IBM

Customer Engineering Manuals Supplement

System/Unit 7094

Re: Form No. R23-2550-1

This Supplement No. S23-4002

Date October 25, 1962

Previous Supplement Nos. S23-4000 and S23-4001

IBM 7094 DATA PROCESSING SYSTEM
CUSTOMER ENGINEERING INSTRUCTION - REFERENCE

This publication is a supplement to the 7094 Customer Engineering Instruction-Reference Manual, Form R23-2550-1. The enclosed section on Trapping replaces corresponding material in the present manual.

International Business Machines Corp., Product Publications Dept., Poughkeepsie, N. Y.

PRINTED IN U.S.A.

TRAPPING

Table of Contents

TRAPPING

Special computer conditions can be indicated in various ways: console lights, triggers, tone signals, program stops, printouts, etc. Some conditions can be tested with instructions but this becomes time consuming. Others stop the computer and require operator intervention; this is not only time consuming but also subject to human error. The answer then lies in letting the computer monitor these special conditions and indicate these facts to the programmer so that he can 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 pre-set location. As the computer is trapped, the program counter contents and indentifications 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 "crime," take appropriate action, and continue again with the main program.

There are five major areas of traps:

1. Transfer traps
2. Arithmetic traps
3. I-O traps
4. Compatibility traps
5. Special traps

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

Figure

7094 Trap Locations

TRAP 2

LOCATION	CONDITION	ADDRESS	DECREMENT
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	Transfe r 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			
00035			
00036	Interval Timer Reset	PC +1	Not Used
00037	Transfer instruction for interval timer Reset trap		
00040	Storage Parity	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

Figure 7094 Trap Locations

TRAP 3

LOCATION	CONDITION	ADDRESS	DECREMENT
00045	Transfer command for 7909 channel B interrupt		
00046	Channel C 7909Int.	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 Cointer	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 7094 Trap Locations

TRAP 4

Store Location and Trap - STR -1000 I, E (Figure)

The objective of the STR instruction is to store the program counter (location of the STR +1) into the address portion of location 00000_8 and automatically trap to location 00002_8. This instruction is not effected by the trapping mode and does not place the computer in trapping mode.

With the bit in position 1 of the instruction, storage bus positions 1 & 2 are routed to program register positions 8 & 9. The instruction is defined non-indexable (02.12.76.1) and normal address modification circuitry from the address register through the index adders is blocked.

The program counter is advanced normally during the initial I cycle to indicate the address of the next sequential instruction. The address register is reset at I9 (D2) time so that a 00000_8 address is available to MAR for the following E cycle. Because this is an unconditional transfer (trap), the conditions met trigger is turned ON at I10 time.

Storing the program counter is accomplished during the following E cycle. "MF store address" causes the address portion of the core storage data word to be destroyed on read-out; the decrement, tag, and prefix portions are retained.

A data flow path for routing the program counter contents to core storage is through the index adders, to positions 21 - 35 of the storage register and from there onto the storage bus. During E0 (D3) time the complement of the program counter is routed to the index adders, and the complement of the index adders to the storage register; the net effect of the two complement routings is to place the true program counter value into the storage register. During E4 (D3) time the storage register is gated to the storage bus and from there into location 00000_8. Notice that when the storage register is set at E2 time, positions S,1-20 are cleared.

(STR 1)
TRAP 5

TRAP 5

At E6 , the program counter is reset; at 10 time, a 1-bit is forced into position 16 of the program counter and address register in time to be sent to MAR for the following I cycle core storage reference.

During the early portion of the next I cycle the address register is routed to the program counter and the operation is completed.

(STR 2)
TRAP 6

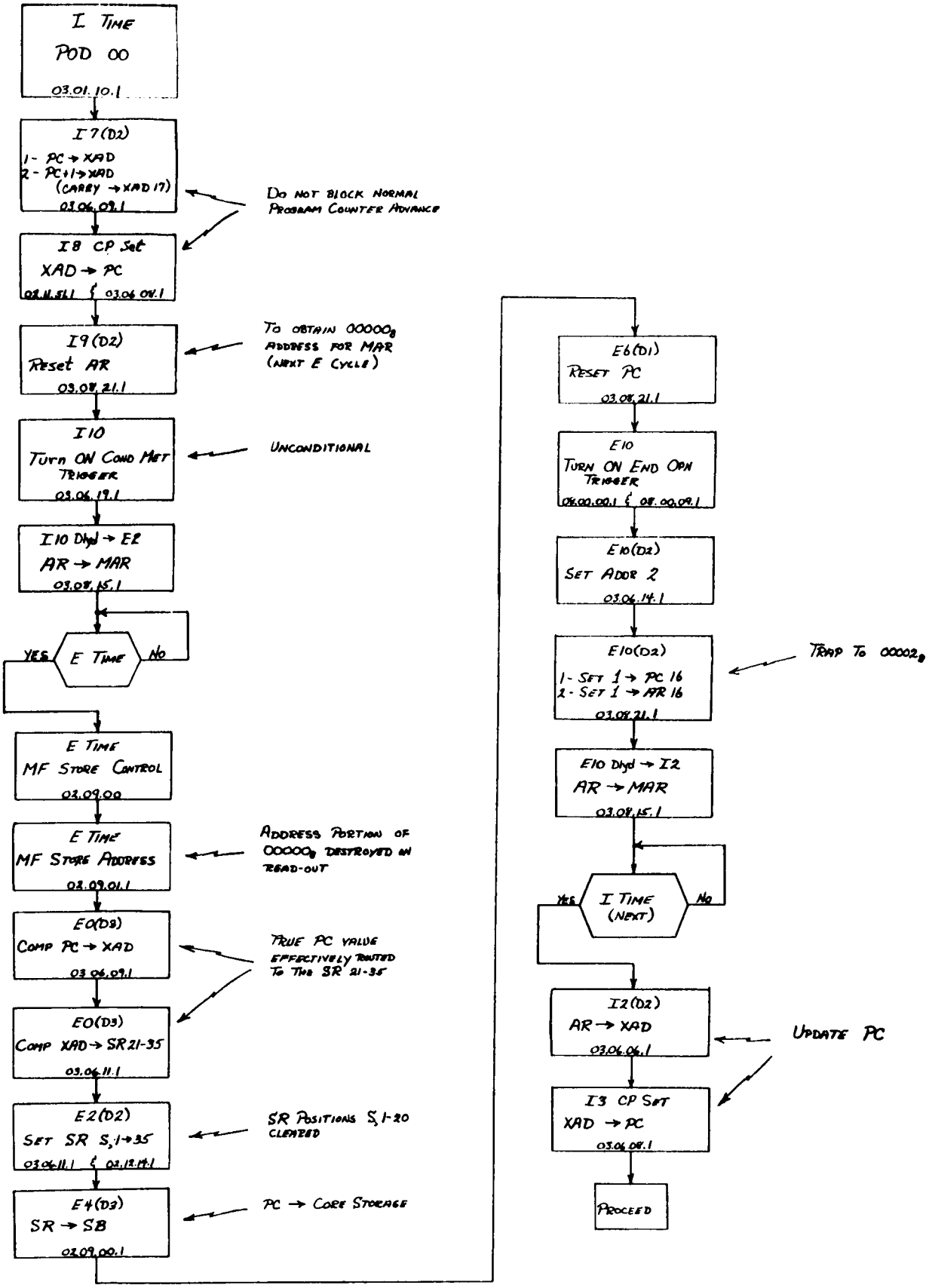

FIGURE NORMAL STR FLOW CHART

Bo 10/62

(STR 3)

TRAP 7

TRANSFER TRAP MODE

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

While 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 location 00001_8 there is normally an unconditional 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 not valid, the subroutine may indicate this fact by some form of machine stop or printout to the console operator.

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

The computer is placed in trap mode by execution of the Enter Trap Mode (ETM) instruction. Exit from the trap mode is accomplished by execution of the Leave Trap Mode (LTM) instruction or depression of either the clear or reset keys on the operator's console. Two instructions are immuned to the trap mode; Trap Transfer (TTR) and Enter Storage Nullification and Transfer (ESNT).

Transfer Trap Mode Instructions

Enter Trapping Mode - ETM +0760···00007 I, L (Figure)

This instruction places the computer in trapping mode by turning ON the trap mode trigger (02, 10.53.1). While in this mode of operation, successful transfer instructions do not actually transfer but, instead, are trapped to location 00001_8. Two instructions are immuned to the trapping mode: Trap Transfer (TTR) and Enter Storage Nullification and Transfer (ESNT).

