

Lawrence Berkeley National Laboratory

Recent Work

Title

FORTTRAN PROGRAM SUMX

Permalink

<https://escholarship.org/uc/item/6mp0r6g9>

Author

Champomier, Lynn.

Publication Date

1964-04-15

University of California

Ernest O. Lawrence Radiation Laboratory

FORTRAN PROGRAM SUMX

Lynn Champomier

4-15- 1964

TWO-WEEK LOAN COPY

*This is a Library Circulating Copy
which may be borrowed for two weeks.
For a personal retention copy, call
Tech. Info. Division, Ext. 5545*

Berkeley, California

UCRL-11222
c.2

DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

UCRL-11222
Mathematics and
Computers
TID-4500 (28th Ed.)

UNIVERSITY OF CALIFORNIA
Lawrence Radiation Laboratory
Berkeley, California
AEC Contract No. W-7405-eng-48

FORTRAN PROGRAM SUMX

Lynn Champomier

April 15, 1964

FORTRAN PROGRAM SUMX

Contents

Abstract	v
Introduction	1
General Introduction	1
Introduction for Physicists	2
I.0. How to Use SUMX.	3
I.1. Subroutine TAPE	7
I.2. Subroutine SELECT	10
I.3. BLOC3	16
I.4. BLOC4	19
I.5. BLOC5	24
I.6. BLOC6	27
I.7. BLOC7	30
I.8. BLOC8	35
I.9. BLOC9	37
II.0. How to Modify SUMX.	40
II.1. General Plan of the Program	41
II.2. Input-Output	48
II.3. Subroutines TAPESR and BL3SR.	56
II.4. A Sample BLOC	58
Acknowledgment	60
III. Appendices	61
A. Glossary	61
B. Operation of SUMX at LRL Berkeley	61
C. Example of a Logical Record	62
D. SUMX Common, Dimension and Equivalence Statements	66

FORTRAN PROGRAM SUMX

Lynn Champomier

Lawrence Radiation Laboratory
University of California
Berkeley, California

April 15, 1964

ABSTRACT

FORTRAN program SUMX is a data-summarizing program. SUMX constructs displays, in the form of lists, histograms, graphs, etc., from information extracted from sets of records on an input tape. Control cards specify the sets of records, the information to be extracted from each set, and finally, what sort of display is to be made from the information.

INTRODUCTION

General Introduction

SUMX is a 709/7090 FORTRAN II program which was originally designed to aid physicists in summarizing large quantities of data. Because of its generality, SUMX may also be of use to people in the social sciences or in other fields.

SUMX can be made to do many things. Some of the more general displays it makes are one- and two-dimensional histograms, scatter diagrams, and ordered lists.

As an illustrative example consider the following:

A survey of a large number of people was made in which the following data were obtained for each person:

1. Survey serial No.
2. Married status
3. Sex
4. Age
5. Number of children
6. Education
7. Height
8. Salary
9. Weight
10. Average liquor bill per month
11. Age of spouse (if any)

.....

The results of the above survey could be put on magnetic tape, with one record for each person. The first word of each tape record could be that person's Survey serial number. The second word might be his age, etc. A magnetic tape with an identification record followed by records similar in nature to those described in our example and ended by an end-of-file mark will henceforth be called a summary tape.

With the summary tape as input, SUMX could perform tasks such as making a two-dimensional histogram of salary versus liquor bill for all people in the survey who are members of the subset composed of college graduates between the ages of 27 and 32 who have at least two children.

The subroutine that reads the summary tape (called Tape), the subroutines that make the histograms, etc. (called BLOCs), and the subroutine that selects the subsets of people in the survey (called Select) all work under command of control cards which are read by the SUMX program at execution time. (Since SUMX was originally written for high-energy physicists, who deal with a set of physical events rather than with a set of people, the term event, instead of the term person, has been used throughout much of this report. Perhaps the term item would have been more acceptable to a wider class of users. We could then think of an n-word record on our summary tape as the n properties (or quantities or variables) that describe our item.)

The write-up of SUMX has been divided into two sections:

Section I describes the use of the more general (List L) subroutines of the existing SUMX program. Section I is written for people who merely wish to use various combinations of the existing program.

Section II gives the requirements for writing additional subroutines for SUMX. These requirements have been kept minimal, since ease of adding (or deleting) subroutines is one of the principal aims of SUMX. Section II also gives a description of the executive and tape-handling programs. Section II is written for people with some knowledge of programming.

A partial compendium of more special purpose (List S) subroutines has been included in Appendix B.

Introduction for Physicists

SUMX is a 709/7090 FORTRAN program designed to aid the physicist in surveying the data resulting from a large number of events processed event by event in the Alvarez Group programs PANAL-PAKAGE and EXAMIN. It can be made to do many things, but was originally conceived to do simple tasks such as making a histogram on the basis of one word from each event on an EXAMIN output tape, or a scatter plot based on two words.

Because tapes from EXAMIN are apt to contain more than one type of event, a selection subroutine, SELECT, has been included in SUMX. SELECT has been made quite general in the hope that it may serve the purposes of most physicists.

One standard read subroutine, TAPE, is supplied, and if a user's summary tape format does not conform to the conventions of the standard read subroutine, he simply provides another (see List S, Appendix B). An alternative procedure is to write a temporary summary tape in the proper format to be read by the standard read subroutine. Two such programs exist at present (see write-up of CARDEX and READX written by Art Rosenfeld and Stan Wojcicki).

I.0. How to Use SUMX

A. The User Must Know the Following:

(a) The proper format of the summary tape. See description of subroutine TAPE.

(b) How to write control cards. See below, and then see descriptions of those subroutines you are interested in.

(c) How to load the binary decks of the subroutines (see Appendix B).

B. Some Necessary Definitions

One time through all the events on any summary tape will be called a pass. A user may wish to make n passes, in a row, on the same summary tape. The execution of these n passes will be called a run. A user may then wish to make a run on one or more additional summary tapes. The execution of all runs, on all summary tapes to be processed, will be called an entire run, i.e., an entire run is one time on the computer. An entire run might consist of one run, which might consist of one pass.

C. How to Write Control Cards for SUMX

These control cards are put behind the FORTRAN Monitor DATA card. All control cards for SUMX are written either in Hollerith or decimal. Control cards are usually broken up into fields of 10 columns or multiples of 10 columns. The symbol b stands for blank; e.g., if the user is told to put ENDbRUN in the first seven columns of a card, then column 4 should be blank and column 7 should be N.

(a) The control cards for each pass must begin with a Hollerith card upon which the user may write any identification he wishes in the first 72 columns. Whatever the user writes will appear at the top of most output pages.

(b) The control cards for each pass must contain a TAPE card, i.e., a card with TAPE in the first four columns. This card transfers control of the program to subroutine TAPE.

For the cards that follow the TAPE card, consult the write-up of TAPE.

(c) The control cards for each pass must contain one or more BLOC cards. These cards call the various BLOCs. For example, to call BLOC7 write BLOC7 in the first five columns, to call BLOC10 write BLOC10 in the first six columns.

For the cards that follow a BLOC card, see the write-up of that particular BLOC.

(d) The control cards for each pass may contain a SELECT card, i.e., a card with SELECT in first six columns. For the cards that follow the SELECT card, see SELECT.

(e) The control cards for each pass must end with an END PASS card, i.e., a card with ENDbPASS in the first eight columns. This card tells SUMX that all control cards for this pass have been read in.

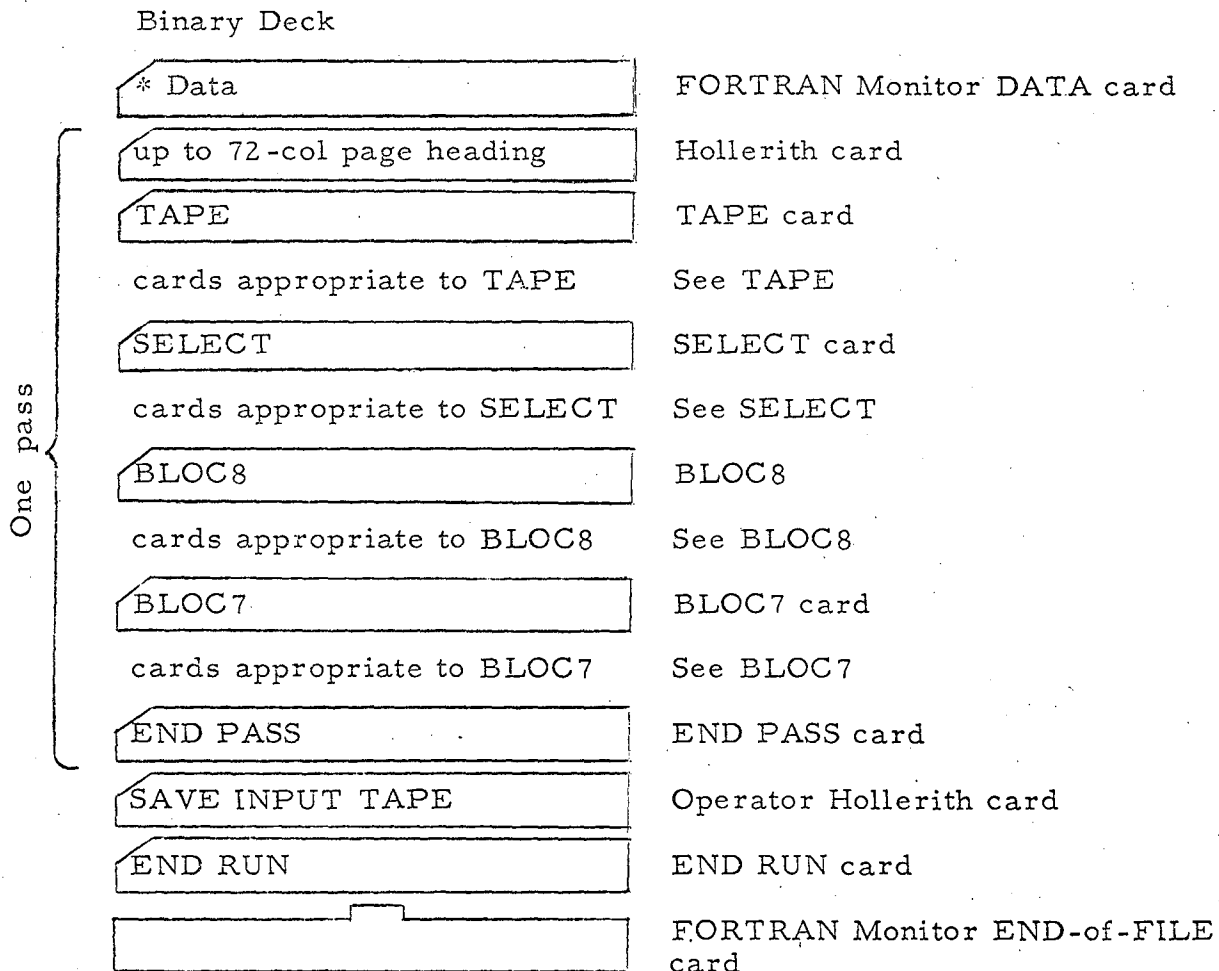
(f) The control cards for each run must be ended with two cards:

First, there must be an operator Hollerith card, i.e., a card upon which the user writes any instructions for the computer operator. The contents of this card are printed on the on-line printer when the run is finished;

Second, the operator Hollerith card must be followed by an END RUN card, i.e., a card with ENDbRUN in the first seven columns. The card causes the operator Hollerith card to be printed on-line, and the computer to stop. The operator reads the operator comments card together with the instructions "TO GET OFF-DEPRESS SWT1--TO CONTINUE-HANG NEW TAPE REEL ON C2--PRESS START."

On the operator Hollerith card, the user should write whether to get off or what tapes are to be mounted or dismounted.

An example of the cards for an entire run consisting of one pass:



An example of the control cards for an entire run consisting of two runs, the first run composed of two passes:

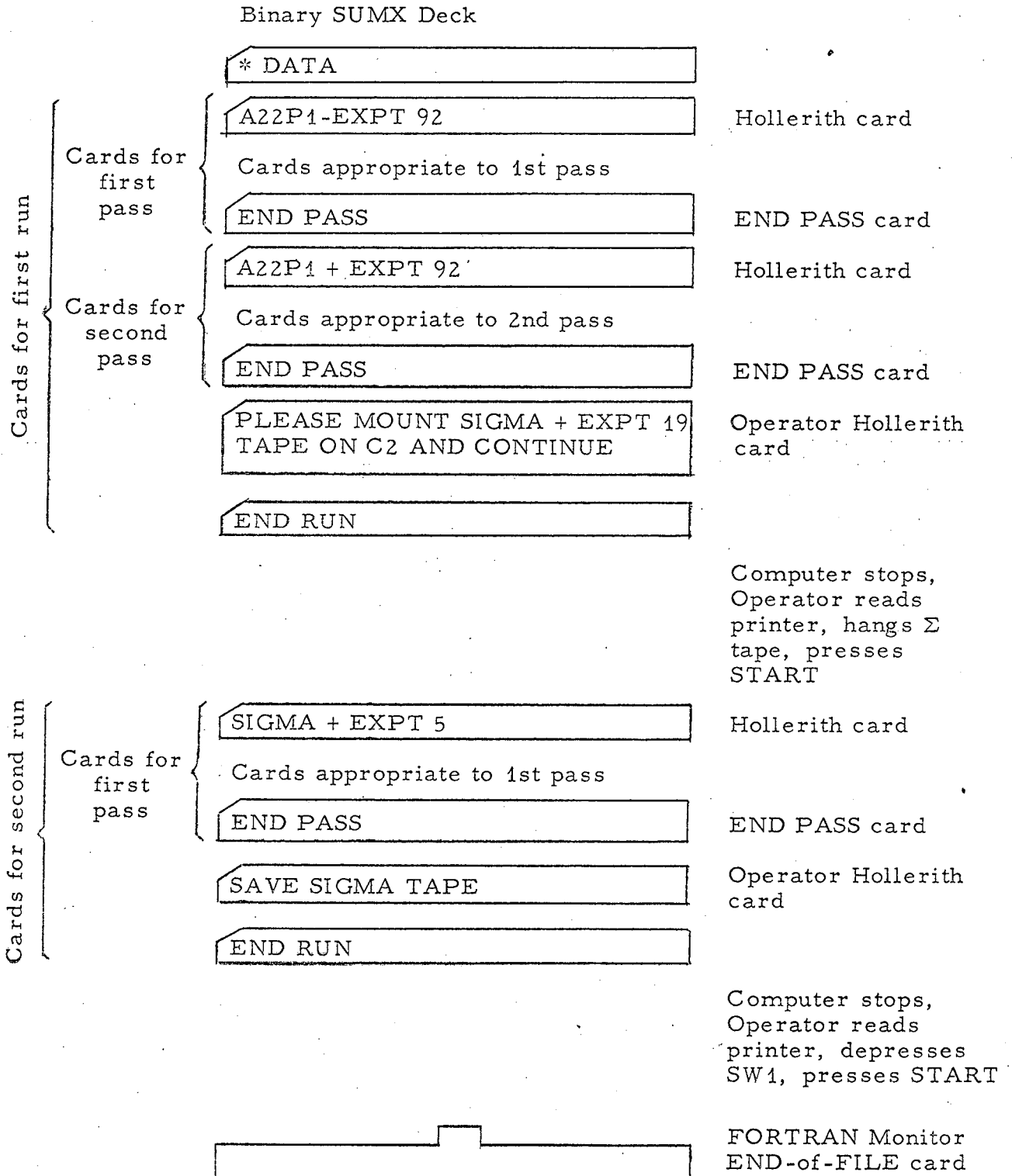


Figure 4 is an actual example of a SUMX run.

SUMX SAMPLE RUN
THE FOLLOWING IS A DESCRIPTION OF A SUMMARY TAPE AND THE CONTROL CARDS USED
ON AN ACTUAL SUMX SAMPLE RUN. THE OUTPUT PRODUCED BY THIS RUN IS INCLUDED
IN THE DESCRIPTION OF THE RESPECTIVE SUBROUTINES.

DESCRIPTION OF THE SUMMARY TAPE
THE SUMMARY TAPE HAD A LABEL AND 330 DATA RECORDS.
THE NTH DATA RECORD HAD THE FOLLOWING COMPOSITION.

WORD	CONTENTS
1	1
2	N
3	3
8	1000.0 -XN (XN IS DECIMAL FORM OF N)
10	XN
11	STANDARD DEVIATION
12	WEIGHT
20	X (X=.02*XN)
22	$P1/2 + 2 \sin X$
23	$P1/2 + 2(\sin X + (1/3)\sin 3X)$
24	$P1/2 + 2(\sin X + (1/3)\sin 3X + (1/5)\sin 5X)$

IBM 407 LISTING OF CONTROL CARDS.

THE CONTROL CARDS ARE USUALLY DIVIDED INTO FIELDS (10 COLUMNS PER FIELD)

FIELD	1	2	3	4	5	6	7	8
COLUMN	10	20	30	40	50	60	70	80
	123456789+123456789+123456789+123456789+123456789+123456789+123456789+12	3456789+						

* DATA
SUMX SAMPLE RUN (THE CONTENTS OF THIS CARD WILL APPEAR ON OUTPUT)
TAPE (FOLLOWING CARD TELLS SUBROUTINE TAPE HOW TO READ THE SUMMARY TAPE)

SELECT 1.
(COMMENTS ARE PERMITTED ON SELECT CONTROL CARDS AFTER COLUMN 40)

TEST2= SUBSET 2 WILL BE ALL RECORDS WITH
INTERVAL 2. 0. 45. WORD 2 BETWEEN 0 AND 45
TEST3= SUBSET 3 WILL BE ALL RECORDS WITH
INTERVAL 10. 10. 30. WORD 10 BETWEEN 10.0 AND 30.0
TEST4= SUBSET 4 WILL BE ALL RECORDS THAT
TEST 2. BELONG TO SUBSET 2
AND
NOT TEST 3. DO NOT BELONG TO SUBSET 3
TEST5= SUBSET 5 WILL BE ALL RECORDS WHERE
INTERVAL 10. 2. 2. TENTH WORD IS 2.0 OR 5.0 OR 9.0 OR 13.0
OR
INTERVAL 10. 5. 5.
OR
INTERVAL 10. 9. 9.
OR
INTERVAL 10. 13. 13.
END SELECT
BLOC4 WILL MAKE IDEO-GRAMS AND/OR RESOLUTION FUNCTIONS

2.
IDEO-GRAM OF SAMPLE DATA IN SUBSET 5
0. 2. 90. 1. 1. 5.
10. 11. 12. 1. 5.
RESOLUTION FUNCTION OF SAME SAMPLE DATA IN SUBSET 5
0. 2. 90. 1. 5.
10. 11. 12. 1. 5.

