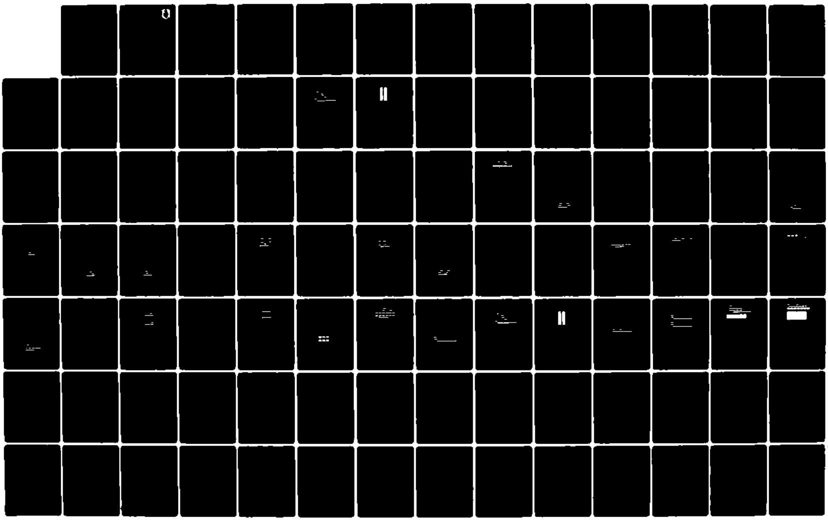
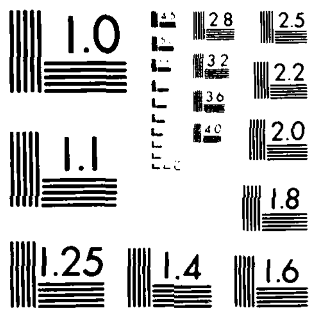
AD-A135 882 AISIM (AUTOMATED INTERACTIVE SIMULATION MODEL) TRAINING 1/2
MANUAL(U) HUGHES AIRCRAFT CO FULLERTON CA GROUND
SYSTEMS GROUP J HEARNE ET AL. 26 FEB 82 ESD-TR-83-217
UNCLASSIFIED F19628-79-C-0153 F/G 9/2 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

ESD-TR-83-217

AD-A135 882

(2)

ELECTRONIC SYSTEMS DIVISION

AISIM TRAINING MANUAL

J. Hearne
S. Kneeburg

Hughes Aircraft Company
Ground Systems Group
Box 3310
Fullerton, California 92634

26 February 1982

Prepared for

ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
DEPUTY FOR ACQUISITION LOGISTICS
AND TECHNICAL OPERATIONS
HANSCOM AFB, MASSACHUSETTS 01731

LEGAL NOTICE

OTHER NOTICES

Review and Approval

This technical report has been reviewed and is approved for publication.

William J. Letendre

WILLIAM J. LETENDRE
Program Manager, Computer Resouce
Management Technology (PE 64740F)

Arthur G. Decelles

ARTHUR G. DECELLES, CAPT, USAF
Project Officer, Requirements
Analysis

FOR THE COMMANDER

Walter W. Turgiss

WALTER W. TURGISS
Director, Engineering and Test
Deputy for Acquisition Logistics
and Technical Operations

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ESD-TR-83-217	2. GOVT ACCESSION NO. AD-A135886	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) AISIM Training Manual		5. TYPE OF REPORT & PERIOD COVERED CDRL No. 119 Final
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) J. Hearne S. Kneeburg		8. CONTRACT OR GRANT NUMBER(s) F19628-79-C-0153
9. PERFORMING ORGANIZATION NAME AND ADDRESS Hughes Aircraft Company Ground Systems Group Box 3310 Fullerton, CA 92634		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Hq Electronic Systems Division (ALEE) Hanscom AFB Massachusetts 01731		12. REPORT DATE 26 February 1982
		13. NUMBER OF PAGES 164
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Same		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

AISIM	Analysis	Architecture
Training	Modeling	Simulation
Design	Processes	

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This document is the Training Manual for the Hughes developed Automated Interactive Simulation Model (AISIM). This manual provides step by step information necessary to begin using AISIM.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

CONTENTS

FIGURES

1. INTRODUCTION

This training manual presupposes virtually no programming experience and is intended to provide step by step information necessary to begin using AISIM. It is not intended as a complete account of the system and many topics covered in the companion AISIM User's Manual are covered here in less detail, or not at all. For further details on the operation of AISIM or on the kind of simulation AISIM is adapted to, the reader is referred to the more detailed AISIM User's Manual.

This manual consists of seven sections. This first section provides a brief overview of AISIM and its main concepts. Sections 2, 3 and 4 concern the Design User Interface (DUI), i.e., that part of AISIM in which models of engineering systems are created. Section 5 describes the complete construction of a simple model. Section 6 turns to the use of the Analysis User Interface--the part of AISIM where simulation and analysis occur--using the model developed in Section 5 as an example. Finally, Chapter 7 will document the modeling, simulation and analysis of a more complex engineering system.

1.1 MODELING

A computer model is a description of a system that is developed as a basis for calculations, predictions or further investigation. In addition, AISIM is especially designed to model systems that incorporate parallel processing. The purpose of an AISIM model is to give information on the workability of a system design, especially by providing statistics that serve both to predict the operation of the modeled system if implemented, and to suggest design modifications.

Modeling is accomplished in AISIM by representing the elements of the system being modeled by AISIM "entities." A detailed description of each entity is provided in Section 3 of the AISIM User's Manual. A general introduction to the types of system elements modeled by these AISIM entities is contained in section 1.3 of this manual.

1.2 OVERVIEW OF THE AISIM SYSTEM

AISIM consists of five subsystems, each of which performs a distinct service. These subsystems are (1) the Design User Interface (DUI); (2) the Analyze User Interface (AUI); (3) the Replot User Interface (RUI); (4) the Hardcopy User Interface (HUI); and (5) the Library User Interface (LUI). Each of these subsystems is briefly described below.

(1) DESIGN USER INTERFACE

The DUI is the facility which enables the user to create or alter models of systems. It contains two sublevels, the Architecture Design Editor (ADE) and the Process Editor Interface (PEI). The ADE is used to construct models of the physical layout of the given system, which is called the architecture. The PEI is used to define the processes or logic that are associated with that architecture. Other model entities are defined at the DUI level.

(2) ANALYZE USER INTERFACE

With the AUI one subjects the model defined in the DUI mode to simulation runs that test the behavior and response of the modeled system to various hypothetical conditions. In this mode statistics are gathered on the operation of the system in simulation and, if desired, graphs of selected parameters are generated (plotted).

(3) REPLOT USER INTERFACE

The REPLOT facility enables the user to plot and compare the statistics from various executions of a model.

(4) HCOPY USER INTERFACE

The Hardcopy mode provides the connection between the AISIM system and a printing device. Process flow-charts constructed in the DUI are printed on an HP2631G printer/plotter.

(5) LIBRARY USER INTERFACE

In the LUI the user is able to break apart and recombine parts of AISIM models, and obtain parts of models from a central system library. This feature is provided because some model components are used in other models and it is sometimes useful to store entire models for later reuse.

1.3 OVERVIEW OF AISIM MODELING CONSTRUCTS

This section provides a brief description of AISIM modeling constructs, to be followed by a more precise description of them in subsequent sections.

With some qualifications, AISIM's modeling constructs can be divided into the following four categories: (1) those used to represent the operations, properties, structure and internal relations of the modeled system itself; (2) those used to represent the environmental stimuli to which the system model is exposed; (3) those which represent the physical layout of the system; and (4) those which represent and facilitate mathematical operations.

1.3.1 ENTITIES REPRESENTING ELEMENTS EXTERNAL TO THE MODELED SYSTEM

1.3.1.1 The Load Entity The Load entity is used to represent aspects of the world outside the modeled system that trigger processes within it. The nature of these triggering stimuli are not dealt with in AISIM; rather, Loads are defined by specifying the nodes at which certain Processes are to take place within a given period (see Scenario), together with specifications of several parameters which indicate the schedule that the Process triggering follows. The definition of a Load will also assign a priority to each of the Processes to be triggered which will determine the priority with which Processes are to be executed in case the same Resource is demanded by several Processes at the same time or in overlapping times.

1.3.1.2 The Scenario Entity A Scenario is used to represent the external demand on a system (i.e., Process triggerings from the outside) throughout a simulation exercise. The Scenario divides a simulation run into a number of periods that determine the frequency with which Loads will be initiated. They will also trigger Processes in a way that is not systematically related to the Loads in order to represent abnormal impositions on the system.

1.3.2 ENTITIES REPRESENTING ELEMENTS INTERNAL TO THE MODELED SYSTEM

1.3.2.1 The Process Entity A Process is used to represent the operations, decisions, actions or activities that can be decomposed and defined in terms of more fundamental AISIM entities, called Primitives. A Process can take place in one or more of the system's nodes (or may execute independent of the nodes) and can make use of one or more Resources.

1.3.2.1.1 The Process Primitives Primitives, of which there are 25, are the elements of which Processes are composed. A Process may be considered to be a collection of Primitives whose sequential execution describes the logic of the Process.

The 25 Primitives can be arranged into nine categories according to similarity of function. For the present, rather than give the meaning of each Primitive individually, it is sufficient to describe the categories and in a general way characterize the roles that members of each will play in the definition of a Process.

1. INTERNAL PROCESS EXECUTION CONTROL. The Primitives,

   COMPARE
   BRANCH
   ENTRY

PROB
LOOP

serve as a "framework" for Processes, enabling the Process to branch (either unconditionally or under certain conditions) to another portion of the Process, or to repeat certain segments of the Process a specified number of times (or until a certain condition is met).

2. RESOURCE ALLOCATION. As mentioned earlier, a Process frequently competes with others for Resources. The Primitives,

ALLOC
DEALLOC
RESET
TEST
LOCK
UNLOCK

govern the allocation of Resources among the various compe[illegible] Processes.

3. PROCESS EXECUTION CONTROL. Since a principal fea[illegible]e of AISIM is its capacity to model parallel Processing, i.e., [illegible]tinct Processes executing at the same time, these Primitiv[illegible] govern the timing of various Processes in the system relat[illegible] to one another. The Primitives,

CALL
SEND
SUSPEND
RESUME
WAIT

will either interrupt the Process in which they stand, or trigger, re-initiate or interrupt some other Process.

5. QUEUE HANDLING. The Primitives,

FILE
FIND
REMOVE

govern the placement and retrieval of Items in Queues that have been defined by the user.

6. ITEM HANDLING. The Primitives,

CREATE
DESTROY

govern the introduction and elimination of a system's transient data elements.

7. VARIABLE MANIPULATION. The Primitives,

ASSIGN
EVAL

assign values to variables (both numerical and non-numerical) and allow for the mathematical manipulation of numerical ones.

8. TIME SEQUENCING. The Primitive,

ACTION

which is associated with the Action entity described below is included in Process definitions to indicate the time a certain Action (or Process, decision, etc.) takes up without further describing the Action's nature.

9. DEBUGGING. The Primitive,

TRACE

is not used to represent a system's operations, but is rather provided as a convenience to the user in the task of tracing a history of Process execution during simulations as a debugging facility.

1.3.2.2 The Resource Entity A Resource represents a component of the modeled system which may be necessary to the execution of a Process. Typically, a Resource will be required for more than one Process. Where several Processes demand a Resource that can serve only one Process at a time all but one will stand in a queue until the Resource is available for them. The order in which the Processes will make use of the contended Resource is a function of the priorities that have been assigned to the Processes in question as well as of the internal structure of the Processes as defined by their Primitives.

1.3.2.3 The Action Entity The Action entity is used to represent any action, activity, decision, etc. that consumes time.

1.3.2.4 The Legal Path Table The Legal Path Table (LPT) is a set of routes or paths between nodes in the system's architecture. The LPT is selected from all the possible paths between the nodes along the links, so that there is but one permissible routing of communication between the various nodes in the architecture. The LPT is accessed by several other elements of AISIM such as the EVAL Primitive, the keywords $NODE, $NXTNODE, and $LINK, and the Message Routing Submodel Processes.

1.3.2.5 The Queue Entity A Queue represents any holding area, such as a memory buffer or job queue, for elements waiting to take up their role in the operation of the system. User-defined Queues can be used as a holding area for Items as well as Resources. A user-defined Queue can be manipulated in a number of ways described later and in the AISIM User's Manual.

1.3.3 ENTITIES WHICH SUPPORT MATHEMATICAL OPERATIONS

1.3.3.1 The Constant Entity A Constant is an entity whose value does not change during a simulation run. Constants are specified or altered in the DUI and can be edited before a simulation run in the AUI but cannot be changed (and do not change) once the execution of a model has begun. Several parameters required in the definition of an AISIM model, (i.e., the length of a simulation run, the number of Resource units, the period length of the simulation and the sizes of Queue) can only take Constants or simple numerics as values.

1.3.3.2 The Variable Entity Variables, by contrast, are entities whose values can change during the exercise of a model. The majority of the parameters in the specification of a model can take Variables as values.

1.3.3.3 The Table Entity Tables are single-value, single-argument functions defined by the user. They may be defined either as discrete, continuous, or alphanumeric and may have from 1 to 15 entries. Tables are invoked by the EVAL Primitive and serve as a supplement to the mathematical operations automatically available as part of the EVAL primitive.

2. CREATING SYSTEM ARCHITECTURES

With the basic understanding of AISIM modeling concepts presented in the previous section, the reader should now be able to interact with the DUI. The exercises here are intended both to deepen the user's grasp of AISIM modeling constructs and to familiarize him with the prompts encountered while interacting with the computer. In general, it is not a good idea to begin the design of an AISIM model without having done research and preparation on paper beforehand. However, as a teaching device, we shall develop fragments of an architecture from requirements formulated as we go along.

The method of logging on is computer-specific so we shall assume that the user has reached the point at which the computer prompts him with,

```
READY
```

which indicates that one is logged on. To obtain access to AISIM, type,

```
EXECUTE AISIM (cr)*
```

You will be offered a collection of information that looks something like that depicted in Figure 1

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
        This is AISIM PRODUCTION VERSION 2.0, which was built from
                          AISIM VERSION 1.1.
                               2/5/82

        *** report any problem to: HERMAN SCHULTZ x-2308 ***
        ***                                              ***
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

Figure 1. Typical Display upon Entering the AISIM READY Level.

and then prompted with,

```
AISIM READY
```

0.

*Hereafter, "(cr)" will indicate a carriage return.

To enter the DUI, type,

d p(test) (cr)

where "test" is the name of the model to be designed.

The user will be prompted with information that looks something like that shown in Figure 2.

```
AISIM READY
d p(test)
CURRENT PARAMETERS IN EFFECT:
VERSION:        PRODUCTION
PROJECT:        TEST
USER:           TF01508
ENTER YES TO PROCEED, NO TO ABORT...
```

Figure 2. Typical Information on Entering the DUI

By typing,

NO (cr)

one will return to the AISIM READY level. Typing,

YES (cr)

will put one in the DUI sublevel and the screen will display a "*" to indicate that you may enter DUI commands.

When the computer displays the prompt "*", enter the Architecture Design Editor (ADE) by typing,

ARCH (cr)

A grid like that in Figure 3 will appear on the screen.

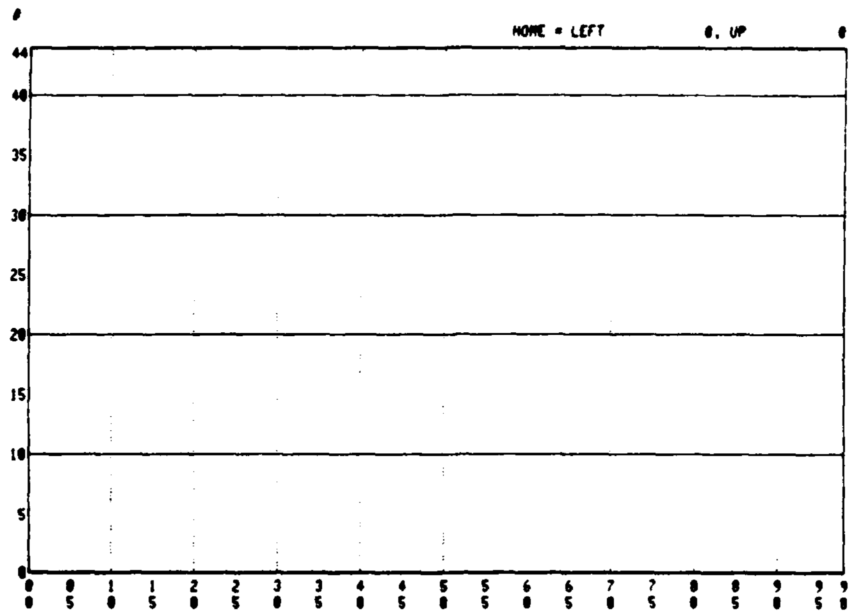


Figure 3. Grid on Which an Architecture is Designed

The AISIM constructs manipulated in the ADE are nodes which represent the hardware elements of a system and links which represent lines of communication between them. The physical layout of the system is represented by picturing nodes and links on the grid to represent various hardware elements of a system and their (possible) lines of communication. A Resource modeling entity is automatically associated with each node or link when it is placed in the architecture.

As a mnemonic aid in distinguishing system elements, AISIM provides fourteen geometrical symbols for nodes. The symbols are called by the three-letter designations given in Figure 4.

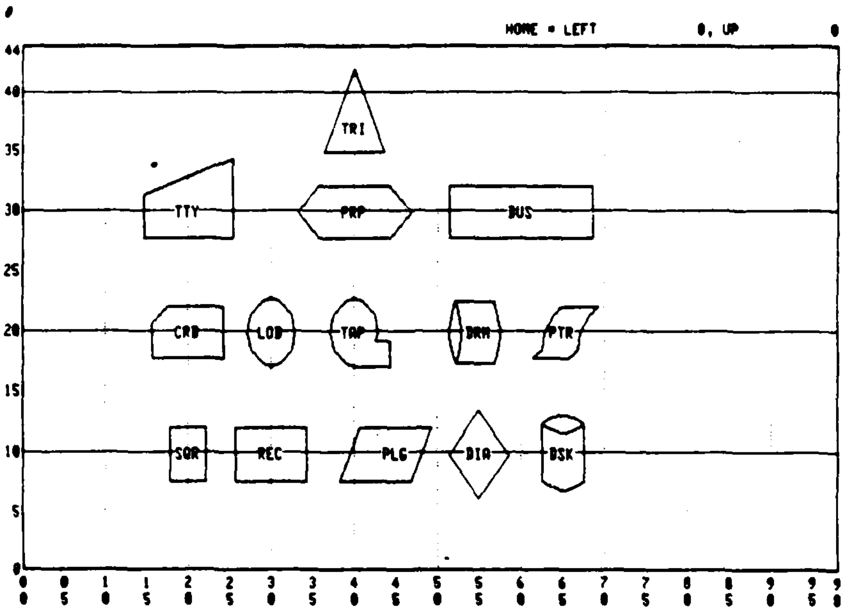


Figure 4. Designations of the Fourteen Symbols

With two exceptions these node symbols differ from one another only in their appearance. The two exceptions are the so-called "leaf-nodes" tty and lod. These nodes may be connected to the modeled architecture by only one link. All other nodes may be connected to any number of other nodes through any number of links. The rationale for this restriction is explained in the AISIM User's Manual, Section 6.3.3.

2.1 DEFINING ATTRIBUTES FOR SYMBOLS

As mentioned earlier, when a symbol is placed in the architecture, an AISIM entity called a Resource is created to represent the hardware element depicted by the node or link. Resources have a number of attributes; some are named system attributes, others are user named and defined attributes. The DEFINE SYMBOL command allows the user to establish attributes to be associated with each symbol type so that when placed in the architecture, the associated Resource will be created with all attributes defined. An example of this use of the DEFINE SYMBOL command is produced by typing:

DEFINE SQR (cr)

(SQR could be replaced with any of the 14 symbol mnemonics or the mnemonic CON if a link is being defined.) The user will be prompted by a form as shown in Figure 5.

RESOURCE NAME:

TOTAL NUMBER OF UNITS:

INITIAL NUMBER OF UNITS:

ATTRIBUTES PRESENT (YES OR NO)

COST:

DESCRIPTION:

Figure 5. First Symbol Definition Form

The user can tab or space through this form using the tab key or space bar of the HP2647A terminal. Any values in the inverse video fields of the form are default values supplied from the AISIM design data base. The user may change these fields by positioning the cursor as described above and then typing over the existing values. The form with the new values can be entered into the data base by striking the "ENTER" key of the HP2647A terminal. The user will then be given another form as shown in Figure 6. This form is blank. The user may enter up to fifteen attribute names and related values of his choice into these fields. For example, when defining attributes of a symbol type which is to represent a disk in the modeled system, the attribute names may be something like seek time, latency, etc., and the values would be the corresponding values for the particular disk being modeled.

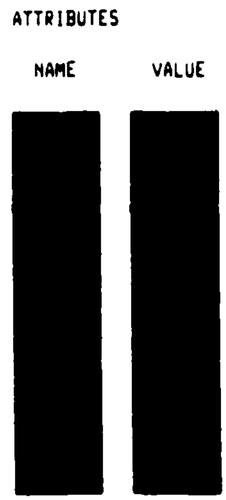


Figure 6. Second Symbol Definition Form

A second form of the DEFINE SYMBOL command takes the form:

DEFINE SYMBOL,RESOURCE NAME (cr)

where SYMBOL is one of the symbol mnemonics or CON and RESOURCE NAME is the name of an existing Resource entity. This command will only be accepted if a Resource entity has been previously defined before entering the ADE. Since the user has not defined any Resource entities in his test data base yet, this command would fail. The trainee might want to try this command later.

If a named Resource entity did exist, forms similar to the forms shown above would be displayed. Instead of the default attributes in the first form and the blank second form, the forms displayed could have the names and values of any attributes previously defined by the Resource entity referenced in the command.

2.2 PLACING NODES ON THE GRID

To place a node at a certain location on the grid--i.e., centered on that location--issue a command designating (1) the type of node to be placed, (2) a user-given name, and (3) horizontal and vertical position coordinates. One can also opt to indicate the size of the geometrical shape if the default value, equal to the number of characters in the user-given name, is unsuitable. To center a square named NODE1 twenty units from the left-hand side and thirty units from the bottom in Figure 4 above, type,

P SQR,NODE1,20,30 (cr)

Figure 7 shows the screen display that would result from this command.

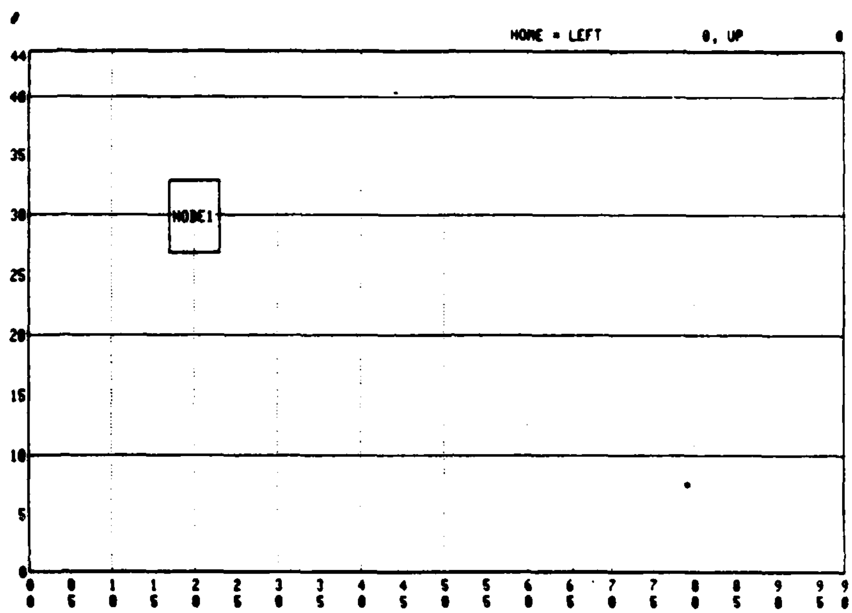


Figure 7. Architectural Grid With a Single Node

All nodes are placed in this way. To place nodes in the positions shown in Figure 8, type the following sequence of commands:

```
P TTY,NODE2,10,10 (cr)

P PRP,NODE3,40,30 (cr)

P TRI,NODE4,85,10 (cr)

P TAP,NODE5,45,15 (cr)

P CRD,NODE6,80,35 (cr)
```

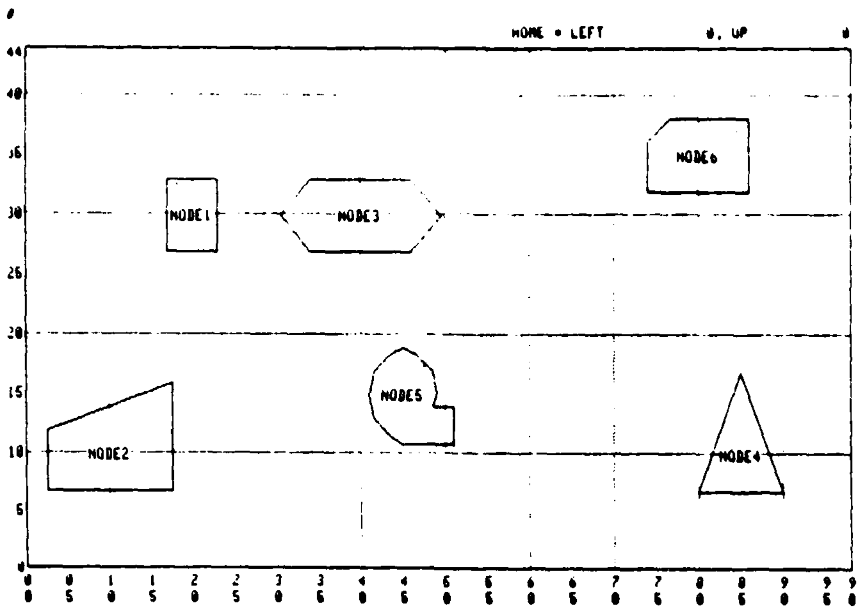


Figure 8. Six Nodes on an Architectural Grid

2.3 CONNECTING NODES

The second step in creating a system architecture is the placement of connections between the nodes. Such connections, or "links", are defined by specifying (1) the node from which the link is to run, (2) the node to which the link is to run, and (3) a user-given name of the link. To place a link called "LINK1" from NODE1 to NODE2, type,

```
CON NODE1,NODE2,LINK1 (cr)
```

This command places a cursor at NODE1; typing any character other than a period causes a straight line to be drawn between the centers of the two nodes, thereby drawing the link. The graphic result is shown in Figure 9.

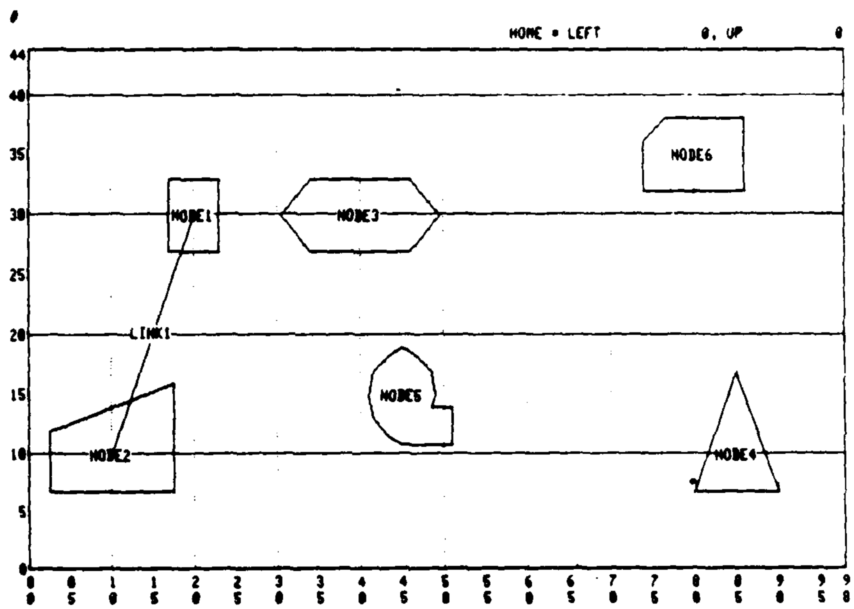


Figure 9. Architecture with One Link Defined

Links need not always appear as straight lines, as is shown in Figure 10.

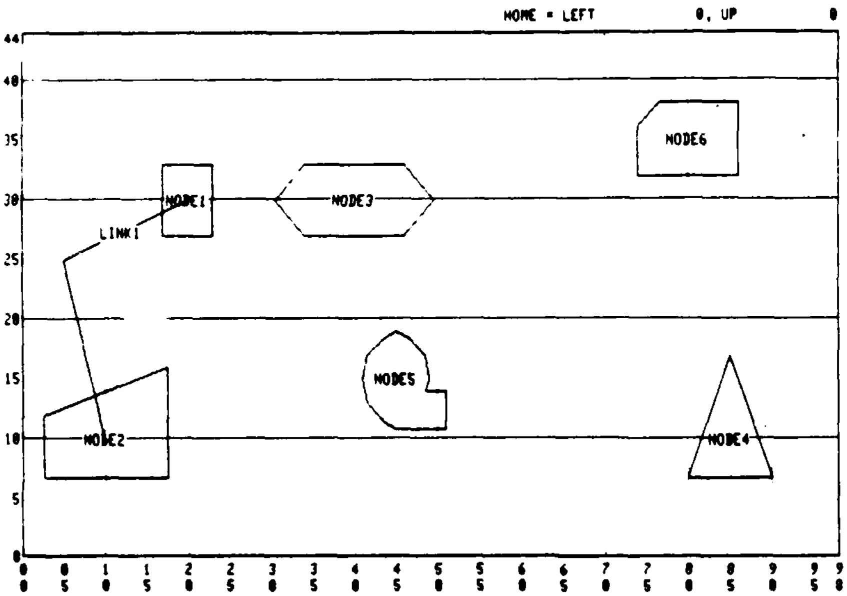


Figure 10. Architecture with a Bent Link

To create links that bend, do not hit a second carriage return after the graphics cursor is displayed (following the CON command). Instead, using the graphics controls on the HP2647A terminal (the ones shaped like arrow heads, not the ones to the far right) the cursor can be moved to the spot where the link is to bend. When the cursor is at the point of the bend, type in a period (.). If no further bending is desired, typing any other non-period character will complete the connection. The resulting connection will resemble the one depicted above in Figure 10.

Links may be given more than one bend by repeating the sequence of moving the cursor and typing a period (.), and then depressing any non-period character only when all the desired bends (up to six) have been created.

To create the links shown in Figure 11, type the following sequence of commands:

```
CON NODE1,NODE4,LINK4 (cr)

CON NODE3,NODE6,LINK3 (cr)
```

CON NODE3,NODE5,LINK5 (cr)

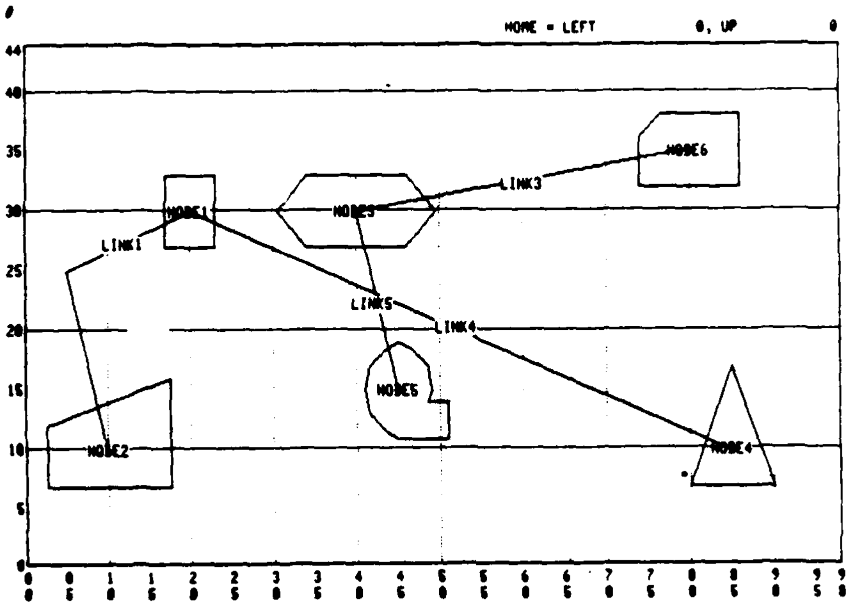


Figure 11. Architecture with Four Links

2.4 CHANGING THE SIZE, TYPE, AND NAME OF NODES AND LINKS

The size, type, or name of nodes and the names of links can be changed using the CHANGE command. By typing the following commands, nodes and links may be altered:

CHG NAME,NODE1,NODEX (cr)

CHG TYPE,NODE2,LOD (cr)

CHG SIZE,NODE3,7 (cr)

CHG NAME,LINK4,LINKZ (cr)

The user may note the changes on his screen. By typing the following commands, the architecture is returned to its original configuration:

CHG NAME,NODEX,NODE1 (cr)

CHG TYPE,NODE2,TTY (cr)

CHG SIZE,NODE3,5 (cr)

CHG NAME,LINKZ,LINK4 (cr)

As mentioned earlier, Resource entities are created by the AISIM system to model the architecture elements. When the name or type of a node is changed or the name of a link is changed, the appropriate changes are also made to the associated Resource entities. That is, when a node or link name is changed, the associated Resource name is changed. When the type of a node is changed, new attributes may replace the existing attributes of the Resource since different attributes may be defined for the new symbol type. Refer to Section 2.1 of this manual.

2.5 DELETING NODES AND LINKS

Existing nodes and links may be deleted from a system architecture with the DELETE command. For this example, to eliminate the connection between NODE1 and NODE2 type,

DELETE LINK1 (cr)

The result at the screen would be that the link named "LINK1" would disappear.

When a node is deleted, all of the links associated with it also disappear. As an example type,

DELETE NODE6 (cr)

The result of deleting LINK1 and NODE6 is shown in Figure 12. Note that LINK3 disappeared also.

