27
7-26-78
25 to 517162

17725
UCID-17226

Lawrence Livermore Laboratory

MATERIAL CONTROL SYSTEM SIMULATOR PROGRAM REFERENCE MANUAL

Roy B. Hollstien

January 24, 1978

Notice

This computer code material was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Department of Energy, nor any of their employees, make 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.

MASTER

This is an informal report intended primarily for internal or limited external distribution. The opinions and conclusions stated are those of the author and may or may not be those of the laboratory.

Prepared for U.S. Energy Research & Development Administration under contract No. W-7405-Eng-48.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

**MATERIAL CONTROL SYSTEM SIMULATOR
PROGRAM REFERENCE MANUAL**

R. B. Hollister

April 25 1978

**LAWRENCE LIVERMORE LABORATORY
University of California
Livermore, California 94550**

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Department of Energy, 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.

PREFACE

The Material Control System Simulator program was developed for the United States Nuclear Regulatory Commission, Office of Nuclear Regulatory Research under research order no. 60-77-012 and the United States Department of Energy under contract no. W-7405-ENG-48.

CONTENTS

1	INTRODUCTION	1
	Purpose of MCSS	1
	Program Organization	1
	Programming Language	2
	Memory Management	2
	Input/Output	3
	Diagnostics	3
2	DATA STRUCTURE	5
	Memory Management	5
	Free Store	5
	Tables	6
	Lists	6
	Stacks	7
	Block Chains	6
	Common	10
	Events	12
	Event Notices	12
	Scheduled Events	13
	Free Events	14
	Functions	15
	System Data	15
	Parameters	16
	Variables	16
	Initial Conditions	17
	Communications	18
	Address	18
	Type	18
	Date	18
	Material	33
	Type	33
	Quantity	33
	Containment	33
	Example	40
3	PROGRAM DESCRIPTION	46
	Macros	46
	Declaration	46
	Structured Programming	46
	Input/Output	47
	Recurrent Code	48
	Control Programs	49
	ASTORE	49
	CEC	49

CLOCK	49
CMB	50
COPYB	50
CREATE	50
CRMD1	51
CRMD2	51
DESTRO	52
DHPSYS	52
DUMP	53
ERROR	53
EVDRS	53
FPNTRS	53
FRMD	53
FSTORE	53
IAC	54
ICH	54
ICHECK	54
IIC	56
ILIC	56
ILP	56
ILV	57
INCREC	57
IP	57
IUNIF	57
IV	57
KEYWD	57
LCS	58
LIINH	58
MAIN	58
MZUR	59
NCBM	60
NSTATS	60
PCOM	60
PHLB	60
PICLB	60
PIDF	60
POP	61
PPLB	61
PUSH	61
PVLB	61
RAC	61
RIC	62
RLIC	62
RLP	62
RLV	62
RNORM	62
RNORMT	62
RP	63
RUNIF	63
RV	63
SAMPL	63
SCHED	63

SEND 64
 SET 64
 TIMER 65
 TRNFRM 65
 UMD 66
 URN 66
 WRMD 66
 ZIA 67
 ZIAC 67
 ZRA 67
 ZRAC 67
 ZTABLE 67

Function Programs 67
 B101 RCMB Receive Material Batches 67
 B102 IDAS Initiate Diversion Action Sequence 68
 B101 TRMB Transport Material Batches 68
 B401 STOM Store Material 69
 B501 INDA Independent Action 69
 B502 MAGA Material Acquisition Action 70
 B503 DEPA Dependent Action 70
 B504 MODA Modification Action 71
 B505 HDAS Halt Diversion Action Sequence 71
 B601 TIMR Timer 72
 B602 SCAS Start Control Action Sequence 72
 B701 MZMB Measure Material Batches 72
 B702 CMBA Compute Material Balance 73
 B703 MNTR Monitor Stimulus Type I 74
 B801 DTBL Decision Table 74
 B901 TRCN Trace Control 75
 B902 RMDS Report Material Diversion Statistics 75

4 ADDITION OF NEW FUNCTIONS 77

Type Number 77
 Communication Control 77
 Subroutine Organization 77

REFERENCES 79

APPENDIX

A SOURCE PROGRAM 80

Macros 80
 Control Programs 87
 Function Programs 144

B	FORTRAN PROGRAM	200
	Control Programs	200
	Function Programs	308

1 INTRODUCTION

PURPOSE OF MCSS

The objective of the Material Control Project at Lawrence Livermore Laboratory is the development of a methodology for use by the Nuclear Regulatory Commission in the assessment of material control and accounting systems. This report describes a Material Control System Simulator (MCSS) program for determination of material accounting uncertainty and system response to particular adversary action sequences that constitute plausible material diversion attempts. The program is intended for use in situations where randomness, uncertainty, or interaction of adversary actions and material control system components make it difficult to assess safeguards effectiveness against particular material diversion attempts. Although MCSS may be used independently in the design or analysis of material handling and processing systems, it has been tailored toward the determination of material accountability and the response of material control systems to adversary action sequences.

PROGRAM ORGANIZATION

The MCSS program has two principal parts, control programs and function programs. The control programs provide a general computer simulation framework within which the specialized function programs interact to simulate material control (MC) systems (1). All of the control programs and function programs are based on the top-down philosophy of systematic

Memory Management

programming (2,3), and their coding is standardized by use of structured programming technique (4,5).

PROGRAMMING LANGUAGE

MCSS is written in a specialized source language that provides macro statements for the control mechanisms of structured programming, input/output, and recurrent segments of code. The MCSS FORTRAN program is generated from the source program by preprocessor PRECOMP available on the CDC 7600 computers at Lawrence Livermore Laboratory (LLL). The source and FORTRAN programs are listed in the Appendices.

MEMORY MANAGEMENT

Space in a one-dimensional array is allocated to control and function programs as it is needed. When space is no longer needed, it is returned to the free storage pool. Integer values at location L of the dynamic storage array are referenced by IA(L). Real values are referenced by RA(L).

The maximum allowable size of the dynamic storage space is limited by the memory available in the computing system on which MCSS is to be run. Array IA is dimensioned and equivalenced to array RA in control program MAIN.

Space is allocated from a chain of free storage blocks by control program ASTORE and is returned to free store by control program FSTORE. Elements in dynamically allocated storage space are located by control function program LCS. These control programs are used by other control programs and function programs to create, manage, and destroy table, list, stack, and chain data structures described in Section 2.

Diagnostics

INPUT/OUTPUT

The MCSS program reads input data from file MCSSIN on device number NI1 and writes its output into file MCSSOUT on device number NO1. User interaction with the program is read from device NI2 and is written on device NO2. The file names and device numbers are assigned in control program MAIN. For the CDC 7600 computers at LLL, the symbolic device numbers are assigned the following values: NI1 = 5, NI2 = 59, NO1 = 5, NO2 = 59.

Input data are read in type- and field-free format. Output reports are generated by the function programs as described in the MCSS user's manual.

DIAGNOSTICS

Error messages are printed and the program is halted when abnormal conditions occur that are caused by either system limitations or the user's violation of MCSS modeling rules. The program in control at the time of the error writes the current simulation time and an error message, then calls control program ERROR which calls program DMPSYS to print a dump of the complete model system data before it halts execution of the program. This is to assist the user in diagnosing the cause of the error.

Warning messages are also written when suspicious conditions arise that should be brought to the attention of the user. Warning messages include the current simulation time, program identification, and an explanatory note. Warning conditions do not produce a dump of the model system or halt execution of the program.

Tracing and interactive debugging capabilities are provided to aid the

Diagnostics

user in development of complex simulation models. See the MCSS user's manual for a description of the run time diagnostic features of the program.

2 DATA STRUCTURE

MEMORY MANAGEMENT

Free Store

Dynamic storage allocation is based on the free store data structure shown in Figure 2.1. Blocks of contiguous locations in one-dimensional array IA/RA are linked to form a free store chain. The first word of each block contains a pointer to the next block on the chain. The second word of each block contains the number of useable words in the block. Locations LFFB and LLFB in common storage contain respectively pointers to the first and last blocks on the free store chain.

FIGURE 2.1 Free store data structure.

Tables

Tables in allocated storage have the data structure shown in Figure 2.2. Tables occupy a single block if enough contiguous locations are available in free store when space for the table is allocated. Otherwise they may occupy two or more chained blocks. The location of a table is the location of the first word of the first block. The first word of each block contains a pointer to the next block in the chain. The second word of each block contains the number of useable locations in the block. The third word of the first block contains the capacity of the table, which is one less than the total number of useable locations. The location of the i th item in the table is given by control function $LCS(L, i+1)$.

FIGURE 2.2 Table data structure.

Lists

Lists in allocated storage space have the data structure shown in Figure 2.3. Lists occupy a single block if enough contiguous locations are available in free store when space for the list is allocated. Otherwise they may occupy two or more chained blocks. The location of a list is the location of the first word of the first block. The first word of each block

contains a pointer to the next block in the chain. The second word of each block contains the number of useable locations in the block. The third word of the first block contains the capacity of the list data structure, which is two less than the total number of useable locations in the blocks. The fourth word of the first block contains the current count of the number of items (words) in the list. The location of the i th item in the list is given by control function $LCS(L, i+2)$.

FIGURE 2.3 List data structure.

Stacks

Stacks in allocated storage space have the data structure shown in Figure 2.4. Stacks occupy a single block if enough contiguous locations are available in free store when space for the stack is allocated. Otherwise they may occupy two or more chained blocks. The location of a stack is the location of the first word of the first block. The first word of each block contains a pointer to the next block in the chain. The second word of each block contains the number of useable locations in the block. The third word of the first block contains the capacity of the stack, which is two less than the total number of useable locations in the blocks. The fourth word

of the first block contains the current count of the number of items in the stack. The location of the last item placed on the stack is given by control function $LCS(L, IA(L+3)+2)$.

FIGURE 2.4 Stack data structure.

Block Chains

Blocks of allocated memory are chained as shown in Figure 2.5 to form block lists, stacks, queues, trees, etc. One useable location in each block of a one-way chain points to the next block on the chain. Two useable locations in each block of a two-way chain are used as pointers, one to the preceding and one to the next block on the chain. Additional pointers are used in more complex configurations; for example, four are used in the interconnected blocks of the MCS material data structure. Locations LF and LL of the blocks on the ends of the chains are usually maintained to simplify the addition and deletion of blocks in one- and two-way chains. The blocks in these data structures may be either single blocks of contiguous locations or chains of blocks allocated from free storage. They may be created and destroyed by the same program or created by one program and destroyed by another.

Memory Management

FIGURE 2.5 Block chain data structures.

Memory Management

Common

The allocated storage array IA/RA is in blank common, and the following symbolic locations are in named common block SYSTEM:

ICTYPE	Type of communication.
!DUMP	Control parameter read from the RUN command line. If set to 1, a dump of the model system is printed just before the simulation is started.
!POINT	Pointer to communicated data.
!SECA	Increment size of the event chain storage allocation.
!TRACE	Control parameter that specifies the level of trace messages written during execution of the program.
LB	Location of the MCSS function that currently has program control.
LBCA	Location of the block checked for adjacency while scanning the free store chain.
LBLIST	Location of the list of blocks currently in the model system.
LBPT	Location of the block pointer table.
LBUF	Location of a block of 80 contiguous locations used as a buffer.
LCSB	Location of the block containing a location returned by control function LCS.
LEC	Location of the event chain.
LFFB	Location of the first block in free store.
LLFE	Location of the first free event notice.
LLE	Location of the last scheduled event notice.
LLFB	Location of the last block in free store.
LLFE	Location of the last free event notice.
LMS	Location of the message stack.
LNE	Location of the next scheduled event.

Memory Management

LP Location of the parameters of the MCS3 function that currently has program control.

LV Location of the variables of the MCS3 function that currently has program control.

LI Location of the initial conditions of the MCS3 function that currently has program control.

LE Location of the last word of the MCS3 function that currently has program control.

MINNFL Minimum number of free locations in free store.

MFROM Sending address of a communication.

MNSE Maximum number of scheduled events.

MTD Receiving address of a communication.

NFE Number of free event notices.

NFL Number of free locations in free store.

NI1 Input device number.

NI2 Interactive input device number.

NO1 Output device number.

NO2 Interactive output device number.

NREP Number of replicated runs in the current job.

NRUN Number of the current run.

NSE Number of scheduled events.

TIME Current simulation time.

TSTOP Stopping time for the current simulation run(s).

EVENTS

Event Notices

As shown in Figure 2.6, event notices are groups of five contiguous locations within the useable space allocated at LEC for scheduled and free event chains. The first word of an event notice points to the next group of five locations on either the scheduled event chain or the free event chain. The second word contains the time of scheduled events. The third word contains the MCSS function in which a scheduled event is to occur. The fourth word contains an identification of the event type. And the fifth word is a pointer to data stored for reference at the time a scheduled event occurs.

A block of 500 useable locations is originally allocated by control program MAIN for event notices. If more than 100 events are scheduled, additional space is allocated in increments of ISECA useable locations. ISECA is assigned by control program MAIN.

FIGURE 2.6 Event notice data structure.

Scheduled Events

Events are scheduled by linking event notices for future events in chronological order on a one-way chain as shown in Figure 2.7. LNE points to the notice of the next scheduled event, and LLE points to the notice of the last scheduled event.

FIGURE 2.7 Scheduled event data structure.

Free Events

Space allocated for event notices but not used in the scheduled event chain is linked in a free event chain as shown in Figure 2.8. LFFE points to the first free event notice, and LLFE points to the last free event notice.

FIGURE 2.8 Free event data structure.

FUNCTIONS

MCSS functions occupy single blocks of contiguous words in allocated storage space, as shown in Figure 2.9. If there are not enough contiguous words available in free store at the time an attempt to create a function is made, an error message is written and the program is halted.

FIGURE 2.9 MCSS function data structure.

System Data

The first five words of a function contain data used by MCSS control programs. The first word contains the function type number. Since the function must occupy a single block, there is no need for the block pointer normally in the first word of allocated storage space. The second word saves the number of useable locations allocated from free store so that all

Functions

of the space can be returned if the function is destroyed. Three useable locations plus the number of parameters plus twice the number of variables are required for the function data structure. The third word contains the number of the function assigned in the user's model system. The fourth and fifth words contain respectively the number of parameters and number of variables used by the function.

Parameters

Parameters are stored in locations LP through LV-1 of IA/RA. There are four types of parameters: 1) integer scalar, 2) integer list, 3) real scalar, and 4) real list. The first $IA(LB+3)$ elements of the local, one-dimensional array IPVT of each function contain an integer value from 1 to 4 that defines the type of the corresponding parameter. Scalar parameters are stored directly in array IA/RA. List parameters are stored indirectly in the list data structure at the location contained in the corresponding scalar integer parameter. The i th integer and real scalar parameters of the function that currently has MCSS control are returned respectively by control functions IP(i) and RP(i). The j th element of the i th integer and real list parameters are returned respectively by control functions ILP(i,j) and RLP(i,j).

Variables

MCSS function variables are stored in locations LV through LI-1 of IA/RA. There are four types of variables identical to the parameter types described above, and the last $IA(LB+4)$ elements of array IPVT contain integer values that define the corresponding variable types. Scalar

Functions

variables are stored directly in array IA/RA and list variables are stored indirectly in the list data structure at the location contained in the corresponding scalar integer variable. The *i*th integer and real variables of the function that currently has MCS control are returned respectively by control functions IV(*i*) and RV(*i*). The *j*th element of the *i*th integer and real list variable are returned respectively by control functions ILV(*i*,*j*) and RLV(*i*,*j*).

Initial Conditions

Initial condition values of function variables are stored in locations LI through LE of IA/RA. Initial condition types are the same as the corresponding variables and are defined by the variable type definitions in array IPVT. The *i*th integer and real initial condition value are returned respectively by control functions IIC(*i*) and RIC(*i*). The *j*th element of the *i*th integer and real list initial condition are returned respectively by control function ILIC(*i*,*j*) and RLIC(*i*,*j*).

COMMUNICATIONS

Address

Each MCSS function in a model system is assigned a positive integer address referred to as the function number. MCSS control programs are assigned the address number zero. Communications among MCSS control programs and functions of the model system are based upon these integer addresses and the communication type ICTYPE. Communications are transmitted from MFROM to MTO if ICTYPE is greater than zero. Response communications are returned to MFROM from MTO if ICTYPE is less than zero. An error message is written and the program is halted if a communication is attempted with ICTYPE equal to zero.

Type

Communication types, identified by the value of ICTYPE, are described in Table 2.1.

Data

Common location IPOINT is used to either convey or point to information conveyed in communications. In type 1 communications IPOINT contains the type number of the function to be created, and no other data are required. IPOINT is set to zero in communication types 2 through 7 which require no additional information. In event communications, type 8, IPOINT contains the location of the fourth word, the event type identification, of the event notice last removed from the scheduled event chain. Data stored for reference at the time the event occurs are, therefore, at location

TABLE 2 1 Communication types.

ICTYPE	Description
1	Create function. Transmitted from control program MAIN to the function specified in a CREATE command line.
2	Destroy function. Transmitted from control program MAIN to the function specified in a DESTROY command line.
3	Assign parameters. Transmitted from control program MAIN to the function specified in a PARAMETERS command line.
4	Assign variable initial condition values. Transmitted by control program MAIN to the function specified in a VARIABLES command line.
5	Write report on output device. Transmitted by control program MAIN to the function specified in a REPORT command line and from report generation functions to other functions whose reports are requested.
6	Initialize. Transmitted from control program MAIN to each function of a model system before each simulation run is started.
7	Dump. Transmitted to each function of a model system by control program DMPSVS to request a dump of the function data structure.
8	Event. Transmitted from control program CLOCK to the function identified in event notices removed in chronological order from the scheduled event chain.
9	Interaction. Transmitted from model function to model function during the operation of the simulated system.

IA(IPPOINT+1). In interaction communications, type 9, IPPOINT contains the location of a message block of ten contiguous words in IA/RA. The data structure of these message blocks is shown in Figure 2.10. The first word points to additional message blocks if required. The second word contains the integer 8, the number of useable locations in the block allocated from

free store. The third word contains a positive, non-zero integer indicating the type of interaction message defined in Table 2.2. The contents of the last eight words of an interaction message block vary, depending upon the interaction message type, and are defined in Tables 2.3 to 2.12.

FIGURE 2.10 Interaction message block data structure

TABLE 2.2 Interaction message types

Type	Definition
1	Transfer material.
2	Material has been transferred
3	What material is available?
4	What material is held up?
5	What material is susceptible to diversion?
6	Material measurement data
7	Stimulus.
8	Perform control action.
9	Condition notice.
10	Control signals.
11	Start control action sequence.

TABLE 2.3 Interaction message:
Transfer material.

Word of Interaction Message Block	Contents
3	1
4	Location of material to be transferred.
5	Source function.
6	Destination function.
7	0
8	0
9	0
10	0

Communications

TABLE 2.4 Interaction message:
Material has been transferred.

Word of Interaction Message Block	Contents
3	2
4	Location of material transferred.
5	0
6	0
7	0
8	0
9	0
10	0

Communications

TABLE 2.5 Interaction message:
What material is available?

Word of Interaction Message Block	Contents
3	3
4	Location of available material.
5	0
6	0
7	0
8	0
9	0
10	0

Communications

TABLE 2.6 Interaction message:
What material is held up?

Word of Interaction Message Block	Contents
3	4
4	Location of held up material.
5	0
6	0
7	0
8	0
9	0
10	0

TABLE 2.7 Interaction message:
What material is susceptible to diversion?

Word of Interaction Message Block	Contents
3	5
4	Location of material susceptible to diversion.
5	Location of the message block last placed on the chain of message blocks pointing to material susceptible to diversion.
6	Number of message blocks that point to material susceptible to diversion.
7	0
8	0
9	0
10	0

TABLE 2.8 Interaction message:
Material measurement data.

Word of Interaction Message Block	Contents
3	6
4	Material type.
5	Observed count.
6	Observed mass.
7	True count.
8	True mass.
9	0
10	0

TABLE 2.9 Interaction message:
Stimulus.

Word of Interaction Message Block	Contents
3	7
4	Location of the real parameter list in which the stimulus characteristics are specified.
5	Index of the first element of the stimulus characteristics in the real list parameter.
6	0
7	0
8	0
9	0
10	0

Communications

TABLE 2.10 Interaction message:
Perform control action.

Word of Interaction Message Block	Contents
3	8
4	Number of control actions.
5	Control code.
6	Control data.
7	Control code.
8	Control data.
9	Location of integer list of control codes and data
10	0

Communications

TABLE 2.11. Interaction message:
Condition notice

Word of Interaction Message Block	Contents
3	9
4	Location of material susceptible to diversion.
5	Location of the message block last placed on the chain of message blocks pointing to material susceptible to diversion.
6	Number of message blocks that point to material susceptible to diversion.
7	0
8	0
9	0
10	0

TABLE 2 12 Interaction message
Control signals.

Word of Interaction Message Block	Contents
3	10
4	Number of control signals transmitted
5	Control signal index
6	Control signal value.
7	Control signal index.
8	Control signal value.
9	Location of integer list of control signal indexes.
10	Location of real list of control signal values.

TABLE 2.13 Interaction message:
Start control action sequence.

Word of Interaction Message Block		Contents
3	11	
4	0	
5	0	
6	0	
7	0	
8	0	
9	0	
10	0	

MATERIAL

Type

Material types are identified by positive, non-zero integer names. Any number of different material types may exist simultaneously in a model system and may be assigned arbitrary integer names.

Quantity

Distinct quantities of each material type are represented by an integer count of discrete items and/or the real value of material mass.

Containment

The type, count, and mass of distinguishable material quantities are stored in a recursive data structure that represents the transitive enclosure of materials in various types of containment configurations. The structure of material data blocks is shown in Figure 2.11, and linkage of adjacent blocks in the data structure is shown in Figure 2.12. The first word of each material data block points to additional data that may describe physical, chemical, or other properties of the material represented by the data block. The second word contains the integer 8, the number of useable locations in the block allocated from free store. The third, fourth, fifth, and sixth words contain pointers to the adjacent blocks in the material data structure as shown in Figure 2.12. The seventh word contains an integer 0, 1, or 2 indicating the containment level of the material quantity. The last three words of material data blocks contain respectively the integer

Material

material type, the count of discrete items, and the mass of material represented.

FIGURE 2.11 Material data block.

FIGURE 2.12 Linkage of adjacent material data blocks.

Material

A recursive definition of material containment is shown in Figure 2.13. In any container A there may be any number of bulk materials 1, 2, ... and any number of containers B, C, ... The bulk materials in any container are assumed to be homogeneously mixed, so they can be removed only in combination with other bulk materials in the container at the time of removal. The block chain data structure representing the recursive definition of material containment is shown in figure 2.14. To simplify the figure, only the 7th and 8th words of material data blocks are shown; linkages defined by pointers LH, LP, LN, and LL are indicated by the lines connecting the simplified material data blocks, and locations LA, LBULKA, etc. are the locations of the first word of each block.

FIGURE 2.13 Recursive definition of material containment.

Material

FIGURE 2.14 Data structure representing the recursive definition of material containment in Figure 2.13.

The block at LA represents all of the material in container A. The containment level is 2 because container A "contains" other containers. The material type is null (indicated by material type zero, which is otherwise an illegal material type code) because container A may contain more than one type of material. The last two words of the material data block at LA contain respectively the count of all discrete material items of all types in container A and the total mass of all materials of all types in container A.

The material data block at LBULKA represents all of the bulk materials

Material

in container A. Its containment level of 1 distinguishes the collection of bulk materials in container A from containers B, C, ... and indicates that the block is one level above blocks that represent specific material quantities. The last two words of the block at LBULKA contain respectively the count of all discrete material items of all types in bulk form within container A but outside containers B, C, ... and the total mass of all materials of all types in bulk form within container A but outside containers B, C,

The blocks at L1A, L2A, ... represent the distinct types of bulk material in container A but outside containers B, C, The containment level 0 of these blocks indicates that the material they represent can not be further subdivided into lower hierarchical levels of containment and by definition represent a quantity of a specific material type.

The material data blocks at LB, LC, ... represent the containers within container A and are roots of subtrees which have the recursive data structure shown in Figure 2.14. Thus, each of the containers B, C, ... may contain bulk materials and/or other containers, etc. etc. to any depth of nested material containment and data structure.

Figure 2.13 and 14 represent the most general material containment configuration which subsumes the special containment configurations shown in Figures 2.15 through 2.19.

Figure 2.15 shows a schematic representation and the data structure for a container A of bulk materials 1, 2, In this case, the LN pointer in the fifth word of the material data block at LBULKA contains a zero indicating there are no containers within container A.

Material

FIGURE 2.15 Container of bulk materials 1, 2, ...

A container that contains only other containers is illustrated in Figure 2.16. Here the LL pointer of the material data block at LA points to the block at LB which represents the first of containers B, C, ...

FIGURE 2.16 Container of other containers.

Material

Figure 2.17 shows the representation of an empty container. The containment level is 2 because the container could contain other containers. The linkage pointers, material type, count, and mass of the data block at LA are set to zero.

FIGURE 2.17 Empty container.

Figure 2.18 shows the representation of free bulk materials 1, 2, ... that are not in a container. The data structure representing these materials is at location LBULK. Free bulk materials are represented by the same sub-tree data structure that represents bulk materials within containers.

FIGURE 2.18 Free bulk materials 1, 2, ...

The containment configuration and data structure representing free containers B, C, ... is shown in Figure 2.19. Free containers are represented by the same sub-tree data structure that represents containers within other containers.

FIGURE 2.19 Free containers.

Example

In a container A, as shown in Figure 2.20, there are four bulk

Material

materials 1, 2, 5, and 6 and four other containers B, C, C, and E. Containers B and C hold the same amounts of bulk materials 3 and 4, and D and E contain the same amounts of bulk materials 7 and 8. Odd material types are counted as discrete items, even material types are measured in mass units, and the amounts of all materials are indicated in Figure 2.20.

FIGURE 2.20 Example material containment configuration.

The data structure for this example is shown in Figure 2.21, where the material data blocks now include the two words that contain the count

Material

and mass of material represented. As before, the pointers to adjacent blocks are represented by the lines that connect the blocks. And locations indicated at the top left of each block are locations of the first word of the ten-word material data blocks allocated by control program ASTORE.

Material

FIGURE 2.21 Data structure for the example material containment configuration in Figure 2.20.

Material

Figure 2.22 shows the material data structure of the example containment configuration as it is printed out by control program WRMD. From left to right, each line of Figure 2.22 contains the following:

L	Location of a material data block.
LH, LP, LN, LL	Locations of blocks adjacent to the block at L.
Level	Containment level of the block at L.
Type	Material type represented by the block at L.
Count	Item count of the quantity of material represented by the block at L.
Mass	Mass of the quantity of material represented by the block at L.

Material

L	LH	LP	LV	LL	Level	Type	Count	Mass
1824	0	1734	0	0	2	6	0	6.00000E+01
1934	0	2034	1824	0	0	0	0	0.
2034	0	2034	1934	0	2	0	0	6.00000E+01
2034	1944	0	2034	0	0	1	100	0.
1874	0	1854	0	0	0	4	0	6.00000E+01
1864	1854	0	1874	0	0	4	0	0.
1854	1844	0	0	1864	1	0	0	6.00000E+01
1914	0	1904	0	0	0	4	0	6.00000E+01
1904	1894	0	1914	0	0	4	0	0.
1894	1884	0	0	1904	1	0	0	6.00000E+01
2004	0	1994	0	0	0	6	0	6.00000E+01
1994	1984	0	2004	0	0	7	700	0.
1984	1974	0	0	1994	1	0	700	6.00000E+01
2044	0	2034	0	0	0	6	0	6.00000E+01
2034	2024	0	2044	0	0	7	700	0.
2024	2014	0	0	2034	1	0	700	6.00000E+01
2014	0	1974	0	2024	2	0	700	6.00000E+01
1974	0	1884	2014	1984	2	0	700	6.00000E+01
1854	0	1844	1974	1894	2	0	300	6.00000E+01
1844	0	1744	1854	1854	2	0	300	6.00000E+01
1944	1444	0	1844	2004	1	0	600	6.00000E+01
1444	0	0	0	1744	2	0	2000	7.00000E+01

FIGURE 2.22 Material data for the example material containment configuration printed by control program WRMO.

3 PROGRAM DESCRIPTION

MACROS

Declaration

The statement USE SPM defines the "file" SPM that contains the macro (cliche) descriptors and invokes the "macro" SPM that initializes parameters used in conditional macro processing. This declaration must precede the first macro statement in each program.

Structured Programming

Macros CASE, ELSEF, ENDPROC, FIN, FOR, IFF, NESTLEVEL, ORIF, OTHERWISE, PERFORM, PROCEDURE, REPEAT, UNTIL, and WHILE provide the control mechanisms of the following structured programming constructs.

```
IFF [logical expression]
  code block
FIN
```

```
IFF [logical expression]
  code block
ELSE
  code block
FIN
```

```
CASE [logical expression]
  code block
ORIF [logical expression]
  code block
ORIF [logical expression]
  code block
...
```

Macros

```
ORIF [logical expression]
  code block
FIN
```

```
CASE [logical expression]
  code block
ORIF [logical expression]
  code block
ORIF [logical expression]
  code block
...
```

```
ORIF [logical expression]
  code block
OTHERWISE
  code block
FIN
```

```
WHILE [logical expression]
  code block
FIN
```

```
REPEAT
  code block
UNTIL [logical expression]
```

```
FOR [variable] = [expression] TO [expression]
  code block
FIN
```

```
FOR [variable] = [expression] TO [expression] BY [expression]
  code block
FIN
```

```
PERFORM procedurename [integer]
```

```
PROCEDURE procedurename [integer]
  code block
ENDPROC [integer]
```

Input/Output

Macros

Macros RD, RDE, RDEND, and WR provide the following simplified input/output statements.

RD [U:d] [1] [2]

The RD macro reads values into list [1] according to format specification [2] from input device d. Default for data directed argument U is input device N11.

RDE [U:d] [1] [2]

...

RDEND

The RDE macro is similar to the RD macro except that the file on device d is closed and control is transferred to the statement following the RDEND statement if an "end of file" is encountered.

WR [U:d] [1] [2]

The WR macro writes from list [1] according to format specification [2] on output device d. Default for data directed argument U is output device N01.

Recurrent Code

Macros BEGINBLOCK, BEGINFUN, BEGINSUB, BLOCKS, CREATE, DESTROY, PARAMETERS, VARIABLES, REPORT, INITIALIZE, DUMP, EVENT, INTERACT, ENOBLOCK, ERROR, RESPOND, SEND, STORE, SYSTEM, TIMER, TR, and WARN simplify the programming of recurrent segments of code.

CONTROL PROGRAMS

ASTORE(LOC,N)

This subroutine allocates space from the free store array !A'RA. The calling program places the required number of useable locations in argument N. If there are less than N useable locations left in free store, the error message OUT OF FREE STORAGE SPACE is written and the program is halted. Otherwise, ASTORE returns with the location of the allocated space in argument LOC and the allocated number of useable locations in N. The allocated number of useable locations may exceed the required number by up to nine locations because ASTORE does not scan the free store chain beyond the first block that contains N or more locations, and free store blocks are not split if blocks of less than ten locations will be left in the free store chain. If there is no block in the free store chain containing N or more useable locations, a chain of blocks containing from N to at most N+9 useable locations will be allocated starting with the largest block then in free store. Trace level 5 messages are written upon entry and just before returning to the calling program.

CEC(MIN,MAX,KODE)

This subroutine checks the event code at !A(IPCINT). If it is less than MIN or greater than MAX, the error message ILLEGAL EVENT CODE is written and the program is halted. Otherwise, the event code is returned in argument KODE.

CLOCK

Control Programs

This subroutine removes notices from the scheduled event chain in chronological order, advances TIME to the time of the scheduled events, and sends event messages to the functions in which the events occur. Once a run has been started by calling CLOCK, TIME is advanced until it reaches a value equal to or greater than TSTOP or there are no more scheduled events to be processed (when this condition exists, TIME is advanced to TSTOP).

CMB(LM,LH,LP,LN,LL,NL,NT,NC,A)

This subroutine creates a material data block with linkage pointers LH, LP, LN, and LL; containment level NL; material type NT; item count NC; and mass A. The location of the material data block is returned in argument LM.

COPYB(L1,L2)

This subroutine copies the storage block at location L1 into a new block at L2. If the new block happens to be larger than the original (see the description of subroutine ASTORE for an explanation of why this is possible), the remainder of the block at L2 is assigned integer value zero. If there are not enough contiguous locations in free store to be allocated in a single new block, the error message NO CONTIGUOUS SPACE AVAILABLE is written and the program is halted.

CREATE(NT,NP,NV,IPVT)

This subroutine creates an instance of function type NT with NP parameters and NV variables. Parameters, variables, and initial condition locations are zeroed using the parameter and variable types defined by the

Control Programs

calling function program. The number assigned by the user to the created function is in common location MTO.

CRMD1(N0,LMS,LM)

This subroutine creates the material data structure shown in Figure 3.1 and assigns values to the data blocks to represent N0 free bulk materials defined by the material specifications at location LMS. The location of the free bulk material data structure is returned in argument LM.

FIGURE 3.1 Material data structure created by CRMD1(N0,LMS,LM)

CRMD2(N1,N0,LMS,LM)

This subroutine creates the material data structure shown in Figure 3.2 and assigns values to data blocks that represent a container that contains N1 other containers, each of which contain N0 bulk materials defined by the material specifications at location LMS. The location of the container of containers is returned in argument LM.

FIGURE 3.2 Material data structure created by CRMD2(N1,N0,LMS,LM).

DESTRO(IPVT)

This subroutine destroys the function at LB using the parameter and variable types in array argument IPVT.

DMPSYS

This subroutine writes the symbolic name and contents of each location in common block SYSTEM, sends a dump message to each function of the model system, and writes the status and contents of the scheduled event and free store chains.

Control Programs

DUMP(IPVT)

This subroutine writes, in dump format, the system data, parameters, variables, and initial conditions of the function at location LB.

ERROR

This subroutine calls subroutine DMPSYS to display the current status of the model system and then halts the program.

EVDRS(N,M EDT)

This subroutine returns a sample value V from a population with the distribution type in integer parameter N and the statistics in real list parameter M of the calling program. Trace level 5 messages are written upon entry and just before returning to the calling program.

FPNTRS(NFUN)

This subroutine set pointers LB, LP, LV, LI, and LE for function NFUN.

FRMD(LM)

This subroutine frees the storage space in the material data structure at location LM. Trace level 5 messages are written upon entry and just before returning to the calling program.

FSTORE(L0C)

This subroutine frees the storage space at location L0C. Trace level 5 messages are written upon entry and just before returning to the calling program.

Control Programs

IAC(L,I)

This function returns the integer value of the Ith useable location in the chain storage beginning at location L.

ICH(ICHAR)

This function returns an integer value corresponding to the position of A1 format character ICHAR in the string: 0 1 ... 9 blank A B .. Z \$ + - / = .) * (' ampersand. A -1 value is returned if ICHAR is not one of the 49 characters above.

ICHECK

This subroutine is called immediately after trace messages are written on the interactive output device. It responds, as described in the MCSS user's manual, to the following interactive inputs:

ABORT
EVENTS(N)
FREE
IA(N)
IA(N,M)
IAC(L,N)
IAC(L,N,M)
ICTYPE
IDUMP
IPoint
ISECA
ITRACE

Control Programs

LB
LBDA
LBLIST
LBPT
LBUF
LCSB
LE
LEC
LFFB
LFFE
LI
LLE
LLFB
LLFE
LMS
LNE
LP
LV
MATL(1L)
MFROM
MINNFL
MNSE
MSG
MTD
NFE
NFL

Control Programs

NI1
NI2
NO1
NO2
NREP
NRUN
NSE
RA(N)
RA(N,M)
RAC(L,N)
RAC(L,N,M)
TIME
TSTOP

IIIC(I)

This function returns the Ith integer initial condition value of the function at LB.

IIIC(I,J)

This function returns the Jth element of integer list initial condition I of the function at LB.

IIP(I,J)

This function returns the Jth element of integer list parameter I of the function at LB.

Control Programs

ILV(I,J)

This function returns the Jth element of integer list variable I of the function at LB.

INCRC(L1)

This subroutine writes the warning message INCREMENTING EVENT CHAIN, then allocates ISECA additional locations to the scheduled and free event chains.

IP(I)

This function returns the Ith integer parameter of the function at LB

IUIF(M,ISP)

This function returns an integer random variable that is uniformly distributed with mean M and maximum deviation (spread) ISP

IV(I)

This function returns the Ith integer variable of the function at LB.

KEYWD(L)

This function compares the A1 format character string starting at location L with the MCSS keywords CREATE, DESTROY, PARAMETER, VARIABLE, REPORT, RUN, and END and returns values 1 through 7 respectively if the character string matches one of the keywords. Value 0 is returned if there is no match.

Control Programs

LCS(LOC,ITH)

This function returns the location of the Ith element in chain storage starting at location LOC.

LIINMIN,M,I)

This function returns the value of LCS argument ITH that locates the Ith item in the Nth set of M items.

MAIN

This is the main (FORTRAN) program of MCSS. Depending upon the requirements of particular computing systems, it may contain statements that define the input and output files MCSSIN and MCSSOUT and the numbers of the devices on which these files reside. For the CDC 7600 computers at Lawrence Livermore Laboratory, the statement
PROGRAM MAIN(MCSSIN,MCSSOUT,TAPE5=MCSSIN,TAPE6=MCSSOUT)
defines these files on device numbers 5 and 6 respectively.

For this system, the statement
CALL CHANGE(5H+MCSS)

is also used to execute the program under drop file name +MCSS. These are the only machine dependent statements in program MAIN, however, other statements may vary in different installations as described below.

In the initialization of dynamic storage, locations IA(LFFB+1) and NFL must be assigned values 2 less than the dimension of array IA/RA.

In creating the block pointer table, the second argument of the statement CALL ASTORE(LBPT,) must be 1 greater than the maximum number of functions to be allowed in a model system.

In creating the message stack, the second argument of the statement CALL ASTORE(LMS,) must be 2 greater than 4 times the maximum number of messages allowed in the stack.

In creating the event chain, the second argument of the statement CALL ASTORE(LEC,) must be 5 times the maximum number of scheduled events, and ISECA must be assigned the increment size of the event chain allocation.

Appropriate values must also be assigned to symbolic device numbers N11, N12, N01, and N02.

Program MAIN writes the heading MATERIAL CONTROL SYSTEM SIMULATOR VERSION x.x date at the beginning of the program output file, then executes each job step specified by keywords in the input file. Comment or annotation lines in the input file are echoed into the output file. When an END keyword is encountered, the program writes the minimum number of free locations that existed during the execution of all preceding job steps and terminates execution of the program.

MSUR(LM,LMS,N,MT,NC,A,NCT,AT)

This subroutine simulates the measurement of the material defined by the data structure at LM according to the Nth measurement specification at location LMS. The material type to be measured is in argument MT. The subroutine returns the measured count and mass in arguments NC and A and the true count and mass in arguments NCT and AT respectively. Trace level 5 messages are written upon entry and just before returning to the calling program.

Control Programs

NCBM(LCC,NC,BM)

This subroutine returns the number of containers NC and total bulk mass BM in the material represented by the data structure at location LCC.

NSTATS(N)

This function returns the number of statistics required to define distribution type N.

PCOM(NT0,NCTYPE,I1,I2,I3,I4,I5,I6,I7,I8)

This subroutine prepares a communication block for transmission from the calling function to function NT0. The communication type is in argument NCTYPE and the integer data values are in arguments I1, I2, . . . I8.

PHLB

This subroutine prints the header of the function data structure at LB in dump format.

PICLB(IPVT)

This subroutine prints the initial condition values of the function data structure at LB in dump format. Array argument IPVT contains the parameter and variable type codes of the calling program.

PIDF(L,K,N)

This subroutine prints the function data structure item at location L in dump format. Argument K contains the parameter or variable type code.

Control Programs

Argument N contains the parameter or variable index of the item to be printed.

POP(L1,N,M)

This subroutine pops N integers off the stack at location L1 and places the values in array M. The first value removed from the stack is placed in location M(N), the second value is placed in location M(N-1), etc.

PPLB(IPVT)

This subroutine prints the parameter values of the function data structure at LB in dump format. Array argument IPVT contains the parameter and variable types of the calling program.

PUSH(L1,N,M)

This subroutine pushes N integers in array M into the stack at location L1. The value in M(1) is pushed onto the stack first, the value in M(2) is pushed onto the stack next, etc.

PVLB(IPVT)

This subroutine prints the variables of the function data structure at LB in dump format. Array argument IPVT contains the parameter and variable types of the calling program.

RAC(L,I)

This function returns the real value of the Ith useable location in chain storage starting at location L.

Control Programs

RIC(I)

This function returns the Ith real initial condition of the function at LB.

RLIC(I,J)

This function returns the Jth element of real list initial condition I of the function at LB.

RLP(I,J)

This function returns the Jth element of real list parameter I of the function at LB.

RLV(I,J)

This function returns the Jth element of real list variable I of the function at LB.

RNORM(RMU,SIGMA)

This function returns an approximation of a normally distributed random variable obtained by summation of 12 sample values from a uniform random number generator. Arguments RMU and SIGMA are respectively the mean and standard deviation of the normal distribution.

RNORMT(RMU,SIGMA,RMAX,RMIN)

This function returns a random variable obtained by truncating the values generated by function RNORM. Arguments RMU and SIGMA are respectively the mean and standard deviation of the untruncated normal

Control Programs

distribution. Arguments RMAX and RMIN are respectively the maximum and minimum values returned by RNDRMT.

RP(I)

This function returns the Ith real parameter of the function at LB.

RUNIFORM,SP)

This function returns a uniformly distributed, real random variable. Argument RM is the mean and SP is the maximum deviation or spread of the uniform distribution.

RV(I)

This function returns the Ith real variable of the function at LB.

SAMPL(I,J,V)

This subroutine returns sample value V of a random variable with distribution type in RAC(I,J+2) and statistics in RAC(I,J+3), RAC(I,J+4), ... etc. Trace level 5 messages are written upon entry and just before returning to the calling program.

SCHED(T,NB,K,LD)

This subroutine schedules an event with code K for occurrence at time T in function NB and with event data at location LD. Argument LD may contain value zero if no data are included in the event notice. An error message is written and the program is halted if there is an attempt to schedule an event prior to the present value of TIME.

Control Programs

SEND

This subroutine transmits communications among the MCSS control programs and functions of the model system. Messages are transmitted from function MFROM to function MTO if ICTYPE is greater than zero. Response messages are returned from function MTO to function MFROM if ICTYPE is less than zero. The sending function MFROM, the receiving function MTO, and the communication type ICTYPE are pushed onto the message stack each time a message is transmitted and are popped off the stack each time a response is returned. If the current addresses MFROM and MTO and the communication type do not agree with the values in the message stack when responses are transmitted, the error message ILLEGAL RESPONSE is written and the program is halted. This verifies the consistency of communications among interacting elements of the model system.

SET(KODE,IPVT)

This subroutine sets parameter and variable initial condition values for the function at LB. Parameter values are set if KODE contains a 1 and initial condition values are set if it contains a 2. The parameter and variable indexes and values to be assigned are read from input device N11 in the form

(i n m) values (i n m) values ...

where i is the parameter or variable index and n and m are the first and last list element indexes respectively. The n and m are omitted for scalar valued parameters or variables. Numerical values are converted to the type specified by array argument IPVT. Elements of the input data are blank

Control Programs

delimited and may be interspersed with any character strings that will not be interpreted as a numeric value.

TIMER

This subroutine supplies timing information based upon the interaction response of a 601 TIMR function. Argument K must contain timing code 2 or 3 of the calling program, otherwise the error message ILLEGAL TIMING CODE is written and the program is halted. If the current value of TIME is within the operating time segments of the calling function, TIMER returns a 1 in argument I and the current value of TIME in argument T. If TIME is not within the operating time segments of the calling function, TIMER returns the timing code in argument I and the time at which the next operating time segment will begin in argument T. For timing code 2, this is the starting time of the next shift, and for timing code 3, it is the starting time of the next interval. Subroutine TIMER is not called by functions with timing code 1 since these functions are specified to be in continuous operation.

TRNFRM(NC,FB,L1,L2)

This subroutine performs the transfer of material from the data structure at location L1 to the data structure at L2. Argument NC contains the number of containers to be transferred. If NC exceeds the number of containers at L1, an error message is written and the program is halted. If NC is negative, all of the containers at L1 are transferred. Argument FB contains the fraction of the bulk materials at L1 to be transferred. Containers and bulk materials may be transferred separately or in combination.

Control Programs

UMD(LM)

This subroutine updates the material data structure at location LM. Values of total count and total mass are summed and assigned to the appropriate blocks in the material data structure as it is scanned from right to left and from bottom to top.

URN(IX,U)

This subroutine returns a pseudo-random variable U that is uniformly distributed on the real interval 0. to 1.0. A corresponding pseudo-random integer IX on the interval 0 to 10000 is returned in argument IX. The pseudo-random sequence is initialized by calling URN with a negative integer in argument IX.

A suitable version of subroutine URN must be prepared for the particular computing system on which MCSS is installed. The version in Appendix A of this manual is for the CDC 7600 computers at Lawrence Livermore Laboratory. There the ORDERLIB subroutine IRNFL(X) is called to initialize the pseudo-random sequence at seed value X, the value used by pseudo-random number generation function RNFL to generate the next member of the sequence. The seed value is output in trace level 5 messages when URN is initialized. ORDERLIB subroutine LRNFL is called to reset RNFL to the value X returned by IRNFL.

WRMD(LM)

This subroutine writes the material data structure at LM on output device N01. The complete contents of material data blocks are written as the data structure is scanned from right to left and from bottom to top.

Function Programs

ZIA(L,N)

This subroutine zeroes N elements of array IA starting at L.

ZIAC(LOC)

This subroutine assigns integer zero values to all elements of the block chain in array IA/RA starting at location LOC.

ZRA(L,N)

This subroutine zeroes N elements of array RA starting at L.

ZRAC(LOC)

This subroutine assigns real zero values to all elements of the block chain in array IA/RA starting at location LOC.

ZTABLE(LOC)

This subroutine assigns integer zero values to the usable locations of the table data structure at location LOC.

FUNCTION PROGRAMS

B101

This subroutine implements 101 RCMB functions that simulate the receipt of material batches entering the model system. Parameters 1 through 14 are assigned by the user. The initial value of variable 1, the number of batches received at a simulated receiving station, is automatically zeroed.

Function Programs

B102

This subroutine implements 102 IDAS functions that simulate the initiation of diversion action sequences. List valued parameter 1 is assigned by the user. There are no model state variables.

B3C1

This subroutine implements 301 TRMB functions that simulate the transport of material batches within the model system. Parameters 1 to 3 are assigned by the user. Parameter 4 is a dummy used to save the message block returned to functions that place material batches in transport. Initial values of variables 1 to 3 which are respectively the number of batches in transport, the maximum number of batches in transport, and the time the maximum number first occurred are automatically zeroed by the subroutine. Dummy variable 4, which contains the location of the message block pointing to the material batch most recently placed in transport, is also automatically assigned value zero.

Communication blocks that point to material in transport are linked on a two-way chain as shown in Figure 3.3. IV(4) points to the block (material) most recently placed in transport. When a transport operation is completed, the blocks are removed from the chain by resetting the pointers of the adjacent blocks. I(A(LN+8)) of a block at LN points to the block previously placed in transport and I(A(LN+9)) points to the block next placed in transport.

FIGURE 3.3 Linkage of communication blocks that point to material in 301 TRMB functions.

B401

This subroutine implements 401 STOM functions that store bulk and/or containerized materials in the model system. Parameters 1 to 6 are assigned by the user. Dummy parameters 7 and 8 respectively point to the data structures that represent materials currently in storage and in holdup. Initial values of variables 1 to 15 used in data collection and computation of storage statistics are automatically initialized by the subroutine.

B501

Function Programs

This subroutine implements 501 INDA functions that simulate independent actions. Parameters 1 to 4 are assigned by the user. There are no model state variables.

B502

This subroutine implements 502 MAQA functions that simulate actions by which material is acquired from other functions in the model system. Parameters 1 to 5 are assigned by the user. Dummy parameter 6 points to the data structure representing diverted material. There are no model state variables.

B503

This subroutine implements 503 DEPA functions that simulate dependent actions. Parameters 1 to 6 are assigned by the user. Dummy parameter 7 points to the report data shown in Figure 3.3. Initial values of variables 1 to 7 are assigned automatically by the subroutine as follows: Variable 1, the time T1 that the action started, is initialized to value -2.0 to indicate that initially the action has not been started. Variable 2, the duration of the DEPA action, and variable 3, the time delay of the next action of the sequence, are initialized to value -1.0. Variable 4, the number of the next action function, and variable 5, the index that points to the time delay statistics of the next action in the sequence, are initialized to integer zero. Variable 6, which points to the parameter list that defines the action sequence, is also initialized to integer zero. Variable 7, the time TD the last condition notice was received, is

Function Programs

initialized to value -1.0 to indicate that initially no condition notices have been received.

FIGURE 3.3 Report data structure of 503 DEPA functions

B504

This subroutine implements 504 MODA functions that modify the values of parameters or the current values of variables in other functions of the model system. Parameters 1 to 6 are assigned by the user. There are no model state variables.

B601

This subroutine implements 601 TIMR functions that supply timing information used by other functions of a model system to control their operating schedules. Parameters 1 to 7 are assigned by the user. Variable 1, the time the current period will end, is initialized to the value of the specified schedule period. Variable 2, the current shift number, and

Function Programs

used to save pointers to data required for continuation of the action sequence.

B601

This subroutine implements 601 TIMR functions that supply timing information used by other functions of a model system to control their operating schedules. Parameters 1 to 7 are assigned by the user. Variable 1, the time the current period will end, is initialized to the value of the specified schedule period. Variable 2, the current shift number, and variable 3, the time the current shift will end, are initialized to integer and real value zero respectively. Variable 4, the time the next shift will begin, is initialized to the first shift starting time of list valued parameter 3. Variable 5, the current interval number, and variable 6, the time the current interval will end, are initialized to integer and real value zero respectively. Variable 7, the time the next interval will begin, is initialized to the first interval starting time of list valued parameter 6.

B602

This subroutine implements 602 SCAS functions that start control action sequences upon command by other functions in a model system. List valued parameter 1 is assigned by the user. There are no model state variables.

B701

This subroutine implements 701 MZMB functions that simulate the measurement of material batches and the transmission of measurement results

Function Programs

to other functions in a model system. Parameters 1 to 10 are assigned by the user. Dummy parameter 11 is used to save the location of message blocks returned to functions that place material in the measurement function. Variable 1, the number of batches currently in measurement; variable 2, the maximum number of batches concurrently measured; and variable 3, the time the maximum first occurred, are initialized to value zero. Dummy variable 4, which points to the message block that points to the material batch most recently placed in the measurement function, is also initialized to integer zero.

8702

This subroutine implements 702 CMBA functions that record data and compute material balance statistics using material measurement information received from other functions in a model system. Parameters 1 to 4 are assigned by the user. Dummy parameter 5 points to the report data structure shown in Figure 3.4.

Function Programs

b702

This subrouting implements 702 CMBA functions that record data and compute material balance statistics using material measurement information received from other functions in a model system. Parameters 1 to 4 are assigned by the user. Dummy parameter 5 points to the report data structure shown in Figure 3.4.

FIGURE 3.4 Report data structure of 702 CMBA functions

B703

This subroutine implements 703 MNTR functions that simulate material

Function Programs

control components that monitor type 1 stimuli generated by other functions in a model system. Parameters 1 to 8 are assigned by the user. Variable 1, the monitor detection signal state, is initialized to integer zero.

B801

This subroutine implements 801 DIFBL functions that provide decision making capability in model systems based on the standard form and notation of limited entry decision tables. Parameters 1 to 1' are assigned by the user. Dummy parameter 12 points to the report data structure shown in Figure 3 5

FIGURE 3 5 Report data structure of 801 DTBL functions

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in \mathbb{R}^2, \quad y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \in \mathbb{R}^2, \quad z = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \in \mathbb{R}^2, \quad w = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \in \mathbb{R}^2, \quad v = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \in \mathbb{R}^2, \quad u = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \in \mathbb{R}^2, \quad t = \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} \in \mathbb{R}^2, \quad s = \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} \in \mathbb{R}^2, \quad r = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix} \in \mathbb{R}^2, \quad q = \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} \in \mathbb{R}^2, \quad p = \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} \in \mathbb{R}^2, \quad o = \begin{pmatrix} o_1 \\ o_2 \end{pmatrix} \in \mathbb{R}^2, \quad n = \begin{pmatrix} n_1 \\ n_2 \end{pmatrix} \in \mathbb{R}^2, \quad m = \begin{pmatrix} m_1 \\ m_2 \end{pmatrix} \in \mathbb{R}^2, \quad l = \begin{pmatrix} l_1 \\ l_2 \end{pmatrix} \in \mathbb{R}^2, \quad k = \begin{pmatrix} k_1 \\ k_2 \end{pmatrix} \in \mathbb{R}^2, \quad j = \begin{pmatrix} j_1 \\ j_2 \end{pmatrix} \in \mathbb{R}^2, \quad i = \begin{pmatrix} i_1 \\ i_2 \end{pmatrix} \in \mathbb{R}^2, \quad h = \begin{pmatrix} h_1 \\ h_2 \end{pmatrix} \in \mathbb{R}^2, \quad g = \begin{pmatrix} g_1 \\ g_2 \end{pmatrix} \in \mathbb{R}^2, \quad f = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} \in \mathbb{R}^2, \quad e = \begin{pmatrix} e_1 \\ e_2 \end{pmatrix} \in \mathbb{R}^2, \quad d = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} \in \mathbb{R}^2, \quad c = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} \in \mathbb{R}^2, \quad b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \in \mathbb{R}^2, \quad a = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \in \mathbb{R}^2, \quad \text{etc.}$$

4. ADDITION OF NEW FUNCTIONS

TYPE NUMBER

MCSS functions are assigned unique 3-digit type number. The most significant digit ranges from 1 to 8 and indicates the class of the model and function as shown in Table 3.1 of the MCSS User's Manual. The least significant two digits indicate the particular function type within a particular class of functions. These have been assigned in numerical order as functions were developed as a needed whenever a function, and this procedure will be continued as new functions are added to the modeling repertoire.

COMMUNICATION CONTROL

Communications among model functions are handled by control subroutine SEND which must be capable of identifying each function type and calling the appropriate subroutine. In the SEND subroutine shown in Appendix A provisions have been made for communications among the first five functions in each category. Dummy subroutines are supplied for those functions not yet defined, so addition of these function types requires only replacement of the dummy with the appropriate new subroutine.

SUBROUTINE ORGANIZATION

The subroutines that implement MCSS functions are simplified and standardized by use of macros BEGINBLOCK, CREATE, DESTROY, PARAMETERS, VARIABLES, REPORT, INITIALIZE, DUMP, EVENT, RESPOND, INTERACTION, PROCEDURE

NEW FUNCTION SUBROUTINE ORGANIZATION

ENDPROC, and ENDBLOCK The standard form of each MCSX function subroutine is as follows

```

SUBROUTINE B///
USE SPH
BEGINBLOCK [///] (number parameters) (number variables) (IPV values)
CREATE
DESTROY
PARAMETERS
VARIABLES
REPORT [XXX mnemonic name]
INITIALIZE
DUMP
EVENT
(event coding)
RESPOND [-8] [0]
INTERACTION
(interaction coding)
RESPOND [-9] (return value of ICTYPE)
PROCEDURE procedurename [:]
(procedure coding)
ENDPROC [!]
(other procedure definitions)
ENDBLOCK

```

The standard macros are replaced as required to create special data structures, return allocated storage space when functions are destroyed, produce error messages when illegal attempts are made to initialize variables, etc. For examples, see the function programs in Appendix A

REFERENCES

1. R. B. Hollister, Material Control System Simulation, Lawrence Livermore Laboratory, January 1978.
2. N. Wirth, Program Development by Stepwise Refinement, Communications of the ACM, 14, No. 4, April, 1971.
3. J. van der Linden, Systematic Programming, An Introduction, Prentice-Hall, Englewood Cliffs, 1973.
4. G. J. Dahl, E. W. Dijkstra, and C. A. R. Hoare, Structured Programming, Academic Press, New York, 1971.
5. E. Yourdon, Techniques of Program Structure and Design, Prentice-Hall, Englewood Cliffs, 1975.

Appendix A
SOURCE PROGRAM

MACROS

```
CLICHE SPM
PARAMETER([1=51]=0, [2=52]=0, [3=53]=0, [4=54]=0, [5=55]=0)
PARAMETER([6=56]=0, [7=57]=0, [8=58]=0, [9=59]=0, [10=60]=0)
PARAMETER([11=61]=0, [12=62]=0, [13=63]=0, [14=64]=0, [15=65]=0)
PARAMETER([16=66]=0, [17=67]=0, [18=68]=0, [19=69]=0, [20=70]=0)
PARAMETER([21=71]=0, [22=72]=0, [23=73]=0, [24=74]=0, [25=75]=0)
PARAMETER([P1=0], [P2=0], [P3=0], [P4=0], [P5=0], [P6=0], [P7=0], [P8=0], [P9=0])
PARAMETER([P10=0], [P11=0], [P12=0], [P13=0], [P14=0], [P15=0], [P16=0], [P17=0])
PARAMETER([P18=0], [P19=0], [P20=0], [P21=0], [P22=0], [P23=0], [P24=0], [P25=0])
[101:]
[102:]
[103:]
[104:]
[105:]
[106:]
[107:]
[108:]
[109:]
[110:]
[111:]
[112:]
[113:]
[114:]
[115:]
[116:]
[117:]
[118:]
[119:]
[120:]
[121:]
[122:]
[123:]
[124:]
[125:]
MACRO CASE, ELSE, ENDPROC, FIN, FOR, IFF, NESTLEVEL, ORIF, OTHERWISE,
PERFORM, PROCEDURE, RD, RDE, RDENS, REPEAT, UNTIL, WHILE, WR,
IBEGINBLOCK, BEGINFUN, BEGINSUB, CREAT, CREATE, DESTROY,
IPARAMETERS, VARIABLES, REPORT, INITIALIZE, DUMP,
IEVENT, INTERACT, ENDBLOCK, ERROR, RESPOND, SEND,
```

```

        ENDCLICHE
        CLICHE CASE
        IF [1]
        ENDCLICHE
        CLICHE ELSE
        DEFAULT [1] [1]
        PARAMETER(EL=1, L=4+2, FLAG=FIN)
        GO TO [5=FIN]
(5) ELSE CONTINUE
        PARAMETER(EL=1, FLAG=0)
        ENDCLICHE
        CLICHE ENDPROC
        PARAMETER(P=EC+1)
        PARAMETER(P=CC+1)
        GO TO [5=P] IF=P
        ENDCLICHE
        CLICHE FIN
        PARAMETER(EL=1, L=4+2, FLAG=FIN)
        IF [1=FLAG] NE 0
(6) ELSE CONTINUE
        IF [1=FLAG] NE 1
(7) ELSE CONTINUE
        IF [1=FLAG] NE 2
        GO TO [5=ELSE]
        ELSE CONTINUE
        IF [1=FLAG] NE 3
(8) ELSE CONTINUE
        ELSE CONTINUE
        PARAMETER(L=L+1)
        ENDCLICHE
        CLICHE FOR
        DEFAULT [4] [1]
        DEFAULT [6] [2]
        PARAMETER(L=L+1)
        NESTLEVEL
        PARAMETER(THEN=1, L=4+1, ELSE=FLAG=FIN)
        IF [2]-(4)
(5) ELSE [1]-(4)
        IF [4]-(1) (3) LE 0 GO TO [6=THEN]
        GO TO [7=FIN]
(8) CONTINUE
        PARAMETER(L=FLAG=0)
        ENDCLICHE
        CLICHE IF
        DEFAULT [4] [0]
        PARAMETER(L=L+1)
        NESTLEVEL
        PARAMETER(THEN=L=L+1, ELSE=FLAG=FIN)
        IF [1] GO TO [2=THEN]
        GO TO [3=ELSE]
(2) CONTINUE

```


Source Program Macros

```

PARAMETER(_[4]=_FLAG=0,_[5]=_FIN=0)
ENDCLICHE
CLICHE NESTLEVEL
DIF([L LT 13]3
NEST LEVEL EXCEEDS 12
PRECOMP TERMINATED BY INTENTIONAL USE OF ILLEGAL GLOBAL ARG.
PARAMETER(_L=[=999])
ENDCLICHE
CLICHE ORIF
DEFAULT [4] [3]
PARAMETER( THEN(1)=_L 4+1,_ELSE,_FLAG,_FIN)
GO TO [= _FIN]
[= _ELSE] IF([1])GO TO [2=_THEN]
GO TO [3=_ELSE]
[2] CONTINUE
PARAMETER(_[4]=_FLAG=0)
ENDCLICHE
CLICHE OTHERWISE
DEFAULT [1] [1]
PARAMETER( ELSE(1)=_L 4+2,_FLAG,_FIN)
GO TO [= _FIN]
[= _ELSE] CONTINUE
PARAMETER(_[1]=_FLAG=C)
ENDCLICHE
CLICHE PERFORM
PARAMETER(_P=30+[1])
ASSIGN [2] TO L[=_P]
GO TO [= _P]
[2] CONTINUE
PARAMETER( P[1]= P[1]+1)
PARAMETER( PL=100+[1])
DIF( P[1].EQ.1)1,
[ PL=_PL ]
[ PL=_PL[2]]
ENDCLICHE
CLICHE PROCEDURE
PARAMETER(_P=50+[1])
[=_P] CONTINUE
ENDCLICHE
CLICHE RD
DEFAULT [U] [N1]
READ([U],[3])[1]
[3] FORMAT([2])
ENDCLICHE
CLICHE RDE
DEFAULT [U] [N1]
READ([U],[3])[1]
IF(EOF([U])GO TO [=49]
[3] FORMAT([2])
ENDCLICHE
CLICHE RDEND
GO TO [1]

```

Source Program Macros

```

[=49] IEOF=}
      CALL UNLOAD (N1)
[1] CONTINUE
      ENDCliche
      CLICHE REPEAT
      PARAMETER(_L=_L+1)
      NESTLEVEL
      PARAMETER(_THEN(1)=_L 4+1,_ELSE,_FLAG,_FIN)
[1=_THEN] CONTINUE
      ENDCliche
      CLICHE UNTIL
      PARAMETER(_THEN(1)=_L 4+1,_ELSE,_FLAG,_FIN)
      IF(1)GO TO [2]
      GO TO [=_THEN]
[2] CONTINUE
      PARAMETER(_L=_L-1)
      ENDCliche
      CLICHE WHILE
      DEFAULT [4] [2]
      PARAMETER(_L=_L+1)
      NESTLEVEL
      PARAMETER(_THEN(1)=_L 4+1,_ELSE,_FLAG,_FIN)
[5=_ELSE] IF(1)GO TO [2=_THEN]
      GO TO [3=_FIN]
[2] CONTINUE
      PARAMETER(_[4=_FLAG]=0)
      ENDCliche
      CLICHE WR
      DEFAULT [U] [NO1]
      WRITE([U],[3])[1]
[3] FORMAT([2])
      ENDCliche
      CLICHE BEGINBLOCK
      PARAMETER(_R=0)
      PARAMETER(_[1=86]=0)
      PARAMETER(_IPVT=[2=87]+[3=88])
      DIMENSION IPVT(_IPVT)
      STORE
      SYSTEM
      DATA IPVT/([4])/
      [A(LV)=A(LV)+1]
      IF(CTYPE GE. 0)GO TO 99991
      GO TO [5=89]
99991 GO TO (99100,99200,99300,99400,99500,99600 99700,99800,99900)
X, CTYPE
99992 RETURN
      ENDCliche
      CLICHE BEGINFUN
      STORE
      SYSTEM
      ENDCliche
      CLICHE BEGINSUB

```

```

STORE
SYSTEM
ENDCLICHE
CLICHE CREATE
99100 CALL CREATE([=86],[=87],[=88] (PVT)
ENDCLICHE
CLICHE CREATE
99200 CALL CREATE([=86],[=87],[=88] (PVT)
RESPOND [-1] (0)
ENDCLICHE
CLICHE DESTROY
99300 CALL DESTROY(PVT)
RESPOND [-2] (0)
ENDCLICHE
CLICHE PARAMETERS
99400 CALL SET(1,(PVT,INTD,REALD)
RESPOND [-3] (0)
ENDCLICHE
CLICHE VARIABLES
99500 CALL SET(2,(PVT,INTD,REALD)
RESPOND [-4] (0)
ENDCLICHE
CLICHE REPORT
99600 CONTINUE
WR ([A(LB+2)] &
  17,SH,FUNCTION 14 1X,1=1THRU)  - FOR REPORT
RESPOND [-5] (0)
ENDCLICHE
CLICHE INITIALIZE
99600 CONTINUE
FOR (I) = 11 TO [A(LB+4)]
  [A(LV+I)]=[A(LI+I)+1]
FIN
RESPOND [-6] (0)
ENDCLICHE
CLICHE DUMP
99700 CALL DUMP(PVT)
RESPOND [-7] (0)
ENDCLICHE
CLICHE EVENT
99800 CONTINUE
TR (I) (EVENT)
ENDCLICHE
CLICHE INTERACT
99900 CONTINUE
TR (3) (INTERACTION)
ENDCLICHE
CLICHE ENDBLOCK
DIF (R.GT D)2,
[=89] CONTINUE
ERROR (ILLEGAL RESPONSE COMMUNICATION (BNTN))
DIF (R.LT,1)1,

```

Source Program Macros

```

[=89] 100000=IA(LP)
      DIF( R.NE.1),
      GO TO (=78)
      DIF( R.NE.2),
      GO TO ((=78), (=77)), 100000
      DIF( R.NE.3),
      GO TO ((=78), (=77), (=78)), 100000
      DIF( R.NE.4),
      GO TO ((=78), (=77), (=78), (=79)), 100000
      DIF( R.NE.5),
      GO TO ((=78), (=77), (=78), (=79), (=80)), 100000
      DIF( R.NE.6),
      GO TO ((=78), (=77), (=78), (=79), (=80), (=81)), 100000
      DIF( R.NE.7),
      GO TO ((=78), (=77), (=78), (=79), (=80), (=81), (=82)), 100000
      DIF( R.NE.8),
      GO TO ((=78), (=77), (=78), (=79), (=80), (=81), (=82), (=83)), 100000
      DIF( R.NE.9),
      GO TO ((=78), (=77), (=78), (=79), (=80), (=81), (=82), (=83), (=84)),
      100000
      DIF( R.NE.10),
      GO TO ((=78), (=77), (=78), (=79), (=80), (=81), (=82), (=83), (=84),
      (=85)), 100000
      END
      ENDCliche
      CLICHE ERROR
      WR I 2 /1X,15H      ERROR      /1X,(+1)H[1]
      CALL ERROR
      ENDCliche
      CLICHE RESPOND
      ICTYPE=11
      IPOINT=12
      GO TO 99992
      ENDCliche
      CLICHE SEND
      PARAMETER( R=_R+1)
      DIF( R.LT.11),
      NUMBER OF REENRANT POINTS EXCEEDS 10
      PRECOMP TERMINATED BY INTENTIONAL USE OF ILLEGAL GLOBAL ARG
      PARAMETER(_R=9999)
      IA(LP)=R
      GO TO 99992
      PARAMETER(_G=_R+75)
[1=_G] CONTINUE
      ENDCliche
      CLICHE STORE
      DIMENSION IA(1)
      COMMON RA(1)
      EQUIVALENCE (IA(1),RA(1))
      ENDCliche
      CLICHE SYSTEM
      COMMON /SYSTEM,MFROM,MTO,ICTYPE,IPOINT,

```

Source Program Macros

```

1TDUMP,ISFCA,ITRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSE,LE,
1LEC,LFFB,LFFE,LI,LLE,LLFB,LLFE,LMS,LNE,LP,LV,MINMF,
1MNSÉ,NFE,NFL,N11,N12,NO1,NO2,NREP,NRUN,NSÉ,TIME,TSTOP
ENDCLICHE
CLICHE TIMER
IFF (IP(1)) EQ 11
    I=1
    T=TIME
ELSE
    MFROM=IA(LB+2)
    MTO=IP(12)
    ICTYPE=9
    IPOINT=0
    SEND
    CALL TIMER(IP(11),I,T)
    FIN
ENDCLICHE
CLICHE TR
IFF (ITRACE.GE.(1))
    WR (TIME,IA(LB+2)) (1X,E10 4,14,2X,(+2)H(2))
    DIF((-2).EQ.2)1,
    WR ((3)) ((4))
    FIN
PARAMETER( T=(1)+1)
IFF (ITRACE.GE. T.AND.(ITRACE.EQ.ITRACE/2 2))
    WR (U:NO2) (TIME,IA(LB+2)) (1X,E10 4,14,2X,(+2)H(2))
    DIF((-2).EQ.2)1,
    WR (U:NO2) ((3)) ((4))
    CALL ICHECK
    FIN
ENDCLICHE
CLICHE WARN
WR (U:59) ( 1 (1X,17H      WARNING      /1X,(+1)H(1))
ENDCLICHE

```

CONTROL PROGRAMS

```

C ASTORE(LOC,N)
C SUBROUTINE ALLOCATES STORAGE SPACE IN ARRAY :A/RA
  SUBROUTINE ASTORE(LOC,N)
    USE SPM
    BEGINSUB
    TR [5] [ASTORE(LOC,N)] [N] [1X,110]
    NUM=N
    IFF [NUM.LT.8]
      NUM=8
    FIN
    IFF [NUM.GT.NFL]
      ERROR [ASTORE OUT OF FREE STORAGE SPACE]
    FIN
    PERFORM SCAN [1]
    CASE [L,LE,0]
      ERROR [ASTORE NO STORAGE BLOCK FOUND IN SCAN]
    ORIF [IA(L+1).GE.NUM+10]
      PERFORM ALLOCATE-SEGMENT [2]
    ORIF [IA(L+1).GE.NUM]
      PERFORM ALLOCATE-BLOCK [3]
    ORIF [IA(L+1).LT.NUM]
      PERFORM ALLOCATE-CHAIN [4]
    FIN
    IFF [NFL.LT.MINNFL]
      MINNFL=NFL
    FIN
    TR [5] [ASTORE(LOC,N)] [LOC,N] [1X,2110]
    RETURN
C
  PROCEDURE SCAN [1]
    L=LFFB
    M=IA(L+1)
    LM=L
    LPP=0
    LPM=0
    WHILE [IA(L+1).LT.NUM.AND.L.NE.0]
      IFF [IA(L+1).GT.M]
        M=IA(L+1)
        LM=L
        LPM=LPP
      FIN
      IFF [LBCA+IA(LBCA+1)+2.EQ.L.OR.L+IA(L+1)+2.EQ.LBCA]
        PERFORM COMBINE-BLOCKS [5]
      FIN
      LPP=L
      L=IA(L)
    FIN
    IFF [L.EQ.0]

```

Source Control Programs

```

        L=LM
        FIN
        LBCA=IA(LBCA)
        IFF [LBCA.EQ.O]
            LBCA=LFFB
        FIN
    ENDPROC [1]
C
    PROCEDURE ALLOCATE-SEGMENT [2]
    IFF [L.EQ.LFFB]
        LFFB=L+2*NUM
    ELSE
        IA(LPP)=L+2*NUM
        FIN
        IFF [L.EQ.LFFB]
            LFFB=L+2*NUM
        FIN
        IA(L+2*NUM)=IA(L)
        IA(L+3*NUM)=IA(L+1)-NUM-2
        NFL=NFL-NUM-2
        IFF [LBCA.EQ.L]
            LBCA=L+2*NUM
        FIN
        IA(L)=O
        IA(L+1)=NUM
        IA(L+2)=NUM
        LOC=L
    ENDPROC [2]
C
    PROCEDURE ALLOCATE-BLOCK [3]
    IFF [L.EQ.LFFB]
        LFFB=IA(L)
    ELSE
        IA(LPP)=IA(L)
        FIN
        NFL=NFL-IA(L+1)
        IFF [LBCA.EQ.L]
            LBCA=IA(L)
            IFF [LBCA.EQ.O]
                LBCA=LFFB
            FIN
        FIN
        IA(L)=O
        IA(L+2)=IA(L+1)
        LOC=L
    ENDPROC [3]
C
    PROCEDURE ALLOCATE-CHAIN [4]
    IFF [LM.NE.LFFB]
        IA(LPM)=O
        IA(LLFB)=LFFB
        LLFB=LPM

```

Source Control Programs

```

LFFB=LM
LPM=0
FIN
L1=LCS(LM,NUM)
L2=LCSB+1+IA(LCSB+1)
WHILE (L.NE.LBCA.AND.L.NE.LCSB)
  L=IA(L)
  FIN
  IFF (L.EQ.LBCA)
    PERFORM REASSIGN-LBCA [6]
  FIN
  IFF (L2-L1.GE.10)
    IA(LM+2)=NUM
    LFFB=L1+1
    IA(LFFB)=IA(LCSB)
    IA(LFFB+1)=L2-L1-2
    NFL=NFL-NUM-2
  ELSE
    IA(LM+2)=NUM+L2-L1
    LFFB=IA(LCSB)
    NFL=NFL-NUM-L2+L1
  FIN
  IFF (LFFB.EQ.0)
    WARN (ASTORE LAST AVAILABLE BLOCK ALLOCATED)
  LFFB=0
  FIN
LOC=LM
ENDPROC [4]
C
PROCEDURE COMBINE-BLOCKS [5]
  IFF (LBCA.LT.L1)
    IA(LBCA)=IA(L)
    IA(LBCA+1)=IA(LBCA+1)+IA(L+1)+2
    IA(LPP)=LBCA
  ELSE
    IA(L+1)=IA(L+1)+IA(LBCA+1)+2
  FIN
  LBCA=IA(LBCA)
  IFF (LBCA.EQ.0)
    LBCA=LFFB
  FIN
  NFL=NFL+2
ENDPROC [5]
C
PROCEDURE REASSIGN-LBCA [6]
  IFF (L2-L1.GE.10)
    LBCA=L1+1
  ELSE
    LBCA=IA(LCSB)
  FIN
ENDPROC [6]
END

```


Source Control Programs

```

C CEC(MIN,MAX,KODE)
C SUBROUTINE CHECKS EVENT CODE AT IA(IPOINT) AND PRINTS ERROR MESSAGE
C IF NOT IN THE INTERVAL [MIN,MAX].
  SUBROUTINE CEC(MIN,MAX,KODE)
    USE SPM
    BEGIN SUB
      KODE=IA(IPOINT)
      IFF [KODE.LT.MIN.OR.KODE.GT.MAX]
        ERROR [CEC ILLEGAL EVENT CODE]
      FIN
    RETURN
  END
C CLOCK
C SUBROUTINE ADVANCES SIMULATION TIME.
  SUBROUTINE CLOCK
    USE SPM
    BEGIN SUB
      WHILE [TIME.LT.TSTOP]
        CASE [LNE.GT.0]
          TIME=RA(LNE+1)
          MFROM=0
          MTO=1/(LNE+2)
          ICTYPL=8
          IPOINT=LNE+3
          CALL SEND
          N=IA(LNE)
          IA(LNE)=LFFE
          LFFE=LNE
          LNE=N
          NSE=NSE+1
          NFE=NFE+1
          ORIF [LNE.EQ.0]
            TIME=TSTOP
          OTHERWISE
            ERROR [CLOCK NEGATIVE LNE]
          FIN
        FIN
      RETURN
    END
C CMB(LM,LH,LPB,LN,LL,NL,NT,NC,A)
C SUBROUTINE CREATES MATERIAL BLOCK INITIALIZED TO LH,LPB,...,A
C AND RETURNS BLOCK LOCATION LM.
  SUBROUTINE CMB(LM,LH,LPB,LN,LL,NL,NT,NC,A)
    USE SPM
    BEGIN SUB
      CALL ASTORE(LM,8)
      IFF [IA(LM).NE.0]
        ERROR [CMB NO CONTIGUOUS SPACE FOR MATERIAL BLOCK]
      FIN
      IA(LM+2)=LH
      IA(LM+3)=LPB
      IA(LM+4)=LN

```

Source Control Programs

```

      IA(LM+5)=LL
      IA(LM+6)=NL
      IA(LM+7)=NT
      IA(LM+8)=NC
      RA(LM+9)=A
      RETURN
    END
C COPYB(L1,L2)
C SUBROUTINE COPIES THE STORAGE BLOCK AT L1 INTO A NEW
C BLOCK AT L2 AT LEAST AS LARGE AS THE ORIGINAL AND SETS
C THE REMAINDER OF THE BLOCK AT L2 TO INTEGER 0.
      SUBROUTINE COPYB(L1,L2)
      USE SPM
      BEGINSUB
      CALL ASTORE(L2,IA(L1+1))
      IFF [IA(L2).GT.0]
        ERROR [COPYB NO CONTIGUOUS SPACE AVAILABLE]
      FIN
      FOR [I] = [1] TO [IA(L2+1)]
        IFF [I.LE. IA(L1+1)]
          IA(L2+1+I)=IA(L1+1+I)
        ELSE
          IA(L2+1+I)=0
        FIN
      FIN
      RETURN
    END
C CREATE(NT,NP,NV,IPVT)
C SUBROUTINE CREATES INSTANCE MTO OF FUNCTION TYPE NT WITH NP PARAMETERS
C AND NV VARIABLES. PARAMETER, VARIABLE AND IC LOCATIONS ARE ZEROED
C USING PARAMETER AND VARIABLE TYPE CODES IN IPVT.
      SUBROUTINE CREATE(NT,NP,NV,IPVT)
      USE SPM
      DIMENSION IPVT(1)
      BEGINSUB
      NSIZE=3+NP+2 NV
      CALL ASTORE(LB,NSIZE)
      IFF [IA(LB).GT.0]
        ERROR [CREATE NO CONTIGUOUS STORAGE FOR FUNCTION]
      FIN
      IA(LCS(LBPT,MTO+1))=LB
      IA(LB)=NT
      IA(LB+2)=MTO
      IA(LB+3)=NP
      IA(LB+4)=NV
      LP=LB+5
      LV=LP+NP
      LI=LV+NV
      LE=LI+NV-1
      FOR [I] = [1] TO [NP]
        IFF [IPVT(I).EQ.3]
          RA(LP+I+1)=0.

```

Source Control Programs

```

        ELSE
          IA(LP+1)=0
        FIN
      FIN
    FOR I1 = 11 TO IN1
      IFF I1PV1(NP+1).EQ.31
        RA(LV-1+1)=0.
        RA(LV-1+1+NV)=0.
      ELSE
        IA(LV-1+1)=0
        IA(LV-1+1+NV)=0
      FIN
    FIN
  RETURN
END
C CRMD1(NO,LMZ,LM)
C SUBROUTINE CREATES NO MATERIALS AT CONTAINMENT LEVEL 1 ACCORDING
C TO MATERIAL SPECIFICATIONS AT LMZ AND RETURNS THE LOCATION IN LM.
SUBROUTINE CRMD1(NO,LMZ,LM)
  USE SPM
  BEGIN SUB
    TR I51 [CRMD1(NO,LMZ, )] [NO,LMZ] [1X,2:10]
    CALL CMB(L1,0,0,0,0,1,0,0,0)
    FOR IN1 = 11 TO IN1
      CALL CMB(L0,0,0,0,0,0,0,0,0)
      IA(L0+7)=RAC(LMZ,L1IN1(NM,7,1))
      IA(L0+8)=RAC(LMZ,L1IN1(NM,7,2)),RAC(LMZ,L1IN1(NM,7,3))
      RA(L0+9)=RNORMT(RAC(LMZ,L1IN1(NM,7,4)),RAC(LMZ,L1IN1(NM,7,5)),
1    RAC(LMZ,L1IN1(NM,7,6)),RAC(LMZ,L1IN1(NM,7,7)))
      IFF IN1.EQ.11
        IA(L0+2)=L1
        IA(L1+5)=L0
      ELSE
        IA(L0+3)=LOP
        IA(L0+4)=L0
      FIN
    LOP=L0
  FIN
  CALL UMD(L1)
  LM=L1
  TR I51 [CRMD1( , ,LM)] [LM] [1X,110]
  RETURN
END
C CRMD2(N1,NO,LMZ,LM)
C SUBROUTINE CREATES N1 CONTAINERS AT CONTAINMENT LEVEL 2 WITH
C NO MATERIALS IN EACH ACCORDING TO MATERIAL SPECIFICATIONS AT
C LMZ AND RETURNS THE LOCATION IN LM.
SUBROUTINE CRMD2(N1,NO,LMZ,LM)
  USE SPM
  BEGIN SUB
    CALL CMB(LM,0,0,0,0,2,0,0,0)
    FOR INC1 = 11 TO IN1

```

Source Control Programs

```

CALL CMB(L2,0,0,0,0,2,0,0,0.)
IFF INC.EQ.1]
  IA(L2+2)=LM
  IA(LM+5)=L2
ELSE
  IA(L2P+4)=L2
  IA(L2+3)=L2P
FIN
L2P=L2
CALL CRMD1(NO,LMZ,L1,
  IA(L2+5)=L1
  IA(L1+2)=L2
FIN
CALL UMD(LM)
RETURN
END
C DESTRO(IPVT)
C SUBROUTINE DESTROYS THE FUNCTION AT LB USING PARAMETER AND VARIABLE
C TYPE CODES IN IPVT.
SUBROUTINE DESTRO(IPVT)
USE SPM
DIMENSION IPVT(1)
BEGINSUB
FOR [I] = [LP] TO [LI-1]
  K=I-LP+1
  IFF [(IPVT(K).EQ.2.OR.IPVT(K).EQ.4).AND.1A(I).GT.0]
    CALL FSTORE(IA(I))
    IFF [I.GE.LV.AND.1A(I+1A(LB+4)).GT.0]
      CALL FSTORE(IA(I+1A(LB+4)))
    FIN
  FIN
FIN
IA(LB)=0
CALL FSTORE(LB)
IA(LCS(LBPT,MTO+1))=0
RETURN
END
C DMPSYS
C SUBROUTINE DUMPS MODEL SYSTEM.
SUBROUTINE DMPSYS
USE SPM
BEGINSUB
WR [ ] [1X,6HCOMMON]
WR [MTO] [10X,3HMT0,112]
WR [MFROM] [8X,5HMF0RM,112]
WR [ICTYPE] [7X,6HICTY'E,112]
WR [POINT] [7X,6H1POINT,112]
WR [DUMP] [8X,5HDUMP,112]
WR [ISECA] [8X,5HISECA,112]
WR [TRACE] [7X,6HTRACE,112]
WR [LB] [11X,2HLB,112]
WR [LBCA] [9X,4HLBCA,112]

```

Source Control Programs

```

WR [LBLIST] [7X,6HLBLIST,112]
WR [LBPT] [9X,4HLBPT,112]
WR [LBUF] [9X,4HLBUF,112]
WR [LCSB] [9X,4HLCSB,112]
WR [LE] [11X,2HLE,112]
WR [LEC] [10X,3HLEC,112]
WR [LFFB] [9X,4HLFFB,112]
WR [LFFE] [9X,4HLLFFE,112]
WR [LI] [11X,2HLI,112]
WR [LLE] [10X,3HLLLE,112]
WR [LLFB] [9X,4HLLFB,112]
WR [LLFE] [9X,4HLLFE,112]
WR [LMS] [10X,3HLMS,112]
WR [LNE] [10X,3HLNE,112]
WR [LP] [11X,2HLP,112]
WR [LV] [11X,2HLV,112]
WR [MINNFI] [7X,6HMINNFI,112]
WR [MINSE] [9X,4HMINSE,112]
WR [INFE] [10X,3HNFE,112]
WR [INFI] [10X,3HNFI,112]
WR [NI1] [10X,3HNI1,112]
WR [NI2] [10X,3HNI2,112]
WR [NO1] [10X,3HNO1,112]
WR [NO2] [10X,3HNO2,112]
WR [INREP] [9X,4HINREP,112]
WR [INRUN] [9X,4HINRUN,112]
WR [INSE] [10X,3HNSE,112]
WR [TIME] [9X,4HTIME,E12.4]
WR [TSTOP] [9X,5HTSTOP,E12.4]
K=1A(LBLIST+3)
FOR [I] = [1] TO [K]
  MFROM=0
  MTO=1A(LCS(LBLIST,I+2))
  ICTYPE=7
  IPOINT=0
  WR [MTO] [1X,8HFUNCTION,15]
  CALL SEND
  FIN
  WR [I] [1X,11HEVENT CHAIN]
  WR [LEC] [10X,3HLEC,112]
  WR [LNE] [10X,3HLNE,112]
  WR [LLE] [10X,3HLLLE,112]
  WR [LFFE] [9X,4HLLFFE,112]
  WR [LLFE] [9X,4HLLFE,112]
  WR [INSE] [10X,3HNSE,112]
  WR [MINSE] [9X,4HMINSE,112]
  WR [INFE] [10X,3HNFE,112]
  WR [ISECA] [9X,5HISECA,112]
  L=LNE
  WHILE [L.GT.0]
    WR [L,1A(L),RA(L+1),1A(L+2),1A(L+3),1A(L+4)] &
      [1X,2I10,5X,E10.4,3I10]

```

Source Control Programs

```

      L=IA(L)
      FIN
      WR ( ) [1X,10HFREE STORE]
      WR [LFFB] [9X,4HLFFB,112]
      WR [LFB] [9X,4HLFB,112]
      WR [NFL] [10X,3HNFL,112]
      WR [LBCA] [9X,4HLBCA,112]
      WR [MINNFL] [7X,6HMINNFL,112]
      L=LFFB
      WHILE [L.GT.0]
        WR [L,IA(L),IA(L+1)] [1X,3:10]
        L=IA(L)
      FIN
      RETURN
      END
C DUMP(IPVT)
C SUBROUTINE DUMPS THE MODEL FUNCTION AT LB.
      SUBROUTINE DUMP(IPVT)
      USE SPM
      BEGIN SUB
      CALL PFLB(IPVT)
      CALL PVLB(IPVT)
      CALL PICLB(IPVT)
      RETURN
      END
C ERROR
C SUBROUTINE PRINTS DIAGNOSTIC DATA AND STOPS PROGRAM.
      SUBROUTINE ERROR
      USE SPM
      BEGIN SUB
      CALL DMPSYS
      CALL EXIT
      RETURN
      END
C EVDRS(N,M,V)
C SUBROUTINE RETURNS SAMPLE VALUE V FROM A POPULATION
C WITH DISTRIBUTION TYPE IN INTEGER PARAMETER N AND STATISTICS IN
C REAL LIST PARAMETER M OF THE CALLING PROGRAM.
      SUBROUTINE EVDRS(N,M,V)
      USE SPM
      BEGIN SUB
      TR (5) [EVDRS(N,M, )] [N,M] [1X,2110]
      CASE [IP(N).EQ.1]
        V=RUNIF(RLP(M,1),RLP(M,2))
      ORIF [IP(N).EQ.2]
        V=RNORM(RLP(M,1),RLP(M,2))
      ORIF [IP(N).EQ.3]
        V=RNORMT(RLP(M,1),RLP(M,2),RLP(M,3),RLP(M,4))
      ORIF [IP(N).EQ.4]
        CALL URN(IX,U)
        V=-RLP(M,1) ALOG(U)

```

Source Control Programs

```

      OTHERWISE
      ERROR [EVORS ILLEGAL DISTRIBUTION TYPE]
      FIN
      TR [5] [EVDP5( , ,V)] [V] [1X,E12.4]
      RETURN
    END
C FPNTRS(NFUN)
C SUBROUTINE SETS POINTERS LB, LP, L, LI, AND LE FOR FUNCTION NFUN.
SUBROUTINE FPNTRS(NFUN)
  USE SPM
  BEGIN SUB
  LB=IA(LCS(LBPT,NFUN+1))
  NP=IA(LB+3)
  NV=IA(LB+4)
  LP=LB+5
  LV=LP+NP
  LI=LV+NV
  LE=LI+NV-1
  RETURN
END
C FRMD(LM)
C SUBROUTINE FREES MATERIAL DATA AT LM.
SUBROUTINE FRMD(LM)
  USE SPM
  BEGIN SUB
  TR [5] [FRMD(LM)] [LM] [1X,110]
  IFF [LM,LE.0]
    WR [LM] [11X,14HNO MATERIAL AT,16]
  ELSE
    MODE=1
    L=LM
    WHILE [L,NE,LM,OR,MODE,NE.4]
      PERFORM STEP [1]
    FIN
    CALL FSTORE(LM)
    TR [5] [FRMD DONE]
  FIN
  RETURN
C
  PROCEDURE STEP [1]
  CASE [MODE,EQ.1]
    CASE [IA(L+5).GT.0]
      L=IA(L+5)
    ORIF [IA(L+4).GT.0]
      MODE=2
      L=IA(L+4)
    OTHERWISE
      MODE=3
  FIN
  ORIF [MODE,EQ.2]
    CASE [IA(L+5).GT.0]
      MODE=1

```

Source Control Programs

```

      L=IA(L+5)
      ORIF [(IA(L+4).GT.0)
      L=IA(L+4)
      OTHERWISE
      MODE=3
      FIN
      ORIF [MODE.EQ.3]
      IFF [L.NE.LM]
      LF=L
      ELSE
      LF=C
      FIN
      CASE [(IA(L+2).GT.0)
      MODE=4
      L=IA(L+2)
      ORIF [(IA(L+3).GT.0)
      L=IA(L+3)
      OTHERWISE
      IFF [L.EQ.LM]
      MODE=4
      ELSE
      PERFORM ERROR [2]
      FIN
      FIN
      IFF [LF.GT.0]
      CALL FSTORE(LF)
      FIN
      ORIF [MODE.EQ.4]
      CASE [(IA(L+4).GT.0)
      LF=0
      MODE=2
      L=IA(L+4)
      ORIF [(IA(L+3).GT.0)
      LF=L
      MODE=3
      L=IA(L+3)
      ORIF [(IA(L+2).GT.0)
      LF=L
      L=IA(L+2)
      OTHERWISE
      PERFORM ERROR [2]
      FIN
      IFF [LF.GT.0]
      CALL FSTORE(LF)
      FIN
      OTHERWISE
      PERFORM ERROR [2]
      FIN
      ENDPROC [1]
C
PROCEDURE ERROR [2]
ERROR [MATERIAL DATA STRUCTURE ERROR (FRMD)]

```


Source Control Programs

```

ENDPROC [2]
END
C FSTORE(LOC)
C SUBROUTINE FREES THE STORAGE SPACE AT LOC.
SUBROUTINE FSTORE(LOC)
USE SPM
BEGINSUB
TR [5] [FSTORE(LOC)] [LOC] [1X,110]
L=LOC
N=TAIL+1
WHILE [IA(L).GT.0]
L=IA(L)
N=N+IA(L+1)
FIN
IA(L)=LFFB
LFFB=LOC
NFL=NFL+N
TR [5] [FSTORE DONE]
RETURN
END
C IAC(L,I)
C FUNCTION RETURNS INTEGER VALUE OF ITEM I IN CHAIN STORAGE AT L.
FUNCTION IAC(L,I)
USE SPM
BEGINFUN
IAC=IA(LCS(L,I))
RETURN
END
C ICH(ICHAR)
C FUNCTION RETURNS AN INTEGER VALUE CORRESPONDING TO THE POSITION
C OF THE ALPHABETIC CHARACTER ICHAR IN THE STRING: 0 1 ... 9 [BLANK]
C A B ... Z $ % ' / = ?
C RETURNS -1 IF ICHAR IS NOT ONE OF THE 49 CHARACTERS IN THE LIST.
FUNCTION ICH(ICHAR)
USE SPM
BEGINFUN
DIMENSION I(49)
DATA I/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,
1H ,1HA,1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI,
1HJ,1HK,1HL,1HM,1HN,1HO,1HP,1HQ,1HR,1HS,
1HT,1HU,1HV,1HW,1HX,1HY,1HZ,1HS,1H*,1H-,
1H/,1H=,1H.,1H),1H.,1H.,1H(,1H',1H&/
K=1
WHILE [K.LE.49.AND.ICHAR.NE.I(K)]
K=K+1
FIN
IF [K.GT.49]
ICH=-1
ELSE
ICH=K-1
FIN
RETURN

```

Source Control Programs

```

      END
C ICHECK
C SUBROUTINE RESPONDS TO THE FOLLOWING INTERACTIVE INPUTS:
C ABORT
C EVENTS(N)
C FREE
C IA(N)
C IA(N,M)
C IAC(L,N)
C IAC(L,N,M)
C ICTYPE
C IDUMP
C IPOINT
C ISECA
C JTRACE
C LB
C LBCA
C LBLIST
C LBPI
C LBUF
C LCSB
C LE
C LEC
C LFFB
C LFFE
C LI
C LLE
C LLFB
C LLFE
C LMS
C LNE
C LP
C LV
C MATL(L)
C MFROM
C MINNFL
C MNSE
C MSG
C MTO
C NFE
C NFL
C NI1
C NI2
C NO1
C NO2
C NREP
C NRUN
C NSE
C RA(N)
C RA(N,M)
C RAC(L,N)
C RAC(L,N,M)

