X-564-66-220

NASA TM X- 55489

TAPE GENERATION PROGRAM

USERS' MANUAL

GPO PRICE $ ______

CFSTI PRICE(S) $ ______

Hard copy (HC) 3.00

Microfiche (MF) .75

ff 653 July 65

MAY 1966

NASA

GODDARD SPACE FLIGHT CENTER

GREENBELT, MD.

X-564-66-220
May 1966

TAPE GENERATION PROGRAM

(Users' Manual)

Prepared by

Mr. Richard C. Lee
Code 564

Goddard Space Flight Center
Greenbelt, Maryland

CONTENTS

Page

I. INTRODUCTION . . . . I- 1

II. RUN SPECIFICATIONS . . . . II- 1
1. Basic Parameters . . . . II- 1
2. Data Record Format Definition . . . . II- 4
a. Data Channels . . . . II- 4
b. Time . . . . II- 7
c. Flags II-10
d. Main Frame Sync Strategy II-15
3. Experiment Data II-16
a. Constant II-18
b. Sine Wave II-19
c. Stair Step II-19
d. Triangle II-19
e. Table Look Up II-20
f. User Subroutine II-20
4. Subcommutator II-21
a. Defining Subcoms II-21
b. Subcom Sync Strategy II-23
5. Perturbations II-25
a. Experiment Noise II-25
b. Transmission Noise II-26
c. Equipment Errors II-26
d. Operator Error II-28
6. Code Conversion II-28

III. OUTPUT . . . . III- 1
1. Tape Format . . . . III- 1
2. Printed Output . . . . III- 1

IV. DECK SETUP . . . . IV- 1

Appendices . . . . A- 1

Summary of Input . . . . A- 1
Summary of Limitations . . . . A- 9
Special Cases A-10
Test Case Using AE-B Satellite Format A-11

I. INTRODUCTION

The STARS data processing lines are currently used to convert satellite telemetry data into a form suitable for input to a general purpose digital computer. The output tape from the STARS lines is referred to as a buffer tape. Subsequent computer programs produce a variety of output data tapes, such as the edit tape, decommutated experimenter tape, spacecraft data tape, etc. It is desirable to simulate these telemetry computer tapes in advance of satellite fabrication and launch, for the timely development and checkout of the associated computer programs.

This program for the UNIVAC 1107 simulates satellite telemetry tapes derived from the STARS data processing lines. The file structure (number of files per tape, records per file, telemetry frames per record, format of each frame, etc.) and data curve definitions are made by input parameter. The tape label, and all parts of the frame format are optional. These optional features defined by input parameters make the program flexible enough to simulate almost any tape with fixed length records.

This document attempts to point out the various features of the program along with the input cards necessary to make use of the features. It also provides a summary of possible input for the user already familiar with the program.

The alphabetical information indicated in columns 1-6 of the input cards identifies the information for the program. Thus these columns should be punched exactly as shown. Column 10 is also used in the same manner. If the information in columns 1-6 is not valid or column 10 is not correct, a diagnostic message is printed. Columns to the right of all data fields on the cards may be used for comments as these columns are not used by the program. The notation XXX used in a data field indicates a positive or negative integer unless a specific notation is given that a fractional value is permitted. In this case, no decimal is needed if an integer is used but the decimal must be punched to obtain fractional values. The number of X's typed in the specific data fields usually indicates the maximum number of digits needed for that item. Ten columns are always read, however, for all data fields. Therefore, the limit of a particular value should be obtained from the table in the appendix and not from the number of X's on the data card. In the card formats which follow, the X's indicate the field where the value of the parameter should be punched.

II. RUN SPECIFICATION

1. Basic Parameters

a. Fixed Input

At the beginning of each run, three cards must always be present. The first two are identification cards and are printed at the top of the output listing to identify the run. The third card contains the number of tapes to be generated in columns 14 and 15. The illustration shows comments starting in column 31.

```
/ 1                                                          72
<-------------CARD 1 OF RUN IDENTIFICATION------------------->

/ 1                                                          72
<-------------CARD 2 OF RUN IDENTIFICATION------------------->

/ 1      6          14 15       31
  NUMTPS             X  X       NUMBER OF TAPES TO BE GENERATED
```

b. Tape Identification

Each tape may have a tape label of any number of BCD or FIELDATA characters as its first record. The set of characters input may be repeated several times if desired, in order to obtain the desired record length. If no tape label is desired, no tape label cards should be input and zero characters should be requested.

```
/ 1      6            15         20    30
  NUMTID         XXXXX      XXXXX      X
                   |          |        |   {0 - Convert tape ID record to
                   |          |        |__ {    IBM BCD code
                   |          |            {1 - Leave tape ID in FIELDATA
                   |          |__ Repeat basic set of characters this number
                   |              of times
                   |__ Number of characters in tape ID
```

```
/ 1                                                          72
<-----------TAPE ID CHARACTERS------------------------------->
                          .
                          .
                          .
/ 1                                                          72
<-----------TAPE ID CHARACTERS------------------------------->
```

c. Tape Parity

Each tape must be designated as odd or even parity. For situations where it is desirable to have some records with the opposite parity, the probability of a given record being of opposite parity can be input. If even parity is requested with a probability of .25 of having odd parity, approximately 25% of the records will have odd parity. This is accomplished by generating a random number between zero and one and comparing it with the input probability before each record is written on tape. If the random number is less than the input probability, the parity is changed.

```
/ 1     6                      15          30
  TPEPAR                       X          X.XX
                               |           └── probability of a record
                               |               having wrong parity
                               |
                               └── tape parity code (1 = odd, 0 = even)
```

d. Tape Density

Each tape must be designated as 200 or 556 density in the same manner as parity was specified.

```
/ 1     6                      15              30
  TPEDEN                      XXX             X.XX
                               |               └── probability of record
                               |                   with wrong density
                               |
                               └── 200 or 556
```

e. Number of Files

Each tape requires an input which indicates the number of files to be generated on that tape.

```
/1     6            15                    21
NUMFLS              XXX                   NUMBER OF FILES
                     |
                     |____number of files to be generated
```

f. File Identification

Each file may be started with a file label in the same manner as the tape label starts the tape.

```
/1     6           15            20         30
NUMFID           XXXXX         XXXXX        X
                   |             |          |
                   |             |          |___ { 0 - Convert to BCD
                   |             |               { 1 - Leave as FIELDATA
                   |             |
                   |             |__Number of times to repeat
                   |
                   |__ number of characters in File ID
```

```
/1                                                            72
<------------------- FILE LABEL CARDS ------------------------->
```

.
.
.
.

```
/1                                                            72
<------------------- FILE LABEL CARDS ------------------------->
```

2. Data Record Format Definition

Each record is composed of a number of data subsets called frames. Each frame consists of a number of binary bits numbered from the left beginning with 1. The bit position in the frame of a particular quantity will always be used here to mean the bit position in the frame of the leftmost (most significant) bit of the value. For example, if a value is to be inserted into the first 18 bits of the frame we would indicate that it starts at bit position 1 and has a bit length of 18 bits. Any bits not defined as part of the data channels, flags, or time words will be set to zero. The length of each record is determined by the number of frames per record and the number of bits per frame. The total record length cannot exceed 10,000 36-bit computer words (360,000 bits). One card is required for each of these parameters.

For frames per record:

/ 1	6	10	20	31
NUMFPR		5	XXXXX	Number of frames per record

For bits per frame:

/ 1	6	10	20	31
NUMBPF		5	XXXXX	Number of bits per frame

a. Data Channels

Each frame may contain a set of data channels. Each data channel is defined as a set of consecutive binary bits within the frame which will contain either simulated experiment data, a synchronization pattern, a binary counter which controls sub-com synchronization, or information from a subcommutator. Each channel is defined independently, but it is assumed that channel j+1 starts in the first frame bit following channel j. The composition of each frame can be as simple or as sophisticated as desired. The simplest case (ignoring the case of zero channels) consists of one channel per frame with a constant value in the channel throughout all frames.