BLOC5 WILL HISTOGRAM WORD 10 WEIGHTED BY WORD 2
1.
HISTOGRAM OF SUBSET4 0. 1. 50. 4.
10. -2.
BLOC6 WILL MAKE A 2-DIMENSIONAL HISTOGRAM OF SUBSET 2
1.
WORD 10 10. 0. 10. 10. 2.
WORD 10 10. -20.0 10. 8.
BLOC7 JOE USER PUT NAME IN COLUMNS 12 TO 24 TO IDENTIFY FILM
1.
APPROX TO SQUARE WAVE $G(X) = P1$ FOR $0 \leq X \leq P1$
 $G(X) = 0$ FOR $P1 \leq X \leq 2P1$
 $Y1(X) = P1/2 + 2\sin(X)$, $Y2(X) = Y1(X) + 2/3 \sin(3X)$, $Y3 = Y2 + 2/5 \sin(5X)$.
X-AXIS
Y1, Y2, Y3 ON Y-AXIS
0.0 0.2 33. -1.0 0.1 45.
20. 22. 20. 23. 20. 24.

BLOC8 WILL MAKE UN-ORDERED LIST OF SUBSET 3
6. 3.
8. 10. 11. 12. 22. 23.
BLOC9 WILL MAKE ORDERED LIST OF SUBSET 4 WORD 8 IS THE KEY WORD
8. -2. 4.
END PASS
ALL DONE (COMMENT TO OPERATOR)
END RUN
(MONITOR END-OF-FILE CARD GOES HERE)

Fig. 1. A SUMX sample run.

I.1. Subroutine TAPE

A. Purpose

Subroutine TAPE reads logical records from the summary tape into a standard location in the computer memory. An n-word logical record would be read into the locations WORD(1), WORD(2), ..., WORD(n). Subroutine SELECT and the BLOCs would then process the contents of WORD(1), WORD(2), ..., WORD(n).

B. Restrictions

1. The version of subroutine TAPE described below allows one and only one logical record per physical record. (Versions of TAPE that read summary tapes which have many logical records per physical record have been written; see List S, Appendix B.)

2. Records may be at most 1000 words long.

C. Requirements for the Summary Tape Format

1. The identification label.

A summary tape should start with an identification label. Two forms of identification label are allowed by this version of TAPE.

a. The BCD form consists of a 14-word BCD record* and an end-of-file mark. The 14 words are written in Hollerith.

The FORTRAN statements

```
WRITE OUTPUT TAPE 22, 5, (LABEL(J), J = 1, 14)
5 FORMAT(14A6)
END FILE 22
```

might be used to produce this label.

b. The binary form consists of a binary record of at least 10 integer (fixed-point) words* and no end-of-file mark. The FORTRAN statement:

WRITE TAPE 22, (LABEL(J), J = 1, 10) might be used to produce a 10-word label.

2. The data records.

These must be written in the binary mode. These records may be at most 1000 words long, not counting the "signal word."* If the user wishes to

* FORTRAN always puts a "signal word" at the beginning of each physical record written by FORTRAN. Subroutine TAPE does not use the signal word, but TAPE does assume it to be present. See READ TAPE, Page 47, IBM Reference Manual 709/7090 FORTRAN Programming System, C28-6054-2.

write records of length N , where $N \leq 128$, then the FORTRAN statement

WRITE TAPE 22, (LIST(J), J = 1, N)

may be used.

If the user wishes to write records of length N , where $128 < N \leq 1000$, then some FAP routine must be used, because the FORTRAN output statement would break the record up into many physical records. For the purpose of writing binary physical records up to 1000 words in length, subroutine WRITE is available (see Alvarez Memo No. 336).

3. Ending the summary tape.

The summary tape is ended by an end-of-file mark.

D. Contents of the Data Records

The contents of the words of the data records are specified by the user; it is the user's responsibility, when using SUMX, to know what quantity the contents of each word represents.

The contents of any of the words on the data records may be in octal or Hollerith form; however, the List L BLOCs were designed to operate only on integer (fixed-point) words or decimal (floating-point) words. Indeed, BLOC8 assumes the first three words of each data record are integer words.

E. The Control Cards

The control cards for subroutine TAPE are as follows:

TAPE				
BIN ₁	FORM	BCD ₂	NTAPE	
Col. 1	11	21	27 31	41

where FORM = $\begin{cases} 0.0 \text{ or blank if identification label is in the binary form,} \\ 1.0 \text{ if identification label is in the BCD form;} \end{cases}$

NTAPE = $n.0$, where n is the number of the symbolic tape unit on which the summary tape is mounted. (If NTAPE is left blank, then SUMX assumes the summary tape is mounted on logical unit 15.)

(At Berkeley LRL, NTAPE may be 21.0, 22.0, 23.0, or 15.0, which correspond to C1, C2, C3, or B5.)

BIN₁ (or BCD₂) is used if the identification label is in the binary form (or is in the BCD form) and if the user wishes a safeguard to insure against processing the wrong summary tape by mistake.

* See preceding footnote.

For example, if the binary form is used and the first word of the label is 352, then the user sets $BIN_1 = 352.0$ on the control card. If the BCD form is used and the second word of the label is EXPb19, then the user sets $BCD_2 = EXPb19$ on the control card. If the user does not wish a safeguard, he may leave these words blank.

F. Subroutine TAPESR (Optional)

Quite often, a user has written a summary tape and then discovered that he has failed to include the value of some variable. If the variable is a function of other variables that are on the summary tape then the user will find TAPESR useful.

TAPESR is a FORTRAN subroutine written by the user and called by subroutine TAPE.

Consider the example given in Appendix C; suppose the user wanted C(word 14) to be Salary—average monthly liquor bill, he would then include in subroutine TAPESR the FORTRAN statement

$WORD(14) = WORD(8) - WORD(10).$

For further details consult Sec. II.3. Also, see Fig. 2 in Sec. I.3.

I.2. Subroutine SELECT

A. Purpose

Consider the set of all logical records that are on a summary tape. It is the purpose of SELECT to determine which of these records belong to subsets that the user wishes to study.

B. Usage

Subroutine TAPE presents the logical records one at a time to SELECT.

Let R denote the current logical record.

Let Test K denote a conditional statement about members of a subset (Subset K) of records on the summary tape.

SELECT determines if Test K is true when applied to record R (and, thus, if R belongs to Subset K).

Any BLOC may be instructed (by setting the control word, NT = K, on one of that BLOC's control cards) to extract data (for the displays that BLOC makes) only from those records that belong to subset K.

The conditional statement, Test K, is composed of simple conditional statements logically connected by ANDs and ORs.

Test K may be defined (for SELECT) by means of control cards.

1. How many different tests may be defined at one time?

A main test, Test 1, and up to 19 additional tests, Test 2, Test 3, ..., Test 20 may be defined by control cards.

NOTE: On the control cards for each BLOC there is the word NT, "number of test." If the user wishes the BLOC to operate only on those records which pass (for example) Test 4, then he sets NT = 4.

2. How is a test defined by control cards?

If the user wishes to define Test 5, for example, he writes

```
TEST5 =
```

This card is followed by other control cards that make simple conditional statements or that tell how the statements are to be logically connected.

For example, referring to the example in the Appendix, page 66, the control cards

```
TEST5 =
```

```
DINTERVAL 4.0 20.0 30.0 (simple conditional statement)
```

AND

INTERVAL 5.0 3.0 7.0 (simple conditional statement)

would define Test 5 to be $(20.0 \leq c(\text{word } 4) \leq 30.0) \text{ AND } (3 \leq c(\text{word } 5) \leq 7)$. Thus, Subset 5 would be those records of people whose age, $c(\text{word } 4)$, was in the interval between 20.0 and 30.0, and whose number of children, $c(\text{word } 5)$, was in the interval between 3 and 7.

NOTE: A D was in column 1 of the first INTERVAL card, since $c(\text{word } 5)$ is in the decimal mode in the example. An I was in column 1 of the second interval card since $c(\text{word } 4)$ was in the integer mode.

3. The simple conditional statements.

a. The INTERVAL statement is of the form: lower bound $\leq c(\text{word } n) \leq$ upper bound. Consider the second conditional statement in the preceding example. We had the statement

$$3 \leq c(\text{word } 5) \leq 7.$$

This statement would be written on the control card in the form (with $M=I$)

MININTERVAL										5.0										3.0										7.0									
Col. 1										9	11					21				31				41															
												N					lower				upper																		
																	bound					bound																	

where $M=I$ means the quantities referred to are to be interpreted as being in integer mode, and $M=D$ means decimal mode.

NOTE: It is conventional throughout SUMX to break control cards up into fields of multiples of ten columns. These fields are either Hollerith or decimal. Blanks are not ignored in Hollerith fields. When punching the values of constants or the location of variables on control cards, always write in decimal, even if the quantities are in integer mode. Thus, in the example above n is written as 5.0 even though $n=5$.

b. The COMPARE statement is of the form $c(\text{word } n) > c(\text{word } m) + \text{Constant}$.

Referring to the example on page 66, we might demand age of spouse + 5.5 years; i.e. $c(\text{word } 4) > c(\text{word } 11) + 5.5$. This statement would be written on a control card in the form (with $M=D$):

MCOMPARE										3.0										11.0										5.5									
Col. 1										8	11					21				31				41															
												n					m					Constant																	

where $M=I$ means the quantities referred to are to be interpreted as being in integer mode, and $M=D$ means decimal mode.

DCOMPARE	0.0	11.0	-30.0
----------	-----	------	-------

TEST8 =

TEST	3.0
------	-----

AND

TEST	5.0
------	-----

NOT TEST	5.0
----------	-----

The AND and OR connector cards are used to connect the simple conditional statements.

These control cards are written

AND

Col. 1

OR

Col. 1

An AND card or an OR card is put between the simple conditional statement cards. If no connector card is put between two simple conditional statement cards, SELECT will OR the statements.

OR has greater binding power than AND; i.e., simple statements are combined disjunctively (see example on next page).

a. Restrictions:

(i) At most 5 compound statements may be AND'ed together to define a test.

(ii) At most 5 simple statements may be OR'ed together to define one of the compound statements mentioned in (a). However, if a test is defined to be only one compound statement, then up to 25 simple statements may be OR'ed together to define the compound statement.

5. Other rules.

Subroutine SELECT is called by a control card with the word SELECT written in the first six columns. The control cards for SELECT are terminated by a control card with END SELECT written in the first ten columns.

(a) If a record R does not pass the main test, Test 1, then R is not processed and the next record on the summary tape is immediately examined by SELECT; this enables the user to quickly discard large subsets in which he has no interest.

(b) If the user is making more than one pass on a summary tape and does not wish to go to the trouble of redefining tests that were defined for the preceding pass, then the control cards

SELECT

RETAIN

END SELECT

will retain (redefine) those tests that were defined for the preceding pass; those tests remain defined as before.

(c) Control cards to define new tests may be added after the RETAIN card.

(d) Suppose Tests 1 through 17 were defined for the preceding pass. The control cards

```

SELECT
RETAIN
CLEAR TEST      5.0
TEST5 =
ICOMPARE        2.0      5.0      -1.0
OR
DINTERVAL       4.0      20.0     30.0
AND
TEST            3.0
TEST18 =
IINTERVAL       5.0      3.0      7.0
END SELECT

```

Col. 1 11 21 31 41

would:

(a) retain (redefine) Tests 1 through 17 to be the same as defined in the preceding pass;

(b) clear (undefine) Test 5;

(c) define Test 5 to be the statement

$[c(\text{word } 2) > c(\text{word } 5) - 1 \text{ or } 20.0 \leq c(\text{word } 4) \leq 30.0]$

and $[\text{record } R \text{ does pass Test } 3]$;

(d) define Test 18 (a previously undefined test) to be the statement $3 \leq c(\text{word } 5) \leq 7$.

Figure 2 is the output of the SUMX sample run shown in Fig. 1.

SUMX SAMPLE RUN (THE CONTENTS OF THIS CARD WILL APPEAR ON OUTPUT)

SELECT DETERMINES THE TRUTH VALUE OF THE TESTS (CONDITIONAL STATEMENTS) AS THEY ARE APPLIED TO THE SUMMARY TAPE RECORDS
IF TEST 1 IS NOT TRUE WHEN APPLIED TO A RECORD, THE RECORD WILL NOT BE PROCESSED FURTHER.

THE TESTS FOR THIS PASS WERE DEFINED AS FOLLOWS

SELECT	(COMMENTS ARE PERMITTED ON SELECT CONTROL CARDS AFTER COLUMN		
TEST2=			SUBSET 2 WILL BE ALL RECORDS WI
IINTERVAL	2.	0.	45. WORD 2 BETWEEN 0 AND 45
TEST3=			SUBSET 3 WILL BE ALL RECORDS WI
DINTERVAL	10.	10.	30. WORD 10 BETWEEN 10.0 AND 30.0
TEST4=			SUBSET 4 WILL BE ALL RECORDS TH
TEST	2.		BELONG TO SUBSET 2
AND			AND
NOT TEST	3.		DO NOT BELONG TO SUBSET 3
TEST5=			SUBSET 5 WILL BE ALL RECORDS WH
DINTERVAL	10.	2.	2. TENTH WORD IS 2.0 OR 5.0 OR 9.0
OR			
DINTERVAL	10.	5.	5.
OR			
DINTERVAL	10.	9.	9.
OR			
DINTERVAL	10.	13.	13.
END SELECT			

WHERE MINTERVAL X B1 B2 MEANS B1 L E C(WORD X) LTE B2
MCOMPARE X Y C C(WORD X) GT C(WORD Y) + C
AND M=(I,D) MEANS VARIABLES ARE (INTEGER,DECIMAL) NUMBERS.

Fig. 2. SELECT output of the SUMX sample run.

I.3. BLOC3

A. Purpose

BLOC3 writes new summary tapes.

B. Usage

BLOC3 has a subroutine called BL3SR (described in Sec. II) which is either a dummy subroutine (and does nothing) or is a FORTRAN subroutine written by the user (which allows him to create new data as functions of the old data in a record.)

BLOC3 and BL3SR used in conjunction with TAPE, TAPESR and SELECT allow the user some flexibility in the creation of new summary tapes. Figure 3 illustrates the logical flow of SUMX when many of the optional features are in use.

The summary tapes written by BLOC3 will have labels (see TAPE Sec I.1). The user has the following options:

$M < 1.0$, reproduce old summary-tape label on the new summary tape.

$M = 1.0$, write new binary label on the new summary tape

where first word of the new label = ID1 (integer)

second word of the new label = ID2 (integer)

$M \geq 2.0$, write new BCD label on the new summary tape.

where first word of the new label = WD1 (BCD)

second word of the new label = WD2 (BCD)

.....

14th word of the new label = WD14 (BCD)

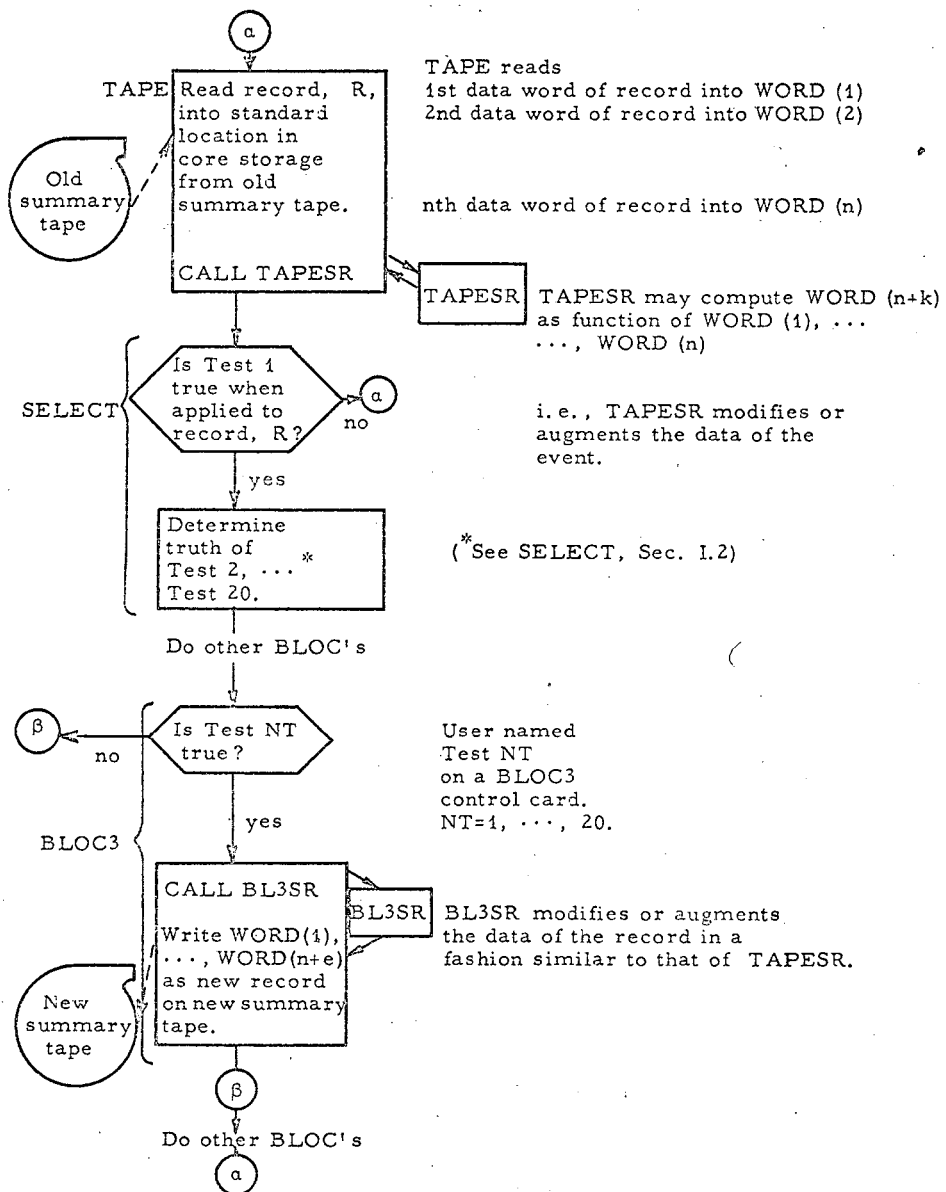
1. Example

Suppose the user wished to create a new summary tape on logical unit 15 (his input tape was on logical 22) of those events which pass Test 7 (i.e. NT = 7.0). The user wishes the new BCD label

EXPO25REEL02Joe Smith Word05Word06Word07

Word08Word09 Word14

This label is composed of 14 words, each of 6 characters.



