MATHESON

NOAA Technical Memorandum ERL SEL-67

HIGH FREQUENCY RADAR SOFTWARE
REFERENCE MANUAL FOR PRODUCT ONE

Space Environment Laboratory
Boulder, Colorado
February 1984

noaa NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION / Environmental Research Laboratories

NOAA Technical Memorandum ERL SEL-67

HIGH FREQUENCY RADAR SOFTWARE REFERENCE MANUAL FOR PRODUCT ONE

David C. Walden
James R. Winkelman
Lorne D. Matheson
Larry D. Schultz
R. Gregg Merrill, Editor

Space Environment Laboratory
Boulder, Colorado
February 1984

UNITED STATES DEPARTMENT OF COMMERCE

Malcolm Baldrige, Secretary

NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION

John V. Byrne, Administrator

Environmental Research Laboratories

Vernon E. Derr Director

NOTICE

Mention of a commercial company or product does not constitute an endorsement by NOAA Environmental Research Laboratories. Use for publicity or advertising purposes of information from this publication concerning proprietary products or the tests of such products is not authorized.

HIGH FREQUENCY RADAR SOFTWARE REFERENCE MANUAL

FOR PRODUCT ONE

PREFACE

This manual for use with the HF Radar Product 1 software was first distributed to users in draft form in 1978. It has been updated several times since then as the software has evolved. It is being formally published now to record the achievement and to form a more convenient reference manual for the use of this product.

CONTENTS

CONTENTS

CONTENTS

1. INTRODUCTION

Larry D. Schultz
David C. Walden

1.1. The Radar [L.D.S.]

The Space Environment Laboratory (SEL) of NOAA has developed a high frequency radar for probing the bottom side of the ionosphere to obtain data from which various ionospheric parameters may be deduced. This advanced ionospheric sounding instrument incorporates digital control, recording, and analysis techniques. It consists of a radar transmitter-receiver unit and a computer-controlled data acquisition system.

1.2. The Hardware [L.D.S.]

The radar hardware contains a high power (10kW peak) pulse transmitter and a pair of receiver channels. Each receiver channel consists of an antenna multiplexer, filters, mixers and IF amplifiers, quadrature detectors and analog filters, and an analog-to-digital converter. A 64-channel analog-to-digital multiplexer monitors various functions (temperatures, voltages, forward and reflected pulse powers, etc.). Two microprocessor units (discussed below) are responsible for the operation and timing of the radar. Microprocessors are more flexible and simpler to design than pure hardware, and run faster than software alone to perform the same functions.

The system computer that controls the overall operation is an Interdata 7/16 or 8/16 minicomputer with the high-speed Arithmetic Logic Unit option and floating point hardware. Its memory consists of 64K directly-addressable bytes (8 bits/byte). It has 16 general purpose registers, 15 of which can be used for indexing. Its instruction set is similar to that of an IBM 360. The memory cycle time is one microsecond for the 7/16 and 750 nanoseconds for the 8/16.

The timing and control of the transmitter-receiver unit is performed by a microprocessor system called the Timing Sequence Generator (TSG). The TSG controls the transmission of a sequence of pulses, initiates data acquisition, and maintains timing for the entire instrument. The TSG performs these functions by executing programs in its memory that are selected by the system computer. The timing and sequencing control functions reside in the TSG because it can operate in parallel with and independently of the system computer.

1.2. The Hardware

The preliminary data processing is done in a second microprocessor called the Front End Processor (FEP). The FEP receives data from the receiving system and filters them. passing a subset of the data to the system computer. The FEP, like the TSG, is also a microcomputer system that runs in parallel with the system computer. The system computer loads programs into the FEP and the TSG selects the FEP program and the FEP input memory. The transfer of data from the FEP into the Interdata computer is done via a Direct Memory Access (DMA) device called EKO (radar Echo data Output). The computer operating system handles the TSG and FEP as peripheral devices attached to the DIO bus. This bus is under Interdata computer control and is used for medium speed, general input and output of 8-bit bytes to other parts of the system. A total of 64 instruments can be addressed on the DIO bus. The EKO device is attached to the multiplexor I/O bus in parallel with the DIO bus.

Several standard peripherals serve the radar: a system console with a graphics Cathode Ray Tube (CRT) display and a full ASCII keyboard; a dual-platter, moving-head disc with a total capacity of 10 megabytes; a Model 33 teletype for system logging, diagnostics, and system boot-up; and a refresh CRT, or system display, used to display data under computer control by GRAPHER.

1.3. The Operating System [L.D.S.]

The overall operation of the computer software is under the control of an Operating System (OS) called Multi-Tasking 2 (MT2), which was purchased along with the computer from the Interdata Corporation. The OS allows program modules to be coded in the form of foreground or background tasks. There can be up to 128 foreground tasks and one background task operating under the control of the OS.

The OS consists of three major modules called Executive, File Manager, and Command Processor. All modules can be overlayed; that is, portions of them can reside on the disc until they are needed.. When the OS needs the disc-resident portions of the File Manager or the Command Processor, it reads them into memory for execution. A non-overlayed version of MT2 occupies about 2/3 of the system computer memory. However, a selectively overlayed version of MT2 requires only about 1/3 of available memory; unfortunately, it runs slower than the non-overlayed system.

There are many options available to the user to define the specific version of MT2 that is desired. The options are specified at system generation time, and they normally cannot be changed at program execution time. Depending upon the amount of documentation required, system generation time will vary from about 30 minutes to about 5 hours of computer console time. The OS is described in three Interdata Corporation manuals that the reader should consult (see OS1, OS2, and OS3 in the

bibliography). Alterations to the OS are described in Chapter 11 of this manual.

1.4. Product One [D.C.W.]

The hardware-software system called Product One provides soundings that scan frequencies to produce an ionogram (I-mode) and repeated soundings on each of a set of frequencies in the kinesonde mode (K-mode).

The data returned by a sounding may be displayed in real time in I- or K-mode as appropriate, or in the skymap mode (S-mode) that displays the direction of arrival for either sounding mode. The displays differentiate between ordinary and extra-ordinary magneto-ionoic components in various combinations selected by the user.

The returned data stored on the disc may be archived on magnetic tape. Data on disc or tape may be recalled and displayed in any appropriate mode.

1.5. Product One tasks [L.D.S. & D.C.W.]

There are six foreground tasks and an overlayed version of MT2 in Product One. The tasks are COUNSEL, SOUNDER, PICKER, MANAGER, GRAPHER, and ANALYSER. The overlayed version of MT2 used in Product One takes about 1/3 of the computer memory, leaving about 1/3 of memory for data buffers, and about 1/3 for HFRADAR tasks. The overlayed portions of MT2 are subsets of File Manager and Command Processor. These overlayed subsets do not slow down the data acquisition and inter-task communication functions of the radar software system, but they do slow down other less critical functions of Product One.

In Product One each peripheral device is associated with only one task during sounding. The transmitter and receivers, the housekeeping ADC, the DIO interface, and the TSG, FEP, and EKO are associated only with SOUNDER; MANAGER is concerned with requests to and from the disc and tape units; the refresh CRT is associated only with GRAPHER; and the system console is associated with COUNSEL. There are no peripherals connected to either PICKER or ANALYSER.

Figure 1.1 is a block diagram of Product One that shows the various peripherals attached to their appropriate tasks. In addition, inter-task communication queueing (see Section 1.7) is shown as either one- or two-way arrows. Data from the receivers enter the input memories of the FEP every 20 milliseconds. After FEP processing, the Interdata transfers the data from the output memories of the FEP into its data buffers via device EKO and a direct memory access device called a SELector CHannel (SELCH).

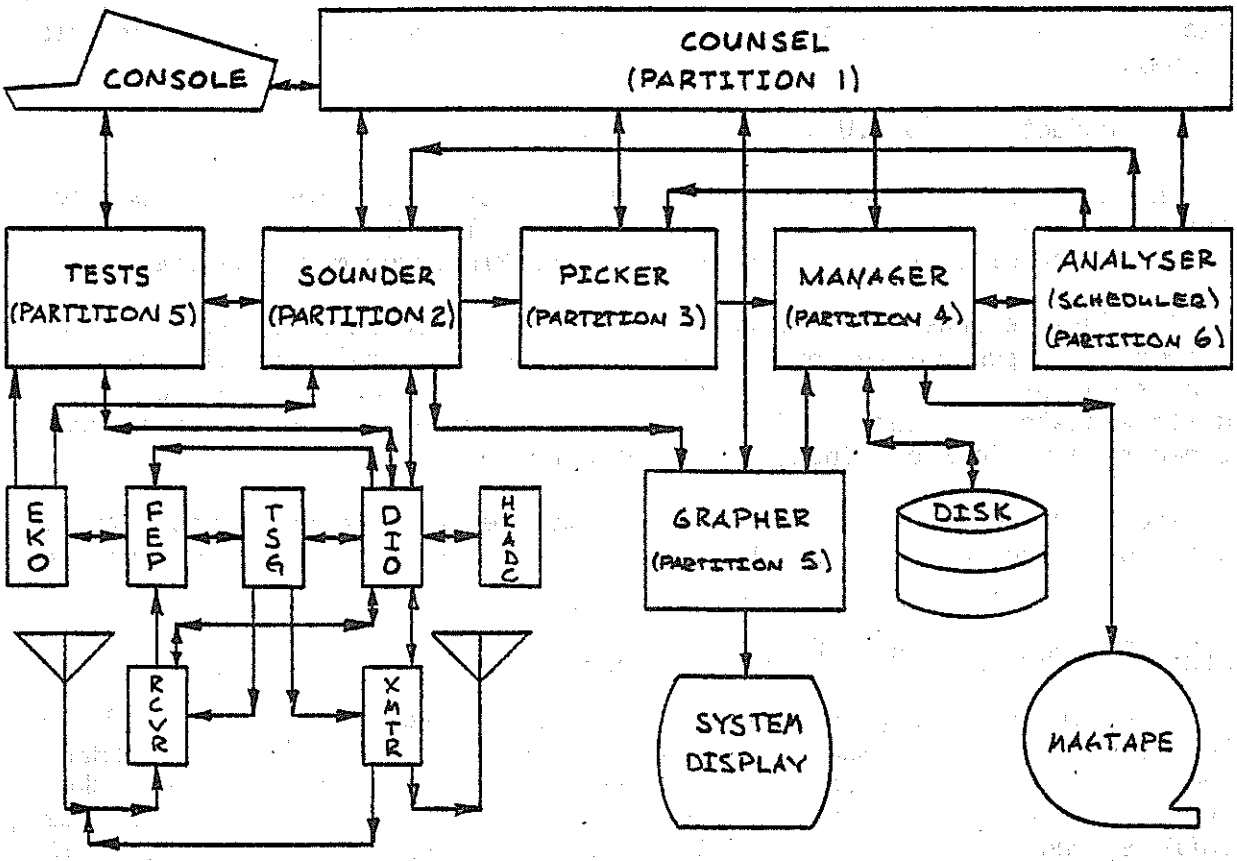


Figure 1.1. PRODUCT ONE BLOCK DIAGRAM

1.6. Task interrelations [L.D.S. & D.C.W.]

SOUNDER — The SOUNDER task controls the transmitter-receiver instruments (frequency synthesizer, RF and IF attenuators, bandpass filters, bandwidth settings) and the TSG-FEP programs. Upon receipt of certain hardware interrupts, SOUNDER sends an address to PICKER pointing to data in the computer memory available to PICKER. SOUNDER accepts scientist-operator commands from COUNSEL to the radar system. SOUNDER also provides access to the TEST system described in Chapter 10.

PICKER — Upon receipt of a queue from SOUNDER, PICKER has the job of either passing all of the data on to MANAGER, or of selecting a subset of the data, thus acting as a data filter. In Product One, PICKER determines if ionospheric

PICKER

echoes from four consecutive transmitter soundings, called a pulse set, arrive at the antennas from approximately the same ionospheric heights. If they do, then the echoes are said to be height- or range-coincident. In its data compaction mode, PICKER will pass to MANAGER only those data that are range-coincident. PICKER also refines the group range estimates. When PICKER has selected its data, it sends MANAGER the location of the data. PICKER also informs SOUNDER that it has completed its job so that SOUNDER can reclaim the memory buffers that it made available to PICKER.

MANAGER

MANAGER transfers the data from PICKER to the disc. MANAGER also sends the location of the PICKER data to GRAPHER and ANALYSER. At the end of a sounding, MANAGER transfers the data from the disc to the magnetic tape.

GRAPHER

GRAPHER converts the compacted data to plotter coordinates and puts the plot data into a memory buffer. The plot buffer data are then sent to the CRT refresh device via an Interdata DMA SELector CHannel (SELCH) that it shares with tape; this SELCH is different from the EKO SELCH.

COUNSEL

Figure 1.1 shows that all tasks accept queues from COUNSEL. This task interacts mainly with the operator of the radar. COUNSEL also has the function of initiating task execution.

ANALYSER

ANALYSER provides for the execution of data analysis, monitoring, or control programs written by individual users. The SEL version of ANALYSER for Product One functions like a command scheduler, executing schedules composed by TEST task SCHED. The scheduler (ANALYSER) accepts three commands, SRUN, SLIST, and SABORT, which are described in Chapter 8. HFRADAR.CSS (Sect. 9.3) is set up to load ANALYSER from file ASKED.TSK, the scheduler task. See Section 10.3 under SCHED for details of the scheduler's operation.

1.7. Task communications [D.C.W.]

The tasks communicate with each other using a system provided by the Interdata MT2 operating system. Each task maintains a circular list, or queue, to which any other foreground task can add a 16-bit address by invoking Supervisor Call 6 (SVC6). The address added to a task's queue is that of an information block of variable length, referred to as a queue block or Q-block. The process of one task's adding a Q-block address to another task's queue is referred to as queueing.

The Q-block normally contains the following information. Byte Ø is an even integer (including Ø) defining the function of the Q-block. Byte 1 is the ASCII identifier of the queueing task, "C" for COUNSEL, "S" for SOUNDER, etc. The status field (bytes 2 and 3) is initialized by the queueing task to -1ØØØ (hex). The remaining bytes (4 through 7) contain commands or addresses, or both, recognized by the queued task. See Section 7.7.

The initial status field value gets changed as follows. The queued task signals the queueing task that the Q-block has been used by adding 1ØØØ + N to the Q-block's status field, where N is an error condition code and the new value of the status field. N = Ø indicates no error, which tells the queueing task that the Q-block information has been received and that the recipient (queued task) is done with it; a Q-block status greater than Ø signals the queued task's detection of an error associated with the Q-block information; a Q-block status less than Ø indicates that whatever action the queueing prompted is incomplete.

Each of the two remaining paragraphs describes an example of inter-task communication, the first from COUNSEL to SOUNDER, and the second from SOUNDER to PICKER.

To change the sounding default starting frequency, the operator/-experimenter types: SFRD FFF, where FFF is the desired new default starting frequency expressed as decimal kHz. The COUNSEL task, having read this information from the console keyboard and determined that it is advice to the SOUNDER task, adds the address of a Q-block to SOUNDER's task queue with an SVC6 call. Byte 1 of the Q-block contains ASCII "C" (43 hex) because COUNSEL did the queueing, and byte Ø contains Ø2 to identify an advice information block; we use the first two bytes in reverse order to name the Q-block "CØ2". Bytes 2 and 3 contain -1ØØØ (hex). The remaining four bytes of the CØ2 Q-block contain the SFRD command's index number and the byte address of FFF, the new starting frequency read into COUNSEL's console input buffer. If SOUNDER determines that FFF is a valid starting frequency, it incorporates this information into the Ionogram Configuration Table (ICT, Section 7.1) for subsequent soundings and adds 1ØØØ (hex) to the CØ2 Q-block's status

field. If FFF is not a valid value, SOUNDER adds 1ØØ1 (hex) to the "CØ2" status. COUNSEL waits for its Q-block status to become non-negative. If it becomes Ø, COUNSEL simply readies itself for more console input, knowing that SOUNDER has accepted the SFRD advice. If the CØ2 Q-block status becomes greater than Ø, COUNSEL knows that SOUNDER has detected an error in the advice given and displays an error message to alert the operator.

When SOUNDER has a new block of echo data ready to deliver to PICKER, it informs PICKER by adding the address of an SØ2 Q-block to PICKER's task queue. The Ø2 in byte Ø is the function code for a sounding data information block; an "S" in byte 1 identifies SOUNDER as the queueing task; the status (bytes 2 and 3) again is -1ØØØ. Bytes 4 and 5 contain the address of the echo data buffer, bytes 6 and 7 contain the address of the corresponding Pulse Configuration Table (PCT, Section 7.2), and bytes 8 and 9 the address of the appropriate KMODE table entry (Section 7.4) if the sounding is K-mode. SOUNDER maintains 6 buffers with corresponding PCTs and SØ2 Q-blocks and uses them in round-robin fashion, so the potential exists for SOUNDER to begin filling a buffer before PICKER has finished with it. This data over-run condition is controlled via the SØ2 Q-blocks' status fields. PICKER will not clear an SØ2 status to Ø (by adding 1ØØØ) until it has finished with the buffer associated with that SØ2 Q-block; SOUNDER will not re-use that buffer or its PCT unless its SØ2 status has been cleared to Ø, which is PICKER's signal to SOUNDER that the buffer has been released. An SØ2 Q-block's status field is never greater than Ø in Product One; a buffer is either free (status Ø) or busy (status less than Ø).

2. SOUNDER

David C. Walden
J. R. Winkelman
Larry D. Schultz

2.1. Introduction [L.D.S.]

The SOUNDER task directs the sequence of events that results in the acquisition of radar echo data into the main system computer memory. In so doing, SOUNDER interacts with the radar transmitter-receiver hardware, the Timing Sequence Generator (TSG) and Front End Processor (FEP) microcomputers, a 64-channel "housekeeping" analog-to-digital converter (HKADC), and the Direct Memory Access (DMA) link between the FEP and the main system computer. This DMA link, operating under an Interdata Selector Channel (SELCH), is referred to as the EKO device. SOUNDER communicates with the radar hardware, the TSG, FEP, and HKADC by way of the 8-bit Digital Input/Output bus (DIO). Device starts and interrupts for the TSG, FEP, and EKO are handled at the Operating System (OS) level by locally written software drivers; the control of the transmitter-receiver hardware, the reading of the HKADC and of the TSG clock, and the loading of the TSG and FEP memories are handled directly at the SOUNDER task level via the DIO bus.

SOUNDER maintains the three main HF Radar system tables, the Environment Table (ET, Sect. 7.3), the Ionogram Configuration Table (ICT, Sect. 7.1), and the Kinesonde Mode Table (KMODE, Sect. 7.4), and constructs a Pulse Configuration Table (PCT, Sect. 7.2) for each transmitted pulse. As each buffer of radar echo data is received via the EKO link from the FEP, SOUNDER passes the buffer's address, together with the address of the associated PCT, to the PICKER task.

2.2. Hardware interrupts and timing [L.D.S]

The sounding software for SOUNDER is designed around the interrupts received from the TSG, FEP, and EKO. SOUNDER invokes Supervisor Call 1 (SVC1) to start these devices and to wait for interrupts from them signaling that their actions are complete. The sequence of these SVC1 calls in SOUNDER is: Start TSG, Wait EKO, Wait FEP, Start FEP, Start EKO, and Wait TSG. The TSG is started every time this sequence of calls is executed; the FEP and EKO devices are started only when the presence of data demands their actions (see Sects. 2.3 and 2.4). A device Wait without a previous corresponding device Start is ignored by the OS. This section discusses the various hardware interrupts and timing requirements of these devices.

2.2. Hardware interrupts and timing

After being commanded by the Interdata computer, the TSG waits for its own internal clock interrupt and then initiates an accurately-timed, single-frequency, sounding sequence. In Product One this is a single pulse. The transmitter can also sound with a sequence of pulses having different phases, a feature that will be used in future products. For a single pulse, the sending of the transmitted pulse and the loading of two FEP input memories (one per channel) take a little over 5 ms from the start of the timing sequence. During this 5 ms, the system computer does not reference any of the instruments on the DIO bus because of various hardware considerations (frequency coherence, bus noise, response times, etc.).

During the time of flight of the pulses, SOUNDER calculates the parameters for the next transmitted pulse. Then it puts itself into a Task-wait state until the radar echoes have returned. During this wait state, the Operating System (OS) allows other tasks to execute in order of their priorities. When the TSG has finished waiting for all the pulse echo data to be received into the input memories of the FEP, an interrupt is sent to the system computer. This interrupt occurs about 5 ms after the TSG starts and is called the TSG DONE Interrupt.

The FEP, under the control of the TSG, starts its program execution when all the radar echo data from one pulse have been received. The primary purpose of the FEP software is to remove all the non-echo signals from the received data. It does this by the techniques of signal deconvolution, thresholding, and running averages. In Product One the FEP reduces the amount of original data by at least a factor of two by thresholding. The FEP filters the radar data and transfers the data from its input memories to its output memories via the FEP scratch memories. For each receiver channel, the FEP finds the maximum peak (maximum maximorum) in the data and includes it in the output for use by SOUNDER. The exact sequence of software operations and execution time depends upon the particular TSG and FEP programs selected. Ultimately, however, the FEP puts the processed radar data into its output memory and signals this event via a hardware interrupt to the Interdata computer. This interrupt is called the FEP DATA READY Interrupt; it typically occurs from 2 to 5 ms after the TSG DONE Interrupt.

Upon command from SOUNDER, EKO transfers the data from the FEP output memory to a buffer in the Interdata memory. The transfer takes from 1 to 5 ms, and its completion generates the EKO DONE Interrupt.

2.3. Data flow [L.D.S]

This section describes the movement of data from the radar receivers into the Interdata system computer memory. The discussion refers to

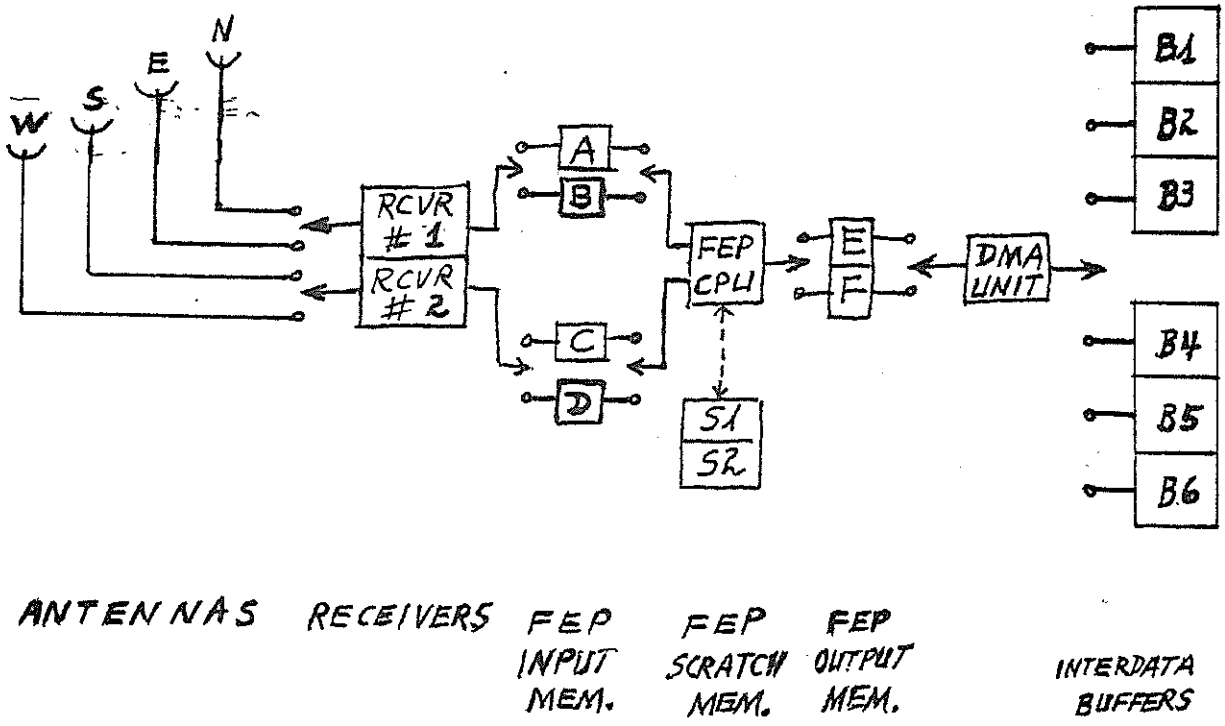


Figure 2.1. DATA FLOW CHART

Figure 2.1, which shows schematically the data flow from the antennas to the memory buffers of the system computer.

The pulses returned from the ionosphere are detected by receiving antennas oriented north, east, south, and west. One pair of antennas is switched on line at a time. Assume for discussion that the antennas are paired by north-south and east-west positions, and that a pair of antennas is switched on line to the two receivers for about 5 ms per pulse.

A given pair of antennas is switched on line for two consecutive pulses, the first at frequency F and the second at frequency F + deltaF, where deltaF is chosen to enable group delay measurements of given accuracy. This means that the antenna-receiver switch in Figure 2.1 toggles at every other pulse.

2.3. Data flow

During the 5-ms receiving period for each pulse, the receivers send their input to the analog-to-digital converters in the FEP. These ADC's produce one X-Y pair of 12-bit data words every 10 microseconds. The TSG routes these digital data to one of the two pairs of FEP input memories. These are labeled as memory pair AC and BD in Figure 2.1. In the mode illustrated by the Figure, the north-south antenna data at frequency F go into FEP memory pair AC, and the north-south antenna data at frequency F + deltaF go into memory pair BD, whereupon the receivers switch to the east-west pair for soundings at frequencies F and F + deltaF. This means that the input memory switches in Figure 2.1 toggle at every pulse.

The TSG controls selection of the FEP input memories. The FEP moves the data from its input memories to its output memories via its internal scratch memories. When data are ready for transfer into the Interdata computer, the Direct Memory Access (DMA) device (EKO) is started by SOUNDER.

The flow of data from specific antenna pairs into the Interdata computer is now considered for successive time intervals. Assume for discussion that pulses are transmitted uniformly in time into the

PULSE NO.	FREQUENCY	ANTENNAS	ELAPSED TIME, ms
1	F1	N-S	0
2	F1 + deltaF	N-S	20
3	F1	E-W	40
4	F1 + deltaF	E-W	60
5	F2	N-S	80
6	F2 + deltaF	N-S	100
7	F2	E-W	120
8	F2 + deltaF	E-W	140
9	F3	N-S	160
10	F3 + deltaF	N-S	180
11	F3	E-W	200
12	F3 + deltaF	E-W	220

Table 2.1. TRANSMISSION SEQUENCE EXAMPLE

ionosphere according to Table 2.1. In the example, the operator has specified north-south and east-west antenna pairs.

2.4. Data pipeline example [L.D.S]

The preceding ionospheric sounding schedule results in the "data pipe-

PERIOD NUMBER	PROCESS 1 RCVR to IM	PROCESS 2 IM to OM	PROCESS 3 OM to ID	PROCESS 4 ID PROCESS	FREQUENCY
1	P1D -> AC1	-	-	-	
2	P2D -> BD1	AC1 -> EF1	-	-	-
3	P3D -> AC2	BD1 -> EF2	EF1 -> B1	-	
4	P4D -> BD2	AC2 -> EF3	EF2 -> B2	B1	
5	P5D -> AC3	BD2 -> EF4	EF3 -> B3	B2	F1
6	P6D -> BD3	AC3 -> EF5	EF4 -> B4	B3	
7	P7D -> AC4	BD3 -> EF6	EF5 -> B5	B4	
8	P8D -> BD4	AC4 -> EF7	EF6 -> B6	B5	
9	P9D -> AC5	BD4 -> EF8	EF7 -> B1	B6	F2
10	P10D -> BD5	AC5 -> EF9	EF8 -> B2	B1	
11	P11D -> AC6	BD5 -> EF10	EF9 -> B3	B2	
12	P12D -> BD6	AC6 -> EF11	EF10 -> B4	B3	
13	-	BD6 -> EF12	EF11 -> B5	B4	F3
14	-	-	EF12 -> B6	B5	
15	-	-	-	B6	

Table 2.2. DATA-FLOW PIPELINE EXAMPLE

line" structure displayed in Table 2.2. The four major processes in the pipeline are: (1) the transfer of data from the receiver into the FEP input memories (IM); (2) the processing of input data within the FEP and the loading of FEP output memories (OM) with processed data; (3) the transfer of data from the FEP output memories to the Interdata (ID); and (4) the data processing within the Interdata computer. These processes form the major column headings of the table. The rows of the table correspond to consecutive pulse periods.

During Period 1, data from Pulse 1 at frequency F1, P1D (i.e., pulse 1 data), are recorded on the N-S antennas, digitized, and are transferred into FEP input memory pair AC as data AC1.

During Period 2, Pulse 2 echo data from the N-S antennas at frequency F1 + deltaF are transferred into FEP input memory BD as data BD1. Also during this period, FEP input memory data AC1 are being processed and the results placed in FEP output memory pair EF via the internal scratch memory. The processed data in the output memory are labeled EF1. This latter transfer occurs while the alternate input memory is being filled. The FEP was designed so that one input memory can be processed while the other input memory is being filled. At the end of this period, one input memory pair (BD) and the output memory pair (EF) are full.

2.4. Data pipeline example

During Period 3, Pulse 3 echo data from the E-W antennas at frequency F1 go into memory pair AC as data AC2. As data are being received from the antennas, the data from Pulse 1 that are stored in the FEP output memory are transferred into a memory buffer in the Interdata computer as data B1. At the conclusion of this latter transfer, the BD1 data from Pulse 2 are transferred into the just-emptied output memory EF and labeled EF2. At the end of this period, the pipeline contains Pulse 1 data in the Interdata computer, Pulse 2 data in the FEP output memory, and Pulse 3 data in the FEP input memory.

During Period 4, Pulse 4 echo data are received via the E-W antennas at frequency F1 + deltaF and are transferred into the BD memory pair as data BD2. Pulse 2 data are transferred from the EF output memory into another Interdata buffer as data B2. Pulse 3 data AC2 are transferred from the FEP input memory pair AC into the output memory as data EF3. The data from the first 4 pulses are now in the pipeline.

By the beginning of Period 7, data from the first four pulses have all been transferred into the Interdata computer. These data in Interdata buffers B1 through B4, taken over a time period of 8Ø ms, constitute one set of data, or pulse set, associated with one frequency sounding of the ionosphere, namely, frequency F1. The data set contains eight peak maxima, one per pulse for each of two channels (see the description of the FEP in Sect. 2.2). SOUNDER computes the average of these maxima and uses it to control the attenuation in the receivers (see commands PKHI and PKLO, Chapter 8).

Task PICKER started processing B1 as soon as it was received. When PICKER finishes with B1, it releases B1 to SOUNDER. Normally PICKER finishes B1 before B2 is ready, and lower priority tasks can execute. If PICKER needs more time than what remains of the 2Ø ms after SOUNDER has run, PICKER can run into the next interval. PICKER normally catches up, but if it doesn't, SOUNDER delays 2Ø ms (one pulse time) and then finds out if it has enough buffers to continue.

At the beginning of Period 11, two complete frequency soundings have been made. Four buffers need to be available for the third frequency sounding. Table 2.2 gives the data currently in the pipeline during this and subsequent periods.

2.5. SOUNDER commands [D.C.W.]

SOUNDER accepts commands from the operator-experimenter through the system console. These commands fall into three groups: (1) ionogram or sounding configuration commands, (2) immediate control commands, and (3) idle-mode commands.

Most of the SOUNDER commands are in the group used to establish the configuration of the next sounding to be run. The configuration commands can be given at any time before the 1- to 19-second countdown begins. When the countdown begins, the specified configuration takes effect and remains in effect for subsequent soundings until explicitly changed. The SFRQ (set starting frequency) and EFRQ (set ending frequency) commands are exceptions in that they affect a single ionogram only. See the descriptions of these two commands, and of SFRD (set default start frequency) and EFRD (set default ending frequency), all in Chapter 8.

Commands SFRQ, EFRQ, SFRD, EFRD, and DFIX apply to ionogram soundings only. All other commands in this group affect both I-mode and kinesonde (K-mode) soundings.

The immediate control commands abort a sounding, turn off the high power transmitter, control the dummy antenna load, select the receiver antenna sequence, and set calibration attenuation; they take effect as soon as they are issued. These commands may be given at any time.

The idle-mode commands can be issued only when the SOUNDER task is in the idle mode, that is, when SOUNDER is neither running a sounding nor waiting to start the next scheduled sounding. SOUNDER is in the idle mode after being first started by COUNSEL and before the first GO is given, and is put into the idle mode by an ABORT or by satisfying count NI (Chapter 8).

Chapter 8 contains descriptions of all commands and their arguments in a single alphabetic order. The commands in each group are listed below. Mandatory letters are underscored.

Ionogram or sounding configuration commands.

COMMAND	MEANING
ANTENNAS	Define antenna selection sequence.
CALA	Specify calibration attenuation.
DESC	Describe experiment in 8 or fewer characters.

2.5. SOUNDER commands

COMMAND	MEANING
DFIX	Set frequency index increment for I-mode sweep.
EFRD	Specify default I-mode ending frequency.
EFRQ	Specify ending frequency for next ionogram only.
EPF	Set maximum number of range-coincident echoes per frequency.
HPE	Specify number of half-words (bytes/2) per echo data set.
KEEP	Set M for M-out-of-N range coincidence for PICKER.
NI	Specify number of soundings to run.
NIPH	Specify number of soundings per hour.
PKHI	Set maximum desired average (mV) for FEP data peaks.
PKLO	Set minimum desired average (mV) for FEP data peaks.
SFRD	Specify default I-mode starting frequency.
SFRQ	Specify starting frequency for next ionogram only.
TSGC	Specify TSG command byte and associated range-Ø correction.
TXATHP	Specify transmitter attenuation for high power.
TXATLP	Specify transmitter attenuation for low power.

Commands EPF and HPE are not fully implemented; they affect only the ICT (Sect. 7.1).

Immediate control commands.

COMMAND	MEANING
ABORT	Abort current or scheduled sounding.
ANTENNAS	Change antenna sequence during sounding.
CALA	Change calibration attenuation during sounding.
DLON	Select transmitter dummy load.
DLOFF	Select transmitter antenna load.
HVOFF	Turn off high power transmitter high voltage (HV).
TXOFF	Turn off high power transmitter HV and filaments.

Idle-mode commands.

COMMAND	MEANING
BOOT FEP	Load the FEP program memory from SYS1:FEPMEM.SND.
BOOT FEP,xxx	Load FEP program memory from SYS1:FEPMEM.xxx.
BOOT TSG	Reset the TSG to bootstrap itself from its own PROM.
BOOT TSG,xxx	Load TSG zero page from SYS1:TSGMEM.xxx.
CDN	Specify sounding countdown value in seconds.

2.5. SOUNDER commands

COMMAND	MEANING
DELF	Specify the delta frequency.
F	Display all selected K-mode frequencies and range limits.
Fn	Display the nth K-mode frequency and range limits.
Fn,FFF	Set the nth K-mode frequency to FFF kHz.
Fn+	Raise the nth frequency one step.
Fn-	Lower the nth frequency one step.
FnX	Delete the nth frequency.
FEPTH	Set FEP data threshold and load FEP.
GO	Start ionogram.
GO mm	Start ionogram mm minutes past the current hour.
GO mm:ss	Start ionogram mm minutes and ss seconds past the current hour.
HVON	Turn on high power transmitter high voltage.
KFIL	Set the NOP fill count for KRUN sounding.
KRUN	Start kinesonde sounding.
KRUN mm	Start KRUN mm minutes past the current hour.
KRUN mm:ss	Start KRUN mm minutes and ss seconds past the current hour.
KSET	Specify number of sets of pulse sets for KRUN.
NOADC	Skip HKADC read.
NOCH1	Ignore channel 1 data peak and zero offsets.
NOCH2	Ignore channel 2 data peak and zero offsets.
NOEKO	Skip EKO start.
NOFEP	Skip FEP start.
NOTSG	Skip TSG start.
PSEQ	Specify pulse set sequence.
RFn	Set the maximum (farthest) range for the nth K-mode frequency.
RGAIN	Save receiver gain profile tables.
RNn	Set the minimum (nearest) range for the nth K-mode frequency.
TEST	Run TEST task.
TPRUN	Run dummy sounding from previously recorded EKO data.
TXON	Turn on high power transmitter filaments.
VFIL	Set the NOP fill count for VUF sounding.
VUF n	Run 30-second test sounding on the nth K-mode frequency.
VUF n mm	Start VUF mm minutes past current hour.
VUF n mm:ss	Start VUF mm minutes and ss seconds past the current hour.

2.6. Program modules and memory utilization [D.C.W.]

To facilitate its modification and to emphasize its structure, SOUNDER is divided into seven program modules. They are listed below in the order that they are loaded when the SOUNDER task is established.

SECTION	NAME	SOURCE FILE	DESCRIPTION
2.7	SPIPE	SNDRPIPE.CAL	Sounding pipeline control
2.8	SADVC	SNDRADVC.CAL	Memory-resident advice
2.9	SNXTF	SNDRNXTF.CAL	Next frequency/pulse selection
2.10	SSUBS	SNDRSUBS.CAL	Local, memory-resident subroutines
2.11	SRENT	SNDREENT.CAL	Global, reentrant subroutines
2.12	STBLS	SNDRTBLS.CAL	Memory-resident tables
2.13	SBFRS	SNDRBFRS.CAL	Task initialization, idle-mode advice, PCT/EKO buffers

The organization of the SOUNDER task program modules reflects several primary considerations. In terms of hierarchy, SPIPE is SOUNDER's main routine, calling subroutines in modules SADVC, SNXTF, SSUBS, and SRENT; SPIPE uses STBLS for variable storage for sounding description and control. The first 6 modules are all required to support a sounding and are therefore memory resident. Beyond the requirement that they be memory resident, modules 1 through 5 are organized with further, but not strict, regard to the hierarcy of subroutine levels.

SADVC and SNXTF are each single, first level, subroutines and are peers: neither calls the other and both call lower level subroutines in modules SSUBS and SRENT.

Subroutines in SSUBS in turn make calls to subroutines in SRENT, but not vice versa.

All of the HF Radar global tables and SOUNDER's major tables are in module STBLS. Some small, local tables, SVC parameter blocks, etc., are stored with the routines that refer to them, and are discussed in the appropriate module sections, 2.7 through 2.13. STBLS contains no executable code.

Module SBFRS contains the SOUNDER task initialization code and routines and tables that are used only in response to idle-mode advice (Sect. 2.5). Since all of SBFRS is used during a sounding for EKO buffers (Sect. 3.4) and PCT's (Sect. 7.2), the idle-mode advice routines are written to disc during SOUNDER initialization and are subsequently read back when they are needed, between soundings only. The idle-mode advice overlay in SBFRS is managed directly by SOUNDER and does not use the MT2 system SVC5 overlay call. The overlay area is restricted to

PCT/EKO buffers 5 and 6 (4116 bytes) to reserve the remaining 4 buffers for TEST mode soundings (Sect. 10.2).

During Product One development it proved useful to be able to retain at least 4 PCT/EKO buffers, one complete pulse set, in memory together; and TEST tasks EXPCT and PLOTEKO assume that there are 4 PCT/EKO buffers available. This assumption will be sacrificed in the future as the number of SOUNDER PCT/EKO buffers is reduced. Only EXPCT and PLOTEKO now require 6 PCT/EKO buffers.

Figure 2.2 represents the SOUNDER memory layout and is scaled to show the relative size of the modules.

2.6.1 SOUNDER flow diagram notes

SOUNDER's flow diagrams, Figures 2.3 through 2.28, are gathered together in Section 2.16.

SOUNDER's flow diagrams are fairly detailed, often referring to specific registers, flags, and tables by name. For a more general understanding, refer to the explanatory text in the module description sections that follow. Some familiarity with Interdata assembly language and conventions is required. Refer to Chapter 7 for descriptions of elements in tables ET, ICT, PCT, and KMODE. The DIO table is described in Section 2.12.8. The flow diagrams distinguish between the current ICT (CICT) used during sounding and the ICT for future soundings (ICT).

The special registers (Sect. 2.7.2) and memory locations used for flags, counters, etc., are referred to by name, e.g.:

GO=-1	set register GO (R1Ø) to -1
NOTSG=Ø	clear location NOTSG to Ø.
INCORE=TEST	copy contents of TEST to switch INCORE
P.TSG=PULSE	copy contents of register PULSE to register P.TSG

The general registers are enclosed in parentheses, e.g.:

NSET=(R13)	replace the contents of NSET with the contents of register 13
(R4,5)=Ø	clear registers 4 and 5 to Ø (32 bits)

The Interdata convention is used for distinguishing between a memory location's contents and address, e.g.:

2.6.1 SOUNDER flow diagram notes

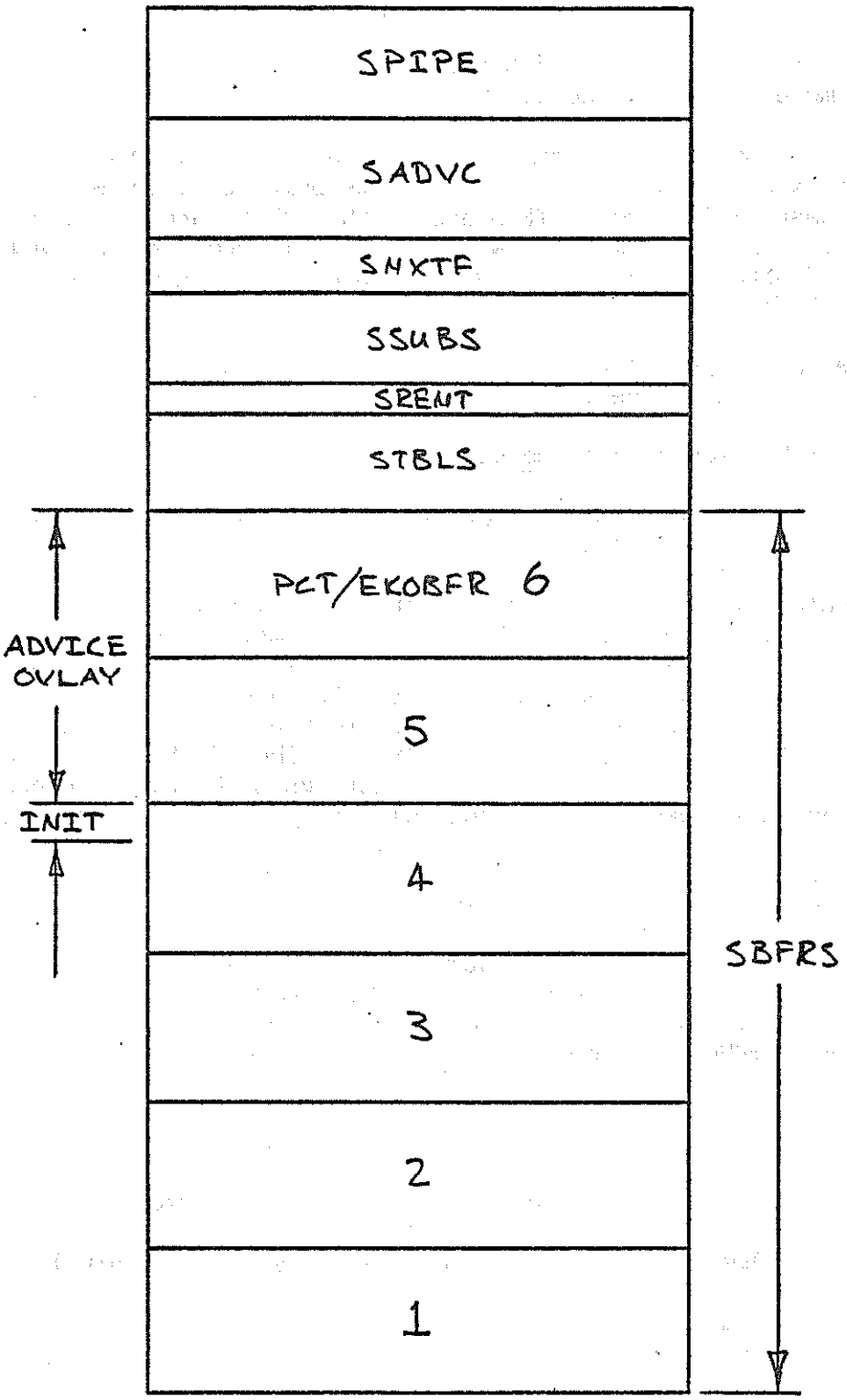


Figure 2.2. SOUNDER MEMORY

```
NXTACT=A(GETADV)    set switch NXTACT to the address of GETADV
```

Elements of tables are referred to by using the table displacement definitions, e.g.:

```
ET.CDN=-1       set the contents of table ET + ET.CDN to -1
```

Operations on circular lists use assembly language mnemonics, e.g.:

```
RTL  SØ2LST      remove element from top of list SØ2LST
ABL  R2,SØ2LST   add element (R2) to bottom of list SØ2LST
```

A rectangle with horizontal divider denotes a subroutine call; the subroutine name appears above the divider, the module in which it resides, below.

Circles labeled with single matching alphabetic characters are used for connectors when connecting arrows are impractical to draw on a given sheet. An off-sheet connection is indicated by a horizontally divided circle, with the sheet number above and the alphabetic connector label below. If an off-sheet connection is to a different module, the destination module name and program label are printed near the circle.

2.7. SOUNDER pipeline and idle-mode control (SPIPE) [D.C.W.]

The SOUNDER task consists of two main loops, both of which are in module SPIPE. The idle-mode advice (IMA) loop is entered at label GETADV after the initialization routine (INIT) has executed at task start-up (Sect. 2.13.1). SOUNDER remains in the IMA loop, checking for advice every 1ØØ ms until a sounding has been initiated by a GO, KRUN, or VUF command (Chapter 8), at which time SOUNDER enters the pipeline code at label NXTIONO and executes within the sounding pipeline control (SPC) loop.

In the SPC loop SOUNDER issues NOP starts to the TSG, marking time until the countdown is completed and the sounding begins. If the ABORT command (Chapter 8) is given at any time, or if NI (Chapter 8) is satisfied, control reverts to the IMA loop; otherwise, SOUNDER remains in the SPC loop and after a sounding issues NOP starts to the TSG, marking time until the next sounding is scheduled to begin. Thus SOUNDER is found to be executing in either the IMA or the SPC loop at all times. The internal switch that controls this is 1/2-word, NXTACT, which is set to the address of GETADV for IMA or the address of NXTIONO for SPC.

2.7.1. The idle-mode advice (IMA) loop

The IMA loop in SPIPE is deceptively small, requiring only a few lines of code. It provides access, however, to all SOUNDER advice routines (Sect. 2.8 and Chapter 8) and to the TEST mode (Sect. 10.2). The distinction between IMA and SPC action is obscured in the TEST mode, since some of the TEST tasks (e.g. DATAFMT, QSTOP) require that the actual SPC code be executed. Thus the SPC loop is tested without involving the other, "downstream", tasks. In addition, the idle-mode command TPRUN simulates SPC action by providing a flow of data without executing the SPC loop to acquire the data. (The SPC loop is executed, however, during a TPRUN countdown to make use of subroutine CLOCK in SSUBS. When the TPRUN countdown is finished, control transfers to GOTPR in SBFRS.)

The IMA loop is discussed in detail below.

2.7.1.1. Enter IMA loop

Refer to Figure 2.3.

Control comes to GETADV after SOUNDER task initialization (Fig. 2.16), after a successful TPRUN (Fig. 2.18), after an ABORT (Fig. 2.7, connector A and Fig. 2.10, connector C), and after NI is satisfied (Fig. 2.5). GETADV sets GO to -1; GO remains -1 until a sounding is initiated. GETADV also clears debugging switches NOTSG, NOFEP, NOEKO, and NOADC to zero (see Chapter 8).

2.7.1.2. Accept advice

The IMA loop actually begins at CKSNDRQ. A call to subroutine ADVICE is made, and at this point advice to SOUNDER is unrestricted. (See Sect. 2.7.2.4 for SPC loop advice.) After the call to ADVICE, the GO register is tested for zero. Unless GO = Ø, an OS delay of 1ØØ ms is executed, and control returns again to CKSNDRQ.

2.7.1.3. Exit IMA loop

If GO = Ø in the above test, a sounding has been initiated and the IMA loop is flagged by setting NXTACT to A(NXTIONO). (See Fig. 2.17, connector G.) (The use of the GO register as an IMA loop flag is the only non-standard use of any of the special pipeline registers discussed in Sect. 2.7.2.)

Before the SPC loop is entered at NXTIONO, however, some precautionary steps are taken: SVC1 calls to wait EKO, wait FEP, and wait TSG are issued in case any previous starts to these devices were left uncleared; the FEP DATA READY signal is cleared in case the last EKO data transfer

was incomplete; the status fields of the SVC1 parameter blocks are cleared to Ø in case any of the above device waits timed out. Next, memory switch INCORE is set to switch TEST. The TEST switch is nonzero if SOUNDER is in the TEST mode, Ø if in normal mode. INCORE is Ø if the SBFRS overlay has been overwritten, and nonzero if the overlay is intact in memory. Thus, if SOUNDER is in TEST mode, the overlay is preserved; otherwise it is overwritten by data when sounding starts. (See Sect. 10.2.) The GO register is now set to -1, as it remains until actual sounding begins. Finally, the PCT address circular list, PCTLST, is synchronized with the SØ2 Q-block address circular list, SØ2LST, to start with EKO buffer 1 and its associated PCT (Sect. 2.12.1). The SPC loop is now entered at NXTIONO (Sect. 2.7.2.1).

2.7.2. The sounding pipeline control (SPC) loop

Except for the very small IMA loop, the SPIPE module is the SPC loop, or pipeline control code, which initiates and controls the data flow described in Sections 2.3 and 2.4. SPC uses registers 5-11 as defined in disc file PCB.CAL under label **SNDR (Sect. 7.6):

GO(R1Ø)	1	Sounding in progress; start TSG with ICT.TSGC (Sect. 7.1) if PULSE > Ø.
	-1	Between soundings; start TSG with scenario Ø (NOP) to mark time.
	Ø	Data overrun; NOP TSG until data pipeline empties.
	X'8ØØØ'	NOP TSG for time fill between K-mode pulse sets.
MODE(R6)	Ø	Ionogram mode sounding initiated by command GO.
	1	Kinesonde mode sounding initiated by command KRUN.
	2	K-mode test run initiated by command VUF.
	-1	Tape replay SPC simulation initiated by command TPRUN.
PULSE(R5)	-1	Between soundings.
	Ø	Sounding just started; start TSG with scenario 1 to calibrate receivers.
	>Ø	TSG DONE interrupt count from sounding start.
PTYPE(R11)		
	-1	Receiver calibration pulse.
	Ø	Sound on frequency F.
	1	Sound on F.
	2	Sound on F + deltaF
	3	Sound on F + deltaF.
	4	NOP fill pulse.

Pulse types Ø-3 as described above are the defaults. See command PSEQ (Chapter 8).

P.TSG(R7) <1 Last TSG start will result in no data (set from GO).
≥1 Normal TSG start (set from PULSE).

P.FEP(R8) <1 No data.
≥1 Echo data at FEP level (set from P.TSG).

P.EKO(R9) <1 No data.
≥1 Echo data at EKO level (set from P.FEP).

The above register definitions are employed by all SOUNDER modules that contain code executed during a sounding, and all subroutines called by SPIPE must return PULSE through PTYPE (R5-R11) intact. The register definitions are invoked at module assembly time by including, in the source code, the directive:

```
COPY  SNDR
```

The main functions of the SPC loop are described below with reference to Figures 2.3-2.5.

2.7.2.1. Initialize for sounding

The SPC is entered at NXTIONO (Fig. 2.3, connector B).

When the countdown to sounding is complete (ET.CDN = Ø), the countdown-in-progress flag is cleared (ET.CDN = -1), and the number of K-mode sets (NSET) is initialized. This count is decremented by the NXTFRQ subroutine if the sounding is K-mode (Sect. 2.9.2.2).

For a KRUN sounding, FKAD and QKAD are initialized for NXTFRQ (Sect. 2.9.1.2) and QPCKR (Sect. 2.7.2.6), respectively, to the address of the first active KMODE table entry. (See Sect. 2.12.4.) The number of lost echoes (NLOST) is zeroed, and unless free run has been selected (NIPH 999), the time to the next sounding (ET.TTNI) is computed from ICT.NIPH.

At PA.4 the SØ4 Q-block status, set to -X'1ØØØ' by CLOCK (Sect. 2.10.9), is tested. If it is Ø, all tasks informed of the start of the sounding have acknowledged the message.

The FEP data limits (MINPK and MAXPK) are computed in ADC counts from PKMIN and PKMAX, as specified by commands PKLO and PKHI (Chapter 8). (These limits are used by subroutine UPAGC for automatic gain control.)

The transmitter state is checked by subroutine CKTXM (Sect. 2.10.6), and an error message is logged if the dump bit is true (Ø).

SPC registers GO, PULSE, and PTYPE are initialized, and NXTFRQ is called to set up the DIO table for the TSG scenario 1 start, which causes the FEP to run its receiver calibration program to establish the zero offsets for the quadrature component data.

2.7.2.2. Setup for next pulse

Since the code from NXTIONO to NXTPLSE is executed only once for each sounding, NXTPLSE (connector D, Fig. 2.3) is the top of the SPC loop while a sounding is in progrress.

If both PULSE and GO are >Ø, the address of the next available PCT is taken from the top of the PCT address circular list (RTL PCTLST) and the list is circulated by returning that address to the bottom of the list (ABL PCTLIST). The PCT for the next pulse is constructed using register PULSE and data from the last call to NXTFRQ: the DIO table, NXTFIX (the next frequency index), DELFLG (deltaF flag), and TFIX (TSG/FEP command byte index, always Ø in Product One). See NXTFRQ, Section 2.9. No PCT is needed for the receiver calibration pulse (PULSE = Ø).

At SETDIO the DIO interface is put into the read/write state by subroutine RWULM (Sect. 2.10.3) and the instruments on the DIO are set up for the next pulse by writing to them the data in the DIO table.

2.7.2.3. Send a start to the TSG

Every pass through STTSG results in a TSG start unless the operator has specified NOTSG (Chapter 8). In this way the HF Radar time-of-day clock is maintained as long as SOUNDER is in the SPC loop. The normal TSG start command byte is taken from CICT.TSGC; if the start is a NOP (scenario Ø) or receiver calibration (scenario 1), the TSG pace and repeat codes (see below) are taken from CICT.TSGC. In either case, the start command byte is put into the SVC1 parameter block at TSGBFR and an SVC1 call starts the TSG.

The TSG pace refers to the time between TSG DONE interrupts, normally 2Ø ms. The pace at which the TSG operates is coded into two bits of CICT.TSGC, allowing for four selectable paces. The actual pace values are determined by straps on the TSG circuit board. The TSG repeat code, also specified by two bits in CICT.TSGC, tells the TSG how many times to execute the specified scenario before issuing the TSG DONE interrupt. Product One uses only a repeat of 1, although subroutine CLOCK (Sect. 2.10.9) keeps proper time with any repeat count. The repeat values are

2.7.2.3. Send a start to the TSG

stored in bytes 16-19 (hex) of TSG page Ø and are copied to table TSGREP when the TSG is bootstrapped.

STTSG writes to the Interdata hex display depending on register GO. If GO = 1, sounding is normal, and NXTFIX (now the current freqency index) is displayed; if GO = Ø, X'D1ED' is displayed to indicate a data overrun; if GO < Ø, the remaining NOP fill count (NFILL) is displayed. The countdown is displayed by subroutine CLOCK.

2.7.2.4. Advice check and next pulse.

Subroutine ADVICE (Sect. 2.8) is called at CKADV to see if the operator has given an immediate or ionogram configuration command (Sect. 2.5). Although non-idle-mode advice is accepted here, the SVC2 command scan is overlayed for Product One (Chapter 11), and thus typing a command results in a disc access. This may be critical during K-mode sounding and when strict time continuity is required. (The number of lost echoes, NLOST, is displayed after any K-mode sounding.)

Subroutine NXTFRQ is now called to set up the DIO table for the next pulse.

2.7.2.5. Wait for EKO data transfer done

The SVC1 call to wait for the EKO DONE interrupt is issued at WTEKO. Wait EKO comes before start EKO in the SPC loop to properly time the data transfer with respect to other events; a wait without a corresponding start is ignored by the OS.

If data at the EKO level are indicated by P.EKO $\geq$ 1 and if data were actually transferred, a check is made to see if this echo is the first received during this sounding. If so, the zero offsets computed by the FEP at receiver calibration (PULSE Ø) are copied from the EKO buffer to the currect ICT table (CICT).

The peak values for channels 1 and 2 are accumulated in TPEAK and the number of peaks in NPEAK. (Subroutine UPAGC (Sect. 2.10.2) computes the average peak for a given pulse set (TPEAK/NPEAK) for automatic gain control.)

The checks for channels 1 and 2 (NOCH1 and NOCH2) are artifacts of earlier days of FEP development and will be removed in Product Two (as will commands NOCH1 and NOCH2).

2.7.2.6. Send SØ2 Q-block to PICKER

Refer to Figure 2.4 here through Section 2.7.2.10.

If WTEKO finds there are no data, a jump is made to WTFEP (connector C). Otherwise control comes through WTEKO to QPCKR (connector A)

QPCKR tests register GO. If sounding is normal (GO = 1), QPCKR gets the address of an SØ2 Q-block into R2 from the bottom of the SØ2 Q-block address circular list (RBL R2,SØ2LST). This Q-block contains the address of the EKO buffer just filled. The address of the Q-block is returned to the bottom of the list (ABL R2,SØ2LST), leaving it intact for GETBFR (GETSØ2).

In K-mode, the current KMODE table (Sect. 7.4) entry address (QKAD) is put in bytes 8 and 9 of the SØ2 Q-block. At this point, in the case there is a data overrun in K-mode (GO = Ø), R2 will contain the address of a special SØ2 Q-block, S2LOST, with zeroes in bytes 8 and 9 to tell PICKER that the echo is lost. If an echo is lost in I-mode, a jump is made to WTFEP.

If PICKER is to be queued, R2 contains the address of the appropriate SØ2 Q-block; R2 is used to initialize the Q-block status to -X'1ØØØ' and to set SVC6.PAR, the queue parameter. The address of the SØ2 Q-block is now added to PICKER's task queue.

2.7.2.7. Wait for FEP DATA READY interrupt

The SVC1 call to wait for FEP DATA READY is issued at WTFEP. FEP DATA READY indicates that the FEP has fully processed the echo data for a given pulse and that the data are in one of the FEP's output memories ready to be transferred to one of SOUNDER's EKO buffers.

The current contents of register P.EKO are copied into RØ in case sounding has stopped and the echoes from the last pulse set have all passed through the pipeline. P.FEP is copied into P.EKO now that the FEP is through with the data from the pulse. If data are still in the pipeline (new P.EKO > Ø), control goes directly to STFEP (connector E, Fig. 2.4); otherwise, if sounding has stopped (GO = -1), the previous P.EKO (RØ) is tested to see if it represented the last echo of the sounding. If so (RØ > Ø), or if the sounding was aborted (NXTACT = A(GETADV)), NXTFIX is set to -1, an impossible value for frequency index. NXTFIX is tested for -1 under SETEX (connector C, Fig. 2.5).

2.7.2.8. Start the FEP

Since the TSG sends the actual signal to the FEP to start processing the data in one of its input memories, a FEP start from the Interdata does no more than set up a FEP wait, as required by MT2. If the TSG is started with NOP scenario Ø, it properly keeps time but sends no signals to the FEP. Therefore, if the last TSG start was NOP Ø, or if the command NOFEP was given after the last ABORT, STFEP issues no FEP start. Otherwise the SVC1 call to set up the FEP wait is made.

2.7.2.9. Get an EKO buffer

If PULSE < 1, denoting that SOUNDER is between soundings, GETBFR is skipped. If PULSE = 1, the FEP is ready to return data from the receiver calibration pulse. Since these data are meaningless, an EKO buffer address is obtained through SØ2LST, but the list is not circulated (RTL SØ2LST, ATL SØ2LST). PICKER is not sent these data, and they are overwritten by the first real echo data.

If real data are coming through (PULSE > 1), GETBFR assumes that a buffer is free (R4 = Ø). If the first echo data from a given pulse set are ready to transfer from the FEP, PTYPE will be 3, and the actual test for free buffers is made. R4 is set to the number of free buffers required for a pulse set: 4 for a normal run or 3 for a TEST sounding. The loop at CKFREE scans the table of SØ2 Q-blocks; R4 is decremented by 1 for each status word that is Ø. If R4 goes to zero, enough buffers are free for the echoes from a complete pulse set. At CKNOP register GO is tested. If data overrun was detected in a previous pass through GETBFR (GO = Ø), and if the buffers have since been released by PICKER (R4 = Ø), and if the data from the interrupted pulse set are all out of the pipeline (P.EKO = Ø), the TSG NOP mode is ended (GO = 1). PTYPE is cleared to Ø (to begin pulse set), and CICT.DFIX is subtracted from NXTFIX; NXTFRQ is called to set up to re-sound on the frequency.

(NXTFRQ correctly establishes the next frequency index, whatever MODE is.)

Back at CKNOP, if GO is not Ø (normal or sounding ended), and if the FEP has data ready to transfer (P.FEP > Ø), but a buffer is not available, the data overrun condition is sensed.

If sounding has not ended (GO = 1), GO is cleared to Ø; the TSG will be sent NOP Ø starts until the pipeline clears. In any case, if the FEP has data ready, its data-ready signal must be cleared by transferring the data out of its output memory. At GETSØ2 the next available SØ2 Q-block address is gotten from the top of SØ2LST, and the list is circulated (RTL SØ2LST, ABL SØ2LST). (Although the data will be transferred even if

GO = Ø, they will not be passed to PICKER (Sect. 2.7.2.6).)

An apparently redundant test is next made on the SØ2 Q-block status field. This is done in case of a data overrun in TEST mode. At SETBFR (connector F, Fig. 2.4) the buffer address (RØ) is extracted from the SØ2 Q-block, and R1 is set to the maximum EKO transfer size minus one. (EKOSIZE is a parameter from SNDR, PCB.CAL, Sect. 7.6.)

2.7.2.10. Start the EKO data transfer

If GETBFR finds there are no data to transfer, a jump is made to WTTSG. Otherwise control comes through GETBFR to STEKO. STEKO computes the EKO buffer end address from registers RØ and R1, as provided by GETBFR, and puts the buffer start and end addresses into the SVC1 parameter block. The SVC1 call to start the data transfer from the FEP is issued unless the command NOEKO was given since the last ABORT.

2.7.2.11. Wait for the TSG DONE interrupt

Refer to Figure 2.5 here through Section 2.7.2.13.

The SVC1 call to wait for TSG DONE is issued at WTTSG. TSG DONE indicates that the data, if any, resulting from the previous TSG start are in one of the FEP's input memories and that the TSG has signaled the FEP to start processing the data. P.TSG is copied into P.FEP; TSG involvement with the pulse is ended.

Subroutine UPAGC is called and if all echoes from a given pulse set have been transferred into EKO buffers, UPAGC updates the receiver gain tables (Sect. 2.12.7) for automatic gain control. Subroutine CLOCK is called to perform time-keeping functions (Sect. 2.10.9), and if a sounding is in progress, CLOCK adds 1 to PULSE to count the TSG DONE interrupt.

2.7.2.12. Read the housekeeping ADC

Unless NOADC has been advised since the last ABORT, one housekeeping ADC channel is read at HKADC each time through the SPC loop. Although HKADC is coded to handle multiple reads per loop, CICT.MRDP (Sect. 7.1) is preset to 1. The DIO interface is put in the read/write state by a call to RWULM (Sect. 2.10.3), and a channel is selected by writing its number to DIO 33. (The current housekeeping ADC channel number is saved in HKCHAN.)

2.7.2.12. Read the housekeeping ADC

After a delay of about 8 microseconds for conversion, the status of the DIO interface is read in a loop at ADCST. When bit 9 (mask = X'4Ø') of the interface status goes to Ø, the conversion is complete and the data are read from DIO 34. The data (12 bits) are stored into the ET at ET.HKADC by channel number (Ø-63).

At HKDONE, if there is no TSG in the system, an OS delay of 2Ø ms is executed to simulate normal TSG timing for debugging.

2.7.2.13. Set SPC loop exit

When the end of an SPC loop is reached, the next action switch (NXTACT) is copied into R12 at SETEX. If no sounding is in progress (PULSE < Ø), a jump through R12 is executed to NXTIONO to mark time till the next scheduled sounding, or to GETADV and the IMA loop if the next scheduled sounding has been aborted. If both PULSE and NXTFIX are non-negative, a sounding is in progress and control goes to NXTPLSE (Sect. 2.7.2.2). If, however, NXTFIX < Ø, WTFEP (Sect. 2.7.2.7) has determined that the echo from the last pulse has returned, and the sounding is terminated (PULSE = -1). The remote control locks are removed from the instruments on the DIO to enable the manual controls, and the transmitter and receiver are fully attenuated. The status field of the SØ6 (sounding end) Q-block is initialized to -1ØØØ (hex) and PICKER is queued. The SØ6 status will remain nonzero until all end-of-sounding action is complete.

At CKFIN subroutine CKLOST is called to report the number of lost K-mode echoes. At this point, if the TEST flag is nonzero, indicating that a TEST run has just finished, NXTACT is set to A(GETADV) and the IMA loop is entered. Otherwise ICT.NIPH is tested. If free run is indicated (ICT.NIPH = Ø), ET.TTNI is set to the incremented countdown value CDNVAL + 1 to allow for a partial second; the countdown for the next sounding will begin when the SØ6 status goes to Ø. (See CLOCK, Fig. 2.14 at CKFRUN.) CICT.NI is now tested. If CICT.NI < Ø, there is no limit on the number of soundings to be run, and control goes through R12 to NXTIONO or GETADV to continue or abort, respectively. If CICT.INUM = CICT.NI, NI is satisfied, and the IMA loop is entered at GETADV. Otherwise control goes through R12 as described above.

2.7.3. SPIPE storage

The following are stored at the end of the SPIPE module, following the SPC code:

STSND	SØ4 (sounding-started) Q-block.
ENSND	SØ6 (sounding-ended) Q-block.
NXTACT	Next action switch (Sect. 2.7).
NLOST	Number of lost echoes for subroutine CKLOST (Sect. 2.10.4).
HKCHAN	Housekeeping ADC channel number.
FEPINIT	Command byte to initialize FEP.
FEPRES	Command byte to reset FEP.
FEPCLR	Command byte to clear FEP data ready signal.
CALAT	Calibration attenuation, set by command CALA.
TXLAT	Low power transmitter attenuation, set by TXATLP.
TXHAT	High power transmitter attenuation, set by TXATHP.
TXMAT	Sounding transmitter attenuation, set by subroutine CKTXM (Sect. 2.10.6).
DLAYPB	OS delay SVC2 parameter block.
NOMS	Number of milliseconds for DELAYPB.
SSVC6	SVC6/SVC4 parameter block.
STTPB	Start TSG SVC1 parameter block.
TSGBFR	TSG start command byte for STTPB.
WTTPB	Wait TSG SVC1 parameter block.
STEPB	Start EKO SVC1 parameter block.
AEKOBFR	Buffer start and end addresses for STEPB (4 bytes).
WTEPB	Wait EKO SVC1 parameter block
STFPB	Start FEP SVC1 parameter block
WTFPB	Wait FEP SVC1 parameter block

2.8. SOUNDER ADVICE subroutine (SADVC) [D.C.W.]

SOUNDER begins execution with its task queue (SNDRQ) enabled; that is, SOUNDER can immediately accept queues from other tasks. SNDRQ is never disabled. However, unlike other tasks, SOUNDER never enables its queue trap; SOUNDER cannot tolerate having its time-critial functions interrupted. Instead, SOUNDER periodically checks to see if anything has been added to SNDRQ by calling subroutine ADVICE. ADVICE is called from the IMA loop (Sect. 2.7.1.2), from the SPC loop during sounding (Sect. 2.7.2.4), and from the TPRUN loop (Sect. 2.13.3).

Module SADVC contains the entry to and the exits from ADVICE, as well as the individual advice routines that are memory-resident during sounding. Disc-resident, or overlayed, advice routines are in module SBFRS (Sect. 2.13), although SADVC contains the routine that reads in the overlay (GETOV). ADVICE entry and exit, and GETOV are discussed below.

Refer to Figures 2.6 through 2.9.

2.8.1. Entry to ADVICE

The ADVICE subroutine has a single entry point; the return link is R15:

```
BAL   R15,ADVICE
```

When SOUNDER is queued, the queue parameter, or the address of the sender's Q-block, is added to the bottom of circular list SNDRQ, the SOUNDER task queue. Therefore, in order to respond to advice in the order that it is given, ADVICE removes entries from the top of SNDRQ (RTL R2,SNDRQ). If the list is empty, indicated by overflow in the condition code, ADVICE returns immediately. Otherwise ADVICE saves the pipeline registers (PULSE, or R5 through R15) in an area, SREGS, in module STBLS (Sect. 2.12.10). R2, which contains the queue parameter, is kept intact by all ADVICE routines, since it is used in the exit routine. After saving the pipeline registers, ADVICE loads registers R7 through R9 with commonly used table addresses. R7-R9 are referred to throughout by special mnemonics:

I. (R7) address of ICT (next ICT)
C. (R8) address of CICT (current ICT)
E. (R9) address of ET

The sender code, from byte 1 of the Q-block, is now checked. SOUNDER accepts advice from COUNSEL (Sect. 2.8.4), ANALYSER (Sect. 2.8.5), and from the TEST tasks (Sect. 2.8.6).

2.8.2. Exit from ADVICE

The individual advice routines, both in SADVC and SBFRS, return through one of two exit routines where an error condition code is loaded into RØ. If the advice is acceptable as given, a jump is made to OKADV, where RØ = Ø; if the advice cannot be accepted, a jump is made to BADADV, where RØ = 1. At CLADV, 1ØØØ (hex) is added to the error condition code (RØ), which in turn is added to the status field (bytes 2 and 3) of the advice Q-block. In the case of advice from COUNSEL, since the status of the CØ2 Q-block is initialized to -1ØØØ (hex), the CØ2 status after exit from ADVICE will be Ø (OKADV) or 1 (BADADV). A positive value in the CØ2 status field evokes an error message from COUNSEL (Fig. 6.3).

Having answered the advice queue by clearing the Q-block status, ADVICE restores the pipeline registers from SREGS and returns through R15.

In future products further use will be made of the error condition code.

2.8.3. Get ADVICE overlay

While in the IMA loop (Sect. 2.7.1), SOUNDER can accept all advice without restriction, including that which is overlayed. However, after a sounding the ADVICE overlay, having been overwritten by EKO data, must be restored from disc. (See Fig. 2.2.)

The code to restore the overlay, written to disc by INIT (Sect. 2.13.1), is at GETOV, under connector C in Figure 2.6. GETOV is entered indirectly through C.ADV (Sects. 2.8.4 and 2.8.4.1) with the address of an overlayed routine in R4. (Only advice from COUNSEL is overlayed in Product One.)

GETOV first checks switch NXTACT (Sect. 2.7) to see if overlayed advice is acceptable. Unless SOUNDER is in the IMA loop, the BADADV exit is taken (Sect. 2.8.2). If idle-mode advice is acceptable, GETOV tests switch INCORE (Sect. 2.7.1.3).

If the overlay is already in memory, GETOV branches to the appropriate routine in SBFRS through R4. If the overlay has been overwritten (INCORE = Ø), it must be restored; logical unit 6 is assigned to disc file SYS1:SOVLAY.SND (Sect. 2.14), written at SOUNDER start up, and switch INCORE is set. The overlay is now read into EKO buffers 6 and 5 (Fig. 2.2) as a contiguous file and LU6 is closed.

The exclusive OR checksum of the overlay, including the checksum byte appended to it by INIT (Sect. 2.13.1), is computed. If the result is Ø, the overlay read is successful, and control goes through R4 to the overlayed advice routine. The overlayed routine in turn exits to OKADV or BADADV (Sect. 2.8.2).

Subsequent calls on GETOV will find the overlay intact until a new sounding is run.

2.8.4. Advice from COUNSEL

The vast majority of SOUNDER advice in Product One comes from COUNSEL, the result of SOUNDER-directed commands typed at the console. These are listed in Section 2.5.

If the sender code in byte 1 of the advice Q-block is an ASCII C, control comes to C.ADV (Fig. 2.6). If the Q-block is CØ4, CANHF has been typed; subroutine RWULM is called to put the DIO interface in the read/write state, OFFTX (Fig. 2.8) is called to shut down the high power

transmitter, and SVC 3,Ø is executed to terminate the SOUNDER task. If the Q-block is CØ2, the operator has typed a normal SOUNDER command, and control goes to CØ2.Q.

Bytes 4 and 5 of the CØ2 Q-block contain the advice index resulting from COUNSEL's scan of the command mnemonic table, SMNEM. This index is loaded into R1 and doubled to become the index to the advice routine address table, ADV1. (SMNEM and ADV1 are discussed in Sect. 2.8.7.) RØ is loaded with the advice routine address (RØ = ADV1(R1)), an address within module SADVC. The address of GETOV is put into R15 (Sect. 2.8.3) in case the main routine to be executed is overlayed. (See Sect. 2.8.4.1.)

Register MODE is now tested to see if a TPRUN is in progress, and if so (MODE = -1), all advice but ABORT is rejected. If SOUNDER is in I-mode or K-mode, control goes through RØ to the appropriate SADVC routine.

Representative examples of SADVC advice routines follow.

2.8.4.1. The GO command

The GO command serves as an example of idle-mode, overlayed advice. After the operator types GO, CØ2.Q branches through RØ to GOIONO (Fig. 2.9) with R15 = A(GETOV). At GOIONO, R4 is loaded with the address of SBFRS routine OVGO, and a jump is made to GETOV through R15 (BR R15). GETOV in turn jumps to routine OVGO (Sect. 2.13.2) through R4, and the actual GO routine is executed. At the end of the GO routine (under connector G, Fig. 2.17), NXTACT is set to the address of NXTIONO, and the SPC loop is entered. A subsequent GO without an intervening ABORT is trapped at GETOV (connector C, Fig. 2.6).

2.8.4.2. The ABORT command

The ABORT command (connector A, Fig. 2.7) can be given at any time and is therefore memory-resident in SADVC. The ABORT routine first tests the MODE register. If a TPRUN is in progress (MODE < Ø), MODE is cleared to Ø to flag a TPRUN ABORT. (See Fig. 2.18 under connector B.) If MODE $\geq$ Ø (I- or K-mode), ET.CDN $\geq$ Ø means that a countdown is in progress, and control goes to ABRT2.

If neither a countdown nor a sounding is in progress (ET.CDN < Ø and PULSE < Ø), NXTACT is set to A(GETADV), and the OKADV exit is taken. If control is in the IMA loop, it will remain there; if control is in the SPC loop, marking time until the next scheduled sounding, the SPC loop will be exited at SETEX (Sect. 2.7.2.13), and the scheduled sounding will be cancelled.

If a sounding is in progress, ABRT2 clears CICT.EFIX to Ø, to end an I-mode run, and clears NFILL (NOP fill counter) and NSET (K-mode set counter) to end a K-mode run. (See NXTFRQ, Sect. 2.9.)

In case a countdown is in progress, the ABORT is not sensed until, at the end of the countdown, NXTFRQ finds that SFIX > EFIX or that NSET = Ø.

2.8.4.3. Transmitter control

Transmitter control routines HVOFF, DLON, and DLOFF are discussed together here because they all share routine TXCOM (center column, Fig. 2.8) and because they demonstrate how hardware control commands are handled.

The high power transmitter is controlled via 5 bits using DIO address 3Ø. Bit Ø is the least significant bit (LSB). The bit assignments are

BIT	CONTROL FUNCTION	Ø	1
Ø	Filament (FIL)	On	Off
1	High voltage (HV)	On	Off
2	Reset (RES)	Active	Inactive
3	Set (SET)	Active	Inactive
4	Dummy load (DL)	Antenna	DL

Table 2.3. DIO 3Ø HIGH POWER TRANSMITTER CONTROL BITS

shown in Table 2.3. RES and SET are used in the TXON and HVON (Sect. 2.13.7) sequences and are otherwise kept inactive (1).

The transmitter relay/status flags are read via DIO 32. The condition listed is indicated by a bit value of Ø. Bit Ø is the LSB. The bit assignments are shown in Table 2.4.

Only bits Ø-2 of Table 2.4 are referenced by this section. Each of the three routines discussed below enter the common code at TXCOM with R4 and R5 set up. R4 contains a hex mask indicating which states (among FIL, HV, and DL) are to be preserved; R5 contains the known part of the command that will be sent via DIO 3Ø, with bits RES and SET always 1, or inactive.

HVOFF turns off the high power transmitter HV and preserves the current states of FIL and DL. R4 is set to 11 (hex); R5 is set to E3Ø (command E to DIO 3Ø).

2.8.4.3. Transmitter control

BIT	CONDITION WHEN Ø
Ø	FIL off
1	HV off
2	DL connected
3	HV dump
4	Relay voltage reset
5	9Ø-second HV delay
6	Fault holdoff
7	Relay voltage failure

Table 2.4. DIO 32 TRANSMITTER RELAY STATUS FLAGS

DLON selects the dummy load, preserving FIL and HV. R4 is set to 3 and R5 to 1C3Ø.

DLOFF selects the antenna, preserving FIL and HV. R4 is set to 3 and R5 to C3Ø.

TXCOM calls ROULM (Sect. 2.10.3) to put the DIO interface in the read-only state, addresses DIO 32, then, after a short delay, reads the DIO 32 flags into R6 and copies them into R3. The copy in R3 is logically ANDed with 3, extracting the two low-order bits, FIL and HV, only. This is because the position of the DL bit is not the same for DIO 3Ø and DIO 32. Next, R6, still intact as read, is tested; if DL is on in R6, it is also turned on in R3, but at bit position 4 instead of 2. R3 now contains the states of FIL, HV , and DL as read from DIO 32, but in DIO 3Ø order. R3 is XORed with R4 to toggle (reverse the sense of) the preserved bits; then R3 is ANDed with R4 to extract only the on bits specified by the preserve mask. The DIO interface is put in the read/write state with a call to RWULM. The preserved bits, now command bits, are positioned to the high-order byte of R3 and are ORed with R5. R3 is then written to DIO 3Ø to command the transmitter. The OKADV exit is taken.

2.8.4.4. The ANTENNAS command

ANTS (Fig. 2.8) is executed to set the receiver antenna selection sequence. The command sent on the DIO to select an antenna is always one less than the number of the antenna selected; the commands to select antennas 1-4 are Ø-3, respectively.

ANTS initializes R3 to 3 (the command upper limit), R4 to the address of ANT (the antenna DIO command table in SNXTF), and R5 to NPTYPE*2 or 8 (the number of entries in table ANT). NPTYPE is a parameter in SNDR, PCB.CAL (Sect. 7.6), and is the number of pulse types, or the number of pulses per pulse set.

At ANTLP, subroutine CONVD (Sect. 2.8.4.6) is called to convert a decimal ASCII antenna number, typed at the console, to binary. The converted antenna number, returned in RØ, is decremented by 1 to form the DIO command. RØ is then tested for validity.

If the antenna command is not <Ø and is not >R3 (3), it is put into table ANT, indexed by R4. If RØ is not Ø-3, the BADADV exit is taken. R5 is decremented by 1; when R5 goes to Ø, the ANT table is full and the OKADV exit is taken. Otherwise the ANT table index (R4) is incremented by 1; and R1, loaded from bytes 6 and 7 of the CØ2 Q-block by CONVD, is incremented by 1 to skip the comma or blank separator between antenna numbers, and put back into CØ2 + 6 (see Sect. 2.8.4.6). A jump is made back to ANTLP to continue the loop.

2.8.4.5. The DELF command

Routine DELF (Fig. 2.8), executed to set deltaF for future soundings, represents a minor exception to the discussion in Section 2.8.3. DELF calls upon GETOV with an SADVC routine address in R4 instead of one in SBFRS. This occurs because, although the main DELF routine, CDELF, resides in SADVC, CDELF calls subroutine GETMAP (Sect. 2.13.9) and executes routine DSPALL (Sect. 2.13.4), both in SBFRS. Therefore, the overlay must be in memory for CDELF to execute, and DELF is an idle-mode command. The same is true for the STSEQ routine for the PSEQ command; both DELF and PSEQ affect the protected frequencies map. (See note under KRUN, Chapter 8.)

DELF jumps to GETOV with the address of CDELF in R4. GETOV in turn jumps back to CDELF with the overlay in memory. CDELF calls CONVD which returns the new deltaF in RØ. If RØ is not between Ø and 255 kHz, the BADADV exit is taken. Otherwise RØ is put into ICT.DELF and routine NEWMAP, common to CDELF and CSTSEQ, is entered.

NEWMAP loads the K-mode map flag, "K", into RØ, sets MAP (Sect. 2.8.8) to -1, and calls subroutine GETMAP. Because MAP now does not equal RØ, a new copy of the K-mode protected frequencies map is read from disc and adjusted. (See GETMAP and ADJMAP, Sect. 2.13.9 and Fig. 2.25.) The loop at NXTKM is entered to check each selected K-mode frequency for validity against the new map. When the KMODE end flag (Ø) is sensed (Sect. 7.4), routine DSPALL in SBFRS (Fig. 2.19, connector A) is executed to display the KMODE table, possibly altered by the DELF or PSEQ command.

2.8.4.5. The DELF command

At NXTKM unused KMODE entries are ignored (<Ø) and selected frequencies (>Ø) are passed through subroutine MAPFIX (Sect. 2.10.10) to check their validity against the new map. If MAPFIX returns conditon code >Ø, the frequency selected is still usable; otherwise the frequency is deleted from the KMODE table.

2.8.4.6. Convert numerical ASCII to binary

The CØ2 advice Q-block from COUNSEL contains, in bytes 6 and 7, the byte address of the argument(s), if any, of a console command. For example, if the operator types:

```
PSEQ  Ø,Ø,1,1
```

CØ2 + 6 points to the space following the "Q". CONVD and CONVH (Fig. 2.9) convert decimal and hexadecimal ASCII arguments, respectively, to binary. They call SVC2 code 15 with function codes X'CØ' and X'4Ø', respectively, both of which ignore leading spaces. The SVC2 calls in both subroutines require that the input string byte address (CØ2 + 6) be in R1; the converted number is returned in RØ.

