GEORGIA INSTITUTE OF TECHNOLOGY
OFFICE OF CONTRACT ADMINISTRATION
SPONSORED PROJECT INITIATION

Date: June 9, 1977

Project Title: Navy Mine Warfare Traning Center Computer Programs

Project No: A-1941

Project Director: Ms. E. W. Martin

Sponsor: Systems Engineering Laboratories; 6901 W. Sunrise Blvd.; Ft. Lauderdale, Fla. 33313

Agreement Period: From 1/24/77 Until 9/30/77 Approx.*

Type Agreement: Purchase Order No. 24926-F(Subcontract under U. S. Navy Prime)

Amount: $120,318

Reports Required: Computer Software

Sponsor Contact Person (s):

Technical Matters	Contractual Matters (thru OCA)
W. Maxey Project Manager	R. Keller Buyer

Systems Engineering Laboratories
6901 W. Sunrise Blvd.
Fort Lauderdale, Fla. 33313
Phone: (305) 587-2900

*To be completed within 5 months after receipt of all SEL software specifications.

Defense Priority Rating: N/A

Assigned to: Radar Instrumentation Laboratories (School/Laboratory)

COPIES TO:

Project Director	Library, Technical Reports Section
Division Chief (EES)	Office of Computing Services
School/Laboratory Director	Director, Physical Plant
Dean/Director—EES	EES Information Office
Accounting Office	Project File (OCA)
Procurement Office	Project Code (GTRI)
Security Coordinator (OCA)	Other
Reports Coordinator (OCA) ✓	

CA-3 (3/76)

GEORGIA INSTITUTE OF TECHNOLOGY
OFFICE OF CONTRACT ADMINISTRATION

SPONSORED PROJECT TERMINATION

Date: 9/29/78

Project Title: Navy Mine Warfare Training Center Computer Program

Project No: A-1941

Project Director: MS. E. W. Martin

Sponsor: Systems Engineering Laboratories

Effective Termination Date: Immediately (Memo dtd. 9/21/78, DEW/EWM)

Clearance of Accounting Charges: ASAP

Grant/Contract Closeout Actions Remaining:

X Final Invoice and Closing Documents (Subcontract under NAvy prime)
_ Final Fiscal Report
X Final Report of Inventions
X Govt. Property Inventory & Related Certificate
_ Classified Material Certificate
_ Other ____________

Assigned to: Radar Instrumentation (School/Laboratory)

COPIES TO:

Project Director	Library, Technical Reports Section ✓
Division Chief (EES)	Office of Computing Services
School/Laboratory Director	Director, Physical Plant
Dean/Director—EES	EES Information Office
Accounting Office	Project File (OCA)
Procurement Office	Project Code (GTRI)
Security Coordinator (OCA)	Other Mr. Borchert/GTRI
Reports Coordinator (OCA)	

CA-4 (3/76)

A-1941

ENGINEERING EXPERIMENT STATION
GEORGIA INSTITUTE OF TECHNOLOGY • ATLANTA, GEORGIA 30332

7 February 1977

Systems Engineering Laboratories
6901 West Sunrise Boulevard
Fort Lauderdale, Florida 33313

Attention: Mr. Bill Maxey, Project Leader
Systems Engineering

Subject: SEL Purchase Order 24246S, EES/GIT Project No. A-1941
"Navy Mine Warfare Training Center Computer Programs"

Gentlemen:

Attached please find the detailed work statement, a milestone schedule, and a set of acceptance criteria for the Navy Mine Warfare Training Center Computer Programs Project. The milestone schedule predicts delivery six months after machine delivery and training, as we had previously negotiated.

The attached document also points out a number of potential difficulties which require action by SEL. Briefly and in no particular order,

- the amount and kind of EES-provided documentation for each delivered program must be specified,

- the source code for the West Point Graphics Compatability System (GCS) and complete details of the Imlac-PDS4/GCS interactions should be available at the start of work,

- SEL guidance may be needed to help solve the problem of the use of the number -0 in calling sequences,

- SEL guidance may be needed to determine how to convert the WAIT routines in the Two-Terminal Routines,

- the GCS seemingly does not have the ability to control two graphics terminals concurrently; the nature of this requirement should be clarified,

- getting the Navy to provide more program documentation and a layout diagram for the database would reduce the risk of unforeseen difficulties in the conversion process,

- SEL guidance is needed on the problem of determining the correctness of the database produced by program B1; EES proposes a small Database Dump and Verification Program for this purpose,

- pictures or movies of graphics programs G1 and G2 in operation would be of substantial value in converting these programs,

- the weight placed on accuracy of numerical computations is still unclear and must be clarified, likely with the Navy.

Sincerely

Gerald N. Cederquist
Project Director

GNC/db

Enclosures

ENGINEERING EXPERIMENT STATION
GEORGIA INSTITUTE OF TECHNOLOGY • ATLANTA, GEORGIA 30332

4 February 1977

MEMORANDUM

TO: Bill Maxey
Systems Engineering Laboratories

FROM: Gerald N. Cederquist

SUBJECT: Work Statement, Milestone Schedule, and Acceptance Test Criteria for Navy Mine Warfare Training Center Computer Programs

DETAILED WORK STATEMENT

Figure 1 shows the computer programs and data files to be provided by EES under this contract. In addition I show how both the Mass Storage Routines and the West Point Graphics Compatibility System interface to these computer programs.

General Approach

First I will cover the general approach we would like to use for each conversion, and then I will go into more detail on specific considerations for each of the computer programs. We currently feel that additional documentation for each program would be useful since it would reduce the risk of improperly converting a program and having to redo the job at some later time. Our eight-step conversion approach is as follows:

1. Generate data-flow and control-flow trees using both cross-reference listings from the compiler and subroutine-calling information provided to us by Tom Heron. These data- and control-flow trees will allow us to quickly assess the impact of changes in either data or control structures in one routine upon other routines in a program. Perhaps these trees are more in the nature of documentation which should have been done by the Navy. However, we feel that if we have this information before we begin converting a program, we will end up making fewer mistakes in the long run.

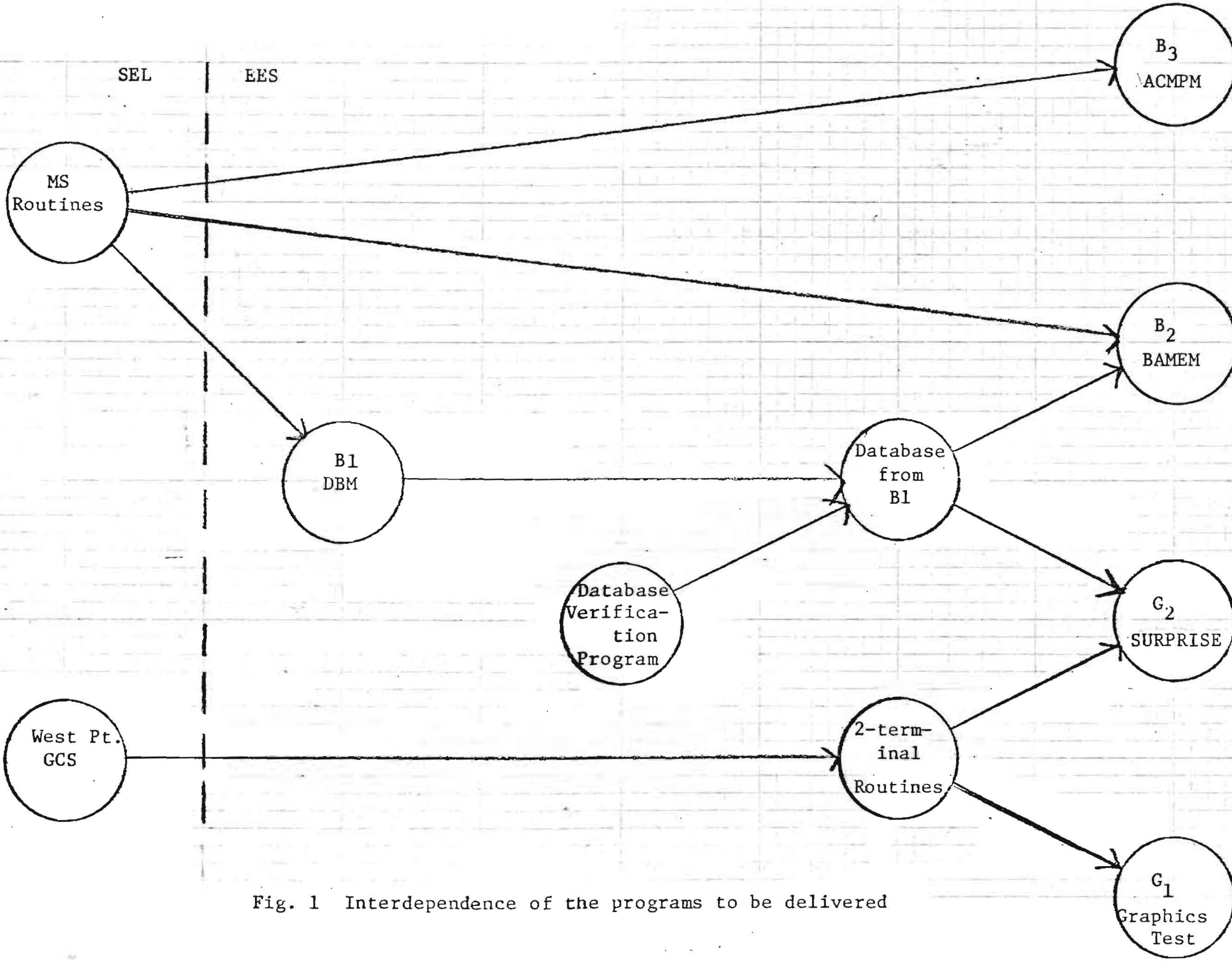


Fig. 1 Interdependence of the programs to be delivered

2. Analyze error comments from the compiler and propose corrections. In some cases the errors will be trivial to correct. However, some errors will be due to machine dependencies and will be more difficult to correct.
3. Find and analyze statements which the compiler did not flag, but which are dependent upon
   - the interactive graphics system
   - the knowledge that there are 10 characters per word
   - other dependencies on the CDC 6600 architecture and on vagaries of the CDC FTN4 compiler.

   Locating and analyzing the intent of some of these statements may be expedited by contact with the Navy. Otherwise, a line by line analysis of each one of the programs will likely be necessary.
4. Fix statements to be changed, leaving an audit trail. Since we anticipate that more than one iteration will likely be necessary to convert a program, we intend leaving all versions of the program on the disc during the conversion process.
5. Add statements to print the beginning and ending time of execution for each program, as required in Section 6 of Attachment 2.
6. Generate a test procedure for each program. This may require contact with the Navy, particularly in the case of the graphics routines since they have no printed output with which to compare our results.
7. Conduct tests according to the test procedure, logging all bugs and leaving an audit trail of statements altered to remove the bugs.
8. Document the conversion process for the program, showing the approach used to overcome machine dependent portions of code.

Specific Considerations for Each Program

The Two-Terminal Routines determine the critical path of the project schedule (see below). Although only 47 out of a total of 68 routines need be converted, the conversion job will be significant nonetheless. We have chosen to take approach 2 of Section 1.4 of Attachment 2; that is, we will provide new subroutines which maintain the old Two-Terminal calling sequences. These new routines will utilize the West Point Graphics Compatibility System to do the actual graphics display operations. Although we have quickly perused the GCS documents, it is still unclear exactly what the mapping is between CDC's machine-dependent IGS routines and the machine-independent GCS routines. We feel it would be helpful to have as soon as possible in-house both the source code for the GCS and complete details on how the IMLAC PDS-4 display will be run by the GCS. We have identified at least three problems in converting the Two-Terminal Routines:

1. The Two-Terminal Routines use the number -0 in their calling sequences quite extensively. This number cannot be represented in the SEL machine. Often in CDC machines, the use of -0 indicates that the underlying data structure really consists of both a switch and a value. If the switch is set false, then the corresponding value has no meaning. On the other hand, the switch is set to true to indicate that the corresponding value has meaning. Thus the value of -0 is used as a sentinel to indicate that there is no valid datum in a variable. If the number 0 is desired as a datum, then the value +0 is used in the CDC machine. The use of -0 is highly machine-dependent and any two's complement machine will not allow it. We may be able to find an alternative value for a sentinel within the SEL floating-point number scheme. Alternatively we may have to modify the Two-Terminal Routine calling sequences to include both a switch and a datum-value parameter everywhere -0 can currently appear.

2. Currently it is unclear what the mapping is between the WAIT routines on the CDC machine and the services provided by the Real Time Monitor. We may need help from SEL to answer this question.
3. On first reading, we cannot find the mechanism which the Graphics Compatibility System uses to send graphical output to more than one graphics device, either display or plotter. How can we overcome this problem?

G1

The code in G1 is heavily dependent upon the CDC Interactive Graphic System. Thus we feel that G1 will make a good test program for the Two-Terminal Routines. It may be worth pointing out to the Navy that the user interface presented by the West Point routines is so much more simple than that presented by the machine-dependent IGS routines that they may be better off using the West Point routines directly. I am presuming here, of course, that the outputs of the acceptance tests are of prime importance to the Navy, rather than how they are produced.

G2

G2 of course is also highly dependent upon the Interactive Graphics System. However, since G2 accesses the data base produced by program B1, there is a significant amount of database-dependent coding within G2. This coding may have to be extensively modified once it is located, depending upon how strongly it uses the fact that there are ten characters in a word. Also for G2 we must produce some sort of routine which allows hard copy of the display buffer on either the flatbed plotter or the electrostatic plotter. In connection with problem 3 cited above, this capability may or may not be already present in the West Point Routines.

B1

Not only must we convert the B1 program, but we must also convert the data base which it produces. We feel that it would be extremely useful to obtain from the Navy the layout of the database, to determine the packing schemes used and the location of the information within the database. If we have to derive this information by reading program B1, it will take much longer than obtaining the information from the Navy directly. Also, as indicated on Figure 1, we propose that we build a Database Dump and Verify Program to be used to test the database produced from the conversion of program B1. Note that the database feeds into both B2 and G2; if the answers produced by these programs are incorrect, it may very well be the result of bad data within the database. Consequently, we feel that constructing a program to verify the data base minimizes our risk on B2 and G2.

B2

Since B2 operates on data produced by B1, we may need to perform extensive modifications on database-dependent code within B2. In addition we may run into computational accuracy problems as we have outlined to you in the past (more on this below).

B3

The most significant error in B3 appears to be the 35,000-word COMMON region needed by the main program. You proposed a fix for this error, which we must have to make B3 run. In addition, since B3 does a significant amount of numerical computation there may be problems with computational accuracy as well.

MILESTONE SCHEDULE

Our proposed milestone schedule for this project is shown in Figure 2. Milestones are shown within circles and interdependencies are shown by the arrows between circles. The time estimates for each phase of the project were produced by reading each program to be converted, noting the number of errors and machine dependencies, and estimating the amount of time it would take both to convert the program and to test the converted program.

The schedule as drawn in Figure 2 pushes all delivery dates as far down the schedule as possible. Note however, that it would be possible to bring

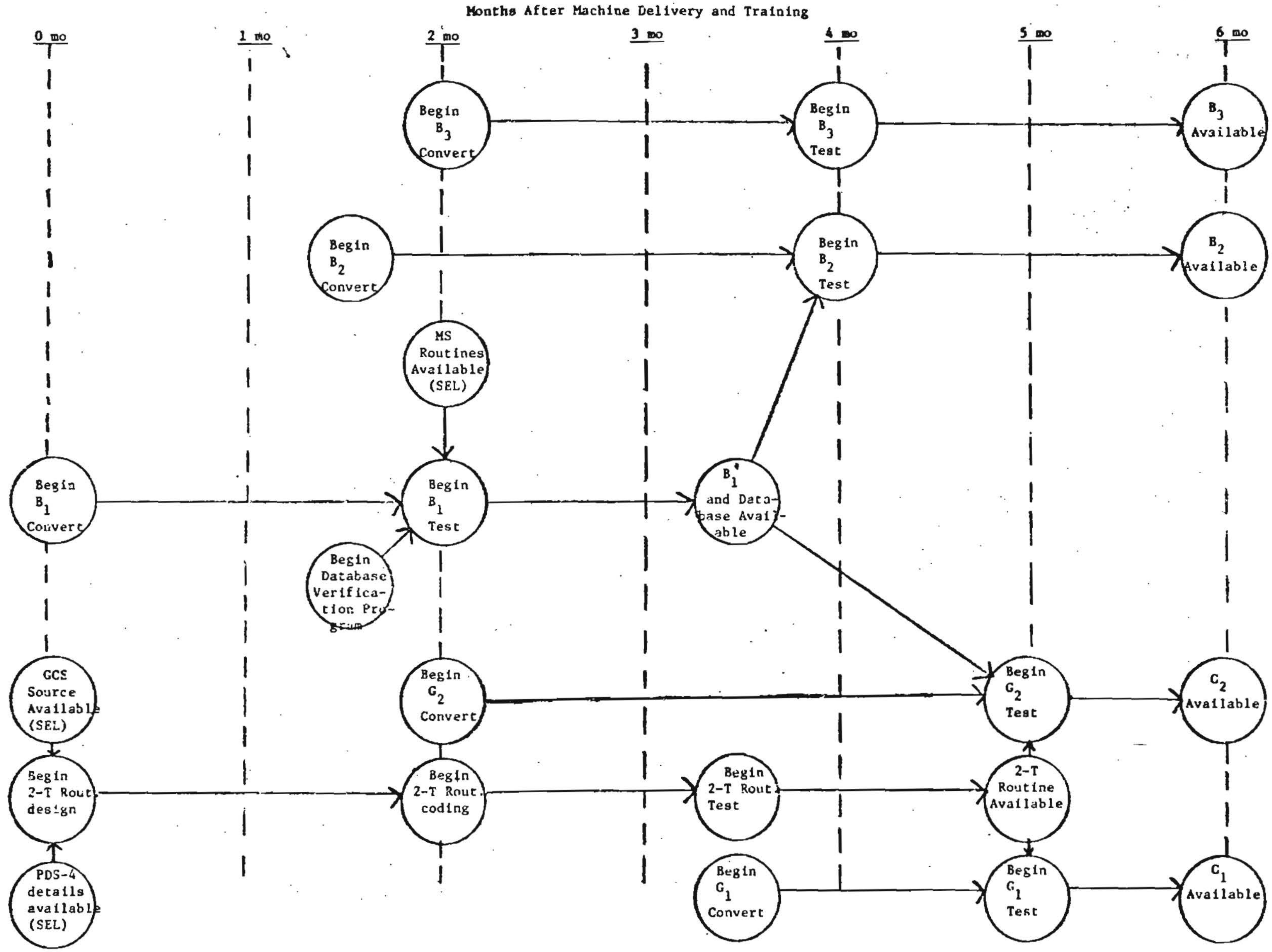


Figure 2. Milestone Schedule

some of the deliveries back toward the beginning of the schedule. For example, program B3 has no dependencies on any other program we are to do. Consequently work on B3 could begin as early as month 0. The schedule for B2, for example, could also be pushed back toward the beginning of the project by one-half month.