All channels within a file contain the same number of characters (multiples of 6 bits), the same word sync pattern (if any), the same word parity (if any), and the same number of bits for the data value. Each channel must be six characters or less. The bits to the left of the data word are filled with zeros. K is the number of characters per channel including the zero fill. The possible different channel formats are as follows:

(1) No word parity or word sync (normal case)

```
                   ┌──── bits per data word
                   │
        00000xxxxxxx
frame bit N─┘          └──frame bit N+6(K)-1
```

(2) Word sync and no word parity.

```
                  ┌──── bits per data word
        00yyyxxxxxxxx
frame bit N─┘  │         └── frame bit N+6(K)-1
               └─word sync
```

(3) Word parity and no word sync.

```
                  ┌──── bits per data word
                  │
        ooooxxxxxxxxz
                              parity bit (odd or even)
frame bit N─┘          └──────── frame bit N+6(K)-1
```

(4) Word sync and word parity

```
                  ┌──── bits per data word
                  │
        oyyyxxxxxxxxz
                       ┌───────  parity bit
frame bit N─┘  │                 frame bit N+6(K)-1
               └word sync
```

The data channel format is defined by the cards:

/ 1	6	10	15	16	17	18	19	20	31
NUMDCH		5				X	X	X	Number of Data Channels

```
/1       6     10        15 16 17 18 19 20    31
 NUMBPW        5                     X  X     Number bits per data word
```

```
/1       6     10        15 16 17 18 19 20    31
 NUMCPW        5                        X     Number of Characters per word
```

```
/1       6     10        15 16 17 18 19 20    31
 BTPOFC        5         X  X  X  X  X  X     Frame bit position of first channel
```

```
/1       6     10                       20    31
 WDPARY        5                        X     0,No word parity; 1,odd; 2, even
```

```
/1       6     10              19 20 21     29 30 31
 WDSYNC        2               X  X  X----X  X  Word sync length and pattern
```

If the number of data channels is zero, these cards would not be necessary for the first file of a run. In subsequent files, the first card would be necessary if non-zero in the previous file. The word sync and word parity cards may be omitted when not being used unless changing formats from the previous file. The number of data channels must not be greater than 128.

In most cases, it is desirable to only define the contents of some subset of the available channels. This may be accomplished by providing a card for each channel of interest. The remaining channels will contain all zero bits in the output record except when bit slippage occurs. The following type card is required for each channel defined and each card defines only one channel.

```
/1       6     10              18 19 20     30   31------------------40
 MNEFME        2               X  X  X      Z    Y-------------------YY
```

XXX	= channel number (0 - 127)
Z=1	means that the channel will contain a data curve defined by curve number Y-YY (See section 3 for data curve characteristics.)
Z=2	means that Y - YY is a pattern of bits used for main frame synchronization. (decimal)
Z=3	means that the channel will contain data from subcommutator YY (1-5) (See section 4 for subcommutator definitions)
Z=4	means that the channel will contain the binary count associated with the subcommutator or group of synchronized subcommutators using a binary counter for synchronization. The count will be shifted left in the channel YY places.

b. Time

The length of each file and some of the experiment data values are functions of time. For these purposes, an error free time is assumed beginning with an input start time for each file, incrementing by an input 'milliseconds per frame,' and ending the file when the input stop time is reached.

```
/ 1      6     10          18 19 20    23--------30     40
  SRTTME        1           X  X  X     Y---------Y      Start Time
```

```
/ 1      6     10      13------------20      30
  NUMMPF        5      Y-------------Y       Milliseconds per frame
```

```
/ 1      6     10          18 19 20    23---------30    40
  STPTME        1           X  X  X     Y----------Y     Stop Time
```

XXX = Day of year

Y - Y = Milliseconds of day

The day of year and/or milliseconds of day may be placed in the frame of data by defining the first bit position of the field and the field length.

/ 1	6	10	15-----------20	29 30	40
BPDOYL		1	X X X X X X	Y Y	Day of Year

/ 1	6	10	15-----------20	29 30	40
BPMODL		1	X X X X X X	Y Y	Milliseconds of day

XXXXXX = Bit position in frame of first bit in day of year

YY = Number of bits (0-36)

Even though the time used as arguments for the data curves has no errors, errors may be introduced into the time contained in the frame. Four different types of time errors may be selected, but no more than one will be effective at any given time. These errors are initiated by comparing a random number between zero and one with the input probability of starting that particular type of time error in a given frame. If no time error is in effect, the test is made for the type 1 time error. If the random number is less than the input probability a type 1 time error is initiated. A new random number is then generated and the process is repeated for the three remaining types of time errors. The time quality flags associated with the different time errors are discussed in 2C. An input parameter is permitted which suppresses the incorrect time word from being entered in the frame. This leaves the time word correct throughout all frames but generates the time flags according to the time error sequences generated. The input parameter which governs the suppression of time errors is:

/ 1	6	10	20	
TMESUP		5	X	Time error suppression

$$X = \begin{cases} 0, & \text{generated time errors are entered in frame} \\ 1, & \text{generated time errors give flags only} \end{cases}$$

1) Type 1 Time Error (Time Bias)

The time bias error adds a constant to the time value. This constant is computed by multiplying the maximum bias error which is input by a random number between zero and one. The sign is then determined by generating another random number. If this second random number is less than .5, the bias is set negative. If the bias is negative and greater in magnitude than the correct time value, the result will be set to zero. Each error sequence will last six seconds. The length is controlled by the time flag table which is discussed in section 2.c. The two input values required to define the time bias error are:

/ 1	6	10	20	30
TMERP1		6	X.XXX	Time Bias Probability

/ 1	6	10	14----------20	30
MAXTBS		5	X X X X X X X	Maximum Time Bias Error

2) Type 2 Time Error (Rate Error)

The time rate error is implemented in exactly the same manner as the bias error except that the bias is applied to the milliseconds per frame for six seconds. The input parameters for rate errors are:

/ 1	6	10	20	30
TMERP2		6	X.X X X	Time Rate Error Probability

/ 1	6	10	15 20	30
MAXTRE		5	X X X X X X	Maximum Time Rate Change

3) Type 3 Time Error (Suspect Time)

This type of error introduces a set of time quality flags for a period of six seconds which indicate that the corresponding time sequence is questionable. The corresponding time sequence generated is error free, however. The input for suspect time is:

/ 1	6	10	20	30
TMERP3		6	X. X X X	Suspect Time Probability

4) Type 4 Time Error (Single Time Discontinuity)

The single discontinuity is generated using the maximum time bias but is applied for only one frame. No abnormal time flags are generated. The input of the maximum time bias was discussed unter Type 1 time error.

/ 1	6	10	20	
TMERP4		6	X. X X X	Single Time Error

c. Flags

The flags generated by this program include the hardware generated flags of the STARS I data processor. Several other flags have also been made available including a provision for the user to provide a subroutine to generate the desired flags.

Any number of available flags may be used during a given run. If no flags are desired, all flag definition cards may be omitted. Note that all flags requiring a subcom to be specified may be defined for each subcom. The different types of flags available are:

Type 1 (Data Quality Flag) - This flag is set when the number of errors in the sync pattern exceeds a set level. This flag differs from the flywheel mode flag in that the flywheel flag is set when the sync pattern errors exceed the tolerance set for sync recognition. The data quality flag is independent of sync strategy and therefore requires the input of a data quality level.

/ 1	6	10	20	
DTQLFL		5	XX	Data Quality Level

Type 2 (Number of Sync Errors)- This flag inserts the number of errors in the main frame sync pattern. When the number of errors exceeds the number of bits reserved for the flag, the high order bits of the count are lost.

Type 3 (Fixed Flag) - This flag pattern will occur in all frames except frames filled with dummy or fill patterns.

Type 4 (End of File) - This flag appears in the last frame of the file.

Type 5 (Word Parity) - This flag is set when the number of channels having a word parity error exceeds the input tolerance.

/ 1	6	10	20	31
WPFLGL		5	XXXX	Word Parity Level

Type 6 (Flywheel Mode) - This flag is set when the number of errors in the main frame sync pattern exceeds the allowable number of errors. See section 2.d. for further information on sync strategy.

Type 7 (Subcom Position) - The position of the specified subcom is inserted into the frame.

Type 8 (In subcom Sync) - The indicated flag pattern is inserted into the frame whenever the subcom specified is in sync. See section 2.d for further information on sync strategy.

Type 9 (Out of subcom Sync) - The indicated flag is set when the subcom specified is out of sync. If requested for the same subcom as type 8, this flag would appear in frames not containing flag 8.

Type 10 (Not Used at Present)

Type 11 (Zero Subcom Position) - This flag is inserted in all frames which contain position zero of the subcom specified.

Type 12 (Subcom sync Pattern in Frame) - The flag is inserted when the sync pattern of the subcom specified appears in the channel specified.

Type 13 (Not Used at Present)

Type 14 (Time Quality) - When time quality flags are desired, the normal flag pattern should be enterea. This flag will appear in all frames except when one of the time error sequences is in effect. The flags which occur during an error sequence represent flags which might be generated by the Time Decoder for the corresponding type of time error assuming BCD and SD time are being decoded.

Seconds

-1 0 1 2 3 4 5 6 7

Type 1 time error	513	90	90	257	257	6	6	513
Type 2 time error	513	10	10	216	216	80	80	513
Type 3 time error	513	10	10	16	16	144	144	513
Type 4 time error	513	513	513	513	513	513	513	513

Error Sequence Starts

Error Sequence Stops

The chart shows the decimal value of the flags which occur during the different time error sequences. The portion from 0 seconds to 6 seconds is controlled by the time flag subroutine TMFGS. The flags outside of these error sequences is completely defined by the normal flag pattern which is input. The length of the time flag pattern is input. The flag sequence during the different time error sequences is controlled by DATA statements in subroutine TMFGS. The following tabls shows the bit patterns for the values in the previous tables. The bit positions are obtained by counting from the right and starting with 1.

