UNIVERSITY OF ILLINOIS
DIGITAL COMPUTER

LIBRARY ROUTINE X 12 - 235

By Roger H. Farrell
(Preliminary work also
done by Lily H. Seshu
Raymond P. Polivka,
Gene H. Golub.)

TITLE	Symbolic Address Decimal Order Input (SADOI) (SADOI Itself)
TYPE	Input Assembly Program
NUMBER OF WORDS	615 Words in the locked out section of the drum
WORDS USED	999-1023 in the high speed memory 11,058-12,799 on drum
TEMPORARY STORAGE	0, 1

DESCRIPTION

This routine is stored on the drum. It is automatically brought into the Williams memory and is not included on the program tape. It is an input routine designed to read instructions with addresses in decimal or symbolic form. It is compatible with the D.O.I. Any program written for the D.O.I. (library routine X 1) and which refers to the D.O.I. only in standard fashion may be used with SADOI. It may be necessary to recompute sum checks for programs to be input by SADOI.

During input the entire Williams memory is used by SADOI. The instructions in SADOI occupy locations 32-562, 999-1023. The remainder of the Williams memory is used for temporary storage. Instruction pairs of the program being read are initially stored on the drum. The correspondence used is drum address 11756+X ⟷ Williams memory address X. When SADOI is instructed to jump to the program which has been read, words of the program are transferred from the drum to the Williams memory and the jump is executed. A more detailed description is given under interludes.

Initially the reading of a program tape must be started by a hold start or a clear start. In either case Williams memory 2-998 is transferred to drum locations 11,758-12,754. SADOI is then placed in the Williams memory. Input then begins.

Each instruction consists of a pair of function digits and an address. Addresses are written in numerical or symbolic form. Numerical addresses are written and punched in decimal form with non-significant zeros omitted. Thus we write 1023, 35, 7. Each numerical address is followed by a terminating symbol K, S, N, J, F, or L. This symbol indicates a method of treating the numerical address and the instruction. For example, L5 29F and 40 7L. A zero address is indicated by writing a terminating symbol immediately following the function digits. For example, OOK and 26F.

Symbolic addresses are written and punched as a pair of parentheses; any combination of symbols or characters may be placed between the parentheses. Examples of distinct symbolic addresses are ( ), (0), (1), (51372), (WQXR). Generally speaking, any two combinations of teletype characters which produce the same sequence of printed characters will represent the same symbolic address.

Numerical and symbolic addresses are most frequently written to refer to memory locations. When symbolic addresses are used this reference is made by using the same symbolic address twice. One use is as the address of an instruction. The other use is as a marker showing which instruction is being referred to. The instruction 40 (A) says store the contents of the accumulator at (A), using "(A)" as an address of the instruction. We call this an inside symbolic address or inside address. Some other instruction pair will use "(A)" as an outside symbolic address or outside address. This instruction pair might be (A) OOF OOF. When this instruction pair is read by SADOI during input the location at which it is to be stored is remembered by SADOI as the real address corresponding to "(A)". This real address is substituted whenever instructions are written with the inside symbolic address (A).

There is no limit to the number of references that may be made to a particular memory location using symbolic addresses. Further, the word being referred to may appear anywhere on the program tape. When symbolic addresses are used SADOI determines the real addresses completely in terms of the relative positions of instructions on the input tape. The programmer will find that he normally thinks in terms of symbolic addresses while writing a program. Numerical addresses can be written only after bookkeeping that is made unnecessary by writing symbolic addresses. Further, the programmer will find that use of symbolic addresses greatly facilitates making corrections in program tapes. For, instructions often may be added or deleted without having to change addresses.

TERMINATING SYMBOLS

K is a directive symbol. A directive is an instruction written with a pair of sexadecimal characters, a numerical address and terminated by K. It tells SADOI where to store words of program. In addition it provides an address which is called the relativizer. This address is the origin against which relative addresses are measured (see discussion of L, below). The first instruction on a tape must be a directive. The instruction following a directive will always be a left hand instruction. The directive is never stored as part of the program being read.

There are four types of directives which are written as follows:

00 n K with n a non-zero numerical address. The following instruction pairs on the tape will go into locations n, n + 1, n + 2, ... The relativizer is given the value n.

00 K If m is the location of the last instruction pair, then set the relativizer to m + 1. The instruction pairs following will go into locations m + 1, m + 2, ...

01 nK If m is the value of the relativizer then the instruction pairs following will go into m + n, m + n + 1, ... The relativizer is not changed.

02 nK Set the relativizer to the address m of the preceding instruction. The following instruction pairs will go into m + n, m + n + 1, ... When using this type of directive the preceding instruction will most often be a dummy instruction written solely to provide a value for the 02 K directive. For example, 00 (A) 02 K will result in the real address corresponding to (A) being used as the relativizer. When the preceding instruction (00 (A)) appears on the tape as a left hand instruction, it will not be stored as part of the program being read.

The programmer should be careful to use symbolic addresses with the 02 K directive only in cases where the symbolic address is previously used as an outside address.

F means fixed address. Thus L5 29F 40F when executed instructs ILLIAC to put the number at location 29 into A, then, store this number at location 0. The pairs of instructions 00F 00F, 80F 00F, and LL 4095F LL 4095F are particularly useful as they set a memory location to 0, -1, -2^{-39} respectively.

L means relative address. If n is the value of the relativizer, the instruction 50 3L will have address n + 3.

N is used with a jump instruction to cause SADOI to jump to a specified location. Let n be the address of the jump instruction. If n ≤ 998 then the words of program are read back from the drum and put into the Williams memory. Control then transfers to the word at location n. The jump instruction is stored at location 1. It is not otherwise stored as part of the program being read.

If the address n ≥ 999 then a new address m is computed and substituted into the jump instruction. m = 93 + (n - 999). The jump (stored at location 1) is then executed. By this means control may be transferred to any part of the Williams memory during input. The following jump instructions are useful:

24 999N:	Stop input. This instruction has no other effect.
26 999N:	Use to mark an address (see example 4)
26 1000N:	Special interlude marker (see interludes, below)
22 1000N:	Special drum storage order (see same, below)
26 1469N:	Jump to the first word of temporary storage.

Example 1

```
00  100 K
L5  L   82 40 F
OF  F   00 F
26 100 N.
```

In this example instructions are stored at locations 100, 101. Control transfers to location 100. The first pair of instructions prints itself in sexadecimal.

J terminations used after right-hand instructions tell SADOI to convert the pair of instructions into a fraction. It is thus a way of reading a few decimal fractions along with the instructions of a program. For this purpose instructions are written in a special form. For example, the fraction .3333 3333 3333 should be written as 00 F 00 3333 3333 3333J and the fraction .1 should be written as 00 F 00 1000 0000 0000J. The function digits of both instructions are zero and the decimal digits of the fraction are written as the address of the right instruction. Twelve decimal digits must always follow the second pair of zeros except that zeros immediately following the decimal point may be omitted. For example, write .001 as 00 F 00 10 0000 0000J. Any fraction in the range 0 to .9999 9999 9999 may be written in this way.

We can extend the range by using the function digits of the left instruction. By writing 80 for the function digits we can subtract 1 from the fraction. For example, the number.8888 8888 8889 would be written as 00F 00 8888 8888 8889J while -0.8888 8888 8889 = -1 +.1111 1111 1111 would be written as 80F 00 1111 1111 1111J. (see also page 17)

S This termination is used to refer to S-parameters. Discussion of S-parameters is deferred until later.

USE OF SYMBOLIC ADDRESSES

The use of symbolic addresses can be illustrated by the following example. In this example library subroutine P 10 has been rewritten using symbolic addresses. This routine is used to print headings.

Example 2

```
(1)  04F        L5 F
     32 (2)     50 (3)     5th hole character
(2)  26 (4)     50 (1)     non 5th hole character
(4)  00 6F      42 (3)
(3)  02 1F      92 F       Substitute proper address in the 92 instruction
(5)  91 4F      40 F       Enter loop
     L7 F       32 (1)     5-hold delay?  If not, repeat.
     92 707F    26 999F    Exit from loop.  See discussion of interludes.
     26 (5)     26 1N
```

For this example to be complete it should be preceded by a directive. Each pair of instructions referred to in the program is marked by an outside address. Thus, in the first instruction pair (1) 04F L5F, (1) is used as an outside address. Similarly (2), ... , (5) are used as outside addresses. Certain of the instructions use these symbolic addresses as addresses. For example, 32 (2) 50 (3) which instructs ILLIAC to jump to the right side of (2) if the accumulator is positive. Otherwise, put the contents of (3) into Q register.

It should be noted that forward and backward references can be made. The third word contains the instruction 50 (1). This refers back to an instruction pair already read by SADOI. The reference is called a backward reference. In the same word, the instruction 26 (4) makes a forward reference to an instruction pair not yet read by SADOI.

In example 2, location 0 is used for temporary storage. This means that no matter where the block of instructions is located in the Williams memory, location 0 will be used for storing results temporarily during the computation. Thus, fixed addresses are used. In the first word, for example, the instruction L5F refers to this temporary storage location.

Last, addresses of certain instructions do not refer to memory locations. In shift instructions, for example, the address further supplements the function digits by telling how far to shift. In such cases fixed addresses are almost always written.

WRITING SYMBOLIC ADDRESSES

An outside symbolic address is written before the function digits of <u>either</u> instruction of a pair. In example 2, the first instruction pair could also have been punched as

04 F (1) L5 F

or as (1) 04 F (1) L5 F

Further, an instruction pair may have several outside addresses. For example, (1) (2) (3) (4) 40 F (5) 81 4 F.

An inside symbolic address is written immediately following the function digits. Only one inside address may be used in an instruction. <u>It must not be followed by a termination K, S, N, J, F, L.</u>

Constants may be added to inside addresses. Thus, the instruction L5 21 (N12) will refer to the 21st word past the word marked by (N12) as an outside address. Generally the form "n ( )" will be used to add n to the real address corresponding to ( ).

The full range of available teletype signs may be used in writing symbolic addresses: numbers, letters, punctuation marks, etc. During input the characters placed between parentheses are read and converted to numerical equivalents. From these numerical equivalents a <u>symbolic address word</u> is formed.

As stated earlier, two combinations of teletype characters which produce the same sequence of printed characters will represent the same symbolic address. Delays, spaces, and LF-CR characters do not affect the symbolic address. Figures-shift and letters-shift characters, on the other hand, may affect the symbolic address. After reading the beginning parenthesis of a symbolic address, SADOI assumes each character following is in figures-shift until a letters-shift character is read. Thus, input of a symbolic address is not terminated by "A". The symbolic address (1A) would be punched using the following six characters: "(", "1", "LTR-SH", "A", "FIG-SH", ")". The symbolic address (12) would be punched using four characters.

Thus + and K, - and S are different. Similarly the letter "X" and times "x" are different. The number "0" and letter "O", because they look the same when printed, are given the same numerical equivalent when read. N, J, F, L print the same regardless of the print shift. Each is given the same numerical equivalent in both print shifts when read. The letter "O" or number "0" is different from the absence of a character. Thus, ( ), (0), and (1), (01) are four distinct symbolic addresses.

The symbolic address recognized by SADOI consists of the last five printed characters. Thus, "(1A2B3C)" is the same as "(A2B3C)" but different from "(2B3C)".

USE OF TEMPORARY STORAGE LOCATIONS 0, 1 BY SADOI

Location 0 is used by SADOI to compute and store numerical addresses during the input of instructions. At the beginning of interludes it is used to compute a sum check.

