NASA CONTRACTOR REPORT 166363

(NASA-CR-166363) NASA PIONEER: VENUS REVERSE PLAYBACK TELEMETRY PROGRAM TR 78-2 (Rensselaer Polytechnic Inst., Troy, N. Y.) 114 p HC A06/MF A01 CSCL 09B

N82-30948

Unclas G3/59 28645

NASA Pioneer - Venus Reverse Playback Telemetry Program TR 78-2

J. W. Modestino
D. G. Daut
A. L. Vickers
K. R. Matis

CONTRACT NAS2-10006
December 1978

NASA

NASA CONTRACTOR REPORT 166363

NASA Pioneer - Venus Reverse Playback Telemetry Program TR 78-2

J. W. Modestino
D. G. Daut
A. L. Vickers
K. R. Matis
Electrical & Systems Engineering Dept.
Rensselaer Polytechnic Institute
Troy, New York

Prepared for
Ames Research Center
under Contract NAS2-10006

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

ORIGINAL PAGE IS
OF POOR QUALITY

NASA Pioneer-Venus Reverse Playback Telemetry Program

J. W. Modestino, D. G. Daut,
A. L. Vickers, K. R. Matis

Communication & Information Processing Group

Electrical & Systems Engineering Department
Rensselaer Polytechnic Institute
Troy, New York 12181

TR 78-2

December 1978

This work was performed under Contract No. NAS2-10006 for the National Aeronautics and Space Administration, Ames Research Center, Moffett Field, CA.

Period Covered

July 1, 1978 to December 1, 1978

Abstract

This report is concerned with describing the NASA Pioneer-Venus Reverse Playback Telemetry Program. This program is a software package developed to decode telemetry data received from the Pioneer-Venus Multiprobe Mission. The program processes recorded data in an off-line mode of operation. The program reads a digital tape containing receiver soft-decisions recorded during the off-line processing, in either the forward or reverse direction, of the Pioneer-Venus pre-detection recordings. The 24-bit frame synchronization word of each telemetry frame is found. Next, the data are decoded on a frame-by-frame basis using either a sequential decoder or a quick-look decoding procedure. Decoded data are available in printout form or may be written onto another digital tape in the proper time sequence.

This report is organized into three basic categories. First, a detailed description of installation dependent software and program-user interaction is provided. Second, a discussion of the frame synchronization algorithm and decoding procedures is presented. Finally, computer program details are included in the Appendices.

ORIGINAL PAGE IS
OF POOR QUALITY

Table of Contents

LIST OF TABLES

List of Figures

ORIGINAL PAGE IS OF POOR QUALITY

1.0 Introduction

The NASA Pioneer-Venus Reverse Playback Telemetry Program is a software package which has been developed to decode telemetry data received from the Pioneer-Venus Multi-probe Mission. The program has been designed to function on the computer facilities of the NASA Ames Research Center. It will process recorded data in an off-line mode of operation. Specifically, the program will read a digital tape containing; 1.) receiver soft-decisions from the Deep Space Network's Symbol Synchronizer which will be recorded during the off-line processing (in both the forward and reverse direction) of the Pioneer-Venus pre-detection recordings; 2.) time tags associated with the data, and 3.) an identification file. The 24-bit frame sync word of each telemetry frame is found, and the direction of the symbols associated with each frame sync word is then reversed, if necessary. Next, the data are decoded on a frame-by-frame basis using either a sequential decoder or a quick-look decoding procedure. Finally, the results are available in printout form or may be written out onto another digital tape in the proper time sequence.

The primary objective of the Reverse Playback Telemetry Program is to minimize information loss due to gaps in the received data. The telemetry from each probe will be sequentially decoded by the Telemetry Processor Assembly during the real-time data processing. These data will contain gaps that appear during the time that there is a need to synchronize or resynchronize any of the units comprising the telemetry system for that particular probe. These data gaps will occur for each probe during the mission at 1.) the initial acquisition of received data, 2.) communications blackout when the spacecraft enters the Venus atmosphere, 3.) the time the probes change data rate, and 4.) times when the receivers lose lock due to probe motion caused by Venus atmospheric wind gusts.

The data which are lost due solely to the synchronization process will be recovered by off-line data processing of the pre-detection recordings made from the open-loop receivers. Figure 1 illustrates the Ground Station configuration. Processing to recover such data will consist of 1.) playing the analog tape of receiver outputs in the reverse direction, 2.) up-converting the signal from the tape recorder to S-band and feeding it into one of the closed-loop receivers in the real-time system, 3.) processing the signal through the receiver, subcarrier demodulator, and symbol synchronizer, and 4.) recording the soft-decisions out of the symbol synchronizer on digital tape. This digital tape is used as input to the Reverse Playback Telemetry Program. Synchronization is established for each frame and the data is decoded using the software sequential decoder employing the Fano Algorithm.

2.0 Program Usage: Initialization of Program at NASA Ames

This section explains the installation dependent sofware developed for the NASA Ames Research Center's IBM 360/67 computer system, as well as the procedures for utilizing the Reverse Playback Telemetry Program. Two procedure definitions (PROCDEFS) have been prepared to perform the tasks of tape mounting, storage allocation, and logical unit assignments. This obviates the need for the program user to type in the sequence of command strings which would normally be required.

Before exercising the program, the user must first submit both the input data tape and the output tape (if one is desired) to the Computer Operator. The tapes and a TSS Job Card are submitted to the attendent in the Computation Center's I/O Room. Each magnetic tape must be labeled according to the recommended format shown below.

User Name	Volume Name	Mail Stop

Tape Label

<u>User Name</u>: This block contains the name of the user authorized to use the computer account along with any tape reference information provided by the user.

<u>Volume Name</u>: JPL (for input tapes at 1600 BPI).
REVIDR (for output tapes at 800 BPI).

<u>Mail Stop</u>: This information tells the I/O Personnel where to return the tapes when program execution is finished.

Once the tapes have been submitted, the user must log on the NASA Ames TSS system. The LOGON procedure for a terminal (ASCII-type) will be given, However, alternate LOGON procedures can be found in NASA Ames Research Center - A Guide to TSS/360. Connecting the terminal with the system is accomplished by initializing the Ames Dataswitch unit associated with the terminal. The system will respond with a hyphen; typing TSS and then pressing the Return Key connects the terminal with the system allowing the LOGON command to be entered -

... NASA AMES SWITCH A ...	(system)
-TSS	(user types TSS, hit Return)
/LOGON b̸userid,,terminalid	(user types in command,
TSS 9.0 PTF49X	hit Control S to send
PI=.879	this and all future
ENTER PASSWORD	responses.)
*****	

where, userid - is the user identification that was assigned to the computer account.

terminalid - the terminal identification number of the conversational terminal being used for the task.

***** - enter user password after being prompted.

Having logged on to the TSS/360 system the Reverse Playback Telemetry Program can now be initialized by entering the appropriate PROCDEF command (RUN1 or RUN2). The first PROCDEF, RUN1, is invoked when the user desires only printed output. The second PROCDEF, RUN2, is invoked in the event the user has an additional requirement for tape output of decoded data. Listings of the PROCDEF files follow along with an explanation of their operation.

RUN 1

This procedure definition consists of the following system commands -

```
PROCDEF RUN1
PARAM $TAPE
DDEF FT03F001,VS,LPRINT.OUT
DEFAULT SYSINX=E
MTMSG
PLEASE OBTAIN 9-TRACK, 1600-BPI TAPE $TAPE WITH NC RING
DDEF FT04F001,PS,INPUT,UNIT=(TA,9D3),DISP=OLD,-
LABEL=(1,NL),VOLUME=(,$TAPE),PROTECT=Y,-
DCB=(DEVD=TA,DEN=3,RECFM=F,LRECL=1042)
DDEF FT06F001,VI,TEMP,RET=T,DISP=NEW,-
DCB=(RECFM=F,LRECL=1042)
DEFAULT SYSINX=G
DRIVER$$
PRINT LPRINT.OUT,PRTSP=EDIT,ERASE=Y
RELEASE FT04F001
RELEASE FT06F001
DELETE INPUT
ERASE TEMP
```

To begin execution of the program with the printout-only option the user issues the command: RUN1 JPL where, RUN1 is the name of the above procedure command file and JPL is the volume or name of the input tape containing the data to be processed. It is possible to specify tape names other than JPL with this procedure definition since the tape name is an input parameter specified by the user.

The basic operation of the procedure command file is to alert the computer operator as to which tape is to be mounted, assign logical Fortran I/O devices for program operation, and initiate program execution. The input tape definition command defines the DDNAME "FT04F001" and associates it with the physical sequential (PS) data set "INPUT". This data set has been previously written (OLD), the tape is unlabeled, and is to be read from a 9-track tape drive having a density of 1600 BPI. The volume name supplied by the user is substituted in for the dummy parameter $TAPE and the particular tape is to be write protected when mounted. The Data Control Block indicates that the physical device is a tape drive (DEVD=TA), the density is 1600 BPI (DEN=3), the record format is of fixed-length (RECFM=F), and the record length is 1042 bytes (LRECL=1042). Lines 0900 and 1000 are the data definition for the mass storage.The DDNAME 'FT06F001" is associated with the virtual indexed (VI) data set named "TEMP". This assigns Fortran logical unit 6 to the temporary disk storage file. The file is specified as being temporary (RET=T) and that it is to be created (DISP=NEW). The character of the data set is described as being of fixed record format with a record length of 1042 bytes. Program execution is initiated by issuing the command DRIVER$$. The user now will be quizzed by the program as to which options are desired. When program execution is terminated the print file will be released to the line printer if output to the line printer was specified. Finally, the device assignments are released and the temporary mass storage file is erased from the user's library.

<u>RUN 2</u>

This procedure definition consists of the following system commands-

```
0000    PROCDEF RUN2
0100    PARAM $TAPEI,$TAPEO
0200    DDEF FT03F001,VS,LPRINT.OUT
0300    DEFAULT SYSINX=E
0400    MTMSG
0500    PLEASE OBTAIN 9-TRACK, 1600-BPI TAPE $TAPEI WITH NO RING
0600    MTMSG
0700    PLEASE OBTAIN 9-TRACK, 800-BPI TAPE $TAPEO WITH RING IN
```

```
DDEF FT04F001,PS,INPUT,UNIT=(TA,9D3),DISP=OLD,-
LABEL=(1,NL),VOLUME=(,$TAPEI),PROTECT=Y,-
DCB=(DEVD=TA,DEN=3,RECFM=F,LRECL=1042)
DDEF FT05F001,PS,DATOUTE1,UNIT=(TA,9D2),DISP=NEW,-
LABEL=(1,NL),VOLUME=(,$TAPEO),PROTECT=N,-
DCB=(DEVD=TA,DEN=2,RECFM=F,LRECL=1216)
DDEF FT05F002,PS,DATOUTE2,UNIT=(TA,9D2),DISP=NEW,-
LABEL=(2,NL),VOLUME=(,$TAPEO),PROTECT=N,-
DCB=(DEVD=TA,DEN=2,RECFM=F,LRECL=1216)
DDEF FT06F001,VI,TEMP,RET=T,DISP=NEW,-
DCB=(RECFM=F,LRECL=1042)
DEFAULT SYSINX=G
DRIVER$$
PRINT LPRINT.OUT,PRTSP=EDIT,ERASE=Y
RELEASE FT04F001
RELEASE FT05F001
RELEASE FT05F002
RELEASE FT06F001
DELETE INPUT
ERASE TEMP
```

To begin execution of the program with the additional capability of writing an output digital tape, the user issues the command: RUN2 JPL,REVIDR where, RUN2 is the name of the above procedure command file, JPL is the name of the input tape, and REVIDR is the name of the output tape to be written. Again, it is possible to specify different tape names as desired since the tape names are input parameters issued by the user.

The operation of this procedure command is identical to that of RUN1 with the addition of an output tape definition. The DDNAME"FT05F001" is associated with the data set "DATOUTE1". This data set is to be created (NEW),the tape is unlabeled, and to be written on a 9-track tape drive having a density of 800 BPI. The record format is fixed with a record length of 1216 bytes. A double end-of-file is written after the last data record on the output tape.

A flowchart which illustrates the basic operation of the program is provided in Fig. 2. The program begins by setting the logical unit numbers for each of the inputs and outputs used by the program. These assignments are given in Table 1. These logical unit numbers must be assigned to specific devices by the user before the program is executed. These assignments were discussed in Sec. 2.0.

The next step in the program is to interact with the user to input the various options and values necessary to guide program execution. The options the user must specify are asked for by the program in a conversational mode of operation. Specifically, the following is asked of the user:

A.) IS DATA RECORDED ON TAPE IN FORWARD OR REVERSE DIRECTION (FWD OR REV)?

The user responds with FWD if data on the tape are recorded in the forward direction, or REV if data on the tape are recorded in the reverse direction.

B.) DO YOU WANT TAPE OUTPUT OR PRINT OUTPUT (TAPE OR PRNT)?

If the user responds with TAPE, program output is written to magnetic tape, whereas, if the user responds with PRNT, program output will be printed on a hard-copy device.

C.) DO YOU WANT PRINTOUT TO THE TERMINAL OR TO THE LINE PRINTER (TERM OR LPRNT)?

The user may choose to have the program output printed on the terminal for short sessions or sent to the line printer when large amounts of data are to be processed.

D.) WHICH PHASE REFERENCE DO YOU WANT - DATA OR DATA-BAR (D OR DB)?

The user must indicate the desired phase reference to be used in mapping the soft-decisions (see Fig. 10). The user responds with D if the DATA phase reference is desired or with DB for the DATA-BAR which is a 180° phase reversed assignment. Fano decoder operation is sensitive to the phase reference used, hence the correct phase assignment is the one which results in a frame being successfully decoded when reliable input data are used,i.e., a frame with high signal-to-noise ratio.

E.) IS THE DATA TO BE DECODED OR IS RAW SYMBOL OUTPUT DESIRED (DEC OR RAW)?

The user must indicate whether data are to be decoded or only the raw data symbols (undecoded) are to be printed out. The raw data symbols are printed out in octal and binary form. This question is not asked if the user has previously specified tape output since raw symbols cannot be outputted to tape.

F.) DO YOU WANT THE FANO DECODER USED OR THE QUICK-LOOK OUTPUTS (FANO OR QL)?

If the data are to be decoded, the user must specify whether the Fano sequential decoding algorithm or the "Quick-Look" algorithm is to be used. This question is not asked if the user has previously specified raw data symbol output.

G.) DO YOU WANT ALL THE FRAMES PRINTED OUT (Y OR N)?

The user must specify whether all frames are to be processed, or if only selected frames are to be processed. Responding with Y will result in all data frames being processed according to the options thus far chosen. If the user wishes to select the frames to be processed a response of N should be issued.

H.) DO YOU WANT TO SPECIFY THE FRAMES TO BE PROCESSED ACCORDING TO TIME TAG OR FRAME NUMBER (TIME OR NUMB)?

If the user specifies that only selected frames are to be processed, it must be indicated whether the frames to be selected will be specified by frame number or time tag. If the user responds with TIME, the program will reply with the following-

SPECIFY THE TIME TAG OF EACH FRAME WHICH IS TO BE PRINTED OUT.
INDICATE YOU ARE FINISHED BY ENTERING A -1.0

HOURS-

The user then enters the hour from 0.0 to 23.0 of the time tag for the desired frame. The number entered must be in floating-point format. The program then prints out:

MINUTES-

The user enters the minutes from 0.0 to 59.0 of the time tag for the desired frame. The number entered must be in floating-point format. The program will then print out:

SECONDS-

The user enters the seconds from 0.0 to 59.0 in floating-point format. After the user has entered the hours, minutes, and seconds of the time tag for a desired frame, the program will continually request this type of information for more frames that are to be selected by time tag. The user indicates that no more frames are to be processed by entering a -1.0 as the time input.

If the user responds to question H with NUMB, the program will reply with the following-

SPECIFY BY NUMBER THE FRAMES WHICH ARE TO BE PROCESSED. THE LAST DIGIT OF EACH NUMBER MUST END UP IN COLUMN 5. INDICATE YOU ARE DONE BY HITTING A RETURN.

ORIGINAL PAGE IS
OF POOR QUALITY

I.) DO YOU WANT RAW SYMBOLS TO BE SYNCHRONIZED (1024) IN A DATA FRAME OR THOSE (1025) CORRESPONDING TO A TAPE RECORD (SYNC OR TREC)?
If the user requests SYNC the raw symbols will be synchronized to agree with a data frame, otherwise the raw symbols will be printed as they appear in a tape record.

The user must now enter the numbers of the frames which are to be processed. There is to be one number per line with each number ending in column 5. Frame number input is terminated by either entering a zero or hitting a return.

After having selected the desired options, the user is asked to specify input values for several parameters. The parameters are:

A.) The DSS number. ENTER DSS NUMBER (14 OR 43)-
The user must respond with either 14 or 43.

B.) Probe identification. ENTER PROBE ID (SP1,SP2,SP3, OR LP)-
The user should respond with SP1, SP2, SP3, or LP corresponding to small probe # 1, small probe # 2, small probe # 3, or the large probe, respectively.

C.) Tape sequence number. ENTER TAPE SEQUENCE NUMBER-
If tape output was requested, the user must specify a two digit tape sequence number.

D.) Year of data recording. ENTER YEAR DATA WAS RECORDED (E.G. 78)-
If tape output was requested, the user is asked to input the last two digits of the year in which data was recorded.

E.) Decoder computations limit. INPUT MAXIMUM NUMBER OF COMPUTATIONS PER FRAME. THE LAST DIGIT OF THE NUMBER MUST END UP IN COLUMN 8. 0 INDICATES DEFAULT = 100,000 COMPUTATIONS.

If the Fano sequential decoder was requested, the user must specify the maximum number of computations allowed per frame (see Sec. 5.3) in integer format with the last digit in column 8. When a zero or blank is entered, the default value of 100,000 computations per frame is set.

After all the options and parameters have been specified, the contents of the input tape are read onto the direct-access device assigned to logical unit 6. The space allotted must be sufficient to accommodate a maximum of 5000 records, each record being 521-16 bit words.

After the data are read onto the direct access device, the data are processed and outputted according to the user's specifications. Upon completing the assigned task, the program asks ARE YOU DONE (Y OR N)? If the user is finished a Y should be entered to terminate execution of the program. Should the user wish to process the data for a different set of options, responding with an N is required. Now the program will request a new set of options by proceeding as described before. It should be noted that if N is the response, the user will not be given the option of choosing a different tape direction. This is due to the fact that the program accesses the data directly from the direct-access file for the subsequent program passes.

Fig. 3 shows a sample session using the Reverse Playbeck Telemetry Program. Here, print output had been sent to the terminal instead of the line printer. Note that if the user mistypes a response, the program asks the same question again and waits for a correct response.

3.0 Quick-Look Decoding

The encoder aboard the spacecraft is a convolutional encoder as illustrated in Fig. 4. The convolutional code employed is non-systematic, i.e., the input data do not appear explicitly in the encoded output sequence. Systematic convolutional codes, on the other hand, are characterized by the fact that the first stage of the shift register is the only connection to one of the channel output ports. The code rate, R, is one-half. That is, for every input information bit there are two channel symbols at the output of the encoder. The encoder has a constraint length of 32 bits. A telemetry frame consists of 512 information bits (1024 channel symbols). The last 24 bits is a Frame Synchronization Word. This is a fixed binary sequence which appears at the end of each frame. It has the form 1 1 1 1 1 0 0 0 1 1 0 0 0 1 0 1 0 1 0 0 1 0 0 1.

The encoder in Fig. 4 outputs two channel symbols denoted by P and $\overline{Q}$, where $\overline{Q}$ is the complement of Q. The value of each symbol is based on the values of 32 selected data bits previously fed into the shift register. Modulo-two adders are connected to the various shift register cells as shown. Each symbol, P and Q, is a logical 'one' if there is an odd number of 'ones' in the selected data bits and a logical 'zero' if there are an even number. The encoding cycle begins at the end of the last bit of each frame synchronization word. At this time the shift register and the flip-flop used to generate the code are reset to a logical 'zero'. If the inverter on the Q channel output port is removed the resulting structure of the set of all output codewords $\underline{y}$ can be viewed in terms of a code tree as shown in Fig. 5. This code tree is obtained by writing along each branch the two digits of $\underline{y}$ representing the encoder output corresponding to the encoder input sequence

$\underline{x}$ at each node.

The non-systematic structure of the code allows for a 'quick-look-in' feature to be exploited (cf [1]). Quick-look decoding permits recovery of the original information sequence from the hard-decisioned received data without resorting to sequential decoding methods simply by modulo-two addition of the received sequences. The quick-look decoding technique is illustrated in Fig. 6. Inverting the Q channel symbol at the encoder prior to transmission requires that the hard-decision made on the second symbol of every received channel pair be complemented (inverted) before the quick-look decoding can be performed. Since the taps for the two output symbols P and Q are identical, except for the second stage, then the original information sequence, D, can be reconstructed by modulo-two addition of the two symbol streams except for a one-bit delay. That is, the operation+ has the form:

$$D_j = P_{j+1} \oplus \bar{\bar{Q}}_{j+1} \qquad j = 0,1,2,\ldots,511 \tag{3.1}$$

This process will recover the complete information sequence except for the last bit received just before resetting the encoder. The state of this bit cannot be predicted. However, the last bit of the original information sequence is the last bit of the frame synchronization word. When operating in the quick-look mode, it requires only that the last bit of the frame sync be ignored. The Reverse Playback Telemetry Program possesses the option of decoding a frame using the quick-look technique. Hard-decisions on the received sequence are obtained from the three-bit soft-decisions provided on the input digital tape.

+ Here $\bar{\bar{Q}}_{j+1}$ represents the logical compliment of $\bar{Q}_{j+1}$. Ideally this should equal Q_{j+1} in the absence of channel noise. We prefer the notation $\bar{\bar{Q}}_{j+1}$ to emphasize that the quantities P_{j+1} and $\bar{\bar{Q}}_{j+1}$ in Eq. (3.1) are channel outputs.

4.0 Frame Synchronization Algorithm

Due to the nature of sequential decoding, it is necessary to synchronize the decoder with the input data before the decoding process can proceed. Synchronization is accomplished by locating the position of the sync word in each frame.

Consider the problem of locating the sync word in n frames of data. It is highly probable that the actual start bits of the data frames do not coincide exactly with the start bits of the physical frame records as they are stored on the digital data tape. This is demonstrated in Fig. 7 a. It is obvious that, in general, the n data frames will span n+1 tape frames.

Let $S_{i,j}$ denote the i'th symbol of the j'th tape frame ($i = 0,1,\ldots,1024$; $j = 1,\ldots, n+1$). As noted previously, the coded symbol stream may be converted to the original data stream by first inverting the hard-decision made on the second symbol of every channel pair and then performing an exclusive-or operation on the channel symbols. However, the correct pairing of the symbols for performing the quick-look decoding operation is not evident in the received data stream. This means that the quick-look decoding operation must be performed on the received data for the two possible combinations of channel symbol pairs, that is

$$1.)\quad S_{0,j} \oplus S_{1,j}\ ;\ S_{2,j} \oplus S_{3,j};\ldots;\ S_{1022,j} \oplus S_{1023,j} \tag{4.1}$$

$$j=1,2,\ldots,n$$

$$2.)\quad S_{1,j} \oplus S_{2,j}\ ;\ S_{3,j} \oplus S_{4,j};\ldots;\ S_{1023,j} \oplus S_{1024,j} \tag{4.2}$$

Now let C_ℓ denote the ℓ'th bit of the frame synchronization word ($\ell = 0,1,\ldots,22$). The recovered data streams, by Eqns. (4.1) and (4.2), are then each correlated with the sync word for each possible sync word location. The sync word location which gives the highest correlation is taken as the true sync word location. This can be stated mathematically as follows:

Define the correlation operator $*$ according to the truth table of Fig. 7b. This operation gives a value of -1 if the symbols to be correlated are different and +1 if the symbols are the same. The operation thus performed is:

$$\max_{m=0,1,\ldots,511} \; \max_{k=0,1} \; \sum_{j=1}^{n} \sum_{\ell=0}^{22} C_{\ell} * \left[S_{\{2m+k+2\ell\},j} \oplus \bar{S}_{\{2m+k+2\ell+1\},j} \right] \tag{4.3}$$

where, k represents the two possible channel symbol pairings and m represents the 512 possible locations of the start of the sync word. Note that the first subscript of each channel symbol term is enclosed in braces. This indicates the term is to be evaluated modulo 1024 for the case k=0 and modulo 1025 for the case k=1, i.e.,

$$\{n\}_{1024} = \begin{cases} n & 0 \leq n \leq 1023 \\ n-1024 & n \geq 1024 \end{cases} \tag{4.4}$$

$$\{n\}_{1025} = \begin{cases} n & 0 \leq n \leq 1024 \\ n-1025 & n \geq 1025 \end{cases} \tag{4.5}$$

This takes into account the possibility that the sync word may cross over a tape frame boundary as illustrated in Fig. 7c. It is easily see that if L bits of the sync word occur at the end of the tape record, then the remaining 24-L sync bits from the previous data frame will occur at the start of the tape record. The modulo operation allows the correlation to be performed on the L bits at the end of the record and the 24-L bits at the beginning of the record.

After the maximum has been found, the values of k and m give the proper channel symbol pairing and location of the sync word starting point. The location of the start of a frame is easily deduced.

4.1 Frame Synchronization Subroutine Description

The subroutine SYNC performs the synchronization of the data for the main program. The call to the routine takes the form:

```
CALL SYNC(J,NREC,IDISK,IFRB,IWORK,IRETA,IQLA,NFRMS,ESNO,IFFRT,ISYCW)
```

where the parameters are:

J - is the first record to be used in locating the sync word.

NREC - the total number of records of input data.

IDISK - is the logical unit number for the random access device (#6).

IFRB - is a (512,9) array used for inputting the data from which the sync word is to be located.

IWORK - is a (1025,9) array used as a work array.

IRETA - is a (1024,8) array in which the unpacked soft-decisions are returned in proper synchronization.

IQLA - is a (512,9) array used as a work array. Upon subroutine completion it contains the quick-look outputs in proper frame sync .

NFRMS - is the number of frames which are to be used in locating the sync word. $(1 \leq NFRMS \leq 8)$

ESNO - is an eight word array in which the signal to noise ratio for each synchronized frame is returned.

IFFRT - is the number of frames synchronized and returned in IQLA and IRETA. $(0 \leq \text{IFFRT} \leq 8)$

ISYCW - is location of the frame sync word returned by the subroutine.

The subroutine determines the location of the frame sync word in the received data sequence and then re-orders the sequence to obtain proper frame synchronization. Having acquired frame sync the subroutine computes the quick-look outputs. Furthermore, due to the inverter on the Q channel symbol output of the encoder, the complement of the $\bar{Q}$ received soft-decisions are computed by the SYNC subroutine providing the encoder outputs P, $\bar{\bar{Q}}$ to the Fano subroutine in proper frame sync.

