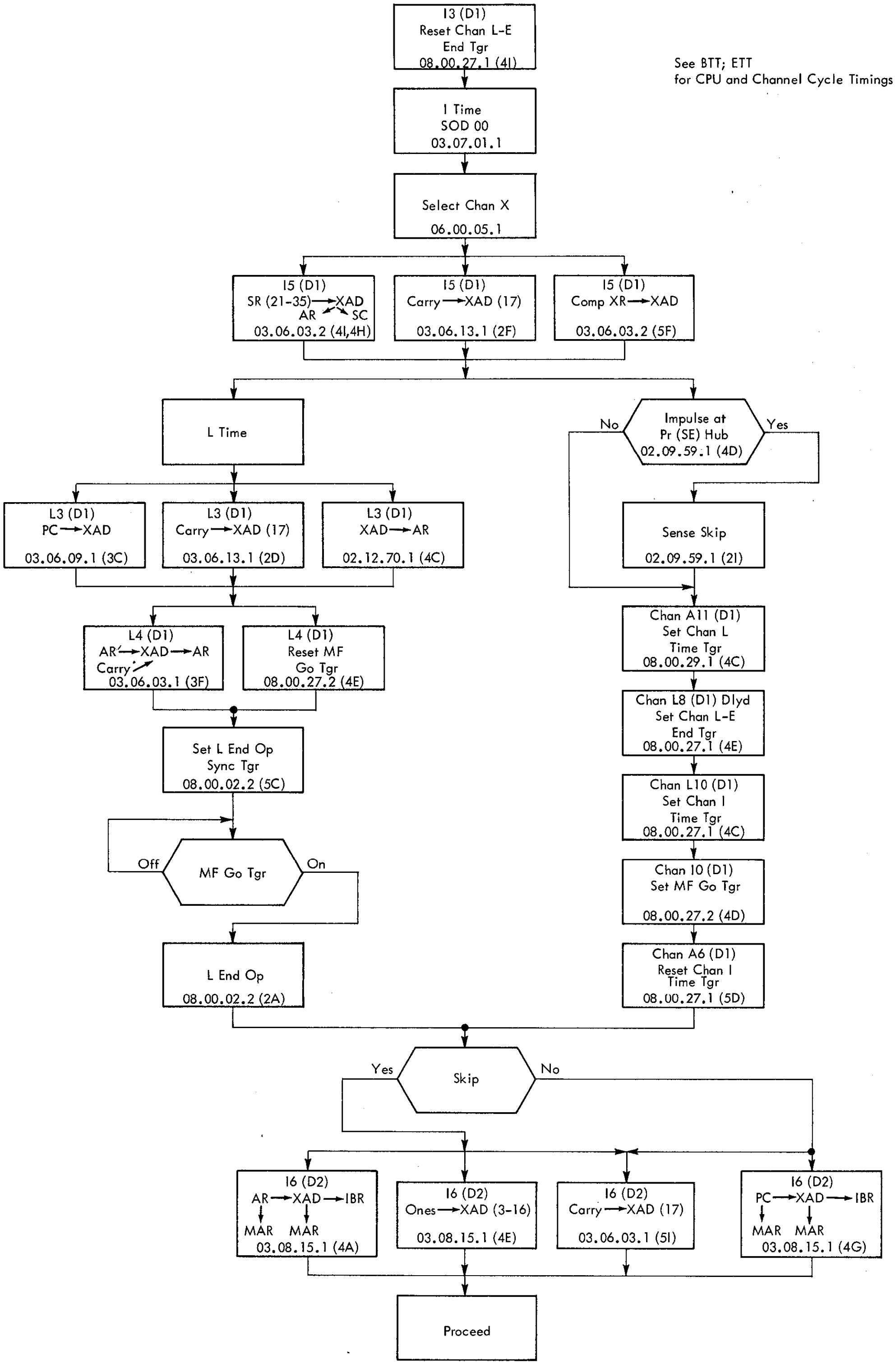


Figure 86. Sense Printer (SPR); Sense Punch (SPU)

Sense Printer Test
SPT + 0760 . . . XXXX I, L-L (Ch L, I)

If there is a signal at the sense entry hub on the printer's control panel of the specified data channel, the computer skips the next instruction. If there is no signal, the computer takes the next instruction in sequence. Addresses 1360 through 10360 specify data channels A through H respectively. See Figure 87.

Skip one and two addresses are formed at L3 and L4 time respectively. The MF go trigger is reset at L4, setting the L-end-op sync trigger. If an impulse is available at the printer sense entry hub, "sense skip" is activated (02.09.59.1-2I). The channel L-time trigger is set



I3 (D1) Reset Chan L-E End Tgr 08.00.27.1 (4I)
I Time SOD 00 03.07.01.1
Select Chan X 06.00.05.1
I5 (D1) SR (21-35)→XAD AR ↙↘ SC 03.06.03.2 (4I,4H)
I5 (D1) Carry→XAD (17) 03.06.13.1 (2F)
I5 (D1) Comp XR→XAD 03.06.03.2 (5F)
L Time
L3 (D1) PC→XAD 03.06.09.1 (3C)
L3 (D1) Carry→XAD (17) 03.06.13.1 (2D)
L3 (D1) XAD→AR 02.12.70.1 (4C)
L4 (D1) AR→XAD→AR Carry↗ 03.06.03.1 (3F)
L4 (D1) Reset MF Go Tgr 08.00.27.2 (4E)
Set L End Op Sync Tgr 08.00.02.2 (5C)
Off
MF Go Tgr
On
L End Op 08.00.02.2 (2A)
No
Impulse at Pr (SE) Hub 02.09.59.1 (4D)
Yes
Sense Skip 02.09.59.1 (2I)
Chan A11 (D1) Set Chan L Time Tgr 08.00.29.1 (4C)
Chan L8 (D1) Dlyd Set Chan L-E End Tgr 08.00.27.1 (4E)
Chan L10 (D1) Set Chan I Time Tgr 08.00.27.1 (4C)
Chan I0 (D1) Set MF Go Tgr 08.00.27.2 (4D)
Chan A6 (D1) Reset Chan I Time Tgr 08.00.27.1 (5D)
Yes
Skip
No
I6 (D2) AR→XAD→IBR MAR MAR 03.08.15.1 (4A)
I6 (D2) Ones→XAD (3-16) 03.08.15.1 (4E)
I6 (D2) Carry→XAD (17) 03.06.03.1 (5I)
I6 (D2) PC→XAD→IBR MAR MAR 03.08.15.1 (4G)
Proceed

See BTT; ETT for CPU and Channel Cycle Timings

Figure 87. Sense Printer Test (SPT)

at Ch A11, and at Ch L8 the Ch L-E end trigger is set. Channel I-time is set at Ch L10, and the MF go trigger is set at Ch I0. The MF go trigger ON allows CPU to end operation. Channel I time is reset at Ch A6.

Transfer on Channel in Operation TCO + 006X (I, L)

If the specified data channel is in operation, the computer transfers to location X. If not, the computer takes the next sequential instruction. Operation codes + 0060- + 0067 check channels A-H respectively. See Figure 88.

There are two TCO CM triggers (02.10.07.1). One may be set at I7, the other at L4. In either case, the PR CM trigger is set at L5.

Transfer on Channel Not in Operation TCN − 006X (I, L)

If the specified data channel is not in operation, the computer transfers to location X. If the channel is in operation, the computer takes the next sequential instruction. Operation codes − 0060- − 0067 check channels A-H respectively.

There are two TCO CM triggers (02.10.07.1). One may be set at I7, the other at L4. In either case, the PR CM trigger is set at L5. See Figure 88.

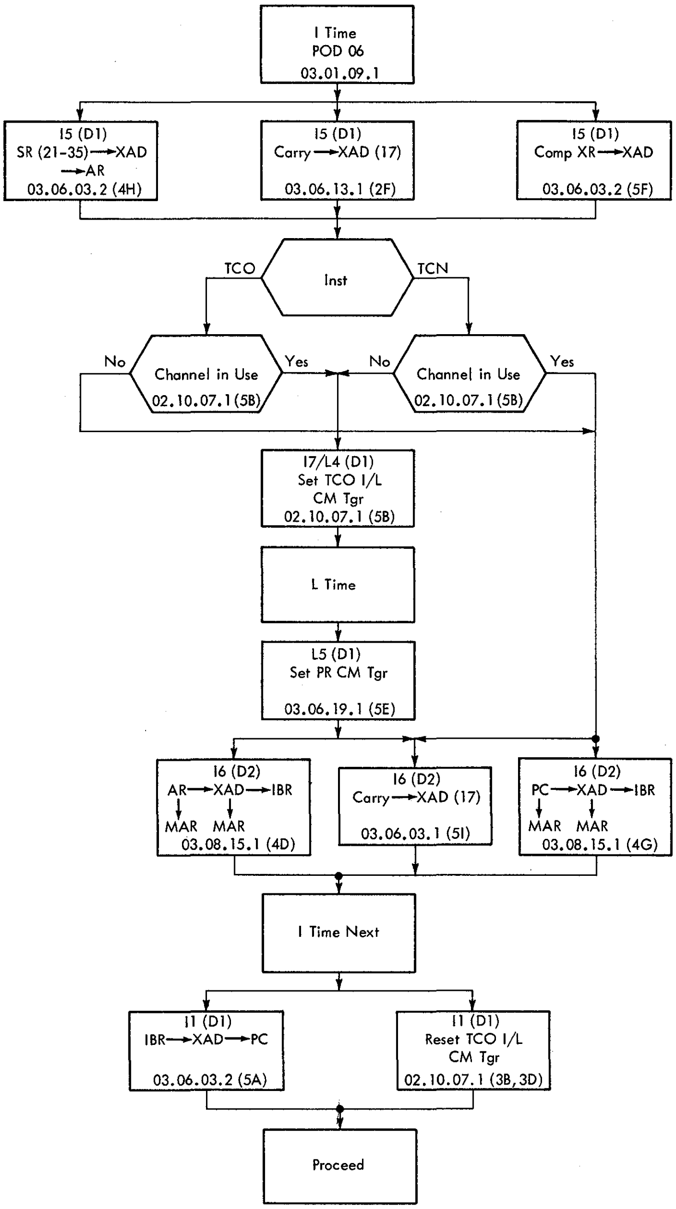
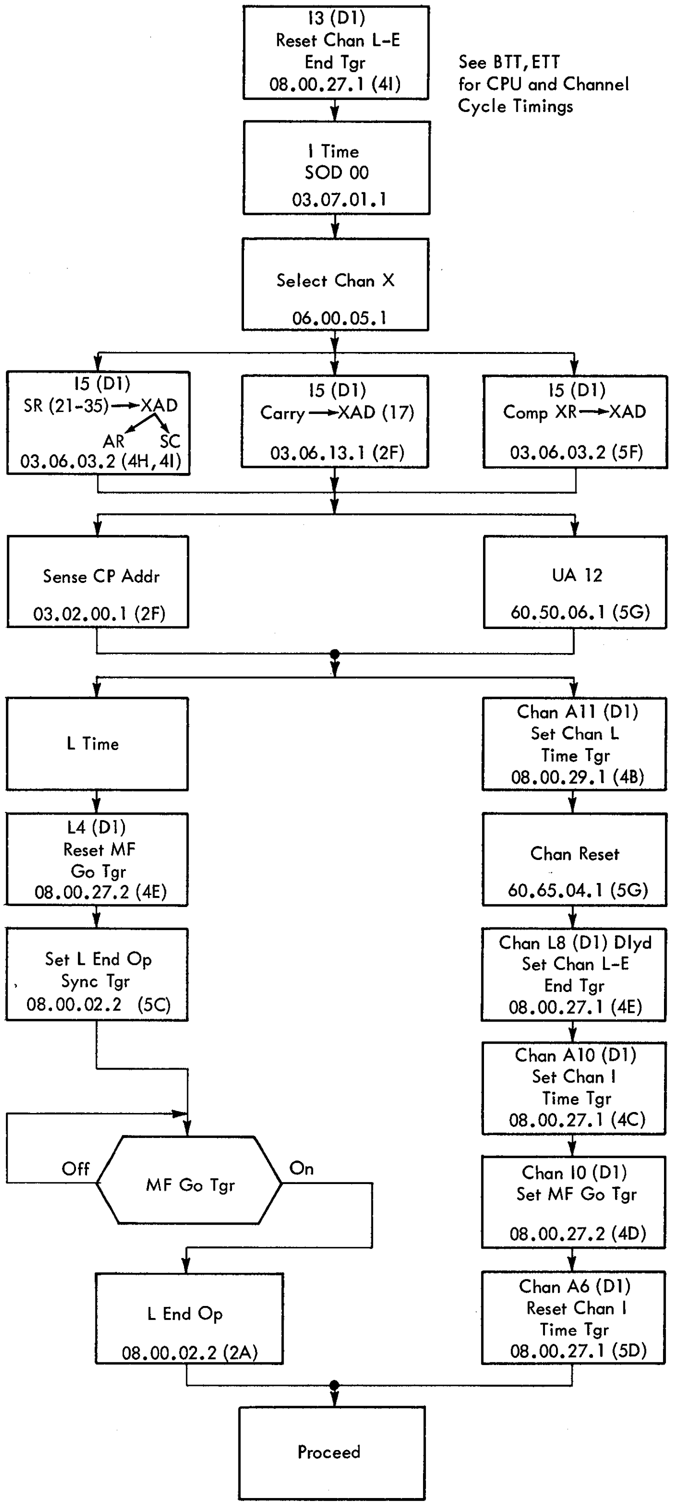


Figure 88. Transfer on Channel in Operation (TCO); Transfer on Channel Not in Operation (TCN)

Reset Data Channel RDC + 0760 . . .
CC 352 I, L-L (Ch L, I)

This instruction initiates a channel reset for the specified channel.

During I-time the channel is specified and the shift counter output decoded. "Sense CP adr" (03.02.00.1), plus UA12 (60.50.06.1-5G) provide a channel reset (60.65.04.1-5G) during channel L time. The MF go trigger is reset at L4, setting the L-end-op sync trigger. Channel L-time is set at Ch A11. At Ch L8, the Ch L-E end trigger is set. Channel I-time is set at Ch A10, and the MF go trigger is set at Ch I0. Setting the MF go trigger allows CPU to end operation, and channel I-time is reset at Ch A6. See Figure 89.



See BTT, ETT for CPU and Channel Cycle Timings

Figure 89. Reset Data Channel (RDC)

IBM 7151 Model 2 Console Control Unit

The IBM 7151 Model 2 Console Control Unit is a separate unit that may be placed at any convenient location within cable length restrictions. It provides manual and semi-automatic control over the system. The console consists of three panels: an operator's panel, a customer engineering test panel, and a marginal check panel. The time spent learning to use the console is returned many times in diagnosing system errors.

This section introduces keys, lamps, switches, test facilities; their function, and any associated logic.

During the progress of a program, the operator may need some amount of control; for example, at a given point in a calculation, the computer is given the instruction to halt. The operator can then make a visual check of the information developed so far in the program. At this point, several alternate manual steps may be performed, depending on the data observed. For these operations, the automatic-manual switch is set to manual. With the computer in this state, the operator may enter and execute an instruction, interrogate any location in storage for a visual check of the information stored, or load data from the operator's panel keys. After the desired manipulations have been made, the machine is returned to automatic status, the start key depressed, and the program continues.

The start and stop of the machine are under control of the master stop trigger. This trigger in turn controls "B cycle interrupt," which gates the I, E, and L cycles.

The keys, switches, and lamps on the console provide a means to: start or stop the machine, step through a program at reduced speed, check the status of the CPU, display or revise the contents of storage, and alter the program.

In addition, the customer engineer has facilities for several testing features which include: auxiliary start and reset key, I/O interlocks, continuous execution of an instruction, power jacks for test equipment, and overlap and diagnostic mode control switches.

Figure 90 designates systems page locations for the keys and indicators located on the console. The console (Figure 91) is divided into three sections: an operator's panel, a customer engineer's test panel, and a marginal check panel.

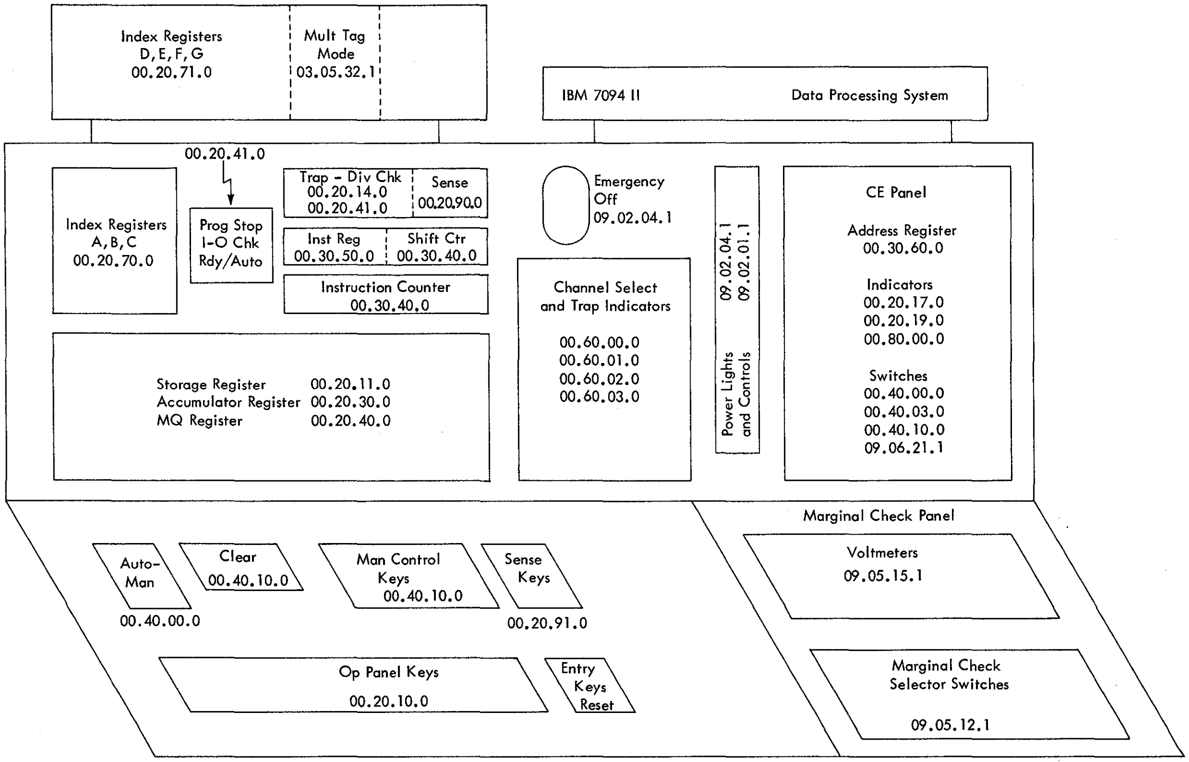

Index Registers D, E, F, G 00.20.71.0
Mult Tag Mode 03.05.32.1
IBM 7094 II
Data Processing System
00.20.41.0
Index Registers A, B, C 00.20.70.0
Prog Stop I-O Chk Rdy/Auto
Trap - Div Chk 00.20.14.0 00.20.41.0
Sense 00.20.90.0
Inst Reg 00.30.50.0
Shift Ctr 00.30.40.0
Instruction Counter 00.30.40.0
Storage Register 00.20.11.0
Accumulator Register 00.20.30.0
MQ Register 00.20.40.0
Emergency Off 09.02.04.1
Channel Select and Trap Indicators
00.60.00.0
00.60.01.0
00.60.02.0
00.60.03.0
09.02.04.1
09.02.01.1
Power Lights and Controls
CE Panel
Address Register 00.30.60.0
Indicators 00.20.17.0 00.20.19.0 00.80.00.0
Switches 00.40.00.0 00.40.03.0 00.40.10.0 09.06.21.1
Auto-Man
00.40.00.0
Clear 00.40.10.0
Man Control Keys 00.40.10.0
Sense Keys
00.20.91.0
Op Panel Keys 00.20.10.0
Entry Keys Reset
Marginal Check Panel
Voltmeters 09.05.15.1
Marginal Check Selector Switches
09.05.12.1

Figure 90. Block Diagram of Console

Operator's Panel

The operator's panel provides for visual checking of the information in the computer and for manual control of the computer's functions. It is also a station from which power may be applied to or removed from the system.

Indicators

All indicators on the console are incandescent. When used to indicate the condition of a register, a lamp ON signifies a one, while a lamp OFF signifies a zero.

Internal Registers

The contents of the internal registers (accumulator, MQ, storage register, instruction counter, instruction register, and index registers) are displayed on the panel.

Multiple Tag Mode

This lamp, when on, indicates the computer is operating in three-index register mode (709/7090 mode). When off, this lamp indicates that the computer is out of multi-tag mode, and that all seven index registers are available to the program (7094 mode).

Trap

This lamp is on whenever the computer is in the transfer trapping mode.