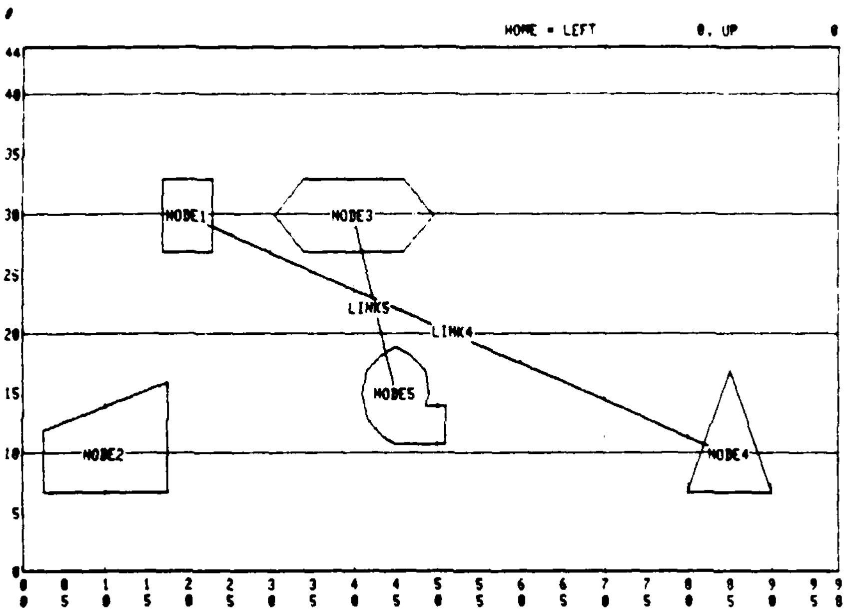


Figure 12. The Result of Deleting LINK1 and NODE6

2.6 MOVING PREVIOUSLY PLACED NODES

The location of a node on the architecture grid may be changed with the MOVE command. For example, to move NODE4, from its current position to the coordinates 55,5 one issues the command:

MOVE NODE4,55,5 (cr)

The graphic result is shown in Figure 13. The symbol is now centered at 55,5 with all of its previously defined connections intact.

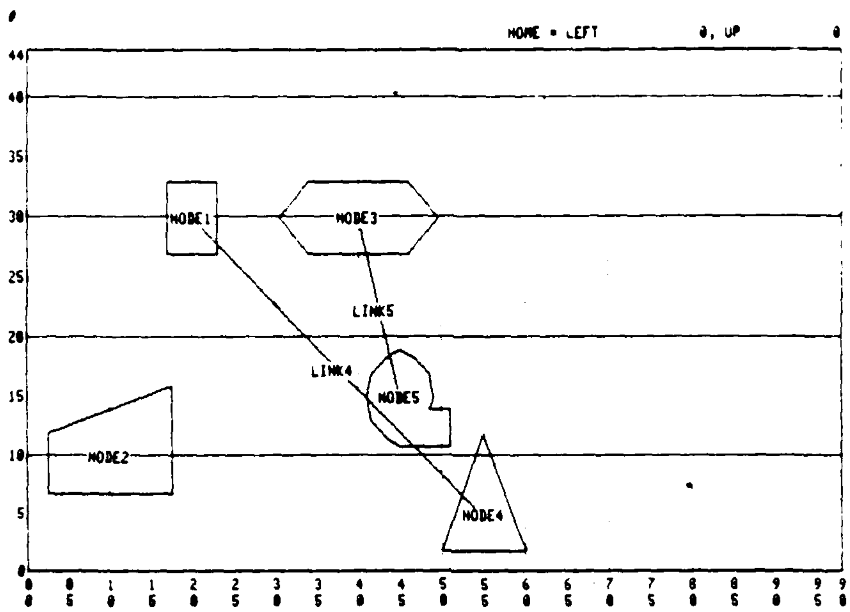


Figure 13. The Result of Moving NODE4

2.7 RECONNECTING EXISTING LINKS

The previous example of moving NODE4 created a problem that can be solved with the command RECONNECT. As Figure 13 shows, the link between NODE1 and NODE4 now runs through NODE5. The connection can be made to bend around NODE5 by first typing,

RECON LINK4 (cr)

This command will delete the existing graphics for the link between NODE1 and NODE4 and, as in the original CONNECT command, place the cursor (+) at NODE1. Using the sequence of cursor movements and periods (.) described in Section 2.3, up to six bends in the existing connection between NODE1 and NODE4 can be created. To complete the connection, type any non-period character. The graphic result will be something like that shown in Figure 14.

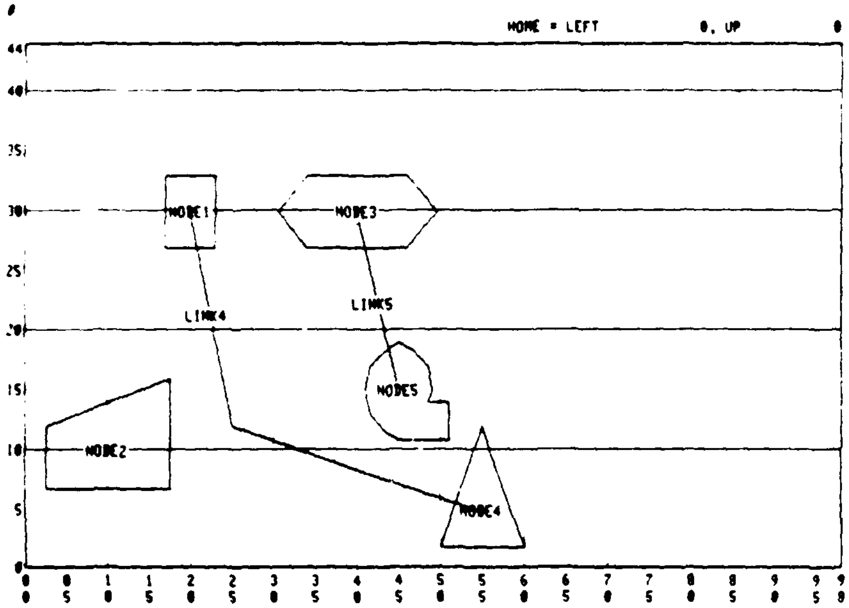


Figure 14. Architecture After Reconnecting

2.8 ALTERING ONE'S VIEW OF THE ARCHITECTURE GRID

The usable grid space in ADE is actually four times what can be displayed on the terminal screen at one time. If an architectural design is too large for the screen to accommodate, different parts of the total workspace can be viewed and manipulated through the WINDOW command. The WINDOW command allows the directional change of the user's view of the grid. The command specifies the direction of change--up, down, right or left--as well as the number of grid units the view is to be changed.

For example, to move the view of the screen in Figure 14 down 15 units, type,

```
WINDOW D,15 (cr)
```

Figure 15 shows the result of this command.

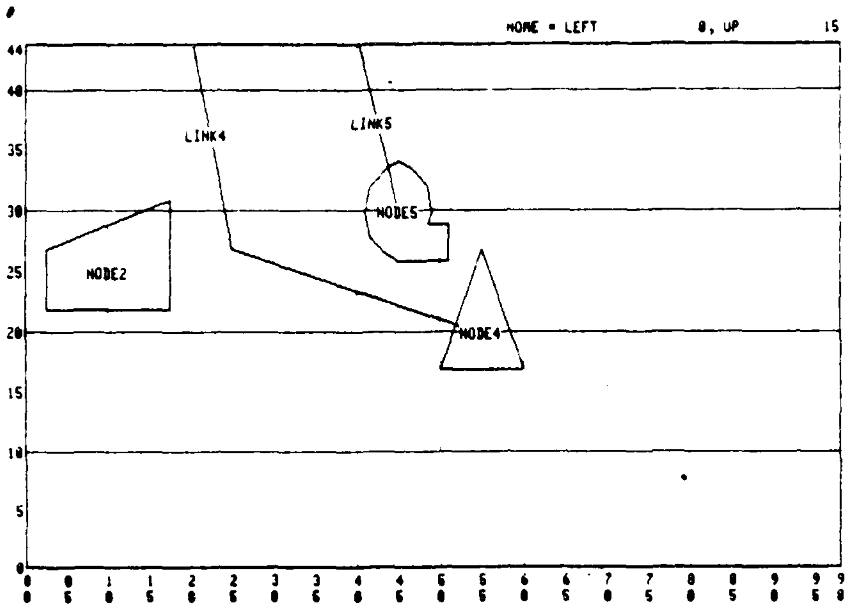


Figure 15. Result of WINDOW Command

The WINDOW command will accomplish both horizontal and vertical movements at the same time. To move our view of the screen further down 15 units and 20 units to the right, type,

WINDOW D,15,R,20 (cr)

Figure 16 shows the result of this command.

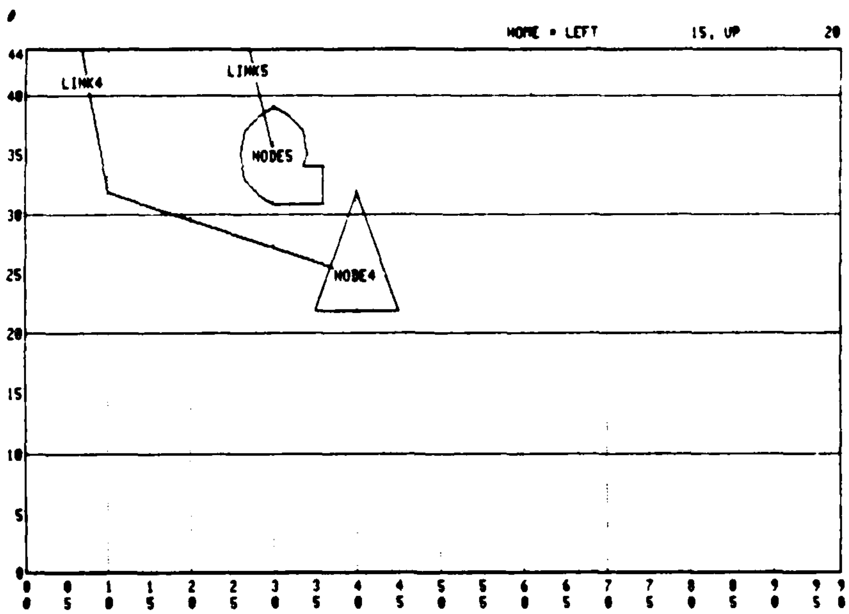


Figure 16. The Result of Further Use of WINDOW

Note that the WINDOW command parameters required to get back to the original (HOME) position are always displayed above the upper right corner of the architecture grid.

2.9 DEFINING LEGAL PATHS

The purpose of the Architecture facility is to specify routes of communication between hardware elements so that Process execution will be realistically related to the physical layout of a system. Such routes are represented by a Legal Path Table which specifies the links and the nodes through which communication from one node to another must take place. There are several methods of defining a Legal Path Table (LPT). Three methods are offered to the user at the end of an ADE session. These methods are predefined algorithms for the definition of an LPT which can be executed optionally at the user's discretion. See the AISIM User's Manual for details of how these algorithms function. For many architectures it is more economical to create the Legal Path Table while defining the configuration of hardware elements rather than using the algorithms mentioned above. If an LPT is generated according to the following discussion, the predefined algorithms should be bypassed since they would erase the LPT so defined.

Suppose we augment the architecture developed above with more links so that it resembles that shown in Figure 17.

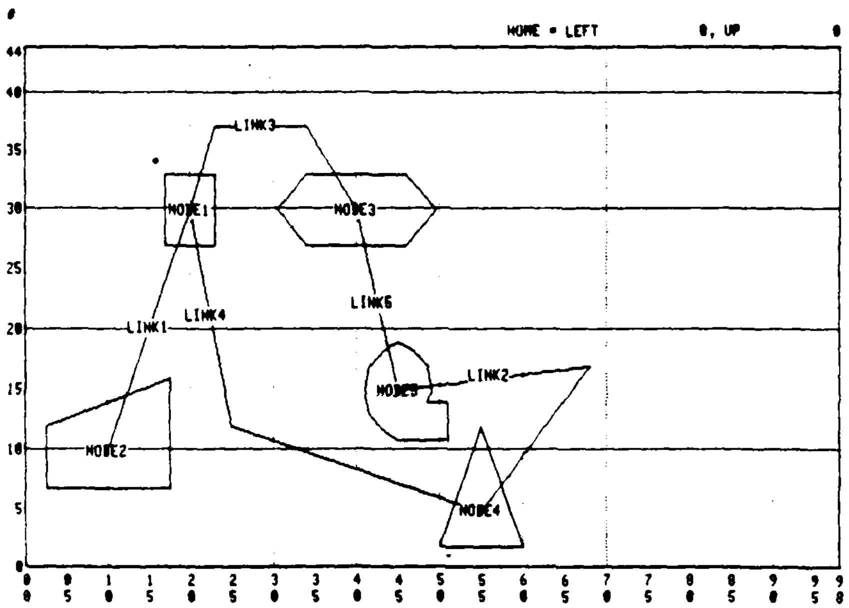


Figure 17. Augmented Architecture

The LPT is defined by means of the command DEFINE PATH. If, for example, NODE1 is to communicate with NODE4 along the communication lines represented by the links LINK3, LINK5, and LINK2, type,

DEFINE PATH,NODE1,NODE4,LINK3,LINK5,LINK2 (cr)

No confirmation will be displayed immediately at the screen, but the Legal Path Table will have been augmented to reflect the new paths. However, the command LIST PATH enables the user to inspect the Legal Path definitions currently in effect. To obtain a listing at the screen of the Legal Path from NODE1 to NODE4, type, LIST PATH,NODE1,NODE4 (cr)

The resulting list is shown in the upper right-hand corner of Figure 18.

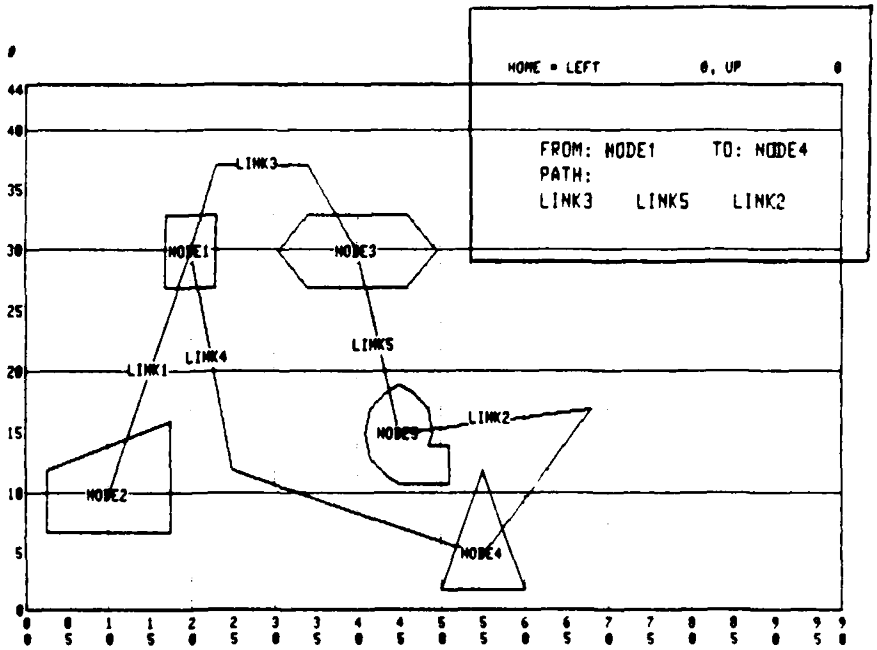


Figure 18. Typical List of Legal Paths Obtained in ADE

Note that paths <u>from</u> NODE3 and NODE5 <u>to</u> NODE4 have been automatically defined by the preceding DEFINE PATH command. The principle is that any time a Legal Path is defined through a number of nodes, the AISIM system creates Legal Paths from all nodes through which the path passes to the destination <u>to</u> node. Care should be taken in defining subsequent Legal Paths according to this method. Any conflicts of path routing in paths defined later would result in the elimination of previously defined paths. Following is an illustration of this operation of the system. Assume that the path has been created as above. If the user should now enter the command:

DEFINE PATH,NODE2,NODE4,LINK1,LINK4 (cr)

not only would the path <u>from</u> NODE2 <u>to</u> NODE4 be established, but the path from NODE1 to NODE4 would be altered to be the direct path via LINK4. The paths defined automatically from the previous command (i.e., the paths from nodes NODE3 and NODE5 to NODE4) would still exist since there was no conflict with these paths and the newly defined path.

3. DEFINING PROCESSES IN THE DUI

Whereas the Architecture Design Editor (ADE) is used to represent the the physical layout of a system, the Process Editor Interface (PEI) is used to represent the logic and data-handling behavior of processes in the system.

This section provides examples to familiarize the user with the commands and prompts used in AISIM. Earlier the user was urged to begin the design of an AISIM model with sufficient research and planning to fully understand the system to be modeled. However, as a teaching device, we shall develop fragments of a Process from requirements formulated as we go along.

The exercises here are intended both to deepen the user's grasp of the Process Primitives and to familiarize him with the prompts encountered while interacting with the computer.

Assuming the trainee has just completed the foregoing section, entered an END command to exit the ADE sublevel, and another END command to bypass the LPT generation, he will be at the DUI level of operation. A "*" should be displayed. To invoke the PEI sublevel of the DUI, one enters the EDIT PROCESS command designating the name of the Process to be edited. Once in the PEI, the user can terminate the PEI session by entering an END command.

Consider first the simplest Process that could be of any use to the modeler of a system: the Process starts, a certain amount of time is taken with an Action and then the Process ends. This Process will be represented in AISIM by the START symbol, the ACTION Primitive and the END symbol. To represent such a Process in AISIM, one begins by issuing the command,

EDIT PROCESS,EXAMPLE,NEW (cr)

which informs the computer that one wishes to design a Process named "EXAMPLE" which has not yet been defined*. The computer will respond with a form to be filled in at the terminal. This is done by typing into the fields provided in the form. The form is shown in Figure 19.

0.

*To alter a Process that has been previously defined, one would not enter the "NEW" part of the command.

START

PROCESS NAME   NODE

ATTRIBUTES ATTACHED (YES OR NO)

PROCESS DESCRIPTION

START BLOCK TYPE

ENTER "PARM" FOR PARAMETER PASSING
ENTER "ITEM" FOR ITEM PASSING
ENTER "STD " FOR STANDARD PROCESS

Figure 19. Initial Form for Process

The NODE field asks for the node in the architecture with which the Process is associated. Since this Process is not yet related to an architecture, the field is left blank. The next field allows the assignment of attributes to the Process. For the present, we shall decline to do so. The field labeled PROCESS DESCRIPTION allows the user to describe the Process. In this case, type "EXAMPLE PROCESS".

Depress the ENTER key (located above the keyboard proper). The cursor will sweep along the fields, the screen will go blank for a moment and then display the image depicted in Figure 20.

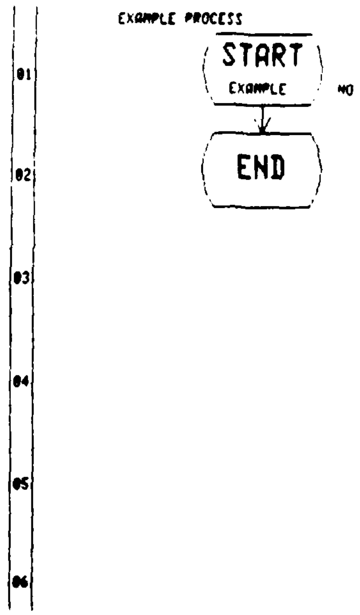


Figure 20. Graphic Display That Follows Entering a Process of the Standard Kind

Much of the information you entered on the form now appears in this graphic representation of EXAMPLE. (The "NO" to the right of the START symbol indicates the the Process has no attached attributes).

To place an ACTION Primitive between the Start and the End symbols, enter the command,

P ACTION (cr)

which tells the computer to place an ACTION Primitive between the last Primitive defined and the End symbol. The computer will now display a new form to be filled in. This is shown in Figure 21.

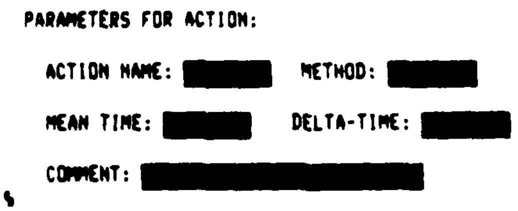


Figure 21. Form for the ACTION Primitive

The field ACTION NAME requests a name for the Primitive in the Process, which should be identical with the name of the associated Action entity (Action entities are described in section 4.1). We shall call it "Delay". The field COMMENT is a place to write a short reminder of what the ACTION Primitive is supposed to represent. The three remaining fields METHOD, MEAN TIME, and DELTA-TIME enable the user to vary the time taken up by the ACTION by invoking various statistical distribution methods (such as exponent, uniform, etc. See section 3.9.1 of the <u>AISIM User's Manual</u> for a description of the valid distributions.) Assume that the ACTION always requires the same amount of time, and hence the MEAN TIME requested will be equal to the duration of the ACTION. Indicate the time with a variable whose value for this example is specified elsewhere, calling it "T1".

The form should then be filled in thus: call the ACTION "Delay", set the method at "uniform", and set the MEAN TIME at "T1". Type the comment field "Action which causes delay". Leave the field labeled DELTA blank. After pressing the "ENTER" key the screen will display a new version of EXAMPLE, as depicted in Figure 22.

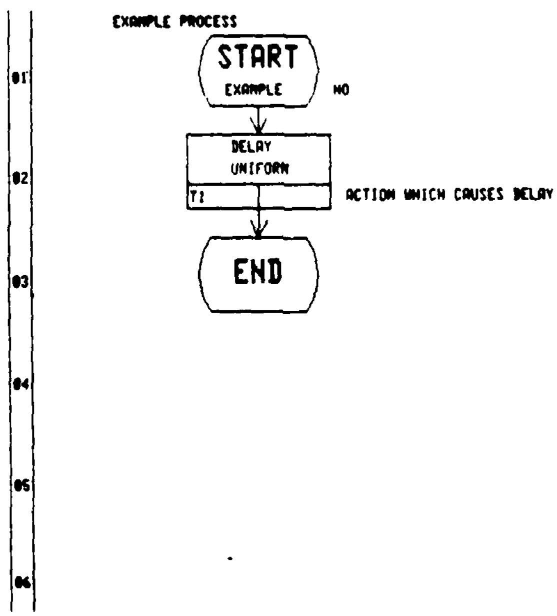


Figure 22. Process with a Single ACTION

3.1 HOLD

EXAMPLE may be augmented in a number of ways. For example, the ACTION Delay may be repeated in succession. There are two ways to repeat this ACTION in a revised version of EXAMPLE. First, one can place more copies of Delay in the Process, one after another, with the command

P ACTION (cr)

This command may be repeated as many times as one wishes the Action to be performed in the Process. The second method of creating several instances of an ACTION, which is less time-consuming, involves the HOLD storage area. HOLD constitutes a storage area for Primitives that are likely to be used more than once with little or no alteration. To place a previously defined Primitive into HOLD, type

HOLD 2 (cr)

The number 2 represents the position of the Primitive in the Process to be stored in HOLD. The position is indicated by the numbers in the column on the left. Hereafter, the ACTION Primitive "Delay" may be placed in a Process by typing,

P HOLD (cr)

Each time this latter command is issued, the user will be presented with the form associated with the Primitive in case any small alterations in its parameters are to be made. Whether or not any alterations are made, pressing the ENTER key will result in the placement of the Primitive stored in HOLD in the Process being edited.

Figure 23 shows the display that will appear after several identical ACTION Primitives have been redefined in succession or after the HOLD storage area containing the ACTION "Delay" has been placed. The procedure of repetition will occur as many times as the user requests it.

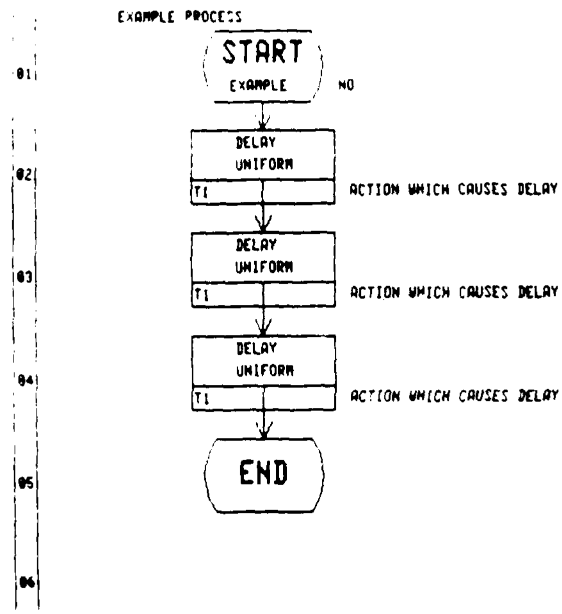


Figure 23. Process with Three Identical ACTION Primitives

3.2 ENTRY AND LOOP

If the ACTION "Delay" is to be performed a certain _n_ number of times, as in the most recent version of EXAMPLE, a much simpler procedure is available than that of placing _n_ instances of the ACTION between the START and END symbols. One can instead indicate more directly that a certain part of a Process is to repeat itself an _n_ number of times. This is accomplished by means of the Primitives LOOP and ENTRY. Figure 24 shows EXAMPLE altered with LOOP and ENTRY Primitives to cause a triple repetition of the ACTION "Delay".

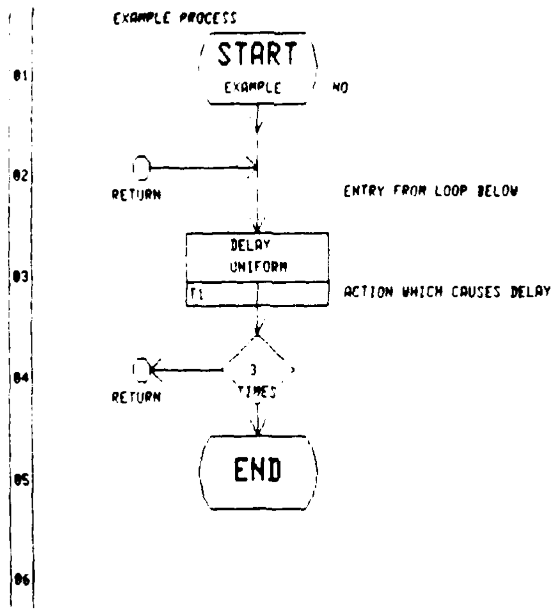


Figure 24. Process with Triple Repetition of the ACTION Delay

The diamond-shaped figure indicates that the line of processing is to be diverted to the point labled "RETURN" above it (it could have been given any label whatever up to 8 characters).

To effect the LOOP Primitive as shown in Figure 24, we must first get rid of the two extra ACTION Primitives. This is done by typing the following commands:

DELETE 3 (cr)

DELETE 3 (cr)

P LOOP (cr)

The screen will show the form shown in figure 25:

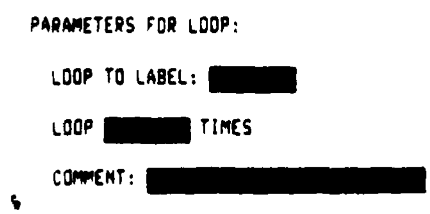


Figure 25. Form for the LOOP Primitive

The first field asks for the name of the entry point to which Process control is to be diverted. The second asks for the number of times control is to be diverted. The remaining two fields are self-explanatory.

The ENTRY Primitive must now be placed above the ACTION Primitive. Since the PLACE (or "P") command by default inserts the new Primitive just before the End symbol, a modified PLACE command is required to place a Primitive elsewhere in the sequence. To use this command, type,

P ENTRY,2 (cr)

The number "2" indicates where the Primitive is to be placed, referenced by the column of numbers on the left of the Process diagram.

The screen will then display the form shown in Figure 26.

```
PARAMETERS FOR ENTRY:

  ENTRY LABEL:

  COMMENT:
```

Figure 26. Form for the ENTRY Primitive

The label will, in this case, be determined by the LOOP label previously defined. The ENTRY LABEL field should be entered exactly as in the LOOP LABEL, i.e., as "RETURN". Type an appropriate comment in the field provided, such as, "ENTRY FROM LOOP BELOW". The result should be as in Figure 24.

3.3 PROB, TEST, COMPARE AND BRANCH

Four other Primitives, PROB, COMPARE, BRANCH, and TEST are similar to LOOP in that they represent a branching to an ENTRY Primitive. EXAMPLE may be altered in the following four ways.

3.3.1 PROB The PROB Primitive is used to indicate, for example, that the re-execution of the ACTION "Delay" has only a certain degree of probability.

Since AISIM has no command for directly replacing one Primitive with another, the existing Primitive LOOP must first be deleted.

Enter the command,

DEL 4 (cr)

where, as before, "4" indicates the location of the Primitive to be deleted. Figure 27 shows the display produced when the LOOP

Primitive is deleted.

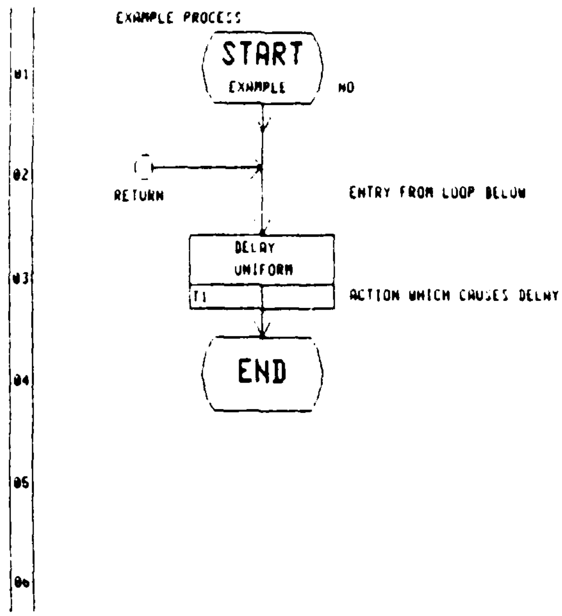


Figure 27. EXAMPLE after Deletion of LOOP Primitive

The PLACE command is used to insert a PROB Primitive between the ACTION "Delay" and the END symbol. Type,

P PROB (cr)

The screen will offer the form shown in Figure 28.

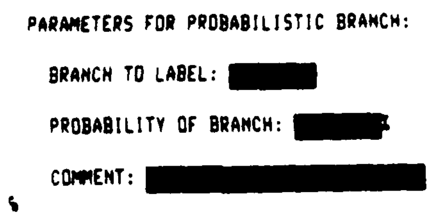


Figure 28. Form for PROB Primitive

Complete the first field with RETURN. The second field should be filled in with the probability of branching, given in percentages. Suppose here a 25% chance of branching. Type the appropriate comment, "25% chance of branching". The resulting display diagram is given in Figure 29.

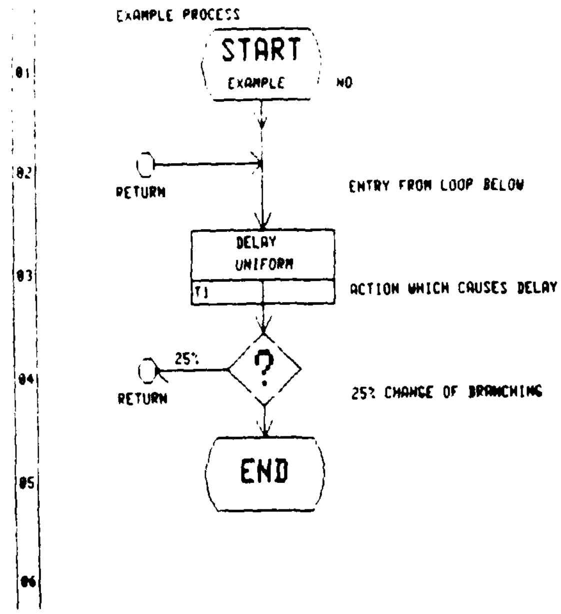


Figure 29. Process with Probabilistic Branch

3.3.2 TEST Another kind of branching Primitive is TEST. As mentioned earlier, Processes often make use of Resources for which there is competition. The TEST Primitive represents the procedure of ascertaining the availability of a given Resource and branching if that Resource is not available, or continuing if it is. This Primitive does not automatically allocate the Resource. To place the TEST Primitive (after having deleted the PROB Primitive from the latest version of EXAMPLE), type,

P TEST (cr)

The screen will display the form shown in Figure 30.

PARAMETERS FOR TEST:

RESOURCE NAME:

BRANCH TO LABEL IF NOT AVAILABLE

COMMENT:

Figure 30. Form for TEST Primitive

In the RESOURCE NAME field type the name of the Resource whose status is to be ascertained. The LABEL and COMMENT are self-explanatory. If the PROB Primitive in the previous version of EXAMPLE is replaced by TEST (as in this example), EXAMPLE will now appear as in Figure 31.

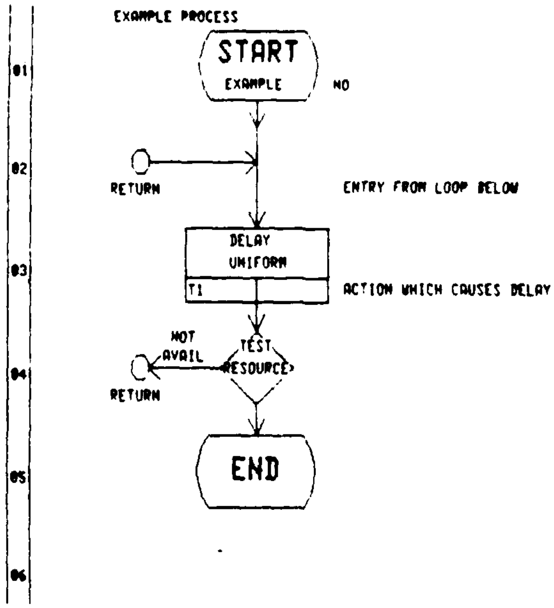


Figure 31. Process with TEST Branching

3.3.3 COMPARE In addition to probabilistic branching, AISIM also allows for conditional branching less specialized than the TEST Primitive. Most of these branchings will require the COMPARE Primitive. The COMPARE Primitive compares two numerical values with respect to some relation and branches to a named ENTRY Primitive if the relation holds.

To place the COMPARE Primitive, delete the previously defined TEST Primitive and type,

P COMPARE (cr)

The screen will display the form depicted in Figure 32.