Seconds

-1 0 1 2 3 4 5 6 7

Type 1 time error	1,10	2,4,5,7	2,4,5,7	1,9	1,9	2,3	2,3	1,10
Type 2 time error	1,10	2,4	2,4	4,5,7,8	4,5,7,8	5,7	5,7	1,10
Type 3 time error	1,10	2,4	2,4	5	5	5,8	5,8	1,10
Type 4 time error	1,10	1,10	1,10	1,10	1,10	1,10	1,10	1,10

Error Sequence Starts

Error Sequence Stops

Type 15 (Supplied by Subroutine) - The position in the frame and number of bits for this are supplied in the same manner as the other flags. When this type of flag is requested, the program calls a subroutine with the statement:

CALL FLGSUB (ITME,IRCDNT,IFMCNT,NERRAC,IFGVAL)

where ITME= Current time in milliseconds

IRCDNT= Record Count

FMCNT= Frame count

NERRAC=Number of errors in sync error pattern

IFGVAL= Place flag value here before returning to main program.

The subroutine supplied by the user must have the following form:

FORTRAN

```
7/8  FOR    FLGSUB,FLGSUB
            SUBROUTINE FLGSUB (ITME,IRCDNT,IFMCNT,
               NERRAC,IFGVAL)
            fortran coding
            RETURN
            END
```

SLEUTH II

```
7/8  ASM    FLGSUB,FLGSUB
FLGSUB*     first Instruction
                        arguments may be addressed by
                          *0,B11     for first
                          *1,B11     for second
                            .           .
                            .           .
                            .           .
                          *4,B11     for fifth
               J           5,B11     EXIT
               END
```

Types 16-20 (Not used at present)

All flags are defined by use of the following card:

```
/ 1       6         10    20        30          40        50 60       70
  FLGDEF            7     XX   XXXXXX      XXXXXX     XXXXX  X   XXXXX
              Flag type ──┘       |           |          |   |     |
               bit position in frame*─┘        |          |   |     |
                          number of bits ─────┘          |   |     |
                                        flag value** ────┘   └─┐   |
                                    subcom number (if required)─┘   |
                                      channel number if type 12─────┘
```

*Position in frame of most significant bit of flag value.

**Flag value is not used for flag types 2, 7 and 15.

d. Sync Strategy

The sync pattern of each frame is considered to be good or bad depending on the number of bits which differ from the actual sync pattern. This division is determined by the input of the maximum allowable errors in the sync pattern.

```
/ 1      6      10        20          31
  ERRALL        5         XXX         Maximum Allowable Errors
```

Main frame lock is dependent on the count of good and bad sync patterns. One counter for the good patterns and a second for the bad patterns. When the good counter reaches a maximum, both counters are reset. When the bad pattern counter reaches a maximum the program assumes that main frame lock is lost and the following occur:

1) The two counters are set to zero.
2) Every remaining character in the record is filled with the dummy data pattern, (Unless suppressed by input parameters)
3) The number of frames to remain out of lock is computed from the number of bits in the sync pattern and the input bit error rate.
4) The time is increased by the number of frames to remain out of lock.
5) If in blanking mode, resume normal mode.

The input, in addition to the maximum allowable sync errors, required to define the main frame sync strategy is:

```
/ 1      6      10        20          31
  TOTSBT        5         XXXX        Total Number Sync Bits
```

```
/ 1      6      10        20          30
  MNSCNS        1         XXXX        XXX
        Maximum for good counter      Maximum for bad counter
```

/ 1	6	10	20	31
DUMPAT		5	XX	Dummy Data Character

/ 1	6	10	20	31
DMDTSP		5	X	Dummy Data Suppress if Non-zero

3. Experiment Data

Two options are available for the generation of data curves. The first is to consider the curves as functions of time. This is the normal mode and is assumed unless overridden by additional input. If it is reasonable to assume that the data word is a function of time, then use the normal mode. Each data curve may be sampled from any channel or any subcom position thus permitting supercommutation within a frame. The data curves may be defined over a time span adequate for the entire run and input for the first file. If no data curve definition cards are input for the second file, the original set are still available. Note however, that if any curve definition cards are input for any file, the previous set is no longer available.

The argument, time, has a range of 0 - 86,400,000 milliseconds. It is considered a circular scale for the purpose of curve definitions. For example, when the time is less than the time indicated on any of the curve definition cards, it is assumed to be in the range of the curve definition card with the largest value of time.

All experiment curves with time as argument are defined by the following input:

/ 1	6	10	20	30	40
CURDEF		4	XXXX	X	XXXXXXXX

Curve number — XXXX

Curve type — X

Beginning time c_i — XXXXXXXX

(card continues on next page)

```
        50          60          70            80
[ XXXXXXXXXX    XXXXXX      XXXXXXXX      XXXXXXXX
[Beginning value x_i
   Slope or amplitude*
                        period p_i
                             standard deviation*
```

*Fractional values are permitted by punching decimal in card.

If it is not reasonable to assume that the experiment is a function of time, it is possible to replace time as the argument with a counter which is incremented each time the curve is sampled. This approach might be beneficial when simulating decom tapes. The definition of these curves is exactly the same as when using time as the argument except that the curve numbers must be in the range 1-50. The experiment curve definition card is repeated here to show the count as the argument.

```
/ 1      6       10          20          30           40
  CURDEF          4          XX           X   XXXXXXXXXXX
             curve number
                        curve type
                            starting counter C_i
```

```
          50            60           70            80
XXXXXXXXXXX   XXXXXXXXXXX  XXXXXXXXXXX  XXXXXXXXXXX
beginning value X_i
  slope or amplitude*
                      period p_i
                           standard deviation*
```

*Fractional values are permitted by punching decimal in card.

Since the curve definition cards look exactly the same for curves with time as the argument as they do for curves with a counter as the argument, it is necessary to provide a means for making this distinction. For the normal mode, no additional input is required. For each curve using a counter type argument, the following card is required.

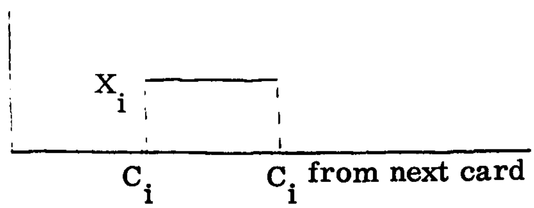
```
/1     6      10          20              30          40
 SPLCUR       2           XX         XXXXXXXX      XXXXXX
                          |                 |           |
         curve number (1-50)                |           |
                   initial counter setting--┘           |
                             increment for counter------┘
```

Different phases of the periodic functions can be obtained by starting the curve definition at an earlier start time than the start of the file.

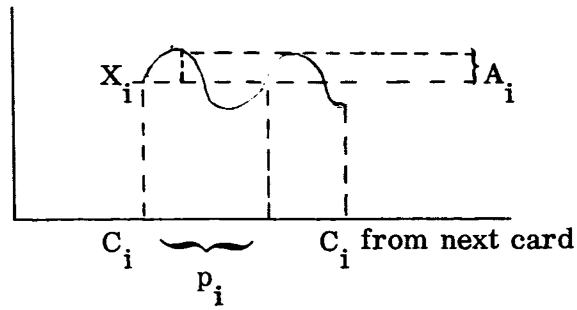
Curve numbers may range between 1 and 2000 for curves with time arguments. Option 2 curves must have curve numbers between 1 and 50. Curve numbers need not be used sequentially. As many cards as needed may be used to define a particular curve. The total number of curve definition cards must not exceed 4000, however. The program sorts the cards by curve number then by time or count within the curve. Once the time or count reaches the value on a particular card, that card defines the curve until the time or count reaches the value on the next card. The column labeled 'standard deviation' will be discussed under 5.a.