```

Source Control Programs

```

C TIME
C TSTOP
SUBROUTINE ICHECK
USE SFM
BEGINSUB
DIMENSION IPVT(1)
L1=LBUF*2
L80=LBUF*81
NI1S=NI1
NO1S=NO1
NI1=NI2
NO1=NO2
PERFORM ACTION (1)
NI1=NI1S
NO1=NO1S
RETURN
C
PROCEDURE ACTION (1)
RD ((IA(1),I=L1,L80)) (80A1)
WHILE ((IA(L1) NE 1H) )
CASE ((IA(L1).EQ.1HA)
NO1=NO1S
ERROR (USER ABOPT)
ORIF ((IA(L1).EQ.1HE)
PERFORM EVENTS (23)
ORIF ((IA(L1).EQ.1HF)
PERFORM FREE (24)
ORIF ((IA(L1).EQ.1HJ)
PERFORM I (2)
ORIF ((IA(L1).EQ.1HL)
PERFORM L (3)
ORIF ((IA(L1).EQ.1HM)
PERFORM M (4)
ORIF ((IA(L1).EQ.1HN)
PERFORM N (5)
ORIF ((IA(L1).EQ.1HR)
PERFORM R (6)
ORIF ((IA(L1).EQ.1HT)
PERFORM T (7)
OTHERWISE
PERFORM PROMPT (8)
FIN
RD ((IA(1),I=L1,L80)) (80A1)
FIN
ENDPROC (1)
C
PROCEDURE : (2)
CASE ((IA(L1+1).EQ.1HA)
PERFORM IA (9)
ORIF ((IA(L1+1).EQ.1HC)
WR (ICTYPE) (IX,6HICTYPE,12)
ORIF ((IA(L1+1).EQ.1HD)

```

Source Control Programs

```

      WR [IDUMP] [IX,5HIDUMP,12]
      ORIF [IA(L1+1).EQ.1HP]
      WR [IPQNT] [IX,6HIPQ,NT,16]
      ORIF [IA(L1+1).EQ.1HS]
      WR [ISECA] [IX,5HISECA,16]
      ORIF [IA(L1+1).EQ.1HT]
      WR [ITRACE] [IX,6HITRACE,12]
      OTHERWISE
      PERFORM PROMPT [8]
      FIN
      ENDPROC [2]
C
      PROCEDURE L [3]
      CASE [IA(L1+1).EQ.1HB]
      PERFORM LB [10]
      ORIF [IA(L1+1).EQ.1HC]
      WR [LCSS] [IX,4HLCSS,16]
      ORIF [IA(L1+1).EQ.1HE]
      PERFORM LE [11]
      ORIF [IA(L1+1).EQ.1HF]
      PERFORM LF [12]
      ORIF [IA(L1+1).EQ.1HI]
      WR [LI] [IX,2HLI,16]
      ORIF [IA(L1+1).EQ.1HL]
      PERFORM LL [13]
      ORIF [IA(L1+1).EQ.1HN]
      WR [LNE] [IX,3HLNE,16]
      ORIF [IA(L1+1).EQ.1HM]
      WR [LMS] [IX,3HLMs,16]
      ORIF [IA(L1+1).EQ.1HP]
      WR [LP] [IX,2HLP,16]
      ORIF [IA(L1+1).EQ.1HV]
      WR [LV] [IX,2HLV,16]
      OTHERWISE
      PERFORM PROMPT [8]
      FIN
      ENDPROC [3]
C
      PROCEDURE M [4]
      CASE [IA(L1+1).EQ.1HA]
      CASE [IA(L1+2).NE.1HT]
      PERFORM PROMPT [8]
      ORIF [IA(L1+3).NE.1HL]
      PERFORM PROMPT [8]
      ORIF [IA(L1+4).NE.1H]
      PERFORM PROMPT [8]
      FIN
      LSCAN=L1+5
      PERFORM SCANLNM [22]
      CALL WRMD(LNM)
      ORIF [IA(L1+1).EQ.1HF]
      WR [MFROM] [IX,5HMFROM,13]

```

Source Control Programs

```

ORIF [IA(L1+1).EQ.1H1]
  WR [MINNFI] [IX,6HMINNFI,16]
OR:F [IA(L1+1).EQ.1H1]
  WR [MNSE] [IX,4HMNSE,14]
ORIF [IA(L1+1).EQ.1HS]
  PERFORM MESSAGE [25]
ORIF [IA(L1+1).EQ.1HT]
  WR [MTO] [IX,3HMT0,13]
OTHERWISE
  PERFORM PROMPT [8]
FIN
ENDPROC [4]
C
PROCEDURE N [5]
CASE [IA(L1+1).EQ.1HF]
  PERFORM NF [14]
ORIF [IA(L1+1).EQ.1H1]
  CASE [IA(L1+2).EQ.1H1]
    WR [N1S] [IX,3HN1,15]
  ORIF [IA(L1+2).EQ.1H2]
    WR [N12] [IX,3HN12,15]
  OTHERWISE
    PERFORM PROMPT [8]
  FIN
ORIF [IA(L1+1).EQ.1H6]
  CASE [IA(L1+2).EQ.1H1]
    WR [N01S] [IX,3HN01,15]
  ORIF [IA(L1+2).EQ.1H2]
    WR [N02] [IX,3HN02,15]
  OTHERWISE
    PERFORM PROMPT [8]
  FIN
ORIF [IA(L1+1).EQ.1HR]
  PERFORM NR [15]
ORIF [IA(L1+1).EQ.1HS]
  WR [NSE] [IX,3HNSE,14]
OTHERWISE
  PERFORM PROMPT [8]
FIN
ENDPROC [5]
C
PROCEDURE R [6]
CASE [IA(L1+1).EQ.1HA.AND. IA(L1+2).EQ.1H()]
  PERFORM RA [16]
ORIF [IA(L1+1).EQ.1HA.AND. IA(L1+2).EQ.1HC]
  PERFORM RAC [17]
OTHERWISE
  PERFORM PROMPT [8]
FIN
ENDPROC [6]
C
PROCEDURE T [7]

```

Source Control Programs

```

CASE [IA(L1+1).EQ.1HI]
  WR [TIME] [IX,4HTIME,E12.4]
ORIF [IA(L1+1).EQ.1HS]
  WR [TSTOP] [IX,5HTSTOP,E12.4]
OTHERWISE
  PERFORM PROMPT [8]
  FIN
ENDPROC [7]
C
PROCEDURE PROMPT [8]
WR [ ] [IX,5HWHAT?]
ENDPROC [8]
C
PROCEDURE IA [9]
CASE [IA(L1+2).EQ.1HI]
  PERFORM IA [10]
ORIF [IA(L1+2).EQ.1HC]
  PERFORM IAC [19]
OTHERWISE
  PERFORM PROMPT [8]
  FIN
ENDPROC [9]
C
PROCEDURE LB [10]
CASE [IA(L1+2).EQ.1H ]
  WR [LB] [IX,2HLB,16]
ORIF [IA(L1+2).EQ.1HC]
  WR [LBCA] [IX,4HLBCA,16]
ORIF [IA(L1+2).EQ.1HL]
  WR [LBLIST] [IX,6HLBLIST,16]
ORIF [IA(L1+2).EQ.1HP]
  WR [LBPT] [IX,4HLBPT,16]
ORIF [IA(L1+2).EQ.1HU]
  WR [LBUF] [IX,4HLBUF,16]
OTHERWISE
  PERFORM PROMPT [8]
  FIN
ENDPROC [10]
C
PROCEDURE LE [11]
CASE [IA(L1+2).EQ.1H ]
  WR [LE] [IX,2HLE,16]
ORIF [IA(L1+2).EQ.1HC]
  WR [LEC] [IX,3HLEC,16]
OTHERWISE
  PERFORM PROMPT [8]
  FIN
ENDPROC [11]
C
PROCEDURE LF [12]
CASE [IA(L1+2).EQ.1HF.AND.[IA(L1+3).EQ.1HB]
  WR [LFFB] [IX,4HLFFB,16]

```

Source Control Programs

```

ORIF [IA(L1+2).EQ.1HF.AND.[A(L1+3).EQ.1HE]
  WR [LLFE] [1X,4HLLFE,16]
OTHERWISE
  PERFORM PROMPT [6]
  FIN
ENDPROC [12]

C
PROCEDURE LL [13]
CASE [IA(L1+2).EQ.1HE]
  WR [LLF] [1X,3HLLF,16]
ORIF [IA(L1+2).EQ.1HF.AND.[A(L1+3).EQ.1+3]
  WR [LLFB] [1X,4HLLFB,16]
ORIF [IA(L1+2).EQ.1HF.AND.[A(L1+3).EQ.1HE]
  WR [LLFE] [1X,4HLLFE,16]
OTHERWISE
  PERFORM PROMPT [6]
  FIN
ENDPROC [13]

C
PROCEDURE NF [14]
CASE [IA(L1+2).EQ.1HE]
  WR [NFE] [1X,3HNFE,14]
ORIF [IA(L1+2).EQ.1HL]
  WR [NFL] [1X,3HNFL,16]
OTHERWISE
  PERFORM PROMPT [6]
  FIN
ENDPROC [14]

C
PROCEDURE NR [15]
CASE [IA(L1+2).EQ.1HE]
  WR [NREP] [1X,4HNREP,14]
ORIF [IA(L1+2).EQ.1HU]
  WR [NRUN] [1X,4HNRUN,14]
OTHERWISE
  PERFORM PROMPT [6]
  FIN
ENDPROC [15]

C
PROCEDURE RA [16]
KIR=2
PERFORM 1/RA [20]
ENDPROC [16]

C
PROCEDURE RAC [17]
KIR=2
PERFORM 1/RAC [21]
ENDPROC [17]

C
PROCEDURE IA [18]
KIR=1
PERFORM 1/RA [20]

```

Source Control Programs

```

C      ENDPROC [18]

      PROCEDURE IAC [19]
      IIR=1
      PERFORM I/RAC [21]
      ENDPROC [19]

C      PROCEDURE I/RAC [20]
      LSCAN=L1+3
      PERFORM SCANLNM [22]
      N=LNM
      LSCAN=LSCAN+1
      CASE [IA(LSCAN).EO.1H)]
        CASE [KIR.EO.1]
          WR [N,IA(N)] [1X,3HIA(,14,1H),110]
          ORIF [KIR.EO.2]
            WR [N,RA(N)] [1X,3HRA(,14,1H),E12.4]
          FIN
        ORIF [IA(LSCAN).EO.1H,]
          LSCAN=LSCAN+1
          PERFORM SCANLNM [22]
          M=LNM
          CASE [KIR.EO.1]
            WR [N,M,[IA(1),I=N,M]] &
              [1X,3HIA(,14,1H,,14,1H),5110/(14X,5110)]
            ORIF [KIR.EO.2]
              WR [N,M,[RA(1),I=N,M]] &
                [1X,3HRA(,14,1H,,14,1H),5E12.4/(14X,5E12.4)]
            FIN
          OTHERWISE
            PERFORM PROMPT [8]
          FIN
      ENDPROC [20]

C      PROCEDURE I/RAC [21]
      IFF [IA(L1+3).NE.1H()]
        PERFORM PROMPT [8]
      ELSE
        LSCAN=L1+4
        PERFORM SCANLNM [22]
        L=LNM
        LSCAN=LSCAN+1
        IFF [IA(LSCAN).NE.1H,]
          PERFORM PROMPT [8]
        ELSE
          LSCAN=LSCAN+1
          PERFORM SCANLNM [22]
          N=LNM
          LSCAN=LSCAN+1
          CASE [IA(LSCAN).EO.1H)]
            CASE [KIR.EO.1]
              WR [L,N,IAC(L,N)] [1X,4H1AC(,14,1H,,14,1H),110]

```


Source Control Programs

```

ORIF [KIR.EQ.2]
  WR [L,N,RAC(L,N)] [1X,4HRAC(,14,1H,,14,1H),E12.4]
  FIN
ORIF [IA(LSCAN).EQ.1H,]
  LSCAN=LSCAN+1
  PERFORM SCANLNM [22]
  M=LNM
  CASE [KIR.EQ.1]
    WR [L,N,M,(IAC(L,1),1=N,M)] &
      [1X,4HIAC(,14,1H,,14,1H,,14,1H),5I10/(14X,5I10)]
  ORIF [KIR.EQ.2]
    WR [L,N,M,(RAC(L,1),1=N,M)] &
      [1X,4HRAC(,14,1H,,14,1H,,14,1H),5E12.4/(14X,5E12.4)]
  FIN
  OTHERWISE
    PERFORM PROMPT [8]
  FIN
  FIN
  FIN
ENDPROC [21]
C
PROCEDURE SCANLNM [22]
  LNM=0
  NCH=[CH([A(LSCAN)])]
  WHILE [NCH.GE.0.AND.NCH.LE.9.AND.LSCAN.LE.L80]
    LNM=10 LNM+NCH
    LSCAN=LSCAN+1
    NCH=[CH([A(LSCAN)])]
  FIN
  LSCAN=LSCAN-1
ENDPROC [22]
C
PROCEDURE EVENTS [23]
  IFF [IA(L1+1).NE.1HV.OR.1A(L1+2).NE.1HE.OR.1A(L1+3).NE.1HN8
    .OR.1A(L1+4).NE.1HT.OR.1A(L1+5).NE.1HS.OR.1A(L1+6).NE.1H(]
    PERFORM PROMPT [8]
  FIN
  LSCAN=L1+7
  PERFORM SCANLNM [22]
  WR [LNM] [1X,4HNEXT,15,7H EVENTS]
  N=0
  L=LNE
  WHILE [N.LE.LNM.AND.L.GT.0]
    WR [L,IA(L),RA(L+1),IA(L+2),1A(L+3),1A(L+4)]&
      [1X,2I10,5X,E10.4,3I10]
    N=N+1
    L=IA(L)
  FIN
ENDPROC [23]
C
PROCEDURE FREE [24]
  WR [ ] [1X,10HFREE STORE]

```

Source Control Programs

```

L=LFFB
WHILE [L.GT.0]
  WR IL,IA(L),IA(L+1) [1X,3110]
  L=IA(L)
FIN
ENDPROC [24]
C
PROCEDURE MESSAGE [25]
  IFF [ICTYPE.EQ.8]
    WR I:IA(IPPOINT),IA(IPPOINT+1):16
    [1X,10HEVENT CODE,13,2X,7HDATA AT,16]
  ELSE
    WR IMFROM,MTO,ICTYPE,IPPOINT:8
    [1X,5HMFROM,16,2X,3HMTO,16,2X,6HICTYPE,13,2X,6HIPPOINT,16]
    IFF [ICTYPE.EQ.9.AND.IPPOINT.GT.0]
      WR I:IA(IPPOINT+1),1=2,9: [1X,7HMESSAGE,816]
    FIN
  FIN
ENDPROC [25]
END
C IIC(I)
C FUNCTION RETURNS THE ITH INTEGER INITIAL CONDITION OF THE FUNCTION AT LB.
  FUNCTION IIC(I)
    USE SPM
    BEGINFUN
    IIC=IA(L)+1,
    RETURN
  END
C ILIC(I,J)
C FUNCTION RETURNS THE JTH ELEMENT OF THE INTEGER LIST INITIAL
C CONDITION I OF THE FUNCTION AT LB
  FUNCTION ILIC(I,J)
    USE SPM
    BEGINFUN
    ILIC=IA(LCS(IA(L)+1),J+2)
    RETURN
  END
C ILP(I,J)
C FUNCTION RETURNS THE JTH ELEMENT OF THE INTEGER LIST PARAMETER I OF
C THE FUNCTION AT LB.
  FUNCTION ILP(I,J)
    USE SPM
    BEGINFUN
    ILP=IA(LCS(IA(LP+1),J+2))
    RETURN
  END
C ILV(I,J)
C FUNCTION RETURNS THE JTH ELEMENT OF THE INTEGER LIST VARIABLE I OF
C THE FUNCTION AT LB.
  FUNCTION ILV(I,J)
    USE SPM
    BEGINFUN

```

Source Control Programs

```

      ILV=IA(LCS(IA(LV+1),J+2))
      RETURN
    END
C INCREC(L1)
C SUBROUTINE INCREMENTS EVENT CHAIN.
  SUBROUTINE INCREC(L1)
    USE SPM
    BEGINSUB
    WARN [INCREC INCREMENTING EVENT CHAIN]
    LAST=L1
    WHILE [IA(LAST).NE.0]
      LAST=IA(LAST)
    FIN
    CALL ASTORE(L2,ISECA)
    IA(LAST)=L2
    K=IA(L2+2)/5
    NFE=NFE+K
    LFE=L2+2
    FOR [I1 = 1] TO [K]
      L=LCS(L2,(I1-1)*5+1)
      IA(L)=LCS(L2,I1*5+1)
      IF(I.EQ.K)IA(L)=0
    FIN
    RETURN
  END
C IP(I)
C FUNCTION RETURNS THE ITH INTEGER PARAMETER OF THE FUNCTION AT LB.
  FUNCTION IP(I)
    USE SPM
    BEGINFUN
    IP=IA(LP+I)
    RETURN
  END
C UNIF(M,ISP)
C FUNCTION RETURNS INTEGER, RANDOM VARIABLE WITH UNIFORM [0,3] DISTRIBUTION.
  FUNCTION UNIF(M,ISP)
    USE SPM
    DATA IX,U/1,0./
    CALL URN(IX,U)
    UNIF=M-ISP+IFIX(U*FLOAT(2*ISP+1))
    RETURN
  END
C IV(I)
C FUNCTION RETURNS THE ITH INTEGER VARIABLE OF THE FUNCTION AT LB.
  FUNCTION IV(I)
    USE SPM
    BEGINFUN
    IV=IA(LV+I)
    RETURN
  END
C KEYWD(L)

```

Source Control Programs

```

C FUNCTION COMPARES THE A1 FORMAT CHARACTERS 1A(L), 1A(L+1), ...
C FOR A MATCH WITH KEYWORDS AND RETURNS THE VALUES BELOW:
C NO MATCH 0
C CREATE 1
C DESTROY 2
C PARAMETER 3
C VARIABLE 4
C REPORT 5
C RUN 6
C END 7

      FUNCTION KEYWD(L)
      USE SPM
      BEGINFUN
      DIMENSION KWDS(10,7)
      DATA KWDS/1HC,1HR,1HE,1HA,1HT,1HE,3 1H ,6,
11HD,1HE,1HS,1HT,1HR,1HO,1HY,2 1H ,7,
11HP,1HA,1HR,1HA,1HM,1HE,1HT,1HE,1HR,9,
11HV,1HA,1HR,1HI,1HA,1HB,1HL,1HE,1H ,8,
11HR,1HE,1HP,1HO,1HR,1HT,3 1H ,6,
11HR,1HU,1HN,6 1H ,3,
11HE,1HN,1HD,6 1H ,3/
      K=0
      J=1
      WHILE (J.LE.7.AND.K.EQ.0)
        K=J
        I=1
        WHILE (I.LE.KWDS(10,J))
          IFF (1A(L+I).NE.KWDS(I,J))
            K=0
            FIN
            I=I+1
          FIN
        J=J+1
      FIN
      KEYWD=K
      RETURN
      END

C LCS(LOC,ITH)
C FUNCTION LOCATES ITH ELEMENT IN CHAIN-BLOCK STORAGE AT LOC.
      FUNCTION LCS(LOC,ITH)
      USE SPM
      BEGINFUN
      LCSB=LOC
      N=0
      WHILE (N+1A(LCSB+1).LT.ITH.AND.LCSB.NE.0)
        N=N+1A(LCSB+1)
        LCSB=1A(LCSB)
      FIN
      IFF (LCSB.LE.0)
        LCS=0
      ELSE
        LCS=LCSB+1+ITH-N

```

Source Control Programs

```

      FIN
      RETURN
    END
C LIJNM(N,M,I)
C FUNCTION RETURNS THE MCS LIST INDEX OF THE ITH ITEM IN THE
C NTH SET OF M ITEMS
      FUNCTION LIJNM(N,M,I)
      USE SPM
      BEGINFUN
      LIJNM=(N-1) M+1+2
      RETURN
    END
C MAIN 4/4/78
C PROGRAM CONTROLS MCSS EXECUTION.
      PROGRAM MAIN(MCSSIN,MCSSOUT,TAPE5=MCSSIN,TAPE6=MCSSOUT)
      USE SPM
      DIMENSION IA(20000)
      COMMON RA(20000)
      EQUIVALENCE ((IA(1),RA(1)))
      SYSTEM
      CALL CHANGE(5H+MCSS)

C
C      ASSIGN I/O DEVICE NUMBERS, TIME, AND ITRACE.
      NI1=5
      NI2=59
      NO1=6
      NO2=59
      TIME=0.
      ITRACE=0

C
C      INITIALIZE DYNAMIC STORAGE
      LFFB=1
      IA(LFFB)=0
      IA(LFFB+1)=19998
      LLFB=1
      NFL=19998
      MINNFL=NFL
      LBCA=1

C
C      CREATE BLOCK POINTER TABLE
      CALL ASTORE(LBPT,101)
      IA(LBPT+2)=IA(LBPT+2)-1
      CALL ZTABLE(LBPT)

C
C      CREATE MESSAGE STACK
      CALL ASTORE(LMS,42)
      IA(LMS+2)=IA(LMS+2)-2
      IA(LMS+3)=0

C
C      CREATE EVENT CHAIN
      CALL ASTORE(LEC,500)
      ISECA=100

```

Source Control Programs

```

C
C  CREATE BUFFER
C  CALL ASTORE(LBUF,80)
C
C  ASSIGN RANDOM NUMBER SEED.
C  U=0.
C  CALL URN(-1,U)
C
C  REQUEST USER INPUT LINE DISPLAY OPTION
C  IDSPL=-1
C  WHILE [IDSPL.LT.0.OR.IDSPL.GT.3]
C    WR [U:NO2] [ ] [1X,13HINPUT DISPLAY]
C    WR [U:NO2] [ ] $
C    /IX,12H0 = NO LINES &
C    /IX,13H1 = RUN LINES &
C    /IX,17H2 = COMMAND LINES &
C    /IX,16H3 = ALL INPUT LINES]
C    RD [U:N12] [IDSPL] [1]
C    FIN
C
C  PRINT HEADING AND USER COMMENTS DOWN TO FIRST KEYWORD.
C  WR [ ] &
C  [1H1,56HMATERIAL CONTROL SYSTEM SIMULATOR  VERSION 1.0  1/1/78]
C  PERFORM PRINT-COMMENTS [1]
C
C  EXECUTE JOB STEPS
C  WHILE [KEYWD(LBUF+2).NE.7]
C    WR [([A(LBUF+1+1),I=1,80)] [1X,80A1]
C    IFF [IDSPL.GT.0]
C      IFF [IDSPL.EQ.1]
C        IFF [KEYWD(LBUF+2).EQ.6]
C          WR [U:NO2] [([A(LBUF+1+1),I=1,80)] [1X,80A1]
C          FIN
C        ELSE
C          WR [U:NO2] [([A(LBUF+1+1),I=1,80)] [1X,80A1]
C          FIN
C      FIN
C    PERFORM STEP [2]
C    PERFORM PRINT-COMMENTS [1]
C    FIN
C
C  PRINT MINIMUM NUMBER OF FREE LOCATIONS
C  WR [MINNPL] &
C  [1X,35HMINIMUM NUMBER OF FREE LOCATIONS = ,15]
C
C  CALL EXIT
C
C  PROCEDURE PRINT-COMMENTS [1]
C  RDE [([A(LBUF+1+1),I=1,80)] [80A1]
C  WHILE [KEYWD(LBUF+2).EQ.0]
C    WR [([A(LBUF+1+1),I=1,80)] [1X,80A1]
C    IFF [IDSPL.EQ.3]

```

Source Control Programs

```

      WR 'U:NO2' [(1A(LBUF+1)),1=1,80] [(1X,80A)]
      FIN
      RDE [(1A(LBUF+1)),1=1,80] [80A1]
      FIN
ENDPROC [1]
RDEND
ERROR [MAIN NO END COMMAND IN INPUT FILE]

C
PROCEDURE STEP [2]
K=KEYWD(LBUF+2)
CASE [K.EQ.1]
  CALL FBLANK(0,1A,INTD,REALD)
  CALL GETV(0,1A,MTO,REAL)
  CALL GETV(0,1A,IPOINT,REAL)
  IFF [MTO.LE.0.OR.MTO.GT.100]
    ERROR [MAIN ILLEGAL FUNCTION NUMBER]
  FIN
  IFF [(1A(LCS(LBPT,MTO+1))).NE.0]
    ERROR &
    [MAIN ATTEMPT TO CREATE A FUNCTION THAT ALREADY EXISTS]
  FIN
  MFROM=0
  ICTYPE=1
  CALL SEND
ORIF [K.EQ.2]
  CALL FBLANK(0,1A,INTD,REALD)
  CALL GETV(0,1A,INT,REAL)
  IFF [(1A(LCS(LBPT,INT+1))).EQ.0]
    ERROR &
    [MAIN ATTEMPT TO DESTROY A FUNCTION THAT DOES NOT EXIST]
  ELSE
    MFROM=0
    MTO=INT
    ICTYPE=2
    IPOINT=0
    CALL SEND
  FIN
ORIF [K.GE.3.AND.K.LE.4]
  CALL FBLANK(0,1A,INTD,REALD)
  CALL GETV(0,1A,INT,REAL)
  MFROM=0
  MTO=INT
  ICTYPE=K
  IPOINT=IDSPL
  CALL SEND
ORIF [K.EQ.5]
  CALL FBLANK(0,1A,INTD,REALD)
  MFROM=0
  CALL GETV(0,1A,MTO,REAL)
  ICTYPE=5
  CALL SEND
ORIF [K.EQ.6]

```

Source Control Programs

```

CALL FBLANK(0,1A,INTD,REALD)
CALL GETV(0,1A,NREP,REAL)
CALL GETV(0,1A,INT,TSTOP)
CALL GETV(0,1A,IDUMP,REAL)
CALL GETV(0,1A,ITRACE,REAL)
CALL GETV(0,1A,IOUT,REAL)
PERFORM MAKE-BLOCK-LIST [3]
NRUN=1
WHILE (NRUN.LE.NREP)
  TIME=0
  PERFORM INITIALIZE-EVENT-CHAIN [6]
  PERFORM SET-INITIAL-CONDITIONS [4]
  IFF (NRUN.EQ.1.AND.IDUMP.EQ.1)
    CALL DMPSYS
  FIN
  CASE (NREP.GT.1.AND.NRUN.EQ.1)
    PERFORM SAVE-FREE-STORE [7]
  ORIF (NRUN.GT.1)
    PERFORM RESTORE-FREE-STORE [8]
  FIN
  CALL CLOCK
  NRUN=NRUN+1
  FIN
  IFF (IOUT.EQ.1)
    PERFORM STANDARD-OUTPUT [5]
  FIN
  CALL FSTORE(LBLIST)
  LBLIST=0
  OTHERWISE
    ERROR (MAIN ILLEGAL KEYWORD)
  FIN
ENDPROC [2]
C
PROCEDURE MAKE-BLOCK-LIST [3]
N=0
K=(IA(LBPT+2)
FOR [1] = [1] TO [K]
  IF (IA(LCS(LBPT,[1])).GT.0) N=N+1
  FIN
CALL ASTORE(LBLIST,N+2)
IA(LBLIST+2)=IA(LBLIST+2)-2
M=0
FOR [1] = [1] TO [K]
  IF (IA(LCS(LBPT,[1])).GT.0)
    M=M+1
    IA(LCS(LBLIST,M+2))=1
  FIN
  FIN
IA(LBLIST+3)=M
ENDPROC [3]
C
PROCEDURE SET-INITIAL-CONDITIONS [4]

```


Source Control Programs

```

K=IA(LBLIST+3)
MFROM=0
IPOINT=0
FOR [I] = [1] TO [K]
  MTO=IA(LCS(LBLIST,I+2))
  ICTYPE=6
  CALL SEND
FIN
ENDPROC [4]

C
C
PROCEDURE STANDARD-OUTPUT [5]
MFROM=0
IPOINT=0
K=IA(LBLIST+3)
WR [ ] [//34H FUNCTION TYPE ENTRIES FRACTION]
N=0
FOR [I] = [1] TO [K]
  M=IA(LCS(LBLIST,I+2))
  CALL FPNTRS(M)
  N=N+IA(LV)
FIN
FOR [I] = [1] TO [K]
  M=IA(LCS(LBLIST,I+2))
  CALL FPNTRS(M)
  B=FLOAT(IA(LV))/FLOAT(N)
  WR (M,IA(LB),IA(LV),B) [1X,18,16,19,F10.4]
FIN
FOR [I] = [1] TO [K]
  MTO=IA(LCS(LBLIST,I+2))
  ICTYPE=5
  CALL SEND
FIN
ENDPROC [5]

C
PROCEDURE INITIALIZE-EVENT-CHAIN [6]
NFE=IA(LEC+1)/5
LFFE=LEC+2
LLFE=LCS(LEC,(NFE-1) 5+1)
NSE=0
MNSE=0
LNE=0
LLE=0
FOR [I] = [1] TO [NFE]
  L=LCS(LEC,(I-1) 5+1)
  IA(L)=LCS(LEC,I 5+1)
  IF(I.EQ.NFE)IA(L)=0
FIN
ENDPROC [6]

C
PROCEDURE SAVE-FREE-STORE [7]
IFF [IA(LFFB+1).EQ.NFL]

```

Source Control Programs

```

LFFBS=LFFB
LLFBS=LLFB
NFLS=NFL
MINNFS=MINNFL
ELSE
LFFBS=0
LLFBS=0
NFLS=0
MINNFS=0
FIN
ENDPROC [7]
C
PROCEDURE RESTORE-FREE-STORE [8]
IFF [LFFBS.GT.0]
LFFB=LFFBS
LLFB=LLFBS
NFL=NFLS
IFF [MINNFL.GT.MINNFS]
MINNFL=MINNFS
FIN
IA(LFFB)=0
IA(LLFB)=NFL
FIN
ENDPROC [8]
END
C MZUR(LM,LMP,N,MT,NC,A,NCT,AT)
C SUBROUTINE MEASURES MATERIAL AT LM ACCORDING TO THE NTH
C MEASUREMENT SPECIFICATION AT LMP AND RETURNS THE MEASURED
C MATERIAL TYPE MT, COUNT NC, MASS A, TRUE COUNT NCT,
C AND TRUE MASS AT.
SUBROUTINE MZUR(LM,LMP,N,MT,NC,A,NCT,AT)
USE SPH
BEGINSUB
TR [5] [MZUR(LM,LMP,N, , , , )] [LM,LMP,N] [IX,3110]
MT=RAC(LMP,LINM(N,6,1))
NC=0
NCT=0
A=0.
AT=0.
MODE=1
L=LM
WHILE [L.NE.LM.OR.MODE.NE.4]
PERFORM STEP [1]
FIN
PERFORM MEASUREMENT [3]
TR [5] [MZUR( , , ,MT,NC,A,NCT,AT)] [MT,NC,A,NCT,AT] &
[IX,2110,E12.4,110,E12.4]
RETURN
C
PROCEDURE STEP [1]
CASE [MODE.EQ.1]
CASE [IA(L+5).GT.0]

```

Source Control Programs

```

      L=IA(L+5)
      ORIF [IA(L+4).GT.0]
        MODE=2
        L=IA(L+4)
      OTHERWISE
        MODE=3
      FIN
    ORIF [MODE.EQ.2]
      CASE [IA(L+5).GT.0]
        MODE=1
        L=IA(L+5)
      ORIF [IA(L+4).GT.0]
        L=IA(L+4)
      OTHERWISE
        MODE=3
      FIN
    ORIF [MODE.EQ.3]
      IFF [IA(L+6).EQ.0.AND.1A(L+7).EQ.MT]
        PERFORM COLLECTMATERIALMT [2]
      FIN
      CASE [IA(L+2).GT.0]
        MODE=4
        L=IA(L+2)
      ORIF [IA(L+3).GT.0]
        L=IA(L+3)
      OTHERWISE
        IFF [L.EQ.LM]
          MODE=4
        ELSE
          PERFORM ERROR [3]
        FIN
      FIN
    ORIF [MODE.EQ.4]
      CASE [IA(L+4).GT.0]
        MODE=2
        L=IA(L+4)
      ORIF [IA(L+3).GT.0]
        MODE=3
        L=IA(L+3)
      ORIF [IA(L+2).GT.0]
        L=IA(L+2)
      OTHERWISE
        PERFORM ERROR [4]
      FIN
    OTHERWISE
      PERFORM ERROR [4]
    FIN
  ENDPROC [1]
C
  PROCEDURE COLLECTMATERIALMT [2]
    NCT=NCT+IA(L+8)
    AT=AT+RA(L+8)

```

Source Control Programs

```

ENDPROC [2]

C
PROCEDURE MEASUREMENT [3]
NC=NCT*(UNIF(IFIX(RAC(LMSP,L1INM(N,6,2))),
1*FIX(RAC(LMSP,L1INM(N,6,3)))
A=AT*(RAC(LMSP,L1INM(N,6,4))
1+RNORM(RAC(LMSP,L1INM(N,6,5)),RAC(LMSP,L1INM(N,6,6)))
ENDPROC [3]

C
PROCEDURE ERROR [4]
ERROR (MATERIAL DATA STRUCTURE ERROR (MZUR))
ENDPROC [4]
END

C NCBM(LOC,NC,BM)
C SUBROUTINE RETURNS THE NUMBER OF CONTAINERS NC AND THE TOTAL BULK
C MASS BM AT LOCATION LOC.
SUBROUTINE NCBM(LOC,NC,BM)
USE SPM
BEGINSUB
TR [5] (NCBM(LOC, , )) [LOC] [1X,110]
CASE [1A(LOC+6).EQ.1]
NC=0
BM=RA(LOC+9)
ORIF [1A(LOC+6).EQ.2]
L=[1A(LOC+6)]
IFF [L.LE.0]
ERROR (NCBM NO SUBDIVISION OF LOC)
FIN
CASE [1A(L+6).EQ.1]
BM=RA(L+9)
NC=0
WHILE [1A(L+4).GT.0]
NC=NC+1
L=[1A(L+4)]
FIN
ORIF [1A(L+6).EQ.2]
BM=0.
NC=1
WHILE [1A(L+4).GT.0]
NC=NC+1
L=[1A(L+4)]
FIN
OTHERWISE
ERROR (NCBM ILLEGAL SUBLOC LEVEL)
FIN
OTHERWISE
ERROR (NCBM ILLEGAL LOC LEVEL)
FIN
TR [5] (NCBM(LOC,NC,BM)) [LOC,NC,BM] [1X,2110,E11.4]
RETURN
END
C NSTATS(N)

```

Source Control Programs

```

C FUNCTION RETURNS THE NUMBER OF STATISTICS REQUIRED TO DEFINE
C DISTRIBUTION TYPE N
  FUNCTION NSTATS(N)
    USE SPM
    BEGINFUN
    DIMENSION NS(4)
    DATA NS/2,2,4,1/
    NSTATS=NS(N)
    RETURN
  END
C PCOM(INTO,NCTYPE,11,12,13,14,15,16,17,18)
C SUBROUTINE PREPARES COMMUNICATION FOR TRANSMISSION.
  SUBROUTINE PCOM(INTO,NCTYPE,11,12,13,14,15,16,17,18)
    USE SPM
    BEGINSUB
    MFROM=IA(LB+2)
    MTQ=INTO
    ICTYPE=NCTYPE
    CALL ASTORE(IPOINT,8)
    IA(IPOINT+2)=11
    IA(IPOINT+3)=12
    IA(IPOINT+4)=13
    IA(IPOINT+5)=14
    IA(IPOINT+6)=15
    IA(IPOINT+7)=16
    IA(IPOINT+8)=17
    IA(IPOINT+9)=18
    RETURN
  END
C PHLB
C SUBROUTINE PRINTS THE HEADER OF THE FUNCTION AT LB IN DUMP FORMAT.
  SUBROUTINE PHLB
    USE SPM
    BEGINSUB
    WR (LB,IA(LB)) (1X,5HLB+00,5X,15,112)
    FOR (I) = 11 TO 14
      L=LB+I
      WR (I,L,IA(L)) (3X,1H+,12,5X,15,112)
    FIN
    RETURN
  END
C PICLB(IPVT)
C SUBROUTINE PRINTS THE INITIAL CONDITION VALUES OF THE FUNCTION AT LB IN
C DUMP FORMAT
  SUBROUTINE PICLB(IPVT)
    USE SPM
    DIMENSION IPVT(1)
    BEGINSUB
    FOR (I) = 11 TO 14IA(LB+4))
      CALL PIDF(LI-1+I,IPVT(IA(LB+3)+1),1-1)
    FIN
    RETURN
  END

```

Source Control Programs

```

END
C PIDF(L,K,N)
C SUBROUTINE PRINTS THE ITEM AT L OF TYPE K IN DUMP FORMAT.
SUBROUTINE PIDF(L,K,N)
  USE SPM
  BEGIN SUB
  CASE (L.EQ.LP)
    M=3HLP*
  ORIF (L.EQ.LV)
    M=3HLV*
  ORIF (L.EQ.LI)
    M=3HLI*
  OTHERWISE
    M=3H*
  FIN
  CASE (K.EQ.1)
    WR (M,N,K,L,IA(L)) (1X,A3,12,215,112)
  ORIF (K.EQ.3)
    WR (M,N,K,L,RA(L)) (1X,A3,12,215,E12.4)
  ORIF (K.EQ.2)
    CASE (IA(L).EQ.0)
      WR (M,N,K,L,IA(L)) (1X,A3,12,215,112)
    ORIF (IA(L).GT.0)
      J=IA(L)+3
      WR (M,N,K,L,IA(L),(IA(LCS(IA(L),2+1)),I=1,J)) &
        (1X,A3,12,215,112/(28X,5)112)
    OTHERWISE
      ERROR (NEGATIVE LIST LOCATION (PIDF))
    FIN
  ORIF (K.EQ.4)
    CASE (IA(L).EQ.0)
      WR (M,N,K,L,IA(L)) (1X,A3,12,215,112)
    ORIF (IA(L).GT.0)
      J=IA(L)+3
      WR (M,N,K,L,IA(L),(RA(LCS(IA(L),2+1)),I=1,J)) &
        (1X,A3,12,215,112/(28X,5E12.4))
    OTHERWISE
      ERROR (NEGATIVE LIST LOCATION (PIDF))
    FIN
  OTHERWISE
    ERROR (ILLEGAL ARGUMENT K (PIDF))
  FIN
  RETURN
END
C POP(L1,N,M)
C SUBROUTINE POPS N INTEGERS OFF THE STACK AT L1 INTO THE
C ARRAY M.
SUBROUTINE POP(L1,N,M)
  USE SPM
  BEGIN SUB
  DIMENSION M(1)
  FOR (I)=1 TO (N)

```

Source Control Programs

```

        M(N+1-1)=IA(LCS(L1,IA(L1+3)+3-1))
        FIN
        IA(L1+3)=IA(L1+3)-N
        RETURN
    END
C PPLB(IPVT)
C SUBROUTINE PRINTS PARAMETERS OF FUNCTION AT LB IN DUMP FORMAT.
    SUBROUTINE PPLB(IPVT)
        USE SPM
        DIMENSION IPVT(1)
        BEGINSUB
        FOR IJ = 11 TO 11A(LB+3)
            CALL PIDF(LP-1+1,IPVT(IJ),1-1)
        FIN
        RETURN
    END
C PUSH(L1,N,M)
C SUBROUTINE PUSHES N INTEGERS IN ARRAY M INTO THE STACK AT L1.
    SUBROUTINE PUSH(L1,N,M)
        USE SPM
        BEGINSUB
        DIMENSION M(1)
        IFF 11A(L1+3)+N.GT.11A(L1+2)
            ERROR [STACK FULL (PUSH)]
        FIN
        FOR IJ = 11 TO 1N
            11A(LCS(L1,IA(L1+3)+2+1))=M(IJ)
        FIN
        11A(L1+3)=IA(L1+3)+N
        RETURN
    END
C PVLB(IPVT)
C SUBROUTINE PRINTS VARIABLES OF THE FUNCTION AT LB IN DUMP FORMAT.
    SUBROUTINE PVLB(IPVT)
        USE SPM
        DIMENSION IPVT(1)
        BEGINSUB
        FOR IJ = 11 TO 11A(LB+4)
            CALL PIDF(LV-1+1,IPVT(11A(LB+3)+1),1-1)
        FIN
        RETURN
    END
C RAC(L,I)
C FUNCTION RETURNS REAL VALUE OF ITEM I IN CHAIN STORAGE AT L.
    FUNCTION RAC(L,I)
        USE SPM
        BEGINFUN
        RAC=RA(LCS(L,I))
        RETURN
    END
C RIC(I)
C FUNCTION RETURNS THE ITH REAL INITIAL CONDITION OF THE FUNCTION AT LB.

```

Source Control Programs

```

FUNCTION RIC(I)
  USE SPM
  BEGINFUN
  RIC=RA(LI+I)
  RETURN
END
C RLIC(I,J)
C FUNCTION RETURNS THE JTH ELEMENT OF REAL LIST INITIAL CONDITION
C I OF THE FUNCTION AT LB.
  FUNCTION RLIC(I,J)
    USE SPM
    BEGINFUN
    RLIC=RA(LCS(IA(LI+I),J+2))
    RETURN
  END
C RLP(I,J)
C FUNCTION RETURNS THE JTH ELEMENT OF THE REAL LIST PARAMETER I OF
C THE FUNCTION AT LB.
  FUNCTION RLP(I,J)
    USE SPM
    BEGINFUN
    RLP=RA(LCS(IA(LP+I),J+2))
    RETURN
  END
C RLV(I,J)
C FUNCTION RETURNS THE JTH ELEMENT OF THE REAL LIST VARIABLE I OF THE
C FUNCTION AT LB.
  FUNCTION RLV(I,J)
    USE SPM
    BEGINFUN
    RLV=RA(LCS(IA(LV+I),J+2))
    RETURN
  END
C RNORM(RMU,SIGMA)
C FUNCTION RETURNS REAL RANDOM VARIABLE WITH NORMAL DISTRIBUTION.
  FUNCTION RNORM(RMU,SIGMA)
    USE SPM
    DATA IX,U/1,0./
    X=0.
    FOR I1 = 111 TO 1121
      CALL URN(IX,U)
      X=X+U
    FIN
    X=-6.*X
    RNORM=RMU+SIGMA X
    RETURN
  END
C RNORMT(RMU,SIGMA,RMAX,RMIN)
C FUNCTION RETURNS REAL RANDOM VARIABLE WITH TRUNCATED NORMAL
C DISTRIBUTION.
  FUNCTION RNORMT(RMU,SIGMA,RMAX,RMIN)
    USE SPM

```


Source Control Programs

```

DATA IX,U/1,0./
X=0.
FOR I1 = I11 TO I12
  CALL URN(IX,U)
  X=X+U
  FIN
  X=-6.+X
  R=RMU+SIGMA X
  IF(R.LT.RMIN)R=RMIN
  IF(R.GT.RMAX)R=RMAX
  RNORMT=R
  RETURN
END
C RP(I)
C FUNCTION RETURNS THE ITH REAL PARAMETER OF THE FUNCTION AT LB.
  FUNCTION RP(I)
    USE SPM
    BEGINFUN
    RP=RA(LP+I)
    RETURN
  END
C RUNIF(RM,SP)
C FUNCTION RETURNS REAL RANDOM VARIABLE WITH UNIFORM DISTRIBUTION.
  FUNCTION RUNIF(RM,SP)
    USE SPM
    BEGINFUN
    DATA IX,U/1,0./
    CALL URN(IX,U)
    RUNIF=RM-SP+2. U SP
    RETURN
  END
C RV(I)
C FUNCTION RETURNS THE ITH REAL VARIABLE OF THE FUNCTION AT LB.
  FUNCTION RV(I)
    USE SPM
    BEGINFUN
    RV=RA(LV+I)
    RETURN
  END
C SAMPL(I,J,V)
C SUBROUTINE RETURNS SAMPLE VALUE V FROM A POPULATION WITH
C DISTRIBUTION TYPE RAC(I,J+2) AND STATISTICS RAC(I,J+3), RAC(I,J+4),
C ETC.
  SUBROUTINE SAMPL(I,J,V)
    USE SPM
    BEGINSUB
    TR I5) [SAMPL(I,J, )] [I,J] [IX,2110]
    CASE [IFIX(RAC(I,J+2)).EQ.1]
      V=RUNIF(RAC(I,J+3),RAC(I,J+4))
    ORIF [IFIX(RAC(I,J+2)).EQ.2]
      V=RNORM(RAC(I,J+3),RAC(I,J+4))
    ORIF [IFIX(RAC(I,J+2)).EQ.3]

```

Source Control Programs

```

      V=RNORMT(RAC(I,J+3),RAC(I,J+4),RAC(I,J+5),RAC(I,J+6))
      ORIF (IFIX(RAC(I,J+2))EQ.4)
      CALL URN(IX,U)
      V=-RAC(I,J+3) ALOG(U)
      OTHERWISE
      ERROR ISAMPL ILLEGAL DISTRIBUTION TYPE]
      FIN
      TR (S) (SAMPL( , ,V)) (V) (IX,E12.4)
      RETURN
      END
C SCHED(T,NB,K,LD)
C SUBROUTINE SCHEDULES EVENT CODE K IN BLOCK NB FOR TIME T WITH EVENT
C DATA AT LOCATION LD (=0 IF NONE)
      SUBROUTINE SCHED(T,NB,K,LD)
      USE SPM
      BEGIN SUB
      IFF (T.LT.TIME)
      ERROR [SCHED ATTEMPT TO SCHEDULE EVENT PRIOR TO PRESENT TIME]
      FIN
      IFF (NFE.EQ.0)
      CALL INCREC(LEC)
      FIN
      RA(LFFE+1)=T
      IA(LFFE+2)=NB
      IA(LFFE+3)=K
      IA(LFFE+4)=LD
      L=LNE
      L1=LFFE
      LFFE=IA(LFFE)
      WHILE (RA(L+1).LE.T.AND.L.NE.0)
      L2=L
      L=IA(L)
      FIN
      IFF (L.EQ.LNE)
      LNE=L1
      IA(LT)=L
      ELSE
      IFF (L.NE.0)
      IA(L2)=L1
      IA(LT)=L
      ELSE
      IA(L2)=L1
      IA(LT)=0
      LLE=L1
      FIN
      FIN
      NFE=NFE-1
      NSE=NSE+1
      IF (NSE.GT.MNSE)MNSE=NSE
      RETURN
      END
C SEND

```

```

C SUBROUTINE TRANSMITS COMMUNICATIONS.
  SUBROUTINE SEND
  USE SPM
  BEGIN SUB
  DIMENSION IC(4), IK(4)
  EQUIVALENCE (MFROM, IC(1))
  IERR=0
  WHILE (IERR.EQ.0)
    TR (5) (SEND MFROM MTO ICTYPE IPOINT) &
    (MFROM, MTO, ICTYPE, IPOINT) (IX, 416)
    CASE (ICTYPE, LT, 0)
      NADDR=MFROM
      CALL POP(LMS, 4, IK)
      IFF (IC(1).NE.IK(1).OR.IC(2).NE.IK(2).OR.IC(3).NE.IK(3))
        ERROR (SEND ILLEGAL RESPONSE)
      FIN
      IFF (IA(LMS+3).GT.0)
        MFROM=IAC(LMS, IA(LMS+3)-1)
        MTO=IAC(LMS, IA(LMS+3))
      FIN
      ORIF (ICTYPE.GT.0)
        NADDR=MTO
        CALL PUSH(LMS, 4, IC)
      OTHERWISE
        ERROR (SEND ILLEGAL ICTYPE)
      FIN
      IFF (NADDR.GT.0)
        PERFORM CALL-BLOCK (1)
      ELSE
        RETURN
      FIN
    FIN
  ERROR (SEND ILLEGAL BLOCK TYPE)
C
  PROCEDURE CALL-BLOCK (1)
  IFF (ICTYPE.EQ.1)
    IPOINT
  ELSE
    CALL FPNTRS(NADDR)
    I=IA(LB)
    FIN
    J=1/100
    K=1-J/100
    CASE (J.EQ.1)
      CASE (K.EQ.1)
        CALL B101
      ORIF (K.EQ.2)
        CALL B102
      ORIF (K.EQ.3)
        CALL B103
      ORIF (K.EQ.4)
        CALL B104
  
```

Source Control Programs

```

ORIF [K.EQ.5]
  CALL B105
ORIF [K.EQ.6]
  CALL B106
ORIF [K.EQ.7]
  CALL B107
ORIF [K.EQ.8]
  CALL B108
ORIF [K.EQ.9]
  CALL B109
ORIF [K.EQ.10]
  CALL B110
OTHERWISE
  TERR=1
  FIN
ORIF [J.EQ.2]
  CASE [K.EQ.1]
    CALL B201
  ORIF [K.EQ.2]
    CALL B202
  ORIF [K.EQ.3]
    CALL B203
  ORIF [K.EQ.4]
    CALL B204
  ORIF [K.EQ.5]
    CALL B205
  ORIF [K.EQ.6]
    CALL B206
  ORIF [K.EQ.7]
    CALL B207
  ORIF [K.EQ.8]
    CALL B208
  ORIF [K.EQ.9]
    CALL B209
  ORIF [K.EQ.10]
    CALL B210
  OTHERWISE
    TERR=1
    FIN
ORIF [J.EQ.3]
  CASE [K.EQ.1]
    CALL B301
  ORIF [K.EQ.2]
    CALL B302
  ORIF [K.EQ.3]
    CALL B303
  ORIF [K.EQ.4]
    CALL B304
  ORIF [K.EQ.5]
    CALL B305
  ORIF [K.EQ.6]
    CALL B306

```

Source Control Programs

```

ORIF [K.EQ.7]
  CALL B307
ORIF [K.EQ.8]
  CALL B308
ORIF [K.EQ.9]
  CALL B309
ORIF [K.EQ.10]
  CALL B310
OTHERWISE
  TERR=1
  FIN
ORIF [J.EQ.4]
  CASE [K.EQ.1]
    CALL B401
  ORIF [K.EQ.2]
    CALL B402
  ORIF [K.EQ.3]
    CALL B403
  ORIF [K.EQ.4]
    CALL B404
  ORIF [K.EQ.5]
    CALL B405
  ORIF [K.EQ.6]
    CALL B406
  ORIF [K.EQ.7]
    CALL B407
  ORIF [K.EQ.8]
    CALL B408
  ORIF [K.EQ.9]
    CALL B409
  ORIF [K.EQ.10]
    CALL B410
  OTHERWISE
    TERR=1
    FIN
ORIF [J.EQ.5]
  CASE [K.EQ.1]
    CALL B501
  ORIF [K.EQ.2]
    CALL B502
  ORIF [K.EQ.3]
    CALL B503
  ORIF [K.EQ.4]
    CALL B504
  ORIF [K.EQ.5]
    CALL B505
  ORIF [K.EQ.6]
    CALL B506
  ORIF [K.EQ.7]
    CALL B507
  ORIF [K.EQ.8]
    CALL B508

```

Source Control Programs

```

ORIF [K.EQ.9]
  CALL B509
ORIF [K.EQ.10]
  CALL B510
OTHERWISE
  TERR=1
  FIN
ORIF [J.EQ.6]
  CASE [K.EQ.1]
    CALL B601
  ORIF [K.EQ.2]
    CALL B602
  ORIF [K.EQ.3]
    CALL B603
  ORIF [K.EQ.4]
    CALL B604
  ORIF [K.EQ.5]
    CALL B605
  ORIF [K.EQ.6]
    CALL B606
  ORIF [K.EQ.7]
    CALL B607
  ORIF [K.EQ.8]
    CALL B608
  ORIF [K.EQ.9]
    CALL B609
  ORIF [K.EQ.10]
    CALL B610
  OTHERWISE
    TERR=1
    FIN
ORIF [J.EQ.7]
  CASE [K.EQ.1]
    CALL B701
  ORIF [K.EQ.2]
    CALL B702
  ORIF [K.EQ.3]
    CALL B703
  ORIF [K.EQ.4]
    CALL B704
  ORIF [K.EQ.5]
    CALL B705
  ORIF [K.EQ.6]
    CALL B706
  ORIF [K.EQ.7]
    CALL B707
  ORIF [K.EQ.8]
    CALL B708
  ORIF [K.EQ.9]
    CALL B709
  ORIF [K.EQ.10]
    CALL B710

```

Source Control Programs

```
      OTHERWISE
      IERR=1
      FIN
ORIF [J.EQ.8]
  CASE [K.EQ.1]
    CALL B801
  ORIF [K.EQ.2]
    CALL B802
  ORIF [K.EQ.3]
    CALL B803
  ORIF [K.EQ.4]
    CALL B804
  ORIF [K.EQ.5]
    CALL B805
  ORIF [K.EQ.6]
    CALL B806
  ORIF [K.EQ.7]
    CALL B807
  ORIF [K.EQ.8]
    CALL B808
  ORIF [K.EQ.9]
    CALL B809
  ORIF [K.EQ.10]
    CALL B810
  OTHERWISE
  IERR=1
  FIN
ORIF [J.EQ.9]
  CASE [K.EQ.1]
    CALL B901
  ORIF [K.EQ.2]
    CALL B902
  ORIF [K.EQ.3]
    CALL B903
  ORIF [K.EQ.4]
    CALL B904
  ORIF [K.EQ.5]
    CALL B905
  ORIF [K.EQ.6]
    CALL B906
  ORIF [K.EQ.7]
    CALL B907
  ORIF [K.EQ.8]
    CALL B908
  ORIF [K.EQ.9]
    CALL B909
  ORIF [K.EQ.10]
    CALL B910
  OTHERWISE
  IERR=1
  FIN
OTHERWISE
```

Source Control Programs

```

      IERR=1
      FIN
    ENDPROC [1]
  END
C SET(KODE,IPVT,INTD,REALD)
C SUBROUTINE SETS PARAMETERS (KODE =1) AND VARIABLE INITIAL CONDITIONS
C (KODE = 2) READ FROM DEVICE N11 IN THE FORM
C (I N M) VALUES (I N M) VALUES
C WHERE I IS THE PARAMETER OR VARIABLE INDEX AND N AND M ARE THE FIRST
C AND LAST ELEMENT LIST INDEXES RESPECTIVELY. VALUES ARE CONVERTED
C TO THE TYPE SPECIFICATION OF PARAMETER/VARIABLE TYPE CODES IN ARRAY
C IPVT: 1 = INTEGER 2 = INTEGER LIST 3 = REAL 4 = REAL LIST
C ITEMS ARE BLANK DELIMITED AND MAY BE INTERSPERSED WITH COMMENTS.
      SUBROUTINE SET(KODE,IPVT,INTD,REALD)
      USE SPM
      BEGINSUB
      DIMENSION IPVT(1)
      L1=LBUF+2
      L80=LBUF+81
      CASE [KODE.EQ.1]
        LPV=1
        LIARA=LP
        IMA=IA(LB+3)
      GRIF [KODE.EQ.2]
        LPV=IA(LB+3)+1
        LIARA=L1
        IMA=IA(LB+4)
      OTHERWISE
        ERROR [SET ILLEGAL ARGUMENT KODE]
      FIN
      LSCAN=L1
      PERFORM GET-INM [1]
      WHILE [1.GT.0]
        IFF [N.EQ.0]
          PERFORM GET-V [2]
          PERFORM SET-ITEM [3]
        ELSE
          FOR [J] = [N] TO [M]
            PERFORM GET-V [2]
            PERFORM SET-LIST [4]
          FIN
        FIN
      PERFORM GET-INM [1]
      FIN
    RETURN
C
C PROCEDURE GET-INM [1]
  IFF [LSCAN.EQ.L1]
    PERFORM READ-LINE [5]
    IFF [KEYWD(L1).GT.0]
      BACKSPACE N11
      I=0

```


Source Control Programs

```

ELSE
  PERFORM SCAN-INM [6]
  FIN
ELSE
  PERFORM SCAN-INM [6]
  FIN
ENDPROC [1]

C
PROCEDURE GET-V [2]
PERFORM SCAN-NUMBER [8]
LPOINT=0
REAL=0.
NCH=ICH(IA(LSCAN))
CASE [NCH.GE.O.AND.NCH.LE.9]
  SIGN=1.
  ORIF [NCH.EQ.38]
    LSCAN=LSCAN+1
    NCH=ICH(IA(LSCAN))
    CASE [NCH.GE.O.AND.NCH.LE.9]
      SIGN=1.
      ORIF [NCH.EQ.42]
        LSCAN=LSCAN+1
        NCH=ICH(IA(LSCAN))
        IFF [NCH.LT.O.OR.NCH.GT.9]
          ERROR [SET ILLEGAL CHARACTER FOLLOWING +.]
          FIN
        SIGN=1.
      OTHERWISE
        ERROR [SET ILLEGAL CHARACTER FOLLOWING +.]
        FIN
    ORIF [NCH.EQ.39]
      LSCAN=LSCAN+1
      NCH=ICH(IA(LSCAN))
      CASE [NCH.GE.O.AND.NCH.LE.9]
        SIGN=-1.
        ORIF [NCH.EQ.42]
          LSCAN=LSCAN+1
          NCH=ICH(IA(LSCAN))
          IFF [NCH.LT.O.OR.NCH.GT.9]
            ERROR [SET ILLEGAL CHARACTER FOLLOWING -.]
            FIN
          SIGN=-1.
        OTHERWISE
          ERROR [SET ILLEGAL CHARACTER FOLLOWING -.]
          FIN
    ORIF [NCH.EQ.42]
      LPOINT=LSCAN
      LSCAN=LSCAN+1
      NCH=ICH(IA(LSCAN))
      IFF [NCH.LT.O.OR.NCH.GT.9]
        ERROR [SET ILLEGAL CHARACTER FOLLOWING .]
        FIN

```

Source Control Programs

```

SIGN=1.
OTHERWISE
  ERROR [SET DIGIT, +, -, OR EXPECTED BUT NOT FOUND]
  FIN
WHILE [NCH.LE.9.OR.NCH.EQ.42]
  CASE [NCH.LE.9]
    REAL=10. REAL+NCH
  ORIF [NCH.EQ.42]
    LPOINT=LSCAN
  OTHERWISE
    ERROR [SET CANNOT INTERPRET VALUE]
    FIN
  LSCAN=LSCAN+1
  IFF [LSCAN.GT.L80]
    ERROR [SET END OF VALUE NOT FOUND ON LINE]
    FIN
  NCH=ICH(IA(LSCAN))
  FIN
  IFF [LPOINT.EQ.0]
    REAL=SIGN REAL
  ELSE
    REAL=SIGN REAL/(10 (LSCAN-LPOINT-1))
    FIN
  INT=REAL
ENDPROC [2]
C
PROCEDURE SET-ITEM [3]
  IFF [I.LT.1.OR.I.GT.IMAX]
    ERROR [SET ILLEGAL INDEX]
    FIN
  CASE [IPVT(LPV+1).EQ.1]
    IA(LIARA+1)=INT
  ORIF [IPVT(LPV+1).EQ.3]
    RA(LIARA+1)=REAL
  OTHERWISE
    ERROR [SET ILLEGAL TYPE]
    FIN
  ENDPROC [3]
C
PROCEDURE SET-LIST [4]
  IFF [I.LT.1.OR.I.GT.IMAX]
    ERROR [SET ILLEGAL INDEX]
    FIN
  IFF [IA(LIARA+1).EQ.0]
    IFF [N.NE.1]
      ERROR [SET N MUST BE 1 WHEN LIST FIRST DEFINED]
      FIN
    IFF [M.GT.100]
      WARN [SET PARAMETER LIST EXCEEDS 100 ELEMENTS]
      FIN
    CALL ASTORE(IA(LIARA+1),M-N+3)
    IA(IA(LIARA+1)+3)=M-N+1

```

Source Control Programs

```

      FIN
      IFF [J.GT.IA(LIARA+1)+3]]
      ERROR [SET INDEX EXCEEDS LENGTH OF LIST]
      FIN
      CASE [IPVT(LPV+1).EQ.2]
      [A(LCS(LIARA+1),2+J)]=INT
      ORIF [IPVT(LPV+1).EQ.4]
      RA(LCS(LIARA+1),2+J)=REAL
      OTHERWISE
      ERROR [SET ILLEGAL TYPE]
      FIN
      ENDPROC [4]
C
      PROCEDURE READ-LINE [5]
      IEOF=0
      RDE [(IA(K),K=L1,L80)] [80A1]
      IFF [KEYWD(L1).EQ.0]
      WR [(IA(K),K=L1,L80)] [1X,80A1]
      IFF [POINT.EQ.3]
      WR [U:NC2] [(IA(K),K=L1,L80)] [1X,80A1]
      FIN
      FIN
      RDEND
      IFF [IEOF.EQ.1]
      ERROR [SET END OF INPUT FILE]
      FIN
      ENDPROC [5]
C
      PROCEDURE SCAN-INM [6]
      WHILE [IA(LSCAN).NE.1H(]
      LSCAN=LSCAN+1
      IFF [LSCAN.GT.L80]
      PERFORM READ-LINE [5]
      IFF [KEYWD(L1).GT.0]
      BACKSPACE N11
      I=0
      N=0
      M=0
      ENDPROC [6]
      ELSE
      LSCAN=L1
      FIN
      FIN
      LSCAN=LSCAN+1
      PERFORM GET-V [2]
      IFF [INT.GT.0.AND.INT.LE.IMAX]
      I=INT
      PERFORM SKIP-BLANKS [7]
      CASE &
      [(IA(LSCAN).EQ.1H).AND.(IPVT(LPV+1).EQ.1.OR.IPVT(LPV+1).EQ.3)]
      N=0

```

Source Control Programs

```

      M=0
    ORIF &
    [ICH(IA(LSCAN)).LE.9.AND.(IPVT(LPV+1).EQ.2.OR.IPVT(LPV+1).EQ.8
    4)]
      PERFORM GET-V [2]
      N=INT
      PERFORM SKIP-BLANKS [7]
      NCH=ICH(IA(LSCAN))
      IFF [NCH.GE.0.AND.NCH.LE.9]
        PERFORM GET-V [2]
        N=INT
      ELSE
        ERROR [SET ILLEGAL LIST INDEX M]
        FIN
      OTHERWISE
        ERROR [SET ILLEGAL LIST INDEX N]
        FIN
    ELSE
      ERROR [SET ILLEGAL LIST INDEX I]
      FIN
    IFF [IA(LSCAN).EQ.1H]
      LSCAN=LSCAN+1
    ELSE
      ERROR [SET RIGHT BRACKET EXPECTED BUT NOT FOUND]
      FIN
    ENDPROC [6]
  C
    PROCEDURE SKIP-BLANKS [7]
    WHILE [IA(LSCAN).EQ.1H ]
      PERFORM INCSCAN [10]
      FIN
    ENDPROC [7]
  C
    PROCEDURE SCAN-NUMBER [8]
    PERFORM SKIP-BLANKS [7]
    NCH=ICH(IA(LSCAN))
    WHILE [NCH.LT.0.OR.(NCH.GT.9.AND.NCH.NE.8
    38.AND.NCH.NE.39.AND.NCH.NE.42)]
      PERFORM SKIP-STRING [9]
      PERFORM SKIP-BLANKS [7]
      NCH=ICH(IA(LSCAN))
      FIN
    IFF [NCH.LT.0]
      ERROR [SET ILLEGAL CHARACTER]
      FIN
    ENDPROC [8]
  C
    PROCEDURE SKIP-STRING [9]
    WHILE [IA(LSCAN).NE.1H ]
      PERFORM INCSCAN [10]
      FIN
    ENDPROC [9]

```

Source Control Programs

```

C
  PROCEDURE INCSKAN [10]
    LSCAN=LSCAN+1
    IFF [LSCAN.GT.L80]
      PERFORM READ.LINE [5]
      IFF [KEYWD(L1).GT.0]
        BACKSPACE N1
        ERROR [SET UNEXPECTED KEYWORD]
      ELSE
        LSCAN=L1
      FIN
    FIN
  ENDPROC [10]
C GETV(KODE,IPVT,INTD,REALD)
C ENTRY RETURNS INTEGER (REAL) VALUE OF THE NEXT NUMBER IN THE
C INPUT DATA.
  ENTRY GETV(KODE,IPVT,INTD,REALD)
    L1=LBUF+2
    L80=LBUF+81
    PERFORM GET-V [2]
    INTD=INT
    REALD=REAL
    RETURN
C FBLANK(KODE,IPVT,INTD,REALD)
C ENTRY INCREMENTS SCAN POINTER LSCAN TO FIRST BLANK.
  ENTRY FBLANK(KODE,IPVT,INTD,REALD)
    L1=LBUF+2
    L80=LBUF+81
    LSCAN=L1
    WHILE [IA(LSCAN).NE.1H ]
      LSCAN=LSCAN+1
      IFF [LSCAN.GT.L80]
        ERROR [SET NO BLANK DELIMITER ON LINE]
      FIN
    FIN
  RETURN
END
C TIMER(K,I,T)
C SUBROUTINE SUPPLIES TIMING INFORMATION BASED UPON THE INTERACTION
C RESPONSE OF A 601 TIMR FUNCTION. I IS SET TO 1 AND T TO TIME IF UP.
C I IS SET TO 2 IF DOWN TILL THE NEXT SHIFT AT TIME T AND TO 3
C IF DOWN TILL THE NEXT INTERVAL AT TIME T.
  SUBROUTINE TIMER(K,I,T)
    USE SPM
    BEGINSUB
    CASE [K.EQ.2]
      IFF [IA(IPCINT+1).EQ.0]
        I=2
        T=RA(IPCINT+3)
      ELSE
        I=1
        T=TIME

```

Source Control Programs

```

      FIN
    CRIF [K.EQ.0]
      IFF [IA(IPOINT+4).EQ.0]
        I=3
        T=RA(IPOINT+6)
      ELSE
        I=1
        T=TIME
      FIN
    OTHERWISE
      ERROR [TIMER ILLEGAL TIMING CODE]
    FIN
    ICTYPE=IAC(LMS,IA(LMS+3)+1)
    IPOINT=IAC(LMS,IA(LMS+3)+2)
    RETURN
  END
C TRNFRM(NC,FB,L1,L2)
C SUBROUTINE TRANSFERS ALL (IF NC < 0) OR THE FIRST NC (IF NC > 0)
C CONTAINERS AND FRACTION FB OF THE BULK MATERIALS AT L1 TO L2.
  SUBROUTINE TRNFRM(NC,FB,L1,L2)
    USE SPM
    BEGIN SUB
      TR [5] [TRNFRM(NC,FB,L1,L2)] [NC,FB,L1,L2] [1X,16,E12.4,216]
      CALL NCBM(L1,NCL1,BML1)
      CASE [NC.LT.0]
        IFF [NCL1.GT.0]
          PERFORM TRANSFERCONTAINERS [1]
        FIN
      ORIF [NC.GT.0]
        IFF [NCL1.GE.NC]
          PERFORM TRANSFERCONTAINERS [1]
        ELSE
          ERROR [TRNFRM NOT ENOUGH CONTAINERS AT L1]
        FIN
      FIN
      IFF [FB.GT.0.]
        IFF [BML1.GT.0.]
          PERFORM TRANSFERBULK [2]
        FIN
      FIN
      CALL UMD(L1)
      CALL UMD(L2)
      TR [5] [TRNFRM DONE]
    RETURN
  C
  PROCEDURE TRANSFERCONTAINERS [1]
    PERFORM DETACHCONTAINERS [3]
    PERFORM ATTACHCONTAINERS [4]
    ENDPROC [1]
  C
  PROCEDURE TRANSFERBULK [2]
    PERFORM LOCATEBULK1 [5]

```

Source Control Programs

```

PERFORM LOCATEBULK2 [6]
PERFORM MOVEBULK [7]
ENDPROC [2]

C
PROCEDURE DETACHCONTAINERS [3]
L=IA(L1+5)
CASE [IA(L+6).EQ.1]
M=L+4
L=IA(L+4)
ORIF [IA(L+6).EQ.2]
M=L1+5
FIN
L1FC=L
IFF [NC.LT.0]
IA(M)=0
ELSE
N=1
WHILE [N.LT.NC]
L=IA(L+4)
N=N+1
FIN
IA(M)=IA(L+4)
IA(L+4)=0
FIN
ENDPROC [3]

C
PROCEDURE ATTACHCONTAINERS [4]
IFF [IA(L2+5).EQ.0]
IA(L2+5)=L1FC
IA(L1FC+2)=L2
ELSE
L=IA(L2+5)
WHILE [IA(L+4).GT.0]
L=IA(L+4)
FIN
IA(L+4)=L1FC
IA(L1FC+3)=L
IA(L1FC+2)=0
FIN
ENDPROC [4]

C
PROCEDURE LOCATEBULK1 [5]
IFF [IA(L1+6).EQ.1]
LIB=L1
ELSE
L=IA(L1+5)
IFF [IA(L+6).EQ.1]
LIB=L
ELSE
ERROR [TRNFRM LOCATING BULK AT L1]
FIN
FIN

```

Source Control Programs

```

C      ENDPROC [5]

PROCEDURE LOCATE BULK2 [6]
CASE [IA(L2+6).EQ.1]
  L2B=L2
ORIF [IA(L2+5).GT.0]
  L=IA(L2+5)
  CASE [IA(L+6).EQ.1]
    L2B=L
  ORIF [IA(L+6).EQ.2]
    CALL CMB(IA(L2+5),L2,0,L,0,1,0,0,0.)
    IA(L+2)=0
    IA(L+3)=IA(L2+5)
    L2B=IA(L2+5)
  OTHERWISE
    ERROR [TRNFRM LOCATING BULK AT L2]
  FIN
ORIF [IA(L2+5).EQ.0]
  CALL CMB(IA(L2+5),L2,0,0,0,1,0,0,0.)
  L2B=IA(L2+5)
OTHERWISE
  ERROR [TRNFRM LOCATING BULK AT L2]
  FIN
ENDPROC [6]

C      PROCEDURE MOVEBULK [7]
L=IA(L1B+5)
WHILE [L.GT.0]
  LH2=L2B
  LP2=0
  LM=L2B+5
  M=IA(LM)
  WHILE [M.GT.0.AND. IA(M+7).NE. IA(L+7)]
    LH2=0
    LP2=M
    M=IA(M+4)
  FIN
  IFF [M.EQ.0]
    IFF [LH2.GT.0]
      CALL CMB(M,LH2,0,0,0,0,IA(L+7),0,0.)
      IA(LH2+5)=M
    ELSE
      CALL CMB(M,0,LP2,0,0,0,IA(L+7),0,0.)
      IA(LP2+4)=M
    FIN
  FIN
  NT=FB FLOAT(IA(L+8))
  IA(M+8)=IA(M+8)*NT
  IA(L+8)=IA(L+8)-NT
  T=FB RA(L+9)
  RA(M+9)=RA(M+9)+T
  RA(L+9)=RA(L+9)-T

```


Source Control Programs

```

        L=IA(L+4)
        FIN
    ENDPROC [7]
    END
C UMD(LM)
C SUBROUTINE UPDATES MATERIAL DATA AT LM.
    SUBROUTINE UMD(LM)
    USE SPM
    BEGIN SUB
    TR [5] [UMD(LM)] [LM] [1X,110]
    MODE=1
    L=LM
    WHILE [L.NE.LM.OR.MODE.NE.4]
        PERFORM STEP [1]
        FIN
    TR [5] [UMD DONE]
    RETURN
C
    PROCEDURE STEP [1]
    CASE [MODE.EQ.1]
        CASE [IA(L+5).GT.0]
            L=IA(L+5)
            ORIF [IA(L+4).GT.0]
                MODE=2
                L=IA(L+4)
            OTHERWISE
                MODE=3
                PERFORM RESET [2]
        FIN
    ORIF [MODE.EQ.2]
        CASE [IA(L+5).GT.0]
            MODE=1
            L=IA(L+5)
            ORIF [IA(L+4).GT.0]
                L=IA(L+4)
            OTHERWISE
                MODE=3
                PERFORM RESET [2]
        FIN
    ORIF [MODE.EQ.3]
        PERFORM ADD [3]
        CASE [IA(L+2).GT.0]
            MODE=4
            L=IA(L+2)
            PERFORM UPDATE [5]
            ORIF [IA(L+3).GT.0]
                L=IA(L+3)
            OTHERWISE
                IFF [L.EQ.LM]
                    MODE=4
                    PERFORM RESET [2]
                    PERFORM UPDATE [5]

```

Source Control Programs

```

        ELSE
            PERFORM ERROR [4]
        FIN
    ORIF [MODE.EQ.4]
        CASE [IA(L+4).GT.0]
            MODE=2
            L=IA(L+4)
            ORIF [IA(L+3).GT.0]
                MODE=3
                L=IA(L+3)
            ORIF [IA(L+2).GT.0]
                L=IA(L+2)
            PERFORM UPDATE [5]
        OTHERWISE
            PERFORM ERROR [4]
        FIN
    OTHERWISE
        PERFORM ERROR [4]
    FIN
ENDPROC [1]
C
PROCEDURE RESET [2]
    N=0
    A=0
ENDPROC [2]
C
PROCEDURE ADD [3]
    IFF [IA(L+8).GT.0.OR.RA(L+9).GT.0.]
        N=N+IA(L+8)
        A=A+RA(L+9)
    ELSE
        PERFORM REMOVEBLOCK [6]
    FIN
ENDPROC [3]
C
PROCEDURE ERROR [4]
    ERROR {UMD MATERIAL DATA STRUCTURE}
ENDPROC [4]
C
PROCEDURE UPDATE [5]
    IA(L+8)=N
    RA(L+9)=A
ENDPROC [5]
C
PROCEDURE REMOVEBLOCK [6]
    IFF [L.NE.LM]
        CASE [IA(L+5).GT.0]
            ERROR {UMD REMOVEBLOCK 1}
        ORIF [IA(L+3).GT.0.AND.IA(L+2).EQ.0]
            IA(IA(L+3)+4)=IA(L+4)
            IFF [IA(L+4).GT.0]

```

Source Control Programs

```

      IA(IA(L+4)+3)=IA(L+3)
      FIN
    OR IF (IA(L+2).GT.0.AND.1A(L+3).EQ.0)
      IA(IA(L+2)+5)=IA(L+4)
      IFF (IA(L+4).GT.0)
        IA(IA(L+4)+2)=IA(L+2)
      FIN
    OTHERWISE
      ERROR [UMD REMOVEBLOCK 2]
      FIN
      CALL FSTORE(L)
      FIN
    ENDPROC [6]
  END
C U:N(IX,U)
C SJBROUTINE RETURNS UNIFORMLY DISTRIBUTED INTEGER IX ON THE INTERVAL
C [ ,10000] AND UNIFORMLY DISTRIBUTED REAL U ON THE INTERVAL [0.,1.).
  SUBROUTINE URN(IX,U)
    USE SPM
    BEGINSUB
      DATA X/D./
      IFF (IX.LT.0)
        CALL IRNFL(X)
        TR [5] [URN] [X] [1X,9HRNFL SEED,E12.4]
        CALL LRNFL(X)
      ELSE
        U=RNFL(0)
        IX=U 10000.+1
      FIN
    RETURN
  END
C WRMD(LM)
C SJBROUTINE WRITES MATERIAL DATA AT LM ON OUTPUT DEVICE NO1.
  SUBROUTINE WRMD(LM)
    USE SPM
    BEGINSUB
      IFF (LM.LE.0)
        WR [LM] [11X,14HNO MATERIAL AT,16]
      RETURN
    FIN
    MODE=1
    L=LM
    WHILE (L.NE.LM.OR.MODE.NE.4)
      PERFORM STEP [1]
    FIN
    PERFORM WRITEBLOCK [2]
    RETURN
  C
    PROCEDURE STEP [1]
    CASE (MODE.EQ.1)
      CASE (IA(L+5).GT.0)
        L=IA(L+5)

```

Source Control Programs

```

ORIF [IA(L+4).GT.0]
  MODE=2
  L=IA(L+4)
  OTHERWISE
    MODE=3
  FIN
ORIF [MODE.EQ.2]
  CASE [IA(L+5).GT.0]
    MODE=1
    L=IA(L+5)
  ORIF [IA(L+4).GT.0]
    L=IA(L+4)
  OTHERWISE
    MODE=3
  FIN
ORIF [MODE.EQ.3]
  IFF [L.NE.LM]
    PERFORM WRITEBLOCK [2]
  FIN
  CASE [IA(L+2).GT.0]
    MODE=4
    L=IA(L+2)
  ORIF [IA(L+3).GT.0]
    L=IA(L+3)
  OTHERWISE
    IFF [L.EQ.LM]
      MODE=4
    ELSE
      PERFORM ERROR [3]
    FIN
  FIN
ORIF [MODE.EQ.4]
  CASE [IA(L+4).GT.0]
    MODE=2
    L=IA(L+4)
  ORIF [IA(L+3).GT.0]
    PERFORM WRITEBLOCK [2]
    MODE=3
    L=IA(L+3)
  ORIF [IA(L+2).GT.0]
    PERFORM WRITEBLOCK [2]
    L=IA(L+2)
  OTHERWISE
    PERFORM ERROR [3]
  FIN
  OTHERWISE
    PERFORM ERROR [3]
  FIN
ENDPROC [1]
C
PROCEDURE WRITEBLOCK [2]
WR [L,(IA(L+1),1=2,0),RA(L+9)] [10X,516,213,110,E12.41

```

Source Control Programs

```

ENDPROC [2]
C
PROCEDURE ERROR [3]
ERROR [MATERIAL DATA STRUCTURE ERROR (WRMD)]
ENDPROC [3]
END
C ZIA(L,N)
C SUBROUTINE ZEROS N ELEMENTS OF IA STARTING AT L.
SUBROUTINE ZIA(L,N)
USE SPM
BEGINSUB
FOR [I] = 1 TO [N]
IA(L+I)=0
FIN
RETURN
END
C ZIAC(LOC)
C SUBROUTINE SETS ALL LOCATIONS IN THE BLOCK CHAIN AT LOC TO
C INTEGER ZERO (0).
SUBROUTINE ZIAC(LOC)
USE SPM
BEGINSUB
L=LOC
N=IA(L+1)
WHILE [N.GT.0]
FOR [I] = 1 TO [N]
IA(L+I)=0
FIN
IFF [IA(L).GT.0]
L=IA(L)
N=IA(L+1)
ELSE
N=0
FIN
FIN
RETURN
END
C ZRA(L,N)
C SUBROUTINE ZEROS N ELEMENTS OF RA STARTING AT L.
SUBROUTINE ZRA(L,N)
USE SPM
BEGINSUB
FOR [I] = 1 TO [N]
RA(L+I)=0.
FIN
RETURN
END
C ZRAC(LOC)
C SUBROUTINE SETS ALL LOCATIONS IN THE BLOCK CHAIN AT LOC TO
C REAL ZERO (0.).
SUBROUTINE ZRAC(LOC)
USE SPM

```

Source Control Programs

```

BEGINSUB
L=LOC
N=IA(L+1)
WHILE (N.GT.0)
  FOR (I) = (1) TO (N)
    RA(L+1+I)=0.
  FIN
  IFF (IA(L).GT.0)
    L=IA(L)
    N=IA(L+1)
  ELSE
    N=0
  FIN
FIN
RETURN
END
C ZTABLE(LOC)
C SUBROUTINE PUTS INTEGER 0 VALUES IN THE TABLE AT LOC.
SUBROUTINE ZTABLE(LOC)
USE SPM
BEGINSUB
N=IA(LOC+2)
IFF (IA(LOC).EQ.0)
  FOR (I) = (1) TO (N)
    IA(LOC+2+I)=0
  FIN
ELSE
  FOR (I) = (1) TO (N)
    IA(LCS(LOC,I+1))=0
  FIN
FIN
RETURN
END

```

Source Function Programs

FUNCTION PROGRAMS

```

C 101 RCMB 3/16/78
C RECEIVE MATERIAL BATCHES
LP+0 1 RESPONSE POINTER
C 1 3 PERIOD OF ARRIVAL SCHEDULE
C 2 1 NUMBER OF ARRIVALS PER PERIOD
C 3 4 SCHEDULED ARRIVAL TIMES
C 4 3 MAXIMUM DEVIATION FROM SCHEDULED ARRIVAL TIMES
C 5 1 NUMBER OF BULK MATERIAL TYPES
C 6 4 BULK MATERIAL SPECIFICATIONS (REPEAT FOR EACH TYPE)
C INTEGER MATERIAL TYPE CODE
C MEAN COUNT
C MAXIMUM UNIFORM DEVIATION OF COUNT
C MEAN MASS
C STANDARD NORMAL DEVIATION OF MASS
C MAXIMUM MASS
C MINIMUM MASS
C 7 1 NUMBER OF MATERIAL TYPES RECEIVED IN CONTAINERS
C 8 4 SPECIFICATIONS FOR MATERIAL RECEIVED IN CONTAINERS
C (REPEAT THE DATA LISTED ABOVE FOR EACH TYPE)
C 9 3 MEAN NUMBER OF CONTAINERS PER ARRIVAL
C 10 3 MAXIMUM UNIFORM DEVIATION OF NUMBER OF CONTAINERS
C 11 1 TIMING CODE
C 12 1 TIMER FUNCTION
C 13 1 DESTINATION FUNCTION
C 14 1 TRANSPORT FUNCTION
LV+0 1 ENTRY COUNT
C 1 1 NUMBER OF BATCHES RECEIVED
C LI+0 0
C 1
SUBROUTINE B101
USE SPM
BEGINBLOCK [101] [15] [2] [1,3,1,4,3,1,4,1,4,3,3,1,1,1,1,1]
CREATE
DESTROY
PARAMETERS
99400 ERROR [101 RCMB ATTEMPT TO SET INITIAL CONDITIONS]
99500 CONTINUE
WR [IA(LB+2),IA(LV+1)] &
[//,9H FUNCTION,14,1X,8H101 RCMB &
/11X,26HNUMBER OF BATCHES RECEIVED,110]
RESPOND [-5] [0]
99600 CONTINUE
IFF [RP(10) GT RP(9)]
ERROR [101 RCMB PARAMETER 10 LARGER THAN PARAMETER 9]
FIN
FOR [I] = [1] TO [IA(LB+4)]
IA(LV-1+I)=IA(LI-1+I)

```

Source Function Programs

```

      FIN
      CALL SCHED(RUNIF(RLP(3,1),RP(4)),1A(LB+2),1,0)
      RESPOND [-6] [0]
      DUMP
      EVENT
      CALL CEC(1,1P(2),NARVL)
      TIMER [11] [12]
      IFF [1.EQ.1]
        PERFORM RECEIVEMATERIALBATCH [1]
        IFF [LM.GT.0]
          PERFORM TRANSPORT [2]
          FIN
          IFF [NARVL.EQ.1P(2)]
            NARVL=1
            I=1
          ELSE
            NARVL=NARVL+1
            I=0
          FIN
          T=RUNIF(FLOAT(IFIX(TIME/RP(1))+1) RP(1)+RLP(3,NARVL),RP(4))
          CALL SCHED(T,1A(LB+2),NARVL,0)
        ELSE
          CALL SCHED(T,1A(LB+2),NARVL,0)
          FIN
          RESPOND [-8] [0]
          INTERACT
          ERROR [10] RCMB ILLEGAL INTERACTION]
          RESPOND [-9] [0]
C
      PROCEDURE RECEIVEMATERIALBATCH [1]
      IFF [1P(5).GT.0]
        CALL CRMD1(1P(5),1P(6),LBLK)
        FIN
      IFF [1P(7).GT.0]
        NC=IFIX(RUNIF(1P(9),1P(10)))
        IFF [INC.GT.0]
          CALL CRMD2(NC,1P(7),1P(8),LCON)
          FIN
      FIN
      CASE [1P(5).GT.0.AND.1P(7).GT.0]
        CALL TRNFRM(0,1.,LBLK,LCON)
        LM=LCON
      ORIF [1P(5).EQ.0.AND.1P(7).GT.0]
        LM=LCON
      ORIF [1P(5).GT.0.AND.1P(7).EQ.0]
        LM=LBLK
      OTHERWISE
        LM=0
      FIN
      ENDPROC [1]
C
      PROCEDURE TRANSPORT [2]

```


Source Function Programs

```

CALL PCOM(IP(14),9,1,LM,IA(LB+2),IP(13),0,0,0,0)
SEND
IF [IA(IP(14)+2).NE.-1]
  ERROR [101] RCMB MATERIAL NOT REMOVED AT ARRIVAL TIME]
FIN
CALL FSTORE(IP(14))
TR [1] [101] RCMB MATERIAL BATCH RECEIVED]
IA(LV+1)=IA(LV+1)+1
ENDPROC [2]
ENDBLOCK
D 102 IDAS 3/16/78
INITIATE DIVERSION ACTION SEQUENCE
LP+0 1 RESPONSE POINTER
C 1 4 MATERIAL DIVERSION OBJECTIVE TYPE (1 = COUNTABLE ITEMS
C OF BULK MATERIAL, 2 = MASS OF BULK MATERIAL, 3 = COUNTABLE
C ITEMS FROM CONTAINERS, 4 = MASS FROM CONTAINERS, 5 =
C CONTAINERS)
C MATERIAL TYPE (0 FOR BLENDED MATERIALS)
C MEAN NUMBER OF COUNTABLE ITEMS
C MAX UNIFORM DEVIATION OF COUNTABLE ITEMS
C MEAN MASS
C STANDARD DEVIATION OF MASS
C MEAN NUMBER OF CONTAINERS
C MAXIMUM UNIFORM DEVIATION OF NUMBER OF CONTAINERS
C EVENT SEQUENCE SPECIFICATION. REPEAT FOR EACH EVENT:
C 1) EVENT FUNCTION (NEGATIVE FOR FIRST EVENT OF SUBSEQUENCES),
C 2) EVENT STARTING TIME DELAY DISTRIBUTION TYPE, 3) EVENT
C STARTING TIME DELAY STATISTICS (0 OR MORE VALUES)
C LV+0 1 ENTRY COUNT
C LI+0 0
SUBROUTINE B102
USE SPM
BEGINBLOCK [102] [2] [1] [1,4,1]
CREATE
DESTROY
PARAMETERS
99400 CONTINUE
ERROR [102] IDAS ATTEMPT TO SET INITIAL CONDITIONS]
REPORT [102] IDAS]
99600 CONTINUE
FOR [1] = [1] TO [IA(LB+4)]
  IA(LV+1)=IA(LI+1)
  FIN
  FOR [1] = [1] TO [IA(IP(1)+3)]
    IF [RLP(1,1).LT.0]
      CALL SAMPL(IP(1),[1+1,TD])
      CALL SCHED(TD,-FIX(RLP(1,1)),1,IP(1))
    FIN
  FIN
RESPOND [-6] [0]
DUMP
EVENT

```

Source Function Programs

```

ERROR [102] [DAS] [ILLEGAL EVENT MESSAGE]
RESPOND [-8] [0]
INTERACT
ERROR [102] [DAS] [ILLEGAL INTERACTION]
RESPOND [-9] [0]
ENDBLOCK

C 301 TRMB 3/16/78
TRANSPORT MATERIAL BATCHES
LP+0 1 RESPONSE POINTER
1 3 MEAN TRANSPORT TIME
2 3 MAXIMUM UNIFORM DEVIATION OF TRANSPORT TIME
3 2 FUNCTIONS NOTIFIED AT CONDITION TIMES
4 1 DUMMY PARAMETER USED AS SAVE LOCATION
LV+0 1 ENTRY COUNT
1 1 NUMBER OF BATCHES IN TRANSPORT
2 1 MAXIMUM NUMBER OF BATCHES IN TRANSPORT
3 3 TIME MAXIMUM NUMBER OF BATCHES FIRST OCCURRED
4 1 LN OF MOST RECENT BATCH IN TRANSPORT
LI+0 0
1 0
2 0
3 0
4 0
SUBROUTINE B301
USE SPM
BEGINBLOCK [301] [5] [5] [1,3,3,2,1,1,1,3,1]
CREATE
99200 CONTINUE
IFF [1V(1).GT.0]
PERFORM FREEMATERIAL [3]
FIN
CALL DESTRO(IPVT)
RESPOND [-2] [0]
PARAMETERS
99400 CONTINUE
ERROR [301] TRMB ATTEMPT TO SET INITIAL CONDITIONS]
99500 CONTINUE
WR [IA(LB+2),IA(LV+2),RA(LV+3)] &
[/, 9H FUNCTION,14,1X,8H301/TRMB &
/11X,38HMAXIMUM NUMBER OF BATCHES IN TRANSPORT,110 &
/11X,27HTIME MAXIMUM FIRST OCCURRED,9X,E12.4]
RESPOND [-3] [0]
99600 CONTINUE
IFF [RP(2).GT.RP(1)]
ERROR [301] TRMB PARAMETER 2 LARGER THAN PARAMETER 1]
FIN
IFF [1V(1).GT.0]
PERFORM FREEMATERIAL [3]
FIN
FOR [1] = [1] TO [IA(LB+4)]
IA(LV-1+1)=IA(LI-1+1)
FIN

```

Source Function Programs

```

RESPOND [-6] [0]
DUMP
EVENT
CALL CEC(1,1,KODE)
LN=IA(IPPOINT+1)
PERFORM TRANSPORT [1]
RESPOND [-8] [0]
INTERACT
CASE [IA(IPPOINT+2),EQ.1]
  IALP+4)=IPPOINT
  CALL COPYB(IPPOINT,LN)
  IALV+1)=IA(LV+1)+1
  IFF [IV(1).GT.IV(2)]
    IA(LV+2)=IV(1)
    RA(LV+3)=TIME
  FIN
  IFF [RP(1).GT.0.]
    IFF [IV(4).LE.0]
      IA(LN+8)=0
      IA(LN+9)=0
      IA(LV+4)=LN
    ELSE
      IA(IV(4)+9)=LN
      IA(LN+8)=IV(4)
      IA(LN+9)=0
      IA(LV+4)=LN
    FIN
  CALL SCHED(TIME+RUNIF(RP(1),RP(2)),IA(LB+2),1,LN)
  PERFORM CONDITION [2]
  TR [3] [301 TRMB TRANSPORT STARTED]
ELSE
  PERFORM TRANSPORT [1]
  FIN
  IPOINT=IP(4)
  IAL(IPPOINT+2)=-1
  ORIF [IA(IPPOINT+2),EQ.5]
    IAL(IPPOINT+2)=-5
    IAL(IPPOINT+3)=0
    IAL(IPPOINT+4)=IV(4)
    IAL(IPPOINT+5)=IV(1)
    TR [3] [301 TRMB MATERIAL ACQUISITION ATTEMPT]
  OTHERWISE
    ERROR [301 TRMB [LEGAL INTERACTION]
  FIN
RESPOND [-9] [IPPOINT]
C
PROCEDURE TRANSPORT [1]
  IFF [IV(1),EQ.1]
    IAL(LV+4)=0
  ELSE
    IFF [IV(4),EQ.LN]
      IAL(LV+4)=IAL(LN+8)

```

Source Function Programs

```

      IFF [IA(LN+8).GT.0]
      IA(IA(LN+8)+9)=0
      FIN
    ELSE
      IFF [IA(LN+8).EQ.0]
      IA(IA(LN+9)+8)=0
    ELSE
      IA(IA(LN+8)+9)=IA(LN+9)
      IA(IA(LN+9)+8)=IA(LN+8)
    FIN
  FIN
FIN
PERFORM CONDITION [2]
IFF [IA(LN+5).GT.0]
  IA(LN+8)=0
  IA(LN+9)=0
  IA(LN+4)=IA(LB+2)
  MFEDM=IA(LB+2)
  MTO=IA(LN+5)
  ICTYPE=9
  IPOINT=LN
  SEND
  IFF [IA(IPOINT+2).NE.-1]
  ERROR [30] TRMB MATERIAL NOT REMOVED AT TRANSPORT TIME]
  FIN
  CALL FSTORE(IPOINT)
ELSE
  CALL FRMD(IA(LN+3))
  CALL FSTORE(LN)
  FIN
  IA(LV+1)=IA(LV+1)-1
  TR [1] [30] TRMB TRANSPORT COMPLETED]
ENDPROC [1]
C
PROCEDURE CONDITION [2]
IFF IIP(3).GT.0]
  CALL PCDM(ILP(3,1),9,9,0,IV(4),IV(1),0,0,0,0)
  FOR [1] = [1] TO [IA(IIP(3)+3)]
    MTO=ILP(3,1)
    ICTYPE=9
    IA(IPOINT+2)=9
    SEND
    IFF [IA(IPOINT+2).NE.-9]
    ERROR [30] TRMB RESPONSE TO CONDITION NOTICE]
    FIN
  FIN
  CALL FSTORE(IPOINT)
  FIN
ENDPROC [2]
C
PROCEDURE FREEMATERIAL [3]
LN=IV(4)

```

Source Function Programs

```

      WHILE (LN.GT.O)
        CALL FRMD(I(LN+3))
        CALL FSTORE(LN)
        LN=I(LN+8)
      FIN
    ENDPROC [3]
  ENDBLOCK
  STOP 3/16/78
C STORE MATERIAL
LP+O 1 RESPONSE POINTER
      1 1 HOLD UP MODEL
          1 = SINGLE DEPOSIT
          2 = RE-DEPOSIT ON ONLY SURFACE AREAS CONTACTED BY
            STATIC MATERIAL
          3 = RE-DEPOSIT WITH WASHDOWN OF ALL INTERNAL SURFACES
              OF THE BULK STORAGE VESSEL
      2 3 END AREA OF BULK STORAGE VESSEL
      3 3 VERTICAL SURFACE AREA CONTACTED PER UNIT MASS BULK STORAGE IF
        HOLD UP MODEL 1 OR 2, OR TOTAL AREA CONTACTED IF MODEL 3
      4 3 BULK STORAGE SURFACE HOLDUP COEFFICIENT
      5 2 REPORT CONTROL
          1) BULK STORAGE DATA
          2) CONTAINER STORAGE DATA
          3) MATERIAL CURRENTLY IN STORAGE
          4) MATERIAL CURRENTLY IN HOLDUP
      6 2 FUNCTIONS NOTIFIED AT DIVERSION OPPORTUNITIES
          - MATERIAL IN STORAGE
      8 1 - MATERIAL IN HOLDUP
LV+O 1 ENTRY COUNT
      1 1 NUMBER CONTAINERS IN STORAGE
      2 1 MAX NUMBER CONTAINERS IN STORAGE
      3 3 TIME MAX NUMBER CONTAINERS FIRST STORED
      4 1 NUMBER CONTAINER COUNT OBSERVATIONS
      5 3 LAST TIME NUMBER CONTAINERS CHANGED
      6 3 SUM OF N(T-TL) VALUES
      7 3 SUM OF N 2(T-TL) VALUES
      8 3 BULK MASS IN STORAGE
      9 3 MAX BULK MASS IN STORAGE
     10 3 TIME MAX BULK MASS FIRST STORED
     11 1 NUMBER BULK MASS OBSERVATIONS
     12 3 LAST TIME BULK MASS CHANGED
     13 3 SUM OF M(T-TL) VALUES
     14 3 SUM OF M 2(T-TL) VALUES
     15 3 TIME DATA COLLECTION STARTED
LI+O 0
      1 0
      2 0
      3 0
      4 0
      5 0
      6 0
      7 0
      8 0
      9 0
     10 0
     11 0
     12 0
     13 0
     14 0
     15 0

```

Source Function Programs

```

00000000      8      0.
00000000     10      0.
00000000     11      0.
00000000     12      0.
00000000     13      0.
00000000     14      0.
00000000     15      0.
SUBROUTINE B401
USE SPN
BEGINBLOCK (401) (9) (16) (1,1,3,3,3,2,2,1,1,1,1,3,1, &
3,3,3,3,3,1,3,3,3,3)
CREATE
CONTINUE
99200  IFF (IP(7).GT.0)
      CALL FRMD(IP(7))
      FIN
      IFF (IP(8).GT.0)
      CALL FRMD(IP(8))
      FIN
      CALL DESTRO(IPVT)
      RESPOND [-2] (0)
PARAMETERS
99400  CONTINUE
ERROR (401) STOM  ATTEMPT TO SET INITIAL CONDITIONS)
99500  CONTINUE
WR (IA(LB+2)) (// 9H FUNCTION,14,1X,8H401 STOM)
IFF (ILP(5,1).EQ.1)
      RA(LV+12)=TIME-RV(15)
      RA(LV+13)=RV(13)/RV(12)
      RA(LV+14)=RV(14)/RV(12)-RV(13)  2
      IFF (RV(14).GT.0.)
      RA(LV+14)=SQRT(RV(14))
      ELSE
      RA(LV+14)=0.
      FIN
WR (RV(8),RV(9),RV(10),IV(11),RV(13),RV(14)) &
/11X,12HBULK STORAGE &
/11X,20HCURRENTLY IN STORAGE,E11.4 &
/11X,19HMAXIMUM MASS STORED,E12.4,6H AT TIME,E12.4 &
/11X,19HNUMBER OBSERVATIONS,I12 &
/11X,4HMEAN,E27.4 &
/11X,18HSTANDARD DEVIATION,E13.4)
FIN
IFF (ILP(5,2).EQ.1)
      RA(LV+5)=TIME-RV(15)
      RA(LV+6)=RV(6)/RV(5)
      RA(LV+7)=RV(7)/RV(5)-RV(6)  2
      IFF (RV(7).GT.0.)
      RA(LV+7)=SQRT(RV(7))
      ELSE
      RA(LV+7)=0.

