TELECOMMUNICATIONS ALARM SURVEILLANCE AND CONTROL (TASC) SYSTEM

PROGRAM SEQUENCER GENERIC 1 AND GENERIC 2

CONTENTS	PAGE
1. GENERAL	1
2. INSTRUCTION SET	3
3. GENERATING A PSEQ JOB	10
A. Defining the Problem	10
B. Flowcharting	11
C. Coding the PS and PF Files	11
D. Testing and Debugging	11
E. Keyboard Entry	15
4. EXECUTION	15
A. Manual Execution	15
B. Automatic Execution	19
C. Security Considerations	20
5. LISTING	20
A. PS or PF File	20
B. Program Sequencer Data Base (PSDB)	20
C. PSEQ File Directory	23
6. EDITING	23
7. ADVANCED TECHNIQUES	25
8. EXAMPLES	26
A. Stuck Sender Hold	26
B. Fire Alarm	27
C. Status Summaries	27
D. Make Busy A Trouble Recorder	27
E. Restoration Plan	27
9. ERROR MESSAGES	28
10. DELETE A PS OR PF FILE	28
11. REFERENCES	28

1. GENERAL

1.01 This section describes the program sequencer feature. The use of this feature can result in more efficient operation of the telecommunications alarm surveillance and control (TASC) central. Read the following sections before attempting to master the material contained in this section:

SECTION	TITLE
190-210-101	TASC System Description
190-210-102	TASC Central Description for Generic 1
190-210-112	TASC Central Description for Generic 2
190-210-300	TASC Central Operations for Generic 1
190-210-302	TASC Data Base Operations for Generic 1
190-210-310	TASC Central Operations for Generic 2
190-210-312	TASC Data Base Operations for Generic 2

Printed in U.S.A.

1.02 This section has been reissued to update the technical data contained to reflect TASC Generic 2.

1.03 Program sequencer (PSEQ) is a software package which provides field personnel with the ability to create and execute meaningful jobs. These jobs describe an ordered sequence of TASC central related/initiated actions which perform a specific task.

1.04 One possible function for PSEQ would be to provide restoration plans for transmission routes. This job would describe the actions required to provide this restoration, and could be executed by a simple one-line keyboard command request. The actions required to perform the restoration plan are described in the PSEQ job; ie, which remote switches are to be operated, which displays or scans for alarm information are to be initiated, etc. The advantage of PSEQ is that it could initiate the job much faster and more accurately than a series of requests performed by the TASC attendant.

1.05 PSEQ jobs can also be executed automatically when an alarm or status point is set. For example, a PSEQ job could execute automatically on the occurrence of a fire alarm. If this happens, a bell will ring several times at the attendants terminal. In addition, a message will be printed indicating that there is a fire in a particular station. This action will be repeated every 5 seconds until either the fire alarm bit clears or the attendant initiates a PSEQ job which was constructed to terminate this PSEQ job.

1.06 As previously mentioned, each PSEQ job describes actions required to accomplish a particular task. This description is in the form of a series of steps, and each step represents a PSEQ instruction. These instructions, described in detail later, provide the ability to initiate a remote switch, scan for information, test the scan point information to determine if certain scan points are set or clear, log messages into the history log, operate the terminal alarm bell, terminate alarm processing of received alarms, and print messages on an output device such as a terminal or line printer. There are other instructions which provide capabilities a PSEQ job can perform, and they will also be described later.

1.07 PSEQ instructions are stored in files. A PSEQ file is a list of data stored on a disc drive which defines the steps required to perform a task. Two files are required for each PSEQ job; they are:

PS—Program Step File

PF—Parameter File

The PS file is an ordered list of data which describes each instruction in a PSEQ job on a step-by-step basis. The PF file, also an ordered list of data, contains the parameters (if any) associated with each instruction in the PS file. For example, a step in the PS file could be described as a remote switch instruction, while the PF file would contain the switch number and remote station number associated with the switch.

1.08 Each PS or PF file must be given a unique name. This name can be up to six characters in length. The first character must not be a blank or a number. Embedded blanks are not allowed. Permissible characters are:

A thru Z
0 thru 9
’’
’ (Apos)
(
)

.
/
;
<
=
>
\
]
[
↑

1.09 Meaningful names should be given to the PS and PF files. For example, a name for a file which provides a restoration plan can be given the name RSP001. This would convey more meaning than a name such as A9>. Also, special consideration should be given to the PF file name since the PSEQ job takes this name.

1.10 All PSEQ jobs produce outputs to one or more output LU numbers. At the very least, there will be a **SEQUENCE COMPLETE** message appearing. Other messages result from errors and PRINT steps, GO steps, SCAN steps, and VERIFICATION steps. The destinations of the outputs can be directed in several ways:

(1) The PR and GO steps each have a self contained output LU specified by the creator of the sequence. These LUs are used unless there is an override LU. This is a case when a sequence is manually executed with an LU specified in the command (**#SEQ,ABC,L29**), or as in the case when the sequence is scheduled by an alarm through an entry in the program sequence data base (PSDB) that specifies an output LU. In both of these cases, the override LU is used for all functions.

(2) The SCAN and VERIFY results, and also the general messages of errors or completion, are directed according to the following priority scheme:

(a) For manual execution (**#SEQ**)

—to the LU number specified with the **SEQ** command

—if no LU specified previously to the LU number specified in the PF file

—if no LU specified previously to the LU from which the sequence was requested.

(b) For automatic execution (via **PSDB**)

—to the LU number specified in the PSDB

—if no LU specified previously to the LU number specified in the PF file

—if no LU specified previously to LU #1.

All the information contained in paragraph 1.10 is summarized in Table A.

1.11 PSEQ jobs are stored in the real-time executive (RTE) file manager area on the fixed disc (LU #3) for Generic 1, or LU #44 in TASC Generic 2. Since there is a finite number of tracks available to the file manager (FMGR) which are shared by various features (SCO, BCO, HIS and PSEQ), users of sequencer may be required to carefully control the number and size of sequences.

1.12 The PS file describes the sequence and type of steps in the PSEQ job. When generating a PSEQ job, the PS file must be created first and the parameters next in the PF file. The PSEQ software is arranged such that more than one PSEQ job can use the same PS file. A maximum of 7 sequences using the same PS file can be executed simultaneously. Each PSEQ job ***must*** have a separate PF file.

2. INSTRUCTION SET

2.01 The PSEQ software provides a set of instructions which can be used to create a PSEQ job

TABLE A

	SEQ, ABC	SEQ, ABC, LX	PSDB, &	PSDB, LX

PF, &	P=S
	E=CO
		P=LX
		E=LX
			P=S
			E=LU#1
				P=LX
				E=LX

PF, LZ	P=S
	E=LZ
		P=LX
		E=LX
			P=S
			E=LZ
				P=LX
				E=LX

Where:

P = PRINT and GO messages
E = Error messages, END messages, and SCAN results

S = as specified in the steps of the PF file
CO = on the requesting console
LX = on logical unit X
LZ = on logical unit Z
& = default value

(see Table B for a summary of all the PSEQ instructions). The following is a description of each of these instructions.

PRINT

The instruction prints an operator-programmed message on a specified output device.

Entry in PS file—**PR**

Parameters in PF file:

(1) Output device

(2) ASCII message

The message may be up to 248 characters in length. Any ASCII character except the exclamation point (!) may be used to construct the message. The output device symbolic code may be any of those specified in Table C.

> *Note 1:* Messages should be kept as short as possible to conserve storage space in the file manager.

> *Note 2:* Entering control G in a print step will result in the ringing of the terminal bell during execution.

ALARM

The ALARM instruction can turn on or off the office major or minor alarm bell which is connected to the TCT.

Entry in PS file—**AL**

Parameters in PF file:

(1) Alarm bell (major or minor)

(2) Turn on (set) or off (clear)

LOG

The LOG instruction logs an operator-programmed message in the history log by identification number (**IDNO**), class number (**CLA#**), or station number (**STA**).

Entry in PS file: **LG**

Parameters in PF file:

(1) ASCII message

(2) Log by IDNO, CLA#, or STA#

The message may be up to 248 characters in length. Any ASCII character, except the exclamation point (!), may be used to construct the message. The message may be logged into the history log by IDNO, CLA# (1-900), or STA# (0-256). Identification numbers are sequentially tagged to alarms that are detected by the TASC central. If the PF file indicates station 0, the LOG will be applied to the station referenced in the most recently executed RS or SC step. An example of an IDNO is the 4-digit number outputted with each alarm and most command operations. Class numbers are assigned during generation of the TASC data base to certain categories of alarms. For example, radio channel alarms could be assigned a class number, and messages could be stored or retrieved via this number.

WAIT

The WAIT instruction provides a preprogrammed delay before execution of the next instruction.

Entry in PS file: **WT**

Parameters in PF file:

A delay specified in either seconds, minutes, hours, or days (**SE, MI, HO, DA**).

TABLE B

STEP	ABBREVIATION	DESCRIPTION
ALARM	AL	Turn on or off major or minor alarm bell
CALL	CA	Call another PSEQ job
MCO	CO	Clear or set a MCO for a particular station
END	EN	Terminates the execution of a PSEQ job
FLAG	FL	Set, increment, or test the flag value
GO	GO	Causes CONTINUE (Y, N, OR STEP#) ? to appear at a selected I/O device
INITIALIZE	IN	Return the state of the PF file to the original operator defined state
JUMP	JP	Branch to a specified instruction step number
LOG	LG	Log a message in the history log
LOOP	LP	Repeat a series of steps a specified number of times
PRINT	PR	Print an operator programmed message on a specified output device
RSW	RS	Perform a remote switch operation at an E- or C-type remote
SCAN	SC	Initiate a status-report type operation at a specified E- or C-type device
SCO	SO	Set or clear an SCO for a particular station
TERMINAL ALARM	TA	Send an alarm indication (bell) to any specified TASC terminal
VERIFY	VR	Compare bits of the previous scan with those in the PF file
WAIT	WT	Preprogrammed delay before execution of the next instruction

TABLE C

INPUT/OUTPUT DEVICE	LU#OR SYMBOL TASC G1	LU#OR SYMBOL TASC G2
Printer at TASC Center	L6 or PR	L6 or PR
Main Console	L1	L1
Partial-Feature Terminals	L25, L26, L27, L28, L30, L31, L32, L33	L23, L24, L25, L26, L27, L28, L29, L30 L31, L32, L33, L34, L35, L36, L37, L38, L39, L40, L41
Full-Feature Terminals	L23, L24, L29	
Default to Terminal Where Entry is Made	CO	CO
Communication Links (ATA)	L51, L53, L55, L57, L59 (see Note)	L51, L53, L55, L57, L59 (see Note)

Note: When an ATA link LU# is entered, the attendant must also enter the station number.

The minimum wait interval is 1 second, and the maximum is 22 days. Longer waits can be created by multiple wait instructions.

GO

The GO instruction causes **CONTINUE (Y, N, OR STEP#)?** to be displayed at a selected I/O device. This in effect causes the PSEQ job execution to be stopped until the attendant types either **Y, N, OR STEP#**.

Where **Y** = continue with job at the next step
Where **N** = abort job
Where **STEP#** = continue with the job at the specified step.

Entry in PS file: **GO**

Parameters in PF file:

I/O device on which the **CONTINUE (Y, N, OR STEP#)?** message is to be displayed (see Table C).

The **GO** instruction is intended to be used after a PRINT instruction which would describe the operation that is about to occur.

Example: LAST PIECE OF COMMON EQUIPMENT ABOUT TO GO
CONTINUE (Y, N, OR STEP#)?

JUMP

The JUMP instruction causes an unconditional jump to a specified instruction step number.

Entry in PS file: **JP**

Parameters in PF file: Step number

REMOTE SWITCH

This instruction causes a remote switch operation to be performed at either an E- or C-type remote station.

Entry in PS file: **RS**

Parameters in PF file:

(1) Station number

(2) Type station (E- or C-type)

(3) Switch number

(4) Switch type (momentary, latch operate, or latch release)

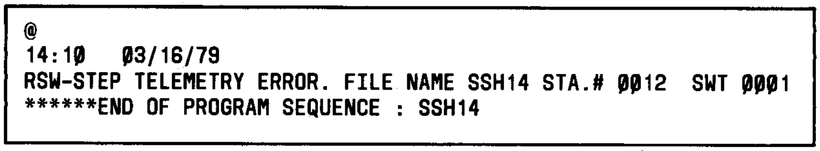
(5) If desired, log by class number or ID number, or do not log.

(6) Class number (if logging by class).

(7) (***For Generic 2 Only***) If switch verification is desired, scan number and bit number. Step number for continuation if continuation is to occur upon verification error.

If it is desired to log this operation, the message that is logged is not operator-defined. The log message is created by the PSEQ software and contains information to identify the station number, switch number, and the PF file name.

> ***Note:*** A telemetry error on an RS step causes the sequence to terminate and output an appropriate error message. Refer to Fig. 1 for an example of this.



```
@
14:1Ø  Ø3/16/79
RSW-STEP TELEMETRY ERROR. FILE NAME SSH14 STA.# ØØ12  SWT ØØØ1
******END OF PROGRAM SEQUENCE : SSH14
```

Fig. 1—Example of Error Message Outputted When a Telemetry Error Occurs on a RS Step of a PSEQ Job

SCAN

The SCAN instruction causes a status-report type operation to be initiated at a specified E- or C-type remote. The current state of all alarms within the scan is printed.

Entry in PS file: **SC**

Parameters in PF file:

(1) Station number

(2) CUR operation (Yes or No)

(3) If it is not a CUR operation, then:

(a) Scan number

(b) Regular or history type of scan.

(4) (***For Generic 2 Only***) Is printing to be suppressed?

(5) (***For Generic 2 Only***) Is logging to be suppressed?

> ***Note:*** A telemetry error on an SC step causes the sequence to terminate and output an appropriate error message. Refer to Fig. 2 for an example of this error message.

VERIFY

The VERIFY instruction compares the status and alarm bits in the previous scan with the bits specified in the PF file. If the bits compare, the PSEQ job continues with the next instruction. If the bits do not compare (ie, verify errors), the PSEQ job can be terminated or the job can be made to jump to an instruction specified in the PF file.

Entry in PS file: **VR**

Parameters in PF file:

(1) Bits, if any, that should be set (Logic 1)

(2) Bits, if any, that should be clear (Logic 0)

(3) If verify error occurs, terminate PSEQ job or jump to another instruction.

The verify instruction uses the scan results of the most recent scan instruction in the job. It is allowable to use more than one verify instruction without intervening scan instructions.

> ***Note:*** This cannot be used with a CUR operation.

CALL

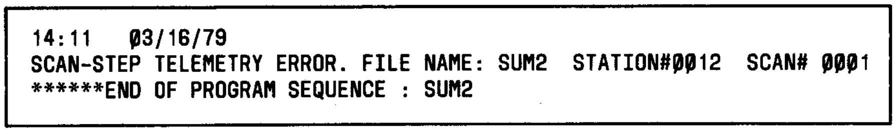
The CALL instruction provides the ability for one PSEQ job to call for execution of another PSEQ job.

Entry in PS file: **CA**

Parameters in PF file:

PF file name of PSEQ job to be called.

This instruction is useful when it is desired to split a lengthy PSEQ job into many parts. After testing and debugging, the parts can be linked together via the **CALL** instruction. Another use is to link existing PSEQ jobs to a newly created job. This saves disc storage since the new job does not have to provide storage for duplication of the existing job. Other details of the **CALL** instruction are given in paragraph 7.02.



```
14:11   Ø3/16/79
SCAN-STEP TELEMETRY ERROR. FILE NAME: SUM2  STATION#ØØ12  SCAN# ØØØ1
******END OF PROGRAM SEQUENCE : SUM2
```

Fig. 2—Example of Error Message Outputted When a Telemetry Error Occurs on a SC Step of a PSEQ Job

TERMINAL ALARM

This instruction sends an alarm indication (bell) to any specified TASC terminal.

Entry in PS file: **TA**

Parameters in PF file:

LU number of terminal

FLAG

The FLAG instruction administers a set of 1536 storage locations (flags) which numerical values in the range of −32,767 through 32,767 may be manipulated by the sequencer. The values are preserved even when no sequence jobs are running, so they may be accessed at a later time by the same or by a different sequence job.

Entry in PS file: **FL**

Entry in PF file:

(1) Flag number

(2) Set the flag to a specified (absolute) value

(3) Increment the flag by specified (change) value (use negative value to decrement)

(4) Compare the flag value to a specified value and branch to one of three specified steps depending upon whether the comparison yields <, = , or >.

Note:

- The flag value may be manually changed by the attendant via the **EDI** command. When the computer outputs the message:

  TYPE OF DATA? Enter: FLAG.

- The flag value may be listed via the **LIS** command. When the computer outputs the message:

  TYPE OF DATA? Enter: FLAG.

- A flag value may be cleared (set to Ø) via the **CLE** command. When the computer outputs the message:

  TYPE OF DATA? Enter: FLAG. Next, enter the flag number which is to be cleared.

- All flag values may be cleared (set to Ø) via the **INT** command. When the computer outputs the message:

  TYPE OF DATA? Enter: FLAG.