MUB-2932

Fig. 3. Logical flow of SUMX.

2. Control Cards.

BLOC3

Col. 1 5

2.0

7.0

. 15.0

WORD11... WORD14

Col. 1 11 21 31 41 51 74
 MODE ID1 ID2 NT NTAPE last 4 words of label

EXPO25REEL02JOE SMITH WORD05WORD06... WORD10

Col. 1 60
 first 10 words of label

where NT = number of test (see Sec. I.2).

NTAPE = logical tape number (see NTAPE in subroutine TAPE, Sec. I.1 for allowable values).

NOTE: If the user were not creating a new BCD label there would not be a third card.

I.4. BLOC4

A. Purpose

BLOC4 makes Gaussian ideograms and resolution functions.

Consider the function

$$f(t) = (2\pi)^{-1/2} \sum_j \frac{W_j}{SF \cdot \sigma_j} \exp \left[-\frac{1}{2} \left(\frac{t - \bar{X}_j}{\sigma_j \cdot SF} \right)^2 \right].$$

Roughly, $f(t)$ is the sum of a family of normal density functions, where $\bar{X}_j$ = mean value, σ_j = standard deviation, and W_j = weight of some quantity associated with the j th event selected from a summary tape, and where SF is a scaling factor.

If there were M quantities associated with each event, we could consider

$$g(t) = \sum_{k=1}^M f(t) = (2\pi)^{-1/2} \sum_j \sum_{k=1}^M \frac{W_{jk}}{SF_k \cdot \sigma_{jk}} \exp \left[-\frac{1}{2} \left(\frac{t - \bar{X}_{jk}}{\sigma_{jk} \cdot SF_k} \right)^2 \right],$$

and would say $g(t)$ has a multiplicity of M .

BLOC4 computes the quantities

$$\int_{\chi_i}^{\chi_{i+1}} g(t) dt = (2\pi)^{-1/2} \sum_j \sum_{k=1}^M \int_{\chi_i}^{\chi_{i+1}} \frac{W_{jk}}{\sigma_{jk} \cdot SF_k} \exp \left[-\frac{1}{2} \left(\frac{t - \bar{X}_{jk}}{\sigma_{jk} \cdot SF_k} \right)^2 \right] dt$$

where χ_i are points along the t -axis defined by $\chi_i = \chi_0 + (i) \Delta\chi_i$, $i = 1, \dots, N$. The values $\int_{\chi_i}^{\chi_{i+1}} g(t) dt$ are called a Gaussian ideogram if the $\bar{X}_{jk} \neq 0$; otherwise they are called a resolution function.

B. Example

Consider a star with two outgoing π^+ . Suppose we want an ideogram of the momentum distribution of outgoing π^+ . There are obviously two π^+ momenta associated with each event, thus, a multiplicity of two. The fitted momentum of the j th event will then yield $\bar{X}_{j1}$ and $\bar{X}_{j2}$ for the two pions, with associated errors σ_{j1} and σ_{j2} . If the chi-squared distribution for these fitted events had about the right shape but was too wide by a factor of 1.4, then an appropriate scaling factor SF , for calculating a resolution function is probably $(1.4)^{1/2} = 1.2$. Further, suppose we have the following:

- $L(\bar{X}_{j1})$ = word 10 of each event,
- $L(\sigma_{j1})$ = word 11 of each event,
- $L(W_{j1})$ = word 12 of each event,
- SF_1 = 1.20
- $L(\bar{X}_{j2})$ = word 25 of each event,

$L(\sigma_{j2})$ = word 26 of each event,

$L(W_{j2})$ = word 27 of each event,

SF_2 = 1.20

Symbol	sample value	Meaning
N	= 2	= number of displays (ideograms and/or resolution functions) to be made ($1 \leq N \leq 10$).
x_0	= 2.5 BeV/c	= starting value of x axis (=0.0 if making resolution function)
Δx	= 0.1 BeV/c	= interval width
$N_{\Delta x}$	= 20	= number of intervals on x axis ($1 \leq N_{\Delta x} \leq 110$).
M	= 2	= multiplicity ($1 \leq M \leq 5$). [i.e., M items (e.g., M energys) per record (event) used in making display]
N_{RF}	= $\begin{Bmatrix} 0 \\ 1 \end{Bmatrix}$	= $\begin{cases} \text{this display is a resolution function} \\ \text{this display is an ideogram} \end{cases}$
NT	= 7	= make display from data obtained only from those records (events) which are members of subset NT (e.g., subset 7) [i.e., those records (events) which pass Test NT (e.g., Test 7)].

Now, suppose the user wished an ideogram of the two π^+ 's and a resolution function of just the first π^+ . The control cards that would cause BLOC4 to construct these displays are listed below.

BLOC4

Col.1 6

Bloc Header Card

2.0

Col.1 N 10

GAUSSIAN IDEOGRAM OF BOTH π^+ S

Col.1

72

Up to 72 Hollerith characters for comments to be printed.

2.5 0.1 20.0 2.0 1.0 7.0

Col.1 11 24 31 41 51

x_0 Δx $N_{\Delta x}$ M N_{RF} NT

10.0	11.0	12.0	1.2	
Col. 1	11	21	31	41

$L(\bar{x}_{j1})$ $L(\sigma_{j1})$ $L(W_{j1})$ SF_1

25.0	26.0	27.0	1.2	
Col. 1	11	21	31	41

$L(\bar{x}_{j2})$ $L(\sigma_{j2})$ $L(W_{j2})$ SF_2

RESOLUTION FUNCTION OF FIRST P1

Col. 1

72

Up to 72 Hollerith characters for comments to be printed.

0.0	0.1	20.0	1.0	0.0	7.0
Col. 1	11	21	31	41	51

x_0 Δx $N_{\Delta x}$ M N_{RF} NT

10.0	11.0	12.0	1.2	
Col. 1	11	21	31	41

$L(\bar{x}_{j1})$ $L(\sigma_{j1})$ $L(W_{j1})$ SF_1

Figure 4 is the BLOC4 output of SUMX sample run shown in Fig. 1.

THESE NUMBERS REFER TO THE INTERVAL NUMBERS ABOVE

Fig. 4(a). BLOC 4 output of the SUMX sample run.

```

SUMX SAMPLE RUN (THE CONTENTS OF THIS CARD WILL APPEAR ON OUTPUT)  L(MEAN)      =WRD(5) 10
RESOLUTION FUNCTION OF SAME SAMPLE DATA IN SUBSET 5              L(STAN.DEV.)=WRD(5) 11
(STANDARD) BLOC4 NT= 5 RESOLUTION FUNCTION OR IDEOGRAM           L(WEIGHT)=WRD(5) 12
TOTAL NUMBER OF EVENTS IN THIS DISPLAY= 4 MULTIPLICITY=1/EVENT   SCALE FACTOR(S)= 1.000
UNDERFLOW= 0. OVERFLOW= 0.0050
WIDTH OF RESOLUTION FUNCTION AT HEIGHT NMAX/2 = 0.887E 00

CONTENTS      0.0688 0.0456 0.0300 0.0233 0.0195 0.0167 0.0148 0.0133 0.0124 0.0117
LIMITS      0.      1 0.20 2 0.40 3 0.60 4 0.80 5 1.00 6 1.20 7 1.40 8 1.60 9 1.80 10 2.00
INTVL NOS

CONTENTS      0.0111 0.0106 0.0102 0.0097 0.0094 0.0092 0.0090 0.0087 0.0085 0.0082
LIMITS      2.00    11 2.20 12 2.40 13 2.60 14 2.80 15 3.00 16 3.20 17 3.40 18 3.60 19 3.80 20 4.00
INTVL NOS

CONTENTS      0.0080 0.0077 0.0075 0.0072 0.0069 0.0066 0.0064 0.0061 0.0058 0.0055
LIMITS      4.00    21 4.20 22 4.40 23 4.60 24 4.80 25 5.00 26 5.20 27 5.40 28 5.60 29 5.80 30 6.00
INTVL NOS

CONTENTS      0.0053 0.0051 0.0048 0.0046 0.0043 0.0041 0.0038 0.0036 0.0034 0.0032
LIMITS      6.00    31 6.20 32 6.40 33 6.60 34 6.80 35 7.00 36 7.20 37 7.40 38 7.60 39 7.80 40 8.00
INTVL NOS

CONTENTS      0.0030 0.0028 0.0026 0.0025 0.0023 0.0021 0.0020 0.0018 0.0017 0.0016
LIMITS      8.00    41 8.20 42 8.40 43 8.60 44 8.80 45 9.00 46 9.20 47 9.40 48 9.60 49 9.80 50 10.00
INTVL NOS

CONTENTS      0.0015 0.0013 0.0012 0.0011 0.0010 0.0010 0.0009 0.0008 0.0007 0.0007
LIMITS     10.00    51 10.20 52 10.40 53 10.60 54 10.80 55 11.00 56 11.20 57 11.40 58 11.60 59 11.80 60 12.00
INTVL NOS

CONTENTS      0.0006 0.0005 0.0005 0.      0.      0.      0.      0.      0.      0.
LIMITS     12.00    61 12.20 62 12.40 63 12.60 64 12.80 65 13.00 66 13.20 67 13.40 68 13.60 69 13.80 70 14.00
INTVL NOS

CONTENTS      0.      0.      0.      0.      0.      0.      0.      0.      0.      0.
LIMITS     14.00    71 14.20 72 14.40 73 14.60 74 14.80 75 15.00 76 15.20 77 15.40 78 15.60 79 15.80 80 16.00
INTVL NOS

```

```

0.0750
0.0735
0.0720
0.0705
0.0690
0.0675 X
0.0660 X
0.0645 X
0.0630 X
0.0615 X
0.0600 X
0.0585 X
0.0570 X
0.0555 X
0.0540 X
0.0525 X
0.0510 X
0.0495 X
0.0480 X
0.0465 X
0.0450 XX
0.0435 XX
0.0420 XX
0.0405 XX
0.0390 XX
0.0375 XX
0.0360 XX
0.0345 XX
0.0330 XX
0.0315 XX
0.0300 XX
0.0285 XXX
0.0270 XXX
0.0255 XXX
0.0240 XXX
0.0225 XXXX
0.0210 XXXX
0.0195 XXXXX
0.0180 XXXXX
0.0165 XXXXXX
0.0150 XXXXXX
0.0135 XXXXXX
0.0120 XXXXXXXX
0.0105 XXXXXXXXXX
0.0090 XXXXXXXXXXXX
0.0075 XXXXXXXXXXXX
0.0060 XXXXXXXXXXXX
0.0045 XXXXXXXXXXXX
0.0030 XXXXXXXXXXXX
0.0015 XXXXXXXXXXXX
0.0000 XXXXXXXXXXXX
123456789+123456789+123456789+123456789+123456789+123456789+123456789+123456789+123456789+123456789+
10      20      30      40      50      60      70      80      90
THESE NUMBERS REFER TO THE INTERVAL NUMBERS ABOVE

```

Fig. 4(b). BLOC 4 output of the SUMX sample run.

I.5. BLOC5

A. Purpose

BLOC5 constructs up to 40 pictorial histograms on a single pass through a summary tape, the histogram being displayed on one page of printouts, and the actual content of the bins provided, with cross referencing, being displayed on the following page. Each histogram may have a multiplicity of up to THREE; i.e., the contents of as many as three locations may be included in the same histogram. Moreover, the histograms may be weighted by specifying, in addition to the locations whose contents are to be histogrammed, the corresponding locations whose contents will be added to the appropriate histogram's bin; a location of 0 for the weighting factor signifies a weight of 1.

B. Usage

Both integer and floating point words may be histogrammed and the corresponding weights may be taken from locations containing either integer or floating point numbers. Locations whose contents are integer numbers are denoted (according to the SUMX convention) by the negative of the actual location. All histograms are computed in floating point form and integerized immediately before printout so that, for example, a weighting factor of 6.5 has some meaning.

The following control cards illustrate how BLOC5 is used to construct histograms of:

(a) a floating point variable (located in word 10) with no weighting factor, where only events passing Test 2 (see SELECT) will be included in the histogram;

(b) a floating point variable (located in word 15) with no weighting factor and an integer variable (located in word 30) with a floating point weighting factor (located in word 25).

(c) three floating point variables, the first having no weighting factor and located in word 50, the second (located in word 51) having an integer weighting factor (located in word 20), and the last (located in word 52) having a floating point weighting factor (located in word 21). Here, the option is taken of not displaying the histogram.

BLOC5					
3.0					
11					
NH					
CHI SQUARE	0.0	1.0	16.0	2.0	
	21	31	41	51	61 71
HOLLERITH	X_0	ΔX	$N_{\Delta X}$	NT	XD
10.0					
$L(X_1)$	11	$L(W_1)$	21		

	0.0	5.0	100.0			
	21	31	41	51	61	71
HOLLERITH	X_0	ΔX	$N_{\Delta X}$	NT	XD	
15.0	-30.0	25.0				
	11	21	31	41		
$L(X_1)$	$L(W_1)$	$L(X_2)$	$L(W_2)$			
	-1.0	.2	10.0		1.0	
	21	31	41	51	61	71
HOLLERITH	X_0	ΔX	$N_{\Delta X}$	NT	XD	
50.0	51.0	-20.0	52.0	21.0		
	11	21	31	41	51	61
$L(X_1)$	$L(W_1)$	$L(X_2)$	$L(W_2)$	$L(X_3)$	$L(W_3)$	

We have

NH = the number of histograms to be constructed on this pass
(Note: $1.0 \leq NH \leq 40.0$),

HOLLERITH = twenty Hollerith characters to be printed as a histogram heading,

X_0 = starting value of histogram,

ΔX = size of histogram interval,

$N_{\Delta X}$ = number of histogram intervals (Note: $1.0 \leq N_{\Delta X} \leq 110.0$),

NT = number of the associated test (Note: $1.0 \leq NT \leq 20.0$),

XD = 0.0 means histogram will be displayed,

XD = 1.0 means histogram will not be displayed,

$L(X_n)$ is the location of nth word to be histogramed, and

$L(W_n)$ is the location of the weight of the nth word.

Figure 5 is the BLOC5 output of the SUMX sample run shown in Fig. 1.

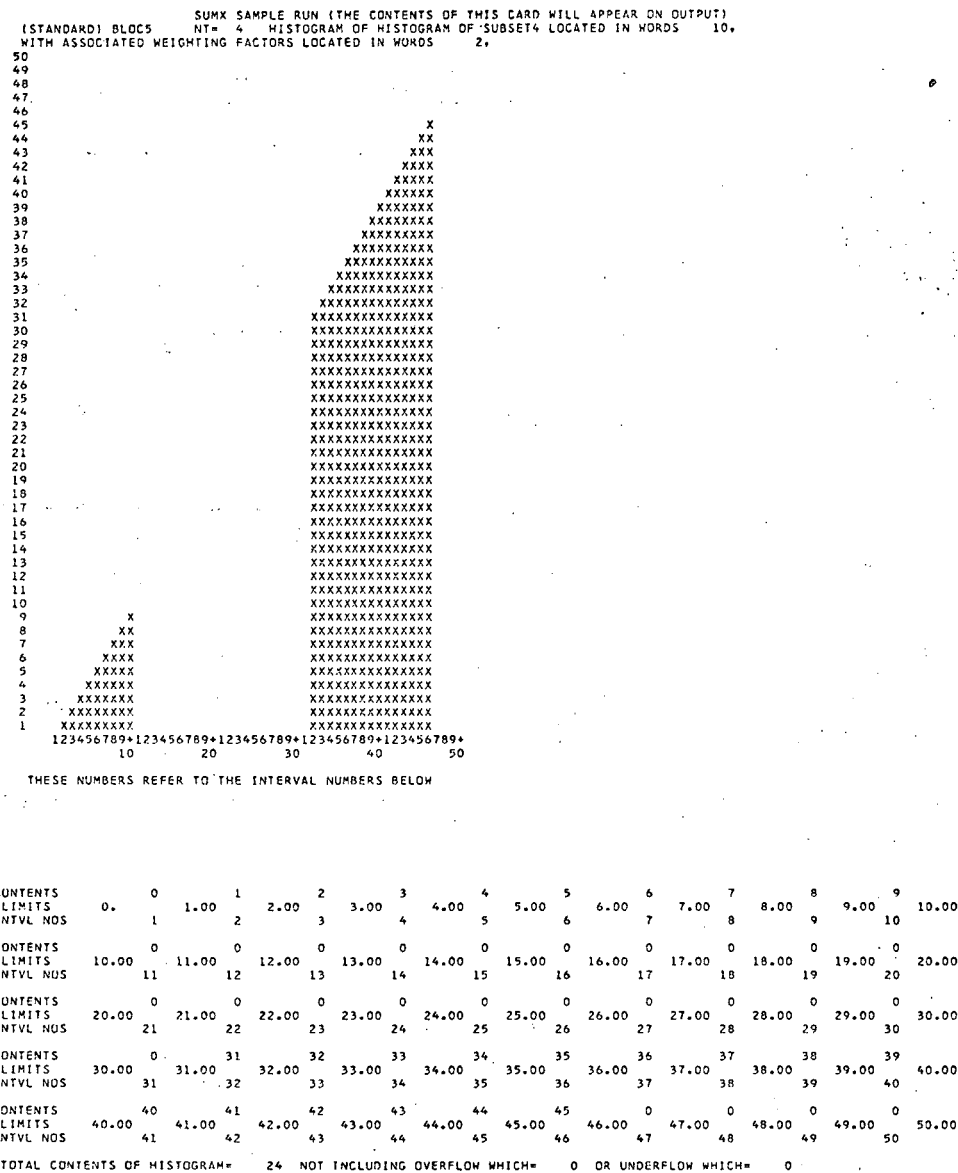


Fig. 5. BLOCK 5 output of the SUMX sample run.