Simulate

This lamp is on when the 7090 is operating in any of the following modes associated with the 704, 709, or 7090 compatibility program:

I/O select and sense trap mode
Copy and locate drum address trap mode
Storage nullification mode

Accumulator Overflow

This lamp turns on any time during a fixed point operation (for example, add, subtract) or a shifting operation that a carry occurs out of AC (1). It is also turned

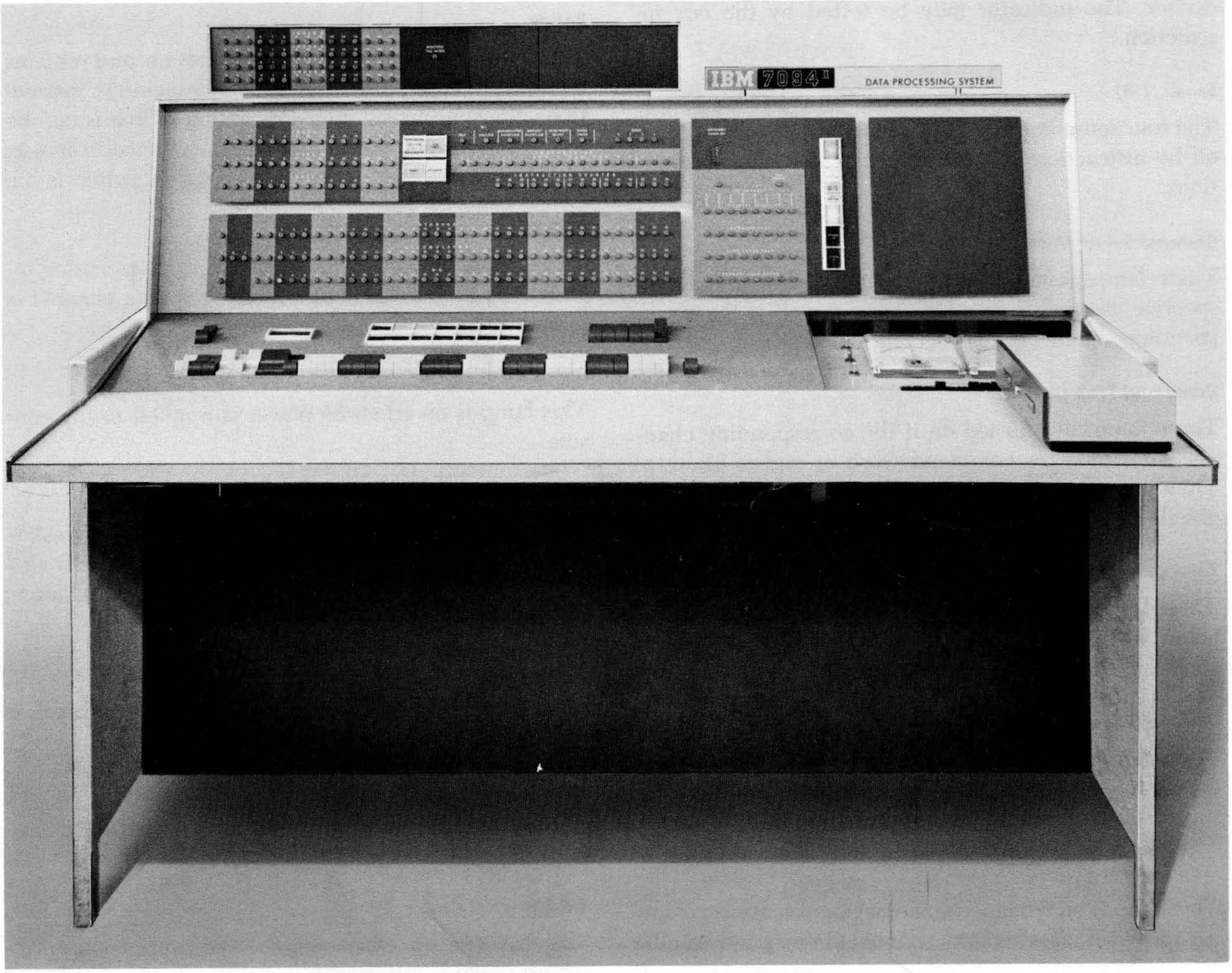


Figure 91. 7151 Model 2 Console

on by a bit in position (P) during the execution of a floating-point instruction while the computer is in compatibility mode. It may be turned off by the TNO or TOV instruction.

Quotient Overflow

This lamp is on whenever the computer is using the compatibility program and an MQ overflow occurs. It may be turned off by the TQO instruction.

Read-Write Select

This lamp is on whenever the channel-in-use trigger is on in any data channel.

Divide Check

This lamp is turned on in fixed point division if the dividend (AC) is greater than or equal to the divisor (SR). In floating-point divide, the lamp is on if the magnitude of the fraction of the dividend is greater than or equal to twice the magnitude of the fraction of the divisor. The indicator may be tested by the DCT instruction.

Sense (1-4)

The four sense lamps on the panel may be turned on or off by instructions and then checked by sense instructions.

Channel Select (A-H)

These lamps, one for each channel, are on if their respective channel is selected. They are off if the corresponding channel is not selected.

Command Trap (A-H)

These lamps are turned on if the corresponding channel is enabled for command word or end-of-file trap. They are off if the corresponding channel is not enabled.

Tape Check Trap (A-H)

These lamps are on if the corresponding channel is enabled for tape check trap. They are off if the corresponding channel is not enabled.

Channel Tape Check (A-H)

These lamps are turned on if a redundancy check error occurs in the corresponding channel. The lamp is turned off on execution of a TRC instruction.

Trap Control

This lamp is on when a channel is not executing a channel trap; it is off when any channel enters a trap condition. While the lamp is off, no channel traps may be executed. Channel traps may be executed only when the lamp is on at the same time as any of the enabled lamps.

Program Stop

This lamp is turned on whenever the computer executes a halt instruction and no data channels are in operation (DVH, VDH, FDH, and DFDH excepted).

I/O Check

This lamp may be turned on:

If an RCH or LCH is decoded and the specified data channel has not been selected.

If, when writing, a data channel data register has not been loaded with a word from storage by the time its contents are to be sent to the output unit.

If, when reading, a data channel data register has not transmitted its contents to storage by the time new information is to be loaded into it from an output unit.

The I/O check lamp may be turned off by the execution of an IOT instruction.

Ready

This lamp comes on after power comes up, and remains on except when: the computer is executing instructions, the continuous enter instruction switch is on, the I/O interlock switch is in manual, the channel is in use, or the memory diagnostic or no overlap switch is on.

Automatic

This lamp is on whenever the computer is executing instructions in automatic or whenever a data channel is in operation.

Console Power-On

This lamp is on when DC power is applied to the console.

Central Computer Power Check

This lamp comes on whenever a fuse or circuit breaker opens in CPU 1 or 2, multiplexor, or core storage. It also lights when core storage has improper oil temperature or low oil pressure.

I/O Power Check

This lamp comes on whenever a fuse or circuit breaker opens in a data channel.

Power

This lamp is turned on whenever DC power is up in core storage.

+6 Marginal Check

This lamp is on whenever the +6-supply marginal-check variable-autotransformer is not in the home position.

–12 Marginal Check

This lamp is on whenever the –12-supply marginal-check variable-autotransformer is not in the home position.

Manual Controls

Figure 92 shows the keys and switches on the operator's panel that start and stop the machine and initiate computer functions. All controls are of the spring-returned variety except the auto-manual key, entry keys, sense keys, and emergency-off switch.

Power On

When the system is in normal-off status, pressing the power-on key starts the power-on sequence. Ready status (power-on) will be reached in about 20 seconds. As power comes on, a clear operation resets all registers and triggers and sets memory to all zeros.

Normal Off

The normal-off key initiates:

Immediate removal of 60-cycle power from the MG set, MG blower, and all frame blowers except memory.

Immediate removal of 400-cycle power from the 30-60 volt memory power supply.

After five seconds, removal of 400-cycle power from the standard memory supply.

After three minutes, removal of power from the memory blowers.

The dropping of the 60-cycle power removes the input to the MG, but the MG rotates at about full speed for longer than five seconds, allowing the memory power to sequence down.

Emergency Off

When the emergency-off switch is pulled, all power is immediately removed from the system except for the voltage to HR24 and 30 points in the power control unit. The emergency-off switch is used only in emergencies because of possible damage to circuits.

Automatic-Manual Key

The auto-manual key controls the rest of the keys on the console. If the key is depressed while a program is running, the computer stops upon completion of the instruction being executed. The data channels complete their operations before stopping.

Entry Keys

Thirty-six entry keys are on the operator's panel (S, 1-35). Depressing a key sets a one in that position; leaving a key normal sets a zero in that position. Information set in the entry keys may be entered into storage, executed, or used for a storage inquiry address.

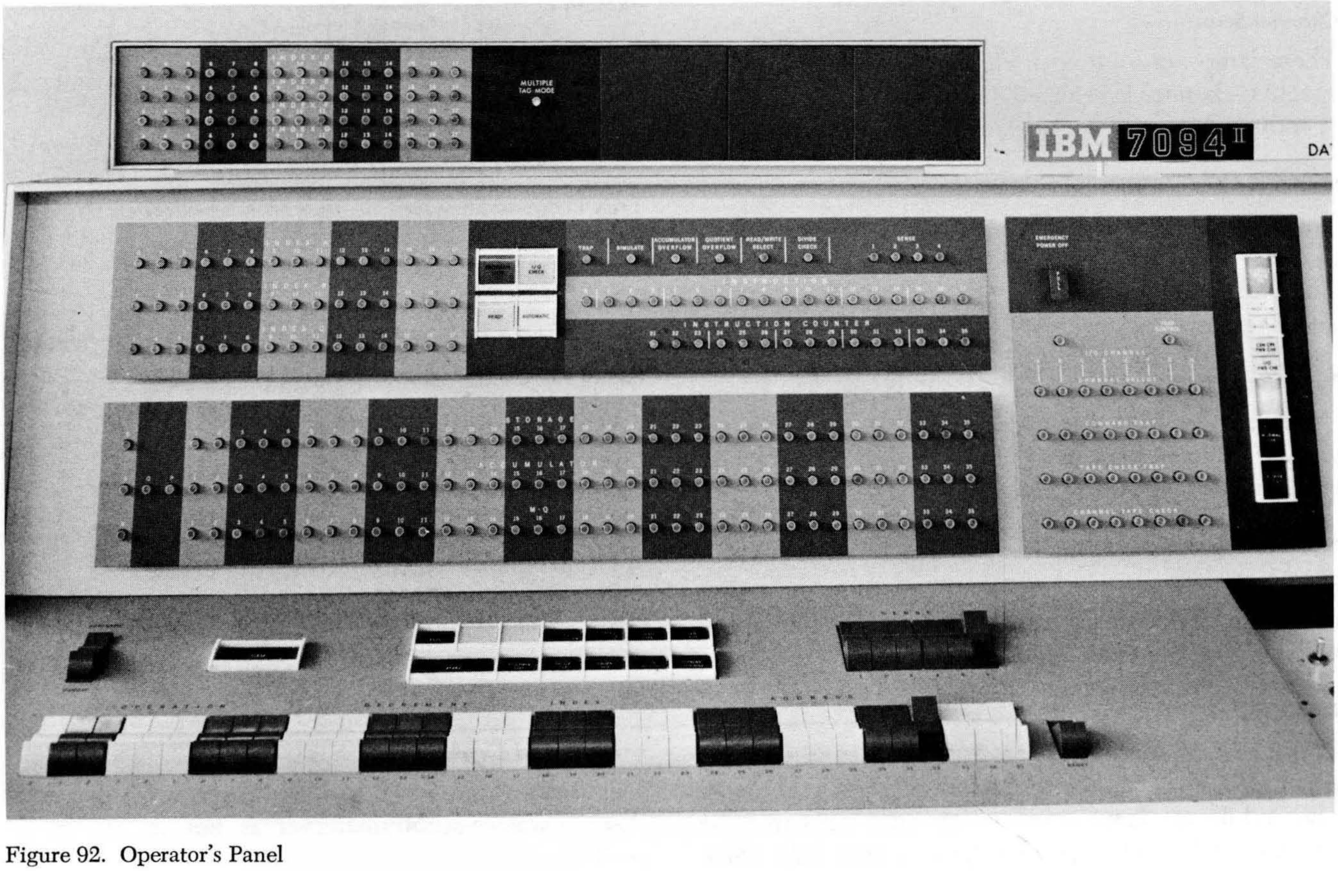


Figure 92. Operator's Panel

The entry keys may all be reset to zero by depressing the reset key to the right of position 35.

Sense Keys

The six sense keys may be set on or off for program control. Then, the condition of the keys may be tested by SWT instructions to determine whether or not the program should skip.

Manual Control Keys

When any manual control key is pressed, a series of single-shots and triggers are set. Separating the key from the usable signal prevents false indications from noise generated by the key (Figure 93). Three single-shots are fired in sequence: a 1-microsecond, a 20-millisecond and a 350-microsecond. The 350-microsecond single-shot is taken through a delay network to develop a 200-microsecond pulse that turns on the manual control trigger. By this time, the key has settled down, and the trigger to perform the desired operation turns on, resetting the manual control trigger.

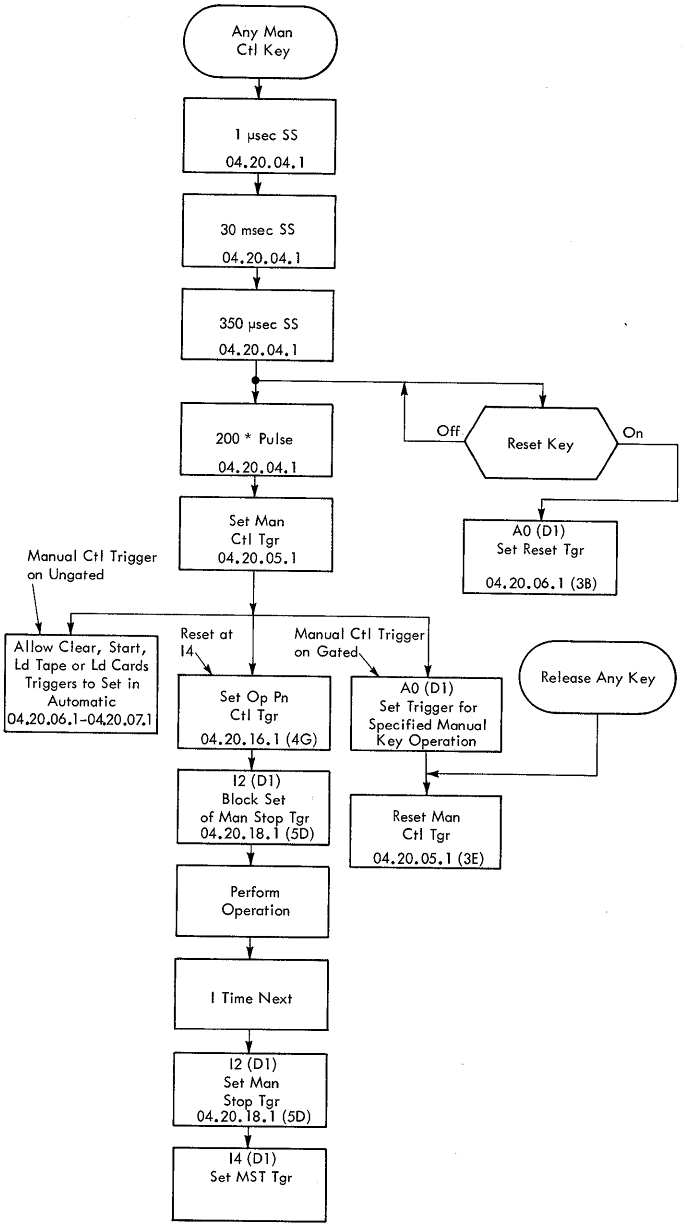

Any Man Ctl Key
1 μsec SS 04.20.04.1
30 msec SS 04.20.04.1
350 μsec SS 04.20.04.1
200 * Pulse 04.20.04.1
Off
Reset Key
On
A0 (D1) Set Reset Tgr 04.20.06.1 (3B)
Set Man Ctl Tgr 04.20.05.1
Manual Ctl Trigger on Ungated
Allow Clear, Start, Ld Tape or Ld Cards Triggers to Set in Automatic 04.20.06.1-04.20.07.1
Reset at 14
Manual Ctl Trigger on Gated
Set Op Pn Ctl Tgr 04.20.16.1 (4G)
A0 (D1) Set Trigger for Specified Manual Key Operation
Release Any Key
12 (D1) Block Set of Man Stop Tgr 04.20.18.1 (5D)
Reset Man Ctl Tgr 04.20.05.1 (3E)
Perform Operation
I Time Next
12 (D1) Set Man Stop Tgr 04.20.18.1 (5D)
14 (D1) Set MST Tgr

Figure 93. Manual Control Keys

Start

If the CPU is in automatic and ready status, pressing the start key initiates machine operation by turning off the master-stop trigger and the program-stop light. The master-stop trigger off conditions "not B cycle interrupt," allowing the computer to proceed. If the system is in manual status, pressing the start key turns off the program-stop light. See Figure 94.

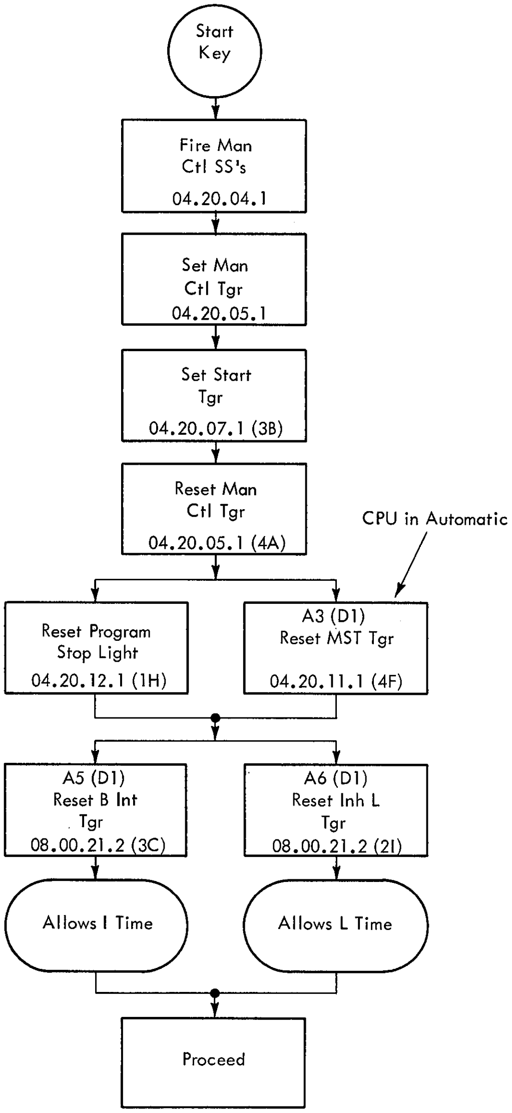

Start Key
Fire Man Ctl SS's 04.20.04.1
Set Man Ctl Tgr 04.20.05.1
Set Start Tgr 04.20.07.1 (3B)
Reset Man Ctl Tgr 04.20.05.1 (4A)
CPU in Automatic
Reset Program Stop Light 04.20.12.1 (1H)
A3 (D1) Reset MST Tgr 04.20.11.1 (4F)
A5 (D1) Reset B Int Tgr 08.00.21.2 (3C)
A6 (D1) Reset Inh L Tgr 08.00.21.2 (2I)
Allows I Time
Allows L Time
Proceed

Figure 94. Start

Clear

The clear key is only operative if the computer is in automatic status. Pressing the clear key:

Fires a one-microsecond single-shot to reset the clock and all channel registers.

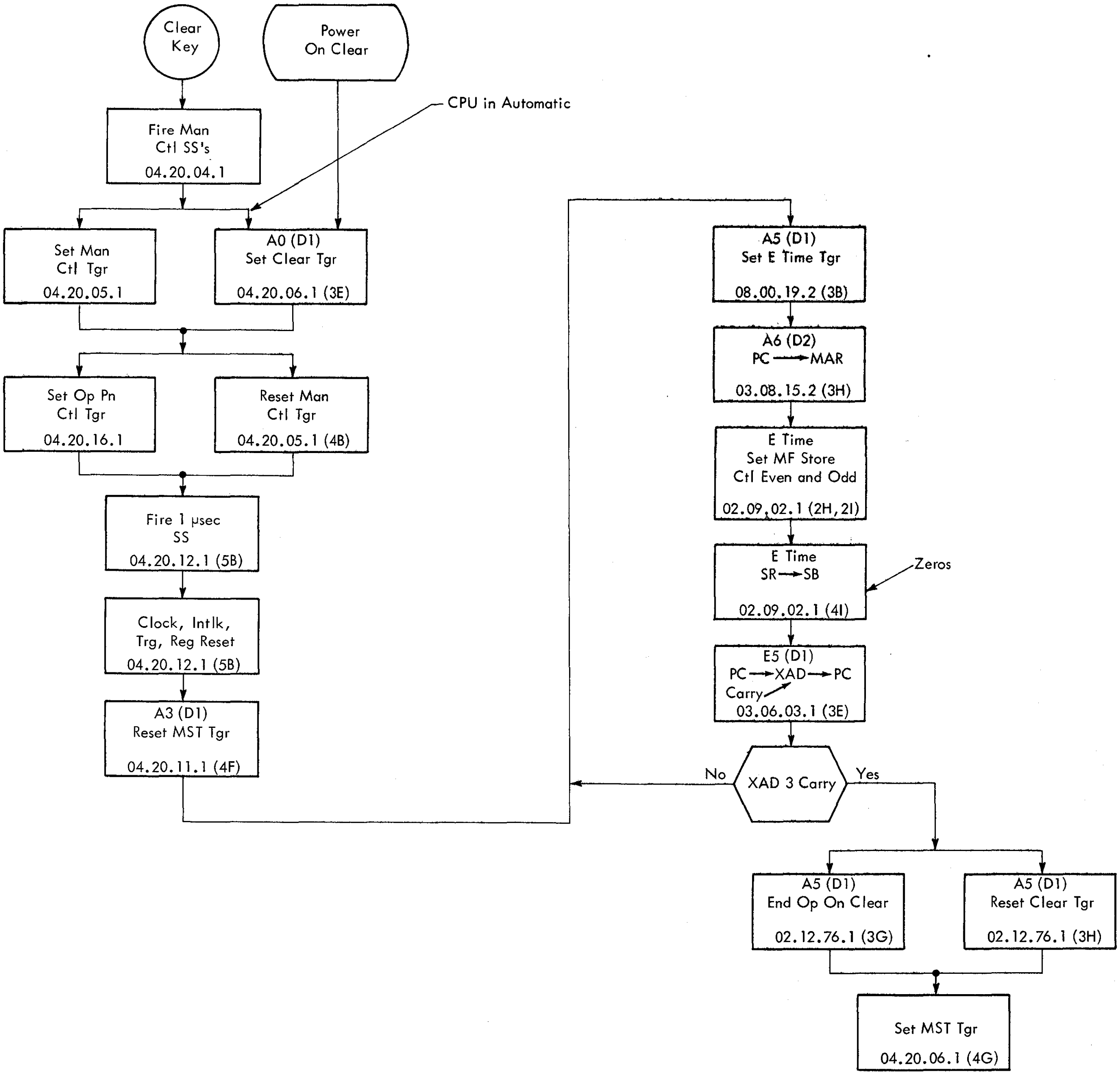
Resets CPU interlocks and registers.

