ANL-6521
Mathematics and Computers
(TID-4500, 17th Ed.)
AEC Research and
Development Report

ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois

A GENERAL 704 SERVICE LIBRARY

by

P. S. Bagus and C. C. J. Roothaan,
Solid State Science Division, Argonne National Laboratory
and Laboratory of Molecular Structure and Spectra,
Physics Department, The University of Chicago

and

W. S. Worley, Jr., and M. C. Wunderlich,
Laboratory of Molecular Structure and Spectra,
Physics Department, The University of Chicago

March 1962

Operated by The University of Chicago
under
Contract W-31-109-eng-38

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or 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 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. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

TABLE OF CONTENTS

	<u>Page</u>
INTRODUCTION.	5
PREPARATION AND USE OF SELF-LOADING TAPE.	7
ENTRIES TO EXECUTE A SERVICE LIBRARY ROUTINE	8
Entry from the Main Program	8
Manual Entry from the Machine Console.	9
Manual Entry Using a Single Call Card.	9
ERRONEOUS CODE WORDS.	12
ALARMS.	12
SNEAK-ON AND SNEAK-OFF.	13
SELF-STARTS.	17
FIRST CODE WORDS	17
FUNCTIONS AND USE OF THE LIBRARY ROUTINES.	18
Common Routines.	18
SERVICE LIBRARY OUTPUT.	22
EFFECTIVE ADDRESSES	23
END-OF-FILE MARKS	23
MULTIPLE CODE WORDS.	24
THE "STANDARD" SERVICE LIBRARY ROUTINES	24
Tape-handling Routines	24
Read Program Tape	26
Write Program Tape.	28
Write Data Tape.	30
Read Data Tape	32
Binary Card Handling Routines	35
Read Binary Cards	36
Write Binary Cards	37
SAP Information Handler	38
Dumping Routines.	42

TABLE OF CONTENTS

	<u>Page</u>
THE "SPECIAL" SERVICE LIBRARY ROUTINES	48
Print-out Routines	48
Octal Print-out	49
Decimal Print-outs	50
Floating-decimal Print-out	54
Eight-column Fixed-decimal Print-out	55
Ten-column Fixed-decimal Print-out	56
BCD Print-out Routine	56
Time Routines	59
Time Print-out or Prepare BCD	59
Time Integer Routine	60
CHANGING THE YEAR IN THE TIME AND DUMP ROUTINES	62
CONSTRUCTION OF LOAD CARDS	62
Single-step Load Cards	62
Double-step Load Cards	64
TABLE OF ALL CODE WORDS	65

LIST OF FIGURES

<u>No.</u>	<u>Title</u>	<u>Page</u>
1.	Schematic of the Single-step Load Cards	63
2.	Diagram of the Single-step Load Card to Read SAP Cards . .	64
3.	Schematic of the Double-step Load Cards	64
4.	Diagram of the Double-step Load Card to Initiate a Complete Octal Core Dump	65
5.	Dump Routine Code Words	65
6.	Read SAP Routine Code Word	66
7.	Binary Card Routines Code Words	66
8.	Data Tape Routines Code Words	66
9.	Program Tape Routines Code Words	66
10.	Print-out Routines Code Words	67
11.	Time Routines Code Words	67

LIST OF TABLES

<u>No.</u>	<u>Title</u>	<u>Page</u>
1.	Error Stops During Sneak-on and Sneak-off Procedures . . .	15
1A.	Error Stops During Self-loading Tape Sneak-on Procedure .	16
2.	First Code Words	18
3.	Output Conventions for Print-out, Dump, and Time Routines	22

A GENERAL 704 SERVICE LIBRARY

by

P. S. Bagus, C. C. J. Roothaan, W. S. Worley, Jr.,
and M. C. Wunderlich

INTRODUCTION

The Solid State Science Service Library at Argonne National Laboratory is a unified set of input, output, and diagnostic routines for the Argonne IBM 704 digital computer, which has a 32-K core, 8-K drum, floating point underflow button, and external chronolog clock. Modifications are possible for any 704 having at least an 8-K drum. The library is designed to give the programmer maximum ease in programming the input and output functions required by his program, without having to spend the time and effort coding new or "modifying" existing subroutines for this purpose. In addition, it provides for the rapid execution of the diagnostic and emergency dumps which are invariably necessary during the preliminary and debugging stages of a program. All the functions of the library may be initiated from the machine console, or may be called for by the main program.

In the description which is to follow, the term "Main program" or simply "Program" shall always designate the program utilizing the Service Library. The word "Routine" shall mean a routine incorporated into the Service Library to perform a specific or diagnostic function. An "Entry" refers to the process used to initiate the execution of a Library routine. The various entries will be discussed in a later section.

The seventeen routines of the Service Library are available to the programmer at the expense of:

- 1) the contents of the accumulator and MQ register;
- 2) the trapping mode;
- 3) the four magnetic drums; and
- 4) the first 8 or the first 2048 cells of core storage, depending on whether the library is used in the sneak-off or no-sneak-off mode, respectively.

More precisely, when using the library the four magnetic drums and the first 8 (or 2048) cells of core storage are off limits to the programmer; furthermore, each library entry destroys the contents of the accumulator and MQ register, and leaves the computer out of the trapping mode.

Otherwise, the entire computer, that is, the contents of cells 8-32767 (or 2048-32767 in the no-sneak-off mode), and all index registers and panel lights are the same after a library entry as before. (The only exception to the latter is the tape check light; however, owing to a much more reliable process, this light is not used to keep track of tape errors in the library tape handling routines.)

The no-sneak-off mode, to be discussed in more detail below, is recommended for the input-output operations of a checked-out program. It achieves a saving in execution time at the expense of 2040 additional cells of core storage.

The Library can be loaded from tape unit 1, either from the console or under program control. The loading process has the same effect on machine contents as a sneak-off library entry; that is, after the loading process, the library occupies the magnetic drums, the computer is out of the trapping mode, the contents of the accumulator and MQ register have been destroyed, and a bootstrap remains in cells 0-7, which permits future programmed library entries; otherwise the machine is undisturbed.

If, during the execution of a program, temporary storage on the magnetic drums is desirable, several areas may be used in a limited way. Cells 1428-2047 of drum 3 are used by the Service Library only during the dumping of a magnetic tape. Therefore, this space is free for use as long as no tapes are dumped. After a sneak-off entry, all of drum 4 may be used for temporary storage; however, the next library entry will destroy the contents of this drum. If still more space on the drums is desired for temporary storage, the library may be loaded by the main program only when a routine is needed. Between Library acquisitions, all four drums are available. The calling sequence to load the library is

PXD	0,0
SSM	
RTB	1
CPY	0
CPY	1
TSX	1,4

After the Library has been loaded, control will be transferred to the first cell following the sequence.

There are several reasons why the Service Library is particularly convenient to use. First of all, the programmer has all main input and output facilities available at all times. This obviates providing separate sub-routines for each type of input and output desired. Also, there is great operational convenience in having emergency routines which stand ready

to be used. This eliminates reading in auxiliary decks for octal or decimal dumps. Secondly, no space in the magnetic core, except cells 0-7, need be set aside for input and output subroutines; all the remaining 32760 cells are available for the main program. A typical Library output entry requires three cells, considerably fewer than a typical output subroutine. A third reason is the reliability of input and output operations. Any data or sections of program which are written to or read from external equipment are systematically tested by the use of logical sum checks. For example, when data are written onto magnetic tape by the Write Data Tape Routine, the first word of the record is the logical sum of the rest of the record. After the record has been written, it is immediately read to insure that it is correct. The logical sum is then tested whenever the data, or portions of the data, are read into the core by the Read Data Tape Routine. In the event of a sum check failure, a second reading is attempted. If the failure recurs, an "Alarm" (to be described later) is printed on the on-line printer informing the operator of the error and the proper corrective measure.

One of the most pleasant features of using the Library for input and output functions is that the condition of all indicators, lights, and registers, except the Accumulator and Multiplier-Quotient registers and the trapping mode, are identical before and after the execution of every Library routine. This obviates keeping track of these devices while using input and output subroutines.

The actual process of going onto the machine is also greatly simplified by the use of the Service Library. The Library can be loaded from a self-loading tape in a few seconds, and the main program may be called in from a "Program Tape" (a tape format found only within this Library, which combines the speed of magnetic tape with the reliability of SAP binary cards) in an equally short time. The net result is that from a cleared machine a computation may be in progress in less than a quarter of a minute.

PREPARATION AND USE OF SELF-LOADING TAPE

To read the Service Library from the self-loading tape, it is necessary to mount the tape on unit number 1 and press the load tape key. When the tape has been loaded, the on-line printer will inform the operator that the Library is in place, and in the event of a drum or tape malfunction it will tell the operator what has occurred and what corrective measure may be attempted. As mentioned before, the tape-loading process requires the first 8 cells of core storage, the 4 magnetic drums, and destroys the Acc, MQ, and Trap mode. It may be loaded while the machine is full without disturbing the contents of any cells or registers except these. The Library is distributed to users as a binary deck, which is used to write the

self-loading tape. The deck may be loaded with the load cards key, and the machine will come to a halt with the Library in place. Restarting the machine causes the tape to be written on unit number 1 and to be fully checked. Restarting the machine a second time will cause a copy of the binary deck to be punched (except for the first card, which is a load card). The process of writing the tape from the deck does not save the condition of the machine and will destroy sizable amounts of core storage. It should only be done on a cleared machine.

ENTRIES TO EXECUTE A SERVICE LIBRARY ROUTINE

A Service Library routine may be executed in any one of three ways: under program control by an entry from the main program; manually from the machine console; and manually from the console by the use of a single call card.

Entry from the Main Program

An entry from the main program consists of the instruction TSX 4,4 followed by one or more code words necessary to determine the Library function fully. The code word immediately following the TSX 4,4 instruction is known as the "First Code Word" and it alone determines which routine is to be executed. Table 2 lists all first code words, and all code words are illustrated in detail in the discussion of the individual routines. If more than a single code word is necessary to specify the operation completely, additional code words, called "Secondary Code Words," must be given.

Typical examples of programmed entries would be the following:

TSX 4,4	Initial jump instruction
PZE 5,4,1	Write data on tape number 5, after advancing 1 record
PZE 100,0,300	The data begin at cell 100 and end at cell 300
or another:	
TSX 4,4	Initial jump instruction
PZE 2,0,1	Print floating decimal numbers on and off line
PZE 400,6,410	First number is at cell 400, the last at cell 410;
etc.	after the printing, write 6 blank lines

After the routine is executed, the Library restores the condition of the machine (with the exceptions listed above) and returns control to the main program at the cell immediately following the final code word of the entry.* Notice that in both examples a secondary code word was necessary to specify the operation.

*Except in the self-start cases to be discussed later.

Manual Entry from the Machine Console

A manual entry consists of performing the following steps:

- 1) Press the Load Drum key. The machine will halt at cell 776 (octal). The four sense lights will be on and MQ cleared. The instruction register will contain HPR.
- 2) Set the machine into the manual mode by putting the manual-automatic switch down.
- 3) Enter the necessary code word into the MQ register from the keys.
- 4) Reset the machine to the automatic mode.
- 5) Press the start key to begin the routine.

If the routine requested requires more than a single code word (as those in the above example do), the machine will stop to receive each secondary code word as it is needed. Whenever the machine stops for a code word, the four sense lights will be turned on and the MQ register will be cleared. This signal is standard throughout the Library. Secondary code words are inserted by performing steps 2-5 listed above. An example of a (very) typical manual entry is the following:

- 1) Press the load drum key.
- 2) Insert the following into MQ from the keys:

<u>Prefix</u>	<u>Decrement</u>	<u>Tag</u>	<u>Address</u>
1	00000	0	00000

- 3) Reset the machine to automatic and start the computer.

An octal dump of the entire core has now been initiated.

Following the execution of a manually initiated routine, the machine halts at cell 3, and the instruction register contains the instruction HTR 2,4.*

Manual Entry Using a Single Call Card

There are two types of call cards, or load cards, which will be called "Single-step" and "Double-step" load cards, respectively. The cards are used in different manners, but their functions are the same: to provide a rapid method of supplying the first code word manually.

*Except in the self-start cases to be discussed later.

1) Single-step Load Card

The single-step load cards are used to provide the first code words for the main input routines. These routines (Read SAP Cards or SAP tape, Read Binary Cards, Read Program Tape) require only the first code word and are initiated by placing the load card in the reader and depressing the load cards key.

The number of single step load cards is limited to nine, and their functions have been definitely assigned. Entries from these cards are sneak-off entries. The cards are:

- 1) Read Binary Cards
- 2) Read SAP Cards
- 3) Read SAP Information from Tape 3
- 4) Read First Program from Tape 2
- 5) Read Second Program from Tape 2
- 6) Read Third Program from Tape 2
- 7) Read Fourth Program from Tape 2
- 8) Read Fifth Program from Tape 2
- 9) Read Sixth Program from Tape 2

The functions of these routines will be explained fully in the sections devoted to the individual routines.

A set of these load cards will be furnished along with the Service Library binary deck.

As can be seen from the functions of these load cards, a set of tape assignment conventions has been adopted. They are:

<u>Tape Unit</u>	<u>Tape Contents</u>
0 (=10)	Output from computations or dumps
1	Self-loading Service Library tape (It is always rewound and ready to be loaded.)
2	Program tape of the main program, possibly in several sections
3	Input data in SAP input format
4-9	Data tapes for intermediate results, tables, etc.

Of the above conventions, only the assignments of tape units 0 and 1 are rigidly fixed. The other tape units may be assigned in any way desired. The advantage in following the assignment conventions for units 2 and 3 is that the single-step load cards may be used.

2) Double-step Load Card

The double-step load card is also a single card, but the method of using it is slightly different. The card is placed in the reader; FIRST THE LOAD DRUM KEY IS PRESSED, AND THEN THE LOAD CARDS KEY.

Double-step load cards are also used to supply first code words. Their number is not limited, however, and a card may be made to supply any conceivable first code word. Instructions for making these and the single-step load cards can be found in the section on the contents of the load cards.