1.6. BLOC6

A. Purpose

BLOC6 constructs up to five two-dimensional histograms on a single pass through a summary tape. Though BLOC6 is limited, for reasons of storage, to dimensions of 15×20 bins, one could easily obtain a 15×60 histogram by spreading the results over three histograms, that is, over three pages. The multiplicity of BLOC6 is one, so that the contents of only one word for the X direction and one word for the Y direction may be included in a given histogram.

B. Usage

Both integer and floating-point words may be histogrammed, locations whose contents are integer numbers being denoted (according to the SUMX convention) by the negative of the actual location.

The following control cards illustrate the manner BLOC6 is used to construct histograms of:

(a) two floating-point variables (located in words 10 and 15) whose values are in the sine-cosine range--only events passing Test 4 (see SELECT) will be included in this histogram;

(b) a floating-point variable (located in word 25) ranging from 0.0 to 100.0 and an integer variable (located in word 30) ranging from 0 to 100;

(c) two integer variables (located in words 50 and 51) ranging from -1000 to 1000 and 0 to 1000, respectively.

BLOC6					
3.0					
11	21				
NH					
COSP11	10.0	-1.0	.2	10.0	4.0
11	21	31	41	51	61
HOLLERITH L(X) X_0 ΔX $N_{\Delta X}$ NT					
COSP12	15.0	-1.0	.1	20.0	
11	21	31	41	51	61
HOLLERITH L(Y) Y_0 ΔY $N_{\Delta Y}$					

1st
Histogram

	25.0	0.0	10.0	10.0		
	11	21	31	41	51	61
HOLLERITH	L(X)	Xo	ΔX	$N_{\Delta X}$	NT	2nd Histogram
	-30.0	0.0	10.0	10.0		
	11	21	31	41	51	61
HOLLERITH	L(Y)	Yo	ΔY	$N_{\Delta Y}$		
	-50.0	-1000.0	150.0	15.0		
	11	21	31	41	51	61
HOLLERITH	L(X)	Xo	ΔX	$N_{\Delta X}$	NT	3rd Histogram
	-51.0	0.0	100.0	10.0		
	11	21	31	41	51	61
HOLLERITH	L(Y)	Yo	ΔY	$N_{\Delta Y}$		

We have

NH = number of histograms to be constructed on this pass (Note: $1.0 \leq NH \leq 5.0$),

HOLLERITH = ten characters of Hollerith to be printed as a histogram heading,

Xo, Yo = starting values of histogram in X and Y directions,

$\Delta X, \Delta Y$ = size of histogram intervals in X and Y directions,

$N_{\Delta X}$ = number of histogram intervals in X direction (Note: $1.0 \leq N_{\Delta X} \leq 15.0$),

$N_{\Delta Y}$ = number of histogram intervals in Y direction (Note: $1.0 \leq N_{\Delta Y} \leq 20.0$),

NT = number of the associated test [Note: $1.0 \leq NT \leq 20.0$ (see SELECT, Sec. 1.2)],

L(X) = location of X coord of point, and

L(Y) = location of Y coord of point.

Figure 6 is the BLOC6 output of the SUMX sample run shown in Fig. 1.

SUMX SAMPLE RUN (THE CONTENTS OF THIS CARD WILL APPEAR ON OUTPUT) PAGE 9

(STANDARD) BLOC6 NT= 2 HISTOGRAM OF WORD 10 VS WORD 10 ,LOCATED IN WORDS 10 AND 10

INTERVALS	0.	10.0	20.0	30.0	40.0	50.0	60.0	70.0	80.0	90.0	100.0	TOTALS
	I TO I	I TO I	I TO I	I TO I	I TO I	I TO I	I TO I	I TO I	I TO I	I TO I	I TO I	I
-20.00	I	I	I	I	I	I	I	I	I	I	I	I
TO	I	I	I	I	I	I	I	I	I	I	I	I
-10.00	I	I	I	I	I	I	I	I	I	I	I	I
TO	I	I	I	I	I	I	I	I	I	I	I	I
-0.	I	I	I	I	I	I	I	I	I	I	I	I
TO	I	I	I	I	I	I	I	I	I	I	I	I
10.00	I	I	I	I	I	I	I	I	I	I	I	I
TO	I	I	I	I	I	I	I	I	I	I	I	I
20.00	I	I	I	I	I	I	I	I	I	I	I	I
TO	I	I	I	I	I	I	I	I	I	I	I	I
30.00	I	I	I	I	I	I	I	I	I	I	I	I
TO	I	I	I	I	I	I	I	I	I	I	I	I
40.00	I	I	I	I	I	I	I	I	I	I	I	I
TO	I	I	I	I	I	I	I	I	I	I	I	I
50.00	I	I	I	I	I	I	I	I	I	I	I	I
TO	I	I	I	I	I	I	I	I	I	I	I	I
60.00	I	I	I	I	I	I	I	I	I	I	I	I
TOTALS	I	I	I	I	I	I	I	I	I	I	I	I

TOTAL CONTENTS OF HISTOGRAM= 45 NOT INCLUDING UNDERFLOW AND OVERFLOW WHICH ARE AS FOLLOWS

UNDERFLOW ON WORD 10 = 0 OVERFLOW ON WORD 10 = 0

UNDERFLOW ON WORD 10 = 0 OVERFLOW ON WORD 10 = 0

Fig. 6. BLOCK 6 output of the SUMX sample run.

I.7. BLOC7

A. Purpose

The purpose of BLOC7 is to make scatter diagrams of points (i.e., pairs of words) obtained from those events (records) which pass test NT (which are members of subset NT). BLOC7 may, at the user's option, plot from zero to two envelopes or graphs upon the scatter diagram.

BLOC7 will make up to 20 scatter diagrams at a time and include up to three points obtained from each record in each scatter diagram (i.e., $1 \leq \text{multiplicity} \leq 3$).

B. Example

Suppose there are two particles associated with those events (which are members of subset 3) whose momenta, P_1 and P_2 , are to be included in the first scatter diagram.

Further suppose we wish a scatter diagram of the momentum, P , associated with those events which are members of subset 9.

Suppose we have: $L(P_1 \text{ tangential}) = \text{word 14,}$
 $L(P_2 \text{ tangential}) = \text{word 15,}$
 $L(P_1 \text{ normal}) = \text{word 27,}$
 $L(P_2 \text{ normal}) = \text{word 28,}$
 $L(P \text{ tang}) = \text{word 11,}$
 $L(P \text{ norm}) = \text{word 12,}$

and $-2 \times 10^2 \text{ BeV/c} \leq P \text{ tang} \leq 2 \times 10^2 \text{ BeV/c}$
 $-10^2 \text{ BeV/c} \leq P \text{ norm} \leq 10^2 \text{ BeV/c.}$

Then the control cards for the two scatter diagrams might be

BLOC7	JOE USER	
Col. 1	12	24

Bloc header card:
users' name for ID.

2.0
Col. 1 10
NCRT

Tells BLOC7 that there
are 2 scatter dia-
grams to be made.

P1 AND P2 TANG VS. NORM P
BLAH
BLAH

3 Hollerith comment
cards. Contents will
appear at top of
scatter diagram.

P TANG ON X AXIS

P NORM ON Y AXIS

-200.0	10.0	40.0	-100.0	10.0	20.0	3.0
--------	------	------	--------	------	------	-----

Col. 1 11 21 31 41 51 61 71

Xo	ΔX	$N_{\Delta X}$	Yo	ΔY	$N_{\Delta Y}$	NT
14.0	27.0	15.0	28.0			

Col. 1 11 21 31 41 51 61

L(X₁) L(Y₁) L(X₂) L(Y₂) L(X₃) L(Y₃)

P ALPHA P TAN VS. P NORM

BLAH

BLAH

P TANG ON X AXIS

P NORM ON Y AXIS

-200.0	10.0	40.0	-100.0	10.0	20.0	9.0
--------	------	------	--------	------	------	-----

Col. 1 11 21 31 41 51 61

Xo	ΔX	$N_{\Delta X}$	Yo	ΔY	$N_{\Delta Y}$	NT
11.0	12.0					

Col. 1 11 21 31 41 51 61

L(X₁) L(Y₁) L(X₂) L(Y₂) L(X₃) L(Y₃)

Hollerith will appear
on X axis.

Hollerith will appear
on Y axis.

Tells BLOC7 how to
construct grid lines
and which records
to obtain points from.

In first scatter diagram,
there are only 2 points
to be plotted, thus
L(X₃) and L(Y₃) are
left blank.

2 Control Cards for
superimposing graphs
on scatter diagram
(following). In this
example, there are no
graphs; thus, both
cards are left blank.

3 Hollerith comment
cards for second
scatter diagram.

Hollerith will appear
on X axis.

Hollerith will appear
on Y axis.

Tells BLOC7 how to
construct grid lines
and which records to
obtain points from.

In second scatter dia-
gram, there is only
one point to be plotted,
thus, L(X₂), ..., L(Y₃)
are left blank.

2 Control Cards for
superimposing graphs
on scatter diagram
(following). In this
example, there are no
graphs; thus, both
cards are left blank.

We use the following notation:

NCRT is number of scatter diagrams to be made ($1 \leq \text{NCRT} \leq 20$),

X₀ is starting value of X axis

X is the distance between fine grid lines,

N_{ΔX} is the number of fine grid lines (every fifth or tenth accentuated,
 $1 \leq N_{\Delta X} \leq 50$),

Y₀
ΔY
N_{ΔY} } are similar quantities for Y axis,

NT is number of subset (or test) from which to obtain records from
which to obtain points to be plotted in scatter diagram,

L(X₁) is location (in record) of X coord of first point to be plotted,

L(Y₁) is location (in record) of Y coord of first point to be plotted, and

L(X₂)
L(Y₂)
L(X₃)
L(Y₃) } are similar quantities for second and third points (if there is a
second or a third point).

C. Subroutine SKIN

1. Purpose

The purpose of subroutine SKIN is to allow the user to superimpose graphs on the scatter diagrams.

2. Standard version

A standard version of SKIN is provided the user. This version plots the kinematical boundary of a Dalitz plot for a three-body reaction

$$1 + 2 \rightarrow 3 + 4 + 5.$$

Six quantities are required by this version of SKIN:

NP = number of points to be plotted ($NP \leq 500.0$),

P₁ = c.m. energy of the 1 + 2 system,

P₂ = mass of particle 3,

P₃ = mass of particle 4,

P₄ = mass of particle 5, and

Mode = 0.0 connect points with straight lines

Mode = 0.1 display points as X's.

Care must be taken to provide $P_1, \dots, P_4$ in the same units. The X and Y axes are the invariant mass squared of particles 4 and 5, and 3 and 5, respectively. The number of points, NP, should not be too small, or otherwise the Dalitz plot envelope will look jagged. For approximately 500 points the routine takes 5 sec of time on the 709 computer.

3. Control Cards

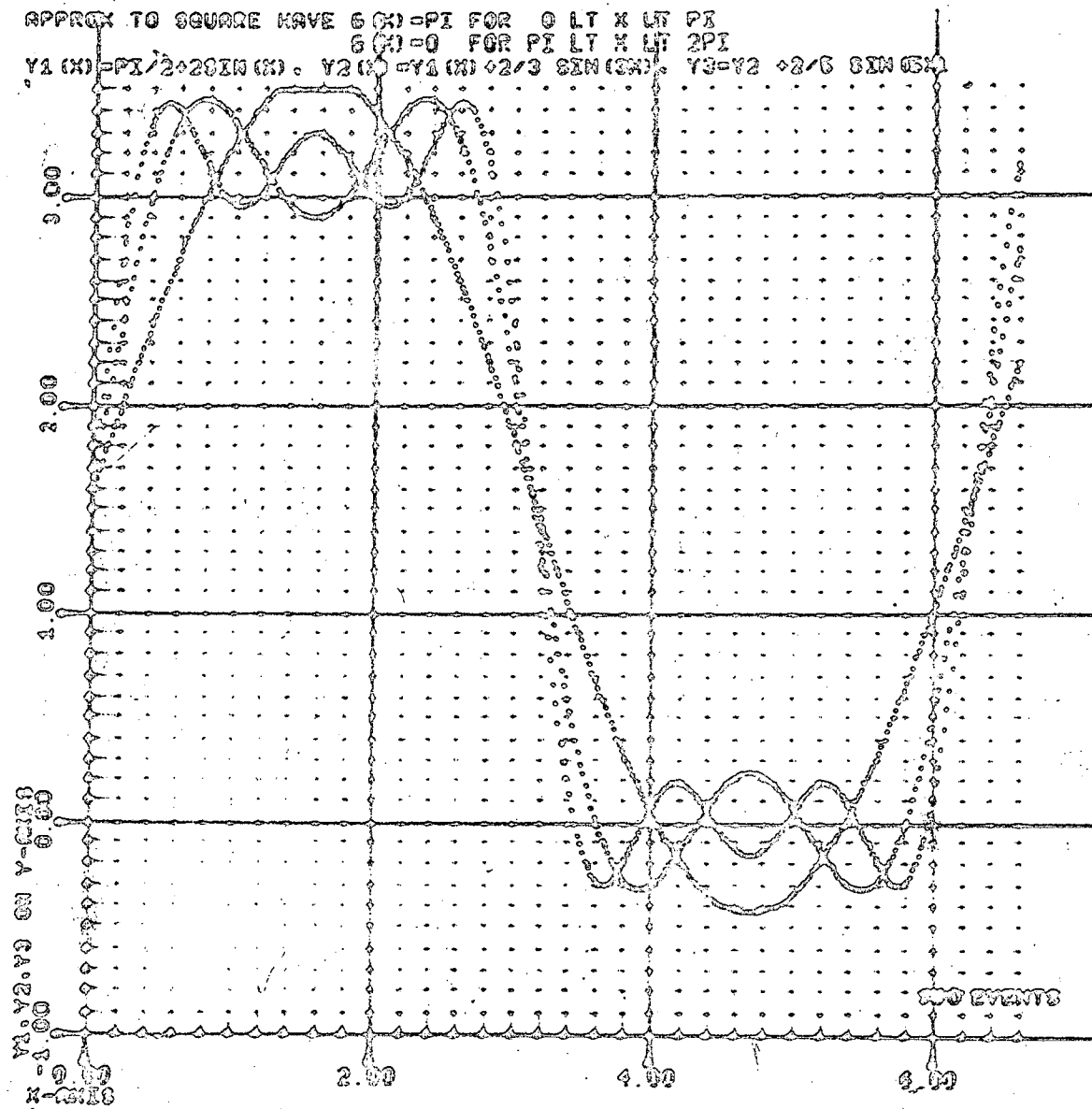
A control card for SKIN causes one graph to be plotted, if the control card is not left blank. Two control cards must always be used, even if they are both left blank.

A control card for SKIN has the form

NP	P ₁	P ₂	P ₃	P ₄	MODE	
Col.1	11	21	31	41	51	61

The user may write his own subroutine SKIN in order to have a different family of curves displayed. A person familiar with FORTRAN programming is advised to read Sec. II of this report and obtain a listing of both BLOC7 and SKIN before undertaking such a task.

Figure 7 is the partial BLOC7 output of the SUMX sample run shown in Fig. 1.



MUB-3217

Fig. 7. Partial BLOCK 7 output of the SUMX sample run.

I.8. BLOC8

A. Purpose

BLOC8 makes a listing of the first, second, and third words (roll, frame, and beam track, assumed to be integers) and up to 12 additional words of each record, on the summary tape, which belongs to subset NT (see SELECT, Sec. I.2).

B. Usage

Consider the example in Appendix C. Suppose the user wished a list of all 13 words of each record that belonged to subset 9. The control cards would be

BLOC8									
10.0 9.0									
Col. 1	11	21							
NW	NT								
4.0 -5.0 -6.0 7.0 8.0 9.0 10.0									
Col. 1	11	21	31	41	51	61	71		
11.0 12.0 13.0									
Col. 1	11	21	31	41	51				

where

NW = the number of words to be listed = 10
(the first three words are always listed).

NOTE: The locations of the fifth and sixth words are given as -5.0 and -6.0, since the minus sign indicates to BLOC8 that these numbers are stored in the integer (fixed point) mode.

NOTE: The third card is not used if $NW \leq 7$.

NOTE: The remaining two fields on the third card would be used if $NW = 12$.

Figure 8 is the BLOC8 output of the SUMX sample run shown in Fig. 1.