During I6 (D2) and I7 (D1) times respectively, the storage register and tag register are set from the storage bus. Positions 21 - 35 of the storage register (the address portion of the instruction) are immediately routed to the address register. The data-flow path to the shift counter is from the address register and through the index adders. The address register is gated through the index adders during I9 (D2) time and, with POD 76 decoding from the program register set into the shift counter at I10 time. With no circuitry to send the computer to I or E time, the next cycle is an automatic L cycle.

During the following L cycle the trap mode trigger is turned ON, the instruction ends operation, and the computer proceeds to I time for the next instruction.

The computer remains in trapping mode until execution of the Leave Trap Mode (LTM) instruction, or the clear or reset buttons are depressed at the operator's console.

Leave Trapping Mode - LTM -0760···00007 I, L (Figure)

This instruction takes the computerout of trapping mode by resetting the trap mode trigger (02.10.53.1).

Execution of the LTM instruction is identical to that of the Enter Trap Mode (ETM) instruction with one exception; the trap mode trigger is reset at 10-time of the L cycle. If the computer is not in trap mode, the LTM has the effect of a NOP.

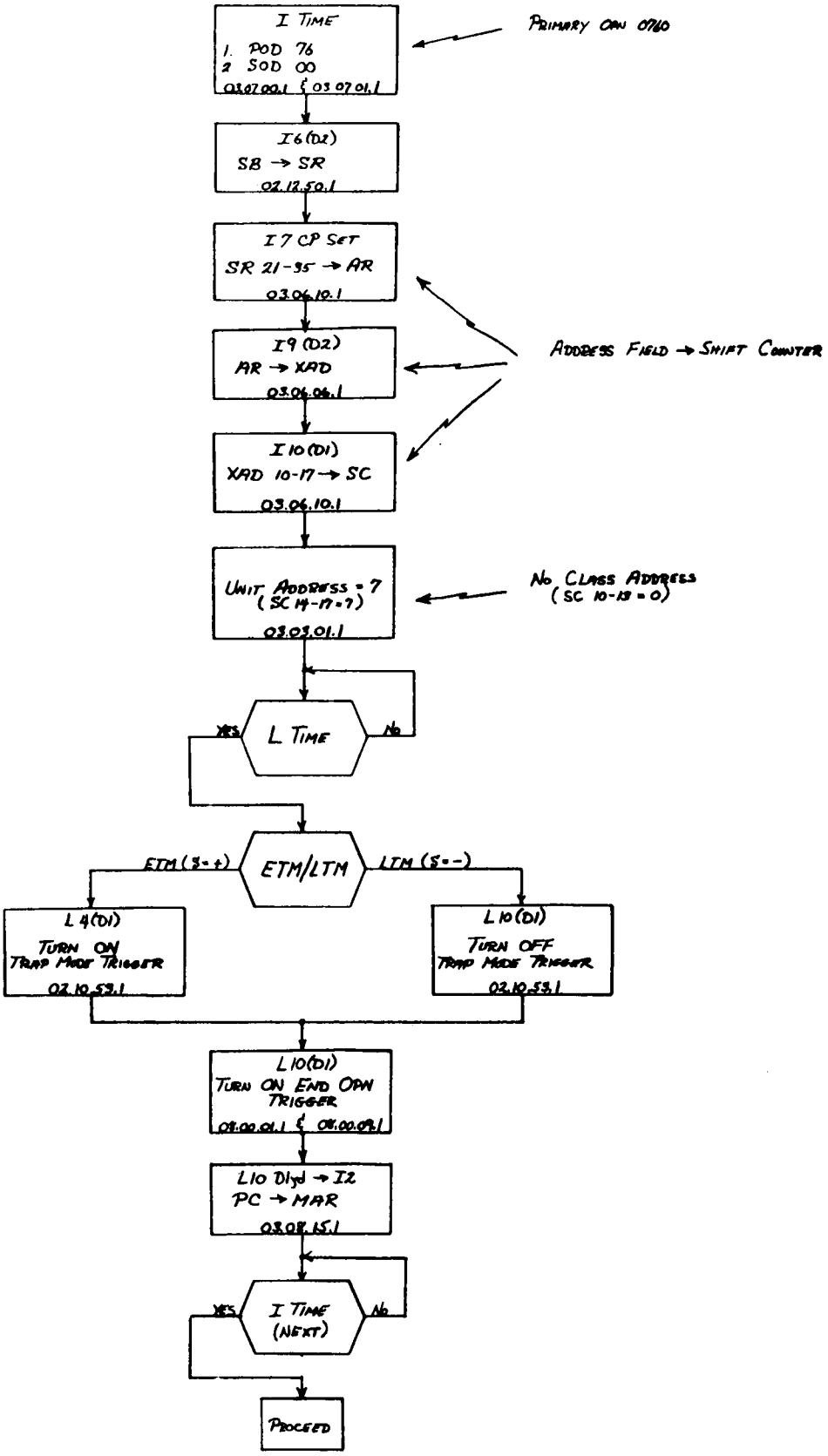

FIGURE ETM/LTM FLOW CHART

Trap Transfer - TTR +0021 I (Figure)

The trap transfer instruction is an unconditional transfer that is not affected by trap mode of the computer. Logically, it operates exactly like a normal Transfer (TRA) instruction.

When the computer is placed in trap mode by the ETM instruction, a trap mode trigger (02.10.53.1) is turned ON and all successful transfer instructions are trapped to location 00001_8. The trap mode circuitry which controls the trapping function is located on 02.11.52.1. Notice that TTR decoding from the program register deactivated this trap circuitry. In this simple manner the computer is "taken out" of trap mode for the duration of the TTR instruction and the transfer is successfully executed without any trapping.

This being a one cycle instruction, overlapping is not allowed by an instruction in the next higher odd location.