ORIGINAL PAGE IS OF POOR QUALITY

5.0 Fano Sequential Decoding Algorithm

The sequential decoder accepts a sequence of demodulator output symbols, denoted by $\underline{r} = \{r_j\}$, and attempts to find a path through the code tree described in Fig. 5 , which has a high likelihood of having produced the sequence $\underline{r}$. It does so by selecting a tentative path into the tree starting at the origin and following at each node the branch that best matches the appropriate segment of $\underline{r}$. Whenever the path that the decoder is currently following becomes too unlikely, a search is initiated for a better path. The attractive feature of a sequential decoder is its ability to determine the correct (transmitted) L-branch path while examining only a small fraction of the $2(2^L-1)$ branches that make up the total set of 2^L possible paths. The actual number of branches examined in decoding a sequence is a random variable. If the channel noise is quite severe, the number of branches that must be examined (some more than once) by the decoder before accepting an L-bit path becomes very large. The examination of a branch, in either the forward or reverse direction, will be referred to as a computation.

The data have been organized into frames of 488 information bits long. The 32-stage shift register in the convolutional encoder is set to a prechosen value of all 0's before transmitting each frame. After each frame is transmitted a prechosen sequence of 24 bits is fed into the encoder. This known sequence, called the tail sequence, serves to terminate the code tree 48 channel symbols after the final node, thereby providing sufficient energy to reliably decode the last information bit. Furthermore, the known tail sequence provides a repeating pattern of channel symbols (independent of data) which is used for obtaining frame synchronization on the received bit stream. Each telemetry frame is composed of 1024 channel symbols corresponding to 512 data bits [2].

The convolutional encoder output symbols are transmitted to the receiving station and are demodulated at the receiver output. Each output symbol represents the sum of the transmitted symbol and the system noise. Each demodulator output is quantized to one of eight amplitude levels. These soft-decisions on the received sequence are made use of by the sequential decoder. The decoder contains a replica of the encoder in which it can try various possible transmitted sequences and compare its output with the channel output. The decoder attempts to find the correct transmitted path through the code tree by making use of reliability information provided by the soft-decisions. In order to minimize the probability of decoding on the wrong path, the decoder must, in the decoding process, attempt to find the most probable path through the tree. By Bayes rule, the a posteriori probability of any path K nodes long is

$$p(\underline{x}^{(j)}|\underline{r}) = \frac{p(\underline{r}|\underline{x}^{(j)})p(\underline{x}^{(j)})}{p(\underline{r})} \tag{5.1}$$

where the vector $\underline{r}$ is the sequence of channel outputs, and $\underline{x}^{(j)}$ is the sequence of channel inputs corresponding to the jth path. Any monotonically increasing function of the a posteriori probability will suffice in characterizing the performance of the decoding process. Since the system noise is additive white Gaussian and the channel is memoryless (1) reduces to a product of probabilities. An equivalent measure of path likelihood is given by the cumulative path metric (up to level K).

$$L_k = \sum_{\ell=1}^{K} \left\{ \sum_{m=1}^{n} \left[\log_2 \frac{p(r_{\ell m}|x_{\ell m}^{(j)})}{p_0(r_{\ell m})} - B \right] \right\} \tag{5.2}$$

Here, $r_{\ell m}$; m=1,2,...,n is the m^{th} receiver output along the ℓ^{th} branch and $x_{\ell m}^{(j)}$ is the associated channel input sequence. The constant B is chosen to have a value such that, for the correct path, L_k will increase on the average and, for incorrect paths, L_k will decrease on the average. The Fano algorithm detects incorrect paths by comparing the cumulative path metric with a running threshold value.

Fig. 8 shows the relationship between L_k and the sequence of thresholds, separated by T_0, as the decoding process evolves. As the decoder proceeds forward along a path, it chooses the most likely branch at each node and then tightens the running threshold so that it is never more than T_0 beneath L_k for the current path. The flow diagram of Fig.9 illustrates the action of the Fano algorithm. If the path should fall below the threshold, the decoder goes into a backward search in which it searches all other paths in order of decreasing branch probability looking for one which does not fall below the present threshold T. If all those paths which originate above the threshold are exhausted, the threshold is reduced by T_0 and the original path is retraced. If this path does not remain above the new threshold, the decoder searches back again looking for a path which does. This process is repeated, reducing the threshold by steps, until success is realized, at which point the decoder continues forward into the decoding process. To eliminate the possibility of the decoder entering into a loop in the searchback mode the threshold is not allowed to increase when the decoder reaches previously explored branches. This action is insured by making use of a binary variable θ which keeps track of when the running threshold may be tightened. The variable θ, initially 0, is set equal to 1 immediately following observation of a running threshold violation on a forward look. As long as θ remains equal to 1 the algorithm prevents tightening of T. As soon as the new node is examined (one never reached before) θ is reset to 0 and tightening is again permitted [3].

5.1 Branch Metric Computation

For the Fano sequential decoder, the cumulative path metric up to level K in the code tree associated with the path $\underline{x}^{(j)}$ on the basis of the receiver output sequence $\underline{r} = (r_1, r_2, \ldots)$ is given by (5.2). The Branch Metric associated with the ℓ^{th} branch along the j^{th} path is given by the inner summation of (1)

$$d(\underline{r}_\ell,\underline{x}_\ell^{(j)}) = \sum_{m=1}^{n}\left[\log_2 \frac{p(r_{\ell m}|x_{\ell m}^{(j)})}{p_0(r_{\ell m})} - B\right] \tag{5.3}$$

Where, B is a bias term chosen to insure that (1) increases in value when the decoder is following the correct path. The bias term will be taken as equal to the code rate R= 1/2.

The Fano decoding algorithm makes use of soft-decisions on the channel outputs (demodulator outputs). Each channel output is quantized to one of eight levels. To a first order approximation, the modulator-channel-demodulator cascade can be replaced by a constant, discrete, memoryless channel with two input letters, eight output letters, and transition probabilities $\{p(\tilde{r}_{\ell m}^{(k)}|x_{\ell m}^{(j)}\}_{k=1}^{8}$. Under the assumption of coherent BPSK modulation with matched filter reception appropriately normalized to the noise power the complete received signal $r(t)=s(t)+n(t)$ can be characterized as

$$r_{\ell m} = x_{\ell m}\sqrt{\frac{2E_s}{N_0}} + n_{\ell m} \tag{5.4}$$

where $x_{\ell m}$ is the transmitted binary symbol (± 1) and $n_{\ell m}$ is a zero-mean, unit variance Gaussian random variable. It follows that

$$p(r_{\ell m}|x_{\ell m}^{(j)}) = \frac{1}{\sqrt{2\pi}} \exp\{-\frac{1}{2}(r_{\ell m}-x_{\ell m}\sqrt{\frac{2E_s}{N_0}})^2\} \tag{5.5}$$

while

$$p_0(r_{\ell m}) = \frac{1}{\sqrt{2\pi}} \exp\{-\frac{1}{2} r_{\ell m}^2\} \tag{5.6}$$

The channel transition probabilities can now be calculated as follows:

$$p(\tilde{r}_{\ell m}^{(k)}|x_{\ell m}^{(j)}) = \int_{E_k}^{E_{k+1}} p(r_{\ell m}=\xi|x_{\ell m}^{(j)})d\xi \tag{5.7}$$

Here, E_k; $k=0,1,\ldots, 8$ are the input bin boundaries of a uniform, symmetric quantizer (Fig.10) and $\tilde{r}_{\ell m}^{(k)}$; $k=1,2,\ldots, 8$ are the output values of the quantizer. (Note: $E_0=-\infty$, $E_8=+\infty$). The quantity $p_0(r_{\ell m})$ is given by (5.6) Hence, for quantized outputs (5.6) becomes

$$p_0(\tilde{r}_{\ell m}^{(k)}) = \sum_{m=0}^{n-1} p_0(x_{\ell m})p(\tilde{r}_{\ell m}^{(k)}|x_{\ell m}) \quad \text{for } k=1,2,\ldots,8 \qquad (5.8)$$

In (5.8) $p_0(x_{\ell m})$ is the a priori probability of transmitting symbol $x_{\ell m}$. It is assumed that both binary symbols are equally likely to be channel inputs.

Noting that there are two channel symbols per branch (n=2) explicit evaluation of the branch metric can now be performed,

$$d(\tilde{\underline{r}}_\ell, \underline{x}_\ell^{(j)}) = \sum_{m=1}^{2}\left[\log_2 \frac{p(\tilde{r}_{\ell m}|x_{\ell m}^{(j)})}{p_0(\tilde{r}_{\ell m})} - \frac{1}{2}\right] \qquad (5.9)$$

$$d(\underline{r}_\ell,\underline{x}_\ell^{(j)}) = \log_2\left[\frac{p(\tilde{r}_{\ell 1}|x_{\ell 1}^{(j)})}{p_0(\tilde{r}_{\ell 1})}\right] + \log_2\left[\frac{p(\tilde{r}_{\ell 2}|x_{\ell 2}^{(j)})}{p_0(\tilde{r}_{\ell 2})}\right] - 1 \qquad (5.10)$$

Evaluating each term of (5.10)

$$\log_2\left[\frac{p(\tilde{r}_{\ell i}|x_{\ell i}^{(j)})}{\sum_{m=0}^{1} p_0(x_{\ell m})p(\tilde{r}_{\ell i}|x_{\ell m})}\right] = \log_2\left[\frac{p(\tilde{r}_{\ell i}|x_{\ell i}^{(j)})}{\frac{1}{2}\, p(\tilde{r}_{\ell i}|0)+\frac{1}{2}\, p(\tilde{r}_{\ell i}|1)}\right] \text{ for } i=1,2, \qquad (5.11)$$

Since, $\tilde{r}_{\ell i}$ can be one of eight possible values, denote $\tilde{r}_{\ell i}^{(k)}$ to be the k^{th} quantizer output value. Then, rewriting the right-hand side of (5.11) yields,

$$\log_2\left[\frac{2p(\tilde{r}_{\ell i}^{(k)}|x_{\ell i}^{(j)})}{p(\tilde{r}_{\ell i}|x_{\ell i}^{(j)})+p(\tilde{r}_{\ell i}|\bar{x}_{\ell i}^{(j)})}\right]$$

where $\bar{x}_{\ell i}^{(j)}$ represents the complement of $x_{\ell i}^{(j)}$. The branch metric expression now becomes

$$d(\tilde{\underline{r}}_\ell^{(\underline{k})},\underline{x}_\ell^{(j)}) = \log_2\left[\frac{2p(\tilde{r}_{\ell 1}^{(k_1)}|x_{\ell 1}^{(j)})}{p(\tilde{r}_{\ell 1}^{(k_1)}|x_{\ell 1}^{(j)})+p(\tilde{r}_{\ell 1}^{(k_1)}|\bar{x}_{\ell 1}^{(j)})}\right] + \log_2\left[\frac{2p(\tilde{r}_{\ell 2}^{(k_2)}|x_{\ell 2}^{(j)})}{p(\tilde{r}_{\ell 2}^{(k_2)}|x_{\ell 2}^{(j)})+p(\tilde{r}_{\ell 2}^{(k_2)}|\bar{x}_{\ell 2}^{(j)})}\right] - 1 \qquad (5.12)$$

ORIGINAL PAGE IS
OF POOR QUALITY

Eqn. (5.12) can be written more simply due to symmetry properties of the quantizer and resulting transition probabilities.

Define: $$TP(K) \triangleq p(\bar{r}_{\ell 1}^{(k_1)} | x_{\ell 1}^{(j)}) \quad (5.13a)$$

$$TP(9-K) \triangleq p(\bar{r}_{\ell 1}^{(k_1)} | \bar{x}_{\ell 1}^{(j)}) \quad (5.13b)$$

$$TP(J) \triangleq p(\bar{r}_{\ell 2}^{(k_2)} | x_{\ell 2}^{(j)}) \quad (5.13c)$$

$$TP(9-J) \triangleq p(\bar{r}_{\ell 2}^{(k_2)} | \bar{x}_{\ell 2}^{(j)}) \quad (5.13d)$$

Branch Metric: $$d(\bar{\underline{r}}_{\ell}^{(\underline{k})}, \underline{x}_{\ell}^{(j)}) = \log_2 \left[\frac{2TP(K)}{TP(K)+TP(9-K)}\right] + \log_2 \left[\frac{2TP(J)}{TP(J)+TP(9-J)}\right] - 1 \quad (5.14)$$

Eqn. (5.14) holds for K=1,2,...,8; J=1,2,...,8. There are 64 possibilities arising from noting that individual branches are represented by a pair of channel symbols each of which is quantized to one of eight possible values. Each value determined by (5.14) is being pre-computed and entered into a table. The decoder makes use of the table to look-up the branch metric value associated with a particular branch at the current node. The look-up table is structured as shown in Fig. 11 .

5.2 Parameters Effecting Sequential Decoder Performance

An important and useful parameter in characterizing the performance of the Fano sequential decoding algorithm is the critical rate R_0 (often referred to as the computational cutoff rate R_{comp} in this context) which for coherent BPSK modulation and equally probable signaling is given by (cf. [3], Chap.6)

$$R_0 = 1 - \log_2 [1 + \int_{-\infty}^{\infty} \sqrt{p\{r|x=1\} p\{r|x=-1\}}\, dr] \quad (5.15)$$

where $p\{r|x=i\}$ is the aposteriori probability of having observed a receiver output value r during any one channel signaling interval given that the corresponding transmitted symbol was $x=i$ for $i = \pm 1$. The decoder makes use of soft decisions on the receiver outputs, hence for quantized receiver outputs, the value of R_0 must be computed according to

$$R_0 = 1 - \log_2[1+\sum_{j=1}^{8}\sqrt{p\{\tilde{r}=q_j|x=1\}p\{\tilde{r}=q_j|x=-1\}}\ ] \tag{5.16}$$

where now $p\{\tilde{r}=q_j|x=i\}$ is the a posteriori probability that the quantized receiver output assumes the j^{th} quantization level q_j; $j=1,2,\ldots,8$ during any signaling interval given that the corresponding transmitted symbol was $x=i$ with $i=\pm 1$. and is given by (5.7)

The convolutional code rate R, in terms of information bits per channel use, normalized to the critical rate R_0 provides an indication of the behavior of the average number of computations required per decoded information digit. The parameter R/R_0 characterizes sequential decoder operation in terms of the channel properties and the quantizer characteristics. A plot of R/R_0 versus E_s/N_0 in dB is shown in Fig. 12. For the system of interest R=1/2, and the quantizer is both symmetric and uniform with input bin boundary spacing $\Delta=0.5$ (normalized in standard deviations of the system noise). It is noted that for values of E_s/N_0 such that $R/R_0<1$ the Fano algorithm will with high probability, decode a frame successfully since the average number of computations required per decoded information bit is finite. Further, when E_s/N_0 is such that $R/R_0\geq 1$ the average number of computations required to decode a frame becomes unbounded. Table II contains selected values of E_s/N_0 in dB and the corresponding values of R/R_0 are provided along with a tabulation of the channel transition probabilities for each case.

A series of simulations were performed for the Fano decoder used in this program [9].

5.3 Fano Algorithm Subroutine Description

The Fano sequential decoding algorithm has been designed as a subroutine which is called by the main driver program. The Fortran code appears in Appendix B. The program sequentially decodes frames of 1024 channel symbols using soft-decisions on the demodulated channel output. Each frame corresponds to 512 information bits of which the last 24 bits is the known frame sync word. This is used by the decoder as a known tail.sequence. Decoding into the tail allows for reliable decoding of the last data information bit. The call to the routine takes the form-

CALL FANO (ESNODB, IN, IOUT, ITCT, ACB, NSC, IDFFLG, ITPFLG)

Where, ESNODB = signal energy to noise density ratio in dB.

IN = input array of 1024 soft-decisions that are input to the Fano decoder.

IOUT = Output array of 512 decoded information bits.

ITCT = on entrance to routine: maximum number of computations allowed per frame desired by user (default is 100,000); on exit from routine: number of forward and reverse moves made by the decoder in processing a frame.

ACB = average number of computations per decoded bit.

NSC = number of corrected channel symbols (equals zero when frame is deleted).

IDFFLG = deleted frame flag:

0 = frame was successfully decoded

1 = frame deleted because the maximum number of computations allowed per frame has been exceeded.

2 = invalid decoder parameter has been encountered.

ITPFLG = flag indicating whether output is written to magnetic tape.

The decoder calculates the channel transition probabilities from the signal-to-noise ratio read from the start of each telemetry frame.

The implicit assumption is that the received sequence has been quantized using a uniform quantizer, wherein the input bin boundaries have been normalized to the standard deviation of the noise power. The value of $\Delta=0.5$ has been found to be near optimum [4]. Having the transition probabilities the decoder proceeds to calculate the branch metrics according to (5.14). The branch metrics are scaled to allow integer computations in the decoding process. The branch metric values are stored in two arrays as shown in Fig. 11. These arrays are accessed by an index which is detemined from the pair of received channel symbols representing a particular branch in the code tree. The IN array contains 1024 values in the range 1 to 8, representing the quantizer levels, the received channel symbols have been assigned.

At the start of each frame the decoder is initially set to zero. The decoding process continues until either the frame is decoded successfully or the frame is deleted. Upon successfully decoding a frame the subroutine returns the average number of computations per bit. This is calculated by dividing the number of forward and reverse moves by the number of information bits in a frame. Also, the number of corrected channel symbols is returned by the subroutine. This is determined by comparing the branch labels determined by the replica of the encoder, which is embedded in the Fano algorithm, with hard-decisions made on the channel output symbols contained in the IN array. A frame is deleted if the maximum number of computations is exceeded in attempting to decode a particular frame. If the Printout option is specified, the bits declared to be decoded will be printed out. Undecoded bits will be indicated by a "9" acting as a place holder in the data bit sequence. The frame sync word (bits 489-512) will be printed out for reference. For example, when a frame deletion is encountered the partially decoded frame will appear on the print-out as follows -

```
FRAME WAS DECODED USING FANO ALGORITHM.

FRAME WAS DELETED.

NUMBER OF SYMBOL ERRORS CORRECTED=    0

AVERAGE NUMBER OF COMPUTATIONS PER BIT=    210.9831085
(# BITS DECODED/MAX. # COMP. PER FRAME)

DECODED DATA-

1001100100110111000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000001000000010000001
0101010000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000101
0011001001010001010100001199999999999999999999999999999999999999

                    TAIL SEQUENCE SHOULD BE-  111110001100010101001001
```

The purpose of printing out deleted frames is to allow the user to re-attempt decoding the particular frame after increasing the maximum number of computations allowed per frame. By making several decoding trials on a difficult frame the user may be able to determine which bits in the partially decoded frame <u>may</u> possess some degree of reliability as the decoder progresses into the known tail sequence. It should be noted, of course, that the accuracy of any and all bits in a partially decoded frame is highly questionable. Furthermore, computer CPU processing time will increase considerably on an IBM 360/67 as the number of computations parameter is increased for frames containing a significant number of channel symbol errors. If the Tape Output option is specified, a deleted frame will be recorded on tape such that bits 193-680 are all zeros and bits 681-704 of a Data Block contain the frame sync word. The Data Block is a data record contained in the Intermediate Data Record (IDR) file. Frame deletion will also occur if the decoder encounters an invalid parameter during execution, and in the print-out made an error message to that effect is printed.

ORIGINAL PAGE IS
OF POOR QUALITY

References

1.) Massey, J. L. and Costello, D.J., "Nonsystematic Convolutional Codes for Sequential Decoding in Space Applications", IEEE Trans. Comm. Tech., Vol. COM-19, No. 5, Oct. 1971, pp. 806-813.

2.) Jacobs, I.M., "Sequential Decoding for Efficient Communication from Deep Space", IEEE Trans. on Comm. Tech., Vol. COM-15, No. 4, August 1967, pp.492-501.

3.) Wozencraft, J.M. and Jacobs, I.M., Principles of Communication Engineering, John Wiley and Sons, New York, 1965.

4.) Modestino, J. W., Mui, S. Y., Reale, T. J., "Performance of Convolutional Codes on Fading Channels Typical of Planetary Entry Missions", NASA Contract No. NGR33-018-188, Report covering the period June 1, 1973 to May 31, 1974.

5.) Fano, R. M., "A Heuristic Discussion of Probabilistic Decoding", IEEE Trans. on Inform. Theory, Vol. IT-9, April 1963, pp. 64-74.

6.) Jordan, K.L., Jr., "Performance of Sequential Decoding in Conjunction with Efficient Modulation", IEEE Trans. Comm. Tech. Vol. COM-14, June 1966.

7.) Layland, J.W. and Lushbaugh, W. A., "A Flexible High-Speed Sequential Decoder for Deep Space Channels", IEEE Trans. Comm. Tech. Vol. COM-19, No. 5, October 1971, pp. 813-820.

8.) Savage, J.E., "Progress in Sequential Decoding", Advances in Communication Systems, Academic Press, 1968, pp. 149-201.

9.) Daut, D.G., "Fano Sequential Decoding Algorithm for the NASA Pioneer-Venus Reverse Playback Telemetry Program", unpublished internal RPI report.

LOGICAL UNIT NUMBER	INPUT-OUTPUT DEVICE
1	user terminal input
2	user terminal output
3	print output
4	tape input
5	tape output
6	random access input and output

Table 1

Logical I/O Device Assignments

```
SEQUENTIAL DECODING PARAMETERS FOR OCTAL CHANNEL --   CODE RATE = 0.50     QSP = 0.50

ESNODB =-4.00000     ESNO = 0.39811     R/RO = 2.00146

SIMULATED CHANNEL CROSSOVER PROBABILITY = 0.18611

TRANSITION PROBABILITIES = 0.27170 0.18542 0.19546 0.16130 0.10420 0.05269 0.02085 0.00837

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

ESNODB =-3.50000     ESNO = 0.44668     R/RO = 1.80683

SIMULATED CHANNEL CROSSOVER PROBABILITY = 0.17228

TRANSITION PROBABILITIES = 0.28951 0.18863 0.19376 0.15581 0.09808 0.04833 0.01864 0.00724

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

ESNODB =-3.00000     ESNO = 0.50119     R/RO = 1.63387

SIMULATED CHANNEL CROSSOVER PROBABILITY = 0.15837

TRANSITION PROBABILITIES = 0.30896 0.19152 0.19141 0.14975 0.09171 0.04397 0.01650 0.00619

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

ESNODB =-2.50000     ESNO = 0.56234     R/RO = 1.48028

SIMULATED CHANNEL CROSSOVER PROBABILITY = 0.14446

TRANSITION PROBABILITIES = 0.33015 0.19397 0.18831 0.14311 0.08514 0.03964 0.01445 0.00523

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

ESNODB =-2.00000     ESNO = 0.63096     R/RO = 1.34404

SIMULATED CHANNEL CROSSOVER PROBABILITY = 0.13064

TRANSITION PROBABILITIES = 0.35322 0.19587 0.18439 0.13588 0.07839 0.03540 0.01251 0.00435
```

TABLE II

```
ESNODB =-1.50000      ESNO = 0.70795      R/RO = 1.22335

SIMULATED CHANNEL CROSSOVER PROBABILITY = 0.11704

TRANSITION PROBABILITIES = 0.37825 0.19706 0.17956 0.12808 0.07152 0.03126 0.01069 0.00357

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

ESNODB =-1.00000      ESNO = 0.79433      R/RO = 1.11660

SIMULATED CHANNEL CROSSOVER PROBABILITY = 0.10376

TRANSITION PROBABILITIES = 0.40533 0.19740 0.17377 0.11974 0.06459 0.02727 0.00901 0.00289

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

ESNODB =-0.50000      ESNO = 0.89125      R/RO = 1.02236

SIMULATED CHANNEL CROSSOVER PROBABILITY = 0.09092

TRANSITION PROBABILITIES = 0.43451 0.19671 0.16694 0.11091 0.05768 0.02348 0.00748 0.00229

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

ESNODB = 0.0          ESNO = 1.00000      R/RO = 0.93941

SIMULATED CHANNEL CROSSOVER PROBABILITY = 0.07865

TRANSITION PROBABILITIES = 0.46582 0.19482 0.15906 0.10165 0.05085 0.01991 0.00610 0.00178

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

ESNODB = 0.50000      ESNO = 1.12202      R/RO = 0.86661

SIMULATED CHANNEL CROSSOVER PROBABILITY = 0.06707

TRANSITION PROBABILITIES = 0.49921 0.19156 0.15010 0.09207 0.04421 0.01661 0.00489 0.00136

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

ESNODB = 1.00000      ESNO = 1.25893      R/RO = 0.80300

SIMULATED CHANNEL CROSSOVER PROBABILITY = 0.05628

TRANSITION PROBABILITIES = 0.53457 0.18675 0.14011 0.08229 0.03783 0.01361 0.00383 0.00101
```