SUMX SAMPLE RUN (THE CONTENTS OF THIS CARD WILL APPEAR ON OUTPUT) PAGE 2
 (STANDARD) BLOCK NT= 3
 10 PER LINE --- ROLL FRAME AND B.T. - FOLLOWED BY WORDS 8 10 11 12 22 23

0	1 10 3	1 11 3	1 12 3	1 13 3	1 14 3	1 15 3	1 16 3	1 17 3	1 18 3	1 19 3
	9.900E 02	9.890E 02	9.880E 02	9.870E 02	9.860E 02	9.850E 02	9.840E 02	9.830E 02	9.820E 02	9.810E 02
	1.000E 01	1.100E 01	1.200E 01	1.300E 01	1.400E 01	1.500E 01	1.600E 01	1.700E 01	1.800E 01	1.900E 01
	2.000E-01	2.000E-01	2.000E-01	2.000E-01	2.000E-01	2.000E-01	2.000E-01	2.000E-01	2.000E-01	2.000E-01
	1.000E 00	1.000E 00	1.000E 00	1.000E 00	1.000E 00	1.000E 00	1.000E 00	1.000E 00	1.000E 00	1.000E 00
	1.968E 00	2.007E 00	2.046E 00	2.085E 00	2.124E 00	2.162E 00	2.200E 00	2.238E 00	2.275E 00	2.313E 00
	2.345E 00	2.416E 00	2.486E 00	2.554E 00	2.620E 00	2.684E 00	2.746E 00	2.806E 00	2.863E 00	2.918E 00
10	1 20 3	1 21 3	1 22 3	1 23 3	1 24 3	1 25 3	1 26 3	1 27 3	1 28 3	1 29 3
	9.800E 02	9.790E 02	9.780E 02	9.770E 02	9.760E 02	9.750E 02	9.740E 02	9.730E 02	9.720E 02	9.710E 02
	2.000E 01	2.100E 01	2.200E 01	2.300E 01	2.400E 01	2.500E 01	2.600E 01	2.700E 01	2.800E 01	2.900E 01
	2.000E-01	2.000E-01	2.000E-01	2.000E-01	2.000E-01	2.000E-01	2.000E-01	2.000E-01	2.000E-01	2.000E-01
	1.000E 00	1.000E 00	1.000E 00	1.000E 00	1.000E 00	1.000E 00	1.000E 00	1.000E 00	1.000E 00	1.000E 00
	2.350E 00	2.386E 00	2.423E 00	2.459E 00	2.494E 00	2.530E 00	2.565E 00	2.599E 00	2.633E 00	2.667E 00
	2.971E 00	3.021E 00	3.068E 00	3.113E 00	3.155E 00	3.195E 00	3.231E 00	3.265E 00	3.296E 00	3.324E 00
20	1 30 3									
	9.700E 02									
	3.000E 01									
	2.000E-01									
	1.000E 00									
	2.700E 00									
	3.349E 00									

Fig. 8. BLOCK 8 output of the SUMX sample run.

I.9. BLOC9

A. Purpose

BLOC9 constructs an ordered list of words.

B. Usage

One word, the key word, KW, is obtained from every record which is a member of subset NT (see SELECT, Sec. I.2).

Let KW(j) denote the key word obtained from the jth record of subset NT.

Let AW(j) denote any other word obtained from the jth record of subset NT.

Let m denote the number of records which are members of subset NT.

BLOC9 forms two lists,

$KW(1), KW(2), \dots, KW(m),$ and

$AW(1), AW(2), \dots, AW(m).$

BLOC9 now puts the KW list into ascending order and orders the AW list according to the order of the KW list.

1. Example

Suppose the lists before ordering were

$n = 1 \quad 2 \quad 3 \quad 4$

$KW(n) = 6 \quad 4 \quad 5 \quad 3$

$AW(n) = 20 \quad 24 \quad 33 \quad 21.$

Then, the lists after ordering would be

$n = 4 \quad 2 \quad 3 \quad 1$

$KW(n) = 3 \quad 4 \quad 5 \quad 6$

$AW(n) = 21 \quad 24 \quad 23 \quad 20.$

Actually, BLOC9 can construct up to two lists similar to $AW(1), \dots, AW(m)$. Thus, we call these two lists $AW_1(1), \dots, AW_1(m)$ and $AW_2(1), \dots, AW_2(m)$.

2. Control Cards

Consider the example given on page 66; suppose the user wished

KW = number of years of school,

AW₁ = salary,

AW₂ = liquor bill,

and that the records belong to subset 7.

Then L(SW) = word 6, L(AW₁) = word 8, L(AW₂) = word 10 and NT = 7.

The control cards would be

BLOC9					
-6.0	8.0	10.0	7.0		
Col. 1	11	21	31	41	51
L(KW)	L(AW ₁)	L(AW ₂)		NT	

NOTE: As is the usual convention in BLOC control cards, the -6.0 indicates to BLOC9 that KW is stored in the integer (fixed point) mode. The convention also holds for AW₁ and AW₂.

Figure 9 is the BLOC9 output of the SUMX sample run shown in Fig. 1.

(STANDARD) BLOC9		SUMX SAMPLE RUN (THE CONTENTS OF THIS CARD WILL APPEAR ON OUTPUT)												PAGE 10	
		NT= 4 ORDERED LIST OF WORD LOCATED AT						8 AND ASSOCIATED WORDS LOCATED AT						2 0	
1	24	23	22	21	20	19	18	17	16	15	0.42				
	9.550E 02	9.560E 02	9.570E 02	9.580E 02	9.590E 02	9.600E 02	9.610E 02	9.620E 02	9.630E 02	9.640E 02					
	45	44	43	42	41	40	39	38	37	36					
11	14	13	12	11	10	9	8	7	6	5	0.83				
	9.650E 02	9.660E 02	9.670E 02	9.680E 02	9.690E 02	9.910E 02	9.920E 02	9.930E 02	9.940E 02	9.950E 02					
	35	34	33	32	31	9	8	7	6	5					
21	4	3	2	1	66367										
	9.960E 02	9.970E 02	9.980E 02	9.990E 02											
	4	3	2	1											
MEDIAN=		0.966E 03		MEAN =		0.974E 03		SQRT OF VARIANCE =		0.164E 02					
SUMX PASS		ALL DONE		(COMMENT TO OPERATOR)											
												PAGE 1			

Fig. 9. BLOCK 9 output of the SUMX sample run.

II.0. How to Modify SUMX

A. SUMX May be Modified at Different Levels.

(a) The writing of subroutines TAPESR and BL3SR requires a knowledge of FORTRAN. The programmer should consult Sec. II.3 and Appendix D, and the diagram in Sec. I.3.

(b) The writing of a new BLOC requires a knowledge of FORTRAN. The programmer should consult Secs. II.1, II.2, and II.4, and Appendix D.

(c) Putting SUMX into operation at a new computer installation requires a knowledge of FAP. The programmer should consult Sec. II.2 and Appendix D.

II.1. General Plan of the Program

A. Purpose

SUMX was planned with one overriding purpose in mind--ease of modification.

B. Introduction

The executive program, SXEXC, views the subroutines TAPE, SELECT, BLOC3, BLOC4, ..., BLOC14, and BLOC15 as "little black boxes." Subroutine TAPE puts the current logical record, R, into a COMMON vector, WORD. Subroutine SELECT applies conditional statements (tests) to the logical record, R, in the vector WORD, and puts the truth values of the tests into a common vector JTW. The BLOCs obtain data from the logical record, R, in the vector WORD. When all logical records have been read by TAPE, the BLOC's make displays of the data they have obtained. This, in a nutshell, is the function of the little black boxes.

Suppose we have two little black boxes, both called, for example, BLOC5, which make different kinds of displays, such as histograms and least-square polynomial approximations. If both these boxes have the same calling sequence, the same standard set of COMMON statements, and the same general organizational structure, it will make no difference to the rest of SUMX which of these routines happen to be loaded into the computer memory.

The standard set of COMMON statements are given in Appendix D.

The required general organizational structure follows:

C. The Phases

SUMX has three phases. All the above little black boxes have in their call statement the argument NPHASE.

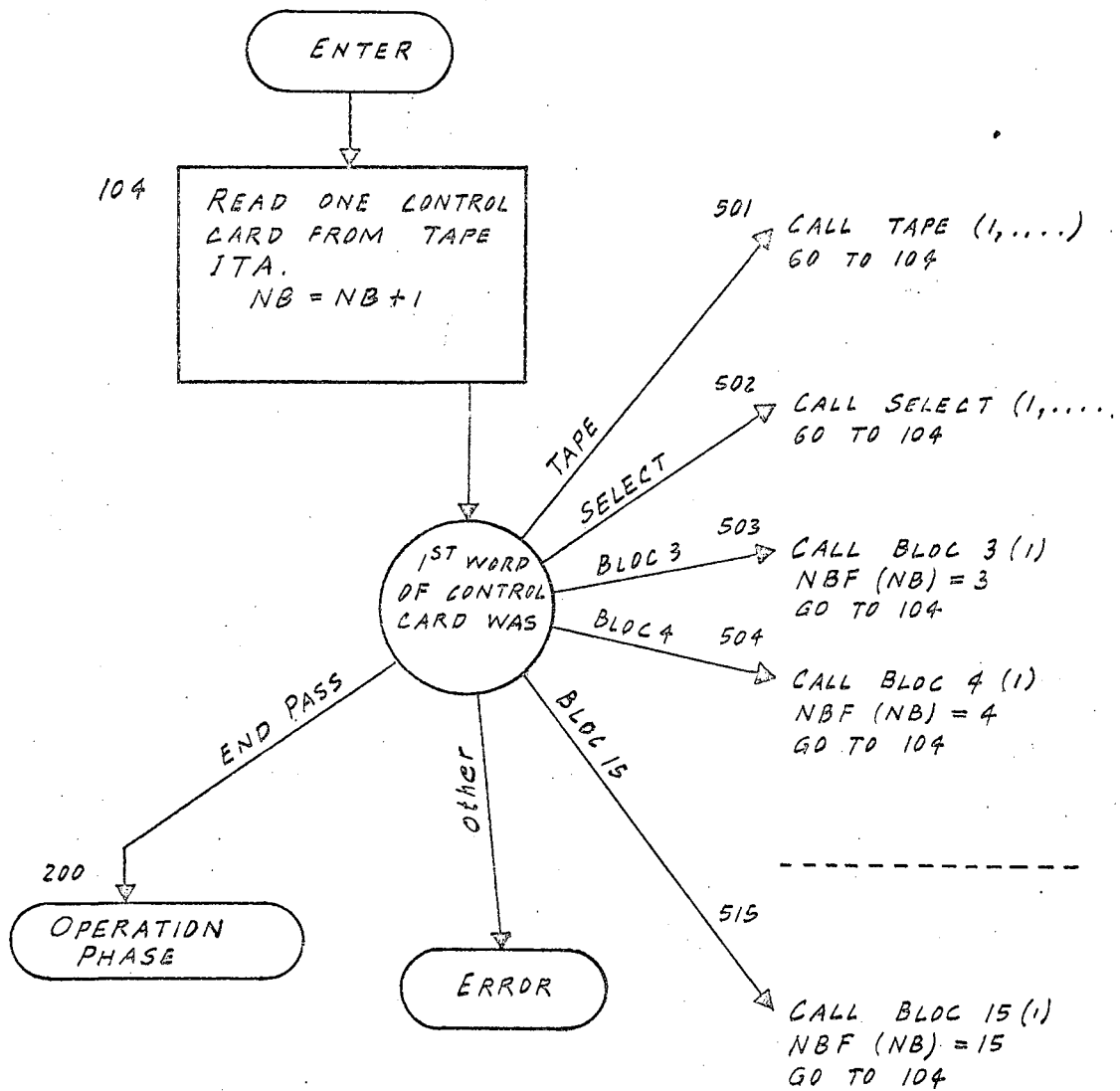
1. Initial phase (NPHASE = 1).

The executive program, SXEXC, reads control cards from the monitor input tape (A2 at Berkeley, symbolically ITA). When SXEXC reads a control card with the name of a subroutine written on the card, SXEXC calls the named subroutine (with NPHASE = 1). The subroutine then reads for itself an appropriate number of cards from ITA, initializes itself, and returns control to SXEXC. SXEXC repeats this process until all control cards for one pass have been read. SXEXC has kept a list (vector NBF) of all subroutines thus called (see Fig. 10).

NOTE: An example of what tape ITA (The SUMX control cards) may look like is given in Sec. I.0.

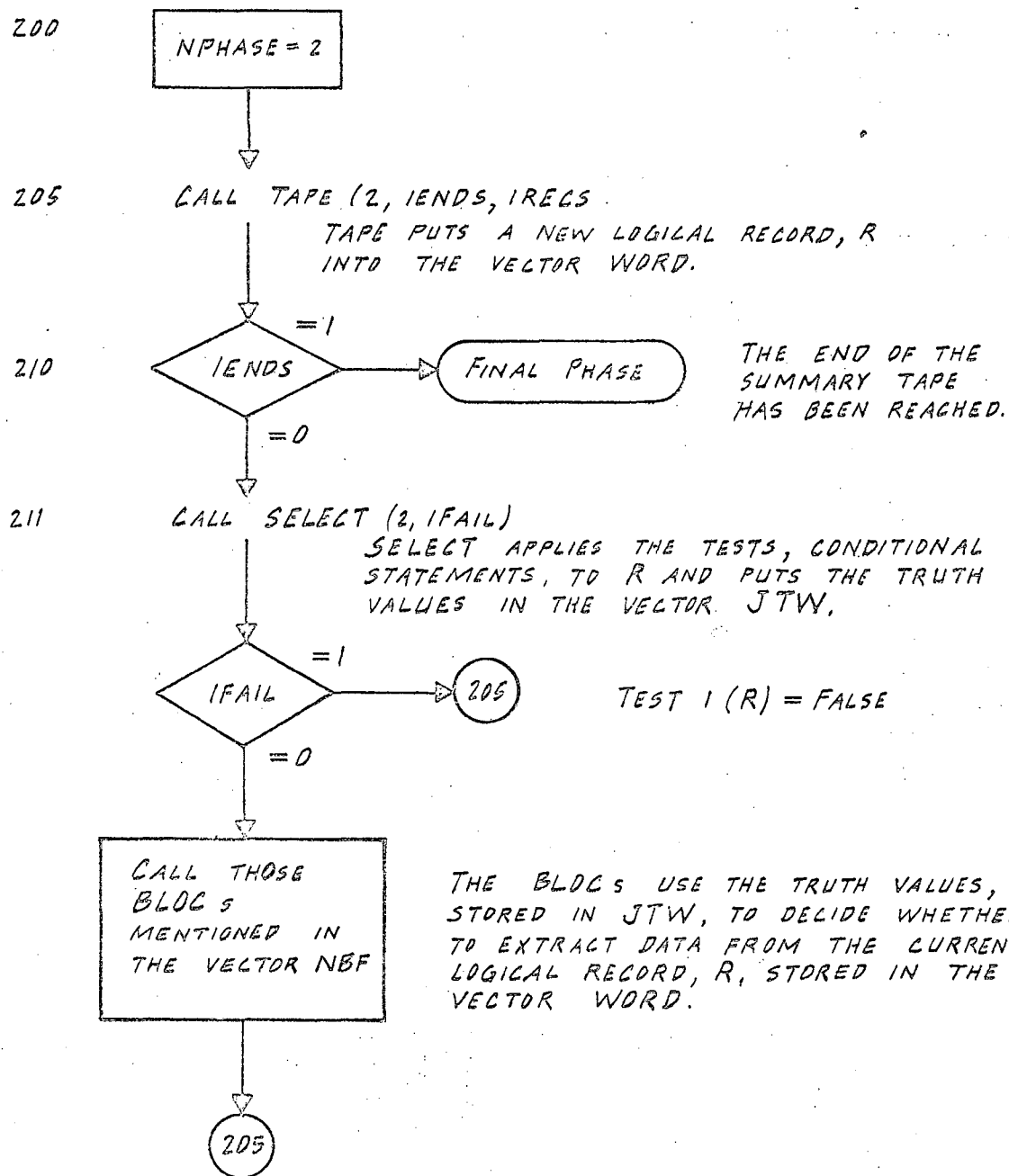
2. Operation phase (NPHASE = 2).

SXEXC calls TAPE, SELECT, and those BLOC's mentioned in vector NBF, until subroutine TAPE indicates the end of the summary tape has been reached. For the flow diagram see Fig. 11.



MUB-2924

Fig. 10. Flow diagram of the initial phase.



MUB-2925

Fig. 11. Flow diagram of the operation phase.

3. Final phase (NPHASE = 3)

SXEXC calls TAPE, SELECT, and those BLOC's mentioned in vector NBF. This time they output their reports, displays, etc., on the monitor output tape (A3 at Berkeley, symbolically ITB).

D. The Subroutines

The following description of the subroutines merely gives their required general organizational structure.

1. Subroutine TAPE

(a) Purpose: Each time TAPE is called during the operation phase, TAPE puts one logical record into the vector WORD.

(b) Calling sequence: TAPE is called by the statement CALL TAPE (NPHASE, IENDS, IRECS).

(c) General flow diagram (see Fig. 12).

2. Subroutine SELECT

(a) Purpose: Each time SELECT is called during the operation phase, SELECT applies the conditional statements (tests) to the logical record which is the vector WORD. The truth values of the tests are put into the vector JTW, where 0 = true and 1 = false.

(b) Calling sequence: SELECT is called by the statement CALL SELECT (NPHASE, IFAIL).