The different curve types are:

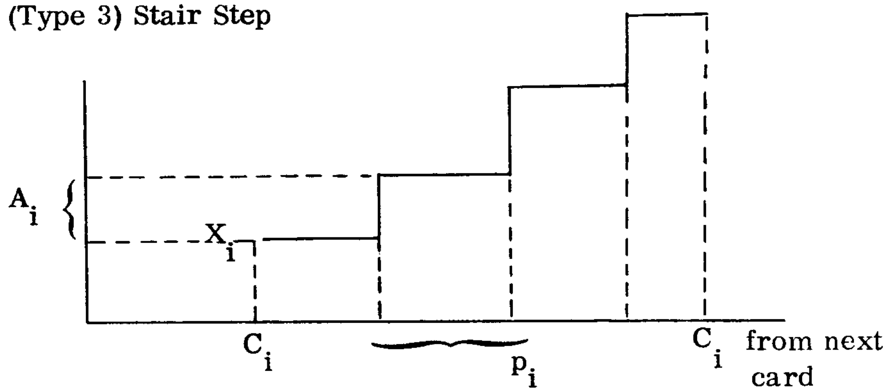
a. (Type 1) Constant



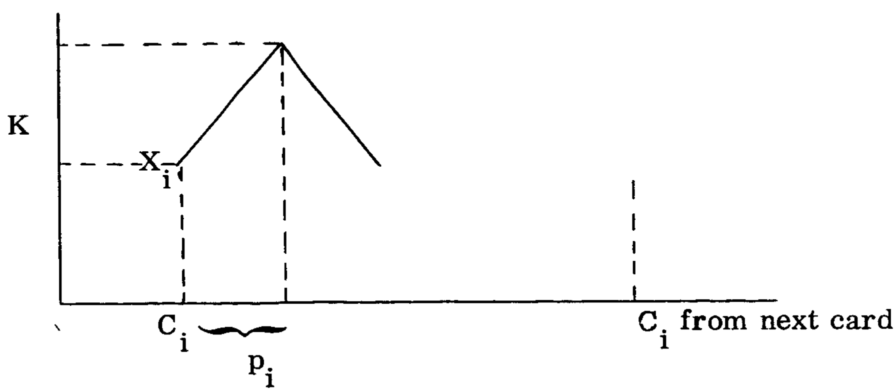
b. (Type 2) Sine Wave



c. (Type 3) Stair Step

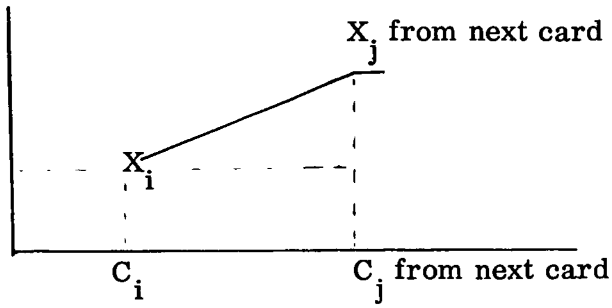


d. (Type 4) Triangle



$$A_i = \frac{K}{P_i} = \text{slope}$$

e. (Type 5) Table Look-up



If next card of same curve number is also type 5, the interpolated value between the two X's will be given as shown. If the next card is of different type or different curve number, the value X_i will result. Note that next card means the next card after the cards have been placed in order by curve number and time (or count).

f. (Type 6) User Subroutine

When a type 6 curve is requested, the program gives control to USRSUB by the statement:

CALL USRSUB (ITME,IRCD,IFMCNT,ICHCNT,NVAL,K)

where

ITME=	time in milliseconds
IRCD=	record number
IFMCNT=	frame number within record
ICHLNT=	channel number
NVAL=	place value of curve here
K=	curve number

Note that K makes it possible to define more than one curve as provided by user subroutine.

The subroutine must have the following form:

FORTRAN

```
7/8  FOR USRSUB,USRSUB

     SUBROUTINE USRSUB (ITME,IRCD,IFMENT,ICHCNT,NVAL,K)

        Fortran coding

     RETURN

     END
```

SLEUTH II

```
7/8  USRSUB,USRSUB

USRSUB*  First instruction
                        Arguments may be addressed by
                             *0,B11     for first
                             *0,B11     for second
                               .           .
                               .           .
                               .           .
                             *5,B11     for sixth
          J    6,B11         EXIT
          END
```

4. Subcommutator (Subcom for short)

a. Defining Subcoms

Five subcoms are available. Each subcom may contain not more than 128 positions. Whenever a data channel in the main frame requests data from a subcom, the subcom checks for its current position and tests to see whether that position should be data or a sync pattern. If data, the curve number is noted, the position of the subcom is increased by 1 (modulo the length of the subcom) and the curve value is then computed for the channel in the same manner as if the channel had originally requested a data curve. If a sync pattern is requested, that pattern is placed in the data channel and the subcom position is increased by 1. (Modulo the length of the subcom).

A subcom is defined by the following card:

```
/ 1      6        10       20        30        40        50
  NUMSUB          3        X         XXX       XXX       X
     Subcom number (1-5)---┘         |         |         |
          Total number positions(1-128)┘       |         |
             Starting position (0-127)---------┘         |
       (If greater than 127, then the starting position is
       computed at random in the range 0-127.)           |
                                        Subcom type------┘
```

The positions of a subcom are defined by:

```
/ 1      6        10       20       30        40              50
  SUBCOM          3        X        XXX       X          XXXXXX
    subcom number----------┘        |         |            |
         subcom position(0-127)-----┘         |            |
             1 if curve, 2 if sync------------┘            |
                   curve number or sync pattern------------┘
                                ( in decimal)
```

The subcom type refers to the method of synchronization used. Type 1 means that if a fixed sync pattern is defined as one of the subcom positions that it will be used as part of the subcom sync strategy. Type 2 means that the subcom is synchronized with a binary counter defined in one of the main frame data channels.

The binary counter is then used to determine if the subcom is in sync. Only one main frame channel may be designated as a binary counter but several of the subcoms may use it for sync strategy. All subcoms which use the binary counter should be designated as synchronized subcoms if slippage is possible. (See section 5.c.) This is done by:

/ 1 6	10	20	30	40	50	60
SYNSUB	9	X	X	X	X	X
		1	2	3	4	5 (Subcoms represented)

non zero values in two or
more positions synchronize the corresponding
subcoms with each other.

Type 1 subcoms may also be synchronized with each other as described above.

As an example, a 1 in column 20 and a 1 in column 60 would synchronize subcom 1 and 5.

To synchronize the output tape records to one of the subcoms, use the card:

/ 1 6	10	20	31
SUBSYN	5	X	Synchronize records to Subcom X

When this card is used and the subcom specified complete its cycle, the remaining frames in the record are completed with the fill data pattern in each character. This pattern will be six binary zeros if no pattern is entered. Another pattern may be entered by the card:

/ 1 6	10	20	31
FILPAT	5	XX	Fill data pattern

b. Subcom Sync Strategy

1. Binary counter - all type 2 subcoms are synchronized by a single binary counter which appears in one of the data channels. Like the main frame sync, subcom lock is maintained by counting good and bad values in

the binary counter. A good or bad value is determined by comparing the binary count against a value which simulates a STARS I hardware counter. This hardware counter is increased by one each frame and is reset only when subcom lock is lost. The binary count is considered to be good when the two counters agree and bad when they do not agree. In the frame after lock is lost, the hardware counter is set equal to the value of the binary counter and the subcom is again considered in lock. The following cards set the maximum values for the good and bad counters for controlling subcom lock.

For good values:

/ 1	6	10	20	30	40	50	60
SBGCNT		9	XX	XX	XX	XX	XX

For bad values:

/ 1	6	10	20	30	40	50	60
SBBCNT		9	XX	XX	XX	XX	XX

Values pertaining to subcom 1 go in columns 19 and 20, for subcom 2 in 29 and 30, etc.

Note: The STARS I system usually is set to require seven consecutive bad values to go out of lock. The input to simulate this setting would be seven bad values before one good value.

2. Sync pattern - Subcom lock is controlled by a position designated as a sync pattern on Type 1 subcoms. Everytime that the sync pattern from the subcom is placed in the main frame of data, it is counted as a good or bad sync pattern according to the number of bits in error. The maximum allowable errors in the subcom sync pattern is entered by the card:

/ 1	6	10	20	30	40	50	60
MALESB		9	XX	XX	XX	XX	XX

Subcom lock is controlled by the input of two values. One is a maximum for the count of bad values and one is a maximum for the count of good patterns. Subcom lock is lost when the bad counter reaches a maximum. The subcom stays out of lock until the next good sync pattern is found. Both counters are reset when the good counter reaches a maximum. See binary counter subcoms for the input of good and bad counter maximums.

5. Perturbations

Many options are available for the introduction of errors. In addition to the time errors discussed in section 2.b., efforts have been made to produce the types of errors which actually occur in PCM telemetry data.

a. Experiment noise - the standard deviation, entered on each curve definition card, controls the amount of experiment noise to be added to the curve defined by that card. After the value of the curve is computed, a random number between zero and one is generated. This random number and the standard deviation, along with the computed curve value are used to recompute the data value. If a is the value of the curve and p is the random number the value with noise is computed using linear interpolation from the table:

i	p_i	f_i
1	.000	$a-5\sigma$
2	.001	$a-3\sigma$
3	.023	$a-2\sigma$
4	.158	$a-\sigma$
5	.500	a
6	.841	$a+\sigma$
7	.977	$a+2\sigma$

8	.999	a+3σ
9	1.000	a+5σ

b. Transmission Noise

Transmission noise is simulated by the addition of bit errors, (i.e., the changing of "1" bits to "0" and "0" bits to "1"). This is accomplished by the input of the probability of a given bit being in error.

/ 1	6	10	20	31
BTERRT		6	X.XXX	Bit error rate

Signal blanking is simulated by replacing the bit error rate by .5. This is controlled by an input blanking percentage. This indicates the percentage of frames to be blanked.