Table II continued

ESNODB = 1.50000 ESNO = 1.41254 R/RO = 0.74771

SIMULATED CHANNEL CROSSOVER PROBABILITY = 0.04640

TRANSITION PROBABILITIES = 0.57174 0.18026 0.12916 0.07244 0.03180 0.01093 0.00294 0.00073

- -

ESNODB = 2.00000 ESNO = 1.58489 R/RO = 0.69998

SIMULATED CHANNEL CROSSOVER PROBABILITY = 0.03751

TRANSITION PROBABILITIES = 0.61041 0.17201 0.11738 0.06270 0.02621 0.00858 0.00220 0.00052

- -

ESNODB = 2.50000 ESNO = 1.77828 R/RO = 0.65910

SIMULATED CHANNEL CROSSOVER PROBABILITY = 0.02966

TRANSITION PROBABILITIES = 0.65021 0.16195 0.10495 0.05324 0.02114 0.00657 0.00160 0.00035

- -

ESNODB = 3.00000 ESNO = 1.99526 R/RO = 0.62448

SIMULATED CHANNEL CROSSOVER PROBABILITY = 0.02288

TRANSITION PROBABILITIES = 0.69063 0.15014 0.09211 0.04424 0.01663 0.00489 0.00113 0.00024

- -

ESNODB = 3.50000 ESNO = 2.23872 R/RO = 0.59551

SIMULATED CHANNEL CROSSOVER PROBABILITY = 0.01717

TRANSITION PROBABILITIES = 0.73105 0.13674 0.07916 0.03587 0.01272 0.00353 0.00077 0.00015

- -

ESNODB = 4.00000 ESNO = 2.51189 R/RO = 0.57165

SIMULATED CHANNEL CROSSOVER PROBABILITY = 0.01250

TRANSITION PROBABILITIES = 0.77077 0.12200 0.06642 0.02831 0.00944 0.00246 0.00050 0.00009

Table II continued

ESNODB = 4.50000   ESNO = 2.81838   R/RO = 0.55241

SIMULATED CHANNEL CROSSOVER PROBABILITY = 0.00879

TRANSITION PROBABILITIES = 0.80899 0.10632 0.05424 0.02166 0.00677 0.00166 0.00032 0.00005

- -

ESNODB = 5.00000   ESNO = 3.16228   R/RO = 0.53721

SIMULATED CHANNEL CROSSOVER PROBABILITY = 0.00595

TRANSITION PROBABILITIES = 0.84491 0.09018 0.04294 0.01601 0.00467 0.00107 0.00019 0.00003

- -

ESNODB = 5.50000   ESNO = 3.54813   R/RO = 0.52551

SIMULATED CHANNEL CROSSOVER PROBABILITY = 0.00386

TRANSITION PROBABILITIES = 0.87776 0.07417 0.03283 0.01137 0.00308 0.00065 0.00011 0.00002

- -

ESNODB = 6.00000   ESNO = 3.98107   R/RO = 0.51691

SIMULATED CHANNEL CROSSOVER PROBABILITY = 0.00239

TRANSITION PROBABILITIES = 0.90687 0.05088 0.02412 0.00774 0.00194 0.00038 0.00006 0.00001

- -

ESNODB = 6.50000   ESNO = 4.46683   R/RO = 0.51072

SIMULATED CHANNEL CROSSOVER PROBABILITY = 0.00140

TRANSITION PROBABILITIES = 0.93175 0.04490 0.01695 0.00501 0.00116 0.00021 0.00003 0.00000

- -

ESNODB = 7.00000   ESNO = 5.01187   R/RO = 0.50660

SIMULATED CHANNEL CROSSOVER PROBABILITY = 0.00077

TRANSITION PROBABILITIES = 0.95215 0.03270 0.01132 0.00307 0.00065 0.00011 0.00001 0.00000

Table II continued

PIONEER VENUS MULTIPROBE MISSION
TELEMETRY DATA RECOVERY AND INTERFEROMETRY EXPERIMENT

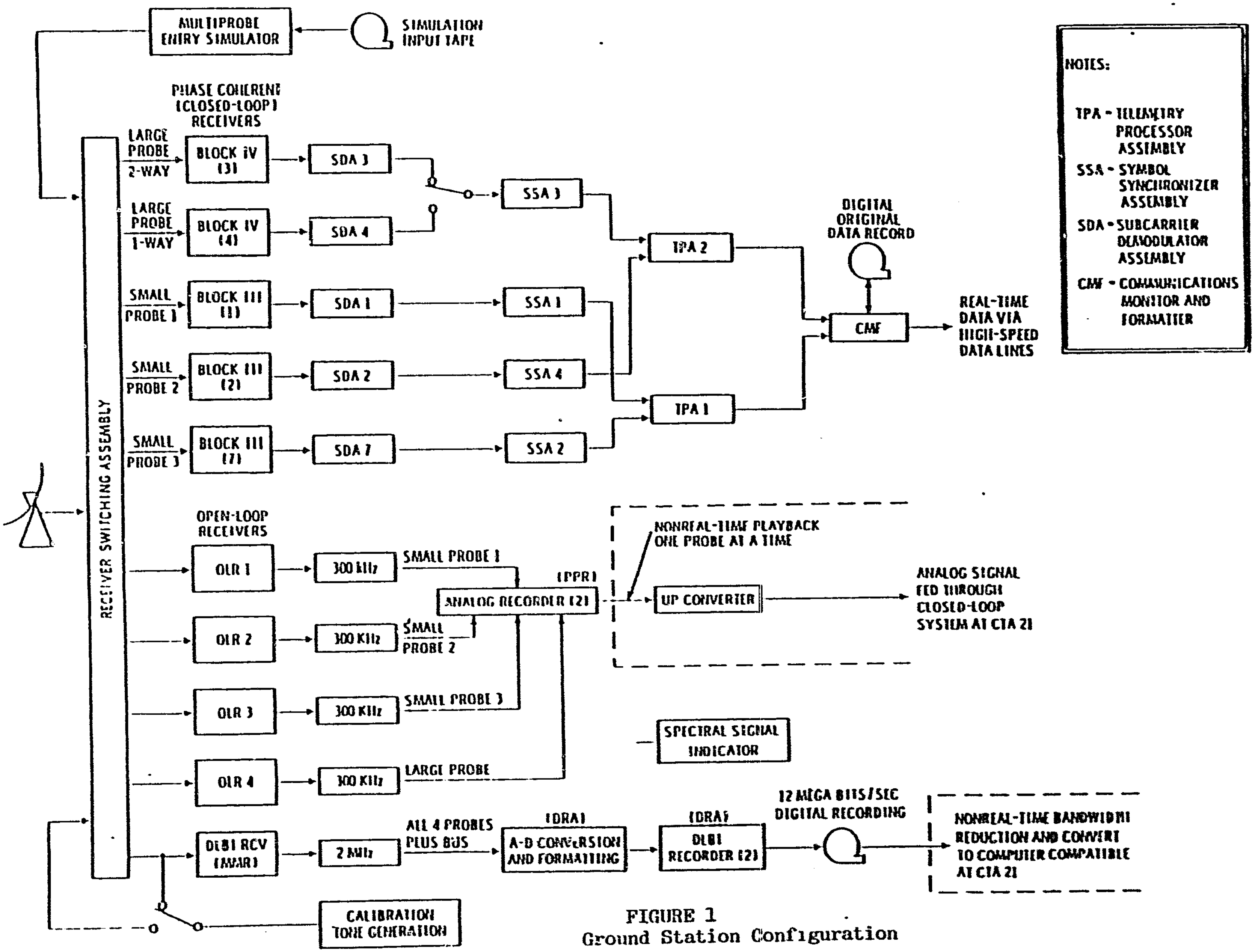


FIGURE 1
Ground Station Configuration

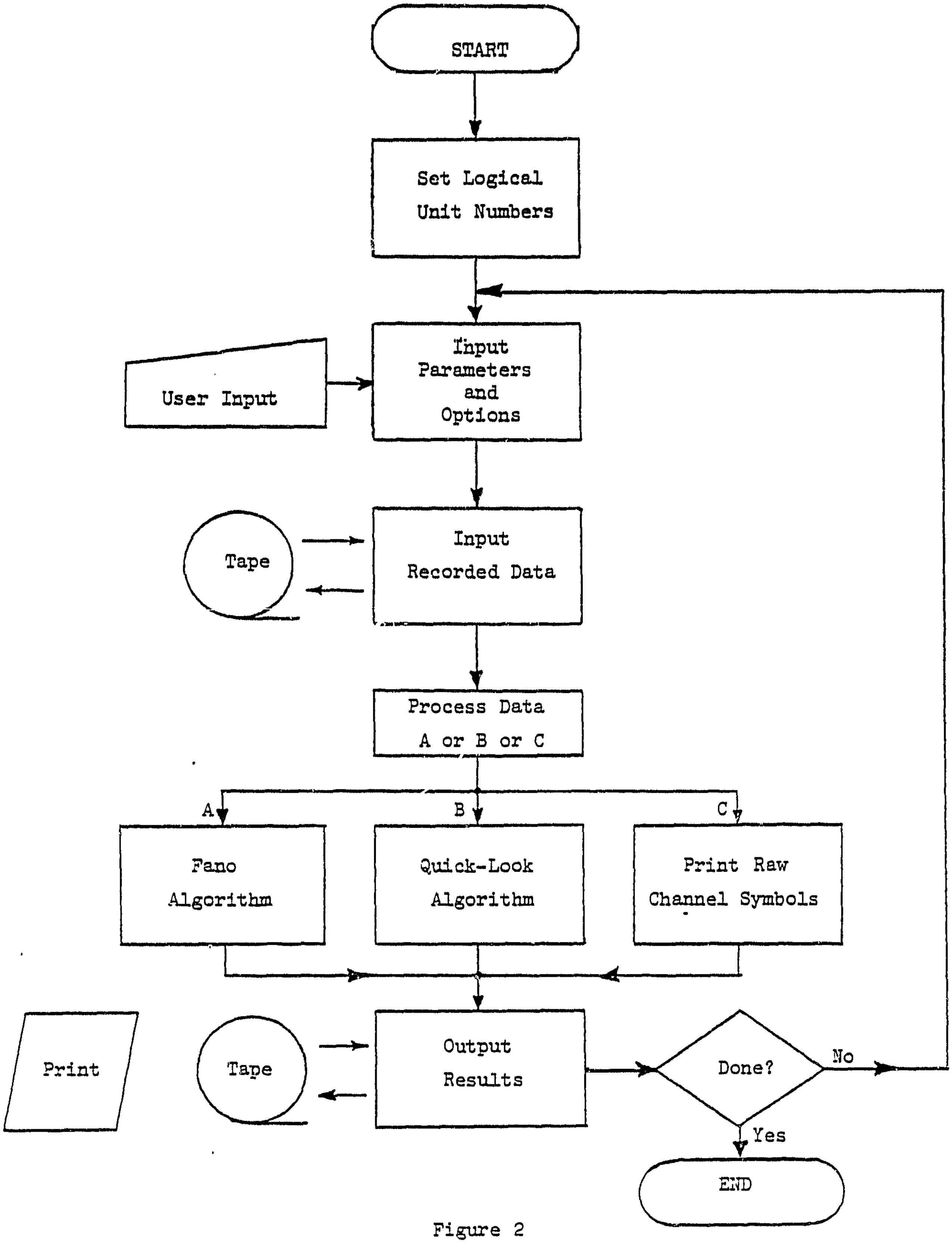


Figure 2

Flowchart for Main Program

```
RUN -LOAD+ANOE:FILE1 1=*SOURCE* 2=*SINK* 3=*PRINT* 4=*T* 5=*TO* 6=-D
#EXECUTION BEGINS

IS DATA RECORDED ON TAPE IN FORWARD OR REVERSE DIRECTION (FWD OR REV)?
REV

DO YOU WANT TAPE OUTPUT OR PRINT OUTPUT (TAPE OR PRNT)?
PRNT

DO YOU WANT PRINTOUT TO THE TERMINAL OR TO THE LINE PRINTER
(TERM OR LPRNT)?
TERM

WHICH PHASE REFERENCE DO YOU WANT-  DATA OR DATA-BAR (D OR DB)?
DB

IS THE DATA TO BE DECODED OR IS RAW SYMBOL OUTPUT DESIRED (DEC OR RAW)?
RAW

DO YOU WANT RAW SYMBOLS TO BE SYNCHRONIZED (1024) IN A DATA FRAME OR
THOSE (1025) CORRESPONDING TO A TAPE RECORD (SYNC OR TREC)?
TREC

DO YOU WANT ALL THE FRAMES PRINTED OUT (Y OR N)?
N

DO YOU WANT TO SPECIFY THE FRAMES TO BE PROCESSED ACCORDING TO
TIME TAG OR FRAME NUMBER (TIME OR NUMB)?
NUMB

SPECIFY BY NUMBER THE FRAMES WHICH ARE TO BE PROCESSED.
THE LAST DIGIT OF EACH NUMBER MUST END UP IN COLUMN 5.
INDICATE YOU ARE DONE BY HITTING A RETURN.
   20
   21


ENTER DSS NUMBER (14 OR 43)-
14

ENTER PROBE ID (SP1,SP2,SP3 OR LP)-
SP1

 278 RECORDS HAVE BEEN READ FROM INPUT TAPE




     PIONEER VENUS REVERSE PLAYBACK TELEMETRY PROGRAM


     PROBE IDENTIFICATION-  SP1


     DSS NUMBER-  DSS 14


     PLAYBACK DAY OF YEAR-  89
```


Figure 3
Sample Output of Program

```
DATA START TIME-
    HOURS-   17
    MINUTES- 53
    SECONDS- 39.680

PRINTOUT OF RAW CHANNEL SYMBOLS FOR TAPE FRAME (RECORD)   20

DATA START TIME-
    HOURS-   17
    MINUTES- 53
    SECONDS-  1.680

DATA STOP TIME-
    HOURS-   17
    MINUTES- 52
    SECONDS- 59.680

SIGNAL-TO-NOISE RATIO (DB)=  6.703125000

RAW CHANNEL SYMBOLS (SOFT-DECISIONS IN OCTAL FORMAT)-

3524352525352434243535240426251434242424341425363524261635353525
1626351526150534061517263425252504001425142624251514252634204235
4541264463323432663242462142524415341435315240542712434434204434
3255016245243516141517352415241534262435352405261615342534342525
3505172525251514262425141535261415352414353425253435241000062405
1425242425350624362574063525263624261634353616142626262535163414
3434162524151424243426152434242616361526242426251525350416243526
1514252414342626342426743414142514162424242424350425242434253625
2435251434371514253524343434152415353414242634343535351625143535
2534753514153635143534253532511455633322434253344264353444211344
4344122433515463424466434343006003312313524325325616263145352434
3346324420552254326423552731553526235622551434343524353525253635
1626252634351535242424343424251424353625342435252515352525353424
2534253724151425160434261524343626143425351534242414242426270626
1624242634352536241534341426242636042424242526142424142425251524
15352605051615261424241424242536041524352624163534342525350505141

RAW CHANNEL SYMBOLS (HARD-DECISIONS IN BINARY FORMAT)-

1010101010101010101010101010101010101010101010101010101010101010
1010101010101010101010101010101010111010101010101010101010110110
0001100001111011001101001101010010101010110101001011010010110010
1100110100101010101010101010101010101010101010101010101010101010
1010101010101010101010101010101010101010101010101010101111101010
1010101010101010101000101010101010101010101010101010101010101010
1010101010101010101010101010101010101010101010101010101010101010
1010101010101010101010001010101010101010101010101010101010101010
1010101010101010101010101010101010101010101010101010101010101010
101000101010101010101010101011011000011111010101100100101000111100
0100111011010001010000010101110111111111101011011001010110010101 0
1100110011001100110011001011001010110011001010101010101010101010
1010101010101010101010101010101010101010101010101010101010101010
1010101010101010101010101010101010101010101010101010101010101010
1010101010101010101010101010101010101010101010101010101010101010
10101010101010101010101010101010101010101010101010101010101010101
```

 Figure 3 - cont.

```
PRINTOUT OF RAW CHANNEL SYMBOLS FOR TAPE FRAME (RECORD)    21

DATA START TIME-
     HOURS-   17
     MINUTES- 52
     SECONDS- 59.680


DATA STOP TIME-
     HOURS-   17
     MINUTES- 52
     SECONDS- 57.680

SIGNAL-TO-NOISE RATIO (DB)=   6.703125000

RAW CHANNEL SYMBOLS (SOFT-DECISIONS IN OCTAL FORMAT)-

1634252425241535340536141424152436252525353434243534372526243634
2525352534142505350634343425172435241535143435253525251504334657
3643633521654246622354143526352513403353562623514262524243352713
4244052243425354342415161315241525241435353536363525143525343635
3436262435351405242504142435363416362526242425243614361424252534
1535353624353634242534263435253435153434353534372524352714241435
1424262526243535243526242424343537143434253415352434353506253524
1516263414153535361670342525252534342424353425241524262535262434
2434241514362535363634353634263425151434142515162524243414243525
3535343415253524342736242422401547612323635241345346242545331355
4245122423624442525545524353235512333206165622462612424354225343
5424252163642242653433660435235425251365150534143434063735353625
2535353625351405272535101425243424342535052626341436242504353415
2524253625243634261515242514342425253434353634353515253425351527
2424343515240534362636253525152725261535343424242426241536341735
24152424243615250524252536151614152434362515163434361414162636252


RAW CHANNEL SYMBOLS (HARD-DECISIONS IN BINARY FORMAT)-

1010101010101010101010101010101010101010101010101010101010101010
1010101010101010101010101010101010101010101010101010101010110000
1001011011000100011100101010101011011101001011010101010100101011
0100101101010100101010101010101010101010101010101010101010101010
1010101010101010101010101010101010101010101010101010101010101010
1010101010101010101010101010101010101010101010101010101010101010
1010101010101010101010101010101010101010101010101010101010101010
1010101010101010101001101010101010101010101010101010101010101010
1010101010101010101010101010101010101010101010101010101010101010
1010101010101010101010101011011000011111010101100100101000111100
0100111011010001010000010101110011111110100011001011010100110101
0010101101001101001011001010110010101100101010101010101010101010
1010101010101010101010101110101010101010101010101010101010101010
1010101010101010101010101010101010101010101010101010101010101010
1010101010101010101010101010101010101010101010101010101010101010
10101010101010101010101010101010101010101010101010101010101010101


                    PROGRAM SUMMARY

TOTAL NUMBER OF FRAMES PROCESSED=        2
```

OF POOR QUALITY

```
ARE YOU DONE (Y OR N)?
N

DO YOU WANT TAPE OUTPUT OR PRINT OUTPUT (TAPE OR PRNT)?
PRNT

DO YOU WANT PRINTOUT TO THE TERMINAL OR TO THE LINE PRINTER
(TERM OR LPRNT)?
TERM

WHICH PHASE REFERENCE DO YOU WANT-  DATA OR DATA-BAR (D OR DB)?
DB

IS THE DATA TO BE DECODED OR IS RAW SYMBOL OUTPUT DESIRED (DEC OR RAW)?
RAW

DO YOU WANT RAW SYMBOLS TO BE SYNCHRONIZED (1024) IN A DATA FRAME OR
THOSE (1025) CORRESPONDING TO A TAPE RECORD (SYNC OR TREC)?
SYNC

DO YOU WANT ALL THE FRAMES PRINTED OUT (Y OR N)?
N

DO YOU WANT TO SPECIFY THE FRAMES TO BE PROCESSED ACCORDING TO
TIME TAG OR FRAME NUMBER (TIME OR NUMB)?
NUMB

SPECIFY BY NUMBER THE FRAMES WHICH ARE TO BE PROCESSED.
THE LAST DIGIT OF EACH NUMBER MUST END UP IN COLUMN 5.
INDICATE YOU ARE DONE BY HITTING A RETURN.
   20

      PIONEER VENUS REVERSE PLAYBACK TELEMETRY PROGRAM


      PROBE IDENTIFICATION-  SP1


      DSS NUMBER-  DSS 14


      PLAYBACK DAY OF YEAR-  89


      DATA START TIME-
          HOURS-   17
          MINUTES- 53
          SECONDS- 39.680


      PRINTOUT OF RAW CHANNEL SYMBOLS FOR SYNCHRONIZED DATA FRAME   20

      DATA START TIME-
          HOURS-   17
          MINUTES- 53
          SECONDS-  1.680


      DATA STOP TIME-
          HOURS-   17
          MINUTES- 52
          SECONDS- 59.680
```


Figure 3 - cont.

SYNC WORD LOCATION= 326

SIGNAL-TO-NOISE RATIO (DB)= 6.703125000

SYNCHRONIZED RAW CHANNEL SYMBOLS (SOFT-DECISIONS IN OCTAL FORMAT)-

```
1546342446643434300600331231352432532561626314535243433463244205
5225432642355273155352623562255143434352435352525363516262526343
5153524242434342425142435362534243525251535252535342425342537241
5142516043426152434362614342535153424241424242627062616242426343
5253624153434142624263604242424252614242414242525152415352605051
6152614242414242425360415243526241635343425253505051416342524252
4153534053614142415243625252535343424353437252624363425253525341
4250535063434342517243524153514343525352525150433465736436335216
5424662235414352635251340335356262351426252424535271342440522434
7535434241516151524152524143535353636352514352534363534362624353
5140524250414243536341636252624242524361436142425253415353536243
5363424253426343525343515343435353437252435271424143514242625262
4353524352624242434353714343425341535243435350625352415162634141
5353536167034252525253434242435342524152426253526243424342415143
6253536363435363426342515143414251516252424341424352535353434152
5352434273624242240154761232363524134534624254533135542451224236
```

SYNCHRONIZED RAW CHANNEL SYMBOLS (HARD-DECISIONS IN BINARY FORMAT)-

```
1000101000000101011101111111111010110110010101100101010110011001 10
0110011001100101100101011001100101010101010101010101010101010101
0101010101010101010101010101010101010101010101010101010101010101
0101010101010101010101010101010101010101010101010101010101010101
0101010101010101010101010101010101010101010101010101010101010101
0101010101010101010101010101010101010101010101010101010101010101
0101010101010101010101010101010101010101010101010101010101010101
0101010101010101010101010101010101010101010101011000010010110110
0010001110010101010101101110100101101010101010010101101001011010
1010010101010101010101010101010101010101010101010101010101010101
0101010101010101010101010101010101010101010101010101010101010101
0101010101010101010101010101010101010101010101010101010101010101
0101010101010101010101010101010101010101010101010101010101010101
0101010100110101010101010101010101010101010101010101010101010101
0101010101010101010101010101010101010101010101010101010101010101
0101010101010101101100001111101010110010010100011110001001110110
```

PROGRAM SUMMARY

TOTAL NUMBER OF FRAMES PROCESSED= 1

ARE YOU DONE (Y OR N)?
N

DO YOU WANT TAPE OUTPUT OR PRINT OUTPUT (TAPE OR PRNT)?
PRNT

DO YOU WANT PRINTOUT TO THE TERMINAL OR TO THE LINE PRINTER
(TERM OR LPRNT)?
TERM

WHICH PHASE REFERENCE DO YOU WANT- DATA OR DATA-BAR (D OR DB)?
DB

Figure 3 -cont.

```
IS THE DATA TO BE DECODED OR IS RAW SYMBOL OUTPUT DESIRED (DEC OR RAW)?
DEC

DO YOU WANT THE FANO DECODER USED OR THE QUICK-LOOK OUTPUTS
(FANO OR QL)?
FANO

INPUT MAXIMUM NUMBER OF COMPUTATIONS PER FRAME.
THE LAST DIGIT OF THE NUMBER MUST END UP IN COLUMN 8.
0 INDICATES DEFAULT=100,000 COMPUTATIONS.
  125000

DO YOU WANT ALL THE FRAMES PRINTED OUT (Y OR N)?
N

DO YOU WANT TO SPECIFY THE FRAMES TO BE PROCESSED ACCORDING TO
TIME TAG OR FRAME NUMBER (TIME OR NUMB)?
NUV

DO YOU WANT TO SPECIFY THE FRAMES TO BE PROCESSED ACCORDING TO
TIME TAG OR FRAME NUMBER (TIME OR NUMB)?
NUMB

SPECIFY BY NUMBER THE FRAMES WHICH ARE TO BE PROCESSED.
THE LAST DIGIT OF EACH NUMBER MUST END UP IN COLUMN 5.
INDICATE YOU ARE DONE BY HITTING A RETURN.
  277


     PIONEER VENUS REVERSE PLAYBACK TELEMETRY PROGRAM


     PROBE IDENTIFICATION-  SP1


     DSS NUMBER-  DSS 14


     PLAYBACK DAY OF YEAR-  89


     DATA START TIME-
         HOURS-   17
         MINUTES- 53
         SECONDS- 39.680



     PRINTOUT OF DECODED DATA FOR FRAME  277

     DATA START TIME-
         HOURS-   17
         MINUTES- 44
         SECONDS- 27.677
```


Figure 3 - cont.

```
DATA STOP TIME-
    HOURS-   17
    MINUTES- 44
    SECONDS- 25.677

SYNC WORD LOCATION=  326

SIGNAL-TO-NOISE RATIO (DB)= 0.5000000000

FRAME WAS DECODED USING FANO ALGORITHM.

FRAME WAS DECODED SUCCESSFULLY.

NUMBER OF SYMBOL ERRORS CORRECTED=    1

AVERAGE NUMBER OF COMPUTATIONS PER BIT=     1.0000000

DECODED DATA-

1001100100111010000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000011111111
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000111110001100010101001001


                PROGRAM SUMMARY

TOTAL NUMBER OF FRAMES PROCESSED=      1
TOTAL NUMBER OF FRAMES DELETED=        0
DELETION RATE=      0.0
SYMBOL ERROR RATE=  0.000977

ARE YOU DONE (Y OR N)?
N

DO YOU WANT TAPE OUTPUT OR PRINT OUTPUT (TAPE OR PRNT)?
TAPE

DO YOU WANT PRINTOUT TO THE TERMINAL OR TO THE LINE PRINTER
(TERM OR LPRNT)?
TERM

WHICH PHASE REFERENCE DO YOU WANT-  DATA OR DATA-BAR (D OR DB)?
DB

DO YOU WANT THE FANO DECODER USED OR THE QUICK-LOOK OUTPUTS
(FANO OR QL)?
FANO

INPUT MAXIMUM NUMBER OF COMPUTATIONS PER FRAME.
THE LAST DIGIT OF THE NUMBER MUST END UP IN COLUMN 8.
0 INDICATES DEFAULT=100,000 COMPUTATIONS.


ENTER TAPE SEQUENCE NUMBER-
12
```


Figure 3 - cont.

```
ENTER YEAR DATA WAS RECORDED (E.G. 78)-
78

PIONEER VENUS REVERSE PLAYBACK TELEMETRY PROGRAM


PROBE IDENTIFICATION-  SP1


DSS NUMBER-  DSS 14


PLAYBACK DAY OF YEAR-  89


DATA START TIME-
    HOURS-   17
    MINUTES- 47
    SECONDS-  7.678


FRAMES DECODED USING FANO ALGORITHM.

SYNC WORD LOCATION=  326

SIGNAL-TO-NOISE RATIO (DB)=  6.968750000

SIGNAL-TO-NOISE RATIO (DB)=  6.890625000

SIGNAL-TO-NOISE RATIO (DB)=  6.890625000

SIGNAL-TO-NOISE RATIO (DB)=  6.890625000

SIGNAL-TO-NOISE RATIO (DB)=  6.890625000

SIGNAL-TO-NOISE RATIO (DB)= 0.5000000000

SIGNAL-TO-NOISE RATIO (DB)= 0.5000000000

SIGNAL-TO-NOISE RATIO (DB)= 0.5000000000

FRAMES DECODED USING FANO ALGORITHM.

SYNC WORD LOCATION=  326

SIGNAL-TO-NOISE RATIO (DB)= 0.5000000000


              PROGRAM SUMMARY

TOTAL NUMBER OF FRAMES PROCESSED=      9
TOTAL NUMBER OF FRAMES DELETED=        0
DELETION RATE=      0.0
SYMBOL ERROR RATE=  0.001411

ARE YOU DONE (Y OR N)?
Y
#EXECUTION TERMINATED
```


Figure 3 - cont.