Location 1 is used by SADOI to form instruction pairs. Once a pair of instructions has formed a word at location 1, the contents of location 1 are stored as part of the program being input. The only cases in which the contents of location 1 are not stored as part of the program being read is when K or N terminated instructions are read.

The right side of location 1 always contains the most recent instruction read. The left side of location 1 always contains the immediately preceding instruction. Consider the following examples:

<u>Example 3</u>

```
      (a)                    (b)

00 F  26 100 F         26 100 F  26 1N
26 1N                  L5 F  40 F
L5 F  40 F
```

In example 3a, the instruction 26 100F appears as a right-hand instruction. The word formed from 00F 26 100F is stored as part of the program being read. When the jump 26 1N is executed, the contents of location 1 is 26064 26001 (sexadecimal). When input resumes, both instructions L5F and 40F must be read before the contents of location 1 are again stored as part of the program. In example 3b, the instruction 26 100F appears as a left-hand instruction. After the right hand instruction 26 1N is read the contents of location 1 appear as before: 26064 26001. This word is not stored as part of the program because of the N termination. As before, the next pair of instructions, L5F and 40F, are read before the contents of location 1 are stored as part of the program being read.

During input when an inside symbolic address is used and a backward reference is made, the real address is substituted immediately into location 1. If an instruction like L5 21 (N12) is written, the numerical part, 21, has been computed and is stored at location 0. When (N12) makes a backward reference, the real address is substituted. Then, N(0) + N(1) is computed and stored at location 1.

When forward references are made during input, sufficient information is retained to allow SADOI to form the correct addresses later. A pseudo address is temporarily substituted into the word at location 1.

S PARAMETERS

Words stored in locations 2-15 of the Williams memory are called S parameters. These numbers give a way of modifying instructions during input. Following the S termination must be a d = 2, 3, 4, 5, 6, 7, 8, 9, K, S, N, J, F, L. This gives the address (in sexadecimal) of a location which is added to the S-terminated instruction before it is stored in the memory.

The right-hand instruction of an instruction pair stored in locations 2-15 may have an inside symbolic address and still be used as an S parameter. The left hand instruction must not have an inside address if it is to be used as an S parameter.

The directives 00K, 01K and 02K enable the programmer to minimize the number of memory locations needed to store his program. Use of the 02K directive for this purpose is illustrated in example 4. An S parameter is stored at location 5. After the interrupting instructions are read SADOI is set so that the following instructions will be stored immediately after the earlier instructions.

Example 4

```
        L5 F  40 1F
    (A) 26 999N           Introduce address (A)
        00 5K             Store S parameter at 5
        00F  00 12F
        00 (A)  02K       Reset SADOI to begin storing at (A)
        L5 1F  82 S5      Instruction using S parameter
```

The instruction (A) 26 999N has no effect other than to introduce the symbolic address (A).

INTERLUDES

Whenever SADOI is told to jump to instructions which have been read, an interlude begins. This is done by writing an N-terminated jump instruction with address less than 999. The jump instruction may appear on tape as if it were a left-hand or a right-hand instruction. Pairs of instructions may be written. For example, the instructions 26 L 26 1N or 26 (A) 26 1N may be used to start interludes. In these cases, the instruction pair is stored at location 1 only.

The system used with the Decimal Order Input (D.O.I.) in which

```
00F  26L
26 1N
```

is written to begin an interlude may also be used. During input the last two instructions read are stored at location 1. The next to last instruction is always the left-hand instruction of the pair stored at location 1.

There are three types of interludes possible with SADOI.

REGULAR INTERLUDES

Regular interludes happen when SADOI is not explicitly told differently (see below). To get ready for the interlude real addresses are substituted into instructions wherever necessary. After the substitutions are completed drum locations 11,758 to 12,754 inclusive are transferred to Williams memory locations 2-998 inclusive. Control then transfers to the right side of location 1. The interlude begins.

Since at the beginning of a regular interlude substitutions of real addresses are made, the programmer should be careful about putting on tape library routines having interludes. The programmer should put such routines at the beginning of his tape before inside addresses are first used or else convert the interlude to the special interlude discussed below.

Use of a symbolic address as an inside address but not as an outside address is a programming error. SADOI detects all errors of this type before stopping. Information is punched about each error (see below). After all possible substitutions are made and errors indicated, the instructions on the drum are transferred to the William's memory. An FF031 stop then occurs.

The symbolic addresses used prior to an interlude are remembered. This information is stored on the drum during the interlude at addresses higher than 11,000.

A small part of the instructions in SADOI are stored at 999-1023. These instructions enable the programmer to jump to SADOI after an interlude for more input. Two types of jumps may be used. When a jump to the left side of location 999 is made the right hand address of the number in Q is used as a new relativizer. If this address has value m then the following instruction pairs will go into m, m + 1, m + 2,... until a directive is received. When a jump to the right side of location 1014 is made the old relativizer is retained. The right-hand address of A determines where the following instructions will go. If this address has value m the following instruction pairs will go into m, m + 1, m + 2, ...

When a regular interlude is ended by one of these jumps the contents of the Williams memory are put back on the drum. Locations 2-998 are moved to 11,758 - 12,754 respectively. SADOI is then put into the Williams memory, information about the previously used symbolic addresses is restored, and input begins.

SPECIAL INTERLUDES

Special interludes are provided so that a small amount of calculation may be accomplished without requiring substitution of real addresses for symbolic addresses and without transferring all of 1000 words from one memory to another. To write a special interlude precede the first instruction pair to be used by the instruction 26 1000N. The first instruction following an N.termination will always be a left-hand instruction. Follow the last instruction pair by a normal jump instruction terminated by N. Only the instructions lying between the two jump instructions will be brought from the drum for use in the interlude.

Example 5

Input one sexadecimal number

```
00K
26 1000N
80 40F  40L
22 1L  L5 1L
22 1014F  26L
26 1N
0000000000
```

The instruction 26 1000N marks the first word to be used in the special interlude. Let the Williams memory location for this word be n. When 26 1000N is read n is substituted into the right-hand address of the word in 1022. This address will have been set to zero previously, and by testing to see whether or not this address is zero SADOI determines whether a regular or a special interlude is to occur.

Let n_2 be the Williams memory location for the last word of the interlude which is read from the tape. Then the words brought from the drum will be placed in locations n_1, $n_1 + 1$, ..., n_2 in the Williams memory. The programmer should be careful not to store instructions to be used outside this range. This may happen if directives are used between the 26 1000N instruction and the normal jump instruction.

Before a special interlude begins, information about symbolic addresses previously used is stored on the drum. After the interlude this information is brought back to the Williams memory again.

Special interludes, like regular interludes, are ended by jumps to 999 or 1014. The meanings of these jumps are the same as for regular interludes. After the end of the special interlude the S parameters 2-15 are copied onto the corresponding drum locations 11,758 - 11,771. Aside from the S parameters only the words in locations used for the special interlude are copied onto the corresponding drum locations.

FAST INTERLUDES

Fast interludes may be written when 44 instruction pairs or less are needed. During input, blocks of instruction pairs are accumulated in the Williams memory before being transferred to the drum. These transfers to the drum take place after

(1) 45 instruction pairs are accumulated
(2) a directive is read
(3) an N terminated instruction with address less than 999 is read
(4) an S termination is used <u>and</u> the S parameter has a right inside address for which a real address must be substituted.

The addresses used for the temporary storage are 563 - 607. If the jump 26 1469 + k F is written then according to the rule cited earlier the fixed address 1469 + k is changed to

$$(1469 + k - 999) + 93 = 563 + k.$$

Consequently the instruction 26 1469N will cause SADOI to jump to the left side of location 563.

Fast interludes must be ended by a <u>jump to the left side of 93</u>. Input resumes immediately. If the instructions used in the interlude go to addresses m, m + 1, ..., m + k then the instructions following will go into m + k + 1, m + k + 2, ... until a directive is read.

Example 5 is rewritten for a fast interlude and presented in

<u>Example 6</u>

Input one sexadecimal number

```
    00 K
80 40F  40 563F
26 93F  00 F
26 1469N
    01 1K
```

In this example fixed addresses are used throughout. In general the instructions will not go to the same place as that occupied by them when the interlude is executed.

LIBRARY ROUTINES STORED ON DRUM

Certain of the more commonly used ILLIAC library routines have been stored in the locked out section of the drum memory. These routines are, at present,

P16, Y1, N12, R1, S4, S5, T4, T5, F1, A1.

By use of a <u>modified directive</u>, SADOI is instructed to take a specified routine from the drum and insert it in the program being read.

A modified directive like (P12) 00 100K consists of two parts:

(1) An outside address which specifies the library routine.

(2) A directive which specifies the relativizer and first memory location for the program.

Any of the other types of directive may be used. For example, (N12) 00K will cause N12 to be placed following the last instruction read. Similarly, the 01K and 02K type of directives will affect the relativizer and store addresses as described earlier.

When a library routine is obtained using a modified directive, the outside address is inserted into the list of symbolic addresses. The real address which corresponds is the location of the first word of the library routine. Thus, for example, when (N12) 00K is written to obtain N12, the entries to N12 may be written:

```
50 n
50 q
. . .
26 (N12).
```

The programmer may use instructions like L5 21(N12) to refer to words interior to a library routine.

When punching teletype tapes the programmer should be careful how he punches P16, Y1, etc. The letters must appear in letters shift. For example, (P16) is punched as

"(","LTR-SH", "P", "FIG-SH", "1", "6", ")"

using seven teletype characters. Note, also that the use of a hyphen in writing P-16 is incorrect.

Miswriting the designating outside address, or asking for a library routine not stored on the drum will result in an immediate FF 030 stop.

Library routines, stored on the drum, which use interludes (F1 for example) execute their interludes without affecting in any way the symbolic addresses used by the programmer.

SELF CHECKING FEATURES

FF 824 This stop occurs because of a sum check failure in transfers of routines from the locked out section of the drum. Depending on the characters of the input tape last read before the stop, the routine may be

(1) Part of SADOI which was incorrectly placed in the Williams memory

(2) A library routine called for by the input tape which was incorrectly read back from the drum.

FF 030 This stop occurs when the programmer asks for a library routine not stored on the drum. Input stops immediately.

FF 031 This stop indicates that the programmer has misused his symbolic addresses. The nature of these errors is always indicated by output on the punch. The FF stop occurs at the beginning of the 1st regular interlude following an error. At this time all instructions read by SADOI are in the Williams memory.

Three types of coding error are covered by this stop.

S parameters: A word having an inside address is used as an S parameter and the real address corresponding to the inside address has not yet been given to SADOI (by use of an outside address). The punched output appears as S 004 355 H, consisting of "S" which identifies the type of error, two decimal addresses, and the symbolic address involved. The first decimal address specifies the location of the S parameter. The second decimal address specifies the location of the instruction using the S parameter.

If the order was, for example, L5 nS4, then this instruction appears in the Williams memory as L5 nF when the stop occurs.

Outside type error: The same symbolic address is used as an outside address for two instructions having different locations. In this case the punched output appears as:

0 367 528 Q. ·

"O" signifies "outside". The first decimal address is the location of the first instruction, the second decimal address is the location of the second instruction. The symbolic address involved is printed last.

During input, when the real address corresponding to a symbolic address is known, the real address is substituted into instructions immediately. In case of duplicate outside addresses, the most recent real address value is the one saved and used during subsequent input.

Inside type error: When a regular interlude is to begin, real addresses are substituted into instructions. When a zero real address or no real address is available, this type of error is indicated. The punched output appears as:

473 K3.

This first thing punched is the location of the instruction using the inside address. The symbolic address involved is punched second.