Conditions cicruits which allow 0's to be written into all storage locations.

The program counter controls the stepping through memory, with XAD(3) carry trigger indicating when all addresses have been zeroed. The turn-on of the XAD(3) carry trigger turns on the master-stop trigger (Figure 95). Overlap mode is inhibited.

Reset

The reset key causes the resetting of all registers (except sense indicator), operator's panel lights (except power-on, ready, and multiple tag mode), and all channels and their associated registers in on-line status. After a reset operation, the CE test panel lights remaining on are: end-op, master stop, and I-cycle time.



Clear Key
Power On Clear
CPU in Automatic
Fire Man Ctl SS's 04.20.04.1
Set Man Ctl Tgr 04.20.05.1
A0 (D1) Set Clear Tgr 04.20.06.1 (3E)
Set Op Pn Ctl Tgr 04.20.16.1
Reset Man Ctl Tgr 04.20.05.1 (4B)
Fire 1 μsec SS 04.20.12.1 (5B)
Clock, Intlk, Trg, Reg Reset 04.20.12.1 (5B)
A3 (D1) Reset MST Tgr 04.20.11.1 (4F)
A5 (D1) Set E Time Tgr 08.00.19.2 (3B)
A6 (D2) PC → MAR 03.08.15.2 (3H)
E Time Set MF Store Ctl Even and Odd 02.09.02.1 (2H, 2I)
E Time SR → SB 02.09.02.1 (4I)
Zeros
E5 (D1) PC → XAD → PC Carry 03.06.03.1 (3E)
No
XAD 3 Carry
Yes
A5 (D1) End Op On Clear 02.12.76.1 (3G)
A5 (D1) Reset Clear Tgr 02.12.76.1 (3H)
Set MST Tgr 04.20.06.1 (4G)

Figure 95. Clear Storage

Display Storage

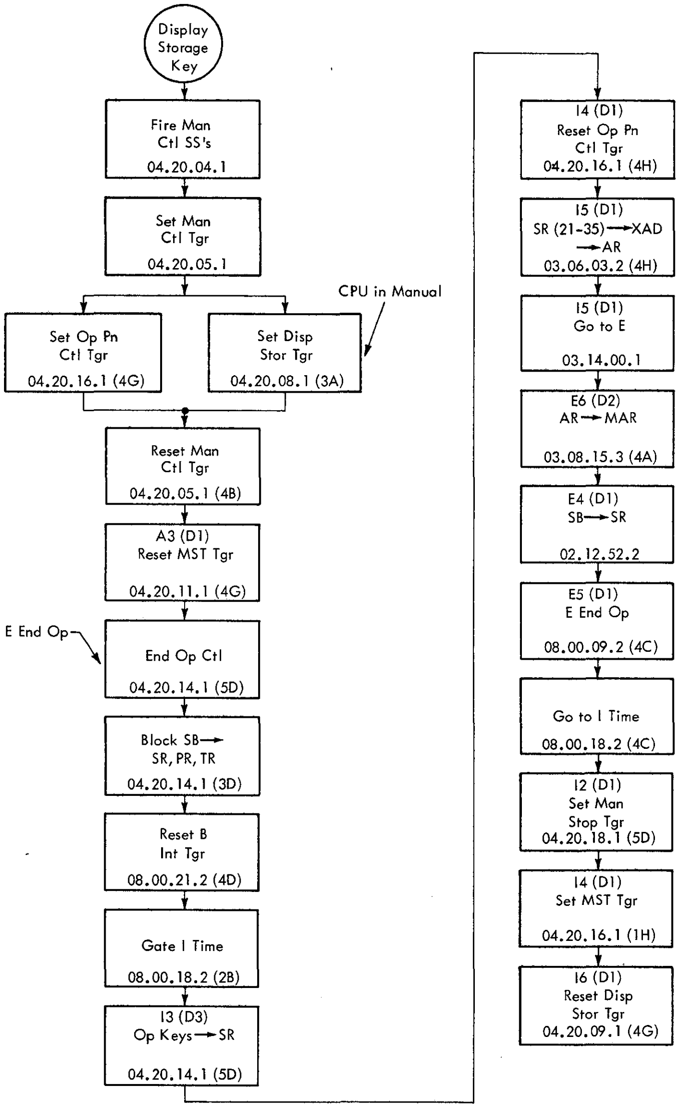
Pressing the display storage key causes sensing of the address portion of the operator's panel keys to determine the address in storage to be displayed. All 36 bit-positions of storage will be displayed in the SR. This is accomplished by:

Turning off the master stop trigger.

Bringing the operator's panel keys to the SR in I time.

Suppressing "storage bus to storage register."

The address portion of the SR is routed to the address register. During E time, the SB is gated to the SR, which contains the 36 bits of the desired address. If a tag or indirect addressing is specified in the operator's keys, the contents of the effective address or of the IA address will be displayed (Figure 96). Depressing the display storage key may wipe out the contents of the IBR. Overlap mode is inhibited.

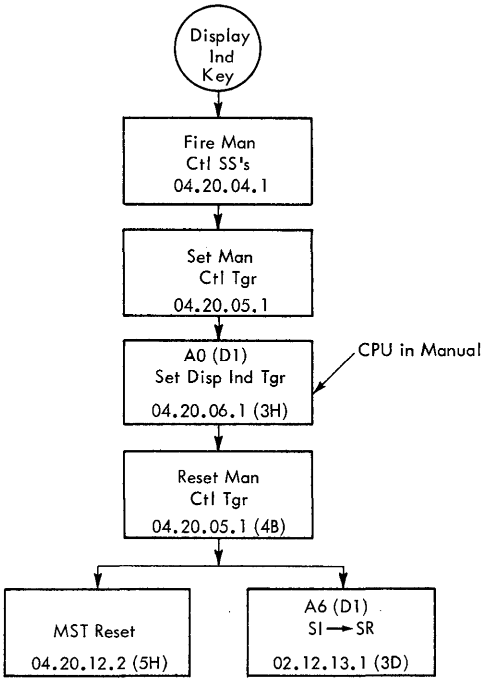


Display Storage Key
Fire Man Ctl SS's 04.20.04.1
Set Man Ctl Tgr 04.20.05.1
Set Op Pn Ctl Tgr 04.20.16.1 (4G)
Set Disp Stor Tgr 04.20.08.1 (3A)
CPU in Manual
Reset Man Ctl Tgr 04.20.05.1 (4B)
A3 (D1) Reset MST Tgr 04.20.11.1 (4G)
E End Op
End Op Ctl 04.20.14.1 (5D)
Block SB→ SR, PR, TR 04.20.14.1 (3D)
Reset B Int Tgr 08.00.21.2 (4D)
Gate I Time 08.00.18.2 (2B)
I3 (D3) Op Keys→SR 04.20.14.1 (5D)
I4 (D1) Reset Op Pn Ctl Tgr 04.20.16.1 (4H)
I5 (D1) SR (21-35)→XAD →AR 03.06.03.2 (4H)
I5 (D1) Go to E 03.14.00.1
E6 (D2) AR→MAR 03.08.15.3 (4A)
E4 (D1) SB→SR 02.12.52.2
E5 (D1) E End Op 08.00.09.2 (4C)
Go to I Time 08.00.18.2 (4C)
I2 (D1) Set Man Stop Tgr 04.20.18.1 (5D)
I4 (D1) Set MST Tgr 04.20.16.1 (1H)
I6 (D1) Reset Disp Stor Tgr 04.20.09.1 (4G)

Figure 96. Display Storage

Display Indicators

The display indicators key gates the true value of the sense indicators to the SR for display. See Figure 97.



Display Ind Key
Fire Man Ctl SS's 04.20.04.1
Set Man Ctl Tgr 04.20.05.1
A0 (D1) Set Disp Ind Tgr 04.20.06.1 (3H)
CPU in Manual
Reset Man Ctl Tgr 04.20.05.1 (4B)
MST Reset 04.20.12.2 (5H)
A6 (D1) SI→SR 02.12.13.1 (3D)

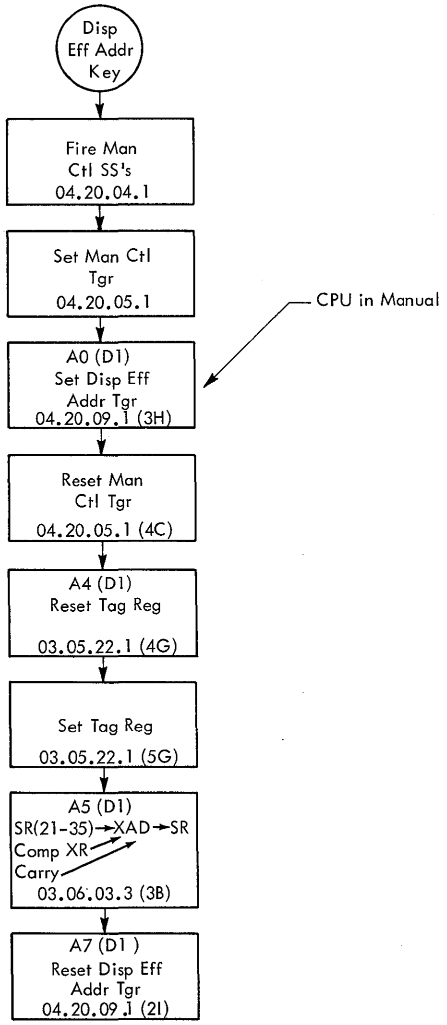
Figure 97. Display Indicators

Display Effective Address

The display effective address key initiates the calculation of the effective address of the instruction in the SR. The actual address from the SR is combined with the 2's complement of the specified index register to produce the effective address. This calculated address is then placed in the address portion of the storage register. Positions S-20 of the storage register are set to 0. The effective address may be calculated only once because positions 18, 19, and 20 of the SR have been set to 0, an indication of no index register. See Figure 98.

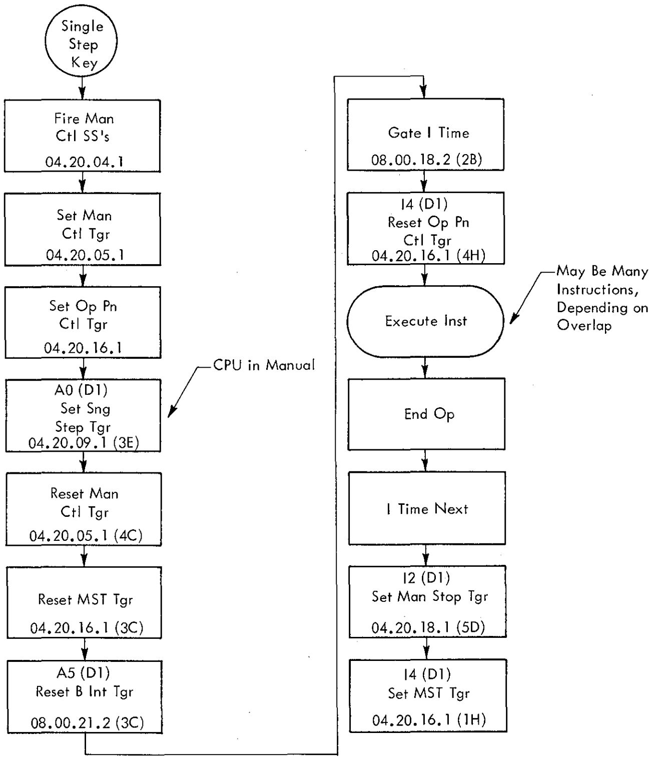
Single Step

Pressing the single-step key results in executing the instruction whose address is in the instruction counter before key depression. The instruction counter will be advanced, or altered under control of the instruction executed once for each time the key is pressed. If an I/O operation is executed, the machine will continue to execute instructions at high speed until the end of the I/O operation. If the continuous-enter instruction switch is on, the single-step key is pressed, and if the 7094 is in manual status; the instruction set in the op keys will be executed once. The single-step key is effective only if the system is in manual status, and not in program stop status. See Figure 99.



Disp Eff Addr Key
Fire Man Ctl SS's 04.20.04.1
Set Man Ctl Tgr 04.20.05.1
CPU in Manual
A0 (D1) Set Disp Eff Addr Tgr 04.20.09.1 (3H)
Reset Man Ctl Tgr 04.20.05.1 (4C)
A4 (D1) Reset Tag Reg 03.05.22.1 (4G)
Set Tag Reg 03.05.22.1 (5G)
A5 (D1) SR(21-35)→XAD→SR Comp XR Carry 03.06.03.3 (3B)
A7 (D1) Reset Disp Eff Addr Tgr 04.20.09.1 (2I)

Figure 98. Display Effective Address

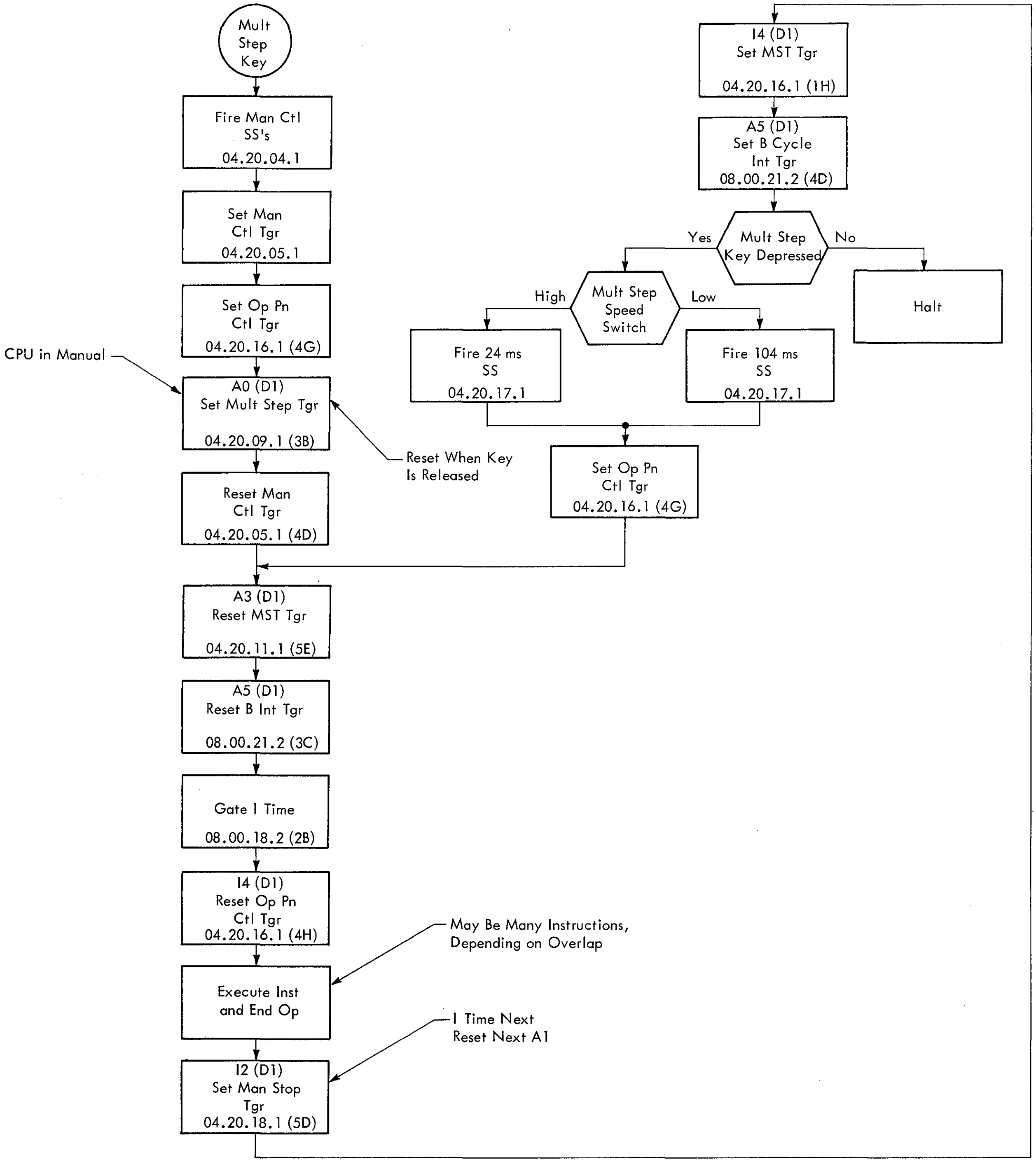


Single Step Key
Fire Man Ctl SS's 04.20.04.1
Set Man Ctl Tgr 04.20.05.1
Set Op Pn Ctl Tgr 04.20.16.1
CPU in Manual
A0 (D1) Set Sng Step Tgr 04.20.09.1 (3E)
Reset Man Ctl Tgr 04.20.05.1 (4C)
Reset MST Tgr 04.20.16.1 (3C)
A5 (D1) Reset B Int Tgr 08.00.21.2 (3C)
Gate I Time 08.00.18.2 (2B)
I4 (D1) Reset Op Pn Ctl Tgr 04.20.16.1 (4H)
Execute Inst
May Be Many Instructions, Depending on Overlap
End Op
I Time Next
I2 (D1) Set Man Stop Tgr 04.20.18.1 (5D)
I4 (D1) Set MST Tgr 04.20.16.1 (1H)

Figure 99. Single Step

Multiple Step

This key is effective only in manual status when the program stop trigger is on. The multiple-step key causes the repetition of single-step operations. The rate of operation is under control of a toggle switch located on the customer engineering test panel. The operator has the choice of low speed with a delay of 104 milliseconds between each instruction, or high speed with a delay of 24 milliseconds between each instruction. The program will continue to run as long as the multiple-step key is pressed, or until a program halt is decoded. See Figure 100.



Mult Step Key
Fire Man Ctl SS's 04.20.04.1
Set Man Ctl Tgr 04.20.05.1
Set Op Pn Ctl Tgr 04.20.16.1 (4G)
CPU in Manual
A0 (D1) Set Mult Step Tgr 04.20.09.1 (3B)
Reset When Key Is Released
Reset Man Ctl Tgr 04.20.05.1 (4D)
A3 (D1) Reset MST Tgr 04.20.11.1 (5E)
A5 (D1) Reset B Int Tgr 08.00.21.2 (3C)
Gate I Time 08.00.18.2 (2B)
I4 (D1) Reset Op Pn Ctl Tgr 04.20.16.1 (4H)
May Be Many Instructions, Depending on Overlap
Execute Inst and End Op
I Time Next Reset Next A1
I2 (D1) Set Man Stop Tgr 04.20.18.1 (5D)
I4 (D1) Set MST Tgr 04.20.16.1 (1H)
A5 (D1) Set B Cycle Int Tgr 08.00.21.2 (4D)
Yes
Mult Step Key Depressed
No
Halt
High
Mult Step Speed Switch
Low
Fire 24 ms SS 04.20.17.1
Fire 104 ms SS 04.20.17.1
Set Op Pn Ctl Tgr 04.20.16.1 (4G)

Figure 100. Multiple Step Key

Enter MQ

The enter MQ key provides a means of loading the MQ register from the operator's entry keys. Then, the information may be loaded into storage by placing the instruction STQ along with the desired address in the entry keys and depressing the enter instruction key. The enter MQ key is effective only in manual status. See Figure 101.

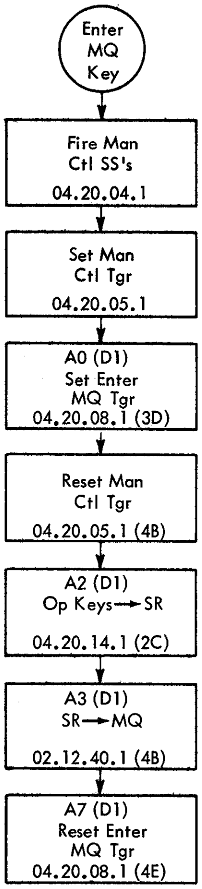
Enter Instruction Key

The enter instruction key executes the instruction entered in the operator's panel keys. The contents of the instruction counter remain unchanged except for a transfer-type instruction; skip instructions will not alter the program counter. The key is effective only in manual status (Figure 102). Overlap mode is inhibited.