PARAMETERS FOR COMPARE

IF OPERAND 1: QUALIFIER 1:

RELATION:

OPERAND 2: QUALIFIER 2:

BRANCH TO:

COMMENT:

Figure 32. Form for COMPARE Primitive

The fields OPERAND 1 and OPERAND 2 hold the variables whose values are to be compared. The values may be represented by arbitrarily chosen names of variables (such as VAR1 and VAR2). They are compared with respect to the following six arithmetical relations indicated by the two letter code:

EQ for "equal to"

NE for "not equal to"

GT for "greater than"

LT for "less than"

GE for "greater than or equal to"

LE for "less than or equal to"

The BRANCH TO and COMMENT labels are now self-explanatory. The two QUALIFIER fields serve several purposes, the most important of which is to allow the comparison of attributes of entities as opposed to simple variables or numerics. The user should for the present disregard the complication posed by these fields and leave them empty. Fill in the OPERAND fields with arbitrarily chosen names of variables, "VAR1" and "VAR2". If the TEST Primitive is replaced by the COMPARE Primitive with the foregoing information entered on its form, the new version of EXAMPLE will be as displayed in Figure 33.

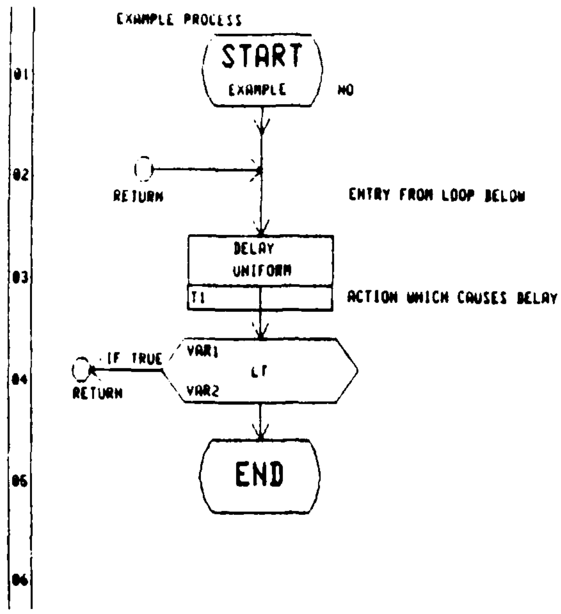


Figure 33. Process with COMPARE Primitive

And thus EXAMPLE is set to return control to the Entry Primitive if the value assigned to the variable VAR1 is less than the value assigned to the variable VAR2. These assignments are presumed to be made elsewhere.

3.4 VARIABLE MANIPULATION

In the previous example of the COMPARE Primitive, note that if the condition solicited is true, i.e., if VAR1 was less than VAR2, EXAMPLE would perform the ACTION "Delay" indefinitely. On each occasion in which the comparison is made, the relation will hold and hence the Process will always be instructed to branch to RETURN. If neither variable changes its value, the Process will continue until it is halted by other causes (such as having a Resource necessary to it allocated elsewhere).

Using two new Primitives, ASSIGN and EVAL, we can alter EXAMPLE so that the ACTION "Delay" does not go on forever but only for a certain maximum time ("Maxtime"). This is accomplished with the ASSIGN Primitive which introduces a new variable for the accumulation of time consumed by the ACTION's execution times and by the EVAL Primitive, which recalculates the value of this accumulated time each time the ACTION is performed.

First, to command the computer to place an ASSIGN Primitive between the START and the ENTRY, type,

P ASSIGN,2 (cr)

The screen will now show the form displayed in Figure 34:

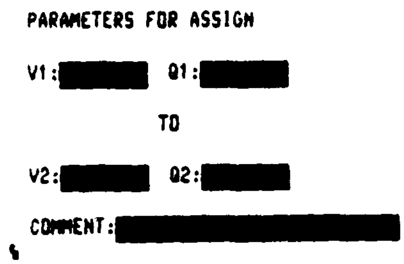


Figure 34. Form for ASSIGN Primitive

For this example disregard the fields labeled Q1 and Q2; they serve the same purpose as do the QUALIFIER fields in the COMPARE Primitive. The purpose of this exercise is to create a temporarily useful, local variable, which we shall call "acctime" whose value represents the amount of time that has been consumed in the repeated execution of "Delay". At the beginning of the Process the initial value of the variable will be zero. Hence, complete the V1 field with "ACCTIME" and the V2 field with "0". When this information is entered, the screen will display the graphic representation shown in Figure 35.

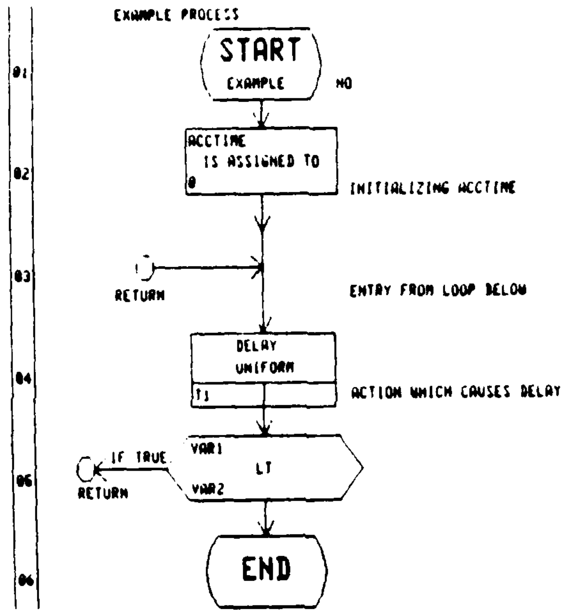


Figure 35. Process with Primitives ASSIGN, ACTION, and COMPARE

To provide an apparatus for updating the variable "acctime" on each occasion of the ACTION's execution, an EVAL Primitive must be placed between the ACTION and COMPARE Primitives. To do this, type,

P EVAL,5 (cr)

The screen will display the form shown in Figure 36.

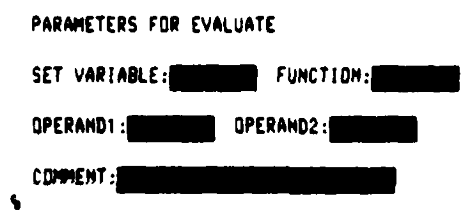


Figure 36. Form for EVAL Primitive

The SET VARIABLE field holds the name of the variable whose value is to be calculated. The FUNCTION field contains the name of the operation to be performed on the two operands contained in the fields OPERAND1 and OPERAND2. A large variety of functional operations are available for this field (see <u>AISIM User's Manual</u>, section 3.9.11 for a list). For this example, the <u>SET VARIABLE</u> field should be "acctime"; the Function, "add"; OPERAND1, "t1" and OPERAND2 "acctime". Type an appropriate comment, such as "evaluating acctime". The graphic representation of EXAMPLE will be:

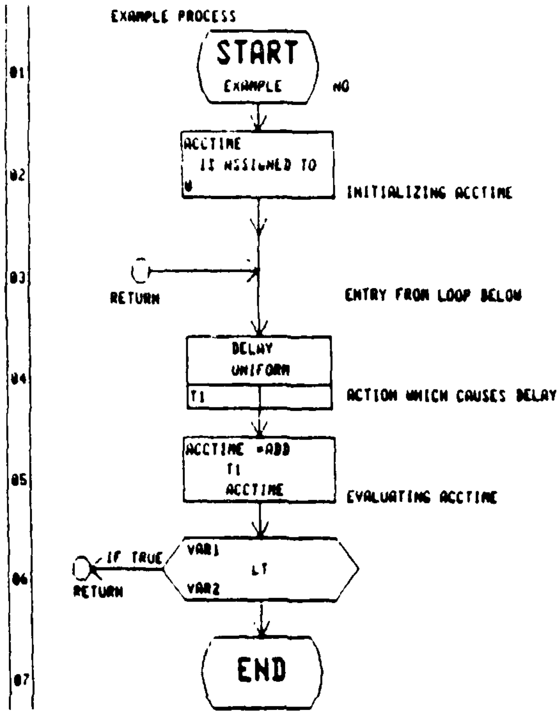


Figure 37. Process with ASSIGN, ACTION, COMPARE, and EVAL

This Primitive instructs the computer to add the current value of the variable "T1" to the value of "ACCTIME", producing an updated figure for the total time consumed by "Delay". Type an appropriate comment such as, "updating accumulated time".

The Process still requires alteration. The variables presently in the COMPARE Primitive must be changed from VAR1 and VAR2, respectively, to "acctime" and "maxtime". To do this, we must edit the COMPARE Primitive by typing,

C 6 (cr)

This command tells the computer that you wish to alter one or more of the previously defined parameters in the Primitive at location 6. The screen will display the form for the Primitive. It can be altered simply by writing over the existing information. When this is done and the form is "entered", EXAMPLE will satisfy the specifications for its alteration. Its graphic representation will be as in Figure 38.

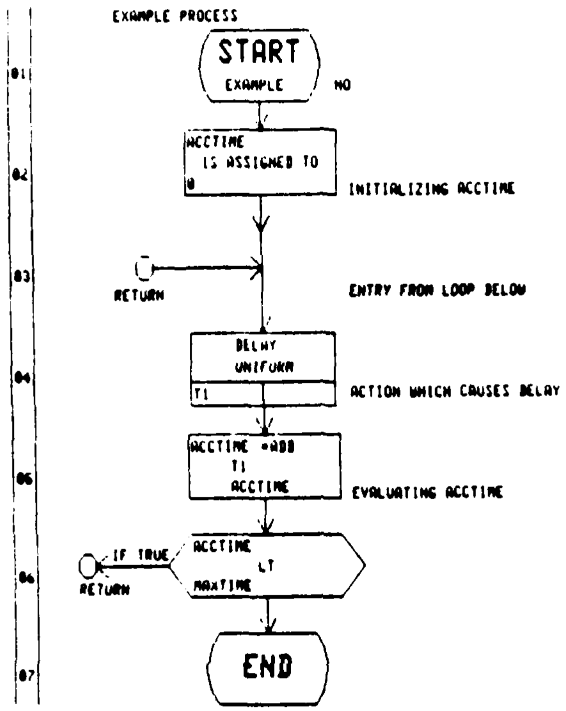


Figure 38. Process with Comparative Branching

3.5 ITEM MANIPULATION

Another group of Primitives is categorized under the headings Queue Handling and Item Handling. These include CREATE, DESTROY, FILE, FIND and REMOVE. The Primitives CREATE and FILE will be used in this example.*

Consider the first version of EXAMPLE which consisted of the single ACTION Primitive "Delay". Suppose now that we conceive of EXAMPLE as one which gives rise to new data elements--messages, information, potential communications. This function of the Process may be represented by means of the CREATE Primitive, which

0.

* Consult the AISIM User's Manual for information on the Primitives DESTROY, FIND and REMOVE

represents the introduction of Items--one of the AISIM modeling entities that represent transient data elements--into the modeled system. To place the Primitive CREATE below the ACTION Delay in EXAMPLE type,

P CREATE (cr)

The form for CREATE is shown in Figure 39.

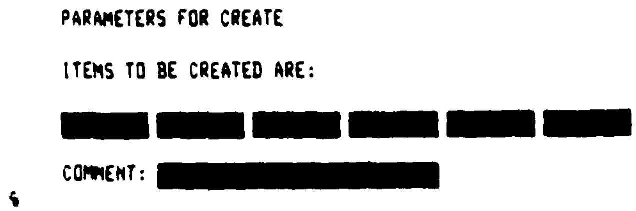


Figure 39. Form for CREATE Primitive

Complete this form with the names of the Items to be created. Enter the Item name "MSG" and an appropriate comment, "TRANSIENT DATA ELEMENT". EXAMPLE will now appear as indicated in Figure 40.

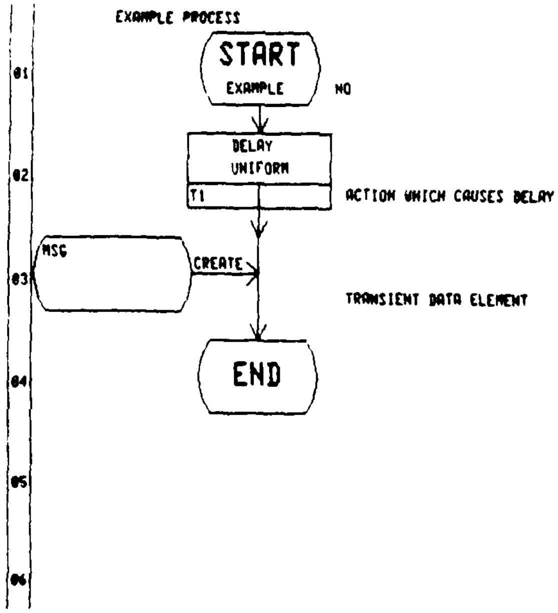


Figure 40. Item Creating Process

Items--transient data elements--can also be filed in holding areas called Queues with the FILE Primitive. To place a FILE Primitive below the CREATE Primitive in EXAMPLE, type, P FILE (cr)

The form for File is as shown in Figure 41.

PARAMETERS FOR FILE:

FILE ITEM NAME: OPTION: ON QUEUE:

COMMENT:

Figure 41. Form for FILE Primitive

Complete the field FILE ITEM NAME with "msg". The OPTION field tells where in the Queue the Item is to be placed. This location is specified relative either to absolute locations on the Queue ("FIRST" and "LAST") or relative to some other Item already on the Queue ("BEFORE" and "NEXT")*. The OPTION field will have as a default parameter LAST. In the ON QUEUE field enter MSG-QUE. The graphic representation of this Process is indicated in Figure 42.

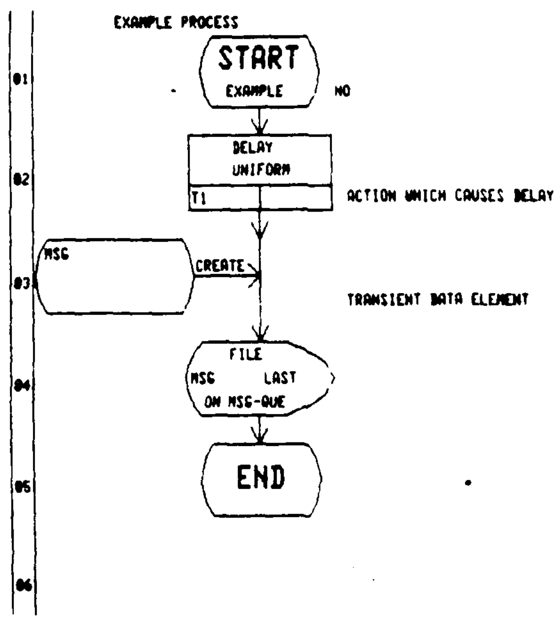


Figure 42. Process which Creates and Files Message Items

0.

*The method by which the system identifies the Item relative to which other Item is to be placed on a Queue (with the OPTION BEFORE or NEXT) need not concern us here. For more on this, see AISIM User's Manual, Section 3.9.12.

3.6 RELATIONS AMONG PROCESSES

This section deals with the relationships that the execution of Processes bear to one another in an AISIM Model, and how one Process, and its execution, affects the execution of another. Processes affect one another's execution in three ways:

1. by sending Items that trigger the execution of another Process.

2. by triggering the execution of another Process through a CALL Primitive where the CALL may or may not pass parameters.

3. by competing for and obtaining use of a Resource needed by another Process.

To understand how parameter and Item "passing" affect the execution of a Process, consider the form completed in the first version of EXAMPLE. In the form presented as a result of the command to edit a Process (i.e., E PROCESS,EXAMPLE,NEW), in the START field TYPE, the choices included "STD", "PARM" and "ITEM", standing for, respectively, "standard" "parameter passing" and "Item passing". These options are distinguished from one another in the following way. A Process can, before it is fully designed, be thought of as a "black box" whose internal workings are unknown. If the Process is conceived to be one that performs its function without having to be given anything in the way of information or data elements it will be a Standard Process. If the Process requires certain data elements--discrete, countable entities--in order to execute, then it is an "Item passing" Process. Finally, if the Process uses values of variables local to another Process, it is a Parameter passing Process.

For the first example, consider Item Passing Processes. In this exercise, delete the File and CREATE Primitives from EXAMPLE. To change EXAMPLE from a Standard Process, as it now is, to an Item Passing Process, type,

    C 1 (cr)

The screen will display the form originally filled out for EXAMPLE. Type "ITEM" over the existing "STD" in START field TYPE. Entering this, the screen will now show this secondary form on which Items needed by this Process are to be written, as shown in Figure 43.

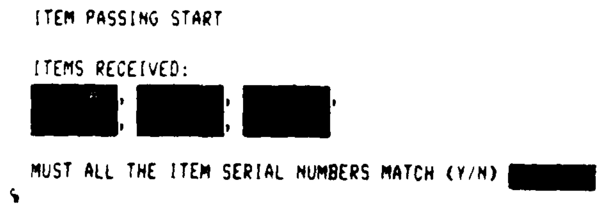


Figure 43. Secondary Form for Process

Type the single Item name "MSG" in the upper left field. Entering this changes the Process representation so that it appears as in Figure 44.

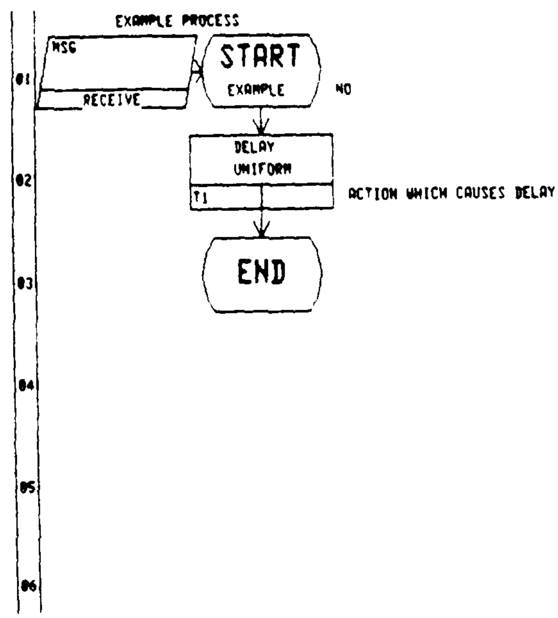


Figure 44. Item Passing Process

The figure to the left of the Start figure indicates that the Process starts when, and only when, the Item MSG is delivered to it from some other Process.

None of the Primitives in the categories Item Handling and Queue Manipulation represent the delivery of Items to a Process. This delivery function is accomplished by the Send. To exemplify Send, a new Process, called "EXAMP-2", must be created. EXAMP-2 triggers the execution of EXAMPLE by delivering Items to it. For this example consider a Process identical except in name to the original EXAMPLE with the single ACTION Delay as depicted in Figure 45.

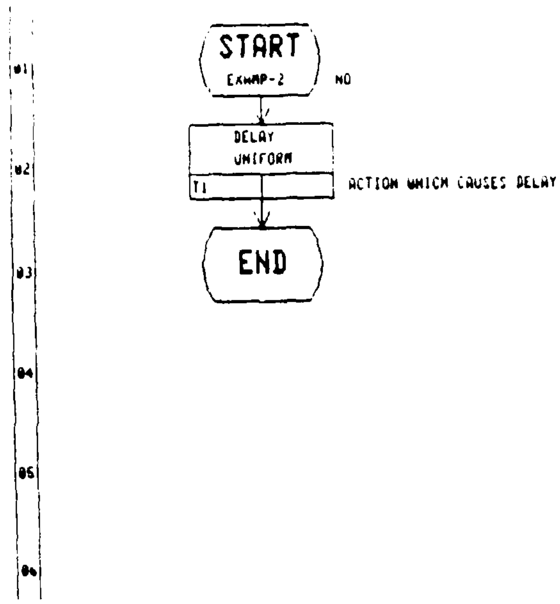


Figure 45. Process with ACTION Primitive

Type the command,

P SEND(cr)

The screen will display the form shown in Figure 46.

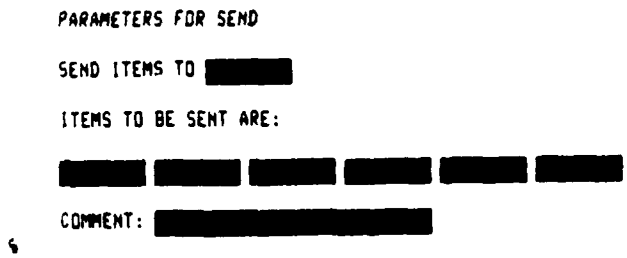


Figure 46. Form for SEND Primitive

Complete the SEND ITEMS TO field with EXAMPLE. In the first field of ITEMS TO BE SENT fields, type MSG. Enter the comment "*SENDING MESSGE ITEM*". Figure 47 shows the graphic representation of the Process that will appear on the screen.

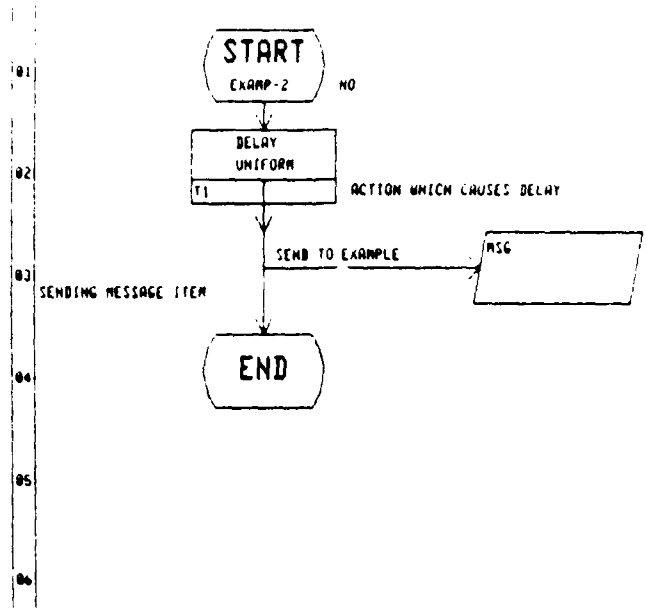


Figure 47. Graphic Representation of EXAMP-2

EXAMP-2 now triggers EXAMPLE by delivering to it Items required for its execution. The Item is automatically created by the Send Primitive. An Item-passing Process may only be initiated through the SEND Primitive in some other Process, although the Items needed and hence the Items sent may be distributed among several Processes or several stages of a single Process.

3.7 RESOURCE ALLOCATION

As mentioned earlier, Processes in an AISIM model frequently make use of Resources. A Resource has a finite capacity which will limit the number of Processes it can accommodate at the same time. The five Primitives which relate to the allocation of such Resources are ALLOC, DEALLOC, RESET, LOCK and UNLOCK.

ALLOC and DEALLOC signal the allocation and deallocation of a Resource by the Process in which they appear. To place the ALLOC Primitive above the ACTION Primitive in EXAMPLE, type,

P ALLOC,2 (cr)

To place a DEALLOC Primitive just above the END symbol in EXAMP-2, type,

P DEALLOC (cr)

The forms for these two primitives are shown below in Figure 48.

PARAMETERS FOR ALLOCATE:

ALLOCATE RESOURCE NAME:

COMMENT:

PARAMETERS FOR DEALLOCATE:

DEALLOCATE RESOURCE NAME:

COMMENT:

Figure 48. Forms for Primitives ALLOC and DEALLOC

In each case, enter the name of the Resource to be allocated or deallocated, such as "CPU", in the field provided. Enter the appropriate comment, "OBTAINING CPU" or "RELEASING CPU" in the COMMENT field.

Placing these primitives in EXAMP-2 (one above the ACTION and one below), produces a graphic representation like that shown in Figure 49.

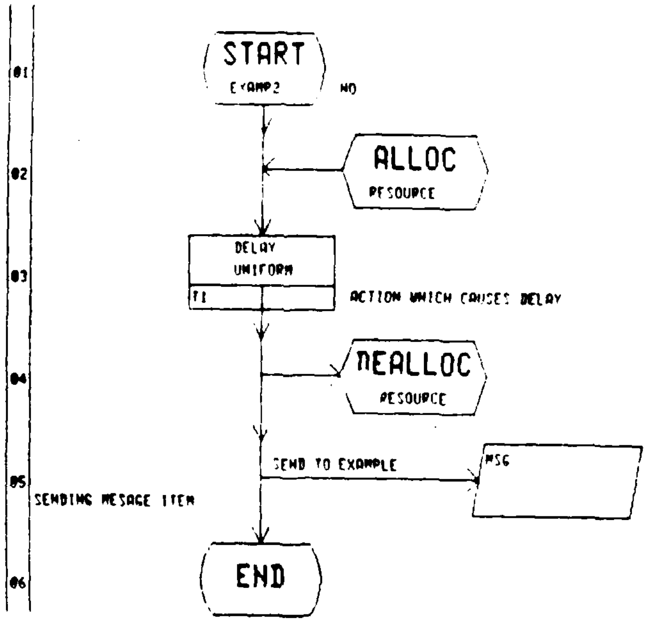


Figure 49. Process which Allocates and Deallocates a Resource

Allocating a Resource does not normally insure the uninterrupted availability of that Resource to a Process. Any Resources may be usurped by a Process with a higher priority. If the Process being modeled is one which, once begun, cannot be interrupted, the Primitives LOCK and UNLOCK must be used.

To obtain the forms for these Primitives one types,

P LOCK,n (cr)

or

P UNLOCK,n (cr)

where n is the position in the Process where the Primitive is to be placed. The forms for these Primitives are shown in Figure 50.

PARAMETERS FOR LOCK:

COMMENT:

PARAMETERS FOR UNLOCK:

COMMENT:

Figure 50. Forms for Primitives LOCK and UNLOCK

These Primitives, if placed below the ALLOC Primitive and above the DEALLOC Primitive in EXAMP-2 would give a graphic representation like that shown in Figure 51.

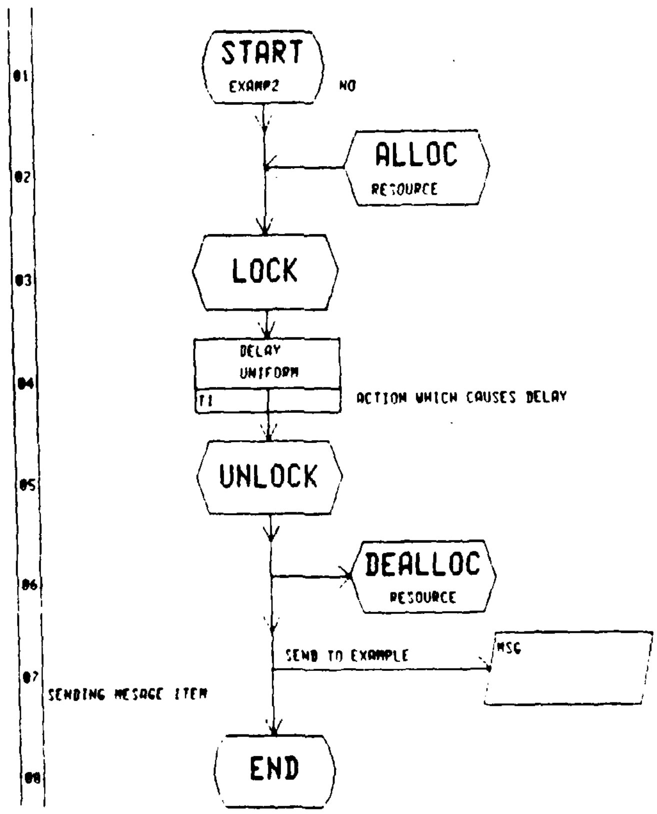


Figure 51. Process with Protected Resources

One final way to affect a Resource is through the RESET Primitive. It is used to reset the capacity of a Resource, where "capacity" is a measure of the number of Processes it will accommodate (support) at one time. For details on its use, see the AISIM User's Manual, Section 3.9.18.

3.8 CALL

The function of the CALL Primitive is similar to that of the SEND Primitive, but whereas the SEND Primitive triggers Item-passing Processes, the CALL Primitive triggers both Standard Processes and parameter-passing Processes. Thus, to understand how CALL works requires a brief discussion of parameter-passing Processes.

A parameter-passing Process is one that is "given" values for input variables and "returns" values for output variables. To create a paramater-passing Process, one would type "PARM" in the field START TYPE in the original form for Process. Entering this information on the Process form yields the secondary form shown in Figure 52.

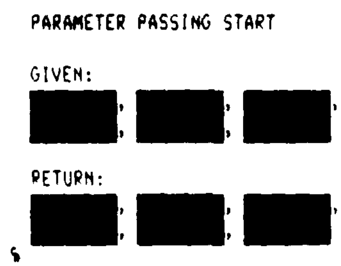


Figure 52. Secondary Form for Parameter-passing Process

On the form in Figure 52, one types the variables whose values are passed to the Process and the variables whose values are passed back.

The CALL Primitive values, i.e., parameters, are passed (GIVEN) to a called Process and RETURNed to the calling Process. Parameter passing can occur only through the use of a CALL Primitive. A CALL Primitive is placed in a Process by typing,

P CALL (cr)

The form for CALL is shown in Figure 53.

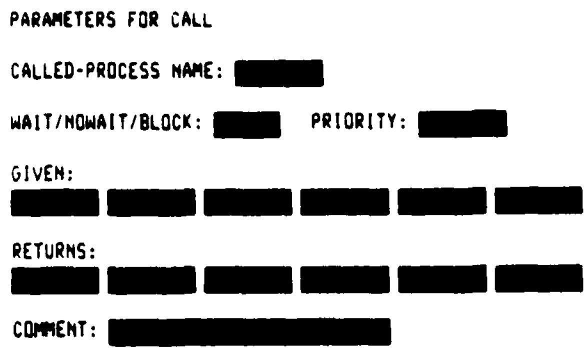


Figure 53. Form for CALL Primitive

The field CALLED-PROCESS NAME asks for the name of the Process to be triggered. The field PRIORITY determines the priority associated with the called Process which will be used in cases of Resource contention. The GIVEN and RETURNS fields hold the local variables whose values are passed to and from the called Process. A CALL Primitive may trigger a Standard Process and hence these fields may be empty. The COMMENT field is self-explanatory. The field labled WAIT/NOWAIT/BLOCK determines whether the calling Process will wait for the called Process before continuing execution or will continue to execute independently of it. The reader is referred to the <u>AISIM User's Manual</u> for details on their use.

4. REMAINING MODEL ELEMENTS

Although the Processes and the Architecture are core modeling elements, their specification does not complete the task of model construction. They must be supplemented by definitions of other elements. These elements are grouped into two categories. The first category consists of those entities explicitly referred to in Processes, namely, Actions, Constants, (global) Variables, Tables, Queues and Resources. The second category consists of the two entities that are used to represent the impact of the environment on the modeled system. All these remaining entities are defined at the DUI level of AISIM operation.

The following two sections briefly describe the parameters, significance and principle commands associated with these remaining entities.

4.1 ACTIONS

Any ACTION Primitive placed in a Process must have a corresponding Action entity defined outside the Process. Such a definition is created by typing,

```
E ACTION,ACTION NAME,NEW (cr)
```

The form for the Action entity is shown in Figure 54.

```
ACTION:
  CLASS:
  DESCRIPTION:
```

Figure 54. Form For Action Entity

The ACTION field should hold the name of an Action referenced in some ACTION Primitive. The CLASS is an optional parameter for the user to provide a categorization--man, machine, etc.-- of the sort of activity the Action represents. It functions as a second comment field. This field does not affect AISIM's operation and may be left blank. The field DESCRIPTION is for any convenient reminder of what the Action represents. It can be the same as the description of the corresponding Process Primitive.

4.2 RESOURCES

As mentioned earlier, any Resource mentioned in a Process--through the ALLOC, DEALLOC, FILE, FIND or REMOVE Primitives--must be defined separately in the PEI. To create a new Resource, type,

E RESOURCE,NAME,NEW (cr)

The screen will display the form shown in Figure 55.

RESOURCE NAME:

TOTAL NUMBER OF UNITS:

INITIAL NUMBER OF UNITS:

ATTRIBUTES PRESENT (YES OR NO)

COST:

DESCRIPTION:

Figure 55. Form For Resource Entity

Complete the first field, RESOURCE NAME, with the name by which it is referred in any Process. The fields TOTAL NUMBER OF UNITS and INITIAL NUMBER OF UNITS indicate, respectively, the maximum number of Processes the Resource can accommodate at any one time and the number of Processes it can accommodate at the beginning of a simulation run (i.e., before being increased or decreased by the RESET Primitive). Enter the appropriate numbers. The field COST functions as any other Resource attribute. DESCRIPTION has its usual function. Type an appropriate description in the field provided.

The field ATTRIBUTES PRESENT indicates whether the Resource has associated with it attributes other than "COST" which can be referred to and manipulated by the Primitives ASSIGN and EVAL. If the user enters "YES" in this field, he will be offered the following secondary form shown in Figure 56.

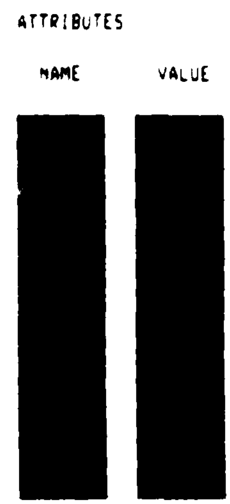


Figure 56. Form For Attributes of an Entity

Up to fifteen attribute names may be entered with their initial values.

Though all Resources referred to require separate definitions, some Resources are defined automatically. For each node or link created in a model's network architecture, a Resource definition of the same name with default parameters is automatically written into the database. In other words, all nodes and links are identified with Resources. Thus, after an architecture has been created the command,

```
               nodename
E RESOURCE,
               linkname
```

can be issued without having to indicate that the Resource entity is new (with "NEW"). Typically, however, not all of a system's Resources will be represented in the architecture and not all of the Resources automatically created in ADE will have any positive role in the operation of the model. That is, such automatically defined Resources need not be invoked in the Process primitives. Importantly, if an operative Resource is to be identified with an architectural element, it should be defined first in ADE and thereafter edited to provide it with suitable parameters (on the assumption that the default parameters are incorrect). ADE will not allow the definition of a node or link whose name is identical with that of a Resource already in existence.

4.3 QUEUES

Not all the Queues functioning in a system model need be defined by the user, since many are implicit in the operation of the system. The general rule is that any Queue manipulated by the FILE, FIND or REMOVE Primitives must be given a separate definition in the DUI, with the exception of these two:

--any Resource idle queue

--any cross-reference set

These are explained in the AISIM User's Manual, Appendix D.

To define a new Queue, type,

E QUEUE,NAME,NEW (cr)

The form for this entity is shown in Figure 57.

QUEUE: SIZE:

DESCR:

Figure 57. Form for Queue Entity

The three fields should be filled in with, respectively, (a) the name of the Queue as found in the FILE, FIND,or REMOVE Primitive which invokes the Queue, (b) the maximum number of Items or Resources that can be placed in it (the default value for which is "infinite") and (c) any useful reminder of the Queue's role in the modeled system.

