LOCKHEED MARTIN ENERGY RESEARCH LIBRARIES
3 4456 0514418 0

CENTRAL RESEARCH LIBRARY
DOCUMENT COLLECTION

OAK RIDGE NATIONAL LABORATORY
operated by
UNION CARBIDE CORPORATION
for the
U.S. ATOMIC ENERGY COMMISSION

ORNL-TM-3287

cy. 97

DATE - February 5, 1971

Instrumentation and Controls Division

HISS, HYBRID INTERFACE SUBSTITUTE SUBROUTINES FOR USE IN DEBUGGING HYBRID PROGRAMS

H. Patterson* O. W. Burke R. S. Booth

ABSTRACT

A set of FORTRAN IV subroutines identified as HISS (Hybrid Interface Substitute Subroutines) was prepared so that by using the extensive printout and monitor system of the IBM-360 computer one can debug a hybrid computer program or test its output against a sample calculation. HISS extensively tests the CALL statements unique to a hybrid program; the resulting error messages, warnings and facts, and statements for future reference help the programmer to convert his initial hybrid program to a production code. HISS also is a file of the hardware and software available on the hybrid computer and of experience and insights in the use of the hybrid.

---

*Summer Participant (1970) from Huntingdon College, Montgomery, Alabama.

OAK RIDGE NATIONAL LABORATORY
CENTRAL RESEARCH LIBRARY
DOCUMENT COLLECTION
LIBRARY LOAN COPY
DO NOT TRANSFER TO ANOTHER PERSON
If you wish someone else to see this document, send in name with document and the library will arrange a loan.
UCN-7969
(3 3-67)

NOTICE This document contains information of a preliminary nature and was prepared primarily for internal use at the Oak Ridge National Laboratory. It is subject to revision or correction and therefore does not represent a final report.

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or 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.

CONTENTS

		Page
1.	Introduction	4
2.	Operations Performed by HISS	5
3.	Messages From HISS	6
4.	Use of HISS	7
5.	Some Incorrect HCR CALL Statements and Resulting Error Messages from HISS	8

LOCKHEED MARTIN ENERGY RESEARCH LIBRARIES

3 4456 0514418 0

1. INTRODUCTION

Hybrid programs are more difficult to debug than digital programs, because hybrid programs incorporate digital programs for control of the hybrid's digital computer, analog and logic patching for control of the hybrid's analog computer, and HCR[1] subroutines for communications between the digital and analog computers. Furthermore, proper use of the HCR subroutines, which can be executed only on the hybrid computer, requires knowledge of the hybrid computer obtained primarily by experience.

Usually the analog and logic patching are debugged separately, which eliminates some complexity in the final debugging of the hybrid program. Likewise, digital computer facilities provide debugging programs, extensive printouts, and programming assistance for debugging digital programs; and, in general, the larger the computer facility, the more extensive the debugging capabilities.

To complement the existing debugging capabilities for digital programs, a collection of FORTRAN IV subroutines labeled HISS (Hybrid Interface Substitute Subroutines) was prepared to detect improper use of the HCR subroutines in hybrid programs prepared for the Controls Department hybrid computer.[2]

HISS can be used to debug all but the analog and logic patching of a hybrid program or to test its output against a hand calculation. The hybrid program and HISS are executed by a digital computer, usually the IBM-360. Any HCR CALL statement that would initiate action during execution on the hybrid will initiate, instead, a HISS subroutine of the same name. The error and warning messages from the computer and from HISS inform the programmer of coding mistakes. Thus, HISS functions as an extension of the IBM-360 compiler.

HISS was designed to be simple and easy to use. It was developed not only to aid in debugging programs, but also to promote more efficient use of the hybrid computer and to complement and extend the present library of hybrid programs. Specific reasons for the development of HISS include:

1. The ISS (Interface Simulation System) program available from Applied Dynamics for debugging hybrid programs was not purchased. Thus, HISS is a replacement for and extension of ISS.

---

[1] Hybrid Communication Routines Operators Manual, Applied Dynamics, Inc., 1969.

[2] R. S. Stone, The Controls Department Hybrid Computer Hardware and Applications, ORNL-TM-3101 (August 24, 1970).