Refer to paragraph 7.03 for further details.

LOOP

The LOOP instruction provides the ability to repeat a series of steps a specified number of times.

Entry in PS file: **LP**

Parameters in PF file:

(1) The step which is the beginning of the loop.

(2) Number which represents one less than the number of times the loop is to be repeated.

A loop can be programmed to perform a series of steps from 1 to 9999 times. After the programmed number of loops have been completed, the instruction immediately after the loop instruction will be executed.

MCO

The MCO instruction terminates the processing of alarms by the TASC central for a particular station.

Entry in PS file: **CO**

Parameters in PF file:

(1) Station number

(2) Terminate (set) or reinstate (clear) alarm processing.

If the PF file indicates station Ø, the MCO will be applied to the station referenced in the most recently executed RS or SC step. Alarms associated with stations that have their alarm processing terminated are ignored by the TASC central and do not result in the ringing of the office alarm bell.

SCO

The SCO instruction will set or clear an SCO for a particular station.

Entry in PS file: **SO**

Parameters in PF file:

(1) Station number

(2) SCO number

(3) Set or Clear.

INITIALIZE

The INITIALIZE instruction will return the state of the parameter file to the original operator defined state.

Entry in PS file: **IN**

Parameters in PF file: None

Refer to paragraphs 7.04 through 7.06 for details concerning the use of this instruction.

END

The END instruction terminates execution of the PSEQ job. It is the last logical instruction to be executed in the PSEQ sequence. More than one **END** instruction may occur if the sequence has several branches.

Entry in PS file: **EN**

Parameter in PF file: None

/E

The /E is not one of the PSEQ instructions, but should be the last physical step in any PSEQ job. Only one **/E** is allowed in each PSEQ job. An **EN** instruction ***must immediately precede*** the **/E** instruction.

Entry in PS file: **/E**

Parameters in PF file: None

3. GENERATING A PSEQ JOB

A. Defining the Problem

3.01 The first step in generating a PSEQ job is to define the problem. This definition might take the form of a paragraph or a listing of steps. The definition should provide details as to which remote stations, scans, etc, are involved. With TASC Generic 2 the definition of the problem should include data that can be used to provide sequence security. The following questions should be answered:

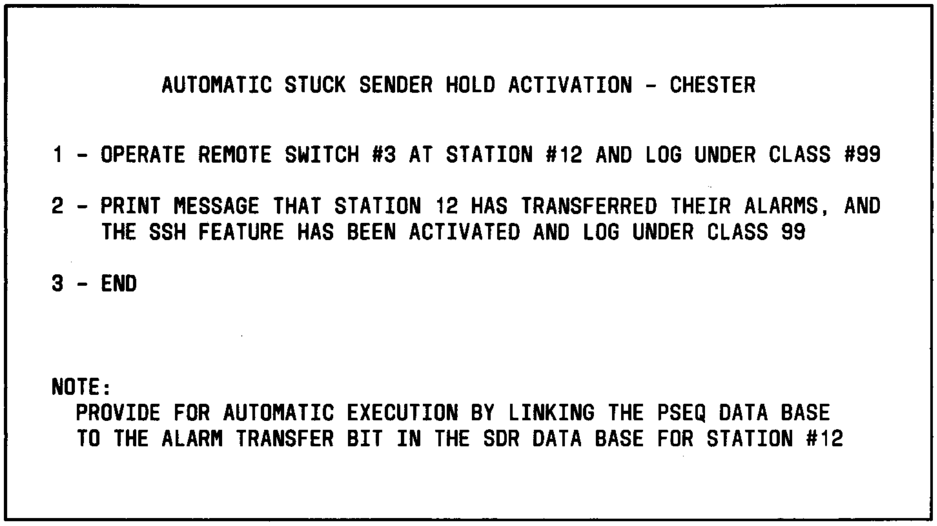
- Is alarm activation allowed (Y or N)?
- What LUs are allowed to manually activate this sequence?
- What LUs are allowable override readout devices?
- What LU is to receive "END OF SEQUENCE" messages if no override LU is specified?

Figure 3 gives the basic outline of a PSEQ job whose function is to activate the stuck sender hold (SSH) feature at the station. It is to be automatically executed based on the transfer of alarms to the TASC attendant. The output for this PSEQ job is as follows:

STATION 12 HAS TRANSFERRED THEIR ALARMS, AND THE SSH FEATURE HAS BEEN ACTIVATED AND LOGGED UNDER CLASS 99.

********END OF PROGRAM SEQUENCE: SSH12**

3.02 In order for this PSEQ job to execute automatically, the sequence must be linked to the alarm transfer bit. This is accomplished by linking the PSEQ data base to the SDR data base. The PSEQ data base provides a cross-reference of the station number, SDR or scan number, and scan point (bit) number. The scan point must also have the processing program PSFD attached to it in the SDR file (see Part 4B—Automatic Execution).



```
AUTOMATIC STUCK SENDER HOLD ACTIVATION - CHESTER

1 - OPERATE REMOTE SWITCH #3 AT STATION #12 AND LOG UNDER CLASS #99

2 - PRINT MESSAGE THAT STATION 12 HAS TRANSFERRED THEIR ALARMS, AND
    THE SSH FEATURE HAS BEEN ACTIVATED AND LOG UNDER CLASS 99

3 - END


NOTE:
  PROVIDE FOR AUTOMATIC EXECUTION BY LINKING THE PSEQ DATA BASE
  TO THE ALARM TRANSFER BIT IN THE SDR DATA BASE FOR STATION #12
```

Fig. 3—Example PSEQ Job Description

B. Flowcharting

3.03 From the definitions of the problem, a flowchart can be made. Each block in the flowchart should be equivalent to one of the instructions previously described. Suggested flowchart symbols are shown in Fig. 4. Note the ***Verify*** and ***Flag*** decision blocks. In a ***Verify*** instruction, if no errors, the job continues with the next instruction. If a verify error occurs, the attendant can cause the PSEQ job to either terminate or jump to a specified instruction. The ***FLAG*** instruction (test flag) allows the ability to jump to 1 of 3 specified steps depending on whether the present flag value is <, >, or = to the specified value. Flowchart FC-1 is an example of a flowchart developed from the definition in Fig. 3. Please note that when referring to a flowchart, the number of that flowchart will be referenced and titled FC-x (example FC-1).

C. Coding the PS and PF Files

3.04 From the PSEQ job flowchart, a PS and PF file can be generated which can be used to enter the job into the TASC computer. To generate a PS or PF file, first translate the flowchart into code via the instructions previously described. Each block in the flowchart, if properly developed, represents one of the instructions. To generate the PS file, simply list these instructions as a series of steps. The parameters for each step are used to generate the PF file. Figure 5A is an example of the PS and PF files generated from the flowchart FC-1. Figure 5B is a form which can be copied and used to generate the PS and PF files.

D. Testing and Debugging

3.05 After completing the code for the PS and PF file, it may be wise to let another person who is knowledgeable with program sequencer examine the work. The reason for this is that testing and debugging a new program sequencer job is accomplished on a "live" TASC System. Defects in the PSEQ job could be catastrophic as far as the monitored equipment is concerned. Again, extreme care should be used in the testing and debugging phase of PSEQ job generation.

3.06 PSEQ jobs that are intended to be run automatically should be debugged before providing for automatic execution. Execute new PSEQ jobs manually via the SEQ command until they are completely debugged. The SEQ command will be described later in paragraph 4.01.

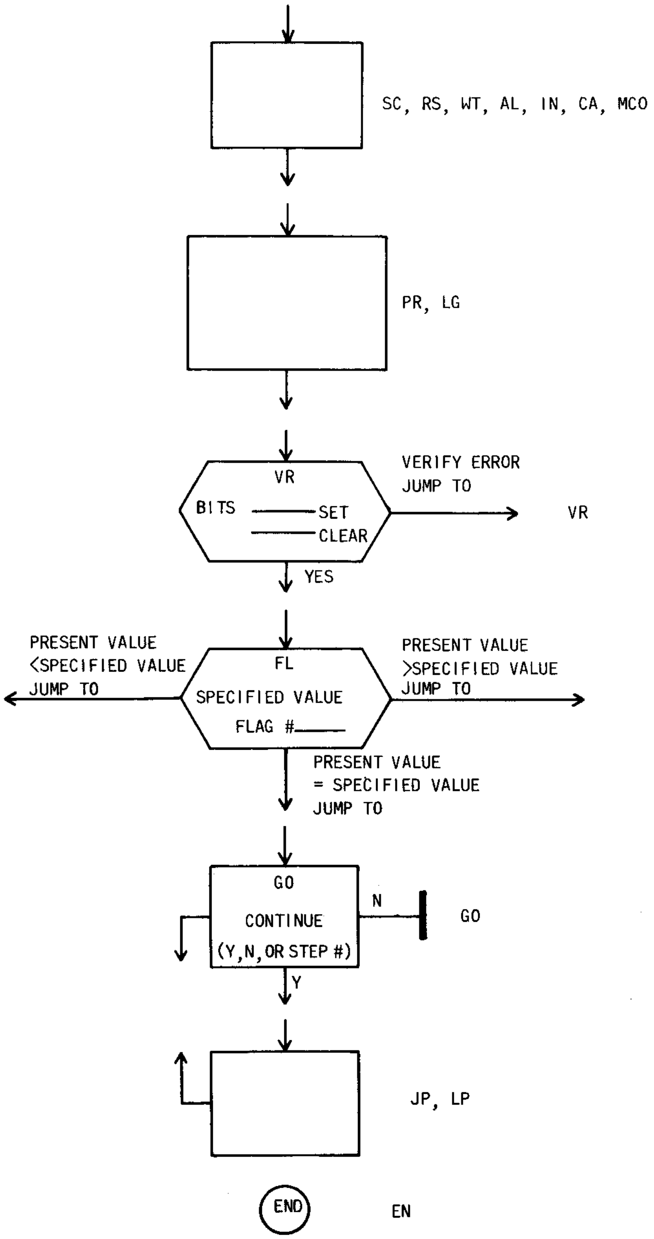

SC, RS, WT, AL, IN, CA, MCO
PR, LG
VR
BITS
SET
CLEAR
VERIFY ERROR JUMP TO
VR
YES
PRESENT VALUE <SPECIFIED VALUE JUMP TO
FL
SPECIFIED VALUE
FLAG #
PRESENT VALUE >SPECIFIED VALUE JUMP TO
PRESENT VALUE = SPECIFIED VALUE JUMP TO
GO
CONTINUE
(Y,N,OR STEP #)
N
GO
Y
JP, LP
END
EN

Fig. 4—Flowchart Symbols

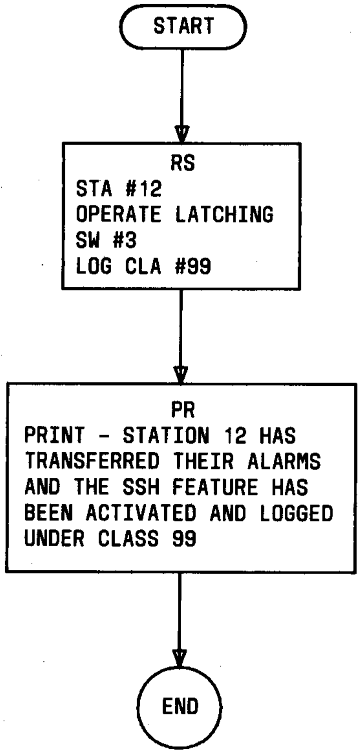
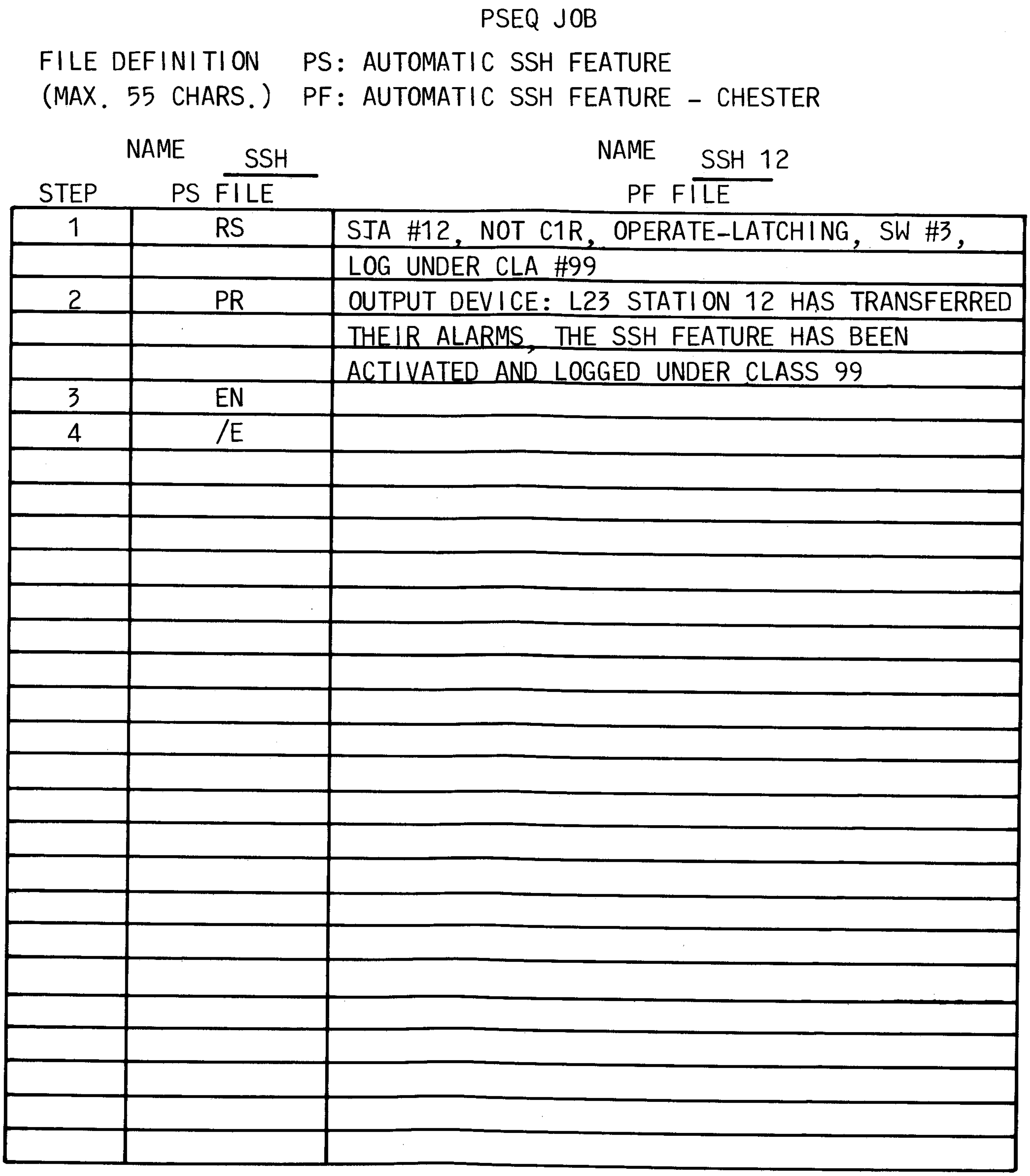

START
RS
STA #12
OPERATE LATCHING
SW #3
LOG CLA #99
PR
PRINT - STATION 12 HAS TRANSFERRED THEIR ALARMS AND THE SSH FEATURE HAS BEEN ACTIVATED AND LOGGED UNDER CLASS 99
END

Fig. FC1—Example of a PSEQ Job Flowchart—Automatic SSH PSEQ Job



PSEQ JOB

FILE DEFINITION PS: AUTOMATIC SSH FEATURE
(MAX. 55 CHARS.) PF: AUTOMATIC SSH FEATURE - CHESTER

NAME SSH

NAME SSH 12

STEP	PS FILE	PF FILE
1	RS	STA #12, NOT C1R, OPERATE-LATCHING, SW #3,
		LOG UNDER CLA #99
2	PR	OUTPUT DEVICE: L23 STATION 12 HAS TRANSFERRED
		THEIR ALARMS, THE SSH FEATURE HAS BEEN
		ACTIVATED AND LOGGED UNDER CLASS 99
3	EN	
4	/E	

Fig. 5A—PSEQ Job Work Sheet Example

PSEQ JOB

FILE DEFINITION PS:
(MAX. 55 CHARS.) PF:

STEP	NAME ______ PS FILE	NAME ______ PF FILE

Fig. 5B—Blank PSEQ Job Work Sheet

E. Keyboard Entry

3.07 The PSEQ job is entered into the TASC central by initiating the **TAS** command at an appropriate keyboard terminal. The **TAS** command can be initiated from any full-feature terminal by simply typing **TAS**, followed by a carriage return. The TASC central will then return several lines of messages similar to Fig. 6. **TAS** can also be entered, followed by an associated number, and the printout of Fig. 6 will not occur. For example, if it is necessary to create a PS file, enter **TAS,10**.