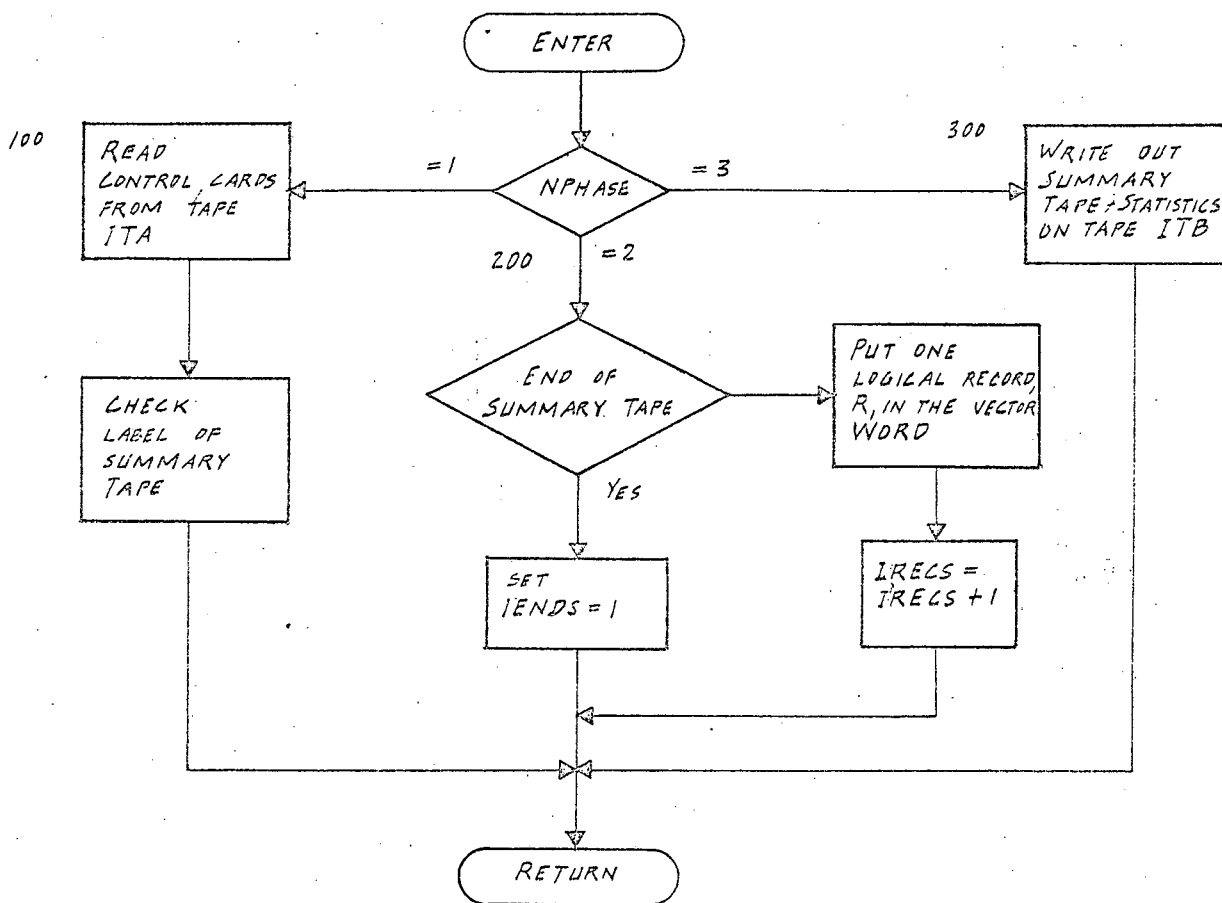
(c) General flow diagram (see Fig. 13).

3. The BLOC Subroutines

(a) Purpose: Each BLOC makes some sort of display.

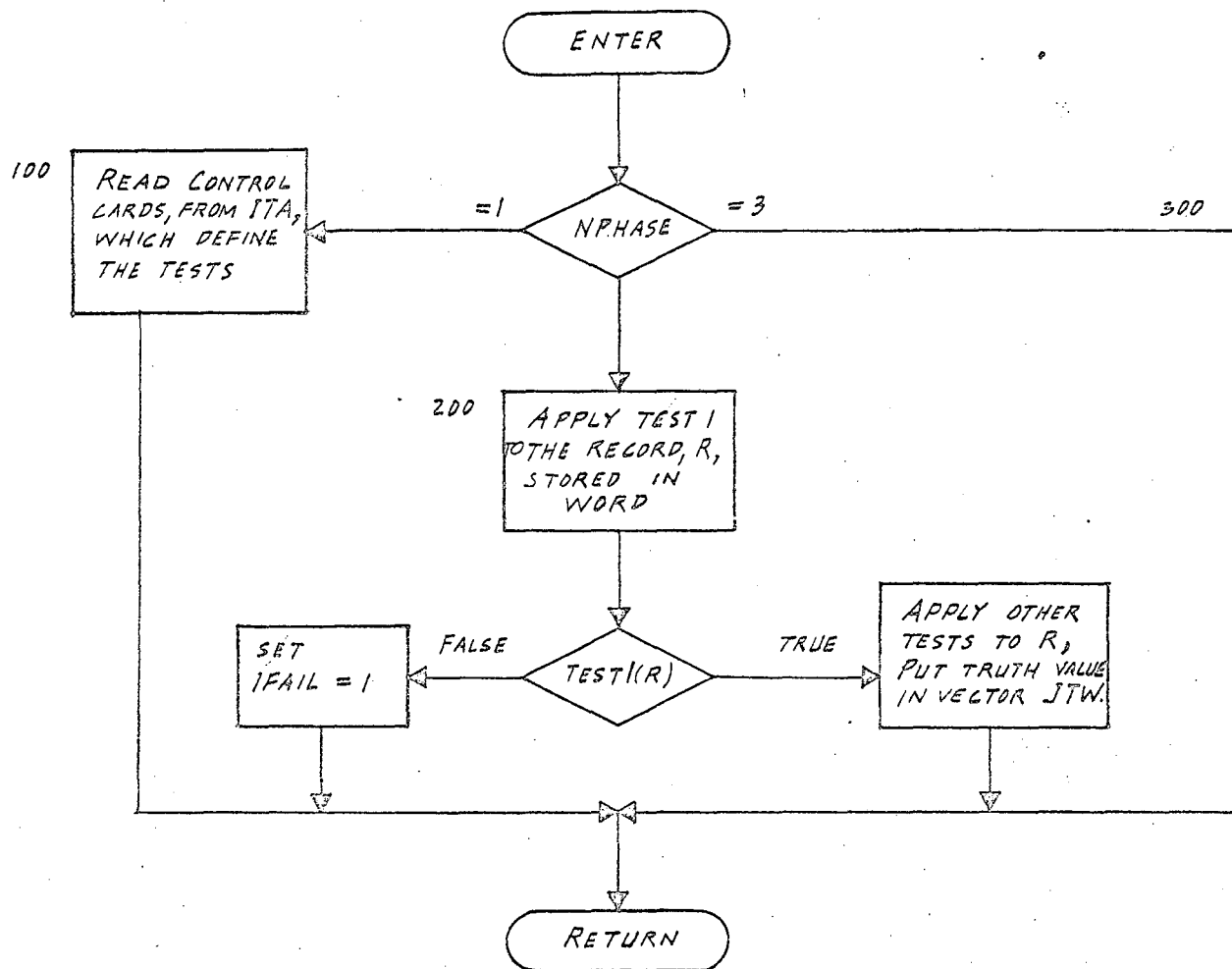
(b) Calling sequence: BLOC N ($3 \leq N \leq 15$) is called by the statement CALL BLOCN (NPHASE).

(c) General flow diagram (see Fig. 14).



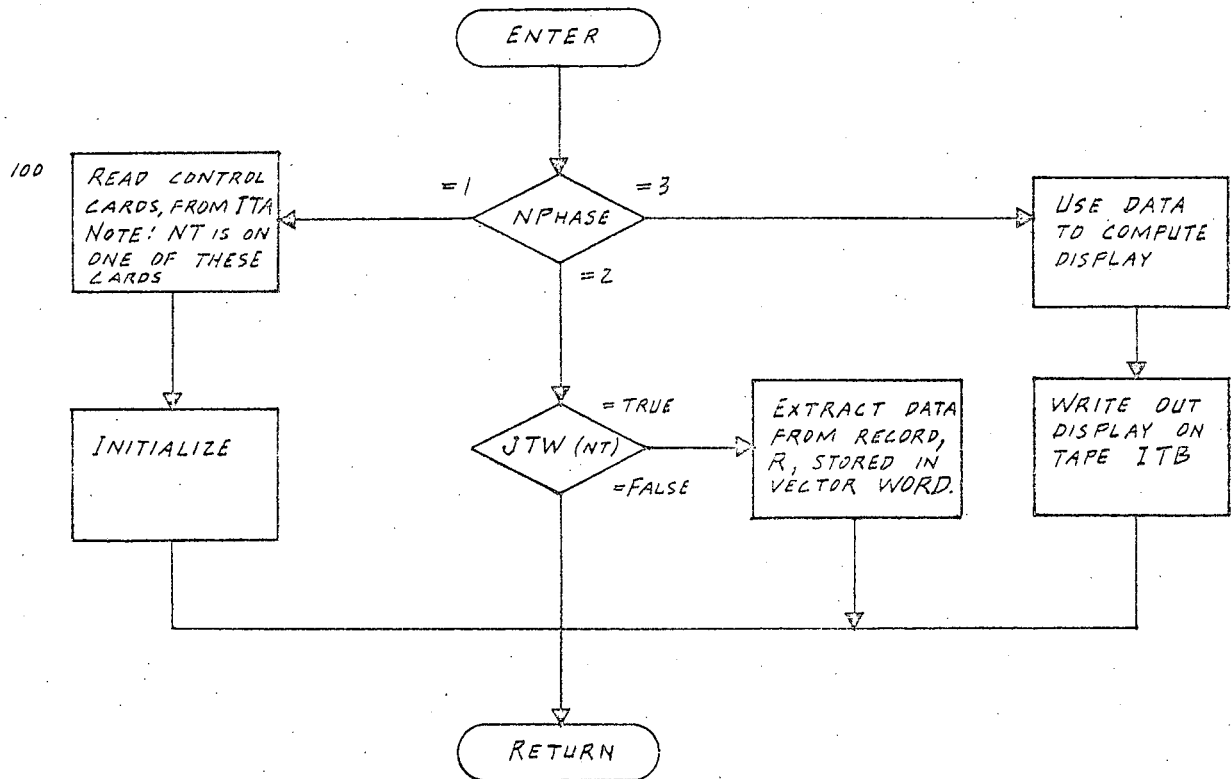
MUB-2926

Fig. 12. Flow diagram of TAPE.



MUB-2927

Fig. 13. Flow diagram of SELECT.



MUB-2928

Fig. 14. Flow diagram of any BLOC.

II.2. Input-Output

A. Introduction

The early versions of SUMX were faced with two interdependent problems.

1. Problem one

FORTTRAN II I-0 is essentially one channel I-0. FORTTRAN II breaks physical records up into short (128 word) physical records. Obviously, a 3-channel buffered I-0 system, allowing physical records up to 1000 words in length, and compatible with FORTTRAN I-0 would be advantageous.

2. Problem two

According to the plan of SUMX, given in Sec. II.1, the subroutines (TAPE, SELECT, BLOC3, ..., BLOC15) should be independent of each other. It should make no difference to the functioning of the rest of SUMX which version of a particular subroutine is loaded into memory. However, without some sort of I-0 executive program, the plan would break down.

(a) Example: Version one of BLOC5 might require a scratch tape mounted on logical unit B3. Version two of BLOC5 might use an oscilloscope camera attached to channel C. BLOC9 might require a scratch tape mounted on logical unit B3. If version one of BLOC5 and BLOC9 were used simultaneously, trouble would result. If version two of BLOC5 were used, it could interfere with other subroutines doing I-0 on channel C.

3. Solution to problems one and two

A tape executive program, STX, has been written to be used with SUMX.

4. Notation

In the FORTTRAN statement WRITE OUTPUT TAPE 22, 35, List the number 22 is the FORTTRAN tape number. Let FTN be an abbreviation for the FORTTRAN tape number.

In the FORTTRAN statement WRITE OUTPUT TAPE ITX, 35, List the variable ITX will be called a symbolic FORTTRAN tape number or a symbolic FTN. If ITX = 22, then the two preceding statements would be equivalent and, at Berkeley, subroutine (IOU) would cause logical tape unit C2 to be written on in either case.

5. The tape executive program

Symbolic FTN's are used exclusively in SUMX. The following table holds at Berkeley LRL.

<u>Symbolic</u>	<u>FTN</u>	<u>Logical tape unit</u>	<u>Function</u>
ITA	2	A2	Monitor input tape
ITB	3	A3	Monitor output tape

Symbolic FTN's are also used to refer to devices other than tape units.

Symbolic FTN	Function of unit
ITA	Monitor input tape
ITB	Monitor input tape
ITP	Printer
ITR	Card reader
ITTV	Cathode ray oscilloscope

NOTE: For modification at other computer installations, this table may be found at the beginning of subroutine STX.

Normally, at a computer installation only a small number of tape units are available for use as input tapes, output tapes, or scratch tapes. The List of Tapes Available table (LTA) is located in subroutine STX.

Table II. List of tapes available (LTA).

*	CODE,,FTN	UNIT	PRIMARY FUNCTION
PZE	**, 11	B1	SCRATCH TAPE
PZE	**, 12	B2	SCRATCH TAPE
PZE	**, 13	B3	SCRATCH TAPE
PZE	**, 4	A4	SCRATCH TAPE
PZE	**, 23	C3	INPUT TAPE OR OUTPUT TAPE
PZE	**, 21	C1	INPUT TAPE OR OUTPUT TAPE
PZE	**, 15	B5	INPUT TAPE OR OUTPUT TAPE
PZE	**, 22	C2	INPUT TAPE OR OUTPUT TAPE
LTA PZE	**, 8	NUMBER OF UNITS IN LIST	

The LTA table is used by subroutines SGET and SCLAIM.

a. SGET. The FORTRAN statement

CALL SGET (ITX)

will cause the symbolic FTN, ITX, to be assigned a value. The value assigned will be the first available FTN from the LTA table.

For example, if a person were writing a new BLOC13 which required a scratch tape, he could include the statement

CALL SGET (IT13)

in the initial portion of his coding. Subsequently, all input-output statements concerning the scratch tape for BLOC13 would use IT13. For example,

WRITE TAPE IT13, LIST

b. SCLAIM

The FORTRAN statement

CALL SCLAIM (ITX)

will reserve the tape unit which has an FTN = ITX.

For example, in subroutine TAPE, the tape unit, ITC, upon which the summary tape is to be mounted, is specified by control card. Suppose 21.0 were written in the appropriate place on the TAPE control card. This would cause ITC to have the value 21. Then, the statement

CALL SCLAIM (ITC)

would cause logical unit C1 to be reserved for the summary tape. In the remainder of subroutine TAPE, the symbolic FTN (ITC) is used whenever the summary tape is manipulated.

c. SCLEAR

The FORTRAN statement

CALL SCLEAR (ITX)

will cause all I-0 being done on the data channel, to which ITX is assigned, to be completed.

For example, suppose ITX = 22 (thus ITX refers to unit C2) and some non-FORTRAN I-0 were going on in channel C. Suppose the user wished to make the statement

READ TAPE ITX, LIST.

As soon as channel C was disconnected, this statement would be executed, probably destroying the contents of the channel C indicators before they could be used by the non-FORTRAN I-0 routine which requires them.

If the statements

CALL SCLEAR (ITX)

READ TAPE ITX, LIST

were used, the subroutine SCLEAR would ensure that any current I-0, on the channel associated with ITX, was "safely completed" before the READ TAPE statement was executed.

Rule. Always precede a group of FORTRAN I-0 statements, which use the symbolic FTN ITX, with the statement

CALL SCLEAR (ITX)

Rule. Always precede a group of FORTRAN statements, which involve the cathode ray oscilloscope, with the statement

CALL SCLEAR (ITIV).

6. The Subroutines S1R, S2R, S3R

a. Purpose: These are the SUMX read routines. They read binary records, as described in Sec. I.1.

b. Usage: Subroutine S2R will be illustrated.
The FORTRAN statement

CALL S2R (IND, DATA, NWORD, ITX)

causes one physical record to be moved into vector DATA,
where NWORD = number of words moved (i.e., the length of the record),
ITX = symbolic FTN or a FTN,
IND = 3 for 5 unsuccessful attempts to read a record,
= 4 for a record of all zeros,
= 5 for an end-of-file, and
= 2 for normal operation.

NOTE: These subroutines need not be preceded by the statement

CALL SCLEAR (ITX).

c. Restrictions: Each of these routines will use 22138 words of memory.

S1R is reserved, during the operation phase, for subroutines TAPE and BLOC9.

No more than one subroutine should use S2R (or S3R) during the operation phase of SUMX because the buffer block can only hold one record at a time.

7. The Subroutines S1W, S2W, S3W

a. Purpose: These are the SUMX write routines. They write binary records, as described in Sec. I.1.

b. Usage: Subroutine S2W will be illustrated.

The FORTRAN statement

CALL S2W (IND, DATA, NWORD, ITX)

causes the first NWORD words of the vector DATA to be written on the tape unit which correspond to symbolic FTN ITX,
where IND = 1 means 1 to 5 blank error gaps have been written,
= 2 means normal operation,
= 3 means bad tape,
= 4 means the physical end-of-tape has been found.

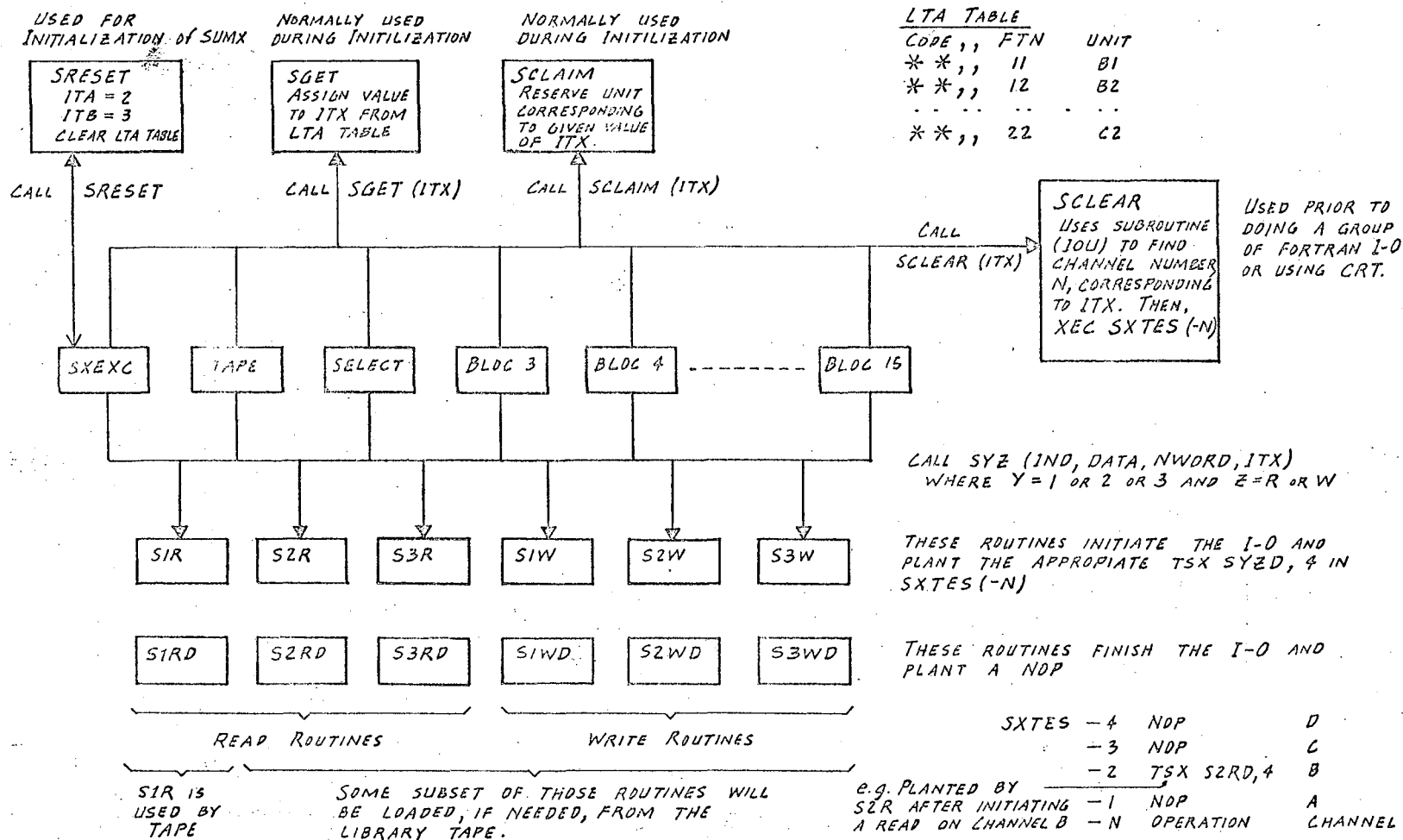
c. Restrictions: Each of these routines will use 2178g words of memory.