4.4 CONSTANTS AND VARIABLES

Constants differ from global Variables only in that they do not change their values during the simulation exercise of a model. This can be puzzling at first since Constants, like Variables, are represented by non-numeric symbols. Also from the user's point of view, however, they behave quite similarly since they can both be altered immediately before the simulation exercise of a model. However, once a value has been assigned to a Constant and a simulation is begun, its value is unchanging. Accidental attempts to alter the value of a Constant through the EVAL or ASSIGN primitives will yield an execution error message.

The forms for Constants and Variables are quite similar and are called up by issuing the command "E" or "EDIT" followed by a space and "CONSTANT" or "VARIABLE", then a comma and the Constant's or Variable's name.

The forms for Constant and Variable are shown in Figure 58.

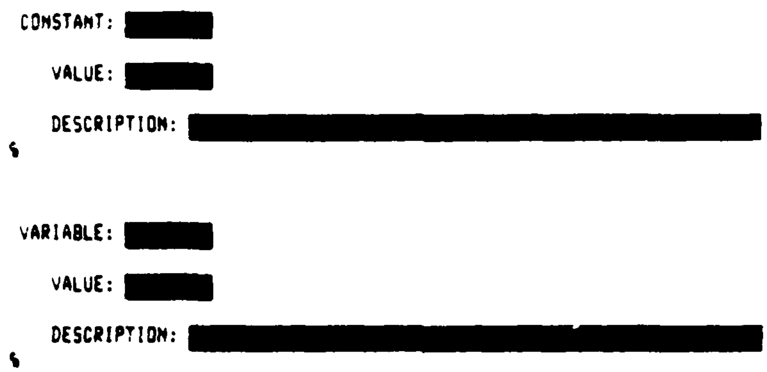


Figure 58. Forms for Constant And Variable

The fields CONSTANT and VARIABLE call for the entities' names. The VALUE fields call for numerical values (initial for Variables, permanent for Constants) and the DESCRIPTION fields call for any description.

4.5 LOADS AND SCENARIOS

The effect of the environment on a model is represented collectively by Loads and Scenarios. The relationship between Loads and Scenarios is this: Loads specify a number of Process triggerings to take place sometime during the simulation exercise of a model. Loads do not specify when the Process triggerings are to take place. Scenarios specify a collection of Loads and/or individual Processes together with a schedule indicating when the specified Loads or Processes are to be initiated.

To define a Load, type

```
E LOAD,NAME,NEW (cr)
```

The form for the Load Entity is shown in Figure 59.

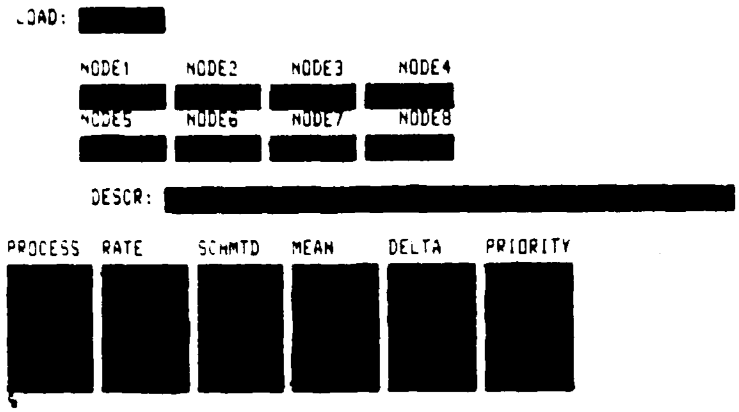


Figure 59. Form for Load Entity

The LOAD field holds the name of the Load. The fields labeled NODE1 through NODE8 indicate the architectural nodes in which the Processes named in the Load take place. The field DESCR is for any helpful description.

The field labeled PROCESS holds up to five names of Processes. The fields SCHMDT, MEAN and DELTA together define the statistical method of distribution to be used in scheduling the Process triggerings. SCHMDT holds the name of the distribution method; MEAN holds the average time between Process initiations (in terms of the simulation clock) and DELTA is a second numerical parameter used only for certain methods. The field MAX # indicates the maximum number of Process instances to be initiated by this Load.

The Scenario entity defines the impact of the environment on the system for the entire simulation exercise of a model. In it one specifies a number of "periods" into which a simulation exercise is to be divided, together with a uniform length each period is to have. One then specifies a collection of Loads or Process to be initiated at a specified time during the simulation. A priority is also given to resolve conflicts in the requests for Resources.

To define a Scenario, type,

    E SCENARIO,NAME,NEW (cr)

The form for the Scenario entity is shown in Figure 60.

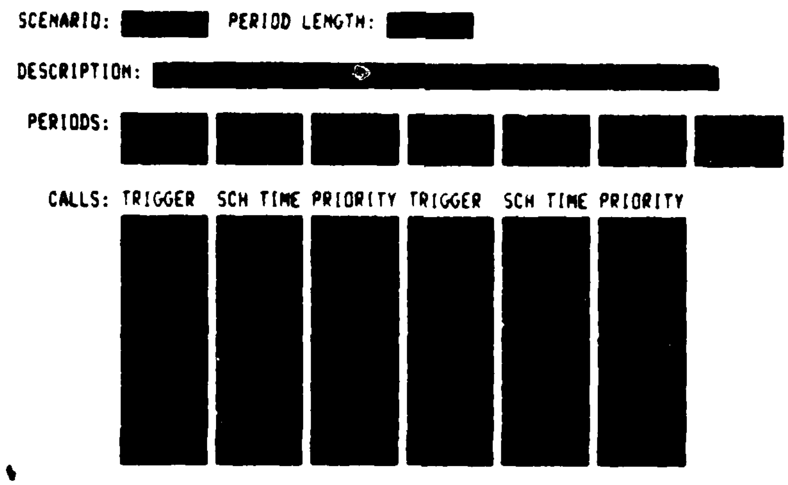


Figure 60. Form for Scenario Entity

The field SCENARIO holds the name of the entity. PERIOD LENGTH is the length of each period. The 14 fields labeled PERIODS are used to indicate the number of periods the Scenario is to have. The number of periods in the Scenario is determined by the number of these fields in which an entry is made. Any user-defined names (i.e., any characters) may be typed in these fields.

The fields labeled TRIGGER take the name of the Load or Process to be initiated. The fields SCH TIME indicate the time at which the Load or Process named immediately to the left is to be initiated. The field PRIORITY is used to assign a priority to the named Load or Process.

5. A WORKING EXAMPLE

This section documents the construction of an AISIM model that can be run through simulation tests and analyzed in the subsequent chapter. The model will be a representation of the transmitter/receiver relationship, an element of any communication system.

The transmitter/receiver relation modeled here is of the "polling" or "mailbox" type, as opposed to the "interrupt" type. In it, one transmitting Process generates messages and delivers them to a buffer. There the messages await treatment from a receiving Process. The transmitting and receiving Processes are not in direct communication with one another. Rather, the transmitter broadcasts messages according to need, and the receiving Process reads them from the buffer at intervals in accordance with expected need. In the system envisioned, transmission is randomized in two respects, (1) in the lengths of transmitted messages and (2) in the intervals between transmission. Reception is undertaken at regular intervals and the time consumed in processing a message is a linear function of its length.

The origination of a message in the transmitting Process will be represented by the creation of an Item (through the CREATE Primitive). The Item will have a variable attribute which will represent its length. Since the length will be randomized over a range of approximately 700 bytes, some mechanism must be incorporated for altering the variable attribute of each data Item (i.e., message). This is accomplished by (1) generating a random number in the range [0,1] subsequent to the creation of each Item, (2) multiplying the random number by twice the average message length and (3) assigning the number so obtained to the message length. This figure will then be used to calculate the time taken to send the message to the buffer (where it will be available to the receiving Process). Through an ACTION Primitive, the clock is then updated in the amount calculated.

In this system the buffer will not be manipulated by both the receiving and the transmitting Processes at the same time, so the buffer is considered a Resource and its allocation and deallocation by the ALLOC and DEALLOC Primitives will prevent it from being accessed simultaneously by both Processes.

5.1 DEFINING PROCESSES

This description of the transmitting Process gives the steps of its execution. The transmitting Process:

(1) Starts

(2) Allocates a Resource BUF1 representing the buffer

(3) Creates a message, represented as an Item called "msg"

(4) Generates a random number between 0 and 1

(5) Multiplies the random number generated by twice the average message length

(6) Assigns the number obtained in the previous step to the Item attribute representing the message length

(6) Updates the clock by an amount proportional to the message length (i.e., in an amount equal to the message length times the transmission rate in seconds per byte)

(7) Delivers the message Item to the Queue called Buffer through the FILE Primitive.

(8) Releases the Resource BUF1 representing the buffer

Figure 61 shows Process flow-chart derived from this description.

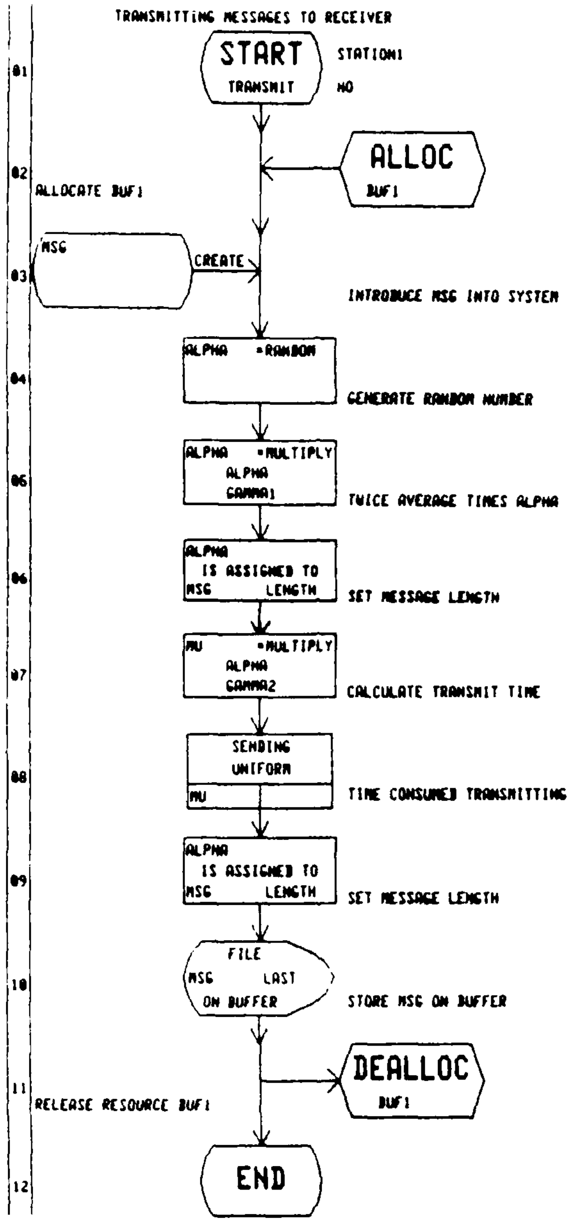


Figure 61. Transmitting Process

The receiving Process will first determine whether or not the buffer is being manipulated by the other Process by testing for utilization of the Resource call BUF1. If the Resource is in use, the Process will abort by branching to the END symbol. If BUF1 is free, the Process will read the next message from the buffer, and calculate a receiving time in roughly the same way that the transmitting time for that same Item was determined in the transmitting Process. The clock is then updated by the amount of time calculated.

This description can be expanded into more specific design requirements. The receiving Process will:

(1) Start.

(2) Test for the availability of the buffer by determining whether or not the Resource is in use through the TEST Primitive. If so, the Processes execution will branch to the END symbol.

(3) The next message Item on the Queue called buffer will be read off through the REMOVE Primitive.

(4) If there is nothing on the buffer, Process execution, as in step (2), will branch to the End. This step will be represented by a COMPARE Primitive.

(5) The message length will be assigned to a local variable through the ASSIGN Primitive.

(6) A receiving time will be calculated to be proportional to the message length (i.e., equal to the message length times some reception speed in seconds per byte).

(7) The clock will be updated through the ACTION Primitive in the amount required to receive the message.

(8) The message Item, having been read, will be eliminated from the system through the DESTROY Primitive.

(9) An ENTRY Primitive will be inserted just before the END symbol of the Process to indicate where execution is to resume from the branchings in steps (2) and (4).

Figure 62 shows the flow-chart representation of the Process derived from these requirements.

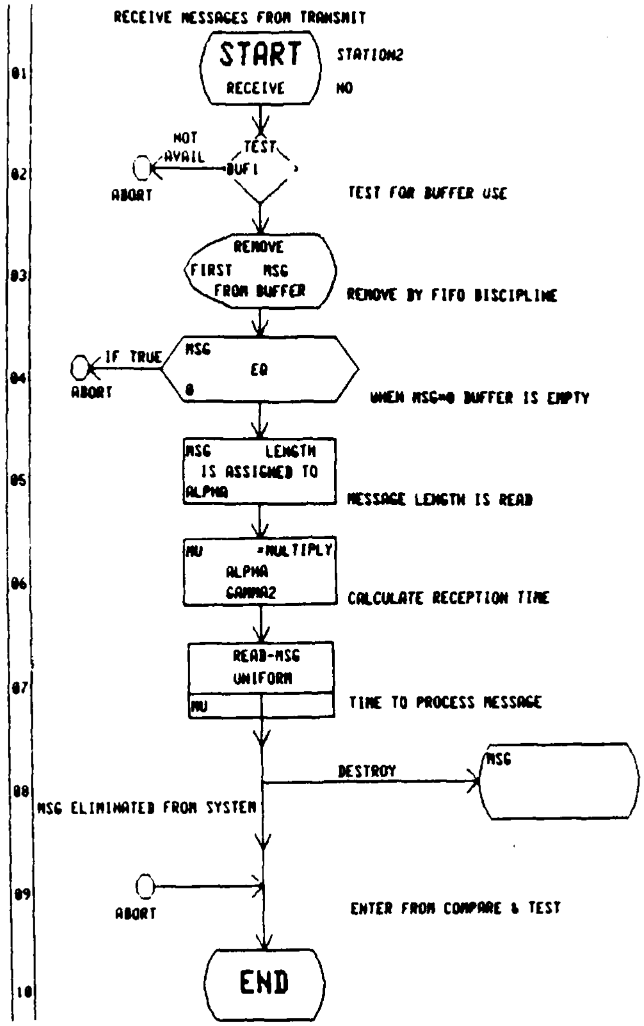


Figure 62. Receiving Process

5.2 REMAINING MODEL ELEMENTS

The remaining model entities must now be defined. These include all the entities mentioned in any Process Primitive. These are the following:

1) The Queue named "BUFFER" onto which messages are filed;

2) The Resource BUF1 which represents a device to protect the buffer against manipulation by two Processes at once;

3) The Item MSG, each instance of which is to represent a message transmitted onto the Queue;

4) The global Variables.

5) The Action entities.

5.2.1 RESOURCE DEFINITIONS

The Resource BUF1 is given proper parameters. It will have only one initial unit and will have a maximum of one. It will retain the default of no attributes and a cost of zero. An appropriate description is: "RESOURCE ASSOCIATED WITH BUFFER".

5.2.2 QUEUE DEFINITIONS

The Queue called "BUFFER", which is accessed by the FILE and REMOVE Primitives, will retain its default value of "INFINITE" holding capacity. A helpful description is: "BUFFER ON WHICH MESSAGES ARE STORED".

5.2.3 ITEM DEFINITION

The Item MSG which represents messages transmitted and received will have one attribute called "LENGTH". Its initial value will be the Literal "$LENGTH", since the value of this attribute will always be assigned within the Process that transmits it to the buffer.

5.2.4 VARIABLE DEFINITION

The Variables "GAMMA1" and "GAMMA2" are defined with initial values of .700 and .002 respectively. These values are used in calculating the transmition and reception time. GAMMA1 is the average message length, and GAMMA2 is the transmition rate.

5.2.5 ACTION DEFINITION

Action entities "SENDING" and "READ-MSG" must be defined in order to satisfy the references in the action Primitives in the Processes TRANSMIT and RECEIVE. The class and description fields can be filled in as desired by the user. These fields have no effect on the simulation.

5.3 LOADS AND SCENARIOS

Finally, we must define the hypothetical conditions to which the modeled system will be exposed. Six Loads are defined for this model. L1, L2 and L3 each trigger the transmitting Process. L11, L22 and L33 each trigger the receiving Process in a schedule of expected need associated with L1, L2 and L3. The triggerings of the transmitting Process are randomized, whereas the triggerings of the receiving Process are scheduled at regular intervals. The complete Load definitions are found on pages 3 through 5 of the Model Verification Report which appears in Appendix A.

The Scenario for this model consists of six periods. The Loads are distributed throughout the simulation period as follows: each pair of Loads is triggered at intervals of 200 units on the simulation clock. The complete definition is found on page 5 of the Model Verification Report in Appendix A.

6. SIMULATION EXERCISES OF AISIM MODELS

The model is now ready to be run through a simulation exercise to determine its behavior under the defined environmental conditions. To begin this exercise, enter the Analysis User Interface (AUI) from the AISIM READY level by typing,

A P(projectname)

Projectname is the name of the model we wish to expose to a simulation exercise. The user will be prompted with information that will look something like that shown in Figure 63.

```
CURRENT PARAMETERS IN EFFECT:
VERSION:        PRODUCTION
PROJECT:        TESTDBC
USER:           TF01508
XLATE/NOXLATE: XLATE
'ENTER YES TO PROCEED, NO TO ABORT...'
```

Figure 63. Information Displayed on Entering The AUI

After declining the abort prompt by typing

YES (cr)

and following the translation of the model, the user is in a position to issue commands before the execution of the simulation.

6.1 INITIALIZING A MODEL

If more than one Scenario has been defined, the system will ask,

WHICH SCENARIO DO YOU WISH TO TRANSLATE?

Type the name of the Scenario that defines the environmental conditions to which the model is to be subjected. For this model, we have defined only one Scenario so the program will perform model initialization. If no errors are detected at this stage the computer will prompt with,

NO ERRORS DETECTED DURING MODEL INITIALIZATION
YOU MAY NOW ENTER COMMANDS

If an error had been made the computer would have prompted with,

ERRORS DETECTED IN MODEL INITIALIZATION

This prompt indicates that some aspect of the model definition is in error. If such is the case, determine what the errors are, see Appendix B of the AISIM User's Manual, and return to the DUI to correct them. The matter of getting to the DUI has already been covered in previous chapters. For this example, assume that the AISIM model is properly defined.

6.2 DEFINING PLOTS

Two choices are available at this point: Proceed to the simulation exercise of the model or request that graphs of some of the activities monitored during the simulation be defined so that they can later be inspected at the terminal.

For example, in the model under consideration, one of our main concerns is to determine whether the buffer onto which the transmitting Process places messages (and from which the receiving Process retrieves the messages) reaches some maximum burden or whether it shows a tendency to infinite queueing. To produce a graph of the behavior of the buffer we type,

DEFPLOT QUEUE,BUFFER

The screen will display selection of aspects of the behavior of a Queue of which a graph can be defined. These are shown in Figure 64.

```
ATTRIBUTES (PLACE AN X NEXT TO ONLY ONE)

  NUMBER IN QUEUE
  NUMBER BLOCKED
  TIME IN QUEUE
  TIME BLOCKED
```

Figure 64. Aspects of Queue Behavior

To define a graph showing the number of Items in the Buffer we would enter an "x" for "NUMBER IN QUEUE". The screen would then display the options for defining the type statistic on the number of Items in the Queue. These options are shown in Figure 65.

STATISTICS (PLACE AN X NEXT TO ONLY ONE)

CURRENT
CUMULATIVE MEAN
CUM STANDARD DEV
CUMULATIVE MIN
CUMULATIVE MAX
PERIOD MEAN
PER STANDARD DEV
PERIOD MIN
PERIOD MAX

Figure 65. Options for Statistics

To calculate the current number of message Items in the Queue called "BUFFER" at any given time, enter an "x" next to "CURRENT". The entities with respect to which graphs can be defined are Resources, Queues, Processes, Items and Variables. Up to ten such graphs may be defined per analyze session.

6.3 STARTING THE SIMULATION

Once the model is initialized and graphs are defined the model may be executed through a simulation run. The execution of the model may be triggered either for the entire Scenario or for a specified number of periods, so that global Variables can be given initial values different from those previously defined in the DUI.

The values of Constants and the initial values of global Variables may also be changed before a simulation exercise begins. The latter option will be chosen in this example to investigate the effect of altering the time required to transmit or to process message Items. To begin the simulation, type,

    GO 1

This command indicates that the simulation is to be run for 1 of the 10 simulation periods defined when the model was created in the DUI. When this first stage of the simulation is completed the screen will offer the following message:

END OF PERIOD
YOU MAY NOW ENTER COMMANDS

6.4 EDITING VARIABLES BETWEEN SIMULATION STAGES

To change the value of a variable, issue the appropriate command with information as to (a) the type of entity to be edited, (b) the name of the entity whose value is to be changed and (c) the new numerical value of the entity. The Variable gamma2 formerly had the value of .002. To change it to .001, type,

E V,gamma2,.001

The simulation may be continued for two more periods by typing,

GO 2

When this stage of the simulation is completed the value of Variables may be changed back to .002 by typing,

E V,gamma2,.002

To command that the the remainder of the Scenario be run through without further interruption, type the GO command without a numeric parameter, thus:

GO

If no mistakes were made in constructing the model that cause the simulation to abort, the computer will prompt, after some time, that the simulation is completed.

The output report for this simulation run appears in Appendix A.

7. A MORE ELABORATE EXAMPLE

In this chapter a communication system slightly more complicated than that designed in Chapter 5 and analyzed in Chapter 6 is constructed. To do this, however, requires that we introduce one further AISIM feature.

7.1 MESSAGE ROUTING SUBMODEL

When one Process is triggered by another through a Call primitive, the called Process will execute in the same architectural node as the one that triggered it, i.e. utilize the same Resource, even if the two Processes are normally associated with different nodes. This is inconvenient in the representation of communication systems in which an activity in one hardware element causes activity in another one. AISIM therefore embodies a submodel to represent the situation in which a Process in one node triggers a Process in another node by communicating through the network architecture. This submodel consists of a collection of Processes and one Item.

The Processes that accomplish this must be placed in a project database with the commands available in the Library User Interface. The entities of this submodel need not be defined anew. For information on the use of this facility, see the AISIM User's Manual, Section 10.

7.2 DEFINING ARCHITECTURAL ELEMENTS

Consider modeling a communication system between two airbases, a headquarters and a command headquarters that communicates directly with a computer disk. Between these end-points are switches that govern the routing of messages through the system. The physical layout of this system is shown in Figure 66.

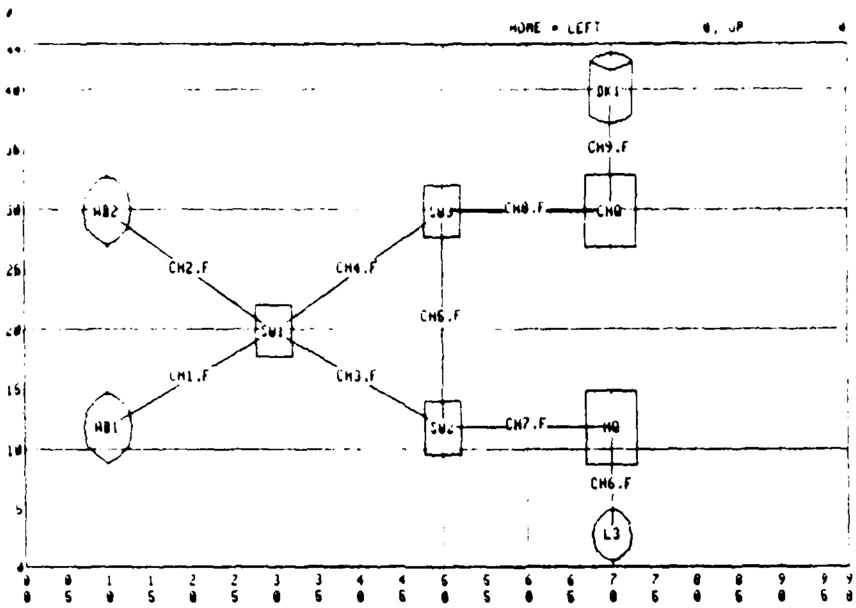


Figure 66. System Architecture

For this example, the shortest paths between the nodes will be used. Therefore, subsequent to defining the architecture, method B is used to create the Legal Path Table. The resulting table is depicted in the analysis report given in Appendix B.

The operations associated with this architecture are as follows. Both airbases periodically broadcast messages to the other nodes in the system and request plans from the command headquarters.

The effect of each broadcast is to (1) stimulate processing in the HQ and CHQ and to cause the updating of information in all other nodes. Periodically also an applications program in L3 requests plans from the CHQ, as do also AB1 and AB2. The effect of any such request is to engage the operation of the disk that communicates directly (and only) with the CHQ.

This description of the main operations of the system implies the following more rigorous listing of the Processes that need to be defined to represent such a system. The Processes required will be:

--A Process to represent the request from the HQ to the CHQ for plans. It will execute in the HQ node and will trigger a Process in the CHQ node.

--A Process to represent the broadcast of data from AB1 and AB2 to all other nodes. This Process will execute in the nodes AB1 and AB2 and will trigger (a) an updating Processes in (a) the HQ node, (b) the CHQ node and (c) each other, i.e., a broadcast in one airbase will update information in the other.

--A Process to represent the updating activity that occurs in the CHQ, triggered by broadcasts from the airbases.

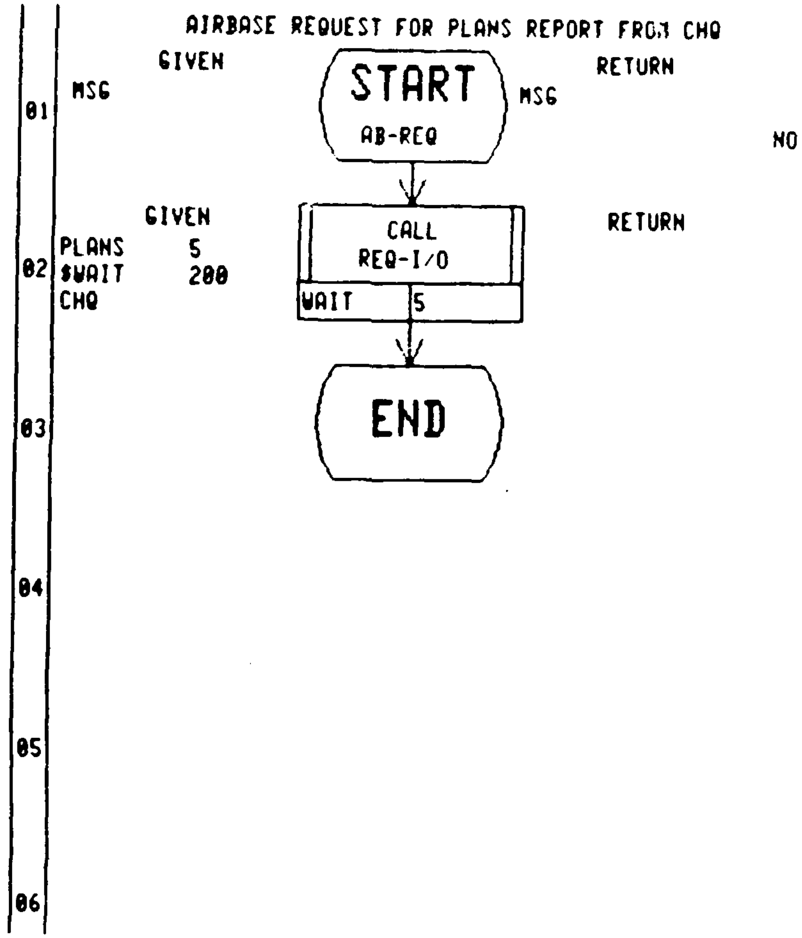
--A Process to represent the updating activity that occurs in the HQ that is triggered by broadcasts from the air-bases.

--A Process to represent the updating activity in the air-bases which is triggered by broadcasts from one another.

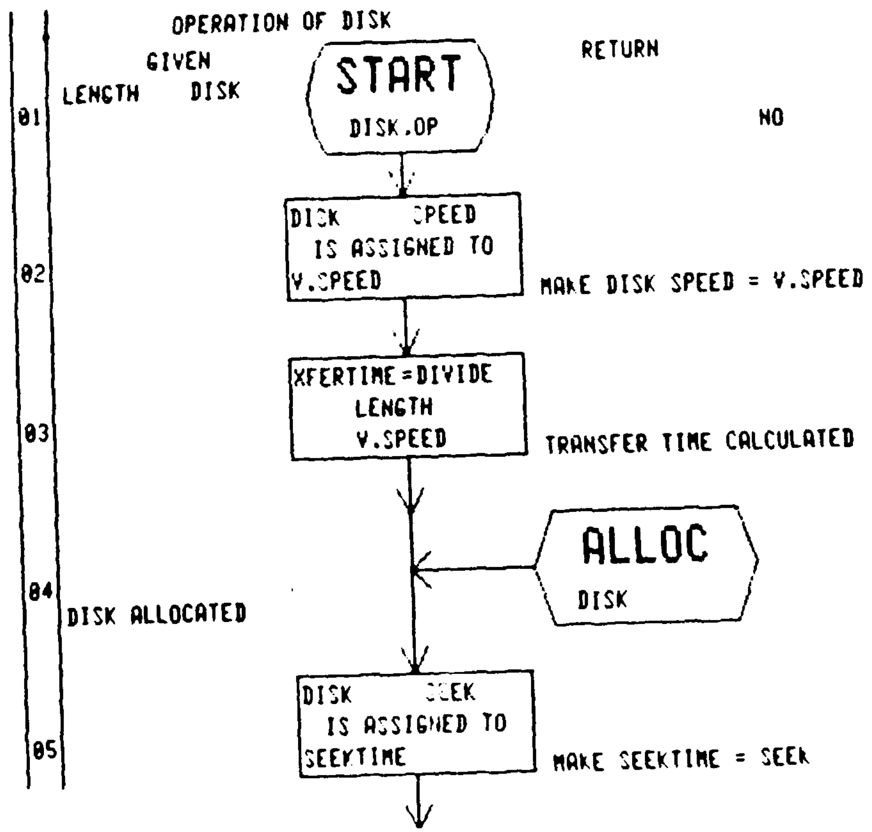
--A Process to represent the formulation of plans at the CHQ, which is triggered by requests from AB1, AB2 and HQ. This Process executes in the CHQ node and triggers another Process representing disk operation in the Disk node.

--A Process to represent the operation of the disk that communicates with the CHQ node.

These descriptions can be used to generate the AISIM Process definitions found on the following pages. The Process flow-charts for each are displayed, together with annotations to clarify the rationale for the steps that might otherwise be obscure.



The Process is given the value of the variable MSG which happens to be an Item.



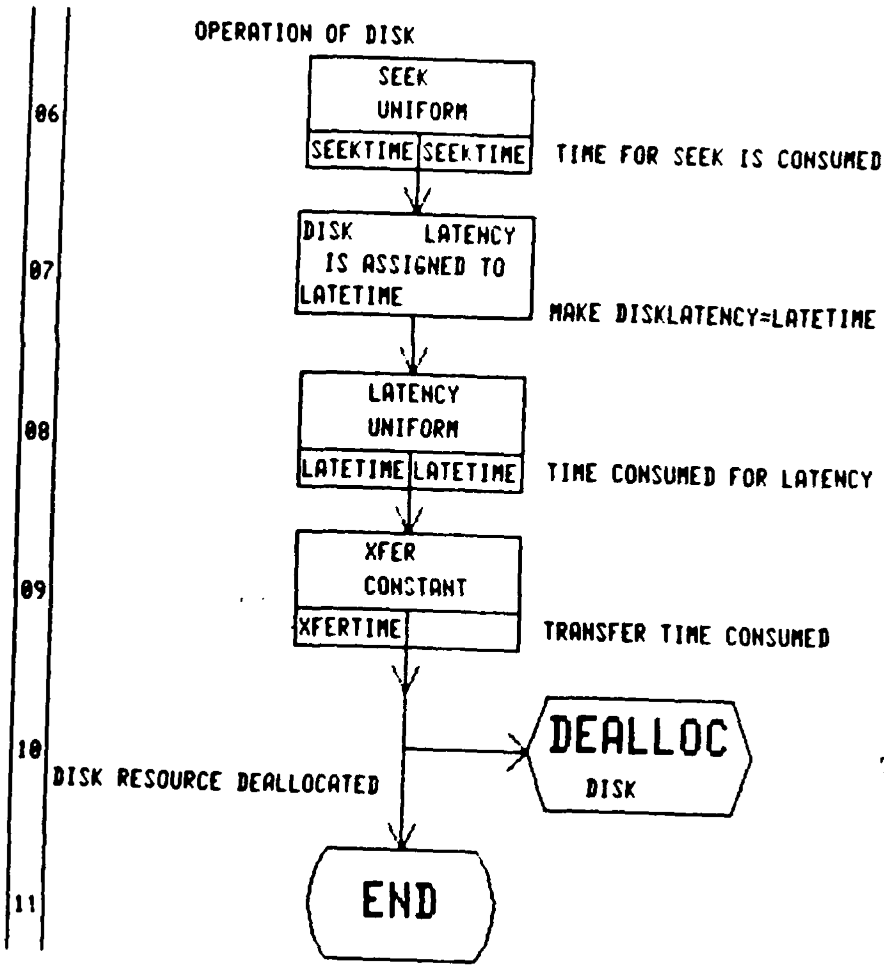
The Process is given the variables LENGTH and DISK which resolve, respecitively, to a numeric and a Resource.

The attribute of the (Resource) DISK called SPEED is assigned to the local variable V.SPEED.

The transfer time is set equal to the length of the message divided by v.speed.

The Resource DISK is allocated.

The value of the disk attribute SEEK is assigned to the local variable SEEKTIME.



Time is taken up finding data on the disk in the amount of SEEKTIME.