3.08 To enter a PS or PF file, enter **10** when the computer responds with **ENTER NUMBER:**. The computer will begin asking for information; respond accordingly. Figure 7 is an example of the printout from entry of the code in Fig. 5A for the PS file. Figures 8 and 9 represent the entry for the corresponding PF files in TASC Generic 1 and 2, respectively. After the file has been entered, the computer will respond with **INPUT FILE DEFINITION (54 CHARS MAX)**. Here the operator may enter a 55-character description of the file. This is useful in that later the operator can request a listing of the file directory and a cross-reference of each file name versus the file description. The keyboard entry for the file directory is described in paragraph 5.04.

3.09 The other features (20 through 90) of the TAS command will be discussed later.

4. EXECUTION

A. Manual Execution

4.01 There are two methods to manually execute a PSEQ job; they are:

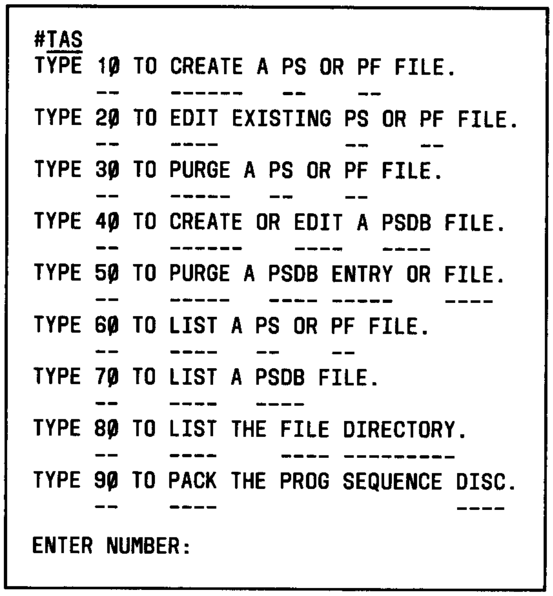
- Interactive entries
- Noninteractive entries.

The interactive entry is accomplished by simply typing **SEQ** followed by a carriage return. The central will respond with **INPUT PF FILE NAME MAX 6 CHARS:**. Type the PF file name associated with the PSEQ job followed by a carriage return. The central will then respond with **READOUT DEVICE (& IF DEFAULT)?** Generally, the attendant's response will be **&**.

4.02 The SEQ command can also be executed noninteractively as follows. First, type **SEQ** followed by a comma; type the PF file name, followed by a comma; then type the logical unit number for the output device and follow this with a carriage return. An example of this is SEQ,RSP101,CO. Note that the noninteractive way of manually executing a PSEQ job requires only one line of keyboard entry versus three lines for the interactive entry.

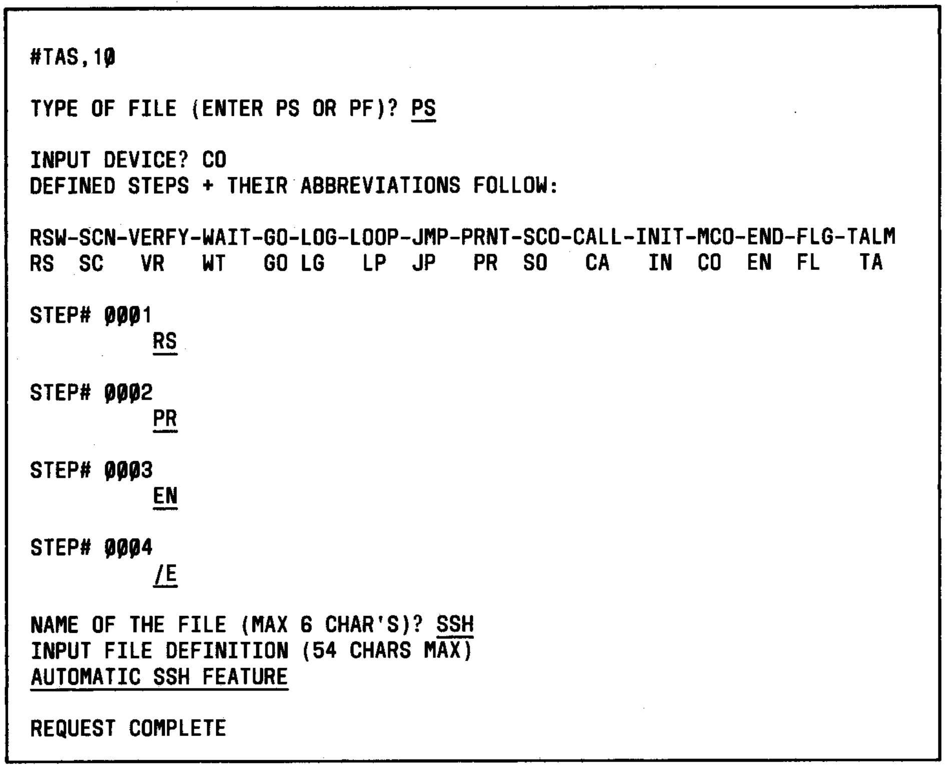
Note:

- Sequence initiation is logged under station #257, class #997.
- All active sequences will be terminated upon bootup. If the sequence was in a WAIT step when the system was halted it will appear active if during boot up a request for restoring system parameters is entered.



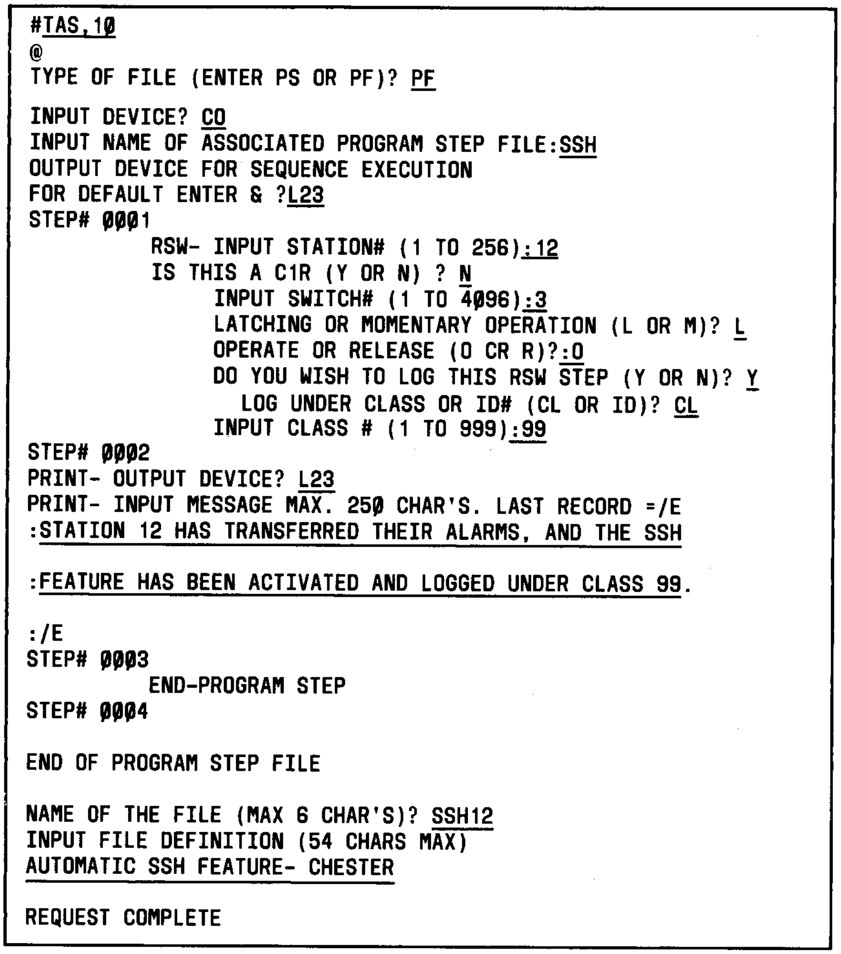
```
#TAS
----
TYPE 1Ø TO CREATE A PS OR PF FILE.
     --    ------   --    --
TYPE 2Ø TO EDIT EXISTING PS OR PF FILE.
     --    ----          --    --
TYPE 3Ø TO PURGE A PS OR PF FILE.
     --    -----   --    --
TYPE 4Ø TO CREATE OR EDIT A PSDB FILE.
     --    ------    ----   ----
TYPE 5Ø TO PURGE A PSDB ENTRY OR FILE.
     --    -----   ---- -----    ----
TYPE 6Ø TO LIST A PS OR PF FILE.
     --    ----   --    --
TYPE 7Ø TO LIST A PSDB FILE.
     --    ----   ----
TYPE 8Ø TO LIST THE FILE DIRECTORY.
     --    ----     ---- ---------
TYPE 9Ø TO PACK THE PROG SEQUENCE DISC.
     --    ----                   ----
ENTER NUMBER:
```

Fig. 6—Printout From Initiating the TAS Command



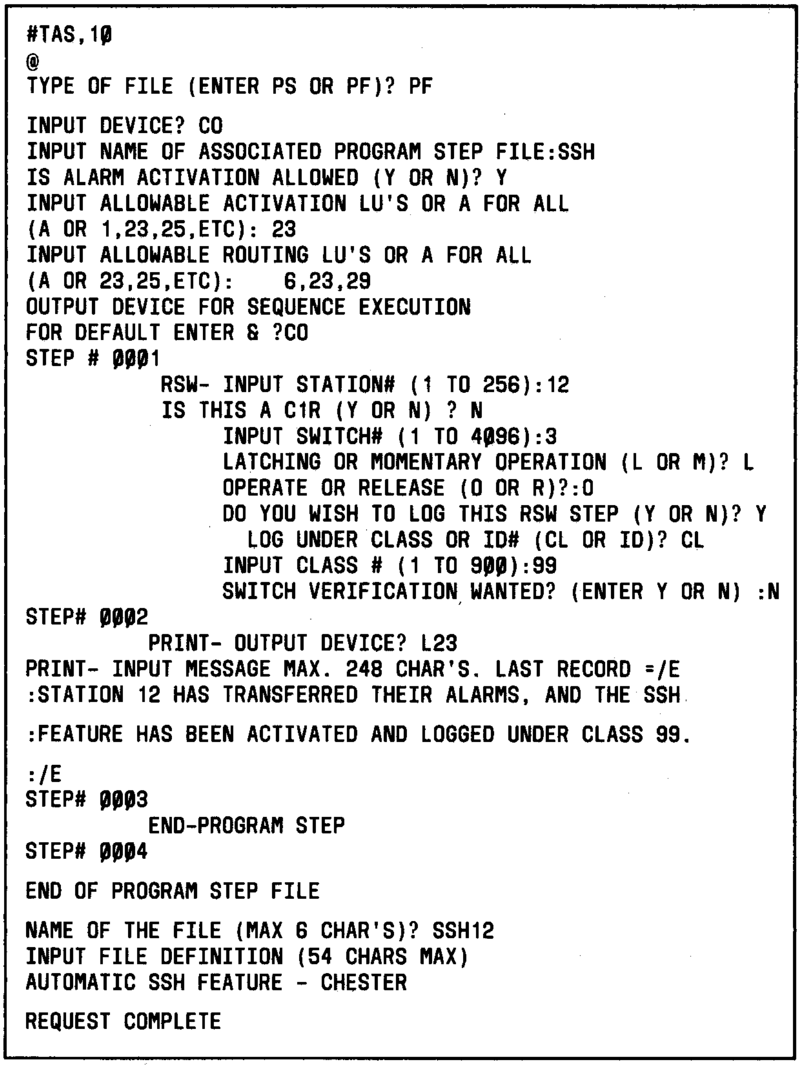
```
#TAS,1Ø

TYPE OF FILE (ENTER PS OR PF)? PS

INPUT DEVICE? CO
DEFINED STEPS + THEIR ABBREVIATIONS FOLLOW:

RSW-SCN-VERFY-WAIT-GO-LOG-LOOP-JMP-PRNT-SCO-CALL-INIT-MCO-END-FLG-TALM
RS  SC   VR   WT   GO LG   LP  JP   PR  SO   CA   IN  CO  EN  FL   TA

STEP# ØØØ1
          RS

STEP# ØØØ2
          PR

STEP# ØØØ3
          EN

STEP# ØØØ4
          /E

NAME OF THE FILE (MAX 6 CHAR'S)? SSH
INPUT FILE DEFINITION (54 CHARS MAX)
AUTOMATIC SSH FEATURE

REQUEST COMPLETE
```

Fig. 7—Printout That Occurs When Ending a PS File



```
#TAS,1Ø
@
TYPE OF FILE (ENTER PS OR PF)? PF

INPUT DEVICE? CO
INPUT NAME OF ASSOCIATED PROGRAM STEP FILE:SSH
OUTPUT DEVICE FOR SEQUENCE EXECUTION
FOR DEFAULT ENTER & ?L23
STEP# ØØØ1
          RSW- INPUT STATION# (1 TO 256):12
          IS THIS A C1R (Y OR N) ? N
               INPUT SWITCH# (1 TO 4Ø96):3
               LATCHING OR MOMENTARY OPERATION (L OR M)? L
               OPERATE OR RELEASE (O CR R)?:O
               DO YOU WISH TO LOG THIS RSW STEP (Y OR N)? Y
                 LOG UNDER CLASS OR ID# (CL OR ID)? CL
               INPUT CLASS # (1 TO 999):99
STEP# ØØØ2
PRINT- OUTPUT DEVICE? L23
PRINT- INPUT MESSAGE MAX. 25Ø CHAR'S. LAST RECORD =/E
:STATION 12 HAS TRANSFERRED THEIR ALARMS, AND THE SSH

:FEATURE HAS BEEN ACTIVATED AND LOGGED UNDER CLASS 99.

:/E
STEP# ØØØ3
          END-PROGRAM STEP
STEP# ØØØ4

END OF PROGRAM STEP FILE

NAME OF THE FILE (MAX 6 CHAR'S)? SSH12
INPUT FILE DEFINITION (54 CHARS MAX)
AUTOMATIC SSH FEATURE- CHESTER

REQUEST COMPLETE
```

Fig. 8—Printout That Occurs When Entering a PS File With Generic 1 Software



```
#TAS,1Ø
@
TYPE OF FILE (ENTER PS OR PF)? PF

INPUT DEVICE? CO
INPUT NAME OF ASSOCIATED PROGRAM STEP FILE:SSH
IS ALARM ACTIVATION ALLOWED (Y OR N)? Y
INPUT ALLOWABLE ACTIVATION LU'S OR A FOR ALL
(A OR 1,23,25,ETC): 23
INPUT ALLOWABLE ROUTING LU'S OR A FOR ALL
(A OR 23,25,ETC):    6,23,29
OUTPUT DEVICE FOR SEQUENCE EXECUTION
FOR DEFAULT ENTER & ?CO
STEP # ØØØ1
           RSW- INPUT STATION# (1 TO 256):12
           IS THIS A C1R (Y OR N) ? N
                INPUT SWITCH# (1 TO 4Ø96):3
                LATCHING OR MOMENTARY OPERATION (L OR M)? L
                OPERATE OR RELEASE (O OR R)?:O
                DO YOU WISH TO LOG THIS RSW STEP (Y OR N)? Y
                  LOG UNDER CLASS OR ID# (CL OR ID)? CL
                INPUT CLASS # (1 TO 9ØØ):99
                SWITCH VERIFICATION WANTED? (ENTER Y OR N) :N
STEP# ØØØ2
          PRINT- OUTPUT DEVICE? L23
PRINT- INPUT MESSAGE MAX. 248 CHAR'S. LAST RECORD =/E
:STATION 12 HAS TRANSFERRED THEIR ALARMS, AND THE SSH

:FEATURE HAS BEEN ACTIVATED AND LOGGED UNDER CLASS 99.

:/E
STEP# ØØØ3
          END-PROGRAM STEP
STEP# ØØØ4

END OF PROGRAM STEP FILE

NAME OF THE FILE (MAX 6 CHAR'S)? SSH12
INPUT FILE DEFINITION (54 CHARS MAX)
AUTOMATIC SSH FEATURE - CHESTER

REQUEST COMPLETE
```

Fig. 9—Printout That Occurs When Making an Entry With Generic 2 Software

B. Automatic Execution

4.03 A PSEQ job can be made to execute automatically when a scan point becomes set. This is accomplished by linking the TASC System data base to the PSEQ data base. The PSEQ data base will provide a cross-reference of station number, SDR or scan number, and scan point number versus the name of the PSEQ job (PF file name) to be executed. (Recall that the name of the PSEQ job is taken as the PF file name.) Remember that the scan point must have the processing program PSFD attached to it in the SDR file. Thus, when a scan point becomes set (active), the TASC System data base "links" to the PSEQ data base which in turn selects the appropriate PSEQ job to execute.

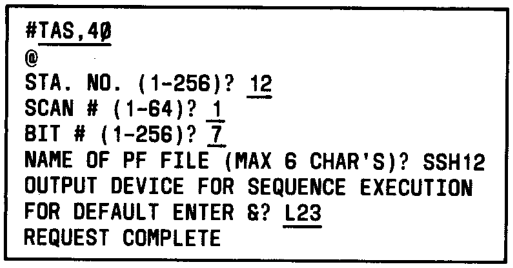
4.04 Figures 10 and 11 are examples of a typical printout of entering information to provide for automatically executing a PSEQ job. To automatically execute of a PSEQ job, first make an entry in the PSEQ data base (PSDB). This is accomplished by initiating the TAS command. To initiate the TAS command, simply type **TAS** followed by a carriage return. A printout similar to Fig. 6 will occur. Where the message **ENTER NUMBER** occurs, type **40** followed by a carriage return. Enter the information requested by the computer as required.