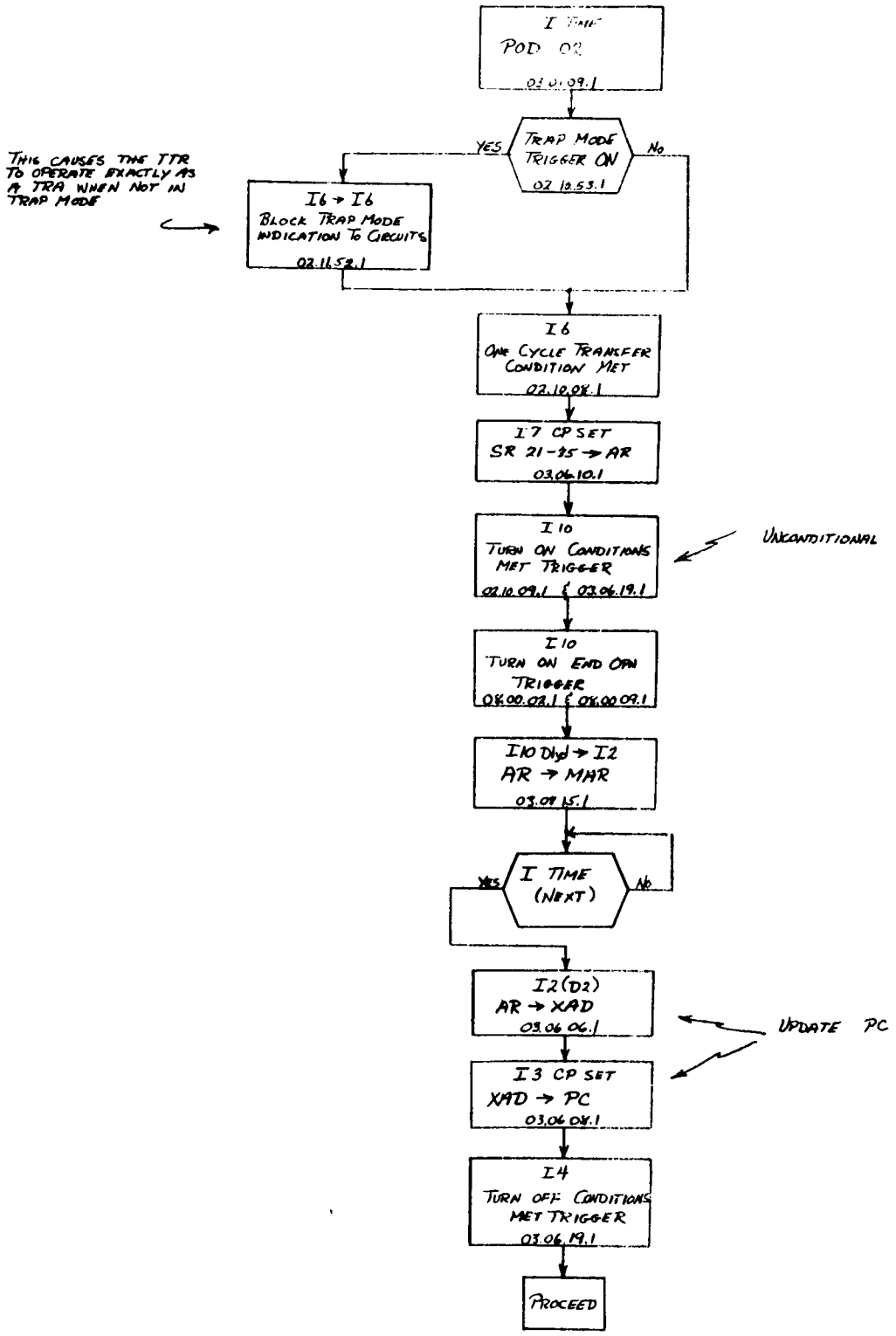

FIGURE TTR FLOW CHART

Bo 10/62

A Typical Trapping Operation - TOV/TNO (Figure)

The objectives of this operation are:

1. To store the present status of the program counter (location of the transfer instruction) into the address portion of core storage location 00000_8. This results in a cyclic makeup of I-E.
2. To trap the program to location 00001_8 if a successful transfer is indicated; or advance the program counter and proceed to the next sequential instruction if a successful transfer is not indicated.
3. To perform the functions called for by the particular instruction.

Accomplishing the first objective requires a block to the normal program counter advance at I8 time; the counter, therefore, continues to indicate the address of the transfer instruction being executed. Placing this value in core storage location 00000_8 requires an E cycle and the functions of a store operation. Because of this, normal "go to I" or "L time call" cycle controls are blocked (08.00.12.1 or 08.00.16.1). "Trap mode trigger OFF" not being active at this time blocks ending operation at the end of the I cycle (08.00.02.1) and prevents "go to I time" (08.00.12.1); "trap mode tgr ON" being active, blocks "L time call" (08.00.16.1) and, in turn, forces "go to E time" (08.00.12.1).

The transfer address contained in positions 21-25 of the transfer instruction performs no functions for this operation but is gated to the address register as a normal I time function. At I9 time, however, the address register is reset to supply the 00000_8 core storage reference location to MAR for the following E cycle. The normal address modification circuitry which gates the index register and address register to the index adders during I9 (D2) time is blocked by the trapping mode ["block I9 (D2) AR to XAD" - 03.06.09.1] .

Storing the program counter is accomplished during the following E cycle. "MF store address" causes the address portion of the core storage data word to be destroyed on read-out; the decrement, tag, and prefix portions are retained.

A data flow path for routing the program counter contents to core storage is through the index adders, to positions 21 -35 of the storage register and from there onto the storage bus. During E0 (D3) time the complement of the program counter is routed to the index adders, and the complement of the index adders to the storage register; the net effect of the two complement routings is to place the true program counter value into the storage register. Notice that the address portion of the storage register is destroyed and preplaced by the contents of the program counter. This destruction does no harm because the transfer-to address of the instruction performs no function when in trap mode.

The decrement portion of the storage register must be saved, however, because it may be that of an index instruction (TIX for example) and needed for modification of the index register later in the operation when the storing has been completed. Saving the decrement is accomplished by effectively looping SR 1-17 through the main adders. Setting occurs again at the same time that the program counter value is being set at E2 time. The storage register is routed to core storage during E4 (D3) time to complete the store phase of the operation.

The second objective is to determine whether or not a trap should be initiated. During the initial I time of the TOV/TNO operation, the status of the accumulator overflow trigger is tested (02.10.08.1) and if the conditions of the test are met, the conditions met trigger is turned ON. At E10 time, the computer must be able to supply MAR with either of two addresses: the location of the next sequential instruction from the program counter; or the trap location of 00001_8 from the address register.

The program counter was blocked during the initial I cycle and, therefore, does not indicate the next sequential instruction location. Stepping the counter is accomplished by incrementing through the index adders during E8 (D3) and back again with an E9 CP set pulse. If the transfer conditions are met, the program counter is reset and 1's forced into position 17 of both the address register and program counter; the address register is gated to MAR and the computer executes the next instruction from location 00001_8. During the early portion of the following I cycle, if the trap was successful, the address register is routed to the program counter and the program proceeds from this point.

An important part of the operation still lies in the last objective; that if performing all of the normal functions of the particular transfer instruction. The one remaining item to be stressed is that the accumulator overflow trigger is turned OFF. Many people have the false impression that if a trap occurs, the transfer instruction isnot executed. Only the actualtransfer is not executed; other functions, such as turning OFF the overflow trigger, incrementing or decrementing index registers, etc. are completed in the normal manner just as though the computer were not in trap mode.

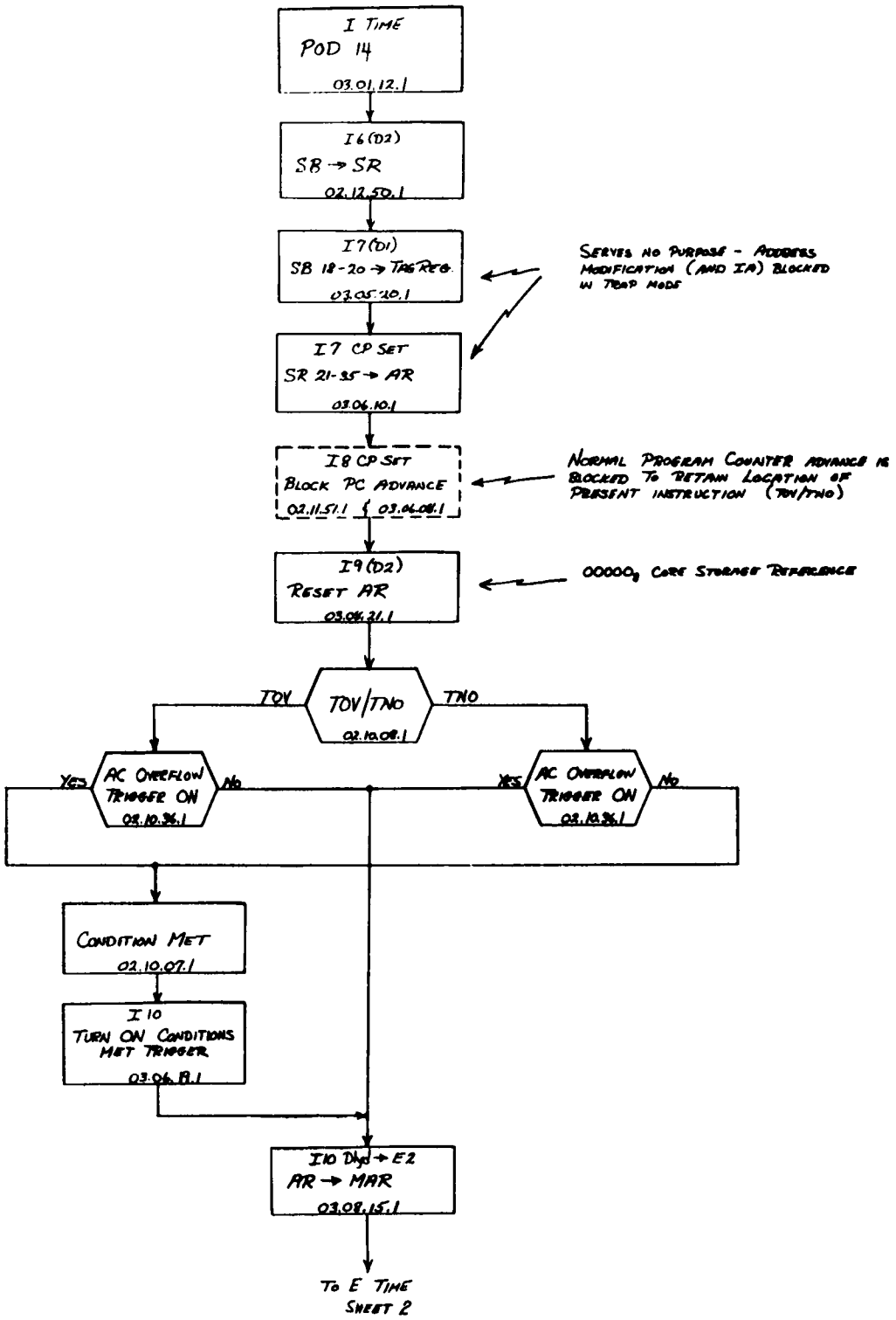

FIGURE TOV/TNO TRAP MODE FLOW CHART

SHEET 1 of 2

(TOVT4)