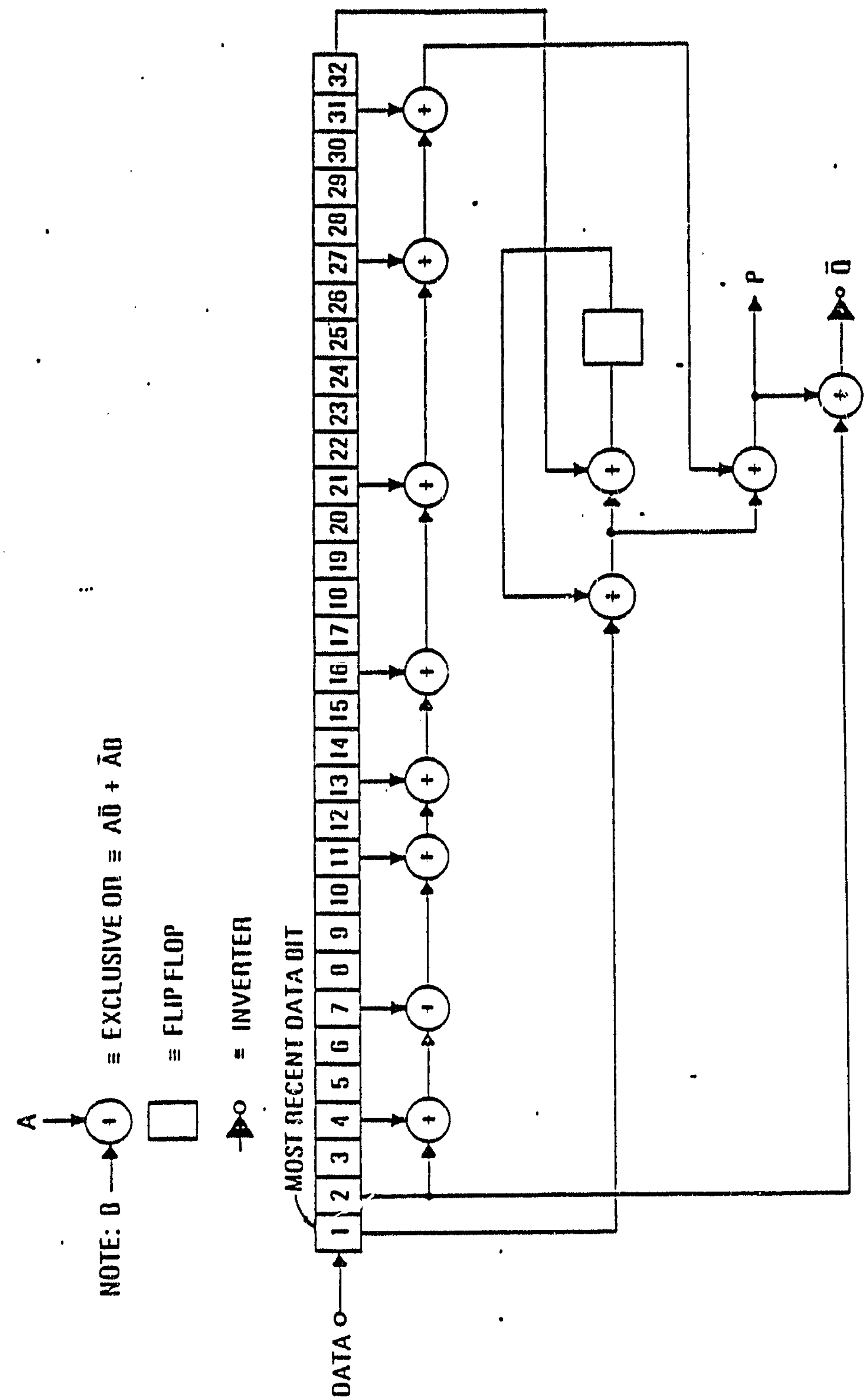


FIGURE 4

REPRODUCED FROM

TITLE

EQUIVALENT NONSYSTEMATIC
ENCODER

PIONEER PROGRAM
NASA
AMES RESEARCH CENTER
MOFFETT FIELD, CALIFORNIA

DOC. NO.	PC-410.01		
FIG.	3.9.8.2		
REV NO. 1	DATE 6-28-74	SHEET 1	OF 1

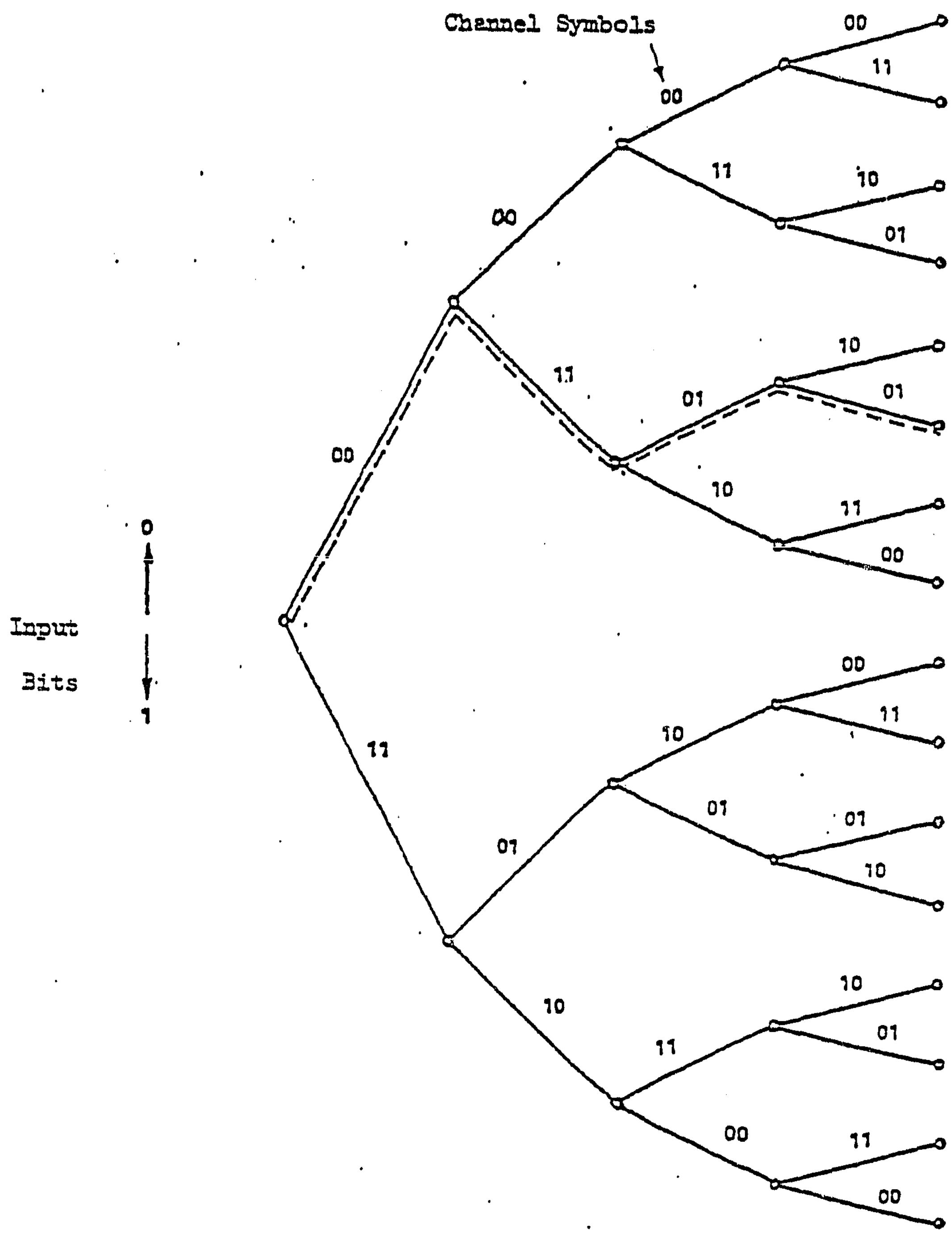


Figure 5

Typical Code Tree Structure

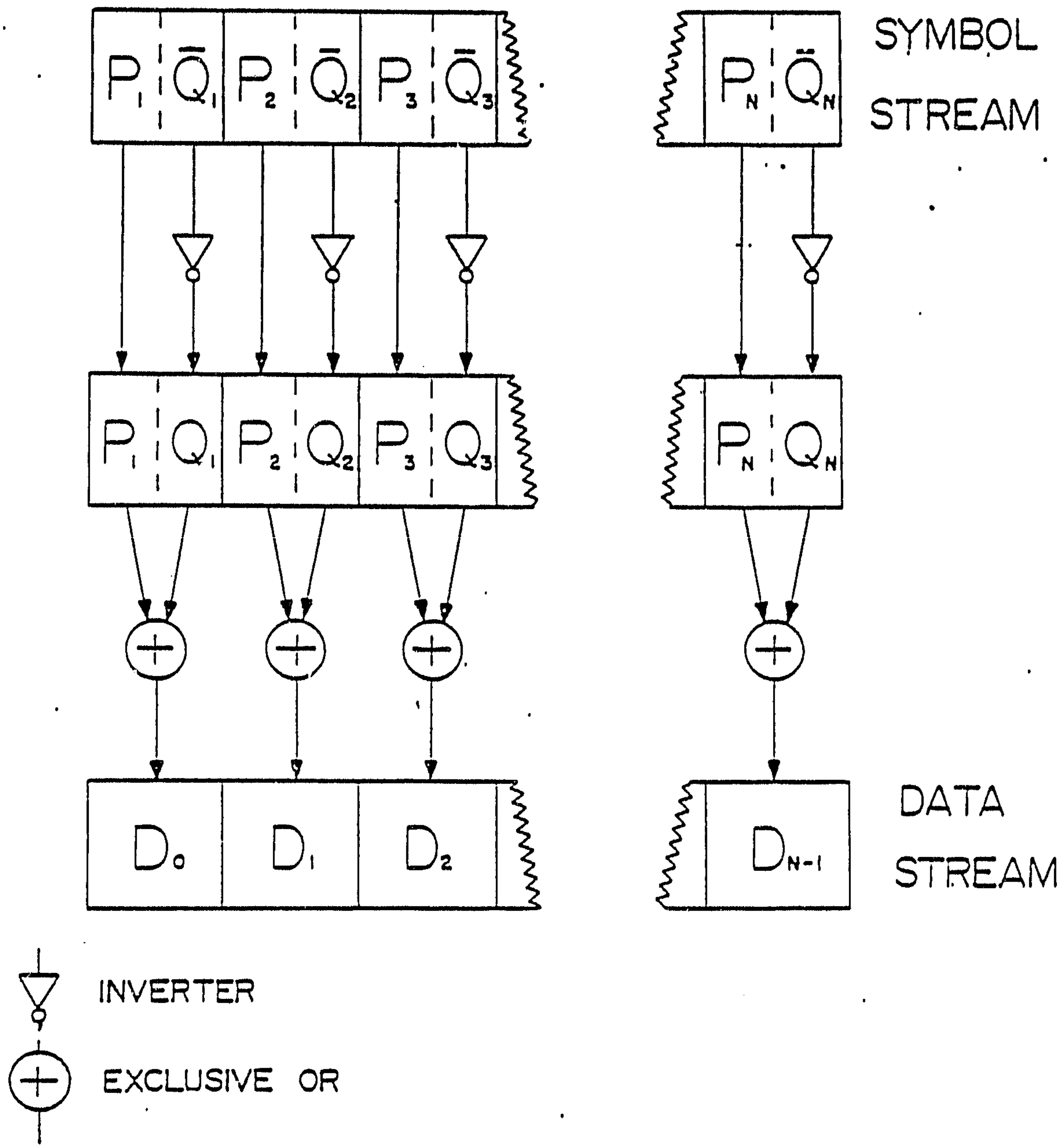
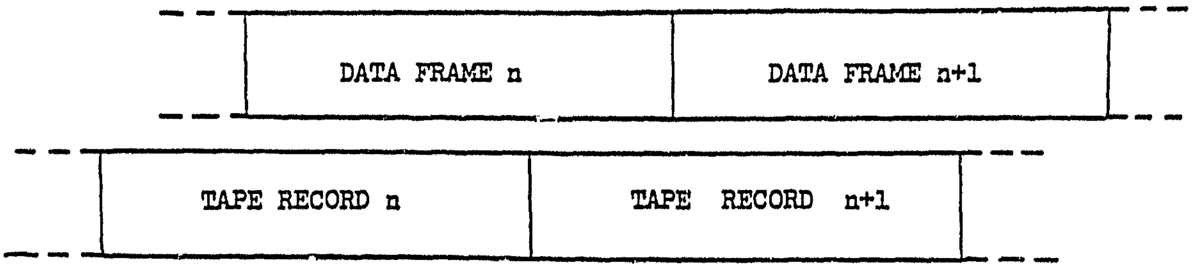


FIGURE 6

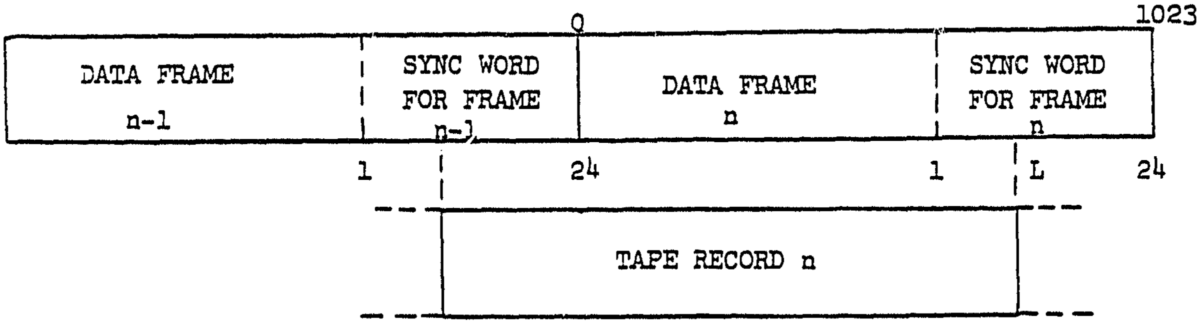
REPRODUCED FROM	TITLE	PIONEER PROGRAM NASA AMES RESEARCH CENTER MOFFETT FIELD, CALIFORNIA
	FUNCTIONAL DIAGRAM FOR RECONSTRUCTION OF DATA STREAM	DOC. NO. PC-410.01
		FIG. 6.3.1.1
	REV NO. 1 / DATE 6-28-74	SHEET 1 OF 1



a.) Overlap of data frames on tape records.

X	Y	(X*Y)
0	0	1
0	1	-1
1	0	-1
1	1	1

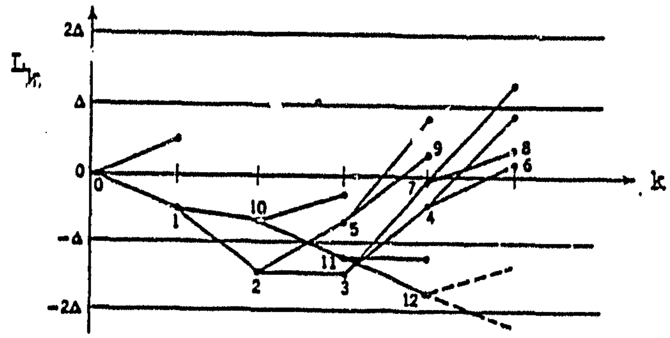
b.) Truth table for correlation operator.



c.) Overlap of sync word on tape record boundary.

Figure 7

Data Frame and Tape Record Relationship



$\Delta = T_0$

Pointer at Node	Running Threshold	θ	Action (× indicates "set $\theta = 1$")		
0	0	0	look at 1	point to 1	
1	0	0	look at 2	point to 2	set $T = -\Delta$
2	$-\Delta$	0	look at 3	point to 3	
3	$-\Delta$	0	look at 4	× look at 2	point to 2
2	$-\Delta$	1	look at 5	× look at 1	set $T = 0$
2	0	1	look at 3	point to 3	
3	0	1	look at 4	point to 4	set $\theta = 0$
4	0	0	look at 6	× look at 3	point to 3
3	0	1	look at 7	point to 7	set $\theta = 0$
7	0	0	look at 8	× look at 3	point to 3
3	0	1	look at 2	point to 2	
2	0	1	look at 5	point to 5	set $\theta = 0$
5	0	0	look at 9	× look at 2	point to 2
2	0	1	look at 1	point to 1	
1	0	1	look at 10	point to 10	set $\theta = 0$
10	0	0	look at 11	point to 11	set $T = -\Delta$
11	$-\Delta$	0	look at 12	point to 12	

Figure 8

An example of the searching procedure used by the Fano Algorithm

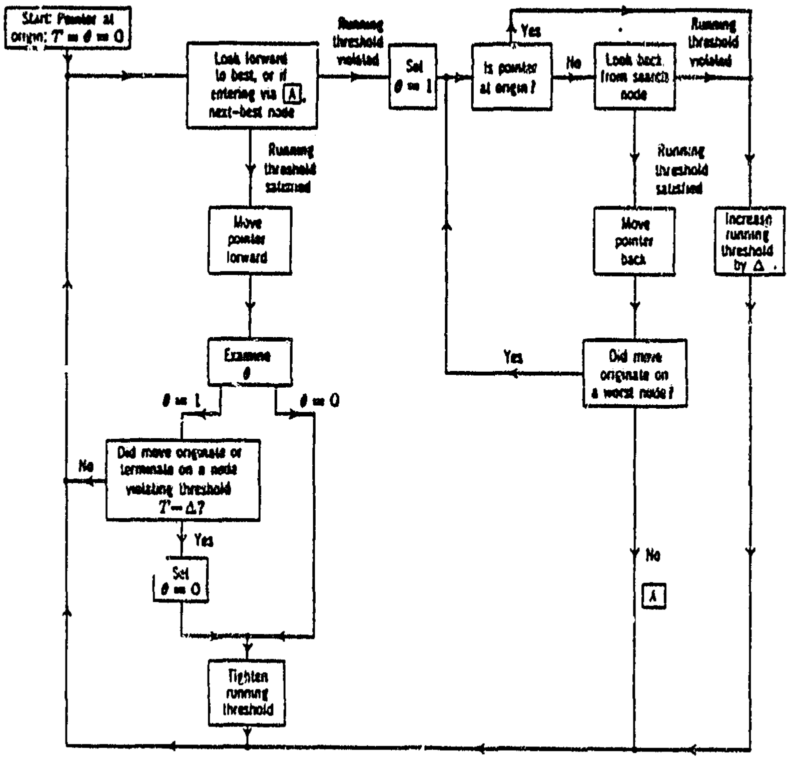
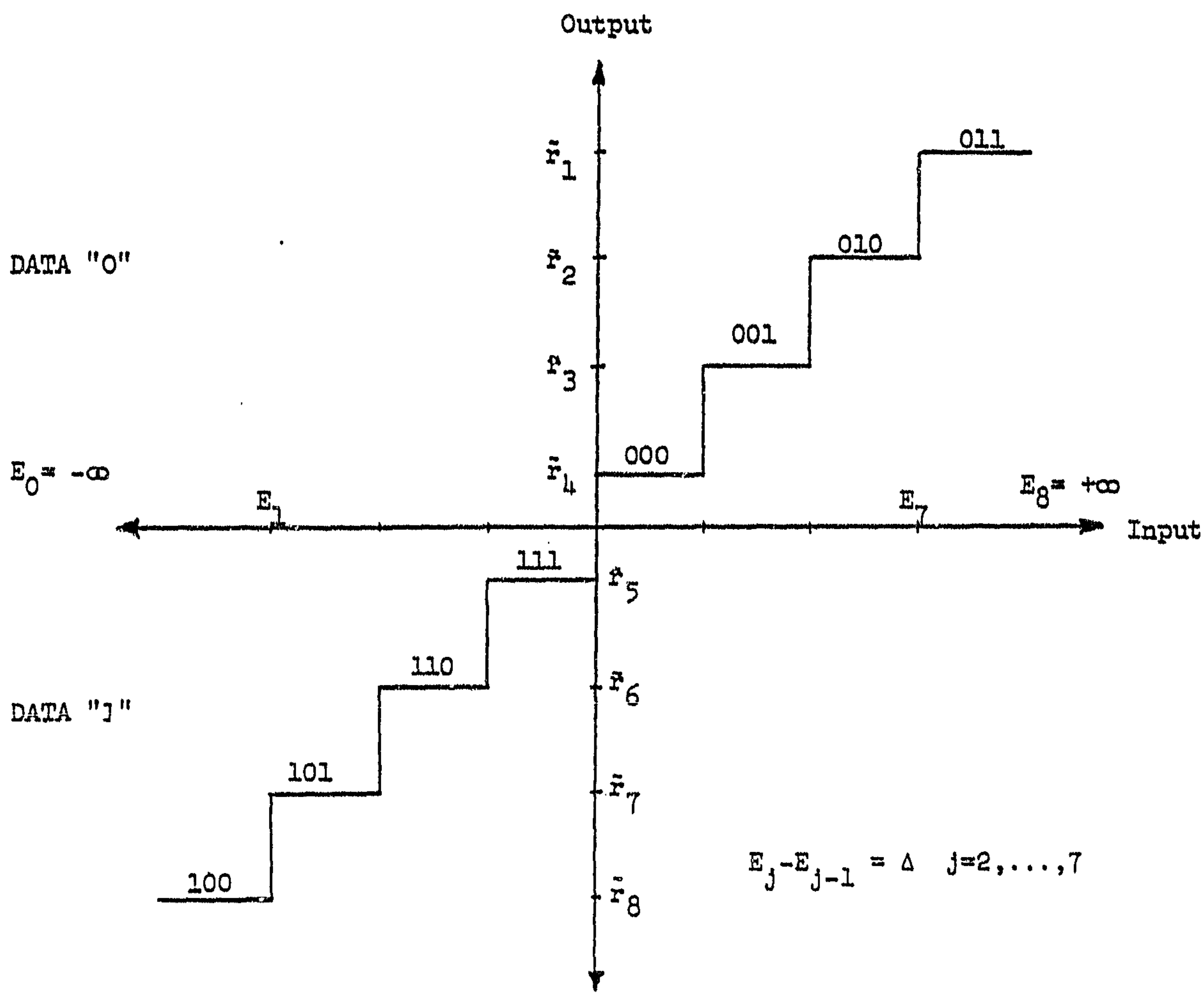


Figure 9

Flow diagram for the Fano Algorithm
employing a flag state for threshold tightening

ORIGINAL PAGE IS
OF POOR QUALITY



3-bit soft-decision code associated with each quantizer output level is shown.

Phase Reference Assignments:

	"DATA"	"DATA-BAR"
$\tilde{r}_1$	1	8
$\tilde{r}_2$	2	7
$\tilde{r}_3$	3	6
$\tilde{r}_4$	4	5
$\tilde{r}_5$	5	4
$\tilde{r}_6$	6	3
$\tilde{r}_7$	7	2
$\tilde{r}_8$	8	1

Figure 10

Uniform Quantizer Characteristic

SCALED BRANCH METRIC VALUES M00 =

	988	949	824	409	-589	-2173	-4046	-7040
	949	911	785	370	-628	-2212	-4085	-7079
0	824	785	659	244	-754	-2338	-4211	-7204
	409	370	244	-170	-1169	-2753	-4626	-7619
Symbol 1	-589	-628	-754	-1169	-2168	-3752	-5625	-8618
	-2173	-2212	-2338	-2753	-3752	-5336	-7209	-10202
1	-4046	-4085	-4211	-4626	-5625	-7209	-9082	-12075
	-7040	-7079	-7204	-7619	-8618	-10202	-12075	-15069

0 Symbol 2 1

SCALED BRANCH METRIC VALUES M01 =

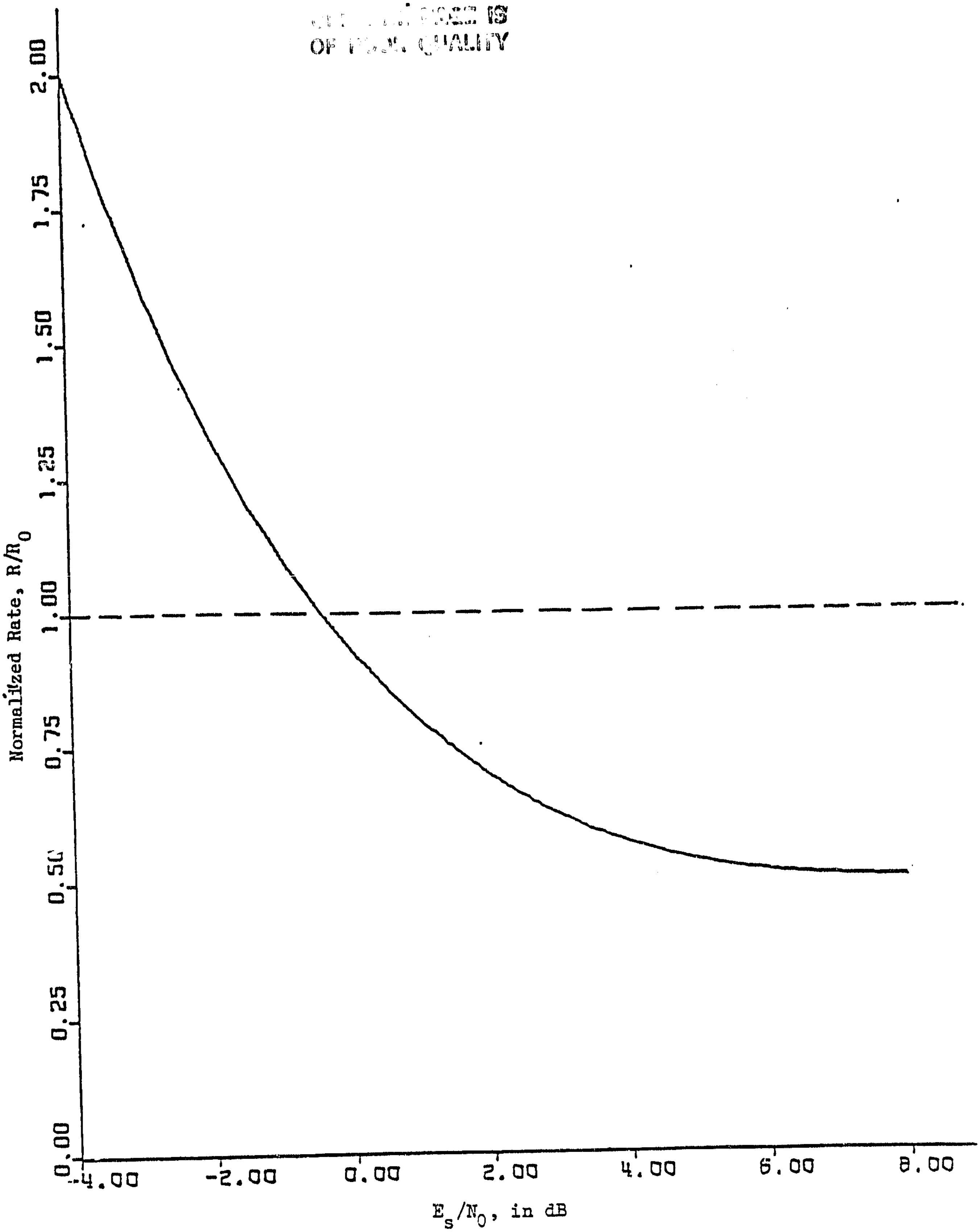
	-7040	-4046	-2173	-589	409	824	949	988
	-7079	-4085	-2212	-628	370	785	911	949
0	-7204	-4211	-2338	-754	244	659	785	824
	-7619	-4626	-2753	-1169	-170	244	370	409
Symbol 1	-8618	-5625	-3752	-2168	-1169	-754	-628	-589
	-10202	-7209	-5336	-3752	-2753	-2338	-2212	-2173
	-12075	-9082	-7209	-5625	-4626	-4211	-4085	-4046
1	-15069	-12075	-10202	-8618	-7619	-7204	-7079	-7040