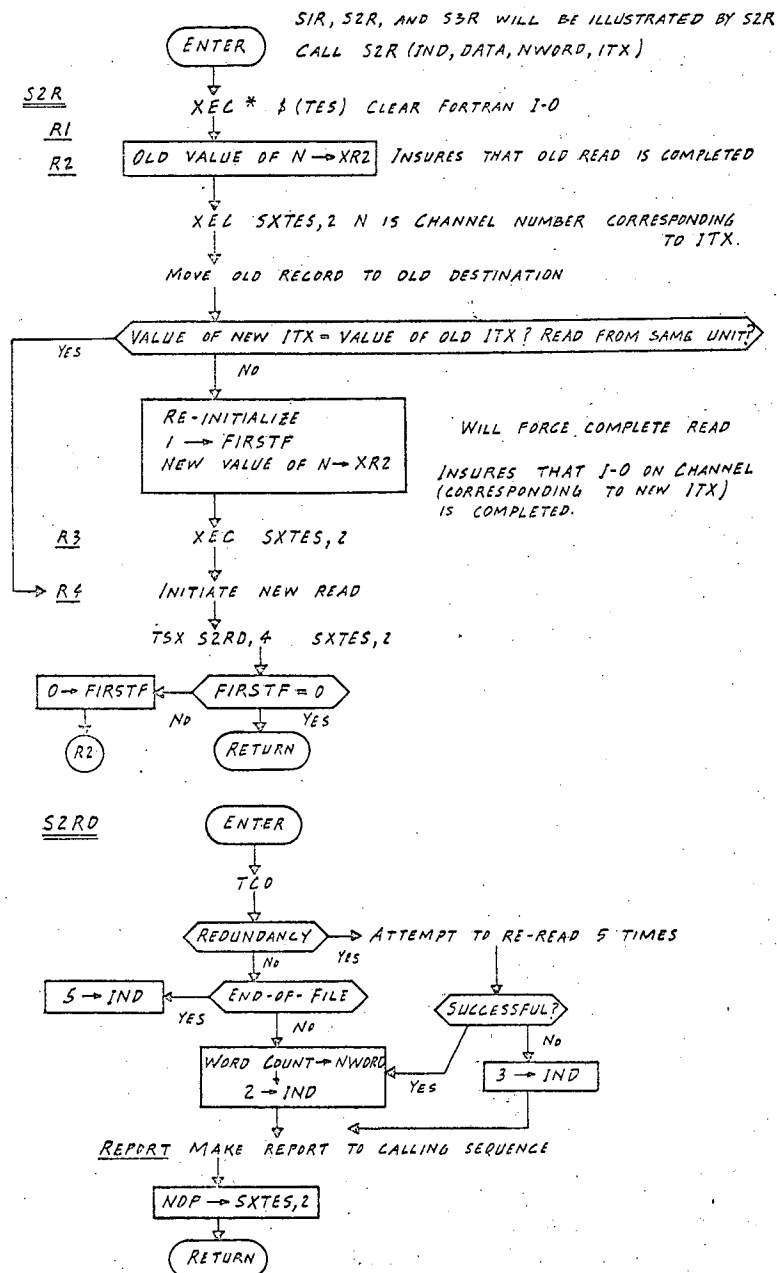
NOTE: These subroutines need not be preceded by the statement

CALL SCLEAR (ITX).

8. The General Plan of the Tape Executive Program, STX

The general plan of STX is illustrated in Fig. 15. The input and output subroutines are given in Figs. 16 and 17, respectively.





MUB-2930

Fig. 16. Flow diagram of S1R, S2R, and S3R.

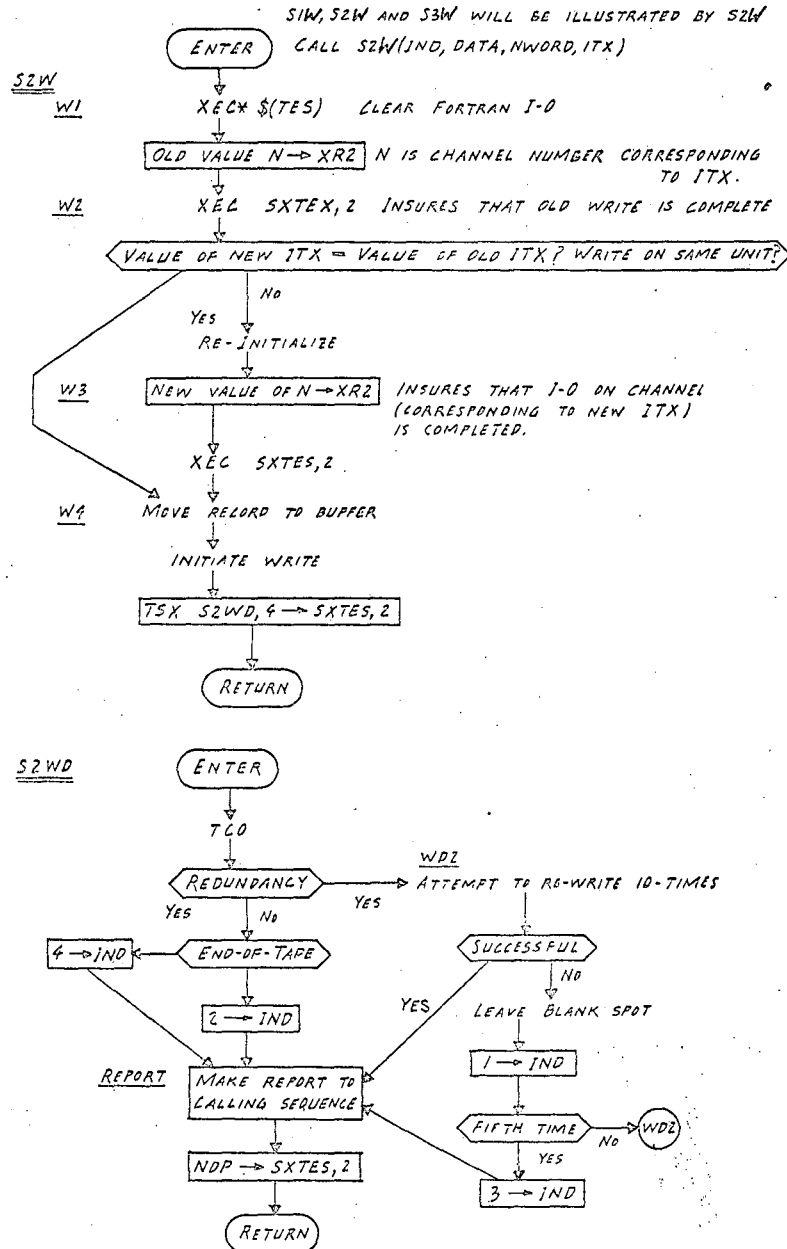


Fig. 17. Flow diagram of S1W, S2W, and S3W.

II.3. Subroutines TAPESR and BL3SR

A. TAPESR

TAPESR is a subroutine of subroutine TAPE. It is either a FORTRAN subroutine written by the user, or it is a dummy subroutine.

1. Purpose

The purpose of TAPESR is to allow the user to compute data, which is not on the summary-tape records, but which is a function of data that is on the summary-tape records.

2. Usage

The usage of TAPESR is illustrated in Fig. 18.

B. BL3SR

BL3SR is a subroutine of BLOC3. It is either a FORTRAN subroutine written by the user, or it is a dummy subroutine.

1. Purpose

The purpose of BL3SR is the same as that of TAPESR. It should be pointed-out, however, that BL3SR may be executed, if the user wishes, only for those records which satisfy TEST NT. TAPESR, on the other hand, will be executed for all records on the summary tape. This may be seen by consulting Fig. 3.

2. Usage

a. If the statement

SUBROUTINE BL3SR (N)

were substituted for the statement

SUBROUTINE TAPESR (N)

in the example on the preceding page, we would have an example of the use of subroutine BL3SR.

On exit from BL3SR, BLOC3 will write an N-word data record. However, the FORTRAN statement,

INWORD = K

(appearing in either TAPESR or BL3SR), will cause BLOC3 to write a K-word data record.

```

C      EXAMPLE OF USE OF THE SUMX SUBROUTINE TAPESR
C
C      SUPPOSE      EACH RECORD IS 100 WORDS LONG ON SUMMARY TAPE
C      AND 14 TH WORD OF EACH RECORD IS X (WHEN RECORD IS READ
C      INTO COMPUTER, X WILL BE IN WORD(14)).
C
C      WANT LOG OF X
C
C      SOLUTION  WRITE FOLLOWING PROGRAM (SUMX CDE STATEMENTS ARE
C      AVAILABLE IN ROOM 346C).
C      PUT PROGRAM IN FRONT OF SUMX BINARY DECK
C      NOW, USE SUMX JUST AS AS THOUGH LOG OF X WERE 114 TH
C      WORD OF EACH EVENT.
C
C      SUBROUTINE TAPESR(N)
C      NOTE - N IS THE TOTAL NUMBER OF WORDS IN THE RECORD THAT HAS JUST
C      BEEN READ INTO THE VECTOR WORD.
C      SUMX CDE STATEMENTS      10 DEC 63
C      COMMON      TEMP      , WORD      , CN
C      DIMENSION  TEMP(1000) , WORD(1000) , CN(150) ,
D      , ITEMP(1000) , IWORD(1000) , HOLLER(12) , JTW(18) ,
D      , IBOUT(1000) , BOUT(1000) , CONTRL(12) , NBF(15)
C      EQUIVALENCE(TEMP, ITEMP), (WORD, IWORD), (WORD, BOUT), (WORD, IBOUT),
C      E(CN, ITA), (CN(2), ITB), (CN(3), ITC), (CN(6), ITTAPE), (CN(8), JGOOD),
C      E(CN(9), INWORD), (CN(10), LMODE), (CN(11), NPVB8), (CN(12), HOLLER),
C      E(CN(24), IPASS), (CN(25), IPAGE), (CN(26), JTZERO), (CN(27), JTW),
C      E(CN(45), CONTRL), (CN(57), NBF), (CN(72), NB), (CN(73), ITP),
C      E(CN(74), ITR), (CN(75), ITTV)
C      END SUMX CDE .....
C      WORD(114)=LOGF(WORD(14))
C      RETURN
C      END

```

Fig. 18. Example of use of SUMX subroutine TAPESR

II.4. A Sample BLOC

The following sample BLOC illustrates the basic structure that all BLOCs must possess; the display is of a very trivial nature, namely to compute $\sum e^{\chi_i}$, where the χ_i are the contents of the L th word of those records (on a summary tape) for which test NT happens to be true. A trivial problem was chosen in the hope that the details of the computation would not obscure the basic structure.

Control cards for the following BLOC might look as follows:

BLOC13			
8.0		5.0	

Col. 1 10 11 20

 L NT

An actual BLOC is shown as an example in Fig. 19.

```

* LABEL
CB13XXX
C THIS SAMPLE BLOC13 IS SUPPOSED TO ILLUSTRATE THE BASIC
C STRUCTURE OF ALL BLOCs.
C EXAMPLE - COMPUTE SUMATION OF EXP(WORD(L)) OF THOSE SUMMARY TAPE
C RECORDS FOR WHICH TEST NT IS TRUE.
C
C SUBROUTINE BLOC13(NPHASE)
C ALWAYS INCLUDE STANDARD SUMX COMMON; DIMENSION;
C EQUIVALENCE STATEMENT.
C SUMX CDE STATEMENTS. 10 DEC 63
C COMMON TEMP , WORD , CN
C DIMENSION TEMP(1000) , WORD(1000) , CN(150),
D ITEMP(1000) , IWORD(1000) , HOLLER(12) , JTW(18),
D IBOUT(1000) , SOUT(1000) , CONTRL(12) , NPF(15)
C EQUIVALENCE(TEMP,ITEMP),(WORD,IWORD),(WORD,BOUT),(WORD,IBOUT),
E(CN,ITA),(CN(2),ITB),(CN(3),ITC),(CN(6),ITTAPE),(CN(8),JG000),
E(CN(9),INWORD),(CN(10),LMODE),(CN(11),NPV88),(CN(12),HOLLER),
E(CN(24),IPASS),(CN(25),IPAGE),(CN(26),JTZERO),(CN(27),JTW),
E(CN(45),CONTRL),(CN(57),NBF),(CN(72),NB),(CN(73),ITP),
E(CN(74),ITR),(CN(75),ITTV)
C END SUMX CDE
C FIRST STATEMENT TELLS WHICH PHASE IS TO BE PERFORMED.
C GO TO(100,200,300),NPHASE
C
C INITIALIZATION PHASE - ENTERED ONCE / PASS.
C READ CONTROL CARDS - ITA DENOTES MONITOR INPUT TAPE.
C PRECEDE INPUT STATEMENT BY CALL SCLEAR(ITA) TO INSURE THAT
C CHANNEL CORRESPONDING TO ITA IS CLEARED.
C
100 CALL SCLEAR(ITA)
C READ INPUT TAPE ITA,101,XL,NT
101 FORMAT (2F10.4)
C CONVERT CONTROL WORDS FROM FLOATING PT. TO FORTRAN
C INTEGERS. THE .000001 TAKES CARE OF ROUNDOFF ERROR
C
L=XL+.000001
NT=XNT+.000001
C SET SUMATION OF EXPONENTIALS = 0.
C SUMEXP=0.0
C INITIALIZATION FINISHED-RETURN TO EXECUTIVE PROGRAM.
C RETURN
C OPERATION PHASE -ENTERED ONCE PER RECORD.
C
C A RECORD ,R, (FROM THE SUMMARY TAPE HAS BEEN PUT INTO
C THE VECTOR WORD. THE N TH WORD OF THE RECORD HAS BEEN
C PUT INTO WORD(N).
C
C IN OPERATION PHASE OF BLOC13 THE EXP OF WORD(L) IS
C COMPUTED AND ADDED TO SUMEXP IF AND ONLY IF TEST NT
C IS TRUE(WHEN APPLIED TO THE RECORD).
C
C THE TRUTH VALUE OF TEST NT IS FOUND IN JTW(NT)
C
200 NT=NT
C NECESSATATED BY FORTRAN IDIOSYNCRACY
C IF(JTW(NT))202,201,202
201 L=L
C SUMEXP=EXPF(WORD(L))+SUMEXP
C RETURN TO EXECUTIVE PROGRAM-FOR NEXT RECORD
C
C RETURN
C FINAL PHASE -END OF SUMMARY TAPE HAS BEEN
C REACHED - NO MORE RECORDS - DO FINAL COMPUTATIONS (IF ANY)
C AND OUTPUT.
C
C GET CURRENT PAGE NUMBER
C 300 IPAGE=IPAGE+1
C OUTPUT INFORMATION - ITB DENOTES MONITOR OUTPUT TAPE.
C PRECEDE OUTPUT STATEMENTS BY CALL SCLEAR(ITB) TO INSURE
C THAT CHANNEL CORRESPONDING TO ITB IS CLEARED.
C CALL SCLEAR(ITB)
C WRITE OUTPUT TAPE ITB,302,NT,IPAGE
C WRITE OUTPUT TAPE ITB,303,L,SUMEXP
302 FORMAT(1H1,5X,19HEXAMPLE-BLOC13 NT= 13,54X,5HPAGE 12)
303 FORMAT(10X,24H SUMMATION OF EXP(WORD( 14, 3H))=E10.3 )
C RETURN
C END

```

Fig. 19. A sample BLOC.

ACKNOWLEDGMENT

I would like to acknowledge the help and suggestions I received from G. Kalbfleisch, R. Karpinsky, M. Leavitt, and S. Wojcicki. Ronald Ross and William Humphrey set the task of writing the original ancestor of SUMX, which made ordered lists from subsets of records. Arthur Rosenfeld suggested that the original version be extended to make histograms and ideograms, and he was the first to utilize the potential of the extended program. Joe Hanna programmed BLOC5 and BLOC6 and was extremely helpful in the implementation of the three-phase BLOC structure of the present SUMX.

Many thanks are due to the numerous people in the Alvarez Group who gave me much advice and assistance.

This work performed under the auspices of the U. S. Atomic Energy Commission.

III. Appendices

A. Glossary

$C(\text{WORD } N)$: the contents of WORD N . For example, if the fifth word of each logical record is χ , then $C(\text{WORD } 5) = \chi$.

Data record: a record that contains data, in contrast to the record used for an identification label.