TRAP 16

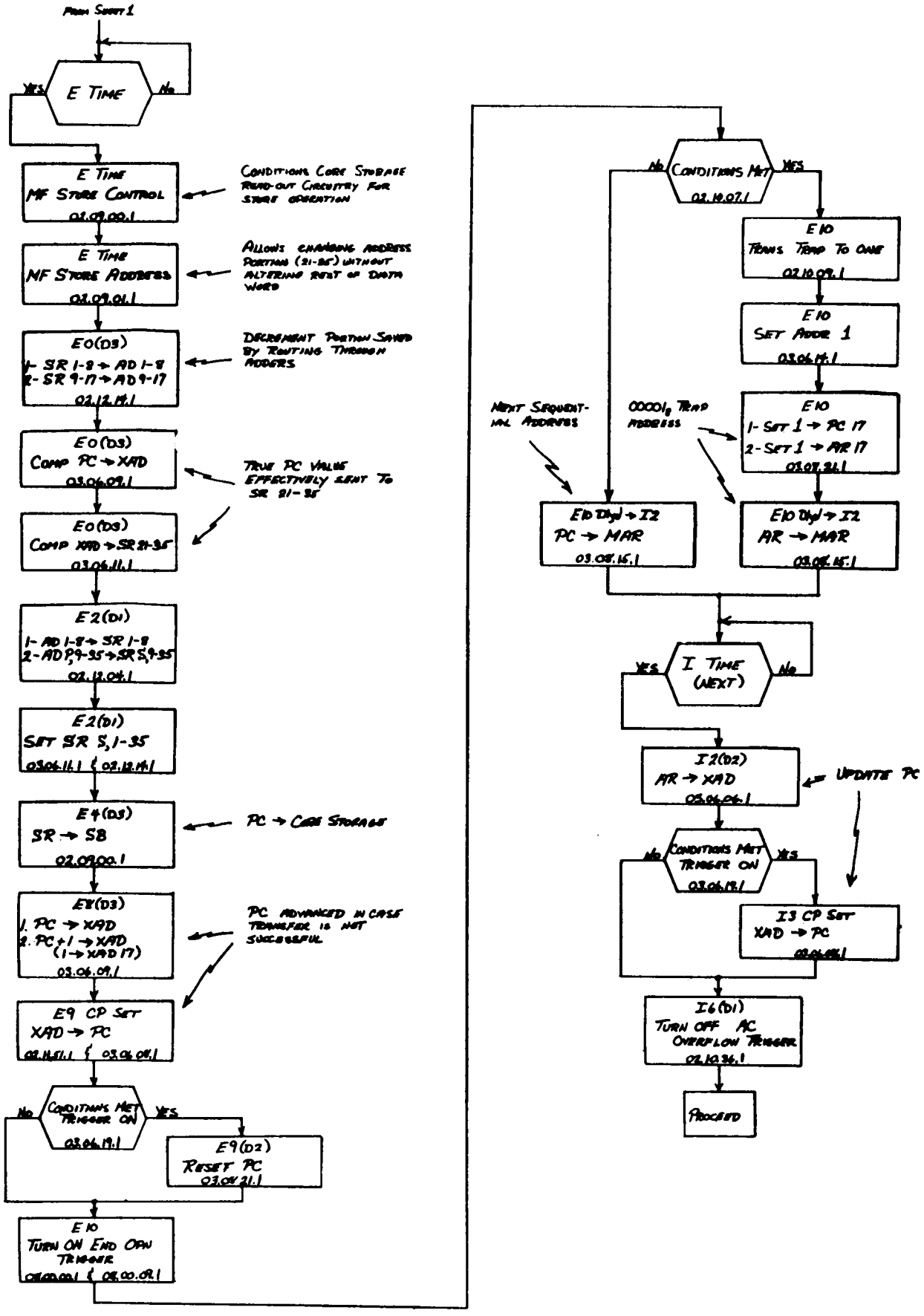

FIGURE TOV/TNO TRAP MODE FLOW CHART

(TOVT 5)

TRAP 17

SHEET 2 of 2

TRAP 17

FLOATING POINT TRAPPING

During floating point operations (or any mathematical operation for that matter) it is necessary to constantly test the answer to determine if it has exceeded the register capacity. To eliminate this continuous, time consuming job of testing with actual instructions, 7094 circuitry automatically assumes the responsibility of continuously monitoring for overflow and underflow conditions. If these conditions arise in either the accumulator 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 identification are:

1- Position 12 - Indicates that a double precision or double load (DLD) instruction addressed an odd location as its data reference

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

3- Position 15 - Indicates an overflow in either the accumulator or MQ.

4- Position 16 - Indicates an overflow or underflow in the accumulator.

5- 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:

Floating-Point Operation	Accumulator	MQ	Positions 12	14	15	16	17	Octal Code
Add, Subtract		Underflow	0	0	0	0	1	01
Multiply,	Underflow	Underflow	0	0	0	1	1	03
Round	Overflow		0	0	1	1	0	06
	Overflow	Overflow	0	0	1	1	1	07
Divide		Underflow	0	1	0	0	1	11
	Underflow		0	1	0	1	0	12
	Underflow	Underflow	0	1	0	1	1	13
		Overflow	0	1	1	0	1	15
DLD or any double precision instruction refering to an odd addressed data location			1	0	0	0	0	20

Notice that detection of an odd-address trap stops the floating point operation and prevents any possible recognition of overflow or underflow conditions.

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

1. Underflow or underflow of the floating point characteristic is detected by examining positions P and Q of the adders and accumulator.
2. MQ overflow and underflow must be recognized as the MQ characteristic is being computed in the adders.
3. Accumulator overflow or underflow may be recognized at any time after the final characteristic has been assigned.

In each of the above cases:

a. Overflow is recognized by a bit in positin P only.

b. Underflow is recognized by bits in both positions P and Q.

Floating Point Trap Operation (Figure)

Objectives of the floating point trap operation are to:

1. Take priority over other possible traps.
2. Store the contents of the program counter (location of the FP instruction +1) into the address portion of location 00000_8.
3. Store identification bits in the decrement portion of location 00000_8.
4. Trap the computer program to location 00010_8.

The various conditions causing a floating point trap were listed in the previous section. The floating point trap trigger (02.10.51.1) is turned ON during the early part of the I cycle following the trapping instruction: I0 time for an odd-address data condition, or I4 time for the remaining conditions. Regardless of the cause, the floating point trap trigger being ON by I4 time allows the first objective of taking priority over other possible traps. If several types of trapping situations occur simultaneously, the floating point trap assumes priority and is serviced first.

At I6 time following the FP trap request, the incoming instruction is blocked from entering the program register; instead, the computer is forced into an STR operation by inserting bits into PR positions S and 9 (03.04.00.1 and 03.04.06.1).

The second objective is to store the location of the FP instruction +1 into the address portion of location 00000_8. The instruction address is retained in the program counter by blocking the normal program counter advance at I8 time of the forced STR operation. The address register is reset at I9 time so that a 00000_8 address is available to MAR for the following E cycle.

Storing the program counter is accomplished during the following E cycle. "MF store address" causes the address portion of the core storage data word to be destroyed on read-out. A data flow path for routing the program counter contents to core storage is through the

index adders, to positions 21 - 35 of the storage register and from there onto the storage bus. During E0 (D3) time the complement of the program counter is routed to the index adders, and the complement of the index adders to the storage register; the net effect of the two complement routings is to place the true program counter value into the storage register. During E4 (D3) time the storage register is gated to the storage bus and from there into location 00000_8. Notice that when the storage register is set at E2 time, positions S, 1-20 are cleared.