FF 032 This stop may occur when a directive appears as a right-hand instruction. The stop indicates that the left-hand instruction which preceded used an inside address which made a forward reference. This may mean that input is out of phase one-half word or that an 02K directive cannot be executed. Input stops immediately.

FF 033 This stop occurs when an N terminated order is read and

(1) it appears as the right-hand order of an order pair, and

(2) the left-hand order of the order pair has an inside address for which the corresponding real address is not known, and

(3) the address of the right hand order is $\leq$ 998.

This error stop may occur at the start of special or regular interludes. The stop is made after the appropriate instructions have been transferred to the Williams memory. There may also be output on the tape punch.

FF 034 This stop occurs when too many instructions of the type AB n (C) are written with n $\neq$ 0. Each time such an instruction is read and the real address corresponding to "(C)" is not known (forward reference), a word is added to the additions list. If the real address is known, the additions list is not increased. This list is limited to 314 entries.

FF 035 This stop occurs when too many distinct symbolic addresses are used. The number of symbolic addresses is limited to 324.

SUM CHECKS

Library routine X-7, used for sum checks, is not well suited for use with SADOI. It is written as an interlude ending with a 36 999F jump. Ending an interlude causes SADOI to reload itself into the Williams memory, thus causing additional drum transfers which may be omitted by using a more efficient sum check.

At the beginning of interludes SADOI automatically computes a sum of the words (order pairs or numbers) transferred from the drum memory to the Williams memory. At the end of the transfer this sum is in location zero and may be used by the programmer for checking.

If the words transferred are $x_1, \ldots, x_n$, the sum computed is defined by the relations

$$s_0 = 0$$

$$s_{i+1} = x_{i+1} + | s_i | \qquad i+1 = 1, 2, \ldots, n.$$

A sum check routine suitable for SADOI may be written as follows and placed at the end of a tape.

```
00996K
L3F 36 (A)          Let (A) represent the
FF F 26 (A)         location of the first
000000F000000F      instruction to be obeyed
26L  26 1N          in the program
```

The last word in this routine is the sum check and should be punched by hand in the form of an instruction after the program is tried the first time. This word is obtained then by reading the contents of the accumulator when the FF stop first occurs.

SPECIAL DRUM LOADING TECHNIQUE

A special instruction has been provided which will allow the programmer to load blocks of instructions directly into any desired drum locations. Instructions to be loaded this way must be written without symbolic addresses. A sum check is not computed.

The following sequence must be followed:

(1) A directive, which sets the relativizer.

(2) Follow the directive by OC nF 22 1000N. The order pairs following on the tape are stored on the drum at n, n + 1, n + 2, ...

(3) To stop the direct loading, write a directive or begin an interlude (regular or special).

Library routines stored on the drum and containing less than 45 words may be transferred by this technique. For example,

```
(R1) 00 100K
00 3000F 22 1000N
```

will put R1 on the drum starting at 3000. The relativizer used is 100.

Special care must be taken when this technique is used. All bookkeeping work of SADOI proceeds as if the instructions being read were being normally stored. Consequently, if instructions previously read used inside addresses and if these instructions would be overwritten were the new instructions still normally stored, then the bookkeeping will be incorrect.

DETAILS OF THE FORMATION OF INSTRUCTIONS DURING INPUT

During input, decimal addresses are computed and stored at location 0. The pair of instructions being formed is always stored at location 1. The following sequence is followed:

(I) Instruction type AB n (C)

1. Put N(1) in A.
2. Read two function digits "A" and "B" shifting A left eight places.
3. Shift A left 12 places and store at 1.
4. Read the address n and store at 0.
5. Read the symbolic address (C).
    a. Forward reference (real address not known).
        (1) Substitute a false address into the right-hand instruction at location 1.
        (2) Test for n = 0. If zero, begin reading the next instruction.
        (3) If $n \neq 0$ put a new word into the additions list. Left most 30 bits are used for the number n. Right most 10 bits determine the symbolic address (c) used. Begin reading the next instruction.
    b. Backward reference (real address known).
        (1) Substitute real address into right-hand instruction at 1.
        (2) Form N(0) + N(1) and store at 1.
        (3) Begin reading the next instruction.

(II) Instruction type AB n D   D a K, S, N, S, F, L termination.

1. Repeat steps 1-4 of I above
2. K, N, F; form N(0) + N(1) and store at 1. In case of F, begin reading the next instruction. In case of K, N, finish execution of the termination.
3. L; form N(0) + N(1) + relativizer and store at 1. Begin reading the next instruction.
4. Sk; form N(0) + N(1) + N(k) and store at 1. Begin reading the next instruction.
5. J; form

$$N(1) + \frac{1}{2}\left[\frac{N(0) - 5 \times 10^{11} \times 2^{-39} + 5 \times 10^{11} \times 2^{-79}}{5 \times 10^{11} \times 2^{-39}}\right] + \frac{1}{2}$$

and store at 1. Begin reading the next instruction.

DATE 7/11/57	RT: 3/5/58; 12/3/58
PROGRAMMED BY R. H. Farrell	
APPROVED BY D. E. Muller	

lgr

PROGRAM NOTES

INTRODUCTION

The following is by no means a complete analysis of SADOI. Some parts of the program are only named while others are analyzed in detail. The following tries to present the ideas which have made SADOI a usuable input routine. Input is something that happens in time. When one understands the general sequence of steps and a few of the minute details one should then be equipped to understand what actually takes place in ILLIAC.

INPUT STARTS

When instructions being read by SADOI are on tape the input taking place is called tape input. When the instructions are on the drum (reading a library routine) the input taking place is called drum input.

The main part of SADOI fills Williams memory locations 30-604. An input start occurs at any point that the main part of SADOI is written into the Williams memory. Immediately after each input start tape or drum input begins.

Interludes are classed as tape controlled or drum controlled according to the type of input at the start of the interlude. Fast interludes are not followed by input starts (and hence may be used to modify SADOI), while regular and special interludes are. SADOI distinguishes four classes of input start:

(1) a hold or clear start

(2) tape controlled regular interludes

(3) tape controlled special interludes

(4) drum controlled regular or special interludes.

During each input start approximately 200 instructions are used to set up SADOI to begin input. The overall sequence followed during input starts is shown in diagram 1.

TAPE INPUT SEQUENCE

To initiate a clear or hold start one character is read by ILLIAC. SADOI assumes this was the first of a pair of function digits and fakes in a zero for this digit. In this case input starts by reading the second function digit. Consequently, a tape may not be started by a symbolic address.

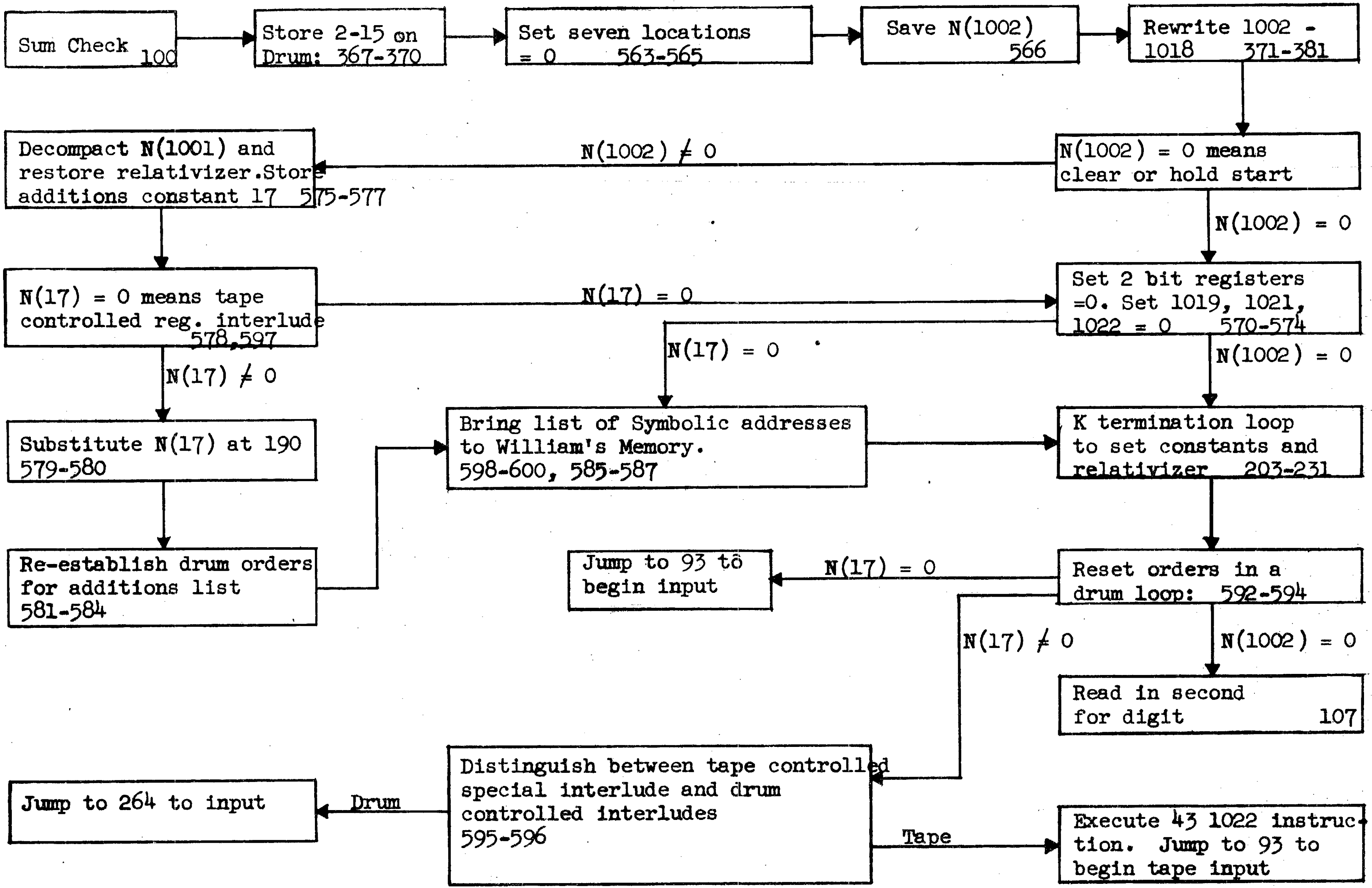
Sum Check 100
Store 2-15 on Drum: 367-370
Set seven locations = 0 563-565
Save N(1002) 566
Rewrite 1002 - 1018 371-381
N(1002) = 0 means clear or hold start
N(1002) ≠ 0
Decompact N(1001) and restore relativizer.Store additions constant 17 575-577
N(1002) = 0
N(17) = 0 means tape controlled reg. interlude 578,597
N(17) = 0
Set 2 bit registers =0. Set 1019, 1021, 1022 = 0 570-574
N(17) = 0
N(1002) = 0
N(17) ≠ 0
Substitute N(17) at 190 579-580
Bring list of Symbolic addresses to William's Memory. 598-600, 585-587
K termination loop to set constants and relativizer 203-231
Re-establish drum orders for additions list 581-584
Jump to 93 to begin input
N(17) = 0
Reset orders in a drum loop: 592-594
N(17) ≠ 0
N(1002) = 0
Read in second for digit 107
Jump to 264 to input
Drum
Distinguish between tape controlled special interlude and drum controlled interludes 595-596
Tape
Execute 43 1022 instruction. Jump to 93 to begin tape input

Diagram 1: INPUT START SEQUENCE

Tape input is started (following interludes) by a jump to location 93 after the set up operations. The jump instructions at 93 provide a variable link to the two main input sections of SADOI. A 26 999N instruction, when read by SADOI, also results in a jump to 93.