The first non-numerical character encountered terminates the conversion, and R1 is updated by SVC2 to point to the terminating character. Any advice routine that makes repeated calls to CONVD or CONVH has to update the byte address in CØ2 + 6. In the above example, after the first call to CONVD, RØ contains Ø, and R1 points to the first comma. Therefore, the PSEQ routine adds 1 to R1 to skip the comma and stores R1 back into CØ2 + 6. The next time CONVD is called, R1 is loaded from CØ2 + 6, SVC2 is called, and R1 is updated to point to the second comma, and so forth.

The SVC2 call sets the condition code (CC) based on the results of the conversion. This CC is tested at CONCOM; if no conversion takes place or if the number is too large, the BADADV exit is taken (BTC 5,BADADV). Finally, just before the subroutine return (common to CONVD and CONVH), the CC is set based on the converted number in RØ (LHR RØ,RØ).

Because of the potential exit to BADADV, CONVD and CONVH are called only from advice routines.

2.8.5. Advice from ANALYSER

In Product One the ANALYSER task functions as a command scheduler (v. SCHED, Sect. 10.3) and answers queues from other tasks. At entry A.ADV in SADVC, the OKADV exit is taken. In case a user-written ANALYSER gives advice to SOUNDER, the code to interpret the advice should be included at A.ADV.

2.8.6. Advice from TEST tasks

All TEST tasks send a TFF Q-block to SOUNDER with only two possible function requests expressed in bytes 4 and 5. If bit 14 of TFF + 4 is on, control goes to ENDTEST (Fig. 2.24, connector A). ENDTEST terminates the TEST task and reinstates GRAPHER in partition 5. (See Sect. 10.1.) If bit 15 of TFF + 4 is on, the TEST task wishes to return control of the console to COUNSEL. COUNSEL initializes the status of the CØ2 advice Q-block to -1ØØØ (hex) and does not initiate another read from the console until the status is cleared, indicating that the advice has been acted upon (Fig. 6.3). In effect, COUNSEL relinquishes control of the console until the CØ2 status is cleared.

On receipt of a CØ2 Q-block resulting from the TEST command, ADVICE saves the address of the CØ2 Q-block in TRQ and returns without clearing the CØ2 status (Fig. 2.24, under CLS6). (This is the only return from ADVICE which does not go through OKADV or BADADV.) Thus the clearing of the CØ2 status is deferred until the TEST task sends the TFF message with bit 15 of TFF + 4 set, giving the TEST task temporary control of the console.

Some TEST tasks release the console shortly after they are started up (e.g., DATAFMT, QSTOP); others keep it until they are ready to be terminated (e.g., FRMAP, PLOTEKO).

2.8.7. SOUNDER advice tables

SMNEM is the SOUNDER task command mnemonic table. Its address is passed to COUNSEL in bytes 6 and 7 of the SØØ Q-block at task startup; COUNSEL scans SMNEM to get the advice index which goes into bytes 4 and 5 of the CØ2 Q-block (Sect. 2.8.4). Required characters are flagged with X'8Ø'; entries end with a null byte, but need not be on 1/2-word boundaries. SMNEM must itself end with two null bytes.

ADV1 is the table of advice routine addresses and corresponds one-to-one with table SMNEM. It is indexed by the advice index, from CØ2 + 4, times 2 (two bytes/entry). ADV1 must start on a 1/2-word boundary. All ADV1 entries are local to SADVC.

SMNEM and ADV1 channel advice based on the first word of an operator command. Multiple word commands, like BOOT TSG, require further scrutiny which is the responsibility of the first-word advice routine.

2.8.8. SADVC storage

In addition to the above tables, SADVC contains the following:

PKDPB	SVC2 parameter block for CONVD (Sect. 2.8.4.6)
PKHPB	SVC2 parameter block for CONVH (Sect. 2.8.4.6)
PTBLS	Table of system table addresses (Sect. 2.8.1)
SNDRQ	SOUNDER task queue (Sect. 2.8)
TRQ	Address of TEST CØ2 Q-block (Sect. 2.8.6)
INCORE	SBFRS overlay switch (Sect. 2.7.1.3)
TEST	TEST mode switch (Sect. 2.7.1.3)
NOTSG	Ø for TSG OK
NOFEP	Ø for FEP OK
NOEKO	Ø for EKO OK
NOADC	Ø for housekeeping ADC OK
NOCHN	Ø,Ø for both channels OK
CLSPB	Parameter block for closing LU6 (Sect. 2.14)
ASMAP	Assign LU6 to SYS1:<MAP>HUSH.SND (protected frequencies maps, Sect. 2.12.9)
MAP	Protected frequencies map flag (ASCII I or K)
ASOVLAY	Assign LU6 to SYS1:SOVLAY.SND (SBFRS overlay)
SOVPB	Read/write SOVLAY.SND
SRGPB	Write RGAIN.SAV (AGC tables, Sect. 2.12.7)
RGREC	RGAIN.SAV record number

2.9. Next frequency and pulse selection (SNXTF) [D.C.W.]

Subroutine NXTFRQ sets up the DIO table (Sect. 2.12.8) for the NXTPLSE-SETDIO routine in module SPIPE (Sect. 2.7.2.2). NXTFRQ also sets DELFLG, the deltaF flag, and TFIX, the ICT.TSGC-ICT.FEPC index (see STFRQ, Sect. 2.9.3). NXTPLSE compacts the DIO table, DELFLG, and TFIX into a PCT (Sect. 7.2); SETDIO sends the commands in the DIO table to the DIO interface device. The DIO table, DELFLG, and TFIX are set up only during normal sounding, that is, when GO = 1 (Sect. 2.7.2).

In Product One there are 4 pulses in each pulse set. NXTFRQ uses register PTYPE to select and execute the appropriate pulse routine and to index the antenna command table, ANT, and the pulse set sequence code table, PSEQ. These tables are set by commands ANTENNAS and PSEQ (Chapter 8), and are used by subroutines STANT and STFRQ (Sect. 2.9.3). PTYPE is initialized to -1 by NXTIONO (Sect. 2.7.2.1), and when called, NXTFRQ executes pulse routine NXTF.CAL to set up for the receiver calibration TSG start. This happens only on the first pulse of each sounding. Thereafter PTYPE is Ø-3, representing pulses 1-4 of a pulse set, unless the K-mode NOP fill count specified by command KFIL or VFIL is positive. Then PTYPE = 4 means that routine NXTK.P4 is to be executed to enter the NOP fill mode. When the fill count is satisfied, PTYPE is cleared to Ø,

and the next pulse set begun.

The sounding mode register, MODE, determines which set of pulse routines is indexed by PTYPE. Although for Product One the pulse routines for pulse types -1, 1, 2, and 3 are the same for both I- and K-modes, SNXTF has separate pulse routine jump tables for the two modes, and in principle the sets of pulse routines can be entirely separate. Likewise, although pulse types 1, 2, and 3 use the same routine, separate entry points are assembled. (See Fig. 2.11, under connector A.) Thus NXTFRQ is entirely table-driven and pulse routines may be inserted or modified with minimum difficulty.

The pulse routines for PTYPE -1 through 3 exit through NXTF.EX (Fig. 2.11, connector C). NXTF.EX adds 1 to PTYPE, keeping PTYPE in the range Ø-3 for I-mode, but allowing PTYPE to go to 4 for the K-mode NOP fill between pulse sets.

Input GO: NXTFRQ is executed only if GO = 1 or 8ØØØ (hex).
MODE: sounding mode.
PTYPE: pulse type for pulse routine selection.
NXTFIX: current pulse set base frequency index.
CICT.DFIX: frequency index increment (Sect. 7.1).
CICT.EFIX: I-mode frequency index limit.
FRQMAP: protected frequencies map (Sect. 2.12.9).
FKAD: KMODE table entry address (Sect. 2.12.4, 7.4).
KMODE: source of K-mode base frequency index.
PULSE: TSG interrupts from sounding start.
NPTYPE: number of pulses per pulse set (see SNDR, Sect. 7.6).
RAGC: receiver AGC profile table for I-mode (Sect. 2.12.7).
RAGCK: AGC profile table for K-mode.
CALAT: calibration attenuation from command CALA.
TXMAT: transmitter attenuation (see Sect. 2.10.6).
NSET: number of sets of K-mode pulse sets, from NXTIONO (Sect. 2.7.2.1).
FILLK: KRUN NOP fill count from command KFIL.
FILLV: VUF NOP fill count from command VFIL.
CICT.DELF: delta frequency (deltaF) from command DELF.
ANT: antennas table from command ANTENNAS (Sect. 2.9.3).
PSEQ: pulse set sequence code table from command PSEQ (Sect. 2.9.3).

2.9. Next frequency pulse selection (SNXTF)

Output GO: set to 8ØØØ (hex) for K-mode NOP fill, set to 1 after NOP fill complete, set to -1 at sounding end.
PTYPE: incremented by 1 by NXTF.EX, cleared to Ø at start of each pulse set.
NXTFIX: incremented by CICT.DFIX for I-mode, set from FKAD for K-mode.
FKAD: advanced to next/first KMODE entry for KRUN, remains constant for VUF.
NSET: decremented by 1 after each pulse set for VUF, after each pass through KMODE (or each set of pulse sets) for KRUN.
NFILL: set from FILLK or FILLV for KRUN or VUF, respectively, at start of NOP fill, then decremented by 1 until NFILL = Ø.
DIO: table of DIO commands.
DELFLG: deltaF flag for PCT.
TFIX: ICT.TSGC-ICT.FEPC index, always Ø.
RAGCAD entry: part 1 is the value of PULSE at which to update AGC profile, part 2 is address of the update in RAGC or RAGCK (see UPAGC, Sect. 2.10.2).
NXTACT: set to A(GETADV) if sounding aborted (Sect. 2.7).
CICT.EFIX: set to NXTFIX at sounding end.
CICT.FFRQ: set from NXTFIX at sounding end.
CICT.ETS: set to ET.SEC at sounding end (see CLOCK, Sect. 2.10.9).
R5(PULSE)-R14: returned intact using storage area SREGS except for GO and PTYPE.
R15: return address.

Method Refer to Figures 2.10 and 2.11.

NXTFRQ saves registers PULSE(R5) through R15 in SREGS then puts PTYPE times 2 into PTYPE to index the I- or K-mode pulse routine jump table. PTYPE*2 is saved in PTMP for subroutines STANT and STFRQ to use to index tables ANT and PSEQ (see Sect. 2.9.3). NXTFRQ puts MODE*2 into MODE to index the sounding mode routine jump table, MODEJT. (See SNDRNXTF.CAL listing.) Control then goes through MODEJT to either IMODE or KMODES, routines discussed below in 2.9.1 and 2.9.2, respectively. Routines that are common to both modes are discussed in 2.9.1. Although NXTFRQ is called during a TPRUN countdown, control always goes through IMODE to NEWGO, with GO = -1.

2.9.1. Ionogram mode pulse set up (IMODE)

IMODE checks the sounding condition by copying GO into R2. Unless GO = 1, IMODE exits immediately to NEWGO (Fig. 2.10, connector B), restores registers PULSE-R15 from SREGS, copies R2 back into GO and returns from NXTFRQ. (NEWGO is always entered with the exit value of GO

in R2.) If GO = 1, sounding is normal, and IMODE loads the pulse routine address, based on PTYPE, into RØ from the I-mode jump table, INXTJT. The INXTJT routine address entries are ordered like the horizontal row of circular connectors in Figure 2.10, going from left to right. The individual routines are discussed below. Before jumping to the appropriate pulse routine through RØ, IMODE loads the addresses of tables DIO, CICT, and ET into R4-R6, which are referred to in the code as D., C., and E., respectively.

2.9.1.1. Receiver calibration pulse setup (NXTF.CAL)

Refer to Figure 2.10, connector F.

NXTF.CAL is executed to set up the first pulse of each sounding when PTYPE = -1. The calibration pulse in this context is not to be confused with the low level replica of the transmitted pulse injected into the system and which appears at range Ø in the FEP data. The purpose of the one-time receiver calibration pulse is to get the X and Y zero-offsets into the FEP data (Sect. 3.4). On the first pulse of each sounding, the TSG is started with scenario 1, regardless of the TSGC command. The TSG in turn signals the FEP to run its special receiver calibration program to calculate the zero-offsets which appear as channel-derived data in all subsequent EKO buffers from the FEP. The zero-offsets are copied into the current ICT from the first good EKO buffer by WTEKO (Sect. 2.7.2.5).

NXTF.CAL selects a frequency of 11111.1 kHz, meaningless except that it is within the range of the synthesizer. NXTF.CAL clears all other DIO table entries to Ø, so that SETDIO will put in maximum RF, IF, calibration, and transmitter attenuations and select the narrowest receiver bandwidth, 1 kHz. Any legal antenna command could be given, but Ø is convenient; command Ø to the unimplemented octave filters means nothing. All DIO table entries are established as reasonable DIO commands, however, so SETDIO does not have to make a special case out of the receiver calibration pulse. NXTF.CAL exits through NXTF.EX (Sect. 2.9.1.4) where PTYPE is incremented to become Ø.

2.9.1.2. I-mode pulse type Ø setup (NXTF.PØ)

Refer to Figure 2.10, connector D.

The first pulse of a pulse set is flagged by PTYPE = Ø. This causes the next base frequency index, NXTFIX, to be selected. NXTF.PØ clears R7 to Ø, to flag that the entry into common code NXTFOK (Sect. 2.9.1.2.1) is from I-mode, and loads the old NXTFIX into R14, to which it adds CICT.DFIX. At CKEND, if R14 > CICT.EFIX, the ionogram sweep has been completed, and control goes to FIXMAX (Sect. 2.9.1.5). Otherwise R14 is passed through MAPF (Sect. 2.10.10) to verify that it represents

an unprotected frequency, in which case control goes to NXTFOK. Otherwise FRQMAP is scanned for the next usable frequency index above that in R14. The next higher unprotected frquency index (R14) is in turn tested against CICT.EFIX to determine if the sweep is to continue. The FRQMAP scan is made faster by skipping over bytes of the map that are Ø, rather than making 8 fruitless calls to MAPF.

2.9.1.2.1. Common code for pulse type Ø (NXTFOK)

NXTFOK is entered with R7 cleared to Ø, if from NXTF.PØ, or set to the address of the appropriate K-mode RF/IF attenuation byte pair, if from NXTK.PØ; R14 is set to the I- or K-mode next base frequency index and is stored into NXTFIX. Subroutine NTOFR (Sect. 2.11) is called and the returned corresponding base frequency, in hundreds of Hz, is saved in NXTFREQ for calls to subroutine STFRQ throughout the pulse set. STFRQ is now called to set the TFIX and DELFLG flags and the BCD synthesizer frequency commands in the DIO table. STANT is called to set the DIO antenna commands. The mode of entry into NXTFOK is now tested by copying R7 into R1. If R1 is not Ø, it has been already set by NXTK.PØ and becomes the second half of the RAGCAD entry at SVAGC. Otherwise a call is made to GTAGC (Sect. 2.10.1) and the address of the appropriate I-mode RF/IF attenuation byte pair is returned in R1. At SVAGC, the pulse at which the AGC profile table, RAGC (I-mode) or RAGCK (K-mode), is to be updated is computed:

```
RØ = PULSE + NPTYPE + 2
```

This calculation allows for the pipeline delay and assumes that UPAGC is called before PULSE is incremented by CLOCK. (See WTTSG, Sect. 2.7.2.11.) The entry to RAGCAD is now made by adding RØ followed by R1 to the bottom of the list. (See UPAGC, Sect. 2.10.2.) The RF/IF attenuation bytes are fetched from the RAGC or RAGCK address in R1 and converted to DIO commands by dividing them by 4 and taking the XOR with F (hex). The commands are put into the DIO table and remain unchanged for the pulse set. The two receiver channels are always attenuated equally. The bandwidth command for both channels is 3, meaning 3Ø kHz. The calibration attenuation command is set:

```
DIO.CALA = (CALAT/8).XOR.X'F'
```

The transmitter attenuation command is set:

```
DIO.TXAT = TXMAT.XOR.X'1F'
```

NXTF.PØ and NXTK.PØ both execute the NXTFOK routine, then exit through NXTF.EX, where PTYPE is incremented to become 1.

2.9.1.3. Setup for pulse types 1-3 (NXTF.P1, NXTF.P2, NXTF.P3)

Refer to Figure 2.11, connector A.

The same pulse routine is executed for I- and K-mode pulse types 1 through 3. For these pulses, only the frequency and antenna commands change in the DIO table; all other DIO commands remain as they were set for pulse type Ø, in Product One. Therefore a call is made to STFRQ, then STANT, and the NXTF.EX exit is taken.

2.9.1.4. Common pulse routine exit (NXTF.EX)

Refer to Figure 2.11, connector C.

All pulse routines for pulse types -1 through 3 exit NXTFRQ through NXTF.EX. NXTF.EX restores the pipeline registers, PULSE (R5)-R15, from SREGS, adds 1 to PTYPE and tests the result. if PTYPE < NPTYPE, the pulse set is still in progress from the point of view of NXTFRQ, and the return from NXTFRQ is taken through R15. If PTYPE = NPTYPE, NXTFRQ has completed the pulse set, and MODE is tested. If MODE = Ø (I), PTYPE is cleared to Ø; and pulse routine NXTF.PØ is executed the next time NXTFRQ is called, starting the next pulse set. For K-mode, the NXTFRQ return is taken with PTYPE = NPTYPE, or 4, and NXTK.P4 (Sect. 2.9.2.2) is executed next to begin the NOP fill sequence.

2.9.1.5. Sounding end (FIXMAX)

Refer to Figure 2.10, connector C.

Control comes to FIXMAX whether a sounding terminates normally or is aborted. If the sounding was aborted (CICT.EFIX = Ø), NXTACT (Sect. 2.7) is set to A(GETADV) to put SOUNDER into the IMA loop (Sect. 2.7.1); in case it is -1, NFILL is cleared to Ø to suppress its display by STTSG (Sect. 2.7.2.3). At SNDEND, CICT.EFIX is set to NXTFIX, the actual final frequency index of the sounding; CICT.EFRQ is also set from NXTFIX. CICT.ETS is set to ET.SEC and is the end time in seconds from 1/1/77, although not necessarily to the nearest second. The pipeline registers are restored from SREGS, and GO is set to -1 to indicate that SOUNDER is between soundings. Finally, before returning from NXTFRQ, FIXMAX tests PULSE. If PULSE = 1, ABORT was issued during the sounding countdown, and NXTFIX is set to -1 to signal SETEX (Sect. 2.7.2.13) that the pipeline is empty.

2.9.2. Kinesonde mode pulse setup and NOP fill (KMODES).

KMODES checks the sounding condition by copying GO into R2, and if GO is -1 or Ø, exits to NEWGO (Sect. 2.9.1). If GO = 1, KMODES goes through a sequence identical to that described for IMODE in Sect. 2.9.1, except that the pulse routine address is taken from KNXTJT, the K-mode jump table. (Only the routines for pulse types Ø and 4 are unique to K-mode; they are discussed below.) If GO = 8ØØØ (hex), the K-mode NOP fill sequence is in progress. (See commands KFIL and KSET, Chapter 8.)

At CKNOP the NOP fill count, NFILL, is decremented by 1, and if NFILL remains positive, KMODES exits through NEWGO, and the NOP fill sequence continues with GO = 8ØØØ (hex). If NFILL is no longer positive, either the sounding was aborted (NSET < 1) and control goes to FIXMAX (Sect. 2.9.1.5), or the NOP fill sequence has ended normally. In the latter case the pipeline registers are restored from SREGS, PTYPE is cleared to Ø to start the next pulse set, and GO is set to 1; control goes through NXTFRQ to NXTK.PØ, and the next pulse set is begun.

2.9.2.1. K-mode pulse type Ø setup (NXTK.PØ)

Refer to Figure 2.10, connector E.

The current KMODE table entry address, FKAD, is loaded into R7. FKAD is initialized by NXTIONO (Sect. 2.7.2.1) for a KRUN and by OVKVU (Sect. 2.13.2) for a VUF sounding. During a K-mode sounding, FKAD remains constant for a VUF and is updated to point to the next used KMODE entry for a new pulse set by NXTK.P4 (Sect. 2.9.2.2) for a KRUN sounding. Using R7 as an index, NXTK.PØ loads the frequency index from the KMODE table entry into R14. R7 is then set to the address of the appropriate RF/IF attenuation byte pair in RAGCK:

$$R7 = (R7 - (A(KMODE) + 2))/3 + A(RAGCK)$$

Each KMODE entry is 6 bytes long; the first KMODE entry is at KMODE + 2. Subtracting A(KMODE) + 2 from the current KMODE table entry address (R7) gives the KMODE entry number times 6. Dividing this result by 3 gives the entry's corresponding displacement into the RAGCK table. (This calculation would be much simpler if the K-mode RF/IF attenuation bytes were stored as part of the KMODE table entires, Sect. 7.4. However, command RGAIN writes the receiver gain profile tables RAGC and RAGCK to disc as one contiguous block.) With R7 and R14 thus set up, control goes to NXTFOK (Sect. 2.9.1.2.1).

2.9.2.2. Start K-mode NOP fill sequence (NXTK.P4)

Refer to Figure 2.11, connector B.

Control comes to NXTK.P4 at the end of each K-mode pulse set. The number of sets of K-mode pulse sets, NSET, initialized by NXTIONO (Sect. 2.7.2.1), is decremented by 1 at the end of each VUF pulse set and at the end of each pass through the KMODE table for a KRUN sounding. In either case when NSET goes to Ø the K-mode sounding is complete, and control goes to FIXMAX (Sect. 2.9.1.5); the NOP fill sequence is not executed following the final K-mode pulse set. So long as NSET remains positive, however, the NOP fill count, NFILL, is set to FILLV or FILLK for a VUF or a KRUN sounding, respectively. FILLV and FILLK are set by commands VFIL and KFIL (Chapter 8). If the initial value of NFILL is Ø, control goes back to NXTFRQ with PTYPE cleared to Ø and GO still set to 1; NXTFRQ branches to NXTK.PØ (Sect. 2.9.2.1) and the next pulse set is started with no NOP fill between pulse sets. If NFILL is positive, GO is set to 8ØØØ (hex) and NXTK.P4 returns from NXTFRQ; STTSG (Sect. 2.7.2.3) will issue NOP starts to the TSG to mark time, and subsequent calls to NXTFRQ will execute at CKNOP until NFILL is counted down to Ø and the next pulse set begun (Sect. 2.9.2).

At the end of each KRUN pulse set, NXTK.P4 updates the KMODE table entry address in FKAD to point to the next KMODE table entry (Sect. 7.4). If the frequency index in the next entry is -1, it has not been set by the F command, and the next entry is checked. If the entry frequency index is positive, it is passed through subroutine MAPFIX (Sect. 2.10.10) to verify that it is unprotected. This test is redundant, since K-mode frequencies are checked against FRQMAP when they are entered by command F (Sect. 2.13.4). The MAPFIX call by NXTK.P4 verifies that the integrity of the KMODE table has not been lost due to some spurious event between or during soundings. A KMODE table entry of Ø signals the end of the table or the end of a KRUN set of pulse sets. NSET is decremented as described above, and KMODE is scanned from the top for its first valid entry, to which FKAD is set. If somehow the KMODE table has been totally destroyed, NSET will be counted down to Ø, and no sounding will take place.

2.9.3. Select appropriate frequency and antenna pair (STFRQ, STANT)

Refer to Figure 2.11.

Subroutine STFRQ and STANT are called by I- and K-mode pulse routines for pulse types Ø - 3, to set the frequency and antenna commands in the DIO table (Sect. 2.12.8). In Product One, only these commands are changed after the first pulse of a pulse set; attenuation and bandwidth settings are not altered within a pulse set.

2.9.3. Select appropriate frequency and antenna pair (STFRQ, STANT)

STFRQ and STANT use tables PSEQ and ANT, respectively, which are set in sounding sequence, or pulse set, order by commands PSEQ and ANTENNAS, respectively (Chapter 8). Both tables are indexed by PTYPE*2, which is saved in PTMP early on by NXTFRQ (Sect. 2.9, Method). ANT is the table of antenna DIO commands (Sect. 2.8.4.4) and is in channel 1-channel 2 order by pulse type. STANT simply loads from ANT, using PTYPE*2 (from PTMP) as the index, and puts the antenna DIO commands into the DIO table at DIO.ANC1 and DIO.ANC2. PSEQ is the pulse set sequence code table. Each of its 4 entries is broken down into 4-bit fields as follows:

BITS	DESCRIPTION
Ø - 3	N/A
4 - 7	N/A
8 - 11	ICT.TSGC - ICT.FEPC index, used to set TFIX
12 - 15	deltaF flag, used to set DELFLG

Bits Ø-7 are not assigned in Product One, but are reserved for future expansion. The TSG-FEP command index is put into TFIX by STFRQ and into PCT.DIO2 by NXTPLSE (see PCT, Sect. 7.2). Since command PSEQ limits its arguments to Ø-7, TFIX is always Ø. (When TFIX is fully implemented, STTSG will get the TSG command byte from CICT.TSGC indexed by TFIX*2, and different TSG scenarios may be selected for different pulses of a pulse set.) The deltaF flag is put into DELFLG by STFRQ and into PCT.DIO2 by NXTPLSE. To the base sounding frequency computed by NXTFOK (Sect. 2.9.1.2.1) and saved in NXTFREQ for the pulse set, STFRQ adds or subtracts Ø through 3 times the delta frequency, depending on DELFLG. The delta frequency comes from CICT.DELF and is set by command DELF. The resulting sounding frequency, in hundreds of Hz, is converted to BCD by subroutine FTOBCD (Sect. 2.11); the 24-bit BCD frequency serves as the three 8-bit DIO commands to the synthesizer, DIO.FREQ through DIO.FREQ + 2.

2.9.4. SNXTF storage

In addition to subroutine NXTFRQ and its subroutines STFRQ and STANT, module SNXTF contains the following:

MODEJT Sounding mode jump table, indexed by MODE*2 (Sect. 2.9, Method).
INXTJT I-mode pulse routine jump table, indexed by PTYPE*2 (Sect. 2.9.1).
KNXTJT K-mode pulse routine jump table, indexed by PTYPE*2 (Sect. 2.9.2).
NXTFIX Next base frequency index for pulse set (Sect. 2.9.1.2).
NXTFREQ Next base frequency for pulse set in hundreds of Hz (Sects. 2.9.1.2.1 and 2.9.3).

NSET	Number of sets of K-mode pulse sets (passes through KMODE table for KRUN, Sect. 2.9.2.2).
NFILL	K-mode NOP fill count (Sect. 2.9.2.2).
FKAD	KMODE table entry address for NXTFRQ (Sect. 2.9.2.2).
QKAD	KMODE table entry address for QPCKR (Sect. 2.7.2.6), set to FKAD by UPAGC (Sect. 2.10.2).
FILLV	VUF NOP fill count, set by command VFIL.
FILLK	KRUN NOP fill count, set by command KFIL.
SETK	Number of sets of KRUN pulse sets, set by command KSET (see NXTIONO, Sect. 2.7.2.1).
PTMP	Temporary PTYPE*2 for STFRQ and STANT (Sect. 2.9.3).
ANT	Antenna command table (Sect. 2.9.3).
PSEQ	Pulse set sequence code table (Sect. 2.9.3).
PTBLS	Table addresses for R4-R6 (Sect. 2.9.1).
DELFLG	DeltaF flag (Sect. 2.9.3).
TFIX	ICT.TSGC-ICT.FEPC index (Sect. 2.9.3).

2.10. Local, memory-resident subroutines (SSUBS) [D.C.W.]

Module SSUBS contains subroutines that are always memory-resident and that are called by the SOUNDER task only. The global subroutines, written by J. R. Winkelman, are in module SRENT (Sect. 2.11). The SSUBS subroutines are discussed below.

2.10.1. GTAGC

Subroutine GTAGC computes the address, within table RAGC (Sect. 2.12.7), of the RF and IF attenuation settings for a given I-mode frequency, based on the hour of the day, TSG time. GTAGC is called by NXTFRQ (Sect. 2.9.1.2.1) and by the F-command overlayed advice routine, OVF (Sect. 2.13.4).

Input R14: I-mode frequency index (see NXTFRQ, Sect. 2.9).
HOUR: hour of day (Ø-23) (see CLOCK, Sect. 2.10.9).
RAGCOF: table of offsets, by time-of-day, into RAGC.
LOIX, HIIX, NRGBD: SNDR parameters (Sect. 7.6).

Output R1: address of I-mode RF/IF attenuation for frequency (R14) and time of day (HOUR).
R2-R14: returned intact using storage area DREGS.
R15: return address.

Method Refer to Figure 2.12.

The frequency band index, times 2, is computed in R1, based on R14. The index into RAGCOF is computed in R2, based on HOUR. The address of the RF/IF byte pair is then computed in R1:

2.10.1. GTAGC

R1 = R1 + RAGCOF(R2) + A(RAGC)

2.10.2. UPAGC

Subroutine UPAGC updates the Automatic Gain Control (AGC) profile tables when all data from a given pulse set are in EKO buffers. UPAGC updates table RAGC for I-mode and table RAGCK for K-mode soundings (Sect. 2.12.7).

In addition, UPAGC copies the KMODE table entry address, FKAD, used by NXTFRQ (Sect. 2.9.2.2) for the next K-mode frequency selection, to QKAD. QPCKR (Sect. 2.7.2.6) puts QKAD into bytes 8-9 of the SØ2 Q-block to identify the K-mode frequency. Thus FKAD is used at frequency selection time, and QKAD is used at pulse echo time. Because of the pipeline delay QPCKR cannot use FKAD as the K-mode frequency identifier; FKAD is updated by NXTFRQ for the next pulse set before all data have returned from the previous pulse set. QKAD is updated in this subroutine because UPAGC determines when the pulse set data have all returned; FKAD and QKAD are not involved in AGC.

UPAGC is called only by WTTSG (Sect. 2.7.2.11).

Input
- RAGCAD: circular list, set up by NXTFRQ (Sect. 2.9.1.2.1).
- PULSE: TSG ticks from sounding start (Sect. 2.7.2).
- TPEAK: total of echo peak values, from WTEKO (Sect. 2.7.2.5).
- NPEAK: number of peaks to average, from WTEKO.
- MINPK, MAXPK: FEP data peak average limits from NXTIONO (Sect. 2.7.2.1).
- RAGC, RAGCK: current AGC profile tables.
- FKAD: KMODE table entry address for pulse set.

Output
- RØ: value of PULSE when all echoes of a pulse set have returned, from RAGCAD.
- R1: A(RF/IF attenuations) in RAGC or RAGCK, from RAGCAD.
- R2-R14: returned intact using storage area SREGS.
- R15: return address.
- TPEAK, NPEAK: zeroed for next pulse set.
- RAGC, RAGCK: updated AGC profile tables.
- QKAD: K-mode frequency identifier for next 4 echoes.

Method Refer to Figure 2.12. UPAGC gets the first part of the RAGCAD entry into RØ from the top of the list. If the list is empty, no sounding is in progress, and UPAGC returns. If the pulse number in RØ equals PULSE, all data for the pulse set are in; otherwise UPAGC returns RØ to the top of list RAGCAD and returns through R15. If RØ = PULSE, UPAGC gets the attenuation address from the new top of RAGCAD into R1, saves the registers, and checks NPEAK.

If NPEAK = Ø, no good data were returned for the pulse set. Otherwise the FEP data peak average for the pulse set (TPEAK/NPEAK) is computed in R3, and TPEAK and NPEAK are zeroed for the next pulse set. The current attenuation bytes pointed to by R1 are gotten from RAGC or RAGCK:

```
LB    R4,Ø(R1)          RF, byte Ø
LB    R5,1(R1)          IF, byte 1
```

Now if R3 < MINPK or if R3 > MAXPK, the attenuation setting in R4 or R5 is adjusted according to the sequence in Table 2.5, keeping both the RF and IF in the Ø-6Ø dB range. If a gain adjustment is necessary, the updated bytes are stored back into RAGC or RAGCK, at the address in R1. Table 2.5 is in Section 2.12.7.

Finally UPAGC copies FKAD into QKAD, restores the registers, and returns.

2.10.3. ROULM and RWULM

ROULM and RWULM are called upon to put the DIO interface in the read-only or the read-write state, respectively. Calls to these subroutines are made from the SPIPE, SADVC, SSUBS, and SBFRS modules.

Input ULM: DIO interface device address, from SNDR (Sect. 7.6).

Output RØ: ULM.
R1: state command (Ø for ROULM, 1 for RWULM).
R2-R14: returned intact.
R15: return address.
DIO interface state selected as specified.

Method Refer to Figure 2.13.

The least significant bit of the interface command byte controls the data-direction state of the device. Although the two most significant bits of the same command byte control the device interrupt status, if the two MSB's are both Ø, the interrupt status is left unchanged.

Command RWCK of TEST task DIO (Sect. 10.3) checks the DIO interface RO/RW command.

2.10.4. CKLOST

CKLOST displays the number of lost echoes at the end of a K-mode run (see CKFIN, Sect. 2.7.2.13).

Input MODE: sounding mode (Sect. 2.7.2).
NLOST: number of lost echoes, from QPCKR (Sect. 2.7.2.6).

Output R2-R13: returned intact.
R14: return address, from R15.
Lost-echoes message if mode is K.

Method Refer to Figure 2.13.

2.10.5. UPADV

UPADV copies the next-advice table (NXTADV or ICT) to the current-advice table (CURADV or CICT, see Sect. 2.12.6). UPADV is called by INIT (Sect. 2.13.1) to establish the ICT constants and defaults in CICT, and by CLOCK (Sect. 2.10.9) when the sounding countdown begins.

Input NXTADV or ICT: accumulated advice, etc., for next sounding.
ADV.: length of advice tables, from STBLS.

Output CURADV or CICT: updated current advice.
R4-R14: returned intact.
R15: return address.

Method Refer to Figure 2.13.

At present only the ICT's are included in the advice tables. The advice length (ADV.) is used instead of the ICT length (ICT.) to allow for future expansion of the advice tables.

2.10.6. CKTXM

CKTXM checks the status of the high power transmitter at the start of each sounding (Sect. 2.7.2.1), sets the high order bit of CICT.STAT (Sect. 7.1), and establishes the transmitter attenuation for the sounding. If the HV dump condition is true, the TX DUMP message is displayed. (See Table 2.4, Sect. 2.8.4.3.)

Input DIO 32 transmitter flags (Table 2.4).
TXLAT: low power TX attenuation, from command TXATLP.
TXHAT: high power TX attenuation, from command TXATHP.
CICT.STAT: general system status byte, low order 7 bits.

Output TXMAT: transmitter attenuation for sounding.
CICT.STAT: high order bit on if HV on.
"TX DUMP" message, if HV dump true.
R4-R13: returned intact.
R14: return address from R15.

Method Refer to Figure 2.13. CKTXM puts the DIO interface in the read-only state with a call to ROULM, reads the transmitter flags via DIO 32, and tests the HV dump bit. If HV dump is true (Ø), a transmitter fault is indicated, and the error message is logged. CKTXM tests the HV bit. If HV off is true (Ø), TXMAT is set to TXLAT for low power sounding, and the most significant bit (MSB) of CICT.STAT is cleared to zero. If HV off is false (1), TXMAT is set to TXHAT for high power sounding, and the MSB of CICT.STAT is turned on. The low order 7 bits of CICT.STAT are preserved in either case. NXTFRQ (Sect. 2.9.1.2.1) computes the transmitter attenuator command from TXMAT and puts it into DIO.TXAT.

2.10.7. ERROR

ERROR logs error messages for the SOUNDER task and, if TEST task HELP is running in the EXPLAIN mode, passes the error information to HELP in the SFF Q-block. (See HELP task, Sect. 10.3.) If SOUNDER is not in the TEST mode, ERROR simply logs the error message and pauses the SOUNDER task to permit operator intervention to clear the error condition. The operator continues after the pause by typing:

TASK SOUNDER; CONTINUE

to the command processor, not to COUNSEL. Temporary control is given to the command processor by typing ESC (escape).

Input RØ: error code, from parameter block status, etc.
TEST: -1 if TEST task HELP is running.
R15: error call location + 4 (return link).
CONTINUE, from operator.
SFF Q-block status, if HELP running.

Output Error message, "ERROR EEEE AT NNNN", where EEEE = (RØ) and NNNN = (R15) - 4.
SFF Q-block to task HELP only.
RØ-R14: returned intact using storage area EREGS.
R15: return address.

2.10.7. ERROR

Method Refer to Figure 2.13.

ERROR saves all registers in EREGS, constructs the error message, sets up the SFF Q-block in case HELP is running, and logs the error message with WRLOG (Sect. 2.10.8). If HELP is running, TEST will have been set to -1 by OVRUNT (Sect. 2.13.8). If such is not the case, ERROR pauses the SOUNDER task at label ERPA. Otherwise, ERROR initializes the SFF Q-block status to -1ØØØ (hex) and queues HELP with the error call location (SFF + 4) and the error code (SFF + 6). ERROR now tests the SFF Q-block status field every 1ØØ ms as long as it remains negative. If the status is cleared to Ø by HELP, the operator has told HELP to continue without a pause; ERROR restores the registers from EREGS and returns. If the status is set positive by HELP, the operator wishes to intervene, and SOUNDER is paused at ERPA. SOUNDER remains in the paused state until the operator explicitly continues the task; then ERROR restores the registers and returns.

Each call to ERROR in the assembly code is labeled ERRNN, where NN is the error call code in hexadecimal. Call codes ØØ through 2E are in use in Product One. Each ERRNN label is unique and is made to be an entry point in the assembly code to get its value, or absolute loaded address, into the SOUNDER load map SYSC:SOUNDER.MAP, which is output by TET (Sect. 9.4). The HELP task reads SOUNDER.MAP and builds a table of ERROR call locations by call code, which it uses in interpreting SOUNDER errors. SOUNDER.MAP may be examined with the Interdata editor or by typing:

```
LIST  SOUNDER.MAP
```

to the command processor, without the HF Radar tasks loaded.

2.10.8. WRLOG

WRLOG logs all SOUNDER messages. The logging device is assigned to LU1 by INIT (Sect. 2.13.1) and is the system console by default for Product One. In order to log a message, SOUNDER must get COUNSEL to HALTIO on its last read request to the console (Sect. 6.1). To accomplish this, SOUNDER queues COUNSEL with the address of a half-word cleared to zero. COUNSEL cancels its last console read request giving SOUNDER temporary console control, and the message is displayed.

Input RØ: message address.
R1: message size in bytes.

Output Q-block ØØØ to COUNSEL to effect HALTIO on console read request.
Message to console.
R2-R14: returned intact using storage area WREGS.
R15: return address.

Method Refer to Figure 2.13.

2.10.9. CLOCK

Subroutine CLOCK maintains the HF Radar system time-of-day (TOD) clock (ET.SEC), the count of seconds remaining until the next sounding (ET.TTNI), and the 1-19 second countdown word that is displayed (ET.CDN). ET.SEC contains elapsed seconds from 1 Jan 1977 and is set by RDTIME (Sect. 2.13.10) when a sounding is initiated by GO, KRUN, or VUF (Sect. 2.13.2); ET.TTNI is initialized in SPIPE and is based on ICT.NIPH; ET.CDN is -1 unless a sounding countdown is in progress. (See Sect. 2.12.5.) CLOCK is called by WTTSG (Sect. 2.7.2.11) in the SPC loop only (Sect. 2.7.2), just after the TSG DONE interrupt is sensed. The value of the TSG tick is based upon the TSG pace code and the repeat code, both represented by two bits in the TSG command byte, CICT.TSGC.

CLOCK uses TIKSEC, the table of TSG interrupts per second indexed by the pace code times 2 (Sect. 2.7.2.3), and TSGREP, the TSG scenario repeat count table indexed by the repeat code (Sect. 2.7.2.3), to compute the value of the TSG tick. TIKSEC is assembled according to how the TSG pace codes are strapped and must be updated and reassembled if the pace code straps are changed. TSGREP is set from bytes 16-19 (hex) of TSG page Ø by BTTSG whenever the TSG is bootstrapped.

The path through CLOCK is quite short unless the beginning of the sounding countdown is sensed. At that time CLOCK performs the sounding initialization that must be completed before the SØ4 queue to PICKER, then sends the SØ4, or sounding-started, Q-block to PICKER. (The SØ6, or sounding-ended, Q-block is sent by SETEX in SPIPE, Sect. 2.7.2.13.)

Input FRSEC: fractional second in TSG ticks, set by RDTIME.
CICT.TSGC: current TSG command byte.
TIKSEC: table of TSG interrupts/second.
TSGREP: table of TSG repeat counts.
ET.TTNI: time to next sounding in seconds, set in SPIPE.
NXTFIX: next frequency index, set to -1 by WTFEP (Sect. 2.7.2.7) or NXTFRQ (Sect. 2.9.1.5) to mean no sounding in progress.

2.10.9. CLOCK

ICT.NIPH: Ø if sounding in free run mode.
CDNVAL: countdown value, set by command CDN.
ET.CDN: countdown remaining, set from CDNVAL, or -1.
ET.SEC: TOD seconds, set by RDTIME.
HRSEC: full seconds into hour, set by RDTIME.
HOUR: hour of day, set by RDTIME.
SØ6 Q-block status: initialized to -1ØØØ (hex) by SETEX, cleared to Ø by other tasks when post-sounding activity is completed.
ICT.INUM: number of last sounding.
ICT.EFIX: set by command EFRQ or -1.
ICT.EFRD: used to set ICT.EFIX if ICT.EFIX = -1.
ICT.SFIX: set by command SFRQ or -1.
ICT.SFRD: used to set ICT.SFIX if ICT.SFIX = -1.
FRQMAP: protected frequencies map.
MODE: sounding mode.
CICT.MODE: used to set SØ4 + 4 if MODE is -1 (TPRUN).
ICT.DFIX: frequency index increment.
NOCHN: receiver channel flags, set by commands NOCH1 and NOCH2.
NPTYPE: number of pulse types in pulse sets (see SNDR, Sect. 7.6).
ICT.M: set by command KEEP.
PULSE: TSG interrupts since sounding start or -1.

Output FRSEC: fractional second, updated.
ET.TTNI: decremented by 1 when second elapses.
ICT.NIPH: cleared to Ø (free run) if soundings scheduled too close together.
ET.CDN: set from CDNVAL when ET.TTNI = CDNVAL, then decremented by 1 each second.
ET.SEC: incremented by 1 each second.
HRSEC: updated seconds into hour.
HOUR: updated hour of day.
ICT.STS; set to ET.SEC + CDNVAL at start of countdown.
ICT.STM: cleared to Ø at countdown start.
ICT.INUM: incremented by 1 each sounding.
ICT.EFIX: set from ICT.EFRD (default end frequency) if no EFRQ command given (ICT.EFIX = -1).
ICT.SFIX: set from ICT.SFRD (default start frequency or next higher unprotected frequency) if no SFRQ command given (ICT.SFIX = -1).
ICT.SFRQ: set from ICT.SFIX.
NXTFIX: set to ICT.SFIX - ICT.DFIX if MODE = Ø (I-mode).
ICT.N: set to NPTYPE times the number of receiver channels.
ICT.M: set to ICT.N if ICT.M > ICT.N.

SVC6.PAR: queue parameter set to A(STSND), the SØ4 Q-block address.
SØ4 + 2: initialized to -1ØØØ (hex).
SØ4 + 4: set to sounding mode times 2.
SØ4 Q-block: sounding-started queue to PICKER.
Hex display of countdown: ET.CDN if ET.CDN > Ø.
R1-R14: returned intact using storage area SREGS, except that PULSE is incremented by 1 if sounding is in progress.
R15: return address.
Error messages if soundings are scheduled too close together, or if the SØ4 queue fails.

Method Refer to Figure 2.14.

When entered, CLOCK saves R1-R15 in SREGS, then loads R4-R7 with A(SSVC6), A(ICT), A(CICT), and A(ET), respectively. R4-R7 are equated to S., I., C., and E., respectively, in module SSUBS.

The TSG repeat code is extracted from the TSG command byte, CICT.TSGC; the repeat count, from table TSGREP indexed by the repeat code, is added to FRSEC. The TSG pace code is extracted from CICT.TSGC and multiplied by 2 to index table TIKSEC. The appropriate TIKSEC entry is subtracted from the new TSG interrupt count, saved in FRSEC, to see if a full second has elapsed.

If the result of the subtraction is negative, a second's worth of TSG ticks or interrupts has not been accumulated, and control goes to the CLOCK exit routine, CLKEX.

If a second has elapsed, FRSEC is cleared to Ø, and ET.SEC, a 32-bit counter, is incremented by 1; HOUR, used by GTAGC (Sect. 2.10.1), is updated if 36ØØ seconds have been accumulated in HRSEC; ET.TTNI is decremented by 1.

ET.CDN is tested; a positive result means that a countdown to sounding is in progress, and control goes to CNTDN. Otherwise ET.TTNI is compared to CDNVAL; if they are not equal, it is not time for a countdown to begin, and control goes to CLKEX. If ET.TTNI = CDNVAL, it is time to begin the countdown as scheduled.

If soundings have not been scheduled too close together, NXTFIX will be negative and the SØ6 Q-block status will be cleared to Ø, indicating that both sounding and post-sounding activity have been completed, and a new sounding may begin. If such is the case, control goes to STCDN. Otherwise ICT.NIPH is tested at CKFRUN.

2.10.9. CLOCK

If ICT.NIPH = Ø, CLOCK has defaulted to the free-run sounding mode or the command NIPH 999 has been given. In either case the countdown is deferred by 1 second by adding 1 to ET.TTNI at DFRCDN. If ICT.NIPH is not Ø, a sounding scheduling error has occurred, and an error message is logged. If the operator continues through the error (Sect. 2.10.7), CLOCK defaults to free run by clearing ICT.NIPH to Ø and exits via DFRCDN.

At STCDN, pre-sounding initializing begins. The protected frequencies map is checked with a call to CKMAP, with call code 2 (Sect. 2.10.11). ET.CDN is set to CDNVAL + 1; ET.CDN is decremented at CNTDN before it is displayed. ICT.STS is set to ET.SEC + CDNVAL and is the time that the sounding actually begins after the countdown. ICT.STM is zeroed; soundings always begin on the second. The sounding is counted by adding 1 to ICT.INUM, which is checked against CICT.NI by CKFIN (Sect. 2.7.2.13). ICT.EFIX is tested. If it is -1, it is set from ICT.EFRD; otherwise it was set for this sounding only by command EFRQ. ICT.SFIX is likewise tested and set to the default, ICT.SFRD, if necessary, but is then passed through MAPF (Sect. 2.10.10) to check it against FRQMAP. ICT.EFIX is not checked at this point, but is set, with ICT.EFRQ, to the actual final base frequency of the sounding by NXTFRQ (Sect. 2.9.1.5). The final value of ICT.EFIX may or may not be that specified, depending upon FRQMAP and whether the sounding is aborted.

.MODE is now tested. If it is -1 (TPRUN), the sounding mode flag, to be put into SØ4 + 4, is taken from CICT.MODE as read from magtape by OVTPR (Sect. 2.13.3), and control goes to STPR. If MODE = Ø (I-mode), NXTFIX is set to ICT.SFIX - ICT.DFIX, to allow for the DFIX increment the first time through NXTFRQ (Sect. 2.9.1.2). (NXTFIX is initialized by NXTFRQ for a K-mode sounding, Sect. 2.9.2.2.) ICT.N is set to NPTYPE (4) times the number of receiver channels specified by NOCHN (Sect. 2.8.8), normally 2, resulting in ICT.N = 8, unless command NOCH1 or NOCH2 has been given. (See WIEKO, Sect. 2.7.2.5.) At this point all pre-sounding initialization that is required before the SØ4 queue is complete, and the next-advice becomes current with a call to UPADV (Sect. 2.10.5). ICT.SFIX and ICT.EFIX are set to -1, and thus commands SFRQ and EFRQ affect one sounding only.

At STPR the SØ4 queue is set up: SVC6.PAR is set to A(STSND), the SØ4 Q-block address; STSND + 2, the SØ4 status field, is set to -1ØØØ (hex); STSND + 4 is set to the sounding mode times 2. The SØ4 Q-block is then sent to PICKER, which in turn passes it on to MANAGER (Sect. 7.7).

During the sounding countdown, that is, as long as ET.CDN > Ø, ET.CDN is written to the hex display panel lights. If ET.CDN > 9, 6 is added to the number displayed to make it appear decimal. ET.CDN is displayed only when a second elapses.

At CLKEX R1-R15 are restored from SREGS, and if a sounding is in progress, 1 is added to the PULSE register before CLOCK returns through R15.

2.10.10. MAPFIX and MAPF

MAPFIX or MAPF is called to check a frequency index against FRQMAP, the protected frequencies map (Sect. 2.12.9). If the frequency is known to be within the range of the synthesizer, LOIX through HIIX (see SNDR, Sect. 7.6), MAPF, the faster of the two subroutines, is called. If a range check is needed, MAPFIX must be called.

Input R14: frequency index to be tested.
FRQMAP: the protected frequencies map.
LOIX: lowest useable frequency index.
HIIX: highest useable frequency index (MAPFIX only).

Output RØ: mask for extracting the single bit in FRQMAP that corresponds to the frequency index in R14.
R1: index to byte of FRQMAP that contains the mapped bit.
R13: -1 if frequency out of range LOIX through HIIX (MAPFIX only); Ø if frequency protected; 1 if frequency not protected.
Condition code (CC): set from R13.
R2-R12, R14: unchanged.
R15: return address.

Method Refer to Figure 2.15.

The first bit in FRQMAP represents frequency index LOIX; 8 frequencies are mapped in a byte. The integer FRQMAP byte index is computed:

R1 = (R14 - LOIX)/8

The mask in RØ is initialized to X'8Ø', then right shifted into position. The right shift count is the low order 3 bits of (R14 - LOIX), or the difference modulo 8. The FRQMAP byte is loaded into R13, then R13 and RØ are logically ANDed into R13, setting the CC.

2.10.11. CKMAP

CKMAP performs a two-fold check on FRQMAP, the protected frequencies map (Sect. 2.12.9): the exclusive OR sum of all FRQMAP bytes is computed, and a population count is made of all 1-bits in the map. The XOR checksum and population count check bytes are stored with the maps on disc and in memory. (See TEST task FRMAP, Sect. 10.3.)

If CKMAP is called with RØ = Ø, the check bytes are only set in memory; the old values are ignored. if RØ > Ø, the newly computed values are compared to the old. A discrepancy indicates that FRQMAP has been altered by an errant routine or operator, and SOUNDER stops dead. ADJMAP (Sect. 2.13.9) calls CKMAP with RØ = Ø, to set the check bytes for a base map newly adjusted to DELF and PSEQ. GETMAP (Sect. 2.13.9) calls CKMAP with RØ = 1 to verify the base map read from disc. CLOCK (Sect. 2.10.9) calls CKMAP with RØ = 2 to verify that the adjusted map has not been altered in memory.

Input RØ: call code, Ø means set check bytes.
FRQMAP: protected frequencies map, either the base map from disc or adjusted to DELF and PSEQ.
MAPSZ: size of map, based on LOIX and HIIX (see SNDR, Sect. 7.6).

Output New check bytes.
RØ-R14: returned intact, using storage area EREGS.
R15: return address.

Method Refer to Figure 2.15.

2.10.12. SSUBS storage

In addition to the subroutines described above, module SSUBS contains the following:

TXMER	"TX DUMP ***" message (Sect. 2.10.6).
TXML	Length in bytes of TXMER.
LSTM	"NN LOST ECHOS." message (Sect. 2.10.4).
LSTL	Length of LSTM.
UNDPB	Parameter block for converting NLOST to ASCII "NN" in message LSTM.
ERRMSG	"ERROR EEEE AT NNNN." message (Sect. 2.10.7).
ERRML	ERRMSG length.
UNPKERR	Convert error code to "EEEE" in ERRMSG.

UNPKLOC	Convert error call location to "NNNN" in ERRMSG.
PAWSBP	Pause SOUNDER task (Sect. 2.10.7).
SFF	SFF Q-block to TEST task HELP (Sect. 2.10.7).
EREGS	ERROR routine register storage.
WRLPB	Parameter block for logging message (Sect. 2.10.8).
WREGS	WRLOG routine register storage.
Q.ID	Storage for SVC6.ID (Sect. 2.10.8).
C.ID	COUNSEL task ID (Sect. 2.10.8).
RAGCAD	Receiver AGC circular list (Sect. 2.10.2).
RAGCOF	Receiver AGC time-of-day tables offset list (Sect. 2.10.1).
TPEAK	FEP data peak total for pulse set (Sect. 2.10.2).
NPEAK	Number of peaks for average (Sect. 2.10.2).
MINPK	Minimum echo peak average in ADC counts (Sects. 2.10.2, 2.7.2.1).
MAXPK	Maximum echo peak average in ADC counts (Sects. 2.10.2, 2.7.2.1).
PKMIN	Minimum peak average in mV, set by command PKLO.
PKMAX	Maximum peak average in mV, set by command PKHI.
TSGREP	TSG scenario repeat count table (Sect. 2.10.9).
TIKSEC	Table of TSG interrupts/second (Sect. 2.10.9).
ZERO	Queue parameter for COUNSEL HALTIO (Sect. 2.10.8).
ONE	32-bit 1 for incrementing ET.SEC (Sect. 2.10.9).
PTBLS	Table addresses for CLOCK (Sect. 2.10.9).
FRSEC	Fractional TOD second (Sect. 2.10.9).
HRSEC	Full seconds into hour (Sect. 2.10.9).
HOUR	Hour of day (Sects. 2.10.9, 2.10.1).
CDNVAL	Countdown value, set by command CDN, used by CLOCK (Sect. 2.10.9).

2.11. Reentrant subroutines [J.R.W.]

A number of reentrant subroutines are available to the user. These subroutines keep all data in registers and do not modify memory so that they may be used simultaneously by several tasks.

The locations of the subroutines are given in the Environment Table. For example, to call NTOFR, assuming the location of ET is in R3:

```
LH    RA,ET.ANTFR(R3)
BALR  RF,RA
```

The subroutines are designed more for speed than accuracy. Typical accuracy is Ø.1%, more than enough for plotting and quick look.

2.11.1. NTOFR

Purpose Frequency index to frequency.

Address ET.ANTFR

Registers RØ-R7: unchanged.
(R8,9): used.
(RA,B): 8-digit frequency in BCD, Hz.
(RC,D): frequency/1ØØ, Hz.
RE: input frequency index, unchanged.
RF: return address.

Method Divide frequency index by 4ØØ giving quotient L and remainder NI, Ø ≤ NI < 4ØØ.

$$NF = ((13*NI + 14788)*NI/8192 + 2225)*NI/8 + 16ØØØØ$$

$$FREQ = NF*2**(L-8)$$

Internally, the frequency index is used instead of the frequency. This allows frequency to be specified as 12 bits and the logarithmic scanning is automatic.

Processing the data requires that the user be able to determine the exact frequency used in the sounding. That frequency is called F, and the sounding made at F + deltaF differs from F by exactly deltaF. There is not an index associated with F + deltaF. The index determines F and a constant deltaF is added to F.

The computer algorithm to calculate F from frequency index NI is given below in two versions. The algorithm was chosen for interchangeability among Fortrans rather than absolute logarithmic accuracy, although it is quite accurate.

Two Fortran versions of NTOFR follow. The first uses double precision floating point integers (Interdata) and the second is for computers with three or more bytes for integer arithmetic (≥24 bits).

```
      FUNCTION FREQ(NI)
C          FREQUENCY INDEX (NI) TO FREQUENCY IN 1ØØ'S OF HZ.
C          9ØØØØØØ. IS MORE THAN HALF THE LARGEST FLOATING POINT
C          INTEGER.  9ØØØØØØ. FORCES TRUNCATION OF THE CALCULATIONS TO
C          INTEGER.
C          ON COMPUTERS THAT ROUND (INTERDATAS WITH HARDWARE DOUBLE
C          PRECISION), SUBTRACT Ø.5 FROM THE LARGE ADDITIVE CONSTANTS.
C          JRW 12/Ø8/78.
```

```
      L = NI/400
      N = NI - 400*L
C          BREAK INTO 9 OCTAVES WITH 400 STEPS IN EACH.
      F = 13*N + 14788
      A = N
          A*F/8192. + 9002225.) - 9000000.
      F = (A*F/8. + 9160000.) - 9000000.
      IF(L.EQ.8) GO TO 2
      L1 = L + 1
      DO 1 J = L1,8
 1    F = F/2.
 2    FREQ = (F + 9000000.) - 9000000.
      RETURN
      END

      FUNCTION FREQ(NI)
C          FREQUENCY INDEX (NI) TO FREQUENCY IN 100'S OF HZ FOR COM-
C          PUTERS WITH 24 OR MORE BITS FOR INTEGER ARITHMETIC.
      L = NI/400
      N = NI - 400*L
      M = 13*N + 14788
      M = (N*M)/8192 + 2225
      M = (N*M)/8 + 160000
      IF(L.EQ.8) GO TO 2
      L1 = L + 1
      DO 1 J = L1,8
 1    M = M/2
 2    FREQ = M
      RETURN
      END
```

2.11.2. FTOBCD

Purpose Frequency to BCD, contained in NTOFR.

Address ET.AFRTB

Registers R0-R7: unchanged.
(R8,9): used.
(RA,B): 8-digit frequency in BCD, Hz.
(RC,D): input frequency/100, Hz.
RE: unused.
RF: return address.

2.11.2. FTOBCD

Method Repeatedly divide the frequency by 10 and pack the 4-bit remainders into a full word.

2.11.3. FRTON

Purpose Frequency to frequency index. This is not an exact inverse of NTOFR, but is close enough for frequencies input to SOUNDER and GRAPHER.

Address ET.AFRTN

Registers R0-R9: unchanged.
RD: frequency index, output.
(RA,B): used.
(RC,D): frequency/100, changed.
RE: used.
RF: return address.

Method The frequency NF is normalized to 5000 (X). The number of shifts subtracted from 9 becomes the octave.

$X = NF/5000,\ 0.5 \leq X < 1.$

$FI = 400*\text{octave} + ((491*X - 1652)*X + 2419)*X - 857$

2.11.4. LOG

Purpose Logarithm base 10 (for dB).

Address ET.ALOGF

Registers R0-R9: unchanged.
(RA,B): used.
RC: log (base 10) * 2048.
RD: input number, changed.
RF: return address.

Method The number X is normalized. The number of shifts subtracted from 14 becomes the characteristic of log base 2.

$8192 \leq X < 16384$

LOG2 = 23873*(X - 11585)/(4*(X + 11585)) + 1024
+ characteristic*2048

LOG10 = 0.30103*LOG2

2.11.5. PHAMP

Purpose Calculate phase and amplitude from East and North components.

Address ET.APAMP

Registers
R8: amplitude, same units as RC and RD.
R9: phase*4096, radians.
(RA,B): used.
RC: East component, changed.
RD: North component, changed.
RE: used.
RF: return address.

Method The angle is reduced to the first octant (0-pi/4). The absolute values of RC and RD are taken and the registers exchanged if necessary so that:

$RC \geq RD \geq 0$

$T = \tan PHI = RD/RC, \quad 0 \leq T \leq 1$

PHI = T*16384/(18150*T*T/4 + 16384)

A = RC*(1 + 5*PHI*tanPHI/8 - PHI*PHI/8)

arctan(RD/RC) = PHI adjusted to the proper octant

Given the definitions and conditions above, A = RC*secPHI. Consider the following series expansions that hold for PHI in the first octant:

secPHI = 1 + PHI**2/2 + 5*PHI**4/24 + 61*PHI**6/720 +
277*PHI**8/8064 + 50521*PHI**10/3628800 + ...

PHI*tanPHI = PHI**2 + PHI**4/3 + 2*PHI**6/15 +
17*PHI**8/315 + 62*PHI**10/2835 + ...

2.11.5. PHAMP

Truncating both series at PHI**4 we have:

```
5*PHI*tanPHI/8 = 5*PHI**2/8 + 5*PHI**4/24
```

Subtracting PHI**2/8 from both sides we obtain:

```
5*PHI*tanPHI/8 - PHI**2/8 = secPHI - 1
```

Therefore:

```
secPHI = 1 + (5*PHI*tanPHI - PHI**2)/8 + R
```

where R is an error term in PHI**6 and higher powers. Multiplying by RC and ignoring the error term yields the equation for A above.

Consider the error term:

```
R = secPHI - 5*PHI*tanPHI/8 - 1 + PHI**2/8
```

Substituting the series expressions gives:

```
R = (PHI**6)*(61/720 - 2/24) + (PHI**8)*(1385/40320 -
    17/504) + (PHI**10)*(50521/3628800 - 31/2268) + ...
  = PHI**6/720 + 5*PHI**8/8064 + 307*PHI**10/1209600 + ...
```

The maximum allowable PHI is pi/4 so that:

```
R(pi/4) = RMAX = 0.000326 + 0.000090 + 0.000023 + ...
        = c. 0.00044
```

2.12. SOUNDER and HF Radar global tables (STBLS) [D.C.W.]

Module STBLS contains the major tables, circular lists, etc., used by the SOUNDER task and all of the tables used globally by the HF Radar tasks. The global tables are described in detail in Chapter 7. For a detailed exaplanation of circular lists, see the 16-Bit Processor User's Manual, Chapter 3, section on List Processing.

Module STBLS contains no executable code.

2.12.1. PCT/EKO buffer circular lists (PCTLST, SØ2LST)

The six PCT/EKO buffer areas (Fig. 2.2) in module SBFRS are allocated for use by SPIPE in round-robin fashion, i.e., 1, 2, 3, 4, 5, 6, 1, 2,... for a normal sounding, or 1, 2, 3, 4, 1, 2,... for a TEST-mode sounding. Each PCT (Sect. 7.2) is associated with one EKO buffer; each SØ2 Q-block in SØ2TBL (Sect. 2.12.2) is associated with one PCT/EKO buffer area.

NXTPLSE (Sect. 2.7.2.2) gets a PCT address from the top of PCTLST, and returns that address to the bottom of PCTLST, just before a pulse is transmitted at STTSG. One time period later, when the FEP processed echo from the transmitted pulse is ready to be transferred to SOUNDER, GETBFR (Sect. 2.7.2.9) gets the address of an SØ2 Q-block from the top of SØ2LST, returns it to the bottom, and from the SØ2 Q-block gets the address of the EKO buffer associated with the PCT above. Thus PCTLST and SØ2LST circulate at the same rate with SØ2LST lagging PCTLST by 1 time period.

PCTLST is initialized as follows:

DISPLACEMENT	BYTES	DESCRIPTION
+Ø	1	NEKOBFS(6) number of slots
+1	1	NEKOBFS(6) number used
+2	1	Ø current top
+3	1	Ø next bottom
+4	2	A(PCT 1)
+6	2	A(PCT 2)
+8	2	A(PCT 3)
+1Ø	2	A(PCT 4)
+12	2	A(PCT 5)
+14	2	A(PCT 6)

SØ2LST is initialized as follows:

DISPLACEMENT	BYTES	DESCRIPTION
Ø	1	NEKOBFS(6) number of slots
+1	1	NEKOBFS(6) number of slots
+2	1	Ø current top
+3	1	Ø next bottom
+4	2	SØ2TBL A(SØ2 1)
+6	2	SØ2TBL+1Ø A(SØ2 2)
+8	2	SØ2TBL+2Ø A(SØ2 3)
+10	2	SØ2TBL+3Ø A(SØ2 4)
+12	2	SØ2TBL+4Ø A(SØ2 5)
+14	2	SØ2TBL+5Ø A(SØ2 6)

2.12.1. PCT/EKO buffer circular lists (PCTLST, SØ2LST)

NEKOBFS is a SNDR parameter (Sect. 7.6). Bytes Ø and 1 are set for normal and TEST mode to 6 (NEKOBFS) and 4, respectively, by OVRUNT and ENDTEST (Sect. 2.13.8) at OVTCOM (Fig. 2.24).

2.12.2. SØ2 queue block table (SØ2TBL)

SØ2TBL contains the 6 SØ2 Q-blocks associated with PCT/EKO buffers 1-6. The addresses of the Q-blocks, in order, are in SØ2LST, above.

DISPLACEMENT	BYTES	DESCRIPTION
+Ø	1Ø	SØ2 1
+1Ø	1Ø	SØ2 2
+2Ø	1Ø	SØ2 3
+3Ø	1Ø	SØ2 4
+4Ø	1Ø	SØ2 5
+5Ø	1Ø	SØ2 6

Each SØ2 Q-block has the following format:

DISPLACEMENT	BYTES	DESCRIPTION
+Ø	1	Always 2
+1	1	Always "S"
+2	2	Status field
+4	2	Address of EKO buffer
+6	2	Address of associated PCT
+8	2	Address of KMODE table entry or Ø

The status is set to -1ØØØ (hex) by QPCKR (Sect. 2.7.2.6) and is tested for Ø by GETBFR (Sect. 2.7.2.9). The KMODE table entry address (Sect. 7.4) is set by QPCKR to the contents of QKAD. All other SØ2 fields remain constant.

An additional dummy SØ2 Q-block, S2LOST, is used by QPCKR if data are lost during a K-mode run.

2.12.3. TSG range zero correction table (TSGDLY)

TSGDLY is a table of TSG range zero corrections, or delays, and is indexed by scenario number times two. Each time the TSG is bootstrapped, the TSG delays are computed for scenarios 3-7. Scenarios Ø-2 currently have no delays associated with them; their entries in TSGDLY are set to zero.

The delays in microseconds are computed from the center of the TX KEY pulse (with TX ENBL 1 active) to the beginning of the FEP write enable (FEP WR ENBL) window. (See TEST task SNRIO display, Sect. 10.3.) Subroutine BTTSG (Sect. 2.13.5.2) does the delay computations using the TSG page Ø scenarios. (See command BOOT, Chapter 8.)

The appropriate value from TSGDLY is put into ICT.TDLY by BTTSG; ICT.TDLY is updated by the TSGC command and enters into PICKER's calculations of echo ranges (Sect. 3.4).

2.12.4. Kinesonde mode table (KMODE)

KMODE, a global table, is defined in Sect. 7.4. The frequency index is set or deleted by command F by entry number (Ø-9), and may be deleted as a result of command DELF or PSEQ (Sect. 2.8.4.5). Ranges are set by commands RN and RF. The number of KMODE entries, NKFRQ, is parameterized in SNDR (Sect. 7.6).

During a K-mode sounding, the KMODE entry address for echoes of the current pulse set, kept in QKAD, is put into bytes 8 and 9 of the SØ2 Q-block to pass the range limits to PICKER.

When a KMODE entry is set or deleted by commands F, RN, or RF, SOUNDER sends GRAPHER the SØ8 Q-block (Sect. 5.4).

2.12.5. Environment table (ET)

The ET is defined in Sect. 7.3. ET.SEC is set by subroutine RDTIME when the TSG clock is read at sounding initiation (GO, KRUN, or VUF). As long as SOUNDER is in the SPC loop (Sect. 2.7.2), subroutine CLOCK adds 1 to ET.SEC each second. CLOCK sets ICT.STS from ET.SEC when a sounding countdown begins (Sect. 2.10.9); NXTFRQ copies ET.SEC to CICT.ETS when a sounding ends (Sect. 2.9.1.5).

ET.CDN is set from CDNVAL, which in turn is set by command CDN, and is decremented by 1 each second during a countdown by CLOCK. When ET.CDN goes to Ø, NXTIONO (Sect. 2.7.2.1) sets it to -1 and initializes for a sounding. ABORT (Sect. 2.8.4.2) tests ET.CDN to see if a countdown is in progress.

2.12.5. Environment table (ET)

ET.TTNI is set at the end or at the start of a sounding depending upon ICT.NIPH. If free run is selected (ICT.NIPH = Ø, from command NIPH 999), ET.TTNI is set to CDNVAL + 1 at CKFIN (Sect. 2.7.2.13), and sequential soundings run back to back. If free run is not selected (Ø < NIPH < 256), ET.TTNI is set to 36ØØ/NIPH by NXTIONO, and the next sounding runs when scheduled. CLOCK subtracts 1 from ET.TTNI each time one second elapses, as long as the SPC loop is executing. When ET.TTNI = CDNVAL, the countdown begins.

ET.ABUF and ET.LBUF are computed and set by INIT (Sect. 2.13.1). INIT also sets ET.SCAN to point to SOUNDER's UDL (SUDL) and command scan table (SMNEM), but COUNSEL later sets ET.SCAN to its own, complete, UDL/SCAN table.

The ET.HKADC sub-table is filled one entry per time period by HKADC (Sect. 2.7.2.12) in the SPC loop.

The logging device and all table (see reference to ET.AICT, Sect. 2.12.6) and routine addresses in ET are assembly constants. This information is made available to the other tasks by INIT putting the ET address into bytes 4 and 5 of COUNSEL's CØØ Q-block at SOUNDER start up.

2.12.6. Ionogram configuration tables (ICT, CICT)

The ICT/CICT table entries are defined in Sect. 7.1 because the CICT is a global table.

The original design of the HF Radar software called for running ionogram sweeps with Product One; kinesondes and other modes of sounding were planned for the future. Viewed from the present stage of development, the ICT was perhaps hastily named; and the word sounding can in most cases properly be substituted for the word ionogram. The original design further specified that during a given sounding the operator-experimenter be permitted to enter advice that should affect the next, but not the current, sounding. Consequently two copies of the ICT are maintained in STBLS.

Ionogram, or sounding, configuration commands (Sect. 2.5) modify the next-ICT, labeled ICT; the next-ICT is copied to the current-ICT, labeled CICT, by a CLOCK call to UPADV just before the SØ4 (sounding start) queue. (See Fig. 2.14, under connector C.) The CICT is used for current sounding control, and the ICT may be set up for the next sounding. However, due to OS memory constraints, the SVC2 code invoked by COUNSEL to scan operator commands is overlayed, and entering commands during a sounding results in disc accesses that may interfere with sounding timing and result in lost echoes.

2.12.6. Ionogram configuration tables (ICT, CICT)

Default sounding parameters and identifiers are assembled into ICT. However, the task revisions must be set by the other tasks after they have started running. Therefore, ET.AICT is initialized to A(ICT), and after the task revisions are set, but before sounding begins, ET.AICT is set to A(CICT). ICT is copied to CICT at SØ4 time, and the CICT is available for reference by the other tasks during sounding and is saved on disc with the sounding data by MANAGER. Thus the task revisions and the default values come to be associated with the data.

SOUNDER references to ICT/CICT entries are more numerous, but also more straightforward, than to ET entries. They are discussed in detail, however, in the various flow diagram narratives as required.

2.12.7. Receiver AGC profile tables (RAGC, RAGCK)

SOUNDER effects receiver automatic gain control (AGC) by adjusting the receiver RF and IF attenuators to keep the signal returns within optimum limits. The limits, MINPK and MAXPK, are set with commands PKLO and PKHI. (See Sects. 2.2, 2.4, and 2.7.2.1.) If the FEP data peak average for a given pulse set is not within these limits, the pulse set data are passed to PICKER, but an adjustment is made to the gain control profile table, resulting in attenuator adjustments for subsequent soundings.

For I-mode AGC an RF and an IF byte are reserved for each of 12 frequency bands in 4 time-of-day tables, all under label RAGC. The 4 tables are for time intervals 2100-0259 (RAGCØ), Ø300-0859 (RAGC1), Ø900-1459 (RAGC2), and 1500 - 2059 (RAGC3). Subroutine GTAGC computes an address within RAGC based on sounding frequency index and the hour of the day (TSG time). The 12 frequency bands are displayed by TEST task AGC (Sect. 10.3). RAGC is initialized to zeroes.

The K-mode AGC table, RAGCK, reserves 2 bytes (RF and IF) for each of the 1Ø selectable K-mode frequencies. When a K-mode frequency is selected by command F, the appropriate RF and IF attenuation bytes from RAGC are copied into RAGCK.

The gain control profile tables for both modes are updated identically. Subroutines NXTFRQ (Sect. 2.9.1.2.1), as part of setting up for the first pulse of a pulse set (NXTF.PØ, NXTK.PØ), puts a two-part entry on the bottom of list RAGCAD. The first part of the entry is the value of PULSE at which all echoes for the pulse set will be in EKO buffers, enabling the FEP data peak average to be computed for the pulse set. (See WTEKO, Sect. 2.7.2.5.) The second part of the entry is the address of the RF/IF attenuation bytes in either RAGC or RAGCK. After WTTSG (Sect. 2.7.2.11), but before CLOCK is called, a call is made to subroutine UPAGC (Sect. 2.10.2). UPAGC, if PULSE is right, finds the peak average, tests it against MINPK and MAXPK, and adjusts the gain con-

trol profile tables at the address in RAGCAD if necessary.

Both RF and IF attenuations lie in the range Ø-6Ø dB in 4-dB steps. The steps are made in the sequence shown in Table 2.5; both receiver channels are attenuated the same. No attenuation changes are made during a pulse set.

2.12.8. DIO tables (DIO, DIODEV)

The DIO table contains the actual command bytes written to the DIO interface to control the transmitter and receiver instruments. DIODEV is a table of device addresses and corresponds one-to-one with table DIO. The octave filters are not yet implemented. The DIO table is set up by subroutine NXTFRQ (Sect. 2.9), and its data, combined with the DIODEV address, is written to the DIO interface by SETDIO (Sect. 2.7.2.2). The DIO command data are compacted into a PCT (Sect. 7.2), associated with each data buffer passed to PICKER, to define the hardware configuration at the time the pulse was transmitted and received.

The DIO table structure comes from SYSC:PCB.CAL (see Chapter 7, introduction) and is described below. See Sect. 7.2 for complete descriptions of the DIO commands. The synthesizer frequency commands in the first 3 bytes are in BCD as output by subroutines NTOFR and FTOBCD (Sect. 2.11). The BCD frequency corresponds to frequency index in PCT.DIO1.

DISPLACEMENT	BYTES	DESCRIPTION	HEX DIO ADDRESS
DIO.FREQ	1	tens, units MHz	1D
DIO.FREQ + 1	1	hundreds, tens kHz	1C
DIO.FREQ + 2	1	units kHz, hundreds Hz	1B
DIO.ANC1	1	antenna, channel 1	1Ø
DIO.ANC2	1	antenna, channel 2	11
DIO.OFC1	1	octave filter, channel 1	12
DIO.OFC2	1	octave filter, channel 2	13
DIO.RFC1	1	RF attenuation, channel 1	14
DIO.RFC2	1	RF attenuation, channel 2	15
DIO.IFC1	1	IF attenuation, channel 1	16
DIO.IFC2	1	IF attenuation, channel 2	17
DIO.BWC1	1	bandwidth, channel 1	18
DIO.BWC2	1	bandwidth, channel 2	19
DIO.CALA	1	calibration attenuation	1A
DIO.TXAT	1	transmitter attenuation	31

RECEIVER GAIN	RF ATTENUATION	IF ATTENUATION
Ø	Ø	Ø
-4	Ø	-4
-8	Ø	-8
-12	Ø	-12
-16	Ø	-16
-2Ø	Ø	-2Ø
-24	-4	-2Ø
-28	-4	-24
-32	-8	-24
-36	-8	-28
-4Ø	-12	-28
-44	-12	-32
-48	-16	-32
-52	-16	-36
-56	-2Ø	-36
-6Ø	-2Ø	-4Ø
-64	-24	-4Ø
-68	-24	-44
-72	-28	-44
-76	-28	-48
-8Ø	-32	-48
-84	-32	-52
-88	-36	-52
-92	-36	-56
-96	-4Ø	-56
*-1ØØ	-4Ø	-6Ø
*-1Ø4	-44	-6Ø
*-1Ø8	-48	-6Ø
.	.	.
.	.	.
.	.	.
*-12Ø	-6Ø	-6Ø

Receiver Gain in dB relative to maximum.
* At these attenuations, overload would occur in the antenna RF preamplifiers supplied.

Table 2.5. ATTENUATION STEPS.

2.12.8. DIO tables (DIO, DIODEV)

Routine SETEX (Sect. 2.7.2.13) assumes this order when removing the remote control locks from DIO devices addressed 1Ø through 1A.

2.12.9. Protected frequencies map (FRQMAP)

FRQMAP contains a bit for each frequency index within the range of the synthesizer, starting with 274 (about 1ØØ kHz) and going up to 3561 (about 3Ø MHz). When a base frequency is selected for sounding in either I- or K-mode, its index is converted to a FRQMAP bit position. If the bit in this position is off, the frequency is protected; if it is on, the frequency can be used for sounding. (See MAPFIX and MAPF, Sect. 2.10.10.)

TEST task FRMAP (Sect. 10.3) constructs the maps and saves them on disc as SYS1:IHUSH.SND (I-mode) and SYS1:KHUSH.SND (K-mode). INIT (Sect. 2.13.1) initializes FRQMAP by reading IHUSH.SND into it. Thereafter each time a sounding is initiated by GO, KRUN, or VUF, SOUNDER reads in the map constructed for the sounding mode, unless the correct map is already in FRQMAP.

Each time a map is read from disc, SOUNDER adjusts it to the current delta frequency (DELF) and to the current pulse set sounding sequence (PSEQ) so that protected frequencies are not violated by sounding on adjacent unprotected base frequencies plus or minus multiples of deltaF. Also, commands DELF and PSEQ force a new map to be read in and adjusted. (See Sect. 2.8.4.5.)

Check bytes are stored with the maps on disc and in FRQMAP. The integrity of the base, or unadjusted, map is checked after it is read from disc; the adjusted map is checked before every sounding. (See CKMAP, Sect. 2.10.11.)

2.12.10. Register storage (SREGS, DREGS)

SREGS and DREGS, each 32 bytes long to accomodate all 16 registers, are at the end of STBLS. SREGS is used for sounding register storage by ADVICE, NXTFRQ, and CLOCK. DREGS, originally set aside for debugging, has evolved into a general register storage area used by SPIPE, SADVC, SSUBS, and SBFRS routines. Other register storage areas are in SSUBS and SBFRS.

2.13. PCT/EKO and SBFRS [D.C.W.]

Module SBFRS serves a dual purpose for task SOUNDER. During a sounding SBFRS is used for Pulse Configuration Tables and EKO buffers. Before and between soundings SBFRS contains code not required for conducting a sounding. This code, which initializes the SOUNDER task and executes idle-mode advice (Sect. 2.5), is described below. The idle-mode ADVICE code, saved as an overlay on disc, is also in memory during TEST soundings. (See Sects. 2.8.3, 2.13.1, and 2.13.12, and Fig. 2.2.)

2.13.1. SOUNDER task initialization

Refer to Figure 2.16.

Task SOUNDER begins execution at INIT after being started by COUNSEL (Sect. 6.1). Because INIT is executed only once, it is loaded into PCT/EKO buffer area 4 (Fig. 2.2) and is destroyed when sounding begins. Thus INIT adds nothing to SOUNDER's total memory requirements.

INIT assigns logical unit 1 to the console, the default logging device (Sect. 2.10.8), and logs the task start-up message. The above calls to MT2 allow COUNSEL, initially with higher priority than SOUNDER, execution time for the CØØ queue to SOUNDER. INIT gets the address of the CØØ Q-block from the top of SNDRQ (Sect. 2.8), and puts the address of the Environment Table (ET) into CØØ + 4. When COUNSEL later queues the other tasks with the CØØ Q-block, they obtain SOUNDER's global information from A(ET) in CØØ + 4 (Sect. 7.3). INIT establishes ET.ABUF and ET.LBUF, sets ET.SCAN to point to SOUNDER's UDL and command scan table, and clears the CØØ Q-block status. MANAGER uses the shareable part of SOUNDER's PCT/EKO buffers in copying the sounding data from disc to tape (Sect. 4.9). After COUNSEL has received all of the tasks' start-up queues, it sets ET.SCAN to its own UDL/SCAN list.

INIT next initializes the FEP by sending a command byte of 18 (hex) to device 77 (hex), the FEP/EKO interface device. The high power transmitter is reset with a call to TXRES (Sect. 2.13.7), and the dummy antenna load is selected, preserving the states of the filaments and the high voltage. Maximum transmitter, RF, IF, and calibration attenuations are selected, and the synthesizer is put in range at 11 MHz. Next the exclusive OR checksum of the idle-mode advice overlay, loaded into PCT/EKO buffers 6 and 5 (Fig. 2.2), is computed and saved in byte OVCKSUM immediately following the overlayed code. The code and OVCKSUM are written to SYS1:SOVLAY.SND. The overlay thus saved is read on demand and tested for a checksum of Ø by GETOV (Sect. 2.8.3).

2.13.1. SOUNDER task initialization

INIT next makes logical unit assignments for LU2-5, as described in Section 2.14, and initializes FRQMAP to the I-mode protected frequencies map with a call to GETMAP (Sect. 2.13.9). LDFEP (Sect. 2.13.5.1) loads the FEP with its default program from SYS1:FEPMEM.SND using the default thresholding factor, 2.5. The TSG is bootstrapped with a call to BTTSG (Sect. 2.13.5.2). The MT2 system clock is set from the TSG clock at SETIME, written by J. R. Winkelman, utilizing RDCLK (Sect. 2.13.10.1). SOUNDER now sends COUNSEL the SØØ Q-block, which contains the address of SOUNDER's UDL and command scan table, SMNEM (Sect. 2.8.7). The default ICT values are copied into CICT by UPADV (Sect. 2.10.5), and SVC6.ID is changed from "COUNSEL" to "PICKER", the task normally queued by SOUNDER. Finally, PULSE is set to -1, and MODE, P.FEP, and P.EKO are cleared to Ø (Sect. 2.7), and control goes to GETADV where the IMA loop is entered (Sect. 2.7.1).

2.13.2. Normal sounding initiation

Refer to Figure 2.17.

Operator commands GO, VUF, and KRUN invoke overlayed advice routines OVGO, OVKVU, and OVKRUN, respectively, by the process described in Sections 2.8.3, 2.8.4, and 2.8.4.1. All three routines enter common code at GORUN (Fig. 2.17, connector D) with R11 and R12 set to sounding mode and the number of K-mode frequencies, respectively.

OVGO clears both R11, indicating I-mode, and R12, indicating K-mode frequencies, to Ø. ET.AICT, preset to A(ICT) to permit the other tasks to put their revisions into ICT before it is copied to CICT, is set to A(CICT). After ensuring that the I-mode protected frequencies map is in FRQMAP with a call to GETMAP (Sect. 2.13.9), OVGO enters the common code at GORUN.

OVKVU and OVKRUN enter common code at STFIX with R4 and R14 set to the highest and lowest frequency indexes, respectively, to be used in the K-mode sounding, information required by GRAPHER. STFIX in turn exits to GORUN.