Load Cards

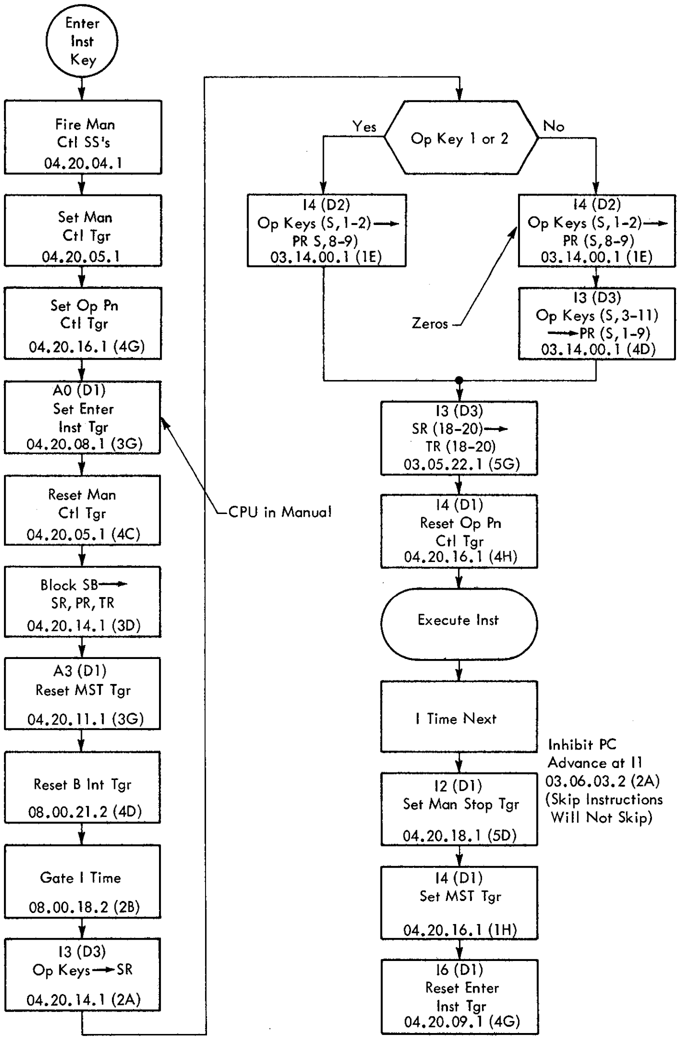
With the automatic-manual key in automatic, the computer in ready status (ready light may be off), and channel A not in manual status, depressing the load cards key: gives an interlock reset to the CPU and all channels not in manual status, selects the card reader on channel A, and turns on indicator S and sets the word counter to three (channel A).

Three words (9 row left and right, and 8 row left) from the first card enter storage location 0, 1, and 2.



Enter MQ Key
Fire Man Ctl SS's 04.20.04.1
Set Man Ctl Tgr 04.20.05.1
A0 (D1) Set Enter MQ Tgr 04.20.08.1 (3D)
Reset Man Ctl Tgr 04.20.05.1 (4B)
A2 (D1) Op Keys→SR 04.20.14.1 (2C)
A3 (D1) SR→MQ 02.12.40.1 (4B)
A7 (D1) Reset Enter MQ Tgr 04.20.08.1 (4E)

Figure 101. Enter MQ



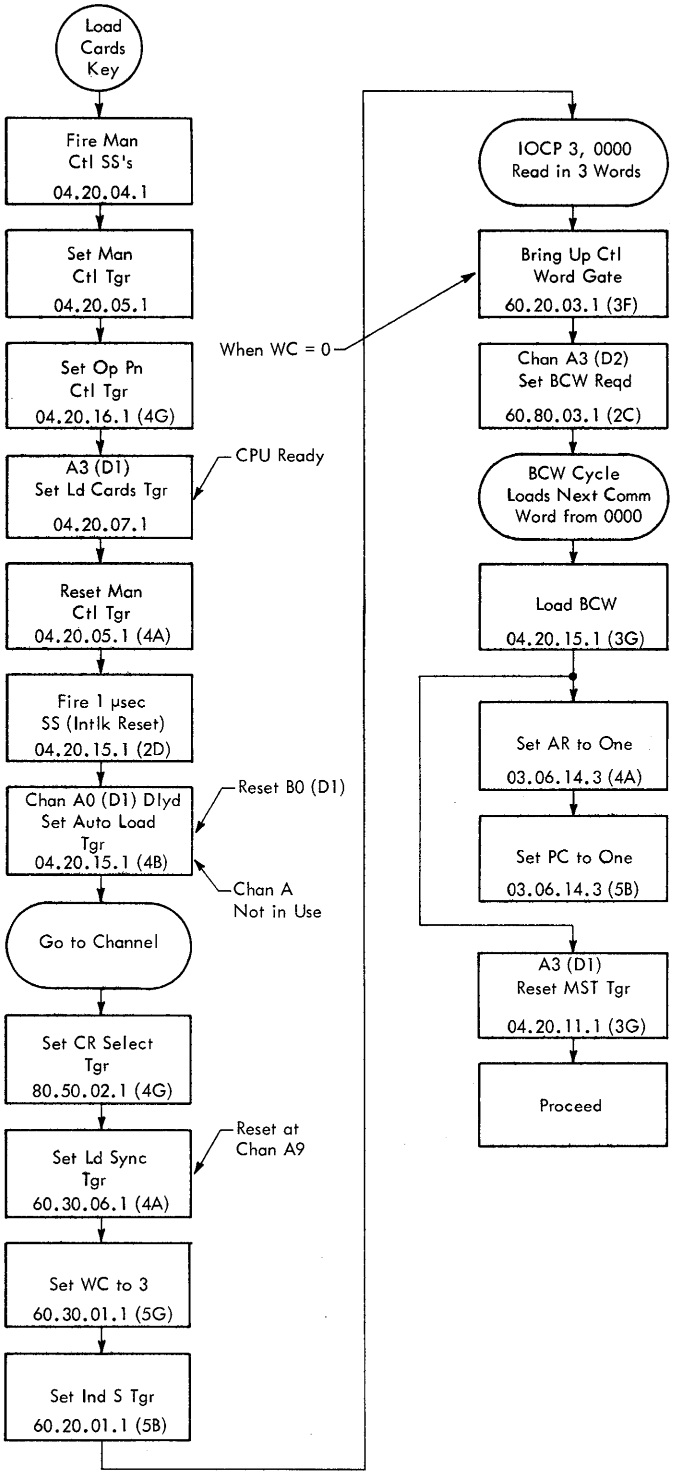
Enter Inst Key
Fire Man Ctl SS's 04.20.04.1
Set Man Ctl Tgr 04.20.05.1
Set Op Pn Ctl Tgr 04.20.16.1 (4G)
A0 (D1) Set Enter Inst Tgr 04.20.08.1 (3G)
Reset Man Ctl Tgr 04.20.05.1 (4C)
Block SB→ SR, PR, TR 04.20.14.1 (3D)
A3 (D1) Reset MST Tgr 04.20.11.1 (3G)
Reset B Int Tgr 08.00.21.2 (4D)
Gate I Time 08.00.18.2 (2B)
I3 (D3) Op Keys→SR 04.20.14.1 (2A)
CPU in Manual
Op Key 1 or 2
Yes
No
I4 (D2) Op Keys (S, 1-2)→ PR S, 8-9) 03.14.00.1 (1E)
I4 (D2) Op Keys (S, 1-2)→ PR (S, 8-9) 03.14.00.1 (1E)
Zeros
I3 (D3) Op Keys (S, 3-11) →PR (S, 1-9) 03.14.00.1 (4D)
I3 (D3) SR (18-20)→ TR (18-20) 03.05.22.1 (5G)
I4 (D1) Reset Op Pn Ctl Tgr 04.20.16.1 (4H)
Execute Inst
I Time Next
Inhibit PC Advance at I1 03.06.03.2 (2A) (Skip Instructions Will Not Skip)
I2 (D1) Set Man Stop Tgr 04.20.18.1 (5D)
I4 (D1) Set MST Tgr 04.20.16.1 (1H)
I6 (D1) Reset Enter Inst Tgr 04.20.09.1 (4G)

Figure 102. Enter Instruction

The word entered in address 0 should be a control word. When the word counter goes to 0, the channel asks for another control word from location 0. The CPU gets its next instruction from location 1. See Figure 103.

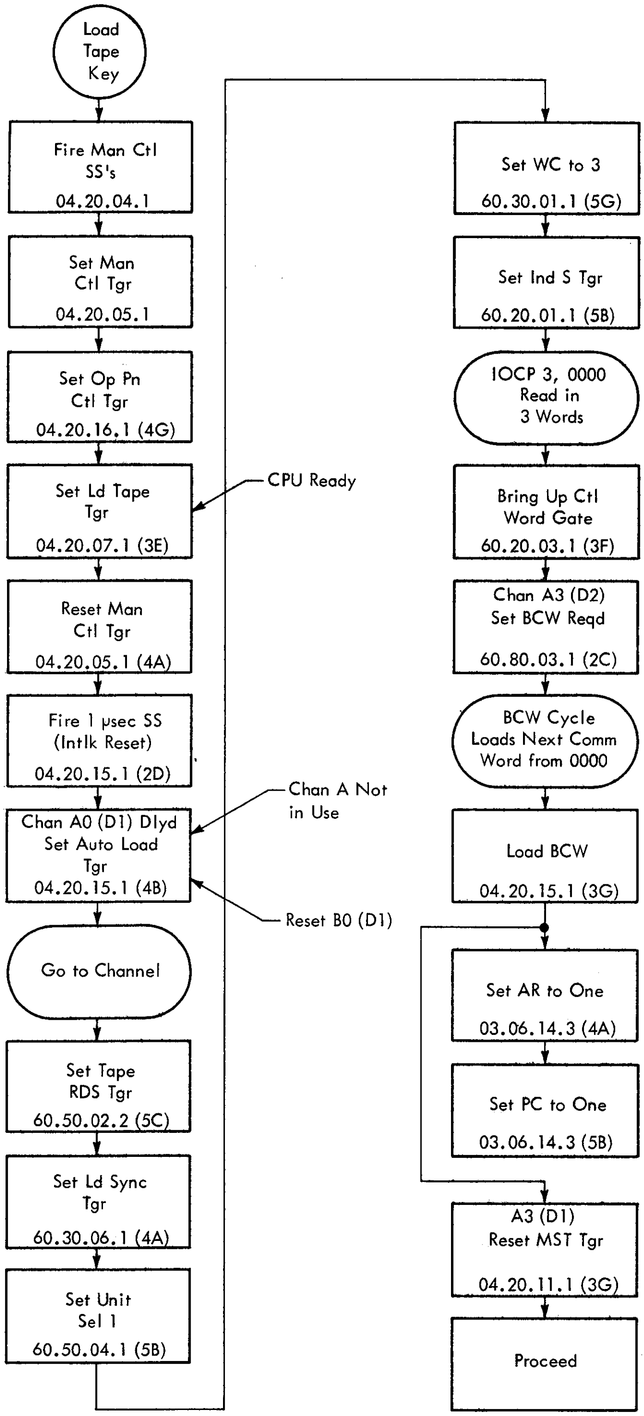
Load Tape

The load tape key works the same as the load cards key, except that it causes a read select of tape unit 1 on channel A instead of the card reader. See Figure 104.



Load Cards Key
Fire Man Ctl SS's 04.20.04.1
Set Man Ctl Tgr 04.20.05.1
Set Op Pn Ctl Tgr 04.20.16.1 (4G)
A3 (D1) Set Ld Cards Tgr 04.20.07.1
CPU Ready
Reset Man Ctl Tgr 04.20.05.1 (4A)
Fire 1 μsec SS (Intlk Reset) 04.20.15.1 (2D)
Chan A0 (D1) Dlyd Set Auto Load Tgr 04.20.15.1 (4B)
Reset B0 (D1)
Chan A Not in Use
Go to Channel
Set CR Select Tgr 80.50.02.1 (4G)
Set Ld Sync Tgr 60.30.06.1 (4A)
Reset at Chan A9
Set WC to 3 60.30.01.1 (5G)
Set Ind S Tgr 60.20.01.1 (5B)
When WC = 0
IOCP 3, 0000 Read in 3 Words
Bring Up Ctl Word Gate 60.20.03.1 (3F)
Chan A3 (D2) Set BCW Reqd 60.80.03.1 (2C)
BCW Cycle Loads Next Comm Word from 0000
Load BCW 04.20.15.1 (3G)
Set AR to One 03.06.14.3 (4A)
Set PC to One 03.06.14.3 (5B)
A3 (D1) Reset MST Tgr 04.20.11.1 (3G)
Proceed

Figure 103. Load Cards



Load Tape Key
Fire Man Ctl SS's 04.20.04.1
Set Man Ctl Tgr 04.20.05.1
Set Op Pn Ctl Tgr 04.20.16.1 (4G)
Set Ld Tape Tgr 04.20.07.1 (3E)
CPU Ready
Reset Man Ctl Tgr 04.20.05.1 (4A)
Fire 1 μsec SS (Intlk Reset) 04.20.15.1 (2D)
Chan A0 (D1) Dlyd Set Auto Load Tgr 04.20.15.1 (4B)
Chan A Not in Use
Reset B0 (D1)
Go to Channel
Set Tape RDS Tgr 60.50.02.2 (5C)
Set Ld Sync Tgr 60.30.06.1 (4A)
Set Unit Sel 1 60.50.04.1 (5B)
Set WC to 3 60.30.01.1 (5G)
Set Ind S Tgr 60.20.01.1 (5B)
IOCP 3, 0000 Read in 3 Words
Bring Up Ctl Word Gate 60.20.03.1 (3F)
Chan A3 (D2) Set BCW Reqd 60.80.03.1 (2C)
BCW Cycle Loads Next Comm Word from 0000
Load BCW 04.20.15.1 (3G)
Set AR to One 03.06.14.3 (4A)
Set PC to One 03.06.14.3 (5B)
A3 (D1) Reset MST Tgr 04.20.11.1 (3G)
Proceed

Figure 104. Load Tape

Customer Engineering Test Panel

In addition to the indicators and manual controls on the operator's panel, the CE has the indicators and controls on the customer engineering test panel.

The indicators provide a means for checking the address register contents, the tally counter, various test triggers, and the cycle time. The switches and jacks provide means to continually execute any instruction, control I/O operation, suppress overlap, and step through instructions cycle-by-cycle.

Figure 105 shows the layout of indicators and controls for this panel.

Indicators

Indicators on the customer engineering test panel are incandescent. Indicators connected to triggers are on when the trigger is on. Indicators concerned with the registers and counters signify a one when they are on and a zero when off.

Address Register

These indicators show the contents of the address register.

Cycle Time

The cycle time indicators show in which cycles the computer is at the time: I, E, L, II, or B.

Multiple Time Error

This lamp will come on if:

1. Two cycle times other than L and B time are on simultaneously, or
2. II time occurs at any time other than E or L time.
3. CPU and channel clocks lose synchronization (Systems 08.00.17.1).

Tally Counter

The tally counter, normally reset to one, is used for both single- and double-precision floating-divide instructions. The tally counter indicators also reflect the status of the fact triggers. The fact triggers are used for single-precision floating add, subtract, and multiply, and for all double-precision floating-point instructions. The indicators are read as a three-position counter. Tally counter indicators two and three on indicate fact three. Indicator one only on indicates fact four (Systems 02.13.61.1 and 02.10.21.1).

DPS

The DPS indicators reflect the status of the double-precision sync triggers one through three. The DPS steps, zero through three, control operations during double-precision floating-point operations (Systems 02.13.63.1).

Carry

This lamp indicates a Q carry during POD 30 and POD 40 instructions (fixed and floating point add and subtract). See Systems 02.02.40.1.

FP

This lamp and the trigger that feeds it are not used on the 7094 II (Systems 02.10.29.1).

9 Carry

This lamp indicates the condition of the adder 9 carry trigger (Systems 02.10.37.1). The adder 9 carry trigger is set during the FRN instruction if there is a bit in MQ9 and an adder 9 carry.

Acc-9P

This lamp indicates the condition of the accumulator 9P register position (Systems 02.03.09.1).

Q Carry

This lamp indicates the condition of the Q carry trigger (Systems 02.10.36.1). This trigger comes on whenever a carry occurs out of adder Q.

Master Stop

This lamp indicates the condition of the master stop trigger (Systems 04.20.11.1).

End Operation

This lamp is on whenever the end operation trigger is on (Systems 08.00.09.2).

Diagnostic Mode

This lamp is on whenever the diagnostic mode trigger is on (Systems 03.08.17.3).

Overlap Mode

This lamp is on whenever the computer is in overlap mode (Systems 03.08.17.3).

Switches

Switches on the customer engineering test panel aid in trouble shooting machine failures, and in testing the computer in various modes of operation.

I/O Interlock Switch

This switch is effective only with the system in manual. When both the auto-manual and I/O interlock switches are set to manual, the computer will stop after executing each instruction. When the I/O interlock switches are set to automatic, the machine will not stop if an I/O device is selected to allow normal operation of the I/O unit, but will stop when the channel disconnects.

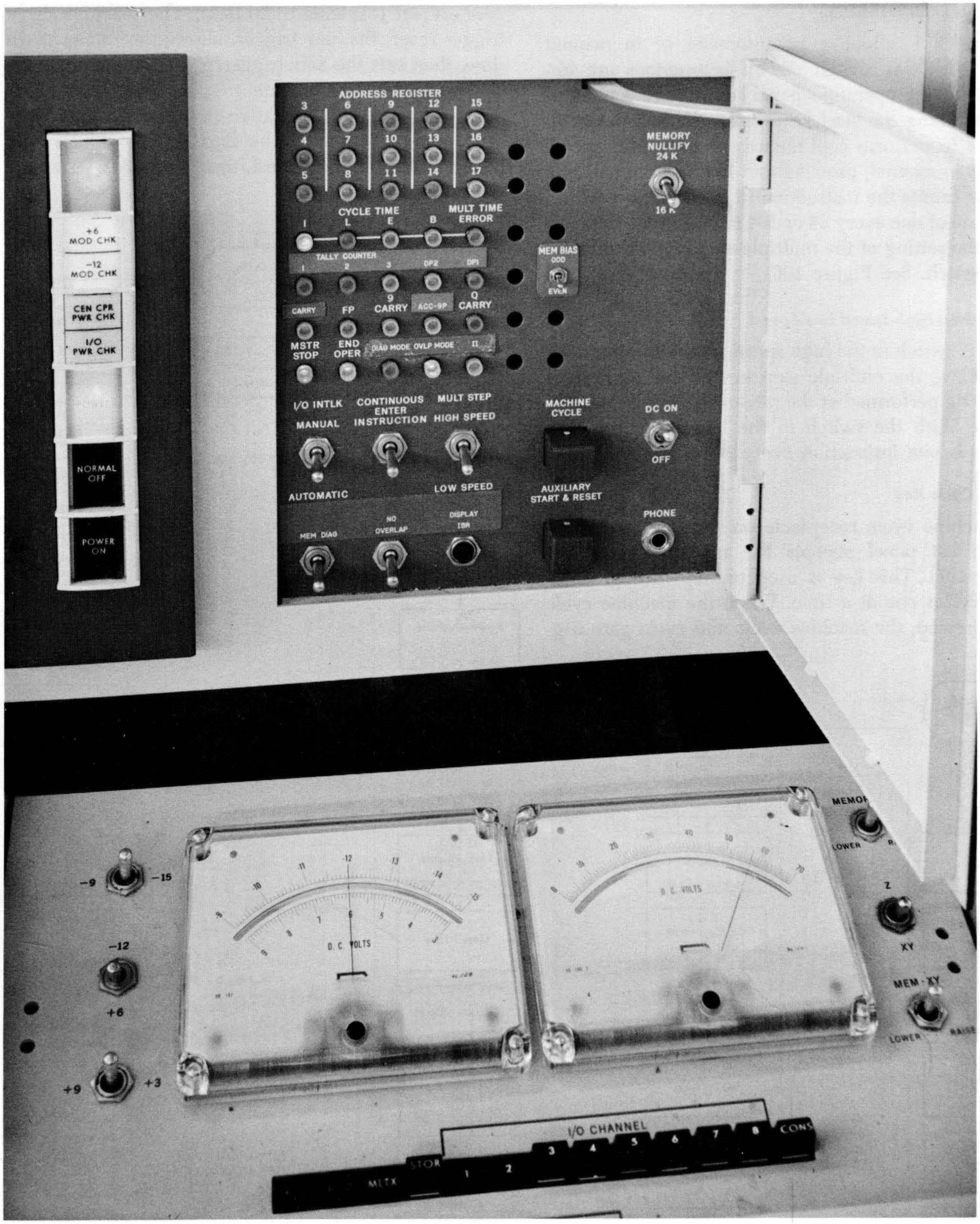


Figure 105. CE Panel

Continuous Enter Instruction

This switch is effective in automatic or in manual status. With this switch on, all instructions are obtained from the operator's panel keys. If CPU is in automatic, pressing the start key causes the instruction in the keys to perform continuously at computer speed. If CPU is in manual, pressing and holding the multiple step key causes the instruction in the keys to perform at the rate of one every 24 or 104 milliseconds, depending on the setting of the multiple-step high-speed low-speed switch. See Figure 106.