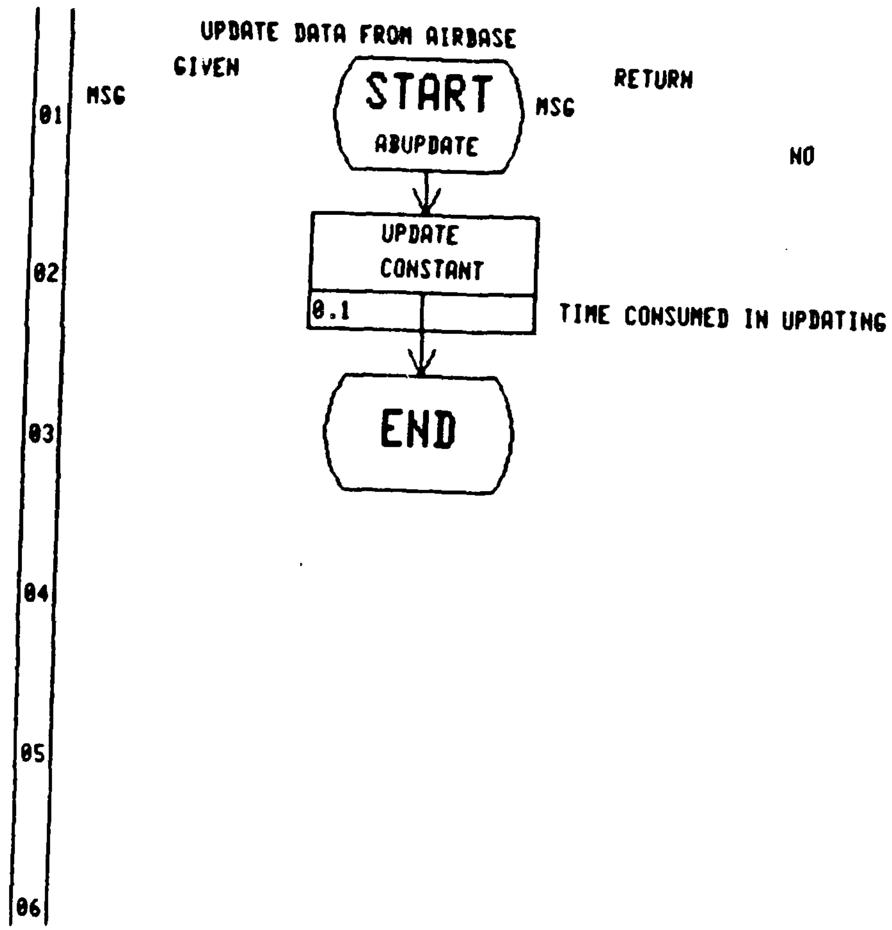
The value of the attribute (of the Resource "DISK") is assigned to the local variable "LATETIME".

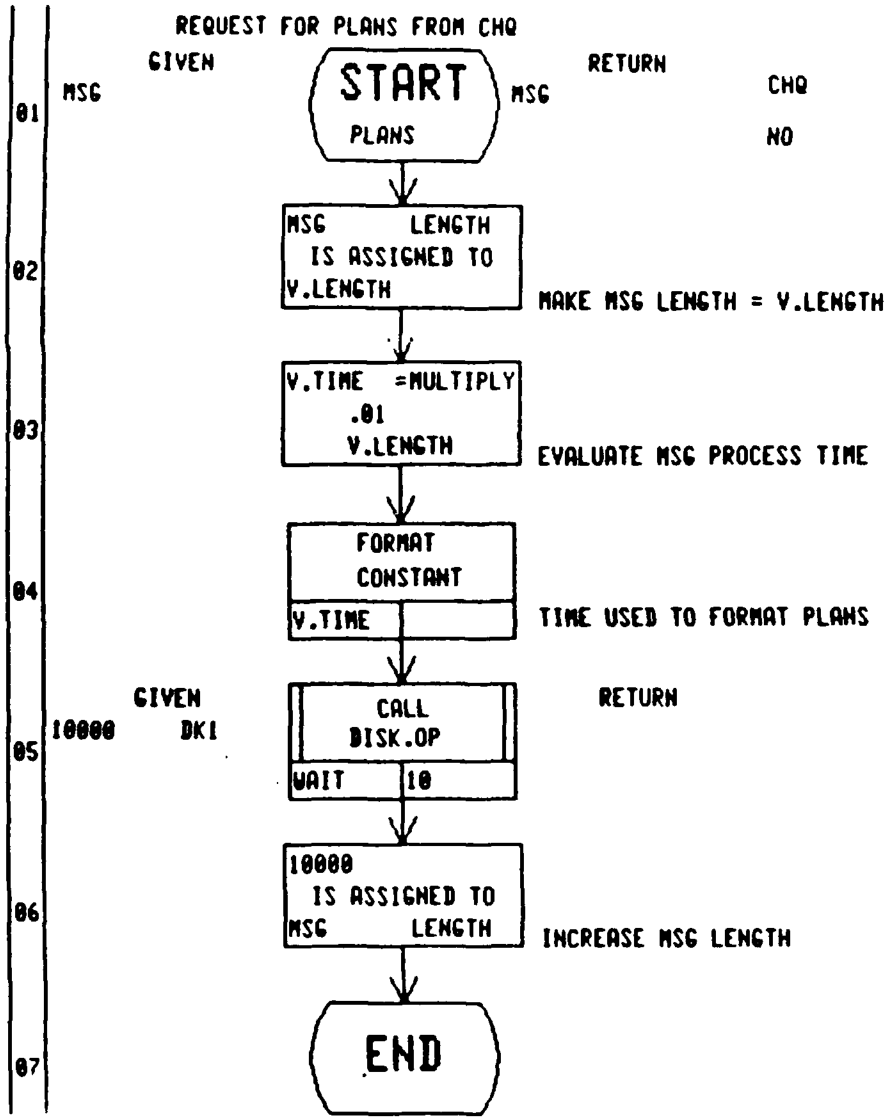
Time is consumed in the amount of LATETIME to represent latency.

Time is consumed to represent transfer time in the constant amount carried by the global constant XFERTIME.

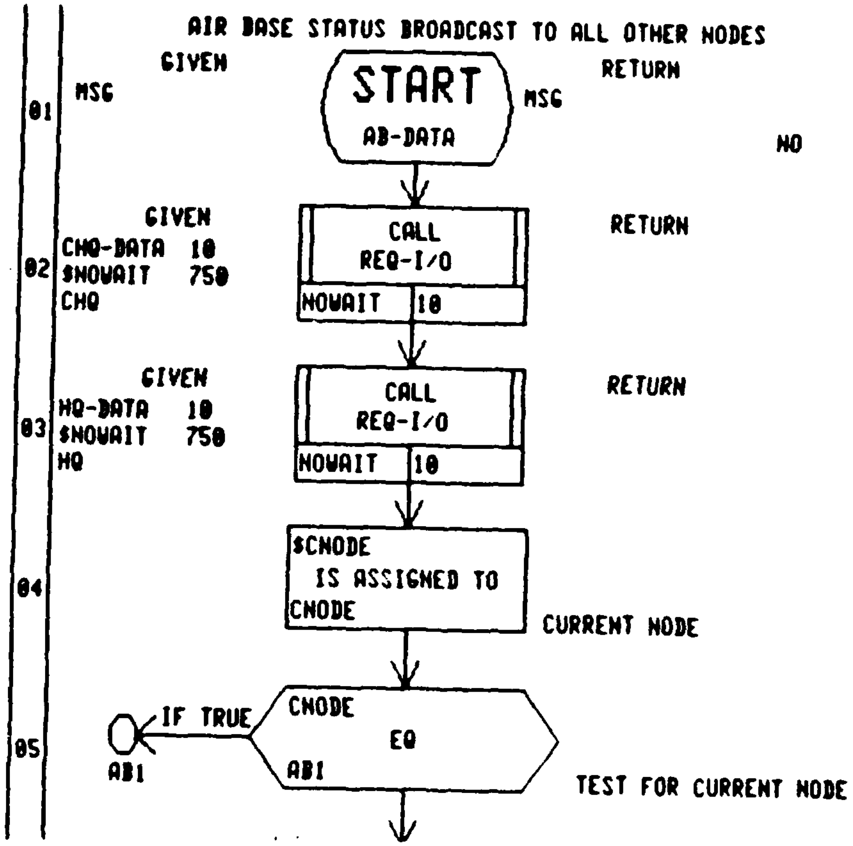
The disk is deallocated.

This Process differs from HQ-DATA and CHQ-DATA only in the Node in which it takes place.





The Process is given the local variable MSG which denotes an Item.

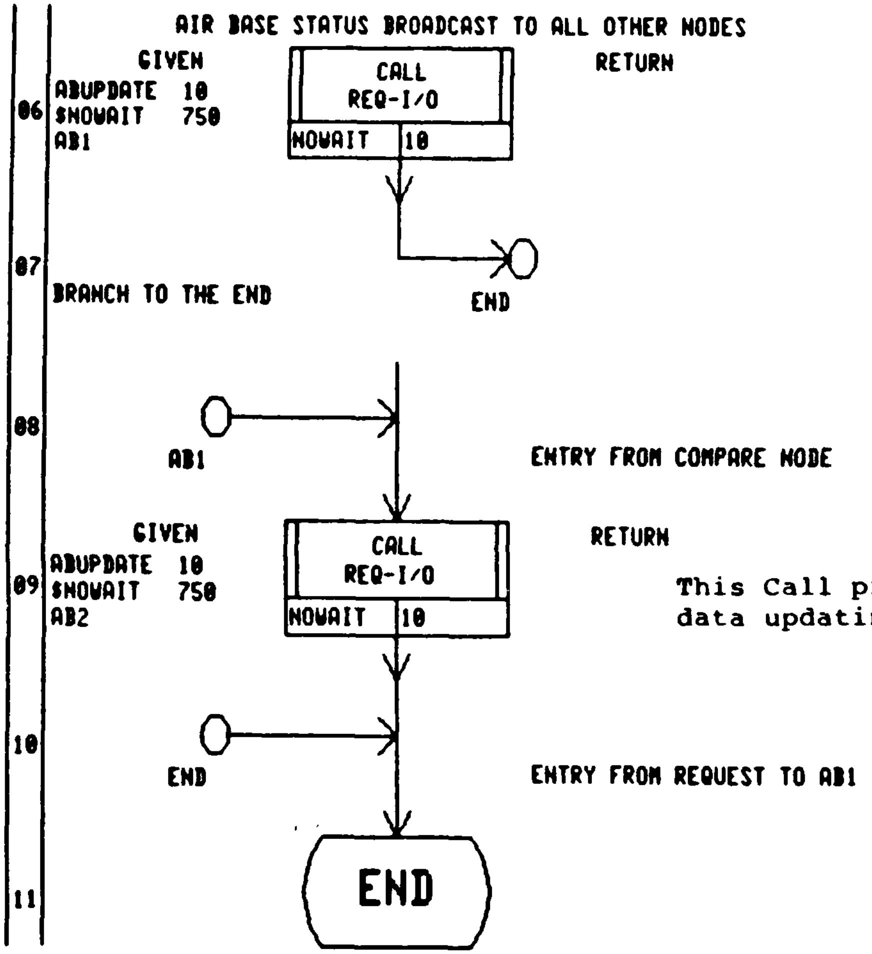


This primititive triggers a data update in the CHQ.

This primitive triggers a data update in the HQ.

The literal "$CNODE" is assigned to the keyword "CNODE" which denotes the Node in which the Process is currently triggering. Remember that the same Process "AB-DATA" takes place in two different Nodes, AB1 and AB2.

This primitive determines whether the Node in which the Process is presently executing is AB1 or AB2. If it is in AB1 it branches to the the Entry primitive labled "AB1" below.



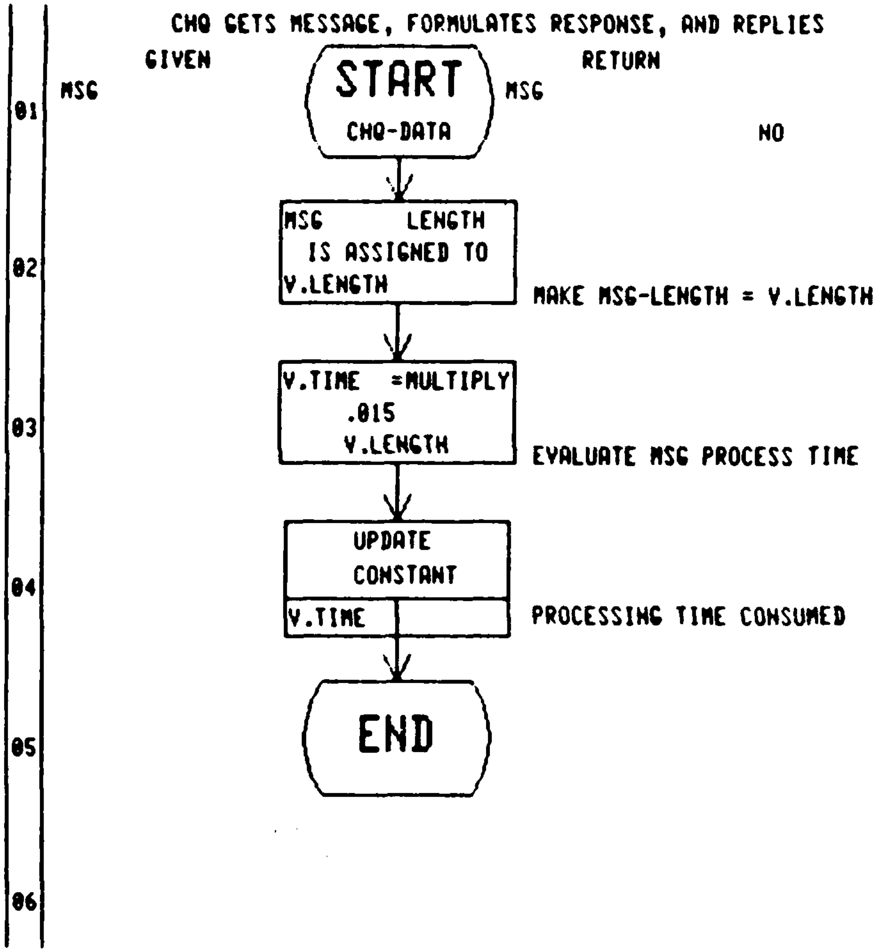
This Call primitive triggers a data update in the AB1 Node.

This End primitive branches to the Entry primitive labled End below.

This Entry primitive is the one branched to if the Process is executing in the AB1 Node.

This Call primitive triggers data updating in the AB2 Node.

This Entry primitive is the one branched to subsequent to the calling of an updating Process in AB2.

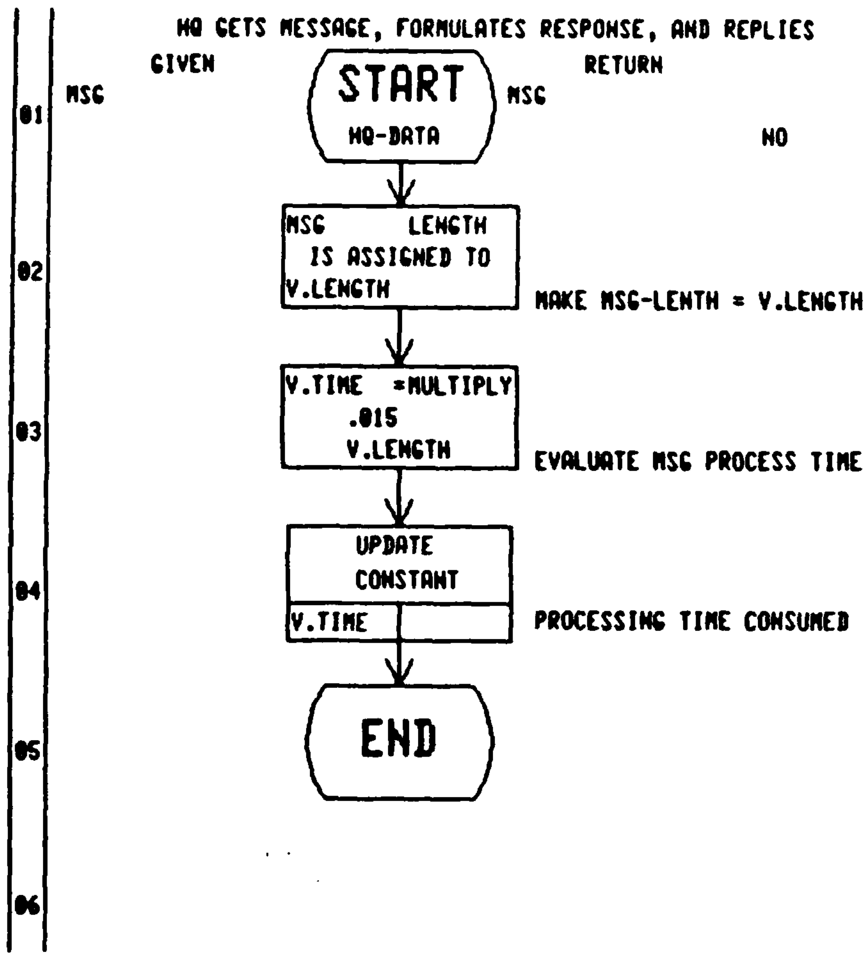


The Process is given the value of a local variable "MSG" which in this case resolves to an Item. representing a transient data element.

The attribute carried by the Item MSG called "LENGTH" is assigned to the variable "V.LENGTH".

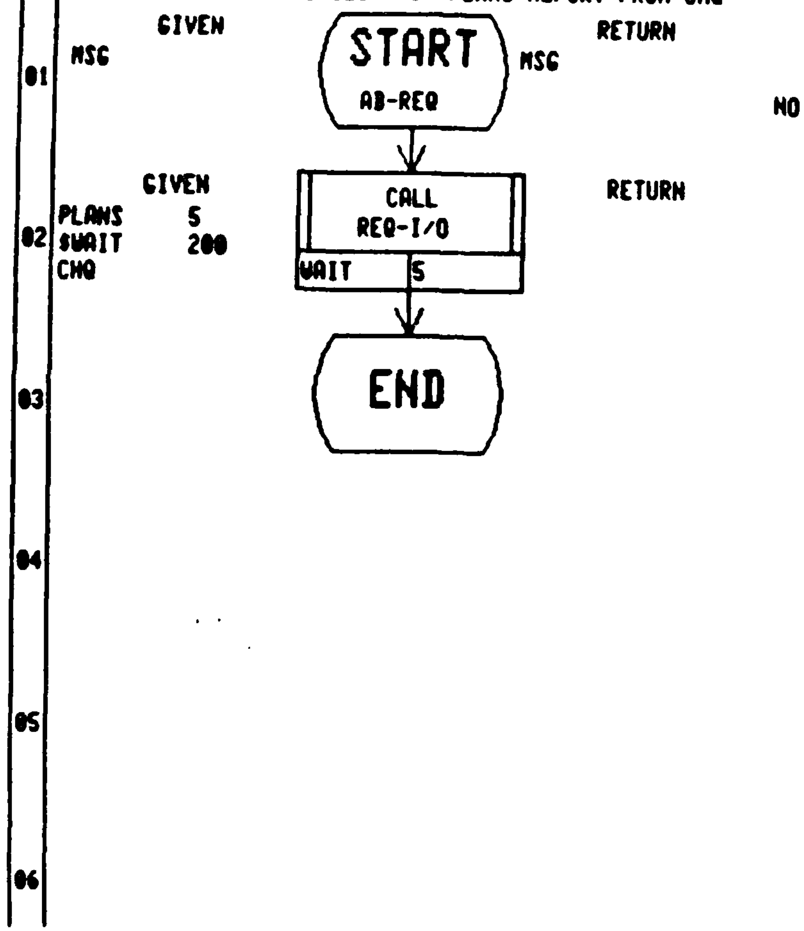
The variable V.TIME which represents the time required to process the data from the airbases is calculated as (V.LENGTH) X (.015).

This Action primitive cons... time equal to the value of V.TIME calculated above.

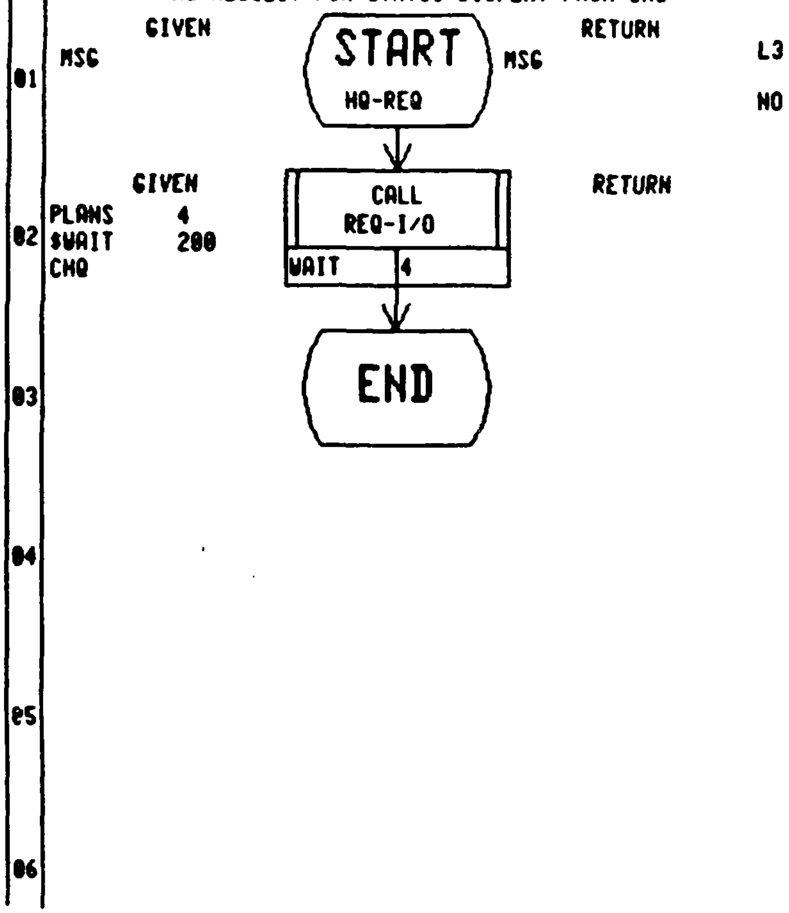


This Process differs from CHQ-DATA only in the Node in which it takes place.

AIRBASE REQUEST FOR PLANS REPORT FROM CHQ



HQ REQUEST FOR STATUS DISPLAY FROM CHQ



7.3 DEFINING REMAINING MODEL ELEMENTS

The components of the model. These must include (1) all Resources accessed in the Processes (2) all Actions which appear as Primitives; (3) all Constants and global Variables invoked in the Processes; (4) Loads; and (5) Scenarios.

7.3.1 RESOURCE DEFINITIONS All the Resources necessary to this model will have been defined automatically with default values while representing the physical layout represented in the Architecture Design Editor. Thus, if the nodes and links have the same names as in Figure 66 above, the following list of Resources will already exist in the model database.

AB1	RESOURCE FOR NODE
AB2	RESOURCE FOR NODE
CHQ	COMMAND HEAD-QUARTERS
CN1.A	RESOURCE FOR CHANNEL CONNECTOR
CN1.B	RESOURCE FOR CHANNEL CONNECTOR
CN2.A	RESOURCE FOR CHANNEL CONNECTOR
CN2.B	RESOURCE FOR CHANNEL CONNECTOR
CN3.A	RESOURCE FOR CHANNEL CONNECTOR (DOUBLE NORMAL SPEED)
CN3.B	RESOURCE FOR CHANNEL CONNECTOR (DOUBLE NORMAL SPEED)
CN4.A	RESOURCE FOR CHANNEL CONNECTOR (DOUBLE NORMAL SPEED)
CN4.B	RESOURCE FOR CHANNEL CONNECTOR (DOUBLE NORMAL SPEED)
CN5.A	RESOURCE FOR CHANNEL CONNECTOR (DOUBLE NORMAL SPEED)
CN5.B	RESOURCE FOR CHANNEL CONNECTOR (DOUBLE NORMAL SPEED)
CN7.A	RESOURCE FOR CHANNEL CONNECTOR
CN7.B	RESOURCE FOR CHANNEL CONNECTOR
CN8.A	RESOURCE FOR CHANNEL CONNECTOR
CN8.B	RESOURCE FOR CHANNEL CONNECTOR
CN9.A	RESOURCE FOR CHANNEL CONNECTOR
CN9.B	RESOURCE FOR CHANNEL CONNECTOR
DK1	DISK FOR COMMAND HEAD-QUARTERS
HQ	HEAD-QUARTERS
L3	RESOURCE FOR NODE
SW1	SWITCH BETWEEN AIRBASES AND OTHER TWO SWITCHES (1&2)
SW2	SWITCH BETWEEN SWITCH 1 & 3 AND HQ
SW3	SWITCH BETWEEN SWITCH 1 & 2 AND CHQ

Figure 67. Defined Resource Entities

Since these Resources are created with default values (an initial unit of 1, a maximum unit of 1, no attributes attached, a cost of 0, and no description), they must be edited to provide helpful descriptions and to give them attributes since attributes of these Resources are accessed in several places in the Message Routing Submodel. Descriptions and attributes can be edited before generating the architecture using the ADE's DEFINE command. See Section 2.1 of this manual.

7.3.2 FILLING IN THE ACTION DEFINITIONS The Action Primitives invoked in the Processes must have corresponding Action entity definitions outside the Process. The ones invoked are these:

[illegible].OH	CHQ PROCESSING OF GRAPHICS REQUEST
[illegible]D.CH	CHQ PROCESSING OF HARD COPY REQUEST
CS.OH	PROCESSING TO PERFORM CONTEXT SWITCHING
DUMMYACT	ACTION TO ENABLE CYCLIC PROGRAM CYCLES
FORMAT	TIME USED TO FORMAT PLANS FROM CHQ
HQ.OH	HQ PROCESSING OF MESSAGE
LATENCY	LATENCY PAUSE SUBSEQUENT TO SEEK
OVERHEAD	TIME FOR GENERAL USE
ROUTE.OH	PROCESSING DELAY TO ROUTE A MESSAGE
SEEK	SEEKING INFORMATION ON DISK
UPDATE	UPDATING INFO SINCE PREVIOUS BROADCAST TO OTHER NODES
XFER	TRANSFER INFORMATION SOUGHT ON DISK
XFER.OH	PROCESSING DELAY TO ROUTE A MESSAGE OVER A CHANNEL

Figure 68. Defined Action Entities

7.3.3 <u>CONSTANTS AND GLOBAL VARIABLES</u> This model contains five global Variables (ABDRATE, ABRATE, HQRATE, TIME1 and VRATE) and one Constant (V.TRACE). Their defined values and descriptions (which explain their role in the model) are as follows:

ABDRATE	Interval between signals.
ABRRATE	Interval between signals.
HQRATE	Interval between signals.
TIME1	Average seek times for disk in milliseconds.
VRATE	Switch to other node channel speed in ms/byte

7.3.4 <u>DEFINING LOADS AND SCENARIOS</u> In this model we wish to represent the several Process triggerings that are due to causes outside the system. First, the AB1 and AB2 will broadcast communications to the other nodes (which trigger updating Processes in them) every minute by an interval scheduling method. In addition, AB1 and AB2 will issue requests for plans from the CHQ sixty times in one hour by an exponential scheduling method. We define a second Load to represent requests from the leaf-node, L3, also for plans from the CHQ. This Process will also be undertaken sixty times per hour, exponentially distributed. The Load definitions implied by these requirements are printed in Appendix B.

The length of the entire Scenario is 360,000 milliseconds (one hour), which is divided into ten periods of six minutes each. To simulate the operation of the system with the worst case, we stipulate that both of the functional Loads are triggered simultaneously, at the beginning of the Scenario. In addition, as a monitoring device, we initiate the Trace Process at the beginning of the simulation run. The parameters for the Scenario implied by these requirements are printed page 18 of the analyze report in Appendix B.

7.4 ANALYZING THE MODEL

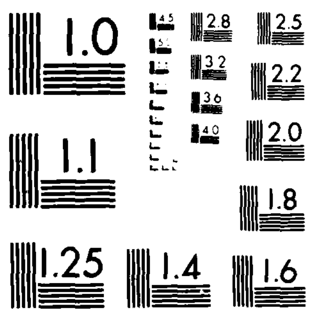
To run the model through a simulation test, invoke the Analysis User Interface from the AISIM READY level. For this example, the simulation will not be interrupted at the ends of periods, nor will graphs be defined.

The analyze report obtained from a simulation run of this model appears in Appendix B.

AD-A135 882 AISIM (AUTOMATED INTERACTIVE SIMULATION MODEL) TRAINING MANUAL(U) HUGHES AIRCRAFT CO FULLERTON CA GROUND SYSTEMS GROUP J HEARNE ET AL. 26 FEB 82 ESD-TR-83-217 2/2

UNCLASSIFIED F19628-79-C-0153 F/G 9/2 NL

END
DATE
FILMED
1 - 84
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

APPENDIX A

SIMULATION REPORT FOR WORKING EXAMPLE

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$                     S I M U L A T I O N   R E P O R T                $
$                            AISIM VERSION 2.0                         $
$                         HUGHES AIRCRAFT COMPANY                      $
$                                08/03/81                              $
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
GLOBAL CONSTANT DEFINITION.....

CONSTANT INITIAL
MNEMONIC VALUE    COMMENT
======== ======== ==================================================

TABLE DEFINITION....


GLOBAL VARIABLE DEFINITION.....

VARIABLE INITIAL
MNEMONIC VALUE    COMMENT
======== ======== ==================================================
GAMMA1   700
GAMMA2   .002

ITEM DEFINITION......


ITEM                       DESCRIPTION
======== ================= ==========================================
MSG                        MSG TO BE TRANSMITTED
         ATTR.    INITIAL
         NAME     VALUE
         ======== ========
         LENGTH   $LENGTH


QUEUE DEFINITION.....

QUEUE    MAXIMUM
MNEMONIC   SIZE   COMMENT
======== ======== ==================================================
BUFFER   INFINITE BUFFER ON WHICH MESSAGES ARE STORED

RESOURCE DEFINITION.....

RESOURCE TOTAL    INITIAL
MNEMONIC # UNITS  # UNITS  DESCRIPTION
======== ======== ======== =========================================
```

```
PAGE     2
BUF1     1        1        RESOURCE ASSOCIATED WITH BUFFER
         ATTR.    INITIAL
         NAME     VALUE
         ======== ========
         COST     0

ARCHITECTURE LEGAL PATH DEFINITION

FROM       TO         NEXT       VIA
DEVICE     DEVICE     DEVICE     LINK
========   ========   ========   ========

ACTION DEFINITION.....

ACTION   ACTION
MNEMONIC CLASS    COMMENT
======== ======== ==================================================
READ-MSG MACHINE  READ A MESSAGE
SENDING  MACHINE  TRANSMIT A MESSAGE

PROCESS DEFINITION.....

PROCESS
MNEMONIC          DESCRIPTION
======== ======== ==================================================
RECEIVE           RECEIVE MESSAGES FROM TRANSMIT


ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
======== ======== ======== ======== ======== =========================
         START             NO
         TEST     BUF1     ABORT             TEST FOR BUFFER USE
         REMOVE   FIRST    MSG      BUFFER   REMOVE BY FIFO DISCIPLINE
         COMPARE  MSG               EQ       WHEN MSG=0 BUFFER IS EMPTY
                  0                 ABORT
         ASSIGN   MSG      LENGTH            MESSAGE LENGTH IS READ
                  ALPHA
         EVAL     MU       MULTIPLY          CALCULATE RECEPTION TIME
                  ALPHA    GAMMA2
         READ-MSG UNIFORM  MU                TIME TO PROCESS MESSAGE
         DESTROY  MSG                        MSG ELIMINATED FROM SYSTEM
ABORT    ENTRY                               ENTER FROM COMPARE & TEST
         END

LOCAL VARIABLES OF PROCESS RECEIVE
===================================================================
    1 BUF1      (R)    2 MSG       (I)    3 BUFFER    (Q)    4 ALPHA
    5 MU               6 READ-MSG  (A)    7
PROCESS
```

```
PAGE    3
MNEMONIC          DESCRIPTION
======== ======== ==================================================
TRANSMIT          TRANSMITTING MESSAGES TO RECEIVER

ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
======== ======== ======== ======== ======== =======================
         START             NO
         ALLOC    BUF1                       ALLOCATE BUF1
         CREATE   MSG                        INTRODUCE MSG INTO SYSTEM
         EVAL     ALPHA    RANDOM            GENERATE RANDOM NUMBER

         EVAL     ALPHA    MULTIPLY          TWICE AVERAGE TIMES ALPHA
                  ALPHA    GAMMA1
         ASSIGN   ALPHA                      SET MESSAGE LENGTH
                  MSG      LENGTH
         EVAL     MU       MULTIPLY          CALCULATE TRANSMIT TIME
                  ALPHA    GAMMA2
         SENDING  UNIFORM  MU                TIME CONSUMED TRANSMITTING
         ASSIGN   ALPHA                      SET MESSAGE LENGTH
                  MSG      LENGTH
         FILE     MSG      LAST     BUFFER   STORE MSG ON BUFFER
         DEALLOC  BUF1                       RELEASE RESOURCE BUF1
         END

LOCAL VARIABLES OF PROCESS TRANSMIT
==================================================================
    1 BUF1      (R)   2 MSG       (I)    3 ALPHA           4 MU
    5 SENDING   (A)   6                  7 BUFFER    (Q)

LOAD DEFINITION.....

LOAD
MNEMONIC          DESCRIPTION
======== ======== ==================================================
L1
         LOAD     NODES
         ======== ======== ======== ========


         PROCESS           SCHEDULE
         MNEMONIC MAX #    METHOD   MEAN     DELTA    PRIORITY
         ======== ======== ======== ======== ======== ========
         TRANSMIT 300      POISSON                    0
LOAD
MNEMONIC          DESCRIPTION
======== ======== ==================================================
L11
```

```
PAGE     4
          LOAD     NODES
          ======== ======== ======== ========


          PROCESS           SCHEDULE
          MNEMONIC MAX #    METHOD   MEAN     DELTA    PRIORITY
          ======== ======== ======== ======== ======== ========
          RECEIVE  600      INTERVAL                   0
LOAD
MNEMONIC          DESCRIPTION
======== ======== ==================================================
L2
          LOAD     NODES
          ======== ======== ======== ========


          PROCESS           SCHEDULE
          MNEMONIC MAX #    METHOD   MEAN     DELTA    PRIORITY
          ======== ======== ======== ======== ======== ========
          TRANSMIT 200      POISSON                    0
LOAD
MNEMONIC          DESCRIPTION
======== ======== ==================================================
L22
          LOAD     NODES
          ======== ======== ======== ========


          PROCESS           SCHEDULE
          MNEMONIC MAX #    METHOD   MEAN     DELTA    PRIORITY
          ======== ======== ======== ======== ======== ========
          RECEIVE  400      INTERVAL                   0
LOAD
MNEMONIC          DESCRIPTION
======== ======== ==================================================
L3
          LOAD     NODES
          ======== ======== ======== ========


          PROCESS           SCHEDULE
          MNEMONIC MAX #    METHOD   MEAN     DELTA    PRIORITY
          ======== ======== ======== ======== ======== ========
          TRANSMIT 100      POISSON                    0
LOAD
```

```
PAGE     5
MNEMONIC          DESCRIPTION
======== ======== ================================================
L33
         LOAD     NODES
         ======== ======== ======== ========


         PROCESS           SCHEDULE
         MNEMONIC MAX #    METHOD   MEAN     DELTA    PRIORITY
         ======== ======== ======== ======== ======== ========
         RECEIVE  200      INTERVAL                   0

SCENARIO DEFINITION....