OVKVU sets R11 to 2, indicating VUF mode, then checks the VUF command argument, n, by examining the nth KMODE table entry. If the nth K-mode frequency has not been set by the F command, control goes to DSPALL (Fig. 2.19, connector A), and the operator is informed of which K-mode frequencies have been selected, just as if the F command had been given with no arguments (Chapter 8). If the nth KMODE table entry is set, its address is put into both FKAD for NXTFRQ and QKAD for QPCKR. (See Sects. 2.9.2.2, 2.7.2.6, and 2.10.2.) Finally, R12 is set to 1, to indicate one K-mode frequency, both R4 and R14 are set to the nth frequency index, and control goes to STFIX.

OVKRUN sets R11 to 1, indicating KRUN mode, then enters a loop at NXTKM that scans through the KMODE table counting the number of frequencies set in KMODE (R12), and finding the lowest (R14) and highest (R4) frequency indexes in the KMODE table. If the KMODE entry count in R12 is Ø, the BADADV exit from ADVICE is taken; otherwise control goes STFIX.

STFIX, the routine common to OVKVU and OVKRUN, checks to see if an I-mode sounding has been run since start-up by examining HRSEC, the number of seconds into the hour. If HRSEC = -1, its pre-set value, no sounding has yet been run, and the BADADV exit is taken. (An I-mode sounding is required as a reference for setting K-mode frequencies.) If HRSEC is not -1, subroutine RDTIME (Sect. 2.13.10) has been called, indicating that an ionogram has been run, and the K-mode run initialization continues. STFIX saves the table address registers (Sect. 2.8.1) and R11 and R12, sets ICT.SFIX to R14, the low frequency index, and calls NTOFR (Sect. 2.11). The frequency output by NTOFR in R12 and R13 is put into ICT.SFRQ. R4, the high frequency index, is copied into R14 for NTOFR, and ICT.EFIX is set to R14. After another call to NTOFR, ICT.EFRQ is set and the registers saved above are restored. A call to GETMAP ensures that the K-mode map is in FRQMAP, and common code GORUN is entered.

GORUN, entered with R11 and R12 set, sets ICT.MODE to R11 and ICT.NKFR to R12. At this point, if ICT.INUM = ICT.NI, the number of soundings specified by command NI have been run, so the BADADV exit is taken. The operator must give another NI command to continue sounding. The default value of ICT.NI is -1, so unless the command NI is given, the above case is moot. For the normal case, GORUN sets RAGCID, the receiver gain profile tables identifier, to ICT.INUM + 1, the number of the sounding just initiated. (RAGCID is written to disc file SYS1:RGAIN.SAV with tables RAGC and RAGCK (Sect. 2.12.7) by command RGAIN. This file serves as input to TEST task AGC.) GORUN now assumes that the GO, VUF, or KRUN command just given is not time specific and sets R5 to CDNVAL, the countdown value set by command CDN. The input character after the command is tested, and if it is the carriage return character, the sounding is indeed scheduled to begin immediately after the countdown, and control goes to CKSYN with R5 = CDNVAL. Otherwise, GORUN converts the command time argument to seconds past the current hour, and with this value in R5, enters CKSYN. (See command GO, Chapter 8.)

CKSYN puts the DIO interface in the read-only state with a call to ROULM (Sect. 2.10.3) and reads the synthesizer status via DIO address 1E (hex). Unless the three synthesizer flags are all true (1), an error message is logged, with the error code being the status as read. (See ERROR, Sect. 2.10.7.) The flags are described below, starting with the most significant bit.

2.13.2. Normal sounding initiation

Synthesizer power	Ø = off, 1 = on
Command source	Ø = local, 1 = remote (Interdata via DIO)
Setting	Ø = out of range, 1 = in range

The synthesizer frequency can be set manually by thumb-wheels on the instrument, and the operator can continue through the error. If the operator needs to run a diagnostic sounding at a particular frequency with the synthesizer command source set to local so that DIO commands from SOUNDER are ignored, the software frequency protection system will be entirely circumvented. Be careful.

At RDTSGT the TSG clock is read by RDTIME (Sect. 2.13.10), setting ET.SEC to the current time of day. R1 is set to ICT.MODE for SETGO, below, and R5, which contains CDNVAL or the specified start time in seconds past the start of the current hour, is tested. If R5 = CDNVAL, control goes to SETGO. Otherwise, the sounding start time in seconds from the current time, ET.SEC, is computed in R5. If the specified start time is in the past or is too soon in the future to allow for the count-down of CDNVAL seconds, the BADADV exit is taken. Otherwise, control goes to SETGO with R1 and R5 set up.

At SETGO, 1 is added to R5 to allow for the current, partially elapsed second, assuming that the TSG clock was not read exactly on the second, and ET.TTNI is set to R5 (CLOCK, Sect. 2.10.9). Finally, GO is cleared to Ø, the flag to go from the IMA to the SPC loop (Sect. 2.7.1.2), and MODE is set to R1, the sounding mode from R11 at GORUN, above, or from OVTPR (Sect. 2.13.3). NXTACT (Sect. 2.7) is set to A(NXTIONO) to keep SOUNDER in the SPC loop marking time until the sounding starts, and the OKADV exit is taken from ADVICE (Sect. 2.8.2).

2.13.3. Magtape replay sounding

Refer to Figure 2.18.

Command TPRUN invokes overlayed advice routine OVTPR. OVTPR checks the TEST flag. If TEST = 1, a TEST task that uses magtape may be active, and the BADADV exit is taken. Otherwise, TEST is set to 1 to lock out the TEST system and thus to avoid a possible magtape conflict. Logical unit 6 is then assigned to magtape, and the CICT and KMODE tables are read in from magtape. (See TEST task TAPEKO, Sect. 10.3.) Tape errors up to this point are logged, control goes to TPRBAD, which sets R3, the exit register, to A(BADADV), and a jump is made to TPREX (Fig. 2.18, connector C). If there are no tape errors, SØ2LST (Sect. 2.12.1) is set to circulate through 4 PCT/EKO buffers as in the TEST mode. Only PCT/EKO buffers 1 through 4 will be used (Sect. 2.13.12). Finally, before control goes to SETGO (Fig. 2.17, connector G, and Sect. 2.13.2), R1 is set to -1, the sounding mode for TPRUN, and R5 is set to CDNVAL. After

SETGO exits ADVICE through OKADV, CLOCK (Sect. 2.10.9) executes the necessary pre-sounding initialization and countdown, and sends the SØ4 queue to PICKER. When the countdown is finished, NXTIONO (Sect. 2.7.2.1) exits the SPC loop, executed only during the countdown for TPRUN, and control goes to GOTPR.

GOTPR is the dummy SPC loop for a TPRUN sounding (Sect. 2.7). Like GETBFR (Sect. 2.7.2.9), GOTPR gets the address of an SØ2 Q-block from the top of SØ2LST and checks the SØ2 status. GOTPR, however, simply waits if necessary until PICKER releases the buffer, returns the A(SØ2) to the bottom of SØ2LST, and reads the tape record into the PCT/EKO buffer associated with the SØ2 Q-block. When tape EOF is sensed, control goes to Q.SØ6, described below. If the operator continues through a tape read error, the buffer word count is cleared to Ø and PICKER ignores the data. At CKIX, the pulse frequency index is extracted from the PCT as read from tape, and at TPRK, table KMODE is scanned for this index. If it is found, its KMODE table entry address is put into SØ2 + 8 (see QPCKR, Sect. 2.7.2.6); if it is not found in KMODE, SØ2 + 8 is cleared to Ø. This is a bit crude: it is not required if the taped sounding is I-mode; for K-mode, it assumes that there are no duplicates in the KMODE table. In any case, the SØ2 queue is sent to PICKER and is indistinguishable from those sent by QPCKR in the normal SPC loop. ADVICE is called next, and if on return from ADVICE, MODE is still -1, the GOTPR loop continues. Otherwise TPRUN has been aborted, and the magtape is rewound to the load point and control goes to Q.SØ6. (See ABORT, Sect. 2.8.4.2.)

Q.SØ6 sends the SØ6, or sounding ended, queue to PICKER whether the run was aborted or the tape EOF sensed. The exit register, R3, is set to A(GETADV), and control goes to TPREX.

TPREX sets SØ2LST back to circulate through all 6 PCT/EKO buffers, and clears the TEST flag to Ø. In case TPREX has been entered from TPRBAD, meaning that the ADVICE subroutine has not been exited, PULSE is set to -1, and MODE is cleared to Ø, clearing the TPRUN command. NXTACT, set to A(NXTIONO) by SETGO for the TPRUN countdown, is set back to A(GETADV). (See Sect. 2.7.) Control goes through R3 to either BADADV or GETADV and to the IMA loop.

2.13.4. Kinesonde mode setup

Operator command F invokes overlayed advice routine OVF (Fig. 2.19). If command F is given with no arguments, control goes to DSPALL where all active KMODE table entries are displayed on the console. DSPALL is also entered from NEWMAP (Sect. 2.8.4.5) as the result of the DELF or PSEQ command, and may be entered from OVKVU (Sect. 2.13.2) if the VUF command argument does not indicate an active KMODE table entry. (See KMODE, Sect. 2.12.4.) If command F has a single argument, n, the nth KMODE

entry is displayed if it is active. If command F has two arguments, n and m, the nth KMODE frequency index is set, stepped, or deleted, depending on m. (See command F, Chapter 8.) The new frequency index is checked against the K-mode protected frequencies map (Sect. 2.10.10), and the RF/IF attenuations established for I-mode are moved from table RAGC to RAGCK (Sect. 2.12.7), and become the base attenuation settings for the next K-mode sounding. These in turn are adjusted during K-mode sounding, if necessary, according to the EKO data peak averages. (See WTEKO, Sect. 2.7.2.5, and UPAGC, Sect. 2.10.2.) The new frequencies actually selected, with associated range limits, are displayed, and GRAPHER is queued with the SØ8 Q-block, into which is put n, the K-mode frequency number, which is also the KMODE table entry number (Sect. 2.12.4). GRAPHER, in response to the SØ8 queue, superimposes a K-mode frequency marker upon the last ionogram plotted on the system display (Sect. 5.4).

The subroutines used by OVF and the other K-mode setup routines are discussed in Section 2.13.4.1.

Commands RN and RF invoke overlayed advice routines OVRN and OVRF (Fig. 2.20), and set the near (minimum) and far (maximum) ranges at displacements +2 and +4, respectively, of KMODE table entry n (see commands RN and RF in Chapter 8). The ranges are specified and displayed in km, but are stored in microseconds. The nth KMODE table entry is displayed, and GRAPHER is queued as described above.

Commands VFIL, KFIL, and KSET invoke overlayed advice routines OVVFIL, OVKFIL, and OVKSET, respectively (Fig. 2.20). These routines set or display FILLV, FILLK, and SETK, respectively. If any of the above three commands is given without an argument, control goes to DSPDNO where the current setting is converted to decimal ASCII and displayed. FILLV and FILLK are used by NXTK.P4 (Sect. 2.9.2.2) to set NFILL, the NOP fill count between K-mode pulse sets. SETK is used by NXTIONO (Sect. 2.7.2.1) to set NSET, the number of K-mode pulse sets, for a KRUN sounding. For a VUF sounding, NSET is computed to be the number of pulse sets required to sound for 3Ø seconds, based on FILLV.

2.13.4.1. K-mode setup subroutines

CKCR (Fig. 2.20) tests the first character after an advice command, and if it is a carriage return, takes the normal return. Any other character causes CKCR to return 4 bytes beyond the normal return with the console input command line pointer in R1. R1 is then used to update CØ2 + 6 for further command line interpretation (CONVD, Sect. 2.8.4.6). Because the skip return is over 4 bytes, the branch instruction following the call to CKCR must not be of the short, or 2-byte, form.

GTAKM (Fig. 2.20) gets the first argument, n, of the F, RN, or RF command and returns the address of the nth KMODE table entry in R5 and n in R6. If n > NKFRQ (Sect. 7.6), the BADADV exit (Sect. 2.8.2) is taken. For this reason, GTAKM is strictly an ADVICE subroutine.

DISPKF (Fig. 2.20) is called with the output of GTAKM in R5 and R6 and displays the nth KMODE table entry with frequency in kHz and range limits in km. DISPKF calls WRLOG (Sect. 2.10.8) to display the entry and then passes n in R6 to QSØ8 to queue GRAPHER.

QSØ8 (Fig. 2.19) is called with n in R6, as output by GTAKM, puts R6 into SØ8 + 4, and sends the SØ8 queue to GRAPHER. Although the SØ8 Q-block status field (SØ8 + 2) is initialized to -1ØØØ (hex), it is never checked. QSØ8 is called whenever a KMODE table entry is modified or displayed.

KMTOUS (Fig. 2.20) gets the second argument of the RN or RF command, a range expressed in km, and returns with the range converted to microseconds in R9. If the input range is negative, KMTOUS takes the BADADV exit from ADVICE (GTAKM, above).

USTOKM (Fig. 2.20) is called with a range in R9 expressed in microseconds. This range is converted to km in the form of a decimal ASCII string stored in BUFR, a general purpose storage and buffer area in module SBFRS. The string is moved from BUFR to the DISPKF message (above) by MUVEM (below). (USTOKM and KMTOUS assume that a radar pulse makes a round trip of 15Ø m in 1 microsecond.)

MUVEM (Fig. 2.20) copies an ASCII string from BUFR to the near or far range field in the DISPKF message. MUVEM is called with the destination address in R1 and the byte count in RØ. The string in BUFR is put there by an SVC2 call made by USTOKM to unpack a decimal range. (See Programmer's Reference Manual.) MUVEM has a second entry point, MUV. MUV is called with the source address in R3, destination address in R1, and byte count in RØ. MUV is called by LDFEP (Sect. 2.13.5.1) and BTTSG (Sect. 2.13.5.2).

2.13.5. Microprocessor loading and bootstrapping

Operator command BOOT invokes overlayed advice routine OVBOOT (Fig. 2.21). OVBOOT scans the BOOT command argument in the same manner that COUNSEL scans the first word of an advice command (Sect. 2.8.7). The BOOT argument mnemonic table is BTSCAN and the corresponding subroutine address table is BTVECT. The BOOT argument index is multipled by 2 to index BTVECT. If this result is negative, the argument is invalid and the BADADV exit (Sect. 2.8.2) is taken. In case the FEP is to be loaded, OVBOOT sets R11 to -1 and RØ to 25. These values tell LDFEP (Sect.

2.13.5.1) that the default FEP thresholding factor of 2.5 is to be used. (See command FEPTH, Chapter 8.) R4 is set to the appropriate subroutine address from the BTVECT table, and R3 is set to the address of "SND", on the assumption that default file SYS1:TSGMEM.SND or SYS1:FEPMEM.SND is to be used by the BOOT subroutine. If the next command line character after the microprocessor name is a carriage return, the default file is indeed used. Otherwise, a file name extension has been specified, and R3 is set to the address of that extension in COUNSEL's memory. (See CONVD, Sect. 2.8.4.6.) In case the FEP has been named, and a non-standard FEPMEM file is to be used, RØ is cleared to Ø, meaning no FEP thresholding. RØ and R3 are used by LDFEP to set ICT.FEPT and ICT.FEPM, respectively. At CALLBT subroutine LDFEP or BTTSG is called through R4, set from BTVECT, above. If a nonexistant (on volume SYS1) TSGMEM or FEPMEM extension is specified, the skip return is taken from the subroutine call, and control goes to BADADV. Otherwise, the OKADV exit is taken.

Command FEPTH invokes overlayed advice routine OVFEPTH (Fig. 2.21). OVFEPTH calls CONVD (Sect. 2.8.4.6) to get the FEP thresholding factor times 1Ø. If OVFEPTH finds this to be a legal entry in table THFCTR, the corresponding FEP program byte to effect the specified thresholding is loaded into R11 from table THRSH. (Tables THFCTR and THRSH are listed in Chapter 8 under command FEPTH.) R3 is set to point to "SND"; the default file SYS1:FEPMEM.SND is used for making thresholding changes. Subroutine LDFEP (Sect. 2.13.5.1) loads the FEP, substituting the byte in R11 for appropriate bytes from the program on the disc. Because the file name extension cannot be specified by the FEPTH command, the skip return cannot be taken from this call to LDFEP, and control goes to OKADV. (If SYS1:FEPMEM.SND is missing, the call to LDFEP by INIT (Sect. 2.13.1) will result in an error message.)

2.13.5.1. Load FEP program memory

LDFEP loads the FEP program memory from a disc file on volume SYS1, the bottom or fixed disc. The file name is always FEPMEM; unless the LDFEP call is from OVBOOT (Sect. 2.13.5) the default file, FEPMEM.SND, is always used. LDFEP is called by INIT (Sect. 2.13.1), OVBOOT, and OVFEPTH.

Input
- RØ: threshold factor for ICT.FEPT.
- R3: address of FEPMEM extension.
- R7: A(ICT) (see I., Sect. 2.8.1).
- R11: new thresholding byte or -1 if no thresholding change.
- SYS1:FEPMEM.xxx: FEP program (Sect. 10.9).

SCTRL: table of 4 FEP control bytes sent to FEP after each of the first 10 program bytes.
FCTRL: table of 5 control bytes with FEP program location number, sent after final program byte.
FDIOA: table of DIO addresses associated one-to-one with table FCTRL.

Output ICT.FEPT: threshold factor.
ICT.FEPM: FEPMEM file extension.
FEXT: FEPMEM file extension in SVC7 parameter block, ASFMEM, to assign LU6 to FEPMEM file.
FEP program memory loaded, with altered thresholding bytes if call from OVFEPTH.
RESET and REINIT commands: sent to FEP before and after loading program memory.
Error message if LU assignment, disc read, or threshold change fails.
R0-R14: returned intact using storage area BTREGS.
R15: return address.

Method Refer to Figure 2.21.

LDFEP saves R0-R15 in BTREGS and copies "SND", or the extension specified in the BOOT command, to FEXT. (See MUV, Sect. 2.13.4.1.) LDFEP resets and reinitializes the FEP by sending command bytes 0 and 18 (hex) to the FEP/EKO interface, device 77 (hex). The DIO interface is put in the read-write state with a call to RWULM (Sect. 2.10.3).

At AS6.F, LU6 is assigned to SYS1:FEPMEM.xxx. If an LU assignment error is sensed, and if extension xxx is other than "SND", R0-R15 are restored and the skip return is taken (see OVBOOT, Sect. 2.13.5). If the trouble is with FEPMEM.SND, an error message is logged, and the assignment is repeated if the operator continues through the error. (See ERROR, Sect. 2.10.7.) If the LU assignment is successful, the file extension is copied from FEXT to ICT.FEPM by MUV. The first FEPMEM record is read from disc, and the information for changing the FEP threshold is loaded into R12-R15 (FEPMEM, Sect. 10.9).

At RDFEP, one FEP program memory record is read from disc into BUFR, SBFRS's general purpose buffer. If the disc read is normal, control goes to RDFOK; read errors are logged. If EOF is sensed on the FEPMEM file, R11 is checked to see if the threshold change, if any, was successfully made. If R11 = -1, no change was requested, or the FEP locations to be changed were both found on the disc and modified before being sent to the FEP. If such

is the case, control goes to RESFEP where LDFEP resets and reinitializes the FEP as described above, closes LU6, restores RØ-R15 from BTREGS, puts the input RØ into ICT.FEPT, and takes the normal return; the FEP loading is complete. If EOF is sensed, but R11 is not -1, then one or both of the specified FEP program locations involved with the threshold change were not found on the disc file, and an error message is logged. The operator can continue through the error, but should assume that the FEPMEM file has somehow been altered and that the FEP load has failed.

At RDFOK, BUFR contains a new FEP program record. In BUFR + Ø is the number of the FEP location to be loaded; BUFR + 1 through BUFR + 11 contain the program bytes, or the image of the FEP location. If R11 = -1, control goes to DLFEP, and the record is sent to the FEP program location indicated by BUFR + Ø. Otherwise, if R12 = BUFR + Ø, the first location to be modified for the FEPTH command is in BUFR; R13 contains the number of the byte to be changed, and BUFR + R13 is set to R11, the program byte from table THRSH (Sect. 2.13.5). Likewise, if R14 = BUFR + Ø, BUFR + R15 is set to R11. After the second FEP location is successfully modified for command FEPTH, R11 is set to -1, and from then on, RDFOK passes control directly to DLFEP.

At DLFEP, the program bytes in BUFR + 1 through BUFR + 11 are sent to the FEP in reverse order, i.e., BUFR + 11 through BUFR + 1. The first 1Ø bytes, BUFR + 11 through BUFR + 2, are sent by loop FRST1Ø in the following manner: a program byte is sent on DIO address 36 (hex) followed by hex control bytes 41, 61, Ø1, and 21. These control bytes are in table SCTRL and are sent on DIO address 37 (hex). The resulting action at the FEP is that the program byte is loaded into an 84-bit register, then right shifted 8 bits. (The low-order 4 bits of BUFR + 11 are finally shifted out and lost.) After the first 1Ø program bytes have been thus sent, the final program byte, BUFR + 1, is sent on DIO 36, followed by hex control bytes 41 and 61 on DIO 37. The 84-bit FEP register is now fully loaded. BUFR + Ø, the FEP program location number, is next sent on DIO 36, followed by control bytes 31, 21, and AØ, and the 84 bits are transferred into the FEP program memory at the location specified. The final 5 control bytes, with BUFR + Ø, are stored in table FCTRL, and the corresponding DIO addresses are in table FDIOA. Control goes back to RDFEP, where the next FEPMEM record is read from disc.

2.13.5.2. TSG bootstrap

Subroutine BTTSG resets the TSG, which causes the TSG memory to be loaded from TSG PROM, and rewrites the TSG zero page if so specified by command BOOT (Chapter 8). BTTSG reads the TSG zero page and from it sets up the TSG repeat count table, TSGREP, for CLOCK (Sect. 2.10.9), and establishes the table of TSG delays, TSGDLY (Sect. 2.12.3). A disc image of the TSG PROM is stored in file SYS1:TSGMEM.SND. Should the load of the zero page from PROM fail, the zero page can be established to its default condition with the command BOOT TSG,SND. BTTSG checks for the existence of file SYS1:TSGMEM.SND, unless a user version of page zero is specified by command BOOT. BTTSG is called by INIT (Sect. 2.13.1) and by OVBOOT (Sect. 2.13.5).

Input
R3: address of SYS1:TSGMEM file extension in ASCII.
R7: A(ICT) (see I., Sect. 2.8.1).
SYS1:TSGMEM.xxx: disc image of TSG page Ø if command BOOT TSG,xxx given.
Page Ø from TSG.
ICT.TSGC: TSG command byte.

Output
ICT.TSGM: TSGMEM file extension.
TEXT: TSGMEM file extension in SVC7 paramter block ASTMEM to assign LU6 to TSGMEM file.
Reset signal to TSG to effect load from PROM.
TSG page Ø, from sector Ø of SYS1:TSGMEM.xxx, if command BOOT TSG,xxx given.
TSGREP: TSG repeat count table.
TSGDLY: table of TSG delays.
ICT.TDLY: TSG delay for current scenario in ICT.TSGC.
Error message if LU6 assignment fails.
R2-R14: returned intact using storage area BTREGS.
RØ: ICT.TDLY.
R1: current TSGDLY index (scenario number times 2).
R15: return address.

Method
Refer to Figure 2.22, Sheet 7-A.

BTTSG saves RØ-R15 in BTREGS and clears R5 to Ø if no write to TSG page Ø is specified in the BOOT command. If a TSG page Ø write is required, R5 is set to 1. The file extension specified in the BOOT command, or the default "SND", is copied to TEXT (see MUV, Sect. 2.13.4.1). At AS6.T, LU6 is assigned to SYS1:TSGMEM.xxx; an assignment error here is handled exactly as it is at AS6.F (Sect. 2.13.5.1). If the assignment is sucessful, MUV is again called to copy the file extension from TEXT to ICT.TSGM.

2.13.5.2. TSG bootstrap

BTTSG now calls RSTSG (Sect. 2.13.5.2.1.) to reset the TSG, effecting the loading of the TSG memory from PROM and putting the TSG into a loop waiting for its non-maskable interrupt (NMI). The NMI is triggered whenever a command byte (read/write or scenario request) is sent to the TSG. (See RWTSG, Sect. 2.13.5.2.5, and U4ØDVR, Sect. 2.15.1.)

BTTSG next puts Ø, the TSG page number, into R1 for subroutines TSGPG, WRTSG, and RDTSG (below), and copies R5, the default flag, into R8. If R8 = Ø, no write to TSG page Ø is required, and control goes to TSGØ where page Ø is read from the TSG; otherwise, TSGPG (Sect. 2.13.5.2.2) is called to read sector Ø from file SYS1:TSGMEM.xxx; WRTSG (Sect. 2.13.5.2.3) writes sector Ø to the TSG zero page. If the write is successful, control goes to TSGØ; otherwise, control goes to CLST6, and tables TSGREP and TSGDLY are unaltered.

At TSGØ, RDTSG (Sect. 2.13.5.2.4) is called to read page Ø from the TSG. If page Ø has just been written, RDTSG performs a read-after-write check. If RDTSG reports a read or a write check error, control goes to CLST6; otherwise, the 4 repeat counts, as read from the TSG, are loaded into R14 and R15. If R14 = Ø, control goes to CLST6; otherwise, the repeat counts are copied into table TSGREP. TSGREP is used by subroutine CLOCK to determine the time value of a TSG tick, or interrupt.

Next, the TSG delays are computed for scenarios 3-7 and are put into table TSGDLY. The delays are computed in microseconds from the center of the transmitter keying pulse to the beginning of the FEP WRite ENable window. (See SNRIO display, Sect. 10.3.) No delays are associated with current default scenarios Ø-2.

At CLST6, LU6 is closed, and RØ-R15 are restored from BTREGS. The current scenario number is extracted from ICT.TSGC and multiplied by 2 to index table TSGDLY. The appropriate value from TSGDLY is put into ICT.TDLY to be used by PICKER in its range calculations. (See PICKIT, Sect. 3.4.) BTTSG takes the normal return through R15.

The subroutines called by BTTSG are discussed below.

2.13.5.2.1. Reset the TSG

Subroutine RSTSG is called by BTTSG and by RWTEX (Sect. 2.13.5.2.3) to reset the TSG.

Input None required.

Output Reset signal sent to TSG.
RØ, R1, R15: altered.
R2-R13: returned intact.
R14: return address.

Method Refer to Figure 2.22, Sheet 7-A.

The reset signal is sent to the TSG via the least significant bit (LSB) of DIO 25. The second LSB of DIO 25, used to simulate serial EIA Teletype input to the TSG debugging monitor, is kept high (1) throughout the reset sequence.

After the DIO interface is put in the read/write state with a call to RWULM (Sect. 2.10.3), the reset is begun by writing 3 to DIO 25. After a delay of 1ØØ ms to allow the TSG to sense the reset, the reset is ended by writing 2 to DIO 25. RSTSG then delays 8ØØ ms to allow time for the TSG load from PROM, and returns through R14.

2.13.5.2.2. Read image of TSG page from disc

Subroutine TSGPG is called by BTTSG to read one page from SYS1:TSGMEM.xxx.

Input R1: TSG page number (disc file sector number, currently always Ø).
Page (256 bytes) of TSG memory image.

Output RØ, R15: altered.
R1-R13: returned intact.
R14: return address.
Error message if disc read fails.

Method Refer to Figure 2.22, Sheet 7-A.

The TSG page image is read from file SYS1:TSGMEM.xxx into BUFR. The TSGMEM file is assumed to be organized in TSG page number order, one sector per TSG page.

2.13.5.2.3. Write page to TSG

Because the DIO was designed to transfer single 8- or 16-bit quantities, the hardware has no built-in "handshaking" signals, which are required for multiple quantity transfers. To circumvent this limitation in order to accomplish TSG zero page swaps, John Taylor of the SEL hardware group has designed a software "handshaking" system utilizing two DIO channels. DIO 24 is used to transfer 8-bit bytes between the Interdata and TSG computers; the second LSB of DIO 25 is used as the TSG data-seen or data-ready flag. The TSG controls the flag, alternating it between Ø and 1, and a byte is transferred for every flag state change. Thus the transfer is insensitive to timing in either computer.

Subroutine WRTSG is called by BTTSG to write one page to the TSG.

Input R1: TSG page number (currently always Ø).
BUFR: image of TSG page from disc (256 bytes).
Second LSB of DIO 25: data-seen flag.

Output RØ: Error condition, Ø for successful write.
Write-request byte sent to TSG.
256 bytes and checksum byte written to TSG.
Error message if write times out.
R1-R14: returned intact using storage area REGS.
R15: return address.

Method Refer to Figure 2.22, Sheet 7-B.

WRTSG saves RØ-R15 in REGS, and calls RWTSG (Sect. 2.13.5.2.5), which sends the write-request byte to the TSG. WRTSG sets up to write 256 bytes from BUFR (the disc image read by TSGPG, Sect. 2.13.5.2.2) to the TSG; zeroes R5, the checksum register; and zeroes bytes E3 and E4 (hex) of BUFR. (Bytes E3 and E4 of the TSG zero page must be Ø for the TSG to function at all; clearing these bytes to zero prevents a TSG hang-up, even though the rest of the data written to the zero page is bad.) The TSG write loop is entered at WRTLP with R1Ø = Ø, the state of the data-seen flag when the TSG is ready to accept the first byte of data.

At WRTLP, WRTSG calls WT25 (Sect. 2.13.5.2.6) to wait until the data-seen flag is in the state specified by R1Ø. When WT25 senses the correct state of the data-seen flag, it toggles, or reverses the state of, the second LSB of R1Ø and takes the normal exit; control goes to WR24 with R1Ø set up for the next data-seen wait at WRTLP. If the correct state of the flag is not sensed in about 1Ø ms, WT25 returns skipping two bytes and an error message is logged. If the operator continues through the error (Sect.

2.10.7), control goes to RWTEX with RØ non-zero, indicating a TSG write error to BTTSG (Sect. 2.13.5.2).

At WR24, WT25 has determined that the TSG has seen the last byte written to it and is ready for another. The DIO interface is put in the read/write state with a call to RWULM (Sect. 2.10.3). The next byte is fetched from BUFR and XORed with R5 to compute a running exclusive OR checksum of all 256 bytes in BUFR. The byte is written to the TSG, and control returns to WRTLP until all 256 bytes have been sent to the TSG. After the full page has been sent, the checksum byte is sent, and control goes to RWTEX with RØ = Ø, indicating to BTTSG that the page write has been successfully completed. (At present the TSG ignores the checksum byte. The write is checked, however, by reading the page back. See RDTSG, Sect. 2.13.5.2.4.)

At RWTEX, the common exit for WRTSG and RDTSG, the error condition in RØ is put into REGS, so that the subroutines exit with the error condition in RØ. If an error is indicated, RWTEX calls RSTSG (Sect. 2.13.5.2.1) to reinitialize the TSG after the incomplete page read or write.

At RWEX, RØ-R15 are restored from REGS, and WRTSG or RDTSG returns to BTTSG.

2.13.5.2.4. Read page from TSG

Subroutine RDTSG is called by BTTSG to read a page from the TSG or to perform a read-after-write check.

Input
R1: TSG page number (currently always Ø).
R8: Ø for read only, 1 for read back to check write.
BUFR: page just written to TSG if R8 = 1.
TSG page read into BUFR if R8 = Ø.
Second LSB of DIO 25: data-ready flag.

Output
RØ: error condition, Ø for no errors.
Read-request byte sent to TSG.
Data acknowledge sent via DIO 24 to TSG.
Error message if read timeout or checksum error.
R1-R14: returned intact using storage area REGS.
R15: return address.

Method Refer to Figure 2.22, Sheet 7-B.

2.13.5.2.4. Read page from TSG

RDTSG saves RØ-R15 in REGS, ORs 4Ø (hex), the read-request flag, with the TSG page number in R1, and calls RWTSG (Sect. 2.13.5.2.5) to send the read-request byte to the TSG. RDTSG then sets up to read 257 bytes (a page plus checksum byte), zeroes R5, the checksumming register, and sets R7 to the TSG data DIO address, 24 (hex). The TSG read loop is then entered at RDTLP with R1Ø = Ø, the state of the TSG data-ready flag when the first data byte is valid on DIO 24. (See WRTSG, Sect. 2.13.5.2.3.)

At RDTLP, RDTSG calls WT25 (Sect. 2.13.5.2.6) to wait until the TSG data is ready on DIO 24. As for WRTSG, R1Ø is updated by WT25; a read timeout is handled like WRTSG handles a write timeout.

At RD24, WT25 has sensed that the TSG has latched the next byte on DIO 24, and DIO 24 is addressed, or selected, by writing its address to the DIO interface. After a delay of about 6 microseconds, the data byte is read via DIO 24 and XORed with the running checksum in R5. R8 is tested, and if it is Ø, a write check is not in progress, and control goes to RDRO. Otherwise, the byte just read is compared to the corresponding byte in BUFR, the byte sent by WRTSG. If the bytes are the same, control goes to RDRO. Otherwise, the read-after-write check has failed, and an error message is logged. The error code contains the correct byte in bits Ø-7 and the erroneous byte in bits 8-15. This code is displayed by ERROR (Sect. 2.10.7). If the operator continues through the error, control goes to RWTEX (Sect. 2.13.5.2.3) with RØ non-zero.

At RDRO, the byte just read from the TSG is put into BUFR, indexed by R2. The DIO interface is put in the read/write state with a call to RWULM (Sect. 2.10.3), and the data from the TSG is acknowledged by writing Ø to DIO 24. On receipt of this acknowledgment, the TSG readies the next byte, and toggles the second LSB of DIO 25. After acknowledging the data, RDTSG loops back to RDTLP until 257 bytes have been read. The 257th byte is the exclusive OR checksum as computed by the TSG. This byte XORed with the checksum computed in R5 (above) should yield Ø. If the XOR of both checksums is Ø, control goes to RWTEX with RØ = Ø. Otherwise, an error message is logged. If the operator continues through the error, control goes to RWTEX with RØ non-zero.

2.13.5.2.5. Send page read or write request to TSG

Subroutine RWTSG is called by WRTSG and RDTSG.

Input R1: Read/write-request byte.

Output Read/write-request byte on DIO 24.
Non-maskable interrupt (NMI) to TSG on DIO 25.
RØ-2, R15: altered.
R3-R13: returned intact.
R14: return address.

Method Refer to Figure 2.22, Sheet 7-B.

The DIO interface is put in the read/write state with a call to RWULM (Sect. 2.10.3), and the read/write request is written on DIO 24. The NMI signal, sent via the third LSB of DIO 25, is started by writing 6 to DIO 25; the second LSB is held high as it is during a TSG reset (Sect. 2.13.5.2.1). After a delay of about 1Ø microseconds for the TSG to sense the NMI, the NMI signal is ended by writing 2 to DIO 25. The TSG, on receipt of the NMI, initializes for the page transfer and when ready, clears the second LSB on DIO 25 to Ø.

The TSG distinguishes between a read/write request and a scenario request by examining the most significant bit (MSB) of the request byte. If the MSB is 1, the request is for a scenario to be run; if it is Ø, the request is for a page read or write, and the TSG examines the second MSB. If this bit is 1, the request is for a page read; if it is Ø, the request is for a page write. The sequence for sending the TSG a scenario request, also referred to as a command byte (Sect. 2.7.2.3), is essentially the same as that described above, and is that used by the TSG driver (Sect. 2.15.1).

2.13.5.2.6. Wait for TSG data-ready or data-seen

In addition to being used to simulate serial EIA Teletype input to the TSG debugging monitor, the second LSB of DIO 25 is used by the TSG to flag data-ready and data-seen conditions for TSG page reads and writes, respectively.

WT25 is called by WRTSG and RDTSG to check the flag.

2.13.5.2.6. Wait for TSG data-ready or data-seen

Input R1Ø: the state of the second LSB of DIO 25 required before continuing with page read or write (Ø or 2).

Output RØ: DIO interface device address.
R1, R6, R15: altered.
R2-R5, R7-R9, R11-R13: returned intact.
R1Ø: input R1Ø.XOR.2 (input state, toggled).
R14: return address.
DIO interface in read-only state.

Method Refer to Figure 2.22, Sheet 7-B.

WT25 puts the DIO interface in the read-only state with a call to ROULM (Sect. 2.10.3) and loads a timeout count in R6. The timeout count allows about 1Ø ms for the TSG to change the flag to its required state.

At WT24, the timeout count is decremented by 1, and if it goes to Ø, WT25 returns skipping two bytes, indicating a read or write timeout. Otherwise, DIO 25 is addressed, and after a delay of about 6 microseconds, DIO 25 is read and the second LSB extracted. The extracted flag is compared to R1Ø, and unless the flag is in the required state, control goes back to WT24. Otherwise, the TSG is ready to continue with the read or write; WT25 toggles the second LSB of R1Ø, in preparation for the next call to WT25, and takes the normal return.

2.13.6. SOUNDER timing delays

SOUNDER uses three methods of delaying its execution for various purposes. First, a very short delay of less than 1Ø microseconds is accomplished by executing the store-multiple (STM) instruction. Examples of this delay are found in CKTXM (Sect. 2.10.6) and TXRES (Sect. 2.13.7.1); the short delay allows time for a reply to be latched in the DIO interface. Second, delays of longer than 1Ø ms can be effected with an SVC2, code 23, call to MT2. Such a delay dismisses SOUNDER allowing other tasks to execute. SOUNDER is not rescheduled to run until the delay time has elapsed. The IMA loop (Sect. 2.7.1.2) uses such a delay to check for operator advice every 1ØØ ms. (See DLAYPB and NOMS, Sect. 2.7.3.) Third, SOUNDER uses subroutine DLAY (Fig. 2.22, Sheet 7-A) for delays of 1 to 9 ms. DLAY is called with a ms count in RØ and simply executes in a timed loop until the count is satisfied. Because DLAY does not permit other tasks to execute, it is only called from other SBFRS code, and never from the SPC loop.

The best that can be said of the delays in SOUNDER is that they are adequate. Because the code must execute on both the Interdata 7/16 and 8/16 machines, which have different cycle times, and because no great effort has gone into making DLAY accurate, the delays cannot be said to be exact. Some delays inserted during coincident hardware and software development may even no longer be required; these will be removed in the future.

2.13.7. High power transmitter control

The HF Radar high power transmitter is controlled by 6 operator commands, two each to control the filaments (FIL), high voltage (HV), and the dummy load (DL). Commands DLON, DLOFF, and HVOFF have immediate effect and are discussed in Section 2.8.4.3. Immediate command TXOFF (Fig. 2.8) turns the FIL and HV off and selects DL and full attenuation. Commands TXON and HVON, which turn on the FIL and HV, respectively, execute timed sequences that would interfere with the SPC loop (Sect. 2.7.2), and for that reason are overlayed at OVTXON and OVHVON.

Four DIO addresses are dedicated to the transmitter. It is controlled via DIO 30 (Table 2.3) and attenuated via DIO 31. (See CKTXM, Sect. 2.10.6.) The transmitter status is read on DIO 32 (Table 2.4). Eight more bits of fault and limit conditions are read by TEST task TXSTAT on DIO 35 (Sect. 10.3).

The TXON, HVON, and TXOFF sequences are diagrammed in Figure 2.27. The actual hex commands issued on DIO 30 that effect a state change are encircled, and the approximate timing is shown below. Broken signal lines indicate indeterminate timing; indeterminate signals are shown in both states. Bit 0 (FIL) is the least significant bit.

Operator command TXON invokes overlayed advice routine OVTXON (Fig. 2.23). OVTXON calls TXRES (Sect. 2.13.7.1), resetting the transmitter. This reset is not part of the sequence in Figure 2.27, but is made to clear any non-current fault conditions and to read the state of DL, which is saved in R5 for SETXMAT, below. Maximum attenuation is put in via DIO 31; FIL and HV are turned off, and DL selected via DIO 30. The TXON sequence now begins with a call to TXRES. After a delay of 90 ms, the filaments are turned on via DIO 30 (FIL = 0). R0 is set to TXLAT, the low power transmitter attenuation set by command TXATLP, and control goes to SETXMAT. The filament warm-up period of about 90 seconds has begun.

Command HVON invokes overlayed advice routine OVHVON (Fig. 2.23). The sequence shown in Figure 2.27 begins with the transmitter reset (TXRES). The DIO 32 status returned by TXRES in R4 is examined. If the filaments are off (FIL = 0), or if the 90-second filament warmup period, as timed by the transmitter controller, is not complete (HV HOLD = 0), the BADADV

exit is taken (Sect. 2.8.2). Otherwise the HVON sequence continues. The current state of DL is saved in R5 for SETXMAT, below, and maximum attenuation and DL are selected. After a delay of 90 ms, the high voltage is turned on via DIO 30 (HV = 0). After another delay of 1 second, the SET signal is taken low for about 10 ms, and the HVON sequence is complete. R0 is set to TXHAT, the high power transmitter attenuation set by command TXATHP, and control goes to SETXMAT.

At SETXMAT, TXMAT is set to R0, the low or high power attenuation setting used during sounding. (See CKTXM, Sect. 2.10.6.) Finally, DL, kept high during the TXON or HVON sequence, is set to its original state, as specified by R5, and the OKADV exit is taken.

2.13.7.1. High power transmitter reset

TXRES is called to reset the transmitter, to clear any non-current fault condition, and to get the transmitter status as read on DIO 32. TXRES is called by INIT (Sect. 2.13.1), OVTXON and OVHVON (Sect. 2.13.7).

Input TX status: read on DIO 32.

Output DIO interface in read/write state.
R1: DIO interface device address.
R3: last command sent on DIO 30 (see Table 2.3).
R4: DIO 32 flags (see Table 2.4).
Transmitter reset: RES active (0) for about 10 ms.
R2, R5-R13: unchanged.
R14: return, from input R15.

Method Refer to Figure 2.23.

TXRES calls ROULM (Sect. 2.10.3) and reads the DIO 32 flags into R4. R4 is then copied into R3; R3 is logically ANDed with a mask having the FIL and HV bits set to 1. DL is positioned for DIO 30, and FIL, HV, and DL are toggled for the DIO 30 command. (See TXCOM, Sect. 2.8.4.3.) A call to RWULM puts the DIO interface in the read/write state, and RES is taken low or active for about 10 ms. SET is kept high or inactive (Fig. 2.27).

2.13.8. TEST task access and termination

Operator command TEST invokes overlayed advice routine OVRUNT (Fig. 2.24). If a TPRUN (Sect. 2.13.3) is in progress, or if a TEST task is already running, the TEST flag will not be 0. If the TEST flag is non-zero, the BADADV exit (Sect. 2.8.2) is taken. Otherwise, the TEST command argument is copied into TNAME, the filename field of ASTSK5, the parameter block used to assign LU6 to the task file to be loaded into

partition 5 (Sect. 10.1); and SVC6.ID is set to "GRAPHER". SWAP5 (Sect. 2.13.8.1) is called to cancel GRAPHER and to load and start the TEST task. Four registers are now set up for OVTCOM, the common routine entered from both OVRUNT and ENDTEST. R8 is cleared to Ø so that the clearing of the CØ2 Q-block status will be deferred (Sect. 2.8.6); R9 is set to the address of table TSTLST (Sect. 2.13.8.2) and is put into SVC6.PAR by OVTCOM; R14 is set to 4Ø4 (hex) to set PCTLST and SØ2LST (Sect. 2.12.1) to circulate on 4 PCT/EKO buffers (GETBFR, Sect. 2.7.2.9); unless the TEST command argument is "HELP", RØ is set to 1 and is put into the TEST flag by OVTCOM to block command TPRUN (Sect. 2.13.3) or another TEST command. If TEST task HELP is being started, RØ is set to -1 for ERROR (Sect. 2.10.7). Control now goes to OVTCOM (below).

ENDTEST (Fig. 2.24, connector A) is entered from ADVICE as the result of the termination queue from a TEST task. (See Sect. 2.8.6.) ENDTEST sets ASTSK5 to assign LU6 to file SYS1:GRAPHER.TSK and calls SWAP5 to cancel the TEST task and to load and restart GRAPHER. (TEST tasks are loaded from SYSC, which must be marked on during TESTing.) Finally the four OVTCOM registers are set up: R8 is set to 1 so that the CØ2 Q-block status will be cleared for the TEST command, if this has not already been done by ADVICE (Sect. 2.8.6); R9 is set to the address of FAKECQ, the CØØ Q-block sent to GRAPHER for a restart (see QUEUE.C, Sect. 5.10); R14 is set to 6Ø6 (hex) for PCTLST and SØ2LST; RØ is zeroed to clear the TEST flag. Control now goes to OVTCOM.

At OVTCOM, the TEST flag is set to RØ, and the TEST mode is either entered or ended. R15 is zeroed, and R14 and R15 are stored into the first 32 bits of PCTLST and SØ2LST; the circular lists will circulate on 4 or 6 PCT/EKO buffers and PCT/EKO buffer 1 will be allocated first (Sect. 2.13.12). The SVC6 parameter block is set up to queue any other task, and SVC6.PAR, the queue parameter is set to R9. A(TSTLST) is sent to a TEST task, or A(FAKECQ) is sent to GRAPHER at Q.T. LU6, assigned to the task (.TSK) file loaded by SWAP5 (Sect. 2.13.8.1), is closed, and R8 is tested. If R8 = Ø, a TEST task was just started; R2, which contains the address of the CØ2 Q-block from the TEST command, is saved in TRQ, and a direct return from ADVICE is taken. The clearing of the CØ2 status is thus deferred, giving the TEST task temporary control of the console (Sect. 2.8.6). If, however, R8 = 1, GRAPHER was just reinstated after a TEST run. SVC6.ID, set to "GRAPHER" by OVRUNT, is set back to "PICKER" for normal sounding. Now if R2 = TRQ, the TEST task kept control of the console until it terminated; the OKADV exit (Sect. 2.8.2) is taken, and the TEST command CØ2 status is cleared, returning control of the console to COUNSEL. If R2 does not equal TRQ, the TEST task has already given up the console and ADVICE has cleared the TEST command CØ2 status. (See Fig. 2.6.) In the latter case R2 contains A(TFF) (Sect. 2.8.6); the TFF status cannot be cleared because GRAPHER has replaced the TEST task in partition 5. Therefore the direct return from ADVICE is taken.

2.13.8.1. Swap tasks in partition 5

SWAP5 (Fig. 2.24) is called alternately by OVRUNT and ENDTEST with the name of the partition 5 task to be loaded and started in TNAME, the filename field of SVC7 parameter block ASTSK5. SWAP5 assigns LU6 to this file with the "TSK" extension. If the LU assignment fails, the BADADV exit (Sect. 2.8.2) is taken from ADVICE. The SVC6 parameter block is set up to cancel the current partition 5 task, the name of which is always "GRAPHER". After canceling the task with an SVC6 call, SWAP5 delays 1 second to allow MT2 to clear the canceled task from the system. (One second seems excessive, but a shorter delay results in an SVC6 error when the new task is loaded.) After the delay at LO.ST, SWAP5 sets up the SVC6 parameter block to load and start the TEST task or GRAPHER. SVC6.LU is set to 6. The TEST task or GRAPHER is now loaded and started with a single SVC6 call. If the call is successful, SWAP5 returns to OVRUNT or ENDTEST. Otherwise, an error message is logged (Sect. 2.10.7), and if the operator continues through the error, the load-start is retried at LO.ST unless a checksum error occurred during the load. In this case, it is necesary to close LU6 and begin again at SWAP5 with the assignment of LU6.

SWAP5 exits with the address of the SVC6 parameter block, SSVC6, in R11. SWAP5 has no input register requirements.

2.13.8.2. TEST task parameter list

After a TEST task is started, it is queued by SOUNDER with the address of TSTLST (Sect. 2.13.8), giving the TEST task access to various subroutines, tables, etc., in SOUNDER memory. The TSTLST structure is in file SYSC:PCB.CAL at block label **TADDRS (Chapter 7). Module SBFRS and all of the TEST tasks are assembled with the TADDRS definitions described below. All TSTLST entries require 2 bytes.

DISPL.	DESCRIPTION
TA.ET	ET address (Sect. 7.3).
TA.ICT	ICT address (Sect. 7.1). A(CICT) is in ET.
TA.NBF	Number of TEST buffers, NTBFS (an SBFRS parameter).
TA.SØ2	SØ2LST address (Sect. 2.12.1).
TA.RID	Receiver gain profile tables identifier (see GORUN, Sect. 2.13.2).
TA.CAL	Address of calibration attenuation, CALAT.
TA.TXL	Address of low power transmitter attenuation, TXLAT (Sect. 2.10.6).

DISPL.	DESCRIPTION
TA.TXH	Address of high power transmitter attenuation, TXHAT (Sect. 2.10.6).
TA.ANT	Address of antennas table, ANT (Sect. 2.9.3).
TA.PSQ	Address of pulse set sequence table, PSEQ (Sect. 2.9.3).
TA.PMH	Address of minimum peak average, PKMIN (Sect. 2.10.2).
TA.PMX	Address of maximum peak average, PKMAX (Sect. 2.10.2).
TA.FMP	Address of protected frequencies map, FRQMAP (Sect. 2.12.9).
TA.GMP	Address of subroutine GETMAP (Sect. 2.13.9).
TA.SMP	Address of protected frequencies map flag, MAP (Sect. 2.13.9).

2.13.9. Protected frequencies map setup

GETMAP is called by INIT (Sect. 2.13.1, OVF (Sect. 2.13.4), and whenever a sounding is initiated by GO, VUF, or KRUN (Sect. 2.13.2) to ensure that the correct protected frequencies map is in FRQMAP (Sect. 2.12.9). The DELF and PSEQ commands also force a call to GETMAP (see NEWMAP, Sect. 2.8.4.5).

Input RØ: flag for desired map ("I" or "K").
SYS1:IHUSH.SND: I-mode map.
SYS1:KHUSH.SND: K-mode map.

Output MAP: map flag, from input RØ.
FRQMAP: protected frequencies map adjusted by ADJMAP.
RØ-R14: returned intact using storage area REGS.
R15: return address.
Error message: if LU assignment or disc read fails.

Method Refer to Figure 2.25.

GETMAP compares RØ to MAP, the current map flag. If RØ = MAP, the desired map is already in FRQMAP, and GETMAP returns. Otherwise MAP, the first byte in the filename field of the SVC7 LU assign parameter block, is set ot RØ, and the registers are saved in REGS. FRQMAP is zeroed, or initialized to protect all frequencies. (Should the operator continue through a GETMAP error (Sect. 2.10.7), sounding is prevented.) GETMAP now assigns LU6 to the appropriate base map file on disc and reads it into FRQMAP. CKMAP is called with call code 1 (Sect. 2.10.11) to verify that the base, or unadjusted map, is intact. (See TEST task FRMAP, Sect. 10.3.) If the base map is good, subroutine ADJMAP is called to adjust it to the current deltaF and pulse set sounding sequence. GETMAP then restores the registers from REGS

and returns.

ADJMAP is called by GETMAP to adjust the base map in FRQMAP to the current delta frequency, set by command DELF, and pulse set sounding sequence, set by command PSEQ. This is done so that protected frequencies are not violated by sounding on an adjacent unprotected base frequency plus or minus multiples of the delta frequency.

Input PSEQ: the pulse set sounding sequence table (Sect. 2.9.3).
NPTYPE: number of pulses per pulse set from SNDR (Sect. 7.6).
ICT.DELF: the delta frequency.
LOIX: the minimum frequency index from SNDR.
FRQMAP: base protected frequencies map as read from disc (see Sect. 2.12.9).

Output FRQMAP: protected frequencies map adjusted to PSEQ and ICT.DELF.
RØ-R14: returned intact using storage area DREGS.
R15: return address.

Method Refer to Figure 2.25.

ADJMAP saves the registers and initializes to scan table PSEQ. In the loop at MAXLP, the greatest plus-deltaF factor is put into R11, and the greatest minus-deltaF factor is put into R1Ø. (See command PSEQ, Chapter 8.) The delta frequency in hundreds of Hz is computed in R3 and R5 from ICT.DELF in kHz. The greatest minus-deltaF is computed as R3*R1Ø; the greatest plus-deltaF is computed as R5*R11. The results are saved as 32-bit values in register pairs (R2,3) and (R4,5), respectively. Storage area BUFR is used to temporarily hold base frequency indexes that need protecting. Its index is initialized in R6; R7 is initialized from LOIX, so that the scan of FRQMAP at ADJNXT (Fig. 2.25, connector A) begins at the start of FRQMAP.

At ADJNXT, 1 is added to R7 which is copied into R14. (Both MAPFIX and NTOFR require the frequency index in R14.) The base frequency index in R14 is passed through MAPFIX (Sect. 2.10.10). If the frequency is already protected, the next one is tested at ADJNXT. If the frequency is unprotected, CKBASF calls NTOFR (Sect. 2.11). The greatest minus-deltaF in (R2,3) is subtracted from the frequency, in hundreds of Hz, returned by NTOFR in (R12,R13), and the difference frequency is converted back to an index by FRTON. The frequency index returned by FRTON in R13, and copied into R14, represents the lowest frequency that would be used in a pulse set based on the index in R7. If MAPFIX finds this lowest frequency to be protected, the top of a protected band would be violated by sounding on the frequency index in R7.

Therefore, the base frequency itself must be protected, and R7 is saved in BUFR at SVBIX.

In like manner, the highest frequency that would be used in a pulse set based on R7 is found by adding (R4,5) to the base frequency. If the highest frequency would encroach upon the bottom of a protected band, the base index, R7, is saved in BUFR at SVBIX.

When all base frequencies have been thus tested, MAPFIX returns a negative condition code, indicating that the index in R14 is out of range, or that the end of FRQMAP has been sensed. At this point, R6 contains the BUFR index of the last entry made by SVBIX and is the limit of the loop of PRFRQ which protects the base frequencies found to encroach upon protected bands. At the end of the PRFRQ loop, FRQMAP has been thus adjusted to the current deltaF and PSEQ table. CKMAP (Sect. 2.10.11) is called with call code Ø to compute and set the check bytes for the adjusted map. The registers are restored from DREGS, and ADJMAP returns.

2.13.10. SOUNDER time initialization

All time-keeping in SOUNDER is based on the clock in the TSG, which is read via DIO 26. The TSG clock is read whenever a sounding is initialized by a GO, KRUN, or VUF command (Sect. 2.13.2). Thereafter, as long as SOUNDER is executing in the SPC loop (Sect. 2.7.2), subroutine CLOCK (Sect. 2.10.9) marks time according to the time value of a TSG tick, or the time between TSG interrupts.

Twelve values, or time elements, are read from the TSG clock in BCD format. A specific BCD time element is selected by writing its function code to DIO 26. Table 2.6 lists the BCD time elements by function code.

Subroutine RDCLK (Sect. 2.13.10.1) performs one complete read of the TSG clock via DIO 26; subroutine RDTIME calls RDCLK and initializes ET.SEC, HOUR, HRSEC, and FRSEC. (See CLOCK, Sect. 2.10.9.)

Input NOTSG: set by command NOTSG if TSG not in system. (See command NOTSG, Chapter 8, and Sects. 2.7.2.3 and 2.7.2.11.)
TIME1: 12-byte TSG clock read, from RDCLK (see Table 2.6).
TIME2: clock read for comparison, from RDCLK.

2.13.10. SOUNDER time initialization

HEX FUNCTION CODE	BCD TIME ELEMENT
Ø	tenths of seconds
1	units of seconds
2	tens of seconds
3	units of minutes
4	tens of minutes
5	units of hours
6	tens of hours
7	units of days
8	tens of days
9	hundreds of days
A	units of years
B	tens of years

Table 2.6. BCD TIME ELEMENTS

XSECS: table of seconds per BCD time element for converting clock read to seconds into the current year. XSECS is in the order shown by Table 2.6, from units of seconds through hundreds of days.
ICT.TSGC: TSG command byte pace code, used to set FRSEC (see CLOCK, Sect. 2.10.9).
TIKSEC: table of TSG ticks per second (see Sect. 2.10.9).

Output FRSEC: fractional second in TSG ticks, based on tenths of seconds as read from TSG clock (time element Ø).
HRSEC: seconds into hour, from time elements 1 through 4.
HOUR: hour of day, from time elements 5 and 6.
ET.SEC: current time in seconds from 1/1/77 (32 bits), computed from time elements 1 through B and XSECS table.
Error message: if clock read fails after 1Ø tries.
RØ-R14: returned intact using storage area REGS.
R15: return address.

Method Refer to Figure 2.26.

RDTIME checks the NOTSG flag, and if NOTSG = Ø, the TSG is in the system, and control goes to TSGIN. If NOTSG is not Ø, the clock cannot be read, and control goes to CLKØ where ET.SEC, FRSEC, HOUR, and HRSEC are all cleared to Ø, thereby setting the current time to the beginning of 1/1/77.

At TSGIN the input registers are saved, and RWULM (Sect. 2.10.3) is called to put the DIO interface in the read/write state for the TSG clock read. R7 is set to A(TIME1), the first read buffer for subroutine RDCLK. (TIME1 is the first 12 bytes of BUFR, the

general purpose buffer and storage area in SBFRS; TIME2 is BUFR + 12 through BUFR + 23.) R9, the clock read retry count, is initialized to 10.

At RD1, RDCLK is called to read the TSG clock into TIME1. R7 is now set to A(TIME2), and at RD2, R9 is decremented by 1. If R9 has been counted down to 0, an error message is logged, and if the operator continues through the error (Sect. 2.10.7), control goes to CLK0, described above. Otherwise RDCLK is again called to read the clock, this time into TIME2. TIME1 + 0 is compared to TIME2 + 0, and unless the time has changed, control goes back to RD2. If the time has changed, RDCLK is called with R7 = A(TIME1). Now all 12 bytes of TIME1 and TIME2 are compared, and if the time has again changed, control goes back to RD1. Otherwise the TSG clock has been successfully read and as close as possible to the tenth of a second. R10 and R11 are cleared to 0 and will be used as a 32-bit accumulator for seconds past 1/1/77. R1 is initialized to 1 to index TIME1 from the units-of-seconds time element; R4 is cleared to 0 to index table XSECS from its first 32-bit entry.

At SECLP (Fig. 2.26, connector B), a BCD time element is loaded into R5 from TIME1 indexed by R1. R6 and R7 are set to the 32-bit entry of XSECS indexed by R4. If the time element in R5 is not 0, SCLP1 adds (R6,7) to (R10,11) and decrements R5 by 1. The add is repeated until R5 goes to 0, computing the 32-bit product of the BCD time element and its corresponding XSECS entry. The SECLP loop computes this product for each time element in TIME1 from units of seconds (element 1) through hundreds of days (element 9), and accumulates the products in (R10,11). When R1, the TIME1 index, is 4 (tens of minutes), R11 is copied into HRSEC, the number of seconds into the hour. When R1 goes to 10, R7 is set to the current year from elements A and B in TIME1, and R6 is set to 1977, the base year. At CKYR, if R6 = R7, control goes to ADJSEC. Otherwise, one year's worth of seconds is added to R10-11 for the year in R6, allowing for the leap year. One is added to R6, and control goes back to CKYR.

At ADJSEC the base year, 1977, has been counted up to the current year. Because 1 January is day 1, not day 0, a day of seconds is subtracted from R10-11. R10-11 is put into ET.SEC, the current time expressed in seconds from 1/1/77. R1 is set to the TSG pace code times two to index table TIKSEC. (See CLOCK, Sect. 2.10.9.) R3 is set to the tenths-of-seconds element from TIME1 + 0. FRSEC, the current fractional second expressed in TSG ticks is computed as follows:

2.13.10. SOUNDER time initialization

FRSEC = (R3*TIKSEC(R1))/1Ø

HOUR is set from time elements 5 and 6 in TIME1 (GTAGC, Sect. 2.10.1).

Finally, the registers are restored from REGS, and RDTIME returns.

2.13.10.1. TSG clock read subroutine

RDCLK is called by RDTIME to do a complete read of the TSG clock using DIO address 26.

Input R7: address of time storage buffer.
DIO 26 time data bytes: high-order 4 bits are function code corresponding to BCD time element in low-order 4 bits.

Output Time element function code: written to DIO 26 to select time element.
TSG time: stored at the address in R7, in the order of Table 2.6.
R7-R13: returned intact.
R14: return address from R15.
Error message: if DIO 26 returns incorrect function code.

Method Refer to Figure 2.26.

RDCLK sets R1 to the DIO interface address (ULM from SNDR, Sect. 7.6) and initializes the time-of-day (TOD) function code by clearing R2 to Ø. Bits 8-11 of R2 hold the function code.

At RDCLP the left byte of R5 is set to the TOD function code in the right byte of R2; the right byte of R5 is set to 26 (hex). DIO 26 is addressed and the TOD function is simultaneously selected by writing R5 to the DIO interface device (WHR R1,R5). After a delay of about 2 ms (Sect. 2.13.6) to permit the TSG clock hardware to select the specified TOD function, the DIO write is repeated, this time to latch the TSG reply in the DIO interface. After a delay of about 6 microseconds, 8 bits of data are read from DIO 26. If the fucntion code in the high order 4 bits differs from that requested (R2, bits 8-11), an error message is logged with the error code of the form ØMØN (hex). M is the requested TOD function code, and N is the incorrect code returned by the TSG. If the operator continues through the error (Sect. 2.10.7), the read of TOD element M is tried again. If the fucntion code is correct, it is right-shifted to form a byte index into the storage area (R7); and the BCD time element, the

low-order 4 bits read from DIO 26, is saved for RDTIME. At the end of RDCLP, 1Ø (hex) is added to R2 to increment the TOD function code, and unless R2 > BØ (hex), the RDCLP loop continues. Otherwise, RDCLK returns to RDTIME with a complete TSG clock read stored in TIME1 or TIME2, depending on R7.

2.13.11. SBFRS storage

In addition to the code described in Sections 2.13.1 through 2.13.10, module SBFRS contains the following:

LOGIN	SOUNDER's start-up message (Sect. 2.13.1).
ASCON	Assign LU1 to console parameter block (Sect. 2.13.1).
ASTSG	Assign 2,TSG: (Sect. 2.13.1).
ASEKO	Assign 3,EKO: (Sect. 2.13.1).
ASFEP	Assign 4,FEP: (Sect. 2.13.1).
ASGAIN	Assign 5,SYS1:RGAIN.SAV (Sect. 2.13.1).
STBLK	The SØØ Q-block (Sect. 2.13.1).
C.ID	COUNSEL task name for SØØ queue.
TIMØ1	TSG clock read buffer, used by SETIME (Sect. 2.13.1).
TIMØ2	Clock read buffer to check TIMØ1, used by SETIME.
SET.TIME	SVC2 parameter block, used by SETIME to set system time.
HH	Hours-minutes-seconds string in SET.TIME (1Ø bytes).
MM	Months-days-years string in SET.TIME (8 bytes).
TD	Seconds of day in SET.TIME (4 bytes).
D3Ø55	Constant used by SETIME to compose MM.

All of the above are stored with the INIT code and are lost once sounding starts.

The following are included in the SBFRS overlay (Sect. 2.8.3).

D983Ø	15*65536/1ØØ, the constant used by subroutine KMTOUS and USTOKM (Sect. 2.13.4.1) in converting ranges between km and microseconds.
UNHPB	Parameter block to convert a number to a hex ASCII string in BUFR, used by DISPKF (Sect. 2.13.4.1).
UNDPB	Parameter block to convert a number to a decimal ASCII string in BUFR, used by DSPDNO and USTOKM (Sect. 2.13.4.1).
KMSG	KMODE table entry displayed by DISPKF (Sect. 2.13.4.1).
KMSGL	Length of KMSG (see WRLOG, Sect. 2.10.8).
DECNO	Decimal number message displayed by DSPDNO (Sect. 2.13.4.1).

2.13.11. SBFRS storage

TIME1	BUFR + Ø through BUFR + 11, used by RDTIME (Sect. 2.13.10).
TIME2	BUFR + 12 through BUFR + 23, used by RDTIME.
XSECS	Table of seconds per TSG clock time element, used by RDTIME (Sect. 2.13.10).
SMIN	Units of minutes entry in XSECS (seconds/minute).
SHR	Units of hours entry in XSECS (seconds/hour).
SDAY	Units of days entry in XSECS (seconds/day).
SYR	Seconds/365 days.
THFCTR	Table of valid FEP threshold factors times 1Ø (see LDFEP, Sect. 2.13.5.1).
THRSH	Table of FEP program bytes corresponding to THFCTR entries.
SCTRL	Table of FEP control bytes sent after program bytes by LDFEP.
FCTRL	Table of final control bytes sent to FEP by LDFEP.
FDIOA	DIO address table corresponding to FCTRL.
BTSCAN	BOOT command argument mnemonic table used by OVBOOT (Sect. 2.13.5).
BTVECT	Subroutine address table, corresponding to BTSCAN, used by OVBOOT.
BOOTPB	Parameter block for scanning BOOT command argument.
.SND	Default TSGMEM and FEPMEM file extension "SND" (see OVBOOT).
RDFPB	Read SYS1:FEPMEM.xxx (Sect. 2.13.5.1).
RDTPB	Read SYS1:TSGMEM.xxx (Sect. 2.13.5.2.2).
NREC	TSGMEM.xxx file record number.
RDHUSH	Read SYS1:IHUSH.SND or SYS1:KHUSH.SND (Sect. 2.13.9).
RDMAG	Read magtape (Sect. 2.13.3).
MEMAD	Magtape read PCT/EKO buffer address in RDMAG parameter block.
ASFMEM	Assign LU6 to SYS1:FEPMEM.xxx parameter block (Sect. 2.13.5.1).
FEXT	FEPMEM file extension field in ASFMEM.
ASTMEM	Assign 6,SYS1:TSGMEM.xxx (Sect. 2.13.5.2).
TEXT	TSGMEM file extension field in ASTMEM.
ASTSK5	Assign 6,GRAPHER/TEST.TSK (Sect. 2.13.8).
TVOLID	.TSK file volume identifier in ASTSK5 parameter block (last byte of volume field, "1" or "C").
TNAME	Filename field in ASTSK5.
ASMAG	Assign 6,MAG: (Sect. 2.13.3).
SØ8	SØ8 Q-block (Sect. 2.13.4).
P.ID	PICKER task name for queueing during normal sounding (Sects. 2.13.1 and 2.13.8).

G.ID	GRAPHER task name for SØ8 queue (see QSØ8, Sect. 2.13.4.1), and for queueing TEST tasks (Sect. 2.13.8).
TSTLST	TEST task parameter list (Sect. 2.13.8.2).
FAKECQ	Fake COUNSEL CØØ Q-block for GRAPHER restart (ENDTEST, Sect. 2.13.8).
REGS	General register storage.
BTREGS	Register storage for LDFEP (Sect. 3.13.5.1) and BTTSG (Sect. 2.13.5.2).
OVCKSUM	Checksum byte for SBFRS overlay and last byte of overlay written to disc. (See INIT, Sect. 2.13.1, and GETOV, Sect. 2.8.3.)

Following OVCKSUM is a general purpose 256-byte buffer, BUFR. ET.ABUF (Sect. 7.3) is set to BUFR + 256 by INIT (Sect. 2.13.1).

2.13.12. PCT's and EKO buffers

The Product One version of module SBFRS reserves memory for 6 PCT/EKO buffer areas. Each area is PCT. + EKOSIZE bytes long. The PCT length (PCT.) is defined by the PCT STRUC in SYSC:PCB.CAL; EKOSIZE is an SNDR parameter (Sect. 7.6). Each EKO buffer is contiguous to its associated PCT with the PCT stored first. The PCT/EKO buffers are numbered in the reverse of the storage order and are allocated for use in the order that they are numbered (Fig. 2.2). For a normal sounding the PCT/EKO buffers are used in round-robin fashion: 1, 2, 3, 4, 5, 6, 1, 2,...; only 4 PCT/EKO buffers are used during a TEST sounding (Sect. 10.2): 1, 2, 3, 4, 1, 2,.... The order of allocation is established by circular lists SØ2LST and PCTLST (Sect. 2.12.1). The start of the PCT/EKO buffers are defined at the very end of module STBLS as labels PCT and EKOBFR, which are used to set up PCTLST and SØ2TBL (Sect. 2.12.2). This means that module SBFRS must be loaded immediately after STBLS. The loading order of SOUNDER task modules, or object files, is controlled by file SYSC:TETSNDR.CSS (Sect. 9.4).

A PCT is associated with each transmitted pulse and hence with one buffer of EKO data from the FEP. NXTPLSE (Sect. 2.7.2.2) compacts the pulse configuration data established by NXTFRQ (Sect. 2.9) into a PCT just before the DIO devices are set up and a pulse (P) is transmitted as the result of a TSG start. One pulse later, when the FEP has finished its work on the echo data from pulse P, the transfer of the data from the FEP to the associated EKO buffer is started by STEKO (Sect. 2.7.2.10). Two pulses after pulse P, the transfer is complete and PICKER is queued with an SØ2 Q-block containing the addresses of both the PCT and its associated EKO buffer (QPCKR, Sect. 2.7.2.6). This sequence of events is followed for both I- and K-mode soundings.

2.13.12. PCT's and EKO buffers

The PCT layout is described in Section 7.2, the EKO buffer in Section 3.4. TEST tasks EXPCT and PLOTEKO (Sect. 10.3) are used to examine PCT's and EKO buffers, respectively.

2.14. SOUNDER Disc Files and LU Assignments [D.C.W.]

SOUNDER uses logical units 1-6, all assigned under program control. LU's 1-5 are assigned permanently by INIT (Sect. 2.13.1) as follows:

LU	DEVICE/FILE NAME	SVC7 PARAMETER BLOCK-MODULE
1	CON: (console)	ASCON-SBFRS
2	TSG:	ASTSG-SBFRS
3	EKO:	ASEKO-SBFRS
4	FEP:	ASFEP-SBFRS
5	SYS1:RGAIN.SAV	ASGAIN-SBFRS

The SVC1 parameter blocks dedicated to LU's 1-5 are:

LU	SVC1 PARAMETER BLOCK-MODULE
1	None (WRLOG, Sect. 2.10.8, uses SVC2 parameter block WRLPB-SSUBS).
2	STTPB(write, proceed)-SPIPE, WTTPB(wait)-SPIPE.
3	STEPB(read, proceed)-SPIPE, WTEPB(wait)-SPIPE.
4	STFPB(dummy read, proceed)-SPIPE, WTFPB(wait)-SPIPE.
5	SRGPB(write, wait)-SADVC.

The TSG, FEP, and EKO device starts and waits are discussed under Section 2.7.2. File RGAIN.SAV is written by command RGAIN (Chapter 8 and Fig. 2.9) and is read and displayed by TEST task AGC (Sect. 10.3). An RGAIN.SAV record consists of RAGCID, the sounding number identifier set by GORUN (Sect. 2.13.2) and the receiver AGC profile tables, RAGC and RAGCK (Sect. 2.12.7). RGAIN.SAV is written by record number, two records per sector, and is terminated by a record with RAGCID set to -1. The RGAIN.SAV record number is initialized to Ø each time the HF Radar software is loaded.

SOUNDER assigns LU6 temporarily to the files that are listed below.

FILE NAME OR DEVICE	SVC7 PARAMETER BLOCK-MODULE	SVC1 PARAMETER BLOCK-MODULE
SYS1:SOVLAY.SND	ASOVLAY-SADVC	SOVPB(read/write)-SADVC
SYS1:IHUSH.SND	ASMAP-SADVC	RDHUSH(read)-SBFRS
SYS1:KHUSH.SND	ASMAP-SADVC	RDHUSH(read)-SBFRS
SYS1:TSGMEM.SND	ASTMEM-SBFRS	RDTPB(read)-SBFRS
SYS1:FEPMEM.SND	ASFMEM-SBFRS	RDFPB(read)-SBFRS
SYSC:<TEST>.TSK	ASTSK5-SBFRS	(TSK file read with SVC6 load)
SYS1:GRAPHER.TSK	ASTSK5-SBFRS	(TSK file read with SVC6 load)
MAG:	ASMAG-SBFRS	RDMAG(read)-SBFRS

All read/write requests to LU6 wait until the device is free before initiating the I/O, then wait until the I/O is complete before returning to the requesting routine. All routines that assign LU6 also close it with parameter block CLSPB that resides in module SADVC.

SYS1:SOVLAY.SND is written by INIT (Sect. 2.13.1) and read by GETOV (Sect. 2.8.3).

In the SBFRS module code, label OVLAY is equated to the start of PCT/EKO buffer 6 (Fig. 2.2), and routine OVGO, the first overlayed advice routine (Sect. 2.13.2), is loaded at OVLAY. File SYS1:SOVLAY.SND contains the SBFRS overlay, that is, OVLAY through OVCKSUM (Sect. 2.13.11).

Files SYS1:IHUSH.SND and KHUSH.SND are written by TEST task FRMAP (Sect. 10.3) and are read by GETMAP (Sect. 2.13.9).

SYS1:TSGMEM.SND contains a copy of the current TSG pages Ø-2 on the TSG PROM. This file used to be read by BTTSG (Sect. 2.13.5.2); BTTSG now reads directly from TSG page zero unless the command BOOT TSG,SND is given. The file is still read by TEST tasks SNRIO and TAPEKO (Sect. 10.3); in the future SNRIO and TAPEKO will also read directly from the TSG. User versions of TSG page zero are stored in files SYS1:TSGMEM.xxx, where "xxx" is any 3-character alphanumeric ASCII file extension. Such files are written by utility task DUTIL (Sect. 10.4), and are read by BTTSG for command BOOT TSG,xxx.

SYS1:FEPMEM.SND contains the default FEP thresholding program and is written by utility task FEPMEM (Sect. 10.9). The file is read by LDFEP (Sect. 2.13.5.1). A user version of the FEP program, SYS1:FEPMEM.xxx, is invoked by command BOOT FEP,xxx. Utility task FEPTEST (Sect. 10.10) reads file SYSC:FEPMEM.SND or SYSC:FEPMEM.xxx. SYS1:FEPMEM.NOT and SYSC:FEPMEM.NOT contain a version of the FEP program that does no thresholding on the data. Command BOOT FEP,NOT causes LDFEP to read the former; the latter can be read by FEPTEST.

2.14. SOUNDER Disc Files and LU Assignments

File SYSC:<TEST>.TSK, where <TEST> is a valid argument of the TEST command, is read by SWAP5 (Sect. 2.13.8.1) via the SVC6 call to load and start the TEST task. The file is output by the MT2 utility TET (Sect. 10.3).

SYS1:GRAPHER.TSK is also read by SWAP5, as described above, when a TEST task is terminated and GRAPHER is reinstated.

MAG:, the magnetic tape device, is read in response to the TPRUN command (Sect. 2.13.3). The magtape must have been written by TEST task TAPEKO (Sect. 10.3).

To summarize, the minimum list of disc files required by SOUNDER is shown below. All of these files are on volume SYS1, the fixed disc.

FILE NAME	TYPE	BYTES/RECORD	CONTIGUOUS BLOCKS
SOVLAY.SND	C	-	20
IHUSH.SND	C	-	4
KHUSH.SND	C	-	4
TSGMEM.SND	C	-	3
FEPMEM.SND	I	12	-
RGAIN.SAV	I	128	-

If all 6 of the above files are intact and if SYS1 is marked on (T4, Sect. 9.1), SOUNDER will run, and commands BOOT TSG, BOOT FEP, and RGAIN will execute. To fully implement the BOOT commands and the TEST system, the following files are needed as well.

FILE NAME	TYPE	BYTES/RECORD	CONTIGUOUS BLOCKS
SYS1:TSGMEM.xxx	C	-	3
SYS1:FEPMEM.NOT	I	12	-
SYS1:FEPMEM.xxx	I	12	-
SYS1:WHUSH.SND	C	-	4
SYSC.SOUNDER.MAP	I	80	-
SYSC:HELP.MNU	I	80	-
SYSC:<TEST>.TSK	C	-	varies
SYS1:GRAPHER.TSK	C	-	varies
MAG:	-	varies	-

Disc files with extension "xxx" are user-written; all others are provided by SEL with each HF Radar software update. WHUSH.SND is the internationally protected frequencies map used by FRMAP (Sect. 10.3); TEST task HELP (Sect. 10.3) reads SOUNDER.MAP (TETSNDR, Sect. 9.4), HELP.MNU, and the <TEST>.TSK files.

Finally, the utility task FEPTEST (Sect. 10.10) always reads versions of the FEPMEM files from SYSC, the removable or top disc. These files, however, are not required by SOUNDER or the TEST system.

2.15. SOUNDER's Drivers [D.C.W.]

SVC1 calls within SOUNDER's SPC loop (Sect. 2.7.2) that start and wait for the TSG, FEP, and EKO devices, invoke drivers at the MT2 OS level. Because these control non-standard devices, they were written at SEL and have been incorporated into the MT2 Executive (EXEC) following conventions described in detail in Interdata's OS/16 MT2 Guide to Writing Drivers.

A Device Control Block (DCB) is associated with each driver. The DCB contains flags, routine addresses, and register storage required by the driver to operate the device. It also contains short blocks of code executed when the interrupt service routines (ISR) are entered and exited.

Each of the three drivers described below contains four sections (see Fig. 2.28). The first section (U4xDVR) is the initialize routine that is invoked by the SVC1 call to start the device. It sets up registers to be used later by the driver and saves R2-R15 in the DCB. U4xDVR executes with external interrupts enabled.

The second section, which is the first ISR, is invoked by U4xDVR simulating an interrupt (SINT) to the device. The second section (ISR1xxx), entered with interrupts disabled, starts the device, sets the timeout count, and enables and arms the device interrupt. ISR1xxx exits through the DCB to the instruction immediately following the SINT in U4xDVR, with external interrupts again enabled. U4xDVR in turn exits to the task scheduler through EXEC routine IOTWAT, which updates the I/O wait thread. The scheduler eventually returns control to the instruction immediately following the SVC1 start call, and SOUNDER continues to execute.

The third section (ISR2xxx) is entered either as a result of a device interrupt signaling completion of the device action, or as the result of a SINT to the device from the system clock ISR signaling that the timeout count, DCB.TOUT, has been decremented to Ø. ISR2xxx determines how it was entered and takes appropriate action. In either case ISR2xxx adds the address of the DCB + 1 to the top of the system I/O termination queue, LIOTRM, and the system is informed that the SVC1 device start has been resolved. ISR2xxx exits through the DCB and the scheduler eventually returns control to the point of interruption. If the SVC1 wait call is made before ISR2xxx is entered, SOUNDER is not scheduled to run until the system sees the appropriate entry on LIOTRM. If the device

has already finished when the SVC1 wait call is made, the system determines from the DCB that the driver is inactive and returns to SOUNDER directly.

The fourth section (U4xTRM) is entered as a result of command CANHF (Chapter 8), to which SOUNDER responds by executing an SVC3 call to cancel itself, or as a result of a CANCEL command to the Command Processor. In the latter case, the system executes an SVC6 call to cancel, or delete, the task. In either case, part of the task-canceling process is to terminate any ongoing I/O, and this is done at the driver level by the termination routine, U4xTRM.

The addresses of the first and fourth sections, or routines, are preset in the DCB's as a result of the SYSGEN process (Sect. 9.5). U4xDVR and ISR1xxx set R15 to A(ISR1xxx) and A(ISR2xxx), respectively. U4xDVR saves R2-R15 in the DCB explicitly, and ISR1xxx exits through DCB.LEAV, which saves R8-R15 in the DCB. Thus the system is informed of which driver section, or routine, is to be executed as the result of the next external event (SVC1 call, interrupt, or cancel) requiring action at the driver level.

The TSG, FEP, and EKO drivers are described below. Refer to Figure 2.28.

2.15.1. The TSG driver

U4ØDVR is the initialize routine for the TSG driver. U4ØDVR sets R15 to A(ISR1TSG), the routine invoked when the SINT is executed; R14 to A(DCB) from R1; R13 to A(TSG start parameter block) from R3; R12 to 37 (hex), the TSG device address from R6; and R1Ø to SVC1.SAD, the address of the TSG command byte, from R13. R1-R7 are set up on entry to U4ØDVR by EXEC. (See OS/16 MT2 Guide to Writing Drivers.) U4ØDVR saves R2-R15 in DCB.RSAV, as required by the OS, then executes a SINT to device 37, invoking ISR1TSG. After ISR1TSG executes, U4ØDVR exits to EXEC routine IOTWAT.

ISR1TSG gets the TSG command byte from the buffer pointed to by R1Ø and sends the command byte to the TSG via DIO 24. After a delay of 2-3 microseconds, depending on the Interdata cycle time, ISR1TSG starts the NMI signal to the TSG (Sect. 2.13.5.2.5) by writing 6 to DIO 25. The NMI signal must last at least 1Ø microseconds, and in effecting the necessary delay, ISR1TSG does the following: sets DCB.TOUT, the timeout count, to 3 seconds; sets R15 to A(ISR2TSG), the routine to be executed on receipt of the TSG DONE interrupt; clears R9 to Ø on the assumption that the TSG start will not time out; adds 1 to R1Ø so that EXEC routine IODONE can compute the length of the data transfer; and sets R11 to A(DCB) + 1 from R14, to be added to the top of LIOTRM by ISR2TSG (below). ISR1TSG can

now end the NMI and does so by writing 2 to DIO 25. Finally, ISR1TSG outputs command 41 (hex) to device 37 and exits through DCB.LEAV. (Command 41 enables and arms the device 37 interrupt and leaves the DIO interface in the read/write state.)

ISR2TSG loads DCB.TOUT into R8, zeroes DCB.TOUT to terminate the timeout countdown by the clock ISR, and tests R8. If R8 is non-zero, the interrupt was TSG DONE, and control goes to IOTSAV. Otherwise, the TSG has timed out, and R8 is set to 8200 (hex). At IOTSAV, R9 (0 or 8200) is put into DCB.RSAV + 10 and becomes the exit value of R7 (device independent status) for EXEC routine IODONE. R11, A(DCB) + 1, is added to the top of LIOTRM, R15 is set to DCB.NOPI so that unsolicited interrupts will be ignored, and ISR2TSG exits through DCB.LEAV.

U40TRM disables and disarms the device 37 interrupt and leaves the DIO interface in the read/write state with command C1 (hex). R9 is set to 8400 (hex), the error code for an unrecoverable error, and control goes to IOTSAV in ISR2TSG (above).

2.15.2. The FEP driver

U41DVR is identical to U40DVR (Sect. 2.15.1), except that R12 is set to 77 (hex), the FEP/EKO device address. The SINT invokes ISR1FEP.

ISR1FEP senses the device 77 status, but in Product One has no need to examine it further. Because the SVC1 call to start the FEP only sets up the OS to wait for the FEP DATA READY interrupt, there is no actual FEP start in the driver; the FEP is started by the TSG. ISR1FEP simply sets DCB.TOUT to 3 and enables and arms the device 77 FEP DATA READY interrupt with command 58 (hex). Note that command 58 includes command bits 11 and 12 in the high (1) state. (See SINIT, Sect. 2.7.1.3, and LDFEP, Sect. 2.13.5.1.) ISR1FEP exits through DCB.LEAV with R15 = A(ISR2FEP).