The effort on B1 starts the project at month 0, allowing about 1/2 month extra in the conversion process for people to gain familiarity with the SEL machine. Note that the Mass Storage Routines which SEL is to provide need not be available until B1 testing begins. Also note we anticipate that writing the Database Verification Program will require only 1 programmer for one-half month; this is under the presumption that we obtain the database layout from the Navy. The database from B1 must be available before B2 testing can begin; consequently the earlier the database is available, the earlier B2 itself becomes available.

The critical path of the schedule is established by the effort on the Two-Terminal Routines. Note that we have included two milestones at the very beginning of the Two-Terminal Routine effort which rely upon input from SEL. Because we anticipate some difficulty with the Two-Terminal Routines, we have allowed a two month design phase using senior-level people at the beginning of the effort. By the time coding and testing is done, we foresee that the Two-Terminal Routines will become available only at the beginning of month 5. Consequently, testing of both the graphics programs cannot begin until month 5. It may make sense, however, to begin the conversion of these graphics programs before the times shown in the schedule. The one-month time span which is shown for testing the graphics programs presumes that we have access to pictures or movies of programs G1 and G2 in operation, so that we know exactly what G1 and G2 are supposed to do (more on this below).

<u>ACCEPTANCE CRITERIA</u>

We recognize your need to meet the acceptance criteria specified in the Navy's Attachment 2. In general we think we can live with these, with certain exceptions detailed below. In our examination of the programs to date, we have found a number of places where the execution time of a program could possibly be significantly reduced. However, we do not intend to modify the programs to reduce execution time unless they cannot run fast enough to pass the Navy's acceptance test.

Two-Terminal Routines

In testing the Two-Terminal Routines, we anticipate that we will build small test programs which will exercise a small number of paths within the routines; all these tests will be performed with only one terminal. When the Two-Terminal Routines successfully run with our own small check-out programs, we will then begin testing them using G1. Our feeling is that the Two-Terminal Routines should be judged acceptable if both G1 and G2 run successfully insofar as their computer-graphics operations are concerned. We will need the support of SEL in order to test the Two-Terminal Routines to see that they can indeed drive two terminals independently. Will this phase of the check-out have to be done in Charleston?

G1

We feel that G1 should be accepted insofar as speed is concerned if it can meet the Navy's timing criteria established in Section 3.2 of Attachment 2. Insofar as the correctness of G1 is concerned, if we stipulate the correctness of the Navy's program, we can simply observe the display to see if G1 is operating properly. For example, we will count the 3,000 characters on the screen to make sure that all 3,000 are present. Similarly we will determine the correctness of the 10-sided-figure portion of the program by inspection. Our feeling is however, that it is highly unlikely that we will be able to reliably count 700 random, 2-inch vectors on the screen. Again it would be very helpful to see G1 in operation to determine exactly what its behavior is supposed to be.

G2

We propose that G2 be accepted insofar as speed is concerned if it can meet the Navy's timing criteria given in Sections 3.4 and 4.1 of Attachment 2.

We feel the accuracy requirement for G2 is not clearly specified. The amount of precision required by Section 5 of Attachment 2 is much too large to be determined by looking at the graphical output. Since we do not know what the output of G2 is supposed to look like, we do not know if any numerical values are displayed on the screen. Presumably these numerical values would be subject to the requirements of Section 5. Again, it would be very useful for us to see G2 in operation and thus determine exactly what it is supposed to do and what its outputs are supposed to look like.

B1

We propose that B1 be accepted insofar as speed is concerned if it meets the timing criteria given by the Navy in Section 3.1 of Attachment 2. We further propose that the accuracy and correctness of B1 be accepted on the basis of the answers produced by our proposed Database Verification Program.

B2 and B3

We propose that B2 and B3 both be accepted insofar as speed is concerned if they meet the timing criteria put forth by the Navy in Sections 3.3, 3.4, and 4.1 of Attachment 2. Plainly the accuracy specifications of Section 5 of Attachment 2 are meant to apply to B2 and B3. However, as we have discussed with you before, there may be intrinsic limitations in 32-bit floating-point arithmetic which preclude achieving this accuracy.

There is an additional accuracy problem in B2. Recall that B2 is a simulation program using Monte-Carlo methods which evaluates mine field plans. Extensive use is made of the random number generator RANF, provided by CDC. Either EES or SEL will provide a random number generator which simulates the functions of RANF. The difficulty is that unless one does a complete simulation in software of CDC's random number generator (which of course substantially slows down the execution of B2), then one cannot guarantee that the same sequence of random numbers will be produced from the SEL random number generator as produced by RANF. Consequently, the portion of the solution space traversed by B2 running on the SEL machine will be different from that of B2 running on the CDC machine. Since the Monte-Carlo simulation is inherently statistical in nature, we cannot offer any guarantee that the statistical averages from a different portion of the solution space (SEL machine) will be the same as the averages from the original portion of the solution space (CDC machine). Consequently we feel there is no way in which either EES or SEL can be expected to meet the accuracy requirement of Section 5 for those answers of B2 which are statistical in nature.

A-1941

ENGINEERING EXPERIMENT STATION

GEORGIA INSTITUTE OF TECHNOLOGY • ATLANTA, GEORGIA 30332

17 October 1977

Systems Engineering Laboratory
6901 West Sunrise Boulevard
Fort Lauderdale, Florida 33313

Attention: Bill Maxey
Doug Beard

Subject: EES/GIT Project A-1941
"Navy Mine Warfare Training Center Computer Programs"
Progress/Status Report, 15 September 1977 through
12 October 1977.

Gentlemen:

The attached two documents comprise a Progress/Status Report for Project A-1941 covering the period from 15 September through 12 October 1977.

ATTACHMENT I is a milestone chart with markers denoting our current status. Appended to the chart is a brief statement relating to each of the numbered milestone markers.

ATTACHMENT II is a plot of our planned versus actual expenditures through 12 October 1977.

Should you have any questions concerning the contents of these attachments, please do not hesitate to call.

Sincerely,

Edith W. Martin
Head, Software Development Branch

EWM/am

Attachments

cc: F. B. Dyer
J. Wilson, OCA
File

ATTACHMENT I

1. The conversion of B3 has been completed, however, it is still not currently obtaining the same answers as the Benchmark. The program has been converted to use double precision variables, with initial resulting values worse than the single precision version. Some question about the validity of the originally supplied Benchmark programs has been raised and is being examined by Navy personnel. It appears that B3 requires a data base (documentation says it does not), and B2 does not use one (documentation says it does).

2. Program B2 is currently under test. All software development and testing is proceeding under RTM revision 6 satisfactorily. A modification to the FORTRAN routine system has been made which allows 15 blocking buffers. This modification has allowed us to resolve some of the major problems we have been having with the file system.

3. Considering errors in dimension statements and sections of the programs that cannot be reached, there is a suspicion that the Benchmark programs supplied to SEL and GIT never executed successfully on the CDC machine. It appears that we are debugging rather than converting the supplied Benchmark programs.

4,6. Program G1 and G2 have progressed as far as is possible without the two terminal routines.

5. Testing of the two terminal routines are proceeding as best as can be expected without the modified GCS and TIS programs. We have received a preliminary version of TIS from SSS which did not work as well as the Imlac supplied TIS and had no observable additional features. Progress with Imlac's TIS-4 is considerably slower than anticipated due to missing features. Details follow:

   Implementation of the two-terminal routines has been delayed primarily because of the lack of a useable TIS-4 Imlac handler. Imlac's TIS-4 malfunctions when an attempt is made to use structures. This problem was verified by Kathy Hersh of Imlac. Modifications to this package are being written by SSS. The current modified version malfunctions when frames, structures, or alphanumers are used. As a result it has become difficult to effectively test the following functions:

   1. pick queue
   2. light registers
   3. text retrieval
   4. tracking cross (code not yet implemented by SSS)
   5. alphanumerics

   Most of the two-terminal routines have been debugged (one-by-one) by printing the various arrays affected and verifying proper actions.

   A serious communications "glitch" exists between the Imlac & SEL. This often results in I021 (unrecoverable I/O error) when input is attempted from the Imlac.

Please know that our sincerest, best efforts are being exerted to expeditiously convert the Navy Mine Warfare Training Programs.

ATTACHMENT I

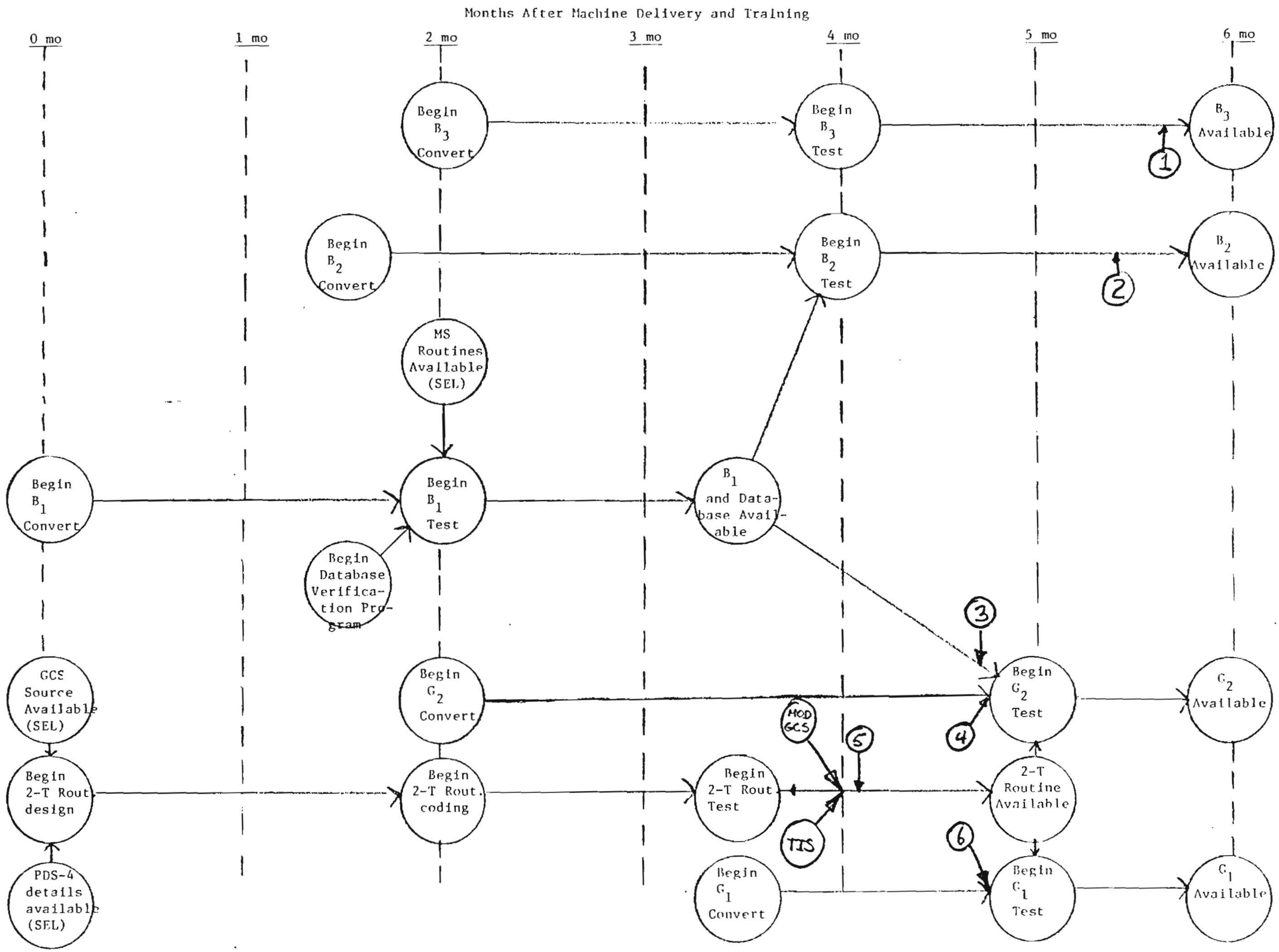
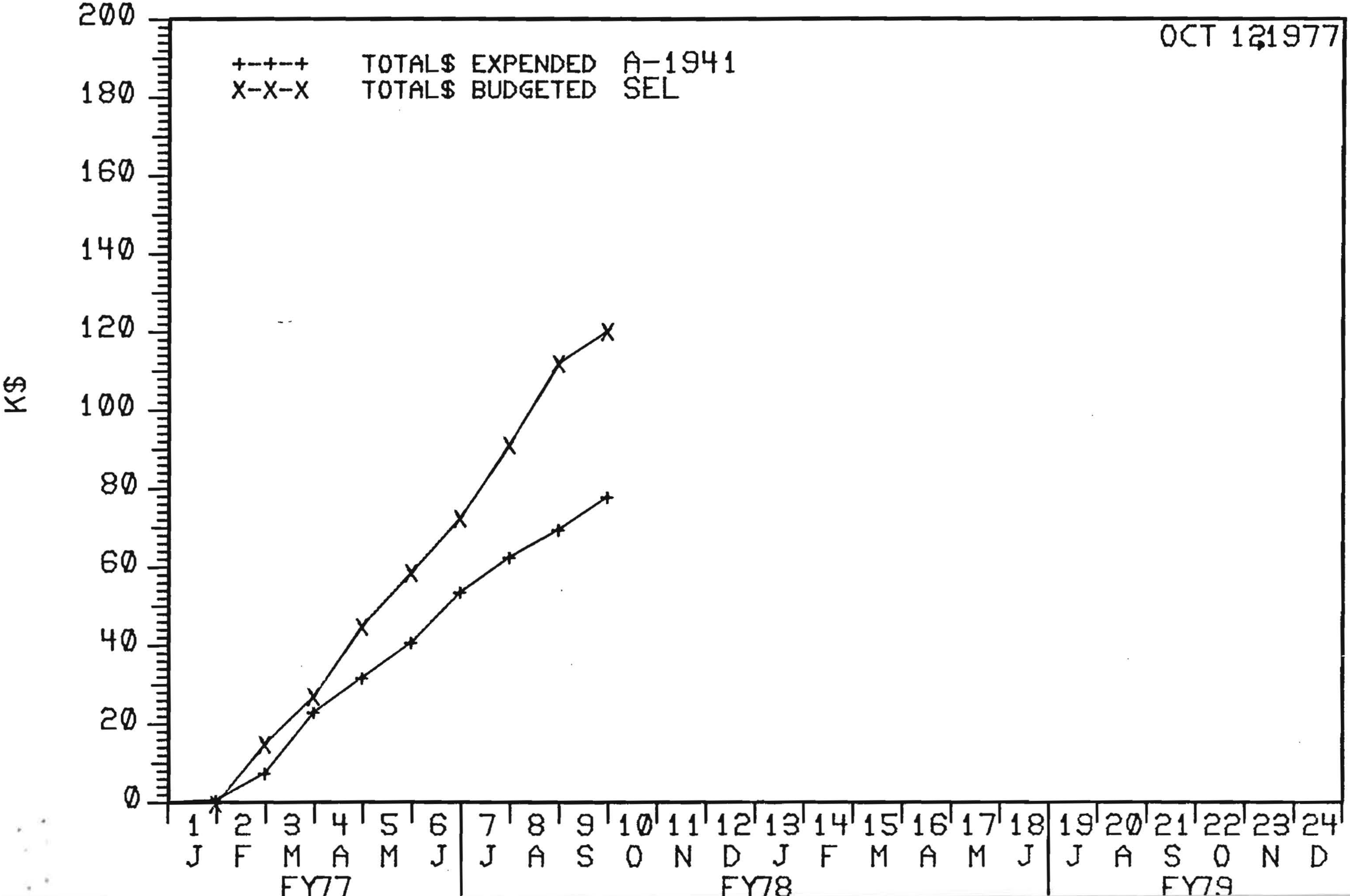