/ 1	6	10	20	31
BLKPCG		6	X.XXX	Blanking percentage

The length of each blanking sequence is computed at random between 5 and 20 frames. The program returns to the input bit error rate upon reaching the end of the blanking sequence or upon losing main frame lock.

c. Equipment Errors

1. Bit Slippage - the number of bits in a channel is increased or decreased. The probability of this occurring within a given channel is input.

/ 1	6	10	20	31
SLPPCT		6	X.XXX	Bit slippage probability

A random number between zero and one is generated corresponding to each channel and compared with the probability of bit slippage within that channel. Whenever the random number is less, bit slippage is selected randomly in the range (-6, 6). If zero is selected, a new selection is made. This slippage is added to any slippage already in effect. All slippage is removed whenever main frame sync goes out of lock.

2. Subcom Slippage - The probability of a subcom incrementing an incorrect number of positions can be input. Each time the subcom is sampled, a random number is generated and compared with the input probability. When the random number is less than the probability, the slippage increment is added to the subcom position counter. This increment is in addition to the normal increment of one position. The amount of slippage allowed is -1 or any positive value not greater than the total positions on the subcom. Each of the five subcoms may have its own slippage probability and amount. Any slippage occurring in one subcom will be added to all others which have been defined as being synchronized with it.

The input cards required to define subcom slippage are:

```
/ 1     6        10              20        30       40       50      60
  SUBPCT         9           X.XXX     X.XXX    X.XXX    X.XXX   X.XXX

  probability of slippage for subcoms 1-5
```

```
/ 1     6        10      20       30        40      50        60
  SYBSLP         8      XXX      XXX       XXX     XXX       XXX

amount of slippage to apply for subcoms 1-5
```

3. Incorrect Record Length - The probability of a given record being of incorrect length is input.

/ 1	6	10	20	31
SLRCDP		6	X.XXX	Short or Long Record Probability

Again, a random number is generated before each record is written. If the random number is less than the input 'short or long record probability,' the length of the record is chosen randomly between zero words and twice the normal record length.

d. Operator Error

Two types of errors are included in this category. One is the omission of an end-of-file indicator at the end of a data file. The second is the insertion of an extra end-of-file indicator at the end of a data file.

/ 1	6	10	20	30
ENDFLP		7	X.XXX	X.XXX

probability of an extra end of file

probability of an end of file being omitted

This card may prove useful for changing data formats within a file. This would be accomplished by indicating to the program that two files are being generated and setting the probability of omitting the end-of-file indicator to 1.0. The data on the tape will then appear as one file if the number of characters in the second file ID record is set to zero.

6. Code Conversion

The tape and file label records may be specified as UNIVAC 1107 FIELDATA code or IBM BCD code. This selection is made by the appropriate parameter on the card which specifies the number of characters in the particular label. The format was given in section II. 1.

III. OUTPUT

1. Tape Format

The tape format will, in general, follow the illustration on page IV-1. This can be modified through the use of the input cards already described. For example, the tape label or any of the file labels may be omitted by entering the number of characters in the label as zero. Both or either of the time words in the frame may be omitted by setting the corresponding field length to zero. If flag definition cards are not entered, then no flags would appear in the frame. An end-of-file indicator may be omitted by setting the probability of omitting an end-of-file to 1.0 for the file in question.

2. Printed Output

The input parameters previously described are all printed with identifying information. This is followed by channel printouts and record printouts if requested. Up to fifty different channels may be printed as the tape is being generated. Each channel printed requires one line of print for each frame. The format of the line is:

(1) record number
(2) frame number within record
(3) channel number
(4) time of day in ms.
(5) curve value without noise (disregard if not curve)
(6) curve value with experiment noise
(7) final curve value with all noise added
(8)
(9) Same as 5, 6, and 7 except printed in octal.
(10)

The data channels to be printed are selected by the following cards, (up to 50).

1	6	10	20	
OUTPO1		5	XXX	Print Channel XXX

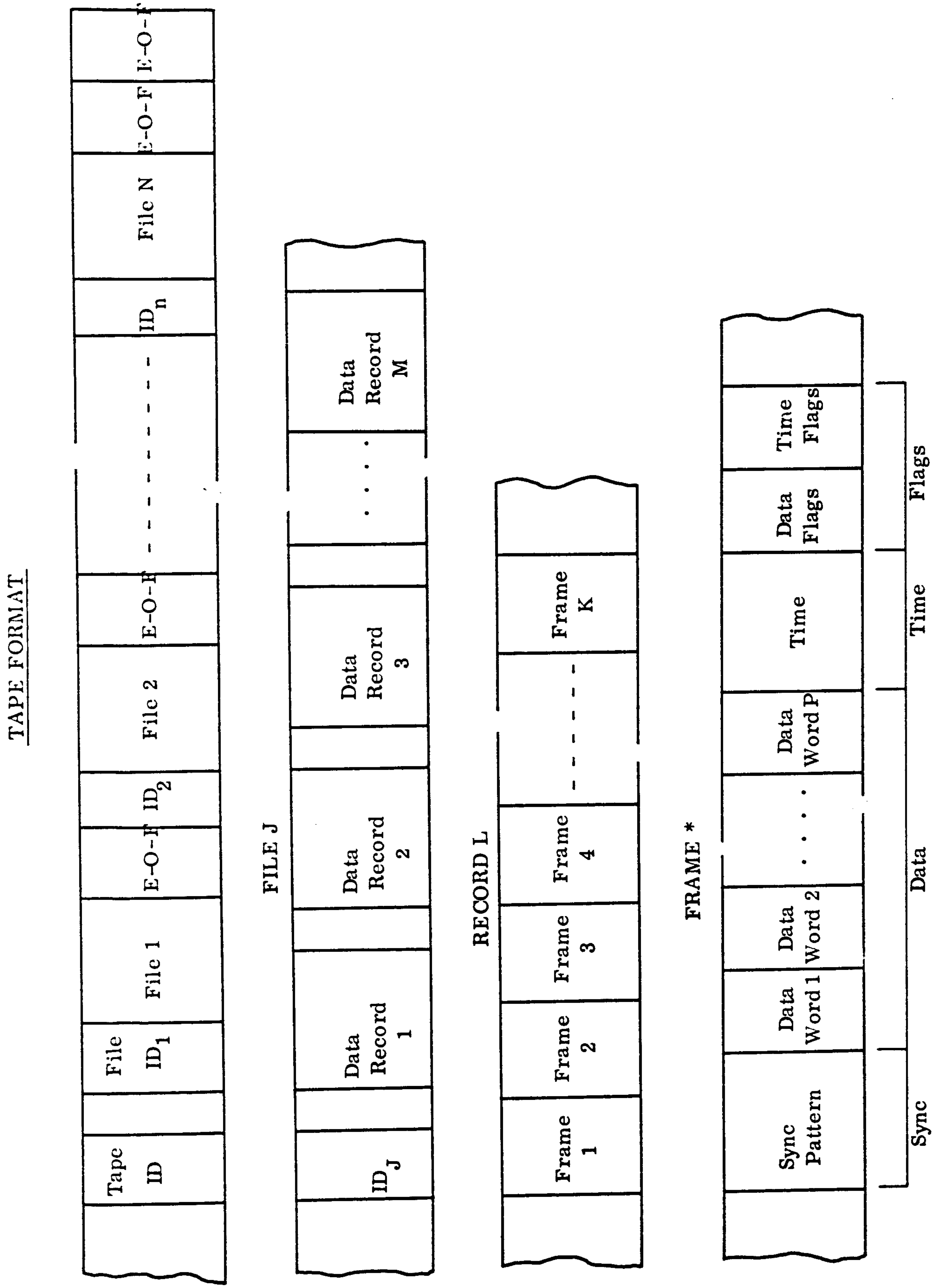
Every Nth record may be printed in octal by the card:

1	6	10	20	
OUTPO2		5	XXX	Print every XXX record

Since the printout becomes too great on long runs, it seems advisable to check the input by first setting up a short run. The printout can then be reduced before the final run is made.

A table showing the distribution of error bits among the different channels is printed at the end of each file.

IV. TAPE FORMAT



*The order of the frame items may be varied by input.

IV. DECK SETUP

———	Run card	
———	ASG card	
———	XQT CUR	Input program from tape
———	TRW C	Tape Y-1473
———	IN C	
———	TRI C	
- - - - -	User subroutines (optional)	
———	XQT TPGEN	
———	Data Cards (see next page	
———	FIN card	

The control cards listed above are standard for the UNIVAC 1107 EXEC II monitor but a brief description of each will be given to eliminate the need of obtaining the referenced manual.

```
/ 1      4       9
  7/8    RUN     NAME,ACC,TIME, PAGES
```

numeric values indicating maximum time and maximum pages

```
/ 1      4       9
  7      ASG     C=TGPM, D=GEN1, E=GEN2
  8
```

The ASG card must correspond to the tape set-up card which is submitted to the dispatcher with the run. This illustration shows two tapes being generated.

