NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

THIS DISSERTATION HAS BEEN MICROFILMED EXACTLY AS RECEIVED

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30. Veuillez prendre connaissance des formules d'autorisation qui accompagnent cette thèse.

LA THÈSE A ÉTÉ MICROFILMÉE TELLE QUE NOUS L'AVONS RÉCU
PERMISSION TO MICROFILM — AUTORISATION DE MICROFILMER

Please print or type — Écrire en lettres moulées ou par typographie

Full Name of Author — Nom complet de l'auteur

Date of Birth — Date de naissance

Country of Birth — Lieu de naissance

Permanent Address — Résidence fixe

University — Université

Degree for which thesis was presented — Grade pour lequel cette these fut présentée

Year this degree conferred — Année d'obtention du grade

Name of Supervisor — Nom du directeur de these

Permission is hereby granted to the NATIONAL LIBRARY OF CANADA to microfilm this thesis and to lend or sell copies of the film

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

Date

Signature

NL-91 (4/77)
AUTOMATED STRUCTURAL DRAFTING
FOR BUILDINGS

by

PATRICIA VALENZUELA

A thesis submitted to the Faculty of
Graduate Studies and Research in partial
fulfillment of the requirements for the
degree of

Master of Engineering

Department of Civil Engineering
Carleton University
Ottawa, Ontario
April 2, 1984
The undersigned recommend to the Faculty of Graduate Studies and Research acceptance of the thesis

"AUTOMATED STRUCTURAL DRAFTING FOR BUILDINGS"

submitted by Patricia Valenzuela, C.E., University of Chile in partial fulfillment of the requirements for the degree of Master of Engineering

[Signature]

THEESIS SUPERVISOR

[Signature]

CHAIRMAN, DEPARTMENT OF CIVIL ENGINEERING

Carleton University
April 2, 1984
ABSTRACT

This thesis deals with a computer program developed to produce graphic displays of structures. The program is integrated into AMECO-17 system of structural engineering programs.

The graphic displays range from single line drawings to finished engineering drawings. The latter consist of framing plans and elevations for steel and concrete structures drawn in accordance with manual drafting standards.

The design results of the AMECO-17 system are directly passed on to the graphic program. Thus the drawings are fully derived within the system, requiring no operator input.

The graphic information can be displayed on a variety of CRT terminals, directly plotted, or transmitted to an interactive graphics system for further processing.

The system is presently operational on Control Data Corporation CYBER series computers.
ACKNOWLEDGEMENTS

The author expresses her gratitude to Prof. J. Humar who supervised the work and provided guidance.

Dr. W. Wright suggested this area of research and gave many helpful suggestions during the course of the work.

The author also wishes to thank Mr. A. A. Palejs, principal author of AMECO-17 system, for guidance and continuing support throughout the project.

Grateful acknowledgement is given to Multiple Access Division of CSG Ltd., specially to Mr. R. Crossan and Mr. J. Kern, for the permission to use their computing facilities in the Montreal offices.

Appreciation is extended to Mrs. M. Voitkus who typed major part of this thesis.
TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>TITLE PAGE</td>
<td>i</td>
</tr>
<tr>
<td>ACCEPTANCE SHEET</td>
<td>ii</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>iii</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>iv</td>
</tr>
<tr>
<td>TABLE OF CONTENTS</td>
<td>v</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>viii</td>
</tr>
<tr>
<td>CHAPTER 1 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Background</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Objective of this Project</td>
<td>2</td>
</tr>
<tr>
<td>CHAPTER 2 OUTLINE OF THE PROJECT</td>
<td>5</td>
</tr>
<tr>
<td>2.1 Overview of AMECO-17 System</td>
<td>6</td>
</tr>
<tr>
<td>2.2 Overview of the Graphic Module</td>
<td>10</td>
</tr>
<tr>
<td>CHAPTER 3 DESCRIPTION OF THE GRAPHIC MODULE</td>
<td>13</td>
</tr>
<tr>
<td>3.1 Introduction</td>
<td>13</td>
</tr>
<tr>
<td>3.2 Computation of 2-D Coordinates</td>
<td></td>
</tr>
<tr>
<td>for Vertical Frames</td>
<td>16</td>
</tr>
<tr>
<td>3.3 Computation of 3-D coordinates</td>
<td></td>
</tr>
<tr>
<td>for Orthogonal Structures</td>
<td>19</td>
</tr>
<tr>
<td>3.4 Computation of 3-D Coordinates</td>
<td></td>
</tr>
<tr>
<td>for Non-orthogonal Structures</td>
<td>23</td>
</tr>
<tr>
<td>3.5 Generation of Building Grid Lines</td>
<td>24</td>
</tr>
<tr>
<td>3.6 Drawing Sizes, Border and Title Block</td>
<td>25</td>
</tr>
<tr>
<td>3.6.1 Border</td>
<td>26</td>
</tr>
<tr>
<td>3.6.2 Title Block</td>
<td>27</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>1.1</td>
<td>Integration of Analysis, Design and Drafting Functions</td>
</tr>
<tr>
<td>2.1</td>
<td>AMECO Integrated Analysis/Design Process</td>
</tr>
<tr>
<td>2.2</td>
<td>AMECO Integrated Analysis/Design Process with Graphics</td>
</tr>
<tr>
<td>3.1</td>
<td>Orthogonal Structure</td>
</tr>
<tr>
<td>3.2</td>
<td>Non-orthogonal Structure</td>
</tr>
<tr>
<td>3.3</td>
<td>Joint Numbering</td>
</tr>
<tr>
<td>3.4</td>
<td>Computation of 2-D Coordinates</td>
</tr>
<tr>
<td></td>
<td>a) Plane Frame with Col. 1 as Reference Stack</td>
</tr>
<tr>
<td></td>
<td>b) Plane Frame with Col. 2 as Reference Stack</td>
</tr>
<tr>
<td>3.5</td>
<td>Column Stack Coordinates for an Orthogonal Structure</td>
</tr>
<tr>
<td>3.7</td>
<td>Standard Border for a Drawing</td>
</tr>
<tr>
<td>3.8</td>
<td>Title Block for a Drawing</td>
</tr>
<tr>
<td>3.9</td>
<td>Standard Margins</td>
</tr>
<tr>
<td>3.10</td>
<td>Multiple Views on a Drawing</td>
</tr>
<tr>
<td>3.12</td>
<td>Conventions used in Drawing the Beams</td>
</tr>
<tr>
<td>3.13</td>
<td>Vertical Bracing System and Walls</td>
</tr>
<tr>
<td></td>
<td>a) Types of Bracing Members</td>
</tr>
<tr>
<td></td>
<td>b) Frame with 'V' Braces and Walls</td>
</tr>
<tr>
<td>3.14</td>
<td>Column Sizes, Splice Points and Column Marks</td>
</tr>
<tr>
<td>3.16</td>
<td>Framing Elevations</td>
</tr>
<tr>
<td>3.17</td>
<td>Steel Building Plan</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>3.18</td>
<td>Floor Plan for a Steel Building showing Secondary Beams</td>
</tr>
<tr>
<td>3.20</td>
<td>Floor Plan for a Steel Building showing column marks and Beam Marks</td>
</tr>
<tr>
<td>3.21</td>
<td>Dimension Lines, Grid Lines and Floor Plan Title</td>
</tr>
<tr>
<td>3.22</td>
<td>Floor Plan of a Concrete Structure</td>
</tr>
<tr>
<td>3.23</td>
<td>Floor Plan of a Concrete Structure showing the Representation of Slabs</td>
</tr>
<tr>
<td>4.1</td>
<td>PLOT Command Facsimile</td>
</tr>
<tr>
<td>4.2</td>
<td>AMECO Integrated Analysis/Design Process with Graphic</td>
</tr>
<tr>
<td>5.1</td>
<td>Member Connectivity Conventions</td>
</tr>
<tr>
<td></td>
<td>a) AMECO</td>
</tr>
<tr>
<td></td>
<td>b) Conventional Programs</td>
</tr>
<tr>
<td>5.2</td>
<td>Calculation of Column Stack Coordinates</td>
</tr>
<tr>
<td>5.3</td>
<td>A Case where Stack Coordinates cannot be Computed</td>
</tr>
<tr>
<td>5.4</td>
<td>Location of Views on Drawings</td>
</tr>
<tr>
<td>5.7</td>
<td>Example of Nested Secondary Beams</td>
</tr>
<tr>
<td>5.8</td>
<td>Example of Identical Beam Lines</td>
</tr>
<tr>
<td>5.9</td>
<td>Position of Column Marks</td>
</tr>
<tr>
<td>5.10</td>
<td>Convention used in Drawing Grid Lines</td>
</tr>
<tr>
<td>5.11</td>
<td>A Non-orthogonal Structure</td>
</tr>
<tr>
<td>5.12</td>
<td>A Complex Non-orthogonal Structure</td>
</tr>
<tr>
<td>B.1</td>
<td>Flow Chart of the Program Section for Plotting the Framing Elevations</td>
</tr>
<tr>
<td>B.2</td>
<td>Flow Chart of the Program Section for Plotting the Floor Plans</td>
</tr>
<tr>
<td>B.3</td>
<td>Flow Chart of Subroutine DIMS</td>
</tr>
<tr>
<td>B.4</td>
<td>Example of Internal Grid Line</td>
</tr>
</tbody>
</table>
CHAPTER 1
INTRODUCTION

1.1 Background

During the past two decades great advances have been made in the application of computer graphics to many fields of human endeavor. The structural engineering profession is no exception, especially since the final product of a design engineering office are drawings - graphical displays of the structure to be built.

As early as the mid-sixties graphic output capabilities were developed for many structural analysis programs, such as STRESS - 3(1), STARDYNE(2), ICES-STRUCL(3) and others. A few years later, similar capabilities were incorporated in the well known finite element programs, such as NASTRAN(4), MASAP(5), ANSYS(6) and others. In their basic form, these graphic displays show to scale line diagrams of the structural model with joint and member numbers on the diagrams. Deformed shapes of the model, shear and moment diagrams, and stress contours are typical other displays obtainable from these programs. For the structural design engineer, these graphic developments are invaluable for verifying the accuracy of the structural model and for evaluating the analysis results.
Computer aided preparation of engineering drawings came of age in the early seventies with the advent of interactive graphics. In their simplest form, interactive graphic systems can be looked upon as super-fast drafting machines. Today, the more advanced turn-key systems, such as APPLICON, AUTOTROL, CALMA, COMPUTER-VISION, described by W. Wright(7), have emerged as very powerful graphic tools for the rapid creation of drawings. These systems are built around a large Data Base, that can for example, store complete tables of structural shape properties, a large variety of drafting symbols and character fonts, standard details and texts, etc., all quickly accessible by referencing a menu. Complete drawings can be stored in the Data Base for future retrieval, revision and reproduction.

These graphic systems are very effective for schematic work, such as circuit diagrams, cable routing layouts, piping isometrics. The efficiency of such systems drops in the preparation of drawings that have to be made to scale, especially when a large amount of design information has to be placed on the drawing. For example, a typical steel floor framing drawing can have over 250 members - beams, columns and bracing members. In such a case the input of member sizes alone can become a very lengthy and error prone operation.

It is obvious that a desirable step would be to eliminate this
manual operation of transferring data from design documents to the graphic system.

1.2 Objective of this Project

A structural engineering computer system, where the results of a design program are directly passed on to graphic program for production of drawings is illustrated in Fig. 1.1. Here, the drawings are fully created within the system, requiring practically no operator input.

The objective of this project was to develop a graphic program and to integrate it into AMECO - 17(8),(9) system of structural engineering programs. AMECO - 17 is a system of integrated structural engineering programs for the analysis and automated design of buildings, bridges and similar structures having steel, concrete or composite members. From the design data produced by AMECO - 17, the graphic module should automatically prepare the following types of design engineering drawings:

1. Framing plans and elevations for steel buildings
2. Framing plans for concrete buildings

These drawings should be as complete as possible, meet the manual drafting standards and be of contract drawing quality.
FIGURE 1.1 Integration of Analysis, Design and Drafting Functions
CHAPTER 2

OUTLINE OF THE PROJECT

This chapter presents a brief description of the external characteristics of the AMECO-17 system, followed by an overview of the graphic module.

The "vertical" integration of the graphic module into AMECO-17 is illustrated by flow charts.
2.1 Overview of AMECO-17 System

AMECO-17 is a system of integrated structural engineering computer programs for the analysis and automated design of buildings, bridges and similar structures having steel, concrete or composite members.

The result of 40 men years of effort, the AMECO system permits the computerization and integration of all design activities - from loading and geometry generation through member design to project cost estimates. Therefore, an entire three-dimensional building of any size can be designed with one pass on the computer.

The AMECO multi-cycle analysis/design flowchart is shown in Fig. 2.1.

The user of AMECO is provided with a concise command language to input specifications for geometry, design criteria and loads.

Phase A of the multi-cycle process interprets the input commands, checks them for syntax and other errors, edits geometry and proceeds to Phase B if no fatal errors are encountered (see Fig. 2.1).

Phase B computes all gravity loads, member stiffnesses, and performs linear analysis by floor segments to establish the
INPUT EDIT
GEOMETRY GENERATION

GEOMETRY EDIT

GRAVITY LOAD GENERATION

APPROXIMATE STRUCTURAL ANALYSIS

MEMBER DESIGN OR CHECKING

LOADING UPDATE

YES SIZES REVISITED

DYNAMIC STRUCTURAL ANALYSIS

STATIC STRUCTURAL ANALYSIS

MEMBER DESIGN OR CHECKING

LOADING UPDATE

NO CYCLE = n

YES SIZES REVISITED

OUTPUT REPORTS

PHASE 'A'

PHASE 'B'

PHASE 'C'

PHASE 'D'

FIGURE 2.1 AMECO integrated analysis/design process
initial member sizes of the structure.

Phase C carries out three-dimensional analysis by a method of joint relaxation (10), followed by member proportioning.

Four structural analysis procedures are available:
1. Linear elastic analysis.
2. Nonlinear geometric analysis including the P-Delta effect.
3. Segmental analysis to simulate the erection of the structure.

For a given structure any one of these procedures or all four of them can be requested.

Member size selection is a code dependent process. For that purpose the requirement of 12 national standards (codes) for the design of concrete and steel members are incorporated into AMECO, using WSD, USD, LSD, Plastic and Ductile design methods.

Three system of units are available: English, Metric and SI.

Concrete structure may consist of flat plates, flat slabs, voided slabs, joist slabs or beam and slab systems with columns and/or shear walls supported on footings on elastic soil.

Steel structures may have composite or noncomposite beams, pinned or continuous, open web steel joists, trusses, columns with flexible or fully rigid connections. Resistance to
horizontal forces is provided by rigid frame action, bracing members or shear walls. Foundation support is provided by footings on elastic soil.

Phase C is repeated as many times as necessary until either the changes in member properties have become less than the prescribed tolerances or the number of cycles specified by the user is reached.

Phase D retrieves from disk file the computational results produced by Phase C and outputs reports requested by the user.

Fig. 2.1 illustrates the automatic process of the program. There are numerous commands available to the user to execute specific modules, to stop the process in any one of the phases, review and save intermediate results and then retrieve these at a later date to restart and to continue the design process.
2.2 Overview of the Graphic Module

The graphic module developed in this project has been designed to produce visual displays of a structure of varying complexities - ranging from scaled, single line diagrams to framing plans and elevations of contract drawing quality. The structure can be of steel or concrete.

The module is totally integrated in AMECO-17 system. AMECO supplies the graphic module with all the data necessary to start the graphic computations. Therefore, no additional information has to be supplied by the user to produce graphical displays of the structure. Simple commands have been developed for the user to select the desired views, to control the amount of information to be displayed and to specify at what phase of the design process these views should be displayed.

Fig. 2.2 illustrates the positions in the AMECO analysis/design process where the module can be called upon to produce graphic displays.

Position I is normally used to display scaled single line diagrams for the purpose of verifying the input data.

Position II can be used to monitor the design process by
FIGURE 2.2 AMECO Integrated analysis/design process with graphics
displaying diagrams with member sizes on them for each design cycle.

Position III is normally activated to produce framing plans and elevations with the final member sizes.

For a medium to large size building, however, it is more practical to bypass Position II and save all results on disk files. These results can be retrieved at a later date and the drawings produced at Position I.

The graphic module is written in ANSI-77 Fortran. It consists of one main program and 30 subroutines and has a total of 6300 Fortran statements. The graphic module uses the Calcomp subroutines for producing the graphical information.
CHAPTER 3

DESCRIPTION OF THE GRAPHIC MODULE

3.1 Introduction

In AMECO a three dimensional structure is internally represented by a series of intersecting two dimensional vertical frames, designated as the X and Y frames. In a true orthogonal structure, the X set consists of frames parallel to the X-global axis while the Y set consists of frames parallel to the Y-global axis, as shown in Fig.3.1. In a nonorthogonal structure the AMECO representation is identical, except, that some or all of the quasi X or Y frames are no longer parallel to the global axis. In fact the frames can even be on a curved surface as illustrated by frame X-7 in Fig.3.2.

There are two data tables in AMECO that are used to initiate the geometric computations. One table contains information pertaining to all X frames while the other has information on all Y frames.

These tables store: member lengths, direction cosines, member and frame connectivities and member properties.
FIGURE 3.1 Orthogonal Structure.

FIGURE 3.2 Non-orthogonal Structure
Frame on a curved surface
Joint coordinates are not computed in AMECO, since they are not required in a structural analysis by joint relaxation. Therefore, the first task in obtaining graphical displays of the structure is to compute the joint coordinates. This is done in two steps. The joint coordinates of each two-dimensional frame are computed first, using a local X-Z system of axes for each frame. The global X, Y, Z joint coordinates for all beam-column connections are then obtained in the second step.

Secondary beam joint coordinates (beam to beam connections) are computed at a later stage, while creating the plot files for floor plans.
3.2 Computation of Two Dimensional Joint Coordinates for each Vertical Frame.

Subroutine PLOTF positions each frame in a matrix, with as many columns as there are column lines in the frame being processed and as many rows as there are levels in the entire structure. Processing starts with the first frame encountered in the AMECO file of frames. All X frames are processed first followed by all Y frames.

The joints in a frame are numbered in sequence along each column stack, starting from the foundation level and proceeding upwards as shown in Fig. 3.3

![Diagram of joint numbering](image)

**FIGURE 3.3 Joint numbering.**
AMECO assumes that the building will have at least one level to which all columns are attached (reference level). The user should ensure that this is so, if necessary by providing suitable dummy columns.

The following steps are involved in the computation of the joint coordinates of each vertical frame, using a local X-Z system of axes:

- For each frame subroutine PLOTS selects as a reference stack the first column stack that has no dummy columns between levels and then sets the origin at the intersection of the reference level with the reference stack.

For a reference stack:
- Subroutine STACDM5 computes the coordinates of all joints below the reference level.
- Subroutine STACUP5 computes the coordinates of all joints above the reference level.

Once all the joint coordinates in the reference stack have been computed, subroutine STACRI5 is called, to move to the next column stack to the right.

This procedure is repeated till all the joint coordinates of the last stack of the frame have been computed. Fig. 3.4(a) illustrates this procedure.
The same method is followed when the first column stack is not the reference stack, except that now the missing joint coordinates to the left of the reference stack are to be computed using subroutine STACLE5 instead of subroutine STACRI5. Fig 3.4(b) illustrates this case.

If the origin selected for the computation of coordinates was not at the intersection of the first column stack with the reference level because of dummy columns in that stack, the origin is now repositioned so that it is at the intersection of the first column stack with the reference level. The coordinates values are modified so that they refer to the repositioned origin. The coordinates information is stored in a disk file.

The above procedure is valid for all X and Y frames.

![Diagram](image)

(a) Plane Frame with column 1 as ref. stack
(b) Plane Frame with column 2 as ref. stack

FIGURE 3.4 Computation of 2-D Coordinates

Note: The arrows and numbers represent the consecutive steps in computing the 2-D joint coordinates.
3.3 Computation of Three Dimensional Joint Coordinates for Orthogonal Structures.

The computation of three dimensional joint coordinates begins at the reference level, using X, Y, Z axes. The origin is set at the first column stack of the first X frame encountered in the AMECO file (reference frame).

The three dimensional joint coordinates are computed as follows:

1) Set the global X coordinates of each column stack of the frame as equal to the corresponding local X coordinates of the joints computed in Section 3.2.

2) Since the structure is orthogonal, the global Y coordinates of each column of the reference frame is set equal to 0.

3) Search for intersecting Y frames.

4) Set the global X coordinate of each column stack of the Y frame equal to the global X coordinate of the intersecting column.

5) Set the global Y coordinate of each column of the Y frame equal to the corresponding local X joint coordinate. The steps 3 to 5 are repeated till all the Y frames that intersect the reference frame have been processed.

6) Find Y frames that do not intersect the reference frame.

7) Find a X frame that intersects with the Y frame under
consideration.

In this new X frame:

8) Find the first column with unknown global Y coordinate and the column closest to it with known global X/Y coordinates.

9) Compute the unknown global X coordinate, based on the length of the beam(s) joining the column of known coordinates and the column of unknown coordinates.

10) Set the global Y coordinate equal to the global Y coordinate of the column of known coordinates.

This process is repeated till all the column coordinates are determined. Fig. 3.5 illustrates the above procedure.

In that figure the coordinates of frames 1-2-3 and intersecting frames 1-6-11-16, 2-7-12-17 and 3-8-13 are determined first. Frames 4-9-14 and 5-10-15 do not intersect frame 1-2-3. The new X frame which does intersect them is 4-5-6-7-8. The first column with unknown Y coordinates is in the intersecting frame 4-9-14 and is column 4. The column closest to it with known coordinates is column 6. The coordinates of columns 4 and 5 are determined by knowing the coordinates of column 6 and the lengths of beams 4-5 and 5-6. After this coordinates along 4-9-14 and 5-10-15 can be determined.
This process produces X/Y coordinates of all column stacks. These coordinates can be used for a graphic display of any of the floors.
FIGURE 3.5 Column stack coordinates for an orthogonal structure.

Note: The arrows and corresponding numbers indicate the consecutive steps to find the column stack coordinates.
3.4 Computation of Three Dimensional Coordinates for Non-orthogonal Structures.

The procedure for determining X, Y and Z coordinates for non-orthogonal structures is in principle identical to the procedure for orthogonal structures.

In the non-orthogonal structures the beams and thus the vertical frames are no longer parallel to the global X/Y axis. The angle of deviation is input by the user and ranges anywhere from 0° to 90°. As in the other cases the structure is again composed of X/Y frames but these frames are no longer parallel to either axis. Therefore in computing the coordinates, use is made of the directional sines and cosines of the beams. In every other respect, the computational procedure is identical to that for orthogonal structures.

Secondary member (beam) coordinates are computed at the time of plotting, and this is explained in Section 3.8.2.
3.5 Generation of Building Grid Lines

Two types of grid line are defined at every floor of the building:

a) X-grid lines parallel to the X-axis.
b) Y-grid lines parallel to the Y-axis.

The X and Y grid line coordinates are generated at the reference level and apply to all levels.

Subroutine GRID5 determines the location of the grid lines. X-grid lines are located by looking up the Y-coordinates of the first column of each X-frame, similarly Y-grid lines are located by looking up the X-coordinate of the last column of each Y-frame.

These values are saved on a disk file for later plotting of frame elevations and floor plans.
3.6 Drawing Size, Border and Title Block

Five drawing sizes can be requested (A, B, C, D, E); the default is "D" corresponding to a 34"x 22" drawing.

3.6.1 The border is plotted by subroutine BORDER and consists of:
   a) Double solid frame.
   b) A cutting dashed line around the frame.

An example of the border lines is shown in Fig. 3.7.
FIGURE 3.7 Standard Border for a Drawing
3.6.2 Title Block

The title block is plotted by subroutine LOGOS, and the text written in it is controlled by AMECO "JOB TITLE" statements input by the user. Fig. 3.8 illustrates a typical title block, reduced about 30%.

Information under the headings 'CONSULTING ENGINEER', 'CLIENT', 'STRUCTURE' and 'PROJECT' text is plotted using the largest character size possible but no larger than .2". For long texts the character size is decreased.

The user may split the above texts in two lines by inserting a slash '/' at the desired location during data input phase.

'JOB' number and 'REVISION' number (input by the user), and 'DATE' (determined by AMECO) are also written in the title block using .125" character size.

Framing views title is written using .15" character size. Boxes are drawn for manually entering draftman's, checker's and engineer's names, and dates.
Space is also provided for professional engineer's seal.
<table>
<thead>
<tr>
<th>CONSULTING ENGINEER</th>
<th>CARLETON UNIVERSITY CIVIL ENGINEERING DEPT.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLIENT name</td>
<td>BRAMALEA LTD.</td>
</tr>
<tr>
<td>PROJECT name</td>
<td>MASTER THESIS</td>
</tr>
<tr>
<td>STRUCTURE</td>
<td>3 STOREY STEEL BUILDING</td>
</tr>
<tr>
<td>FRAMING view</td>
<td>FRAMING ELEVATIONS</td>
</tr>
</tbody>
</table>

| Engineer's seal     |                                             |

<table>
<thead>
<tr>
<th>SCALE</th>
<th>AS HEARD</th>
<th>DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>04/13/12</td>
<td></td>
</tr>
</tbody>
</table>

FIGURE 3.8 Title Block of a Drawing
3.7 Location of views on drawings

Two types of views can be drawn, framing elevations and framing plans.

Based on manual drafting practice, margins have been preset as shown in Fig. 3.9.

Subroutine FRAME computes the space required for each view by considering the actual size of the framing elevation or floor plans, the extension lines, dimension lines, grid lines with their circles and margins.

Subroutine FRAME attempts to place as many views (elevation/plans) on a single drawing as possible. An example is shown in Fig. 3.10. When a drawing is filled and more views are to be plotted, a new drawing is initiated.

If a single view does not fit on the drawing (based on user supplied scale and drawing size) a message is printed and processing is terminated.

Subroutine FRAME also sets the new origin for the next view to be plotted.
FIGURE 3.9 Standard Margins
FIGURE 3.10 Multiple Views on a Drawing
3.8 Plotting of Views

There are two types of views that can be requested:

Framing elevations

Floor plans

All members are plotted using .02 mm thick lines. The building grid and dimension lines are .01 mm thick. For a drawing made in Imperial units, all distances are written in feet, inches and fractions of inches (e.g. 3/8, 1/16). Member sizes are written in inches.

For drawings made in SI or Metric units, all distances and member sizes are written in mm.

3.8.1 Framing Elevations

As explained earlier in Section 3.7, subroutine FRAME locates the elevation to be plotted on the drawing and sets its origin. In the plotting of the elevations, the joint coordinates determined by the procedure described in Section 3.2 are used.

The elevations are plotted in the following sequence:

1) Columns

The columns are represented as follows:
Real columns are shown by a solid line.
Dummy columns when plotted are shown by a single dashed line.
Each column stack is drawn from the top level downwards.
Each column is drawn from its top joint downwards using its length supplied in the input.
Each column bottom end is represented by a short horizontal line. If the length of the column has been wrongly defined during the input phase the error will appear on the plot, otherwise the short line will be masked by the connecting beam.
2) Beam and Slab Representation

In general, each beam is drawn from its left joint towards the right, using its length supplied in the input.

If the frame is not parallel to the X or Y axis, or if some beams of the frame are not parallel to the X/Y axis, the length is modified by the directional cosine of the beam.

If the beam is too short, a gap will appear at the right end of the beam indicating the error.

Real beams and slabs are represented by a single solid line. Dummy beams, if they are to be drawn, are represented by a single dashed line.

Different conventions are used in drawing the beam depending on the type of connection. If the beam is continuous it is drawn to its full length. If the beam is pin connected a gap of 2 mm is left at the ends of the beam.

These conventions are shown in Fig. 3.12.
FRAMING ELEVATION LINE
SCALE 1 = 20

FIGURE 3.12 Conventions used in drawing the beams
3) **Vertical Bracing System and Walls**

Five types of bracing are available in AMECO: right bracing, left bracing, X-bracing, A-bracing and V-bracings. These are illustrated in Fig. 3.13(a).

The bracing end coordinates are computed as follows:

Subroutine PLOT5 finds the joint numbers to which the bracing ends are attached, by using AMECO functions MBRDN52, JTEL52 and JTRI52. Based on this information the joint coordinates are obtained and the bracing members are drawn.

The bracing members are represented by a solid line, without any gap at the ends. Walls are represented as X-bracing and drawn using dashed lines.

Fig. 3.13(b) illustrates the display of steel V-bracing and concrete shear walls, the latter represented by dashed X-braces.
FIGURE 3.13 Vertical Bracing System and Walls
4) Beam Connections, Marks and Sizes

(a) Beam connections

Rigid connections.

Steel- A rigid connection is represented by a solid triangle and is drawn by subroutine SYBMS5.

Concrete- no symbol is drawn.

Pinned connections.

In both steel and concrete a gap of 2 mm is left at the member ends.

(b) Beam, slab and wall marks.

Subroutine LABEL5 writes the member mark under the beam/slab/wall line using a character size of .125".

The beam, slab and wall marks are supplied by the user while generating the input data for AMECO.

(c) Beam sizes. Subroutine SIZE5 writes the corresponding beam size (profile name) above the beam, also using a character size of .125". It should be noted that the beam sizes on framing elevations are shown only for steel structures.
FIGURE 3.14 Column sizes, splice points and Column marks
5) Column sizes, splice points and column marks.

For writing column marks and sizes a character size of .094" is used.

(a) Column sizes and splice points.

For steel drawings, subroutine SIZE5 draws the splice points as a short horizontal line across the corresponding column at the locations requested by the user. An example is shown in Fig. 3.14.

The column sizes (profile names) are also written by subroutine SIZE5. The size information is placed to the left of the column near a splice point.

It should, however, be noted that column sizes are shown only for steel structures.

(b) Column marks.

Subroutine LABEL5 writes the column marks under the column stack.

A column mark consists of the letter C followed by the column number. The column marks are as supplied by the user while preparing the input data for AMECO.
6) **Dimension Lines and Building Grid Lines.**

(a) **Dimension Lines.**

Three different types of dimension lines are drawn on the elevations by subroutine DIM5.

A vertical dimensioning line is drawn to the left of the first column stack to indicate the interstorey heights.

In addition, two horizontal lines are shown, one to indicate the overall width of the elevation, and the other, the distances between the grid lines. An example appears in Fig. 3.16.

The dimensions are written using a character size of .125". If the space available for the dimensions is not adequate, the character size is decreased.

(b) **Building Grid Lines.**

The positioning of the grid lines has already been discussed in Section 3.5. Following the convention described above, subroutine LINE5 draws a grid line towards the top of each column stack, as an extension of the column line.

A grid line is represented by a dashed dotted line that ends in a circle of .5" diameter, with a label inside it, as shown in Fig. 3.16.
FRAMING ELEVATION LINE
SCALE 1 = 20

FIGURE 3.16 Framing Elevations
Letters A to Z are used to label grid lines parallel to the global Y-axis.

Numbers 1 to 99 are used to label grid lines parallel to the global X-axis.

7) Elevation Titles.

A title is written below each framing elevation, identifying it by the grid line label (placed in a circle), as shown in Fig. 3.16.

The scale is written below the elevation title.
3.8.2 Floor Framing Plans.

The floor plans are plotted in the following sequence:

1. Columns.
2. Beams, walls and slabs.
3. Column marks.
4. Main beam marks and end connections.
5. Secondary beam marks.
6. Main beam/wall/slab sizes.
7. Secondary beam sizes.
8. Floor plan titles.

For clarity, the description of the floor plans are presented in two parts; first for Steel structures and then for Concrete structures.

3.8.3 Floor Plan for Steel Structures.

1) Columns.

The column representation is worked out by subroutine SYMB5.

Columns are drawn as 'I' shapes, 50% larger than their actual dimensions. The web is drawn parallel to either X or Y-global axis or at an oblique angle, depending on the information supplied during input.

2) Main Beams, Secondary Beams and Walls.

A continuous beam is represented by a single solid line drawn to the full length of the beam.
A simply supported beam is represented by a single solid line drawn 2 mm shorter at the pinned ends. All secondary beams are assumed to be simply supported and are drawn accordingly. An example is shown in Fig. 3.17.

Each main or secondary beam can support a maximum of 8 secondary beams.

Subroutine SECBM computes the secondary beam coordinates as follows:

- The subroutine first identifies the secondary beam supported by X main beams using function JLBMS.
- It then uses function FXBMS to obtain from the AMECO database the length of the secondary beam and the distance of its left end from either the left end of the supporting beam or from the immediately preceding secondary beam should there be one.

The information generated by SECBM is used for plotting the secondary beams. A similar procedure is followed in plotting secondary beams supported by Y main beams.

Once all the secondary beams supported by the X/Y main beams have been determined a search is made for secondary beams supported by other secondary beams. The steps described in the foregoing paragraphs are then followed to determine the coordinates of such secondary beams.
FIGURE 3.17 Steel Building Plan
In order to avoid interferences between marks/sizes of supporting beams and the secondary beams framing into such beam, the secondary beams may be drawn shorter at one or both ends. An example is shown in Fig. 3.18.

Walls are represented by two solid lines.

3) Column Marks.

In order to avoid interferences between column marks and the marks/sizes of adjacent beams, a search is made for the best position of the column mark. This is done in subroutine POSIT, there are four possible locations for each column mark, as shown in Fig. 3.19.

Position 1 is the one normally selected if there is no interference. Position 2 to 4 are alternative positions, if none of them satisfy the requirement for clarity, the character size is decreased. Finally if no place can be found for the column mark an asterisk is plotted. Some examples of column marks can be seen in Fig. 3.20.

4) Main Beam Marks and End Connections.

Main beam and wall marks are written using a character size of .125". Under special circumstances, i.e. for short spans, the character size is decreased. If marks become unreadable they are replaced by an asterisk.
FIGURE 3.18 Floor Plan for Steel Building showing Secondary Beams
The main beam and wall marks are written out by subroutine LABEL5 and are located under the beam line parallel to it, as shown in Fig. 3.20.

A rigid connection is indicated by a solid triangle at the rigidly connected end. The symbol is drawn by subroutine SYMB5.

A pinned connection is shown by a gap of 2 mm between the connected members.

5) Secondary Beam Marks.

Secondary beam marks are written by subroutine SECBM, they are located under the beam and parallel to it.

To distinguish between a main and a secondary beam, the secondary beam mark is written in a character size of .094", or even smaller for short spans.

6) Main Beam/Wall sizes.

Beam sizes are written by subroutine SIZE5 just above the beam and parallel to it.

A character size of .125" is used, and it may be decreased for short spans or even replaced by an asterisk if the character size becomes too small.

Wall thicknesses are not written.
FIGURE 3.19 Positions for placing Column Marks

FIGURE 3.20 Floor Plan for a Steel Building showing Column marks and Beam Marks
7) **Secondary Beam sizes.**

Secondary beam sizes are written by subroutine SIZE5 and are located above the beam line and parallel to it.

Character sizes used are similar to those used for secondary beam marks.

8) **Dimension and Grid Lines.**

(a) **Dimension Lines.**

Subroutine DIM5 draws the dimension lines on floor plans. As shown in Fig. 3.21, three different types of dimension lines are drawn. These are:

1. Dimension line to indicate the overall floor width in the Y-direction. This line is drawn parallel to the global Y-axis and at 1.25" from the first Y-frame.