Decimal mode: floating point.

Entire run: see Sec. I.0.

Event: see Introduction.

File: a group of physical records (on a magnetic tape) separated from other groups of records by special marks and gaps.

Hollerith characters: any characters available on IBM key-punch machines, with the exception of the 8-4 punch. Hollerith characters are used for making comments.

Integer mode: fixed point.

Label: see Sec. I.1.

$L(\chi)$: the location of χ in a logical record. For example, if χ is the fifth word of a logical record, then $L(\chi) = 5$ (written 5.0 in SUMX).

Logical record: a group of words which contains the information that describes one event (or one person or one transaction or one item).

Multiplicity: see Secs. I.4 and I.5.

Pass: see Sec. I.0.

Physical record: a group of words (on a magnetic tape) separated from other groups of words by special marks and gaps.

Record: a logical record or a physical record (depending on context).

Run: see Sec. I.0.

Summary tape: defined in Introduction; requirements are given in Sec. I.1.

Test: see Sec. I.2.

Test $K(R)$: Test K applied to the current logical record, R .

B. Operation of SUMX at LRL Berkeley

Portions of SUMX are on the FORTRAN II library tape. List L (see below) is a list of all subroutines on the library tape. The subroutines TAPE,

SELECT, BLOC3, ..., BLOC9 mentioned in List L are explicitly documented in the first section of this report.

1. Loading SUMX into the Computer Memory

The List-B (see Fig. 20) binary decks are used to control the loading of the List L subroutines into the computer memory.

A 3-card binary deck labeled SUMX00 to SUMX02 will cause all SUMX subroutines on the library tape to be loaded into memory.

Since all these subroutines will not fit into memory simultaneously, the loading of unwanted subroutines must be suppressed. Dummy binary decks are used for this purpose. For example, suppose the user did not need BLOC7, BLOC8, and BLOC9. His deck would be the following:

Job Card
Monitor ID card
* XEQ card
SUMX 63 (00-02) deck
B7DUMMY (00-01) deck
B8DUMMY (00-01) deck
B9DUMMY (00-01) deck
* DATA card
SUMX control cards (see Sec. I.0)
End-of-file card.

a. List S Subroutines

List S (see Fig. 20) describes SUMX subroutines which are not on the Library tape. These subroutines are usually special versions of the List L subroutines (Fig. 20). The number of subroutines appearing in List S is expected to increase. Binary decks for these subroutines are available. Accompanying each subroutine appearing in List S is a reference to more complete documentation of that subroutine.

To use these subroutines, simply insert their binary decks after the * XEQ card and before the SUMX63 (00-02) deck.

C. Example of a Logical Record

1. Purpose

This example is referred to in other sections of this report merely to provide a handy illustrative example for those sections.

2. Example

This example is a detailed version of the example given in the introduction; it illustrates a typical logical record on a summary tape.

THE SUMX 3 SYSTEM APRIL 1964

LIST B BINARY
BINARY DECKS (USED ON A2) TO CONTROL LOADING OF LIST L SUBROUTINES
FROM THE LIBRARY TAPE
INSERT A SUBSET OF THESE DECKS BETWEEN ID CARD AND DATA CARD

LIBRARY TAPE	DECK	CARDS	DESCRIPTION
SUMX63	SUMX	3	ALWAYS USED
B3DUMY	BLOC3	2	DUMMY-DELETES LOADING OF REAL BLOC3 FROM LIB TAPE
B4DUMY	BLOC4	2	DUMMY-DELETES LOADING OF REAL BLOC4 FROM LIB TAPE
B5DUMY	BLOC5	2	DUMMY-DELETES LOADING OF REAL BLOC5 FROM LIB TAPE
B6DUMY	BLOC6	2	DUMMY-DELETES LOADING OF REAL BLOC6 FROM LIB TAPE
B7DUMY	BLOC7	2	DUMMY-DELETES LOADING OF REAL BLOC7 FROM LIB TAPE
B8DUMY	BLOC8	2	DUMMY-DELETES LOADING OF REAL BLOC8 FROM LIB TAPE
B9DUMY	BLOC9	2	DUMMY-DELETES LOADING OF REAL BLOC9 FROM LIB TAPE

LIST L LIBRARY
C SUBROUTINES OF SUMX 63 ON THE LRL FORTRAN 2 LIBRARY TAPE
NOTE-- A (C,*) IN COLUMN ONE DENOTES A (FORTRAN,FAP) SUBROUTINE

NAME OF SUBROUTINE	APPROX. LENGTH OF DECK	IN DECK	DECK IN	COMMENTS
* * * * *	* * * * *	* * * * *	* * * * *	* * * * *
* * * * * ADDITIONAL COMMENTS (IF NECESSARY) * * * * *				

CSXE163 SXEXC 915 EXECUTIVE ROUTINE. USES STX,DECMPL.

CTAP263 TAPE 724 READS ONE PHYSICAL RECORD (AT A TIME) FROM THE
C SUMMARY TAPE INTO A STANDARD LOCATION (WORD) IN MEMORY. USES STX,SIR, SIDR.
C USER MAY WRITE HIS OWN SUBROUTINE TAPESR(N) - A DUMMY TAPESR IS PROVIDED.

CSEL363 SELECT 2858 MAKES TESTS ON RECORD (EVENT) TO DETERMINE WHICH
C SUBSETS THE RECORD MAY BELONG. TESTS T1-T20 DETERMINE SUBSETS S1 T O S20
C USES DECODE(SX2063) AND WRTOUT AND BCDCON
C READS NEW CONTROL CARDS ONLY

CBL3001 BLOC3 647 WRITES A NEW SUMMARY TAPE FROM (MODIFIED) C ONTENTS
C OF BLOC. USER MAY WRITE HIS OWN SUBROUTINE BL3SR(N). USES SIW,STX,TA PESR

CBL41C3 BLOC4 2976 MAKES UP TO TEN RESOLUTION FUNCTIONS AND/OR
C GAUSSIAN IDEA-GRAMS
C PRINTER DISPLAY - 110 BINS - MULTIPLICITY=5 . USES AAERF2

CBL51D3 BLOC5 6703 MAKES 40 ONE-DIM WEIGHTED HISTOGRAMS.(INT OR DEC)
C PRINTER DISPLAY - 110 BINS - MULTIPLICITY=3 .

CBL6263 BLOC6 2738 MAKES FIVE TWO-DIM HISTOGRAMS(INT OR DEC)
C PRINTER DISPLAY - 15X20 BINS-MULTIPLICITY=1 .

CBL71B3 BLOC7 5052 MAKES UP TO 20 SCATTER DIAGRAMS
C CRT DISPLAY -MULTIPLICITY=3 . MAKES UP TO 2 S KINS
C USER MAY WRITE HIS OWN SKIN - KINEMATICAL SKIN IS PROVIDED.
C USES STX,SIR,SIW,SDNORM AND CRT PACKAGE -DOES NOT REQUIRE GRAPH.
C NOTE. LENGTH OF CRT PACKAGE = 596

Fig. 20(a). The SUMX system.

C NOTE. LENTH OF GRAPH =2000

CBL8163 BLOC8 631 MAKES A LIST OF WRDS(1),(2),(3) (ROLL,FRM,B EAM TR)
C AND UP TO 12 ADDITIONAL WORDS / RECORD

CBL9163 BLOC9 4052 MAKES AN ORDERED LIST OF UP TO 1000 WORDS
C UP TO 2 ASSOCIATED WORDS MAY BE INCLUDED IN THE LIST.
C USES STX,S1R,S1W

*STX001 SRESET 250 TAPE EXECUTIVE ROUTINES.
* SXTES MAKE AUTOMATIC SCRATCH TAPE ASSIGNMENTS AND
* SCLEAR TAKE CARE OF ASYNCRONUS I-O.
* SCLAIM
* SGET

*SID001 SIDR 140 READS BCD LABEL ON SUMMARY TAPE.

*S1R1B3 S1R 1155 READS BINARY RECORDS UP TO 1001 WORDS IN LE NGTH
* RESERVED FOR USE BY TAPE,BLOC7 AND BLOC9

*S1W001 S1W 1144 WRITES BINARY RECORDS UP TO 1001 WORDS IN L ENGTH
* MAY BE USED BY ANY SUBROUTINE,IF SHORT ON STORAGE,USE S1W, OTHERWISE
* USE S2W OR S3W.

*SX1242 DECM PG 34 SUBROUTINE OF SXEXC
*DEC163 DECODE 41 SUBROUTINE OF SELECT -NEW CONTROL CARDS
CSX2063 WRTOUT 40 WRITES OUT CONTROL CARDS ON ITB
C7B2242 SKIN 597 SUBROUTINE OF BLOC7 - MAKES KINEMATICAL ENV ELOPE
*AAERF2 FREQ 61 SUBROUTINE OF BLOC4
*7B1242 SDNORM 30 SUBROUTINE OF BLOC7
*S2R1B3 S2R 1155 READS BINARY RECORDS UP TO 1001 WORDS IN LE NGTH
* MAY BE USED BY ANY SUBROUTINE.
*S3R1B3 S3R 1155 READS BINARY RECORDS UP TO 1001 WORDS IN LE NGTH
*S2W063 S2W 1144 WRITES BINARY RECORDS UP TO 1001 WORDS IN L ENGTH.
*S3W063 S3W 1144 WRITES BINARY RECORDS UP TO 1001 WORDS IN L ENGTH.
*10DUMY BLOC10 DUMMY SUBROUTINES
* TO
*15DUMY BLOC15
*TSRDUM TAPESR
*3SRDUM BL3SR

LIST S SPECIAL

C SUBROUTINES OF SUMX 63 FOR WHICH BINARY DECKS ARE AVAILABLE
C THESE DECKS MAY BE INSERTED IN FRONT OF THE BINARY DECK SUMX63

CTAP3B3 TAPE 1869 PUTS ONE LOGICAL RECORD (AT A TIME) INTO
C THE VECTOR WORD (BOUT). ALLOWS MANY LOGICAL RECORDS/PHYSICAL RECORD.
C USES STX,S1R,SIDR
C THE CONTROL CARDS ARE THE SAME AS THOSE OF TAP263
C TAP3B3 WILL PROCESS TAPES WRITTEN IN THE STANDARD FORMAT OR
C THE FORMAT DESCRIBED IN ALVAREZ PROGRAMMING NOTE P-54 (PAGE 27)

CTAP5A3 TAPE 215 TAPE SIMULATOR - THIS SUBROUTINE SIMULATES
C AN INPUT TAPE REEL. THIS SUBROUTINE MAY BE USED TO CHECK OUT OTHER

Fig. 20(b). The SUMX system.

C PORTIONS OF SUMX WITHOUT THE BOTHER OF USING AN INPUT TAPE. TAP5A3
C SIMPLY PUTS A LOGICAL RECORD IN WORD, EACH TIME IT IS CALLED.
C TO SIMULATE N LOGICAL RECORDS, WRITE N.O.(BETWEEN COLS.1 AND11) ON CO NTROL
C CARD.
C THE FORMAT OF THE SIMULATED RECORDS IS DESCRIBED IN THE SUMX SAMPLE RUN

CSEL2N2 SELECT 3924 MAKES TESTS ON RECORD (EVENT) TO DETERMINE WHICH
C SUBSETS THE RECORD MAY BELONG. TESTS TO-T18 DETERMINE SUBSETS SO T O S18
C USES DECODE.
C THIS VERSION OF SELECT IS DESCRIBED IN ALVEREZ MEMO 389.

*SE1262 DECODE 47 SUBROUTINE OF SELECT

CBS4063 BLOC4 12422 MAKES UP TO 30 GAUSSIAN IDEA-GRAMS (AND RESO LUTION-
C FUNCTIONS)/ PASS. CRT-DISPLAY - 100 BINS - MULTIPLICITY=5
C USES GRAPH (APPROX 2000 WORDS)
C THE CONTROL CARDS ARE DESCRIBED BY McLEAVITT IN P-44

CBS5063 BLOC5 10140 MAKES UP TO 40 ONE-DIM HISTOGRAMS (INTEGER O R DEC.).
C CRT-DISPLAY - 110 BINS - MULTIPLICITY=3
C USES GRAPH (APPROX 2000 WORDS)
C USES SKETCH - USER MUST SUPPLY HIS OWN SKETCH (IT MAY BE A DUMMY)
C THE CONTROL CARDS ARE DESCRIBED BY McLEAVITT IN P-44

CBS5363 BLOC5 2126 MAKES UP TO 5 HISTOGRAMS
C PRINTER DISPLAY - 110 BINS -MULTIPLICITY=5
C SEE MEMO P-44

CBL6363 BLOC6 10785 MAKES UP TO 15 2-DIMENSIONAL HISTOGRAMS
C MAXIMUM LENGTH AND HEIGHT ARE 15 AND 40
C ALL TOTAL ARE IN FIXED POINT ALLOWING FOR WEIGHTED ENTRIES
C SEE MEMOS 389 AND P-44

CBS7263 BLOC7 5166 SAME AS BL71B3, EXCEPT THAT...
C POINT FOR MULTIPLICITY 1= .
C POINT FOR MULTIPLICITY 2= T
C POINT FOR MULTIPLICITY 3= X
C USES SAME CONTROL CARDS AS BL71B3

CB15063 BLOC15 MAKES A SELECTIVE DUMP OF THE SUMMARY TAPE
C MAY BE USED AS AN ACCOUNTING AID.

Fig. 20(c). The SUMX system.

<u>n</u>	<u>c (word n)</u>	<u>typical value</u>	<u>mode</u>
1.	Survey serial No.	2375	integer
2.	Married (0 = no, 1 = yes)	1	integer
3.	Sex (0 = male, 1 = female)	0	integer
4.	Age in years	27.3	decimal
5.	No. of children	1	integer
6.	No. of years of school	17	integer
7.	Height in meters	1.87	decimal
8.	Salary in dollars	710.00	decimal
9.	Weight in kilograms	72.93	decimal
10.	Average monthly liquor bill	4.30	decimal
11.	Age of spouse (if any)	26.3	decimal
12.	Weight of 1st child (if any)	30.64	decimal
13.	Weight of 2nd child (if any)	0.0	decimal

D. SUMX Common, Dimension and Equivalence Statements

A standard set of common, dimension, and equivalence statements appears at the beginning of most SUMX subroutines. These statements, the SUMX CDE statements, may be followed by dimension and equivalence statements peculiar to the subroutines in which they appear. The SUMX CDE statements are used for intercommunication between subroutines. (See Fig. 21.)

The following is a brief explanation of each of the entries in the listing in Fig. 21.

(a) TEMP, ITEMP

TEMP(1), to TEMP(1000) are used to store intermediate computations temporarily.

(b) WORD, IWORD, BOUT, IBOUT

WORD(1) to WORD(n), $n \leq 1000$, are filled by subroutine TAPE with the first n words of the current logical record, R.

BOUT is equivalent to WORD. BOUT is an historical remnant.

(c) CN, ICN

CN is a 150-word vector used to hold control words.

J CN(J)

- 1 ITA Symbolic FTN for the monitor input tape
 (ITA = 2 at Berkeley LRL).
- 2 ITB Symbolic FTN for the monitor output tape
 (ITB = 3 at Berkeley LRL).
- 3 ITC Symbolic FTN for the summary tape.
- 4 No longer used.
- 5 No longer used.
- 6 ITTAPE If a value (NTAPE) is not given for ITC on the subroutine
 TAPE control cards, then ITC is given the value of
 ITTAPE (ITTAPE = 15 at Berkeley LRL).
- 7 No longer used.
- 8 JGOOD Number of logical records that satisfy the main test, Test 1.
- 9 INWORD Number of words in the current logical record, R.
- 10 LMODE = 1 if summary tape has Binary ID label,
 = 2 if summary tape has BCD ID label.
- 11 NPVB8 A flag associated with the standard (List L) version of
 BLOC8.
- 12 HOLLER HOLLER is a 12-word vector. HOLLER contains the card
 image of the Hollerith control card.
- 24 IPASS = 0 if subroutine TAPE has read a normal record.
- 25 IPAGE The current page number. IPAGE is used during output.
- 26 JTZERO Always = 0 = true. JTZERO is the truth value of the
 "vacuous conditional statement," Test 0.

J CN(J)

- 27 JTW JTW is an 18-word vector.

JTW(NT) = 1, if test NT applied to the current record R
is false; otherwise, JTW(NT) = 0.

NOTE: Due to an historical oversight the vector JTW is limited to 18 words. However, it may safely be used as though it were a 30-word vector. It would require a reassembly of the entire SUMX program, before SELECT could handle over 30 tests.

45 CONTRL CONTRL is a 12-word vector. CONTRL contains the card image on the last control card read by subroutine SXEXC.

72 NBF NBF is a 15-word vector. NBF contains the list of BLOCs to be called by SXEXC.

72 NB NB is the length of the list in NBF.

73 ITP Symbolic FTN for the printer.

74 ITR Symbolic FTN for the card reader.

75 ITTV Symbolic FTN for the cathode-ray oscilloscope.

```

C      SUMX CDE STATEMENTS      10 DEC 63
COMMON      TEMP      WORD      CN
DIMENSION   TEMP(1000) , WORD(1000) , CN(150) ,
D           ITEMP(1000) , IWORD(1000) , HOLLER(12) , JTW(18) ,
D           IBOUT(1000) , BOUT(1000) , CONTRL(12) , NBF(15)
EQUIVALENCE(TEMP,ITEMP),(WORD,IWORD),(WORD,BOUT),(WORD,IBOUT),
E(CN,ITA),(CN(2),ITB),(CN(3),ITC),(CN(6),ITTAPE),(CN(8),JGOOD),
E(CN(9),INWORD),(CN(10),LMODE),(CN(11),NPVBS),(CN(12),HOLLER),
E(CN(24),IPASS),(CN(25),IPAGE),(CN(26),JTZERO),(CN(27),JTW),
E(CN(45),CONTRL),(CN(57),NBF),(CN(72),NB),(CN(73),ITP),
E(CN(74),ITR),(CN(75),ITTV)
C      END SUMX CDE

```

Fig. 21. SUMX common, dimension, and equivalence statements.

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

- A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or
- B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