SCENARIO
MNEMONIC          DESCRIPTION
======== ======== ================================================
SCEN

PERIOD
LENGTH
========
100

         PERIOD   PERIOD   PERIOD   PERIOD   PERIOD   PERIOD   PERIOD
         MNEMONIC MNEMONIC MNEMONIC MNEMONIC MNEMONIC MNEMONIC MNEMONIC
======== ======== ======== ======== ======== ======== ======== ========
         1        2        3        4        5        6

TRIGGER  TIME  TO SCHEDULE TRIGGER  TIME  TO SCHEDULE
MNEMONIC SCHEDULE PRIORITY MNEMONIC SCHEDULE PRIORITY
======== ======== ======== ======== ======== ========
L1       100      0        L11      100      0
L2       300      0        L22      300      0
L3       500      0        L33      500      0

####    0 ERRORS WERE DETECTED DURING MODEL INITIALIZATION
```

PAGE 6

SIMULATION TIME = 600.00000 UNITS

VARIABLE REPORT

NUMERIC VARIABLES...

VARIABLE	TOTAL SAMPLES.	VALUE CURRENT...	MEAN......	STD DEV...	MINIMUM...	MAXIMUM...
GAMMA1	1	700.000	700.000	0.	700.000	700.000
GAMMA2	3	.002	.002	.000	.001	.002

NON-NUMERIC VARIABLES...

VARIABLE	CURRENT TYPE	CURRENT VALUE

PAGE 7

SIMULATION TIME = 600.00000 UNITS

ITEM REPORT

ITEM NAME	NUMBER CREATED	NUMBER DESTR'D	TIME IN SYSTEM MINIMUM...	MAXIMUM...	AVERAGE...	STD DEV...
MSG	596	500	70.49	200.88	142.90	33.70

PAGE 8

SIMULATION TIME = 600.00000 UNITS

QUEUE REPORT

QUEUE	TOTAL NUMBER	CURRENT...	MEAN......	STD DEV...	MINIMUM...	MAXIMUM...
BUFFER						
FILED ON	596					
REMOVED FROM	509					
# IN QUEUE		96.000	126.479	107.532	0.	300.000
TIME IN QUEUE			141.754	33.714	67.930	199.654
TASKS BLOCKED	0					
TASKS RESUMED	0					
# BEING BLOCKED		0.	0.	0.	0.	0.
TIME BLOCKED			0.	0.	0.	0.

```
SIMULATION TIME =        600.00000 UNITS

RESOURCE REPORT

                    TOTAL
RESOURCE            NUMBER   CURRENT... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
BUF1
      INTO IDLE      597
    OUT OF IDLE      596
         # IDLE               1.000       .513       .500      0.          1.000
      IDLE TIME                            .515      6.349       .000     100.026

      INTO BUSY      596
    OUT OF BUSY      596
         # BUSY               0.          .487       .500      0.          1.000
      BUSY TIME                            .491       .348       .000       1.398

    INTO INACT.        0
  OUT OF INACT.        0
     # INACTIVE               0.         0.         0.         0.         0.
  INACTIVE TIME                          0.         0.         0.         0.

      INTO WAIT      596
    OUT OF WAIT      596
      # WAITING               0.         7.123     10.523      0.         36.000
      WAIT TIME                          7.171      6.273       .000      24.357

    CURRENTLY ALLOCATED
           TO PROCESSES: NONE

    PROCESSES CURRENTLY
                WAITING: NONE
```

SIMULATION TIME = 600.00000 UNITS

ACTION REPORT

ACTION	TOTAL SAMPLES	MEAN.....	STD DEV...	MINIMUM..	MAXIMUM...	% TIME OF TOTAL.
READ-MSG						
USEFUL TIME	500	.675	.410	.001	1.398	56.244
DELAY TIME	500	0.	0.	0.	0.	

ACTION	TOTAL SAMPLES	MEAN.....	STD DEV...	MINIMUM..	MAXIMUM...	% TIME OF TOTAL.
SENDING						
USEFUL TIME	596	.491	.348	.000	1.398	48.728
DELAY TIME	596	0.	0.	0.	0.	

```
PAGE   11

SIMULATION TIME =        600.00000 UNITS

PROCESS REPORT

                   TOTAL
PROCESS          SAMPLES. SUM....... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
RECEIVE
          TOTAL     1200    337.464       .281       .425         0.      1.398
   PROCESS WAIT        0         0.         0.         0.         0.         0.
  RESOURCE WAIT        0         0.         0.         0.         0.         0.

        TOTAL #  # AUTO   # CALL   # OF     # NOT    # TIMES
        SCHEDULE SCHEDULE SCHEDULE COMPLETE COMPLETE SUSPEND.
        ======== ======== ======== ======== ======== ========
           1200     1200        0     1200        0        0

         ITEM     CREATED  RECEIVED SENT     DESTR'D
         ======== ======== ======== ======== ========
         MSG             0        0        0      500

                     PROCESS  HOLDING  TIME
         ITEM     # SMPLS  MEAN...... MINIMUM..  MAXIMUM... STD DEV...
         ======== ======== ========== ========== ========== ==========
         MSG           500        .67        .00       1.40        .41

PROCESS           DESCR PTION
======== ======== ==================================================
RECEIVE           RECEIVE MESSAGES FROM TRANSMIT

COUNT ENTRY     OPCODE   PARM     PARM     PARM     COMMENT
===== ========  ======== ======== ======== ======== ========================
 1200           START             NO
 1200           TEST     BUF1     ABORT             TEST FOR BUFFER USE
 1200           REMOVE   FIRST    MSG      BUFFER   REMOVE BY FIFO DISCIPLINE
 1200           COMPARE  MSG               EQ       WHEN MSG=0 BUFFER IS EMPTY
 1200                    0                 ABORT
  500           ASSIGN   MSG      LENGTH            MESSAGE LENGTH IS READ
  500                    ALPHA
  500           EVAL     MU       MULTIPLY          CALCULATE RECEPTION TIME
  500                    ALPHA    GAMMA2
  500           READ-MSG UNIFORM  MU                TIME TO PROCESS MESSAGE
  500           DESTROY  MSG                        MSG ELIMINATED FROM SYSTEM
 1200 ABORT     ENTRY                               ENTER FROM COMPARE & TEST
 1200           END

                   TOTAL
PROCESS          SAMPLES. SUM....... MEAN...... STD DEV... MINIMUM... MAXIMUM...
```

TRANSMIT

TOTAL	596	4566.221	7.661	6.335	.008	25.420
PROCESS WAIT	0	0.	0.	0.	0.	0.
RESOURCE WAIT	596	4273.891	7.171	6.273	.000	24.357

TOTAL # SCHEDULE	# AUTO SCHEDULE	# CALL SCHEDULE	# OF COMPLETE	# NOT COMPLETE	# TIMES SUSPEND.
596	596	0	596	0	0

ITEM	CREATED	RECEIVED	SENT	DESTR'D
MSG	596	0	0	0

PROCESS HOLDING TIME

ITEM	# SMPLS	MEAN......	MINIMUM..	MAXIMUM...	STD DEV...
MSG	596	.49	.00	1.40	.35

PROCESS		DESCRIPTION
TRANSMIT		TRANSMITTING MESSAGES TO RECEIVER

COUNT	ENTRY	OPCODE	PARM	PARM	PARM	COMMENT
596		START		NO		
596		ALLOC	BUF1			ALLOCATE BUF1
596		CREATE	MSG			INTRODUCE MSG INTO SYSTEM
596		EVAL	ALPHA	RANDOM		GENERATE RANDOM NUMBER
596						
596		EVAL	ALPHA	MULTIPLY		TWICE AVERAGE TIMES ALPHA
596			ALPHA	GAMMA1		
596		ASSIGN	ALPHA			SET MESSAGE LENGTH
596			MSG	LENGTH		
596		EVAL	MU	MULTIPLY		CALCULATE TRANSMIT TIME
596			ALPHA	GAMMA2		
596		SENDING	UNIFORM	MU		TIME CONSUMED TRANSMITTING
596		ASSIGN	ALPHA			SET MESSAGE LENGTH
596			MSG	LENGTH		
596		FILE	MSG	LAST	BUFFER	STORE MSG ON BUFFER
596		DEALLOC	BUF1			RELEASE RESOURCE BUF1
596		END				

APPENDIX B

SIMULATION REPORT FOR ELABORATE EXAMPLE

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$                   S I M U L A T I O N   R E P O R T                 $
$                          AISIM VERSION 2.0                          $
$                       HUGHES AIRCRAFT COMPANY                       $
$                              08/03/81                               $
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
GLOBAL CONSTANT DEFINITION.....

CONSTANT INITIAL
MNEMONIC VALUE    COMMENT
======== ======== ==================================================
V.TRACE  0        DEFAULT IS NO TRACE ON

TABLE DEFINITION....


GLOBAL VARIABLE DEFINITION.....

VARIABLE INITIAL
MNEMONIC VALUE    COMMENT
======== ======== ==================================================
ABDRATE  60000    INTERVAL RATE BETWEEN SIGNALS
ABRRATE  36000    INTERVAL RATE BETWEEN SIGNALS
HCRRATE  72000    INTERVAL BETWEEN SIGNALS
TIME1    30       AVERAGE SEEK TIME FOR DISK IN MILLISECONDS
VRATE    1.6276   SWITCH-OTHER NODE  CHANNEL SPEED IN MS/BYTE

ITEM DEFINITION......


ITEM                      DESCRIPTION
======== ================ ============================================
MSG                       MESSAGE FOR INTERNODE COMMUNICATION FROM INPUT
         ATTR.    INITIAL
         NAME     VALUE
         ======== ========
         CNODE    $CNODE
         FNODE    $CNODE
         LENGTH   99999999
         PTASK    $ERROR
         RESPONSE $WAIT
         RTASK    $ERROR
         TASKPRI  99999999
         TNODE    $CNODE
         TYPE     $REQ
```

```
QUEUE DEFINITION.....

QUEUE    MAXIMUM
MNEMONIC   SIZE   COMMENT
======== ======== ================================================

RESOURCE DEFINITION.....

RESOURCE TOTAL    INITIAL
MNEMONIC # UNITS  # UNITS  DESCRIPTION
======== ======== ======== ========================================
AB1      1        1        RESOURCE FOR NODE
         ATTR.    INITIAL
         NAME     VALUE
         ======== ========
         COST     0
         NETINSTR 80
         OSOVHD   0
         SPEED    5000

AB2      1        1        RESOURCE FOR NODE
         ATTR.    INITIAL
         NAME     VALUE
         ======== ========
         COST     0
         NETINSTR 80
         OSOVHD   0
         SPEED    5000

CHQ      1        1        COMMAND HEAD-QUARTERS
         ATTR.    INITIAL
         NAME     VALUE
         ======== ========
         COST     0
         NETINSTR 80
         OSOVHD   0
         SPEED    5000

CH1.A    1        1        RESOURCE FOR CHANNEL CONNECTOR
         ATTR.    INITIAL
         NAME     VALUE
         ======== ========
         COST     0
         RATE     VRATE

CH1.B    1        1        RESOURCE FOR CHANNEL CONNECTOR
         ATTR.    INITIAL
         NAME     VALUE
         ======== ========
```

```
PAGE     3
         COST      0
         RATE      VRATE

CH2.A    1         1         RESOURCE FOR CHANNEL CONNECTOR
         ATTR.     INITIAL
         NAME      VALUE
         ======== ========
         COST      0
         RATE      VRATE

CH2.B    1         1         RESOURCE FOR CHANNEL CONNECTOR
         ATTR.     INITIAL
         NAME      VALUE
         ======== ========
         COST      0
         RATE      VRATE

CH3.A    1         1         RESOURCE FOR CHANNEL CONNECTOR
         ATTR.     INITIAL
         NAME      VALUE
         ======== ========
         COST      0
         RATE      0.4069

CH3.B    1         1         RESOURCE FOR CHANNEL CONNECTOR
         ATTR.     INITIAL
         NAME      VALUE
         ======== ========
         COST      0
         RATE      0.4069

CH4.A    1         1         RESOURCE FOR CHANNEL CONNECTOR
         ATTR.     INITIAL
         NAME      VALUE
         ======== ========
         COST      0
         RATE      0.4069

CH4.B    1         1         RESOURCE FOR CHANNEL CONNECTOR
         ATTR.     INITIAL
         NAME      VALUE
         ======== ========
         COST      0
         RATE      0.4069

CH5.A    1         1         RESOURCE FOR CHANNEL CONNECTOR
         ATTR.     INITIAL
         NAME      VALUE
         ======== ========
```

```
         COST      0
         RATE      0.4069

CH5.B    1         1         RESOURCE FOR CHANNEL CONNECTOR
         ATTR.     INITIAL
         NAME      VALUE
         ========  ========
         COST      0
         RATE      0.4069

CH6.A    1         1         RESOURCE FOR CHANNEL CONNECTOR
         ATTR.     INITIAL
         NAME      VALUE
         ========  ========
         COST      0
         RATE      VRATE

CH6.B    1         1         RESOURCE FOR CHANNEL CONNECTOR
         ATTR.     INITIAL
         NAME      VALUE
         ========  ========
         COST      0
         RATE      VRATE

CH7.A    1         1         RESOURCE FOR CHANNEL CONNECTOR
         ATTR.     INITIAL
         NAME      VALUE
         ========  ========
         COST      0
         RATE      VRATE

CH7.B    1         1         RESOURCE FOR CHANNEL CONNECTOR
         ATTR.     INITIAL
         NAME      VALUE
         ========  ========
         COST      0
         RATE      VRATE

CH8.A    1         1         RESOURCE FOR CHANNEL CONNECTOR
         ATTR.     INITIAL
         NAME      VALUE
         ========  ========
         COST      0
         RATE      VRATE

CH8.B    1         1         RESOURCE FOR CHANNEL CONNECTOR
         ATTR.     INITIAL
         NAME      VALUE
         ========  ========
```

```
         COST      0
         RATE      VRATE

CH9.A    1         1         RESOURCE FOR CHANNEL CONNECTOR .
         ATTR.     INITIAL
         NAME      VALUE
         ======== ========
         COST      0
         RATE      VRATE

CH9.B    1         1         RESOURCE FOR CHANNEL CONNECTOR
         ATTR.     INITIAL
         NAME      VALUE
         ======== ========
         COST      0
         RATE      VRATE

DK1      1         1         DISK FOR COMMAND HEAD-QUARTERS
         ATTR.     INITIAL
         NAME      VALUE
         ======== ========
         COST      0
         LATDELTA  15
         LATENCY   15
         OSOVHD    0
         SEEK      TIME1
         SPEED     20000

HQ       1         1         HEAD-QUARTERS
         ATTR.     INITIAL
         NAME      VALUE
         ======== ========
         COST      0
         NETINSTR  80
         OSOVHD    0
         SPEED     1000

L3       1         1         RESOURCE FOR NODE
         ATTR.     INITIAL
         NAME      VALUE
         ======== ========
         COST      0
         NETINSTR  80
         OSOVHD    0
         SPEED     1000

SW1      1         1         SWITCH BETWEEN AIRBASES AND OTHER TWO SWITCHES (1&2)
         ATTR.     INITIAL
         NAME      VALUE
```

```
         ======== ========
         COST     0
         NETINSTR 80
         SPEED    1000

SW2      1        1        SWITCH BETWEEN SWITCH 1 & 3 AND HQ
         ATTR.    INITIAL
         NAME     VALUE
         ======== ========
         COST     0
         NETINSTR 80
         SPEED    1000

SW3      1        1        SWITCH BETWEEN SWITCH 1 & 2 AND CHQ
         ATTR.    INITIAL
         NAME     VALUE
         ======== ========
         COST     0
         NETINSTR 80
         SPEED    1000
```

ARCHITECTURE LEGAL PATH DEFINITION

FROM DEVICE	TO DEVICE	NEXT DEVICE	VIA LINK
AB1	AB1	SW1	CH1.A
AB1	AB2	SW1	CH1.A
AB1	CHQ	SW1	CH1.A
AB1	DK1	SW1	CH1.A
AB1	HQ	SW1	CH1.A
AB1	L3	SW1	CH1.A
AB1	SW1	SW1	CH1.A
AB1	SW2	SW1	CH1.A
AB1	SW3	SW1	CH1.A
AB2	AB1	SW1	CH2.A
AB2	CHQ	SW1	CH2.A
AB2	DK1	SW1	CH2.A
AB2	HQ	SW1	CH2.A
AB2	L3	SW1	CH2.A
AB2	SW1	SW1	CH2.A
AB2	SW2	SW1	CH2.A
AB2	SW3	SW1	CH2.A
CHQ	AB1	SW3	CH8.A
CHQ	AB2	SW3	CH8.A
CHQ	DK1	DK1	CH9.A
CHQ	HQ	SW3	CH8.A
CHQ	L3	SW3	CH8.A
CHQ	SW1	SW3	CH8.A

CHQ	SW2	SW3	CH8.A
CHQ	SW3	SW3	CH8.A
DK1	AB1	CHQ	CH9.B
DK1	AB2	CHQ	CH9.B
DK1	CHQ	CHQ	CH9.B
DK1	HQ	CHQ	CH9.B
DK1	L3	CHQ	CH9.B
DK1	SW1	CHQ	CH9.B
DK1	SW2	CHQ	CH9.B
DK1	SW3	CHQ	CH9.B
HQ	AB1	SW2	CH7.A
HQ	AB2	SW2	CH7.A
HQ	CHQ	SW2	CH7.A
HQ	DK1	SW2	CH7.A
HQ	L3	L3	CH6.A
HQ	SW1	SW2	CH7.A
HQ	SW2	SW2	CH7.A
HQ	SW3	SW2	CH7.A
L3	AB1	HQ	CH6.B
L3	AB2	HQ	CH6.B
L3	CHQ	HQ	CH6.B
L3	DK1	HQ	CH6.B
L3	HQ	HQ	CH6.B
L3	SW1	HQ	CH6.B
L3	SW2	HQ	CH6.B
L3	SW3	HQ	CH6.B
SW1	AB1	AB1	CH1.B
SW1	AB2	AB2	CH2.B
SW1	CHQ	SW3	CH4.A
SW1	DK1	SW3	CH4.A
SW1	HQ	SW2	CH3.A
SW1	L3	SW2	CH3.A
SW1	SW2	SW2	CH3.A
SW1	SW3	SW3	CH4.A
SW2	AB1	SW1	CH3.B
SW2	AB2	SW1	CH3.B
SW2	CHQ	SW3	CH5.A
SW2	DK1	SW3	CH5.A
SW2	HQ	HQ	CH7.B
SW2	L3	HQ	CH7.B
SW2	SW1	SW1	CH3.B
SW2	SW3	SW3	CH5.A
SW3	AB1	SW1	CH4.B
SW3	AB2	SW1	CH4.B
SW3	CHQ	CHQ	CH8.B
SW3	DK1	CHQ	CH8.B
SW3	HQ	SW2	CH5.B
SW3	L3	SW2	CH5.B
SW3	SW1	SW1	CH4.B

```
ACTION DEFINITION.....

ACTION   ACTION
MNEMONIC CLASS    COMMENT
======== ======== ====================================================
CHQGD.OH MACHINE  CHQ PROCESSING OF GRAPHICS REQUEST
CHQHD.OH MACHINE  CHQ PROCESSING OF HARD COPY REQUEST
CS.OH    CPU      PROCESSING TO PERFORM CONTEXT SWITCHING
DUMMYACT MACHINE  ACTION TO ENABLE  CYCLIC PROGRAM CYCLES
FORMAT   MACHINE  TIME USED TO FORMAT PLANS FROM CHQ
HQ.OH    MACHINE  HQ PROCESSING OF MESSAGE
LATENCY  MACHINE  LATENCY PAUSE SUBSEQUENT TO SEEK
OVERHEAD MACHINE  TIME FOR GENERAL USE
ROUTE.OH CPU      PROCESSING DELAY TO ROUTE A MESSAGE
SEEK     MACHINE  SEEKING INFORMATION ON DISK
UPDATE   AIRBASE  UPDATING INFO SINCE PREVIOUS BROADCAST TO OTHER NODES
XFER     MACHINE  TRANSFER INFORMATION SOUGHT ON DISK
XFER.OH  CHANNEL  PROCESSING DELAY TO ROUTE A MESSAGE OVER A CHANNEL

PROCESS DEFINITION.....

PROCESS
MNEMONIC          DESCRIPTION
======== ======== ====================================================
AB-DATA           AIR BASE STATUS BROADCAST TO ALL OTHER NODES


ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
======== ======== ======== ======== ======== ==========================
         START             NO
         GIVEN    MSG
         RETURN   MSG
         CALL     REQ-I/O  NOWAIT   10       PROCESS REQUEST TO CHQ
         GIVEN    CHQ-DATA 10       $NOWAIT
                  750      CHQ
         CALL     REQ-I/O  NOWAIT   10       PROCESS REQUEST TO HQ
         GIVEN    HQ-DATA  10       $NOWAIT
                  750      HQ
         ASSIGN   $CNODE                     CURRENT NODE
                  CNODE
         COMPARE  CNODE             EQ       TEST FOR CURRENT NODE
                  AB1               AB1
         CALL     REQ-I/O  NOWAIT   10       PROCESS REQUEST TO AB1
         GIVEN    ABUPDATE 10       $NOWAIT
                  750      AB1
         BRANCH   END      100               BRANCH TO THE END
AB1      ENTRY                               ENTRY FROM COMPARE NODE
```

```
PAGE     9
         CALL     REQ-I/O  NOWAIT   10       PROCESS REQUEST TO AB2
         GIVEN    ABUPDATE 10       $NOWAIT
                  750      AB2
END      ENTRY                               ENTRY FROM REQUEST TO AB1
         END

LOCAL VARIABLES OF PROCESS AB-DATA
=======================================================================
    1 MSG       (I)    2 REQ-I/O   (P)    3 CHQ-DATA  (P)    4 CHQ       (R)
    5 HQ-DATA   (P)    6 HQ        (R)    7 CNODE            8 AB1       (R)
    9 ABUPDATE  (P)   10 AB2       (R)
PROCESS
MNEMONIC          DESCRIPTION
======== ======== ======================================================
AB-REQ            AIRBASE REQUEST FOR PLANS REPORT FROM CHQ


ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
======== ======== ======== ======== ======== =========================
         START             NO
         GIVEN    MSG
         RETURN   MSG
         CALL     REQ-I/O  WAIT     5        PROCESS REQUEST TO CHQ
         GIVEN    PLANS    5        $WAIT
                  200      CHQ
         END

LOCAL VARIABLES OF PROCESS AB-REQ
=======================================================================
    1 MSG       (I)    2 REQ-I/O   (P)    3 PLANS     (P)    4 CHQ       (R)

PROCESS
MNEMONIC          DESCRIPTION
======== ======== ======================================================
ABUPDATE          UPDATE DATA FROM AIRBASE


ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
======== ======== ======== ======== ======== =========================
         START             NO
         GIVEN    MSG
         RETURN   MSG
         UPDATE   CONSTANT 0.1               TIME CONSUMED IN UPDATING
         END

LOCAL VARIABLES OF PROCESS ABUPDATE
=======================================================================
    1 MSG       (I)    2 UPDATE    (A)
PROCESS
```

```
MNEMONIC          DESCRIPTION
======== ======== ==================================================
CHLIO             FULL AND HALF DUPLEX CHANNEL LOGIC

ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
======== ======== ======== ======== ======== =========================
         START    ALL      NO
         GIVEN    MSG
         ASSIGN   MSG      CNODE             SET INTERNAL NODE CURRENT
                  $CNODE
         ASSIGN   MSG      TNODE             GET DESTINATION NODE (MSG)
                  TO.NODE
         ASSIGN   $NXTNODE TO.NODE           SET NEXT NODE TO DEST N
                  NXT.NODE
         ASSIGN   $CHANNEL TO.NODE           GET CHANNEL TO NEXT NODE
                  CHANNEL
         ALLOC    CHANNEL                    OBTAIN CHANNEL FOR X  FER
         ASSIGN   CHANNEL  RATE              WHAT IS RATE IN MSEC/BYTE?
                  VSPEED
         ASSIGN   MSG      LENGTH            MESSAGE LENGTH IN BYTES
                  VLENGTH
         EVAL     VM.OVHD  MULTIPLY          CALCULATE TRANSFER TIME
                  VSPEED   VLENGTH
         XFER.OH  CONSTANT VM.OVHD           DELAY DUE TO TRANSFER TIME
         ASSIGN   NXT.NODE                   MESSAGE RESIDES IN NEXT
                  MSG      CNODE
         ASSIGN   NXT.NODE                   SET INTERNAL NODE REGISTER
                  $CNODE
         DEALLOC  CHANNEL                    FREE UP CHANNEL AFTER XFER
         CALL     IHANDLER NOWAIT   0        INDICATE INTERUPT IN NEXT
         GIVEN    MSG
         END

LOCAL VARIABLES OF PROCESS CHLIO
=======================================================================
    1 MSG      (I)   2 TO.NODE        3 NXT.NODE       4 CHANNEL
    5 VSPEED         6 VLENGTH        7 VM.OVHD        8 XFER.OH    (A)
    9 IHANDLER (P)
PROCESS
MNEMONIC          DESCRIPTION
======== ======== ==================================================
CHQ-DATA          CHQ GETS MESSAGE, FORMULATES RESPONSE, AND REPLIES

ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
======== ======== ======== ======== ======== =========================
         START             NO
         GIVEN    MSG
```

```
         RETURN   MSG
         ASSIGN   MSG      LENGTH            MAKE MSG-LENGTH = V.LENGTH
                  V.LENGTH
         EVAL     V.TIME   MULTIPLY          EVALUATE MSG PROCESS TIME
                  .015     V.LENGTH
         UPDATE   CONSTANT V.TIME            PROCESSING TIME CONSUMED
         END

LOCAL VARIABLES OF PROCESS CHQ-DATA
=====================================================================
    1 MSG       (I)    2 V.LENGTH         3 V.TIME          4 UPDATE     (A)

PROCESS
MNEMONIC          DESCRIPTION
======== ======== ==================================================
CONTROL           OPERATING SYSTEM : CONTEXT SWITCHING


ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
======== ======== ======== ======== ======== ========================
         START    ALL      NO
         GIVEN    MSG
         ASSIGN   MSG      CNODE             CURRENT NODE IS CPU
                  CP
         ALLOC    CP                         SIGNAL CURRENT CPU BUSY
         ASSIGN   CP       OSOVHD            MEAN CONTEXT SWITCH TIME
                  M.OVHD
         CS.OH    CONSTANT M.OVHD            DELAY CONTEXT SWITCH TIME
         COMPARE  MSG      TYPE     EQ       IF RESPONSE- RESUME PARENT
                  $REQ              REQUEST
         ASSIGN   MSG      PTASK             TASK TO RESUME IS IN MSG
                  TASK
         RESUME   TASK              QUEUE UP TASK FOR NODE
         BRANCH   DESTROY  100               END MESSAGE LIFE
REQUEST  ENTRY                               ELSE-> CALL REQUESTED PROC
         ASSIGN   MSG      RTASK             EXECUTE THE CALLED PROCESS
                  PROCESS
         CALL     PROCESS  WAIT     0        WAIT UNTIL COMPLETE
         GIVEN    MSG
         RETURN   MSG
         COMPARE  MSG      RESPONSE EQ       IF WAIT -> SEND MSG BACK
                  $NOWAIT           DESTROY
         ASSIGN   $RESP                      CHANGE MSG RESPONSE TYPE
                  MSG      TYPE
         ASSIGN   MSG      FNODE             SWITCH FROM AND TO NODES
                  MSG      TNODE
         ASSIGN   MSG      CNODE             CURRENT NODE IS FROM NODE
                  MSG      FNODE
         CALL     CHLIO    WAIT     0        RETURN MESSAGE TO ORIGIN
```

```
         GIVEN    MSG
         BRANCH   END      100
DESTROY  ENTRY                               TERMINATE MESSAGE AT DEST.
         DESTROY  MSG                        NO RESPONSE-TERMINATE MSG
END      ENTRY
         DEALLOC  CP                         INDICATE CP SWITCH DONE
         END

LOCAL VARIABLES OF PROCESS CONTROL
======================================================================
    1 MSG      (I)    2 CP              3 M.OVHD          4 CS.OH     (A)
    5 TASK            6 PROCESS   (X)   7 CHLIO     (P)
PROCESS
MNEMONIC          DESCRIPTION
======== ======== ====================================================
DISK.OP           OPERATION OF DISK

ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
======== ======== ======== ======== ======== =======================
         START             NO
         GIVEN    LENGTH   DISK
         ASSIGN   DISK     SPEED             MAKE DISK SPEED = V.SPEED
                  V.SPEED
         EVAL     XFERTIME DIVIDE            TRANSFER TIME CALCULATED
                  LENGTH   V.SPEED
         ALLOC    DISK                       DISK ALLOCATED
         ASSIGN   DISK     SEEK              MAKE SEEKTIME = SEEK
                  SEEKTIME
         SEEK     UNIFORM  SEEKTIME SEEKTIME TIME FOR SEEK IS CONSUMED
         ASSIGN   DISK     LATENCY           MAKE DISKLATENCY=LATETIME
                  LATETIME
         LATENCY  UNIFORM  LATETIME LATETIME TIME CONSUMED FOR LATENCY
         XFER     CONSTANT XFERTIME          TRANSFER TIME CONSUMED
         DEALLOC  DISK                       DISK RESOURCE DEALLOCATED
         END

LOCAL VARIABLES OF PROCESS DISK.OP
======================================================================
    1 LENGTH          2 DISK            3 V.SPEED         4 XFERTIME
    5 SEEKTIME        6 SEEK      (A)   7 LATETIME        8 LATENCY   (A)
    9 XFER     (A)
PROCESS
MNEMONIC          DESCRIPTION
======== ======== ====================================================
ESR-CALL          OPERATING SYSTEM: EXECUTIVE SERVICE REQUEST (CALL )

ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
```

```
======== ======== ======== ======== ======== =======================
         START    ALL      NO
         GIVEN    MSG
         ASSIGN   $TASK                      $TASK= INSTANCE TO RESUME
                  MSG      PTASK
         ASSIGN   MSG      RESPONSE          OPTION= $WAIT OF $NOWAIT
                  RESP.OPT
         CALL     ROUTER   WAIT     0        INITIATE ROUTING TO DEST.
         GIVEN    MSG
         COMPARE  RESP.OPT          EQ       SHOULD PARENT SUSPEND ?
                  $NOWAIT           END
         SUSPEND                             PROCESS CALLED WAIT
END      ENTRY                               CONTINUE OR RESUME POINT
         END

LOCAL VARIABLES OF PROCESS ESR-CALL
====================================================================
    1 MSG       (I)    2 RESP.OPT         3 ROUTER    (P)
PROCESS
MNEMONIC          DESCRIPTION
======== ======== ==================================================
HQ-DATA           HQ GETS MESSAGE, FORMULATES RESPONSE, AND REPLIES

ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
======== ======== ======== ======== ======== =======================
         START             NO
         GIVEN    MSG
         RETURN   MSG
         ASSIGN   MSG      LENGTH            MAKE MSG-LENTH = V.LENGTH
                  V.LENGTH
         EVAL     V.TIME   MULTIPLY          EVALUATE MSG PROCESS TIME
                  .015     V.LENGTH
         UPDATE   CONSTANT V.TIME            PROCESSING TIME CONSUMED
         END

LOCAL VARIABLES OF PROCESS HQ-DATA
====================================================================
    1 MSG       (I)    2 V.LENGTH         3 V.TIME          4 UPDATE     (A)

PROCESS
MNEMCNIC          DESCRIPTION
======== ======== ==================================================
HQ-REQ            HQ REQUEST FOR STATUS DISPLAY FROM CHQ

ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
======== ======== ======== ======== ======== =======================
         START    L3       NO
```

```
        GIVEN    MSG
        RETURN   MSG
        CALL     REQ-I/O  WAIT     4        MAKES I/O REQUEST TO CHQ
        GIVEN    PLANS    4        $WAIT
                 200      CHQ
        END

LOCAL VARIABLES OF PROCESS HQ-REQ
======================================================================
    1 MSG       (I)    2 REQ-I/O   (P)    3 PLANS     (P)    4 CHQ       (R)

PROCESS
MNEMONIC         DESCRIPTION
======== ======== ==================================================
IHANDLER         OPERATING SYSTEM : INTERUPT HANDLING AND ROUTING


ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
======== ======== ======== ======== ======== =========================
         START    ALL      NO
         GIVEN    MSG
         ASSIGN   MSG      CNODE             INDICATE CURRENT NODE CPU
                  CP
         COMPARE  MSG      CNODE    EQ       IS MSG AT DESTINATION ?
                  MSG      TNODE    CONTROL
         ASSIGN   CP       NETINSTR          MONITOR OVERHEAD FOR PLOT
                  M.OVHD
         ALLOC    CP                         OBTAIN CP-HANDLE INTERUPT
         ROUTE.OH CONSTANT M.OVHD            DELAY FOR ROUTING
         DEALLOC  CP                         RELEASE CPU TO OTHERS
         CALL     CHLIO    NOWAIT   0        FORWARD MESSAGE WITH I/O
         GIVEN    MSG
         BRANCH   END      100
CONTROL  ENTRY                               MESSAGE AT DESTINATION
         COMPARE  MSG      TYPE     EQ       IF RESPONSE-UP PRIORITY
                  $RESP             HPCONTRL
         ASSIGN   MSG      TASKPRI           SET MESSAGE PRIORITY
                  PRIORITY
HPCONTRL ENTRY                               PRIORITY=0 IF UNDEFINED
         CALL     CONTROL  NOWAIT   PRIORITY CONTEXT SWITCH MESSAGE
         GIVEN    MSG
END      ENTRY
         END

LOCAL VARIABLES OF PROCESS IHANDLER
======================================================================
    1 MSG       (I)    2 CP                3 M.OVHD              4 ROUTE.OH  (A)
    5 CHLIO     (P)    6 PRIORITY          7 CONTROL   (P)
PROCESS
```

```
PAGF   15
MNEMONIC          DESCRIPTION
======== ======== ==================================================
PLANS             REQUEST FOR PLANS FROM CHQ


ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
======== ======== ======== ======== ======== ======================
         START    CHQ      NO
         GIVEN    MSG
         RETURN   MSG
         ASSIGN   MSG      LENGTH            MAKE MSG LENGTH = V.LENGTH
                  V.LENGTH
         EVAL     V.TIME   MULTIPLY          EVALUATE MSG PROCESS TIME
                  .01      V.LENGTH
         FORMAT   CONSTANT V.TIME            TIME USED TO FORMAT PLANS
         CALL     DISK.OP  WAIT     10       CALLING PROCESS DISK.OP
         GIVEN    10000    DK1
         ASSIGN   10000                      INCREASE MSG LENGTH
                  MSG      LENGTH
         END

LOCAL VARIABLES OF PROCESS PLANS
=================================================================
    1 MSG       (I)    2 V.LENGTH        3 V.TIME          4 FORMAT    (A)
    5 DISK.OP   (P)    6 DK1       (R)
PROCESS
MNEMONIC          DESCRIPTION
======== ======== ==================================================
REQ-I/O           GENERATE A PROCESS REQUEST MESSAGE AND INITIATE I/O


ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
======== ======== ======== ======== ======== ======================
         START    ALL      NO
         GIVEN    PROCESS  PRIORITY RESP.OPT
                  MSG.LNTH TO.NODE
         CREATE   MSG                        CREATE MESSAGE DATA TO RTE
         ASSIGN   $CNODE                     INDICATE CURRENT NODE
                  MSG      CNODE
         ASSIGN   $CNODE                     INDICATE CURRENT NODE FROM
                  MSG      FNODE
         ASSIGN   PROCESS                    INDICATE REQUESTED PROCESS
                  MSG      RTASK
         ASSIGN   PRIORITY                   INDICATE RELATIVE PRIORITY
                  MSG      TASKPRI
         ASSIGN   RESP.OPT                   $NOWAIT OR $ WAIT ON CALL
                  MSG      RESPONSE
         ASSIGN   MSG.LNTH                   INDICATE LENGTH IN BYTES
                  MSG      LENGTH
```

```
         COMPARE  TO.NODE           EQ       WHERE DOES PROCESS RESIDE
                  $NO               END
         COMPARE  TO.NODE           EQ       DEFAULT TO NODE SELECT
                  $YES              GETNODE
         ASSIGN   TO.NODE                    ELSE-> NODE IS GIVEN
                  MSG      TNODE
         BRANCH   END      100
GETNODE  ENTRY                               DETERMINE NODE FROM PROC.
         ASSIGN   $NODE    PROCESS           $NODE OF PROCESS IN DEF
                  MSG      TNODE
END      ENTRY                               SEND MSG FOR SERVICE
         CALL     ESR-CALL WAIT     0        EXECUTIVE SERVICING OF MSG
         GIVEN    MSG
         END

LOCAL VARIABLES OF PROCESS REQ-I/O
=====================================================================
    1 PROCESS   (X)    2 PRIORITY          3 RESP.OPT          4 MSG.LNTH
    5 TO.NODE          6 MSG       (I)     7 ESR-CALL  (P)
PROCESS
MNEMONIC          DESCRIPTION
======== ======== ==================================================
ROUTER            OPERATING SYSTEM : INTERUPT HANDLING AND ROUTING


ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
======== ======== ======== ======== ======== =========================
         START    ALL      NO
         GIVEN    MSG
         ASSIGN   MSG      CNODE             INDICATE CURRENT NODE CPU
                  CP
         COMPARE  MSG      CNODE    EQ       IS MSG AT DESTINATION ?
                  MSG      TNODE    CONTROL
         ASSIGN   CP       NETINSTR          MONITOR OVERHEAD FOR PLOT
                  M.OVHD
         ROUTE.OH CONSTANT M.OVHD            DELAY FOR ROUTING
         CALL     CHLIO    NOWAIT   0        FORWARD MESSAGE WITH I/O
         GIVEN    MSG
         BRANCH   END      100
CONTROL  ENTRY                               MESSAGE AT DESTINATION
         COMPARE  MSG      TYPE     EQ       IF RESPONSE-UP PRIORITY
                  $RESP             HPCONTRL
         ASSIGN   MSG      TASKPRI           SET MESSAGE PRIORITY
                  PRIORITY
HPCONTRL ENTRY                               PRIORITY=0 IF UNDEFINED
         CALL     CONTROL  NOWAIT   PRIORITY CONTEXT SWITCH MESSAGE
         GIVEN    MSG
END      ENTRY
         END
```

```
LOCAL VARIABLES OF PROCESS ROUTER
=====================================================================
    1 MSG        (I)   2 CP               3 M.OVHD           4 ROUTE.OH  (A)
    5 CHLIO      (P)   6 PRIORITY         7 CONTROL    (P)
PROCESS
MNEMONIC          DESCRIPTION
======== ======== ====================================================
TRACE             TURN ON TRACE OUTPUT


ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
======== ======== ======== ======== ======== =========================
         START    ALL      NO
         COMPARE  V.TRACE           EQ       TEST IF FLAG SET FOR TRACE
                  0                 NOTRACE
         TRACE    ON
NOTRACE  ENTRY
         END

LOAD DEFINITION.....

LOAD
MNEMONIC          DESCRIPTION
======== ======== ====================================================
ABLOAD            COMMUNICATIONS FROM AIRBASES
         LOAD     NODES
         ======== ======== ======== ========
         AB1      AB2


         PROCESS           SCHEDULE
         MNEMONIC MAX #    METHOD   MEAN     DELTA    PRIORITY
         ======== ======== ======== ======== ======== ========
         AB-DATA  60       INTERVAL ABDRATE           10
         AB-REQ   60       EXPONENT ABRRATE           5
LOAD
MNEMONIC          DESCRIPTION
======== ======== ====================================================
HQLOAD            REQUEST DATA FROM CHQ
         LOAD     NODES
         ======== ======== ======== ========
         L3


         PROCESS           SCHEDULE
         MNEMONIC MAX #    METHOD   MEAN     DELTA    PRIORITY
         ======== ======== ======== ======== ======== ========
         HQ-REQ   60       EXPONENT HQRRATE           4
```

```
SCENARIO DEFINITION....

SCENARIO
MNEMONIC          DESCRIPTION
======== ======== ================================================
TEST01            SCENARIO FOR MINI MITRE 1

PERIOD
LENGTH
========
360000

         PERIOD   PERIOD   PERIOD   PERIOD   PERIOD   PERIOD   PERIOD
         MNEMONIC MNEMONIC MNEMONIC MNEMONIC MNEMONIC MNEMONIC MNEMONIC
======== ======== ======== ======== ======== ======== ======== ========
         1        2        3        4        5        6        7
         PERIOD   PERIOD   PERIOD   PERIOD   PERIOD   PERIOD   PERIOD
         MNEMONIC MNEMONIC MNEMONIC MNEMONIC MNEMONIC MNEMONIC MNEMONIC
======== ======== ======== ======== ======== ======== ======== ========
         8        9        10

TRIGGER  TIME  TO SCHEDULE TRIGGER  TIME  TO SCHEDULE
MNEMONIC SCHEDULE PRIORITY MNEMONIC SCHEDULE PRIORITY
======== ======== ======== ======== ======== ========
ABLOAD   0        0        HQLOAD   0        0
TRACE    0        0

####    0 ERRORS WERE DETECTED DURING MODEL INITIALIZATION
```

```
PAGE  19
SIMULATION TIME =    3600000.00000 UNITS
CONSTANT REPORT

         CURRENT
CONSTANT VALUE...
======== ==========
V.TRACE       0.
```