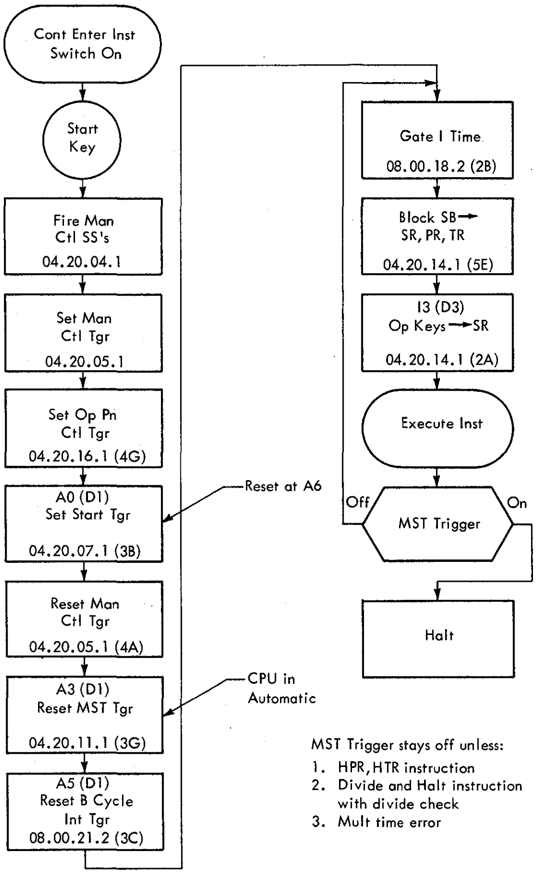
Multiple-Step High-Speed Low-Speed

With this switch in the high speed position, depressing and holding the multiple step key results in instructions being performed at the rate of one every 24 milliseconds. With the switch in the low speed position, the rate is one instruction every 104 milliseconds.

Machine Cycle Key

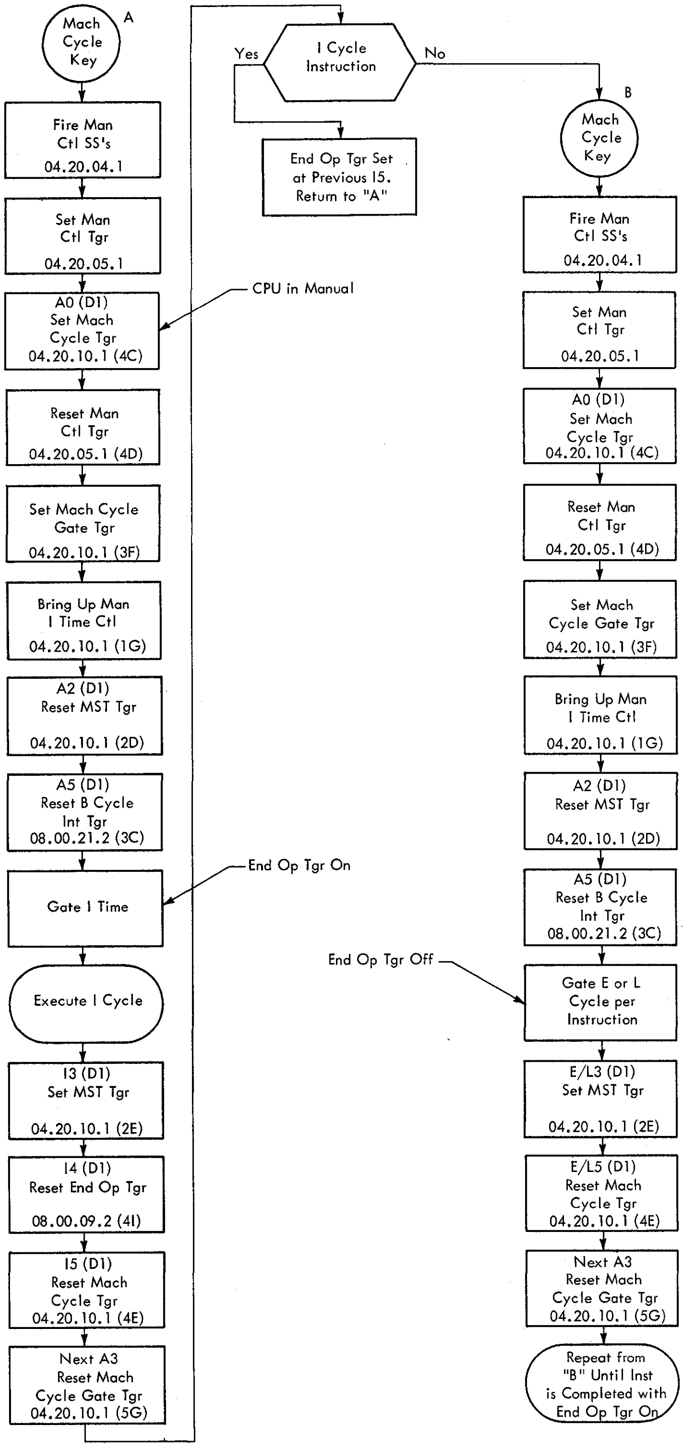
The machine cycle receptacle on the customer engineering test panel accepts the machine cycle key (Figure 107). This key is used to step through individual cycles one at a time. When the machine cycle key is pressed, the machine cycle and cycle gate triggers are set (Systems 04.20.10.1). The machine cycle trigger resets the MST trigger, allows the proper cycle time, then sets the MST trigger.

The machine cycle gate trigger allows two shifts per cycle on a shift operation (Systems 02.10.46.1). It also forces a wait until the shift counter equals zero before ending operation in L time (Systems 08.00.02.2),



Cont Enter Inst Switch On
Start Key
Fire Man Ctl SS's 04.20.04.1
Set Man Ctl Tgr 04.20.05.1
Set Op Pn Ctl Tgr 04.20.16.1 (4G)
A0 (D1) Set Start Tgr 04.20.07.1 (3B)
Reset at A6
Reset Man Ctl Tgr 04.20.05.1 (4A)
A3 (D1) Reset MST Tgr 04.20.11.1 (3G)
CPU in Automatic
A5 (D1) Reset B Cycle Int Tgr 08.00.21.2 (3C)
Gate I Time 08.00.18.2 (2B)
Block SB→ SR, PR, TR 04.20.14.1 (5E)
I3 (D3) Op Keys→SR 04.20.14.1 (2A)
Execute Inst
Off
MST Trigger
On
Halt
MST Trigger stays off unless:
1. HPR, HTR instruction
2. Divide and Halt instruction with divide check
3. Mult time error

Figure 106. Continuous Enter Instruction



Mach Cycle Key
A
Fire Man Ctl SS's 04.20.04.1
Set Man Ctl Tgr 04.20.05.1
A0 (D1) Set Mach Cycle Tgr 04.20.10.1 (4C)
CPU in Manual
Reset Man Ctl Tgr 04.20.05.1 (4D)
Set Mach Cycle Gate Tgr 04.20.10.1 (3F)
Bring Up Man I Time Ctl 04.20.10.1 (1G)
A2 (D1) Reset MST Tgr 04.20.10.1 (2D)
A5 (D1) Reset B Cycle Int Tgr 08.00.21.2 (3C)
End Op Tgr On
Gate I Time
Execute I Cycle
I3 (D1) Set MST Tgr 04.20.10.1 (2E)
I4 (D1) Reset End Op Tgr 08.00.09.2 (4I)
I5 (D1) Reset Mach Cycle Tgr 04.20.10.1 (4E)
Next A3 Reset Mach Cycle Gate Tgr 04.20.10.1 (5G)
Yes
I Cycle Instruction
No
End Op Tgr Set at Previous I5. Return to "A"
B
Mach Cycle Key
Fire Man Ctl SS's 04.20.04.1
Set Man Ctl Tgr 04.20.05.1
A0 (D1) Set Mach Cycle Tgr 04.20.10.1 (4C)
Reset Man Ctl Tgr 04.20.05.1 (4D)
Set Mach Cycle Gate Tgr 04.20.10.1 (3F)
Bring Up Man I Time Ctl 04.20.10.1 (1G)
A2 (D1) Reset MST Tgr 04.20.10.1 (2D)
A5 (D1) Reset B Cycle Int Tgr 08.00.21.2 (3C)
End Op Tgr Off
Gate E or L Cycle per Instruction
E/L3 (D1) Set MST Tgr 04.20.10.1 (2E)
E/L5 (D1) Reset Mach Cycle Tgr 04.20.10.1 (4E)
Next A3 Reset Mach Cycle Gate Tgr 04.20.10.1 (5G)
Repeat from "B" Until Inst is Completed with End Op Tgr On

Figure 107. Machine Cycle Key

and is used to hold up "manual I-time control" (Systems 08.00.18.2).

When the key is not in use, a plug (sent with the system) that shorts pins one and three must be inserted.

Memory-Diagnostic

This switch, when turned on (up), does not place the computer in diagnostic mode but permits the use of the instructions Enter Diagnostic Mode and Transfer (EDAT), and Leave Diagnostic Mode and Transfer (LDAT). When the computer is in diagnostic mode, address position three instead of 17 determines which memory is addressed. Diagnostic mode also blocks XAD to MAR gating (Systems 03.06.28.8). Consequently, only one instruction is fetched from memory during an I cycle. The IBR-loaded trigger is not set, and the II cycle following an I cycle must also fetch its instruction from memory. The switch, when down (off), unconditionally takes the computer out of diagnostic mode and makes EDAT and LDAT act like a NOP (Systems 03.08.17.3).

No Overlap Switch

This switch when turned up (on) does not take the computer out of overlap mode but permits the use of the instructions Enter Overlap Mode and Transfer (ELAP), and Leave Overlap Mode and Transfer (LLAP). The switch, when down (off), places the computer in overlap mode and makes ELAP and LLAP act like a NOP (Systems 03.08.17.3). For the effect on the computer of overlap mode, see the section, "Overlap."

Display IBR

Depressing this key replaces the contents of the storage register with the contents of the IBR (Figure 108).

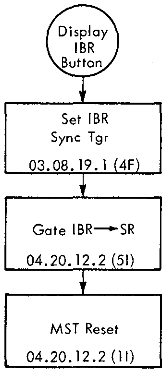


Figure 108. Display IBR

Memory Bias

This switch will select either the odd or even memory for biasing (Systems 09.06.21.1).

Phone Jack

The phone jack, in conjunction with the phone jacks on other units in the system, provides a means of communication between customer engineers working at different units.

Memory Nullify

This switch is used with the compatibility package. When it is in the 16K position, 16K core storage positions are available to the 704 program; in the 24K position, 8K core storage positions are available to the 704 program.

DC On

The DC on switch controls the 400-cycle power supplied only to the 7151 Model 2. Putting this switch in the off position removes all power to the console except the convenience outlets and the reset motor for the operator's keys. All voltages should be normal about ten seconds after turning this switch to the on position.

Marginal Check Panel

The system biasing network can be useful to the customer engineer. The controls are located on the marginal check console on the 7151 Model 2. Any single gate or combination of gates in any single module, or combination of modules, can be biased at the same time. The only exception is the 7302 memory. This module is under control of only one key (A) and the whole module will be biased when this key is pressed.

Each module has a key for selecting it, and each gate has a key with key A for gate A, B for gate B, etc.

In biasing, all gates must be selected before varying the voltage. After the voltage is varied, another gate cannot be selected until returning to normal voltage. If gate A is being biased and you decide to bias gate B instead, gate A MC relay must be dropped out, and the MC voltage must be taken to normal before selecting gate B. If this is not done, gate A MC relay will have a hold through the B MC relay points. This means that both gates A and B would be biased instead of gate B only, as desired.

It is possible to vary the +60 volts in the 7302. These controls are also located in the 7151 Model 2, on the MC panel.

In varying the different voltages, there are two meters to monitor the amount of voltage being varied. One meter is for the +60 in the 7302, and one meter is for the +6 and −12 for the rest of the system. The meters indicate what the voltage is if the MC relays are picked and the points are properly adjusted. While biasing, a periodic check of the voltages should be made. See Figure 105.

Compatibility

With the evolution of new machines, additional features are added to increase their speed and versatility. Attempts are also made to allow existing programs to be run unaltered on the new system. Cases of com- be run unaltered on the new system. Cases of com- and again with the 7094 II system. In each instance, it is "upward compatibility," in which programs from a lower system can be executed on a higher system, not in reverse.

7090/7094 II Index Register Compatibility

The main area of compatibility change within the 7094 II is the availability of four additional index registers. The three tag bits (18, 19, 20) on the 7090 and lower systems indicate any one or combination of three index registers. On the 7094 II the same three bits, when decoded in the seven index register mode, indicate any one of seven index registers:

TAG BITS			7094 II INDEX	7090 INDEX
18	19	20	REGISTERS	REGISTERS
0	0	0	None	None
0	0	1	XRA(1)	XRA(1)
0	1	0	XRB(2)	XRB(2)
0	1	1	XRD(3)	XRA, XRB(1, 2)
1	0	0	XRC(4)	XRC(4)
1	0	1	XRE(5)	XRA, XRC(1, 5)
1	1	0	XRF(6)	XRB, XRC(2, 4)
1	1	1	XRG(7)	XRA, XRB, XRC(1, 2, 4)

To allow the 7094 II to operate in either a three or seven index register mode, two instructions are available to the programmer: Enter Multiple Tag Mode (EMTM), and Leave Multiple Tag Mode (LMTM). The normal power-on mode of operation is the multiple tag mode (7090, three index register mode). This allows 709/7090 programs to be loaded and executed in the 7094 II. If the programmer wishes to take advantage of the additional index registers, he may give the LMTM instruction. Depression of the clear key will set the multiple tag mode trigger.

Enter Multiple Tag Mode EMTM − 0760 . . . 0016 (I, L)

This instruction sets the multiple tag mode trigger, and lights the multiple tag mode lamp on the operator's console. Mutliple tag mode limits the program to use of index registers A, B, and C (1, 2, 4), and permits OR'ing the index registers. This mode is the normal power-on mode of operation; it may be entered also by depressing the clear key. See Figure 109.

Leave Multiple Tag Mode LMTM + 0760 . . . 0016 (I, L)

This instruction resets the multiple tag mode trigger and turns off the multiple tag mode lamp on the operator's console. Resetting the multiple tag mode trigger places the computer in the seven index register mode. The program may now use any one of seven index registers. OR'ing the index registers is inhibited. The EMTM instruction or depression of the clear key will take the computer out of the seven index register mode. See Figure 109.

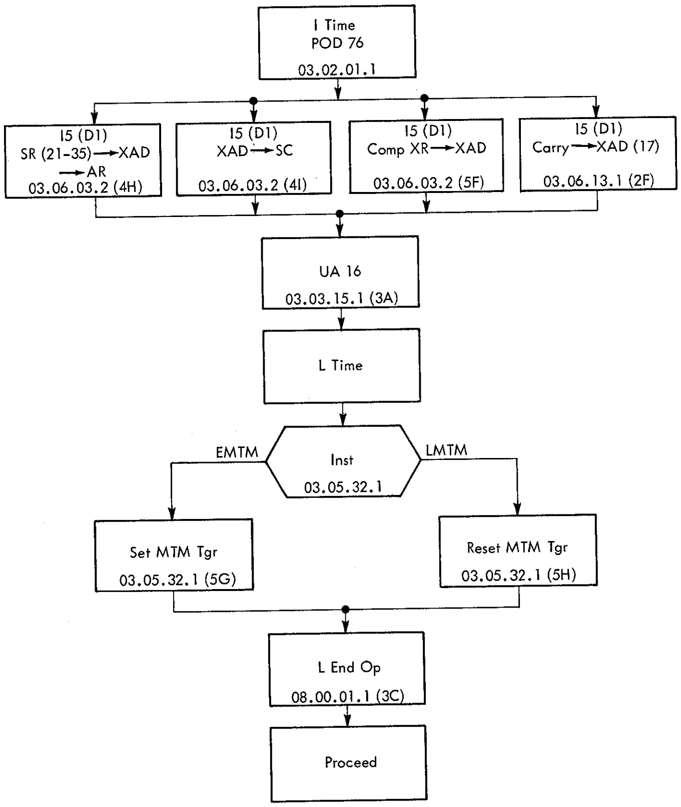


Figure 109. Enter Multiple Tag Mode (EMTM); Leave Multiple Tag Mode (LMTM)

704/7094 II Floating-Point Trap Mode Compatibility

Floating-point instructions are available on the 704, but floating-point trapping is not a standard feature. Overflow and underflow conditions require instructions such as TOV, TNO, or TQO to determine the resultant status of the characteristics in the accumulator and MQ.

The FP trap is a standard feature in subsequent computers. Therefore, for 704 programs to run unmodified on 709, 7090, 7094, and 7094 II systems, the FP trap mode of operation must be nullified. Two instructions are provided to control the mode of operation: leave floating trap mode (LFTM) and enter floating trap mode (EFTM).

The 7094 II is normally in the floating-point trap mode. The clear, reset, or load key will place the computer in the floating-point trap mode of operation (02.10.71.1).

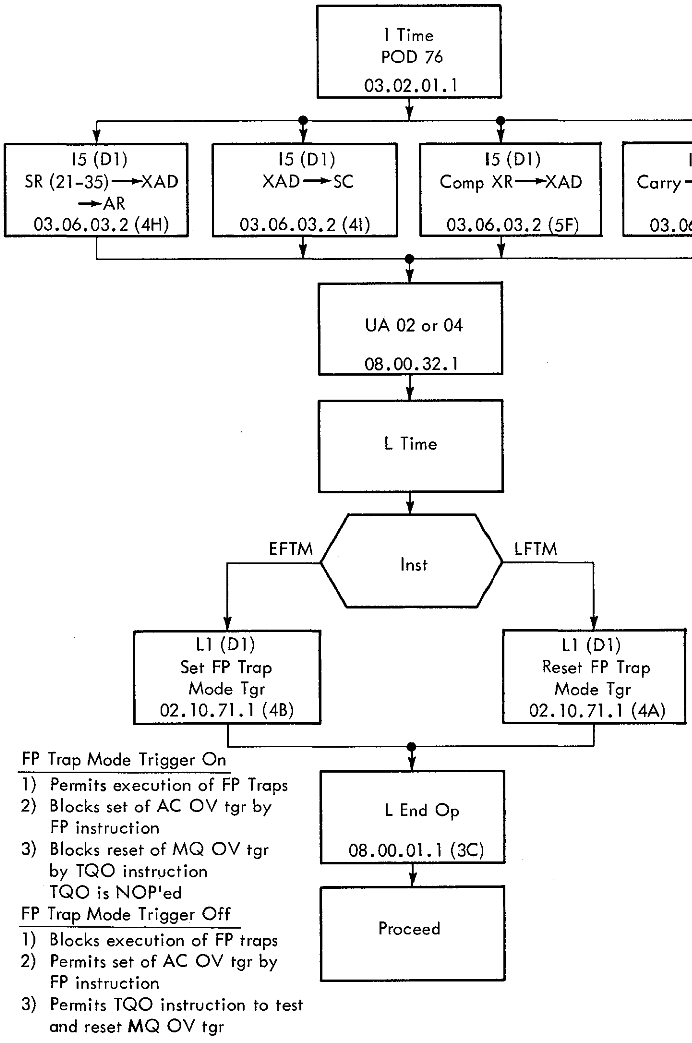
Enter Floating-Trap Mode EFTM — 0760 . . . 0002 (I, L)

This instruction sets the FP trap mode trigger, placing the computer in the FP trap mode of operation. Overflows and underflows while performing floating-point instructions are automatically trapped. See the section, "Trapping" for floating-point trap operation, identification bits and trap addresses. FP trap mode is the normal power-on mode of operation; it is also entered by depression of the clear, reset or load keys. See Figure 110.

The FP trap mode trigger on: (1) Permits execution of floating point traps; (2) Blocks set of accumulator OV trigger by an FP instruction; (3) Blocks reset of MQ OV trigger by the TQO instruction. TQO is NOP'ed.

Leave Floating-Trap Mode LFTM — 0760 . . . 0004 (I, L)

This instruction resets the FP trap mode trigger, taking the computer out of FP trap mode. See Figure 110.



FP Trap Mode Trigger On

1) Permits execution of FP Traps
2) Blocks set of AC OV tgr by FP instruction
3) Blocks reset of MQ OV tgr by TQO instruction TQO is NOP'ed

FP Trap Mode Trigger Off

1) Blocks execution of FP traps
2) Permits set of AC OV tgr by FP instruction
3) Permits TQO instruction to test and reset MQ OV tgr

Figure 110. Enter Floating-Trap Mode (EFTM); Leave Floating-Trap Mode (LFTM)

The FP trap mode trigger off: (1) Blocks execution of FP traps; (2) Permits the set of accumulator OV trigger by an FP instruction; (3) Permits the TQO instruction to test and reset the MQ OV trigger.

704/709-7090/7094 II I/O Compatibility

I/O compatibility is accomplished by a combination of machine hardware, instructions, and an applied programming support package. The I/O compatibility program is normally located starting at 40,000$_8$ (on 32K systems). All select, sense, copy or LDA instructions are not directly executed, but cause an automatic trap into the compatibility program. To protect the compatibility program from possible destruction by the object program, 7094 II circuitry nullifies the upper half of core storage. Memory nullify, select trap, and copy trap modes of operation are established by instructions available to the programmer.