```

Source Function Programs

```

      FIN
      WR [IV(1),IV(2),RV(3),IV(4),RV(6),RV(7)] &
      [/11X,17HCONTAINER STORAGE &
      /11X,20HCURRENTLY IN STORAGE,I1,&
      /11X,21HMAXIMUM NUMBER STORED,I10,6H AT TIME,E11.4 &
      /11X,19HNUMBER OBSERVATIONS,I12 &
      /11X,4HMEAN,E27.4 &
      /11X,16HSTANDARD DEVIATION,E13.4]
      FIN
      IFF [ILP(5,3).EQ.1]
      WR [ ] [/11X,19HMATERIAL IN STORAGE]
      CALL WRMD(IP(7))
      FIN
      IFF [ILP(5,4).EQ.1]
      WR [ ] [/11X,18HMATERIAL IN HOLDUP]
      CALL WRMD(IP(8))
      FIN
      IFF [ILP(5,1).EQ.1.OR.ILP(5,2).EQ.1]
      WR [RV(15),TIME] &
      [/11X,20HREPORT TIME INTERVAL,E11.4,3H TO,E11.4]
      FIN
      RESPOND [-5] [0]
      CONTINUE
99600 IFF [IP(1).LT.1.OR.IP(1).GT.3]
      ERROR [401 STOM ILLEGAL HOLDUP MODEL SPECIFICATION]
      FIN
      IFF [IP(7).GT.0]
      CALL FRMD(IP(7))
      FIN
      CALL CMB(IA(LP+7),0,0,0,0,2,0,0,0.)
      IFF [IP(8).GT.0]
      CALL FRMD(IP(8))
      FIN
      CALL CMB(IA(LP+8),0,0,0,0,1,0,0,0.)
      FOR [I] = [1] TO [IA(LB+4)]
      IA(LV-I+1)=IA(LI-I+1)
      FIN
      RESPOND [-6] [0]
      DUMP
      EVENT
      ERROR [401 STOM ILLEGAL EVENT NOTICE]
      INTERACT
      CASE [IA(IPPOINT+2).EQ.1]
      PERFORM STORE [1]
      PERFORM CONDITION [10]
      ORIF [IA(IPPOINT+2).EQ.5]
      PERFORM DIVERSIONATTEMPT [2]
      ORIF [IA(IPPOINT+2).EQ.3]
      PERFORM INSTORE [3]
      ORIF [IA(IPPOINT+2).EQ.4]
      PERFORM INHOLDUP [4]
      ORIF [IA(IPPOINT+2).EQ.2]

```

Source Function Programs

```

        PERFORM REMOVE [5]
        PERFORM CONDITION [10]
    OTHERWISE
        ERROR [40] STOM ILLEGAL INTERACTION]
    FIN
    RESPOND [-9] [IPOINT]
C
    PROCEDURE STORE [1]
    CALL NCBM(IA(IPOINT+3),NC,BM)
    IA(LV+1)=IV(1)+NC
    RA(LV+8)=RV(8)+BM
    IFF (RV(8).GT.RV(9))
        BMX=RV(8)-RV(9)
        RA(LV+9)=RV(8)
        RA(LV+10)=TIME
    ELSE
        BMX=0.
    FIN
    IFF IBM.GT.0.AND.(LP(5,1).EQ.1)
        PERFORM UPDTBULKDATA [6]
    FIN
    IFF INC.GT.0.AND.(LP(5,2).EQ.1)
        PERFORM UPDTCONTDATA [7]
    FIN
    CALL TRNFRM[-1,1,,IA(IPOINT+3),IP(7)]
    CALL FRMD(IA(IPOINT+3))
    CASE (IP(1).EQ.1.AND.BMX.GT.0.)
        PERFORM SINGLEDEPOSITHOLDUP [8]
    ORIF (IP(1).GE.2.AND.BM.GT.0)
        PERFORM REDEPOSITHOLDUP [9]
    FIN
    IA(IPOINT+2)=-1
    TR [3] [40] STOM MATERIAL PLACED IN STORAGE]
    ENDPROC [1]
C
    PROCEDURE DIVERSIONATTEMPT [2]
    IA(IPOINT+2)=-5
    IA(IPOINT+5)=IP(7)
    TR [3] [40] STOM DIVERSION ATTEMPT]
    ENDPROC [2]
C
    PROCEDURE INSTORE [3]
    IA(IPOINT+2)=-3
    IA(IPOINT+3)=IP(7)
    TR [3] [40] STOM REQUEST FOR AVAILABLE (STORED) MATERIAL]
    ENDPROC [3]
C
    PROCEDURE INHOLDUP [4]
    IA(IPOINT+2)=-4
    IA(IPOINT+3)=IP(8)
    TR [3] [40] STOM REQUEST FOR HELD UP MATERIAL]
    ENDPROC [4]

```


Source Function Programs

```

C
PROCEDURE REMOVE [5]
CALL NCBM(IA(IP(1)+3),NC,BM)
IA(LV+1)=IV(1)-NC
RA(LV+8)=RV(8)-BM
IFF [BM.GT.0.AND.[LP(5,1).EQ.1]
PERFORM UPDTBULKDATA [6]
FIN
IFF [NC.GT.0.AND.[LP(5,2).EQ.1]
PERFORM UPDTCONTDATA [7]
FIN
IA(IP(1)+2)=-2
IP(1)=0
TR 131 [40] SYOM MATERIAL REMOVED FROM STORAGE]
ENDPROC [5]

C
PROCEDURE UPDTBULKDATA [6]
RA(LV+13)=RV(13)+RV(8) (TIME-RV(12))
RA(LV+14)=RV(14)+RV(8) RV(8) (TIME-RV(12))
IA(LV+11)=IV(11)+1
RA(LV+12)=TIME
ENDPROC [6]

C
PROCEDURE UPDTCONTDATA [7]
RA(LV+6)=RV(6)+IV(1) (TIME-RV(5))
RA(LV+7)=RV(7)+IV(1) :V(1) (TIME-RV(5))
IA(LV+4)=IV(4)+1
IFF [IV(1).GT.IV(2)]
IA(LV+2)=IV(1)
RA(LV+3)=TIME
FIN
RA(LV+5)=TIME
ENDPROC [7]

C
PROCEDURE SINGLEDEPOSITHOLDUP [8]
IFF [BMX.GT.0.]
F=(RP(2)+RP(3) RV(8)) RP(4)/RV(8)
PERFORM TRANSFERTHOLDUP [11]
FIN
ENDPROC [8]

C
PROCEDURE REDEPOSITHOLDUP [9]
IFF [BMX.GT.0.]
CALL TRNFRM(0,1.,IP(8),IP(7))
ELSE
CALL NCBM(IP(8),NC,BMH)
CASE [IP(1).EQ.2]
IFF [BMX.GT.0.]
F=((BMH-RP(2) RP(4)) (RV(8)/RV(9))+RP(2) RP(4))/BMH
CALL TRNFRM(0,F,IP(8),IP(7))
FIN
F=(RP(2)+RP(3) RV(8)) RP(4)/RV(8)

```

Source Function Programs

```

PERFORM TRANSFERTO HOLDUP [11]
ORIF [IP(1).EQ.3]
  IFF [BMH.GT.0]
    CALL TRNFRM(0,1.,IP(8),IP(7))
  FIN
  F=RP(3) RP(4)/RV(8)
  PERFORM TRANSFERTO HOLDUP [11]
FIN
ENDPROC [9]
C
PROCEDURE CONDITION [10]
  IFF [IP(6).GT.0]
    CALL PCOM([LP(6,1) 9,9,IP(7).0,0,0,0,0])
    FOR [I] = [1] TO [1A(IP(6)+. 1]
      MTC=LP(6,1)
      ICTYPE=9
      SEND
      IFF [1A(IP(6)).NE.-9]
        ERROR [40] STOP RESPONSE TO CONDITION NOTICE]
      FIN
    FIN
    CALL FSTORE(IP(6))
  FIN
ENDPROC [10]
C
PROCEDURE TRANSFERTO HOLDUP [11]
  IFF IF.GT.1.1
    CALL TRNFRM(0,1.,IP(7),IP(8))
  ELSE
    CALL TRNFRM(0,F,IP(7),IP(8))
  FIN
ENDPROC [11]
ENDBLOCK
C 501 INDA 3/16/78
C INDEPENDENT ACTION
C LP+0 1 RESPONSE POINTER
C 1 1 DURATION TIME DISTRIBUTION TYPE
C 2 4 DURATION TIME STATISTICS
C 3 4 STIMULUS SPECIFICATIONS (REPEAT FOR EACH TYPE)
C MONITOR FUNCTION
C STIMULUS TYPE
C DELAY TIME DISTRIBUTION TYPE
C DELAY TIME STATISTICS
C DURATION TIME DISTRIBUTION TYPE
C DURATION TIME STATISTICS
C INTENSITY DISTRIBUTION TYPE
C INTENSITY STATISTICS
C LV+4 2 FUNCTIONS NOTIFIED AT CONDITION TIMES
C LI+0 1 ENTRY COUNT
C 0
SUBROUTINE B501

```

Source Function Programs

```

USE SPM
BEGINBLOCK [501] [5] [1] [1,1,4,4,2,1]
CREATE
DESTROY
PARAMETERS
99400 CONTINUE
ERROR [501] INDA ATTEMPT TO SET INITIAL CONDITIONS]
REPORT [501] INDA]
INITIALIZE
DUMP
EVENT
CASE [IA(IPOINT).GT.0]
  CALL EVORS(1,2,0)
  IFF [IP(4).GT.0]
    CALL SCHED(TIME+D,IA(LB+2),0,0)
  FIN
  I=IA(IPOINT)
  L=IA(IPOINT+1)
  INXT=I+2+NSTATS(IFIX(RAC(L,I+3)))
  IFF [IFIX(RAC(L,INXT+2)).GT.0.AND.INXT.LT.IA(L+3)]
    CALL SAMPL(L,INXT+1,TD)
    CALL SCHED(TIME+D+TD,IFIX(RAC(L,INXT+2)),INXT,L)
  FIN
  IFF [IP(3).GT.0]
    PERFORM STIMULI [1]
  FIN
ORIF [IA(IPOINT).EQ.0]
  IFF [IP(4).EQ.0]
    ERROR [501] INDA CONDITION LIST]
  FIN
  CALL PCOM(ILP(4,1),9,9,0,0,0,0,0,0)
  FOR [I] = [1] TO [IA(IP(4)+3)]
    MFROM=IA(LB+2)
    MTO=ILP(4,1)
    ICTYPE=9
    IA(IPOINT+2)=9
    SEND
    IFF [IA(IPOINT+2).NE.-9]
      ERROR [501] INDA RESPONSE TO CONDITION NOTICE]
    FIN
  FIN
  CALL FSTORE(IPOINT)
  FIN
RESPOND [-8] [0]
INTERACT
ERROR [501] INDA ILLEGAL INTERACTION]
RESPOND [-9] [0]

C
PROCEDURE STIMULI [1]
  I=1
  CALL PCOM(IFIX(RLP(3,1)),9,7,IP(3),I+.,0,0,0,0,0)
  WHILE [I.LE.IA(IP(3)+3)]

```

Source Function Programs

```

MFROM=IA(LB+2)
MTO=RLP(3,1)
ICTYPE=9
IA(IPOINT+2)=7
IA(IPOINT+4)=1+1
SEND
IF [IA(IPOINT+2).NE.-7]
  ERROR [501] NDA STIMULI DATA NOT ACCEPTED]
FIN
I=1+2
I=I+NSTATS([FIX(RLP(3,1))]+1)
I=I+NSTATS([FIX(RLP(3,1))]+1)
I=I+NSTATS([FIX(RLP(3,1))]+1)
FIN
CALL FSTORE(IPOINT)
ENDPROC [1]
ENDBLOCK
C 502 MAQA 3/16/78
C ADVERSARY MATERIAL ACQUISITION ACTION
CC LP+0 1 RESPONSE POINTER
CC 1 1 DURATION TIME DISTRIBUTION TYPE
CC 2 4 DURATION TIME STATISTICS
CC 3 4 STIMULUS SPECIFICATIONS (REPEAT FOR EACH TYPE)
CC STIMULUS MONITOR FUNCTION
CC STIMULUS TYPE
CC DELAY TIME DISTRIBUTION TYPE
CC DELAY TIME STATISTICS
CC DURATION TIME DISTRIBUTION TYPE
CC DURATION TIME STATISTICS
CC INTENSITY DISTRIBUTION TYPE
CC INTENSITY STATISTICS
CC 4 2 TARGET FUNCTIONS
CC 5 2 FUNCTIONS NOTIFIED AT CONDITION TIMES
CC 6 1 -DIVERTED MATERIAL
CC LV+0 1 ENTRY COUNT
C LI+0 0
SUBROUTINE B502
USE SPM
BEGINBLOCK [502] [7] [1] [1,1,4,4,2,2,1,1]
DATA IX,U/1,0./
CREATE
99200 CONTINUE
IF [IP(6).GT.0]
  CALL FRMD(IP(6))
  FIN
  CALL DESTRO(IPVT)
  RESPOND [-2] [0]
  PARAMETERS
99400 CONTINUE
ERROR [502 MAQA ATTEMPT TO SET INITIAL CONDITIONS]
99500 CONTINUE
WR [IA(LB+2)] &

```

Source Function Programs

```

      I//,9H FUNCTION,14,1X,8H502 MAGA &
      /1X,17HDIVERTED MATERIAL]
      CALL WRMD(IP(6))
      RESPOND I-5] [0]
99600 CONTINUE
      IFF [IP(6).GT.0]
          CALL FRMD(IP(6))
          FIN
      CALL CMB(IA(LP+6),0,0,0,0,2,0,0,0.)
      FOR I1 = [1] TO [IA(LB+4)]
          [A(LV-1+I1)=IA(LI-1+I1)]
          FIN
      RESPOND I-6] [0]
      DUMP
      EVENT
      CASE [IA(IPOINT).GT.0]
          CALL EVDRS(1,2,0)
          IFF [IP(5).GT.0]
              CALL SCHED(TIME+D,IA(LB+2),0,0)
              FIN
          I=IA(IPOINT)
          J=IA(IPOINT+1)
          INXT=I+2+NSTATS(IFIX(RAC(J,I+3)))
          IFF [IFIX(RAC(J,INXT+2)).GT.0.AND.INXT.LT.1A(J+3)]
              CALL SAMPL(J,INXT+1,TD)
              CALL SCHED(TIME+D+TD,IFIX(RAC(J,INXT+2)),INXT,J)
              FIN
          IFF [IP(3).GT.0]
              PERFORM STIMULI [1]
              FIN
          IFF [IP(4).GT.0]
              PERFORM ATTEMPT [2]
              FIN
      ORIF [IA(IPOINT).EQ.0]
          IFF [IP(5).EQ.0]
              ERROR [502 MAGA OPPORTUNITY LIST]
              FIN
          CALL PCOM(ILP(5,1),9,9,0,0,0,0,0,0)
          FOR I1 = [1] TO [IA(IP(5)+3)]
              MTO=ILP(5,1)
              [A(IPOINT+2)=9]
              SEND
              IFF [IA(IPOINT+2).NE.-9]
                  ERROR [502 MAGA RESPONSE TO OPPORTUNITY NOTICE]
                  FIN
              FIN
          CALL FSTORE(IPOINT)
          FIN
      RESPOND I-6] [0]
      INTERACT
      IFF [IA(IPOINT+2).EQ.3]
          [A(IPOINT+2)=-3]

```

Source Function Programs

```

      IA(IPPOINT+3)=IP(6)
    ELSE
      ERROR [502 MAQA ILLEGAL INTERACTION]
    FIN
    RESPOND [-9] [IPPOINT]
  C
    PROCEDURE STIMULI [1]
    I=1
    CALL PCOM(1FIX(RLP(3,1)),9,7,IP(3),1+1,0,0,0,0,0)
    WHILE [1.LE.1A(IP(3)+3)]
      MTC=RLP(3,1)
      ICTYPE=9
      IA(IPPOINT+2)=7
      IA(IPPOINT+4)=1+1
      SEND
      IFF [1A(IPPOINT+2).NE.-7]
        ERROR [502 MAQA STIMULI DATA NOT ACCEPTED]
      FIN
      I=I+2
      I=I+NSTATS(1FIX(RLP(3,1)))+1
      I=I+NSTATS(1FIX(RLP(3,1)))+1
      I=I+NSTATS(1FIX(RLP(3,1)))+1
    FIN
    CALL FSTORE(IPPOINT)
  ENDPROC [1]
  C
    PROCEDURE ATTEMPT [2]
    FOR [I]= [1] TO [1A(IP(4)+3)]
      CALL PCOM(ILP(4,1),9,5,0,0,0,0,0,0,0)
      SEND
      IFF [1A(IPPOINT+2).NE.-5]
        ERROR [502 MAQA TARGET FUNCTION RESPONSE]
      FIN
      CASE [1A(IPPOINT+3).GT.0]
        LSM=1A(IPPOINT+3)
        PERFORM REMOVE MATERIAL [3]
      ORIF [1A(IPPOINT+4).GT.0]
        CALL URN(IX,U)
        N=U FLOCAT([1A(IPPOINT+5)) .9999+1
        L=1A(IPPOINT+4)
        M=1
        WHILE [M.LT.N]
          L=1A(L+8)
          IFF [L.LE.0]
            ERROR [502 MAQA MESSAGE BLOCK LINKAGE]
          FIN
          M=M+1
        FIN
        LSM=1A(L+3)
        PERFORM REMOVE MATERIAL [3]
      FIN
    CALL FSTORE(IPPOINT)

```

Source Function Programs

```

      FIN
      ENDPROC [2]
C
      PROCEDURE REMOVE MATERIAL [3]
      CALL NCBM(LSM,NC,2M)
      I=IFIX(RAC(J,3))
      CASE [I.EQ.1]
      IFF [2M.GT.0.]
      PERFORM REMOVEBULKITEMS [4]
      FIN
      ORIF [I.EQ.2]
      IFF [2M.GT.0.]
      PERFORM REMOVEBULKMASS [5]
      FIN
      ORIF [I.EQ.3]
      IFF [NC.GT.0]
      PERFORM REMOVECONTAINERITEMS [6]
      FIN
      ORIF [I.EQ.4]
      IFF [NC.GT.0]
      PERFORM REMOVECONTAINERBULK [7]
      FIN
      ORIF [I.EQ.5]
      IFF [NC.GT.0]
      PERFORM REMOVECONTAINERS [8]
      FIN
      OTHERWISE
      ERROR [502 MAQA ILLEGAL MATERIAL DIVERSION OBJECTIVE TYPE]
      FIN
      ENDPROC [3]
C
      PROCEDURE REMOVEBULKITEMS [4]
      PERFORM LOCATEBULKLSM [9]
      L=IA[LSMB+5]
      WHILE [L.NE.0.AND.1A(L+7).NE.IFIX(RAC(J,4))]
      L=IA(L+4)
      FIN
      IFF [L.EQ.0]
      LBI=0
      ELSE
      IFF [1A(L+8).GT.0]
      LBI=L+8
      ELSE
      LBI=0
      FIN
      FIN
      IFF [LBI.GT.0]
      CALL CMB(LDM,0,0,0,0,1,0,0,0.)
      CALL CMB(1A(LDM+5),LDM,0,0,0,0,IFIX(RAC(J,4)),0,0.)
      N=IUNIF(IFIX(RAC(J,5)),IFIX(RAC(J,6)))
      CASE [N.GT.1A(LBI)]
      N=1A(LBI)

```

Source Function Programs

```

      ORIF (N.LT.0)
        N=0
      FIN
      IA(I(LDM+5)+8)=N
      IA(LB1)=IA(LB1)-N
      CALL TRNFRM(0.1,LDM,IP(6))
      CALL FSTORE(LDM)
      FIN
C    ENDPROC [4]

      PROCEDURE REMOVEBULKMASS [5]
      PERFORM LOCATEBULKLSM [9]
      IFF (IFIX(RAC(J,4)) EQ 0)
        A=RNORM(IFIX(RAC(J,7)),IFIX(RAC(J,8)))
        IFF (A.LT.0.)
          A=0.
        FIN
      CALL TRNFRM(0,A/BM,LSM,IP(6))
      ELSE
        L=IA(LSMB+5)
        WHILE (L.NE.0.AND. IA(L+7).NE. IFIX(RAC(J,4)))
          L=IA(L+4)
        FIN
        IFF (L.NE.0)
          A=RNORM(RAC(J,7),RAC(J,8))
          CASE (A.LT.0.)
            A=0
          ORIF (A.GT. RA(L+9))
            A=RA(L+9)
          FIN
          CALL TRNFRM(0,A/RA(L+9),LSM,IP(6))
        FIN
      FIN
C    ENDPROC [5]

      PROCEDURE REMOVECONTAINERITEMS [6]
      PERFORM LOCATECONTAINER [10]
      LSMSAV=LSM
      LSM=LSMC
      PERFORM REMOVEBULKITEMS [4]
      LSM=LSMSAV
      ENDPROC [6]
C

      PROCEDURE REMOVECONTAINERBULK [7]
      PERFORM LOCATECONTAINER [10]
      LSMSAV=LSM
      LSM=LSMC
      PERFORM REMOVEBULKMASS [5]
      LSM=LSMSAV
      ENDPROC [7]
C

      PROCEDURE REMOVECONTAINERS [8]

```


Source Function Programs

```

N=1UNIF(1FIX(RAC(J,9)),1FIX(RAC(J,10)))
CASE [N.LT.0]
  N=0
ORIF [N.GT.NC]
  N=NC
FIN
CALL TRNFRM(N,0.,LSM,IP(6))
ENDPROC [8]

C
PROCEDURE LOCATEBULKLSM [9]
  IFF [IA(LSM+6).EQ.1]
    LSMB=LSM
  ELSE
    L=IA(LSM+5)
    IFF [IA(L+6).EQ.1]
      LSMB=L
    ELSE
      ERROR [502 MAQA LOCATING BULK AT LSM]
    FIN
  FIN
ENDPROC [9]

C
PROCEDURE LOCATECONTAINER [10]
  L=IA(LSM+5)
  IFF [IA(L+6).EQ.1]
    L=IA(L+4)
  FIN
  IFF [IA(L+6).NE.2]
    ERROR [502 MAQA LOCATING CONTAINER MATERIAL DATA STRUCTURE]
  FIN
  N=0
  K=L
  WHILE [K.NE.0]
    K=IA(K+4)
    N=N+1
  FIN
  CALL URN(IX,U)
  I=U FLOAT(N) .9999+1
  N=1
  K=L
  WHILE [N.LT.1]
    K=IA(K+4)
    N=N+1
  FIN
  LSMC=K
ENDPROC [10]
ENDBLOCK

C 503 DEPA 3/16/78
C DEPENDENT ACTION
C LP+0 1 RESPONSE POINTER
C 1 1 DURATION TIME DISTRIBUTION TYPE
C 2 4 DURATION TIME STATISTICS

```

Source Function Programs

```

C      3 4 STIMULUS SPECIFICATIONS (REPEAT FOR EACH TYPE)
C      STIMULUS MONITOR FUNCTION
C      STIMULUS TYPE
C      DELAY TIME DISTRIBUTION TYPE
C      DELAY TIME STATISTICS
C      DURATION TIME DISTRIBUTION TYPE
C      DURATION TIME STATISTICS
C      INTENSITY DISTRIBUTION TYPE
C      INTENSITY STATISTICS
C      4 1 MODE: 1 = CONDITIONAL STARTING OF ACTION
C      2 = CONDITIONAL COMPLETION OF ACTION
C      5 3 REFERENCE TIME
C      6 2 FUNCTIONS NOTIFIED AT COMPLETION TIMES
C      7 1 -REPORT DATA
LV+0 1 ENTRY COUNT
1 3 TIME T1 ACTION STARTED
2 3 DURATION OF ACTION
3 3 TIME DELAY OF NEXT ACTION
4 1 NEXT ACTION FUNCTION
5 1 NEXT ACTION FUNCTION INDEX INXT
6 1 - ACTION SEQUENCE LIST
7 3 TIME TO THE LAST CONDITION NOTICE WAS RECEIVED
LI+0 0
1 -2.
2 -1.
3 -1.
4 0
5 0
6 0
7 -1.
SUBROUTINE B503
USE SPM
BEGINBLOCK [503] [8] [8] [1,1,4,4,1,3,2,1,1,3,3,3,1,1,1,3]
CREAT
RA(LI+1)=-2.
RA(LI+2)=-1.
RA(LI+3)=-1.
RA(LI+7)=-1.
99200 RESPOND [-1] [0]
CONTINUE
IF IP(7).GT.0]
CALL FSTORE(IP(7))
FIN
CALL DESTRO(IPVT)
RESPOND [-2] [0]
PARAMETERS
99400 CONTINUE
ERROR [503 DEPA ATTEMPT TO SET INITIAL CONDITIONS]
99500 CONTINUE
PERFORM REPORT [4]
RESPOND [-5] [0]
99600 CONTINUE

```

Source Function Programs

```

      IFF [IP(7).EQ.0]
        CALL ASTORE([A(LP+7),8)
      FIN
      IFF [INRUN.EQ.1]
        IALCS(IP(7),1)=0
        RALCS(IP(7),2)=0
        RALCS(IP(7),3)=0
        RALCS(IP(7),4)=0
        RALCS(IP(7),5)=1.0E+50
      FIN
      FOR [I] = [1] TO [A(LB+4)]
        IALV(I+1)=IAL(I+1+1)
      FIN
      RESPOND [-6] [0]
      DUMP
      EVENT
      CASE [IA(IPPOINT).GT.0]
        CALL EVDRS(1,2,RA(LV+2))
        I=IA(IPPOINT)
        L=IA(IPPOINT+1)
        IALV+5=L
        INXT=I+2+NSTATS(IFIX(RAC(L,I+3)))
        IFF [IFIX(RAC(L,INXT+2)).GT.0.AND.INXT.LT.IA(L+3)]
          CALL SAMPL(L,INXT+1,TD)
          RA(LV+3)=TD
          IALV+4=IFIX(RAC(L,INXT+2))
          IALV+5=INXT
        FIN
      CASE [IP(4).EQ.1]
        IFF [RV(7).GT.TIME-RP(5)]
          PERFORM START1 [1]
        ELSE
          RA(LV+1)=-1.
        FIN
      ORIF [IP(4).EQ.2]
        PERFORM START2 [2]
      OTHERWISE
        ERROR [503 DEPA [ILLEGAL MODE SPECIFICATION]
      FIN
      ORIF [IA(IPPOINT).EQ.0]
        IFF [IP(6).EQ.0]
          ERROR [503 DEPA [CONDITION LIST]
        FIN
        CALL PCOM(ILP(6,1),9,9,0,0,0,0,0,0)
        FOR [I] = [1] TO [IA(IP(6)+3)]
          MTO=ILP(6,1)
          IAL(IPPOINT+2)=9
        SEND
        IFF [IA(IPPOINT+2).NE.-9]
          ERROR [503 DEPA [RESPONSE TO CONDITION NOTICE]
        FIN
      FIN

```

Source Function Programs

```

      CALL FSTORE(IPPOINT)
      FIN
      RESPOND [-8] [0]
      INTERACT
      IFF [IA(IPPOINT*2).NE.9]
        ERROR [503 DEPA ILLEGAL INTERACTION]
      FIN
      CASE [IP(4).EQ.1]
        CASE [RV(1).LT.-1.5]
          RA(LV*7)=TIME
          ORIF [RV(1).LT.-.5]
            PERFORM START1 [1]
          OTHERWISE
            ERROR [503 DEPA CONDITIONAL START]
            FIN
        ORIF [IP(4).EQ.2]
          CASE [RV(1).LT.-1.5]
            RA(LV*7)=TIME
            ORIF [RV(1).GE.(0.)AND.IV(4).GT.0]
              IFF [TIME.LE.RV(1)+RV(2)]
                CALL SCHED(RV(1)+RV(2)+RV(3),IV(4),IV(5),IV(6))
              ELSE
                CALL SCHED(TIME+RV(3),IV(4),IV(5),IV(6))
              IA(LV*4)=0
              IA(LV*5)=0
              IA(LV*6)=0
            FIN
          FIN
        FIN
      PERFORM UPDATEDATA [5]
      IA(IPPOINT*2)=-9
      RESPOND [-5] [IPPOINT]
C
      PROCEDURE START1 [1]
      IFF [IV(4).GT.0]
        CALL SCHED(TIME+RV(2)+RV(3),IV(4),IV(5),IV(6))
      FIN
      IFF [IP(3).GT.0]
        PERFORM STIMULI [3]
      FIN
      IFF [IP(6).GT.0]
        CALL SCHED(TIME+RV(2),IA(LB*2),0,0)
      FIN
      ENDPROC [1]
C
      PROCEDURE START2 [2]
      RA(LV*1)=TIME
      IFF [IP(3).GT.0]
        PERFORM STIMULI [3]
      FIN
      IFF [IP(6).GT.0]
        CALL SCHED(TIME+RV(2),IA(LB*2),0,0)

```

Source Function Programs

```

      FIN
      IFF [IV(4).GT.0]
      IFF [RV(7).GT.TIME+RV(2)+RV(5)]
      CALL SCHED(TIME+RV(2)+RV(3),IV(4),IV(5),IV(6))
      IA(LV+4)=0
      IA(LV+5)=0
      IA(LV+6)=0
      FIN
      FIN
      ENDPROC [2]
C
      PROCEDURE STIMULI [3]
      I=1
      CALL PCOM(FIX(RLP(3,1)),9,7,IP(3),I+1,0,0,0,0,0)
      WHILE [I.LE.IA(IP(3)+3)]
      MTO=RLP(3,I)
      ICTYPE=9
      IA(IPOINT+2)=7
      IA(IPOINT+5)=I+1
      SEND
      IFF [IA(IPOINT+2).NE.-7]
      ERROR (503 DEPA STIMULI DATA NOT ACCEPTED)
      FIN
      I=I+2
      I=I+NSTATS(FIX(RLP(3,1)))+1
      I=I+NSTATS(FIX(RLP(3,1)))+1
      I=I+NSTATS(FIX(RLP(3,1)))+1
      FIN
      CALL FSTORE(IPOINT)
      ENDPROC [3]
C
      PROCEDURE REPORT [4]
      WR [IA(LB+2)] &
      [//,9H FUNCTION,14,1X,8H503 DEPA]
      RA(LCS(IP(7),2))=RA(LCS(IP(7),2))/FLOAT(IA(LCS(IP(7),1)))
      SD=RA(LCS(IP(7),3))/FLOAT(IA(LCS(IP(7),1)))-RA(LCS(IP(7),2)) 2
      IFF [SD.GT.0.]
      SD=SD*SD
      ELSE
      SD=0.
      FIN
      WR [RA(LCS(IP(7),2)),SD] &
      [11X,22HMEAN TIME OF CONDITION,E12.4 &
      /11X,18HSTANDARD DEVIATION,E12.4]
      WR [IA(LCS(IP(7),1)) RA(LCS(IP(7),4)),RA(LCS(IP(7),5))] &
      [11X,22HNUMBER OF OBSERVATIONS,110 &
      /11X,12HMAXIMUM TIME,E12.4 &
      /11X,12HMINIMUM TIME,E12.4]
      ENDPROC [4]
C
      PROCEDURE UPDATEDATA [5]
      IA(LCS(IP(7),1))=IA(LCS(IP(7),1))+1

```

Source Function Programs

```

RA(LCS(IP(7),2))=RA(LCS(IP(7),2))+TIME
RA(LCS(IP(7),3))=RA(LCS(IP(7),3))+TIME TIME
IFF (TIME GT. RA(LCS(IP(7),4)))
  RA(LCS(IP(7),4))=TIME
  FIN
IFF (TIME LT. RA(LCS(IP(7),5)))
  RA(LCS(IP(7),5))=TIME
  FIN
ENDPROC (5)
ENDBLOCK
C 504 MODA 3/16/78
C MODIFICATION ACTION
C LP+0 1 RESPONSE POINTER
C 1 1 DURATION TIME DISTRIBUTION TYPE
C 2 4 DURATION TIME STATISTICS
C 3 4 STIMULUS SPECIFICATIONS (REPEAT FOR EACH TYPE)
C STIMULUS MONITOR FUNCTION
C STIMULUS TYPE
C DELAY TIME DISTRIBUTION TYPE
C DELAY TIME STATISTICS
C DURATION TIME DISTRIBUTION TYPE
C DURATION TIME STATISTICS
C INTENSITY DISTRIBUTION TYPE
C INTENSITY STATISTICS
C 4 2 MODIFICATION SPECIFICATIONS (REPEAT FOR EACH MODIFICATION)
C FUNCTION
C PARAMETER/VARIABLE
C FIRST LIST INDEX
C LAST LIST INDEX
C 5 4 NEW VALUES
C 6 4 INITIAL VALUES
C 7 2 CONDITION FUNCTIONS
C LV+0 1 ENTRY COUNT
C LI+0 0
SUBROUTINE B504
USE SPM
BEGINBLOCK [504] [8] [1] [1,1,4,4,2,4,4,2,1]
CREATE
DESTROY
PARAMETERS
99400 CONTINUE
ERROR [504 MODA] ATTEMPT TO SET INITIAL CONDITIONS]
REPORT [504 MODA]
99600 CONTINUE
FOR [1] = [1] TO [1A(LB+4)]
  [A(LV-1+1)]=1A(LI-1+1)
  FIN
NPVAL=6
PERFORM MODIFY [2]
RESPOND [-6] [0]
DUMP
EVENT

```

Source Function Programs

```

CASE [IA(IPPOINT).GT.0]
  CALL EVDRS(1,2,D)
  IFF [IP(7).GT.0]
    CALL SCHED(TIME+D,IA(LB+2),0,0)
  FIN
  I=IA(IPPOINT)
  L=IA(IPPOINT+1)
  INXT=I+2+INSTATS(IFIX(RAC(L,I+3)))
  IFF [IFIX(RAC(L,INXT+2)).GT.0.AND.[INXT.LT.[A(L+3)]]
    CALL SAMPL(L,INXT+1,TD)
    CALL SCHED(TIME+D+TD,IFIX(RAC(L,INXT+2)),INXT,L)
  FIN
  IFF [IP(3).GT.0]
    PERFORM STIMULI [1]
  FIN
  IFF [IP(4).GT.0]
    NPVAL=5
    PERFORM MODIFY [2]
  FIN
ORIF [IA(IPPOINT).EQ.0]
  IFF [IP(7).EQ.0]
    ERROR [504 MODA CONDITION LIST]
  FIN
  CALL PCOM([LP(7,1),9,9,0,0,0,0,0,0,0])
  FOR [I] = [1] TO [IA(IP(7)+3)]
    MTO=LP(7,1)
    IA(IPPOINT+2)=9
    SEND
    IFF [IA(IPPOINT+2).NE.-9]
      ERROR [504 MODA RESPONSE TO CONDITION NOTICE]
    FIN
  FIN
  CALL FSTORE(IPPOINT)
  FIN
RESPOND [-8] [0]
INTERACT
ERROR [504 MODA ILLEGAL INTERACTION]
RESPOND [-9] [0]

C
PROCEDURE STIMULI [1]
  I=1
  CALL PCOM([IFIX(RLP(3,1)),9,7,IP(3),I+1,0,0,0,0,0])
  WHILE [I.LE.[A(IP(3)-3)]]
    MFROM=IA(LB+2)
    MTO=RLP(3,1)
    ICTYPE=9
    IA(IPPOINT+2)=7
    IA(IPPOINT+4)=I+1
    SEND
    IFF [IA(IPPOINT+2).NE.-7]
      ERROR [504 MODA STIMULI DATA NOT ACCEPTED]
    FIN
  FIN

```

Source Function Programs

```

      I=I+2
      I=I+NSTATS(IFIX(RLP(3,I)))+1
      I=I+NSTATS(IFIX(RLP(3,I)))+1
      I=I+NSTATS(IFIX(RLP(3,I)))+1
      FIN
C     CALL FSTORE(IPPOINT)
      ENDPROC [1]

C     PROCEDURE MODIFY [2]
      I=1
      J=1
      WHILE [1.LE.[A(IP(4)+3)]
        NF=ILP(4,I)
        LBF=IA(LCS(LBPT,NF+1))
        IFF [LBF.GT.0]
          PERFORM MODIFYFUNCTION [3]
        ELSE
          ERROR [504 MODA  ATTEMPT TO MODIFY AN UNDEFINED FUNCTION]
        FIN
        I=I+4
      FIN
      ENDPROC [2]

C     PROCEDURE MODIFYFUNCTION [3]
      NPV=ILP(4,I+1)
      N=ILP(4,I+2)
      M=ILP(4,I+3)
      CASE [NPV.GT.0]
        LPVF=LBF+5
        PERFORM ASSIGNNEWPARAMETER/VARIABLE [4]
      ORIF [NPV.LT.0]
        NPV=-NPV
        PVF=LBF+5+IA(LBF+3)
        PERFORM ASSIGNNEWPARAMETER/VARIABLE [4]
      OTHERWISE
        ERROR [504 MODA  ILLEGAL PARAMETER/VARIABLE NUMBER ZERO]
      FIN
      ENDPROC [3]

C     PROCEDURE ASSIGNNEWPARAMETER/VARIABLE [4]
      CASE [N.EQ.0]
        CASE [M.EQ.0]
          IA(LPVF+NPV)=IFIX(RLP(NPVAL,J))
        ORIF [M.EQ.1]
          RA(LPVF+NPV)=RLP(NPVAL,J)
        OTHERWISE
          ERROR [504 MODA  LAST INDEX FOR SCALAR]
        FIN
      J=J+1
      ORIF [N.GT.0]
        IFF [M.GT.0.AND.M.GE.N]
          FOR [K] = [N] TO [M]

```


Source Function Programs

```

      IA(LCS(IA(LPVF+NPV),K+2))=IFIX(RLP(NPVAL,J))
      J=J+1
      FIN
    ELSE
      ERROR [504 MODA LAST INDEX FOR INTEGER LIST]
      FIN
    DRIF [N LT.O]
      N=-N
      IFF [M.GT.O.AND.M.GE.N]
        FOR [K] = [N] TO [M]
          RA(LCS(IA(LPVF+NPV),K+2))=RLP(NPVAL,J)
          J=J+1
        FIN
      ELSE
        ERROR [504 MODA LAST INDEX FOR REAL LIST]
        FIN
      FIN
    ENDPROC [41]
    ENDBLOCK
CC 505 HDAS 3/18/78
    HALT DIVERSION ACTION SEQUENCE
CCCC LP+0 1 RESPONSE POINTER
CCCC LV+0 1 ENTRY COUNT
CCCC 1 1 HALT STATUS
CCCC 2 1 SAVE I = IA(IPOINT)
CCCC 3 1 SAVE L = IA(IPOINT+1)
CCCC LI+0 0
CCCC 1 0
CCCC 2 0
CCCC 3 0
    SUBROUTINE B505
    USE SPM
    BEGINBLOCK [505] [1] [4] [1,1,1,1]
    CREATE
    DESTROY
99300 CONTINUE
    ERROR [505 HDAS ATTEMPT TO SET PARAMETERS]
    VARIABLES
    REPORT [505 HDAS]
    INITIALIZE
    DUMP
    EVENT
    IA(LV+2)=IA(IPOINT)
    IA(LV+3)=IA(IPOINT+1)
    IFF [IV(1).EQ.0]
      PERFORM CONTINUATION [1]
    FIN
    RESPOND [-8] [0]
    INTERACT
    IFF [IA(IPOINT+2).EQ.8]
      IFF [IV(1).EQ.1]
        IA(LV+1)=0

```

Source Function Programs

```

        PERFORM CONTINUATION [1]
        FIN
    ELSE
        ERROR [505 HDAS ILLEGAL INTERACTION]
        FIN
    IA(IPOINT+2)=-8
    RESPOND [-9] [0]
C
    PROCEDURE CONTINUATION [1]
    I=I(2)
    L=I(3)
    INXT=I+2+NSTATS(IFIX(RAC(L,I+3)):
    IFF (IFIX(RAC(L,INXT+2)).GT.0.AND.INXT.LT.IA(L+3))
    CALL SAMPL(L,INXT+1,TD)
    CALL SCHED(TIME+TD,IFIX(RAC(L,INXT+2)),INXT,L)
    FIN
    ENDPROC [1]
    ENDBLOCK
C 601 TIMR 3/16/78
C
C LP+0 1 RESPONSE POINTER
C 1 3 PERIOD
C 2 1 NUMBER OF SHIFTS PER PERIOD
C 3 4 SHIFT STARTING TIMES
C 4 4 SHIFT ENDING TIMES
C 5 1 NUMBER OF INTERVALS PER PERIOD
C 6 4 INTERVAL STARTING TIMES
C 7 4 INTERVAL ENDING TIMES
C LV+0 1 ENTRY COUNT
C 1 3 TIME CURRENT PERIOD WILL END
C 2 1 CURRENT SHIFT
C 3 3 TIME CURRENT SHIFT WILL END
C 4 3 TIME NEXT SHIFT WILL BEGIN
C 5 1 CURRENT INTERVAL
C 6 3 TIME CURRENT INTERVAL WILL END
C 7 3 TIME NEXT INTERVAL WILL BEGIN
C LI+0 0
C 1 RP(1)
C 2 0
C 3 0
C 4 RLP(3,1)
C 5 0
C 6 0
C 7 RLP(6,1)
C
C SUBROUTINE B601
C USE SPH
C BEGINBLOCK [601] [8] [8] [1,3,1,4,4,1,4,4,1,3,1,3,3,1,3,3]
C CREATE
C DESTROY
C PARAMETERS
99400 CONTINUE
    ERROR [601 TIMR ATTEMPT TO SET INITIAL CONDITIONS]

```

Source Function Programs

```

9960G REPORT [601 TIMR:
CONTINUE
RA(LV+1)=RP(1)
RA(LV+4)=RLP(3,1)
RA(LV+7)=RLP(6,1)
FOR I1 = I1 TO I1(LB+4)
  IA(LV+1+I1)=IA(LV+1+I1)
FIN
CALL SCHED(TIME,IA(LB+2),1,0)
IFF IIP(2).GT.0)
CALL SCHED(RV(4),IA(LB+2),2,0)
FIN
IFF IIP(5).GT.0)
CALL SCHED(RV(7),IA(LB+2),3,0)
FIN
RESPOND [-5] I0)
DUMP
EVENT
CALL CEC(1,5,KODE)
CASE [KODE.EQ.1]
  RA(LV+1)=TIME+RP(1)
  CALL SCHED(RV(1),MTO,1,0)
  TR I1 [601 TIMR BEGIN PERIOD]
  TPB=TIME
  IA(LV+2)=0
  IA(LV+5)=0
ORIF [KODE.EQ.2]
  IA(LV+2)=-IA(LV+2)+1
  RA(LV+3)=TPB+RLP(4,IV(2))
  CALL SCHED(RV(3),MTO,4,0)
  IFF IIV(2).NE.IP(2))
    RA(LV+4)=TPB+RLP(3,IV(2)+1)
  ELSE
    RA(LV+4)=TPB+RP(1)+RLP(3,1)
  FIN
  TR I1 [601 TIMR BEGIN SHIFT]
ORIF [KODE.EQ.3]
  IA(LV+5)=-IA(LV+5)+1
  RA(LV+6)=TPB+RLP(7,IV(5))
  CALL SCHED(RV(6),MTO,5,0)
  IFF IIV(5).NE.IP(5))
    RA(LV+7)=TPB+RLP(6,IV(5)+1)
  ELSE
    RA(LV+7)=TPB+RP(1)+RLP(6,1)
  FIN
  TR I1 [601 TIMR BEGIN INTERVAL]
ORIF [KODE.EQ.4]
  IA(LV+2)=-IA(LV+2)
  CALL SCHED(RV(4),MTO,2,0)
  RA(LV+3)=0
  TR I1 [601 TIMR END SHIFT]
ORIF [KODE.EQ.5]

```

Source Function Programs

```

      IA(LV+5)=-IA(LV+5)
      CALL SCHED(RV(7),MTO,3,0)
      RA(LV+6)=0
      TR (1) (601 TIMR END INTERVAL)
      FIN
      RESPOND [-8] (0)
      INTERACT
      TR (3) (601 TIMR RETURNING TIMING DATA)
      RESPOND [-9] (LV+1)
      ENDBLOCK
C 602 SCAS 3/16/78
C START CONTROL ACTION SEQUENCE
C LP+0 1 RESPONSE POINTER
C 1 4 ACTION SEQUENCE SPECIFICATION. REPEAT FOR EACH ACTION:
C 1) FUNCTION (NEGATIVE FOR FIRST ACTION OF SUBSEQUENCES),
C 2) ACTION STARTING TIME DELAY DISTRIBUTION TYPE, 3) ACTION
C STARTING TIME DELAY STATISTICS (0 OR MORE VALUES)
C LV+0 1 ENTRY COUNT
C LI+0 0
      SUBROUTINE B602
      USE SPM
      BEGINBLOCK (602) (2) (1) (1,4,1)
      CREATE
      DESTROY
      PARAMETERS
99400 CONTINUE
      ERROR (602 SCAS ATTEMPT TO SET INITIAL CONDITIONS)
      REPORT (602 SCAS)
      INITIALIZE
      DUMP
      EVENT
      ERROR (602 SCAS ILLEGAL EVENT MESSAGE)
      RESPOND [-8] (0)
      INTERACT
      IFF (IA(IPOINT+2) EQ 11)
      FOR (1) = (1) TO (IA(IP(1)+3))
      IFF (RLP(1,1) LT 0)
      CALL SAMPL(IP(1),1+1,TD)
      CALL SCHED(TIME+TD,-IFIX(RLP(1,1)),1,IP(1))
      FIN
      ELSE
      FIN
      ERROR (602 SCAS ILLEGAL INTERACTION)
      FIN
      IA(IPOINT+2)=-11
      RESPOND [-9] (IPOINT)
      ENDBLOCK
C 701 MZMB 3/16/78
C MEASURE MATERIAL BATCHES
C LP+0 1 RESPONSE POINTER
C 1 1 NUMBER OF MATERIAL TYPES MEASURED
C 2 4 MEASUREMENT SPECIFICATIONS (REPEAT FOR EACH TYPE)

```

Source Function Programs

```

C      INTEGER MATERIAL TYPE CODE
C      MFAN COUNT ERROR
C      MAXIMUM UNIFORM DEVIATION OF COUNT ERROR
C      SYSTEMATIC MASS ERROR
C      MEAN OF RANDOM MASS ERROR
C      STANDARD DEVIATION OF RANDOM MASS ERROR
C      3 3 MEAN TIME TO COMPLETE ALL MEASUREMENTS
C      4 3 MAXIMUM UNIFORM DEVIATION OF MEASUREMENT TIME
C      5 1 TIMING CODE
C      6 1 TIMER FUNCTION
C      7 2 MONITOR FUNCTION
C      8 1 DESTINATION FUNCTION
C      9 1 TRANSPORT FUNCTION
C      10 2 FUNCTIONS NOTIFIED AT CONDITION TIMES
C      11 1 DUMMY PARAMETER USED FOR SAVE LOCATION
C      LV=0 1 ENTRY COUNT
C      1 1 NUMBER OF BATCHES IN FUNCTION
C      2 1 MAXIMUM NUMBER OF BATCHES CONCURRENTLY MEASURED
C      3 3 TIME MAXIMUM FIRST OCCURRED
C      4 1 LN OF MOST RECENT BATCH IN MEASUREMENT
C      LI=0 0
C      1 0
C      2 0
C      3 0
C      4 0
C      SUBROUTINE B701
C      USE SPM
C      BEGINBLOCK [701] [12] [5] [1,1,4,3,3,1,1,2,1,1,2,1,1,1,1,3,1]
C      CREATE
99200  CONTINUE
C      IFF [1V(1).GT.0]
C      PERFORM FREEMATERIAL [4]
C      FIN
C      CALL DESTROY(PVT)
C      RESPOND [-2] [0]
C      PARAMETERS
99400  CONTINUE
C      ERROR [701] MZMB ATTEMPT TO SET INITIAL CONDITIONS)
99500  CONTINUE
C      WR [(A(LB+2),A(LV+2),RA(LV+3)) &
C      [//,9H FUNCTION,I4,1X,8H701/MZMB &
C      /11X,40HMAXIMUM NUMBER OF BATCHES IN MEASUREMENT,110 &
C      /11X,27HTIME MAXIMUM FIRST OCCURRED,11X,E12 4]
C      RESPOND [-5] [0]
99600  CONTINUE
C      IFF [1V(1).GT.0]
C      PERFORM FREEMATERIAL [4]
C      FIN
C      FOR I1 = [1] TO [A(LB+4)]
C      A(LV-1+I1)=A(LI-1+I1)
C      FIN
C      RESPOND [-6] [0]

```

```

DUMP
EVENT
CALL CEC(1,1,MODE)
LN=IA(IPPOINT+1)
LM=IA(LV+3)
TIMER [5] [6]
IFF [1] GT 1)
    CALL SCHED(1,IA(LB+2),1,LN)
ELSE
    PERFORM MEASUREMENT [1]
    TR [1] [70] MZMB MEASUREMENT COMPLETED)
    PERFORM TRANSPORT [2]
    FIN
RESPOND [-8] [0]
INTERACT
CASE [IA(IPPOINT+2) EQ 1]
    IA(LP+1)=IPPOINT
    CALL COPYB(IPPOINT,LN)
    IA(LV+1)=IA(LV+1)+1
    IFF [IV(1) GT IV(2)]
        IA(LV+2)=IV(1)
        IA(LV+3)=TIME
    FIN
    IFF [IV(4) LE 0]
        IA(LN+8)=0
        IA(LN+9)=C
        IA(LV+4)=LN
    ELSE
        IA(IV(4)+9)=LN
        IA(LN+8)=IV(4)
        IA(LN+9)=C
        IA(LV+4)=LN
    FIN
    TIMER [5] [6]
    CALL SCHED(1+RUNIF(RP(3),RP(4)),IA(LB+2),1,LN)
    PERFORM CONDITION [3]
    TR [3] [70] MZMB MEASUREMENT STARTED)
    IPPOINT=IP(1)
    IA(IPPOINT+2)=-1
ORIF [IA(IPPOINT+2) EQ 5]
    IA(IPPOINT+2)=-5
    IA(IPPOINT+3)=0
    IA(IPPOINT+4)=IV(4)
    IA(IPPOINT+5)=IV(1)
    TR [3] [70] MZMB MATERIAL ACQUISITION ATTEMPT)
OTHERWISE
    ERROR [70] MZMB [ILLEGAL INTERACTION]
    FIN
RESPOND [-9] [IPPOINT]
C
PROCEDURE MEASUREMENT [1]
CALL PCOM(ILP(7,1),9,6,0,0,0,0,0,0)

```

Source Function Programs

```

FOR (N) = 11 TO (IP(1))
  CALL MZUR(LM, IP(2), N, IA(IP(POINT)+3), IA(IP(POINT)+4), RA(IP(POINT)+5))
  IA(IP(POINT)+6), PA(IP(POINT)+7))
  FOR (I) = 11 TO (IA(IP(7)+3))
    MFROM=IA(LB+2)
    MTO=IP(7,1)
    ICTYPE=9
    IA(IP(POINT)+2)=6
    SEND
    IFF [IA(IP(POINT)+2) NE -6]
      ERROR (70) MZMB RESPONSE TO MEASUREMENT DATA)
    FIN
  FIN
  CALL FSTORE(IP(POINT))
ENDPROC [1]
C
PROCEDURE TRANSPORT [2]
  IFF (IV(1).EQ 1)
    IA(LV+4)=0
  ELSE
    IFF (IV(4).EQ LN)
      IA(LV+4)=IA(LN+8)
      IFF [IA(LN+8) GT 0]
        IA(IA(LN+8)+9)=0
      FIN
    ELSE
      IFF [IA(LN+8) EQ 0]
        IA(IA(LN+9)+8)=0
      ELSE
        IA(IA(LN+8)+9)=IA(LN+9)
        IA(IA(LN+9)+8)=IA(LN+8)
      FIN
    FIN
  FIN
  PERFORM CONDITION [3]
  IA(LN+8)=0
  IA(LN+9)=0
  IA(LN+4)=IA(LB+2)
  MFROM=IA(LB+2)
  MTO=IP(9)
  ICTYPE=9
  IP(POINT)=LN
  IA(LN+5)=IP(8)
  SEND
  IFF [IA(IP(POINT)+2) NE -1]
    ERROR (70) MZMB MATERIAL NOT REMOVED AT COMPLETION TIME)
  FIN
  CALL FSTORE(IP(POINT))
  IA(LV+1)=IA(LV+1)-1
ENDPROC [2]
C

```

Source Function Programs

```

PROCEDURE CONDITION [3]
  IFF IIP(10).GT.0
  CALL PCOM(IIP(10),9,9,0,IV(4),IV(1),0,0,0,0)
  FOR I1 = 1 TO IIA/IIP(10)+31
    I12=I1/IIP(10,1)
    I12/PE=9
    SE'D
    IFF IIA(IPOINT+2).NE.-9
      ERROR I701 MZMB RESPONSE TO CONDITION NOTICE
    FIN
    CALL FSTORE(IPOINT)
  FIN
FIN
ENDPROC [3]

C
PROCEDURE FREEMATERIAL [4]
  LN=IV(4)
  WHILE (LN.GT.0)
    CALL FRMD(12/LN+31)
    CALL FSTORE(LN)
    LN=IA(LN+8)
  FIN
ENDPROC [4]
ENDBLOCK

C 702 CMBA 3/16/78
COMPUTE MATERIAL BALANCE
  LP+0 1 RESPONSE POINTER
  1 1 NUMBER OF REPORTS
  2 4 REPORT SPECIFICATIONS (REPEAT FOR EACH REPORT)
    INTEGER MATERIAL TYPE CODE
    DATA 1 = OBSERVED 2 = TRUE 3 = ERROR
    VARIABLE 1 = COUNT 2 = MASS
    SAMPLING 1 = EACH INPUT/OUTPUT 2 = REPLICATE
    REPLICATE SAMPLING TIME
    NUMBER HISTOGRAM CELLS
    WIDTH OF HISTOGRAM CELLS
    LOWER LIMIT OF FIRST HISTOGRAM CELL
  3 2 INPUT FUNCTIONS
  4 2 OUTPUT FUNCTIONS
  5 1 - POINTERS TO REPORT DATA
  LV+0 1 ENTRY COUNT
  LI+0 0
  SUBROUTINE B702
  USE SPM
  BEGINBLOCK [702] [6] [1] [1,1,4,2,1,1]
  CREATE
  99200 CONTINUE
  IFF IIP(5).GT.0
    FOR IN = 1 TO IIP(1)
      CALL FSTORE(IAC(IIP(5),IN))
    FIN
    CALL FSTORE(IIP(5))

```


Source Function Programs

```

      FIN
      CALL DESTROY(PVT)
      RESPOND [-2] [0]
      PARAMETERS
      CONTINUE
99400  ERROR [702 CMBA  ATTEMPT TO SET INITIAL CONDITIONS]
99500  CONTINUE
      PERFORM REPORT [1]
      RESPOND [-3] [0]
99600  CONTINUE
      FOR [N] = [1] TO [IP(1)]
        IFF (RAC(IP(2), LIINM(N, 8, 4)) .EQ. 2.)
          CALL SCHED(RAC(IP(2), LIINM(N, 8, 5)), IA(LB+2), N, 0)
        FIN
      FIN
      IA(LV)=0
      IFF [IP(5) .EQ. 0]
        PERFORM CREATEREPORTDATA [2]
      FIN
      IFF [NRUN .EQ. 1]
        PERFORM RESETREPORTDATA [3]
      ELSE
        PERFORM ZEROCURRENTVALUES [4]
      FIN
      RESPOND [-6] [0]
      DUMP
      EVENT
      CALL DEC(1, IP(1), N)
      PERFORM SAMPLECURRENTVALUE [5]
      TR [1] [702 CMBA  REPLICATE SAMPLE TAKEN]
      RESPOND [-8] [0]
      INTERACT
      IFF [IA(:POINT+2) .NE. 6]
        ERROR [702 CMBA  ILLEGAL INTERACTION]
      FIN
      FOR [N] = [1] TO [IP(1)]
        IFF (IFIX(RAC(IP(2), LIINM(N, 8, 1))) .EQ. 1A(IP(POINT+3)))
          PERFORM SCANINPUT/OUTPUTLISTS [6]
          CASE [1 .EQ. 1]
            SIGN=1.
            PERFORM UPDATEVALUE [7]
          ORIF [1 .EQ. 2]
            SIGN=-1.
            PERFORM UPDATEVALUE [7]
          OTHERWISE
            ERROR [702 CMBA  DATA RECEIVED FROM ILLEGAL I/O FUNCTION]
          FIN
          ISAMP=IFIX(RAC(IP(2), LIINM(N, 8, 4)))
          CASE [ISAMP .EQ. 1]
            PERFORM SAMPLECURRENTVALUE [5]
          ORIF [ISAMP .NE. 2]
            ERROR [702 CMBA  ILLEGAL SAMPLING PARAMETER]

```

Source Function Programs

```

      FIN
    FIN
  IA(IPOINT+2)=-6
  TR [3] [702 CMBA MEASUREMENT DATA RECEIVED]
  RESPOND [-9] [IPOINT]

C
PROCEDURE REPORT [1]
FOR [N] = [1] TO [IP(1)]
  LET(LCS(IP(5),N))
  RA(LCS(L,3))=RAC(L,3)/FLOAT(IAC(L,1))
  RA(LCS(L,4))=RAC(L,4)/FLOAT(IAC(L,1))-RAC(L,3) 2
  IFF [RAC(L,4).LT.0.]
    ERROR [702 CMBA NEGATIVE VARIANCE]
  FIN
  RA(LCS(L,4))=SORT(RAC(L,4))
  WR [IAC(L,2) N] 8
  [IH1,8HFUNCTION,14,1X,8H702 CMBA,19X 10HREPORT NO.,13]
  I=RAC(IP(2),LIINM(N,8,1))
  WR [I] [11X,13H MATERIAL TYPE,11B]
  I=RAC(IP(2),LIINM(N,8,2))
  CASE [I.EQ.1]
    WR [ ] [11X,4HDATA,26X,8HOBSERVED]
    ORIF [I.EQ.2]
      WR [ ] [11X,4HDATA,26X,4HTRUE]
    ORIF [I.EQ.3]
      WR [ ] [11X,4HDATA,26X,23HERROR (OBSERVED - TRUE)]
    OTHERWISE
      WR [ ] [11X,26HILLEGAL DATA SPECIFICATION]
  FIN
  I=RAC(IP(2),LIINM(N,8,3))
  CASE [I.EQ.1]
    WR [ ] [11X,8H VARIABLE,22X,5HCOUNT]
    ORIF [I.EQ.2]
      WR [ ] [11X,8H VARIABLE,22X,4HMASS]
    OTHERWISE
      WR [ ] [11X,30HILLEGAL VARIABLE SPECIFICATION]
  FIN
  I=RAC(IP(2),LIINM(N,8,4))
  CASE [I.EQ.1]
    WR [ ] [11X,8HSAMPLING,22X,22HEACH INPUT/OUTPUT TIME]
    ORIF [I.EQ.2]
      WR [ ] [11X,8HSAMPLING,22X,9HREPLICATE]
    OTHERWISE
      WR [ ] [11X,30HILLEGAL SAMPLING SPECIFICATION]
  FIN
  WR [RAC(IP(2),LIINM(N,8,5)) IAC(L,1)] 8
  [11X,23HREPLICATE SAMPLING TIME 5X,E12.4 &
,8X,22HNUMBER OF OBSERVATIONS,8X,110]
  WR [RAC(L,3),RAC(L,4)] 8
  [11X,10HMEAN VALUE,19X,E11.4 &
,8X,18HSTANDARD DEVIATION,12X,E10.4]

```

Source Function Programs

```

WR [RAC(L,5),RAC(L,6)] &
[11X,13HMAXIMUM VALUE,16X,E11.4 &
,8X,21HTIME OF MAXIMUM VALUE,9X,E10.4]
WR [RAC(L,7),RAC(L,8)] &
[11X,13HMINIMUM VALUE,16X,E11.4 &
,8X,21HTIME OF MINIMUM VALUE,9X,E10.4]
[FF [IFIX(RAC(IP(2),L11NM(N,8,6))) .GT. 0]
WR [ ] &
[11X,11HLOWER LIMIT,26X,7HD1STR1-,4X,3HO.0,9X,4HO.25 &
,8X,3HO.5,9X,4HO.75,8X,3H1.0 &
,11X,7HOF CELL,8X,5HCOUNT,6X,7HOF DENSITY,5X,6HBUTTON,6X &
,49H+
X=RAC(IP(2),L11NM(N,8,8))-RAC(IP(2),L11NM(N,8,7))
DIS=0.
DIS(LBUF+30)=1H
FOR IM = 11 TO [IFIX(RAC(IP(2),L11NM(N,8,6)))]
DEN=FLOAT([AC(L,8+M)])/FLOAT([AC(L,1))
DIS=DIS+DEN
X=X+RAC(IP(2),L11NM(N,8,7))
FOR [J] = [31] TO [80]
[AC(LBUF+J)=1H
FIN
K=49. DEN+.999
FOR [J] = [1] TO [K]
[AC(LBUF+30+J)=1H
FIN
K=49. DIS
[AC(LBUF+31+K)=1H+
WR [X, [AC(L,8+M), DEN, DIS, ([AC(LBUF+J), J=30,80)] &
[10X,E11.4,110,5X,F6.4,6X,F6.4,5X,5[A]]
FIN
FIN
ENDPROC [1]
C
PROCEDURE CREATEREPORTDATA [2]
CALL ASTORE([AC(LP+5),IP(1)])
FOR IN = [1] TO [IP(1)]
CALL ASTORE([AC(LCS(IP(5),N)),8+IFIX(RAC(IP(2),L11NM(N,8,6)))]
FIN
ENDPROC [2]
C
PROCEDURE RESETREPORTDATA [3]
[IF [IP(5),LE.0]
ERROR [702 CMBA NO REPORT DATA TO RESET]
FIN
FOR IN = [1] TO [IP(1)]
CALL ZIAC([AC(IP(5),N)])
FOR [I] = [2] TO [8]
RA(LCS([AC(IP(5),N),I))=0.
FIN
RA(LCS([AC(IP(5),N),7))=1.0E+50

```

Source Function Programs

```

      FIN
      ENDPROC [3]
C
      PROCEDURE ZEROCURRENTVALUES [4]
      FOR (N) = [1] TO [IP(1)]
        L=IAC(IP(5),N)
        RA(LCS(L,2))=0.
      FIN
      ENDPROC [4]
C
      PROCEDURE SCANINPUT/OUTPUTLISTS [6]
      K=1
      WHILE (K.LE. IAC(IP(3)+3).AND. IAC(IP(3),K+2).NE. MFROM)
        K=K+1
      FIN
      IFF (K.LE. IAC(IP(3)+3))
        I=1
      ELSE
        K=1
        WHILE (K.LE. IAC(IP(4)+3).AND. IAC(IP(4),K+2).NE. MFROM)
          K=K+1
        FIN
        IFF (K.LE. IAC(IP(4)+3))
          I=2
        ELSE
          I=0
        FIN
      FIN
      ENDPROC [6]
C
      PROCEDURE SAMPLECURRENTVALUE [5]
      L=IAC(IP(5),N)
      IAC(LCS(L,1))=IAC(L,1)+1
      CV=RAC(L,2)
      RA(LCS(L,3))=RAC(L,3)+CV
      RA(LCS(L,4))=RAC(L,4)+CV CV
      IFF (CV.GT. RAC(L,5))
        RA(LCS(L,5))=CV
        RA(LCS(L,6))=TIME
      FIN
      IFF (CV.LT. RAC(L,7))
        RA(LCS(L,7))=CV
        RA(LCS(L,8))=TIME
      FIN
      IFF (IFIX(RAC(IP(2),LIINM(N,8,6))).GT.0)
        ICELL=IFIX((CV-RAC(IP(2),LIINM(N,8,8)))/
1      RAC(IP(2),LIINM(N,8,7)))+1
        CASE (ICELL.LT.1)
          ICELL=1
        ORIF (ICELL.GT. IFIX(RAC(IP(2),LIINM(N,8,6))))
          ICELL=IFIX(RAC(IP(2),LIINM(N,8,6)))
        FIN

```

Source Function Programs

```

        ICELL=LCS(L,8+ICELL)
        IA(ICELL)=IA(ICELL)+1
        FIN
    ENDPROC [5]
C
    PROCEDURE UPDATEVALUE [7]
    IDATA=IFIX(RAC(IP(2),LINM(N,8,2)))
    IVAR=IFIX(RAC(IP(2),LINM(N,8,3)))
    L=IAC(IP(5),N)
    CASE [IVAR.EQ.1]
        CASE [IDATA.EQ.1]
            RA(LCS(L,2))=RA(LCS(L,2))+SIGN FLOAT(IA(IPPOINT+4))
        ORIF [IDATA.EQ.2]
            RA(LCS(L,2))=RA(LCS(L,2))+SIGN FLOAT(IA(IPPOINT+6))
        ORIF [IDATA.EQ.3]
            RA(LCS(L,2))=RA(LCS(L,2))+SIGN FLOAT(IA(IPPOINT+4)
            -IA(IPPOINT+6))
        1 OTHERWISE
            ERROR [702 CMBA ILLEGAL REPORT DATA PARAMETER]
            FIN
        ORIF [IVAR.EQ.2]
            CASE [IDATA.EQ.1]
                RA(LCS(L,2))=RA(LCS(L,2))+SIGN RA(IPPOINT+5)
            ORIF [IDATA.EQ.2]
                RA(LCS(L,2))=RA(LCS(L,2))+SIGN RA(IPPOINT+7)
            ORIF [IDATA.EQ.3]
                RA(LCS(L,2))=RA(LCS(L,2))+SIGN (RA(IPPOINT+5)-RA(IPPOINT+7))
            OTHERWISE
                ERROR [702 CMBA ILLEGAL REPORT DATA PARAMETER]
                FIN
        OTHERWISE
            ERROR [702 CMBA ILLEGAL REPORT VARIABLE PARAMETER]
            FIN
    ENDPROC [7]
    ENDBLOCK
C 703 MNTR 3/16/78
C MONITOR STIMULUS TYPE 1
C LP+0 1 RESPONSE POINTER
C 1 4 SENSITIVITY CHARACTERISTICS PAIRS
C 2 4 TIME DELAY DISTRIBUTION TYPE AND STATISTICS
C 3 1 MODE: 1 = RANDOM INTERVAL 2 = STIMULUS TRACKING 3 = LATCHING
C 4 4 DISTRIBUTION TYPE AND STATISTICS OF RANDOM COMPONENT OF
C DETECTION SIGNAL INTERVAL
C 5 1 SIGNAL INDEX
C 6 3 DETECTION SIGNAL AMPLITUDE
C 7 3 DETECTION FUNCTIONS
C 8 3 MEAN TIME BETWEEN SPURIOUS OUTPUTS
C LV+0 1 ENTRY COUNT
C LI+0 1 DETECTION SIGNAL STATE
C 1 0
C 1 0
    SUBROUTINE B703

```

Source Function Programs

```

USE SPM
BEGINBLOCK [703] [9] [2] [1,4,4,1,4,1,3,2,3,1,1]
DATA [X,U/1,0.7]
CREATE
DESTROY
PARAMETERS
CONTINUE
99400 ERROR [703 MNTR ATTEMPT TO SET INITIAL CONDITIONS]
REPORT [703 MNTR]
99600 CONTINUE
IF [RP(8).GT.0.]
CALL URN(X,U)
T=-RP(8) ALOG(U)
CALL SCHED(T,[A(LB+2),3,0])
FIN
FOR [I] = [1] TO [A(LB+4)]
[A(LV+1)]=[A(LI-1+)]
FIN
RESPOND [-6] [0]
DUMP
EVENT
CALL CEC(1,2,KODE)
CASE [KODE.EQ.1]
O=RP(6)
PERFORM SIGNAL [1]
[A(LV+1)]=1
ORIF [KODE.EQ.2]
O=0
PERFORM SIGNAL [1]
[A(LV+1)]=0
ORIF [KODE.EQ.3]
O=RP(6)
PERFORM SIGNAL [1]
[A(LV+1)]=1
TR [1] [703 MNTR SPURIOUS OUTPUT]
FIN
RESPOND [-8] [0]
INTERACT
CASE [A(IPOINT+2).EQ.7]
L=[A(IPOINT+3)]
I=[A(IPOINT+4)]
IF [FIX(RAC(L,I+2)).NE.1]
ERROR [703 MNTR ILLEGAL STIMULUS TYPE]
FIN
I=I+1
CALL SAMPL(L,I,T1)
I=I+NSTATS(IF[X(RAC(L,I+2))]+1)
CALL SAMPL(L,I,T2)
I=I+NSTATS(IF[X(RAC(L,I+2))]+1)
CALL SAMPL(L,I,A)
N=[A(IP(1)+3)-1]
WHILE [RLP(1,N).GT.A.AND.N.GE.1]

```

Source Function Programs

```

      N=N-2
      FIN
      IFF {N.GT.0}
      PD=RLP(1,N+1)
      ELSE
      PD=0.
      FIN
      CALL URN(IX,U)
      IFF {U.LE.PD}
      CALL SAMPL(IP(2),1,T3)
      T=TIME+T1+T3
      CALL SCHED(T,IA(LB+2),1,0)
      CASE {IP(3).EQ.1}
      CALL SAMPL(IP(4),1,T4)
      T=T+T4
      CALL SCHED(T,IA(LB+2),2,0)
      ORIF {IP(3).EQ.2}
      CALL SAMPL(IP(4),1,T4)
      T=T+T2+T4
      IFF {T.LT.0.}
      T=0.
      FIN
      CALL SCHED(T,IA(LB+2),2,0)
      ORIF {IP(3).NE.3}
      ERROR [703 MNTR ILLEGAL MODE]
      FIN
      FIN
      IA(IP(POINT+2))=-7
      ORIF {IA(IP(POINT+2)).EQ.0.AND. IA(IP(POINT+3)).EQ.1.AND. &
      IA(IP(POINT+4)).EQ.0}
      IFF {IV(1).EQ.1}
      Q=0
      PERFORM SIGNAL [1]
      IA(LV+1)=0
      ELSE
      ERROR [703 MNTR RESET]
      FIN
      IA(IP(POINT+2))=-8
      OTHERWISE
      ERROR [703 MNTR ILLEGAL INTERACTION]
      FIN
      RESPOND [-9] [IP(POINT)]
C
      PROCEDURE SIGNAL [1]
      CALL PCOM(ILP(7,1),9,10,1,IP(5),0,0,0,0,0)
      RA(IP(POINT+5))=0
      FOR {K} = [1] TO {IA(IP(7)+3)}
      ICTYPE=9
      MTO=ILP(7,K)
      SEND
      IFF {IA(IP(POINT+2)).NE.-10}
      ERROR [703 MNTR MONITOR SIGNAL NOT ACCEPTED]

```

Source Function Programs

```

      FIN
      CALL FSTORE(IP(POINT))
      ENDPROC [1]
      ENDBLOCK
C 801 DTBL 3/16/78
DECISION TABLE
LP+0 1 RESPONSE POINTER
      2 INPUT SIGNAL INDEXES
      3 1 NUMBER OF CONDITION ROWS
      4 4 CONDITION TRIPLETS
      5 2 OUTPUT SIGNAL INDEXES
      6 1 NUMBER OF ACTION ROWS
      7 4 ACTION PAIRS
      8 1 NUMBER OF DECISION RULES
      9 2 DECISION RULES
      10 2 FUNCTIONS TO RECEIVE OUTPUT SIGNALS
      11 1 NUMBER OF REPORTS
      11 4 REPORT SPECIFICATIONS (REPEAT FOR EACH REPORT)
          SIGNAL INDEX
          SAMPLING TIME
          NUMBER OF CELLS
          WIDTH OF CELLS
          LOWER LIMIT OF FIRST CELL
      12 1 - POINTERS TO REPORT DATA
LV+0 1 ENTRY COUNT
      1 4 INPUT SIGNALS
      2 4 OUTPUT SIGNALS
      3 3 TIME OF SCHEDULED DECISION EVENT
L1+0 0
      1 0
      2 0
      3 -1
      SUBROUTINE B801
      USE SPM
      BEGINBLOCK [801] [13] [4] [1,2,1,4,2,1,4,1,2,2,1,4,1,1,4,4,3]
      CREATE
99200 CONTINUE
      IF IP(12).GT.0]
      FOR [N] = [1] TO [IP(10)]
      CALL FSTORE(IAC(IP(12),N))
      FIN
      CALL FSTORE(IP(12))
      FIN
      CALL DESTRO(IPVT)
      RESPOND [-2] [0]
      PARAMETERS
      VARIABLES
99500 CONTINUE
      PERFORM REPORT [1]
      RESPOND [-5] [0]
99600 CONTINUE

```


Source Function Programs

```

IA(LV+0)=0
IFF [IP(1).LE.0]
  ERROR [801] DTBL  UNDEFINED INPUT SIGNAL INDEXES AT IC TIME]
ELSE
  IFF [IV(1).LE.0]
    CALL ASTORE(IA(LV+1),IA(IP(1)+3)+2)
    IA(IV(1)+2)=IA(IV(1)+1)-2
    IA(IV(1)+3)=IA(IP(1)+3)
  FIN
  IFF [IC(1).LE.0]
    ERROR [801] DTBL  UNDEFINED INPUT IC VALUES AT IC TIME]
  ELSE
    FOR [I] = [1] TO [IA(IP(1)+3)]
      RA(LCS(IV(1),I+2))=RA(LCS(IC(1),I+2))
    FIN
  FIN
  IFF [IP(4).LE.0]
    ERROR [801] DTBL  UNDEFINED OUTPUT SIGNAL INDEXES AT IC TIME]
  ELSE
    IFF [IV(2).LE.0]
      CALL ASTORE(IA(LV+2),IA(IP(4)+3)+2)
      IA(IV(2)+2)=IA(IV(2)+1)-2
      IA(IV(2)+3)=IA(IP(4)+3)
    FIN
    IFF [IC(2).LE.0]
      ERROR [801] DTBL  UNDEFINED OUTPUT IC VALUES AT IC TIME]
    ELSE
      FOR [I] = [1] TO [IA(IP(4)+3)]
        RA(LCS(IV(2),I+2))=RA(LCS(IC(2),I+2))
      FIN
    FIN
    FOR [N] = [1] TO [IP(10)]
      CALL SCHED(RAC(IP(1)),LIINM(N,5,2),IA(LB+2),N,0)
    FIN
    IFF [IP(12).EQ.0]
      PERFORM CREATEREPORTDATA [2]
    FIN
    IFF [INRUN.EQ.1]
      PERFORM RESETREPORTDATA [3]
    ELSE
      PERFORM ZEROCURRENTVALUES [4]
    FIN
    RA(LV+3)=-1.
    RESPOND [-6] [0]
    DUMP
    EVENT
    CALL CEC(-1,IP(10),N)
    IFF [IN.EQ.-1]
      PERFORM DECISION [7]
    IFF [NEWSIG.EQ.1]

```

Source Function Programs

```

        PERFORM TRANSMITOUTPUTSIGNALS [8]
    FIN
ELSE
    PERFORM SAMPLECURRENTVALUES [5]
    FIN
RESPOND [-8] [0]
INTERACT
    IFF [IA(IPPOINT+2).NE.10]
        ERROR [801 DTBL ILLEGAL INTERACTION]
    FIN
    PERFORM UPDATEINPUTSIGNALS [6]
    IFF [IRV(3).LT.TIME]
        CALL SCHED(TIME,1A(LB+2),-1,0)
        RA(LV+3)=TIME
    FIN
    TR [3] [801 DTBL CONTROL SIGNAL RECEIVED]
    IA(IPPOINT+2)=-10
    RESPOND [-9] [IPPOINT]
C
PROCEDURE REPORT [1]
FOR [N] = [1] TO [IP[10]]
    L=1A(LCS(IP[12],N))
    RA(LCS(L,3))=RAC(L,3)/FLOAT(IA(L,1))
    RA(LCS(L,4))=RAC(L,4)/FLOAT(IA(L,1))-RAC(L,3) 2
    IFF [RAC(L,4).LT.0.]
        ERROR [801 DTBL NEGATIVE VARIANCE]
    FIN
    RA(LCS(L,4))=SORT(RAC(L,4))
    WR [IA(LB+2),N] &
    [11] 8HFUNCTION, [4, 1X, 8H801 DTBL, 19X, 10HREPORT NO., 13]
    T=RAC(IP[11], LIINM(N,5,1))
    WR [1] [11X, 14HCONTROL SIGNAL, 117]
    T=RAC(IP[11], LIINM(N,5,2))
    WR [1] [11X, 13HSAMPLING TIME, 15X, E12.4]
    WR [RAC(L,3), RAC(L,4)] &
    [11X, 10HMEAN VALUE, 19X, E11.4 &
    8X, 18HSTANDARD DEVIATION, 12X, E10.4]
    WR [RAC(L,5), RAC(L,6)] &
    [11X, 13HMAXIMUM VALUE, 16X, E11.4 &
    8X, 21HTIME OF MAXIMUM VALUE, 9X, E10.4]
    WR [RAC(L,7), RAC(L,8)] &
    [11X, 13HMINIMUM VALUE, 16X, E11.4 &
    8X, 21HTIME OF MINIMUM VALUE, 9X, E10.4]
    WR [IA(L,1)] &
    [11X, 22HNUMBER OF OBSERVATIONS, 8X, 110]
    IFF [LIFIX(RAC(IP[11], LIINM(N,5,3))).GT.0]
        WR [1] &
        [11X, 11HLOWER LIMIT, 26X, 7HDISTR1-, 4X, 3H0.0, 9X, 4H0.25 &
        8X, 3H0.5, 9X, 4H0.75, 8X, 3H1.0 &
        11X, 7HOF CELL, 8X, 5HCOUNT, 5X, 7HDENSITY, 5X, 6HBUTION, 6X &
        49H+
        +
        +
        X=RAC(IP[11], LIINM(N,5,5))-RAC(IP[11], LIINM(N,5,4)) +]

```

Source Function Programs

```

DIS=0.
IA(LBUF+30)=1H
FOR (M) = 11 TO (IFIX(RAC(IP(11),L1NM(N,5,3)))
    DEN=FLOAT(IAC(L,8*M))/FLOAT(IAC(L,1))
    DIS=DIS+DEN
    X=X+RAC(IP(11),L1NM(N,5,4))
    FOR (J) = 31 TO 801
        IA(LBUF+J)=1H
    FIN
    K=49. DEN+.999
    FOR (J) = 11 TO (K)
        IA(LBUF+30+J)=1H
    FIN
    K=49. DIS+.999
    IA(LBUF+30+K)=1H+
    WR (X,IAC(L,8*M),DEN,DIS,(IA(LBUF+J),J=30,801) &
    10X,E11.4,110,5X,F6.4,6X,F6.4,5X,51A1)
    FIN
FIN
ENDPROC [1]
C
PROCEDURE CREATEREPORTDATA [2]
CALL ASTORE(IA(LP+12),IP(10))
FOR (N) = 11 TO IIP(10)
    CALL ASTORE(IA(LCS(IP(12),N)),8*IFIX(RAC(IP(11),L1NM(N,5,3)))
FIN
ENDPROC [2]
C
PROCEDURE RESETREPORTDATA [3]
IF IIP(12).LE.0]
    ERROR 1801 DTBL NO REPORT DATA TO RESET]
FIN
FOR (N) = 11 TO IIP(10)
    CALL ZIAC(IAC(IP(12),N))
    FOR (I) = 12 TO 18]
        RA(LCS(IAC(IP(12),N),I))=0.
    FIN
    RA(LCS(IAC(IP(12),N),7))=1.0E+50
FIN
ENDPROC [3]
C
PROCEDURE ZEROCURRENTVALUES [4]
FOR (N) = 11 TO IIP(10)
    L=IAC(IP(12),N)
    RA(LCS(L,2))=0.
FIN
ENDPROC [4]
C
PROCEDURE SAMPLECURRENT VALUES [5]
L=IAC(IP(12),N)
PERFORM GETCURRENTVALUEV [15]

```

Source Function Programs

```

IA(LCS(L,1))=IAC(L,1)+1
CV=RAC(L,2)
RA(LCS(L,3))=RAC(L,3)+CV
RA(LCS(L,4))=RAC(L,4)+CV CV
IFF [CV.GT.RAC(L,5)]
    RA(LCS(L,6))=CV
    RA(LCS(L,6))=TIME
FIN
IFF [CV.LT.RAC(L,7)]
    RA(LCS(L,7))=CV
    RA(LCS(L,8))=TIME
FIN
IFF [IFIX(RAC(IP(1)),L1NM(N,5,3)).GT.0]
    ICELL=IFIX((CV-RAC(IP(1)),L1NM(N,5,5)))/
1    RAC(IP(1),L1NM(N,5,4))+1
    CASE [ICELL.LT.1]
        ICELL=1
    ORIF [ICELL.GT.IFIX(RAC(IP(1)),L1NM(N,5,3))]
        ICELL=IFIX(RAC(IP(1)),L1NM(N,5,3))
    FIN
    ICELL=LCS(L,8+ICELL)
    IA(ICELL)=IA(ICELL)+1
    FIN
ENDPROC [5]
C
PROCEDURE UPDATEINPUTSIGNALS [6]
L=0
N=0
IFF [IA(IP(1)+3).GT.0]
    M=IA(IP(1)+3)
    I=1
    IFF [IA(IP(1)+4).GT.0]
        WHILE [I.LE.M.AND.ILP(1,1).NE.1A(IP(1)+4)]
            I=I+1
        FIN
        IFF [I.LE.M]
            RA(LCS(IV(1),2+1))=RA(IP(1)+5)
            L=1
            N=N+1
        FIN
    FIN
    IFF [IA(IP(1)+6).GT.0]
        I=1
        WHILE [I.LE.M.AND.ILP(1,1).NE.1A(IP(1)+6)]
            I=I+1
        FIN
        IFF [I.LE.M]
            RA(LCS(IV(1),2+1))=RA(IP(1)+7)
            L=1
            N=N+1
        FIN
    FIN
    FIN

```

Source Function Programs

```

      IFF [N.LT.1A(IPPOINT+3)]
        FOR [K] = [1] TO [1A(1A(IPPOINT+8)+3)]
          [I]=1
          WHILE [I.LE.M.AND.1LP(1,1).NE.1AC(1A(IPPOINT+8),K+2)]
            I=I+1
          FIN
          IFF [I.LE.M]
            RA(LCS(IV(1),2+1))=RAC(1A(IPPOINT+9),K+2)
            L=1
            N=N+1
          FIN
        FIN
      FIN
    IFF [L.EQ.0]
      ERROR [801 DTBL NO INPUT SIGNALS IN THOSE RECEIVED]
    FIN
    IFF [N.NE.1A(IPPOINT+3)]
      ERROR [801 DTBL NUMBER CONTROL SIGNALS TRANSMITTED]
    FIN
  ELSE
    ERROR [801 DTBL ZERO CONTROL SIGNALS TRANSMITTED]
  FIN
  LIM=1A(IV(1)+3)
  TR [1] [INPUT] &
  [(RAC(IV(1),1+2),1=1,LIM)] [1X,5E12.4]
ENDPROC [6]

C
PROCEDURE DECISION [7]
  J=0
  JSAT=0
  NEWSIG=0
  WHILE [J.LT.1P(7).AND.JSAT.EQ.0]
    J=J+1
    PERFORM TESTCONDITIONSJ [9]
  FIN
  IFF [JSAT.EQ.1]
    PERFORM ACTIONSJ [10]
  ELSE
    ERROR [801 DTBL NO CONDITION SATISFIED]
  FIN
ENDPROC [7]

C
PROCEDURE TRANSMITOUTPUTSIGNALS [8]
  LIM=1A(IV(2)+3)
  TR [1] [OUTPUT] &
  [(RAC(IV(2),1+2),1=1,LIM)] [1X,5E12.4]
  IFF [1P(9).GT.0]
    CALL PCOM(1LP(9,1),9,10,1A(IP(4)+3),0,0,0,0,1P(4),1V(2))
    FOR [I] = [1] TO [1A(IP(9)+3)]
      MFROM=1A(LB+2)
      MTO=1LP(9,I)
      ICTYPE=9
    
```

Source Function Programs

```

        IA(IPPOINT+2)=10
        SEND
        IFF [IA(IPPOINT+2).NE.-10]
            ERROR [80] DTBL RESPONSE TO OUTPUT SIGNALS]
        FIN
    FIN
    CALL FSTORE(IPPOINT)
    FIN
ENDPROC [8]
C
    PROCEDURE TESTCONDITIONSJ [9]
    I=0
    JSAT=1
    WHILE [I.LT.IP(2).AND.JSAT.EQ.1]
        I=I+1
        PERFORM TESTCONDITIONIJ [11]
    FIN
ENDPROC [9]
C
    PROCEDURE ACTIONSJ [10]
    N=(J-1) (IP(2)+IP(5))+IP(2)
    FOR [I] = [1] TO [IP(5)]
        IFF [ILP(8,N+1).EQ.1]
            PERFORM ACTIONI [12]
        FIN
    FIN
    TR [I] [RULE] [J] [110]
ENDPROC [10]
C
    PROCEDURE TESTCONDITIONIJ [11]
    K=(J-1) (IP(2)+IP(5))+1
    CASE [ILP(8,K).EQ.-1]
        PERFORM CONDITIONI [13]
        IFF [ICON.NE.-1]
            JSAT=0
        FIN
    ORIF [ILP(8,K).EQ.1]
        PERFORM CONDITIONI [13]
        IFF [ICON.NE.1]
            JSAT=0
        FIN
    ORIF [ILP(8,K).NE.0]
        ERROR [80] DTBL ILLEGAL CONDITION ENTRY]
    FIN
ENDPROC [11]
C
    PROCEDURE ACTIONI [12]
    L=(I-1) 2+1
    INDX=IFIX(RLP(6,L))
    CASE [INDX.GT.0]
        K=1
        WHILE [K.LE.IA(IP(4)+3).AND.INDX.NE.ILP(4,K)]

```

Source Function Programs

```

      K=K+1
      FIN
      IFF [K.GT.1A(IP(4)+3)]
      ERROR [801 DTBL ILLEGAL SIGNAL INDEX IN ASSIGNMENT ACTION]
    ELSE
      RA(LCS(IV(2),K+2))=RLP(6,L+1)
      NEWSIG=1
      FIN
    ORIF [INDX.LT.0]
      CALL PCOM(-INDX,9,11,0,0,0,0,0,0)
      TR [1] [START] [MTO] [110]
      SEND
      IFF [1A(IPOINT+2).NE.-11]
      ERROR [801 DTBL RESPONSE TO CONTROL MESSAGE]
      FIN
      CALL FSTORE(IPOINT)
    OTHERWISE
      ERROR [801 DTBL ILLEGAL ZERO VALUE IN ACTION PAIR]
      FIN
    ENDPROC [12]

C
    PR 'CEDURE CONDITION: [13]
    L=(I-1)/3+1
    INDX=FIX(RLP(3,L))
    PERFORM GETINPUTSIGNALVALUEV [14]
    SI=V
    L=L+1
    JOP=FIX(RLP(3,L))
    L=L+1
    CASE [JOP.LE.6]
      INDX=FIX(RLP(3,L))
      PERFORM GETINPUTSIGNALVALUEV [14]
      SK=V
    ORIF [JOP.LE.12]
      SK=RLP(3,L)
    OTHERWISE
      ERROR [801 DTBL ILLEGAL RELATIONAL OPERATOR IN CONDITION]
      FIN
    IFF [JOP.GT.6]
      JOP=JOP-6
      FIN
    CASE [JOP.LE.3]
      CASE [JOP.EQ.1]
        IFF [SI.EQ.SK]
          CON=1
        ELSE
          CON=-1
        FIN
      ORIF [JOP.EQ.2]
        IFF [SI.LT.SK]
          CON=1
        ELSE

```

Source Function Programs

```

        ICON=-1
        FIN
    ORIF [JOP.EQ.3]
        IFF [S1.GT.SK]
            ICON=1
        ELSE
            ICON=-1
        FIN
    FIN
    ORIF [JOP.LE.6]
        CASE [JOP.EQ.4]
            IFF [S1.NE.SK]
                ICON=1
            ELSE
                ICON=-1
            FIN
        ORIF [JOP.EQ.5]
            IFF [S1.LE.SK]
                ICON=1
            ELSE
                ICON=-1
            FIN
        ORIF [JOP.EQ.6]
            IFF [S1.GE.SK]
                ICON=1
            ELSE
                ICON=-1
            FIN
        FIN
    FIN
ENDPROC [13]
C
PROCEDURE GETINPUTSIGNALVALUEV [14]
K=1
WHILE [K.LE.1A(IP(1)+3).AND.INDX.NE.ILP(1,K)]
    K=K+1
    FIN
    IFF [K.GT.1A(IP(1)+3)]
        ERROR [801 DTBL SIGNAL OF CONDITION ENTRY NOT AN INPUT]
    ELSE
        V=RLV(1,K)
    FIN
ENDPROC [14]
C
PROCEDURE GETCURRENTVALUEV [15]
K=1
WHILE [K.LE.1A(IP(4)+3).AND.ILP(4,K).NE. &
    IF(XTRAC(IP(1),LINM(N,5,1)))]
    K=K+1
    FIN
    IFF [K.GT.1A(IP(4)+3)]

```


Source Function Programs

```

        ERROR [80] DTBL SIGNAL OF REPORT SPECIFICATION NOT AN OUTPUT]
      ELSE
        RA(LCS(L,2))=RLV(2,K)
      FIN
    ENDPROC [15]
  ENDBLOCK
C 901 TRCN 3/16/78
C TRACE CONTROL
C LP+0 1 RESPONSE POINTER
C LV+0 1 ENTRY COUNT
C LI+0 0
  SUBROUTINE B901
    USE SPM
    BEGINBLOCK [90] [1] [1] [1,1]
    CREATE
    DESTROY
    PARAMETERS
99400 CONTINUE
    ERROR [90] TRCN ATTEMPT TO SET INITIAL CONDITIONS]
    REPORT [90] TRCN]
99600 CONTINUE
    CALL SCHED(0,1A(LB+2),1,0)
    RESPOND [-6] [0]
    DUMP
    EVENT
    CALL CFC(1,1,KODE)
    WR [U:59] [TIME] [1X,E10.4,2X,11HNEW TRACE?]
    RD [U:59] [N] [1]
    WHILE [N.LT.0.OR.N.GT.6]
      WR [U:59] [ ] [20HILLEGAL TRACE VALUE]
      RD [U:59] [N] [1]
    FIN
    ITRACE=N
    WR [U:59] [TRACE] [1,3HITRACE IS NOW,13]
    WR [U:59] [ ] [1X,24HNEXT TRACE CONTROL TIME?]
    RD [U:59] [TNEXT] IF80.0]
    WHILE [TNEXT.LE.TIME]
      WR [U:59] [TIME] [1X,11HTIME IS NOW,E12.4]
      RD [U:59] [TNEXT] IF80.0]
    FIN
    CALL SCHED(TNEXT,1A(LB+2),1,0)
    WR [U:59] [TNEXT] [1X,23HNEXT TRACE CONTROL TIME,E12.4]
    RESPOND [-8] [0]
    INTERACT
    ERROR [90] TRCN ILLEGAL INTERACTION]
    RESPOND [-9] [0]
  ENDBLOCK
C 902 RMDS 3/16/78
C REPORT MATERIAL DIVERSION STATISTICS
C LP+0 1 RESPONSE POINTER
C 1 2 MAGA FUNCTIONS TABULATED
C 2 4 TIME REPORT SPECIFICATION

```

Sub Function Programs

```

C      NUMBER OF CELLS
C      WIDTH OF CELLS
C      LOWER LIMIT OF FIRST CELL
C      3 4 DIVERTED MATERIAL REPORT SPECIFICATION
C      NUMBER OF CELLS
C      WIDTH OF CELLS
C      LOWER LIMIT OF FIRST CELL
C      4 1 - TIME REPORT DATA
C      5 1 - MATERIAL REPORT DATA
C      LV=0 1 ENTRY COUNT
C      LI=0 0
C      SUBROUTINE B902
C      USE SPM
C      BEGINBLOCK [902] [6] [1] [1,2,4,4,1,1,1]
C      CREATE
99200 CONTINUE
      IFF [IP(4).GT.0]
        CALL FSTORE(IP(4))
      FIN
      IFF [IP(5).GT.0]
        CALL FSTORE(IP(5))
      FIN
      CALL DESTRO(IPVT)
      RESPOND [-2] [0]
      PARAMETERS
99400 CONTINUE
      ERROR [50] INDA ATTEMPT TO SET INITIAL CONDITIONS]
99500 CONTINUE
      PERFORM REPORT [1]
      RESPOND [-5] [0]
99600 CONTINUE
      IA(LV)=0
      IFF [IP(4).EQ.0]
        CALL ASTORE(IA(LP+4),8+IFIX(RLP(2,1)))
      FIN
      IFF [IP(5).EQ.0]
        CALL ASTORE(IA(LP+5),8+IFIX(RLP(3,1)))
      FIN
      IFF [NRUN.EQ.1]
        PERFORM RESETREPORTDATA [3]
      ELSE
        PERFORM ZEROCURRENTVALUES [4]
      FIN
      RESPOND [-6] [0]
      DUMP
      EVENT
      CASE [IA(IPOINT).GT.0]
        I=IA(IPOINT)
        L=IA(IPOINT+1)
        INXT=1+2+NSTATS(IFIX(RAC(L,1+3)))
        IFF [IFIX(RAC(L,INXT+2)).GT.0.AND.INXT.LT.IA(L+3)]
          CALL SAMPL(L,INXT+1,TD)

```

Source Function Programs

```

CALL SCHED(TIME+TD,IFIX(RAC(L,INXT+2)),INXT,L)
FIN
PERFORM SAMPLE {5}
OTHERWISE
  ERROR {902 RMDS ILLEGAL EVENT CODE}
  FIN
RESPOND [-9] {0}
INTERACT
  ERROR {50} INDA ILLEGAL INTERACTION}
RESPOND [-9] {0}

C
PROCEDURE REPORT {1}
  IFF {IAC(IPI4),1}.GT.0.}
  FOR {N} = {1} TO {2}
    L=IPI3+N
    RA(LCS(L,3))=RAC(L,3)/FLOAT(IAC(L,1))
    RA(LCS(L,4))=RAC(L,4)/FLOAT(IAC(L,1))-RAC(L,3) 2
    IFF {RAC(L,4)}.GT.0.}
    RA(LCS(L,4))=SORT(RAC(L,4))
  ELSE
    RA(LCS(L,4))=0.
  FIN
  WR {JA(LB+2)} &
  {1H,8HFUNCTION,14,1X,8H902 RMDS}
  CASE {N.EQ.1}
    WR { } {11X,24HSEQUENCE COMPLETION TIME}
  ORIF {N.EQ.2}
    WR { } {11X,17HDIVERTED MATERIAL}
  FIN
  WR {RAC(L,3),RAC(L,4)} &
  {11X,10HMEAN VALUE,19X,E11.4 &
  8X,18HSTANDARD DEVIATION,12X,E10.4}
  CASE {N.EQ.1}
    WR {RAC(L,5),RAC(L,7)} &
    {11X,13HMAXIMUM VALUE,16X,E11.4/ &
    11X,13HMINIMUM VALUE,16X,E11.4}
  ORIF {N.EQ.2}
    WR {RAC(L,5),RAC(L,6)} &
    {11X,13HMAXIMUM VALUE,16X,E11.4 &
    8X,21HTIME OF MAXIMUM VALUE,9X,E10.4}
    WR {RAC(L,7),RAC(L,8)} &
    {11X,13HMINIMUM VALUE,16X,E11.4 &
    8X,21HTIME OF MINIMUM VALUE,9X,E10.41}
  FIN
  WR {IAC(L,1)} &
  {11X,22HNUMBER OF OBSERVATIONS,8X,110}
  CASE {N.EQ.1.AND.IF{X(RLP(2,1))}.GT.0}
    PERFORM PRINTHIST {2}
  ORIF {N.EQ.2.AND.IF{X(RLP(3,1))}.GT.0}
    PERFORM PRINTHIST {2}
  FIN
FIN

```

Source Function Programs

```

ELSE
  WR [IA(LB+2)] &
  [/9H FUNCTION,14,1X,8H902 RMDS &
  /11X,15HNO OBSERVATIONS]
  FIN
ENDPROC [1]

C
PROCEDURE PRINTHIST [2]
  WR [1] &
  [/11X,11HLOWER LIMIT,26X,7HDISTR1-,4X,3H0.0,9X,4H0.25 &
  8X,3H0.5,9X,4H0.75,8X,3H1.0 &
  /11X,7HCF CELL,8X,5HCOUNT,5X,7HDENSITY,5X,6HBUTION,6X &
  49H+]
  CASE [N EQ.1]
    X=RLP(2,3)-RLP(2,2)
  CRIF [N EQ.2]
    X=RLP(3,3)-RLP(3,2)
  FIN
  DIS=0
  IA(LBUF+30)=1H
  FOR [M] = [1] TO [IFIX(RLP(1+N,1))]
    DEN=FLOAT(IAC(L,8+M))/FLOAT(IAC(L,1))
    DIS=DIS+DEN
    X=X+RLP(1+N,2)
    FOR [J] = [1] TO [80]
      IA(LBUF+J)=1H
    FIN
    K=49, DEN+.999
    FOR [J] = [1] TO [K]
      IA(LBUF+30+J)=1H
    FIN
    K=49, DIS+.999
    IA(LBUF+30+K)=1H+
  WR [X,IAC(L,8+M),DEN,DIS,([IA(LBUF+J),J=30,80)] &
  /10X,E11.4,110,5X,F6.4,6X,F6.4,5X,51A1]
  FIN
ENDPROC [2]

C
PROCEDURE RESETREPORTDATA [3]
  CALL ZIAC(IP(4))
  CALL ZIAC(IP(5))
  FOR [I] = [2] TO [8]
    RA(LCS(IP(4),1))=0.
    RA(LCS(IP(5),1))=0.
  FIN
  RA(LCS(IP(4),7))=1.0E+50
  RA(LCS(IP(5),7))=1.0E+50
ENDPROC [3]

C
PROCEDURE ZEROCURRENTVALUES [4]
  RA(LCS(IP(4),2))=0.
  RA(LCS(IP(5),2))=0.