The third objective is to store identification bits into the decrement of location 00000_8. "MF store decrement" (02.09.01.1) is activated by the floating point trap condition and causes the decrement portion of the location 00000_8 to be destroyed on read-out. The address portion is destroyed by the forced store and trap; therefore, only the prefix and tag bits remain unchanged. Notice that the identification bits are not routed to the storage bus via the storage register; instead, they are OR'ed directly to the storage bus during the E cycle.

By E6 time, the program counter has been saved in core storage and is reset. At E10 time, a bit is forced into program counter position 14 where it is sent to MAR. The program, therefore, traps to location 00010_8 to accomplish the fourth objective.

To complete the operation, the various trapping indicators are reset; the DP trap trigger (double precision odd-address condition) at E7 time and the remaining triggers at the following I1 time.

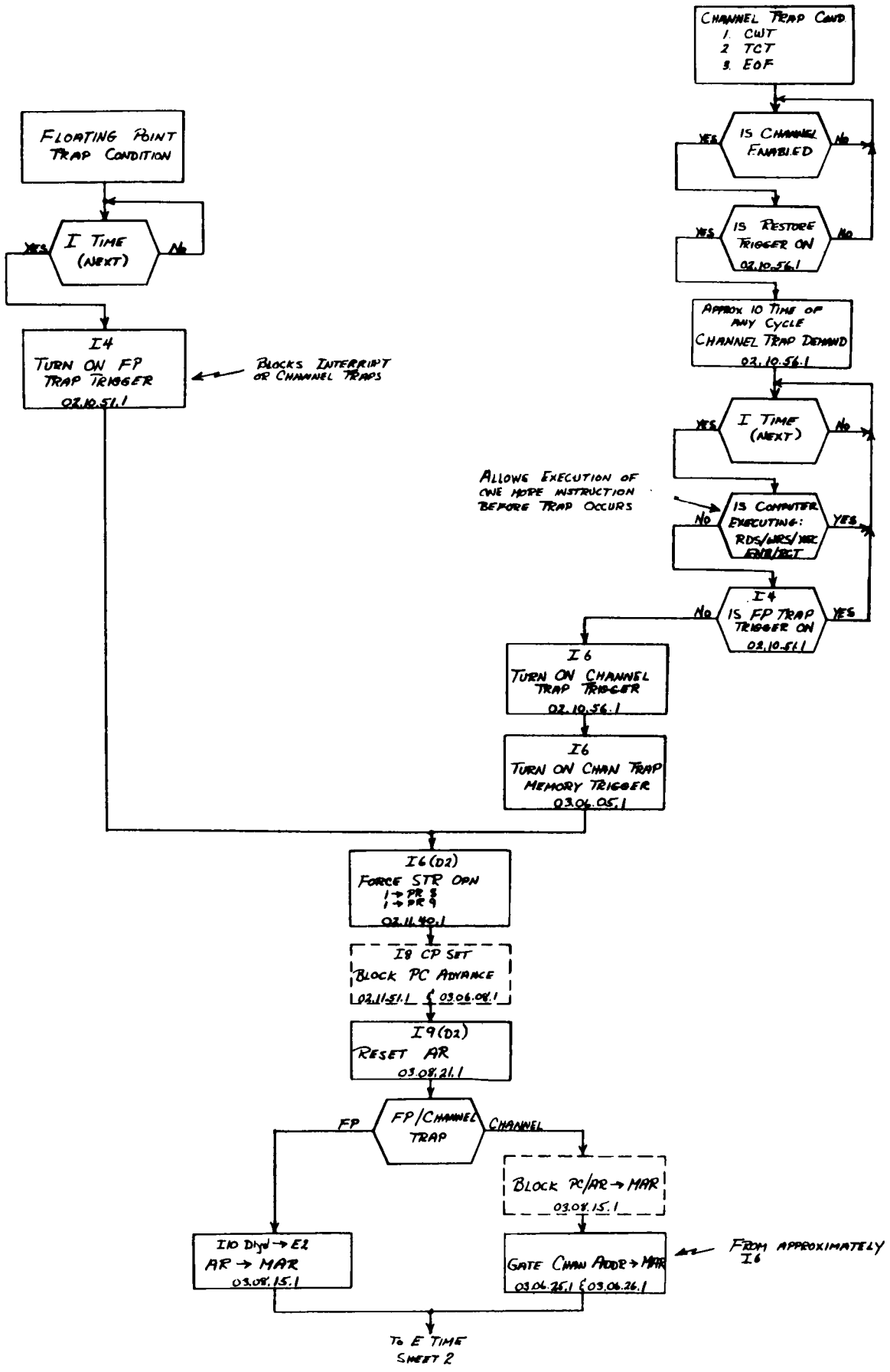

FIGURE I/P/DATA CHANNEL TRAP FLOW CHART

SHEET 1 of 3

(FPT 5)

