

AD A137 057

COMPUTER GENERATION OF PLAN OF ACTION AND MILESTONE
SCHEDULE (U) DAVID W TAYLOR NAVAL SHIP RESEARCH AND
DEVELOPMENT CENTER BET., A SHUFORD ET AL. NOV R3
DTNSRDC (MLD R3/27)

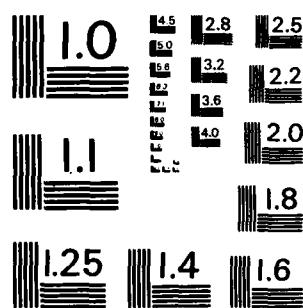
1/1

UNCLASSIFIED

F/G 9/2

NI

END
DATE
FILED
2 284
DTM



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS - 1963 - A

(12)

ADA137057

DTNSRDC-CM(LD-83/27)

COMPUTER GENERATION OF PLAN OF ACTION AND MILESTONE SCHEDULE

DAVID W. TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CENTER

Bethesda, Maryland 20084



COMPUTER GENERATION OF PLAN OF ACTION AND MILESTONE SCHEDULE

bv

Annette Shuford
Susan Becker

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

COMPUTATION, MATHEMATICS AND LOGISTICS DEPARTMENT
DEPARTMENTAL REPORT



DTNSRDC-CM(LD-83/27)

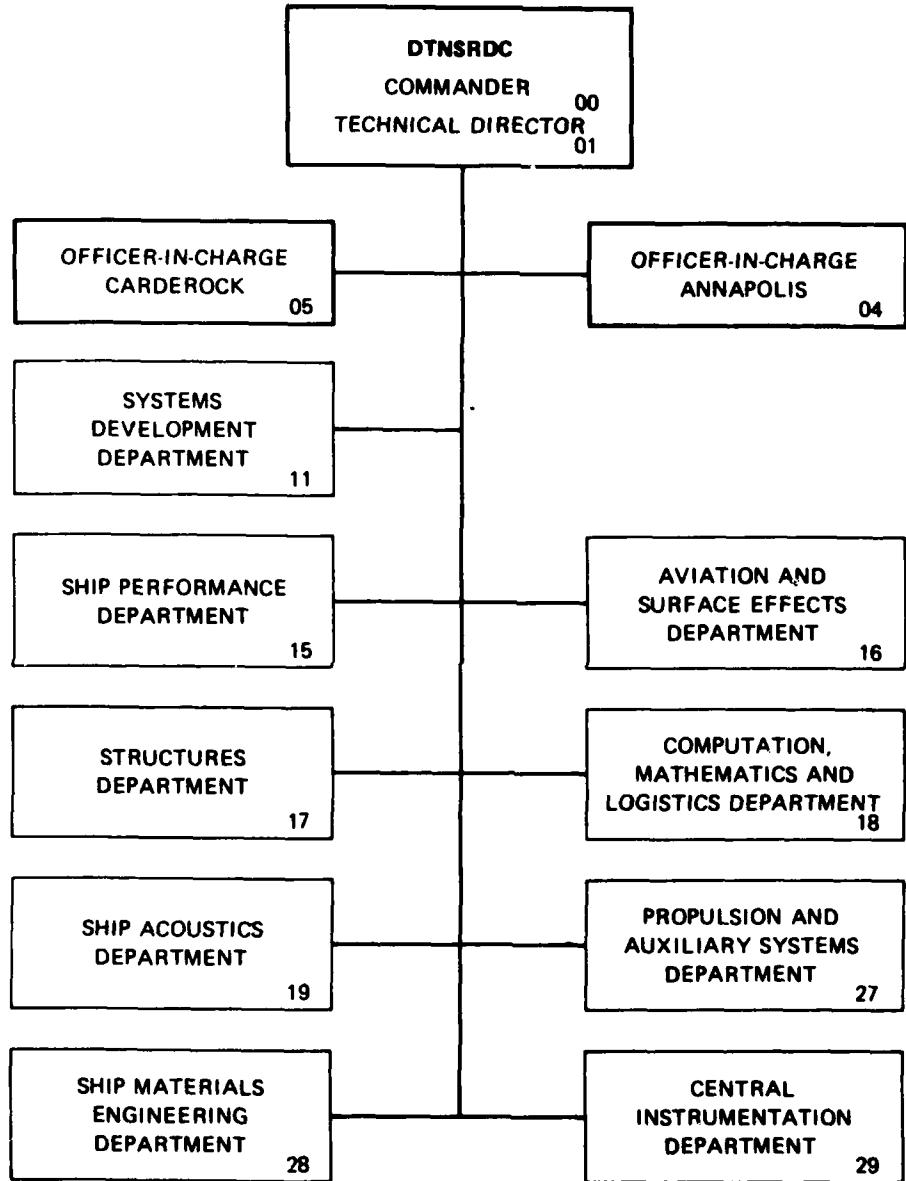
November 1983

DTIC FILE COPY

NW-DTNSRDC 5802/30 (2-80)
(supersedes 3880/46)

84 01 20 04 1

MAJOR DTNSRDC ORGANIZATIONAL COMPONENTS



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER DTNSRDC-CMLD-83/27	2. GOVT ACCESSION NO. AIAA1057	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) COMPUTER GENERATION OF PLAN OF ACTION AND MILESTONE SCHEDULE	5. TYPE OF REPORT & PERIOD COVERED Final	
7. AUTHOR(s) Annette Shuford Susan Becker	6. PERFORMING ORG. REPORT NUMBER	
9. PERFORMING ORGANIZATION NAME AND ADDRESS David Taylor Naval Ship Research and Development Center Bethesda, Maryland 20084	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NIF 1828-023	
11. CONTROLLING OFFICE NAME AND ADDRESS Navy Publications and Printing Service Bldg. 157-3, Washington Navy Yard Washington, DC 200734	12. REPORT DATE November 1983	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	13. NUMBER OF PAGES 81	
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) POAM TECTRONIX DISSPLA CALCOMP PLOT GRAPHICS FORTRAN		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The Computer Generated Plan of Action and Milestones (POAM) Program tracks the progress of a project using computer graphics. Included is an explanation of DISSPLA, the computer graphics language used to graph the POAM chart. The program is implemented on the CDC 6600 computer at DTNSRDC.		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 68 IS OBSOLETE
S/N 0102-LF-014-6601**UNCLASSIFIED**

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

TABLE OF CONTENTS

	Page
ABSTRACT	1
1.0. INTRODUCTION AND BACKGROUND	1
2.0. PROGRAM DESCRIPTION	1
3.0. SETTING UP THE PROGRAM	3
3.1. User's Input	3
3.2. Sample Task	3
3.3. Predefined Variables	6
3.4. FORTRAN Subroutines	6
3.4.1. Subroutine VECTPL	8
3.4.2. Subroutine MARKPL	8
3.4.3. Subroutine YCOORD	9
3.4.4. Subroutine CURVPL	13
3.5. Using DISSPLA Subroutines	13
3.6. Plot Generation	14
4.0. USER DEFINED POAMs	15
4.1. The Blank Chart	15
4.2. Chart Statements	18
5.0. DISSPLA	18
5.1. Levels of DISSPLA	18
5.2. DISSPLA Functions	21
5.2.1. MESSAG	21
5.2.2. HEIGHT	22
5.2.3. BASALF and MIXALF	22
5.3. Chart Vectors and Markers	25
5.3.1. Vectors	25

5.3.2	Markers	27
5.4.	CURVE vs MESSAG	27
5.5.	User Defined Ticks (Scaling)	31
5.6.	Routines	35
6.0.	CONCLUSION	35
APPENDIX A		39
APPENDIX B		42
APPENDIX C		46
APPENDIX D		52
APPENDIX E		59
APPENDIX F		67

LIST OF FIGURES

	Page
1 - Example of a Finished POAM	2
2 - Program Deck	4
3 - Plot for Sample Task	7
4 - Manually Produced Blank Chart	17
5 - Bar Chart Containing Extensive CLUSTR Imperfections	19
6 - Example Program Deck (Showing levels of DISSPLA)	20
7 - Upper Case Letter Sizes	23
8 - Vector Types	26
9 - Location of Months Chart	28
10 - Location of Month Markers Chart	29
11 - CURVE versus MESSAG Example Plot	30
12 - CURVE versus MESSAG Example Program	32

LIST OF TABLES

1 - Table of Variables	10
2 - Table of DISSPLA Routines	36

ABSTRACT

The Computer Generated Plan of Action and Milestones (POAM) Program tracks the progress of a project using computer graphics. Included is an explanation of DISSPLA, the computer graphics language used to graph the POAM chart. The program is implemented on the CDC 6600 computer at DTNSRDC.

1.0. INTRODUCTION AND BACKGROUND

The Computer Generated Plan of Action and Milestones (POAM) Program was developed to generate a Plan of Action and Milestones Chart using computer graphics. These charts are used in reports and system Decision Papers. (see Figure 1). The charts show the progress and methodology of a project. The original charts were generated by hand (ruler and pencil with a final typed copy) and by graphics procedures. The computer program will enable the user to generate these graphs faster, and more efficiently.

The program uses Fortran, and Display Integrated Software System and Plotting LAnguage (DISSPLA). DISSPLA is a software package that enables the user to produce graphics. The user is advised to become familiar with the introductory protions (Part A. Sections 1 through 4) of the DISSPLA manual. This manual explains how POAMs are produced through interactive and batch computer methods.

2.0. PROGRAM DESCRIPTION

A POAM chart indicates the progress of a project by showing the various steps of the project and their initiation and completion dates. Each step is given a number or Action I'. Then, a brief description of the step is given, followed by the official responsible for completing its task. Following this is a vector (a line indicating the duration of the action) which points to the fiscal year and month of the start and stop dates of that step. These dates are verified in the final columns of the chart. This report describes how to generate such a POAM.

The most recent CGPP (Computer Generated POAM Program) uses a basic skeleton or frame of the chart and allows the user to fill in the pertinent information. This frame consists of the POAM chart and its headings. The code for producing this chart should be the first section of the program, and then any specific information can be added. The code to produce this frame is given in Appendix A.

PRMIS II PLAN OF ACTION AND MILESTONES

MILESTONE ID	DESCRIPTION	RESPONSIBLE OFFICIAL	FY 81 AND SUPPORTERS	FY 82 AND SUPPORTERS	FY 83 AND SUPPORTERS	FY 84 AND SUPPORTERS	FY 85 AND SUPPORTERS	INITIATION DATES ESTIMATED / ACTUAL	COMPLETION DATES ESTIMATED / ACTUAL
MILESTONE 0 - MISSION ANALYSIS/PROJECT INITIATION									
0.1	Development of PRMS	NPSHQ						JAN 81	FEB 81
0.2	Establish Project Office	NPSHQ						FEB 81	FEB 81
0.3	Propose PRMS, AFIP, and Telecommunications Materiel	DTNSROD						FEB 81	FEB 81
0.4	Develop Plan of Action and Milestones	NAVDAC						FEB 81	FEB 81
0.5	Approval of PRMS							FEB 81	FEB 81
MILESTONE 1 - CONCEPT DEVELOPMENT PHASE									
1.1	Select Plan Issues PECI Issues	NPSHQ							
1.2	Conduct General Functional Requirements Analysis	DTNSROD							
1.3	Define Plan Experiments/Demonstrations	DTNSROD							
1.4	Develop Preliminary Economic Analysis	DTNSROD							
1.5	Develop Preliminary Budget and Cost Estimate	DTNSROD							
1.6	Develop Pilot System Specifications	DTNSROD							
1.7	Prepare Project Management Plan	DTNSROD							
1.8	Prepare Executive Summary	DTNSROD							
1.9	Develop Data Requirements	DTNSROD							
1.10	Prepare Preliminary Acquisition Strategy Annex	DTNSROD							
1.11	Prepare Support Planning Annex	DTNSROD							
1.12	Develop Pilot Source Data Equipment Specification	DTNSROD							
1.13	Pilot Computer Acquisition	DTNSROD							
1.14	User & Review Requirements	NPSHQ							
1.15	Initial Source Data Equipment Requirement	DTNSROD							
1.16	Forward SDR 1 for Review	DTNSROD							
1.17	Review SDR 1	NPSHQ							
1.18	Approve SDR 1	NAVSUP							
1.19	Review SDR 1	NAVTRADOC							
1.20	Approve SDR 1	NAVMCOP							
1.21	Review SDR 1	CAO							
1.22	Approve SDR 1	DTNSROD							
MILESTONE 11 - DEFINITION DESIGN PHASE									
2.1	Analyze results of Questionnaire	DTNSROD						DEC 81	DEC 82
2.2	Conduct & Finalize Pilot Experiment/Demonstration	DTNSROD						DEC 82	DEC 82
2.3	Prepare Functional Description	DTNSROD						MAY 82	MAY 82
2.4	Update Support Planning Annex	DTNSROD						JUN 82	JUN 82
2.5	Perform Economic Analysis	DTNSROD						JUN 82	JUN 82
2.6	TSPP	DTNSROD						JUN 82	JUN 82
2.7	Complete Data Requirements Document	DTNSROD						JUN 82	JUN 82
2.8	Develop Selection Document	DTNSROD						JUN 82	JUN 82
2.9	Review Selection Document	NPSHQ						JUN 82	JUN 82
2.10	Update SDR 1	NAVSUP						JUN 82	JUN 82
2.11	Review SDR 1	NAVMCOP						JUN 82	JUN 82
2.12	Update SDR 1	CAO						JUN 82	JUN 82
2.13	Review SDR 1	DTNSROD						JUN 82	JUN 82

Figure 1 - Example of a Finished POAM

To allow the user to add information with ease, several variables and subroutines have been predefined. These variables eliminate the need to compute the exact positions of the various vectors and dates. These variables and subroutines will be explained in detail in Sections 3.3 and 3.4, respectively.

If this basic frame is not sufficient for a particular application, the user may construct one (see Section 4) by changing the values used in the subroutines and variables. These changes should be minor and are based on the dimensions of the chart. The information in Appendix B is an example of such a situation.

3.0. SETTING UP THE PROGRAM

The program deck is divided into four basic sections: (1) The control cards necessary to run the program; (2) the code that produces the POAM frame, which can be copied directly from the example in Appendix A; (3) The information that "fills" the frame, all of which is applied by the user; and (4) the cards which close up the program and terminate the run. The user should set up the program deck as shown in Figure 2.

3.1. User's Input

When the basic frame of the chart is complete, the user is ready to add information to the chart. First the user inputs this information in a format suitable for a POAM chart by answering, or responding to the following questions:

- a. What is the ACTION ID of this step or task? If there is none, give the step an identification number.
- b. Briefly (no more than 50 characters) describe the function or purpose of this step.
- c. Determine who is responsible for completing this task.
- d. Determine the year and month that the task will be initiated.
- e. Determine the year and month that the task will be finished.
- f. What (if any) is the estimated initiation date?
- g. What (if any) is the actual initiation date?
- h. What (if any) is the estimated completion date?
- i. What (if any) is the actual completion date?

Session For	
...S GRA&I	
DTIC TAB	
Unannounced	
Justification	
By _____	
Distribution/ _____	
Availability Codes	
Dist	Avail and/or Special
A-1	



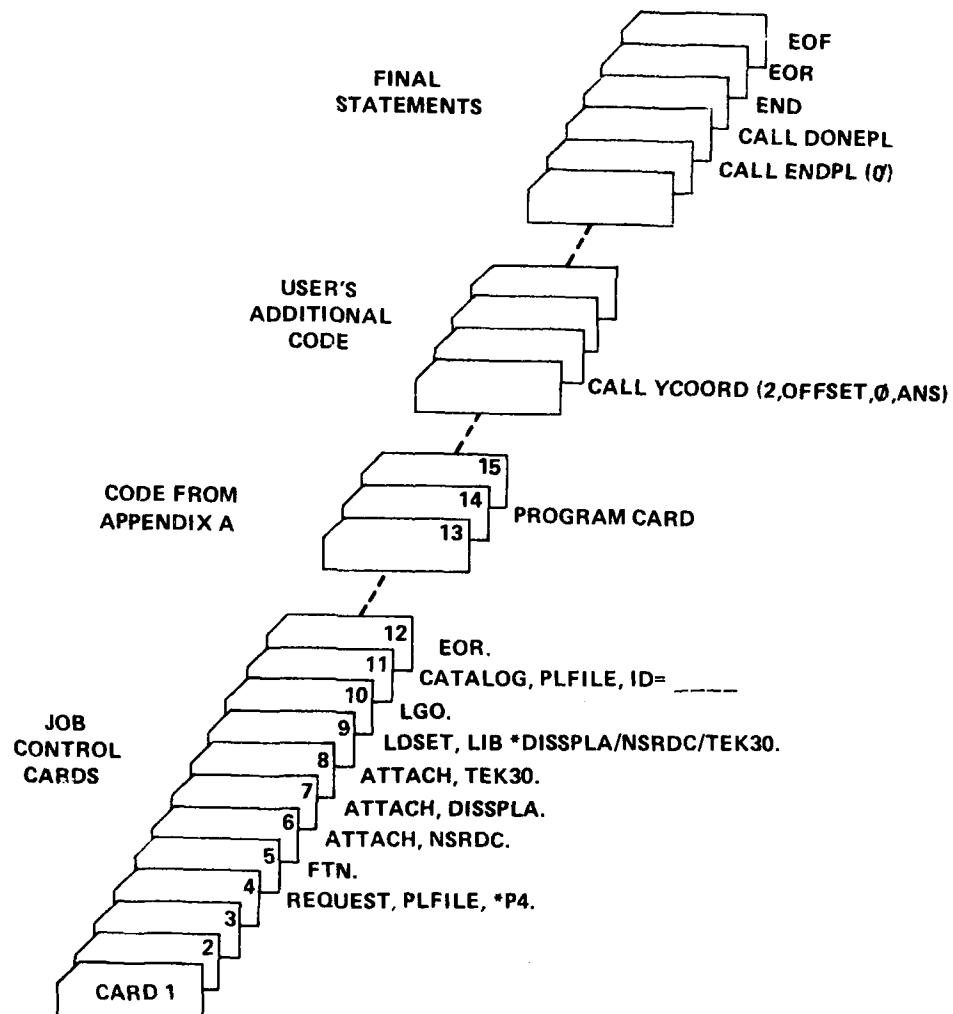


Figure 2 - Program Deck

Once these questions have been answered, the user is ready to put this information in terms acceptable to the DISSPLA program.

3.2. Sample Task

The following example illustrates one step of a project and the code necessary to plot this step. The Action ID for this example will be 0.0, the Description will be, "Define Specifications for Project A," and the Responsible Official is DTNSRDC. This action started in October of 1982 and will finish in March of 1984. The estimated initiation date was September 1982, the actual initiation date was October 1982. It is estimated that this task will be completed in March of 1984. At this point the actual completion date is not known.

To plot this example, the following DISSPLA and FORTRAN commands are needed.

```
CALL YCOORD(2,OFFSET,1,ANS)
CALL MESSAG("0.0$",100,XBLK1,ANS)
CALL MESSAG("DTNSRDC$",100,XBLK2,ANS)
CALL MESSAG("DEFINE SPECIFICATIONS FOR PROJECT A$",100,XBLK3,ANS)
CALL VECTPL(3HOCT,2,SP1)
CALL VECTPL(3HMAR,3,SP2)
CALL MARKPL(3HMAR,3,SP3)
ANS2 = ANS + 0.1
CALL VECTOR(SP1,ANS2,ANS,0)
CALL VECTOR(SP1,ANS,SP2,ANS,0)
CALL MARKER(5)
CALL CURVPL(ANS,YVALU)
CALL CURVE(SP3,YVALU,1,1)
CALL MESSAG("SEP 82$",100,XBLK9,ANS)
CALL MESSAG("OCT 82$",100,XBLK10,ANS)
CALL MESSAG("MAR 84$",100,XBLK11,ANS)
```

These commands will be briefly described. For a more specific description of how the variables are used see Section 3.3, how the DISSPLA functions are used see Section 5, and how the FORTRAN subroutines are used see Section 3.4.