Figure 2. Milestone Schedule


OCT 12 1977
+-+-+ TOTAL$ EXPENDED A-1941
X-X-X TOTAL$ BUDGETED SEL
K$
200
180
160
140
120
100
80
60
40
20
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
J F M A M J J A S O N D J F M A M J J A S O N D
FY77
FY78
FY79

ATTACHMENT 1

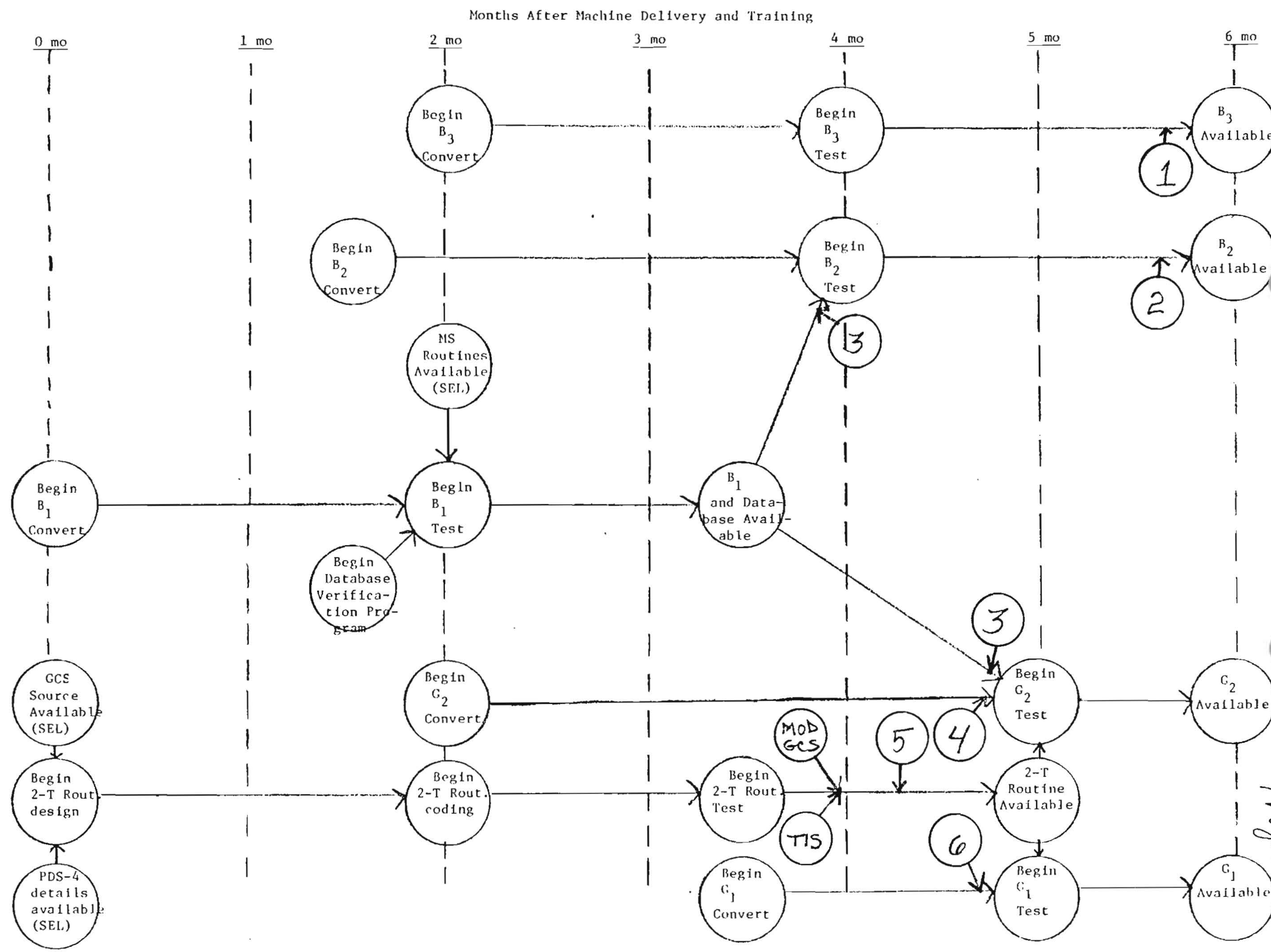


Figure 2. Milestone Schedule

A-1441
Progress rept #3

ATTACHMENT I

1. DFM4 is to be run under Level Six this week. No complications are anticipated.

2. Additional COMMONBLOCK and DIMENSION errors were found in program BAMEN. These were corrected. The program thereafter proceeded further than before; however, other problems remain. As a result of the last meeting with the Navy and SEL on October 17, 1977 at Charleston, we were supplied with the latest versions of BAMEN and ACMPM by Charleston. We are still awaiting official approval to use these versions of the BAMEN and ACMPM programs in place of the former benchmark. Assuming that there will be no problems in regard to this, we are proceeding as agreed. The newer versions are not yet completely operative on our SEL system. Hopefully, the installation of these programs will be completed within a week.

3. RE: Problems with IMLAC TIS handler modifications and its interface to G.C.S. and the two-terminal routines.

   Recently Tim Yuknavitch and Roy Deere of S.S.S. visited EES in an effort to resolve some of the problems associated with their modifications to the TIS 4.I IMLAC display handler. Following is a list of problems which still have not been completely resolved.

   a. Structures in the original TIS 4.I are displayed as they are defined. This is unacceptable. EES requests that S.S.S. modify TIS so that structures are displayed only when invoked. EES has been able to invoke structures with the original TIS, but not with the S.S.S. modified version.

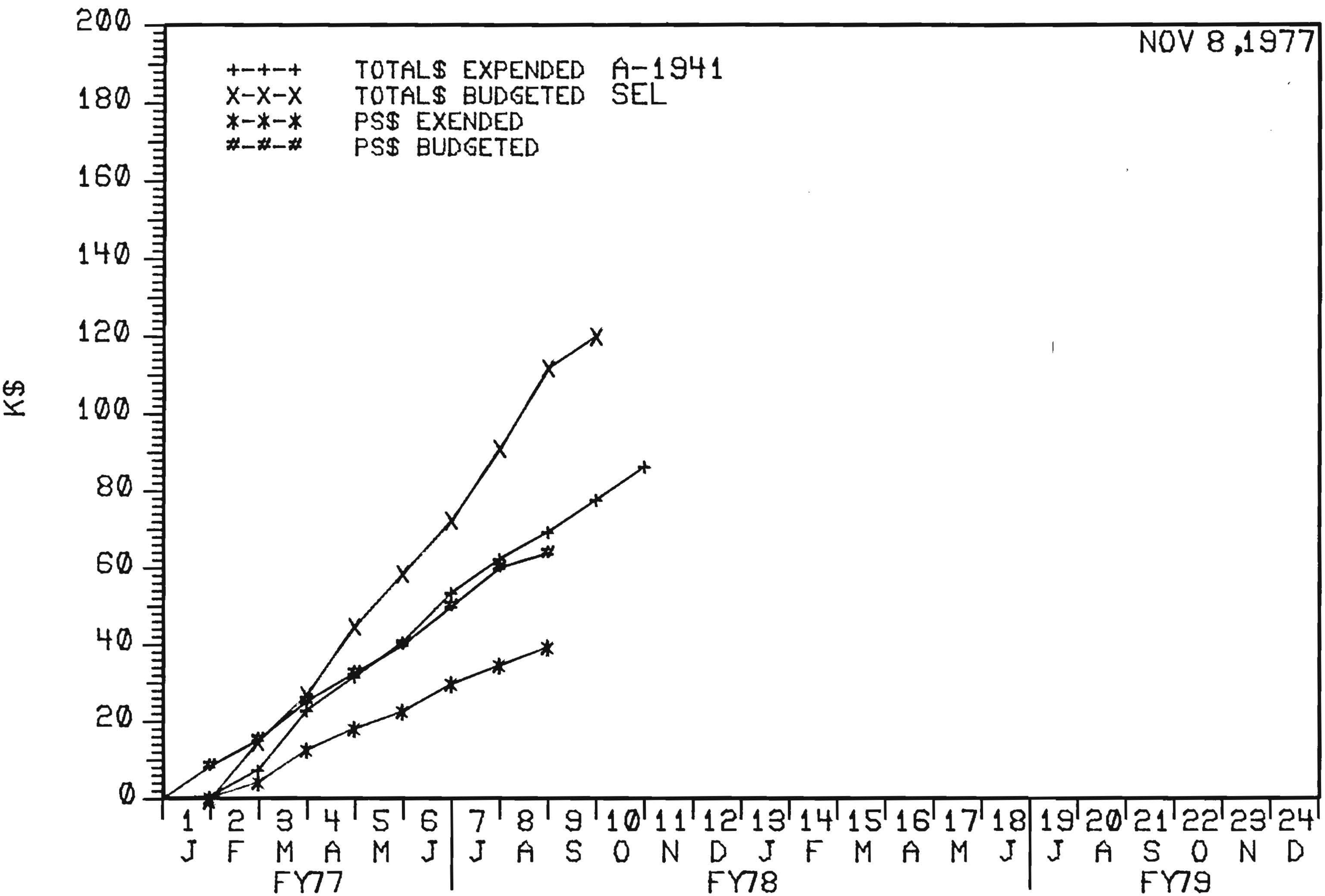
   b. The modified version of TIS malfunctions whenever an attempt is made to use alphanumerics. Quite often oscillation of images on the screen occurs when alphanumerics are used.

   c. A problem exists when the tracking cross is active and it is moved close to a light pen sensitive item. Unpredictable picks may occur since there may be ambiguity as to whether the tracking cross or another item is being picked. EES requests that S.S.S. modify the handler so that when the tracking cross is active and a pick occurs a check is first made to see if the pick occured within a small area around the tracking cross. This should be a square of the approximate size of the cross itself. Only if the pick occurs within this area should the program assume a tracking cross hit, otherwise the standard algorithm for pick items should be used.

   d. EES has experienced problems with both the modified and unmodified versions of TIS when multiple frame definitions are attempted. The problem can be attributed to a communication problem between the SEL and IMLAC. The original TIS works correctly when time delays are inserted between frame definitions. Unusual results have been observed with the modified version when multiple frame definitions are attempted. Additional, unwanted, lines may appear on the screen and/or portions of the display may be missing. S.S.S. witnessed this problem while they were visiting EES. They attribute it to the communications problem.

4. For your information we would like to stress the impact of delays in delivery of the TIS routines. i.e., Further work on G1 and G2 is not possible without the completed TIS. There is a minimum of time required to convert these programs once the TIS routines are received. The TIS continues to define the critical path.

5. Pursuant to discussions between D. E. Wrege and SEL regarding CALCOMP hardcopy capabilities for the BROOMALL, we feel we can satisfy the Navy requirements. This effort should require approximately two weeks of development time prior to checkout at Charleston. Checkout should be completed in two to three days providing the BROOMAL documentation is correct for the system as installed by SEL. The funds available through the existing SEL-GTRI contract should be sufficient to cover this task as well as the current work order, however, a written request for this additional work is desired. Upon receipt of said request work will proceed.



NOV 8,1977
+-+-+ TOTAL$ EXPENDED A-1941
X-X-X TOTAL$ BUDGETED SEL
--* PS$ EXENDED
#-#-# PS$ BUDGETED
K$
0
20
40
60
80
100
120
140
160
180
200
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
J F M A M J J A S O N D J F M A M J J A S O N D
FY77
FY78
FY79

IMLAC TWO TERMINAL ROUTINES (ITTR)

INTERACTIVE GRAPHICS SYSTEM

CONTENTS

IMLAC TWO TERMINAL ROUTINES (ITTR)

INTRODUCTION

AN INTERACTIVE GRAPHICS SYSTEM IS A POWERFUL TOOL FOR ACTIVE ANALYSIS AND SOLUTION OF PROBLEMS. THE USER CAN CREATE, DISPLAY, STORE, RETRIEVE, AND MODIFY GRAPHICS FORMS USING THE IMLAC TWO TERMINAL ROUTINES. ITTR ALLOWS TWO GRAPHICS TERMINALS TO INTERACT WITH A SINGLE APPLICATION PROGRAM.

THE IMLAC TWO TERMINAL ROUTINES ARE DESIGNED SO THAT THE USER MAY WRITE GRAPHICS APPLICATION PROGRAMS IN FORTRAN WITHOUT CONCERNING HIMSELF WITH THE MECHANICS OF THE ACTUAL GRAPHICS DISPLAY DEVICE. THE ITTR SOFTWARE CONSISTS OF A LIBRARY OF SUBROUTINES THAT PROVIDE SIMPLIFIED ACCESS TO THE GRAPHICS HARDWARE WITHOUT LIMITING APPLICATIONS OR DATA STRUCTURES.

THE SYSTEM CAN HANDLE PROBLEMS THAT:

1. CAN BE REPRESENTED IN GRAPHIC, GEOMETRIC, OR SYMBOLIC FORM (E.G. SCHEMATICS, LAYOUTS, DIAGRAMS, AND TRAJECTORY)

2. CAN BE DESCRIBED USING MATHEMATICAL FUNCTIONS

3. REQUIRE HUMAN INTERVENTION

GENERAL SOFTWARE OPERATION

INTERACTIVE GRAPHICS SOFTWARE OPERATES AS TWO SEPARATE BUT COMMUNICATING GROUPS OF ROUTINES. ONE SET OF ROUTINES OPERATES IN THE CENTRAL (SEL 32) COMPUTER WHILE THE OTHER OPERATES IN THE REMOTE DISPLAY COMPUTER (IMLAC PDS-4) SEE FIGURE 1.

CENTRAL COMPUTER SOFTWARE

THE CENTRAL COMPUTER SOFTWARE (ITTR) CONSISTS OF A

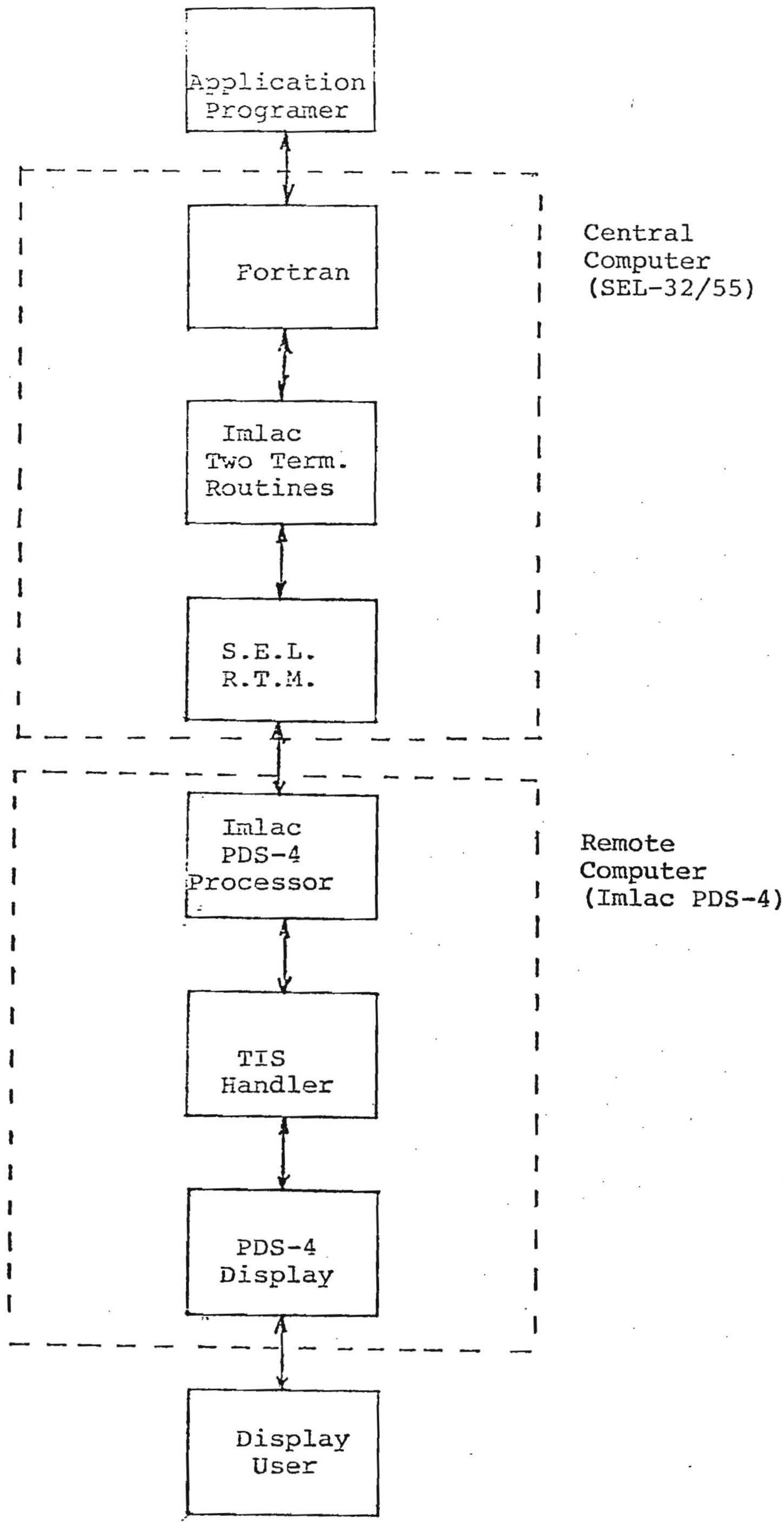

Application Programer
Fortran
Imlac Two Term. Routines
S.E.L. R.T.M.
Central Computer (SEL-32/55)
Imlac PDS-4 Processor
TIS Handler
PDS-4 Display
Remote Computer (Imlac PDS-4)
Display User

Figure 1

LIBRARY OF FORTRAN CALLABLE SUBROUTINES WHICH ARE WRITTEN IN FLECS (SEE APPENDIX V). THE LIBRARY WAS DESIGNED TO PERMIT REAL-TIME USE OF A SEL-32 BY A GRAPHICS CONSOLE OPERATOR.