2. The hybrid computer is presently limited to a single teletypewriter for printout, which precludes the extensive use of computer dumps and the expanded printing of parameters and arrays during the debugging of a program. Extensive printout, however, is inexpensive to obtain from the IBM-360 computer.
3. Use of HISS reduces the hybrid computer time required for debugging since programs checked by HISS should execute properly on the hybrid if the analog and logic patching are correct.
4. The Controls Department hybrid computer is a unique installation. Many subroutines mentioned in the HCR manual and more than one-fourth of the patch holes on the analog and logic patchboards are not operational because all of the hardware available for the AD-4 analog computer and interface system was not purchased. HISS provides a readily available and easily modified file of which hardware and software mentioned in the manuals is actually available on the hybrid.
5. The manuals and computer programs associated with the HCR's are not completely adequate for our purpose. The HCR error subroutine, HYTEST, checks only the arguments in the CALL statement for low-speed HCR subroutines and provides no checks for high-speed subroutines.
6. HISS is a convenient file of experience and insights in the use of the hybrid. As more experience is gained in the use of the hybrid, it will be incorporated in HISS. Particular attention was given to the detection by HISS of subtle errors caused by intricate logic or by idiosyncrasies of the hybrid that would affect the proper execution of a program and perhaps not be found without detailed testing of the program output.

2. OPERATIONS PERFORMED BY HISS

HISS does not simulate the analog computer. During actual execution of a program by the hybrid computer, any CALL statement that brings numbers to the digital computer from the analog will result in the input of values that depend on the analog computer's patching and operation. When the digital computer reaches the same CALL statements under HISS, the numbers returned by HISS are reference numbers that should prevent underflow or overflow errors during the subsequent calculations of the digital computer.

HISS does make a complete check of the arguments supplied in the CALL statement. Specifically, the following questions would be considered by the HISS subroutine initiated by the CALL statement:

1. Are the argument range and type correct? For example, a request for hardware not available on the hybrid or for hardware using its octal address when the binary address is required, or vice versa, or a request that a coefficient device be set beyond its range would all initiate error messages from HISS.

2. Is this a valid CALL statement at this point in the program execution? A command to read the setting of a coefficient device while the analog computer is in the operate mode is not valid and would initiate an error message from HISS. This invalid CALL statement would disconnect the coefficient device from the analog patchboard, which would result in an incorrect input to an amplifier during the solution of the problem.
3. Is the configuration of analog computer such that the desired operation would be properly executed? For instance, use of a high-speed HCR subroutine such as SCANH returns proper numerical values only if the MUX (ADC/multiplexer system)[1] is connected. Likewise, a command to update the value of certain digital coefficients is valid only if the update code in the update register for these coefficients has been properly set.
4. What HCR CALL statements are anticipated because of the CALL statement currently being tested? For example, the HCR subroutine SETBLK usually places the next value for a digital coefficient in a register but does not change the digital coefficient itself. The digital coefficient is changed to the value in the register by a proper CALL statement to the HCR subroutine UPDAT. Thus, UPDAT is usually used after SETBLK.

3. MESSAGES FROM HISS

The messages from HISS are of three general types; errors, warnings and facts, and statements for future reference. There is an option that, if desired, only the error and warning messages are printed.

Error messages are generated whenever a coding error is detected. The message often includes a suggestion of how to correct the error. A CALL statement to any high-speed HCR subroutine, for example, can contain only decimal addresses of analog components. If HISS detects a valid octal address rather than the valid decimal address, an error message is printed followed by a suggestion that the HCR subroutine ADDR be used to obtain the decimal address.

HISS statements that convey warnings and facts are derived primarily from experience with the hybrid and are extremely valuable to the novice programmer. These messages indicate coding which could cause an error in execution and they inform the programmer of features of the hybrid not generally known or recorded in the manuals. For example, each use of the HISS subroutine ITEST, initiates the warning message that two successive calls to ITEST are required to ensure that the current status of a specified sense line is returned. Contrary to what one would think, the first call to ITEST usually yields the previous--not current--state of the sense line. An example of a statement of fact is the message from HISS initiated by a call to the HCR subroutine INMUX with an argument of zero. The programmer is informed that the MUX has been connected, but any MUX readings taken during the next 3 msec are subject to error due to switch bounce.