The first call to YCOORD is a FORTRAN subroutine call. The code for this subroutine is given in Appendix C. This subroutine obtains the Y-coordinate

of the second line from the beginning of the chart and stores the result in the variable "ANS." This variable will be used to plot other information on that same line.

The next three calls to MESSAG plot the "ACTION ID" (0.0), the "RESPONSIBLE OFFICIAL" (DTNSRDC), and the "DESCRIPTION" (Define Specifications for Project A) in different columns, on the same line by using different X-coordinates for the different calls to MESSAG. The X-coordinates are stored in the XBLK variables. "XBLK1" is the X-coordinate for the first column of the chart, "XBLK2" for the second column, and so on. The values for these variables can be obtained from Table 1.

The calls to VECTPL are used to obtain the X-coordinates for the start date (in this case October 1982) and the finish date (March 1984). X-coordinate of the starting point of the vector is stored in "SP1"; value of the finish date is stored in "SP2."

MARKPL is similar to VECTPL in that, given a month and a year (3HMAR, 3-March in the third fiscal year), it returns the X-coordinate of the position of the marker or termination symbol. This value is stored in "SP3."

There are two calls to VECTOR; the first plots a vertical vector and the second, a horizontal vector. The vertical vector is used to show the start date, and the horizontal vector shows the duration of that task.

The call to MARKER allows the user to select the type of termination symbol that will be used to show the finish date. The user is advised to use "Marker(5)" if the task is incomplete or "MARKER(9)" if it is completed. The call to CURVPL will get the Y-coordinate of the marker, and the call to CURVE plots the marker.

The last three calls to MESSAG are similar to the earlier calls except for their X-coordinates, which again, are varied by using the XBLK variables. When no date exists, the column is left blank, and there is no call to MESSAG. The plot for this example is shown in Figure 3.

3.3. Predefined Variables

As an aid to the user, some variables have been predefined. These variables represent frequently used coordinates, which are initialized at the beginning of the program, or values returned from subroutines and frequently used in DISSPLA subroutines. They are described in Table 1.

PLAN OF ACTION AND MILESTONES

ACTION ID	DESCRIPTION	RESPONSIBLE OFFICIAL	FY 82	FY 83	FY 84	FY 85	FY 86	INITIATION DATES		COMPLETION DATES	
			ON GOING ACTIONS	ESTIMATED	ACTUAL	ESTIMATED	ACTUAL				
0.3	Define Specifications for Project A	OTNSRDC						SEP 82	OCT 82	NOV 84	

Figure 3 - Plot for Sample Task

3.4. FORTRAN Subroutines

The subroutines in this section calculate X- and Y-coordinates. To obtain the Y-coordinate for a line, call subroutine YCOORD. To obtain the X-coordinate of a specified month and year, call subroutines MARKPL and VECTPL. To convert a Y-coordinate from inches to user-defined ticks, call subroutine CURVPL. The following sections describe these subroutines and their use.

3.4.1. Subroutine VECTPL

Subroutine VECTPL returns the X-coordinates of the endpoints of the vector. In the call to VECTPL the user defines the month and the fiscal year of the endpoint. Since there are two endpoints to a vector, there are two calls to VECTPL:

```
CALL VECTPL (3HJUL,2,SP1)
CALL VECTPL (3HMAR,5,SP2)
```

where

3HJUL is the month of July. The user gives only the first three letters of the month (The 3H in front of JUL is CDC FORTRAN's way of defining characters).

2 is the second fiscal year. Since the user defines the range of fiscal years (e.g., 82-86 or 90-94), only the number 1, 2, 3, 4, or 5 are used by the subroutine.

SP1 is the value of the X-coordinate which represents the left endpoint of the vector. SP2 represents the right endpoint of the vector. Both values are given in inches.

3.4.2 Subroutine MARKPL

This subroutine returns the X-coordinate used to plot the marker. The calling arguments are the same as those of subroutine VECTOR. The only difference between them is that the returned X-coordinate value for VECTPL is in inches, whereas the returned X-coordinate here is in user-defined ticks (see Section 5.9). An example of a call to MARKPL is

CALL MARKPL (3HOCT,4,SP3)

where

3HOCT is the month which the marker will point to. In
this case it is October.

4 is the fiscal year which the marker will point to.

SP3 is the X-coordinate in user-defined ticks.

3.4.3 Subroutine YCOORD

This subroutine returns the Y-coordinate used to plot an entire line of information. An example of the calling arguments to YCOORD are:

TABLE 1 - Table of Variables

VARIABLE	DESCRIPTION
* <u>XBLK1</u>	The X-Coordinate for the first Vertical Bar or block. It is used to write messages underneath the "ACTION-ID" heading of the chart. In the POAM generated by this report XBLK1 = 0.21
* <u>XBLK2</u>	The X-Coordinate for the second vertical bar or block. It is used to write messages underneath the "Description" heading of the chart. In the POAM generated by this report XBLK2 = 0.80.
* <u>XBLK3</u>	The X-Coordinate for the third Vertical Bar. It is used to write messages underneath the "Responsible Official" heading of the chart. In the POAM generated by this report XBLK3 = 5.2.
* <u>XBLK9</u>	The X-Coordinate for the ninth Vertical Bar. It is used to write messages underneath the "Estimated Initiation Date" heading of the chart. In the POAM generated by this report XBLK9 = 12.37.
* <u>XBLK10</u>	The X-Coordinate for the tenth Vertical Bar. It is used to write messages underneath the "Actual Initiation Date" heading of the chart. In the POAM generated by this report XBLK10=13.375.
* <u>XBLK11</u>	The X-Coordinate for the eleventh Vertical Bar. It is used to write messages underneath the "Estimated Completion Date" heading of the chart. In the POAM generated by this report XBLK11=14.37.
* <u>XBLK12</u>	The X-Coordinate for the twelfth Vertical Bar. It is used to write messages underneath the "Actual Completion Date" heading of the chart. In the POAM generated by this report XBLK12=15.375.
ANS	The Y-Coordinate of a line that is returned from a call to Subroutine YCOORD. (See Section 3.4.3.)
ANS2	Used to plot a vertical vector of length 0.1 inches. This vertical vector is used to show the initiation month.
ANS3	A Y-Coordinate that is offset 0.05 inches from the current Y-Cordinate of the line (i.e. ANS). It is used to underline a Milestone heading.

TABLE 1 - Table of Variables (Cont'd)

VARIABLE	DESCRIPTION
SP1	A X-Coordinate of a given month and Fiscal Year. Its value is obtained through a call to VECTPL. It is the X-Coordinate of the leftmost endpoint of the vector.
SP2	A X-Coordinate of a given month and Fiscal Year. Its value is obtained through a call to VECTPL. It is the X-Coordinate of the rightmost endpoint of the vector.
SP3	The X-Coordinate used to plot the marker. It is returned from a call to MARKPL and its value is in user defined ticks (see Section 3.4.2. of this manual) as opposed to inches. *See Figure 9 for all of these.

CALL YCOORD (3, OFFSET,0,ANS)

where:

3 is the line number (from the top of the chart), of the line that the user is currently plotting. This number must be an integer and must fall between 1 and 43.

OFFSET is used to offset lines immediately following a Milestone heading so that some space (one line) is left between them.

0 Zero indicates that this line is not a Milestone heading. A 1 indicates that this line is a Milestone heading and the subroutine will offset the line (move it down a line on the chart) immediately following it.

ANS is the returned value of the Y-coordinate of the line specified. It is used to plot messages and vectors (but not markers) on that single line.

Another function of ANS is to underline Milestone Headings. This is accomplished by decreasing its value by 0.05. This value is stored in ANS3 as shown below:

ANS3 = ANS - 0.05

After this a call to

VECTOR (0.80,ANS3,4.95,ANS3,0)

will underline the heading (see Section 5.3.1). ANS is also used to offset the left endpoint of a vector so that the endpoint becomes a vertical vector.

This offset is stored in ANS2 as shown below:

ANS2 = ANS + 0.1

Vertical

Vector ! _____ ^

3.4.4. Subroutine CURVPL

This subroutine returns the Y-coordinate used to plot the marker. The returned value is in units of user-defined ticks because the call to subroutine CURVE must be in user-defined ticks and not in inches. This subroutine takes the Y-coordinate (in inches) of the current line and converts inches to user-defined ticks. See Section 5.5 for an explanation of user-defined ticks. An example of a call to CURVPL is:

```
CALL CURVPL (ANS, YVALU)
```

where:

ANS - is the current Y-coordinate in inches. This value is obtained by calling subroutine YCOORD.

YVALU - is the returned Y-coordinate in user-defined ticks.

3.5. Using DISSPLA Subroutines

Use of the subroutines discussed in Section 3.4 makes the following DISSPLA subroutines easier to implement.

After a call to subroutine YCOORD, the returned value, ANS, can be used in any call to MESSAG on that entire line. For example,

```
CALL MESSAG ("1.2$", 100, XBLK1,ANS)
```

```
CALL MESSAG ("(DTNSRDC)$", 100, XBLK3,ANS)
```

Now the user need only decide which message to plot and under which heading it belongs (which XBLK variable to use).

After a call to subroutine VECTPL, the returned values SP1 and SP2 can be used to plot the vector as follows:

```
ANS2 = ANS + 0.1 (See Section 5.3.1)
```

```
CALL VECTOR (SP1,ANS2,SP1,ANS,0)
```

```
CALL VECTOR (SP1,ANS,SP2,ANS,0)
```

Here, the first call to vector is used to obtain the left endpoint which is a vertical vector. The second call to vector is used to connect the vertical vector with the right endpoint. The vector now looks like this:

! _____

A call to curve will plot the marker on the right endpoint of the vector so that is is complete. An example of such a call is:

```
CALL CURVE (SP3,YVALU,1,1)
```

SP3 is obtained from a previous call to MARKPL, and YVALU is obtained from a previous call to CURVPL. The final vector now looks like this:

! _____
By using the subroutines described in Section 3.4 the user can avoid manually computing many of the values needed to call the DISSPLA subroutines. For a program listing of such calls see Appendix F.

3.6. Plot Generation.

In order to generate a plot similar to the one in Figure 1, the following procedure must be used:

- a. Set up the program as in Figure 2. In the control card section make sure that a PLFILE (Plot file) is requested. A PLFILE contains the machine language, of the information specified in the program and enables a plotting device to produce a plot from the catalogued PLFILE.
- b. Run the program in a batch mode since a large amount of core is required in use of DISSPLA routines.
- c. After the program runs, check the day file to see that there are no errors and that a PLFILE has been created and cataloged.
- d. For Tektronix plotting devices:

After the tektronix machine has been initialized and you are logged in, interactively attach the PLFILE that appears in the day file, as follows:

ATTACH, PLFILE, ID=C_____, CY=____

Then

ATTACH, DISPOST

LIBRARY, DISPOST

TEK 300

The computer should respond with

*****TEKTRONIX POST-PROCESSOR*****

ENTER DIRECTIVES

Press the space bar several times and return. The screen will flash and then be blank. After several seconds the plot will be drawn line by line and letter by letter on the screen.

When the plotting is finished a bell will sound and the cursor will appear in the top left corner. To get back into command mode type any character followed by a return. Do this twice. The computer will give the following response:

```
END POST-PROCESSOR  
END PLOT  
COMMAND-
```

Type this command only when plotting has stopped and you have received a hard copy if desired.

e. For CALCOMP plotting devices.

The following file must be created as a file or deck and a 7- or 9-track tape must be provided for the computer.

```
(User Initials) , CM200000, P4, MT 1.  
CHARGE, (Initials), (Access Number).  
VSN,TAPE10=SLOT##=TAPENAME.  
REQUEST, TAPE10, HY, RING.  
ATTACH, PLFILE, ID=(Initials),CY##.  
ATTACH, DISPOST.  
LIBRARY, DISPOST.  
POD936.  
EOR  
EOF
```

##numbers from your program.

Batch the above program and submit a request to the computer operator for a CALCOMP plot to be produced from the tape designated in the statement:

VSN,TAPE10=SLOT##=TAPENAME.

An example of the complete program used in Figure 2 and the Tektronix plot generated through the PLFILE produced by the program is given in Appendix B.

4.0. USER-DEFINED POAMS

The input parameters for generating any POAM chart are of three different types, those that generate (1) a blank chart, (2) the statements that appear on the chart, and (3) the markers and vectors that appear on that chart.

4.1. The Blank Chart.

Before beginning the actual programming, the blank chart must be designed by hand on graph paper indicating the actual lengths, heights, columns, and

rows desired. Label the page dimensions, the physical origin, and the X,Y location of the lines of the chart as in Figure 4.

The columns on the chart are created by using the VBARS routine of DISSPLA. This routine draws vertical bars. The input parameters needed to use VBARS are the bar width, bar height (designated by the Y coordinates of the bars end points), and the location of the midpoint of the bar. The following VBARS command is an example:

```
CALL VBARS (X(1), (Y1(1), Y2(1),1)
X(1) - This array (line 1 of the chart file Appendix D, page D-1)
       contains the X-coordinate of the bar midpoint on the X-axis.
Y1(1) - This array (line 4 of the chart file) contains the Y-coordinate
       of an end point of the bar.
Y2(1) - This array (line 5 of the chart file) contains the Y-coordinate
       of the other end point of the bar.
1      - the number of bars
```

The width of the bars is set through the routine BARWID (lines 17, 20, 22 and 24 of the chart file). The width of the bars will remain the same until another call to BARWID is made.

The routine HBARS (line 32 of chart file) creates horizontal bars. The input parameters needed to use this routine are the bar length, the bar width and the bar midpoint.

```
CALL HBARS (X1(1), X2(1), Y3(1), 1)
X1(1) - This array (line 2 of the chart file) contains the X-coordinate
       of an endpoint of the bar.
X2(1) - This array (line 6 of the chart file) contains the Y-coordinate
       of the other endpoint of the bar.
Y3(1) - This array (line 6 of the chart file) contains the Y-coordinate
       of the bar's midpoint along the Y axis.
1      - the number of bars to be produced.
```

The routine BARPAT (line 14 of chart file) controls the type of shading patterns that appear on the bar. In the chart file the bars required no shading pattern thus IPAT = 0 in the BARPAT (IPAT) routine.

The routine CLUSTR (line 18 of the chart file) controls the gaps between sets of bars. The arguments of the CLUSTR routine are

NCLSTR - Number of sets of bars to be clustered. This number is equivalent to the number of calls to VBARS or HBARS.

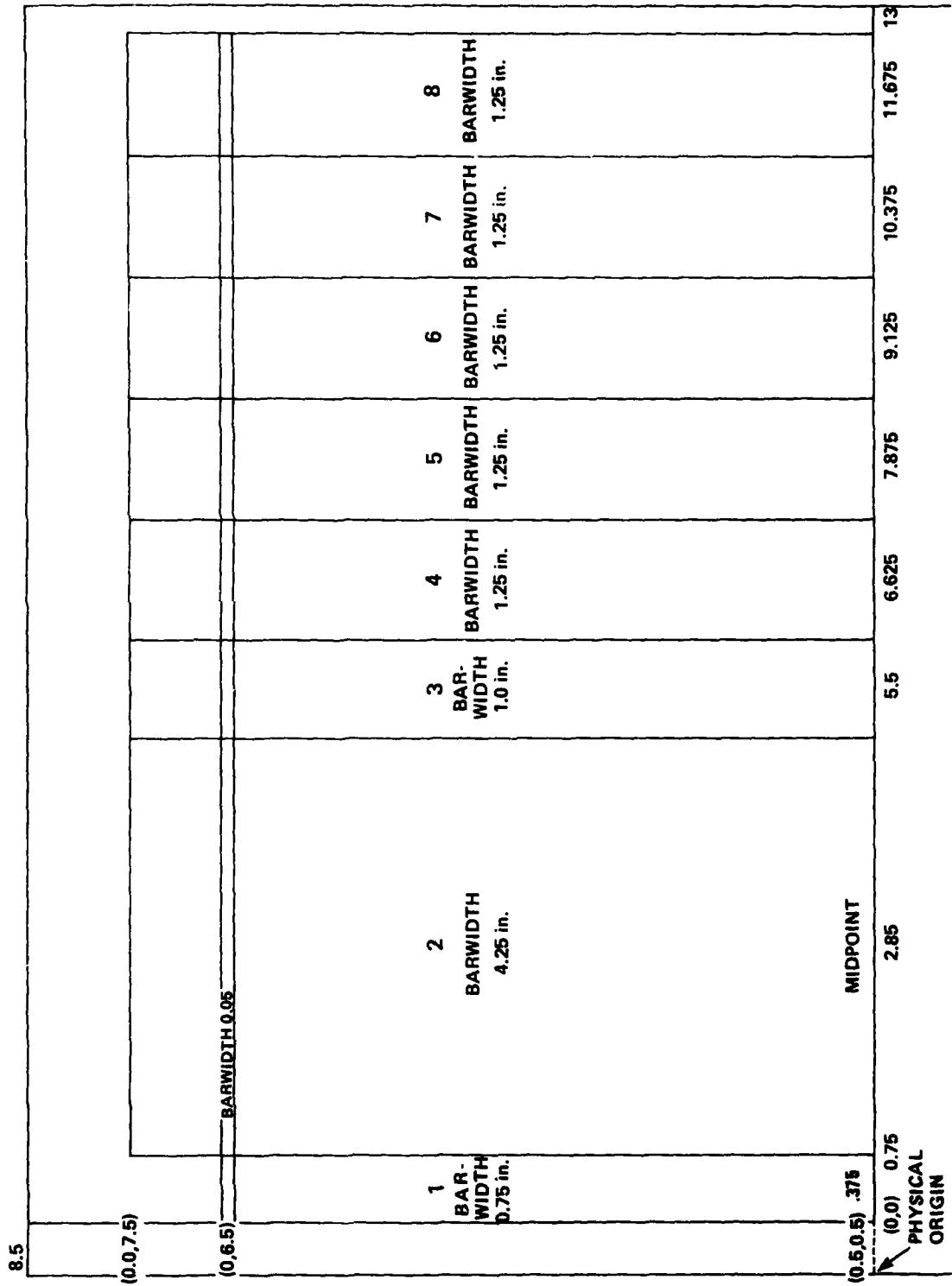


Figure 4 - Manually Produced Blank Chart

GAP - The amount of gap, in inches, between individual bars within the cluster.

The user is advised to examine CLUSTR carefully before using it, for numerous errors have been encountered. The program does complete its task but ignores the CALL to CLUSTR in the process. The errors exist in the internal workings of the routine as it exists on the computer system at this site. Figure 5 is an example of what can happen if CLUSTR is incorrectly used.

The Tektronix and CALCOMP plots accompany the chart file in Appendix E.

4.2. Chart Statements.

The input parameters required to generate statements include the letter height, the letter type, and the X-Y position of the lower left corner of the first letter of the statement. These parameters are discussed in the next section, and examples can be found in Appendix F (The POAM2 file).