TRAP 22

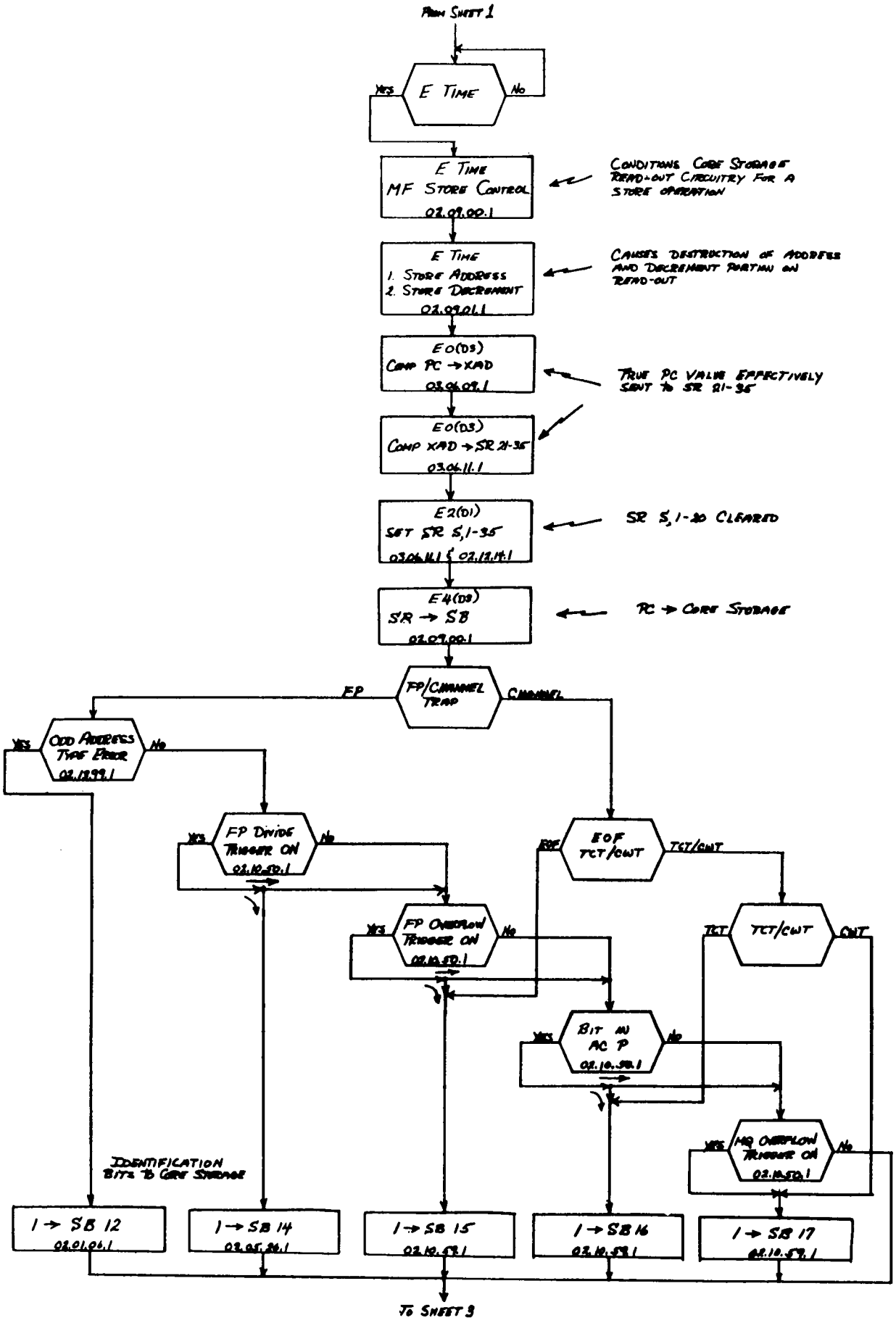

FIGURE FP/DATA CHANNEL TRAP FLOW CHART

SHEET 2 OF 3

(FPT 6)
TRAP 23

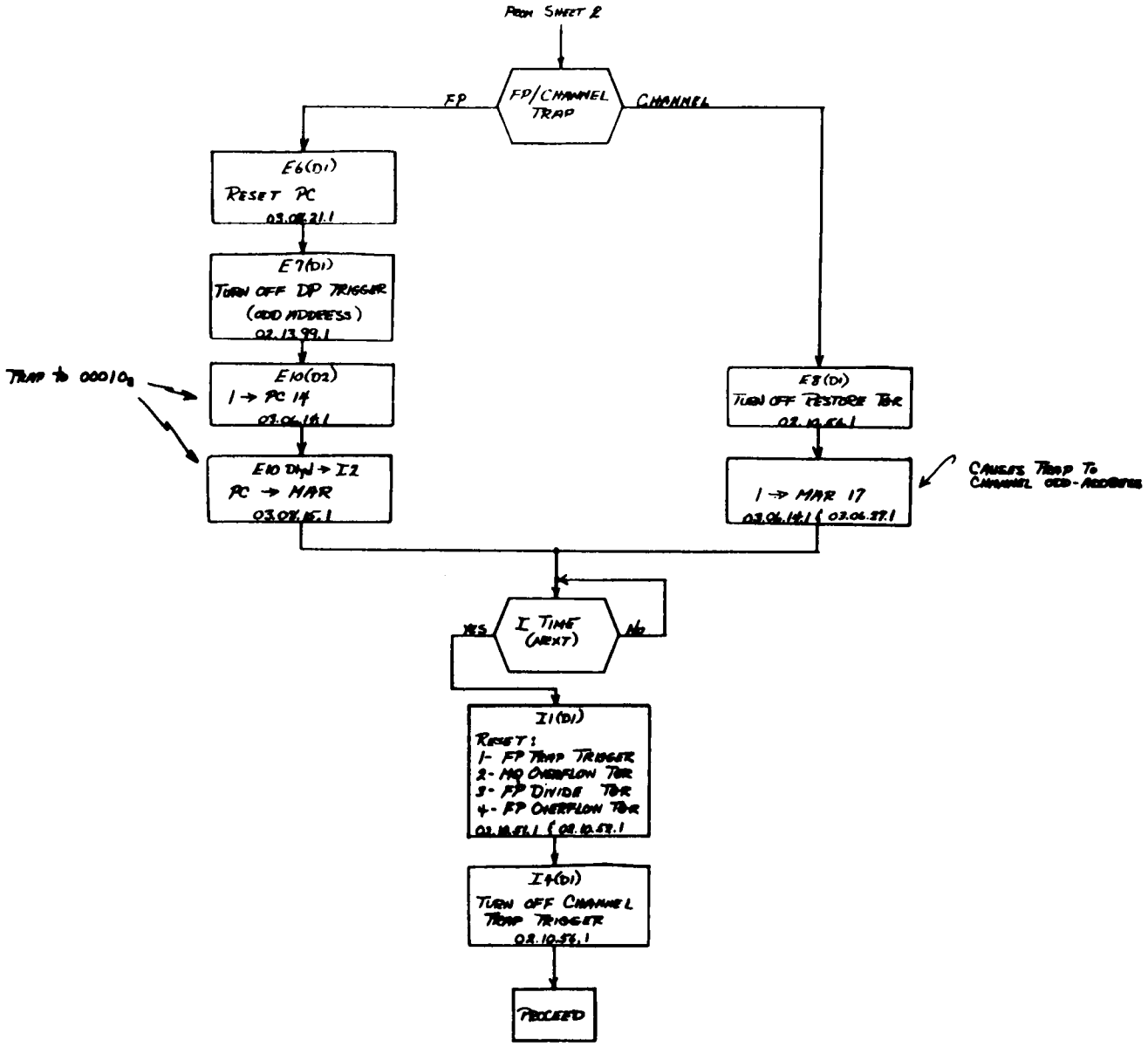

FIGURE FP/DATA CHANNEL TRAP FLOW CHART

SHEET 3 of 3

(PPT 7)

TRAP 24

DATA CHANNEL TRAPPING (Figure)

Without the trapping feature valuable instruction time must be used to test channel conditions or 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.

The three types of channel traps and their meanings are:

1. Control Word Trap (CWT) - 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. This CWT trap can also occur with the 7909 data channel because of a Trap and Wait (TWT) command.

2. Tape Check Trap (TCT) - Recognition of a redundancy check condition in the 7607 data channel.

3. 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 as follows:

CHANNEL	STORE THE PROGRAM COUNTER AT	EXECUTE THE NEXT INSTRUCTION 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 as follows:

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 have been properly enabled by an Enable (ENB) instruction. Any one or any combination of the trap conditions can be enabled by the one instruction. A trap condition that arises when the channel is not enabled for it, is remembered until such time that a proper ENB is executed for the channel or until the trap condition is reset (Figure).

Channel traps are somewhat low on the order of priority for being serviced by the computer. This causes no concern because any delay of a channel trap does not interfere with the I-O operation; every 7607 data channel trap is accompanied by an disconnect condition within the channel.

Channel traps are normally serviced the following I time after being received. However, special conditions (excluding Interrupts which are a special feature) can delay trapping in the computer:

1. If the computer is executing a floating point or DLD isntruction which requires a FP trap, the channel trap is delayed until after the FP trap has stored the program counter and trapped to location 00010_8.

2. If the computer is in the process of executing a POD 34 (CAS/LAS) instruction. During POD 34 execution, the program counter is incremented to anticipate a skip 1 or 2 condition and, therefore, does not indicate a correct value to be stored into location 00000_8.

3. Any trap demand arriving during the execution of a read select, write select, ENB, or RCT instruction is delayed for one instruction. For the read or write select, this

delay prevents a possible I-O Check by allowing a Reset and Load Channel (RCH) instruction to be executed before entering the trap subroutine; for the ENB or RCT, it allows a transfer to be completed back to the main program.

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

An important point to remember, too, is if the instruction in the odd trap location does not alter the contents of the program counter, the program resumes from the point at which the trap occurred.