Statements for future reference allow the programmer to trace the sequence of operations requested by his valid CALL statements. For example, a CALL to the HCR subroutine IC can result in the HISS message that the console is in the Initial Condition Mode. Likewise, a valid call to the HCR subroutine LEX with an argument of zero will result in the HISS message that the Logic Execute Mode is now off.

4. USE OF HISS

The minimum requirement for incorporating HISS into a hybrid program is to place a labeled common card and CALL SETUP(N) card in the main program. (N is 0 if only error warning messages are desired and 1 if all messages from HISS are desired.) However, the use of HISS does have pitfalls, and those detected are listed below:

1. HISS supplies only reference values when HCR subroutines such as READ and ITEST are called. If a particular value is required from these and other HISS subroutines for testing the program output or transfer of control, this value must be supplied as a FORTRAN statement directly after the CALL statement.
2. Any CALL statement associated with a DEC tape such as CALL IFILE or CALL OFILE is not included in HISS. These CALL statements should either be removed from the hybrid program or dummy subroutines should be supplied by the programmer.
3. The standard input device and output device numbers for the hybrid are 2 and 5, respectively. (Care should be taken in the preparation of the IBM-360 control cards to ensure that the read and write statements of the hybrid program will be executed.)
4. Several parameters associated with the configuration of the analog are defined by the CALL SETUP(N) card which must be included in the initial part of the main program. The analog configuration defined by these parameters is very untypical of normal operation, but is chosen so that HISS will detect an omission in the hybrid program of essential CALL statements to the HCR subroutines INITA, CONSO, and INMUX. The analog configuration defined by the CALL SETUP(N) statement is:
    a. The update codes for digital coefficients that can be updated are set equal to 2.
    b. The MUX is not connected.
    c. The LOGIC EXECUTE switch is off.
    d. The COEF-X IN switch is off.
    e. The analog is in the HOLD mode.
    f. The analog logic is in the RUN mode.

5. SOME INCORRECT HCR CALL STATEMENTS AND RESULTING ERROR MESSAGES FROM HISS

Table 1 is a collection of CALL statements in the form of a hybrid main program that will initiate the printing of several messages coded into HISS. The COMMENT cards included with the CALL statement indicate which error is made in the subsequent CALL. The output generated by this program is shown in Table 2. The number to the left of a message is the same as the number of the CALL statement in the main program that initiated the message. This program does not perform any logical calculation; it was written to illustrate the usefulness of HISS.

Table 1. Several CALL Statements That Initiate Messages from HISS

```
      PROGRAM ANALZ
      COMMON/HISS/ II,NRA,KRA,MUPD(5),ISMUX,MCDEA,MODEL,LOGEX,ICOEFX
     1,IPTE
      DIMENSION IA(2),IV(2),IAB(5),IVSTB(5),IVSCAN(1),IVS(1),IVS1(1)
      DATA IA/3040,3240/,IV/02000,05000/,IAB/1536,2221,1170,1153,1171/,
     1IVSTB/05000,20000,20000,10000,40000/
C  GET A PRINTOUT OF ALL HISS MESSAGES.
      CALL SETUP(0)
C  TURN ON THE LOGIC EXECUTE MODE.
    1 CALL LEX(IE,1)
C     CALL INITA WITH IMPROPER ARGUMENT.
    2 CALL INITA(IE,1)
C     CALL CONSC WITH IMPROPER ARGUMENT.
    3 CALL CONSC(IE,2)
C     CALL TSCAL WITH IMPROPER ARGUMENT.
    4 CALL TSCAL(IE,10)
    5 CALL CLRAT(IE,2)
C     CALL IC WHILE IN LOGIC EXECUTE MODE.
    6 CALL IC(IE)
C  TURN OFF THE LOGIC EXECUTE MODE.
    7 CALL LEX(IE,0)
C     CALL STIND WITH AN AMPLIFIER ADDRESS.
    8 CALL STIND(IE,1300,14000)
C  PUT ANALOG IN OPERATE MODE.
    9 CALL OP(IE)
C     CALL STIND WHEN IN OP MODE.
   10 CALL STIND(IE,3000,09000)
C     CALL STIND WITH A POT SETTING THAT IS OUT OF OBTAINABLE RANGE.
   11  CALL STIND(IE,3000,15000)
C CALL STIND TO SET A HAND SET POT TO A VALUE OUT OF THE OBTAINABLE RANGE.
   12 CALL STIND(IE,3032,20000)
C     CALL STIND TO SET A DCU TO A NEGATIVE VALUE.
   13 CALL STIND(IE,2220,-5000)
C  CALL READ TO DISCONNECT THE MUX.
      CALL READ(IE,0000,IV1)
C  CALL OP WHILE IN THE IC MODE AND WITH THE MUX NOT CONNECTED.
   14 CALL OP(IE)
C  15 CALL STINA WHILE IN THE OP MODE.
C     CALL STINA WITH ADDRESSES OF NON-EXISTING POTS ON OUR COMPUTER.
   16 CALL STINA(IE,IA,IV,1,1,2)
C     CALL STBLK WHILE IN OP MODE.
C     CALL STBLK TO DO THE FOLLOWING IN SUCCESSION
C     SET A POT.
C     SET A DAC, BUT USING THE OCTAL ADDRESS OF THE DAC.
C     SET A DAC, USING THE DECIMAL ADDRESS OF THE DAC.
C     SET A DAC THAT MUST BE UPDATED.
C     SET A DCU TO A VALUE THAT IS OUT OF THE OBTAINABLE RANGE.
   17 CALL STBLK(IAB,IVSTB,5)
C     USE WAIT SUBROUTINE TO GET A ONE SECOND TIME DELAY.
   18 CALL WAIT(10000)
C CONNECT MULTIPLEXER WITH NO SOFTWARE DELAY TO ALLOW SWITCH TO SETTLE.
   19  CALL INMUX(IE,0)
C     CALL SCANF WITHOUT PULSING TO ALLOW MUX SWITCH TO SETTLE.
   20 CALL SCANF(3216,IVSCAN,1)
C     DISCONNECT THE MULTIPLEXER.
      CALL READ(IE,1000,IV2)
```