The input sections are so called because they contain the instructions that read coded information and translate it into numbers. Various other sections are excluded because their functions are to execute the terminations or bookkeeping. Speed of operation while reading tapes was deemed critical. Consequently, some bookkeeping uses available time between reading of characters on the tape. In other cases similar bookkeeping operations will be found repeated several times in order to eliminate the need for testing which phase of the input was last completed.

I. Tape input section: 43-69, 104-147

A. Read decimal address 43-69

B. Read symbolic address 115-147

C. Read second function digit 104-114

II. Drum input section: 262-327.

The bookkeeping and terminations will be discussed in detail below. The following diagram shows the tape input sequence.

DRUM INPUT SEQUENCE

The maximum compacting by Y4 is to store N words of program in 4/5N locations. Consequently, 36 compacted words placed in the Williams memory will yield at most 45 words of program. For this reason 45 words temporary storage was allowed, locations 563-607. In operation the drum input section brings 36 words from the drum and stores them in locations 572-607. The words are then picked off one at a time and decoded. The resulting instruction pairs are stored in 563-607, starting at 563.

For many library routines stored on the drum 36 words are sufficient to contain the entire routine. When 36 words do not suffice and the current 36 words have been exhausted, the assembled part of the routine is transferred to the drum. Another block of 36 words is put into the Williams memory. The process thus continues until the entire routine has been obtained.

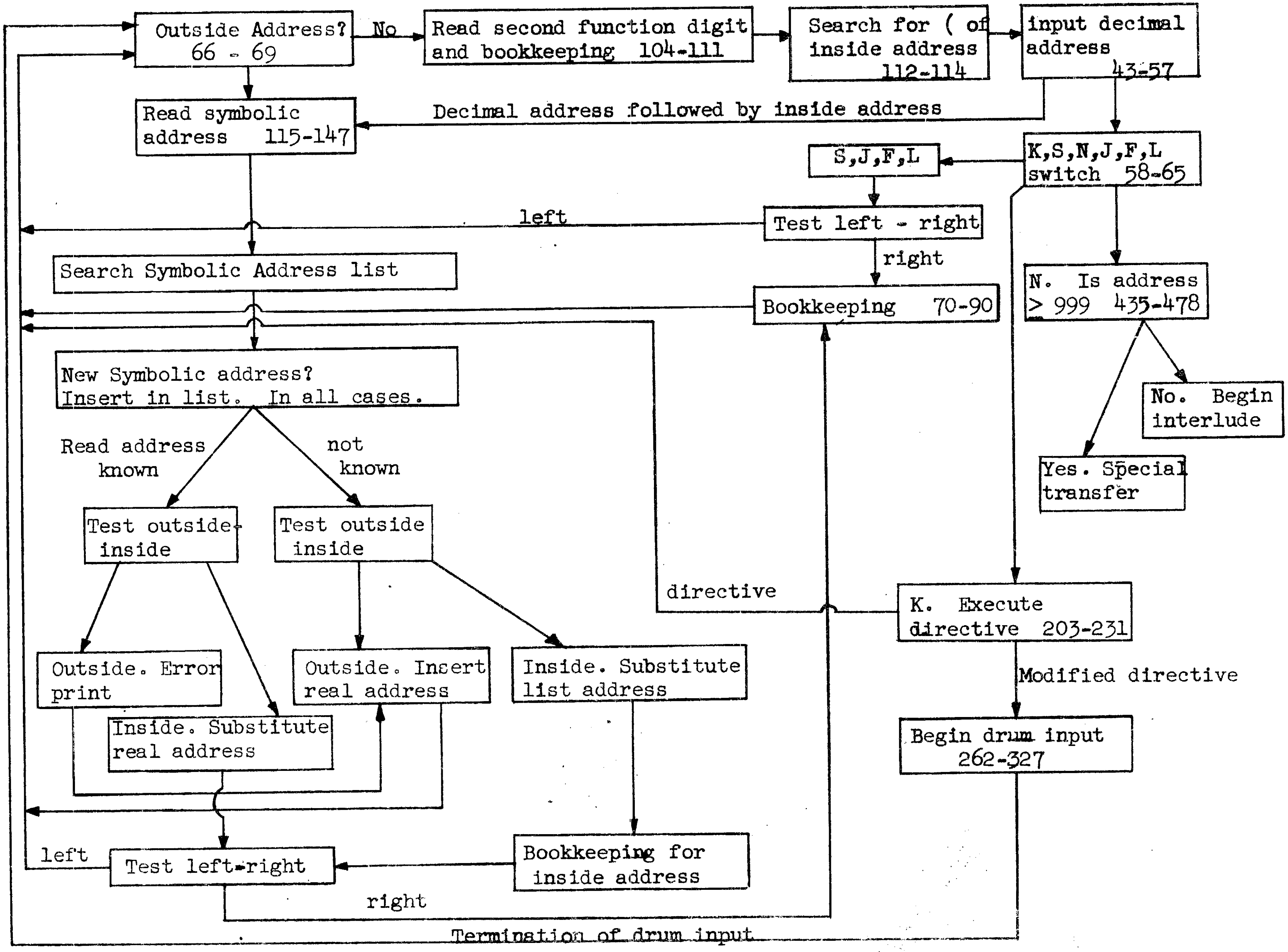
Outside Address? 66 - 69
No
Read second function digit and bookkeeping 104-111
Search for (of inside address 112-114
input decimal address 43-57
Decimal address followed by inside address
Read symbolic address 115-147
K,S,N,J,F,L switch 58-65
S,J,F,L
left
Test left - right
right
Search Symbolic Address list
Bookkeeping 70-90
N. Is address > 999 435-478
New Symbolic address? Insert in list. In all cases.
No. Begin interlude
Yes. Special transfer
Read address known
not known
Test outside-inside
Test outside inside
directive
K. Execute directive 203-231
Outside. Error print
Outside. Insert real address
Inside. Substitute list address
Modified directive
Inside. Substitute real address
Begin drum input 262-327
left
Test left-right
Bookkeeping for inside address
right
Termination of drum input

Diagram 2: TAPE INPUT SEQUENCE

If during drum input an interlude is initiated, certain information must be retained in order to restart the drum input after the interlude. Consequently, locations 1019-1021, 1023 are used for temporary storage by the drum input section. They are set equal zero at all times except during drum input. Their use is

1019: A counter for which of the 5 blocks of 8 bits was last taken.

1020: Word currently being decoded.

1021: Drum order set to bring next word from drum.

1023: Sum check on library routines.

In addition 1022 is used when a library routine calls for a special interlude.

Where possible the drum input section was interwoven with other parts of SADOI. This was done to shorten the program but also to keep the drum input compatible with tape input. Thus the K, S, and N termination sections of SADOI are used during both types of input. The bookkeeping section (69-92) is also used during drum input after each instruction pair of a routine is formed. Because of this sharing

(1) S parameters may be automatically preset during drum input

(2) The special transfers (N terminated jumps with address ≥ 999) may be used during drum input

(3) The various types of directives may be used during drum input

(4) S parameters having symbolic addresses will be properly handled.

The following diagram shows the drum input sequence.

READING SYMBOLIC ADDRESSES

The parentheses "(" and ")" have no effect other than to initiate and terminate reading a symbolic address. Each character on the tape following "(" is examined by means of a 32 position switch (132-147) and has one of the following effects:

(1) None. Spaces, delays and LF-CR characters are skipped.

(2) Change from letters to figures shift or from figures to letters shift. Letters and figures shift characters.

(3) Terminate reading of the symbolic address. Closing ")".

(4) Add six bits to the symbolic address word being formed in ILLIAC

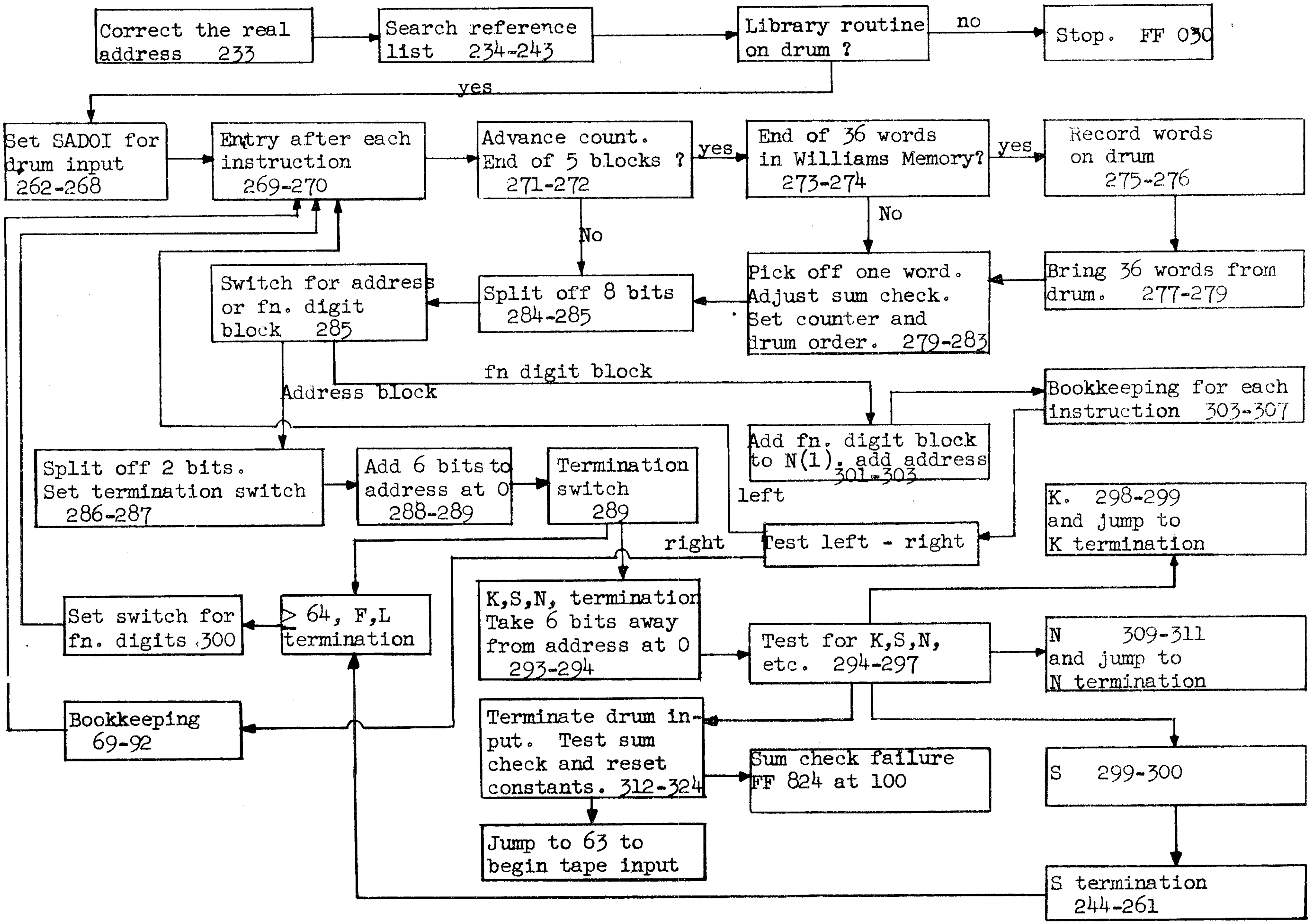
Correct the real address 233
Search reference list 234-243
Library routine on drum ?
no
Stop. FF 030
yes
Set SADOI for drum input 262-268
Entry after each instruction 269-270
Advance count. End of 5 blocks ? 271-272
yes
End of 36 words in Williams Memory? 273-274
yes
Record words on drum 275-276
No
No
Switch for address or fn. digit block 285
Split off 8 bits 284-285
Pick off one word. Adjust sum check. Set counter and drum order. 279-283
Bring 36 words from drum. 277-279
fn digit block
Address block
Bookkeeping for each instruction 303-307
Add fn. digit block to N(1), add address 301-303
Split off 2 bits. Set termination switch 286-287
Add 6 bits to address at 0 288-289
Termination switch 289
left
right
Test left - right
K. 298-299 and jump to K termination
Set switch for fn. digits 300
> 64, F,L termination
K,S,N, termination Take 6 bits away from address at 0 293-294
Test for K,S,N, etc. 294-297
N 309-311 and jump to N termination
Bookkeeping 69-92
Terminate drum input. Test sum check and reset constants. 312-324
Sum check failure FF 824 at 100
S 299-300
Jump to 63 to begin tape input
S termination 244-261

Diagram 3: DRUM INPUT SEQUENCE

In general any character (except parentheses) which will cause a printed mark causes the addition of six bits to the symbolic address word being formed. The composition of the block is as follows: let $b_0b_1...b_5$ be the digits of the block. Then

b_0	$= 0$	figures shift
	$= 1$	letters shift
b_1	$= 0$	non-5-hole character
	$= 1$	5-hole character.