TAPE SET-UP

Job________ Code_________ Sponsor____________________

TAPE ASSIGNMENT LEVEL			
ASG Label	Reel Label and Order	PFR	Channel
C=TGPM	Y-1473	F	
D=GEN1	BLANK	F	
E=GEN2	BLANK	F	

CHANNEL REQUEST LEVEL
Unit

/ 1	4	9
7/8	XQT	CUR

/	4	9
	TRW	C

/	4	9
	IN	C

/	4	9
	TRI	C

/ 1	4	9
7/8	XQT	TPGEN

/ 1	3
7/8	FIN

III. DATA SETUP

Run identification

Number of tapes

Number of characters in tape ID

Characters for tape ID

Tape parity

Tape density

Number of files

Number of characters in file ID

Characters for file ID

File data for file 1 *

Blank card

File data for file 2 *

Blank card

⋮

File data for last file of tape *

Blank card

Tape data

* Order not important within these decks

APPENDIX

SUMMARY OF INPUT

All input is summarized in this section for quick reference. R means that the cards are required. O means optional.

Run Identification Cards (2 cards)

/ 1 … 72	← Run identification information →	R
/ 1 … 72	← Run identification information →	R

Number of Tapes

/ 1	6	15	
NUMTPS		XX	R

Tape Label

/ 1	6	15	20	30	
NUMTID		No. Char.	Repeat	0 if BCD	R

/1 … 72	← Tape Label Characters →	O
⋮	⋮	⋮
/1 … 72	← Tape label characters →	O

Tape Parity

/ 1	6	15	20	
TPEPAR		Parity code	Error Prob.	R

Number of Files

/ 1	6	15	
NUMFLS		No. Files	R

File Label

/ 1	6	15	20	30	
NUMFID		No. char.	Repeat	O if BCD	R

/ 1	72	
← File label characters →		O

⋮

/ 1	72	
← File label characters →		O

Frames per Record

/ 1	6	10	20	
NUMFPR		5	No. of Frames	R

Bits per Frame

/ 1	6	10	20	
NUMBPF		5	No. of bits	R

Number of Data Channels

/ 1	6	10	20	
NUMDCH		5	No. of Channels	R

Bits per Word

/ 1	6	10	20	
NUMBPW		5	No. of Bits	R

Characters per Word

/ 1	6	10	20	
NUMCPW		5	No. of Char.	R

Bit Position of First Channel

/1	6	10	20	
BTPOFC		5	Bit pos.	R

Word Parity

/1	6	10	20	
WDPARY		5	Code	O

Word Syne

/1	6	10	20	30	
WDSYNC		2	No. of bits	Pattern	O

Main Frame Definitions

/1	6	10	20	30	40	
MNEFME		2	Channel	Curve or sync or subcom	Pattern or number	O

Start Time

/1	6	10	20	30	
SRTTME		1	Day	Ms	R

Number of Milliseconds per Frame

/1	6	10	20	
NUMMPF		5	Ms	R

Stop Time

/1	6	10	20	30	
STPTME		1	Day	Ms	R

Day Position in Frame and Number of Bits

/1	6	10	20	30	
BPDOYL		1	Bit position	No. of bits	O

Ms Position in Frame and Number of Bits

/ 1	6	10	20	30	
BPMODL		1	Bit position	No. of bits	O

Time Error Suppression

/ 1	6	10	20	
TMESUP		5	Code	O

Time Bias Probability

/ 1	6	10	20	
TMERP1		6	Prob.	O

Time Rate Error Probability

/ 1	6	10	20	
TMERP2		6	Prob.	O

Time Bias Maximum

/ 1	6	10	20	
MAXTBS		5	Ms	O

Time Rate Error Maximum

/ 1	6	10	20	
MAXTRC		5	Ms	O

Suspect Time Probability

/ 1	6	10	20	
TMERP3		6	Prob.	O

Single Time Error Probability

/ 1	6	10	20	
TMERP4		6	Prob.	O

Data Quality Flag Level

/ 1	6	10	20	
DTQLFL		5	Level	O

Word Parity Flag Level

/ 1	6	10	20	
WPFLGL		5	Level	O

Flag Definitions

/ 1	6	10	20	30	40	50	60	70	
FLGDEF		7	Type	Pos.	Length	Value	Subcom	Channel	O

Sync Strategy Error Tolerance

/ 1	6	10	20	
ERRALL		5	Max. Allowable Errors	O

Total Number of Main Frame Sync Bits

/ 1	6	10	20	
TOTSBT		5	No. of Bits	O

Sync Counter Maximum

/ 1	6	10	20	30	
MNSCNS		1	Good	Bad	O

Dummy Data Character (in decimal)

/ 1	6	10	20	
DUMPAT		5	Value	O

Dummy Data Suppression Code

/ 1	6	10	20	
DMDTSP		5	Code	O

Curve Definition

/ 1	6	10	20	30	40	50	60	70	80	
CURDEF		4	Number	Type	Time	Value	Amp.	Per.	σ	O

Special Curve Definitions

/ 1	6	10	20	30	40	
SPLCUR		2	Number	Counter	Increment	O

Subcom Definition

/ 1	6	10	20	30	40	50	
NUMSUB		3	No.	No. of positions	Starting Position	Type	O

Subcom Position Definition

/ 1	6	10	20	30	40	50	
SUBCOM		3	Subcom No.	Position	Curve or Sync	Number or Pattern	O

Synchronize Subcoms

/ 1	6	10	20	30	40	50	60	
SYNSUB		9	1 or 0	1 or 0	1 or 0	1 or 0	1 or 0	O

Synchronize Records To Subcom

/ 1	6	10	20	
SUBSYN		5	Subcom No.	O

Fill Data Pattern (1 character in decimal)

/ 1	6	10	20	
FILPAT		5	Value	O

Subcom Lock Good Counter Maximums

/ 1	6	10	20	30	40	50	60	
SBGCNT		9	Subcom 1	Subcom 2	Subcom 3	Subcom 4	Subcom 5	O

Subcom Lock Bad Counter Maximum

/ 1	6	10	20	30	40	50	60	
SBBCNT		9	Subcom 1	Subcom 2	Subcom 3	Subcom 4	Subcom 5	O

Maximum Allowable Errors in Subcom Sync Patterns

/ 1	2	10	20	30	40	50	60	
MALESB		9	Subcom 1	Subcom 2	Subcom 3	Subcom 4	Subcom 5	O

Transmission Noise (Bit error Probability)

/ 1	6	10	20	
BTERRT		6	Prob.	O

Blanking Percentage

/ 1	6	10	20	
BLKPCG		6	Pct.	O

Bit Slippage (probability of slippage within a channel)

/ 1	6	10	20	
SLPPCT		6	Prob.	O

Subcom Slippage Probability

/ 1	6	10	20	30	40	50	60	
SUBPCT		9	Subcom 1	Subcom 2	Subcom 3	Subcom 4	Subcom 5	O

Amount of Subcom Slippage

/ 1	6	10	20	30	40	50	60	
SUBSLP		8	Subcom 1	Subcom 2	Subcom 3	Subcom 4	Subcom 5	O

Probability of Incorrect Record Length

/ 1	6	10	20		
SLRCDP		6	Prob.		O

Probability of Missing or Adding an End-of-File

/ 1	6	10	20	30	
ENDFLP		7	Miss. Prob.	Add. Prob.	O

Output Option 1

/				
OUTPO1	5	XXX	Print Channel XXX	O

Output Option 2

/				
OUPTO2	5	XXX	Print every XXX record	O

SUMMARY OF LIMITATIONS

Item	Limits
Run identification	144 characters
Tapes in one run	less than 21 tapes (use with D, E, F . . . .)
Tape ID record	less than 10000 bits
Tape parity code	1=odd, 0=even
Tape density	200 or 556
Files per tape	length of tape
File ID record	less than 10000 bits
Frames per record	10000/bits per frame
Bits per frame	must be less than 10000
Data channels	not more than 128
Bits per data word	less than 36
Characters per channel	not more than 6
Word parity code	0= omit, 1= odd, 2= even
Word sync length	less than 36 bits
Channel number	0 - 127
Flag value	less than 36 bits
Curve number (time argument)	1 - 2000
Curve number (count argument)	1 - 50
Subcom number	1 - 5
Subcom position	0 - 127

SPECIAL CASES

In certain situations, the flexibilities of the program can be used to accomplish results which are not directly obtainable. The two examples which follow illustrate the many possibilities.

Problem 1.

How can the tape record format be changed within a file?

Solution:

Generate two files on the tape with no end-of-file between them and omit the file ID on the second file. This is accomplished by ending the first file where the format change is to take place. The file data for the first file must contain a card setting the probability of missing an end-of-file to 1.00. The number of characters in the file ID of the second file must be set to zero.

Problem 2.

How can different channels be defined in alternate frames? For example, define channel 5, 8, 10 and 12 in odd numbered frames and channels 7, 13 and 20 in even numbered frames.