THE IMLAC TWO TERMINAL ROUTINES FUNCTION NEARLY IDENTICAL TO THE TWO TERMINAL ROUTINES USED AT THE NAVY MINE WARFARE TRAINING CENTER AT DALGREN, VA. THE FEW MINOR DIFFERENCES ARE NOTED IN APPENDIX I. ITTR ARE MACHINE DEPENDENT AND ARE USABLE ONLY WITH A SPECIAL TIS4 IMLAC HANDLER AND WITH THE SEL 32.

THE ITTR PACKAGE IS DESIGNED TO EMULATE MOST FEATURES OF THE C.D.C. 274 INTERACTIVE GRAPHICS SYSTEM, WHILE MAINTAINING THE USER INTERFACE DEFINED BY THE ORIGINAL TWO TERMINAL ROUTINES.

REMOTE COMPUTER SOFTWARE

THE REMOTE PORTION OF THE INTERACTIVE GRAPHICS SOFTWARE OPERATES ON ILS (INTERNAL LOGICAL STRUCTURE) COMMANDS TRANSMITTED BY THE SEL-32 HOST. THIS SOFTWARE, CALLED 'TIS', EXECUTES ON AN IMLAC PDS-4 DISPLAY SYSTEM.

FUNCTIONAL DESCRIPTION OF ITTR

THE FOLLOWING IS A TECHNICAL DISCUSSION OF THE INTERNAL STRUCTURE OF THE IMLAC TWO TERMINAL ROUTINES. THE DISCUSSION ASSUMES THAT THE READER IS CONVERSANT WITH THE TWO TERMINAL ROUTINE USER CALLS AND HAS A WORKING KNOWLEDGE OF FORTRAN.

THE IMLAC TWO TERMINAL ROUTINES CONSIST OF TWO PARTS. THE MAJORITY OF THE ITTR MODULES ARE CONCERNED WITH BUILDING GRAPHICS DATA STRUCTURES AND THEN TRANSMITTING THEM TO THE GRAPHICS CONSOLE. EQUALLY IMPORTANT ARE THE INTERACTIVE FUNCTIONS AVAILABLE IN ITTR. ADVANCED QUEUEING TECHNIQUES ALLOW FLEXIBILITY IN DEFINING USER INTERACTIONS WITH AN APPLICATIONS PROGRAM. FIGURE 2 IS A SIMPLIFIED FLOW DIAGRAM OF THE OPERATION OF ITTR.

BUILDING GRAPHICS DATA STRUCTURES

THE MAJORITY OF ITTR MODULES BUILD GRAPHICS DATA STRUCTURES USING SUB-ITEM DESCRIPTION FLAGS. THESE FLAGS DETERMINE THE TYPE OF STRUCTURE TO BE SENT TO THE GRAPHICS CONSOLE. THEY CAN FLAG REQUESTS FOR A CIRCLE, A LINE, TEXT, POSITION DEFINITION, OR A

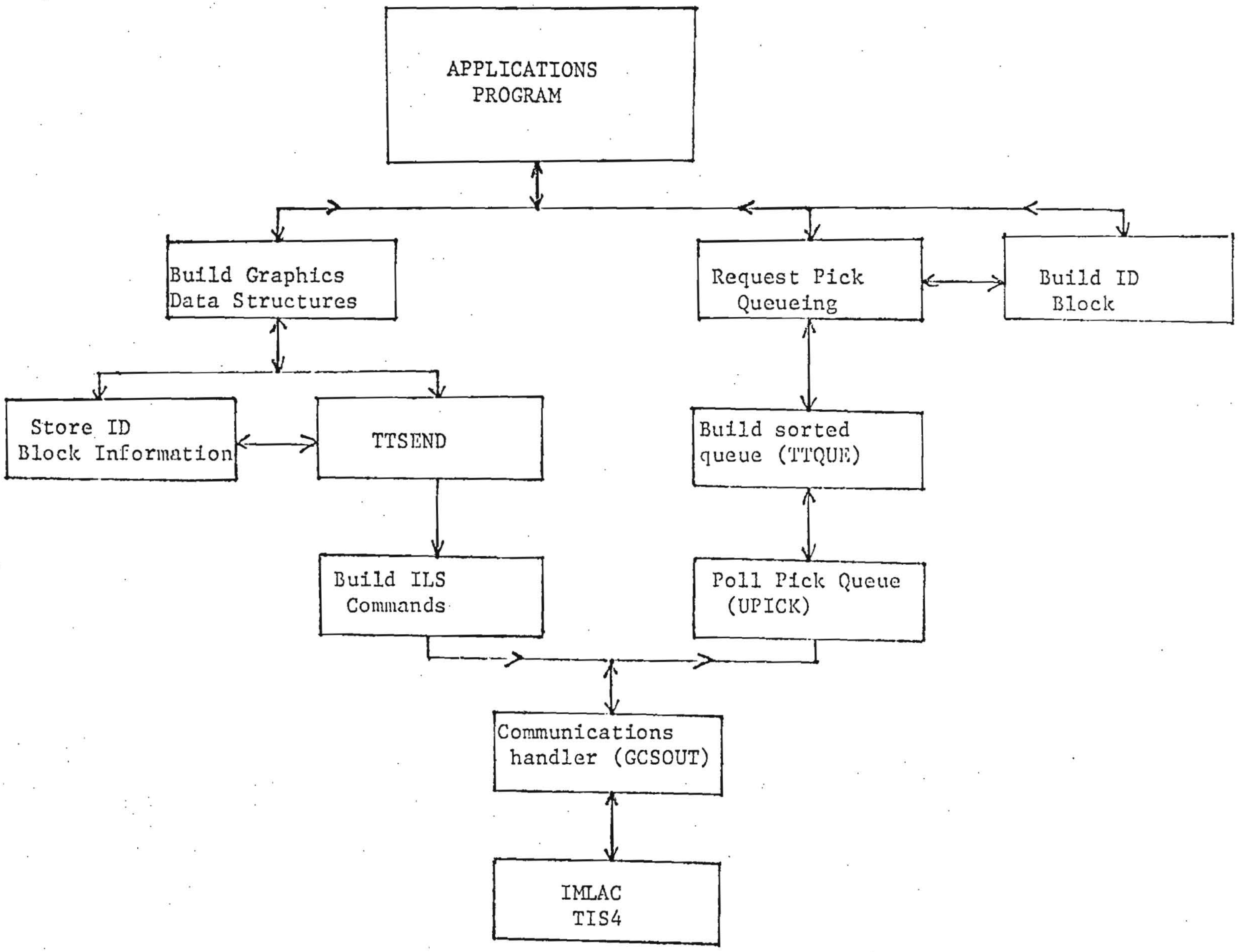

APPLICATIONS PROGRAM
Build Graphics Data Structures
Request Pick Queueing
Build ID Block
Store ID Block Information
TTSEND
Build sorted queue (TTQUE)
Build ILS Commands
Poll Pick Queue (UPICK)
Communications handler (GCSOUT)
IMLAC TIS4

Figure 2

REQUEST TO USE A MACRO. SEE COMMON DATA REGION /TTRB1/ DOCUMENTATION BELOW FOR A DEFINITION OF THE SUB-ITEM FLAG VALUES. BOTH MACROS AND ITEMS UTILIZE THE SUB-ITEM DESCRIPTION FLAGS TO BUILD GRAPHICS STRUCTURES. THE SUB-ITEM FLAGS ARE COMBINED WITH ANY NECESSARY RELATED INFORMATION IN IBUFA(B). LINE AND POSITION DEFINITION SUB-ITEM FLAGS ARE FOLLOWED BY APPROPRIATE COORDINATES, A CIRCLE BY THE REQUESTED RADIUS, AND INVOKE MACRO BY THE REFERENCE NUMBER(S) OF THE MACRO(S) TO BE INVOKED. THE TEXT SUB-ITEM FLAG WOULD BE FOLLOWED BY THE REQUESTED ASCII CHARACTER STRING TO BE DISPLAYED. WHEN THE USER ELECTS TO SEND THE BUILD INFORMATION CONTAINED IN IBUFA(B) TO THE TERMINAL (USUALLY THROUGH A CALL TO PUT) IT IS TRANSMITTED USING THE TTSEND MODULE.

TTSEND IS USED TO TRANSMIT EITHER STRUCTURES (MACROS) OR FRAMES (ITEMS) TO THE TERMINAL. TTSEND STORES I.D. BLOCK INFORMATION IN ARRAY TTREF, INDEXED BY IREF, FOR USE LATER BY THE PICK QUEUEING ROUTINES. AN I.D. BLOCK CONTAINS IDDT, IDDC, INFOA, INFOB (SEE SECTION 5). IREF IS RETURNED TO THE USER CONTAINING THE CONSOLE NUMBER (NEEDED BY GETLID,GIERAS, AND MERASE). SINCE THE TIS4 HANDLER CAN ONLY ACCEPT 0-255 FRAME OR STRUCTURE REFERENCE NUMBERS, THE IREF NUMBERS ARE MAPPED INTO THIS RANGE FOR TRANSMISSION TO THE TERMINAL. TTSEND DECODES IBUF AND TRANSMITS ILS (INTERNAL LOGICAL STRUCTURE) COMMANDS TO THE TIS4 IMLAC HANDLER TO BUILD THE DESIRED GRAPHIC STRUCTURES AT THE TERMINAL.

IBUF IS ALSO TRANSMITTED TO A RANDOM ACCESS DISK FILE FOR USE BY THE HARD COPY ROUTINES. THE HARD COPY ROUTINES READ FROM THE FILE AND DECODE THE STRUCTURES INTO APPROPRIATE CALLS TO THE PLOTTING ROUTINES SUPPLIED BY VERSATEC AND BROOMALL.

INTERACTIVE QUEUEING

THE QUEUEING FUNCTIONS PRESENT IN ITTR ALLOW AN OPERATOR, WORKING AT HIS OWN SPEED, TO QUEUE REQUESTS AHEAD OF APPLICATION PROGRAM EXECUTION. THESE REQUESTS MAY BE FROM THE LIGHTPEN OR THE FUNCTION KEYBOARD.

IN ORDER TO REDUCE THE PROCESSING TIME A CONSOLE OPERATOR MUST WAIT BETWEEN REQUESTS A SPECIAL QUEUEING PROCESS HAS BEEN IMPLEMENTED. THE QUEUEING PROCESS ALLOWS BUTTON, STRING, AND SINGLE PICK TYPES TO BE QUEUED. A COMPLETE QUEUE IS CONSIDERED ASSEMBLED ONLY WHEN THE OPERATOR HAS PICKED THE BUTTON TYPE. THE STRING AND SINGLE PICK TYPES CAN BE USED FOR SPECIAL INFORMATION ASSOCIATED WITH A BUTTON PICK. THE APPLICATION PROGRAM MAY PROCESS A COMPLETE QUEUE AT ONCE RATHER

THAN WAITING FOR SEVERAL INDIVIDUAL ITEMS.

QUEUE HANDLER FUNCTIONS

THE IMLAC 'TIS' HANDLER STORES AN UNSORTED FIRST IN FIRST OUT (FIFO) QUEUE LOCALLY. THE CENTRAL COMPUTER PERIODICALLY POLLS THE REMOTE COMPUTER AND THEN USES ARRAYS PICKQ AND WAITQ TO SORT INTO THE STANDARD C.D.C. 274 QUEUE STRUCTURE.

TWO QUEUES, PICKQ AND WAITQ, ARE MAINTAINED INTERNALLY IN THE IMLAC TWO TERMINAL ROUTINES. ONLY THE PICKQ IS ACCESSIBLE TO THE APPLICATION PROGRAM.

THE WAIT QUEUE FUNCTIONS AS A TEMPORARY INPUT BUFFER USED FOR ARRANGING AND ASSEMBLING A COMPLETE SET OF ID BLOCKS. SINCE THE WAITQ IS NOT ACCESSIBLE TO THE APPLICATION PROGRAM, THE USER IS PREVENTED FROM RECEIVING AN INCOMPLETE SET OF ID BLOCKS IF HE REQUESTS TRANSFER OF BLOCKS PRIOR TO THE ASSEMBLY OF A COMPLETE QUEUE.

A BUTTON PICK CAUSES AN ORDERED TRANSFER OF BLOCKS FROM THE WAIT QUEUE TO THE FETCH QUEUE (PICKQ) FOR RETRIEVAL BY THE APPLICATION PROGRAM. THE ASSOCIATED STRING AND SINGLE PICK ID BLOCKS ARE TRANSFERRED TO THE PICKQ AS THE RESULT OF A BUTTON PICK. THE USER MAY THEN RETRIEVE ID BLOCKS AS NEEDED BY CALLING GIFID AND/OR GIFSID.

DESCRIPTION OF COMMON DATA REGIONS USED BY ITTR

CRT2TERM

CRT2TERM IS THE BLOCKDATA SUBPROGRAM USED BY ITTR TO INITIALIZE SEVERAL COMMON REGIONS. WHEN CATALOGING A TASK WITH ITTR, THE USER SHOULD ALWAYS BE CERTAIN THE TWO TERMINAL ROUTINES ARE PROPERLY INITIALIZED BY USING THE 'INCLUDE CRT2TERM' CATALOG DIRECTIVE WHEN THE MAIN SEGMENT OF THE PROGRAM IS CATALOGED.

FOLLOWING ARE EXPLANATIONS OF THE DATA CONTAINED IN THE VARIOUS COMMON REGIONS INITIALIZED BY CRT2TERM.

FOR THE COMMON REGION /GCOM/ :

```
NCONA = USER SPECIFIED CONSOL NUMBER CORRESPONDING TO CONSOL #1
NCONB = USER SPECIFIED CONSOL NUMBER CORRESPONDING TO CONSOL #2
NBYTEA = NUMBER OF BYTES USED IN IBUFA IN CREATING DISPLAY ITEM
MBYTEA = MAXIMUM NUMBER OF BYTES AVAILABLE IN IBUFA
IBUFA  = BUFFER AREA FOR STORING ITEM UNDER CONSTRUCTION
IFR1XA = FRAMING VALUE FOR X AXIS (START) CONSOL A
IFR1YA = FRAMING VALUE FOR Y AXIS (START) CONSOL A
IFR2XA = FRAMING VALUE FOR X AXIS (END)   CONSOL A
IFR2YA = FRAMING VALUE FOR Y AXIS (END)   CONSOL A
IOB1XA = OBJECT SPACE BOUNDS FOR X (START) CONSOL A
IOB1YA = OBJECT SPACE BOUNDS FOR Y (START) CONSOL A
IOB2XA = OBJECT SPACE BOUNDS FOR X (END)   CONSOL A
IOB2YA = OBJECT SPACE BOUNDS FOR Y (END)   CONSOL A
SUB1XA = SUBJECT SPACE BOUNDS FOR X (START) CONSOL A
SUB1YA = SUBJECT SPACE BOUNDS FOR Y (START) CONSOL A
SUB2XA = SUBJECT SPACE BOUNDS FOR X (END)   CONSOL A
SUB2YA = SUBJECT SPACE BOUNDS FOR Y (END)   CONSOL A
         SEE SUBROUTINE 'SPACE'
CXA    = (IOB2X-IOB1Y)/(SUB2X-XUB1X) SUBJ TO OBJ CONVERSION
CYA    = (IOB2Y-IOB1Y)/(SUB2Y-SUB1Y) SUBJ TO OBJ CONVERSION
TXA    = -SUB1X*CXA + IOB1X       SUBJ TO OBJ CONVERSION
TYA    = -SUB1Y*CYA + IOB1Y         SUBJ TO OBJ CONVERSION
```

SIMILARLY FOR VARIABLES ENDING IN B ONLY FOR CONSOL B

TTREF IS USED TO STORE ALL ID INFORMATION ABOUT A DISPLAY ITEM OR MACRO. AN ID BLOCK CONTAINS IDDC, IDDT, INFOA, AND INFOB. FOR MORE INFORMATION ABOUT ID INFORMATION SEE THE USER DOCUMENTAITION IS SECTION 5. IN ADDITION TO NORMAL DISPLAY ITEMS TTREF IS USED TO STORE INFORMATION FOR THE FUNCTION KEYBOARD (FNT-1), LIGHTPEN SWITCH, AND THE END OF MESSAGE (EOM) ID BLOCK.