$b_2,...,b_5$ correspond to the 8, 4, 2 and 1 hole positions of the tape respectively and $b_i = 1$ when the position is punched, $b_i = 0$ when the position is not punched.

Certain exceptions mentioned in the description of SADOI are made to this rule.

The most recently added block of six bits is always added to the symbolic address word on the left end by means of a right shift into Q. Part of this right shift is done by the 90 instruction at 118. The shift of four positions right occurs before the new digits read from tape are gated into A. Since the 90 instruction does not clear A, care is taken to make sure N(A) = 0 after the right shift. In particular the constant $N(34) = -1 + 16 \times 2^{-39}$ which is added to N(A) for 5-hole characters eliminates the minus sign introduced by the 90 instruction.

SYMBOLIC ADDRESS COMPILER

The symbolic address word is formed by successive right shifts into Q. The first step following the reading is to execute the following instructions:

```
SJ F   00 1F
40 946F   43 946F
89 1F   L4 946F
40945F.
```

These instructions complete the formation of the symbolic address word and stores it temporarily at 945, the head of the symbolic address list. The word with sign digit changed is stored at 946.

Each time a symbolic address is read the list is searched to determine whether a corresponding symbolic address word is in the list. In long programs the searching time becomes significant since the total time is an n^2 process. The arrangement used in SADOI results from the following arguments (see words 153-162).

(1) When coding for ILLIAC loop speeds are faster when addresses are increased than when decreased.

(2) If a symbolic address was previously used it was most likely one of the more recently received addresses. Therefore the search through the list should begin at the end of the list containing the most recently received symbolic address words.

(3) The first symbolic address word is therefore stored at 944. Each new word is stored at the next lower address.

(4) By storing each new word temporarily at 945,946 and beginning the search at the low address end, the search will always be automatically terminated, either by finding a word in the list or by finding the new word at the head of the list.

(5) A loop to accomplish the search requires 3-1/2 words with a subloop of 2-1/2 words. Read around ratio for this loop was well above 200. It is consequently duplicated to reduce the read around ratio value.

The right address position of the symbolic address word is used to store the real address corresponding to the symbolic address. When examining a symbolic address word SADOI determines whether the real address has been received by whether the right-hand address is non-zero or not.

When a symbolic address word is already in the list, then, when the search terminates, the right address of N(A) is the right address of the symbolic address word. This address is saved by executing a 42 17F instruction.

An inside symbolic address makes a forward reference when the real address is not known and a backward reference when the real address is known. These cases are treated separately. At some point real addresses must be substituted wherever inside addresses were used. This is done immediately when the real address is known. The compiler thus treats the following cases:

I. Real address known N(17) ≠ 0

A. Symbolic address read was an outside address. Then an error has been made. (165-166). Indicate error and proceed as in II B.

B. Inside symbolic address was read. Substitute real address into right address of N(1). Then form N(0) + N(1) and store at 1. (172-174)

II. Real address not known

A. New symbolic address. Determine whether word found in the list was N(945). If so put N(945) at the bottom of the list and proceed as in II B or II C.

B. Outside symbolic address was read. Substitute the real address into the symbolic address word.

C. Inside symbolic address was read

1. N(0) = 0 so no constant is to be added. Substitute into N(1) the location of the symbolic address word in the Williams memory (a number ≥ 512).

2. $N(0) \neq 0$. Form a new word composed as follows: The left 30 bits equal the right 30 bits of N(0); the right address is the location of the symbolic address word. Store this word on the drum in the additions list until needed. Substitute into N(1) the location the additions word will have in the Williams memory when it is used (a number < 512).

BACK SUBSTITUTION PROCESS (479-562)

52 words called 2 bit registers are used by SADOI to keep a record of where real addresses must be substituted later. A digit 1 means a substitution must be made while a digit 0 means no substitution is required. One register thus serves 40 instructions or 20 instruction pairs that have been read. Let an instruction pair have location n (the address it will ultimately have in the Williams ,memory) and let

$$n = 20q + r \quad \text{with} \quad 0 \leq r < 20.$$

Then 947 + q is the location of the 2 bit register. After each directive this constant is computed by the K-termination section (see 217-226) and stored at 26. If $a_0 a_1 \ldots a_{39}$ is the 2 bit register then a_{2r} is the digit position for left instructions, a_{2r+1} for right instructions.

Throughout input a counter word is stored at location 25. This word has one non zero digit a_i where $i = 2r$ or $2r + 1$ according as the instruction being read is a left or right instruction. The counter word is used to add or delete non zero digits from the 2 bit registers as each new instruction is read.

During the back substitution for real addresses the 2 bit registers are successively tested to determine whether they are equal zero. When a non-zero 2 bit register is found the corresponding 20 words are brought from the drum. The required substitutions are made and the 20 words replaced on the drum.

The parts of SADOI required during the back substitution are the temporary storage locations 16-29, the numerical constants 30-42, and instructions 357-562. The list of additions words stored on the drum are transferred to the Williams memory overwriting locations 43-356 (the upper limit is variable). The list of symbolic addresses is then transferred to the drum overwriting the additions list (which is not to be saved). The back substitution then begins.

When a substitution is required the address n of the instruction involved is examined:

(1) $n < 512$ means an addition is required.

(2) $n > 512$ means no addition is required.

The following diagram shows in detail the operation of instructions 479-562.

USE OF THE WILLIAMS AND DRUM MEMORIES AND SUM CHECKS

SADOI is permanently stored at locations 1027-1641 on the drum. It is set so that the first word of the reference list is stored at 1850. To change this address it is necessary to change the drum instruction at 1252 and adjust the sum check at 1612.

When a clear or hold start is used the drum bootstraps load locations 1614-1640 into the Williams memory locations 999-1025. Control transfers to 1020 of the Williams memory. The words at 1019-1025 are used during hold or clear starts to restore Williams memory locations 2-32 after the drum bootstraps have finished their work. They are not reloaded into the Williams memory during other input starts.

The sum check at 1641 is used by the drum bootstrap routines but not otherwise. The sum check at 1612 is used each time the main part of SADOI is put into the Williams memory. Let $N(n) = X_n$ be the number stored at location n while the main part of SADOI is being brought to the William's memory. The drum loop computes the following sum

$$Y_{31} = X_{31}$$

$$Y_{n+1} = X_{n+1} + |Y_n|$$

The values Y_n are stored at 31. Let $X_{603} = - |Y_{602}| + X_{604}$. X_{603} is the sum check constant stored on the drum at 1612. X_{604} is the last word brought from the drum. Therefore $Y_{604} = X_{604} + |Y_{603}| = X_{604} + | X_{603} + |Y_{602}|| = X_{604} + |X_{604}|$. Since X_{604} is a negative number, $Y_{604} = 0$.

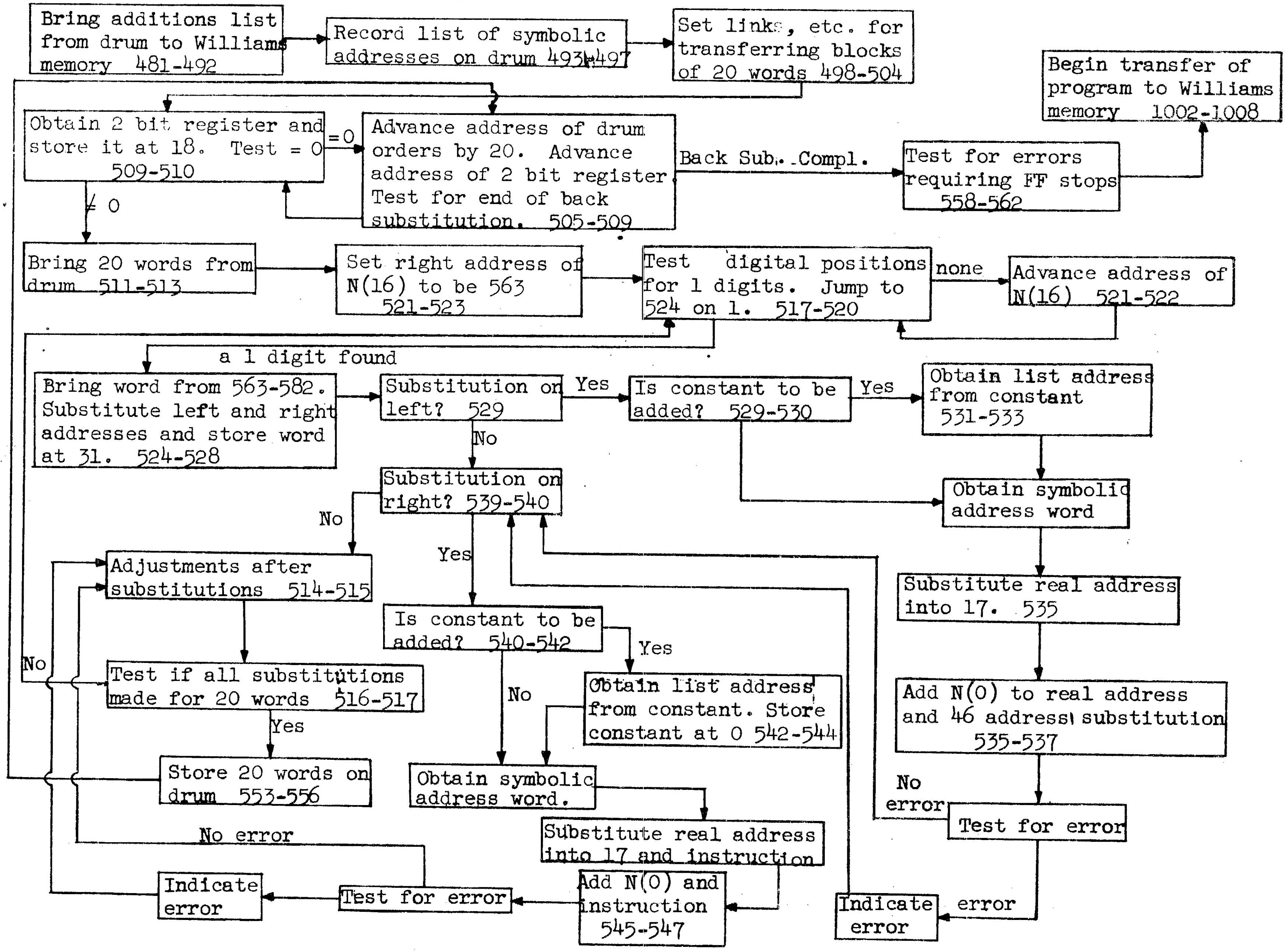
Bring additions list from drum to Williams memory 481-492
Record list of symbolic addresses on drum 493-497
Set links, etc. for transferring blocks of 20 words 498-504
Begin transfer of program to Williams memory 1002-1008
Obtain 2 bit register and store it at 18. Test = 0 509-510
= 0
Advance address of drum orders by 20. Advance address of 2 bit register. Test for end of back substitution. 505-509
Back Sub. Compl.
Test for errors requiring FF stops 558-562
≠ 0
Bring 20 words from drum 511-513
Set right address of N(16) to be 563 521-523
Test digital positions for 1 digits. Jump to 524 on 1. 517-520
none
Advance address of N(16) 521-522
a 1 digit found
Bring word from 563-582. Substitute left and right addresses and store word at 31. 524-528
Substitution on left? 529
Yes
Is constant to be added? 529-530
Yes
Obtain list address from constant 531-533
No
Substitution on right? 539-540
Obtain symbolic address word
No
Yes
Adjustments after substitutions 514-515
Substitute real address into 17. 535
Is constant to be added? 540-542
Yes
No
Test if all substitutions made for 20 words 516-517
No
Obtain list address from constant. Store constant at 0 542-544
Add N(0) to real address and 46 address substitution 535-537
Yes
Store 20 words on drum 553-556
Obtain symbolic address word.
No error
Test for error
Substitute real address into 17 and instruction
No error
Add N(0) and instruction 545-547
Indicate error
Test for error
Indicate error
error