Note:

- Sequence initiation is logged under station #257, class #997.

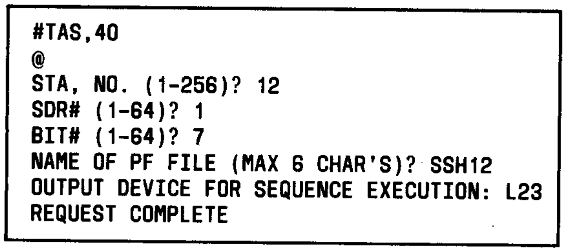
- All active sequences will be terminated upon bootup. If the system was in a WAIT step when the system was halted it will appear active if during boot up a request for restoring system parameters is entered.

4.05 When the PSEQ data base entry has been made, link the TASC data base to the PSEQ data base. This is accomplished by initiating the EDI command. To initiate the EDI command, type **EDI** followed by a carriage return. When the central asks for **TYPE OF DATA?**, type **SDR** followed by a carriage return. Respond to the questions asked by the central. When the processing program name is to be entered, type **PSFD** (Fig. 12). Note, in Fig. 12, bit number 7 is linked to PSFD.



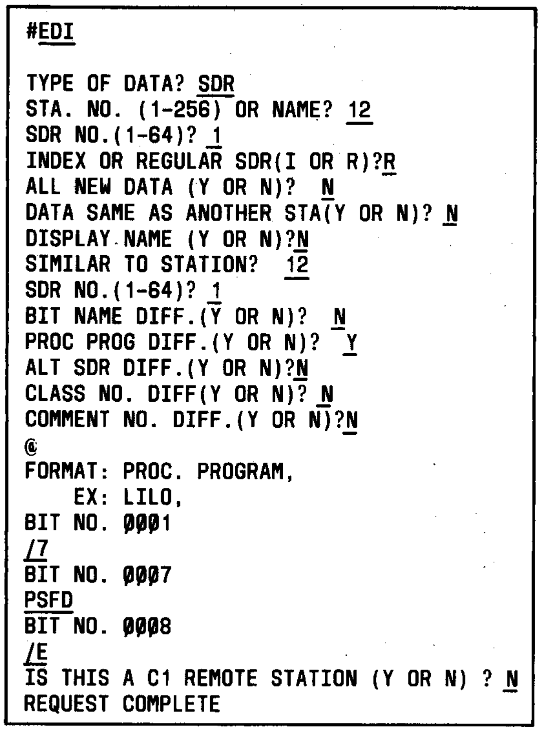
```
#TAS,4Ø
@
STA. NO. (1-256)? 12
SCAN # (1-64)? 1
BIT # (1-256)? 7
NAME OF PF FILE (MAX 6 CHAR'S)? SSH12
OUTPUT DEVICE FOR SEQUENCE EXECUTION
FOR DEFAULT ENTER &? L23
REQUEST COMPLETE
```

Fig. 10—Printout That Occurs When Making an Entry into the PSDB File for TASC Generic 1 Software



```
#TAS,40
@
STA, NO. (1-256)? 12
SDR# (1-64)? 1
BIT# (1-64)? 7
NAME OF PF FILE (MAX 6 CHAR'S)? SSH12
OUTPUT DEVICE FOR SEQUENCE EXECUTION: L23
REQUEST COMPLETE
```

Fig. 11—Printout That Occurs When Making an Entry into the PSDB File for TASC Generic 2 Software



```
#EDI

TYPE OF DATA? SDR
STA. NO. (1-256) OR NAME? 12
SDR NO.(1-64)? 1
INDEX OR REGULAR SDR(I OR R)?R
ALL NEW DATA (Y OR N)?  N
DATA SAME AS ANOTHER STA(Y OR N)? N
DISPLAY NAME (Y OR N)?N
SIMILAR TO STATION?  12
SDR NO.(1-64)? 1
BIT NAME DIFF.(Y OR N)?  N
PROC PROG DIFF.(Y OR N)?  Y
ALT SDR DIFF.(Y OR N)?N
CLASS NO. DIFF(Y OR N)? N
COMMENT NO. DIFF.(Y OR N)?N
@
FORMAT: PROC. PROGRAM,
    EX: LILO,
BIT NO. ØØØ1
/7
BIT NO. ØØØ7
PSFD
BIT NO. ØØØ8
/E
IS THIS A C1 REMOTE STATION (Y OR N) ? N
REQUEST COMPLETE
```

Fig. 12—Printout That Occurs When Making an Entry to Link the TASC Data Base to the PSEQ Data Base

C. Security Considerations

4.06 In TASC Generic 1 software any PSEQ job may be executed manually from any full-feature terminal. The readout device parameter may be specified as any LU. Any PSEQ job may be activated automatically if the SDR and PSDB data bases call for the running of the job.

4.07 In TASC Generic 2 software there is a security feature which places three requirements on all manual execution of a PSEQ job. These requirements are:

- The command filtering table (CFT) for the requesting LU must allow the execution of the SEQ command.
- The security section of the PF file of the PSEQ job must indicate that the requesting LU is an allowable activation LU (LU1 has no special privileges here).
- The security of the PF file must also indicate that the readout device specified in the SEQ command is one of the allowable readout devices. The default readout device (**&**) is always permitted.

4.08 Security is also associated with the automatic execution of any PSEQ job with Generic 2. As in Generic 1, the SDR and PSDB data bases have to be appropriately formatted, but with Generic 2 the security section of the PF file must also indicate that alarm activation is allowable. Furthermore, all of the outputs (PR, SC, GO, etc) form a sequence that has been activated from an alarm and will automatically be directed to the output LU specified in the PSDB file in Generic 2.

5. LISTING

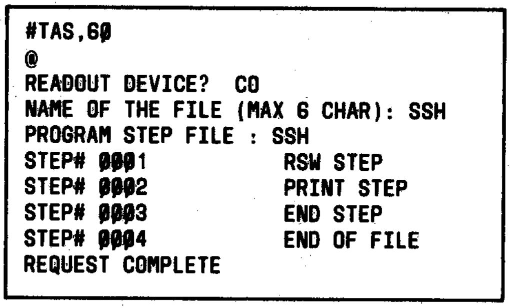
5.01 Listing is accomplished by initiating the TAS command. The TAS command is initiated by simply typing **TAS** followed by a carriage return. The central will return with several lines of messages similar to Fig. 6.

A. PS or PF File

5.02 To list a PS or PF file, type **60** where the central responds with **ENTER NUMBER:**. Follow this entry by a carriage return. Other responses will be required of the attendant, such as the file name and the output device. Permissible entries for the output device are given in Table C. Examples of typical listings are given in Fig. 13, 14, and 15.

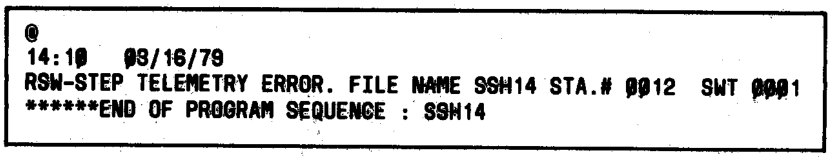
B. Program Sequencer Data Base (PSDB)

5.03 To list the PSDB, type **70** where the computer responds with **ENTER NUMBER:** in the TAS command. Follow this entry by a carriage return. Figure 16 is an example of the listing of a typical PSDB.



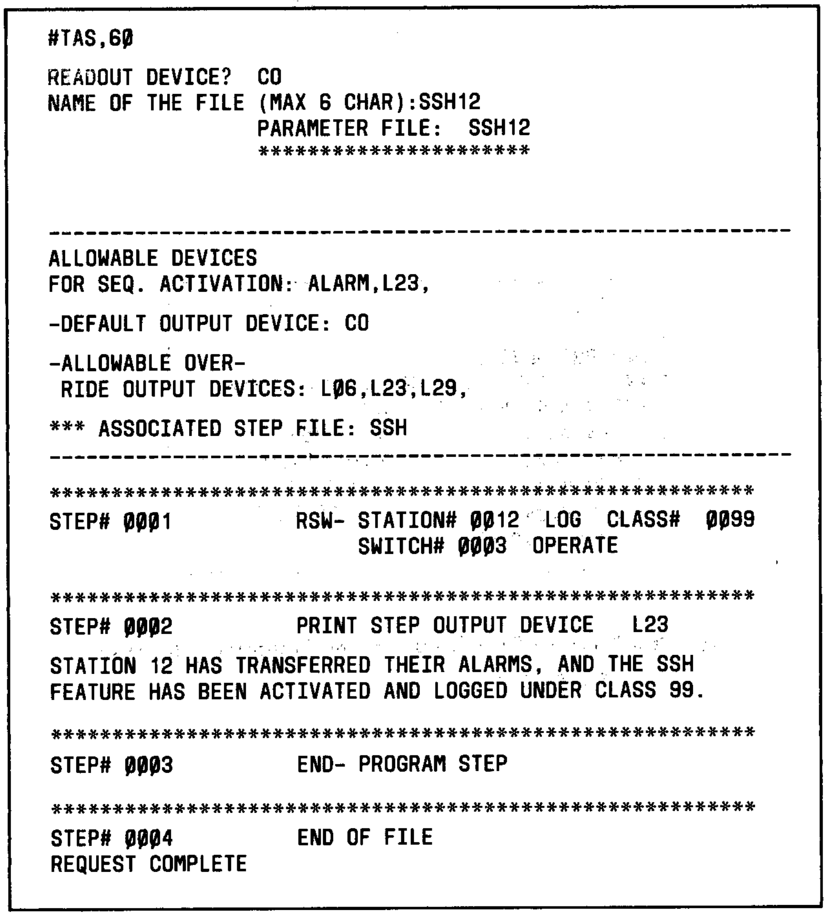
```
#TAS,6Ø
@
READOUT DEVICE?  CO
NAME OF THE FILE (MAX 6 CHAR): SSH
PROGRAM STEP FILE : SSH
STEP# ØØØ1           RSW STEP
STEP# ØØØ2           PRINT STEP
STEP# ØØØ3           END STEP
STEP# ØØØ4           END OF FILE
REQUEST COMPLETE
```

Fig. 13—Printout That Occurs When Requesting a Listing of a PS File



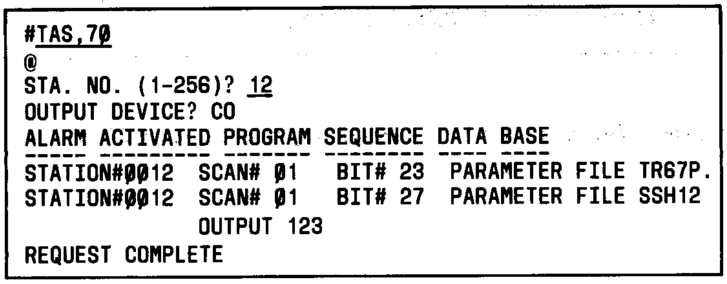
```
@
14:1Ø   Ø8/16/79
RSW-STEP TELEMETRY ERROR. FILE NAME SSH14 STA.# ØØ12  SWT ØØØ1
******END OF PROGRAM SEQUENCE : SSH14
```

Fig. 14—Printout That Occurs When Requesting a Listing of a PF File With Generic 1



```
#TAS,6Ø
READOUT DEVICE?  CO
NAME OF THE FILE (MAX 6 CHAR):SSH12
                 PARAMETER FILE:  SSH12
                 **********************



------------------------------------------------------------
ALLOWABLE DEVICES
FOR SEQ. ACTIVATION: ALARM,L23,
-DEFAULT OUTPUT DEVICE: CO
-ALLOWABLE OVER-
 RIDE OUTPUT DEVICES: LØ6,L23,L29,
*** ASSOCIATED STEP FILE: SSH
------------------------------------------------------------
*********************************************************
STEP# ØØØ1          RSW- STATION# ØØ12  LOG  CLASS#  ØØ99
                         SWITCH# ØØØ3  OPERATE

*********************************************************
STEP# ØØØ2          PRINT STEP OUTPUT DEVICE   L23
STATION 12 HAS TRANSFERRED THEIR ALARMS, AND THE SSH
FEATURE HAS BEEN ACTIVATED AND LOGGED UNDER CLASS 99.
*********************************************************
STEP# ØØØ3          END- PROGRAM STEP
*********************************************************
STEP# ØØØ4          END OF FILE
REQUEST COMPLETE
```

Fig. 15—Printout That Occurs When Requesting a Listing of a PF File With Generic 2



```
#TAS,7Ø
@
STA. NO. (1-256)? 12
OUTPUT DEVICE? CO
ALARM ACTIVATED PROGRAM SEQUENCE DATA BASE
----- --------- ------- -------- ---- ----
STATION#ØØ12  SCAN# Ø1    BIT# 23  PARAMETER FILE TR67P.
STATION#ØØ12  SCAN# Ø1    BIT# 27  PARAMETER FILE SSH12
              OUTPUT 123
REQUEST COMPLETE
```

Fig. 16—Printout That Occurs When Requesting a Listing of the PSDB File Entry

C. PSEQ File Directory

5.04 The PSEQ file directory lists each PS, PF, and PSDB files that have been entered into the TASC central. This file will also list any active PF files. To list this directory, type **80** where the computer responds with **ENTER NUMBER:** in the TAS command. Follow this entry by a carriage return. Figure 17 is an example of a PSEQ file directory listing.

6. EDITING

6.01 Editing is accomplished by using the TAS command as the following summary indicates:

Generic 1 Software

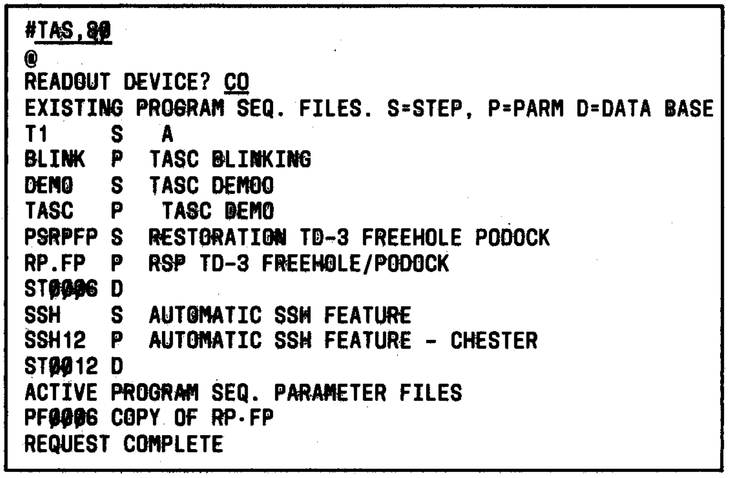
TAS, 20 edits PS or PF files
TAS, 30 deletes PS or PF files
TAS, 50 deletes PSDB entries.

Generic 2 Software

TAS, 20 edits PF files only
TAS, 30 deletes PS or PF files
TAS, 50 deletes PSDB entries.

With Generic 1, any full-feature terminal may make a change as described before. With Generic 2, the terminal making the changes must satisfy the following two requirements:

- It must have the TAS command specified in its command filtering table (CFT).
- It must be specified as an allowable activation device in the PF file itself (the one exception to this is in the deletion of a PS file where there is no security specified).



```
#TAS,80
@
READOUT DEVICE? CO
EXISTING PROGRAM SEQ. FILES. S=STEP, P=PARM D=DATA BASE
T1     S   A
BLINK  P  TASC BLINKING
DEMO   S  TASC DEMOO
TASC   P   TASC DEMO
PSRPFP S  RESTORATION TD-3 FREEHOLE PODOCK
RP.FP  P  RSP TD-3 FREEHOLE/PODOCK
ST0006 D
SSH    S  AUTOMATIC SSH FEATURE
SSH12  P  AUTOMATIC SSH FEATURE - CHESTER
ST0012 D
ACTIVE PROGRAM SEQ. PARAMETER FILES
PF0006 COPY OF RP·FP
REQUEST COMPLETE
```

Fig. 17—Printout That Occurs When Requesting a Listing of the File Directory

6.02 To delete a program step (PS) file or a parameter file (PF), enter **TAS, 30**. Figure 18 is an example of this operation. To delete a PSEQ data base (PSDB) entry, enter **TAS, 50.** Figure 19 is an example of this operation.

PS Files

6.03 Editing of PS files is accomplished in Generic 1 by using the edit command, **TAS, 20.** To use this feature, a paper reference copy of the PS file to be changed must be available. Enter **TAS, 20** and answer the questions appropriately. When all of the preliminary questions have been answered, the computer will output **STEP # ØØØ1** and wait for an attendant input. At that point, change the 2-letter function by entering in the new abbreviation, or jump to another step by using the slash (/) feature; example /28 will move the prompt immediately to Step 28. When the new step is reached, change the 2-letter function as described earlier. If the edited file is to remain the same length as the original, terminate the edit by moving to the final step (EN—END OF FILE) and enter **/E**. A given PS file can be shortened by changing one of the intermediate steps to an END OF FILE (using /E), or lengthened by changing the former END OF FILE to a different function and then continuing with additional functions ultimately finishing with an END (EN) and a /E.