LAYOUT OF TTREF ARRAY:

TTREF(IREF,INFO,ITYPE)

IREF INDEX IS WHAT IS RETURNED TO CALLER OF TTR ROUTINES. USED BY THE APPLICATION PROGRAM FOR FUTURE REFERENCE OF A DISPLAY ITEM. CONTAINS THE CONSOLE NUMBER.

IREF = 1 - 510   1-255 FOR NCON=1   256-510 FOR NCON=2

ISREF = 1 - 255 IS MAPPED FROM IREF FOR TRANSMISSION TO THE TERMINAL

```
  ISREF = IREF - (NCON-1)*255


  I.D. INFO FOR THE FOLLOWING ARE STORED IN SPECIAL
  LOCATIONS TO PREVENT CONFUSION WITH OTHER ITEMS
  WHEN MAKING HARD COPY FROM THE RANDOM ACCESS FILE.

  NOTE: THE FIRST NUMBER IS FOR NCON=1, SECOND FOR 2

  IREF=255 & 510 IS RESERVED FOR KEYBOARD (FNT-1)
  IREF=254 & 509 IS RESERVED FOR LIGHTPEN SWITCH
  IREF=253 & 508 IS RESERVED FOR EOM


ITYPE FLAGS WHETHER A TTREF ENTRY IS FOR A MACRO OR ITEM.
  ITYPE=0,1     0=MACRO 1=ITEM


INFO = 1,2,3     FOR TTREF(IREF,INFO,ITYPE)

TTREF(IREF,1,ITYPE)      = 0 IF INDEX NOT USED
          BYTE 0 = REDUNDANT CONSOL FLAG USED TO PREVENT
                   THE POSSIBILITY OF A REAL TTREF ENTRY
                   BEING ZERO
          BYTE 1 = IS NOT USED
          BYTE 2 = IDDC NUMBER SUPPLIED BY APPLICATIONS PROGRAM
          BYTE 3 = IDDT MASK
                  BIT 7 = IGNORE AS PICK ITEM
                  BIT 6 = SINGLE PICK
                  BIT 5 = STRING PICK
                  BIT 4 = BUTTON PICK
TTREF(IREF,2,ITYPE) INFOA
TTREF(IREF,3,ITYPE) INFOB


JNCON(NCON)      GIVES THE ACTUAL CONSOLE NUMBER
                 USED INTERNALLY.  JNCON NORMALLY
                 EQUALS NCON.  A CALL TO CSWTCH
                 FLIPS THE INTERNAL REFERENCE.
                 USED TO AVOID CONFUSION OF IREF
                 REFERENCES AFTER CSWTCH IS CALLED.


FOR THE COMMON REGION /TTRCHARS/ :


TTRCHARS DEFINES A NUMBER OF SPECIAL SYMBOLS
USED PRIMARILY FOR CHARACTER STRING
MANIPULATION AND SCREEN FORMATING.

CSPACE = ACSII BLANK

CEOL   = ACSII CARRIAGE RETURN

CPLUS     = ACSII PLUS
```

```
CMINUS     = ACSII MINUS

CPOINT     = ACSII PERIOD

CHPLINE    = 96  IS THE NUMBER OF CHARACTERS PER DISPLAY LINE

LNSPSCR    = 62  IS THE NUMBER OF LINES PER DISPLAY PAGE

SCRWDTH    = 2496 IS THE OBJECT SPACE WIDTH OF THE DISPLAY

           SCREEN IN IGS UNITS (SEE APPENDIX III)

SCRHGTH    = 2496 IS THE OBJECT SPACE HEIGHT OF THE DISPLAY
           PAGE IN IGS UNITS (SEE APPENDIX III)
```

FOR THE COMMON REGION /TTRB1/ :

TTRB1 CONTAINS DATA FOR THE SUB-ITEM DESCRIPTION FLAGS. THE SUB-ITEM DESCRIPTION FLAGS ARE USED TO FLAG THE TYPE OF ITEM CONTAINED IN THE IBUF (A OR B) ARRAY. TTSEND AND THE HARD COPY ROUTINES DECODE THE ITEM TYPE FLAGS AND BUILD THE APPROPRIATE GRAPHICS DATA STRUCTURES.

```
TCIRCLE    FLAGS A REQUEST FOR A CIRCLE
TDEFPOS    FLAGS A REQUEST TO DEFINE POSITION (INVISIBLE MOVE)
TLINES     FLAGS A REQUEST TO DRAW A VISIBLE LINE
TTEXT      FLAGS A REQUEST TO OUTPUT TEXT
TUSEMAC    FLAGS A REQUEST TO USE A MACRO

MZERO      IS THE -0 FLAG = 32768
```

ISEGADD(NCON) IS THE FLAG TO REMEMBER WHERE A LINE SEGMENT DESCRIPTION STARTED. IT IS NON-ZERO AFTER CALLS TO STARTF,LINSEG,LINES OR CONTIL AND ZERO OTHERWISE. NCON = 1 OR 2.

FOR THE COMMON REGION /INFO/ :

INFO IS USED TO STORE DATA WHICH DEFINES ICODE, IDDC, AND ALL PICKQ AND WAITQ INFORMATION.

THE FOLLOWING DEFINE THE VARIOUS LEGAL ICODE & IDDT VALUES.

```
ENPICK     WHEN ICODE = ENPICK THEN THE ASSOCIATED
           ITEM IS SENSITIVE TO A LIGHTPEN PICK

HIGH       WHEN ICODE = HIGH THE ASSOCIATED ITEM
```

IS DISPLAYED AT HIGH INTENSITY

NOPCK	WHEN IDDT = NOPCK PICK QUEUEING IS IGNORED
BUTPCK	WHEN IDDT = BUTPCK THE ASSOCIATED ITEM IS TREATED AS A BUTTON PICK
INHIB	WHEN ICODE = INHIB DISPLAY PROCESSING IS INHIBITED UNTIL THE NEXT RESET SEQUENCE (THE ITEM IS NOT DISPLAYED)
MEDIUM	WHEN ICODE = MEDIUM THEN THE ASSOCIATED ITEM IS DISPLAYED AT MEDIUM INTENSITY.
SNGPCK	WHEN IDDT = SNGPCK THEN THE ASSOCIATED DISPLAY ITEM IS TREATED AS A SINGLE PICK WHEN IT IS QUEUED.
BLINK	WHEN ICODE = BLINK THEN THE ITEM IS BLINKED WHEN IT IS DISPLAYED.
LOW	WHEN ICODE = LOW THE ASSOCIATED ITEM IS DISPLAYED AT LOW INTENSITY.
STRPCK	WHEN IDDT = STRPCK THEN THE ASSOCIATED ITEM IS TREATED AS A STRING PICK WHEN QUEUED.

THE PICKQ AND WAITQ'S ARE DEFINED AS FOLLOWS:

PICKQ(NCON,I)
WAITQ(NCON,I)

NCON = 1 OR 2
I   = 0 TO 32

THE QUEUES CONTAIN INFORMATION NECESSARY TO ASSEMBLE VALID ID BLOCKS.

THE FOLLOWING LOCATIONS IN THE QUEUES ARE RESERVED FOR SPECIAL INFOMATION USED IN THE SORTING PROCESS:

PICKQ(NCON,0)	= 0 IF PICKQ EMPTY
	= 1 IF PICKQ COMPLETE
	= N WHERE N IS LAST STRING PICK INSERTED
WAITQ(NCON,0)	= POINTER TO LAST ITEM TAKEN OUT OF WAITQ

GCSBLOCK

GCSBLOCK CONTAINS LOGICAL FILE CODES AND BUFERRING USED BY THE IMLAC I/O ROUTINES. THE USER SHOULD

USE THE 'INCLUDE GCSBLOCK' CATALOG DIRECTIVE AT
CATALOG TIME TO INITIALIZE COMMUNICATIONS TO
THE IMLAC TERMINALS AND THE HISTORY FILE.

FOLLOWING ARE EXPLANATIONS OF THE COMMON DATA
REGIONS INITIALIZED BY GCSBLOCK:

FOR THE COMMON REGION /DEVICE/ :

IPDS CONTAINS THE LOGICAL FILE CODE FOR THE
CURRENTLY ACTIVE IMLAC DISPLAY CONSOLE. IHF
CONTAINS THE LOGICAL FILE CODE OF THE HISTORY FILE.

FOR THE COMMON REGION /TBUFFER/ :

BUFF BUFFERS I/O DATA FOR THE IMLAC DISPLAY CONSOLE.

FOR THE COMMON REGION /SWLFC/ :

LFC CONTAINS THE CURRENT LOGICAL FILE CODE
ASSOCIATED WITH THE USERS REFERENCE TO
TO CONSOLE 1 AND CONSOLE 2.

THE REGION CBGCS IS USED BY THE LOWEST LEVEL
GRAPHICS ROUTINES TO STORE VARIOUS STATUS
INFORMATION.

THE REGION /CBGCS/ IS DEFINED AS FOLLOWS:

XX IS THE CURRENT SOFTWARE BEAM POSITION
IN THE HORIZONTAL DIRECTION.

YY IS THE CURRENT SOFTWARE BEAM POSITION
IN THE VERTICAL DIRECTION.

MODE = 0 FOR ABSOLUTE MODE
= 1 FOR RELATIVE MODE

IDASH = 0 FOR SOLID LINES
= 1 FOR DASHED LINES
= 2 FOR BROKEN LINES
= 3 FOR CENTERLINE

ISTUFF = 0 FOR HISTORY OFF

= 1 FOR HISTORY ON

DESCRIPTION OF ITTR INTERNAL ROUTINES

CONVDIG

CALL CONVDIG(IXIMLAC,IYIMLAC,IXIGS,IYIGS)

CONVERTS FROM IMLAC DEVICE SPACE TO VIRTUAL SPACE (OBJECT OF THE TWO TERMINAL ROUTINES). IMLAC DEVICE SPACE IS (0,0) TO (2048,2048). THE OBJECT OF THE TWO TERMINAL ROUTINES IS (-1248,-1248) TO (1248,1248).

CONVIGD

CALL CONVIGD(IXIGS,IYIGS,IXIMLAC,IYIMLAC)

CONVERTS IGS SPACE (OBJECT OF TWO TERMINAL ROUTINES) TO IMLAC DEVICE SPACE. THIS ROUTINE IS THE OPPOSITE OF CONVDIG.

COPYF

CALL COPYF(NCON,WIDTHPL,NAME,EXT)

SUBROUTINE COPYF IS USED TO PRODUCE HARD COPY PLOTS OF WHAT IS CURRENTLY DISPLAYED ON THE IMLAC.

THIS ROUTINE PLOTS ON EITHER THE VERSATEC OR THE BROOMALL PLOTTERS. NPLOT IS THE VARIABLE USED TO SPECIFY WHICH PLOTTER IS TO BE USED. WHEN NPLOT IS 1, THE VERSATEC IS USED; WHEN NPLOT IS 2 OR 4, THE BROOMALL IS USED, WHERE 2 PRODUCES A 12" BY 12" PLOT AND 4 PRODUCES A 24" BY 24" PLOT. NPLOT MUST BE SET PROPERLY BY THE APPLICATION PROGRAM AND SENT THROUGH THE PSWITCH COMMON.

THIS ROUTINE READS INFORMATION FROM THE HISTORY FILE WHICH IS CREATED BY THE TWO TERMINAL ROUTINES WHENEVER AN ITEM OR MACRO IS DISPLAYED ON THE IMLAC AND THE HISTORY FILE SWITCH IS ON (SEE USET).

PLOTS ON THE VERSATEC USE THE FOLLOWING ROUTINES:

VMODE: VARIOUS VERSAPLOT SUBROUTINES EXAMINE MODE TABLE

VARIABLES FOR LOGICAL DEFAULT OR COMMUNICATED SETTINGS. SUBROUTINE MODE PROVIDES PROGRAM CONTROL OVER THE TOTAL MODE TABLE ORGANIZATION ALLOWING THE USER TO REQUEST OR ALTER VARIABLES AS REQUIRED. COPYF SETS SEVERAL OF THESE VARIABLES IN THE PROCEDURE INIT-VERSATEC. MODE 1 SETS THE UNITS-OF MEASURE, MODE 2 AND MODE 3 SET THE X AND Y PLOTTING LIMITS, RESPECTIVELY, AND MODE 4 IS USED TO SET THE CHARACTER HEIGHT AND WIDTH. COPYF ALSO SETS MODE 10 IN THE PROCEDURE DRAW-LINE-SEGMENTS. MODE 10 DEFINES A LINE MASK BIT PATTERN WORD THAT IS USED WHEN DRAWING LINES. THE INTEGER VARIABLES SOLID, DOT, DASH, AND DASHD ARE USED TO REPRESENT SOLID LINES, DOTTED LINES, DASHED LINES, AND DASH-DOTTED LINES.

VDRAW: USED TO DRAW LINES FROM PRESENT POSITION TO THE NEW X-Y POSITION, OR MOVE THE POSITION OF THE 'PEN' TO THE NEW X-Y COORDINATE, OR SPECIFY END OF PLOT OR END OF SEVERAL PLOTS.

VNOTE: OUTPUTS TEXT AT THE SPECIFIED X-Y POSITION.

BROOMALL PLOTS ARE GENERATED BY USING THE FOLLOWING ROUTINES:

BSTART: USED TO PERFORM INITIALIZATION FUNCTIONS SUCH AS SETTING INTERNAL PARAMTETERS.

BPARST: USED TO INITIALIZE THE XPAR AND YPAR ARRAYS. VALUES FOR THE XPAR AND YPAR ARRAYS ARE ALTERED BY THE INIT-BROOM PROCEDURE. XPAR(2) AND YPAR(2) GIVE THE PAPER SIZE; XPAR(3) AND YPAR(3) IS THE ORIGIN MEASURED IN INCHES FROM THE ABSOLUTE ORIGIN; XPAR(4) AND YPAR(4) ARE THE SCALE FACTORS; XPAR(5) IS THE MAGNIFICATION OR REDUCTION FACTOR; AND YPAR(5) SELECTS THE PEN.

BSYMBOL: OUTPUTS TEXT AT THE X-Y POSITION.

BLINES: DRAWS A LINE OR LINES CONNECTING N POINTS CONTAINED IN AN ARRAY OF N ELEMENTS. A MINIMUM OF ONE LINE CAN BE DRAWN WHERE THE INITIAL POSITION AND END POSITION ARE GIVEN.

THE BROOMALL ROUTINES DO NOT PROVIDE THE CAPABILITY TO MOVE THE 'PEN'; THUS, IT WAS NECESSARY TO KEEP TWO VARIABLES, XPAR9 AND YPAR9, WHICH CONTAIN THE X AND Y LOCATION OF WHERE THE PEN SHOULD BE AT ANY PARTICULAR TIME.

GDVIC

CALL GDVIC(INEW)

GDVIC CHANGES THE INTENSITY OF DISPLAY ITEMS. IT

IS CALLED WITH AN INTENSITY OF INEW=0 (BEAM OFF) TO INEW=15 (HIGHEST INTENSITY).
GDVIC MUST BE CALLED WITHIN A FRAME DEFINITION
DEFAULT INTENSITY IS INEW=15 (HIGHEST)
GDVIC TRANSMITS AN ILS CODE TO THE IMLAC
HANDLER TO EFFECT THE CHANGE IN INTENSITY LEVEL.
ICODE INCLUDES INFORMATION FOR ONLY THREE
INTENSITY LEVELS. TTSEND MAPS ICODE INTO INEW = 5, 10,15 AND THEN CALLS GDVIC TO EFFECT THE NEW INTENSITY CHANGE.

MZEROC

FUNCTION MZEROC(IX)

MZEROC IS A LOGICAL FUNCTION WHICH
IS TRUE ONLY IF IX = MZERO (32768)

RANF

FUNCTION RANF(ISEED)

32 BIT RANDOM NUMBER GENERATOR

SETKEYS

CALL SETKEYS(IKEY,IH,IV)

SETS PROPER BIT IN IH AND IV FOR ID BLOCKS
IKEY = 1 - 16 (RETURNED BY IMLAC)
IH AND IV ARE AS SPECIFIED IN THE ORIGINAL
TWO TERMINAL ROUTINES.

SETSPACE

CALL SETSPACE(IX,IY,IC,XA,YA,XB,YB)

SETSPACE DETERMINES WHERE A POINT (IX,IY)
IS LOCATED WITH RESPECT TO THE
SPACE WITH LOWER LEFT HAND CORNER
(XA,YA) AND UPPER RIGHT HAND CORNER
(XB,YB). IC RETURNS A FOUR BIT CODE
WHICH CONTAINS THE POINTS LOCATION.
SETSPACE IS USED WITH TTLSCIS FOR LINE
SCISSORING (CLIPPING).