5.0. DISSPLA.

The Computer Generated POAM Program (CGPP) uses two types of code. The FORTRAN code is used to define variables and arrays, and to perform any other functions that can not be done in DISSPLA. The DISSPLA code is used to define and generate all the graphics used in the POAM.

DISSPLA is a series of 220 routines, each performing particular plotting tasks. Each routine may be used only at designated levels of DISSPLA.

5.1. Levels of DISSPLA.

In Figure 6, the 38th card calls the first DISSPLA routine used (CALL COMPRS). At this point the status of DISSPLA is level 0. In this level the Common blocks of DISSPLA are blank and device initializing occurs.

The device initializing routine BGNPL, card 39, causes DISSPLA to enter level 1, the "ready to start Plot" status of DISSPLA. In level 1 the page dimensions, physical origin, axis lengths, and step sizes are not yet defined. CARDS 40 and 41 define the page dimensions and physical origin.

The routine TITLE (card 43) causes DISSPLA to enter level 2. In this status the page dimensions, axis lengths and physical origin have been defined. Without cards 40 and 41 all these variables would still be set using standard values by the system. For instance, the page size would be 11" x 8 1/2" at its standard values.

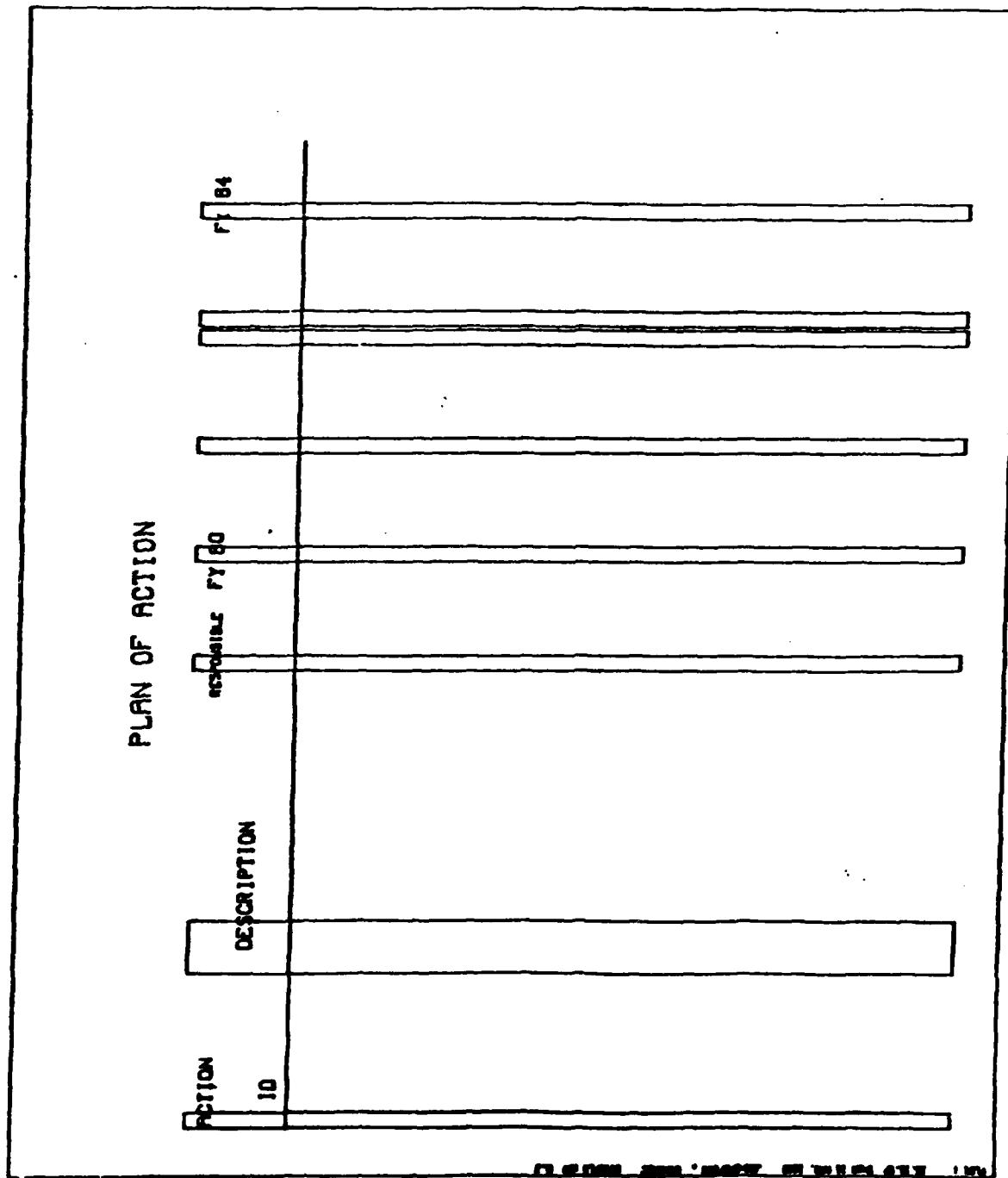


Figure 5 - Bar Chart Containing Extensive CLUSTR Imperfections

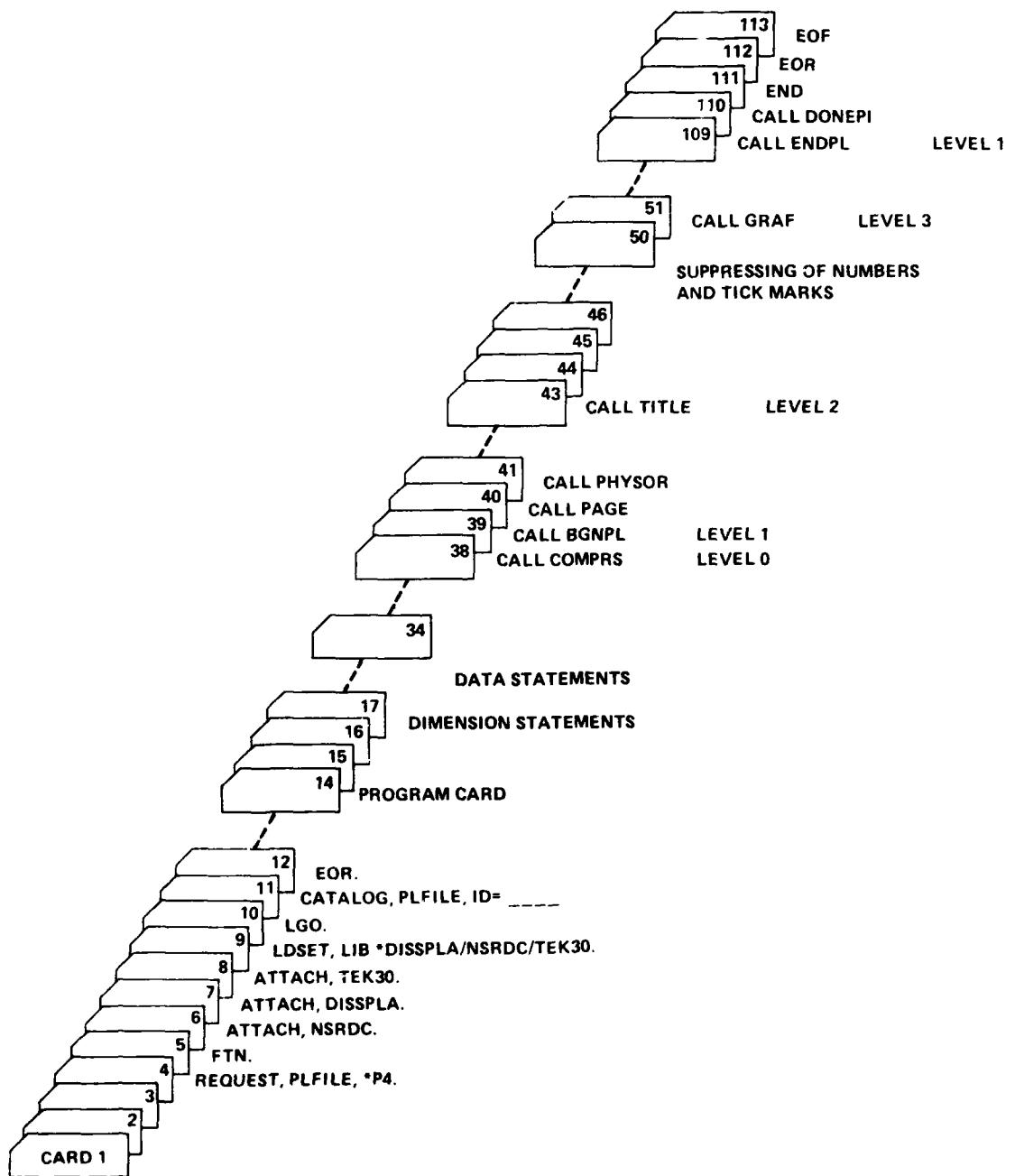


Figure 6 - Example Program Deck
(Showing levels of DISSPLA)

Cards 44 through 50 designate or suppress numbers or tick marks that will appear on the X- and Y-axes.

The routine GRAF (card 51) sets up the basic linear axis and causes DISSPLA to enter level 3. In this level the plot is fully determined, i.e., the parameters required in all previous levels have been supplied.

Cards 52 through 108 write statements, symbols and vectors of different lengths and heights in the plot area.

The routine ENDPL (card 109) terminates the current plot and brings DISSPLA back to level 1.

The routine DONEPL (card 110) is used when all plots have been complete. In this example there is only one plot, but if several plots had been generated, DONEPL would not be called until the final plot was complete. DONEPL terminates plot generation completely and closes the file in use.

5.2. DISSPLA Functions

In order to create a POAM the user should be familiar with several DISSPLA functions. These functions and their arguments are discussed in the following sections. Further information can be obtained from a DISSPLA reference manual.

5.2.1. MESSAG.

The routine MESSAG and similar routines like REALNO, RLINT, RLMESS, etc., control the writing of statements and numbers in the plot area. The MESSAG routine places the left-hand corner of the first character in the statement at a particular X-Y position on the plot. This routine and its arguments are as follows:

```
CALL MESSAG (LMESS, IMESS, XPOS, YPOS)
```

where

LMESS - character string (statement to be written)

IMESS - the number of characters in the string

XPOS - X coordinate of string position

YPOS - Y coordinate of string position

The use of 100 for the argument IMESS allows the character string to be plotted without the \$.

Example

CALL MESSAG ("ID\$", 3, D.O. 0.0)

produces

ID\$

whereas

CALL MESSAG ("ID\$", 100, 0.0, 0.0)

produces

ID

5.2.2. HEIGHT.

Letter heights are set at a standard of 0.14 inches. It is advisable to set letter heights at multiples of 0.07 inches for DISSPLA statements. Nevertheless, any letter size is acceptable through the use of the routine HEIGHT (lines 1, 4, 8, 11, 13 and 20 of Appendix F).

Letter sizes should be neither too small nor too large. The Tektronix produced charts in Figure 7 contain examples of letter heights (sizes). Since the Tektronix hard copier reduces the size of all the copies it produces, the CALCOMP plotter must be used to obtain hard copies of plots at their actual dimensions.

5.2.3. BASALF and MIXALF

DISSPLA provides seven standard alphabets and an option to construct one's own alphabet (see the DISSPLA manual). The standard alphabets available are

<u>TYPE of ALPHABET</u>	<u>DISSPLA ABBREVIATION</u>
Capital Roman	"STANDARD"
Lower Case Roman	"L/CSTD"
Capital Greek	"GREEK"
Lower Case Greek	"L/CGREEK"
Capital Russian	"RUSSIAN"
Lower Case Russian	"L/CRUSSN"
Hebrew	"HEBREW"

The routine that determines which of these alphabets will be used in the writing of statements and numbers on a plot is BASALF. In the following statement one of the DISSPLA abbreviations is substituted for LALPHA.

CALL BASALF (LALPHA)

LETTER SIZES (INCHES)

0.08	NSRDC
0.10	NSRDC
0.12	NSRDC
0.14	NSRDC
0.16	NSRDC
0.18	NSRDC
0.20	NSRDC
0.22	NSRDC
0.24	NSRDC
0.26	NSRDC
0.28	NSRDC
0.30	NSRDC

Figure 7 - Upper Case Letter Sizes

LETTER SIZES (INCHES)

0.35	NSRDC
0.40	NSRDC
0.45	NSRDC
0.50	NSRDC
0.55	NSRDC
0.65	NSRDC
0.75	NSRDC
	NSRDC
	NSRDC
	NSRDC

Every letter of the plotted statement will be written in the alphabet designated by LALPHA.

In order to plot a statement which contains a mixture of upper and lower case characters or a mixture of totally different alphabet types, the routine MIXALF must be used. The following statement will cause all of the characters enclosed in parentheses in the argument LMESS in MESSAGE to be written in the alphabet designated by LALPHA.

CALL MIXALF (LALPHA)

Examples of the use of these two routines are given in lines 1 through 115 of Appendix F, pages F-1 through F-5.

5.3. Chart Vectors and Markers.

The vectors and markers in the POAM are generated by two DISSPLA routines, VECTOR and MARKER. A brief description of how each is used in the CGPP is given.

5.3.1. Vectors.

The VECTOR routine draws a line from (XFROM, YFROM) to (XTO, YTO) and places shaded or unshaded arrows of different shapes and sizes at one, both or neither of the endpoints of the line. The arguments of the VECTOR routine are:

CALL VECTOR (XFROM, YFROM, XTO, YTO, IVEC)

XFROM - X value of the originating point

YFROM - Y value of the originating point

XTO - X value of the point pointed to

YTO - Y value of the point pointed to

IVEC - an integer describing the vector type and size of the arrowheads (see Figure 8).

Example

CALL VECTOR (0.0, 0.0, 1.0, 1.0, 11)

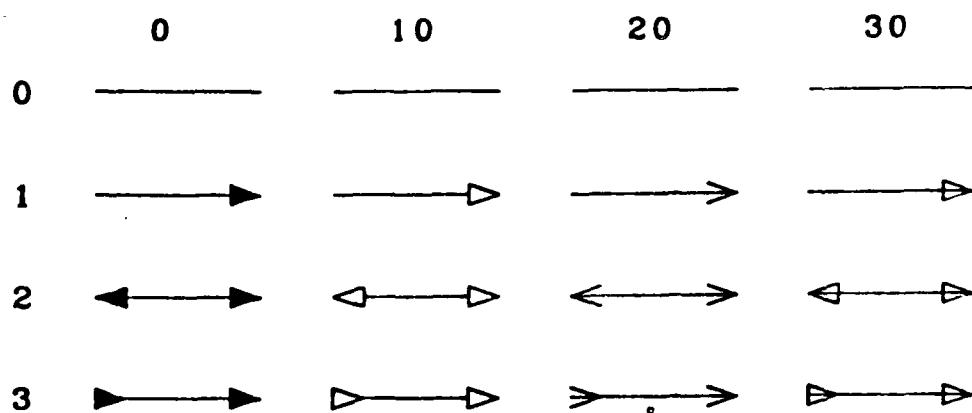
! _____ ^

(0.0, 0.0) (1.0, 1.0)

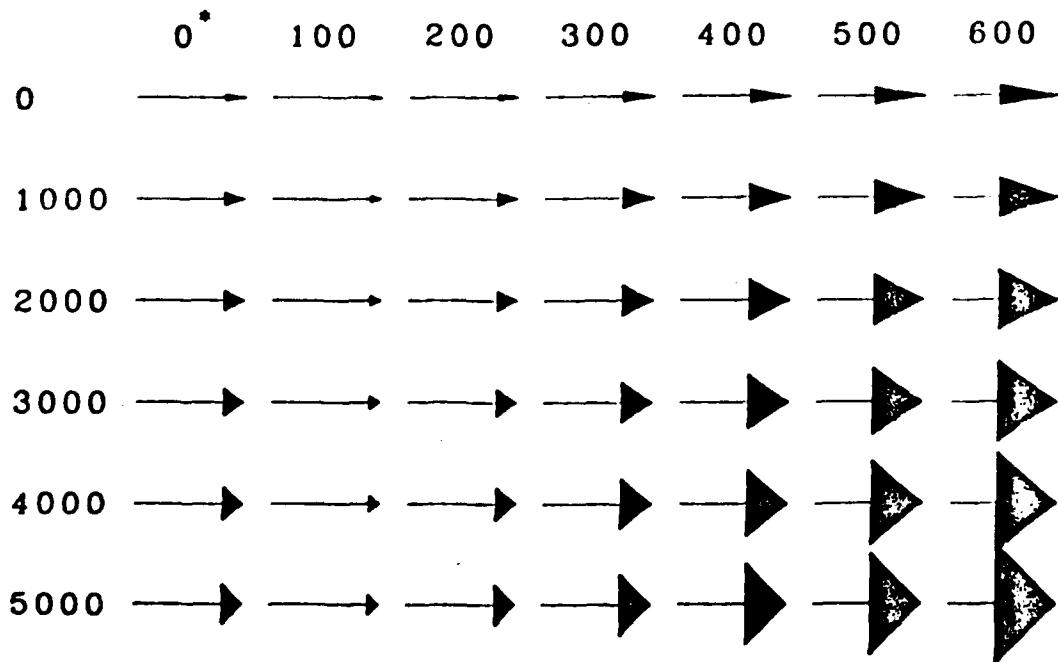
The VECTOR routine is used in Appendix F, pages F-3, lines 7 and 8, to underline statements and to draw horizontal and vertical lines in the fiscal year-month sections of the POAM1 chart in Appendix F.

In order to use VECTOR effectively the exact X-Y coordinates of the locations in which VECTOR is to be used must be known. In the fiscal year-

Vector Types



Vector Tip Sizes



* 0 means tip length equals letter height

Figure 8 - Vector types

month section of the POAM1 chart, the location of each month (ONDJFMAMJJAS) in each fiscal year (FY--) must be provided. Since it is difficult and time consuming to figure out these locations each time the need for VECTOR is encountered, the LOCATION OF MONTHS CHART was constructed (see Figure 9). Ordinarily the Y location in VECTOR is already known, so the chart in Figure 9 contains only the X coordinates.

5.3.2. MARKERS.

In order to use the routine MARKER the routine CURVE must also be used. The MARKER routine only selects, from a group of fifteen types of characters, the characters to be used by CURVE and the sequence in which these characters are to be used. The CURVE routine contains the following arguments:

CALL CURVE. (XARAY, YARAY, NPNTS, IMARK)

XARAY - array holding X values (lines 1 through 5 of Appendix F, page F-2)

YARAY - array holding Y values (line 6 of Appendix F, page F2-1)

NPNTS - number of points to be plotted from these arrays

IMARK - determines whether or not the plotted points will be connected

MARKER is designed to mark the points plotted by CURVE. The POAM1 chart uses the MARKER and CURVE routines to indicate the completion dates of the Milestones in the following manner:

FY81	FY82
ONDJFMAMJJAS	ONDJFMAMJJAS
! _____	^
Initiation date	Completion date
FEB 81	FEB 82

As in the case of the year-month locations required to use the VECTOR routine efficiently, a chart has been constructed to give the locations for MONTH MARKERS, (see Figure 10). The marker must be centered under the month to which it belongs, so that the X,Y location of a month is not the same as the X,Y location of the marker that designates it.