6.04 After the PS file steps have been modified and the /E has been entered, the central will ask **DO YOU WANT TO KEEP THE OLD FILE (Y OR N)?**. The decision is the attendants; however, ***remember that the execution of sequence XYZ requires not only PF XYZ but also the PS file from which the PF XYZ was made.*** This implies that the act of editing a PS file, destroying the original, and reassigning the same name should be followed by deletion and possibly recreation of the PFs (if any) that were created using the original PS file.

> ***Caution: Whenever a PS file has been changed, its associated PF files must be edited to reflect those changes, or be purged.***

```
#TAS,3Ø
@
NAME OF THE FILE (MAX  6 CHAR'S)? SSH12

REQUEST COMPLETE
```

Fig. 18—Printout That Occurs When Requesting to Delete a File

```
#TAS,5Ø
@
STA. NO. (1-256)? 12
SCAN # (1-64) OR IF ALL ENTER A : 1
BIT # (1-256)? 7
REQUEST COMPLETE
```

Fig. 19—Printout That Occurs When Deleting a PSDB File Entry

PF Files

6.05 Editing the PF file is accomplished by using the edit command, TAS, 2Ø. To use this feature, a paper reference copy of the PF file to be changed must be available. Enter **TAS, 2Ø** and answer the questions appropriately. When all of the preliminary questions have been answered, the central will output **STEP# ØØØ1** followed by a question that is appropriate for the type of function called out in the specified PS file. At that point, change the answers as desired (see limitations in paragraph 6.06) or proceed to another step using the slash feature. When the new step is reached, the central will ask more questions appropriate to that particular function. Use the slash to proceed forward or backward; however, ultimately terminate the edit by arriving at the final step in the file (EN Step).

Generic 1 Only

6.06 Limitations do exist in the editing of PRINT and LOG steps. It is recommended that neither of these steps be edited. However, if they must, the new copy must maintain the identical number of characters as the old copy. Character counts must include a value of 2 extra for each line (carriage return and line feed). In determining character count, be careful to include all spaces, punctuation, and nonprinting characters. Figure 20 is an example of an edit made to a print step. The output device for this step was changed from L23 to L24.

Generic 2 Only

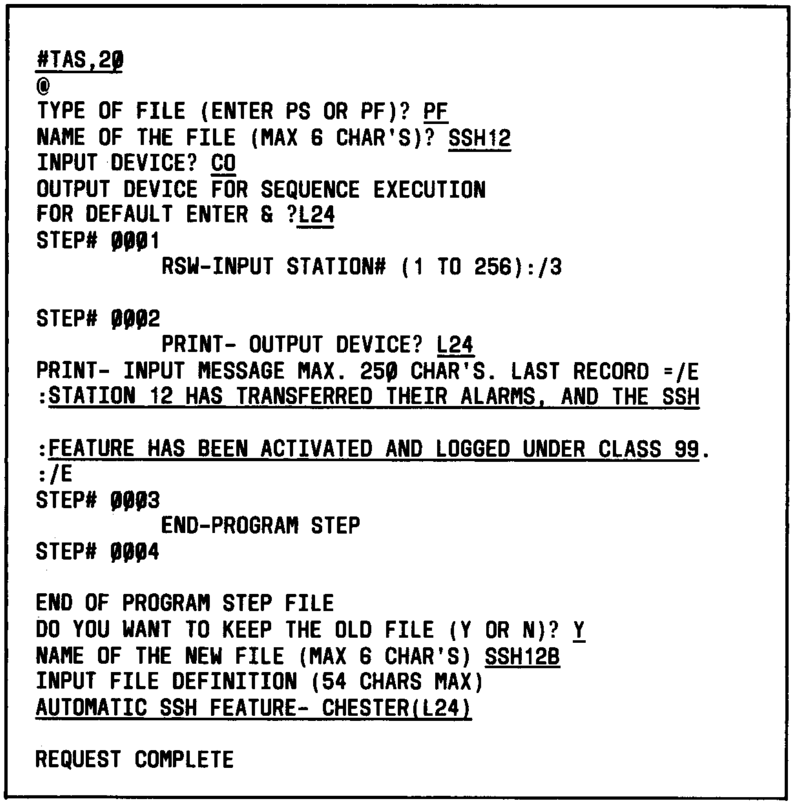
6.07 The PRINT and LOG steps can be edited and the length of the message changed. The maximum numbers is 248. The only restriction to character counting is that it must include a value of 2 extra for each line (carriage return and line feed).

7. ADVANCED TECHNIQUES

7.01 The next few paragraphs will be concerned with the use of the CALL, FLAG, INITIALIZE, and LOOP instructions. Also, the use of the incremental parameter will be discussed. There are several pecularities pertaining to these that deserve to be discussed in detail.

7.02 The CALL instruction is not exactly like the CALL instruction used in other systems' software. In the program sequencer, the CALL instruction causes the called job to be scheduled immediately. Both the calling sequence and the called sequence will then execute more or less concurrently. (See example in Part 8C—Status Summaries.)

> ***Note:*** Some jobs require that the called job is complete before the calling job continues. This requires the use of the flag instruction as described in paragraph 7.03 (b).



```
#TAS,2Ø
@
TYPE OF FILE (ENTER PS OR PF)? PF
NAME OF THE FILE (MAX 6 CHAR'S)? SSH12
INPUT DEVICE? CO
OUTPUT DEVICE FOR SEQUENCE EXECUTION
FOR DEFAULT ENTER & ?L24
STEP# ØØØ1
          RSW-INPUT STATION# (1 TO 256):/3

STEP# ØØØ2
          PRINT- OUTPUT DEVICE? L24
PRINT- INPUT MESSAGE MAX. 25Ø CHAR'S. LAST RECORD =/E
:STATION 12 HAS TRANSFERRED THEIR ALARMS, AND THE SSH

:FEATURE HAS BEEN ACTIVATED AND LOGGED UNDER CLASS 99.
:/E
STEP# ØØØ3
          END-PROGRAM STEP
STEP# ØØØ4

END OF PROGRAM STEP FILE
DO YOU WANT TO KEEP THE OLD FILE (Y OR N)? Y
NAME OF THE NEW FILE (MAX 6 CHAR'S) SSH12B
INPUT FILE DEFINITION (54 CHARS MAX)
AUTOMATIC SSH FEATURE- CHESTER(L24)

REQUEST COMPLETE
```

Fig. 20—Example of the Printout That Occurs When Making An Edit to a PF File

7.03 The FLAG step manipulates a specified flag value. This feature can be used in a myriad of ways. One example of its use is given in Part 8B—Fire Alarm. Other generalized examples are as follows:

(a) Flag can be used as a counter—incrementing by 1 on every execution. After 50 executions, a branch to a PR step can indicate the status of the job. Another FL step can return the counter to 0.

(b) A **CALL** step is used in JOB1—calling for another job, JOB2, that must complete before JOB1 continues. JOB1 may set a specified flag, call JOB2, wait 5 seconds, test the flag, go back to the **WAIT** step and remain in that loop until JOB2 changes the value of the flag to the new value expected by JOB1. JOB1 then tests again, finds the new value, and branches to another part of JOB1.

No flag management is provided by the PSEQ software. Therefore, it is recommended that a record be kept of all flag numbers and their use. This will prevent using the same flag number in two different and incompatible (as far as flag values are concerned) PSEQ jobs. Figure 21 is an example of a current flag number work sheet. Figure 22 is a blank flag number work sheet which can be reproduced locally.

7.04 Consider the INITIALIZE and LOOP instructions. (Refer to Part 8C, Status Summaries, of this section). The loop instruction provides the ability to perform a series of steps a number of times. The number of times a step is to be performed is entered into the parameter file as one less than the number of times the sequence is to be performed. If the series of steps to be repeated contains an **RS** or an **SC** step, the station numbers or the scan numbers can be incremented by 1 immediately after that particular step is executed. In this way, looping back to that step several times can allow a series of related events to be requested. To introduce the incrementing feature, the station or scan number must be entered with the letter I appended (station 2Ø would be 2ØI).

Note:

- This feature can only be used on station numbers or scan numbers of an RS or SC step.
- Station numbers are incremented to the next formatted station number in the STTB file which is not a C1I type remote.
- Scan numbers are incremented to the next formatted scan number.
- Use of this feature helps to conserve the file manager area.

7.05 The INITIALIZE instruction is used with the incremental parameter. The incremental parameter can be used with the station number or scan number as required in the REMOTE SWITCH and SCAN instructions. Its purpose is to increment the station number or scan number each time the instruction is executed. The incremental parameter is initially set to the original operator-defined value.

7.06 The purpose of the INITIALIZE instruction is to reset the incremental parameter to the original operator-defined values. This in effect sets the station numbers and scan numbers to their original values. The initialize instruction will also reset the "loop counter" to zero.

8. EXAMPLES

A. Stuck Sender Hold

8.01 The stuck sender hold (SSH) feature can be automatically activated via program sequencer. The PSEQ job which performs this function is explained in Part 3 (Generating a PSEQ Job) of this section.

B. Fire Alarm

8.02 This job is useful in assuring that a fire alarm is acted upon promptly (see flowchart FC-2 and the associated Fig. 23). When a fire alarm occurs, FIRE1 will automatically run. At a specified terminal, FIRE1 will ring a bell several times (to get the attendant's attention) and will cause the following message to be printed:

```
***FIRE**FIRE*** STA 11 HOLMDEL G1-B1 TO STOP
          MESSAGE TYPE: SEQ,OFF1
     0020 09:31 03/07/79 ST 011 HOLMDEL

     SGR 01 NO NEW ALARMS----OLD ALARMS----
     FIRE ALARM
```

This message will be repeated every 5 seconds until either the fire alarm bit clears or the attendant manually executes the PSEQ job called OFF1 (see Fig. 24 and flowchart FC-3). Automatic execution of FIRE1 is accomplished by linking the TASC System data base to the PSEQ data base as described in Part 4B (Automatic Execution) of this section.

C. Status Summaries

8.03 The following example (refer also to flowchart FC-4) consists of 4 different PSEQ jobs:

(1) Sum—obtains a status summary of all building alarms, and outputs them on LU #23 (#SEQ, SUM).

(2) ALTSUM—obtains a status summary of all switching alarms, and outputs them on LU #24 (#SEQ, ALTSUM).

(3) ARBSUM—obtains a status summary of all transmission alarms, and outputs them on LU #29 (#SEQ, ARBSUM).

(4) SUMCAL—does all the functions in the first three PSEQ jobs (#SEQ, SUMCAL).

8.04 The operation of SUM, ALTSUM, and ARBSUM are identical except for the station and loop values in the parameter file. For this reason, only PSEQ job SUM will be discussed. All three PF files (SUM, ALTSUM, and ARBSUM) share the PS file SUMPS. Refer to Fig. 25 for example printouts of the SUM, ALTSUM, and ARBSUM jobs with an explanation of the step numbers.

8.05 The SUMCAL, when executed, calls PSEQ jobs SUM, ALTSUM, and ARBSUM to execute. Thus, all three status summaries (building, switching, and transmission) can be obtained by initiating one PSEQ job—SUMCAL. Refer to Fig. 26 for an example listing of the PSEQ job—SUMCAL.

> ***Note:*** All three jobs (SUM, ALTSUM, and ARBSUM) will run simultaneously. This could cause the outputs to interweave if the outputs are all directed to the same LU device. To avoid this, use the FLAG instruction as explained in paragraph 7.03.

D. Make Busy A Trouble Recorder

8.06 The PSEQ job (see flowchart FC-5) will make the trouble recorder busy to all circuits requesting a trouble record card. It will also execute automatically when linked to the TBL RCDR MN bit in the SDR data base. Automatic execution is described in Part 4B of this section. Refer to Fig. 27 for an example of a PSEQ TRB57 job.

E. Restoration Plan

8.07 Chart 1 is a job description of a TD-3 restoration plan between freehold and podock. This procedure was used to generate the flowchart in FC-6. Refer to Fig. 28 and 29 for examples of PS and PF file listings.

•

9. ERROR MESSAGES

9.01 The following is a list of file manager error codes and their meanings which may occur when using program sequencer.

FMGR ERROR CODES

ERROR	MEANING
17	ILLEGAL READ OR WRITE ON A TYPE 0 FILE. REPORT BUG.
16	ILLEGAL FILE TYPE OR SIZE.
15	ILLEGAL FILE NAME TRY AGAIN.
14	THE DIRECTORY IS FULL. REPORT SYSTEM BUG.
13	CARTRIDGE LOCKED. TRY AGAIN.
12	START OR END OF FILE READ. UNEXPECTEDLY.
11	DATA CONTROL BLOCK NOT OPEN. REPORT PROGRAM BUG.
10	NOT ENOUGH PARAMETERS PASSED. REPORT PROGRAM BUG.
09	ATTEMPT TO OPEN TYPE 0 FILE AS TYPE 1 OR ERROR APOSN.
08	FILE IS CURRENTLY OPEN OR LOCKED. TRY AGAIN.
07	INVALID SECURITY CODE.
06	CARTRIDGE OR FILE NOT FOUND OR NO ROOM LEFT, TRY AGAIN.
05	ATTEMPT TO READ TO A POSITION OR RECORD NOT WRITTEN, BUG. •
04	THE FILE IS TOO LONG OR RECORD SIZE ERROR IS FOUND, BUG.
03	BACKSPACE NOT LEGAL FOR THIS TYPE OF FILE, BUG.
02	DUPLICATE FILE NAME, TRY ANOTHER NAME.
01	THE DISC IS DOWN, PLEASE UP THE DISC.

10. DELETE A PS OR PF FILE

10.01 Except as noted in paragraph 6.01, a PS or PF file can be deleted from the TASC software when it is no longer needed. This will, in turn, free those files assigned file manager storage area for other possible use. To delete a PS or PF file, perform the following activity. Initiate the **TAS** command by entering **TAS, 30** followed by a carriage return. Respond to the questions asked by the central. Figure 15 is a typical example.

10.02 After one or more PS or PF files have been deleted, pack the program sequencer disc storage area. This packing entails closing the storage area gaps between the PS and PF files. To pack the PSEQ disc storage area, initiate the **TAS** command. When the computer responds with **ENTER NUMBER:**, enter **90**. The packing operation will cause considerable disc activity for 1 to 5 minutes.

Note:

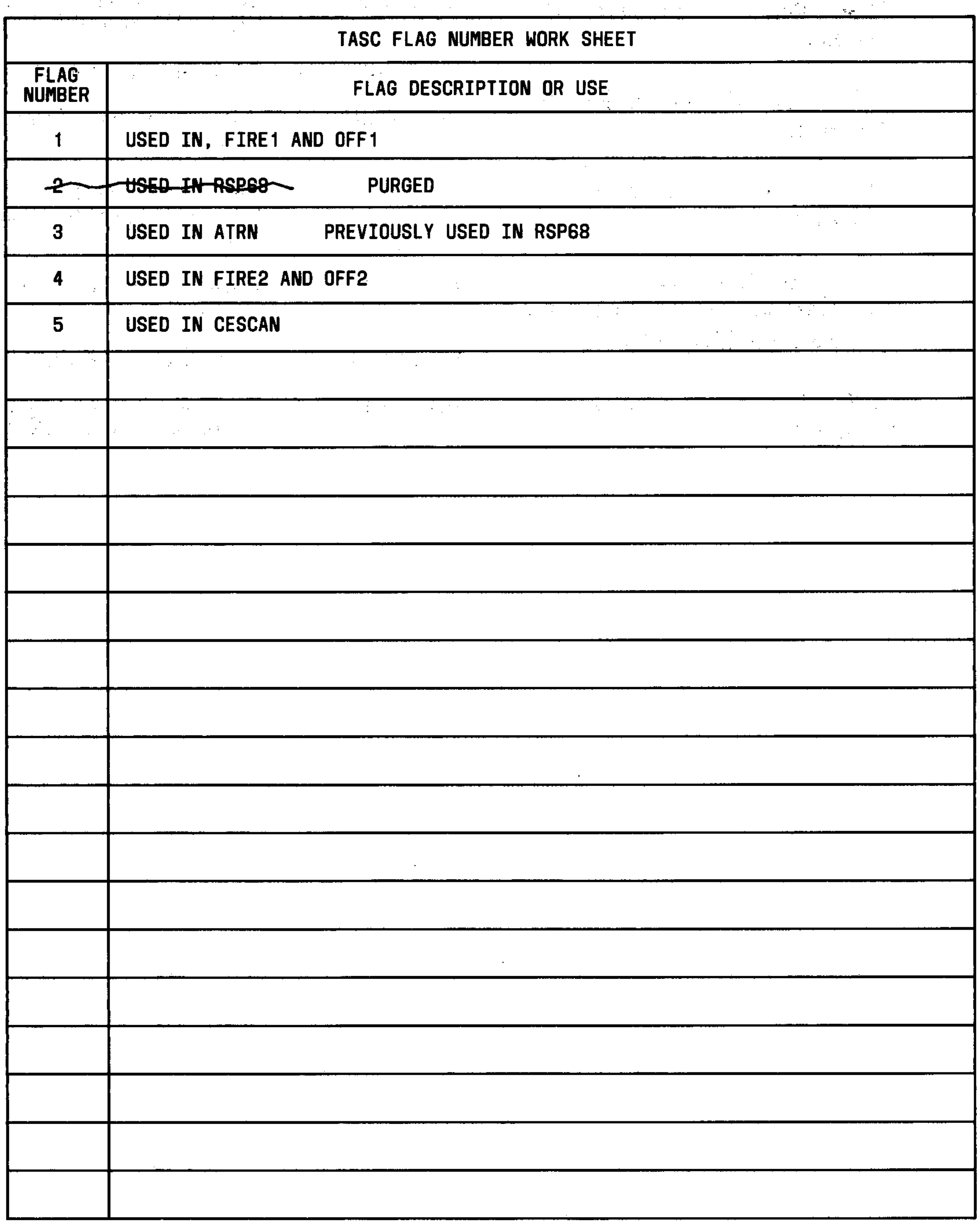
- The disc should never be packed while a sequence is running.
- Ideally, the disc should be packed at a time when there is relatively little system activity and it is close to half past the hour.
- The disc is automatically packed whenever the system is booted up.