TTLSCIS

```
CALL TTLSCIS(X1,Y1,X2,Y2,KSHOW,XA,YA,XB,YB)
```

TTLSCIS CLIPS THE LINE (X1,Y1) , (X2,Y2)
INSIDE FRAME WITH LOWER LEFT CORNER (XA,YA) AND
UPPER RIGHT CORNER (XB,YB)

KSHOW RETURNED = 0 IF LINE IS OUTSIDE FRAME
KSHOW RETURNED = 1 IF LINE IS ENTIRELY WITHIN FRAME
KSHOW RETURNED = 2 IF LINE HAS BEEN CLIPPED

TTQUE

```
FUNCTION TTQUE(NCON)
```

TTQUE FETCHES QUEUE INFORMATION FROM THE IMLAC
AND SORTS IT INTO THE STANDARD CDC TYPE QUEUE
STRUCTURE. ARRAYS USED ARE: PICKQ AND WAITQ.
TTQUE IS A LOGICAL FUNCTION WHICH IS TRUE
WHEN A COMPLETE QUEUE HAS BEEN CONSTRUCTED.

TTRGET

```
CALL TTRGET(IREF,NCON,ITEMF)
```

TTRGET RETRIEVES THE BYTE ARRAY FOR A DISPLAY ITEM OR
MACRO FROM THE RANDOM ACCESS HISTORY FILE USED TO
STORE THEM. USED FOR "COPY", "MOVE", AND TO
MAKE HARD COPY. THE BYTE STREAM FOR THE ITEM OR
MACRO IS STORED IN IBUF AND NBYTE IS SET
IS SET CORRECTLY. THE ITEMF FLAG IS ZERO
FOR A MACRO OTHERWISE A ITEM IS RETRIEVED
FROM THE HISTORY FILE.

TTRPUT

```
CALL TTRPUT(IREF,NCON,ITEMF)
```

TTRPUT IS THE INVERSE OF TTRGET. THE CURRENT IBUF AND
NBYTE IS WRITTEN INTO THE RANDOM ACCESS HISTORY FILE.

TTSCON

```
CALL TTSCON(NCON)
```

TTSCON SWITCHES I/O BETWEEN CONSOLE 1 AND 2.

TTSEND

CALL TTSEND(IREF,IDDT,IDDC,INFOA,INFOB,NCON,ITEMF)

TTSEND TRANSMITS EITHER MACROS OR ITEMS TO THE TERMINAL. I.D. BLOCK INFORMATION CONSISTING OF (IDDT, IDDC, INFOA, INFOB) IS STORED IN ARRAY TTREF THROUGH A CALL TO TTSETR. IREF IS RETURNED BY TTSETR FOR FUTURE REFERENCE OF THE ITEM OR MACRO BY ITTR OR THE APPLICATION PROGRAM. IREF CONTAINS THE CONSOLE NUMBER.

SINCE 'TIS' CAN ONLY ACCEPT REFERENCE NUMBERS IN THE RANGE FROM 0 TO 255, THE ITEM OR MACRO REFERENCE NUMBER (IREF) IS MAPPED INTO INAME FOR TRANSMISSION TO 'TIS'.

THE ALGORITHM USED IS:
INAME = IREF - (JNCON(NCON)-1)*255

APPROPRIATE ROUTINES ARE CALLED BY TTSEND TO TRANSMIT THE CORRECT ILS COMMANDS TO THE IMLAC PDS-4 HANDLER.

WHEN ITEMF=0 TTSEND TRANSMITS MACROS. OTHERWISE, ITEMS ARE TRANSMITTED.

TTSETR

CALL TTSETR(IREF,IDDT,IDDC,INFOA,INFOB,NCON,ITEMF)

TTSETR FINDS AN UNUSED INDEX IN TTREF TO STORE INFO, STORES THE APPROPRIATE INFORMATION, AND RETURNS THE INDEX IN IREF. ALL REFERENCES TO THE DISPLAY ITEM OR MACRO IS THROUGH THE INDEX IREF. ARGUMENTS ARE AS IN DESCRIPTION OF TTREF. AN ERROR MESSAGE IS PRINTED IF THE TTREF ARRAY IS FULL. IREF IS RETURNED TO TTSEND 0 TO FLAG THE FULL TABLE. TTSEND THEN IGNORES THE CURRENT REQUEST TO SEND ITEMS OR MACROS UNTIL THE USER MAKES ROOM IN THE TTREF ARRAY BY DELETEING AN ITEM OR MACRO.

ITEMF = 0 FOR MACROS
ITEMF = 1 FOR NORMAL NEW ITEM PROCESSING
ITEMF = 99 TO FLAG CALLS FROM MOVIT
ITEMF = 999 TO FLAG CALLS FROM COPY

TTSTORE

CALL TTSTORE(N,IA,NCON)

TTSTORES STORES SUB-ITEM DESCRIPTIONS IN
THE APPROPREATE IBUF FOR CONSOL NCON.
N IS THE NUMBER OF WORDS TO STORE OF THE
ARRAY IA. A CHECK IS MADE TO SEE IF SUB-
ITEM WILL FIT. THE FLAG 'ISEGADD' IS ALSO
CLEARED FOR CONSOL. 'ISEGADD' IS USED
BY CONTIL TO DETERMINE IF THE DESCRIPTION
BUFFER HAS BEEN ASSEMBLED CORRECTLY
BEFORE THE CALL TO CONTIL WAS ISSUED.

UATFR

CALL UATFR(INAME)

UATFR LOGICALLY ASSOCIATES THE FRAME
"INAME" WITH THE TRACKING CROSS
SUBSEQUENT MOVEMENT OF THE TRACKING CROSS
CAUSES AN IDENTICAL VECTOR DISPLACEMENT
OF THE FRAME.

UCRCLE

CALL UCRCLE(X,Y,R)

UCRCLE MAKES A CIRCLE OF RADIUS R AT COORDINATES
(X,Y) RELATIVE TO THE CURRENT BEAM POSITION.
THE BEAM IS LEFT IN THE SAME POSITION IT WAS
BEFORE THE CALL.

UDEACTIV

CALL UDEACTIV(INAME)

UDEACTIV TURNS OFF THE FRAME "INAME" AT THE
IMLAC TERMINAL. STORAGE IS DEALLOCATED ONLY
AT THE IMLAC.

UDISLP

CALL UDISLP(INAME)

DISABLES LIGHT PEN PICKS FOR THE FRAME "INAME"
SUBSEQUENT PICKS WILL NOT BE QUEUED.

UENBLP

CALL UENBLP(INAME,INDEX)

ENABLES THE LIGHT PEN FOR THE FRAME 'INAME'
SO THAT WHEN THAT FRAME IS PICKED BY THE
LIGHT PEN, 'INDEX' WILL BE PLACED IN THE
PICK QUEUE FOR SUBSEQUENT RETRIEVAL.

UEOM

```
CALL UEOM(CHAR,INDEX)
```

CAUSES LIGHT REG TEXT RETRIEVAL TO TERMINATE UPON
ENTRY OF CHAR SPECIFIED WITH SUBSEQUENT ENTRY
OF INDEX INTO THE PICK QUEUE.
INDEX = 0 CLEARS E.O.M. LIST

UFRAM

```
CALL UFRAM(INAME)
```

UFRAM STARTS A FRAME DEFINITION BY SENDING
THE PROPER ILS CODE.

UGETX

```
CALL UGETX(IX,IY,ISTAT)
```

RETURNS COORDINATES OF THE TRACKING CROSS IN
IX AND IY AFTER A MOVE.
ALSO DELETES THE TRACKING CROSS

```
BIT 5  1=DELETE FRAME AND DETACH
       0=LEAVE FRAME ON
BIT 6 1=DELETE T.C. AND DETACH IF ATTACHED
      0=LEAVE T.C.
BIT 7 1=DETACH FROM FRAME
```

THIS ROUTINE RETURNS IMLAC DEVICE
SPACE COORDINATES

```
ISTAT:  0 LEAVES ATTACHED FRAME AND T.C.
        ADD 2 TO DELETE T.C.
        ADD 1 TO DETACH
```

UFREN

```
CALL UFREN (INAME)
```

UFREN ENDS A FRAME DEFINITION BY SENDING
THE PROPER ILS CODE.

UGTEXT

UGTEXT RETRIEVES THE LIGHT REGISTER TEXT BUFFER.
CAUSES 162 BYTES OF DATA TO BE XMITTED TO HOST
ARRAY RETURNS LIGHT REGISTER TEXT BUFFER IN
CORRECT FORMAT N=NUMBER CHARS INCLUDING
THE EOM CHAR

UMOVE

CALL UMOVE(X,Y)

UMOVE POSITIONS THE BEAM AT RELATIVE OR
ABSOLUTE LOCATION (X,Y) DEPENDING UPON THE STATUS
OF MODE (SEE USET).

UPEN

CALL UPEN(X,Y)

UPEN DRAWS A LINE FROM THE CURRENT BEAM
POSITION TO RELATIVE OR ABSOLUTE BEAM POSITON
(X,Y) DEPENDING UPON THE STATUS OF
MODE (SEE USET).

UPICK

CALL UPICK(IQUEUE)

COPY CURRENT STATUS OF PICK QUEUE INTO IQUEUE
PDS-4 TRANSMITS 128 BYTES OF DATA DOWM THE
TTY LINE.

UPRINT

CALL UPRINT(DUM,DUM,ICHAR)

UPRINT DISPLAYS THE CHARACTER ARRAY ICHAR
AT THE CURRENT BEAM LOCATION. THE CHARACTER
ARRAY IS TERMINATED BY THE ' # ' TERMINATOR.
IF AN ATTEMPT TO DISPLAY MORE THAN 96 CHARACTERS
IS MADE THEN THE LINE IS CLIPPED AND AN ERROR
MESSAGE IS PRINTED ON THE 'LO' FILECODE.

UPSET

CALL UPSET(IDUM,IDUM)

UPSET IS A NULL ROUTINE USED FOR GCS COMPATABILITY

UPUTX

CALL UPUTX(IX,IY)

POSITIONS TRACKING CROSS AT ABSOLUTE BEAM
COORDINATES (IX,IY)
THIS ROUTINE MUST BE SENT COORDINATES
IN IMLAC DEVICE SPACE.

URING

CALL URING

RING IMLAC BELL

USET

USET CHANGES VARIOUS SWITCHES IN
THE COMMON BLOCK CBGCS. IMODE SHOULD
CONTAIN A FOUR CHARACTER LITERAL
CONTAINING THE DESIRED MODE CHANGE.

```
CALL USET(IMODE)
          IMODE = ABSO  FOR ABSOLUTE MODE
          IMODE = RELA  FOR RELATIVE MODE
          IMODE = LINE  FOR SOLID LINES
          IMODE = DASH  FOR DASHED LINES
          IMODE = BROK  FOR BROKEN LINES
          IMODE = CENT  FOR CENTER LINE MODE
          IMODE = OFF   TO TURN OFF HISTORY FILE
          IMODE = ON    TO TURN ON HISTORY FILE
          IMODE = INIT  FOR INITIALIZATION

                   INITIALIZATION IS AS FOLLOWS:

                   ABSOLUTE MODE IS SELECTED.
                   LINE TYPE IS SOLID.
                   HISTORY FILE IS TURNED ON.
                   INTERNAL BEAM POSITION IS (0.,0.)

                   NOTE THAT A CALL TO CLEAN
                   CAUSES INITIALIZATION AS ABOVE.
```

AN ERROR MESSAGE IS PRINTED IF
USET IS CALLED WITH AN INCORRECT MODE.

USTRC

CALL USTRC(INAME)

USTRC STARTS A STRUCTURE DEFINITION BY
SENDING THE PROPER ILS CODES.

UTER

CALL UTER(INAME)

UTER ENDS A STRUCTURE DEFINITION BY SENDING
THE PROPER ILS CODES.

UTEXT

CALL UTEXT(X,Y,N)

CAUSES A LIGHT REGISTER OF N CHARACTERS
TO APPEAR AT ABSOLUTE BEAM COORDINATES (X,Y)
AFTER CALL ENTRIES TO KEYBOARD WILL BE
ECHOED TO THE DISPLAY UNTIL EOM.

IMLAC TWO TERMINAL ROUTINE USER DOCUMENTATION

COORDINATE SYSTEMS

SUBJECT SPACE COORDINATES

USER GENERATED X-Y COORDINATES IN ARBITRARY UNITS OF
MEASURE; FLOATING POINT COORDINATES ARE USED.
TO DIFFERENTIATE FLOATING POINT 0 AND INTEGER 0 ON THE
SEL ALL REFERENCES IN THE APPLICATION PROGRAMS TO
' -0 ' SHOULD BE REPLACED BY MZERO = 32768.

OBJECT SPACE COORDINATES

RASTER UNIT COORDINATES (IGS) COORESPONDING TO THE
ACTUAL DEVICE SPACE OF THE CDC 1700/274 GRAPHICS
SUBSYSTEM. THESE COORDINATES ARE MAPPED INTO
THE DEVICE SPACE OF THE IMLAC PDS-4 BY THE

SPECIAL CALLING PARAMETERS
TO THE ITTR ROUTINES

IBEAM	BEAM CONTROL PARAMETER =0 WHEN SEGMENT IS NOT DISPLAYED, IBEAM = 1 SEGMENT IS DISPLAYED ACCORDING TO ISTYLE
ICODE	RESET CONTROL CODE BIT PATTERN OF THE FORM SOOTFBB WHERE: S = 0 DISABLE ITEMS SENSITIVITY TO LIGHTPEN PICK S = 1 ENABLE ITEMS SENSITIVITY TO LIGHTPEN PICK T=0 NORMATL DISPLAY PROCESSING T=1 INHIBIT DISPLAY PROCESSING UNTIL NEXT RESET SEQUENCE F=0 DON'T BLINK DISPLAY ITEM WHEN DISPLAYED F=1 BLINK DISPLAY ITEM BB=01 DISPLAY ITEM AT LOW INTENSITY BB=10 DISPLAY ITEM WITH BEAM AT MEDIUM INTENSITY BB=11 DISPLAY ITEM AT HIGH INTENSITY
IDDC	DISPLAY ITEM IDENTIFICATION CODE; ASSIGNED BY THE PROGRAMMER; RETURNED BY THE QUEUEING ROUTINES
IDDT	DISPLAY ITEM TYPE CODE IDDT = 1 IGNORE PICK QUEUEING FOR THIS ITEM IDDT = 2 QUEUE AS A SINGLE PICK IDDT = 4 QUEUE AS A STRING PICK ITEM IDDT = 8 QUEUE AS A BUTTON PICK ITEM
IH	HORIZONTAL COORDINATE OF AN ITEM IN IGS UNITS
IV	VERTICAL COORDINATE OF AN ITEM IN IGS UNITS
INFOA	USER INFORMATION WORD
INFOB	USER INFORMATION WORD
IREF	REFERENCE VALUE FOR ADDRESSING A DISPLAY ITEM INCLUDES THE CONSOLE NUMBER
ISTYLE	DETERMINES THE TYPE OF LINE SEGMENT TO BE DISPLAYED ISTYLE = 0 LINES ARE SOLID ISTYLE = 4Z0AAA SEGMENT IS DASHED (CDC 5252B)

ISTYLE = 4Z0DB6 SEGMENT IS BROKEN (CDC 6666B)
ISTYLE = 4Z0EBA SEGMENT IS CENTER LINE (CDC 7272B)

KSHOW — SCISSOR FLAG
KSHOW = 0 LINE IS OUTSIDE FRAME
KSHOW = 1 LINE IS COMPLETELY WITHIN FRAME
KSHOW = 2 LINE HAS BEEN SCISSORED

MREF — REFERENCE NUMBER OF A MACRO

NCON — GRAPHICS CONSOLE NUMBER (1 OR 2)

ITTR CALLING SEQUENCES

ATCHX(IREF,NCON)
ATTACHES A DISPLAY ITEM TO THE TRACKING CROSS.

CHAR(N,CHARS,IVAL)
PUTS THE N-TH CHAR IN CHARS INTO IVAL (LEFT-JUSTIFIED)

CHECK(N,ALPHA,ILOOK,IER)
CHECKS N CHARS IN ALPHA ACCORDING TO ILOOK.

ILOOK = 0 LOOK FOR UNSIGNED INTEGERS
ILOOK = 1 LOOK FOR SIGNED I OR F TYPE.
IER = 1 IF ERROR FOUND, = 0 OTHERWISE.

CIRCLE(XCEN,YCEN,RADIUS,ISTYLE,NCON)
DRAWS A CIRCLE AT THE RELATIVE LOCATION (XCEN,YCEN) OF RADIUS.

CLEAN(NCON)
CLEARS TABLES, SETS EOM, AND INITIALIZES ITTR.

CONTIL(X,Y,IBEAM,NCON)
USED TO CONTINUE A LINE FIGURE DESCRIPTION AFTER A CALL TO STARTF, LINSEG, OR LINES.

COPY(IROLD,XNEW,YNEW,ICODE,IRNEW,IDDT,IDDC,INFOA,INFOB,NCON)
DUPLICATES A DISPLAY ITEM FOUND AT IROLD AND PLACES IT AT (XNEW,YNEW) WITH REFERENCE NUMBER IRNEW.

CSWTCH(ITEMP)

SWITCHES CONSOLE IDENTIFICATION.
ITEMP SHOULD BE DIMENSIONED ITEMP(369).
ITEMP IS USED AS A TEMPORARY ARRAY FOR
SWAPPING GRAPHICS STATUS AREAS BETWEEN
THE TWO CONSOLES.