PAGE 20

SIMULATION TIME = 3600000.00000 UNITS

VARIABLE REPORT

NUMERIC VARIABLES...

	TOTAL	VALUE				
VARIABLE	SAMPLES.	CURRENT...	MEAN......	STD DEV...	MINIMUM...	MAXIMUM...
ACDRATE	1	60000.000	60000.000	0.	60000.000	60000.000
ABRRATE	1	36000.000	36000.000	0.	36000.000	36000.000
HORRATE	1	72000.000	72000.000	0.	72000.000	72000.000
TIME1	1	30.000	30.000	0.	30.000	30.000
VRATE	1	1.628	1.628	0.	1.628	1.628

NON-NUMERIC VARIABLES...

VARIABLE	CURRENT TYPE	CURRENT VALUE

PAGE 21

SIMULATION TIME = 3600000.00000 UNITS

ITEM REPORT

ITEM NAME	NUMBER CPEATED	NUMBER DESTR'D	TIME IN SYSTEM MINIMUM...	MAXIMUM...	AVERAGE...	STD DEV...
MSG	522	521	2997.82	2426851.67	322496.42	538431.97

SIMULATION TIME = 3600000.00000 UNITS

RESOURCE REPORT

RESOURCE	TOTAL NUMBER	CURRENT...	MEAN......	STD DEV...	MINIMUM...	MAXIMUM...
AB1						
INTO IDLE	180					
OUT OF IDLE	179					
# IDLE		1.000	1.000	.001	0.	1.000
IDLE TIME			19805.514	21506.933	.002	72193.117
INTO BUSY	179					
OUT OF BUSY	179					
# BUSY		0.	.000	.001	0.	1.000
BUSY TIME			.413	.369	.002	1.080
INTO INACT.	0					
OUT OF INACT.	0					
# INACTIVE		0.	0.	0.	0.	0.
INACTIVE TIME			0.	0.	0.	0.
INTO WAIT	179					
OUT OF WAIT	179					
# WAITING		0.	0.	0.	0.	1.000
WAIT TIME			.380	.355	.002	.996

CURRENTLY ALLOCATED
TO PROCESSES: NONE

PROCESSES CURRENTLY
WAITING: NONE

RESOURCE	TOTAL NUMBER	CURRENT...	MEAN......	STD DEV...	MINIMUM...	MAXIMUM...
AB2						
INTO IDLE	180					
OUT OF IDLE	179					
# IDLE		1.000	1.000	.001	0.	1.000
IDLE TIME			19805.101	20510.246	.002	64024.710
INTO BUSY	179					
OUT OF BUSY	179					
# BUSY		0.	.000	.001	0.	1.000
BUSY TIME			.440	.368	.002	.992

```
PAGE   23
    INTO INACT.        0
  OUT OF INACT.        0
     # INACTIVE                0.         0.         0.         0.         0.
  INACTIVE TIME                           0.         0.         0.         0.

      INTO WAIT      179
    OUT OF WAIT      179
      # WAITING                0.         0.         0.         0.      1.000
      WAIT TIME                         .407       .350       .000       .992

     CURRENTLY ALLOCATED
            TO PROCESSES: NONE

     PROCESSES CURRENTLY
                  WAITING: NONE

                   TOTAL
RESOURCE           NUMBER   CURRENT... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
CHQ
      INTO IDLE      287
    OUT OF IDLE      286
         # IDLE                1.000       .238       .426       0.       1.000
      IDLE TIME                       2913.735  10424.679       .000  58768.574

      INTO BUSY      286
    OUT CF BUSY      286
         # BUSY                0.          .762       .426       0.       1.000
      BUSY TIME                       9593.122   8029.981     11.350  16365.016

    INTO INACT.        0
  OUT OF INACT.        0
     # INACTIVE                0.         0.         0.         0.         0.
  INACTIVE TIME                           0.         0.         0.         0.

      INTO WAIT      286
    OUT OF WAIT      286
      # WAITING                0.       44.107     37.688       0.      96.000
      WAIT TIME                     555189.716 624558.143       .000  2.424E 06

     CURRENTLY ALLOCATED
            TO PROCESSES: NONE

     PROCESSES CURRENTLY
                  WAITING: NONE

                   TOTAL
RESOURCE           NUMBER   CURRENT... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
```

```
PAGE   24
CH1.A
      INTO IDLE       238
    OUT OF IDLE       237
         # IDLE                1.000       .935       .247       0.         1.000
      IDLE TIME                       13958.689  20910.584       .000  56338.000

      INTO BUSY       237
    OUT OF BUSY       237
         # BUSY                0.          .065       .247       0.         1.000
      BUSY TIME                         994.299    389.309    325.520   1221.399

    INTO INACT.         0
  OUT OF INACT.         0
     # INACTIVE                0.         0.         0.         0.         0.
  INACTIVE TIME                           0.         0.         0.         0.

      INTO WAIT       237
    OUT OF WAIT       237
      # WAITING                0.          .060       .311       0.         2.000
      WAIT TIME                         918.870   1009.749       0.      2441.399

    CURRENTLY ALLOCATED
           TO PROCESSES: NONE

    PROCESSES CURRENTLY
                WAITING: NONE

                 TOTAL
RESOURCE         NUMBER   CURRENT... MEAN...... STD DEV... MINIMUM... MAXIMUM...
===============  ======== ========== ========== ========== ========== ==========
CH1.B
      INTO IDLE       120
    OUT OF IDLE       119
         # IDLE                1.000       .709       .454       0.         1.000
      IDLE TIME                       20979.747  19678.971       .018  58780.099

      INTO BUSY       119
    OUT OF BUSY       119
         # BUSY                0.          .291       .454       0.         1.000
      BUSY TIME                        8811.838   7527.453   1220.701  16276.921

    INTO INACT.         0
  OUT OF INACT.         0
     # INACTIVE                0.         0.         0.         0.         0.
  INACTIVE TIME                           0.         0.         0.         0.

      INTO WAIT       119
    OUT OF WAIT       119
      # WAITING                0.          .043       .203       0.         1.000
```

WAIT TIME			1302.553	3549.593	.000	16116.024

CURRENTLY ALLOCATED
TO PROCESSES: NONE

PROCESSES CURRENTLY
WAITING: NONE

RESOURCE	TOTAL NUMBER	CURRENT...	MEAN......	STD DEV...	MINIMUM...	MAXIMUM...
CH2.A						
INTO IDLE	238					
OUT OF IDLE	237					
# IDLE		1.000	.935	.247	0.	1.000
IDLE TIME			13958.689	20910.584	.000	56338.000
INTO BUSY	237					
OUT OF BUSY	237					
# BUSY		0.	.065	.247	0.	1.000
BUSY TIME			994.299	389.309	325.520	1221.399
INTO INACT.	0					
OUT OF INACT.	0					
# INACTIVE		0.	0.	0.	0.	0.
INACTIVE TIME			0.	0.	0.	0.
INTO WAIT	237					
OUT OF WAIT	237					
# WAITING		0.	.060	.311	0.	2.000
WAIT TIME			918.870	1009.749	0.	2441.399

CURRENTLY ALLOCATED
TO PROCESSES: NONE

PROCESSES CURRENTLY
WAITING: NONE

RESOURCE	TOTAL NUMBER	CURRENT...	MEAN......	STD DEV...	MINIMUM...	MAXIMUM...
CH2.B						
INTO IDLE	120					
OUT OF IDLE	119					
# IDLE		1.000	.709	.454	0.	1.000
IDLE TIME			20979.111	18918.780	.022	58780.099
INTO BUSY	119					
OUT OF BUSY	119					

```
       # BUSY                 0.          .291        .454      0.         1.000
    BUSY TIME                         8011.824    7527.494   1220.701  16276.992

   INTO INACT.         0
 OUT OF INACT.         0
    # INACTIVE                0.          0.          0.        0.         0.
 INACTIVE TIME                            0.          0.        0.         0.

     INTO WAIT       119
   OUT OF WAIT       119
     # WAITING                0.          .034        .180      0.         1.000
     WAIT TIME                        1018.494    3265.229       .001  15220.232

   CURRENTLY ALLOCATED
          TO PROCESSES: NONE

   PROCESSES CURRENTLY
               WAITING: NONE

                TOTAL
RESOURCE        NUMBER   CURRENT... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
CH3.A
     INTO IDLE       119
   OUT OF IDLE       118
        # IDLE               1.000        .990        .100      0.         1.000
     IDLE TIME                       29722.518   29723.551      0.     62601.399

     INTO BUSY       118
   OUT OF BUSY       118
        # BUSY                0.          .010        .100      0.         1.000
     BUSY TIME                         305.526        .225    305.175    305.749

   INTO INACT.         0
 OUT OF INACT.         0
    # INACTIVE                0.          0.          0.        0.         0.
 INACTIVE TIME                            0.          0.        0.         0.

     INTO WAIT       118
   OUT OF WAIT       118
     # WAITING                0.          .004        .061      0.         1.000
     WAIT TIME                         112.878     112.588       .001    225.574

   CURRENTLY ALLOCATED
          TO PROCESSES: NONE

   PROCESSES CURRENTLY
               WAITING: NONE
```

RESOURCE	TOTAL NUMBER	CURRENT...	MEAN......	STD DEV...	MINIMUM...	MAXIMUM...
CH3.B						
INTO IDLE	1					
OUT OF IDLE	0					
# IDLE		1.000	1.000	0.	1.000	1.000
IDLE TIME			0.	0.	0.	0.
INTO BUSY	0					
OUT OF BUSY	0					
# BUSY		0.	0.	0.	0.	0.
BUSY TIME			0.	0.	0.	0.
INTO INACT.	0					
OUT OF INACT.	0					
# INACTIVE		0.	0.	0.	0.	0.
INACTIVE TIME			0.	0.	0.	0.
INTO WAIT	0					
OUT OF WAIT	0					
# WAITING		0.	0.	0.	0.	0.
WAIT TIME			0.	0.	0.	0.

CURRENTLY ALLOCATED
TO PROCESSES: NONE

PROCESSES CURRENTLY
WAITING: NONE

RESOURCE	TOTAL NUMBER	CURRENT...	MEAN......	STD DEV...	MINIMUM...	MAXIMUM...
CH4.A						
INTO IDLE	239					
OUT OF IDLE	238					
# IDLE		1.000	.987	.112	0.	1.000
IDLE TIME			14690.230	21806.612	.000	59389.700
INTO BUSY	238					
OUT OF BUSY	238					
# BUSY		0.	.013	.112	0.	1.000
BUSY TIME			192.706	112.089	81.380	306.049
INTO INACT.	0					
OUT OF INACT.	0					
# INACTIVE		0.	0.	0.	0.	0.
INACTIVE TIME			0.	0.	0.	0.

```
      INTO WAIT      238
    OUT OF WAIT      238
      # WAITING               0.          .004       .061      0.          1.000
      WAIT TIME                         56.506     97.122      0.        225.875

    CURRENTLY ALLOCATED
           TO PROCESSES: NONE

    PROCESSES CURRENTLY
                WAITING: NONE

                TOTAL
RESOURCE        NUMBER   CURRENT... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
CH4.B
      INTO IDLE      121
    OUT OF IDLE      120
         # IDLE               1.000       .864       .342      0.          1.000
      IDLE TIME                      16689.460  10712.746  12213.126  77623.700

      INTO BUSY      120
    OUT OF BUSY      120
         # BUSY               0.          .136       .342      0.          1.000
      BUSY TIME                       4069.309       .325   4069.000   4069.996

    INTO INACT.        0
  OUT OF INACT.        0
     # INACTIVE               0.         0.         0.         0.         0.
  INACTIVE TIME                          0.         0.         0.         0.

      INTO WAIT      120
    OUT OF WAIT      120
      # WAITING               0.         0.         0.         0.          1.000
      WAIT TIME                           .309       .325       .000       .996

    CURRENTLY ALLOCATED
           TO PROCESSES: NONE

    PROCESSES CURRENTLY
                WAITING: NONE

                TOTAL
RESOURCE        NUMBER   CURRENT... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
CH5.A
      INTO IDLE       49
    OUT OF IDLE       48
         # IDLE               1.000       .999       .033      0.          1.000
      IDLE TIME                      74085.978  77006.891    244.398 413891.878
```

```
      INTO BUSY       48
    OUT OF BUSY       48
         # BUSY                0.          .001       .033      0.         1.000
      BUSY TIME                          81.756       .339     81.384     82.379

    INTO INACT.        0
  OUT OF INACT.        0
     # INACTIVE                0.         0.         0.         0.         0.
  INACTIVE TIME                           0.         0.         0.         0.

      INTO WAIT       48
    OUT OF WAIT       48
      # WAITING                0.         0.         0.         0.         1.000
      WAIT TIME                            .376       .339       .004       .999

    CURRENTLY ALLOCATED
           TO PROCESSES: NONE

    PROCESSES CURRENTLY
                WAITING: NONE

                  TOTAL
RESOURCE          NUMBER   CURRENT... MEAN...... STD DEV... MINIMUM... MAXIMUM...
============== ======== ========== ========== ========== ========== ==========
CH5.B
      INTO IDLE       49
    OUT OF IDLE       49
         # IDLE                1.000       .946       .227      0.         1.000
      IDLE TIME                       70532.827  77227.154  12222.011 420416.946

      INTO BUSY       48
    OUT OF BUSY       48
         # BUSY                0.          .054       .227      0.         1.000
      BUSY TIME                        4069.368       .319   4069.003   4069.997

    INTO INACT.        0
  OUT OF INACT.        0
     # INACTIVE                0.         0.         0.         0.         0.
  INACTIVE TIME                           0.         0.         0.         0.

      INTO WAIT       48
    OUT OF WAIT       48
      # WAITING                0.         0.         0.         0.         1.000
      WAIT TIME                            .369       .319       .003       .997

    CURRENTLY ALLOCATED
           TO PROCESSES: NONE
```

PROCESSES CURRENTLY
WAITING: NONE

RESOURCE	TOTAL NUMBER	CURRENT...	MEAN......	STD DEV...	MINIMUM...	MAXIMUM...
CH6.A						
INTO IDLE	48					
OUT OF IDLE	48					
# IDLE		0.	.787	.410	0.	1.000
IDLE TIME			59007.342	77923.555	.185	408209.942
INTO BUSY	48					
OUT OF BUSY	47					
# BUSY		1.000	.213	.410	0.	1.000
BUSY TIME			16276.388	.322	16276.008	16276.989
INTO INACT.	0					
OUT OF INACT.	0					
# INACTIVE		0.	0.	0.	0.	0.
INACTIVE TIME			0.	0.	0.	0.
INTO WAIT	48					
OUT OF WAIT	48					
# WAITING		0.	0.	0.	0.	1.000
WAIT TIME			.383	.323	.003	.993

CURRENTLY ALLOCATED
TO PROCESSES: CHLIO

PROCESSES CURRENTLY
WAITING: NONE

RESOURCE	TOTAL NUMBER	CURRENT...	MEAN......	STD DEV...	MINIMUM...	MAXIMUM...
CH6.B						
INTO IDLE	49					
OUT OF IDLE	48					
# IDLE		1.000	.996	.066	0.	1.000
IDLE TIME			73830.020	76999.078	.358	413647.838
INTO BUSY	48					
OUT OF BUSY	48					
# BUSY		0.	.004	.066	0.	1.000
BUSY TIME			325.950	.351	325.520	326.516
INTO INACT.	0					
OUT OF INACT.	0					

```
      # INACTIVE                 0.         0.         0.         0.         0.
   INACTIVE TIME                            0.         0.         0.         0.

      INTO WAIT        48
    OUT OF WAIT        48
      # WAITING                  0.          .000       .007     0.         1.000
      WAIT TIME                             3.680     22.273     0.       156.358

    CURRENTLY ALLOCATED
           TO PROCESSES: NONE

    PROCESSES CURRENTLY
                WAITING: NONE

                TOTAL
RESOURCE        NUMBER   CURRENT... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
CH7.A
      INTO IDLE      49
    OUT OF IDLE      48
         # IDLE            1.000         .996       .066     0.         1.000
      IDLE TIME                     73838.502  77004.727       .878 413648.358

      INTO BUSY      48
    OUT OF BUSY      48
         # BUSY            0.            .004       .066     0.         1.000
      BUSY TIME                       325.872       .294   325.528    326.512

    INTO INACT.       0
  OUT OF INACT.       0
     # INACTIVE            0.          0.         0.         0.         0.
  INACTIVE TIME                        0.         0.         0.         0.

      INTO WAIT      48
    OUT OF WAIT      48
      # WAITING            0.          0.         0.         0.         1.000
      WAIT TIME                          .353       .294       .008       .992

    CURRENTLY ALLOCATED
           TO PROCESSES: NONE

    PROCESSES CURRENTLY
                WAITING: NONE

                TOTAL
RESOURCE        NUMBER   CURRENT... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
CH7.B
      INTO IDLE     167
```

OUT OF IDLE	166					
# IDLE		1.000	.743	.437	0.	1.000
IDLE TIME			16096.416	22225.381	.000	62986.574
INTO BUSY	166					
OUT OF BUSY	166					
# BUSY		0.	.257	.437	0.	1.000
BUSY TIME			5574.391	6825.653	1220.700	16276.993
INTO INACT.	0					
OUT OF INACT.	0					
# INACTIVE		0.	0.	0.	0.	0.
INACTIVE TIME			0.	0.	0.	0.
INTO WAIT	166					
OUT OF WAIT	166					
# WAITING		0.	.081	.367	0.	2.000
WAIT TIME			1760.093	3166.189	0.	13581.711

CURRENTLY ALLOCATED
TO PROCESSES: NONE

PROCESSES CURRENTLY
WAITING: NONE

RESOURCE	TOTAL NUMBER	CURRENT...	MEAN......	STD DEV...	MINIMUM...	MAXIMUM...
CH8.A						
INTO IDLE	169					
OUT OF IDLE	168					
# IDLE		1.000	.240	.427	0.	1.000
IDLE TIME			5014.336	22939.276	6.130	203945.042
INTO BUSY	168					
OUT OF BUSY	168					
# BUSY		0.	.760	.427	0.	1.000
BUSY TIME			16276.317	.323	16275.996	16276.996
INTO INACT.	0					
OUT OF INACT.	0					
# INACTIVE		0.	0.	0.	0.	0.
INACTIVE TIME			0.	0.	0.	0.
INTO WAIT	168					
OUT OF WAIT	168					
# WAITING		0.	0.	0.	0.	1.000
WAIT TIME			.321	.323	.000	1.000

```
    CURRENTLY ALLOCATED
           TO PROCESSES: NONE

    PROCESSES CURRENTLY
                WAITING: NONE
```

RESOURCE	TOTAL NUMBER	CURRENT...	MEAN......	STD DEV...	MINIMUM...	MAXIMUM...
CH8.B						
INTO IDLE	287					
OUT OF IDLE	286					
# IDLE		1.000	.945	.228	0.	1.000
IDLE TIME			11754.441	17075.324	0.	57558.875
INTO BUSY	286					
OUT OF BUSY	286					
# BUSY		0.	.055	.228	0.	1.000
BUSY TIME			695.206	440.844	325.520	1221.609
INTO INACT.	0					
OUT OF INACT.	0					
# INACTIVE		0.	0.	0.	0.	0.
INACTIVE TIME			0.	0.	0.	0.
INTO WAIT	286					
OUT OF WAIT	286					
# WAITING		0.	.020	.141	0.	2.000
WAIT TIME			250.543	365.644	.000	1455.211

```
    CURRENTLY ALLOCATED
           TO PROCESSES: NONE

    PROCESSES CURRENTLY
                WAITING: NONE
```

RESOURCE	TOTAL NUMBER	CURRENT...	MEAN......	STD DEV...	MINIMUM...	MAXIMUM...
CH9.A						
INTO IDLE	1					
OUT OF IDLE	0					
# IDLE		1.000	1.000	0.	1.000	1.000
IDLE TIME			0.	0.	0.	0.
INTO BUSY	0					
OUT OF BUSY	0					
# BUSY		0.	0.	0.	0.	0.
BUSY TIME			0.	0.	0.	0.

```
  INTO INACT.        0
 OUT OF INACT.       0
    # INACTIVE            0.         0.         0.         0.         0.
 INACTIVE TIME                       0.         0.         0.         0.

    INTO WAIT        0
  OUT OF WAIT        0
    # WAITING             0.         0.         0.         0.         0.
    WAIT TIME                        0.         0.         0.         0.

  CURRENTLY ALLOCATED
        TO PROCESSES: NONE

  PROCESSES CURRENTLY
             WAITING: NONE

                TOTAL
RESOURCE        NUMBER   CURRENT... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
CH9.B
    INTO IDLE        1
  OUT OF IDLE        0
       # IDLE             1.000      1.000      0.         1.000      1.000
    IDLE TIME                        0.         0.         0.         0.

    INTO BUSY        0
  OUT OF BUSY        0
       # BUSY             0.         0.         0.         0.         0.
    BUSY TIME                        0.         0.         0.         0.

  INTO INACT.        0
 OUT OF INACT.       0
    # INACTIVE            0.         0.         0.         0.         0.
 INACTIVE TIME                       0.         0.         0.         0.

    INTO WAIT        0
  OUT OF WAIT        0
    # WAITING             0.         0.         0.         0.         0.
    WAIT TIME                        0.         0.         0.         0.

  CURRENTLY ALLOCATED
        TO PROCESSES: NONE

  PROCESSES CURRENTLY
             WAITING: NONE

                TOTAL
RESOURCE        NUMBER   CURRENT... MEAN...... STD DEV... MINIMUM... MAXIMUM...
```

```
============== ======= ========= ========= ========= ========= =========
DK1
     INTO IDLE      169
   OUT OF IDLE      168
        # IDLE               1.000      .995      .046      0.        1.000
     IDLE TIME                      21148.898 22960.015  3799.320 220172.951

     INTO BUSY      168
   OUT OF BUSY      168
        # BUSY               0.         .002      .046      0.        1.000
     BUSY TIME                         44.895    18.992     4.130    87.020

   INTO INACT.        0
 OUT OF INACT.        0
    # INACTIVE               0.        0.        0.        0.        0.
 INACTIVE TIME                         0.        0.        0.        0.

     INTO WAIT      168
   OUT OF WAIT      168
     # WAITING               0.        0.        0.        0.        1.000
     WAIT TIME                          .347      .337      .000      .998

   CURRENTLY ALLOCATED
         TO PROCESSES: NONE

   PROCESSES CURRENTLY
              WAITING: NONE

                TOTAL
RESOURCE        NUMBER   CURRENT... MEAN...... STD DEV... MINIMUM... MAXIMUM...
============== ======= ========= ========= ========= ========= =========
HQ
     INTO IDLE      215
   OUT OF IDLE      214
        # IDLE               1.000      .997      .050      0.        1.000
     IDLE TIME                      16768.088 18958.141   245.878  64207.274

     INTO BUSY      214
   OUT OF BUSY      214
        # BUSY               0.         .003      .050      0.        1.000
     BUSY TIME                         42.495    34.162    11.250    80.993

   INTO INACT.        0
 OUT OF INACT.        0
    # INACTIVE               0.        0.        0.        0.        0.
 INACTIVE TIME                         0.        0.        0.        0.

     INTO WAIT      214
   OUT OF WAIT      214
```

```
     # WAITING                0.          0.          0.          0.          1.000
     WAIT TIME                             .404        .349        .000         .993

   CURRENTLY ALLOCATED
          TO PROCESSES: NONE

   PROCESSES CURRENTLY
               WAITING: NONE

                  TOTAL
RESOURCE          NUMBER    CURRENT... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
L3
     INTO IDLE        95
   OUT OF IDLE        94
        # IDLE                1.000       1.000       0.          0.          1.000
     IDLE TIME                         37863.632  67301.862       .008 424485.938

     INTO BUSY        94
   OUT OF BUSY        94
        # BUSY                0.          0.          0.          0.          1.000
     BUSY TIME                             .388        .322        .008         .989

   INTO INACT.         0
 OUT OF INACT.         0
    # INACTIVE                0.          0.          0.          0.          0.
 INACTIVE TIME                            0.          0.          0.          0.

     INTO WAIT        94
   OUT OF WAIT        94
     # WAITING                0.          0.          0.          0.          1.000
     WAIT TIME                             .388        .322        .008         .989

   CURRENTLY ALLOCATED
          TO PROCESSES: NONE

   PROCESSES CURRENTLY
               WAITING: NONE

                  TOTAL
RESOURCE          NUMBER    CURRENT... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
SW1
     INTO IDLE       595
   OUT OF IDLE       594
        # IDLE                1.000         .987        .114      0.          1.000
     IDLE TIME                          5886.436  11443.351      0.      57398.700

     INTO BUSY       594
```

```
   OUT OF BUSY       594
        # BUSY                  0.          .013       .114     0.          1.000
     BUSY TIME                            80.271       .283    80.000      80.996

   INTO INACT.         0
  OUT OF INACT.        0
     # INACTIVE                 0.         0.         0.        0.          0.
  INACTIVE TIME                            0.         0.        0.          0.

     INTO WAIT       594
   OUT OF WAIT       594
     # WAITING                  0.          .005       .073     0.          2.000
     WAIT TIME                            32.606     39.529     0.        149.028

   CURRENTLY ALLOCATED
          TO PROCESSES: NONE

   PROCESSES CURRENTLY
                WAITING: NONE

                 TOTAL
RESOURCE        NUMBER   CURRENT... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
SW2
     INTO IDLE       215
   OUT OF IDLE       214
        # IDLE                1.000         .995       .069     0.          1.000
     IDLE TIME                         16653.843  18829.567      .028   62906.574

     INTO BUSY       214
   OUT OF BUSY       214
        # BUSY                  0.          .005       .069     0.          1.000
     BUSY TIME                            80.432       .315    80.000      80.999

   INTO INACT.         0
  OUT OF INACT.        0
    # INACTIVE                  0.         0.         0.        0.          0.
  INACTIVE TIME                            0.         0.        0.          0.

     INTO WAIT       214
   OUT OF WAIT       214
     # WAITING                  0.          .000       .004     0.          1.000
     WAIT TIME                              .707      3.991     0.         58.778

   CURRENTLY ALLOCATED
          TO PROCESSES: NONE

   PROCESSES CURRENTLY
                WAITING: NONE
```

RESOURCE	TOTAL NUMBER	CURRENT...	MEAN......	STD DEV...	MINIMUM...	MAXIMUM...
SW3						
INTO IDLE	455					
OUT OF IDLE	454					
# IDLE		1.000	.990	.100	0.	1.000
IDLE TIME			7798.707	9472.552	.033	59614.875
INTO BUSY	454					
OUT OF BUSY	454					
# BUSY		0.	.010	.100	0.	1.000
BUSY TIME			80.291	.337	80.000	80.999
INTO INACT.	0					
OUT OF INACT.	0					
# INACTIVE		0.	0.	0.	0.	0.
INACTIVE TIME			0.	0.	0.	0.
INTO WAIT	454					
OUT OF WAIT	454					
# WAITING		0.	.000	.005	0.	1.000
WAIT TIME			.526	3.819	.000	74.658

CURRENTLY ALLOCATED
TO PROCESSES: NONE

PROCESSES CURRENTLY
WAITING: NONE

SIMULATION TIME = 3600000.00000 UNITS

ACTION REPORT

ACTION	TOTAL SAMPLES	MEAN.....	STD DEV...	MINIMUM..	MAXIMUM...	% TIME OF TOTAL.
CHGCD.OH						
USEFUL TIME	0	0.	0.	0.	0.	0.
DELAY TIME	0	0.	0.	0.	0.	