11. REFERENCES

11.01 The following sections contain additional information which may be referred to as necessary.

SECTION	TITLE
190-210-101	TASC System Description
190-210-102	TASC Central Description for Generic 1

SECTION	TITLE
190-210-103	TASC Remote Equipment Description
190-210-112	TASC Central Description for Generic 2
190-210-300	TASC Central Operations for Generic 1
190-210-301	TASC Data Base Planning Using E2A SAC Remotes and CDO for Generic 1 Satellite Units
190-210-302	TASC Data Base Operations for Generic 1
190-210-304	TASC Data Base Planning Using C1, E1, E2, and E2A APR Remotes
190-210-305	TASC Application Programs
190-210-310	TASC Central Operations for Generic 2
190-210-311	TASC Data Base Planning Using E2A SAC Remotes and CDO for Generic 2
190-210-312	TASC Data Base Operations for Generic 2
190-210-500	TASC System Maintenance
865-200-120	Engineering and Implementation Methods System (EIMS) for TASC



TASC FLAG NUMBER WORK SHEET	
FLAG NUMBER	FLAG DESCRIPTION OR USE
1	USED IN, FIRE1 AND OFF1
~~2~~	~~USED IN RSP68~~ PURGED
3	USED IN ATRN PREVIOUSLY USED IN RSP68
4	USED IN FIRE2 AND OFF2
5	USED IN CESCAN

Fig. 21—Example of a Flag Number Work Sheet

TASC FLAG NUMBER WORK SHEET

FLAG NUMBER	FLAG DESCRIPTION OR USE

Fig. 22—Blank Flag Work Sheet

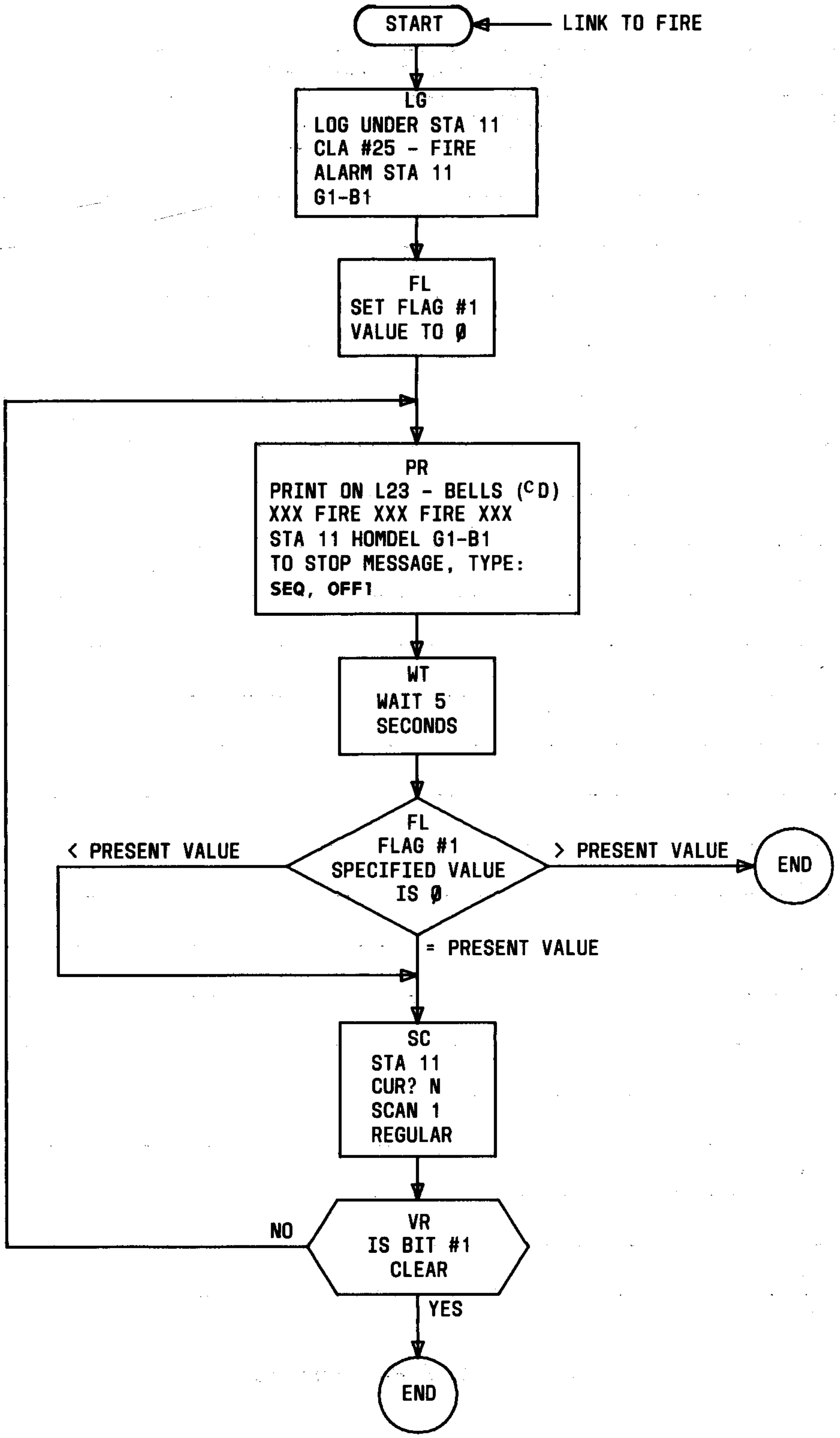


Fig. FC2—Flowchart 2— PSEQ Job FIRE1

EXPLANATION OF STEPS

STEP #1 Logs the fact that there was a fire alarm (all fire alarms could be logged under class #25).

STEP #2 Initialized (sets to Ø) flag #1. This is necesary because the value of flag #1 may or may not be zero when FIRE1 first runs. When flag #1 is greater than Ø, FIRE1 willl terminate via Step #5.

STEP #3 Rings the bell serveral times at the specified terminal, prints out the fact that there is a fire alarm, and gives instructions as to how FIRE1 can be terminated.

NOTE: The bells are rung serveral times at the terminal by entering several control Ds into the print step. Control D will not appear in a TAS,60 listing.

STEP #4 Provides a 5 second wait

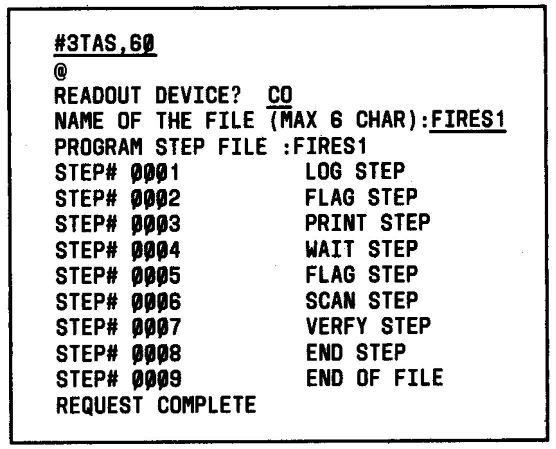
STEP #5 Tests flag #1. If the value of flag #1 is Ø, the sequence continues on to Step #6. However, if its value is greater than Ø, the sequence jumps to Step #8 and terminates. The PREQ job OFF1 sets flag #1 to 1. Therefore, running OFF1 will cause FIRE1 to terminate.

STEP #6 Scans station #11, group 1 and outputs the results on the specified output device (L23).

STEP #7 Uses the results of Step #6 to verify whether or not the fire alarm is still standing in. If it is, the sequence will jump to Step #3. If the alarm has retired, the sequence continues to Step #8 and terminates.

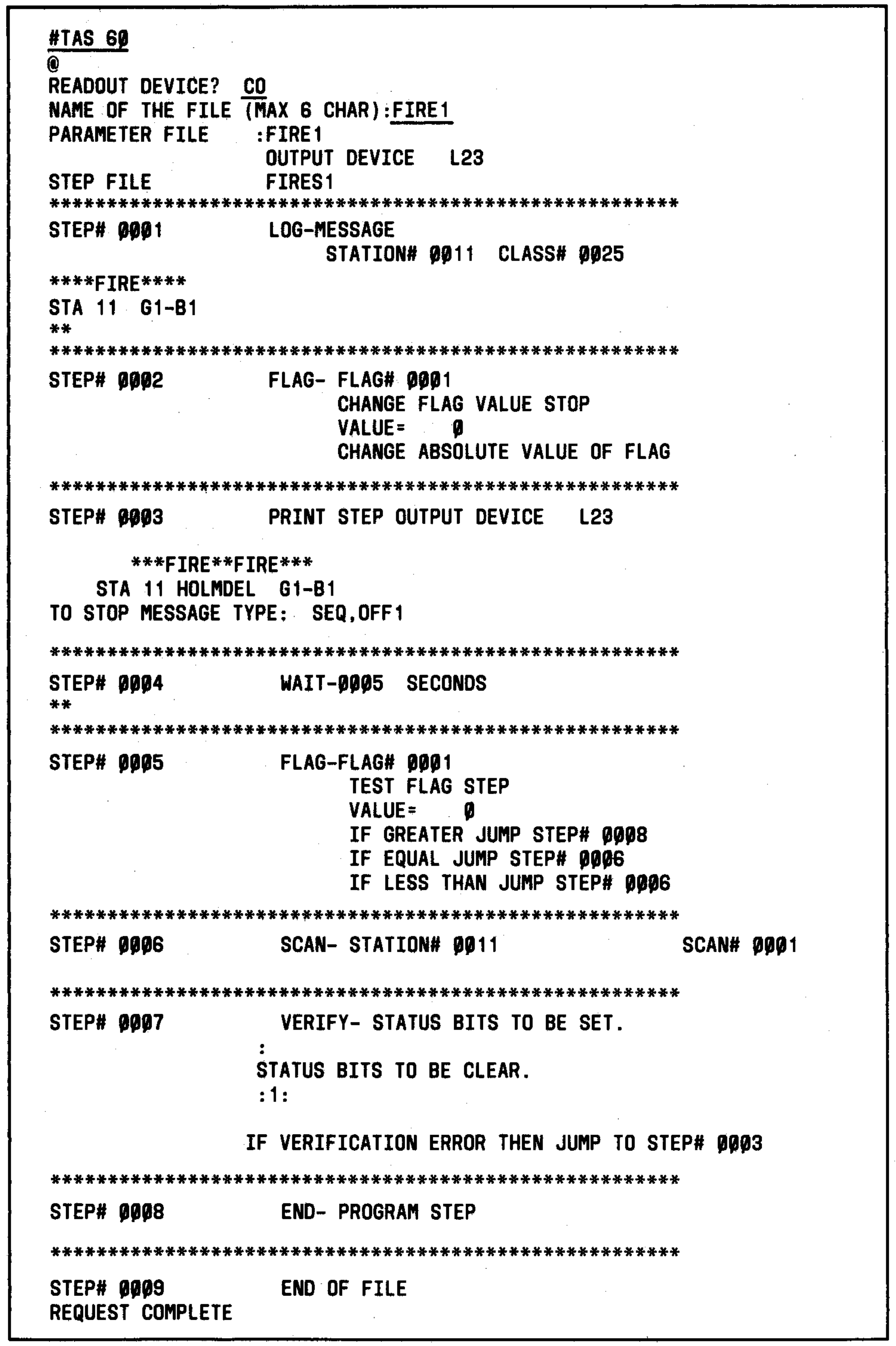
STEP #8 End

STEP #9 End of the file FIRE1



```
#3TAS,6Ø
@
READOUT DEVICE?  CO
NAME OF THE FILE (MAX 6 CHAR):FIRES1
PROGRAM STEP FILE :FIRES1
STEP# ØØØ1          LOG STEP
STEP# ØØØ2          FLAG STEP
STEP# ØØØ3          PRINT STEP
STEP# ØØØ4          WAIT STEP
STEP# ØØØ5          FLAG STEP
STEP# ØØØ6          SCAN STEP
STEP# ØØØ7          VERFY STEP
STEP# ØØØ8          END STEP
STEP# ØØØ9          END OF FILE
REQUEST COMPLETE
```

PRINTOUT OF THE PS FILE FIRE1



```
#TAS 6Ø
@
READOUT DEVICE?  CO
NAME OF THE FILE (MAX 6 CHAR):FIRE1
PARAMETER FILE    :FIRE1
                   OUTPUT DEVICE   L23
STEP FILE          FIRES1
*******************************************************
STEP# ØØØ1          LOG-MESSAGE
                         STATION# ØØ11  CLASS# ØØ25
****FIRE****
STA 11  G1-B1
**
*******************************************************
STEP# ØØØ2          FLAG- FLAG# ØØØ1
                          CHANGE FLAG VALUE STOP
                          VALUE=    Ø
                          CHANGE ABSOLUTE VALUE OF FLAG

*******************************************************
STEP# ØØØ3          PRINT STEP OUTPUT DEVICE   L23

       ***FIRE**FIRE***
    STA 11 HOLMDEL  G1-B1
TO STOP MESSAGE TYPE:  SEQ,OFF1

*******************************************************
STEP# ØØØ4           WAIT-ØØØ5  SECONDS
**
*******************************************************
STEP# ØØØ5           FLAG-FLAG# ØØØ1
                           TEST FLAG STEP
                           VALUE=    Ø
                           IF GREATER JUMP STEP# ØØØ8
                           IF EQUAL JUMP STEP# ØØØ6
                           IF LESS THAN JUMP STEP# ØØØ6
*******************************************************
STEP# ØØØ6           SCAN- STATION# ØØ11               SCAN# ØØØ1

*******************************************************
STEP# ØØØ7           VERIFY- STATUS BITS TO BE SET.
                   :
                   STATUS BITS TO BE CLEAR.
                   :1:

                  IF VERIFICATION ERROR THEN JUMP TO STEP# ØØØ3
*******************************************************
STEP# ØØØ8           END- PROGRAM STEP

*******************************************************
STEP# ØØØ9           END OF FILE
REQUEST COMPLETE
```

PRINTOUT OF THE PF FILE FIRE1

Fig. 23— Example of the Printout for PS and PF Files FIRE1

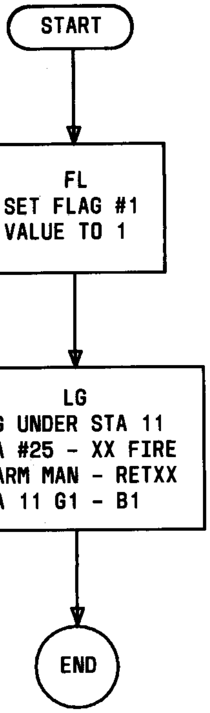


Fig. FC3—Flowchart 3— PSEQ Job OFF1

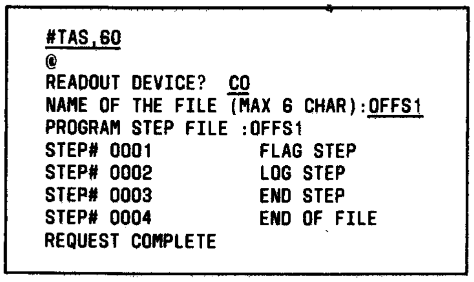
EXPLANATION OF STEPS

STEP #1 Sets the value of flag #1 to 1. This will cause FIRE1 to terminate.

STEP #2 Logs a message under station 11 class #25 that the fire alarm sequence is being manually terminated.