```

Source Function Programs

ENDPROC [4]

```

PROCEDURE SAMPLE [5]
MDO=RAC(L,3)
FOR [N] = [1] TO [2]
  L=IP(3*N)
  CASE [N EQ. 1]
    RA(LCS(L,2))=TIME
  ORIF [N EQ. 2]
    IFF [IP(1).GT.0]
      PERFORM GETDIVERTEDMATERIAL [6]
    FIN
  FIN
  IA(LCS(L,1))=IAC(L,1)+1
  CV=RAC(L,2)
  RA(LCS(L,3))=RAC(L,3)+CV
  RA(LCS(L,4))=RAC(L,4)+CV CV
  IFF [CV GT. RAC(L,5)]
    RA(LCS(L,5))=CV
    RA(LCS(L,6))=TIME
  FIN
  IFF [CV LT. RAC(L,7)]
    RA(LCS(L,7))=CV
    RA(LCS(L,6))=TIME
  FIN
  IFF [IFIX(RLP(1+N,1)).GT.0]
    ICELL=IFIX((CV-RLP(1+N,3))/RLP(1+N,2))+1
    CASE [ICELL LT. 1]
      ICELL=1
    ORIF [ICELL GT. IFIX(RLP(1+N,1))]
      ICELL=IFIX(RLP(1+N,1))
    FIN
    ICELL=LCS(L,8+ICELL)
    IA(ICELL)=IA(ICELL)+1
  FIN
FIN

```

ENDPROC [5]

C

```

PROCEDURE GETDIVERTEDMATERIAL [6]
IFF [IA(IP(1)+3).LE.0]
  ERROR [902 RMDS NO MAGA FUNCTIONS TO TABULATE]
FIN
CALL PCOM(ILP(1,1),9,3,0,0,0,0,0,0)
FOR [I] = [1] TO [IA(IP(1)+3)]
  MTO=ILP(1,1)
  IA(IPOINT+2)=3
  SEND
  IFF [IA(IPOINT+2) NE.-3]
    ERROR [902 RMDS RESPONSE OF TABULATED FUNCTION]
  FIN
  LM=IA(IPOINT+3)
  CASE [MDO EQ. 1 OR MDO EQ. 3]

```

Source Function Programs

```

      RA(LCS(L,2))=FLOAT(IA(LM+8))
      CRIF (MDC EO 2 OR MDC EO 4)
      RA(LCS(L,2))=RA(LM+9)
      CRIF (MDC EO 5)
      CALL NCRP(LM,LC,BM)
      RA(LCS(L,2))=FLOAT(ND)
      OTHERWISE
      ERROR (902 RMDS  ILLEGAL MATERIAL DIVERSION OBJECTIVE TYPE)
      FIN
    FIN
  CALL FSTORE(1POINT)
ENDPROC (6)
ENDBLOCK

```

Appendix B
 FORTRAN PROGRAM

CONTROL PROGRAMS

```

C ASTORE(LOC,N)
C SUBROUTINE ALLOCATES STORAGE SPACE IN ARRAY IA/RA.
  SUBROUTINE ASTORE(LOC,N)
    DIMENSION IA(1)
    COMMON RA(1)
    EQUIVALENCE (IA(1),RA(1))
    COMMON /SYSTEM/MFROM,MT0,ICTYPE,IPOINT,
    1IDUMP,ISECA,ITRACE,L0,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
    1LEC,LFFB,LFFE,L1,LLE,LLFB,LLFE,LMS,LNE,LP,LV,MINNFI,
    1MNSE,NFE,NFL,N11,N12,N01,N02,NREP,NRUN,NSE,TIME,TSTOP
    IF(ITRACE.GE.5)GO TO 00026
    GO TO 00027
00026 CONTINUE
    WRITE(N01,00029)TIME,IA(LB+2)
00029 FORMAT(1X,E10.4,I4,2X,11HASTORE( ,N))
    WRITE(N01,00030)N
00030 FORMAT(1X,I10)
00027 CONTINUE
    IF(ITRACE.GE.6.AND.(ITRACE.EQ.ITRACE/2))GO TO 00031
    GO TO 00032
00031 CONTINUE
    WRITE(N02,00034)TIME,IA(LB+2)
00034 FORMAT(1X,E10.4,I4,2X,11HASTORE( ,N))
    WRITE(N02,00035)N
00035 FORMAT(1X,I10)
    CALL ICHECK
00032 CONTINUE
    NUM=N
    IF(NUM.LT.8)GO TO 00036
    GO TO 00037
00036 CONTINUE
    NUM=8
00037 CONTINUE
    IF(NUM.GT.NFL)GO TO 00039
    GO TO 00040
00039 CONTINUE
    WRITE(N01,00042)
00042 FORMAT(1X,15H      ERROR      /1X,
X 33HASTORE OUT OF FREE STORAGE SPACE)
    CALL ERROR
  
```

FORTRAN Control Programs

```

00040 CONTINUE
      ASSIGN 00043 TO L00001
      GO TO 00001
00043 CONTINUE
      IF(L.LE.0)GO TO 00044
      GO TO 00045
00044 CONTINUE
      WRITE(NO1,00047)
00047 FORMAT(1X,15H      ERROR      /1X,
X      38HASTOR: NO STORAGE BLOCK FOUND IN SCAN)
      CALL ERROR
      GO TO 00046
00045 IF(IA(L+1).GE.NUM+10)GO TO 00048
      GO TO 00049
00048 CONTINUE
      ASSIGN 00050 TO L00002
      GO TO 00002
00050 CONTINUE
      GO TO 00046
00049 IF(IA(L+1).GE.NUM)GO TO 00051
      GO TO 00052
00051 CONTINUE
      ASSIGN 00053 TO L00003
      GO TO 00003
00053 CONTINUE
      GO TO 00046
00052 IF(IA(L+1).LT.NUM)GO TO 00054
      GO TO 00055
00054 CONTINUE
      ASSIGN 00056 TO L00004
      GO TO 00004
00056 CONTINUE
00055 CONTINUE
00046 CONTINUE
      IF(INFL.LT.MINNFL)GO TO 00057
      GO TO 00058
00057 CONTINUE
      MINNFL=NFL
00058 CONTINUE
      IF(ITRACE.GE.5)GO TO 00060
      GO TO 00061
00060 CONTINUE
      WRITE(NO1,00063)TIME,IA(LB+2)
00063 FORMAT(1X,E10.4,14,2X,13HASTORE(LOC,N))
      WRITE(NO1,00064)LOC,N
00064 FORMAT(1X,2110)
00061 CONTINUE
      IF(ITRACE.GE.6.AND.(ITRACE.EQ.ITRACE/2 2))GO TO 00065
      GO TO 00066
00065 CONTINUE
      WRITE(NO2,00068)TIME,IA(LB+2)
00068 FORMAT(1X,E10.4,14,2X,13HASTORE(LOC,N))

```


FORTRAN Control Programs

```

WRITE(NQ2,00069)LOC,H
00069 FORMAT(1X,2I10)
      CALL ICHECK
00066 CONTINUE
      RETURN
C
00001 CONTINUE
      L=LFFB
      M=IA(L+1)
      LM=L
      LPP=0
      LPM=0
00070 IF(IA(L+1).LT.NUM.AND.L.NE.0)GO TO 00071
      GO TO 00072
00071 CONTINUE
      IF(IA(L+1).GT.M)GO TO 00073
      GO TO 00074
00073 CONTINUE
      M=IA(L+1)
      LM=L
      LPM=LPP
00074 CONTINUE
      IF(LBCA+IA(LBCA+1)+2.EQ.L.OR.L+IA(L+1)+2.EQ.LBCA)GO TO 00076
      GO TO 00077
00076 CONTINUE
      ASSIGN 00079 TO L00005
      GO TO 00005
00079 CONTINUE
00077 CONTINUE
      LPP=L
      L=IA(L)
      GO TO 00070
00072 CONTINUE
      IF(L.EQ.0)GO TO 00080
      GO TO 00081
00080 CONTINUE
      L=LM
00081 CONTINUE
      LBCA=IA(LBCA)
      IF(LBCA.EQ.0)GO TO 00083
      GO TO 00084
00083 CONTINUE
      LBCA=LFFB
00084 CONTINUE
      GO TO L00001, (00043)
C
00002 CONTINUE
      IF(L.EQ.LFFB)GO TO 00086
      GO TO 00087
00086 CONTINUE
      LFFB=L+2+NUM
      GO TO 00088

```

FORTRAN Control Programs

```

00087 CONTINUE
      IA(LPP)=L+2*NUM
00088 CONTINUE
      IF(L.EQ.LLFB)GO TO 00089
      GO TO 00090
00089 CONTINUE
      LLFB=L+2*NUM
00090 CONTINUE
      IA(L+2*NUM)=IA(L)
      IA(L+3*NUM)=IA(L+1)-NUM-2
      NFL=NFL-NUM-2
      IF(LBCA.EQ.L)GO TO 00092
      GO TO 00093
00092 CONTINUE
      LBCA=L+2*NUM
00093 CONTINUE
      IA(L)=0
      IA(L+1)=NUM
      IA(L+2)=NUM
      LOC=L
      GO TO L00002, (00050)
C
00003 CONTINUE
      IF(L.EQ.LFFB)GO TO 00095
      GO TO 00096
00095 CONTINUE
      LFFB=IA(L)
      GO TO 00097
00096 CONTINUE
      IA(LPP)=IA(L)
00097 CONTINUE
      NFL=NFL-IA(L+1)
      IF(LBCA.EQ.L)GO TO 00098
      GO TO 00099
00098 CONTINUE
      LBCA=IA(L)
      IF(LBCA.EQ.0)GO TO 00101
      GO TO 00102
00101 CONTINUE
      LBCA=LFFB
00102 CONTINUE
00099 CONTINUE
      IA(L)=0
      IA(L+2)=IA(L+1)
      LOC=L
      GO TO L00003, (00053)
C
00004 CONTINUE
      IF(LM.NE.LFFB)GO TO 00104
      GO TO 00105
00104 CONTINUE
      IA(LPM)=0

```

FORTRAN Control Programs

```

        IA(LFFB)=LFFB
        LFFB=LPM
        LFFB=LM
        LPM=0
00105 CONTINUE
        L1=LCS(LM,NUM)
        L2=LCSB+1+IA(LCSB+1)
00107 IF(L.NE.LBCA.AND.L.NE.LCSB)GO TO 00108
        GO TO 00109
00108 CONTINUE
        L=IA(L)
        GO TO 00107
00109 CONTINUE
        IF(L.EQ.LBCA)GO TO 00110
        GO TO 00111
00110 CONTINUE
        ASSIGN 00113 TO L00006
        GO TO 00006
00113 CONTINUE
00111 CONTINUE
        IF(L2-L1.GE.10)GO TO 00114
        GO TO 00115
00114 CONTINUE
        IA(LM+2)=NUM
        LFFB=L1+1
        IA(LFFB)=IA(LCSB)
        IA(LFFB+1)=L2-L1-2
        NFL=NFL-NUM-2
        GO TO 00116
00115 CONTINUE
        IA(LM+2)=NUM+L2-L1
        LFFB=IA(LCSB)
        NFL=NFL-NUM-L2+L1
00116 CONTINUE
        IF(LFFB.EQ.0)GO TO 00117
        GO TO 00118
00117 CONTINUE
        WRITE(59,00120)
00120 FORMAT(/1X,17H      WARNING      /1X,
X      38HASTORE  LAST AVAILABLE BLOCK ALLOCATED)
00118 CONTINUE
        LFFB=0
        LOC=LM
        GO TO L00004, (00056)
C
00005 CONTINUE
        IF(LBCA.LT.L)GO TO 00121
        GO TO 00122
00121 CONTINUE
        IA(LBCA)=IA(L)
        IA(LBCA+1)=IA(LBCA+1)+IA(L+1)+2
        IA(LPP)=LBCA

```

FORTTRAN Control Programs

```

GO TO 00123
00122 CONTINUE
      IA(L+1)=IA(L+1)+IA(LBCA+1)+2
00123 CONTINUE
      LBCA=IA(LBCA)
      IF(LBCA.EQ.0)GO TO 00124
      GO TO 00125
00124 CONTINUE
      LBCA=LFFB
00125 CONTINUE
      NFL=NFL+2
      GO TO L00005, (00079)
C
00006 CONTINUE
      IF(L2-L1.GE.10)GO TO 00127
      GO TO 00126
00127 CONTINUE
      LBCA=L1+1
      GO TO 00129
00128 CONTINUE
      LBCA=IA(LCSB)
00129 CONTINUE
      GO TO L00006, (00113)
END
C CEC(MIN,MAX,KODE)
C SUBROUTINE CHECKS EVENT CODE AT IA(IPOINT) AND PRINTS ERROR MESSAGE
C IF NOT IN THE INTERVAL (MIN,MAX).
      SUBROUTINE CEC(MIN,MAX,KODE)
      DIMENSION IA(1)
      COMMON RA(1)
      EQUIVALENCE (IA(1),RA(1))
      COMMON /SYSTEM/MFROM,MT0,ICTYPE,IPOINT,
      1IDUMP,ISECA,ITRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
      1LEC,LFFB,LFFE,L1,LLE,LLFB,LLFE,LMS,LNE,LP,LV,MINNFL,
      1MNSE,NFE,NFL,NI1,NI2,NO1,NO2,NREP,NRUN,NSE,TIME,TSTOP
      KODE=IA(IPOINT)
      IF(KODE.LT.MIN.OR.KODE.GT.MAX)GO TO 00026
      GO TO 00027
00026 CONTINUE
      WRITE(NO1,00029)
00029 FORMAT(/1X,15H      ERROR      /1X,23HCEC  ILLEGAL EVENT CODE)
      CALL ERROR
00027 CONTINUE
      RETURN
      END
C CLOCK
C SUBROUTINE ADVANCES SIMULATION TIME.
      SUBROUTINE CLOCK
      DIMENSION IA(1)
      COMMON RA(1)
      EQUIVALENCE (IA(1),RA(1))
      COMMON /SYSTEM/MFROM,MT0,ICTYPE,IPOINT,

```

FORTRAN Control Programs

```

11DUMP, ISECA, ITRACE, LB, LBCA, LBLIST, LBPT, LBUF, LCSB, LE,
1LEC, LFFB, LFFE, LI, LLE, LLFB, LLFE, LMS, LNE, LP, LV, MINNFL,
1MNSF, NFE, NFL, NI1, NI2, NO1, NO2, NREP, NRUN, NSE, TIME, TSTOP
00026 IF(TIME,LT,TSTOP)GO TO 00027
      GO TO 00028
00027 CONTINUE
      IF(LNE.GT.0)GO TO 00029
      GO TO 00030
00029 CONTINUE
      TIME=RA(LNE+1)
      MFROM=0
      MTO=IA(LNE+2)
      ICTYPE=8
      IPOINT=LNE+3
      CALL SEND
      N=IA(LNE)
      IA(LNE)=LFFE
      LFE=LNE
      LNE=N
      NSE=NSE-1
      NFE=NFE+1
      GO TO 00031
00030 IF(LNE.EQ.0)GO TO 00032
      GO TO 00033
00032 CONTINUE
      TIME=TSTOP
      GO TO 00031
00033 CONTINUE
      WRITE(NO1,00034)
00034 FORMAT(/IX,15H      ERROR      /IX,19HCLOCK  NEGATIVE LNE)
      CALL ERROR
00031 CONTINUE
      GO TO 00026
00028 CONTINUE
      RETURN
      END
C CMB(LM,LH,LPB,LN,LL,NL,NT,NC,A)
C SUBROUTINE CREATES MATERIAL BLOCK INITIALIZED TO LH,LPB,...,A
C AND RETURNS BLOCK LOCATION LM.
      SUBROUTINE CMB(LM,LH,LPB,LN,LL,NL,NT,NC,A)
      DIMENSION IA(1)
      COMMON RA(1)
      EQUIVALENCE (IA(1),RA(1))
      COMMON /SYSTEM/MFROM,MTO,ICTYPE,IPOINT,
11DUMP, ISECA, ITRACE, LB, LBCA, LBLIST, LBPT, LBUF, LCSB, LE,
1LEC, LFFB, LFFE, LI, LLE, LLFB, LLFE, LMS, LNE, LP, LV, MINNFL,
1MNSF, NFE, NFL, NI1, NI2, NO1, NO2, NREP, NRUN, NSE, TIME, TSTOP
      CALL ASTORE(LM,8)
      IF(IA(LM).NE.0)GO TO 00026
      GO TO 00027
00026 CONTINUE
      WRITE(NO1,00029)

```

FORTRAN Control Programs

```

00029 FORMAT(/1X,15H      ERROR      /1X,
X      43HCMB NO CONTIGUOUS SPACE FOR MATERIAL BLOCK)
      CALL ERROR
00027 CONTINUE
      IA(LM+2)=LH
      IA(LM+3)=LPB
      IA(LM+4)=LN
      IA(LM+5)=LL
      IA(LM+6)=NL
      IA(LM+7)=NT
      IA(LM+8)=NC
      RA(LM+9)=A
      RETURN
      END
C COPYB(L1,L2)
C SUBROUTINE COPIES THE STORAGE BLOCK AT L1 INTO A NEW
C BLOCK AT L2 AT LEAST AS LARGE AS THE ORIGINAL AND SETS
C THE REMAINDER OF THE BLOCK AT L2 TO INTEGER 0.
      SUBROUTINE COPYB(L1,L2)
      DIMENSION IA(1)
      COMMON RA(1)
      EQUIVALENCE (IA(1),RA(1))
      COMMON /SYSTEM/MFROM,MTG,ICTYPE,IPONT,
      1IDUMP,ISECA,ITRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
      1LEC,LFFB,LFFE,L1,LLE,LLFB,LLFE,LMS,LNE,LP,LV,MINNFI,
      1MNSI,NFE,NFL,N11,N12,N01,N02,NREP,NRUN,NSI,TIME,TSTOP
      CALL ASTORE(L2,IA(L1+1))
      IF(IA(L2).GT.0)GO TO 00026
      GO TO 00027
00026 CONTINUE
      WRITE(N01,00029)
00029 FORMAT(/1X,15H      ERROR      /1X,
X      36HCOPYB NO CONTIGUOUS SPACE AVAILABLE)
      CALL ERROR
00027 CONTINUE
      I=1-(1)
00030 IF((1)-(1-IA(L2+1))).LE.0)GO TO 00031
      GO TO 00032
00031 CONTINUE
      IF(1.LE.1-IA(L1+1))GO TO 00033
      GO TO 00034
00033 CONTINUE
      IA(L2+1)=IA(L1+1)
      GO TO 00035
00034 CONTINUE
      IA(L2+1)=0
00035 CONTINUE
      GO TO 00030
00032 CONTINUE
      RETURN
      END

```

FORTRAN Control Programs

```

C CREATE(NT,NP,NV,IPVT)
C SUBROUTINE CREATES INSTANCE MTO OF FUNCTION TYPE NT WITH NP PARAMETERS
C AND NV VARIABLES. PARAMETER, VARIABLE AND C LOCATIONS ARE ZEROED
C USING PARAMETER AND VARIABLE TYPE CODES IN .PVT.
      SUBROUTINE CREATE(NT,NP,NV,IPVT)
      DIMENSION IPVT(1)
      DIMENSION A(1)
      COMMON /A/
      EQUIVALENCE (A(1),RA(1))
      COMMON /SYSTEM/MFROM,MTO,ICTYPE,IPONT,
1DUMP,ISECA,ITRACE,LB,LBCA,LBLTST,LBPT,LBUF,LCSB,LE,
1LEC,LFFB,LFFE,L1,LLE,LLFB,LLFE,LMS,LME,LP,LV,MINNFI,
1MNSE,NFE,NFL,NT1,NI2,NO1,NO2,NREP,NUN,NSE,TIME,TSTOP
      NSIZE=3*(NP+2)*NV
      CALL ASTORE(LB,NSIZE)
      IF(1A(LB).GT.0)GO TO 00026
      GO TO 00027
00026 CONTINUE
      WRITE(NO1,00029)
00029 FORMAT(/IX,15H      ERROR      /IX,
X      42HCREATE NO CONTIGUOUS STORAGE FOR FUNCTION)
      CALL ERROR
      CONTINUE
00027 IF(LCS(LBPT,MTO+1))=LB
      IA(LB)=NT
      IA(LB+2)=MTO
      IA(LB+3)=NP
      IA(LB+4)=NV
      LP=LB+5
      LV=LP+NP
      LI=LV+NV
      LE=LI+NV-1
      I=1-11)
00030 I=I+1
      IF(11) (1-(NP)),LE.0)GO TO 00031
      GO TO 00032
00031 CONTINUE
      IF(1PVT(1),EQ.3)GO TO 00033
      GO TO 00034
00033 CONTINUE
      RA(LP-1+1)=0
      GO TO 00035
00034 CONTINUE
      IA(LP-1+1)=0
00035 CONTINUE
      GO TO 00030
00032 CONTINUE
      I=1-11)
00036 I=I+1
      IF(11) (1-(NV)),LE.0)GO TO 00037
      GO TO 00038
00037 CONTINUE

```

FORTRAN Control Programs

```

      IF(IPVT(INP+1).EQ.3)GO TO 00039
      GO TO 00040
00039 CONTINUE
      RA(LV-1+1)=0.
      RA(LV-1+1+NV)=0.
      GO TO 00041
00040 CONTINUE
      IA(LV-1+1)=0
      IA(LV-1+1+NV)=0
00041 CONTINUE
      GO TO 00036
00039 CONTINUE
      RETURN
      END
C CRMD1(NO,LMZ,L")
C SUBROUTINE CREATES NO MATERIALS AT CONTAINMENT LEVEL 1 ACCORDING
C TO MATERIAL SPECIFICATIONS AT LMZ AND RETURNS THE LOCATION IN LM.
      SUBROUTINE CRMD1(NO,LMZ,LM)
      DIMENSION IA(1)
      COMMON RA(1)
      EQUIVALENCE (IA(1),RA(1))
      COMMON /SYSTEM/MFROM,MT0,ICTYPE,IPPOINT,
      1IDUMP,ISECA,ITRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
      1LEC,LFFB,LFFE,LI,LLE,LLFB,LLFE,LMS,LME,LP,LV,MINMFL,
      1MNSE,NFE,NFL,N11,N12,N01,N02,NREP,NRUN,NSE,TIME,ISTOP
      IF(ITRACE.GE.5)GO TO 00026
      GO TO 00027
00026 CONTINUE
      WRITE(N01,00029)TIME,IA(LB+2)
00029 FORMAT(1X,E10.4,I4,2X,15HCRMD1(NO,LMZ, ))
      WRITE(N01,00030)NO,LMZ
00030 FORMAT(1X,2I10)
00027 CONTINUE
      IF(ITRACE.GE.6.AND.(ITRACE.EQ.ITRACE/2 2))GO TO 00031
      GO TO 00032
00031 CONTINUE
      WRITE(N02,00034)TIME,IA(LB+2)
00034 FORMAT(1X,E10.4,I4,2X,15HCRMD1(NO,LMZ, ))
      WRITE(N02,00035)NO,LMZ
00035 FORMAT(1X,2I10)
      CALL ICHECK
00032 CONTINUE
      CALL CMB(L1,0,0,0,0,1,0,0,0.)
      NM=1-(1)
00036 NM=NM+1
      IF((1) (NM-(NO)).LE.0)GO TO 00037
      GO TO 00038
00037 CONTINUE
      CALL CMB(L0,0,0,0,0,0,0,0,0.)
      IA(L0+7)=RAC(LMZ,LINM(NM,7,1))
      IA(L0+8)=RUNIF(RAC(LMZ,LINM(NM,7,2)),RAC(LMZ,LINM(NM,7,3)))
      RA(L0+9)=RNORMT(RAC(LMZ,LINM(NM,7,4)),RAC(LMZ,LINM(NM,7,5)),

```


FORTRAN Control Programs

```

1      RAC(LMZ,LIINM(NM,7,6)),RAC(LMZ,LIINM(NM,7,7))
      IF(NM.EQ.1)GO TO 00039
      GO TO 00040
00039 CONTINUE
      IA(L0+2)=L1
      IA(L1+5)=L0
      GO TO 00041
00040 CONTINUE
      IA(L0+3)=LOP
      IA(LOP+4)=L0
00041 CONTINUE
      LOP=L0
      GO TO 00036
00038 CONTINUE
      CALL UMD(L1)
      LM=L1
      IF(TRACE.GE.5)GO TO 00042
      GO TO 00043
00042 CONTINUE
      WRITE(N01,00045)TIME,IA(LB+2)
00045 FORMAT(1X,E10.4,I4,2X,13HCRMD1( , ,LM))
      WRITE(N01,00046)LM
00046 FORMAT(1X,I10)
00043 CONTINUE
      IF(TRACE.GE.6.AND.(TRACE.EQ.1TRACE/2 2))GO TO 00047
      GO TO 00048
00047 CONTINUE
      WRITE(N02,00050)TIME,IA(LB+2)
00050 FORMAT(1X,E10.4,I4,2X,13HCRMD1( , ,LM))
      WRITE(N02,00051)LM
00051 FORMAT(1X,I10)
      CALL ICHECK
00048 CONTINUE
      RETURN
      END
C CRMD2(N1,NO,LMZ,LM)
C SUBROUTINE CREATES N1 CONTAINERS AT CONTAINMENT LEVEL 2 WITH
C NO MATERIALS IN EACH ACCORDING TO MATERIAL SPECIFICATIONS AT
C LMZ AND RETURNS THE LOCATION IN LM.
      SUBROUTINE CRMD2(N1,NO,LMZ,LM)
      DIMENSION IA(1)
      COMMON RA(1)
      EQUIVALENCE (IA(1),RA(1))
      COMMON /SYSTEM/MFROM,MT0,ICTYPE,IP0INT,
11DUMP,ISECA,TRACE,LB,LBCA,LB1ST,LBPT,LBUF,LCSB,LE,
1LEC,LFEB,LFEE,L1,LLE,LLFB,LLFE,LMS,LNE,LP,LV,MINNFI,
1MNSE,NFE,NFL,N11,N12,N01,N02,NREP,NRUN,NSE,TIME,TSTOP
      CALL CHB(LM,0,0,0,0,2,0,0,0)
      NC=1-(1)
00026 NC=NC+1
      IF(1) (NC-(N1)).LE.0)GO TO 00027
      GO TO 00028

```

FORTRAN Control Programs

```

00027 CONTINUE
      CALL CMB(L2,0,0,0,0,2,0,0,0.)
      IF(NC.EQ.1)GO TO 00029
      GO TO 00030
00029 CONTINUE
      IA(L2+2)=LM
      IA(LM+5)=L2
      GO TO 00031
00030 CONTINUE
      IA(L2P+4)=L2
      IA(L2+3)=L2P
00031 CONTINUE
      LPP=L2
      CALL CRMD1(NC,LM2,L1)
      IA(L2+5)=L1
      IA(L1+2)=L2
      GO TO 00026
00028 CONTINUE
      CALL UMD(LM)
      RETURN
      END
C DESTROY(IPVT)
C SUBROUTINE DESTROYS THE FUNCTION AT LB USING PARAMETER AND VARIABLE
C TYPE CODES IN IPVT.
      SUBROUTINE DESTRO(IPVT)
      DIMENSION IPVT(1)
      DIMENSION IA(1)
      COMMON RA(1)
      EQUIVALENCE (IA(1),RA(1))
      COMMON /SYSTEM/MFROM,MTG,ICTYPE,IPOINT,
1DUMP,ISECA,ITRACE,LB,BCA,LBLIST,LBPT,LBUF,LCSB,LE,
1LEC,LFPB,LFFE,L1,LLE,L1PB,LLFE,LMS,LNE,LP,LV,MINNPL,
1INSE,NFE,NFL,N11,N12,NOT,NO2,NREP,NRUN,NSE,TIME,TSTOF
      I=LP-(1)
00026 I=I+1
      IF((I)-(LI-1)).LE.0)GO TO 00027
      GO TO 00028
00027 CONTINUE
      K=I-LP+1
      IF((IP-T(K).EQ.2.OR.IPVT(K).EQ.4).AND. IA(I).GT.0)GO TO 00029
      GO TO 00030
00029 CONTINUE
      CALL FSTORE(IA(I))
      IF(.GE.LV.AND. IA(I+IA(LB+4)).GT.0)GO TO 00032
      GO TO 00033
00032 CONTINUE
      CALL FSTORE(IA(I+IA(LB+4)))
00033 CONTINUE
00030 CONTINUE
      GO TO 00026
00028 CONTINUE
      IA(LB)=0

```

```

      CALL FSTORE(LB)
      IALLCS(LBPT, MTO+1) = 0
      RETURN
      END
C DMPSYS
C SUBROUTINE DUMPS MODEL SYSTEM.
      SUBROUTINE DMPSYS
      DIMENSION IA(1)
      COMMON RA(1)
      EQUIVALENCE (IA(1), RA(1))
      COMMON /SYSTEM/MFROM, MTO, ICTYPE, IPOINT,
      1 IDUMP, ISECA, ITRACE, LB, LBCA, LBLIST, LBPT, LBUF, LCSB, LE,
      1 LEC, LFF3, LFFE, LI, LLE, LLFB, LLFE, LNS, LME, LP, LV, MINNFL,
      1 MNSE, NFE, NFL, NI1, NI2, NO1, NO2, NREP, NRUN, NSE, TIME, TSTOP
      WRITE(N01, 00026)
00026 FORMAT(1X, 6HCOMMON)
      WRITE(N01, 00027) MTO
00027 FORMAT(10X, 3HMT0, I12)
      WRITE(N01, 00028) MFROM
00028 FORMAT(8X, 5HMFROM, I12)
      WRITE(N01, 00029) ICTYPE
00029 FORMAT(7X, 6HICTYPE, I12)
      WRITE(N01, 00030) IPOINT
00030 FORMAT(7X, 6HIPOINT, I12)
      WRITE(N01, 00031) IDUMP
00031 FORMAT(8X, 5HIDUMP, I12)
      WRITE(N01, 00032) ISECA
00032 FORMAT(8X, 5HISECA, I12)
      WRITE(N01, 00033) ITRACE
00033 FORMAT(7X, 6HITRACE, I12)
      WRITE(N01, 00034) LB
00034 FORMAT(11X, 2HLB, I12)
      WRITE(N01, 00035) LBCA
00035 FORMAT(9X, 4HLBCA, I12)
      WRITE(N01, 00036) LBLIST
00036 FORMAT(7X, 6HLBLIST, I12)
      WRITE(N01, 00037) LBPT
00037 FORMAT(9X, 4HLBPT, I12)
      WRITE(N01, 00038) LBUF
00038 FORMAT(9X, 4HLBUF, I12)
      WRITE(N01, 00039) LCSB
00039 FORMAT(9X, 4HLCSE, I12)
      WRITE(N01, 00040) LE
00040 FORMAT(11X, 2HLE, I12)
      WRITE(N01, 00041) LEC
00041 FORMAT(10X, 3HLEC, I12)
      WRITE(N01, 00042) LFFB
00042 FORMAT(9X, 4HLFFB, I12)
      WRITE(N01, 00043) LFFE
00043 FORMAT(9X, 4HLFFE, I12)
      WRITE(N01, 00044) LI
00044 FORMAT(11X, 2HLI, I12)

```

FORTRAN Control Programs

```

00045 WRITE(N01,00045)LLE
      FORMAT(10X,3HLLLE,112)
00046 WRITE(N01,00046)LLFB
      FORMAT(9X,4HLLFB,112)
00047 WRITE(N01,00047)LLFE
      FORMAT(9X,4HLLFE,112)
00048 WRITE(N01,00048)LMS
      FORMAT(10X,3HLMS,112)
00049 WRITE(N01,00049)LNE
      FORMAT(10X,3HLNE,112)
00050 WRITE(N01,00050)LP
      FORMAT(11X,2HLP,112)
00051 WRITE(N01,00051)LV
      FORMAT(11X,2HLV,112)
00052 WRITE(N01,00052)MINNFL
      FORMAT(7X,6HMINNFL,112)
00053 WRITE(N01,00053)MNSE
      FORMAT(9X,4HMNSE,112)
00054 WRITE(N01,00054)NFE
      FORMAT(10X,3HNFE,112)
00055 WRITE(N01,00055)NFL
      FORMAT(10X,3HNFL,112)
00056 WRITE(N01,00056)NI1
      FORMAT(10X,3HNI1,112)
00057 WRITE(N01,00057)NI2
      FORMAT(10X,3HNI2,112)
00058 WRITE(N01,00058)N01
      FORMAT(10X,3HNO1,112)
00059 WRITE(N01,00059)N02
      FORMAT(10X,3HNO2,112)
00060 WRITE(N01,00060)NREP
      FORMAT(9X,4HNREP,112)
00061 WRITE(N01,00061)NRUN
      FORMAT(9X,4HNRUN,112)
00062 WRITE(N01,00062)NSE
      FORMAT(10X,3HNSE,112)
00063 WRITE(N01,00063)TIME
      FORMAT(9X,4HTIME,E12.4)
00064 WRITE(N01,00064)TSTOP
      FORMAT(8X,5HTSTOP,E12.4)
      K=1+(LBLIST+3)
      I=1-I
00065 IF(I) (1-(K)).LE.0)GO TO 00066
      GO TO 00067
00066 CONTINUE
      MFROM=0
      MTO=1A(LCS(LBLIST,1+2))
      ICTYPE=7
      IPOINT=0
      WRITE(N01,00068)MTO
00068 FORMAT(1X,8HFUNCTION,15)

```

```

CALL SEND
GO TO 00055
00067 CONTINUE
00069 WRITE(NO1,00069)
00069 WRITE(NO1,00070)EG
00070 WRITE(NO1,00071)E2
00071 WRITE(NO1,00072)LINE
00072 WRITE(NO1,00072)LINE
00073 WRITE(NO1,00073)LINE
00074 WRITE(NO1,00074)LINE
00075 WRITE(NO1,00075)LINE
00076 WRITE(NO1,00076)LINE
00077 WRITE(NO1,00077)LINE
00078 WRITE(NO1,00078)LINE
00079 IF(LT.01GO TO 00080
GO TO 00081
00080 CONTINUE
00082 WRITE(NO1,00082)LA(L),RA(L+1),LA(L+2),LA(L+3),LA(L+4)
00082 WRITE(NO1,210,3X,E10.4,310)
GO TO 00079
00081 CONTINUE
00083 WRITE(NO1,100FREE STORE)
00084 WRITE(NO1,00084)LFB
00085 WRITE(NO1,00085)LINE
00086 WRITE(NO1,00086)LINE
00087 WRITE(NO1,00087)LINE
00088 WRITE(NO1,00088)LINE
00089 IF(LT.01GO TO 00090
GO TO 00091
00090 CONTINUE
00092 WRITE(NO1,00092)LA(L),LA(L+1)
00092 WRITE(NO1,310)
GO TO 00089

```

FORTRAN Control Programs

```

00091 CONTINUE
      RETURN
      END
C DUMP(IPVT)
C SUBROUTINE DUMPS THE MODEL FUNCTION AT LB.
      SUBROUTINE DUMP(IPVT)
      DIMENSION IA(1)
      COMMON RA(1)
      EQUIVALENCE (IA(1),RA(1))
      COMMON /SYSTEM/MFROM,MTD,ICTYPE,IPOINT,
11DUMP,ISECA,ITRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
1LEC,LFFB,LFFE,LI,LLE,LLFB,LLFE,LMS,LNE,LP,LV,MINNFI,
1MNSE,NFE,NFL,N11,N12,NO1,NO2,NREP,NRUN,NSE,TIME,TSTOP
      CALL PFLB
      CALL PVLB(IPVT)
      CALL PICLB(IPVT)
      RETURN
      END
C ERROR
C SUBROUTINE PRINTS DIAGNOSTIC DATA AND STOPS PROGRAM.
      SUBROUTINE ERROR
      DIMENSION IA(1)
      COMMON RA(1)
      EQUIVALENCE (IA(1),RA(1))
      COMMON /SYSTEM/MFROM,MTD,ICTYPE,IPOINT,
11DUMP,ISECA,ITRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
1LEC,LFFB,LFFE,LI,LLE,LLFB,LLFE,LMS,LNE,LP,LV,MINNFI,
1MNSE,NFE,NFL,N11,N12,NO1,NO2,NREP,NRUN,NSE,TIME,TSTOP
      CALL DMPSYS
      CALL EXIT
      RETURN
      END
C EVDRS(N,M,V)
C SUBROUTINE RETURNS SAMPLE VALUE V FROM A POPULATION
C WITH DISTRIBUTION TYPE IN INTEGER PARAMETER N AND STATISTICS IN
C REAL LIST PARAMETER M OF THE CALLING PROGRAM.
      SUBROUTINE EVDRS(N,M,V)
      DIMENSION IA(1)
      COMMON RA(1)
      EQUIVALENCE (IA(1),RA(1))
      COMMON /SYSTEM/MFROM,MTD,ICTYPE,IPOINT,
11DUMP,ISECA,ITRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
1LEC,LFFB,LFFE,LI,LLE,LLFB,LLFE,LMS,LNE,LP,LV,MINNFI,
1MNSE,NFE,NFL,N11,N12,NO1,NO2,NREP,NRUN,NSE,TIME,TSTOP
      IF (ITRACE.GE.5) GO TO 00026
      GO TO 00027
00026 CONTINUE
      WRITE(NO1,00029)TIME,IA(LB+2)
00029 FORMAT(1X,E10.4,2X,12HEVDRS(N,M, ))
      WRITE(NO1,00030)N,M
00030 FORMAT(1X,2I10)

```

FORTRAN Control Programs

```

00027 CONTINUE
      IF(I TRACE.GE.6.AND.(I TRACE.EQ.I TRACE/2 2))GO TO 00031
      GO TO 00032
00031 CONTINUE
      WRITE(NO2,00034)TIME,IA(LB+2)
00034 FORMAT(1X,E10.4,I4,2X,12HEVDRS(N,M, ))
      WRITE(NO2,00035)N,M
00035 FORMAT(1X,I210)
      CALL ICHECK
00032 CONTINUE
      IF(IP(N).EQ.1)GO TO 00036
      GO TO 00037
00036 CONTINUE
      V=RUNIF(RLP(M,1),RLP(M,2))
      GO TO 00038
00037 IF(IP(N).EQ.2)GO TO 00039
      GO TO 00040
00039 CONTINUE
      V=RNORM(RLP(M,1),RLP(M,2))
      GO TO 00038
00040 IF(IP(N).EQ.3)GO TO 00041
      GO TO 00042
00041 CONTINUE
      V=RNORMT(RLP(M,1),RLP(M,2),RLP(M,3),RLP(M,4))
      GO TO 00038
00042 IF(IP(N).EQ.4)GO TO 00043
      GO TO 00044
00043 CONTINUE
      CALL URN(IX,U)
      V=-RLP(M,1) ALOG(U)
      GO TO 00038
00044 CONTINUE
      WRITE(NO1,00045)
00045 FORMAT(1X,15H ERROR /1X,
X 31HEVDRS ILLEGAL DISTRIBUTION TYPE)
      CALL ERROR
00046 CONTINUE
      IF(I TRACE.GE.5)GO TO 00046
      GO TO 00047
00046 CONTINUE
      WRITE(NO1,00049)TIME,IA(LB+2)
00049 FORMAT(1X,E10.4,I4,2X,12HEVDRS( , ,V))
      WRITE(NO1,00050)V
00050 FORMAT(1X,E12.4)
00047 CONTINUE
      IF(I TRACE.GE.6.AND.(I TRACE.EQ.I TRACE/2 2))GO TO 00051
      GO TO 00052
00051 CONTINUE
      WRITE(NO2,00054)TIME,IA(LB+2)
00054 FORMAT(1X,E10.4,I4,2X,12HEVDRS( , ,V))
      WRITE(NO2,00055)V
00055 FORMAT(1X,E12.4)

```

FORTRAN Control Programs

```

      CALL ICHECK
00052 CONTINUE
      RETURN
      END
C FPNTRS(NFUN)
C SUBROUTINE SETS POINTERS LB, LP, LV, LI, AND LE FOR FUNCTION NFUN.
      SUBROUTINE FPNTRS(NFUN)
      DIMENSION IA(1)
      COMMON RA(1)
      EQUIVALENCE (IA(1),RA(1))
      COMMON /SYSTEM/MFROM, MTO, ICTYPE, IPOINT,
      1 DUMP, ISECA, ITRACE, LB, LBCA, LBLIST, LBPT, LBUF, LCSB, LE,
      1 LEC, LFFB, LFFE, LI, LLE, LLFB, LLFE, LMS, LNE, LP, LV, MINNFL,
      1 MNSE, NFE, NFL, NI1, NI2, NO1, NO2, NREP, NRUN, NSE, TIME, TSTOP
      LB=IA(LCS+LBPT,NFUN+1)
      NP=IA(LB+3)
      NV=IA(LB+4)
      LP=LB+5
      LV=LP+NP
      LI=LV+NV
      LE=LI+NV-1
      RETURN
      END
C FRMD(LM)
C SUBROUTINE FREES MATERIAL DATA AT LM.
      SUBROUTINE FRMD(LM)
      DIMENSION IA(1)
      COMMON RA(1)
      EQUIVALENCE (IA(1),RA(1))
      COMMON /SYSTEM/MFROM, MTO, ICTYPE, IPOINT,
      1 DUMP, ISECA, ITRACE, LB, LBCA, LBLIST, LBPT, LBUF, LCSB, LE,
      1 LEC, LFFB, LFFE, LI, LLE, LLFB, LLFE, LMS, LNE, LP, LV, MINNFL,
      1 MNSE, NFE, NFL, NI1, NI2, NO1, NO2, NREP, NRUN, NSE, TIME, TSTOP
      IF (ITRACE.GE.5) GO TO 00026
      GO TO 00027
00026 CONTINUE
      WRITE(NO1,00029)TIME,IA(LB+2)
00029 FORMAT(1X,E10.4,14,2X,8HFRMD(LM))
      WRITE(NO1,00030)LM
00030 FORMAT(1X,110)
00027 CONTINUE
      IF (ITRACE.GE.6.AND.(ITRACE.EQ.ITRACE/2 2))GO TO 00031
      GO TO 00032
00031 CONTINUE
      WRITE(NO2,00034)TIME,IA(LB+2)
00034 FORMAT(1X,E10.4,14,2X,8HFRMD(LM))
      WRITE(NO2,00035)LM
00035 FORMAT(1X,110)
      CALL ICHECK
00032 CONTINUE
      IF (LM.LE.0)GO TO 00036
      GO TO 00037

```


FORTRAN Control Programs

```

00036 CONTINUE
      WRITE(N01,00039)LM
00039 FORMAT(11X,14HNO MATERIAL AT,16)
      GO TO 00038
00037 CONTINUE
      MODE=1
      L=LM
00040 IF(L.NE.LM.OR.MODE.NE.4)GO TO 00041
      GO TO 00042
00041 CONTINUE
      ASSIGN 00043 TO L00001
      GO TO 00001
00043 CONTINUE
      GO TO 00040
00042 CONTINUE
      CALL FSTORE(LM)
      IF(1TRACE.GE.5)GO TO 00044
      GO TO 00045
00044 CONTINUE
      WRITE(N01,00047)TIME,1A(LB+2)
00047 FORMAT(1X,E10.4,14,2X,9HFRMD DONE)
00045 CONTINUE
      IF(1TRACE.GE.6.AND.(1TRACE.EQ.1TRACE/2 2))GO TO 00048
      GO TO 00049
00048 CONTINUE
      WRITE(N02,00051)TIME,1A(LB+2)
00051 FORMAT(1X,E10.4,14,2X,9HFRMD DONE)
      CALL ICHECK
00049 CONTINUE
00038 CONTINUE
      RETURN
C
00001 CONTINUE
      IF(MODE.EQ.1)GO TO 00052
      GO TO 00053
00052 CONTINUE
      IF(1A(L+5).GT.0)GO TO 00055
      GO TO 00056
00055 CONTINUE
      L=1A(L+5)
      GO TO 00057
00056 IF(1A(L+4).GT.0)GO TO 00058
      GO TO 00059
00058 CONTINUE
      MODE=2
      L=1A(L+4)
      GO TO 00057
00059 CONTINUE
      MODE=3
00057 CONTINUE
      GO TO 00054
00053 IF(MODE.EQ.2)GO TO 00060

```

FORTRAN Control Programs

```

GO TO 00061
00060 CONTINUE
IF(IA(L+5).GT.0)GO TO 00062
GO TO 00063
00062 CONTINUE
MODE=1
L=IA(L+5)
GO TO 00064
00063 IF(IA(L+4).GT.0)GO TO 00065
GO TO 00066
00065 CONTINUE
L=IA(L+4)
GO TO 00064
00066 CONTINUE
MODE=3
00064 CONTINUE
GO TO 00054
00061 IF(MODE.EQ.3)GO TO 00067
GO TO 00068
00067 CONTINUE
IF(L.NE.LM)GO TO 00069
GO TO 00070
00069 CONTINUE
LF=L
GO TO 00071
00070 CONTINUE
LF=0
00071 CONTINUE
IF(IA(L+2).GT.0)GO TO 00072
GO TO 00073
00072 CONTINUE
MODE=4
L=IA(L+2)
GO TO 00074
00073 IF(IA(L+3).GT.0)GO TO 00075
GO TO 00076
00075 CONTINUE
L=IA(L+3)
GO TO 00074
00076 CONTINUE
IF(L.EQ.LM)GO TO 00077
GO TO 00078
00077 CONTINUE
MODE=4
GO TO 00079
00078 CONTINUE
ASSIGN 00080 TO L00002
GO TO 00002
00080 CONTINUE
00079 CONTINUE
00074 CONTINUE
IF(LF.GT.0)GO TO 00081

```

FORTRAN Control Programs

```

00081 GO TO 00082
00081 CONTINUE
      CALL FSTORE(LF)
00082 CONTINUE
      GO TO 00054
00086 IF(MODE.EQ.4)GO TO 00084
      GO TO 00085
00084 CONTINUE
      IF(IA(L+4).GT.0)GO TO 00086
      GO TO 00087
00086 CONTINUE
      LF=0
      MODE=2
      L=IA(L+4)
      GO TO 00088
00087 IF(IA(L+3).GT.0)GO TO 00089
      GO TO 00090
00089 CONTINUE
      LF=L
      MODE=3
      L=IA(L+3)
      GO TO 00088
00090 IF(IA(L+2).GT.0)GO TO 00091
      GO TO 00092
00091 CONTINUE
      LF=L
      L=IA(L+2)
      GO TO 00088
00092 CONTINUE
      ASSIGN 00093 TO L00002
      GO TO 00002
00093 CONTINUE
00088 CONTINUE
      IF(LF.GT.0)GO TO 00094
      GO TO 00095
00094 CONTINUE
      CALL FSTORE(LF)
00095 CONTINUE
      GO TO 00054
00085 CONTINUE
      ASSIGN 00097 TO L00002
      GO TO 00002
00097 CONTINUE
00054 CONTINUE
      GO TO L00001, (00043)
C
00002 CONTINUE
      WRITE(NO1,00098)
00098 FORMAT(/1X,15H      ERROR      /1X,
X      36HMATERIAL DATA STRUCTURE ERROR (FRMD))
      CALL ERROR
      GO TO L00002, (00080,00093,00097)

```

FORTRAN Control Programs

```

      END
C FSTORE(LOC)
C SUBROUTINE FREES THE STORAGE SPACE AT LOC.
      SUBROUTINE FSTORE(LOC)
      DIMENSION IA(1)
      COMMON RA(1)
      EQUIVALENCE (IA(1),RA(1))
      COMMON /SYSTEM/MFROM,MT0,ICTYPE,1POINT,
      11DUMP,ISECA,ITRACE,LB,LBCA,LB1ST,LBPT,LBUF,LC8B,LE,
      1LEG,LFFB,LFFE,L1,LLE,LFB,LFE,LMS,LNE,LP,LV,M1NNFL,
      1MNSF,NFE,NFL,N11,N12,N01,N02,NREP,NRUN,NSE,TIME,TSTOP
      IF(ITRACE.GE.5)GO TO 00026
      GO TO 00027
00026 CONTINUE
      WRITE(N01,00029)TIME,IA(LB+2)
00029 FORMAT(1X,E10.4,I4,2X,11HFSTORE(LOC))
      WRITE(N01,00030)LOC
00030 FORMAT(1X,I10)
00027 CONTINUE
      IF(ITRACE.GE.6.AND.(ITRACE.EQ.ITRACE/2 2))GO TO 00031
      GO TO 00032
00031 CONTINUE
      WRITE(N02,00034)TIME,IA(LB+2)
00034 FORMAT(1X,E10.4,I4,2X,11HFSTORE(LOC))
      WRITE(N02,00035)LOC
00035 FORMAT(1X,I10)
      CALL ICHECK
00032 CONTINUE
      L=LOC
      N=IA(L+1)
00036 IF(IA(L).GT.0)GO TO 00037
      GO TO 00038
00037 CONTINUE
      L=IA(L)
      N=N+IA(L+1)
      GO TO 00036
00038 CONTINUE
      IA(L)=LFFB
      LFFB=LOC
      NFL=NFL+N
      IF(ITRACE.GE.5)GO TO 00039
      GO TO 00040
00039 CONTINUE
      WRITE(N01,00042)TIME,IA(LB+2)
00042 FORMAT(1X,E10.4,I4,2X,11HFSTORE DONE)
00040 CONTINUE
      IF(ITRACE.GE.6.AND.(ITRACE.EQ.ITRACE/2 2))GO TO 00043
      GO TO 00044
00043 CONTINUE
      WRITE(N02,00046)TIME,IA(LB+2)
00046 FORMAT(1X,E10.4,I4,2X,11HFSTORE DONE)
      CALL ICHECK

```

FORTRAN Control Programs

```

00044 CONTINUE
      RETURN
      END
C IAC(L,I)
C FUNCTION RETURNS INTEGER VALUE OF ITEM I IN CHAIN STORAGE AT L.
      FUNCTION IAC(L,I)
      DIMENSION IAC(L,1)
      COMMON RA(1)
      EQUIVALENCE (IAC(1),RA(1))
      COMMON /SYSTEM/MFROM,MTO,ICTYPE,IPPOINT,
      1DUMP,ISECA,ITRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
      1LEC,LFFB,LFFE,L1,LLE,LLFB,LLFE,LMS,LNE,LP,LV,MINNFI,
      1MNSE,NFE,NFL,N11,N12,N01,N02,NREP,NRUN,NSE,TIME,TSTOP
      1IAC=LCS(L,I)
      RETURN
      END
C ICH(CHAR)
C FUNCTION RETURNS AN INTEGER VALUE CORRESPONDING TO THE POSITION
C OF THE A1 FORMAT CHARACTER ICHAR IN THE STRING: 0 1 ... 9 (BLANK)
C A B ... Z $ + - / = . ' (APOSTROPHE) &
C RETURNS -1 IF ICHAR IS NOT ONE OF THE 49 CHARACTERS IN THE LIST.
      FUNCTION ICH(CHAR)
      DIMENSION I(49)
      DIMENSION IAC(1)
      COMMON RA(1)
      EQUIVALENCE (IAC(1),RA(1))
      COMMON /SYSTEM/MFROM,MTO,ICTYPE,IPPOINT,
      1DUMP,ISECA,ITRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
      1LEC,LFFB,LFFE,L1,LLE,LLFB,LLFE,LMS,LNE,LP,LV,MINNFI,
      1MNSE,NFE,NFL,N11,N12,N01,N02,NREP,NRUN,NSE,TIME,TSTOP
      DATA I /1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,
      11H,1HA,1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI,
      11HJ,1HK,1HL,1HM,1HN,1HO,1HP,1HQ,1HR,1HS,
      11HT,1HU,1HV,1HW,1HX,1HY,1HZ,1H,1H,1H,1H,
      1H,1H,1H,1H,1H,1H,1H,1H,1H,1H,1H,1H,
      K=1
00026 IF(K.LE.49.AND ICHAR.NE.I(K))GO TO 00027
      GO TO 00028
00027 CONTINUE
      K=K+1
      GO TO 00026
00028 CONTINUE
      IF(K.GT.49)GO TO 00029
      GO TO 00030
00029 CONTINUE
      ICH=-1
      GO TO 00030
00030 CONTINUE
      ICH=K-1
00031 CONTINUE
      RETURN
      END

```

FORTRAN Control Programs

```

C ICHECK
C SUBROUTINE RESPONDS TO THE FOLLOWING INTERACTIVE INPUTS:
C ABORT
C EVENTS(N)
C FREE
C IA(N)
C IA(N,M)
C IAC(L,N)
C IAC(L,N,M)
C ICTYPE
C IDUMP
C IPOINT
C ISECA
C ITRACE
C LB
C LBCA
C LBLIST
C LBPT
C LBUE
C LCSD
C LE
C LEC
C LFFB
C LFFE
C LI
C LLE
C LLFB
C LLFE
C LMS
C LNE
C LP
C LV
C MATL(L)
C MFROM
C MINNFL
C MNSE
C MSG
C MTD
C NFE
C NFL
C NI1
C NI2
C NO1
C NO2
C NREP
C NRUN
C NSE
C RA(N)
C RA(N,M)
C RAC(L,N)
C RAC(L,N,M)
C TIME

```

FORTAN Control Programs

```

C TSTOP
SUBROUTINE ICHECK
  DIMENSION IA(1)
  COMMON RA(1)
  EQUIVALENCE ((IA(1),RA(1)))
  COMMON /SYSTEM/MFROM,MT0,ICTYPE,IPPOINT,
  1IDUMP,ISECA,ITRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
  1LEG,LFFB,LFFE,L1,LLE,LLFB,LLFE,LHS,LNE,LP,LV,MINNFL,
  1HNS,NFE,NFL,N11,N12,N01,N02,NREP,NRUN,NSE,TIME,TSTOP
  DIMENSION IPVT(1)
  L1=LBUFF+2
  L80=LBUFF+81
  N11=N11
  N01=N01
  N11=N12
  N01=N02
  ASSIGN 00026 TO L00001
  GO TO 00001
00026 CONTINUE
  N11=N11S
  N01=N01S
  RETURN
C
00001 CONTINUE
  READ(N11,00027)((IA(I),I=L1,L80)
00027 FORMAT(80A1)
00028 IF(IA(L1).NE.1H)GO TO 00029
  GO TO 00030
00029 CONTINUE
  IF(IA(L1).EQ.1HA)GO TO 00031
  GO TO 00032
00031 CONTINUE
  N01=N01S
  WRITE(N01,00034)
00034 FORMAT(1X,15H      ERROR      /1X,10HUSER ABORT)
  CALL ERROR
  GO TO 00033
00032 IF(IA(L1).EQ.1HE)GO TO 00035
  GO TO 00036
00035 CONTINUE
  ASSIGN 00037 TO L00023
  GO TO 00023
00037 CONTINUE
  GO TO 00033
00036 IF(IA(L1).EQ.1HF)GO TO 00038
  GO TO 00039
00038 CONTINUE
  ASSIGN 00040 TO L00024
  GO TO 00024
00040 CONTINUE
  GO TO 00033
00039 IF(IA(L1).EQ.1HI)GO TO 00041

```

```

00041 GO TO 00042
      CONTINUE
      ASSIGN 00043 TO L00002
      GO TO 00002
00043 CONTINUE
      GO TO 00033
00042 IF (IA(L1).EQ.1HL)GO TO 00044
      GO TO 00045
00044 CONTINUE
      ASSIGN 00046 TO L00003
      GO TO 00003
00046 CONTINUE
      GO TO 00033
00045 IF (IA(L1).EQ.1HM)GO TO 00047
      GO TO 00048
00047 CONTINUE
      ASSIGN 00049 TO L00004
      GO TO 00004
00049 CONTINUE
      GO TO 00033
00048 IF (IA(L1).EQ.1HN)GO TO 00050
      GO TO 00051
00050 CONTINUE
      ASSIGN 00052 TO L00005
      GO TO 00005
00052 CONTINUE
      GO TO 00033
00051 IF (IA(L1).EQ.1HR)GO TO 00053
      GO TO 00054
00053 CONTINUE
      ASSIGN 00055 TO L00006
      GO TO 00006
00055 CONTINUE
      GO TO 00033
00054 IF (IA(L1).EQ.1HT)GO TO 00056
      GO TO 00057
00056 CONTINUE
      ASSIGN 00058 TO L00007
      GO TO 00007
00058 CONTINUE
      GO TO 00033
00057 CONTINUE
      ASSIGN 00059 TO L00008
      GO TO 00008
00059 CONTINUE
00033 CONTINUE
      READ(1, 00060) (IA(I), I=L1, L80)
00060 FORMAT(80A1)
      GO TO 00028
00030 CONTINUE
      GO TO L00001, (00026)
c

```


FORTRAN Control Programs

```

00002 CONTINUE
      IF (IA(L1+1).EQ.1HA)GO TO 00061
      GO TO 00062
00061 CONTINUE
      ASSIGN 00064 TO L00009
      GO TO 00009
00064 CONTINUE
      GO TO 00063
00062 IF (IA(L1+1).EQ.1HC)GO TO 00065
      GO TO 00066
00065 CONTINUE
      WRITE(NO1,00067)ICTYPE
00067 FORMAT(1X,6HICTYPE,I2)
      GO TO 00063
00066 IF (IA(L1+1).EQ.1HD)GO TO 00068
      GO TO 00069
00068 CONTINUE
      WRITE(NO1,00070)IDUMP
00070 FORMAT(1X,5HIDUMP,I2)
      GO TO 00063
00069 IF (IA(L1+1).EQ.1HP)GO TO 00071
      GO TO 00072
00071 CONTINUE
      WRITE(NO1,00073)IPPOINT
00073 FORMAT(1X,6HIPPOINT,I6)
      GO TO 00063
00072 IF (IA(L1+1).EQ.1HS)GO TO 00074
      GO TO 00075
00074 CONTINUE
      WRITE(NO1,00076)ISECA
00076 FORMAT(1X,5HISECA,I6)
      GO TO 00063
00075 IF (IA(L1+1).EQ.1HT)GO TO 00077
      GO TO 00078
00077 CONTINUE
      WRITE(NO1,00079)ITRACE
00079 FORMAT(1X,6HITRACE,I2)
      GO TO 00063
00078 CONTINUE
      ASSIGN 00080 TO L00008
      GO TO 00008
00080 CONTINUE
00083 CONTINUE
      GO TO L00002, (00043)
C
00003 CONTINUE
      IF (IA(L1+1).EQ.1HB)GO TO 00081
      GO TO 00082
00081 CONTINUE
      ASSIGN 00084 TO L00010
      GO TO 00010
00084 CONTINUE

```

FORTRAN Control Programs

```

      GO TO 00083
00082 IF (IA(L1+1).EQ.1HC)GO TO 00085
      GO TO 00086
00085 CONTINUE
      WRITE(N01,00087)LCSE
00087 FORMAT(1X,4HLCSE,16)
      GO TO 00083
00086 IF (IA(L1+1).EQ.1HE)GO TO 00088
      GO TO 00089
00088 CONTINUE
      ASSIGN 00090 TO L00011
      GO TO 00011
00090 CONTINUE
      GO TO 00083
00089 IF (IA(L1+1).EQ.1HF)GO TO 00091
      GO TO 00092
00091 CONTINUE
      ASSIGN 00093 TO L00012
      GO TO 00012
00093 CONTINUE
      GO TO 00083
00092 IF (IA(L1+1).EQ.1HI)GO TO 00094
      GO TO 00095
00094 CONTINUE
      WRITE(N01,00096)L1
00096 FORMAT(1X,2HL1,16)
      GO TO 00083
00095 IF (IA(L1+1).EQ.1HL)GO TO 00097
      GO TO 00098
00097 CONTINUE
      ASSIGN 00099 TO L00013
      GO TO 00013
00099 CONTINUE
      GO TO 00083
00098 IF (IA(L1+1).EQ.1HN)GO TO 00100
      GO TO 00101
00100 CONTINUE
      WRITE(N01,00102)LNE
00102 FORMAT(1X,3HLE,16)
      GO TO 00083
00101 IF (IA(L1+1).EQ.1HM)GO TO 00103
      GO TO 00104
00103 CONTINUE
      WRITE(N01,00105)LMS
00105 FORMAT(1X,3HMS,16)
      GO TO 00083
00104 IF (IA(L1+1).EQ.1HP)GO TO 00106
      GO TO 00107
00106 CONTINUE
      WRITE(N01,00108)LP
00108 FORMAT(1X,2HLP,16)
      GO TO 00083

```

FORTRAN Control Programs

```

00107 IF (IA(L1+1).EQ.1HV)GO TO 00109
      GO TO 00110
00109 CONTINUE
      WRITE(NO1,00111)LV
00111 FORMAT(1X,2HLV,16)
      GO TO 00083
00110 CONTINUE
      ASSIGN 00112 TO L00008
      GO TO 00008
00112 CONTINUE
00083 CONTINUE
      GO TO L00003, (00046)
C
00004 CONTINUE
      IF (IA(L1+1).EQ.1HA)GO TO 00113
      GO TO 00114
00113 CONTINUE
      IF (IA(L1+2).NE.1HT)GO TO 00116
      GO TO 00117
00116 CONTINUE
      ASSIGN 00119 TO L00008
      GO TO 00008
00119 CONTINUE
      GO TO 00118
00117 IF (IA(L1+3).NE.1HL)GO TO 00120
      GO TO 00121
00120 CONTINUE
      ASSIGN 00122 TO L00008
      GO TO 00008
00122 CONTINUE
      GO TO 00118
00121 IF (IA(L1+4).NE.1H)GO TO 00123
      GO TO 00124
00123 CONTINUE
      ASSIGN 00125 TO L00008
      GO TO 00008
00125 CONTINUE
00124 CONTINUE
00118 CONTINUE
      LSCAN=L1+5
      ASSIGN 00126 TO L00022
      GO TO 00022
00126 CONTINUE
      CALL WRMD(LNM)
      GO TO 00115
00114 IF (IA(L1+1).EQ.1HF)GO TO 00127
      GO TO 00128
00127 CONTINUE
      WRITE(NO1,00129)MFROM
00129 FORMAT(1X,5HFROM,13)
      GO TO 00115
00128 IF (IA(L1+1).EQ.1H)GO TO 00130

```

FORTRAN Control Programs

```

00130 GO TO 00131
00130 CONTINUE
00132 WRITE(N01,00132)MINNFL
00132 FORMAT(1X,6HMINNFL,16)
00131 GO TO 00115
00131 IF(1A(L1+1).EQ.1HN)GO TO 00133
00131 GO TO 00134
00133 CONTINUE
00135 WRITE(N01,00135)MNSE
00135 FORMAT(1X,4HMNSE,14)
00134 GO TO 00115
00134 IF(1A(L1+1).EQ.1HS)GO TO 00136
00134 GO TO 00137
00136 CONTINUE
00136 ASSIGN 00136 TO L00025
00136 GO TO 00025
00138 CONTINUE
00138 GO TO 00115
00137 IF(1A(L1+1).EQ.1HT)GO TO 00139
00137 GO TO 00140
00139 CONTINUE
00141 WRITE(N01,00141)MTO
00141 FORMAT(1X,3HMT0,13)
00140 GO TO 00115
00140 CONTINUE
00140 ASSIGN 00142 TO L00008
00140 GO TO 00008
00142 CONTINUE
00115 CONTINUE
00115 GO TO L00004, (00049)
C
00005 CONTINUE
00005 IF(1A(L1+1).EQ.1HF)GO TO 00143
00005 GO TO 00144
00143 CONTINUE
00143 ASSIGN 00146 TO L00014
00143 GO TO 00014
00146 CONTINUE
00146 GO TO 00145
00144 IF(1A(L1+1).EQ.1HI)GO TO 00147
00144 GO TO 00148
00147 CONTINUE
00147 IF(1A(L1+2).EQ.1HI)GO TO 00149
00147 GO TO 00150
00149 CONTINUE
00152 WRITE(N01,00152)NI1S
00152 FORMAT(1X,3HNI1,15)
00150 GO TO 00151
00150 IF(1A(L1+2).EQ.1H2)GO TO 00153
00150 GO TO 00154
00153 CONTINUE
00153 WRITE(N01,00155)NI2

```

FORTRAN Control Programs

```

00155 FORMAT(1X,3HN12,15)
GO TO 00151
00154 CONTINUE
ASSIGN 00156 TO L00008
GO TO 00008
00156 CONTINUE
00151 CONTINUE
GO TO 00145
00148 IF(IA(L1+1).EQ.'0')GO TO 00157
GO TO 00158
00157 CONTINUE
IF(IA(L1+2).EQ.'H1')GO TO 00159
GO TO 00160
00159 CONTINUE
WRITE(NO1,00162)NO15
00162 FORMAT(1X,3HN01,15)
GO TO 00161
00160 IF(IA(L1+2).EQ.'H2')GO TO 00163
GO TO 00164
00163 CONTINUE
WRITE(NO1,00165)NO2
00165 FORMAT(1X,3HN02,15)
GO TO 00161
00164 CONTINUE
ASSIGN 00166 TO L00008
GO TO 00008
00166 CONTINUE
00161 CONTINUE
GO TO 00145
00158 IF(IA(L1+1).EQ.'HR')GO TO 00167
GO TO 00168
00167 CONTINUE
ASSIGN 00169 TO L00015
GO TO 00015
00169 CONTINUE
GO TO 00145
00168 IF(IA(L1+1).EQ.'HS')GO TO 00170
GO TO 00171
00170 CONTINUE
WRITE(NO1,00172)NSE
00172 FORMAT(1X,3HNSE,14)
GO TO 00145
00171 CONTINUE
ASSIGN 00173 TO L00008
GO TO 00008
00173 CONTINUE
00145 CONTINUE
GO TO L00005, (00052)
C
00006 CONTINUE
IF(IA(L1+1).EQ.'HA'.AND. IA(L1+2).EQ.'HK')GO TO 00174
GO TO 00175

```

FORTRAN Control Programs

```

00174 CONTINUE
      ASSIGN 00177 TO L00016
      GO TO 00016
00177 CONTINUE
      GO TO 00176
00175 IF (IA(L1+1).EQ.1HA.AND.1A(L1+2).EQ.1HC)GO TO 00178
      GO TO 00179
00178 CONTINUE
      ASSIGN 00180 TO L00017
      GO TO 00017
00180 CONTINUE
      GO TO 00176
00179 CONTINUE
      ASSIGN 00181 TO L00008
      GO TO 00003
00181 CONTINUE
00176 CONTINUE
      GO TO L00006, (00055)
C
00007 CONTINUE
      IF (IA(L1+1).EQ.1HI)GO TO 00182
      GO TO 00183
00182 CONTINUE
      WRITE(NO1,00185)TIME
00185 FORMAT(1X,4HTIME,E12.4)
      GO TO 00184
00183 IF (IA(L1+1).EQ.1HS)GO TO 00186
      GO TO 00187
00186 CONTINUE
      WRITE(NO1,00188)TSTOP
00188 FORMAT(1X,5HTSTOP,E12.4)
      GO TO 00184
00187 CONTINUE
      ASSIGN 00189 TO L00008
      GO TO 00006
00189 CONTINUE
00184 CONTINUE
      GO TO L00007, (00058)
C
00008 CONTINUE
      WRITE(NO1,00190)
00190 FORMAT(1X,5HWHAT?)
      GO TO L00008, (00059,00083,00112,00119,00122,00125,
X      00142,00156,00166,00173,00181,00185)
C
00009 CONTINUE
      IF (IA(L1+2).EQ.1H())GO TO 00191
      GO TO 00192
00191 CONTINUE
      ASSIGN 00194 TO L00018
      GO TO 00018
00194 CONTINUE

```

FORTRAN Control Programs

```

GO TO 00193
00192 IF (IA(LI+2).EQ.1HC)GO TO 00195
GO TO 00196
00195 CONTINUE
ASSIGN 00197 TO L00019
GO TO 00019
00197 CONTINUE
GO TO 00193
00196 CONTINUE
ASSIGN 00198 TO L00008
GO TO 00008
00198 CONTINUE
00193 CONTINUE
GO TO L00009, (00064)
C
00010 CONTINUE
IF (IA(LI+2).EQ.1H )GO TO 00199
GO TO 00200
00199 CONTINUE
WRITE(N01,00202)LB
00202 FORMAT(1X,2HLB,16)
GO TO 00201
00200 IF (IA(LI+2).EQ.1HC)GO TO 00203
GO TO 00204
00203 CONTINUE
WRITE(N01,00205)LBCA
00205 FORMAT(1X,4HLBCA,16)
GO TO 00201
00204 IF (IA(LI+2).EQ.1HL)GO TO 00206
GO TO 00207
00206 CONTINUE
WRITE(N01,00208)LBLIST
00208 FORMAT(1X,6HLBLIST,16)
GO TO 00201
00207 IF (IA(LI+2).EQ.1HP)GO TO 00209
GO TO 00210
00209 CONTINUE
WRITE(N01,00211)LBPT
00211 FORMAT(1X,4HLBPT,16)
GO TO 00201
00210 IF (IA(LI+2).EQ.1HU)GO TO 00212
GO TO 00213
00212 CONTINUE
WRITE(N01,00214)LBUF
00214 FORMAT(1X,4HLBUF,16)
GO TO 00201
00213 CONTINUE
ASSIGN 00215 TO L00008
GO TO 00008
00215 CONTINUE
00201 CONTINUE
GO TO L00010, (00084)

```

FORTRAN Control Programs

```

C
00011 CONTINUE
      IF(IA(L1+2).EQ.1H)GO TO 00216
      GO TO 00217
00216 CONTINUE
      WRITE(NO1,00219)LE
00219 FORMAT(1X,2HLE,16)
      GO TO 00218
00217 IF(IA(L1+2).EQ.1HC)GO TO 00220
      GO TO 00221
00220 CONTINUE
      WRITE(NO1,00222)LEC
00222 FORMAT(1X,3HLEC,16)
      GO TO 00218
00221 CONTINUE
      ASSIGN 00223 TO L00008
      GO TO 00008
00223 CONTINUE
00218 CONTINUE
      GO TO L00011, (00090)
C
00012 CONTINUE
      IF(IA(L1+2).EQ.1HF.AND.1A(L1+3).EQ.1HB)GO TO 00224
      GO TO 00225
00224 CONTINUE
      WRITE(NO1,00227)LFFB
00227 FORMAT(1X,4HLFFB,16)
      GO TO 00226
00225 IF(IA(L1+2).EQ.1HF.AND.1A(L1+3).EQ.1HE)GO TO 00228
      GO TO 00229
00228 CONTINUE
      WRITE(NO1,00230)LFFE
00230 FORMAT(1X,4HLFFE,16)
      GO TO 00226
00229 CONTINUE
      ASSIGN 00231 TO L00008
      GO TO 00008
00231 CONTINUE
00226 CONTINUE
      GO TO L00012, (00093)
C
00013 CONTINUE
      IF(IA(L1+2).EQ.1HE)GO TO 00232
      GO TO 00233
00232 CONTINUE
      WRITE(NO1,00235)LLE
00235 FORMAT(1X,3HLE,16)
      GO TO 00234
00233 IF(IA(L1+2).EQ.1HF.AND.1A(L1+3).EQ.1HB)GO TO 00236
      GO TO 00237
00236 CONTINUE
      WRITE(NO1,00238)LLFB

```


FORTRAN Control Programs

```

00238 FORMAT(1X,4HLLFB,16)
GO TO 00234
00237 IF(1A(L1+2).EQ.1HF.AND.1A(L1+3).EQ.1HE)GO TO 00239
GO TO 00240
00239 CONTINUE
WRITE(NO1,00241)LLFE
00241 FORMAT(1X,4HLLFE,16)
GO TO 00234
00240 CONTINUE
ASSIGN 00242 TO L00008
GO TO 00008
00242 CONTINUE
00234 CONTINUE
GO TO L00013, (00099)
C
00014 CONTINUE
IF(1A(L1+2).EQ.1HE)GO TO 00243
GO TO 00244
00243 CONTINUE
WRITE(NO1,00246)NFE
00246 FORMAT(1X,3HNFE,14)
GO TO 00245
00244 IF(1A(L1+2).EQ.1HL)GO TO 00247
GO TO 00248
00247 CONTINUE
WRITE(NO1,00249)NFL
00249 FORMAT(1X,3HNFL,16)
GO TO 00245
00248 CONTINUE
ASSIGN 00250 TO L00008
GO TO 00008
00250 CONTINUE
00245 CONTINUE
GO TO L00014, (00146)
C
00015 CONTINUE
IF(1A(L1+2).EQ.1HE)GO TO 00251
GO TO 00252
00251 CONTINUE
WRITE(NO1,00254)NREP
00254 FORMAT(1X,4HNREP,14)
GO TO 00253
00252 IF(1A(L1+2).EQ.1HU)GO TO 00255
GO TO 00256
00255 CONTINUE
WRITE(NO1,00257)NRUN
00257 FORMAT(1X,4HNRUN,14)
GO TO 00253
00256 CONTINUE
ASSIGN 00258 TO L00008
GO TO 00008
00258 CONTINUE

```

FORTRAN Control Programs

```

00253 CONTINUE
      GO TO L00015, (00169)
C
00016 CONTINUE
      KIR=2
      ASSIGN 00259 TO L00020
      GO TO 00020
00259 CONTINUE
      GO TO L00016, (00177)
C
00017 CONTINUE
      KIR=2
      ASSIGN 00260 TO L00021
      GO TO 00021
00260 CONTINUE
      GO TO L00017, (00180)
C
00018 CONTINUE
      KIR=1
      ASSIGN 00261 TO L00020
      GO TO 00020
00261 CONTINUE
      GO TO L00018, (00194)
C
00019 CONTINUE
      KIR=1
      ASSIGN 00262 TO L00021
      GO TO 00021
00262 CONTINUE
      GO TO L00019, (00197)
C
00020 CONTINUE
      LSCAN=L1+3
      ASSIGN 00263 TO L00022
      GO TO 00022
00263 CONTINUE
      N=LNH
      LSCAN=LSCAN+1
      IF(1A(LSCAN).EQ.1H))GO TO 00264
      GO TO 00265
00264 CONTINUE
      IF(KIR.EQ.1)GO TO 00267
      GO TO 00268
00267 CONTINUE
      WRITE(N01,00270)N,1A(N)
00270 FORMAT(1X,3H1A(,14,1H),110)
      GO TO 00269
00268 IF(KIR.EQ.2)GO TO 00271
      GO TO 00272
00271 CONTINUE
      WRITE(N01,00273)N,RA(N)
00273 FORMAT(1X,3HRA(,14,1H),E12.4)

```

FORTRAN Control Programs

```

00272 CONTINUE
00269 CONTINUE
      GO TO 00266
00265 IF (IA(LSCAN).EQ.1H) GO TO 00274
      GO TO 00275
00274 CONTINUE
      LSCAN=LSCAN+1
      ASSIGN 00276 TO L00022
      GO TO 00022
00276 CONTINUE
      M=LNH
      IF (KIR.EQ.1) GO TO 00277
      GO TO 00278
00277 CONTINUE
      WRITE(N01,00280)N,M,(IA(I),I=N,M)
00280 FORMAT(1X,3HIA(,I4,1H,,I4,1H),5I10/(14X,5I10))
      GO TO 00279
00278 IF (KIR.EQ.2) GO TO 00281
      GO TO 00282
00281 CONTINUE
      WRITE(N01,00283)N,M,(RA(I),I=N,M)
00283 FORMAT(1X,3HRA(,I4,1H,,I4,1H),5E12.4/(14X,5E12.4))
00282 CONTINUE
00279 CONTINUE
      GO TO 00266
00275 CONTINUE
      ASSIGN 00284 TO L00008
      GO TO 00008
00284 CONTINUE
00286 CONTINUE
      GO TO L00020, (00259,00261)
C
0002 CONTINUE
      IF (IA(L1+3).NE.1H) GO TO 00285
      GO TO 00286
00285 CONTINUE
      ASSIGN 00288 TO L00008
      GO TO 00008
00288 CONTINUE
      GO TO 00287
00286 CONTINUE
      LSCAN=L1+4
      ASSIGN 00289 TO L00022
      GO TO 00022
00289 CONTINUE
      L=LNH
      LSCAN=LSCAN+1
      IF (IA(LSCAN).NE.1H) GO TO 00290
      GO TO 00291
00. CONTINUE
      ASSIGN 00293 TO L00008
      GO TO 00008

```

FORTRAN Control Programs

```

00293 CONTINUE
GO TO 00292
00291 CONTINUE
      LSCAN=LSCAN+1
      ASSIGN 00294 TO L00022
      GO TO 00022
00294 CONTINUE
      N=LNLM
      LSCAN=LSCAN+1
      IF (IA(LSCAN).EQ.1H) GO TO 00295
      GO TO 00296
00295 CONTINUE
      IF (KIR.EQ.1) GO TO 00298
      GO TO 00299
00298 CONTINUE
      WRITE(N01,00301) L,N,IAC(L,N)
00301 FORMAT(1X,4HIAC(,14,1H,,14,1H),110)
      GO TO 00300
00299 IF (KIR.EQ.2) GO TO 00302
      GO TO 00303
00302 CONTINUE
      WRITE(N01,00304) L,N,RAC(L,N)
00304 FORMAT(1X,4HRAC(,14,1H,,14,1H),E12.4)
00303 CONTINUE
00300 CONTINUE
      GO TO 00297
00296 IF (IA(LSCAN).EQ.1H) GO TO 00305
      GO TO 00306
00305 CONTINUE
      LSCAN=LSCAN+1
      ASSIGN 00307 TO L00022
      GO TO 00022
00307 CONTINUE
      M=LNLM
      IF (KIR.EQ.1) GO TO 00308
      GO TO 00309
00308 CONTINUE
      WRITE(N01,00311) L,N,M,(IAC(L,I),I=N,M)
00311 FORMAT(1X,4HIAC(,14,1H,,14,1H,,14,1H),5I10/(14X,5I10))
      GO TO 00310
00309 IF (KIR.EQ.2) GO TO 00312
      GO TO 00313
00312 CONTINUE
      WRITE(N01,00314) L,N,M,(RAC(L,I),I=N,M)
00314 FORMAT(1X,4HRAC(,14,1H,,14,1H,,14,1H),5E12.4/(14X,5E12.4))
00313 CONTINUE
00310 CONTINUE
      GO TO 00297
00306 CONTINUE
      ASSIGN 00315 TO L00008
      GO TO 00008
00315 CONTINUE

```

FORTRAN Control Programs

```

00297 CONTINUE
00292 CONTINUE
00287 CONTINUE
      GO TO L00021, (00260,00262)
C
00022 CONTINUE
      LNM=0
      NCH=1CH(1A(LSCAN))
00316 IF(NCH.GE 0.AND.NCH.LE.9.AND.LSCAN.LE.L80)GO TO 00317
      GO TO 003 8
00317 CONTINUE
      LNM=LNM+NCH
      LSCAN=LSCAN+1
      NCH=1CH(1A(LSCAN))
      GO TO 00316
00318 CONTINUE
      LSCAN=LSCAN-1
      GO TO L00022, (00126,00263,00276,00289,00294,00307)
C
00023 CONTINUE
      IF(1A(L1+1).NE.1HV.OR.1A(L1+2).NE.1HE.OR.1A(L1+3).NE.1HN .OR.1A(
X      L1+4).NE.1HT.OR.1A(L1+5).NE.1HS.OR.1A(L1+6).NE.1H)GO TO
X      00319
      GO TO 00320
00319 CONTINUE
      ASSIGN 00323 TO L00008
      GO TO 00008
00322 CONTINUE
00320 CONTINUE
      LSCAN=L1+7
      ASSIGN 00323 TO L00022
      GO TO 00022
00323 CONTINUE
      WRITE(NO1,00324)LNM
00324 FORMAT(1X,4HNEXT,15,7H EVENTS)
      N=0
      L=LNE
00325 IF(N.LE.LNM.AND.L.GT.0)GO TO 00326
      GO TO 00327
00326 CONTINUE
      WRITE(NO1,00328)L,1A(L),RA(L+1),1A(L+2),1A(L+3),1A(L+4)
00328 FORMAT(1X,2110,5X,E10.4,3110)
      N=N+1
      L=1A(L)
      GO TO 00325
00327 CONTINUE
      GO TO L00023, (00037)
C
00024 CONTINUE
      WRITE(NO1,00329)
00329 FORMAT(1X,10HFREE STORE)
      L=LFFB

```

FORTRAN Control Programs

```

00330 IF(L.GT.0)GO TO 00331
GO TO 00332
00331 CONTINUE
WRITE(NO1,00333)L,IA(L),IA(L+1)
00333 FORMAT(1X,3I10)
L=IA(L)
GO TO 00330
00332 CONTINUE
GO TO L00024, (00040)
C
00025 CONTINUE
IF(1CTYPE.EQ.8)GO TO 00334
GO TO 00335
00334 CONTINUE
WRITE(NO1,00337)IA(IP0INT),IA(IP0INT+1)
00337 FORMAT(1X,10HEVENT CODE,13,2X,7HDATA AT,16)
GO TO 00336
00335 CONTINUE
WRITE(NO1,00339)MFROM,MT0,1CTYPE,IP0INT
00338 FORMAT(1X,5HMFROM,16,2X,3HMT0,16,2X,6H1CTYPE,13,2X,6H1P0INT,16)
IF(1CTYPE.EQ.9.AND.IP0INT.GT.0)GO TO 00339
GO TO 00340
00339 CONTINUE
WRITE(NO1,00342)(IA(IP0INT+I),I=2,9)
00342 FORMAT(1X,7HMESSAGE,816)
00340 CONTINUE
00336 CONTINUE
GO TO L00025, (00138)
END
C IIC(I)
C FUNCTION RETURNS THE ITH INTEGER INITIAL CONDITION OF THE FUNCTION AT LB.
FUNCTION IIC(I)
DIMENSION IA(1)
COMMON RA(1)
EQUIVALENCE (IA(1),RA(1))
COMMON /SYSTEM/MFROM,MT0,1CTYPE,IP0INT,
11DUMP,ISECA,ITRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
11EC,LFFB,LLFE,L1,LLE,LLFB,LLFE,LMS,LNE,LP,LV,MINNFL,
11NSE,NFE,NFL,N11,N12,NO1,NO2,NREP,NRUN,NSE,TIME,TSTOP
IIC=IA(L+I)
RETURN
END
C ILIC(I,J)
C FUNCTION RETURNS THE JTH ELEMENT OF THE INTEGER LIST INITIAL
C CONDITION I OF THE FUNCTION AT LB.
FUNCTION ILIC(I,J)
DIMENSION IA(1)
COMMON RA(1)
EQUIVALENCE (IA(1),RA(1))
COMMON /SYSTEM/MFROM,MT0,1CTYPE,IP0INT,
11DUMP,ISECA,ITRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
11EC,LFFB,LLFE,L1,LLE,LLFB,LLFE,LMS,LNE,LP,LV,MINNFL,

```

FORTRAN Control Programs

```

1MNSE,NFE,NFL,N11,N12,N01,N02,NREP,NRUN,NSE,TIME,TSTOP
1LIC=1A(LCS(1A(L1+1),J+2))
RETURN
END
C ILP(1,J)
C FUNCTION RETURNS THE JTH ELEMENT OF THE INTEGER LIST PARAMETER 1 OF
C THE FUNCTION AT LB.
FUNCTION ILP(1,J)
DIMENSION 1A(1)
COMMON RA(1)
EQUIVALENCE (1A(1),RA(1))
COMMON /SYSTEM/MFROM,MT0,ICTYPE,IPOINT,
11DUMP,ISECA,ITRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
1LEC,LFFB,LFFE,L1,LLE,LLFB,LLFE,LMS,LNE,LP,LV,MINNFL,
1MNSE,NFE,NFL,N11,N12,N01,N02,NREP,NRUN,NSE,TIME,TSTOP
1LP=1A(LCS(1A(LP+1),J+2))
RETURN
END
C ILV(1,J)
C FUNCTION RETURNS THE JTH ELEMENT OF THE INTEGER LIST VARIABLE 1 OF
C THE FUNCTION AT LB.
FUNCTION ILV(1,J)
DIMENSION 1A(1)
COMMON RA(1)
EQUIVALENCE (1A(1),RA(1))
COMMON /SYSTEM/MFROM,MT0,ICTYPE,IPOINT,
11DUMP,ISECA,ITRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
1LEC,LFFB,LFFE,L1,LLE,LLFB,LLFE,LMS,LNE,LP,LV,MINNFL,
1MNSE,NFE,NFL,N11,N12,N01,N02,NREP,NRUN,NSE,TIME,TSTOP
1LV=1A(LCS(1A(LV+1),J+2))
RETURN
END
C INCREC(L1)
C SUBROUTINE INCREMENTS EVENT CHAIN.
SUBROUTINE INCREC(L1)
DIMENSION 1A(1)
COMMON RA(1)
EQUIVALENCE (1A(1),RA(1))
COMMON /SYSTEM/MFROM,MT0,ICTYPE,IPOINT,
11DUMP,ISECA,ITRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
1LEC,LFFB,LFFE,L1,LLE,LLFB,LLFE,LMS,LNE,LP,LV,MINNFL,
1MNSE,NFE,NFL,N11,N12,N01,N02,NREP,NRUN,NSE,TIME,TSTOP
WRITE(59,00026)
00026 FORMAT(/1X,17H      WARNING      /1X,
X      31HINCREC INCREMENTING EVENT CHAIN)
LAST=L1
00027 IF(1A(LAST).NE.0)GO TO 00028
GO TO 00029
00028 CONTINUE
LAST=1A(LAST)
GO TO 00027
00029 CONTINUE