2. Dimension line to indicate the distances between X grid lines. This line is drawn parallel to the Y axis at .875" from the first Y-frame.

3. Dimension line to indicate the distances between the secondary beams. This line is drawn at .5" from the first Y-frame.

At this time the program draws dimension lines for only those secondary beams that frame into perimeter beams which are close to the dimension lines referred
to in 1 and 2 above.

A similar set of lines are drawn in the X direction. The dimensions are written using a character size of .125"; the character size is decreased if the available space for the dimension is too small.

(b) Grid Lines.

The position of the grid lines have been discussed in Section 3.5. Following the convention described there, subroutine LINE5 draws grid lines on the plans. Each grid line connects a series of columns and is represented by a dashed dotted line that ends in a circle of .5" diam. with a label inside it. A letter from A to Z identifies the Y grid lines and a number from 1 to 99 identifies the X grid line. (Fig. 3.21).

9) Floor Plan Titles.

A title is written below each floor plan using a character size of .2". The floor plan is identified by its level number. The scale is written below the floor plan title. (Fig. 3.21).
FRAMING PLAN LEVEL 2
SCALE 1 = 20

FIGURE 3.21 Dimension Lines, Grid lines and Floor Plan Titles
2.8.4 Floor Plans for Concrete Structures.

The plotting of the floor plan is executed in a sequence similar to that used in plotting the floor plan for steel structures and in a very similar manner.

1) Column Representation.

The columns on the floor plan of a concrete structure are drawn by subroutine SYMBS. Columns are represented by their real shape (square, rectangle or round). The column faces can be oriented parallel to either X or Y axis or at an oblique angle.

2) Beams, Walls and Slabs.

In general, beams are represented by their real width and length and are drawn using double dashed lines. Secondary beam coordinates are computed in a manner similar to that explained in Section 3.8.2(2). On the drawings the secondary beams are represented in the same manner as the main beams.

If interference occurs between marks/sizes of supporting beams and secondary beams, the latter are drawn shorter than their real length. Example can be seen in Fig. 3.22.

Walls are represented by double solid lines
Slabs are represented by solid lines with half arrow head at each end, specific representation of the slabs is
however shown on the drawings only if the marks of the beams/slabs/walls have been requested by the user during the input phase. Examples are shown in Fig. 3.23.

3) **Column Marks.**

Subroutine POSIT locates the column marks as described in Section 3.8.2(3).

4) **Main Beam Marks and End Connections.**

The main beam marks are written in the same manner as for steel beams. Details are described in Section 3.8.2(4). A rigid beam-column connection is represented by joining the column and the beam together. A gap of 2 mm is left at a pinned end of the beam.

5) **Secondary Beam Marks.**

The details are similar to those described in Section 3.8.2(5) for steel structures.

6) **Main Beam/Wall/Slab Sizes.**

Main beam sizes are written by subroutine SIZE5; they are located above the beam parallel to it. A beam size shows the width and depth of the beam - in inches for drawings made in Imperial units, and in mm for drawings made in SI or Metric units.
FIGURE 3.23 Floor Plan of a Concrete Structure showing the Representation of Slabs
A character size of 0.125" is used.

Wall and slab sizes are omitted.

7) Secondary Beam Sizes.

The secondary beam sizes are written in the same manner as main beam sizes, except that a character size of 0.094" is used.

8) Dimensions and Building Grid Lines.

These are drawn in a manner similar to that for steel structures in Section 3.9.2(8). Examples can be seen in Fig. 3.22.

9) Floor Plan Titles.

The details are similar to those described for a steel structure in Section 3.8.2(9).
CHAPTER 4

EXECUTION OF THE GRAPHIC PROGRAM

The graphic module, integrated in the AMECO - 17 system, is operational on Control Data Corp. CYBER Series computers, under the NOS operating system.

AMECO-17 is executed in batch mode. Therefore, as a first step, an input file is created (or an existing file retrieved). This file consists of AMECO commands describing the structure, the loads acting on it, the design criteria, etc. - all in accordance with the AMECO User’s Manuals(8),(9). The file is then supplemented with the desired PLOT commands. The sentence structure and the application of the PLOT commands are described in the following sections of this chapter.

Together with appropriate NOS Control Statements, that load AMECO, the input file is then submitted to the CYBER machine for processing.

Besides the files that the design modules of AMECO may create, the PLOT commands will generate permanent "Plot Files". These files can be displayed on a variety of graphic CRT terminals or plotted directly on a hard copy plotter. In addition, "Plot Files" can be transmitted to an Inter-
active Graphic System for re-formatting and further graphics processing.

This chapter presents the rules for the PLOT commands and describes the various optional procedures available for obtaining graphic displays.
4.1 Command Description

General.

The commands are formed according to the AMECO language conventions. These conventions include the following definitions.

Statements are "All instructions and data entered into the computer for AMECO processing in the form of sentences and in free format."

Command is "a statement starting with a verb".

Labelled data is "data entered in any order, in a command with appropriate labels".

Plot commands direct the program to produce a plot file of a specific view. Labelled data defines the specific characteristic required in the view.

The Plot commands have the same structure as the other AMECO commands, therefore they follow the same rules. These are:

- Default values have been preset for most of the labelled data; they are indicated by arrows in the facsimile of the command.
- All words can be abbreviated to the first four characters.
- When shorter abbreviations are allowed, the required
characters are underlined in the facsimile (Fig. 4.1).
- Words with horizontal bars above can be omitted without changing the meaning of the statement.
- Prepositions: 'AT', 'FROM', 'AND', 'WITH', 'ON', 'FOR', are ignored by the lexical analyser, they can be used to construct more legible statements.
- A maximum of 80 characters can be entered per line.
- A hyphen may be used at the end of a line to continue a long Plot Command on the next line. A blank must precede the hyphen.

The command structure is shown by the command facsimile in Fig. 4.1. The command begins with a verb and is followed by one or more modifiers and then by a series of labelled data. More than one command modifier can be included with one command. Labelled data should follow the command modifier. A new command modifier thus terminates the list of labelled data. All components of the command are separated from each other by either one or more spaces or a comma. Only one of the items enclosed in a pair of braces, on the facsimile of the command should be selected at one time.

The Plot Commands are analysed by subroutine CIPLLOT. Subroutine CIPLLOT reads the plot commands, checks for illegal entries, changes default values to the entered
values and stores flags to execute the plotting according to the user requirements. If an illegal entry is found a message is printed indicating the error and processing is terminated in the input phase.
Figure 4.1 PLOT COMMAND facsimile.

NOTE: Labels PL1 to PL5 and PLD1 to PLD8 are not part of the command, they are included here only for the purpose of references.
4.1.1 PLOT COMMANDS.

\[
\begin{array}{c}
\text{ALL} \\
\text{FRAME} \\
\text{FLOOR} \ (l_1, l_2, \ldots, l_5 \text{ to } l_8) \\
\text{XFRAME} \ (fx_1, fx_2, \ldots, fx_5 \text{ to } fx_8) \\
\text{YFRAME} \ (fy_1, fy_2, \ldots, fy_5 \text{ to } fy_8)
\end{array}
\]

(PL1) Enter "ALL" to plot all vertical frames and all floor plans of the structure.
Labelled data - enter data as described in Section 4.1.2
E.g. PLOT ALL
E.g. PLOT ALL, MARKS NO
This command will produce drawings without member marks.

(PL2) Enter "FRAME" to plot all the frames in X and Y directions.
Labelled data - enter data as described in Section 4.1.2
E.g. PLOT FRAME
E.g. PLOT FRAME, SIZE OF BEAM YES
This command will produce drawings of X and Y frames with all beam sizes shown on the drawing.

(PL3) Enter "FLOOR" to plot all the floors of the structure.
Enter level numbers \(l_1, l_2, \ldots, l_3\) to \(l_8\) to plot selected floor plans.
Labelled data - enter data as described in Section 4.1.2
e.g. PLOT FLOORS
This command will produce a plot file containing all
the floor plans of the structure.
e.g. PLOT FLOOR 1
e.g. PLOT FLOOR 2 TO 4 6 9
The last command will produce a plot file for plotting
floor plans of levels 2, 3, 4, 6 and 9.
e.g. PLOT FLOOR 1 MARKS FOR MAIN BEAM NO
This command produces a plot of level 1 without main
beam marks.

(PL4)
Enter "XFRAME" to plot all the X-direction frames of
the structure.
Enter frame ID's fx₁, fx₂, ... fx₅ to fx₈ to plot
selected frames.
Labelled data - enter data as described in Section 4.1.2.
CAUTION: Enter XFRAME as one word, otherwise a syntax
error will occur.
e.g. PLOT XFRAME
This command produces a plot file containing all the X
direction frames of the structure.
e.g. PLOT XFRAMES 1 6
This command produces a plot file for drawing X-direction
frames 1 and 6.
e.g. PLOT XFRAME 1 6, DUMMY NO, SCALE 1/10
This statement produces a plot file for drawing X-
direction frames 1 and 6 to scale of 1:10 and with the
dummy members omitted.

(PL5) Enter "YFRAME" to plot all the Y-direction frames of
the structure.
Enter frame ID's fy1, fy2, ..., fy8 to fy8 to plot
selected Y-frames
Labelled data - enter data as described in Section 4.1.2.
CAUTION: Enter YFRAME as one word, otherwise a syntax
error will occur.
Same as command (PL4), but applied to Y-direction frames.
In general, each command (PL1 to PL5) will initiate a new
drawing.
4.1.2 Labelled data in Plot Commands

Labelled data are words followed by modifier words or numeric data and can be entered in a random order in the Plot Command. The data items need to be entered only where they differ from the default values.

(PLD1) Enter "NO" when dimensions and grid lines are to be omitted.

   e.g. PLOT FLOOR DIM NO, XFRAME 1 TO 4

   This statement produces a plot file for all the 'floor plans of the structure without dimensions and grid lines and also will create another drawing(s) containing X-direction frames 1, 2, 3, 4 with dimensions and grid lines shown.

(PLD2) Enter "NO" when dummy members are to be omitted.

   e.g. PLOT XFRAME 3 DUMMY NO

(PLD3) Enter "NO" when secondary beams are not to be plotted in any or all the floor plans.

   e.g. PLOT FLOOR SECO BEAMS NO

   This statement produces a plot file for drawing all the floor plans without secondary beams.

(PLD4) Enter scale "1/n" to change the default scale.

   In the Imperial units scale designation 1/n stands for
1 inch equal to n feet. The default value is 1/20 implying 1 inch equal to 20 feet.

For SI or Metric units the scale designation 1/n stands for 1 mm equal to n mm, the default value is 1/200.

The slash "/" in the scale designation is optional and can be replaced by a blank, so then 1 n is the same as 1/n.

For Imperial units:

  e.g. PLOT FLOOR 5, SCALE 1/15

For SI or Metric units:

  e.g. PLOT FLOOR 5, SCALE 1 100

The last command produces a plot file for drawing the floor plan at level 5 at a scale which implies 1 mm on the drawing represents 100 mm of real dimension.

(PLDS) Enter "NO" to omit all member marks (columns, main and secondary beams)

  e.g. PLOT YFRAME 3, MARKS NO

This command will produce a plot file for drawing the Y-direction frame 3 without any member marks.

  e.g. PLOT YFRAME 3, MARKS FOR BEAM NO

This command will produce a plot file for drawing Y frame 3 with column marks but with no beam marks.
e.g. PLOT FLOOR 2, MARKS OF BEAM NO
This command will produce a plot file for drawing the floor plan at level 2 without main and secondary beam marks.

(PLD6) Enter "YES" to write the sizes of either all the members or of only the selected type of members.
e.g. PLOT FLOOR 12, SIZE BEAM YES
This command will produce a plot file for floor level 12 on which the beam sizes (main and secondary) will be shown.

(PLD7) Enter drawing size to modify the default size "D" that corresponds to a 34"x 22" drawing.
e.g. PLOT FLOOR 1, DRAW SIZE A
This will produce a plan of floor level 1 on a 11"x 8 1/2" drawing size:

Drawing sizes available are:

<table>
<thead>
<tr>
<th></th>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>11</td>
<td>8 1/2</td>
</tr>
<tr>
<td>B</td>
<td>17</td>
<td>11</td>
</tr>
<tr>
<td>C</td>
<td>22</td>
<td>17</td>
</tr>
<tr>
<td>D</td>
<td>34</td>
<td>22</td>
</tr>
<tr>
<td>E</td>
<td>44</td>
<td>34</td>
</tr>
</tbody>
</table>

All the sizes are given in inches.
(PLO8) Enter "NO" to omit rigid connection representation for steel drawings.
This instruction applies to steel structures only.
e.g. PLOT FLOOR 6, CONN BEAM NO
This will produce a plot file for the floor plan of level 6 in which the rigid connection symbols(solid triangles) will be omitted from the rigidly connected beam-column joints.

4.2 Purposes of Plotting

Graphic displays can serve three different objectives:

a) Verification of input data.
b) Schematic displays of design results.
c) Production of final drawings.

a) Verification of input data

AMECO-17 is a system that can handle a wide range of structures. For a successful design the first requirement is that the geometry of the model entered into the computer represent the real structure in every detail.

To achieve this objective it is necessary to carry out a very careful check of the geometry (member dimensions, connectivities, etc.)
depending on whether the grid is an X or Y grid.
- Elevation/section title identified by grid line letter or number.
- Scale at which the elevation/section was drawn.

b) Concrete structures

- Framing elevation/section drawn to specified scale and units of measurement.
- Column marks.
- Beam marks.
- Level numbers.
- Dimension showing overall width of the elevation/section.
- Dimension indicating interstorey heights.
- Building grid lines identified by letters or numbers depending on whether the grid is an X or a Y grid.
- Elevation/section title identified by grid line letter or number.
- Scale at which the elevation/section was drawn.

All framing plan and elevation drawings have a border and a title block.

Appendix A illustrates five drawings produced by the graphic module.
execute phase A only and come to a stop thereafter as illustrated in Fig. 4.2.

The following example illustrates the use of these two commands.

execute phase A only and come to a stop thereafter as illustrated in Fig. 4.2.

The following example illustrates the use of these two commands.

execute phase A only and come to a stop thereafter as illustrated in Fig. 4.2.

The following example illustrates the use of these two commands.

execute phase A only and come to a stop thereafter as illustrated in Fig. 4.2.

The following example illustrates the use of these two commands.

execute phase A only and come to a stop thereafter as illustrated in Fig. 4.2.

The following example illustrates the use of these two commands.
Schematic Display of Design Results

Generally, for the design of large structures more than one design cycle is required. Referring to Fig. 4-2, the cycling process in Phase C will be repeated as many times as has been requested by the user or till the changes in member sizes is less than the allowed tolerance.

Phase C provides the user the opportunity to obtain a graphic display of the member sizes for each design cycle.

Two sets of commands are required to obtain a graphic display of the design results:

1) One or more Plot Command to specify which are to be displayed. Framing elevations provide the user the column and beam sizes for all or specified frames, while the floor plans provide him the main and secondary beam sizes for all or specified levels.

2) The execution command to carry out the plotting cycle by cycle is "EXECUTE ALL;PLOT EACH CYCLE" and like other execution commands it must be entered last.

The command "EXECUTE ALL, PLOT EACH CYCLE", instructs the program to execute Phases "A" to "D", producing a plot file for each design cycle in phase "C".
FIGURE 4.2 AMECO integrated analysis/design process with graphics
Sometimes, the user may desire to review the results of one cycle before proceeding to the next cycle. A set of commands permit the user to do the following:

1. Execute one design cycle.
2. Save the results.
3. Produce the plot file.

The last of these commands has been added as a result of the work described in this thesis. To be able to produce a plot file after a design cycle, the user must input:

a) Plot commands that will select the views to be plotted, displaying the member sizes most recently selected.

b) The Execution Command "EXECUTE" which will carry out a design and produce a plot file after completion of a design cycle.

If plotting was not requested during a run but the user desires to have a graphic display of the member sizes, he can retrieve the results, enter the appropriate Plot command and insert "EXECUTE PLOT" command as the last instruction.
c) Production of Final Drawings

At this stage, the objective of plotting is to produce a set of engineering drawings which include framing plans, elevations and sections. Appendix A contains five example drawings.

Each engineering drawing will contain the following information:

1) Framing Plans

a) Steel structure floor plans...

- Floor plans drawn to specified scale and units of measurement.
- Column marks.
- Main and secondary beam sizes.
- Dimensions showing the overall widths of the floor plan, distances between grid-lines and distances between secondary and main/secondary beams.
- Building grid lines identified by letters in the case of Y-direction grid lines, and by numbers in the case of X direction grid lines.
- Floor plan title and floor level number.
- Scale at which the floor plan was drawn.
b) **Concrete structure floor plans.**

The concrete floor plans show the same information as the steel plans described above. The member representation is obviously different. Thus:
- beams are drawn with two dashed lines.
- columns are drawn as rectangles, squares or circles.
- flat plate/slat members are not shown.

Also the edge of the floor slab is indicated by drawing a line at the exterior faces of beams/columns on the perimeter of the floor.

(2) **Framing Elevations and Sections**

a) **Steel structures.**
   - Framing elevations and or sections drawn to the specified scale and units of measurement.
   - Column marks.
   - Column splice points.
   - Column sizes.
   - Beam sizes.
   - Level numbers
   - Dimension showing the overall width of the elevation.
   - Dimensions indicating the interstorey heights.
   - Building grid lines identified by letters or numbers.
depending on whether the grid is an X or Y grid.
- Elevation/section title identified by grid line letter or number.
- Scale at which the elevation/section was drawn.

b) Concrete structures

- Framing elevation/section drawn to specified scale and units of measurement.
- Column marks.
- Beam marks.
- Level numbers.
- Dimension showing overall width of the elevation/section.
- Dimension indicating interstorey heights.
- Building grid lines identified by letters or numbers depending on whether the grid is an X or a Y grid.
- Elevation/section title identified by grid line letter or number.
- Scale at which the elevation/section was drawn.

All framing plan and elevation drawings have a border and a title block.

Appendix A illustrates five drawings produced by the graphic module.
CHAPTER 5

DISCUSSIONS

5.1 General

The principal objective of this project is to implement a graphic module for AMECO-17 system, that would produce automatically structural framing drawings.

This Chapter deals with some of the programming problems encountered and identifies the deficiencies and limitations of the graphic module at its present state of development.

This is followed by a description and a qualitative evaluation of five sample drawings presented in Appendix A.

5.2 Interface with AMECO-17 Files

The graphic module has been programmed to produce drawings totally within the system - all input data to the graphic must come from the files of the "parent" AMECO-17 System.

The structure of AMECO files was set up years ago to accommodate its joint relaxation process, for structural analysis, with not much thought given to future plotting needs. One striking example of this is the fact that
AMECO has no joint coordinate tables - it never computes the joint coordinates.

Furthermore, its member connectivity tables are quite different from those in conventional structural analysis programs. For example, an AMECO file for a typical beam states that at given distances along the span, certain tributary beams frame into it.

As an example of the problem in plotting that arise because of the conventions followed by AMECO in defining connectivity, consider the frames shown in Fig. 5.1. To draw B301 of Fig. 5.1(a), a search has to be made through all the beam files to find which beam "supports" the left end of B301. Once, the supporting beam B101 has been found then knowing its end coordinates and the direction cosine and the distance to B301 from the end of B101, the left end coordinate of B301 can be established.

In a similar manner, the right end coordinates of B301 must be found. It is only after these coordinates have been established that the beam can be drawn.

For conventional programs, the plotting procedure is far more simpler. Referring to Fig. 5.1(b), the member incidence table should state that member 7 goes from joint 4 to 9; joint coordinate table would provide the location of joints 4 and 9. Based on this information member 7 can be easily drawn from joints 4 to 9, without any searches or computations.
FIGURE 5.1 Member Connectivity Convention

The efficiency of AMECO is largely attributed to the fact that structural analysis is performed totally in core (e.g. 2,700 joints solved in 130,000 word memory). To achieve this, in many AMECO files, more than one datum is stored in each word. For example, a 3 bit field is allocated for storing member cross-section index, e.g. 1 for T-beam, 2 for slab, etc. Thus indices for 20 members can be stored in one 60 bit word. Because of the presence of these specialized storage algorithms, considerable effort had to be spent to become familiar with the structure of data files. Further, to access the data contained in the files, extensive use had to be made of unpacking routines.
5.3 Joint Coordinates

The computation of the joint coordinates forms the starting point in the graphic module.

The computation of the two-dimensional and three-dimensional joint coordinates has been described in detail in Sections 3.2 and 3.3. These computations were based on three assumptions:

a) Columns are vertical elements joined together by horizontal members.

b) All columns are defined on at least one level.

c) Any three-dimensional structure is composed of a series of intersecting frames.

The example shown in Fig. 5.2 illustrates step by step the method used in computing the column stack coordinates. It is assumed that local frame coordinates have been computed earlier.

The example shown in Fig. 5.3 explains in detail a case where all the two-dimensional coordinates have been computed, but some of the three-dimensional joint coordinates cannot be computed and therefore the floor plan cannot be plotted.
FIGURE 5.2 Calculation of Column Stack Coordinates.

1) Set frame 1-2-3 as reference frame.

2) Set X-coordinates of column stacks 1, 2 and 3 equal to the corresponding X joint coordinates computed in the local X-Z axes of the reference frame. Set Y-coordinates of column stacks 1, 2 and 3 equal to 0. This is valid because the structure is orthogonal.

3) Compute coordinates of intersecting frame 1-5-9.

For column stacks 5 and 9:
Set the Y-coordinates as equal to the local X-coordinates of joints 5 and 9 respectively.
Set the X-coordinates as equal to the X-coordinate of column stack 1.

4) By a similar procedure obtain the coordinates of column stacks 6, 10, 7 and 11.
5) Find frame 4-8 that does not intersect frame 1-2-3.
6) Find frame 4-5-6-7 that does intersect frame 4-8.
7) Repeat step 3 to compute the coordinates of column stacks 4 and 8.

FIGURE 5.3 A Case where Stack Coordinates cannot be Computed
In this case, the floor plan is formed by X-frames 1-2-3 and 4-5-6-7, 8-9-10-11 and Y-frames 4-8, 5-9, 6-10 and 7-11.
According to AMECO convention there is no Y-frame connecting column 1 and points A and B. Similarly there is no frame connecting column 2 and points C and D; also no frame connects column 3 and points E and F.

Even though, all the AMECO frames have been input and the two-dimensional coordinates computed, the structure is indeterminate from the point of view of plotting, because
there are no connections between frame 1-2-3 and either of frames 4-5-6-7 and 8-9-10-11.
The elevations can be plotted, but not the floor plan.
This situation can be remedied by entering dummy columns at points A to F.
5.4 Location of Views on Drawings

A drawing may display one or more views - plans or elevations. This is a function of the size of the drawing, scale and size of the individual views.

Refering to Fig. 5.4, the following steps are involved in positioning the views on the drawing:

1) Set origin at 'A' and draw drawing border and title block.
2) Determine the size of view I.
3) Check if the view fits on the drawing.
4) Position the view on the drawing and reset origin at 'B'.
5) Draw view I.
6) Determine the size of view II, considering that X-direction grid lines must match up in views I and II.
7) Repeat steps 3 to 5.
8) Determine size of view III.
9) Check if view III fits next to view II, otherwise attempt to place it below view I.
10) Reset origin at 'D' and draw view III.
11) Repeat procedure until the drawing is full, then start a new drawing.
The determination of the size of the view is itself a rather complex operation. In this determination one has to consider:
a) the actual size of the framing; b) the length of the grid lines with their circles; c) the length of the extension lines; d) the requirement that $X$-direction grid lines must line up from one adjacent view to the next; the width of margins; e) the title of the view.
FIGURE 5.4 Location of Views on Drawings.
5.5 Plotting Secondary Members

The end coordinates of the secondary beams are computed at the
time a plot file is being created.

The following steps are involved in plotting:

1) Determine the main/secondary beam that supports the left
   end of the secondary beam.
2) Determine the length of the supported secondary beam.
3) If supporting beam marks have been requested:
   a) Check if interferences will occur between the secondary
      beam and the lettering for the supporting beam mark.
   b) If interferences occurs, it is necessary to shorten the
      length of the supported beam at its left end.
4) Determine the main/secondary beam that supports the right
   end of the secondary beam.
5) If supporting beam sizes have been requested:
   a) Check if interferences will occur between the secondary
      beam and the lettering for the supporting beam size.
   b) If interferences occurs, it is necessary to shorten the
      length of the secondary beam at its right end.

Example of secondary beams are presented in Fig. 5.7
An AMECO floor framing plan may have a number of beams with
the same mark, as shown in Fig. 5.8. Consequently it becomes
a rather involved logistics problem to plot the beams with
identical marks and to make sure that indeed all of them have been plotted.

FIGURE 5.7 Example of Nested Secondary Beams

FIGURE 5.8 Example of Identical Beam lines
5.6 Column Marks

One of the most important requirement in a drawing is clarity. Thus interference of lettering and lines should be avoided as far as possible.

Sometimes the space available to plot the column mark and the beam mark/size of adjacent member is not large enough and interferences may occur.

A very involved procedure was developed to avoid this interferences. The procedure works as follows:

Any column mark in a floor plan can have 4 different positions, as shown in Fig. 5.9.

![Diagram of Column Marks]

FIGURE 5.9 Position of Column Marks

The principles involved in the calculation of the coordinates of the lettering block containing the column mark are the same irrespective of the position selected. This description of the method is for position 1, and is based on the assumption that the user has instructed the program to plot the main beam marks/sizes as well as the column marks.
Near position 1, the column has attached to it beams A and B. The coordinates of the column mark are calculated taking into account:

a) Angle of deviation of the column from the X-axis.
b) Angle of deviation of beam A from the X-axis.

Once the position of the column mark has been defined, it is checked against the position of the mark of beams A and B. If interferences are encountered, the mark is moved to position 2, 3, 4 and a similar procedure is followed for determining the coordinates of the mark in its new position.
5.7 Building Grid Lines

There are two types of grid lines:

a) X grid lines, they are parallel to the X-direction and are drawn near the first column stack of each X frame.

b) Y grid lines, they are parallel to the Y-direction and are drawn near the last column stack of each Y frame.

The position of each X/Y grid line is set by the position of the first/last column stack of the corresponding frame. According to this definition, any column out of line with the first/last column stack is drawn eccentric to the grid line. Fig. 5.10 illustrates this point - the columns along grid lines 1, 2 and 3 are not in line.

For non-orthogonal structures, not all the grid lines will be properly defined. Therefore it is advisable, in the plot command to suppress the grid line generation.

Fig. 5.11 illustrates a case where the grid lines cannot be drawn properly.
FRAMING PLAN LEVEL 3
SCALE 1 - 10

FIGURE 5.10 Convention used in Drawing Grid Lines
FIGURE 5.11 A non-orthogonal structure.

The Y grid lines are going to be placed at column 13, 14, 15 and 16, since column stack 13 is the last column stack of frame 1-5-9-13. There will be no grid line at column 9 because columns 9 and 13 both belong to the same Y frame.

For "severely" non-orthogonal structures, such as the one illustrated in Fig. 5.12, the currently programmed grid generation procedure most likely will produce unexpected results.
FIGURE 5.12 A Complex Non-orthogonal Structure
5.8 Evaluation of Sample Drawings

Appendix 'A' contains five sample drawings, all of size 'D' photo-reduced to 17" x 11" sheets. These were plotted using ink pens on Calcomp model 1039 drum plotter.

Drawing 1 and 2 are typical of 12 structural steel drawings for a 14 storey steel structure built in Culver City, California. The drawings were created by retrieving the actual design result files created in 1982.

Drawing 1 is representative of a steel framing of average complexity encountered in practice. This drawing shows:

a) all structural members and their sizes.

b) dimensions locating 95% of all the members.

c) building grid lines.

d) beam/column moment connections, identified by solid triangles.

e) hinged connection represented by 2 mm gap at member end.

f) column cross-section, drawn to 50% larger scale.

These drawings do not show:

a) dimensional location of secondary beam in mid-part of the plan for level 4.

b) location of floor openings.
c) floor projection beyond column lines.

d) number of studs for composite beams.

Overall, the drawing is about 95% complete.

Drawing 2 is representative of steel building elevation drawings, showing moment resistant frames.

The drawing shows:

a) all beam and column sizes.

b) column splice points identified by horizontal bars.

c) simple (web) connections identified by 2 mm gap at beam ends.

d) moment connections identified by solid triangles.

e) dimensions between grid lines and floor levels.

f) beam and column marks for cross-referencing to design documents.

It could be advisable to have the column web orientation shown at the bottom of the column stacks, although this is already shown on floor plans.

The drawing is 100% complete.

Drawings 3 and 4 show a 7 storey concrete building composed of one-way slabs, flat slabs, beams, columns and shear walls.
Drawing 3 is representative of a typical, regular layout of floor framing.

The drawing shows:

a) all beams (drawn to scale) with their sizes indicated.
b) columns (drawn to scale), but with sizes not indicated.
c) walls (drawn to scale), but with thicknesses not indicated.
d) slab extended on the perimeter to the face of the columns, but with projection dimensions not indicated.
e) building grid lines and distances between grid lines.

If desired the beam/slab marks could have been displayed on the drawing for cross-referencing with reinforcing schedules prepared by AMECO.

The drawing lacks the following information:

a) floor openings.
b) wall thicknesses.
c) slab thicknesses.
d) floor projection dimensions.

The drawing is about 85% complete.
Drawing 4 shows a longitudinal and a transverse framing elevation of the building.

In engineering practice framing elevation drawings for concrete buildings are normally not made. Floor plans, reinforcing schedules with member sizes and detail drawings are normally sufficient to fully describe the structure. However, such a drawing could be used to supplement the design calculations.

Drawing 5 illustrates a typical flat plate floor plan having the interior columns offset from the building grid lines. The drawing shows:

a) all beams (drawn to scale) with their sizes indicated.
b) columns (drawn to scale), but with sizes not indicated.
c) slab extended on the perimeter to the edge of the columns, but with projection dimensions not indicated.
d) walls (drawn to scale), but thicknesses not indicated.
e) building grid lines and distances between grid lines.

The drawing lacks the following information:

a) floor openings.
b) wall sizes.
c) offsets of interior columns from grid lines.
d) reinforcing steel band lines (column/middle strips).
SUMMARY AND CONCLUSIONS

A graphic module has been developed and integrated into an existing system of structural engineering programs, known by the acronym AMECO-17. As programmed, the graphic module can produce structural steel and concrete framing plans and elevations to serve three different objectives:

1. Verification of input data.

   The graphic displays are powerful tools for debugging the input data, specially for large structures. These schematic drawings can be consider to be fully finished for both, steel and concrete structures.

2. Schematic displays of design results.

   The user can obtain a graphic display of the member sizes for each design cycle.

   For steel structures the drawings can be considered as almost complete. In the case of concrete structures the completeness is less since no member sizes are shown in the framing elevations.

3. Production of final drawings.

   The drawings are completely generated within the system, requiring no operator input. The degree of completion varies and it is higher for steel structures than for concrete structures.
The graphic module provides with the opportunity for direct transfer of data from the program to the drawing sheet. Even though the degree of completion varies, they provide with an important amount of information that avoid the need to do a lot of tedious work. Furthermore, the drawings can be completed by means of an interactive graphic system.

This graphic module is by no means a finished product and more effort could be invested to implement additional capabilities, such as: cantilever projections, user selected column marks, floor openings, user specified grid lines, dimensioning of interior secondary beams. For concrete structures, column capitals and drop panels should be shown and reinforcing schedules plotted on the drawings. Also standard concrete details could be shown.

The development of this graphic module was made possible by the existance of the AMECO-17 system. Based on research on literature, AMECO-17 is one of the few analysis/design systems that can produce, for an entire building, the complete design information needed for automated structural drafting.
REFERENCES


APPENDIX A  Five Sample Drawings
FRAMING PLAN LEVEL 14
SCALE 1:20
FRAMING ELEVATION LINE 1
APPENDIX B

Program and Subroutines Description

This section presents a brief description and flowcharts of the main program and the 30 subroutines of the graphic module. The program and the subroutines together have approximately 6300 Fortran statements. The complete Fortran listing is presented in Appendix C.

The coding is done in AHSI-77 Fortran. Advantage is taken of the CDC Fortran 5 extensions, e.g. direct access I/O, Decode/Encode routines for alphanumeric conversions, etc.

No description is presented for the bit manipulating routines that already existed in AMECO prior to the commencement of this project.
(1) **Subroutine CIPLDT**

CIPLDT decodes all the Plot commands and saves the requested information on a disk file. It is called by AMECO overlay OVL11 in Phase "A", whenever the word PLOT is encountered in the input stream of commands.

(2) **Subroutine PLOT5**

PLOT5 computes the two-dimensional joint coordinates, using subroutines: STACDN5, STACUP5, STACLE5, STACRI5.

Also, it finds the joint number at which the bracing ends are attached using AMECO functions: MBRON52, JTEL52 and JTRI52. PLOT5 is called by AMECO overlay OVL52 in Phase "A" or "C", (Fig. 2.2).

(3) **Subroutine STACDN5**

STACDN5 finds the coordinates of all joints below the reference level. It uses Function MCOLDN52.

(4) **Function MCOLDN52**

This AMECO function finds the joint number at the bottom of a column.

(5) **Subroutine STACUP5**

STACUP5 finds the coordinates in local X-Z axis of all joints above the reference level. It uses Function MCOLU52.
(6) Function MCOLU52
This AMECO function finds the joint number at the top of
a column.

(7) Subroutine STACLE5
STACLE5 computes the two-dimensional joint coordinates in
local X-Z axis of the joint at the reference stack and at
the reference level.

(8) Subroutine STACRI5
Similar to subroutine STACLE5 except that it looks up the
joint at the right of the reference stack.

(9) Function MBRDN52
MBRDN52 finds the joint number at the bottom end of the brace.

(10) Function JTLE52
Function JTLE52 finds the joint number at the left end of the
beam.

(11) Function JTRI52
Function JTRI52 finds the joint number at the right end of the
beam.
(12) **Program OVL53**

This is the main program of the graphic module its ultimate function is to compute the three-dimensional joint coordinates to produce the plot files for the requested views/drawings.

OVL53 is an overlay and is loaded in memory immediately after OVL52 is executed, either in Phase "A" or Phase "C".

Fig. B.1 shows the flowchart for the plotting of framing elevations.

Fig. B.2 shows the flowchart for the plotting of floor plans.

OVL53 calls subroutines 13 through 31. The latter are described in the remaining part of this appendix.

(13) **Subroutine ARROW**

ARROW draws the arrows for the dimensioning lines.

(14) **Subroutine BORDER**

BORDER plots the double border for a drawing.

(15) **Subroutine DIM5**

DIM5 does the dimensioning of framing elevations and floor plans. Fig. B.3 shows the flowchart for this subroutine.
FIGURE B.1 Flow Chart of the Program Section for Plotting the Framing Elevations.
STEEL DWG?

CONV. SYMB.?

SYMB5

BEAM MARK?

LABEL5

STEEL DWG?

CONV. SYMB.?
FIGURE B.1 (cont.)
FIGURE B.2 Flow Chart of the Program Section for Plotting the Floor Plans.
CONNECTION SYMBOL?

SYMB5

COLS. MARK?

YES

POSIT

NO

LABEL5

NO

MAIN BEAM SIZE?

YES

SIZES5

NO

MAIN BEAM MARK?

NO

FIGURE B.2 (cont.)
FIGURE B.2 (cont.)