0 Symbol 2 1

Figure 11

Branch Metric Array Table

$E_s/N_0 = 0.0$ dB



Sequential Decoding Parameters for Octal Channel

Figure 12

Appendix A

Main Driver Program Fortran Code

```
C
C THIS PROGRAM READS A DIGITAL TAPE WHICH CONTAINS THE
C SOFT-DECISIONS FROM THE DEEP SPACE NETWORK'S SYMBOL
C SYNCHRONIZER FOR THE DATA WHICH WILL BE RECORDED
C DURING THE OFF-LINE PROCESSING OF THE PIONEER-
C VENUS PRE-DETECTION RECORDINGS.  ALSO INCLUDED ON
C THE TAPE IS TIMING AND IDENTIFICATION INFORMATION.  THE
C PROGRAM FINDS THE 24 BIT SYNC WORD OF EACH TELEMETRY
C FRAME ON THE TAPE, THEN SEQUENTIALLY DECODES EACH FRAME,
C AND WRITES A DIGITAL TAPE WHICH WILL CONTAIN THE DECODED
C DATA ALONG WITH TIMING INFORMATION AND AN IDENTIFICATION
C RECORD.
C
C SPECIFY THE VARIABLES. DOUBLE PRECISION VARIABLES
C ARE USED FOR HOLDING CHARACTERS ONLY.
C
      DOUBLE PRECISION DSS14,DSS43,DSS,O1,O2,OUTOPT
C
C HALFWORD INTEGERS WILL BE USED FOR THE MOST PART
C IN PROCESSING THE DATA.
C
      COMMON IPHASE
      INTEGER*2 IASC,IND,ISPA,ILPA,ISEQ,IFR,IFR1,IFRB,
     *IWORK,IRETA,IQLA,IFLBR,ILMA,IIDA,IF.IWD,IEV,IYEAR,
     *IYEARS,IHALF,IDATA,IDAYH,IDAYT,IDAYU,IPBID,
     *IFILL,MOR2,MAND2,MLSL2,IHEX1
C
C DIMENSION ARRAYS USED BY PROGRAM.
C
      DIMENSION IFRAME(100),SFRAME(100,3),IFR(521),
     *IFR1(521),
     *IFRB(521,9),IWORK(1025,9),IRETA(1024,8),IQLA(512,9),
     *ESNO(8),IFLBR(608),IHALF(2),NSC(8),ACB(8),
     *IDFFLG(8),ITCT(8)
C
C EQUIVALENCE VARIABLES. THIS IS DONE FOR THE MOST PART
C TO FACILITATE LOGICAL OPERATIONS AND READING AND
C WRITING FROM TAPE.
C
      EQUIVALENCE (IVAL,IFR(3)),(IFULL,IHALF(1)),(IFLBR(7),
     *IVAL5)
C
C SET UP DATA CONSTANTS USED IN PROGRAM. FOR THE MOST PART
C THESE ARE ALL ALPHANUMERIC CONSTANTS.
      DATA IFWDB,IREVB,ITP,IPR,IDECA,IRAWA,IFANA,IQLAA,
     *IYES,INO,ISPA,ILPA,ILP1A,ISP1A,ISP2A,ISP3A,
     *DSS14,DSS43,ILMA,IIDA,IWD,IEV,IYEAR,IFILL,ITMA,
     *INMBA,IHEX1,I8A,I9A,I10A,I11A/'FWD ','REV ',
     *'TAPE','PRNT','DEC ','RAW ','FANO','QL  ',
     *'Y   ','N   ','SP','LP','LP  ','SP1 ','SP2 ',
     *'SP3 ','DSS 14  ','DSS 43  ',Z4C4D,Z4944,Z6744,
     *Z4556,'78',Z8888,'TIME','NUMB',ZFFFF,Z00003800,
     *Z00003900,Z00313000,Z00313100/
      DATA IDDD,IDBAR,O1,O2/'D   ','DB  ','TERM    ','LPRNT   '/
      DATA IRTF,IRDF/'TREC','SYNC'/
C
C FORMAT STATEMENTS USED BY PROGRAM
C
```

```
      10 FORMAT(1H0,'IS DATA RECORDED ON TAPE IN FORWARD ',
     *'OR REVERSE DIRECTION (FWD OR REV)?')
C
      11 FORMAT(1H0,'IS THE DATA TO BE DECODED OR IS RAW ',
     *'SYMBOL OUTPUT DESIRED (DEC OR RAW)?')
C
      12 FORMAT(1H0,'DO YOU WANT THE FANO DECODER USED OR',
     *' THE QUICK-LOOK OUTPUTS',
     */1X,'(FANO OR QL)?')
C
      13 FORMAT(1H0,'DO YOU WANT TAPE OUTPUT OR PRINT ',
     *'OUTPUT (TAPE OR PRNT)?')
C
      14 FORMAT(1H0,'SPECIFY BY NUMBER THE FRAMES WHICH ARE TO ',
     *'BE PROCESSED. ',
     */1X,'THE LAST DIGIT OF EACH NUMBER MUST END ',
     *'UP IN COLUMN 5.',
     */1X,'INDICATE YOU ARE DONE BY HITTING A RETURN.')
C
      15 FORMAT(1H0,'DO YOU WANT ALL THE FRAMES PRINTED ',
     *'OUT (Y OR N)?')
C
      16 FORMAT(1H0,'DO YOU WANT TO SPECIFY THE FRAMES TO ',
     *'BE PROCESSED ACCORDING TO ',
     */1X,'TIME TAG OR FRAME NUMBER (TIME OR NUMB)?')
C
      17 FORMAT(1H0,'ENTER DSS NUMBER (14 OR 43)-')
C
      18 FORMAT(1H0,'ENTER PROBE ID (SP1,SP2,SP3 OR LP)-')
C
      19 FORMAT(1H0,'ENTER TAPE SEQUENCE NUMBER-')
C
      20 FORMAT(A4)
C
      21 FORMAT(A2)
C
      22 FORMAT(I5)
C
      23 FORMAT(1H0,'HOURS-   ')
C
      24 FORMAT(1H0,'MINUTES- ')
C
      25 FORMAT(1H0,'SECONDS- ')
C
      26 FORMAT(F12.5)
C
      27 FORMAT(A2,I1)
C
      28 FORMAT(4(128A2),9A2)
C
      29 FORMAT(1H1,5X,'PIONEER VENUS REVERSE PLAYBACK TELEMETRY ',
     *'PROGRAM',/)
C
      30 FORMAT(1H0,5X,'PROBE IDENTIFICATION-  ',A4,/)
C
      31 FORMAT(1H0,5X,'DSS NUMBER-  ',A8,/)
C
      32 FORMAT(1H0,5X,'DATA START TIME-',/10X,'HOURS-   ',I2,
     */10X,'MINUTES- ',I2,/10X,'SECONDS- ',F6.3//)
C
```

```
      33 FORMAT(1H0,'ENTER YEAR DATA WAS RECORDED (E.G. 78)-')
C
      34 FORMAT(1H0,/20X,'PROGRAM SUMMARY',//6X,
        *'TOTAL NUMBER ',
        *'OF FRAMES PROCESSED= ',I5,/6X,'TOTAL ',
        *'NUMBER OF FRAMES ',
        *'DELETED=    ',I5,/6X,'DELETION RATE=     ',F10.6,
        */6X,'SYMBOL ERROR RATE=',F10.6)
C
      35 FORMAT(1H0,/22X,'PROGRAM SUMMARY',//6X,'TOTAL NUMBER ',
        *'OF FRAMES PROCESSED= ',I5)
C
      36 FORMAT(1H0,'ARE YOU DONE (Y OR N)?')
C
      37 FORMAT(4(128A2),96A2)
C
      38 FORMAT(1H0,'INPUT MAXIMUM NUMBER OF COMPUTATIONS',
        *' PER FRAME. ',
        */1X,'THE LAST DIGIT OF THE NUMBER MUST',
        *' END UP IN COLUMN 8.',
        */1X,'0 INDICATES DEFAULT=100,000 COMPUTATIONS.')
C
      39 FORMAT(I8)
C
      40 FORMAT(I2)
C
      41 FORMAT(1H0,'ERROR OCCURRED WHEN TAPE WAS READ.',
        */1X,'TAPE IS INCORRECTLY FORMATTED OR TAPE DATASET',
        */1X,'WAS INCORRECTLY SPECIFIED.')
C
      42 FORMAT(1H0,'RANDOM ACCESS FILE COULD NOT BE OPENED.',
        */1X,'THE PROBLEM IS MOST LIKELY DUE TO INSUFFICIENT TEMPORARY',
        */1X,'STORAGE ALLOCATION OR INCORRECT DATASET DEFINITION.')
C
      43 FORMAT(1H0,'WHICH PHASE REFERENCE DO YOU WANT-   ',
        *'DATA OR DATA-BAR (D OR DB)?')
C
      44 FORMAT(1H0,'DO YOU WANT PRINTOUT TO THE TERMINAL OR TO THE '
        *,'LINE PRINTER ',
        */1X,'(TERM OR LPRNT)?')
C
      45 FORMAT(1H0,'SPECIFY THE TIME TAG OF EACH FRAME WHICH',
        *' IS TO BE PRINTED OUT.',
        */1X,'INDICATE YOU ARE FINISHED BY ENTERING A -1.0',
        */1X,'(INCLUDE DECIMAL POINT WHEN ENTERING NUMBERS).')
C
      46 FORMAT(1H0,I4,' RECORDS HAVE BEEN READ FROM INPUT TAPE')
C
      47 FORMAT(A8)
C
      48 FORMAT(1H0,'DO YOU WANT RAW SYMBOLS TO BE SYNCHRONIZED (1024) ',
        *'IN A DATA FRAME OR ',
        */1X,'THOSE (1025) CORRESPONDING TO A TAPE RECORD (SYNC OR TREC)?'
C
      49 FORMAT(1H0,5X,'FRAMES DECODED USING QUICK-LOOK ALGORITHM.')
C
      51 FORMAT(1H0,5X,'FRAMES DECODED USING FANO ALGORITHM.')
C
      52 FORMAT(1H0,5X,'SYNC WORD LOCATION=',I5)
C
      53 FORMAT(1H0,5X,'SIGNAL-TO-NOISE RATIO (DB)=',G17.10)
C
```

```
   54 FORMAT(1H0,5X,'PLAYBACK DAY OF YEAR- ',I3,/)
C
   55 FORMAT(1H0,5X,'FRAME NO.',I5,2X,'SYNC ',
     *'WORD LOCATION=',I5)
C
C SET DEVICE NUMBERS FOR FORTRAN I/O.
C IIN:INPUT FROM TERMINAL.
C IMES:OUTPUT TO TERMINAL.
C IPRINT:PRINT OUTPUT.
C ITAPEI:INPUT FROM TAPE.
C ITAPEO:OUTPUT TO TAPE.
C IDISK:INPUT AND OUTPUT TO DIRECT ACCESS DEVICE.
C
      IPASS=0
   50 IIN=1
      IMES=2
      IPRINT=3
      ITAPEI=4
      ITAPEO=5
      IDISK=6
C
C FIND OUT WHAT OPTIONS USER WANTS AND SET THE
C APPROPRIATE FLAGS.
C
C FIND OUT WHAT DIRECTION DIGITAL DATA ON TAPE IS STORED IN.
C
      IF(IPASS .GT. 0)  GO TO 200
  100 WRITE(IMES,10)
      READ(IIN,20) IDIR
      IF(IDIR.NE.IFWDB.AND.IDIR.NE.IREVB) GO TO 100
C
C FIND OUT WHETHER PRINT OR TAPE OUTPUT IS DESIRED.
C
  200 WRITE(IMES,13)
      READ(IIN,20) IOUT
      IF(IOUT.NE.ITP.AND.IOUT.NE.IPR) GO TO 200
      IOPT=IDECA
  220 WRITE(IMES,44)
      READ(IIN,47) OUTOPT
      IF(OUTOPT.NE.01 .AND. OUTOPT.NE.02) GO TO 220
      IF(OUTOPT.EQ.01) IPRINT=IMES
C
C FIND OUT ABOUT PHASE.
C
  250 WRITE(IMES,43)
      READ(IIN,20) IPHASE
      IF(IPHASE.NE.IDDD.AND.IPHASE.NE.IDBAR) GO TO 250
      IF(IOUT.EQ.ITP) GO TO 400
C
C SINCE PRINTOUT WAS CHOSEN, FIND OUT IF USER
C WANTS DATA DECODED OR JUST WANTS TO SEE RAW SYMBOLS.
C
  300 WRITE(IMES,11)
      READ(IIN,20) IOPT
      IF(IOPT.NE.IDECA.AND.IOPT.NE.IRAWA) GO TO 300
      IF(IOPT.EQ.IRAWA) GO TO 350
      GO TO 400
  350 WRITE(IMES,48)
      READ(IIN,20) IRWD
      IF(IRWD.NE.IRTF .AND. IRWD.NE.IRDF) GO TO 350
      GO TO 600
C
C THE DATA IS TO BE DECODED. FIND OUT WHETHER
C TO USE FANO OR QUICK-LOOK DECODER.
C
```

```
  400 WRITE(IMES,12)
      READ(IIN,20) IDEC
      IF(IDEC.NE.IFANA.AND.IDEC.NE.IQLAA) GO TO 400
      IF(IDEC.EQ.IQLAA) GO TO 500
  450 WRITE(IMES,38)
      READ(IIN,39,ERR=450) ICOMPT
      IF(ICOMPT .EQ. 0) GO TO 460
      IF(ICOMPT.LT.512 .OR. ICOMPT.GT.10000000) GO TO 450
      GO TO 500
  460 ICOMPT=100000
C
C SEE IF ALL FRAMES ARE TO BE PROCESSED, OR IF ONLY A
C SELECTED NUMBER OF FRAMES ARE DESIRED.
C
  500 CONTINUE
  600 WRITE(IMES,15)
      READ(IIN,20) IANS
      IF(IANS.NE.IYES.AND.IANS.NE.INO) GO TO 600
C
C IF ALL THE FRAMES ARE TO BE PRINTED OUT,
C SET AN INDICATOR AND MOVE ON.
C
      IFRAME(1)=-1
      IF(IANS.EQ.IYES) GO TO 1650
C
C ONLY SELECTED FRAMES ARE TO BE PRINTED OUT.
C FIND OUT IF USER WANTS TO SELECT FRAME BY
C TIME TAG OR FRAME NUMBER.
C
  700 WRITE(IMES,16)
      READ(IIN,20) ITYPE
      IF(ITYPE.NE.ITMA.AND.ITYPE.NE.INMBA) GO TO 700
      IF(ITYPE.EQ.ITMA) GO TO 1100
C
C USER WANTS TO SELECT FRAMES BY NUMBER.
C READ IN THE DESIRED FRAME NUMBERS.
C
      WRITE(IMES,14)
      DO 900 I=1,99
  800 READ(IIN,22,ERR=800) IFRAME(I)
      IF(IFRAME(I).EQ.0) IFRAME(I)=-1
      IF(IFRAME(I).EQ.-1) GO TO 1000
  900 CONTINUE
C
C MAKE SURE THERE'S A -1 INDICATING LAST FRAME.
C THEN MOVE ON.
C
 1000 IFRAME(100)=-1
      GO TO 1650
C
C USER WANTS TO SPECIFY FRAME BY TIME. ENTER THE TIMES.
C
 1100 WRITE(IMES,45)
      DO 1500 I=1,99
C
C GET HOURS.
C
 1200 WRITE(IMES,23)
      READ(IIN,26,ERR=1200) SFRAME(I,1)
      IF(SFRAME(I,1).EQ.-1.0) GO TO 1600
      IF(SFRAME(I,1).LT.0.0.OR.SFRAME(I,1).GT.24.0)GOTO 1200
```

```
C
C GET MINUTES.
C
 1300 WRITE(IMES,24)
      READ(IIN,26,ERR=1300) SFRAME(I,2)
      IF(SFRAME(I,2).EQ.-1.0) GO TO 1600
      IF(SFRAME(I,2).LT.0.0.OR.SFRAME(I,2).GT.60.0)GOTO 1300
C
C GET SECONDS.
C
 1400 WRITE(IMES,25)
      READ(IIN,26,ERR=1400) SFRAME(I,3)
      IF(SFRAME(I,3).EQ.-1.0) GO TO 1600
      IF(SFRAME(I,3).LT.0.0.OR.SFRAME(I,3).GT.60.0)GOTO 1400
 1500 CONTINUE
C
C MAKE SURE THERE'S A -1 INDICATING LAST FRAME.
C THEN MOVE ON.
C
 1600 SFRAME(100,1)=-1.0
C
C GET DSS NUMBER FROM USER.
C
 1650 IF(IPASS .GT. 0)  GO TO 1900
 1700 WRITE(IMES,17)
      READ(IIN,40,ERR=1700) IDSS
      IF(IDSS.NE.14.AND.IDSS.NE.43) GO TO 1700
C
C GET PROBE IDENTIFICATION.
C
 1800 WRITE(IMES,18)
      READ(IIN,27,ERR=1800) IASC,IND
      IF(IASC.NE.ISPA.AND.IASC.NE.ILPA) GO TO 1800
      IPBID=11
      IF(IASC.EQ.ILPA) GO TO 1900
      IF(IND.LT.1.OR.IND.GT.3) GO TO 1800
      IPBID=7+IND
C
C GET TAPE SEQUENCE NUMBER.
C
 1900 IF(IOUT.EQ.IPR) GO TO 1950
      WRITE(IMES,19)
C
      READ(IIN,21) ISEQ
C
C GET YEAR DATA WAS RECORDED.
C
      WRITE(IMES,33)
      READ(IIN,21) IYEAR
C
C ALL THE OPTIONS HAVE BEEN SPECIFIED AND PARAMETERS ENTERED.
C START PROCESSING THE INPUT TAPE.
C
C FIRST READ THE CONTENTS OF THE TAPE ONTO A DIRECT ACCESS DEVICE.
C
 1950 CONTINUE
      IF(IPASS.GT.0) GO TO 2400
      CALL DEFINE(6,5000,1042,IV)
      IF(IV.NE.1) GO TO 5200
      DO 2000 I=1,5000
      READ(ITAPEI,28,END=2100) IFR
      IV=I
      WRITE(IDISK) IFR
```

```
 2000 CONTINUE
C
C SET NUMBER OF RECORDS TO MAX. SHOULD NEVER GET HERE.
C
      NREC=5000
      GO TO 2200
C
C SET THE NUMBER OF RECORDS.
C
 2100 NREC=I-1
C
C SEE IF THE DATA IS RECORDED FORWARDS OR BACKWARDS.
C
 2200 WRITE(IMES,46) NREC
      IF(IDIR.EQ.IFWDB) GO TO 2400
C
C DATA ON TAPE IS IN REVERSE TIME ORDER. REVERSE THE TAPE
C RECORDS AND SYMBOLS IN EACH RECORD.
C
      NREC2=(NREC+1)/2
      DO 2300 I=1,NREC2
      NVAL=NREC-I+1
      IV=I
      READ(IDISK) IFR
      IV=NVAL
      READ(IDISK) IFR1
      CALL REVSIM(IFR)
      CALL REVSIM(IFR1)
      IV=I
      WRITE(IDISK) IFR1
      IV=NVAL
      WRITE(IDISK) IFR
 2300 CONTINUE
C
C PRINT THE HEADER FOR THE OUTPUT LISTING.
C
 2400 WRITE(IPRINT,29)
      IGDIP=IPBID-7
      GO TO (2500,2600,2700,2800),IGDIP
 2500 WRITE(IPRINT,30) ISP1A
      IVAL6=I8A
      GO TO 2900
 2600 WRITE(IPRINT,30) ISP2A
      IVAL6=I9A
      GO TO 2900
 2700 WRITE(IPRINT,30) ISP3A
      IVAL6=I10A
      GO TO 2900
 2800 WRITE(IPRINT,30) ILP1A
      IVAL6=I11A
C
C PRINTOUT DSS NUMBER.
C
 2900 DSS=DSS14
      IF(IDSS.EQ.43) DSS=DSS43
      WRITE(IPRINT,31) DSS
C
C PRINTOUT DATA START TIME. THE DATA START TIME
C WILL BE CONTAINED IN THE FIRST DATA RECORD. SO
C READ IN THE FIRST RECORD AND GET THE DATA.
C
```

```
      IV=1
      READ(IDISK) IFR
C
C GET PLAYBACK DAY OF YEAR.
C
      ISDAY=IFR(3)/2
      ISEC=MAND(IVAL,131071)
      IHOUR=ISEC/3600
      ISEC=ISEC-(IHOUR*3600)
      IMIN=ISEC/60
      ISEC=ISEC-(IMIN*60)
      IMIL=IFR(5)
      SECOND=FLOAT(ISEC)+.001*FLOAT(IMIL)
      WRITE(IPRINT,54) ISDAY
      WRITE(IPRINT,32) IHOUR,IMIN,SECOND
C
C SEE IF PRINT OUTPUT IS REQUESTED. IF IT ISN'T, GO
C TO TAPE PROCESSING PORTION OF PROGRAM IF ALL FRAMES
C ARE TO BE PROCESSED.
C
      II=1
      NFRAMS=0
      NDELT=0
      NSYMBT=0
      NSCT=0
      IF(IOUT.EQ.ITP .AND. IANS.EQ.IYES) GO TO 3700
C
C FIND OUT IF ALL FRAMES ARE TO BE PROCESSED.
C
      IF(IANS.EQ.IYES) GO TO 3950
C
C ONLY SELECTED FRAMES ARE TO BE PROCESSED.  PROCESS EACH FRAME.
C
C IF TAPE OUTPUT OF SELECTED FRAMES IS DESIRED, FIRST WRITE
C OUT FILE HEADER.
C
      IF(IOUT.EQ.ITP) GO TO 3700
 2950 K=1
      IF(IOUT.EQ.ITP .AND. IDEC.EQ.IQLAA) WRITE(IPRINT,49)
      IF(IOUT.EQ.ITP .AND. IDEC.NE.IQLAA) WRITE(IPRINT,51)
C
C FIRST SEE HOW FRAME WAS SELECTED.
C
 3000 IF(ITYPE.EQ.INMBA) GO TO 3100
C
C USER WANTS TO FIND FRAME BY TIME REFERENCE.
C
      IF(SFRAME(II,1).EQ.-1.0.OR.SFRAME(II,2).EQ.-1.0.OR.
     *SFRAME(II,3).EQ.-1.0) GO TO 3600
C
C CALL A ROUTINE TO FIND THE FRAME NUMBER CORRESPONDING TO THE
C TIME TAG SPECIFIED.
C
      CALL FRFIND(II,SFRAME,IDISK,IFR,NREC,NFR,IV)
      IF(NFR.LT.1) GO TO 3400
      NFRAMS=NFRAMS+1
      GO TO 3200
C
C USER SELECTED FRAME BY NUMBER.
C
```

```
 3100 NFR=IFRAME(II)
      IF(NFR.EQ.-1) GO TO 3600
      IF(NFR.LT.1.OR.NFR.GT.NREC) GO TO 3400
      NFRAMS=NFRAMS+1
C
C NOW HAVE THE FRAME NUMBER. SEE IF RAW SYMBOL OUTPUT IS DESIRED.
C
 3200 IF(IOPT.EQ.IRAWA) GO TO 3500
C
C USER WANTS DATA DECODED. FIRST FIND THE SYNC WORD AND GET THE FRAME.
C
      CALL SYNC(NFR,NREC,IDISK,IFRB,IWORK,IRETA,IQLA,
     *1,ESNO,IFFRT,ISYCW,IV)
      IF(IFFRT.GT.0) GO TO 3250
      NDELT=NDELT+1
      GO TO 3400
C
C SEE WHAT KIND OF DECODER USER WANTS.
C
 3250 IDFFLG(1)=-1
      IF(IOUT.EQ.ITP) WRITE(IPRINT,55) NFR,ISYCW
      IF(IOUT.EQ.ITP) WRITE(IPRINT,53) ESNO(1)
      IF(IDEC.EQ.IQLAA) GO TO 3300
C
C USER WANTS FANO ALGORITHM USED, SO CALL ROUTINE TO DO IT.
C
      ITCT(1)=ICOMPT
      CALL FANO(ESNO(1),IRETA,IQLA,ITCT(1),ACB(1),NSC(1),IDFFLG(1),IOUT
      NDELT=NDELT+1
      IF(IDFFLG(1).NE.0) GO TO 3300
      NDELT=NDELT-1
      NSYMBT=NSYMBT+1024
      NSCT=NSCT+NSC(1)
C
C DATA IS DECODED PRINT IT OUT.
C
 3300 IF(IOUT.EQ.ITP) GO TO 6000
      CALL FRPRNT(NFR,IQLA,ACB,NSC,IDFFLG,IDISK,IFR,
     *IDEC,1,IPRINT,ISYCW,ESNO,IV)
C
C GO ON AND GET NEXT SELECTED FRAME.
C
 3400 II=II+1
      GO TO 3000
C
C RAW SYMBOL OUTPUT REQUESTED. CALL ROUTINE TO DO IT.
C
C FIRST GET S/N RATIO AND SYNCHRONIZE THE (MAPPED) SYMBOLS
C
 3500 CALL SYNC(NFR,NREC,IDISK,IFRB,IWORK,IRETA,IQLA,
     *1,ESNO,IFFRT,ISYCW,IV)
C
C TRANSFER (MAPPED) SYMBOLS AND S/N RATIO TO IWORK ARRAY.
C FIRST UNCOMPLEMENT THE SECOND SYMBOL OF EACH CHANNEL PAIR.
C
      DO 3525  I=1,1023,2
      J=I+1
      IWORK(I,1)=IRETA(I,1)
      IWORK(J,1)=9-IRETA(J,1)
 3525 CONTINUE
      IWORK(1025,1)=ISYCW
```

```
      CALL RSPRNT(NFR,IDISK,IFR,IPRINT,IWORK,NREC,ESNO,
     *IRWD,IPHASE,IV)
      GO TO 3400
C
C ALL DECODING DONE. PRINT SUMMARY BLOCK.
C
 3600 IF(NFRAMS .EQ. 0) GO TO 3650
      II=II-1
      IF(IOUT.EQ.ITP) GO TO 6002
 3650 GO TO 4700
C
C TAPE OUTPUT IS REQUESTED. GENERATE AND WRITE OUT FILE HEADER.
C
 3700 NFRAMS=0
      NDELT=0
      NSYMBT=0
      NSCT=0
      IFLBR(1)=0
      IFLBR(2)=84
      IFLBR(3)=ILMA
      IFLBR(4)=IIDA
      IFLBR(5)=20992
      IFLBR(6)=25
      IF(IDSS.EQ.43) IFLBR(6)=134
      IVAL5=IVAL6
      IFLBR(9)=0
      IFLBR(10)=MLSR2(ISEQ,8)
      IFLBR(10)=IFLBR(10)-192
      IFULL=ISEQ
      IFLBR(11)=(MAND(IFULL,255)-192)*256
      IF=70
      IF(IDIR.EQ.IREVB) IF=82
      IFLBR(11)=IFLBR(11)+IF
      IFLBR(12)=IWD
      IF(IDIR.EQ.IREVB) IFLBR(12)=IEV
      IFLBR(13)=(MLSR2(IYEAR,8)-192)+48*256
      IFULL=IYEAR
      IFLBR(14)=(MAND(IFULL,255)-192)*256
      DO 3800 III=15,23
      IFLBR(III)=0
 3800 CONTINUE
C
C GET DATA START TIME,DAY OF YEAR.
C
      IV=1
      READ(IDISK) IFR
      ISEC=MAND(IVAL,131071)
      IYEARS=IFR(3)/2
      IFLBR(24)=IYEARS
      IMIL=IFR(5)
      IHSEC=IMIL/10+ISEC*100
C
C GET DATA STOP TIME.
C
      IV=NREC
      READ(IDISK) IFR
      IYEARS=IFR(3)/2
      IFLBR(25)=IYEARS/256
      IFLBR(26)=MLSL2(IYEARS,8)
      IFULL=IHSEC
      IFLBR(26)=IHALF(1)+IFLBR(26)
      IFLBR(27)=IHALF(2)
      ISEC=MAND(IVAL,131071)
```

```
      IMIL=IFR(5)
      IHSEC=IMIL/10+ISEC*100
      IFULL=IHSEC
      IFULL=MLSL(IFULL,8)
      IFLBR(28)=IHALF(1)
      IFLBR(29)=IHALF(2)
      DO 3900 III=30,608
      IFLBR(III)=0
 3900 CONTINUE
C
C WRITE OUT THE FILE LABEL RECORD.
C
      WRITE(ITAPEO,37) IFLBR
C
C SET UP THE COMMON PART OF THE DATA BLOCK.
C
      IFLBR(1)=1
      IFLBR(2)=1216
      IFLBR(3)=150
      IFLBR(4)=150
      IFLBR(5)=8
      IFLBR(6)=0
      IFLBR(7)=0
      IFLBR(8)=0
C
C IF TAPE OUTPUT OF SELECTED FRAMES IS DESIRED RETURN
C TO DECODE THOSE FRAMES.
C
      IF(IANS.EQ.INO) GO TO 2950
C
C IF ALL FRAMES ARE DESIRED START DECODING,
C EIGHT FRAMES AT A TIME.
C
 3950 IF(IOPT.EQ.IRAWA) GO TO 5300
      J=1
 4000 CALL SYNC(J,NREC,IDISK,IFRB,IWORK,IRETA,IQLA,
     *8,ESNO,IFFRT,ISYCW,IV)
      IF(IFFRT.LE.0) GO TO 4400
      NFRAMS=NFRAMS+IFFRT
C
C IF QUICK-LOOK CODE IS BEING USED, DON'T CALL FANO DECODER.
C
      IF(IDEC .NE. IQLAA)  GO TO 4030
      IF(IOUT.EQ.ITP) WRITE(IPRINT,49)
      IF(IOUT.EQ.ITP) WRITE(IPRINT,52) ISYCW
      DO 4020  I=1,IFFRT
      IF(IOUT.EQ.ITP) WRITE(IPRINT,53) ESNO(I)
 4020 CONTINUE
 4030 IF(IDEC.EQ.IQLAA) GO TO 4150
C
C CALL FANO DECODER TO DECODE EACH FRAME.
C
      IF(IOUT.EQ.ITP) WRITE(IPRINT,51)
      IF(IOUT.EQ.ITP) WRITE(IPRINT,52) ISYCW
      DO 4100 I=1,IFFRT
      IF(IOUT.EQ.ITP) WRITE (IPRINT,53) ESNO(I)
      ITCT(I)=ICOMPT
      NDELT=NDELT+1
      CALL FANO(ESNO(I),IRETA(1,I),IQLA(1,I),ITCT(I),ACB(I),
     *NSC(I),IDFFLG(I),IOUT)
      IF(IDFFLG(I).NE.0)  GO TO 4100
      NDELT=NDELT-1
```

```
      NSYMBT=NSYMBT+1024
      NSCT=NSCT+NSC(I)
 4100 CONTINUE
C
C SEE IF TAPE OUTPUT IS DESIRED.
C
 4150 IF(IOUT.EQ.ITP) GO TO 4175
C