If the routine initiated by the double-step load card requires one or more secondary code words, they must be supplied as though the entry were a manual entry. The machine will halt, to wait for these secondary code words, with the sense lights on and the MQ register cleared, and each word is inserted by performing steps 2-5 given in the manual entry section.

It should be noted that the double-step load card has an advantage over the single-step card when it is to be used for diagnostic purposes. The double-step card does not disturb index register 4, whereas, the single-step card makes use of it.

After the execution of a routine initiated by a load card, the machine halts at cell 3 in exactly the same way as for a manual entry.*

*Except for the self-start and no-sneak-off cases to be discussed later.

ERRONEOUS CODE WORDS

In the event that an erroneous code word is supplied, either during a programmed or a manual entry, an alarm will inform the operator of the error. The corrective measure printed in the alarm will be to restart to insert the code word manually. If the machine is restarted, it will halt with the sense lights on and MQ clear to receive the correct code word. The correct code word is, of course, inserted as in a manual entry. The code words originally given will appear in the alarm print out. (See example of alarm print-out in the next section.) If the entry required one or more secondary code words, only the code words up to and including the illegal one will be printed. Therefore, in the cases of multiple code words, the LAST code word printed will be the ERRONEOUS ONE.* In all cases of illegal code word alarms, restarting the machine will cause the Library to wait for a code word to replace the last one printed in the alarm.

ALARMS

During the earlier parts of this description, it was stated that certain errors or malfunctions caused alarms to be printed out on the on-line printer. Within the Library is an elaborate system to detect malfunctions of external equipment and obvious human errors in the code words. An example of equipment malfunction would be the failure of a logical sum check test. The Library routines are stored on the drums, and the logical sum of that routine follows each routine. As the routine is read into its operating location, its logical sum is again computed, and if this sum does not agree with the sum on the drum, an error occurred during the reading process. An example of an obvious error in a code word would be the request to write data onto tape unit number 13 (which does not exist). An alarm fully describes the entry, error, and proper corrective measure. It is printed at the top of a page in the following format:

*There are three exceptions to this rule - in the Floating Decimal and the two Fixed Decimal print-out routines when the request is made to prepare and store the BCD characters of the converted numbers. (This will be explained in the sections about these routines.) In these cases neither code word by itself causes an alarm condition, but together they do. If this alarm occurs, both code words appear in the alarm print out, and restarting causes the machine to wait for the second code word (the last one printed out). The first code word may be replaced instead by inserting the code word into the MQ register and manually transferring to cell 1000. (Octal)

SERVICE LIBRARY ALARM

PROGRAM ENTRY FROM CELL 23432	(Octal address of the TSX 4,4 instruction)
or	
MANUAL ENTRY	(If the entry was manual)
+04 00000 00015	(Prefix, tag, decrement, and address of code words)
WRITE DATA TAPE	(Routine requested)
INCORRECT TAPE UNIT SPECIFIED IN ENTRY CODE WORD	(Nature of error)
RESTART TO INSERT CODE WORD MANUALLY	(Proper corrective action)

As will be seen, the routine requested in the above alarm requires two code words. There was an error in the first code word, however (since tape unit number 15 - octal - does not exist), so the only code word printed was the first code word. The alarm structure pervades all parts of the Library. Detailed information about the alarms can be found in the discussions of the individual routines.

SNEAK-ON AND SNEAK-OFF

As stated earlier, only 8 cells of core storage must be set aside in order to use the Service Library. During operation, these cells contain instructions which initiate a process to save the first 2048 cells of core storage (cells 0-2047) on drum 4 and call in the basic framework of the Library, the "Common Routines." Both of these transfers are accomplished, with sum checking, by instructions in the first 8 cells. This process is known as the "Sneak-on." One of the common routines, the "Select" routine, then interprets the first code word and calls in the requested routine. All of the transfers of information are sum checked except the series of instructions executed within the first 8 cells. These instructions are read from the drums without sum checking.

All Service routines are executed within the first 2048 cells. When the routine has finished, a process similar to the sneak-on is executed in the first 8 cells to restore the contents of the first 2048 cells from the image on drum 4. It then restores the previously listed registers and indicators to their original contents and returns control to the main program. This process is known as the "Sneak-off." As in the sneak-on, everything is sum checked except the sequences of instructions executed in the first 8 cells.

Since the process of sneaking on and off requires accesses to the drums, the time required is large in comparison with computing time. Therefore, provision has been made to enter the Library several times in succession without sneaking on and off each time. This mode of operation is known as the "No-Sneak-off Mode." In this mode, the common routines remain in the first 2048 cells, and the original contents of these cells remain in drum 4. After an entry, all indicators, lights, and registers, which are normally saved, are restored, and control is returned to the main Program without sneaking-off. The next programmed entry immediately sends control to the select routine. If the first 2048 cells can be set aside for the Library, the Service routines can operate without ever having to sneak-off. This can result in considerable savings of time if numerous entries to the Library are made by the main program.

If drum 4 is used for temporary storage between Library entries, it must be used following a sneak-off entry. If it is used after a no-sneak-off entry, the Library will never be able to sneak-off or communicate with drum 4 for any reason, because of the logical sum tests. While the Library is in place, the logical sum of drum 4 is saved, and it is used whenever any routine writes to or reads from this drum.

The sneak-off or no-sneak-off mode of operation is specified by the sign of the first code word. If the sign is positive, the Library sneaks off after the routine is finished. If the sign is negative, the no-sneak-off mode is enforced, and the Library returns control to the main program without sneaking off. If a no-sneak-off entry is made which would return control to a location within the first 2048 cells, an alarm occurs because this section of the program would remain on drum 4, not in the core, after a no-sneak-off entry. If this alarm occurs, restarting will enforce the sneak-off mode and the program may continue. All of the examples of entries given above are sneak-off entries. Changing the signs of all first code words would make them no-sneak-off entries. All single-step load cards make sneak-off entries, but double-step cards may request either sneak-off or no-sneak-off.

After a programmed no-sneak-off entry, control is returned to the main program following the entry (if this location is not within the first 2048 cells). After a manual or load card no-sneak-off entry, the machine halts at cell 776 (octal) with the four sense lights on and the MQ register cleared to receive another first code word. In this way, several manual entries may be made in succession without having to press the load drum key each time.

During the processes of sneaking on and off, the logical sum of the information sent to and read from drum 4 is tested. The logical sum of the common routines is tested as it is read in during the sneak-on, and it is also checked when it is read in the Write Data Tape routine (which is forced to use a modified sneak-on in some cases - see the description of the Write Data Tape routine). If any of these checks fail, the particular

process in which the failure occurred is repeated. If the failure occurs a second time, an alarm condition exists. However, since these processes are all executed within the first 8 cells, there is no facility available to print out information concerning the error. Therefore, the only thing that can be done is to stop the process, and the machine hangs up on a copy instruction. The nature of the error can be determined by the location of the copy instruction.

Table 1

ERROR STOPS DURING SNEAK-ON AND SNEAK-OFF PROCEDURES

If the Machine Hangs up at Cell:	The Cause of the Error Was:
4	Sum check failure while writing the contents of the first 2048 cells to drum 4. Immediately after the writing, the drum is read to insure that it was correctly written. If in 2 tries it is not possible to obtain the same check sum while writing and reading, the machine hangs up at cell 4.
2	Sum check failure while reading in the common routines. The logical sum of the common routines is stored immediately behind them in drum 2. If in 2 attempts the logical sum obtained while reading in the common routines does not agree with the sum on the drum, the machine hangs up at cell 2. At this point, the contents of the first 2048 cells are on drum 4 correctly, but cells 8-639 have been destroyed in the core.
0	Sum check failure while restoring the contents of the first 2048 cells. In the sneak-off process the contents of drum 4 are read back into the core. If the logical sum obtained while restoring these cells does not agree with the check sum computed when the information was originally written on the drum, and the failure occurs twice, the machine hangs up at cell 0.
6	Sum check failure while reading in the common routines and the Write Data Tape routine. This process will be explained in the discussion of the Write Data Tape routine. The hang-up is at cell 6.

If any of these hang-ups occur, there is no general corrective measure that may be taken. At the time this description is being written, the Library has been used for over 2 years and a hang-up has occurred only

2 times. In both these cases investigation showed the drums to be out of order. During the first 8 months the Library was in use, no hang-up occurred. However, during this same time, a bug was present in the sneak-off procedure; this would have produced a hang-up if the logical sum had failed even on the first try. Experience indicates, therefore, that if one of these errors occurs it is probably due to machine malfunction, and the proper corrective action is to call the engineers.

Loading the Library from the self-loading tape begins with a process very similar to the sneak-on. In this process the first 2048 cells are saved on drum 4, and the tape-loading program is set into place. Check sums are tested in exactly the same way as in sneaking-on the common routines. The loading program then writes the Library contents onto the first 3 drums and initiates the usual sneak-off. If the tape was loaded by using the load tape key, the machine halts at cell 3 as after a manual Library entry. If tape loading was initiated by the program-calling sequence, control is sent to the first cell following the calling sequence. The tape sneak-on differs from the usual sneak-on in two ways: 1) the series of instructions executed in the first 8 cells are read in from the tape, rather than from the drums; 2) the alarm conditions which occur before printing facilities are available do permit corrective action.

Table 1A

ERROR STOPS DURING SELF-LOADING TAPE SNEAK-ON PROCEDURE

If during tape loading the machine halts at cell:	5	166 (octal)
The instruction register contains:	HTR -5,4	HTR 167 (octal)
The error was:	Same as sneak-on cell 4 hang up	Sum check failure while reading in the loading program
The corrective measure is:	If initiated by load tape key, press load tape key again. If initiated by calling sequence, restart.	
		Restart

SELF-STARTS

There are three routines (Read Binary Cards, Read Program Tape, Read SAP Cards or Tape) in which information on the cards (or tape) may overrule the normal exiting procedure after execution of the routine. In these cases control is sent, either directly or after a halt, to a starting address specified within the cards (or tape). Control transfers of this type are known as "Self-starts." If control is sent directly to the starting address, the self-start is known as a "Flying-start." If the machine halts and sends control to the starting address when restarted, the self-start is known as a "Halt-and-start." If a self-start occurs, the nature of the entry is ignored except for the sneak-off or no-sneak-off mode. If the entry causes a sneak-off halt-and-start, the machine halts at cell 3. If the halt-and-start occurred during a no-sneak-off entry, the machine halts at cell 606 (octal). In both of these cases, the instruction register contains HTR 2,4, and restarting sends control to the self-start address. Cards which cause a self-start are known as "self-start" cards, and a tape program (to be described later) which asks for a self-start is known as a "self-starting" program. If the cards or tapes are not of the self-start type, the Library exits according to the entry, in the normal way. After an entry which reads self-starting information, the lights, indicators, and registers which the Library saves, are restored, with the exception of index register 4, before control is sent to the starting address.

FIRST CODE WORDS

First code words are interpreted by their prefix, tag, and address - in that order. Thus, any bits in the prefix (other than the sign bit which determines sneak-off or no-sneak-off) will determine the routine. If the prefix is PZE or MZE, a non-zero tag will determine the routine. If both the prefix and tag are zero, then the address determines the routine. If all 3 are zero, the Library proceeds as though the desired routine has already been executed, and returns to the main program in either the sneak-off or the no-sneak-off mode (as indicated by the sign of the prefix). Thus, a first code word of all zeros will cause the Library to simply sneak-off. Clearly, there exist more non-zero addresses than Library routines. Only addresses 1-7 correspond to Library routines, and any other higher addresses cause a "non-existing routine" alarm.

In Table 2 are listed the first code words. Only portions of the words which determine the routine are given. The functions of the rest of the first code words are discussed in the treatments of the individual routines.

Table 2

FIRST CODE WORDS

Prefix	Tag	Address	Routine Requested
±1			Octal Dump
±2			Floating Decimal Dump
±3			Fixed Decimal Dump
±0	1		Read SAP Cards or SAP Tape
±0	2		Punch Binary Cards
±0	3		Read Binary Cards
±0	4		Write Data Tape
±0	5		Read Data Tape
±0	6		Write Program Tape
±0	7		Read Program Tape
±0	0	1	Octal Integer Print-out
±0	0	2	Floating Decimal Print-out or BCD Acquire
±0	0	3	8-column Fixed Decimal Print-out or BCD Acquire
±0	0	4	BCD Print-out
±0	0	5	Time Print-out or BCD Acquire
±0	0	6	Time Integer Acquire
±0	0	7	10-column Fixed Decimal Print-out or BCD Acquire

The first group are known as "dumping" routines, the second as "standard" routines, and the third as "special" routines. Notice that, for the standard routines, even tags request output and odd tags call for input.

FUNCTIONS AND USE OF THE LIBRARY ROUTINES

Common Routines

The common routines are general purpose utility subroutines, which are used by all Service Library routines. In addition, there are two sections which are used as emergency measures. The common subroutines are: Select, Exit, Alarm, Print, and Read-Write Data. The short emergency sections are known as "Rescue" and "Restore Core and Halt."

The print subroutine writes a single line on the on-line printer. It is designed to be used with the standard Share 2 board (the printer board used with the SAP assembly program). Entries to the Print subroutine are made by the Alarm subroutine, and by any Service routine which produces on-line output. All on-line output produced by the Service Library is printed by this subroutine.

The function of the Alarm subroutine is to prepare the successive lines of the alarm print-outs. It may be entered from any of the Service routines, or from the Select, Exit, or Read-Write Data subroutines. The Alarm subroutine first prepares the line to be printed, and then enters the Print subroutine to accomplish the actual printing.

The Read-Write Data subroutine is used for all communication between the Library and the main program. While the Library is in operation, the main program occupies cells 2048-32767 in the magnetic core and cells 8-2047 on drum 4. All Library routines occupy cells 640 ff. while they are working, and operate on data in the remainder of cells 640-2047 which are not occupied by the routines themselves. Whenever data must be transferred between this working space and the main program, all communication with cells of the main program below 2048 must be channeled to or from drum 4. The Read-Write Data subroutine (RWD) transfers contiguous blocks of machine words between the working space and the main program, taking into account that the first 2040 cells of the main program lie on drum 4. RWD is entered by all Library routines and by the Select subroutine. Whenever information is read from or written to drum 4, the logical check sum is tested, and altered if necessary.