Solution:

Define one of the subcommutators, say subcom 4, with fourteen positions and starting at position 0. Indicate that data for channels 5, 7, 8, 10, 12, 13 and 20 will be obtained from subcom 4. Now subcom 4 will cycle every two frames so the first seven positions are defined as the data desired in 5, 7, 8, 10, 12, 13 and 20 of the odd frames. Thus positions 0, 2, 3, 4 are defined in the first seven and 8, 12, and 13 are defined in the second seven positions of subcom 4 to supply data to channels 7, 13 and 20 of the even numbered frames.

TEST #1 - AE-B

Buffer tape label at beginning of tape Buffer file label before each file. Single end-of-file after each file. Double end-of-file at end of tape. Subcom count in Channel 3 - 5 bits:

0000	Count

Subcom length 32/1 (one synchronized). Main channels numbered 0-47. 27 bit sync pattern: $(765\ 514\ 600)_8$ ΔT for frame = 50 milliseconds. Time mode: BCD & SD. Density digital tape: 556 bpi

Frame sync:

- Tolerable errors = 3 bits
- Fast search counter = 2
- Lock counter = 3
- Slow search = 7

Subframe sync: binary count bad patterns in sequence = 7.

9 bits/word

2 characters/telemetry word

bit error probability = 10^{-3}

blanking 3%

Buffer tape label: (BCD)

Channel	
1-5	Satellite ID
7	Processing Line Number
9	Operator Number
11-13	Day count of Processing
15-18	Buffer tape Number

Buffer File Label: (BCD)

1-3	Station Number
5-8	Analog Tape Number
10-11	Analog File Number
13-15	Day Count of Data
17-18	Buffer File Number

Each 18 character label is repeated 192 times to form a 576 (36 bit) word record.

36 bits

1	Flag*	Time Quality	Day Count
2	Milliseconds of Day		
3	Pressure Cal^{O}	1	2
4	SUB Com Count 3	4	5
5	6	7	8
6	NRC-1 9	GCA 10	NRC-2DC 11
7	NRC-2AC 12	13	14
8	15	16	17
9	18	19	20
10	Subcom 1 21	Subcom 2 22	23
11	24	25	26
12	O REF 27	28	S REF 29
13	30	31	32
14	NRC - 1 33	GCA 34	NRC-2DC 35
15	NRC-2AC 36	37	38
16	39	40	41
17	42	43	44
18	Frame Sync 45	Frame Sync 46	Frame Sync 47

12 bits

*	S	SC Count	S	SC Count

S = flag for SC recycle

SC word = sub com word

Use user subroutine to repeat

MAIN FRAME DATA

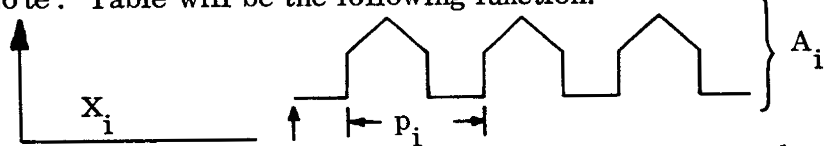
	σ	Count
NRC-1	5	$X_i = 100$, $A_i = 100$, $p_i = 1000$ ms
GCA	6	$X_i = 100$, $A_i = 100$, $p_i = 1000$
NRC-2DC	7	"
NRC-2AC	8	"
O REF	2	$(82)_{10}$ constant
S REF	2	$(475)_{10}$ constant
Pressure Cal	0	→7 - 6 - 5 - 3 - 7 →
		begin 1 second end

SUB COM 1

FRAME 0	0	$(007)_8$ constant
FRAME 16	0	$(00\ 46)_8$ first file
FRAME 19	0	constant $X_i = 256$
FRAME 20	0	sine wave $X_i = 100$, $A_i = 10$, $p_i = 2000$
FRAME 21	0	sine wave $X_i = 1$, $A_i = 5$, $p_i = 1000$
FRAME 22	0	triangle $X_i = 100$, $B_i = 10^{-2}$, $p_i = 2000$
FRAME 23	0	table* $X_i = 100$, $A_i = 20$, $p_i = 1000$
FRAME 24	1	sine wave $X_i = 100$, $A_i = 10$, $p_i = 2000$
FRAME 25	1	$(400)_{10}$ constant
FRAME 26	1	sine wave $X_i = 100$, $A_i = 5$, $p_i = 1000$
FRAME 27	1	triangle $X_i = 100$, $B_i = 10^{-2}$, $p_i = 2000$
FRAME 28	1	table* $X_i = 100$, $A_i = 20$, $p_i = 1$---
FRAME 17	1	$(400)_{10}$ constant
FRAME 19	2	Constant: $X_i = 256$, $X_i = 100$
FRAME 20	2	Sine wave: $X_i = 100$, $A_i = 10$, $p_i = 2000$
FRAME 21	2	Sine wave: $X_i = 100$, $A_i = 5$, $p_i = 1000$
FRAME 22	2	Triangle: $X_i = 100$, $B_i = 10^{-2}$, $p_i = 2000$

	σ	Count
FRAME 23	2	Table*: $X_i = 100$, $A_i = 20$, $p_i = 1000$
FRAME 24	3	Sine wave: $X_i = 100$, $A_i = 10$, $p_i = 2000$
FRAME 25	3	Sine wave: $X_i = 100$, $A_i = 5$, $p_i = 1000$
FRAME 26	3	Triangle: $X_i = 100$, $B_i = 10^{-2}$, $p_i = 2000$
FRAME 27	3	Table*: $X_i = 100$, $A_i = 20$, $p_i = 1000$
FRAME 28	4	Sine wave: $X_i = 100$, $A_i = 10$, $p_i = 2000$
FRAME 29	4	Sine wave: $X_i = 100$, $A_i = 5$, $p_i = 1000$
FRAME 7	0	Sine wave: $X_i = 300$, $A_i = 300$, $p_i = 1000$

*Note: Table will be the following function:



Enter as a table in the first case. Compute with user subroutine in the other cases.

Number of types = 1

Tape parity: odd

Percentage of records which denote from parity: 1%

Tape density: 556

Percentage of records which should very from this density: 1%

Number of files = 1

Number of records in file: 150 (4 numbers)

Word parity: none

Word sync: none

Slippage percentage: 1/50,000 bits

AE-B TEST CASE INPUT

```
  !   RUN    VN,C08,20,300
  !   ASG    C=TGPM,D=GEN
  !   XQT    CUR
      TRW    C
      IN    C
      TRW    C

  !   XQT  TPGEN
                        TAPE GENERATION FOR AE-B
                             SEPT. 1965
 NUMTPS         1            NUMBER OF TAPE
 NUMTID        18   191                        TAPE ID CHAR. COUNT AND REPEAT
 999990102025000046                       TAPE ID
 TPEPAR         1            .01               TAPE PARITY
 TPEDEN       556            .01               TAPE DENSITY
 NUMFLS         1                              NUMBER OF FILES
 NUMFID        18   191                        FILE ID CHAR. COUNT AND REPEAT
 020000100010243001                             FILE ID
 NUMBPF    5          648                      NUMBER BITS PER FRAME
 NUMFPR    5           32                      NUMBER FRAMES PER RECORD
 NUMDCH    5           48                      DATA CHANNELS
 NUMCPW    5            2                      CHARACTERS PER WORD
 NUMBPW    5            9                      BITS PER WORD
 SRTTME    1          243   86000000           START TIME
 NUMMPF    5           50                      MILLISECONDS PER FRAME
 STPTME    1          243   86240000           STOP TIME
 BPDOYL    1           25         12           DAY OF YEAR POSITION AND LENGTH
 BPMODL    1           37         36           MS OF DAY POSITION AND LENGTH
 TOTSBT    5           27                      NUMBER OF SYNC BITS
 NUMSUB    3            2         32           0         2           DEFINE SUBCOM  2
 NUMSUB    3            1         32           0         2           DEFINE SUBCOM  1
 BTPOFC    5           73                      BIT POSITION OF FIRST CHANNEL
 BTERRT    6       .001                        BIT ERROR RATE
 DUMPAT    5           16                      DUMMY DATA PATTERN
 BLKPCG    6       .03                         BLANKING PERCENTAGE
 ERRALL    5            3                      ERRORS ALLOWABLE
 SYNSUB    9            1          2                                    SYNCHRONIZE SUBCOMS
 MNSCNS    1            3          7           SYNC STRATEGY COUNTERS
 SUBSYN    5            1                      SYNCHRONIZE TO SUBCOM 2
 SBBCNT    9            7          7           0         0          0   SUBCOM SYNC COUNT
 SBGCNT    9            1          1           0         0          0
 MNEFMF    2           45          2         501         DEFINE SYNC PATTERN
 MNEFME    2           46          2         332
 MNEFMF    2           47          2         384
 MNEFME    2            0          1           1         CHANNEL 0    PRESSURE CAL
 MNEFMF    2            3          4           0                      BINARY COUNTER
 MNEFME    2            9          1           2         CHANNEL  9   NRC-1
 MNEFMF    2           10          1           3         CHANNEL 10   GCA
 MNEFME    2           11          1           4         CHANNEL 11   NRC-2DC
```