Table 1. (continued)

```
C CALL SCANH WITH AN OCTAL ADDRESS AND WITH THE MULTIPLEXER NOT CONNECTED.
   21 CALL SCANH(622C,IVS,1)
C CONNECT THE MULTIPLEXER AND ALLOW APPROX. 20 MSEC FOR SWITCH TO SETTLE.
   22 CALL INMUX(IE,1)
C     CONVERT AN OCTAL ADDRESS TO A DECIMAL ADDRESS.
      CALL ADDR(6222,IAB1)
C     READ THE OUTPUT VALUE OF THE DEVICE WHOSE ADDRESS WAS CONVERTED.
      CALL SCANH(IAB1,IVS1,1)
C     GO TO THE LOGIC EXECUTE MODE WHILE IN THE RUN MODE.
   23 CALL LEX(IE,1)
C     CALL IC WHILE IN THE LOGIC EXECUTE MODE.
   24 CALL IC(IE)
C     TURN OFF THE LOGIC EXECUTE MODE.
   25 CALL LEX(IE,0)
C     GO TO THE LOAD MODE.
   26 CALL LOAD(IE)
C CALL ITEST WHILE IN THE LOAD MODE, WITH N=1, AND LINE NUMBER 16.
   27 CALL ITEST(IE,1,16)
      M=16
C     CALL ITSTM WHILE IN THE LOAD MODE.
   28 CALL ITSTM(IE,C,M)
C     CALL SENSW WHILE IN THE LOAD MODE, AND WITH ARGUMENT N EQUAL TO 1.
   29 CALL SENSW(1,MS)
      CALL EXIT
      END
```