Diagram:

24

DIMENS.

YES

DIM5

NO

MORE FLOORS TO PLOT?

YES

20

NO

MORE VIEWS TO PLOT?

YES

19

NO

STOP
FIGURE B.3 Flow Chart for Subroutine DIM5
(16) Subroutine ENCO

The objective of this subroutine is to encode the beam/column profile names for steel structures and the 'packed' sizes for concrete structures.

(17) Subroutine FRAME

Subroutine FRAME locates each view into the drawing, checks if the drawing size is large enough, starts a new drawing when necessary or sets a flag to stop the process.

(18) Subroutine GRIDL

GRIDL sets the origin of an internal grid line, such as A in Fig. B.4, so as to avoid intersecting the beam on which the grid line terminates (beam B703 in Fig. B.4).

![Diagram of internal grid lines]

FIGURE B.4 Example of Internal Grid Lines
(19) Subroutine GRID5
GRID5 sets the origin of the X/Y building grid lines.

(20) Subroutine IDIM5
IDIM5 encodes the dimensions in feet, inches and fractions of inches if the drawing is done in Imperial units of measurements. The subroutine also plots the dimensions in Imperial or in SI or Metric.

(21) Function FXBMS5
This function computes the distances between secondary beams.

(22) Function JLBMS5
JLBMS5 looks up the mark of the secondary beam.

(23) Subroutine LABEL5
LABEL5 plots:
1) beam and column marks in elevations and in floor plans.
2) view titles.

(24) Subroutine LINE5
LINE5 draws the grid lines with the corresponding circles and labels.

(25) Subroutine LOGOS5
LOGOS5 draws the complete title block including the corresponding text.
(26) **Subroutine POSIT**

POSIT positiones the column marks to avoid interferences with the lettering for beam marks/sizes.

(27) **Subroutine R53**

R53 reads all the floor data from the AMECO files.

(28) **Subroutine SECBM**

SECBM looks up all the secondary beams, draws them and positions the secondary beam marks.

(29) **Subroutine SIZE5**

SIZE5 plots all the main/secondary beam sizes and column sizes for steel or concrete structures.

(30) **Subroutine UNP53**

UNP53 computes the direction sine and cosine of each main beam.

(31) **Subroutine SYMB5**

SYMB5 draws:

1) End connection symbols for steel structures.

2) Column representation for steel and concrete structural plans.
**Suproutine C1plot (NE)**

```plaintext
COMMON MAT, MEM, LEV, ISOLV, LFVIN, LFIN, LVMAX, NL, IFLEV, IFFRM
COMMON ZZZZ, NME, DAT(3), IDOC(6), IDOS(6), DAT, ICRIS(6), ICRI6(6)
COMMON JSTOP, NSSTOP, IPRT, ILINK, EC1, EC5, MAXFR, ITESN, IRUN, LVU, LNU
COMMON KAK, LIST(10), IO9, CODE(2), MDT(20), UF(20)
COMMON TTC1, NEZ(90)

COMMON JSF(10), BGN(20), LEN(20)
COMMON TAKI(10), JET(10), KET(20), LET(20)
INTEGER KAK(90, 2), NPE(11, 3)
COMMON PACK1, CT(80), O4(80), KEY(80)
COMMON PLOTIN/JPLT(25), IJPLT(25)
COMMON/DYPIN/LYN(10), DYN(20)

COMMON/PLAN/1, IDUM(35)
DIMENSION PL(30), IRIT(140)
DATA PL/SHALL, AFRAI, AFPE, AFRES, AHFLOO, AFHYRA, AFYFRA,
*AFHYM, AFHUM, AFLE, AFHPLA, AFHSE, AFHCON, AFMARK,
*AFHSIZE, AFJOIN, AFSCAT, AFHMP, AFDEVI, AFMAIN, AFTEKT,
*2HMP, AFHCALC, AFHCOU, AFHNB, AFYES, AFHPD, AFZD/

SET IBUG EQ 53 IF DEBUG IS NEEDED
WORD PLOT WAS READ BY CARDI

**IBUG=LET(11)**

### Main commands after plot

<table>
<thead>
<tr>
<th>Command</th>
<th>Set</th>
<th>To</th>
<th>After</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>IPLT(1)</td>
<td>6</td>
<td>SECOND COMMAND ONLY</td>
</tr>
<tr>
<td>FRAM</td>
<td>1</td>
<td>(3)</td>
<td>TO (5), SECOND COMMAND</td>
</tr>
<tr>
<td>PERS</td>
<td>5</td>
<td>(4)</td>
<td>TO (7)</td>
</tr>
<tr>
<td>TRUS</td>
<td>7</td>
<td>(3)</td>
<td>TO (9)</td>
</tr>
<tr>
<td>FLOOR</td>
<td>4</td>
<td>3,</td>
<td>TO (7)</td>
</tr>
<tr>
<td>XFRAM</td>
<td>2</td>
<td>3, 5, 7</td>
<td>SECOND COMMAND, LABEL DATA</td>
</tr>
<tr>
<td>YFRAM</td>
<td>3</td>
<td>3,</td>
<td>TO (6)</td>
</tr>
</tbody>
</table>

### Secondary commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Followed by default</th>
<th>Stored on</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIMENSION</td>
<td>YES=1</td>
<td>IPLT(8)=0</td>
</tr>
<tr>
<td>DUMMY</td>
<td>YES=1</td>
<td>IPLT(9)=0</td>
</tr>
<tr>
<td>LEVEL OR LEV</td>
<td>YES=1</td>
<td>IPLT(12)=0</td>
</tr>
<tr>
<td>SECOND BEAMS</td>
<td>YES=1</td>
<td>IPLT(16)=0</td>
</tr>
<tr>
<td>CONNECTION BEAM</td>
<td>YES=1</td>
<td>IPLT(5)=0</td>
</tr>
<tr>
<td>SECOND</td>
<td>YES=1</td>
<td>IPLT(6)=0</td>
</tr>
<tr>
<td>COLUMN</td>
<td>YES=1</td>
<td>IPLT(13)=0</td>
</tr>
<tr>
<td>BEAM MAIN</td>
<td>YES=1</td>
<td>IPLT(14)=0</td>
</tr>
<tr>
<td>SECOND</td>
<td>YES=1</td>
<td>IPLT(15)=0</td>
</tr>
<tr>
<td>COLUMN</td>
<td>NO=0</td>
<td>IPLT(17)=1</td>
</tr>
<tr>
<td>BEAM MAIN</td>
<td>NO=0</td>
<td>IPLT(18)=1</td>
</tr>
<tr>
<td>SECOND</td>
<td>NO=0</td>
<td>IPLT(19)=1</td>
</tr>
<tr>
<td>JOINT</td>
<td>NO=0</td>
<td>IPLT(11)=1</td>
</tr>
<tr>
<td>SCALE</td>
<td>1/20</td>
<td>IPLT(20)=SC</td>
</tr>
<tr>
<td>DMF(MODIF. FACTOR)</td>
<td>-</td>
<td>IPLT(18)=F</td>
</tr>
<tr>
<td>DEVICE</td>
<td>TEKTRONIK</td>
<td>TEP=1</td>
</tr>
</tbody>
</table>
```
SUBROUTINE CIPLOT/2 74/74  OPT=1

C
C D R A W I N G  S I Z E  A
C
C I F
C
C I F
C
C I F
C
C I F
C
C R A N K  L O C G
C
C S E T  I P L O T  T O  D E F A U L T  V A L U E S  J P L O T
C
I = 2
LET(12) = LET(12) + 1
IF (C401) .EQ. 4.MEPL(IF) RETURN
C
C
CALL LAST(I, NR)
TF(I GT NB) GO TO 4
IF (KEY(I), EQ, 1) GO TO 1003
4 CONTINUE
DO 2 II = 1, 25
I PLOT(II) = J PLOT(II)
2 CONTINUE
IF (IDUM(1),EQ, 7, OR, IDUM(1), EQ, 8) I PLOT(22) = IDUM(1)
IF (IPUG, EQ, 53, OR, IBUE, EQ, 63)
* WRITE(6, *IDUM**, 215) *IDUM(1), I PLOT(22)
IF (I GT NB) GO TO 10
DO 5 J = 1, 4
IF (C01) .NE. PL(J) GO TO 5
GO TO (10, 20, 30, 40) J
CALL GOTOER
5 CONTINUE
GO TO 50
C
C * A L L *  O R  * A L L *  F O L O W E D  B Y  S E C O N D  C O M M A N D
C S E T  I P L O T(1) = 6
C
10 CONTINUE
I = I + 1
I PLOT(1) = 6
CALL LAST(I, NB)
IF (I GT NB) GO TO 195
IF (KEY(I), EQ, 1) GO TO 1003
GO TO 120
C
C S E T  I P L O T(1) = 1
C
20 CONTINUE
I PLOT(1) = 1
21 CONTINUE
I = I + 1
CALL LAST(I, NB)
IF (I GT NB) GO TO 195
IF (KEY(I), EQ, 1) GO TO 1003
IF(CCC(I).EQ.PL(J))GO TO 21
DO 22 J=3,5
IF(CCC(I).EQ.PL(J))GO TO 190

22 CONTINUE
60 TO 120

C PERSPECTIVE FOLLOWED BY P,P, COMM, 4 TO 7
C SET I PLOT(1)=5
30 CONTINUE
I PLOT(1)=5
I=I+1
CALL LAST(I,NB)
IF(I GT NB)GO TO 195
IF(KEY(I).EQ.1)GO TO 1005
DO 32 J=4,7
IF(COI J).EQ.PL(J))GO TO 190
32 CONTINUE
60 TO 1004

C TRUSSES OR TR. FOLLOWED BY 3 5 TO 7, S,C
C SET I PLOT(1)=7

40 CONTINUE
I PLOT(1)=7
I=I+1
CALL LAST(I,NR)
IF(I GT NB)GO TO 195
IF(KEY(I).EQ.1)GO TO 1005
DO 42 J=3,7
IF(COI J).EQ.PL(J))GO TO 190
42 CONTINUE
60 TO 120

C PP, COMMANDS 5 TO 7
C
50 CONTINUE
DO 52 J=5,7
IF(COI J).NE.PL(J))GO TO 52
JJ=J+4
60 TO(69,70,20)JJ
CALL 50TOER
52 CONTINUE
60 TO 120

C FLOOR/FLOOR ALL/FLOOR+PP,COMM/FLOOR+LABEL DATA/
C FLOOR+SECOND COMM/
C SET I PLOT(1)=2
C
60 CONTINUE
I PLOT(1)=4
60 TO 90

C XFRAM/XFRAM+PP, COMM/XFRAM+LABEL DATA/
C XFRAM+SECOND COMM/
C SET I PLOT(1)=2
C
70 CONTINUE
IPLCT(I) = 2
GO TO 90

C SAME AS XFRAM
C YFRAM
C SET IPLCT(I) = 3
C
90 CONTINUE
IPLCT(I) = 3

90 CONTINUE
I = I + 1
CALL LAST(I, NB)
IF(I.GT.NB) GO TO 195
IF(KEY(I).EQ.1) GO TO 100
DO 92 J = 1, 7
IF(CO(I).EQ.PL(J)) GO TO 190
92 CONTINUE
GO TO 120

C SAVE DATA: L1 TO L2/L1 L3..... L5 TO L6
C PACK LAPEL DATA IN PLOT(2), IPLCT(3), IPLCT(4)
C
100 CONTINUE
ITEM = I
ITEM = CO(I)
IF(IPUG.EQ.55) PRINT 999, 1, ITEM
999 FORMAT(1, 102, 146)
CALL SQ2BITS(I, IPLCT(2), ITEM(1), ICR)
I = I + 1
CALL LAST(I, NB)
IF(I.GT.NB) GO TO 195
IF(KEY(I).EQ.1) GO TO 100
DO 102 J = 1, 7
IF(CO(I).EQ.PL(J)) GO TO 190
102 CONTINUE
IF(CO(I).NE.4MTHRU) GO TO 120
I = I + 1
CALL LAST(I, NB)
IF(I.GT.NB) GO TO 180
IF(KEY(I).EQ.0) GO TO 182
L2 = CO(I)
IF(CO(I).LT.6) L2 = CO(I) - 1
IF(CO(I).EQ.(L2 + 1)) GO TO 184
L1 = CO(I)
GO TO 106

184 L1 = CO(I)
106 CONTINUE
DO 108 J = L1, L2
CALL SQ2BITS(I, IPLCT(2), J, IER)
108 CONTINUE
GO TO 90

C SECONDARY COMMANDS
C
120 CONTINUE
I0 = 1
IF(IPUG.EQ.55) PRINT 993, I0, 1, CO(I)
C33 F9M1AT(* I0.+214.7X.A10)
DO 122 J=8+12
    IF(CO(I).NE.PL(J))60 TO 122

121 I=I+1
    CALL LAST(I,NB)
    IF(I.GT.NB)GO TO 1001
    IF(KEY(I).EQ.1)GO TO 1003
    IF(CO(I).EQ.PL(26))GO TO 180
    IF(CO(I).EQ.PL(27))GO TO 144

    IF(J.LT.12)GO TO 1004
    IF(J.EQ.12.AND.CO(I).EQ.PL(25))60 TO 121
    GO TO 1204
122 CONTINUE
    DO 124 J=13,16
        IF(CO(I).NE.PL(J))GO TO 124
        I=I+1
        JQ=2
        IF(IBUE.EQ.53)PRINT 993.I0.+1,CO(I)
        CALL LAST(I,NR)
        IF(I.GT.NR)GO TO 1001
        IF(KEY(I).EQ.1)GO TO 1003

        IF(CO(I).EQ.PL(26))GO TO 140
        IF(CO(I).EQ.PL(27))GO TO 144
        IF(J.EQ.16)GO TO 1804
        IF(J.EQ.13)GO TO 1806
        IF(J.EQ.12.16)GO TO 176

        GO TO 1004
    124 CONTINUE
C
    C SCALE AND DMF
C
    DO 128 J=17,18
        IF(CO(I).NE.PL(J))GO TO 128
        I=I+1
        CALL LAST(I,MR)
        IF(I.GT.MR)GO TO 1001
        IF(KEY(I).EQ.0)GO TO 1002
        IF(J.EQ.18)GO TO 125

        IF(CO(I).NE.1)GO TO 1808
        I=I+1
        IF(J.EQ.17)IPLT(20)=CO(I)
    125 IF(J.EQ.18)IPLT(18)=CO(I)
        I=I+1
        CALL LAST(I,NB)
        IF(I.GT.NB)GO TO 195
        IF(KEY(I).EQ.1)GO TO 1803
        DO 126 JJ=2,7
            IF(CO(I).EQ.PL(JJ))GO TO 190
        126 CONTINUE
        GO TO 120
    128 CONTINUE
        IF(CO(I).EQ.PL(19))GO TO 130
        IF(CO(I).EQ.HDRAW)GO TO 260
        GO TO 1804
C
    C DEVICE
C

110 CONTINUE
I=I+1
CALL LAST(1,IV)
IF(I.GT.NP)GO TO 1001
IF(KEY(I).EQ.1)GO TO 1003
GO TO 120 JI=23
IF(CO(I).NE.PL(JJ))GO TO 134
IF(JJ.EQ.23)IPL(7)=2
IF(JJ.EQ.23)IPL(7)=3

134 CONTINUE
I=I+1
CALL LAST(1,IV)
IF(I.GT.NP)GO TO 195
IF(KEY(I).EQ.1)EC TO 1002
GO 176 J1=2
IF(CO(I).EQ.PL(J1))60 TO 190

136 CONTINUE
GO TO 120

138 CONTINUE

C

C *YES*

C

140 CONTINUE
IF(J.LT.15)GO TO 140
IF(J.EQ.16)GO TO 142
IF(J.EQ.23)GO TO 141
IPL(17)=1

141 IPL(1A)=1
IPL(19)=1
GO TO 100

142 CONTINUE
IPL(111)=1
GO TO 100

C

C *NO* & TO 16

C

144 CONTINUE
J2=J-7
GO TO (150,152,154,156,158,160,162,164,J2)
CALL GCOER

150 CONTINUE
IPL(8)=0
GO TO 100

152 CONTINUE
IPL(9)=0
GO TO 100

154 CONTINUE
IPL(12)=0
GO TO 100

156 CONTINUE
IPL(16)=0
GO TO 100

158 CONTINUE
IPL(5)=0
IPL(16)=0
SUBROUTINE CIPLT/7 74/74 OPT=1

GO TO 140
140 CONTINUE
IPL(14) = 0
IPL(15) = 0
GO TO 140
142 CONTINUE
IF (J1 .EQ. 25) GO TO 163
IPL(17) = 1
163 IPL(19) = 0
IPL(11) = 0
GO TO 140
164 CONTINUE
IPL(11) = 1
GO TO 140

C

C WORD "BEAM" FOR CONNECTION, MARK, AND SIZE
C

166 CONTINUE
J1 = 0
IF (J .EQ. 14 .OR. J .EQ. 15) J1 = 25
I = I + 1
CALL LAST(I, NB)
IF (I .GT. NB) GO TO 1001
IF (KEY(I) .EQ. 1) GO TO 1003
IF (BAG .EQ. 53) PRINT 999, I, CO(I)
999 FORMAT(*, IQS*, '16.1X, A10')
IF (CO(I) .EQ. PL(20)) GO TO 168
IF (CO(I) .EQ. PL(12)) GO TO 172
IF (CO(I) .EQ. PL(26)) GO TO 140
IF (CO(I) .EQ. PL(27)) GO TO 144
GO TO 1904

C

C "MAIN"
C

168 CONTINUE
I = I + 1
CALL LAST(I, NP)
IF (I .GT. NP) GO TO 1001
IF (KEY(I) .EQ. 1) GO TO 1003
IF (CO(I) .EQ. PL(27)) GO TO 179
IF (J .LT. 15) GO TO 180
IPL(14) = 1
GO TO 140
170 CONTINUE
IF (J .EQ. 15) GO TO 180
IF (J .EQ. 13) IPL(15) = 0
IF (J .EQ. 14) IPL(14) = 0
GO TO 180

C

C "SECONDARY"
C

172 CONTINUE
I = I + 1
CALL LAST(I, NB)
IF (I .GT. NB) GO TO 1001
IF (KEY(I) .EQ. 1) GO TO 1003
IF (CO(I) .EQ. PL(27)) GO TO 174
IF(J.LT.15) GO TO 180
IPLCT(15)=1
GO TO 170

174 CONTINUE
IF(J.EQ.15) GO TO 180
IF(J.EQ.14) IPLCT(6)=0
IF(J.EQ.14) IPLCT(15)=0
GO TO 180

C *COLUMN*

17A CONTINUE
I=I+1
CALL LAST(I, NP)
IF(I.GT.NP) GO TO 1001
IF(KEY(I).EQ.11) GO TO 1003
IF(CO(I).EQ.PL(27)) GO TO 170
IF(J.EQ.14) GO TO 180
IF(J.EQ.15) IPLCT(17)=1
GO TO 180

17B CONTINUE
IF(J.EQ.15) GO TO 180
IF(J.EQ.14) IPLCT(13)=0
GO TO 180

C DRAWING SIZE
C DEFAULT SIZE D IPLCT(21)=3

200 CONTINUE
I=I+1
CALL LAST(I, NB)
IF(I.GT.NB) GO TO 1001
IF(KEY(I).EQ.1) GO TO 1003
IF(CO(I).EQ.4HSL) GO TO 200
ITEMP2=9
IF(CO(I).EQ.1H) ITEM=2=5
IF(CO(I).EQ.2HM) ITEM=2=7
IF(CO(I).EQ.1HC) ITEM=2=5
IF(CO(I).EQ.1MD) ITEM=2=3

IF(CO(I).EQ.1ME) ITEM=2=1
IF(ITEMP2.EQ.0) GO TO 1004
IPLCT(21)=ITEMP2

C

C WRITE OUT AND GO BACK
C
C********************************************

180 CONTINUE
I=I+1
CALL LAST(I, NB).
IF(I.GT.NB) GO TO 195
IF(KEY(I).EQ.1) GO TO 1003
DO 182 JJ=2,7
IF(CO(I).EQ.PL(JJ)) GO TO 190
182 CONTINUE
GO TO 120

190 CONTINUE
LEFT(12) - LET(12) + 1
195 CONTINUE
WRITE(*,750) I PLOT
750 FORMAT(I20)
IF(IPUG.EQ.53) PRINT 995, I PLOT(I), LET(1?)
995 FORMAT(I5,*174)
C0 960 IPOS=1*140
CALL USOZR(1, I PLOT(2), IPOS, 1PI, TEP)
IRIT(IPPOS)=IPI
960 CONTINUE
C0 980 I=1*140
IF(IPUG.EQ.53) PRINT 996, I, IRIT(I)
986 FORMAT(* 1019, 2013)
550 CONTINUE
IF(IPUG.EQ.53) PRINT 997, (I PLOT(J), J=2*4)
997 FORMAT(* 1019, 3120)
IF(IPUG.EQ.53) PRINT 998, (I PLOT(J), J=5*20)
998 FORMAT(* 1019, 514)
C PEWIND F1
C DO 962 IRC=1*2
C READ(*,750) I PLOT
C DO 961 IPOS=1*140
C CALL USOZR(1, I PLOT(2), IPOS, 1BI, IRC)
C IRIT(IPOS)=IRI
C 901 CONTINUE
C DO 952 I=1*140
C IF(IPUG.EQ.53) PRINT 996, I, TBIT(I)
C 952 CONTINUE
C 902 CONTINUE
IF(I.GT.NB) GO TO 1010
GO TO 4
C ------------------------------------ MESSAGES
C INCOMPLETE
1001 CONTINUE
IF(KEY(I).NE.0) CO(I)=4H =
CALL SYNTAX(I, I, CO(I)) + 1
GO TO 1010
C NUMBER EXPECTED
1002 CALL SYNTAX(2, I, CO(I)) + 1
GO TO 1010
C WORD EXPECTED
1003 CALL SYNTAX(3, I, CO(I)) + 1
GO TO 1010
C ILLEGAL WORD
1004 CALL SYNTAX(4, I, CO(I)) + 1
GO TO 1010
C ILLEGAL NUMBER
1005 CALL SYNTAX(5, I, CO(I)) + 1
GO TO 1010
C ILLEGAL WORD, WORD ASSUMED ---
1006 CALL SYNTAX(6, I, CO(I), DUM)
GO TO 1010
C ILLEGAL NUMBER, NUMBER ASSUMED ---
1007 CALL SYNTAX(7, I, CO(I), WS)
GO TO 1010
C MAX EXPECTED VALUE EXCEEDED, VALUE ACCEPTED
1009 CALL SYNTAX(9, I, CO(I), 7.)
SUBROUTINE CIPLCT/IO  7/17A  (F1=1)

C
1010 CONTINUE C

RETURN
END

<table>
<thead>
<tr>
<th>ADDRESS</th>
<th>BLOCK</th>
<th>DESCRIPTION</th>
<th>TYPE</th>
<th>SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>6420</td>
<td></td>
<td></td>
<td>REAL</td>
<td>20</td>
</tr>
<tr>
<td>0P</td>
<td>PACK</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1705</td>
<td></td>
<td></td>
<td>REAL</td>
<td>20</td>
</tr>
<tr>
<td>1765</td>
<td></td>
<td></td>
<td>REAL</td>
<td>2</td>
</tr>
<tr>
<td>1717</td>
<td></td>
<td></td>
<td>REAL</td>
<td>3</td>
</tr>
<tr>
<td>1455</td>
<td></td>
<td></td>
<td>INTEGER</td>
<td>140</td>
</tr>
<tr>
<td>1671</td>
<td></td>
<td></td>
<td>INTEGER</td>
<td>6</td>
</tr>
<tr>
<td>1038</td>
<td></td>
<td></td>
<td>INTEGER</td>
<td>6</td>
</tr>
<tr>
<td>74P</td>
<td></td>
<td></td>
<td>INTEGER</td>
<td>6</td>
</tr>
<tr>
<td>74R</td>
<td></td>
<td></td>
<td>INTEGER</td>
<td>6</td>
</tr>
<tr>
<td>60B</td>
<td></td>
<td></td>
<td>INTEGER</td>
<td>6</td>
</tr>
<tr>
<td>66B</td>
<td></td>
<td></td>
<td>INTEGER</td>
<td>6</td>
</tr>
<tr>
<td>0P</td>
<td>PACK</td>
<td></td>
<td>INTEGER</td>
<td>35</td>
</tr>
<tr>
<td>1700A</td>
<td></td>
<td></td>
<td>INTEGER</td>
<td>6</td>
</tr>
<tr>
<td>12P</td>
<td></td>
<td></td>
<td>INTEGER</td>
<td>6</td>
</tr>
<tr>
<td>15P</td>
<td></td>
<td></td>
<td>INTEGER</td>
<td>6</td>
</tr>
<tr>
<td>1675B</td>
<td></td>
<td></td>
<td>INTEGER</td>
<td>6</td>
</tr>
<tr>
<td>1149</td>
<td></td>
<td></td>
<td>INTEGER</td>
<td>6</td>
</tr>
<tr>
<td>422P</td>
<td></td>
<td></td>
<td>INTEGER</td>
<td>6</td>
</tr>
<tr>
<td>31P</td>
<td>PACK</td>
<td></td>
<td>INTEGER</td>
<td>25</td>
</tr>
<tr>
<td>1710P</td>
<td></td>
<td></td>
<td>INTEGER</td>
<td>6</td>
</tr>
<tr>
<td>113B</td>
<td></td>
<td></td>
<td>INTEGER</td>
<td>6</td>
</tr>
<tr>
<td>1704B</td>
<td></td>
<td></td>
<td>INTEGER</td>
<td>6</td>
</tr>
<tr>
<td>1218</td>
<td></td>
<td></td>
<td>INTEGER</td>
<td>6</td>
</tr>
<tr>
<td>38</td>
<td></td>
<td></td>
<td>INTEGER</td>
<td>6</td>
</tr>
<tr>
<td>475B</td>
<td></td>
<td></td>
<td>INTEGER</td>
<td>6</td>
</tr>
<tr>
<td>1676B</td>
<td></td>
<td></td>
<td>INTEGER</td>
<td>6</td>
</tr>
<tr>
<td>1707B</td>
<td></td>
<td></td>
<td>INTEGER</td>
<td>6</td>
</tr>
<tr>
<td>1677B</td>
<td></td>
<td></td>
<td>INTEGER</td>
<td>6</td>
</tr>
<tr>
<td>1208</td>
<td></td>
<td></td>
<td>INTEGER</td>
<td>6</td>
</tr>
<tr>
<td>1674B</td>
<td></td>
<td></td>
<td>INTEGER</td>
<td>6</td>
</tr>
</tbody>
</table>

C

RETURN
END
EQUIVALENCE (JER(1),PHIC(1)),(ISTAC(1),CP(1))
CALL ZED0(ISTAC,140)

C START TO COMPUTE JOINT COORDINATES

CALL ZF=0(CY+092)
CALL ZF=0(CZ+992)
CALL ZF=0(IFLA+4)
CALL ZF=0(JBR+992)

IPUG=LF(I1)
LVI=LV=LVNC=IDOS(4)+1
NCP=1+(LVNC-LV1)/LV4

C COMPUTE FIRST STACK AND LAST STACK NO. ON FRAME NFR

KA=FIRST STACK

KP=LAST STACK

DO 50 IFR=1,NFR

KA=COL1(NFR,1W)
IF(IFR.LT.IFR)KP=COL1(NFR+1,1W)-1
IF(IFR.GT.IFR)KP=MB

C SEARCH FOR ONE STACK WITHOUT SHORT DUMMY COLUMN

C THIS STACK WILL BE REFERENCE STACK

C SET FLAR=IFLA =0 TO INDICATE STACK HAD BEEN FOUND

C SET ISTAC(K)=0 ,ISTAC(K)=1 IF SHORT OR DUMMY THROUGHOUT COLUMN

C HAS BEEN FOUND

IFLA=1
LEVEL=LV4
DO 3 LL=KA,KP

2 LVN=(KK-1)*LV4+LEVEL
IF(CSTIF(LVN).LE.1.E-20)GO TO 4
GO TO 3

4 CONTINUE
IF(KK.EQ.KB)LEVEL=LEVEL-1
IF(KK.NE.KB)GO TO 3
KK=KK+1

GO TO 2
3 CONTINUE

9 CONTINUE

DO 100 K=KA,KB
ISTAC(K)=0
IF(K.NE.KA)GO TO 6

NCP=KA
LV=LV1
LVNC=(NCP-1)*LV4+LV

6. CONTINUE

C SEARCH FOR JOINT BELOW REFERENCE LEVEL

10 =4
IF(IPUG.EQ.53)
1PRINT 996,10,K,LV,LVNC,NCP,KB,IDOS(3),KC,KCC,CX(LVNC),CX(LVNC)
996 FORMAT(10,9I4,2F18.2)
CALL STACDM5(LV,LVNC,K)
10 = 1
IF(IFLAG .EQ. 1) GO TO 66
IF(IBUG .EQ. 53)

1PRINT 991, IQ, KV, LV, LVNC
991 FORMAT(I0, 9I4, 14)
IF(IBUG .EQ. 53)

1PRINT 993, (KK, C2(KK), KK, LVNC, LVNC)
993 FORMAT(I0, 9I4, 14,F10.2)
5 CALL STACUP5(LV, LVNC, K)
10 = 2
IF(IFLAG .EQ. 0) GO TO 68
IF(IBUG .EQ. 53)

1PRINT 994, IQ, KV, LV, LVNC, L, CX(L), C2(L)
6A CONTINUE
IF(ISTAC(K), EQ. 1) GO TO 8
IF(L .EQ. 0 .AND. IFLAG .EQ. 0) GO TO 8
IF(L .EQ. 0 .AND. IFLAG .EQ. 1) GO TO 11
IF(L .EQ. LEVEL .AND. IFLAG .EQ. 0) GO TO 8
IF(IFLAG .EQ. 0) GO TO 7
11 KCC = K
KCC = K
7 CONTINUE
IFLAG = 0
LV = LV1
LVNC = (NCP-1) * LVNA * LV
CX(LVNC) = 0.
C2(LVNC) = 0.
GO TO 6
8 CONTINUE
10 = 8
IF(IFUG=EQ.5) THEN
   PRINT *,-10,K,NCNP,KA,RP,KC,IFLA,IFR,NFR
END IF

IF(IFLA.EQ.1) THEN
   IF(IFLAG.EQ.1) THEN
      GO TO 84
   ELSE
      IF(NCP.GT.0 .AND. KCP.EQ.1) THEN
         GO TO 90
      END IF
   END IF
   GO TO 85
END IF

84 CONTINUE
LV=LV1
LVNC=(KCP-1)*LVM4+LV
GO TO 85

85 CONTINUE
LV=LV1
LVNC=(NCNP-1)*LVM4+LV

C ADJUST X-COORDINATES
C
C LOOK UP FOR NEXT STACK TO THE RIGHT
C
86 CALL STACR15(LV,NCNP,LVNC,K)
   IF(KC.EQ.KA) THEN
      GO TO 87
   ELSE
      IF(IFLAG.EQ.0) THEN
         GO TO 100
      END IF
   END IF
   GO TO 86

87 CONTINUE
IFLAG=0
LV=LV1
NCNP=KCP
LVNC=(NCNP-1)*LVM4+LV
GO TO 85

90 CONTINUE
IF(KC.EQ.KA) THEN
   GO TO 100
END IF
IF(IFLAG.EQ.1) THEN
   GO TO 100
END IF
LV=LV1
LVNC=(KC-1)*LVM4+LV
IFLAG=0
CALL STACLE5(LV,NCNP,LVNC,K)
CALL STACOM5(LV,LVNC,K)

998 FORMAT(10*16,814,2F7.2)!

CALL STACUP5(LV,LVNC,K)
IF(IPUG.EQ.53) THEN
   PRINT 998,10,K,LV1,LV2,LVNC,L,KC,KCC,CX(L),CZ(L)
END IF

998 CONTINUE

C IF(KC.EQ.KA) THEN
   GO TO 100
END IF
LV1=(KA-1)*LVM4+1
LV2=KB+LVM4
CX1=CX1(LV1)
DO 96 KK=LV1,LV2
   IF(CX4(KK).EQ.999) THEN
      GO TO 96
   END IF
96 CONTINUE
C END OF ROUTINE
C DETERMINE JOINT FOR BRACINGS
96 CONTINUE
100 CONTINUE

IF(MSW.EQ.1) GO TO 500
KP1=KB-1
DO 400 I=KA,KP1
DO 350 LV=2*LVMA+LV
LN2=(I-1)*LVMA+LV
400 CONTINUE

IQ=10
IF(IEUG.EQ.53) 1PRINT 099, IO, NFR, LN2, KA, KP1
IF(IPR(LN2).LT.10) GO TO 350
NRM=1
LVNC1=LN2
350 NCPR=KP1
LVNC1=JPR152(NCP*LVNC1)
IF(LVNC1.EQ.0) GO TO 380
IF(CLINT(LVNC1).LT.1) GO TO 320
GO TO 360
320 CONTINUE

NCPR=NRM
JJ=MERD5(2*LV,1+LN2,DUM(1),1,LVT)
JR=MERD5(2*LV,NCPR+LVNC1,DUM(1),1,LVT)
JQ=11
IF(IEUG.EQ.53) 1PRINT 099, IO, LN2, NCPR, LV, JJ, JR
099 FORMAT(18B8)
JBR(LN2)=JJ+JR+10**5
380 CONTINUE
400 CONTINUE
500 CONTINUE
NC=MB+LVMA

501 CONTINUE

IF(IEUG.EQ.53) 1PRINT 997, IO, NFR, NB, NC, LVMA
WRITE(A1,800) IW, IU, NW, NB, NC, IFR, INWAY(IW), LVMA, MB, LD, MSW
800 FORMAT(1I10)
WRITE(A1,810)(MKB(K),K=1,NB),(MKC(K),K=1,MB)
1,(COL1(K),K=1,IFR)
WRITE(A1,805)(KST(K),K=1,NB)
805 FORMAT(I15)
WRITE(A1,810)(BSTIF(K),K=1,NB),
1(CLEN(K),CSTIF(K),CX(K),CZ(K),PHIC(K),DV(K),K=1,NC),