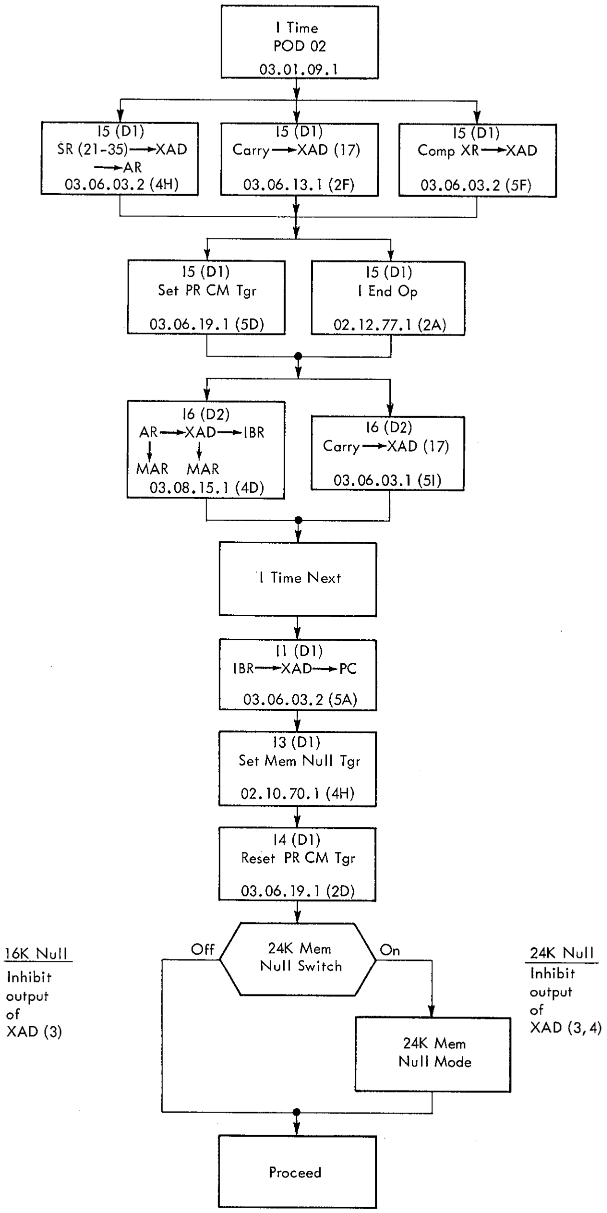


Figure 111. Enter Storage Nullification and Transfer (ESNT)

Enter Storage Nullification and Transfer ESNT — 0021 (I)

This instruction turns on the memory nullify trigger and the simulate light on the operator's console, and transfers to the address specified. When this trigger is on, the upper half or three-quarters of storage, depending upon the position of the nullify switch, cannot be referenced by the 704 or 709 program. The trigger can be reset by depressing the clear, reset, or load keys, by the LSNM instruction, or by execution of either a select or copy trap. ESNT is not affected by the transfer trap mode. See Figure 111.

Memory nullify mode blocks access to the upper half of storage by inhibiting the output of index adder position three (Systems 03.05.40.1). If the 24K-16K memory nullify feature is installed, the upper three-quarters of memory can be nullified with the switch in the 24K position. This is done by inhibiting the output of index adder positions three and four. The display storage and clear keys override the effect of memory nullify mode (Systems 02.17.76.2) so that it is possible to display or clear all of storage.

Leave Storage Nullification Mode LSNM — 0760 . . . 0010 (I, L)

This instruction resets the memory nullify trigger, allowing reference to all of storage. The setting of the memory nullify 24K-16K switch is ignored. See Figure 112.

Enter Select Trap Mode ESTM — 0760 . . . 0005 (I, L)

This instruction sets the I/O trap trigger (Systems 02.10.70.1), and lights the simulate indicator on the operator's console. While in this mode, I/O select and sense instructions are trapped. Instructions that will be trapped are: RDS, WRS, BSR, BSF, WEF, REW, RUN, SDH, SDL, BTT, ETT, SPR, SPT, and SPU. See Figure 112.

The location +1 of the trapped instruction is stored in location 10,000$_8$ or 40,000$_8$ (depending on the size of core storage), and program control is transferred to 10,001$_8$ or 40,001$_8$. When the trap occurs, the memory nullify, I/O trap, and copy trap triggers are reset and the simulate light is turned off. The triggers and indicator may be turned off also by the reset, clear, or any of the load keys. The select trap operation is covered in detail in a subsequent section.

Enter Copy Trap Mode ECTM — 0760 . . . 0006 (I, L)

This instruction sets the copy trap trigger (Systems 02.10.70.1), and lights the simulate indicator on the operator's console. While in this mode, CPY, CAD, and LDA instructions are trapped. See Figure 112.

The location +1 of the trapped instruction is stored in location 10,000$_8$ or 40,000$_8$ (depending on the size of core storage) and program control is transferred to location 10,002$_8$ or 40,002$_8$. When the trap occurs, the memory nullify, I/O trap, and copy trap triggers are reset and the simulate light turned off. The triggers and indicator may also be turned off by the reset, clear, or by any of the load keys. The copy trap operation is covered in detail in a subsequent section.

I/O Select Trap Operation

In the I/O select trap mode of operation, all instructions which would either select an I/O device or sense conditions concerned with that device are blocked, and the computer program is trapped to location 40,001$_8$. At the time of the trap, the trapping instruction's location +1 is stored in the address portion of location 40,000$_8$. With these facts, the I/O compatibility program can determine the instruction causing the trap and take appropriate action. See Figure 113.

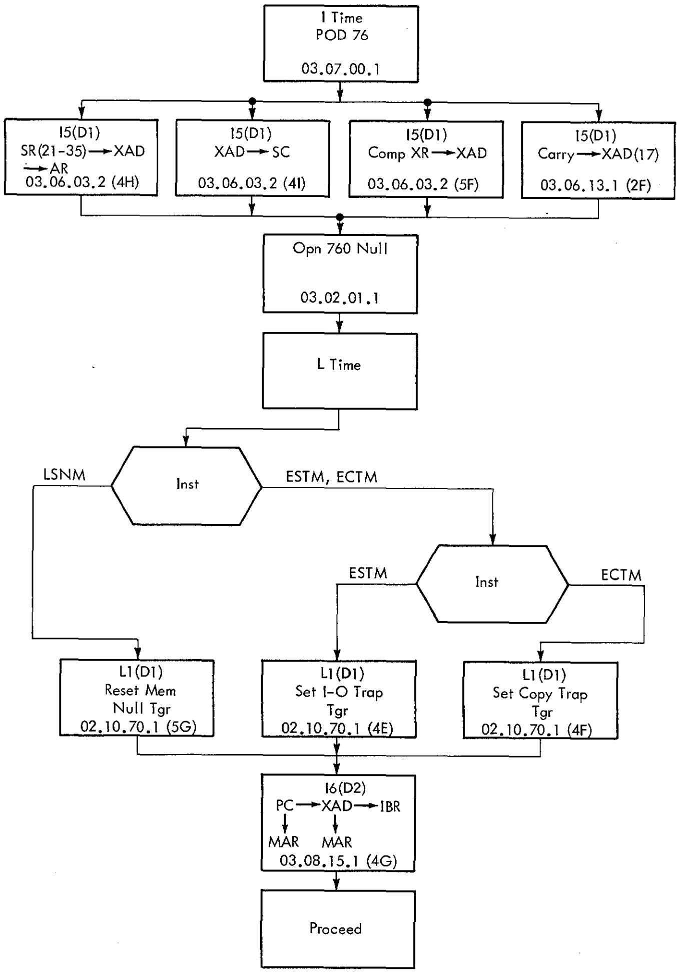


Figure 112. Leave Storage Nullification Mode (LSNM); Enter Select Trap Mode (ESTM); Enter Copy Trap Mode (ECTM)

The various select and sense instructions are all based on POD 76 decoding, but are detected at several circuit locations. SOD outputs (other than SOD 00) and BTT/ETT (a primary operation 760) are recognized as "SOD selects," as shown on Systems 03.02.00.1. They include:

SOD	02	RDS(+0762)
SOD	04	BSR/BSF(±0764)
SOD	06	WRS(+0766)
SOD	10	WEF(+0770)
SOD	12	REW/RUN(±0772)
SOD	16	SDH/SDL(+0776..cc2uu)
POD	760	BTT/ETT(±0760..cc000)

The result of these conditions would normally allow gating the various class selects out to the I/O devices (i.e., tape class address, CR class address). When the computer is in I/O select trap mode, these class addresses are not sent to the various devices; instead, "trap on I/O" is activated (02.10.71.1), and the computer proceeds with the trap operation. To be accurate, "CP class address 34" (03.02.00.1) is not blocked to the channel. This causes no trouble since there is no corresponding channel designation (SR 23, 24, 25, or 26) to go with the select instruction in 704 programs. Without a complete set of necessary signals sent to the attached channels, the channel cannot set its class and unit select triggers.

I/O select trap mode, by degating the various class addresses, inhibits setting the channel L-time trigger (08.00.29.1), which in turn prevents resetting the MF go trigger.

Punch and printer sense instructions are detected early in the L cycle (02.10.71.1). They set the I/O trap trigger, which degates the corresponding sense address and places the computer in a trap condition.

Two instructions available to 704 users for testing end-of-tape or tape redundancy conditions (ETT, RTT) also set the I/O trap trigger, (02.10.71.1) placing the computer in a trap condition.

Selecting the CRT on a standard computer will set the I/O check trigger and force L end op. However, if the drum is selected, and the computer is in I/O trap mode, a trap will occur. The compatibility program can simulate the drum. If the computer is not in I/O trap mode, selecting the drum will set the I/O check trigger and force L end op.

When an I/O select trap is started, blocking the class address prevents setting the channel L-time trigger. Instead, the computer proceeds to L time and forces a store and trap operation (02.10.54.1). The memory nullify trigger is reset so that it is possible to store and trap in the high end of storage. During the L cycle, address register three is set, providing a value of

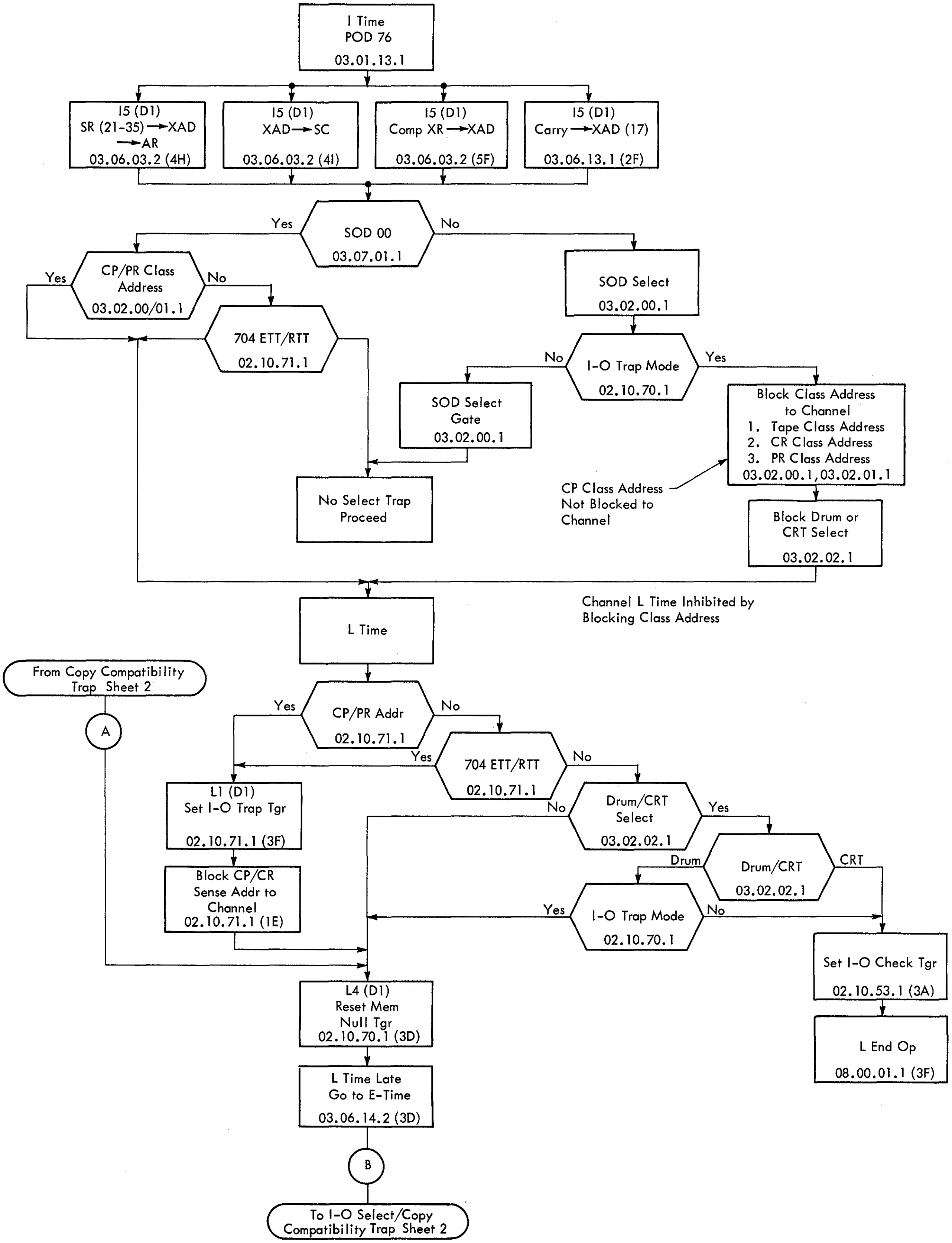


Figure 113, Sheet 1 of 2. I/O Select/Copy Compatibility Trap

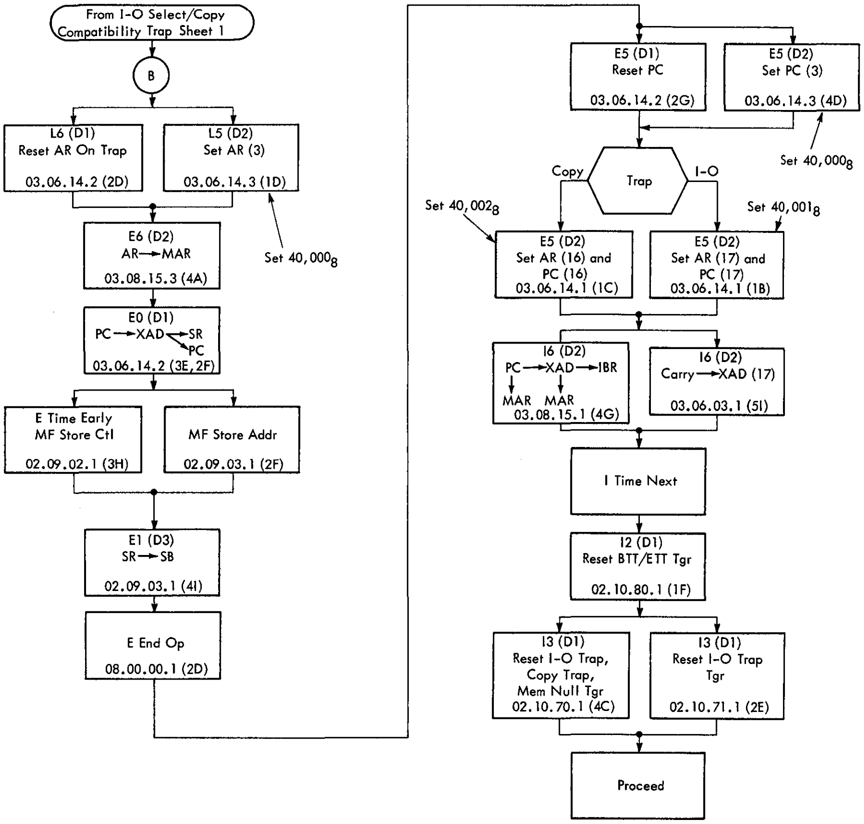


Figure 113, Sheet 2 of 2. I/O Select/Copy Compatibility Trap

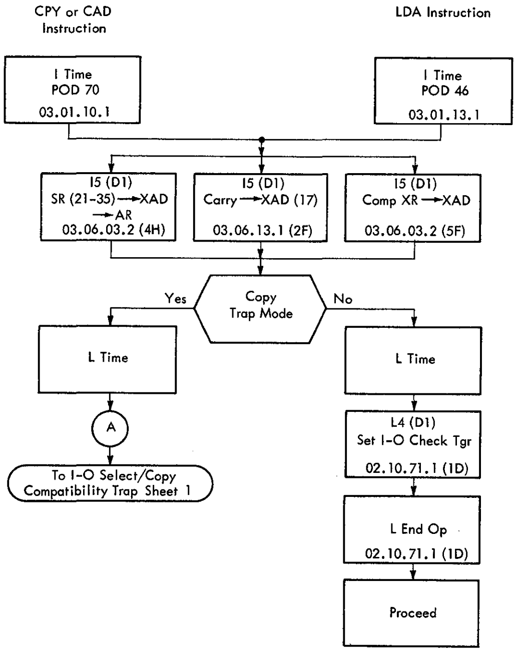


Figure 114. Copy Compatibility Trap

$40,000_8$. Store and trap forces the computer to E time. Early in the E cycle the program counter contents are routed to the storage register via the index adders, then to the storage bus. The location +1 of the instruction causing the trap has now been stored in the address portion of location $40,000_8$.

During the E cycle, bits are placed in program counter positions three and 17. An address of $40,001_8$ is now available. The program counter is gated to MAR, and the next instruction to be processed is from $40,001_8$. This starts the compatibility program for I/O select trap. The BTT, ETT, I/O trap, copy trap and memory nullify triggers are reset during the following I cycle.

I/O Copy Trap Operation

In the I/O copy trap mode of operation, all copy (CPY or CAD) and locate drum address (LDA) instructions are blocked. The computer is forced instead into a store and trap operation. The location +1 of the instruction causing the trap is stored in the address portion of location $40,000_8$, and the computer traps to $40,002_8$ (Figure 114). If these instructions are encountered when the computer is not in copy trap mode, the I/O check trigger is set during L time and the computer ends operation.

When a copy trap is started, the computer proceeds to L time and forces a store and trap operation (02.10.54.1). The memory nullify trigger is reset so it is possible to store and trap in the high end of storage. During the L cycle, address register three is set, providing a value of $40,000_8$. Store and trap forces the computer to E time. Early in the E cycle the program register contents are routed to the storage register via the index adders, then to the storage bus. The location +1 of the instruction causing the trap has now been stored in the address portion of location $40,000_8$.

During the E cycle, bits are placed in program counter positions three and 16. An address of $40,002_8$ is now available. The program counter is gated to MAR, and the next instruction to be processed is from $40,002_8$. This starts the compatibility program for an I/O copy trap. The BTT, ETT, I/O trap, copy trap and memory nullify triggers are reset during the following I cycle.

I/O Compatibility II Program Operation

The compatibility II program and special 7090/7094 II compatibility instructions make possible the execution of 704 and 709 programs by the 7094 II. The compatibility instructions are used by the compatibility program to monitor the 704 or 709 program and to make changes necessary for the proper execution of certain 704 and 709 instructions by the 7094 II. One half of the 7094 II core storage capacity is required for the modification processes (Figure 115), while the 704 or 709 program occupies the remaining half.

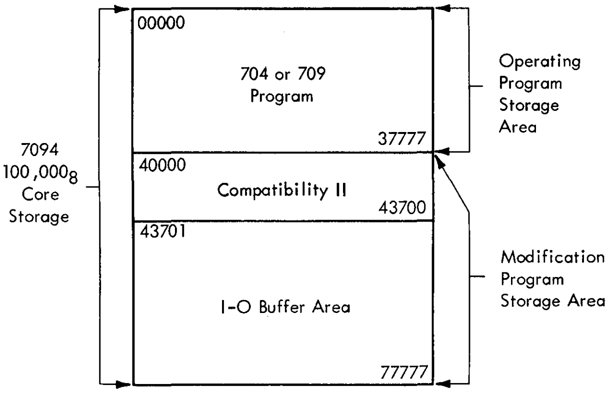

00000
704 or 709 Program
37777
40000
Compatibility II
43700
43701
I-O Buffer Area
77777
7094 $100,000_8$ Core Storage
Operating Program Storage Area
Modification Program Storage Area

Figure 115. Compatibility Core Storage Assignments

Basic Operation

The 704 I/O address scheme is different from that of the 709, 7090, or 7094 II. The 704 method of execution of I/O operations also differs from that of the 709, 7090, or 7094 II; 704 or 709 may use drums for storage, where the 7090/7094 II has no provision for drum storage. Because there are no drums, and the addressing scheme differs, it is necessary to modify 704 I/O addresses, simulate 704 I/O methods of execution, and simulate 704 or 709 drum addresses to run a 704 or 709 program on the 7094 II. The compatibility II program performs the modification and simulation automatically.

Under control of compatibility II, the $100,000_8$ core storage is divided into two halves. The execution of the 704 or 709 program is carried out in the lower half, with no normal programmed reference to locations $40,000_8$ through $77,777_8$. Entry to the compatibility program (stored in 40,000 through 43,700) is gained through trapping 704 select or copy instructions, or 704/709 drum instructions. The remainder of storage (43,701 through $77,777_8$) serves as a buffer controlled by the compatibility program. Data words are collected in the buffer to be treated as a block of words in 7094 fashion. Simulated drum addresses also appear in the buffer area (Figure 116).

Storage Nullification

The core storage capacity available to the operation program (Figure 115) is limited to $40,000_8$ locations. Larger addresses, when developed, are reduced by $40,000_8$.

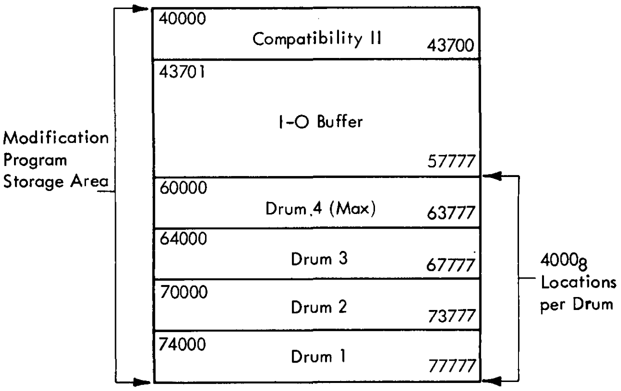


Figure 116. Modification Storage Assignments

Compatibility Traps

Access to the compatibility program and to the I/O buffer is gained only through trapping a 704 or 709 instruction. Once an instruction is trapped, control moves to the compatibility program, and the higher part of core storage is no longer nullified. The compatibility program makes the necessary modifications, returns to storage nullification, and transfers control back to the operating program.