5.4. CURVE vs MESSAG.

It has been observed that a point measured from the Physical Origin by the DISSPLA routine CURVE is not the same point when used in the DISSPLA routine MESSAG. In the example (Figure 11) the same points were used in the CURVE and MESSAG routines as follows:

MONTH	X POSITIONS					FY 85
	FY 81	FY 82	FY 83	FY 84		
1 OCT	6.12	7.37	8.62	9.87	11.12	
2 NOV	6.22	7.47	8.72	9.97	11.22	
3 DEC	6.32	7.57	8.82	10.07	11.32	
4 JAN	6.42	7.67	8.92	10.17	11.42	
5 FEB	6.52	7.77	9.02	10.27	11.52	
6 MAR	6.62	7.87	9.12	10.37	11.62	
7 APR	6.72	7.97	9.22	10.47	11.72	
8 MAY	6.82	8.07	9.32	10.57	11.82	
9 JUN	6.92	8.17	9.42	10.67	11.92	
10 JUL	7.02	8.27	9.52	10.77	12.02	
11 AUG	7.12	8.37	9.62	10.87	12.12	
12 SEP	7.22	8.47	9.72	10.97	12.22	

Figure 9 - Location of Months Chart

LOCATION OF MONTH MARKERS

MONTH	X POSITIONS											
	FY 81	FY 82	FY 83	FY 84	FY 85	FY 86	FY 87	FY 88	FY 89	FY 90	FY 91	
1 OCT	6.11	7.36	8.61	9.88	11.11	11.21						
2 NOV	6.21	7.46	8.71	9.96								
3 DEC	6.31	7.58	8.81	10.06								
4 JAN	6.41	7.68	8.91	10.16								
5 FEB	6.51	7.76	9.01	10.26								
6 MAR	6.61	7.86	9.11	10.36								
7 APR	6.69	7.94	9.19	10.44								
8 MAY	6.79	8.04	9.28	10.54								
9 JUN	6.89	8.14	9.39	10.64								
10 JUL	6.99	8.24	9.49	10.74								
11 AUG	7.09	8.34	9.59	10.84								
12 SEP	7.19	8.44	9.69	10.94								

Figure 10 - Location of Month Markers Chart

PLAN OF ACTION AND MILESTONES

0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
MILESTONE 1 - 01/25/04 RELEASE OF SOURCE CODE INITIATION		RELEASE								

Figure 11 – CURVE versus MESSAG Example Program

Example 1

```
CALL MESSAG ("0.0$", 100, 0.20, 5.96)
CALL MARKER (2)
CALL CURVE (XA1(4), YA1(1), 1, 1)
```

These statements correspond to lines 7, 12, and 13 of the program listing
YA1(1) equals 5.96 from line 6 of Figure 12.

Example 2

```
CALL MESSAG ("0.1$," 100, 0.20, 3.03)
CALL MARKER (2)
ALL CURVE (XA2(12), YA1 (15), 1, 1)
```

These statement correspond to lines 11, 14 and 17 of the program listing given
in Figure 12. XA2(12) equals 8.44 from line 2, Figure 12, and YA1(15) is
equal to 3.03 from line 6, Figure 12.

If Examples 1 and 2 are compared with the plot in Figure 11, then from
these these example statements there must be differences in the units of
measure (scaling is occurring) used in the Y-coordinates of the CURVE and
MESSAG statements. The user needs to be aware of these differences in order
to obtain the desired results.

5.5. User-Defined Ticks (Scaling)

To set up the POAM chart the user calls two DISSPLA subroutines:

```
CALL TITLE (LTITLE, ITITLE, LXNAME, IXNAME, LYNAME, IYNAME, XAXIS, YAXIS)
```

(See the DISSPLA pocket manual page 8 for a description of these arguments)

```
CALL GRAF (XORIG, XSTEP, XMAX, YORIG, YSTEP, YMAX)
```

(See the DISSPLA pocket manual page 11 for a description of these arguments)

When a call to GRAPH is made rather than a call to GRAF, all the units
used in plotting are in inches. When GRAF is used, however, the user defines
his/ her own ticks for the X and Y axes. To determine how many inches per user
defined tick, the following formula is provided.

User-defined X-AXIS ticks = XAXIS / (XMAX - XORIG)

User-defined Y-AXIS ticks = YAXIS / (YMAX - YORIG)

In the subroutine CURVPL the YAXIS ticks are 1.11 inches long. This is
because the YMAX = 9.0, YAXIS = 10.0 and the YORIG = 0. If the user changes
either of these he should also change the 1.11 inches in Subroutine CURVPL to
the correct user-defined YAXIS tick length.

```

**** CM200000, P4.
CHARGE,++++,*****.
C ***** IS THE USER'S CHARGE NUMBER
C ++++ IS THE USER'S INITIALS
REQUEST,PLFILE,XPF.
FTN.

ATTACH,NSRDC.
ATTACH,DISSPLA.
ATTACH,TEK30.
LDSET,LIB-DISSPLA/NSRDC/TEK30.
LGO.
CATALOG,PLFILE,1D-+++.
EOR

C-----MAIN PROGRAM-----
PROGRAM PAM12(INPUT,OUTPUT,TAPE10,TAPES-INPUT,TAPE6-OUTPUT)
DIMENSION X(12),X1(1),X2(1),X3(1),X4(1),Y1(1),Y2(1),Y3(1),Y4(1)
DIMENSION XA1(12),XA2(12),XA3(12),XA4(12),XA5(12),YA1(17),YA5(1)
DATA X/0.375,2.85,5.5,6.625,7.875,9.125,10.375,11.625,12.75,13.75,
K14,75,15,75/
DATA X1/0.0/
DATA X2/12.25/
DATA Y1/0.0/
DATA Y2/7.2/
DATA Y3/6.5/
DATA XA1/6.11,6.21,6.31,6.41,6.51,6.61,6.69,6.79,6.89,6.99,7.09,7.
K19/
(2) DATA XA2/7.36,7.46,7.56,7.66,7.76,7.86,7.94,8.04,8.14,8.24,8.34,8.
K44/
(3) DATA XA3/8.61,8.71,8.81,8.9; 9.01,9.11,9.19,9.29,9.39,9.49,9.59,9.
K69/

```

Figure 12 - CTRV1 VERSUS MESSAGE Example Program

```
C  
C          CHART      HEADINGS  
C  
CALL UBARS(X(5),Y(1),Y2(1),1)  
CALL UBARS(X(6),Y(1),Y2(1),1)  
CALL UBARS(X(7),Y(1),Y2(1),1)  
CALL UBARS(X(8),Y(1),Y2(1),1)  
CALL BARWID(0.05)  
CALL THICRU(5)  
CALL HBARS(X1(1),X2(1),Y3(1),1)
```

```
CALL BASALF('L/CSTD')  
CALL MIXALF('STANDARD')  
CALL HEIGHT(0.095)  
1)   CALL MESSAGE('0.08',100,0.20,5.96)  
2)   CALL MESSAGE('MILESTONE 0 - MISSION ANALYSIS/PROJECT INITIATION')  
3)   CALL VECTOR(0.89,5.91,4.95,5.91,0)  
4)   CALL MESSAGE('0.15',100,0.20,3.03)  
5)   CALL MESSAGE('DEVELOPMENT OF (MENS)',100,0.80,3.03)  
6)   CALL MESSAGE('NPSSMO',100,5.2,3.03)  
7)   CALL MARKER(2)  
8)   CALL CURVE(XA1(4),YA1(1),1,1)  
9)   CALL MARKER(2)  
10)  CALL CURVE(XA1(12),YA1(15),1,1)  
11)  CALL ENDPL(0)  
12)  CALL DONEPL  
END  
<BOTTOM OF FILE>
```

```

(4) DATA XA4/9.86,9.96,10.06,10.16,10.26,10.36,10.44,10.54,10.64,10.74
     K,10.84,10.94/
(5) DATA XA5/11.11,11.21,11.31,11.41,11.51,11.61,11.69,11.79,11.89,11.
     K99,12.09,12.19/
(6) DATA YA1/5.96,5.77,5.6,5.42,5.23,4.63,4.45,4.28,4.09,3.92,3.74,3.5
     K7,3.38,3.21,3.03,2.86,2.68/
C
C   CHART ASSEMBLY BEGINS HERE
C
        CALL COMPRS
        CALL BGNPL(1)
        CALL PAGE(13.0,11.0)
        CALL PHYSOR(0.5,0.5)
        CALL NOBRDR
        CALL TITLE(2H ,2,2H ,2,2H ,2,12.25,8.5)
        CALL HEIGHT(0.21)
        CALL MESSAGE('PLAN OF ACTION AND MILESTONES',100,3.5,8.4)
        CALL YNONUM
        CALL XNONUM
        CALL YTICKS(0)
        CALL XTICKS(0)
        CALL BARPAT(0)
        CALL GRAF(0.0,1.0,12.25,0.0,1.0,7.5)
        CALL THICRU(3)
        CALL BARUID(0.75)
        CALL CLUSTR(6.0,0.0)
        CALL UBARS(X(1),Y1(1),Y2(1),1)
        CALL BARUID(4.25)
        CALL UBARS(X(2),Y1(1),Y2(1),1)
        CALL BARUID(1.0)
        CALL UBARS(X(3),Y1(1),Y2(1),1)
        CALL BARUID(1.25)
        CALL UBARS(X(4),Y1(1),Y2(1),1)

```

5.6. ROUTINES

The routine chart contains the names, arguments, descriptions, levels and DISSPLA manual locations of all the routines used in the POAM chart program. The levels of DISSPLA, which were explained in Section 5.1 of this manual, are important in the correct use of any DISSPLA routine. Level i-j p/s or level i-j indicate the level(s) in which the routine may be used. Level i-j denotes that the option routine may be called while DISSPLA is in level i through j. The p/s denotes that the routine is of the parameter setting type.

6.0. CONCLUSION

The POAM chart can be used for many applications. The user has the choice of using the frame that spans 5 years, or of defining his own to suit a particular application. The predefined POAM chart is easier to use and eliminates the need to learn DISSPLA in detail. If the user needs to define his own POAM chart, the DISSPLA manual should be read before doing so.

TABLE 2 - TABLE OF DISSPLA ROUTINES

ROUTINE	ARGUMENTS	LEVEL	DESCRIPTION	MANUAL LOCATION
				SECTION PAGE
BARPAT	IPAT	1-3p/s	Specifies the shading pattern to be used on vertical and horizontal bars.	G2.1 G2-4
BARWID	BARWTH	1-3p/s	Specifies the width of a particular bar.	G2.2 G2-6
BASALF	LALPHA	1-3p/s	Specifies the type of alphabet to be used in the writing of words and statements. Example: Roman, Greek, Standard and their upper and lower cases.	B6.1 B6-1
BGNPL	IBEGIN	1	Initializing routine.	A4.1 A4-1
CLUSTR	NCLSTR, GAP	1-p/s	Specifies the gap width between the number of bars designated by NCLSTR for each call to VBAR on HBAR	G2.3 G2-8
COMPRS		0	Device initializing prior to the first call to a DISSPLA subroutine.	F3.1 F-1
CURVE	XARAY, YARAY NPNTS, IMARK	3	Plots points with specified markers and connects those plotted points with a line, if so specified.	A4.4 A4-7 B5.3 B5-2 B5.4 B5-2 B29 B29-1
DONEPL		1	Terminates plot generation and closes the file in use.	A1.1 A1-2 F3.1 F3-1
ENDPL	IPLOT	2-3p/s	Terminates the current plot, moves the pen back to it's original position, commences a new plot, and brings DISSPLA back to level one.	A4.5 A4-9

Table 2 (Continued)

ROUTINE	ARGUMENTS	LEVEL	DESCRIPTION	MANUAL LOCATION SECTION	PAGE
GRAF	XORIG, XSTP, XMAX, YORIG, YSTR, YMAX	2	This is a primary graph set up routine and it takes DISSPLA to level three.	A4.3	A4-4
*HBARS	X1ARAY, X2ARAY, YARAY, NPNTS	3	Generates horizontal bars.	G2	G2-2
*HEIGHT	HITE	1-3p/s	Sets the height of numbers and letter in inches. For best results use multiples of 0.07.	B5.2 E2.5 B5.4	B5-1 E2-6 B5-2
*MARKER	ISYM	1-3p/s	This subroutine selects from 15 symbols, the symbol or sequence of symbols used to mark the plot for all calls to curve.	B5.3	B5-2
*MESSAG	LMESS, IMESS XPOS, YPOS	1-3p/s	This subroutine will place the lower left hand corner of the first of the string (statement) LMESS, which contains IMESS (number of) characters, at the position XPOS, YPOS from the current physical origin.	B2.1 A5.2 B5.6 E20.1 B5.5 A5.2.2 A5.2.1 B5.1	B2-1 A5-2 B5-4 B20-1 B5-3 A5-4 A5-3 B5-1
MIXALF	LALPHA	1-3p/s	This subroutine will cause all the characters (letters) enclosed in parentheses in a statement to be written in the alphabet designated by LALPHA.	B6.2 B25.1	B6-2 B25-1
NOBRDR		1p/s	This will cause TITLE not to draw a page outline	B2.2	B2-1
PAGE	PAGEX, PAGEY	1p/s	This subroutine must be called before TITLE or AREA2D or it will be ignored. It's function is that of setting the length and height of the page on which your plot will appear.	B3.1 B19 B2.2	B3-1 B19-1 B2-1

Table 2(Continued)

ROUTINE	ARGUMENTS	LEVEL	DESCRIPTION	MANUAL LOCATION	
				SECTION	PAGE
PHYSOR	XPHYS, YPHYS	1p/s	This subroutine instructs TITLE to set the physical origin at a position (XPH4S, YPHYS) inches from the lower left hand corner of the page.	B18.1	B18-1
THICRV	ITHICK	1p/s	This subroutine will redraw a curve (ITHICK) number of times. Each time the curve is redrawn the new curve is slightly offset from the previous curve. Thus the curve appears darker and thicker.	B15.2	
TITLE	LTITLE, ITITLE, LXNAME, IXNAME, LYNAME, IYNAME, XAXIS, YAXIS	1	This subroutine takes DISSPLA to level two. It also defines axis lengths and axis labels.	A4.2	A4-1
VBARS	XARAY, Y1ARAY, Y2ARAY, NPNTS	3	This subroutine draws vertical bars.	G2	G2-2
VECTOR	XFROM, YFROM, XTO, YTO, IVEC	3	This subroutine draws vectors from (XFROM, YFROM) to (XTO, YTO) and places shade arrows of different shapes and size at either one or both ends of the vector.	B21.1	B21-1
XNONUM		1-3p/s	Suppresses all levels of X axis labelling or numbering.	B11.8.3 C5.1.4 G4.6	B11-16 C5-2 G4-13
XTICKS	ITICKS	1-2p/s	This subroutine is used to suppress or place as many tick marks as one desires along the X axis.	B11.3 B11.8.4 C5.1.2	B11-2 B11-6 C5-1
YNONUM		1-3p/s	Suppresses all levels of Y axis labeling or numbering.	B11.8.3 C5.1.4 G4.6	B11-16 C5-2 G4-13
YTICKS	ITICKS	1-2p/s	This subroutine is used to suppress or place as many tick marks as one desires along the Y axis.	B11.3 B11.8.4 C5.1.2	B11-2 B11-6 C5-1

APPENDIX A
SAMPLE PROGRAM FOR POAM FRAME

CAXX,CM200000,P4.
CHARGE,CAXX,XXXXXX99999.
REQUEST,PLFILE,*PF.
FTN.
ATTACH,NSRDC.
ATTACH,DISSPLA.
ATTACH,TEK30.
LDSET,LIB=DISSPLA/NSRDC/TEK30.

LGO.
CATALOG,PLFILE, ID=CAXX.
EOR

C-----MAIN PROGRAM-----

```
PROGRAM POAM(INPUT,OUTPUT,TAPE10,TAPE5=INPUT,TAPE6=OUTPUT)
DIMENSION X(8),X1(1),X2(1),Y1(1),Y2(1),Y3(1)
1. DATA X/0.375,2.85,5.5,6.625,7.875,9.125,10.375,11.625/
2. DATA X1/0.0/
3. DATA X2/12.25/
4. DATA Y1/0.0/
5. DATA Y2/7.5/
6. DATA Y3/6.5/
C
C   CHART ASSEMBLY BEGINS HERE
C
CALL COMPRS
CALL BGNPL(1)
7. CALL PAGE(13.0,11.0)
8. CALL PHYSOR(0.5,0.5)
9. CALL TITLE("PLAN OF ACTION$",100,2H ,2,2H ,2,12.25,8.5)
10.    CALL YNONUM
11.    CALL XNONUM
12.    CALL YTICKS(0)
13.    CALL XTICKS(0)
14.    CALL BARPAT(0)
15.    CALL GRAF(0.0,1.0,12.25,0.0,1.0,7.5)
16.        CALL THICRV(3)
17.        CALL BARWID(0.75)
18.        CALL CLUSTR(6.0,0.0)
19.    CALL VBARS(X(1),Y1(1),Y2(1),1)
20.        CALL BARWID(4.25)
21.    CALL VBARS(X(2),Y1(1),Y2(1),1)
22.        CALL BARWID(1.0)
23.    CALL VBARS(X(3),Y1(1),Y2(1),1)
24.        CALL BARWID(1.25)
25.    CALL VBARS(X(4),Y1(1),Y2(1),1)
26.    CALL VBARS(X(5),Y1(1),Y2(1),1)
27.    CALL VBARS(X(6),Y1(1),Y2(1),1)
28.    CALL VBARS(X(7),Y1(1),Y2(1),1)
29.    CALL VBARS(X(8),Y1(1),Y2(1),1)
30.        CALL BARWID(0.05)
31.        CALL THICRV(5)
32.    CALL HBARS(X1(1),X2(1),Y3(1),1)
```

C CHART HEADINGS
C

CALL ENDPL(1)
CALL DONEPL
END

APPENDIX B
SAMPLE PROGRAM AND PLOT OF A USER DEFINED FRAME

CAXX, QM200000, P4.
CHARGE, CAXX, XXXXX99999.
REQUEST, PLFILE, *PF.
FTN.
ATTACH, NSRDC.
ATTACH, DISSPLA.
ATTACH, TEK30.
LDSET, LIB=DISSPLA/NSRDC/TEK30.
LGO.
CATALOG, PLFILE, ID=CAXX.
EOR

C-----MAIN PROGRAM-----
PROGRAM PAM1(INPUT,OUTPUT,TAPE10,TAPE5=INPUT,TAPE6=OUTPUT)
DIMENSION X(8),X1(1),X2(1),Y1(1),Y2(1),Y3(1)
DIMENSION XA1(12),XA2(12),XA3(12),XA4(12),XA5(12),YA1(17)
DATA X /0.375,2.85,5.5,6.625,7.875,9.125,10.375,11.625/
DATA X1 /0.0/
DATA X2 /12.25/
DATA Y1 /0.0/
DATA Y2 /7.2/
DATA Y3 /6.5/
DATA XA1 /6.11,6.21,6.31,6.41,6.51,6.61,6.69,6.79,6.89,6.99,7.09,7.19/
DATA XA2 /7.36,7.46,7.56,7.66,7.76,7.86,7.94,8.04,8.14,8.24,8.34,8.44/
DATA XA3 /8.61,8.71,8.81,8.91,9.01,9.11,9.19,9.29,9.39,9.49,9.59,9.69/
DATA XA4 /9.86,9.96,10.06,10.16,10.26,10.36,10.44,10.54,10.64,10.74,
*10.84,10.94/
DATA XA5 /11.11,11.21,11.31,11.41,11.51,11.61,11.69,11.79,11.89,11.99,
*12.09,12.19/
DATA YA1 /5.96,5.77,5.6,5.42,5.23,4.63,4.45,4.28,4.09,3.92,3.74,3.57,
*3.38,3.21,3.03,2.86,2.68/