Table 2. Messages from HISS Initiated by the CALL Statements of Table 1.

```
   YOU HAVE TURNED THE LOGIC EXECUTE MODE ON.

   ******************************   ERROR   ******************************
1  SINCE YOU WERE IN THE RUN MODE WHEN YOU CALLED LEX,REP OP STARTED WITH THE CALL.
   THE FIRST ANALOG MODE WILL BE THE MODE THAT HAPPENED TO BE ON THE ABC CLOCK AT THE TIME LEX WAS CALLED.

   ** WARNING ** YOU HAVE CALLED LEX WITH MUX NOT CONNECTED.  IF THE LAST DEVICE ADDRESSED WAS A COEFF. DEVICE, AN ERROR WILL RESULT.
   HAVE MUX CONNECTED PRIOR TO CALLING LEX TO INSURE AGAINST THE ABOVE SITUATION. CALL INMUX BEFORE CALLING LEX.

   ******************************   ERROR   ******************************
2  IN  INITA  ,  YOU HAVE CALLED FOR CONSOLE NO.    1.THE ONLY ONE AVAILABLE IS 0.

   ******************************   ERROR   ******************************
3  IN  CONSO  ,  YOU HAVE CALLED FOR CONSOLE NO.    2.THE ONLY ONE AVAILABLE IS 0.

   ******************************   ERROR   ******************************
4  YOU HAVE SELECTED AN INVALID TIME SCALE MULTIPLIER M=   10. CORRECT RANGE IS M=0,1,2,3.

   IN  CLRAT
5  YOU HAVE SELECTED LOGIC CLOCK OUTPUT BY EXTERNAL RATE (OPERATOR MUST SUPPLY).

   ******************************   ERROR   ******************************
6  YOU HAVE CALLED IC WHILE IN THE LOGIC EXEC. MODE. THIS RESULTS IN A NO-OP.

   YOU HAVE TURNED THE LOGIC EXECUTE MODE OFF.
7
   ** WARNING ** YOU HAVE CALLED LEX WITH MUX NOT CONNECTED.  IF THE LAST DEVICE ADDRESSED WAS A COEFF. DEVICE, AN ERROR WILL RESULT.
   HAVE MUX CONNECTED PRIOR TO CALLING LEX TO INSURE AGAINST THE ABOVE SITUATION. CALL INMUX BEFORE CALLING LEX.

   ******************************   ERROR   ******************************
8  IN STIND,  1300 IS NOT A VALID ADDRESS.

   CONSOLE IS IN OPERATE MODE.
9  ** WARNING ** YOU HAVE CALLED OP WITH MUX NOT CONNECTED. IF THE LAST DEVICE ADDRESSED WAS A COEFF. DEVICE, AN ERROR WILL RESULT.
   HAVE MUX CONNECTED PRIOR TO CALLING OP, TO INSURE AGAINST THE ABOVE SITUATION. CALL INMUX  BEFORE CALLING OP.

   ******************************   ERROR   ******************************
10 MUST NOT SET COEFF DEVICE IN OP MODE WITH STIND

   YOU HAVE REQUESTED A SETTING OF 15000 FOR IVPOT 3000. THE PROPER RANGE IS 0-10000.
11
   ******************************   ERROR   ******************************
   MUST NOT SET COEFF DEVICE IN OP MODE WITH STIND

   ******************************   ERROR   ******************************
    3032 IS A HAND SET POT AND CANNOT BE SET BY SUBROUTINE.

   ******************************   ERROR   ******************************
12 YOU HAVE REQUESTED A SETTING OF 20000 FOR POT 3032. THE PROPER RANGE IS 0-16384.

   ******************************   ERROR   ******************************
   MUST NOT SET COEFF DEVICE IN OP MODE WITH STIND

   ******************************   ERROR   ******************************
   YOU HAVE REQUESTED A SETTING OF -5000 FOR DCU 2220. THE PROPER RANGE IS 0-32768.
13
   ******************************   ERROR   ******************************
   MUST NOT SET COEFF DEVICE IN OP MODE WITH STIND

   CONSOLE IS IN OPERATE MODE.
14 ** WARNING ** YOU HAVE CALLED OP WITH MUX NOT CONNECTED. IF THE LAST DEVICE ADDRESSED WAS A COEFF. DEVICE, AN ERROR WILL RESULT.
   HAVE MUX CONNECTED PRIOR TO CALLING OP, TO INSURE AGAINST THE ABOVE SITUATION. CALL INMUX  BEFORE CALLING OP.

   ******************************   ERROR   ******************************
15 MUST NOT SET COEFF DEVICE IN OP MODE WITH STINA

   ******************************   ERROR   ******************************
   IN STINA,  3040 IS NOT A VALID ADDRESS.
16
   ******************************   ERROR   ******************************
   IN STINA,  3240 IS NOT A VALID ADDRESS.
```