ISR2FEP clears R9 to 0 on the assumption that the FEP has not timed out, loads R8 with DCB.TOUT, then zeroes DCB.TOUT. If R8, the entry value of DCB.TOUT, is non-zero, ISR2FEP has been entered as a result of the FEP DATA READY interrupt; otherwise, the FEP has timed out, and R9 is set to 8200 (hex). At IOTSAV, DCB.RSAV + 10 is set to R9 for IODONE (see ISR2TSG, Sect. 2.15.1). The A(DCB) + 1 is added to LIOTRM, R15 is set to DCB.NOPI, and ISR2FEP exits through DCB.LEAV.

U41TRM sets R9 to 8400 (hex), and since ongoing I/O is not in question, control simply goes to IOTSAV in ISR2FEP (above).

2.15.3. The EKO driver

U43DVR is identical to U40DVR (Sect. 2.15.1), except that R12 is set to the FEP/EKO device address, 77 (hex), R11 is set to the driver device address, that of SELCH F1, and the SINT invokes ISR1EKO.

ISR1EKO stops the SELCH with command 8, then writes SVC1.SAD, the EKO buffer starting address, and SVC1.EAD, the EKO buffer ending address, to the SELCH. These are the addresses put in AEKOBFR by STEKO (Sect. 2.7.2.10). The SELCH is an I/O device that can be shared among a number of devices for data transfers. For this reason, and even though SELCH F1 is dedicated to the EKO device, ISR1EKO associates the EKO device with SELCH F1 by reading the device 77 status. Finally, ISR1EKO sets DCB.TOUT to 3, starts the SELCH with command 30 (hex), and exits through DCB.LEAV with R15 = A(ISR2EKO). Although the SELCH is started here, the presence of the FEP DATA READY signal, which is tied to the FEP interrupt (Sect. 2.15.2), actually starts the data transfer from the FEP to one of SOUNDER's EKO buffers.

ISR2EKO sets R9 to 0 or 8200 depending on DCB.TOUT and zeroes DCB.TOUT. At NOTOUT, ISR2EKO reads the SELCH address register into R10 to get the address of the last data transferred, and unless R10 = SVC1.SAD, increments R10 by 1 for IODONE's data transfer length computation. If R10 = SVC1.SAD, no data were transferred, and IODONE will compute the transfer to be 0 (SVC1.SAD - R10). At ARM77, the FEP DATA READY signal is cleared with command 8 in case the data transfer was incomplete. (See note, below.) Also, because the EKO DONE interrupt disables the FEP interrupt (FEP and EKO share device address 77), the device 77 interrupt is re-enabled and re-armed with command 58 (ISR1FEP, Sect. 2.15.2). At IOTSAV, DCB.RSAV + 10 is set to R9 for IODONE, A(DCB) + 1 is added to LIOTRM, R15 is set to DCB.NOPI, and ISR2EKO exits through DCB.LEAV.

Note: two constraints are placed upon the length of the EKO data transfer. First, the first 16 bits of the FEP output memeory contain the number of half-words to be transferred. This count is decremented by 1 for each half-word transferred, and when the count goes to 0, the transfer is considered complete, and the EKO DONE interrupt is generated. Second, the SELCH is given the input buffer limits (ISR1EKO, above), and if during the data transfer, the buffer end address is reached, the SELCH terminates the transfer and issues the EKO DONE interrupt, even through the half-word count remains positive. However, the FEP DATA READY signal is automatically cleared only when the half-word count goes to 0, and the FEP DATA READY signal must be cleared before the TSG will issue a new start to the FEP. Therefore, in case the EKO data transfer is incomplete from the FEP's standpoint, the FEP DATA READY signal must be explicitly cleared after the EKO DONE interrupt is sensed so that the data pipeline

can function.

U43TRM sets R9 to 8400 for IOTERM's R7, stops the SELCH with command 8, and exits through IOTSAV (ISR2EKO, above).