2(TRANS(K),K=1,LVMA)
810 FORMAT(1E12.5)
WRITE(A1,820)(IBR(K),JBR(K),K=1,NB)
820 FORMAT(I15)
IFRAM=1
IF(INWAY(IW).EQ.2) IFRAM=1
IF(IEUG.EQ.53) 1PRINT 995, IFRAM
995 FORMAT(* **** FRAME **** DIRECTION *****)
IF(IEUG.EQ.53) 1PRINT 999
999 FORMAT(* IO *,* JOINT NO. LEV CX CZ CLEN BLEN * )
<table>
<thead>
<tr>
<th>13D</th>
<th>/CONST/</th>
<th>REAL</th>
<th>Y</th>
<th>HR[1]</th>
<th>630228</th>
<th>/FRAME/</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>/BATS/</td>
<td>REAL</td>
<td>10</td>
<td>MX</td>
<td>1116B</td>
<td>/FRAME/</td>
</tr>
<tr>
<td>5070</td>
<td>/FRAME/</td>
<td>REAL</td>
<td>21</td>
<td>IER</td>
<td>1362B</td>
<td>/FRAME/</td>
</tr>
<tr>
<td>3070</td>
<td>/DATE/</td>
<td>REAL</td>
<td>980</td>
<td>IFIR1</td>
<td>52635B</td>
<td>/FRAME/</td>
</tr>
<tr>
<td>7216</td>
<td>/FRAME/</td>
<td>REAL</td>
<td>980</td>
<td>IFUSG</td>
<td>1338B</td>
<td>/FRAME/</td>
</tr>
<tr>
<td>642R</td>
<td></td>
<td>REAL</td>
<td>20</td>
<td>ICRIIC</td>
<td>1038B</td>
<td>/FRAME/</td>
</tr>
</tbody>
</table>

| 1447 | /FRAME/ | REAL | 980| ICRTIS | 758B   | /FRAME/ |
| 3373 | /FRAME/ | REAL | 989| IDFR   | 684328B| /FRAME/ |
| 6535 | /FRAME/ | REAL | 12 | TDCC   | 688B   | /FRAME/ |
| 1441 | /FRAME/ | REAL | 10 | IDOS   | 668B   | /FRAME/ |
| 550B | /DATE/  | REAL | 28 | IEOF   | 128B   | /FRAME/ |
| 2975 | /FRAME/ | REAL | 992| IFFRM  | 128B   | /FRAME/ |

| 370B | /DATE/  | REAL | 112| IFRA   | 08B    | /FRAME/ |
| 00   | /FRAME/ | REAL | 989| IFLEV  | 118B   | /FRAME/ |
| 132B | /FRAME/ | REAL | 980| IFIR   | 1443B  | /FRAME/ |
| 253B | /DATE/  | REAL | 989| IFRAM1 | 1376B  | /FRAME/ |
| 275B | /DATE/  | REAL | 10 | INAY   | 1148B  | /FRAME/ |
| 1701B| /FRAME/ | REAL | 992| INFR   | 1118B  | /FRAME/ |

| 423B |         | REAL | 2  | INLEV  | 1148B  | /FRAME/ |
| 1012 | /FRAME/ | REAL | 60 | INMB   | 112B   | /FRAME/ |
| 00   | /CONST/ | REAL | 10 | INWAY  | 118B   | /FRAME/ |
| 1311B| /FRAME/ | REAL | 992| IOP    | 422B   | /FRAME/ |
| 1505B| /FRAME/ | REAL | 992| IPRINT | 1138B  | /FRAME/ |

| 3702B| /FRAME/ | REAL | 992| IG     | 1351B  | /FRAME/ |

| 1560B|         | REAL | 992| I RUN  | 121B   | /FRAME/ |
| 3167B| /FRAME/ | REAL | 992| I SOLV | 3B     | /FRAME/ |
| 1353B|         | REAL | 3  | ISTAT1 | 33127B | /FRAME/ |
| 55B  |         | REAL | 3  | ITC1   | 475B   | /FRAME/ |
| 5070B| /FRAME/ | REAL | 992| ITEST  | 120B   | /FRAME/ |
| 266B | /FRAME/ | REAL | 60 | IIV    | 5070B  | /FRAME/ |

| 3116B| /FRAME/ | REAL | 992| IVAY   | 1442B  | /FRAME/ |
| 3312B| /FRAME/ | REAL | 992| IVIX   | 1376B  | /FRAME/ |
| 5070B|         | REAL | 35 | IWI    | 5071B  | /FRAME/ |
| 3567B| /FRAME/ | REAL | 992| JBR    | 1347B  | /FRAME/ |
| 5070B| /FRAME/ | REAL | 992| JET    | 724B   | /FRAME/ |

| 115B |         | REAL | 60 | JIF    | 5070B  | /FRAME/ |
| 116B |         | REAL | 28 | JSTO P | 111B   | /FRAME/ |
| 5070B| /FRAME/ | REAL | 91 | K      | 1347B  | /FRAME/ |
| 6503B| /FRAME/ | REAL | 91 | KA     | 1342B  | /FRAME/ |
| 6162B| /FRAME/ | REAL | 992| KAK    | 124B   | /FRAME/ |
C
DIMENSION LIND2(171),CJP(18),CLDAT(18,F)
COMMON /ADAT2/ LIND2,CJP,CLDAT
C
EQUIVALENCE (LINDEX,LIND2)(CJP,HIZ),(CLDAT,HIZ(19))
COMMON /AIN2/ MSOP2,HIND2(91)
C
FRAME DATA
DIMENSION CGA(490),CGA1(99),DIST(39,2)
1,1,MC(149,2),COL1(30,2),INWAY(2),INFR(2),INPA(2),INLE(2)
2,MC(2),LDTY(28),PERC(90,2)
C
LDLOD8(28),LDWAY(28),ULTVAL(28),LATIN(28)
INTEGER COL1,BYPASS,BAYMY
C
COMMON TO A AND B
DIMENSION BSTIF(980),BSTIF1(980),BLEN(980),BLEN1(980)
1,1,MB(982),CSTIF1(992),CSTIF1(992),CLENI(992),CAK(992),PA(992)
2,PDI(992),PMAS(540),PMSTI(90,2)
C
3,INR(990),IBR1(990),LDYES(11),P01(992)
EQUIVALENCE (PO1(1),PA1)
C
FRAME B ONLY
DIMENSION DR(992),DR1(992),DV(992),PA(992),PHIC(992)
A,PHIC1(992),TAI(992),TD(91),P1(91),SEL(91),TG(91),TH(91)
R,TRANS(11),TRANS(11),NCI(148,2)
C
COMMON/FRAME/CGA,CGA1,LDWS,LDSW,DISM,MC1,COL1,INWAY,INFR,INPA
A,INLE(2),LDTY,PERC,PMEX,BYPASS,INWAY,INFR,MR1,LEVX
B,BSTIF,BSTIF1,BLEN,BLEN1,MB,CSTIF1,CSTIF1,CAK,PA,PMAS
C,PMAS1,PMST1,DR1,DR1,DV,PA,PHIC,PHIC1,TA1,TD,P1,SEL,TG,TH,TRANS
R,TRANS1,NC1,KNOM,KNOM1,KSHR,KSHR1,KD1,KD2
C
LDLOD8,LDWAY,HRFL1(91),HRFL1(91),LDYES,BUF(12)
EQUIVALENCE (NIF,LSAVE)
COMMON/LEN/BLN,MRP,LBAST
C
DIMENSION CX(992),CZ(992)
EQUIVALENCE (CX(11),CZ(11),(CZ(1),DR1))
COMMON/PLTFL/FLAK,KN,LN,LEVEL
DIMENSION ISTAC(190)
EQUIVALENCE (JBR,PHIC1),(ISTAC(1),DR1(1))
IF(IRUG.EQ.11) THEN
  IF(IFLAG.EQ.0) AND .AND. IRUG.EQ.53) THEN
    STOP
  ELSE
    FORMAT(*,100,*,JOINMC,LEV,CX,C2,CLEN,BLEN)
  END IF
END IF

10 L=MCLD*2*(LV+LVNC)
   IF(CLEN+LVNC).LE.0) GO TO 40
   IF(L.EQ.0) GO TO 40
   IF(IFLAG.EQ.0) GO TO 15
   IF(CLEN+LVNC).GT.1) THEN
     IF(IFLAG.EQ.0) THEN
       ISTAT(K)=1
     ELSE
       SSTAT(K)=1
     END IF
   END IF
   IF(STAT(K).EQ.100) THEN
     ISTAT(K)=1
   ELSE
     IF(IFLAG.EQ.0) THEN
       SSTAT(K)=1
     END IF
   END IF
   IF(CLEN+LVNC).LE.20) AND .AND. (LVNC-L).GT.1) THEN
     SSTAT(K)=1
   END IF
   CONTINUE
15 CV(L)=C(L)+LVNC
   IF(CLEN+LVNC).LE.1) GO TO 40
   IF(CLEN+LVNC).EQ.1) THEN
     LVNC=(C-1)+LVNC+LV+L-LVNC
   END IF
   CL(L)=C(L)+LVNC
   GO TO 25
20 C2(L)=C(L)+LVNC-1.*CLEN(LVNC)
   IF(IFLAG.EQ.1) THEN
     CONTINUE
   END IF
   IF(CLEN+LVNC).EQ.2) THEN
     CONTINUE
   END IF
   L1=L+1
   L2=L+L-1
   DO 25 T=L1,L2
     CX(I))=-999
  25 CONTINUE
  26 CONTINUE
  28 CONTINUE
  IF(IFLAG.EQ.0) AND .AND. IRUG.EQ.53) Then
    STOP
  ELSE
    FORMAT(*,100,*,JOINMC,LEV,CX(L),CZ(L),
     CLEN(LVNC),BLEN(LVNC))
  END IF
992 CV(L)=LEVD
   LVNC=L
   IF(LVNC.EQ.1) THEN
     GO TO 30
   END IF
   CONTINUE
   RETURN
40 CONTINUE
C ERROR MESSAGE COL. DOES NOT EXIST
C
IF(IRUG.EQ.53)
  PRINT 994,L,LVNC,L,CLEN(LVNC)
994 FORMAT(*,ERROR FROM STADK LV LVNC L CLEN=*,SMIT*,F12.2)
STOP
RETURN
* NO PATH TO THIS STATEMENT
END
COMMON MAT, FM, LEVEL, ISOLV, LFIN, LNFINT, LMAX, LMAXV, NL, IFLEV, IFRM
COMMON ZZ, ZE, NDAT, IDDC(6), IDOS(6), MAFL, ICRI(6), ICRI(6)
COMMON JSTOP, STOP, IPRINT, LNKEC, EC, MAXF, ITEST, IRUN, LUE, LNU
COMMON KAK, LTR(10), IOF, CODE(3), ND(20), UF(20)
COMMON TC, NFZ(90), USF(10)
COMMON FON(20), DON(20)
COMMON TAKI(10), JET(10)
COMMON KET(20), LET(20)
COMMON HATS, PATP(10)
COMMON MASSES, MASS(24)
DIMENSION NME(11+3)
INTEGER KAK(90)
COMMON /CONSTS/ CONSTS(10), IEOF, ACC(?)
COMMON /TFUS/ TFUS(7)
COMMON FNLNT, DUM(35), LEDAD
COMMON DAD, INDEX(92)
COMMON /DADAT/ INDEX(171), HIZ(20), INDEX(21)
*DI, CBM(22), 28), CAI(28), V1Z(28), RC(22, 28)
DIMENSION LIND2(171), CJP(18), CLDAT(10+6)
COMMON /DADAT/ LIND2, CJP, CLDAT
C EQUIVALENCE (INDEX, LIND2), (CJP, HIZ), (CLDAT, HI719))
COMMON /DAD2/ MSOP2, MIND2(91)
C FRAME DATA
DIMENSION CGA(90), CGA(90), DIST(30, 2)
1, MKC(140), COL(30, 2), INWAY(2), IFR(2), INMP(2), INLEV(2)
2, HI(2), LDY(2), BPERC(90, 2)
5, LDOLY(28), UTV(2), LTV(28), LATINF28)
INTEGER COL, BYPASS, BAYM
C COMMON TO A AND R
DIMENSION BSTIF(980), PSTIF(980), BLEN(960), PLEN(980)
1, MKB(980), CSTIF(992), CSTIF(992), CLEN(992), CAY(992), PB(992)
2, PC(992), PMAS(940), PMAS(450), PMAS(40, 2)
3, IBR(980), IBR(980), LDY(28), PD(992)
EQUIVALENCE (PD(1), PB(1))
C FRAME B ONLY
DIMENSION DR(992), DR(992), DTV(992), PAT(992), PHIC(992)
A, PHIC(992), TAI(992), TR(91), P(91), P(91), SELI(91), T(91), T(91)
B, TRANSI(91), TRANSI(91), NC(1140, 2)
C COMMON /FRAMB/ CGA, CGA, LSW, LSWI, DIST, MKC, COL, INWAY, INR, INMP
A, INLEV, HX, LDY, BPERC, MBEX, Bypass, INAY, IFR, MB, H1LEV
B, BSTIF, BSTIF, BLEN, BLEN, MKB, CSTIF, CSTIF, CLEN, CAY, PB, PC, PMAS
C, PMAS, PMAS, DR, DR, DTV, PAT, PHIC, PHIC, TAI, TDI, PB, P(91), SELI, T, T, TRANS
B, TRANS, NC, KHM, KHM, KSFI, KSHR, KSHR, KDY, KDVI
C, LDOLY(980), MIFS, JSIF, DESL, LTV, BAYM, EST, INV(1, 1)
IBR, IBR, KST(980), KSTI(980), MSW, DFR(140, 2), HRJ(992), HRJ(1992)
6, STB
HLDOLY, LTV, MFL(91), MFLI(91), LDYES, BUF(12)
EQUIVALENCE(MIFS, LSVEA)
COMMON /LEN/ BLEN, NRM, NBLAST
C DIMENSION CX(992), CZ(992)
EQUIVALENCE (CX(1), PA(1)), CZ(1), DR(1)
COMMON /PLOT/ FLA, KC, KNC, LOC, LEVEL
DIMENSION ISTAC(140)
EQUIVALENCE (JBR, PHIC(1)), (ISTAC(1), DR(1))
SUBROUTINE STACK

L1 = LVNC

10 L = MCOLU = 2(LV + LVNC)

15 CONTINUE

20 C2(L) = C2(LVNC)

25 LEV = LV + L - LVNC

30 CONTINUE

C ERROR EXIT (COL. DOES NOT EXIST)

IF (IBUG EQ 53)
1PRINT 994, LV, LVNC, L, CLENI(L)
994 FORMAT(*) ERROR FROM STADD LV LVNC L CLENI, 3I4, F12.2)
STOP
SUBROUTINE STANCES(LV,NCP,LVNC,Y)
COMMON MAT,MEM,LEV,ISOLV,LYFIL,LMFTN,LMAX,LMN,TLEV,IFRM
COMMON NULL,NME,DATE(7),1DOCC(6),1DOS(6),MAXFL,ICRIS(6),ICRIC(6)
COMMON STOP,STOP,IPRN,ILINK,ECI,ES5,MAXFR,AFTEST,IRUN,LW,LAU
COMMON KAK,LIST(10),1D9,CODE(2),MD(20),UF(28)
COMMON TCI,NE(90),USF(10)
COMMON PON(28),LOK(20)
COMMON TAKI(10),E(T(10)
COMMON KET(20),LET(20)
COMMON N/FATS/HATP(10)
COMMON/MASS/MASS(24)
DIMENSION NME(113)
INTEGER KAK(199,2)
COMMON /CONSTS/CONSTS(10),1EOF,ACC(3)
COMMON/TUFS/TUFS(T)
COMMON/MLN7/NUM(35),LHEAD
COMMON/DADO/MINDEX(92)
COMMON/DADO/LINDEX(171),HIZ(2P),BINDEX(21)
COMMON/LD1(2P),CMB(2,2,28),CAF(28),V1Z(28),RZ(2,2,2P)
DIMENSION LIND(217),C(18),CLDAT(I0,E)
COMMON /DADO/LINDEX(171),HIZ(19)
COMMON /DADO2/LINDEX(128),HIZ2(19)
COMMON /DADO2/MSOP2,MIN(491)
FRAM DATA
DIMENSION CBA(90),CBAL(90),DIST(302)
1*KMC(140,2),COL1(30,2),INWAV(2),INFR(2),INMP(2),IDEV(2)
2*HT(2),LDFY(28),PERC(90,2)
3*LDCOL(28),LDFAY(28),ULTV2(28),LATIN(28)
INTEGER COL1,BYPASS,LAYX
COMMON TO A AND B
DIMENSION BSTIF(980),BSTIF(980),RLEN(980),RLEN(980)
1*MKB(90),CSTIF(992),CSTIF(992),CLEP(992),CAY(992),PB(992)
2*PCL(992),PMAS(540),PMAS(992),PMAS(992)
3*BR(992),IBR(992),LQY(992),ID(992)
EQUIVALENCE (BO(91),PB(91))
FRAM B ONLY
DIMENSION OR1(992),OR1(992),OVI(992),PA(992),PHIC(992)
1*PHIC1(992),TA(992),DO(992),P1(992),SPE(992),TG(1140),TH(991)
B,TRANS(991),TRANS(91),NC1(140,2)
COMMON/FRAMB/CGA,CGA1,LDSW,LDSW1,DIST,MKC,COL1,INWAV,INFR,INMB
A,INLEV,HT,LDFY,PERC,MBEX,BYPASS,INWAV,IFR,MB,MR1,LEVX
B,BSTIF,BSTIF1,BLEN1,MKB,CSTIF,CSTIF1,CLEN,CAX,PA,PC,PMAS
C,PMAS1,PMAS2,DR1,DR2,PA,PHIC,PHIC1,TA,TD,P1,SE1,TE1,TH,TRANS
D,TRANS,NC1,KPOM,KMOM,KSHR,KSHAR,DP1,KD1,KD2
E,LDSW90,IDFJ,DELTA1,LVMA,EST,LT1,LT1
FIB1,IBR1,KST(990),KST(990),M6,IDFR(140),HRJ(992),HRJ1(992)
G,SEB
H,LDOL,LDFAY,HRFL1(91),HRFL1(91),LQYES,BUF(12)
EQUIVALENCE(NIF,LSAVE)
COMMON/LEN/TBLEN,NRM,LBLAST
DIMENSION CX(992),CZ(992)
EQUIVALENCE (CX(1),PA(1))*CZ(1)*DR(1)
COMMON/PLT/FLAX,MC,MC,MC,LEVEL
COMMON/ALFA,SINA,COSA,ALFA,1D3
DIMENSION ISTAC(140)

EQUIVALENCE (UPR,PMIC(1)),(ISTAC(1),XRT(1))
IF(RUG.EQ.11)
IF(IFIGUC.EQ.53)

PRINT * ,NCP,LVNC,LV,KC,KCC
590 FORMAT(*,IQLE3,NCP,LVNC,LV,L*614)
L=JLTE(1,NCP,LVNC)
IF(L.EQ.0)GO TO 10
IF((STIF(L).LT.0.).GE.50)GO TO 10
IF((LEN(L).EQ.0.).LE.60)GO TO 10
CALL U.*-2(KST(L),AL,AL,C)

PV=COSA*FLN(L)
CV(L)=C*FLN(L)
CZ(L)=CZ(L)/LVM3
NCP=1+(L-LV)/LVM4
LVNC=L
KCC=KCC-1
IF(IFIGUC.EQ.53)
1PRINT * ,NCP,LVNC,LV,KC,KCC
591 FORMAT(*,IQLE2,NCP,LVNC,LV,L*614)
RETURN
10 CONTINUE

C ERROR MESSAGE 1: BLN DOES NOT EXIST
1IF(IFIGUC.EQ.53)
1PRINT * ,NCP,LVNC,LV,KC,KCC
592 FORMAT(*,IQLE3,NCP,LVNC,LV,L*614)
STOP
RETURN
NC PATH TO THIS STATEMENT
END

ILE MAP ----- (LDA=1) ----- ADDRESS -- BLOCK ---- -- PROPERTIES ---- TYPE ---- SIZE ---- NAME ---- ADDRESS ---- BLOCK -----

13B /CONSTS/ REAL 3 CLOAT 275P /DADAT2
2B /ALFA/ REAL CLEN1 17013P /FRAM3/
156B *S* REAL CODE 423B //
127B *S* REAL COL1 1012P /FRAM3/
140B *S* REAL CCNSTS 0P /CCNSTS
0.0B /RATS/ REAL 10 COSA 1P /ALFA/
50705R /FRAM3/ INTEGER CSTIF 13113P /FRAM3/
307B /DADAT/ REAL 21 CSTIF1 15053P /FRAM3/
5317R /FRAM3/ REAL 980 CX 37027P /FRAM3/
7243B /FRAM3/ REAL 980 CZ 31167P /FRAM3/
141R REAL DAT 55P //
9642B // REAL 20 DELTA 50702B /FRAM3/
1447R /FRAM3/ REAL 980 DIST 266B /FRAM3/
3373B /FRAM3/ REAL 980 DR 31167P /FRAM3/
6553B /FRAM3/ REAL 12 DR1 33127P /FRAM3/
1441B /FRAM3/ INTEGER DUM 00 /DRN/7
550B. /DADAT/ REAL 2B CV 35067B /FRAM3/
20753B /FRAM3/ REAL 992 E 50704B /FRAM3/
370B /DADAT/ REAL 112 EC1 115B //
38B /FRAM3/ REAL 90 EC5 116B //
132B /FRAM3/ REAL 90 EST 50706P /FRAM3/
253B /DADAT2/ REAL 16 MITZ 253B /DADAT/
SUBROUTINE STATE(2) 74/74  CFI=1

EQUIVALENT (UP,PH1C(11)),(ISTE(1),"P1(1))
IPUG="T"(11)
LDEF="X(1)+LVNC"

IF(ISTE(10),EQ,1,LANC,IFLAE,ER,1600) TO 10
IF(ISTE(LVNC),L.E.,0) GO TO 10
IF(LFAE(LVNC),L.E.,0) GO TO 10
CALL UGE(2(KST(LVNC),A1+2,P)
C=CSL(FLA(LVNC)
Cx(LY)=C(LY(LVNC)+PM

IF(IUG,FQ,53)PRINT 052,L,LANC,ALFA
C02 FORMAT(21,10,1,51 L,LANC,LANC,CE,14,F12.2)
C7(L)=C7(LVNC)
I0=1
IF(IUG,FQ,53)
1PRINT 050,IC,10,LANC,L,LVNC,ACP,LX(LVNC),C2(LVNC),C2(L)

C00 FORMAT(10,10,1,51 L,LANC,LANC,CE,14,F12.2)
ACP=1+(L-LV)/LVNC
RETURN
END

C ERROR EXIT READ DOES NOT EXIST
RETURN
END

SOFTWARE--LONG--ADDRESS--BLOCK--PROPERTY--TYPE-----SIZE--NAME--ADDRESS--BLOCK--

13P /CONSTS/
28 /ALFA/
134B *S*
175B *S*
136B *S*
08 /BATS/
5075B /FRAM/
307B /DACAT/
5317B /FRAM/
7243B /FRAM/
137B
6420 //
1447P /FRAM/
3373P /FRAM/
6553B /FRAM/
1441B /FRAM/
5508 /DACAT/
20753B /FRAM/
3708 /DACAT/
08 /FRAM/
132P /FRAM/
253P /DACAT/
2758 /DACAT/
1701P /FRAM/
423B //
1012B /FRAM/
0A /CONSTS/
18 /ALFA/

REAL 3 CSTIF 13113P /FRAMP/
REAL CSTIF 15053P /FRAM/
REAL CX 37027P /FRAM/
REAL C2 31167P /FRAM/
REAL DAY 558 //
REAL 10 DELTA 50702B /FRAM/
INTEGER 21 DIST 266B //
REAL 980 DR 31167B /FRAM/
REAL 980 DR 33127P /FRAM/
REAL 980 CUM 08 //
REAL 20 DV 350679 /FRAM/
REAL E 50704B /FRAM/
REAL 980 EC1 115P //
REAL 980 EC5 116B //
REAL 12 EST 50706B /FRAM/
INTEGER 2538 //
REAL 28 MRFL 65053B /FRAM/
REAL 992 MRFL 65206B /FRAM/
REAL 112 MRJ 61062B /FRAM/
REAL 99 MRJ1 63022B /FRAM/
REAL 90 MR 116B //
REAL 18 IBR 50711B /FRAM/
REAL 60 IBR 52635B //
REAL 992 IBUG 1338 //
REAL 2 IRIC 133B //
REAL 60 ICRI 75P //
REAL 10 IDFR 60432B //
REAL //
PROGRAM CVL53.
COMMON MTRF, FR, LEV, ISOLV, LEVIN, LVFIN, LVMAX, LHSAL, LSEL, LFLV, LFFRM
COMMON ZIIZ, INME, DAD(3), IDOC(6), IDOS(6), MAXFL, ICRISS(6), ICRC(6)
COMMON JSTOP,NSTSTOP,IPRINT,ILINK,EC1,EC5,MAXMR,ITEST,IRUN,LVU,LUU
COMMON JAX, LIST(10), JYC(10), VD(10), UF(10)
COMMON FCTC(10), USEF(10)
COMMON FON(20), LOMQ(20)
COMMON FAKE(10), JFT(10)
COMMON KEA(20), LETE(20)
DIMENSION NM(11,3)
INTEGER JAK(90,2)
COMMON FLOT1, FSTIF(980,2), FLEN(980,2), KKR(980,2), KST(980,2),
    JMBR(980,2), CSTIF(992,2), CLEK(992,2), CX(992,2), CZ(992,2),
    2INMAY(2), PRIC(992,2), DV(992,2), TRANS(992,2), C11(12), C22(12),
    3SUBK(992,2)
COMMON BATS, BATS(10)
COMMON LOTO, LOTO(10)
COMMON ALFA, ALFA, COSA, ID3
COMMON BMFL1, WIDTH(30,20), DEPTH(30,20)
COMMON BMFL2, SPAN(30,20), XM(30,20), LAM(30,20,20), JYM(30)
COMMON BMFL3, EYE(30,20), LNE(30)
COMMON CMFL1, CMFL1, FC1, FCY1
COMMON R16I, R16I(30,20)
COMMON BRACE, IBRA(30,20), LAR(5)
COMMON CLFL1, FC5, FY5, CY5, FK5(9)
COMMON CLFL2, LX1(140), IX(140), Y(140), JY(140), FCL(140,9)
COMMON CLFL3, X(140), YF1(140), Z(140), XMAX(140), YMAX(140), KSC(140),
    KC(140)
COMMON CLFL6, NQX, NQX, NEX(140), MAX(140)
COMMON CLFL5, KBAR(140,2)
COMMON CLFL7, MCD(140,3), VY(9), STUB(9), IF0(9), X(9), Y(9), INX(9)
COMMON CLFL8, TOR(140,2), TORM(140)
COMMON RIBS, RIBS(10)
COMMON MOVE, MOVNO, KMP(21,10), PM(21,10), MSTART(10,10),
    MEND(10,10), IREV(10), NW(10), PACT(10), MIT(10)
COMMON ALS, ALFA, ALFY(4)
COMMON BRC5, IBRC(10), EBERC(10)
COMMON GENI, ESTI(12,3), IPFL(30,2), KANT(30,2)
COMMON CPLR, CPLR(10)
COMMON CONTS, CONTS(10), EOF, ACC(3)
COMMON DYN, LYN(10), DYN(20)
COMMON ALFA, ALFAC(10)
COMMON BMFL1, STP, STP1, STPY, STP1, MB1(2), IF2(2), CCRD(140,2)
1: COL1, ICFL(140), IFMR, GRID(140,2), NCOL(2), KNC(140,2)
4: CLEV(99), IAU6, STZI, STZ2, IFLAG1(110), IFLAG2(140),
5: SIDE(10), IAGL, IPLOT(25,15), JPL, IGR(140,2),
6: GRID(140,2), DUMMY(240)
INTEGER COL1(140,2)
DIMENSION MB(2,1), KNC(2), XR(5), YR(5)
COMMON DATA(3), RSP2, RIND(41)
COMMON BDAT2, AIND(1171), CUP(18), CLAT(10,6)
COMMON SPLICE, MFLP(3)
COMMON REJ2, KUZ, UZ1, UZ2, UZ3, UZ4, UZ5, UZ6, UZ7, UF8, UF25, UZ10, UZ11,
    UZ12, UZ15, UZ16, UZ17, UZ18
COMMON THETFL, DUM(1890)
DIMENSION IFRA(250), YR1(250), YR1(250), MKAY(250), RMSH(21)
1: BIMSH(30,20), OSQ(250), BIM(250)
EQUIVALENCE (DUM(1),IFRB(1)),(DUM(251),XBL(1)),(DUM(501),YBL(1))
EQUIVALENCE (DUM(1001),MPM(1)),(DUM(1022),RINSH(1,1))
EQUIVALENCE (DUM(1622),OP(211)),(DUM(1872),INUMX)
EQUIVALENCE (DUMY(1),RIN(1))
REAL MARG(10)
DATA MARG/5.2,1.75,1.25,2.25,2.5/
DATA SIZE/54,44,42,34,34,17,22,11,17,8,5,11/

C
C ******** INITIALIZE PLOT
IF(LFT(12).EQ.0) GO TO 5998
IF(LFT(12).EQ.-1)GO TO 6

C
CALL PLOTLD(0,0,0,98)
CALL NEWPEN(1)

C
C ******** INITIALIZE VARIABLES

6 CONTINUE
INUE=LFT(11)

CALL ZER0(C11,4)
CALL ZERO(CLEV,90)
CALL ZER0(IFBB,1000)
ENDFILE B1
REWO B1
KPL0T=LFT(12)

IF(LFT(12).EQ.-1)GO TO 8
DO 7 J=1,KPL0T
READ(81,750,EN=7) (IFLOT(I,J),I=1,25)
7 CONTINUE
8 CONTINUE

DO 1 II=1,2
READ(81,800,END=10000) IMX,IMMX,MB1,IW,NC2(IW),IFR2(IW),INWAY(IW),
ILVM4,
MB1(IW),LD,MSW
10000 NW=MB2(IW)
NC=NC2(IW)
IFR=IFR2(IW)
MB=MB1(IW)

800 FORMAT(B24/E10)
READ(81,890,END=10001)(MKB(K,IW),K=1,NW),MKC(K,IW),K=1,NW)
1
COL(K,IW),K=1,IFR
10001 READ(81,805,END=10002)(KST(K,IW),K=1,NB)
805 FORMAT(B24/E10)

10002 READ(81,810,END=10003)(BSTMF(K,IW),BLEN(K,IW),K=1,NB),
1
(CLEN(K,IW),CSIF(K,IW),CX(K,IW),CZ(K,IW),PHIC(K,IW),DV
2(K,IW),
K=1,NC),TRANS(K,IW),K=1,ILVM4)
810 FORMAT(B24/E10)

10003 READ(81,820,END=10004)(TBR(K,IW),JBR(K,IW),K=1,NB)
820 FORMAT(B24/E10)

10004 IF(IMX.EQ.1)GO TO 2
1 CONTINUE
2 CONTINUE
IF(EOF(81).EQ.0)GO TO 9999
DO 5 II=1,IMX
MB=MB1(IW)
5 CONTINUE
DO 4 LV=2,LVM4

DO 2 I=1,MB
LV1=(1-1)*LVN+LV
IF(IPUG.EQ.53)
1PRINT 898,II,LV1,I,INX,LVN,NC,LV1,LEVEL,
IC(LV1),CZ(LV1,II)
898 FORMAT(* ** DEBU63 ** 814.2F12.2)
IF(CY(LV1,II).EQ.999.AND.CZ(LV1,II).EQ.999) GO TO 3
IF(CY(LV1,II).EQ.999.AND.CZ(LV1,II).EQ.0) GO TO 3
IF(CLEN(LV1,II).EQ.0) GO TO 3
CLEV(LV)=CZ(LV1,II)
LEVEL=LV
2 CONTINUE
3 CONTINUE

C
C ****** DETERMINE SCALE
C
IF(IPUG.EQ.53)
1PRINT 896
896 FORMAT(* ** DEBU63 ** joint No. (KA) LEVEL CX C2 *)
1CSTIF, BSTIF*)
STY=0.
IF(IBUG.NE.53)GO TO 1020
DO 1010 IW=1,INX
MB=MBI(IW)
DO 1010 KKK=1,MB
DO 1010 LV=1,LVN
LLN=(KKK-1)*LVN+LV
CALL UNPS5(KST(LLN,IW),A1,A2,A)
IF(IBUG.EQ.53)
1PRINT 895,LLN,KKK,LV,OX(LLN,IW),CZ(LLN,IW),CSTIF(LLN,IW),
1BSTIF(LLN,IW)
895 FORMAT(* ** DEBU63 ** 3I6,2X,2F8.2,2E8.2)
IF(IBUG.EQ.53)
1PRINT 881,BLE(LLN,IW),ALFA
881 FORMAT(* ** DEBU63 ** BLE N ALF )
1010 CONTINUE
1020 CONTINUE
DO 1030 JJ=1,KBLOT
IF(IPLOT(JJ).EQ.6)GO TO 1040
IF(IPLOT(JJ).LE.3)GO TO 1040
1030 CONTINUE
SCALE=20.
GO TO 1045
1040 CONTINUE
SCALE=IPLOT(JJ)
IF(SCALE.EQ.0)SCALE=20.
1045 CONTINUE
STZI=0.
STXN=STXZ=STXN=STXZ=STZM=STZM=0.
DO 20 IVI=1,INX
IFRI=IFRP(IWI)
MB=MBI(IWI)
DO 20 KI=1,IFRI
KA=COLI(NH,IWI)
IF(INN.LT.IFR1)KA=COLI(NH+1,IWI)-1
IF(INN.EQ.IFR1)KA=MB
20 CONTINUE

DO 20 IS1=KA,KP
   L=IS1+LVMA
   LL2=(IS1-1)+LVMA+1
   LL1=(IS1-1)+LVMA+(IDOS(3)+5)
   IF(CX(LL1,W1),GT,STXM)STXX=CX(LL1,W1)
   IF(CX(LL1,W1),EQ,.999,.AND.,CZ(L,W1),EQ,.999.)GO TO 20
   IF(CZ(L,W1),GT,STZ2)STZZ=CZ(L,W1)
   IF(CZ(LL2,W1),LE,STZZ1)STZZ1=CZ(LL2,W1)

20 CONTINUE
   STXN=STZ2/SCALE-MARG(4)
   STZN=STZ2/SCALE-MARG(7)
   STXM=STXN/SCALE
   STZM=STZZ/SCALE+MARG(8)
   IF(IFIPG,EQ,53)
      PRINT 875,STXX,STZ2,STZZ2,STXMN,STZMN,STXMX,STZMX,STXM
      R75 FORMAT(*) ** DEBUG LOCATION ** *,PF12.2)
      MB=MB1(1)
      MBY=MB1(2)
      MAX=MAX(*MBX,MBY)
      IF(IFIPG,EQ,53)
   IPRINT 888,MAX,MBX,MBY,MB1
   R888 FORMAT(*) *** MPI*** *,SI20)
C
C     C SET: ALL STACK COORD TO -9999.
C
C     DO 3000 IV=1,2
   IN=INVER(IN)
   MAXI=MB1(IN)
   DO 3000 N=1,MAXI
      KN=MB1(IN,W1)
      IF(KN.LE.0)GO TO 3000
      CORD(KN,IN)=-9999.
      CORD(KN,IN1)=-9999.
   3000 CONTINUE
      MBX=MB1(1)
      IFRX=IFR2(1)
      KAX=COL1(IN,1)
      IF(IFRX,GT,1)KRY=COL1(IN,1)-1
      IF(IFRX,EQ,1)KRX=MBX
      BM=0.
      IV=1
   DO 3500 K=KAX,KBX
      LVNC=K-KAX+LVMA+IDOS(3)+1
      KN=MB1(IN,1)
      CORD(KN,1)=CX(LVNC,1)
      CORD(KN,2)=BM
      IF(BLEN(LVNC,W1),LE,0.)GO TO 3500
      CALL UNP53(KST(LVNC,W1),A1,A2,B)
      BM=CORD(KN,2)+ALFA/BLEN(LVNC,W1)
      IF(IFIPG,EQ,53)
   IPRINT 887,KN,MB,Y,LVNC,CORD(KN,1),CORD(KN,2),ALFA,BM
   R887 FORMAT(*) ** DEBUG PL33 ** *,3I4,4F8.2)
   887 CONTINUE
   III=IV-1
   IV=IV+1
   IFR=IFR4(1)
   DO 3610 IZ1,IFRY
   KA=COL1(IV,2)
   3610 CONTINUE

IF(I.LT.IFRY) KB=COL1(IF1+1,2)-1
IF(I.EQ.IFRY) KB=MBY
CORDY=0.
CORDY=0.
DO 3600 K=KA,KP
MK=MKC(K1+2)
DO 3510 K=KA,KB
LVNC=(KK-1)*LVM4+(IDCS(3)+1)
MK=MKC(KK+1)
IF(MK.EQ.MK1) GO TO 3520
3510 CONTINUE
GO TO 3600
3520 CONTINUE
LVNC1=(K-1)*LVM4+(IDCS(3)+1)
DO 3580 K1=KA,KB
LVNC2=(K1-1)*LVM4+(IDOS(3)+1)
MK=MKC(K1+2)
IF(K.EQ.K1) GO TO 3570
BM=0.
IF(ABS(LVNCl+21).LE.0.0) GO TO 3540
CALL UNPS3(KST(LVNCl+2)+A1+2+B)
BM=ALFA/BLEN(LVNCl+2)
3540 CONTINUE
CORD(MK,2)=CX(LVNCl+2)+CORDY-CX(LVNCl+2)
CORD(MK,1)=CORDX-BM
LVNC1=LVNC2
3570 CONTINUE
CORDX=CORD(MK,1)
CORDY=CORD(MK,2)
3580 CONTINUE
GO TO 3610
3600 CONTINUE
3610 CONTINUE
DO 3640 IW=1,2
IW=INVFR(IW)
IFR=IFR2(IW)
DO 3630 IF=1,IFR
KA=COL1(IF1,TM)
IF(IF.LT.IF) KB=COL1(IF1+1,2)-1
IF(IF.EQ.IFR) KB=MBY
DO 3620 IK=KA,KB
MK=MKC(KK+1)
IF(CORD(MK1+1)+NEG.9999.160 TO 3630
3620 CONTINUE
3630 CONTINUE
3640 CONTINUE
DO 3680 IW=1,2
IW=INVER(IW)
IFR=IFR2(IW)
DO 3670 IF=1,IFR
KA=COL1(IF1,TM)
IF(IF.LT.IF) KB=COL1(IF1+1,2)-1
IF(IF.EQ.IFR) KB=MBY
IF(IF.EQ.IFR.IF.1+1,2) KB=MBY
W1=0
CORDX=0
WN2=0
DO 3662 IK=KA*KB
    N=KB
    MK=MKC(IK,IW)
3642 CONTINUE
    IF(IIBMGE.53) GO TO 3643
    PRINT 882 IW,IW1,IW,IF+IFR+MP1(IW),IK,KA,KB,MK,N1,CORD(MK,IW)
882 FORMAT(4** DEBUG PL1,10(I4,1F10.2)
3643 CONTINUE
    IF(CORD(MK,IW),NE.-9999.) GOTO 3645
    IF(N1.NE.0.AND.IK.EQ.KB) GO TO 3650
    IF(N1,N.0) GO TO 3660
    N1=IK
    IF(IK.EQ.KB) GO TO 3642
    GO TO 3660
3645 CONTINUE
    IF(N1.NE.0) GO TO 3652
    GO TO 3660
3650 CONTINUE
    IF(N1-1.LT.KA) GO TO 3654
    N=IK
    N1=N1-1
    MK=MKC(KA*IW)
    GO TO 3654
3652 CONTINUE
    N=IK
    LVNC=(N-1)+LVM4+(IDOS(3)+1)
    CORD1=CX(LVNC,IW)
3654 CONTINUE
    DO 3657 NN=N1,N
    IF(NN.EQ.KB) GO TO 3657
    NN1=N+NN-NN
    NN2=NN1-1
    LVNC1=(NN2-1)+LVM4+(IDOS3+1)
    IORDI=1
    IF(N1,NE.KB) GO TO 3653
    NN1=NN
    NN2=NN1+1
    LVNC1=(NN1+1)+LVM4+(IDOS3+1)
    IORDI=4
3653 CONTINUE
    MK1=MKC(NN1,IW)
    MK2=MKC(NN2,IW)
    IF(CORD(MK1,IW),EQ.-9999.,OR.,CORD(MK2,IW),EQ.-9999.) GO TO 3667
    IF(CORD1(MK1,IW),EQ.-9999.,OR.,CORD1(MK2,IW),EQ.-9999.) GO TO 3667
    GO TO 3667
3663 CONTINUE
    NN1=N+NN-NN
    NN2=NN1-1
    LVNC1=(NN2-1)+LVM4+(IDOS3+1)
    MK=MKC(NN1,IW)
    MK2=MKC(NN2,IW)
    IORDI=1
3667 CONTINUE
    IF(CORD(MK1,IW),EQ.-9999.,OR.,CORD1(MK1,IW),EQ.-9999.) GO TO 3657
    GO TO 884
884 FORMAT(4** DEBUG PL1,10(I4,1F10.2)

PROGRAM OVL57/7 74/74 OPT=1 FTN * 14552

ILVNC1*CORD1,CORD(MK,IW1),CX(ILVNC1,IW1),CORD(MK1,IW1)
RM=0.