There are three alarms which may occur in the RWD subroutine. They are all due to check sum failure while transferring information to or from the image on drum 4. In all cases restarting the machine re-attempts the transfer. The alarms are:

DRUM FAILURE ON READING FROM IMAGE RESTART TO TRY AGAIN

This alarm occurs if the logical sum of the drum contents computed during a reading attempt does not agree with the check sum saved and tested by the Library during sneak-on or a previous RWD entry.

DRUM FAILURE BEFORE WRITING TO IMAGE RESTART TO TRY AGAIN

Before writing to the image it is necessary to compute the logical sum of the portion of the image which is not to be written over. This must be done to compute the check sum of the new image. This alarm occurs if the check sum of the total image fails while the drum is being read to compute the partial check sum of the new image. At this point nothing has been written over the image.

DRUM FAILURE AFTER WRITING TO IMAGE RESTART TO TRY AGAIN

After the image, or portion of the image, is re-written, it is immediately read and tested against the newly computed check sum. This alarm occurs as the result of check sum failure during this final reading pass. At this point the image has been written over. Restarting causes both the writing and reading passes to be re-executed.

The Select subroutine is entered as soon as the common routines are in place in the sneak-on process, or immediately in case of an entry following a no-sneak-off entry. It first saves the condition of all lights, indicators, and registers which are used by the Service routines (except index register 4 and the accumulator overflow light, which would have already been saved during a sneak-on entry, by the sneak-on process itself). It then acquires the first code word and interprets it. If the previous entry was no-sneak-off, and if it requested the same routine now asked for, the routine is still in place and control is sent to it. If the routine is not currently in place, the Select subroutine sets it into place and transfers control to it. All Library routines operate in cells 1200-3777 (octal). The common routines occupy the first 1177 cells.

There are two alarms which may occur in the Select subroutine. They are:

NONEXISTENT ROUTINE REQUESTED RESTART TO INSERT CODE WORD MANUALLY

This alarm occurs if a first code word is given with a zero prefix and tag, and an address not in the range 1-7. Such a first code word does not correspond to any Library routine. Restarting permits the manual insertion of a proper first code word.

DRUM FAILURE ON ACQUIRING SERVICE ROUTINE RESTART TO TRY AGAIN

This alarm occurs if the logical check sum stored behind a service routine does not agree with the sum computed while reading the routine into its operating location. Restarting re-attempts the routine acquisition.

The Exit subroutine is entered as soon as the Service function requested has been completed. It first restores the condition of all saved registers and indicators except index register 4 and the accumulator overflow light, (which will be used and restored by the sneak-off process if the entry was sneak-off). If the entry was sneak-off, it then sets up and initiates

the sneak-off process. If the entry was no-sneak-off, it restores index register 4 and the accumulator overflow light, and returns control to the main program. If the entry caused self-start information to be read, the Exit subroutine modifies the normal exiting procedure and causes the proper self-start to be executed. The contents of the saved lights, indicators, and registers are restored during any of the possible exiting procedures.

There is a single alarm which may occur in the Exit subroutine.

ATTEMPTED RETURN TO PROGRAM IN IMAGE RESTART TO RESTORE IMAGE AND CONTINUE

This alarm occurs following a no-sneak-off entry if control is to be sent to a location in the range 8-2047. After a no-sneak-off entry, these locations remain in the image on drum 4. Restarting enforces the sneak-off mode, and the program may continue. This will in no way impair future entries.

The Rescue section is used when a program has run wild and stopped while the common routines are in the core. It is entered by manually transferring to cell 400 (octal). The contents of the accumulator, MQ register, and index register 4 are saved in cells 1175, 1176, and 1177 (octal), respectively. Then the current contents of cells 0-2047 are written onto drum 3,* and the contents of drum 4 are restored to the first section of the core. Complete dumps of the core and drum 3 will then provide the entire contents of the main program and the portion of the Library in the core at the time of the program malfunction. If the rescue section has been destroyed itself, the best procedure is simply to dump the entire core and retire to deep, concentrated study.

The Restore Core and Halt section is primarily used by three Library routines (BCD Print out, Read SAP Cards or Tape, Write Program Tape), but may be used manually by transferring to cell 700 (octal). It is used by the routines following an alarm for which there is no corrective action which would enable the routine to continue. (These conditions will be described in the sections about the routines.) The net effect of using this section is that the core is restored to its condition prior to the last entry (except Acc, MQ, cells 0-7, Trap), and the machine halts at cell 3 as after a manual entry. If the entry was a programmed entry, a manual HTR 0,4 will repeat the entry.

*This will destroy part of the Service Library. The dump programs will remain and can be used, but the Library will have to be re-acquired if it is to be used further.

It is occasionally convenient to know the address of the Exit subroutine. Transferring to it manually will cause the machine to return to the main program or halt, depending on the entry. The Exit subroutine begins at cell 526 (octal).

It is also convenient to be able to stop a Library routine manually in the middle of its function and insert a new first code word. This is done by inserting the new first code word into the MQ register and manually transferring to cell 1000 (octal).

The manual transfer to cell 1000 provides a corrective action in case of a feed jam during a card-reading operation. If a jam occurs, the corrective actions consist of: 1) repair and replace the entire card deck and ready the reader; 2) insert the Read Binary or Read SAP cards code word into MQ; and 3) perform the manual transfer to cell 1000. The routine will continue and exit properly regardless of where or when the feed jam occurred.

SERVICE LIBRARY OUTPUT

For routines which write BCD output (except the dumps), the nature of the output is specified in the decrement of the first code word. The output comes off-line and/or on-line. Off-line output is written onto tape unit 0 (=10) and may be printed on the off-line printer under "program" control. On-line output is printed immediately on the on-line printer, and is an exact copy of any off-line output prepared at the same time. There are two types of off-line-only output. The first of these permits the output to be put on-line also by putting down sense-switch 6. The dumping routines always have this type of output. The second type of off-line-only output may not be monitored by sense-switch 6. Sense-switch 6 is the only sense switch used by Service Library routines, and it is only used for this purpose.

Output appears according to the following table:

Table 3

OUTPUT CONVENTIONS FOR PRINT-OUT, DUMP, AND TIME ROUTINES

If the First Code Word Decrement is:	Output Appears:
-1	Off-line only; it cannot be monitored with SSW 6.
0	Off-line only; it may be monitored with SSW 6. All dumping routines give this type of output.
1	Both on-line and off-line.
2	On-line only; never off-line.
3 through -2	No output for Read SAP Cards or Tape routine. Identified with 2 for BCD and Octal print outs.
3 through 7	Identified with 2 for Floating Decimal, Fixed Decimal, and Time print outs.
8 through -2	Indicates special functions for Floating Decimal, 8-column Fixed Decimal, 10-column Fixed Decimal, and Time routines. In these cases no output is written. They will be explained in the sections about the individual routines.

EFFECTIVE ADDRESSES

In the dumping and print-out routines, for which it is necessary to specify the upper and lower addresses of a block of words to be printed out, certain cases occur frequently enough to make it desirable to provide convenient methods of furnishing their code words. In these cases, unspecified addresses (zero) are replaced by the addresses of the usual case. For example, in the earlier example of a manually initiated octal core dump, the unspecified addresses were interpreted as the lowest and highest core addresses. If a tag of 1-4 (requesting dumps of drums 1-4 respectively) had been specified, the addresses would have been interpreted as the highest and lowest drum addresses. Any non-zero address is taken as an effective address. A zero address is assigned an effective address according to the routine. The effective addresses assigned by the particular routines are listed in the descriptions of the routines.

END-OF-FILE MARKS

All information or output written onto magnetic tape by the Service Library lies within the first file of the tape. Whenever a record or series of records are to be followed by an end-of-file, the mark is placed behind the last record written. Then the tape is positioned in front of the end-of-file gap so that further entries simply lengthen the file. An end-of-file is always written in this way by the Write Data Tape, Write Program Tape, and Dumping routines.* It is written optionally by the output routines (Octal Print Out, Floating Decimal Print Out, Eight or Ten Column Fixed Decimal Print Outs, and BCD Print Out), so that the time required to move over the end-of-file gap may be saved when a computation requires many entries to these routines. It is never written by the Time Print Out routine, as the printing option of this routine is designed primarily to initiate the output from a computation.

*Strictly speaking, the Write Program Tape does not leave the tape positioned in front of the EOF, but preceding the format record of the program just written. This record is the last record of the program and is followed by the EOF.

MULTIPLE CODE WORDS

In all Library routines in which a variable number of secondary code words is permitted, all except the final one must have a negative sign. The final secondary code word is identified by a positive sign. If the number of secondary code words is fixed, however, the signs have meanings different from this. The routines which permit multiple secondary code words are Write Program Tape, Print BCD, and Read Data Tape.

THE "STANDARD" SERVICE LIBRARY ROUTINES

Tape-handling Routines

The Service Library contains four routines to read and write magnetic tapes of two types, "Data" tapes and "Program" tapes. Each type of tape has its own unit of information, written in a tape format designed for that unit of information. Any entry to a Library tape-handling routine will cause a complete unit of information to be transmitted between magnetic tape and the main program.*

The unit of information on a data tape is the record. The record consists of a first word which is the logical sum of the rest of the record, and a block of machine words which were written from consecutive cells in the core. These records are known as "Data" records, and may contain any contiguous block of machine words which can sit in the core (except in cells 0-7). Data records, or any contiguous portion thereof, may be read into any block of consecutive cells (in the core) of the proper size. (Check sums are only for testing and are never read into the main program by Data or Program tape-handlers.)

The physical layout of a data tape is simply one or more data records followed by an end-of-file mark. The EOF is always written after each data record. The file may contain as many data records as the length of the tape permits.

The unit of information on a program tape is the "Program." This consists of one or more records which contain the machine words stored on the tape, and a final two-word control record called a "Format" record. The former, called "Program records," contain at most 1024 (2000 octal) machine words. Larger contiguous blocks of machine words, stored in the program, are divided into one or more program records containing 1024 words, and a

*The Read Data Tape is an exception to this rule. It is possible to read only a portion of the data tape unit of information.

final program record containing up to 1024 words. Preceding the machine words on a program record are two control words. The first of these is the 1's complement of the logical sum of the rest of the record. The 1's complement is used in order to prevent confusion between program tapes and data tapes. Neither can be read as the other. The second word on the program record contains the core address of the first machine word stored in the record. Unlike data tapes, program tapes contain only nonrelocatable words, and a program must be read into the core in its entirety. Up to 6 contiguous blocks of machine words, each up to 32760 words in length, may be stored in the program records of a single program. The format record indicates the end of the program on tape and contains the number of program records in the program plus 2 (for positioning purposes), and determines whether the program is a self-starting program. In the self-start case, this record contains the starting address.

The layout of a program tape is almost identical with that of a data tape in that it consists of a single file of units of information (in this case, programs). An EOF is always written after each program. The only difference in the layout of a program tape is that the file is begun by an initial two-word control record. This record signifies the beginning of a program tape, and must be written whenever a program is first written onto a blank tape. The length of the tape is the only limit to the number of programs contained in the file.

Data tapes are designed as storage for tables, intermediate results, constants, etc., which are read into the machine and used during a computation. Program tapes are primarily designed for storing checked-out main programs. A program stored in this form is very similar to one stored as a deck of nonrelocatable SAP binary cards. All information is check summed, and the tape itself contains the core destination addresses of the words. Program tapes may also be self-starting, as is usual with a binary deck. Program tape has two great advantages over a card deck, however. First of all, the time required to read a program into the core is only a minute fraction of the time used reading in a binary deck of the same program. Secondly, any of the programs on a program tape are immediately and repeatedly available.

Although the formats and purposes of the two types of tape are different, they have several operational similarities. Whenever any tape record is read, the logical check sum (the first word of any record) is tested against the sum accumulated while reading the record. No information is accepted unless the two sums agree. If this test fails, the reading is attempted once more, and if the failure recurs an alarm occurs. When any record is written, it is immediately read back to insure that it was written correctly. All records must pass this test, and if a record fails twice an alarm occurs.

The first code words of all tape-handling routines differ only in the tag. The tape unit number is always specified in the address of this code word, and tape-positioning is always requested in the decrement. Decrements in the range 0 through 16383 cause the tape to be advanced this number of units of information prior to the reading or writing process. Decrements in the range -1 through -16383 cause the tape to be initially backspaced this number of units of information. (The minus means 2's complement.) This method of initial tape-positioning permits any record or program to be read or written by any entry.

The four Service Library tape routines are: Read Program Tape (RPT), Write Program Tape (WPT), Write Data Tape (WDT), and Read Data Tape (RDT). These routines will now be discussed in this order.

Read Program Tape

The Read Program Tape routine is identified by a zero prefix and a tag of 7 in the first code word. The tape contains the core addresses of the information in the program; therefore, only a first code word is necessary to fully specify the operation. The functions of the address and decrement portions of this code word were discussed above. Single-step load cards are available to read any of the first 6 programs from tape unit number 2. After RPT is in place, it positions the specified tape according to the decrement of the first code word, and reads an entire program into place. When the routine finds the format record, which follows the last program record, it examines the record to determine the correct exiting procedure. If the format record indicates a self-start, the Library exits and sends control, either directly or after a halt, to the address contained in the record. If it is not a self-start record, the Library exits according to the entry.

Several alarms may occur in RPT. Following is a list of these alarms and their corrective actions:

ILLEGAL TAPE UNIT
RESTART TO INSERT CODE WORD MANUALLY

This alarm occurs when the tape unit specified in the address of the first code word is not in the range 1-9. If the machine is restarted, the entire correct first code word may be inserted as in a manual entry.