```

FORTRAN Control Programs

```

      CALL ASTORE(L2,ISECA)
      JA(LAST)=L2
      K=TAIL2+2)/5
      NFE=NFE+K
      LPFE=L2+2
      LLFE=LCS(L2,(K-1)*5+1)
      I=1-(1)
00030  I=I+1
      IF((1)-(1-(K)).LE.0)GO TO 00031
      GO TO 00032
00031  CONTINUE
      L=LCS(L2,(I-1)*5+1)
      JA(L)=LCS(L2,I*5+1)
      IF(1.EQ.K)JA(L)=0
      GO TO 00030
00032  CONTINUE
      RETURN
      END
C 1P(1)
C FUNCTION RETURNS THE ITH INTEGER PARAMETER OF THE FUNCTION AT LB.
      FUNCTION 1P(I)
      DIMENSION JA(1)
      COMMON RA(1)
      EQUIVALENCE (JA(1),RA(1))
      COMMON /SYSTEM/MFROM,MT0,ICTYPE,IPOINT,
1  DUMP,ISECA,ITRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
1  LEC,LFFB,LFEE,L1,LE,LLFB,LLFE,LMS,LNE,LP,LV,MINNFI,
1  MNSE,NFE,NFL,N11,N12,N01,N02,NREP,NRUN,NSE,TIME,TSTOP
      IP=JA(LP+I)
      RETURN
      END
C IUNIF(M,ISP)
C FUNCTION RETURNS INTEGER RANDOM VARIABLE WITH UNIFORM DISTRIBUTION.
      FUNCTION IUNIF(M,ISP)
      DATA IX,U/1,0./
      CALL URN(IX,U)
      IUNIF=M-ISP+IFIX(U*FLOAT(2*ISP+1))
      RETURN
      END
C IV(1)
C FUNCTION RETURNS THE ITH INTEGER VARIABLE OF THE FUNCTION AT LB.
      FUNCTION IV(I)
      DIMENSION JA(1)
      COMMON RA(1)
      EQUIVALENCE (JA(1),RA(1))
      COMMON /SYSTEM/MFROM,MT0,ICTYPE,IPOINT,
1  DUMP,ISECA,ITRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
1  LEC,LFFB,LFEE,L1,LE,LLFB,LLFE,LMS,LNE,LP,LV,MINNFI,
1  MNSE,NFE,NFL,N11,N12,N01,N02,NREP,NRUN,NSE,TIME,TSTOP
      IV=JA(LV+I)
      RETURN
      END

```


FORTRAN Control Programs

```

C KEYWD(L)
C FUNCTION COMPARES THE A1 FORMAT CHARACTERS IA(L), IA(L+1), ...
C FOR A MATCH WITH KEYWORDS AND RETURNS THE VALUES BELOW:
C NO MATCH 0
C CREATE 1
C DESTROY 2
C PARAMETER 3
C VARIABLE 4
C REPORT 5
C RUN 6
C END 7

FUNCTION KEYWD(L)
  DIMENSION IA(1)
  COMMON RA(1)
  EQUIVALENCE (IA(1),RA(1))
  COMMON /SYSTEM/MFROM,MT0,ICTYPE,IP0INT,
  1IDUMP,ISECA,ITRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
  1LEC,LFFB,LFFE,L1,LLE,LLFB,LLFE,LMS,LNE,LP,LV,MINNFI,
  1MNSE,NFE,NFL,N11,N12,NO1,NO2,NREP,NRUN,NSE,TIME,TSTOP
  DIMENSION KWDS(10,7)
  DATA KWDS/1HC,1HR,1HE,1HA,1HT,1HE,3 1H,6,
  11HD,1HE,1HS,1HT,1HR,1HO,1HY,2 1H,7,
  11HP,1HA,1HR,1HI,1HM,1HE,1HT,1HE,1HR,9,
  11HV,1FA,1HR,1HI,1HA,1HB,1HL,1HE,1H,8,
  11HR,1HE,1HP,1HO,1HR,1HT,3 1H,6,
  11HR,1HU,1HN,6 1H,3,
  11HE,1HN,1HD,6 1H,3/
  K=0
  J=1
00026 IF(J.LE.7.AND.K.EQ.0)GO TO 00027
  GO TO 00028
00027 CONTINUE
  K=J
  J=1
00029 IF(1.LE.KWDS(10,J))GO TO 00030
  GO TO 00031
00030 CONTINUE
  IF(IA(L-1+J).NE.KWDS(1,J))GO TO 00032
  GO TO 00033
00032 CONTINUE K=0
00033 CONTINUE
  I=I+1
  GO TO 00029
00031 CONTINUE
  J=J+1
  GO TO 00026
00028 CONTINUE
  KEYWD=K
  RETURN
  END
C LCS(LOC,1TH)

```

FORTRAN Control Programs

```

C FUNCTION LOCATES ITH ELEMENT IN CHAIN-BLOCK STORAGE AT LOC.
FUNCTION LCS(LOC,ITH)
  DIMENSION IA(1)
  COMMON RA(1)
  EQUIVALENCE (IA(1),RA(1))
  COMMON /SYSTEM/MFROM,MTD,ICTYPE,IPOINT,
11DUMP,ISECA,ITRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
11LEC,LFFB,LFFE,L1,LLE,LLFB,LLFE,LMS,LNE,LP,LV,MINNFI,
11MNSE,NFE,NFL,N11,N12,N01,N02,NREP,NRUN,NSE,TIME,TSTOP
  LCSB=LOC
  N=0
00026 IF(N+IA(LCSB+1).LT.ITH.AND.LCSB.NE.0)GO TO 00027
  GO TO 00028
00027 CONTINUE
  N=N+IA(LCSB+1)
  LCSB=IA(LCSB)
  GO TO 00026
00028 CONTINUE
  IF(LCSB.LE.0)GO TO 00029
  GO TO 00030
00029 CONTINUE
  LCS=0
  GO TO 00031
00030 CONTINUE
  LCS=LCSB+1+ITH-N
00031 CONTINUE
  RETURN
END

C LIINM(N,M,1)
C FUNCTION RETURNS THE LCS LIST INDEX OF THE ITH ITEM IN THE
C NTH SET OF M ITEMS.
FUNCTION LIINM(N,M,1)
  DIMENSION IA(1)
  COMMON RA(1)
  EQUIVALENCE (IA(1),RA(1))
  COMMON /SYSTEM/MFROM,MTD,ICTYPE,IPOINT,
11DUMP,ISECA,ITRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
11LEC,LFFB,LFFE,L1,LLE,LLFB,LLFE,LMS,LNE,LP,LV,MINNFI,
11MNSE,NFE,NFL,N11,N12,N01,N02,NREP,NRUN,NSE,TIME,TSTOP
  LIINM=(N-1)*M+1+2
  RETURN
END

C MAIN 4/4/78
C PROGRAM CONTROLS MCSS EXECUTION.
PROGRAM MAIN(INPT,OTPT,TTY,TAPE5=INPT,TAPE6=OTPT,TAPE59=TTY)
  DIMENSION IA(20000)
  COMMON RA(20000)
  EQUIVALENCE (IA(1),RA(1))
  COMMON /SYSTEM/MFROM,MTD,ICTYPE,IPOINT,
11DUMP,ISECA,ITRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
11LEC,LFFB,LFFE,L1,LLE,LLFB,LLFE,LMS,LNE,LP,LV,MINNFI,
11MNSE,NFE,NFL,N11,N12,N01,N02,NREP,NRUN,NSE,TIME,TSTOP

```

FORTRAN Control Programs

```

C
C      ASSIGN I/O DEVICE NUMBERS, TIME, AND ITRACE.
      NI1=5
      NI2=59
      NO1=6
      NO2=59
      TIME=0.
      ITRACE=0
C
C      INITIALIZE DYNAMIC STORAGE
      LFFB=1
      IA(LFFB)=0
      IA(LFFB+1)=19998
      LFFB=1
      NFL=19998
      MINNFL=NFL
      LBGA=1
C
C      CREATE BLOCK POINTER TABLE
      CALL ASTORE(LBPT,101)
      IA(LBPT+2)=IA(LBPT+2)-1
      CALL ZTABLE(LBPT)
C
C      CREATE MESSAGE STACK
      CALL ASTORE(LMS,42)
      IA(LMS+2)=IA(LMS+2)-2
      IA(LMS+3)=0
C
C      CREATE EVENT CHAIN
      CALL ASTORE(LEC,500)
      ISECA=100
C
C      CREATE BUFFER
      CALL ASTORE(LBUF,60)
C
C      ASSIGN RANDOM NUMBER SEED.
      U=0.
      CALL URN(-1,U)
C
C      REQUEST USER INPUT LINE DISPLAY OPTION
      IDSPL=-1
00026 IF(IDSPL.LT.0.OR.IDSPL.GT.3)GO TO 00027
      GO TO 00028
00027 CONTINUE
      WRITE(NO2,00029)
00029 FORMAT(1X,13HINPUT DISPLAY)
      WRITE(NO2,00030)
00030 FORMAT(1X,12H0 = NO LINES /1X,13H1 = RUN LINES /1X,
      X 17H2 = COMMAND LINES /1X,19H3 = ALL INPUT LINES)
      READ(NI2,00031)IDSPL
00031 FORMAT(11)
      GO TO 00026

```

FORTRAN Control Programs

```

00028 CONTINUE
C
C      PRINT HEADING AND USER COMMENTS DOWN TO FIRST KEYWORD.
      WRITE(N01,00032)
00032 FORMAT(1H1,
X56HMATERIAL CONTROL SYSTEM SIMULATOR   VERSION 1.0   4/4/78)
      ASSIGN 00033 TO L00001
      GO TO 00001
00033 CONTINUE
C
C      EXECUTE JOB STEPS
00034 IF(KEYWD(LBUF+2).NE.7)GO TO 00035
      GO TO 00036
00035 CONTINUE
      WRITE(N01,00037)((1A(LBUF+1+I),I=1,80)
00037 FORMAT(1X,80A1)
      IF(IDSPL.GT.0)GO TO 00038
      GO TO 00039
00038 CONTINUE
      IF(IDSPL.EQ.1)GO TO 00041
      GO TO 00042
00041 CONTINUE
      IF(KEYWD(LBUF+2).EQ.6)GO TO 00044
      GO TO 00045
00044 CONTINUE
      WRITE(N02,00047)((1A(LBUF+1+I),I=1,80)
00047 FORMAT(1X,80A1)
00045 CONTINUE
      GO TO 00043
00042 CONTINUE
      WRITE(N02,00048)((1A(LBUF+1+I),I=1,80)
00048 FORMAT(1X,80A1)
00043 CONTINUE
00039 CONTINUE
      ASSIGN 00049 TO L00002
      GO TO 00002
00049 CONTINUE
      ASSIGN 00050 TO L00001
      GO TO 00001
00050 CONTINUE
      GO TO 00034
00036 CONTINUE
C
C      PRINT MINIMUM NUMBER OF FREE LOCATIONS
      WRITE(N01,00051)MINNFL
00051 FORMAT(1X,3HEND/1X,35HMINIMUM NUMBER OF FREE LOCATIONS = ,15)
C
C      CALL EXIT
C
00001 CONTINUE
      READINIT,00052)((1A(LBUF+1+I),I=1,80)
      IF(IFIX(EOF(N1)).NE.0)GO TO 99999

```

FORTRAN Control Programs

```

00052 FORMAT(80A1)
00053 IF(KEYWD(LBUF+2).EQ.0)GO TO 00054
GO TO 00055
00054 CONTINUE
WRITE(NO1,00056)((IA(LBUF+1+I),I=1,80)
00056 FORMAT(1X,80A1)
IF(108PL.EQ.3)GO TO 00057
GO TO 00058
00057 CONTINUE
WRITE(NO2,00060)((IA(LBUF+1+I),I=1,80)
00060 FORMAT(1X,80A1)
00058 CONTINUE
READ(N11,00061)((IA(LBUF+1+I),I=1,80)
IF((IFIX(EOP(N11)).NE.0)GO TO 99999
00061 FORMAT(80A1)
GO TO 00053
00055 CONTINUE
GO TO L00001, (00033,00050)
GO TO 00062
99999 IEOF=1
00062 CONTINUE
WRITE(NO1,00063)
00063 FORMAT(/1X,15H      ERROR      /1X,
X      34HMAIN NO END COMMAND IN INPUT FILE)
CALL ERROR
C
00002 CONTINUE
K=KEYWD(LBUF+2)
IF(K.EQ.1)GO TO 00064
GO TO 00065
00064 CONTINUE
CALL FBLANK(0,1A,INTD,REALD)
CALL GETV(0,1A,MTO,REAL)
CALL GETV(0,1A,IP,INT,REAL)
IF(MTO.LE.0.OR.MTO.GT.100)GO TO 00067
GO TO 00068
00067 CONTINUE
WRITE(NO1,00070)
00070 FORMAT(/1X,15H      ERROR      /1X,
X      29HMAIN ILLEGAL FUNCTION NUMBER)
CALL ERROR
00068 CONTINUE
IF(IA(LCS(LBPT,MTO+1)).NE.0)GO TO 00071
GO TO 00072
00071 CONTINUE
WRITE(NO1,00074)
00074 FORMAT(/1X,15H      ERROR      /1X,
X      34HMAIN ATTEMPT TO CREATE A FUNCTION THAT ALREADY EXISTS)
CALL ERROR
00072 CONTINUE
MFROM=0
ICTYPE=1

```

FORTRAN Control Programs

```

      CALL SEND
      GO TO 00066
00065 IF(K.EQ.2)GO TO 00075
      GO TO 00076
00075 CONTINUE
      CALL FBLANK(0,IA,INTD,REALD)
      CALL GETV(0,IA,INT,REAL)
      IF(IA(LCS(LBPT,INT+1)).EQ.0)GO TO 00077
      GO TO 00078
00077 CONTINUE
      WRITE(NO1,00080)
00080 FORMAT(/1X,15H      ERROR      /1X,
X      35HMAIN ATTEMPT TO DESTROY A FUNCTION THAT DOES NOT EXIST)
      CALL ERROR
      GO TO 00079
00078 CONTINUE
      MFROM=0
      MTO=INT
      ICTYPE=2
      IPOINT=0
      CALL SEND
00079 CONTINUE
      GO TO 00066
00076 IF(K.GE.3.AND.K.LE.4)GO TO 00081
      GO TO 00082
00081 CONTINUE
      CALL FBLANK(0,IA,INTD,REALD)
      CALL GETV(0,IA,INT,REAL)
      MFROM=0
      MTO=INT
      ICTYPE=K
      IPOINT=IDSPL
      CALL SEND
      GO TO 00066
00082 IF(K.EQ.5)GO TO 00083
      GO TO 00084
00083 CONTINUE
      CALL FBLANK(0,IA,INTD,REALD)
      MFROM=0
      CALL GETV(0,IA,MTO,REAL)
      ICTYPE=5
      CALL SEND
      GO TO 00066
00084 IF(K.EQ.6)GO TO 00085
      GO TO 00086
00085 CONTINUE
      CALL FBLANK(0,IA,INTD,REALD)
      CALL GETV(0,IA,NREP,REAL)
      CALL GETV(0,IA,INT,TSTOP)
      CALL GETV(0,IA,IOUMP,REAL)
      CALL GETV(0,IA,ITRACE,REAL)
      CALL GETV(0,IA,IOUT,REAL)

```

FORTRAN Control Programs

```

        ASSIGN 00087 TO L00003
        GO TO 00003
00087 CONTINUE
        NRUN=1
00088 IF(NRUN.LE.NREP)GO TO 00089
        GO TO 00090
00089 CONTINUE
        TIME=0.
        ASSIGN 00091 TO L00006
        GO TO 00006
00091 CONTINUE
        ASSIGN 00092 TO L00004
        GO TO 00004
00092 CONTINUE
        IF(NRUN.EQ.1.AND.IDUMP.EQ.1)GO TO 00093
        GO TO 00094
00093 CONTINUE
        CALL DMPSYS
00094 CONTINUE
        IF(NREP.GT.1.AND.NRUN.EQ.1)GO TO 00095
        GO TO 00097
00096 CONTINUE
        ASSIGN 00099 TO L00007
        GO TO 00007
00099 CONTINUE
        GO TO 00098
00097 IF(NRUN.GT.1)GO TO 00100
        GO TO 00101
00100 CONTINUE
        ASSIGN 00102 TO L00008
        GO TO 00008
00102 CONTINUE
00101 CONTINUE
00098 CONTINUE
        CALL CLOCK
        NRUN=NRUN+1
        GO TO 00088
00090 CONTINUE
        IF(IGOUT.EQ.1)GO TO 00103
        GO TO 00104
00103 CONTINUE
        ASSIGN 00106 TO L00005
        GO TO 00005
00106 CONTINUE
00104 CONTINUE
        CALL FSTORE(LBLIST)
        LBLIST=0
        GO TO 00066
00066 CONTINUE
        WRITE(N01,00107)
00107 FORMAT(/1X,15H      ERROR      /1X,21HMAIN  ILLEGAL KEYWORD)
        CALL ERROR

```

FORTRAN Control Programs

```

00066 CONTINUE
      GO TO L00002, (00049)
C
00003 CONTINUE
      N=0
      K=IA(LBPT+2)
      I=1-(I)
00108 I=I+1
      IF((I) (I-(K)).LE.0)GO TO 00109
      GO TO 00110
00109 CONTINUE
      IF(IA(LCS(LBPT,I+1)).GT.0)N=N+1
      GO TO 00108
00110 CONTINUE
      CALL ASTORE(LBLIST,N+2)
      IA(LBLIST+2)=IA(LBLIST+2)-2
      M=0
      I=1-(I)
00111 I=I+1
      IF((I) (I-(K)).LE.0)GO TO 00112
      GO TO 00113
00112 CONTINUE
      IF(IA(LCS(LBPT,I+1)).GT.0)GO TO 00114
      GO TO 00115
00114 CONTINUE
      M=M+1
      IA(LCS(LBLIST,M+2))=I
00115 CONTINUE
      GO TO 00111
00113 CONTINUE
      IA(LBLIST+3)=M
      GO TO L00003, (00087)
C
00004 CONTINUE
      K=IA(LBLIST+3)
      MFROM=0
      IPOINT=0
      I=1-(I)
00117 I=I+1
      IF((I) (I-(K)).LE.0)GO TO 00118
      GO TO 00119
00118 CONTINUE
      MTO=IA(LCS(LBLIST,I+2))
      ICTYPE=6
      CALL SEND
      GO TO 00117
00119 CONTINUE
      GO TO L00004, (00092)
C
C
00005 CONTINUE
      MFROM=0

```


FORTRAN Control Programs

```

IPOINT=0
K=IA(LBLIST+3)
WRITE(N01,00120)
00120 FORMAT(//34H FUNCTION TYPE ENTRIES FRACTION)
N=0
I=1-(1)
00121 I=I+1
IF((1) (1-(K)).LE.0)GO TO 00122
GO TO 00123
00122 CONTINUE
M=IA(LCS(LBLIST,I+2))
CALL FPNTRS(M)
N=N+IA(LV)
GO TO 00121
00123 CONTINUE
I=1-(1)
00124 I=I+1
IF((1) (1-(K)).LE.0)GO TO 00125
GO TO 00126
00125 CONTINUE
M=IA(LCS(LBLIST,I+2))
CALL FPNTRS(M)
B=FLOAT(IA('V'))/FLOAT(N)
WRITE(N01,00127)M,IA(LB),IA(LV),B
00127 FORMAT(1X,I8,I6,I9,F10.4)
GO TO 00124
00126 CONTINUE
I=1-(1)
00128 I=I+1
IF((1) (1-(K)).LE.0)GO TO 00129
GO TO 00130
00129 CONTINUE
MTD=IA(LCS(LBLIST,I+2))
ICTYPE=5
CALL SEND
GO TO 00128
00130 CONTINUE
GO TO L00005, (00106)
C
00006 CONTINUE
NFE=IA(LEC+1)/5
LFFE=LEC+2
LLFE=LCS(LEC,(NFE-1) 5+1)
NSE=0
MNSE=0
LNE=0
LLE=0
I=1-(1)
00131 I=I+1
IF((1) (1-(NFE)).LE.0)GO TO 00132
GO TO 00133
00132 CONTINUE

```

FORTRAN Control Programs

```

      L=LCS(LEC,(I-1) S+1)
      IA(L)=LCS(LEC,I S+1)
      IF(I.EQ.NFE)IA(L)=0
00133 GO TO 00131
      CONTINUE
      GO TO L00006, (00091)
C
00007 CONTINUE
      IF(IA(LFFB+1).EQ.NFL)GO TO 00134
      GO TO 00135
00134 CONTINUE
      LFFBS=LFFB
      LLFBS=LLFB
      NFLS=NFL
      MINNFS=MINNFL
      GO TO 00136
00135 CONTINUE
      LFFBS=0
      LLFBS=0
      NFLS=0
      MINNFS=0
00136 CONTINUE
      GO TO L00007, (00099)
C
00008 CONTINUE
      IF(LFFBS.GT.0)GO TO 00137
      GO TO 00138
00137 CONTINUE
      LFFB=LFFBS
      LLFB=LLFBS
      NFL=NFLS
      IF(MINNFL.GT.MINNFS)GO TO 00140
      GO TO 00141
00140 CONTINUE
      MINNFL=MINNFS
00141 CONTINUE
      IA(LFFB)=0
      IA(LFFB+1)=NFL
00138 CONTINUE
      GO TO L00008, (00102)
      END
C MZUR(LM,LMSP,N,MT,NC,A,NCT,AT)
C SUBROUTINE MEASURES MATERIAL AT LM ACCORDING TO THE NTH
C MEASUREMENT SPECIFICATION AT LMSP AND RETURNS THE MEASURED
C MATERIAL TYPE MT, COUNT NC, MASS A, TRUE COUNT NCT,
C AND TRUE MASS AT.
      SUBROUTINE MZUR(LM,LMSP,N,MT,NC,A,NCT,AT)
      DIMENSION IA(1)
      COMMON RA(1)
      EQUIVALENCE (IA(1),RA(1))
      COMMON /SYSTEM/MFROM,MT0,ICTYPE,IPOINT,
      I1DUMP,ISECA,ITRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,

```

FORTAN Control Programs

```

      ILEC, LFFB, LFFE, LI, LLE, LLFB, LLFF, LMS, LME, LP, LV, MINIFL,
      MNSE, NFE, NFI, NI1, NI2, NO1, NO2, NREP, NRUN, NSE, TIME, TSTOP
      IF (ITRACE GE 5) GO TO 00026
      GO TO 00027
00026 CONTINUE
      WRITE(N01,00029)TIME,IA(LB+2)
00029 FORMAT(1X,E10.4,I4,2X,26HZUR(LM,LMSP,N, , , , , ))
      WRITE(N01,00030)LM,LMSP,N
00030 FORMAT(1X,3I10)
00027 CONTINUE
      IF (ITRACE GE 6 AND (ITRACE EQ ITRACE/2 2)) GO TO 00031
      GO TO 00032
00031 CONTINUE
      WRITE(N02,00034)TIME,IA(LB+2)
00034 FORMAT(1X,E10.4,I4,2X,26HZUR(LM,LMSP,N, , , , , ))
      WRITE(N02,00035)LM,LMSP,N
00035 FORMAT(1X,3I10)
      CALL ICHECK
00032 CONTINUE
      MT=RAC(LMSP,LIIN(N,6,1))
      NC=0
      NCT=0
      A=0
      AT=0
      MCDE=1
      L=LM
00036 IF (L.NE.LM.OR.MODE.NE.4) GO TO 00037
      GO TO 00038
00037 CONTINUE
      ASSIGN 00039 TO L00001
      GO TO 00001
00039 CONTINUE
      GO TO 00036
00038 CONTINUE
      ASSIGN 00040 TO L00003
      GO TO 00003
00040 CONTINUE
      IF (ITRACE GE 5) GO TO 00041
      GO TO 00042
00041 CONTINUE
      WRITE(N01,00044)TIME,IA(LB+2)
00044 FORMAT(1X,E10.4,I4,2X,26HZUR( , , , MT,NC,A,NCT,AT))
      WRITE(N01,00045)MT,NC,A,NCT,AT
00045 FORMAT(1X,2I10,E12.4,I10,E12.4)
00042 CONTINUE
      IF (ITRACE GE 6 AND (ITRACE EQ ITRACE/2 2)) GO TO 00046
      GO TO 00047
00046 CONTINUE
      WRITE(N02,00049)TIME,IA(LB+2)
00049 FORMAT(1X,E10.4,I4,2X,26HZUR( , , , MT,NC,A,NCT,AT))
      WRITE(N02,00050)MT,NC,A,NCT,AT
00050 FORMAT(1X,2I10,E12.4,I10,E12.4)

```

FORTRAN Control Programs

```

      CALL ICHECK
00047 CONTINUE
      RETURN
C
00001 CONTINUE
      IF(MODE.EQ.1)GO TO 00051
      GO TO 00052
00051 CONTINUE
      IF(IA(L+5).GT.0)GO TO 00054
      GO TO 00055
00054 CONTINUE
      L=IA(L+5)
      GO TO 00056
00055 IF(IA(L+4).GT.0)GO TO 00057
      GO TO 00058
00057 CONTINUE
      MODE=2
      L=IA(L+4)
      GO TO 00056
00058 CONTINUE
      MODE=3
00056 CONTINUE
      GO TO 00053
00052 IF(MODE.EQ.2)GO TO 00059
      GO TO 00060
00059 CONTINUE
      IF(IA(L+5).GT.0)GO TO 00061
      GO TO 00062
00061 CONTINUE
      MODE=1
      L=IA(L+5)
      GO TO 00063
00062 IF(IA(L+4).GT.0)GO TO 00064
      GO TO 00065
00064 CONTINUE
      L=IA(L+4)
      GO TO 00063
00065 CONTINUE
      MODE=3
00063 CONTINUE
      GO TO 00053
00060 IF(MODE.EQ.3)GO TO 00066
      GO TO 00067
00066 CONTINUE
      IF(IA(L+6).EQ.0.AND. IA(L+7).EQ.MT)GO TO 00068
      GO TO 00069
00068 CONTINUE
      ASSIGN 00071 TO L00002
      GO TO 00002
00071 CONTINUE
00069 CONTINUE
      IF(IA(L+2).GT.0)GO TO 00072

```

FORTRAN Control Programs

```

00072 GO TO 00073
      CONTINUE
      MODE=4
      L=IA(L+2)
00073 IF(IA(L+3).GT.0)GO TO 00075
      GO TO 00076
00075 CONTINUE
      L=IA(L+3)
      GO TO 00074
00076 CONTINUE
      IF(L.EQ.LN)GO TO 00077
      GO TO 00078
00077 CONTINUE
      MODE=4
      GO TO 00079
00078 CONTINUE
      ASSIGN 00080 TO L00003
      GO TO 00083
00080 CONTINUE
00079 CONTINUE
00074 CONTINUE
      GO TO 00053
00067 IF(MODE.EQ.4)GO TO 00081
      GO TO 00082
00081 CONTINUE
      IF(IA(L+4).GT.0)GO TO 00083
      GO TO 00084
00083 CONTINUE
      MODE=2
      L=IA(L+4)
      GO TO 00085
00084 IF(IA(L+3).GT.0)GO TO 00086
      GO TO 00087
00086 CONTINUE
      MODE=3
      L=IA(L+3)
      GO TO 00085
00087 IF(IA(L+2).GT.0)GO TO 00088
      GO TO 00089
00088 CONTINUE
      L=IA(L+2)
      GO TO 00085
00089 CONTINUE
      ASSIGN 00090 TO L00004
      GO TO 00004
00090 CONTINUE
00085 CONTINUE
      GO TO 00053
00082 CONTINUE
      ASSIGN 00091 TO L00004
      GO TO 00004

```

FORTRAN Control Programs

```

00091 CONTINUE
00093 CONTINUE
GO TO L00001, (00039)

C
00002 CONTINUE
NCT=NCT+IA(L+8)
AT=AT+RA(L+9)
GO TO L00002, (00071)

C
00003 CONTINUE
NC=NCT+IUNIF(IFIX(RAC(LMSP,LINM(N,6,2))),
1 IFIX(RAC(LMSP,LINM(N,6,3))),
1 A=AT+RAC(LMSP,LINM(N,6,4))
1+RNORM(RAC(LMSP,LINM(N,6,5)),RAC(LMSP,LINM(N,6,6)))
GO TO L00003, (00040,00080)

C
00004 CONTINUE
WRITE(N01,00092)
00092 FORMAT(1X,15H ERROR /1X
X 36H MATERIAL DATA STRUCTURE ERROR (MZUR))
CALL ERROR
GO TO L00004, (00090,00091)
END

C NCBM(LOC,NC,BM)
C SUBROUTINE RETURNS THE NUMBER OF CONTAINERS NC AND THE TOTAL BULK
C MASS BM AT LOCATION LOC.
SUBROUTINE NCBM(LOC,NC,BM)
DIMENSION IA(1)
COMMON RA(1)
EQUIVALENCE (IA(1),RA(1))
COMMON /SYSTEM/MFROM,MT0,ICTYPE,IP0,INT,
1 DUMP,SECA,ITRACE,LB,LSCA,LBLIST,LBPT,LBUF,LCSB,LE,
1 LEC,LFFB,LFFE,L1,L2,LFB,LFE,LMS,LNE,LP,LV,MINNFL,
1 MNSE,NFE,NFL,N11,N12,N01,N02,NREP,NRUN,NSE,TIME,TSTOP
IF(ITRACE.GE.5)GO TO 00026
GO TO 00027

00026 CONTINUE
WRITE(N01,00029)TIME,IA(LB+2)
00029 FORMAT(1X,E10.4,I4,2X,13HNCBM(LOC, , ))
WRITE(N01,00030)LOC
00030 FORMAT(1X,I10)
00027 CONTINUE
IF(ITRACE.GE.6.AND.(ITRACE.EQ.ITRACE/2))GO TO 00031
GO TO 00032

00031 CONTINUE
WRITE(N02,00034)TIME,IA(LB+2)
00034 FORMAT(1X,E10.4,I4,2X,13HNCBM(LOC, , ))
WRITE(N02,00035)LOC
00035 FORMAT(1X,I10)
CALL ICHECK
00032 CONTINUE
IF(IA(LOC+6).EQ.1)GO TO 00036

```

FORTRAN Control Programs

```

GO TO 00037
00036 CONTINUE
      NC=0
      BM=RA(L0C+9)
GO TO 00038
00037 IF (IA(L0C+8).EQ.2)GO TO 00039
GO TO 00040
00039 CONTINUE
      L=IA(L0C+5)
      IF (L.LE.0)GO TO 00041
GO TO 00042
00041 CONTINUE
      WRITE(N01,00044)
00044 FORMAT(/1X,15H      ERROR      /1X,27HNOCBM NO SUBDIVISION OF LOC)
      CALL ERROR
00042 CONTINUE
      IF (IA(L+6).EQ.1)GO TO 00045
GO TO 00046
00045 CONTINUE
      BM=RA(L+9)
      NC=0
00048 IF (IA(L+4).GT.0)GO TO 00049
GO TO 00050
00049 CONTINUE
      NC=NC+1
      L=IA(L+4)
GO TO 00048
00050 CONTINUE
GO TO 00047
00046 IF (IA(L+6).EQ.2)GO TO 00051
GO TO 00052
00051 CONTINUE
      BM=0.
      NC=1
00053 IF (IA(L+4).GT.0)GO TO 00054
GO TO 00055
00054 CONTINUE
      NC=NC+1
      L=IA(L+4)
GO TO 00053
00055 CONTINUE
GO TO 00047
00052 CONTINUE
      WRITE(N01,00056)
00056 FORMAT(/1X,15H      ERROR      /1X,26HNOCBM ILLEGAL SUBLOC LEVEL)
      CALL ERROR
00047 CONTINUE
GO TO 00038
00040 CONTINUE
      WRITE(N01,00057)
00057 FORMAT(/1X,15H      ERROR      /1X,23HNOCBM ILLEGAL LOC LEVEL)
      CALL ERROR

```

FORTAN Control Programs

```

00038 CONTINUE
      IF(I TRACE.GE.5)GO TO 00058
      GO TO 00059
00058 CONTINUE
      WRITE(NC1,00061)TIME,IA(LB+2)
00061 FORMAT(1X,E10.4,14,2X,15HNCBM(LCC,NC,3M))
      WRITE(NC1,00062)LCC,NC,3M
00062 FORMAT(1X,2110,E11.4)
00059 CONTINUE
      IF(I TRACE.GE.6.AND.(I TRACE.EQ.I TRACE/2 2))GO TO 00063
      GO TO 00064
00063 CONTINUE
      WRITE(NC2,00066)TIME,IA(LB+2)
00066 FORMAT(1X,E10.4,14,2X,15HNCBM(LCC,NC,3M))
      WRITE(NC2,00067)LCC,NC,3M
00067 FORMAT(1X,2110,E11.4)
      CALL ICHECK
00064 CONTINUE
      RETURN
      END
C NSTATS(N)
C FUNCTION RETURNS THE NUMBER OF STATISTICS REQUIRED TO DEFINE
C DISTRIBUTION TYPE N
      FUNCTION NSTATS(N)
      DIMENSION IA(1)
      COMMON RA(1)
      EQUIVALENCE (IA(1),RA(1))
      COMMON /SYSTEM/MFROM,MT0,ICTYPE,IPOINT,
11DUMP,ISECA,I TRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
1LEC,LFFB,LFFE,L1,LLE,LFB,LFE,LMS,LNE,LPLV,MINNFI,
1MNSF,NFE,NFL,N11,N12,N01,N02,NREP,NRUN,NSF,TIME,TSTOP
      DIMENSION NS(4)
      DATA NS/2,2,4,1/
      NSTATS=NS(N)
      RETURN
      END
C PCCM(NT0,NCTYPE,11,12,13,14,15,16,17,18)
C SUBROUTINE PREPARES COMMUNICATION FOR TRANSMISSION.
      SUBROUTINE PCCM(NT0,NCTYPE,11,12,13,14,15,16,17,18)
      DIMENSION IA(1)
      COMMON RA(1)
      EQUIVALENCE (IA(1),RA(1))
      COMMON /SYSTEM/MFROM,MT0,ICTYPE,IPOINT,
11DUMP,ISECA,I TRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
1LEC,LFFB,LFFE,L1,LLE,LFB,LFE,LMS,LNE,LPLV,MINNFI,
1MNSF,NFE,NFL,N11,N12,N01,N02,NREP,NRUN,NSF,TIME,TSTOP
      MFROM=IA(LB+2)
      MT0=NT0
      ICTYPE=NCTYPE
      CALL ASTORE(IPOINT,8)
      IA(IPOINT+2)=11
      IA(IPOINT+3)=12

```


FORTRAN Control Programs

```

IA(IPOINT+4)=13
IA(IPOINT+5)=14
IA(IPOINT+6)=15
IA(IPOINT+7)=16
IA(IPOINT+8)=17
IA(IPOINT+9)=18
RETURN
END
C PHLB
C SUBROUTINE PRINTS THE HEADER OF THE FUNCTION AT LB IN DUMP FORMAT.
SUBROUTINE PHLB
  DIMENSION IA(1)
  COMMON RA(1)
  EQUIVALENCE (IA(1),RA(1))
  COMMON /SYSTEM/MFROM,MT0,ICTYPE,IPOINT,
  1IDUMP,ISECA,ITRACE,LB,LBCA,LB1ST,LBPT,LBUF,LCSB,LE,
  1LEC,LFFB,LFFE,L1,LLE,LFFB,LFFE,LMS,LINE,LP,LV,MINNFI,
  1MNSI,NFE,NFL,N11,N12,NO1,NO2,NREP,NRUN,NSI,TIME,TSTOP
  WRITE(NO1,00026)LB,IA(LB)
00026 FORMAT(1X,5HLB+00,5X,15,112)
  I=1-11
00027 I=I+1
  IF((1)-(1-(4)).LE.0)GO TO 00028
  GO TO 00029
00028 CONTINUE
  L=LB+1
  WRITE(NO1,00030)I,L,IA(L)
00030 FORMAT(3X,1H+,12,5X,15,112)
  GO TO 00027
00029 CONTINUE
  RETURN
  END
C PICLB(IPVT)
C SUBROUTINE PRINTS THE INITIAL CONDITION VALUES OF THE FUNCTION AT LB IN
C DUMP FORMAT
SUBROUTINE PICLB(IPVT)
  DIMENSION IPVT(1)
  DIMENSION IA(1)
  COMMON RA(1)
  EQUIVALENCE (IA(1),RA(1))
  COMMON /SYSTEM/MFROM,MT0,ICTYPE,IPOINT,
  1IDUMP,ISECA,ITRACE,LB,LBCA,LB1ST,LBPT,LBUF,LCSB,LE,
  1LEC,LFFB,LFFE,L1,LLE,LFFB,LFFE,LMS,LINE,LP,LV,MINNFI,
  1MNSI,NFE,NFL,N11,N12,NO1,NO2,NREP,NRUN,NSI,TIME,TSTOP
  I=1-11
00026 I=I+1
  IF((1)-(1-(IA(LB+4))).LE.0)GO TO 00027
  GO TO 00028
00027 CONTINUE
  CALL PIDF(L1-1+1,IPVT(IA(LB+3)+1),1-1)
  GO TO 00026
00028 CONTINUE

```

FORTRAN Control Programs

```

      RETURN
      END
C PIDF(L,K,N)
C SUBROUTINE PRINTS THE ITEM AT L OF TYPE K IN DUMP FORMAT.
      SUBROUTINE PIDF(L,K,N)
      DIMENSION IA(1)
      COMMON RA(1)
      EQUIVALENCE (IA(1),RA(1))
      COMMON /SYSTEM/MFROM,MT0,ICTYPE,IP0INT,
      1IDUMP,ISECA,ITRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
      1LEC,LFFB,LFPE,L1,LLE,LLFB,LLFE,LMS,LNE,LP,LV,MINHFL,
      1MINSE,NFE,NFL,N11,N12,N01,NOZ,NREP,NRUN,NSE,TIME,TSTOP
      IF(L.EQ.LP)GO TO 00026
      GO TO 00027
00026 CONTINUE
      M=3HLP+
      GO TO 00028
00027 IF(L.EQ.LV)GO TO 00029
      GO TO 00030
00029 CONTINUE
      M=3HLV+
      GO TO 00028
00030 IF(L.EQ.LI)GO TO 00031
      GO TO 00032
00031 CONTINUE
      M=3HL1+
      GO TO 00028
00032 CONTINUE
      M=3H+
00028 CONTINUE
      IF(K.EQ.1)GO TO 00033
      GO TO 00034
00033 CONTINUE
      WRITE(N01,00036)M,N,K,L,IA(L)
00036 FORMAT(1X,A3,I2,2I5,I12)
      GO TO 00035
00034 IF(K.EQ.3)GO TO 00037
      GO TO 00038
00037 CONTINUE
      WRITE(N01,00039)M,N,K,L,RA(L)
00039 FORMAT(1X,A3,I2,2I5,E12.4)
      GO TO 00035
00038 IF(K.EQ.2)GO TO 00040
      GO TO 00041
00040 CONTINUE
      IF(IA(L).EQ.0)GO TO 00042
      GO TO 00043
00042 CONTINUE
      WRITE(N01,00045)M,N,K,L,IA(L)
00045 FORMAT(1X,A3,I2,2I5,I12)
      GO TO 00044
00043 IF(IA(L).GT.0)GO TO 00046

```

FORTRAN Control Programs

```

GO TO 00047
CONTINUE
00046 J=IA(IA(L)+3)
WRITE(N01,00048)M,N,K,L,IA(L),(IA(LCS(IA(L),2+1)),I=1,J)
00048 FORMAT(1X,A3,12,215,112/(28X,5112))
GO TO 00044
00047 CONTINUE
WRITE(N01,00049)
00049 FORMAT(/1X,15H ERROR /1X,
X 29HNEGATIVE LIST LOCATION (P:DF))
CALL ERROR
CONTINUE
00044 GO TO 00035
00041 IF(K.EQ.4)GO TO 00050
GO TO 00051
00050 CONTINUE
IF(IA(L).EQ.0)GO TO 00052
GO TO 00053
00052 CONTINUE
WRITE(N01,00055)M,N,K,L,IA(L)
00055 FORMAT(1X,A3,12,215,112)
GO TO 00054
00053 IF(IA(L).GT.0)GO TO 00056
GO TO 00057
00056 CONTINUE
J=IA(IA(L)+3)
WRITE(N01,00058)M,N,K,L,IA(L),(IA(LCS(IA(L),2+1)),I=1,J)
00058 FORMAT(1X,A3,12,215,112/(28X,5E12.4))
GO TO 00054
00057 CONTINUE
WRITE(N01,00059)
00059 FORMAT(/1X,15H ERROR /1X,
X 29HNEGATIVE LIST LOCATION (P:DF))
CALL ERROR
CONTINUE
GO TO 00063
00051 CONTINUE
WRITE(N01,00060)
00060 FORMAT(/1X,15H ERROR /1X,25HILLEGAL ARGUMENT K (P:DF))
CALL ERROR
00035 CONTINUE
RETURN
END
C POP(L1,N,M)
C SUBROUTINE POPS N INTEGERS OFF THE STACK AT L1 INTO THE
C ARRAY M.
SUBROUTINE POP(L1,N,M)
DIMENSION IA(1)
COMMON RA(1)
EQUIVALENCE (IA(1),RA(1))
COMMON /SYSTEM/MFROM,MTO,ICTYPE,IPPOINT,
11DUMP,ISECA,ITRACE,LB,LBCA,LB1ST,LBFT,LBUF,LCSB,LE,

```

FORTRAN Control Programs

```

1LEC,LFFB,LFFE,L1,LLE,LLFB,LLFE,LMS,LNE,LP,LV,MINNFL,
IMNSE,NFE,NFL,N11,N12,NO1,NO2,NREP,NRUN,NSE,TIME,TSTOP
DIMENSION M(1)
I=1
00026 I=I+1
IF((I)-(I-(N)) LE 0)GO TO 00027
GO TO 00028
00027 CONTINUE
MIN=1-I=[A(LCS(L1,IA(L1+3)+3-1))
GO TO 00026
00028 CONTINUE
IA(L1+3)=[A(L1+3)-N
RETURN
END
C PPLB(IPVT)
C SUBROUTINE PRINTS PARAMETERS OF FUNCTION AT LB IN DUMP FORMAT
SUBROUTINE PPLB(IPVT)
DIMENSION IPVT(1)
DIMENSION IA(1)
COMMON RA(1)
EQUIVALENCE (IA(1),RA(1))
COMMON /SYSTEM/MFROM,MT0,ICTYPE,IPOINT,
1DUMP,ISECA,ITRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
1LEC,LFFB,LFFE,L1,LLE,LLFB,LLFE,LMS,LNE,LP,LV,MINNFL,
IMNSE,NFE,NFL,N11,N12,NO1,NO2,NREP,NRUN,NSE,TIME,TSTOP
I=1
00026 IF((I)-(IA(LB+3))) LE 0)GO TO 00027
GO TO 00028
00027 CONTINUE
CALL PIDF(LP-1+1,IPVT(1),1-1)
GO TO 00026
00028 CONTINUE
RETURN
END
C PUSH(L1,N,M)
C SUBROUTINE PUSHES N INTEGERS IN ARRAY M INTO THE STACK AT L1.
SUBROUTINE PUSH(L1,N,M)
DIMENSION IA(1)
COMMON RA(1)
EQUIVALENCE (IA(1),RA(1))
COMMON /SYSTEM/MFROM,MT0,ICTYPE,IPOINT,
1DUMP,ISECA,ITRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
1LEC,LFFB,LFFE,L1,LLE,LLFB,LLFE,LMS,LNE,LP,LV,MINNFL,
IMNSE,NFE,NFL,N11,N12,NO1,NO2,NREP,NRUN,NSE,TIME,TSTOP
DIMENSION M(1)
IF (IA(L1+3)+N.GT. IA(L1+2))GO TO 00026
GO TO 00027
00026 GO TO 00027
00027 CONTINUE
00029 FORMAT(1X,15H ERROR /1X,17HSTACK FULL (PUSH))
CALL ERROR

```

```

00027 CONTINUE
      I=I+1
00030 IF(I) (I-(N)).LE.DIGO TO 00031
      GO TO 00032
00031 CONTINUE
      IA(LCS(L1,IA(L1+3)+2+1))=M(I)
      GO TO 00030
00032 CONTINUE
      IA(L1+3)=[A(L1+3)+N
      RETURN
      END
C PVLB(IPVT)
C SUBROUTINE PRINTS VARIABLES OF THE FUNCTION AT LB IN DUMP FORMAT.
      SUBROUTINE PVLB(IPVT)
      DIMENSION IPVT(1)
      DIMENSION IA(1)
      COMMON RA(1)
      EQUIVALENCE (IA(1),RA(1))
      COMMON /SYSTEM/MFROM,MTO,ICTYPE,IPONT,
      11DUMP,ISECA,ITRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
      1LEC,LFFB,LFFE,L1,LLE,LLFB,LLFE,LMS,LNE,LP,LV,MNNFL,
      1MNSE,NFE,NFL,N11,N12,N01,N02,NREP,NRUN,NSE,TIME,TSTOP
      I=I+1
00026 I=I+1
      IF(I) (I-(IA(LB+4))).LE.DIGO TO 00027
      GO TO 00028
00027 CONTINUE
      CALL PIDF(LV-1+1,IPVT(IA(LB+3)+1),I-1)
      GO TO 00026
00028 CONTINUE
      RETURN
      END
C RAC(L,I)
C FUNCTION RETURNS REAL VALUE OF ITEM I IN CHAIN STORAGE AT L.
      FUNCTION RAC(L,I)
      DIMENSION IA(1)
      COMMON RA(1)
      EQUIVALENCE (IA(1),RA(1))
      COMMON /SYSTEM/MFROM,MTO,ICTYPE,IPONT,
      11DUMP,ISECA,ITRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
      1LEC,LFFB,LFFE,L1,LLE,LLFB,LLFE,LMS,LNE,LP,LV,MNNFL,
      1MNSE,NFE,NFL,N11,N12,N01,N02,NREP,NRUN,NSE,TIME,TSTOP
      RAC=RA(LCS(L,I))
      RETURN
      END
C RIC(I)
C FUNCTION RETURNS THE ITH REAL INITIAL CONDITION OF THE FUNCTION AT LB.
      FUNCTION RIC(I)
      DIMENSION IA(1)
      COMMON RA(1)
      EQUIVALENCE (IA(1),RA(1))

```

FORTAN Control Programs

```

COMMON /SYSTEM/MFROM, MTO, ICTYPE, IPOINT,
11DUMP, ISECA, ITRACE, LB, LBGA, LBLIST, LBPT, LBUF, LCSB, LE,
1LEC, LFFB, LFFE, LI, LLE, LLFB, LLFE, LMS, LNE, LP, LV, MINNFI,
1MNSF, NFE, NFL, NI1, NI2, NO1, NO2, NREP, NRUN, NSF, TIME, TSTOP
RLC=RA(LI+1)
RETURN
END
C RLC(I, J)
C FUNCTION RETURNS THE JTH ELEMENT OF REAL LIST INITIAL CONDITION
C 1 OF THE FUNCTION AT LB
FUNCTION RLC(I, J)
DIMENSION IA(1)
COMMON RA(1)
EQUIVALENCE (IA(1), RA(1))
COMMON /SYSTEM/MFROM, MTO, ICTYPE, IPOINT,
11DUMP, ISECA, ITRACE, LB, LBGA, LBLIST, LBPT, LBUF, LCSB, LE,
1LEC, LFFB, LFFE, LI, LLE, LLFB, LLFE, LMS, LNE, LP, LV, MINNFI,
1MNSF, NFE, NFL, NI1, NI2, NO1, NO2, NREP, NRUN, NSF, TIME, TSTOP
RLC=RA(LCS(IA(LI+1), J+2))
RETURN
END
C RLP(I, J)
C FUNCTION RETURNS THE JTH ELEMENT OF THE REAL LIST PARAMETER 1 OF
C THE FUNCTION AT LB
FUNCTION RLP(I, J)
DIMENSION IA(1)
COMMON RA(1)
EQUIVALENCE (IA(1), RA(1))
COMMON /SYSTEM/MFROM, MTO, ICTYPE, IPOINT,
11DUMP, ISECA, ITRACE, LB, LBGA, LBLIST, LBPT, LBUF, LCSB, LE,
1LEC, LFFB, LFFE, LI, LLE, LLFB, LLFE, LMS, LNE, LP, LV, MINNFI,
1MNSF, NFE, NFL, NI1, NI2, NO1, NO2, NREP, NRUN, NSF, TIME, TSTOP
RLP=RA(LCS(IA(LP+1), J+2))
RETURN
END
C RLV(I, J)
C FUNCTION RETURNS THE JTH ELEMENT OF THE REAL LIST VARIABLE 1 OF THE
C FUNCTION AT LB
FUNCTION RLV(I, J)
DIMENSION IA(1)
COMMON RA(1)
EQUIVALENCE (IA(1), RA(1))
COMMON /SYSTEM/MFROM, MTO, ICTYPE, IPOINT,
11DUMP, ISECA, ITRACE, LB, LBGA, LBLIST, LBPT, LBUF, LCSB, LE,
1LEC, LFFB, LFFE, LI, LLE, LLFB, LLFE, LMS, LNE, LP, LV, MINNFI,
1MNSF, NFE, NFL, NI1, NI2, NO1, NO2, NREP, NRUN, NSF, TIME, TSTOP
RLV=RA(LCS(IA(LV+1), J+2))
RETURN
END
C RNORM(RMU, SIGMA)
C FUNCTION RETURNS REAL RANDOM VARIABLE WITH NORMAL DISTRIBUTION.
FUNCTION RNORM(RMU, SIGMA)

```

FORTRAN Control Programs

```

      DATA IX,U/1,0./
      X=0.
      I=1-(1)
00026 IF(I) (I-(12)).LE.0)GO TO 00027
      GO TO 00028
00027 CONTINUE
      CALL URN(IX,U)
      X=X+U
      GO TO 00026
00028 CONTINUE
      X=-6.*X
      RNORM=RMU+SIGMA X
      RETURN
      END
C RNORMT(RMU, SIGMA, RMAX, RMIN)
C FUNCTION RETURNS REAL RANDOM VARIABLE WITH TRUNCATED NORMAL
C DISTRIBUTION.
      FUNCTION RNORMT(RMU, SIGMA, RMAX, RMIN)
      DATA IX,U/1,0./
      X=0.
      I=1-(1)
      I=1+1
00026 IF(I) (I-(12)).LE.0)GO TO 00027
      GO TO 00028
00027 CONTINUE
      CALL URN(IX,U)
      X=X+U
      GO TO 00026
00028 CONTINUE
      X=-6.*X
      R=RMU+SIGMA X
      IF(R.LT.RMIN)R=RMIN
      IF(R.GT.RMAX)R=RMAX
      RNORMT=R
      RETURN
      END
C RP(I)
C FUNCTION RETURNS THE ITH REAL PARAMETER OF THE FUNCTION AT LB.
      FUNCTION RP(I)
      DIMENSION IA(1)
      COMMON RA(1)
      EQUIVALENCE (IA(1),RA(1))
      COMMON /SYSTEM/MFROM,MTQ,ICTYPE,IPCONT,
      IDUMP,ISECA,ITRACE, LB,LBCA,LBLIST,LBPT, _BUF,LCSB,LF,
      ILEC,LFFB,LFFE,L1,LE,LLFB,LLFE,LMS,LNE,LP,LV,WINHFL,
      MNSE,MFE,NFL,N11,N12,N01,N02,NREP,NRUN,NSE,TIME,TSTOP
      RP=RA(LP+I)
      RETURN
      END
C RUNIF(RM,SP)
C FUNCTION RETURNS REAL RANDOM VARIABLE WITH UNIFORM DISTRIBUTION.

```

FORTRAN Control Programs

```

FUNCTION RUNIF(RM,SP)
  DIMENSION IA(1)
  COMMON RA(1)
  EQUIVALENCE (IA(1),RA(1))
  COMMON /SYSTEM/MFROM, MTO, ICTYPE, IPOINT,
  IIDUMP, ISECA, ITRACE, LB, LBGA, LBLIST, LBPT, LBUF, LCSB, LE,
  ILEC, LFFB, LFFE, LI, LLE, LLFB, LLFE, LMS, LNE, LP, LV, MINNPL,
  IMNSE, NFE, NPL, NI1, NI2, NO1, NO2, NREP, NRUN, NSE, TIME, TSTOP
  DATA IX, U1, D, /
  CALL URN(IX,U)
  RUNIF=RM-SP+2. U SP
  RETURN
END
C RV(1)
C FUNCTION RETURNS THE ITH REAL VARIABLE OF THE FUNCTION AT LB.
  FUNCTION RV(I)
    DIMENSION IA(1)
    COMMON RA(1)
    EQUIVALENCE (IA(1),RA(1))
    COMMON /SYSTEM/MFROM, MTO, ICTYPE, IPOINT,
    IIDUMP, ISECA, ITRACE, LB, LBGA, LBLIST, LBPT, LBUF, LCSB, LE,
    ILEC, LFFB, LFFE, LI, LLE, LLFB, LLFE, LMS, LNE, LP, LV, MINNPL,
    IMNSE, NFE, NPL, NI1, NI2, NO1, NO2, NREP, NRUN, NSE, TIME, TSTOP
    RV=RA(LV+I)
    RETURN
  END
C SAMPL(I,J,V)
C SUBROUTINE RETURNS SAMPLE VALUE V FROM A POPULATION WITH
C DISTRIBUTION TYPE RAC(I,J+2) AND STATISTICS RAC(I,J+3), RAC(I,J+4),
C ETC.
  SUBROUTINE SAMPL(I,J,V)
    DIMENSION IA(1)
    COMMON RA(1)
    EQUIVALENCE (IA(1),RA(1))
    COMMON /SYSTEM/MFROM, MTO, ICTYPE, IPOINT,
    IIDUMP, ISECA, ITRACE, LB, LBGA, LBLIST, LBPT, LBUF, LCSB, LE,
    ILEC, LFFB, LFFE, LI, LLE, LLFB, LLFE, LMS, LNE, LP, LV, MINNPL,
    IMNSE, NFE, NPL, NI1, NI2, NO1, NO2, NREP, NRUN, NSE, TIME, TSTOP
    IF( ITRACE .GE. 5) GO TO 00026
    GO TO 00027
00026 CONTINUE
    WRITE(N01,00029)TIME,IA(LB+2)
00029 FORMAT(IX,E10.4,14,2X,12HSAMPL(1,J, ))
    WRITE(N01,00030)I,J
00030 FORMAT(IX,2I10)
00027 CONTINUE
    IF( ITRACE .GE. 6 .AND. ( ITRACE .EQ. ITRACE/2 ) ) GO TO 00031
    GO TO 00032
00031 CONTINUE
    WRITE(N02,00034)TIME,IA(LB+2)
00034 FORMAT(IX,E10.4,14,2X,12HSAMPL(1,J, ))
    WRITE(N02,00035)I,J

```


FORTRAN Control Programs

```

00035 FORMAT(1X,2110)
      CALL ICHECK
00032 CONTINUE
      IF(1/FIX(RAC(I,J+2)).EQ.1)GO TO 00036
      GO TO 00037
00036 CONTINUE
      V=RUNIF(RAC(I,J+3),RAC(I,J+4))
      GO TO 00038
00037 IF(1/FIX(RAC(I,J+2)).EQ.2)GO TO 00039
      GO TO 00040
00039 CONTINUE
      V=RNORM(RAC(I,J+3),RAC(I,J+4))
      GO TO 00038
00040 IF(1/FIX(RAC(I,J+2)).EQ.3)GO TO 00041
      GO TO 00042
00041 CONTINUE
      V=RNORMT(RAC(I,J+3),RAC(I,J+4),RAC(I,J+5),RAC(I,J+6))
      GO TO 00038
00042 IF(1/FIX(RAC(I,J+2)).EQ.4)GO TO 00043
      GO TO 00044
00043 CONTINUE
      CALL URN(1X,U)
      V=-RAC(I,J+3) ALOG(U)
      GO TO 00038
00044 CONTINUE
      WRITE(NC1,00045)
00045 FORMAT(1X,15H      ERROR      /1X,
      X      3111SAMPL ILLEGAL DISTRIBUTION TYPE)
      CALL ERROR
00036 CONTINUE
      IF(1/TRACE.GE.5)GO TO 00046
      GO TO 00047
00046 CONTINUE
      WRITE(NC1,00049)TIME,1A(LB+2)
00049 FORMAT(1X,E10.4,14,2X,12HSAMPL( ,V))
      WRITE(NC1,00050)V
00050 FORMAT(1X,E12.4)
00047 CONTINUE
      IF(1/TRACE.GE.6.AND.(1/TRACE.EQ.1/TRACE/2.2))GO TO 00051
      GO TO 00052
00051 CONTINUE
      WRITE(NC2,00054)TIME,1A(LB+2)
00054 FORMAT(1X,E10.4,14,2X,12HSAMPL( ,V))
      WRITE(NC2,00055)V
00055 FORMAT(1X,E12.4)
      CALL ICHECK
00052 CONTINUE
      RETURN
      END
C SCHED(T,NB,K,LD)
C SUBROUTINE SCHEDULES EVENT CODE K IN BLOCK NB FOR TIME T WITH EVENT
C DATA AT LOCATION LD (=0 IF NONE).

```

FORTRAN Control Programs

```

SUBROUTINE SCHED(T,NB,K,LD)
DIMENSION IA(1)
COMMON RA(1)
EQUIVALENCE (IA(1),RA(1))
COMMON /SYSTEM/MFROM,MT0,ICTYPE,IPOINT,
1DUMP,ISECA,ITRACE,LE,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
1LEC,LFFB,LFFE,L1,LLE,LFB,LFE,LMS,LNE,LP,LV,MINNEL,
1MNSE,NFE,N1,L,N11,N12,N01,N02,NREP,NRUN,NSÉ,TIME,TSTOP
IF(T.LT.TIME)GO TO 00026
GO TO 00027
00026 CONTINUE
WRITE(N01,00029)
00029 FORMAT(/1X,15H ERROR /1X
X 53HSCHED ATTEMPT TO SCHEDULE EVENT PRIOR TO PRESENT TIME)
CALL ERROR
00027 CONTINUE
IF(NFE.EQ.0)GO TO 00030
GO TO 00031
00030 CONTINUE
CALL INCREC(LEC)
00031 CONTINUE
RA(LFFE+1)=T
IA(LFFE+2)=NB
IA(LFFE+3)=K
IA(LFFE+4)=LD
L=LNE
L1=LFFE
LFFE=IA(LFFE)
00033 IF(RA(L+1).LE.T.AND.L.LNE.0)GO TO 00034
GO TO 00035
00034 CONTINUE
L2=L
L=IA(L)
GO TO 00033
00035 CONTINUE
IF(L.EQ.LNE)GO TO 00036
GO TO 00037
00036 CONTINUE
LNE=L1
IA(L1)=L
GO TO 00035
00037 CONTINUE
IF(L.NE.0)GO TO 00039
GO TO 00040
00039 CONTINUE
IA(L2)=L1
IA(L1)=L
GO TO 00041
00040 CONTINUE
IA(L2)=L1
IA(L1)=0
LLE=L1

```

```

00041 CONTINUE
00038 CONTINUE
      NFE=NFE+1
      NSE=NSC+1
      IF (NSE GT. MNSE/MNSE=NSE)
        RETURN
      END
C SEND
C SUBROUTINE TRANSMITS COMMUNICATIONS.
      SUBROUTINE SEND
      DIMENSION IA(1)
      COMMON RA(1)
      EQUIVALENCE ((IA(1),RA(1)))
      COMMON /SYSTEM/MFROM, MTO, ICTYPE, IPOINT,
11DUMP, ISECA, ITRACE, LB, LBGA, LBLIST, LBPT, LBUF, LCSB, LE,
      ILEC, LFFB, LFFE, LI, LLE, LLFB, LLFE, LMS, LME, LP, LV, MINIR,
      IMNSE, NFE, NFL, NI1, NI2, NO1, NO2, NREP, NRUN, NSE, TIME, TSTOP
      DIMENSION IC(4), IP(4)
      EQUIVALENCE (MFROM, IC(1))
      IERR=0
00026 IF (IERR EQ. 0) GO TO 00027
      GO TO 00028
00027 CONTINUE
      IF (ITRACE.GE.5) GO TO 00029
      GO TO 00030
00029 CONTINUE
      WRITE(N01,00032)TIME,IA(LB+2)
00032 FORMAT(1X,E10.4,14,2X,20HSEND MFROM MTO ICTYPE IPOINT)
      WRITE(N01,00033)MFROM,MTO,ICTYPE,IPOINT
00033 FORMAT(1X,416)
00030 CONTINUE
      IF (ITRACE.GE.6.AND. (ITRACE.EQ.ITRACE/2 2)) GO TO 00034
      GO TO 00035
00034 CONTINUE
      WRITE(N02,00037)TIME,IA(LB+2)
00037 FORMAT(1X,E10.4,14,2X,20HSEND MFROM MTO ICTYPE IPOINT)
      WRITE(N02,00038)MFROM,MTO,ICTYPE,IPOINT
00038 FORMAT(1X,416)
      CALL ICHECK
00035 CONTINUE
      IF (ICTYPE.LT.0) GO TO 00039
      GO TO 00040
00039 CONTINUE
      NADDR=MFROM
      CALL POP(LMS,4,IK)
      IF (IC(1).NE.IK(1).OR.IC(2).NE.IK(2).OR.IC(3).NE.IK(3)) GO TO
X      00042
      GO TO 00043
00042 CONTINUE
      WRITE(N01,00045)
00045 FORMAT(1X,15H      ERROR      /1X,22HSEND ILLEGAL RESPONSE)
      CALL ERROR

```

FORTRAN Control Programs

```

00043 CONTINUE
      IF(I(LMS+3).GT.0)GO TO 00046
      GO TO 00047
00046 CONTINUE      MFROM=IAC(LMS,IA(LMS+3)-1)
                   MTO=IAC(LMS,IA(LMS+3))
00047 CONTINUE
      GO TO 00041
00040 IF(CTYPE GT 0)GO TO 00049
      GO TO 00050
00049 CONTINUE
      NADDR=MTO
      CALL PUSH(LMS,4,IC)
      GO TO 00041
00050 CONTINUE
      WRITE(N01,00051)
00051 FORMAT(/1X,15H      ERROR      /1X,20HSEND  ILLEGAL ICTYPE)
      CALL ERROR
00041 CONTINUE
      IF(NADDR.GT.0)GO TO 00052
      GO TO 00053
00052 CONTINUE
      ASSIGN 00055 TO L00001
      GO TO 00001
00055 CONTINUE
      GO TO 00054
00053 CONTINUE
      RETURN
00054 CONTINUE
      GO TO 00026
00028 CONTINUE
      WRITE(N01,00056)
00056 FORMAT(/1X,15H      ERROR      /1X,24HSEND  ILLEGAL BLOCK TYPE)
      CALL ERROR
C
00001 CONTINUE
      IF(CTYPE.EQ.1)GO TO 00057
      GO TO 00056
00057 CONTINUE
      I=IP0INT
      GO TO 00059
00058 CONTINUE
      CALL FPNTRS(NADDR)
      I=IA(ILB)
00059 CONTINUE
      J=1/100
      K=I-J*100
      IF(J.EQ.1)GO TO 00060
      GO TO 00061
00060 CONTINUE
      IF(K.EQ.1)GO TO 00063
      GO TO 00064

```

FORTRAN Sorting Program

```

00061 CONTINUE
      CALL B101
      GO TO 00065
00064 IF (K.EQ.2)GO TO 00066
      GO TO 00067
00066 CONTINUE
      CALL B102
      GO TO 00065
00067 IF (K.EQ.3)GO TO 00068
      GO TO 00069
00068 CONTINUE
      CALL B103
      GO TO 00065
00069 IF (K.EQ.4)GO TO 00070
      GO TO 00071
00070 CONTINUE
      CALL B104
      GO TO 00065
00071 IF (K.EQ.5)GO TO 00072
      GO TO 00073
00072 CONTINUE
      CALL B105
      GO TO 00065
00073 IF (K.EQ.6)GO TO 00074
      GO TO 00075
00074 CONTINUE
      CALL B106
      GO TO 00065
00075 IF (K.EQ.7)GO TO 00076
      GO TO 00077
00076 CONTINUE
      CALL B107
      GO TO 00065
00077 IF (K.EQ.8)GO TO 00078
      GO TO 00079
00078 CONTINUE
      CALL B108
      GO TO 00065
00079 IF (K.EQ.9)GO TO 00080
      GO TO 00081
00080 CONTINUE
      CALL B109
      GO TO 00065
00081 IF (K.EQ.10)GO TO 00082
      GO TO 00083
00082 CONTINUE
      CALL B110
      GO TO 00065
00083 CONTINUE
      IERR=1
00065 CONTINUE
      GO TO 00062

```

```

00081 IF (K.EQ 2100) TO 00084
00082 GO TO 00085
00083 CONTINUE
00084 IF (K.EQ 2100) TO 00086
00085 GO TO 00087
00086 CONTINUE
00087 CALL B201
00088 GO TO 00089
00089 IF (K.EQ 2100) TO 00091
00090 GO TO 00092
00091 CONTINUE
00092 CALL B203
00093 GO TO 00088
00094 IF (K.EQ 2100) TO 00096
00095 GO TO 00094
00096 CONTINUE
00097 CALL B204
00098 GO TO 00088
00099 IF (K.EQ 5100) TO 00095
00100 GO TO 00096
00101 CONTINUE
00102 CALL B205
00103 GO TO 00088
00104 IF (K.EQ 6100) TO 00097
00105 GO TO 00098
00106 CONTINUE
00107 CALL B206
00108 GO TO 00088
00109 IF (K.EQ 7100) TO 00099
00110 GO TO 00100
00111 CONTINUE
00112 CALL B207
00113 GO TO 00088
00114 IF (K.EQ 8100) TO 00101
00115 GO TO 00102
00116 CONTINUE
00117 CALL B208
00118 GO TO 00088
00119 IF (K.EQ 9100) TO 00103
00120 GO TO 00104
00121 CONTINUE
00122 CALL B209
00123 GO TO 00088
00124 IF (K.EQ 10100) TO 00105
00125 GO TO 00106
00126 CONTINUE
00127 CALL B210

```

FORTRAN Control Programs

```

      GO TO 00088
00106 CONTINUE
      IERR=1
00088 CONTINUE
      GO TO 00062
00085 IF(J.EQ.3)GO TO 00107
      GO TO 00108
00107 CONTINUE
      IF(K.EQ.1)GO TO 00109
      GO TO 00110
00109 CONTINUE
      CALL B301
      GO TO 00111
00110 IF(K.EQ.2)GO TO 00112
      GO TO 00113
00112 CONTINUE
      CALL B302
      GO TO 00111
00113 IF(K.EQ.3)GO TO 00114
      GO TO 00115
00114 CONTINUE
      CALL B303
      GO TO 00111
00115 IF(K.EQ.4)GO TO 00116
      GO TO 00117
00116 CONTINUE
      CALL B304
      GO TO 00111
00117 IF(K.EQ.5)GO TO 00118
      GO TO 00119
00118 CONTINUE
      CALL B305
      GO TO 00111
00119 IF(K.EQ.6)GO TO 00120
      GO TO 00121
00120 CONTINUE
      CALL B306
      GO TO 00111
00121 IF(K.EQ.7)GO TO 00122
      GO TO 00123
00122 CONTINUE
      CALL B307
      GO TO 00111
00123 IF(K.EQ.8)GO TO 00124
      GO TO 00125
00124 CONTINUE
      CALL B308
      GO TO 00111
00125 IF(K.EQ.9)GO TO 00126
      GO TO 00127
00126 CONTINUE
      CALL B309

```

FORTRAN Control Programs

```

00127 GO TO 00111
00127 IF(K.EQ.1)GO TO 00128
00127 GO TO 00129
00128 CONTINUE
00128 CALL B310
00128 GO TO 00111
00129 CONTINUE
00129 IERR=1
00111 CONTINUE
00129 GO TO 00062
00108 IF(J.EQ.4)GO TO 00130
00108 GO TO 00131
00130 CONTINUE
00130 IF(K.EQ.1)GO TO 00132
00130 GO TO 00133
00132 CONTINUE
00132 CALL B401
00132 GO TO 00134
00133 IF(K.EQ.2)GO TO 00135
00133 GO TO 00136
00135 CONTINUE
00135 CALL B402
00135 GO TO 00134
00136 IF(K.EQ.3)GO TO 00137
00136 GO TO 00138
00137 CONTINUE
00137 CALL B403
00137 GO TO 00134
00138 IF(K.EQ.4)GO TO 00139
00138 GO TO 00140
00139 CONTINUE
00139 CALL B404
00139 GO TO 00134
00140 IF(K.EQ.5)GO TO 00141
00140 GO TO 00142
00141 CONTINUE
00141 CALL B405
00141 GO TO 00134
00142 IF(K.EQ.6)GO TO 00143
00142 GO TO 00144
00143 CONTINUE
00143 CALL B406
00143 GO TO 00134
00144 IF(K.EQ.7)GO TO 00145
00144 GO TO 00146
00145 CONTINUE
00145 CALL B407
00145 GO TO 00134
00146 IF(K.EQ.8)GO TO 00147
00146 GO TO 00148
00147 CONTINUE
00147 CALL B408

```


FORTRAN Control Programs

```

      GO TO 00134
00148 IF(K.EQ.9)GO TO 00149
      GO TO 00150
00149 CONTINUE
      CALL B409
      GO TO 00134
00150 IF(K.EQ.10)GO TO 00151
      GO TO 00152
00151 CONTINUE
      CALL B410
      GO TO 00134
00152 CONTINUE
      IERR=1
00134 CONTINUE
      GO TO 00062
00131 IF(J.EQ.5)GO TO 00153
      GO TO 00154
00153 CONTINUE
      IF(K.EQ.1)GO TO 00155
      GO TO 00156
00155 CONTINUE
      CALL B501
      GO TO 00157
00156 IF(K.EQ.2)GO TO 00158
      GO TO 00159
00158 CONTINUE
      CALL B502
      GO TO 00157
00159 IF(K.EQ.3)GO TO 00160
      GO TO 00161
00160 CONTINUE
      CALL B503
      GO TO 00157
00161 IF(K.EQ.4)GO TO 00162
      GO TO 00163
00162 CONTINUE
      CALL B504
      GO TO 00157
00163 IF(K.EQ.5)GO TO 00164
      GO TO 00165
00164 CONTINUE
      CALL B505
      GO TO 00157
00165 IF(K.EQ.6)GO TO 00166
      GO TO 00167
00166 CONTINUE
      CALL B506
      GO TO 00157
00167 IF(K.EQ.7)GO TO 00168
      GO TO 00169
00168 CONTINUE
      CALL B507

```

FORTRAN Control Programs

```

GO TO 00157
00169 IF(K.EQ.8)GO TO 00170
GO TO 00171
00170 CONTINUE
CALL B508
GO TO 00157
00171 IF(K.EQ.9)GO TO 00172
GO TO 00173
00172 CONTINUE
CALL B509
GO TO 00157
00173 IF(K.EQ.10)GO TO 00174
GO TO 00175
00174 CONTINUE
CALL B510
GO TO 00157
00175 CONTINUE
IERR=1
00157 CONTINUE
GO TO 00062
00154 IF(J.EQ.6)GO TO 00176
GO TO 00177
00176 CONTINUE
IF(K.EQ.1)GO TO 00178
GO TO 00179
00178 CONTINUE
CALL B501
GO TO 00180
00179 IF(K.EQ.2)GO TO 00181
GO TO 00182
00181 CONTINUE
CALL B502
GO TO 00180
00182 IF(K.EQ.3)GO TO 00183
GO TO 00184
00183 CONTINUE
CALL B503
GO TO 00180
00184 IF(K.EQ.4)GO TO 00185
GO TO 00186
00185 CONTINUE
CALL B504
GO TO 00180
00186 IF(K.EQ.5)GO TO 00187
GO TO 00188
00187 CONTINUE
CALL B505
GO TO 00180
00188 IF(K.EQ.6)GO TO 00189
GO TO 00190
00189 CONTINUE
CALL B506

```

FORTRAN Control Programs

```

GO TO 00180
00190 IF(K.EQ.7)GO TO 00191
GO TO 00192
00191 CONTINUE
CALL B607
GO TO 00180
00192 IF(K.EQ.8)GO TO 00193
GO TO 00194
00193 CONTINUE
CALL B608
GO TO 00180
00194 IF(K.EQ.9)GO TO 00195
GO TO 00196
00195 CONTINUE
CALL B6C9
GO TO 00180
00196 IF(K.EQ.10)GO TO 00197
GO TO 00198
00197 CONTINUE
CALL B610
GO TO 00180
00198 CONTINUE
ERR=1
00180 CONTINUE
GO TO 00062
00177 IF(J.EQ.7)GO TO 00199
GO TO 00200
00199 CONTINUE
IF(K.EQ.1)GO TO 00201
GO TO 00202
00201 CONTINUE
CALL B701
GO TO 00203
00202 IF(K.EQ.2)GO TO 00204
GO TO 00205
00204 CONTINUE
CALL B702
GO TO 00203
00205 IF(K.EQ.3)GO TO 00206
GO TO 00207
00206 CONTINUE
CALL B703
GO TO 00203
00207 IF(K.EQ.4)GO TO 00208
GO TO 00209
00208 CONTINUE
CALL B704
GO TO 00203
00209 IF(K.EQ.5)GO TO 00210
GO TO 00211
00210 CONTINUE
CALL B705

```

FORTRAN Control Programs

```

00211 GO TO 00203
      IF(K.EQ.6)GO TO 00212
      GO TO 00213
00212 CONTINUE
      CALL B706
      GO TO 00203
00213 IF(K.EQ.7)GO TO 00214
      GO TO 00215
00214 CONTINUE
      CALL B707
      GO TO 00203
00215 IF(K.EQ.8)GO TO 00216
      GO TO 00217
00216 CONTINUE
      CALL B708
      GO TO 00203
00217 IF(K.EQ.9)GO TO 00218
      GO TO 00219
00218 CONTINUE
      CALL B709
      GO TO 00203
00219 IF(K.EQ.10)GO TO 00220
      GO TO 00221
00220 CONTINUE
      CALL B710
      GO TO 00203
00221 CONTINUE
      IERR=1
00203 CONTINUE
      GO TO 00062
00200 IF(J.EQ.8)GO TO 00222
      GO TO 00223
00222 CONTINUE
      IF(K.EQ.1)GO TO 00224
      GO TO 00225
00224 CONTINUE
      CALL B801
      GO TO 00226
00225 IF(K.EQ.2)GO TO 00227
      GO TO 00228
00227 CONTINUE
      CALL B802
      GO TO 00226
00228 IF(K.EQ.3)GO TO 00229
      GO TO 00230
00229 CONTINUE
      CALL B803
      GO TO 00226
00230 IF(K.EQ.4)GO TO 00231
      GO TO 00232
00231 CONTINUE
      CALL B804

```

FORTRAN Control Programs

```

      GO TO 00226
00232 IF(K.EQ.5)GO TO 00233
      GO TO 00234
00233 CONTINUE
      CALL B805
      GO TO 00226
00234 IF(K.EQ.6)GO TO 00235
      GO TO 00236
00235 CONTINUE
      CALL B806
      GO TO 00226
00236 IF(K.EQ.7)GO TO 00237
      GO TO 00238
00237 CONTINUE
      CALL B807
      GO TO 00226
00238 IF(K.EQ.8)GO TO 00239
      GO TO 00240
00239 CONTINUE
      CALL B808
      GO TO 00226
00240 IF(K.EQ.9)GO TO 00241
      GO TO 00242
00241 CONTINUE
      CALL B809
      GO TO 00226
00242 IF(K.EQ.10)GO TO 00243
      GO TO 00244
00243 CONTINUE
      CALL B810
      GO TO 00226
00244 CONTINUE
      IERR=1
00226 CONTINUE
      GO TO 00062
00223 IF(J.EQ.9)GO TO 00245
      GO TO 00246
00245 CONTINUE
      IF(K.EQ.1)GO TO 00247
      GO TO 00248
00247 CONTINUE
      CALL B901
      GO TO 00249
00248 IF(K.EQ.2)GO TO 00250
      GO TO 00251
00250 CONTINUE
      CALL B902
      GO TO 00249
00251 IF(K.EQ.3)GO TO 00252
      GO TO 00253
00252 CONTINUE
      CALL B903

```

FORTRAN Control Programs

```

00253 GO TO 00249
      IF(K.EQ.4)GO TO 00254
00254 CONTINUE
      CALL B904
      GO TO 00249
00255 IF(K.EQ.5)GO TO 00256
      GO TO 00257
00256 CONTINUE
      CALL B905
      GO TO 00249
00257 IF(K.EQ.6)GO TO 00258
      GO TO 00259
00258 CONTINUE
      CALL B906
      GO TO 00249
00259 IF(K.EQ.7)GO TO 00260
      GO TO 00261
00260 CONTINUE
      CALL B907
      GO TO 00249
00261 IF(K.EQ.8)GO TO 00262
      GO TO 00263
00262 CONTINUE
      CALL B908
      GO TO 00249
00263 IF(K.EQ.9)GO TO 00264
      GO TO 00265
00264 CONTINUE
      CALL B909
      GO TO 00249
00265 IF(K.EQ.10)GO TO 00266
      GO TO 00267
00266 CONTINUE
      CALL B910
      GO TO 00249
00267 CONTINUE
      IERR=1
00249 CONTINUE
      GO TO 00062
00246 CONTINUE
      IERR=1
00062 CONTINUE
      GO TO L00001, (C0055)
      END
C SET(KODE,1PVT,INTD,REALD)
C SUBROUTINE SETS PARAMETERS (KODE =1) AND VARIABLE INITIAL CONDITIONS
C (KODE = 2) READ FROM DEVICE N11 IN THE FORM
C (I N M) VALUES (I N M) VALUES
C WHERE I IS THE PARAMETER OR VARIABLE INDEX AND N AND M ARE THE FIRST
C AND LAST ELEMENT LIST INDEXES RESPECTIVELY. VALUES ARE CONVERTED
C TO THE TYPE SPECIFICATION OF PARAMETER/VARIABLE TYPE CODES IN ARRAY

```

FORTRAN Control Programs

```

C IPVT: 1 = INTEGER 2 = INTEGER LIST 3 = REAL 4 = REAL LIST.
C ITEMS ARE BLANK DELIMITED AND MAY BE INTERSPERSED WITH COMMENTS.
SUBROUTINE SET(KODE,IPVT,INTD,REALD)
  DIMENSION IA(1)
  COMMON RA(1)
  EQUIVALENCE (IA(1),RA(1))
  COMMON /SYSTEM/MFROM,MTD,ICTYPE,IPOINT,
  1 DUMP,ISCA,ITRACE,LB,LBGA,LBLIST,LBPT,LBUF,LCSB,LE,
  1 LEC,LFB,LFE,L1,LLE,LLF8,LLFE,LMS,LNE,LP,LV,MINNFI,
  1 MNSE,NFE,NFL,N11,N12,N01,N02,NREP,NRUN,NSF,TIME,TSTOP
  DIMENSION IPVT(1)
  L1=LBUF+2
  LB0=LBUF+81
  IF(KODE.EQ.1)GO TO 00026
  GO TO 00027
00026 CONTINUE
  LPV=1
  LIARA=LP
  IMA=IA(LB+3)
  GO TO 00028
00027 IF(KODE.EQ.2)GO TO 00029
  GO TO 00030
00029 CONTINUE
  LPV=IA(LB+3)+1
  LIARA=L1
  IMA=IA(LB+4)
  GO TO 00028
00030 CONTINUE
  WRITE(N01,00031)
00031 FORMAT(1X,15H ERROR /1X,25HSET ILLEGAL ARGUMENT KODE)
  CALL ERROR
00028 CONTINUE
  LSCAN=L1
  ASSIGN 00032 TO L00001
  GO TO 00001
00032 CONTINUE
00033 IF(I.GT.0)GO TO 00034
  GO TO 00035
00034 CONTINUE
  IF(N.EQ.0)GO TO 00036
  GO TO 00037
00036 CONTINUE
  ASSIGN 00039 TO L00002
  GO TO 00002
00039 CONTINUE
  ASSIGN 00040 TO L00003
  GO TO 00003
00040 CONTINUE
  GO TO 00038
00037 CONTINUE
  J=N-11
00041 J=J+1

```

FORTRAN Control Programs

```

      IF(I) (J-(M)).LE.0)GO TO 00042
      GO TO 00043
00042 CONTINUE
      ASSIGN 00044 TO L00002
      GO TO 00002
00044 CONTINUE
      ASSIGN 00045 TO L00004
      GO TO 00004
00045 CONTINUE
      GO TO 00041
00043 CONTINUE
00038 CONTINUE
      ASSIGN 00046 TO L00001
      GO TO 00001
00046 CONTINUE
      GO TO 00033
00035 CONTINUE
      RETURN
C
00001 CONTINUE
      IF(LSCAN.EQ.L1)GO TO 00047
      GO TO 00046
00047 CONTINUE
      ASSIGN 00050 TO L00005
      GO TO 00005
00050 CONTINUE
      IF(KEYWD(L1).GT.0)GO TO 00051
      GO TO 00052
00051 CONTINUE
      BACKSPACE N11
      I=0
      GO TO 00053
00052 CONTINUE
      ASSIGN 00054 TO L00006
      GO TO 00006
00054 CONTINUE
00053 CONTINUE
      GO TO 00049
00048 CONTINUE
      ASSIGN 00055 TO L00006
      GO TO 00006
00055 CONTINUE
00049 CONTINUE
      GO TO L00001, (00032,00046)
C
00002 CONTINUE
      ASSIGN 00056 TO L00008
      GO TO 00008
00056 CONTINUE
      LPRINT=0
      REAL=0.
      NCH=1CH(1A(LSCAN))

```


FORTRAN Control Programs

```

      IF(NCH.GE.0.AND.NCH.LE.9)GO TO 00057
      GO TO 00054
00057 CONTINUE
      SIGN=1.
      GO TO 00059
00058 IF(NCH.EQ.38)GO TO 00060
      GO TO 00061
00060 CONTINUE
      LSCAN=LSCAN+1
      NCH=ICH(IA(LSCAN))
      IF(NCH.GE.0.AND.NCH.LE.9)GO TO 00062
      GO TO 00063
00062 CONTINUE
      SIGN=1.
      GO TO 00064
00063 IF(NCH.EQ.42)GO TO 00065
      GO TO 00066
00065 CONTINUE
      LSCAN=LSCAN+1
      NCH=ICH(IA(LSCAN))
      IF(NCH.LT.0.OR.NCH.GT.9)GO TO 00067
      GO TO 00068
00067 CONTINUE
      WRITE(N01,00070)
00070 FORMAT(/1X,15H      ERROR      /1X,
X      34HSET ILLEGAL CHARACTER FOLLOWING +.)
      CALL ERROR
00068 CONTINUE
      SIGN=1.
      GO TO 00064
00066 CONTINUE
      WRITE(N01,00071)
00071 FORMAT(/1X,15H      ERROR      /1X,
X      33HSET ILLEGAL CHARACTER FOLLOWING +)
      CALL ERROR
00064 CONTINUE
      GO TO 00059
00061 IF(NCH.EQ.39)GO TO 00072
      GO TO 00073
00072 CONTINUE
      LSCAN=LSCAN+1
      NCH=ICH(IA(LSCAN))
      IF(NCH.GE.0.AND.NCH.LE.9)GO TO 00074
      GO TO 00075
00074 CONTINUE
      SIGN=-1.
      GO TO 00076
00075 IF(NCH.EQ.42)GO TO 00077
      GO TO 00078
00077 CONTINUE
      LSCAN=LSCAN+1
      NCH=ICH(IA(LSCAN))

```

FORTRAN Control Programs

```

      IF(NCH.LT.0.OR.NCH.GT.9)GO TO 00079
      GO TO 00080
00079 CONTINUE
      WRITE(N01,00082)
00082 FORMAT(/1X,15H      ERROR      /1X,
X      34HSET ILLEGAL CHARACTER FOLLOWING -. )
00080 CONTINUE
      SIGN=-1.
      GO TO 00076
00076 CONTINUE
      WRITE(N01,00083)
00083 FORMAT(/1X,15H      ERROR      /1X,
X      32HSET ILLEGAL CHARACTER FOLLOWING -)
00076 CONTINUE
      GO TO 00059
00073 IF(NCH.EQ.42)GO TO 00084
      GO TO 00085
00084 CONTINUE
      LPOINT=LSCAN
      LSCAN=LSCAN+1
      NCH=ICHAR(A(LSCAN))
      IF(NCH.LT.0.OR.NCH.GT.9)GO TO 00086
      GO TO 00087
00086 CONTINUE
      WRITE(N01,00089)
00089 FORMAT(/1X,15H      ERROR      /1X,
X      32HSET ILLEGAL CHARACTER FOLLOWING .)
00087 CONTINUE
      SIGN=1.
      GO TO 00059
00085 CONTINUE
      WRITE(N01,00090)
00090 FORMAT(/1X,15H      ERROR      /1X,
X      44HSET DIGIT, +, -, OR . EXPECTED BUT NOT FOUND)
      CALL ERROR
00059 CONTINUE
00091 IF(NCH.LE.9.OR.NCH.EQ.42)GO TO 00092
      GO TO 00093
00092 CONTINUE
      IF(NCH.LE.9)GO TO 00094
      GO TO 00095
00094 CONTINUE
      REAL=+0. REAL+NCH
      GO TO 00096
00095 IF(NCH.EQ.42)GO TO 00097
      GO TO 00098
00097 CONTINUE
      LPOINT=LSCAN
      GO TO 00096

```

FORTRAN Control Programs

```

00098 CONTINUE
      WRITE(N01,00099)
00099 FORMAT(/1X,15H      ERROR      /1X,26HSET CANNOT INTERPRET VALUE)
      CALL ERROR
00096 CONTINUE
      LSCAN=LSCAN+1
      IF(LSCAN.GT.L80)GO TO 00100
      GO TO 00101
00100 CONTINUE
      WRITE(N01,00103)
00103 FORMAT(/1X,15H      ERROR      /1X,
X      34HSET END OF VALUE NOT FOUND ON LINE)
      CALL ERROR
00101 CONTINUE
      NCH=1CH(1A(LSCAN))
      GO TO 00097
00093 CONTINUE
      IF(LP0INT.EQ.0)GO TO 00104
      GO TO 00105
00104 CONTINUE
      REAL=SIGN REAL
      GO TO 00106
00105 CONTINUE
      REAL=SIGN REAL/(10 (LSCAN-LP0:NT-1))
00106 CONTINUE
      INT=REAL
      GO TO L00002, (00039,00044)
C
00003 CCNTINUE
      IF(1.LT.1.OR.1.GT.1MAX)GO TO 00107
      GO TO 00108
00107 CONTINUE
      WRITE(N01,00110)
00110 FORMAT(/1X,15H      ERROR      /1X,17HSET ILLEGAL INDEX)
      CALL ERROR
00108 CONTINUE
      IF(1PVT(LP0+1).EQ.1)GO TO 00111
      GO TO 00112
00111 CONTINUE
      1A(LIARA+1)=INT
      GO TO 00113
00112 IF(1PVT(LP0+1).EQ.3)GO TO 00114
      GO TO 00115
00114 CONTINUE
      RA(LIARA+1)=REAL
      GO TO 00113
00115 CONTINUE
      WRITE(N01,00116)
00116 FORMAT(/1X,15H      ERROR      /1X,16HSET ILLEGAL TYPE)
      CALL ERROR
00113 CONTINUE
      GO TO L00003, (00040)

```

FORTRAN Control Programs

```

C
00004 CONTINUE
      IF(I.LT.1.OR.I.GT.IMAX)GO TO 00117
      GO TO 00118
00117 CONTINUE
      WRITE(NO1,00120)
00120 FORMAT(/1X,15H      ERROR      /1X,17HSET (LEGAL INDEX)
      CALL ERROR
00118 CONTINUE
      IF(IA(LIARA+1).EQ.0)GO TO 00121
      GO TO 00122
00121 CONTINUE
      IF(N.NE.1)GO TO 00124
      GO TO 00125
00124 CONTINUE
      WRITE(NO1,00127)
00127 FORMAT(/1X,15H      ERROR      /1X,
X      39HSET N MUST BE 1 WHEN LIST FIRST DEFINED)
      CALL ERROR
00125 CONTINUE
      IF(M.GT.100)GO TO 00128
      GO TO 00129
00128 CONTINUE
      WRITE(59,00131)
00131 FORMAT(/1X,17H      WARNING      /1X,
X      39HSET PARAMETER LIST EXCEEDS 100 ELEMENTS)
00129 CONTINUE
      CALL ASTORE(IA(LIARA+1),M-N+3)
      IACTA(LIARA+1)+3=N-N+1
00122 CONTINUE
      IF(J.GT.1A(IA(LIARA+1)+3))GO TO 00132
      GO TO 00133
00132 CONTINUE
      WRITE(NO1,00135)
00135 FORMAT(/1X,15H      ERROR      /1X,
X      32HSET INDEX EXCEEDS LENGTH OF LIST)
      CALL ERROR
00133 CONTINUE
      IF(IPVT(LPV+1).EQ.2)GO TO 00136
      GO TO 00137
00136 CONTINUE
      IA(LCS(IA(LIARA+1),2+J))=INT
      GO TO 00138
00137 IF(IPVT(LPV+1).EQ.4)GO TO 00139
      GO TO 00140
00139 CONTINUE
      RA(LCS(IA(LIARA+1),2+J))=REAL
      GO TO 00138
00140 CONTINUE
      WRITE(NO1,00141)
00141 FORMAT(/1X,15H      ERROR      /1X,16HSET (LEGAL TYPE)
      CALL ERROR

```

FORTRAN Control Programs

```

00138 CONTINUE
      GO TO L00004, (00045)
C
00005 CONTINUE
      IEOF=0
      READ(N11,00142)((IA(K),K=L1,L80)
      IF((FIX(EOF(N11)))NE.0)GO TO 99999
00142 FORMAT(80A1)
      IF(KEYWD(L1).EQ.0)GO TO 00143
      GO TO 00144
00143 CONTINUE
      WRITE(N01,00146)((IA(K),K=L1,L80)
00146 FORMAT(1X,80A1)
      IF(IPOINT.EQ.3)GO TO 00147
      GO TO 00148
00147 CONTINUE
      WRITE(N02,00150)((IA(K),K=L1,L80)
00150 FORMAT(1X,80A1)
00148 CONTINUE
00144 CONTINUE
      GO TO 00151
99999 IEOF=1
00151 CONTINUE
      IF(IEOF.EQ.1)GO TO 00152
      GO TO 00153
00152 CONTINUE
      WRITE(N01,00155)
00155 FORMAT(/1X,15H      ERROR      /1X,21HSET END OF INPUT FILE)
      CALL ERROR
00153 CONTINUE
      GO TO L00005, (00050)
C
00006 CONTINUE
00156 IF(IA(LSCAN).NE.1H)GO TO 00157
      GO TO 00158
00157 CONTINUE
      LSCAN=LSCAN+1
      IF(LSCAN.GT.L80)GO TO 00159
      GO TO 00160
00159 CONTINUE
      ASSIGN 00162 TO L00005
      GO TO 00005
00162 CONTINUE
      IF(KEYWD(L1).GT.0)GO TO 00163
      GO TO 00164
00163 CONTINUE
      BACKSPACE N11
      I=0
      N=0
      M=0
      GO TO L00006, (00054,00055)
      GO TO 00165

```

FORTRAN Control Programs

```

00164 CONTINUE
      LSCAN=L1
00165 CONTINUE
00166 CONTINUE
      GO TO 00156
00158 CONTINUE
      LSCAN=LSCAN+1
      ASSIGN 00166 TO L00002
      GO TO 00002
00166 CONTINUE
      IF (INT.GT.0.AND.INT.LE.IMAX)GO TO 00167
      GO TO 00168
00167 CONTINUE
      I=INT
      ASSIGN 00170 TO L00007
      GO TO 00007
00170 CONTINUE
      IF (IA(LSCAN).EQ.1H).AND.(IPVT(LPV+1).EQ.1.OR.IPVT(LPV+1).EQ.2))GO
      X      TO 00171
00171 CONTINUE
      GO TO 00172
      N=0
      M=0
      GO TO 00173
00172 IF (ICH(IA(LSCAN)).LE.9.AND.(IPVT(LPV+1).EQ.2.OR.IPVT(LPV+1)
      X      .EQ. 4))GO TO 00174
      GO TO 00175
00174 CONTINUE
      ASSIGN 00176 TO L00002
      GO TO 00002
00176 CONTINUE
      N=INT
      ASSIGN 00177 TO L00007
      GO TO 00007
00177 CONTINUE
      NCH=ICH(IA(LSCAN))
      IF (NCH.GE.0.AND.NCH.LE.9)GO TO 00178
      GO TO 00179
00178 CONTINUE
      ASSIGN 00181 TO L00002
      GO TO 00002
00181 CONTINUE
      M=INT
      GO TO 00180
00179 CONTINUE
      WRITE(N01,00182)
00182 FORMAT(/1X,15H      ERROR      /1X,24HSET ILLEGAL LIST INDEX M)
      CALL ERROR
00180 CONTINUE
      GO TO 00173
00175 CONTINUE
      WRITE(N01,00183)

```

FORTRAN Control Programs

```

00183 FORMAT(/1X,15H      ERROR      /1X,24HSET (LLEGAL LIST INDEX I)
      CALL ERROR
00173 CONTINUE
      GO TO 00169
00168 CONTINUE
      WRITE(NO1,00184)
00184 FORMAT(/1X,15H      ERROR      /1X,24HSET (LLEGAL LIST INDEX I)
      CALL ERROR
00169 CONTINUE
      IF(IA(LSCAN).EQ.1H)GO TO 00185
      GO TO 00186
00185 CONTINUE
      LSCAN=LSCAN+1
      GO TO 00187
00186 CONTINUE
      WRITE(NO1,00188)
00188 FORMAT(/1X,15H      ERROR      /1X,
X      40HSET RIGHT BRACKET EXPECTED BUT NOT FOUND)
00187 CONTINUE
      GO TO L00006, (00054,00055)
C
00007 CONTINUE
00189 IF(IA(LSCAN).EQ.1H)GO TO 00190
      GO TO 00191
00190 CONTINUE
      ASSIGN 00192 TO L00010
      GO TO 00010
00192 CONTINUE
      GO TO 00189
00191 CONTINUE
      GO TO L00007, (00170,00177)
C
00008 CONTINUE
      ASSIGN 00193 TO L00007
      GO TO 00007
00193 CONTINUE
      NCH=ICH(IA(LSCAN))
00194 IF(NCH.LT.0.OR.(NCH.GT.9.AND.NCH.NE.38.AND.NCH.NE.39.AND.NCH.NE.
X      42))GO TO 00195
      GO TO 00196
00195 CONTINUE
      ASSIGN 00197 TO L00009
      GO TO 00009
00197 CONTINUE
      ASSIGN 00198 TO L00007
      GO TO 00007
00198 CONTINUE
      NCH=ICH(IA(LSCAN))
      GO TO 00194
00196 CONTINUE
      IF(NCH.LT.0)GO TO 00199