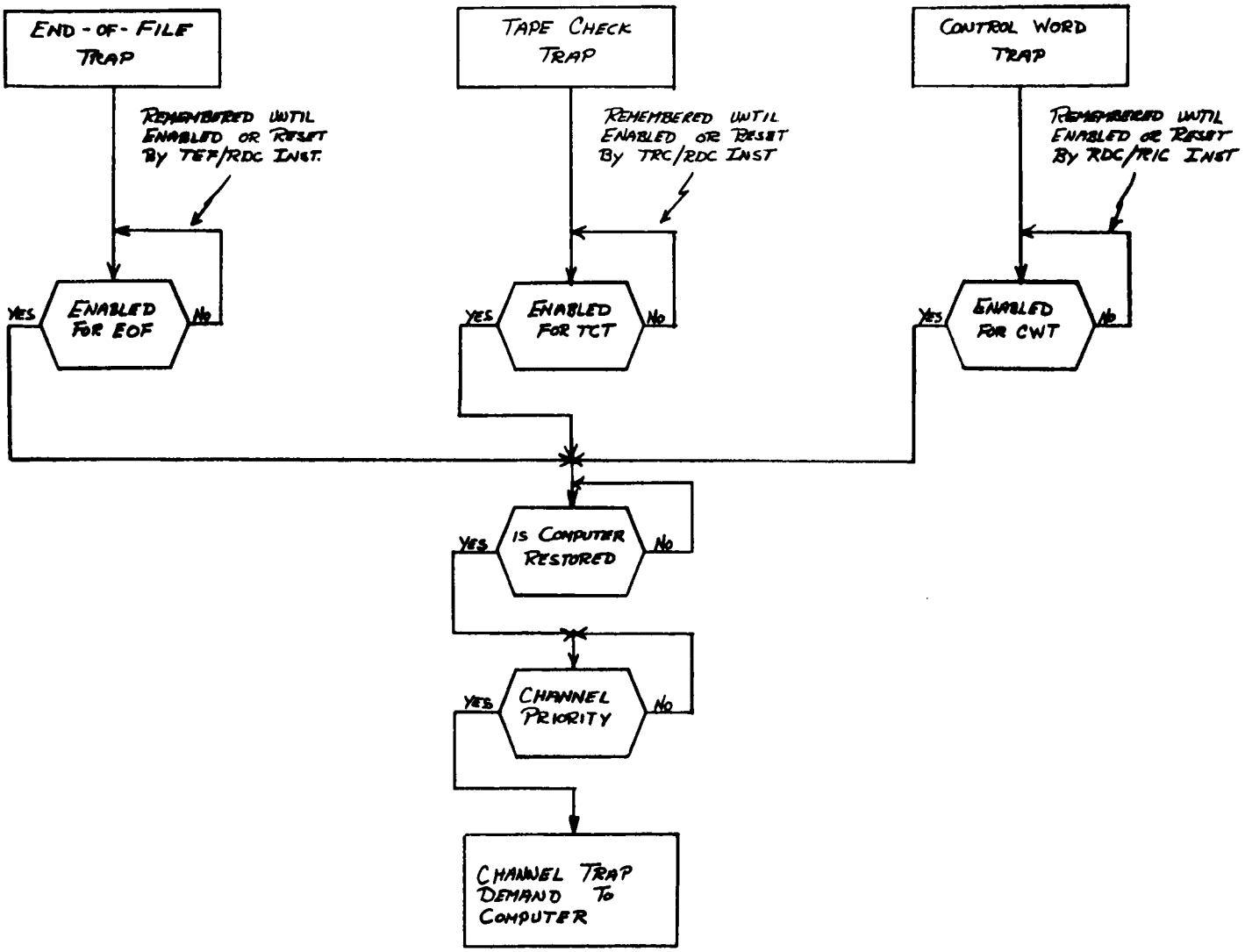

FIGURE CHANNEL TRAP SEQUENCE

B. 11/62

Data Channel Trap Instructions

Enable - ENB +0564 I, E (Figure)

The object of the ENB instruction is to enable (condition) the I-O system for various traps. Bit combinations in the data word referred to by the ENB instruction determine which traps are to be enabled on the attached channels

DATA CHANNEL	DECREMENT POSITION (TCT Trap)	ADDRESS POSITION (CWT or 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

During the initial I cycle, the address register is set to the core storage reference and the computer proceeds to E time. Enable decoding is sent to all channels where it is mixed with an early E-time pulse to reset all enable triggers. It is in this manner that each ENB instruction cancels the effect of previous ENB instructions.

At approximately half way through the E cycle, the data word is read out of core storage and set into the MDR. It is at this time that the outputs (MDBO) are decoded in the multiplexor circuits; corresponding signals are sent to the channels where enabling triggers are turned ON.

At the following I1 time the computer's restore trigger is turned ON and its output gated to the various channels. If a trap request is waiting in the channel at this time, a trap demand is returned for servicing. Notice, however, that this demand will be delayed until after execution of the instruction following the ENB. This gives the programmer an opportunity of transferring back to his main program before being confronted by a second trap condition. When the trap is finally serviced, the restore trigger is turned OFF and further trapping is inhibited until execution of a subsequent ENB or RCT instruction.

Overlapping is not allowed while executing the ENB instruction. Cycle transition of either I/E or E/I would cause only half of the necessary E cycle to be sent to the channel circuitry.

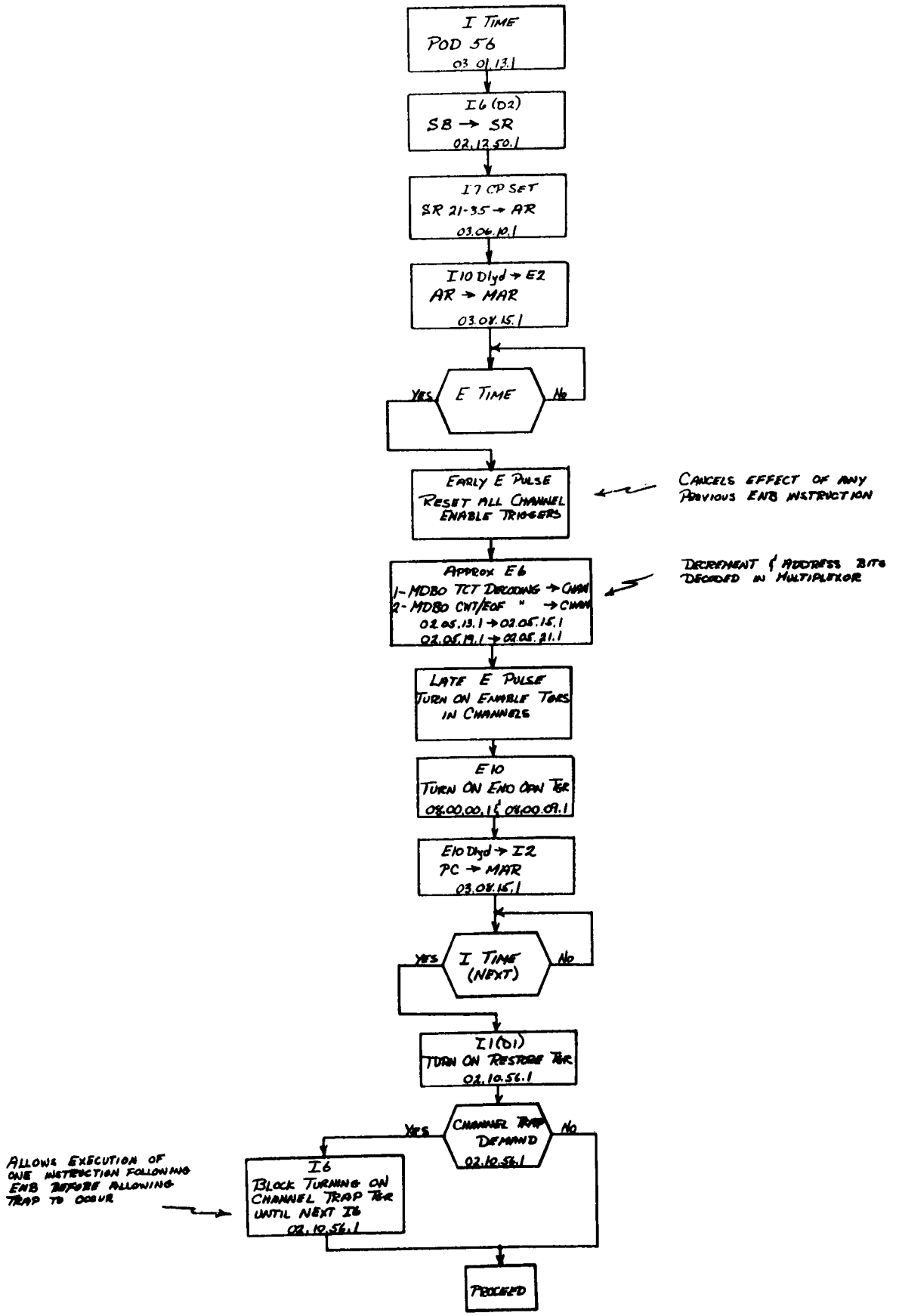

FIGURE ENB FLOW CHART

(ENB 3)

Restore Channel Traps - RCT +0760 ··· 00014 I, L (Figure)

When a channel trap is serviced, all further trapping is inhibited until either an ENB or RCT instruction is executed. The Restore Channel Trap (RCT) instruction removes this inhibiting effect and allows subsequent trapping to occur. Execution of the RCT does not effect the enabling statuses in the various channels as originally set up by an ENB instruction.

Because this is a POD 76 instruction, positions 21 - 35 of the storage register are routed to the shift counter via the address register and index adders. "Restore gate" results from the decoding of 14_8 in shift counter positions 14-17 and contributes to "restore" decoding on 02.10.61.1.

The RCT instruction follows the normal POD 76 cyclic makeup of an I, L cycle. The L cycle performs no logic except to act as a means of ending operation.