Table 2. (continued)

```
   IN STBLK

   ******************************   ERROR   ******************************
   YOU ARE USING STBLK OR STINH IN OP MODE.

   ******************************   ERROR   ******************************
   IN STBLK,  1536 AND  3000 ARE THE DECIMAL AND OCTAL ADDRESSES OF A POT. STBLK SETS ONLY DACS.

17 ******************************   ERROR   ******************************
   IN STBLK,  2221 IS AN INVALID DECIMAL ADDRESS. HAVE YOU CALLED SUBROUTINE ADDR FOR ADDRESS CONVERSION.

   DAC ADDRESSED 2201 HAS UPDATE CODE 4

   A CALL TO UPDATE IS NEEDED TO SET DAC 2201

   ******************************   ERROR   ******************************
   YOU HAVE REQUESTED A SETTING OF 40000 FOR DCU 2223. THE PROPER RANGE IS 0-32768.

18 IN WAIT ,  YOU HAVE SPECIFIED A SOFTWARE DELAY OF    2231999 MICROSECONDS.

19 YOU HAVE CONNECTED TO THE MUX. ANY MUX READINGS TAKEN  WITHIN 3 MILLISEC ARE SUBJECT TO ERROR DUE TO SWITCH BOUNCE.

   ******************************   ERROR   ******************************
20 IF ARGUMENT N IN YOUR INMUX CALL WAS ZERO, DID AT LEAST 3 MSEC ELAPSE BETWEEN CALLING INMUX AND SCANH.

   ******************************   ERROR   ******************************
   IN SCANH OR READH,  6220 IS AN INVALID ADC/MUX ADDRESS. HAVE YOU CALLED SUBROUTINE ADDR FOR ADDRESS.

21 ******************************   ERROR   ******************************
   CALL INMUX AT LEAST 3 MSEC PRIOR TO CALLING SCANH OR CALL INMUX WITH ARGUMENT,N, EQUAL 1.

   YOU HAVE CONNECTED TO THE MUX. THE DIG COMP WILL DELAY FOR APPROX. 20 MILLISEC. BEFORE EXECUTING THE NEXT STATEMENT.

22 ******************************   ERROR   ******************************
   IF ARGUMENT N IN YOUR INMUX CALL WAS ZERO, DID AT LEAST 3 MSEC ELAPSE BETWEEN CALLING INMUX AND SCANH.

   YOU HAVE TURNED THE LOGIC EXECUTE MODE ON.

23 ******************************   ERROR   ******************************
   SINCE YOU WERE IN THE RUN MODE WHEN YOU CALLED LEX,REP OP STARTED WITH THE CALL.
   THE FIRST ANALOG MODE WILL BE THE MODE THAT HAPPENED TO BE ON THE ABC CLOCK AT THE TIME LEX WAS CALLED.

   ******************************   ERROR   ******************************
24 YOU HAVE CALLED IC WHILE IN THE LOGIC EXEC. MODE. THIS RESULTS IN A NO-OP.

25 YOU HAVE TURNED THE LOGIC EXECUTE MODE OFF.

26 LOGIC IS IN LOAD MODE.

   ******************************   ERROR   ******************************
   YOU HAVE CALLED ITEST WHILE NOT IN RUN MODE. YOU WILL  ALWAYS READ A ZERO WHEN NOT IN THE RUN MODE.

   ******************************   ERROR   ******************************
   ARGUMENT N MUST BE ZERO IN ITEST.

27 ******************************   ERROR   ******************************
   IN ITEST,    16 IS NOT A VALID LINE NUMBER. CORRECT RANGE IS 0-15.

   ** WARNING ** YOU MUST CALL ITEST TWICE IN SUCCESSION  IN ORDER TO GET THE CURRENT STATUS OF THE SPECIFIED SENSE LINE.

   ITEST=1

   ******************************   ERROR   ******************************
   YOU HAVE CALLED ITSTM WHEN NOT IN THE RUN MODE. YOU WILL READ ALL ZEROS FROM THE SENSE LINE REGISTER.

28 YOU MUST CALL ITSTM TWICE IN SUCCESSION IN ORDER TO COMPARE TO THE CURRENT STATUS OF THE SENSE LINE REGISTER.

   ITSTM=0

   ******************************   ERROR   ******************************
   IN SENSW, ARGUMENT,N, MUST=0 SINCE WE HAVE ONLY 1 SENSE LINE REGISTER.

   ******************************   ERROR   ******************************
29 YOU HAVE CALLED SENSW WHEN NOT IN THE RUN MODE. YOU WILL ALWAYS READ A ZERO WHEN NOT IN RUN.

   YOU MUST CALL SENSW TWICE IN SUCCESSION IN ORDER TO GET THE CURRENT STATUS OF THE SENSE LINE REGISTER.

   IN SENSW ,  SENSE LINE REGISTER READS 1C101C1C10101010 (2ND ARGUMENT)= 4349.
```