C PRINTOUT IS DESIRED.  CALL ROUTINE TO DO THE PRINTING.
C
      CALL FRPRNT(J,IQLA,ACB,NSC,IDFFLG,IDISK,IFR,
     *IDEC,IFFRT,IPRINT,ISYCW,ESNO,IV)
      GO TO 4400
C
C TAPE OUTPUT REQUESTED. WRITE OUT THE DATA BLOCKS.
C
 4175 DO 4200 I=1,IFFRT
      INDEX=(I-1)*75
      IARG=9+INDEX
      IFLBR(IARG)=25206
      IARG=10+INDEX
      IFLBR(IARG)=25
      IF(IDSS.EQ.43) IFLBR(IARG)=134
      IVALLL=IFLBR(IARG)
      IFLBR(IARG)=MOR(IVALLL,9984)
      IFULL=9728+IPBID
      IF(IPBID .LT. 11) IFULL=8960+IPBID
      IFULL=IFULL+10665984
      IFULL=MLSL(IFULL,8)
      IARG=12+INDEX
      IFLBR(IARG)=IHALF(1)
      IARG=13+INDEX
      IFLBR(IARG)=IHALF(2)
      IARG=11+INDEX
      IFLBR(IARG)=MOR(MLSL(170,8),222)
      IV=J+I
      READ(IDISK) IFR
      ISEC=MAND(IVAL,131071)
      IMIL=IFR(5)
      IHSEC=IMIL/10+ISEC*100
      IFULL=IHSEC
      IARG=13+INDEX
      IFLBR(IARG)=IFLBR(IARG)+IHALF(1)
      IARG=14+INDEX
      IFLBR(IARG)=IHALF(2)
      IYEARS=IFR(3)/2
      IDAYH=IYEARS/100
      IDAYT=(IYEARS-IDAYH*100)/10
      IDAYU=IYEARS-IDAYH*100-IDAYT*10
      IARG=15+INDEX
      IFLBR(IARG)=IDAYH*4096+IDAYT*256+IDAYU*16
      IARG=16+INDEX
      IFLBR(IARG)=IMIL/100+I*256
      IDAYH=128
      IF(IDFFLG(I).NE.0) IDAYH=130
      IARG=17+INDEX
      IFLBR(IARG)=MLSL2(IDAYH,8)
      IARG=18+INDEX
      IFLBR(IARG)=0
      IARG=19+INDEX
      IFLBR(IARG)=IFR(8)
      IARG=20+INDEX
      IFLBR(IARG)=IFR(7)
```

```
      IARG=21+INDEX
      CALL DPACK(IQLA(1,I),IFLBR(IARG))
      IARG=53+INDEX
      IFLBR(IARG)=NSC(I)
      IARG=54+INDEX
      IFLBR(IARG)=ITCT(I)/64
      DO 4200 III=55,83
      IARG=III+INDEX
      IFLBR(IARG)=0
 4200 CONTINUE
C
C SEE IF FILLER BLOCKS HAVE TO BE INSERTED.
C
      IF(IFFRT.EQ.8) GO TO 4350
C
C WRITE UP FILLER BLOCKS.
C
      IFRT1=IFFRT+1
      DO 4300 III=IFRT1,8
      INDEX=(III-1)*75
      IARG=9+INDEX
      IFLBR(IARG)=25206
      IARG=10+INDEX
      IFLBR(IARG)=25
      IF(IDSS .EQ. 43) IFLBR(IARG)=134
      IVALLL=IFLBR(IARG)
      IFLBR(IARG)=MOR(IVALLL,9984)
      IARG=11+INDEX
      IFLBR(IARG)=MOR(MLSL(170,8),190)
      DO 4300 IIII=1,72
      IARG=11+INDEX+IIII
      IFLBR(IARG)=IFILL
 4300 CONTINUE
 4350 WRITE(ITAPEO,37) IFLBR
      IFLBR(1)=IFLBR(1)+1
C
C PROCESS NEXT GROUP OF FRAMES.
C
 4400 J=J+8
      IF(J.GT.NREC) GO TO 4500
      GO TO 4000
C
C DATA DECODED. NOW WRITE TAPE ENDING IF TAPE OUTPUT REQUESTED.
C
 4500 IF(IOUT.NE.ITP) GO TO 4700
      IFLBR(1)=IHEX1
      DO 4600 I=2,608
      IFLBR(I)=0
 4600 CONTINUE
      WRITE(ITAPEO,37) IFLBR
      END FILE ITAPEO
      END FILE ITAPEO
C
C WRITE SUMMARY BLOCK FOR THIS PASS THROUGH THE PROGRAM.
C
 4700 IF(IOPT.EQ.IDECA.AND.IDEC.EQ.IFANA) GO TO 4800
      WRITE(IPRINT,35) NFRAMS
      GO TO 4900
 4800 DELRT=0.0
      SERT=0.0
      IF(NFRAMS.NE.0) DELRT=FLOAT(NDELT)/FLOAT(NFRAMS)
      IF(NSYMBT.NE.0) SERT=FLOAT(NSCT)/FLOAT(NSYMBT)
```

```
      WRITE(IPRINT,34) NFRAMS,NDELT,DELRT,SERT
 4900 WRITE(IMES,36)
      READ(IIN,20)IANS
      IF(IANS.NE.IYES.AND.IANS.NE.INO) GO TO 4900
      IF(IANS.EQ.IYES) GO TO 5000
      IPASS=IPASS+1
      GO TO 50
 5000 STOP
C
C ERROR IN ACCESSING TEMPORARY FILE.
C
 5200 WRITE(IMES,42)
      STOP
C
 5300 NFRAMS=0
      DO 5400 I=1,NREC
      NFRAMS=NFRAMS+1
      CALL RSPRNT(I,IDISK,IFR,IPRINT,IWORK,NREC,ESNO,
     *IRWD,IPHASE,IV)
 5400 CONTINUE
      GO TO 4700
C
C OUTPUT TO TAPE THE FRAMES SELECTED AFTER PROCESSING.
C
 6000 INDEX=(K-1)*75
      IARG=9+INDEX
      IFLBR(IARG)=25206
      IARG=10+INDEX
      IFLBR(IARG)=25
      IF(IDSS .EQ. 43) IFLBR(IARG)=134
      IVALLL=IFLBR(IARG)
      IFLBR(IARG)=MOR(IVALLL,9984)
      IFULL=9728+IPBID
      IF(IPBID .LT. 11) IFULL=8960+IPBID
      IFULL=IFULL+10665984
      IFULL=MLSL(IFULL,8)
      IARG=12+INDEX
      IFLBR(IARG)=IHALF(1)
      IARG=13+INDEX
      IFLBR(IARG)=IHALF(2)
      IARG=11+INDEX
      IFLBR(IARG)=MOR(MLSL(170,8),222)
      IV=NFR+1
      READ(IDISK) IFR
      ISEC=MAND(IVAL,131071)
      IMIL=IFR(5)
      IHSEC=IMIL/10+ISEC*100
      IFULL=IHSEC
      IARG=13+INDEX
      IFLBR(IARG)=IFLBR(IARG)+IHALF(1)
      IARG=14+INDEX
      IFLBR(IARG)=IHALF(2)
      IYEARS=IFR(3)/2
      IDAYH=IYEARS/100
      IDAYT=(IYEARS-IDAYH*100)/10
      IDAYU=IYEARS-IDAYH*100-IDAYT*10
      IARG=15+INDEX
      IFLBR(IARG)=IDAYH*4096+IDAYT*256+IDAYU*16
      IARG=16+INDEX
      IFLBR(IARG)=IMIL/100+K*256
      IDAYH=128
      IF(IDFFLG(1).NE.0) IDAYH=130
      IARG=17+INDEX
```

```
      IFLBR(IARG)=MLSL2(IDAYH,8)
      IARG=18+INDEX
      IFLBR(IARG)=0
      IARG=19+INDEX
      IFLBR(IARG)=IFR(8)
      IARG=20+INDEX
      IFLBR(IARG)=IFR(7)
      IARG=21+INDEX
      CALL DPACK(IQLA(1,1),IFLBR(IARG))
      IARG=53+INDEX
      IFLBR(IARG)=NSC(1)
      IARG=54+INDEX
      IFLBR(IARG)=ITCT(1)/64
      DO 6001 III=55,83
      IARG=III+INDEX
      IFLBR(IARG)=0
 6001 CONTINUE
C
C CHECK IF DONE AND SEE IF FILLER BLOCKS HAVE TO BE INSERTED.
C
      K=K+1
      IF(K .EQ. 9) GO TO 6004
      KK=II+1
      IF(SFRAME(KK,1).NE.-1.0 .AND. SFRAME(KK,2).NE.-1.0 .AND.
     *SFRAME(KK,3).NE.-1.0 .AND. IFRAME(KK).NE.-1) GO TO 3400
C
C WRITE UP FILLER BLOCKS.
C
 6002 DO 6003 III=K,8
      INDEX=(III-1)*75
      IARG=9+INDEX
      IFLBR(IARG)=25206
      IARG=10+INDEX
      IFLBR(IARG)=25
      IF(IDSS .EQ. 43) IFLBR(IARG)=134
      IVALLL=IFLBR(IARG)
      IFLBR(IARG)=MOR(IVALLL,9984)
      IARG=11+INDEX
      IFLBR(IARG)=MOR(MLSL(170,8),190)
      DO 6003 IIII=1,72
      IARG=11+INDEX+IIII
      IFLBR(IARG)=IFILL
 6003 CONTINUE
 6004 WRITE(ITAPEO,37) IFLBR
      IFLBR(1)=IFLBR(1)+1
      K=1
      KK=II+1
      IF(SFRAME(KK,1).NE.-1.0 .AND. SFRAME(KK,2).NE.-1.0 .AND.
     *SFRAME(KK,3).NE.-1.0 .AND. IFRAME(KK).NE.-1)  GO TO 3400
C
C IF ALL SELECTED FRAMES ARE PROCESSED WRITE TAPE ENDING.
C
      IFLBR(1)=IHEX1
      DO 6005 I=2,608
      IFLBR(I)=0
 6005 CONTINUE
      WRITE(ITAPEO,37) IFLBR
      END FILE ITAPEO
      END FILE ITAPEO
      GO TO 4700
      END
```

```
      SUBROUTINE REVSIM(IFR)
C
C
C THIS SUBROUTINE REVERSES THE SYMBOLS IN AN INPUT DATA
C RECORD.
C
C DIMENSION THE ARRAY AND DECLARE THE HALFWORD INTEGERS.
C
      INTEGER*2 IFR(521),IVAL1,IVAL2,IFLIP
C
C REVERSE EACH WORD AND REVERSE THE SYMBOLS IN EACH WORD.
C
      DO 100 I=1,256
      IARG=521-I+1
      IVAL1=IFLIP(IFR(I+9))
      IVAL2=IFLIP(IFR(IARG))
      IFR(I+9)=IVAL2
      IFR(IARG)=IVAL1
  100 CONTINUE
      RETURN
      END
      FUNCTION IHARD(I1)
C
C
C THIS FUNCTION GENERATES A HARD DECISION.
C
      INTEGER*2 I1
      IHARD=1
      IF(I1.LE.4) IHARD=0
      RETURN
      END
      INTEGER FUNCTION MAP*2(I1)
      COMMON IPHASE
      DATA IDB/'DB  '/
C
C
C FUNCTION TO MAP SOFT-DECISIONS.
C
      INTEGER*2 I1
      IF(IPHASE.EQ.IDB) GO TO 100
C
C 'DATA' PHASE REFERENCE.
C
      MAP=12-I1
      IF(I1.LT.4) MAP=4-I1
      RETURN
C
C
C 'DATA-BAR' PHASE REFERENCE.
C
  100 MAP=5+I1
      IF(I1.GT.3) MAP=I1-3
      RETURN
      END
      SUBROUTINE SYNC(JS,NREC,IDISK,IFRB,IWORK,IRETA,
     *IQLA,NFRMS,ESNO,IFFRT,ISYCW,IV)
C
C
C THIS SUBROUTINE LOCATES THE SYNC WORD.
```

```
C
C DECLARE VARIABLES AND DIMENSION ARRAYS.
C
      INTEGER*2 IFRB(521,9),IWORK(1025,9),IRETA(1024,8),
     *IQLA(512,9),ISWORD(23)
      DIMENSION ESNO(8),IPOS(2),ISYNC(2)
      DATA ISWORD/1,1,1,1,1,0,0,0,1,1,0,0,0,1,
     *0,1,0,1,0,0,1,0,0/
C
C SEE WHAT WE HAVE TO WORK WITH.
C
      IFFRT=NFRMS
      IF(NFRMS+JS.GT.NREC) IFFRT=NREC-JS
      IF(IFFRT.LE.0) GO TO 1100
      IFRT1=IFFRT+1
      JJ=JS
C
C READ THE FRAMES IN.
C
      DO 100 I=1,IFRT1
      INDEX=JJ+I-1
      IV=INDEX
      READ(IDISK) (IFRB(J,I),J=1,521)
C
C GET THE SIGNAL-TO-NOISE RATIO.
C
      IESNO=IFRB(8,I)
      ESNO(I)=FLOAT(IESNO)/128.0
  100 CONTINUE
C
C EXPAND THE FRAMES.
C
      DO 200 I=1,IFRT1
      CALL UNPCK(IFRB(1,I),IWORK(1,I))
  200 CONTINUE
C
C GO THROUGH BOTH SYMBOL PAIRS.
C
      ISYNC(1)=-25
      ISYNC(2)=-25
      DO 500 I=1,2
C
C COMBINE THE SYMBOLS.
C
      DO 300 J=1,IFRT1
      DO 300 K=1,512
      INDEX=2*K+I-2
      INDEX1=INDEX+1
      IVAL=IHARD(IWORK(INDEX,J))
      IVAL1=1-IHARD(IWORK(INDEX1,J))
      IQLA(K,J)=MXOR(IVAL,IVAL1)
  300 CONTINUE
C
C NOW SEARCH FOR THE SYNC WORD.
C
      DO 500 J=1,512
      ICOL=0
      DO 400 K=1,IFRT1
      DO 400 L=1,23
      LL=J+L-1
      IF(LL.GT.512) LL=LL-512
      IOR=1
```

```
      IF(ISWORD(L).NE.IQLA(LL,K)) IOR=-1
      ICOL=ICOL+IOR
  400 CONTINUE
      IF(ICOL.LT.ISYNC(I)) GO TO 500
      IPOS(I)=J
      ISYNC(I)=ICOL
  500 CONTINUE
C
C  LOCATE SYNC WORD.
C
      I=1
      IF(ISYNC(2).GT.ISYNC(1)) I=2
      LL=IPOS(I)+24-1
      IF(LL.GT.512) LL=LL-512
      ISYCW=LL
      LL=LL*2-2+I
      LLL=LL
C
C NOW SET UP THE RETURN ARRAY WITH THE SOFT-DECISIONS.
C
      DO 800 I=1,IFFRT
      DO 700 J=1,1024
      JJ=LL+J-1
      IF(JJ.NE.1025) GO TO 600
      JJ=JJ+1
      LL=LL+1
  600 IRETA(J,I)=IWORK(JJ,I)
  700 CONTINUE
      LL=LLL
  800 CONTINUE
C
C NOW GO BACK AND CALCULATE THE HARD-DECISIONS AND
C THE QUICK-LOOK OUTPUTS.
C
      DO 1000 I=1,IFFRT
      DO 900 J=1,511
      INDEX=2*J+1
      INDEX1=INDEX+1
      IVAL=IHARD(IRETA(INDEX,I))
      IVAL1=1-IHARD(IRETA(INDEX1,I))
      IRETA(INDEX1,I)=9-IRETA(INDEX1,I)
      IQLA(J,I)=MXOR(IVAL,IVAL1)
  900 CONTINUE
      IRETA(2,I)=9-IRETA(2,I)
      IQLA(512,I)=1
 1000 CONTINUE
 1100 RETURN
      END
      SUBROUTINE FRPRNT(NFR,IQLA,ACB,NSC,IDFFLG,IDISK,IFR,
     *IDEC,ICNT,IPRINT,ISYCW,ESNO,IV)
C
C
C THIS SUBROUTINE PRINTS OUT FRAMES.
C
C DECLARE VARIABLES AND DIMENSION ARRAYS.
C
      INTEGER*2 IQLA(512,8),IFR(521),IFRD(2)
      DIMENSION ACB(8),NSC(8),IDFFLG(8),ESNO(8)
      EQUIVALENCE (IVAL,IFRD(1))
      DATA IDECA/'FANO'/
C
C FORMAT STATEMENTS USED BY ROUTINE.
C
```

```
   10 FORMAT(1H0,//6X,'PRINTOUT OF DECODED DATA FOR FRAME ',I4,//6X,
     *'DATA START TIME-',/10X,
     *'HOURS-     ',I2,/10X,'MINUTES- ',
     *I2,/10X,'SECONDS- ',F6.3)
   11 FORMAT(1H0,/6X,'DATA STOP TIME-',/10X,'HOURS-     ',I2,/10X,
     *'MINUTES- ',I2,/10X,'SECONDS- ',F6.3)
   12 FORMAT(1H0,5X,'FRAME WAS DECODED SUCCESSFULLY.')
   13 FORMAT(1H0,5X,'FRAME WAS DELETED.')
   14 FORMAT(1H0,5X,'FRAME WAS DECODED USING QUICK-LOOK ',
     *'ALGORITHM.')
   15 FORMAT(1H0,5X,'FRAME WAS DECODED USING FANO ',
     *'ALGORITHM.')
   16 FORMAT(1H0,5X,'NUMBER OF SYMBOL ERRORS CORRECTED= ',
     *I4)
   17 FORMAT(1H0,5X,'AVERAGE NUMBER OF COMPUTATIONS PER BIT=',F14.7)
   18 FORMAT(1H0,5X,'DECODED DATA-',//3X,64I1,/3X,64I1,
     */3X,64I1,/3X,64I1,/3X,64I1,/3X,64I1,/3X,64I1,/3X,64I1)
   19 FORMAT(1H0,'                           TAIL SEQUENCE SHOULD BE-   ',
     *'111110001100010101001001')
   20 FORMAT(1H0,5X,'INVALID DECODER PARAMETER.')
   21 FORMAT(1H0,5X,'SYNC WORD LOCATION=',I5)
   22 FORMAT(1H0,5X,'SIGNAL-TO-NOISE RATIO (DB)=',G17.10)
   23 FORMAT(1H ,5X,'(# BITS DECODED/MAX. # COMP. PER FRAME)')
C
C LOOP THROUGH EACH FRAME.
C
      DO 600 I=1,ICNT
C
C GET DATA START TIME.
C
      INDEX=NFR+I-1
      IV=INDEX
      READ(IDISK) IFR
      IFRD(1)=IFR(3)
      IFRD(2)=IFR(4)
      ISEC=MAND(IVAL,131071)
      IMIL=IFR(5)
      IHOUR=ISEC/3600
      ISEC=ISEC-(IHOUR*3600)
      IMIN=ISEC/60
      ISEC=ISEC-(IMIN*60)
      SECOND=FLOAT(ISEC)+.001*FLOAT(IMIL)
C
C PRINT OUT DATA START TIME.
C
      WRITE(IPRINT,10) INDEX,IHOUR,IMIN,SECOND
C
C GET DATA START TIME.
C
      INDEX1=INDEX+1
      IV=INDEX1
      READ(IDISK) IFR
      IFRD(1)=IFR(3)
      IFRD(2)=IFR(4)
      ISEC=MAND(IVAL,131071)
      IMIL=IFR(5)
      IHOUR=ISEC/3600
      ISEC=ISEC-(IHOUR*3600)
      IMIN=ISEC/60
      ISEC=ISEC-(IMIN*60)
      SECOND=FLOAT(ISEC)+.001*FLOAT(IMIL)
```

```
C
C PRINT OUT DATA STOP TIME.
C
      WRITE(IPRINT,11) IHOUR,IMIN,SECOND
C
C PRINT OUT SYNC WORD LOCATION AND S/N RATIO
C
      WRITE(IPRINT,21) ISYCW
      WRITE(IPRINT,22) ESNO(I)
C
C CHECK ON DECODER.
C
      IF(IDEC.EQ.IDECA) GO TO 100
C
C QUICK-LOOK DECODER USED.
C
      WRITE(IPRINT,14)
      GO TO 200
C
C FANO DECODER USED.
C
  100 WRITE(IPRINT,15)
C
C SEE IF FRAME WAS DECODED SUCCESSFULLY.
C
  200 IF(IDEC.EQ.IDECA.AND.IDFFLG(I).NE.0) GO TO 300
C
C FRAME WAS DECODED.
C
      IF(IDEC.NE.IDECA) GO TO 400
      WRITE(IPRINT,12)
      GO TO 400
C
C FRAME WAS DELETED.
C
  300 WRITE(IPRINT,13)
C
C OUTPUT NUMBER OF SYMBOL ERRORS AND AVERAGE NUMBER OF
C COMPUTATIONS PER BIT.
C
  400 IF(IDEC.NE.IDECA) GO TO 500
      WRITE(IPRINT,16) NSC(I)
      WRITE(IPRINT,17) ACB(I)
      IF(IDFFLG(I) .EQ. 0)  GO TO 500
      WRITE(IPRINT,23)
C
C OUTPUT THE DECODED FRAME.
C
  500 WRITE(IPRINT,18) (IQLA(J,I),J=1,512)
      IF(IDFFLG(I).EQ.0.OR.IDEC.NE.IDECA) GO TO 600
      IF(IDFFLG(I).NE.1) GO TO 550
      WRITE(IPRINT,19)
      GO TO 600
  550 WRITE(IPRINT,20)
  600 CONTINUE
      RETURN
      END
      SUBROUTINE FRFIND(II,SFRAME,IDISK,IFR,NREC,NFR,IV)
C
C
C THIS SUBROUTINE LOCATES THE FRAME WITH THE GIVEN TIME
C TIME TAG. IF NO FRAME CAN BE FOUND, A -1 IS RETURNED
```

```
C AS THE FRAME NUMBER.
C
C DECLARE VARIABLES AND DIMENSION ARRAYS.
C
      INTEGER*2 IFR(521),IFRD(2)
      EQUIVALENCE (IVAL,IFRD(1))
      DIMENSION SFRAME(100,3)
C
C GET THE TIME REQUESTED IN SECONDS.
C
      TIME=SFRAME(II,1)*3600.0+SFRAME(II,2)*60.0+SFRAME
     *(II,3)-0.005
C
C SET FRAME NOT READY FLAG.
C
      NFR=-1
C
C GO THROUGH THE RECORDS AND FIND THE TIME.
C
      K=NREC-1
      DO 100 I=1,K
      IV=I
      READ(IDISK) IFR
      IFRD(1)=IFR(3)
      IFRD(2)=IFR(4)
C
C GET THE TIME FOR EACH RECORD.
C
      ISEC=MAND(IVAL,131071)
      IMIL=IFR(5)
      TIMES=FLOAT(ISEC)+.001*FLOAT(IMIL)
      IV=I+1
      READ(IDISK) IFR
      IFRD(1)=IFR(3)
      IFRD(2)=IFR(4)
      ISEC=MAND(IVAL,131071)
      IMIL=IFR(5)
      TIMES1=FLOAT(ISEC)+0.001*FLOAT(IMIL)
      NFR=I
      IF(TIME.GE.TIMES .AND. TIME.LT.TIMES1) GO TO 200
      IF(TIME.GE.TIMES1 .AND. TIME.LT.TIMES) GO TO 200
  100 CONTINUE
      IF(NFR .EQ. NREC) NFR=-1
C
C NO FRAME FOUND. BAIL OUT.
C
  200 RETURN
      END
      SUBROUTINE RSPRNT(NFR,IDISK,IFR,IPRINT,IWORK,NREC,ESNO,
     *IRWD,IPHASE,IV)
C
C
C SUBROUTINE TO PRINT OUT RAW SYMBOLS.
C
C DECLARE VARIABLES AND DIMENSION ARRAYS.
C
      DIMENSION ESNO(8)
      INTEGER*2 IFR(521),IFRD(2),IWORK(1025),MAP
      EQUIVALENCE (IVAL,IFRD(1))
      DATA IRDF,IDBAR/'SYNC','DB  '/
```

```
C
C FORMAT STATEMENTS USED BY ROUT.NE.
C
    9 FORMAT(1H0,//6X,'PRINTOUT OF RAW CHANNEL ',
     *'SYMBOLS FOR SYNCHRONIZED DATA FRAME ',
     *I4,//6X,'DATA START TIME-',/10X,'HOURS-    ',
     *I2,/10X,'MINUTES- ',I2,/10X,'SECONDS- ',F6.3)
   10 FORMAT(1H0,//6X,'PRINTOUT OF RAW CHANNEL ',
     *'SYMBOLS FOR TAPE FRAME (RECORD) ',I4,//6X,'DATA START TIME-',
     */10X,'HOURS-    ',I2,/10X,'MINUTES- ',I2,/10X,'SECONDS- ',F6.3)
   11 FORMAT(1H0,/6X,'DATA STOP TIME-',/10X,'HOURS-    ',I2,/10X,
     *'MINUTES- ',I2,/10X,'SECONDS- ',F6.3)
   12 FORMAT(1H0,/6X,'RAW CHANNEL SYMBOLS ',
     *'(SOFT-DECISIONS IN OCTAL FORMAT)-')
   13 FORMAT(1H0,//6X,'RAW CHANNEL SYMBOLS ',
     *'(HARD-DECISIONS IN BINARY FORMAT)-')
   14 FORMAT(/3X,64I1,
     */3X,64I1,/3X,64I1,/3X,64I1,/3X,64I1,/3X,64I1,
     */3X,64I1,/3X,64I1,/3X,64I1,/3X,64I1,/3X,64I1,
     */3X,64I1,/3X,64I1,/3X,64I1,/3X,64I1,/3X,65I1)
   15 FORMAT(1H0,5X,'SYNC WORD LOCATION=',I5)
   16 FORMAT(1H0,5X,'SIGNAL-TO-NOISE RATIO (DB)=',G17.10)
   17 FORMAT(1H0,/5X,'SYNCHRONIZED RAW CHANNEL SYMBOLS ',
     *'(SOFT-DECISIONS IN OCTAL FORMAT)-')
   18 FORMAT(1H0,//5X,'SYNCHRONIZED RAW CHANNEL SYMBOLS ',
     *'(HARD-DECISIONS IN BINARY FORMAT)-')
C
C GET START TIME FOR THE FRAME AND PRINT IT OUT.
C
      IF(NFR.LT.1.OR.NFR.GT.NREC) GO TO 200
      IV=NFR
      READ(IDISK) IFR
      IFRD(1)=IFR(3)
      IFRD(2)=IFR(4)
      ISEC=MAND(IVAL,131071)
      IMIL=IFR(5)
      IHOUR=ISEC/3600
      ISEC=ISEC-(IHOUR*3600)
      IMIN=ISEC/60
      ISEC=ISEC-(IMIN*60)
      SECOND=FLOAT(ISEC)+.001*FLOAT(IMIL)
C
C GET THE CHANNEL SYMBOLS UNPACKED BEFORE WRITING OVER THE FRAME.
C THEN GET STOP TIME.
C
C FIRST CHECK IF PRINTED RAW SYMBOLS ARE TO BE SYNCHRONIZED.
C
      IF(IRWD .NE. IRDF) GO TO 20
      WRITE(IPRINT,9) NFR,IHOUR,IMIN,SECOND
      GO TO 30
   20 WRITE(IPRINT,10) NFR,IHOUR,IMIN,SECOND
      CALL UNPCKS(IFR,IWORK)
   30 IF(NFR+1.GT.NREC) GO TO 100
      IV=NFR+1
      READ(IDISK) IFR
      IFRD(1)=IFR(3)
      IFRD(2)=IFR(4)
      ISEC=MAND(IVAL,131071)
      IMIL=IFR(5)
      IHOUR=ISEC/3600
      ISEC=ISEC-(IHOUR*3600)
      IMIN=ISEC/60
      ISEC=ISEC-(IMIN*60)
      SECOND=FLOAT(ISEC)+.001*FLOAT(IMIL)
      WRITE(IPRINT,11) IHOUR,IMIN,SECOND
```

```
C
      IF(IRWD .NE. IRDF)  GO TO 100
C
C     WRITE OUT CHANNEL SYMBOLS AS THEY APPEAR IN A SYNCHRONIZED
C     DATA FRAME.  FIRST UN-MAP THE SYMBOLS.
C
      IF(IPHASE .EQ. IDBAR)  GO TO 40
      DO 35  I=1,1024
      M=IWORK(I)
      I1=12-M
      IF(M .LE. 4)  I1=4-M
      IWORK(I)=I1
   35 CONTINUE
      GO TO 46
   40 DO 45  I=1,1024
      M=IWORK(I)
      I1=M-5
      IF(M .LE. 4)  I1=M+3
      IWORK(I)=I1
   45 CONTINUE
C
C WRITE OUT CHANNEL SYMBOLS FOR (SYNC) FRAME.
C
   46 WRITE(IPRINT,15) IWORK(1025)
      WRITE(IPRINT,16) ESNO(1)
      WRITE(IPRINT,17)
      WRITE(IPRINT,14) (IWORK(I), I=1,1024)
      DO 50  I=1,1024
      IWORK(I)=IHARD(MAP(IWORK(I)))
   50 CONTINUE
      WRITE(IPRINT,18)
      WRITE(IPRINT,14) (IWORK(I), I=1,1024)
      RETURN
C
C WRITE OUT CHANNEL SYMBOLS FOR TAPE RECORD.
C
  100 WRITE(IPRINT,16) ESNO(1)
      WRITE(IPRINT,12)
      WRITE(IPRINT,14) IWORK
      DO 150  I=1,1025
      IWORK(I)=IHARD(MAP(IWORK(I)))
  150 CONTINUE
      WRITE(IPRINT,13)
      WRITE(IPRINT,14) IWORK
  200 RETURN
      END
      SUBROUTINE UNPCK(IFR,IWORK)
C
C THIS SUBROUTINE UNPACKS A FRAME OF CHANNEL SYMBOLS
C FROM A DATA RECORD.
C
C DIMENSION ARRAYS AND DECLARE VARIABLES.
C
      INTEGER*2 IFR(521),IWORK(1025),IS0,IS1,IS2,MAP
C
C GO THROUGH THE DATA RECORD AND UNPACK EACH SYMBOL.
C
      DO 100 I=1,512
      INDEX=I+9
      CALL CHSYM(IFR(INDEX),IS0,IS1,IS2)
      IND=2*I-1
```

```
      IWORK(IND)=MAP(IS0)
      IWORK(IND+1)=MAP(IS1)
  100 CONTINUE
      CALL CHSYM(IFR(521),IS0,IS1,IS2)
      IWORK(1025)=MAP(IS2)
      RETURN
C
      ENTRY UNPCKS(IFR,IWORK)
      DO 200  I=1,512
      INDEX=I+9
      CALL CHSYM(IFR(INDEX),IS0,IS1,IS2)
      IND=2*I-1
      IWORK(IND)=IS0
      IWORK(IND+1)=IS1
  200 CONTINUE
      CALL CHSYM(IFR(521),IS0,IS1,IS2)
      IWORK(1025)=IS2
      RETURN
      END
```