TAPE NOT POSITIONED AT FORMAT RECORD
IF WRONG TAPE DESIGNATION CORRECT AND RESTART

A program tape must be positioned at a format record in order to be read or written by the program tape routines (unless writing the first program on a blank tape, of course). When the tape is mounted on the unit,

it is automatically positioned at the initial format record, and any program tape entry will leave the tape positioned at a format record. Therefore, in proper operation, a program tape should always be positioned at a format record. If this alarm occurred because the dials on the tape units were incorrectly set, the dials may be corrected and the machine restarted.

CHECK SUM FAILURE
RESTART TO TRY AGAIN

This occurs in the event of a reading sum check failure. Restarting will attempt to read the record again.

THIS IS NOT A PROGRAM TAPE
IF WRONG TAPE DESIGNATION CORRECT AND RESTART

This alarm may occur for many reasons. They will not be treated in detail here. In general, it occurs whenever RPT finds tape behavior which could not possibly occur with a legitimate program tape. If the tape dials were set incorrectly, they may be corrected and the machine restarted.

POSITIONING IMPOSSIBLE
IF WRONG TAPE DESIGNATION CORRECT AND RESTART

This alarm occurs when the positioning requested in the decrement of the first code word would move the tape out of the file of programs. It may be caused by either too many advances or too many backspaces. The corrective action is the same as in the previous alarm.

In all cases in which an alarm condition was detected by moving the wrong tape, the incorrectly moved tape is reset to its original position before the alarm. Correcting the dials and restarting will not leave any tape incorrectly disturbed. Diagrammatically, the RPT code word looks like this:

Prefix	Decrement	Tag	Address
± 0	Positioning	7	Tape Unit Number

Examples of entries to this routine are:

TSX 4,4 Read a program from unit number 2 after backspacing three
PZE 2,7,-3 programs, sneak-off.

TSX 4,4 Read a program from unit number 5, after advancing two
MZE 5,7,2 programs, don't sneak-off.

Write Program Tape

The Write Program Tape routine is identified by a zero prefix and a tag of 6 in the first code word. The uses of the address and decrement of this code word were discussed above. The WPT routine first positions the tape according to the decrement of the first code word and then writes a program onto the tape. It then writes the EOF, and positions the tape ahead of the last format record. The WPT routine requires one or more secondary code words to specify its operation. Up to 6 secondary code words may be given in a single entry. All except the last must have a negative sign.

Secondary code words for WPT are of 3 types. The first of these, a "Zero" secondary code word, is either plus or minus zero. A zero secondary code word must be given immediately after the first code word. A negative zero causes WPT to rewind the designated tape and write the initial two-word format record. This must be done whenever a program is written onto a blank tape. A positive zero causes WPT to position the tape according to the decrement of the first code word and then exit according to the entry. The second type of secondary code word is the "Block Address" code word. This type of secondary code word may have either sign and may be given in any position. The address portion contains the first address of a block of machine words to be written onto the tape, and the decrement portion contains the last address of this block. The WPT routine divides the block of machine words into as many 1024 word sections as are necessary and writes them, together with the core address of the first word of each section, onto the tape. If the sign of a block address code word is negative, WPT looks for another secondary code word. If the sign is positive, a format record is written to indicate the end of the program and exits. (This type of format record does not cause a self-start, but causes RPT to exit normally.) The last type of secondary code word is the "End record" code word. These code words cause self-start format records to be written at the end of the program. An end record code word must be positive in sign and must not: 1) be given as the first secondary code word; 2) be given immediately following an initial zero secondary code word. The address portion of this code word contains the self-start address, and the decrement must contain a zero or a one. A zero causes a halt-and-start record, and a one indicates a flying-start record. An address of seven or less is an alarm condition.

The alarms which may occur during the execution of WPT are:

ILLEGAL TAPE UNIT

RESTART TO INSERT CODE WORD MANUALLY

POSITIONING IMPOSSIBLE

IF WRONG TAPE DESIGNATION CORRECT AND RESTART

THIS IS NOT A PROGRAM TAPE
IF WRONG TAPE DESIGNATION CORRECT AND RESTART

TAPE NOT POSITIONED AT FORMAT RECORD
IF WRONG TAPE DESIGNATION CORRECT AND RESTART

These alarms are identical with the ones discussed in the previous section. The causes are the same, and the corrective actions are also the same.

CHECK SUM FAILURE
RESTART TO TRY AGAIN

This occurs if any record cannot be properly read immediately after it is written. Restarting attempts to write the record again.

TOO MANY CODE WORDS
RESTART TO RESTORE CORE AND HALT

This alarm occurs if 6 negative secondary code words are found. No corrective action which would permit the routine to continue is possible since the machine has no way of knowing which, if any, of the code words is the last one.

ILLEGAL CODE WORD
RESTART TO INSERT CODE WORD MANUALLY

This alarm may be caused by:

- 1) A zero secondary code word which is not the first secondary code word.
- 2) A block address secondary code word with an address less than or equal to 7, or an address greater than the decrement.
- 3) An end record secondary code word given as the initial secondary code word, immediately following an initial zero secondary code word, or with a negative sign.

Restarting permits the correct code word to be inserted manually.

As in RPT, correcting the dials and restarting after an alarm will not leave any tape incorrectly moved. Diagrammatically, the code words are:

<u>Prefix</u>	<u>Decrement</u>	<u>Tag</u>	<u>Address</u>	<u>Code Word</u>
+0	Positioning	6	Tape Unit Number	First
+0	0	0	0	Zero
+0	Last Address		First Address	Block Address
+0	1 or 0 1 = Flying Start 0 = Halt-and-Start		Self-Start Address	End Record

The blank portions are not examined. Examples of programmed entries are:

TSX 4,4 Write a program on unit number 2 with no initial
PZE 2,6 positioning, sneak-off.
MZE Initiate the tape
MZE 100,0,400 Include cells 100 through 400
MZE 5600,0,7200 Include cells 5600 through 7200
PZE 5600,0,1 Write a flying-start record to start at cell 5600

TSX 4,4 Backspace the program tape on unit number
MZE 4,6,-8 4 eight programs, don't sneak-off.
PZE

TSX 4,4 Write a program on unit number 3, after advancing
PZE 3,6,2 two programs, sneak-off.
MZE 4096,0,5069 Include cells 4096 through 5069
PZE 576,0,5789 Include cells 576 through 5789, write an end of program
format record.

Write Data Tape

The Write Data Tape routine is identified by a zero prefix and a tag of 4 in the first code word. This routine writes a single binary record onto the tape specified in the first code word, after positioning according to the decrement. An EOF is written after each record in the manner described earlier. One secondary code word is necessary to fully specify the function of the routine. The address of the secondary code word contains the first address of the block of machine words to be written in the record, and the decrement contains the last address of this block.

The WDT routine is capable of writing a single record of any contiguous block of words in the core, no matter how large. In order to write such a record, all words of the block must sit in the core at the same time. If the block is so large that WDT cannot operate in the usual execution location (cells 640-2047), a sneak-off procedure is carried out, and the data record is written by instructions in the first 7 cells. After the record is written, the common routines and WDT sneak back on, and

the tape is tested for the correct check sum. This mode of operation is known as the sneak-off mode for WDT. The check sum of the combined common routines and WDT is tested when the sneak-back-on occurs, and a double failure causes the hang-up in cell 6 which was listed in Table 1.

The alarms which may occur while WDT is operating are:

INCORRECT TAPE UNIT SPECIFIED IN ENTRY CODE WORD
RESTART TO INSERT CODE WORD MANUALLY

This alarm occurs if the tape unit number specified in the first code word is not in the range 1-9. If the machine is restarted, the correct code word may be inserted.

POSITIONING REQUESTED IMPOSSIBLE
IF WRONG TAPE DESIGNATION CORRECT AND RESTART

This alarm occurs if advancing is requested which would move the tape past the EOF. There is no foolproof way to detect too many back-spaces. If the tape dials were incorrectly set, they may be corrected and the machine restarted. Any tape wrongly moved will be repositioned before this alarm.

ILLEGAL SECOND CODE WORD
RESTART TO INSERT CODE WORD MANUALLY

This alarm occurs when the address of the second code word (the first secondary code word) is less than or equal to 7, or if the address is greater than the decrement. Restarting permits the correct second code word to be inserted manually.

TAPE CHECK SUM FAILURE
RESTART TO TRY AGAIN

This alarm occurs if the record cannot be read properly immediately after it is written. Restarting will attempt to write and read the record again. Before this alarm occurs, 2 writing attempts will have been tried, and each attempt will have included 2 reading attempts.

SNEAK OFF MODE DRUM FAILURE
RESTART TO TRY AGAIN

Information for sneaking-back-on must be stored on the drums when WDT operates in the sneak-off mode. All transfers are check summed, and a double failure causes this alarm. Restarting attempts the transfer once more.

Diagrammatically, the WDT code words are:

Prefix	Decrement	Tag	Address	Code Word
<u>±0</u>	Positioning Last Address	4	Tape Unit Address First Address	First Second

The blank portions are not examined.

Examples of programmed entries are:

TSX 4,4	Write a record on unit number 6, after advancing seven
PZE 6,4,7	records, sneak off. The data begin at cell 246 and end
PZE 246,0,357	with cell 357.
TSX 4,4	Write a record on unit number 8, after backspacing two
MZE 8,4, -2	records, don't sneak off. The data begin at cell 100 and
PZE 100,0,200	end with cell 200.
TSX 4,4	Write a record on unit number 5, with no initial positioning,
PZE 5,4	sneak off. The data begin at cell 200 and end with cell 300.
PZE 200,0,300	

Read Data Tape

The Read Data Tape routine is identified by a zero prefix and a tag of 5 in the first code word. RDT first positions the tape to any data record in the first file, according to the decrement of the first code word, and then reads any contiguous portion of the data record into any block of consecutive cells (except 0-7). The RDT routine always requires one secondary code word, and may require 2. If both secondary code words are given, the first must have a negative sign. The first and last addresses into which the data are to be read are specified in the address and decrement of the first secondary code word, respectively. The second secondary code word permits reading to begin at any word on the data record. The address portion of this code word contains the tape address of the first word on the record to be read. On the record, the address of the check sum is zero; the other addresses follow consecutively. A request for a first tape address of zero is identified with address one. These are exactly the tape addresses which appear in a Service Library dump of the record.

There are 2 special features of RDT which may be requested in the first secondary code word. If the tag is 4, 5, 6, or 7, the tape will be positioned according to the decrement of the first code word, and the routine will exit. If this "Position only" option is used, RDT will not look for a second secondary code word. Only the tag is examined in this case.

The second feature permits records to be read without having to specify the last core destination address. If a decrement of zero is given in the first secondary code word, the data record will be read until either the entire record has been read, or until a word has been read into the highest cell of the core. If an entire record is to be read, it is only necessary to supply the first address. This mode of operation is known as the "No last address" mode.

The alarms which may occur in RDT are:

POSITIONING REQUESTED IMPOSSIBLE
IF WRONG TAPE DESIGNATION CORRECT AND RESTART

NONEXISTENT TAPE UNIT
RESTART TO INSERT CODE WORD MANUALLY

These two alarms are the same as in WDT.

ILLEGAL SECOND CODE WORD
RESTART TO INSERT CODE WORD MANUALLY

This alarm occurs if the address portion of the second code word (the first secondary code word) is less than or equal to 7, or if it is greater than a non-zero address in the decrement. (A zero decrement indicates the no-last-address mode.) Restarting permits inserting the correct code word.

TAPE CHECK SUM FAILURE
RESTART TO TRY AGAIN

This alarm occurs if the record check sum does not agree with the check sum computed while reading the data record. If this alarm occurs repeatedly for a record, it is probable that the record is not a data record, or that the tape is malfunctioning. Restarting attempts the reading process again.

RECORD TOO SHORT
IF WRONG TAPE DESIGNATION CORRECT AND RESTART

This alarm occurs whenever the core addresses specified in the second code word ask for a block of data larger than the block on the record. This alarm can never occur while reading in the no-last-address mode. If the tape dials were set incorrectly, they may be corrected and the machine restarted.

NONEXISTENT ADDRESS ON TAPE
IF WRONG TAPE DESIGNATION CORRECT AND RESTART

This alarm occurs if a first tape address is given in the third code word which is greater than the address of any word in the data record.

If the tape dials were incorrectly set, they may be corrected and the machine restarted.

NO RECORD HERE
IF WRONG TAPE DESIGNATION CORRECT AND RESTART

This alarm occurs if the tape is positioned behind the last data record in the file before being read. If the tape dials were incorrectly set, they may be corrected and the machine restarted.

IMPORTANT

As in the program tape-handling, several of the above alarms may be triggered as a result of incorrectly set tape dials. However, principally due to the fact that a BST instruction is interpreted as a NOP when the tape is rewound, there is no foolproof scheme for keeping track of data tape-positioning - without spending very disproportionate amounts of time moving the tape. Therefore, of the above alarms, only the POSITIONING REQUESTED IMPOSSIBLE, in the case of TOO MANY ADVANCES, restores the wrongly moved tape. Program tape routines will never leave any tape wrongly disturbed, but every RDT or WDT alarm which has a corrective action of resetting the tape dials and restarting, except the alarm mentioned here, might leave a tape wrongly moved. Care should be exercised in the assignment of units containing data tapes.

Diagrammatically, the code words for RDT are:

Prefix	Decrement	Tag	Address	Code Word
+0	Positioning	5	Tape Unit Number	First
+0	Last Address or	0	First Address	Second
	0	or		
		4		

+ = No third code word 0 = Usual reading case
- = Third code word 4 = Position only
0 = No last address mode

Prefix	Decrement	Tag	Address	Code Word
			First Tape Address	Third

Blank sections are not examined.

Examples of programmed RDT entries are:

TSX 4,4	Read data record from unit number 6, after advancing
MZE 6,5,2	two records, don't sneak off. Starting with the 25th word
MZE 100,0,400	on the record, read in data beginning at cell 100 and
PZE 25	going through cell 400.
TSX 4,4	Backspace the data tape on unit number 7 by nine records.
PZE 7,5, -9	
PZE 0,4	
TSX 4,4	Read a data record from unit number 8, with no initial
PZE 8,5	positioning. Begin reading at the first data word on the
PZE 500	record. Read until the record is finished or until cells
	500-32767 are filled.