INTERNAL DISTRIBUTION

1. N. J. Ackermann, Jr.
2. R. K. Adams
3. H. I. Adler
4. R. G. Alsmiller, Jr.
5. D. W. Altom
6. J. L. Anderson
7. N. G. Anderson
8. S. I. Auerbach
9. S. J. Ball
10. C. J. Barton
11. A.E.G. Bates
12. S. E. Beall
13. M. Bender
14. J. L. Blankenship
15-39. R. S. Booth
40. C. J. Borkowski
41. A. A. Brooks
42. A. R. Buhl
43. C. E. Burgart
44. O. W. Burke
45. V. R. Cain
46. D. W. Cardwell
47. R. S. Carlsmith
48. H. P. Carter
49. N. E. Clapp
50. F. H. Clark
51. C. E. Clifford
52. C. C. Coutant
53. K. E. Cowser
54. J. A. Cox
55. F. L. Culler
56. R. A. Dandl
57. G. de Saussure
58. S. J. Ditto
59. J. W. Elwood
60. M. H. Fontana
61. J. L. Fowler
62. R. M. Freestone, Jr.
63. D. N. Fry
64. J. H. Frye
65. J. H. Gibbons
66. W. R. Grimes
67. D. G. Jacobs
68. W. H. Jordan
69. S. V. Kaye
70. M. T. Kelley
71. M. J. Kelly
72. R. C. Kryter
73. T. A. Lincoln
74. J. L. Liverman
75. R. S. Livingston
76. F. C. Maienschein
77. K. Z. Morgan
78. D. J. Nelson
79. P. Nelson
80. L. C. Oakes
81. R. W. Peelle
82. R. B. Perez
83. A. M. Perry
84. T. W. Pickel
85. H. Postma
86. P. S. Rohwer
87. D. P. Roux
88. G. S. Sadowski
89. H. E. Seagren
90. W. H. Sides
91. R. S. Stone
92. E. A. Straker
93. E. G. Struxness
94. E. H. Taylor
95. D. B. Trauger
96. A. M. Weinberg
97- 98. Central Research Library
99. Document Reference Section
100-104. Laboratory Records Department
105. Laboratory Records, ORNL R.C.
106. ORNL Patent Office
107-121. Division of Technical Information Extension
122. Laboratory and University Division, ORO

EXTERNAL DISTRIBUTION

123. D. F. Cope, U. S. Atomic Energy Commission, Oak Ridge, Tennessee 37830
124. R. H. Davidson, Tennessee Valley Authority, 303 Power Building, Chattanooga, Tennessee 37401
125. E. P. Epler, 712 Florida Ave., Oak Ridge, Tennessee 37830
126. H. Eckenrode, Applied Dynamics, Inc., 4306 Governor's Drive, Huntsville, Alabama 35807
127. Ronald Feit, U. S. Atomic Energy Commission, Washington, D. C. 20545
128. V. Gates, Applied Dynamics, Inc., 2275 Platt Road, Ann Arbor, Mich.
129. F. A. Gifford, Jr., U. S. Atomic Energy Commission, Oak Ridge, Tennessee 37830
130. S. H. Hanauer, U. S. Atomic Energy Commission, Washington, D. C. 20545
131. P. L. Havenstein, U. S. Atomic Energy Commission, Washington, D. C. 20545
132. T. W. Kerlin, Nuclear Eng. Dept., University of Tennessee, Knoxville, Tennessee 37916
133. K. O. Laughon, Jr., U. S. Atomic Energy Commission, Oak Ridge, Tennessee 37830
134. F. C. Legler, U. S. Atomic Energy Commission, Washington, D. C. 20545
135-159. H. Patterson, Huntingdon College, Montgomery, Alabama
160. J. R. Patterson, U. S. Atomic Energy Commission, Washington, D. C. 20545
161. J. Pidkowicz, U. S. Atomic Energy Commission, Oak Ridge, Tennessee 37830
162. J. C. Robinson, Nuclear Eng. Dept., University of Tennessee, Knoxville, Tennessee 37916