IF(BLEN(ILVNC1,IW1),LE,0.0) GO TO 3655
CALL UNP53(KST(ILVNC1,IW1),A1,A2,P)
PM=COSA/BLEN(ILVNC1,IW1)

3655 CONTINUE
IF(CORD(MK,IW1),NE,9999.) GO TO 3656
IF(CORDI.EQ.0)
1CORD(MK,IW1)=CORD(MK1,IW1)*PM
IF(CORDI.EQ.1)
1CORD(MK,IW1)=CORD(MK1,IW1)-BM
MK2=MKC(KA,IW1)
IF(CORD(MK2,IW1),EQ,0..AND..N,NE,KP)
1CORD(MK,IW1)=CX(ILVNC1,IW1)

3656 CONTINUE
IF(CORD(MK,IW1),NE,-9999.) GO TO 3657
IQ=2
IF(IPUG,EQ,53)
1PRINT 844,1Q,IW1,IW1,IF,IK,N1,N1,N,KP,PK,MK,MK1
ILVNC1,CORD1,CORD(MK1,IW1),CX(ILVNC1,IW1),CORD(MK1,IW1)

BM=0.
IF(BLEN(ILVNC1,IW1),LE,0.0) GO TO 3658
CALL UNP53(KST(ILVNC1,IW1),A1,A2,P)
BM=ALFA/BLEN(ILVNC1,IW1)

3658 CONTINUE
IF(CORD(MK,IW1),NE,-9999.) GO TO 3659
IF(CORDI,EQ.0..AND..IW.EQ.2)
1CORD(MK,IW1)=CORD(MK1,IW1)-PM
IF(CORDI,EQ.0..AND..IW,EQ.1)
1CORD(MK,IW1)=CORD(MK1,IW1)+BM
IF(CORDI,EQ.1..AND..IW,EQ.1)
1CORD(MK,IW1)=CORD(MK1,IW1)-PM
IF(CORDI,EQ.1..AND..IW,EQ.2)
1CORD(MK1,IW1)=CORD(MK11,IW1)+BM
MK2=MKC(KA,IW1)
IF(CORD(MK2,IW1),EQ,0..AND..N,NE,KB)
1CORD(MK,IW1)=CX(ILVNC1,IW1)

3659 CONTINUE
IQ=3
IF(IPUG,EQ,53)
1PRINT 844,1Q,IW1,IW1,IF,IK,N1,N1,N,KA,KB,PK,MK,MK1
ILVNC1,BM,CORD(MK1,IW1),CX(ILVNC1,IW1),CORD(MK1,IW1),ALFA
884 FORMAT(* DEB BUG PL2 *= 13 14,5F10.2)

3657 CONTINUE
IW=0

3660 CONTINUE
3662 CONTINUE
IF(IPUG,NE,53) GO TO 3670
DO 3665 IK=KA,KB
MK=KCC(IK,IW1)

PRINT 880,IF,IW1,MK,CORD(MK,IW),CORD(MK1,IW1)
880 FORMAT(* DEB BUG PL *= 3 7 6,2F8.2)

3665 CONTINUE
3670 CONTINUE
3680 CONTINUE
WRITE(16)CORD
IF(LPTO(1).GE.1.OR.LTO(1).GE.3) GO TO 9998

3685 CONTINUE
DO 3690 J=1,KPLOT
IF(IPLT(1+JJ).EQ.6) GO TO 3694
IF(IPLT(1+JJ).LE.3) GO TO 3694.
3690 CONTINUE
SCALE=20.
GO TO 3696

3694 CONTINUE
SCALE=IPLT(20+JJ)
IF(SCALE.EQ.0.*SCALE=20.
3696 CONTINUE
STPX=STPX1=STPY=STPY1=0.
STPXMN=STPYMN=STPXMX=STPYMX=0.
DO 4000 IV=1,2
MAX=M1(IV)
DO 4000 ISI=1,MAX
MK=M1(IV,ISI)
IF(CORD(MK,1).LT.STPX)STPX=CORD(MK,1)
IF(CORD(MK,1).GT.STPX)STPX=CORD(MK,1)
IF(CORD(MK,2).LT.STPY)STPY=CORD(MK,2)
IF(CORD(MK,2).GT.STPY)STPY=CORD(MK,2)
4000 CONTINUE
STPXMN=STPY/SCALE
STPXMX=STPX/SCALE=MARG(9)
STPXMN=STPX/SCALE
STPYMN=STPY/SCALE=MARG(9)
STPIX=STPXMX-STPXMN
STPIY=STPYMN-STPYMN
STM=AMAX1(STPIX,STPIY)/SCALE
IFPL3=0
WALL=AMAT(18)
CALL GRID5(LVM4)
DO 5200 J=1,KPLOT
JPL=J
IFPL=0
IFPL1=0
IFPL2=0
IFPL4=0

C IFPL1 0 WRITES ERROR MESSAGES FOR FRAMES
C IFPL2 0 WRITES ERROR MESSAGES FOR FLOORS
C IFPL3 COUNTS NO. OF PLOTS COMMANDS

IF(IPLT(1+J).LT.2) GO TO 5110
IF(IPLT(1+J).EQ.5.OR.IPLTO(1+J).EQ.7) GO TO 5200
IF(IPLTO(1+J).EQ.6) GO TO 5110
DO 5100 II=1,140
CALL MM30J(R1(IPLT(12+J)),II,IBI,IER)
PKPLOT(II)=IBI
IF(IRI.EQ.1) IFPL=1+IFPL
5100 CONTINUE
IF(IFPL.GT.0) GO TO 5115
DO 5112 II=1,140
WPLOT(II)=11
5112 CONTINUE
5110 CONTINUE
IF(IPLOT(I,J) .EQ. 4) GO TO 5160

5115 CONTINUE
DO 5000 IW=1,IMX
IFR=IFR+(IW)
MB=MPI(IW)
DO 10 I=1,IFR
KA=COL1(I,W)
IF(N.LT.IFR)KP=COL1(N+1)*W)-1
IF(N.EQ.IFR)KP=MB
DO 10 I=KA,KP
L=LAYM
IF((XLYL+I)*EP.999..AND.(ZL+I)*EP.999..)GO TO 10
IF((XLYL+I)*ET.STX STX=XL+I)
10 CONTINUE
SCALE=IFLOT(20,J)
IF(SCALE.EQ.0.)SCALE=20.
DIRX=1
IF(INWAY(IW),EQ.2)DIRX=-1
IF(INWAY(IW),EQ.1.AND.IPFLOT(I,J),EQ.2)GO TO 5000
IF(INWAY(IW),EQ.1.AND.IPFLOT(I,J),EQ.3)GO TO 5000
IF(IBUG,EQ.53)
1PRINT 899,DIRX
899 FORMAT(* ** DEBUG FRAME IN **.A2** DIRECTION ****)
ORGX=ORGX+X
IFPL3=IFPL3+1
ICOUNT=0
MCOUNT=0
IF(IFPL.EQ.0)IFPL=IFR
IFPL4=IFPL
DO 2880 K=1,IFR
KA=COL1(K,W)
MK=MK(KA,W)
IQ=1
IF(IBUG,EQ.53)
1PRINT 873, IQ, IW, INWAY(IW), J, KPLOT, IFP, MK, IPLOT(I,J), (K, KPLOT(I,K)),
2880 IK=1,20, IFPL3,ICOUNT, MCOUNT, IFPL4
873 FORMAT(*** PLOT=1 ****, 3213)
IF(K,L.T. IFR)KB=COL1(K+1,W)-1
IF(K.EQ.IFR)KB=MB
IF(IPLOT(I,J) .EQ. 1 OR IPLOT(I,J) .EQ. 6)GO TO 2015
IF(KPLOT(MK) .EQ. 0)GO TO 2000

2885 CONTINUE
IFPL1=1
ICOUNT=ICOUNT+1
IPLA=0
ORGX=ORGX+1
IF(IFPL3.EQ.2)ORGX=ORGY1
IF(IFPL.EQ.2)IFPL=IFLOT(I,J)
IF(IPL.EQ.0)IPS=3
CALL FRAME(IPLA,ICOUNT,MCOUNT,KA,KB,IW,LAYM,ST2MN,ORGX,ORGY,
SCALE,ORGX, ORGY1, IPS)
IF(IFPL3.EQ.1)ORGY=ORGY1
IQ=3
IF(IBUG,EQ.53)PRINT 873, IQ, MK, IW, IPLOT(I,J)
IF(IBUG,EQ.53)PRINT 806, ORGX, ORGY
IF(MCOUNT.GE.999) GO TO 2^20
PRINT 877

877 FORMATE ** MESSAGE PLOT-1 *** DRAWING ELEV. NEEDS***
1* LARGER PAGE *****
CALL REMARK(1, DRAWING SIZE*)
CALL REARK(1, TOO SMALL*)
STOP

2020 CONTINUE

IF(MCOUNT.EQ.0.AND.IFPL3.EQ.1) GO TO 2023
IF(MCOUNT.EQ.0) GO TO 2022
ORGX = 0
ORY = 0

2022 CONTINUE

IF(IFPL1.GE.2.AND.MCOUNT.EQ.0) ORGX = 0.
IF(IFPL1.GE.2.AND.MCOUNT.EQ.0) ORGY = 0.
IF(IFPL4.EQ.5) PRINT 806, ORGX, ORGY1

806 FORMATE ** LOCATION 3x3F8.2**
IF(ICOUNT.GT.1.AND.MCOUNT.EQ.0) GO TO 2021
CALL PLOT(ORGX3, ORGY3 + 3)
CALL PLOT(-1.0, -3)

2023 CONTINUE

IF(MCOUNT.EQ.0.AND.ICOUNT.GT.1) GO TO 2021
SIDEX = SIDE(IPS + 1)
SIDEY = SIDE(IPS)
CALL NEWPEN(11)
CALL ROPHER(-8, -3,0.0, STDX, STDFX, STFY, -3, -3, 03)

CALL NEWPEN(1)
CALL PLOT(-5, -5, 3)
XARR(1) = -.5
XARR(2) = -.5
YARR(1) = -.5

YARR(2) = STDFX + 5
YARR(3) = STDFX + 5
YARR(3) = YARR(2)
XARR(4) = XARR(3)
YARR(4) = YARR(3)
XARR(5) = XARR(4)
YARR(5) = YARR(4)

CALL DASH(XARR,YARR,5,1,1,5)
IFPL4 = IFPL4 - 1
IFPL5 = IFPL4 - IFPL4
IS = 1
IM = 2
IF(IPLOT(2^2, 1).EQ.1) IM = 3
IF(IPLOT(2^2, 1).EQ.1) IM = 2
CALL LOGO(SH, IS, LW, IFPL5, IPS, ORGX, ORGY, SCALE)

2021 CONTINUE

IF(LOCQ.EQ.53) PRINT 807, ICOUNT, MCOUNT, K, ORGX1, ORGY1

807 FORMAT ** (LOCATION 2 *** 3I4, 4F8.2)
CALL PLOT(ORGX, ORGY - 3)

IF(MCOUNT.GE.998)
  CALL FRAME(IFPLA, MCOUNT, KA, KB, TW, LWM, STZMN, ORGX, ORGY
  * SCALE, ORGX1, ORGY1, IPS)
  CALL PLOT(STXMN, STZMN + 3)
  CALL SYMBOL(STXMN, STZMN + 01, 3, 0, 0, -1)
  CALL PLOT(STXHN, STZMN + 3)
  CALL SYMBOL(STXHN, STZMN + 01, 3, 0, 0, -1)
IF(IPUG.EQ.53)
IPRINT 990

990 FORMAT(*) ** DEBUG ** TOTAL NC, FR FRAME NO. STK(ND)**
1* STK(MB) STK(MB) MAXLEV1 MAXLEV2*)
IF(IBUG.EQ.53)
IPRINT 991, IFR, K, KA, KB, MB, LVMA, LEVEL

491 FORMAT(*) ** DEBUG**, 15, 9, X, 14, 7, X, 13, 5, X, 13, 4, X, 16, 7, X, 14, 6, X, I3,
CALL NEWPEN(I1)

DO 150 I=K,KP
    LN1=(I-1)*LVMA+1
    LN2=I*LVMA
    IF(IPUG.EQ.53)
IPRINT 992

592 FORMAT(*) ** DEBUG ** STAC N2, LAST JOINT FIRST JOINT COOR.**
1* JOINT NO. CX CZ CLEN1 (*)
IF(IPUG.EQ.53)
IPRINT 993, I, LN2, LN1, LN2, CX(LN2, IW), CZ(LN2, IW), CLEN1(LN2, IW)

993 FORMAT(*) ** DEBUG ** I5, 5, X, 15, 8, X, 16, 7, X, 17, 4, 12, 2, 5, F, 2,
50 IF(LN2.LT.LN1) GO TO 900
IITYPE=3

C
C

IF(CX(LN2, IW).EQ.999.AND.CZ(LN2, IW).EQ.999.)GO TO 160
IF(CLEN1(LN2, IW).EQ.0.)GO TO 160
BM=R

C

COL=0.
CX1=CX(LN2, IW)
CZ1=CZ(LN2, IW)
C1111=COL/SCALE
C2211=CZ1/SCALE
IF(I.EQ.KB160 TO 70

IF(BLEN(LN2, IW).EQ.0.)GO TO 70
CALL UNP53(KST(LN2, IW), A1, A2, B)
BM=1.0-ALFA/ALFA**.5/BLEN(LN2, IW)

70 COL=1./CLEN1(LN2, IW)
CALL PLOT(CX1/SCALE, CZ1/SCALE, 3)
CXX=CX1+BM
CZZ=CZ1+COL
C1112=CX1/SCALE
C2222=CZ2/SCALE
IF(CSTIF(LN2, IW).LE.1.E-10.OR.COL.EQ.9) GO TO 100
CALL PLOT(CX1/SCALE, CZ2/SCALE, 2)
IITYPE=4

9 IITYPE=4NLINE

IF(IBUG.EQ.53)
IPRINT 897, LN2, IW, CX1, CZ1, CX2, CZ2
897 FORMAT(*) ** DEBUG ** 314, 4, 12, 2,
60 TO 120

100 CONTINUE

IF(IPLT(O1).EQ.0) GO TO 120
CALL DASH(C11, C22, 2, 1, 2)
CALL PLOT(CX1/SCALE, CZ1/SCALE, 3)
IITYPE=4NDASH

120 CONTINUE

IF(COL.EQ..9) GO TO 900
60 TO 160

160 CONTINUE
IF (I, EQ, K, AND, IBUG, EQ, 5) PRINT 994
994 FORMAT (3, ** DECL, COLS ** FROM JOINT), CX, CZ, TC, JOINT
IF (IBUG, EQ, 53)
1 PRINT 995, LN2, CX(LN2, IW), CZ(LN2, IW), (LN2-1), CX1, CZ2
1 CLEMNT(LN2, IW), I I I TYP E
095 FORMAT (3, ** DECL, COLS ** 14, S, 9.2, S, 9.2, 8, X, 14, S, 9.2, S, 9.2, S, 14)
1 (4Y, 4X))
LN2=LN2-1
GO TO 990
990 CONTINUE
LN2=LN2+1
IF (IPLN1(T, 9, J), EQ, 0, AND, LSTIF, (LN2, IW), LE, 1, E-10) GO TO 1000
IF (LN2, LE, LN1) GO TO 1000
C1=CX(LN2, IW)+5/SCALE
C2=CZ(LN2, IW)/SCALE
C3=C1+1./SCALE
CALL PLS4T(C1, C2, 3)
CALL PLS4T(C3, C2, 2)
IF (C3, LT, 0.9) LN2=LN2-1
IF (C1, EQ, 0.9, AND, LN2, GE, LN1) GO TO 50
1000 CONTINUE
DO 1200 LV2=2, LEVEL
LN2=(KA+1)/LvMA+LV2
LN3=(KB-1)/LvMA+LV2
IF (IBUG, EQ, 53)
1 PRINT 890
890 FORMAT (1, ** DECL, ** LEVEL NO., LAST JOINT FIRST JOINT COOR.**, 1, JOINT NO., CX, CZ, BLFN *)
IF (IBUG, EQ, 53)
1 PRINT 891, LV2, LN3, LN4, LN5, CX(LN2, IW), CZ(LN2, IW), BLFN(LN2, IW)
891 FORMAT (4H)
1050 IF (LN2, GE, LN3) GO TO 1200
I I I TYP E=0H
LN4=LN2+LvMA
IF (CX(LN2, IW), EQ, 999, AND, CZ(LN2, IW), EQ, 999) GO TO 1160
IF (BLFN(LN2, IW), EQ, 0, ) GO TO 1060
IF (CX(LN4, IW), EQ, 999, AND, CZ(LN4, IW), EQ, 999) GO TO 1160
1060 CONTINUE
CX1=CX(LN2, IW)/SCALE
CZ1=CZ(LN2, IW)/SCALE
C11(1)=CX1
C22(1)=CZ1
BM=A0.
GAP1=0.
GAP2=0.
IF (BLFN(LN2, IW), EQ, 8.8, ) GO TO 1070
CALL UNP53(KST(LN2, IW), A1, A2, A3)
BR=16, ALFA, ALFA, ** 15/BLFN(LN2, IW)
K1=0
K2=0
IF (IPLN1(T, 5, J), EQ, 0, ) GO TO 1070
IF (A1, LE, 0.05) K1=1
IF (A2, LE, 0.05) K2=1
IF (K1, EQ, 0.6, GAP1=0.03
IF (K2, EQ, 0.6, GAP2=0.03
CX1=CX1+GAP1
1070 CALL PLOT(CX1,CZ1,3)
   IF(IRUG.EQ.53)

   874 PRINT 874,J,LG2,K1,K2,IPLAN(5,J),GAP1,GAP2,CX1,CX2
   CK2=CX1+RI/SCALE-(GAP1+GAP2)
   C22=CZ1
   IF(MAT.EQ.1160 TO 1170
   IF(RSTIF(LN2,IW).LE.1.E-15)GO TO 1150

   C DRAW A REAL BEAM
   C
   IF(RSTIF(LN2,IW).EQ.WALL)GO TO 1150
   IITYPE=4
   CALL PLOT(CX2,CZ2,2)
   GO TO 1160

   C DRAW A DUMMY BEAM OR A WALL
   C
   1150 CONTINUE
   C11(2)=CX2
   C22(2)=CZ1
   IF(IPLAN(9,J).EQ.0 .AND.RSTIF(LN2,IW).LE.1.E-15)GO TO 1160
   CALL DASH(C11,C22,2,1,2)
   IITYPE=4
   C11(2)=CX2
   C22(2)=CZ1
   IF(IPLAN(9,J).EQ.0 .AND.RSTIF(LN2,IW).LE.1.E-15)GO TO 1160
   CALL DASH(C11,C22,2,1,2)

   1160 CONTINUE
   IF(LV2.EQ.2 .AND.RUG.EQ.53)PRINT 892

   892 FORMAT(20 ** DEBUG BEAM ** FROM JOINT C
   CX1 CZ1 TO JOINT C
   CX2 CZ2 BLEN BNTYP*)
   LN4=LN2+LVMA
   IF(RUG.EQ.53)
   *PRINT 893,LN2,CX(LN2,IW),CZ(LN2,IW),LN4,CX(LN4,IW),CZ(LN4,IW),
   LN2=LN4
   GO TO 1170
   1170 CONTINUE

   C DRAW A REAL BEAM
   C
   IF(RSTIF(LN2,IW).EQ.WALL)GO TO 1180
   IF(IW.EQ.1)ITEM=MOD(MKB(LN2,IW),10000)
   IF(IW.EQ.2)ITEM=MKB(LN2,IW)/10000
   NL=ITEM/100
   NSP=ITEM-NLI/100
   IF(WIDTH(NLI+NSP).EQ.4 .AND.DEPTH(NLI,NSP).EQ.4.)GO TO 1180
   IITYPE=4
   CALL PLOT(CX2,CZ2,2)
   GO TO 1182

   C DRAW A DUMMY BEAM OR A WALL
   C
   1180 CONTINUE
   C11(2)=CX2
   C22(2)=CZ1
   IF(IPLAN(9,J).EQ.0 .AND.RSTIF(LN2,IW).LE.1.E-15)GO TO 1182
   CALL DASH(C11,C22,2,1,2)
II TYPE=** HDASH**

1182 CONTINUE

IF X.Y.EQ.2 .AND. IBUG.EQ.53) PRINT 892
LNY=LNN2+LVNM
IF (MIN.EQ.0.53)
*PRINT 892, LNY, CX(LNN2+IW), CZ(LNN2+IW), LNY, CX(LNN4+IW), CZ(LNN4+IW) *
* BM**, II TYPE **
LNN2=LNN4
GO TO 150

1200 CONTINUE

*************** BRACING PLOT ****************

IF (FSW.EQ.0.1) GO TO 520

UNPACK BRACING INFORMATION

IF (IFUG.EQ.53)
PRINT 996

996 FORMAT (** DEEUG BRACING INFORMATION**)
KB1=KB-1
DO 490 I=KA,KB1
DO 490 LV=2*LVNM
LNN2=(I-1)*LVNM+LV
IF (TPR(LNN2+IW).LT.1.0) GO TO 380
JR=JBR(LNN2+IW)/(10.05)
JL=JBL(LNN2+IW)-JR*10.005
IC=MOD(IBR(LNN2+IW)/10.10)
DO 500 IT=1,KB
LNM=LVM4+IT-LNN2
IF (CX(LNM+IW).EQ.999 .AND. CZ(LNM+IW).EQ.999.) GO TO 300

C1X=CN(LNM+IW)/SCALE
C1Z=CN(LNM+IW)/SCALE
GO TO 310

300 CONTINUE

310 CONTINUE

GO TO (320,330,320,340,350), IC

CALL GOTOER

( RBR TYPE 1 AND RIGHT BRACE OF TYPE 3(XBR) )

320 CONTINUE

C2X=CN(JR+IW)/SCALE
C2Z=CN(JR+IW)/SCALE
IF (BSTIF(LNM+IW).EQ.WALL .AND. IC.EQ.3) GO TO 322
CALL PLOT(C1X,C1Z,3)
CALL PLOT(C2X,C2Z,2)
IF (IC.EQ.1) GO TO 380
GO TO 330

322 CONTINUE

C11(1)=C1X
C11(2)=C2X
C22(1)=C1Z
C22(2)=C2Z
CALL DASH(C11, C22, 1, 2)

330 CONTINUE

330 CONTINUE
C LBR TYPE 2 AND LEFT BRACE OF TYPE 3 (LBR)
C
C1X=CX(LN2+IW)/SCALE
C1Z=CZ(LN2+IW)/SCALE
C2X=CX(JL+IW)/SCALE
C2Z=CZ(JL+IW)/SCALE
IFIRST==(LN2+IW).EQ.WALL.AND.1C.EQ.3) G0 TO 332
CALL PL*T(C1X+C1Z,3)
CALL PL*T(C2Y+C2Z,2)
GO TO 390
332 CONTINUE
C11(1)=C1X
C11(2)=C2X
C22(1)=C1Z
C22(2)=C2Z
CALL DASH(C11+C22,2,1.0)
GO TO 390
340 CONTINUE
C APR TYPE 4
C
C1=(C1X*SCALE+CX(LN2+IW))/SCALE
C1Z=CZ(LN2+IW)/SCALE
C2X=CX(JL+IW)/SCALE
C2Z=CZ(JL+IW)/SCALE
C3X=CX(JR+IW)/SCALE
C3Z=CZ(JR+IW)/SCALE
CALL PLOT(C1+C1Z,3)
CALL PLEC(C2X+C2Z,2)
CALL PLET(C1+C17,3)
CALL PLOT(C3X+C3Z,2)
GO TO 380
350 CONTINUE
C VRR TYPE 5
C
C1=(CX(JL+IW)+CX(JR+IW))*5/SCALE
C2=CZ(JL+IW)/SCALE
C2X=CX(LN2+IW)/SCALE
C2Z=CZ(LN2+IW)/SCALE
CALL PLOT(C1+C1Z,3)
CALL PLET(C1+C1Z,3)
CALL PLOT(C2X+C2Z,2)
CALL PLOT(C3X+C3Z,2)
GO TO 380
380 CONTINUE
IF(ISLY.EQ.2.AND.IBUG.EQ.53)PRINT 997
997 FORMA** ** DEBUG ** TYPE LN1 CX CZ JR *
1** CX CZ *)
IF(ISBUG.NE.53)GO TO 383
IF(IC.EQ.1 OR IC.EQ.3)PRINT 998;IC,LN1,CX(LN1+IW),CZ(LN1+IW),JR*
1CX(JR+IW),CZ(JR+IW)
998 FORMA** ** DEBUG ** 9.723A5,18,14.2F8,2,23,14.2F8,2)
383 CONTINUE
IF(ISBUG.EQ.53)GO TO 385
IF(IC.EQ.2 OR IC.EQ.3)PRINT 998;IC,JL,CX(JL+IW),CZ(JL+IW),LN2*
1CX(LN2+IW),CZ(LN2+IW)

385 CONTINUE
IF (IC.EQ.4 .AND. IVUG.EQ.53) PRINT 998, IC, JL, CX(JL, IW), CZ(JL, IW), JR,
1 CX(JF+IW), CZ(JF+IW)
IF (IC.EQ.5 .AND. IVUG.EQ.53) PRINT 998, IC, JL, CX(JL, IW), CZ(JL, IW), LN2,
1 CZ(LN2+IW), CZ(LN2+IW)
390 CONTINUE
400 CONTINUE
C20 KB1=KB-1
C
C WRITE FEAM MARKS
C
CALL NEWPEN(1)
CALL ZEO(DUM, 1999)
DO 500 LV=Z+LVMA
LINE=0
DO 550 I=KA, KB1
LN2=I-I+1+LVMA-LV
IF (IC(LV2, IW).EQ.599 .AND. CZ(LN2+IW).EQ.999) GO TO 550
IF (BLEN(LN2+IW).LE.0) GO TO 550
CX=NX(LN2+IW)/SCALE
C17=CN(LN2+IW)/SCALE
DO 530 II=1, KB
LN1=II+LVMA-LN2
IF (IC(LV1, IW).EQ.999 .AND. CZ(LN1+IW).EQ.999) GO TO 530
C2X=NX(LN1+IW)/SCALE
C2Z=CN(LN1+IW)/SCALE
530 CONTINUE
540 CONTINUE
C1=(C1X+C2X)*.5
C2=C1Z
IF (IV.EQ.1) MARK=MOD(MARK, LN2+IW), 10000
IF (IV.EQ.2) MARK=MARK, LN2+IW), 10000
MNI=MARK/100
NSP=MARK-MNI*100
IF (IPL0T(18, J).EQ.0 .AND. IPL0T(19, J).EQ.0) GO TO 535
IF (MN0D(LV).LE.0) GO TO 535
CALL STINDX(12, LN0D, 91, 0)
CALL READMS(12, LN0D, 171, LV-1)
CALL STINDX(12, LN0D, 171, 0)
IF (L0D(MNI+1).LE.0) GO TO 535
IF (MNI.EQ.LINE) GO TO 541
CALL READMS(12, BMSP, 21, NLI)
LINE=MNI
541 CONTINUE
BMSP(MNI, NSP)=BMSP(NSP)
535 CONTINUE
K1=K2=0
CALL UNP53(KST(LN2+IW), A1, A2, B1)
IF (A1.EQ.05) K1=1
IF (A2.EQ.05) K2=1
IF (FMAT.EQ.1) GO TO 548
IF (IPL0T(18, J).EQ.0) GO TO 542
CALL S1ZE(S1, J, IW, A1, C2, LN2, SCALE)
542 CONTINUE
IF (IPL0T(18, J).EQ.0) AND STIF(LN2+IW), J, LE=15460 TO 550
IF (IPL0T(5, J).EQ.0) GO TO 545
IF(K1, EQ, 0) GO TO 545
CALL SYM5(1, K1, SCALE, C1, C2, L1, L2, IV)

545 CONTINUE
IF(IPT(I14, J) .EQ. 0) GO TO 547
CALL LABEL5(I1, J, C1, C2, KA, KB, LVMA, IW, LN, SCALE)

547 CONTINUE
IF(IPT(I5, J) .EQ. 0) GO TO 550
IF(K2, EQ, 0) GO TO 550
CALL SYM5(2, K2, SCALE, C2, C2, L2, L2, IV)
GO TO 550

550 CONTINUE
C MARKS AND SYMBOLS FOR BEAMS (CONCRETE STRUCTURES)

WID=WIDTH(NLI, N5P)
DEP=DEPTH(NLI, N5P)
IF(MAT, EQ, 1) GO TO 546
IF(IPT(I18, J) .EQ. 0) GO TO 546
IF(I, ET, KA) GO TO 543
CALL ENCO(30)

543 CONTINUE
CALL SIZE5(4, J, IV, C1, C2, L2, SCALE)

546 CONTINUE
IF(IPT(I9, J) .EQ. 1) GO TO 548
IF(WID, EQ, 4, AND: DEP, EQ, 4) GO TO 550

549 CONTINUE
IF(IPT(I5, J) .EQ. 0) GO TO 552
IF(K1, EQ, 0) GO TO 552
IF(MAT, EQ, 1) GO TO 552
CALL SYM5(1, K1, SCALE, C1, C2, LK, L2, IV)

552 CONTINUE
IF(IPT(I14, J) .EQ. 0) GO TO 554
CALL LABEL5(I1, J, C1, C2, KA, KB, LVMA, IW, LN, SCALE)

554 CONTINUE
IF(IPT(I5, J) .EQ. 0) GO TO 550
IF(K2, EQ, 0) GO TO 550
IF(MAT, EQ, 1) GO TO 550
CALL SYM5(2, K2, SCALE, C2, C2, L2, L2, IV)

550 CONTINUE
580 CONTINUE
IF(MAT, EQ, 1) GO TO 590
IF(IPT(I17, J) .EQ. 0) GO TO 590
CA=KA
C1=KB
CALL SIZE5(3, J, IV, CA, C1, CB, LVMA, SCALE)

590 CONTINUE
IF(IPT(I8, J) .EQ. 0) GO TO 595
CALL DH5(I1, J, KA, KB, IW, LVMA, L, SCALE, STZMK)

595 CONTINUE
C LEVEL NUMBERS AND COLUMN MARKS
CALL LABEL5(2, J, C1, C2, KA, KB, LVMA, IV, LN, SCALE)

2000 CONTINUE
IF(IPT(I, EQ, 0) GO TO 4305
CALL PLOT140(6, 1, ORG1, -3)

5000 CONTINUE

5160 CONTINUE
IF(IPUG, EQ, 53)

PRINT ".", IPILOT(1, J)