Binary Card Handling Routines

The Service Library contains two routines which read and punch binary cards, respectively. The routines are Read Binary Cards (RBC) and Write Binary Cards (WBC). These routines work with binary cards of two types: "Data" cards and "End" cards.

Data cards are binary cards written in the standard Share non-relocatable format. Each card contains up to 24 words, of which the first two, 9left and 9rite, are used for control purposes. The word 9left contains the number of data words on the card (i.e., all words other than 9left and 9rite) in its decrement portion, and the core address of the first data word in its address section. The word 9rite is the logical sum of 9left and all data words. The data words appear in the standard arrangement - 8left, 8rite, 7left, 7rite, 6left, etc. - and are read into consecutive cells in this order, beginning at the address specified in 9left. The logical sum is tested whenever RBC reads a data card.

End cards are identified by a zero data word count in the decrement of 9left, and are of three types: Blank, Flying start, and Halt-and-start. A blank end card is one in which 9left and 9rite are both zero. A Flying start end card is characterized by a nonzero address portion, and prefix of one, in 9left. A Halt-and-start end card is identical with the Flying start card, except that the 9left prefix is zero. For both of these cards, the address portion of 9left contains the self-start address. When RBC finds an end card it terminates the reading operation and exits according to the card. A blank end card causes the Library to exit according to the entry. Flying or halt-and-start end cards cause flying or halt-and self-starts, respectively, to the address specified in the address portion of 9left.

Since cells 0-7 may never be used by the main program, self-start cards whose self-starting addresses are in the range 1-7 are identified with

blank end cards. Data cards whose data words would be read into these cells cause an alarm condition.

Read Binary Cards

The Read Binary Cards routine is identified by a zero prefix and a tag of 3 in the first code word. The RBC routine reads data cards until an end card is found, and then exits according to the end card. The combination of data cards followed by an end card is known as a binary deck, and is subject to the restrictions that it may not begin with an end card, nor terminate without an end card.

The first restriction is imposed because it would be nonsensical to make an entry to RBC when there were no data cards to be read. The routine is not an elaborate scheme to make a transfer. The second restriction was motivated by the fact that through oversight, or perhaps because of card-feed failure, it is possible that all cards which should be read are not fed into the reader. The reading operation ends only when RBC finds a termination signal which was supplied by a willful act on the part of the operator.

The RBC routine requires only the first code word, which may be quickly supplied by the use of a single-step load card. The address portion of the code word is not examined, but the decrement is used to provide greater flexibility in the reading process. It is sometimes helpful to be able to read information from binary cards to addresses other than those specified on the cards. This is accomplished by the use of the decrement of the first code word. In the RBC routine data words are read into the addresses obtained by adding this decrement to the addresses specified on the data cards. The single-step load card provides a first code word with a decrement of zero. This causes the data words to be read into exactly those cells specified by the cards. If any address bias would cause data words to be read into the first 8 cells, an alarm occurs. The bias is never added to the self-start address of an end card.

The alarms which may occur in RBC are:

DESTINATION 0-7

This occurs when the reading process would read words into the first 8 cells. There is no general corrective measure. The only remedial steps which might be helpful would be manual transfers to Exit or Restore Core and Halt. It is also possible at this time to insert a new first code word into MQ and transfer to cell 1000 (octal).

NO CARDS RESTART AFTER CORRECTING DECK

This alarm occurs if no deck sits in the reader or if the end card of the deck was missing. The proper cards may be placed in the card reader hopper, and the machine restarted.

FIRST CARD IS END CARD RESTART AFTER CORRECTING DECK

This occurs if no data cards precede an end card. The machine may be restarted after the deck is corrected.

CHECK SUM FAILURE RESTART TO IGNORE

This alarm occurs if 9rite does not agree with the logical sum computed while reading a data card. Restarting causes the routine to ignore the failure and continue.

Diagrammatically, the RBC code word is:

<u>Prefix</u>	<u>Decrement</u>	<u>Tag</u>	<u>Address</u>	<u>Code Word</u>
±0	Address Bias	3	Not used	First

Examples of programmed entries are:

TSX 4,4	Read a binary deck into the addresses specified on the data
MZE 0,3	cards, don't sneak off.
TSX, 4,4	Read a binary deck into addresses 1000 lower than those
PZE 0,3, -1000	given on the data cards, sneak off.

The single step call card supplies the code word PZE 0,3.

Write Binary Cards

The Write Binary Cards routine is identified by a zero prefix and a tag of 2 in the first code word. Only a first code word is necessary. The WBC routine performs one of two functions, depending on the code word. It either punches a self-start end card, or punches data cards of a contiguous block of machine words.

For a self-start card, the self-start address is given in the address of the first code word, and the decrement must contain a 1 or a 0, for flying or halt-and starts, respectively. The self-start address must not be 7 or a lower number. For data cards of a block of words, the first

address of the block is given in the address portion of the code word, and the last address of the block is given in the decrement.

No part of the block may lie below cell 8. The WBC routine divides the block into as many 22 word sections as necessary and punches data cards in the format described above.

After punching data cards, WBC does not advance the cards in the punch. In this way, a deck of data cards of several nonadjacent blocks may be conveniently collected. After punching an end card, however, 2 more cards are moved through the punch. Therefore, it is not necessary to feed any cards through the punch before removing the end card and all data cards which precede it.

The only alarm in WBC is:

ILLEGAL CODE WORD

RESTART TO INSERT CODE WORD MANUALLY

This occurs if the code word contains an address portion of 7 or less, or if the decrement portion is both less than the address portion and greater than 1. If the machine is restarted the correct code word may be inserted manually.

Diagrammatically, the two types of code words are:

Prefix	Decrement	Tag	Address	Code Word
± 0	Last Address	2	First Address	Block Addresses
± 0	1 or 0	2	Self-start Address	Self-start
	1 = Flying Start			
	0 = Halt-and-start			

Examples of programmed entries are:

TSX 4,4 Punch data cards of the block beginning at cell 100 and
MZE 100,2,300 ending with cell 300, don't sneak-off.

TSX 4,4 Punch a flying-start card to start at cell 100, sneak-off.
PZE 100,2,1

SAP Information Handler

The Service Library contains a routine, Read SAP Cards or Tape (RSAP), which can read information directly from SAP cards, or from tape prepared from SAP cards on the off-line equipment. The SAP cards

are punched in the format used by the SAP 3-7 assembly program, and are subject only to the limitation that no symbolic or arithmetic expressions may be given in the location, address, tag, or decrement fields. (Initial minus signs are permitted in the address and decrement fields.) If any of these fields are specified, they must contain decimal integers.

The operation field (columns 8, 9, and 10) may contain any operation or extended operation code accepted by SAP 3-7 which may be translated directly into a computer word or words. In addition, the control codes ORG, END, and FLP are accepted. Extended codes such as SWT, RTB, RDR, WPR, and SLN are all interpreted properly. Cards with operation fields of OCT or DEC may contain several numbers, in the usual way. Decimal input is accepted in either fixed-point or floating-point form, and, unlike SAP 3-7, this routine interprets numbers to the full 35-bit accuracy of the machine.

Core destination addresses are handled exactly as SAP 3-7 treats them. An origin card causes an address counter to be set and this counter is advanced by one for each word converted. Whenever a card is found which contains a non-blank location field, the counter is reset to that address. In addition, it is possible to have the counter decreased by one for each word, rather than increased. Cards read this way are said to be read in the "Backward" mode. When an entry to RSAP is made, the routine always begins to operate in the forward mode. If a card is found with FLP (Flip) in the operation field, the mode is changed. A new origin card will not change the mode; only a FLP card will do this.

As the SAP information is being read, it is listed on the off-line and/or on-line equipment as indicated in Table 3.

The RSAP routine is identified by a zero prefix and a tag of 1 in the first code word. Only a first code word is required, and may be supplied by single-step load cards to read information from cards or from tape unit number 3. The decrement of the code word is used to indicate which type of output is requested, and the address portion indicates the input unit. An address of zero causes the information to be read from the card reader. An address of 1-9 requests the information to be read from tape unit number 1-9, respectively.

SAP information is read until an end condition occurs. This may be an end card or end-of-file. A blank card or end-of-file causes RSAP to exit according to the entry. If they are sensed before any information is read, an alarm occurs. A card with operation code END is interpreted as a flying-start card, and a flying start to the address given in the address field occurs. No other card should follow an END card since several cards might be read into a buffer before the END card is interpreted.

The RSAP alarms are:

ILLEGAL CODE WORD
RESTART TO INSERT CODE WORD MANUALLY

This alarm occurs if the address of the code word is not in the range 0-9. No other addresses correspond to permissible input units. If the machine is restarted, the correct code word may be inserted as in a manual entry.

INITIAL EOF
RESTART AFTER CORRECTING

This alarm occurs if the information begins with a blank card or end-of-file. The machine may be restarted after the card deck or tape containing the information is corrected. A flying-start card is accepted as a first card.

ILLEGAL PUNCH
RESTART AFTER CORRECTING

This alarm occurs only when the input is from cards. Only certain combinations of punches on the cards correspond to BCD characters. Punches which do not correspond to legal characters trigger this alarm. A binary card will almost always cause it. The machine may be restarted after the card deck is corrected.

PARITY FAILURE
RESTART TO TRY AGAIN

This alarm may only occur when input is from tape. It is caused by a parity check failure during the reading process. Restarting attempts to read the record again. If this alarm recurs repeatedly, it is probable that the tape being read is not a proper input tape.

RECORD OF INCORRECT LENGTH
RESTART AFTER CORRECTING

This alarm occurs only when an improper input tape is being read. The machine may be restarted after the correct tape is ready to be read.

ILLEGAL CHARACTER
RESTART TO RESTORE AND HALT

This alarm occurs whenever a symbolic or improper arithmetic expression appears in the location, address, tag, or decrement field. It also occurs if the operation field does not contain a legitimate operation

code, or if punches appear in columns 7 or 11. When the alarm occurs, the card or record containing the illegal character is printed along with the alarm. This alarm has no general corrective action. If the machine is restarted, the Library will exit and halt at cell 3.

NUMBER OUT OF RANGE RESTART TO RESTORE AND HALT

This alarm occurs when a floating decimal number, which is too large or too small to be converted to a normalized floating-point machine word, is read. It also occurs when a fixed-point decimal number is given which cannot be converted into a machine word. There is no general corrective measure. Restarting will cause the routine to exit, and the machine will halt at cell 3. The card which contained the number will be printed out in the alarm.

ILLEGAL DESTINATION RESTART TO RESTORE AND HALT

This occurs when the destination address of a word lies in the range 0-7. This may occur as the result of the address counter entering the forbidden range from either side, or from an initial blank location field. If the machine is restarted, the Library will exit and halt at cell 3.

Diagrammatically, the RSAP code word is:

Prefix	Decrement	Tag	Address	Code Word
±0	Type of Output	1	Input Unit	First
	-1 = off-line only, no monitoring with SSW6		0 = Cards 1-9 = Tape Unit 1-9, respectively	
	0 = off-line only, may be monitored with SSW6			
	1 = on-line and off-line			
	2 = on-line only			
	3 through -2 = no output			

Examples of programmed entries are:

TSX 4,4	Read SAP information from cards, don't sneak off, list
MZE 0,1,1	the information on and off line.
TSX 4,4	Read SAP information from tape unit number 3, sneak
PZE 3,1,4	off. Do not list the information.

The single-step load cards supply first code words PZE 0, 1 and PZE 3, 1 to read from cards or tape unit 3, respectively.

Dumping Routines

The Service Library contains routines which provide octal, floating-decimal, and fixed-decimal dumps of core, drums, or magnetic tapes. They are all identified by a non-zero prefix in the first code word. A prefix of ± 1 designates octal, ± 2 floating decimal, and ± 3 fixed decimal.

The three dumps all provide for core, drum, or magnetic tape diagnostics, and differ only in format and type of information given.

The octal dump displays octal addresses, machine words in the SAP format, mnemonic interpretations, the date, time, and an exact specification of the type and extent of the dump. In addition, an octal core dump displays the condition of all lights, registers, and indicators saved by the Service Library, at the time the dump is taken. It also displays the condition of the floating-point underflow button and the sense switches.

If several octal core dumps are made in succession in the no-sneak-off mode, the panel condition is displayed only in the first. The octal dump does not print zero words unless they occur adjacent to non-zero words, or occur as the final word of a record on magnetic tape.

The floating-decimal dump also displays octal addresses, the date and time, and an exact specification of the type and extent of the dump. Normalized floating decimal numbers are printed in the following format: $\pm 1.1234568 \pm 12$ ($= \pm 1.1234568 \times 10^{\pm 12}$). Zeros and unnormalized numbers are not printed unless they occur adjacent to non-zero normalized numbers, or occur as the final word of a tape record. Zeros appear as ± 0 and unnormalized numbers are blanked, except when as the final word of a tape record. In this case the words "LAST WORD" appear.

The fixed-decimal dump also prints octal addresses, the date and time, and the exact specification of the type of dump. Like the floating-decimal dump, it displays normalized floating-point numbers. However, the format is different, and unlike the floating-decimal dump, the range of permissible numbers is restricted. The format is: ± 123.45678 . This dump was designed to display only those numbers in the machine which would be likely to be intermediate results in a computation. It singles out floating numbers in the range from 0.000005 through 999.99999. Numbers greater than this range are treated in the same way as the floating-decimal dump handles zeros and unnormalized numbers. Numbers less than 0.000005 appear as 0.00000; numbers greater than 999.9999 are blanked, except when as the last word of tape record. In this case,

the words "LAST WORD" appear. Zeros and unnormalized numbers are treated identically as in the floating-decimal dump, except that zero appears as ± 0 . instead of $\pm 0.$

All dumping output goes onto tape unit 0 and is followed by an end-of-file mark. It can be monitored at any time with sense switch 6.

Since the first 8 cells of core storage are used by the Library, these cells are never included in a core dump. A request for a core dump beginning at cell 0-7 will be assigned an effective first address of 8. Any sections of the first 4 drums may be dumped, although, in general, the Service Library will occupy all drum storage except cells 1428 through 2047 on drum 3 while operating. If the RESCUE section was used, it will be necessary to dump all of drum 3.