ACTION	TOTAL SAMPLES	MEAN.....	STD DEV...	MINIMUM..	MAXIMUM...	% TIME OF TOTAL.
CHQND.OH						
USEFUL TIME	0	0.	0.	0.	0.	0.
DELAY TIME	0	0.	0.	0.	0.	

ACTION	TOTAL SAMPLES	MEAN.....	STD DEV...	MINIMUM..	MAXIMUM...	% TIME OF TOTAL.
CS.OH						
USEFUL TIME	689	.420	.343	.000	.998	.008
DELAY TIME	689	0.	0.	0.	0.	

ACTION	TOTAL SAMPLES	MEAN.....	STD DEV...	MINIMUM..	MAXIMUM...	% TIME OF TOTAL.
DUMMYACT						
USEFUL TIME	0	0.	0.	0.	0.	0.
DELAY TIME	0	0.	0.	0.	0.	

ACTION	TOTAL SAMPLES	MEAN.....	STD DEV...	MINIMUM..	MAXIMUM...	% TIME OF TOTAL.
FORMAT						
USEFUL TIME	168	2.347	.337	2.000	2.998	.011
DELAY TIME	168	0.	0.	0.	0.	

ACTION	TOTAL SAMPLES	MEAN.....	STD DEV...	MINIMUM..	MAXIMUM...	% TIME OF TOTAL.
HQ.OH						
USEFUL TIME	0	0.	0.	0.	0.	0.
DELAY TIME	0	0.	0.	0.	0.	

ACTION	TOTAL SAMPLES	MEAN.....	STD DEV...	MINIMUM..	MAXIMUM...	% TIME OF TOTAL.

ACTION	TOTAL SAMPLES	MEAN.....	STD DEV...	MINIMUM..	MAXIMUM...	% TIME OF TOTAL.
LATENCY						
USEFUL TIME	168	14.989	8.552	.132	30.289	.070
DELAY TIME	168	0.	0.	0.	0.	

ACTION	TOTAL SAMPLES	MEAN.....	STD DEV...	MINIMUM..	MAXIMUM...	% TIME OF TOTAL.
OVERHEAD						
USEFUL TIME	0	0.	0.	0.	0.	0.
DELAY TIME	0	0.	0.	0.	0.	

ACTION	TOTAL SAMPLES	MEAN.....	STD DEV...	MINIMUM..	MAXIMUM...	% TIME OF TOTAL.
ROUTE.OH						
USEFUL TIME	1880	80.243	.306	80.000	80.999	4.190
DELAY TIME	1880	0.	0.	0.	0.	

ACTION	TOTAL SAMPLES	MEAN.....	STD DEV...	MINIMUM..	MAXIMUM...	% TIME OF TOTAL.
SEEK						
USEFUL TIME	168	29.723	17.260	.490	60.410	.139
DELAY TIME	168	0.	0.	0.	0.	

ACTION	TOTAL SAMPLES	MEAN.....	STD DEV...	MINIMUM..	MAXIMUM...	% TIME OF TOTAL.
UPDATE						
USEFUL TIME	354	8.026	5.234	.100	12.225	.079
DELAY TIME	354	0.	0.	0.	0.	

ACTION	TOTAL SAMPLES	MEAN.....	STD DEV...	MINIMUM..	MAXIMUM...	% TIME OF TOTAL.
XFER						
USEFUL TIME	168	.812	.313	.500	1.498	.004
DELAY TIME	168	0.	0.	0.	0.	

ACTION	TOTAL SAMPLES	MEAN.....	STD DEV...	MINIMUM..	MAXIMUM...	% TIME OF TOTAL.
XFER.OH						
USEFUL TIME	2047	3904.679	6018.850	81.380	16276.996	222.024
DELAY TIME	2047	0.	0.	0.	0.	

```
SIMULATION TIME =    3600000.00000 UNITS

PROCESS REPORT

                     TOTAL
PROCESS            SAMPLES. SUM....... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
AB-DATA
          TOTAL      118    0.          0.          0.          0.          0.
   PROCESS WAIT        0    0.          0.          0.          0.          0.
  RESOURCE WAIT        0    0.          0.          0.          0.          0.

        TOTAL #  # AUTO   # CALL   # OF     # NOT    # TIMES
        SCHEDULE SCHEDULE SCHEDULE COMPLETE COMPLETE SUSPEND.
        ======== ======== ======== ======== ======== ========
            118      118        0      118        0        0

PROCESS           DESCRIPTION
======== ======== ====================================================
AB-DATA           AIR BASE STATUS BROADCAST TO ALL OTHER NODES

COUNT ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
===== ======== ======== ======== ======== ======== =========================
  118          START             NO
  118          GIVEN    MSG
  118          RETURN   MSG
  118          CALL     REQ-I/O  NOWAIT   10       PROCESS REQUEST TO CHQ
  118          GIVEN    CHQ-DATA 10       $NOWAIT
  118                   750      CHQ
  118          CALL     REQ-I/O  NOWAIT   10       PROCESS REQUEST TO HQ
  118          GIVEN    HQ-DATA  10       $NOWAIT
  118                   750      HQ
  118          ASSIGN   $CNODE                     CURRENT NODE
  118                   CNODE
  118          COMPARE  CNODE             EQ       TEST FOR CURRENT NODE
  118                   AB1               AB1
   59          CALL     REQ-I/O  NOWAIT   10       PROCESS REQUEST TO AB1
   59          GIVEN    ABUPDATE 10       $NOWAIT
   59                   750      AB1
   59          BRANCH   END      100               BRANCH TO THE END
   59 AB1      ENTRY                               ENTRY FROM COMPARE NODE
   59          CALL     REQ-I/O  NOWAIT   10       PROCESS REQUEST TO AB2
   59          GIVEN    ABUPDATE 10       $NOWAIT
   59                   750      AB2
  118 END      ENTRY                               ENTRY FROM REQUEST TO AB1
  118          END

                 TOTAL
```

```
PROCESS         SAMPLES. SUM....... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
AB-REQ
          TOTAL      120  6.248E 07 520643.198 252860.380  37771.563 878047.214
   PROCESS WAIT      120  6.248E 07 520643.198 252860.380  37771.563 878047.214
  RESOURCE WAIT        0        0.         0.         0.         0.         0.

        TOTAL #  # AUTO   # CALL   # OF     # NOT    # TIMES
        SCHEDULE SCHEDULE SCHEDULE COMPLETE COMPLETE SUSPEND.
        ======== ======== ======== ======== ======== ========
            120      120        0      120        0        0

PROCESS           DESCRIPTION
======== ======== ==================================================
AB-REQ            AIRBASE REQUEST FOR PLANS REPORT FROM CHQ

COUNT ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
===== ======== ======== ======== ======== ======== ==========================
  120          START             NO
  120          GIVEN    MSG
  120          RETURN   MSG
  120          CALL     REQ-I/O  WAIT     5        PROCESS REQUEST TO CHQ
  120          GIVEN    PLANS    5        $WAIT
  120                   200      CHQ
  120          END

                 TOTAL
PROCESS         SAMPLES. SUM....... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
ABUPDATE
          TOTAL      118     75.319       .638       .354       .100      1.080
   PROCESS WAIT        0     0.         0.         0.         0.         0.
  RESOURCE WAIT        0     0.         0.         0.         0.         0.

        TOTAL #  # AUTO   # CALL   # OF     # NOT    # TIMES
        SCHEDULE SCHEDULE SCHEDULE COMPLETE COMPLETE SUSPEND.
        ======== ======== ======== ======== ======== ========
            116        0      118      118        0        0

         ITEM     CREATED  RECEIVED SENT     DESTR'D
         ======== ======== ======== ======== ========
         MSG             0        0        0        0

                    PROCESS  HOLDING  TIME
         ITEM     # SMPLS  MEAN...... MINIMUM..  MAXIMUM... STD DEV...
         ======== ======== ========== ========== ========== ==========
         MSG           118        .64        .10       1.08        .35

PROCESS           DESCRIPTION
```

ABUPDATE — UPDATE DATA FROM AIRBASE

COUNT	ENTRY	OPCODE	PARM	PARM	PARM	COMMENT
118		START		NO		
118		GIVEN	MSG			
118		RETURN	MSG			
118		UPDATE	CONSTANT 0.1			TIME CONSUMED IN UPDATING
118		END				

PROCESS	TOTAL SAMPLES.	SUM.......	MEAN......	STD DEV...	MINIMUM...	MAXIMUM...
CHLIO						
TOTAL	2047	9.095E 06	4443.055	6034.026	81.380	18702.407
PROCESS WAIT	0	0.	0.	0.	0.	0.
RESOURCE WAIT	2048	1.103E 06	538.421	1657.106	0.	16116.024

TOTAL # SCHEDULE	# AUTO SCHEDULE	# CALL SCHEDULE	# OF COMPLETE	# NOT COMPLETE	# TIMES SUSPEND.
2048	0	2048	2047	1	0

ITEM	CREATED	RECEIVED	SENT	DESTR'D
MSG	0	0	0	0

PROCESS HOLDING TIME

ITEM	# SMPLS	MEAN......	MINIMUM..	MAXIMUM...	STD DEV...
MSG	2047	4443.05	81.38	18702.41	6034.03

PROCESS	DESCRIPTION
CHLIO	FULL AND HALF DUPLEX CHANNEL LOGIC

COUNT	ENTRY	OPCODE	PARM	PARM	PARM	COMMENT
2048		START	ALL	NO		
2049		GIVEN	MSG			
2048		ASSIGN	MSG	CNODE		SET INTERNAL NODE CURRENT
2048			$CNODE			
2048		ASSIGN	MSG	TNODE		GET DESTINATION NODE (MSG)
2048			TO.NODE			
2048		ASSIGN	$NXTNODE	TO.NODE		SET NEXT NODE TO DEST N
2048			NXT.NODE			
2048		ASSIGN	$CHANNEL	TO.NODE		GET CHANNEL TO NEXT NODE
2048			CHANNEL			

```
 2048        ALLOC    CHANNEL                      OBTAIN CHANNEL FOR X  FER
 2048        ASSIGN   CHANNEL  RATE                WHAT IS RATE IN MSEC/BYTE?
 2048                 VSPEED
 2048        ASSIGN   MSG      LENGTH              MESSAGE LENGTH IN BYTES
 2048                 V.LENGTH
 2048        EVAL     VM.OVHD  MULTIPLY            CALCULATE TRANSFER TIME
 2048                 VSPEED   VLENGTH
 2048        XFER.OH  CONSTANT VM.OVHD             DELAY DUE TO TRANSFER TIME
 2047        ASSIGN   NXT.NODE                     MESSAGE RESIDES IN NEXT
 2047                 MSG      CNODE
 2047        ASSIGN   NXT.NODE                     SET INTERNAL NODE REGISTER
 2047                 $CNODE
 2047        DEALLOC  CHANNEL                      FREE UP CHANNEL AFTER XFER
 2047        CALL     IHANDLER NOWAIT   0          INDICATE INTERUPT IN NEXT
 2047        GIVEN    MSG
 2047        END

                   TOTAL
PROCESS          SAMPLES. SUM....... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
CHQ-DATA
          TOTAL      118   1387.260     11.756       .259     11.350     12.166
   PROCESS WAIT        0        0.          0.         0.         0.         0.
  RESOURCE WAIT        0        0.          0.         0.         0.         0.

        TOTAL #  # AUTO   # CALL   # OF     # NOT    # TIMES
        SCHEDULE SCHEDULE SCHEDULE COMPLETE COMPLETE SUSPEND.
        ======== ======== ======== ======== ======== ========
           118        0      118      118        0        0

         ITEM     CREATED  RECEIVED SENT     DESTR'D
         ======== ======== ======== ======== ========
         MSG             0        0        0        0

                   PROCESS  HOLDING  TIME
         ITEM     # SMPLS  MEAN...... MINIMUM..  MAXIMUM... STD DEV...
         ======== ======== ========== ========== ========== ==========
         MSG           118      11.76      11.35      12.17        .26

PROCESS           DESCRIPTION
======== ======== ======================================================
CHQ-DATA          CHQ GETS MESSAGE, FORMULATES RESPONSE, AND REPLIES

COUNT ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
===== ======== ======== ======== ======== ======== ===========================
  118          START             NO
  118          GIVEN    MSG
  118          RETURN   MSG
  118          ASSIGN   MSG      LENGTH            MAKE MSG-LENGTH = V.LENGTH
```

```
PAGE   45
  118                    V.LENGTH
  118           EVAL     V.TIME   MULTIPLY          EVALUATE MSG PROCESS TIME
  118                    .015     V.LENGTH
  118           UPDATE   CONSTANT V.TIME            PROCESSING TIME CONSUMED
  118           END

                  TOTAL
PROCESS         SAMPLES. SUM....... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
CONTROL
         TOTAL       689  1.615E 08 234440.181 487580.344       .002  2.424E 06
   PROCESS WAIT      690  2.745E 06   3978.465   6976.703       .100  16276.996
  RESOURCE WAIT      689  1.588E 08 230456.357 486573.500       .000  2.424E 06

        TOTAL #  # AUTO   # CALL   # OF     # NOT    # TIMES
        SCHEDULE SCHEDULE SCHEDULE COMPLETE COMPLETE SUSPEND.
        ======== ======== ======== ======== ======== ========
             689        0      689      689        0        0

         ITEM     CREATED  RECEIVED SENT     DESTR'D
         ======== ======== ======== ======== ========
         MSG             0        0        0      521

                   PROCESS  HOLDING  TIME
         ITEM     # SMPLS  MEAN...... MINIMUM.. MAXIMUM... STD DEV...
         ======== ======== ========== ========= ========== ==========
         MSG          1211  131118.63       .00 2423842.59  384351.85

PROCESS           DESCRIPTION
======== ======== ==================================================
CONTROL           OPERATING SYSTEM : CONTEXT SWITCHING

COUNT ENTRY     OPCODE   PARM     PARM     PARM     COMMENT
===== ======== ======== ======== ======== ======== ========================
  689           START    ALL      NO
  689           GIVEN    MSG
  689           ASSIGN   MSG      CNODE             CURRENT NODE IS CPU
  689                    CP
  689           ALLOC    CP                         SIGNAL CURRENT CPU BUSY
  689           ASSIGN   CP       OSOVHD            MEAN CONTEXT SWITCH TIME
  689                    M.OVHD
  689           CS.OH    CONSTANT M.OVHD            DELAY CONTEXT SWITCH TIME
  689           COMPARE  MSG      TYPE     EQ       IF RESPONSE- RESUME PARENT
  689                    $PEQ              REQUEST
  167           ASSIGN   MSG      PTASK             TASK TO RESUME IS IN MSG
  167                    TASK
  167           RESUME   TASK              QUEUE UP TASK FOR NODE
  167           BRANCH   DESTROY  100               END MESSAGE LIFE
  522 REQUEST   ENTRY                               ELSE-> CALL REQUESTED PROC
```

```
 522          ASSIGN   MSG      RTASK              EXECUTE THE CALLED PROCESS
1522                   PROCESS
 522          CALL     PROCESS  WAIT      0        WAIT UNTIL COMPLETE
 522          GIVEN    MSG
 522          RETURN   MSG
 522          COMPARE  MSG      RESPONSE EQ        IF WAIT -> SEND MSG BACK
 522                   $NOWAIT            DESTROY
 168          ASSIGN   $RESP                       CHANGE MSG RESPONSE TYPE
 168                   MSG      TYPE
 168          ASSIGN   MSG      FNODE              SWITCH FROM AND TO NODES
 168                   MSG      TNODE
 168          ASSIGN   MSG      CNODE              CURRENT NODE IS FROM NODE
 168                   MSG      FNODE
 168          CALL     CHLIO    WAIT      0        RETURN MESSAGE TO ORIGIN
 168          GIVEN    MSG
 168          BRANCH   END      100
 521 DESTROY  ENTRY                                TERMINATE MESSAGE AT DEST.
 521          DESTROY  MSG                         NO RESPONSE-TERMINATE MSG
 689 END      ENTRY
 689          DEALLOC  CP                          INDICATE CP SWITCH DONE
 689          END

                 TOTAL
PROCESS         SAMPLES. SUM....... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
DISK.OP
          TOTAL      168   7542.384     44.895     18.992      4.130     87.020
   PROCESS WAIT        0      0.          0.         0.         0.         0.
  RESOURCE WAIT      168     58.252       .347       .337       .000       .998

        TOTAL #  # AUTO   # CALL   # OF     # NOT    # TIMES
        SCHEDULE SCHEDULE SCHEDULE COMPLETE COMPLETE SUSPEND.
        ======== ======== ======== ======== ======== ========
            168        0      168      168        0        0

PROCESS           DESCRIPTION
======== ======== ================================================
DISK.OP           OPERATION OF DISK

COUNT ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
===== ======== ======== ======== ======== ======== =======================
  168          START             NO
  168          GIVEN    LENGTH   DISK
  168          ASSIGN   DISK     SPEED             MAKE DISK SPEED = V.SPEED
  168                   V.SPEED
  168          EVAL     XFERTIME DIVIDE            TRANSFER TIME CALCULATED
  168                   LENGTH   V.SPEED
  168          ALLOC    DISK                       DISK ALLOCATED
  168          ASSIGN   DISK     SEEK              MAKE SEEKTIME = SEEK
```

```
PAGE   47
  168                     SEEKTIME
  168          SEEK       UNIFORM  SEEKTIME SEEKTIME TIME FOR SEEK IS CONSUMED
  168          ASSIGN     DISK     LATENCY           MAKE DISKLATENCY=LATETIME
  168                     LATETIME
  168          LATENCY    UNIFORM  LATETIME LATETIME TIME CONSUMED FOR LATENCY
  168          XFER       CONSTANT XFERTIME          TRANSFER TIME CONSUMED
  168          DEALLOC    DISK                       DISK RESOURCE DEALLOCATED
  168          END

                TOTAL
PROCESS        SAMPLES. SUM....... MEAN...... STD DEV... MINIMUM... MAXIMUM...
============== ======== ========== ========== ========== ========== ==========
ESR-CALL
         TOTAL     521  6.553E 07 125772.347 248440.463     80.000 878047.214
  PROCESS WAIT     522  41795.506     80.068       .196     80.000     80.996
 RESOURCE WAIT     167     56.907       .341       .329       .002       .996

        TOTAL #  # AUTO   # CALL   # OF     # NOT    # TIMES
        SCHEDULE SCHEDULE SCHEDULE COMPLETE COMPLETE SUSPEND.
        ======== ======== ======== ======== ======== ========
            522        0      522      521        1        0

         ITEM     CREATED  RECEIVED SENT     DESTR'D
         ======== ======== ======== ======== ========
         MSG             0        0        0        0

                    PROCESS  HOLDING  TIME
         ITEM     # SMPLS  MEAN...... MINIMUM..  MAXIMUM... STD DEV...
         ======== ======== ========== ========== ========== ==========
         MSG           522        .07        0.        1.00        .20

PROCESS           DESCRIPTION
======== ======== ==================================================
ESR-CALL          OPERATING SYSTEM: EXECUTIVE SERVICE REQUEST (CALL )

COUNT ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
===== ======== ======== ======== ======== ======== ========================
  522          START    ALL      NO
  522          GIVEN    MSG
  522          ASSIGN   $TASK                      $TASK= INSTANCE TO RESUME
  522                   MSG      PTASK
  522          ASSIGN   MSG      RESPONSE          OPTION= $WAIT OF $NOWAIT
  522                   RESP.OPT
  522          CALL     ROUTER   WAIT     0        INITIATE ROUTING TO DEST.
  522          GIVEN    MSG
  522          COMPARE  RESP.OPT          EQ       SHOULD PARENT SUSPEND ?
  522                   $NOWAIT           END
  168          SUSPEND                             PROCESS CALLED WAIT
  521 END      ENTRY                               CONTINUE OR RESUME POINT
```

```
  521          END

                    TOTAL
PROCESS           SAMPLES. SUM....... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
HQ-DATA
          TOTAL      118   1378.545     11.683       .375     11.250     12.225
   PROCESS WAIT        0         0.         0.         0.         0.         0.
  RESOURCE WAIT        0         0.         0.         0.         0.         0.

        TOTAL #  # AUTO   # CALL   # OF     # NOT    # TIMES
        SCHEDULE SCHEDULE SCHEDULE COMPLETE COMPLETE SUSPEND.
        ======== ======== ======== ======== ======== ========
             118        0      118      118        0        0

         ITEM     CREATED  RECEIVED SENT     DESTR'D
         ======== ======== ======== ======== ========
         MSG             0        0        0        0

                  PROCESS  HOLDING  TIME
         ITEM     # SMPLS  MEAN...... MINIMUM..  MAXIMUM... STD DEV...
         ======== ======== ========== ========== ========== ==========
         MSG           118      11.68      11.25      12.23        .38

PROCESS           DESCRIPTION
======== ======== ==================================================
HQ-DATA           HQ GETS MESSAGE, FORMULATES RESPONSE, AND REPLIES

COUNT ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
===== ======== ======== ======== ======== ======== ========================
  118          START             NO
  118          GIVEN    MSG
  118          RETURN   MSG
  118          ASSIGN   MSG      LENGTH            MAKE MSG-LENTH = V.LENGTH
  118                   V.LENGTH
  118          EVAL     V.TIME   MULTIPLY          EVALUATE MSG PROCESS TIME
  118                   .015     V.LENGTH
  118          UPDATE   CONSTANT V.TIME            PROCESSING TIME CONSUMED
  118          END

                    TOTAL
PROCESS           SAMPLES. SUM....... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
HQ-REQ
          TOTAL       47  3.022E 06  64295.511   9015.565  54524.451  88274.384
   PROCESS WAIT       47  3.022E 06  64295.511   9015.565  54524.451  88274.384
  RESOURCE WAIT        0         0.         0.         0.         0.         0.

        TOTAL #  # AUTO   # CALL   # OF     # NOT    # TIMES
```

```
        SCHEDULE SCHEDULE SCHEDULE COMPLETE COMPLETE SUSPEND.
        ======== ======== ======== ======== ======== ========
              48       48        0       47        1        0

PROCESS           DESCRIPTION
======== ======== ====================================================
HQ-REQ            HQ REQUEST FOR STATUS DISPLAY FROM CHQ

COUNT ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
===== ======== ======== ======== ======== ======== =========================
   48          START    L3       NO
   48          GIVEN    MSG
   48          RETURN   MSG
   48          CALL     REQ-I/O  WAIT     4        MAKES I/O REQUEST TO CHQ
   48          GIVEN    PLANS    4        $WAIT
   48                   200      CHQ
   47          END

                 TOTAL
PROCESS         SAMPLES. SUM....... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
IHANDLER
          TOTAL     2047 128683.828     62.865     51.069      0.       229.028
   PROCESS WAIT        0      0.         0.         0.         0.         0.
  RESOURCE WAIT     1358  19793.352     14.575     30.719      0.       149.028

        TOTAL #  # AUTO   # CALL   # OF     # NOT    # TIMES
        SCHEDULE SCHEDULE SCHEDULE COMPLETE COMPLETE SUSPEND.
        ======== ======== ======== ======== ======== ========
            2047        0     2047     2047        0        0

         ITEM     CREATED  RECEIVED SENT     DESTR'D
         ======== ======== ======== ======== ========
         MSG             0        0        0        0

                   PROCESS  HOLDING  TIME
         ITEM     # SMPLS  MEAN...... MINIMUM..  MAXIMUM... STD DEV...
         ======== ======== ========== ========== ========== ==========
         MSG          2047      62.86         0.     229.03      51.07

PROCESS           DESCRIPTION
======== ======== ====================================================
IHANDLER          OPERATING SYSTEM : INTERUPT HANDLING AND ROUTING

COUNT ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
===== ======== ======== ======== ======== ======== =========================
 2047          START    ALL      NO
 2047          GIVEN    MSG
 2047          ASSIGN   MSG      CNODE             INDICATE CURRENT NODE CPU
```

```
 2047                    CP
 2047          COMPARE   MSG       CNODE     EQ        IS MSG AT DESTINATION ?
 2047                    MSG       TNODE     CONTROL
 1358          ASSIGN    CP        NETINSTR            MONITOR OVERHEAD FOR PLOT
 1358                    M.OVHD
 1358          ALLOC     CP                            OBTAIN CP-HANDLE INTERUPT
 1358          ROUTE.OH  CONSTANT  M.OVHD              DELAY FOR ROUTING
 1358          DEALLOC   CP                            RELEASE CPU TO OTHERS
 1358          CALL      CHLIO     NOWAIT    0         FORWARD MESSAGE WITH I/O
 1358          GIVEN     MSG
 1358          BRANCH    END       100
  689 CONTROL  ENTRY                                   MESSAGE AT DESTINATION
  689          COMPARE   MSG       TYPE      EQ        IF RESPONSE-UP PRIORITY
  689                    $RESP               HPCONTRL
  522          ASSIGN    MSG       TASKPRI             SET MESSAGE PRIORITY
  522                    PRIORITY
  689 HPCONTRL ENTRY                                   PRIORITY=0 IF UNDEFINED
  689          CALL      CONTROL   NOWAIT    PRIORITY  CONTEXT SWITCH MESSAGE
  689          GIVEN     MSG
 2047 END      ENTRY
 2047          END
```

PROCESS	TOTAL SAMPLES.	SUM.......	MEAN......	STD DEV...	MINIMUM...	MAXIMUM...
PLANS						
TOTAL	168	7878.384	46.895	18.992	6.130	89.020
PROCESS WAIT	168	7542.384	44.895	18.992	4.130	87.020
RESOURCE WAIT	0	0.	0.	0.	0.	0.

TOTAL # SCHEDULE	# AUTO SCHEDULE	# CALL SCHEDULE	# OF COMPLETE	# NOT COMPLETE	# TIMES SUSPEND.
168	0	168	168	0	0

ITEM	CREATED	RECEIVED	SENT	DESTR'D
MSG	0	0	0	0

PROCESS HOLDING TIME

ITEM	# SMPLS	MEAN......	MINIMUM..	MAXIMUM...	STD DEV...
MSG	168	46.90	6.13	89.02	18.99

PROCESS	DESCRIPTION
PLANS	REQUEST FOR PLANS FROM CHQ

COUNT ENTRY OPCODE PARM PARM PARM COMMENT

```
===== ======== ======== ======== ======== ======== =========================
  168          START    CHQ      NO
  168          GIVEN    MSG
  168          RETURN   MSG
  168          ASSIGN   MSG      LENGTH            MAKE MSG LENGTH = V.LENGTH
  168                   V.LENGTH
  168          EVAL     V.TIME   MULTIPLY          EVALUATE MSG PROCESS TIME
  168                   .01      V.LENGTH
  168          FORMAT   CONSTANT V.TIME            TIME USED TO FORMAT PLANS
  168          CALL     DISK.OP  WAIT     10       CALLING PROCESS DISK.OP
  168          GIVEN    10000    DK1
  168          ASSIGN   10000                      INCREASE MSG LENGTH
  168                   MSG      LENGTH
  168          END

                 TOTAL
PROCESS         SAMPLES. SUM....... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
REQ-I/O
          TOTAL      521  6.553E 07 125772.347 248440.463     80.000 878047.214
   PROCESS WAIT      521  6.553E 07 125772.3.7 246440.463     80.000 878047.214
  RESOURCE WAIT        0   0.          0.         0.         0.         0.

        TOTAL #  # AUTO   # CALL   # OF     # NOT    # TIMES
        SCHEDULE SCHEDULE SCHEDULE COMPLETE COMPLETE SUSPEND.
        ======== ======== ======== ======== ======== ========
             522        0      522      521        1        0

         ITEM     CREATED  RECEIVED SENT     DESTR'D
         ======== ======== ======== ======== ========
         MSG           522        0        0        0

                  PROCESS  HOLDING  TIME
         ITEM     # SMPLS  MEAN...... MINIMUM..  MAXIMUM... STD DEV...
         ======== ======== ========== ========== ========== ==========
         MSG           522        .07       0.         1.00        .20

PROCESS           DESCRIPTION
======== ======== ==================================================
REQ-I/O           GENERATE A PROCESS REQUEST MESSAGE AND INITIATE I/O

COUNT ENTRY     OPCODE   PARM     PARM     PARM     COMMENT
===== ======== ======== ======== ======== ======== =========================
  522          START    ALL      NO
  522          GIVEN    PROCESS  PRIORITY RESP.OPT
  522                   MSG.LNTH TO.NODE
  522          CREATE   MSG                        CREATE MESSAGE DATA TO RTE
  522          ASSIGN   $CNODE                     INDICATE CURRENT NODE
  522                   MSG      CNODE
```

```
PAGE    52
  522          ASSIGN   $CNODE                       INDICATE CURRENT NODE FROM
  522                   MSG      FNODE
  522          ASSIGN   PROCESS                      INDICATE REQUESTED PROCESS
  522                   MSG      RTASK
  522          ASSIGN   PRIORITY                     INDICATE RELATIVE PRIORITY
  522                   MSG      TASKPRI
  522          ASSIGN   RESP.OPT                     $NOWAIT OR $ WAIT ON CALL
  522                   MSG      RESPONSE
  522          ASSIGN   MSG.LNTH                     INDICATE LENGTH IN BYTES
  522                   MSG      LENGTH
  522          COMPARE  TO.NODE           EQ         WHERE DOES PROCESS RESIDE
  522                   $NO               END
  522          COMPARE  TO.NODE           EQ         DEFAULT TO NODE SELECT
  522                   $YES              GETNODE
  522          ASSIGN   TO.NODE                      ELSE-> NODE IS GIVEN
  522                   MSG      TNODE
  522          BRANCH   END      100
    0 GETNODE  ENTRY                                 DETERMINE NODE FROM PROC.
    0          ASSIGN   $NODE    PROCESS             $NODE OF PROCESS IN DEF
    0                   MSG      TNODE
  522 END      ENTRY                                 SEND MSG FOR SERVICE
  522          CALL     ESR-CALL WAIT     0          EXECUTIVE SERVICING OF MSG
  522          GIVEN    MSG
  521          END

                  TOTAL
PROCESS         SAMPLES. SUM....... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
ROUTER
         TOTAL      522  41795.506     80.068       .196     80.000     80.996
  PROCESS WAIT        0         0.         0.         0.         0.         0.
 RESOURCE WAIT        0         0.         0.         0.         0.         0.

        TOTAL #  # AUTO   # CALL   # OF     # NOT    # TIMES
        SCHEDULE SCHEDULE SCHEDULE COMPLETE COMPLETE SUSPEND.
        ======== ======== ======== ======== ======== ========
             522        0      522      522        0        0

         ITEM     CREATED  RECEIVED SENT     DESTR'D
         ======== ======== ======== ======== ========
         MSG             0        0        0        0

                   PROCESS  HOLDING  TIME
         ITEM     # SMPLS  MEAN...... MINIMUM..  MAXIMUM... STD DEV...
         ======== ======== ========== ========== ========== ==========
         MSG           522      80.07      80.00      81.00        .20

PROCESS          DESCRIPTION
======== ======= ==================================================
```

```
ROUTER            OPERATING SYSTEM : INTERUPT HANDLING AND ROUTING

COUNT ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
===== ======== ======== ======== ======== ======== =========================
  522          START    ALL      NO
  522          GIVEN    MSG
  522          ASSIGN   MSG      CNODE             INDICATE CURRENT NODE CPU
  522                   CP
  522          COMPARE  MSG      CNODE    EQ       IS MSG AT DESTINATION ?
  522                   MSG      TNODE    CONTROL
  522          ASSIGN   CP       NETINSTR          MONITOR OVERHEAD FOR PLOT
  522                   M.OVHD
  522          ROUTE.OH CONSTANT M.OVHD            DELAY FOR ROUTING
  522          CALL     CHLIO    NOWAIT   0        FORWARD MESSAGE WITH I/O
  522          GIVEN    MSG
  522          BRANCH   END      100
    0 CONTROL  ENTRY                               MESSAGE AT DESTINATION
    0          COMPARE  MSG      TYPE     EQ       IF RESPONSE-UP PRIORITY
    0                   $RESP             HPCONTRL
    0          ASSIGN   MSG      TASKPRI           SET MESSAGE PRIORITY
    0                   PRIORITY
    0 HPCONTRL ENTRY                               PRIORITY=0 IF UNDEFINED
    0          CALL     CONTROL  NOWAIT   PRIORITY CONTEXT SWITCH MESSAGE
    0          GIVEN    MSG
  522 END      ENTRY
  522          END

                  TOTAL
PROCESS          SAMPLES. SUM....... MEAN...... STD DEV... MINIMUM... MAXIMUM...
=============== ======== ========== ========== ========== ========== ==========
TRACE
          TOTAL        1       0.         0.         0.         0.         0.
   PROCESS WAIT        0       0.         0.         0.         0.         0.
  RESOURCE WAIT        0       0.         0.         0.         0.         0.

         TOTAL #  # AUTO   # CALL   # OF     # NOT    # TIMES
         SCHEDULE SCHEDULE SCHEDULE COMPLETE COMPLETE SUSPEND.
         ======== ======== ======== ======== ======== ========
                1        1        0        1        0        0

PROCESS           DESCRIPTION
======== ======== ==================================================
TRACE             TURN ON TRACE OUTPUT

COUNT ENTRY    OPCODE   PARM     PARM     PARM     COMMENT
===== ======== ======== ======== ======== ======== =========================
    1          START    ALL      NO
    1          COMPARE  V.TRACE           EQ       TEST IF FLAG SET FOR TRACE
    1                   0                 NOTRACE
```

```
PAGE   54
     0            TRACE     ON
     1 NOTRACE    ENTRY
     1            END
```

```
PAGE   54
     0            TRACE     ON
     1 NOTRACE    ENTRY
     1            END
```

MED

8