Diagram 4: BACK SUBSTITUTION

During input instruction pairs are stored on the drum. Locations 11,756-12,799 are allocated for this purpose. The correspondence used is location X $\longleftrightarrow$ drum address 11,756 + X. The bookkeeping of SADOI is so set that any location $0 \leq X \leq 1039$ may be used. However, only words having locations $2 \leq X \leq 998$ will be brought from the drum during interludes. The extra words which are available on the drum can be obtained only by special coding.

Storage locations with addresses $11,000 < u \leq 11,755$ are used to store the various lists computed by SADOI.

During input the only other material stored on the drum are the additions constants. Small blocks (13 words) are held temporarily in the Williams memory at 608-620. When a full block is accumulated the words are transferred to the drum reversing their order during the transfer. The first block of 13 are stored at 11,742-11,754.

During interludes the manner in which the various lists are moved depends on the type of interlude.

I. Regular tape controlled interludes
The additions list and 2 bit registers are not saved. The list of symbolic addresses is stored on the drum. The order of the list is reversed during the transfer so that the low address end of the list is stored at 11,754.

II. During all other types of interludes (excluding fast interludes)
The three lists are stored on the drum. The additions list, already there, is not moved. The Williams memory from the low end of the symbolic address list to 998 is stored on the drum "below" the additions list, the order being reversed during the transfer. The combined length of 2 bit registers, symbolic address list and additions list cannot exceed 700 words.

The use of the Williams memory is:

0. Sum check and decimal addresses computed during input

1. Instruction pairs being formed

2-15. S parameters

16-31. Temporary storage for SADOI

16. General purpose temporary storage

17. During input of symbolic addresses, $N(17) = 32 \times 2^{-39}$ for letters shift, = 0 for figures shift. Used only for right address substitution otherwise.

18. Used during reading of symbolic addresses and back substitution.

19. General purpose storage

20. = 0 if no outside addresses for current instruction pair. Otherwise contains the location of the most recently received outside address for the instruction pair.

21. ≠ 0 when reading outside addresses
= 0 when reading inside addresses

22. A word with at most 2 non zero digits. These digits are the positions in the 2 bit register corresponding to current instruction pair.

23. Bits to be added to the two bit register

24. Temporary storage for the current 2 bit register

25. Counter word

26. Address for current 2 bit register in list

27-28. Set aside for use by the error print sections

29. Counter word delayed one right shift

30. The constant zero

31. N(31) = - 1/2 when reading right instructions
= 1/2 when reading left instructions

32-562. Instructions of SADOI

563-604. Instructions of SADOI used only during input starts

563-607. Temporary storage for blocks of 45 instruction pairs

43-357. William's memory storage for additions list during back substitution

608-620. Temporary storage while 13 additions words are accumulated

621-944. List of symbolic addresses

945-946. Temporary storage for symbolic address word during search of the list

947-998. The 2 bit registers

999-1025. The part of SADOI put into the Williams memory by the drum bootstrap routines

1001. Relativizer

1019-1025. Used only during input starts after clear or hold starts to restore 2-32 of the Williams memory. Otherwise

1019. = 0 except during drum input. Used for counter

1020. = 0 except during drum input. Store word being decoded

1021. = 0 except during drum input. Store drum order

1022. Used to indicate errors have been made. Also used to indicate special interludes

1023. Use to store accumulated sum during computation of sum check when reading back a library routine.

1002-1018. Is spoken of as the control block. These locations are rewritten at <u>each</u> input start.

84. Right address = location for the current instruction pair

86. Right address = location in 563-607 where next instruction pair will be stored temporarily

180. Right address = one less than the location in 608-620 where next additions word will be stored

190. Right address = one less than the location the next additions word will have in the Williams memory.

999. Right address = location of last word in symbolic address list.

BOOKKEEPING

Each new instruction (except K and N terminated) is treated by SADOI as overwriting instructions already stored. The memory location may be zero as after clear starts. This case is also included. The 2 bit registers must therefore be modified continually during input in order to keep track of where substitutions of real addresses must be made.

Since a word is added to the instructions being accumulated on the drum only after complete instruction pairs are received modification of two positions in a 2 bit register is made simultaneously after the complete instruction pair is received.

After the instruction pair is received the previous two digits of the register are irrelevant. These positions are therefore set to zero. New 1 digits are then added for each instruction of the pair that will require a substitution later.

Accumulation of information for the bookkeeping is accomplished by use of five storage locations 22-26.

26. The location of the current 2 bit register

24. Temporary storage of the 2 bit register (to avoid many address substitutions).

25. The counter word. This word is as described earlier. During input of the function digits it is shifted right one shift. A copy of the unshifted word is stored at 29.

22. Before the right shift of N(25), the sum N(22) + N(25) is formed and stored at 22. After two instructions are read N(24) -[N(24) · N(22)] sets two positions of the 2 bit register = 0, where "." means logical and.

23. For each instruction which requires a substitution later, N(23) + N(29) is formed and stored. The total correction of the 2 bit register after an instruction pair has the form N(24) -[N(24) · N(22)]+ N(23). See 69-73

When the right instruction of a pair is terminated by K or N N(1) is not stored as part of the program being read. Modification of the 2 bit register does not take place. However, N(22) and N(23) are computed. Since N(23) ≠ 0 means the use of a forward reference in the instruction pair the test N(23) ≠ 0 results in the FF032 and FF033 stops.

During input starts, after K, and after instruction pairs are read, N(22) and N(23) are set to zero. N terminations with addresses ≥ 999 get special treatment. It was decided that before executing a special jump SADOI should be reset so that the next instruction would be a left instruction and in general that the N terminated instruction was not read. Words 439-444 therefore

(1) back up the counter word to its previous position,

(2) destroy information about inside and outside addresses having been used:
N(20) = 0 N(22) = 0 N(23) = 0,

(3) set for left-hand instruction: N(31) = 1/2.

The 2 bit register held at 24 is replaced in the list when K, S, N terminations are read or when N(25) = 0.

S TERMINATIONS

Use of S parameters is made almost unnecessary by the details of SADOI. They were included, however, to make SADOI compatible with the D.O.I. In certain cases it is convenient not only to use S parameters but to have a facility allowing use of inside addresses. Some provision is made for this in SADOI.

First, any number which may be used as an S parameter is stored simultaneously in the Williams memory and on the drum. These are the words having locations 2, 3, ..., L. If the number is <u>not</u> used as an S parameter then it may have inside addresses on both sides which make forward references. Backward references never cause difficulty because the real address is substituted immediately. If it <u>is</u> used as a S parameter only the right instruction may use an inside address making a forward reference.

When an S termination is written SADOI checks the 2 bit register at 947 to determine whether a substitution is needed. The current 2 bit register at 25 is replaced in the list just to make sure the register at 947 is actually there. In case a substitution is needed, then the 2 copies of the S parameter must be changed. The instruction pairs (563-607) being held temporarily in the Williams memory are put on the drum to make sure the second copy of the S parameter is actually on the drum. SADOI then determines whether the required real address has been received. If so corrections are made. Otherwise, an error is indicated by the tape punch.

After special interludes which end with 999 or 1014 jumps to SADOI locations 2-15 are transferred to the corresponding drum locations regardless of the locations specified for the special interlude.

LOCATION	ORDER	NOTES
1027	40 1012	Words of bootstrap stored on drum at 1027-1039. These set up a loop at Williams memory locations 604-609 to read main body of SADOI into the Williams memory. Remainder of bootstrap is in Williams memory 1013-1016. Start at 1013.
	26 1012	
1028	F5 607	
	40 1015	
1029	40 1016	
	26 1012	
1030	40 604	
	26 607	
1031	00 0	
	F5 607	
1032	40 605	
	26 607	
1033	40 607	
	L5 608	
1034	40 606	
	26 607	
1035	L4 604	
	46 608	
1036	40 609	
	26 607	
1037	40 31	
	22 604	
1038	40 608	
	26 607	
1039	40 31	
	L6 31	
1040	00 0	First words of main part of SADOI. Stored at 31, 32.
31	00 0	
1041	00 0	
32	00 2	