C
C CHART ASSEMBLY BEGINS HERE
C
CALL COMPRS
CALL BGNPL(1)
CALL PAGE(13.0,11.0)
CALL PHYSOR(0.5,0.5)
CALL NOBRDR
CALL TITLE(2H ,2,2H ,2,2H ,2,12.25,8.5)
CALL HEIGHT(0.21)
CALL MESSAG("PLAN OF ACTION AND MILESTONES\$",100,3.5,8.4)
CALL YNONUM
CALL XNONUM
CALL YTICKS(0)
CALL XTICKS(0)
CALL BARPAT(0)
CALL GRAF(0.0,1.0,12.25,0.0,1.0,7.5)
CALL THICRV(3)
CALL BARWID(0.75)
CALL CLUSTR(6.0,0.0)
CALL VBARS(X(1),Y1(1),Y2(1),1)
CALL BARWID(4.25)

```
CALL VBARS(X(2),Y1(1),Y2(1),1)
    CALL BARWID(1.0)
CALL VBARS(X(3),Y1(1),Y2(1),1)
    CALL BARWID(1.25)
CALL VBARS(X(4),Y1(1),Y2(1),1)
CALL VBARS(X(5),Y1(1),Y2(1),1)
CALL VBARS(X(6),Y1(1),Y2(1),1)
CALL VBARS(X(7),Y1(1),Y2(1),1)
CALL VBARS(X(8),Y1(1),Y2(1),1)
    CALL BARWID(0.05)
    CALL THICRV(5)
CALL HBARS(X1(1),X2(1),Y3(1),1)
```

C
C
C

```
CHART HEADINGS
CALL HEIGHT(0.10)
CALL BASALF("STANDARD")
CALL MESSAG("ACTION$",100,0.08,.7.8)
    CALL HEIGHT(0.10)
CALL MESSAG("RESPONSIBLE$",100,5.05,7.8)
    CALL HEIGHT(0.15)
CALL MESSAG("FY 81$",100,6.3,7.8)
CALL MESSAG("FY 82$",100,7.6,7.8)
CALL MESSAG("FY 83$",100,8.9,7.8)
CALL MESSAG("FY 84$",100,10.1,7.8)
CALL MESSAG("FY 85$",100,11.4,7.8)
CALL MESSAG("DESCRIPTION$",100,2.0,7.5)
CALL MESSAG("ID$",100,0.20,7.5)
    CALL HEIGHT(0.10)
CALL MESSAG("OFFICIAL$",100,5.2,7.5)
    CALL HEIGHT(0.10)
CALL MESSAG("ONDJFMAMJJAS$",100,6.12,7.5)
CALL MESSAG("ONDJFMAMJJAS$",100,7.37,7.5)
CALL MESSAG("ONDJFMAMJJAS$",100,8.62,7.5)
CALL MESSAG("ONDJFMAMJJAS$",100,9.87,7.5)
CALL MESSAG("ONDJFMAMJJAS$",100,11.12,7.5)
```

C
C
C

```
CHART INFORMATION
CALL BASALF("L/CSTD")
CALL MIXALF("STANDARD")
CALL MESSAG("0.0$,100.0.20,7.0")
CALL MESSAG("(MILESTONE 0 - MISSION ANALYSIS/PROJECT INITIATION$",
*100,0.80,7.0)
    CALL VECTOR(0.80,6.95,4.95,6.95,0)
    CALL MESSAG("0.1$",100,0.20,6.7)
    CALL MESSAG("(D)EVELOPMENT OF (MENS)$",100,0.80,6.7)
    CALL MESSAG("(NPPSMO)$",100,5.2,6.7)
        CALL MARKER(9)
    CALL CURVE(XA1(4),YA1(1),1,1)
CALL ENDPL(0)
CALL DONEPL
```

PLAN OF ACTION	

2.5.100 NIVEAU 220MIG 10000-00 2000 1000 10 1000 10 1000

APPENDIX C

**PROGRAM AND PLOT
(PREDEFINED FRAME)**

CA XX,CM200000, P4.
CHARGE,CA XX,XXXXX99999.
REQUEST,PLFILE,PF.
FTN.
ATTACH,NS RDC.
ATTACH,DISPLA.
ATTACH,TEK30.
LDSET,LIB=DISPLA/NS RDC/TEK30.
L00.
CATALOG,PLFILE, ID=CA XX.
EOR

C-----MAIN PROGRAM-----

```
PROGRAM PAM2( INPUT,OUTPUT,TAPE10,TAPE5=INPUT,TAPE6=OUTPUT)
DIMENSION X(7),X1(1),X2(1),X3(1),X4(1)
DIMENSION Y1(1),Y2(1),Y3(1),Y4(1),Y5(1)
DATA X/0.375,2.85,5.5,6.625,7.875,9.125,10.375/
DATA X1/0.0/
DATA X2/11.00/
DATA X3/6.0/
DATA X4/8.50/
DATA Y1/0.0/
DATA Y2/7.2/
DATA Y3/6.5/
DATA Y4/6.8/
DATA Y5/7.0/
```

C
C CHART ASSEMBLY BEGINS HERE
C

```
CALL COMPRS
CALL BGNPL(1)
CALL PAGE(13.0,11.0)
CALL PHYSOR(0.5,0.5)
CALL TITLE("PLAN OF ACTION$",100,2H ,2,2H ,2,11.0,8.5)
    CALL YNUM
    CALL XNUM
    CALL YTICKS(0)
    CALL XTICKS(0)
    CALL BARPAT(0)
    CALL GRAF(0.0,1.0,11.0,0.0,1.0,7.5)
        CALL THICRV(3)
        CALL BARWID(0.75)
        CALL CLUSTR(6.0,0.0)
    CALL VBARS(X(1),Y1(1),Y2(1),1)
        CALL BARWID(4.25)
    CALL VBARS(X(2),Y1(1),Y2(1),1)
        CALL BARWID(1.0)
    CALL VBARS(X(3),Y1(1),Y2(1),1)
        CALL BARWID(1.25)
    CALL VBARS(X(4),Y1(1),Y4(1),1)
    CALL VBARS(X(5),Y1(1),Y4(1),1)
    CALL VBARS(X(6),Y1(1),Y4(1),1)
        CALL BARWID(0.05)
```

```

        CALL THICRV( 5)
        CALL HBARS(X1( 1),X2( 1),Y3( 1),1)
        CALL BARWID(0.45)
        CALL CLUS TR(2.0,0.0)
        CALL HBARS(X3( 1),X4( 1),Y5( 1),1)
        CALL HBARS(X4( 1),X2( 1),Y5( 1),1)

C
C   CHART HEADING S
C
1.    CALL HEIGHT(0.10)
2.    CALL BASALF("STANDARD")
3.    CALL MESSAG("ACTION$",100,0.08,7.8)
4.    CALL HEIGHT(0.10)
5.    CALL MESSAG("RESPONSIBLE$",100,5.05,7.8)
6.    CALL MESSAG("INITIATION DATES $",100,6.5,7.8)
7.    CALL MESSAG("COMPLETION DATES $",100,9.00,7.8)
8.    CALL HEIGHT(0.15)
9.    CALL MESSAG("DESCRIPTION$",100,2.0,7.5)
10.   CALL MESSAG("ID$",100,0.20,7.5)
11.   CALL HEIGHT(0.10)
12.   CALL MESSAG("OFFICIAL$",100,5.2,7.5)
13.   CALL HEIGHT(0.10)
14.   CALL MESSAG("ESTIMATED$",100,6.2,7.5)
15.   CALL MESSAG("ACTUAL$",100,7.47,7.5)
16.   CALL MESSAG("ESTIMATED$",100,8.7,7.5)
16.   CALL MESSAG("ACTUAL$",100,9.97,7.5)

C
C   CHART INFORMATION
C
17.   CALL BASALF("L/CS TD")
18.   CALL MIXALF("STANDARD")
19.   CALL HEIGHT(0.095)
20.   CALL MESSAG("0.0$",100,0.20,7.0)
21.   CALL MESSAG("MILESTONE 0 - MISSION ANALYSIS / PROJECT INITIATION
*)$",100,0.80,7.0)
22.   CALL VECTOR(0.80,6.95,4.95,6.95,0)
23.   CALL MESSAG("0.1$",100,0.20,6.7)
24.   CALL MESSAG("(D)EVELOPMENT OF (MENS)$",100,0.80,6.7)
25.   CALL MESSAG("(NPPSMO)$",100,5.2,6.7)
26.   CALL MESSAG("0.2$",100,0.20,6.5)
27.   CALL MESSAG("(E)STABLISH (P)ROJECT (O)FFICE$",100,0.80,6.5)
28.   CALL MESSAG("(NPPSMO)$",100,5.2,6.5)
29.   CALL MESSAG("0.3$",100,0.20,6.3)
30.   CALL MESSAG("(A)PPPOINT (PM), (ADP), AND (T)ELECOMMUNICATIONS (M)APS$"
*,100,0.80,6.3)
31.   CALL MESSAG("(NPPSMO)$",100,5.2,6.3)
32.   CALL MESSAG("0.4$",100,0.20,6.1)
33.   CALL MESSAG("(D)EVELOP (P)LAN OF (A)CTION AND (M)ILESTONE$",100
*,0.80,6.1)
34.   CALL MESSAG("(DTNSRDC)$",100,5.2,6.1)
35.   CALL MESSAG("0.5$",100,0.20,5.9)
.    CALL MESSAG("(A)PROVAL OF (MENS)$",100,0.80,5.9)

```

. CALL MESSAG("(NAV DAC)\$", 100, 5.2, 6.7)
 . CALL MESSAG("1.0\$", 100, 0.20, 5.5)
 39. CALL MESSAG("(MILESTONE I - CONCEPT DEVELOPMENT PHASE
 *)\$", 100, 0.80, 5.5)
 40. CALL VECTOR(0.80, 5.45, 4.35, 5.43, 0)
 . CALL MESSAG("1.1\$", 100, 0.20, 5.2)
 . CALL MESSAG("(S)UBMIT (POM) (R)EQUIREMENT/(PIF) (P)ROJECT\$"
 *, 100, 0.80, 5.2)
 . CALL MESSAG("(NPPSMO)\$", 100, 5.2, 5.2)
 . CALL MESSAG("1.2\$", 100, 0.20, 5.0)
 45. CALL MESSAG("(C)ONDUCT (G)ENERAL (F)UNCTIONAL (R)EQUIREMENTS
 *(A)NALYSIS\$", 100, 0.80, 5.0)
 . CALL MESSAG("(DTNS RDC)\$", 100, 5.2, 5.0)
 . CALL MESSAG("1.3\$", 100, 0.20, 4.8)
 . CALL MESSAG("(D)EFINE (P)ROTOTYPING (E)XPERIMENTS/(D)EMONSTRATIONS\$"
 *, 100, 0.80, 4.8)
 . CALL MESSAG("(DTNS RDC)\$", 100, 5.2, 4.8)
 50. CALL MESSAG("1.4\$", 100, 0.20, 4.6)
 . CALL MESSAG("(D)EVELOP (P)RELIMINARY (E)CONOMIC (A)NALYSIS\$"
 *, 100, 0.80, 4.6)
 . CALL MESSAG("(DTNS RDC)\$", 100, 5.2, 4.6)
 . CALL MESSAG("1.5\$", 100, 0.20, 4.4)
 . CALL MESSAG("(D)EVELOP (P)RELIMINARY (B)UDGET AND (C)OST (E)STIMATES\$"
 *, 100, 0.80, 4.4)
 55. CALL MESSAG("(DTNS RDC)\$", 100, 5.2, 4.4)
 . CALL MESSAG("1.6\$", 100, 0.20, 4.2)
 . CALL MESSAG("(P)REPARE (P)ROJECT (M)ANAGEMENT (P)LANS\$"
 *, 100, 0.80, 4.2)
 . CALL MESSAG("(DTNS RDC)\$", 100, 5.2, 4.2)
 . CALL MESSAG("1.7\$", 100, 0.20, 4.0)
 60. CALL MESSAG("(P)REPARE (R)ESOURCES (A)NNEX\$"
 *, 100, 0.80, 4.0)
 . CALL MESSAG("(DTNS RDC)\$", 100, 5.2, 4.0)
 . CALL MESSAG("1.8\$", 100, 0.20, 3.8)
 . CALL MESSAG("(P)REPARE (P)RELIMINARY (A)CQUISITION (S)TRATEGY
 *(A)NNEX\$", 100, 0.80, 3.8)
 . CALL MESSAG("(DTNS RDC)\$", 100, 5.2, 3.8)
 65. CALL MESSAG("1.9\$", 100, 0.20, 3.6)
 . CALL MESSAG("(P)REPARE (S)UPPORT (P)LAN (A)NNEX\$"
 *, 100, 0.80, 3.6)
 . CALL MESSAG("(DTNS RDC)\$", 100, 5.2, 3.6)
 . CALL MESSAG("1.10\$", 100, 0.20, 3.4)
 . CALL MESSAG("(F)ORWARD (SDP) 1 FOR (R)EVIEW \$"
 *, 100, 0.80, 3.4)
 70. CALL MESSAG("(DTNS RDC)\$", 100, 5.2, 3.4)
 . CALL MESSAG("1.11\$", 100, 0.20, 3.2)
 . CALL MESSAG("(R)EVIEW (SDP) 1 \$"
 *, 100, 0.80, 3.2)
 . CALL MESSAG("(NPPSMO)\$", 100, 5.2, 3.2)
 . CALL MESSAG("1.12\$", 100, 0.20, 3.0)
 75. CALL MESSAG("(A)PROVE (SDP) 1 \$"
 *, 100, 0.80, 3.0)

. CALL MESSAG("(NAV DAC)\$",100,5.2,3.0)
C
C DATE MANIPULATIONS
C
. CALL MESSAG("(JAN 81)\$",100,9.97,6.7)
. CALL MESSAG("(FEB 81)\$",100,9.97,6.5)
. CALL MESSAG("(FEB 81)\$",100,9.97,6.3)
80. CALL MESSAG("(FEB 81)\$",100,6.20,6.1)
. CALL MESSAG("(FEB 81)\$",100,7.47,6.1)
. CALL MESSAG("(FEB 81)\$",100,8.70,6.1)
. CALL MESSAG("(FEB 81)\$",100,9.97,6.1)
. CALL MESSAG("(APR 81)\$",100,9.97,5.9)
85. CALL MESSAG("(OCT 81)\$",100,9.97,5.7)
. CALL MESSAG("(MAR 81)\$",100,6.20,5.0)
. CALL MESSAG("(MAR 81)\$",100,7.47,5.0)
. CALL MESSAG("(FEB 82)\$",100,8.70,5.0)
. CALL MESSAG("(APR 81)\$",100,6.20,4.8)
90. CALL MESSAG("(APR 81)\$",100,7.47,4.8)
. CALL MESSAG("(DEC 81)\$",100,8.70,4.8)
. CALL MESSAG("(JUN 81)\$",100,6.20,4.6)
. CALL MESSAG("(JUN 81)\$",100,7.47,4.6)
. CALL MESSAG("(FEB 82)\$",100,8.70,4.6)
95. CALL MESSAG("(JUN 81)\$",100,6.20,4.4)
. CALL MESSAG("(JUN 81)\$",100,7.47,4.4)
. CALL MESSAG("(FEB 82)\$",100,8.70,4.4)
. CALL MESSAG("(DEC 81)\$",100,6.20,4.2)
. CALL MESSAG("(JAN 82)\$",100,7.47,4.2)
100. CALL MESSAG("(FEB 82)\$",100,8.70,4.2)
. CALL MESSAG("(DEC 81)\$",100,6.20,4.0)
. CALL MESSAG("(DEC 81)\$",100,7.47,4.0)
. CALL MESSAG("(FEB 82)\$",100,8.70,4.0)
. CALL MESSAG("(DEC 81)\$",100,6.20,3.8)
105. CALL MESSAG("(FEB 82)\$",100,8.70,3.8)
. CALL MESSAG("(DEC 81)\$",100,6.20,3.6)
. CALL MESSAG("(JAN 82)\$",100,7.47,3.6)
. CALL MESSAG("(FEB 82)\$",100,8.70,3.6)
. CALL MESSAG("(APP 82)\$",100,6.20,3.4)
110. CALL MESSAG("(APR 82)\$",100,8.70,3.4)
. CALL MESSAG("(APR 82)\$",100,6.20,3.2)
. CALL MESSAG("(MAY 82)\$",100,8.70,3.2)
. CALL MESSAG("(MAY 82)\$",100,6.20,3.0)
. CALL MESSAG("(JUN 82)\$",100,8.70,3.0)
115. CALL ENDPL(1)
CALL DQEPL
END

PLAN OF ACTION AND MILESTONES

ACTION ID	DESCRIPTION	RESPONSIBLE OFFICIAL	INITIATION DATES		COMPLETION DATES	
			ESTIMATED	ACTUAL	ESTIMATED	ACTUAL
MILESTONE 0 - MISSION ANALYSIS/PROJECT INITIATION						
0.0	Development of MENS	NPPSMO			JUN 81	
0.1	Establish Project Office	NPPSMO			FEB 82	
0.2	Define PM, ADP, and Telecommunications MARS	NPPSMO			FEB 81	
0.3	Approval Plan of Action and Milestones	DTNSRODC	FEB 81		FEB 81	
0.4	Develop Plan of Action and Milestones	DTNSRODC			FEB 81	
0.5	Approval of MENS	DTNSRODC			APR 81	
MILESTONE 1 - CONCEPT DEVELOPMENT PHASE						
1.0	Submit POM Requirements/PfP Project	NPPSMO			OCT 81	
1.1	Conduct General Functional Requirements Analysis	DTNSRODC	MAR 81		FEB 82	
1.2	Define Pilot Experiments/Demonstrations	DTNSRODC	APR 81		DEC 81	
1.3	Develop Preliminary Economic Analysis	DTNSRODC	JUN 81		FEB 82	
1.4	Develop Preliminary Budget and Cost Estimate	DTNSRODC	JUN 81		JUN 81	
1.5	Prepare Project Management Plan	DTNSRODC	DEC 81		FEB 82	
1.6	Prepare Resources Annex	DTNSRODC	DEC 81		DEC 82	
1.7	Prepare Preliminary Acquisition Strategy	DTNSRODC	DEC 81		FEB 82	
1.8	Prepare Support Plan Annex	DTNSRODC	DEC 81		JAN 82	
1.9	Forward SOP 1 for Review	DTNSRODC	APR 82		APR 82	
1.10	Review SOP 1	NPPSMO	APR 82		MAY 82	
1.11	Approve SOP 1	DTNSRODC	MAY 82		JUN 82	
1.12						

APPENDIX D
CHART FILE AND PLOTS

CA XX ,CM200000, P4.
CHARGE ,CA XX ,XXXXX99999.
REQUEST, PLFILE, *PF.

FTN.
ATTACH ,NS RDC.

ATTACH ,DISSPLA.

ATTACH ,TEK30.

LDSET, LIB= DISSPLA/NSRDC/TEK30.

LGO.

CATALOG, PLFILE, ID= CA XX .