Appendix B

Fano Algorithm Subroutine Fortran Code

```
      SUBROUTINE FANO(ESNODB,IN,IOUT,ITCT,ACB,NSC,IDFFLG,ITPFLG)
C
C     FORTRAN FANO SEQUENTIAL DECODER FOR OCTAL CHANNEL --
C
C          THIS PROGRAM SEQUENTIALLY DECODES FRAMES OF 1024 CHANNEL
C     SYMBOLS USING SOFT-DECISIONS ON THE DEMODULATED CHANNEL
C     OUTPUT.  EACH FRAME CORRESPONDS TO 512 INFORMATION BITS OF
C     WHICH THE LAST 24 IS A KNOWN FRAME SYNC WORD WITH THE FOLLOWING
C     PATTERN:  111110001100010101001001.  THIS KNOWN TAIL SEQUENCE
C     IS USED BY THE FANO ALGORITHM IN THE DECODING PROCESS.
C
C     IN = INPUT ARRAY OF 1024 SOFT-DECISIONS THAT ARE INPUT
C          TO THE FANO DECODER
C     IOUT = OUTPUT ARRAY OF 512 DECODED INFORMATION BITS
C     ITCT = ON ENTRANCE TO ROUTINE, IS THE MAXIMUM NUMBER
C            OF COMPUTATIONS ALLOWED PER FRAME
C     ACB = AVERAGE NUMBER OF COMPUTATIONS PER INFORMATION BIT
C           DECLARED AS BEING DECODED.
C     NSC = NUMBER OF CORRECTED CHANNEL SYMBOLS
C           ( =0 WHEN FRAME IS DELETED)
C     IDFFLG = DELETED FRAME FLAG
C              0 = FRAME SUCCESSFULLY DECODED
C              1 = FRAME DELETED - MAXIMUM NUMBER OF COMPUTATIONS
C                  ALLOWED PER FRAME HAS BEEN EXCEEDED
C              2 = INVALID DECODER PARAMETER
      DIMENSION LI(600),IREG(600),LMDA(600),IP0(600),IP1(600)
     *          ,QL(7),TP(8),D(8),QV(64),M00(64),M01(64),ITAIL(36)
     *          ,INHD(1500)
      INTEGER*2 IN(1024),IOUT(512)
C
C        ESNODB = SIGNAL ENERGY TO NOISE DENSITY RATIO IN DB
C        QSP = QUANTIZATION SPACING
C              (NORMALIZED TO STANDARD DEVIATIONS OF THE NOISE)
C        IFL = TOTAL FRAME LENGTH
C        ITL = LENGTH OF FIXED TAIL SEQUENCE
C        ITRIMA = MAXIMUM TRIALS PER BLOCK
C              (SET EQUAL TO ITCT INITIALLY)
C        ITAP0A,ITAP1A = REGISTER TAP MASKS, LEFT JUSTIFIED
C        BIAS = METRIC BIAS PER BRANCH
C        TO = THRESHOLD SPACING
C        SCALE = SCALE FACTOR FOR INTEGER COMPUTATIONS
      ITRIMA=ITCT
      DATA ITP /'TAPE'/
      DATA QSP /0.5/
      DATA ITAP0A,ITAP1A /ZADD6F7DD,ZEDD6F7DD/
      DATA BIAS /1.0/
      DATA TO /3.0/
      DATA SCALE /1000.0/
      DATA IFL /512/
      DATA ITL /24/
      DATA ITAIL(1),ITAIL(2),ITAIL(3),ITAIL(4) /1,1,1,1/
      DATA ITAIL(5),ITAIL(6),ITAIL(7),ITAIL(8) /1,0,0,0/
      DATA ITAIL(9),ITAIL(10),ITAIL(11),ITAIL(12) /1,1,0,0/
      DATA ITAIL(13),ITAIL(14),ITAIL(15),ITAIL(16) /0,1,0,1/
      DATA ITAIL(17),ITAIL(18),ITAIL(19),ITAIL(20) /0,1,0,0/
      DATA ITAIL(21),ITAIL(22),ITAIL(23),ITAIL(24) /1,0,0,1/
      DATA RALOG2 /1.442695/
C
      ISIGN=-1073741824
      ISIGN=ISIGN+ISIGN
      MAX=2*IFL
```

```
C
C     PROGRAM INITIALIZATION
C     INITIALIZE CHOICE ARRAY
C     LI = 0 FOR BEST CHOICE (0 ON EVEN NODE TIE, 1 ON ODD)
C          1 FOR WORST CHOICE (1 ON EVEN NODE TIE, 0 ON ODD)
C          2 FOR KNOWN BIT IN TAIL SEQUENCE
      DO 32 I=1,IFL
   32 LI(I)=0
      IFLM1=IFL-1
      IFLP1=IFL+1
      ITB=IFLP1-ITL
      DO 33 I=ITB,IFLP1
   33 LI(I)=2
      ITO=TO*SCALE
      ESNO=EXP(ESNODB/10.0*ALOG(10.0))
      TH=SQRT(2.0*ESNO)
C
C     CALCULATE CHANNEL TRANSITION PROBABILITIES
      DO 34 I=1,7
   34 QL(I)=TH-FLOAT(4-I)*QSP
      QV(1)=0.5*(2.0-ERFC(0.7071068*QL(1)))
      TP(1)=QV(1)
      DO 35  I=2,7
      QV(I)=0.5*(2.0-ERFC(0.7071068*QL(I)))
   35 TP(I)=QV(I)-QV(I-1)
      TP(8)=1.0-QV(7)
      PE=1.0-QV(4)
C
C     CALCULATE BRANCH METRICS
      DO 36  I=1,8
      IM9=9-I
   36 D(I)=ALOG(TP(I)*2.0/(TP(I)+TP(IM9)))*RALOG2
      DO 37  I=1,8
      DO 37  J=1,8
      MET=(D(I)+D(J)-BIAS)*SCALE
      K=(I-1)*8+J
      M00(K)=MET
      K=(I-1)*8+9-J
   37 M01(K)=MET
C
C     OBTAIN HARD-DECISIONS FROM RECEIVED CHANNEL SYMBOLS
      DO 41  J=1,MAX
      IF(IN(J) .GE. 5)  GO TO 40
      INHD(J)=0
      GO TO 41
   40 INHD(J)=1
   41 CONTINUE
C
C     INDEX RECEIVED SYMBOLS (SOFT-DECISIONS) INTO BRANCH METRIC ARRAY
C     EACH PAIR OF SYMBOLS DEFINES AN ARRAY ELEMENT
      K=0
      DO 50  I=1,MAX,2
      INDEX=8*(IN(I)-1)+IN(I+1)
      K=K+1
      IN(K)=INDEX
   50 CONTINUE
C
C     INITIALIZE DECODER BEFORE START OF EACH FRAME
      IP0A=0
      IP1A=0
      ITCT=0
      IT=0
      L=0
```

```
      IFLAG=0
      N=1
      LI(1)=0
C
C     DECODE IFL NODES
   80 IF(N-IFL)  81,81,310
   81 IND=IN(N)
      INDI=65-IND
      IF(IP0A) 135,131,131
  131 IF(IP1A) 134,133,133
  133 M0=M00(IND)
      M1=M00(INDI)
      GO TO 139
  134 M0=M01(IND)
      M1=M01(INDI)
      GO TO 139
  135 IF(IP1A) 137,136,136
  136 M0=M01(INDI)
      M1=M01(IND)
      GO TO 139
  137 M0=M00(INDI)
      M1=M00(IND)
C
C     CHECK FOR KNOWN BIT
  139 IF(LI(N)-2)140,165,999
  140 IF(M0-M1) 145,141,149
  141 IF(N-2*(N/2)) 999,149,145
  145 IF(LI(N)) 999,160,170
  149 IF(LI(N)) 999,170,160
C
C     TRIAL DECISION IS A ONE
  160 LMDA(N)=M1
      IREG(N)=1
      LT=L+M1
      GO TO 210
C
C     KNOWN TAIL SEQUENCE, DO NOT TIGHTEN THRESHOLD
  165 IARG=N-ITB+1
      IF(ITAIL(IARG))  999,200,201
  200 LMDA(N)=M0
      IREG(N)=0
      LT=L+M0
      IF(LT-IT)  270,166,166
  201 LMDA(N)=M1
      IREG(N)=1
      LT=L+M1
      IF(LT-IT)  270,166,166
  166 IF(IFLAG) 999,230,250
C
C     TRIAL DECISION IS A ZERO
  170 LMDA(N)=M0
      IREG(N)=0
      LT=L+M0
  210 IF(LT-IT) 270,219,219
  219 IF(IFLAG) 999,220,250
C
C     CHECK IF THRESHOLD CAN BE TIGHTENED
C     UPDATE TRIALS COUNT
  220 IF(LT-(IT+IT0)) 230,221,221
  221 IT=IT+IT0
  230 L=LT
      ITCT=ITCT+1
```

```
      IF(IREG(N)) 999,232,231
  231 IP1A=MXOR(IP1A,ITAP1A)
  232 IF(IP1A) 234,233,233
  233 IP1A=MOVEL(IP1A,1)
      IP1(N)=0
      GO TO 235
  234 IP1A=MOVEL(IP1A,1)
      IP1(N)=ISIGN
  235 IF(IREG(N)) 999,237,236
  236 IP0A=MXOR(IP0A,ITAP0A)
  237 IF(IP0A) 239,238,238
  238 IP0A=MOVEL(IP0A,1)
      IP0(N)=0
      GO TO 242
  239 IP0A=MOVEL(IP0A,1)
      IP0(N)=ISIGN
C
C     PROCEED TO NEXT NODE
  242 N=N+1
      IF(LI(N)-2) 243,80,999
  243 LI(N)=0
      GO TO 80
C
C     CHECK IF RUNNING THRESHOLD IS VIOLATED
  250 IF(LT-(IT+IT0)) 260,230,230
  260 IFLAG=0
      GO TO 230
C
C     SEARCH OTHER LIKELY PATHS - STEP DECODER BACKWARDS
  270 IFLAG=1
  280 ITCT=ITCT+1
      IF(ITCT-ITRIMA) 282,440,440
  282 IF(N-1) 999,300,284
  284 LB=L-LMDA(N-1)
      IF(LB-IT) 300,285,285
  285 N=N-1
      L=LB
      IF(IREG(N)) 999,286,287
  286 IP1A=MXOR(MOVER(IP1A,1),IP1(N))
      IP0A=MXOR(MOVER(IP0A,1),IP0(N))
      GO TO 296
  287 IP1A=MXOR(MXOR(MOVER(IP1A,1),IP1(N)),ITAP1A)
      IP0A=MXOR(MXOR(MOVER(IP0A,1),IP0(N)),ITAP0A)
  296 IF(LI(N)) 999,297,280
  297 LI(N)=1
      GO TO 81
C
C     RELAX RUNNING THRESHOLD VALUE
  300 IT=IT-IT0
      IF(LI(N)-2) 301,80,999
  301 LI(N)=0
      GO TO 80
C
C     IFL NODES HAVE BEEN DECODED OR TRIALS LIMIT EXCEEDED
  310 N=N-1
C             ITCT = ON EXIT FROM ROUTINE, NUMBER OF FORWARD AND
C                    REVERSE MOVES MADE DURING DECODING PROCESS
C             IT = FINAL THRESHOLD VALUE (SCALED)
C             L = FINAL PATH METRIC VALUE (SCALED)
C             LMDA = PATH METRIC (SCALED)
C             IREG(N) = DECODED DATA
```

```
C
C       CONVERT BRANCH SYMBOL ARRAYS TO 0,1 FORMAT
        DO 416  J=1,N
        IF(IPO(J))  410,412,999
    410 IPO(J)=1
    412 IF(IP1(J))  414,416,999
    414 IP1(J)=1
    416 CONTINUE
C
C       CALCULATE THE AVERAGE NUMBER OF COMPUTATIONS PER
C       DECODED INFORMATION BIT
        ACB=FLOAT(ITCT)/FLOAT(IFL)
C
C       DETERMINE THE NUMBER OF CORRECTED CHANNEL SYMBOLS
        NSC=0
        K=1
        DO 424  J=1,N
        IF(INHD(K)-IPO(J))  420,421,420
    420 NSC=NSC+1
    421 IF(INHD(K+1)-IP1(J))  422,423,422
    422 NSC=NSC+1
    423 K=K+2
    424 CONTINUE
        DO 430 J=1,N
        IOUT(J)=IREG(J)
    430 CONTINUE
C
C       NORMAL EXIT FROM DECODER
        IDFFLG=0
        RETURN
    440 CONTINUE
C
C       DELETE FRAME - MAXIMUM NUMBER OF COMPUTATIONS ALLOWED
C       HAS BEEN EXCEEDED
        N=N-1
        IDFFLG=1
        NSC=0
C
C       CALCULATE AVERAGE NUMBER OF COMPUTATIONS PER
C       BIT DECLARED AS BEING DECODED
        ACB=FLOAT(ITCT)/FLOAT(N)
        IF(ITPFLG .EQ. ITP)  GO TO 450
        DO 441  J=1,N
        IOUT(J)=IREG(J)
    441 CONTINUE
        M=N+1
        DO 442  J=M,512
        IOUT(J)=9
    442 CONTINUE
        RETURN
    450 ILT=IFL-ITL
        DO 451  J=1,ILT
        IOUT(J)=0
    451 CONTINUE
        DO 452  J=1,ITL
        IOUT(J+488)=ITAIL(J)
    452 CONTINUE
        RETURN
```

```
C
C      DELETE FRAME - INVALID DECODER PARAMETER WAS
C      ENCOUNTERED DURING EXECUTION
   999 IDFFLG=2
       NSC=0
       ACB=0.0
       ILT=IFL-ITL
       DO 500  J=1,ILT
       IOUT(J)=0
   500 CONTINUE
       DO 510  J=1,ITL
       IOUT(J+488)=ITAIL(J)
   510 CONTINUE
       RETURN
       END
```