Two types of instructions can be trapped: selects and copies. The 704 select instructions that can be trapped are RDS, WRS, WEF, BSF, BST, REW, SPT, SPR, SPU, ETT, and RTT. The 704 and 709 copy instructions that can be trapped are CPY, CAD, and LDA.

Restrictions

Although the compatibility II program allows execution of 704 and 709 programs on the 7094 II, certain precautions must be taken to permit correct operation.

704 Programs on the 7094

1. The program must fit into $40{,}000_8$ storage locations.

2. The high-order position of the address field of the instruction word should be blank.

3. The indirect address portion of instruction should not indicate indirect addressing.

4. Except for magnetic drums, the 7094 II must include all items of equipment specified by the 704 program.

5. The MQ register cannot be used when a CPY or RDS instruction is trapped following an RDS tape instruction. The contents of the MQ at that time are destroyed and the MQ remains under control of the compatibility program until the last CPY has been processed.

6. Only +0766 - - - - - 00333 is recognized as an I/O delay instruction.

7. The standard version of the compatibility program makes no provision for the correct restoring of index registers after processing a trapped I/O instruction with a multiple tag. Effective address computation in the compatibility program produces a logical OR sum of the contents of the specified index register. This would not result in normal execution of a program on the 704. A special multiple tag patch is available. This allows proper execution of 704 programs which contain I/O instructions with multiple tags, but increases the time required to process each copy by one and one-half times the normal copy simulation time.

8. Echo checking is simulated, and always checks without regard for what was actually written.

9. Sense printer tests are always affirmative. A sense printer patch is available to permit an affirmative result only on alternate tests when it would normally be affirmative in a 704 program.

10. The load cards, load tape, and load drum manual operations must not be operated after compatiblity II is loaded. These operations must be simulated to avoid disabling the compatibility traps.

709 Programs on the 7094

It is not necessary to use compatibility II with 709 programs on the 7094 II unless the 709 program specifies drums; however, the increased speed of the 7094 II causes shrinkage of computed delays and may result in improper I/O operations that depend on computed delays.

Channel traps that occur when the 7094 II is controlled by the compatibility program will transfer control back to the 709 program before the compatibility program has completed its modifications.

As in 704/709, the 7094 II must include all items of equipment specified by the 709 program. Magnetic tape density must be set manually at the tape unit.

Program Operation

The following events occur in order as the compatibility II program is put into operation:

1. Compatibility program is loaded with manual load cards.

2. Control card assigns tape addresses and drum locations.

3. Compatibility program loads 704 or 709 program by simulating load cards, load tape or load drum.

4. Trailer card (part of compatibility II deck) is loaded by compatibility program to terminate compatibility operation when 704 or 709 program has completed its run.

5. The 704 or 709 operation program starts, with the 7094 II in storage nullification, select trap, and copy tape mode.

6. Select, copy, and drum instructions are trapped in the compatibility program, where necessary modifications are made. The 704 or 709 program then resumes.

7. The 704 or 709 program ends, and information read from the trailer card terminates the compatibility operation.

Preparing and Loading the Compatibility II Program

The compatibility II program deck contains all the routines necessary to modify I/O addresses, I/O operations and storage location addresses, and to properly terminate the operation program. However, proper operation of the compatibility program depends upon its being informed what 7094 II tape units are to be used and how many (if any) drums are to be simulated. This is the purpose of the control card. This card is inserted between the compatibility program and the 704 or 709 operating program.

Columns 1 through 10 are punched to assign the 7094 II equivalents of 704 tape units 1 through 0. Assignment is made by channel A, B, C, D, E, or F, with the first assignment (starting in column 1) being tape unit 1 (on that channel), the second assignment to that channel being tape unit 2, and so on. For example, a control card punched A, A, blank, C, blank, B, E, C, A, B in columns 1 through 10 indicates the following assignments:

704 TAPE UNITS	709 TAPE UNIT
1	A1
2	A2
Not used	Unassigned
4	C1
Not used	Unassigned
6	B1
7	E1
8	C2
9	A3
0	B2

If columns 1 through 10 are all blank, the compatibility program assigns all 704 tapes as though an A were punched in odd-numbered columns and a B in even-numbered columns.

A 1, 2, 3, or 4 may be punched in column 11 to indicate the number of drums to be simulated. Drum addresses are automatically assigned (Figure 116).

Following the preparation of the control card, it is inserted between the compatibility II deck and the operating program deck. The trailer card follows the operating program deck. The assembled deck is readied in the card reader, console panel key 35 is depressed, and load cards key is depressed. The trailer card remains in the card reader until the operating program is complete. When a final 704 program halt occurs, depress the load cards key to read the trailer card. The compatibility program will then complete the previous job and halt at symbolic location NEXT. At this time, if other 704 programs remain to be processed, set the console panel keys and tape units as required for the next program. Depress the start key to read in the new control card and 704 program to be processed.

Reading a Record

When the RDS instruction is encountered in the 704 program, the 7094 II prevents the selection of the I/O device and forces a store and trap operation. The location +1 of the select instruction is stored in location 10,000$_8$ or 40,000$_8$, and program control is transferred to location 10,001$_8$ or 40,001$_8$. This is a function of the STR instruction in compatibility mode. The compatibility program provides 7094 II instructions to select the unit and read the entire record into the I/O buffer. Control is then returned to the 704 program and, as CPY instructions are encountered, words will be taken from the buffer and stored. As each CPY instruction is encountered, the 7094 II again forces an STR instruction, the location +1 of the CPY instruction is stored in 10,000$_8$ or 40,000$_8$, and program control is transferred to 10,002$_8$ or 40,002$_8$. The compatibility program will control the movement of words from the buffer to storage.

If an EOF mark is encountered, an EOF skip is simulated when the first CPY is trapped. Zeros will be placed in the MQ.

NOTE: This would have to be a separate select instruction because the 704 can read only one record at a time.

The compatibility program uses the AC when simulating CPY, CAD, or LDA. However, it saves the original contents and restores them when returning to the 704 program.

The trapping of a CPY after the last word of a record has been transferred causes the simulation of an end-of-record skip.

When the 7094 II traps on a select or copy instruction, it resets the select trap, copy trap, and memory nullify triggers. The memory null trigger is reset as soon as the trap occurs so that reference can be made to the high-order portion of core storage. These triggers are turned on again before returning control to the 704 program.

Writing a Record

When a WRS instruction is encountered in the 704 program, it is trapped, the address of the select instruction is converted, and control is returned to the 704 program without selecting the I/O unit. Subsequent CPY instructions are trapped and words are transmitted to the buffer area. These words are not written until another select instruction is encountered. This method of operation insures that the entire previous record has been transmitted to the buffer. If this was the last record to the 704 program, it would not be written until the trailer card is read.

When WRS instructions for writing two or more tapes simultaneously are trapped, the writing is simulated as follows:

1. The compatibility program sets up a table consisting of all the tape units to be simultaneously written.
2. Subsequent CPY or CAD instructions cause transmission of words from the 704 program into the buffer area.
3. When the next select instruction occurs following the CPY instructions associated with the simultaneous writing, the record waiting in the buffer area is written on the indicated tape units. Execution of the 704 program is delayed until the writing is initiated for all tape drives.

Other Operations

When any of the following instructions are trapped, the equivalent 7094 II operation is performed immediately, and control is returned to the 704 program: BSF, BSR, REW, WEF.

The compatibility program keeps a record of end-of-tape indications and executes the appropriate transfer if an ETT instruction is found in the 704 program.

A redundancy check on any channel when an RTT instruction is trapped will indicate the existence of a redundancy to the 704 program.

If a write operation is waiting to be executed and a SPT, SPR, or SPU instruction is given, it will be executed with the write operation. If a write operation is not waiting to be executed when the instruction is trapped, the instruction is executed with the next write operation encountered.

Drum Simulation Procedure

When a magnetic drum(s) is to be simulated, the compatibility program reserves a block of locations in the uppermost part of storage to be used for simulating drums. The size of the block reserved depends on the number of drums to be simulated (maximum of four) with $4{,}000_8$ locations reserved for each drum (Figure 116). A block of $4{,}000_8$ locations is assigned for use as a particular drum according to the order in which drum references are encountered. The lowest block of locations is used for the first drum referred to, the next lowest for the second, and so forth. If the $4{,}000_8$ allocated locations are exceeded, references to addresses over $4{,}000_8$ by CPY or CAD will be written by starting in location 00000_8.

When a select drum instruction is trapped and drums are to be simulated, the compatibility program determines the block of locations assigned to the drum to be selected. The next LDA instruction trapped will then specify the location within the selected block at which transmission is to begin. Subsequent trapping of CPY or CAD instructions causes direct transmission of words between the 704 program and the simulated drum.

NOTE: Although up to four drums may be simulated, only the number required by the program should be indicated in the control card. This is desirable in order to make the buffer area as large as possible for tape I/O.

24K Null Compatibility II

The 7094 II has an optional feature which will nullify storage at 8K as well as 16K. The larger nullified area of storage increases the tape buffer size while simulating drums. Reducing the address field to 13 binary bits eliminates 8K indexing problems with complement addressing. In the standard compatibility program, the top part of core storage is reserved for simulated drums. In the 24K nullification version, the simulated drums are located between 8K and 16K.

A switch on the CPU console controls storage nullification. The two positions on this switch are labeled 24K and 16K. With the switch in the 24K position, 8K (lower) is available to the 704 program, and 24K (upper) is available to the 24K compatibility II program. Note that traps initiated by select-copy-LDA- or sense-type instructions are not affected by the 24K/16K switch; all compatibility traps:

1. Store contents of program counter in 21-35 of location $40{,}000_8$.
2. Take next instruction from location 40K + 1 or 2.

The purpose of the switch is to de-activate index register position four and position four of the program counter and address register. Thus, 8K 704 programs that incorporate iterative routines (loops in conjunction with index registers) will operate correctly on a 7094 II with:

1. The 24K/16K switch in the 24K position.
2. Use of the 24K compatibility II deck.

Compatibility Diagnostics

Two diagnostics are available for checking the system's ability to run compatibility: XCOMC and XN16A.

XCOMC: This diagnostic checks the system's ability to nullify a 32K storage at 16K and to nullify a 32K storage at 8K, if the feature is installed on the 7094 II. This diagnostic will check the system's ability to trap when encountering 704 select, sense, CPY, CAD, or LDA instructions and to perform floating-point operations as the 704 would do them.

XN16A: This diagnostic is used on a 7094 II with a 16K storage; it performs the same tests as XCOMC.

Appendix. Alphabetic Listing and Index of 7094 II Instructions

OPERATION CODE						SYSTEMS	OVERLAP	
ALPHA	OCTAL	INSTRUCTION	CYCLES	INDEX	IA	PAGE	I LAP	E/L LAP
ACL	0361	Add and Carry Logical Word	2 I,E	X	X	02.10.02.1	X	X
ADD	0400	Add	2 I,E	X	X	02.09.91.2	X	X
ADM	0401	Add Magnitude	2 I,E	X	X	02.09.93.1	X	X
ALS	0767	Accumulator Left Shift	② I,L-L	X	—	02.09.70.1	X	X
ANA	—0320	AND to Accumulator	3 I,E,L	X	X	02.09.46.1	—	—
ANS	0320	AND to Storage	4 I,E,L,E	X	X	02.09.46.1	—	—
ARS	0771	Accumulator Right Shift	② I,L-L	X	—	02.09.70.1	X	X
AXC	—0774	Address to Index Complemented	1 I	—	—	03.06.07.1	X	—
AXT	0774	Address to Index True	1 I	—	—	03.06.13.2	X	—
BSF	—0764	Backspace File	2 I,Lc	X		06.01.12.1	—	—
BSR	0764	Backspace Record	2 I,Lc	X		06.01.12.1	—	—
BTT	0760..c000	Beginning of Tape Test	3 I,Lc,Ic	X		02.10.80.1	—	—
CAL	—0500	Clear and Add Logical Word	2 I,E	X	X	02.10.02.1	X	X
CAQ	—0114	Convert by Addition from MQ	⑧ I,L,E-E	—	—	02.09.49.1	—	—
CAS	0340	Compare AC with Storage	2 I,E	X	X	02.09.41.1	X	X
CHS	0760..0002	Change Sign	2 I,L	X	—	02.09.57.1	—	—
CLA	0500	Clear and Add	2 I,E	X	X	02.10.02.1	X	X
CLM	0760..0000	Clear Magnitude	2 I,L	X	—	02.09.57.1	—	—
CLS	0502	Clear and Subtract	2 I,E	X	X	02.10.02.1	X	X
COM	0760..0006	Complement Magnitude	2 I,L	X	—	02.09.57.1	—	—
CRQ	—0154	Convert by Replacement from MQ	⑧ I,L,E-E	—	—	02.09.49.1	—	—
CVR	0114	Convert by Replacement from AC	⑧ I,L,E-E	—	—	02.09.49.1	—	—
DCT	0760..0012	Divide Check Test	2 I,L	X	—	02.09.58.1	—	—
DFAD	0301	Double-Precision FP Add	* 13 I,E,L-L	X	X	02.13.65.1	X	—
DFAM	0305	Double-Precision FP Add Magnitude	* 13 I,E,L-L	X	X	02.13.65.1	X	—
DFDH	—0240	Double-Precision FP Divide or Halt	* 17 I,E,L-L	X	X	02.13.87.1	X	—
DFDP	—0241	Double-Precision FP Divide or Proceed	* 17 I,E,L-L	X	X	02.13.87.1	X	—
DFMP	0261	Double-Precision FP Multiply	* 9 I,E,L-L	X	X	02.13.81.1	X	—
DFSB	0303	Double-Precision FP Subtract	* 13 I,E,L-L	X	X	02.13.65.1	X	—
DFSM	0307	Double-Precision FP Subtract Magnitude	* 13 I,E,L-L	X	X	02.13.65.1	X	—
DLD	0443	Double Load	2 I,E	X	X	02.09.39.1	X	—
DST	—0603	Double Store	3 I,E,E	X	X	02.09.02.1	X	X(1st E)
DUAM	—0305	Double-Precision Unnormalized FP Add Magnitude	* 10 I,E,L-L	X	X	02.13.65.1	X	—
DUFA	—0301	Double-Precision Unnormalized FP Add	* 10 I,E,L-L	X	X	02.13.65.1	X	—
DUFM	—0261	Double-Precision Unnormalized FP Multiply	* 9 I,E,L-L	X	X	02.13.81.1	X	—
DUFS	—0303	Double-Precision Unnormalized FP Subtract	* 10 I,E,L-L	X	X	02.13.65.1	X	—
DUSM	—0307	Double-Precision Unnormalized FP Subtract Magnitude	* 10 I,E,L-L	X	X	02.13.65.1	X	—
DVH	0220	Divide or Halt	7 I,E,5L	X	X	02.13.84.1	X	X
DVP	0221	Divide or Proceed	7 I,E,5L	X	X	02.13.84.1	X	X
ECTM	—0760..0016	Enter Copy Trap Mode	2 I,L	X	—	02.10.70.1	—	—
EDAT	0045	Enter Diagnostic Mode and Transfer	1 I	X	X	03.08.17.3	—	—
EFTM	—0760..0002	Enter Floating Trap Mode	2 I,L	X	—	02.10.71.1	—	—
ELAP	0047	Enter Overlap Mode and Transfer	1 I	X	X	03.08.17.3	—	—
EMTM	—0760..0016	Enter Multiple Tag Mode	2 I,L	X	—	03.05.32.1	—	—
ENB	0564	Enable from X	2 I,Ec	X	X	02.10.61.1	—	—
ENK	0760..0004	Enter Keys	2 I,L	X	—	04.20.14.1	—	—
ERA	0322	Exclusive OR to Accumulator	3 I,E,L	X	X	02.09.40.1	—	—
ESNT	—0021	Enter Storage Null and Transfer	1 I	X	X	02.10.70.1	X	—
ESTM	—0760..0005	Enter Select Trap Mode	2 I,L	X	—	02.10.70.1	—	—
ETM	0760..0007	Enter Trapping Mode	2 I,L	X	—	02.10.53.1	—	—
ETT	—0760..c000	End of Tape Test	2 I,L	X	—	02.10.80.1	—	—
FAD	0300	Floating Add	* 10 I,E,L-L	X	X	02.13.47.1	X	X

OPERATION CODE						SYSTEMS	OVERLAP	
ALPHA	OCTAL	INSTRUCTION	CYCLES	INDEX	IA	PAGE	I LAP	E/L LAP
FAM	0304	Floating Add Magnitude	* 10 I,E,L-L	X	X	02.09.93.1	X	X
FDH	0240	Floating Divide or Halt	* 6 I,E,4L	X	X	02.13.85.1	X	X
FDP	0241	Floating Divide or Proceed	* 6 I,E,4L	X	X	02.13.85.1	X	X
FMP	0260	Floating Multiply	* 4 I,E,L,L	X	X	02.13.79.1	X	X
FRN	0760..0011	Floating Round	2 I,L	X	—	02.10.27.1	—	—
FSB	0302	Floating Subtract	* 10 I,E,L-L	X	X	02.09.93.1	X	X
FSM	0306	Floating Subtract Magnitude	* 10 I,E,L-L	X	X	02.09.93.1	X	X
HPR	0420	Halt and Proceed	2 I,L	—	—	03.01.02.1	—	—
HTR	0000	Halt and Transfer	2 I,L	X	X	03.01.02.1	—	—
IIA	0041	Invert Indicators from AC	1 I	—	—	02.12.67.1	—	—
IIL	—0051	Invert Indicators of Left Half	1 I	—	—	02.12.62.1	—	—
IIR	0051	Invert Indicators of Right Half	1 I	—	—	02.12.62.1	—	—
IIS	0440	Invert Indicators from Storage	2 I,E	X	X	02.12.68.1	X	—
IOT	0760..0005	Input/Output Check Test	2 I,L	X	—	02.10.53.1	—	—
LAC	0535	Load Complement of Address in Index	2 I,E	—	—	02.12.74.1	X	—
LAS	—0340	Logical Compare Accumulator with Storage	2 I,E	X	X	02.09.41.1	X	X
LBT	0760..0001	Low-Order Bit Test	2 I,L	X	—	02.09.58.1	—	—
LCHA	0544	Load Channel A (7607)	3 I,Lc,Ec	X	X	08.00.22.1	—	—
LCHB	—0544	Load Channel B (7607)	3 I,Lc,Ec	X	X	08.00.22.1	—	—
LCHC	0545	Load Channel C (7607)	3 I,Lc,Ec	X	X	08.00.22.1	—	—
LCHD	—0545	Load Channel D (7607)	3 I,Lc,Ec	X	X	08.00.22.1	—	—
LCHE	0546	Load Channel E (7607)	3 I,Lc,Ec	X	X	08.00.22.1	—	—
LCHF	—0546	Load Channel F (7607)	3 I,Lc,Ec	X	X	08.00.22.1	—	—
LCHG	0547	Load Channel G (7607)	3 I,Lc,Ec	X	X	08.00.22.1	—	—
LCHH	—0547	Load Channel H (7607)	3 I,Lc,Ec	X	X	08.00.22.1	—	—
LDAT	—0045	Leave Diagnostic Mode and Transfer	1 I	X	X	03.08.17.3	—	—
LDC	—0535	Load Complement of Decrement in Index	2 I,E	—	—	02.12.74.1	X	—
LDI	0441	Load Indicators	2 I,E	X	X	02.12.68.1	X	—
LDQ	0560	Load MQ	2 I,E	X	X	02.12.40.1	X	X
LFT	—0054	Left Half Indicators Off Test	3 I,L,L	—	—	02.12.67.1	—	—
LFTM	—0760..0004	Leave Floating Trap Mode	2 I,L	X	—	02.10.71.1	—	—
LGL	—0763	Logical Left Shift	② I,L-L	X	—	02.09.70.1	X	X
LGR	—0765	Logical Right Shift	② I,L-L	X	—	02.09.70.1	X	X
LLAP	—0047	Leave Overlap Mode and Transfer	1 I	X	X	03.08.17.3	—	—
LLS	0763	Long Left Shift	② I,L-L	X	—	02.09.70.1	X	X
LMTM	0760..0016	Leave Multiple Tag Mode	2 I,L	X	—	03.05.32.1	—	—
LNT	—0056	Left Half Indicators On Test	3 I,L,L	—	—	02.12.67.1	—	—
LRS	0765	Long Right Shift	② I,L-L	X	—	02.09.70.1	X	X
LSNM	—0760..0010	Leave Storage Nullification Mode	2 I,L	X	—	02.10.70.1	—	—
LTM	—0760..0007	Leave Trapping Mode	2 I,L			02.10.53.1	—	—
LXA	0534	Load Index from Address	2 I,E	—	—	02.12.74.1	X	—
LXD	—0534	Load Index from Decrement	2 I,E	—	—	02.12.74.1	X	—
MPR	—0200	Multiply and Round	* 5 I,E,L-L	X	X	02.13.73.1	X	X
MPY	0200	Multiply	* 4 I,E,L-L	X	X	02.13.73.1	X	X
MSE	—0760	Minus Sense (See SLT)	2 I,L	X	—	02.09.58.1	—	—
NOP	0761	No Operation	2 I,L	X	—	08.00.01.1	—	—
NZT	—0520	Storage Not-Zero Test	2 I,E	X	X	02.11.50.1	X	—
OAI	0043	Or Accumulator to Indicators	1 I	—	—	02.12.67.1	—	—
OFT	0444	Off Test for Indicators	4 I,E,L,L	X	X	02.12.68.1	—	—
ONT	0446	On Test for Indicators	4 I,E,L,L	X	X	02.12.68.1	—	—
ORA	—0501	OR to Accumulator	2 I,E	X	X	02.10.02.1	X	X
ORS	—0602	OR to Storage	2 I,E	X	X	02.09.02.1	X	X
OSI	0442	OR Storage to Indicators	2 I,E	X	X	02.12.68.1	X	—
PAC	0737	Place Complement of Address in Index	1 I	—	—	02.12.74.1	X	—
PAI	0044	Place Accumulator in Indicators	1 I	—	—	02.12.67.1	—	—
PAX	0734	Place Address in Index	1 I	—	—	02.12.74.1	X	—
PBT	—0760..0001	P Bit Test	2 I,L	X	—	02.09.58.1	—	—
PCA	0756	Place Complement of Index in Address	1 I	—	—	02.12.74.1	X	—
PCD	—0756	Place Complement of Index in Decrement	1 I	—	—	02.12.74.1	X	—
PDC	—0737	Place Complement of Decrement in Index	1 I	—	—	02.12.74.1	X	—
PDX	—0734	Place Decrement in Index	1 I	—	—	02.12.74.1	X	—
PIA	—0046	Place Indicator in Accumulator	1 I	—	—	02.12.67.1	—	—
PSE	0760	Plus Sense (See SLF, SLN, SWT, SPU, SPT, SPR)						