EOR

C-----MAIN PROGRAM-----

```
PROGRAM PAM1( INPUT, OUTPUT, TAP E10, TAP E5 =INPUT, TAP E6 =OUTPUT)
DIMENSION X(8),X1( 1),X2( 1),Y1( 1),Y2( 1),Y3( 1)
DIMENSION XA1( 12),XA2( 12),XA3( 12),XA4( 12),XA5( 12),YA1( 17)
DATA X /0. 375, 2.85, 5.5, 6.625, 7.875, 9.125, 10.375, 11.625/
DATA X1 /0.0/
DATA X2 /12.25/
DATA Y1 /0.0/
DATA Y2 /7.2/
DATA Y3 /6.5/
1. DATA XA1 /6.11, 6.21, 6.31, 6.41, 6.51, 6.61, 6.69, 6.79, 6.89, 6.99, 7.09, 7.19/
2. DATA XA2 /7.36, 7.46, 7.56, 7.66, 7.76, 7.86, 7.94, 8.04, 8.14, 8.24, 8.34, 8.44/
3. DATA XA3 /8.61, 8.71, 8.81, 8.91, 9.01, 9.11, 9.19, 9.29, 9.39, 9.49, 9.59, 9.69/
4. DATA XA4 /9.86, 9.96, 10.06, 10.16, 10.26, 10.36, 10.44, 10.54, 10.64, 10.74,
*10.84, 10.94/
5. DATA XA5 /11.11, 11.21, 11.31, 11.41, 11.51, 11.61, 11.69, 11.79, 11.89, 11.99,
*12.09, 12.19/
6. DATA YA1 /5.96, 5.77, 5.6, 5.42, 5.23, 4.63, 4.45, 4.28, 4.09, 3.92, 3.74, 3.57,
*3.38, 3.21, 3.03, 2.86, 2.68/
```

C

C CHART ASSEMBLY BEGINS HERE

C

```
CALL COMPRS
CALL BGNPL( 1)
CALL PAGE( 13.0, 11.0)
CALL PHYSOR( 0.5, 0.5)
CALL TITLE("PLAN OF ACTION AND MILESTONES", 100, 2H , 2, 2H , 2, 12.25, 8.5)
    CALL YNUM
    CALL XNUM
    CALL YTICKS( 0)
    CALL XTICKS( 0)
    CALL BARPAT( 0)
    CALL GRAF( 0.0, 1.0, 12.25, 0.0, 1.0, 7.5)
    CALL THICRV( 3)
    CALL BARWID( 0.75)
    CALL CLUSRTR( 6.0, 0.0)
    CALL VBARS( X( 1), Y1( 1), Y2( 1), 1)
    CALL BARWID( 4.25)
    CALL VBARS( X( 2), Y1( 1), Y2( 1), 1)
    CALL BARWID( 1.0)
    CALL VBARS( X( 3), Y1( 1), Y2( 1), 1)
```

```

    CALL BARWID(1.25)
    CALL VBARS(X(4),Y 1( 1 ),Y 2( 1 ),1)
    CALL VBARS(X(5),Y 1( 1 ),Y 2( 1 ),1)
    CALL VBARS(X(6),Y 1( 1 ),Y 2( 1 ),1)
    CALL VBARS(X(7),Y 1( 1 ),Y 2( 1 ),1)
    CALL VBARS(X(8),Y 1( 1 ),Y 2( 1 ),1)
        CALL BARWID(0.05)
        CALL THICRV(5)
    CALL HBARS(X 1( 1 ),X 2( 1 ),Y 3( 1 ),1)

C   CHART    HEADING S
C
    CALL HEIGHT(0.10)
    CALL BASALF("STANDARD")
    CALL MESSAG("ACTION$",100,0.08,.7.8)
        CALL HEIGHT(0.10)
    CALL MESSAG("RESPONSIBLE$",100,5.05,7.8)
        CALL HEIGHT(0.15)
    CALL MESSAG("FY 81$",100,6.3,7.8)
    CALL MESSAG("FY 82$",100,7.6,7.8)
    CALL MESSAG("FY 83$",100,8.9,7.8)
    CALL MESSAG("FY 84$",100,10.1,7.8)
    CALL MESSAG("FY 85$",100,11.4,7.8)
    CALL MESSAG("DESCRIPTION$",100,2.0,7.5)
    CALL MESSAG("ID$",100,0.20,7.5)
        CALL HEIGHT(0.10)
    CALL MESSAG("OFFICIAL$",100,5.2,7.5)
        CALL HEIGHT(0.10)
    CALL MESSAG("ONDJFMAMJJAS$ ",100,6.12,7.5)
    CALL MESSAG("ONDJFMAMJJAS$ ",100,7.37,7.5)
    CALL MESSAG("ONDJFMAMJJAS$ ",100,8.62,7.5)
    CALL MESSAG("ONDJFMAMJJAS$ ",100,9.87,7.5)
    CALL MESSAG("ONDJFMAMJJAS$ ",100,11.12,7.5)

C   CHART    INFORMATION
C
    CALL BASALF("L/CS TD")
    CALL MIXALF("STANDARD")
        CALL HEIGHT(0.095)
    CALL MESSAG("0.0$",100,0.20,7.0)
    CALL MESSAG("(MILESTONE(0)- MISSION ANALYSIS / PROJECT INITIATION$",
    &100,0.80,7.0)
        CALL VECTOR(0.80,6.95,4.95,6.95,0)
    CALL MESSAG("0.1$",100,0.20,6.7)
    CALL MESSAG("(D)EVELOPMENT OF (MENS)$",100,0.80,6.7)
    CALL MESSAG("(NPPSMO)$",100,5.2,6.7)
        CALL MARKER(2)
    CALL CURVE(XA1(4),YA1(1),1,1)
    CALL MESSAG("0.2$",100,0.20,6.5)
    CALL MESSAG("(E)STABLISH (P)ROJECT (O)FFICE$",100,0.80,6.5)
    CALL MESSAG("(NPPSMO)$",100,5.2,6.5)
        CALL MARKER(2)

```

```

CALL CURVE(XA1(5),YA1(2),1,1)
CALL MESSAG("0.3$",100,0.20,6.3)
CALL MESSAG("(A)PPPOINT (PM), (ADP) AND (T)ELECOMMUNICATIONS (MARS)$",
&100,0.80,6.3)
CALL MESSAG("(NPPSMO)$",100,5.2,6.3)
    CALL MARKER(2)
CALL CURVE(XA1(5),YA1(3),1,1)
CALL MESSAG("0.4$",100,0.20,6.1)
CALL MESSAG("(D)EVELOP (P)LAN OF (A)CTION AND (M)ILESTONE$",
&100,0.80,6.1)
CALL MESSAG("(DTNSRDC)$",100,5.2,6.1)
    CALL MARKER(2)
CALL CURVE(XA1(5),YA1(4),1,1)
CALL MESSAG("0.5$",100,0.20,5.9)
CALL MESSAG("(A)PROVAL OF (MENS)$",100,0.80,5.9)
CALL MESSAG("(NAVDAC)$",100,5.2,5.9)
    CALL MARKER(2)
CALL CURVE(XA1(7),YA1(5),1,1)
CALL MESSAG("1.0$",100,0.20,5.5)
CALL MESSAG("(MILESTONE) 1 - (CONCEPT DEVELOPMENT PHASE$",100,0.80,5.5)
CALL VECTOR(0.80,5.45,4.35,5.45,0)
CALL MESSAG("1.1$",100,0.20,5.2)
CALL MESSAG("(S)UBMIT (POM) (R)EQUIREMENT/(PIE) (P)ROJECTS$",
&100,0.80,5.2)
CALL MESSAG("(NPPSMO)$",100,5.2,5.2)
    CALL MARKER(2)
CALL CURVE(XA1(1),YA1(6),1,1)
CALL MESSAG("1.2$",100,0.20,5.0)
CALL MESSAG("(C)ONDUCT (G)ENERAL (F)UNCTIONAL (R)EQUIREMENTS (A)NALYSIS$"
&,100,0.80,3.6)
CALL MESSAG("(DTNSRDC)$",100,5.2,5.0)
CALL VECTOR(6.62,5.1,6.62,5.0,0)
CALL VECTOR(6.62,5.0,7.77,5.0,0)
    CALL MARKER(2)
CALL CURVE(XA2(5),YA1(7),1,1)
CALL MESSAG("1.3$",100,0.20,4.8)
CALL MESSAG("(D)EFINE (P)ILOT (E)XPERIMENTS/(D)EMONSTRATIONS$",
&100,0.80,3.6)
CALL MESSAG("(DTNSRDC)$",100,5.2,4.8)
CALL VECTOR(6.72,4.9,6.72,4.8,0)
CALL VECTOR(6.72,4.8,7.57,4.8,0)
    CALL MARKER(2)
CALL CURVE(XA2(3),YA1(8),1,1)
CALL MESSAG("1.4$",100,0.20,4.6)
CALL MESSAG("(D)EVELOP (P)RELIMINARY (E)CONOMIC (A)NALYSIS$",
&100,0.80,4.6)
CALL MESSAG("(DTNSRDC)$",100,5.2,4.6)
CALL VECTOR(6.92,4.7,6.92,4.6,0)
CALL VECTOR(6.92,4.6,7.77,4.6,0)
    CALL MARKER(2)
CALL CURVE(XA2(5),YA1(9),1,1)
CALL MESSAG("1.5$",100,0.20,4.4)

```

```

CALL MESSAG(" (D)EV ELOP (P)RELIMINARY (B)UDGET AND (C)OST (E)STIMATES$",
&100,0.80,4.4)
CALL MESSAG(" (DTNSRDC)$",100,5.2,4.4)
CALL VECTOR( 6.92, 4.5, 6.92, 4.4, 0)
CALL VECTOR( 6.92, 4.4, 7.77, 4.4, 0)
    CALL MARKER(2)
CALL CURVE(XA2( 5),YA1( 10),1,1)
CALL MESSAG("1.6$",100,0.20,4.2)
CALL MESSAG(" (P)REPARE (P)ROJECT (M)ANAGEMENT (P)LAN$",
&100,0.80,4.2)
CALL MESSAG(" (DTNSRDC)$",100,5.2,4.2)
CALL VECTOR( 7.67, 4.3, 7.67, 4.2, 0)
CALL VECTOR( 7.67, 4.2, 7.77, 4.2, 0)
    CALL MARKER(2)
CALL CURVE(XA2( 5),YA1( 11),1,1)
CALL MESSAG("1.7$",100,0.20,4.0)
CALL MESSAG(" (P)REPARE (R)ESOURCES (A)NNEX$",
&100,0.80,4.0)
CALL MESSAG(" (DTNSRDC)$",100,5.2,4.0)
CALL VECTOR( 7.57, 4.1, 7.57, 4.0, 0)
CALL VECTOR( 7.57, 4.0, 7.77, 4.0, 0)
    CALL MARKER(2)
CALL CURVE(XA2( 5),YA1( 12),1,1)
CALL MESSAG("1.8$",100,0.20,3.8)
CALL MESSAG(" (P)REPARE (P)RELIMINARY (A)CQUISITION (S)TRATEGY$",
&100,0.80,3.8)
CALL MESSAG(" (DTNSRDC)$",100,5.2,3.8)
CALL VECTOR( 7.57, 3.9, 7.57, 3.8, 0)
CALL VECTOR( 7.57, 3.8, 7.77, 3.8, 0)
    CALL MARKER(2)
CALL CURVE(XA2( 5),YA1( 13),1,1)
CALL MESSAG("1.9$",100,0.20,3.6)
CALL MESSAG(" (P)REPARE (S)UPPORT (P)LAN (A)NNEX$",100,0.80,3.6)
CALL MESSAG(" (DTNSRDC)$",100,5.2,3.6)
CALL VECTOR( 7.67, 3.7, 7.67, 3.6, 0)
CALL VECTOR( 7.67, 3.6, 7.77, 3.6, 0)
    CALL MARKER(2)
CALL CURVE(XA2( 5),YA1( 14),1,1)
CALL MESSAG("1.10$",100,0.20,3.4)
CALL MESSAG(" (F)ORWARD (SDP) 1 FOR (R)EVIEW$",100,0.80,3.4)
CALL MESSAG(" (DTNSRDC)$",100,5.2,3.4)
    CALL MARKER(2)
CALL CURVE(XA2( 5),YA1( 15),1,1)
CALL MESSAG("1.11$",100,0.20,3.2)
CALL MESSAG(" (R)EVIEW (SDP) 1$",100,0.80,3.2)
CALL MESSAG(" (NPPSMO)$",100,5.2,3.2)
CALL VECTOR( 7.97, 3.3, 7.97, 3.2, 0)
CALL VECTOR( 7.97, 3.2, 8.07, 3.2, 0)
    CALL MARKER(2)
CALL CURVE(XA2( 8),YA1( 16),1,1)
CALL MESSAG("1.12 $",100,0.20,3.0)
CALL MESSAG(" (A)PROVE (SDP) 1$",100,0.80,3.0)

```

```
CALL MESSAG("(NAVDAC)$",100,5.2,3.0)
CALL VECTOR(8.07,3.1,8.07,3.0,0)
CALL VECTOR(8.07,3.0,8.17,3.0,0)
    CALL MARKER(2)
CALL CURVE(XA2(9),YA1(17),1,1)
CALL ENDPL(1)
CALL DONEPL
END
```

PLAN OF ACTION AND MILESTONES

ACTION ID	DESCRIPTION	RESPONSIBLE OFFICIAL	FY 81	FY 82	FY 83	FY 84	FY 85
			OND/DMN/JRS	OND/DMN/JRS	OND/DMN/JRS	OND/DMN/JRS	OND/DMN/JRS
0.0	MILESTONE 0 - MISSION ANALYSIS/PROJECT INITIATION						
0.1	Development of MENS	NPPSMO	△				
0.2	Establish Project Office	NPPSMO	△				
0.3	Appoint PM, NPF, and Telecommunications MPPS	NPPSMO	△				
0.4	Develop Plan of Action and Milestones	DNSRDC	△				
0.5	Approval of MENS	DNSRDC	△				
1.0	MILESTONE 1 - CONCEPT DEVELOPMENT PHASE:						
1.1	Submit PON Requirements/PIT Project	NPPSMO	△				
1.2	Conduct General Functional Requirements Analysis	DNSRDC					
1.3	Define Pilot Experiments/Demonstrations	DNSRDC					
1.4	Develop Preliminary Economic Analysis	DNSRDC					
1.5	Develop Preliminary Budget and Cost Estimates	DNSRDC					
1.6	Prepare Project Management Plan	DNSRDC					
1.7	Prepare Resources Annex	DNSRDC					
1.8	Prepare Preliminary Acquisition Strategy	DNSRDC					
1.9	Prepare Support Plan Annex	DNSRDC					
1.10	Forward SPP 1 for Review	DNSRDC					
1.11	Review SPP 1	NPPSMO					
1.12	Approve SPP 1	NPPSMO					

2 9 02 1985 20:04:56 000000 000 77 100 97 55 E 1 070

APPENDIX E
POAM2 FILE AND PLOTS

```

CAXX,CM200000,P4.
CHARGE,CAXX,XXXXXX99999.
REQUEST,PLFILE,*PF.
FTN.
ATTACH,NSRDC.
ATTACH,DISSPLA.
ATTACH,TEK30.
LPSET,LIB=DISSPLA/NSRDC/TEK30.
LGO.
CATALOG,PLFILE,ID=CAXX.
EOR
C-----MAIN PROGRAM-----
PROGRAM PAM1(INPUT,OUTPUT,TAPE10,TAPE5=INPUT,TAPE6=OUTPUT)
DIMENSION X(12),X1(1),X2(1),X3(1),X4(1),Y1(1),Y2(1),Y3(1),Y4(1),Y5(1)
DIMENSION XA1(12),XA2(12),XA3(12),XA4(12),XA5(12),YA1(17)
DATA X /0.375,2.85,5.5,6.625,7.875,9.125,10.375,11.625,12.75,13.75,
& 14.75,15.75/
DATA X1 /0.0/
DATA X2 /16.25/
DATA X3 /12.25/
DATA X4 /14.25/
DATA Y1 /0.0/
DATA Y2 /7.2/
DATA Y3 /6.5/
DATA Y4 /6.8/
DATA Y5 /7.0/
1. DATA XA1 /6.11,6.21,6.31,6.41,6.51,6.61,6.69,6.79,6.89,6.99,7.09,7.19/
2. DATA XA2 /7.36,7.46,7.56,7.66,7.76,7.86,7.94,8.04,8.14,8.24,8.34,8.44/
3. DATA XA3 /8.61,8.71,8.81,8.91,9.01,9.11,9.19,9.29,9.39,9.49,9.59,9.69/
4. DATA XA4 /9.86,9.96,10.06,10.16,10.26,10.36,10.44,10.54,10.64,10.74,
*10.84,10.94/
5. DATA XA5 /11.11,11.21,11.31,11.41,11.51,11.61,11.69,11.79,11.89,11.99,
*12.09,12.19/
6. DATA YA1 /5.96,5.77,5.6,5.42,5.23,4.63,4.45,4.28,4.09,3.92,3.74,3.57,
*3.38,3.21,3.03,2.86,2.68/
C
C CHART ASSEMBLY BEGINS HERE
C
CALL COMPRS
CALL BGNPL(1)
CALL PAGE(17.0,11.0)
CALL PHYSOR(0.5,0.5)
CALL TITLE(2H ,2,2H ,2,2H ,2,16.25,8.5)
CALL HEIGHT(0.21)
CALL MESSAG("PLAN OF ACTION AND MILESTONES$",100,5.5,8.4)
    CALL YNOUNM
    CALL XNOUNM
    CALL YTICKS(0)
    CALL XTICKS(0)
    CALL BARPAT(0)
    CALL GRAF(0.0,1.0,16.25,0.0,1.0,7.5)
    CALL THICRV(3)

```

```
CALL BARWID(0.75)
CALL CLUSTR(12.0,0.0)
CALL VBARS(X(1),Y1(1),Y2(1),1)
CALL BARWID(4.25)
CALL VBARS(X(2),Y1(1),Y2(1),1)
CALL BARWID(1.0)
CALL VBARS(X(3),Y1(1),Y2(1),1)
CALL BARWID(1.25)
CALL VBARS(X(4),Y1(1),Y2(1),1)
CALL VBARS(X(5),Y1(1),Y2(1),1)
CALL VBARS(X(6),Y1(1),Y2(1),1)
CALL VBARS(X(7),Y1(1),Y2(1),1)
CALL VBARS(X(8),Y1(1),Y2(1),1)
CALL BARWID(1.0)
CALL VBARS(X(9),Y1(1),Y4(1),1)
CALL VBARS(X(10),Y1(1),Y4(1),1)
CALL VBARS(X(11),Y1(1),Y4(1),1)
CALL VBARS(X(12),Y1(1),Y4(1),1)
CALL BARWID(0.05)
CALL THICRV(5)
CALL HBARS(X1(1),X2(1),Y3(1),1)
CALL BARWID(0.45)
CALL CLUSTR(2.0,0.0)
CALL HBARS(X3(1),X4(1),Y5(1),1)
CALL HBARS(X4(1),X2(1),Y5(1),1)
```