Appendix C

Utility Subroutine Assembler Code

```
*
*
BEGN     CSECT READONLY
         ENTRY MXOR,MOVEL,MOVER,MAND,MLSL,MLSR
         ENTRY MOR,MOR2,MAND2,MLSL2,MLSR2,IFLIP
MXOR     B     5+5(15)           ROUTINE TO EXCLUSIVE OR THE
         DC    X'5'              CONTENTS OF TWO FULLWORDS.
         DC    CL5'MXOR'
         SAVE  (14,12)
         USING MXOR,15
         L     2,0(1)
         L     0,0(2)
         L     1,4(1)
         L     1,0(1)
         XR    0,1
         LA    15,0
         LM    2,12,28(13)
         MVI   12(13),X'FF'
         BR    14
*
*
MOVEL    B     5+5(15)           ROUTINE TO LOGICAL SHIFT LEFT ONE
         DC    X'5'              POSITION THE CONTENTS OF A FULLWORD.
         DC    CL5'MOVEL'
         SAVE  .14,12)
         USING MOVEL,15
         L     1,0(1)
         L     0,0(1)
         SLL   0,1
         LA    15,0
         LM    2,12,28(13)
         MVI   12(13),X'FF'
         BR    14
*
*
MOVER    B     5+5(15)           ROUTINE TO LOGICAL SHIFT RIGHT ONE
         DC    X'5'              POSITION THE CONTENTS OF A FULLWORD.
         DC    CL5'MOVER'
         SAVE  (14,12)
         USING MOVER,15
         L     1,0(1)
         L     0,0(1)
         SRL   0,1
         LA    15,0
         LM    2,12,28(13)
         MVI   12(13),X'FF'
         BR    14
*
*
MAND     B     5+5(15)           ROUTINE TO AND THE CONTENTS
         DC    X'5'              OF TWO FULLWORDS.
         DC    CL5'MAND'
         SAVE  (14,12)
         USING MAND,15
         L     2,0(1)
         L     0,0(2)
         L     1,4(1)
         L     1,0(1)
         NR    0,1
         LA    15,0
         LM    2,12,26(13)
         MVI   12(13),X'FF'
         BR    14
*
```

```
*
MLSL     B     5+5(15)            ROUTINE TO LOGICAL SHIFT LEFT THE
         DC    X'5'               CONTENTS OF A FULLWORD A SPECIFIED
         DC    CL5'MLSL'          NUMBER OF POSITIONS.
         SAVE  (14,12)
         USING MLSL,15
         BALR  9,0
         L     2,0(1)
         L     0,0(2)
         L     1,4(1)
         L     1,0(1)
         LA    2,0
LOOPL    SLL   0,1
         LA    2,1(2)
         CR    1,2
         BNE   LOOPL
         LA    15,0
         LM    2,12,28(13)
         MVI   12(13),X'FF'
         BR    14
*
*
MLSR     B     5+5(15)            ROUTINE TO LOGICAL SHIFT RIGHT THE
         DC    X'5'               CONTENTS OF A FULLWORD A SPECIFIED
         DC    CL5'MLSR'          NUMBER OF POSITIONS.
         SAVE  (14,12)
         USING MLSR,15
         L     2,0(1)
         L     0,0(2)
         L     1,4(1)
         L     1,0(1)
         LA    2,0
LOOPR    SRL   0,1
         LA    2,1(2)
         CR    1,2
         BNE   LOOPR
         LA    15,0
         LM    2,12,28(13)
         MVI   12(13),X'FF'
         BR    14
*
*
MOR      B     3+5(15)            ROUTINE TO INCLUSIVE OR THE
         DC    X'3'               CONTENTS OF TWO FULLWORDS.
         DC    CL3'MOR'
         SAVE  (14,12)
         USING MOR,15
         L     2,0(1)
         L     0,0(2)
         L     1,4(1)
         L     1,0(1)
         OR    0,1                INCLUSIVE OR
         LA    15,0
         LM    2,12,28(13)
         MVI   12(13),X'FF'
         BR    14
*
*
MOR2     B     5+5(15)            ROUTINE TO INCLUSIVE OR THE CONTENTS
         DC    X'5'               OF A HALFWORD AND A FULLWORD.
         DC    CL5'MOR2'
         SAVE  (14,12)
         USING MOR2,15
         L     2,0(1)             GET ADDRESS OF FIRST OPERAND
         LH    0,0(2)             LOAD REGO WITH DATA IN HALFWORD
```

```
         N     0,=X'0000FFFF'
         L     1,4(1)          GET ADDRESS OF SECOND OPERAND
         L     1,0(1)          LOAD REG1 WITH DATA IN FULLWORD
         OR    0,1             INCLUSIVE OR
         LA    15,0
         LM    2,12,28(13)
         MVI   12(13),X'FF'
         BR    14
*
*
MAND2    B     5+5(15)         ROUTINE TO AND THE CONTENTS
         DC    X'5'            OF TWO HALFWORDS.
         DC    CL5'MAND2'
         SAVE  (14,12)
         USING MAND2,15
         L     2,0(1)
         LH    0,0(2)
         L     1,4(1)
         LH    1,0(1)
         NR    0,1
         N     0,=X'0000FFFF'
         LA    15,0
         LM    2,12,28(13)
         MVI   12(13),X'FF'
         BR    14
*
*
MLSL2    B     5+5(15)         ROUTINE TO LOGICAL SHIFT LEFT THE
         DC    X'5'            CONTENTS OF A HALFWORD A SPECIFIED
         DC    CL5'MLSL2'      NUMBER OF POSITIONS.
         SAVE  (14,12)
         USING MLSL2,15
         L     2,0(1)          GET ADDRESS OF FIRST OPERAND
         LH    0,0(2)          LOAD REG0 WITH DATA IN HALFWORD
         N     0,=X'0000FFFF'
         L     1,4(1)          GET ADDRESS OF SECOND OPERAND
         L     1,0(1)          LOAD REG1 WITH SHIFT DATA
         LA    2,0
LOOPL2   SLL   0,1             SHIFT LEFT ONE POSTION
         LA    2,1(2)          INCREMENT LOOP COUNTER
         CR    1,2             CHECK IF SHIFTED ENOUGH
         BNE   LOOPL2
         LA    15,0
         LM    2,12,28(13)
         MVI   12(13),X'FF'
         BR    14
*
*
MLSR2    B     5+5(15)         ROUTINE TO LOGICAL SHIFT RIGHT THE
         DC    X'5'            CONTENTS OF A HALFWORD A SPECIFIED
         DC    CL5'MLSR2'      NUMBER OF POSITIONS.
         SAVE  (14,12)
         USING MLSR2,15
         L     2,0(1)
         LH    0,0(2)
         N     0,=X'0000FFFF'
         L     1,4(1)
         L     1,0(1)
         LA    2,0
LOOPR2   SRL   0,1
```

```
                LA    2,1(2)
                CR    1,2
                BNE   LOOPR2
                LA    15,0
                LM    2,12,28(13)
                MVI   12(13),X'FF'
                BR    14
        *
        *
        IFLIP   B     5+5(15)          ROUTINE TO SWITCH THE POSITIONS OF
                DC    X'5'             THE 3-BIT SOFT-DECISIONS PACKED
                DC    CL5'IFLIP'       INTO A HALFWORD.
                SAVE  (14,12)
                USING IFLIP,15
                L     2,0(1)           GET ADDRESS OF OPERAND
                LH    1,0(2)           GET HALFWORD OPERAND
                L     2,BM1            GET MASK FOR BITS 14-16
                NR    2,1              REG2=BITS 14-16
                SLL   2,11             SHIFT BITS TO NEW POSITION
                L     3,BM2            GET MASK FOR BITS 3-5
                NR    3,1              REG3=BITS 3-5
                SRL   3,11             SHIFT BITS TO NEW POSITION
                L     4,CBM1
                NR    1,4              AND OUT OLD BITS 14-16
                XR    1,3              OR IN NEW BITS 14-16
                L     4,CBM2
                NR    1,4              AND OUT OLD BITS 3-5
                XR    1,2              OR IN NEW BITS 3-5
                LR    0,1
                LA    15,0
                LM    2,12,28(13)
                MVI   12(13),X'FF'
                BR    14
        BM1     DC    X'00000007'
        BM2     DC    X'00003800'
        CBM1    DC    X'FFFFFFF8'
        CBM2    DC    X'FFFFC7FF'
                END
```

```
*
*  SUBROUTINE TO PACK DECODED BITS - CALL DPACK(INARR,IOUTAR)
DPACK     CSECT READONLY
          B     5+5(15)
          DC    X'5'
          DC    CL5'DPACK'
          SAVE  (14,12)
          USING DPACK,15
*  GET INPUT-OUTPUT ARRAY ADDRESSES
          L     0,0(1)            GET ADDRESS OF INARR
          L     1,4(1)            GET ADDRESS OF IOUTAR
          LA    3,0               SET REG3=0
          LR    4,3               SET REG4=0
*  INITIALIZE OUTPUT ARRAY TO ZERO
          LR    5,1               SET REG5=ADDR OF IOUTAR
LOOPZ     STH   3,0(5)            STORE 0 IN HALFWORD LOCATION
          LA    5,2(5)            GET ADDRESS OF NEXT LOCATION
          LA    4,1(4)            INCREMENT LOOP COUNTER
          C     4,=F'32'          TEST IF FINISHED
          BNE   LOOPZ
*  GET DATA FROM INPUT ARRAY AND PACK INTO HALFWORDS
          LR    5,0               SET REG5=ADDR OF INARR
          LR    7,3               INITIALIZE WORD LOOP COUNTER
LOOPW     LR    4,3               INITIALIZE BIT LOOP COUNTER
          LA    9,1               SET REG9=1
          LR    8,3               INITIALIZE PACKING REGISTER
LOOPB     SLL   8,1               SHIFT PACKING REGISTER LEFT
          LH    6,0(5)            LOAD INPUT DATA
          CR    3,6               CHECK IF BIT IS A ZERO
          BE    CONT
          XR    8,9               OR 1 TO REG8
CONT      LA    5,2(5)            GET ADDRESS OF NEXT HALFWORD
          LA    4,1(4)            INCREMENT BIT LOOP COUNTER
          C     4,=F'16'          TEST IF COUNTER=16
          BE    OUTW              THIS WORD IS PACKED
          B     LOOPB             CONTINUE PACKING THIS WORD
OUTW      STH   8,0(1)            STORE PACKED DATA IN OUTPUT
          LA    1,2(1)            GET NEXT OUTPUT WORD ADDRESS
          LA    7,1(7)            INCREMENT WORD LOOP COUNTER
          C     7,=F'32'          TEST IF COUNTER=32
          BNE   LOOPW             START PACKING NEXT WORD
*  RETURN TO CALLING PROGRAM
          LA    15,0
          LM    2,12,28(13)
          MVI   12(13),X'FF'
          BR    14
          END
```

```
*
*  SUBROUTINE TO UNPACK SOFT-DECISIONS - CALL CHSYM(IVAL,IS0,IS1,IS2)
CHSYM    CSECT READONLY
         B     5+5(15)
         DC    X'5'
         DC    CL5'CHSYM'
         SAVE  (14,12)
         USING CHSYM,15
*  GET INPUT-OUTPUT PARAMETER ADDRESSES
         L     2,0(1)         GET ADDRESS OF IVAL
         LH    0,0(2)         GET CONTENTS OF IVAL
         L     2,4(1)         GET ADDRESS OF IS0
         L     3,8(1)         GET ADDRESS OF IS1
         L     4,12(1)        GET ADDRESS OF IS2
*  MASK UNWANTED BITS AND SAVE 3-BIT CHANNEL SYMBOLS
         L     5,MK0
         NR    5,0            SAVE ONLY BITS 3-5
         SRL   5,11           LEFT JUSTIFY
         STH   5,0(2)         SAVE IN IS0
         L     5,MK1
         NR    5,0            SAVE ONLY BITS 6-8
         SRL   5,8            LEFT JUSTIFY
         STH   5,0(3)         SAVE IN IS1
         L     5,MK2
         NR    5,0            SAVE ONLY BITS 14-16
         STH   5,0(4)         SAVE IN IS2
*  RETURN TO CALLING PROGRAM
         LA    15,0
         LM    2,12,28(13)
         MVI   12(13),X'FF'
         BR    14
         DC    BL2'0000'
MK0      DC    X'00003800'
MK1      DC    X'00000700'
MK2      DC    X'00000007'
         END
```

Appendix D

Technical Memos from NASA Ames and JPL

ORIGINAL PAGE IS
OF POOR QUALITY

R. B. MILLER

JET PROPULSION LABORATORY

INTEROFFICE MEMORANDUM AUG 29 1977
#3384-77-087
25 August 1977
File________

TO: R. B. Miller

FROM: J. H. Wilcher JHW

SUBJECT: Recording Formats for the Pioneer-Venus Reverse Playback Program

The Telemetry Processor Assembly has available, as inputs from the Symbol Synchronizer Assembly (SSA), three formats for symbol data. Each of the three formats have their distinctive use in the Telemetry Processor Assembly. However, in order to best serve the needs of the Pioneer Venus Project in its decoding of the Reverse Playback data, it is proposed that the project select the format, from the three available, that best fits their needs.

The three formats available are as follows:

1. UNSYNCHRONIZED

Byte	Bits			
	0	1	2 3 4	5 6 7
1	0	0	S_0	S_1
2	0	0	S_1	S_2
3	0	0	S_2	S_3
4	0	0	S_3	S_4
5	0	0	S_4	S_5
	0	0		
	0	0		

where S_0, S_1, S_2 ... S_n are the quantized symbol with the weighting as shown in Table 1.

2. SYNCHRONIZED

Byte	Bits 0	1	2 3 4	5 6 7
1	0	0	S_0	S_1
2	0	0	S_2	S_3
3	0	0	S_4	S_5
4	0	0	S_6	S_7
5	0	0	S_8	S_9

where S_0, S_1, S_3 ... S_n are the quantized symbols with weighting as shown in Table 1.

3. BLOCK

Byte	Bits 0	1 2 3 4 5 6 7
1	S_0	—— M_0 ——
2	S_1	—— M_1 ——
3	S_2	—— M_2 ——
4	S_3	—— M_3 ——
5	S_4	—— M_4 ——

where S_0 ... Sn are the sign bits and M_0 ... Mn are the 7-bit magnitudes associated with the sign bits.

The only remaining unanswered questions pertain to: (1) the desired record length in bits (bytes); (2) what additional data, if any, is required.

JW:ab

cc: W. Frey WJF
E. C. Gatz
R. Murray

ORIGINAL PAGE IS
OF POOR QUALITY

Analog Voltage Range (From SDA)	Bits							
	Output of A/D From SSA					Formatted Output From SSA/TPA Coupler		
	Sign	1	2	3	4	1	2	3
+5V to 0.93	0	1	1	B_3	B_4	0	1	1
	0	1	0	B_3	B_4			
	0	0	1	B_3	B_4			
0.93 to 0V	0	0	0	B_3	B_4	0	B_3	B_4
0V to 0.93	1	1	1	B_3	B_4	1	B_3	B_4
0.93 to -5V	1	0	0	B_3	B_4	1	0	0
	1	0	1	B_3	B_4			
	1	1	0	B_3	B_4			

Table 1.

ORIGINAL PAGE IS OF POOR QUALITY

JET PROPULSION LABORATORY

INTEROFFICE MEMORANDUM
#3384-78-039

8 August, 1978

TO: J. H. Wilcher

FROM: R. L. Murray

SUBJECT: Digital Tape Format for the Pioneer Venus Reverse Playback Process.

The tapes produced by the Pioneer Venus Reverse Playback Process will be standard 9 track digital tapes, recorded at 1600 bpi, using phase encoding.

Each tape record will consist of 521 16-bit words (see attached figure). An End of File (EOF) mark will be written after the last record on each data tape. A detailed description of the tape format follows.

1. Word 1 contains the tape record number (0,1,2, etc. through 65,536).

2. Word 2 contains the following playback equipment lock status information:

   Bit 10 = Receiver Lock Status.

   0 = In Lock
   1 = Out of Lock

   Bit 11 = SDA Lock Status

   0 = In Lock
   1 = Out of Lock

   Bit 12 = SSA Lock Status

   0 = In Lock
   1 = Out of Lock

3. Bits 6 through 14 of word 3 contain the playback day of year in binary.

4. Bit 15 of word 3 and bits 0 through 15 of word 4 contain the time of day of playback in binary seconds (B15 of word 3 is the MSB and B15 of word 4 is the LSB).

5. Bits 6 through 15 of word 5 contain the milliseconds of second of playback in binary.

6. Word 6 contains the following playback equipment configuration information:

   Bits 0 - 3 = The receiver number (1-7)
   Bits 4 - 7 = The SDA number (1-8)
   Bits 8 -11 = The SSA number (1 or 2)
   Bits 12-15 = The TPA number (1 or 2)

7. Word 7 contains the receiver AGC reading in two's complement binary representation with the binary point located between bits 8 and 9.

8. Word 8 contains the SSA Signal-to-Noise Ratio in dbm in two's complement binary representation with the binary point located between bits 8 and 9.

9. Word 9 contains zeroes.

10. Words 10 through 521 contain quantized symbol values associated with 512 telemetry data bits (1024 symbols). So is the first symbol received by the digital recording program, S_1 the second, etc.

RLM:amb

Distribution

E. C. Gatz
R. B. Miller

PIONEER VENUS REVERSE PLAYBACK
DIGITAL TAPE FORMAT

Words / Bits 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Words						
1	RECORD NUMBER					
2	LOCK STATUS					
3	0 0 0 0 0 0			DAY OF YEAR		
4	SECONDS OF DAY					
5	0 0 0 0 0 0			MILLISECONDS		
6	RCVR	SDA		SSA	TPA	
7	RECEIVER AGC					
8	SSA SNR					
9	SPARE					
10	00	S0	S1	00	S1	S2
11	00	S2	S3	00	S3	S4
	(512 WORDS)					
521	00	S1022	S1023	00	S1023	S0124

Enclosure 1

P-V REVERSE PLAYBACK DECODING

OUTPUT TAPE FORMAT

The Output tapes shall be in the standard Pioneer Intermediate Data Record (IDR) format. These tapes will be used in the Pioneer Venus Data Records Processing System (DRPS) to generate the Experiment Data Records (EDR).

The IDR is recorded on a 9-track, 800 bpi, unlabeled magnetic tape. The recording method is NRZ-1 (non-return-to-zero, change on ONE). A "1" bit is produced by reversal of flux polarity. The general format of the IDR file is shown in Figure A.

IDR FILES

An IDR file may be of any length, depending on the amount of data on the CODED DATA tape. It shall consist of a File Label Record, Data Records, and an All Zeroes Record. An IDR file shall be an integral number of Data Records; if the last Data Record is not full, it is completed with filler blocks (defined below under Data Records). There shall be one IDR file on each output tape. The end of a tape is always indicated by a double end-of-file mark.

IDR RECORDS

a. File Label Record. The File Label Record will always be the first record of an IDR file. It consists of 608 16-bit words (9728 bits) comprised of a 16-bit record sequence number (all zeroes), 8 zero bits, 5976 bits of data, and 3728 unused bits. The data portion will contain information specifying the IDR file type and contents. Table 1 describes the format of the IDR File Label Record.

b. Data Records. The File Label Record is followed by N Data Records, where N is the appropriate number of records necessary to provide all data for a selected Probe, station, and pass. The format of the Data Record is shown in Table 2. Each Data Record consists of 608 16-bit words (9728 bits) comprised of a 15-bit record sequence number (1 through 32767), 7 additional 16-bit items of information, and 8 1200-bit blocks. The format of each 1200-bit block within the Data Record is shown in Table 3. Each block can be either a Telemetry Block or a Filler Block. (Filler Blocks are inserted as the last blocks in an IDR File, if necessary, to complete an integral number of Data Records.) A Telemetry Block consists of the information shown in Table 2. A Filler Block consists of:

Bits 1-48: same information described for the Telemetry Block in Table 2

Bits 49-1200: repetitive $4210_{(8)}$ pattern (i.e., 100010001000100010001000. . .)

c. All Zeroes Record. The last record in each IDR file shall consist of:

Bits 1-16: All 1's

Bits 17-9728: All zeroes

This record shall always precede the double end-of-file mark which indicates the end of tape.

FIGURE A - IDR FILE FORMAT

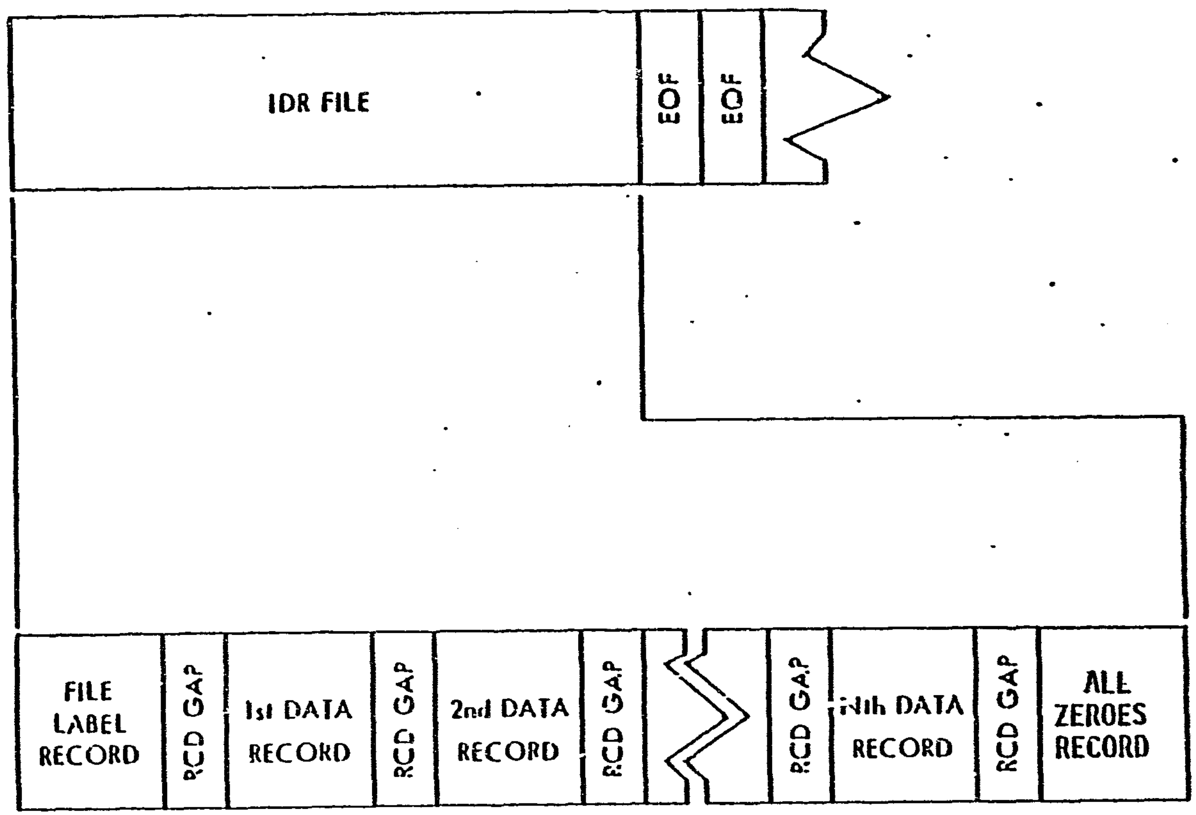


TABLE 1 - FILE LABEL RECORD FORMAT

Bit No.	Length (Bits)	Content	Description
1-16	16	All zeroes	Record Sequence No.
17-24	8	All zeroes	Spare
25-48	24	124 114 $115_{(8)}$	3 ASCII Characters ("TLM")
49-72	24	111 104 $122_{(8)}$	Tape ID; 3 ASCII Characters ("IDR")
73-96	24	$031_{(8)}$ or $206_{(8)}$	DSS Number$^{(1),(2)}$; 031 = DSS 14; 206 = DSS 43
97-120	24	$70_{(8)}$ 8 $71_{(8)}$ 9 $30460_{(8)}$ 10 $30461_{(8)}$ 11	SP#1 SP#2 SP#3 LP4 Probe ID$^{(2)}$ (ASCII)
121-152	32	All zeroes	Spare bits
153-168	16	2 ASCII Characters	Tape Sequence No.$^{(2)}$
169-192	24	"FWD" or "REV"	Data Type; 3 ASCII Characters$^{(2)}$
193-200	8	$60_{(8)}$	ASCII Character "0"
201-216	16	2 ASCII Characters$^{(1)}$	Data Start Time, Year
217-360	144	All zeroes	Bits not used
361-384	24	Binary Integer	Data Start Time, Day of Year
385-408	24	Binary Integer	Data Stop Time, Day of Year
409-432	24	Binary Integer	Data Start Time of Day (GMT) in 1/100 of second
433-456	24	Binary Integer	Data Stop Time of Day (GMT) in 1/100 of second
457-9728	9272	0 0	All zeroes

Notes:
(1) Right justified with leading character zeroes.
(2) Manual input.

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE 2 - DATA RECORD FORMAT

Bit No.	Length (Bits)	Content	Description
1-16	16	Binary Integer	Record Sequence Number: Increments 1 for each Data Record (from 1 to 32767)
17-32	16	$2300_{(8)}$	Record Size (in 8-bit bytes)
33-48	16	$0226_{(8)}$	Size of each block (in 8-bit bytes)
49-64	16	$0226_{(8)}$	Same as bits 33-48
65-80	16	$0010_{(8)}$	Number of blocks in record
81-128	48	All zeroes	Spare bits
129-1328	1200	Data Block	1st Block(1)
1329-2528	1200	Data or Filler Block	2nd Block(1)
2529-3728	1200	" " " "	3rd Block(1)
3729-4928	1200	" " " "	4th Block(1)
4929-6128	1200	" " " "	5th Block(1)
6129-7328	1200	" " " "	6th Block(1)
7329-8528	1200	" " " "	7th Block(1)
8529-9728	1200	" " " "	8th Block(1)

Note:
(1) See Table 3 for description of each block.

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE 3 - DATA BLOCK FORMAT

Bit No.	Length (Bits)	Content	Description
1-24	24	$30473047_{(8)}$	Sync Code
25-32	8	$031_{(8)}$ or $206_{(8)}$	Identifies the DSS station where the data were received: 031 = DSS14; 206 = DSS 43 (2)
33-40	8	$252_{(8)}$	Destination Code 252 = ARC
41-48	8	$336_{(8)}$ or $276_{(8)}$	Block Format Code: 336 = Data Block; 276 = Filler Block (Filler Blocks inserted as the last blocks of an IDR file, if necessary to complete an integral number of Data Records)
49-51 (1)	3	$5_{(8)}$	Gross Data Description
52-58 (1)	7	$013_{(8)}$	User Dependent Type Code
59-65 (1)	7	$114_{(8)}$ or $106_{(8)}$	Data Dependent Type Code: (2) 114 = Large Probe; 106 = Small Probe
66-72 (1)	7		Spacecraft No. (2):
		$010_{(8)}$	Small Probe #1 = 010
		or $011_{(8)}$	Small Probe #2 = 011
		or $012_{(8)}$	Small Probe #3 = 012
		or $013_{(8)}$	Large Probe = 013
73-96 (1)	24	Binary Integer	Greenwich Mean Time (of day) in binary 1/100 seconds at which the last symbol of the block was received at the station
97-98 (1)		0	Spare Bits
99-108 (1)			Day of Year Code:
(99-100) (1)	2	Binary Integer	Hundreds digit
(101-104) (1)	4	Binary Integer	Tens digit
(105-108) (1)	4	Binary Integer	Units digit
109-120 (1)	12	Binary Integer	Block Serial Number which increments one count for each data block in a file

ORIGINAL PAGE IS
OF POOR QUALITY

Table 3 - Data Block Format (cont'd.)

Bit No.	Length (Bits)	Content	Description
121-128[1]	8	Binary Integer	Millisecond Clock: a binary representation of the least significant decimal digit of the Greenwich Mean Time (derived from the same clock as that in bits 73-96)
129-136[1]	8	$200_{(8)}$ or $202_{(8)}$	Telemetry Status: $200_{(8)}$ = Decoded Frame; $202_{(8)}$ = Deleted Frame
137-160[1]	24	All zeroes	Spare Bits
161-176[1]	16	Two's Complement Binary No.	Symbol SNR[3]: bit 161 is the sign bit (0=positive, 1=negative) and the binary point is located between bits 169 and 170
177-192[1]	16	Two's Complement Binary No.	Receiver AGC[3]: bit 177 is the sign bit (0=positive, 1=negative) and the binary point is located between bits 185 and 186
193-704[1]	512	Data for decoded frame or deleted frame	Decoded Frame: Bit 193 is the first bit of the frame and bit 704 is the last bit of the Frame Sync Word (i.e., bits 681 through 704 contain the Frame Sync Word; 111110001100010101001001) Deleted Frame: Bits 193 through 680 are all zeroes, bits 681 through 704 contain the Frame Sync Word defined above.
705-720[1]	16	Binary Integer	Number of symbol errors in the frame
721-736[1]	16	Binary Integer	Number of computations performed in decoding the frame, in binary divided by 64, right justified

ORIGINAL PAGE IS
OF POOR QUALITY

Table 3 - Data Block Format (cont'd.)

Bit No.	Length (Bits)	Content	Description
737-1200[1]	464	All zeroes	Unused bits

Notes:

(1) Filler Blocks contain a repetitive $4210_{(8)}$ (i.e., 100010001000) pattern in bits 49-1200 of each block.

(2) Manual input.

(3) Manual input if it is not available on the Coded Data Tape.

Enclosure 2

P-V REVERSE PLAYBACK DECODING

PRINTOUT FORMAT

The computer printout shall contain the following information:

1. Title Block:

   a. Probe Identification

   b. DSS Number

   c. Data Start Time

2. For Each Telemetry Frame Which is Printed:

   a. Frame number (i.e., block number) in decimal integer.

   b. Frame start time and stop time in hours, minutes, and seconds (seconds to 3 decimal places).

   c. Indication of whether it is a Deleted Frame.

   d. Indication of whether the frame was decoded by means of the "Quick-Look" decoding algorithm or the Fano algorithm; or whether it is a printout of the raw undecoded data symbols.

   e. Number of symbol errors corrected (where applicable).

   f. Average computations per bit, ACB (where applicable). The ACB is the number of computations performed, divided by the number of bits (512) in the frame.

   g. The 512 bits of decoded data or the raw data symbols (undecoded) in octal and binary form.

3. Summary Block:

   a. Total number of frames processed.

   b. Total number of frames deleted (where applicable).

   c. Deletion rate (where applicable); i.e., number of frames deleted, divided by number of frames processed.

   d. Symbol error rate (where applicable); i.e., number of symbols corrected, divided by the total number of symbols processed.

NASA-Ames