It is possible to dump any record, or series of records, of any file on any magnetic tape. As can be seen from the above discussion, any dump of a magnetic tape record will specify the exact length of the record. The tape may be mounted on any tape unit except number 0. As a tape is dumped, the records within the file are numbered consecutively beginning with the first record of the file. The first record is numbered one. End-of-file marks are recorded. Within a record, the words are addressed consecutively, beginning with zero. Each tape dump is designed to operate within a single file. Dumping several files requires several entries. A particular dump is terminated when it finds an end-of-file mark. If the routine was asked to dump more records than the file contained, an alarm is printed, and the fact is displayed on the dump, but the computer does not halt for any corrective measure. The tape check light is used during a tape dump, and the dump displays the upper and lower record addresses of any tape errors which occurred 3 times in succession. It also tells whether the record was written in the binary or decimal (BCD) mode. A tape is accepted as probably in the decimal mode if the check light is turned on during every reading or rereading attempt.

As stated above, the dump format is determined by the prefix. The rest of the code words is interpreted in the same way for all 3 types of dumps.

The type of storage to be dumped is specified in the tag of the first code word. A tag of zero requests core; a tag of 1-4 requests drum 1-4, respectively; and a tag of 5-7 requests a dump of magnetic tape. Only the first code word is needed for tags of 0-6; a tag of 7 requires one secondary code word.

If the tag is zero, the address and decrement portions of the first code word contain the lower and higher address, respectively, of the core section to be dumped. A decrement of zero is identified with the highest

address of the core, and an address of 0-8 is identified with the lowest core address. Thus, a first code word with a prefix of 1, (2, 3) and the rest blank, requests an octal (decimal, fixed-decimal) dump of the entire core. A last address less than 8 is an alarm condition. If a first and last address are given in the decrement and address, instead of the address and decrement, respectively, the routine will automatically switch them. Thus, the order of the addresses is irrelevant unless zeros appear.

For tags of 1-4, the first and last addresses of the section of the drum to be dumped are given in the address and decrement portions of the first code word, respectively. The range of addresses on a drum is 0-2047. A decrement of zero or greater than 2047 is identified with 2047. If non-zero addresses were specified in the wrong order, the routine will switch the lower address to the address portion. If the lower address is greater than 2047 an alarm results.

For tags of 5-7 (tape dumps), the tape unit is specified in the address portion of the first code word. Units 1-9 are the only acceptable units as output from the dump is written to unit 0 (=10). The use of the decrement of the code word differs according to the particular case.

If the tag is 5, the decrement specifies how many records are dumped. The tape is first rewound, and the number of records specified in the decrement are dumped. A decrement of zero means dump the entire first file. The tape is finally rewound.

If the tag is 6, the decrement specifies record positioning. The tape is first positioned according to the decrement, following the positioning convention described in the Data Tape routines. A single record is then dumped. The tape is not rewound, nor is it repositioned.

If the tag is 7, the decrement specifies file positioning. A secondary code word is needed in this case. This option was included to permit dumping any portion of any file on any tape. Since all Library tape-handling routines deal only with an initial single file, this dump was included for the benefit of those who wished to dump multifile tapes written outside of the Service Library, using the Service Library dumps. The tape is initially positioned to the beginning of the current file. It is then positioned according to the usual convention, except that an advance (backspace) of one means advance (backspace) one file, not one record. An alarm occurs if this positioning would run the tape off of the reel. The numbers of the first and last record in the desired file to be dumped are specified in the address and decrement, respectively, of the secondary code word. The first record is numbered one, and an address portion of zero in the secondary code word is identified with one. A decrement of zero means dump until the end-of-file. If two non-zero addresses are specified in the reverse order, the routine will interchange them. After the requested records have been dumped, the tape is not rewound nor repositioned.

The alarms which may occur during a dump are:

ILLEGAL CODE WORD
RESTART TO INSERT CODE WORD MANUALLY

This alarm occurs if a higher address less than 8 is specified in a core dump; if a lower address greater than 2047 is specified in a drum dump; or if a tape unit not in the range 1-9 is specified in a tape dump. Restarting permits the correct code word to be inserted.

TAPE HAD FEWER RECORDS THAN REQUESTED

This alarm occurs if an attempt is made to dump more records within a file than the file contains. This alarm will never occur if a request is made to dump until the end-of-file. The computer does not stop, nor is any corrective action necessary. The above sentence also is displayed on the dump output.

POSITIONING IMPOSSIBLE
RESTART TO INSERT CODE WORD MANUALLY

This alarm occurs in a tag 6 case if the record positioning would advance the tape beyond the end-of-file mark. (It is not possible to detect too many record backspaces.) In this case the tape is repositioned before the alarm.

This alarm may also occur in the tag 7 case if the file positioning would move the tape beyond the end-of-tape mark, or behind the beginning of tape mark. It will also occur if the number of the first record to be dumped in the file is greater than the number of records contained in the file. In the first 2 cases, no repositioning takes place; in the third case, the tape will be left as though only the file positioning occurred.

Restarting permits inserting a different code word. If the programmer is familiar with the repositioning conventions, and has the alarm print-out at hand, any information needed from the tape can easily be obtained by inserting an altered code word.

DRUM FAILURE
RESTART TO TRY AGAIN

When dumping a long tape record, it is necessary to make several reading passes at the tape, reading consecutive sections of the record into the working space until the entire record has been examined. This takes time. In order to reduce the time required, a section of the core directly above cell 2047 is temporarily saved on drum 3. This

permits larger sections of the tape record to be read at each pass. Immediately following the dump, the section of the core is restored. These transfers are check summed, and a check sum failure during either transfer causes this alarm. Restarting attempts to make the transfer again. Experience here shows that the repeated occurrence of this alarm indicates machine or drum malfunction. In the authors' experience, it has never occurred.

Diagrammatically the dump code words are:

<u>Prefix</u>	<u>Decrement</u>	<u>Tag</u>	<u>Address</u>	<u>Code Word</u>
± 1 ± 2 ± 3	See below	0-7	See below	
<u>tag = 0, core dump</u>				
± 1	Higher core address	0	Lower core address	
± 2	(0 = 7777 octal)		(0 = 10 octal)	
± 3			(less than 10 = 10 octal)	
<u>tag = 1-4, drum 1-4 dump</u>				
± 1	Higher drum address	1-4	Lower drum address	
± 2	(0 = 3777 octal)			
± 3	(greater than 3777 = 3777 octal)			
<u>tag = 5, tape dump</u>				
± 1	No. of records to	5	Tape unit number	
± 2	be dumped			
± 3	(0 = whole file)			
<u>tag = 6, tape dump</u>				
± 1	Positioning in terms	6	Tape unit number	
± 2	of records			
± 3				
<u>tag = 7, tape dump</u>				
± 1	Positioning in terms	7	Tape unit number	First code
± 2	of files			word
± 3				
	Last record		First record	Secondary
	(0 = to EOF)		(0 = 1)	code word

Examples of programmed entries are:

TSX 4,4 PON 40,3,100	Dump cells 40 through 100 on drum 3, in the octal format, sneak-off.
TSX 4,4 MTW 4,5,3	Dump the first 3 records of the tape on unit 4, in the decimal format, rewind, don't sneak-off.
TSX 4,4 PTH 8,7, -2 PZE 2,0,4	Backspace the tape on unit 8 by two files, dump records 2 through 4 of the file in the fixed decimal format, sneak-off.
TSX 4,4 PON	Dump the entire core in the octal format, sneak-off.

THE "SPECIAL" SERVICE LIBRARY ROUTINES

The special routines are a set of output routines designed to be used by main programs for writing output from normal computations. These routines are identified by zero prefixes and tags, and addresses of 1-7 in the first code word. The special routines consist of three groups: the print-out routines, the BCD routine, and the time routines.

Print-out Routines

The routines in this group are the Octal Print Out, Floating Decimal Print Out, Eight Column Fixed Decimal Print Out, and Ten Column Fixed Decimal Print Out. They are identified by tags and prefixes of zero, and addresses of 1,2,3, and 7, respectively, in the first code word. These routines are designed to print out a contiguous block of numbers from the magnetic core in the format requested. The output is prepared on tape unit 0 (=10), and the on-line/off-line options are specified according to Table 3. The 3 decimal print-out routines may alternatively perform a different function which is controlled by the decrement of the first code word, and will be explained below.

All of the print-out routines require a single secondary code word. The use of the sign bit of this code word differs according to the routine, but the use of the remainder of the code word is identical for all print-out routines.

The address and decrement portions of the secondary code word contain the addresses of the first and last number to be printed, respectively. A zero decrement is assigned an effective address of 77777 (octal), and an address portion in the range 0-8 is assigned an effective address of 8. If the address given in the address portion is greater than that given in the decrement portion, the block of words will be printed out backwards. Thus, any contiguous block of numbers may be printed out forward or backwards. The exact process of determining effective addresses and the order of printing is as follows:

- 1) A zero decrement portion is replaced by 32767 (= 77777 octal).
- 2) The addresses specified in the secondary code word are positioned so that the larger appears in the decrement portion. If this required a permutation of the addresses, the routine is set to print backwards.
- 3) The larger address is tested for size. If it is less than 8, an alarm occurs.
- 4) The lower address is assigned an effective address of 8 if it is 7 or less.

Unlike the dumping routines, the print-out routines do not omit any numbers in the block specified. Thus, for example, while the decimal dump would omit a large section of unnormalized numbers, a request for a floating-decimal print out of the block would include each and every unnormalized number, even though they would all appear as blanks. This feature enables one to print out tables of numbers with blanks in any desired locations.

Following the writing of the block of numbers, up to 7 blank lines may be written, as specified by a tag of 0-7, respectively, in the secondary code word. Unlike the dumps, an EOF mark is not always written following the output, but is only written when a bit appears in position 1 of the secondary code word. Bit 2 of the secondary code word causes the printing to begin at the top of a new page. A bit in the sign position has no effect for the Octal Print Out routine, but causes the decimal routines to replace plus signs by blanks.

The routines will now be discussed individually.

Octal Print-out

This routine is identified by an address of 1 in the first code word. It prints numbers as 12-digit octal integers, with eight numbers to the line, in the following format (the periods indicate blank spaces):

..124642.357753..124642.357753..124642.357753..124642.357753, etc.

The sign bit of the secondary code word is not examined by this routine. The functions of the remainder of the secondary code word are described above, and the decrement of the first code word is interpreted according to Table 3.

There is a single alarm which may occur in the Octal Print Out routine:

ILLEGAL CODE WORD
RESTART TO INSERT CODE WORD MANUALLY

This alarm occurs when the higher effective address of the block to be printed is less than 8. Restarting permits the manual insertion of the correct secondary code word.

Diagrammatically the Octal Print Out code words are:

Prefix	Decrement	Tag	Address	Code Word
± 0	Output flag	0	1	First code word

-1 = off-line only, no monitoring with sense switch 6.

0 = off-line only, output may be monitored with sense switch 6.

1 = output appears off-line and on-line

2 through -2 = on-line only

Prefix	Decrement	Tag	Address	Code Word
S 1 2	Last address (0 = 77777 octal)	No. of final blank lines	First address (0 = 10 octal)	Secondary

S is not examined

1 = terminal EOF

2 = start new page

Examples of programmed entries are:

TSX 4,4 PZE 1,0,1 PTH 100,0,300	Print as octal integers the numbers beginning at cell 100 and extending through cell 300. Printing appears forward and is not followed by any blank lines; output appears on and off-line, sneak-off. Begin at the top of a new page, and write a final end-of-file mark.
TSX 4,4 MZE 1,0,-1 PZE 600,3,200	Print as octal integers the numbers beginning at cell 200 and extending through cell 600. Printing appears backwards, and is followed by 3 blank lines. Output appears off-line only and cannot be monitored with sense switch 6. Printing does not begin a new page, nor is the output followed by an end-of-file mark, no sneak-off.
TSX 4,4 PZE 1 PON 32760,4,0	Print as octal integers the numbers beginning at cell 32760 and extending through cell 32767. Printing appears forward and is followed by 4 blank lines, sneak-off. Printing begins a new page. No end-of-file mark follows the output, and the output appears off-line only but may be monitored by sense switch 6.

Decimal Print-outs

The 3 decimal print-out routines have an alternative function which is requested by decrements of 8 through -2 in the first code word.

If such a decrement appears, the routines prepare the BCD characters of up to one line of the numbers, and then store the characters beginning at the address specified in the decrement of the first code word. If the addresses specified in the secondary code word request more than a single line of printing, the first line which would be printed is converted to BCD characters and stored. This feature was designed to permit editing of decimal output, or inserting of sentences or other aesthetically pleasing characters into a line of decimal numbers.

As the explanation of BCD characters and the preparation of computer output given in the 704 manual is somewhat lacking in clarity, a brief and concise description of the processes used in the preparation of Service Library output will be given here, to aid in understanding the prepare BCD options of the decimal print-out routines. Although this discussion will not fully illuminate the mysteries of zones and digits, it will suffice to provide an understanding of the techniques used in the preparation of Service Library output.

Numbers, letters, or symbols of any type which are to be printed out during a computation must be stored in the memory of the computer. Since all words in the computer consist of 36 binary bits, it has been necessary to adopt a code by which several bits represent a single symbol. In the conventional code adopted by IBM, 6 consecutive bits represent a symbol and are known as the BCD character for that symbol. When discussing BCD characters, it is more convenient to speak in terms of the 2 octal digits which correspond to the 6 binary bits, and this will be the practice in this discussion. Examples of the BCD character representation are: the plus sign is represented by octal 20, the minus sign by octal 40, numerals from 0 through 9 are represented by the octal numbers 00 through 11, respectively. Even the blank symbol requires a BCD character, which is octal 60.