C
C CHART HEADINGS
C

```
CALL HEIGHT(0.10)
CALL BASALF("STANDARD")
CALL MESSAG("ACTION$",100,0.08,7.8)
CALL HEIGHT(0.10)
CALL MESSAG("RESPONSIBLE$",100,5.05,7.8)
CALL MESSAG("INITIATION DATES $",100,12.5,8.7)
CALL MESSAG("COMPLETION DATES $",100,14.5,8.7)
CALL HEIGHT(0.15)
CALL MESSAG("FY 81$",100,6.3,7.8)
CALL MESSAG("FY 82$",100,7.6,7.8)
CALL MESSAG("FY 83$",100,8.9,7.8)
CALL MESSAG("FY 84$",100,10.1,7.8)
CALL MESSAG("FY 85$",100,11.4,7.8)
CALL MESSAG("DESCRIPTION$",100,2.0,7.5)
CALL MESSAG("ID$",100,0.20,7.5)
CALL HEIGHT(0.10)
CALL MESSAG("OFFICIAL$",100,5.2,7.5)
CALL HEIGHT(0.10)
CALL MESSAG("ONDJFMAMJJAS$",100,6.12,7.5)
CALL MESSAG("ONDJFMAMJJAS$",100,7.37,7.5)
CALL MESSAG("ONDJFMAMJJAS$",100,8.62,7.5)
CALL MESSAG("ONDJFMAMJJAS$",100,9.87,7.5)
CALL MESSAG("ONDJFMAMJJAS$",100,11.12,7.5)
CALL MESSAG("ESTIMATED$",100,12.37,7.5)
```

```
CALL MESSAG("ACTUAL$",100,13.375,7.5)
CALL MESSAG("ESTIMATED$",100,14.37,7.5)
CALL MESSAG("ACTUAL$",100,15.375,7.5)
```

C
C
C

CHART INFORMATION

```
CALL BASALF("L/CSTD")
CALL MIXALF("STANDARD")
CALL HEIGHT(0.095)
CALL MESSAG("0.0$",100,0.20,7.0)
CALL MESSAG("(MILESTONE(0)- MISSION ANALYSIS/PROJECT INITIATION$",
&100,0.80,7.0)
CALL VECTOR(0.80,6.95,4.95,6.95,0)
CALL MESSAG("0.1$",100,0.20,6.7)
CALL MESSAG("(D)EVELOPMENT OF (MENS)$",100,0.80,6.7)
CALL MESSAG("(NPPSMO)$",100,5.2,6.7)
    CALL MARKER(2)
CALL CURVE(XA1(4),YA1(1),1,1)
CALL MESSAG("0.2$",100,0.20,6.5)
CALL MESSAG("(E)STABLISH (P)ROJECT (O)FFICE$",100,0.80,6.5)
CALL MESSAG("(NPPSMO)$",100,5.2,6.5)
    CALL MARKER(2)
CALL CURVE(XA1(5),YA1(2),1,1)
CALL MESSAG("0.3$",100,0.20,6.3)
CALL MESSAG("(A)POINT (PM), (ADP) AND (T)ELECOMMUNICATIONS (MARS)$",
&100,0.80,6.3)
CALL MESSAG("(NPPSMO)$",100,5.2,6.3)
    CALL MARKER(2)
CALL CURVE(XA1(5),YA1(3),1,1)
CALL MESSAG("0.3$",100,0.20,6.1)
CALL MESSAG("(D)EVELOP (P)LAN OF (A)CTION AND (M)ILESTONES$",
&100,0.80,6.1)
CALL MESSAG("(DTNSRDC)$",100,5.2,6.1)
    CALL MARKER(2)
CALL CURVE(XA1(5),YA1(4),1,1)
CALL MESSAG("0.5$",100,0.20,5.9)
CALL MESSAG("(A)PROVAL OF (MENS)$",100,0.80,5.9)
CALL MESSAG("(NAVDAC)$",100,5.2,5.9)
    CALL MARKER(2)
CALL CURVE(XA1(7),YA1(5),1,1)
CALL MESSAG("1.0$",100,0.20,5.5)
CALL MESSAG("(MILESTONE) 1 - (CONCEPT DEVELOPMENT PHASE$",100,0.80,5.5)
CALL VECTOR(0.80,5.45,4.35,5.45,0)
CALL MESSAG("1.1$",100,0.20,5.2)
CALL MESSAG("(S)UBMIT (POM) (R)EQUIREMENT/(PIE) (P)ROJECTS$",
&100,0.80,5.2)
CALL MESSAG("(NPPSMO)$",100,5.2,5.2)
    CALL MARKER(2)
CALL CURVE(XA1(1),YA1(6),1,1)
CALL MESSAG("1.2$",100,0.20,5.0)
CALL MESSAG("(C)ONDUCT (G)ENERAL (F)UNCTIONAL (R)EQUIREMENTS (A)NALYSIS$"
&,100,0.80,3.6)
```

```
CALL MESSAG("(DTNSRDC)$",100,5.2,5.0)
CALL VECTOR(6.62,5.1,6.62,5.0,0)
CALL VECTOR(6.62,5.0,7.77,5.0,0)
    CALL MARKER(2)
CALL CURVE(XA2(5),YA1(7),1,1)
CALL MESSAG("1.3$",100,0.20,4.8)
CALL MESSAG("(D)EFINE (P)ILOT (E)XPERIMENTS/(D)EMONSTRATIONS$",
&100,0.80,3.6)
CALL MESSAG("(DTNSRDC)$",100,5.2,4.8)
CALL VECTOR(6.72,4.9,6.72,4.8,0)
CALL VECTOR(6.72,4.8,7.57,4.8,0)
    CALL MARKER(2)
CALL CURVE(XA2(3),YA1(8),1,1)
CALL MESSAG("1.4$",100,0.20,4.6)
CALL MESSAG("(D)EVELOP (P)RELIMINARY (E)CONOMIC (A)NALYSIS$",
&100,0.80,4.6)
CALL MESSAG("(DTNSRDC)$",100,5.2,4.6)
CALL VECTOR(6.92,4.7,6.92,4.6,0)
CALL VECTOR(6.92,4.6,7.77,4.6,0)
    CALL MARKER(2)
CALL CURVE(XA2(5),YA1(9),1,1)
CALL MESSAG("1.5$",100,0.20,4.4)
CALL MESSAG("(D)EVELOP (P)RELIMINARY (B)UDGET AND (C)OST (E)STIMATES$",
&100,0.80,4.4)
CALL MESSAG("(DTNSRDC)$",100,5.2,4.4)
CALL VECTOR(6.92,4.5,6.92,4.4,0)
CALL VECTOR(6.92,4.4,7.77,4.4,0)
    CALL MARKER(2)
CALL CURVE(XA2(5),YA1(10),1,1)
CALL MESSAG("1.6$",100,0.20,4.2)
CALL MESSAG("(P)REPARE (P)ROJECT (M)ANAGEMENT (P)LAN$",
&100,0.80,4.2)
CALL MESSAG("(DTNSRDC)$",100,5.2,4.2)
CALL VECTOR(7.67,4.3,7.67,4.2,0)
CALL VECTOR(7.67,4.2,7.77,4.2,0)
    CALL MARKER(2)
CALL CURVE(XA2(5),YA1(11),1,1)
CALL MESSAG("1.7$",100,0.20,4.0)
CALL MESSAG("(P)REPARE (R)ESOURCES (A)NNEX$",
&100,0.80,4.0)
CALL MESSAG("(DTNSRDC)$",100,5.2,4.0)
CALL VECTOR(7.57,4.1,7.57,4.0,0)
CALL VECTOR(7.57,4.0,7.77,4.0,0)
    CALL MARKER(2)
CALL CURVE(XA2(5),YA1(12),1,1)
CALL MESSAG("1.8$",100,0.20,3.8)
CALL MESSAG("(P)REPARE (P)RELIMINARY (A)CQUISITION (S)TRATEGY$",
&100,0.80,3.8)
CALL MESSAG("(DTNSRDC)$",100,5.2,3.8)
CALL VECTOR(7.57,3.9,7.57,3.8,0)
CALL VECTOR(7.57,3.8,7.77,3.8,0)
    CALL MARKER(2)
```

```
CALL CURVE(XA2(5),YA1(13),1,1)
CALL MESSAG("1.9$",100,0.20,3.6)
CALL MESSAG("(P)REPRAE (S)UPPORT (P)LAN (A)NNEX$",100,0.80,3.6)
CALL MESSAG("(DTNSRDC)$",100,5.2,3.6)
CALL VECTOR(7.67,3.7,7.67,3.6,0)
CALL VECTOR(7.67,3.6,7.77,3.6,0)
    CALL MARKER(2)
CALL CURVE(XA2(5),YA1(14),1,1)
CALL MESSAG("1.10$",100,0.20,3.4)
CALL MESSAG("(F)ORWARD (SDP) 1 FOR (R)EVIEW$",100,0.80,3.4)
CALL MESSAG("(DTNSRDC)$",100,5.2,3.4)
    CALL MARKER(2)
CALL CURVE(XA2(5),YA1(15),1,1)
CALL MESSAG("1.11$",100,0.20,3.2)
CALL MESSAG("(R)EVIEW (SDP) 1$",100,0.80,3.2)
CALL MESSAG("(NPPSMO)$",100,5.2,3.2)
CALL VECTOR(7.97,3.3,7.97,3.2,0)
CALL VECTOR(7.97,3.2,8.07,3.2,0)
    CALL MARKER(2)
CALL CURVE(XA2(8),YA1(16),1,1)
CALL MESSAG("1.12$",100,0.20,3.0)
CALL MESSAG("(A)PPROVE (SDP) 1$",100,0.80,3.0)
CALL MESSAG("(NAVDAC)$",100,5.2,3.0)
CALL VECTOR(8.07,3.1,8.07,3.0,0)
CALL VECTOR(8.07,3.0,8.17,3.0,0)
    CALL MARKER(2)
CALL CURVE(XA2(9),YA1(17),1,1)
```

C
C
C

DATE MANIPULATIONS

```
CALL MESSAG("(JAN 81)$",100,15.375,6.7)
CALL MESSAG("(FEB 81)$",100,15.375,6.5)
CALL MESSAG("(FEB 81)$",100,15.375,6.3)
CALL MESSAG("(FEB 81)$",100,12.37,6.1)
CALL MESSAG("(FEB 81)$",100,13.375,6.1)
CALL MESSAG("(FEB 81)$",100,14.37,6.1)
CALL MESSAG("(FEB 81)$",100,15.375,6.1)
CALL MESSAG("(APR 81)$",100,15.375,5.9)
CALL MESSAG("(OCT 81)$",100,15.375,5.2)
CALL MESSAG("(MAR 81)$",100,12.37,5.0)
CALL MESSAG("(FEB 82)$",100,14.37,5.0)
CALL MESSAG("(APR 81)$",100,12.37,4.8)
CALL MESSAG("(APR 81)$",100,13.375,4.8)
CALL MESSAG("(DEC 81)$",100,14.37,4.8)
CALL MESSAG("(JUN 81)$",100,12.37,4.6)
CALL MESSAG("(JUN 81)$",100,13.375,4.6)
CALL MESSAG("(FEB 82)$",100,14.37,4.6)
CALL MESSAG("(JUN 81)$",100,12.37,4.4)
CALL MESSAG("(JUN 81)$",100,13.375,4.4)
CALL MESSAG("(FEB 82)$",100,14.37,4.4)
CALL MESSAG("(DEC 81)$",100,12.37,4.2)
CALL MESSAG("(JAN 82)$",100,13.375,4.2)
```

```
CALL MESSAG("(FEB 82)$",100,14.37,4.2)
CALL MESSAG("(DEC 81)$",100,12.37,4.0)
CALL MESSAG("(DEC 81)$",100,13.375,4.0)
CALL MESSAG("(FEB 82)$",100,14.37,4.0)
CALL MESSAG("(DEC 81)$",100,12.37,3.8)
CALL MESSAG("(FEB 82)$",100,14.37,3.8)
CALL MESSAG("(DEC 81)$",100,12.37,3.6)
CALL MESSAG("(JAN 82)$",100,13.375,3.6)
CALL MESSAG("(FEB 82)$",100,14.37,3.6)
CALL MESSAG("(APR 82)$",100,12.37,3.4)
CALL MESSAG("(APR 82)$",100,14.37,3.4)
CALL MESSAG("(APR 82)$",100,12.37,3.2)
CALL MESSAG("(MAY 82)$",100,14.37,3.2)
CALL MESSAG("(MAY 82)$",100,12.37,3.0)
CALL MESSAG("(JUN 82)$",100,14.37,3.0)
CALL ENDPL(0)
CALL DONEPL
END
```

PLAN OF ACTION AND MILESTONES

ACTION ID	DESCRIPTION	RESPONSIBILITIES				INITIATION DATES				COMPLETION DATES			
		FY 81	FY 82	FY 83	FY 84	OND FORMATS	OND FORMATS	OND FORMATS	OND FORMATS	ESTIMATED	ACTUAL	ESTIMATED	ACTUAL
MILESTONE 0 MISSION ANALYSIS/PROJECT INITIATION													
0.9	Development of MENS	NPP/MO								JUN 81		FEB 81	
0.1	Establish Project Office	NPP/MO								FEB 81		FEB 81	
0.2	Appoint PM, TEP, and Telecommunications Manager	NPP/MO											
0.3	Develop Plan of Action and Milestones	DTNSRODC											
0.4	Approval of MENS	NAIVORC											
0.5	Approval of MENS												
MILESTONE 1 - CONCEPT DEVELOPMENT PHASE													
1.0	Submit PDR Requirements/PITF Project	NPP/MO								OCT 81			
1.1	Contract General Functional Requirements Analysis	DTNSRODC											
1.2	Define Pilot Experiments/Demonstrations	DTNSRODC											
1.3	Develop Preliminary Economic Analysis	DTNSRODC											
1.4	Develop Preliminary Budget and Cost Estimate	DTNSRODC											
1.5	Prepare Project Management Plan	DTNSRODC											
1.6	Prepare Resources Annex	DTNSRODC											
1.7	Prepare Preliminary Acquisition Strategy Annex	DTNSRODC											
1.8	Prepare Support Plan Annex	DTNSRODC											
1.9	Forward SDR 1 for Review	DTNSRODC											
1.10	Review SDR 1	NPP/MO											
1.11	Approve SDR 1	NAIVORC											
1.12	Approve SDR 1												

APPENDIX F
POAM'S FILE AND PLOTS

CAXX,CM200000,P4.
CHARGE,CAXX,XXXXX99999.
REQUEST,PLFILE,*PF.
FTN.
ATTACH,NSRDC.
ATTACH,DISSPLA.
ATTACH,TEK30.
LDSET,LIB=DISSPLA/NSRDC/TEK30.
LGO.
CATALOG,PLFILE,ID=CAXX.

EOR

C-----MAIN PROGRAM-----

```
PROGRAM POAM(INPUT,OUTPUT,TAPE10,TAPE5=INPUT,TAPE6=OUTPUT)
DIMENSION X(12),X1(1),X2(1),X3(1),X4(1),Y1(1),Y2(1),Y3(1),Y4(1),Y5(1)
DIMENSION XA1(12),XA2(12),XA3(12),XA4(12),XA5(12),YA1(17)
DIMENSION MM(12)
REAL OFFSET
DATA MM /3HOCH, 3HNOV, 3HDEC, 3HJAN, 3HFEB, 3HMAR, 3HAPR, 3HMAY,
*3HJUN, 3HJUL, 3HAUG, 3HSEP/
DATA X /0.375,2.85,5.5,6.625,7.875,9.125,10.375,11.625,12.75,13.75,
&14.75,15.75/
DATA X1 /0.0/
DATA X2 /16.25/
DATA X3 /12.25/
DATA X4 /14.25/
DATA Y1 /0.0/
DATA Y2 /8.7/
DATA Y3 /8.0/
DATA Y4 /8.3/
DATA Y5 /8.5/
DATA XA1 /6.11,6.21,6.31,6.41,6.51,6.61,6.69,6.79,6.89,6.99,7.09,7.19/
DATA XA2 /7.36,7.46,7.56,7.66,7.76,7.86,7.94,8.04,8.14,8.24,8.34,8.44/
DATA XA3 /8.61,8.71,8.81,8.91,9.01,9.11,9.19,9.29,9.39,9.49,9.59,9.69/
DATA XA4 /9.86,9.96,10.06,10.16,10.26,10.36,10.44,10.54,10.64,10.74,
*10.84,10.94/
DATA XA5 /11.11,11.21,11.31,11.41,11.51,11.61,11.69,11.79,11.89,11.99,
*12.09,12.19/
DATA YA1 /7.685,7.495,7.325,7.145,6.955,6.355,6.175,6.005,5.815,
*5.645,5.465,5.95,5.105,4.935,4.755,4.585,4.405/
OFFSET = 0.0
ANS = 0.0
ANS3 = 0.0
SP1 = 0.0
SP2 = 0.0
SP3 = 0.0
YVALU = 0.0
ANS2 = 0.0
XBLK1 = 0.21
XBLK2 = 0.80
XBLK3 = 5.2
XBLK9 = 12.37
XBLK10 = 13.375
```

XBLK11 = 14.37
XBLK12 = 15.375

C
C CHART ASSEMBLY BEGINS HERE
C
CALL COMPRS
CALL BGNPL(1)
CALL BLOWUP(1.0/1.2)
CALL PAGE(17.0,11.5)
CALL PHYSOR(0.5,0.5)
CALL NOBRDR
CALL TITLE(2H ,2,2H ,2,2H ,2,16.25,10.0)
CALL HEIGHT(0.21)
CALL MESSAG("PLAN OF ACTION AND MILESTONES\$",100,5.5,8.4)
CALL YNOUNM
CALL XNOUNM
CALL YTICKS(0)
CALL XTICKS(0)
CALL BARPAT(0)
CALL GRAF(0.0,1.0,16.25,0.0,1.0,9.0)
CALL THICRV(3)
CALL BARWID(0.75)
CALL CLUSTR(12.0,0.0)
CALL VBARS(X(1),Y1(1),Y2(1),1)
CALL BARWID(4.25)
CALL VBARS(X(2),Y1(1),Y2(1),1)
CALL BARWID(1.0)
CALL VBARS(X(3),Y1(1),Y2(1),1)
CALL BARWID(1.25)
CALL VBARS(X(4),Y1(1),Y2(1),1)
CALL VBARS(X(5),Y1(1),Y2(1),1)
CALL VBARS(X(6),Y1(1),Y2(1),1)
CALL VBARS(X(7),Y1(1),Y2(1),1)
CALL VBARS(X(8),Y1(1),Y2(1),1)
CALL BARWID(1.0)
CALL VBARS(X(9),Y1(1),Y4(1),1)
CALL VBARS(X(10),Y1(1),Y4(1),1)
CALL VBARS(X(11),Y1(1),Y4(1),1)
CALL VBARS(X(12),Y1(1),Y4(1),1)
CALL BARWID(0.05)
CALL THICRV(5)
CALL HBARS(X1(1),X2(1),Y3(1),1)
CALL BARWID(0.45)
CALL CLUSTR(2.0,0.0)
CALL HBARS(X3(1),X4(1),Y5(1),1)
CALL HBARS(X4(1),X2(1),Y5(1),1)