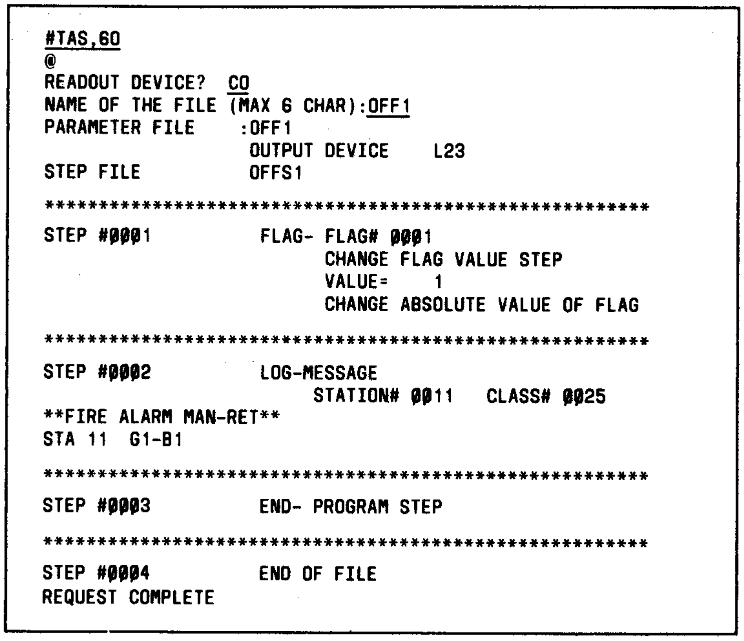
STEP #3 End

STEP #4 End of the file OFF1



```
#TAS,60
@
READOUT DEVICE?  CO
NAME OF THE FILE (MAX 6 CHAR):OFFS1
PROGRAM STEP FILE :OFFS1
STEP# 0001          FLAG STEP
STEP# 0002          LOG STEP
STEP# 0003          END STEP
STEP# 0004          END OF FILE
REQUEST COMPLETE
```

PRINTOUT OF THE PS FILE OFFS1



```
#TAS,60
@
READOUT DEVICE?  CO
NAME OF THE FILE (MAX 6 CHAR):OFF1
PARAMETER FILE     :OFF1
                   OUTPUT DEVICE    L23
STEP FILE          OFFS1
*********************************************************
STEP #ØØØ1          FLAG- FLAG# ØØØ1
                          CHANGE FLAG VALUE STEP
                          VALUE=    1
                          CHANGE ABSOLUTE VALUE OF FLAG
*********************************************************
STEP #ØØØ2          LOG-MESSAGE
                         STATION# ØØ11   CLASS# ØØ25
**FIRE ALARM MAN-RET**
STA 11  G1-B1
*********************************************************
STEP #ØØØ3          END- PROGRAM STEP
*********************************************************
STEP #ØØØ4          END OF FILE
REQUEST COMPLETE
```

PRINTOUT OF THE PF FILE OFF1

Fig. 24—Example of the Printout for PS and PF Files OFF1

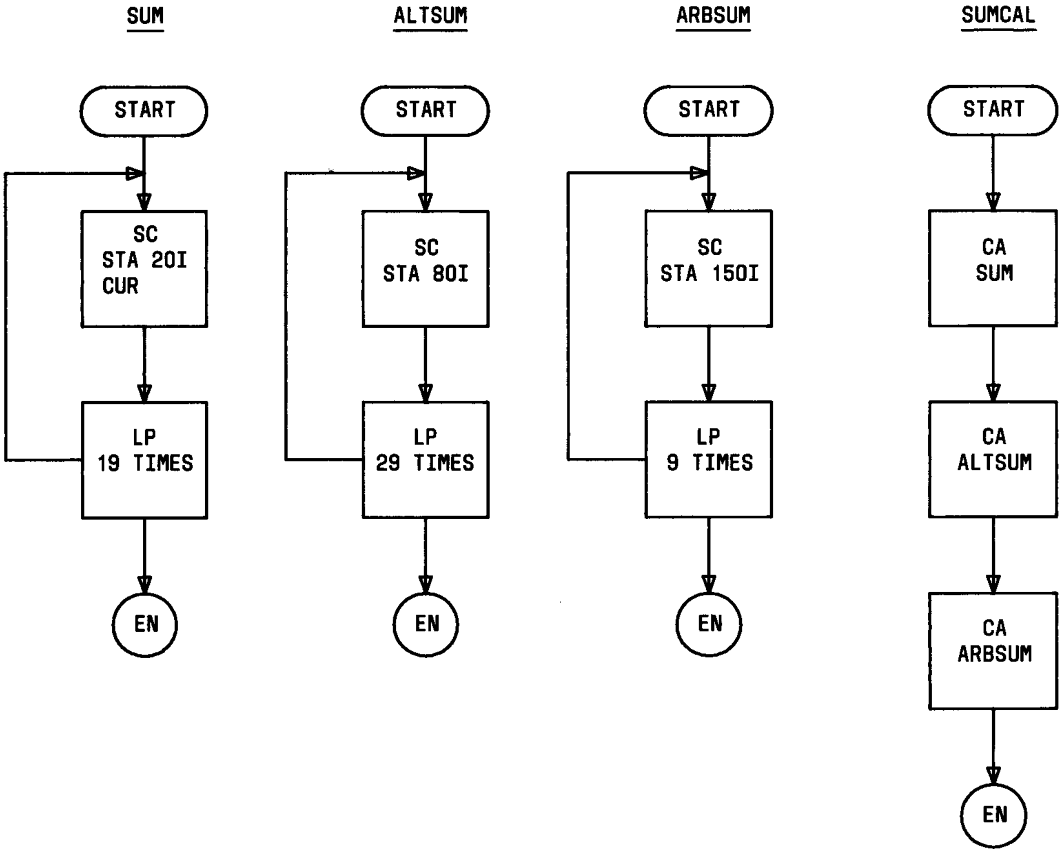


Fig. FC4—Flowchart 4— Flowcharts for PSEQ Jobs SUM, ALTSUM, ARBSUM, and SUMCAL

EXPLANATION OF STEPS

PSEQ JOB SUM

STEP #1 Will do a CUR operation on station 20 with increment.

STEP #2 Will loop back to STEP #1 nineteen times. A CUR operation will now have been performed on stations 20 through 39.

STEP #3 End

STEP #4 End of the file SUM

```
#TAS,60
@
READOUT DEVICE?  CO
NAME OF THE FILE (MAX 6 CHAR):SUMPS
PROGRAM STEP FILE :SUMPS
STEP# 0001          SCAN STEP
STEP# 0002          LOOP STEP
STEP# 0003          END STEP
STEP# 0004          END OF FILE
REQUEST COMPLETE
```

LISTING OF THE PS FILE SUMPS

```
#TAS,60
@
READOUT DEVICE?  CO
NAME OF THE FILE (MAX 6 CHAR):SUM
PARAMETER FILE     :SUM
                    OUTPUT DEVICE     L23
STEP FILE           SUMPS
*********************************************************
STEP #0001          SCAN- STATION# 0020  INCREMENT         CUR OPERATION
*********************************************************
STEP #0002          LOOP- STEP# 0001  0019-TIMES
*********************************************************
STEP #0003          END- PROGRAM STEP
*********************************************************
STEP #0004          END OF FILE
REQUEST COMPLETE
```

LISTING OF THE PF FILE SUM

```
#TAS,60
@
READOUT DEVICE?  CO
NAME OF THE FILE (MAX 6 CHAR):ALTSUM
PARAMETER FILE     :ALTSUM
                    OUTPUT DEVICE     L24
STEP FILE           SUMPS
*********************************************************
STEP #0001          SCAN- STATION# 0080  INCREMENT              CUR OPERATION
*********************************************************
STEP #0002          LOOP- STEP# 0001  0029-TIMES
*********************************************************
STEP #0003          END- PROGRAM STEP
*********************************************************
STEP #0004          END OF FILE
REQUEST COMPLETE
```

LISTING OF THE PF FILE ALTSUM

```
#TAS,60

READOUT DEVICE?  CO
NAME OF THE FILE (MAX 6 CHAR):ARBSUM
PARAMETER FILE     :ARBSUM
                    OUTPUT DEVICE     L29
STEP FILE           SUMPS
*********************************************************
STEP #0001          SCAN- STATION# 150  INCREMENT               CUR OPERATION
*********************************************************
STEP #0002          LOOP- STEP# 0001  0009-TIMES
*********************************************************
STEP #0003          END- PROGRAM STEP
*********************************************************
STEP #0004          END OF FILE
REQUEST COMPLETE
```

LISTING OF THE PF FILE ARBSUM

Fig. 25—Example of the Listings of the PS and PF Files (SUMPS, SUM, ALTSUM, and ARBSUM) With Step Explanation

EXPLANATION OF STEPS

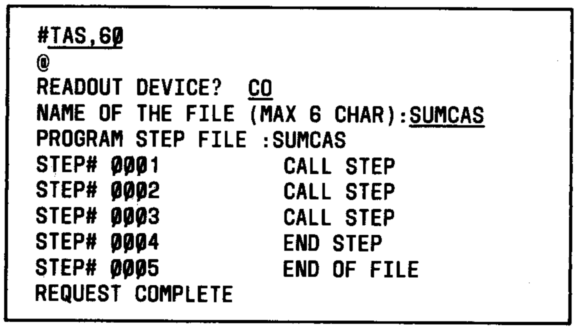
STEP #1 call PSEQ job SUM to execute

STEP #2 call PSEQ job ALTSUM to execute

STEP #3 call PSEQ job ARBSUM to execute

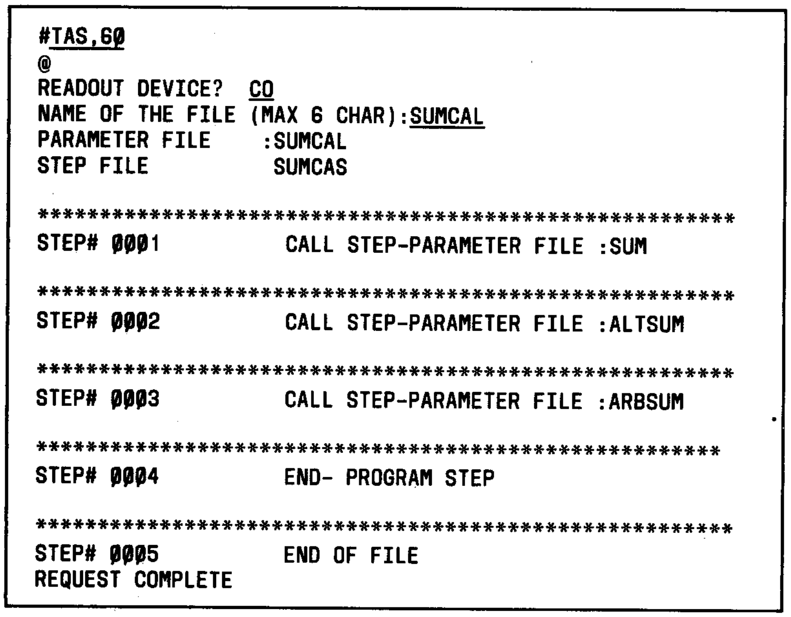
STEP #4 End

STEP #5 end of the file SUMCAL



```
#TAS,6Ø
@
READOUT DEVICE?  CO
NAME OF THE FILE (MAX 6 CHAR):SUMCAS
PROGRAM STEP FILE :SUMCAS
STEP# ØØØ1         CALL STEP
STEP# ØØØ2         CALL STEP
STEP# ØØØ3         CALL STEP
STEP# ØØØ4         END STEP
STEP# ØØØ5         END OF FILE
REQUEST COMPLETE
```

LISTING OF THE PS FILE SUMCAS



```
#TAS,6Ø
@
READOUT DEVICE?  CO
NAME OF THE FILE (MAX 6 CHAR):SUMCAL
PARAMETER FILE    :SUMCAL
STEP FILE          SUMCAS

********************************************************
STEP# ØØØ1          CALL STEP-PARAMETER FILE :SUM

********************************************************
STEP# ØØØ2          CALL STEP-PARAMETER FILE :ALTSUM

********************************************************
STEP# ØØØ3          CALL STEP-PARAMETER FILE :ARBSUM

*******************************************************
STEP# ØØØ4          END- PROGRAM STEP

********************************************************
STEP# ØØØ5          END OF FILE
REQUEST COMPLETE
```

LISTING OF THE PF FILE SUMCAL

Fig. 26—Example of the PSEQ Job—SUMCAL

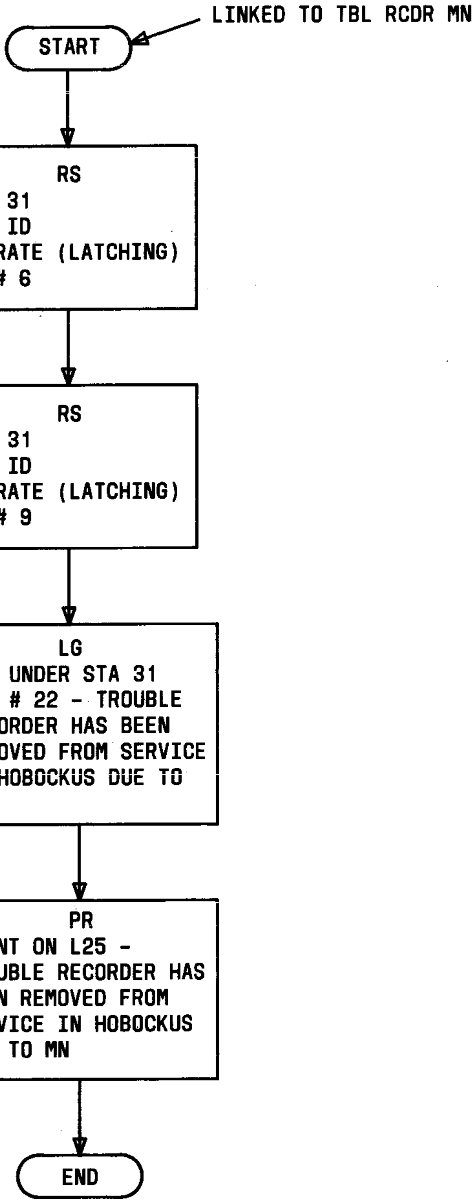


Fig. FC5—Flowchart 5—Flowchart for the Trouble Recorder Make Busy Sequence

EXPLANATION OF STEPS

STEP #1 operates (latching) the trouble recorder made busy (TRMB) switches at station #31.

STEP #2 operates (latching) the control active (CTL) switch at station #31

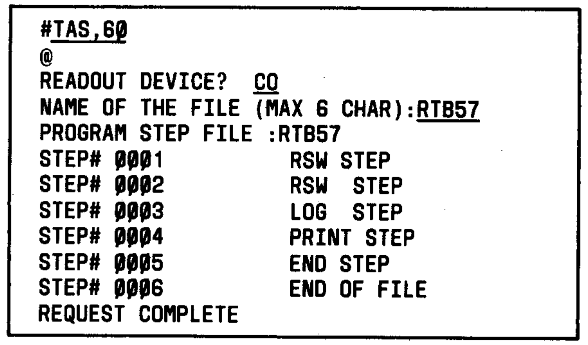
STEP #3 logs the fact that the trouble recorder has been removed from service

STEP #4 prints out a message indicating that the trouble recorder has been removed from service

STEP #5 end

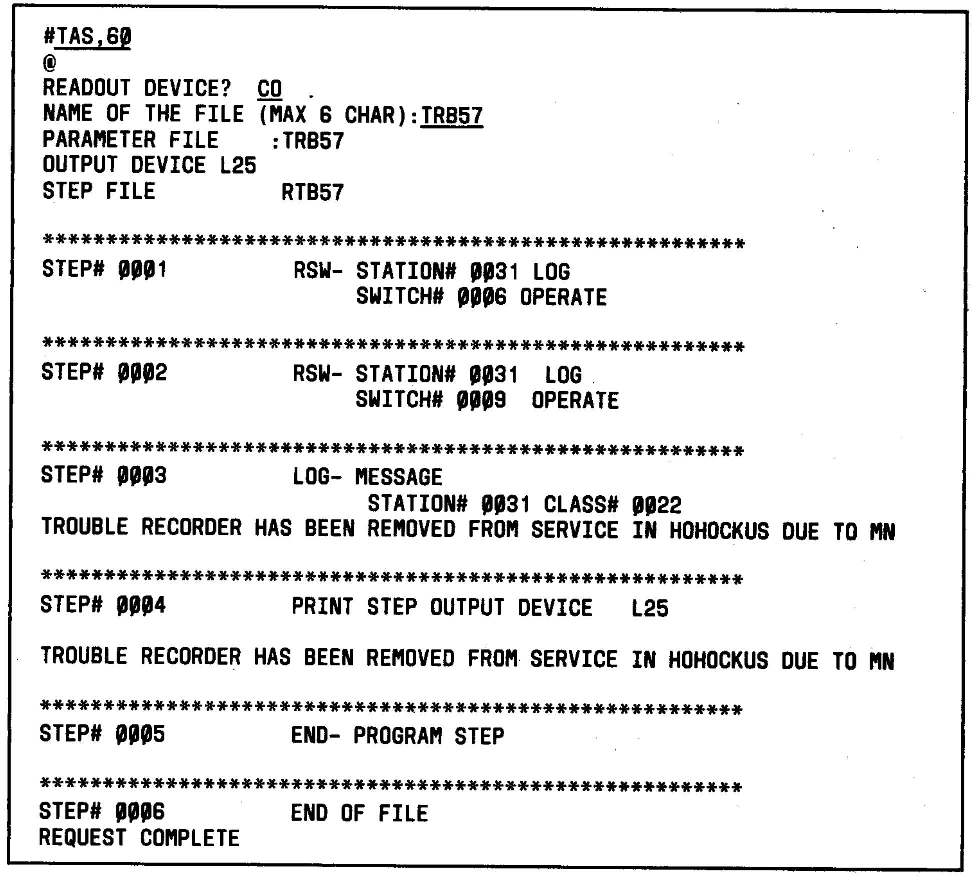
STEP #6 end of the file TRB57

NOTE:
The log and print steps come after the RS step. This prevents the logging or printing of these messges if a telemetry error occurs on either of the RS steps.



```
#TAS,6Ø
@
READOUT DEVICE?  CO
NAME OF THE FILE (MAX 6 CHAR):RTB57
PROGRAM STEP FILE :RTB57
STEP# ØØØ1          RSW STEP
STEP# ØØØ2          RSW  STEP
STEP# ØØØ3          LOG  STEP
STEP# ØØØ4          PRINT STEP
STEP# ØØØ5          END STEP
STEP# ØØØ6          END OF FILE
REQUEST COMPLETE
```