MNEFME	2	12	1	5	CHANNEL 12 NRC-2AC			
MNEFME	2	21	3	1	CHANNEL 21 SUBCOM			
MNEFME	2	22	3	2	CHANNEL 22 SUBCOM			
MNEFME	2	27	1	6	CHANNEL 27 0-REFERANCE			
MNEFME	2	29	1	7	CHANNEL 29 5-REFERANCE			
MNEFME	2	33	1	2	CHANNEL 33 NRC-1			
MNEFME	2	34	1	3	CHANNEL 34 GCA			
MNEFME	2	35	1	4	CHANNEL 35 NRC-2DC			
MNEFME	2	36	1	5	CHANNEL 36 NRC-2AC			
CURDEF	4	1	1	86000000	7			
CURDEF	4	1	1	86001000	6			
CURDEF	4	1	1	86002000	5			
CURDEF	4	1	1	86003000	3			
CURDEF	4	1	1	86004000	7			
CURDEF	4	2	2	86000000	100	100	1000	5.0
CURDEF	4	3	2	86000000	100	100	1000	6.0
CURDEF	4	4	2	86000000	100	100	1000	7.0
CURDEF	4	5	2	86000000	100	100	1000	8.0
CURDEF	4	6	1	86000000	82			2.0
CURDEF	4	7	1	86000000	475			2.0
CURDEF	4	1	1	86300000	7			
CURDEF	4	1	1	86301000	6			
CURDEF	4	1	1	86302000	5			
CURDEF	4	1	1	86303000	3			
CURDEF	4	1	1	86304000	7			
CURDEF	4	8	1	86000000	7			
CURDEF	4	9	1	86000000	38			
CURDEF	4	9	1	86300000	39			
CURDEF	4	10	1	86000000	256			
CURDEF	4	11	2	86000000	100	10	64000	
CURDEF	4	12	2	86000000	100	5	32000	
CURDEF	4	13	4	86000000	100	.01	64000	
CURDEF	4	14	5	86000000	100			
CURDEF	4	14	5	86016000	100			
CURDEF	4	14	5	86016000	110			
CURDEF	4	14	5	86024000	120			
CURDEF	4	14	5	86032000	110			
CURDEF	4	14	5	86032000	100			
CURDEF	4	14	5	86048000	100			
CURDEF	4	14	5	86048000	110			
CURDEF	4	14	5	86056000	120			
CURDEF	4	14	5	86064000	110			
CURDEF	4	14	5	86064000	100			
CURDEF	4	14	5	86080000	100			
CURDEF	4	14	5	86080000	110			
CURDEF	4	14	5	86088000	120			
CURDEF	4	14	5	86096000	110			
CURDEF	4	14	5	86096000	100			
CURDEF	4	14	5	86112000	100			
CURDEF	4	14	5	86112000	110			
CURDEF	4	14	5	86120000	120			
CURDEF	4	14	5	86128000	110			
CURDEF	4	14	5	86128000	100			
CURDEF	4	14	5	86144000	100			
CURDEF	4	14	5	86144000	110			

CURDEF	4	14	5	86152000	120			
CURDEF	4	14	5	86160000	110			
CURDEF	4	14	5	86160000	100			
CURDEF	4	14	5	86176000	100			
CURDEF	4	14	5	86176000	110			
CURDEF	4	14	5	86184000	120			
CURDEF	4	14	5	86192000	110			
CURDEF	4	14	5	86192000	100			
CURDEF	4	14	5	86208000	100			
CURDEF	4	14	5	86208000	110			
CURDEF	4	14	5	86216000	120			
CURDEF	4	14	5	86224000	110			
CURDEF	4	14	5	86224000	100			
CURDEF	4	14	5	86240000	100			
CURDEF	4	14	5	86240000	110			
CURDEF	4	14	5	86248000	120			
CURDEF	4	14	5	86300000	100			
CURDEF	4	14	5	86316000	100			
CURDEF	4	14	5	86316000	110			
CURDEF	4	14	5	86324000	120			
CURDEF	4	14	5	86332000	110			
CURDEF	4	14	5	86332000	100			
CURDEF	4	14	5	86348000	100			
CURDEF	4	14	5	86348000	110			
CURDEF	4	14	5	86356000	120			
CURDEF	4	14	5	86364000	110			
CURDEF	4	14	5	86364000	100			
CURDEF	4	14	5	86380000	100			
CURDEF	4	14	5	86380000	110			
CURDEF	4	14	5	86388000	120			
CURDEF	4	14	5	86396000	110			
CURDEF	4	14	5	86396000	100			
CURDEF	4	14	5	12000	100			
CURDEF	4	14	5	12000	110			
CURDEF	4	14	5	20000	120			
CURDEF	4	14	5	28000	110			
CURDEF	4	14	5	28000	100			
CURDEF	4	14	5	44000	100			
CURDEF	4	14	5	44000	110			
CURDEF	4	14	5	52000	120			
CURDEF	4	14	5	60000	110			
CURDEF	4	14	5	60000	100			
CURDEF	4	14	5	76000	100			
CURDEF	4	14	5	76000	110			
CURDEF	4	14	5	84000	120			
CURDEF	4	14	5	92000	110			
CURDEF	4	14	5	92000	100			
CURDEF	4	14	5	108000	100			
CURDEF	4	14	5	108000	110			
CURDEF	4	14	5	116000	120			
CURDEF	4	14	5	124000	110			
CURDEF	4	14	5	124000	100			
CURDEF	4	14	5	200000	100			
CURDEF	4	15	2	86000000	100	10	64000	1.0
CURDEF	4	16	1	86000000	400			1.0

CURDEF	4	17	2	86000000	100	5	32000	1.0
CURDEF	4	18	4	86000000	100	.01	64000	1.0
CURDEF	4	19	6	86000000				1.0
CURDEF	4	20	1	86000000	256			2.0
CURDEF	4	21	2	86000000	100	10	64000	2.0
CURDEF	4	22	2	86000000	100	5	32000	2.0
CURDEF	4	23	4	86000000	100	.01	64000	2.0
CURDEF	4	24	6	86000000				2.0
CURDEF	4	25	2	86000000	100	10	64000	3.0
CURDEF	4	26	2	86000000	100	5	32000	3.0
CURDEF	4	27	4	86000000	100	.01	64000	3.0
CURDEF	4	28	6	86000000				3.0
CURDEF	4	29	2	86000000	100	10	64000	3.0
CURDEF	4	30	2	86000000	100	5	32000	4.0
CURDEF	4	31	2	86000000	300	300	32000	
SUBCOM	3	1	0	1	8		CONSTANT 7	
SUBCOM	3	1	16	1	9			
SUBCOM	3	1	19	1	10			
SUBCOM	3	1	20	1	11			
SUBCOM	3	1	21	1	12			
SUBCOM	3	1	22	1	13			
SUBCOM	3	1	23	1	14			
SUBCOM	3	1	24	1	15			
SUBCOM	3	1	25	1	16			
SUBCOM	3	1	26	1	17			
SUBCOM	3	1	27	1	18			
SUBCOM	3	1	28	1	19			
SUBCOM	3	1	17	1	16			
SUBCOM	3	2	19	1	20			
SUBCOM	3	2	20	1	21			
SUBCOM	3	2	21	1	22			
SUBCOM	3	2	22	1	23			
SUBCOM	3	2	23	1	24			
SUBCOM	3	2	24	1	25			
SUBCOM	3	2	25	1	26			
SUBCOM	3	2	26	1	27			
SUBCOM	3	2	27	1	28			
SUBCOM	3	2	28	1	29			
SUBCOM	3	2	29	1	30			
SUBCOM	3	2	7	1	31			
FLGDEF	7	11	7	1	1	2		
FLGDEF	7	7	8	5		2		
FLGDEF	7	11	1	1	1	1		
FLGDEF	7	7	2	5		1		
FLGDEF	7	14	13	12	513			
SLPPCT	6	.0002						
TMERP2	6	.002						
MAXTRC	5	40						

FIN

0215

```
FOR   USRSUB,USRSUB
      SUBROUTINE USRSUB(ITIME,IRCD,IFME,ICHCNT,NVALUE,K)
      IPER = 32000
      IF(K .EQ. 28 ) IPER = 48000
      ITME = ITME - 86000000
      IF(ITME .LT. 0 ) ITME = 86400000 + ITIME
      ITWP = MOD(ITME,IPER)
      J = IPER/2
      J1 = IPER*3/4
      IF(ITWP .LT. J ) GO TO 10
      IF (ITWP .LT. J1) GO TO 20
      NVALUE = 120 + (ITWP - J1 )*(-10/(IPER-J1))
      RETURN
   20 NVALUE = 110 + (ITWP - J )*(10/(J1-J))
      RETURN
   10 NVALUE = 100
      RETURN
      END
```