CVTOBJ(X,Y,IH,IV,NCON)
CONVERTS SUBJECT SPACE (X,Y, REAL) TO OBJECT SPACE
COORDINATES (IH,IV INTEGER).

CVTSUB(IH,IV,X,Y,NCON)
INVERSE OF CVTOBJ.

DEFPOS(X,Y,ICODE,NCON)
POSITIONS BEAM FOR ITEM UNDER CONSTRUCTION.

DIF(I,J,K)
COMPUTES K = I - J. USED TO SOLVE DATA TYPING PROBLEMS.

DTCHX(NCON)
DETACHES DISPLAY ITEM FROM TRACKING CROSS.

EOMOUT(LINE,NC)
REPLACES EOM CHAR IN LINE WITH A BLANK. NC IS
DECREMENTED.

ENDJOB(NCON)
CLEARS SCREEN OF CONSOLE NCON.

ERASE(N,IREF)
ERASE N ITEMS FROM SCREEN REFERENCED BY IREF.

ERRMES(N1,CHAR1,N2,CHAR2)
DISPLAYS ERROR MESSAGES.

FLIP(I,J)
SWITCHES VALUES OF I AND J.

FLIPC(NCON)
FLIPS CONSOLES ACTIVE STATUS.

FRAME(CLCX,CLCY,CUCX,CUCY,NCON)
SETS FRAME VALUES FOR DISPLAYING NCON. THE
FRAME IS SPECIFIED (CLCX,CLCY) FOR LOWER LEFT
CORNER AND (CUCX,CUCY) FOR UPPER RIGHT.

GETCXY(NCON,X,Y)
GETS SUBJECT SPACE COORDINATES OF THE TRACKING CROSS WHEN THE LAST BUTTON WAS PICKED.

GETLID(IDDT,IDDC,INFOA,INFOB,IH,IV)
GETS ID INFORMATION STORED IN THE LAST BUFFON PICK ID FETCHED BY WAIT.

GETREG(NCON,ARRAY,NCHAR,X,Y,IDDT,IDDC,ICODE,IREF)
GETS INFO FROM THE LIGHT REGISTER AND REMOVES THE EOM CHARACTER. NCHAR IS RETURNED WITH THE CORRECT NUMBER OF CHARS IN ARRAY.

GIERAS(IREF)
ERASES THE DISPLAY ITEM IREF FROM THE REFRESH BUFFER.

GIFID(NCON,IDDT,IDDC,INFOA,INFOB,IH,IV)
GETS ID INFORMATION FOR THE SINGLE PICK ITEM ASSOCIATED WITH THE LAST BUTTON PICK. IF NO SINGLE PICK ITEM IS STORED THEN IDDT IS RETURNED AS 0.

GIFSID(NCON,N,IDDT,IDDC,IDWA,IDWB,IH,IV)
GETS ID INFORMATION FOR STRING PICKS. THE ID INFORMATION IDDT----IH MUST BE DIMENSIONED CORRECTLY IN THE CALLING PROGRAM. N IS THE NUMBER OF STRING PICK ID BLOCKS DESIRED. N IS RETURNED EQUAL TO THE NUMBER OF STRING PCIK ITEMS ACTUALLY QUEUED (UP TO 30).

ICENT(NCHAR)
GIVES THE X-COORDINATE IN IGS OBJECT SPACE TO CENTER NCHAR CHARACTERS.

INTGR(X)
DETERMINES IF X IS INTEGER OR REAL. THIS FUNCTION IS TRUE IF X IS AN INTEGER AND FALSE OTHERWISE. FLOATING POINT ZERO IS REPRESENTED BY MZERO=32768 ON THE SEL.

LCX(CHARNO)
COMPUTES THE DGU (OBJECT SPACE) OF THE HORIZONTAL CHARACTER POSITION CHARNO.
CHARNO = 1 - 97

```
LINCZR(XS,YS,XE,YE,KSHOW,XSS,YSS,XES,YES,NCON)
          SCISSORS (CLIPS) THE LINE (XS,YS) , (XE,YE)
          ACCORDING TO THE CURRENT FRAME VALUES.
          KSHOW = 0 LINE NOT IN FRAME
          KSHOW = 1 LINE IS COMPLETELY WITHIN FRAME
          KSHOW = 2 LINE HAS BEEN CLIPPED


  LINES(X,Y,IBEAM,N,ISTYLE,NCON)
          CREATES A DESCRIPTION OF N LINE SEGMENTS AS
          PART OF A DISPLAY ITEM.  X,Y,IBEAM MUST
          BE DIMENSIONED AT LEAST N+1.


  LINSEG(X1,Y1,X2,Y2,IBEAM,ISTYLE,NCON)
          CREATES DESCRIPTION OF A LINE AS PART OF A
          DISPLAY ITEM.


  LLY(LINENO)
          THIS FUNCTION RETURNS THE DGU (OBJECT SPACE)
          VALUE OF THE GIVEN LINE NUMBER.  LINENO = 1 - 62


  MATCHX(MREF,NCON)
          ATTACHES THE ALREADY DEFINED MACRO TO THE
          TRACKING CROSS SO THAT THE MACRO MOVES WITH
          THE T.C. ACROSS THE SCREEN.  MREF REFERENCES
          THE MACRO TO BE ATTACHED.


  MAKMAC(MREF,NCON)
          STORES DISPLAY ITEM DESCRIPTION AS A MACRO
          WITH REFERENCE VALUE MREF.


  MDTCHX(NCON)
          DETACHES AN ATTACHED MACRO FROM THE TRACKING
          CROSS.  THE MACRO DISAPPEARS.


  MERASE(N,MREF)
          ERASE N MACRO ITEMS FROM THE REFRESH BUFFER.
          MREF IS AN ARRAY OF N MACRO REFERENCE NUMBERS.


  MOVIT(IREF,XNEW,YNEW,ICODE,IDOT,IDDC,INFOA,INFOB,NCON)
          MOVES A DISPLAY ITEM FROM ITS CURRENT LOCATION
          TO (XNEW,YNEW).  CAN ALSO BE USED TO CHANGE
          BLINK STATUS, DISPLAY STATUS OR LIGHTPEN
          SENSITIVITY.  IF XNEW = MZERO (32768) THE
          ITEMS LOCATION IS NOT CHANGED.
```

```
PIKCLR(NCON)
          INSURES A CLEARED PICK QUEUE.


PUT(IREF,IDDT,IDDC,INFOA,INFOB,NCON)
          GENERAL ROUTINE FOR SENDING DISPLAY ITEMS
          TO THE IMLAC.


PUTBOX(NC)
          DRAWS A 12 INCH SQUARE ON THE SCREEN.


PUTBUT(X,Y,NCHAR,CHAR,IDDC,IREF,NCON)
          DEFINES A BUTTON DISPLAY ITEM AS A STRING
          OF CHARACTERS (CHAR ARRAY).


PUTPK(X,Y,NCHAR,CHAR,ICODE,IDDC,IREF,NCON)
          DEFINES SINGLE PICK DISPLAY ITEM AND DISPLAYS IT.


PUTREG(X,Y,NCHAR,NCON)
          PUTS A LIGHT REGISTER OF SIZE NCHAR ON
          THE SCREEN AT (X,Y).


PUTSPK(X,Y,NCHAR,CHAR,ICODE,IDDC,IREF,NCON)
          DEFINES A STRING PICK ITEM AND DISPLAY IT.


PTTWG(X,Y,NCHAR,ARRAY,IDDT,IDDC,ICODE,IREF,IGO,
          JDDC,IH,IV,NCON)
          SAME AS PUTWG EXCEPT IH,IV IS RETURNED FOR THE
          LAST PICK ON THE SCREEN.


PUTTXT(X,Y,NCHAR,CHAR,ICODE,IREF,NCON)
          PUTS A NON-PICKABLE DISPLAY ITEM
          ON THE SCREEN NCON.


PUTWG(X,Y,NCHAR,ARRAY,IDDT,IDDC,ICODE,IREF,IGO,JDDC,NCON)
          PUTWG PUTS AN NCHAR CHARACTER LIGHT REGISTER
          ON THE SCREEN AT X,Y ; WAITS FOR TYPE-IN
          OF INFORMATION ; GETS TYPED-IN INFO INTO ARRAY ;
          DISPLAY INFO AT IX,IY
          IGO = 2 IF BUTTON PICK IS MADE OUTSIDE JURISDICTION
                  OF THIS ROUTINE
          IGO = 1 OTHERWISE


PUTX(X,Y,NCON)
          PUTX PUTS T.C. AT COORDINATES (X,Y) ON SCREEN.


RANGEF(X,Y,NCON)
          RANGEF DETERMINES WHETHER OR NOT THE POINT(X,Y)
```

LIES WITHIN THE SPACE DEFINED BY THE FRAME SUBROUTINE THIS FUNCTION IS TRUE IF IN RANGE, FALSE OTHERWISE.

RANGES(X,Y,NCON)
DETERMINES WHETHER OR NOT THE POINT (X,Y) LIES WITHIN THE SPACE DEFINED BY THE SPACE SUBROUTINE. THIS FUNCTION IS TRUE IF WITHIN SPACE.

RTADJ(TEMP,NC,IL)
RTADJ WILL RIGHT ADJUST THE FIRST NC CHARS IN TEMP IN A FIELD OF THE 1ST IL CHARS OF TEMP.

SEND(N,A,B)
SENDS N WORDS FROM A TO B.

SETFKY(IDDT,IDDC,INFOA,INFOB,NCON)
DETERMINES HOW THE KEYBOARD (FNT-1) IS TO BE TREATED, I.E. AS IGNORE,SINGLE, STRING, OR BUTTON PICK.

SETPEN(IDDT,IDDC,INFOA,INFOB,NCON)
SETPEN MAKES IT POSSIBLE TO ADD SIGNIFICANCE TO THE ACT OF ACTIVATING THE LIGHTPEN SWITCH. A SWITCH IS DISPLAYED IN THE UPPER LEFT HAND CORNER OF THE IMLAC SCREEN.

SETEOM(ICODE,IDDT,IDDC,INFOA,INFOB,NCON)
SETS EOM CHARACTER ASSOCIATED WITH THE C.R. KEY AND DETERMINES HOW IT WILL BE TREATED.

SPACE(X1,Y1,X2,Y2,IH1,IV1,IH2,IV2,NCON)
SETS SUBJECT AND OBJECT SPACE BOUNDS.

STARTF(X,Y,ISTYLE,NCON)
STARTS A LINE FIGURE AND CONTROLS ITS APPEARANCE. THE FIGURE WILL APPEAR AT THE CURRENT BEAM POSITION.

START1(NUMB)
SETS THE GRAPHICS SYSTEM CONSOLE NUMBER FOR THE CONSOL WHICH THE USER WILL REFER TO AS CONSOLE 1.

START2(NUMB)
SAME AS START1, BUT FOR CONSOLE 2.

SUM(I,J,K)
COMPUTES K EQUAL TO I + J.

```
TEXT(X,Y,NCHAR,CHAR,ICODE,NCON)
          CREATES A STRING OF NCHAR FROM CHAR
          IN AN ITEM UNDER CONSTRUCTION.


UNDERL(IX1,IY1,NCHAR,ICODE,IREF,NCON)
          UNDERLINES NCHAR CHARS BEGINNING AT IX1,IY1.
          MUST BE CALLED WITH INTEGER VALUES FOR IX1,IY1.


USEMAC(MREF,N,NCON)
          PUTS MACRO CALL DESCRIPTION INTO AN ITEM DESCRIPTION
          BUFFER.  THE MACRO MUST HAVE BEEN CREATED BY
          MAKMAC.


WAIT(NCON,IR,IDDT,IDDC,INFOA,INFOB,IH,IV)
          WAITS FOR A BUTTON PICK OR RETURNS IMMEDIATELY IF
          NO QUEUED BYTTON PICK.
          IR = 0 MEANS WAIT FOR PICK
          IR = 1 MEANS RETURN IMMEDIATELY IF NO PICK


WAITB(IR,IDDT,IDDC,INFOA,INFOB,IH,IV)
          IR = 0 WAIT FOR INTERRUPT FROM EITHER CONSOLE
          IR = 1 LOOK FOR INTERRUPT, IF NONE RETURN
          NC = NUMBER OF INTERRUPTING CONSOLE


WAITMS
          PUTS WAIT MESSAGE ON INACTIVE CONSOLE
          ENTRY POINT GOMS PUTS UP A GO MESSAGE ON ACTIVE
          CONSOLE.


WRITED(INDATA,NW,FORMAT,OUTBUF,NC,X,Y,ICODE,IDDC,IREF,NCON)
          WRITE DISPLAY DECODES THE NW WORDS OF DATA WITH
          FORMAT AND PUTS THE RESULT INTO THE 1ST NC CHARS
          OF OUTBUF.  NC CHAR OF OUTBUFF ARE THEN DISPLAYED
          ON THE SCREEN AT (X,Y) ACCORDING TO ICODE.
          IF IDDC > 0 THEN THE DISPLAYED INFO IS TO BE
          TREATED AS A BUTTON.
```

APPENDIX I

OPERATIONAL VARIATIONS BETWEEN ITTR AND TTR

1. SCREEN FORMAT IS NOW 97 CHARACTERS PER LINE BY 62 LINES. THIS FORMAT IS DEPENDENT UPON THE IMLAC TIS4 HANDLER.

2. 504 GRAPHICS ITEMS MAY BE CREATED (252 PER TERMIMAL). 510 GRAPHICS MACROS MAY BE CREATED (255 PER TERMINAL).

3. IDDT = 16 DOES NOT BLINK AN ITEM WHEN PICKED.

4 GIERAS CAN ONLY DELETE A SINGLE DISPLAY ITEM.

5. THE LIGHT PEN SWITCH HAS BEEN IMPLEMENTED AS A BUTTON ON THE SCREEN. IF THE USER DESIRES TO USE THE FUNCTION KEYBOARD (FNT-1) AS A SWITCH, HE SHOULD CALL SETFKY. I.D. BLOCKS QUEUED BY THE FUNCTION KEYBOARD ARE IDENTICAL EXCEPT FOR IH AND IV WHICH CONTAIN THE KEY NUMBER STRUCK. TO USE A FUNCTION KEY AS A LIGHT PEN SWITCH AN APPLICATION PROGRAM WOULD NEED TO CHECK IH AND IV WHEN THE FUNCTION KEYBOARD I.D. BLOCK IS QUEUED.

6. WHEN STRUCTURES (MACROS) ARE INVOKED THE BEAM POSITION IS LEFT IN THE SAME PLACE IT WAS BEFORE THE MACRO WAS USED.

APPENDIX II

DEFINITIONS

FUNCTION KEYBOARD

THE 16-KEY FUNCTION KEYBOARD (FNT-1) CAN BE USED BY THE APPLICATION TO SIGNAL A REQUESTED OPERATION. ANY CHANGE IN THE STATUS OF THE KEY PRODUCES AN INTERRUPT TO THE 'TIS' IMLAC HANDLER.

ALPHANUMERIC KEYBOARD

THE APLHANUMERIC KEYBOARD PROVIDES TYPEWRITER-LIKE SYMBOLIC INPUT TO AN APPLICATION PROGRAM. KEY STRIKES CAUSE AN INTERRUPT TO THE 'TIS' HANDLER ONLY WHEN A LIGHT REGISTER IS PRESENT ON THE SCREEN.

LIGHTPEN

THE LIGHTPEN MAY BE USED FOR EITHER TRACKING OR

PICKING. TRACKING IS USED TO SPECIFY A LIGHT SOURCE POSITION ON THE SCREEN (MOVE THE TRACKING CROSS). TRACKING IS ONLY USABLE WHEN THE TRACKING CROSS IS VISIBLE ON THE SCREEN. PICKING MAY BE USED TO SELECT EITHER A BUTTON PICK, A SINGLE PICK, OR A STRING PICK ITEM.

LIGHT REGISTER

THE LIGHT REGISTER ALLOWS A GRAPHICS APPLICATION PROGRAM TO RETRIEVE ALPHANUMERIC INFORMATION. WHEN AN LIGHT REGISTER IS ACTIVE THE ALPHANUMERIC KEYBOARD MAY BE USED TO TYPE INFORMATION INTO THE LIGHT REGISTER. THE INFORMATION IS QUEUED WHEN THE USER TYPES THE CURRENT END OF MESSAGE CHARACTER. (USUALLY A CARRIAGE RETURN).

LIGHT BUTTONS

LIGHT BUTTONS ARE LIGHT SPOTS ON THE SCREEN WHICH MAY TAKE THE FORM OF A CHARACTER STRING, A LINE, A SYMBOL, OR ANY VISIBLE STRUCTURE.

SINGLE PICK

ONLY THE COPY OF THE ID BLOCK FOR THE LATEST SINGLE PICK DISPLAY ITEM CHOSEN IS KEPT IN THE QUEUE ASSOCIATED WITH A LIGHT BUTTON, REGARDLESS OF HOW MANY SUCH ITEMS ARE PICKED.