LISTING OF THE PS FILE RTB57



```
#TAS,6Ø
@
READOUT DEVICE?  CO
NAME OF THE FILE (MAX 6 CHAR):TRB57
PARAMETER FILE    :TRB57
OUTPUT DEVICE L25
STEP FILE          RTB57

********************************************************
STEP# ØØØ1          RSW- STATION# ØØ31 LOG
                         SWITCH# ØØØ6 OPERATE

********************************************************
STEP# ØØØ2          RSW- STATION# ØØ31  LOG
                         SWITCH# ØØØ9  OPERATE

********************************************************
STEP# ØØØ3          LOG- MESSAGE
                          STATION# ØØ31 CLASS# ØØ22
TROUBLE RECORDER HAS BEEN REMOVED FROM SERVICE IN HOHOCKUS DUE TO MN

********************************************************
STEP# ØØØ4          PRINT STEP OUTPUT DEVICE   L25

TROUBLE RECORDER HAS BEEN REMOVED FROM SERVICE IN HOHOCKUS DUE TO MN

********************************************************
STEP# ØØØ5          END- PROGRAM STEP

********************************************************
STEP# ØØØ6          END OF FILE
REQUEST COMPLETE
```

LISTING OF THE PF FILE TRB57

Fig. 27—Example of PSEQ Job TRB57 With an Explanation of the Steps

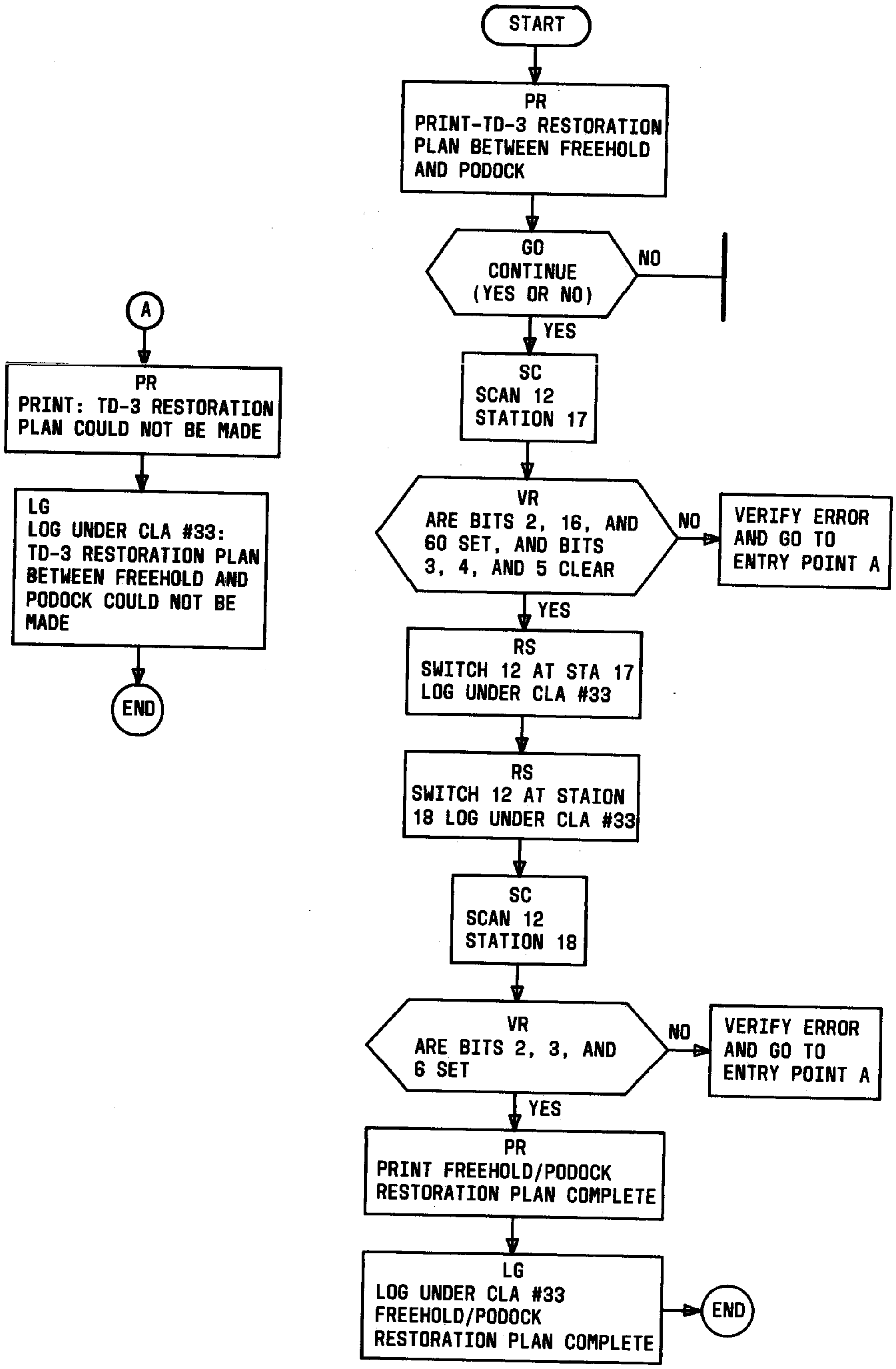
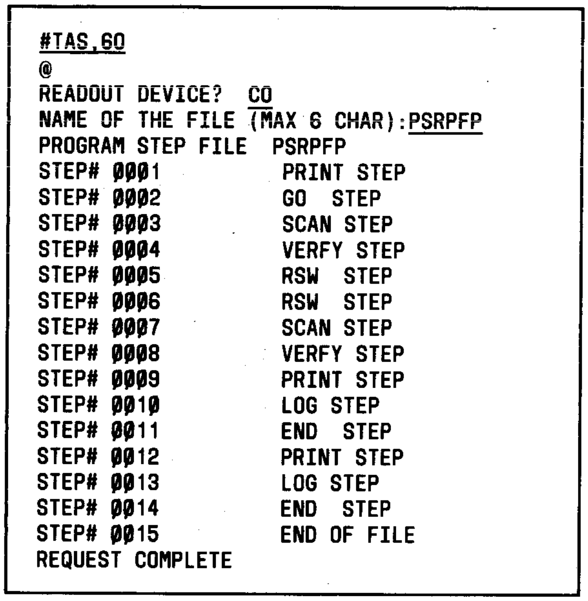
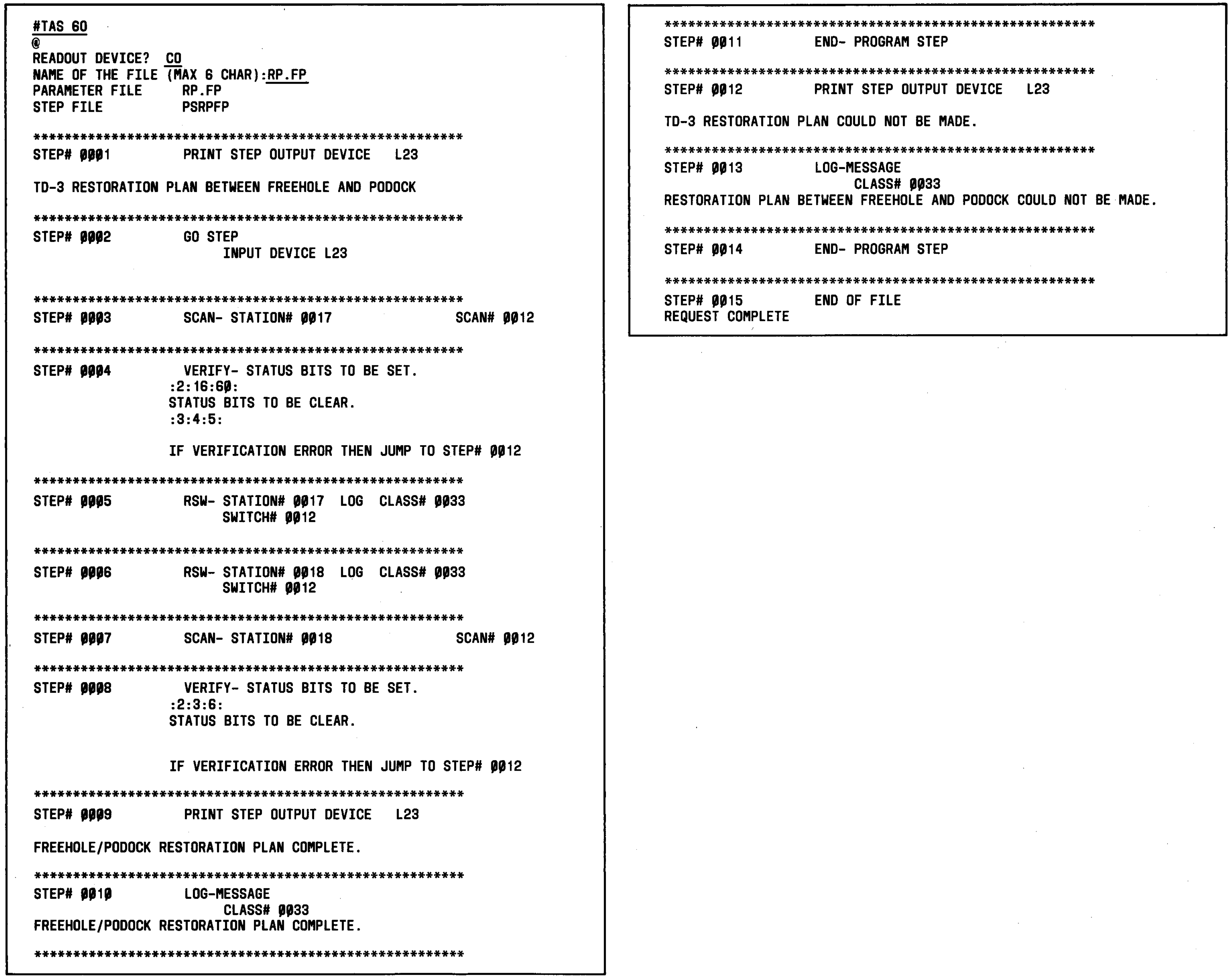

START
PR
PRINT-TD-3 RESTORATION PLAN BETWEEN FREEHOLD AND PODOCK
GO
CONTINUE (YES OR NO)
NO
YES
SC
SCAN 12 STATION 17
VR
ARE BITS 2, 16, AND 60 SET, AND BITS 3, 4, AND 5 CLEAR
NO
VERIFY ERROR AND GO TO ENTRY POINT A
YES
RS
SWITCH 12 AT STA 17 LOG UNDER CLA #33
RS
SWITCH 12 AT STAION 18 LOG UNDER CLA #33
SC
SCAN 12 STATION 18
VR
ARE BITS 2, 3, AND 6 SET
NO
VERIFY ERROR AND GO TO ENTRY POINT A
YES
PR
PRINT FREEHOLD/PODOCK RESTORATION PLAN COMPLETE
LG
LOG UNDER CLA #33 FREEHOLD/PODOCK RESTORATION PLAN COMPLETE
END
A
PR
PRINT: TD-3 RESTORATION PLAN COULD NOT BE MADE
LG
LOG UNDER CLA #33: TD-3 RESTORATION PLAN BETWEEN FREEHOLD AND PODOCK COULD NOT BE MADE
END

Fig. FC6— Flowchart 6— Flowchart for a TD-3 Restoration Plan Between Freehold and Podock



```
#TAS,60
@
READOUT DEVICE?  CO
NAME OF THE FILE (MAX 6 CHAR):PSRPFP
PROGRAM STEP FILE  PSRPFP
STEP# ØØØ1          PRINT STEP
STEP# ØØØ2          GO  STEP
STEP# ØØØ3          SCAN STEP
STEP# ØØØ4          VERFY STEP
STEP# ØØØ5          RSW  STEP
STEP# ØØØ6          RSW  STEP
STEP# ØØØ7          SCAN STEP
STEP# ØØØ8          VERFY STEP
STEP# ØØØ9          PRINT STEP
STEP# ØØ1Ø          LOG STEP
STEP# ØØ11          END  STEP
STEP# ØØ12          PRINT STEP
STEP# ØØ13          LOG STEP
STEP# ØØ14          END  STEP
STEP# ØØ15          END OF FILE
REQUEST COMPLETE
```

Fig. 28—Listing of the PS File PSR PFR-RSP TD-3 Freehold-Podock



```
#TAS 60
@
READOUT DEVICE?  CO
NAME OF THE FILE (MAX 6 CHAR):RP.FP
PARAMETER FILE     RP.FP
STEP FILE          PSRPFP

*******************************************************
STEP# ØØØ1         PRINT STEP OUTPUT DEVICE   L23

TD-3 RESTORATION PLAN BETWEEN FREEHOLE AND PODOCK

*******************************************************
STEP# ØØØ2         GO STEP
                        INPUT DEVICE L23


*******************************************************
STEP# ØØØ3         SCAN- STATION# ØØ17                  SCAN# ØØ12

*******************************************************
STEP# ØØØ4         VERIFY- STATUS BITS TO BE SET.
                 :2:16:6Ø:
                 STATUS BITS TO BE CLEAR.
                 :3:4:5:

                 IF VERIFICATION ERROR THEN JUMP TO STEP# ØØ12

*******************************************************
STEP# ØØØ5         RSW- STATION# ØØ17  LOG  CLASS# ØØ33
                        SWITCH# ØØ12

*******************************************************
STEP# ØØØ6         RSW- STATION# ØØ18  LOG  CLASS# ØØ33
                        SWITCH# ØØ12

*******************************************************
STEP# ØØØ7         SCAN- STATION# ØØ18                  SCAN# ØØ12

*******************************************************
STEP# ØØØ8         VERIFY- STATUS BITS TO BE SET.
                 :2:3:6:
                 STATUS BITS TO BE CLEAR.


                 IF VERIFICATION ERROR THEN JUMP TO STEP# ØØ12

*******************************************************
STEP# ØØØ9         PRINT STEP OUTPUT DEVICE   L23

FREEHOLE/PODOCK RESTORATION PLAN COMPLETE.

*******************************************************
STEP# ØØ1Ø         LOG-MESSAGE
                        CLASS# ØØ33
FREEHOLE/PODOCK RESTORATION PLAN COMPLETE.

*******************************************************
```

```
*******************************************************
STEP# ØØ11         END- PROGRAM STEP

*******************************************************
STEP# ØØ12         PRINT STEP OUTPUT DEVICE   L23

TD-3 RESTORATION PLAN COULD NOT BE MADE.

*******************************************************
STEP# ØØ13         LOG-MESSAGE
                        CLASS# ØØ33
RESTORATION PLAN BETWEEN FREEHOLE AND PODOCK COULD NOT BE MADE.

*******************************************************
STEP# ØØ14         END- PROGRAM STEP

*******************************************************
STEP# ØØ15         END OF FILE
REQUEST COMPLETE
```

Fig. 29—Listing of the PF File RP-FP RSP TD-3 Freehold-Podock

CHART 1

JOB DESCRIPTION OF A
TD-3 RESTORATION PLAN
BETWEEN FREEHOLD AND PODOCK

APPARATUS:

None

STEP	PROCEDURE
1	Print message describing this PSEQ job.
2	Allow for attendant to authorize this job—GO instruction.
3	At station 17, scan display 12.
4	If bits 2, 16, and 60 are not set and bits 3, 4, and 5 are clear, go to Step 5.
5	Operate switch 12 at F2 stations 17 and 18, and log this number under class class number 33.
6	Operate switch 12 at E2 station 18, and log this number under class number 33.
7	Scan display 12 at station 18.
8	If bits 2, 3, and 6 are clear, move to Step 11. If bits 2, 3, and 6 are set, go to Step 8.
9	Print message that restoration is complete.
10	Log similar message under class number 33.
11	End—Program step.
12	Print message that restoration cannot be made.
13	Log similar message under class number 33.
14	End—Program step.