2.16. Flowcharts

2.16. Flowcharts [D.C.W.]

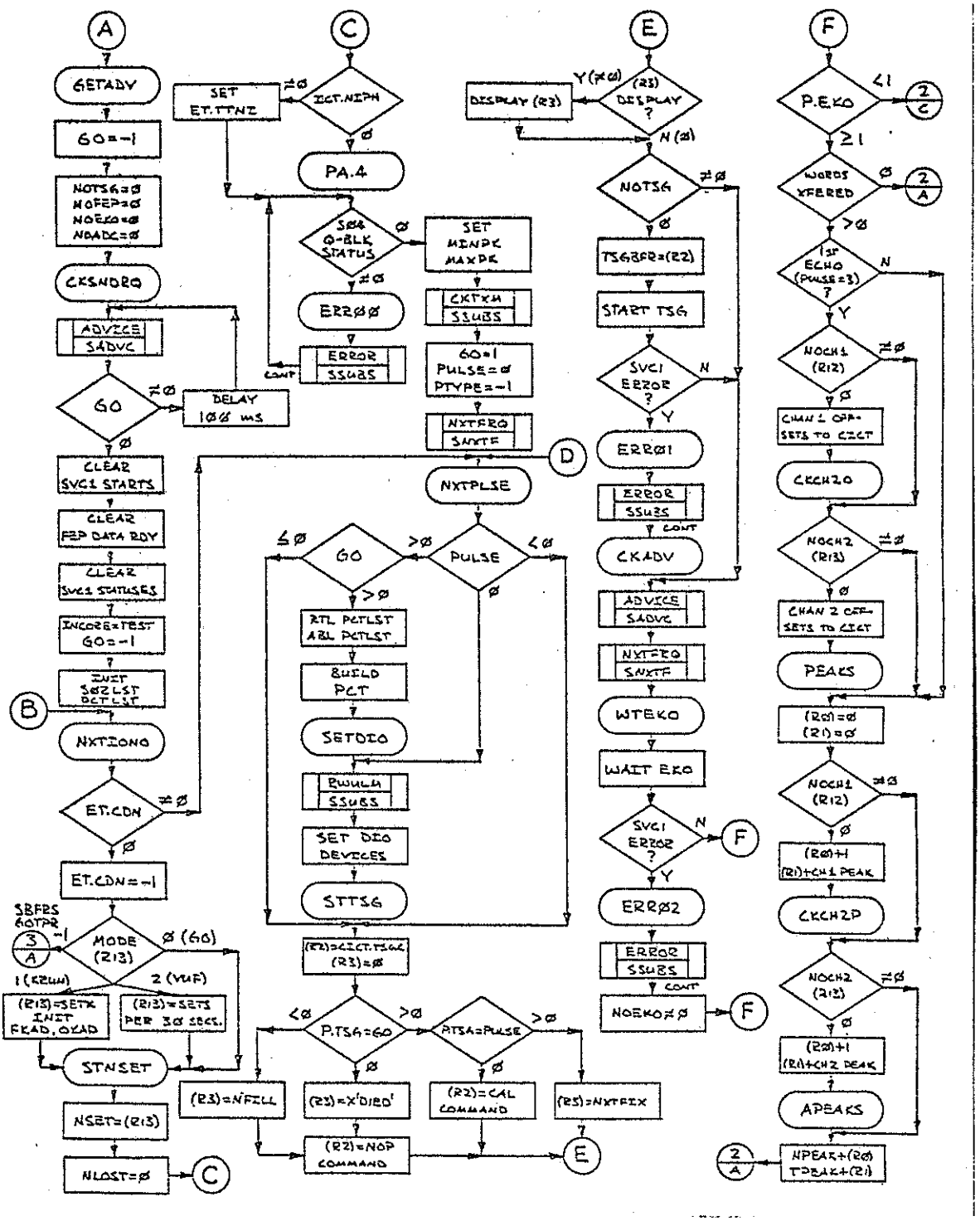


Figure 2.3. SPIPE Sheet 1

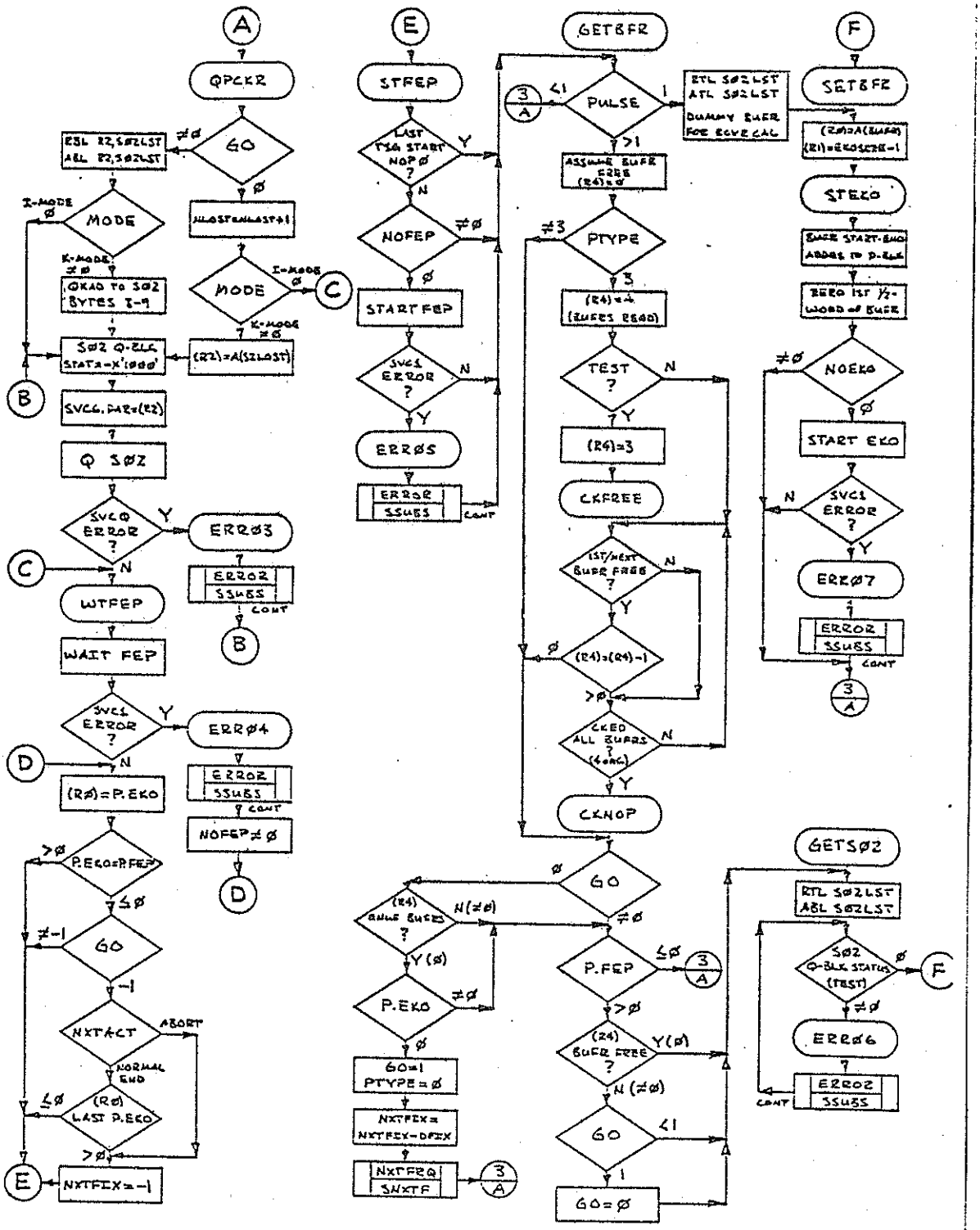


Figure 2.4. SPIPE Sheet 2

2.16. Flowcharts

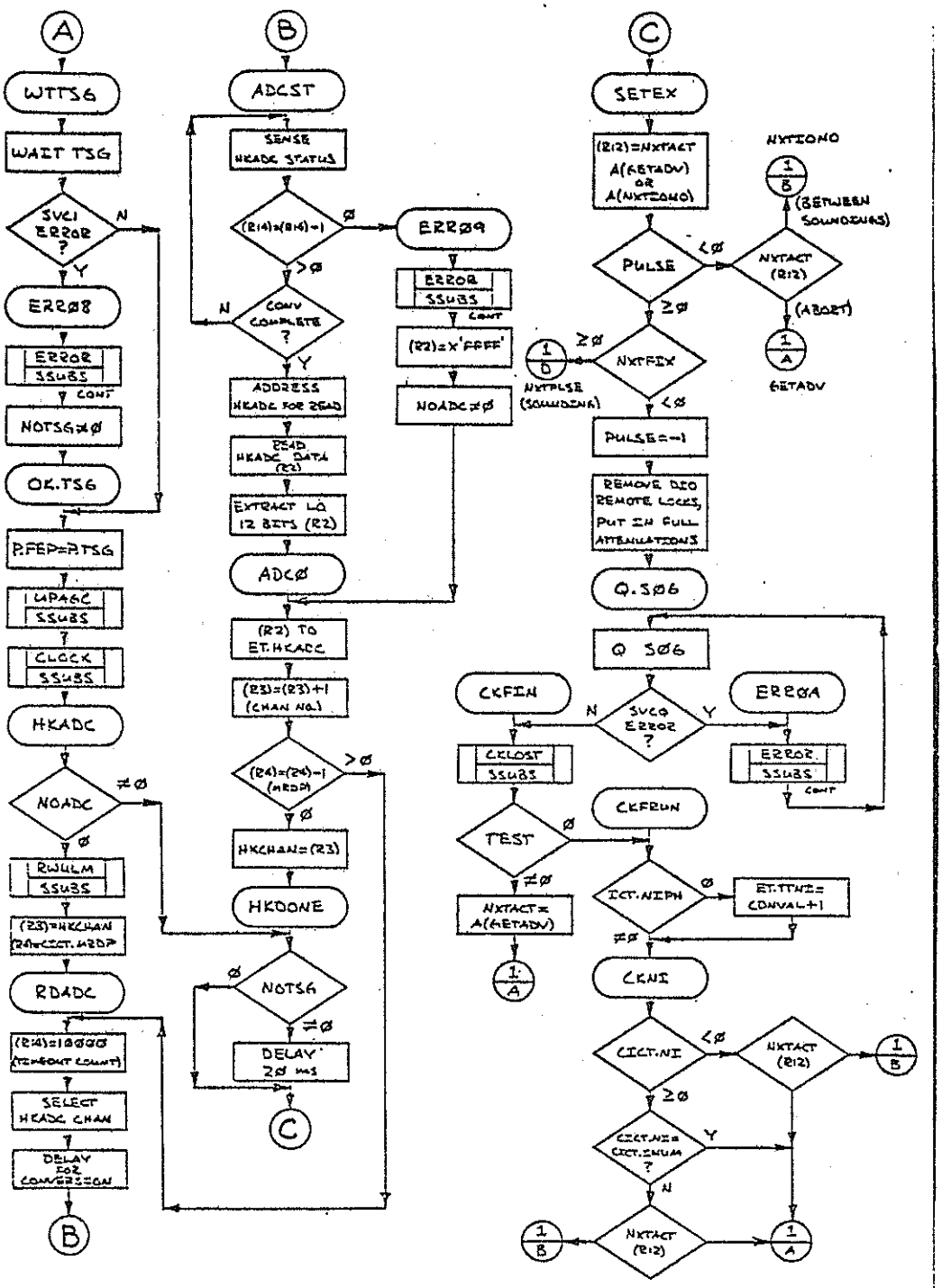


Figure 2.5. SPIPE Sheet 3

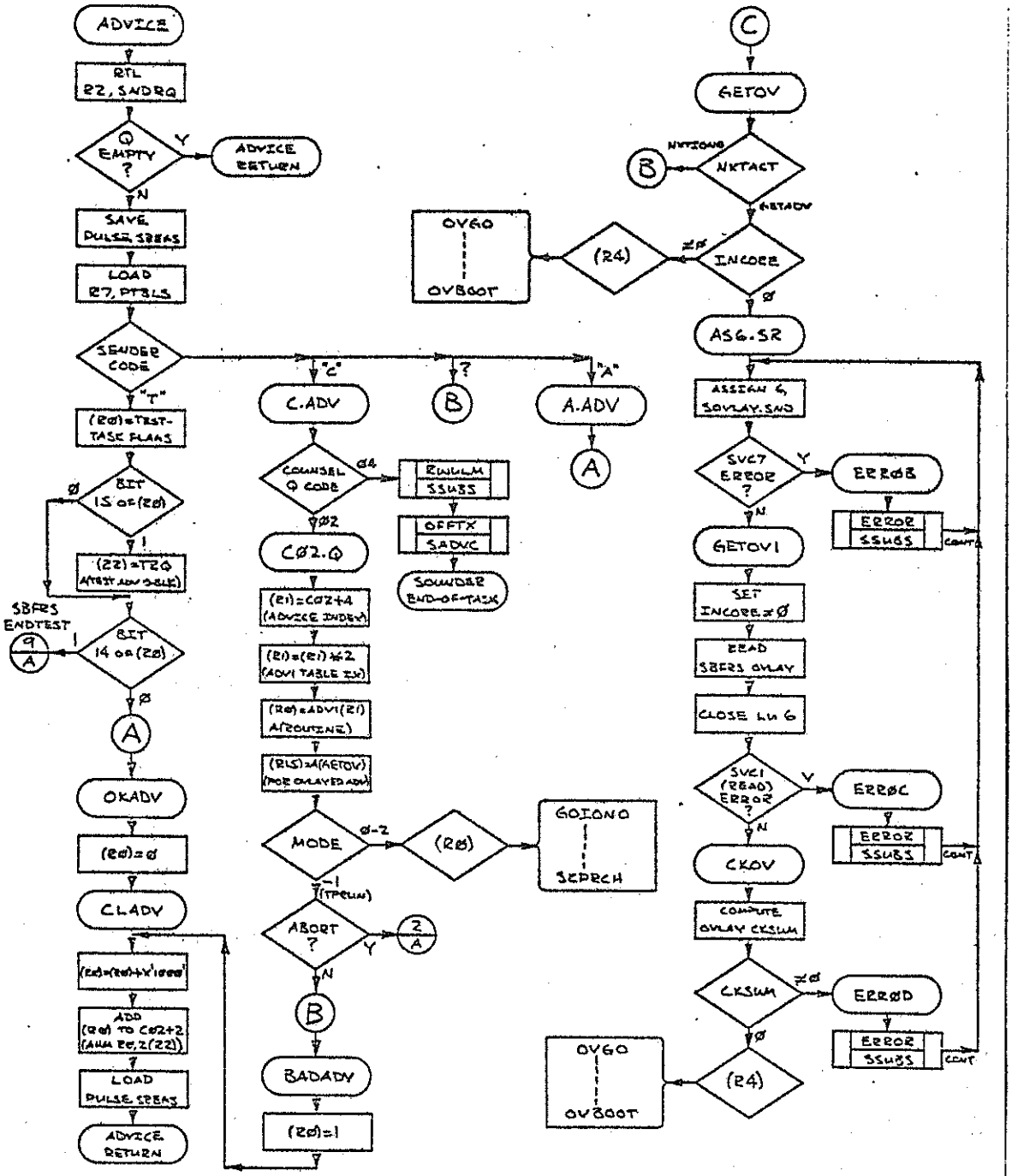


Figure 2.6. SADVC Sheet 1

2.16. Flowcharts

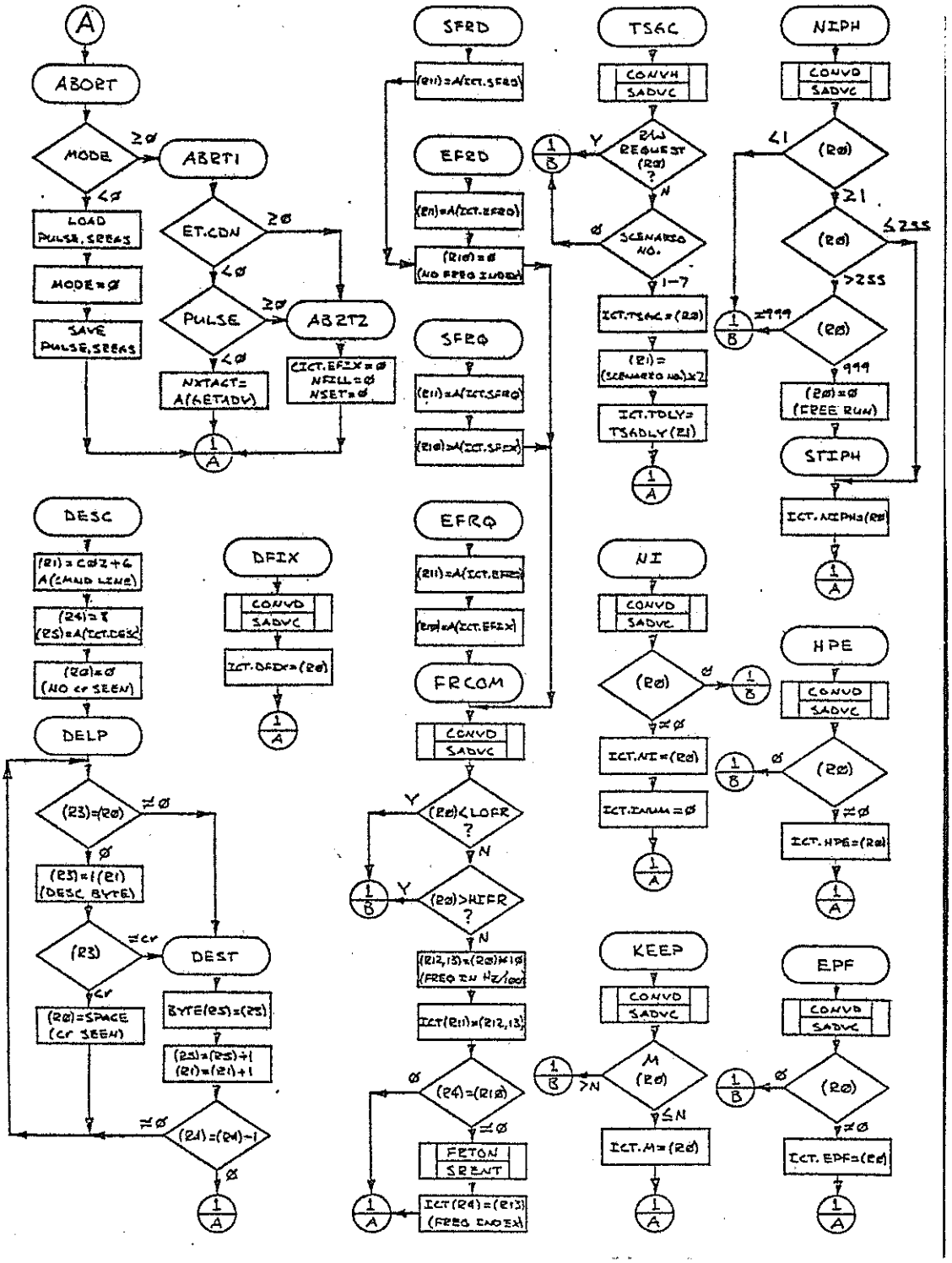


Figure 2.7. SADVC Sheet 2

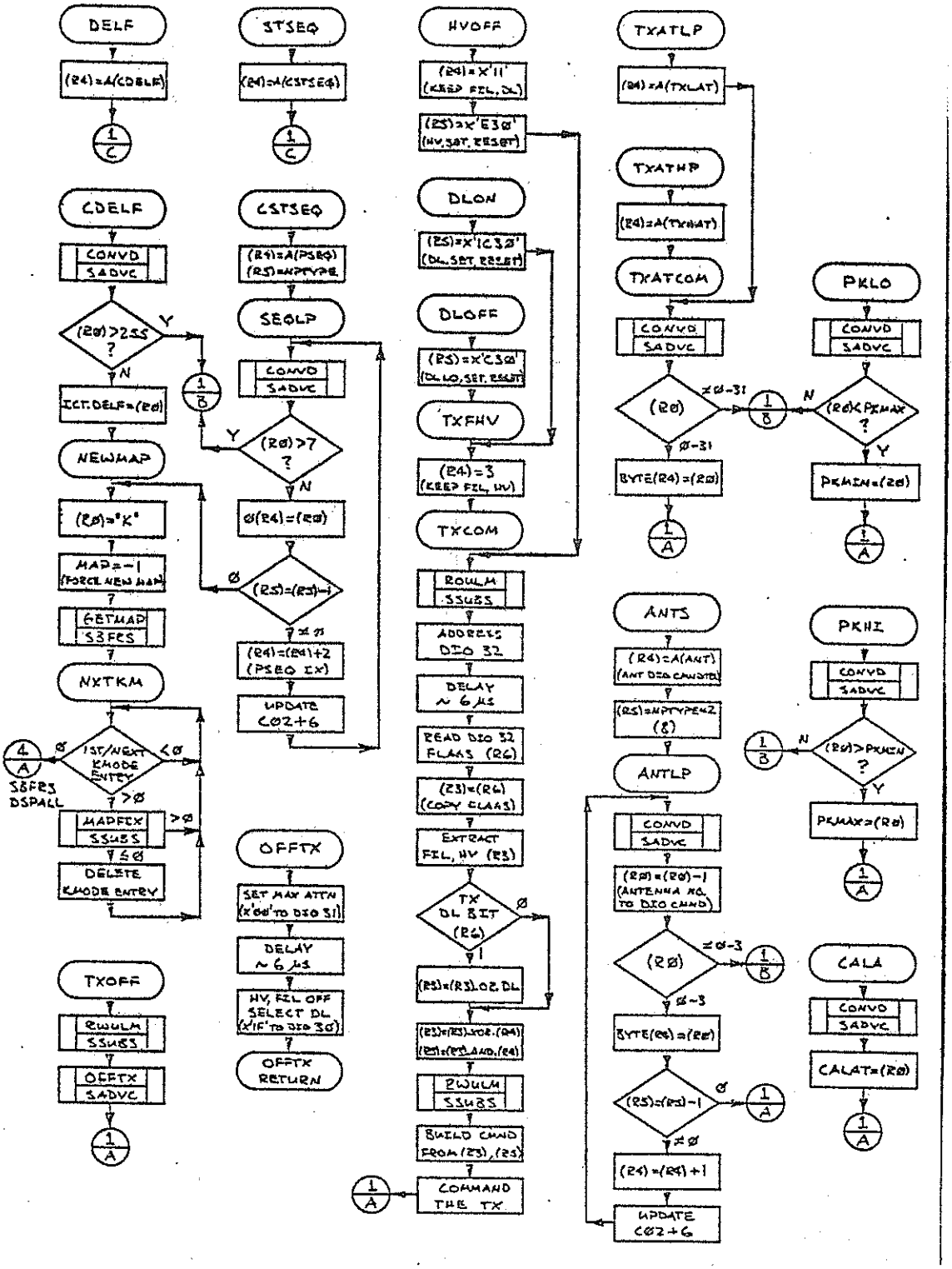


Figure 2.8. SADVC Sheet 3

2.16. Flowcharts

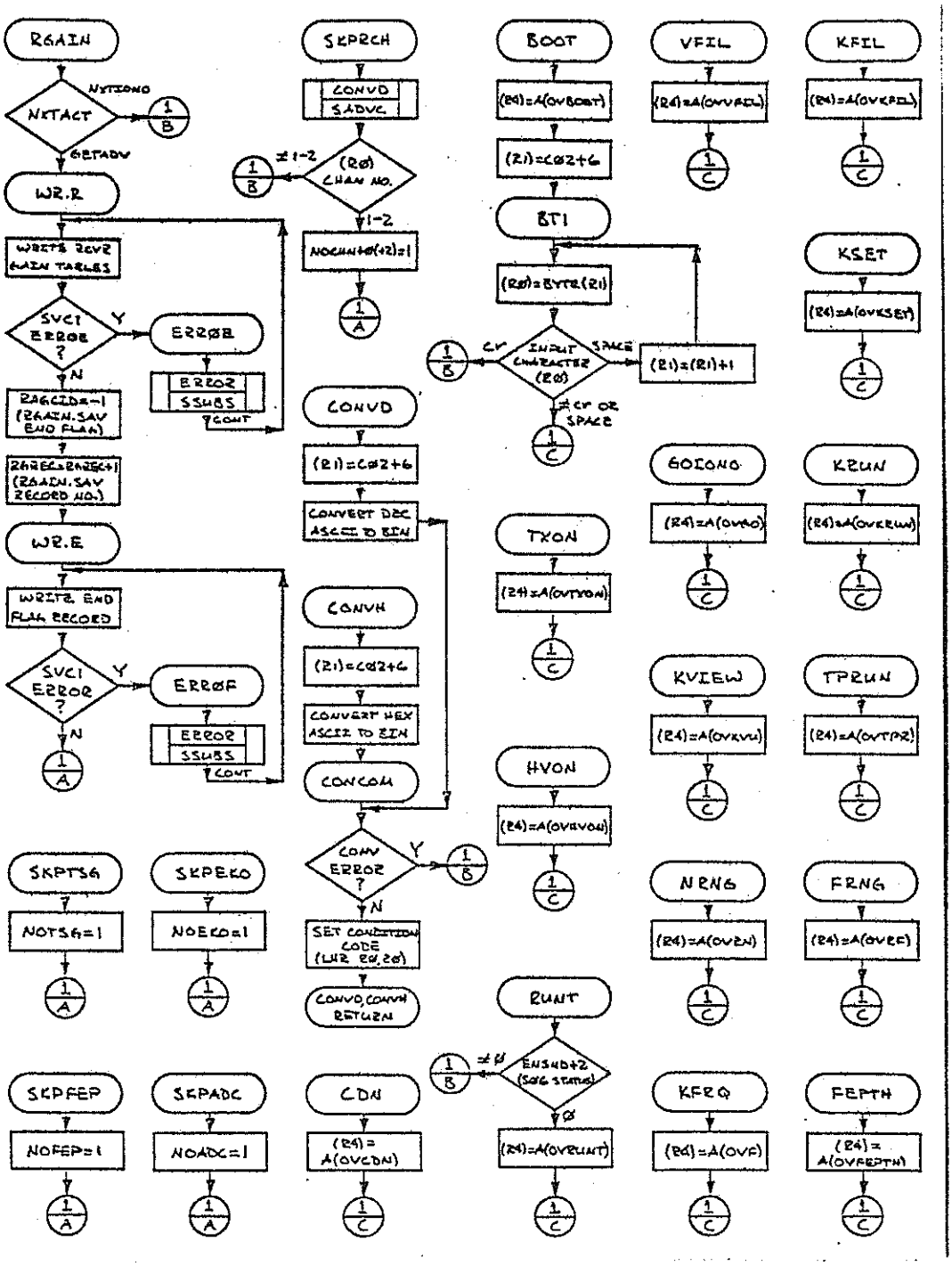


Figure 2.9. SADVC Sheet 4

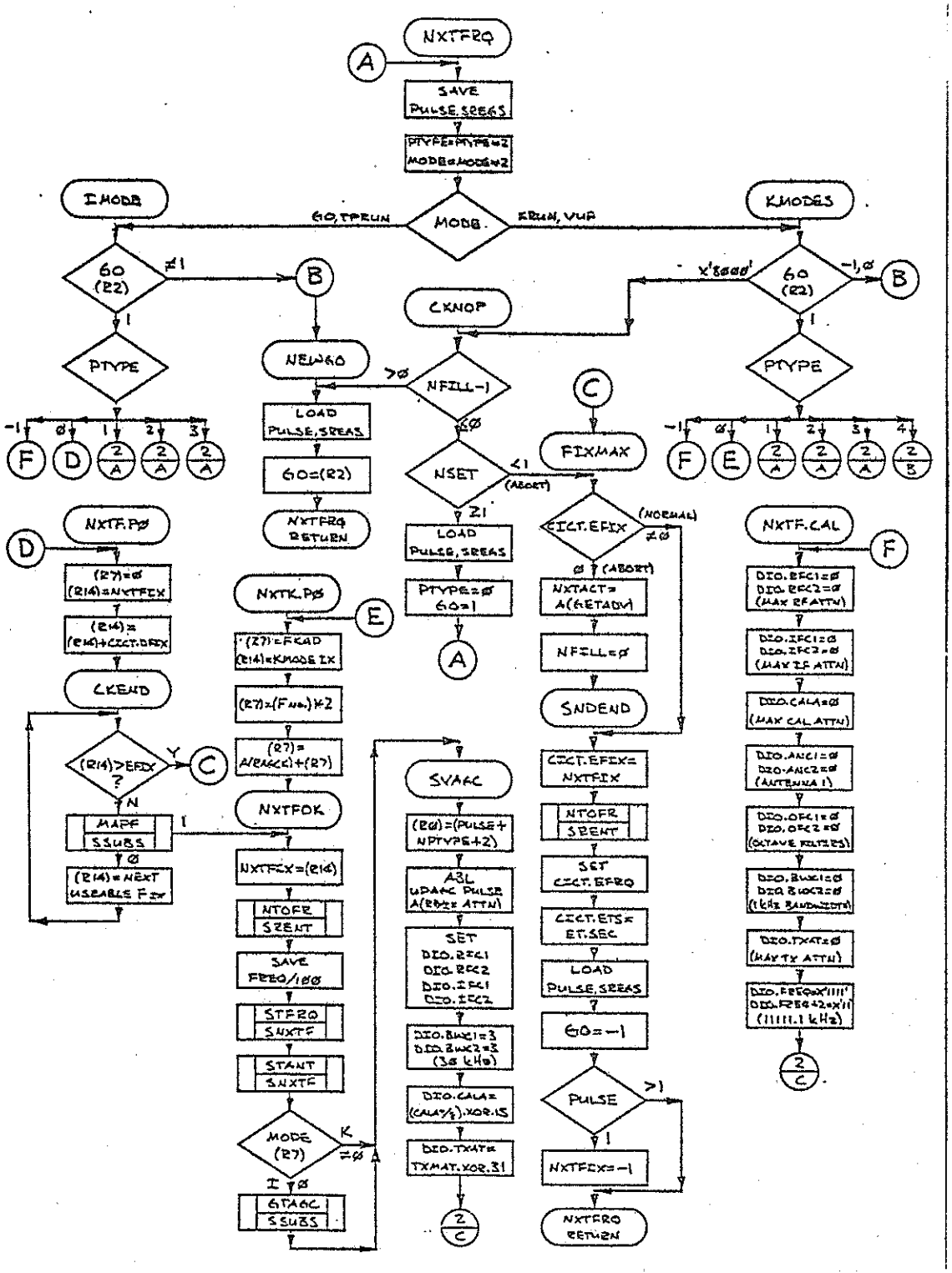


Figure 2.10. SNXTF Sheet 1

2.16. Flowcharts

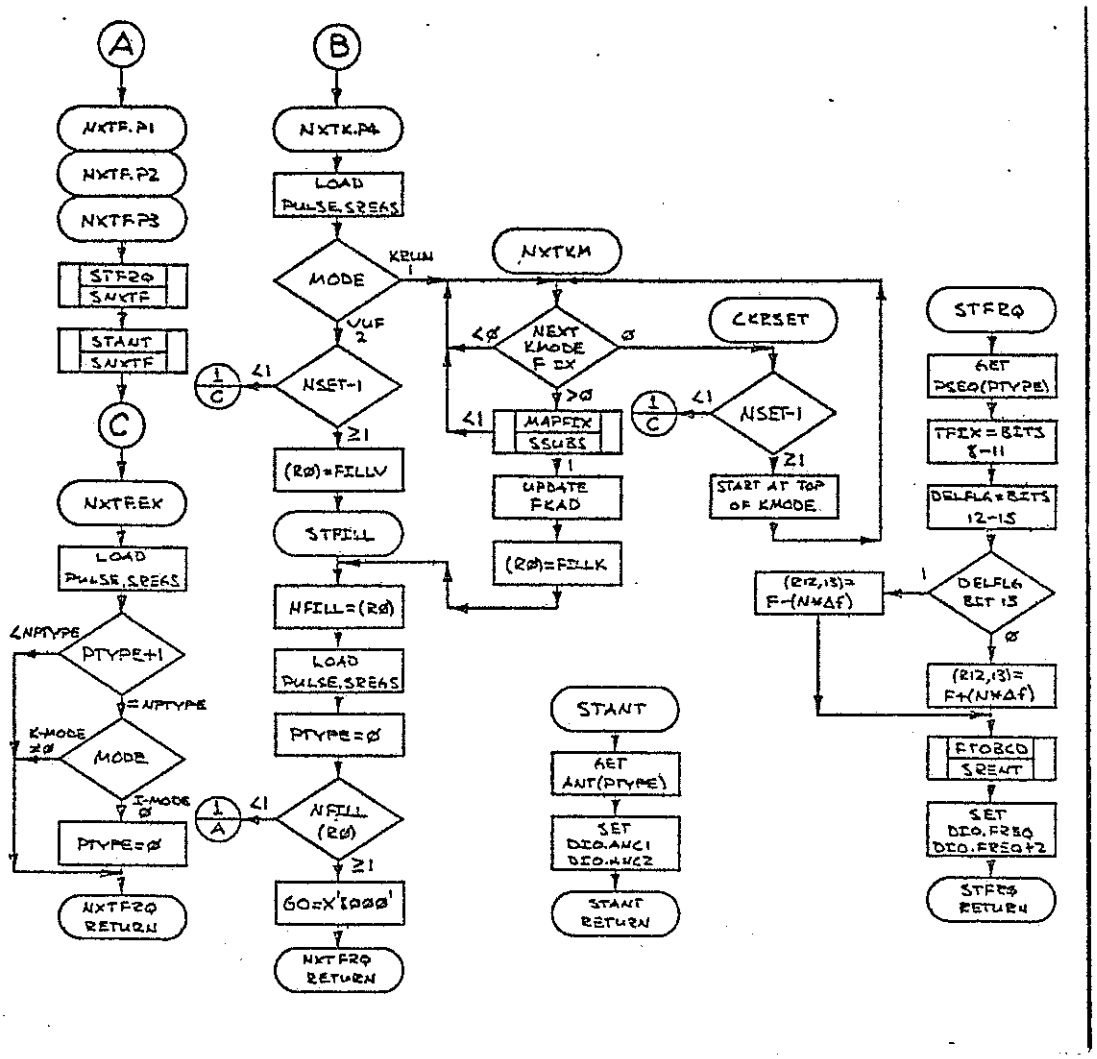


Figure 2.11. SNXTF Sheet 2

2.16. Flowcharts

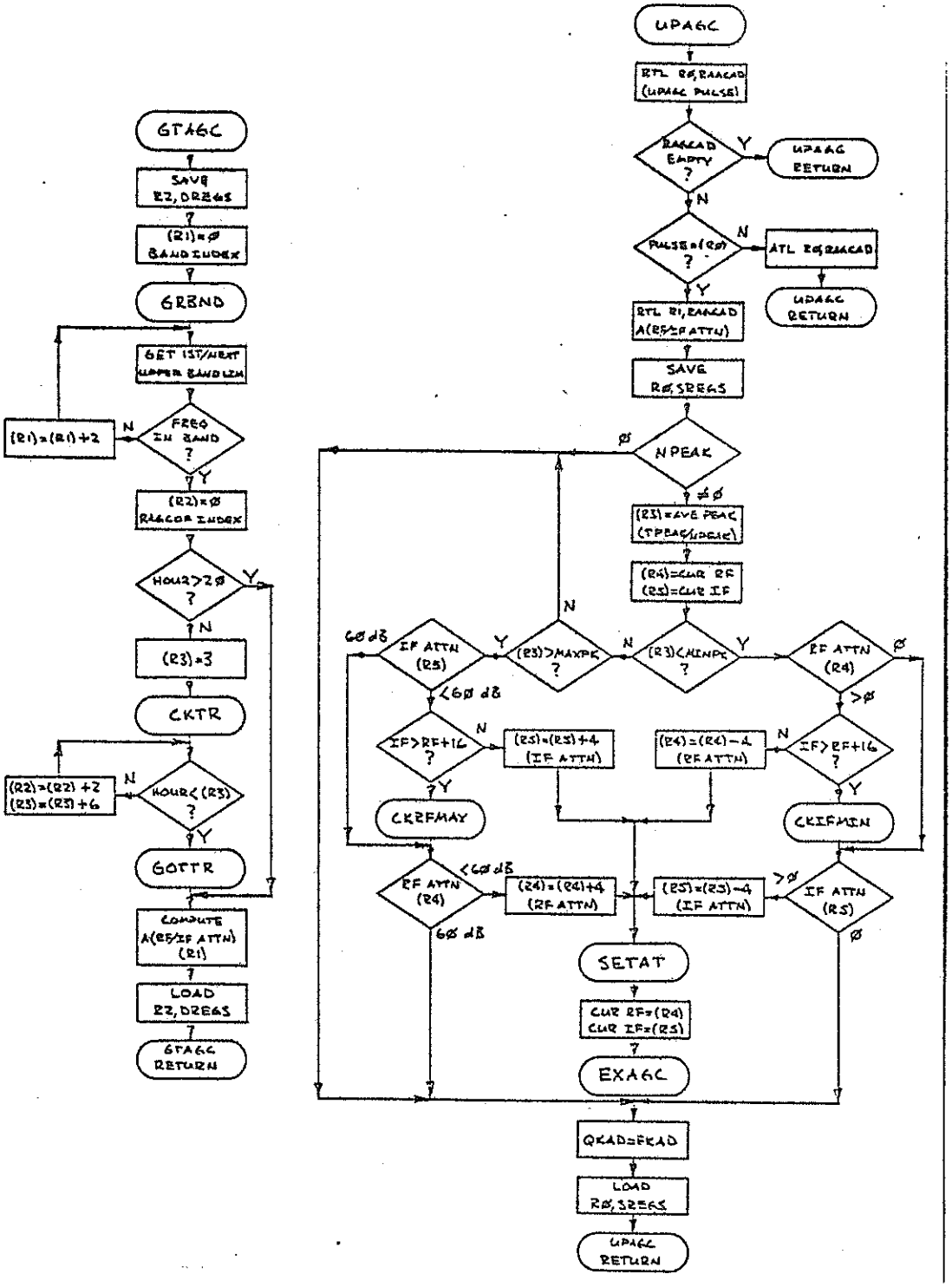


Figure 2.12. SSUBS Sheet 1

2.16. Flowcharts

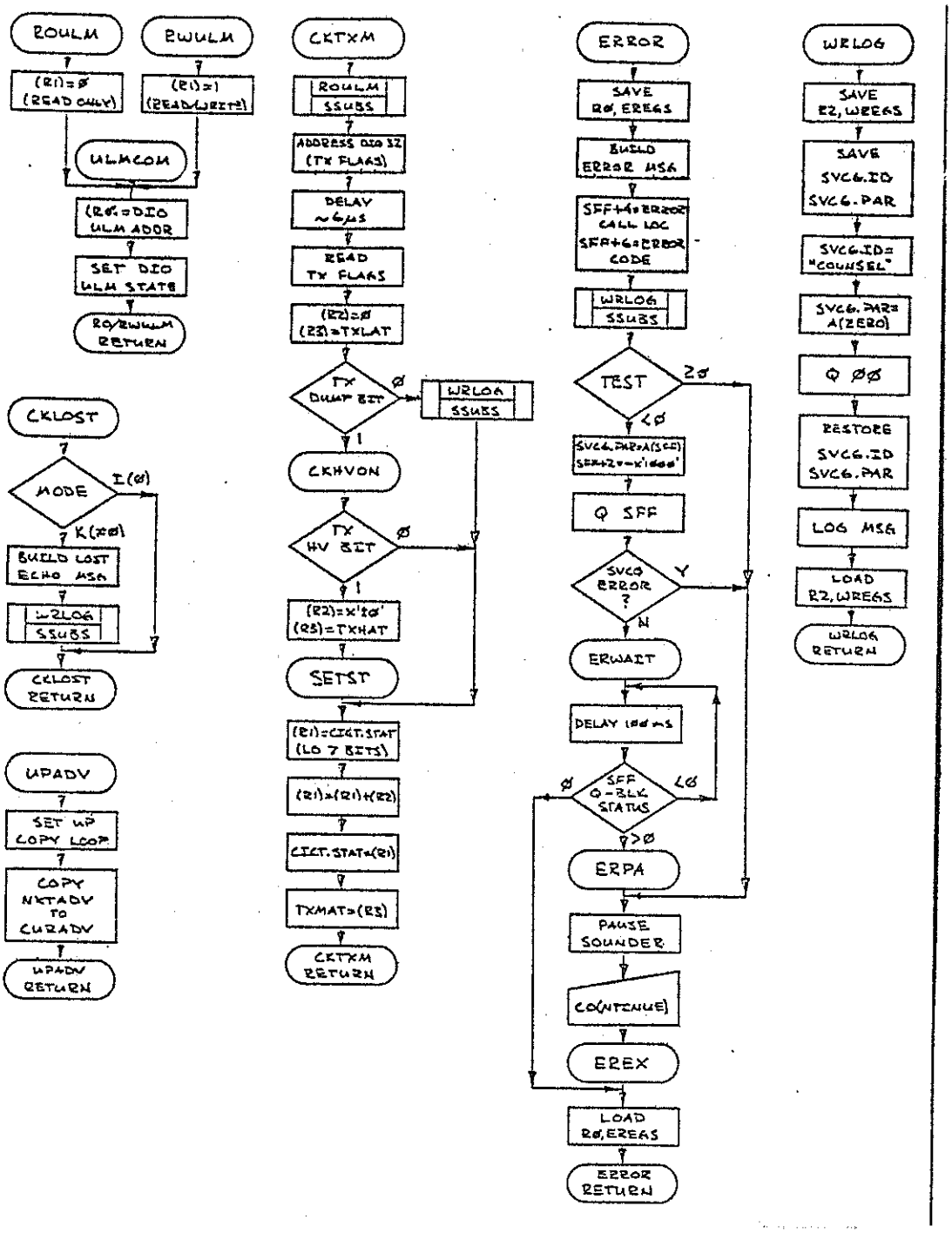


Figure 2.13. SSUBS Sheet 2

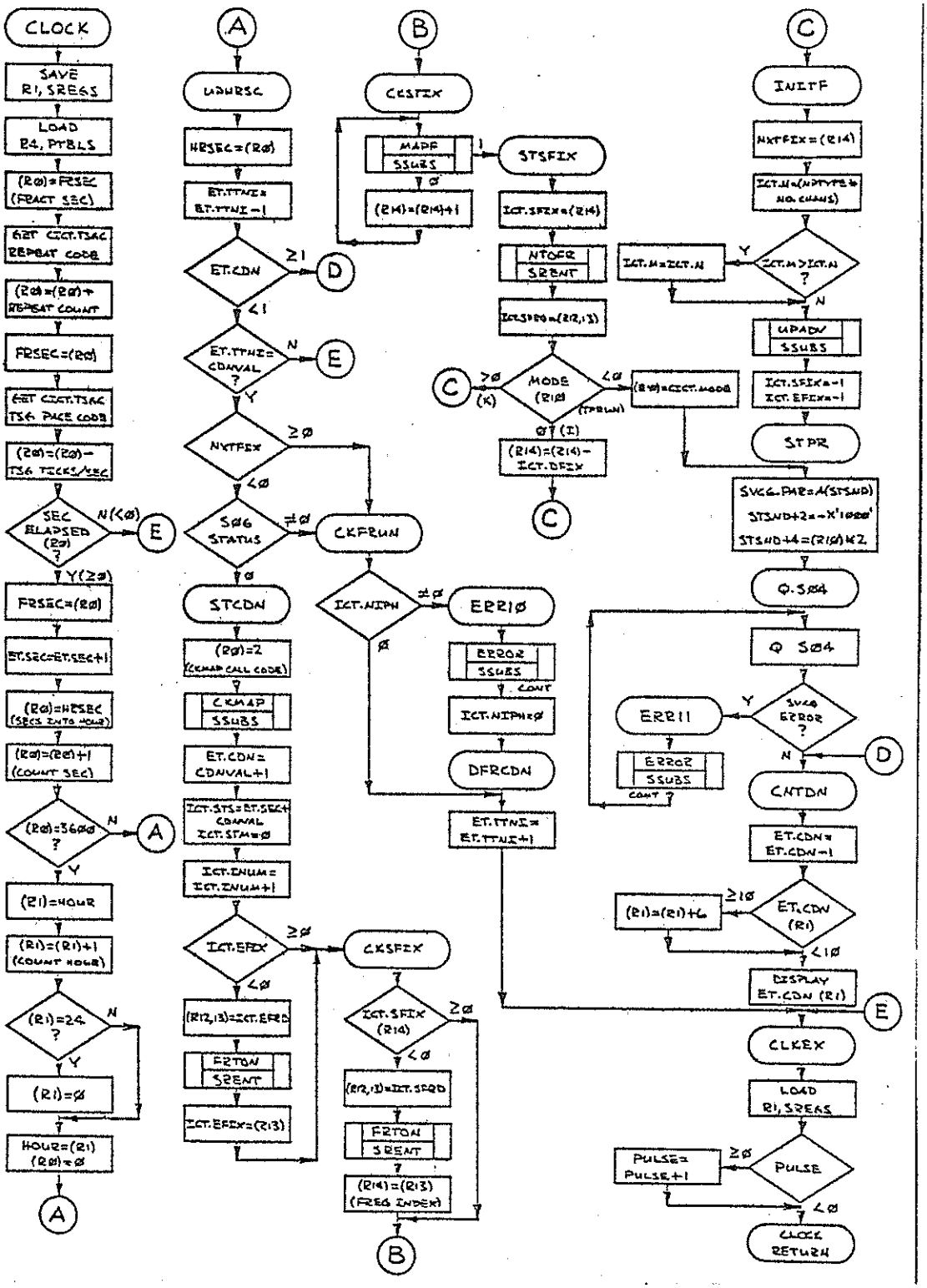


Figure 2.14. SSUBS Sheet 3

2.16. Flowcharts

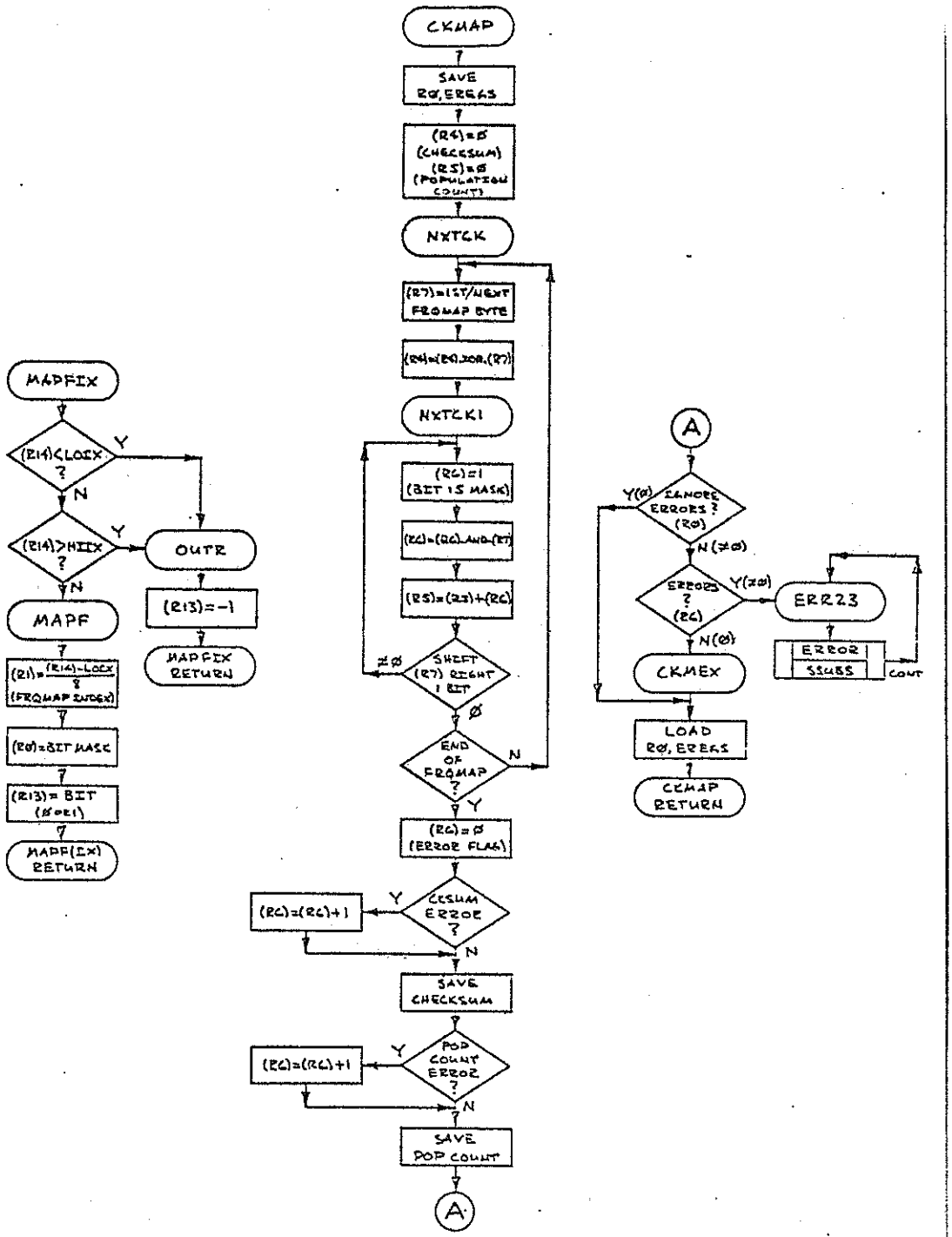


Figure 2.15. SSUBS Sheet 4

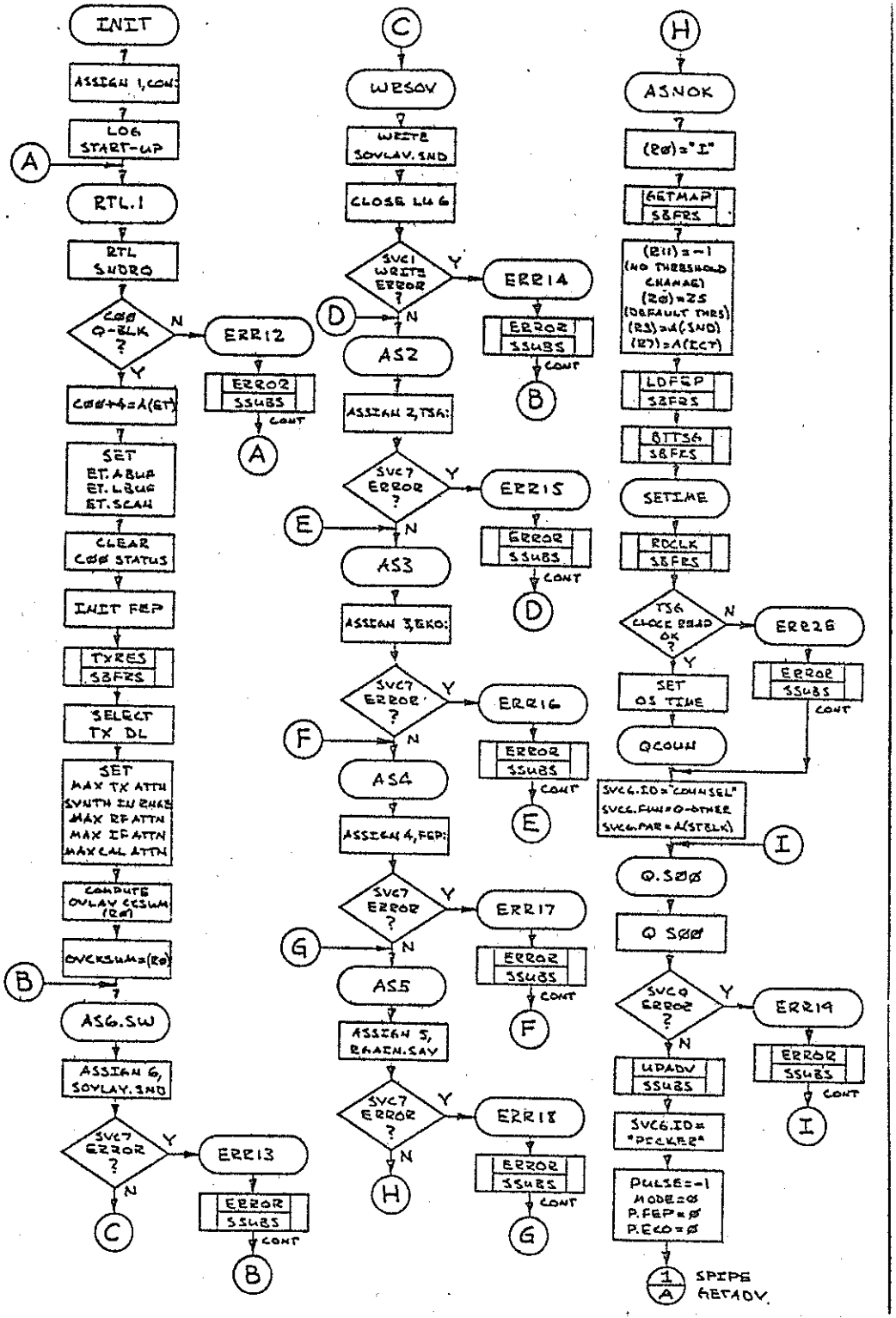


Figure 2.16. SBFRS Sheet 1

2.16. Flowcharts

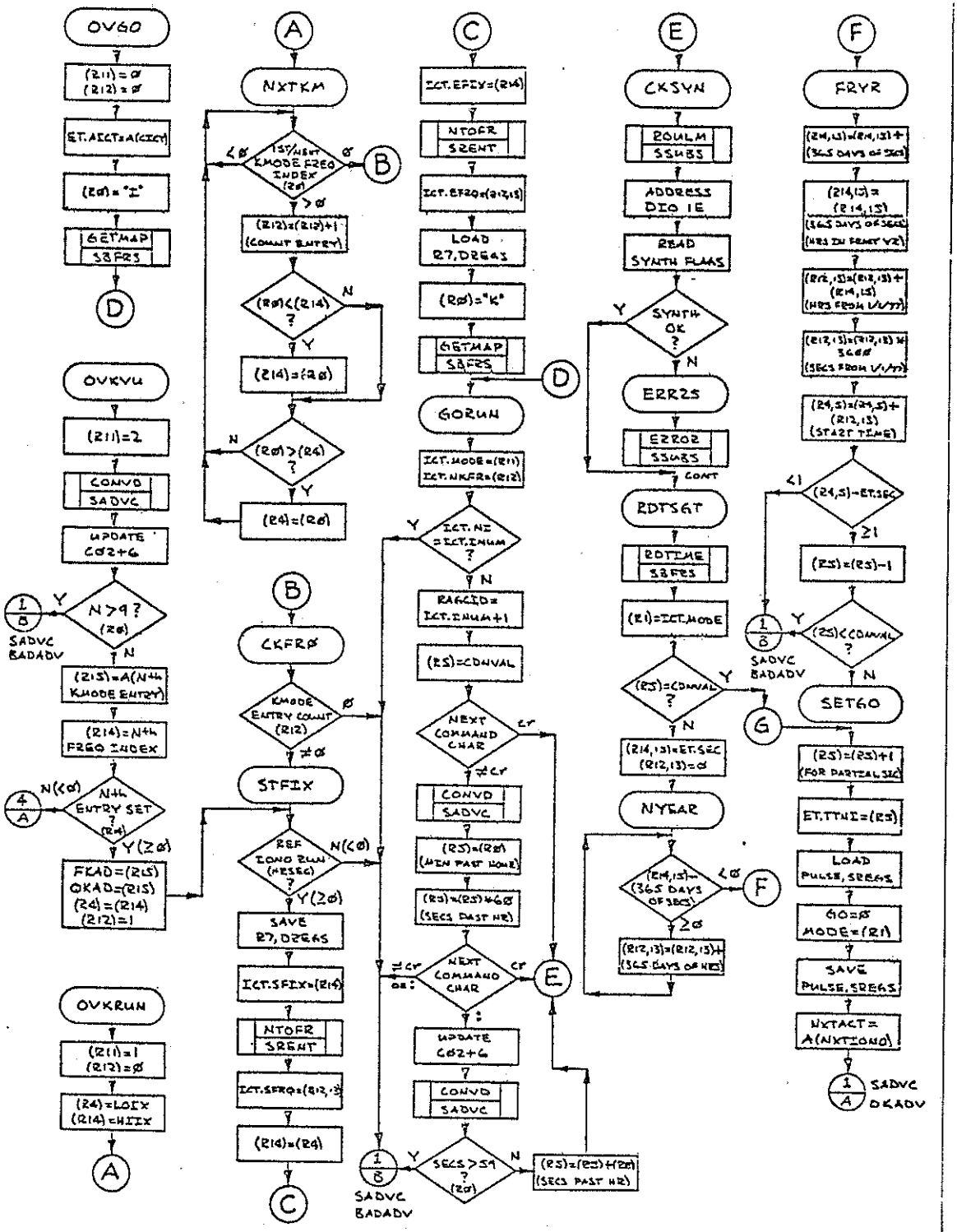


Figure 2.17. SBFRS Sheet 2

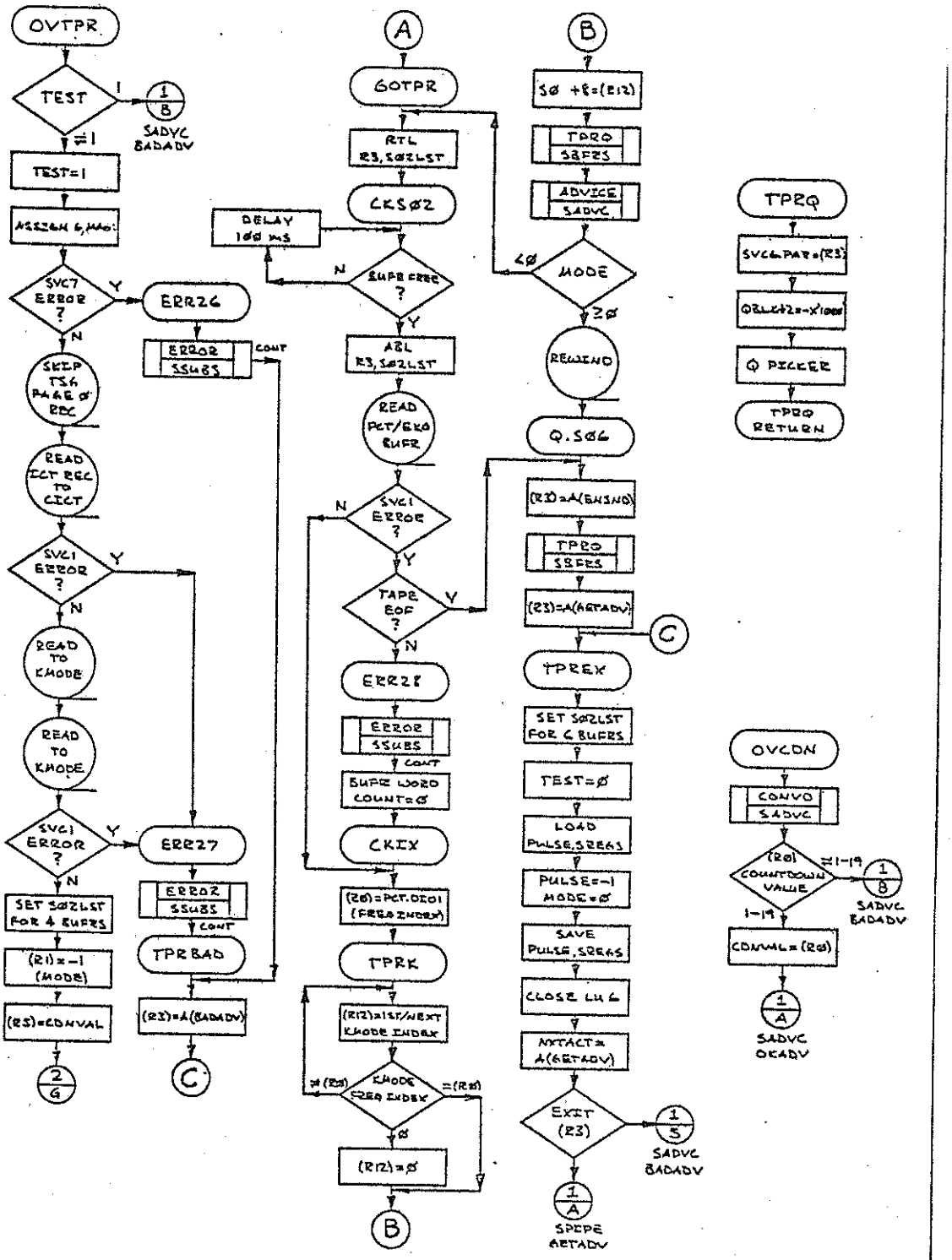


Figure 2.18. SBFRS Sheet 3

2.16. Flowcharts

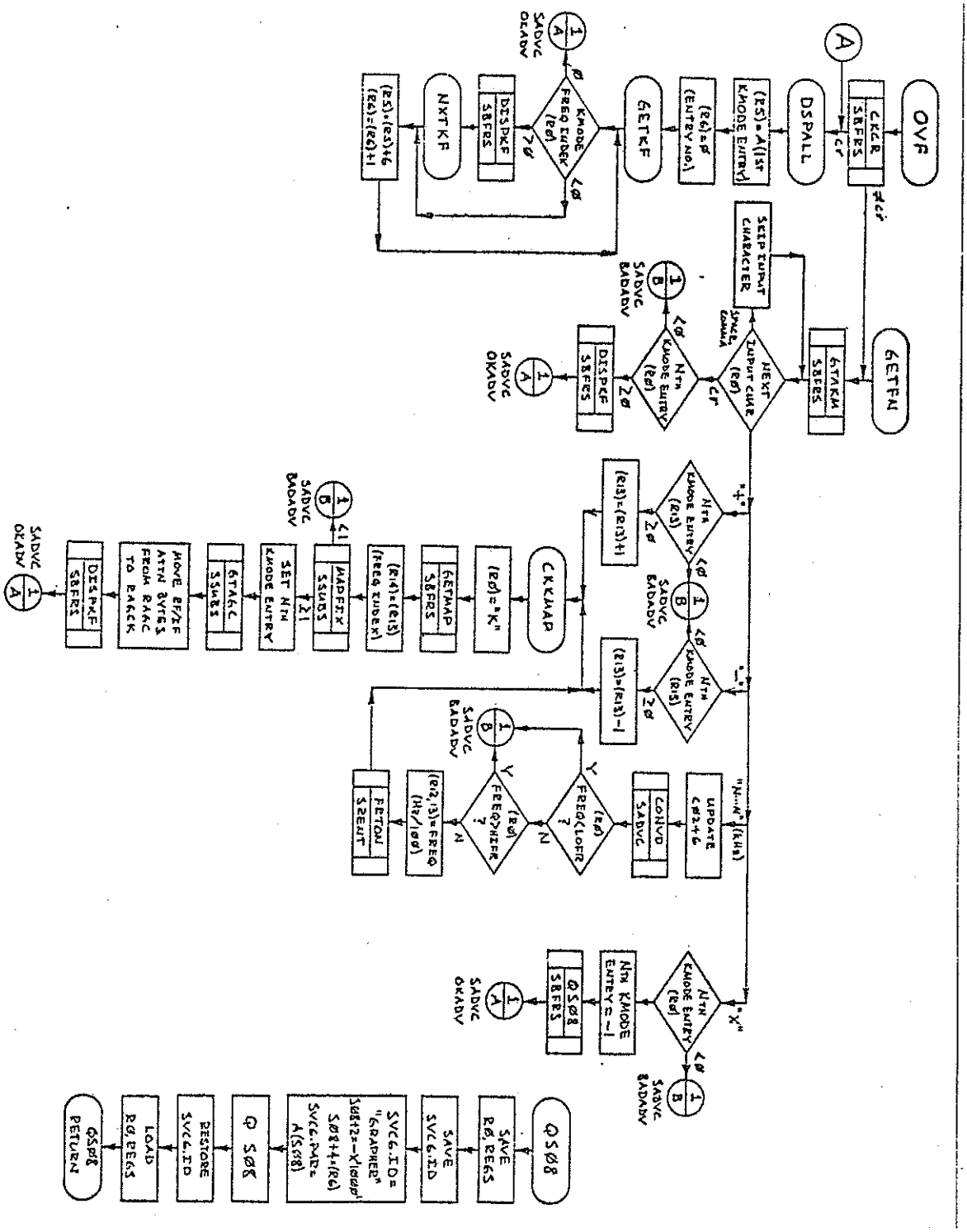


Figure 2.19. SEFRS Sheet 4

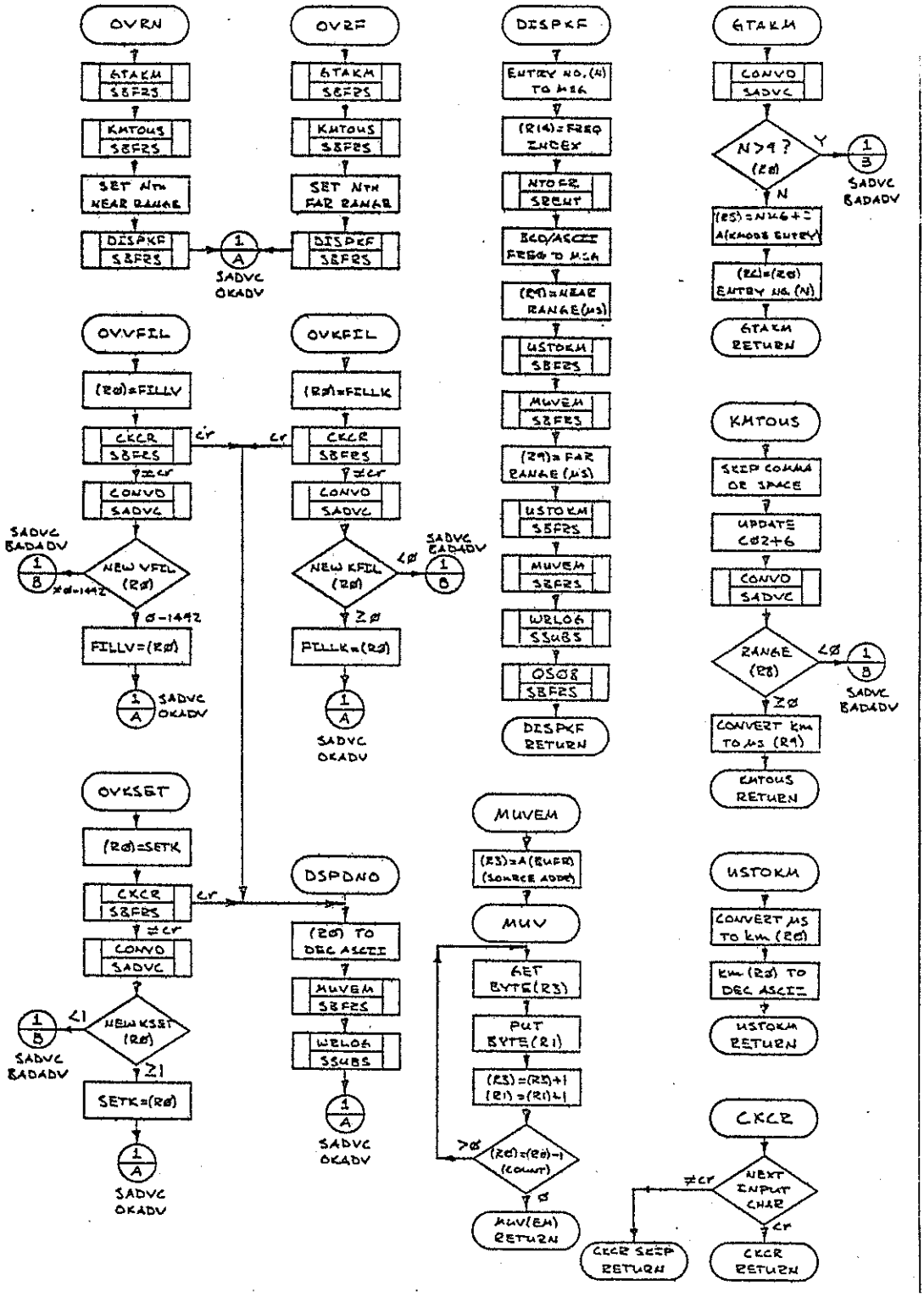


Figure 2.20. SEFRS Sheet 5

2.16. Flowcharts

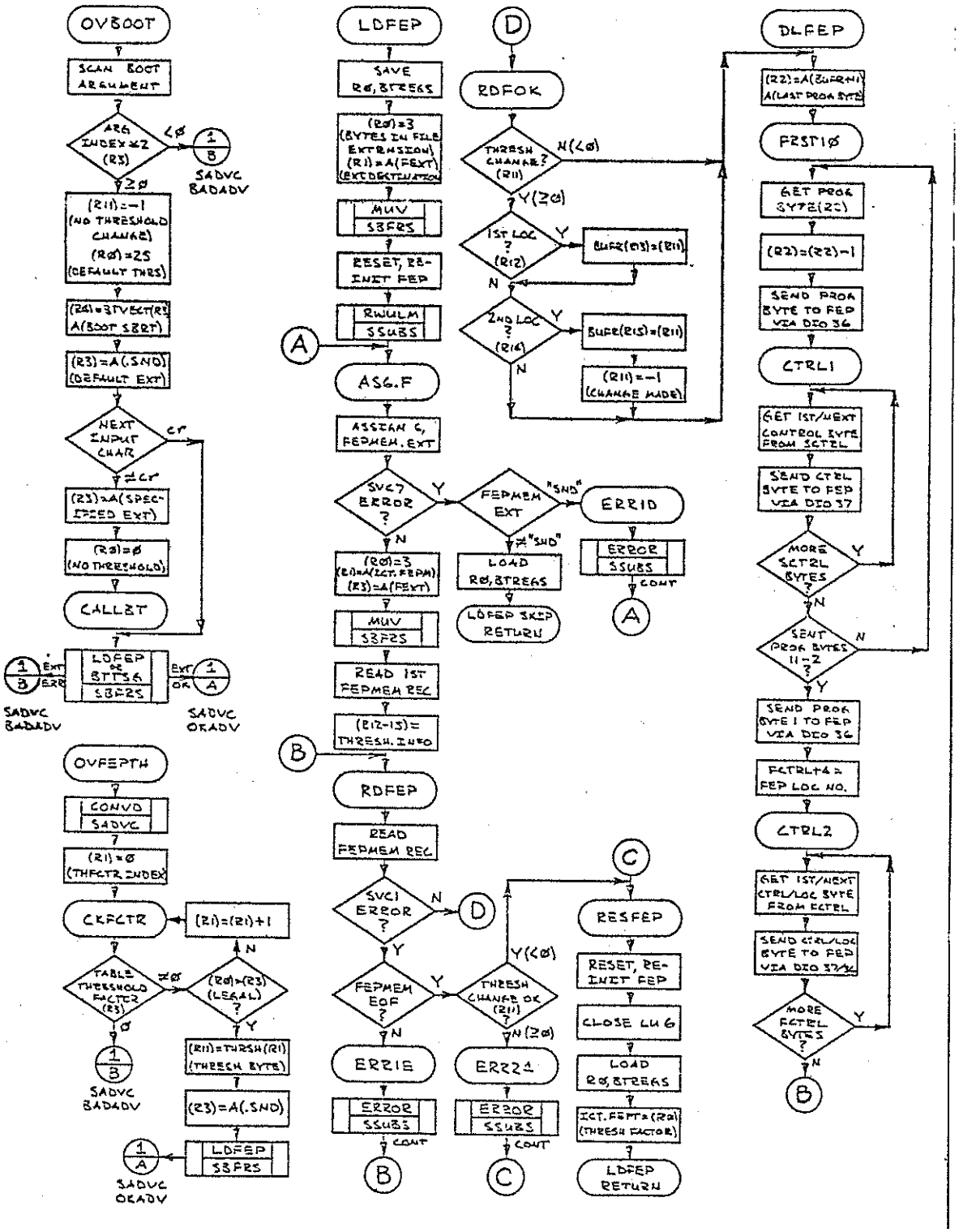


Figure 2.21. SBFRS Sheet 6

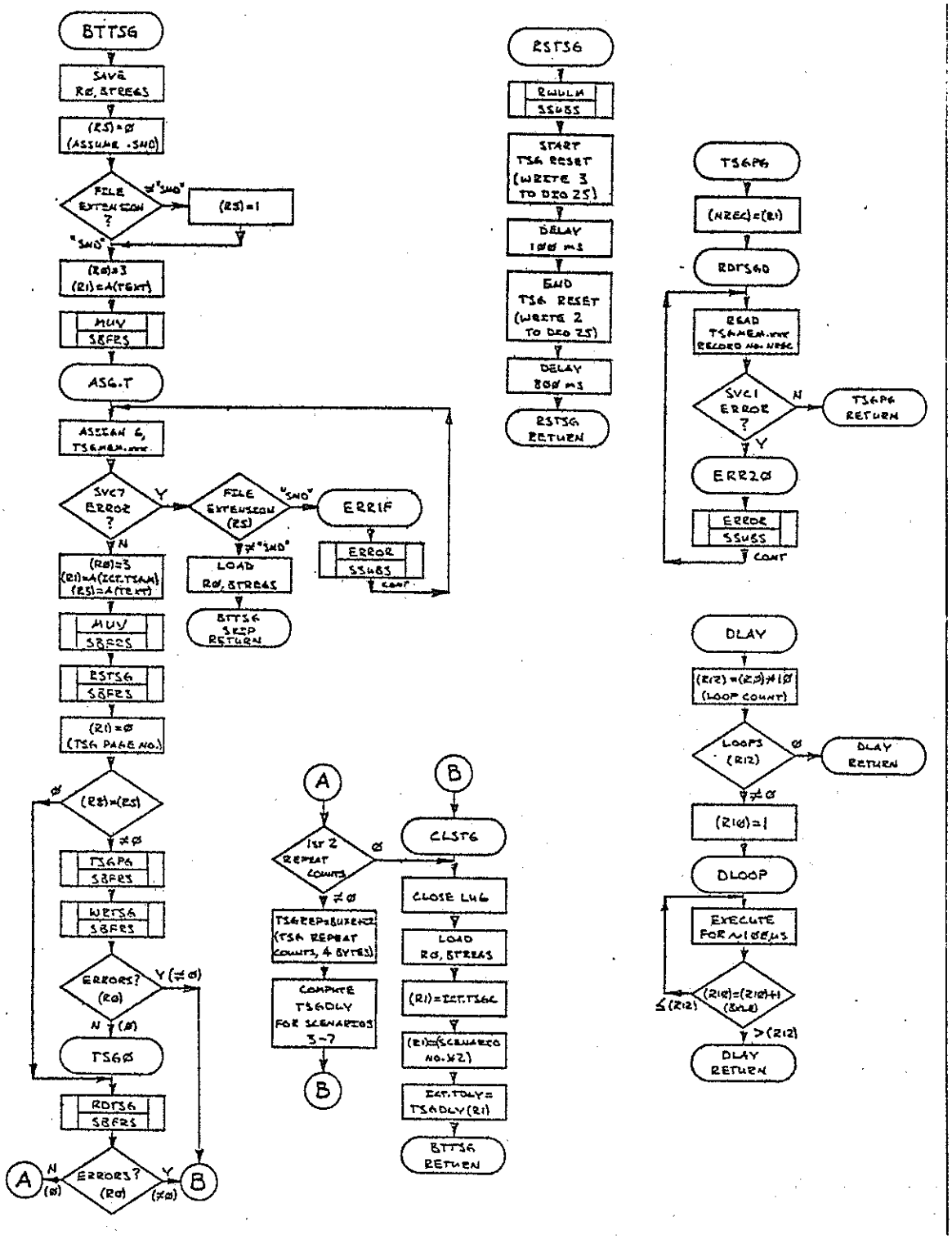


Figure 2.22. SBFRS Sheet 7-A

2.16. Flowcharts

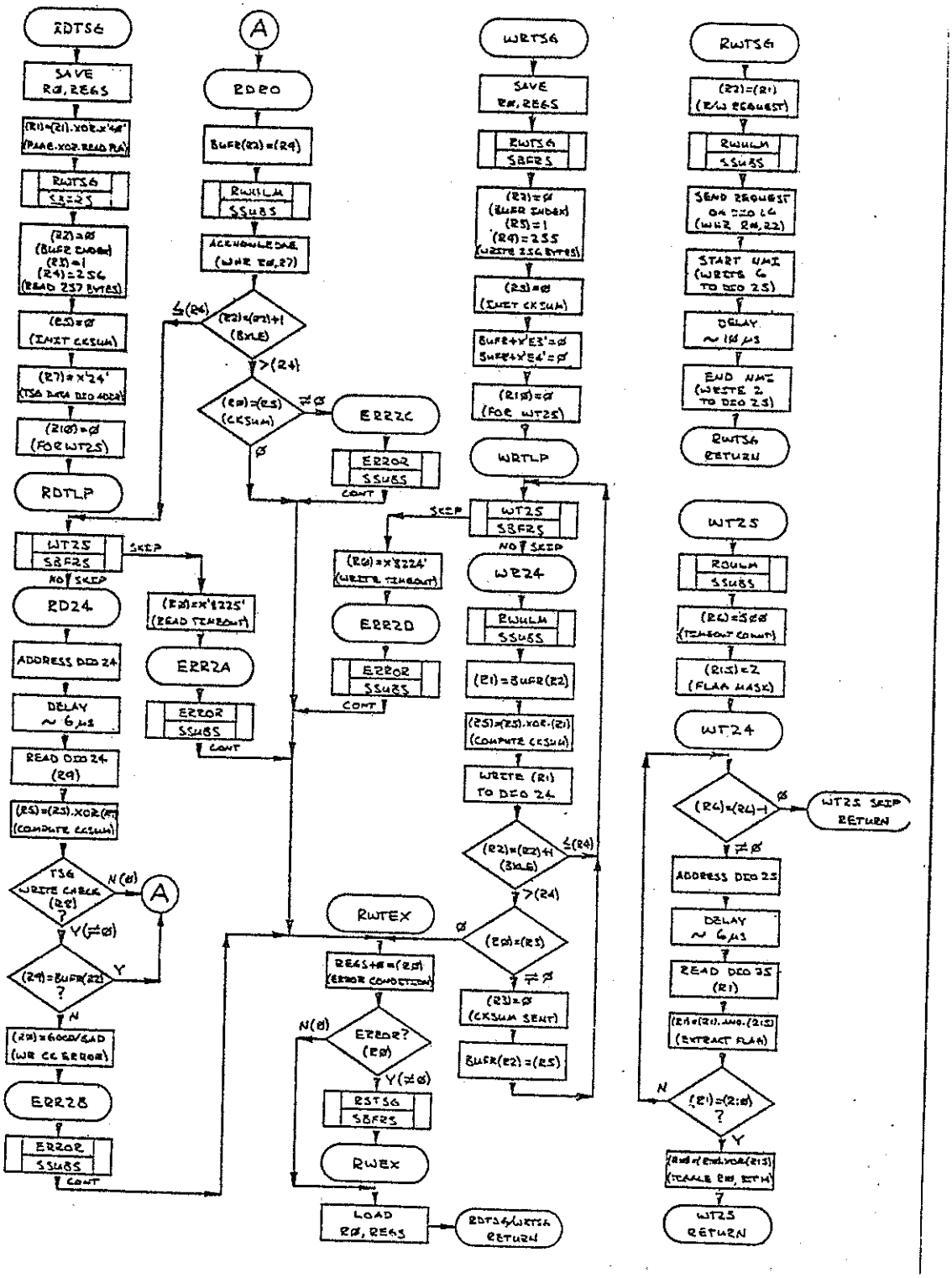


Figure 2.22. SBFRS Sheet 7-B

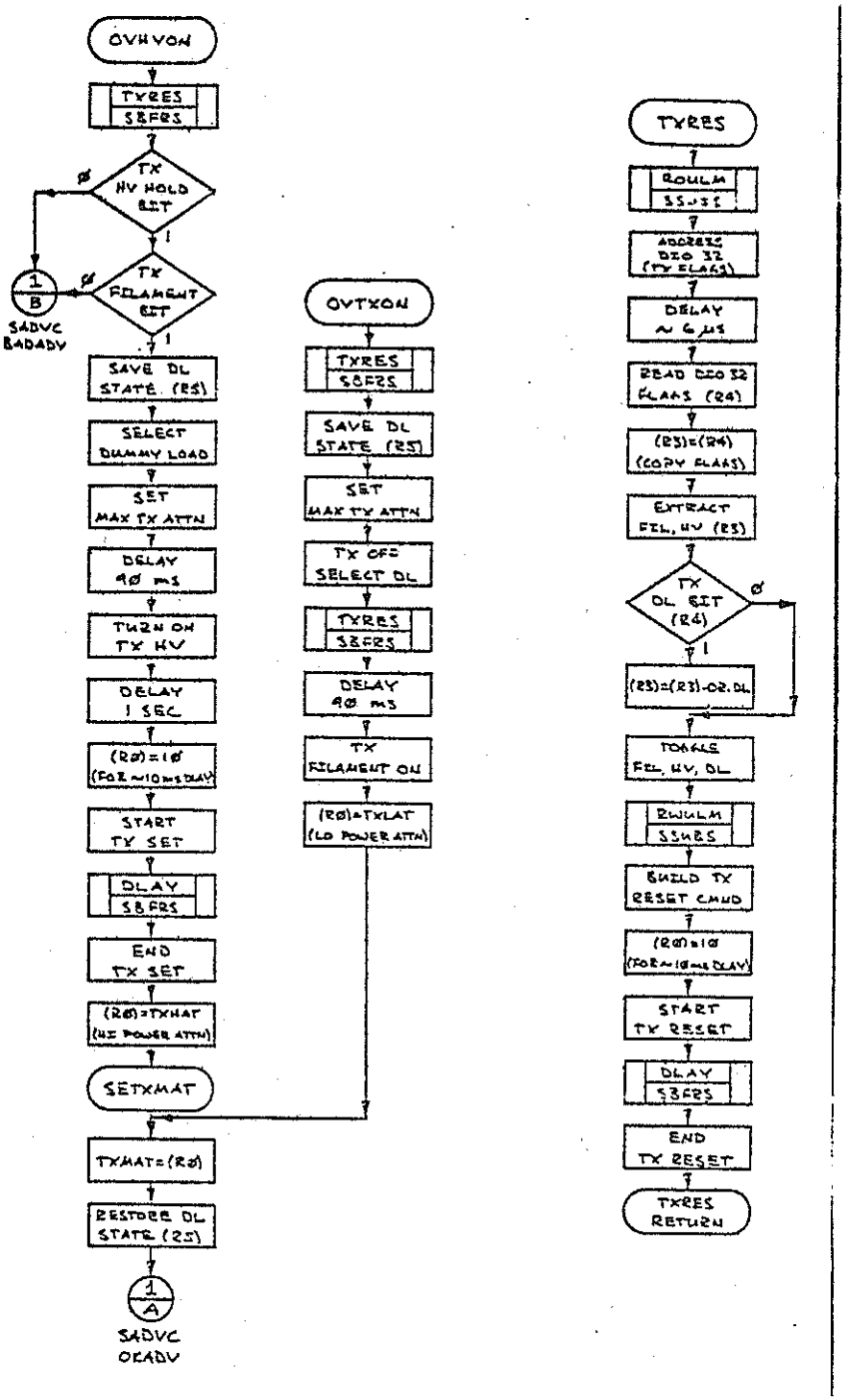


Figure 2.23. SEFRS Sheet 8

2.16. Flowcharts

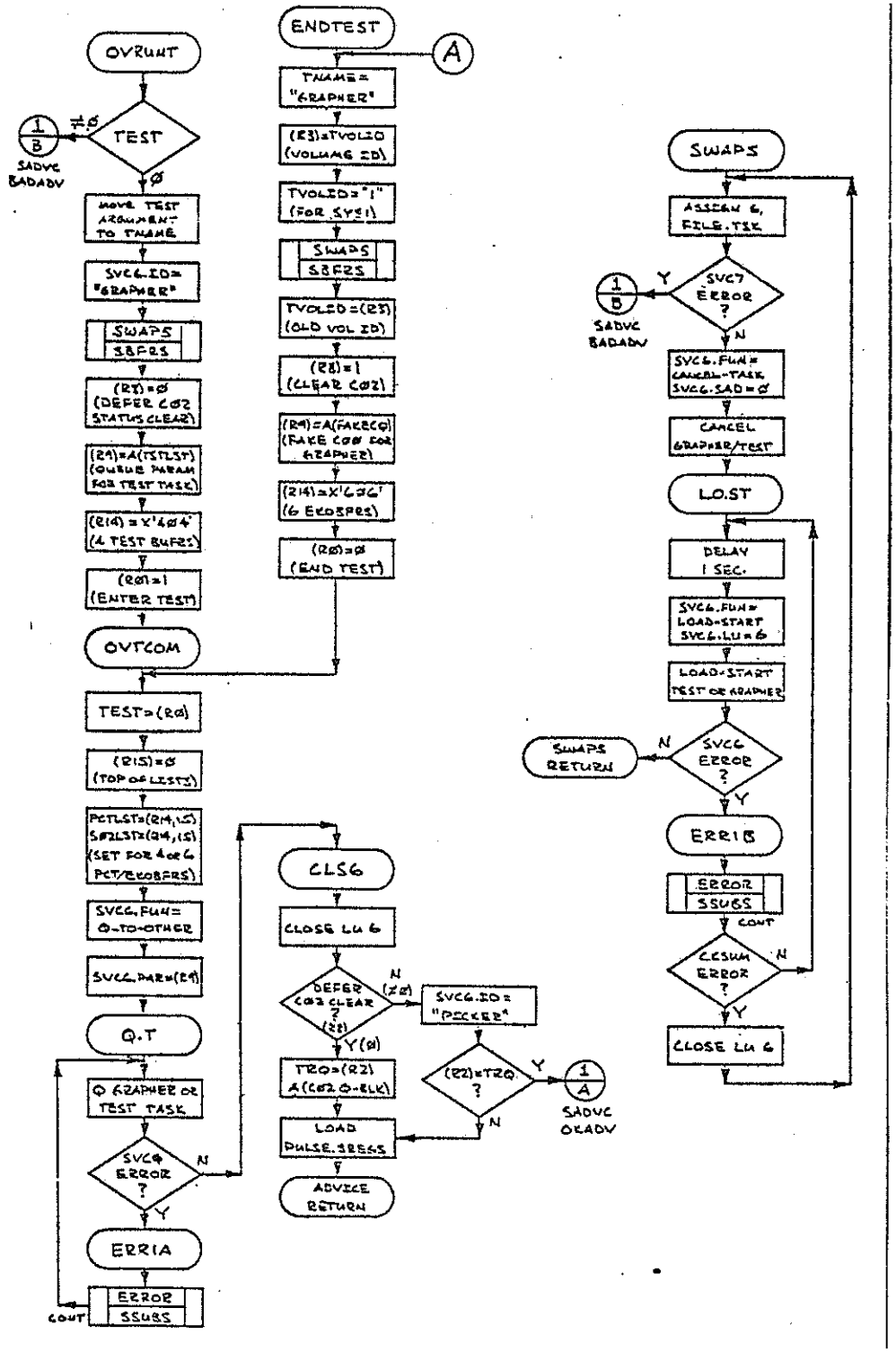


Figure 2.24. SBFRS Sheet 9

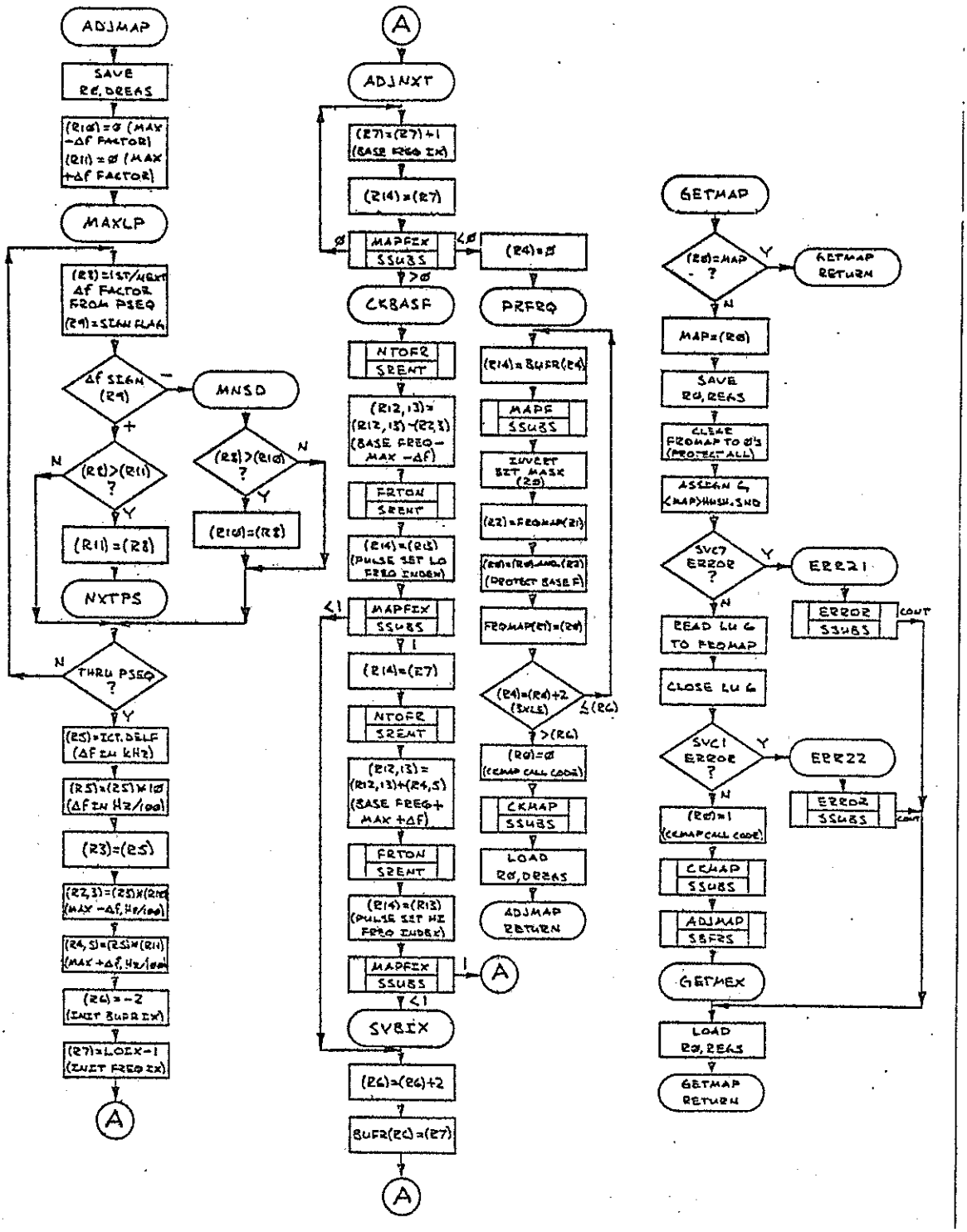


Figure 2.25. SBFRS Sheet 10

2.16. Flowcharts

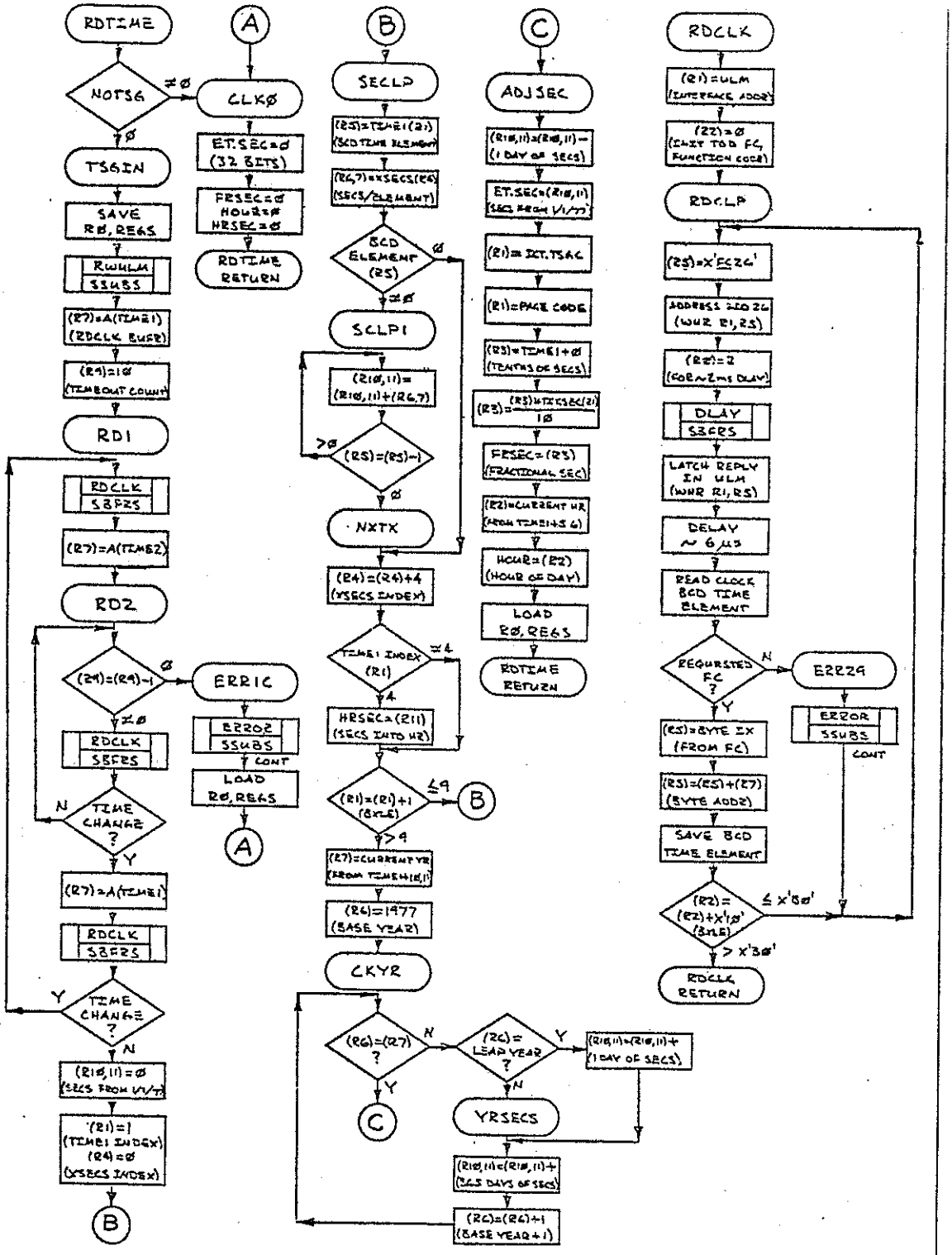


Figure 2.26. SBFRS Sheet 11

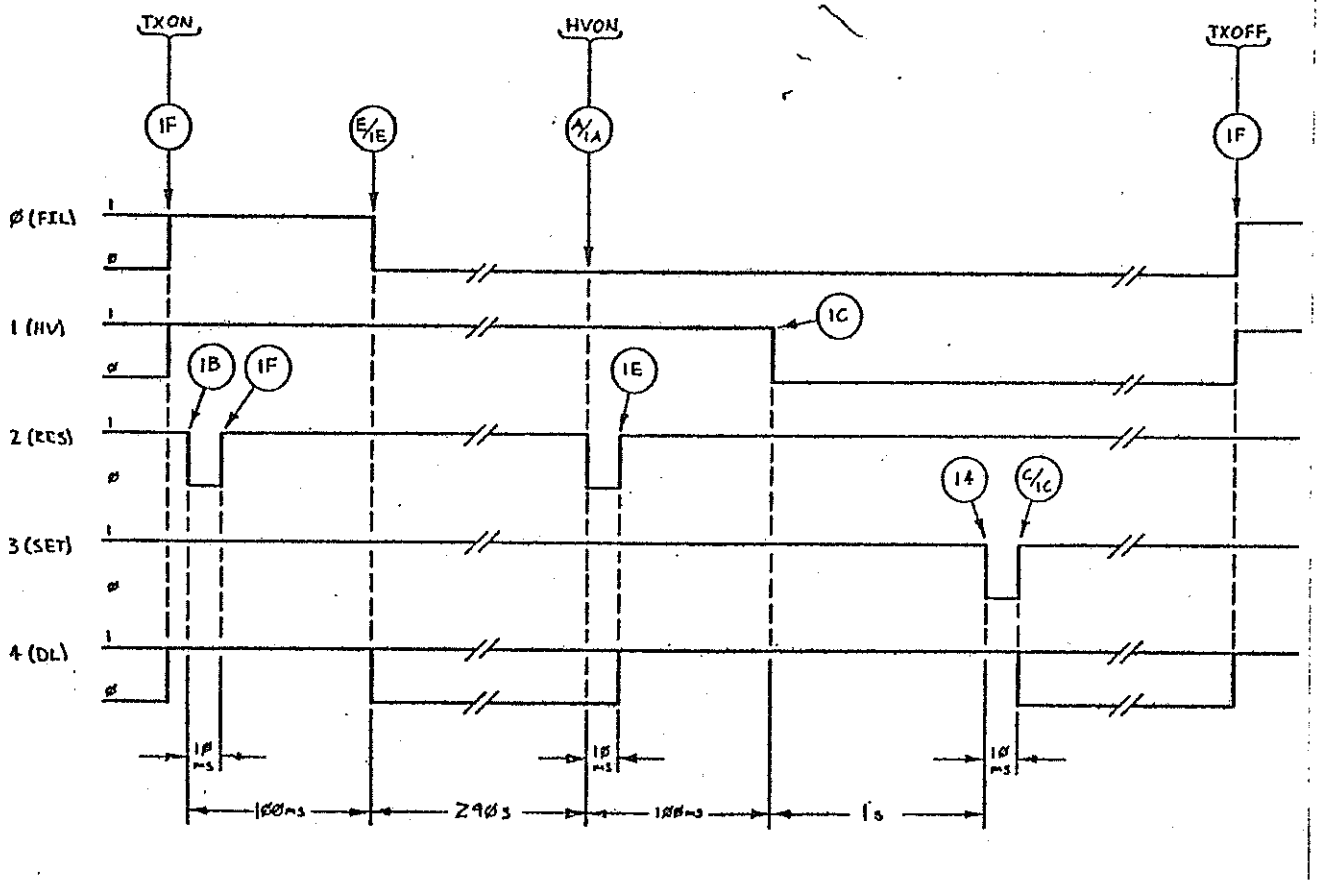


Figure 2.27. TRANSMITTER TIMING

2.16. Flowcharts

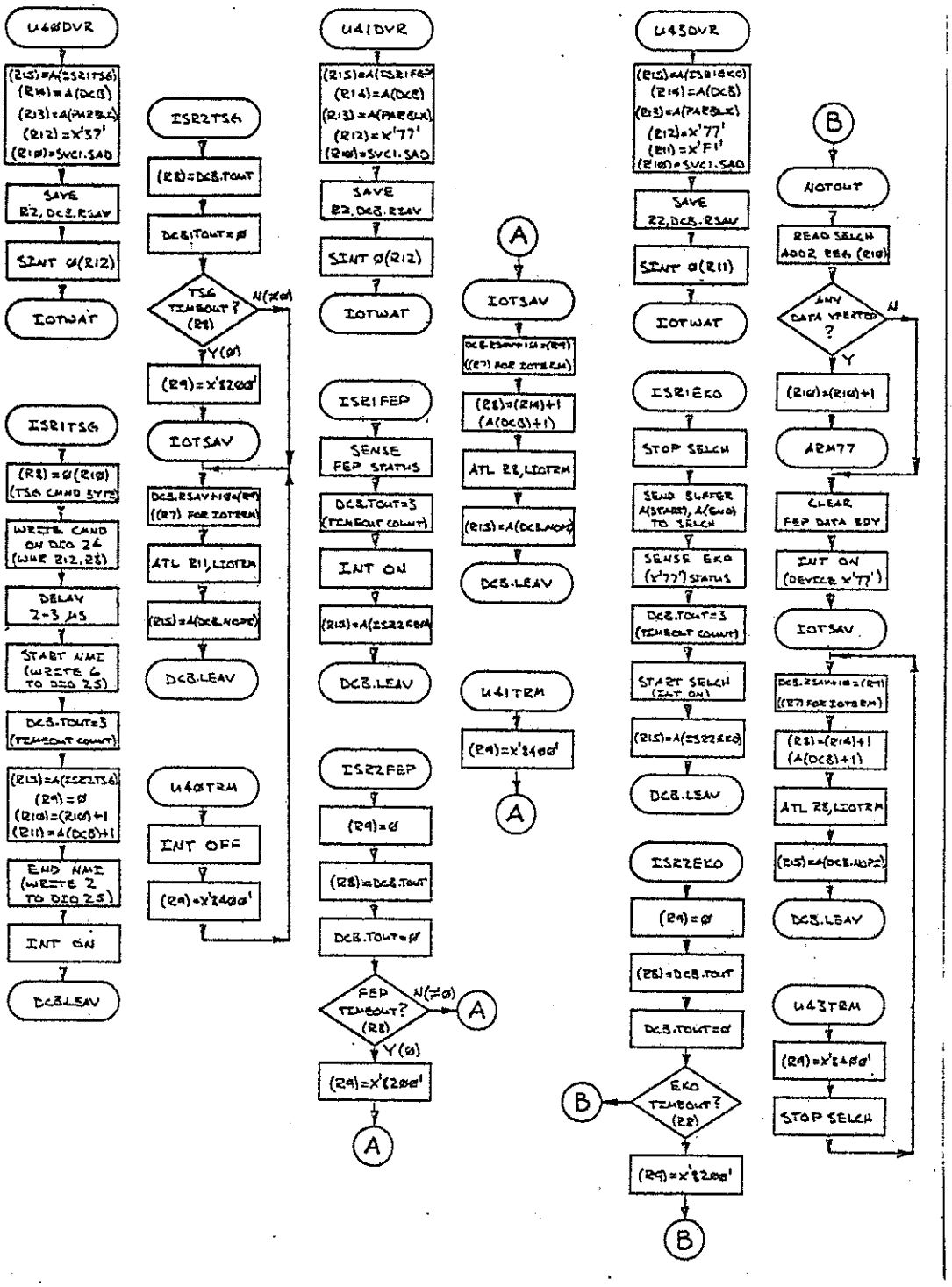


Figure 2.28. SOUNDER DRIVERS

3. PICKER

James R. Winkelman

3.1. Introduction

In Product One, Task PICKER selects range-coincident echoes from the data supplied by SOUNDER and sends the selected data on to MANAGER. It acknowledges the receipt of ionogram start and stop Q-blocks from SOUNDER and start-up from COUNSEL, which also generates a logging message.

Figure 3.1 is a block diagram of PICKER and its queue and is referred to in the descriptions that follow.

PICKER uses several definitions from SNDR in PCB.CAL. SNDR is a block of definitions for SOUNDER parameters created at assembly time. These definitions are edited into PCB.CAL and entered into PICKER with the COPY SNDR command during assembly (see SNDR, Sect. 7.6).

EKOSIZE = 2Ø48 is the length of the EKO buffer in bytes.

NPTYPE = 4 is the number of pulses in a pulse set.

Q = 4 specifies the use of SVC4 (rather than SVC6)

P.MAXEKO = 8 means that a PØ2 Q-block will be limited to 8 echoes.

P.NECHO = 22 saves space within PICKER for only 22 echoes for coincidence testing.

P.MICRO = 6Ø means that two ranges less than 6Ø microseconds apart are defined as the same range.

PIK.TEST = 1 means that certain values useful in testing PICKER are saved.

3.2. Input

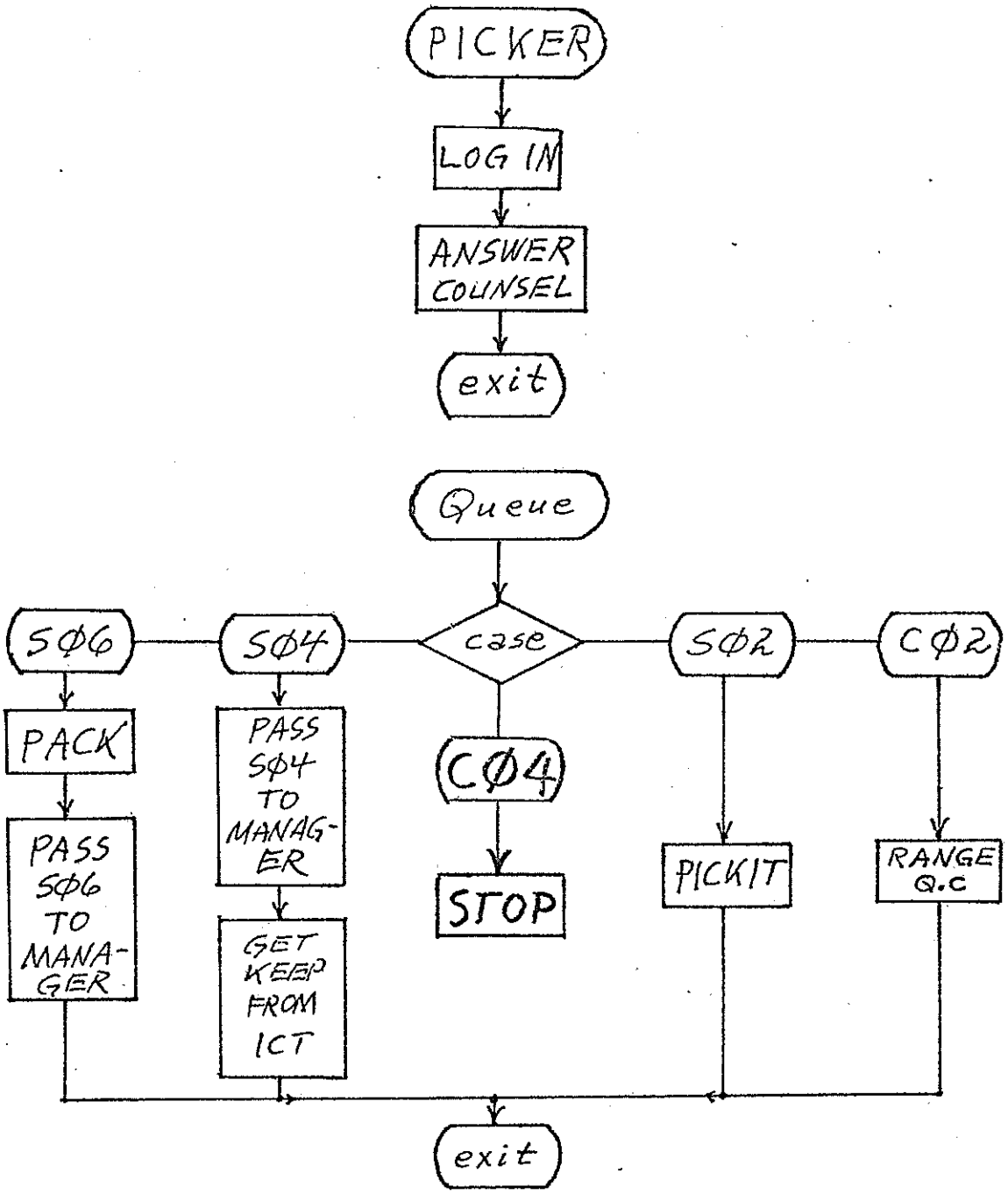


Figure 3.1. PICKER

3.2. Input

PICKER keeps ranges with at least M echoes in them, where M is in location ICT.M (Section 7.1) and may be changed with command KEEP (Chapter 8).

The CØ2 Q-block from COUNSEL defines range limits on echo locating which can be used to speed processing (Section 3.10).

Q-blocks SØ4 and SØ6 from SOUNDER start and stop an ionogram and are passed on to MANAGER (Section 4.2).

The SØ2 Q-block (Section 3.4) contains the address of the Pulse Configuration Table (PCT, Section 7.2) and the FEP BUFFER.

3.3. Output

PICKER answers COUNSEL and sends a logging message during start-up.

SØ4 and SØ6 Q-blocks are sent to MANAGER; the SØ4 immediately upon receipt, and the SØ6 after the last data block has been sent.

PICKER sends a PØ2 Q-block to MANAGER giving the location of a buffer containing:

Length in bytes, 1 word.
PPPSET (currently 4) PCT blocks, 5 words each.
P.MAXEKO (currently 8) ECHO blocks, 17 words each.

The ECHO blocks are a range in microseconds and 8 X-Y pairs.

See Section 4.3 for details of the output.

3.4. PICKIT

Subroutine PICKIT, Figure 3.2, has the following input and output.

Input SØ2 Queue block:

Ø	'Ø2', 'S' (bytes Ø and 1)
2	Status, '-1ØØØ' in, 'ØØØØ' out.
4	Address of BUFFER.
6	Address of PCT.

BUFFER contains data from channels 1 and 2 in the format given below. The first two blocks contain derived data generated by the FEP for each channel. X_o and Y_o are

3.4. PICKIT

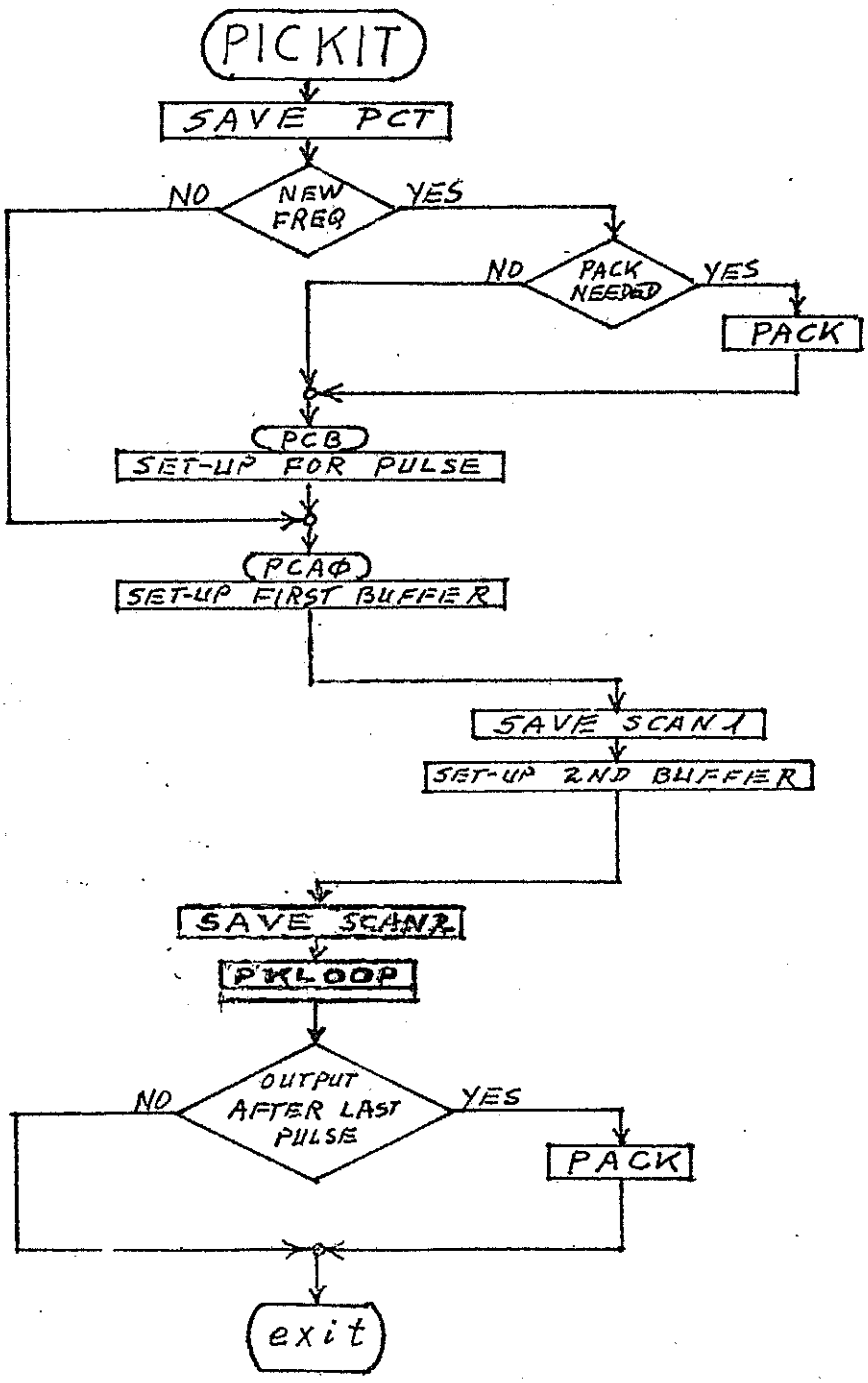


Figure 3.2. PICKIT

zero-offsets related to calibration; they are subtracted from the data. The mean is at present unused. The peak enters into the average peak computed by SOUNDER for comparison with PKHI and PKLO (Chapter 8) to accomplish automatic gain control. R is the range for each set of data; R_o is always zero and corresponds to the calibration pulse. P points to the next R.

Ø	length	
1	start of channel 2 data, address B2	
2	X_o	channel 1 derived data
3	Y_o	
4	Mean	
5	Peak	
6	X_o	channel 2 derived data
7	Y_o	
8	Mean	
9	Peak	
10	Ø	unused
11	Ø	
12	Ø	
13	Ø	
14	R_o	channel 1 data
15	P_o	
16	X_o	
17	Y	
-	---	
B2	R_o	channel 2 data
B2+1	P_o	
B2+2	X_o	
B2+3	Y	
-	---	

Output

TABLE, used by PACK to form a PØ2 Queue block (Section 3.3). TABLE has P.NECHO (SNDR parameter, currently 22) blocks of 18 words each.

The descriptions below give byte addresses.

TABLE + Ø contains the number of ranges included in this average range.

3.4. PICKIT

TABLE + 2 contains the average range in microseconds corrected for antenna, receiver, and TSG delays (ICT.ADLY, ICT.TDLY).

TABLE + 4 + 4*n contains the nth X value.

TABLE + 6 + 4*n contains the nth Y value.

3.5. PKLOOP

PKLOOP finds a range in the TABLE defined in PICKIT according to the flowchart in Figure 3.3.

Input — BUFFER from SOUNDER.
Registers 1 - 3 from SCAN1 and SCAN2.

Output — TABLE as defined in the description of PICKIT.

Registers — PKLOOP makes the following internal use of registers:

- RØ: always zero.
- R1: Ø: SCAN1 range is smaller.
  1: SCAN2 range is smaller.
  2: both ranges are within P.MICRO (currently 6Ø) microseconds of each other.
- R2: range from SCAN2.
- R3: range from SCAN1, then count of items averaged into new range.
- R5: range in TABLE.
- R6: current table location.
- RA - RF: multiple register move.

3.6. SCAN

Input — SCANn table, n = 1 or 2.

SOUNDER BUFFER.

Registers: all 16 registers are loaded from SCANn before SCAN is called. Details about register use appear below.

SCAN1 defines the processing of the data from channel 1; SCAN2 does the same for channel 2.

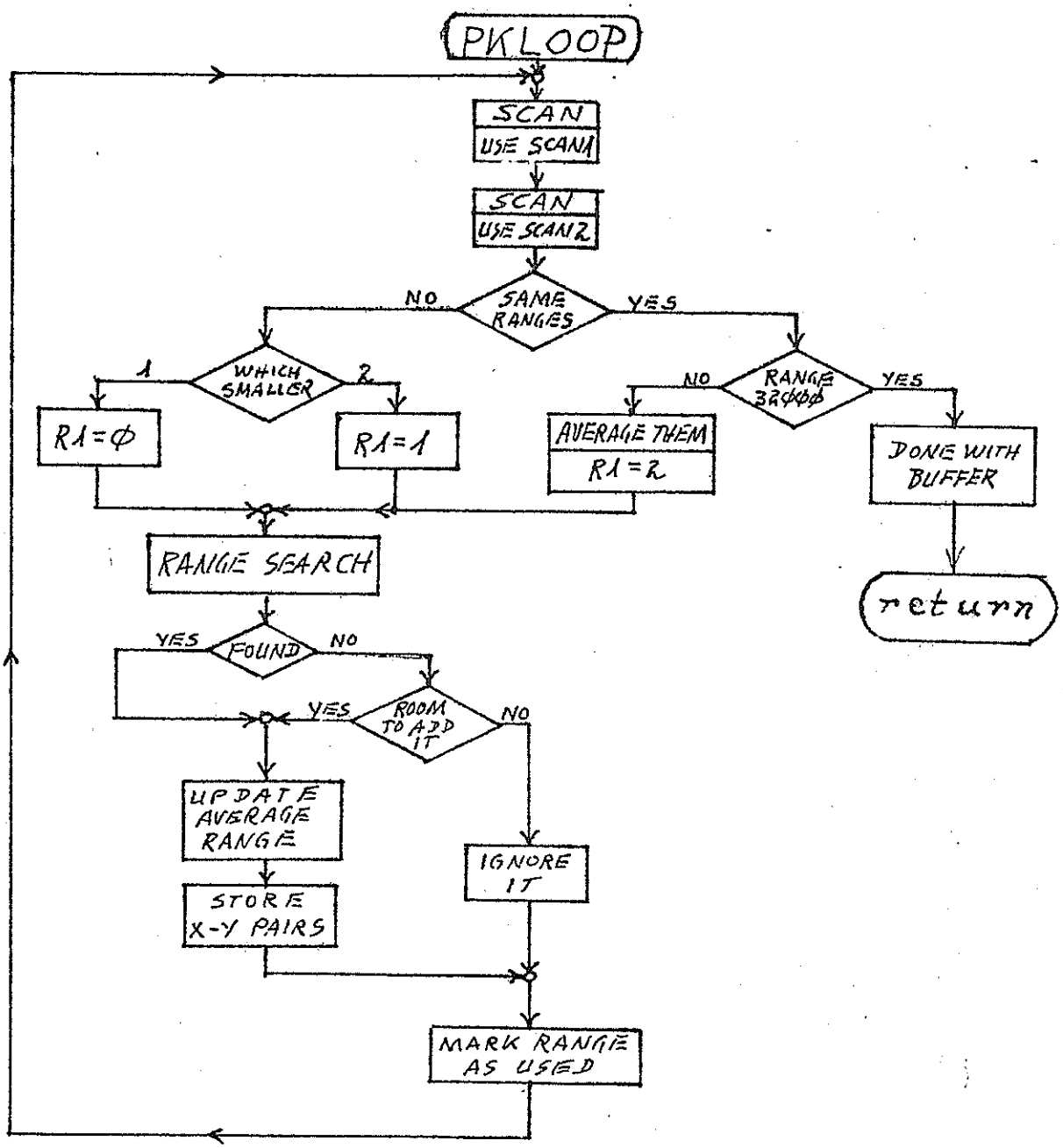


Figure 3.3. PKLOOP

3.6. SCAN

Output

SCANn table updated.

All registers except RØ are saved after exit from SCAN. Many of these registers are set by PICKIT to define the two buffers.

Method

Refer to Figure 3.4.

A single pass is made through the buffer data. PEAKR is called for any peak established and then a return is made to PKLOOP.

During set-up for processing a new echo, if the echo does not overlap the range specified by the RANGE command, the echo is not processed. A second check is made within PEAKR so that SCAN will not return a value outside the requested range.

For VUF and KRUN the range is always tested against the frequency table given in the SØ2 block. For the ionogram mode, the range will be tested only if the operator has given a RANGE command. The same range limits will be used for all frequencies.

SCAN finds a local peak using amplitudes from JABS. If the second point after this peak is higher, the search continues for a local peak. If the second point is not higher, the local peak index is saved and SCAN looks for a second local peak. If SCAN moves four points past the first peak without detecting a second peak, the first peak is established. If a second peak is found, the higher of the two peaks is established. Peaks with less than three points between them (4Ø microseconds) are never output as two separate peaks.

Registers

The following describes register usage in detail. RØ through R2 are used as temporary registers within SCAN.

RØ Defined only on input as the start of the amplitude buffer. It is not saved on exit (which preserves the input value).

R1 X-BAR out, output only if a new range is computed. X-BAR is the average of the peak X and the next X.

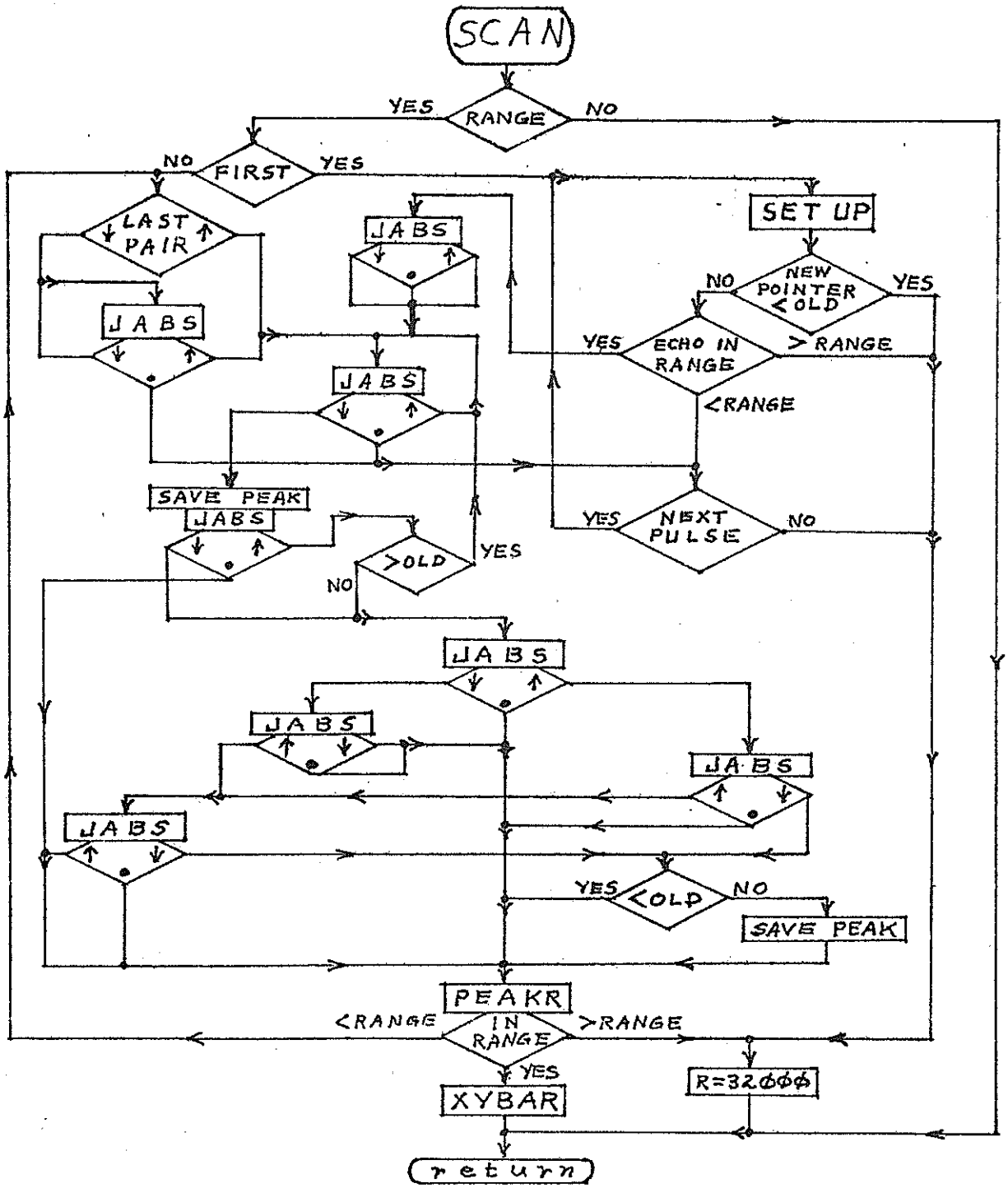


Figure 3.4. SCAN

3.6. SCAN

R2 Y-BAR out, output only if a new range is computed. Y-BAR is the average of the peak Y and the next Y.

R3 The calculated range, which is the essential output of this subroutine. SCAN interprets the input values of R3 as follows. R3 = -1 for the call to SCAN the first time within a buffer (to do initialization). R3 = Ø if the previous range was used and a new range is to be calculated. R3 > Ø if no new range is needed (immediate exit).

R3 is calculated by a five-point, least squares quadratic through the points surrounding the peak amplitude (see PEAKR method, Section 3.7).

If there are no more echoes, the range is set to 32ØØØ microseconds, recognized as too large to be real.

R4 X_o, the zero-offset in X from BUFFER (Section 3.4).

R5 Y_o, the zero-offset in Y from BUFFER (Section 3.4).

R6 Current amplitude for comparison with the new amplitude.

R7 Base range for the echo. The range within the echo is added to this range.

R8 Subroutine link. RF was not used because at one time the subroutine used LM and STM regularly:

```
LM   RØ,SCANn
BAL  R8,SCAN
STM  R1,SCANn + 2
```

R9 The end of the buffer, meaning that there are no more echoes past this index.

RA The index of the peak, offset by PX (internal variable, -4 bytes).

RB Index for sum of absolute values of X and Y, an approximate amplitude. These amplitudes overlap the data in the buffer. They start at the value given in RØ for each echo.

RC Index of the last X-Y pair in the echo set. It is used to test for done. It is offset from the true end of echo by 2*ENDB (12 bytes).

RD This is the index for a suspected or verified peak. It is offset from the true peak by PXY (internal variable, currently -8).

RE Index of the current X-Y pair being examined.

RF The location of BUFFER from SOUNDER. This is the base address for all pointers.

3.7. PEAKR

Input

SOUNDER BUFFER (Section 3.4).
Registers:
R4: X_o, zero-offset.
R5: Y_o, zero-offset.
R7: Base range.
RA: peak index of Amplitude.
RD: peak index of X.

Output

R1: X-BAR.
R2: Y-BAR.
R3: range in microseconds.

Method

A through E are five successive amplitudes with C the peak value.

$$R1 = [X(peak) + X(next)]/2 - X_o$$

$$R2 = [Y(peak) + Y(next)]/2 - Y_o$$

$$\text{new range} = 7*[(A - D/2) + (B/2 - E)]/[(A - D/2) - C - (B/2 - E)]$$

$$R3(range) = 5*[RANGE + \text{new range}] - ICT.ADLY + ICT.TDLY$$

Where:

RANGE = R7(base range) + R1(peak index) -
R0(defined at input)

If RANGE exceeds MAXIMUM, then R = 32000; If RANGE is less than MINIMUM, the last pair is tested to search for a new range (see the PEAKR block in Figure 3.4). MAXIMUM and

3.7. PEAKR

MINIMUM are defined by the command RANGE described in Section 3.9 under Q.C, or by the SØ2 Q-block during KRUN and VUF.

3.8. JAPS

Input — SOUNDER BUFFER.
R2: subroutine link.
R6: old amplitude.
RB: index of amplitudes.
RC: end of BUFFER to compare to RE.
RE: index within BUFFER.

Output — RØ: temporary.
R6: new amplitude.
RB and RE: indexes stepped to next point.

Method — The amplitude is calculated as the sum of the absolute values of X and Y. This amplitude estimate is fully adequate to locate peaks because the peak is expected to have constant phase across it.

Normal exit — End of BUFFER reached. Indicated as a "." in Figure 3.4 (SCAN).

Skip exit — Skips over a full word instruction (normal branch) if not at the end of the buffer. The new amplitude has been compared to the old. A conditional branch is normally used at this point and is indicated as "↑" (going up) or "↓" (going down) in Figure 3.4.

3.9. PACK

This subroutine packs into a PØ2 buffer the PCB's and data in TABLE (defined in PICKIT, 3.4) that satisfy the M-out-of-N criterion.

Figure 3.5 describes the flow of PACK.

MLOST and NLOST are internal variables set to zero during system generation. Their use, described below, serves as an indication of system loading.

MLOST and NLOST are computed only if PIK.TEST is not zero. TESTNMFS in PICKER.MAP points at NLOST. MLOST is the next word after NLOST.

3.9. PACK

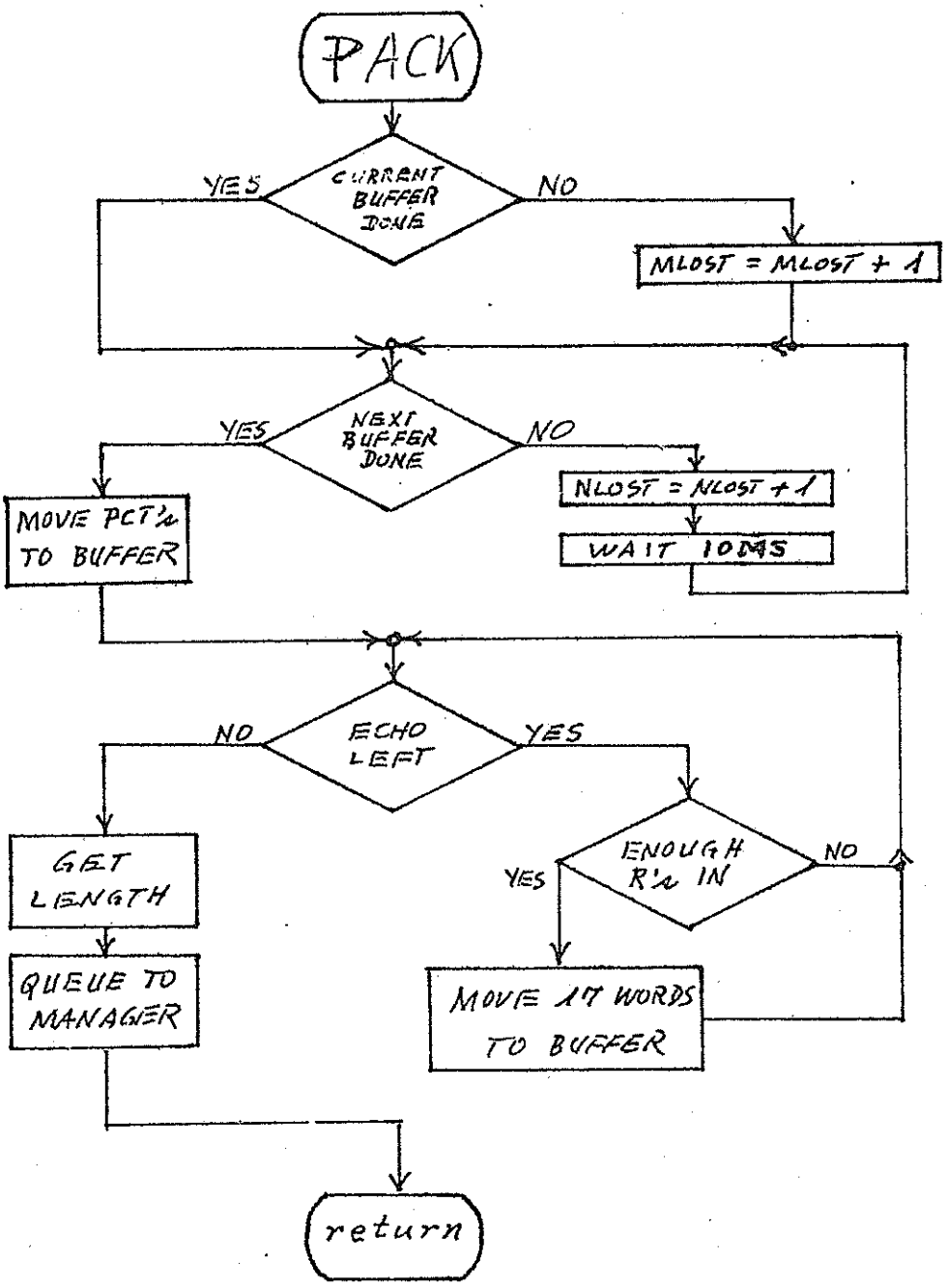


Figure 3.5. PACK

3.9. PACK

MLOST is incremented if the previous PØ2 (Section 3.3) block is still busy. This can indicate whether two blocks are really necessary.

NLOST is incremented if the PØ2 block to be packed is still busy. PICKER then waits 1Ø milliseconds and tries again until the block is free.

Input — TABLE from PICKIT.
ICT.M (Section 7.1) set by command KEEP (Chapter 8).

Output — PØ2 queue block.

Registers:
Ø: Ø2 'P'
2: status = '-1ØØØ'
4: PICKRn, n = 1 or 2

The total count of peaks detected by SCAN is accumulated during packing into FTIME. The count of peaks actually packed into PØ2 blocks is accumulated in STIME. STIME*256/FTIME is stored in ICT.ECHO at the end of a sounding.

ICT.ECHO is an indication of the noise rejection by PICKER. If FTIME overflows, ICT.ECHO is zero until STIME overflows, after which ICT.ECHO can be almost anything. Overflows occur when 65536 peaks are counted (many peaks at many frequencies).

ICT.ECHO is not required for processing, so its questionable validity at very large counts is not considered serious.

3.10. Q.C

Q.C processes Q-block CØ2 from COUNSEL in PICKER (see Figure 3.2).

Input — RANGE [n] typed by the operator (see below).

Output — PICKER returns only ranges within those defined by RANGE.

Method — If n is zero or missing, no range testing is done. Otherwise, the ranges set up for frequency n in the K-mode (see commands RN and RF in Chapter 8) are used to speed processing by ignoring echoes outside the assigned range. Range intervals should be somewhat wider than actually required to prevent loss of echoes near the limits. Normal time jitter may put some of the echo returns just outside the range so that fewer returns are chosen for coincidence.

4. MANAGER

James R. Winkelman

4.1. Introduction

This chapter describes the MANAGER task and the routines MANAGER uses.

In Product One, MANAGER tranfers data from PICKER to the disc and, at the end of an ionogram, from the disc to magnetic tape. MANAGER supplies GRAPHER and ANALYSER with the location of PICKER output.

A flowchart of MANAGER and its Q-blocks appears in Figure 4.1.

4.2. Input

Q-blocks SØ4 and SØ6 start and stop an ionogram and are passed on to ANALYSER and GRAPHER.

CØ2 Q-blocks from COUNSEL are initiated by the commands TAPE [n], FORMAT [n], SORT [n], CLOSE, DROP m, SAVE m, COPY n, TNEW, and NOANALYSER. These commands are described in Section 4.7.

The PØ2 buffer contains the actual data.

4.3. Tape and Disc Output

MANAGER outputs an ionogram on disc and the same ionogram as the next file on tape followed by a double end-of-file, if the operator requested tape.

The tape file consists of an ICT (Ionogram Configuration Table, Sect. 7.1) record followed by PØ2 data records. The data records are fixed length, using as many bytes as would be required by the longest possible record.

The remainder of this section describes the tape format.

Definitions

Byte: 8-bit data quantity.

Word: 16-bit, 2-byte quantity. The Interdata computer's 2's-complement arithmetic means that a word W greater than or equal to 32768 has the arithmetic value W - 65536.

ICT: The Ionogram Configuration Table is described in Section 7.1, but the displacement order is that given below rather than the alphabetic order of the

4.3. Tape and Disc Output

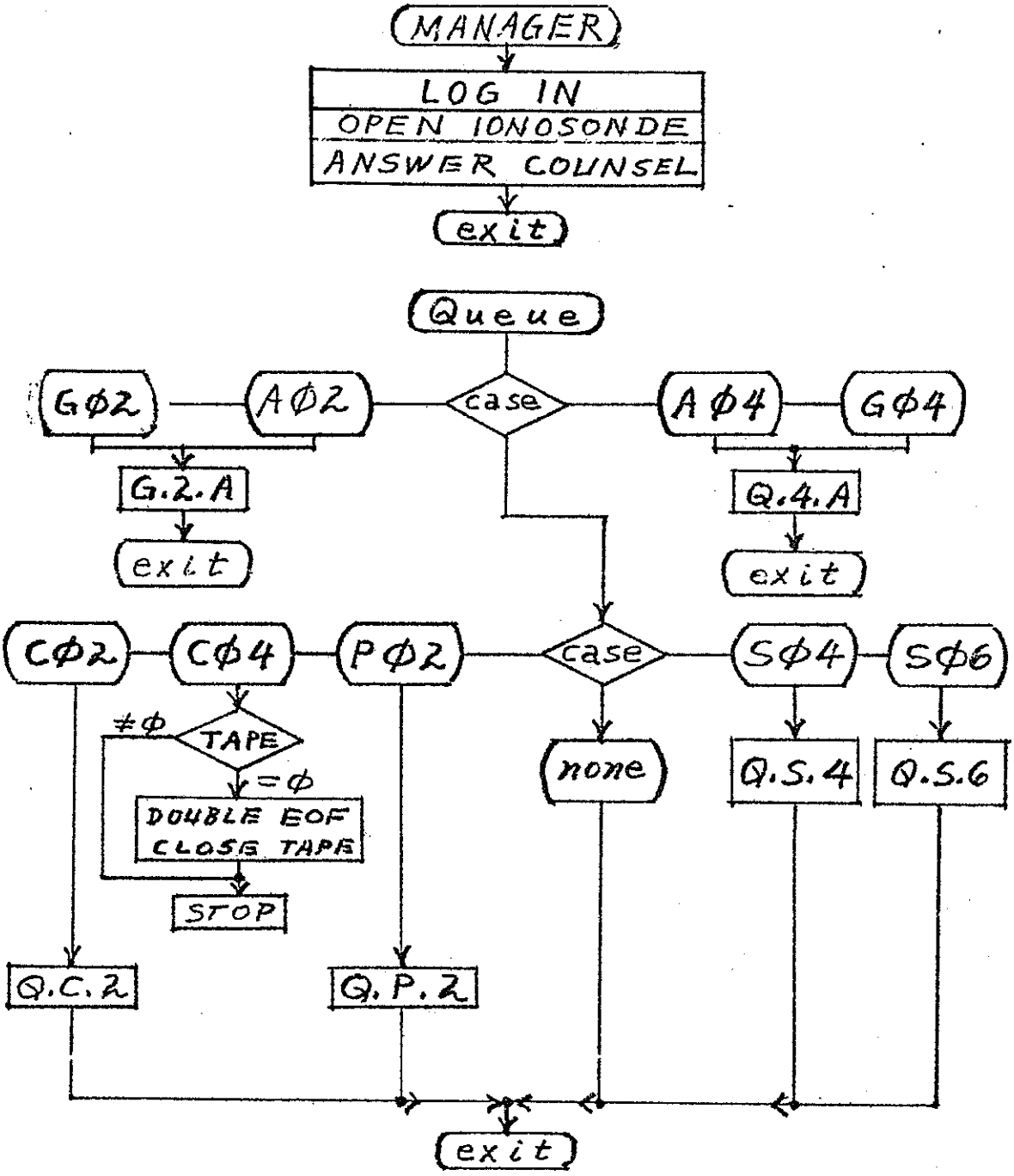


Figure 4.1. MANAGER AND QUEUE

descriptions.

PCT: The Pulse Configuration Table is described in Section 7.2 in the same order as it appears in a tape file.

Format — One file (sounding) consists of the ICT followed by any number of Data Records as described below.

ICT — The ICT currently consists of 194 bytes beginning with the length at displacement Ø.

The displacements appear in the order given in Table 4.1 (read column-by-column).

Data Record — A Data Record takes up 42 + 34*n bytes, or 21 + 17*n words, where n is the number of ECHO BLOCKS in the record; n may be Ø. The fixed 21 words at the start of each block contain the record length and four PCT's.

A Data Record contains the following words in default sequence (see command PSEQ, Chapter 8):

	WORD	DESCRIPTION
	1	Byte count for the Record.
	2-6	PCT, 1st antenna pair, frequency F.
	7-11	PCT, 2nd ANT pair, frequency F.
	12-16	PCT, 1st ANT pair, frequency F + deltaF.
	17-21	PCT, 2nd ANT pair, frequency F + deltaF.
ECHO BLOCK	22	First word of ECHO BLOCK1, 17 words: R-BAR = average range value (echo time) in microseconds of all echoes in BLOCK1.
	23	X-amplitude of first echo.
	24	Y-amplitude of first echo.
	25-26	X- and Y-amplitudes of second echo.
	. . .	. . .
	37-38	X- and Y-amplitudes of eighth echo.
	39	ECHO BLOCK2, R-BAR.
	4Ø-55	ECHO BLOCK2, 8 X-Y amplitude pairs.
	. . .	. . .
	9Ø-1Ø6	ECHO BLOCK5.
	. . .	. . .
	22+17*(n-1)	ECHO BLOCKn, first word.
	. . .	. . .
	22+17*n-1	ECHO BLOCKn, last word.

4.3. Tape and Disc Output

DISPLACEMENT	BYTES	DISPLACEMENT	BYTES
LEN	1	2YØ	2
STAT	1	ADLY	2
MODE	1	TDLY	2
NKFR	1	OPOB	2
LAT	4	M	1
LONG	4	N	1
SITE	4	DGA	1
DESC	8	FGA	1
STS	4	DELF	1
STM	2	LTIM	1
ETS	4	TSGM	3
TSGC	1	FEPM	3
FEPC	1	FEPT	1
15 more pairs	3Ø	ECHO	1
SFRD	4	REVC	2
EFRD	4	REVS	2
SFRQ	4	REVP	2
EFRQ	4	REVM	2
NI	2	REVG	2
NIPH	1	REVA	2
MRDP	1	C1A1	6
INUM	2	C1A2	6
SFIX	2	C1A3	6
EFIX	2	C1A4	6
DFIX	2	C2A1	6
HPE	2	C2A2	6
EPF	2	C2A3	6
1XØ	2	C2A4	6
1YØ	2	C1OR	4
2XØ	2	C2OR	4

Table 4.1. ICT DISPLACEMENTS

4.4. Method

MANAGER writes data to disc according to the data's frequency index. At the end of an ionogram, the data are copied from the disc to tape, skipping missing data (zero sectors).

4.5. Reads and writes

Reads and writes are done by storing the six-word command in DISK.WR and executing it.

Registers — The registers contain the following information:

RA — Command, function in first byte, logical unit (LU) in second byte. LU = 2 for disc, 3 for tape, and 4 for a save file on disc. The commands are:

3CØ2	write disc starting at given random sector
3CØ4	write save file at given random sector
5CØ2	read disc starting at a given random sector
5CØ4	read save file at given random sector
38Ø3	write tape
88Ø3	write end of file (tape)
82Ø3	backspace file
84Ø3	forward space one file
9ØØ3	forward space one record
Ø8Ø3	wait for tape done

RB — Status on output (unused on input).

RC — First word address in the buffer.

RD — Last word address in the buffer.

RE — Zero as upper half word of a random address for random sector calls.

RF — Random sector address.

4.6. ZERO

Input

SOUNDER BUFFER (address and length in ET, Environment Table, Sect. 7.3).

IONOSOND file on disc

Count-down word from ET.

Output

File IONOSOND is entirely zero if there was time. There is always time during SORT, but only CDN - 1 (1-18) seconds during the GO countdown. CDN is the countdown set in SOUNDER.

Method

Refer to Figure 4.2.

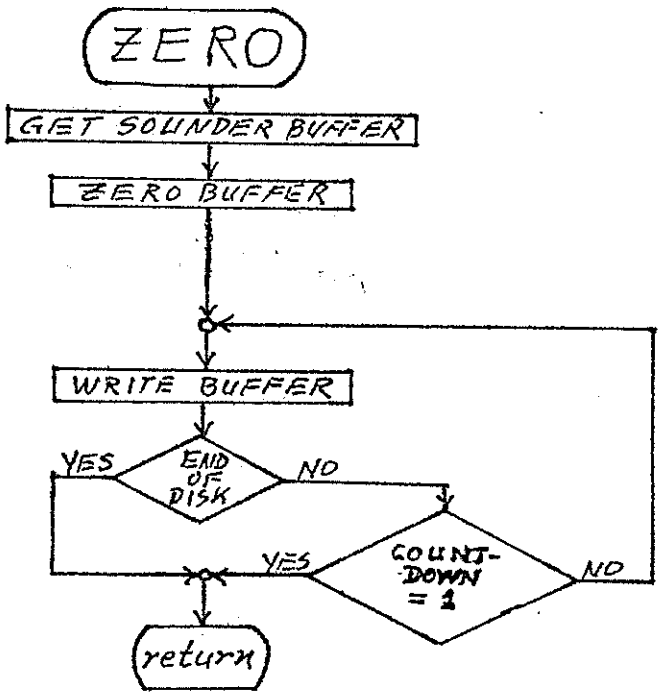


Figure 4.2. ZERO

BUFFER is zeroed and written as long contiguous blocks until end of disc file, or the count-down reaches one second.

4.7. Q.C.2

Q.C.2 processes tape and disc control commands as shown in Figure 4.3. Each command appears below with its options and actions.

TAPE [n] — If the tape is not ready or the write ring is missing, MANAGER responds with "CHECK TAPE".

n = Ø or missing: open tape.

n = 1: open tape, skip to double EOF, and backspace over the second EOF.

Further tape control uses the following three commands.

FORMAT [n] — n = Ø or missing: normal format (default).

n = 1: compact format without EOF.

n > 1: compact format with EOF's between soundings.

SORT [n] — n = Ø or missing: no sort (default).

n = 1: sort ionograms into ascending frequency order; requires LU2 IONOSOND. LU2 is zeroed (see ZERO, Sect. 4.6) by the SORT command to eliminate the sectors for protected frequencies. During sounding, the data in PØ2 Q-blocks are written to the proper sectors of LU2 (see Q.P.2, Sect. 4.10), thus sorting them. After sounding, the non-zero sectors are copied to tape. If the starting frequency and DFIX are not changed after a SORT, further zeroing is unnecessary but is done (as much as possible) during countdown.

CLOSE — Close tape, write double EOF, and backspace. No tape will be written until a COPY or TAPE command; n for the TAPE command must be missing or Ø.

SAVE [m]
DROP [m] — Previous soundings may be saved on disc. The most recent sounding is always available to GRAPHER or ANALYSER. If MANAGER is in the "DROP" mode, the next sounding is written over the current sounding, replacing it. If MANAGER is in the "SAVE" mode, the next sounding is written after the current sounding. The data are packed into the LU4 buffer, normally IONOSOND.SAV, in the compact tape format discussed in Section 4.12.

4.7. Q.C.2

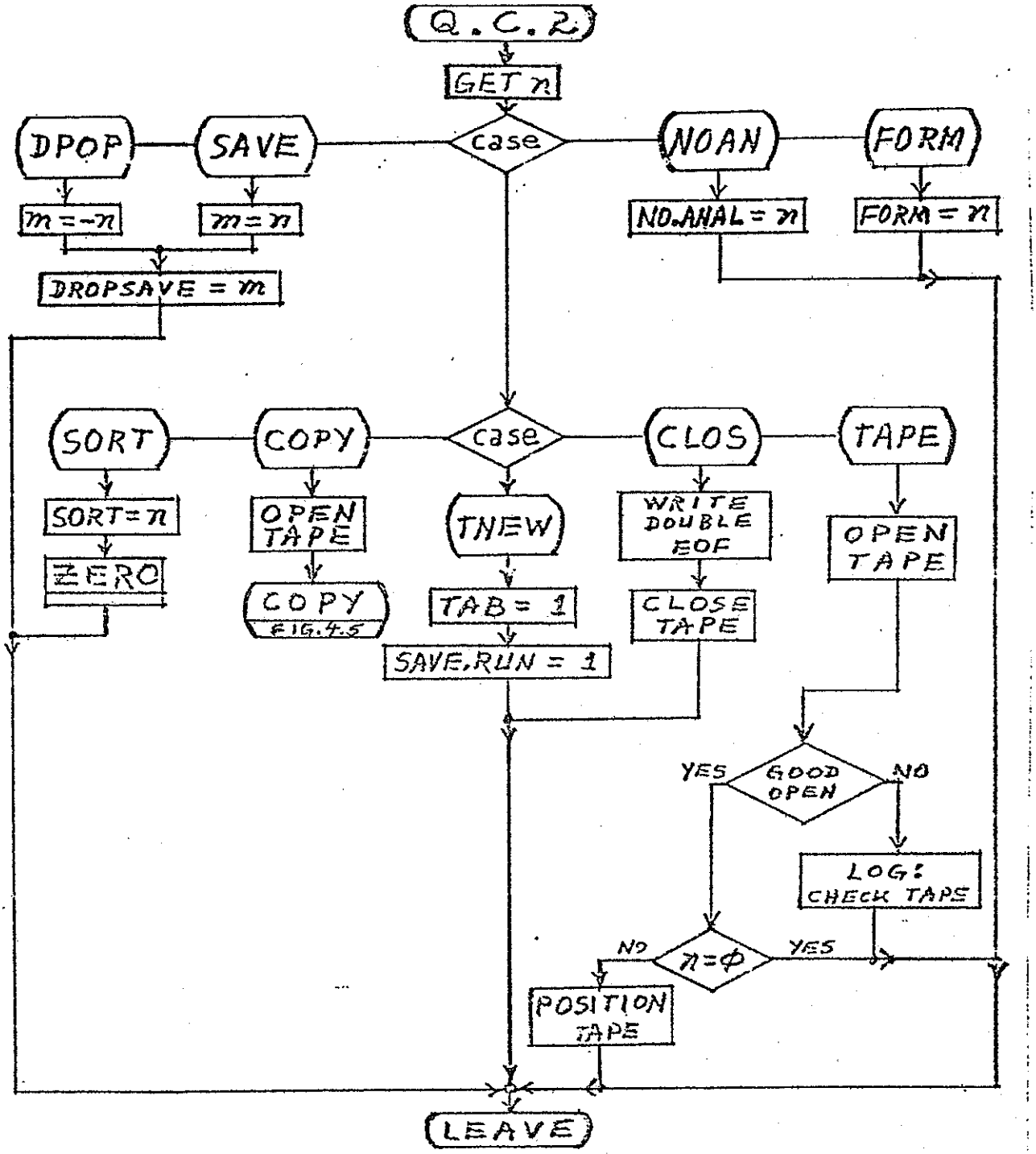


Figure 4.3. Q.C.2

MANAGER starts in the SAVE mode and remains in that mode until a SAVE or DROP is executed.

DROP [m]: MANAGER remains in the DROP mode for m soundings, replacing the current sounding and the next m - 1 soundings, then reverts to the SAVE mode.

SAVE [m]: MANAGER remains in the SAVE mode for m soundings and then reverts to the DROP mode.

If m = Ø or is missing, it is defined as 1.

If m equals or exceeds 999, MANAGER remains in the mode without counting down m.

COPY [n] Copy all of LU4 to tape in compact format.

If n is neither zero nor blank, a double EOF is written after the last record. MANAGER is left in the same mode as CLOSE. No TAPE command is needed except to position the tape if required.

COPY might be used for writing tape after a sequence of soundings at maximum rate or for making all recording to a tape at once so as not to leave the tape on all the time.

ABORT should precede COPY because the time to write a tape may be long and COPY does not recognize SØ4 start Q-blocks. TNEW following a COPY restarts the LU4 file at sector 1.

The most recent S.LENG soundings . SAVE'd are written in standard compact tape format. If more soundings had been made since starting or the last TNEW, they are written as a continuous sequence of 4Ø96-byte records. The ICT is then the next sector in the record containing the end of the previous sounding rather than being the first sector of a new record.

TNEW Rewind LU 4 and restart at sector 1.

NOANALYSER [n]

n = Ø or missing: ignore the ANALYSER task (only SØ4 and SØ6 Q-blocks are passed from SOUNDER). This saves the more than a millisecond it takes to send a PØ2 Q-block and schedule ANALYSER to ignore it.

4.7. Q.C.2

n ≠ Ø: pass Q-blocks to ANALYSER.

If MANAGER was assembled with S.ANAL in SNDR (Sect. 7.6) equal to zero (no ANALYSER, ever), then NOAN causes a SCAN ERROR because the command NOAN doesn't exist.

Two other variables defined in SNDR at assembly time are Q (use SVC6 and SVC9, or use SVC4 and SVC8) and S.LENG. S.LENG = 21 is the number of previous soundings that will be kept. The buffer may hold more or less than S.LENG soundings (some K-mode soundings are very long), but only S.LENG pointers will be saved. S.LENG may not be longer than 125.

4.8. Q.S.4

Input

SØ4 Q-block from SOUNDER.

Some file size information from the ICT.

Output

A zero (or nearly so) IONOSOND file if SORT has been selected.

R3: length of data record in sectors.

R6: highest sector used.

R7: length of data record in bytes.

Method

Refer to Figure 4.4.

ICT.HPE (half-words per echo) is multipled by ICT.EPF (echoes per frequency) and doubled to give bytes per frequency. ICT.OPOR (bytes in header) is added to give bytes per record. This result minus one is output in R7.

The value in R7 is divided by 256 (bytes per sector) and rounded up to give the value in R3.

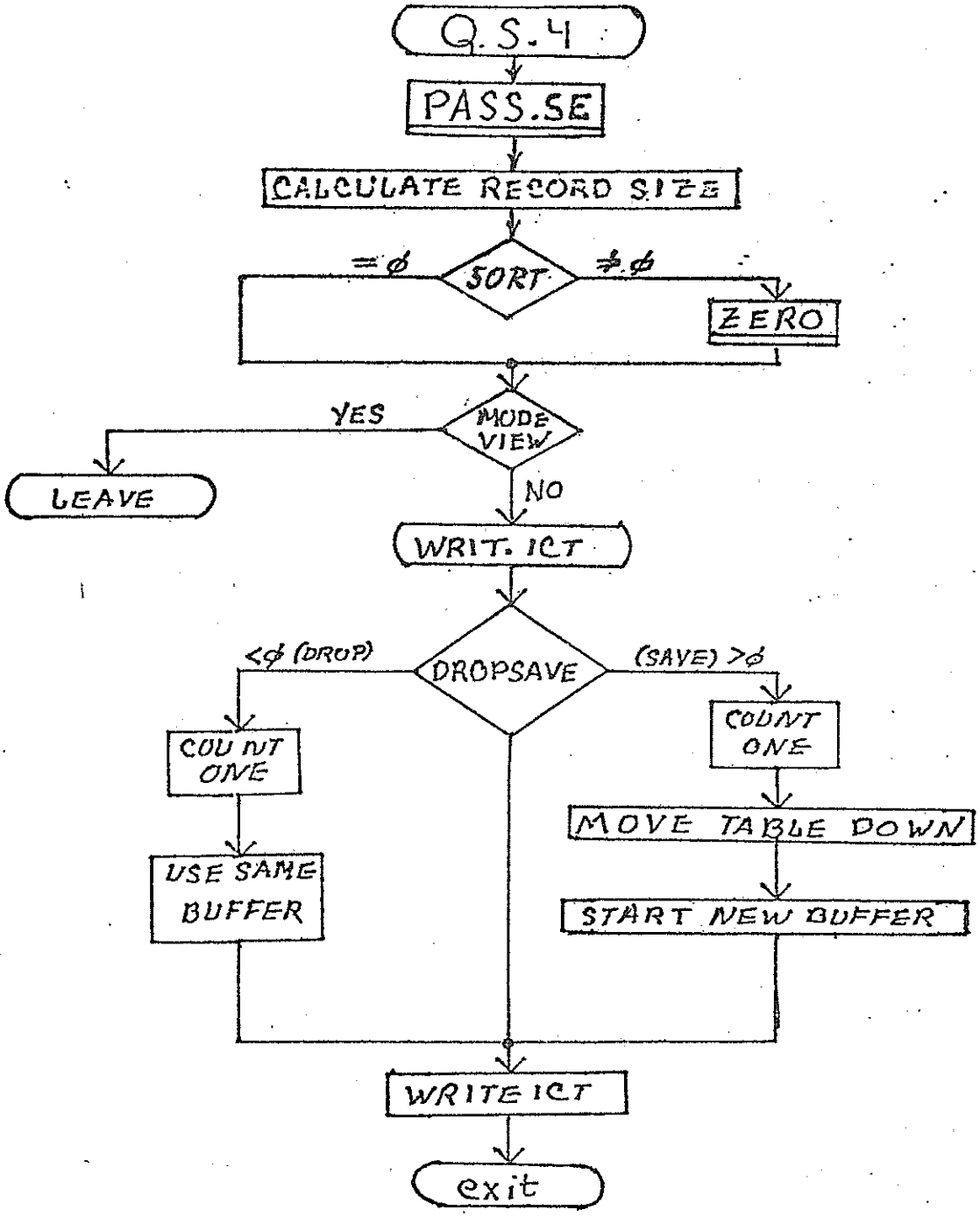


Figure 4.4. Q.S.4, WRITE.ICT

4.9. Q.S.6

Input — SØ6 Q-block from SOUNDER indicating that the sounding is complete.

The ionogram on disc.

Output — One file of data on the tape if the operator has requested tape output.

Method — Refer to Figure 4.5.

The ICT is copied to tape. Sequential records of data are read. If the second word is zero (cannot happen if this frequency was sounded), the record is written onto tape and appended to LU4. After the last record has been written, two EOF's are written and then the tape is backspaced so that it is positioned between the EOF's.

The appropriate sectors of LU4 are copied to tape during the writing of a compact tape (Sect. 4.12).

Registers — R3: length of a record in sectors.

R6: highest sector used.

R7: length of a record in bytes.

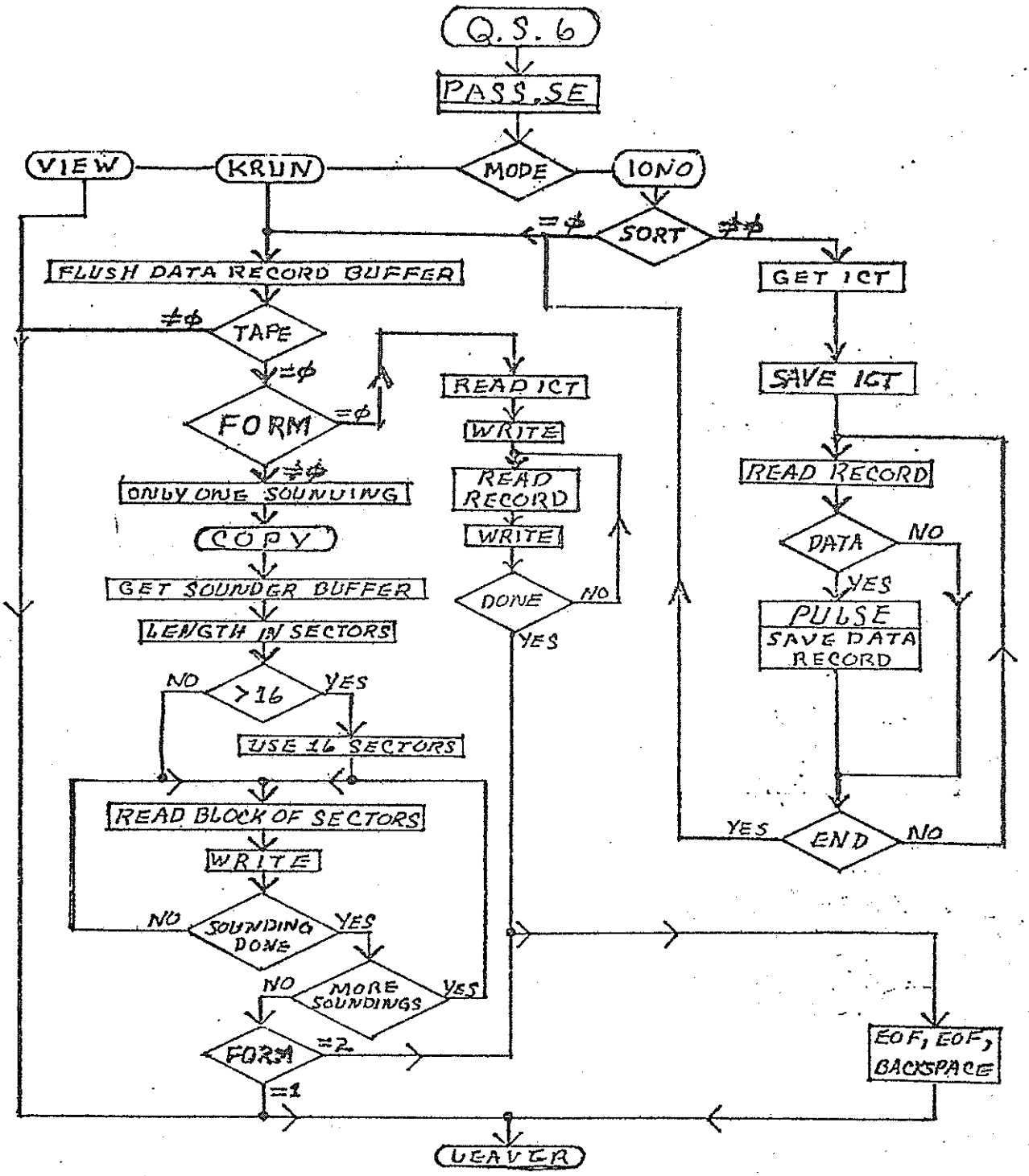


Figure 4.5. Q.S.6 & COPY

4.10. Q.P.2

Input — PØ2 Q-block from PICKER.