STRING PICK

ONE COPY OF A STRING PICK DISPLAY ITEM ID BLOCK IS KEPT IN THE QUEUE FOR EACH TIME SUCH AN ITEM IS PICKED.

BUTTON PICK

ONE COPY OF THE ID BLOCK FOR A LIGHT BUTTON IS KEPT IN QUEUE FOR EACH TIME SUCH AN ITEM IS PICKED.

ID BLOCK

AN IDENTIFICATION BLOCK OF CODED INFORMATION ASSOCIATED WITH A DISPLAY ITEM. (IDOT, IDDC, INFOA, INFOB)

SCISSOR

DROPPING AN ENTITY FROM THE DISPLAY WHEN ITS COORDINATES EXCEED A PRESET RANGE ON THE DISPLAY

GRID.

ITEM

A DISPLAYABLE BYTE STREAM WHICH CONTAINS THE BEGINNING COORDINTATES WHERE THE INFORMATION IN THE BYTE STREAM IS TO APPEAR.

MACRO

A DISPLAY BYTE STREAM WHICH CAN APPEAR IN A NUMBER OF LOCATIONS ON THE SCREEN WITHOUT DUPLICATION OF THE BYTE STREAM.

PICK

THE SELECTION OF AN ITEM WITH THE LIGHTPEN OR THE FUNCTION KEYBOARD.

APPENDIX III

PROCEDURE FOR RUNNING THE TESTPGM ITTR TEST

TESTPGM HAS BEEN DESIGNED TO TEST THOSE FEATURES OF THE IMLAC TWO TERMINAL ROUTINES NOT SPECIFICALLY TESTED BY G1 AND G2. TESTPGM IS AN EXTENSION OF THE G1 TEST PROGRAM. IT CONTAINS A NUMBER OF MODIFICATIONS WHICH ALLOW TESTING OF MODULES NOT TESTED BY THE ORIGINAL BENCHMARK.

FOLLOWING IS A LIST OF ITTR SUBROUTINES TESTED BY THE TESTPGM TEST PROGRAM:

START1	SPACE	PUTBOX	PUTTXT
DEFPOS	CIRCLE	PUT	PUTX
PUTBUT	PUTSPK	PUTPK	WAIT
GIFID	GIFSID	GETLID	UGETX
DTCHX	ATCHX	GETCXY	CLEAN
PIKCLR	FRAME	LINCZR	LINSEG
CONTIL	PUTX	STARTF	MAKMAC
COPY	MATCHX	MDTCHX	SETFKY
GIERAS	PTTWG	TEXT	BUILD
FLIP	COPYF	GOMS	WAITMS
WAITB	CSWTCH	USET	

TO CHECK PROPER OPERATION OF THE ROUTINES LISTED ABOVE, FOLLOW THE PROCEDURE OUTLINED BELOW:

1. START PROGRAM EXECUTION BY SELECTING 'TESTPGM' ON THE NEARBY ADM-3 VIDEO DISPLAY TERMINAL. FOLLOW THE G1 BENCHMARK PROCEDURES. PICK 'FINAL TEST' ON THE INITIAL INPUT PAGE.

2. SOME HEADER INFORMATION WILL APPEAR AT THE TOP OF THE SCREEN; THE PICK TEST BLOCK APPEARS AT THE BOTTOM LEFT HAND CORNER; SEVERAL PICKABLE ITEMS APPEAR AT THE RIGHT SIDE OF THE SCREEN; A CIRCLE APPEARS IN THE MIDDLE OF THE SCREEN; THE TRACKING CROSS APPEARS IN THE CENTER OF THE CIRCLE.

3. PICK 'ATTACH FRA' TO ATTACH THE CIRCLE TO THE THE TRACKING CROSS. MOVE THE LIGHT PEN NEAR THE TRACKING CROSS AND THEN SLOWLY MOVE THE LIGHT PEN ACROSS THE SCREEN. THE CIRCLE WILL MOVE WITH THE TRACKING CROSS.

4. PICK 'DETACH TC' AND AGAIN MOVE THE TRACKING CROSS . THE CIRCLE WILL NO LONGER MOVE WITH THE TRACKING CROSS.

5. MOVE THE TRACKING CROSS TO A POSITION WHERE THERE IS AT LEAST 2 INCHES OF CLEARANCE. NOW PICK 'MAKE COPY'. A CIRCLE WILL BE COPIED TO THE SCREEN AT THE CURRENT TRACKING CROSS LOCATION. MOVE THE TRACKING CROSS TO A NEW LOCATION AND AGAIN PICK 'MAKE COPY'. ANOTHER CIRCLE WILL APPEAR.

6. PICK 'DEL TRACK'. THE TRACKING CROSS WILL DISAPPEAR FROM THE SCREEN.

ID BLOCK QUEUING TEST

DEFINITIONS

SINGLE PICK

ONLY THE COPY OF THE ID BLOCK FOR THE LATEST SINGLE PICK DISPLAY ITEM CHOSEN IS KEPT IN THE QUEUE ASSOCIATED WITH A LIGHT BUTTON, REGARDLESS OF HOW MANY SUCH ITEMS ARE PICKED.

STRING PICK

ONE COPY OF A STRING PICK DISPLAY ITEM ID BLOCK IS KEPT IN THE QUEUE FOR EACH TIME SUCH AN ITEM IS PICKED.

BUTTON PICK

ONE COPY OF THE ID BLOCK FOR A LIGHT BUTTON IS KEPT IN QUEUE FOR EACH TIME SUCH AN ITEM IS PICKED.

'TYPE' IS THE PICK TYPE

'IDDT' IS THE TYPE CODE
IDDT = 2 SINGLE PICK
IDDT = 4 STRING PICK
IDDT = 8 BUTTON PICK

'IDDC' IS THE DISPLAY ITEM IDENTIFICATION CODE. IN THE FOLLOWING TEST THIS NUMBER APPEARS IN PARENTHESIS TO THE RIGHT OF 'SINGLE PCK', 'STRING PICK', AND 'BUTTON PCK'.

'IH' IS THE HORIZONTAL COORDINATE WHERE THE PICK OCCURED. REMEMBER THAT IGS COORDINATES ARE (-1248,-1248) TO (1248,1248).

'IV' IS THE VERTICAL COORDINATE WHERE THE PICK OCCURRED.

7. PICK 'STRING PCK' (BOTTOM CENTER OF SCREEN).

8. PICK 'SINGLE PCK'

9. PICK 'STRING PCK'

10. PICK 'STRING PCK'

11. PICK 'BUTTON PCK'

12. THE CURRENT QUEUE STATUS WILL BE DISPLAYED AT THE BOTTOM LEFT CORNER OF THE SCREEN. THE BUTTON PICK ID INFO, WHICH CAUSED A COMPLETE QUEUE TO BE ASSEMBLED, WILL BE DISPLAYED FIRST, FOLLOWED BY THE SINGLE PICK ID INFO. NEXT THE THREE STRING PICK ID BLOCKS WILL BE DISPLAYED. FINALLY, THE ORIGNINAL BUTTON PICK INFO WILL BE DISPLAYED AGAIN TO SHOW THE PROPER FUNCTIONING OF THE GETLID SUBROUTINE.

THE DISPLAY APPEARS AS FOLLOWS:

TYPE	IDDT	IDDC	IH	IV
BUTTON	8	6	--	--
SINGLE	2	7	--	--
STRING	4	8	--	--
	4	8	--	--
	4	8	--	--
GETLID	8	6	--	--

NOTE THAT IH AND IV CONTAIN THE EXACT COORDINATES OF THE LOCATION WHERE THE PICK TOOK PLACE. THE SPECIFIC NUMBERS WILL VARY FROM TRIAL TO TRIAL. THE COORDINATES DISPLAYED ARE NORMAL IGS OBJECT SPACE COORDINATES.

13. PICK 'NEW QUEUE'. THE SCREEN WILL BE CLEARED AND THE INITIAL INPUT PAGE WILL AGAIN BE DISPLAYED WITH THE QUEUE INFORMATION CLEARED.

14. REPEAT STEPS 7-12.

15. PICK 'NEXT TEST'. AN UNUSUAL FIGURE WILL APPEAR IN THE CENTER OF THE SCREEN. STARTING AT IGS COORDINATES (-500,-500) A SCISSORED LINE SEGMENT OF THE 'CENTER-LINE' TYPE IS DISPLAYED. THE SCISSORED COORDINATES OF THIS LINE ARE PRINTED ON THE 'LO' FILECODE. SEGMENT #2 IS OF TYPE 'CENTER-LINE' PRODUCED WITH A CALL TO CONTIL. SEGMENT #3 IS A LINE OF TYPE 'BROKEN'. SEGMENT #4 ILLUSTRATES THE 'DASHED' LINE TYPE.

16. PICK 'ATTACH MACRO'. AN ODD FIGURE WILL APPEAR AT THE LOCATION OF THE TRACKING CROSS.

17. MOVE THE TRACKING CROSS AND NOTICE THAT THE FIGURE MOVES IN THE DIRECTION OF TRACKING CROSS MOVEMENTS.

18. PICK 'DETACH MACRO'. THE MACRO WILL DISAPPEAR.

19. PICK 'END PROGRAM'. THE SCREEN WILL CLEAR AND PROGRAM EXECUTION WILL TERMINATE.

SOME ADDITIONAL TESTPGM INFORMATION

1. CURRENT PICK QUEUE STATUS IS DISPLAYED ONLY AFTER A BUTTON PICK. UP TO FIVE STRING PICKS MAY BE DISPLAYED ON THE IMLAC SCREEN. BY PICKING THE 'STRING PICK' AT DIFFERENT X COORDINATE LOCATIONS EACH TIME, ONE CAN EXAMINE THE VARIATION IN THE IH COORDINATE RETURNED TO THE HOST. IT IS NECESSARY TO PICK 'NEW QUEUE' BEFORE A NEW QUEUE CAN BE ASSEMBLED AND DISPLAYED.

2. 'MAKE COPY' COPIES A CIRCLE TO THE CURRENT TRACKING CROSS LOCATION.

3. PICKING 'NEXT TEST' STARTS A NEW DISPLAY PAGE WHICH SHOWS LINE TYPES, SCISSORING, AND MACROS USED WITH THE TRACKING CROSS. THE COORDINATES OF THE SCISSORED LINE ARE PRINTED ON THE 'LO' FILECODE.

4. PICKING ATTACH WHEN THE MACRO IS ALREADY ATTACHED CAUSES THE MACRO TO BE LEFT AT CURRENT LOCATION AND A NEW MACRO TO BE ATTACHED. THIS IS BECAUSE MACROS ARE ATTACHED AS ITEMS AFTER ANY EXISTING ATTACHED ITEM IS DETACHED. DETACHING (VIA A PICK OF 'DETACH MACRO') CAUSES THE ATTACHED MACRO TO DISAPPEAR (AS SPECIFIED IN TTR DOCUMENTATION).

APPENDIX IV

USING G1 AND G2 TO TEST ITTR

CATALOGING INFORMATION FOR THE G1 AND G2 BENCHMARK PROGRAMS

IN ORDER TO CATALOG A GRAPHICS TASK THE FOLLOWING FILE ASSIGNMENTS MUST BE MADE:

HF1 SHOULD BE ASSIGNED TO A HISTORY FILE

THE HISTORY FILE MUST BE CREATED AS FOLLOWS:

```
CREATE M.TTRHF1,DC,2000
PARTIT M.TTRHF1,1,510,1400
PARTIT M.TTRHF1,2,510,1400
```

GT1 SHOULD BE ASSIGNED TO TERMINAL NCON=1
GT2 SHOULD BE ASSIGNED TO TERMINAL NCON=2

CATALOGING THE G1 BENCHMARK

G1 IS CATALOGED AS A MAIN SEGMENT WITH TWO OVERLAY LEVELS (6 OVERLAYS). THE MODULE IS STRUCTURED NEARLY IDENTICAL TO THE WAY IT ORIGINALLY WAS ON THE C.D.C.

A SAMPLE CATALOG PROGRAM USED TO CREATE THE G1 OBJECT MODULE IS REPRODUCED BELOW.

```
$JOB CATG1
$OBJECT
(OBJECT MODULE OF G1)
$ALLOCATE 25000
$EXECUTE CATALOG
ASSIGN1 HF=M.TTRHF1
ASSIGN2 LO=SLO,1000
ASSIGN4 1=LO
ASSIGN3 5=TY73
CATALOG BMARK157 RT U 4 NOM
INCLUDE CRT2TERM
INCLUDE GCSBLOCK
PROGRAM BMARK1 BUILD
CATALOG G1010EX OV NOM
LINKBACK BMARK157
PROGRAM CNTRL
CATALOG G1020EX OV NOM
LINKBACK BMARK157
PROGRAM WAITER
LORIGIN G1020EX
CATALOG G1021EX OV NOM
LINKBACK G1020EX BMARK157
PROGRAM CHARMM
CATALOG G1022EX OV NOM
LINKBACK G1020EX BMARK157
PROGRAM VECTOR
CATALOG G1023EX OV NOM
LINKBACK G1020EX BMARK157
PROGRAM TCROSS
CATALOG G1024EX OV NOM
LINKBACK G1020EX BMARK157
PROGRAM FKEY
$EOJ
$$
```

CATALOGING THE G2 BENCHMARK

G2 IS CATALOGED AS A MAIN SEGMENT WITH ONE OVERLAY LEVEL. (4 OVERLAYS) A SINGLE OVERLAY LEVEL WAS CHOSEN TO SIMPLIFY THE CONVERSION OF C.D.C. CALLS TO OVERLAY UTILITIES.

A SAMPLE CATALOG PROGRAM USED TO CREATE THE G2 OBJECT MODULE IS REPRODUCED BELOW.

```
$JOB CATG2
$OBJECT
(OBJECT MODULE OF G2)
$ALLOCATE 25000
$EXECUTE CATALOG
ASSIGN2 VER=SVO,1000
ASSIGN2 XYP=SXO,1000
ASSIGN3 GT1=TY75
ASSIGN1 HF=M.TTRHF1
ASSIGN1 77=TAPE77
ASSIGN2 1=SLO,1000
ASSIGN2 LO=SLO,1000
CATALOG BMARK257 RT U 4 NOM
PROGRAM PAGE0
PROGRAM DRIVE TSK1 RTRVM TWO
PROGRAM STASK1 QWERTY DISPLS
PROGRAM XOVERLY
INCLUDE CRT2TERM
INCLUDE GCSBLOCK
INCLUDE V:COM1
INCLUDE DBFETCH
CATALOG G21150EX OV NOM
PROGRAM SPAGE2 CURVES CHANGE
PROGRAM SPAGE1 SDSPLY Q
CATALOG G21127EX OV NOM
PROGRAM DAMP SWCALC AREAS RTHN
PROGRAM PACT
INCLUDE AREA UNPACT DASDIO UNIQID FETCH SRCH CIRRD UNPACK
CATALOG G21145EX OV NOM
PROGRAM SCURVE TSCALE
PROGRAM MAXE
$EOJ
$$
```

CHANGES IN THE LOGIC OF G2

1. THE VARIABLE IY IS NOW RESET TO IY=20 EACH TIME SUBROUTINE DRIVE IS CALLED. THIS PREVENTS THE INITIAL INPUT PAGE FROM DISAPPEARING OFF THE BOTTOM OF THE PAGE WHEN DRIVE IS CALLED MORE THAN ONCE.

2. ONLY 'SURPRISE' AND 'TERMINATE JOB' CAN BE PICKED ON THE INITIAL INPUT PAGE.

'TERMINATE JOB ' NOW CLEARS THE SCREEN.

3. THE DO LOOP AT LINE 16 OF SPAGE2 HAS BEEN FIXED TO PREVENT OVERWRITING MEMORY. IVAL IS DIMENSIONED IVAL(6), BUT THE DO LOOP AT 16 ZEROED 10 ELEMENTS OF IVAL.

4. THE CODE IN SCURVE HAS BEEN CHANGED TO PREVENT A CALL TO MOVIT WHEN THE ENDJOB BUTTON IS PICKED. THE ENDJOB BUTTON HAS A IDDC OF 27. THE PROGRAM CORRECTLY CALLS CLEAN WHICH CLEARS BOTH THE SCREEN AND TABLES. CALLING MOVIT (BELOW LINE 150) AFTER THE TABLES HAVE BEEN CLEARED RESULTED IN ABNORMAL TERMINATION OF THE JOB.

5. CODE IN SDSPLY HAS BEEN CHANGED TO PREVENT PICKING ANY ITEM OTHER THAN EOM. WHEN THE PROG. BEGINS EXECUTION THE VALUES WHICH ARE REQUESTED BY SDSPLY (CALLED BY SPAGE1) CONTAIN RANDOM DATA. A PICK OTHER THAN EOM WOULD CAUSE THIS RANDOM DATA TO BE WRITTEN TO THE SCREEN. (WRITED CALLS ENCODE WHICH BOMBS) THE GOTO 55 BELOW LINE 20 HAS BEEN CHANGED TO TRANSFER CONTROL TO 'CALL WAIT' IF ANYTHING OTHER THAN EOM IS PICKED.

6. SPAGE2 HAS BEEN CHANGED TO PREVENT OVERFLOW AND UNDERFLOW WHEN THE BLOCK OF CALCULATIONS BELOW LINE 155 ARE EXECUTED BY THE SEL.