802 FORMAT(* DEBUG PLOT *** , T4)
IF(IPILOT(1, J), EQ, 4) GO TO 4012
IF(IPILOT(1, J), EQ, 6) GO TO 4012
GO TO 5?P0

4012 CONTINUE

SCALE = IPILOT(2, J)

TF = SCALE / EQ, 0, ) SCALE = 20.
CALL ZER0(IFL, 140)
DR8 = COR8 = 0.
COUNT = 0
MCOUNT = 0
IFPL = IFPL + 1

REWIND LVU
IF(IFPL3, EQ, 1) IFPL4 = IFPL
DO = 300 LV5 = 2, LEVEL
CALL .ZER0(IFLAG1, 280)
LV2 = LEVEL - LV5 + 2
TF(IFBG, EQ, 53, AND, LV5, EQ, 2) PRINT BAS

685 FORMAT(* *** DEBUG PL3 *** LEV LV M X/4 FRINT FR, 0. STNB.***
1* STL SL2 MRC LVM FROM TO BM CX *)
IF(IFPL, EQ, 0) GO TO 4015
IQ = 2

IF(IPUG, EQ, 53)

APRINT A3, IQ, T1, INWAY(T1), T1, KPILOT, IFPL, KA, IPILOT(1, J), (KKPLOT(IK)).
2IK = 1, 20)

4015 CONTINUE

IF(ACC(3), EQ, 2) GO TO 4011
WRITE(16) IPILOT, JPL, MB1, IFR2, COL1, ICFL, IFRM,
1MCOL, MRC, IBUG, IFLAG1, IFLAG, KKPLOT, IGRID1, IFPB, MKB1, INUM,
2STPX, STPX1, STPY, STPY1, CORD, GRID1, CLEV
1STZ2, STZ2, SIDE, MARG, DUMMY, XB1, YB1, BM8, BM8, OR2, PIM

4011 CONTINUE

IFPL2 = 1
CALL R53(1, LVU, LIG, JIG, NIG, LVR)
TF(ACC(3), EQ, 2) GO TO 5325
BACKSPACE (16)

READ(16) IPILOT, JPL, MB1, IFR2, COL1, ICFL, IFRM,
1MCOL, MRC, IBUG, IFLAG1, IFLAG, KKPLOT, IGRID1, IFPB, MKB1, INUM,
2STPX, STPX1, STPY, STPY1, CORD, GRID1, CLEV
1STZ2, STZ2, SIDE, MARG, DUMMY, XB1, YB1, BM8, BM8, OR2, PIM
DO 865 IN1 = 1, 2
INCOL = INCOL(IN)

DO 865 INN1 = 1, NCOL1
IF(GRID1(INN1, IN), GE, 777777) GRID1(INN1, IN) = 777777.777

865 CONTINUE

5325 CONTINUE

IFMAT(, EQ, 1160) GO TO 4014
IF(IPILOT(1, J), EQ, 8, AND, IPILOT(19, J), EQ, 0) GO TO 4014
CALL STINDEX (12, MINO2, 910)
IF(FIND2(LVR + 1), LE, 0) GO TO 4014
CALL READSM(12, LIND2, 171, LVR)
CALL STINDEX (12, LIND2, 171, 0)
IF(FIND2(LMM + 1), LE, 0) GO TO 4014
DO 5326 LMM = 1, LG

CALL READSM(12, BM8, 21, LMM)
DO 5250 JMM=1,J6G
PINFH(LMM,JMM)=PNSH(JMM)
IF(ITUG.EQ.53)
1PRINT 5330,LMM,JMM,LNO2(LMM+1),PNSH(LMM,JMM)
5330 FORMAT(*,POSITIC*,2010,125,1X,A15)
5250 CONTINUE
5300 CONTINUE

4014 CONTINUE

DO 4013 I=1,2
MB=MB1(I)
IFR=IFR2(I)
IWR=INVER(I)
DO 4013 I=1,IFR
IF(COL(I).LT.1)
4013 K=KCOL(I)

IF(I.EQ.IFR)KB=COL(I+1-I)
IF(I.EQ.IFR)KB=MB
DO 4013 K=KB,KB
MK=MKC(K+I)
IF(KSC(KW).EQ.0)GO TO 4013
ICFL(MK)=1

4013 CONTINUE

CALL ZERO(IFBR*1000)
IF(KKPLTF(LVR).EQ.0.AND.IPLTF(J)).NE.6)GO TO 4300
IO=3
IF(IFUG.EQ.53)
*PRINT 878,IO,J1,J2,J3,LEVEL,LV2,LVR
ICOUNT=ICOUNT+1
IFLA=1
ORX3=ORGX1
IF(IFPL3.EQ.2)ORY3=ORGY1
IF(IPS.EQ.0)IPS=IPLTF(21,J)
IF(IPS.EQ.0)IPS=3
CALL FRAME(IFLA,ICOUNT,KCOUNT,LV2,KB,IW,LVM4,STPYMN,ORGX,ORY,
SCALE=ORX1,ORY1,IPS)
IF(IFPL3.EQ.1)ORY3=ORGY1
IF(MCOUNT.NE.999)GO TO 4016
PRINT 878

A78 FORMAT(*,*,MESSAGE PLOT-2,*** FRAMING PLAN NEEDS***)
1* LARGER PAGE ******
CALL REMARK(* DRAWING SIZE*)
CALL REMARK(* TOO SMALL*)
STOP

4016 CONTINUE

IF(MCOUNT.EQ.0.AND.IFPL3.EQ.1)GO TO 4017
IF(MCOUNT.EQ.0)GO TO 4022
ORX3=ORGX1
ORY3=ORGY1

4022 CONTINUE

IF(IFPL3.EQ.2.AND.MCOUNT.EQ.0)ORX3=0.
IF(IFPL3.EQ.2.AND.MCOUNT.EQ.0)ORY3=0.

IF(MCOUNT.EQ.0.AND.MCOUNT.EQ.0)GO TO 4019
CALL PLOT(ORX3,ORY3,5)
CALL PLOT(1,0,5,-3)

4017 CONTINUE

IF(MCOUNT.EQ.0.AND.ICOUNT.EQ.1)GO TO 4019
SIDEY=SIDE(IPS+1)
SIDEY=SIDE(IPS)
CALL NEWPEN(11)
CALL BORDER(IDX2*SIDE*SIDF*4*2)*0*3)
CALL NEWPEN(1)
CALL Plot(1,5,ORYG+3)
XARR(1)=-6
YARR(2)=-6
YARR(1)=-6
YARR(2)=SIDE(IPS+)*5
XARR(1)=SIDE(IPS+1)*4
YARR(3)=YARR(2)
XARR(4)=XARR(3)
YARR(4)=YARR(1)
XARR(5)=XARR(1)
YARR(5)=YARR(1)
CALL DASH(XARR,YARR,5,1,5)
IF(NEG.EQ.5) PRINT 807,ICOUNT,MCOUNT,K,ORYG1,ORYG1
IFPL4=IFPL4+1
IFFL5=IFFL5-IFPL4
IS=2
IM=2
IF(IPLT(22,J).EQ.8)IM=3
IF(IPLT(22,J).EQ.7)IM=1
CALL LOGO51(IS,LVR+IW+IFPL5,IPS,ORYG,ORYG,SCALE)
4010 CONTINUE
CALL Plot(ORYG,ORYG,-3)
IF(NCOUNT.EQ.99)
CALL FRAME(IFPL1,MCOUNT,MCOUNT,LV2,KP,LVM4,STPMX,ORYG,ORYG,
SCALE,ORYG1,ORYG1,IPS)
CALL Plot(STPMX,STPMX,3)
CALL SYMBOL(STPMX,STPMX,0,1,3,0,0,1)
CALL Plot(STPMX,STPMX,3)
CALL SYMBOL(STPMX,STPMX,0,1,3,0,0,1)
C
C IF(LVR+4E<(LV2-1)) STOP
CALL NEWPEN(11)
IF(NMAT.EQ.1)GO TO 4295
DO 4250 I=1,2
MB=MB+1(IW)
IFR=IFR+1(IW)
IW=INVER(IW)
DO 4250 I=1,IFR
KA=COLI(I,1)
IF(I.I.LT.IFR)KA=COLI(I+1,1)
IF(I.I.EQ.IFR)MB=MB
DO 4250 K=KA,KB
MK=MKC(K,IW)
LVM=LVM4+LV2
C1=CORD(MK,IW)/SCALE
C2=CORD(MK,IM)/SCALE
IF(LVMEQ.2)C1=C2
IF(LVM.EQ.4)C1=CORD(MK,1)/SCALE
C1111=C1
C2211=C2
IF(KSCMK.EQ.9)GO TO 4820
IF(KSCMK.EQ.8)GO TO 4820
CALL Plot(C1+C2,3)
CALL SYM85(K1, SCALE, C1, C2, K1, IW)

4020 CONTINUE

4025 CONTINUE
CALL PLNT(C1, C2, 3)
LVNC=(K-1)*LVMA+LV2
IF(K.EQ.K1)GO TO 4200
IF(CX(LVNC+IW), EQ.999.. AND CZ(LVNC+IW), EQ.999.)GO TO 4200
PM=0
IF(BLEN(LVNC+IW), LE.0)GO TO 4100
PM=1./BLEN(LVNC+IW)

4100 CONTINUE
CALL UNP53(KST, LVNC, IW), A1, A2, P
K1=K2=0
IF(A1.GT.05)K1=1
IF(A2.GT.05)K2=1
GAP1=GAP2=0
MK1=MKC(K+1, IW)
ALC=ALFA(MK)
ALC1=ALFA(MK1)
COSAL=COS(ALC)
SINAL=SIN(ALC)
COSAL1=COS(ALC1)
SINAL1=SIN(ALC1)
GAP=FCL(MK, 4)*1.5/(24.*SCALE)
GAPP=FCL(MK1, 4)*1.5/(24.*SCALE)
IF(PM, EQ.2)GO TO 4102
GAP3=GAP4=0.
IF(KSC(MK), EQ.12)GO TO 4103
IF(ALC, EQ.0)GO TO 4101
IF(ALFA, NE.0)GO TO 4104

4101 CONTINUE
IF(APS(ALFA, SINAL), GT., 6)GO TO 4104
GAP3=GAP4=COSAL
GAP3=GAP4=SINAL
GO TO 4104

4103 CONTINUE
IF(ALC, EQ.0)GO TO 4104
IF(ALC, LT.0)GO TO 4105
IF(ABS(SINAL-COSAL), GT., 05)GO TO 4104
GAP3=GAP=SINAL
GAP4=GAP=COSAL
GO TO 4104

4105 CONTINUE
IF(ALFA, LE.0)GO TO 4104
IF(ABS(COSAL+ALFA), GT., 05)GO TO 4104
GAP3=GAP=COSAL
GAP4=GAP=SINAL
GO TO 4104

4102 CONTINUE
GAP3=GAP4=0.

C Y DIRECTION

IF(KSC(MK), EQ.12)GO TO 4109
IF(ALC, EQ.0)GO TO 4104
IF(ALC, LT.0)GO TO 4107
IF(ABS(COSAL+ALFA), GT., 05)GO TO 4104
GAP3 = GAP * COSAL
GAP4 = GAP * SINAL
GO TO 4104

4107 CONTINUE
IF (GAP (< COSAL - ALPHA) .GT. 0) GO TO 4104
GAP3 = GAP * COSAL
GAP4 = GAP * SINAL
GO TO 4104

4108 CONTINUE
IF (ALPHA .LE. 0.160) GO TO 4110
IF (ALPHA .LE. 0.160) GO TO 4104

4110 CONTINUE
IF (GAP (< ALFA - SINAL) .GT. 0.160) GO TO 4104
GAP3 = GAP * SINAL
GAP4 = GAP * COSAL
IF (ALPHA .LE. 0.160) GAP3 = -GAP3
IF (ALPHA .LE. 0.160) GAP4 = -GAP4

4104 CONTINUE
IF (K1 .EQ. 0) GAP1 = 0.853
IF (K2 .EQ. 0) GAP2 = 0.853
C1C = C1 + GAP1 * COSA * GAP3
C2C = C2 + GAP1 * ALFA * GAP4
IF (IWT .EQ. 1) GO TO 4186
C1C = C1 + GAP1 * ALFA * GAP3
C2C = C2 + GAP1 * COSA * GAP4

4106 CONTINUE
CALL PL(T(C1C, C2C, 3))
IF (IWT .EQ. 2) GO TO 4120
GAP3 = GAP4 = 0.
IF (KSC (MK1) .EQ. 12) GO TO 4112
IF (ALFA .LT. 0.160) GO TO 4111
IF (ALFA .LT. 0.160) GO TO 4108

4111 CONTINUE
IF (GAP (< ALFA - SINAL) .GT. 0) GO TO 4108
GAP3 = -GAP * COSAL
GAP4 = -GAP * SINAL
GO TO 4108

4112 CONTINUE
IF (ALPHA .LE. 0.160) GO TO 4108
IF (ALPHA .LE. 0.160) GO TO 4114
IF (ABS(SINAL - COSA) .GT. 0.05) GO TO 4109
GAP3 = GAP * SINAL
GAP4 = GAP * COSAL
GO TO 4108

4114 CONTINUE
IF (ALPHA .LE. 0.160) GO TO 4108
IF (ABS(COSA + SINAL) .GT. 0.05) GO TO 4108
GAP3 = GAP * SINAL
GAP4 = GAP * COSAL

4108 CONTINUE
C1C = 1 + COSA / SCALE - GAP2 + COSA * GAP3
C2C = 1 + ALFA / SCALE - GAP2 + ALFA * GAP4 + FCL (MK1, 1W1)*1.5/(12 * SCALE)

4120 CONTINUE
IF (IWT .EQ. 1) GO TO 4125
GAP3 = GAP4 = 0.

C Y DIRECTION
C
IF(KSC(NK1).EQ.12)GO TO 4116
IF(ALC1.EQ.0)GO TO 4122
IF(ALC1.LT.0)GO TO 4115
IF(ALPHA(COSAL1+ALFA).GT.1.5)GO TO 4122
GAP3=-GAP*CSAL1
GAP4=-GAP*SINAL1
GO TO 4122

4115 CONTINUE
IF(ALPHA(COSAL1).GT.5)GO TO 4122
GAP3=-GAP*CSAL1
GAP4=-GAP*SINAL1
GO TO 4122

4116 CONTINUE
IF(ALC1.NE.0)GO TO 4117
IF(ALFA.NE.0)GO TO 4122

4117 CONTINUE
IF(ALPHA-SINAL1).GT.5 GO TO 4122
GAP3=-GAP*SINAL1
GAP4=GAP*CSAL1
IF(ALC1.LE.0)GAP4=GAP4

4122 CONTINUE
C1=C1-B*ALFA*SCALE+GAP2*ALFA*GAP3*FCL(NK1,IN1)*1.5/(12.*SCALE)
C3=C2-B*CSAL/SCALE-GAP2*CSAL*GAP4

4125 CONTINUE
IF(RSTIF(LVNC+IV)-LE.1*E-15)GO TO 4150
IF(IS1.EQ.2)GO TO 4130
CALL PLOT(C3,C2,2)
GO TO 4200

4130 CALL PLOT(C1,C3,2)
GO TO 4200

4150 CONTINUE
C112)=C3
C22(1)=C2
IF(IS1.EQ.2)C1112)=C1
IF(IS1.EQ.2)C22(2)=C3
IF(IS1.EQ.2)GAP4=GAP4
CALL DASH(C11,C22,2,1,1,2)

4200 CONTINUE
MKBM=MK*LVNC+IV
IF(IS1.EQ.1)MARK=MOD(MKBM,1000)
IF(IS1.EQ.2)MARK=MKBM/1000
L11=MARK/100
IS1=MARK-L11*100
CALL SQ2BITS(2,IFBB(L11),IS1,1,1,IER)

IF(IS1.EQ.53)
1PRINT 883,IV,MK,NK1,KSC(NK1),KSC(NK1),C1,C2,C3,C2
2C1=C3*GAP1*GAP2*GAP3*GAP4*FCL(NK4)*FCL(NK1,4)*CSAL*ALFA
3*CSAL*SINAL

R83 FORMAT(2,1X,DEBUG,IV,1X,515,6F6.2,18F5.2)
C1142)=C3
C22(2)=C2
IF(IS1.EQ.2)C112)=C1
IF(IS1.EQ.2)C22(2)=C3
IF(IS1.EQ.53)PRINT 886,LV2,LEVEL,IV,10,IFR,K,KA,KB
1HMK,LVNC,C1111,C2211,C112),C22(2),C32(2),BLEM(LVNC+IV),CX(LVNC+IV)

R86 FORMAT(2,1X,DEBUG,PL3,1X,18I4,6F8.2)
4260 CONTINUE
  IF(KP1 .LE. 10) GO TO 4269
  CALL SECBM(J1,J2,J1,JXG,J1+E1+LV2+LV4)

4269 CONTINUE
  DO 4270 I=1,IW
    WP=RP1(IW)
  K=IFR(IW)
  IF(K .LT. IFR) WP=COL1(I+1+I)
  IF(K .EQ. IFR) WP=MB
  DO 4280 K=KA,WP
    MK=MC(K+IW)
    LVNC=(K-1)+LV4+LV2
    C1=CORD(MK,IW)/SCALE
    C2=CORD(MK,IW+1)/SCALE
    IF(IW.EQ.2) C1=C2
    IF(IW.EQ.2) C2=CORD(MK,IW)/SCALE
    K1=K2=0
    CALL UNP53(KST,LVNC,IW+1,A2,R)
    IF(A1.GT.0.05) K1=1
    IF(A2.GT.0.05) K2=1
    GAP1=GAP2=0.

C COMPUTE GAP FOR DRAWING SYMBOLS AND WRITING BEAM MARKS

C C1X=C1Z SYMR. COORD AT LEFT END
C C2X C2Z SYMB. COORD. AT RIGHT END
C C1W=C2W MKR COORD.
C GAP1 GAP AT LEFT END
C GAP2 GAP AT RIGHT END
C K1=K2=0 PINNED CONN.
C K1=K2=1 RIGID CONN.

MK1=MKC(K+1+I)
ALC=ALFAC(MK)
ALC1=ALFAC(MK1)
COSAL=COS(ALC)
SINAL=SIN(ALC)
COSAL1=COS(ALC1)
SINAL1=SIN(ALC1)
GAP=FCL(MK,4)+1.5/(24.*SCALE)
GAP2=FCL(MK1,4)+1.5/(24.*SCALE)

4280 CONTINUE
  IF(IABS(ALFA-SINAL).GT.0.05) GO TO 4283
  GAP3=GAP=COSAL
  GAP4=GAP=SINAL
  GO TO 4283

4283 CONTINUE
  IF(ALC.EQ.-0.) GO TO 4283
  IF(ALC.EQ.-0.) GO TO 4283
  IF(IABS(SINAL-COSAL).GT.0.05) GO TO 4283
GAP3 = GAP + SINAL
GAP4 = GAP + COSAL

4288 CONTINUE
IF (ALFA .LE. 5.0) GO TO 4283
IF (GAP3 .LE. 0.0 .AND. GAP4 .LE. 0.0) GO TO 4284
GAP3 = GAP + SINAL
GAP4 = GAP + COSAL

4283 CONTINUE
GAP3 = GAP4 = 0

C Y DIRECTION
C
IF (KSC (MK) .EQ. 12) GO TO 4291
IF (ALC .EQ. 0.0) GO TO 4283
IF (ALC .GT. 0.0) GO TO 4284
IF (ABS (COSAL + ALFA) .GT. .05) GO TO 4283
GAP3 = GAP + COSAL
GAP4 = GAP + SINAL

4289 CONTINUE
IF (ABS (COSAL - ALFA) .GT. .05) GO TO 4283
GAP3 = GAP + COSAL
GAP4 = GAP + SINAL

4290 CONTINUE
IF (ALC .NE. 0.0) GO TO 4292
IF (ALFA .NE. 0.0) GO TO 4283

4291 CONTINUE
IF (ALC .NE. 0.0) GO TO 4292
IF (ALFA .NE. 0.0) GO TO 4283

4292 CONTINUE
IF (ABS (ALFA - SINAL) .GT. .05) GO TO 4283
GAP3 = GAP + SINAL
GAP4 = GAP + COSAL
IF (ALC .GT. 0.0) GAP3 = GAP3
IF (ALC .LT. 0.0) GAP4 = GAP4

4283 CONTINUE
IF (K1 .EQ. 0.0) GAP1 = 0.033
IF (K2 .EQ. 0.0) GAP2 = 0.033
C1X = C1 - GAP1 * ALFA + GAP3
C1X = C2 - GAP1 * COSA + GAP4
IF (IV .EQ. 1) GO TO 4270
C1X = C1 - GAP1 * ALFA + GAP3
C1X = C2 - GAP1 * COSA + GAP4

4270 CONTINUE
BM = 0.
IF (K .EQ. KB) GO TO 5320
IF (BLMN (LVNC, IV) .EQ. 0.0) GO TO 5320
BM = 1./(BLMN (LVNC, IV) * SCALE)
IF (IV .EQ. 280) GO TO 4276
GAP3 = GAP4 = 0.
IF (KSC (MK1) .EQ. 12) GO TO 4294
IF (ALC1 .NE. 0.0) GO TO 4293
IF (ALFA .NE. 0.0) GO TO 4274

4293 CONTINUE
IF (ABS (ALFA - SINAL1) .GT. .05) GO TO 4274
GAP3 = GAP + COSAL1
GAP4 = GAP + SINAL1
COSC = COS(ALC1)
CWEB = CWEB

IF (ALC1.LT.80.) CWEB1 = -CWEB1 - 0.99/COSC
C1L = CWEB + COSC - CFLAN - SINC * 0.47
C2L = CFLAN - 0.67
C3L = C1L
C4L = C2 + C2L
IF (IWIN.EQ.2) GO TO 4371

CALL UD53(ISI,1,IVC1,LVNC).EQ.41.A2.B)
IF (ALFA.EQ.0.160 TO 4371
ALFA1 = SIN(ALFA)
ALFA2 = ATAN2(C2L,C1L)
IPO = SQRT(C1L**2 + C2L**2)
C3L = C1L * IPO + COS(ALFA1 + ALFA2)
C4L = C2 + IPO * SIN(ALFA1 + ALFA2) - 0.47

4371 CONTINUE
IF (IUG.EQ.53)
PRINT 999, MK, CWEB, CFLAN, SINC, COSC, C1L, C2L, C3L, C4L, ALFA1, ALFA2
999 FORMAT ('COL}}, MK }*I, 12, FE, 3}
CALL PLOT(1,C2,3)

CALL NEWPEN(1)

WRITE BEAM MARKS AND CONNECTION TYPE (FLOOR PLAN)

IF (IPILOT(9),J).EQ.0.AND. RSTIF(LVNC,IW).LE.1.E-15) GO TO 4375
IF (I1.EQ.0.160 TO 4375
IF (I1.EQ.1.160 TO 4375
IF (IPILOT(5),J).EQ.0.160 TO 4375
CALL SYM(54,K1,SCALE,C1X,C2X,K1)

4375 CONTINUE
LVNC = FK - 1) * LVW = LV2
IF (IPILOT(13),J).EQ.0.160 GO TO 4357
IF (I1.EQ.1) MARK = MOD(MK, (LVNC + IW) * 10000)
IF (I1.EQ.2) MARK = MK / (LVNC + IW) / 10000
LI1 = MARK / 100
IS1 = MARK - LI1 * 100
JMV = JMV(1,11)
IF (I1SC4(MK1,LF,.1)160 TO 4388
CALL POSIT(JMV, IV, MK, KM1, JMV, LVNC, C1L, C2L, C3L, C4L, C1W, C2W, SCALE)
IF (IPILOT(9),J).EQ.0.AND. KSC(MK).EQ.0.160 GO TO 4357
CALL LABEL5(3, J, C5L, C4L, C3L, C2L, LVW, 1W, MK, SCALE)

4357 CONTINUE
IF (I1.EQ.0.160 GO TO 4300
IF (SPAN(LI1,IS1),LF,.1)160 TO 4388
IF (IPILOT(18),J).EQ.0.160 TO 4358
RATIO = 5
IF (I1.EQ.0.160 TO 4359
RATIO = WIDTH(LI1,IS1).EQ.0.160 TO 4359
RATIO = WIDTH(LI1,IS1)/DEPTW(LI1,IS1)

4359 CONTINUE
IF (RATIO.GE.5.160 TO 4359
IF (RATIO.EQ.4.160 TO 4359
IF (RATIO.EQ.3.160 TO 4359
IF (RATIO.EQ.2.160 TO 4359
IF (RATIO.EQ.1.160 TO 4359
IF (RATIO.EQ.0.160 TO 4359
IF (RATIO.EQ.-1.160 TO 4359
IF (RATIO.EQ.-2.160 TO 4359
IF (RATIO.EQ.-3.160 TO 4359
IF (RATIO.EQ.-4.160 TO 4359
IF (RATIO.EQ.-5.160 TO 4359
CALL SITES(J, I, IV, C1W, C2W, LVNC, SCALE)
IF (I1JX1).EQ.1.AND. JMV(JX2).EQ.1) GO TO 4383
IF (I1.EQ.2.160 TO 4381
C1W = C1W - C2W + WDT1 + WDT2 * ALFA
GO TO 4274

4294 CONTINUE
IF(ALC1.EQ.0.) GO TO 4274
IF(ALC1.LT.0.) GO TO 4296
IF(ABS(COSAL1+.65).LT.0.5) GO TO 4274
GAP3=GAPP+SINAL1
GAP4=GAPP+COSAL1
GO TO 4274

4296 CONTINUE
IF(ALFA.LE.0.) GO TO 4274
IF(ABS(COSA+SIGNAL1).GT.0.5) GO TO 4274
GAP3=GAPP+SINAL1
GAP4=GAPP+COSAL1

4274 CONTINUE
C2Z=C1+FMA*COSA-GAP2-COSA+GAP3
C2Z=2**N+ALFA-GAP2+ALFA+GAP4+FCL(MK1,IV1)*1.5/12.*SCALE

4276 CONTINUE
IF(IW.EQ.1) GO TO 4284
GAP3=GAP4=8.

C DIRECTION

IF(KSC(MK1).EQ.12) GO TO 4298
IF(ALC1.EQ.0.) GO TO 4278
IF(ALC1.LT.0.) GO TO 4297
IF(ABS(COSAL1+ALFA).LT.0.5) GO TO 4278
GAP3=GAPP+COSAL1
GAP4=GAPP+SINAL1
GO TO 4278

4297 CONTINUE
IF(ABS(COSA+SINAL1).GT.0.5) GO TO 4278
GAP3=GAPP+COSAL1
GAP4=GAPP+SINAL1
GO TO 4278

4298 CONTINUE
IF(ALC1.NE.0.) GO TO 4299
IF(ALFA.NE.0.) GO TO 4278

4299 CONTINUE
IF(ABS(ALFA-SINAL1).GE.0.5) GO TO 4278
GAP3=GAPP+SINAL1
GAP4=GAPP+COSAL1
IF(ALC1.LE.0.) GAP4=-GAP4

4278 CONTINUE
C2Z=C1-8M+ALFA+GAP2+ALFA+GAP3+FCL(MK1,IV1)*1.5/12.*SCALE
C2Z=C2+8M+COSA-GAP2-COSA+GAP4

4284 CONTINUE
C COMPUTE CIW+CZW
C
MK2=MKC(MK1+1,IV1)
C2W=(CORD(MK2,IV1)+CORD(MK1,IV1))/(SCALE*2.)
CIW=C1+8M+.5*COSA
IF(IW.EQ.2) CIW=C1-8M+.5+ALFA
IF(IW.EQ.1) CIW=C2-8M+.5+COSA
IF(IBUG.EQ.53) 1PRINT ATR1+1,IV1,VNC+MK1+KSC(MK1)+KSC(MK1)+MK1+K2+CI+CW
2+CI+CI+CI2+C2Z+C2Z+CIW+CZW+ALFA+COSA+SINAL+COSAL
C
C COMPUTE C3L,CAL FOR COLUMN MARKS (STEEL)
C
E=320  FACTOR=.75
HGT=FACTOR*.125
CFLAN=1.5*EGK(1)*24.*SCALE
ACLI=ALFA1+MW
IF(KSCM).EQ.12.AND.ALCI.LT.0.)ALCI=0.
IF(KSCM).EQ.11.AND.ALCI.GT.0.)ALCI=0.
SINC=SIGN(ALCI)
CSC=CO(1)
CIL=CLEF+SIGN+CFLAN+COSC*.47
C2L=.047-HGT
IF(KSCM).EQ.11)CIL=CLEF+COSC-CFLAN-SINC*.47
C2L=C1+CIL
C4L=C2+C2L
IF(LW.EQ.2)GO TO 4279
CALL UNPS3(KSTLVNC,IW),41,42,B
IF(ALFA.EQ.0.)GO TO 4279
ALFA=ASIN(ALFA)
ALFA2=ATAN2(C2L/CIL)
HIPO=SIGN(CIL*2+C2L*2)
C4L=C1+HIPO+COS(ALFA1+ALFA2)
C4L=C2+HIPO+SIN(ALFA1+ALFA2)*.047
4279 CONTINUE
CALL PLOT(C1,C2,3)
CALL NEWPEN(1)
C
C WRITE BEAM MARKS AND CONNECTION TYPE (FLOOR PLAN)
C
IF(I*PLOT(3,J),EQ.0).AND.*(STIF)(LVNC,IW),LE.1.E-15)GO TO 4285
IF(K1.EQ.0)GO TO 4285
IF(I*PLOT(3,J),EQ.0)GO TO 4285
CALL SYMB(4,K1,SCALE,C1X,C1Z,K,IW)
4285 CONTINUE
LVNC=(K-1)+MV4+L2
IF(I*PLOT(3,J),EQ.0)GO TO 4261
IF(LW.EQ.1)MARK=MOD(MKB(LVNC,IW),10000)
IF(LW.EQ.2)MARX=MOD(MKB(LVNC,IW),1000)
LI1=MARK/100
MARK=MARK%101
IF(KSCM).LE.0)GO TO 4290
CALL POSIT(J1,IV,MK,KA,JS1,LM1,LVNC,CIL,C2L,C3L,C4L,C1V,C2V,SCALE)
IF(I*PLOT(3,J),EQ.1)GO TO 4260
IF(CPRD(MK),EQ.0)GO TO 4261
4260 CONTINUE
CALL LABLE(3,MK3,C3L,C4L,KA,KA,LM,LM1,LVNC,MK,SCALE)
4261 CONTINUE
IF(LW1.EQ.0.5)WRITE(6,(*=0153PL=3014)*)
MARK=I*PLOT(3,J1),J1)J1=1,2,3)
IF(K.EQ.KB)GO TO 4286
IF(STIF(LVNC,MK),EQ.0)ALL=0 TO 4262
IF(STIF(LVNC,MK),EQ.0)ALL=0 TO 4262
IF(STIF(LVNC,MK),EQ.0)ALL=0 TO 4262
IF(STIF(LVNC,MK),EQ.0)ALL=0 TO 4262
IF(STIF(LVNC,MK),EQ.0)ALL=0 TO 4262
IF (ESTIF(LVN=1).LE.1) EE(15) GO TO 4262
IF (IPILOT(1) .EQ. 0) GO TO 4262

IF (IPILOT(14) .EQ. 0) GO TO 4786
CALL LABEL(2,J,1,W,C2W,LYN=1,LYN=2,LYN=3,LYN=4,LYN=5)

4262 CONTINUE
IF (IPILOT(14).EQ.0) GO TO 4766
CALL LABEL(2,J,1,W,C2W,LYN=1,LYN=2,LYN=3,LYN=4,LYN=5)

4266 CONTINUE
IF (K2.EQ.0) GO TO 4290
IF (IPILOT(5,J).EQ.0) GO TO 4290
CALL SYM(15,K,2,SCALE,C2X,C2Y,K,W)

4280 CONTINUE
IF (IPILOT(14,J).EQ.0) GO TO 4764
CALL SECBI(2,J,1,W,C1W,C2W,LYN=1,LYN=2,LYN=4,LYN=5)
IF (IPILOT(13,J).EQ.0) GO TO 4764
CALL SIZE(2,J,1,W,C1W,C2W,LYN=1,LYN=2,LYN=3,LYN=4,LYN=5)

4264 CONTINUE
LVM=LY2-1
KAA=COL1(11,1)
MK=KCI(KAA+1)
C1=SYTIV/2
C2=STPY/M
IF (IPILOT(12,J).EQ.0) GO TO 4290
MR=RP(11)
IFR1=IFP2(1)
SMX1=0.
SMX2=0.
DO 4304 IF=1,IFR1
KA=COL1(IF+1)
4304 CONTINUE
IF (IPILOT(12,J).EQ.0) GO TO 4290
DO 4304 K=KA,KB
K1=KB-K+KA
MK=K1C(11,1)
MK1=K1C(K+1)
LV2=K(1-1)+LVM+LV2
IF (C1(0,LV2+1)).EQ.999. AND. C2(LV1+1)).EQ.999. GO TO 4303
4303 CONTINUE
IF (C1(0,LV2+1)).EQ.999. AND. C2(LV1+1)).EQ.999. GO TO 4304
4304 CONTINUE
C1((SMX2-SMX1)+.5/SCALE
CALL LABEL(56,JO,C1,C2,KA,KB,LYN=4,LYN=5,LYN=6,LYN=7)

4290 CONTINUE
IF (IPILOT(8,J).EQ.0) GO TO 4265
CALL DTM52(J,1,KA,KB,IV,LYN=4,LYN=5,LYN=6,SCALE,STPY/M)

4265 CONTINUE
CALL PLOT(0,8,10,30)
GO TO 4300

C
C CONCRETE STRUCTURE
C
4295 CONTINUE
DO 4360 IV=1,2
MB=MB1(IW)
IFR=IFR2(IW)
IW1=INVER(IW)
DO,4360,T=1,IFR
KA=COL1(I+IW)
IF(T .LT. IFR),KP=COL1(I+IW)-1
IF(T .GE. IFR),KP=MB
DO,4360,K=KA,KP
MK=MK+K(IW)
LVNC=(K-1)+LV4+LV2
C1=CORD(MK,IW)/SCALE
C2=CORD(MK,IW)/SCALE
IF(T .EQ. 90),C1=C2
IF(T .LE. 90),C2=CORD(MK,IW)/SCALE
C1=I+1
C2=I+2
IF(C1+LVNC,IW).EQ.9999 .AND.C2(LVNC,IW).EQ.9999)GO TO 4360
IF(KSC(MK).EQ.90)GO TO 4322
IF(KSC(MK).EQ.90)GO TO 4360
CALL PLOT(C1,C2,3)
CALL SYMB(LS,K1,SCALE,C1,C2,K+IW)
4322 CONTINUE
LVNC=(K-1)+LV4+LV2
IF(T .EQ. MK)GO TO 4355
BM=0.
IF(REN(LVNC,IW).LE.0)GO TO 4330
BM=1./BLEN(LVNC,IW)
4330 CONTINUE
CALL UNP53(KST,LVNC,IW),A1,A2,B
K1=K2=0
IF(A1,E0.05),K1=1
IF(A2,E0.05),K2=1
GAP1=GAP2=0.
MK=MKC(K+1,IW)
IF(T .EQ. 2160)GO TO 4332
GAP1=X(MK)/12.*SCALE
GAP2=X(MK)/12.*SCALE
GO TO 4334
4332 CONTINUE
GAP1=X(MK)/12.*SCALE
GAP2=X(MK)/12.*SCALE
IF(KSC(MK).EQ.3 .OR.KSC(MK).EQ.10)GAP1=Y(MK)/12.*SCALE
IF(KSC(MK).EQ.3 .OR.KSC(MK).EQ.10)GAP2=Y(MK)/12.*SCALE
4334 CONTINUE
IF(K1,E0.01),GAP1=GAP1+2.*HGT
IF(K2,E0.01),GAP2=GAP2+2.*HGT
C1=C1+GAP1*COSA
C2=C2+GAP1+ALFA+FCL(MK,IW)/(12.*SCALE)
IF(T .EQ. 1160)GO TO 4336
C1=C1-GAP1+ALFA+FCL(MK,IW)/(12.*SCALE)
C2=C2-GAP1+ALFA+FCL(MK,IW)/(12.*SCALE)
4336 CONTINUE
CALL PLOT(C1,C2,3)
IF(T .EQ. 2160)GO TO 4340
4340 CONTINUE
4342 IF(T .EQ. 2160)GO TO 4342
C3=C2*B*(1-Alfa*Alfa)**.5/SCALE-GAP2*COSA
C1=C1-R*ALFA/SCALE+GAP2*ALFA+FCL(MK1+MK1)/(12*SCALE)

4342 CONTINUE
IF(IBSTIF(LVNC+IU1.LE.1.E-15) GO TO 4350
MKBM=MKB(LVNC+IU1)
IF(ITV.EQ.1) MARK=MOD(MKBM,18000)
IF(ITV.EQ.2) MARK=MKB/10000
LI1=MARK+100
IS1=MARK-LI1+100
RATIO=K
IF(LI1.EQ.4,3,ANG2,IBUS.EQ.53) PRINT 999,MARK,MK,MK1,C1C2,C2,C3,
1WDTH=(LI1,IS1),DEPTH(LI1,IS1),ALFA,GAP1,GAP2,X(MK),X(MK1)
888 FOR=1,(* BM=301,315,1111,131)
IF(DEPTH(LI1,IS1),LE.8.16) GO TO 4341
RATIO=WDTH(LI1,IS1),DEPTH(LI1,IS1)
4341 CONTINUE
IF(IBSTIF(LVNC+IU1,EQ.WALL)WDTH1=WDTH(LI1,IS1)*.5/(SCALE+.12)
IF(ITV.EQ.2) GO TO 4345
IF(IBSTIF(LVNC+IU1),NE.WALL)GC TO 4343
C2C=C2+WDTH1
CALL PLOT(C1C2+C2,C3)
C2=C2-WDTH1*2
CALL PLOT(C3,C2,C2)
CALL PLOT(C3+C2,C3)
CALL PLOT(C1C2+C2,C3)
C2C2=C2+WDTH1*2
CALL PLOT(C1C2+C2,C3)
GO TO 4355
4343 CONTINUE
IF(RATIO.EQ.5) GO TO 4344
C111=C11C1
WDTH1=WDTH(LI1,IS1)/(SCALE+.24)
C221=C2C2+YBIN
C1112=C3
C2212=C2-YBIN
CALL DASH(C111+C2212+1,1)
C2211=C2C2+YBIN
C2212=C2+YBIN
CALL DASH(C111+C2212+1,1)
LY1=LY(MK)
JY1=JY(MK)
LY2=LY(MK1)
JY2=JY(MK1)
IF(JY1(JY2).LT.1.AND.IWKLY1.NE.WJY1) GO TO 4355
IF(LY1.EQ.0.AND.IWK(CMK).EQ.9) GO TO 4355
YBIN2=YBIN
YBIN=YBIN/((24.*SCALE))
IF(KSC(MK).EQ.9) YBIN=YBIN/(12.*SCALE)
IF(KSC(MK).EQ.8) YBIN=(X(MK1)/SCALE=.24)
IF(YBIN2.81=YBIN1YBIN2
C22C2=C2C2+YBIN
IF(UNWLY1.EQ.JY1.AND.IWK(MK1).EQ.JY2+C22C2=C2C2+YBIN
CALL PLOT(C1C2+C2,C3)
YBIN1=YBIN1/(SCALE=.24)
IF(KSC(MK1).EQ.2) YBIN1=X(MK1)/(SCALE=.24)
IF(KSC(MK1).EQ.2) YBIN1=X(MK1)/(12.*SCALE)
IF(YBIN2.81=YBIN1YBIN1=YBIN2
C2C2=C2-YBM

IF(JMX(LY1).EQ.JY1.AND.JMX(LY2).EQ.JY2) C2C2=C2+YBM

CALL PLOT(C1+C2C2+2)

GO TO 4355.