R3, R6, and R7 from earlier calls to both Q.P.2 and Q.S.4.

Output — Record on disc, either on LU2 or compacted onto LU4.

Updated R6.

Method — Refer to Figure 4.6.

In the absence of SORT, only the writing of compacted data on LU4 is done.

The buffer from PØ2 is written onto disc at a computed sector location. ICT.SFIX (starting frequency index) is subtracted from the frequency index for this buffer. The result is divided by ICT.DFIX (the frequency index spacing of the soundings) giving a quotient which is an integer number: Ø for the lowest frequency, 1 for the next frequency used, etc.

The quotient is multipled by R3 (the number of sectors per data record) and the product is the sector address used in the write. If the sector used is larger than R6, then R6 is set to that sector.

The actual length of the PØ2 block is used in the write so that only one sector may be written.

4.11. PASS.ON

Input — PØ2, SØ4, or SØ6 Q-block.

Output — The same as the input, with the status changed, sent to ANALYSER and GRAPHER.

Method — Refer to Figure 4.7.

The status is -1ØØØ for each task using the Q-block. MANAGER subtracts 2ØØØ before sending it to both ANALYSER and GRAPHER. If Q-blocks are not sent to ANALYSER, either because S.ANAL = Ø in SNDR (Sect. 7.6) caused no assembly of ANALYSER PASS-ON or because NOANAL = Ø, only 1ØØØ is subtracted.

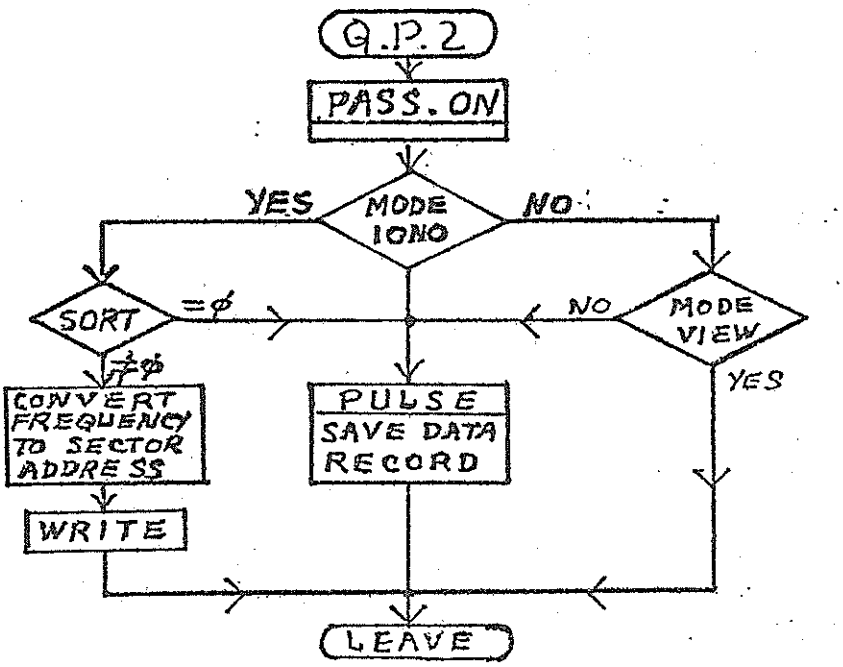


Figure 4.6. Q.P.2

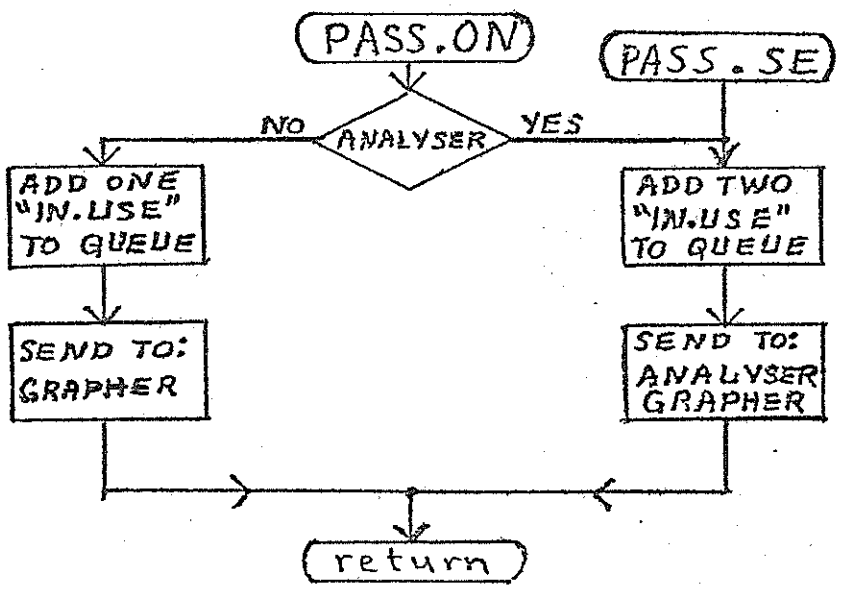


Figure 4.7. PASS.ON and PASS.SE

S.ANAL = Ø sends SØ4 and SØ6 Q-blocks to ANALYSER, but no PØ2 data blocks.

4.12. Disc and Tape Formats

Two different files, each with its own format, are kept on disc. Two tape formats are also available. All formats are based on recording an ICT block followed by as many PØ2 blocks as necessary. The PØ2 blocks are recorded unchanged and are generally called data records.

Logical Unit 2 (LU2) is usually assigned to the contiguous file IONOSOND. This file name is in a specific ASSIGN statement in the CSS file used to load and start a sounding.

LU2 is not used during a K-mode sounding, before the first TAPE command, or after a CLOSE or a COPY.

If SORT ≠ Ø, MANAGER assumes that frequencies are being sounded in a pseudo-random order and sorts the data records onto disc using LU2.

No ICT is saved on LU2. A fixed number of sectors sufficient to hold the largest possible PØ2 block is allotted to each data record. In the current format using 17 words per echo, two sectors are allotted because 8 echoes require 2 sectors. If the maximum number of echoes per PØ2 block were 6 or less, only one sector would be needed. If more than 13 echoes were saved, a third sector would be needed.

Using the current two sectors per record for an example, LU2 is formatted as follows:

SECTOR	CONTENTS
Ø, 1	PØ2(ICT.SFIX)
2, 3	PØ2(ICT.SFIX + ICT.DFIX)
4, 5	PØ2(ICT.SFIX + 2*ICT.DFIX)
. . .	. . .
LAST-1, LAST	PØ2(ICT.EFIX)

where: LAST = 2*(ICT.EFIX - ICT.SFIX)/ICT.DFIX + 1

It follows from this that the length of the file LU2 must be as great as LAST, which may limit the choices of ICT.SFRD, ICT.EFRD, and DFIX. ICT.SFRD and ICT.EFRD are converted to frequency indexes ICT.SFIX and ICT.EFIX.

At the end of the sounding, LU2 is copied to tape in the non-compact format and compacted to LU4 for later reference.

Logical Unit 4 (LU4) is normally file IONOSOND.SAV and can be on either disc. This file should be very large so that it can contain several soundings for later reference.

Sector Ø of LU4 contains the following pointers:

Ø	S.LENG
1	maximum sector in file (length of file)
2	first unused sector
3	first sector of current sounding
4	first sector of previous sounding
...	...
S.LENG + 2	first sector of the S.LENG - 1st sounding before the current one
S.LENG + 3	unused

S.LENG is the number of soundings stored (from SNDR, Sect. 7.6). It is normally less than can be held in sector Ø (125 soundings).

The first unused sector will be used for the next sounding. It defines overlap of old soundings during wrap-around. Between TNEW and wrap-around, the file has valid data from 1 to the first unused sector.

Note that the word before the pointer to the desired sector contains the pointer to the next sector after the desired one.

The maximum sector in the file starts as 2ØØØØ and becomes the actual length of the file during wrap-around.

The pointers in TAB within MANAGER are the same as recorded in sector Ø of LU4. These pointers point to a sector containing the ICT. The "compact tape format" is simply the copying of an integral number of 256-byte sectors (not more than 16) to a tape record and the writing of as many records as necessary. The last record in a sounding and the last record in the file (before wrap-around) may be shorter than the others.

A program reading the first record of a compact tape file would read exactly the same data and format as if that program read 4Ø96 bytes (16 sectors) from LU4 starting at the sector being pointed to.

First sector: ICT block with enough fill at the end to use 256 bytes.

4.12. Disc and Tape Formats

Remaining sectors: the first two bytes are a record number starting with zero and skipping 4883. = X"1313", which is used as an EOF mark on the disc. The data are treated as 254-byte data blocks, ignoring the record numbers. The data format completely ignores sector boundaries.

These data are standard PØ2 Q-blocks in which the first word is the length of the data block. The next PØ2 block starts immediately after this number of bytes in the present block.

For example:

SECTOR BYTES	FILE BYTES	CONTENTS
Ø - 193	Ø - 193	ICT block
	194 - 255	fill (unused)
Ø - 1	256 - 257	Ø, the first data sector
2 - 3	258 - 259	178, size of PØ2 block with 4 echoes
4 - 179	26Ø - 435	remainder of the data record
18Ø - 181	436 - 437	144, size of PØ2 block with 3 echoes
182 - 255	438 - 511	74 more bytes of the PØ2 record
Ø - 1	512 - 513	1, the second data sector
2 - 69	514 - 581	the last 68 bytes of the data record
7Ø -	582 -	the next PØ2 block

Additional data records are essentially copies of this.

A zero length of PØ2 block ends the data. The data blocks break across tape records in the same way they break across sectors.

The format would be more convenient without the sector numbers, but the pseudo-EOF's could occur anywhere in the data and the current procedure avoids them. The compact format is a fast way to write tape and does not waste much tape with record gaps.

The LU4 file, while being useful for writing the compact tape format and for saving several soundings on disc to be written all at once, was designed to provide data from previous soundings to GRAPHER and ANALYSER. Either of these routines sends a GØ2 or AØ2 Q-block to MANAGER to identify the sounding and to get an ICT. They then send GØ4 or AØ4 Q-blocks for each data record. This process ends when an EOF is reached, an SØ4 (start of new sounding) is received, or a new GØ2 or AØ2 is given.

The discussion will refer only to GØ2 and GØ4 because AØ2 and AØ4 serve the same purposes for ANALYSER and discussing only one simplifies the discussion.

Figure 4.1 shows the code entries that process these Q-blocks, and Figure 4.8 expands the code for these entries.

When GRAPHER wants data from a previous sounding, it sends a GØ2 Q-block to MANAGER in the following format:

```
Ø2 'G'
STATUS
A(ICT)          Address for new ICT
SNUM            Sounding number
```

The sounding number is zero for the current sounding, 1 for the previous sounding, 2 for the one before that, etc. This number is used to index into TAB to get the pointer to the sector containing the ICT. TAB is stepped through to recognize soundings which are so old that they have been overwritten.

If the GØ2 comes while MANAGER is still working on an SØ6 (end of sounding) or on an AØ2 and AØ4 request, the GØ2 is stacked for later use. When the data are available, an MØ2 Q-block is sent to GRAPHER with the address of the ICT as the second word. If the data are not available this word is set to zero.

Data records are requested by a GØ4 block from GRAPHER to MANAGER. No information beyond the GØ4 is used from this Q-block. MANAGER gets a data record and returns the address as the second word in an MØ2 Q-block. The data record is in one of SOUNDER's data buffers, which might get zeroed as soon as an SØ4 is sent. This second word is set to zero for an EOF on the selected data block. A new GØ2 is required to either continue to the next sounding or to reread the previous one. A GØ2 requesting file number 999 (obviously illegal) will end the data request before an EOF is reached.

Routine LEAVER (Fig. 4.8) is used as the exit from SØ6 and occasionally GØ4 and AØ4. Its purpose is to allow GØ2 and AØ2 Q-block calls to be stacked, especially after a sounding. For example: at the end of a sounding GRAPHER sees the SØ6, knows that the sounding is done, and sends a GØ2 to MANAGER; MANAGER is still busy writing tape so it stacks the GØ2 request. When the tape is done, an exit through LEAVER takes the GØ2 off the stack and processes it instead of exiting.

4.12. Disc and Tape Formats

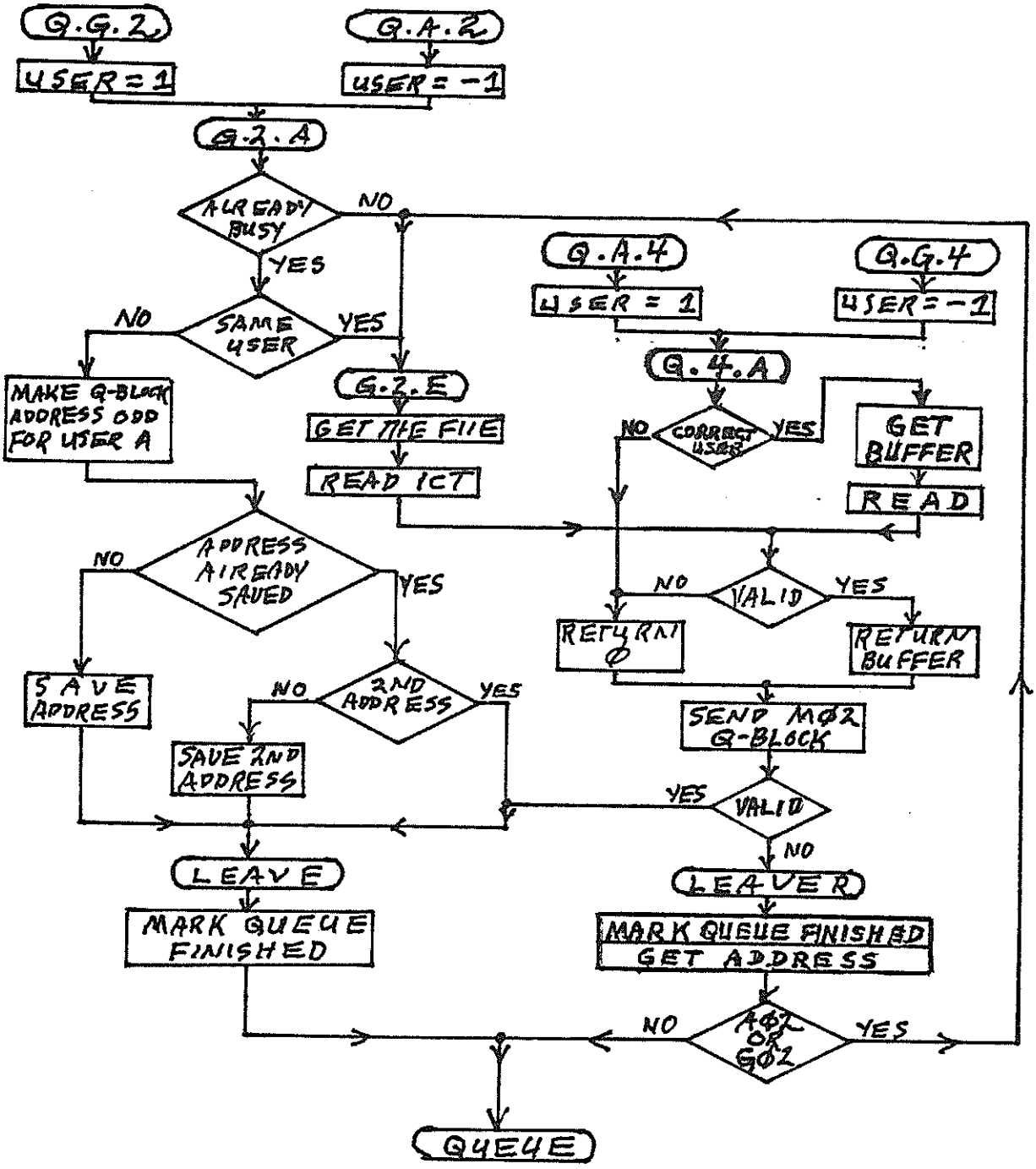


Figure 4.8
Q-block processing, LEAVE, LEAVER

5. GRAPHER

Lorne D. Matheson

5.1. Introduction

Task GRAPHER is responsible for displaying selected portions of the data in suitable formats on the refresh graphics device. It does analysis to select the data to be plotted and converts that data into plotter commands. GRAPHER also handles the display refresh. It has no responsibility for permanent archiving of any digital data. In the event that a camera is permanently attached to the refresh display, GRAPHER would be responsible for the camera control.

In future systems, still under the same basic organization, GRAPHER would handle the light-pen inputs from the refresh CRT and graphics inputs and outputs from the remote graphics terminal, that is, the Tektronix 4012, via a phone line. Graphics input from the 4012 would also be handled by GRAPHER with MANAGER cooperation. Graphics portions of ANALYSER tasks might be handled either by GRAPHER or ANALYSER.

Sections 5.1 through 5.8 contain all of the information necessary for normal operation of GRAPHER and may be considered an operator's manual. All of the GRAPHER commands are described in these sections. Section 5.9 is a brief description of the organization of the tasks, while Section 5.10 describes the plotter driver and hardware. Section 5.11 contains the software flowcharts. These latter sections document various phases of the processing and are needed only for fuller understanding, program modification, or debugging.

Fundamental to a thorough understanding of the present version of GRAPHER is an understanding of the refresh graphics hardware and the I/O driver, PLTDVR, written for it and resident in the system. PLTDVR is called by an SVC1 call (see Section 1.7). The driver normally does an I/O initiate-and-proceed, using the SELCH for output without CPU intervention. At the termination of I/O, GRAPHER receives a Q-block and must re-initiate the I/O for the refreshing of the display. A fuller explanation of the driver, PLTDVR, and the hardware is in Section 5.10.

GRAPHER will exist in two versions, depending on whether a constant DEBUG is set to zero or non-zero at assembly time. DEBUG non-zero adds more diagnostics, taking more space and time. During program development, DEBUG will probably be set non-zero, while it will be zero at assembly time once that particular version seems debugged.

5.1. Introduction

Almost all of the information required by GRAPHER is contained in the Q-block sent automatically by the other tasks, so very few inputs to GRAPHER are required from the console. These inputs will be discussed under the separate program subtasks to which they apply.

As an aid to understanding the task, you may consider the task to be in one of several states or subtasks that continue execution until GRAPHER switches to another subtask. These subtasks are listed below.

Subtask Ø — Provides test patterns and plotter control; no other subtask has yet executed. The display shows a test pattern or the grid for a sample ionogram.

Subtask 1 — Ionogram mode (I-mode). The display is a conventional one of frequency vs. virtual height.

Subtask 2 — Put kinesonde frequency markers on the last ionogram plotted. The last frequency changed is specifically marked, if present.

Subtask 3 — View kinesonde frequency for a short time. The data are not archived and are for only one frequency.

Subtask 4 — Kinesonde run. This is the normal running kinesonde mode (K-mode).

Subtask 5 — Ask for a previous ICT from disc in preparation for display of previous data.

Subtask 6 — Get the previous pulse data from the disc for display.

5.2. Subtask Ø: Test Patterns and Plotter Control

It may be important to verify the operation of the plotting software and hardware independently of the rest of the system's hardware and software. When the task is initially started, a standard test pattern is displayed consisting only of points and vectors. This pattern tests several features of the plotter controller and the refresh logic, but it does not test the character generator. Several long diagonal lines are plotted to display a test of linearity.

Several command sequences that can be used to plot lines and points, but which are not used by the normal ionogram plotting sequences, are used for testing. An alternate display is a sample ionogram grid, which does check the character generator. Several commands from the console that begin with the characters [P may be used to check and control the CRT plotting unit. These commands appear in Chapter 8 alphabetized

without regard to the initial left bracket ([). Several commands beginning with G, whose primary functions are in other subtasks, also change the display.

[PA — Displays only the points from the test pattern. This may be used as a preliminary check on the point plotter, the fundamental block of the plotting hardware. The point test pattern checks quite a few modes of plotting points, including several not used by the normal ionogram plotting.

[PB — Displays only the vectors from the test pattern. Again, several modes are tested that are not used in the normal plotting.

[PZ — Displays the points and vectors in the test pattern, the same display as comes up when the task is initiated.

Once the task has started an ionogram, the plot buffer may overwrite the test pattern, so the above three instructions are treated as NO-OP's.

[PE — Ends the refreshing of the display. This only affects the refresh logic, and does not change the generation of the plot buffer. If it is suspected that during periods that the transmitter is pulsing, there is too much to do in the allotted time, or the I/O requests are somehow interfering with each other, this command may give some relief.

[PS — Starts the refresh logic again. This rescinds [PE or may be used to restart the display if somehow an I/O interrupt was missed.

[PE and [PS are always available, regardless of the subtask that GRAPHER is executing.

[PC — Generates an ionogram grid and labels from the ICT (Ionogram Configuration Table, Section 7.1) stored in the task. A sample ICT is in the task at initiation time; it is changed to the last ICT seen by GRAPHER in Subtasks 1 or 5 later in the processing. This command is primarily used for diagnostics and checking task operation.

The display comes up with a test pattern of points, points and vectors, vectors alone, and vectors and points in a continuous cycle. This is the normal operational mode of the display. Alternation of points and vectors may be terminated by the command GNOMV and restarted by GMOV. GONX and GONO select points or vectors. These commands may be

issued during a sounding and take effect immediately. GONX and GONO have the same effect as [PA and [PB although the mechanism is quite different.

GONO — Suppress the last segment of a multi-segment buffer.

GONX — Suppress the penultimate segment of a multi-segment buffer.

GNOMV — Suppress movement or change of the buffer segment alternations, plotting all simultaneously.

GMOV — Restore movement or alternation of multi-segment buffers.

5.3. Subtask 1: Ionogram Mode

Subtask 1 displays a conventional ionogram, that is, virtual height as a function of frequency. The frequency axis is logarithmic and displays at least 3 to 1Ø MHz. This range is automatically expanded to include all sounded frequencies in a run. Changing the basic 3-to-1Ø MHz range would require changing a few constants in the program and reassembling. The philosophy is to include at least two frequency markers that are labeled, and enough additional frequency tick marks to include all of the sounded frequencies. Points off the bottom or top of the virtual height axis are not plotted. For compatability with Subtask 5, an ionogram from a previous sounding set, the ICT is copied into the task.

This subtask is initiated by a SØ4-Ø SOUNDER start Q-block (the -Ø means that the third word of the block contains Ø, signifying that the next sequence of soundings will be an ionogram). At this time the ICT is copied into the task and the grid for a new ionogram calculated and displayed. The points are added to the plot buffer and displayed as each set is received from PICKER. The end-of-sounding Q-block, SØ6, causes the plot buffer to be stored on the disc for possible later use by subtask 2, below. PICKER Q-blocks before SØ4 or after SØ6 cause error messages to appear on the console.

The PØ2 buffers normally contain a calibration pulse that, when plotted, occurs at a very low height; it is not usually plotted because it contains very little, if any, useful information. The plotting of calibration pulses is controlled by the following two commands.

GCAL — Allow calibration pulses to be plotted.

GNOCA — Plot no calibration pulses.

The above commands may be issued during a sounding and take effect immediately.

The minimum and maximum heights on the ionogram may be changed. Any change takes place at SØ2 time before the next ionogram. The following commands specify heights that are truncated to 1ØØ's of kilometers of virtual height; at least two labels must appear on the height scale.

GHMIN hhh Set the minimum height to hhh km.

GHMAX hhh Set the maximum height to hhh km.

The plot normally alternates display of the ordinary (O) and extraordinary (X) echoes. The display may be altered by the following commands.

GNOMV Suppress alternation of the O and X echoes.

GMOV Restore alternation of the O and X echoes.

GONO Plot the O echoes without the X echoes.

GONX Plot the X echoes without the O echoes.

The differentiation of O and X echoes on the plot requires a means of distinguishing between them in the data. Since this discrimination is needed for each echo but only influences the display and does not change the original data, the computation should be fast and sometimes wrong, rather than slow and always correct. The same algorithm is used for all display modes. The algorithm would like values from orthogonal, physically coincident antennas at the same time. Using our current four-antenna array and sounding sequence, and averaging the N-S antennas and the E-W antennas, the orthogonal, and physically coincident criteria are satisfied, but one set of data is one pulse time later. Therefore, a large doppler shift may confuse the O-X determination. Other antenna sequences may be confused by off-vertical returns, etc.

The method compares the phase of a return with the phase on the orthogonal antenna. Ideally, these would differ by 9Ø degrees with one sense of rotation being O and the other X. The method actually used corresponds to rotating both returns such that the first lies on the plus X axis. The second return is then in the upper or lower half plane, corresponding to O or X. Whether the upper half plane is O or X is best determined by experiment with real data. The test used is the sign of

5.3. Subtask 1: Ionogram Mode

$$-X_B * Y_A + Y_B * X_A$$

where

X_A is the first component of the first antenna;
X_A is the second component of the first antenna;
X_B is the first component of the orthogonal antenna;
Y_B is the second component of the orthogonal antenna.

5.4. Subtask 2: Kinesonde Frequencies on Ionogram Plot

This subtask is used to compare frequencies selected for the kinesonde mode with the last ionogram plotted. It is initiated by a SØ8 Q-block. If an ionogram plot buffer is not present, one is read from the disc. The frequency index markers are added to the plot. If the frequency indicated in SØ8 is not zero, that marker is specifically identified by the character "S".

SØ8 Q-blocks are initiated by commands F, RF, and RN.

Because the plot buffer rather than raw deta is read from the disc, GHMIN, GHMAX, GCAL, and GNOCA have no effect until the next ionogram or kinesonde sounding. GMOV, GNOMV, GONO, and GONX work properly.

5.5. Subtask 3: Kinesonde View

This subtask is initiated by an SØ4-4 Q-block which results from a VUF n command. The nth frequency is sounded repeatedly, eventually followed by an SØ6 Q-block. The X-Y components of the returns are plotted. Two plots are displayed: the left one at frequency F and the right one at F + deltaF. Two points per echo are added to each plot. Normally these two points will differ by 9Ø degrees of phase, with the sense of rotation corresponding to O or X returns. GCAL, GNOCA, GMOV, GNOMV, GONO, and GONX all operate immediately. The normal alternation is between ordinary and extraordinary returns. The center of a short vector indicates the center of the two coordinate systems. The scales of each plot are presently plus or minus 3 volts.

5.6. Subtask 4: Kinesonde Run

This subtask is initiated by an SØ4 Q-block that results from a KRUN command. A K-mode run contains from 1 to 1Ø frequencies. The main purpose of the display is to allow a subjective evaluation of the chosen frequencies during the run. The basic display is the plot of the X-Y components of the return, the same as in the Kinesonde View mode described in Section 5.5.

A run with one frequency has a display identical to the Kinesonde View display. When more than one frequency is present, just the basic frequency is plotted, with no plotting of the F + deltaF return. Two- and three-frequency plots have the same size graphs as the one-frequency plots, with four or more frequencies having the basic graph one-half the size. The frequencies are plotted in order of their occurrence in the KMODE table (Section 7.4). The first valid frequency plots in position one, the second in position two, etc. See Figure 5.1 for the sequencing of frequencies on the display. GCAL, GNOCA, GMOV, GONO, and GONX all operate immediately. The normal alternation of points is between the ordinary and extraordinary returns.

5.7. Subtasks 5 and 6: Display Prior Data Set

These tasks are initiated by the GPION and GPSKY commands described below. GRAPHER asks MANAGER for the proper data set. The plotting options are those currently in use for the particular type of plot. Since the data are reprocessed, scaling may be changed, calibrations added or deleted from the plots, a different type of display selected, etc.

GPION nnn — Get the previous I- or K-mode data set number nnn, where nnn is Ø for the last set stored and counts back in time.

GPSKY nnn — Get the previous data set as in GPION and display in skymap mode (S-mode).

GOVER n — Force GRAPHER overlay n (n = 1 to 3) for processing and display. GOVER Ø restores automatic selection of overlay. See Section 5.9.

5.8. Skymap Data Display

This display shows the apparent point of reflection with the zenith being the center and the scales being kilometers from the zenith, assuming the reflection takes place at the virtual height. North is up with east to the right. The inner marks are at one-third the distance of the outer marks. The default range is 60 kilometers for the outer marks. The formula used is an approximation by Dr. A. K. Paul:

$$P_X' = (h'/f)*c/(2*pi*d)*(X_E*Y_W - X_W*Y_E)/(X_E*X_W + Y_E*_W)$$

where

P_X' is the apparent east deviation of the reflection point in km;
h' is the virtual height in km;
f is the frequency in MHz;
d is the distance between the east and west antennas in meters;
c is the velocity of light in m/micresecond (c. 300);
X_E is the first component of the echo on the east antenna.
Y_E is the second component of the echo on the east antenna.
X_W and Y_W are the west antenna components.

The version presently on the system assumes that the antenna array is rotated 45 degrees such that the north antenna actually lies to the NE. The output data are rotated 45 degrees to correctly aline north. It is not obvious how to insure that all of the hardware phasings are correct so that the formula works correctly and is in the right sense. The computations are critically dependent on correct phasing of all of the receivers. Incorrect phasing in an I- or K-mode display may confuse O-X separation, or change the sense of rotation in the kinesonde data, or both, but the basic display looks correct.

The S-mode display may be generated by a GPSKY nnn command for data off the disc or a GOVER 3 command to force GRAPHER overlay 3 to be resident for real-time display during either I- or K-mode runs. GCAL, GNOCA, GMOV, GNOMV, GONO, and GONX all operate immediately. SFMAX and SFMIN in GRAFPRO are the maximum and minimum frequency indices to be plotted on the display. Presently they must be changed using a MEMORY command, but suitable commands will be added later.

5.9. GRAPHER Task Organization

After initialization, no COUNSEL Q-blocks to Grapher change the subtask being executed. All changes in subtask state are caused by Q-blocks from SOUNDER, or the GPION nnn or GPSKY nnn commands and their MANAGER responses.

At present, I/O consists of refreshing the plotter, and writing the last ionogram plot buffer to the disc and retrieving it when necessary. The refresh is handled by a standard SVC1, initiate and procede, under SELCH control with a queue message when the I/O is finished. The task must then re-initiate the I/O for refresh graphics to continue. After the first ionogram has started, the test pattern may have been overwritten and is considered no longer available. The CRT plotter is assigned to logical unit 9 and the file for the plotter buffer, SYS1:GRAPHER.PLT, is assigned to unit 8. Unit 7 is assigned to GRAPHER.LMO, the overlay file. Units 7 and 8 are programmed as I/O and wait until complete.

Presently three overlays are used for processing different types of displays. They are on file GRAPHER.LMO and are named GROV1 through GROV3. All of the overlays can plot both ionogram and kinesonde data in some fashion. Each overlay has at least two parts, one to do the frame and setup and the other to process the buffer from PICKER.

GROV1 plots I-mode data, GROV2 K-mode data, and GROV3 makes an S-mode plot of any data. Another overlay, GROVØ, will be added to process some of the messages from COUNSEL. Overlaying is rather primitive and requires the calling task to keep track of the position in the overlay file. The overlay file may only be read forward or rewound and read forward. The rewind and read generates an I/O queue entry upon completion of the rewind. GRAPHER is programmed to ignore this queue.

Three problems existed previously in GRAPHER and its interactions with the system. Two of these have been corrected by code. An additional Q-block, MØ4, was added to stop the refresh CRT during tape write and to start it again upon completion of the tape write. This is necessary to consistantly write tape records of the correct length. An intermittant error exists in the hardware or software, or both, that only shows up during simultaneous tape write and CRT refresh. We hope this problem will be solved in the future and the display will continue to operate while the tape is being written.

An additional modification to GRAPHER was necessary to prevent program hangup if the display is missing or bad. If more than five consecutive I/O calls to the plotter are bad, the refresh logic is turned off just as by a [PE. I/O may be tried again by the [PS for at least five times.

5.9. GRAPHER Task Organization

A problem, not corrected by software, exists if the plot buffers take too little time. Since GRAPHER has a higher priority than COUNSEL, COUNSEL may be locked out; [PE allows COUNSEL to be scheduled.

5.10. Refresh Graphics Hardware and Driver

The hardware operates as a 16-bit device. Each 16-bit word consists of 4 bits of command structure and 12 bits of additional data, often CRT screen addresses. It typically takes two words to specify a point, one of X and one for Y. In order to plot a vector, the point plotter must be sent to the start of a vector with the intensity off and then the increments in X and Y must be sent to the vector generator. Peculiar interactions exist between the character generator and the point and vector generators. The time required for the device to be ready to accept the next word can vary between 1/2 microsecond (accept the next word of an X-Y pair) to milliseconds (plot a long vector at high intensity).

The driver for the CRT, PLTDVR, operates with standard SVC1 calls, using the SELCH, with one exception. If the last word address of the buffer area to be output is even and less than 16 words from the start, then the area referenced is to be interpreted as a table of first-last address pairs. The driver runs through the table before considering the I/O finished and before sending a Q-block/interrupt to the calling task, or before returning to the calling task, if the call was I/O and wait. This feature was added to the driver so that a plot buffer can be constructed with different segments (O and X components) and some or all of the segments plotted without shuffling the plot buffer. This feature of the driver can also be used to blink particular features of the plot. The driver also contains the basics needed to read the light pen, but GRAPHER does not presently use the light pen. The light pen address is stored in the SVC1 parameter block at SVC1.RAD + 2 when the light pen is active.

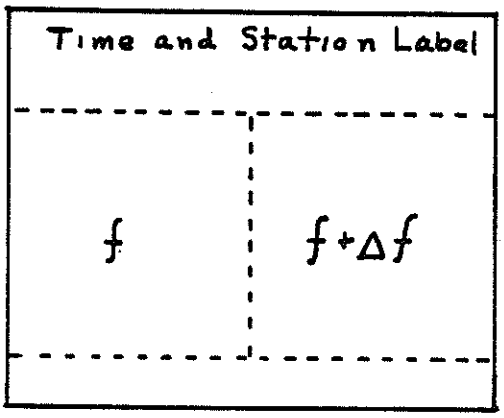
SVC1.RAD is zero upon exit unless the light pen was active. SVC1.RAD upon entry contains the number of times the buffers are to be plotted (once for ≤ 1).

The driver must decide if the starting address and ending address refer to one buffer or to a table of buffer segments. If the parameter block addresses are more than 20 apart, one buffer is assumed. If the end address is odd, one buffer is assumed. Otherwise, these words point to a table of pointers to buffer segments SAD1, EAD1, SAD2, EAD2, SAD3, EAD3, etc.

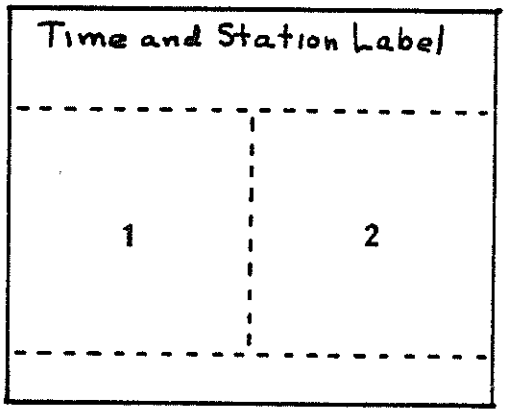
The SVC1 parameter block must have a standard MT2 function code of ØØ1x xxxx, write data transfer. Any other function code will be rejected with a 'CØ' device independent status. Several other errors may cause status errors:

STATUS	CAUSE(S)
'CØ'	Function code not ØØ1x xxxx
	Starting address odd.
	Starting address less than ending address.
'9Ø'	CRT not available for some reason.
'84'	System abort.
'82'	Timeout.

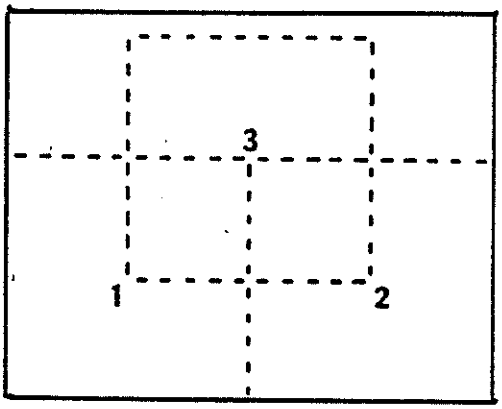
5.11. Figures



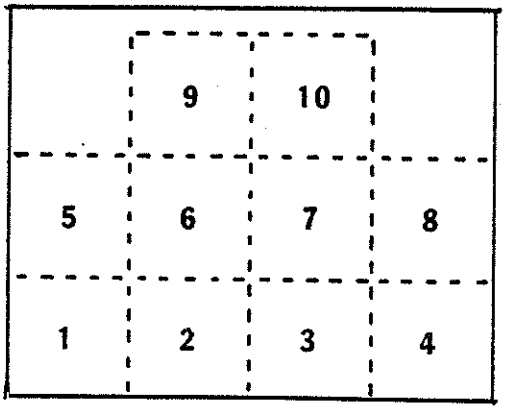
One Frequency



Two Frequencies



Three Frequencies



Four to Ten Frequencies

Figure 5.1
Kinesonde Scan & Kinesonde View Displays

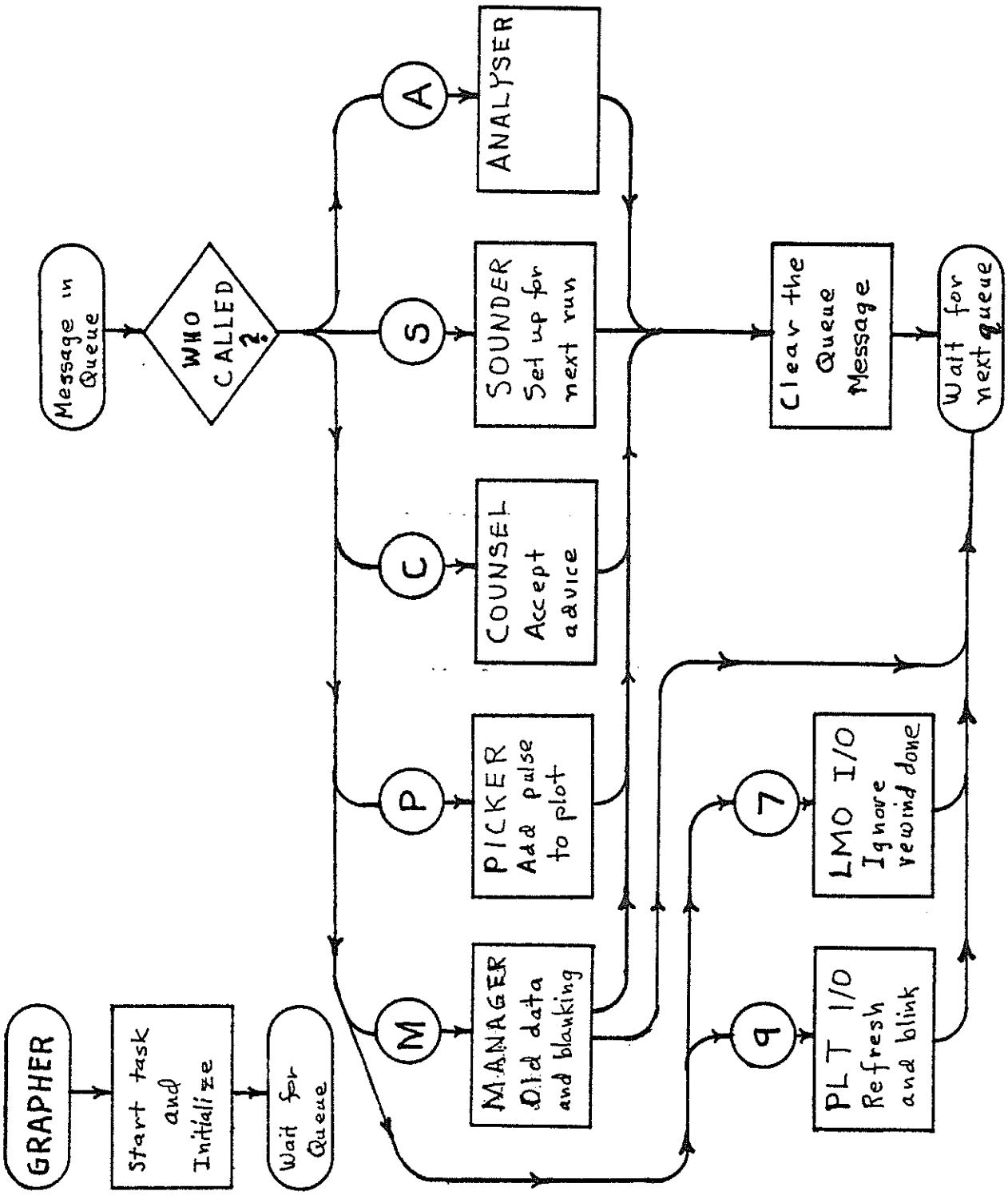


Figure 5.2. GRAPHER & Queue Processing

5.11. Figures

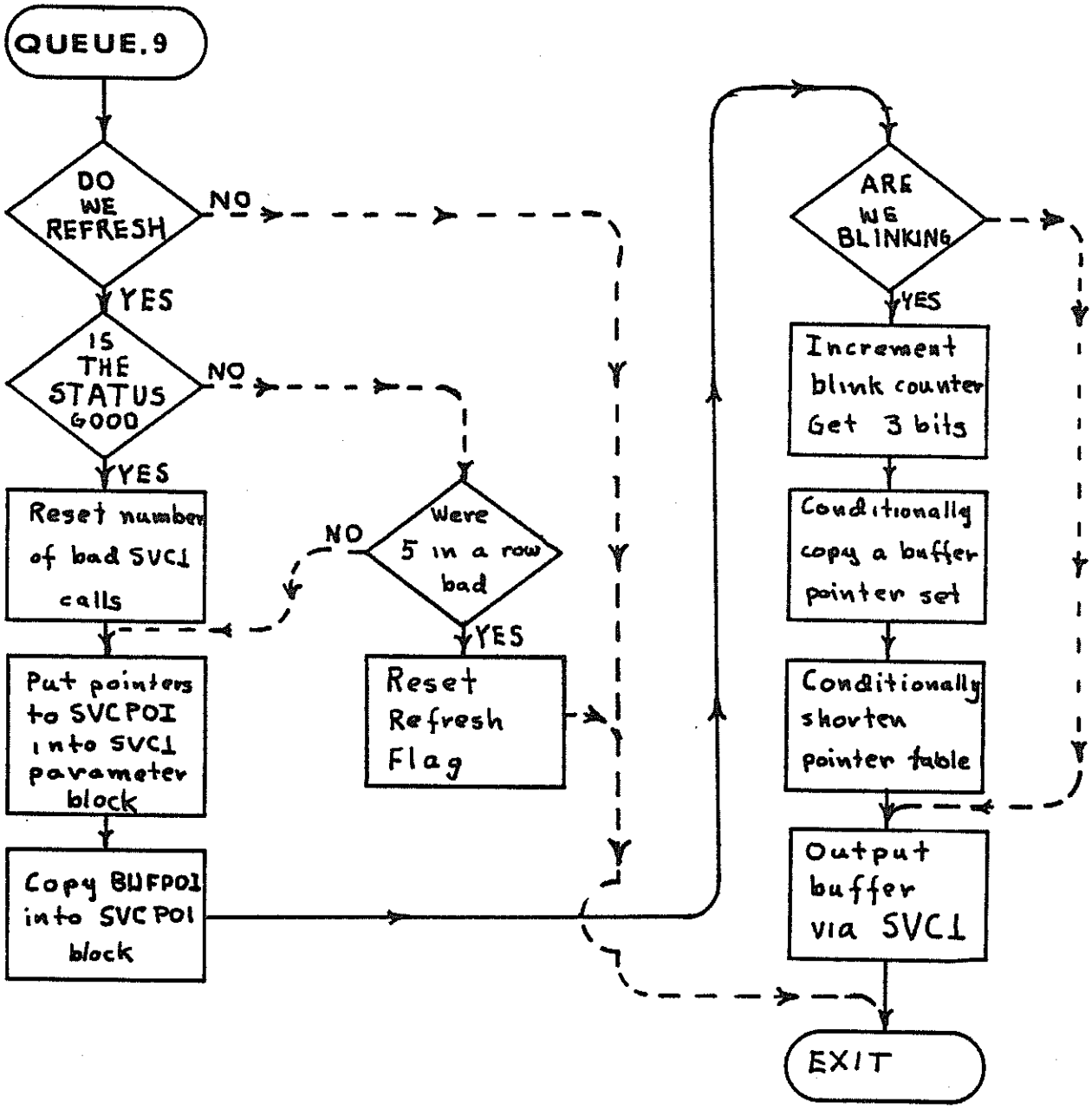


Figure 5.3. QUEUE.9

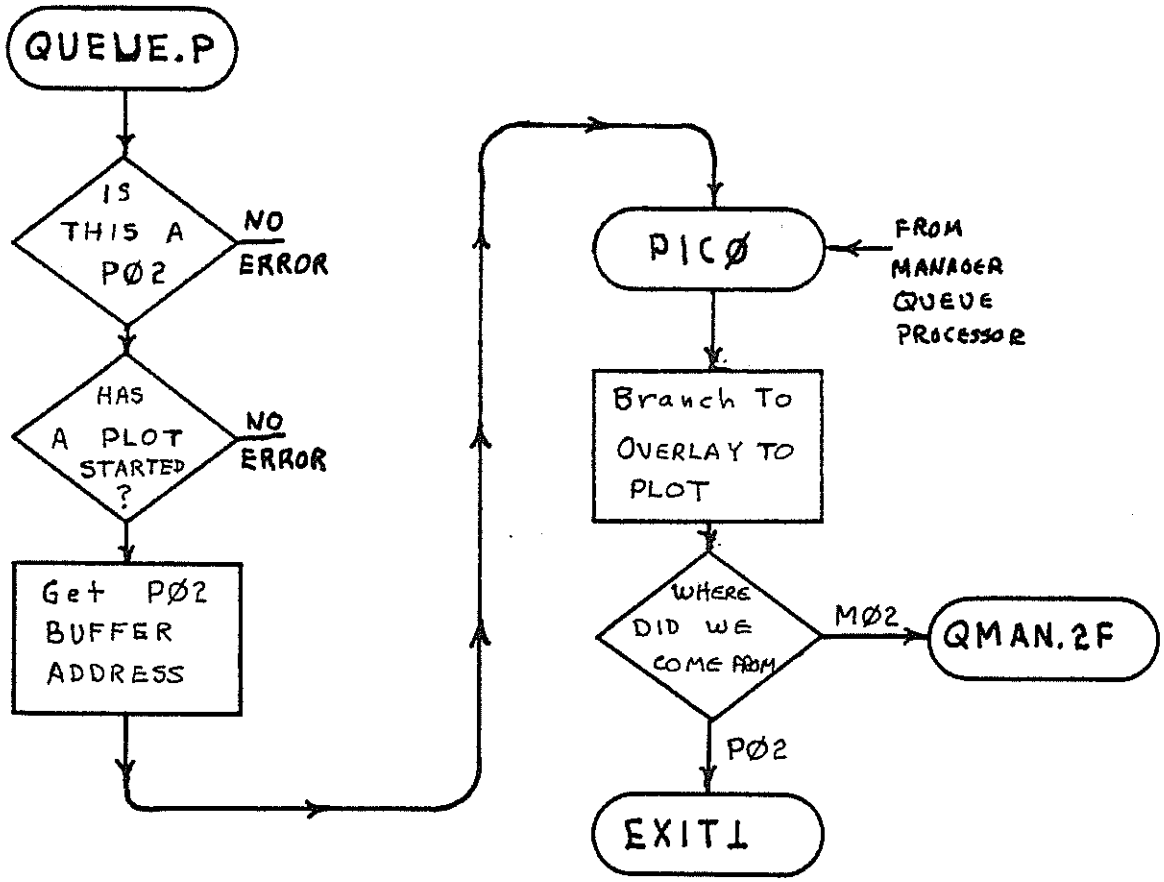


Figure 5.4. QUEUE.P

5.11. Figures

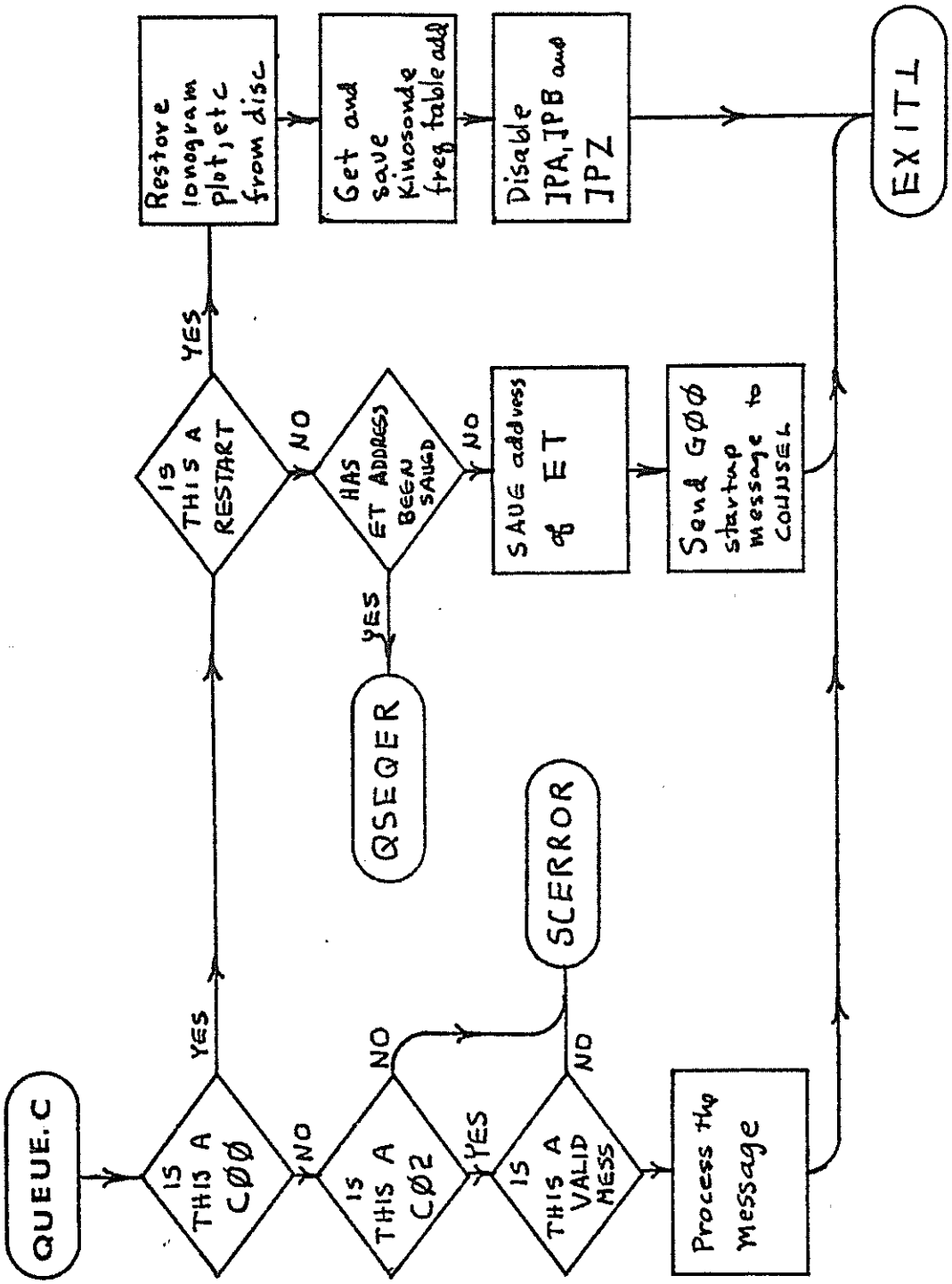


Figure 5.5. QUEUE.C

6. COUNSEL

James R. Winkelman

6.1. Introduction

COUNSEL serves four primary purposes: (1) during start-up it loads and initiates execution of all other tasks; (2) it accepts all commands from the console device and sends them to the appropriate tasks; (3) it provides debugging commands that enable the user to examine and to alter memory; (4) it will ultimately monitor various hardware and software parameters.

COUNSEL is started with a file assigned to Logical Unit 1 (LU1). This file is normally CON:. A command file from disc or TTY: may be assigned. This command file must end with the command CEND. If the GO command appears in a command file, it must be followed immediately by CEND.

COUNSEL executes a HALTIO command to CON: in response to a "ØØ" queue from a task that needs to write a message on the console.

For example:

```
SVC   2,LOG.ERR          log an error message
```

Then queue COUNSEL with the "ØØ" Q-block:

```
Ø     'ØØ','ØØ'          task letter is true Ø
                         (not ASCII)
```

The reverse order also works (see Sect. 2.10.8) except for ANALYSER, which is lower priority than COUNSEL.

Flowcharts for COUNSEL appear at the end of the chapter.

6.2. Debug Mode

The debug mode allows the user to examine memory and to alter it. COUNSEL verifies addresses and prints the old and proposed new (if any) contents in hexadecimal and both 2's-complement decimal values (that is, the value and the value minus 65536).

In addition to the commands, COUNSEL defines values for four letters, W, X, Y, and Z, for the user's convenience; these are described with the commands and the format conventions below.

PAUSE, CEND, CANHF, and COMMAND are not debug commands but are listed here because they are commands to COUNSEL.

PAUSE — Pause for 30 seconds.

CEND — End of special COUNSEL command file. Open CON: as the input device and continue.

CANHF — Cancel all tasks. A C04 is sent to the other tasks, then COUNSEL cancels itself.

COMMAND fn — Open file fn as a command file. File fn normally defines a set of parameters, such as the K-mode frequencies and ranges. The file ends with CEND or another COM. Files are not nested: the new one replaces the old.

The debug commands described below also appear in Chapter 8. Any one of these commands invokes the debug mode. Underscored letters are mandatory.

Numeric values in fields required by commands may be either hexadecimal or decimal as shown under "Field format" below.

BIAS field — Define the bias, or base value, for displacements. The specified bias becomes the current value of W until changed by another BIAS command. If there are additional fields, they are ignored.

MEMORY field1[sfield2]
Display the contents of the memory address in field1, and if a separator s and field2 appear, replace the current contents with the value of field2 by following COUNSEL's question mark with OK or YES.

After an OK or YES, the results are shown as if an ME X had been typed. A NO or anything else simply types a COUNSEL prompt.

If more than two fields appear, COUNSEL uses the last two, which enables the user to make corrections before typing the carriage return that ends the command.

The separators are the comma (,) and the virgule (/), which are interchangeable.

As soon as COUNSEL has displayed the values, X = field1, Y = field2 if present and X otherwise, and Z = the contents of address X. These may be used in subsequent fields as shown under Field format, below.

LINE field — Display 12 half-words, or 24 bytes, beginning with the address in the field.

After the command is executed, Y = X + 16, that is, it points to 8 half-words after the address in field, and Z = X + 24, or 12 half-words after the address.

Field format — The components of the fields required by the commands obey the following conventions:

NNNN	Hexadecimal number ending with the first non-hex character.
MMMM.	Decimal number which must end with a decimal point.
+	Add the next value to the field thus far.
-	Subtract the next value from the field thus far.
blank	Add or subtract according to the last sign read. A field starts as though the user had typed +Ø, so that a blank in the absence of a sign means +.
, or /	Field separators, interchangeable.
W	The value of BIAS.
X	The value of the last address used.

6.2. Debug Mode

Y — The value of the previous data field, or address (X) if there was no data field with the previous command, except after LINE when Y = X + 16.

Z — The contents of the memory location addressed by the last address field, except after LINE when Z = X + 24.

Examples — The following is a sample run with comments. D means decimal and H means hex. COUNSEL uses > as its prompt character and the colon to identify its responses.

```
*COUNSEL>BI EØØØ
*COUNSEL: EØØØ:
Command acknowledged by display of value.
*COUNSEL>ME W 5A
Blank means + by default.
*COUNSEL: EØ5A:  2EØ   736 -64800
          addr    H     D    -D
*COUNSEL>ME X / 1234
*COUNSEL: EØ5A:  2EØ   736 -64800 / 1234  4660 -60876 ?
          addr    H     D    -D  data H     D    -D
Replacement ignored:  neither OK nor YES follow.
*COUNSEL>ME 5 +2,Z
*COUNSEL:    7:    Ø     Ø -ØØØØØ /  2EØ   736 -64800 ?
Replacement specifically prevented.
*COUNSEL>NO
*COUNSEL>ME W5+2,Z
Scan of command cannot interpret W5.
*COUNSEL: SCAN ERROR
*COUNSEL>ME W 5 2,Z
*COUNSEL: EØØ7: EØØØ 57344 -08192 /    Ø     Ø -ØØØØØ ?
*COUNSEL>NO
*COUNSEL>ME W 52,Z
*COUNSEL: EØ52:    Ø     Ø -ØØØØØ / EØØØ 57344 -08192 ?
Execute replacement.
*COUNSEL>YES
*COUNSEL: EØ52: EØØØ 57344 -08192
*COUNSEL>MEM Y,12345.-345.
Decimal replacement value.
*COUNSEL: EØ52: EØØØ 57344 -08192 / 2EEØ 12ØØØ -53536 ?
*COUNSEL>Y
*COUNSEL: EØ52: 2EEØ 12ØØØ -53536
*COUNSEL>ME X/Z
*COUNSEL: EØ52: 2EEØ 12ØØØ -53536 / EØØØ 57344 -08192 ?
*COUNSEL>OK
*COUNSEL: EØ52: EØØØ 57344 -08192
```

6.3. Start up

The partitions are set and the tasks are loaded before starting COUNSEL. Logical unit 1 should be assigned to the console or a command file.

COUNSEL starts the rest of the tasks in the order SOUNDER, GRAPHER, PICKER, MANAGER, and ANALYSER. A Q-block is sent to each task after starting and each task must send a "STARTED" queue to COUNSEL before the next task is started.

The queue sent by COUNSEL is:

```
Ø  'ØØ'
1  'C'
2  status    unused
4  A(ET)     Environment Table address from SOUNDER
```

The "STARTED" queue returned by the task is:

```
Ø  'ØØ'
1  '*'       * is task initial S, G, P, M, or A
2  status    unused
4  A(UDL)    "User Dedicated Locations"
6  A(SCAN)   task scan table
```

The UDL and SCAN tables for all tasks except COUNSEL are referenced by ET.SCAN. The task must save the location of the ET table when the task is started (Sect. 7.3).

6.4. Scan Table

All console commands are read by COUNSEL, which then searches the scan tables of the tasks in the order they were started. The scan tables are in the format required by the SVC 2,17 "Mnemonic Table Scan" described in the Programmer's Reference Manual OS/16 MT2.

The task whose scan table contains a match is sent a queue:

```
Ø  'ØØ'
1  'C'
2  status    -'1ØØØ'
4  COMMAND
6  LINE
```

6.4. Scan Table

Status is returned as zero if the command was valid and >Ø if in error. The value of the error is typed on the console.

COMMAND is the number of the item in the scan table; Ø is the first, 1 the second, 2 the third, etc.

LINE is the address of the next byte in the input buffer, immediately after the command word which was recognized by the scan. The task is responsible for any further scans or data conversions from the line (see SVC 2,17 and SVC 2,15 in the reference given above).

ET.SCAN contains the address of SCAN.TAB in COUNSEL. Table 6.1

SCAN.TAB	SOUNDER UDL
+2	SOUNDER Scan Table
+4	GRAPHER UDL
+6	GRAPHER Scan Table
+8	PICKER UDL
+1Ø	PICKER Scan Table
+12	MANAGER UDL
+14	MANAGER Scan Table
+16	ANALYSER UDL
+18	ANALYSER Scan Table

Table 6.1. COUNSEL SCAN TABLE

describes the contents of the scan table that starts at SCAN.TAB.

During start up, only tasks appearing earlier in this table than the started task are known.

Table 6.1 defines both starting order and scan priority. If two tasks use the same name, the task which appears earlier in the above table is called.

6.5. Flowcharts

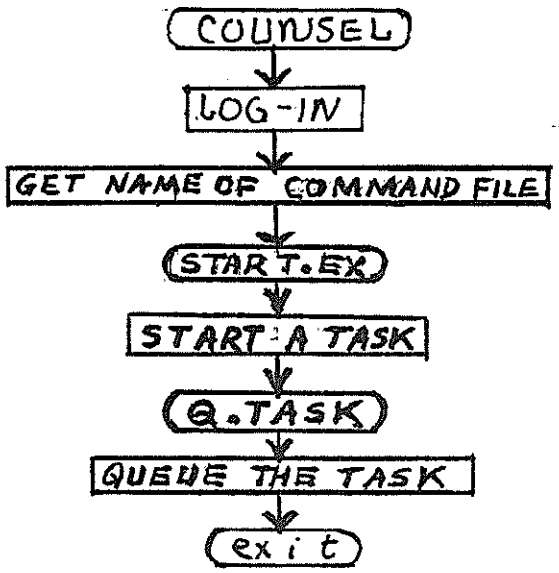


Figure 6.1
COUNSEL, START.EX, Q.TASK

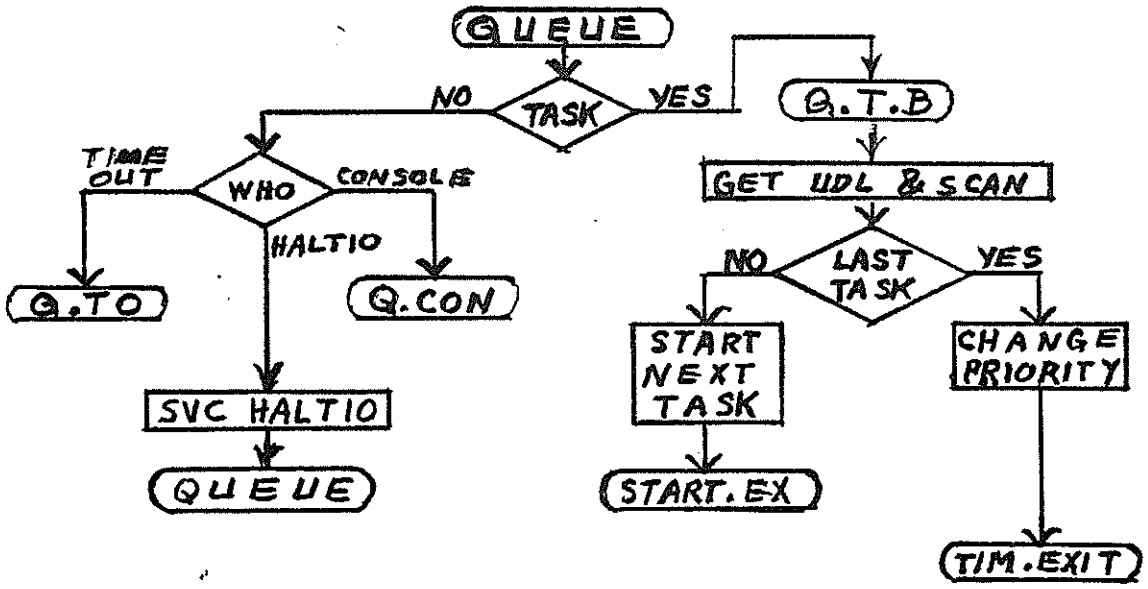


Figure 6.2. QUEUE

6.5. Flowcharts

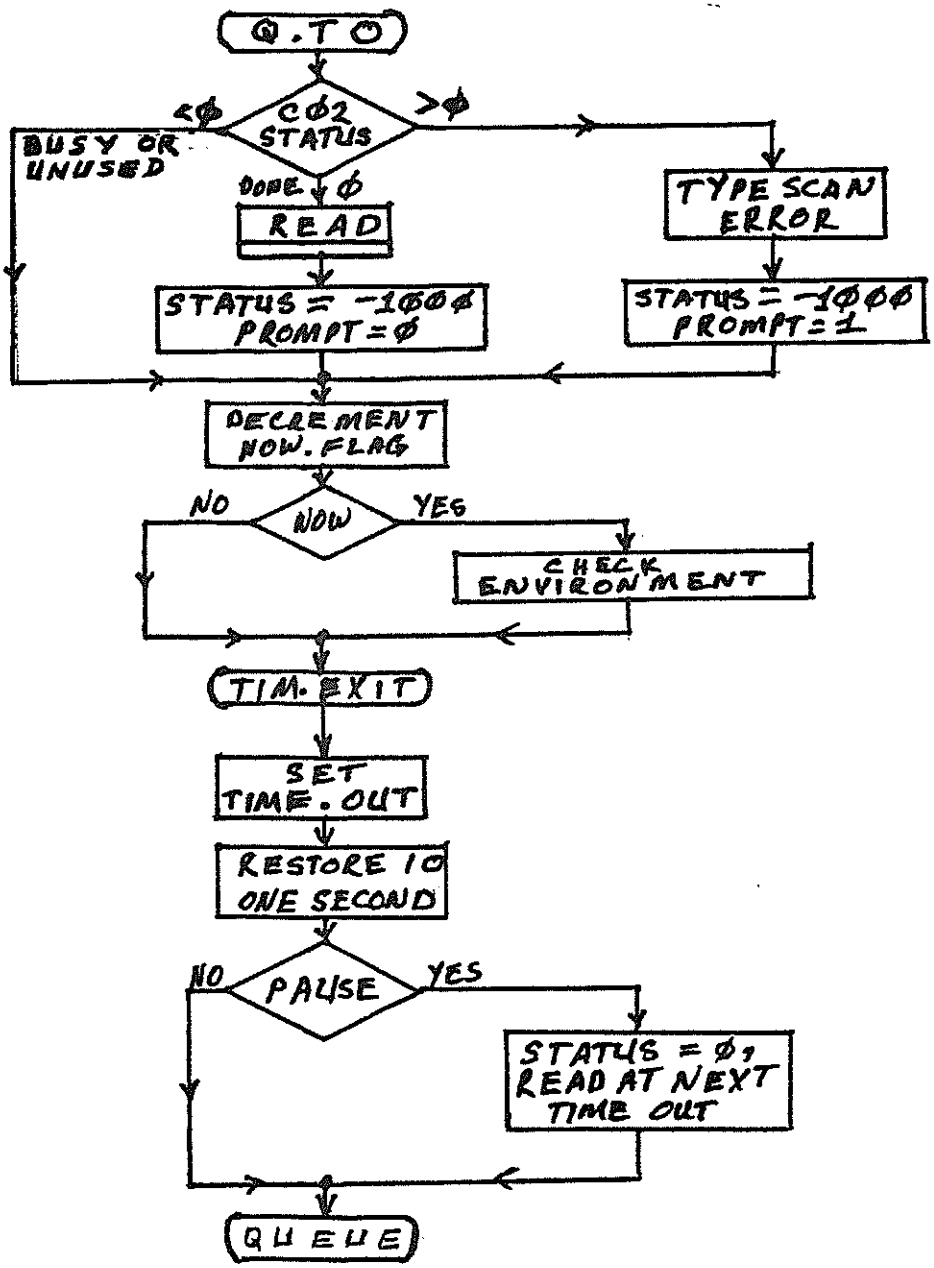


Figure 6.3. Q.TO
TIM.EXIT

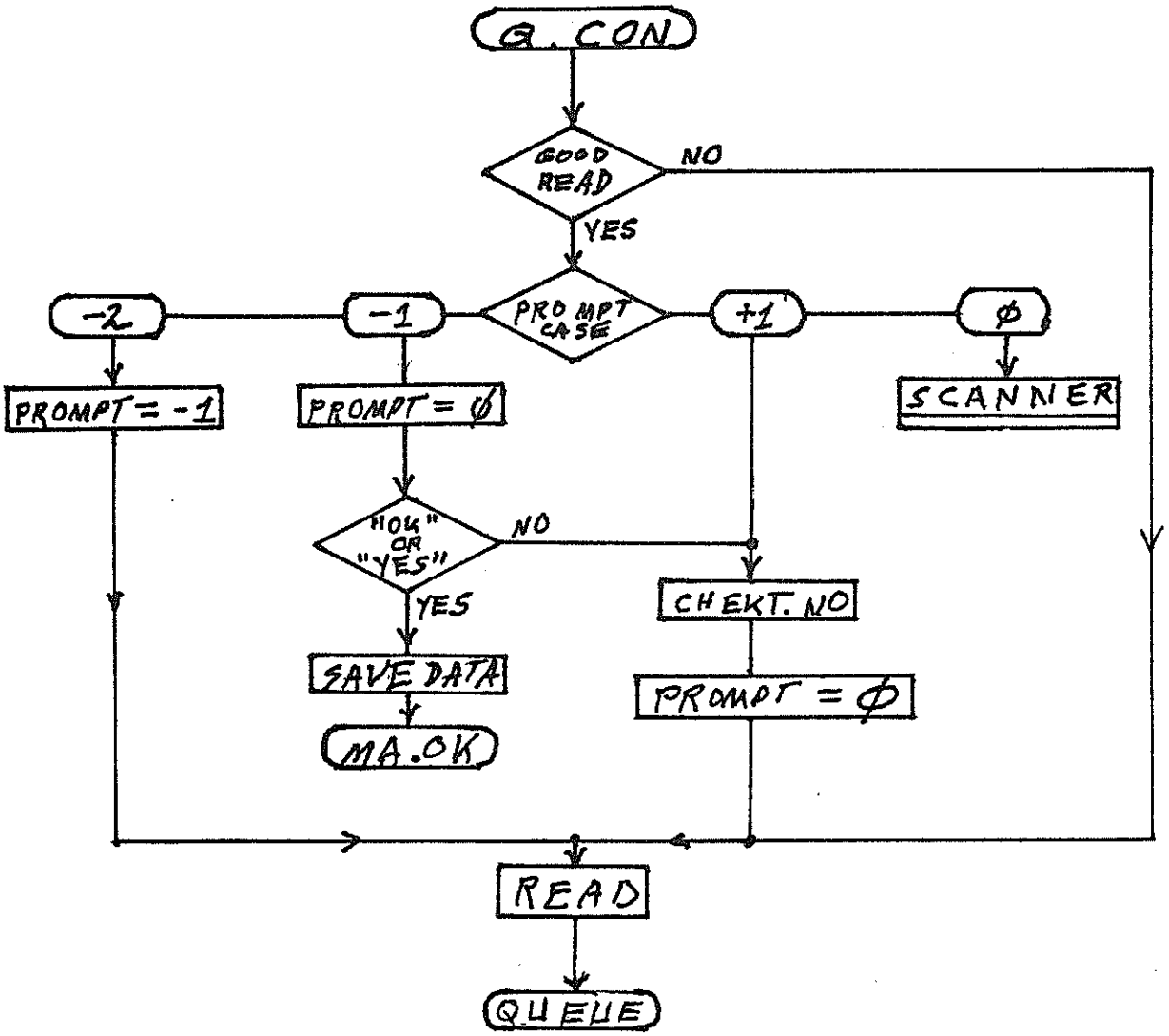


Figure 6.4. Q.CON

6.5. Flowcharts

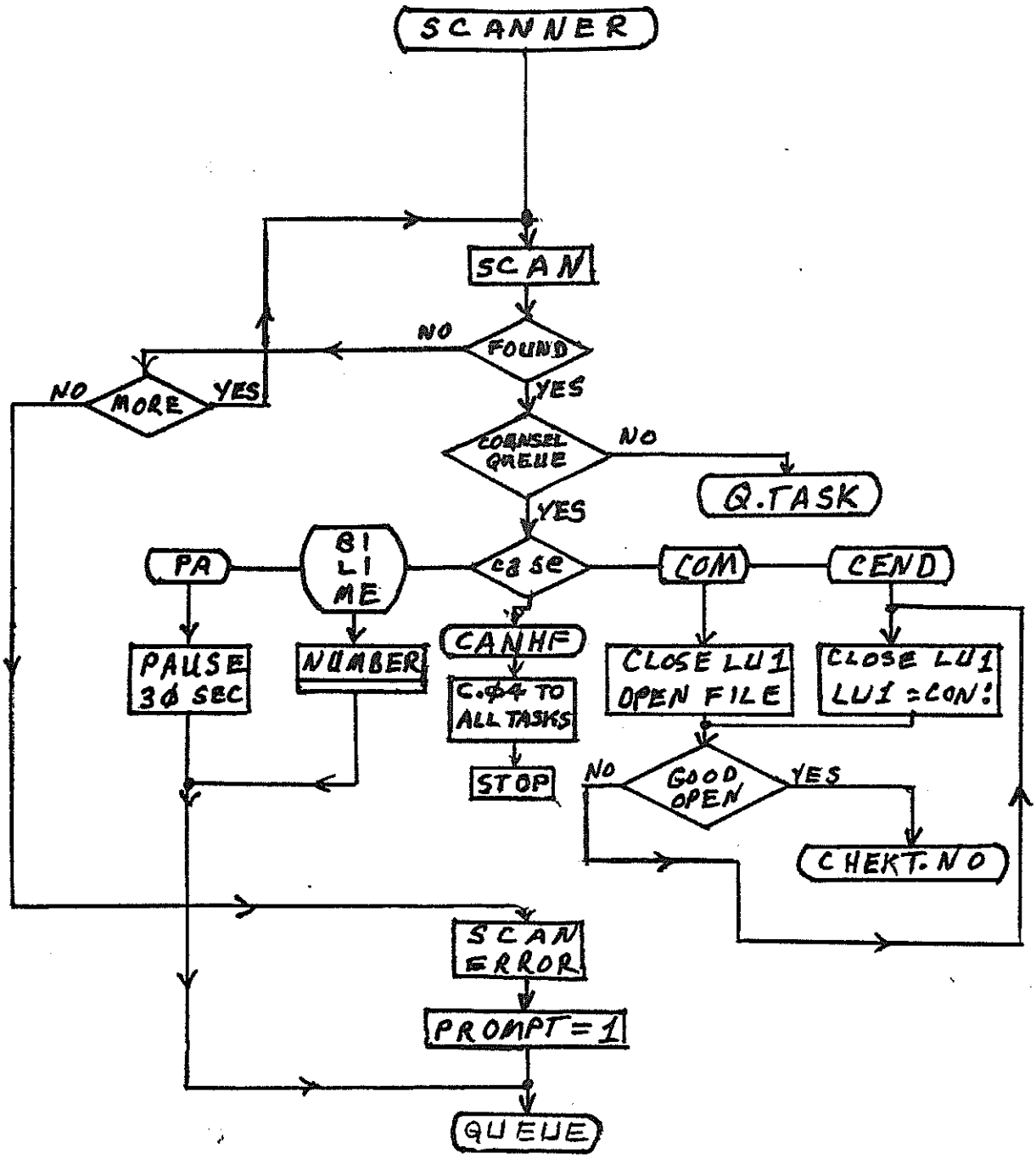


Figure 6.5. SCANNER

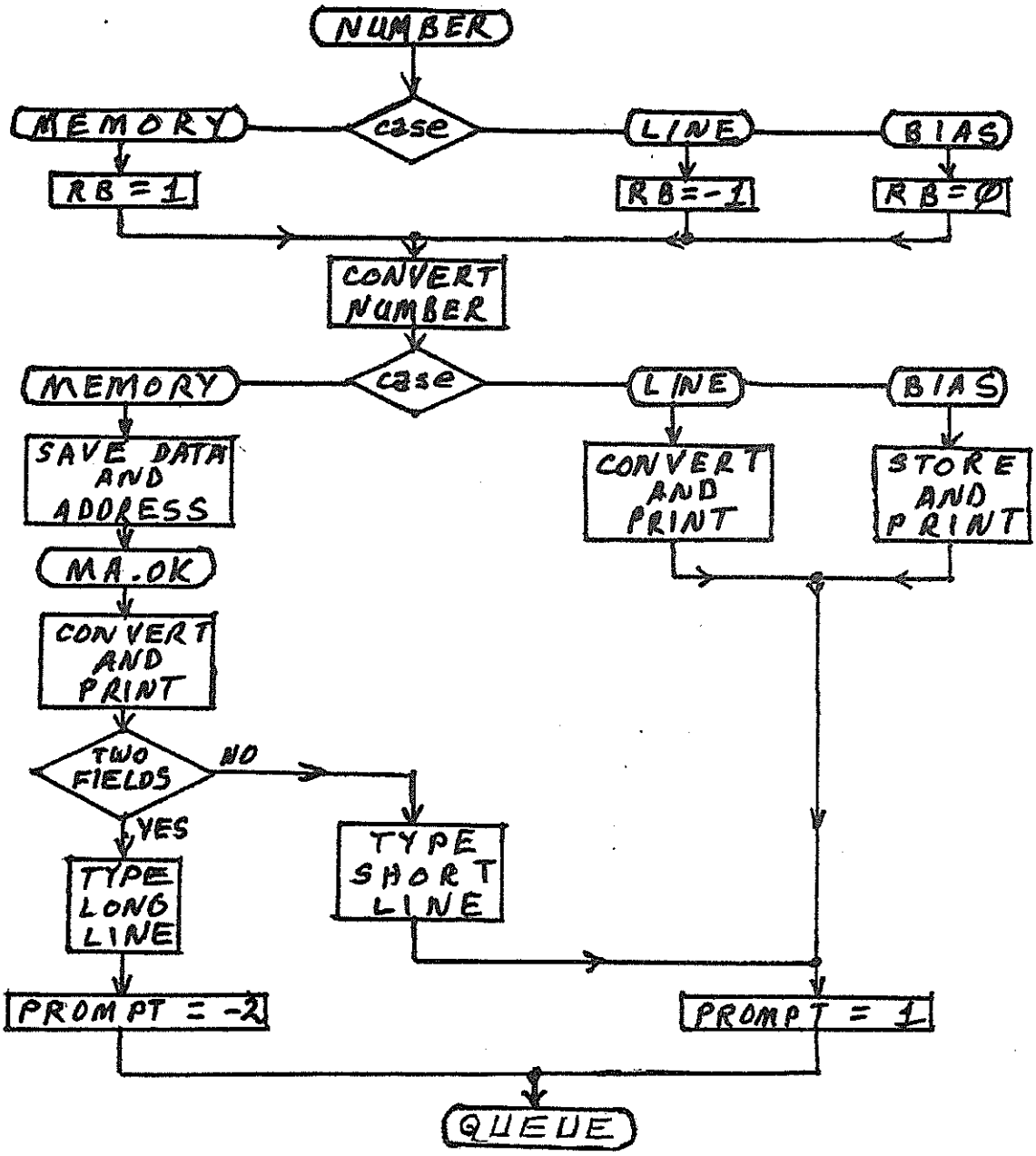


Figure 6.6. NUMBER, MA.OK

7. TABLES AND Q-BLOCKS

David C. Walden

This chapter provides detailed descriptions of the global tables and the Q-blocks used by Product One.

In the tabular descriptions of the ICT and of the other tables in the first three sections, the displacement is the symbolic word location used in the software to address each entry in the table. In Sections 7.1 and 7.3, the displacements are in alphabetic order instead of their order in the Interdata computer. The descriptions include references to task descriptions that discuss given entries.

Preset values come from code assembly and computation during SOUNDER start-up. Assembly code parameters used to preset table entries, table lengths, etc., can be integers, text, or expressions involving other pre-defined values. Groups of pre-defined values are made available at assembly time by invoking the COPY directive to the Assembler from the assembly source code. The argument of the COPY directive must be the label of a parameter group in file SYSC:PCB.CAL (see the CAL User's Manual). The HF Radar software uses this facility to define the structures and lengths of the tables described in this section, and to define the local site constants used to preset certain values in the ICT (Sect. 7.1). The label of the local site constants in file PCB.CAL is **LOCAL. LOCAL must be edited to establish the site location and the geometry of the antenna layout in the ICT. This information is saved with the data by MANAGER. LOCAL parameter definition labels are all prefixed with "LOC."; LOCAL is further described in Section 7.5.

All commands referred to below appear in alphabetic order in Chapter 8.

7.1. Ionogram (sounding) Configuration Table (ICT)

The order of displacements in Interdata memory for the ICT is given in Section 4.3.

DISPLACEMENT	BYTES	DESCRIPTION
ICT.1XØ	2	Receiver channel 1 XØ offset from the FEP (see Sect. 3.4).
ICT.1YØ	2	Receiver channel 1 YØ offset, as above.

7.1. Ionogram (sounding) Configuration Table (ICT)

DISPLACEMENT	BYTES	DESCRIPTION
ICT.2XØ	2	Receiver channel 2 XØ offset, as above.
ICT.2YØ	2	Receiver channel 2 YØ offset, as above.
ICT.ADLY	2	Antenna/receiver range Ø correction preset to LOC.ADLY (Sect. 7.5).
ICT.C1A1	6	Channel 1, antenna 1 X-Y-Z coordinates preset to LOC.X11, LOC.Y11, and LOC.Z11.
ICT.C1A2	6	Channel 1, antenna 2 X-Y-Z preset to LOC.X12, LOC.Y12, LOC.Z12.
ICT.C1A3	6	Channel 1, antenna 3 X-Y-Z from LOC.X13, LOC.Y13, LOC.Z13.
ICT.C1A4	6	Channel 1, antenna 4 X-Y-Z from LOC.X14, LOC.Y14, LOC.Z14.
ICT.C1OR	4	Channel 1 antenna orientations preset to LOC.O11, LOC.O12, LOC.O13, LOC.O14.
ICT.C2A1	6	Channel 2, antenna 1 X-Y-Z from LOC.X21, LOC.Y21, LOC.Z21.
ICT.C2A2	6	Channel 2, antenna 2 X-Y-Z from LOC.X22, LOC.Y22, LOC.Z22.
ICT.C2A3	6	Channel 2, antenna 3 X-Y-Z from LOC.X23, LOC.Y23, LOC.Z23.
ICT.C2A4	6	Channel 2, antenna 4 X-Y-Z from LOC.X24, LOC.Y24, LOC.Z24.
ICT.C2OR	4	Channel 2 antenna orientations preset to LOC.O21, LOC.O22, LOC.O23, LOC.O24.
ICT.DELF	1	Delta frequency (kHz) set by command DELF.
ICT.DESC	8	Experiment description, up to 8 ASCII characters, set by command DESC.
ICT.DFIX	2	Frequency index increment set by command DFIX.
ICT.DGA	1	DIO generation algorithm index. Currently 1 is the only value available.
ICT.ECHO	1	Output data merit ratio set by task PICKER.
ICT.EFIX	2	Ending frequency index derived like ICT.SFIX below.
ICT.EFRD	4	Default end frequency set by command EFRD.
ICT.EFRQ	4	Actual end frequency set by command EFRQ.
ICT.EPF	2	Echoes per frequency set by command EPF.
ICT.ETS	4	Sounding end time in seconds from 1/1/77, computed from the start time and the number of pulses per sounding.
ICT.FEPC	1	FEP command byte (program start address) found by the TSG program specified by ICT.TSGC for use by the FEP. ICT.TSGC and ICT.FEPC are consecutive in the table and are immediately followed by 3Ø bytes for 15 more TSG/FEP pairs.