During the following I time, the operation is identical to that of ENB; the restore trigger is turned ON at I1 time (02,10.56.1) and its outputs gated to the various channels. If a trap request is waiting in the channel at this time, a trap demand is returned for servicing. Notice again, that this demand will be delayed until after execution of the instruction following the RCT. This gives the programmer an opportunity of transferring back into the main program before a second trap is initiated.

Data overlap is possible by an instruction in the next higher odd location.

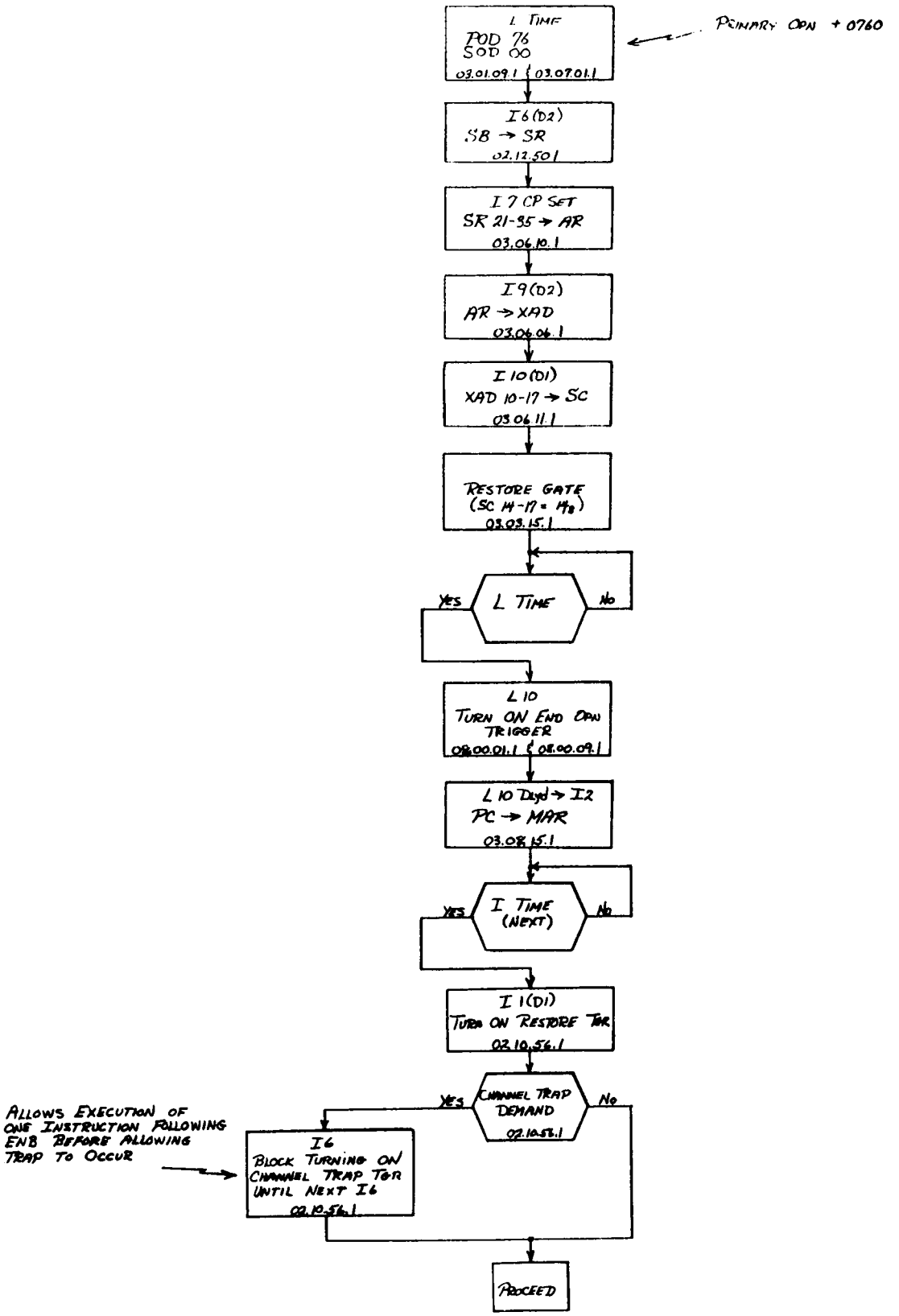

FIGURE RCT FLOW CHART

(RCT 2)

TRAP 33

Data Channel Trap Operation (Figure)

A trap condition must first have the channel enabled before it can make any request to the computer. If enabled, the channel then seeks priority with the other channels in case there are simultaneous requests. Once the request is initiated, the computer still has the ability to block trapping until such a time that the programmer sees fit; this is accomplished by the restore trigger (02.10.56.1) and either an ENB or RCT instruction.

Execution of the trap has three main objectives:

1. To store the contents of the program counter (location of the next instruction to be executed in the computer program) into the address portion of the specified channel trap even address (00012_8 for channel A).
2. To store identification bits in the decrement portion of the channel trap even address.
3. Trap the computer program to cause execution of the instruction in the specified trap odd address (00013_8 for channel A).

Assuming that a floating point or interrupt trap has not taken priority, the channel trap trigger is turned ON at I6 time of the cycle following receipt of a channel trap demand at the computer. The incoming instruction is blocked from being set into the program register; instead, the computer is forced into an STR operation by inserting bits into PR positions S and 9 (03.04.00.1 and 03.04.06.1).

In accomplishing the first objective, the program counter is blocked from the normal stepping during the forced STR I time. In this manner, the address of the next instruction to be executed in the main program is retained in the program counter. The address register is reset to 00000_8 at I9 time of the STR but its output is not used for the core storage reference. Instead, "MAR gate" (03.08.15.1) is blocked

and the channel supplies its own specific "channel trap address" to the multiplexor MAR circuits.

Storing the program counter occurs during the following E cycle. "MF store address" causes the address portion of the core storage location to be destroyed on read-out. Storing the program counter is similar to the traps discussed previously: the program counter is routed through the index adders and set into positions 21 - 35 of the storage register; double complementing of the program counter and index adder outputs effectively places the true value in the storage register. The storage register is gated onto the storage bus during E4 (D3) time; positions S, 1-20 of the storage register are cleared.

The second objective is to store the identification bits in the decrement. These bits are OR'ed to the storage bus positions 15, 16, or 17 after testing for the appropriate indication from the channel. "MF store decrement" (02.09.01.1) is activated by the channel trap condition and causes the decrement portion of the channel trap even address to be destroyed on read-out. With information stored in the address and decrement portions of the trap address, only the prefix and tag positions premain unchanged.

The third objective must now be performed; that of trapping to the specific channel trap odd address. Notice at this point that the program counter is not reset as it was during the floating point trap operation. Because of this, if the instruction at the trap address is not some type of an unconditional transfer, the program continues in sequence again under control of the program counter following execution of the odd - address trap instruction. At E8 time of the forced STR operation, the restore trigger (02.10.56.1) is turned OFF to prevent any further trapping until execution of a subsequent ENB or RCT instruction. The restore trigger going OFF at E8 time activates "chan trap

TRA" (03.06.14.1) which in turn forces a bit onto MAR 17 (03.06.27.1). This, in addition to the MAR lines still active due to the "channel trap address" (03.06.25.1 and 03.06.26.1) causes the computer to execute the instruction in the next, odd location.

At the next I4 time, the channel trap trigger is turned OFF and the operation is completed.

When the channel trap trigger was initially turned ON, the channel trap memory trigger was also turned ON (03.06.05.1). This memory trigger remains ON until the I cycle following execution of the odd-address trap instruction and prevents the normal I time stepping of the program counter. In this manner, the program resumes sequantial operation in the main program if the odd address instruction was not an unconditional transfer.

CUT ALONG LINE

COMMENT SHEET

IBM 7094 DATA PROCESSING SYSTEM (SUPPLEMENT)

CUSTOMER ENGINEERING INSTRUCTION-REFERENCE, FORM S23-4002

FROM

NAME _____

OFFICE NO. _____

FOLD

CHECK ONE OF THE COMMENTS AND EXPLAIN IN THE SPACE PROVIDED

FOLD

☐ 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.

ATTN: CE MANUALS, DEPARTMENT 296

CUT ALONG LINE

FOLD

FOLD

STAPLE

STAPLE