4344 CONTINUE

C
C SYMBOL FOR SLABS XDIRECTION
C
ALFA1=1.0/57.2958
ALFA2=A16M(ALFA1)
ARRY0=1.0*COS(ALFA1+ALFA2)
ARRY0=1.0*SIN(ALFA1+ALFA2)
LY1=LY(YK)
JY1=JY(YK)
LY2=LY(YK1)
JY2=JY(YK1)

IF(JY(JK).LT.1.AND.JMX(LY1).LT.JY1) GO TO 4352

IF(LY1.EQ.0.AND.KSC(JK).EQ.0) GO TO 4352

YBM=YMK)/(24.*SCALE)

IF(KSC(JK).EQ.91) YBM=YMK1)/(24.*SCALE)

C3C=C2C-YBM

IF(JMX(LY1).EQ.JY1.AND.JMX(LY2).EQ.JY2) C3C=C2C+YBM

CALL PLOT(C1C3C+3)

YBM=YMK1)/(24.*SCALE)

IF(KSC(JK).EQ.91)YBM=YMK1)/(24.*SCALE)

C3C=C2+YBM

IF(JMX(LY1).EQ.JY1.AND.JMX(LY2).EQ.JY2) C3C=C2+YBM

CALL PLOT(C3C+3)

4352 CONTINUE

IF(1PLOT(14,J).EQ.0) GO TO 4355

C3C=CICGAP1+.125)*COSA
C3C=C2C+(GAP1+.125)*ALFA
C4C=CIC-EAP2+.125)*COSA
C5C=C2C-(GAP2+.125)*ALFA

CALL PLOT(C5C+C2C+3)
CALL PLOT(C3C+C5C+2)
CALL PLOT(C3C+C2C+3)

C3C=C3C+ARROY
C2C=C2C+ARROY
CALL PLOT(C3C+C2C+2)
CALL PLOT(C3C+C5C+3)

C4C=C4C-ARROX
C5C=C5C-ARROX
CALL PLOT(C4C+C5C+2)

GO TO 4355

4345 CONTINUE

IF(0STIF(LYNMC+1).NE.WALL) GO TO 4346

C1=C1+CID1
XD=1.5*MK1/(SCALE*24+)
IF(KSC(JK).EQ.5).OR.KSC(JK).EQ.18) XD=XD+Y(YK)/YMK

CALL PLOT(C1C2C+3)
CALL PLOT(C1C3C+2)
C1=C1+CID1+2.
CALL PLOT(C1C5C+2)
CALL PLOT(C1C2C+2)
C1=C1+CID1+2.

CALL PLOT(C1C4C+2)
GO TO 4355

4346 CONTINUE

IF (RATIO.GE.5.E-6) GO TO 4347

YBIN=WIDTH(L1) .IS1 .IS1) / (SCALE*24.)
C11(1)=C1+YBIN*CO3A
C22(1)=C2+YBIN*ALFA
C11(2)=C1+YBIN*CO3A
C22(2)=C3+YBIN*ALFA
CALL DASH(C11,C22,2++.1+++1)

C11(1)=C1+YBIN*CO3A
C11(2)=C1+YBIN*CO3A
C22(1)=C2+YBIN*ALFA
C22(2)=C3+YBIN*ALFA
CALL DASH(C11,C22,2++.1+++1)

LY1=LX(MK)
JY1=JX(MK)
LY2=LX(MK)
JY2=JX(MK)

IF (JX(MK).GT.1.AND.JX(MK).ME.JY1) GO TO 4355
IF (LY1.EQ.0.AND.KSC(MK).EQ.0) GO TO 4355

YBIM2=YBIN

YBIM=X(MK) / (24.*SCALE)

IF (KSC(MK).EQ.0) YBIN=X(MK) / (24.*SCALE)

YBIM2=G1+YBIN1+YBIN1=YBIN2
C2C2=C1-CYBIN+CO3A

IF (JXLY1).EQ.J1.AND.JX(MK).EQ.JY2) C2C2=C1+YBIN+CO3A

CALL PLOT(C2C2,C2C3)

YBIN1=X(MK1) / (SCALE*24.)

IF (KSC(MK1).EQ.0) YBIN1=X(MK1) / (24.*SCALE)

YBIM2=G1+YBIN1+YBIN1=YBIN2
C2C2=C1-CYBIN+CO3A

IF (JXLY1).EQ.J1.AND.JX(MK).EQ.JY2) C2C2=C1+YBIN+CO3A

CALL PLOT(C2C2,C3,2)

GO TO 4355

4347 CONTINUE

C
C SYMOL FOR SLABS YDIRECTION

C

ALFA1=15.*57.29578

ALFA2=AS(IN(ALFA))

LY1=LX(MK)
JY1=JX(MK)
LY2=LX(MK)
JY2=JX(MK)

IF (JX(MK).GT.1.AND.JX(MK).ME.JY1) GO TO 4353

IF (LY1.EQ.0.AND.KSC(MK).EQ.0) GO TO 4353

YBIM=X(MK) / (24.*SCALE)

IF (KSC(MK).EQ.0) YBIN=X(MK) / (24.*SCALE)

C3=C1+YBIN

IF (JXLY1).EQ.J1.AND.JX(MK).EQ.JY2) C3=C1+YBIN

CALL PLOT(C3,C2,C3)

YBIN=X(MK1) / (SCALE*24.)

IF (KSC(MK1).EQ.0) YBIN=X(MK1) / (24.*SCALE)

C3=C1+YBIN

IF (JXLY1).EQ.J1.AND.JX(MK).EQ.JY2) C3=C1+YBIN

CALL PLOT(C3,C3,2)

4355 CONTINUE
IF(I.EQ.0) GO TO 4355
C3C=CIC-(GAP1+125)*ALFA
C2C=C2C-(GAP1+125)*COSA
C4C=CIC-(GAP2+125)*ALFA
C5C=CIC-(GAP2+125)*CSCA
CALL PLOT(C3C,C2C,0)
CALL PLOT(C4C,C5C,2)
CALL PLOT(C3C,C2C,3)
CALL PLOT(C4C,C5C,3)
CALL PLOT(C3C-141,IV(ALFA1+ALFA2)
C2C=C2C+1*COS(ALFA1+ALFA2)
CALL PLOT(C3C,C2C,2)
CALL PLOT(C4C,C5C,3)
CIC=CIC+1*SIN(ALFA2+ALFA1)
C5C=C5C+1*COS(ALFA2+ALFA1)
CALL PLOT(CIC,C5C,2)
GO TO 4355
4350 CONTINUE
C112(C)=C3
C222(C)=C2
IF(I.W.EQ.2)C112(C)=C1
IF(I.W.EQ.2)C222(C)=C3
IF(I.PLOT(9,J).EQ.0)GO TO 4355
CALL DASH(C112,C222,1,1,2)
4355 CONTINUE
CALL SGBITS(2,IFBB(ILI),IS1,1,IER)
IF(I.W.EQ.13)
IPRINT 835,IW,I(WK),KSC,(MK),KSC(MK),CIC,C2C,C3,C2,
C12,C3,GAP1,GAP2,X(MK),X(MK)
C112(C)=C3
C222(C)=C2
IF(I.W.EQ.2)C112(C)=C1
IF(I.W.EQ.2)C222(C)=C3
IF(I.W.EQ.0.53)PRINT 886,LY2,LEVEL,IV,IW,IFR,K,KA,KB,
KV,LVNC,C112(1),C222(1),C112(2),C222(2),BLN(LVNC,IV),EX(LVNC,IV)
4360 CONTINUE
IF(I.PLOT(16,J).E.Q.0)GO TO 4362
CALL SECRED(IW,IVN,1,IVN,LIG,V,LV2,LVN4)
4362 CONTINUE
DO 4360 IV=1,WK
MB=MB11(W)
IFR=IFR2(W)
IW=IW1(W)
DO 4380 I=1,IFR
KB=COL1(I,IV)
IF(1.LT.IF)KB=COL1(I+1,IV)-1
IF(I.EQ.IFR)KB=MB
DO 4380 K=KA,KB
MK=MK(C(K,1)
LVNC=K-1LVM4+LV2
C1=COR0(MK,IV)/SCALE
C2=COR0(MK,IV)/SCALE
IF(I.W.EQ.0)C1=C2
IF(I.W.EQ.2)C2=COR0(MK,IV)/SCALE
K1=K2=K
CALL XHME(K,K8(LVNC,IV),A1,A2,B)
IF(1)AT=K8(IV)
IF(K2.B.EQ.95)K2=1
C COMPUTE GAP FOR DRAWING SYMBOLS AND WRITING BEAM MARKS
C C1X=C1Z SYMP. COORD AT LEFT END
C C2X C2Z SYMP. COORD. AT RIGHT END
C C1W=C2W MRK COORD.
C GAP1 GAP AT LEFT END
C GAP2 GAP AT RIGHT END
C K1=K2=0 PINNED CONN.
C K1=K2=1 RIGID CONN.
C
MK1=MKC(K+1+1W)
LX1=LX(MK)
JX1=JX(MK)
LX2=LX(MK1)
JX2=JX(MK1)
IF(IW.EQ.2)GO TO 4364
LX1=LY(MK)
JX1=JY(MK)
LX2=LY(MK1)
JX2=JY(MK1)
4364 CONTINUE
GAP1=X(MK)/(24.*SCALE)
GAP2=X(MK1)/(24.*SCALE)
IF(IW.EQ.1)GO TO 4372
4370 CONTINUE
IF(KSC(MK).EQ.3.OR.KSC(MK1).EQ.10)GAP1=Y(MK)/(24.*SCALE)
IF(KSC(MK1).EQ.3.OR.KSC(MK1).EQ.10)GAP2=Y(MK1)/(24.*SCALE)
4372 CONTINUE
IF(K1.EQ.0)GO TO 4374
IF(K2.EQ.0)GO TO 4374
C1X=C1+GAP1
C1Z=C2+GAP1+ALFA/COSA
BM=0.
IF(K.EQ.KB)GO TO 4373
4373 CONTINUE
IF(BLEN(LVMC,IW).EQ.8.)GO TO 4373
BM=1./(BLEN(LVMC,IW)*SCALE)
C2X=C1+BM*COSA-GAP2
C2Z=C2+BM*ALFA-GAP2+ALFA/COSA
IF(IW.EQ.1)GO TO 4374
C1X=C1+GAP1+ALFA/COSA
C1Z=C2+GAP1
C2X=C1+BM*ALFA-GAP2
C2Z=C2+BM+ALFA-GAP2+ALFA/COSA
4374 CONTINUE
C C COMPUTE C1W+C2W
C
MK2=MKC(K+1+1W)
MK2=MK2(MK2(IW+2W))
MARK=MOD(MRKB+1000,)
IF(IW.EQ.2)MARK=MKB/1000
L1=MARK/100
IS1=MARK-L1/100
WIDTH1=WDTM(L11,IS1)/(SCALE=24.)
YD1=0.
YDIM1=0.
IF(JX) = CT.1.AND.(JMK(J1)) NE.1X11) GO TO 436A
IF(IY) EQ.0.AND.KSC(M2) EQ.91) GO TO 436A
IF(IY) EQ.1160 TO 4367
Y1N = X(MK1) / (24.0*SCALE)
IF(KSC(MK1).EQ.9) Y1N = Y(MK1) / (24.0*SCALE)
Y1M = Y(MK1) / (24.0*SCALE)
IF(KSC(M1).EQ.0) Y1M = Y(MK1) / (24.0*SCALE)
GO TO 436A

4367 CONTINUE
Y1N = Y(MK1) / (24.0*SCALE)
IF(KSC(MK1).EQ.9) Y1N = Y(MK1) / (24.0*SCALE)
Y1M = Y(MK1) / (24.0*SCALE)
IF(KSC(MK1).EQ.0) Y1M = Y(MK1) / (24.0*SCALE)

4368 CONTINUE
Y1N = Y(MK1) / (24.0*SCALE)
IF(DEPTH(M1),IS1).EQ.00) GO TO 4369
RATIO = WIDTH (M1,IS1)/DEPTH(M1,IS1)
IF(RATIO. GE. 5.) WIDTH = 0.

4369 CONTINUE
WIDTH = MAX (Y1N, Y1N, WIDTH)

IF(IY) EQ.21C11 = C1 - B.5*COSA - WIDT1 - ALFA
IF(IY) EQ.21C12 = C1 - B.5*ALFA - WIDT1 - COSA
IF(IY) EQ.11C11 = C1 - B.5*ALFA - WIDT1 - COSA
IF(IY) EQ.11C11 = C1 - B.5)
PRINT RTN, M1, Y1M, Y1N, MK, MK1, KSC(MK1), KSC(MK1), K1, K2, C1, C2
2, C1X, C1Y, C1Z, C2X, C2Y, C2Z, ALFA, COSA
C
C COMPUTE CSLCAL FOR COLUMN MARKS (CONCRETE)
C
4373 CONTINUE
FACTOR = .75
HGT = FACTOR * 12
LX = Y1N
JX = JX(MK)
LY = Y1N
JY = JY(MK)

IF(DEPTH(L1,JX). EQ.0.) OR.DEPTH(L1,JY). EQ.0.) GO TO 4376
WIDTH = WIDTH + DEPTH(L1,JX)
WIDTH = WIDTH + DEPTH(L1,JY)
RATIO = WIDTH / DEPTH(L1,JX)
RATIO = WIDTH / DEPTH(L1,JY)
IF(RATIO. GE. 5.) WIDTH = 0.
IF(RATIO. GE. 5.) WIDTH = 0.
GO TO 4379

4376 CONTINUE
WIDTH = 0.
WIDTH = 0.

4379 CONTINUE
CWEB1 = MAX (MK1, WIDTH, WIDTH)
CPLAN1 = MAX (MK1, WIDTH, WIDTH)
IF(KSC(MK1).EQ.3) CPLAN1 = MAX (MK1, WIDTH, WIDTH)
CWEB = 1.5*CWEB1 / (24.0*SCALE)
CPLAN = 1.5*CPLAN1 / (24.0*SCALE)
ALCI = ALFA(MK1)
IF(ALCI. GE. 80.) ALC1 = 0.
SINC = SIN (ALCI)
COSC=COS(ALC1)
CWEB1=CWEB

IF(ALC1.LT.0.1)CWEB1=-CWEB1-.094/COSC
C1L=CWEB*COSC-CLFLN+SINC+.047
C2L=-CFLAN-M6T
C3L=C1*C3L
C4L=C2*C2L

IF(IN.EQ.2160)GO TO 4371
CALL UNF53(KST(LVNC,IW),A1,A2,A8)
IF(ALFA.GT.0.160)GO TO 4371
ALFA1=ASIN(ALFA)
ALFA2=ATAN(C2L/C1L)
UP=SORT(C1L**2+C2L**2)
C1L=C1L*UP/COS(ALFA1+ALFA2)
C2L=C2L*UP/SIN(ALFA1+ALFA2)**.047

4371 CONTINUE
IF(IRUG.EQ.53)
1PRINT 999,NK,CWEB,CLFLN,SINC,COSC,C1,C2,C3L,C4L,C1L,C2L,ALFA1,ALFA2
999 FORMAT(* COL,NK *,I4,12F8.3)
CALL PLOT(C1,C2,C3)

CALL NEWPEN(1)

WRITE BEAM MARKS AND CONNECTION TYPE (FLOOR PLAN)

IF(IPLT(J,1,1).EQ.0.0.AND.KSTIF(LVNC,IW).LE.1.E-15)GO TO 4375

IF(k1.EQ.1)GO TO 4375

IF(IPLT(5,J).EQ.0.0)GO TO 4375
CALL SYM(4,j,1,SCALE,C3L,C12,K,IW)

4375 CONTINUE
LVNC=EX-1,1+LV=M+LV2
IF(IPLT(13,J).EQ.0.0)GO TO 4377
IF(IW,EQ.1)MARK=MOD(MKB(LVNC,IW),10000)
IF(IW,EQ.2)MARK=MKB(LVNC,IW)/10000
L11=MARK/100
IS1=MARK-L11*100
JMW=JMW(L11)
IF(KSCA4M.K.EQ.0.160)GO TO 4380
CALL POSIT(J,IV,NK,K,KA,JPHN,LVNC,C1L,C2L,C3L,C4L,C1W,C2W,SCALE)
IF(IPLT(9,J).EQ.0.0.AND.KSCA4M.EQ.9.0)GO TO 4377
CALL LABEL5(3,JoW,C4L,KJ,KA,KB,LVM4,IV,MK,SCALE)

4377 CONTINUE
IF(EQ.KB)GO TO 4380
IF(SPANL11(l,IS1).LE.8,160)GO TO 4380
IF(IPLT(16,J).EQ.0.0)GO TO 4358
RATIO=.5
IF(DEPTH(L11,IS1).EQ.0.0)GO TO 4359
RATIO=WIDTM1L1,IS1/DEPTWML1,IS1

4359 CONTINUE
IF(RATIO.GE.5.0)GO TO 4358
IF(BSTIF(LVNC,IW).EQ.WALL)GO TO 4358
IF(BSTIF(LVNC,IW).LE.1.E-15)GO TO 4358
WIDT=WIDTM1L1,IS1/(24.,SCALE)
CALL SIZES(J,JW,C1W,C3L,WIDT,SCALE)
IF(JMKH(JL1),EQ.JK1.AND.JM(X,LX2),EQ.JK2)GO TO 4383
IF(IW.EQ.2160)GO TO 4381
C1N=C1N+6*WIDT1=WIDT2)*ALFA
C2W=C2W-(WIDT1-WIDT2)*COSA
GO TO 4381
4381 CONTINUE
C1W=C1W+(WIDT1-WIDT2)*COSA
C2W=C2W+(WIDT1-WIDT2)*ALFA
4382 CONTINUE
IF(IPL0T14.0).EQ.0) GO TO 4387
WIDT2=WIDT4*(L1+IS1)/(40.*SCALE)
IF(KRAT1.GE.5.*WIDT2) GOTO 4385
C1W=C1W-2.*WIDT1*COSA
C2W=C2W-2.*WIDT1*ALFA
IF(JW1X(IY1).EQ.X1)X1=0.0 X=JW1X(IY2).EQ.X2)GO TO 4387
C1W=C1W-(WIDT1-WIDT2)*COSA
C2W=C2W-(WIDT1-WIDT2)*ALFA
GO TO 4387
4385 CONTINUE
C1W=C1W-2.*WIDT1*ALFA
C2W=C2W-2.*WIDT1*COSA
C1W=C1W+(WIDT1-WIDT2)*ALFA
C2W=C2W+(WIDT1-WIDT2)*COSA
4387 CONTINUE
CALL LNLX(4,J,C1W+C2W,KA,KP,LVM4,Id,LVNC,SCALE)
4377 IF(K2.EQ.0) GO TO 4380
4380 CONTINUE
IF(IPL0T15.0).EQ.0) GO TO 4384
CALL SECBN(2,J1,XMX+LIG+NR+LV2+LVM4)
4382 CONTINUE
IF(IPL0T19.0).EQ.0) GO TO 4384
CALL LNSO(2,J1,J1+C1W+C2W,LVNC,SCALE)
4384 CONTINUE
IF(IPL0T8.0).EQ.0) GO TO 4386
CALL DIN5(2,J1,KA,KP,IV,LVM4,LV2,SCALE,STPMN)
4386 CONTINUE
LV2=LV2-1
KA=COL1(1,1)
MK=MKC(KAA+1)
C2=STPMN
IF(IPL0T14.0).EQ.0) GO TO 4390
MB=(R11)
IFR=IFR2(1)
SMX=0
SMX2=0
DO 4396 IF=1,IFR
KA=COL1(IF+1)
IF(IF.LE.IFR)KR=COL1(IF+1)-1
IF(IF.GE.IFR)KB=MB
DO 4396 KA=KA+1
K1=KR+KA
MK=MKC(K+1)
MK1=MKC(K1+1)
LV1=(K-1)+LVM4+LV2
LV2=(K1-1)+LVM4+LV2
IF(C1X(LV1).EQ.999).AND.(C2X(LV1).EQ.999) GO TO 4395
IF(CORD(MK+1).LT.SMX1)SMX1=CORD(MK+1)
4395 CONTINUE
4396 CONTINUE
C1=(SMX2-SMNX1)*S/SCALE
CALL LABELS(6,J,C1,C2,KB,LVMA,1W,LVW,SCALE)
4390 CONTINUE
CALL PLOT(0.0,0.0,39)
4390 CONTINUE
IF(IFPL2.EQ.0) GOTO 4310
GOTO 5200
4305 CONTINUE
C C ERROR MESSAGE****** STACK NOS. Do not correspond to
C ANY COLUMN IN FRAMES
C
PRINT 871
871 FORMAT(* *** MESSAGE PL-1 *** STACK NOS DO NOT*
1* CORRESPOND TO ANY FRAME *)
GOTO 998
4310 CONTINUE
C C ERROR MESSAGE ****** LEVEL NO. to be plotted do not
C CORRESPOND TO ANY LEVEL OF THE STRUCTURE
C
PRINT 872
872 FORMAT(* *** MESSAGE PL-2 *** LEVELS TO BE PLOTTED*
1* DO NOT CORRESPOND TO THE ACTUAL LEVELS*)
GOTO 998
5200 CONTINUE
9999 CALL PLOT(0.0,0.0,999)
IF(LOTO(1).EQ.5) STOP
9998 CONTINUE
END

--- MAP (LO=A) ---
ADDRESS -- BLOCK ---- PROPERTIES ------ TYPE ------ SIZE -- NAME ---- ADDRESS ---- BLOCK -----
138 /CONSTS/
   REAL          3 PMSH 1750B /THETFL/
143558 REAL          BON  642B //
143568 REAL          BSTIF  08 /PLOT1/
08 /ALFA/
   REAL          B1 14331B
08 /ALFAC/
   REAL          140 CA 14334B
141338 REAL          CA 14335B
144140 REAL          CFLAN 14404B
144420 REAL          CFLAN 14470B
48 /ALS/
   REAL          CJP 253B /DATATZ/
144420 REAL          CLDAT 275B /DATATZ/
144430 REAL          CLENI 27010B /PLOT1/
141268 REAL          CLEV 2367B /BMFL4/
141278 REAL          COL  423B //
141308 REAL          COL1 1266B
08 /BATS/
   REAL          10 COL1 449B /BMFL4/
50508 /BMFL4/    REAL          240 CONS 0B /CONSTS/
17758 /THETFL/    REAL          600 CORD 100B /BMFL4/
36608 /PLOT1/    REAL          196B CORDY 14172B
141648 REAL          CORDY 14172B
SUBROUTINE ARROW(10,C1,C12,C27)

C

C

ALFA1=15.0/57.29578
ARIOY=.07*SIN(ALFA1)
ARIOY=.07*COS(ALFA1)
IF(10.EQ.2)GO TO 100
X1=C1

Y1=C12
X2=C1
Y2=C27
XC1=X1-ARIOY
YC1=Y1+ARIOY
XC2=X1+ARIOY
YC2=YC1
XC3=X2-ARIOY
YC3=Y2-ARIOY
XC4=X2+ARIOY
YC4=YC3
GO TO 120

100 CONTINUE
ARIOY=ARIOY
ARIOY=.07*SIN(ALFA1)
X1=C12
Y1=C1
X2=C27
Y2=C1
XC1=X1+ARIOY
YC1=Y1+ARIOY
XC2=XC1
YC2=Y1+ARIOY
XC3=X2-ARIOY
YC3=Y2+ARIOY
XC4=XC3
YC4=Y2+ARIOY

120 CONTINUE
CALL PLOT(X1,Y1,3)
CALL PLOT(X1,Y1,2)
CALL PLOT(XC1,YC1,2)
CALL PLOT(XC1,YC1,2)
CALL PLOT(XC2,YC2,2)
CALL PLOT(X2,Y2,3)
CALL PLOT(XC3,YC3,2)
CALL PLOT(XC4,YC4,2)
RETURN
END

FILE MAP--LOC=6
ADDRESS--BLOCK----PROPERTIES-----TYPE--------SIZE-------NAME------ADDRESS--BLOCK--

132B   4 REAL          ID       1 DUMMY-A
133B   1 REAL          XC1       141B
134B   1 REAL          XC2       143B
2 DUMMY-ARG REAL          XC3       145R
3 DUMMY-ARG REAL          XC4       147B
4 DUMMY-ARG REAL          X1       135R
SUBROUTINE BORDER (XORG, YORG, XFRAME, YFRAME, WBORDER, RADBORD)

PMIN = 0.07
PENTHlk = P
WIDTH = MIN ((XFRAME - YFRAME) - 2. * WBORDER, PENTHlk * WIDT / 3)
IF (PENTHlk * WIDT / 2. > WIDTH) PENTHlk = WIDTH / 2. * NHANCE
IF (PENTHlk > PMIN) PENTHlk = PMIN
FRAME = WBORDER + RADBORD

DELTAx = XFRAME - 2. * FRAME
DELTAy = YFRAME - 2. * FRAME
ISMOOTH = 25
XINCR = 1.570796 / ISMOOTH
X = XFRAME - FRAME + XORG
Y = FRAME + YORG

NHANCE = 5. * WIDT / PENTHlk
NHANCE = MIN (NHANCE, NHANCE, NHANCE)
IF (PENTHlk * NHANCE > WIDT / 2.) PENTHlk = WIDT / 2. / NHANCE
DO 3 N = 1, NHANCE
XSTART = XFRAME - WBORDER - (N - 5) * PENTHlk / YORG
YSTART = YFRAME - FRAME + YORG

RAD = RADBORD - (N - 5) * PENTHlk
IF (RAD > 0.) GO TO 5
X = XFRAME - FRAME + XORG + RAD
Y = FRAME + YORG - RAD

YSTART = YFRAME - FRAME + YORG + RAD

DELTAx = XFRAME - 2. * (FRAME - RAD)
DELTAy = YFRAME - 2. * (FRAME - RAD)
CALL PLOT (XSTART, YSTART, 5)

DO 2 M = 1, 4
ANGLE = M * 1.570796
X = X + DELTAx * COS (ANGLE)
Y = Y + DELTAy * SIN (ANGLE)
IF (RAD > 0.) GO TO 4
CALL PLOT (X, Y, 2)

GO TO 2
4 CONTINUE

ISMOO = ISMOO + 1
DO 1 L = 1, ISMOO

XANGLE = ANGLE + (L - 1) * XINCR - 1.570796
XX = X + RAD * COS (XANGLE)
YY = Y + RAD * SIN (XANGLE)
1 CALL PLOT (XX, YY, 1)
2 CONTINUE
3 CONTINUE
RETURN
END

IF MAP -- (LO=1)

-ADDRESS--BLOCK-------PROPERTIES------TYPE------SIZE------NAME-----ADDRESS--BLOCK--

| 241B | REAL | ISMOO | 226B |
| 242B | REAL | L | 243B |
| 245B | REAL | M | 248B |
| 223B | REAL | N | 233B |
| 242B | INTEGER | NHANCE | 232P |
SUBROUTINE DIMS(DIMx, DIMy, SCALE, SCALEy, SCALEz, STZMx, STZMy, STZMz)
COMMON 7ZZZ, NME, DAT, I1Q, LVMA, LVMb, LVV, SCALE, STZMX, STZMY, STZMZ
COMMON JSTOP, NSTOP, IPT, LNKEC, EC, EC5, MAXFR, ITEST, IRUN, LVU, LNU
COMMON KAK, LIST, MD, UF, MD, IF, KFT
COMMONotic(90), OPT, OPT, OPT, OPT, OPT
INTEGER KAK(90, 2), NME(11, 3)
COMMON PLOT, PLOT, PLOT, PLOT, PLOT, PLOT, PLOT, PLOT, PLOT, PLOT
COMMON (HMF, STP, STP, STP, STP, STP, STP, STP, STP, STP)
1 1CO, ICFL(140), IFB, GRID(140, 2), NCOL(2), MCK(140, 2)
4 CLEV(90), IBUM, STZ(1), STZ(2), IFLAG(140), IFLAG(140), KPM(140)
5 SIDE(10), MARG, IGRID(25, 15), JPL, IGRID(140), IGRID(140, 2)
6 DUMMY(240)
COMMON (HMF, SPAN, XBM, XBM, XBM, XBM, XBM, XBM, XBM, XBM)
COMMON (CLFL, LY, LY, LY, LY, LY, LY, LY, LY, LY)
REAL MARG(10)
INTEGER COL(140, 2)
COMMON (REZl, KUZ, UZ1, UZ2, UZ3, UZ4, UZ5, UZ6, UZ7, UZ8, UZ9, UZ10, UZ11, UZ12, UZ13, UZ14, UZ15, UZ16, UZ17, UZ18)
DIMENSION LEVEL(90)

C

FACTOR = 0.75
MGT = FACTOR * 0.75
LEVEL = LVMA
IF (KIDEv.x, 2) GO TO 550
DO 50 K = KA, KB

30 LVMA = (KK - 1) * LVMA + LEVEL
IF (CK(LVMA, IV) EQ 999 .AND. CZ(LVMA, IV) EQ 999) GO TO 40
GO TO 60

40 CONTINUE
IF (K[EQ, KB]) LEVEL = LEVEL - 1
IF (K[EQ, KB]) GO TO 50
K = K + 1
GO TO 30

50 CONTINUE
60 CONTINUE
C2Z = CLEV(LEVEL) / SCALE
C1Z = 1.25
C2Z = CLE[LEVEL + 1]/SCALE
CALL PLOT(C1Z, C2Z, 1)
DO 130 I = 2, LEVEL

C C2Z VCOORD I OF LEVEL I
C C1Z VCOORD I OF LEVEL I - 1
C

C2Z = CLEV(LEVEL) / SCALE
C1Z = CLEV(LEVEL - 1) / SCALE
SUBROUTINE DIM5/2  74/74  OPT=1  FTN 5.1-552  84

IF(I.GT.2) GO TO 110
C12=CZ(LVN+1)*SCALE
CALL PLOT(-.125,C12,3)
CALL PLOT(-1.39,C12,2)

110 CONTINUE
IF(IPSU.EQ.53) PRINT 997*LEVEL+C12 = C22
FORMAT(* DIM5-1*214,2F10.2)
CALL ARROWS1(C1,C12,C22)

130 CONTINUE

C WRITE INTERSTORY HEIGHT
C
DO 190 I=2*LEVEL
DO 140 K=KA,KR
LVN1=(K-1)*LM4+1
IF(CZ(LVN1+1)*EQ.999) GO TO 180
DIMEN=ABS(CZ(LVN1+1)-CZ(LVN1+1)*UZ11)
C12=(CLEV1)/DIMEN*.5*UZ11/SCALE
C1X=1.297
IF(IPSU.EQ.53) PRINT 998*LEVEL+DIM5-2*14,4F10.2
FORMAT(* DIM5-2*14,4F10.2)
HT=FACTOR*125
ANGLE=90.
CALL DIM5(11*LEVEL+DIMNEW12,C1X,ANGLE,HT,SCALE)
GO TO 190

180 CONTINUE

190 CONTINUE
DO 220 K=KA,KB
DO 210 L=2*LEVEL
L1=LEVEL-L+2
LVN=(K-1)*LM4+L1
LEV(K)=L1
IF(CZ(LVN+1)*EQ.999 AND CZ(LVN+1)*EQ.999) GO TO 210
GO TO 220

210 CONTINUE

220 CONTINUE
ITEM=0
DO 400 K=KA,KB
LVN=(K-1)*LM4+LEVEL(K)
EXO=STZMX*.5-CZ(LVN+1)/SCALE
IF(IPSU.EQ.53) PRINT 996*LM4,K,KB,LEV(K),STZMX,CZ(LVN+1)/SCALE,EXO
FORMAT(* DIM5-3*14,3F12.2)
CALL LIN5(1,K,KB,LM4,IOU,LM4,EXO,SCALE)

400 CONTINUE

C

LVN1=(KA-1)*LM4+(IDOS3+3)*1
LVN2=(KB-1)*LM4+(IDOS3+3)*1
C1X=CM(LVN1+1)/SCALE
C1X=(C1XLMV+1)/SCALE
LEV(K)=2
DO 410 K=KA,KB
IF(LEV(K).GT.LEVI) LEVI=LEV(K)
IF(LEV(K).EQ.LEVE(K))K=K

410 CONTINUE
LVNS=K1=3*LM9+LEV1
C1Z=CZ(LVNS+1)/SCALE
SUBROUTINE DIM5/3

CALL PLOCT(C1X,C2Z,3)
CALL PLOCT(C2X,C2Z,2)
C2Z=C2Z+.375
CALL PLOCT(C1X,C2Z,3)
CALL PLOCT(C2X,C2Z,2)
CALL ARROW(?Z,C2Z,C1X,C2X)
DIMA=0.