WILLIAMS MEMORY LOCATIONS ONLY ARE INDICATED. DRUM
LOCATIONS ARE 1009 GREATER THAN WILLIAMS MEMORY LOCATIONS

```
LOCATION        ORDER                         NOTES

  30     00    0     00    0
   1     00    0     00    0          Numerical Constants
   2     00    0     00    2
   3     00    0     00   10
   4     80    0     00   16
   5     00    0     00   20
   6     00    0     00   32
   7     00    0     00   40
   8     00    0     00  999
   9     741701      282048

  40     7L4095      LL3072
   1     7L4095      LL2048
   2     00    0     00  132
   3     40   16     41    0          ________Read in Decimal Address ________
   4     49   21     L5   16          Set up
   5     L0   33     36   56
   6     L4   33     40    0
   7     91    4     32   48          Read Loop
   8     26   53     L0   33
   9     36   56     10    3

  50     F4    0     00    2
   1     F4    0     00    1
   2     40    0     26   47
   3     F0   32     40   16          Check 5-hole characters for
   4     L7   16     36   47          start of inside address
   5     22  115     FF   53
   6     F4   56     42   57          Set switch
   7     L5    0     26    0
   8     L5   70     22   74          K
   9     L5  251     22   74          S
```

```
LOCATION        ORDER                        NOTES
  60       L5 158     22  74         N
   1       50  39     22  97         J
   2       L4   1     26  64         F
   3       L41001     L4   1         L and final operations of
   4       40   1     L1  31            J and S.
   5       40  31     32  69         Left-right switch.  Jump after right inst.
   6       91   4     36 104         Search for (of
   7       F0  32     40  16         outside address.
   8       L7  16     36  66
   9       26 116     50  24

                               Bookkeeping
  70       J0  22     -1 203         Modify current 2-bit reg. after
   1       L4  24     L4  23         inst. pair
   2       40  24     L3  25         Test counter word for zero
   3       36  74     22  81
   4       F5  74     42  77         Replace 2-bit register in list
   5       L5  26     42  76         when counter = 0 or for K, S, N
   6       L5  24     40   0
   7       50  30     26   0
   8       89   1     40  25         When counter word = 0, get next
   9       F5  26     42  80         2-bit register and reset counter word.

  80       42  26     L5   0
   1       40  24     41  22
   2       41  23     L5  84         Store S parameters in Williams  Memory.
   3       L0  91     36  85
   4       L5   1     40   0
   5       F5  84     40  84
   6       L5   1     40 563         Store inst. pair temporarily at
   7       41  20     F5  86         563 - 607
   8       40  86     L0  92
   9       32  90     50  89
```

```
LOCATION        ORDER                        NOTES

  90      26 357     26  66
   1      L5   1     40  16
   2      75   1     40 608
   3      26  66     22  63          26 999N
   4      22 101     26  95          26 1000N
   5      L5   1     10  20          Set drum input
   6      L4 103     40 362
   7      26  93     L0  39          J termination
   8      66  39     10   1
   9      -J   0     22  63

 100      L3  31     32 361          Sum check test
   1      FF2084     L5  84          Set special interlude address
   2      421022     26  93
   3      86  11
   4      10   4     L5  25          Read second function digit;
   5      40  29     L4  22          Save old counter word and
   6      40  22     L5   1          advance counter word one
   7      00   4     80   4          shift.
   8      50  30     00  12
   9      40   1     L5  25

 110      10   1     32 111
   1      L4  25     40  25
   2      91   4     36  43          Search for (of
   3      F0  32     40  16          inside address.
   4      L7  16     36 112          ---------Input Symbolic Address---------
   5      41   0     41  21
   6      L1  42     40  18          Set for fig. shift
   7      41  17     51  17
   8      10   2     90   4          Core of input loop
   9      L4  42     42 120
LOCATION      ORDER                NOTES

 120     L4  18     26   0     Jump to switch
   1     L0  17     26 118     Entry for N, J, F, L
   2     L4  34     26 118     Entry for "normal" 5-hole char.
   3     L5  36     40  17     Change of letters-figures shift;
   4     L0  42     40  18     entry when character is skipped.
   5     01   4     22 118
   6     40  16     L3  17     ) or A
   7     36 148     L5  16
   8     26 122     40  16     zero or P
   9     L3  17     36 131

 130     L5  16     26 118
   1     F5  37     26 118
   2     32 128     26 125     0  delay      P  delay
   3     36 118     26 122     1  S          Q  D
   4     36 118     26 125     2  CR-LF      W  CR-LF
   5     36 118     26 122     3  (          E  B
   6     36 118     26 123     4  Ltr.Sh.    R  Ltr.Sh.
   7     36 118     26 122     5  ,          T  V
   8     36 118     26 126     6  )          Y  A
   9     36 118     26 122     7  /          U  X

 140     36 118     26 125     8  delay      I  delay
   1     36 118     26 122     9  =          O  G
   2     36 118     26 122     +  .          K  M
   3     36 118     23 123     -  Nr.sh.     S  Nr.Sh.
   4     36 121     26 122     N  '          N  H
   5     36 121     26 122     J  :          J  C
   6     36 121     26 122     F  x          F  Z
   7     36 121     26 125     L  space      L  space
   8     -J   0     30   1     ---------Symbolic Address List Comp-------
   9     40 946     43 946
```

```
LOCATION        ORDER                         NOTES

 150      89   1     36 151
   1      L4 946     40 945
   2      L5 999     22 153
   3      F5 160     42 154        Search list of symbolic addresses.
   4      L1 946     L4   0        Note double loop because of high
   5      36 159     L4  40        read around ratio.
   6      36 159     42  17
   7      L5 154     42 160
   8      26 163     00 435
   9      F5 154     42 160

 160      L1 946     L4   0
   1      36 153     L4  40
   2      36 153     42  17
   3      L3  17     36 167        Test real add. = 0
   4      L3  21     32 172        Test inside-outside
   5      L5 160     42  20        Outside. Word in list has
   6      42 171     26 328        non-zero address.
   7      L5 160     L0 198        Test for new sym. Add.
   8      36 192     L3  21        Test inside-outside.
   9      36 175     L5 160        1st time outside. Substitute

 170      42 171     42  20        real address.
   1      L5  84     42   0
   2      26  66     L5  17        Inside when real address is
   3      L4   0     L4   1        known.
   4      40   1     26 188
   5      L3   0     32 185        Inside when real address is
   6      F5 180     42 180        not known. Jump to 185 when
   7      50  30     L5 160        no constant to be added.
   8      42  17     L5   0        Otherwise add new word to
   9      00  10     L4  17        additions list.
```

```
LOCATION        ORDER                               NOTES

180      50  30     40 607
  1      L5 180     L0 199
  2      32 447     26 184
  3      L5 261     42 180
  4      F5 190     42 190
  5      26 186     L5 160        For inside, subst. list address
  6      42   1     L5  29        and add bit to those to be added
  7      L4  23     40  23        to 2 bit register.
  8      49  21     L1  31        Bookkeeping after inside add.
  9      40  31     32  69

190      26  66     00  42
  1      +6  66     00 357
  2      L5 999     F0  30        New symbolic address. Decrease
  3      42 999     42 160        list address by one and add new
  4      42 195     89   1        word to the list.
  5      L4 946     40   0
  6      L1 999     L4 200        Test if list capacity is exceeded.
  7      32 204     22 168
  8      L1 946     L4 945
  9      50  30     40 620

200      491002     -5 620
  1      L5 151     42 999
  2      26  66     00  93
  3      L3  23     32 204        ------------K Termination------------
                                  Test for error
  4      26 480     L5   1
  5      10  16     40   1        Determine function digit and set switch
  6      01   4     L4 221
  7      42 208     50 207
  8      26 357     26   0
  9      L5   0     22 214        00 K
```

```
LOCATION        ORDER                        NOTES

210     L51001      22 212          01 K
  1     L5    1     10    4         02 K
  2     421001      L4    0
  3     42   84     42    0
  4     22 217      401001
  5     L3    0     32 216
  6     23 212      L5   84
  7     26 212      50    0         Compute address and position of bit
  8     09    1     40   25         in counter word.
  9     67   35     40   18

220     F0   30     42 224
  1     11    1     -5 209
  2     L4 254      42   26
  3     42 225      L3   18
  4     32 225      19    0
  5     40   25     L5    0
  6     40   24     L5 367          Set drum order
  7     L4    0     40 362
  8     49   31     49   21
  9     41   22     41   23

230     L3   20     36   93         Test whether outside address calling
  1     L5   20     42 232          for library routine.  Set correct real
  2     L51001      42    0         address.  Search reference word list on
  3     L5 242      40 235          drum.  End search on -1.
  4     411023      41   20
  5     85   11
  6     36 238      L0   32
  7     36 479      L4   32
  8     L0 945      32 239
  9     221 240     L4   41
```

```
LOCATION       ORDER                        NOTES

  240      36 262    F5 235
    1      40 235    26 235
    2      85  11    001850
    3      81   5    42 247     ------------S Termination-----------
    4      10   1    42 248     Read extra character and set addresses.
    5      42 260    L4 367
    6      40 259    49  16
    7      50  16    11   0
    8      J0 947    L5   0     Determine is inside address used.
    9      42 251    40  16

  250      22 250    -3 243     Is correction to be made?
    1      36 260    L5   0
    2      42  17    42  16
    3      L3  17    36 341     Can correction be made?
    4      L5 947    -0 947     Change 2 bit register
    5      40 947    L5  26
    6      42 257    50 256     Record words on drum.
    7      26 357    L5   0     Replace 2 bit register and correct
    8      40  24    L5  16     copies of S parameter.
    9      86  11

  260      L5  16    40   0
    1      L4   0    22  93     ------------Modified Y2------------
    2      L0  41    L4 242
    3      401021    491019     Store drum order and modify counter
    4      L51022    40  19     Save info. on spec. interludes and errors.
    5      411022    L5 296     Modify important jump addresses.
    6      42  90    L5 325
    7      40  93    L5  92     Set end constant.
    8      42 379    26 269
    9      41   0    50   0
```

```
LOCATION        ORDER                         NOTES

 270      F5 270     42 285          Set add.-fn. dig. switch.
   1      L51019     00   1          Advance count. End of 5 blocks?
   2      401019     32 283
   3      F5 279     40 279          End of 36 words from drum?
   4      L0 327     32 279
   5      L5 294     42 279          Record words on drum.
   6      42 375     50 276
   7      26 357     L51021          Bring next group of 36
   8      40 374     50 278          words from the drum.
   9      26 373     L5 608          Pick off one word. Adjust

 280      401020     L61023          sum check, drum order, counter.
   1      401023     F51021
   2      401021     19   4
   3      401019     L51020
   4      50  32     10   8          Split  off 8 bits.
   5      401020     22   0          Add.-fn. dig. switch.
   6      LN   0     01   2          Set termination switch.
   7      F4 287     42 289
   8      L5   0     00   6          Add 6 bits to address at 0.
   9      40   0     26   0

 290      26 271     00 607          ≥ 64
   1      L41001     40   0          L
   2      40  16     22 300          F
   3      L5   0     10   6          K, S, N.
   4      40   0     -3 572          Test for interlude
   5      32 309     L4 286          Test for termination of drum input.
   6      32 312     -5 269          Test for K, S.
   7      L4 286     32 299
   8      L5  16     40   0          K termination.
   9      26  58     01   7          S termination.
```

```
LOCATION        ORDER                        NOTES

  300     22 244     F5 300         Set switch for fn. digits.
    1     22 270     L5   1         Read fn. digit block.
    2     00  20     L4   0         Add to word and bookkeeping.
    3     40   1     L5  25
    4     L4  22     40  22
    5     L5  25     10   1
    6     36 307     L4  25
    7     40  25     L1  31
    8     40  31     32  69
    9     26 269     L5   1         N termination.

  310     L0  16     40   1
    1     L5  16     40   0
    2     26  60     49  21         Test sumcheck. Reset constants,
    3     L5 327     42 279         addresses. Terminate drum input.
    4     L5  19     401022
    5     L51021     40 317
    6     F5 317     401021
    7     85  11
    8     F41023     401023
    9     L31023     32 320

  320     26 101     L5 324
    1     42  90     L5 326
    2     40  93     411019
    3     411020     411021
    4     49  31     26  66
    5     26 269     22 291         Constants
    6     26  66     22  63
    7     +6 373     L5 608
    8     41  27     L5  84         ---------Print Outside Type Error---------
    9     L0  17     42  27
```

```
LOCATION        ORDER                                NOTES
 330       L3   27     36   66
   1       92  131     92    2
   2       92  963     L5   20
   3       42  382     L5   17
   4       42   27     50  334
   5       26  407     L5   84