When output is written from the machine, the first step is to construct or obtain the BCD characters of all symbols which are to appear. When the BCD characters are ready, the procedure followed depends on whether the output is to appear on- and/or off-line. The process required to print a line on-line is somewhat involved, and it is not relevant to the understanding of the prepare BCD options of the decimal print-out routines, so it will not be discussed here. For off-line output the procedure consists of writing the BCD characters onto magnetic tape in a format which is interpreted and printed by the 1401 computer or other off-line auxiliary equipment. Since each BCD character requires 6 binary bits, a computer word may contain up to 6 BCD characters, i.e., each word contains the code for up to 6 symbols. When the SAP assembler finds a card with an operation code of "BCD" in columns 8-10, it divides the symbols appearing in columns 13 ff. into groups of 6 and constructs a word of BCD characters

for each group. Up to 120 symbols may appear on a single line of output, so it is seen that a full line of printing requires 20 computer words of BCD characters.

Actually, only a maximum of 118 symbols ever appear in Service Library output. The first character of the first BCD word is interpreted as printer carriage control by the 1401 computer, and the second character is ignored because of the formats adopted for the Library print-out routines. As will be seen in the discussion of the BCD print-out routine, a request to print out 20 words of BCD will cause all symbols stored in the 20 words to appear, except the rightmost two characters of the 20th word.

Knowledge of the format of the output tapes is also nonessential to the understanding of the operation of the routines, so it will not be explained here.

The decimal print-out routines print floating-point numbers in one of two formats, 8-column or 10-column. In the 8-column format, each line contains 8 numbers; in the 10-column format, 10 numbers appear in each line. When the prepare and store BCD option is used, the number of BCD words prepared depends on the number of floating-point numbers to be converted (at most a full line may be requested) and upon the format specified. In the 8-column format, the BCD characters of 2 floating-point numbers require 5 words of BCD. The first number requires 3 words of BCD, and the second number requires 2 additional words. In the 10-column format, each floating-point number requires 2 words of BCD characters. In either case, the BCD characters of a full line require 20 BCD words. If less than a full line is to be prepared and stored, only the BCD of the numbers specified will be stored. The BCD words are always stored consecutively in cells with increasing addresses. For partial lines, the number of BCD words stored is given in the following table.

Number of Numbers to be Converted	Number of Words for BCD characters	
	10-column Format	8-column Format
1	2	3
2	4	5
3	6	8
4	8	10
5	10	13
6	12	15
7	14	18
8 = full line for 8-column format	16	20
9	18	
10 = full line for 10-column format	20	

The use of all of the secondary code word except the sign bit was explained above. The conventions listed there are correct for all of the decimal print-out routines. The use of the sign bit is also identical for all 3 decimal print-out routines. If a negative sign is given in the secondary code word, plus signs appearing before the decimal numbers are changed to blank spaces. It is sometimes aesthetically desirable, when preparing a table of numbers, to have only the minus signs appearing. If the sign of the second code word is positive, the plus signs appear as indicated below in the print out formats.

There is a single alarm which is identical for all of the decimal print-out routines.

ILLEGAL ADDRESSES RESTART TO INSERT CODE WORD MANUALLY

This alarm may occur for 2 reasons. The first is that the higher effective address of the block of numbers to be printed is less than 8. In this case, restarting permits the manual insertion of a correct secondary code word.

The second cause of this alarm occurs during a request to prepare and store the BCD characters of some numbers. The BCD words stored during such an operation may not be stored in cells higher than the highest cell in the magnetic core, as this would overlap into an important section of the Service Library. If the entry would cause BCD words to be stored beyond the highest cell of the core, this alarm occurs. If the computer is restarted, the secondary code word may be inserted manually. If, on the other hand, the error occurred due to an incorrectly specified starting storage address, in the decrement of the first code word, rather than a request to convert too many numbers, the proper first code word should be inserted into the MQ register and machine control manually transferred to cell 1000 (octal). This is the alarm mentioned earlier in which neither code word alone causes an alarm condition, but together both code words do cause an alarm condition. If the first code word is reinserted after this alarm and if the entry was a manual entry, it will also be necessary to reinsert the secondary code word as was originally done in the entry. Otherwise, only the erroneous code word must be replaced.

Examples of programmed entries to the decimal print-out routines may be mentally constructed from the examples of Octal Print Out entries. It is necessary only to change the addresses of the first code words to 2, 3, or 7. The functions performed by the Service Library will differ only in that the words will be printed in the floating-decimal, 8-column fixed-decimal, or 10-column fixed-decimal format, respectively. Changing the

signs of the secondary code words would cause all plus signs to be replaced by blank spaces. An example of the prepare and store BCD entries would be:

TSX 4,4	Prepare the BCD characters of the 6 numbers beginning
PZE 3,0,600	at cell 4000 and ending at cell 4005. Store the charac-
PZE 4000,0,4005	ters starting at cell 600; sneak-off. The characters are
	in the 8-column fixed-decimal format.

The 3 decimal print-out routines will now be discussed individually.

Floating-decimal Print-out

The floating-decimal print-out routine is identified by a zero prefix and tag, and address of 2 in the first code word. An entry will cause the routine to print out decimal numbers in the following 8-column format (the periods prior to the plus or minus signs designate blank spaces, those after the signs, the decimal point,):

..±.57932468±13..±.24683579±24..±.0 , etc.

The decimal numbers are rounded correctly in the last figure. Zeros appear as ±.0, and unnormalized numbers are blanked out. The blanking of unnormalized numbers permits the construction of numerical tables with particular elements, rows, or columns blanked out. An entry causes a block to be printed out contiguously, so to blank out the desired locations; one simply inserts unnormalized numbers into the cells which will be printed at these locations.

The functions of the various sections of the code words were described in detail above. Diagrammatically the code words are:

<u>Prefix</u>	<u>Decrement</u>	<u>Tag</u>	<u>Address</u>	<u>Code Word</u>
±0	Output flag	0	2	First
	-1 = off-line only, no monitoring with sense switch 6			
	0 = off-line only, may be monitored with sense switch 6			
	1 = off-line and on-line			
2 through	7 = on-line only			
8 through	-2 = Prepare BCD and store at this address			
<u>Prefix</u>	<u>Decrement</u>	<u>Tag</u>	<u>Address</u>	<u>Code Word</u>
S 1 2	Last address (0 = 77777 octal)	No. of final blank lines	First address (0 = 10 octal)	Secondary

S = suppress plus
1 = final EOF
2 = New page

Eight-column Fixed-decimal Print-out

The 8-column fixed-decimal print-out routine is identified by a zero prefix and tag, and address of 3 in the first code word. This routine also prints decimal numbers in an 8-column format, but the numbers are displayed as fixed-point numbers rather than in the floating-decimal scientific form. This routine was primarily designed for displaying final numerical results of a computation. The fixed-decimal format is as follows (the periods prior to the plus or minus signs designate blank spaces; the periods in the numbers designate decimal points):

..±123.45678.....±135.55776 , etc.

or

...±76.33554.....±54.76984 , etc.

or

....±4.44667.....±2.12345 , etc.

or

....±0.14466.....±0.00000 , etc.

These numbers are rounded correctly in the fifth decimal place. As is seen from the format, only numbers in the range from .000005 through 999.999994 appear. Numbers below this range (in magnitude) appear as 0.00000; numbers above this range are blanked out. Zeros appear as ±0., and unnormalized numbers are blanked out. Blank sections in a numerical table are obtained exactly as for the floating-decimal print-out routine, except that numbers greater than 999.999994 as well as unnormalized numbers will cause blank slots.

The functions of the various sections of the code words were described in detail above. Diagrammatically the code words are:

<u>Prefix</u>	<u>Decrement</u>	<u>Tag</u>	<u>Address</u>	<u>Code Word</u>
±0	Output flag	0	3	First
	-1 = off-line only, no monitoring with sense switch 6			
	0 = off-line only, may be monitored with sense switch 6			
	1 = on-line and off-line			
	2 through 7 = on-line only			
	8 through -2 = Prepare BCD and store at this address			

<u>Prefix</u>	<u>Decrement</u>	<u>Tag</u>	<u>Address</u>	<u>Code Word</u>
S 1 2	Last address (0 = 77777 octal)	No. of final blank lines	First address (0 = 10 octal)	Secondary

S = suppress plus signs

1 = Final EOF

2 = New page

These diagrams also represent the ten-column fixed decimal decimal routine, if the address portion of the first code word is changed to 7.

Ten-column Fixed-decimal Print-out

The 10-column fixed-decimal print-out is identified by a zero prefix and tag, and address of 7 in the first code word. It is identical with the 8-column fixed-decimal print-out in every respect, except that 10 numbers appear across the page instead of 8. The format appears as follows:

..±123.45679..±135.55776..±432.12357 , etc.

or

...±67.77953...±45.55773....±4.42686....±0.03357....±0.00000 , etc.

The previous section about the 8-column fixed-decimal print-out routine contains all other information.

BCD Print-out Routine

The BCD routine is identified by a zero prefix and tag, and address of 4 in the first code word. This routine prints up to a single line of BCD characters for each secondary code word given. Unlike the print-out routines, this routine does not manufacture any BCD characters, but simply prints out characters which are already in the memory of the machine. The BCD characters which are printed begin at the third character of the line, so a maximal line may contain 20 BCD words minus 2 characters, or 118 BCD characters.

The output is prepared according to the decrement of the first code word, as listed in Table 3. It may appear off-line only (with or without sense-switch monitoring), on- and off-line, or on-line only. In addition, each line may be followed by up to 7 blank lines, and may begin a new page or be followed by an end-of-file mark on the output tape.

The BCD routine will accept up to 6 secondary code words. Each secondary code word requests a single line. When examining the secondary code words, the routine interprets the first secondary code word with a positive sign as the final secondary code word. Therefore, when several secondary code words are given, all but the final one must have a negative sign. This convention is standard throughout the Library for multiple code word routines. If 6 secondary code words appear, all with negative signs, an alarm occurs.

Bits 1-35 of the secondary code words are interpreted almost exactly as in the print-out routines. A bit in position 1 causes the end-of-file mark to be written on the output tape at the conclusion of the BCD writing.* A bit in position 2 causes the printing to begin a new page. Each line of BCD may be followed by up to 7 blank lines as specified in the tag of the secondary code word. The address and decrement portions of the secondary code word contain the initial and final addresses of the BCD characters to be printed, respectively. Effective addresses are assigned to the address and decrement portions according to the procedure outlined in the print-out routine discussion, except that: 1) a higher effective address of 7 or less is replaced by 8 instead of causing an alarm; 2) BCD characters are always printed forward; if the higher effective address turns up in the address portion, the address and decrement portions are permuted, and the BCD printing appears forward. If more than 20 BCD words are requested by the effective addresses, only the first 20 words, beginning at the lower effective address, will be printed.

A word of caution is necessary at this point. It is possible to print a line of decimal numbers by first requesting one of the decimal routines to prepare the BCD characters, and then using the print BCD routine to accomplish the actual printing. This will not produce the same result as simply printing the numbers with the first entry. As can be seen from the format diagrams of the decimal print-out routines, the lines of numbers always begin with 2 blank spaces, and in some cases the numbers will extend to the rightmost characters of the line. If this line of BCD characters is printed by the BCD routine, 2 additional blank spaces will be inserted at the start of the line, and the 2 rightmost characters will be lost. The prepare BCD functions of the decimal print-out routines were designed for special editing requirements, not as intermediate steps in printing a line of numbers.

The BCD routine has one alarm.

*More exactly, Bit 1 of the final secondary code word of the entry causes the end-of-file. Bits 1 of all intermediate secondary code words are ignored.

TOO MANY CODE WORDS RESTART TO RESTORE CORE AND HALT

This alarm occurs if 6 or more secondary code words with negative signs are given. Restarting causes the Library to sneak-off and halt as after a manual sneak-off entry. No other corrective measure is possible at this point because in a programmed entry the routine has no way of telling which, if any, of the code words was to be the final one. This alarm occurs after the first 6 lines of BCD have been printed.

Diagrammatically, the BCD routine code words are:

<u>Prefix</u>	<u>Decrement</u>	<u>Tag</u>	<u>Address</u>	<u>Code Word</u>
± 0	Output flag	0	4	First

-1 = off-line only, no monitoring with sense switch 6

0 = off-line only, may be monitored with switch 6

1 = on- and off-line

2 through -2 = on-line only

<u>Prefix</u>	<u>Decrement</u>	<u>Tag</u>	<u>Address</u>	<u>Code Word</u>
± 1 2	Last address (0 = 77777 octal)	No. of final blank lines	First address (0 = 10 octal)	Secondary

+ = final secondary code word

- = one or more secondary code words following this one

1 = Write EOF on output tape after printing (if final secondary code word)

2 = Printing appears at top of new page.

Examples of programmed entries to the BCD routine are:

TSX	4,4	Print the BCD words in cells 100 through 104 on-line and
PZE	4,0,1	off-line, sneak-off. Print at the top of a new page, no
PON	100,0,104	final blank lines, no terminal end-of-file mark.

TSX	4,4	Print BCD sections, begin at top of new page, follow with
MZE	4,0,2	terminal end-of-file mark, printing comes on-line only,
MON	80,2,85	don't sneak off. Print the sections: cells 80-85 followed
MZE	86,3,90	by two blank lines, cells 86-90 followed by three blank
MZE	91,0,95	lines, cells 91-95 with no blank lines following, and
PTW	96,6,97	cells 96-97 followed by 6 blank lines.

Time Routines

The time routines are identified by zero prefixes and tags, and addresses of 5 or 6 in the first code word. An address of 5 designates the time print out or prepare BCD routine; an address of 6 requests the time integer routine. Both of these routines were designed to enable the program to communicate with the chronolog clock and make use of the information obtained. The two time routines will now be described individually.

Time Print-out or Prepare BCD

This routine is identified by an address of 5 in the first code word. It requires only the first code word, and the function of the routine is specified by the decrement of the code word. Decrements of -1, 0-7 cause the routine to print out the date and time in the following format: (initial periods designate blanks)

..MAR 24, 1962 11.15 AM

The output appears on-line and/or off-line according to Table 3 (according to the conventions established for the decimal print-out routines). The printing comes at the top of a new page and is followed by a single blank line. The use of this option was intended to be the initiation of output from a computation.