7.1. Ionogram (sounding) Configuration Table (ICT)

DISPLACEMENT	BYTES	DESCRIPTION
ICT.FEPM	3	FEP memory disc file extension set by command BOOT FEP[,EXT].
ICT.FEPT	1	FEP thresholding factor (x10) set by command FEPTH.
ICT.FGA	1	Frequency generation algorithm index, like ICT.DGA above.
ICT.HPE	2	Halfwords per echo set by command HPE.
ICT.INUM	2	Sounding number started at 1 by system restart or by command NI.
ICT.LAT	4	Local latitude, preset to LOC.LAT1 and LOC.LAT2 (Sect. 7.5), positive North.
ICT.LEN	1	Length of ICT, bytes, preset.
ICT.LONG	4	Local longitude, preset to LOC.LNG1 and LOC.LNG2 (Sect. 7.5), positive East.
ICT.LTIM	1	Local time meridian in tenths of hours, preset to LOC.LTIM (Sect. 7.5), positive East.
ICT.M	1	Select M, set by command KEEP, out of N pulses, where N is in ICT.N, below.
ICT.MODE	1	Sounding mode (hex), 0, I - mode; 1 - 2, K-mode; 3, B-mode; 4 - 7, P-modes; 8, R-mode; 40, TSG; 41, FEP; 42, remarks; 50 - 5F, 10-microsecond; 60 - 6F, HKADC; all others currently undefined.
ICT.MRDP	1	Number of HKADC multiplexer channels to read per sounding pulse, preset.
ICT.N	1	Number of pulses per frequency computed from channel selection (see command KEEP). Refer to ICT.M, above.
ICT.NI	2	Number of soundings to run set by command NI.
ICT.NIPH	1	Number of soundings per hour set by command NIPH.
ICT.NKFR	1	Number of frequencies used in kinesonde-mode sounding.
ICT.OPOB	2	Offset to height-coincident data in buffer with its address in the P02 Q-block (see Sect. 3.3).
ICT.REVA	2	ANALYSER task revision set by ANALYSER.
ICT.REVC	2	COUNSEL task revision set by COUNSEL.
ICT.REVG	2	GRAPHER task revision set by GRAPHER.
ICT.REVM	2	MANAGER task revision set by MANAGER.
ICT.REVP	2	PICKER task revision set by PICKER.
ICT.REVS	2	SOUNDER task revision set by SOUNDER.
ICT.SFIX	2	Starting frequency index derived from the start frequency and the protected frequencies map.
ICT.SFRD	4	Default start frequency set by command SFRD.
ICT.SFRQ	4	Actual start frequency set by command SFRQ.
ICT.SITE	4	Sounder site code, preset to LOC.SIT1 and LOC.SIT2 (Sect. 7.5).

7.1. Ionogram (sounding) Configuration Table (ICT)

DISPLACEMENT	BYTES	DESCRIPTION
ICT.STAT	1	General system status byte. Bit Ø (most significant bit): Ø = high-power transmitter high voltage off. 1 = high-power transmitter high voltage on. Bits 1 - 7: to be determined.
ICT.STM	2	Sounding start time, milliseconds part, always Ø.
ICT.STS	4	Sounding start time in seconds from 1/1/77 derived from the start time defined by commands GO, KRUN, or VUF.
ICT.TDLY	2	TSG range Ø correction set by command TSGC.
ICT.TSGC	1	TSG command byte set by command TSGC.
ICT.TSGM	3	TSG memory disc file extension set by command BOOT TSG[,EXT].

7.2. Pulse Configuration Table (PCT)

DISPLACE-MENT	BYTES	BITS	BIT COUNT	DESCRIPTION
PCT.TIK	2	Ø-15	16	Count of TSG DONE interrupts since start of sounding.
PCT.DIO1	2	Ø-1	2	Bandwidth command, channel 1.
		2-3	2	Bandwidth command, channel 2. The bandwidth commands correspond to frequency as follows: ØØ = 1 kHz Ø1 = 3 kHz 1Ø = 1Ø kHz 11 = 3Ø kHz
		4-15	12	Frequency index described under EFRD.
PCT.DIO2	2	Ø-4	5	Transmitter attenuation command, high- or low-power value from TXATH or TXATL depending on whether the high power is on or off. To obtain the attenuation, invert the command by subtracting it from 31 decimal or taking the XOR (exclusive OR) with 1F hex. The attenuation is in the range Ø through 31 dB in steps of 1 dB.
		5-7	3	Delta-frequency flag: see table with command PSEQ in Chapter 8. The first two flags are used by the current default algorithm.

DISPLACEMENT	BYTES	BITS	BIT COUNT	DESCRIPTION
		8-11	4	Calibration attenuation command. To obtain the attenuation, invert the command by subtracting it from 15 decimal or taking the XOR with F hex, and multiply by 8. The attenuation ranges from Ø through 12Ø dB in 8-dB steps and may be set by CALA.
		12-15	4	ICT.TSGC/ICT.FEPC index, currently unused and set to Ø.
PCT.DIO3	2	Ø-3	4	Antenna command, channel 1.
		4-7	4	Antenna command, channel 2. Add 1 to the command to get the antenna number set by ANT. Although only 4 antennas are now in use, this command provides for as many as 16.
		8-11	4	Octave filter command, channel 1.
		12-15	4	Octave filter command, channel 2. The octave filter, or RF band selector, commands are not implemented and are always set to Ø.
PCT.DIO4	2	Ø-3	4	RF attenuation command, channel 1.
		4-7	4	RF attenuation command, channel 2.
		8-11	4	IF attenuation command, channel 1.
		12-15	4	IF attenuation command, channel 2. To obtain the attenuation from any of the above four commands, invert the command by subtracting it from 15 decimal or taking the XOR with F hex, and multiply by 4. These attenuations lie in the range Ø through 6Ø dB in 4-dB steps.

The DIOn information comes from the DIO table, Sect. 2.12.8.

7.3. Environment Table (ET)

The five routines with addresses in ET are in SOUNDER (Sect. 2.11) but are available to other tasks including user-written ANALYSER or TEST tasks.

DISPLACEMENT	BYTES	DESCRIPTION
ET.ABUF	2	Address of shared EKO buffers area. Between soundings, part of the EKO buffers area is available to other tasks, including user-written versions of ANALYSER. The length of the shared area is in ET.LBUF, below.
ET.AFRTB	2	A(FTOBCD), the address of routine FTOBCD to convert hex frequency into BCD frequency.
ET.AFRTN	2	A(FRTON), the address of routine FRTON to convert frequency into frequency index.
ET.AICT	2	Address of ICT.
ET.AKIN	2	Address of kinesonde-mode table in SOUNDER memory.
ET.ALOGF	2	A(LOG), the address of routine LOG to compute the common or natural log.
ET.ANTFR	2	A(NTOFR), the address of routine NTOFR to convert frequency index into frequency.
ET.APAMP	2	A(PHAMP), the address of routine PHAMP to compute phase and amplitude.
ET.CDN	2	Countdown value, ≥Ø or -1.
ET.HKACD	128	Table of HKADC readings.
ET.LBUF	2	Length of shared EKO buffers area with address in ET.ABUF, above.
ET.LOGD	4	Mnemonic name of logging device, preset to CON:.
ET.SCAN	2	Address of UDL/SCAN list for sending commands under program control. See Section 6.3.
ET.SEC	4	Time-of-day in seconds past 1/1/77.
ET.TTNI	2	Time to next sounding (seconds).

7.4. Kinesonde Mode Table (KMODE)

DISPLACEMENT	BYTES	DESCRIPTION
+Ø	2	Number of entries in table (1Ø).
+2	6	K-mode frequency Ø entry.
+8	6	1st entry.
+14	6	2nd.
+2Ø	6	3rd.
+26	6	4th.
+32	6	5th.
+38	6	6th.
+44	6	7th.
+5Ø	6	8th.
+56	6	9th.
+62	2	End-of-table flag (Ø).

Each entry of 6 bytes contains the following:

DISPLACEMENT	BYTES	DESCRIPTION
+Ø	2	Frequency index set by command F, or -1 if not specified.
+2	2	Minimum range set by command RN. Default is 5Ø km (334 microseconds).
+4	2	Maximum range set by command RF. Default is 5ØØ km (3334 microseconds).

Ranges are set and displayed in kilometers, but are stored in microseconds.

7.5. Local Site Constants (LOCAL)

The HF Radar software parameters and data identifiers which are site-dependent reside in disc file SYSC:PCB.CAL and begin at block label **LOCAL. SOUNDER uses the definitions from **LOCAL to preset a number of values in the ICT (Sect. 7.1). The ICT in turn is saved with the sounding data by MANAGER. The responsibility for updating **LOCAL to correct descriptions of each site must be taken by someone at the site who is familiar with Interdata's editor, assembler, and task establisher. The following is required:

EDIT **LOCAL values in SYSC:PCB.CAL
Assemble SOUNDER's tables (CAL SNDRTBLS)
Establish the SOUNDER task (TETSNDR SOUNDER,bias,6Ø)

7.5. Local Site Constants (LOCAL)

See Section 9.4 (CSS files) and the EDIT manual for further details.

The table that follows gives the name (as opposed to displacement) and description of each entry. This table enables the user to set and change site parameters without changing SOUNDER, though SOUNDER must be re-assembled.

NAME	DESCRIPTION
LOC.LAT1	Latitude, degrees, positive North (Boulder is at 40).
LOC.LAT2	Latitude, fractional degrees.
LOC.LNG1	Longitude, degrees, positive East (Boulder is at -105).
LOC.SIT1	Site code (bytes 0-1).
LOC.SIT2	Site code (bytes 2-3)
LOC.ADLY	Antenna/receiver range 0 correction, microseconds
LOC.LTIM	Local time meridian, tenths of hours, positive East (Boulder is at -70).
LOC.X11	Channel 1, antenna 1 X coordinate, plus or minus meters/8.
LOC.Y11	Channel 1, antenna 1 Y.
LOC.Z11	Chan. 1, ant. 1 Z.
LOC.X12	Chan. 1, ant. 2 X.
LOC.Y12	Chan. 1, ant. 2 Y.
LOC.Z12	Chan. 1, ant. 2 Z.
LOC.X13	Chan. 1, ant. 3 X.
LOC.Y13	Chan. 1, ant. 3 Y.
LOC.Z13	Chan. 1, ant. 3 Z.
LOC.X14	Chan. 1, ant. 4 X.
LOC.Y14	Chan. 1, ant. 4 Y.
LOC.Z14	Chan. 1, ant. 4 Z.
LOC.X21	Chan. 2, ant. 1 X.
LOC.Y21	Chan. 2, ant. 1 Y.
LOC.Z21	Chan. 2, ant. 1 Z.
LOC.X22	Chan. 2, ant. 2 X.
LOC.Y22	Chan. 2, ant. 2 Y.
LOC.Z22	Chan. 2, ant. 2 Z.
LOC.X23	Chan. 2, ant. 3 X.
LOC.Y23	Chan. 2, ant. 3 Y.
LOC.Z23	Chan. 2, ant. 3 Z.
LOC.X24	Chan. 2, ant. 4 X.
LOC.Y24	Chan. 2, ant. 4 Y.
LOC.Z24	Chan. 2, ant. 4 Z.
LOC.O11	Chan. 1, ant. 1 orientation in 1/256ths of 360 degrees.
LOC.O12	Chan. 1, ant. 2 orientation.
LOC.O13	Chan. 1, ant. 3.
LOC.O14	Chan. 1, ant. 4.

NAME	DESCRIPTION
LOC.021	Chan. 2, ant. 1.
LOC.022	Chan. 2, ant. 2.
LOC.023	Chan. 2, ant. 3.
LOC.024	Chan. 2, ant. 4.

7.6. HF Radar Software Parameter Definitions (SNDR)

To make them universally available at assembly time, commonly used parameters are included in file SYSC:PCB.CAL at block label **SNDR. All of SOUNDER's modules, PICKER, MANAGER, and a number of the TEST tasks are assembled with the COPY SNDR directive in their code. SNDR may be modified as may LOCAL (Sect. 7.5), but changes must be made circumspectly.

In SNDR, labels are equated to numerical values with the EQU directive. They are listed below and grouped as they are in SNDR.

SOUNDER pipeline register definitions (see Sect. 2.7.2):

NAME	VALUE	DESCRIPTION
PULSE	5	(R5) TSG ticks from sounding start
MODE	6	(R6) sounding mode
P.TSG	7	(R7) PULSE at TSG level
P.FEP	8	(R8) PULSE at FEP level
P.EKO	9	(R9) PULSE at EKO level
GO	10	(R10) sounding condition
PTYPE	11	(R11) pulse type

Device and DIO addresses (all values are hexadecimal):

NAME	VALUE	DESCRIPTION
TSGPIA	24	TSG PIA (data) DIO address
TSGEIA	25	TSG EIA (flags) DIO address
TSGCLK	26	TSG clock DIO address
HKSEL	33	Housekeeping ADC (HKADC) channel select DIO address
HKREAD	34	HKADC data DIO address
FEP1	36	FEP data write DIO address
FEP2	37	FEP control byte write DIO address
ULM	37	DIO interface device address
FEP	77	FEP/EKO interface device address

7.6. HF Radar Software Parameter Definitions (SNDR)

Miscellaneous definitions:

NAME	VALUE	DESCRIPTION
LOCAL	10(hex)	DIO local control bit mask
EKOSIZE	2048	maximum bytes in EKO transfer from FEP
NEKOBFS	6	number of PCT/EKO buffers
NRGBD	12	number of receiver frequency bands for AGC
NPTYPE	4	number of pulse types (pulses/pulse set)
Q	4	Q-other-task SVC call number
NKFRQ	10	number of KMODE table entries allowed
LOIX	274	lowest synthesizer frequency index
HIIX	3561	highest synthesizer frequency index
LOFR	100	lowest synthesizer frequency in kHz
HIFR	30000	highest synthesizer frequency in kHz
MAPSZ	(HIIX - LOIX + 1)/8 + 4	number of bytes in protected frequencies map
P.MAXEKO	8	number of echoes in PICKER's output buffer
P.NECHO	22	number of echoes in PICKER's internal table
P.MICRO	60	delta in microseconds for range coincidence
PIK.TEST	1	non-zero to include PICKER testing code
S.LENG	21	number of soundings MANAGER will save on disc
S.ANAL	1	zero if there is no ANALYSER

7.7. Q-blocks

This section contains descriptions of the Q-blocks used for inter-task communications in Product One. In general the first four bytes of a Q-block are in the standard format described in Section 1.7; exceptions are noted below. Some Q-blocks contain only these first 4 bytes, while others contain from 1 to 3 additional information fields of 2 bytes each. There is no standard format for the additional information fields.

Individual Q-blocks are described in Sections 7.7.1 through 7.7.6, grouped by the task that originates or sends them. Section 7.7.7 summarizes the Q-blocks by recipient.

7.7.1. COUNSEL Q-blocks

C00 C00 + 0: 0.
C00 + 1: "C".
C00 + 2: status.
C00 + 4: address of Environment Table (ET) from SOUNDER.

Destination: the C00 Q-block is first sent to SOUNDER, then in turn to PICKER, MANAGER, GRAPHER, and ANALYSER.

Purpose: this is the first Q-block each task receives after being started by COUNSEL. It is the signal to start task execution and is also used to promulgate global information.

Recipient action: SOUNDER puts the address of the ET in C00 + 4. The other tasks, including COUNSEL, subsequently pick up A(ET), and thus the addresses of the global tables and of the reentrant subroutines (Sect. 2.11) are known. Each recipient task clears the C00 status, initialized to -1000 (hex), when it is through with the C00 Q-block.

C02 C02 + 0: 2.
C02 + 1: "C".
C02 + 2: status.
C02 + 4: task command index, number of item in task's SCAN (0 first, 1 second, etc.). See Sect. 2.8.4.
C02 + 6: command line address after scan.

Destination: the C02 Q-block is sent to the task whose command mnemonic scan table contains a match of the first word of the command typed at the console, unless the command is directed to COUNSEL.

Purpose: the C02 Q-block notifies a task that a console command has been directed to it. The command index indicates which command has been given; C02 + 6 contains the byte address, in COUNSEL's console input buffer, of the command argument list, if any. (See Sect. 2.8.4.6.)

Recipient action: the queued task attempts to act on the advice given, and, if successful, clears the C02 status to 0. If the advice was not appropriately given, the status is set to a positive value. So long as the status is left in its initial state, -1000 (hex), COUNSEL will not initiate another command read from the console.

7.7.1. COUNSEL Q-blocks

C04 C04 + 0: 4.
C04 + 1: "C".
C04 + 2: status.

Destination: COUNSEL sends the C04 Q-block to all other tasks when command CANHF is given.

Purpose: the C04 Q-block signals that an orderly shutdown of the HF Radar software has been requested.

Recipient action: each task gets its own house in order and then terminates itself. The status-clearing convention is not observed because COUNSEL also terminates itself.

7.7.2. SOUNDER Q-blocks

S00 S00 + 0: 0.
S00 + 1: "S".
S00 + 2: status.
S00 + 4: address of SOUNDER's User Dedicated Locations (UDL).
S00 + 6: address of SOUNDER's command mnemonic scan table.

Destination: COUNSEL.

Purpose: S00 is sent to inform COUNSEL that SOUNDER has successfully started and to give COUNSEL the addresses of SOUNDER's UDL and scan table. (See Sect. 6.3 and ET.SCAN, Sect. 7.3.)

Recipient action: COUNSEL copies S00 + 4 and S00 + 6 to its UDL/SCAN table, then starts PICKER. The status is ignored.

S02 S02 + 0: 2.
S02 + 1: "S".
S02 + 2: status.
S02 + 4: EKO buffer address.
S02 + 6: address of associated PCT (Sect. 7.2).
S02 + 8: KMODE table entry address (Sect. 7.4), or 0.

Destination: PICKER (or TEST task, Sect. 7.7.7).

Purpose: S02 is sent to inform PICKER of the PCT/EKO buffer addresses associated with the latest data from the FEP. If the sounding is K-mode, S02 + 8 points PICKER to the range limits, or is 0 if the K-mode echo is lost.

Recipient action: PICKER copies the PCT to its memory, then scans the EKO buffer for range coincident echoes. When through with the EKO buffer, PICKER clears the SØ2 status to Ø. (See Sect. 10.3 for TEST task action.)

SØ4 SØ4 + Ø: 4.
SØ4 + 1: "S".
SØ4 + 2: status.
SØ4 + 4: sounding mode.
Ø = GO (I-mode sounding).
2 = KRUN (K-mode sounding).
4 = VUF (K-mode single-frequency test sounding).

Destination: PICKER (or TEST task, Sect. 7.7.7).

Purpose: SØ4 is sent at the beginning of the countdown to signal the start of a sounding.

Recipient action: PICKER initializes for the sounding and passes the SØ4 Q-block to MANAGER. MANAGER in turn passes it to GRAPHER and to the ANALYSER task, if any (Sect. 4.11). When all pre-sounding action is complete, the SØ4 status is cleared to Ø. (See Sect. 10.3 for TEST task action.)

SØ6 SØ6 + Ø: 6.
SØ6 + 1: "S".
SØ6 + 2: status.

Destination: PICKER (or TEST task, Sect. 7.7.7).

Purpose: SØ6 signals the end of a sounding.

Recipient action: PICKER packs its final output buffer, if any, then passes the SØ6 Q-block to MANAGER. MANAGER in turn passes it to GRAPHER and to the ANALYSER task, if it is running. When all post-sounding action is complete, the SØ6 status is cleared to Ø. (See Sect. 10.3 for TEST task action.)

SØ8 SØ8 + Ø: 8.
SØ8 + 1: "S".
SØ8 + 2: status.
SØ8 + 4: K-mode frequency number (KMODE table entry number, Sect. 7.4).

7.7.2. SOUNDER Q-blocks

Destination: GRAPHER.

Purpose: SØ8 informs GRAPHER of a change in a K-mode frequency or range limit.

Recipient action: GRAPHER displays the K-mode frequnecy markers on the previous ionogram plot. The status is ignored.

SFF
- SFF + Ø: FF (hex).
- SFF + 1: "S".
- SFF + 2: status.
- SFF + 4: address of ERROR call in SOUNDER.
- SFF + 6: error code.

Destination: the HELP TEST task (Sect. 10.3).

Purpose: SFF passes SOUNDER error data to HELP for diagnosis.

Recipient action: see ERROR, Sect. 2.10.7.

In addition to the Q-blocks described above, SOUNDER sends the following:

CØØ to GRAPHER after terminating a TEST task (see ENDTEST, Sect. 2.13.8).

A(TSTLST) to TEST tasks (see OVRUNT, Sect. 2.13.8, and TSTLST, Sect. 2.13.8.2).

ØØ to COUNSEL for temporary access to the console (see WRLOG, Sect. 2.10.8).

7.7.3. PICKER Q-blocks

PØØ
- PØØ + Ø: Ø.
- PØØ + 1: "P".
- PØØ + 2: status.
- PØØ + 4: address of PICKER's UDL.
- PØØ + 6: address of PICKER's command mnemonic scan table.

Destination: COUNSEL.

Purpose: see SØØ, Section 7.7.2.

Recipient action: COUNSEL copies PØØ + 4 and PØØ + 6 to its UDL/SCAN table, then starts MANAGER.

PØ2 PØ2 + Ø: 2.
PØ2 + 1: "P".
PØ2 + 2: status.
PØ2 + 4: address of PICKER output buffer (sounding data record).

Destination: MANAGER.

Purpose: PØ2 gives MANAGER the address of the picked data for a pulse set and the associated PCT's.

Recipient action: MANAGER passes the PØ2 to GRAPHER and to ANALYSER, if it is running, then saves the PØ2 buffer on disc.

In addition to sending the PØØ and PØ2 Q-blocks, PICKER also passes on SØ4 and SØ6 to MANAGER. MANAGER, on receipt of the SØ4 Q-block, passes it to GRAPHER and ANALYSER (Sect. 4.11); zeroes the IONOSOND file on disc, if required and if time allows; does some initializaing based on commands DROP and SAVE; and writes the current ICT to disc. (See Q.S.4, Sect. 4.8.)

On receipt of the SØ6 Q-block from PICKER, MANAGER passes it to GRAPHER and ANALYSER, and if the operator has selected tape output (Sect. 4.7), copies the sounding data from disc to tape. (See Q.S.6, Sect. 4.9.)

7.7.4. MANAGER Q-blocks

MØØ MØØ + Ø: Ø.
MØØ + 1: "M".
MØØ + 2: status.
MØØ + 4: address of MANAGER's UDL.
MØØ + 6: address of MANAGER's command mnemonic scan table.

Destination: COUNSEL.

Puprose: see SØØ, Section 7.7.2.

Recipient action: COUNSEL copies MØØ + 4 and MØØ + 6 to its UDL/SCAN table, then starts GRAPHER.

MØ2 MØ2 + Ø: 2.
MØ2 + 1: "M".
MØ2 + 2: status.
MØ2 + 4: address of ICT or data record, or Ø.

Destination: GRAPHER (or user-written ANALYSER).

Purpose: MØ2 is sent in response to GØ2 and GØ4 requests (Sect. 7.7.5). AØ2 and AØ4 (Sect. 7.7.6) are the equivalent requests from a user-written ANALYSER task. When GRAPHER requests data from a previous sounding with the GØ2 Q-block, MANAGER responds with MØ2 + 4 = A(ICT), or MØ2 + 4 = Ø if the data are not available. GRAPHER requests data records with the GØ4 Q-block; MANAGER responds with MØ2 + 4 = A(data record), or MØ2 + 4 = Ø if the data EOF is sensed.

Recipient action: Depending upon whether the MØ2 Q-block is a response to a request for a previous sounding (file request) or to a data record request, GRAPHER either gets scaling and labeling information from the ICT or plots the data record on the system display.

The action of ANALYSER, a user-written task, is undetermined. AØ2 and AØ4 are not implemented in the current ANALYSER scheduler.

MØ4 MØ4 + Ø: 4.
MØ4 + 1: "M".
MØ4 + 2: status.
MØ4 + 4: system display write flag.
Ø = suspend refreshing system display.
1 = resume refreshing, if refreshing was selected before tape write began.

Destination: GRAPHER.

Purpose: the system display, or refresh CRT, and the magnetic tape share an I/O Selector Channel (SELCH), and a hardware problem has resulted in occasional anomalous or missing tape records. The hardware solution remains illusive, and the conflict is avoided by simply not running the magtape and the system display concurrently. MANAGER tells GRAPHER when the tape write begins and ends.

Recipient action: GRAPHER refrains from refreshing the system display while MANAGER copies the sounding data from disc to tape. When the disc-to-tape operation is completed, GRAPHER resumes refreshing the CRT, unless it is inoperative or command [PE was given. GRAPHER does not clear the status of the MØ4 queue to suspend refreshing until the next refresh cycle is to begin.

In addition to sending the above Q-blocks, MANAGER also passes SØ4, SØ6, and PØ2 to GRAPHER. GRAPHER, on receipt of the SØ4 Q-block, initializes a display buffer based on the sounding mode in SØ4 + 4 (Sect. 7.7.2). The PØ2 queue causes GRAPHER to add the new data to its plot buffers; the SØ6 signals that the sounding has ended.

MANAGER also passes SØ4 and SØ6 to ANALYSER, if it exists (see PASS.ON, Sect. 4.11). The current ANALYSER scheduler adds 1ØØØ (hex) to the SØ4 or SØ6 status field, and issues no schedule commands during a sounding.

7.7.5. GRAPHER Q-blocks

GØØ GØØ + Ø: Ø.
GØØ + 1: "G".
GØØ + 2: status.
GØØ + 4: address of GRAPHER's UDL.
GØØ + 6: address of GRAPHER's command mnemonic scan table.

Destination: COUNSEL.

Purpose: see SØØ, Sect. 7.7.2.

Recipient action: COUNSEL copies GØØ + 4 and GØØ + 6 to its UDL/SCAN table, then starts ANALYSER.

GØ2 GØ2 + Ø: 2.
GØ2 + 1: "G".
GØ2 + 2: status.
GØ2 + 4: address for sounding ICT.
GØ2 + 6: sounding number or 999 (end data request).

Destination: MANAGER.

Purpose: GRAPHER's GØ2 queue is a file request, or a request for data from a previous sounding (Sects. 4.12 and 5.7).

Recipient action: if the requested file is available, MANAGER reads the file's ICT from disc, copies the ICT to GRAPHER memory, and puts the ICT buffer address in MØ2 + 4. If the data do not exist or have been overwritten, MØ2 + 4 is cleared to Ø. In either case the MØ2 Q-block is sent in response to the GØ2.

7.7.5. GRAPHER Q-blocks

G04 G04 + 0: 4.
G04 + 1: "G".

Destination: MANAGER.

Purpose: the G04 queue is a request for data from the sounding (data) file selected by the previous G02.

Recipient action: MANAGER reads the next data record and sends the M02 response to GRAPHER. If EOF is sensed, M04 + 4 is cleared to 0; otherwise, the buffer address of the data record is put into M02 + 4. The status clearing convention is not used.

MANAGER is written to handle file (A02) and data (A04) requests from a user-written ANALYSER task. GRAPHER and ANALYSER requests are logically identical from MANAGER's standpoint and can be interspersed (Sect. 4.12).

7.7.6. ANALYSER Q-blocks

A00 A00 + 0: 0.
A00 + 1: "A".
A00 + 2: status.
A00 + 4: address of ANALYSER's UDL.
A00 + 6: address of ANALYSER's command mnemonic scan table.

Destination: COUNSEL.

Purpose: see S00, Section 7.7.2.

Recipient action: COUNSEL copies A00 + 4 and A00 + 6 to its UDS/SCAN table, re-orders the table in command priority, and puts the table address into ET.SCAN (Sect. 7.3). All tasks have been started by and have responded to COUNSEL; COUNSEL is ready to accept sounding advice from the console.

The A00 Q-block is sent by the current ANALYSER stub and must also be sent by any user-written ANALYSER task. A02 and A04 requests can be sent and should be coded by the user to function like G02 and G04 (Sect. 7.7.5). A06 has tentatively been reserved for frequency selection feedback to SOUNDER. If A06 or other queues to SOUNDER are implemented, the user must also insert appropriate routines at A.ADV (Sect. 2.8.5).

When ANALYSER acts as the scheduler (see TEST task SCHED, Sect. 10.3), it sends C02 Q-blocks as though it were COUNSEL (7.7.1).

7.7.7. Q-block summary by recipient

COUNSEL	SØØ	SOUNDER started
	PØØ	PICKER started
	MØØ	MANAGER started
	GØØ	GRAPHER started
	AØØ	ANALYSER started
	ØØØ	Request to HALTIO for console write (see WRLOG, Sect. 2.10.8)
SOUNDER	CØØ	start
	CØ2	Console advice
	CØ4	Terminate
	AØ6	Frequency selection (must be user-implemented)
	TFF	TEST-task-termination/console-release requests
PICKER	CØØ	Start
	CØ2	Console advice
	CØ4	Terminate
	SØ4	Sounding start
	SØ2	Sounding data
	SØ6	Sounding end
MANAGER	CØØ	Start
	CØ2	Console advice
	CØ4	Terminate
	SØ4	Sounding start
	SØ6	Sounding end
	PØ2	Sounding data record
	GØ2	File request
	AØ2	File request
	GØ4	Data request
	AØ4	Data request
GRAPHER	CØØ	Start (from COUNSEL)
	CØØ	Restart (from SOUNDER)
	CØ2	Console advice
	CØ4	Terminate
	SØ4	Sounding start
	SØ6	Sounding end
	PØ2	Sounding data record
	SØ8	K-mode frequency, range limits
	MØ2	Response to file, data request
	MØ4	System display off, on

7.7.7. Q-block summary by recipient

ANALYSER C00 Start
C02 Console advice (user-implemented)
C04 Terminate
S04 Sounding start
S06 Sounding end
P02 Sounding data record
M02 Response to file or data request (user-implemented)

All TEST tasks receive the address of table TSTLST from SOUNDER as the start queue. (See Sects. 2.13.8 and 2.13.8.2.) In addition, the following TEST tasks receive queues from SOUNDER (Sect. 10.3):

DATAFMT (Test format of EKO data from FEP)
S04 Sounding start
S02 Sounding data
S06 Sounding end

CKDUP (Test for duplicate EKO buffers)
S04 Start
S02 Data
S06 End

TAPEKO (Copy PCT/EKO buffers to magtape)
S04 Start
S02 Data
S06 End

QSTOP (Test queues normally send to PICKER by SOUNDER)
S04 Start
S02 Data
S06 End

HELP (See ERROR, Sect. 2.10.7 and EXPLAIN, Sect. 10.3)
S04 Start
S02 Data
S06 End
SFF SOUNDER error information

EXOUT, in order to gain access to data from a previous sounding from MANAGER, assumes the role of GRAPHER, inasmuch as EXOUT sends a G02 queue to MANAGER to select a sounding data file and a G04 queue to get a data record. In turn, EXOUT receives two queues from MANAGER:

EXOUT (Examine PICKER output data)
M02 Response to G02
M02 Response to G04

8. COMMANDS (Console Advice)

Lorne D. Matheson
David C. Walden
James R. Winkelman

This chapter gives details of all commands that may be entered from the console device. The commands are in a single alphabetic order. The marginal line after each command gives the initial of the task that acts on the command. The S for SOUNDER is followed by Idle-mode, Immediate, or Ion. conf. (for sounding configuration) as described in Section 2.5.

In commands, mandatory letters are underlined.

ABORT
S, Immediate.

Terminate the running of a sounding or cancel scheduled sounding. If the ABORT command is given during the running of a sounding, the sounding is terminated at the end of the current frequency's pulse set. If a sounding is aborted during the countdown (see GO commands), the countdown continues to Ø but no pulses are transmitted; since SOUNDER sends the sounding-started Q-block (SØ4, see Section 2.10.9) at the beginning of the countdown and the sounding-ended Q-block (SØ6) as a result of the ABORT, MANAGER saves a null sounding on the disc.

If the ABORT command is given after the completion of a sounding, i.e., between soundings, the next scheduled sounding is canceled because an ABORT leaves SOUNDER in the idle mode, during which it does nothing until a new GO is given (see Sect. 2.7.1).

At the end of a sounding, whether it runs to completion or is ABORTed, SOUNDER sets the receiver and transmitter to maximum attenuation and sets the control mode for receiver devices to local.

The ABORT command cancels any NOTSG, NOFEP, NOEKO, or NOADC commands (qq. v.) previously issued. That is, for the next sounding SOUNDER will assume that the TSG, FEP, EKO, and housekeeping ADC (HKADC) are cabled into the system and functioning normally.

An ABORT during a tape replay run (see TPRUN) causes the magnetic tape to be rewound to the load point.

ANTENNAS

ANTENNAS list
S, Ion. conf.

Specify the receiver antenna selection sequence. Four pairs of arguments are required, separated by commas or spaces. They select the antenna inputs (1-4) for pulses 1-4 of a pulse set; each pair specifies the input for channel 1 and channel 2.

The selection of antenna inputs depends on how the antennas are plugged into the inputs. If the N, E, S, and W antennas are plugged into inputs 1, 2, 3, and 4, respectively, then the command:

ANT 1,3 1,3 2,4 2,4

provides the transmission sequence displayed in Figure 2.1 and in Table 2.1. The use of commas and a space in the example emphasizes the pairings.

S, Immediate. If ANT is issued during sounding, it immediately changes the selection sequence.

BIAS field
C, Debug.

Set hexadecimal bias value for debugging.

BOOT FEP
S, Idle-mode.

Initialize and load the FEP. The FEP is reset, initialized and, via the DIO, sent its thresholding program, which must reside on disc as file SYS1:FEPMEM.SND; this file is also used by command FEPTH.

BOOT FEP,xxx
S, Idle-mode.

Load the FEP from file SYS1:FEPMEM.xxx where xxx, the file name extension, is any three alphanumeric characters.

BOOT TSG
S, Idle-mode.

Bootstrap the TSG. A reset signal is sent to the TSG, which causes the TSG program memory and zero page to be loaded from the TSG PROM.

BOOT TSG,xxx
S, Idle-mode.

Load the TSG zero page from file SYS1:TSGMEM.xxx, where xxx, the file name extension, is any three alphanumeric characters.

CALA n
S, Ion. conf.

Set the calibration attenuation in dB. The calibration attenuator is set in 8-dB steps from Ø through 12Ø dB. If an out-of-step setting is specified, the next lower step is selected. For example, CALA 21 is equivalent to CALA 16. The calibration attenuation is initialized to 16 dB.

S, Immediate. If CALA is issued during sounding, it immediately changes the calibration attenuation.

CANHF
C — Cancel all tasks.

CDN n
S, Idle-mode. Specify the countdown value in seconds. The countdown can be set to 1 through 19 seconds. The default value is 10 seconds.

CEND
C — End a special COUNSEL command file. Open CON: as the input device and continue.

CLOSE
M — Close tape, write double EOF, and backspace.

COMMAND fn
C — Open file fn as a command file that normally contains parameter definitions such as K-mode frequencies and ranges. The file must end with CEND or another COM. A new command file replaces any old one.

COPY [n]
M — Copy the LU4 buffer, normally IONOSOND.SAV to tape in the compact mode described in Section 4.12. If n is neither zero nor blank, a double EOF is written after the last record. MANAGER is left in the same mode as that defined by TAPE 5. ABORT should precede COPY. See the associated commands in this chapter and Section 4.7 for details.

DELF f
S, Idle-mode. Set the delta frequency in kHz. DELF is initialized to 8 kHz and must be in the range 0 through 255 kHz. This is the deltaF in Table 2.1 used by command PSEQ. DELF readjusts the current protected frequencies map (see KRUN NOTE).

DESC desc
S, Ion. conf. Follow command by up to 8 ASCII characters describing the next sounding. Fewer than 8 characters are right filled with blanks by SOUNDER.

DFIX i
S, Ion. conf. Set the base frequency index increment value. The DFIX index is initialized to 10, but can be set to any positive decimal integer. This is the logarithmic increment between frequencies F1, F2, and F3 in Tables 2.1 and 2.2. It represents an ever-increasing increment in kHz as the base frequency sweeps from the low to the high limits of an I-mode sounding.

DLOFF

DLOFF
S, Immediate. Select the transmitter antenna load in place of the dummy load. The filament and high voltage states are unaffected.

DLON
S, Immediate. Select the transmitter dummy load (DL). DLON preserves the states of the filaments and the high voltage.

DROP [m]
M Drop m soundings (including the current one) by writing m soundings over the current one on disc. MANAGER enters the SAVE mode when the countdown ends. If m = Ø or is missing, it is set to 1. If it equals or exceeds 999, MANAGER remains in the DROP mode without counting down.

EFRD f
S, Ion. conf. Set the I-mode default ending frequency in kHz. The default ending frequency is initialized to 1Ø MHz, but can be set to any value up to 3Ø MHz provided that EFRD > SFRD.

For example, EFRD 15ØØØ sets the default ending frequency to 15 MHz. This value can be over-ridden by another EFRD command or by the EFRQ command for a single ionogram.

NOTE: All I-mode starting and ending frequencies are checked against the protected frequencies map (Sect. 2.12.9). If a specified starting frequency is protected, the next higher unprotected frequency is used; if a specified ending frequency is protected, the next unprotected frequency that is DFIX steps lower is used.

Instead of manipulating the sounding frequency directly, SOUNDER calculates the next frequency for I-mode by doing arithmetic on a frequency index. The frequency index ranges from Ø through 36ØØ representing frequencies from 62.5 kHz through 32 MHz; the index is linear in the log of the frequency. The frequency range represented is 9 octaves with 4ØØ selectable frequencies per octave, as shown in Table 8.1. Subroutine NTOFR is called to convert frequency index to frequency; FRTON converts frequency to the nearest corresponding frequency index. Because of hardware constraints, SOUNDER limits the actual frequency range to 1ØØ kHz through 3Ø.Ø MHz, corresponding to indices 274 through 3561. See SNDR, Section 7.6.

The general formula relating frequency and index I is f = Ø.Ø625*2**(I/4ØØ) MHz, where * means multiplication and ** exponentiation.

FREQUENCY INDEX	FREQUENCY, kHz
Ø	62.5
4ØØ	125.Ø
8ØØ	25Ø.Ø
12ØØ	5ØØ.Ø
16ØØ	1ØØØ.Ø
2ØØØ	2ØØØ.Ø
24ØØ	4ØØØ.Ø
28ØØ	8ØØØ.Ø
32ØØ	16ØØØ.Ø
36ØØ	32ØØØ.Ø

Table 8.1. FREQUENCY INDEX vs. FREQUENCY

EFRQ f
S, Ion. conf. Set the ending frequency for the next ionogram in kHz. For example, EFRQ 21234 sets the ending frequency for the next ionogram to 21.234 MHz. Subsequent ionograms end at or near the default ending frequency (see EFRD) unless EFRQ is repeated. See the NOTE under EFRD about protected frequencies and indexing.

EPF n
S, Ion. conf. Specify the maximum number of range-coincident echoes per frequency for PICKER. EPF is initialized to 8.

F
S, Idle-mode. Display all previously selected K-mode frequencies and their associated range limits.

F n
S, Idle-mode. Display the previously selected nth K-mode frequency and its range limits, where n lies between Ø and 9 inclusively. The following options affect the previously selected nth frequency:

F n+ Raise the nth frequency one step.

F n- Lower the nth frequency one step.

F nX Delete the nth frequency.

F n,FFF
S, Idle-mode. Define the nth K-mode frequency, n between Ø and 9 inclusively, as FFF kHz if the frequency is unprotected.

FEPTH n
S, Idle-mode. Specify the FEP data threshold and load the FEP. The argument for the FEPTH command is the desired FEP threshold factor, times 1Ø, to make it integer. SOUNDER associates this factor with a FEP program byte (see table below) which is inserted into the FEP program as it is

FEPTH n

given to the FEP. The default FEP threshold factor is 2.5, associated with program byte 72.

FEP thresholding values:

FACTOR x 10	PROGRAM BYTE
15	8
20	40
25	72
30	104
35	136
40	168
45	200
50	232

FEPTH causes the FEP to be loaded from the default FEPMEM disc file SYS1:FEPMEM.SND.

FORMAT [n]
M — Specify tape format:
n = 0 or missing: normal format (default).
n = 1: compact format without EOF.
n > 1: compact format with EOF's between soundings.

GCAL
G — Plot the calibration pulses with the other data.

GHMAX hhh
G — Set the upper edge of the ionogram display to hhh kilometers virtual height. The specified value is truncated to 100-km increments. GHMAX must be at least 200 km greater than GHMIN.

GHMIN hhh
G — Set the lower edge of the ionogram display to hhh kilometers virtual height. The specified value is truncated to 100-km increments. GHMIN must be at least 200 km below GHMAX.

GMOV
G — Allow movement/segment alternation while plotting multiple segment buffers.

GNOCA
G — Suppress the plotting of the calibration pulses, which is the default condition.

GNOMV
G — Plot all segments in multiple buffers without alternation.

GO
S, Idle-mode.

Immediately start the countdown followed by the production of an ionogram. The countdown may be set by CDN to 1 through 19 seconds.

GO mm
S, Idle-mode.

Schedule an ionogram to run at mm minutes past the current hour displayed on the TSG clock. The N-second countdown will begin N seconds before mm minutes past the current hour.

GO mm:ss
S, Idle-mode.

Schedule an ionogram to run at mm minutes and ss seconds past the current hour. The N-second countdown will begin N seconds before the specified time.

SOUNDER accepts any form of the GO command after it has been loaded from disc; thereafter, a GO must follow an ABORT, or a GO may be issued after NI (the number of soundings to run, q. v.) has been satisfied. If a time-specific GO command is given, it must be given at least N seconds before the specified time to allow for the countdown. Seconds (ss) must be in the range Ø through 59; minutes (mm) may range from Ø through 546. Thus, an ionogram can be scheduled to run up to 9.1 hours after the start of the current hour (9.1 hours is 32760 seconds; 32767 is the largest positive integer than can be expressed in a 16-bit half-word).

For example, if the TSG clock reads 10:15:37, the command GO 120 will result in scheduling an ionogram to run at 12:00:00, or 120 minutes past the start of the current hour 10; GO 16:7 will run an ionogram at 10:16:07; If N = 10 seconds, GO 15:45 would be rejected as an illegal start time.

GONO
G

Plot ordinary ray without the extraordinary ray.

GONX
G

Plot extraordinary ray without the ordinary ray.

GOVER n
G

Force GRAPHER overlay n for processing and display, where n has the following values:

GOVER n

1: ionogram display;
2: K-mode display;
3: skymap display;
Ø: restore automatic overlay selection.

GPION nnn
G — Get ionogram/kinosonde plot nnn from the disc. Plot number ØØØ refers to the last data set put on the disc, ØØ1 refers to the penultimate set put on the disc, etc.

GPSKY nnn
G — Get the previous data set as in GPION and display it in skymap mode.

HPE n
S, Ion. conf. — Specify the number of half-words per echo for the picked data. HPE is initialized to 17, allowing for a range value and 8 X-Y pairs.

HVOFF
S, Immediate. — Turn off the high power transmitter high voltage. If the system is left idle for any length of time, it is advisable to turn off the high voltage. HVOF may be issued at any time, however, and does not affect the transmitter filaments or the state of the dummy load (DL).

HVON
S, Idle-mode. — Turn on the high power transmitter high voltage. The high power high voltage is controlled independently of the filaments, but the filaments must be on and warmed up before the HV is turned on. If TXON has not been issued or the TX HV HOLD is true, indicating incomplete filament warm-up, HVON produces an error message.

HVON preserves the state of the dummy load (DL). See Figure 2.27.

KEEP m
S, Ion. conf. — Specify the M-out-of-N range coincidence criterion for PICKER. The number of possible coincidences (N) is set to 8 for dual-channel FEP data blocks and to 4 for a single channel. M can be set by the KEEP command to any integer in the range 1 through N. (See NOCH1 and NOCH2 as well as KEEP.) KEEP applies to both I- and K-mode soundings.

KFIL [n]
S, Idle-mode. — Set the overall sampling rate for KRUN by defining the number of NOP TSG starts between discrete frequency soundings (pulse sets). The default value is 1, which gives 1Ø frequency soundings per second if the 2Ø-ms TSG interrupt is selected (see TSGC).

KFIL [n]

If n is absent, display the current KRUN NOP fill count.

KRUN
S, Idle-mode.

Run a K-mode sounding on the previously selected frequency set in the KMODE table. Data are accumulated on the disc and, if selected, dumped to tape. Individual pulse sets are separated by KFIL NOP periods.

NOTE: Starting times for KRUN and VUF soundings may be specified as for the GO command. The pulse set sounding sequence for K-mode is controlled by commands DELF, PSEQ, and ANTENNAS, as for I-mode soundings. However, for K-mode, DELF and PSEQ settings should precede frequency selection, since these commands could force encroachment upon protected frequency bands by previously selected frequencies.

Both DELF and PSEQ cause the protected frequencies map to be readjusted and the currently selected K-mode frequencies to be confirmed as unprotected. Since previously selected K-mode frequencies could be deleted by this process, the current updated list of KMODE table entries is displayed, as though command F had been given, as a result of the DELF or PSEQ command. See Section 2.8.4.5.

KSET [n]
S, Idle-mode.

Specify the number of sets of discrete frequency soundings on the frequencies selected by command F n. The default value is 512. Assuming the 20-ms basic sounding cycle and 4 pulses per frequency pulse set, the sampling rate for each frequency of a set of N selected will be:

(1/N)*(1000/(80 + 20*KFIL)) per second

The run period (see KRUN) will be:

N*KSET*(80 + 20*KFIL)/1000 seconds

If n is absent, display the current value of KSET.

LINE field
C, Debug.

Display the next 24 bytes (12 half-words) after the field address in hexadecimal. After execution, LINE defines Y = X + 16 and Z = X + 24 (in bytes) rather than the usual definitions described in Section 6.2. X, however, retains its definition as the given field value.

MEMORY f1[sf2]

MEMORY f1[sf2]
C, Debug.

Display contents of memory address f1. If a separator s, either comma (,) or virgule (/) interchangeably, and field f2 appear, replace the contents of address f1 with the value f2. The possible values for these fields are described in Section 6.2.

NI n
S, Ion. conf.

Specify the number of soundings to run. NI is initialized to -1, which allows the running of soundings until ABORTed. NI can be set to any positive decimal integer up to the half-word limit, 32767. The sounding number (ICT.INUM, Section 7.1) is incremented from 1. When ICT.INUM = ICT.NI, SOUNDER is put into the idle mode (see Section 2.5), whereupon a new NI command and GO must be given in order to continue sounding.

NIPH n
S, Ion. conf.

Specify the number of soundings per hour to run. NIPH is initialized to 1. NIPH may be in the range 1 through 255 soundings per hour, or may be set to 999, which specifies free run, or back-to-back soundings. If NIPH is set too low to permit all tasks to complete within the NIPH schedule, SOUNDER defaults to free run as though NIPH had been set to 999.

NOADC
S, Idle-mode.

Flag the housekeeping ADC as not in the system. SOUNDER will not wait for the HKADC end-of-conversion signal if NOADC is used. NOADC remains in effect until an ABORT is issued.

NOANALYSER [n]
M

Ignore the ANALYSER task if n is Ø or missing (only SØ4 and SØ6 passed from SOUNDER to save time). Pass Q-blocks to ANALYSER if n is nonzero. See Section 4.7 for details.

NOCH1
S, Idle-mode.

Flag no FEP data for receiver channel 1 (see NOCH2).

NOCH2
S, Idle-mode.

Flag no FEP data for receiver channel 2. If either NOCH1 or NOCH2 is issued, N, the number of possible range coincidences, is reduced by 4. (See the KEEP command.) This and the previous command are useful in debugging.

NOEKO
S, Idle-mode.

Flag the EKO device as not in the system. SOUNDER can run without starting the DMA transfer of the FEP data. NOEKO remains in effect until an ABORT is issued.

NOFEP
S, Idle-mode.

Flag the FEP as not in the system. SOUNDER can run without the FEP if the FEP DATA READY (to the Interdata) and the FEP DONE (to the TSG) signals are strapped to simulate completed FEP action. NOFEP remains in effect until an ABORT is issued.

NOTSG
S, Idle-mode.

Flag the TSG as not in the system. For debugging, SOUNDER can run without the TSG by using the OS delay instead of waiting for the TSG DONE interrupt. NOTSG remains in effect until an ABORT is issued.

NOTE: If SOUNDER detects an error condition indicating that the TSG, FEP, EKO, or HKADC is not functioning normally, an error message is logged and the SOUNDER task is paused. If at this point the operator types:

TASK SOUNDER;CONTINUE

the SOUNDER task flags the device in question as not in the system, just as if one of the above four commands were issued, and continues to run. The operator can now ABORT the sounding and intervene, for example, by BOOTing the TSG or FEP or by turning on the power for the receiver devices. The ABORT cancels the not-in-system flag(s) (as described under ABORT) and permits the other tasks to terminate gracefully, e.g., MANAGER will write the double EOF to magnetic tape, if required. The operator can continue without having to reload the HF Radar software from disc.

[PA
G

Display test pattern points only.

PAUSE
C, Debug.

Pause 30 seconds during debugging.

[PB
G

Display test pattern vectors only.

[PC
G

Generate ionogram grid and labels from the ICT stored in GRAPHER; primarily used for graphics diagnostics and system tests.

[PE
G

End display refreshing.

PKHI n

PKHI n
S, Ion. conf. Specify the maximum FEP averaged peak in mV. PKHI is initialized to 9ØØØ mV. If the FEP data peak average computed by SOUNDER for a pulse set is greater than this value, more attenuation is put into the receivers for the current frequency band.

PKLO n
S, Ion. conf. Specify the minimum FEP averaged peak in mV. PKLO and PKHI (above) are used to set the limits for the automatic gain control (AGC) routine for the receivers. PKLO is initialized to 1ØØØ mV. If the FEP data peak average for a pulse set is less than this value, less attenuation is put into the receivers for the current frequency band. See the discussion of the FEP in Section 2.2 and the discussion of Period 7 in the example of Section 2.4.

[PS
G Restart display refreshing.

PSEQ n1,n2,n3,n4
S, Idle-mode. Specify pulse set sequence. Four arguments separated by commas or spaces are required. They select the synthesizer frequency settings, based on sounding frequency F, for pulses 1-4, respectively, of a pulse set. Valid arguments are Ø-7 and have the following meanings:

Ø: base frequency F
1: F + deltaF
2: F + 2*deltaF
3: F + 3*deltaF
4: F
5: F - deltaF
6: F - 2*deltaF
7: F - 3*deltaF

For example:

PS Ø,1,Ø,1

provides the pulse set sounding sequence displayed in Table 2.1.

The default sequence is Ø,Ø,1,1.

PSEQ readjusts the protected frequencies map (see KRUN NOTE).

[PZ
G — Display test pattern points and vectors.

RANGE [n]
P — Control I-mode range testing in PICKER. If n is zero or missing, no range testing is done. Otherwise, the range limits established for frequency n in the K-mode (see commands RN and RF) are used in the I-mode to exclude echoes outside the assigned range. See Section 3.10.

RF n,rmax
S, Idle-mode. — Define the maximum (farthest) range, rmax, in kilometers for frequency n of a K-mode sounding set (see command F n). The default range value is 500 km.

RGAIN
S, Idle-mode. — Save the receiver gain profile tables used for the ionogram last run and the K-mode last run. The IF and RF attenuation settings computed by the AGC routine (see PKLO and PKHI) are written to disc file SYS1:RGAIN.SAV, identified by sounding number ICT.INUM (Section 7.1). This file can be examined later by TEST task AGC.

RN n,rmin
S, Idle-mode. — Define the minimum (nearest) range, rmin, in kilometers for frequency n of a K-mode sounding set (see command F n). The default range value is 50 km.

SABORT
A — Abort the schedule initiated with a previous SRUN command. If a sounding is in progress, it will continue unless it is terminated with the ABORT command. If the scheduler (ANALYSER) is not active, the SABORT command is ignored.

SAVE [m]
M — Save the next m soundings on disc by writing them in order after the current sounding. The data are packed into the LU4 buffer, normally IONOSOND.SAV in compact format (Section 4.12). MANAGER starts in this mode and remains in it until this command sets a count or a DROP is executed. MANAGER enters the DROP mode when the SAVE countdown ends.

If m = 0 or is missing, it is defined as 1. If it equals or exceeds 999, MANAGER remains in the SAVE mode without counting down.

SFRD f
S, Ion. conf. — Set the default starting frequency in kHz. Unless specifically advised to do otherwise (see SFRQ) for a particular ionogram, SOUNDER starts the frequency sweep for an ionogram using the value in the current Ionogram Configuration Table (ICT) at displacement ICT.SFRD (Section 7.1). This value is initialized to 2 MHz but can

SFRD f

be set to any value between 100 kHz and up to, but less than, 30 MHz with the SFRD advice command.

The value entered must be expressed in kHz. For example, SFRD 4000 sets the default starting frequency to 4 MHz. This value can be over-ridden by another SFRD command or by the SFRQ command for a single ionogram.

See the NOTE under EFRD about protected frequencies and indexing.

SFRQ f
S, Ion. conf.

Set the starting frequency for the next ionogram in kHz. For example, SFRQ 2500 sets the starting frequency for the next ionogram to 2.5 MHz. Subsequent ionograms start at the default starting frequency (see SFRD) unless the SFRQ command is repeated. See the NOTE under EFRD about protected frequencies and indexing.

SLIST xxx
A

List schedule xxx to the console. The schedule identifier, xxx, is that specified by command SSAVE of TEST task SCHED (Sect. 10.3). Command SLIST is rejected if a sounding is in progress, or if the scheduler (ANALYSER) is active. See command SRUN xxx,n (below) for error messages.

SORT [n]
M

If n = 0 or is missing, no sort to tape (default). If n > 0, zero LU2 IONOSOND, wait for sounding data to be written on LU2 in ascending frequency order, and copy the non-zero sectors of LU2 onto tape.

SRUN xxx
A

Execute command from schedule xxx. The first time-specific GO or KRUN command (GO* or KRUN*) will get its argument from the first times-table entry. (See TEST task SCHED, command TIME, Sect. 10.3). See the next command for error messages.

SRUN xxx,n
A

Execute schedule xxx, but use the nth times-table entry for the first GO* or RUN*.

The SRUN command is rejected if a sounding is in progress, or if the scheduler (ANALYSER) is already active.

A schedule just executed, or terminated with SABORT (above), may be repeated by typing SRUN or SRUN ,n.

Errors resulting from SLIST and SRUN are listed below.

ERR 1 Schedule xxx does not exist.
ERR 2 The scheduler is active.
ERR 3 Time-table entry is invalid (SRUN only).
ERR 4 Sounding in progress.

SWAP
OS mod.

Write the current user memory onto file SWAPFILE.ROL. if a swap has occurred previously, the user memory is read from the other half of SWAPFILE.ROL; if not, user memory remains unchanged. Swap time-out is set to Ø for no time out. See Section 11.11.

TAPE [n]
M

If n is zero or missing, open tape. If n = 1, open tape, skip to double EOF, and backspace over the second EOF. See also CLOSE, FORMAT and SORT.

TEST task
S, Idle-mode.

Replace GRAPHER with the TEST task named by the argument. For a description of currently available TEST tasks, see Section 10.1.

TNEW
M

Rewind LU4 and restart at sector 1. This restarts the LU4 file after a COPY for subsequent writing of ionograms on disc in the DROP or SAVE modes. See Section 4.7 for details.

TPRUN
S, Idle-mode.

Run dummy sounding from previously recorded EKO data. The data are read from a magnetic tape written by TEST task TAPEKO (Section 10.3). The ICT and the KMODE table are established from tape, and after the countdown, SOUNDER reads the PCT/EKO buffer records and sends SØ2 Q-blocks to PICKER as it would during normal sounding. The sounding mode is set from ICT.MODE (Section 7.1) as read from tape. When the tape EOF is sensed, SOUNDER sends the SØ6 Q-block to PICKER.

During a TPRUN, SOUNDER does nothing but read tape and send Q-blocks to PICKER. The other tasks function normally, although tape recording must not be specified for MANAGER (see the TAPE command). During TPRUN, SOUNDER rejects all advice except ABORT, which command sends the SØ6 Q-block to PICKER and causes tape rewind. The KMODE table is not restored after a TPRUN; the ICT is properly updated when a normal sounding is started. TPRUN locks out TEST tasks; TEST locks out TPRUN.

TSGC n

TSGC n
S, Ion. conf.

Specify hexadecimal TSG program command byte and, by inference, the associated TSG range Ø correction. TSGC is initialized to A4 hex, specifying 5Ø pulses per second and a repeat count of 1 for program number 4, but may be set to any legal TSG command byte. See the Technical Manual for the TSG, Table 5.2, page 5-41, for descriptions of current TSG programs. TSGC applies to both I- and K-mode soundings.

TXATHP n
S, Ion. conf.

Set the high power transmitter attenuation in dB. This attenuation is initialized to 9 dB. The range is Ø-31 dB in 1-dB steps. The initial value is chosen to optimize high-power transmitter operation at maximum power. The attenuation may be increased from this value to decrease power.

TXATLP n
S, Ion. conf.

Set the low power transmitter attenuation in dB. The transmitter attenuator is set in 1-dB steps from Ø through 31 dB. This attenuation is initialized to 6 dB, which optimizes low power transmitter operation at maximum power.

TXOFF
S, Immediate.

Turn off the high power transmitter. The high power TX filaments and high voltage are shut down immediately, the dummy load is selected, and and the transmitter is fully attenuated. TXOFF can be issued at any time.

TXON
S, Idle-mode.

Turn on the high power transmitter. The high power TX ON sequence turns on the filaments and initiates a 9Ø-second filament warm-up period during which the transmitter high voltage cannot be turned on (see HVON). TXON selects maximum transmitter attenuation and preserves the state of the dummy load (DL). See Figure 2.27.

NOTE: When the HF Radar is first started, maximum transmitter attenuation is put in and the transmitter dummy load is selected, but the current states of the filaments and the transmitter high voltage are preserved.

VFIL [n]
S, Idle-mode.

Set the overall sampling rate for VUF by defining the number of NOP TSG starts between single-frequency pulse sets. The default value is n = 16.

If n is absent, display the current VUF NOP fill count.

VUF n
S, Idle-mode.

Run a 30-second single-frequency test sounding at the nth K-mode frequency (see F n) with VFIL NOP periods between individual pulse sets. No recording on disc or tape is made of the sounding. The system reverts to the command mode and ionogram display at the end of the sounding.

See the KRUN NOTE for a description of command interactions.

9. OPERATION

James R. Winkelman

This chapter begins with a tutorial for new users. The order of its steps allows for practice and introduces tape handling after everything else is going. Section 9.2 discusses emergencies for practiced as well as new users. Section 9.3 gives a short, practical operating sequence with reference to the tutorial steps for details.

9.1. How to Run a Sounding: Tutorial for the New User

The operating sequence step numbers have the prefix T for tutorial and to provide reference for Section 9.3.

T1 Power on — Verify that the power is on. Every installation will have its own power-up sequence.

Hints: plug the plugs into the wall; open the back of the computer and receiver racks and snap the switches on the plug assemblies; there may be one plug assembly on each side of the rack (four in all). The switch for the terminal is on the pedestal under the keyboard (use Braille or a flashlight). On the NSF system, the terminal is left on but plugged into one of the plug assemblies turned on earlier. Newer systems have a single switch on the front of each rack.

T2 Start Disc — 7/16: depress the left button (labeled START STOP) on the blue thing in the bottom of the computer rack.

8/16: Rock POWER ON switch (lower right) to ON and it will light. Rock the RUN-LOAD switch (lower center) to RUN.

Have a cup of coffee (40 - 1000 seconds). Eventually the ready light will go on: you must wait for it. Anything typed during disc not ready causes a system crash.

Comment on 8/16 Interdata discs:
This disc system has internal checks on disc speed and temperature. If the disc was cold when started, it can take a long time before the READY light goes on. It takes even longer for the PROTect light to go out when you switch off the write protection. Once the system is stable, the protect switches can be simply switched on and off.

T3 Screen

Clear the screen on the 4012 terminal. The screen comes on slowly to full brightness and should be cleared by depressing the RESET PAGE key on the upper left of the keyboard. This key can be pressed at any time without affecting the system. After pressing the key, wait for the flashing prompt before continuing typing.

T4 Unprotect

To run a sounding, the bottom (fixed) disc must be unprotected. To run the tests, the top (cartridge) disc must also be unprotected.

You have two choices:
a) Unprotect both discs: 7/16 buttons out, 8/16 buttons bottom edge in. The lights behind the switches will go out on the 7/16 immediately, and on the 8/16 sooner or later.
b) Go to step T5 and then at some time before typing HFRADAR (T6), set the switches as in a) above and type:

```
MARK DSC1:,ON
MARK DSC2:,ON
```

Side issue.
The bottom disc is mounted permanently in the unit and is often referred to as "fixed". Our system refers to it as device DSC2:. When it is marked on, the name of the data volume recorded on DSC2: is displayed (SYS1 in this case). In referring to the device (discheck, mark on, etc.), use the device name DSC2:. In referring to the data or files, use the volume name SYS1.

The removable cartridge (upper disc) is device DSC1:. To run soundings the cartridge SYSC must be mounted; the gummed label on the cartridge is the same as its volume name.

The names and statuses of devices may be determined by typing:

```
DISPLAY DEVICE
```

The devices DSCn: will list as OFF or the volume name will appear. If the name is followed by PROT, the disc may not be written upon without marking it on (T4).

9.1. How to Run a Sounding: Tutorial for the New User

T5 INI

Press the INI button on the right of the INTERDATA console momentarily. The wait light will go out and 10 - 12 seconds later the following message will appear on the screen of the 4012:

```
OS16 MT2 W03-00
DSC2:   SYS1
```

T6 Initialize

Type the following commands:

```
SET TIME mm/dd/yy,hh:mm:ss
VOLUME SYS1
HFRADAR
```

The marking on of protected discs must precede HFRADAR.

Relax! The file HFRADAR will set partitions, load the tasks, assign files, and finally start COUNSEL. Eventually all tasks will be logged in and the FEP and TSG loaded. When this is all done, the prompt *COUNSEL> appears on the screen. It's time for you to do something.

T7 Sounder

Type any sounder commands desired, waiting for a prompt before typing the next (some are slow). Delay tape commands until T11 and T12 while practicing.

T8 GO

Type GO to run a sounding.

T9 Stop

After the sounding type AB to stop the automatic sequence of soundings in order to practice something else.

Personal bias (JRW): leave the transmitter off during these early practice sessions and use a KEEP 2 to get noise on the plot. It's not pretty, but very quiet for the neighbors.

T10 Tape use

To use tape:

Make certain there is a write ring in the groove in the back of the tape. Mount the tape following the directions in the lower left of the 8/16 drive or the upper right of the door on the 7/16.

Nothing has to be opened. The tape simply slides into slots and goes around pulleys. Wrap the end of the tape two or three times around the take-up reel, keep taut, and keep finger out of hole while pressing the LOAD button.

Keep trying. The ON-OFF switch is at the lower left of a 7/16 unit; the POWER button corresponds on an 8/16.

The tape will move to the load point (a bright spot on tape). If the tape keeps moving for a long time, press REWIND.

Press ON LINE. Close the door. Relax.

To remove a tape, press ON LINE on a 7/16 to get it off line, or RESET on an 8/16. Press REWIND. After the tape is rewound, press REWIND again. Remove the tape. Remove the write ring if desired.

T11 Record on Tape

Recording on tape requires a mounted tape, on-line, and write enable. If the tape is positioned properly, type TAPE to make normal format recordings. If you wish to add data to the end of an existing tape and the tape is not at the correct position from previous recordings, type TAPE 1.

T12 Terminate

When all is done, type:

PAUSE

You get an * but no COUNSEL>. Immediately type:

CANHF

This command is followed by a group of END OF TASK's, .BG last, then a Display Time.

If SOUNDER were paused before the CANHF, there could be an error messasge (I plan to eliminate it in my next operating system changes). This error or similar ones ruin CANHF! Try a CANJRW, which will cancel the tasks from the other direction and usually fails only when reaching the task already canceled by CANHF.

Warning: CANJRW used in place of CANHF has a high probability of crashing the whole system (see Section 9.2).

After all else but the display time, type DISPLAY MEMORY to give the names of tasks that survived. For each of them type:

T12 Terminate

9.1. How to Run a Sounding: Tutorial for the New User

TASK taskname;CANCEL

T13 Protect

When done using the system, type:

MARK DSC1:,ON,PROT
MARK DSC2:,ON,PROT

Set switches to protect.

T14 Power off

When turning off the system completely, release the START STOP switch on the 7/16 or move the RUN LOAD switch to LOAD and wait until the start switch goes out or the LOAD light comes on. This removes power from the disc gently and is important.

9.2. Emergencies

What to do until the doctor comes. For both experienced and new users, some suggestions on what to do when nothing happens or something goes wrong.

Symptom: you type and nothing happens, not even the echoing of characters. Sometimes the display stops counting seconds. This is termed a *CRASH* and happens to everyone occasionally.

After a crash, press INI. If SYSC were unprotected before the crash, type DSC1 (which closes all open files on DSC1:). Do a V SYSC followed by a DSC2. A disc check cannot be run automatically from its own volume.

You have started the system and are getting many error messages. Nothing is running quite right. Someone probably shut it off wihtout closing down properly. Try DSC1 and DSC2 as in the paragraph above.

If none of the above work, get help! I don't know what to do either!

9.3. Practical Operating Sequence

The following operating sequence makes a better use of time than the tutorial sequence in Section 9.1. This practical sequence serves as a general checklist of necessary processes and necessary times to do optional processes. Reference to details in Section 9.1 is made by the T-step number.

1 Power — Turn on both switches in the digital cabinet and both switches in the analog cabinet (T1).

2 Disc — Turn on the disc (T2).

3 Screen — Clear the 4Ø12 screen (T3).

4 Tape & TX — While the disc comes up to speed: mount a tape (T10), turn on the transmitter, or get something to drink (local option).

5 Unprotect — After the disc READY light is on, unprotect the fixed disc (T4). The rightmost switch on the disc must be out and its light off.

6 DSC & File — Set DSC switch down and file switch to 1.

7 INI — Press the INI button on the Interdata console (T5). After the system has been loaded, a response appears on the 4Ø12.

8 Initialize — Type (T6):

```
SET TIME mm/dd/yy,hh:mm:ss
MARK DSC2:,ON
VOLUME SYS1
HFRADAR
```

9 *COUNSEL> — Relax until the *COUNSEL> prompt appears and then enter desired commands (T6, T7).

1Ø GO — Type GO after all advice has been given to the programs.

11 AB — Type AB after a sounding to prevent the next automatic sounding.

The following commands are required to close down the system (T12, T13).

88 PA PAUSE

89 CANHF CANHF

90 Protect MARK DSC2:,ON,P
MARK DSC1:ON,P

91 Disc off Turn off the disc unit.

92 Power off When the ready light goes out, you may turn off power switches as required. Enjoy the silence.

9.4. Assembly and Loading

In order to make a change in the HF Radar software, the files must be edited, assembled, established as a task, and finally loaded and started. This sections provides information on these procedures.

CSS files CSS files appear to the casual user to be the same as system commands. They may not be abbreviated and are actually lists of commands. They are widely used to save typing and to obviate the need to remember all of the details of running certain programs.

The CSS files mentioned in the first three sections include HFRADAR, CANHF, and DSC1.

The editor is called by typing EDIT. Get the file to be edited and SAVE the edited version. The names are associated with the listings. The assembly files are extension .CAL, which get assembled into .OBJ and established into .TSK, the actual tasks.

Current versions are:

COUNSEL	SNDRPIPE
PICKER	SNDRADVC
MANAGER	SNDRNXTF
GRAFPRO	SNDRSUBS
GRAFRAM	SNDREENT
GRAFBUF	SNDRTBLS
ASKED	SNDRBFRS

Assembly

Option CROSS for obtaining cross reference lists is always present. The assembler is invoked by CAL followed by the name, listing, and options. Examples:

```
CAL PICKER
```

PICKER.CAL is assembled into PICKER.OBJ with listing on PICKER.LST.

```
CAL MANAGER,,NLSTC
```

MANAGER.CAL is assembled into MANAGER.OBJ with listing on MANAGER.LST with option NLSTC.

```
CAL COUNSEL,MAG:
```

COUNSEL.CAL is assembled into COUNSEL.OBJ with listing on magnetic tape.

TET

There are three Task Establishment Tasks: TETGRAF, TETSNDR, and TET. They are invoked with a name, bias, and priority.

The priorities have been assigned as follows:

COUNSEL	43
SOUNDER	60
PICKER	70
MANAGER	80
GRAPHER	90
ANALYSER	110
.BG	128

COUNSEL resets its priority to 100 after the last task is started.

The biases are found from the CSS file that will load and run the tasks, in this case HFRADAR (see below). The tasks may load in any order, but they must find a partition with exactly the same starting address as their bias at TET time.

Examples:

```
TETGRAF GRAPHER,CAØØ,9Ø
```

combines the three GRAPHER OBJ files and creates GRAPHER.TSK and GRAPHER.MAP.

```
TETSNDR SOUNDER,6622,6Ø
```

combines the seven SOUNDER OBJ files and creates SOUNDER.TSK and SOUNDER.MAP.

```
TET COUNSEL,6ØØØ,43
```

COUNSEL.OBJ to COUNSEL.TSK with the map in COUNSEL.MAP.

Load and run

HFRADAR.CSS controls loading. The six paritions are set. Each partition must start at a bias which was assigned during TET. The distance to the next partition must be long enough to hold the task to be loaded into the partition.

The load commands specify both the name assigned to the task and the name of the file to be loaded. The task names appear many times in the code and should not be changed.

Figure 9.1 is a listing of HFRADAR.CSS. Note the LOAD ANALYSER, ASKED.TSK.

Some of the tasks require assignments of logical units for both flexibility and to save space in the task.

GRAPHER needs a contiguous file to save the plot buffer of the latest ionogram. This file, GRAPHER.PLT must be at least 26 sectors long in the current version:

```
ALLOCATE GRAPHER.PLT,CO,26
```

MANAGER requires two files:

LU2, normally IONOSOND.
LU4, normally IONOSOND.SAV

LU2 is used for sorting and intermediate storage for normal format I-mode tapes. It is not used for K-mode or compact format tapes. Its length is defined in Section 4.12 as LAST.

```
*  HFRADAR.CSS  Ø3/29/79  JRW, DCW
TASK .BG
$COPY
*LOAD: 1/6ØØØ 2/6622 3/BØØØ 4/CØØØ 5/CAØØ 6/EBØØ .SYS/FØØØ
$NOCOPY
SET PARTITION 1/FØØØ,.SYS/FØØØ
SET PARTITION 6/EBØØ,5/CAØØ,4/CØØØ,3/BØØØ,2/6622,1/6ØØØ
LOAD COUNSEL,COUNSEL.TSK
LOAD SOUNDER,SOUNDER.TSK
LOAD PICKER,PICKER.TSK
LOAD MANAGER,MANAGER.TSK
TASK MANAGER
ASSIGN 2,IONOSOND.
ASSIGN 4,IONOSOND.SAV
LOAD GRAPHER,GRAPHER.TSK
LOAD ANALYSER,ASKED.TSK
TASK COUNSEL
ASSIGN 1,CON:
*THE ABOVE CON: MAY BE REPLACED BY A COMMAND FILE
*IF DESIRED.
START
$EXIT
```

Figure 9.1. HFRADAR.CSS

Sample lengths for IONOSOND.:

3ØØØ kHz to 15ØØØ kHz (93Ø,DFIX=2),(186Ø,DFIX=1)
1ØØ kHz to 3ØØØØ kHz (3292,DFIX=2),(6584,DFIX=1)

If the disc is running short of space, you must limit DFIX, the width of the scan, or the number of echoes. The above estimates are halved for a maximum of 6 echoes per PØ2 block rather than 8.

LU4 is used to save old soundings. With compact tape or KRUN, LU4 must be large enough to hold the entire sounding. Sizes given for LU2 are always large enough, but may not leave room to save the previous sounding.

I recommend starting with a clean SYS1:, adding an OS16 system or two, a copy routine, a few CSS's, the necessary .TSK's, and then defining GRAPHER.PLT and IONOSOND. A FILI command will give both unused space and maximum

contiguous space allowable for files. Allocate the largest IONOSOND.SAV allowable while leaving perhaps 100 sectors unused.

9.5. SYSGEN

System generation consists of defining the parameters of the system needed or wanted, and generating that system as OS16/MT2. The description includes a CUPS file, but, because the resulting CUP file must be edited before it can be used, the desired conditions may be edited in without using a CUPS file.

The SYSGEN disc (SYSD) is nearly full and contains only those files that are necessary. The CSS files generally use SYS1 for scratch files. TET716 and TET816 put both SYSMAP and the OS16 file onto SYS1.

EXEC16.CAL, FMGR16.CAL, and CMDP16.CAL are the original files from Interdata. They may be deleted, yielding about 7000 sectors. If SELS is used instead of SEL, files EXEC.CAL, FMGR.CAL and CMDP.CAL will have records of 70 rather than 80 bytes. These save 10 bytes*25000 records, or about 900 sectors; this reduction does not affect assemblies. If a listing is needed, one of the files may be made with SEL, assembled then deleted, and the next file made.

CUP xxx — CUP (Configuration Utility Program) starts with a source file CUPS and generates a CAL file CUP. No SELCH device is specified for the display unit PLT because this results in an error (CUP doesn't run right). The command file CUP.CSS xxx converts the source file CUPS.xxx to the CAL file CUP.xxx, where xxx is the extension, normally an identifying number for the CUP file. This cup file must be edited to insert the line at line 203 and line 288 defining the SELCH for device PLT. The SEL. definitions were inserted as lines 250 to 254. Lines 248 and 249 are not needed, as they state that SVC6 and SVC2 are not overlayed, which is the default condition.

EDIT — Editing takes place as follows:

```
*EDIT
.BG: OS/16 EDIT R00
.BG> GET CUP.700
.BG> CHANGE \$PLT@\,\ FFF0\,1-\PLT@\1
.BG: 191 DC FFF0BSY,0,U42TRM,0
.BG> INSERT
.BG> 202.01   DC X'F000',FF57BSY
.BG> 202.02
```

```
.BG> INCLUDE \CSS\,CUPS.700,\SEL.\-
.BG> INSERT \FFF0BSY\1
.BG> 282.01 FF57BSY DC 0
.BG> 282.02
.BG> SAVE CUP.700
.BG> END
```

This CUP file is used to control the assembly of the systems programs. For the 8/16 use X'F200' above.

CALSQ xxx — This file uses CUP.xxx and programs EXEC.CAL, FMGR.CAL, and CMDP.CAL to produce EXEC.xxx, FMGR.xxx, and CMDP.xxx. No listings are produced so as to reduce the assembly time from several hours to less than a half hour. Furthermore, there is not enough space on the disc for all of the listings.

TET716 xxx — EXEC.xxx, FMGR.xxx, CMDP.xxx and DSCDMA.OBJ (716 disc driver), SELLIB.OBJ (the radar drivers) and DLIB16.OBJ combine to produce system OS16Wxxx.xxx and map file SYSMAP.xxx.

TET816 xxx — Like TET716 for the 8/16. It does not use DSCDMA which includes TRAPS for the floating point IFX, FLOAT, STME, and LME.

The following is for people who want to do more systems work.

CALSYS — CALSYS assembles a systems program as follows:

```
CALSYS name,xxx,output,options
```

where the name is normally EXEC, FMGR, or CMDP; and xxx is as described under CALSQ.

Output:
null uses name .LST, nonnull is the actual file or device. The listing may use as many as 8000 sectors.

Options:
assembly options, such as CROSS and NLSTC, separated by commas. An assembly with cross references may take 99 minutes.

Example:

```
CALSYS CMDP,718,,CROSS,NLSTC
```

CALSYS1 xxx The same as CALSYS except that the listing file is SYS1:NAME.LST. There is often no room on the system generation disc.

Editing files this large is difficult, slow, and requires a lot of disc space. The files are normally modified by use of "SOURCE UPDATER" which uses the sequence numbers for updating. Two local programs SEL and COMP simulate parts of "SOURCE UPDATER" and are described below.

SEL SEL does the SELECT function ten times faster. It has the following arguments:

```
SEL  S,IN,OUT
```

S is the source operator file, normally small like E.035. IN is the input file like EXEC16.CAL. OUT is the output file like EXEC.CAL.

Method:

S and IN are read a record at a time. If the sequence number for IN is less than that for S, the record for IN is copied to OUT and a new record is read from IN; similarly for S less than IN. If the sequence numbers for S and IN agree, S is written, IN is forgotten, and new records are read from both S and IN.

S may be exactly a SELECT file for "SOURCE UPDATER". Any lines without sequence numbers are ignored.

Examples:

```
SEL  E.ØØ1,EXEC16.CAL,EXEC1.CAL
SEL  E.ØØ2,EXEC1.CAL,EXEC.CAL
```

This will result in the same EXEC.CAL file as:

```
SEL  E.ØØ1,E.ØØ2,E.Ø12
SEL  E.Ø12,EXEC16.CAL,EXEC.CAL
```

These examples reveal that the operation is associative:

$$(a \oplus b) \oplus c = a \oplus (b \oplus c)$$

The development disc will contain both EXEC16.CAL as received from Interdata and EXEC.CAL and E.Ø21 which produced it. New releases of changes will be to EXEC.CAL, that is, they will have to be applied to E.Ø21 and EXEC16.CAL or can simply be applied to EXEC.CAL. If you keep both versions, space will be at a premium.

COMP

COMP does the COMPARE function faster as well as keeping the files alligned by sequence number (an insertion does not result in all future lines being listed) and generating an output file that could be used in SEL to change one of the files to the other in some cases.

COMP has the following arguments:

```
COMP  IN1,IN2,OUT
```

IN1 and IN2 are compared and differences are printed with a 1 or 2 according to which file was listed. OUT is optional, but if it is present, it can be used in

```
SEL  OUT,IN1,IN2
```

to regenerate IN2 from IN1 except where lines of IN2 have actually been deleted. To delete using SEL, include a comment card with the sequence number to be deleted. The line remains but generates no code when assembled.

9.6. Disc Compress, Copy, and List

This section collects a miscellany of useful, related operations.

DISCMP

There are six disc compresses:

DISCMP1M.CSS	DSC1: to MAG:
DISCMPM1.CSS	MAG: to DSC1:
DISCMP2M.CSS	DSC2: to MAG:
DISCMPM2.CSS	MAG: to DSC2:
DISCMP12.CSS	DSC1: to DSC2:
DISCMP21.CSS	DSC2: to DSC1:

These CSS's copy a complete disc to or from another disc or magnetic tape. They are used to save a disc so that it can be recovered later. They are also used to copy a new set of files from another installation onto a disc. If a

disc has insufficient contiguous sectors, the disc may be compressed to tape and back to disc.

The buffers used are 12k in size. This long buffer runs faster and more reliably than a shorter one.

I know of no way to recover from a tape error.

To tape:
Mount a tape at load point and run DISCMP1M or DISCMP2M. The disc being copied must be protected.

From tape:
No file name on disc can match a file name on tape. This example starts from an empty disc:

```
MARK DSC1:,OF
INIT DSC1:,SYSB
DSC1
DISCMPM1
```

This process takes about an hour.

The DSC1 checks for sectors that may be hard to read. It is more thorough than READCHECK in INIT.

| LIS

This operation lists lines on the 4012 (CON:) and has the following arguments:

```
LIS.CSS name,F,L
```

Lines F through L are displayed. The line number and date are printed at the top of each page.

If F is omitted, listing starts with line 1.

If L is omitted, the listing ends with line 65530.

Any separator may be used between F and L.

Lines longer than 72 characters are continued on the next line preceded by ">-->".

| TYP

This is the same as LIS for the teletype (TTY:).

CPY

This is a replacement program for OS COPY for many of the normal copy needs. It has the following arguments:

```
CPY name1,name2
```

A file called name2 is made that is the same as name1, that is, with the same record size, type (contiguous or indexed), etc. If both old and new files are on disc, the new file will be contiguous if the old file had record length 256.

COP

COP copies a file from one disc to the other:

```
COP name,from,to
```

The volume names are assumed to be of the form SYSx. Any separators may take the place of the commas.

```
COP EXEC.CAL,D,1
```

copies the file SYSD:EXEC.CAL to SYS1:EXEC.CAL.

9.7. Saving and Restoring the Disc

The following routines constitute an alternative to DISCMP, which is extremely sensitive to tape condition and is slow.

The n with the first three commands described below is 1 for DSC1:, the top disc (removable cartridge), or 2 for DSC2:, the bottom (fixed) disc.

SAVDISK n
GETDISK n

SAVDISK copies the disc to tape, one record for each of the 408 tracks. Additional identical records are written for any record with a parity error during writing.

GETDISK copies the tape to a disc, allowing for duplicate records.

The tape format for each file, one disc to a file, is: end-of-file (EOF), 40-byte label, and 408 or more 12290-byte data records. The tape should be positioned ahead of the EOF. For example, to GETDISK the second file on a tape, type:

```
FF MAG:
FF MAG:
BF MAG:
```

The first FF will move a few inches over the EOF at the start of the first file. The second FF will skip the file (about 3 minutes). The BF will only move the tape about an inch.

Before writing the tape, SAVDISK will ask for a label. Type in up to 40 characters. For example:

```
SYSD 06/21/79 JRWINKELMAN.
```

GETDISK will display the label. If the file is correct, type Y. If the file is not the one desired, type N. The program will pause and you may reposition the tape using FF and BF commands. Type CON to continue the program and it will display the new label.

SAVDISK and GETDISK have some advantages over Interdata's DISCOMP: they are faster, they allow multiple files on a tape, they identify the files, preserve the created and used dates, and the have good error recovery.

They have the disadvantage of not compressing the disc.

COMDISK n — COMDISK does a partial compression of the disc without the use of tape. The number of sectors required for the data is determined. Any sector associated with an indexed file which is above the necessary boundary is copied to an unused sector below that boundary. Only contiguous files remain above the boundary.

JC file name — Copies files to another location on disc. JC ends with an allocate-size error that resets the file manager to assign the next file as low on disc as possible.

This routine is normally used to copy contiguous files as low on the disc as they will fit, leaving the top of disc as empty (long contiguous space) as possible. It moves continguous files, not handled by COMDISK, under the supervision of an operator.

MAKESYS1 a,b,c,d

MAKESYS1 initializes the fixed disc as name SYS1 and copies the files needed to run soundings from SYSC. It allocates IONOSOND., length 3072; IONOSOND.SAV, length 10000; SWAPFILE.ROL, length 384; and contiguous files for GRAPHER.PLT and SOVLAY.SND.