OPERATION CODE						SYSTEMS	OVERLAP	
ALPHA	OCTAL	INSTRUCTION	CYCLES	INDEX	IA	PAGE	I LAP	E/L LAP
PXA	0754	Place Index in Address	1 I	—	—	02.12.74.1	X	—
PXD	—0754	Place Index in Decrement	1 I	—	—	02.12.74.1	X	—
RCD	0762..cc321	Read Card Reader	2 I,Lc	X	—	03.02.00.1	—	—
RCHA	0540	Reset and Load Channel A (7607)	3 I,Lc,Ec	X	X	08.00.22.1	—	—
RCHB	—0540	Reset and Load Channel B (7607)	3 I,Lc,Ec	X	X	08.00.22.1	—	—
RCHC	0541	Reset and Load Channel C (7607)	3 I,Lc,Ec	X	X	08.00.22.1	—	—
RCHD	—0541	Reset and Load Channel D (7607)	3 I,Lc,Ec	X	X	08.00.22.1	—	—
RCHE	0542	Reset and Load Channel E (7607)	3 I,Lc,Ec	X	X	08.00.22.1	—	—
RCHF	—0542	Reset and Load Channel F (7607)	3 I,Lc,Ec	X	X	08.00.22.1	—	—
RCHG	0543	Reset and Load Channel G (7607)	3 I,Lc,Ec	X	X	08.00.22.1	—	—
RCHH	—0543	Reset and Load Channel H (7607)	3 I,Lc,Ec	X	X	08.00.22.1	—	—
RCT	0760..0014	Restore Channel Traps	2 I,L	X	—	02.10.61.1	—	—
RDCA	0760..01352	Reset Data Channel A (7607)	2 I,Lc	X	—	60.65.04.1	—	—
RDCB	0760..02352	Reset Data Channel B (7607)	2 I,Lc	X	—	60.65.04.1	—	—
RDCC	0760..03352	Reset Data Channel C (7607)	2 I,Lc	X	—	60.65.04.1	—	—
RDCD	0760..04352	Reset Data Channel D (7607)	2 I,Lc	X	—	60.65.04.1	—	—
RDCE	0760..05352	Reset Data Channel E (7607)	2 I,Lc	X	—	60.65.04.1	—	—
RDCF	0760..06352	Reset Data Channel F (7607)	2 I,Lc	X	—	60.65.04.1	—	—
RDCG	0760..07352	Reset Data Channel G (7607)	2 I,Lc	X	—	60.65.04.1	—	—
RDCH	0760..10352	Reset Data Channel H (7607)	2 I,Lc	X	—	60.65.04.1	—	—
RDS	0762	Read Select	2 I,Lc	X	—	06.01.12.1	—	—
REW	0772	Rewind	2 I,Lc	X	—	06.01.12.2	—	—
RFT	0054	Right Half Indicators Off Test	3 I,L,L	—	—	02.12.67.1	—	—
RIA	—0042	Reset Indicators from Accumulator	1 I	X	—	02.12.67.1	—	—
RICA	0760..01350	Reset Channel A (7909)	2 I,Lc	X	—	50.63.01.1	—	—
RICB	0760..02350	Reset Channel B (7909)	2 I,Lc	X	—	50.63.01.1	—	—
RICC	0760..03350	Reset Channel C (7909)	2 I,Lc	X	—	50.63.01.1	—	—
RICD	0760..04350	Reset Channel D (7909)	2 I,Lc	X	—	50.63.01.1	—	—
RICE	0760..05350	Reset Channel E (7909)	2 I,Lc	X	—	50.63.01.1	—	—
RICF	0760..06350	Reset Channel F (7909)	2 I,Lc	X	—	50.63.01.1	—	—
RICG	0760..07350	Reset Channel G (7909)	2 I,Lc	X	—	50.63.01.1	—	—
RICH	0760..10350	Reset Channel H (7909)	2 I,Lc	X	—	50.63.01.1	—	—
RIL	—0057	Reset Indicators of Left Half	1 I	—	—	02.12.62.1	—	—
RIR	0057	Reset Indicators of Right Half	1 I	—	—	02.12.62.1	—	—
RIS	0445	Reset Indicators from Storage	2 I,E			02.12.62.1	X	—
RND	0760..0010	Round	2 I,L	X	—	02.09.57.1	—	—
RNT	0056	Right Half Indicators On Test	3 I,L,L	—	—	02.12.67.1	—	—
RPR	0762..cc361	Read Printer	2 I,Lc	X	—	03.02.01.1	—	—
RQL	—0773	Rotate MQ Left	② I,L-L	X	—	02.09.70.1	X	X
RSCA	0540	Reset and Start Channel A (7909)	3 I,Lc,Ec	X	X	08.00.22.1	—	—
RSCB	—0540	Reset and Start Channel B (7909)	3 I,Lc,Ec	X	X	08.00.22.1	—	—
RSCC	0541	Reset and Start Channel C (7909)	3 I,Lc,Ec	X	X	08.00.22.1	—	—
RSCD	—0541	Reset and Start Channel D (7909)	3 I,Lc,Ec	X	X	08.00.22.1	—	—
RSCE	0542	Reset and Start Channel E (7909)	3 I,Lc,Ec	X	X	08.00.22.1	—	—
RSCF	—0542	Reset and Start Channel F (7909)	3 I,Lc,Ec	X	X	08.00.22.1	—	—
RSCG	0543	Reset and Start Channel G (7909)	3 I,Lc,Ec	X	X	08.00.22.1	—	—
RSCH	—0543	Reset and Start Channel H (7909)	3 I,Lc,Ec	X	X	08.00.22.1	—	—
RTB	0762..cc2uu	Read Tape Binary	2 I,Lc	X	—	03.02.00.1	—	—
RTD	0762..cc2uu	Read Tape Decimal	2 I,Lc	X	—	03.02.00.1	—	—
RUN	—0772	Rewind and Unload	2 I,Lc	X	—	06.01.12.2	—	—
SBM	—0400	Subtract Magnitude	2 I,E	X	X	02.09.93.1	X	X
SCA	0636	Store Complement of Index in Address	2 I,E	—	—	02.09.03.1	X	X
SCD	—0636	Store Complement of Index in Decrement	2 I,E	—	—	02.09.03.1	X	X
SCDA	0644	Store Channel Diagnostic A (7909)	2 I,Ec	X	X	50.21.01.1	—	—
SCDB	—0644	Store Channel Diagnostic B (7909)	2 I,Ec	X	X	50.21.01.1	—	—
SCDC	0645	Store Channel Diagnostic C (7909)	2 I,Ec	X	X	50.21.01.1	—	—
SCDD	—0645	Store Channel Diagnostic D (7909)	2 I,Ec	X	X	50.21.01.1	—	—
SCDE	0646	Store Channel Diagnostic E (7909)	2 I,Ec	X	X	50.21.01.1	—	—
SCDF	—0646	Store Channel Diagnostic F (7909)	2 I,Ec	X	X	50.21.01.1	—	—
SCDG	0647	Store Channel Diagnostic G (7909)	2 I,Ec	X	X	50.21.01.1	—	—
SCDH	—0647	Store Channel Diagnostic H (7909)	2 I,Ec	X	X	50.21.01.1	—	—
SCHA	0640	Store Channel A	2 I,Ec	X	X	03.01.12.1	—	—
SCHB	—0640	Store Channel B	2 I,Ec	X	X	03.01.12.1	—	—
SCHC	0641	Store Channel C	2 I,Ec	X	X	03.01.12.1	—	—
SCHD	—0641	Store Channel D	2 I,Ec	X	X	03.01.12.1	—	—

OPERATION CODE						SYSTEMS	OVERLAP	
ALPHA	OCTAL	INSTRUCTION	CYCLES	INDEX	IA	PAGE	I LAP	E/L LAP
SCHE	0642	Store Channel E	2 I,Ec	X	X	03.01.12.1	—	—
SCHF	—0642	Store Channel F	2 I,Ec	X	X	03.01.12.1	—	—
SCHG	0643	Store Channel G	2 I,Ec	X	X	03.01.12.1	—	—
SCHH	—0643	Store Channel H	2 I,Ec	X	X	03.01.12.1	—	—
SDN	0776	Set Density	2 I,Lc	—	—	60.50.02.3	—	—
SIL	—0055	Set Indicators of Left Half	1 I	—	—	02.12.67.1	—	—
SIR	0055	Set Indicators of Right Half	1 I	—	—	02.12.67.1	—	—
SLF	0760..0140	Sense Lights Off	2 I,L	X	—	02.09.60.1	—	—
SLN	0760..014n	Sense Lights On	2 I,L	X	—	02.09.60.1	—	—
SLQ	—0620	Store Left Half MQ	2 I,E	X	X	02.09.03.1	X	X
SLT	0760..014n	Sense Light Test	2 I,L	X	—	02.09.58.1	—	—
SLW	0602	Store Logical Word	2 I,E	X	X	02.09.02.1	X	X
SPR	0760..cc3nn	Sense Printer	2 I,Lc	X	—	03.02.01.1	—	—
SPT	0760..cc360	Sense Printer Test	2 I,Lc	X	—	03.02.01.1	—	—
SPU	0760..cc34n	Sense Punch	2 I,Lc	X	—	03.02.00.1	—	—
SSM	—0760..0003	Set Sign Minus	2 I,L	X	—	02.09.57.1	—	—
SSP	0760..0003	Set Sign Plus	2 I,L	X	—	02.09.57.1	—	—
STA	0621	Store Address	2 I,E	X	X	02.09.03.1	X	X
STCA	0544	Start Channel A (7909)	2 I,Ec	—	—	50.21.01.1	—	—
STCB	—0544	Start Channel B (7909)	2 I,Ec	—	—	50.21.01.1	—	—
STCC	0545	Start Channel C (7909)	2 I,Ec	—	—	50.21.01.1	—	—
STCD	—0545	Start Channel D (7909)	2 I,Ec	—	—	50.21.01.1	—	—
STCE	0546	Start Channel E (7909)	2 I,Ec	—	—	50.21.01.1	—	—
STCF	—0546	Start Channel F (7909)	2 I,Ec	—	—	50.21.01.1	—	—
STCG	0547	Start Channel G (7909)	2 I,Ec	—	—	50.21.01.1	—	—
STCH	—0547	Start Channel H (7909)	2 I,Ec	—	—	50.21.01.1	—	—
STD	0622	Store Decrement	2 I,E	X	X	02.09.03.1	X	X
STI	0604	Store Indicators	2 I,E	X	X	02.09.02.1	X	X
STL	—0625	Store Instruction Location Counter	2 I,E	X	X	02.09.03.1	X	X
STO	0601	Store	2 I,E	X	X	02.09.02.1	X	X
STP	0630	Store Prefix	2 I,E	X	X	02.09.03.1	X	X
STQ	—0600	Store MQ	2 I,E	X	X	02.09.02.1	X	X
STR	—1000	Store Location and Trap	2 I,E	—	—	02.11.52.1	—	—
STT	0625	Store Tag	2 I,E	X	X	02.09.03.1	X	X
STZ	0600	Store Zero	2 I,E	X	X	02.09.02.1	X	X
SUB	0402	Subtract	2 I,E	X	X	02.09.93.1	X	X
SWT	0760..016n	Sense Switch Test	2 I,L	X	—	02.09.59.1	—	—
SXA	0634	Store Index in Address	2 I,E	—	—	02.09.03.1	X	X
SXD	—0634	Store Index in Decrement	2 I,E	—	—	02.09.03.1	X	X
TCNA	—0061	Transfer on DSCA Not in Operation	2 I,L	X	X	06.01.00.1	—	—
TCNB	—0061	Transfer on DSCB Not in Operation	2 I,L	X	X	06.01.00.1	—	—
TCNC	—0062	Transfer on DSCC Not in Operation	2 I,L	X	X	06.01.01.1	—	—
TCND	—0063	Transfer on DSCD Not in Operation	2 I,L	X	X	06.01.01.1	—	—
TCNE	—0064	Transfer on DSCE Not in Operation	2 I,L	X	X	06.01.02.1	—	—
TCNF	—0065	Transfer on DSCF Not in Operation	2 I,L	X	X	06.01.02.1	—	—
TCNG	—0066	Transfer on DSCG Not in Operation	2 I,L	X	X	06.01.03.1	—	—
TCNH	—0067	Transfer on DSCH Not in Operation	2 I,L	X	X	06.01.03.1	—	—
TCOA	0060	Transfer on DSCA in Operation	2 I,L	X	X	06.01.00.1	—	—
TCOB	0061	Transfer on DSCB in Operation	2 I,L	X	X	06.01.00.1	—	—
TCOC	0062	Transfer on DSCC in Operation	2 I,L	X	X	06.01.01.1	—	—
TCOD	0063	Transfer on DSCD in Operation	2 I,L	X	X	06.01.01.1	—	—
TCOE	0064	Transfer on DSCE in Operation	2 I,L	X	X	06.01.02.1	—	—
TCOF	0065	Transfer on DSCF in Operation	2 I,L	X	X	06.01.02.1	—	—
TCOG	0066	Transfer on DSCG in Operation	2 I,L	X	X	06.01.03.1	—	—
TCOH	0067	Transfer on DSCH in Operation	2 I,L	X	X	06.01.03.1	—	—
TEFA	0030	Transfer on DSCA End of File	2 I,Lc	X	X	06.01.00.1	—	—
TEFB	—0030	Transfer on DSCB End of File	2 I,Lc	X	X	06.01.00.1	—	—
TEFC	0031	Transfer on DSCC End of File	2 I,Lc	X	X	06.01.01.1	—	—
TEFD	—0031	Transfer on DSCD End of File	2 I,Lc	X	X	06.01.01.1	—	—
TEFE	0032	Transfer on DSCE End of File	2 I,Lc	X	X	06.01.02.1	—	—
TEFF	—0032	Transfer on DSCF End of File	2 I,Lc	X	X	06.01.02.1	—	—
TEFG	0033	Transfer on DSCG End of File	2 I,Lc	X	X	06.01.03.1	—	—
TEFH	—0033	Transfer on DSCH End of File	2 I,Lc	X	X	06.01.03.1	—	—
TIF	0046	Transfer if Indicators Off	2 I,L	X	X	02.10.57.1	—	—
TIO	0042	Transfer if Indicators On	2 I,L	X	X	02.10.57.1	—	—

OPERATION CODE						SYSTEMS	OVERLAP	
ALPHA	OCTAL	INSTRUCTION	CYCLES	INDEX	IA	PAGE	I LAP	E/L LAP
TIX	2000	Transfer on Index	1 I	—	—	03.06.07.1	X	—
TLQ	0040	Transfer on Low MQ	2 I,L	X	X	02.09.30.1	—	—
TMI	—0120	Transfer on Minus	1 I	X	X	02.12.77.1	—	—
TNO	—0140	Transfer on No Overflow	1 I	X	X	02.12.77.1	—	—
TNX	—2000	Transfer on No Index	1 I	—	—	03.06.13.1	X	—
TNZ	—0100	Transfer on No Zero	1 I	X	X	02.12.77.1	—	—
TOV	0140	Transfer on Overflow	1 I	X	X	02.12.77.1	—	—
TPL	0120	Transfer on Plus	1 I	X	X	02.12.77.1	—	—
TQO	0161	Transfer on MQ Overflow	1 I	X	X	02.12.77.1	—	—
TQP	0162	Transfer on MQ Plus	1 I	X	X	02.12.77.1	—	—
TRA	0020	Transfer	1 I	X	X	02.12.77.1	X	—
TRCA	0022	Transfer on DSCA Redundancy Check	2 I,Lc	X	X	06.01.00.1	—	—
TRCB	—0022	Transfer on DSCB Redundancy Check	2 I,Lc	X	X	06.01.00.1	—	—
TRCC	0024	Transfer on DSCC Redundancy Check	2 I,Lc	X	X	06.01.01.1	—	—
TRCD	—0024	Transfer on DSCD Redundancy Check	2 I,Lc	X	X	06.01.01.1	—	—
TRCE	0026	Transfer on DSCE Redundancy Check	2 I,Lc	X	X	06.01.02.1	—	—
TRCF	—0026	Transfer on DSCF Redundancy Check	2 I,Lc	X	X	06.01.02.1	—	—
TRCG	0027	Transfer on DSCG Redundancy Check	2 I,Lc	X	X	06.01.03.1	—	—
TRCH	—0027	Transfer on DSCH Redundancy Check	2 I,Lc	X	X	06.01.03.1	—	—
TSX	0074	Transfer and Set Index	1 I	—	—	03.06.07.1	X	—
TTR	0021	Trap Transfer	1 I	X	X	02.12.77.1	X	—
TXH	3000	Transfer on Index High	1 I	—	—	02.12.76.1	X	—
TXI	1000	Transfer with Index Incremented	1 I	—	—	02.12.76.1	X	—
TXL	—3000	Transfer with Index Low or Equal	1 I	—	—	02.12.76.1	X	—
TZE	0100	Transfer on Zero	1 I	X	X	02.12.77.1	—	—
UAM	—0304	Unnormalized Add Magnitude	* 7 I,E,L-L	X	X	02.13.47.1	X	X
UFA	—0300	Unnormalized Floating Add	* 7 I,E,L-L	X	X	02.13.47.1	X	X
UFM	—0260	Unnormalized Floating Multiply	4 I,E,L,L	X	X	02.13.79.1	X	X
UFS	—0302	Unnormalized Floating Subtract	* 7 I,E,L-L	X	X	02.09.93.1	X	X
USM	—0306	Unnormalized Subtract Magnitude	* 7 I,E,L-L	X	X	02.09.93.1	X	X
VDH	0224	Variable Length Divide or Halt	* 7 I,E,L-L	X	—	02.13.84.1	X	X
VDP	0225	Variable Length Divide or Proceed	* 7 I,E,L-L	X	—	02.13.84.1	X	X
VLM	0204	Variable Length Multiply	* 5 I,E,L-L	X	—	02.13.73.1	X	X
WEF	0770	Write End of File	2 I,Lc	X	—	06.01.12.2	—	—
WPB	0766..cc362	Write Printer Binary	2 I,Lc	X	—	03.02.01.1	—	—
WPD	0766..cc361	Write Printer Decimal	2 I,Lc	X	—	03.02.01.1	—	—
WRS	0766	Write Select	2 I,Lc	X	—	03.07.01.1	—	—
WTB	0766..cc2uu	Write Tape Binary	2 I,Lc	X	—	03.02.00.1	—	—
WTD	0766..cc2uu	Write Tape Decimal	2 I,Lc	X	—	03.02.00.1	—	—
XCA	0131	Exchange AC and MQ	1 I	—	—	02.10.06.1	X	—
XCL	—0130	Exchange Logical AC and MQ	1 I	—	—	02.10.06.1	X	—
XEC	0522	Execute	1 I	X	X	03.06.05.1	—	—
ZET	0520	Storage Zero Test	2 I,E	X	X	02.11.50.1	X	—

c = 1 to 10, A to H or Channel
n = a number from 1 to 6
u = unit address
o = average time
* = maximum time

CUT ALONG LINE

FOLD

FOLD

COMMENT SHEET

IBM 7094 II DATA PROCESSING SYSTEM—VOLUME 3

CUSTOMER ENGINEERING INSTRUCTION—MAINTENANCE. FORM 223-2723-0

FROM

NAME ______________________________

OFFICE NO. ______________________________

CHECK ONE OF THE COMMENTS AND EXPLAIN IN THE SPACE PROVIDED

□ SUGGESTED ADDITION (PAGE   , TIMING CHART, DRAWING, PROCEDURE, ETC.)

□ SUGGESTED DELETION (PAGE   )

□ ERROR (PAGE   )

EXPLANATION

FOLD

FOLD

NO POSTAGE NECESSARY IF MAILED IN U.S.A.

FOLD ON TWO LINES, STAPLE, AND MAIL

STAPLE STAPLE

FOLD FOLD

FIRST CLASS
PERMIT NO. 81

POUGHKEEPSIE, N. Y.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN U. S. A.

POSTAGE WILL BE PAID BY

IBM CORPORATION
P.O. BOX 390
POUGHKEEPSIE, N.Y. 12602

ATTN: CE MANUALS, DEPARTMENT B96

CUT ALONG LINE

FOLD FOLD

STAPLE STAPLE

CE
System
Maintenance
Library

System

CUT HERE

223-2723-0

IBM®

International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, N.Y. 10601