```

FORTRAN Control Programs

```

GO TO 00200
00199 CONTINUE
WRITE(N01,00202)
00202 FORMAT(/1X,15H      ERROR      /1X,21HSET ILLEGAL CHARACTER)
CALL ERROR
00200 CONTINUE
GO TO L00008, (00058)
C
00009 CONTINUE
00203 IF (IA(LSCAN).NE.1H )GO TO 00204
GO TO 00205
00204 CONTINUE
ASSIGN 00206 TO L00010
GO TO 00010
00206 CONTINUE
GO TO 00203
00205 CONTINUE
GO TO L00009, (00197)
C
00010 CONTINUE
LSCAN=LSCAN+1
IF (LSCAN.GT.L80)GO TO 00207
GO TO 00208
00207 CONTINUE
ASSIGN 00210 TO L00005
GO TO 00005
00210 CONTINUE
IF (KEYWD(L1).GT.0)GO TO 00211
GO TO 00212
00211 CONTINUE
BACKSPACE N11
WRITE(N01,00214)
00214 FORMAT(/1X,15H      ERROR      /1X,22HSET UNEXPECTED KEYWORD)
CALL ERROR
GO TO 00213
00212 CONTINUE
LSCAN=L1
00213 CONTINUE
00208 CONTINUE
GO TO L00010, (00192,00206)
C GETV(KODE,IPVT,INTD,REALD)
C ENTRY RETURNS INTEGER (REAL) VALUE OF THE NEXT NUMBER IN THE
C INPUT DATA.
ENTRY GETV
L1=LBUF+2
L80=LBUF+81
ASSIGN 00215 TO L00002
GO TO 00002
00215 CONTINUE
INTD=INT
REALD=REAL
RETURN

```


FORTRAN Control Programs

```

C FBLANK(KODE,IPVT,INTD,REALD)
C ENTRY INCREMENTS SCAN POINTER LSCAN TO FIRST BLANK.
  ENTRY FBLANK
    L1=LBUF+2
    L80=LBUF+81
    LSCAN=L1
00216 IF(I(A(LSCAN),NE.1H)GO TO 00217
    GO TO 00218
00217 CONTINUE
    LSCAN=LSCAN+1
    IF(LSCAN.GT.L80)GO TO 00219
    GO TO 00220
00219 CONTINUE
    WRITE(N01,00222)
00222 FORMAT(/1X,15H  ERROR /1X,
X 30HSET NO BLANK DELIMITER ON LINE)
    CALL ERROR
00220 CONTINUE
    GO TO 00216
00218 CONTINUE
    RETURN
  END
C TIMER(K,I,T)
C SUBROUTINE SUPPLIES TIMING INFORMATION BASED UPON THE INTERACTION
C RESPONSE OF A 601 TIMER FUNCTION. I IS SET TO 1 AND T TO TIME IF UP.
C I IS SET TO 2 IF DOWN TILL THE NEXT SHIFT AT TIME 1 AND TO 3
C IF DOWN TILL THE NEXT INTERVAL AT TIME T.
  SUBROUTINE TIMER(K,I,T)
    DIMENSION I(A(1))
    COMMON RA(1)
    EQUIVALENCE (I(A(1)),RA(1))
    COMMON /SYSTEM/MFROM,MTD,ICTYPE,IPPOINT,
11DUMP,ISECA,ITRACE,LB,LBCA,LBLS1T,LBPT,LBUF,LCSB,LE,
11LEC,LFFB,LFFE,L1,LLE,LLFB,LLFE,LMS,LNE,LP,LY,MINNFI,
11MNSF,NFE,NFL,N11,N12,N01,N02,NREP,NRUN,NSF,TIME,TSTOP
    IF(K.EQ.2)GO TO 00026
    GO TO 00027
00026 CONTINUE
    IF(I(A(IPPOINT+1).EQ.0)GO TO 00029
    GO TO 00030
00029 CONTINUE
    I=2
    T=RA(IPPOINT+3)
    GO TO 00031
00030 CONTINUE
    I=1
    T=TIME
00031 CONTINUE
    GO TO 00028
00027 IF(K.EQ.3)GO TO 00032
    GO TO 00033
00032 CONTINUE

```

FORTRAN Control Programs

```

      IF(IA(IPPOINT+4).EQ.0)GO TO 00034
      GO TO 00035
00034 CONTINUE
      I=3
      T=RA(IPPOINT+6)
00035 CONTINUE
      I=1
      T=TIME
00036 CONTINUE
      GO TO 00028
00033 CONTINUE
      WRITE(N01,00037)
00037 FORMAT(/1X,15H          ERROR          /1X,26HTIMER  ILLEGAL TIMING CODE)
      CALL ERROR
00028 CONTINUE
      ICTYPE=IAC(LMS,IA(LMS+3)+1)
      IPPOINT=IAC(LMS,IA(LMS+3)+2)
      RETURN
      END
C TRNFRM(NC,FB,L1,L2)
C SUBROUTINE TRANSFERS ALL (IF NC < 0) OR THE FIRST NC (IF NC > 0)
C CONTAINERS AND FRACTION FB OF THE BULK MATERIALS AT L1 TO L2.
      SUBROUTINE TRNFRM(NC,FB,L1,L2)
      DIMENSION IA(1)
      COMMON RA(1)
      EQUIVALENCE (IA(1),RA(1))
      COMMON /SYSTEM/MFROM,MT0,ICTYPE,IPPOINT,
      IDUMP,ISECA,ITRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
      LLEC,LFFB,LFFE,L1,LLE,LLFB,LLFE,LMS,LNE,LP,LV,MINNFI,
      IMNSE,NFE,NFL,N11,N12,N01,N02,NREP,NR0W,NSE,TIME,TSTOP
      IF(ITRACE.GE.5)GO TO 00026
      GO TO 00027
00026 CONTINUE
      WRITE(N01,00029)TIME,IA(LB+2)
00029 FORMAT(1X,E10.4,I4,2X,19HTRNFRM(NC,FB,L1,L2))
      WRITE(N01,00030)NC,FB,L1,L2
00030 FORMAT(1X,I6,E12.4,2I6)
00027 CONTINUE
      IF(ITRACE.GE.6.AND.(ITRACE.EQ.ITRACE/2 2))GO TO 00031
      GO TO 00032
00031 CONTINUE
      WRITE(N02,00034)TIME,IA(LB+2)
00034 FORMAT(1X,E10.4,I4,2X,19HTRNFRM(NC,FB,L1,L2))
      WRITE(N02,00035)NC,FB,L1,L2
00035 FORMAT(1X,I6,E12.4,2I6)
      CALL ICHECK
00032 CONTINUE
      CALL NCBI(L1,NCL1,BML1)
      IF(NC.LT.0)GO TO 00036
      GO TO 00037
00036 CONTINUE

```

FORTRAN Control Programs

```

      IF(NCL1.GT.0)GO TO 00039
      GO TO 00040
00036 CONTINUE
      ASSIGN 00042 TO L00001
      GO TO 00001
00042 CONTINUE
00040 CONTINUE
      GO TO 00038
00037 IF(NC.GT.0)GO TO 00043
      GO TO 00044
00043 CONTINUE
      IF(NCL1.GE.NC)GO TO 00045
      GO TO 00046
00045 CONTINUE
      ASSIGN 00048 TO L00001
      GO TO 00001
00048 CONTINUE
      GO TO 00047
00046 CONTINUE
      WRITE(N01,00049)
00049 FORMAT(1X,15H      ERROR      /1X,
X      34HTRNFRM NOT ENOUGH CONTAINERS AT L1)
00047 CALL ERROR
00044 CONTINUE
00038 CONTINUE
      IF(FB.GT.0.1GO TO 00050
      GO TO 00051
00050 CONTINUE
      IF(BML1.GT.0.)GO TO 00053
      GO TO 00054
00053 CONTINUE
      ASSIGN 00056 TO L00002
      GO TO 00002
00056 CONTINUE
00054 CONTINUE
00051 CONTINUE
      CALL UMD(L1)
      CALL UMD(L2)
      IF(ITRACE.GE.5)GO TO 00057
      GO TO 00058
00057 CONTINUE
      WRITE(N01,00060)TIME,1A(LB+2)
00060 FORMAT(1X,E10.4,14,2X,11HTRNFRM DONE)
00058 CONTINUE
      IF(ITRACE.GE.6.AND.(ITRACE.EQ.ITRACE/2 2))GO TO 00061
      GO TO 00062
00061 CONTINUE
      WRITE(N02,00064)TIME,1A(LB+2)
00064 FORMAT(1X,E10.4,14,2X,11HTRNFRM DONE)
      CALL ICHECK
00062 CONTINUE

```

FORTRAN Control Programs

```

      RETURN
C
00001 CONTINUE
      ASSIGN 00065 TO L00003
      GO TO 00003
00065 CONTINUE
      ASSIGN 00066 TO L00004
      GO TO 00004
00066 CONTINUE
      GO TO L00001, (00042,00048)
C
00002 CONTINUE
      ASSIGN 00067 TO L00005
      GO TO 00005
00067 CONTINUE
      ASSIGN 00068 TO L00006
      GO TO 00006
00068 CONTINUE
      ASSIGN 00069 TO L00007
      GO TO 00007
00069 CONTINUE
      GO TO L00002, (00056)
C
00003 CONTINUE
      L=IA(L+5)
      IF(IA(L+6).EQ.1)GO TO 00070
      GO TO 00071
00070 CONTINUE
      M=L+4
      L=IA(L+4)
      GO TO 00072
00071 IF(IA(L+6).EQ.2)GO TO 00073
      GO TO 00074
00073 CONTINUE
      M=L+5
00074 CONTINUE
00072 CONTINUE
      LIFC=L
      IF(NC.LT.0)GO TO 00075
      GO TO 00076
00075 CONTINUE
      IA(M)=0
      GO TO 00077
00076 CONTINUE
      N=1
00078 IF(N.LT.NC)GO TO 00079
      GO TO 00080
00079 CONTINUE
      L=IA(L+4)
      N=N+1
      GO TO 00078
00080 CONTINUE

```

FORTRAN Control Programs

```

      IA(M)=IA(L+4)
      IA(L+4)=0
00077 CONTINUE
      GO TO L00003, (00065)
C
00004 CONTINUE
      IF(IA(L2+5).EQ.0)GO TO 00081
      GO TO 00082
00081 CONTINUE
      IA(L2+5)=L1FC
      IA(L1FC+2)=1
      GO TO 00083
00082 CONTINUE
      L=IA(L2+5)
00084 IF(IA(L+4).GT.0)GO TO 00085
      GO TO 00086
00085 CONTINUE
      L=IA(L+4)
      GO TO 00084
00086 CONTINUE
      IA(L+4)=L1FC
      IA(L1FC+3)=L
      IA(L1FC+2)=0
00083 CONTINUE
      GO TO L00004, (00065)
C
00005 CONTINUE
      IF(IA(L1+6).EQ.1)GO TO 00087
      GO TO 00088
00087 CONTINUE
      L1B=L1
      GO TO 00089
00088 CONTINUE
      L=IA(L1+5)
      IF(IA(L+6).EQ.1)GO TO 00090
      GO TO 00091
00090 CONTINUE
      L1B=L
      GO TO 00092
00091 CONTINUE
      WRITE(N01,00093)
00093 FORMAT(/1X,15H      ERROR      /1X,26HTRNFRM LOCATING BULK AT L1)
00092 CONTINUE
      GO TO 00089
00089 CONTINUE
      GO TO L00005, (00067)
C
00006 CONTINUE
      IF(IA(L2+6).EQ.1)GO TO 00094
      GO TO 00095
00094 CONTINUE
      L2B=L2

```

FORTRAN Control Programs

```

      GO TO 00096
00095 IF (IA(L2+5).GT.0)GO TO 00097
      GO TO 00098
00097 CONTINUE
      L=IA(L2+5)
      IF (IA(L+6).EQ.1)GO TO 00099
      GO TO 00100
00099 CONTINUE
      L2B=L
      GO TO 00101
00100 IF (IA(L+6).EQ.2)GO TO 00102
      GO TO 00103
00102 CONTINUE
      CALL CMB(IA(L2+5),L2,0,L,0,1,0,0,0.)
      IA(L2)=0
      IA(L+3)=IA(L2+5)
      L2B=IA(L2+5)
      GO TO 00101
00103 CONTINUE
      WRITE(NO1,00104)
00104 FORMAT(/1X,15H      ERROR      /1X,26HTRNFRM LOCATING BULK AT 12)
      CALL ERROR
00101 CONTINUE
      GO TO 00096
00098 IF (IA(L2+5).EQ.0)GO TO 00105
      GO TO 00106
00105 CONTINUE
      CALL CMB(IA(L2+5),L2,0,0,0,1,0,0,0.)
      L2B=IA(L2+5)
      GO TO 00096
00106 CONTINUE
      WRITE(NO1,00107)
00107 FORMAT(/1X,15H      ERROR      /1X,26HTRNFRM LOCATING BULK AT L2)
      CALL ERROR
00096 CONTINUE
      GO TO L00006, (00068)
C
00007 CONTINUE
      L=IA(L15+5)
00108 IF (L.GT.0)GO TO 00109
      GO TO 00110
00109 CONTINUE
      LH2=L2B
      LP2=0
      LM=L2B*5
      M=IA(LM)
00111 IF (M.GT.0.AND.1A(M+7).NE.1A(L+7))GO TO 00112
      GO TO 00113
00112 CONTINUE
      LH2=0
      LP2=M
      M=IA(M+4)

```

FORTRAN Control Programs

```

GO TO 00111
00113 CONTINUE
IF(M.EQ.0)GO TO 00114
GO TO 00115
00114 CONTINUE
IF(LH2.GT.0)GO TO 00117
GO TO 00118
00117 CONTINUE
CALL CMB(M,LH2,0,0,0,0,1A(L+7),0,0.)
1A(LH2+5)=M
GO TO 00119
00118 CONTINUE
CALL CMB(M,0,LP2,0,0,0,1A(L+7),0,0.)
1A(LP2+4)=M
00119 CONTINUE
00115 CONTINUE
NT=8*FLOAT(1A(L+8))
1A(M+8)=1A(M+8)+NT
1A(L+8)=1A(L+8)-NT
T=FB*RA(L+9)
RA(M+9)=RA(M+9)+T
RA(L+9)=RA(L+9)-T
L=1A(L+4)
GO TO 00108
00110 CONTINUE
GO TO L00007, (00069)
END
C UMD(LM)
C SUBROUTINE UPDATES MATERIAL DATA AT LM.
SUBROUTINE UMD(LM)
DIMENSION 1A(1)
COMMON RA(1)
EQUIVALENCE (1A(1),RA(1))
COMMON /SYSTEM/MFROM,MT0,ICTYPE,IPOINT,
11DUMP,ISECA,ITRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
1LEC,LFFB,LFFE,L1,LLE,LLFB,LLFE,LMS,LNE,LP,LV,MINNFI,
1MNSF,NFE,NFL,N11,N12,N01,N02,NREP,NRUN,NSE,TIME,TSTOP
IF(ITRACE.GE.5)GO TO 00026
GO TO 00027
00026 CONTINUE
WRITE(N01,00029)TIME,1A(LB+2)
00029 FORMAT(1X,E10.4,14,2X,7HUMD(LM))
WRITE(N01,00030)LM
00030 FORMAT(1X,110)
00027 CONTINUE
IF(ITRACE.GE.6.AND.(ITRACE.EQ.ITRACE/2 2))GO TO 00031
GO TO 00032
00031 CONTINUE
WRITE(N02,00034)TIME,1A(LB+2)
00034 FORMAT(1X,E10.4,14,2X,7HUMD(LM))
WRITE(N02,00035)LM
00035 FORMAT(1X,110)

```

FORTRAN Control Programs

```

      CALL ICHECK
00032 CONTINUE
      MODE=1
      L=LM
00036 IF(L.NE.LM.OR.MODE.NE.4)GO TO 00037
      GO TO 00038
00037 CONTINUE
      ASSIGN 00039 TO L00001
      GO TO 00001
00039 CONTINUE
      GO TO 00036
00038 CONTINUE
      IF(TRACE.GE.5)GO TO 00040
      GO TO 00041
00040 CONTINUE
      WRITE(NO1,00043)TIME,IA(LB+2)
00043 FORMAT(1X,E10.4,14,2X,8HUMD DONE)
00041 CONTINUE
      IF(TRACE.GE.6.AND.(TRACE.EQ.1TRACE/2 2))GO TO 00044
      GO TO 00045
00044 CONTINUE
      WRITE(NO2,00047)TIME,IA(LB+2)
00047 FORMAT(1X,E10.4,14,2X,8HUMD DONE)
      CALL ICHECK
00045 CONTINUE
      RETURN
C
00001 CONTINUE
      IF(MODE.EQ.1)GO TO 00048
      GO TO 00049
00048 CONTINUE
      IF(IA(L+5).GT.0)GO TO 00051
      GO TO 00052
00051 CONTINUE
      L=IA(L+5)
      GO TO 00053
00052 IF(IA(L+4).GT.0)GO TO 00054
      GO TO 00055
00054 CONTINUE
      MODE=2
      L=IA(L+4)
      GO TO 00053
00055 CONTINUE
      MODE=3
      ASSIGN 00056 TO L00002
      GO TO 00002
00056 CONTINUE
00053 CONTINUE
      GO TO 00050
00049 IF(MODE.EQ.2)GO TO 00057
      GO TO 00058
00057 CONTINUE

```


FORTRAN Control Programs

```

      IF (IA(L+5).GT.0)GO TO 00059
      GO TO 00060
00059 CONTINUE
      MODE=1
      L=IA(L+5)
      GO TO 00061
00060 IF (IA(L+4).GT.0)GO TO 00062
      GO TO 00063
00062 CONTINUE
      L=IA(L+4)
      GO TO 00061
00063 CONTINUE
      MODE=3
      ASSIGN 00064 TO L00002
      GO TO 00062
00064 CONTINUE
00061 CONTINUE
      GO TO 00050
00058 IF (MODE.EQ.3)GO TO 00065
      GO TO 00066
00065 CONTINUE
      ASSIGN 00067 TO L00003
      GO TO 00063
00067 CONTINUE
      IF (IA(L+2).GT.0)GO TO 00068
      GO TO 00069
00068 CONTINUE
      MODE=4
      L=IA(L+2)
      ASSIGN 00071 TO L00005
      GO TO 00065
00071 CONTINUE
      GO TO 00070
00069 IF (IA(L+3).GT.0)GO TO 00072
      GO TO 00073
00072 CONTINUE
      L=IA(L+3)
      GO TO 00070
00073 CONTINUE
      IF (L.EQ.LM)GO TO 00074
      GO TO 00075
00074 CONTINUE
      MODE=4
      ASSIGN 00077 TO L00002
      GO TO 00062
00077 CONTINUE
      ASSIGN 00078 TO L00005
      GO TO 00065
00078 CONTINUE
      GO TO 00076
00075 CONTINUE
      ASSIGN 00079 TO L00004

```

FORTRAN Control Programs

```

GO TO 00004
00079 CONTINUE
00076 CONTINUE
00070 CONTINUE
GO TO 00050
00066 IF(MODE.EQ.4)GO TO 00080
GO TO 00081
00080 CONTINUE
IF(IA(L+4).GT.0)GO TO 00082
GO TO 00083
00082 CONTINUE
MODE=2
L=IA(L+4)
GO TO 00084
00083 IF(IA(L+3).GT.0)GO TO 00085
GO TO 00086
00085 CONTINUE
MODE=3
L=IA(L+3)
GO TO 00084
00086 IF(IA(L+2).GT.0)GO TO 00087
GO TO 00088
00087 CONTINUE
L=IA(L+2)
ASSIGN 00089 TO L00005
GO TO 00005
00089 CONTINUE
GO TO 00084
00088 CONTINUE
ASSIGN 00090 TO L00004
GO TO 00004
00090 CONTINUE
00084 CONTINUE
GO TO 00050
00081 CONTINUE
ASSIGN 00091 TO L00004
GO TO 00004
00091 CONTINUE
00050 CONTINUE
GO TO L00001, (00039)
C
00002 CONTINUE
N=0
A=0.
GO TO L00002, (00056,00064,00077)
C
00003 CONTINUE
IF(IA(L+8).GT.0.OR.RA(L+7).GT.0.)GO TO 00092
GO TO 00093
00092 CONTINUE
N=N+IA(L+8)
A=A+RA(L+9)

```

FORTRAN Control Programs

```

00093 GO TO 00094
      CONTINUE
      ASSIGN 00095 TO L00006
      GO TO 00006
00095 CONTINUE
00094 CONTINUE
      GO TO L00003, (00067)
C
00004 CONTINUE
      WRITE(N01,00096)
00096 FORMAT(/1X,15H      ERROR      /1X,27HUMD MATERIAL DATA STRUCTURE)
      CALL ERROR
      GO TO L00004, (00079,00090,00091)
C
00005 CONTINUE
      IA(L+8)=N
      RA(L+9)=A
      GO TO L00005, (00071,00078,00089)
C
00006 CONTINUE
      IF(L.NE.LM)GO TO 00097
      GO TO 00098
00097 CONTINUE
      IF(IA(L+5).GT.0)GO TO 00100
      GO TO 00101
00100 CONTINUE
      WRITE(N01,00103)
00103 FORMAT(/1X,15H      ERROR      /1X,17HUMD REMOVEBLOCK 1)
      CALL ERROR
      GO TO 00102
00101 IF(IA(L+3).GT.0.AND. IA(L+2).EQ.0)GO TO 00104
      GO TO 00105
00104 CONTINUE
      IA(IA(L+3)+4)=IA(L+4)
      IF(IA(L+4).GT.0)GO TO 00106
      GO TO 00107
00105 CONTINUE
      IA(IA(L+4)+3)=IA(L+3)
00107 CONTINUE
      GO TO 00102
00105 IF(IA(L+2).GT.0.AND. IA(L+3).EQ.0)GO TO 00109
      GO TO 00110
00109 CONTINUE
      IA(IA(L+2)+5)=IA(L+4)
      IF(IA(L+4).GT.0)GO TO 00111
      GO TO 00112
00111 CONTINUE
      IA(IA(L+4)+2)=IA(L+2)
00112 CONTINUE
      GO TO 00102
00110 CONTINUE
      WRITE(N01,00114)

```

FORTRAN Control Programs

```

00114 FORMAT(/IX,15H      ERROR      /IX,17HMD REMOVEBLOCK 2)
      CALL ERROR
00102 CONTINUE
      CALL FSTORE(L)
00098 CONTINUE
      GO TO L00006, (00095)
      END
C URN(IX,U)
C SUBROUTINE RETURNS UNIFORMLY DISTRIBUTED INTEGER IX ON THE INTERVAL
C (1,10000) AND UNIFORMLY DISTRIBUTED REAL U ON THE INTERVAL (0.,1.).
      SUBROUTINE URN(IX,U)
      DIMENSION IA(1)
      COMMON RA(1)
      EQUIVALENCE ((IA(1),RA(1))
      COMMON /SYSTEM/MFROM,MT0,ICTYPE,IPONT,
      IDUMP,ISECA,ITRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
      ILEC,LFFB,LFFE,L1,LLE,LLFB,LLFE,LMS,LNE,LP,LV,MINRFL,
      MNSE,NFE,NFL,N11,N12,N01,N02,NREP,NRUN,NSE,TIME,TSTOP
      DATA X/O.
      IF(IX.LT.0)GO TO 00026
      GO TO 00027
00026 CONTINUE
      CALL RANGET(X)
      IF(ITRACE.GE.5)GO TO 00029
      GO TO 00030
00029 CONTINUE
      WRITE(N01,00032)TIME,IA(LB+2)
00032 FORMAT(IX,E10.4,14,2X,3HURN)
      WRITE(N01,00033)X
00033 FORMAT(IX,SHRANF SEED,E12.4)
00030 CONTINUE
      IF(ITRACE.GE.6.AND.(ITRACE.EQ.1TRACE/2 2))GO TO 00034
      GO TO 00035
00034 CONTINUE
      WRITE(N02,00037)TIME,IA(LB+2)
00037 FORMAT(IX,E10.4,14,2X,3HURN)
      WRITE(N02,00038)X
00038 FORMAT(IX,SHRANF SEED,E12.4)
      CALL ICHECK
00035 CONTINUE
      CALL RANSET(X)
      GO TO 00029
00027 CONTINUE
      U=РАНF(0)
      IX=U 10000.+1
00028 CONTINUE
      RETURN
      END
C WRMD(LM)
C SUBROUTINE WRITES MATERIAL DATA AT LM ON OUTPUT DEVICE N01.
      SUBROUTINE WRMD(LM)
      DIMENSION IA(1)

```

FORTTRAN Control Programs

```

COMMON RA(1)
EQUIVALENCE (JA(1),RA(1))
COMMON /SYSTEM/MFROM,MTD,ICTYPE,IPPOINT,
1:IDUMP,ISECA,ITRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
1:LEC,LFFB,LFFE,L1,L2,L3,LFB,LLFE,LMS,LME,LP,LV,MINNLF,
1:HNSE,NFE,NFL,N11,N2,N3,N51,N02,NREP,NRUN,NSE,TIME,TSTOP
IF(LH,LE,0)GO TO 00026
GO TO 00027
00026 CONTINUE
WRITE(N01,00029)LM,
00029 FORMAT(11X,14HND MATERIAL AT,16)
RETURN
00027 CONTINUE
MODE=1
L=LM
IF(L,NE,LM,OR,MODE,NE,4)GO TO 00031
GO TO 00032
00031 CONTINUE
ASSIGN 00033 TO L00001
GO TO 00001
00033 CONTINUE
GO TO 00030
00032 CONTINUE
ASSIGN 00034 TO L00002
GO TO 00002
00034 CONTINUE
RETURN
C
00001 CONTINUE
IF(MODE,EQ,1)GO TO 00035
GO TO 00036
00035 CONTINUE
IF(IA(L+5).GT.0)GO TO 00038
GO TO 00039
00038 CONTINUE
L=IA(L+5)
GO TO 00040
00039 IF(IA(L+4).GT.0)GO TO 00041
GO TO 00042
00041 CONTINUE
MODE=2
L=IA(L+4)
GO TO 00040
00042 CONTINUE
MODE=3
00040 CONTINUE
GO TO 00037
00036 IF(MODE,EQ,2)GO TO 00043
GO TO 00044
00043 CONTINUE
IF(IA(L+5).GT.0)GO TO 00045
GO TO 00046

```

FORTRAN Control Programs

```

00045 CONTINUE
      MODE=1
      L=IA(L+5)
      GO TO 00047
00046 IF(IA(L+4).GT.0)GO TO 00043
      GO TO 00049
00048 CONTINUE
      L=IA(L+4)
      GO TO 00047
00049 CONTINUE
      MODE=3
00047 CONTINUE
      GO TO 00037
00044 IF(MODE.EQ.3)GO TO 00050
      GO TO 00051
00050 CONTINUE
      IF(L.NE.LM)GO TO 00052
      GO TO 00053
00052 CONTINUE
      ASSIGN 00055 TO L00002
      GO TO 00002
00055 CONTINUE
00053 CONTINUE
      IF(IA(L+2).GT.0)GO TO 00056
      GO TO 00057
00056 CONTINUE
      MODE=4
      L=IA(L+2)
      GO TO 00058
00057 IF(IA(L+3).GT.0)GO TO 00059
      GO TO 00060
00059 CONTINUE
      L=IA(L+3)
      GO TO 00058
00060 CONTINUE
      IF(L.EQ.LM)GO TO 00061
      GO TO 00062
00061 CONTINUE
      MODE=4
      GO TO 00063
00062 CONTINUE
      ASSIGN 00064 TO L00003
      GO TO 00003
00064 CONTINUE
00063 CONTINUE
00058 CONTINUE
      GO TO 00037
00051 IF(MODE.EQ.4)GO TO 00065
      GO TO 00066
00065 CONTINUE
      IF(IA(L+4).GT.0)GO TO 00067
      GO TO 00068

```

FORTRAN Control Programs

```

00067 CONTINUE
      MODE=2
      L=IA(L+4)
      GO TO 00069
00068 IF(IA(L+3).GT.0)GO TO 00070
      GO TO 00071
00070 CONTINUE
      ASSIGN 00072 TO L00002
      GO TO 00002
00072 CONTINUE
      MODE=3
      L=IA(L+3)
      GO TO 00069
00071 IF(IA(L+2).GT.0)GO TO 00073
      GO TO 00074
00073 CONTINUE
      ASSIGN 00075 TO L00002
      GO TO 00002
00075 CONTINUE
      L=IA(L+2)
      GO TO 00069
00074 CONTINUE
      ASSIGN 00076 TO L00003
      GO TO 00003
00076 CONTINUE
00069 CONTINUE
      GO TO 00037
00066 CONTINUE
      ASSIGN 00077 TO L00003
      GO TO 00003
00077 CONTINUE
00037 CONTINUE
      GO TO L00001, (00033)
C
00002 CONTINUE
      WRITE(N01,00078) L, (IA(L+1), I=2,8), RA(L+9)
00078 FORMAT(10X,5I6,2I3,1I0,E12.4)
      GO TO L00002, (00034,00055,00072,00075)
C
00003 CONTINUE
      WRITE(N01,00079)
00079 FORMAT(1X,15H      ERROR      /1X,
X      36H MATERIAL DATA STRUCTURE ERROR (WRMD))
      CALL ERROR
      GO TO L00003, (00064,00076,00077)
      END
C Z1A(L,N)
C SUBROUTINE ZEROS N ELEMENTS OF IA STARTING AT L.
      SUBROUTINE Z1A(L,N)
      DIMENSION IA(1)
      COMMON RA(1)
      EQUIVALENCE (IA(1),RA(1))

```

FORTRAN Control Programs

```

COMMON /SYSTEM/MFROM, MTO, ICTYPE, IPOINT,
1 IDUMP, ISECA, ITRACE, LB, LBCA, LBLIST, LBPT, LBUF, LCSB, LE,
1 LEC, LFFB, LFFE, LI, LLE, LLFB, LLFE, LMS, LNE, LP, LV, MINNFL,
1 MNSE, NFE, NFL, NI1, NI2, NO1, NO2, NREP, NRUN, NSE, TIME, TSTOP
1=1-11)
00026 1=1+1
      IF((1) (1-(N)), LE.0)GO TO 00027
      GO TO 00028
00027 CONTINUE
      1A(L-1+1)=0
      GO TO 00026
00028 CONTINUE
      RETURN
      END
C ZIAC(LOC)
C SUBROUTINE SETS ALL LOCATIONS IN THE BLOCK CHAIN AT LOC TO
C INTEGER ZERO (0).
      SUBROUTINE ZIAC(LOC)
      DIMENSION 1A(1)
      COMMON RA(1)
      EQUIVALENCE (1A(1), RA(1))
      COMMON /SYSTEM/MFROM, MTO, ICTYPE, IPOINT,
1 IDUMP, ISECA, ITRACE, LB, LBCA, LBLIST, LBPT, LBUF, LCSB, LE,
1 LEC, LFFB, LFFE, LI, LLE, LLFB, LLFE, LMS, LNE, LP, LV, MINNFL,
1 MNSE, NFE, NFL, NI1, NI2, NO1, NO2, NREP, NRUN, NSE, TIME, TSTOP
      L=LOC
      N=1A(L+1)
00026 IF(N.GT.0)GO TO 00027
      GO TO 00028
00027 CONTINUE
      1=1-11)
      IF((1) (1-(N)), LE.0)GO TO 00030
      GO TO 00031
00030 CONTINUE
      1A(L+1+1)=0
      GO TO 00029
00031 CONTINUE
      IF(1A(L).GT.0)GO TO 00032
      GO TO 00033
00032 CONTINUE
      L=1A(L)
      N=1A(L+1)
      GO TO 00034
00033 CONTINUE
      N=0
00034 CONTINUE
      GO TO 00026
00028 CONTINUE
      RETURN
      END
C ZRA(L,N)

```


FORTRAN Control Programs

```

C SUBROUTINE ZEROS N ELEMENTS OF RA STARTING AT L.
  SUBROUTINE ZRA(L,N)
    DIMENSION IA(1)
    COMMON RA(1)
    EQUIVALENCE (IA(1),RA(1))
    COMMON /SYSTEM/MFROM, MTO, ICTYPE, IPOINT,
    1 IDUMP, ISECA, ITRACE, LB, LBGA, LBLIST, LBPT, LBUF, LCSB, LE,
    1 LEC, LFFB, LFFE, LI, LLE, LLFB, LLFE, LMS, LNE, LP, LV, MINNFI,
    1 MNSE, NFE, NFL, NI1, NI2, NO1, NO2, NREP, NRUN, NSE, TIME, TSTOP
    I=1-[1]
00026 I=I+1
    IF(I) (1-(N)).LE.0)GO TO 00027
    GO TO 00028
00027 CONTINUE
    RA(L-1+I)=0
    GO TO 00026
00028 CONTINUE
    RETURN
  END
C ZRAC(L)
C SUBROUTINE SETS ALL LOCATIONS IN THE BLOCK CHAIN AT LOC TO
C REAL ZERO (0.).
  SUBROUTINE ZRAC(LOC)
    DIMENSION IA(1)
    COMMON RA(1)
    EQUIVALENCE (IA(1),RA(1))
    COMMON /SYSTEM/MFROM, MTO, ICTYPE, IPOINT,
    1 IDUMP, ISECA, ITRACE, LB, LBGA, LBLIST, LBPT, LBUF, LCSB, LE,
    1 LEC, LFFB, LFFE, LI, LLE, LLFB, LLFE, LMS, LNE, LP, LV, MINNFI,
    1 MNSE, NFE, NFL, NI1, NI2, NO1, NO2, NREP, NRUN, NSE, TIME, TSTOP
    L=LOC
    N=IA(L+1)
00026 IF(N.GT.0)GO TO 00027
    GO TO 00028
00027 CONTINUE
    I=1-[1]
    I=I+1
00029 IF(I) (1-(N)).LE.0)GO TO 00030
    GO TO 00031
00030 CONTINUE
    RA(L+1+I)=0.
    GO TO 00029
00031 CONTINUE
    IF(IA(L).GT.0)GO TO 00032
    GO TO 00033
00032 CONTINUE
    L=IA(L)
    N=IA(L+1)
    GO TO 00034
00033 CONTINUE
    N=0
00034 CONTINUE

```

FORTRAN Control Programs

```

00025 GO TO 00026
      CONTINUE
      RETURN
      END
C ZTABLE(LOC)
C SUBROUTINE PUTS INTEGER O VALUES IN THE TABLE AT LOC.
      SUBROUTINE ZTABLE(LOC)
      DIMENSION IA(1)
      COMMON RA(1)
      EQUIVALENCE (IA(1),RA(1))
      COMMON /SYSTEM/MFROM,MTD,ICTYPE,IPOINT,
1IDUMP,ISECA,ITRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
1LEC,LFFB,LFE,I1,LLE,LLFB,LFE,LMS,LINE,LP,LV,MINNFI,
1MISE,NFE,NFL,NI1,NI2,NO1,NO2,NREP,NRUN,NSE,TIME,ISTOP
      N=IA(LOC+2)
      IF(IA(LOC).EQ.O)GO TO 00026
      GO TO 00027
00026 CONTINUE
      I=1-I1)
00029 IF(I1) (1-(N)).LE.O)GO TO 00030
      GO TO 00031
00030 CONTINUE
      IA(LOC+2+I)=O
      GO TO 00029
00031 CONTINUE
      GO TO 00028
00027 CONTINUE
      I=1-I1)
00032 IF(I1) (1-(N)).LE.O)GO TO 00033
      GO TO 00034
00033 CONTINUE
      IA(LCS(LOC,I+1))=O
      GO TO 00032
00034 CONTINUE
00028 CONTINUE
      RETURN
      END

```

FORTRAN Function Programs

FUNCTION PROGRAMS

```

C 101 RCMB 3/16/78
RECEIVE MATERIAL BATCHES
CCCC LP+0 1 RESPONSE POINTER
      1 3 PERIOD OF ARRIVAL SCHEDULE
      2 1 NUMBER OF ARRIVALS PER PERIOD
      3 4 SCHEDULED ARRIVAL TIMES
      4 3 MAXIMUM DEVIATION FROM SCHEDULED ARRIVAL TIMES
      5 1 NUMBER OF BULK MATERIAL TYPES
      6 4 BULK MATERIAL SPECIFICATIONS (REPEAT FOR EACH TYPE)
          INTEGER MATERIAL TYPE CODE
          MEAN COUNT
          MAXIMUM UNIFORM DEVIATION OF COUNT
          MEAN MASS
          STANDARD NORMAL DEVIATION OF MASS
          MAXIMUM MASS
          MINIMUM MASS
      7 1 NUMBER OF MATERIAL TYPES RECEIVED IN CONTAINERS
      8 4 SPECIFICATIONS FOR MATERIAL RECEIVED IN CONTAINERS
          (REPEAT THE DATA LISTED ABOVE FOR EACH TYPE)
      9 3 MEAN NUMBER OF CONTAINERS PER ARRIVAL
     10 3 MAXIMUM UNIFORM DEVIATION OF NUMBER OF CONTAINERS
     11 1 TIMING CODE
     12 1 TIMER FUNCTION
     13 1 DESTINATION FUNCTION
     14 1 TRANSPORT FUNCTION
CCCC LV+0 1 ENTRY COUNT
      1 1 NUMBER OF BATCHES RECEIVED
C LI+0 0
      1 0
SUBROUTINE B101
DIMENSION IPVT(17)
DIMENSION IA(1)
COMMON RA(1)
EQUIVALENCE (IA(1),RA(1))
COMMON /SYSTEM/MFROM, MTO, ICTYPE, IPOINT,
1IDUMP, ISECA, ITRACE, LB, LBGA, LBLIST, LBPT, LBUF, LCSB, LE,
1LEC, LFFB, LFFE, LI, LLE, LLFB, LLFE, LMS, LNE, LP, LV, MINNF,
1MNSE, NFE, NFL, NI1, NI2, NO1, NO2, NREP, NRUN, NSE, TIME, TSTOP
DATA IPVT/1,3,1,4,3,1,4,1,4,3,3,1,1,1,1,1,1/
IA(LV)=IA(LV)+1
IF(ICTYPE.GE.0)GO TO 99991
GO TO 00026
99991 GO TO (99100,99200,99300,99400,99500,99600,99700,99800,99900)
X, ICTYPE
99992 RETURN
99100 CALL CREATE(101,15,2,IPVT)
      ICTYPE=-1
      IPOINT=0

```

FORTRAN Function Programs

```

GO TO 99992
99200 CALL DESTRO(IPVT)
      ICTYPE=-2
      IPOINT=0
      GO TO 99992
99300 CALL SET(1,IPVT,INTD,REALD)
      ICTYPE=-3
      IPOINT=0
      GO TO 99992
99400 CONTINUE
      WRITE(ND1,00027)
00027 FORMAT(/1X,15H          ERROR          /1X,
X      43H101 RCMB  ATTEMPT TO SET INITIAL CONDITIONS)
      CALL ERROR
99500 CONTINUE
      WRITE(ND1,00028)IA(LB+2),IA(LV+1)
00028 FORMAT(/,9H FUNCTION,14,1X,8H101 RCMB /11X,
X      25HNUMBER OF BATCHES RECEIVED,110)
      ICTYPE=-5
      IPOINT=0
      GO TO 99992
99600 CONTINUE
      IF(RP(10).GT.RP(9))GO TO 00029
      GO TO 00030
00029 CONTINUE
      WRITE(ND1,00032)
00032 FORMAT(/1X,15H          ERROR          /1X,
X      46H101 RCMB  PARAMETER 10 LARGER THAN PARAMETER 9)
      CALL ERROR
00030 CONTINUE
      I=1-(1)
00033 IF(I) (1-(IA(LB+4)))LE.0)GO TO 00034
      GO TO 00035
00034 CONTINUE
      IA(LV-1+1)=IA(L1-1+1)
      GO TO 00033
00035 CONTINUE
      CALL SCHED(RUNIF(RLP(3,1),RP(4)),IA(LB+2),1,0)
      ICTYPE=-6
      IPOINT=0
      GO TO 99992
99700 CALL DUMP(IPVT)
      ICTYPE=-7
      IPOINT=0
      GO TO 99992
99800 CONTINUE
      IF(TRACE.GE.1)GO TO 00036
      GO TO 00037
00036 CONTINUE
      WRITE(ND1,00039)TIME,IA(LB+2)
00039 FORMAT(1X,E10.4,14,2X,5HEVENT)

```

FORTRAN Function Programs

```

00037 CONTINUE
      IF (ITRACE.GE 2 AND. ITRACE.EQ. ITRACE/2 2))GO TO 00040
      GO TO 00041
00040 CONTINUE
      WRITE(NC2,00043)TIME,IA(LB+2)
00043 FORMAT(1X,E10.4,14,2X,SHEVENT)
      CALL ICHECK
00041 CONTINUE
      CALL CEC(1,IP(2),NARVL)
      IF(IP(11).EQ.1)GO TO 00044
      GO TO 00045
00044 CONTINUE
      I=1
      T=TIME
      GO TO 00046
00045 CONTINUE
      MFROM=IA(LB+2)
      MTO=IP(12)
      ICTYPE=9
      IPOINT=0
      IA(LP)=1
      GO TO 99992
00047 CONTINUE
      CALL TIMER(IP(11),I,T)
00046 CONTINUE
      IF(I.EQ.1)GO TO 00048
      GO TO 00049
00048 CONTINUE
      ASSIGN 00051 TO L00001
      GO TO 00001
00051 CONTINUE
      IF(LM.GT.0)GO TO 00052
      GO TO 00053
00052 CONTINUE
      ASSIGN 00055 TO L00002
      GO TO 00002
00055 CONTINUE
00053 CONTINUE
      IF(NARVL.EQ.IP(2))GO TO 00056
      GO TO 00057
00056 CONTINUE
      NARVL=1
      I=1
      GO TO 00058
00057 CONTINUE
      NARVL=NARVL+1
      I=0
00058 CONTINUE
      T=RUNIF(FLOAT(IFIX(TIME/RP(1))+1) RP(1)+RLP(3,NARVL),RP(4))
      CALL SCHED(T,IA(LB+2),NARVL,0)
      GO TO 00050
00049 CONTINUE

```

FORTRAN Function Programs

```

      CALL SCHED(T,IA(LB+2),NARVL,0)
00050 CONTINUE
      ICTYPE=-8
      IPOINT=0
      GO TO 99992
99900 CONTINUE
      IF(ITRACE.GE.3)GO TO 00059
      GO TO 00060
00059 CONTINUE
      WRITE(N01,00062)TIME,IA(LB+2)
00062 FORMAT(1X,E10.4,14,2X,11HINTERACTION)
00060 CONTINUE
      IF(ITRACE.GE.4.AND.(ITRACE.EQ.ITRACE/2 2))GO TO 00063
      GO TO 00064
00063 CONTINUE
      WRITE(N02,00066)TIME,IA(LB+2)
00066 FORMAT(1X,E10.4,14,2X,11HINTERACTION)
      CALL ICHECK
00064 CONTINUE
      WRITE(N01,00067)
00067 FORMAT(1X,15H      ERROR      /1X,
X      29H101 RCMB  ILLEGAL INTERACTION)
      CALL ERROR
      ICTYPE=-9
      IPOINT=0
      GO TO 99992
C
00001 CONTINUE
      IF(IP(5).GT.0)GO TO 00068
      GO TO 00069
00068 CONTINUE
      CALL CRMD1(IP(5),IP(6),LBLK)
00069 CONTINUE
      IF(IP(7).GT.0)GO TO 00071
      GO TO 00072
00071 CONTINUE
      NC=IFIX(RUNIF(IP(9),IP(10)))
      IF(NC.GT.0)GO TO 00074
      GO TO 00075
00074 CONTINUE
      CALL CRMD2(NC,IP(7),IP(8),LCON)
00075 CONTINUE
00072 CONTINUE
      IF(IP(5).GT.0.AND.IP(7).GT.0)GO TO 00077
      GO TO 00078
00077 CONTINUE
      CALL TRNFRM(0,1.,LBLK,LCON)
      LM=LCON
      GO TO 00079
00078 IF(IP(5).EQ.0.AND.IP(7).GT.0)GO TO 00080
      GO TO 00081
00080 CONTINUE

```

FORTRAN Function Programs

```

      LM=LCON
      GO TO 00079
00081 IF (IP(51).GT.0.AND.IP(7).EQ.0)GO TO 00082
      GO TO 00083
00082 CONTINUE
      LM=LBLK
      GO TO 00079
00083 CONTINUE
      LM=0
00079 CONTINUE
      GO TO L00001, (00051)
C
00002 CONTINUE
      CALL PCOM(IP(14),9,1,LM,IA(LB+2),IP(13),0,0,0,0)
      IA(LP)=2
      GO TO 99992
00084 CONTINUE
      IF(IA(IPOINT+2).NE.-1)GO TO 00085
      GO TO 00086
00085 CONTINUE
      WRITE(N01,00088)
00088 FORMAT(1X,15H      ERROR      /1X,
X      45H101 RCMB MATERIAL NOT REMOVED AT ARRIVAL TIME)
      CALL ERROR
00086 CONTINUE
      CALL FSTORE(IPOINT)
      IF(ITRACE.GE.1)GO TO 00089
      GO TO 00090
00089 CONTINUE
      WRITE(N01,00092)TIME,IA(LB+2)
00092 FORMAT(1X,E10.4,14,2X,33H101 RCMB MATERIAL BATCH RECEIVED)
00090 CONTINUE
      IF(ITRACE.GE.2.AND.(ITRACE.EQ.1TRACE/2 2))GO TO 00093
      GO TO 00094
00093 CONTINUE
      WRITE(N02,00096)TIME,IA(LB+2)
00096 FORMAT(1X,E10.4,14,2X,33H101 RCMB MATERIAL BATCH RECEIVED)
      CALL ICHECK
00094 CONTINUE
      IA(LV+1)=IA(LV+1)+1
      GO TO L00002, (00055)
00026 100000=IA(LP)
      GO TO (00047,00084), 100000
      END
C 102 IDAS 3/16/78
C INITIATE DIVERSION ACTION SEQUENCE
C LP=0 1 RESPONSE POINTER
C 1 4 MATERIAL DIVERSION OBJECTIVE TYPE (1 = COUNTABLE ITEMS
C OF BULK MATERIAL, 2 = MASS OF BULK MATERIAL, 3 = COUNTABLE
C ITEMS FROM CONTAINERS, 4 = MASS FROM CONTAINERS, 5 =
C CONTAINERS)
C MATERIAL TYPE (0 FOR BLENDED MATERIALS)

```

FORTRAN Function Programs

```

C      MEAN NUMBER OF COUNTABLE ITEMS
C      MAX UNIFORM DEVIATION OF COUNTABLE ITEMS
C      MEAN MASS
C      STANDARD DEVIATION OF MASS
C      MEAN NUMBER OF CONTAINERS
C      MAXIMUM UNIFORM DEVIATION OF NUMBER OF CONTAINERS
C      EVENT SEQUENCE SPECIFICATION. REPEAT FOR EACH EVENT:
C      1) EVENT FUNCTION (NEGATIVE FOR FIRST EVENT OF SUBSEQUENCES),
C      2) EVENT STARTING TIME DELAY DISTRIBUTION TYPE 3) EVENT
C      STARTING TIME DELAY STATISTICS (0 OR MORE VALUES)
C LV+0 1 ENTRY COUNT
C LI+0 0
      SUBROUTINE B102
      DIMENSION IPVT(3)
      DIMENSION IA(1)
      COMMON RA(1)
      EQUIVALENCE (IA(1),RA(1))
      COMMON /SYSTEM,MFROM,MTO,ICTYPE,IPOINT,
      1DUMP,ISECA,TRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
      1LEC,LFB,LFFE,L1,LLE,LLFB,LLFE,LMS,LNE,LP,LV,MINNFI,
      1MNSE,NFE,NFL,N11,N12,N01,N02,NREP,NRUN,NSE,TIME,TSTOP
      DATA IPVT/1,4,1/
      IA(LV)=IA(LV)+1
      IF (ICTYPE.GE.0) GO TO 99991
      GO TO 00026
99991 GO TO (99100,99200,99300,99400,99500,99600,99700,99800,99900)
      X, ICTYPE
99992 RETURN
99100 CALL CREATE(102,2,1,IPVT)
      ICTYPE=-1
      IPOINT=0
      GO TO 99992
99200 CALL DESTRO(IPVT)
      ICTYPE=-2
      IPOINT=0
      GO TO 99992
99300 CALL SET(1,IPVT,INTD,REALD)
      ICTYPE=-3
      IPOINT=0
      GO TO 99992
99400 CONTINUE
      WRITE(N01,00027)
00027 FORMAT(/1X,15H ERROR /1X,
      X 49H102 IDAS ATTEMPT TO SET INITIAL CONDITIONS)
99500 CONTINUE
      WRITE(N01,00028)IA(LB+2)
00028 FORMAT(/,9H FUNCTION,I4,1X,6H102 IDAS/11X,9HNO REPORT)
      ICTYPE=-5
      IPOINT=0
      GO TO 99992
99600 CONTINUE

```


FORTRAN Function Programs

```

00029 I=I-1)
      I=I+1
      IF((I) (1-(IA(LB+4))))LE.0)GO TO 00030
      GO TO 00031
00030 CONTINUE
      IA(LV-1+1)=IA(LI-1+1)
      GO TO 00029
00031 CONTINUE
      I=I-1)
      I=I+1
      IF((I) (1-(IA(IP(1)+3))))LE.0)GO TO 00033
      GO TO 00034
00033 CONTINUE
      IF(RLP(1,1).LT.0)GO TO 00035
      GO TO 00036
00035 CONTINUE
      CALL SAMPL(IP(1),I+1,TD)
      CALL SCHED(TD,-IFIX(RLP(1,1)),I,IP(1))
00036 CONTINUE
      GO TO 00032
00034 CONTINUE
      ICTYPE=-6
      IPOINT=0
      GO TO 99992
99700 CALL DUMP(IPVT)
      ICTYPE=-7
      IPOINT=0
      GO TO 99992
99800 CONTINUE
      IF(ITRACE.GE.1)GO TO 00038
      GO TO 00039
00038 CONTINUE
      WRITE(NO1,00041)TIME,IA(LB+2)
00041 FORMAT(1X,E10.4,14,2X,SHEVENT)
00039 CONTINUE
      IF(ITRACE.GE.2.AND.(ITRACE.EQ.ITRACE/2 2))GO TO 00042
      GO TO 00043
00042 CONTINUE
      WRITE(NO2,00045)TIME,IA(LB+2)
00045 FORMAT(1X,E10.4,14,2X,SHEVENT)
      CALL ICHECK
00043 CONTINUE
      WRITE(NO1,00046)
00046 FORMAT(/1X,15H      ERROR      /1X,
X      31H102 IDAS  ILLEGAL EVENT MESSAGE)
      CALL ERROR
      ICTYPE=-8
      IPOINT=0
      GO TO 99992
99900 CONTINUE
      IF(ITRACE.GE.3)GO TO 00047
      GO TO 00048

```

FORTRAN Function Programs

```

00047 CONTINUE
      WRITE(N01,00050)TIME,IA(LB+2)
00050 FORMAT(1X,E10.4,14,2X,11HINTERACTION)
00048 CONTINUE
      IF(1TRACE.EQ.4.AND.(1TRACE.EQ.1TRACE/2))GO TO 00051
      GO TO 00052
00051 CONTINUE
      WRITE(N02,00054)TIME,IA(LB+2)
00054 FORMAT(1X,E10.4,14,2X,11HINTERACTION)
      CALL ICHECK
00052 CONTINUE
      WRITE(N01,00055)
00055 FORMAT(/1X,15H      ERROR      /1X,
X      29H102 IDAS  ILLEGAL INTERACTION)
      CALL ERROR
      ICTYPE=3
      IPOINT=0
      GO TO 99992
00026 CONTINUE
      WRITE(N01,00056)
00056 FORMAT(/1X,15H      ERROR      /1X,
X      37H11LEGAL RESPONSE COMMUNICATION (BNNN))
      CALL ERROR
      END
C 301 TRMB      3/16/78
C TRANSPORT MATERIAL BATCHES
C LP+0 1 RESPONSE POINTER
C 1 3 MEAN TRANSPORT TIME
C 2 3 MAXIMUM UNIFORM DEVIATION OF TRANSPORT TIME
C 3 2 FUNCTIONS NOTIFIED AT CONDITION TIMES
C 4 1 DUMMY PARAMETER USED AS SAVE LOCATION
C LV+0 1 ENTRY COUNT
C 1 1 NUMBER OF BATCHES IN TRANSPORT
C 2 1 MAXIMUM NUMBER OF BATCHES IN TRANSPORT
C 3 3 TIME MAXIMUM NUMBER OF BATCHES FIRST OCCURRED
C 4 1 LN OF MOST RECENT BATCH IN TRANSPORT
C LI+0 0
C 1 0
C 2 0
C 3 0
C 4 0
      SUBROUTINE B301
      DIMENSION IPVT(10)
      DIMENSION IA(1)
      COMMON RA(1)
      EQUIVALENCE (IA(1),RA(1))
      COMMON /SYSTEM/MFROM,MT0,ICTYPE,IPOINT,
      1IDUMP,ISECA,1TRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
      1LEG,LFFB,LFPE,L1,LLE,LLFB,LLFE,LMS,LNE,LP,LV,MINNFI,
      1INSE,NFE,NFL,N11,N12,N01,N02,NREP,NRUN,NSE,TIME,TSTOP
      DATA IPVT/1,3,3,2,1,1,1,1,3,1/
      IA(LV)=IA(LV)+1

```

FORTRAN Function Programs

```

      IF(ICTYPE.GE.0)GO TO 99991
      GO TO 00026
99991 GO TO (99100,99200,99300,99400,99500,99600,99700,99800,99900)
      X ICTYPE
99992 RETURN
99100 CALL CREATE(301,5,5,IPVT)
      ICTYPE=-1
      IPOINT=0
      GO TO 99992
99200 CONTINUE
      IF(IV(1).GT.0)GO TO 00027
      GO TO 00026
00027 CONTINUE
      ASSIGN 00030 TO L00003
      GO TO 00003
00030 CONTINUE
00026 CONTINUE
      CALL DESTRO(IPVT)
      ICTYPE=-2
      IPOINT=0
      GO TO 99992
99300 CALL SET(,IPVT,INTD,REALD)
      ICTYPE=-3
      IPOINT=0
      GO TO 99992
99400 CONTINUE
      WRITE(NO1,00031)
00031 FORMAT(/1X,15H      ERROR      /1X,
      X 43H301 TRMB  ATTEMPT TO SET INITIAL CONDITIONS)
      CALL ERROR
99500 CONTINUE
      WRITE(NO1,00032)IA(LB+2),IA(LV+2),RA(LV+3)
00032 FORMAT(/,9H FUNCTION,14,1X,9H301/TRMB /11X,
      X 36HMAXIMUM NUMBER OF BATCHES IN TRANSPORT,110 /11X,
      X 27HTIME MAXIMUM FIRST OCCURRED,9X,E12.4)
      ICTYPE=-5
      IPOINT=0
      GO TO 99992
99600 CONTINUE
      IF(RP(2).GT.RP(1))GO TO 00033
      GO TO 00034
00033 CONTINUE
      WRITE(NO1,00036)
00036 FORMAT(/1X,15H      ERROR      /1X,
      X 45H301 TRMB  PARAMETER 2 LARGER THAN PARAMETER 1)
      CALL ERROR
00034 CONTINUE
      IF(IV(1).GT.0)GO TO 00037
      GO TO 00036
00037 CONTINUE
      ASSIGN 00040 TO L00003
      GO TO 00003

```

FORTRAN Function Programs

```

00040 CONTINUE
00038 CONTINUE
      I=I-1)
00041 I=I+1
      IF(1) (1 (1A, LB+4))) .LE. 0) GO TO 00042
      GO TO 00043
00042 CONTINUE
      1A(LV-1+1)=1A(LI-1+1)
      GO TO 00041
00043 CONTINUE
      ICTYPE=-6
      IPOINT=0
      GO TO 99992
99700 CALL DUMP(IPVT)
      ICTYPE=-7
      IPOINT=0
      GO TO 99992
99800 CONTINUE
      IF(1TRACE.GE.1) GO TO 00044
      GO TO 00045
00044 CONTINUE
      WRITE(NOT, 00047) TIME, 1A(LB+2)
00047 FORMAT(1X, E10.4, 14, 2X, 5HEVENT)
00045 CONTINUE
      IF(1TRACE.GE.2.AND.(1TRACE.EQ.1TRACE/2 2)) GO TO 00048
      GO TO 00049
00048 CONTINUE
      WRITE(NOT, 00051) TIME, 1A(LB+2)
00051 FORMAT(1X, E10.4, 14, 2X, 5HEVENT)
      CALL ICHECK
00049 CONTINUE
      CALL CEC(1, 1, KODE)
      LN=1A(IPOINT+1)
      ASSIGN 00052 TO L00001
      GO TO 00001
00052 CONTINUE
      ICTYPE=-8
      IPOINT=0
      GO TO 99992
99900 CONTINUE
      IF(1TRACE.GE.3) GO TO 00053
      GO TO 00054
00053 CONTINUE
      WRITE(NOT, 00056) TIME, 1A(LB+2)
00056 FORMAT(1X, E10.4, 14, 2X, 11HINTERACTION)
00054 CONTINUE
      IF(1TRACE.GE.4.AND.(1TRACE.EQ.1TRACE/2 2)) GO TO 00057
      GO TO 00058
00057 CONTINUE
      WRITE(NOT, 00060) TIME, 1A(LB+2)
00060 FORMAT(1X, E10.4, 14, 2X, 11HINTERACTION)
      CALL ICHECK

```

FORTRAN Function Programs

```

00058 CONTINUE
      IF(IA(IPOINT+2).EQ.1)GO TO 00061
      GO TO 00062
00061 CONTINUE
      IA(LP+4)=IPOINT
      CALL COPYB(IPOINT,LN)
      IA(LV+1)=IA(LV+1)+1
      IF(IV(1).GT.IV(2))GO TO 00064
      GO TO 00065
00064 CONTINUE
      IA(LV+2)=IV(1)
      RA(LV+3)=TIME
00065 CONTINUE
      IF(RP(1).GT.0.)GO TO 00067
      GO TO 00068
00067 CONTINUE
      IF(IV(4).LE.0)GO TO 00070
      GO TO 00071
00070 CONTINUE
      IA(LN+8)=0
      IA(LN+9)=0
      IA(LV+4)=LN
      GO TO 00072
00071 CONTINUE
      IA(IV(4)+9)=LN
      IA(LN+8)=IV(4)
      IA(LN+9)=0
      IA(LV+4)=LN
00072 CONTINUE
      CALL SCHED(TIME+RUNIF(RP(1),RP(2)),IA(LB+2),1,LN)
      ASSIGN 00073 TO L00002
      GO TO 00002
00073 CONTINUE
      IF(ITRACE.GE.3)GO TO 00074
      GO TO 00075
00074 CONTINUE
      WRITE(NO1,00077)TIME,IA(LB+2)
00077 FORMAT(1X,E10.4,14,2X,27H301 TRMB TRANSPORT STARTED)
00075 CONTINUE
      IF(ITRACE.GE.4.AND.(ITRACE.EQ.ITRACE/2 2))GO TO 00078
      GO TO 00079
00078 CONTINUE
      WRITE(NO2,00081)TIME,IA(LB+2)
00081 FORMAT(1X,E10.4,14,2X,27H301 TRMB TRANSPORT STARTED)
      CALL ICHECK
00079 CONTINUE
      GO TO 00069
00068 CONTINUE
      ASSIGN 00082 TO L00001
      GO TO 00001
00082 CONTINUE
00069 CONTINUE

```

FORTRAN Function Programs

```

      IPOINT=IP(4)
      IA(IPOINT+2)=-1
      GO TO 00063
00062 IF(IA(IPOINT+2).EQ.5)GO TO 00083
      GO TO 00084
00083 CONTINUE
      IA(IPOINT+2)=-5
      IA(IPOINT+3)=0
      IA(IPOINT+4)=IV(4)
      IA(IPOINT+5)=IV(1)
      IF(ITRACE.GE.3)GO TO 00085
      GO TO 00086
00085 CONTINUE
      WRITE(NO1,00088)TIME,IA(LB+2)
00088 FORMAT(1X,E10.4,14,2X,38H301 TRMB MATERIAL ACQUISITION ATTEMPT)
00086 CONTINUE
      IF(ITRACE.GE.4.AND.(ITRACE.EQ.ITRACE/2 2))GO TO 00089
      GO TO 00090
00089 CONTINUE
      WRITE(NO2,00092)TIME,IA(LB+2)
00092 FORMAT(1X,E10.4,14,2X,38H301 TRMB MATERIAL ACQUISITION ATTEMPT)
      CALL ICHECK
00090 CONTINUE
      GO TO 00063
00084 CONTINUE
      WRITE(NO1,00093)
00093 FORMAT(/1X,15H ERROR /1X,
X 29H301 TRMB 1LLEGAL INTERACTION)
      CALL ERROR
00063 CONTINUE
      ICTYPE=-3
      IPOINT=IPPOINT
      GO TO 99992
C
00001 CONTINUE
      IF(IV(1).EQ.1)GO TO 00094
      GO TO 00095
00094 CONTINUE
      IA(LV+4)=0
      GO TO 00096
00095 CONTINUE
      IF(IV(4).EQ.LN)GO TO 00097
      GO TO 00098
00097 CONTINUE
      IA(LV+4)=IA(LN+8)
      IF(IA(LN+8).GT.0)GO TO 00100
      GO TO 00101
00100 CONTINUE
      IA(IA(LN+8)+9)=0
00101 CONTINUE
      GO TO 00099
00098 CONTINUE

```

FORTRAN Function Programs

```

      IF (IA(LN+8).EQ.0)GO TO 00103
      GO TO 00104
00103 CONTINUE      IA(IA(LN+9)+8)=0
      GO TO 00105
00104 CONTINUE      IA(IA(LN+8)+9)=IA(LN+9)
                    IA(IA(LN+9)+8)=IA(LN+8)
00105 CONTINUE
00099 CONTINUE
00096 CONTINUE
      ASSIGN 00106 TO L00002
      GO TO 00002
00106 CONTINUE
      IF (IA(LN+5).GT.0)GO TO 00107
      GO TO 00108
00107 CONTINUE
      IA(LN+8)=0
      IA(LN+9)=0
      IA(LN+4)=IA(LB+2)
      MFROM=IA(LB+2)
      MTO=IA(LN+5)
      ICTYPE=9
      IPOINT=LN
      IA(LP)=1
      GO TO 99992
00110 CONTINUE
      IF (IA(IPOINT+2).NE.-1)GO TO 00111
      GO TO 00112
00111 CONTINUE
      WRITE(N01,00114)
00114 FORMAT(/1X,15H      ERROR      /1X,
X      48H301 TRMB MATERIAL NOT REMOVED AT TRANSPORT TIME)
00112 CONTINUE
      CALL FSTORE(IPOINT)
      GO TO 00109
00108 CONTINUE
      CALL FRMD(IA(LN+3))
      CALL FSTORE(LN)
00109 CONTINUE
      IA(LV+1)=IA(LV+1)-1
      IF (ITRACE.GE.1)GO TO 00115
      GO TO 00116
00115 CONTINUE
      WRITE(N01,00118)TIME,IA(LB+2)
00118 FORMAT(1X,E10.4,14,2X,29H301 TRMB TRANSPORT COMPLETED)
00116 CONTINUE
      IF (ITRACE.GE.2.AND.(ITRACE.EQ.ITRACE/2))GO TO 00119
      GO TO 00120
00119 CONTINUE
      WRITE(N02,00122)TIME,IA(LB+2)

```

FORTRAN Function Programs

```

00122 FORMAT(1X,E10.4,14.2X,29H301 TRMB  TRANSPORT COMPLETED)
      CALL ICHECK
00120 CONTINUE
      GO TO L00001, (00052,00082)
C
00002 CONTINUE
      IF(IP(3).GT.0)GO TO 00123
      GO TO 00124
00123 CONTINUE
      CALL PCOM(ILP(3,1),9,9,0,IV(4),IV(1),0,0,0,0)
      I=1-I
      I=I+I
      IF(I) (I-(IA(IP(3)+3))).LE.0)GO TO 00127
      GO TO 00128
00127 CONTINUE
      HTG=ILP(3,1)
      ICTYPE=9
      IA(IPOINT+2)=9
      IA(LP)=2
      GO TO 99992
00129 CONTINUE
      IF(IA(IPOINT+2).NE.-9)GO TO 00130
      GO TO 00131
00130 CONTINUE
      WRITE(ND1,00133)
00133 FORMAT(/1X,15H      ERROR      /1X,
X      38H301 TRMB  RESPONSE TO CONDITION NOTICE)
      CALL ERROR
00131 CONTINUE
      GO TO 00126
00128 CONTINUE
      CALL FSTORE(IPOINT)
00124 CONTINUE
      GO TO L00002, (00073,00106)
C
00003 CONTINUE
      LN=IV(4)
00134 IF(LN.GT.0)GO TO 00135
      GO TO 00136
00135 CONTINUE
      CALL FRMD(IA(LN+3))
      CALL FSTORE(LN)
      LN=IA(LN+8)
      GO TO 00134
00136 CONTINUE
      GO TO L00003, (00030,00040)
00026 I00000=IA(LP)
      GO TO (00110,00129), I00000
      END
C 401 STOM 3/16/78
C STORE MATERIAL
C LP+0 1 RESPONSE POINTER

```


FORTRAN Function Programs

```

C      1 1 HOLD UP MODEL
C      1 = SINGLE DEPOSIT
C      2 = RE-DEPOSIT ON ONLY SURFACE AREAS CONTACTED BY
C          STATIC MATERIAL
C      3 = RE-DEPOSIT WITH WASHDOWN OF ALL INTERNAL SURFACES
C          OF THE BULK STORAGE VESSEL
C      2 3 END AREA OF BULK STORAGE VESSEL
C      3 3 VERTICAL SURFACE AREA CONTACTED PER UNIT MASS BULK STORAGE IF
C          HOLD UP MODEL 1 OR 2, OR TOTAL AREA CONTACTED IF MODEL 3
C      4 3 BULK STORAGE SURFACE HOLDUP COEFFICIENT
C      5 2 REPORT CONTROL
C          1) BULK STORAGE DATA
C          2) CONTAINER STORAGE DATA
C          3) MATERIAL CURRENTLY IN STORAGE
C          4) MATERIAL CURRENTLY IN HOLDUP
C      6 2 FUNCTIONS NOTIFIED AT DIVERSION OPPORTUNITIES
C      7 1 - MATERIAL IN STORAGE
C      8 1 - MATERIAL IN HOLDUP
C      LV+0 1 ENTRY COUNT
C      1 1 NUMBER CONTAINERS IN STORAGE
C      2 1 MAX NUMBER CONTAINERS IN STORAGE
C      3 3 TIME MAX NUMBER CONTAINERS FIRST STORED
C      4 1 NUMBER CONTAINER COUNT OBSERVATIONS
C      5 3 LAST TIME NUMBER CONTAINERS CHANGED
C      6 3 SUM OF N(T-TL) VALUES
C      7 3 SUM OF N 2(T-TL) VALUES
C      8 3 BULK MASS IN STORAGE
C      9 3 MAX BULK MASS IN STORAGE
C      10 3 TIME MAX BULK MASS FIRST STORED
C      11 1 NUMBER BULK MASS OBSERVATIONS
C      12 3 LAST TIME BULK MASS CHANGED
C      13 3 SUM OF M(T-TL) VALUES
C      14 3 SUM OF M 2(T-TL) VALUES
C      15 3 TIME DATA COLLECTION STARTED
C      LI+0 0
C      1 0
C      2 0
C      3 0
C      4 0
C      5 0
C      6 0
C      7 0
C      8 0
C      9 0
C      10 0
C      11 0
C      12 0
C      13 0
C      14 0
C      15 0
C      SUBROUTINE B401
C      DIMENSION IPVT(25)

```

FORTRAN Function Programs

```

DIMENSION IA(1)
COMMON RA(1)
EQUIVALENCE (IA(1),RA(1))
COMMON /SYSTEM/MFROM, MTO, ICTYPE, IPOINT,
1IDUMP, ISECA, ITRACE, LB, LBCA, LBLIST, LBPT, LBUF, LCSB, LE,
1LEC, LPFB, LPFE, LI, LLE, LLFB, LLFE, LMS, LNE, LP, LV, MINNFI,
1MNS, NFE, NFL, NI1, NI2, NO1, NO2, NREP, NRUN, NSE, TIME, TSTOP
DATA IPVT/1,1,3,3,3,2,2,1,1,1,1,1,3,1,3,3,3,3/
1A(LV)=IA(LV)+1
IF(ICTYPE.GE.0)GO TO 99991
GO TO 00026
99991 GO TO (99100,99200,99300,99400,99500,99600,99700,99800,99900)
X ICTYPE
99992 RETURN
99100 CALL CREATE(401,9,16,IPVT)
ICTYPE=-1
IPOINT=0
GO TO 99992
99200 CONTINUE
IF(IP(7).GT.0)GO TO 00027
GO TO 00028
00027 CONTINUE
CALL FRMD(IP(7))
00028 CONTINUE
IF(IP(8).GT.0)GO TO 00030
GO TO 00031
00030 CONTINUE
CALL FRMD(IP(8))
00031 CONTINUE
CALL DESTRO(IPVT)
ICTYPE=-2
IPOINT=0
GO TO 99992
99300 CALL SET(1,IPVT,INTD,REALD)
ICTYPE=-3
IPOINT=0
GO TO 99992
99400 CONTINUE
WRITE(NO1,00033)
00033 FORMAT(1X,15H ERROR /1X,
X CALL 49H401 STOM ATTEMPT TO SET INITIAL CONDITIONS)
99500 CONTINUE
WRITE(NO1,00034)IA(LB+2)
00034 FORMAT(/,9H FUNCTION,14,1X,6H401 STOM)
IF(ILP(5,1).EQ.1)GO TO 00035
GO TO 00036
00035 CONTINUE
RA(LV+12)=TIME-RV(15)
RA(LV+13)=RV(13)/RV(12)
RA(LV+14)=RV(14)/RV(12)-RV(13) 2
IF(RV(14).GT.0.)GO TO 00038

```

FORTTRAN Function Programs

```

GO TO 00039
00038 CONTINUE
      RA(LV+14)=SORT(RV(14))
GO TO 00040
00039 CONTINUE
      RA(LV+14)=0.
00040 CONTINUE
      WRITE(NO1,00041)RV(8),RV(9),RV(10),IV(11),RV(13),RV(14)
00041 FORMAT(/11X,12HBUCK STORAGE, /11X,20HCURRENTLY IN STORAGE,E11
X      4 /11X,19HMAXIMUM MASS STORED,E12.4,8H AT TIME,E12.4 /
X      11X,19HNUMBER OBSERVATIONS,112 /11X,4HMEAN,E27.4 /11X,
X      18HSTANDARD DEVIATION,E13.4)
00036 CONTINUE
      IF(ILP(5,2).EQ.1)GO TO 00042
GO TO 00043
00042 CONTINUE
      RA(LV+5)=TIME-RV(15)
      RA(LV+6)=RV(6)/RV(5)
      RA(LV+7)=RV(7)/RV(5)-RV(6) 2
      IF(RV(7).GT.0.)GO TO 00045
GO TO 00046
00045 CONTINUE
      RA(LV+7)=SORT(RV(7))
GO TO 00047
00046 CONTINUE
      RA(LV+7)=0.
00047 CONTINUE
      WRITE(NO1,00048)IV(1),IV(2),RV(3),IV(4),RV(6),RV(7)
00048 FORMAT(/11X,17HCONTAINER STORAGE /11X,20HCURRENTLY IN STORAGE
X      ,111 /11X,21HMAXIMUM NUMBER STORED,110,8H AT TIME,E11
X      .4 /11X,19HNUMBER OBSERVATIONS,112 /11X,4HMEAN,
X      E27.4 /11X,18HSTANDARD DEVIATION,E13.4)
00043 CONTINUE
      IF(ILP(5,3).EQ.1)GO TO 00049
GO TO 00050
00049 CONTINUE
      WRITE(NO1,00052)
00052 FORMAT(/11X,19HMATERIAL IN STORAGE)
      CALL WRMD(IP(7))
00050 CONTINUE
      IF(ILP(5,4).EQ.1)GO TO 00053
GO TO 00054
00053 CONTINUE
      WRITE(NO1,00056)
00056 FORMAT(/11X,18HMATERIAL IN HOLDUP)
      CALL WRMD(IP(8))
00054 CONTINUE
      IF(ILP(5,1).EQ.1.OR.ILP(5,2).EQ.1)GO TO 00057
GO TO 00058
00057 CONTINUE
      WRITE(NO1,00060)RV(15) TIME
00060 FORMAT(/11X,20HREPORT TIME INTERVAL,E11.4,9H TO,E11.4)

```

FORTRAN Function Programs

```

00058 CONTINUE
      ICTYPE=-5
      IPOINT=0
      GO TO 99992
99600 CONTINUE
      IF(IP(1).LT.1.OR.IP(1).GT.3)GO TO 00061
      GO TO 00062
00061 CONTINUE
      WRITE(N01,00064)
00064 FORMAT(1X,15H      ERROR      /1X,
X 44H401 STOM  ILLEGAL HOLDUP MODEL SPECIFICATION)
      CALL ERROR
00062 CONTINUE
      IF(IP(7).GT.0)GO TO 00065
      GO TO 00066
00065 CONTINUE
      CALL FRMD(IP(7))
00066 CONTINUE
      CALL CMB(IA(LP+7),0,0,0,0,2,0,0,0.)
      IF(IP(8).GT.0)GO TO 00068
      GO TO 00069
00068 CONTINUE
      CALL FRMD(IP(8))
00069 CONTINUE
      CALL CMB(IA(LP+8),0,0,0,0,1,0,0,0.)
      I=I+1
00071 IF(I) (1-(IA(LB+4))).LE.0)GO TO 00072
      GO TO 00073
00072 CONTINUE
      IA(LV-I+1)=IA(LI-I+1)
      GO TO 00071
00073 CONTINUE
      ICTYPE=-6
      IPOINT=0
      GO TO 99992
99700 CALL DUMP(IPVT)
      ICTYPE=-7
      IPOINT=0
      GO TO 99992
99800 CONTINUE
      IF(ITRACE.GE.1)GO TO 00074
      GO TO 00075
00074 CONTINUE
      WRITE(N01,00077)TIME,IA(LB+2)
00077 FORMAT(1X,E10.4,I4,2X,5HEVENT)
00075 CONTINUE
      IF(ITRACE.GE.2.AND.(ITRACE.EQ.ITRACE/2 2))GO TO 00078
      GO TO 00079
00078 CONTINUE
      WRITE(N02,00081)TIME,IA(LB+2)
00081 FORMAT(1X,E10.4,I4,2X,5HEVENT)

```

FORTRAN Function Programs

```

      CALL ICHECK
00079 CONTINUE
      WRITE(N01,00082)
00082 FORMAT(/1X,15H  ERROR /1X,
X  CALL 30H401 STOM  ILLEGAL EVENT NOTICE)
      CALL ERROR
99900 CONTINUE
      IF(ITRACE.GE.3)GO TO 00083
      GO TO 00084
00083 CONTINUE
      WRITE(N01,00086)TIME,1A(LB+2)
00086 FORMAT(1X,E10.4,14,2X,11HINTERACTION)
00084 CONTINUE
      IF(ITRACE.GE.4.AND.(ITRACE.EQ.ITRACE/2))GO TO 00087
      GO TO 00088
00087 CONTINUE
      WRITE(N02,00090)TIME,1A(LB+2)
00090 FORMAT(1X,E10.4,14,2X,11HINTERACTION)
      CALL ICHECK
00088 CONTINUE
      IF(1A(IPOINT+2).EQ.1)GO TO 00091
      GO TO 00092
00091 CONTINUE
      ASSIGN 00094 TO L00001
      GO TO 00001
00094 CONTINUE
      ASSIGN 00095 TO L00010
      GO TO 00010
00095 CONTINUE
      GO TO 00093
00092 IF(1A(IPOINT+2).EQ.5)GO TO 00096
      GO TO 00097
00096 CONTINUE
      ASSIGN 00098 TO L00002
      GO TO 00002
00098 CONTINUE
      GO TO 00093
00097 IF(1A(IPOINT+2).EQ.3)GO TO 00099
      GO TO 00100
00099 CONTINUE
      ASSIGN 00101 TO L00003
      GO TO 00003
00101 CONTINUE
      GO TO 00093
00100 IF(1A(IPOINT+2).EQ.4)GO TO 00102
      GO TO 00103
00102 CONTINUE
      ASSIGN 00104 TO L00004
      GO TO 00004
00104 CONTINUE
      GO TO 00093
00103 IF(1A(IPOINT+2).EQ.2)GO TO 00105

```

FORTRAN Function Programs

```

00105 GO TO 00106
      CONTINUE
      ASSIGN 00107 TO L00005
      GO TO 00005
00107 CONTINUE
      ASSIGN 00108 TO L00010
      GO TO 00010
00108 CONTINUE
      GO TO 00093
00106 CONTINUE
      WRITE(NC1,00109)
00109 FORMAT(/1X,15H      ERROR      /1X,
X      29H401 STOM  ILLEGAL INTERACTION)
      CALL ERROR
00093 CONTINUE
      ICTYPE=-9
      IPOINT=IPOINT
      GO TO 99992
C
00001 CONTINUE
      CALL NCBM(IA(IPOINT+3),NC,BM)
      IA(LV+1)=IV(1)+NC
      RA(LV+8)=RV(8)+BM
      IF(RV(8).GT.RV(9))GO TO 00110
      GO TO 00111
00110 CONTINUE
      BMX=RV(8)-RV(9)
      RA(LV+9)=RV(8)
      RA(LV+10)=TIME
      GO TO 00112
00111 CONTINUE
      BMX=0
00112 CONTINUE
      IF(BM.GT.0.AND.(LP(5,1).EQ.1))GO TO 00113
      GO TO 00114
00113 CONTINUE
      ASSIGN 00116 TO L00006
      GO TO 00006
00116 CONTINUE
00114 CONTINUE
      IF(NC.GT.0.AND.(LP(5,2).EQ.1))GO TO 00117
      GO TO 00118
00117 CONTINUE
      ASSIGN 00120 TO L00007
      GO TO 00007
00120 CONTINUE
00118 CONTINUE
      CALL TRNFRM(-1,1,IA(IPOINT+3),IP(7))
      CALL FRMD(IA(IPOINT+3))
      IF(IP(1).EQ.1.AND.BMX.GT.0.)GO TO 00121
      GO TO 00122
00121 CONTINUE

```

FORTRAN Function Programs

```

        ASSIGN 00124 TO L00008
        GO TO 00008
00124 CONTINUE
        GO TO 00123
00122 IF(IP(1).GE.2.AND.BM.GT.0)GO TO 00125
        GO TO 00126
00125 CONTINUE
        ASSIGN 00127 TO L00009
        GO TO 00009
00127 CONTINUE
00126 CONTINUE
00123 CONTINUE
        IA(IPPOINT+2)=-1
        IF(ITRACE.GE.3)GO TO 00128
        GO TO 00129
00128 CONTINUE
        WRITE(NO1,00131)TIME,IA(LB+2)
00131 FORMAT(1X,E10.4,14,2X,36H401 STOM MATERIAL PLACED IN STORAGE)
00129 CONTINUE
        IF(ITRACE.GE.4.AND.(ITRACE.EQ.ITRACE/2 2))GO TO 00132
        GO TO 00133
00132 CONTINUE
        WRITE(NO2,00135)TIME,IA(LB+2)
00135 FORMAT(1X,E10.4,14,2X,36H401 STOM MATERIAL PLACED IN STORAGE)
        CALL ICHECK
00133 CONTINUE
        GO TO L00001, (00094)
C
00002 CONTINUE
        IA(IPPOINT+2)=-5
        IA(IPPOINT+3)=IP(7)
        IF(ITRACE.GE.3)GO TO 00136
        GO TO 00137
00136 CONTINUE
        WRITE(NO1,00139)TIME,IA(LB+2)
00139 FORMAT(1X,E10.4,14,2X,27H401 STOM DIVERSION ATTEMPT)
00137 CONTINUE
        IF(ITRACE.GE.4.AND.(ITRACE.EQ.ITRACE/2 2))GO TO 00140
        GO TO 00141
00140 CONTINUE
        WRITE(NO2,00143)TIME,IA(LB+2)
00143 FORMAT(1X,E10.4,14,2X,27H401 STOM DIVERSION ATTEMPT)
        CALL ICHECK
00141 CONTINUE
        GO TO L00002, (00098)
C
00003 CONTINUE
        IA(IPPOINT+2)=-3
        IA(IPPOINT+3)=IP(7)
        IF(ITRACE.GE.3)GO TO 00144
        GO TO 00145
00144 CONTINUE

```

FORTRAN Function Programs

```

      WRITE(N01,00147)TIME,IA(LB+2)
00147 FORMAT(1X,E10.4,14,2X,
X 49H401 STOM REQUEST FOR AVAILABLE (STORED) MATERIAL)
00148 CONTINUE
      IF(1TRACE.GE.4.AND.(1TRACE.EQ.1TRACE/2 2))GO TO 00148
      GO TO 00149
00148 CONTINUE
      WRITE(N02,00151)TIME,IA(LB+2)
00151 FORMAT(1X,E10.4,14,2X,
X 49H401 STOM REQUEST FOR AVAILABLE (STORED) MATERIAL)
      CALL ICHECK
00149 CONTINUE
      GO TO L00003, (00101)
C
00004 CONTINUE
      IA(IPOINT+2)=-4
      IA(IPOINT+3)=IP(8)
      IF(1TRACE.GE.3)GO TO 00152
      GO TO 00153
00152 CONTINUE
      WRITE(N01,00155)TIME,IA(LB+2)
00155 FORMAT(1X,E10.4,14,2X,38H401 STOM REQUEST FOR HELD UP MATERIAL)
00153 CONTINUE
      IF(1TRACE.GE.4.AND.(1TRACE.EQ.1TRACE/2 2))GO TO 00156
      GO TO 00157
00156 CONTINUE
      WRITE(N02,00159)TIME,IA(LB+2)
00159 FORMAT(1X,E10.4,14,2X,38H401 STOM REQUEST FOR HELD UP MATERIAL)
      CALL ICHECK
00157 CONTINUE
      GO TO L00004, (00104)
C
00005 CONTINUE
      CALL NCBM(IA(IPOINT+3),NC,BM)
      IA(LV+1)=IV(1)-NC
      RA(LV+8)=RV(8)-BM
      IF(BM.GT.0.AND.ILP(5,1).EQ.1)GO TO 00160
      GO TO 00161
00160 CONTINUE
      ASSIGN 00163 TO L00006
      GO TO 00006
00163 CONTINUE
00161 CONTINUE
      IF(NC.GT.0.AND.ILP(5,2).EQ.1)GO TO 00164
      GO TO 00165
00164 CONTINUE
      ASSIGN 00167 TO L00007
      GO TO 00007
00167 CONTINUE
00165 CONTINUE
      IA(IPOINT+2)=-2
      IPOINT=0

```


FORTRAN Function Programs

```

      IF(1TRACE.EQ.3)GO TO 00168
      GO TO 00159
00168 CONTINUE
      WRITE(N01,00171)TIME,IA(LB+2)
00171 FORMAT(1X,E10.4,14,2X,39H401 STOM MATERIAL REMOVED FROM STORAGE
      X
00169 CONTINUE
      IF(1TRACE.EQ.4.AND.(1TRACE.EQ.1TRACE/2 2))GO TO 00172
      GO TO 00173
00172 CONTINUE
      WRITE(N02,00175)TIME,IA(LB+2)
00175 FORMAT(1X,E10.4,14,2X,39H401 STOM MATERIAL REMOVED FROM STORAGE
      X
      CALL ICHECK
00173 CONTINUE
      GO TO L00005, (00107)
C
00006 CONTINUE
      RA(LV+13)=RV(13)+RV(8) (TIME-RV(12))
      RA(LV+14)=RV(14)+RV(8) RV(8) (TIME-RV(12))
      IA(LV+11)=IV(11)+1
      RA(LV+12)=TIME
      GO TO L00006, (00116,00163)
C
00007 CONTINUE
      RA(LV+6)=RV(6)+IV(1) (TIME-RV(5))
      RA(LV+7)=RV(7)+IV(1) IV(1) (TIME-RV(5))
      IA(LV+4)=IV(4)+1
      IF(IV(1).GT.IV(2))GO TO 00176
      GO TO 00177
00176 CONTINUE
      IA(LV+2)=IV(1)
      RA(LV+3)=TIME
00177 CONTINUE
      RA(LV+5)=TIME
      GO TO L00007, (00120,00167)
C
00008 CONTINUE
      IF(BMX.GT.0.1)GO TO 00179
      GO TO 00180
00179 CONTINUE
      F=(RP(2)+RP(3) RV(8)) RP(4)/RV(8)
      ASSIGN 00182 TO L00011
      GO TO 00011
00182 CONTINUE
00180 CONTINUE
      GO TO L00008, (00124)
C
00009 CONTINUE
      IF(BMX.GT.0.)GO TO 00183
      GO TO 00184
00183 CONTINUE

```

FORTRAN Function Programs

```

      CALL TRNFRM(0,1.,IP(8),IP(7))
      GO TO 00185
00184 CONTINUE
      CALL NCBM(IP(8),NC,BMH)
      IF(IP(1).EQ.2)GO TO 00186
      GO TO 00187
00186 CONTINUE
      IF(BMH.GT.0.)GO TO 00189
      GO TO 00190
00189 CONTINUE
      F=((BMH-RP(2) RP(4)) (RV(8)/RV(9))+RP(2) RP(4))/BMH
      CALL TRNFRM(0,F,IP(8),IP(7))
00190 CONTINUE
      F=(R(2)+RP(3) RV(8)) RP(4)/RV(8)
      ASSIGN 00192 TO L00011
      GO TO 00011
00192 CONTINUE
      GO TO 00188
00187 IF(IP(1).EQ.3)GO TO 00193
      GO TO 00194
00193 CONTINUE
      IF(BMH.GT.0)GO TO 00195
      GO TO 00196
00195 CONTINUE
      CALL TRNFRM(0,1.,IP(8),IP(7))
00196 CONTINUE
      F=RP(3) RP(4)/RV(8)
      ASSIGN 00198 TO L00011
      GO TO 00011
00198 CONTINUE
00194 CONTINUE
00188 CONTINUE
00185 CONTINUE
      GO TO L00009, (00127)
C
00010 CONTINUE
      IF(IP(6).GT.0)GO TO 00199
      GO TO 00200
00199 CONTINUE
      CALL PCOM(ILP(6,1),9,9,IP(7),0,0,0,0,0,0)
      I=1-(1)
      I=1+I
00202 IF(1) (1-(1A(IP(6)+3))),LE.0)GO TO 00203
      GO TO 00204
00203 CONTINUE
      MTO=ILP(6,1)
      ICTYPE=9
      1A(LP)=1
      GO TO 99992
00205 CONTINUE
      IF(1A(IPOINT).NE.-9)GO TO 00206
      GO TO 00207

```

FORTRAN Function Programs

```

00206 CONTINUE
      WRITE(NO1,00209)
00209 FORMAT(/IX,15H      ERROR      /IX,
X      38H401 STOM  RESPONSE TO CONDITION NOTICE)
      CALL ERROR
00207 CONTINUE
      GO TO 00202
00204 CONTINUE
      CALL FSTORE(IPPOINT)
00200 CONTINUE
      GO TO L00010, (00095,00106)
C
00011 CONTINUE
      IF(F.GT.1.)GO TO 00210
      GO TO 00211
00210 CONTINUE
      CALL TRNFRM(0,1.,IP(7),IP(8))
      GO TO 00212
00211 CONTINUE
      CALL TRNFRM(0,F,IP(7),IP(8))
00212 CONTINUE
      GO TO L00011, (00182,00192,00198)
00026 !00000=IA(LP)
      GO TO 00205
      END
C 501 INDA 3/16/78
C INDEPENDENT ACTION
C LV+0 1 RESPONSE POINTER
C      1 1 DURATION TIME DISTRIBUTION TYPE
C      2 4 DURATION TIME STATISTICS
C      3 4 STIMULUS SPECIFICATIONS (REPEAT FOR EACH TYPE)
C          MONITOR FUNCTION
C          STIMULUS TYPE
C          DELAY TIME DISTRIBUTION TYPE
C          DELAY TIME STATISTICS
C          DURATION TIME DISTRIBUTION TYPE
C          DURATION TIME STATISTICS
C          INTENSITY DISTRIBUTION TYPE
C          INTENSITY STATISTICS
C      4 2 FUNCTIONS NOTIFIED AT CONDITION TIMES
C LV+0 1 ENTRY COUNT
C L+0 0
      SUBROUTINE B501
      DIMENSION IPVT(6)
      DIMENSION IA(1)
      COMMON RA(1)
      EQUIVALENCE (IA(1),RA(1))
      COMMON /SYSTEM/MFROM,MT0,ICTYPE,IPPOINT,
      1IDUMP,ISECA,ITRACE,LA,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
      1LEC,LFPB,LFFE,L1,L2,LLB,LFE,LMS,LNE,LP,LV,MINNFL,
      1MNSF,NFE,NFL,N11,N12,NO1,NO2,NREP,NRUN,NSE,TIME,TSTOP
      DATA IPVT/1,1,4,4,2,1/

```

FORTRAN Function Programs

```

      IA(LV)=IA(LV)+1
      IF(ICTYPE.GE.0)GO TO 99991
      GO TO 00026
99991 GO TO (99100,99200,99300,99400,99500,99600,99700,99800,99900)
      X, ICTYPE
99992 RETURN
99100 CALL CREATE(501,5,1,IPVT)
      ICTYPE=-1
      IPOINT=0
      GO TO 99992
99200 CALL DESTRO(IPVT)
      ICTYPE=-2
      IPOINT=0
      GO TO 99992
99300 CALL SET(1,IPVT,INTD,REALD)
      ICTYPE=-3
      IPOINT=0
      GO TO 99992
99400 CONTINUE
      WRITE(NO1,00027)
00027 FORMAT(/1X,15H      ERROR      /1X,
      X      43H501 INDA  ATTEMPT TO SET INITIAL CONDITIONS)
      CALL ERROR
99500 CONTINUE
      WRITE(NO1,00028)IA(LB+2)
00028 FORMAT(/,9H FUNCTION,14,1X,8H501 INDA/11X,9HNO REPORT)
      ICTYPE=-5
      IPOINT=0
      GO TO 99992
99600 CONTINUE
      I=1-(1)
00029 I=I+1
      IF(1) (1-(IA(LB+4))),LE.0)GO TO 00030
      GO TO 00031
00030 CONTINUE
      IA(LV-1+1)=IA(LI-1+1)
      GO TO 00029
00031 CONTINUE
      ICTYPE=-6
      IPOINT=0
      GO TO 99992
99700 CALL DUMP(IPVT)
      ICTYPE=-7
      IPOINT=0
      GO TO 99992
99800 CONTINUE
      IF(TRACE.GE.1)GO TO 00032
      GO TO 00033
00032 CONTINUE
      WRITE(NO1,00035)TIME,IA(LB+2)
00035 FORMAT(1X,E10.4,14,2X,5HEVENT)
00033 CONTINUE

```

FORTRAN Function Programs

```

      IF (ITRACE.GE.2.AND.((ITRACE.EQ.ITRACE/2 2))GO TO 00036
      GO TO 00037
00036 CONTINUE
      WRITE(N02,00039)TIME,1A(LB+2)
00039 FORMAT(1X,E10.4,14,2X,5HEVENT)
      CALL ICHECK
00037 CONTINUE
      IF (IA(IPINT).GT.0)GO TO 00040
      GO TO 00041
00040 CONTINUE
      CALL EVDP5(1,2,D)
      IF (IP(4).GT.0)GO TO 00043
      GO TO 00044
00043 CONTINUE
      CALL SCHED(TIME+D,1A(LB+2),0,0)
00044 CONTINUE
      I=IA(IPINT)
      L=IA(IPINT+1)
      INXT=I+2+NSTATS(IFIX(RAC(L,1+3)))
      IF (IFIX(RAC(L,INXT+2)).GT.0.AND.INXT.LT.1A(L+3))GO TO 00046
      GO TO 00047
00046 CONTINUE
      CALL SAMPL(L,INXT+1,TD)
      CALL SCHED(TIME+D+TD,IFIX(RAC(L,INXT+2)),INXT,L)
00047 CONTINUE
      IF (IP(3).GT.0)GO TO 00049
      GO TO 00050
00049 CONTINUE
      ASSIGN 00052 TO L00001
      GO TO 00001
00052 CONTINUE
00050 CONTINUE
      GO TO 00042
00041 IF (IA(IPINT).EQ.0)GO TO 00053
      GO TO 00054
00053 CONTINUE
      IF (IP(4).EQ.0)GO TO 00055
      GO TO 00056
00055 CONTINUE
      WRITE(N01,00058)
00058 FORMAT(1X,15H      ERROR      /1X,24H501 (NDA CONDITION LIST)
      CALL ERROR
00056 CONTINUE
      CALL PCOM(ILP(4,1),9,9,0,0,0,0,0,0,0)
      I=1-(1)
      I=I+1
      IF ((1)-(1-(IA(IP(4)+3))).LE.0)GO TO 00060
      GO TO 00061
00060 CONTINUE
      MFROM=1A(LB+2)
      MTO=ILP(4,1)
      ICTYPE=9