C
C CHART HEADINGS
C
CALL HEIGHT(0.10)
CALL BASALF("STANDARD")
CALL MESSAG("ACTION\$",100,0.08,9.4)

```
    CALL HEIGHT(0.10)
    CALL MESSAG("RESPONSIBLE$",100,5.05,9.4)
    CALL MESSAG("INITIATION DATES $",100,12.5,9.4)
    CALL MESSAG("COMPLETION DATES $",100,14.5,9.4)
        CALL HEIGHT(0.15)
    CALL MESSAG("FY 81$",100,6.3,9.4)
    CALL MESSAG("FY 82$",100,7.6,9.4)
    CALL MESSAG("FY 83$",100,8.9,9.4)
    CALL MESSAG("FY 84$",100,10.1,9.4)
    CALL MESSAG("FY 85$",100,11.4,9.4)
    CALL MESSAG("DESCRIPTION$",100,2.0,9.05)
    CALL MESSAG("ID$",100,0.20,9.05)
        CALL HEIGHT(0.10)
    CALL MESSAG("OFFICIAL$",100,5.2,9.05)
        CALL HEIGHT(0.10)
    CALL MESSAG("ONDJFMAMJJAS$",100,6.12,9.05)
    CALL MESSAG("ONDJFMAMJJAS$",100,7.37,9.05)
    CALL MESSAG("ONDJFMAMJJAS$",100,8.62,9.05)
    CALL MESSAG("ONDJFMAMJJAS$",100,9.87,9.05)
    CALL MESSAG("ONDJFMAMJJAS$",100,11.12,9.05)
    CALL MESSAG("ESTIMATED$",100,12.37,9.05)
    CALL MESSAG("ACTUAL$",100,13.375,9.05)
    CALL MESSAG("ESTIMATED$",100,14.37,9.05)
    CALL MESSAG("ACTUAL$",100,15.375,9.05)
```

C
C
C

CHART INFORMATION

```
    CALL BASALF("L/CSTD")
    CALL MIXALF("STANDARD")
    CALL HEIGHT(0.095)
    CALL YCOORD(1,OFFSET,1,ANS)
    CALL MESSAG("0.0$",100,XBLK1,ANS)
    CALL MESSAG("(MILESTONE 0 - MISSION ANALYSIS / PROJECT INITIATION$",
*100,XBLK2,ANS)
    ANS3 = ANS - 0.05
    CALL VECTOR(0.80,ANS3,4.95,ANS3,0)
    CALL YCOORD(2,OFFSET,0,ANS)
    CALL MESSAG("0.1$",100,XBLK1,ANS)
    CALL MESSAG(" (D) EVELOPMENT OF ((MENS)$",100,XBLK2,ANS)
    CALL MESSAG(" (NPPSMO)$",100,XBLK3,ANS)
    CALL MESSAG(" (JAN 81)$",100,XLBK12,ANS)
    CALL MARKER(9)
    CALL MARKPL(MM(4),1,SP3)
    CALL CURVPL(ANS,YVALU)
    CALL CURVE(SP3,YVALU,1,1)
    CALL YCOORD(3,OFFSET,0,ANS)
    CALL MESSAG("0.2$",100,XBLK1,ANS)
    CALL MESSAG(" (E) STABLISH (P)ROJECT (O)FFICE$",100,XBLK2,ANS)
    CALL MESSAG(" (NPPSMO)$",100,XBLK3,ANS)
    CALL MESSAG(" (FEB 81)$",100,XLBK12,ANS)
    CALL MARKER(9)
    CALL MARKPL(MM(5),1,SP3)
```

```

CALL CURVPL(ANS,YVALU)
CALL CURVE(SP3,YVALU,1,1)
CALL YCOORD(4,OFFSET,0,ANS)
CALL MESSAG("0.3$",100,XBLK1,ANS)
CALL MESSAG(("(A)PPOINT (PM), (ADP) AND (T)ELECOMMUNICATIONS (M)ARS
*$",100,XBLK2,ANS)
CALL MESSAG("(NPPSMO)$",100,XBLK3,ANS)
CALL MESSAG("(FEB 81)$",100,XLBK12,ANS)
CALL MARKER(9)
CALL MARKPL(MM(5),1,SP3)
CALL CURVPL(ANS,YVALU)
CALL CURVE(SP3,YVALU,1,1)
CALL YCOORD(5,OFFSET,0,ANS)
CALL MESSAG("0.4$",100,XBLK1,ANS)
CALL MESSAG(("(D)EVELOP (P)LAN OF (A)CTION AND (M)ILESTONES$",
100,XBLK2,ANS)
CALL MESSAG("(DTNSRDC)$",100,XBLK3,ANS)
CALL MESSAG("(FEB 81)$",100,XLBK9,ANS)
CALL MESSAG("(FEB 81)$",100,XLBK10,ANS)
CALL MESSAG("(FEB 81)$",100,XLBK11,ANS)
CALL MESSAG("(FEB 81)$",100,XLBK12,ANS)
CALL MARKER(9)
CALL MARKPL(MM(5),1,SP3)
CALL CURVPL(ANS,YVALU)
CALL CURVE(SP3,YVALU,1,1)
CALL YCOORD(6,OFFSET,0,ANS)
CALL MESSAG("0.5$",100,XBLK1,ANS)
CALL MESSAG(("(A)PROVAL OF (MENS)$", 100,XBLK2,ANS)
CALL MESSAG("(NAVDAC)$",100,XBLK3,ANS)
CALL MESSAG("(APR 81)$",100,XLBK12,ANS)
CALL MARKER(9)
CALL MARKPL(MM(7),1,SP3)
CALL CURVPL(ANS,YVALU)
CALL CURVE(SP3,YVALU,1,1)
CALL YCOORD(8,OFFSET,1,ANS)
CALL MESSAG("1.0$",100,XBLK1,ANS)
CALL MESSAG("(MILESTON I) - (CONCEPT DEVELOPMENT PHASE)$",
*100,XBLK2,ANS)
ANS3 = ANS - 0.05
CALL VECTOR(0.80,ANS3,4.35,ANS3,0)
CALL YCOORD(9,OFFSET,0,ANS)
CALL MESSAG("1.1$",100,XBLK1,ANS)
CALL MESSAG("(S)UBMIT (POM) (I)SSUES (PECI) (I)SSUES$",
100,XBLK2,ANS)
CALL MESSAG("(NPPSMO)$",100,XBLK3,ANS)
CALL MESSAG("(OCT 81)$",100,XLBK12,ANS)
CALL MARKER(9)
CALL MARKPL(MM(1),1,SP3)
CALL CURVPL(ANS,YVALU)
CALL CURVE(SP3,YVALU,1,1)
CALL YCOORD(10,OFFSET,0,ANS)
CALL MESSAG("1.2$",100,XBLK1,ANS)

```

```
CALL MESSAG ("(C)ONDUCT (G)ENERAL (F)UNCTIONAL (R)EQUIREMENTS
*(A)NALYSIS$",100,XBLK2,ANS)
CALL MESSAG("(DTNSRDC)$",100,XBLK3,ANS)
CALL MESSAG("(MAR 81)$",100,XLBK9,ANS)
CALL MESSAG("(MAR 81)$",100,XLBK10,ANS)
CALL MESSAG("(FEB 82)$",100,XLBK11,ANS)
CALL MESSAG("(MAR 82)$",100,XLBK12,ANS)
CALL MARKER(9)
CALL VECTPL(MM(5),1,SP1)
CALL VECTPL(MM(7),2,SP2)
CALL MARKPL(MM(7),2,SP3)
ANS2 = ANS + 0.1
VECTOR(SP1,ANS2,SP1,ANS,0)
VECTOR(SP1,ANS,SP2,ANS,0)
CALL CURVPL(ANS,YVAL U)
CALL CURVE(SP3,YVAL U,1,1)
CALL YCOORD(11,OFSSET,0,ANS)
CALL MESSAG("1.3$",100,XBLK1,ANS)
CALL MESSAG ("(D)EFINE (P)ROTOT (E)XPERIMENTS/(D)EMONSTRATIONS
$$",100,XBLK2,ANS)
CALL MESSAG("(DTNSRDC)$",100,XBLK3,ANS)
CALL MESSAG("(APR 81)$",100,XLBK9,ANS)
CALL MESSAG("(APR 81)$",100,XLBK10,ANS)
CALL MESSAG("(JAN 82)$",100,XLBK11,ANS)
CALL MESSAG("(JAN 82)$",100,XLBK12,ANS)
CALL MARKER(9)
CALL VECTPL(MM(6),1,SP1)
CALL VECTPL(MM(4),2,SP2)
CALL MARKPL(MM(4),2,SP3)
ANS2 = ANS + 0.1
VECTOR(SP1,ANS2,SP1,ANS,0)
VECTOR(SP1,ANS,SP2,ANS,0)
CALL CURVPL(ANS,YVAL U)
CALL CURVE(SP3,YVAL U,1,1)
CALL YCOORD(12,OFSSET,0,ANS)
CALL MESSAG("1.4$",100,XBLK1,ANS)
CALL MESSAG ("(D)EVELOP (P)RELIMINARY (E)CONOMIC (A)NALYSIS
$$",100,XBLK2,ANS)
CALL MESSAG("(DTNSRDC)$",100,XBLK3,ANS)
CALL MESSAG("(JUN 81)$",100,XLBK9,ANS)
CALL MESSAG("(JUN 81)$",100,XLBK10,ANS)
CALL MESSAG("(MAR 81)$",100,XLBK11,ANS)
CALL MESSAG("(MAR 82)$",100,XLBK12,ANS)
CALL MARKER(9)
CALL VECTPL(MM(9),1,SP1)
CALL VECTPL(MM(7),2,SP2)
CALL MARKPL(MM(7),2,SP3)
ANS2 = ANS + 0.1
VECTOR(SP1,ANS2,SP1,ANS,0)
VECTOR(SP1,ANS,SP2,ANS,0)
CALL CURVPL(ANS,YVAL U)
CALL CURVE(SP3,YVAL U,1,1)
```

```

CALL YCOORD(13,OFFSET,0,ANS)
CALL MESSAG("1.5$",100,XBLK1,ANS)
CALL MESSAG ("(D)EVELOP (P)RELIMINARY (B)UDGET AND (C)OST (E)STIMATE
$$",100,XBLK2,ANS)
CALL MESSAG("(DTNSRDC)$",100,XBLK3,ANS)
CALL MESSAG("(JUN 81)$",100,XLBK9,ANS)
CALL MESSAG("(JUN 81)$",100,XLBK10,ANS)
CALL MESSAG("(FEB 82)$",100,XLBK11,ANS)
CALL MESSAG("(MAR 82)$",100,XLBK12,ANS)
CALL MARKER(9)
CALL VECTPL(MM(9),1,SP1)
CALL VECTPL(MM(6),2,SP2)
CALL MARKPL(MM(6),2,SP3)
ANS2 = ANS + 0.1
VECTOR(SP1,ANS2,SP1,ANS,0)
VECTOR(SP1,ANS,SP2,ANS,0)
CALL CURVPL(ANS,YVAL U)
CALL CURVE(SP3,YVAL U,1,1)
CALL YCOORD(14,OFFSET,0,ANS)
CALL MESSAG("1.6$",100,XBLK1,ANS)
CALL MESSAG ("(D)EVELOP (P)ILOT (S)YSTEM (S)PECIFICATIONS
$$",100,XBLK2,ANS)
CALL MESSAG("(DTNSRDC)$",100,XBLK3,ANS)
CALL MESSAG("(OCT 81)$",100,XLBK9,ANS)
CALL MESSAG("(OCT 81)$",100,XLBK10,ANS)
CALL MESSAG("(MAR 82)$",100,XLBK11,ANS)
CALL MESSAG("(MAR 82)$",100,XLBK12,ANS)
CALL MARKER(9)
CALL VECTPL(MM(1),1,SP1)
CALL VECTPL(MM(6),2,SP2)
CALL MARKPL(MM(6),2,SP3)
ANS2 = ANS + 0.1
VECTOR(SP1,ANS2,SP1,ANS,0)
VECTOR(SP1,ANS,SP2,ANS,0)
CALL CURVPL(ANS,YVAL U)
CALL CURVE(SP3,YVAL U,1,1)
CALL YCOORD(15,OFFSET,0,ANS)
CALL MESSAG("1.7$",100,XBLK1,ANS)
CALL MESSAG ("(P)REPARE (P)ROJECT (M)ANAGEMENT (P)LAN
$$",100,XBLK2,ANS)
CALL MESSAG("(DTNSRDC)$",100,XBLK3,ANS)
CALL MESSAG("(DEC 81)$",100,XLBK9,ANS)
CALL MESSAG("(JAN 82)$",100,XLBK10,ANS)
CALL MESSAG("(MAR 82)$",100,XLBK11,ANS)
CALL MESSAG("(MAR 82)$",100,XLBK12,ANS)
CALL MARKER(9)
CALL VECTPL(MM(3),1,SP1)
CALL VECTPL(MM(6),2,SP2)
CALL MARKPL(MM(6),2,SP3)
ANS2 = ANS + 0.1
VECTOR(SP1,ANS2,SP1,ANS,0)
VECTOR(SP1,ANS,SP2,ANS,0)

```

```
CALL CURVPL(ANS,YVALU)
CALL CURVE(SP3,YVALU,1,1)
CALL ENDPL(0)
CALL DONEPL
END
```

```
SUBROUTINE VECTPL(MONT,MYEAR,S PLOT)
DIMENSION MONTE(11)
DATA MONTE/3HNOV, 3HDEC, 3HJAN, 3HFEB, 3HMAR, 3HAPR, 3HMAY, 3HJUN,
*3HJUL, 3HAUG, 3HSEP/
IF (MYEAR .EQ. 1) S PLOT = 6.12
IF (MYEAR .EQ. 2) S PLOT = 7.37
IF (MYEAR .EQ. 3) S PLOT = 8.62
IF (MYEAR .EQ. 4) S PLOT = 9.87
IF (MYEAR .EQ. 5) S PLOT = 11.12
IF (MONT .EQ. MONTE(1)) S PLOT = S PLOT + 0.10
IF (MONT .EQ. MONTE(2)) S PLOT = S PLOT + 0.20
IF (MONT .EQ. MONTE(3)) S PLOT = S PLOT + 0.30
IF (MONT .EQ. MONTE(4)) S PLOT = S PLOT + 0.40
IF (MONT .EQ. MONTE(5)) S PLOT = S PLOT + 0.50
IF (MONT .EQ. MONTE(6)) S PLOT = S PLOT + 0.60
IF (MONT .EQ. MONTE(7)) S PLOT = S PLOT + 0.70
IF (MONT .EQ. MONTE(8)) S PLOT = S PLOT + 0.80
IF (MONT .EQ. MONTE(9)) S PLOT = S PLOT + 0.90
IF (MONT .EQ. MONTE(10)) S PLOT = S PLOT + 1.0
IF (MONT .EQ. MONTE(11)) S PLOT = S PLOT + 1.10
RETURN
END
```

```
SUBROUTINE YCOORD(LN,OFFSET,HEADING,ANSWER)
BASE = 8.925
VARYNO = 0.2
ANSWER = BASE - (VARYNO *LN) - OFFSET
IF (HEADING .EQ. 1) OFFSET = OFFSET + 0.1
RETURN
END
```

```
SUBROUTINE MARKPL(MNTH,JYR,SS)
DIMENSION MON(12)
DATA MON/3HOCT, 3HNOV, 3HDEC, 3HJAN, 3HFEB, 3HMAR, 3HAPR, 3HMAY,
*3HJUN, 3HJUL, 3HAUG, 3HSEP/
IF (MNTH .EQ. MON(1)) SS = 6.11
```

```
IF (MNTH .EQ. MON(2)) SS = 6.21
IF (MNTH .EQ. MON(3)) SS = 6.31
IF (MNTH .EQ. MON(4)) SS = 6.41
IF (MNTH .EQ. MON(5)) SS = 6.51
IF (MNTH .EQ. MON(6)) SS = 6.61
IF (MNTH .EQ. MON(7)) SS = 6.69
IF (MNTH .EQ. MON(8)) SS = 6.79
IF (MNTH .EQ. MON(9)) SS = 6.89
IF (MNTH .EQ. MON(10)) SS = 6.99
IF (MNTH .EQ. MON(11)) SS = 7.09
IF (MNTH .EQ. MON(12)) SS = 7.19
IF (JYR .EQ. 2) SS = SS + 1.25
IF (JYR .EQ. 3) SS = SS + 2.50
IF (JYR .EQ. 4) SS = SS + 3.75
IF (JYR .EQ. 5) SS = SS + 5.00
RETURN
END
```

```
SUBROUTINE CURVPL (ALNVAL,RESULT)
RESULT = (ALNVAL/1.11) + 0.03
RETURN
END
```

PLAN OF ACTION AND MILESTONES

ACTION ID	DESCRIPTION	RESPONSIBLE OFFICIAL	FY 81			FY 82			FY 83			FY 84			FY 85			INITIATION DATES		COMPLETION DATES	
			ONDIAFMILJAS	ESTIMATED	ACTUAL	ESTIMATED	ACTUAL														
0.0	MILESTONE 0 - MISSION ANALYSIS/PROJECT INITIATION	NPPSMO NPPSMO NPPSMO DTSRDC NPPDC																JUN 81 FTB 81 FTB 81 FTB 81 FTB 81	FTB 81 FTB 81 FTB 81 FTB 81 FTB 81	FTB 81 FTB 81 FTB 81 FTB 81 FTB 81	FTB 81 FTB 81 FTB 81 FTB 81 FTB 81
0.1	Development of MENS																				
0.2	Establish Project Office																				
0.3	Assign PM NPP, and Telecommunications Home																				
0.4	Develop Plan of Action and Milestones																				
0.5	Report of MENS																				
1.0	MILESTONE 1 - CONCEPT DEVELOPMENT PHASE																				
1.1	Submit RFI Issues FCC Issues																				
1.2	Product General Functional Requirements Analysis																				
1.3	Define Pilot Experiment/Demonstrations																				
1.4	Develop Preliminary Economic Analysis																				
1.5	Develop Preliminary Budget and Cost Estimate																				
1.6	Develop Pilot System Specifications																				
1.7	Prepare Project Management Plan																				

INITIAL DISTRIBUTION

Copies

30	NPPS
12	DTIC

CENTER DISTRIBUTION

Copies	Code	Name
2	1808s	D. Wildy
1	182	A. Camara
1	1821	M. Culpepper
1	1828	M. Gray
1	1828	J. Garner
30	1828	S. Becker
1	522.1	Library (C)
1	522.2	Library (A)

DTNSRDC ISSUES THREE TYPES OF REPORTS

- 1. DTNSRDC REPORTS, A FORMAL SERIES, CONTAIN INFORMATION OF PERMANENT TECHNICAL VALUE. THEY CARRY A CONSECUTIVE NUMERICAL IDENTIFICATION REGARDLESS OF THEIR CLASSIFICATION OR THE ORIGINATING DEPARTMENT.**
- 2. DEPARTMENTAL REPORTS, A SEMIFORMAL SERIES, CONTAIN INFORMATION OF A PRELIMINARY, TEMPORARY, OR PROPRIETARY NATURE OR OF LIMITED INTEREST OR SIGNIFICANCE. THEY CARRY A DEPARTMENTAL ALPHANUMERICAL IDENTIFICATION.**
- 3. TECHNICAL MEMORANDA, AN INFORMAL SERIES, CONTAIN TECHNICAL DOCUMENTATION OF LIMITED USE AND INTEREST. THEY ARE PRIMARILY WORKING PAPERS INTENDED FOR INTERNAL USE. THEY CARRY AN IDENTIFYING NUMBER WHICH INDICATES THEIR TYPE AND THE NUMERICAL CODE OF THE ORIGINATING DEPARTMENT. ANY DISTRIBUTION OUTSIDE DTNSRDC MUST BE APPROVED BY THE HEAD OF THE ORIGINATING DEPARTMENT ON A CASE-BY-CASE BASIS.**

ATE
LMED
8