If it is desired to insert the date and time into a sentence, the routine can alternatively prepare the BCD characters of the date and time. To request this function a decrement in the range 8 through -4 is given in the code word. When such a decrement is found the routine prepares the four BCD words:

MAR 24, 1962 11.15 AM..

and stores them beginning at the address specified in the decrement of the code word. If this option is used, no printing occurs. As can be seen from the BCD words, the 2 final blank characters allow the date and time to appear as the rightmost 4 words of a line printed by the BCD routine. Decrements of -3 or -2 cause an alarm as the four BCD words would wrap around the end of the core if they were to be stored starting at either of these addresses.

The alarms which may occur in this routine are:

CLOCK FAILURE
RESTART TO EXIT

If the clock cannot be read properly in 10 tries, this alarm occurs. To get the information from the clock, it is necessary to read it through the on-line printer. Certain check bits are also transmitted, and

an improper check bit means the clock was not read properly. This alarm has never occurred, so experience indicates that it will only occur when the clock is on the blink. Restarting causes the Library to exit according to the entry. Failure of the clock should not be permitted to impede the entire computation.

ILLEGAL CODE WORD

RESTART TO INSERT CODE WORD MANUALLY

This alarm occurs if a decrement of -3 or -2 is given in the first code word. These addresses are illegal because the four BCD words could not be stored starting at either address. Restarting permits the correct code word to be inserted.

Diagrammatically the code word is:

Prefix	Decrement	Tag	Address	Code Word
± 0	Operation flag	0	5	First
	-1 = off-line only, no monitoring with sense switch 6			
	0 = off-line only, monitoring with switch 6 permitted			
	1 = on- and off-line			
2 through	7 = on-line only			
8 through	-4 = prepare BCD characters and store them starting at this address			
	-3 or -2 are illegal			

Time Integer Routine

This routine is identified by an address of 6 in the first code word. It also requires only a first code word. The function of this routine is to prepare the time of day as 4 octal integers, and insert these integers into the decrement of any cell. The particular cell is specified in the decrement of the code word. Only the decrement portion of the cell designated is replaced, the rest of the cell remains undisturbed. Decrements in the range 8 through -1 are legal; but those less than 8 cause an alarm, since these cells are always occupied by Service Library bootstraps.

The time will be stored in the rightmost 4 octal digits of the decrement. The first 2 octal digits represent the hour, and the second 2 digits designate the minutes. The hours run from zero to 24 according to the military convention, and the minutes run as usual from zero to 59. Since 2 octal digits may represent any number from zero through 63, any possible time may be contained in 4 octal digits.

The function of this routine is to permit a program to begin computation and run for a specific length of time automatically. This is done by acquiring the time when the computation begins and at periodic intervals. Whenever the difference between the initial time and the most recently obtained time becomes greater than the length of time the program was to run, the program stops.

The alarms which may occur in the time integer routine are:

CLOCK FAILURE
RESTART TO EXIT

This alarm is identical with the same alarm in the time print-out or prepare BCD routine.

ILLEGAL CODE WORD
RESTART TO INSERT CODE WORD MANUALLY

This alarm occurs if a decrement of 7 or less is given in the first code word. Restarting permits the manual insertion of a correct code word.

Diagrammatically the code word is:

<u>Prefix</u>	<u>Decrement</u>	<u>Tag</u>	<u>Address</u>	<u>Code Word</u>
±0	Address of word to con- tain time integer	0	6	First
	7 or less = alarm			

CHANGING THE YEAR IN THE TIME AND DUMP ROUTINES

The information read from the chronolog clock does not include the year which appears in the time print out, time BCD words, and dump headings. The routines themselves contain this information. Therefore, it is necessary to remake the Service Library each January 1 to bring the information stored in the Time and Dump routines up to date.

When the Library is distributed to a user, 2 additional sets of cards will be furnished. The first set will consist of all single-step load cards; the second set contains the cards used to change the year. The year-change cards are blue and have the year printed in the rightmost 4 columns of the card preceded by an "A," "B," "C," or "D." There are "A," "B," "C," and "D" cards for each year, and all 4 of these cards must be inserted into the Service Library binary deck to change the year. If all 4 year cards in the binary deck are not of the same year, DRUM FAILURE ON ACQUIRING SERVICE ROUTINE alarms will occur when the Time and/or Dump routines are requested.

The Service Library binary deck is punched on brown cards except for an initial green loading card and the blue "A," "B," "C," and "D" cards of the current year. The "A," "B," "C," and "D" cards for the new year replace the "A," "B," "C," and "D" cards of the old year. After the old, blue year cards have been replaced by the new ones, the deck is used to write the self-loading Service Library tape in the usual way. The Time and Dump routines in the newly made version will contain the new year.

CONSTRUCTION OF LOAD CARDS

Single-step Load Cards

A complete set of these cards will be furnished with the Service Library, punched on green cards with the identification printed in columns 73-80 as described below. Additional cards may be made by duplicating those furnished, or from the diagram in Figure 1. The identification code in columns 73-80 is only for convenience; the card will work equally well without it.

			+
			-
			0
			1
			2
			3
			4
			5
RDR 1			6
CPY 1	LDA 6		7
PZE 2042	TXI 1,4,N		8
TXI 1,4,1	CPY 8,4		9
LXA 0,4	CPY 2,4		

Fig. 1. Schematic of the single-step load cards

The routine obtained by a single-step load card is determined by the value of N in the 7rite decrement, as follows (the minus means 2's complement):

<u>N</u>	<u>Routine Requested</u>	<u>Identification code on card</u>
-13	Read SAP information from tape unit number 3	RSAP T3
-12	Read SAP cards	RSAP CDS
-11	Read Binary Cards	RBC
-10	Read first program from program tape on unit number 2	RPT PR1
-9	Read second program from unit 2	RPT PR2
-8	Read third program from unit 2	RPT PR3
-7	Read fourth program from unit 2	RPT PR4
-6	Read fifth program from unit 2	RPT PR5
-5	Read sixth program from unit 2	RPT PR6

An example of a single-step load card is shown in Figure 2. It is a picture of the Read SAP Cards load card, in which the instructions are represented as 12-digit octal integers. Notice that the 7rite decrement contains 77764 which is the 2's complement of 12.

			+
			-
			0
			1
			2
			3
			4
076200000301			5
070000000001	046000000006		6
000000003772	177764400001		7
100001400001	070000400010		8
053400400000	070000400002		9

Fig. 2. Diagram of the single-step load card to read SAP cards

Double-step Load Cards

A double-step load card may be made to supply any conceivable first code word. It has a slight advantage over the single-step card for initiating the dumping routines in that it doesn't require the use of index register 4. It was for this reason that single-step cards were not made for initiating dumps. Double-step cards are quite easy to construct. Only the 9-row of a binary card is needed, and its contents are shown in Figure 3. Figure 4 illustrates the double-step load card used to initiate a complete octal dump of the magnetic core.

			+
			-
			0
			1
			2
			3
			4
			5
			6
			7
			8
TXI or TRA 511	First Code Word		9

Fig. 3. Schematic of the double-step load cards

			+
			-
			0
			1
			2
			3
			4
			5
			6
			7
			8
			9
100000000777	36 37	100000000000	72 73 80

Fig. 4. Diagram of the double-step load card to initiate a complete octal core dump.

TABLE OF ALL CODE WORDS

The tables on the following pages contain diagrams of all possible Service Library code words. It should be remembered that it is the first code word which determines the routine to be executed. In all cases, the sign of the first code word designates the sneak-off or no-sneak-off mode. A positive sign denotes the sneak-off mode, and a negative sign denotes the no-sneak-off mode.

It will be noticed that for cases where several different types of second or secondary code words are possible, the alternatives have been grouped together.

For a more detailed explanation of the code words, the alarms, or the various functions of a routine the reader is referred to the main body of the report.

	PREFIX	DECREMENT	TAG	ADDRESS	
First	± 1 Octal	Last ¹	0 1 - 4	First ¹	CORE DRUM
	± 2 Floating Decimal	No of Records	5	Tape Unit	TAPE
	± 3 Fixed Decimal	Record Positioning ³	6		
		File Positioning ³	7 ²		
Second ²		Last Record ¹		First Record	

Fig. 5. Dump routine code words

PREFIX	DECREMENT	TAG	ADDRESS
± 0	Output Mode ⁵	1	Input Unit ⁴

Fig. 6. Read SAP routine code word

PREFIX	DECREMENT	TAG	ADDRESS	
± 0	Last	2	First	PUNCH
	1 or 0 ⁶		Self Start Address	
± 0	Destination Bias	3		READ

Fig. 7. Binary card routines code words

	PREFIX	DECREMENT	TAG	ADDRESS	
First	± 0	Positioning ³	4	Tape Unit	WRITE
Second		Last		First	
First	± 0	Positioning ³	5	Tape Unit	READ
Second	$\pm 0^7$	Last ⁶	0 4 ⁹	First	
Third ⁷				Record First	

Fig. 8. Data tape routines code words

	PREFIX	DECREMENT	TAG	ADDRESS	
	± 0	Positioning ³	7	Tape Unit	READ
First	± 0	Positioning ³	6	Tape Unit	WRITE
Secondary	$\pm 0^{11,12}$	0	0	0	
	$\pm 0^1$	Last		First	
	± 0	1 or 0 ⁶		Self Start Address	

Fig. 9. Program tape routines code words

	PREFIX	DECREMENT	TAG	ADDRESS	
First	± 0	Output Mode ¹⁴	0	1	OCTAL
Second	S, I, 2 ¹⁶	Last ¹⁶	No of Final Blank Lines	First ¹⁷	
First	± 0	Output Mode ¹⁵	0	2	FLOATING DECIMAL
Second	S, I, 2 ¹⁹	Last ¹⁶	No of Final Blank Lines	First ¹⁷	
First	± 0	Output Mode ¹⁵	0	3	EIGHT COLUMN
Second	S, I, 2 ¹⁹	Last ¹⁶	No of Final Blank Lines	First ¹⁷	FIXED DECIMAL
First	± 0	Output Mode ¹⁵	0	7	TEN COLUMN
Second	S, I, 2 ¹⁹	Last ¹⁶	No of Final Blank Lines	First ¹⁷	FIXED DECIMAL
First	± 0	Output Mode ¹⁴	0	4	BCD
Secondary	S ¹⁸ , I, 2 ²⁰	Last ¹⁶	No of Final Blank Lines	First ¹⁷	

Fig. 10. Print-out routines code words

PREFIX	DECREMENT	TAG	ADDRESS	
± 0	Operation ¹³	0	5	PRINT or BCD
± 0	Destination	0	6	INTEGER

Fig. 11. Time routines code words

Footnotes for Code Word Table

1. A zero in this portion of the code word requests the maximal operation possible. For example, a zero in the decrement of an octal core dump code word is interpreted as the highest core address. A zero in the address portion of the same code word is interpreted as the lowest core address.
2. The second code word is needed if and only if the tag of the first code word is 7.
3. For all tape positioning, decrements from 1 to 16383 mean advance this many units of information, and decrements from -1 to -16383 mean backspace this many units of information. This positioning is always done prior to the tape operation. The tape operation itself will also cause the tape to be advanced by a unit of information (except for the tape dumps). The unit of information on a data tape is the record; on a program tape it is the program. In the tape dumps, the tag 5 case always begins at the first of a tape and leaves it rewound; the tag 6 case first positions the tape relative to its position at the time the dump is taken, and then dumps a single record and leaves the tape positioned behind the record dumped: The tag 7 case also does no final positioning, and for this case the initial positioning unit of information is the file. (A request for no file positioning causes the tape to be moved to the beginning of the file in which it currently stands.)

4. 0 = SAP cards
1-9 = SAP tape mounted on tape unit 1-9 respectively
5. -1 = off-line only, may not be monitored with SSW 6
0 = off-line only, may be monitored with SSW 6
1 = off-line and on-line
2 = on-line only
3 through -2 means no output
6. 0 = Halt and start
1 = Flying start
7. A third code word to specify the first word to be read from the record is given if and only if the sign of the second code word is negative. If no third code word is given the first word to be read from the record is the first data word of the record.
8. Zero in this position means to read the record until the entire record has been read or until a word has been read into the highest core address, whichever occurs first.
9. A 4(5,6, or 7) in the tag position means position the tape according to the decrement of the first code word only. In this case no information is read from the tape, and the routine does not look for a third code word.
10. This is a multiple secondary code-word routine.
11. In the multiple secondary code-word cases, the final secondary code word is designated by a positive sign. All secondary code words prior to the final one must have negative signs.
12. A negative zero code word causes the routine to write the initial 2-word record on the tape. This must be done whenever writing a program onto a blank tape. A positive zero code word means to position the tape according to the decrement of the first code word only. If a zero code word occurs it must be the first secondary code word.
13. -1 = off-line only, may not be monitored with SSW 6
0 = off-line only, may be monitored with SSW 6
1 = off-line and on-line
2 through 7 = on-line only
8 through -4 means prepare BCD characters and store them starting at this address.
14. -1 = off-line only, may not be monitored with SSW 6
0 = off-line only, may be monitored with SSW 6
1 = off-line and on-line
2 through -2 = on-line only

15. -1 = off-line only, may not be monitored with SSW 6
0 = off-line only, may be monitored with SSW 6
1 = off-line and on-line
2 through 7 = on-line only
8 through -2 means prepare BCD characters and store them starting at this address
16. 0 = highest core address (= 77777 octal)
17. 0 = lowest core address (= 10 octal)
18. The sign bit is ignored.
A bit in the 1 position causes an end-of-file to be written on the output tape after the output operation.
A bit in the 2 position causes the printing to begin at the top of a new page.
19. A bit in the sign position causes plus signs to be replaced by blank spaces.
A bit in the 1 position causes an end-of-file to be written on the output tape after the output operation.
A bit in the 2 position causes the printing to begin at the top of a new page.
20. The sign bit is used for the multiple secondary code word designations.
A bit in the 1 position of the final secondary code word causes an end-of-file to be written on the output tape after the output operation.
A bit in the 2 position causes the BCD to be written at the top of a new page.