```

FORTRAN Function Programs

```

      IA(IPOINT+2)=9
      IA(LP)=1
      GO TO 99992
00062 CONTINUE
      IF(IA(IPOINT+2).NE.-9)GO TO 00063
00063 CONTINUE
      WRITE(NO1,00066)
00066 FORMAT(/1X,15H      ERROR      /1X,
X      38H501 INDA  RESPONSE TO CONDITION NOTICE)
      CALL ERROR
00064 CONTINUE
      GO TO 00059
00061 CONTINUE
      CALL FSTORE(IPOINT)
00054 CONTINUE
00042 CONTINUE
      ICTYPE=-8
      IPOINT=0
      GO TO 99992
99900 CONTINUE
      IF(ITRACE.GE.3)GO TO 00067
      GO TO 00068
00067 CONTINUE
      WRITE(NO1,00070)TIME,IA(LB+2)
00070 FORMAT(1X,E10.4,14,2X,11HINTERACTION)
00068 CONTINUE
      IF(ITRACE.GE.4.AND.(ITRACE.EQ.ITRACE/2 2))GO TO 00071
      GO TO 00072
00071 CONTINUE
      WRITE(NO2,00074)TIME,IA(LB+2)
00074 FORMAT(1X,E10.4,14,2X,11HINTERACTION)
      CALL ICHECK
00072 CONTINUE
      WRITE(NO1,00075)
00075 FORMAT(/1X,15H      ERROR      /1X,
X      29H501 INDA  ILLEGAL INTERACTION)
      CALL ERROR
      ICTYPE=-9
      IPOINT=0
      GO TO 99992
C
00001 CONTINUE
      I=1
      CALL PCOM(FIX(RLP(3,I)),9,7,1P(3),I+1,0,0,0,0,0)
00076 IF(1.LE.IA(IP(3)+3))GO TO 00077
      GO TO 00078
00077 CONTINUE
      MFROM=IA(LB+2)
      MTO=RLP(3,I)
      ICTYPE=9
      IA(IPOINT+2)=7

```

FORTRAN Function Programs

```

      IA(IPOINT+4)=I+1
      IA(LP)=2
      GO TO 99992
00079 CONTINUE
      IF(IA(IPOINT+2).NE.-7)GO TO 00080
      GO TO 00081
00080 CONTINUE
      WRITE(N01,00083)
00083 FORMAT(/1X,15H      ERROR      /1X,
X      35H501 INDA  STIMULI DATA NOT ACCEPTED)
00081 CALL ERROR
      CONTINUE
      I=I+2
      I=I+NSTATS(IFIX(RLP(3,I)))
      I=I+NSTATS(IFIX(RLP(3,I)))
      I=I+NSTATS(IFIX(RLP(3,I)))
      GO TO 00075
00078 CONTINUE
      CALL FSTORE(IPOINT)
      GO TO L00001, (00052)
00026 I00000=IA(LP)
      GO TO (00062,00079), I00000
      END
C 502 MAQA 3/16/78
C ADVERSARY MATERIAL ACQUISITION ACTION
C LP+0 1 RESPONSE POINTER
C 1 1 DURATION TIME DISTRIBUTION TYPE
C 2 4 DURATION TIME STATISTICS
C 3 4 STIMULUS SPECIFICATIONS (REPEAT FOR EACH TYPE)
C STIMULUS MONITOR FUNCTION
C STIMULUS TYPE
C DELAY TIME DISTRIBUTION TYPE
C DELAY TIME STATISTICS
C DURATION TIME DISTRIBUTION TYPE
C DURATION TIME STATISTICS
C INTENSITY DISTRIBUTION TYPE
C INTENSITY STATISTICS
C 4 2 TARGET FUNCTIONS
C 6 2 FUNCTIONS NOTIFIED AT CONDITION TIMES
C 6 1 -DIVERTED MATERIAL
C LV+0 1 ENTRY COUNT
C LI+0 0
      SUBROUTINE 6502
      DIMENSION IPVT(8)
      DIMENSION IA(1)
      COMMON RA(1)
      EQUIVALENCE (IA(1),RA(1))
      COMMON /SYSTEM/MFROM,MTG,ICTYPE,IPOINT,
1)DUMP,ISECA,ITRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
1)EC,LFFB,LFFE,L1,LLE,LFB,LFE,LMS,LNE,LPLV,MINNFI,
1)MNSF,NFE,NFL,N11,N12,N01,N02,NREP,NRUN,NSE,TIME,TSTOP
      DATA IPVT/1,1,4,4,2,2,1,1/

```

FORTRAN Function Programs

```

      IA(LV)=IA(LV)+1
      IF(ITYPE.GE.0)GO TO 99991
      GO TO 00025
99991 GO TO (99100,99200,99300,99400,99500,99600,99700,99800,99900)
      X, ICTYPE
99992 RETURN
      DATA IX,U/1,0./
99100 CALL CREATE(502,7,1,IPVT)
      ICTYPE=-1
      IPOINT=0
      GO TO 99992
99200 CONTINUE
      IF(IP(6).GT.0)GO TO 00027
      GO TO 00028
00027 CONTINUE
      CALL FRMD(IP(6))
00028 CONTINUE
      CALL DESTRO(IPVT)
      ICTYPE=-2
      IPOIN=0
      GO TO 99992
99300 CALL SET(,IPVT,INTD,REALD)
      ICTYPE=-3
      IPOINT=0
      GO TO 99992
99400 CONTINUE
      WRITE(NO1,00030)
00030 FORMAT(/1X,15H      ERROR      /1X,
      X 43H502 MAQA  ATTEMPT TO SET INITIAL CONDITIONS)
      CALL ERROR
99500 CONTINUE
      WRITE(NO1,00031)IA(LB+2)
00031 FORMAT(/,9H FUNCTION,14,1X,8H502 MAQA /11X,17HDIVERTED MATERIAL
      X
      CALL WRMD(IP(6))
      ICTYPE=-5
      IPOINT=0
      GO TO 99992
99600 CONTINUE
      IF(IP(6).GT.0)GO TO 00032
      GO TO 00033
00032 CONTINUE
      CALL FRMD(IP(6))
00033 CONTINUE
      CALL CMS(IA(LP+6),0,0,0,0,2,0,0,0.)
      I=1-11
00035 IF(1) (1-(IA(LB+4))).LE.0)GO TO 00036
      GO TO 00037
00036 CONTINUE
      IA(LV-1+1)=IA(LI-1+1)
      GO TO 00035

```


FORTRAN Function Programs

```

00037 CONTINUE
      ICTYPE=-6
      IPOINT=0
      GO TO 99992
99700 CALL DUMP(IPVT)
      ICTYPE=-7
      IPOINT=0
      GO TO 99992
99800 CONTINUE
      IF(ITRACE.GE.1)GO TO 00038
      GO TO 00039
00038 CONTINUE
      WRITE(NO1,00041)TIME,IA(LB+2)
00041 FORMAT(1X,E10.4,14,2X,SHEVENT)
00039 CONTINUE
      IF(ITRACE.GE.2.AND.(ITRACE.EQ.ITRACE/2))GO TO 00042
      GO TO 00043
00042 CONTINUE
      WRITE(NO2,00045)TIME,IA(LB+2)
00045 FORMAT(1X,E10.4,14,2X,SHEVENT)
      CALL ICHECK
00043 CONTINUE
      IF(IA(IPOINT).GT.0)GO TO 00046
      GO TO 00047
00046 CONTINUE
      CALL EVDRS(1,2,D)
      IF(IP(5).GT.0)GO TO 00049
      GO TO 00050
00049 CONTINUE
      CALL SCHED(TIME+D,IA(LB+2),0,0)
00050 CONTINUE
      I=IA(IPOINT)
      J=IA(IPOINT+1)
      INXT=I+2+NSTATS(IFIX(RAC(J,I+3)))
      IF(IFIX(RAC(J,INXT+2)).GT.0.AND.INXT.LT.IA(J+3))GO TO 00052
      GO TO 00053
00052 CONTINUE
      CALL SAMPL(J,INXT+1,TD)
      CALL SCHED(TIME+D+TD,IFIX(RAC(J,INXT+2)),INXT,J)
00053 CONTINUE
      IF(IP(3).GT.0)GO TO 00055
      GO TO 00056
00055 CONTINUE
      ASSIGN 00058 TO L00001
      GO TO 00001
00058 CONTINUE
00056 CONTINUE
      IF(IP(4).GT.0)GO TO 00059
      GO TO 00060
00059 CONTINUE
      ASSIGN 00062 TO L00002
      GO TO 00002

```

FORTRAN Function Programs

```

00062 CONTINUE
00060 CONTINUE
GO TO 00048
00047 IF(IA(IPPOINT).EQ.0)GO TO 00063
GO TO 00064
00063 CONTINUE
IF(IP(5).EQ.0)GO TO 00065
GO TO 00066
00065 CONTINUE
WRITE(N01,00068)
00068 FORMAT(/1X,15H      ERROR      /1X,26H502 MAQA  O?PORTUNITY LIST)
CALL ERROR
00066 CONTINUE
CALL PCOM(ILP(5,1),9,9,0,0,0,0,0,0,0)
I=1-(1)
I=I+1
00069 IF(I) (1-(IA(IP(5)+3))).LE.0)GO TO 00070
GO TO 00071
00070 CONTINUE
MTO=ILP(5,1)
IA(IPPOINT+2)=9
IA(LP)=1
GO TO 99992
00072 CONTINUE
IF(IA(IPPOINT+2).NE.-9)GO TO 00073
GO TO 00074
00073 CONTINUE
WRITE(N01,00076)
00076 FORMAT(/1X,15H      ERROR      /1X,
X 40H502 MAQA  RESPONSE TO OPPORTUNITY NOTICE)
CALL ERROR
00074 CONTINUE
GO TO 00069
00071 CONTINUE
CALL FSTORE(IPPOINT)
00064 CONTINUE
00048 CONTINUE
ICTYPE=-8
IPPOINT=0
GO TO 99992
99900 CONTINUE
IF(ITRACE.GE.3)GO TO 00077
GO TO 00078
00077 CONTINUE
WRITE(N01,00080)TIME,IA(LB+2)
00080 FORMAT(1X,E10.4,14,2X,11HINTERACTION)
00078 CONTINUE
IF(ITRACE.GE.4.AND.(ITRACE.EQ.(ITRACE/2 2)))GO TO 00081
GO TO 00082
00081 CONTINUE
WRITE(N02,00084)TIME,IA(LB+2)
00084 FORMAT(1X,E10.4,14,2X,11HINTERACTION)

```

FORTRAN Function Programs

```

      CALL ICHECK
00082 CONTINUE
      IF (IA(IPOINT+2).EQ.3) GO TO 00085
      GO TO 00086
00085 CONTINUE
      IA(IPOINT+2)=-3
      IA(IPOINT+3)=IP(6)
      GO TO 00087
00086 CONTINUE
      WRITE(NO1,00088)
00088 FORMAT(/1X,15H      ERROR      /1X,
X      29H502 MAQA  ILLEGAL INTERACTION)
      CALL ERROR
00087 CONTINUE
      ICTYPE=9
      IPOINT=IPOINT
      GO TO 99992
C
00001 CONTINUE
      I=1
      CALL PCOM(FIX(RLP(3,1)),9,7,IP(3),I+1,0,0,0,0,0)
00089 IF (I.LE. IA(IP(3)+3)) GO TO 00090
      GO TO 00091
00090 CONTINUE
      MTO=RLP(3,1)
      ICTYPE=9
      IA(IPOINT+2)=7
      IA(IPOINT+4)=I+1
      IA(LP)=2
      GO TO 99992
00092 CONTINUE
      IF (IA(IPOINT+2).NE.-7) GO TO 00093
      GO TO 00094
00093 CONTINUE
      WRITE(NO1,00096)
00096 FORMAT(/1X,15H      ERROR      /1X,
X      35H502 MAQA  STIMULI DATA NOT ACCEPTED)
      CALL ERROR
00094 CONTINUE
      I=I+2
      I=I+NSTATS(FIX(RLP(3,1)))+1
      I=I+NSTATS(FIX(RLP(3,1)))+1
      I=I+NSTATS(FIX(RLP(3,1)))+1
      GO TO 00089
00091 CONTINUE
      CALL FSTORE(IPOINT)
      GO TO L00001, (00058)
C
00002 CONTINUE
      II=1-(1)
      II=II+1
00097 IF ((1) (1[-(IA(IP(4)+31)).LE.0]) GO TO 00098

```

FORTRAN Function Programs

```

00098 GO TO 00099
00099 CONTINUE
      CALL PCON(ILP(4,1),9,5,0,0,0,0,0,0)
      IA(LP)=3
      GO TO 99992
00100 CONTINUE
      IF(IA(IPOINT+2).NE.-5)GO TO 00101
      GO TO 00102
00101 CONTINUE
      WRITE(N01,00104)
00104 FORMAT(/X,15H      ERROR      /1X,
X      34H502 MAQA TARGET FUNCTION RESPONSE)
00102 CONTINUE
      IF(IA(IPOINT+3).GT.0)GO TO 00105
      GO TO 00106
00105 CONTINUE
      LSM=IA(IPOINT+3)
      ASSIGN 00108 TO L00003
      GO TO 00003
00108 CONTINUE
      GO TO 00107
00106 IF(IA(IPOINT+4).GT.0)GO TO 00109
      GO TO 00110
00109 CONTINUE
      CALL URN(IX,U)
      N=U*FLOAT(IA(IPOINT+5))*.9999+1
      L=IA(IPOINT+4)
      M=1
00111 IF(M.LT.N)GO TO 00112
      GO TO 00113
00112 CONTINUE
      L=IA(L+8)
      IF(L.LE.0)GO TO 00114
      GO TO 00115
00114 CONTINUE
      WRITE(N01,00117)
00117 FORMAT(/1X,15H      ERROR      /1X,
X      30H502 MAQA MESSAGE BLOCK LINKAGE)
      CALL ERROR
00115 CONTINUE
      M=M+1
      GO TO 00111
00113 CONTINUE
      LSM=IA(L+3)
      ASSIGN 00118 TO L00003
      GO TO 00003
00118 CONTINUE
00110 CONTINUE
00107 CONTINUE
      CALL FSTORE(IPOINT)
      GO TO 00097

```

FORTRAN Function Programs

```

00099 CONTINUE
      GO TO L00002, (00062)
C
00003 CONTINUE
      CALL NCBM(LSM,NC,BM)
      I=FIX(RAC(J,3))
      IF(I.EQ.1)GO TO 00119
      GO TO 00120
00119 CONTINUE
      IF(BM.GT.0.)GO TO 00122
      GO TO 00123
00122 CONTINUE
      ASSIGN 00125 TO L00004
      GO TO 00004
00125 CONTINUE
00123 CONTINUE
      GO TO 00121
00120 IF(I.EQ.2)GO TO 00126
      GO TO 00127
00126 CONTINUE
      IF(BM.GT.0.)GO TO 00128
      GO TO 00129
00128 CONTINUE
      ASSIGN 00131 TO L00005
      GO TO 00005
00131 CONTINUE
00129 CONTINUE
      GO TO 00121
00127 IF(I.EQ.3)GO TO 00132
      GO TO 00133
00132 CONTINUE
      IF(NC.GT.0)GO TO 00134
      GO TO 00135
00134 CONTINUE
      ASSIGN 00137 TO L00006
      GO TO 00006
00137 CONTINUE
00135 CONTINUE
      GO TO 00121
00133 IF(I.EQ.4)GO TO 00138
      GO TO 00139
00138 CONTINUE
      IF(NC.GT.0)GO TO 00140
      GO TO 00141
00140 CONTINUE
      ASSIGN 00143 TO L00007
      GO TO 00007
00143 CONTINUE
00141 CONTINUE
      GO TO 00121
00139 IF(I.EQ.5)GO TO 00144
      GO TO 00145

```

FORTRAN Function Programs

```

00144 CONTINUE
      IF(NC.GT.0)GO TO 00146
      GO TO 00147
00146 CONTINUE
      ASSIGN 00149 TO L00008
      GO TO 00008
00149 CONTINUE
00147 CONTINUE
      GO TO 00121
00145 CONTINUE
      WRITE(N01,00150)
00150 FORMAT(/1X,15H      ERROR      /1X,
X      51H502 MAQA  ILLEGAL MATERIAL DIVERSION OBJECTIVE TYPE)
      CALL ERROR
00121 CONTINUE
      GO TO L00003, (00108,00118)
C
00004 CONTINUE
      ASSIGN 00151 TO L00009
      GO TO 00009
00151 CONTINUE
      L=IA(1SMB+5)
00152 IF(L.NE.0.AND.1A(L+7).NE.?FIX(RAC(J,4)))GO TO 00153
      GO TO 00154
00153 CONTINUE
      L=1A(L+4)
      GO TO 00152
00154 CONTINUE
      IF(L.EQ.0)GO TO 00155
      GO TO 00156
00155 CONTINUE
      LB1=0
      GO TO 00157
00156 CONTINUE
      IF(1A(L+8).GT.0)GO TO 00158
      GO TO 00159
00158 CONTINUE
      LB1=L+8
      GO TO 00160
00159 CONTINUE
      LB1=0
00160 CONTINUE
00157 CONTINUE
      IF(LB1.GT.0)GO TO 00161
      GO TO 00162
00161 CONTINUE
      CALL CHB(LDM,0,0,0,0,1,0,0,0.)
      CALL CHB(1A(LDM+5),LDM,0,0,0,0,IFIX(RAC(J,4)),0,0.)
      N=1UNIF(IFIX(RAC(J,5)),IFIX(RAC(J,6)))
      IF(N.GT.1A(LB1))GO TO 00164
      GO TO 00165
00164 CONTINUE

```

FORTRAN Function Programs

```

      N=IA(LB1)
      GO TO 00166
00165 IF(N.LT.0)GO TO 00167
      GO TO 00168
00167 CONTINUE
      N=0
00168 CONTINUE
00166 CONTINUE
      IA(IA(LDM+5)+8)=N
      IA(LB1)=IA(LB1)-N
      CALL TRNFRM(0,1,LDM,IP(6))
      CALL FSTORE(LDM)
00162 CONTINUE
      GO TO L00004, (00125)
C
00005 CONTINUE
      ASSIGN 00169 TO L00009
      GO TO 00009
00169 CONTINUE
      IF(IFIX(RAC(J,4)).EQ.0)GO TO 00170
      GO TO 00171
00170 CONTINUE
      A=RNORM(IFIX(RAC(J,7)),IFIX(RAC(J,8)))
      IF(A.LT.0.)GO TO 00173
      GO TO 00174
00173 CONTINUE
      A=0.
00174 CONTINUE
      CALL TRNFRM(0,A/BH,LSM,IP(6))
      GO TO 00172
00171 CONTINUE
      L=IA(LSMB+5)
00176 IF(L.NE.0.AND.1A(L+7).NE.IFIX(RAC(J,4)))GO TO 00177
      GO TO 00178
00177 CONTINUE
      L=1A(L+4)
      GO TO 00176
00178 CONTINUE
      IF(L.NE.0)GO TO 00179
      GO TO 00180
00179 CONTINUE
      A=RNORM(RAC(J,7),RAC(J,8))
      IF(A.LT.0.)GO TO 00182
      GO TO 00183
00182 CONTINUE
      A=0.
      GO TO 00184
00183 IF(A.GE.RA(L+9))GO TO 00185
      GO TO 00186
00185 CONTINUE
      A=RA(L+9)
00186 CONTINUE

```

FORTRAN Function Programs

```

00184 CONTINUE
      CALL TRNFRM(0,A/RA(L+9),LSM,1P(6))
00180 CONTINUE
00172 CONTINUE
      GO TO L00005, (00131)
C
00006 CONTINUE
      ASSIGN 00187 TO L00010
      GO TO 00010
00187 CONTINUE
      LSMSAV=LSM
      LSM=LSMC
      ASSIGN 00188 TO L00004
      GO TO 00004
00188 CONTINUE
      LSM=LSMSAV
      GO TO L00006, (00137)
C
00007 CONTINUE
      ASSIGN 00189 TO L00010
      GO TO 00010
00189 CONTINUE
      LSMSAV=LSM
      LSM=LSMC
      ASSIGN 00190 TO L00005
      GO TO 00005
00190 CONTINUE
      LSM=LSMSAV
      GO TO L00007, (00143)
C
00008 CONTINUE
      N=IUNIF(IFIX(RAC(J,9)),IFIX(RAC(J,10)))
      IF(N.LT.0)GO TO 00191
      GO TO 00192
00191 CONTINUE
      N=0
      GO TO 00193
00192 IF(N.GT.NC)GO TO 00194
      GO TO 00195
00194 CONTINUE
      N=NC
00195 CONTINUE
00193 CONTINUE
      CALL TRNFRM(N,0,LSM,1P(6))
      GO TO L00008, (00149)
C
00009 CONTINUE
      IF(IA(LSM+6).EQ.1)GO TO 00196
      GO TO 00137
00196 CONTINUE
      LSMB=LSM
      GO TO 00198

```


FORTRAN Function Programs

```

00197 CONTINUE
      L=IA(LSM+5)
      IF(IA(L+6).EQ.1)GO TO 00199
      GO TO 00200
00199 CONTINUE
      LSMB=L
      GO TO 00201
00200 CONTINUE
      WRITE(NO1,00202)
00202 FORMAT(/1X,15H      ERROR      /1X,
X      30H502 MAOA LOCATING BULK AT LSM)
      CALL ERROR
00201 CONTINUE
00198 CONTINUE
      GO TO L00009, (00151,00169)
C
00010 CONTINUE
      L=IA(LSM+5)
      IF(IA(L+6).EQ.1)GO TO 00203
      GO TO 00204
00203 CONTINUE
      L=IA(L+4)
00204 CONTINUE
      IF(IA(L+6).NE.2)GO TO 00206
      GO TO 00207
00206 CONTINUE
      WRITE(NO1,00209)
00209 FORMAT(/1X,15H      ERROR      /1X,
X      52H502 MAOA LOCATING CONTAINER MATERIAL DATA STRUCTURE)
      CALL ERROR
00207 CONTINUE
      N=0
      K=L
00210 IF(K.NE.0)GO TO 00211
      GO TO 00212
00211 CONTINUE
      K=IA(K+4)
      N=N+1
      GO TO 00210
00212 CONTINUE
      CALL URN(IX,U)
      I=U FLOAT(N) .9999+1
      N=1
      K=L
00213 IF(N.LT.1)GO TO 00214
      GO TO 00215
00214 CONTINUE
      K=IA(K+4)
      N=N+1
      GO TO 00213
00215 CONTINUE
      LSMC=K

```

FORTRAN Function Programs

```

GO TO L00010, (00187,00189)
00026 100000=1A(LP)
GO TO (00072,00092,00100), 100000
END
503 DEFA 3/16/78
DEPENDENT ACTION
LP#0 1 1 RESPONSE POINTER
1 1 DURATION TIME DISTRIBUTION TYPE
2 4 DURATION TIME STATISTICS
3 4 STIMULUS SPECIFICATIONS (REPEAT FOR EACH TYPE)
STIMULUS MONITOR FUNCTION
STIMULUS TYPE
DELAY TIME DISTRIBUTION TYPE
DELAY TIME STATISTICS
DURATION TIME DISTRIBUTION TYPE
DURATION TIME STATISTICS
INTENSITY DISTRIBUTION TYPE
INTENSITY STATISTICS
4 1 MODE: 1 = CONDITIONAL STARTING OF ACTION
2 = CONDITIONAL COMPLETION OF ACTION
5 2 REFERENCE TIME
6 2 FUNCTIONS NOTIFIED AT COMPLETION TIMES
7 1 -REPORT DATA
LV#0 1 ENTRY COUNT
1 3 TIME T1 ACTION STARTED
2 3 DURATION OF ACTION
3 3 TIME DELAY OF NEXT ACTION
4 1 NEXT ACTION FUNCTION
5 1 NEXT ACTION FUNCTION INDEX INXT
6 1 - ACTION SEQUENCE LIST
7 3 TIME TO THE LAST CONDITION NOTICE WAS RECEIVED
LI#0 0
1 -2.
2 -1.
3 -1.
4 0
5 0
6 0
7 -1
SUBROUTINE B503
DIMENSION IPVT(16)
DIMENSION A(1)
COMMON RA(1)
EQUIVALENCE (IA(1),RA(1))
COMMON /SYSTEM/MFROM, MTO, ICTYPE, IPOINT,
I1DUMP, ISECA, ITRACE, L8, LBGA, LBLIST, LBPT, LBUF, LCSB, LE,
ILEC, LFFB, LFFE, LI, LLE, LLFB, LLFE, LMS, LNE, LP, LV, MINNFI,
IMNSE, NFE, NFL, NI1, NI2, NO1, NO2, NREP, NRUN, NSE, TIME, TSTOP
DATA IPVT/1,1,4,4,1,3,2,1,1,3,3,3,1,1,1,3/
IA(LV)=IA(LV)+1
IF(ICTYPE.GE.O)GO TO 99991
GO TO 00026

```

FORTRAN Function Programs

```

99991 GO TO (99100,99200,99300,99400,99500,99600,99700,99800,99900)
X, ICTYPE
99992 RETURN
99100 CALL CREATE(503,8,8,IPVT)
      RA(LI+1)=-2.
      RA(LI+2)=-1.
      RA(LI+3)=-1.
      RA(LI+7)=-1.
      ICTYPE=-1
      IPOINT=0
      GO TO 99992
99200 CONTINUE
      IF(IP(7).GT.0)GO TO 00027
      GO TO 00028
00027 CONTINUE
      CALL FSTORE(IP(7))
00028 CONTINUE
      CALL DESTRO(IPVT)
      ICTYPE=-2
      IPOINT=0
      GO TO 99992
99300 CALL SET(1,IPVT,INTD,REALD)
      ICTYPE=-3
      IPOINT=0
      GO TO 99992
99400 CONTINUE
      WRITE(N01,00030)
00030 FORMAT(/1X,15H      ERROR      /1X,
X      43H503 DEPA  ATTEMPT TO SET INITIAL CONDITIONS)
      CALL ERROR
99500 CONTINUE
      ASSIGN 00031 TO L00004
      GO TO 00004
00031 CONTINUE
      ICTYPE=-5
      IPOINT=0
      GO TO 99992
99600 CONTINUE
      IF(IP(7).EQ.0)GO TO 00032
      GO TO 00033
00032 CONTINUE
      CALL ASTORE(IA(LP+7),8)
00033 CONTINUE
      IF(NRUN.EQ.1)GO TO 00035
      GO TO 00036
00035 CONTINUE
      IA(LCS(IP(7),1))=0
      RA(LCS(IP(7),2))=0.
      RA(LCS(IP(7),3))=0.
      RA(LCS(IP(7),4))=0.
      RA(LCS(IP(7),5))=1.0E+50
00036 CONTINUE

```

FORTRAN Function Programs

```

I=1-(I)
00038 I=I+1
IF(I) (1-(I*(LB+4))).LE.0)GO TO 00039
GO TO 00040
00039 CONTINUE
GO I*(LV-1+I)=I*(L1-1+I)
GO TO 00038
00040 CONTINUE
ICTYPE=6
IPPOINT=0
GO TO 99992
99700 CALL DUMP(IPVT)
ICTYPE=7
IPPOINT=0
GO TO 99992
99800 CONTINUE
IF(TRACE.GE.1)GO TO 00041
GO TO 00042
00041 CONTINUE
WRITE(NO1,00044)TIME,I*(LB+2)
00044 FORMAT(1X,E10.4,I4,2X,5HEVENT)
00042 CONTINUE
IF(TRACE.GE.2.AND.(TRACE.EQ.1TRACE/2 2))GO TO 00045
GO TO 00048
00045 CONTINUE
WRITE(NO2,00048)TIME,I*(LB+2)
00048 FORMAT(1X,E10.4,I4,2X,5HEVENT)
CALL ICHECK
00046 CONTINUE
IF(I*(IPPOINT).GT.0)GO TO 00049
GO TO 00050
00049 CONTINUE
CALL EVDRS(1,2,RA(LV+2))
I=I*(IPPOINT)
L=I*(IPPOINT+1)
I*(LV+6)=L
INXT=1+2*NSTATS(IFIX(RAC(L,I+3)))
IF(IFIX(RAC(L,INXT+2)).GT.0.AND.INXT.LT.I*(L+3))GO TO 00052
GO TO 00053
00052 CONTINUE
CALL SAMPL(L,INXT+1,TD)
RA(LV+3)=TD
I*(LV+4)=IFIX(RAC(L,INXT+2))
I*(LV+5)=INXT
00053 CONTINUE
IF(IP(4).EQ.1)GO TO 00055
GO TO 00056
00055 CONTINUE
IF(RV(7).GT.TIME-RP(5))GO TO 00058
GO TO 00059
00058 CONTINUE
ASSIGN 00061 TO L00001

```

FORTRAN Function Programs

```

00061 GO TO 00001
00061 CONTINUE
00062 GO TO 00060
00059 CONTINUE      RA(LV+1)=-1.
00060 CONTINUE
00061 GO TO 00057
00056 IF(IP(4).EQ.2)GO TO 00062
00062 GO TO 00063
00062 CONTINUE
00063 ASSIGN 00064 TO L00002
00064 GO TO 00002
00064 CONTINUE
00064 GO TO 00057
00063 CONTINUE
00065 WRITE(NO1,00065)
00065 FORMAT(/1X,15H      ERROR      /1X,
X      38H503 DEPA      ILLEGAL MODE SPECIFICATION)
00057 CALL ERROR
00057 CONTINUE
00057 GO TO 00051
00050 IF(IA(IP0INT).EQ.0)GO TO 00066
00050 GO TO 00067
00066 CONTINUE
00066 IF(IP(6).EQ.0)GO TO 00068
00066 GO TO 00069
00068 CONTINUE
00071 WRITE(NO1,00071)
00071 FORMAT(/1X,15H      ERROR      /1X,24H503 DEPA      CONDITION LIST)
00069 CALL ERROR
00069 CONTINUE
00069 CALL PCOM(ILP(6,1),9,9,0,0,0,0,0,0,0)
00072 I=1-1)
00072 I=I+1
00072 IF(I) (1-(IA(IP(6)+3))).LE.0)GO TO 00073
00072 GO TO 00074
00073 CONTINUE
00073 MTO=ILP(6,1)
00073 IA(IP0INT+2)=9
00073 IA(LP)=1
00073 GO TO 99992
00075 CONTINUE
00075 IF(IA(IP0INT+2).NE.-9)GO TO 00076
00075 GO TO 00077
00076 CONTINUE
00076 WRITE(NO1,00079)
00079 FORMAT(/1X,15H      ERROR      /1X,
X      38H503 DEPA      RESPONSE TO CONDITION NOTICE)
00077 CALL ERROR
00077 CONTINUE
00077 GO TO 00072
00074 CONTINUE

```

FORTRAN Function Programs

```

      CALL FSTORE(IPPOINT)
00067 CONTINUE
00051 CONTINUE
      ICTYPE=-8
      IPPOINT=0
      GO TO 99992
99900 CONTINUE
      IF(ITRACE.GE.3)GO TO 00080
      GO TO 00081
00080 CONTINUE
      WRITE(N01,00083)TIME,IA(LB+2)
00083 FORMAT(1X,E10.4,14,2X,11HINTERACTION)
00081 CONTINUE
      IF(ITRACE.GE.4.AND.(ITRACE.EQ.ITRACE/2 2))GO TO 00084
      GO TO 00085
00084 CONTINUE
      WRITE(N02,00087)TIME,IA(LB+2)
00087 FORMAT(1X,E10.4,14,2X,11HINTERACTION)
      CALL ICHECK
00085 CONTINUE
      IF(IA(IPPOINT+2).NE.9)GO TO 00088
      GO TO 00089
00088 CONTINUE
      WRITE(N01,00091)
00091 FORMAT(/1X,15H      ERROR      /1X,
X      27H503 DEPA  (LEGAL INTERACTION)
      CALL ERROR
00089 CONTINUE
      IF(IP(4).EQ.1)GO TO 00092
      GO TO 00093
00092 CONTINUE
      IF(RV(1).LT.-1.5)GO TO 00095
      GO TO 00096
00095 CONTINUE
      RA(LV+7)=TIME
      GO TO 00097
00096 IF(RV(1).LT.-.5)GO TO 00098
      GO TO 00099
00098 CONTINUE
      ASSIGN 00100 TO L00001
      GO TO 00001
00000 CONTINUE
      GO TO 00097
00099 CONTINUE
      WRITE(N01,00101)
00101 FORMAT(/1X,15H      ERROR      /1X,27H503 DEPA  (CONDITIONAL START)
      CALL ERROR
00097 CONTINUE
      GO TO 00094
00093 IF(IP(4).EQ.2)GO TO 00102
      GO TO 00103
00102 CONTINUE

```

FORTRAN Function Programs

```

      IF(RV(1).LT.-1.5)GO TO 00104
      GO TO 00105
00104 CONTINUE
      RA(LV+7)=TIME
      GO TO 00106
00105 IF(RV(1).GE.(D.).AND.IV(4).GT.0)GO TO 00107
      GO TO 00108
00107 CONTINUE
      IF(TIME.LE.RV(1)+RV(2))GO TO 00109
      GO TO 00110
00109 CONTINUE
      CALL SCHED(RV(1)+RV(2)+RV(3),IV(4),IV(5),IV(6))
      GO TO 00111
00110 CONTINUE
      CALL SCHED(TIME+RV(3),IV(4),IV(5),IV(6))
      IA(LV+4)=0
      IA(LV+5)=0
      IA(LV+6)=0
00111 CONTINUE
00108 CONTINUE
00106 CONTINUE
00103 CONTINUE
00094 CONTINUE
      ASSIGN 00112 TO L00005
      GO TO 00005
00112 CONTINUE
      IA(IPOINT+2)=-9
      ICTYPE=-9
      IPOINT=IPOINT
      GO TO 99992
C
00001 CONTINUE
      IF(IV(4).GT.0)GO TO 00113
      GO TO 00114
00113 CONTINUE
      CALL SCHED(TIME+RV(2)+RV(3),IV(4),IV(5),IV(6))
00114 CONTINUE
      IF(IP(3).GT.0)GO TO 00116
      GO TO 00117
00116 CONTINUE
      ASSIGN 00119 TO L00003
      GO TO 00003
00119 CONTINUE
00117 CONTINUE
      IF(IP(6).GT.0)GO TO 00120
      GO TO 00121
00120 CONTINUE
      CALL SCHED(TIME+RV(2),IA(LB+2),0,0)
00121 CONTINUE
      GO TO L00001, (00061,00100)
C
00002 CONTINUE

```

FORTRAN Function Programs

```

      RA(LV+1)=TIME
      IF(IP(3).GT.0)GO TO 00123
      GO TO 00124
00123 CONTINUE
      ASSIGN 00126 TO L00003
      GO TO 00003
00126 CONTINUE
00124 CONTINUE
      IF(IP(6).GT.0)GO TO 00127
      GO TO 00128
00127 CONTINUE
      CALL SCHED(TIME+RV(2),IA(LB+2),0,0)
00128 CONTINUE
      IF(IV(4).GT.0)GO TO 00130
      GO TO 00131
00130 CONTINUE
      IF(RV(7).GT.TIME+RV(2)-RP(5))GO TO 00133
      GO TO 00134
00133 CONTINUE
      CALL SCHED(TIME+RV(2)+RV(3),IV(4),IV(5),IV(6))
      IA(LV+4)=0
      IA(LV+5)=0
      IA(LV+6)=0
00134 CONTINUE
00131 CONTINUE
      GO TO L00002, (00064)
C
00003 CONTINUE
      I=1
      CALL PCOM(IFIX(RLP(3,1)),9,7,IP(3),1+1,0,0,0,0,0)
00136 IF(I.LE.1A(IP(3)+3))GO TO 00137
      GO TO 00138
00137 CONTINUE
      MTO=RLP(3,1)
      ICTYPE=9
      IA(IP(3)+2)=7
      IA(IP(3)+5)=1+1
      IA(LP)=2
      GO TO 99992
00139 CONTINUE
      IF(IA(IP(3)+2).NE.-7)GO TO 00140
      GO TO 00141
00140 CONTINUE
      WRITE(NO1,00143)
00143 FORMAT(/IX,15H      ERROR      /IX,
X      35H503 DEPA  STIMULI DATA NOT ACCEPTED)
      CALL ERROR
00141 CONTINUE
      I=1+2
      I=1+NSTATS(IFIX(RLP(3,1)))+1
      I=1+NSTATS(IFIX(RLP(3,1)))+1
      I=1+NSTATS(IFIX(RLP(3,1)))+1

```



```

00138 GO TO 00136
CONTINUE
CALL FSTORE('POINT')
GO TO 00003, (00119,00126)
C
00004 CONTINUE
WRITE(NO1,00144)IA(LB+2)
00144 FORMAT(/,9H FUNCTION,14,1X,8H503 DEPA)
RA(LCS(IP(7),2))=RA(LCS(IP(7),2))/FLOAT(IA(LCS(IP(7),1)))
SD=RA(LCS(IP(7),3))/FLOAT(IA(LCS(IP(7),1)))-RA(LCS(IP(7),2)) 2
IF(SD GT 0.)GO TO 00145
GO TO 00146
00145 CONTINUE
SD=SD*SD
GO TO 00147
00146 CONTINUE
SD=0.
00147 CONTINUE
WRITE(NO1,00148)RA(LCS(IP(7),2)),SD
00148 FORMAT(11X,22HMEAN TIME OF CONDITION,E12.4 /11X,
X 18HSTANDARD DEVIATION,E12.4)
WRITE(NO1,00149)IA(LCS(IP(7),1)),RA(LCS(IP(7),4)),RA(LCS(IP(7),5)
X )
00149 FORMAT(11X,22HNUMBER OF OBSERVATIONS,110 /11X,12HMAXIMUM TIME,
X E12.4 /11X,12HMINIMUM TIME,E12.4)
GO TO 00004, (00031)
C
00005 CONTINUE
IA(LCS(IP(7),1))=IA(LCS(IP(7),1))+1
RA(LCS(IP(7),2))=RA(LCS(IP(7),2))+TIME
RA(LCS(IP(7),3))=RA(LCS(IP(7),3))+TIME
IF(TIME GT RA(LCS(IP(7),4)))GO TO 00150
GO TO 00151
00150 CONTINUE
RA(LCS(IP(7),4))=TIME
00151 CONTINUE
IF(TIME LT RA(LCS(IP(7),5)))GO TO 00153
GO TO 00154
00153 CONTINUE
RA(LCS(IP(7),5))=TIME
00154 CONTINUE
GO TO 00005, (00112)
00026 100000=IA(LP)
GO TO (00075,00139), 100000
END
C 504 MODA 3/16/78
C MODIFICATION ACTION
C LP+0 1 RESPONSE POINTER
C 1 1 DURATION TIME DISTRIBUTION TYPE
C 2 4 DURATION TIME STATISTICS
C 3 4 STIMULUS SPECIFICATIONS (REPEAT FOR EACH TYPE)
C STIMULUS MONITOR FUNCTION

```

FORTRAN Function Programs

```

C      STIMULUS TYPE
C      DELAY TIME DISTRIBUTION TYPE
C      DELAY TIME STATISTICS
C      DURATION TIME DISTRIBUTION TYPE
C      DURATION TIME STATISTICS
C      INTENSITY DISTRIBUTION TYPE
C      INTENSITY STATISTICS
C      4 2 MODIFICATION SPECIFICATIONS (REPEAT FOR EACH MODIFICATION)
C      FUNCTION
C      PARAMETER/VARIABLE
C      FIRST LIST INDEX
C      LAST LIST INDEX
C      5 4 NEW VALUES
C      6 4 INITIAL VALUES
C      7 2 CONDITION FUNCTIONS
C      LV+0 1 ENTRY COUNT
C      LI+0 0
C      SUBROUTINE B504
C      DIMENSION IPVT(9)
C      DIMENSION IA(1)
C      COMMON RA(1)
C      EQUIVALENCE (IA(1),RA(1))
C      COMMON /SYSTEM/MFROM, MTO, ICTYPE, IPOINT,
C      IIDUMP, ISECA, ITRACE, LB, LBGA, LBLIST, LBPT, LBUF, LCSB, LE,
C      ILEC, LFFB, LPFE, LI, LLE, LLFB, LLFE, LMS, LNE, LP, LV, MINNFI,
C      IMNSE, NFE, NPL, NI1, NI2, NO1, NO2, NREP, NRUI, NSE, TIME, TSTOP
C      DATA IPVT/1,1,4,2,4,2,4,2,1/
C      IA(LV)=IA(LV)+1
C      IF (ICTYPE.GE.0) GO TO 99991
C      GO TO 00026
C 99991 GO TO (99100,99200,99300,99400,99500,99600,99700,99800,99900)
C      * ICTYPE
C 99992 RETURN
C 99100 CALL CREATE(504,8,1,IPVT)
C      ICTYPE=-1
C      IPOINT=0
C      GO TO 99992
C 99200 CALL DESTRO(IPVT)
C      ICTYPE=-2
C      IPOINT=0
C      GO TO 99992
C 99300 CALL SET(1,IPVT,INTD,REALD)
C      ICTYPE=-3
C      IPOINT=0
C      GO TO 99992
C 99400 CONTINUE
C      WRITE(NO1,00027)
C 00027 FORMAT(/IX,15H      ERROR      /IX,
C      * 43H504 MODA  ATTEMPT TO SET INITIAL CONDITIONS)
C 99500 CONTINUE
C      WRITE(NO1,00028)IA(LB+2)

```

FORTRAN Function Programs

```

00028 FORMAT(//.9H FUNCTION,14,1X,6H504 MODA/11/.9HNO REPORT)
      ICTYPE=-5
      IPOINT=0
      GO TO 99992
99600 CONTINUE
      I=I+1
00029 IF(I) ((I-(IA(LB+4))) .LE. 0) GO TO 00030
      GO TO 00031
00030 CONTINUE
      IA(LV-1+I)=IA(L)-1+I
      GO TO 00029
00031 CONTINUE
      NPVAL=6
      ASSIGN 00032 TO L00002
      GO TO 00002
00032 CONTINUE
      ICTYPE=-6
      IPOINT=0
      GO TO 99992
99700 CALL DUMP(IPVT)
      ICTYPE=-7
      IPOINT=0
      GO TO 99992
99800 CONTINUE
      IF(TRACE .GE. 1) GO TO 00033
      GO TO 00034
00033 CONTINUE
      WRITE(NO1,00036) TIME, IA(LB+2)
00036 FORMAT(1X,E10.4,14,2X,5HEVENT)
00034 CONTINUE
      IF(TRACE .GE. 2 .AND. (TRACE .EQ. 1 TRACE/2)) GO TO 00037
      GO TO 00038
00037 CONTINUE
      WRITE(NO2,00040) TIME, IA(LB+2)
00040 FORMAT(1X,E10.4,14,2X,5HEVENT)
00038 CONTINUE
      CALL ICHECK
00039 CONTINUE
      IF(IA(IPOINT) .GT. 0) GO TO 00041
      GO TO 00042
00041 CONTINUE
      CALL EVDRS(1,2,D)
      IF(IP(7) .GT. 0) GO TO 00044
      GO TO 00045
00044 CONTINUE
      CALL SCHED(TIME+D,IA(LB+2),0,0)
00045 CONTINUE
      I=IA(IPOINT)
      L=IA(IPOINT+1)
      INXT=I+2+NSTATS((FIX(RAC(L,I+3)))
      IF((FIX(RAC(L,INXT+2)) .GT. 0 .AND. INXT .LT. IA(L+3)) GO TO 00047
      GO TO 00048

```

FORTRAN Function Programs

```

00047 CONTINUE
      CALL SAMPL(L,INXT+1,TD)
      CALL SCHED(TIME+D+TD,IFIX(RAC(L,INXT+2)),INXT,L)
00048 CONTINUE
      IF(IP(3).GT.0)GO TO 00050
      GO TO 00051
00050 CONTINUE
      ASSIGN 00053 TO L00001
      GO TO 00001
00053 CONTINUE
00051 CONTINUE
      IF(IP(4).GT.0)GO TO 00054
      GO TO 00055
00054 CONTINUE
      NPVAL=5
      ASSIGN 00057 TO L00002
      GO TO 00002
00057 CONTINUE
00055 CONTINUE
      GO TO 00043
00042 IF(IA(IPPOINT).EQ.0)GO TO 00058
      GO TO 00059
00058 CONTINUE
      IF(IP(7).EQ.0)GO TO 00060
      GO TO 00061
00060 CONTINUE
      WRITE(N01,00063)
00063 FORMAT(/IX,15H      ERROR      /IX,24H504 MODA  CONDITION LIST)
      CALL ERROR
00061 CONTINUE
      CALL PCOM(ILP(7,1),9,9,0,0,0,0,0,0,0)
      I=1-11)
00064 I=I+1
      IF(I) (1-(IA(IP(7)+3))).LE.0)GO TO 00065
      GO TO 00066
00065 CONTINUE
      MTD=ILP(7,1)
      IA(IPPOINT+2)=9
      IA(LP)=1
      GO TO 99992
00067 CONTINUE
      IF(IA(IPPOINT+2).NE.-9)GO TO 00068
      GO TO 00069
00068 CONTINUE
      WRITE(N01,00071)
00071 FORMAT(/IX,15H      ERROR      /IX,
X 38H504 MODA  RESPONSE TO CONDITION NOTICE)
      CALL ERROR
00069 CONTINUE
      GO TO 00064
00066 CONTINUE
      CALL FSTORE(IPPOINT)

```

FORTRAN Function Programs

```

00059 CONTINUE
00043 CONTINUE
      ICTYPE=-8
      IPOINT=0
      GO TO 99992
99900 CONTINUE
      IF(IITRACE.GE.3)GO TO 00072
      GO TO 00073
00072 CONTINUE
      WRITE(N01,00075)TIME,IA(LB+2)
00075 FORMAT(1X,E10.4,14,2X,11#INTERACTION)
00073 CONTINUE
      IF(IITRACE.GE.4.AND.(IITRACE.EQ.IITRACE/2 2))GO TO 00076
      GO TO 00077
00076 CONTINUE
      WRITE(N02,00079)TIME,IA(LB+2)
00079 FORMAT(1X,E10.4,14,2X,11#INTERACTION)
      CALL ICHECK
00077 CONTINUE
      WRITE(N01,00080)
00080 FORMAT(/1X,15H      ERROR      /1X
X      29H504 MODA  !LLEGAL INTERACTION)
      CALL ERROR
      ICTYPE=-9
      IPOINT=0
      GO TO 99992
C
00001 CONTINUE
      I=1
      CALL PCOM(1FIX(RLP(3,1)),9,7,IP(3),1+1,0,0,0,0,0)
00081 IF(1.LE.1A(IP(3)+3))GO TO 00082
      GO TO 00083
00082 CONTINUE
      MFRGM=1A(LB+2)
      MTO=RLP(3,1)
      ICTYPE=9
      1A(IPOINT+2)=7
      1A(IPOINT+4)=1+1
      1A(LP)=2
      GO TO 99992
00084 CONTINUE
      IF(1A(IPOINT+2).NE.-7)GO TO 00085
      GO TO 00086
00085 CONTINUE
      WRITE(N01,00088)
00088 FORMAT(/1X,15H      ERROR      /1X
X      35H504 MODA  STIMULI DATA NOT ACCEPTED)
      CALL ERROR
00086 CONTINUE
      I=1+2
      I=1+NSTATS(1FIX(RLP(3,1)))+1
      I=1+NSTATS(1FIX(RLP(3,1)))+1

```

FORTRAN Function Programs

```

      I=1+NSTATS(IFIX(RLP(3,1)))+1
      GO TO 00081
00083 CONTINUE
      CALL FSTORE(IPPOINT)
      GO TO L00001, (00053)
C
00002 CONTINUE
      I=1
      J=1
00089 IF(I.LE.1A(IF(4)+3))GO TO 00090
      GO TO 00091
00090 CONTINUE
      NF=ILP(4,1)
      LBF=1A(LCS(LBPT,NF+1))
      IF(LBF.GT.0)GO TO 00092
      GO TO 00093
00092 CONTINUE
      ASSIGN 00095 TO L00003
      GO TO 00003
00095 CONTINUE
      GO TO 00094
00093 CONTINUE
      WRITE(N01,00096)
00096 FORMAT(/1X,15H      ERROR      /1X,
X 49H504 MODA  ATTEMPT TO MODIFY AN UNDEFINED FUNCTION)
      CALL ERROR
00094 CONTINUE
      I=1+4
      GO TO 00089
00091 CONTINUE
      GO TO L00002, (00032,00057)
C
00003 CONTINUE
      NPV=ILP(4,1+1)
      N=ILP(4,1+2)
      M=ILP(4,1+3)
      IF(NPV.GT.0)GO TO 00097
      GO TO 00098
00097 CONTINUE
      LPVF=LBF+5
      ASSIGN 00100 TO L00004
      GO TO 00004
00100 CONTINUE
      GO TO 00099
00098 IF(NPV.LT.0)GO TO 00101
      GO TO 00102
00101 CONTINUE
      NPV=-NPV
      LPVF=LBF+5+1A(LBF+3)
      ASSIGN 00103 TO L00004
      GO TO 00004
00103 CONTINUE

```

FORTRAN Function Programs

```

      GO TO 00099
00102 CONTINUE
      WRITE(N01,00104)
00104 FORMAT(/1X,15H      ERROR      /1X,
X      48H504 MODA  ILLEGAL PARAMETER/VARIABLE NUMBER ZERO)
      CALL ERROR
00099 CONTINUE
      GO TO 1,00003, (00095)
C
00004 CONTINUE
      IF(N.EQ.0)GO TO 00105
      GO TO 00106
00105 CONTINUE
      IF(M.EQ.0)GO TO 00108
      GO TO 00109
00108 CONTINUE
      IA(LPVF+NPV)=IFIX(RLP(NPVAL,J))
      GO TO 00110
00109 IF(M.EQ.1)GO TO 00111
      GO TO 00112
00111 CONTINUE
      RA(LPVF+NPV)=RLP(NPVAL,J)
      GO TO 00110
00112 CONTINUE
      WRITE(N01,00113)
00113 FORMAT(/1X,15H      ERROR      /1X,
X      31H504 MODA  LAST INDEX FOR SCALAR)
      CALL ERROR
00110 CONTINUE
      J=J+1
      GO TO 00107
00106 IF(N.GT.0)GO TO 00114
      GO TO 00115
00114 CONTINUE
      IF(M.GT.0.AND.M.GE N)GO TO 00116
      GO TO 00117
00116 CONTINUE
      K=N-11)
00119 K=K+1
      IF(1) (K-(M)).LE.0)GO TO 00120
      GO TO 00121
00120 CONTINUE
      IA(LCS(IA(LPVF+NPV),K+2))=IFIX(RLP(NPVAL,J))
      J=J+1
      GO TO 00119
00121 CONTINUE
      GO TO 00118
00117 CONTINUE
      WRITE(N01,00122)
00122 FORMAT(/1X,15H      ERROR      /1X,
X      37H504 MODA  LAST INDEX FOR INTEGER LIST)
      CALL ERROR

```

FORTRAN Function Programs

```

00118 CONTINUE
      GO TO 00107
00115 IF(N.LT.0)GO TO 00123
      GO TO 00124
00123 CONTINUE
      N=-N
      IF(M.GT.0.AND.M.GE.N)GO TO 00125
      GO TO 00126
00125 CONTINUE
      K=N-1
00128 K=K+1
      IF(1) (K-(M)).LE.0)GO TO 00129
      GO TO 00130
00129 CONTINUE
      RA(LCS(1A(LPVF+NPV),K+2))=RLP(NPVAL,J)
      J=J+1
      GO TO 00128
00130 CONTINUE
      GO TO 00127
00126 CONTINUE
      WRITE(NO1,00131)
00131 FORMAT(/1X,15H ERROR /1X,
X 34H504 MODA LAST INDEX FOR REAL LIST)
      CALL ERROR
00127 CONTINUE
00124 CONTINUE
00107 CONTINUE
      GO TO L00004, (00100,00103)
00026 100000=1A(LP)
      GO TO (00067,00084), 100000
      END
C 505 HDAS 3/16/78
C HALT DIVERSION ACTION SEQUENCE
C LP+0 1 RESPONSE POINTER
C LV+0 1 ENTRY COUNT
C 1 1 HALT STATUS
C 2 1 SAVE 1 = 1A(1POINT)
C 3 1 SAVE .. = 1A(1POINT+1)
C LI+0 0
C 1 0
C 2 0
C 3 0
      SUBROUTINE B505
      DIMENSION IPVT(5)
      DIMENSION 1A(1)
      COMMON RA(1)
      EQUIVALENCE (1A(1),RA(1))
      COMMON /SYSTEM/MFROM,MT0,ICTYPE,1POINT,
11DUMP,ISECA,ITRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
11LEC,LFEB,LFFE,L1,LLE,LLFB,LLFE,LMS,LINE,LP,LV,MINNFL,
11MSE,NFE,NFL,N11,N12,NO1,NO2,NREP,NRUN,NSE,TIME,TSTOP
      DATA IPVT/1,1,1,1,1/

```


FORTRAN Function Programs

```

      IA(LV)=IA(LV)+1
      IF(ICTYPE GE.0)GO TO 99991
      GO TO 00026
99991 GO TO (99100,99200,99300,99400,99500,99600,99700,99800,99900)
      X, ICTYPE
99992 RETURN
99100 CALL CREATE(505,1,4,IPVT)
      ICTYPE=-1
      IPOINT=0
      GO TO 99992
99200 CALL DESTROY(IPVT)
      ICTYPE=2
      IPOINT=0
      GO TO 99992
99300 CONTINUE
      WRITE(NO1,00027)
00027 FORMAT(/1X,15H      ERROR      /1X,
      X      35H505.HDAS  ATTEMPT TO SET PARAMETERS)
99400 CALL SET(2,IPVT,INTD,REALD)
      ICTYPE=-4
      IPOINT=0
      GO TO 99992
99500 CONTINUE
      WRITE(NO1,00028)IA(LB+2)
00028 FORMAT(/,9H FUNCTION,14,1X,35H505 HDAS/11X,9HNO REPORT)
      ICTYPE=5
      IPOINT=0
      GO TO 99992
99600 CONTINUE
      I=1-(I)
00029 IF(I) (1-(IA(LB+4))).LE.0)GO TO 00030
      GO TO 00031
00030 CONTINUE
      IA(LV-1+1)=IA(LI-1+1)
      GO TO 00029
00031 CONTINUE
      ICTYPE=-6
      IPOINT=0
      GO TO 99992
99700 CALL DUMP(IPVT)
      ICTYPE=-7
      IPOINT=0
      GO TO 99992
99800 CONTINUE
      IF(TRACE GE.1)GO TO 00032
      GO TO 00033
00032 CONTINUE
      WRITE(NO1,00035)TIME,IA(LB+2)
00035 FORMAT(1X,E10.4,14,2X,5HEVENT)
00033 CONTINUE

```

FORTRAN Function Programs

```

      IF(ITRACE.GE.2.AND.(ITRACE.EQ.ITRACE/2 2))GO TO 00036
      GO TO 00037
00036 CONTINUE
      WRITE(N02,00039)TIME,IA(LB+2)
00039 FORMAT(1X,E10.4,14,2X,5HEVENT)
      CALL ICHECK
00037 CONTINUE
      IA(LV+2)=IA(IPOINT)
      IA(LV+3)=IA(IPOINT+1)
      IF(IV(1).EQ.0)GO TO 00040
      GO TO 00041
00040 CONTINUE
      ASSIGN 00043 TO L00001
      GO TO 00001
00043 CONTINUE
00041 CONTINUE
      ICTYPE=-8
      IPOINT=0
      GO TO 99992
99900 CONTINUE
      IF(ITRACE.GE.3)GO TO 00044
      GO TO 00045
00044 CONTINUE
      WRITE(N01,00047)TIME,IA(LB+2)
00047 FORMAT(1X,E10.4,14,2X,11HINTERACTION)
00045 CONTINUE
      IF(ITRACE.GE.4.AND.(ITRACE.EQ.ITRACE/2 2))GO TO 00048
      GO TO 00049
00048 CONTINUE
      WRITE(N02,00051)TIME,IA(LB+2)
00051 FORMAT(1X,E10.4,14,2X,11HINTERACTION)
      CALL ICHECK
00049 CONTINUE
      IF(IA(IPOINT+2).EQ.8)GO TO 00052
      GO TO 00053
00052 CONTINUE
      IF(IV(1).EQ.1)GO TO 00055
      GO TO 00056
00055 CONTINUE
      IA(LV+1)=0
      ASSIGN 00058 TO L00001
      GO TO 00001
00058 CONTINUE
00056 CONTINUE
      GO TO 00054
00053 CONTINUE
      WRITE(N01,00059)
00059 FORMAT(/1X,15H      ERROR      /1X,
X      29H505 HDAS  ILLEGAL INTERACTION)
      CALL ERROR
00054 CONTINUE
      IA(IPOINT+2)=-8

```

FORTRAN Function Programs

```

      ICT/PE=-9
      IPOINT=0
      GO TO 99992
C
00001 CONTINUE
      I=I+1
      L=I+1
      INXT=1+2*NSTATS*(I/I*(RAC(L,1+3)))
      IF(I/I*(RAC(L,INXT+2))>0 AND INXT LT 1A(L+3))GO TO 00060
00060 CONTINUE
      CALL SAMPL(L,INXT+1,TD)
      CALL SCHED(TIME+TD,I/I*(RAC(L,INXT+2)),INXT,L)
00061 CONTINUE
      GO TO 00001, (00043,00058)
00026 CONTINUE
      WRITE(ND1,00063)
00063 FORMAT(1X,15H      ERROR      /1X,
X      37ILLEGAL RESPONSE COMMUNICATION (BNNN))
      CALL ERROR
      END
C 601 TIMR 3/16/78
C TIMER
LP+0 1 RESPONSE POINTER
      1 3 PERIOD
      2 1 NUMBER OF SHIFTS PER PERIOD
      3 4 SHIFT STARTING TIMES
      4 4 SHIFT ENDING TIMES
      5 1 NUMBER OF INTERVALS PER PERIOD
      6 4 INTERVAL STARTING TIMES
      7 4 INTERVAL ENDING TIMES
LV+0 1 ENTRY COUNT
      1 3 TIME CURRENT PERIOD WILL END
      2 1 CURRENT SHIFT
      3 3 TIME CURRENT SHIFT WILL END
      4 3 TIME NEXT SHIFT WILL BEGIN
      5 1 CURRENT INTERVAL
      6 3 TIME CURRENT INTERVAL WILL END
      7 3 TIME NEXT INTERVAL WILL BEGIN
LI+0 0
      1 RP(1)
      2 0
      3 0
      4 RLP(3,1)
      5 0
      6 0
      7 RLP(6,1)
      SUBROUTINE B6D1
      DIMENSION IPVT(16)
      DIMENSION 1A(1)
      COMMON RA(1)
      EQUIVALENCE (1A(1),RA(1))

```

FORTRAN Function Programs

```

COMMON /SYSTEM/MFROM, MTO, ICTYPE, IPOINT,
1IDUMP, ISECA, ITRADE, LB, LBGA, LBLIST, LBPT, LBUF, LCSD, LE,
1LEF, LFFB, LFE, LI, LLE, LLFB, LLE, LMS, LNE, LP, LV, MINFL,
1MISE, NFE, NFL, N11, N12, NO1, NO2, NREP, NRUN, NSE, TIME, TSTOP
DATA IPVT/1,3,1,4,4,1,4,4,1,3,1,3,3,1,3,3/
IA(LV)=IA(LV)+1
IF(ICTYPE GE.0)GO TO 99991
GO TO 00026
99991 GO TO (99100,99200,99300,99400,99500,99600,99700,99800,99900)
      / ICTYPE
99992 RETURN
99100 CALL CREATE(601,8,8,IPVT)
      ICTYPE=-1
      IPOINT=0
      GO TO 99992
99200 CALL DESTRO(IPVT)
      ICTYPE=-2
      IPOINT=0
      GO TO 99992
99300 CALL SET(1,IPVT,INTD,REALD)
      ICTYPE=-3
      IPOINT=0
      GO TO 99992
99400 CONTINUE
      WRITE(NO1,00027)
00027 FORMAT(/1X,15H      ERROR      /1X,
X      43H601 TIMR  ATTEMPT TO SET INITIAL CONDITIONS)
      CALL ERROR
99500 CONTINUE
      WRITE(NO1,00028)IA(LB+2)
00028 FORMAT(/,9H FUNCTION,14,1X,8H601 TIMR/11X,9HNC REPCRT)
      ICTYPE=-5
      IPOINT=0
      GO TO 99992
99600 CONTINUE
      RAL(I+1)=RP(1)
      RAL(I+4)=RLP(3,1)
      RAL(I+7)=RLP(6,1)
      I=I+11
00029 I=I+1
      IF((I-(IA(LB+4))).LE.0)GO TO 00030
      GO TO 00031
00030 CONTINUE
      IA(LV-1+1)=IA(LI-1+1)
      GO TO 00029
00031 CONTINUE
      CALL SCHED(TIME,IA(LB+2),1,0)
      IF(IP(2).GT.0)GO TO 00032
      GO TO 00033
00032 CONTINUE
      CALL SCHED(RV(4),IA(LB+2),2,0)
00033 CONTINUE

```

FORTRAN Function Programs

```

      IF (TYPE) GT 0 GO TO 00035
      GO TO 00036
00034 CONTINUE
      CALL SCHED(RV(7),IA(LB+2),3,0)
00036 CONTINUE
      ICTYPE=6
      IPDINT=0
      GO TO 00032
00035 CALL DUMP(IP/T)
      ICTYPE=7
      IPDINT=0
      GO TO 00032
00038 CONTINUE
      IF (ITRACE GE 1) GO TO 00038
      GO TO 00039
00038 CONTINUE
      WRITE(N01,00041)TIME,IA(LB+2)
00041 FORMAT(1X,E10.4,14,2X,5HEVENT)
00039 CONTINUE
      IF (ITRACE GE 2 AND (ITRACE EQ ITRACE/2 2)) GO TO 00042
      GO TO 00043
00042 CONTINUE
      WRITE(N02,00045)TIME,IA(LB+2)
00045 FORMAT(1X,E10.4,14,2X,5HEVENT)
      CALL ICHECK
00043 CONTINUE
      CALL CEC(1,5,KODE)
      IF (KODE EQ 1) GO TO 00046
      GO TO 00047
00046 CONTINUE
      RA(LV+1)=TIME+RP(1)
      CALL SCHED(RV(1),MT0,1,0)
      IF (ITRACE GE 1) GO TO 00049
      GO TO 00050
00049 CONTINUE
      WRITE(N01,00052)TIME,IA(LB+2)
00052 FORMAT(1X,E10.4,14,2X,22H601 TIMR BEGIN PERIOD)
00050 CONTINUE
      IF (ITRACE GE 2 AND (ITRACE EQ ITRACE/2 2)) GO TO 00053
      GO TO 00054
00053 CONTINUE
      WRITE(N02,00056)TIME,IA(LB+2)
00056 FORMAT(1X,E10.4,14,2X,22H601 TIMR BEGIN PERIOD)
      CALL ICHECK
00054 CONTINUE
      TPB=TIME
      IA(LV+2)=0
      IA(LV+5)=0
      GO TO 00048
00047 IF (KODE EQ 2) GO TO 00057
      GO TO 00058
00057 CONTINUE

```

```

      RA(LV+2)=TPB+RLP(4,IV(2))
      CALL SCHED(RV(3),MTO,4,0)
      IF(IV(2) NE IP(2))GO TO 00059
00058 GO TO 00060
00059 CONTINUE
      RA(LV+4)=TPB+RLP(3,IV(2))
      GO TO 00061
00060 CONTINUE
      RA(LV+4)=TPB+RP(1)+R.P(3)
00061 CONTINUE
      IF(ITRACE GE 1)GO TO 00062
      GO TO 00063
00062 CONTINUE
      WRITE(N01,00065)TIME,IA(LB+2)
00065 FORMAT(IX,E10.4,I4,2X,21H601 TIMR BEGIN INTERVA
00063 CONTINUE
      IF(ITRACE GE 2 AND (ITRACE EQ ITRACE/2))GO TO 00064
      GO TO 00067
00066 CONTINUE
      WRITE(N02,00069)TIME,IA(LB+2)
00069 FORMAT(IX,E10.4,I4,2X,21H601 TIMR BEGIN INTERVA
      CALL ICHECK
00067 CONTINUE
      GO TO 00048
00068 IF(KODE EQ 3)GO TO 00070
      GO TO 00071
00070 CONTINUE
      RA(LV+5)=TPB+RLP(7,IV(5))
      CALL SCHED(RV(6),MTO,5,0)
      IF(IV(5) NE IP(5))GO TO 00072
      GO TO 00073
00072 CONTINUE
      RA(LV+7)=TPB+RLP(6,IV(5)+1)
      GO TO 00074
00073 CONTINUE
      RA(LV+7)=TPB+RP(1)+R.P(6,1)
00074 CONTINUE
      IF(ITRACE GE 1)GO TO 00075
      GO TO 00076
00075 CONTINUE
      WRITE(N01,00078)TIME,IA(LB+2)
00078 FORMAT(IX,E10.4,I4,2X,24H601 TIMR BEGIN INTERVAL
00076 CONTINUE
      IF(ITRACE GE 2 AND (ITRACE EQ ITRACE/2))GO TO 00079
      GO TO 00080
00079 CONTINUE
      WRITE(N02,00082)TIME,IA(LB+2)
00082 FORMAT(IX,E10.4,I4,2X,24H601 TIMR BEGIN INTERVA
      CALL ICHECK
00080 CONTINUE

```

FORTRAN Function Programs

```

00071 GO TO 00048
      IF(KODE.EQ.4)GO TO 00083
      GO TO 00084
00083 CONTINUE
      IA(LV+2)=-IA(LV+2)
      CALL SCHED(RV(4),MT0,2,0)
      RA(LV+3)=0
      IF(ITRACE.GE.1)GO TO 00085
      GO TO 00086
00085 CONTINUE
      WRITE(N01,00088)TIME,IA(LB+2)
00088 FORMAT(1X,E10.4,14.2X,19H601 TIMR  END SHIFT)
00086 CONTINUE
      IF(ITRACE.GE.2 AND (ITRACE.EQ ITRACE/2 2))GO TO 00089
      GO TO 00090
00089 CONTINUE
      WRITE(N02,00092)TIME,IA(LB+2)
00092 FORMAT(1X,E10.4,14.2X,19H601 TIMR  END SHIFT)
      CALL ICHECK
00090 CONTINUE
      GO TO 00048
00084 IF(KODE.EQ.5)GO TO 00093
      GO TO 00094
00093 CONTINUE
      IA(LV+5)=-IA(LV+5)
      CALL SCHED(RV(7),MT0,3,0)
      RA(LV+6)=0
      IF(ITRACE.GE.1)GO TO 00095
      GO TO 00096
00095 CONTINUE
      WRITE(N01,00098)TIME,IA(LB+2)
00098 FORMAT(1X,E10.4,14.2X,22H601 TIMR  END INTERVAL)
00096 CONTINUE
      IF(ITRACE.GE.2 AND (ITRACE.EQ ITRACE/2 2))GO TO 00099
      GO TO 00100
00099 CONTINUE
      WRITE(N02,00102)TIME,IA(LB+2)
00102 FORMAT(1X,E10.4,14.2X,22H601 TIMR  END INTERVAL)
      CALL ICHECK
00100 CONTINUE
00094 CONTINUE
00048 CONTINUE
      ICTYPE=-8
      IPOINT=0
      GO TO 99992
99900 CONTINUE
      IF(ITRACE.GE.0)GO TO 00103
      GO TO 00104
00103 CONTINUE
      WRITE(N01,00106)TIME,IA(LB+2)
00106 FORMAT(1X,E10.4,14.2X,11HINTERACTION)
00104 CONTINUE

```

FORTRAN Function Programs

```

      IF((ITRACE.GE.4.AND.(ITRACE.EQ.ITRACE/2 2))GO TO 00107
      GO TO 00108
00107 CONTINUE
      WRITE(N02,00110)TIME,IA(LB+2)
00110 FORMAT(1X,E10.4,14,2X,31H601 TIMR RETURNING TIMING DATA)
      CALL ICHECK
00108 CONTINUE
      IF((ITRACE.GE.3)GO TO 00111
      GO TO 00112
00111 CONTINUE
      WRITE(N01,00114)TIME,IA(LB+2)
00114 FORMAT(1X,E10.4,14,2X,31H601 TIMR RETURNING TIMING DATA)
00112 CONTINUE
      IF((ITRACE.GE.4.AND.(ITRACE.EQ.ITRACE/2 2))GO TO 00115
      GO TO 00116
00115 CONTINUE
      WRITE(N02,00118)TIME,IA(LB+2)
00118 FORMAT(1X,E10.4,14,2X,31H601 TIMR RETURNING TIMING DATA)
      CALL ICHECK
00116 CONTINUE
      ICTYPE=-9
      IPOINT=LV+1
      GO TO 99992
00026 CONTINUE
      WRITE(N01,00119)
00119 FORMAT(/1X,15H      ERROR      /1X,
      X      37HILLEGAL RESPONSE COMMUNICATION (BNNN))
      CALL ERROR
      END
C 602 SCAS 3/16/78
      START CONTROL ACTION SEQUENCE
C LP+0 1 RESPONSE POINTER
C 1 4 ACTION SEQUENCE SPECIFICATION. REPEAT FOR EACH ACTION:
C 1) FUNCTION (NEGATIVE FOR FIRST ACTION OF SUBSEQUENCES)
C 2) ACTION STARTING TIME DELAY DISTRIBUTION TYPE, 3) ACTION
C STARTING TIME DELAY STATISTICS (0 OR MORE VALUES)
C LV+0 1 ENTRY COUNT
C LI+0 0
      SUBROUTINE B602
      DIMENSION IPVT(3)
      DIMENSION IA(1)
      COMMON RA(1)
      EQUIVALENCE (IA(1),RA(1))
      COMMON /SYSTEM/MFROM,HTO,ICTYPE,IPOINT,
      1IDUMP,ISECA,ITRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
      1LEC,LFFB,LFPE,L1,L2,L3,L4,L5,L6,L7,L8,L9,L10,L11,L12,L13,
      1MSE,NFE,NFL,N11,N12,N01,N02,NREP,NRUN,NSE,TIME,TSTOP
      DATA IPVT/1,4,1/
      IA(LV)=IA(LV)+1
      IF(ICTYPE.GE.0)GO TO 99991
      GO TO 00026
99991 GO TO (99100,99200,99300,99400,99500,99600,99700,99800,99900)

```


FORTRAN Function Programs

```

X, ICTYPE
99992 RETURN
99100 CALL CREATE(602,2,1,IPVT)
      ICTYPE=-1
      IPOINT=0
      GO TO 99992
99200 CALL DESTRO(IPVT)
      ICTYPE=-2
      IPOINT=0
      GO TO 99992
99300 CALL SET(1,IPVT,INTD,REALD)
      ICTYPE=-3
      IPOINT=0
      GO TO 99992
99400 CONTINUE
      WRITE(NO1,00027)
00027 FORMAT(1X,15H      ERROR      /1X,
X      43H602 SCAS  ATTEMPT TO SET INITIAL CONDITIONS)
      CALL ERROR
99500 CONTINUE
      WRITE(NO1,00028)IA(LB+2)
00028 FORMAT(//.9H FUNCTION,14,1X,8H602 SCAS/11X,9HNO REPORT)
      ICTYPE=-5
      IPOINT=0
      GO TO 99992
99600 CONTINUE
      I=1-11
00029 IF(I) (1-(IA(LB+4))),LE.0)GO TO 00030
      GO TO 00031
00030 CONTINUE
      IA(LV-1+1)=IA(LI-1+1)
      GO TO 00029
00031 CONTINUE
      ICTYPE=-6
      IPOINT=0
      GO TO 99992
99700 CALL DUMP(IPVT)
      ICTYPE=-7
      IPOINT=0
      GO TO 99992
99800 CONTINUE
      IF(ITRACE.GE.1)GO TO 00032
      GO TO 00033
00032 CONTINUE
      WRITE(NO1,00035)TIME,IA(LB+2)
00035 FORMAT(1X,E10.4,14,2X,5HEVENT)
00033 CONTINUE
      IF(ITRACE.GE.2.AND.(ITRACE.EQ.ITRACE/2 2))GO TO 00036
      GO TO 00037
00036 CONTINUE
      WRITE(NO2,00039)TIME,IA(LB+2)

```

FORTRAN Function Programs

```

00039 FORMAT(1X,E10.4,14,2X,5HEVENT)
      CALL ICHECK
00037 CONTINUE
      WRITE(N01,00040)
00040 FORMAT(/1X,15H      ERROR      /1X,
X      31H602 SCAS  ILLEGAL EVENT MESSAGE)
      CALL ERROR
      ICTYPE=-8
      IPOINT=0
      GO TO 99992
99900 CONTINUE
      IF(ITRACE.GE.3)GO TO 00041
      GO TO 00042
00041 CONTINUE
      WRITE(N01,00044)TIME,IA(LB+2)
00044 FORMAT(1X,E10.4,14,2X,11HINTERACTION)
00042 CONTINUE
      IF(ITRACE.GE.4.AND.(ITRACE.EQ.ITRACE/2 2))GO TO 00045
      GO TO 00046
00045 CONTINUE
      WRITE(N02,00048)TIME,IA(LB+2)
00048 FORMAT(1X,E10.4,14,2X,11HINTERACTION)
      CALL ICHECK
00046 CONTINUE
      IF(IA(IPOINT+2).EQ.11)GO TO 00049
      GO TO 00050
00049 CONTINUE
      I=I-11
00052 I=I+1
      IF(1) ((I-(IA(IP(1)+3))).LE.0)GO TO 00053
      GO TO 00054
00053 CONTINUE
      IF(RLP(1,I).LT.0)GO TO 00055
      GO TO 00056
00055 CONTINUE
      CALL SAMPL(IP(1),I+1,TD)
      CALL SCHED(TIME+TD,-IFIX(RLP(1,I)),1,IP(1))
00056 CONTINUE
      GO TO 00052
00054 CONTINUE
      GO TO 00051
00050 CONTINUE
      WRITE(N01,00058)
00058 FORMAT(/1X,15H      ERROR      /1X,
X      29H602 SCAS  ILLEGAL INTERACTION)
      CALL ERROR
      CONTINUE
      IA(IPOINT+2)=-11
      ICTYPE=-9
      IPOINT=IPOINT
      GO TO 99992
00026 CONTINUE

```

FORTRAN Function Programs

```

      WRITE(N01,00059)
00059 FORMAT(/X,15H      ERROR      /1X,
      X      37HILLEGAL RESPONSE COMMUNICATION (BNNN))
      CALL ERROR
      END
C 701 MZMB 3/16/78
C MEASURE MATERIAL BATCHES
C LP+0 1 RESPONSE POINTER
C 1 1 NUMBER OF MATERIAL TYPES MEASURED
C 2 4 MEASUREMENT SPECIFICATIONS (REPEAT FOR EACH TYPE)
C      INTEGER MATERIAL TYPE CODE
C      MEAN COUNT ERROR
C      MAXIMUM UNIFORM DEVIATION OF COUNT ERROR
C      SYSTEMATIC MASS ERROR
C      MEAN OF RANDOM MASS ERROR
C      STANDARD DEVIATION OF RANDOM MASS ERROR
C 3 3 MEAN TIME TO COMPLETE ALL MEASUREMENTS
C 4 3 MAXIMUM UNIFORM DEVIATION OF MEASUREMENT TIME
C 5 1 TIMING CODE
C 6 1 TIMER FUNCTION
C 7 2 MONITOR FUNCTIONS
C 8 1 DESTINATION FUNCTION
C 9 1 TRANSPORT FUNCTION
C 10 2 FUNCTIONS NOTIFIED AT CONDITION TIMES
C 11 1 DUMMY PARAMETER USED FOR SAVE LOCATION
C LV+0 1 ENTRY COUNT
C 1 1 NUMBER OF BATCHES IN FUNCTION
C 2 1 MAXIMUM NUMBER OF BATCHES CONCURRENTLY MEASURED
C 3 3 TIME MAXIMUM FIRST OCCURRED
C 4 1 LN OF MOST RECENT BATCH IN MEASUREMENT
C LI+0 1
C 1 0
C 2 0
C 3 0
C 4 0
      SUBROUTINE B701
      DIMENSION IPVT(17)
      DIMENSION IA(1)
      COMMON RAC(1)
      EQUIVALENCE (IA(1),RA(1))
      COMMON /SYSTEM/MFROM,MT0,ICTYPE,IPOINT,
      1DUMP,ISECA,ITRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
      1LEC,LFFB,LFFE,L1,LLB,LLFB,LLFE,LMS,LNE,LP,LV,MINNFI,
      1MNSE,NFE,NFL,N11,N12,N01,N02,NREP,NRUN,NSE,TIME,TSTOP
      DATA IPVT/1,1,4,3,3,1,1,2,1,1,2,1,1,1,1,3,1/
      IA(LV)=IA(LV)+1
      IF (ICTYPE.GE.0)GO TO 99991
      GO TO 00026
99991 GO TO (99100,99200,99300,99400,99500,99600,99700,99800,99900)
      X, ICTYPE
99992 RETURN
99100 CALL CREATE(701,12,5,IPVT)

```

FORTRAN Function Programs

```

      ICTYPE=-1
      IPOINT=0
      GO TO 99992
99200 CONTINUE
      IF(IV(1).GT.0)GO TO 00027
      GO TO 00028
00027 CONTINUE
      ASSIGN 00030 TO L00004
      GO TO 00004
00030 CONTINUE
00028 CONTINUE
      CALL DESTRO(IPVT)
      ICTYPE=-2
      IPOINT=0
      GO TO 99992
99300 CALL SET(1,IPVT,INTO,REALD)
      ICTYPE=-3
      IPOINT=0
      GO TO 99992
99400 CONTINUE
      WRITE(NO1,00031)
00031 FORMAT(/1X,15H      ERROR      /1X,
X      43H701 MZMB  ATTEMPT TO SET INITIAL CONDITIONS)
      CALL ERROR
99500 CONTINUE
      WRITE(NO1,00032)IA(LB+2),IA(LV+2),RA(LV+3)
00032 FORMAT(/,9H FUNCTION,14,1X,8H701/MZMB /11X,
X      40HMAXIMUM NUMBER OF BATCHES IN MEASUREMENT,110 /11X,
X      27HTIME MAXIMUM FIRST OCCURRED,11X,E12.4)
      ICTYPE=-5
      IPOINT=0
      GO TO 99992
99600 CONTINUE
      IF(IV(1).GT.0)GO TO 00033
      GO TO 00034
00033 CONTINUE
      ASSIGN 00036 TO L00004
      GO TO 00004
00036 CONTINUE
00034 CONTINUE
      I=I+1
00037 IF((I-(IA(LB+4))).LE.0)GO TO 00038
      GO TO 00039
00038 CONTINUE
      IA(LV-I+1)=IA(LI-I+1)
      GO TO 00037
00039 CONTINUE
      ICTYPE=-6
      IPOINT=0
      GO TO 99992
99700 CALL DUMP(IPVT)

```

FORTRAN Function Programs

```

      ICTYPE=-7
      IPOINT=0
      GO TO 99992
99800 CONTINUE
      IF( ITRACE GE.1)GO TO 00040
      GO TO 00041
00040 CONTINUE
      WRITE(N01,00043)TIME,IA(LB+2)
00043 FORMAT(1X,E10.4,14,2X,5HEVENT)
00041 CONTINUE
      IF( ITRACE GE.2.AND.( ITRACE.EQ. ITRACE/2 2))GO TO 00044
      GO TO 00045
00044 CONTINUE
      WRITE(N02,00047)TIME,IA(LB+2)
00047 FORMAT(1X,E10.4,14,2X,5HEVENT)
      CALL ICHECK
00045 CONTINUE
      CALL CEC(1,1,KODE)
      LN=IA(IPOINT+1)
      LP=IA(LN+3)
      IF( IP(5).EQ.1)GO TO 00048
      GO TO 00049
00048 CONTINUE
      I=1
      T=TIME
      GO TO 00050
00049 CONTINUE
      MFROM=IA(LB+2)
      MTO=IP(6)
      ICTYPE=9
      IPOINT=0
      IA(LP)=1
      GO TO 99992
00051 CONTINUE
      CALL TIMER(IP(5),I,T)
00050 CONTINUE
      IF( I GT.1)GO TO 00052
      GO TO 00053
00052 CONTINUE
      CALL SCHED(T,IA(LB+2),1,LN)
      GO TO 00054
C 753 CONTINUE
      ASSIGN 00055 TO L00001
      GO TO 00001
00055 CONTINUE
      IF( ITRACE GE.1)GO TO 00056
      GO TO 00057
00056 CONTINUE
      WRITE(N01,00059)TIME,IA(LB+2)
00059 FORMAT(1X,E10.4,14,2X,31H701 MZMB MEASUREMENT COMPLETED)
00057 CONTINUE
      IF( ITRACE GE.2.AND.( ITRACE.EQ. ITRACE/2 2))GO TO 00060

```

FORTRAN Function Programs

```

00060 GO TO 00061
00060 CONTINUE
00063 WRITE(N02,00063)TIME,IA(LB+2)
00063 FORMAT(1X,E10.4,14,2X,31H701 MZMB MEASUREMENT COMPLETED)
00061 CALL ICHECK
00061 CONTINUE
00061 ASSIGN 00064 TO L00002
00064 GO TO 00002
00064 CONTINUE
00054 CONTINUE
00054 ICTYPE=-8
00054 IPOINT=0
00054 GO TO 99992
99900 CONTINUE
99900 IF( ITRACE.GE.3)GO TO 00065
99900 GO TO 00066
00065 CONTINUE
00065 WRITE(N01,00065)TIME,IA(LB+2)
00065 FORMAT(1X,E10.4,14,2X,11HINTERACTION)
00066 CONTINUE
00066 IF( ITRACE.GE.4.AND.( ITRACE.EQ ITRACE/2 2))GO TO 00069
00066 GO TO 00070
00069 CONTINUE
00069 WRITE(N02,00072)TIME,IA(LB+2)
00072 FORMAT(1X,E10.4,14,2X,11HINTERACTION)
00072 CALL ICHECK
00070 CONTINUE
00070 IF(IA(IPOINT+2).EQ.1)GO TO 00073
00070 GO TO 00074
00073 CONTINUE
00073 IALP+1)=IPOINT
00073 CALL COPYB(IPOINT,LN)
00073 IALV+1)=IALV+1)+1
00073 IF:IV(1).GT.IV(2))GO TO 00076
00073 GO TO 00077
00076 CONTINUE
00076 IALV+2)=IV(1)
00076 IALV+3)=TIME
00077 CONTINUE
00077 IF(IV(4).LE.0)GO TO 00079
00077 GO TO 00080
00079 CONTINUE
00079 IALN+8)=0
00079 IALN+9)=0
00079 IALV+4)=LN
00079 GO TO 00081
00080 CONTINUE
00080 IALV(4)+9)=LN
00080 IALN+8)=IV(4)
00080 IALN+9)=0
00080 IALV+4)=LN
00081 CONTINUE

```

FORTRAN Function Programs

```

      IF(IP(5).EQ.1)GO TO 00082
      GO TO 00083
00082 CONTINUE
      I=1
      T=TIME
      GO TO 00084
00083 CONTINUE
      MFROM=IA(LB+2)
      MTO=IP(6)
      ICTYPE=9
      IPOINT=0
      IA(LP)=2
      GO TO 99992
00085 CONTINUE
      CALL TIMER(IP(5),I,T)
00084 CONTINUE
      CALL SCHED(T+RUNIF(RP(3),RP(4)),IA(LB+2),I,LN)
      ASSIGN 00085 TO L00003
      GO TO 00003
00086 CONTINUE
      IF(ITRACE.GE.3)GO TO 00087
      GO TO 00088
00087 CONTINUE
      WRITE(NO1,00090)TIME,IA(LB+2)
00090 FORMAT(1X,E10.4,14,2X,29H701 MZMB MEASUREMENT STARTED)
00088 CONTINUE
      IF(ITRACE.GE.4.AND.(ITRACE.EQ.ITRACE/2 2))GO TO 00091
      GO TO 00092
00091 CONTINUE
      WRITE(NO2,00094)TIME,IA(LB+2)
00094 FORMAT(1X,E10.4,14,2X,29H701 MZMB MEASUREMENT STARTED)
      CALL ICHECK
00092 CONTINUE
      IPOINT=IP(11)
      IA(IPOINT+2)=-1
      GO TO 00075
00074 IF(IA(IPOINT+2).EQ.5)GO TO 00095
      GO TO 00096
00095 CONTINUE
      IA(IPOINT+2)=-5
      IA(IPOINT+3)=0
      IA(IPOINT+4)=1V(4)
      IA(IPOINT+5)=1V(1)
      IF(ITRACE.GE.3)GO TO 00097
      GO TO 00098
00097 CONTINUE
      WRITE(NO1,00100)TIME,IA(LB+2)
00100 FORMAT(1X,E10.4,14,2X,38H701 MZMB MATERIAL ACQUISITION ATTEMPT)
00098 CONTINUE
      IF(ITRACE.GE.4.AND.(ITRACE.EQ.ITRACE/2 2))GO TO 00101
      GO TO 00102
00101 CONTINUE

```

FORTRAN Function Programs

```

      WRITE(N02,00104)TIME,IA(LB+2)
00104 FORMAT(1X,E10.4,14,2X,38H701 MZMB MATERIAL ACQUISITION ATTEMPT)
      CALL ICHECK
00102 CONTINUE
      GO TO 00075
00096 CONTINUE
      WRITE(N01,00105)
00105 FORMAT(/1X,15H          ERROR          /1X,
X      29H701 MZMB ILLEGAL INTERACTION)
      CALL ERROR
00075 CONTINUE
      ICTYPE=-9
      IPOINT=IPOINT
      GO TO 99992
C
00001 CONTINUE
      CALL PCOM(ILP(7,1),9,6,0,0,0,0,0,0)
      N=1-(1)
00106 N=N+1
      IF((1) (N-(IP(1))) .LE. 0) GO TO 00107
      GO TO 00108
00107 CONTINUE
      CALL MZUR(LM,IP(2),N,IA(IPOINT+3),IA(IPOINT+4),RA(IPOINT+5)
1      ,IA(IPOINT+6),RA(IPOINT+7))
      I=1-(1)
00109 I=I+1
      IF((1) (I-(IA(IP(7)+3))) .LE. 0) GO TO 00110
      GO TO 00111
00110 CONTINUE
      MFROM=IA(LB+2)
      MTC=ILP(7,1)
      ICTYPE=9
      IA(IPOINT+2)=6
      IA(LP)=3
      GO TO 99992
00112 CONTINUE
      IF(IA(IPOINT+2) .NE. -6) GO TO 00113
      GO TO 00114
00113 CONTINUE
      WRITE(N01,00116)
00116 FORMAT(/1X,15H          ERROR          /1X,
X      38H701 MZMB RESPONSE TO MEASUREMENT DATA)
      CALL ERROR
00114 CONTINUE
      GO TO 00109
00111 CONTINUE
      GO TO 00106
00108 CONTINUE
      CALL FSTORE(IPOINT)
      GO TO L00001, (00055)
C
00002 CONTINUE

```


FORTRAN Function Programs

```

      IF (IV(1).EQ.1)GO TO 00117
      GO TO 00118
00117 CONTINUE
      IA(LV+4)=0
      GO TO 00119
00118 CONTINUE
      IF (IV(4).EQ.LN)GO TO 00120
      GO TO 00121
00120 CONTINUE
      IA(LV+4)=IA(LN+8)
      IF (IA(LN+8).GT.0)GO TO 00123
      GO TO 00124
00123 CONTINUE
      IA(IA(LN+8)+9)=0
00124 CONTINUE
      GO TO 00122
00121 CONTINUE
      IF (IA(LN+8).EQ.0)GO TO 00126
      GO TO 00127
00126 CONTINUE
      IA(IA(LN+9)+8)=0
      GO TO 00128
00127 CONTINUE
      IA(IA(LN+8)+9)=IA(LN+9)
      IA(IA(LN+9)+8)=IA(LN+8)
00128 CONTINUE
00122 CONTINUE
00119 CONTINUE
      ASSIGN 00129 TO L00003
      GO TO 00003
00129 CONTINUE
      IA(LN+8)=0
      IA(LN+9)=0
      IA(LN+4)=IA(LB+2)
      MFROM=IA(LB+2)
      MTO=IP(9)
      ICTYPE=9
      IPOINT=LN
      IA(LN+5)=IP(8)
      IA(L )=4
      GO TO 99992
00130 CONTINUE
      IF (IA(IPOINT+2).NE.-1)GO TO 00131
      GO TO 00132
00131 CONTINUE
      WRITE(N01,00134)
00134 FORMAT(/1X,15H      ERROR      /1X,
X      49H701 M2M) MATERIAL NOT REMOVED AT COMPLETION TIME)
      CALL ERROR
00132 CONTINUE
      CALL FSTORE(IPOINT)
      IA(LV+1)=IA(LV+1)-1

```

FURTRAN Function Programs

```

      GO TO L00002, (00064)
C
00003 CONTINUE
      IF(IP(10).GT.0)GO TO 00135
      GO TO 00136
00135 CONTINUE
      CALL PCON(ILP(10,1),9,9,0,IV(4),IV(1),0,0,0,0)
      I=1-(1)
      I=I+1
00138 IF((1) (1-(IA(IP(10)+3)))LE.0)GO TO 00139
      GO TO 00140
00139 CONTINUE
      MTD=ILP(10,1)
      ICTYPE=9
      IA(LP)=5
      GO TO 99992
00141 CONTINUE
      IF(IA(IP(10)+2).NE.-9)GO TO 00142
      GO TO 00143
00142 CONTINUE
      WRITE(NO1,00145)
00145 FORMAT(/1X,15H      ERROR      /1X,
X      38H701 HZMB RESPONSE TO CONDITION NOTICE)
      CALL ERROR
00143 CONTINUE
      CALL FSTORE(IP(10))
      GO TO 00138
00140 CONTINUE
00136 CONTINUE
      GO TO L00003, (00086,00129)
C
00004 CONTINUE
      LN=IV(4)
00146 IF(LN.GT.0)GO TO 00147
      GO TO 00148
00147 CONTINUE
      CALL FRMD(IA(LN+3))
      CALL FSTORE(LN)
      LN=IA(LN+8)
      GO TO 00146
00148 CONTINUE
      GO TO L00004, (00030,00036)
00026 I00000=IA(LP)
      GO TO (00051,00085,00112,00130,00141), I00000
      END
C 702 CMBA 3/16/78
C COMPUTE MATERIAL BALANCE
C LP+0 1 RESPONSE POINTER
C 1 1 NUMBER OF REPORTS
C 2 4 REPORT SPECIFICATIONS (REPEAT FOR EACH REPORT)
C      INTEGER MATERIAL TYPE CODE
C      DATA: 1 = OBSERVED 2 = TRUE 3 = ERROR

```

FORTRAN Function Programs

```

C      VARIABLE: 1 = COUNT, 2 = MASS
C      SAMPLING: 1 = EACH INPUT/OUTPUT 2 = REPLICATE
C      REPLICATE SAMPLING TIME
C      NUMBER HISTOGRAM CELLS
C      WIDTH OF HISTOGRAM CELLS
C      LOWER LIMIT OF FIRST HISTOGRAM CELL
C      INPUT FUNCTIONS
C      OUTPUT FUNCTIONS
C      POINTERS TO REPORT DATA
C LV=0 1 ENTRY COUNT
C LI=0 0
      SUBROUTINE B702
      DIMENSION IPVT(7)
      DIMENSION IA(1)
      COMMON RA(1)
      EQUIVALENCE (IA(1), RA(1))
      COMMON /SYSTEM/MFROM, MTO, ICTYPE, IPOINT,
      IDUMP, ISECA, ITRACE, LB, LBCA, LBLIST, LBPT, LBUF, LCSB, LE,
      LEC, LFFB, LFFE, LI, LLE, LLFB, LLFE, LMS, LNE, LP, LV, MINNFI,
      IMNSE, NFE, NFL, NI1, NI2, NO1, NO2, NREP, NRUN, NSE, TIME, TSTOP
      DATA IPVT/1,1,4,2,2,1,1/
      IA(LV)=IA(LV)+1
      IF(ICTYPE.GE 0)GO TO 99991
      GO TO 00028
99991 GO TO (99100, 99200, 99300, 99400, 99500, 99600, 99700, 99800, 99900)
      X, ICTYPE
99992 RETURN
99100 CALL CREATE(702, 6, 1, IPVT)
      ICTYPE=-1
      IPOINT=0
      GO TO 99992
99200 CONTINUE
      IF(IP(5).GT.0)GO TO 00027
      GO TO 00028
00027 CONTINUE
      N=1-11)
00030 N=N+1
      IF(1) (N-(IP(1))),LE.0)GO TO 00031
      GO TO 00032
00031 CONTINUE
      CALL FSTORE(IAC(IP(5),N))
      GO TO 00030
00032 CONTINUE
      CALL FSTORE(IP(5))
00028 CONTINUE
      CALL DESTRO(IPVT)
      ICTYPE=-2
      IPOINT=0
      GO TO 99992
99300 CALL SET(1, IPVT, INTD, REALD)
      ICTYPE=-3
      IPOINT=0