   6       42   27     50  336
   7       26  407     F5  337
   8       26  382     89    1
   9       L41022      36  171

 340       401022      26  171
   1       92  131     92  259        ---------Print S type error---------
   2       92  706     92  707
   3       92  963     41   27
   4       L5  253     42  382
   5       50   32     L5  248
   6       42   27     50  346
   7       26  407     L5   84
   8       42   27     50  348
   9       26  407     F5  349

 350       26  382     L31019
   1       36  354     89    1
   2       L4   19     36  356
   3       40   19     26  356
   4       89    1     L41022
   5       36  356     401022
   6       L5    0     22   93
   7       +5    0     42  366        -------------------
   8       L5  368     42  361        Store words on the drum.
   9       L5   86     42  369        Closed subroutine. Set
```

```
LOCATION        ORDER                          NOTES

  360     L5 361     L0 369        initially to store 2 - 15
    1     32 365     L5   2        on the drum.
    2     86  11     023566
    3     F5 362     40 362
    4     F5 361     40 361
    5     22 360     L5 368
    6     42  86     22 563        Link.
    7     86  11     023564
    8     00   0     00 563
    9     32 365     L5  16

  370     86  11     023563
                                   --------------------
    1     36 374     22   0        Link.    Bring words
    2     00   0     42 375        from the drum. Closed
    3     +5 507     42 371        subroutine. Set initially
    4     85  11     001617        to rewrite 1002 - 1018.
    5     32 375     401002
    6     F5 374     40 374
    7     F5 375     40 375
    8     L0 379     26 371
    9     -2 375     401019

  380     85  11     023564
    1     85  11     023563
                                   --------------------
    2     42 388     L5   0        Entry.   Print symbolic address.
    3     10  10     40  27        Closed subroutine.  Address of word
    4     41  28     19   3        to be printed is substituted in
    5     40  21     26 390        382.
    6     L5  21     36 389
    7     L5  28     32 388
    8     92 707     22   0        Link.
    9     04   1     40  21
```

```
LOCATION        ORDER                          NOTES
 390      50  28     L5  27
   1      10   6     40  27
   2      -5   0     00   2
   3      40 946     10   2
   4      43 946     -3   0
   5      36 386     -J   0
   6      36 400     89   1
   7      L4  28     40  28
   8      32 399     92 259
   9      26 400     92 707

 400      51 946     01   4
   1      -4   0     36 403
   2      50 405     22 403
   3      50 389     00   3
   4      00   3     42 405
   5      02   1     92   0
   6      26 386     00 549          ------------------------
   7      +5   0     42 416          Entry.  Convert binary
   8      50  32     L5  27          address to decimal and print.
   9      00  20     50 417          Closed subroutine. Address

 410      -0   0     36 410          to be printed is stored at 27.
   1      -4   0     50 418
   2      -0   0     36 412
   3      -4   0     46  27
   4      00  12     L4  27
   5      00   8     82  12
   6      92 963     22   0          Link.
   7      00  99     L0   0
   8      00   9     LL   0          ---------------------
   9      41  27     L5 532          Print inside type error.
```

```
LOCATION        ORDER                         NOTES

  420     42 434    L5 531
    1     L0 551    36 423
    2     L5 533    26 424
    3     L5 535    10  20
    4     42 382    22 427
    5     41  27    L5 406
    6     42 434    L5 544
    7     42 382    92 131
    8     L5 381    L0 367
    9     L0 368    L4  16

  430     42  27    50 430
    1     26 407    F5 431
    2     26 382    89   1
    3     L41022    32 434
    4     401022    26   0
    5     L5   1    L4   0      ---------N Termination---------
    6     40   1    L5   0      Construct jump instruction.
    7     L0  38    32 438
    8     26 445    L4 202      Jumps with address ≥ 999.
    9     42   1    41  22

  440     41  23    L1  31
    1     36 442    23 442
    2     L5  29    L4  29
    3     40  25    49  31
    4     41  20    22   1
    5     41   0    50 445      Record words on drum.
    6     26 357    F5 456      Set link and record remainder
    7     42 457    L5 180      of additions list on drum.
    8     42 452    L5 370
    9     L4 458    F0 452
```

```
LOCATION       ORDER                          NOTES

  450     40 453     40 370        Note that the order is reversed
    1     L5 452     50  30        during the transfer.
    2     22 456     L5   0
    3     86  11
    4     F5 453     40 453
    5     L5 452     F0  30
    6     42 452     L0 458
    7     32 452     26 183
    8     22 456     L5 608
    9     L51022     42  17        Test for special interlude.

  460     L3  17     32 476
    1     L5  17     421008        Set addresses in control block
    2     421003     L4 367        (999 - 1018) for special interlude.
    3     401009     L0 478
    4     401002     L5  84
    5     421017     421018
    6     L31019     36 468        Test for tape-drum input.
    7     L5  19     401022
    8     L3  23     36 470        Test for error in jump at
    9     L5 480     40   1        location 1.

  470     L5 190     50  32        Compact address at 190 with
    1     10  10     -5   0        relativizer and store at 1001.
    2     L41001     401001
    3     L5 523     42 452        Store 2 bit registers and symbolic
    4     L5 999     42 458        address list on drum.
    5     L51005     42 457
    6     22 448     L31019        Test for drum or regular interlude.
    7     36 481     26 467
    8     01   0
                                   ---------Back Substitution and---------
    9     FF  48     FF  49        Regular interludes for tape input
```

```
LOCATION        ORDER                          NOTES

  480      FF  50      FF  51
    1      L5 190      42 488       Transfer additions list from drum to
    2      L0 191      32 483       Williams  memory. This loop is set
    3      FF  52      L5 381       initially to bring symbolic address
    4      F0 488      L4 497       list, etc., from the drum during
    5      40 487      L5 488       input starts.
    6      50  30      22 491
    7      85  11      023563
    8      32 488      40   0
    9      F5 487      40 487

  490      L5 488      F0  30
    1      42 488      L0 497
    2      36 487      26 588       See word 593. Changed during input starts.
    3      42 458      L5 513       Store list of symbolic addresses
    4      42 452      F5 491       on drum.
    5      42 457      L5 478
    6      L4 381      26 449
    7      32 488      40  43
    8      L5 367      40 381
    9      L5 380      40 374

  500      L5 368      L4  35       Set links, etc., for transferring
    1      42 369      42 379       blocks of 20 words to and
    2      L5 373      46 361       from the drum.
    3      46 369      L5 556
    4      42 371      22 509
    5      L5 381      L4  35       Advance addresses.
    6      40 381      L0 478
    7      40 374      F5 509       Entry after transfer to drum.
    8      40 509      L0 557
    9      36 558      F1 947       Test 2 bit register for 0.
```

```
LOCATION        ORDER                          NOTES

 510        40  18      F3  18
   1        36 505      L5 368         In case 2 bit reg. ≠ 0,
   2        42 361      42 375         bring 20 words from the
   3        26 374      00 944         drum.
   4        40  16      L5  18         Adjustment after subst.
   5        80   2      F4  32         real addresses.
   6        40  18      F3  18         Test for completion of subst.
   7        36 553      L5  18         Test for inside addresses requiring
   8        36 524      80   1         substitution of real addresses. Both
   9        36 524      80   1         instructions of a pair tested each
                                       time through loop.


 520        F4  32      40  18
   1        F5  16      40  16
   2        22 517      L5 368         Entry after bringing 20 words.
   3        22 521      00 998
   4        L5  16      42 525         Set up for address substitution.
   5        42 549      L5   0
   6        46 531      42 542
   7        46 535      42 544
   8        40  31      F1  18         Substitution for left instruction.
   9        36 539      L5 531         Test is constant to be added?

 530        L0 551      32 534
   1        L5   0      42 533         Set up if constant to be added.
   2        40   0      50 539
   3        43   0      L5   0
   4        22 535      41   0
   5        L5   0      42  17         Substitution of address in left
   6        00  10      L4   0         instruction.
   7        00  10      46  31
   8        L3  17      36 419         Test for error.
   9        F1  18      80   1         Substitution for right instruction.
```

```
LOCATION        ORDER                          NOTES

  540     36 549     41   0        Test is constant to be added to
    1     L5 542     L0 552        right instruction?
    2     32 544     L5   0        Set up if constant is to be added.
    3     42 544     10  10
    4     40   0     L5   0
    5     42  17     43  31        Substitute address.
    6     L5  17     L4  31
    7     L4   0     40  31
    8     L3  17     36 425        Test for error.
    9     L5  31     40   0

  550     F5  16     26 514
    1     L5 512     42 533        Constants
    2     32 544     L5 512
    3     L5 381     40 362        Record modified words on drum.
    4     L4  35     40 381
    5     L5  16     42 369
    6     22 361     00 522
    7     36 558     F1 999        Test for error in jump instruction
    8     L51022     36 560        at location 1. Begin interlude.
    9     L5 479     40   1

  560     L3  23     361002
    1     L5 480     40   1
    2     261002
    3     23 107     41   0        ---------Input Starts---------
    4     41  30     41  17        Entry at right of 563 from 366.
    5     41  20     41   1
    6     L51002     42   0        Save information on type of start.
    7     40  16     50 567        Rewrite 1002 - 1018.
    8     26 373     L3  16        Hold or clear start when N(16)=0
    9     32 570     26 575
```

```
LOCATION         ORDER                                 NOTES              PAGE 20   X 12
  570        42 574     41 947          Set 2 bit registers = 0
    1        F5 570     40 570
    2        L0 601     32 570
    3        411019     411021          Set locations for special interludes,
    4        411022     22 588          errors and drum input.
    5        501001     01  10          In case of interludes decompact 1001.
    6        40  17     11  10          Re-establish relativizer.
    7        -5   0     401001
    8        L3  17     32 597          N(17)= 0 means regular interlude.
    9        F5 579     42 594          For special and drum input interludes

  580        L5  17     42 190          replace constant at 190. Re-establish
    1        42 488     L5 487          drum orders for additions list and
    2        L4 497     F0 488          bring back symbolic address list and 2 bit
    3        40 487     L4 478          registers.
    4        40 370     L5 523          Transfer lists to Williams memory.
    5        42 488     L5 999
    6        42 497     L5 487
    7        26 484
    8        41   1     L5 600          Enter K termination loop to set up
    9        42 231     L5  16          constants for SADOI

  590        36 210     26 214
    1        L5 202     42 231
    2        L5 112     42 497          Reset some orders and start input.
    3        L5 602     40 492
    4        50  30     26 563
    5        L51021     32 596          Distinguish between special and
    6        26 264     431021          drum input interludes.
    7        26  93     F5 597          Regular interludes
    8        26 570     L5 202
    9        42 594     L5 513
```

```
LOCATION      ORDER                        NOTES
  600      26 585    00 591
    1      N2 574    41 999                Constants
    2      36 487    L5 999
    3      00   0    00   0                sum check constant.
    4      80   0    26 100                Overwrite bootstrap to start SADOI

 1614      49 1002                         999 type interludes
  999      S5  945
 1615      42 1002
 1000      22 1008
 1616      00    0                         Relativizer
 1001      00    0
 1617      85   11
 1002      02 3566                         Bring assembled
 1618      32 1003
 1003      40    2                         program to
 1619      L6    0
 1004      40    0                         Williams memory.
 1620      F5 1002
 1005      40 1002
 1621      F5 1003
 1006      40 1003
 1622      L0 1018
 1007      36 1002
 1623      22    1
 1008      L5   16                         Put contents of
 1624      86   11
 1009      02 3580                         Williams memory
 1625      F5 1009
 1010      40 1009                         on drum.
 1626      F5 1008
 1011      40 1008
```

LOCATION	ORDER	NOTES
1627	L0 1017	
1012	32 1008	Bootstrap start
1628	F5 1015	
1013	40 607	Waste
1629	26 1015	1014 interludes
1014	26 1000	
1630	85 11	for main part of
1015	00 1027	
1631	40 608	SADOI
1016	26 607	
1632	K2 1	constants.
1017	L5 999	
1633	S2 1003	
1018	40 999	2 - 31
1634	05 11	Restore
1019	03 512	
1635	85 11	after SADOI
1020	03 482	
1636	32 1021	is first
1021	40 2	
1637	F5 1021	called for. Used
1022	40 1021	
1638	F5 1020	as storage locations
1023	40 1020	
1639	L0 1019	by SADOI.
1024	36 1020	
1640	41 1002	
1025	22 1008	Sum check
1641	8S 1759	
	13 2340	