It also copies the operating systems OS16Wa.000, OS16Wb.001, OS16Wc.002, and OS16Wd.003 from SYSC to SYS1.

The arguments a, b, c, and d can be null but are usually the number of a SYSGEN such as 718, 701, 802, etc.

Example:

```
MAKESYS1 700,702,800,802
```

copies many files and tasks from SYSC to SYS1 including:

```
OS16W700.000
OS16W702.001
OS16W800.002
OS16W802.003
```

10. UTILITY SOFTWARE

David C. Walden
James R. Winkelman

10.1. Introduction [D.C.W.]

An effort has been made to provide the HF Radar users with a set of utility and diagnostic routines closely associated with the basic software. These routines provide general as well as detailed information about the health of both the hardware and the sounding software. Since most hardware malfunctions are detected by the SOUNDER task, and since the software data pipeline within the Interdata computer is initiated by SOUNDER, we have:

1. Added a TEST mode to SOUNDER.
2. Used foreground partition 5, allocated to GRAPHER, for running the Test tasks.

Between soundings, and if GRAPHER has nothing to do, TEST tasks can be loaded into partition 5 and executed. When TESTing is completed, GRAPHER is reinstated and normal sounding can resume. Thus, TEST commands are an extension of the SOUNDER advice command set, and the time-consuming and irritating necessity of canceling the HF Radar software in order to test the system is obviated. Furthermore, TESTing can be done under conditions that most nearly resemble those of actual sounding.

There are a number of SEL tasks that have not been incorporated into the TEST task system. These will be referred to as utility tasks to distinguish them from the tasks that are callable with the TEST command. Some of the utility tasks will later be modified to run as TEST tasks; some were written before the SOUNDER task and their major routines were later moved to SOUNDER, making their functions largely redundant (e.g., FEPTEST); some do not properly belong in the TEST system (e.g., DUTIL, FEPMEM). It should be further noted that while most of the TEST tasks depend on table and subroutine addresses, etc., from SOUNDER, a few of them can run as stand-alone programs. These latter are TXSTAT, SCLOOP, SNRIO, HKADC, TSGINT, PIPLN, and DUMP. Any of these seven TEST tasks can be run as a stand-alone program if no other tasks are active by typing:

TRUN <name>,

where <name> is the name of the TEST task. To terminate a stand-alone TEST task, type the ABORT command to put the task in paused state. Then type:

10.1. Introduction

TASK <name>;CANCEL

Do not attempt to run non-stand-alone TEST tasks in this manner because the results are at best unpredictable.

Instructions for running the utility tasks are included in the sections devoted to them.

The set of TEST/utility tasks at this stage of development is far from comprehensive. The tasks were written as they were needed in the course of hardware and software development, and, as may be said of any assortment of tools, they vary widely in usefulness. Some of the newer tasks have been used but little, so their true usefulness has not been established. We are most interested in your experience with the TEST/utility software and in your suggestions for expansion and improvement.

All commands end with a carriage return, designated by "(cr)". It is usually omitted from the descriptions, but appears whenever it is necessary to distinguish forms of a command and when it is itself a command.

Mandatory command characters are underlined. Optional arguments appear in square brackets.

TEST and utility tasks display the prompt character ">" whenever they require command input from the console. If a task requests further input, the request itself is the prompt.

10.2. TEST-mode Access [D.C.W]

To run any TEST task, give the following advice to SOUNDER:

ABORT	SOUNDER must be in idle mode.
TEST <TEST>	<TEST> = name of test task.

All TEST tasks that accept commands are terminated with the ABORT command; the others terminate themselves. When a TEST task is terminated, GRAPHER is automatically reinstated and normal sounding can be resumed.

SOUNDER has a total of 6 EKO buffers available to it. During normal sounding, all 6 buffers are used to minimize the chance of a data overrun. In the TEST mode, only 4 EKO buffers are allocated for the taking of data from the FEP. Otherwise, TEST and normal soundings are identical as far as SOUNDER is concerned. However, since the TEST task in partition 5 assumes PICKER's position in the queueing chain and does

not pass on the Q-blocks it receives, PICKER and MANAGER are idle during a TEST sounding.

It is not possible to sound with a TEST task that is not capable of handling the messages that SOUNDER normally sends to PICKER. Such a task keeps control of the console until it terminates itself, so the starting command (GO, KRUN, or VUF) cannot be given to SOUNDER. Tasks that do support a TEST sounding release the console back to COUNSEL and wait for the SØ4 Q-block from SOUNDER, indicating that a TEST sounding has been initiated by a GO, KRUN, or VUF command.

10.3. TEST Tasks [D.C.W]

Available TEST task descriptions follow. If a task is listed as missing by the MENU command in task HELP (below), it may only have been left open by a previous crash. If such is the case, running DISCHECK will make it available again. Remember that both discs (top and bottom) are used by the TEST mode and must be MARKed ON. If a TEST task has been established with the wrong BIAS (as reported by HELP, below), it can be re-established with:

TETEST <TASK>,BIAS,65

where <TASK> is the TEST task name, BIAS is the correct BIAS as listed by HELP. 65 is the priority of all TEST tasks.

All TEST tasks are invoked by

TEST <TASK>

where <TASK> is the name of a task given below.

HELP — Interpret SOUNDER task errors and describe all available TEST tasks.

HELP commands are:

(cr) — A carriage return by itself gives a brief list of HELP commands.

EXPLAIN — After EX is typed and the COUNSEL prompt is seen, type GO, KRUN, or VUF to run the TEST sounding. If an error is sensed in SOUNDER code, the following occurs: after SOUNDER types ERROR EEEE AT NNNN, SOUNDER queues the HELP task with the SFF q-block to pass it the values EEEE (error code) and NNNN (location of the error). HELP interprets the error message and types:

ERRNN in SNDRXXXX at LLLLR.

ERRNN is the label of the error call in SOUNDER code, XXXX is the SOUNDER program ID (see the SOUNDER load map, SOUNDER.MAP), and LLLL is the relative location of the error call in SNDRXXXX. The above information can be used to refer to specific SOUNDER listings for more details about the error. HELP then gives a brief description of the error with follow-up suggestions. EXPLAIN remains in effect for one TEST sounding only. When the sounding ends, HELP is terminated, and must be restarted (TEST HELP) if more testing is required.

ERROR EEEE,NNNN
Interpret a previous SOUNDER error message as described under EXPLAIN, above.

MENU [name]
List all available TEST tasks and commands if name absent. Otherwise, list description and options for the named TEST task only.

ABORT Terminate the HELP task if in other than the EXPLAIN mode.

FRMAP Construct and display the protected-frequencies maps for sounding in ionogram or kinesonde mode. Although the nominal range of the system is 0.10-30. MHz, the ultimate responsibility for frequency selection within this range must be taken at each site as local conditions dictate.

We have assumed that all sites will wish to protect the WWV channels at 2.5, 5, 10, 15, 20, and 25 MHz: a map protecting these frequencies, plus and minus 50 kHz, is kept on disc file SYS1:WHUSH.SND. This file is used as the starter map for the I-mode and K-mode local maps which are saved on disc files SYS1:IHUSH.SND and SYS1:KHUSH.SND, respectively.

File WHUSH.SND must be intact for new maps to be constructed. A back-up copy is kept on volume SYSC of the disc.

In general, K-mode maps will be more restrictive than I-mode maps: frequencies may be protected from repetitious soundings, while not protected from a single sounding during a sweep.

SOUNDER is responsible for having the appropriate map in memory for a given sounding. After reading a map from disc, SOUNDER adjusts it to the current delta frequency (see DELF, Chapter 8) and to the current pulse set sounding sequence (see PSEQ, Chapter 8) so that protected frequencies are not violated by sounding on an adjacent unprotected base frequency plus or minus multiples of the delta frequency.

FRMAP constructs only the unadjusted, base maps, but displays protected frequency bands for both base and adjusted maps.

The FRMAP base map commands are:

GET Select a map for listing or editing, or both. When FRMAP requests the map type, respond with:

I	for the I-mode map
K	for the K-mode map
IN	for the internationally protected frequencies map

The IN-map may be listed, but cannot be modified.

If a sounding mode (I or K) map is selected, the copy on disc is read into memory, if the file exists. If it does not exist, a new map is created and initialized to protect all frequencies. The map selected becomes the current map and remains so until a new GET is typed.

The current map exists only in memory until explicitly written to disc (see SAVE and ABORT).

PROTECT f1[-f2]
Edit the current map to protect frequency f1 or band f1-f2. The frequencies are expressed in decimal kHz. The band protected is the specified band expanded 50 kHz on each end to allow for hardware uncertainties. No check is made to see if part or all of the specified band is already protected. The frequency band finally protected reflects the algorithms used to convert frequency to frequency index (SOUNDER subroutines FRTON and NTOFR).

UNPROTECT f1[-f2]
Edit the current map to release protection of frequency f1 or of band f1-f2. The specified band, expanded 50 kHz on each end, is unprotected. No check is made to see if all or part of a protected band is being unprotected. The command:

UN 100-30000

unprotects the entire useable frequency band, except for internationally protected frequencies.

LIST Display the frequency bands protected by the current base map. The internationally protected frequencies bands are listed first and, if a sounding mode (I or K) map is the current one, the locally protected bands follow. Finally, the combined protected bands are displayed.

SAVE Save the current map on disc for use by the SOUNDER task. The current map may be edited further or a new current map may be selected (see GET).

Note: If any file or disc errors are detected during the execution of the above base map commands, the MT2 system error codes are displayed (see the Programmer's Reference Guide, Chapter 8); FRMAP requests another command and does not execute the command in which the error was detected. If it is necessary to get back to the MT2 command processor in order to mark a disc on, rename a file, etc., ABORT the FRMAP task and type PAUSE to COUNSEL. Corrective action can now be taken.

The following two commands are used to display the frequency bands actually protected by the map in SOUNDER memory, after SOUNDER has adjusted the map to the current values of DELF and PSEQ (Chapter 8). To display the useable, unprotected frequencies for an ionogram sweep requires consideration of the frequency index increment (DFIX, Chapter 8); such a display is planned for the future.

ICUR Display SOUNDER's current ionogram mode protected frequencies bands.

KCUR — Display SOUNDER's current kinesonde mode protected frequencies bands.

Finally:

ABORT — Terminate FRMAP task. ABORT can be issued at any time. However, if ABORT is typed before a new or newly edited map has been written to disc, a warning to SAVE the map is displayed, and FRMAP waits for a new command. A second consecutive ABORT is taken seriously. A GET without an apparently desirable SAVE results in the above action, also. A second consecutive GET results in the old current map being lost.

CONFIG — Display the configuration for the next sounding. All sounding variables subject to SOUNDER advice are displayed, with the advice commands that affect them shown in parentheses.

There are now no commands.

TXSTAT — Display the status of the high power transmitter. The transmitter status display is a facsimile of the transmitter control panel, with X indicating an LED that is lit.

The commands are:

STATUS — Display the current status of the transmitter, including the attennuation setting.

ABORT — Terminate the TXSTAT task.

TSGINT — Test the TSG interrupt and associated software. TSGINT enters an endless loop that starts the TSG by sending it the specified command byte via Supervisor Call 1 (SVC1), which invokes the TSG driver at the system level. After starting the TSG, TSGINT reports errors, if there are any. In case the TSG start resulted in an FEP start, TSGINT clears the FEP DATA READY signal before waiting for the TSG DONE interrupt. If there are no errors sensed by the SVC1 TSG wait call, the loop counter is displayed in the hex display panel lights and TSGINT loops again. The SVC1 TSG start and wait calls are identical to those made by SOUNDER; TSGINT issues no calls to the FEP or EKO drivers. The TSGINT loop may be stopped at any time by typing (cr).

TSGINT recognizes the following commands:

RUN — Run the TSGINT loop using command byte AØ, which selects the TSG 2Ø-ms interrupt and NOP scenario Ø, which does not start the FEP.

RUN nn — Run the loop using TSG command byte nn. The value of nn is rejected unless the high order bit is on, indicating a timing sequence scenario request. If the high order bit is off, the command byte is interpreted as a TSG read/write request.

(cr) — Stop the loop. If a TSGINT loop is in progress, it is stopped; otherwise, the (cr) is ignored.

ABORT — Terminate the TSCINT task.

PIPLN — Test the basic Interdata-TSG-FEP-EKO loop and associated software. PIPLN enters an endless loop that is a simplified version of SOUNDER's pipeline, with an identical sequence to the main events:

```
Start TSG
delay
Wait for EKO DONE
delay
Wait for FEP DATA READY
delay
Start FEP
delay
Start EKO
delay
Wait for TSG DONE
delay
```

The delays are operator-selectable and load the loop to simulate time loading in the HF Radar software. The device starts and waits are made with SVC 1 calls, identical to SOUNDER's. SOUNDER and PIPLN SVC 1 calls invoke the same TSG, FEP, and EKO drivers at the Operating System level. Start and wait errors are reported as they occur. The loop counter is displayed in the hex display panel lights. The PIPLN loop may be stopped at any time by typing (cr).

The commands are:

RUN — Run the PIPLN loop using TSG command byte A5, which selects the TSG 2Ø-ms interrupt and test scenario 5, which puts a second calibration pulse into the data.

RUN nn — Run the loop using TSG command byte nn. The value of nn is rejected if it invokes a TSG read/write request (high order bit Ø).

(cr) — Stop the loop, if in progress, else ignore the (cr).

SETD — Specify the loop delays (discussed above) in decimal, integer milliseconds. The delays are only approximate. The total of the six delays should not exceed the value of the TSG interrupt selected (normally 2Ø ms). The operator is led through the DELAYS command; delays are associated with the specific start or wait that they follow.

DELAYS — Display the currently specified delays.

ABORT — Terminate the PIPLN task.

SNRIO — Display the current TSG scenarios stored in TSGMEM.SND. Scenarios are displayed in time scale. Major tick marks are 1 millisecond apart; minor tick marks are 1ØØ microseconds apart. (Future plans call for displaying TSG scenarios by state change and for composing and displaying new TSG scenarios.)

At present, TSG scenarios are entered or modified by reading the first sector of disc file TSGMEM.SND with utility task DUTIL (Section 10.4), modifying the appropriate bytes, and re-writing the sector to disc.

SNRIO commands are:

CURRENT — Display one of TSG scenarios currently in TSG page zero. Select a scenario (Ø-7) in response to the SNRIO request.

TEST — Display test (staircase) scenario, which demonstrates that SNRIO works.

ABORT — Terminate the SNRIO task.

DATAFMT

Thoroughly check the format of the EKO data from the FEP (Section 3.4) during a sounding and list bad buffers at the console. DATAFMT answers all queues from SOUNDER, but does not pass them on. Therefore, MANAGER and PICKER are idle. DATAFMT offers the option at the console of displaying the first bad buffer encountered or the number of bad buffers during the TEST sounding. After DATAFMT has started and the COUNSEL prompt is seen, type GO, KRUN, or VUF to start TEST sounding. If sounding stops prematurely (watch the lights), type ABORT. The bad buffer will be displayed if that option was chosen. (At the end of a sounding, SOUNDER puts in maximum RF, IF, and calibration attenuations, which turn on all LED panel lights for these devices.)

There are no commands for DATAFMT.

CKDUP

Check for successive duplicate EKO buffers from the FEP. Duplicate buffers are an indication that the FEP input/output memories are not being switched. CKDUP handles SOUNDER queues like DATAFMT.

After CKDUP has started and the COUNSEL prompt has been seen, type GO, KRUN, or VUF to start TEST sounding. When ABORT is typed to COUNSEL during or after the sounding, CKDUP reports the number of duplicated buffers detected during the run.

There are no commands for CKDUP.

TAPEKO

Copy the EKO data as received from the FEP to magnetic tape. When first started with the TEST TAPEKO command, TAPEKO writes the first of four sounding information records to tape. The first tape record contains the current TSG scenarios, or TSG page Ø, as read from disc file SYS1:TSGMEM.SND. TAPEKO next advises the operator to select one of the slower TSG sounding rates, and returns control of the console to the COUNSEL task. When the COUNSEL prompt appears, select the sounding rate with the TSGC command. TSGC CN results in pulsing 1Ø times per second; TSGC EN is one pulse per second, where N is the desired TSG scenario number.

A slower sounding rate is required to allow time for the tape write. For FEP programs and TSG scenarios that result in small EKO data transfers, and hence small tape records, a pulse repetition frequency of 1Ø pps is

adequate to avoid data over-run; large EKO buffers require the 1 pps sounding rate.

When TAPEKO receives the SØ4 Q-block from SOUNDER indicating that GO, KRUN, or VUF has been typed (Chapter 8), TAPEKO writes the rest of the sounding information records to tape. Record 2 is the current ICT from SOUNDER (Section 7.1); record 3 is the K-mode sounding table (Section 7.4) from SOUNDER. Record 4 is a duplicate of record 3, for now. (Having 4 sounding information records ahead of the data makes tape positioning for tasks EXPCT and PLOTEKO simple, since there are currently 4 pulses per pulse set.)

Data records follow the 4 control records; one data record is written for each SØ2 Q-block from SOUNDER. A data record consists of a PCT (Section 7.2) and its associated EKO buffer (Section 3.4). The data records written to tape vary in length as the volume of data from the FEP varies.

At sounding end, indicated by the SØ6 Q-block from SOUNDER, TAPEKO writes two EOF marks to tape and backspaces over the second EOF, leaving the tape positioned for another TAPEKO sounding. Thus, a TAPEKO tape contains one sounding per file with double EOF's indicating the end of information.

TAPEKO does not allow for multiple reels. If a tape error is sensed in writing the first control record, TAPEKO is terminated; tape errors once the sounding has started do not terminate TAPEKO or the sounding, but do prevent any further writes to tape. A message is displayed at sounding end if the tape is not good. The sounding can be aborted at any time.

Since the magnetic tape unit is attached to MANAGER during normal sounding, it is the operator's responsibility to avoid conflict between MANAGER and TAPEKO (see the TAPE command, Chapter 8) and to mount and dismount tape reels as needed.

TAPEKO output tapes serve as input for TPRUN soundings (Chapter 8) and can be examined using TEST tasks EXPCT and PLOTEKO.

TAPEKO has no command input. Soundings are set up, started, and aborted through advice to SOUNDER.

EXPCT — Examine Pulse Configuration Tables (PCT's) for the last four pulses. If SOUNDER was last run in TEST mode, and if there was no data over-run during TEST sounding, the last 4 PCT's will represent one pulse set. If such is the case, the displayed PULSE numbers will be four consecutive integers beginning with 4*(n-1) + 1, where n is the pulse set number.

Tapes written by TEST task TAPEKO (above) may also serve as input to EXPCT. If the TAPE command is given to EXPCT (see below), the four PCT's displayed will always represent one pulse set.

EXPCT commands are:

CURRENT — Display the current four PCT's. The PCT's currently in memory may be the last four left from the test sounding or may have been read from a TAPEKO tape.

TAPE — Display the next set of PCT's on magnetic tape. Four PCT/EKO buffer records are read into SOUNDER's EKO buffers and the PCT's are displayed. The tape used must have been written by TEST task TAPEKO.

TAPE f — Display the set of PCT's from tape for frequency f, expressed in decimal kHz. If frequency f was not used in sounding, the PCT's for the next higher frequency are displayed. The tape is scanned forward for the requested frequency.

RW — Rewind magnetic tape.

FF — Forward space one tape file to next sounding.

ABORT — Terminate EXPCT task.

PLOTEKO — Plot and list EKO data from the FEP. The last four SOUNDER EKO buffers are available for plotting and listing. If SOUNDER was last run in TEST mode, and if there was no data over-run during the TEST sounding, the last four EKO buffers will represent one pulse set (see EXPCT).

Tapes written by TEST task TAPEKO (above) may also be used as input to PLOTEKO. If the TAPE command is given to PLOTEKO (see below), the four EKO buffers will always represent one pulse set

PLOTEKO accepts the following commands:

PLOT n Plot EKO buffer n, n = 1 through 4. Receiver channel 1 amplitude is plotted above the horizontal center line, channel 2, below. Range (virtual height or time) increases from left to right. Range Ø is the start of the FEP WRite ENaBLe window (see SNRIO display), the time at which the FEP, under command of the TSG, begins listening for returns. Full scale range (at extreme right) represents 5.12 ms after the beginning of the FEP WRite ENaBle. Thus, about 768 km of range are represented on the X-axis. The computed amplitude (see AMP, BARXY) is plotted vertically and is scaled against the largest amplitude found in all groups of both channels. Data groups are delineated in range by tick marks extending up or down from the X-axis for receiver channels 1 or 2, respectively, and are numbered left to right from Ø. Data groups (the "spikes" plotted) may or may not be true echoes; PICKER's job is to separate the echoes from the noise. PLOTEKO plots all data that passes the FEP thresholding criteria.

DETAIL n Plot individual groups of EKO buffer n (1-4) if within 4Ø FEP range units of each other. One FEP range unit equals 5 microseconds, or about 75Ø m. The channel 1 group is plotted to the left of the vertical center line, the channel 2 group to the right. Straight lines connect the amplitudes of the individual X-Y pairs taken at FEP ADC sampling intervals (of 1Ø microseconds) and phase is plotted with phi's as plotting symbols. Phase ranges between -pi and +pi radians (as labeled for each channel) with a center line at Ø phase for each channel. Range increases from bottom to top. Group Ø, the calibration pulse. always occurs at range Ø, and is plotted first by DETAIL. If the starting ranges for the groups plotted side-by-side are not identical, the plots will be offset so that range is mirrored left to right.

DETAIL n,G1
Plot individual data groups G1 of EKO buffer n (1-4) for both receiver channels. (The data groups, as displayed by the PLOT command, are numbered from left to right, as range increases, counting from Ø, the calibration pulse.) If a group number G1 exists for either or both channels, it is plotted.

DETAIL n,G1,G2
Plot group G1 of channel 1 with a group G2 of channel 2, both from EKO buffer n. Data group numbering is described above.

DETAIL plots are scaled vertically (range) to fit the screen; if the point density is high, a horizontal bar is plotted for phase, rather than the phi symbol. The ordinates are scaled against the largest amplitude found in all groups of both channels.

CONT — Continue with DETAIL plots of single groups from EKO buffer selected by the DETAIL command. CONT is ignored if all groups have been plotted.

LIST n — List EKO buffer (1 - 4) contents at the console in signed decimal.

AMP — Compute amplitude as SQRT(X**2 + Y**2) (FORTRAN notation). This is the default computation and remains in effect until BARXY is typed.

BARXY — Compute amplitude as ABS(X) + ABS(Y). This is the amplitude approximation used by PICKER. BARXY remains in effect until AMP is typed.

TAPE — Read the next set (4) of PCT/EKO buffer records from magnetic tape written by TEST task TAPEKO. The sounding data is read into SOUNDER's EKO buffers and is available for plotting or listing.

TAPE f — Scan forward until the pulse set data for frequency f or the next higher frequency is found and read the set into SOUNDER's EKO buffers.

RW — Rewind the magnetic tape.

FF — Forward space one tape file to the next sounding.

ABORT — Terminate PLOTEKO task.

EXOUT — Examine sounding output data. These data are the output from the PICKER task, written to the disc by MANAGER.

The data from a given pulse set are displayed on the console as described in the following paragraphs.

The four PCT's for the pulse set are listed in order in their compacted, hexadecimal format as described in Section 7.2.

If PICKER, given its range gating and KEEP criteria, has found time-coincident echoes in the EKO data from the FEP for the pulse set, EXOUT displays the average range in microseconds of the echoes, together with the corresponding virtual height in kilometers.

Next, EXOUT displays the quadrature components of the peaks of the time-coincident echoes, as derived by PICKER, together with true amplitude and phase (degrees), as computed by subroutine PHAMP (Sect. 2.11).

These data are listed in 8 columns that alternate channels 1 and 2 for the four pulses of the pulse set. The default order is shown below. (See commands PSEQ and ANTENNAS, Chapter 8.) The columns are numbered from left to right. F is the base frequency of the pulse set; deltaF is set by command DELF and its default value is 8 kHz.

COL.	PULSE	CHAN.	FREQUENCY	ANTENNA
1	1	1	F	N
2	1	2	F	S
3	2	1	F	E
4	2	2	F	W
5	3	1	F + deltaF	E
6	3	2	F + deltaF	W
7	4	1	F + deltaF	N
8	4	2	F + deltaF	S

The EXOUT task commands are described below.

GPION n — Get data set for previous sounding n. The data set number, n, is described under GRAPHER command GPION in Chapter 8.

LIST — Display the first/next pulse set data or data record of attached data set n.

LIST F — Display the data record for frequency F (kHz) or the next higher frequency, if frequency F was not used in the sounding. The data set is scanned forward to find the appropriate data record.

LIST ALL — Display all data records from the current position in the data set to the end.

(cr) — If a data set has been selected with the GPION command above, a carriage return is equivalent to the LIST command; otherwise, it is ignored.

ABORT — Terminate the EXOUT task.

SCHED — Construct a schedule of sounding commands. Task SCHED enables the operator/experimenter to pre-select a sequence of HF Radar sounding commands, to intermix sounding modes, to nest repeated sequences, and to specify optional sequences of commands in case command errors occur. Schedules of commands saved on disc by SCHED can subsequently be read and executed by an SEL-written ANALYSER task supplied with Product One.

SEL's scheduling version of ANALYSER (referred to below as the scheduler) assumes the role of COUNSEL in the HF Radar software to the extent of sending SOUNDER, PICKER, MANAGER, and ANALYSER itself CØ2 advice Q-blocks. The information in bytes 4-7 of the scheduler's CØ2 Q-block (Sect. 7.7.1) is obtained from the file of scheduled commands written to disc by SCHED.

SCHED builds the file as follows: when the operator/-experimenter enters an HF Radar task command to be scheduled, SCHED checks its validity by scanning the tasks' command mnemonic tables. (Essentially the same scan is performed by COUNSEL when a command is typed at the console. ET.SCAN (Sect. 7.3) contains the address of the mnemonic tables address list.) If SCHED finds the command to be valid, it saves the information needed to construct a CØ2 Q-block, the ASCII text of the command,

and the index of the task to which the address of the CØ2 Q-block will be sent by the scheduler. The command information is thus accumulated and is written to disc by SCHED command SSAVE (below).

It should be noted that only commands to SOUNDER, PICKER, MANAGER, and ANALYSER can be scheduled. COUNSEL's commands are not directly pointed to by ET.SCAN; GRAPHER is replaced temporarily in partition 5 by SCHED, so its command list is not available to be scanned. Furthermore, SOUNDER commands TEST and TPRUN, and ANALYSER commands SRUN and SLIST, are explicitly excluded. However, all commands essential to unattended, scheduled sounding can be used. In addition, since task commands are queued whether typed at the console or sent under program control. any appropriate command can be given at the console even though the scheduler is active. Thus a schedule can be adjusted, or even overridden, if desired, provided the rules concerning command usage, discussed in Chapter 8, are observed.

SCHED accepts the time-specific forms of GO, KRUN, and VUF, as discussed in Chapter 8. In addition, to provide more flexible timing, SCHED allows for the construction of a table of starting times. (See command TIME, below.) The times-table entries are numbered from 1 to 6Ø and are referenced in order each time an asterisk (*) is seen as the argument of the GO or KRUN command by the scheduler. For example, if the first times-table entry is "3:17" and the scheduler encounters GO* or KRUN*, "3:17" is substituted for the *, and the sounding starts 3 minutes and 17 seconds past the current hour. The second times-table entry will be referenced by the next GO* or KRUN*. The end of the table is established by SCHED to be the highest numbered entry, n, set by the TIME n command. When the scheduler senses the end of the table, it begins again with entry 1. All entries 1 through n need not be set. If a command, GO*, causes reference to an entry that has not been set, it is as if the command GO(cr) had been given.

All standard forms of the VUF command can be scheduled; at present, however, VUFn* is not implemented

After the scheduler has sent a CØ2 queue to SOUNDER initiating a sounding, that is a GO, KRUN, or VUF command, it issues no further commands until the sounding and all

post-sounding activities have been completed.

The SCHED task commands are described below.

BUILD — Enter the schedule-building mode. After BUILD has been issued, SCHED accepts valid task commands as well as any of the schedule control commands, which are described below. The user is advised to use the most compressed form possible of a given command to avoid overflowing SCHED's text storage area, which is 510 bytes long. (For example, KE8 is preferrred over KEEP 8.) An attempt to overcome this limitation will be made in the future; for now, an overflow produces an error message.

SSAVE xxx — Write the schedule just built to disc file SYSC:SSCHED.xxx for use by the scheduling version of ANALYSER. This file contains the information that the scheduler needs for composing C02 Q-blocks, together with the text of the schedule, which can be displayed by issuing command SLIST (Chapter 8). (See also command SRUN xxx, Chapter 8.)

TSAVE name.xxx — Save the text of the schedule on disc file SYSC:name.xxx. This file can be modified with EDIT, the system editor, and can be the argument for a subsequent GET command (below).

ABORT — Terminate the SCHED task.

After the BUILD mode has been entered, SCHED accepts the following schedule control commands:

REPEAT n — Repeat the sequence of task commands that follows the REP command n times, where n > 1. A REP loop is terminated by ENDR or ENDB (below). Loops may be nested 10 deep.

ENDR — Terminate the innermost REP loop. Commands REP and ENDR are roughly equivalent to the FORTRAN statements DO and CONTINUE.

IFERR — Begin the specification of a sequence of task commands to be executed in case the command preceding IFE resulted in an error. The IFE command is rejected if given within the body of an IFE sequence. An IFE sequence is terminated by ENDE (below) or, by implication, ENDR or ENDB (below).

ENDE — End an IFE sequence. The scheduler skips over all commands between IFE and ENDE unless the command preceding IFE resulted in an error.

ENDB — End the BUILD mode. The schedule built can now be saved with SSAVE or TSAVE (above), or both, and a new BUILD can begin. From the scheduler's viewpoint, single or nested REP loops are terminated by ENDB as is an IFERR sequence. ENDB is also an implied SABORT (Chapter 8), although the scheduler will satisfy any and all REP loops terminated by ENDB before stopping.

TIME n mm[:ss]

Specify time argument for a GO* or KRUN* command. The first argument, n, is the times-table entry number and is in the range 1-60. The time argument must at least specify minutes past the hour and may specify minutes and seconds. The TIME command can be issued at any time in the form given above. Out of the BUILD mode, command TIME with no arguments lists the times-table to the console.

LIST — List the partial or completed schedule to the console. The LIST command can be given at any time.

GET name.xxx

Get BUILD-mode commands from SYSC:name.xxx. The file can be the output from a previous TSAVE command (above), or from the system editor, EDIT. GET files may contain any BUILD-mode command except GET. Because a second BUILD command without an intervening ENDB is ignored, GET files to be concatenated can each begin with BUILD. An ENDB encountered in a GET file terminates the BUILD mode; if no ENDB is seen, SCHED remains in the BUILD mode and accepts further command input from the console. Thus, BUILD-mode commands from the console can be concatenated with those from disc files.

Examples — Sample schedules are described below.

For clarity, commands in the examples are not compressed, and indeed need not be for short schedules. Note that SCHED checks the validity of only the task command, not the command argument(s). A command accepted by SCHED could be rejected by the task to which it is ultimately sent if the command argument string is inappropriate. (See Chapter 8 for more complete descriptions of the task commands.)

Example 1 initializes the high power transmitter, then runs an ionogram to establish the receiver gain profile tables for the AGC (Sects. 2.12.7 and 2.10.2). Example 1 demonstrates the use of error option sequences.

COMMAND	DESCRIPTION
BUILD	Enter SCHED build mode.
SFRD 15ØØ	Set default starting frequency.
EFRD 15ØØØ	Set default ending frequency.
DFIX 1	Set frequency index increment.
TXON	Begin transmitter filament warm-up.
GO	Run ionogram to allow time for warm-up.
ABORT	Put SOUNDER in idle mode.
HVON	Turn on transmitter high voltage.
IFERR	Begin first error option sequence.
GO	Allow more time for filament warm-up.
ABORT	Put SOUNDER in idle mode.
HVON	Try HVON again.
ENDE	End first error sequence.
IFERR	Begin error sequence for second HVON.
SABORT	Abort schedule; second HVON failed.
ENDE	End second error sequence.
DLOF	Select antenna load.
GO	Run ionogram with HV on.
ABORT	Put SOUNDER in idle mode.
ENDB	End build mode and schedule (implied SABORT).

The first ionogram should allow enough time for the 9Ø-second transmitter filament warm-up between the TXON and the HVON commands. If not, the HVON command will produce an error; the first error option sequence runs another ionogram and tries again to turn on the high voltage. If the second HVON command fails, the schedule is aborted. If either HVON command is executed without error, the scheduler skips the subsequent error option commands and runs a final I-mode sounding with the high voltage on and the antenna load selected for the transmitter. Such a sounding is required to initialize the receiver gain profile tables.

Example 2 assumes that all initialization has been done for I- and K-mode sounding in order to demonstrate simply the use of REPEAT and ENDR. Example 2 produces an ionogram followed by five consecutive K-mode soundings; this sequence is repeated three times.

COMMAND	DESCRIPTION
BUILD	Enter build mode.
REPEAT 3	Begin outer loop.
GO	Run I-mode sounding.
ABORT	Idle SOUNDER.
REPEAT 5	Begin inner loop.
KRUN	Run K-mode sounding.
ABORT	Idle SOUNDER.
ENDR	End inner loop.
ENDR	End outer loop.
ENDB	End build mode and schedule (implied SABORT).

In this case, one or both of the ENDR's may be omitted, since ENDB implies ENDR, if ENDR is missing. The following sequence is equivalent to that above; the commands are compressed.

```
BU
REP3
GO
AB
REP5
KR
AB
ENDB
```

Example 3 demonstrates the use of time-specific sounding starts. It is designed to run three "standard" I-mode soundings, from 1.5 through 15 MHz, on each quarter hour; to run a K-mode sounding on 4 frequencies five minutes past each quarter hour, employing a TSG scenario and a FEP program designed for sounding the upper D and the E regions; and to run three upper D- and E-region I-mode soundings, from 1.5 through 3.5 MHz, ten minutes past each quarter hour. This quarter-hourly sequence is repeated for eight hours.

The following explanation assumes that command SRUN xxx,7 is given to the scheduler (ANALYSER), so that the first time-specific I-mode sounding starts at 30 minutes past the hour. (See command SRUN, Chapter 8.)

COMMAND	DESCRIPTION
BUILD	Enter build mode.
SFRD 15ØØ	Set default I-mode starting frequency.
GO	Run I-mode sounding; initialize for AGC.
ABORT	Idle SOUNDER.
FØ 17ØØ	Select K-mode frequency Ø (Sect. 7.4).
F1 22ØØ	Select K-mode frequency 1.
F2 26ØØ	Select K-mode frequency 2.
F3 32ØØ	Select K-mode frequency 3.
RFØ 11Ø	Ignore echoes above 11Ø km for frequency Ø.
RF1 11Ø	Ignore echoes above 11Ø km for frequency 1.
RF2 11Ø	Ignore echoes above 11Ø km for frequency 2.
RF3 11Ø	Ignore echoes above 11Ø km for frequency 3.
KSET 25Ø	Sound on each selected K-mode frequency 25Ø times.
KFIL 1	Select 1 NOP fill period between K-mode pulse sets.
TAPE	Save sounding data on magtape.
FORMAT 1	Select compact tape format.
REPEAT 8	Begin hourly sequence loop.
REPEAT 4	Begin quarter-hourly sequence loop.
DFIX 4	Set frequency index increment for the next 3 "standard" ionograms.
EFRD 15ØØØ	Set default ending frequency for next 3 ionograms.
RANGE	Remove PICKER's echo range testing restrictions.
GO*	Run "standard" I-mode sounding at 3Ø, 45, 6Ø, 15, 3Ø,... minutes past start of current hour.
ABORT	Idle SOUNDER task.
REPEAT 2	Begin loop to run 2 more I-mode soundings.
GO	Run next I-mode sounding as soon as possible.
ABORT	Idle SOUNDER.
ENDR	End "standard" I-mode sounding loop.
TSGC A7	Select TSG scenario 7 to run every 2Ø ms.
BOOT FEP,NOT	Load FEP with program that does no thresholding.

COMMAND	DESCRIPTION
KRUN*	Run K-mode sounding at 35, 50, 5, 20, 35,... minutes past start of current hour.
ABORT	Idle SOUNDER.
EFRD 3500	Set default I-mode ending frequency for next 3 soundings.
DFIX 1	Set frequency index increment for next 3 ionograms.
RANGE 3	Set PICKER's echo search I-mode sounding to the range of 50-110 km.
GO*	Run I-mode sounding at 40, 55, 10, 25, 40... minutes past start of current hour.
ABORT	Idle SOUNDER.
REPEAT 2	Begin loop to run 2 more I-mode soundings.
GO	Run next I-mode sounding as soon as possible.
ABORT	Idle SOUNDER.
ENDR	End upper D- & E-region I-mode sounding loop.
TSGC A4	Select default TSG scenario to run every 20 ms.
BOOT FEP	Load FEP with standard (default) thresholding program.
ENDR	End quarter-hourly sequence loop.
ENDR	End hourly sequence loop.
TIME1 60	TIME1-TIME12 set up the times-table.
TIME2 5	
TIME3 10	
TIME4 15	
TIME5 20	
TIME6 25	
TIME7 30	
TIME8 35	
TIME9 40	
TIME10 45	
TIME11 50	
TIME12 55	
ENDB	End build mode and schedule (implied SABORT).

Simple schedules that do not mix sounding modes are easily set up and do not require the use of the scheduler. For example, to produce an ionogram on each quarter hour for an 8-hour period, type the following commands at the console to COUNSEL:

COMMAND	DESCRIPTION
NIPH 4	Run 4 I-mode soundings per hour.
NI 32	Stop after 32 soundings or 8 hours.
GO 60	Start at the beginning of the next hour.

Notice that only one GO is required for all 32 soundings, and that the GO is not followed by ABORT. Repeated soundings under the control of the scheduler, however, require that SOUNDER be put in the idle mode with ABORT in order to accept a new GO, KRUN, or VUF.

DIO — Set and check the radar receiver instruments on the DIO bus. A...A is the argument of a command.

The DIO commands are:

1ANT AA — Select channel 1 antenna, 1-4.

2ANT AA — Select channel 2 antenna, 1-4.

1RF AA — Select channel 1 RF attenuation, 0-60 dB.

2RF AA — Select channel 2 RF attenuation, 0-60 dB.

1IF AA — Select channel 1 IF attenuation, 0-60 dB.

2IF AA — Select channel 2 IF attenuation, 0-60 dB.

1FIL AA — Select channel 1 bandwidth filter, 1, 2, 10, or 30 kHz.

2FIL AA — Select channel 2 bandwidth filter, 1, 3, 10, or 30 kHz.

CAL AA — Select calibration attenuation, 0-120 dB.

FREQ AAAAA.A
Select synthesizer frequency in kHz.

DIO DD,AA — Command DIO device DD (hex) with hex value AA.

If any of the above commands are given without an argument AA, the current setting of the device is displayed.

HSK CC — Read Housekeeping ADC channel CC and display the reading in volts and in ADC counts (hex).

LOCAL — During a sounding, SOUNDER puts the receiver instruments in remote control mode, disabling the front panel manual controls. LOCAL removes the remote lock and enables the LOCAL CONTROL push switch, which, when depressed, enables the front panel controls.

RWCK — Check the DIO interface read-write/read-only mode command. Both states are tested 10000 times and failures reported.

STATUS — Display the current receiver instrument settings.

ABORT — Terminate the DIO task.

SCLOOP — Set up read/write loop for checking the DIO with logic analyzer or oscilloscope. Loops can mix reads and writes or can be declared to be either exclusively read or write from any point of the loop to loop end. (SCLOOP will be made a part of TEST task DIO in the future.)

SCLOOP commands are:

LOOP — Set up a loop. The program displays the command number (1-100) and requests the command type, unless the loop has been previously declared read or write (RL or WL, below). Valid responses to the "READ/WRITE:" command type request are:

- R — Make this command a read.
- W — Make this command a write.
- RL — Make this and all following commands reads.
- WL — Make this and all following commands writes.
- (cr) — All commands for the loop have been entered. All that remains to be specified is the loop count.

After the operator has responded to the "READ/WRITE:" request with other than (cr), or if RL or WL has previously been typed, SCLOOP requests the DIO address that is to be read from or written to. The response is any hex number from 0 to 3F. If a number greater than 3F is entered, only the low order 6 bits will be used. (Not all addresses from 0 to 3F are used by the HF Radar hardware.) If RL or WL has previously been typed, a (cr) typed in response to "DIO ADDRESS:" terminates the

read/write command input and the loop count is requested.

If the current command is a write, SCLOOP next requests "DATA:". The response is any hex number from Ø through FF. If a number greater than FF is typed, only the low order 8 bits will be written to the DIO address specified.

After the write data or the read DIO address has been entered, SCLOOP requests "DELAY:". The response is a decimal number Ø-32767 and is roughly the number of milliseconds to delay after the current command has been executed in the loop. The delay timing loop is not accurate; an attempt to "time" it will be made in the future. A delay request of Ø results in a delay of about 2Ø microseconds.

When the operator terminates command input with (cr) in response to 'READ/WRITE:" or to "DIO ADDRESS:", or if 1ØØ commands have been entered, SCLOOP requests "LOOP COUNT:", the number of times each command in the loop is to be executed. The response is Ø-32767 (decimal) or (cr) only, if looping is to continue until interrupted (below).

After the loop count is speicifed, the DIO read/write loop is executed. A loop may run to completion, given a finite loop count, or it may be interrupted at any time with a single (cr) typed at the console. An interrupted loop may be re-started with another (cr). All specified loop parameters remain intact until a new LOOP command is typed. Thus, a loop, once set up, may be started and stopped at will by typing a series of (cr)'s.

DATA — Display the data read, if any, the last time through the loop. This command cannot be given unless the loop has been interrupted. After the read data has been displayed, the interrupted loop may be restarted with (cr).

ABORT — Terminate SCLOOP task.

QSTOP — Report all messages (queues) that SOUNDER sends to PICKER during a sounding. The first message should be SØ4, meaning that sounding has started. SØ2 messages report that new data from the FEP is available to PICKER. There should be four SØ2 messages for each frequency (one pulse set). For K-mode, it should be noted that the SØ2 Q-block counters are 8 bits wide, so the counts are modulo 256. The frequency (kHz) and the frequency index (FIX) are

displayed. The last message should be SØ6, meaning that sounding has ended. QSTOP verifies that SOUNDER is properly queueing PICKER, and is also useful for verifying that protected frequency bands are being respected. After QSTOP has started and the COUNSEL prompt is seen, type GO, KRUN, or VUF to start TEST sounding. When ABORT is typed to COUNSEL during or after sounding, QSTOP displays all messages (queues) received from SOUNDER.

There is no command input.

AGC — Display the receiver gain tables used by the Automatic Gain Control (AGC) system in SOUNDER.

The AGC commands are:

- CURRENT — Display the current tables in SOUNDER.
- RGAIN — Display the tables for a previous sounding. AGC asks for the number of a sounding whose tables were saved on disc file SYS1:RGAIN.SAV by SOUNDER command RGAIN (see Chapter 8).
- ABORT — Terminate AGC task.

HKADC — Exercise the Housekeeping ADC unit.

The HKADC commands are:

- SELCH — Select a channel, Ø-63. (decimal).
- CHAN — Display current channel number.
- DATA — Read current channel.
- RDADC — Read range of channels (decimal). For example, 1Ø.,15. reads channels 1Ø-15 inclusively.
- CHLOOP — Read specified list of channels. Each channel is read 1Ø times (round robin) following a delay in ms specified in response to the task's request.
- ABORT — Terminate HKADC task.

All HKADC readings are displayed in volts and ADC counts (hex). Future plans call for displaying readings in approporiate engineering units with channel definitions and acceptable limits included.

DUMP — For installations with a TTY model 40 line printer attached to PASLA address 87 (hex), this task will print selected segments of memory. In the absence of a Model 40, this task does nothing.

DUMP requests start through end memory addresses to be dumped to the printer. If the response is ABORT, DUMP is terminated. Otherwise, the specified block of memory is written in ASCII hex to the TTY model 40. If the printer is not ready, DUMP will wait until it is (. . . and wait . . . and wait . . .).

10.4 Program DUTIL Access and Status [D.C.W.]

Disc utility program DUTIL enables the user to read, search, modify, and write disc sectors, and to examine the names of files, file indexes, and volume directories on the disc.

DUTIL bypasses all system protections and must be used with extreme caution.

The following commands provide access to DUTIL and its status.

DUTIL — Run program DUTIL. This command invokes file DUTIL.CSS, which sets partitions, loads, and starts DUTIL. If a message reveals that file DUTIL.CSS is unavailable, type:

```
SE PA 1/F000
SE PA 1/7000
LO DUTIL
T DUTIL
ST
```

All numerical input to DUTIL is hexadecimal; leading zeroes are not needed. All values displayed by DUTIL are hexadecimal. Only devices DSC1 (top or removable disc) or DSC2 (bottom or fixed disc) may be specified.

*[COMMENTS] — DUTIL ignores command lines beginning with an asterisk. Comments and documentation of the run may be entered in this way.

CURRENT — Displays current device, sector, volume name, volume directory start sector, and volume bit map start sector.

DEVICE — Specify the current device, DSC1 or DSC2, in response to DUTIL's request. The device name must not appear with the command.

END — Exit from DUTIL.

PREAMBLE — For 7/16's only; destroys all current data. DUTIL formats the device (DSC1 or DSC2) specified in response to its request by writing each sector as 256 bytes of test pattern X'A5' with its four-byte preamble. The device formatted becomes the current device.

PAUSE — Send PAUSE to the system command processor. Re-enter DUTIL by typing:

CONTINUE

10.5. DUTIL Sector Commands [D.C.W.]

The sector commands enable the user to read, search, modify, and write sectors on the disc specified by DEVICE (10.4).

The current sector is the last sector read, written, transferred, or zeroed, i.e., the sector currently in the memory buffer.

DISC — Display the sectors used and the sectors free on the current device as indicated by the current disc bit map. DISC also displays the size of the largest contiguous block of free sectors.

EXAMINE — Display the current sector.

MODIFY DD,H1[,H2,...,HN]
Modify the contents of the sector currently in the memory buffer. DD is the byte displacement at which the modification begins. Subsequent values of one or two hexadecimal digits are put, byte by byte, into contiguous buffer byte locations. Up to 16 bytes may be modified per command. When the modification is complete, the current buffer contents are displayed. The buffer must then be written on the disc sector.

READ HHHH Read into the buffer and display sector HHHH.

SEARCH HHH1[,HHH2,...,HHHN]
Search the sector currently in the buffer for the specified half-word value(s). Up to 8 values may be specified. When a match is found, "(+DD)=HHHH" is displayed, where DD is the byte displacement of the matching half-word in the sector, and HHHH is the contents of the word. DD is always even. If no matching value is found, there is no display.

If SEARCH without a value is typed, the entire current sector is displayed sequentially, one half-word per line.

WRITE Write the buffer on the current sector, if any.

XFER HHHH Transfer (write) the current buffer to sector HHHH of the disc. HHHH becomes the current sector.

ZERO HHHH Write zeroes in sector HHHH, which becomes the current sector.

10.6. DUTIL File Commands [D.C.W.]

The file commands enable the user to obtain file name and file parameters. In addtion, the hexadecimal editor provides means of changing words in files, and NEXT finds file index and directory blocks.

FILES List all files residing on the current device.

FILES NAME.EXT
List the parameters of the specified file.

FILES NAME.- List all files with the specified name and all existing extensions.

FILES -.EXT List all files with the given extension.

FILES - List all files with null extensions.

FILES -.- Same as FILES alone.

HEXEDIT NAME.EXT
Enable hexadecimal edit of the specified file, which must be contiguous and must reside on the current device.

After the edit mode is entered, type:

HHHH

where HHHH is the byte address of the half-word cell to be modified. DUTIL responds by displaying:

+HHHH/ CCCC

where CCCC is the current contents of cell HHHH. To modify the cell, type

NNNN (1)

where NNNN is the desired new contents of cell HHHH. To leave cell HHHH as it is, type only a carriage return:

(cr) (2)

To open the next half-word cell for editing (byte address HHHH + 2), type line feed followed by carriage return instead of a carraige return at (1) or (2):

(lf)(cr)

DUTIL will indicate when a modified sector has been written on the disc. At any time (cr)(cr) is typed, the edit mode and HEXEDIT terminate.

NEXT Read consecutive sectors, starting with the current sector, until a file index or directory block is found, and display the block sector. Such a block is one that has a first word of zero, which may be a binary word rather than a block. This routine was written by J. R. Winkelman to assist in the repair of directory strings (see section 10.7).

10.7. Notes on Volume Directories [D.C.W]

Each 5-Mbyte disc has a volume directory (VD) that contains the information required by the file manager (FMGR) of the operating system (OS) to access files on the disc. The VD itself is contructed somewhat like an indexed file in that each VD block contains the pointer to, or the sector number of, the next VD block in the chain. Sector number zero points to the first VD block, which is always 3Ø. The last VD block in the chain points to zero. The VD is initialized to use the following sectors: Ø through 5A in steps of 6, with 5A pointing to 31; 31 through 5B in steps of 6; 32 through 5C by 6; 33 through 5D by 6; 34 through 5E by 6; 35 through 5F by 6, with 5F pointing to Ø.

Should the VD become full, FMGR assigns additional blocks, as needed, and appends them to the end of the VD chain. These blocks are assigned as available from the disc bit map, so their sector numbers are unpredictdable. For example, if the bit map reveals that sector number 3BØE is the next availabe block, the end of the chain will have 5F pointing to 3BØE, which in turn points to Ø. In this way the chain is lengthened as the number of files grows, with the last VD sector always pointing to Ø.

The first 4 bytes of a VD block contain the pointer to the next block. The first two bytes should always be Ø because the largest sector number is 4C7F (19583 decimal). Occassionally, a VD block gets written back to the disc with a non-zero value in the first two bytes. When an attempt is made to access a file beyond such a bad VD block, an EOM error results. DUTIL can be used to correct the bad sector in order to reconstruct the VD block chain. If a file known to exist cannot be accessed or listed, DUTIL's NEXT command can be used to read through the chain until the break is located and repaired.

10.8. LION and TAPIR [J.R.W]

LION (LIst IONogram) lists to the printer and as such is not of general utility. It is, however, an often-used program. LION was written to determine what is on a tape for debugging. It was not conceived to be used by anyone else. No replacement program using more proper formats has been written.

The format is mixed decimal and hex to display a PØ2 record. This is a terrible format for the ICT. LION has been expanded to print compact formats also. The formats may be interspersed on tape.

The first line of each record is a record number in decimal and the length of the tape record in bytes. The first number on the following line is the first word of the PØ2 block and is the length of usable data (normally shorter than the actual record length). It is used to determine the number of lines to print so as to suppress echoes that don't belong with this data.

The remainder of the second line of each record is the contents of the four PCT's. The first word in each PCT is decimal. The remaining lines within the record are the echoes with the X-Y pairs in hex and the range in decimal.

TAPIR is a printing program that simply lists any file, 24 hex words per line. The last line in each record has 24 words regardless of the actual length of the data.

The primary use of these two programs outside the course might be to provide the base for your own listing programs by replacing the printer subroutine with one that will run locally.

10.9. FEPMEM [D.C.W]

FEPMEM creates a disc file containing an FEP program to serve as input to FEP load routines. The output of FEPMEM is used by utility task FEPTEST and by SOUNDER to load the FEP program memory.

To date, FEP programs have been composed on an SEL in-house Tektronix 4Ø51 system and written to paper tape. The paper tape is transferred to a standard, indexed, 8Ø-character disc file in the Interdata computer using EDIT or one of several copy routines. This file serves as input to FEPMEM and thus the ASCII paper tape is converted to a usable form.

Each line of input contains 12 decimal ASCII fields separated by commas. The first field is the FEP program location number; the remaining 11 fields contain the data loaded into that FEP location. The program data fields are sent to the FEP in reverse order of their appearance in the input line. Thus, each line of input supplies the information for loading one FEP program location. The input lines may be in any order.

The output disc file is indexed and has 12 bytes per record. The first record contains information for changing the FEP threshold (see FEPTH, Chapter 8). At present, byte number 9 of both FEP locations 26 and 154 are changed when the FEP threshold is changed. Thus, the first output record contains 26, 9, 154, 9, Ø, Ø, or ØØ1A, ØØØ9, ØØ9A, ØØØ9, ØØØØ, ØØØØ in hex. The remaining output records are hex images of the decimal ASCII input lines. Input format and disc file errors are

reported.

To run the FEPMEM task, invoke the disc command file FEPMEM.CSS by typing:

```
FEPMEM INFILE,FEPMEM.xxx
```

where INFILE is the ASCII input file name and xxx is the FEPMEM output file name extension.

There is no command input.

10.10. FEPTEST [D.C.W]

FEPTEST tests the Front End Processor (FEP) and the transfer of the FEP data to the Interdata computer. To run this task, invoke disc command file FEPTEST.CSS by typing:

```
FEPTEST
```

If FEPTEST.CSS is unavailable, type:

```
SE PA 1/F000
SE PA 1/7000
LO FEPTEST
T FEPTEST
ST
```

When FEPTEST starts, it displays a message to terminate all decimal numerical input with a period. If no period is typed, the number typed is assumed to be hexadeciaml. For example, 63. = 63 decimal, 63 = 99 decimal.

FEPTEST accepts the following commands:

LOAD — Load the FEP program memory. FEPTEST requests the extension of an FEPMEM disc file residing on volume SYSC. Currently there are two FEP memory files available:

SYSC:FEPMEM.SND contains the current sounding version of the FEP programs with the default thresholding factor of 2.5 (see command FEPTH, Chapter 8).

SYSC:FEPMEM.NOT contains a version of the FEP programs that does no thresholding on the data.

Duplicate copies of these programs are kept on disc volume SYS1 for sounding.

The LOAD command resets the FEP (see RESET, below) and then loads the FEP program memory from the disc file whose extension is specified.

STTSG

Send a start to the TSG to set up a read from the FEP. The start issued to the TSG uses scenario number 4. Before the STTSG command is given, the TSG must be loaded by SOUNDER for a previous sounding or by task TSGBOOT (Section 10.11). The TSG is started in order to start the FEP, which in turn makes its FEP DATA READY signal true when its data are ready to transfer. The various FEP read commands (READ, BLOCK, SBLOCK, and SVCRD) require a previous STTSG or a dummy FEP DATA READY signal from the small microswitch mounted on the device-77 interface board and labeled FEP INT.

READ

Read data from the FEP directly, 16 bits per read, bypassing the SELCH and the system software drivers. READ's first request is "SENSE STATUS BIT:". Valid responses are:

12	check busy/done bit after read
14	check EOM bit after read
15	check device unavailable bit after read
(cr)	ignore status

If (cr) only is typed, READ requests "DELAY IN MSECS.:". Respond with the decimal number of milliseconds to wait between reads. READ next requests "NO. OF HALF-WORDS:". Type Ø-1Ø24 (decimal); Ø means ignore the READ command. READ reads directly from device 77 (FEP/EKO) using the read halfword (RH) instruction; the status is checked or the program waits between reads, as specified. The data are displayed when READ is satisfied.

BLOCK

Read a block of data from the FEP. BLOCK waits for FEP DATA READY true, then reads the first 16 bits with a read halfword register (RHR). The first halfword read is the number of halfwords to be transferred in the block. BLOCK continues to read (with RH), checking device 77 status bits 14 (EOM) and 12 (busy/done), until EOM is true or until the 2Ø48-byte buffer is full. The block-read terminating condition (EOM SENSED or BUFFER FULL) is displayed and if EOM was sensed, the data are displayed.

SBLOCK — Read a block of data from the FEP through the SELCH. After stopping the SELCH, the SELCH status is read. If it is not Ø, the error is reported and FEPTEST requests another command. If the SELCH is ready, SBLOCK sets up a transfer of 2Ø48 bytes by writing its buffer start and end addresses to the SELCH. SBLOCK waits for FEP DATA READY true then starts the SELCH, initiating the data transfer. SBLOCK then waits until SELCH DONE is true. The data transfer count as read from the SELCH and the data are then displayed.

SVCRD — Read a block of data from the FEP through the SELCH via a call to the system EKO driver (SVC1). This is the manner in which SOUNDER reads the data from the FEP via the EKO device. SVCRD requests "NO. OF HALFWORDS:". The maximum transfer is 1Ø24 (decimal) halfwords. The transferred data are displayed.

SCLOOP — Execute an infinite FEP read loop for a logic analyzer or an oscilloscope. Specify status bit or delay in the same way as for the READ command above.

To get out of SCLOOP or out of any FEP read hangup, type:

TASK FEPTEST	Specify FEPTEST task
PAUSE	Pause FEPTEST
START	Re-start FEPTEST

RESET — Reset device 77 (FEP/EKO). Hex commands ØØ and 18 are sent to device 77 with the output command (OC) instruction.

PAUSE — Pause the FEPTEST task.

END — Terminate the FEPTEST task.

10.11. Subroutine WRCON [D.C.W]

The job of displaying formatted lines on the console from an assembly language program ranks somewhere between a time-consuming chore to a full-blown nightmare, depending upon what needs to be done. Subroutine WRCON was written to simplify this job. It is included here because all of the TEST and utility tasks make liberal use of it. WRCON is the only external subroutine in use by the TEST and utility tasks that is external to the HF Radar sounding software. It resides by itself on the disc as WRCON.CAL; command file TETEST.CSS, used to build TEST tasks (Section 10.3), includes WRCON.OBJ after the object file of the main task code.

10.11. Subroutine WRCON

Programs or modules that make subroutine calls to WRCON require the source code statement:

```
EXTRN     WRCON
```

to permit TET, the task establisher, to provide the subroutine linkage. The EXTRN statement must precede any references to WRCON within the calling module's code. WRCON is not included in the HF Radar sounding software because it is too large (3DE hex or 990 decimal bytes) and provides "bells and whistles" too extravagant for the more memory-efficient sounding tasks.

WRCON outputs to logical unit 1, normally assigned to the console. LU1, however, can be assigned to any device (TTY:, MAG:), or disc file that accepts 76-byte ASCII records. If LU1 has not been properly assigned, or if any write error occurs, WRCON stops with the message:

```
END OF TASK 111
```

Other error conditions are discussed below.

WRCON writes with SVC1 using a function code of 29 (hex). The SVC1 call is suspended until the output device is free; the calling task is not re-scheduled to run until the write is complete. Writing is in image format, suppressing the task ID that otherwise appears on every console line. The buffer in which WRCON composes output lines is 74 bytes long. WRCON appends (cr) and (lf) (required by image formatting) when appropriate so that output records are from 2 to 76 bytes long.

The basic calling sequence is:

```
BAL    R15,WRCON          WRITE TO LOGICAL UNIT 1
DC     A(format)          LINE FORMAT ADDRESS
[DC    ARG1,...,ARGn      ARGUMENT LIST]
DC     15                 ARGUMENT LIST TERMINATOR
(next statement)
```

Only the argument list is optional. Format or argument list programming errors that could kill the system cause

```
END OF TASK 15
```

The line composed up to the argument in error is written before the task stops. Errors that are not potentially fatal simply result in scrambled output; the task is allowed to continue. General registers 0-14 are returned to the user intact; R15 will contain the address of the argument

list terminator plus 2.

A WRCON call argument can be either a register number (Ø-14) or a memory address; the register or memory halfword can contain numerical data to be formatted, an integer count, or the address of an ASCII string or sub-format.

The WRCON format is an ASCII string in the calling task's memory, defined by a program statement, such as:

```
FMT     DB    C'TEXT',Ø
```

The above statement defines 5 bytes at displacement FMT in the module. The 5 bytes are the hex representations of the ASCII characters 'TEXT' followed by a null (Ø) byte. WRCON scans the format from left to right; the null byte flags the end of the format for a given WRCON call and is required. If a format is too long to fit comfortably on a line, an equivalent to the above is:

```
FMT     DB    C'T'              STORE "T"
        DB    C'EXT',Ø          STORE "EXT" NULL
```

There is no practical limit to the number of continuation statements; the text is stored contiguously but must be terminated with a null byte. The label, FMT, need not be on a halfword boundary. The assembly language sequence:

```
        BAL   R15,WRCON
        DC    FMT,15
CONT    NOP (or other executable statement)
```

writes the ASCII characters "TEXT" to logical unit 1 and then continues execution at CONT. If LU1 is the console (CON:), the writing begins at the cursor position. WRCON pre-sets its composing or output buffer to spaces (2Ø hex). As it scans the format string, byte-by-byte, the character picked up is put unchanged into the next output buffer byte position unless it is the up-arrow character (↑, 5E hex). ↑ is the flag used to request a special function ("bell or whistle"). ↑ can be written, but only indirectly (see ↑C, below). The function mnemonic immediately follows the ↑ and is one or two upper case alphabetic ASCII characters, the virgule (/), or the left or right parenthesis.

Though most functions refer to the argument list, some do not. Those that do, advance an argument list pointer in WRCON that must, at format end, point to the argument list terminator, 15 (F hex). After the WRCON call is satisfied, program execution resumes just after the halfword containing 15. For a given WRCON call, the number of ↑-flags in the

format that invoke a reference to the argument list must equal the number of argument list entries.

In the detailed descriptions below, functions that refer to the argument list are flagged with "Arg. list". WRCON distinguishes between registers and memory addresses in the argument list by their magnitudes. WRCON output buffer character positions are referred to below as columns; columns are numbered Ø-75. The numeric value or ASCII string passed to WRCON via an argument list entry is referred to as the object of the argument. For more detailed information about assembly language program statements, refer to the Common Assembler Language (CAL) User's Manaul and to the 16-bit Processor User's Manual.

(The examples below are not presented as examples of good coding practice, but to demonstrate the WRCON functions in simple form.)

↑D
Arg. list

The object of the argument is a numeric value to be written in decimal ASCII. Five column positions are used; leading zeros are suppressed. The following example produces the string:

____1st_day_of_JAN,_1977

where an underscore means a space.

```
        LIS   R3,1                LOAD REGISTER 3 WITH +1
        BAL   R15,WRCON           WRITE LINE
        DC    FMT,R3,YR,15
        . . .
YR      DC    1977                DEFINE CONSTANT 1977
FMT     DB    C'↑Dst day of JAN,↑D',Ø
```

↑RDnn
Arg. list

The object is numeric to be written in decimal ASCII. Only the rightmost (low order) nn digits are written; nn itself is Ø1 through Ø5. Leading zeroes are written. The following example produces "DAY_ØØ1,_1979".

```
        LIS   RØ,1                +1 IN REGISTER Ø
        BAL   R15,WRCON           WRITE LINE
        DC    FMT,RØ,YR,15
        . . .
YR      DC    9
FMT     DB    C'DAY ↑RDØ3,197↑RDØ1',Ø
```

If nn > Ø5, Ø5 is used; if nn = ØØ, no digits are written.

↑H
Arg. list

The object is numeric to be written in hexadecimal ASCII. Four columns are used; leading zeroes are written. The following code produces the line "ØØ64_HEX_is__1ØØ_DEC.".

```
        BAL   R15,WRCON
        DC    FMT,D1ØØ,D1ØØ,15
        . . .
D1ØØ    DC    1ØØ
FMT     DB    C'↑H HEX is↑D DEC.',Ø
```

↑RHnn
Arg. list

The object is numeric to be written in hex ASCII. Only the rightmost (low order) nn digits are written; nn is Ø1 through Ø4. Leading zeroes are written, so ↑RHØ4 is equivalent to ↑H. If nn > Ø4, Ø4 is used; if nn = ØØ, no digits are written. The example below writes "64_HEX=1ØØ_DEC".

```
        LH    R14,D1ØØ          LOAD 1ØØ INTO REG 14
        BAL   R15,WRCON
        DC    FMT,R14,D1ØØ,15
        . . .
D1ØØ    DC    1ØØ
FMT     DB    C'↑RHØ2 HEX=↑RDØ3 DEC',Ø
```

↑C
Arg. list

The object is an ASCII string to be copied into the output. The object string is formed like a format statement and must be terminated with a null byte or (cr) (ØD hex). The code produces:

The_↑_flag_can_be_written_indirectly.

```
        BAL   R15,WRCON
        DC    FMT,STR,15
        . . .
STR     DB    C'The ↑ flag can b',Ø
FMT     DB    C'↑Ce written indirectly.',Ø
```

The format string(s) and the argument object string(s) can be concatenated in any fashion.

↑W
Arg. list

The object is a two-byte (16-bit) ASCII string to be copied into the output. No terminator is required. The code for "ABCDEFG" is:

```
         LHI   R7,C'EF'          "EF" IN REGISTER 7
         BAL   R15,WRCON
         DC    FMT,S1,R7,15
         . . .
FMT      DB    C'A↑WD↑WG',Ø
S1       DB    C'BC'
```

↑/ Terminate line with a carriage return and a line feed. This function has no argument object. Multiple ↑/ functions can appear in a format; multiple lines can be written with one call to WRCON. (In all of the preceding examples, the console cursor would be left at the column following the last character written.) To write:

```
LINE 1
LINE 2
```

use the following code.

```
         BAL   R15,WRCON
         DC    FMT,15
         . . .
FMT      DB    C'LINE 1↑/LINE 2↑/',Ø
```

WRCON clears its output buffer to blanks (2Ø hex) after each ↑/ and resets the column number to Ø.

↑Tnn Tab to column nn, which becomes the next column used. (Remember that the first character position is column Ø.) For example:

```
LINE 1
     LINE 2
          LINE 3

         BAL   R15,WRCON
         DC    FMT,15
         . . .
FMT      DB    C'LINE 1↑/↑TØ5LINE 2↑/↑T1ØLINE 3↑/',Ø
```

The column number, nn, is a two-digit decimal integer ØØ - 74. Tabbing need not be forward. For example, "1234.5" can be written by:

```
        BAL   R15,WRCON
        DC    FMT,N,N,15
        . . .
N       DC    12345
FMT     DB    C'↑D↑TØ4.↑RDØ1',Ø
```

"12345" is first copied to the output buffer; the 5 in column 4 is over-written with a period.

The line written is terminated at the last column number used. For example, "12", the high order two digits, can be written by:

```
        BAL   R15,WRCON
        DC    FMT,N,15
        . . .
N       DC    12345
FMT     DB    C'↑D↑TØ2↑/',Ø
```

"12345" is first copied to the output buffer; the "34" in columns two and three is over-written with (cr) and (lf). The output record is four bytes long.

↑TA
Arg. list

The object of the argument is an integer column number. ↑TA works just like ↑Tnn except the column number is a variable that can be computed. For example:

```
LINE 1
     LINE 2
          LINE 3

        LIS   R2,5                FIRST COLUMN, LINE 2
        LIS   R3,1Ø               FIRST COLUMN, LINE 3
        BAL   R15,WRCON
        DC    FMT,R2,R3,15
        . . .
FMT     DB    C'LINE1↑/↑TALINE 2↑/'
        DB    C'↑TALINE 3↑/',Ø
```

Notice that a null byte terminates the format continuation statement only.

↑Snn

Skip over nn columns. For example, write the following pair of lines:

```
Ø123456789
*        *

         BAL   R15,WRCON
         DC    FMT,15
         . . .
FMT      DB    C'Ø123456789↑/*↑SØ8*↑/',Ø
```

Again, nn is a two-digit decimal integer from ØØ through 74.

↑SA Arg. list

The object is the number of columns to skip. ↑SA functions like ↑Snn except that the number of columns skipped is variable. For example, write the following pair of lines:

```
Ø123456789
*    *    *

         LIS   R5,4                4 IN REGISTER 5
         BAL   R15,WRCON
         DC    FMT,R5,R5,15
         . . .
FMT      DB    C'Ø123456789↑/*↑SA*↑SA*↑/',Ø
```

↑I Arg. list

Ignore the argument. ↑I is useful if one call to WRCON is used for several purposes. For example, "TEXT" is produced by:

```
         LHI   RØ,FMT2             ADDRESS OF FMT2
CALL     STH   RØ,FMTAD            PUT IN CALL SEQUENCE
         BAL   R15,WRCON
FMTAD    DC    Ø,R13,15
         . . .
FMT1     DB    C'PAGE↑D',Ø
FMT2     DB    C'TEXT↑I',Ø
```

"PAGE 1" is produced by:

```
         LHI   RO,FMT1             ADDRESS OF FMT2
         LIS   R13,1               1 IN REGISTER 13
         B     CALL                WRITE PAGE NUMBER
```

↑Mnnc

Write the character c nn times, nn from ØØ through 74. For example, produce the line "***** TITLE *****":

```
        BAL   R15,WRCON
        DC    FMT,15
        . . .
FMT     DB    C'↑MØ5* TITLE ↑MØ5*↑/',Ø
```

↑MAc
Arg. list

Write the character c a variable number of times according to the argument object. For example, produce the same example as above:

```
        LIS   RØ,5                  5 IN REGISTER Ø
        BAL   R15,WRCON
        DC    FMT,RØ,RØ,15
        . . .
FMT     DB    C'↑MA* TITLE ↑MA*↑/',Ø
```

The character c can be any printable non-control character except the single quote, which is used by CAL to delimit ASCII strings. ↑ is usable; (lf) might work (so far untested) but it won't reinitialize the output buffer as ↑/ does.

↑F
Arg. list

The object of the argument is another format string terminated by a null byte. When WRCON sees ↑F, it scans through the sub-format string to its null before continuing the scan of the format string in which the ↑F appears. Sub-formats can invoke their own sub-formats to a total depth of 5 levels. The following example produces "ABCDE":

```
        BAL   R15,WRCON
        DC    FMT,SUB1,SUB2,SUB3,15
        . . .
FMT     DB    C'A↑FE',Ø
SUB1    DB    C'B↑F',Ø
SUB2    DB    C'C↑F',Ø
SUB3    DB    C'D',Ø
```

Since SUB3 contains no ↑-flags, SUB2 could read:

```
SUB2    DB    C'C↑C',Ø
```

↑Xnn

Execute a segment of the format nn times, nn from ØØ through 99 (decimal). Since ↑Xnn (and ↑XA, below) must always be used with ↑(and ↑), which are used to delimit the format segment to be multiply executed, examples will follow the description of ↑).

↑XA
Arg. list — The object of the argument is the repeat count for the multiple execution of a format segment. ↑XA functions like ↑Xnn except for how the repeat count is passed.

↑(— Begin multiply-executable format segment.

↑) — End multiply executable format segment.

Examples of ↑Xnn↑(...↑) and ↑XA↑(...↑) follow.

To write 15 blank lines:

```
         BAL   R15,WRCON
         DC    FMT,15
         . . .
FMT      DB    C'↑X15↑(↑/↑)',Ø
```

To write N blank lines:

```
         LHI   R9,N                N = LINE COUNT
         BAL   R15,WRCON
         DC    FMT,R9,15
         . . .
FMT      DB    C'↑XA↑(↑/↑)',Ø
```

Sequences of multiply-executable format segments can be nested to 5 levels, but counting argument list references can get tricky. The following code produces these three lines:

```
XXXX__XXXX__XXXX__XXXX__
XXXX__XXXX__XXXX__XXXX__
XXXX__XXXX__XXXX__XXXX__
```

```
         BAL   R15,WRCON
         DC    FMT
         DC    ST,ST,ST,ST
         DC    ST,ST,ST,ST
         DC    ST,ST,ST,ST
         DC    15
         . . .
FMT      DB    C'↑XØ3↑(↑XØ4↑(↑C__↑)↑/↑)',Ø
ST       DB    C'XXXX',Ø
```

Notice that only ↑C refers to the argument list, but it does so 4x3 = 12 times. Thus the argument list contains 12 references to the string ST.

10.11. Subroutine WRCON

Finally, any meaningfull string of hex bytes can be sent as control to the Tektronix 4Ø12 console. To clear the screen and home the cursor, for example:

```
        BAL   R15,WRCON
        DC    FMT,15
        . . .
FMT     DB    X'1B',X'C',Ø
```

Since WRCON writes to the console in image mode, one can also select the Tektronix 4Ø12 GRAPH mode or the GIN (graphics input) mode. Refer to the Tektronix 4Ø12 Computer Display Terminal User's Instruction Manual for specific details.

Upcoming revisions of WRCON will return the write status in general register Ø, so although RØ will still be available for argument passing, it will be altered by a call to WRCON. Also, a special call to WRCON will allow the user to set the output record buffer size and to assign LU1.

11. OS MODIFICATIONS

James R. Winkelman

11.1. Introduction

This chapter describes the modifications that have been made to operating system OS16/MT2, Revision 3.

It was decided to use a standard operating system for the first sounder software. The Interdata OS16/MT2 allowed concurrent tasks and had good communication between tasks. The slower the operating system, the slower the soundings. The soundings are being done at 2Ø millisecond intervals because the system does not allow 1Ø.

Two approaches are planned to return the sounding repetition time to 1Ø milliseconds: the FEP can do more processing, which will allow PICKER to do less; and OS16 can itself be speeded up.

Revisions have been made to OS16 to allow it to run faster, to simplify operator interactions, and to help toward remote operation.

Setting SEL.ALL to one in the CUP file (see SYSGEN, Section 9.5) invokes most of the changes. The remainder of the chapter describes the changes in detail.

11.2. SEL.SAFE

SEL.SAFE = Ø, default COMPROC*ST

This variable controls several conditions. It replaces and extends some testing made when CMPROC*ST = Ø. In this situation, there is no command processor or the system is single-task and core-resident. Many tests are not made because there is little a user could do if an error were detected.

Tests made for parameter blocks or buffers starting on an even byte is an example of a class of tests that are deleted. A debugged program runs faster without these tests and that is the purpose of this selection.

A much faster priority reordering is included under SEL.SAFE. This sorting previously used 24% of the computer time during sounding.

11.3. SEL.DISK

| SEL.DISK = 1, default SEL.ALL

When a disc is marked on, the directory block is not physically changed. In the original system, if a disc was marked on during a crash, it could not be marked on or off after the system was restarted, a level of safety required only by payroll files and the like, if then.

When the system is booted, all discs attached are marked ON or ON PROTECTED as indicated by hardware protection at the boot time. This is useful for remote operation and convenient to operators.

11.3. SEL.TIME

| SEL.TIME = 1, default SEL.ALL

Time is displayed on the console in the format M DD HH MM SS. Only the last digit of the month is displayed.

11.4. SEL.IMAG

| SEL.IMAG = 1, default SEL.ALL

The image mode to the console requires that the user provide the carriage return and line feed if they are desired. The system, however, | provides prompts: an "*", the task name, and a ":" or ">". SEL.IMAG ≠ Ø requires that the user provide any of the above that may be wanted. This mode is useful for graphing and gives the user total control of the device while in image mode. There are no changes to the normal mode.

11.5. SEL.FIL

| SEL.FIL = 1, default SEL.ALL

DATE and time of creation and last assignment are stored in the directory. The file listing format is changed also.

The number of sectors used and free are printed as well as the longest string of contiguous sectors. A contiguous file longer than the number after CONT: cannot be allocated.

The format contains the file name, extension, type (INdexed or COntiguous), size (IN: bytes per record; CO: hex sector address of first sector; length is always 256 bytes per sector), keys, count of records (IN) or sectors (CO), directory address (the hex address of the sector which contains the directory of this file), creation date (MMDDYY

HHMMSS), and date and time of last assignment when the disc was not protected.

11.6. SEL.CON

SEL.CON, default 72

The system assumes the length of a console line is either 72 characters or LOGLEN, whichever is longer. This version uses SEL.CON instead of the 72.

11.7. SEL.FEP

SEL.FEP = X'77', default Ø

A SINT SEL.FEP is given every 8.333 milliseconds (1Ø milliseconds for 5Ø cycle) if SEL.FEP is not zero. A sounding cannot be run in a system with SEL.FEP ≠ Ø. A set of test programs can be run, however, to measure the time used by OS16/MT2.

11.8. SEL.SVC

SEL.SVC = 1, default Ø

A default of SEL.ALL would have been preferred, but SEL.SVC must be defined long before SEL.ALL is available.

Two specialized SVC calls are used heavily by the system: SVC 9,- (wait for queue entry); and SVC 6,- (add to the queue of another task). Both of these calls are slow because they share numerous resources with other tasks.

SVC 8,- exactly replaces SVC 9,- and is about Ø.7 milliseconds faster. SVC 4,- replaces SVC 6,- (add to queue only) and is about Ø.8 milliseconds faster. SVC 6,- calls remain unchanged and must be used for any SVC 6 function except add-to-queue. Add-to-queue may be done by either an SVC 6,- or an SVC 4,-. The SVC 4 uses only the task name, the status, and the address of the Q-block.

11.9. SEL.CMD

SEL.CMD = 1, default SEL.ALL

When the system is started, the command file STARTOSW.CSS is executed from SYS1 as a normal CSS file. It can be doing anything from running a program to set the time of day clock from the TSG, to running a complete sounding using a file similar to HFRADAR.CSS. Remember that COUNSEL can be started with a command file ending with GO and CEND.

11.10. SEL.ALL

SEL.ALL = 1, default Ø

SEL.ALL is used as default for most of the other control parameters. It also controls some changes that are not separately controlled. Most of these changes are minor.

An additional SVC 2 command has been added. The parameter block is:

```
DB      OPTION,1Ø
DC      C'HH:MM:SS  MM/DD/YY'
DC      TIME,OFDAY
```

The 22 bytes must be in exactly the order and format given, including "/" and ":". The four bytes TIME,OFDAY are the second of the day as a full word integer. This integer should match the time given in ASCII. If OPTION is 1, the date given in this parameter block is the European format DD/MM/YY.

The system has been changed so that a PAUSE to a PAUSED or dormant task does not cause an error.

11.11. SEL.SWAP

SEL.SWAP = 15, default 15*SEL.ALL

SEL.SWAP allows the swapping of all user memory, including tasks and partitions, between two states. The only difference to the user, who may wish to know which state he is in, is that the second state types only two-character task names for input and output to the console. For example, the VOLUME command types SYSVOL in one state and SYSV in the other.

Swaps occur for three reasons: the user types the command SWAP (see below); the program calls an SVC 2,11; or the previous SVC 2,11 timed out.

SWAP

This is a new command to the command processor. The current user memory is written onto the file SWAPFILE.ROL. If a swap has occurred previously, the user memory is read from the other half of SWAPFILE.ROL. If no previous swap had occurred, the user memory remains unchanged. Swap time-out is set to zero (no time out).

SYS1:SWAPFILE.ROL must be allocated as a contiguous file of 384 sectors on a 7/16 and 896 sectors on an 8/16 with extended memory supported. SYS1 may not be protected during a swap.

The new swap SVC 2 has the following parameter block:

```
DC      11,TIME.OUT
```

This SVC 2 causes an immediate swap and the SVC 2 does not exit until the user memory has been swapped back. TIME.OUT is the time in seconds until this user's memory is swapped back in. If TIME.OUT is zero, no swap back will occur until the operator types SWAP. A TIME.OUT of less tha 60 seconds is not recommended.

The normal uses of this swapping would be a normal user (Fortran and editing) who is swapped out for a minute occasionally by an HFRADAR program that would use the SVC 2,11. If both users use the SVC 2,11, the time out has priority over the current user and might cause problems.

The user may tell which user memory he is in by observing the prompts given when tasks do input and output to the console. The prompts in the "other" memory use only the first two characters of the task name. This would allow a user to run both an ionogram and a kinesonde sounding and know which mode he was in from the COUNSEL prompt. The two sounding programs must use different IONOSOND.SAV files.

The user is responsible for interactions between user memories in the use of I/O devices such as tape. The tape can be assigned to both users in an incompatible mode.

The system tries to allocate the console in a friendly manner. If no typing has been done recently, it is useful to give a carrige return to see what is going on. Any return within SEL.SWAP seconds, currently 15 seconds, of time to swap will cause an immediate swap. Any line taking less that 15 seconds to type will not be interrupted.

Care must be taken to close files in both user memories before protecting the disc using the memory. EDIT uses the files EDITTEMP.nnn, which may become numerous if EDIT crashes or is turned off without being cancelled.

CONSOLE ADVICE COMMANDS

TEST TASKS AND COMMANDS

TASK	PAGE*	COMMANDS	PAGE*
AGC	27	ABort	27
		CUrrent	27
		RGain	27
CKDUP	10	--	--
CONFIG	7	--	--
DATAFMT	10	--	--
DIO	24/25	1Ant AA	24
		1Fil AA	24
		1If AA	24
		1Rf AA	24
		2Ant AA	24
		2Fil AA	24
		2If AA	24
		2Rf AA	24
		ABort	25
		Cal AA	24
		DIo DD,AA	24
		FReq AAAAA.A	24
		HSk CC	24
		LOcal	25
		RWck	25
		STatus	25
DUMP	28	--	--
EXOUT	15/16	(cr)	16
		ABort	16
		GPion n	16
		LIst	16
		LIst F	16
		LIst All	16
EXPCT	12	ABort	12
		CUrrent	12
		FF	12
		RW	12
		TApe [f]	12
FRMAP	2/7	ABort	7
		GEt	5
		ICur	6
		KCur	7
		LIst	6
		PRotect f1[-f2]	5
		SAve	6
		UNprotect f1[-f2]	6
HELP	3/4	(cr)	3
		ABort	4
		ERror EEEE,NNNN	4
		EXplain	3/4
		MEnu [name]	4
HKADC	27/28	ABort	27
		CHan	27
		CHLoop	27
		DAta	27
		RDadc	27
		SElch	27
PIPLN	8/9	(cr)	9
		ABort	9
		Delays	9
		Run [nn]	9
		Setd	9
PLOTEKO	12/15	ABort	15
		AMp	14
		BArxy	14
		COnt	14
		DEtail[n[,G1[,G2]]]	13/14
		FF	15
		LIst n	14
		PLot n	13
		RW	14
		TApe [f]	14
QSTOP	25/27	--	11
SCHED	16/24	ABort	18
		BUild	18
		ENDB	19
		ENDE	19
		ENDR	18
		GEt name.xxx	19
		IFErr	18
		LIst	19
		REPeat n	18
		SSAve xxx	18
		TIme n mm[:ss]	19
		TSAve name.xxx	18
SCLOOP	25/26	ABort	26
		DAta	26
		LOop	25/26
SNRIO	9	ABort	9
		CUrrent	9
		TEst	9
TAPEKO	10/12	--	--
TSGINT	7/8	(cr)	8
		ABort	8
		Run [nn]	8
TXSTAT	7	ABort	7
		STatus	7

* ALL PAGE NUMBERS ARE IN CHAPTER 10.