```

FORTRAN Function Programs

```

GO TO 99992
99400 CONTINUE
WRITE(NO1,00033)
00033 FORMAT(/1X,15H ERROR /1X,
X 43H702 CMAA ATTEMPT TO SET INITIAL CONDITIONS)
CALL ERROR
CONTINUE
99500 CONTINUE
ASSIGN 00034 TO L00001
GO TO 00001
00034 CONTINUE
ICTYPE=-5
IPOINT=0
GO TO 99992
99600 CONTINUE
N=N+1
00035 N=N+1
IF(1) (N-(IP(1))).LE.0)GO TO 00036
GO TO 00037
00036 CONTINUE
IF(RAC(IP(2),LIINM(N,8,4)).EQ.2.)GO TO 00038
GO TO 00033
00038 CONTINUE
CALL SCHED(RAC(IP(2),LIINM(N,8,5)),IA(LB+2),N,0)
00039 CONTINUE
GO TO 00035
00037 CONTINUE
IA(LV)=0
IF(IP(5).EQ.0)GO TO 00041
GO TO 00042
00041 CONTINUE
ASSIGN 00044 TO L00002
GO TO 00002
00044 CONTINUE
00042 CONTINUE
IF(NRUN.EQ.1)GO TO 00045
GO TO 00046
00045 CONTINUE
ASSIGN 00048 TO L00003
GO TO 00003
00048 CONTINUE
GO TO 00047
00046 CONTINUE
ASSIGN 00049 TO L00004
GO TO 00004
00049 CONTINUE
00047 CONTINUE
ICTYPE=-6
IPOINT=0
GO TO 99992
99700 CALL DUMP(IPVT)
ICTYPE=-7
IPOINT=0

```

FORTRAN Function Programs

```

GO TO 99992
99800 CONTINUE
IF(I TRACE.GE.1)GO TO 00050
GO TO 00051
00050 CONTINUE
WRITE(NO1,00053)TIME,IA(LB+2)
00053 FORMAT(1X,E10.4,14,2X,5HEVENT)
00051 CONTINUE
IF(I TRACE.GE.2.AND.(I TRACE.EQ.I TRACE/2 2))GO TO 00054
GO TO 00055
00054 CONTINUE
WRITE(NO2,00057)TIME,IA(LB+2)
00057 FORMAT(1X,E10.4,14,2X,5HEVENT)
CALL ICHECK
00055 CONTINUE
CALL CEC(1,IP(1),I)
ASSIGN 00058 TO L00005
GO TO 00005
00058 CONTINUE
IF(I TRACE.GE.1)GO TO 00059
GO TO 00060
00059 CONTINUE
WRITE(NO1,00062)TIME,IA(LB+2)
00062 FORMAT(1X,E10.4,14,2X,32H702 CMBA REPLICATE SAMPLE TAKEN)
00060 CONTINUE
IF(I TRACE.GE.2.AND.(I TRACE.EQ.I TRACE/2 2))GO TO 00063
GO TO 00064
00063 CONTINUE
WRITE(NO2,00066)TIME,IA(LB+2)
00066 FORMAT(1X,E10.4,14,2X,32H702 CMBA REPLICATE SAMPLE TAKEN)
CALL ICHECK
00064 CONTINUE
ICTYPE=-8
IPOINT=0
GO TO 99992
99900 CONTINUE
IF(I TRACE.GE.3)GO TO 00067
GO TO 00068
00067 CONTINUE
WRITE(NO1,00070)TIME,IA(LB+2)
00070 FORMAT(1X,E10.4,14,2X,11HINTERACTION)
00068 CONTINUE
IF(I TRACE.GE.4.AND.(I TRACE.EQ.I TRACE/2 2))GO TO 00071
GO TO 00072
00071 CONTINUE
WRITE(NO2,00074)TIME,IA(LB+2)
00074 FORMAT(1X,E10.4,14,2X,11HINTERACTION)
CALL ICHECK
00072 CONTINUE
IF(IA(IPOINT+2).NE.6)GO TO 00075
GO TO 00076
00075 CONTINUE

```

FORTRAN Function Programs

```

      WRITE(N01,00078)
00078 FORMAT(/1X,15H      ERROR      /1X,
X      29H702 CMBA  ILLEGAL INTERACTION)
      CALL ERROR
      CONTINUE
      N=N+1)
00079 N=N+1)
      IF((1) (N-(IP(1))) LE.0) GO TO 00080
      GO TO 00081
00080 CONTINUE
      IF((FIX(RAC(IP(2),L(INM(N,8,1))) EQ.1A(IP(1)+3)) GO TO 00082
      GO TO 00083
00082 CONTINUE
      ASSIGN 00085 TO L00006
      GO TO 00006
00085 CONTINUE
      IF(1 EQ.1) GO TO 00086
      GO TO 00087
00086 CONTINUE
      SIGN=1.
      ASSIGN 00089 TO L00007
      GO TO 00007
00089 CONTINUE
      GO TO 00088
00087 IF(1 EQ.2) GO TO 00090
      GO TO 00091
00090 CONTINUE
      SIGN=-1.
      ASSIGN 00092 TO L00007
      GO TO 00007
00092 CONTINUE
      GO TO 00088
00091 CONTINUE
      WRITE(N01,00093)
00093 FORMAT(/1X,15H      ERROR      /1X,
X      49H702 CMBA  DATA RECEIVED FROM ILLEGAL I/O FUNCTION)
      CALL ERROR
      CONTINUE
      ISAMP=FIX(RAC(IP(2),L(INM(N,8,4)))
      IF(ISAMP EQ.1) GO TO 00094
      GO TO 00095
00094 CONTINUE
      ASSIGN 00097 TO L00005
      GO TO 00005
00097 CONTINUE
      GO TO 00096
00095 IF(ISAMP NE.2) GO TO 00098
      GO TO 00099
00098 CONTINUE
      WRITE(N01,00100)
00100 FORMAT(/1X,15H      ERROR      /1X,
X      36H702 CMBA  ILLEGAL SAMPLING PARAMETER)

```

FORTRAN Function Programs

```

00099 CALL ERROR
00099 CONTINUE
00096 CONTINUE
00083 CONTINUE
00081 GO TO 00079
00081 CONTINUE
      IA(IPPOINT+2)=-6
      IF(I TRACE.GE.3)GO TO 00101
      GO TO 00102
00101 CONTINUE
      WRITE(N01,00104)TIME,IA(LB+2)
00104 FORMAT(1X,E10.4,14,2X,35H702 CMBA MEASUREMENT DATA RECEIVED)
00102 CONTINUE
      IF(I TRACE.GE.4.AND.(I TRACE.EQ.I TRACE/2 2))GO TO 00105
      GO TO 00106
00105 CONTINUE
      WRITE(N02,00108)TIME,IA(LB+2)
00108 FORMAT(1X,E10.4,14,2X,35H702 CMBA MEASUREMENT DATA RECEIVED)
      CALL ICHECK
00106 CONTINUE
      ICTYPE=-9
      IPPOINT=IPPOINT
      GO TO 99992

C
00001 CONTINUE
      N=1-(1)
00109 N=N+1
      IF((1) (N-(IP(1))) LE.0)GO TO 00110
      GO TO 00111
00110 CONTINUE
      L=1A(LCS(IP(5),N))
      RA(LCS(L,3))=RAC(L,3)/FLOAT(IAC(L,1))
      RA(LCS(L,4))=RAC(L,4)/FLOAT(IAC(L,1))-RAC(L,3) 2
      IF(RAC(L,4).LT.0.)GO TO 00112
      GO TO 00113
00112 CONTINUE
      WRITE(N01,00115)
00115 FORMAT(1X,15H ERROR /1X,27H702 CMBA NEGATIVE VARIANCE)
      CALL ERROR
00113 CONTINUE
      RA(LCS(L,4))=SORT(RAC(L,4))
      WRITE(N01,00116)IA(LB+2),N
00116 FORMAT(1H,8HFUNCTION,14,1X,8H702 CMBA,19X,10HREPORT NO.,18)
      I=RAC(IP(2),LIINM(N,8,1))
      WRITE(N01,00117)
00117 FORMAT(11X,13HMATERIAL TYPE,118)
      I=RAC(IP(2),LIINM(N,8,2))
      IF(I.EQ.1)GO TO 00118
      GO TO 00119
00118 CONTINUE
      WRITE(N01,00121)
00121 FORMAT(11X,4HDATA,26X,8HOBSERVED)

```

FORTRAN Function Programs

```

GO TO 00120
00119 IF(I.EQ.2)GO TO 00122
GO TO 00123
00122 CONTINUE
WRITE(N01,00124)
00124 FORMAT(11X,4HDATA,26X,4HTRUE)
GO TO 00120
00123 IF(I.EQ.3)GO TO 00125
GO TO 00126
00125 CONTINUE
WRITE(N01,00127)
00127 FORMAT(11X,4HDATA,26X,23HERROR (OBSERVED - TRUE))
GO TO 00120
00126 CONTINUE
WRITE(N01,00128)
00128 FORMAT(11X,20HILLEGAL DATA SPECIFICATION)
00120 CONTINUE
I=RAC(IP(2),LIINM(N,8,3))
IF(I.EQ.1)GO TO 00129
GO TO 00130
00129 CONTINUE
WRITE(N01,00132)
00132 FORMAT(11X,8HVARIALE,22X,5HCOUNT)
GO TO 00131
00130 IF(I.EQ.2)GO TO 00133
GO TO 00134
00133 CONTINUE
WRITE(N01,00135)
00135 FORMAT(11X,8HVARIALE,22X,4HMASS)
GO TO 00131
00134 CONTINUE
WRITE(N01,00136)
00136 FORMAT(11X,30HILLEGAL VARIABLE SPECIFICATION)
00131 CONTINUE
I=RAC(IP(2),LIINM(N,8,4))
IF(I.EQ.1)GO TO 00137
GO TO 00138
00137 CONTINUE
WRITE(N01,00140)
00140 FORMAT(11X,9HSAMPLING,22X,22HEACH INPUT/OUTPUT TIME)
GO TO 00139
00138 IF(I.EQ.2)GO TO 00141
GO TO 00142
00141 CONTINUE
WRITE(N01,00143)
00143 FORMAT(11X,8HSAMPLING,22X,9HREPLICATE)
GO TO 00139
00142 CONTINUE
WRITE(N01,00144)
00144 FORMAT(11X,30HILLEGAL SAMPLING SPECIFICATION)
00139 CONTINUE
WRITE(N01,00145)RAC(IP(2),LIINM(N,8,5)),IAC(L,1)

```


FORTRAN Function Programs

```

00145 FORMAT(11X,23HREPLICATE SAMPLING TIME,5X,E12.4,8X,
X      22HNUMBER OF OBSERVATIONS,8X,I10)
WRITE(N01,00146)RAC(L,3),RAC(L,4)
00146 FORMAT(11X,10HMEAN VALUE,19X,E11.4,8X,18HSTANDARD DEVIATION,
X      12X,E10.4)
WRITE(N01,00147)RAC(L,5),RAC(L,6)
00147 FORMAT(11X,13HMAXIMUM VALUE,16X,E11.4,8X,
X      21HTIME OF MAXIMUM VALUE,9X,E10.4)
WRITE(N01,00148)RAC(L,7),RAC(L,8)
00148 FORMAT(11X,13HMINIMUM VALUE,16X,E11.4,8X,
X      21HTIME OF MINIMUM VALUE,9X,E10.4)
IF((FIX(RAC(IP(2),LIINM(N,8,6)))>.GT.0)GO TO 00149
GO TO 00150
00149 CONTINUE
WRITE(N01,00152)
00152 FORMAT(11X,11HLOWER LIMIT,26X,7HDISTRI-,4X,3HO.0,9X,
X      4HO.25,8X,3HO.5,9X,4HO.75,8X,3HI.0,11X,
X      7HOF CELL,8X,5HCOUNT,5X,7HDENSITY,5X,6HBTION,6X,
X      49H+
X      X=RAC(IP(2),LIINM(N,8,8))-RAC(IP(2),LIINM(N,8,7))
X      DIS=0.
X      IA(LBUF+30)=1H)
M=1-I11
00153 M=M+1
IF((M-(FIX(RAC(IP(2),LIINM(N,8,6))))>.LE.0)GO TO 00154
GO TO 00155
00154 CONTINUE
DEN=FLOAT(IA(L,8+M))/FLOAT(IA(L,1))
DIS=DIS+DEN
X=X+RAC(IP(2),LIINM(N,8,7))
J=31-(1)
J=J+1
IF((J-(80))>.LE.0)GO TO 00157
GO TO 00158
00157 CONTINUE
IA(LBUF+J)=1H
GO TO 00156
00156 CONTINUE
K=49. DEN+.999
J=1-(1)
J=J+1
IF((J-(K))>.LE.0)GO TO 00160
GO TO 00161
00160 CONTINUE
IA(LBUF+30+J)=1H
GO TO 00159
00161 CONTINUE
K=49. DIS
IA(LBUF+31+K)=1H+
WRITE(N01,00162)X,IA(L,8+M),DEN,DIS,(IA(LBUF+J),J=30,80)
00162 FORMAT(10X,E11.4,I10,5X,F6.4,6X,F6.4,5X,51A1)
GO TO 00153

```

FORTRAN Function Programs

```

00155 CONTINUE
00160 CONTINUE
      GO TO 00109
00111 CONTINUE
      GO TO L00001, (00034)
C
00002 CONTINUE
      CALL ASTORE(1A(LP+5),IP(1))
      N=1-(1)
00163 N=N+1
      IF(1) (N-(IP(1))).LE.0)GO TO 00164
      GO TO 00165
00164 CONTINUE
      CALL ASTORE(1A(LCS(IP(5),N)),8+1FIX(RAC(IP(2),L1:CM(N,8,6)))
      GO TO 00163
00165 CONTINUE
      GO TO L00002, (00044)
C
00003 CONTINUE
      IF(IP(5).LE.0)GO TO 00166
      GO TO 00167
00166 CONTINUE
      WRITE(NO1,00169)
00169 FORMAT(/1X,15H      ERROR      /1X,
X      33H702 CMBA NO REPORT DATA TO RESET)
      CALL ERROR
00167 CONTINUE
      N=1-(1)
00170 N=N+1
      IF(1) (N-(IP(1))).LE.0)GO TO 00171
      GO TO 00172
00171 CONTINUE
      CALL ZIAC(1AC(IP(5),N))
      I=2-(1)
00173 I=I+1
      IF(1) (I-(8)).LE.0)GO TO 00174
      GO TO 00175
00174 CONTINUE
      RA(LCS(1AC(IP(5),N),1))=0.
      GO TO 00173
00175 CONTINUE
      RA(LCS(1AC(IP(5),N),7))=1.0E+50
      GO TO 00170
00172 CONTINUE
      GO TO L00003, (00048)
C
00004 CONTINUE
      N=1-(1)
00176 N=N+1
      IF(1) (N-(IP(1))).LE.0)GO TO 00177
      GO TO 00178
00177 CONTINUE

```

FORTRAN Function Programs

```

      L=IAC(IP(5),N)
      RA(LCS(L,2))=0.
00178 GO TO 00176
      CONTINUE
      GO TO L000G4, (00049)
C
00006 CONTINUE
      K=1
00179 IF(K.LE. IA(IP(3)+3).AND. IAC(IP(3),K+2) NE MFROM)GO TO 00180
      GO TO 00181
00180 CONTINUE
      K=K+1
      GO TO 00179
00181 CONTINUE
      IF(K.LE. IA(IP(3)+3))GO TO 00182
      GO TO 00183
00182 CONTINUE
      I=1
      GO TO 00184
00183 CONTINUE
      K=1
00185 IF(K.LE. IA(IP(4)+3).AND. IAC(IP(4),K+2).NE. MFROM)GO TO 00186
      GO TO 00187
00186 CONTINUE
      K=K+1
      GO TO 00185
00187 CONTINUE
      IF(K.LE. IA(IP(4)+3))GO TO 00188
      GO TO 00189
00188 CONTINUE
      J=2
      GO TO 00190
00189 CONTINUE
      J=0
00190 CONTINUE
00184 CONTINUE
      GO TO L00006, (00085)
C
00005 CONTINUE
      L=IAC(IP(5),N)
      IA(LCS(L,1))=IAC(L,1)+1
      CV=RAC(L,2)
      RA(LCS(L,3))=RAC(L,3)+CV
      RA(LCS(L,4))=RAC(L,4)+CV CV
      IF(CV.GT. RAC(L,5))GO TO 00191
      GO TO 00192
00191 CONTINUE
      RA(LCS(L,5))=CV
      RA(LCS(L,6))=TIME
00192 CONTINUE
      IF(CV.LT. RAC(L,7))GO TO 00194
      GO TO 00195

```

FORTRAN Function Programs

```

00194 CONTINUE
      RA(LCS(L,7))=CV
      RA(LCS(L,8))=TIME
00195 CONTINUE
      IF((IFIX(RAC(IP(2),LIINM(N,8,6)))>.GT.0)GO TO 00197
      GO TO 00198
00197 CONTINUE
      ICELL=IFIX((CV-RAC(IP(2),LIINM(N,8,8)))/
1      RAC(IP(2),LIINM(N,8,7)))+1
      IF(ICELL.LT.1)GO TO 00200
      GO TO 00201
00200 CONTINUE
      ICELL=1
      GO TO 00202
00201 IF(ICELL.GT.IFIX(RAC(IP(2),LIINM(N,8,6))))GO TO 00203
      GO TO 00204
00203 CONTINUE
      ICELL=IFIX(RAC(IP(2),LIINM(N,8,6)))
00204 CONTINUE
00202 CONTINUE
      ICELL=LCS(L,8+ICELL)
      IA(ICELL)=IA(ICELL)+1
00198 CONTINUE
      GO TO L00005, (00058,00097)
C
00007 CONTINUE
      IDATA=IFIX(RAC(IP(2),LIINM(N,8,2)))
      IVAR=IFIX(RAC(IP(2),LIINM(N,8,3)))
      L=IA(IP(5),N)
      IF((IVAR.EQ.1)GO TO 00205
      GO TO 00206
00205 CONTINUE
      IF(IDATA.EQ.1)GO TO 00208
      GO TO 00209
00208 CONTINUE
      RA(LCS(L,2))=RA(LCS(L,2))+SIGN FLOAT(IA(IP(POINT+4))
      GO TO 00210
00209 IF(IDATA.EQ.2)GO TO 00211
      GO TO 00212
00211 CONTINUE
      RA(LCS(L,2))=RA(LCS(L,2))+SIGN FLOAT(IA(IP(POINT+6))
      GO TO 00210
00212 IF(IDATA.EQ.3)GO TO 00213
      GO TO 00214
00213 CONTINUE
      RA(LCS(L,2))=RA(LCS(L,2))+SIGN FLOAT(IA(IP(POINT+4)
1      -IA(IP(POINT+6))
      GO TO 00210
00214 CONTINUE
      WRITE(N01,00215)
00215 FORMAT(/IX,15H      ERROR      /IX,
X      39H702 CMBA  ILLEGAL REPORT DATA PARAMETER)

```

FORTRAN Function Programs

```

00210 CALL ERROR
      CONTINUE
      GO TO 00207
00206 IF(IVAR.EQ.2)GO TO 00216
      GO TO 00217
00216 CONTINUE
      IF(IDATA.EQ.1)GO TO 00218
      GO TO 00219
00218 CONTINUE
      RA(LCS(L,2))=RA(LCS(L,2))+SIGN RA(IPPOINT+5)
      GO TO 00220
00219 IF(IDATA.EQ.2)GO TO 00221
      GO TO 00222
00221 CONTINUE
      RA(LCS(L,2))=RA(LCS(L,2))+SIGN RA(IPPOINT+7)
      GO TO 00220
00222 IF(IDATA.EQ.3)GO TO 00223
      GO TO 00224
00223 CONTINUE
      RA(LCS(L,2))=RA(LCS(L,2))+SIGN (RA(IPPOINT+5)-RA(IPPOINT+7))
      GO TO 00220
00224 CONTINUE
      WRITE(N01,00225)
00225 FORMAT(1X,15H      ERROR      /1X,
X      39H702 CMBA  ILLEGAL REPORT DATA PARAMETER)
      CALL ERROR
      CONTINUE
      GO TO 00207
00217 CONTINUE
      WRITE(N01,00226)
00226 FORMAT(1X,15H      ERROR      /1X,
X      43H702 CMBA  ILLEGAL REPORT VARIABLE PARAMETER)
      CALL ERROR
00207 CONTINUE
      GO TO 00007, (00089,00092)
00026 CONTINUE
      WRITE(N01,00227)
00227 FORMAT(1X,15H      ERROR      /1X,
X      37HILLEGAL RESPONSE COMMUNICATION (BNNN))
      CALL ERROR
      END
C 703 HNTR 3/16/78
C MONITOR STIMULUS TYPE 1
C LP+0 1 RESPONSE POINTER
C 1 4 SENSITIVITY CHARACTERISTICS PAIRS
C 2 4 TIME DELAY DISTRIBUTION TYPE AND STATISTICS
C 3 1 MODE: 1 = RANDOM INTERVAL 2 = STIMULUS TRACKING 3 = LATCHING
C 4 4 DISTRIBUTION TYPE AND STATISTICS OF RANDOM COMPONENT OF
      DETECTION SIGNAL INTERVAL
C 5 1 SIGNAL INDEX
C 6 3 DETECTION SIGNAL AMPLITUDE
C 7 2 DECISION FUNCTIONS

```

FORTRAN Function Programs

```

C      8 3 MEAN TIME BETWEEN SPURIOUS OUTPUTS
C      LV+0 1 ENTRY COUNT
C      LI+0 1 DETECTION SIGNAL STATE
C      1 0
C      1 0
SUBROUTINE B703
  DIMENSION IPVT(11)
  DIMENSION IA(1)
  COMMON RA(1)
  EQUIVALENCE (IA(1),RA(1))
  COMMON /SYSTEM/MFROM, MTO, ICTYPE, IPOINT,
  1 IDUMP, ISECA, ITRACE, LB, LBGA, LBLIST, LBPT, LBUF, LCSB, LE,
  1 IEC, LFFB, LFFE, LI, LLE, LLFB, LLFE, LMS, LNE, LP, LV, MINNFL,
  1 MNSE, NFE, NFL, NI1, NI2, NO1, NO2, NREP, NRUN, NSE, TIME, TSTOP
  DATA IPVT/1,4,4,1,4,1,3,2,3,1,1/
  IA(LV)=IA(LV)+1
  IF(ICTYPE.GE.0)GO TO 99991
  GO TO 00026
99991 GO TO (99100,99200,99300,99400,99500,99600,99700,99800,99900)
  X, ICTYPE
99992 RETURN
  DATA IX,U/1,0./
99100 CALL CREATE(703,9,2,IPVT)
  ICTYPE=-1
  IPOINT=0
  GO TO 99992
99200 CALL DESTRO(IPVT)
  ICTYPE=-2
  IPOINT=0
  GO TO 99992
99300 CALL SET(1,IPVT,INTD,REALD)
  ICTYPE=-3
  IPOINT=0
  GO TO 99992
99400 CONTINUE
  WRITE(NO1,00027)
00027 FORMAT(/1X,15H ERROR /1X,
  X 43H703 MNTR ATTEMPT TO SET INITIAL CONDITIONS)
  CALL ERROR
99500 CONTINUE
  WRITE(NO1,00028)IA(LB+2)
00028 FORMAT(/1X,9H FUNCTION,I4,1X,8H703 MNTR/11X,9HNO REPORT)
  ICTYPE=-5
  IPOINT=0
  GO TO 99992
99600 CONTINUE
  IF(IRP(1).GT.0.)GO TO 00029
  GO TO 99900
00029 CONTINUE
  CALL URN(IX,U)
  T=-RP(8) ALOG(U)
  CALL SCHED(T,IA(LB+2),3,0)

```

FORTRAN Function Programs

```

00030 CO. NUC
      I=1 }
00032 I=I+1
      IF((I) (1-(IA(LB+4))) .LE. 0) GO TO 00033
      GO TO 00034
00033 CONTINUE
      IALV=I+1)=IA(LI-1+1)
      GO TO 00032
00034 CONTINUE
      ICTYPE=-6
      IPOINT=0
      GO TO 99992
99700 CALL DUMP(IPVT)
      ICTYPE=-7
      IPOINT=0
      GO TO 99992
99800 CONTINUE
      IF(ITRACE.GE.1) GO TO 00035
      GO TO 00036
00035 CONTINUE
      WRITE(NO1,00038)TIME,IA(LB+2)
00038 FORMAT(1X,E10.4,14,2X,SHEVENT)
00036 CONTINUE
      IF(ITRACE.GE.2.AND.(ITRACE.EQ.(ITRACE/2 2))) GO TO 00039
      GO TO 00040
00039 CONTINUE
      WRITE(NO2,00042)TIME,IA(LB+2)
00042 FORMAT(1X,E10.4,14,2X,SHEVENT)
      CALL ICHECK
00040 CONTINUE
      CALL CEC(1,2,KODE)
      IF(KODE.EQ.1) GO TO 00043
      GO TO 00044
00043 CONTINUE
      Q=RP(6)
      ASSIGN 00046 TO L00001
      GO TO 00001
00046 CONTINUE
      IALV+1)=1
      GO TO 00045
00044 IF(KODE.EQ.2) GO TO 00047
      GO TO 00048
00047 CONTINUE
      Q=0.
      ASSIGN 00049 TO L00001
      GO TO 00001
00049 CONTINUE
      IALV+1)=0
      GO TO 00045
00048 IF(KODE.EQ.3) GO TO 00050
      GO TO 00051
00050 CONTINUE

```

FORTRAN Function Programs

```

      Q=RP(6)
      ASSIGN 00052 TO L00001
      GO TO 00001
00052 CONTINUE
      IA(LV+1)=1
      IF(1TRACE.GE.1)GO TO 00053
      GO TO 00054
00053 CONTINUE
      WRITE(N01,00056)TIME,IA(LB+2)
00056 FORMAT(1X,E10.4,14,2X,25H703 MNTR SPURIOUS OUTPUT)
00054 CONTINUE
      IF(1TRACE.GE.2.AND.(1TRACE.EQ.1TRACE/2 2))GO TO 00057
      GO TO 00058
00057 CONTINUE
      WRITE(N02,00060)TIME,IA(LB+2)
00060 FORMAT(1X,E10.4,14,2X,25H703 MNTR SPURIOUS OUTPUT)
      CALL ICHECK
00058 CONTINUE
00051 CONTINUE
00045 CONTINUE
      ICTYPE=-8
      IPOINT=0
      GO TO 99992
99900 CONTINUE
      IF(1TRACE.GE.3)GO TO 00061
      GO TO 00062
00061 CONTINUE
      WRITE(N01,00064)TIME,IA(LB+2)
00064 FORMAT(1X,E10.4,14,2X,11HINTERACTION)
00062 CONTINUE
      IF(1TRACE.GE.4.AND.(1TRACE.EQ.1TRACE/2 2))GO TO 00065
      GO TO 00066
00065 CONTINUE
      WRITE(N02,00068)TIME,IA(LB+2)
00068 FORMAT(1X,E10.4,14,2X,11HINTERACTION)
      CALL ICHECK
00066 CONTINUE
      IF(IA(IPOINT+2).EQ.7)GO TO 00069
      GO TO 00070
00069 CONTINUE
      L=IA(IPOINT+3)
      I=IA(IPOINT+4)
      IF(1FIX(RAC(L,I+2)).NE.1)GO TO 00072
      GO TO 00073
00072 CONTINUE
      WRITE(N01,00075)
00075 FORMAT(/1X,15H      ERROR      /1X,
X 31H703 MNTR  ILLEGAL STIMULUS TYPE)
      CALL ERROR
00073 CONTINUE
      J=I+1
      CALL SAMPL(L,I,T1)

```


FORTRAN Function Programs

```

      I=1+NSTATS(IFIX(RAC(L,1+2)))+1
      CALL SAMPL(L,1,T2)
      I=1+NSTATS(IFIX(RAC(L,1+2)))+1
      CALL SAMPL(L,1,A)
      N=1A(IP(1)+3)-1
00076 IF(RLP(1,N).GT.A.AND.N.GE.1)GO TO 00077
      GO TO 00078
00077 CONTINUE
      N=N-2
      GO TO 00076
00078 CONTINUE
      IF(N.GT.0)GO TO 00079
      GO TO 00080
00079 CONTINUE
      PD=RLP(1,N+1)
      GO TO 00081
00080 CONTINUE
      PD=0.
00081 CONTINUE
      CALL URN(IX,U)
      IF(U.LE.PD)GO TO 00082
      GO TO 00083
00082 CONTINUE
      CALL SAMPL(IP(2),1,T3)
      T=TIME+T1+T3
      CALL SCHED(T,1A(LB+2),1,0)
      IF(IP(3).EQ.1)GO TO 00085
      GO TO 00086
00085 CONTINUE
      CALL SAMPL(IP(4),1,T4)
      T=T+T4
      CALL SCHED(T,1A(LB+2),2,0)
      GO TO 00087
00086 IF(IP(3).EQ.2)GO TO 00088
      GO TO 00089
00088 CONTINUE
      CALL SAMPL(IP(4),1,T4)
      T=T+T2+T4
      IF(T.LT.0.)GO TO 00090
      GO TO 00091
00090 CONTINUE
      T=0.
00091 CONTINUE
      CALL SCHED(T,1A(LB+2),2,0)
      GO TO 00087
00089 IF(IP(3).NE.3)GO TO 00093
      GO TO 00094
00093 CONTINUE
      WRITE(NO1,00095)
00095 FORMAT(/IX,15H      ERROR      /IX,22H703 MNTR  ILLEGAL MODE)
      CALL ERROR
00094 CONTINUE

```

FORTRAN Function Programs

```

00087 CONTINUE
00083 CONTINUE
      IA(IPOINT+2) = 7
      GO TO 00071
00070 IF(IA(IPOINT+2).EQ.8.AND. IA(IPOINT+3).EQ.1.AND. IA(IPOINT+4).EQ.
X      0100 TO 00096
      GO TO 00097
00096 CONTINUE
      IF(IV(1).EQ.1)GO TO 00098
      GO TO 00099
00098 CONTINUE
      Q=0.
      ASSIGN 00101 TO L00001
      GO TO 00001
00101 CONTINUE
      IA(LV+1)=0
      GO TO 00100
00099 CONTINUE
      WRITE(N01,00102)
00102 FORMAT(/1X,15H      ERROR      /1X,15H703 MNTR RESET)
      CALL ERROR
00100 CONTINUE
      IA(IPOINT+2)=-8
      GO TO 00071
00087 CONTINUE
      WRITE(N01,00103)
00103 FORMAT(/1X,15H      ERROR      /1X,
X      29H703 MNTR ILLEGAL INTERACTION)
      CALL ERROR
00071 CONTINUE
      ICTYPE=-9
      IPOINT=IPOINT
      GO TO 99992
C
00001 CONTINUE
      CALL PCOM(ILP(7,1),9,10,1,IP(5),0,0,0,0,0)
      RA(IPOINT+5)=0
      K=1-I1)
00104 K=K+1
      IF(I1) (K-(IA(IP(7)+3))).LE.0)GO TO 00105
      GO TO 00106
00105 CONTINUE
      ICTYPE=9
      MTD=ILP(7,K)
      IA(LP)=1
      GO TO 99992
00107 CONTINUE
      IF(IA(IPOINT+2).NE.-10)GO TO 00108
      GO TO 00109
00108 CONTINUE
      WRITE(N01,00111)
00111 FORMAT(/1X,15H      ERROR      /1X,

```

FORTRAN Function Programs

```

X      37H703 MNTR  MONITOR SIGNAL NOT ACCEPTED)
00109 CALL ERROR
      CONTINUE
      GO TO 00104
00106 CONTINUE
      CALL FSTORE(IPPOINT)
      GO TO L00001 (00046,00049,00052,00101)
00026 I00000=IA(LP)
      GO TO 00107
      END
C 801 DTBL 3/16/78
C DECISION TABLE
C LP+0 1 RESPONSE POINTER
C      1 2 INPUT SIGNAL INDEXES
C      2 1 NUMBER OF CONDITION ROWS
C      3 4 CONDITION TRIPLETS
C      4 2 OUTPUT SIGNAL INDEXES
C      5 1 NUMBER OF ACTION ROWS
C      6 4 ACTION PAIRS
C      7 1 NUMBER OF DECISION RULES
C      8 2 DECISION RULES
C      9 2 FUNCTIONS TO RECEIVE OUTPUT SIGNALS
C     10 1 NUMBER OF REPORTS
C     11 4 REPORT SPECIFICATIONS (REPEAT FOR EACH REPORT)
C          SIGNAL INDEX
C          SAMPLING TIME
C          NUMBER OF CELLS
C          WIDTH OF CELLS
C          LOWER LIMIT OF FIRST CELL
C          - POINTERS TO REPORT DATA
C LV+0 1 ENTRY COUNT
C      1 4 INPUT SIGNALS
C      2 4 OUTPUT SIGNALS
C      3 3 TIME OF SCHEDULED DECISION EVENT
C LI+0 1
C      1 0
C      2 0
C      3 -1
      SUBROUTINE B601
      DIMENSION IPVT(17)
      DIMENSION IA(1)
      COMMON RA(1)
      EQUIVALENCE (IA(1),RA(1))
      COMMON /SYSTEM/MFROM, MTO, ICTYPE, IPPOINT,
1IDUMP, ISECA, ITRACE, LB, LBGA, LBLIST, LBPT, LBUF, LCSB, L.E.
1LEC, LFFB, LFFE, LI, LLE, LLFB, LLFE, LMS, LNE, LP, LV, MINNPL,
1MNSE, NFE, NFL, NI1, NI2, NO1, NO2, NREP, NRUN, NSE, TIME, TSTOP
      DATA IPVT/1,2,1,4,2,1,4,1,2,2,1,4,1,1,4,4,3/
      IA(LV)=IA(LV)+1
      IF(ICTYPE.GE.0)GO TO 99991
      GO TO 00026
99991 GO TO (99100,99200,99300,99400,99500,99600,99700,99800,99900)

```

FORTRAN Function Programs

```

X, ICTYPE
99992 RETURN
99100 CALL CREATE(801,13,4,IPVT)
    ICTYPE=-1
    IPOINT=0
    GO TO 99992
99200 CONTINUE
    IF(IP(12).GT.0)GO TO 00027
    GO TO 00028
00027 CONTINUE
    N=1-11
00030 N=N+1
    IF((1) (N-(IP(10))) LE.0)GO TO 00031
    GO TO 00032
00031 CONTINUE
    CALL FSTORE(IAC(IP(12),N))
    GO TO 00030
00032 CONTINUE
    CALL FSTORE(IP(12))
00028 CONTINUE
    CALL DESTRO(IPVT)
    ICTYPE=-2
    IPOINT=0
    GO TO 99992
99300 CALL SET(1,IPVT,INTD,REALD)
    ICTYPE=-3
    IPOINT=0
    GO TO 99992
99400 CALL SET(2,IPVT,INTD,REALD)
    ICTYPE=-4
    IPOINT=0
    GO TO 99992
99500 CONTINUE
    ASSIGN 00033 TO L00001
    GO TO 00001
00033 CONTINUE
    ICTYPE=-5
    IPOINT=0
    GO TO 99992
99600 CONTINUE
    IAC(LV+0)=0
    IF(IP(1) LE.0)GO TO 00034
    GO TO 00035
00034 CONTINUE
    WRITE(NO1,00037)
00037 FORMAT(/1X,15H      ERROR      /1X,
X      51H801 DTBL UNDEFINED INPUT SIGNAL INDEXES AT IC TIME)
    CALL ERROR
    GO TO 00036
00035 CONTINUE
    IF(IV(1) LE.0)GO TO 00038
    GO TO 00039

```

FORTRAN Function Programs

```

00038 CONTINUE
      CALL ASTORE(IA(LV+1),IA(IP(1)+3)+2)
      IA(IV(1)+2)=IA(IV(1)+1)-2
      IA(IV(1)+3)=IA(IP(1)+3)
00039 CONTINUE
00036 CONTINUE
      IF(IIC(1).LE.0)GO TO 00041
      GO TO 00042
00041 CONTINUE
      WRITE(N01,00044)
00044 FORMAT(/1X,15H      ERROR      /1X,
X      46H801 DTBL  UNDEFINED INPUT IC VALUES AT IC TIME)
      CALL ERROR
      GO TO 00043
00042 CONTINUE
      I=I-(1)
00045 I=I+1
      IF(I) (1-(IA(IP(1)+3))) .LE. 0)GO TO 00046
      GO TO 00047
00046 CONTINUE
      RA(LCS(IV(1),I+2))=RA(LCS(IIC(1),I+2))
      GO TO 00045
00047 CONTINUE
00043 CONTINUE
      IF(IP(4).LE.0)GO TO 00048
      GO TO 00049
00048 CONTINUE
      WRITE(N01,00051)
00051 FORMAT(/1X,15H      ERROR      /1X,
X      52H801 DTBL  UNDEFINED OUTPUT SIGNAL INDEXES AT IC TIME)
      CALL ERROR
      GO TO 00050
00049 CONTINUE
      IF(IV(2).LE.0)GO TO 00052
      GO TO 00053
00052 CONTINUE
      CALL ASTORE(IA(LV+2),IA(IP(4)+3)+2)
      IA(IV(2)+2)=IA(IV(2)+1)-2
      IA(IV(2)+3)=IA(IP(4)+3)
00053 CONTINUE
00050 CONTINUE
      IF(IIC(2).LE.0)GO TO 00055
      GO TO 00056
00055 CONTINUE
      WRITE(N01,00058)
00058 FORMAT(/1X,15H      ERROR      /1X,
X      47H801 DTBL  UNDEFINED OUTPUT IC VALUES AT IC TIME)
      CALL ERROR
      GO TO 00057
00056 CONTINUE
      I=I-(1)
00059 I=I+1

```

FORTRAN Function Programs

```

      IF((1) (1-(1A(IP(4)+3))) .LE. 0) GO TO 00060
      GO TO 00061
00060 CONTINUE
      RA(LCS(IP(2),1+2))=RA(LCS(1:IC(2),1+2))
      GO TO 00059
00061 CONTINUE
00057 CONTINUE
      N=N+1
00062 N=N+1
      IF((1) (N-(IP(10))) .LE. 0) GO TO 00063
      GO TO 00064
00063 CONTINUE
      CALL SCHED(RAC(IP(11),L1INH(N,5,2)),1A(LB+2),N,0)
      GO TO 00062
00064 CONTINUE
      IF(IP(12).EQ.0) GO TO 00065
      GO TO 00066
00065 CONTINUE
      ASSIGN 00068 TO L00002
      GO TO 00002
00068 CONTINUE
00066 CONTINUE
      IF(INRUN.EQ.1) GO TO 00069
      GO TO 00070
00069 CONTINUE
      ASSIGN 00072 TO L00003
      GO TO 00003
00072 CONTINUE
      GO TO 00071
00070 CONTINUE
      ASSIGN 00073 TO L00004
      GO TO 00004
00073 CONTINUE
00071 CONTINUE
      RA(LV+3)=-1.
      ICTYPE=-6
      IPINT=0
      GO TO 99992
99700 CALL DUMP(IPVT)
      ICTYPE=-7
      IPINT=0
      GO TO 99992
99800 CONTINUE
      IF(ITRACE.GE.1) GO TO 00074
      GO TO 00075
00074 CONTINUE
      WRITE(NO1,00077)TIME,1A(LB+2)
00077 FORMAT(1X,E10.4,{4,2X,5HEVENT)
00075 CONTINUE
      IF(ITRACE.GE.2.AND.(ITRACE.EQ.ITRACE/2 2))GO TO 00078
      GO TO 00079
00078 CONTINUE

```

FORTRAN Function Programs

```

      WR E(N02,00081) TIME,IA(LB+2)
00081 FORMAT(1X,E10.4,14,2X,5HEVENT)
      CALL 1CHECK
00079 CONTINUE
      CALL CEC(-1,IP(10),N)
      IF(N.EQ.-1)GO TO 00082
      GO TO 00083
00082 CONTINUE
      ASSIGN 00085 TO L00007
      GO TO 00007
00085 CONTINUE
      IF(NEWSTG.EQ.1)GO TO 00086
      GO TO 00087
00086 CONTINUE
      ASSIGN 00089 TO L00008
      GO TO 00008
00089 CONTINUE
00087 CONTINUE
      GO TO 00034
00083 CONTINUE
      ASSIGN 00090 TO L00005
      GO TO 00005
00090 CONTINUE
00084 CONTINUE
      ICTYPE=-8
      IPOINT=0
      GO TO 99992
99900 CONTINUE
      IF(1TRACE.GE.3)GO TO 00091
      GO TO 00092
00091 CONTINUE
      WRITE(N01,00094)TIME,IA(LB+2)
00094 FORMAT(1X,E10.4,14,2X,11HINTERACTION)
00092 CONTINUE
      IF(1TRACE.GE.4.AND.(1TRACE.EQ.1TRACE/2 2))GO TO 00095
      GO TO 00096
00095 CONTINUE
      WRITE(N02,00098)TIME,IA(LB+2)
00098 FORMAT(1X,E10.4,14,2X,11HINTERACTION)
      CALL 1CHECK
00096 CONTINUE
      IF(IA(IPOINT+2).NE.10)GO TO 00099
      GO TO 00100
00099 CONTINUE
      WRITE(N01,00102)
00102 FORMAT(/1X,15H      ERROR      /1X,
X      29H801 DTBL  ILLEGAL INTERACTION)
      CALL ERROR
00100 CONTINUE
      ASSIGN 00103 TO L00006
      GO TO 00006
00103 CONTINUE

```

FORTRAN Function Programs

```

      IF(RV(3).LT.TIME)GO TO 00104
      GO TO 00105
00104 CONTINUE
      CALL SCHED(TIME,IA(LB-2),-1,0)
      RA(LV+3)=TIME
00105 CONTINUE
      IF(ITRACE.GE.3)GO TO 00107
      GO TO 00108
00107 CONTINUE
      WRITE(N01,00110)TIME,IA(LB+2)
00110 FORMAT(1X,E10.4,14,2X,33H01 DTBL CONTROL SIGNAL RECEIVED)
00108 CONTINUE
      IF(ITRACE.GE.4.AND.(ITRACE.EQ.ITRACE/2 2))GO TO 00111
      GO TO 00112
00111 CONTINUE
      WRITE(N01,00113)TIME,IA(LB+2)
00114 FORMAT(1X,E10.4,14,2X,33H01 DTBL CONTROL SIGNAL RECEIVED)
      CALL TCHECK
00112 CONTINUE
      IA(IPCINT+2)=-10
      ICTYPE=-9
      IPOINT=IPCINT
      GO TO 99992
C
00001 CONTINUE
      N=1-11)
00115 N=N+1
      IF((1) (N-(IP(10))) .LE. 0)GO TO 00116
      GO TO 00117
00116 CONTINUE
      L=IA(LCS(IP(12),N))
      RA(LCS(L,3))=RAC(L,3)/FLOAT(IA(L,1))
      RA(LCS(L,4))=RAC(L,4)/FLOAT(IA(L,1))-RAC(L,3) 2
      IF(RAC(L,4).LT.0.)GO TO 00118
      GO TO 00119
00118 CONTINUE
      WRITE(N01,00121)
00121 FORMAT(1X,15H ERROR /1X,27H01 DTBL NEGATIVE VARIANCE)
      CALL ERROR
00119 CONTINUE
      RA(LCS(L,4))=SQRT(RAC(L,4))
      WRITE(N01,00122)IA(LB+2),N
00122 FORMAT(1H,8HFUNCTION,14,1X,8H01 DTBL,19X,1CHREPORT NO.,13)
      I=RAC(IP(11),LINM(N,5,1))
      WRITE(N01,00123)
00123 FORMAT(11X,14HCONTROL SIGNAL,117)
      Y=RAC(IP(11),LINM(N,5,2))
      WRITE(N01,00124)T
00124 FORMAT(11X,13HSAMPLING TIME,15X,E12.4)
      WRITE(N01,00125)RAC(L,3),RAC(L,4)
00125 FORMAT(11X,10HMEAN VALUE,19X,E11.4 ,6X,18HSTANDARD DEVIATION,
X 12X,E10.4)

```


FORTRAN Function Programs

```

      WRITE(N01,00126)RAC(L,5),RAC(L,6)
00126 FORMAT(11X,13HMAXIMUM VALUE,16X,E11.4,8X,
X      21HTIME OF MAXIMUM VALUE,9X,E10.4)
      WRITE(N01,00127)RAC(L,7),RAC(L,8)
00127 FORMAT(11X,13HMINIMUM VALUE,16X,E11.4,8X,
X      21HTIME OF MINIMUM VALUE,9X,E10.4)
      WRITE(N01,00128)IAC(L,1)
00128 FORMAT(11X,22HNUMBER OF OBSERVATIONS,8X,110)
      IF(IFIX(RAC(IP(11),LIINM(N,5,3)))>.0T.0)GO TO 00129
      GO TO 00130
00129 CONTINUE
      WRITE(N01,00132)
00132 FORMAT(11X,11HLOWER LIMIT,26X,7HDISTRI-,4X,3H0.0,9X,
X      4H0.25,8X,3H0.5,9X,4H0.75,8X,3H1.0,11X,
X      7HOF CELL,8X,5HCOUNT,5X,7HDENSITY,5X,6HBUTION,6X,
X      49H+
X      X=RAC(IP(11),LIINM(N,5,5))-RAC(IP(11),LIINM(N,5,4))
X      DIS=0.
X      I(A(LBUF+30))=1H
      M=1-(1)
00133 M=M+1
      IF((1) (M-(IFIX(RAC(IP(11),LIINM(N,5,3))))).LE.0)GO TO 00134
      GO TO 00135
00134 CONTINUE
      DEN=FLOAT(IAC(L,8+M))/FLOAT(IAC(L,1))
      DIS=DIS+DEN
      X=X+RAC(IP(11),LIINM(N,5,4))
      J=31-(1)
      J=J+1
00136 IF((1) (J-(80)).LE.0)GO TO 00137
      GO TO 00138
00137 CONTINUE
      I(A(LBUF+J))=1H
      GO TO 00136
00138 CONTINUE
      K=49. DEN+.999
      J=1-(1)
      J=J+1
00139 IF((1) (J-(K)).LE.0)GO TO 00140
      GO TO 00141
00140 CONTINUE
      I(A(LBUF+30+J))=1H
      GO TO 00139
00141 CONTINUE
      K=49. DIS+.999
      I(A(LBUF+30+K))=1H+
      WRITE(N01,00142)X,IAC(L,8+M),DEN,DIS,(I(A(LBUF+J)),J=30,80)
00142 FORMAT(10X,E11.4,110,5X,F6.4,6X,F6.4,5X,51A1)
      GO TO 00133
00135 CONTINUE
00130 CONTINUE
      GO TO 00115

```

FORTRAN Function Programs

```

00117 CONTINUE
      GO TO L00001, (00033)
C
00002 CONTINUE
      CALL ASTORE(IA(LP+12),IP(10))
      N=1-11)
00143 N=N+1
      IF(1) (N-(IP(10))).LE.0)GO TO 00144
      GO TO 00145
00144 CONTINUE
      CALL ASTORE(IA(LCS(IP(12),N)),8+IFIX(RAC(IP(11),LIINM(N,5,3))))
      GO TO 00143
00145 CONTINUE
      GO TO L00002, (00068)
C
00003 CONTINUE
      IF(IP(12),LE.0)GO TO 00145
      GO TO 00147
00146 CONTINUE
      WRITE(NO1,00149)
00149 FORMAT(/1X,15H      ERROR      /1X,
X      33H801 DTBL NO REPORT DATA TO RESET)
      CALL ERROR
00147 CONTINUE
      N=1-11)
00150 N=N+1
      IF(1) (N-(IP(10))).LE.0)GO TO 00151
      GO TO 00152
00151 CONTINUE
      CALL ZIAC(IAC(IP(12),N))
      I=2-11)
00153 I=I+1
      IF(1) (I-(8)).LE.0)GO TO 00154
      GO TO 00155
00154 CONTINUE
      RA(LCS(IAC(IP(12),N),I))=0.
      GO TO 00153
00155 CONTINUE
      RA(LCS(IAC(IP(12),N),7))=1.0E+50
      GO TO 00150
00152 CONTINUE
      GO TO L00003, (00072)
C
00004 CONTINUE
      N=1-11)
00156 N=N+1
      IF(1) (N-(IP(10))).LE.0)GO TO 00157
      GO TO 00158
00157 CONTINUE
      L=IAC(IP(12),N)
      RA(LCS(L,2))=0.
      GO TO 00156

```

FORTRAN Function Programs

```

00158 CONTINUE
      GO TO L00004, (00073)
C
00005 CONTINUE
      L=IAC(IP(12),N)
      ASSIGN 00159 TO L00015
      GO TO 00015
00159 CONTINUE
      IA(LCS(L,1))=IAC(L,1)+1
      CV=RAC(L,2)
      RA(LCS(L,3))=RAC(L,3)*CV
      RA(LCS(L,4))=RAC(L,4)*CV CV
      IF(CV.GT.RAC(L,5))GO TO 00160
      GO TO 00161
00160 CONTINUE
      RA(LCS(L,5))=CV
      RA(LCS(L,6))=TIME
00161 CONTINUE
      IF(CV.LT.RAC(L,7))GO TO 00163
      GO TO 00164
00163 CONTINUE
      RA(LCS(L,7))=CV
      RA(LCS(L,8))=TIME
00164 CONTINUE
      IF(IFIX(RAC(IP(11),LI:NM(N,5,3))).GT.0)GO TO 00166
      GO TO 00167
00166 CONTINUE
      ICELL=IFIX((CV-RAC(IP(11),LI:NM(N,5,5)))/
1      RAC(IP(11),LI:NM(N,5,4)))+1
      IF(ICELL.LT.1)GO TO 00169
      GO TO 00170
00169 CONTINUE
      ICELL=1
      GO TO 00171
00170 IF(ICELL.GT.IFIX(RAC(IP(11),LI:NM(N,5,3))))GO TO 00172
      GO TO 00173
00172 CONTINUE
      ICELL=IFIX(RAC(IP(11),LI:NM(N,5,3)))
00173 CONTINUE
00171 CONTINUE
      ICELL=LCS(L,8+ICELL)
      IA(ICELL)=IA(ICELL)+1
00167 CONTINUE
      GO TO L00005, (00090)
C
00006 CONTINUE
      L=0
      N=0
      IF(IA(IP(POINT)+3).GT.0)GO TO 00174
      GO TO 00175
00174 CONTINUE
      M=IA(IP(1)+3)

```

FORTRAN Function Programs

```

      I=1
      IF(IA(IPOINT+4).GT.0)GO TO 00177
      GO TO 00178
00177 CONTINUE
00180 IF(I.LE.M.AND.(LP(1,1).NE.IA(IPOINT+4))GO TO 00181
      GO TO 00182
00181 CONTINUE
      I=I+1
      GO TO 00180
00182 CONTINUE
      IF(I.LE.M)GO TO 00183
      GO TO 00184
00183 CONTINUE
      RA(LCS(IV(1),2+1))=RA(IPOINT+5)
      L=1
      N=N+1
00184 CONTINUE
00178 CONTINUE
      IF(IA(IPOINT+6).GT.0)GO TO 00186
      GO TO 00187
00186 CONTINUE
      I=1
00189 IF(I.LE.M.AND.(LP(1,1).NE.IA(IPOINT+6))GO TO 00190
      GO TO 00191
00190 CONTINUE
      I=I+1
      GO TO 00189
00191 CONTINUE
      IF(I.LE.M)GO TO 00192
      GO TO 00193
00192 CONTINUE
      RA(LCS(IV(1),2+1))=RA(IPOINT+7)
      L=1
      N=N+1
00193 CONTINUE
00187 CONTINUE
      IF(N.LT.IA(IPOINT+3))GO TO 00195
      GO TO 00196
00195 CONTINUE
      K=I-1
00198 K=K+1
      IF(1) (K-(IA(IA(IPOINT+8)+3))).LE.0)GO TO 00199
      GO TO 00200
00199 CONTINUE
      I=1
00201 IF(I.LE.M.AND.(LP(1,1).NE.IA(IA(IPOINT+8),K+2))GO TO 00202
      GO TO 00203
00202 CONTINUE
      I=I+1
      GO TO 00201
00203 CONTINUE
      IF(I.LE.M)GO TO 00204

```

FORTRAN Function Programs

```

GO TO 00205
00204 CONTINUE      RA(LCS(IV(1),2+1))=RAC(IA(IPCINT+9),K+2)
                   L=1
                   N=N+1
00205 CONTINUE
GO TO 00198
00200 CONTINUE
00196 CONTINUE
      IF(L.EQ.0)GO TO 00207
00207 CONTINUE
      WRITE(N01,00210)
00210 FORMAT(/1X,15H      ERROR      /1X,
X      44H801 DTBL NO INPUT SIGNALS IN THOSE RECEIVED)
      CALL ERROR
00208 CONTINUE
      IF(N.NE.1A(IPCINT+3))GO TO 00211
      GO TO 00212
00211 CONTINUE
      WRITE(N01,00214)
00214 FORMAT(/1X,15H      ERROR      /1X,
X      44H801 DTBL NUMBER CONTROL SIGNALS TRANSMITTED)
      CALL ERROR
00212 CONTINUE
      GO TO 00176
00175 CONTINUE
      WRITE(N01,00215)
00215 FORMAT(/1X,15H      ERROR      /1X,
X      43H801 DTBL ZERO CONTROL SIGNALS TRANSMITTED)
      CALL ERROR
00176 CONTINUE
      LIM=1A(IV(1)+3)
      IF(ITRACE.GE.1)GO TO 00216
      GO TO 00217
00216 CONTINUE
      WRITE(N01,00219)TIME,1A(LB+2)
00219 FORMAT(1X,E10.4,14,2X,5HINPUT)
      WRITE(N01,00220)(RAC(IV(1),I+2),I=1,LIM)
00220 FORMAT(1X,5E12.4)
00217 CONTINUE
      IF(ITRACE.GE.2.AND.(ITRACE.EQ.ITRACE/2 2))GO TO 00221
      GO TO 00222
00221 CONTINUE
      WRITE(N02,00224)TIME,1A(LB+2)
00224 FORMAT(1X,E10.4,14,2X,5HINPUT)
      WRITE(N02,00225)(RAC(IV(1),I+2),I=1,LIM)
00225 FORMAT(1X,5E12.4)
      CALL ICHECK
00222 CONTINUE
      GO TO 000006, (00103)
C

```

FORTRAN Function Programs

```

00007 CONTINUE
      JSAT=0
      NEWSIG=0
00226 IF(J.LT.IP(7) AND.JSAT.EQ.0)GO TO 00227
      GO TO 00228
00227 CONTINUE
      JSAT=1
      ASSIGN 00229 TO L00009
      GO TO 00009
00229 CONTINUE
      GO TO 00226
00228 CONTINUE
      IF(JSAT.EQ.1)GO TO 00230
      GO TO 00231
00230 CONTINUE
      ASSIGN 00233 TO L00010
      GO TO 00010
00233 CONTINUE
      GO TO 00232
00231 CONTINUE
      WRITE(NO1,00234)
00234 FORMAT(1X,15H      ERROR      /1X
      X 32H80) DTBL NO CONDITION SATISFIED)
00232 CONTINUE
      GO TO L00007, (00085)
C
00008 CONTINUE
      LIM=IA(IV(2)+3)
      IF(ITRACE.GE.1)GO TO 00235
      GO TO 00236
00235 CONTINUE
      WRITE(NO1,00238)TIME,1A(LB+2)
00238 FORMAT(1X,E10.4,14,2X,6HOUTPUT)
      WRITE(NO1,00239)(RAC(IV(2),I+2),I=1,LIM)
00239 FORMAT(1X,5E12.4)
00236 CONTINUE
      IF(ITRACE.GE.2.AND.(ITRACE.EQ.ITRACE/2))GO TO 00240
      GO TO 00241
00240 CONTINUE
      WRITE(NO2,00243)TIME,1A(LB+2)
00243 FORMAT(1X,E10.4,14,2X,6HOUTPUT)
      WRITE(NO2,00244)(RAC(IV(2),I+2),I=1,LIM)
00244 FORMAT(1X,5E12.4)
      CALL ICHECK
00241 CONTINUE
      IF(IP(5).GT.0)GO TO 00245
      GO TO 00246
00245 CONTINUE
      CALL PCOM(ILP(9,1),9,10,1A(IP(4)+3),0,0,0,0,IP(4),IV(2))
      I=1-(1)

```

FORTRAN Function Programs

```

00248 I=I+1
      IF((I) (1-(IA(IP(9)+3))) .LE. 0) GO TO 00249
      GO TO 00250
00249 CONTINUE
      MFROM=IA(LB+2)
      KTO=ILP(9,1)
      ICTYPE=9
      IA(IPPOINT+2)=10
      IA(LP)=1
      GO TO 99992
00251 CONTINUE
      IF(IA(IPPOINT+2), NE, -10) GO TO 00252
      GO TO 00253
00252 CONTINUE
      WRITE(N01,00255)
00255 FORMAT(/1X,15H      ERROR      /1X,
X      36H801 DTBL RESPONSE TO OUTPUT SIGNALS)
      CALL ERROR
00253 CONTINUE
      GO TO 00248
00250 CONTINUE
      CALL FSTORE(IPPOINT)
00246 CONTINUE
      GO TO L00008, (00089)
C
00009 CONTINUE
      I=0
      JSAT=1
00256 IF(1.LT.IP(2).AND.JSAT.EQ.1) GO TO 00257
      GO TO 00258
00257 CONTINUE
      I=I+1
      ASSIGN 00259 TO L00011
      GO TO 00011
00259 CONTINUE
      GO TO 00256
00258 CONTINUE
      GO TO L00009, (00229)
C
00010 CONTINUE
      N=(J-1) (IP(2)+IP(5))+IP(2)
      I=1-(1)
      I=I+1
00260 IF((I) (1-(IP(5))) .LE. 0) GO TO 00261
      GO TO 00262
00261 CONTINUE
      IF(ILP(8,N+1).EQ.1) GO TO 00263
      GO TO 00264
00263 CONTINUE
      ASSIGN 00266 TO L00012
      GO TO 00012
00266 CONTINUE

```

FORTRAN Function Programs

```

00264 CONTINUE
GO TO 00260
00262 CONTINUE
IF(IITRACE.GE 1)GO TO 00267
GO TO 00268
00267 CONTINUE
WRITE(N01,00270)TIME,1A(LB*2)
00270 FORMAT(1X,E10.4,14,2X,4HRULE)
WRITE(N01,00271)J
00271 FORMAT(110)
00268 CONTINUE
IF(IITRACE.GE 2.AND.(ITRACE.EQ.ITRACE/2 2))GO TO 00272
GO TO 00273
00272 CONTINUE
WRITE(N02,00275)TIME,1A(LB*2)
00275 FORMAT(1X,E10.4,14,2X,4HRULE)
WRITE(N02,00276)J
00276 FORMAT(110)
CALL ICHECK
00273 CONTINUE
GO TO L00010, (00233)
C
00011 CONTINUE
K=(J-1) (IP(2)+IP(5))+1
IF(ILP(8,K).EQ.-1)GO TO 00277
GO TO 00278
00277 CONTINUE
ASSIGN 00280 TO L00013
GO TO 00013
00280 CONTINUE
IF(ICON.NE.-1)GO TO 00281
GO TO 00282
00281 CONTINUE
JSAT=0
00282 CONTINUE
GO TO 00279
00278 IF(ILP(8,K).EQ.1)GO TO 00284
GO TO 00285
00284 CONTINUE
ASSIGN 00286 TO L00013
GO TO 00013
00285 CONTINUE
IF(ICON.NE.1)GO TO 00287
GO TO 00288
00287 CONTINUE
JSAT=0
00288 CONTINUE
GO TO 00279
00285 IF(ILP(8,K).NE.0)GO TO 00290
GO TO 00291
00290 CONTINUE
WRITE(N01,00292)

```


FORTRAN Function Programs

```

00292 FORMAT(/1X,15H      ERROR      /1X,
X      3H801 DTBL  ILLEGAL CONDITION ENTRY)
      CALL ERROR
00291 CONTINUE
00279 CONTINUE
      GO TO L00011, (00259)
C
00012 CONTINUE
      L=(I-1)/2+1
      INDX=IFIX(RLP(6,L))
      IF(INDX.GT.0)GO TO 00293
      GO TO 00294
00293 CONTINUE
      K=1
00296 IF(K.LE.1A(IP(4)+3).AND.INDX.NE.ILP(4,K))GO TO 00297
      GO TO 00298
00297 CONTINUE
      K=K+1
      GO TO 00296
00298 CONTINUE
      IF(K.GT.1A(IP(4)+3))GO TO 00299
      GO TO 00300
00299 CONTINUE
      WRITE(NO1,00302)
00302 FORMAT(/1X,15H      ERROR      /1X,
X      5H801 DTBL  ILLEGAL SIGNAL INDEX IN ASSIGNMENT ACTION)
      CALL ERROR
      GO TO 00301
00300 CONTINUE
      RA(LCS(IV(2),K+2))=RLP(6,L+1)
      NEWSIG=1
00301 CONTINUE
      GO TO 00295
00294 IF(INDX.LT.0)GO TO 00303
      GO TO 00304
00303 CONTINUE
      CALL PCOM(-INDX,9,11,0,0,0,0,0,0)
      IF(1TRACE.GE.1)GO TO 00305
      GO TO 00306
00305 CONTINUE
      WRITE(NO1,00308)TIME,1A(LB+2)
00308 FORMAT(1X,E10.4,14,2X,5HSTART)
      WRITE(NO1,00309)MT0
00309 FORMAT(110)
00306 CONTINUE
      IF(1TRACE.GE.2.AND.(1TRACE.EQ.1TRACE/2))GO TO 00310
      GO TO 00311
00310 CONTINUE
      WRITE(NO2,00313)TIME,1A(LB+2)
00313 FORMAT(1X,E10.4,14,2X,5HSTART)
      WRITE(NO2,00314)MT0
00314 FORMAT(110)

```

FORTRAN Function Programs

```

00311 CALL ICHECK
      CONTINUE
      IA(LP)=2
      GO TO 99992
00315 CONTINUE
      IF(IA(IPOINT+2).NE.-1)GO TO 00316
      GO TO 00317
00316 CONTINUE
      WRITE(NC1,00319)
00319 FORMAT(/1X,15H      ERROR      /1X,
X 37H801 DTBL  RESPONSE TO CONTROL MESSAGE)
      CALL ERROR
00317 CONTINUE
      CALL FSTORE(IPOINT)
      GO TO 00295
00304 CONTINUE
      WRITE(NC1,00320)
00320 FORMAT(/1X,15H      ERROR      /1X,
X 43H801 DTBL  ILLEGAL ZERO VALUE IN ACTION PAIR)
      CALL ERROR
00295 CONTINUE
      GO TO L00012, (00266)
C
00013 CONTINUE
      L=(I-1)/3+1
      INDX=IFIX(RLP(3,L))
      ASSIGN 00321 TO L00014
      GO TO 00014
00321 CONTINUE
      SI=V
      L=L+1
      JOP=IFIX(RLP(3,L))
      L=L+1
      IF(JOP.LE.6)GO TO 00322
      GO TO 00323
00322 CONTINUE
      INDX=IFIX(RLP(3,L))
      ASSIGN 00325 TO L00014
      GO TO 00014
00325 CONTINUE
      SK=V
      GO TO 00324
00323 IF(JOP.LE.12)GO TO 00326
      GO TO 00327
00326 CONTINUE
      SK=RLP(3,L)
      GO TO 00324
00327 CONTINUE
      WRITE(NC1,00328)
00328 FORMAT(/1X,15H      ERROR      /1X,
X 50H801 DTBL  ILLEGAL RELATIONAL OPERATOR IN CONDITION)
      CALL ERROR

```

FORTRAN Function Programs

```

00324 CONTINUE
      IF(JOP.GT.6)GO TO 00329
      GO TO 00330
00329 CONTINUE
      JOP=JOP+6
00330 CONTINUE
      IF(JOP.LE.3)GO TO 00332
      GO TO 00333
00332 CONTINUE
      IF(JOP.EQ.1)GO TO 00335
      GO TO 00336
00335 CONTINUE
      IF(S1.EQ.SK)GO TO 00338
      GO TO 00339
00338 CONTINUE
      ICON=1
      GO TO 00340
00339 CONTINUE
      ICON=-1
00340 CONTINUE
      GO TO 00337
00336 IF(JOP.EQ.2)GO TO 00341
      GO TO 00342
00341 CONTINUE
      IF(S1.LT.SK)GO TO 00343
      GO TO 00344
00343 CONTINUE
      ICON=1
      GO TO 00345
00344 CONTINUE
      ICON=-1
00345 CONTINUE
      GO TO 00337
00342 IF(JOP.EQ.3)GO TO 00346
      GO TO 00347
00346 CONTINUE
      IF(S1.GT.SK)GO TO 00348
      GO TO 00349
00348 CONTINUE
      ICON=1
      GO TO 00350
00349 CONTINUE
      ICON=-1
00350 CONTINUE
00347 CONTINUE
00337 CONTINUE
      GO TO 00334
00333 IF(JOP.LE.6)GO TO 00351
      GO TO 00352
00351 CONTINUE
      IF(JOP.EQ.4)GO TO 00353
      GO TO 00354

```

FORTRAN Function Programs

```

00353 CONTINUE
      IF(SI.NE.SK)GO TO 00356
      GO TO 00357
00356 CONTINUE
      ICON=1
      GO TO 00358
00357 CONTINUE
      ICON=-1
00358 CONTINUE
      GO TO 00355
00354 IF(JOP.EQ.5)GO TO 00359
      GO TO 00360
00359 CONTINUE
      IF(SI.LE.SK)GO TO 00361
      GO TO 00362
00361 CONTINUE
      ICON=1
      GO TO 00363
00362 CONTINUE
      ICON=-1
00363 CONTINUE
      GO TO 00355
00360 IF(JOP.EQ.6)GO TO 00364
      GO TO 00365
00364 CONTINUE
      IF(SI.GE.SK)GO TO 00366
      GO TO 00367
00366 CONTINUE
      ICON=1
      GO TO 00368
00367 CONTINUE
      ICON=-1
00368 CONTINUE
00365 CONTINUE
00355 CONTINUE
00352 CONTINUE
00334 CONTINUE
      GO TO L00013, (00280,00286)
C
00014 CONTINUE
      K=1
00369 IF(K.LE.(A(IP(1))+3).AND.INDX.NE.ILP(1,K))GO TO 00370
      GO TO 00371
00370 CONTINUE
      K=K+1
      GO TO 00369
00371 CONTINUE
      IF(K.GT.(A(IP(1))+3))GO TO 00372
      GO TO 00373
00372 CONTINUE
      WRITE(NC1,00375)
00375 FORMAT(1X,15H      ERROR      /1X,

```

FORTRAN Function Programs

```

X      48H801 DTBL  SIGNAL OF CONDITION ENTRY NOT AN INPUT)
CALL  ERROR
GO TO 00374
00373 CONTINUE
      V=RLV(I,K)
00374 CONTINUE
      GO TO 00014, (00321,00325)

C
00015 CONTINUE
      K=1
00376 IF(K.LE.(A(IP(4)+3) AND .ILP(4,K).NE. IFIX(RAC(IP(1)),LINM(N,5,1
X      )))GO TO 00377
      GO TO 00378
00377 CONTINUE
      K=K+1
      GO TO 00376
00378 CONTINUE
      IF(K.GT.(A(IP(4)+3)))GO TO 00379
      GO TO 00380
00379 CONTINUE
      WRITE(M01,00382)
00382 FORMAT(1X,15H      ERROR      /1X
X      54H801 DTBL  SIGNAL OF REPORT SPECIFICATION NOT AN OUTPUT)
CALL  ERROR
GO TO 00381
00380 CONTINUE
      RA(LCS(L,2))=RLV(2,K)
00381 CONTINUE
      GO TO 00015, (00159)
00026 100000=1A(LP)
      GO TO (00251,00315), 100000
      END
C 901 TRCN 3/16/78
C TRACE CONTROL
C LP+0 1 RESPONSE POINTER
C LV+0 1 ENTRY COUNT
C LI+0 0
      SUBROUTINE B901
      DIMENSION IPVT(2)
      DIMENSION IA(1)
      COMMON RA(1)
      EQUIVALENCE (IA(1),RA(1))
      COMMON /SYSTEM/MFROM,MTD,ICTYPE,IPONT,
      11DUMP,ISECA,ITRACE,LS,LBCA,LBLS,LBPT,LBUF,LCSB,LE,
      1LEC,LFFB,LFFE,L1,LLE,LLFB,LLFE,LMS,LNE,LP,LV,MINFL,
      1MNS,NFE,NFL,NT1,NT2,NO1,NO2,NREP,NRUN,NSE,TIME,TSTOP
      DATA IPVT/1,1/
      IA(LV)=IA(LV)+1
      IF(CTYPE.GE.0)GO TO 99991
      GO TO 00325
99991 GO TO (99100,99200,99300,99400,99500,99600,99700,99800,99900)

```

FORTRAN Function Programs

```

X, ICTYPE
99992 RETURN
99100 CALL CREATE(901,1,1,IPVT)
      ICTYPE=-1
      IPOINT=0
      GO TO 99992
99200 CALL DESTROY(IPVT)
      ICTYPE=-2
      IPOINT=0
      GO TO 99992
99300 CALL SET(1,IPVT,INTD,REALD)
      ICTYPE=-3
      IPOINT=0
      GO TO 99992
99400 CONTINUE
      WRITE(N01,00027)
00027 FORMAT(1X,15H      ERROR      /1X,
X      43H901 TRCN ATTEMPT TO SET INITIAL CONDITIONS)
      CALL ERROR
99500 CONTINUE
      WRITE(N01,00028)IA(LB+2)
00028 FORMAT(//,9H FUNCTION,14,1X,8H901 TRCN/11X,9HNO REPORT)
      ICTYPE=-5
      IPOINT=0
      GO TO 99992
99600 CONTINUE
      CALL SCHED(0,IA(LB+2),1,0)
      ICTYPE=-6
      IPOINT=0
      GO TO 99992
99700 CALL DUMP(IPVT)
      ICTYPE=-7
      IPOINT=0
      GO TO 99992
99800 CONTINUE
      IF(ITRACE GE.1)GO TO 00029
      GO TO 00030
00029 CONTINUE
      WRITE(N01,00032)TIME,IA(LB+2)
00032 FORMAT(1X,E10.4,14,2X,5HEVENT)
00030 CONTINUE
      IF(ITRACE GE.2.AND.(ITRACE.EQ.ITRACE'2 2))GO TO 00033
      GO TO 00034
00033 CONTINUE
      WRITE(N02,00036)TIME,IA(LB+2)
00036 FORMAT(1X,E10.4,14,2X,5HEVENT)
      CALL ICHECK
00034 CONTINUE
      CALL CEC(1,1,KODE)
      WRITE(59,00037)TIME
00037 FORMAT(1X,E10.4,2X,11HNEW ITRACE?)
      READ(59,00038)N

```

FORTRAN Function Programs

```

00038 FORMAT(11)
00039 IF (N.LT.D OR N.GT.6)GO TO 00040
00040 CONTINUE
00041 WRITE(59,00042)
00042 FORMAT(20HILLEGAL 1TRACE VALUE)
00043 READ(59,00043)N
00043 FORMAT(11)
00043 GO TO 00039
00041 CONTINUE
1TRACE=N
WRITE(59,00044)1TRACE
00044 FORMAT(13H1TRACE IS NOW,13)
WRITE(59,00045)
00045 FORMAT(1X,24HNEXT TRACE CONTROL TIME?)
READ(59,00046)TNEXT
00046 FORMAT(F80.0)
00047 IF(TNEXT.LE.TIME)GO TO 00048
00047 GO TO 00049
00048 CONTINUE
WRITE(59,00050)TIME
00050 FORMAT(1X,11HTIME IS NOW,E12.4)
READ(59,00051)TNEXT
00051 FORMAT(F80.0)
00051 GO TO 00047
00049 CONTINUE
CALL SCHED(TNEXT,1A(LB+2),1,0)
WRITE(59,00052)TNEXT
00052 FORMAT(1X,23HNEXT TRACE CONTROL TIME,E12.4,
1CTYPE=-8
1POINT=0
GO TO 99992
99900 CONTINUE
IF(1TRACE.GE.3)GO TO 00053
00053 GO TO 00054
00053 CONTINUE
WRITE(N01,00056)TIME,1A(LB+2)
00056 FORMAT(1X,E10.4,14,2X,11HINTERACTION)
00054 CONTINUE
IF(1TRACE.GE.4.AND.(1TRACE.EQ.1TRACE/2))GO TO 00057
00057 GO TO 00058
00057 CONTINUE
WRITE(N02,00060)TIME,1A(LB+2)
00060 FORMAT(1X,E10.4,14,2X,11HINTERACTION)
CALL ICHECK
00058 CONTINUE
WRITE(N01,00061)
00061 FORMAT(1X,15H ERROR /1X,28H901/TRCN ILLEGAL INTERACTION
X
CALL ERROR
1CTYPE=-9
1POINT=0

```

FORTRAN Function Programs

```

GO TO 99992
00026 CONTINUE
WRITE(N01,00062)
00062 FORMAT(/1X,15H      ERROR      /1X,
X      37HILLEGAL RESPONSE COMMUNICATION (BNNN))
CALL ERROR
END
902 RMDS      3/16/78
REPORT MATERIAL DIVERSION STATISTICS
LP+0 1 RESPONSE POINTER
1 2 MAQA FUNCTIONS TABULATED
2 4 TIME REPORT SPECIFICATION
NUMBER OF CELLS
WIDTH OF CELLS
LOWER LIMIT OF FIRST CELL
3 4 DIVERTED MATERIAL REPORT SPECIFICATION
NUMBER OF CELLS
WIDTH OF CELLS
LOWER LIMIT OF FIRST CELL
4 1 - TIME REPORT DATA
5 1 - MATERIAL REPORT DATA
LV+0 1 ENTRY COUNT
LI+0 0
SUBROUTINE 9902
DIMENSION IPVT(7)
DIMENSION IA(1)
COMMON RA(1)
EQUIVALENCE (IA(1),RA(1))
COMMON /SYSTEM/MFROM,MTG,ICTYPE,IPPOINT,
11DUMP,ISECA,ITRACE,LB,LBCA,LBLIST,LBPT,LBUF,LCSB,LE,
1LEC,LFFB,LFFE,L1,LLE,LLFB,LLFE,LMS,LNE,LP,LV,MINNFI,
1MNSI,NFE,NFI,NI1,NI2,N01,N02,NREP,NRUN,NSE,TIME,TSTOP
DATA IPVT/1,2,4,4,1,1,1/
IA(LV)=IA(LV)+1
IF(ICTYPE.GE.0)GO TO 99991
GO TO 00026
99991 GO TO (99100,99200,99300,99400,99500,99600,99700,99800,99900)
X, ICTYPE
99992 RETURN
99100 CALL CREATE(902,6,1,IPVT)
ICTYPE=-1
IPPOINT=0
GO TO 99992
99200 CONTINUE
IF(IP(4).GT.0)GO TO 00027
GO TO 00026
00027 CONTINUE
CALL FSTORE(IP(4))
00026 CONTINUE
IF(IP(5).GT.0)GO TO 00030
GO TO 00031
00030 CONTINUE

```


FORTRAN Function Programs

```

      CALL FSTORE(IP(5))
00031 CONTINUE
      CALL DESTRO(IPVT)
      ICTYPE=-2
      IPOINT=0
      GO TO 99992
99300 CALL SET(1,IPVT,INTD,REALD)
      ICTYPE=-3
      IPOINT=0
      GO TO 99992
99400 CONTINUE
      WRITE(N01,00033)
00033 FORMAT(/1X,15H      ERROR      /1X,
X      43H501 INDA  ATTEMPT TO SET INITIAL CONDITIONS)
      CALL ERROR
99500 CONTINUE
      ASSIGN 00034 TO L00001
      GO TO 00001
00034 CONTINUE
      ICTYPE=-5
      IPOINT=0
      GO TO 99992
99500 CONTINUE
      IA(LV)=0
      IF(IP(4).EQ.0)GO TO 00035
      GO TO 00036
00035 CCNTINUE
      CALL ASTORE(IA(LP+4),8+IFIX(RLP(2,1)))
00036 CONTINUE
      IF(IP(5).EQ.0)GO TO 00038
      GO TO 00039
00038 CONTINUE
      CALL ASTORE(IA(LP+5),8+IFIX(RLP(3,1)))
00039 CONTINUE
      IF(NRUN.EQ.1)GO TO 00041
      GO TO 00042
00041 CONTINUE
      ASSIGN 00044 TO L00003
      GO TO 00003
00044 CONTINUE
      GO TO 00043
00042 CONTINUE
      ASSIGN 00045 TO L00004
      GO TO 00004
00045 CONTINUE
00043 CONTINUE
      ICTYPE=-6
      IPOINT=0
      GO TO 99992
99700 CALL DUMP(IPVT)
      ICTYPE=-7
      IPOINT=0

```

FORTRAN Function Programs

```

GO TO 99992
99800 CONTINUE
IF(TRACE.GE.1)GO TO 00046
GO TO 00047
00046 CONTINUE
WRITE(N01,00049)TIME,IA(LB+2)
00049 FORMAT(1X,E10.4,14,2X,5HEVENT)
00047 CONTINUE
IF(TRACE.GE.2.AND.(TRACE.EQ.(TRACE/2 2))GO TO 00050
GO TO 00051
00050 CONTINUE
WRITE(N02,00053)TIME,IA(LB+2)
00053 FORMAT(1X,E10.4,14,2X,5HEVENT)
CALL 1CHECK
00051 CONTINUE
IF(IA(IPOINT).GT.0)GO TO 00054
GO TO 00055
00054 CONTINUE
I=IA(IPOINT)
L=IA(IPOINT+1)
INXT=I+2+NSTATS(IFIX(RAC(L,I+3)))
IF(IFIX(RAC(L,INXT+2)).GT.0.AND.!NXT.LT.IA(L+3))GO TO 00057
GO TO 00056
00057 CONTINUE
CALL SAMPL(L,INXT+1,TD)
CALL SCHED(TIME+TD,IFIX(RAC(L,INXT+2)),INXT,L)
00058 CONTINUE
ASSIGN 00060 TO L00005
GO TO 00005
00060 CONTINUE
GO TO 00056
00055 CONTINUE
WRITE(N01,00061)
00061 FORMAT(1X,15H      ERROR      /1X,28H902 RMDS  1LLEGAL EVENT CODE
X
CALL ERROR
00056 CONTINUE
ICTYPE=-8
IPOINT=0
GO TO 99992
99900 CONTINUE
IF(TRACE.GE.3)GO TO 00062
GO TO 00063
00062 CONTINUE
WRITE(N01,00065)TIME,IA(LB+2)
00065 FORMAT(1X,E10.4,14,2X,11HINTERACTION)
00063 CONTINUE
IF(TRACE.GE.4.AND.(TRACE.EQ.(TRACE/2 2))GO TO 00066
GO TO 00067
00066 CONTINUE
WRITE(N02,00069)TIME,IA(LB+2)
00069 FORMAT(1X,E10.4,14,2X,11HINTERACTION)

```

FORTRAN Function Programs

```

00067 CALL ICHECK
00067 CONTINUE
WRITE(N01,00070)
00070 FORMAT(/1X,15H ERROR /1X,
X 29H501 INDA ILLEGAL INTERACTION)
CALL ERROR
ICTYPE=-9
IPOINT=0
GO TO 99992
C
00001 CONTINUE
IF(IAC(IP(4),1).GT.0.)GO TO 00071
GO TO 00072
00071 CONTINUE
N=N+1
00074 N=N+1
IF(1) (N-(2)).LE.0)GO TO 00075
GO TO 00076
00075 CONTINUE
L=IP(3+N)
RA(LCS(L,3))=RAC(L,3)/FLOAT(IAC(L,1))
RA(LCS(L,4))=RAC(L,4)/FLOAT(IAC(L,1))-RAC(L,3) 2
IF(RAC(L,4).GT.0.)GO TO 00077
GO TO 00078
00077 CONTINUE RA(LCS(L,4))=SQRT(RAC(L,4))
GO TO 00079
00078 CONTINUE RA(LCS(L,4))=0.
00079 CONTINUE
WRITE(N01,00080)IA(LB+2)
00080 FORMAT(1H,8HFUNCTION,14,1X,8H902 RMSD)
IF(N.EQ.1)GO TO 00081
GO TO 00082
00081 CONTINUE
WRITE(N01,00084)
00084 FORMAT(11X,24HSEQUENCE COMPLETION TIME)
GO TO 00083
00082 IF(N.EQ.2)GO TO 00085
GO TO 00086
00085 CONTINUE
WRITE(N01,00087)
00087 FORMAT(11X,17HDIVERTED MATERIAL)
00086 CONTINUE
00083 CONTINUE
WRITE(N01,00088)RAC(L,3),RAC(L,4)
00088 FORMAT(11X,10HMEAN VALUE,19X,E11.4
X 18HSTANDARD DEVIATION,12X,E10.4) .8X,
IF(N.EQ.1)GO TO 00089
GO TO 00090
00089 CONTINUE
WRITE(N01,00092)RAC(L,5),RAC(L,7)

```

FORTRAN Function Programs

```

00092 FORMAT(11X,13HMAXIMUM VALUE,16X,E11.4/          11X,
X      13HMINIMUM VALUE,16X,E11.4)
GO TO 00091
00090 IF(N.EQ.2)GO TO 00093
GO TO 00094
00093 CONTINUE
WRITE(NO1,00095)RAC(L,5),RAC(L,6)
00095 FORMAT(11X,13HMAXIMUM VALUE,16X,E11.4          ,8X,
X      21HTIME OF MAXIMUM VALUE,9X,E10.4)
WRITE(NO1,00096)RAC(L,7),RAC(L,8)
00096 FORMAT(11X,13HMINIMUM VALUE,16X,E11.4          ,8X,
X      21HTIME OF MINIMUM VALUE,9X,E10.4)
00094 CONTINUE
00091 CONTINUE
WRITE(NO1,00097)IAC(L,1)
00097 FORMAT(11X,22HNUMBER OF OBSERVATIONS,8X,110)
IF(N.EQ.1.AND.(FIX(RLP(2,1)).GT.0)GO TO 00098
GO TO 00099
00098 CONTINUE
ASSIGN 00101 TO L00002
GO TO 00002
00101 CONTINUE
GO TO 00100
00099 IF(N.EQ.2.AND.(FIX(RLP(3,1)).GT.0)GO TO 00102
GO TO 00103
00102 CONTINUE
ASSIGN 00104 TO L00002
GO TO 00002
00104 CONTINUE
00103 CONTINUE
00100 CONTINUE
GO TO 00074
00076 CONTINUE
GO TO 00073
00072 CONTINUE
WRITE(NO1,00105)IAC(LB+2)
00105 FORMAT(//9H FUNCTION,14,1X,8H902 RMDS- /11X,15HNO OBSERVATIONS
X      )
00073 CONTINUE
GO TO L00001, (00034)
C
00002 CONTINUE
WRITE(NO1,00106)
00106 FORMAT(//11X,11HLOWER LIMIT,26X,7HDISTR1- 4X,3HO.0,9X,4HO.25 ,8X,
X      3HO.5,9X,4HO.75,8X,3HO.0 /11X,7HOF CELL,8X,5HCOUNT,5X,
X      7HDENSITY,5X,6HBUTION,6X ,
X      49H+ + + + +
X      IF(N.EQ.1)GO TO 00107
GO TO 00108
00107 CONTINUE
X=RLP(2,3)-RLP(2,2)
GO TO 00109

```

FORTRAN Function Programs

```

00108 IF(N.EQ.2)GO TO 00110
      GO TO 00111
00110 CONTINUE
      X=RLP(3,3)-RLP(3,2)
00111 CONTINUE
00109 CONTINUE
      DIS=0
      IA(LBUF+30)=1H
      M=1-(1)
00112 M=M+1
      IF((1) (M-(IFIX(RLP(1+N,1))))).LE.0)GO TO 00113
      GO TO 00114
00113 CONTINUE
      DEN=FLOAT(IAC(L,8+M))/FLOAT(IAC(L,1))
      DIS=DIS+DEN
      X=X+RLP(1+N,2)
      J=31-(1)
00115 J=J+1
      IF((1) (J-(80)).LE.0)GO TO 00116
      GO TO 00117
00116 CONTINUE
      IA(LBUF+J)=1H
      GO TO 00115
00117 CONTINUE
      K=49. DEN+.999
      J=1-(1)
00118 J=J+1
      IF((1) (J-(K)).LE.0)GO TO 00119
      GO TO 00120
00119 CONTINUE
      IA(LBUF+30+J)=1H
      GO TO 00118
00120 CONTINUE
      K=49. DIS+.999
      IA(LBUF+30+K)=1H+
      WRITE(NO1,00121)X,IAC(L,8+M),DEN,DIS,(IA(LBUF+J),J=30,80)
00121 FORMAT(10X,E11.4,110,5X,F6.4,6X,F6.4,5X,51A1)
      GO TO 00112
00114 CONTINUE
      GO TO L00002, (00101,00104)
C
00003 CONTINUE
      CALL ZIAC(IP(4))
      CALL ZIAC(IP(5))
      J=2-(1)
00122 J=J+1
      IF((1) (J-(8)).LE.0)GO TO 00123
      GO TO 00124
00123 CONTINUE
      RA(LCS(IP(4),J))=0.
      RA(LCS(IP(5),J))=0.
      GO TO 00122

```

FORTRAN Function Programs

```

00124 CONTINUE
RA(LCS(IP(4),7))=1.0E+50
RA(LCS(IP(5),7))=1.0E+50
GO TO L00003, (00044)
C
00004 CONTINUE
RA(LCS(IP(4),2))=0.
RA(LCS(IP(5),2))=0.
GO TO L00004, (00045)
C
00005 CON :NUE
MDO=RAC(L,3)
N=1-11)
00125 IF((1) (N-(2)).LE.0)GO TO 00126
GO TO 00127
00126 CONTINUE
L=IP(3+N)
IF(N.EQ.1)GO TO 00128
GO TO 00129
00128 CONTINUE
RA(LCS(L,2))=TIME
GO TO 00130
00129 IF(N.EQ.2)GO TO 00131
GO TO 00132
00131 CONTINUE
IF(IP(1).GT.0)GO TO 00133
GO TO 00134
00133 CONTINUE
ASSIGN 00136 TO L00006
GO TO 00006
00136 CONTINUE
00134 CONTINUE
00132 CONTINUE
00130 CONTINUE
IA(LCS(L,1))=IAC(L,1)+1
CV=RAC(L,2)
RA(LCS(L,3))=RAC(L,3)+CV
RA(LCS(L,4))=RAC(L,4)+CV CV
IF(CV.GT.RAC(L,5))GO TO 00137
GO TO 00138
00137 CONTINUE
RA(LCS(L,5))=CV
RA(LCS(L,6))=TIME
00138 CONTINUE
IF(CV.LT.RAC(L,7))GO TO 00140
GO TO 00141
00140 CONTINUE
RA(LCS(L,7))=CV
RA(LCS(L,8))=TIME
00141 CONTINUE
IF(IFIX(RLP(1+N,1)).GT.0)GO TO 00143

```

FORTRAN Function Programs

```

GO TO 00144
00143 CONTINUE
      ICELL=IFIX((CV-RLP(1+N,3))/RLP(1+N,2))+1
      IF(ICELL.LT.1)GO TO 00146
      GO TO 00147
00146 CONTINUE      ICELL=1
      GO TO 00148
00147 IF(ICELL.GT.IFIX(RLP(1+N,1)))GO TO 00149
      GO TO 00150
00149 CONTINUE      ICELL=IFIX(RLP(1+N,1))
00150 CONTINUE
00148 CONTINUE      ICELL=LCS(L,8+1CELL)
      IA(ICELL)=IA(ICELL)+1
00144 CONTINUE
      GO TO 00125
00127 CONTINUE
      GO TO L00005, (00060)
C
00006 CONTINUE
      IF(IA(IP(1)+3).LE.0)GO TO 00151
      GO TO 00152
00151 CONTINUE
      WRITE(NO1,00154)
00154 FORMAT(/1X,15H      ERROR      /1X,
X      39H902 RMDS  NO MAGA FUNCTIONS TO TABULATE)
      CALL ERROR
00152 CONTINUE
      CALL PCOM(ILP(1,1),9,3,0,0,0,0,0,0)
      I=1-(1)
00155 I=I+1
      IF((1)-(1-IA(IP(1)+3))).LE.0)GO TO 00156
      GO TO 00157
00156 CONTINUE
      MTO=ILP(1,1)
      IA(IPOINT+2)=3
      IA(LP)=1
      GO TO 99992
00158 CONTINUE
      IF(IA(IPOINT+2).NE.-3)GO TO 00159
      GO TO 00160
00159 CONTINUE
      WRITE(NO1,00162)
00162 FORMAT(/1X,15H      ERROR      /1X,
X      40H902 RMDS  RESPONSE OF TABULATED FUNCTION)
      CALL ERROR
00160 CONTINUE
      LM=IA(IPOINT+3)
      IF(MDO.EQ.1.OR.MDO.EQ.3)GO TO 00163
      GO TO 00164

```

FORTRAN Function Programs

```

00163 CONTINUE
      RA(LCS(L,2))=FLOAT(IA(LM+8))
      GO TO 00165
00164 IF(MDO.EQ.2.OR.MDO.EQ.4)GO TO 00166
      GO TO 00167
00166 CONTINUE
      RA(LCS(L,2))=RA(LM+9)
      GO TO 00165
00167 IF(MDO.EQ.5)GO TO 00168
      GO TO 00169
00168 CONTINUE
      CALL NCBM(LM,NC,BM)
      RA(LCS(L,2))=FLOAT(NC)
      GO TO 00165
00169 CONTINUE
      WRITE(ND1,00170)
00170 FORMAT(/1X,15H      ERROR      /1X,
X      51H902 RMDS  ILLEGAL MATERIAL DIVERSION OBJECTIVE TYPE)
      CALL ERROR
00165 CONTINUE
      GO TO 00155
00157 CONTINUE
      CALL FSTORE(IPOINT)
      GO TO L00006, (00136)
00025 100000=1A(LP)
      GO TO 00158
      END
      SUBROUTINE B103
      RETURN
      END
      SUBROUTINE B104
      RETURN
      END
      SUBROUTINE B105
      RETURN
      END
      SUBROUTINE B106
      RETURN
      END
      SUBROUTINE B107
      RETURN
      END
      SUBROUTINE B108
      RETURN
      END
      SUBROUTINE B109
      RETURN
      END
      SUBROUTINE B110
      RETURN
      END
      SUBROUTINE B201

```


FORTRAN Function Programs

```
RETURN
END
SUBROUTINE B202
RETURN
END
SUBROUTINE B203
RETURN
END
SUBROUTINE B204
RETURN
END
SUBROUTINE B205
RETURN
END
SUBROUTINE B206
RETURN
END
SUBROUTINE B207
RETURN
END
SUBROUTINE B208
RETURN
END
SUBROUTINE B209
RETURN
END
SUBROUTINE B210
RETURN
END
SUBROUTINE B302
RETURN
END
SUBROUTINE B303
RETURN
END
SUBROUTINE B304
RETURN
END
SUBROUTINE B305
RETURN
END
SUBROUTINE B306
RETURN
END
SUBROUTINE B307
RETURN
END
SUBROUTINE B308
RETURN
END
SUBROUTINE B309
RETURN
```

FORTRAN Function Programs

```
END
SUBROUTINE B310
RETURN
END
SUBROUTINE B402
RETURN
END
SUBROUTINE B403
RETURN
END
SUBROUTINE B404
RETURN
END
SUBROUTINE B405
RETURN
END
SUBROUTINE B406
RETURN
END
SUBROUTINE B407
RETURN
END
SUBROUTINE B408
RETURN
END
SUBROUTINE B409
RETURN
END
SUBROUTINE B410
RETURN
END
SUBROUTINE B506
RETURN
END
SUBROUTINE B507
RETURN
END
SUBROUTINE B508
RETURN
END
SUBROUTINE B509
RETURN
END
SUBROUTINE B510
RETURN
END
SUBROUTINE B603
RETURN
END
SUBROUTINE B604
RETURN
END
```

FORTRAN Function Programs

```
SUBROUTINE B605
RETURN
END
SUBROUTINE B606
RETURN
END
SUBROUTINE B607
RETURN
END
SUBROUTINE B608
RETURN
END
SUBROUTINE B609
RETURN
END
SUBROUTINE B610
RETURN
END
SUBROUTINE B704
RETURN
END
SUBROUTINE B705
RETURN
END
SUBROUTINE B706
RETURN
END
SUBROUTINE B707
RETURN
END
SUBROUTINE B708
RETURN
END
SUBROUTINE B709
RETURN
END
SUBROUTINE B710
RETURN
END
SUBROUTINE B802
RETURN
END
SUBROUTINE B803
RETURN
END
SUBROUTINE B804
RETURN
END
SUBROUTINE B805
RETURN
END
SUBROUTINE B806
```

FORTRAN: Function Programs

```
RETURN
END
SUBROUTINE B807
RETURN
END
SUBROUTINE B808
RETURN
END
SUBROUTINE B809
RETURN
END
SUBROUTINE B810
RETURN
END
SUBROUTINE B903
RETURN
END
SUBROUTINE B904
RETURN
END
SUBROUTINE B905
RETURN
END
SUBROUTINE B906
RETURN
END
SUBROUTINE B907
RETURN
END
SUBROUTINE B908
RETURN
END
SUBROUTINE B909
RETURN
END
SUBROUTINE B910
RETURN
END
```