DIMA=J.
DO 500 K=KA,KB
LVN=K*LVPL+IDOS(3)+1
LNY=(K-1)*LVPL+IDOS(3)+1
C2Z=C1Z+.5
C2X=C1X/C2Z+1/S

C1X=CV(LVN+1)/SCALE
IF(K.EQ.KP)GO TO 415
C2X=CV(LVNN+1)/SCALE
CALL ARROW(C2Z,C2X,C1X,C2X)
DIMA=1./(BLEN(LVN+1)*U711)

DIMA=DIMA+DIMA
GO TO 420

415 CONTINUE
DIMA=DIMA
C2Z=C2Z+.375
C1X=C1X*DIMA+U711/SCALE

420 CONTINUE

ANGLE=0.
HGT=FACTOR*125
CALL IDIM5(2,K,KB,DIMA,C3Z,C1X,ANGLE,HGT,SCALE)

500 CONTINUE
GO TO 1900

C FLOOR PLAN DIMENSIONS

550 CONTINUE

C FIND GRID LINE FOR X/Y DIR. PER FLOOR

CALL ZER(I1G(1),28)
IGRX1=A*MAX(NCOL(1),NCOL(2))
IGRY1=A*MAX(NCOL(1),NCOL(2))
IGRX2=0
IGRX2=0
PB=F1(1)

IW2=2
IF2=IFR(1)
DO 650 IF=1,IF2
KAA=COL(1,IF,1)
IF(S=1,IF2)KB=COL(1,IF,1)-1
IF(IS=1,IF2)KB=KAA

DO 600 K=KAA,KBB
LVN=(K-1)*LVPL+LNV
IF(C1X(LVN)+EQ.999.,AND.,C1X(LVN)+EQ.999.)GO TO 600
MK1=MCCR(1)
IGRX=IGRID1(MK1)+100
IGRY=GRID1(MK1)+100

600 IF(IGRX.EQ.0.OR.,IGRY.EQ.0)GO TO 600

650 CONTINUE

600 CONTINUE

END
SUBROUTINE DIME/4

IF(IGRX.LT.IGRY)IGRX=IGRY
IF(IGRX. GT.IGRY)IGRX=IGRY

600 CONTINUE
650 CONTINUE

C FIND OUT IF FIRST/LAST COL OF EACH FRAME IS LOCATED
C INSIDE LIMITS, IF IT IS SO CALL GRIDL, OTHERWISE
C DRAW GRID LINES CALLING LINES

DO 700 I=1,2
  KB=MB(I)
  INV=INVER(I)
  IFR=IFR(I)
  DO 700 IFR=1,2
    DO 700 IF=1,IFR
    KB=COL1(IF,1)
    IF(IF.EQ.IFR)KB=COL1(IF,1,1)-1
    IF(IF.EQ.IFR)KB=MB
    DO 680 K=KAA(K)+KB
    IF(IBUG.EQ.53)WRITE(6,(*"DIMSP1="+AI4,*))
      MK1+1=IKRX(I),IGRY(I),IKRX(I),IGRY(I),IKRX(I),IGRY(I)

DO 700 CONTINUE
680 CONTINUE

IF(IT1.EQ.2.KB+KAA>0)GB=2
   LVN=IK1(I)+LVN4+LVV
   IF(IN(LVN+1).LT.999..AND.C(775))G8=999.60 TO 680
   IF(MK1(MK1+KAA+KB.IF.IGRY.EQ.IGRY2)10 TO 655
   IF(IBUG.EQ.53)WRITE(6,(*"DIMSP2="+AI4,*))
     MK1+1=KAA+KB,IKRX(I),IGRY(I),IKRX(I),IGRY(I),IKRX(I),IGRY(I)
     K1=KB
     IF(I.EQ.2)K1=KBB+KAA
     LVN=IK1(I)+LVN4+LVV
     IF(IN(LVN+1).LT.999..AND.C(775))G8=999.60 TO 655
     IF(IBUG.EQ.53)WRITE(6,(*"DIMSP1="+AI4,*))
       MK1+1=KAA+KB,IKRX(I),IGRY(I),IKRX(I),IGRY(I),IKRX(I),IGRY(I)
       K1=KB
       IF(I.EQ.2)K1=KAA
CALL GRIDL(MK1+1,K1,IKRX(I),IGRY(I),LVN4+LVV)

660 CONTINUE

TOTAL=STYM+.5
IF(I.EQ.2)TOTAL=STYM/SCALE+MAR61(-).5
IF(IBUG.EQ.53)WRITE(6,(*"DIMSP11="+AI4,*))
MK1+1=1,IF(IBUG.EQ.53)WRITE(6,(*"DIMSP11="+AI4,*))
660 CONTINUE
CALL LINES42,IKRX(I),IGRY(I),MK1+1,LVV+1,TOTAL,SCALE)
GO TO 700
680 CONTINUE
700 CONTINUE
C DIMENSION LINES TOTAL X/Y DIRECTION.
C DO 750 I=1,2
   HGT=FACTOR1*125
   IF(I.EQ.2)GO TO 720
   ANGLE=25.
   CX1=GRID1(IGRX1,I)/SCALE-1.*5
   CY1=GRID1(IGRY2,2)/SCALE
   CY2=GRID1(IGRY1,2)/SCALE
   DIMEN=(GRID1(IGRY1,2)-GRID1(IGRY2,2))/UZ11
   CALL PLOT(CY1,CY1,3)
   CALL PLOT(CX1,CY2,2)
   CALL ARROW1(CX1,CY1,CY2)
   CALL IDIMS(3,IGRY1,1,IGRY2,2,DIMEN,CX1,CY1,ANGLE,HGT,SCALE)
   CX1=CX1+.375
   CALL PLOT(CX1,CY1,3)
   CALL PLOT(CX1,CY2,2)
   GO TO 750
720 CONTINUE
   ANGLE=0.
   DIMEN=(GRID1(IGRX2,1)-GRID1(IGRX1,1))/UZ11
   CX2=GRID1(IGRX2,1)/SCALE
   CY1=GRID1(IGRY1,2)/SCALE+1.25
   CALL PLOT(CX1,CY1,3)
   CALL PLOT(CX2,CY1,2)
   CALL ARROW2(CY1,CX1,CX2)
   CALL IDIMS(2,IGRY1,1,IGRY2,2,DIMEN,CY1,CX1,ANGLE,HGT,SCALE)
   CY1=CY1+.375
   CALL PLOT(CX1,CY1,3)
   CALL PLOT(CX2,CY1,2)
750 CONTINUE
C DIMENSION LINES BETWEEN GRID LINES AND ARROW HEADS
C DO 800 I=1,2
   MB=MBI(I)
   IFR=IFR2(I)
   IM2=INVER(I)
   DIMEN=0.
   CORD2=0.
   DO 800 IF=1,IFR
   KAA=COL1(IF+1)
   IF(CF1.LT.IFR*KBO=COL1(IF+1,1)-1
   IF(IF.EQ.0.IFR*KBO=NO
   DO 780 K=KAA,KBO
   K1=K
   IF(I.EQ.2)K1=KBO-K+KAA
   LVU=(K1-1)*4+LV+S+LVV
   IF(CX1+LVU,1).EQ.999..AND.CY1+LVU,1).EQ.999..GO TO 780
   PK=CK(K1,1)
   TRVR=GRID1(1,TR1)/100
   IGR1=MOD(GRID1(1,PK)+100)
IF(IGR1.EQ.0.OR.IGRX.EQ.0)GO TO 780
IGRY=MAXH(NCOL(1),NCOL(2))-IFRZ(1)+IGR1
HGT=FACTOR*125
IF(IFR.EQ.1)GO TO 800
IF(I.EQ.2)GO TO 760
IS=3
ID=1
CORD2=CORD(MK,IW2)
IF(IGRY.EQ.IGRY1)GO TO 800
IYY=IGRY-1
755 CONTINUE
IF(IYY.LT.IGRY1)GO TO 800
IF(IGRID(IYY,7).NE.0)GO TO 757
IYY=IYY+1
GO TO 755
757 CONTINUE
CORD1=GRID1(IYY,2)
IGR1=IGRY1
IGR2=IGRY2
IGR=IGRY
ANGLE=90.
CX1=GRID1(IGRX1+1)/SCALE-.875
CY1=CORD2/SCALE
CY2=CORD1/SCALE
GO TO 770
760 CONTINUE
IS=2
ID=2
CORD2=CORD(MK,IW2)
IF(IGRX.EQ.IGRX2)GO TO 800
IXX=IGRX+1
765 CONTINUE
IF(IXX.GT.IGRX2)GO TO 800
IF(IGRID(IXX,1).NE.0)GO TO 767
IXX=IXX+1
767 CONTINUE
CORD1=GRID1(IXX,1)
IGR1=IGRX1
IGR2=IGRX2
IXX=IGRX
ANGLE=9.
CX1=GRID1(IGRY1+2)/SCALE-.875
CY1=CORD2/SCALE
CY2=CORD1/SCALE
770 CONTINUE
IF(IXX.EQ.53)WRITE(6,*("=DIMPS=",614,2F15.3))
MK1=GRID1(IGRY1+1,IGRY2)+CORD1,CORD2
DIXX=(CORD1-CORD2)/UZ11
CALL IDIMP(IXX,0,GRA,DIXX,CX1,CY1,ANGLE,HGT,SCALE)
CALL ABDRMTD(CX1,CY1,CY2)
GO TO 800
780 CONTINUE
800 CONTINUE
IF(IPLOT(16,J)).EQ.0)GO TO 1800
GO 800 1=1,2
IIXX=INVER(I11)
IPIX=IFRZ(IW2)
SUBROUTINE DIS/7 74/74 OPT=1 FTN 5.1+552

IFRI=IFR2(I)
DO 880 ISE=1, IFR1
MGT=FACTOR=.125
IVOR=0
KAS=COL(I,ISE+1)
IF(ISE.LT.IFR1) KBS=COL(I,ISE+1,I)-1
IF(ISE.EQ.IFR1) KBS=MRI(I)
IF(IEQ.2) GO TO 820

DO 815 K=KAS, KBS
LVNC=(K-1)+LVN4+LVV
IF(IPUG.EQ.53) PRINT 990, KAS, KBS, K, I, LVNC, (LVNC+1), C2(LVNC+1)
IF(CX(LVNC+1), EQ.999, AND, C2(LVNC+1), EQ.999) GO TO 815
990 FORMAT(* DIMS-S1=514,2F20.7)
KAS=K

GO TO 816

815 CONTINUE
GO TO 880

816 CONTINUE
MK=MKC(KAS+1)
LX1=LY4(MK)
JX1=JY4(MK)
GO TO 825

820 CONTINUE
DO 822 K=KAS, KBS
K1=KBS+K+KAS
LVNC=(K1-1)+LVN4+LVV
IF(IBUG.EQ.53) PRINT 990, KAS, KBS, K1, I, LVNC, (LVNC+1), C2(LVNC+1)
IF(CX(LVNC+1), EQ.999, AND, C2(LVNC+1), EQ.999,60 TO 822
KBS=K1
GO TO 823

822 CONTINUE
GO TO 880

823 CONTINUE
MK=MKC(KBS+1)
LX1=LX4(MK)
JX1=JX4(MK)

825 CONTINUE
IF(CX(LVNC+1), EQ.999, AND, C2(LVNC+1), EQ.999) GO TO 880
IF(IBUG.EQ.53) PRINT 999, I, MK, LX1, JX1
IF(LX1, EQ.0) GO TO 880
IF(JX1, EQ.0) GO TO 858
IF(SPAI(LX1, JX1-1), LE.0) GO TO 850

C COL. IN MIDDLE OF BR. LINE

C DO 830 ISE=1, IFR3
KASI=COL(I,ISE+1,IV2)
IF(ISE.LT.IFR3) KBS1=COL(I,ISE+1,IV2)-1
IF(ISE.EQ.IFR3) KBS1=M01(IV2)
GO 827 K=KAS1, KBS1
MK=MKC(K, IV2)
IF(MK, EQ. MK1) GO TO 834

827 CONTINUE

830 CONTINUE
GO TO 880

834 CONTINUE
MK=MK-1
SUBROUTINE DIM5/N
74/74
OPT=1
FTN 5.1/552
84

MK2=MKC(NK1,1M2)
IF(MK2.EQ.MK3)GO TO 850

LX3=LX(MK3)
JX3=JX(MK3)
IF(SSTOP(XJ3,JX3).LE.0.)GO TO 880
LX2=LX1
JX2=JX1-1
GO TO 850

850 CONTINUE
IF(SSTOP(LX1,JX1).LE.0.)GO TO 880
LX2=LX1
JX2=JX1
MK2=MK

860 CONTINUE
DBM=DBM+0.
VAL=0.
IF(IPLT(J,L).EQ.1)VAL=.052
DO 870 KN=1,8
MKT=FACTOR*125
I,J,L,R=P(LX2,JX2,KN)
IF(DBG.EQ.53)PRINT 999,1,MK,LX2,JX2,I,J,L,MK2,MK3
999 FORMAT(* DIMS=5,0,T15)
IF(I,J,L,.AND.MK,EQ.MK2)GO TO 875
IF(I,J,L,.EQ.160)GO TO 850
DBM=FDBM(LX2,JX2,KN)/12.
L1=L1/I,J,L/100
IS1=I,J,L-L1*100
LI2=LX(MK)
JS2=JX(MK)
IF(J,J,.EQ.2)LI2=LX(MK)
IF(J,J,.EQ.2)IS2=JX(MK)
SP2=SPAN(L12,IS2)/12.
SP1=SPAN(LI1,IS1)/12.
SP=SP1-SP2
IF(SP2.EQ.SP1)SP=0.
IF(SP.EQ.0.)SP=DBM
SIM1=SP/DBM
ALF=ASIN(SIM1)
COS1=ASIN(SIM1)
IF(DBG.EQ.53)WRITE(6,*('DIM5=660=15.6F10.3'))
#MK1*LI1*IS1*LX2,JX2,LI2,IS2,SP1,SP2,SP,SP,SP,DBM,SIM1,COS1
IF(I,J,.EQ.2)GO TO 865
DBM=DBM+DBM
CX1=CORD(MK2,1)/SCALE-.5
CY1=CORD(MK2,1)/SCALE+DBM1/SCALE
DBM1=DBM1+DBM
CX2=CORD(MK2,1)/SCALE+DBM1/SCALE
CY2=CORD(MK2,1)/SCALE+DBM1/SCALE
CX3=CX2-.5

T14=1
ID=1
ANGLE=90.
CALL PLOT(CX1,CY1,3)
CALL PLOT(CX1,CY2,2)
CALL ARROW(ID,CX1,CY1,CY2)
CALL PLOT(CX2,CY2,3)
CALL PLOT(CX3,CY2,2)
```plaintext
SUBROUTINE DINS /9
74/74 OPT=1
FTN 5.1.552

HGT=FACTOR*1.25
DBM2=DBM/U211

CALL IDIMS(IS,IVOR,MK,DRM2,CXI,CY1,ANGLE,HGT,SCALE)
GO TO 870

869 CONTINUE
DRM=DBM*SIN1
CXI=CORD(MK,1W2)/SCALE*DBM1/SCALE
CY1=CORD(MK2,1)/SCALE*5

DBM1=DRM*DRM1
CX2=CORD(MK2,1W2)/SCALE*DBM1/SCALE
CY2=CORD(MK2,1)/SCALE*125*VAL
CY3=CY2*5
ID=2
IS=4

ANGLE=0.
CALL PLOT(CXI,CY1,3)
CALL PLOT(CX2,CY2,3)
CALL ARROW(ID,CY1,CX1,CX2)
CALL PLOT(CX2,CY3,2)

HGT=FACTOR*1.25
DBM2=DBM/U211
CALL IDIMS(IS,IVOR,MK,DRM2,CY1,CX1,ANGLE,HGT,SCALE)

870 CONTINUE
875 CONTINUE
IF(MK.ER.MK2)MK3=MK2
IF(MK.EQ.MK2)GO TO 880
GO TO 850

880 CONTINUE
1000 CONTINUE
RETURN
END

---
LE MAP--(LO=A)
-ADDRESS--BLOCK-----PROPERTIES------TYPE------SIZE-----NAME---ADDRESS---BLOCK---

<table>
<thead>
<tr>
<th>3847B</th>
<th>REAL</th>
<th>CY2</th>
<th>277B8</th>
</tr>
</thead>
<tbody>
<tr>
<td>2723B</td>
<td>REAL</td>
<td>CY3</td>
<td>3053B</td>
</tr>
<tr>
<td>3650B</td>
<td>REAL</td>
<td>196B</td>
<td>C2</td>
</tr>
<tr>
<td>642B</td>
<td>REAL</td>
<td>20</td>
<td>C1</td>
</tr>
<tr>
<td>6B</td>
<td>REAL</td>
<td>1960</td>
<td>C1X</td>
</tr>
<tr>
<td>27010B</td>
<td>REAL</td>
<td>1984</td>
<td>C1Z</td>
</tr>
<tr>
<td>2367B</td>
<td>/MFL4/</td>
<td>90</td>
<td>C11</td>
</tr>
<tr>
<td>423B</td>
<td>/</td>
<td>2</td>
<td>C2X</td>
</tr>
<tr>
<td>2712B</td>
<td>/</td>
<td>271B</td>
<td></td>
</tr>
<tr>
<td>498B</td>
<td>/MFL4/</td>
<td>280</td>
<td>22</td>
</tr>
<tr>
<td>500B</td>
<td>/MFL4/</td>
<td>280</td>
<td>C3Z</td>
</tr>
<tr>
<td>3009B</td>
<td>/</td>
<td>DAT</td>
<td>950</td>
</tr>
<tr>
<td>2772B</td>
<td>/</td>
<td>DBM</td>
<td>3032B</td>
</tr>
<tr>
<td>3053B</td>
<td>/</td>
<td>DBM1</td>
<td>3032B</td>
</tr>
<tr>
<td>20110B</td>
<td>/PLOT1/</td>
<td>DBM2</td>
<td>3052B</td>
</tr>
<tr>
<td>32710B</td>
<td>/PLOT1/</td>
<td>DIMN</td>
<td>2721B</td>
</tr>
<tr>
<td>2766B</td>
<td>/</td>
<td>DIMN1</td>
<td>2737B</td>
</tr>
<tr>
<td>2771B</td>
<td>/</td>
<td>DUMMY</td>
<td>3050B /*MFL4/</td>
</tr>
<tr>
<td>3481B</td>
<td>/</td>
<td>DV</td>
<td>4612B /PLOT1/</td>
</tr>
<tr>
<td>2767B</td>
<td>/</td>
<td>EC1</td>
<td>115R</td>
</tr>
</tbody>
</table>
```
GO TO 390
100 ENCODE(5, 321, BIMSH(LM, JN)) IWD, ISGN, IDEP
   GO TO 390
110 ENCODE(5, 322, BIMSH(LM, JN)) IWD, ISGN, IDEP
   GO TO 390
120 ENCODE(6, 323, BIMSH(LM, JN)) IWD, ISGN, IDEP
   GO TO 390
130 ENCODE(7, 324, BIMSH(LM, JN)) IWD, ISGN, IDEP
   GO TO 390
140 ENCODE(5, 331, BIMSH(LM, JN)) IWD, ISGN, IDEP
   GO TO 390
150 ENCODE(6, 332, BIMSH(LM, JN)) IWD, ISGN, IDEP
   GO TO 390
160 ENCODE(7, 333, BIMSH(LM, JN)) IWD, ISGN, IDEP
   GO TO 390
170 ENCODE(7, 343, BIMSH(LM, JN)) IWD, ISGN, IDEP
   GO TO 390
180 ENCODE(8, 342, BIMSH(LM, JN)) IWD, ISGN, IDEP
   GO TO 390
190 ENCODE(9, 344, BIMSH(LM, JN)) IWD, ISGN, IDEP
   GO TO 390
200 ENCODE(9, 344, BIMSH(LM, JN)) IWD, ISGN, IDEP
300 CONTINUE
   IF (IRUG.EQ.53) PRINT 999, LM, JN, IWD, IDEP, LETH, LETD, BIMSH(LM, JN)
310 CONTINUE
350 CONTINUE

C
311 FORMAT(11, A1, I1)
312 FORMAT(11, A1, I2)
313 FORMAT(11, A1, I3)
321 FORMAT(12, A1, I1)
322 FORMAT(12, A1, I2)
323 FORMAT(12, A1, I3)
324 FORMAT(12, A1, I4)
331 FORMAT(13, A1, I1)
332 FORMAT(13, A1, I2)
333 FORMAT(13, A1, I3)
334 FORMAT(13, A1, I4)
342 FORMAT(14, A1, I2)
343 FORMAT(14, A1, I3)
344 FORMAT(14, A1, I4)
999 FORMAT(* ENCO *, 616, IX, A10)

C
RETURN
END

LE MAP -- (LO=A)
-ADDRESS--BLOCK-----PROPERTIES------TYPE--------SIZE------NAME-----ADDRESS--BLOCK--
  50590 /BMFL4/ EVV    REAL     240    DEPTH   1130B /BMFL4/
  1773B /THETF1/ EVV    REAL     600    DUM      0B /THETF1/
  17500 /THETF1/ EVV    REAL      21    DUMMY    50580 /BMFL4/
  6420 //              REAL      20    DUM      0B /BMFL4/
  4233 //              REAL      2    ECl     1158 //
  55B //              REAL      3    EC9     1160 //
SUBROUTINE ENCO1

COMMON MAT, A, R, EY1, ISQY, IVFIN, LMAX, NL, IFLEV, IFFRP
COMMON ZZZZ, X, Y, DAT(3), TDC(6), IDOS(6), DAT, ICRIS(6), ICRI(6)
COMMON JSTOP, NSTOP, IPRINT, LNK, EC1, EC5, MAXFR, ITEST, IRUN, LVU, LNU
COMMON X, Y, LIST(10), IO4, CODE(2), MD(20), UF(20)
COMMON TTI, TJ, NEZ(90)
COMMON USF, BOM(20), LON(20)
COMMON TAK(10), JFET(10), KFET(20), LET(20)

INTEGER KAM(40, 21), NME(11, 3)
COMMON/RFZ1/KUZ1, UZ2, UZ3, UZ4, UZ5, UZ6, UZ7, UZ8, UZ9, UZ10, UZ11
UZ12, UZ13, UZ14, UZ15, UZ16, UZ17, UZ18
COMMON/RGFL/WIDTH(30, 20), DEPTH(30, 20)
COMMON/RGFL/SPAN(30, 20), XBM(30, 20, 4), YBM(30, 20, 4), JMX(30)
COMMON/THETF/NUM(1690)

COMMON/RF/NUM(12600), DUMMY(240)

DIMENSION IFAB(250), X_min(250), X_max(250), X_minh(250), X_maxh(250)

EQUIVALENCE (DUM(1), IFAB(1)), (DUM(1), X_min(1)), (DUM(1), X_max(1))
EQUIVALENCE (DUM(1), X_min(1)), (DUM(1), X_max(1))
EQUIVALENCE (DUM(1), X_min(1)), (DUM(1), X_max(1))

LET1 = LET1
CALL ZERO(PIMSH, 600)
ISIGN = 0
DO 350 J = 1, LET1

LET1 = LET1

JMJX = JMJX(LMJ) - 1
DO 310 JMJ = 1, JMJX

IDIW = MINWLM(JMJ), UZ10 + 0.01
IDEP = DEPTLM(JMJ), UZ10 + 0.01
LET1 = LET1

IF(IDIW > 9.0 AND IDIW <= 99) LET1 = 2
IF(IDIW > 99.0 AND IDIW <= 999) LET1 = 3
IF(IDIW > 999.0 AND IDIW <= 9999) LET1 = 4
IF(IDEP > 9.0 AND IDEP <= 99) LET1 = 2
IF(IDEP > 99.0 AND IDEP <= 999) LET1 = 3
IF(IDEP > 999.0 AND IDEP <= 9999) LET1 = 4

GO TO 20, 30, 40, 50, 60, LET1

CALL GOTOER

20 CONTINUE
GO TO 60, 70, 80, LET1

CALL GOTOER

30 CONTINUE
GO TO 100, 110, 120, 130, LETD

CALL GOTOER

40 CONTINUE
GO TO 150, 160, 170, LETD

CALL GOTOER

50 CONTINUE
GO TO 180, 180, 190, 200, LETD

CALL GOTOER

60 ENCOD13, 31, 31, PIMSH(LMJ, JMJ), IDIW, ISIGN, IDEP
GO TO 300

70 ENCOD14, 312, PIMSH(LMJ, JMJ), IDIW, ISIGN, IDEP
GO TO 300

80 ENCOD15, 313, PIMSH(LMJ, JMJ), IDIW, ISIGN, IDEP
Common MAT, MEM, LEV, ISOLV, LVFIN, LMFIN, LVMAX, LNM, IFLEV, IFFRM
COMMON ZZZZ, NME, DAT(3), IDOC(6), IDOS(6), IDAT, ICRI(6), ICRRC(6)
COMMON JSTOP, NSTOP, PRINT, ILINK, ECI, ECV, MAY, TEST, IPUNL, LVUK, LNU
COMMON MAF, LIST(16), 109, CODE(2), MN(20), UF(20)
COMMON YTC, MEZ(90)
COMMON USF(10), BON(20), BDN(20)
COMMON TAK(10), JET(10), KET(20), LET(20)
INTEGER MAF, SIZX, SIZZ (10)
COMMON PLOT, PSIF(9850), EMEN(980, 2), MBR(980, 2), KST(980, 2)
1 (9800), CSTIF(992, 2), CLEN(992, 2), W(992, 2), C7(992, 2)
2, INWAY(12), PMIC(992, 2), DV(992, 2), TRANS(90, 2), C11(2), C22(2)
3, JFR(980, 2)
COMMON BFL, STPX, STPY, STPY, MB1(2), MP1(2), IFR(140, 2)
1, COL1, ICFL(140), IFBM, GRID1(140, 2), NCOL(2), MKC(140, 2)
4, CLEV(90), IBUG, STZ1, STZ2, EFLAG(140), EFLAG(140), SSIKPL(140)
5, SIDE(10), MARG, PLOT(25, 15), JPL, IGRID1(140), IGRID2(140, 2)
6, DUMMY(240)
INTEGER COL1(140, 2)
DIMENSION MDIM(30), BDIM(30)
REAL MARG(10)
DATA MARG/5*2, 1, 75, 1, 45, 2, 25, 2, 5/
SIZX = SIZX1 = SIZZ = SIZZ1 = 0
SIDE3 = SIDE(IPS) = MARG(4)
SIDE4 = SIDE(IPS+1) = MARG(5) - 4.5
IF (IPLA.EQ.1) GOTO 160
DO 10 IPP = MAF, KMB
LNX = (IP = 1) * LVMA + IDOS(3) + 1
LNY = IPP - LVMA
LNZ = (IP = 1) * LVMA + 1
IF (CXL(LNX, IV) .GT. SIZZ) SIZX = CZ(LNX, IV)
IF (CXL(LNX, IV) .LE. 1999) AND (CZ(LNY, IV) .EQ. 1999) GOTO 10
IF (CXL(LNY, IV) .GT. SIZZ) SIZZ = CZ(LNY, IV)
10 CONTINUE
SIZX = SIZX / SCALE
SIZZ1 = SIZZ1 / SCALE
SIZZ = SIZZ / SCALE
IF (IBUG .EQ. 53)
*PRINT 991, ICONT, MCONT, KALN, SIZX1, SIZX, SIZZ1, SIZZ,
*STZM, ORGK, ORGY
991 FORMAT(12, ** DEBUG LOCATION 1 **, 4I4, 7F12.2)
IF (MCONT .NE. 998) GOTO 12
MCOUNT = 0
60 TO 15
12 CONTINUE
IF (ICOUNT .LT. 1) GOTO 30
15 CONTINUE
ADIM (ICOUNT) = SIZX + MARG(1) + MARG(6)
BDIM (ICOUNT) = SIZZ / SCALE - STZM + MARG(2) + MARG(8)
IF (ADIM (ICOUNT) .LT. SIDE4) GOTO 20
IF (BDIM (ICOUNT) .LT. SIDE3) GOTO 20
ORGK = MARG(1) + MARG(6)
ORGY = SIDE(IPS) - MARG(2) - MARG(8) - SIZZ / SCALE
KCONT = ICOUNT
SIZEX = SIZX
IF(MCOUNT.NE.998)SIZEX1=SIZEX

ORGX2=0.

GO TO 80

20 CONTINUE

C PAPER SIZE TOO SMALL

MCOUNT=199

RETURN

30 CONTINUE

C OTHER FRAMING ELEV. SIDE BY SIDE

ORGY2=0.ORGX2+ORGX

ADIM(IHCOUNT)=ADIM(IHCOUNT-1)+MARG(1)+MARG(6)+SIZE

BDIM(IHCOUNT)=MARG(2)+MARG(8)+SIZE-STZMN

IF(ADIM(IHCOUNT).GT. Side4) GO TO 40

IF(BDIM(IHCOUNT).GT. Side8) GO TO 50

C SET NEW ORIGIN FOR FRAMING ELEV.

ORGY=SIZEY+MARG(1)+MARG(6)

ORGY=SIDE4+SIZEY+STZMN-(MARG(2)+MARG(8))-ORGY

ORGY=0.

SIZEY=SIZEX

MCOUNT=0

GO TO 80

C TRY TO LOCATE BELOW

40 CONTINUE

ADIM(IHCOUNT)=MARG(1)+MARG(6)+SIZE

IF(ADIM(IHCOUNT).GT. Side4) GO TO 50

BDIM(IHCOUNT)=BDIM(KCOUNT)+BDIM(IHCOUNT)

IF(BDIM(IHCOUNT).GT. Side3) GO TO 50

KCOUNT=IHCOUNT

ORGX=MARG(1)+MARG(6)-ORGX2

ORGY=-(MARG(2)+MARG(8)+STZ/SCALE-STZMN)

SIZEY=SIZEX

MCOUNT=0

GO TO 80

50 CONTINUE

C NEW PAGE

MCOUNT=998

GO TO 15

80 CONTINUE

ADIM1=ADIM(IHCOUNT)

BDIM1=BDIM(IHCOUNT)

IF(MCOUNT.NE.998) GO TO 85

ADIM1=ADIM(IHCOUNT-1)

BDIM1=BDIM(IHCOUNT-1)

85 CONTINUE

XLEFT=SIDE(IPS+1)-ADIM1-4.5

YLEFT=SIDE(IPS)-BDIM1

IF(MCOUNT.NE.998)SIZEX1=SIZEX
ORG Y1=XLEFT+SIZEX1+4.5  

ORG Y1=ST2MN-YLEFT  

IF (TPUE.EQ.55)  
1PRINT 990,ICOUNT,MOUNT,KCOUNT,ADIM(ICOUNT),ADIM(ICOUNT),  
2ST2MN,ORGX,ORGY,ORGX1,ORGY1,ORGX2,ORGY2,XLEFT,YLEFT  
990 FORMAT(2** DEBUG LOCATION **,3I4,10F8.2)  
RETURN  

100 CONTINUE  

C  
FLOOR PLANS  
C  

SPX=SPX1=SPY=SPY1=0.  
DO 110 II=1,2  
MV=MK1(II)  
IFR=IFR2(II)  
DO 110 III=1,IFR  
KA1=COL1(III+II)  
IF (III.LT.IFR) KB1=COL1(III+III+II)-1  
IF (III.EQ.IFR) KB1=MA  
DO 110 KK=KA1+KB1  
MK=MK1(KK,II)  
LVRI=KA  
LN=(KK-1)*LVMA+LVRI-1.  
IF (CFL(MK),NE.916) GO TO 105  
IF (CFL1(LN,II).EQ.999) AND (.CZ(LN,II).EQ.999) GO TO 110  

105 CONTINUE  
IF (CORD(MK,1).LT.SPX1) SPX=CORD(MK,1)  
IF (CORD(MK,1).GT.SPX1) SPY=CORD(MK,1)  
IF (CORD(MK,2).LT.SPY1) SPX=CORD(MK,2)  
IF (CORD(MK,2).GT.SPY1) SPY=CORD(MK,2)  
110 CONTINUE  

SPX=SPX/SCALE  
SPX1=SPY1/SCALE  
SPY=SPY/SCALE  
SPY1=SPY1/SCALE  
IF (TPUE.EQ.53)  
*PRINT 991,ICOUNT,MOUNT,LVRI,LVMA,SPX,SPX1,SPY1,SPY,  
*SPX,ORGX,ORGY  
IF (MCOUNT.NE.998) GO TO 112  
MCOUNT=0  
GO TO 115  

112 CONTINUE  
IF (ICOUNT.EQ.1) GO TO 130  
115 CONTINUE  

ADIM(ICOUNT)=SPX1-SPX+MARG(9)+MARG(1)  
BDIM(ICOUNT)=SPY1-SPY+MARG(10)+MARG(2)  
IF (ADIM(ICOUNT).GT.SIDE4) GO TO 120  
IF (BDIM(ICOUNT).GT.SIDE3) GO TO 120  
ORGX=MARG(9)+MARG(11)-SPX  
ORGY=SIDE4-IP51-MARG(10)-SPY1-MARG(2)  
KCOUNT=ICOUNT  
SIZEX=SPX1-SPX  
IF (MCOUNT.NE.998) SIZEX1=SIZEX  
ORGY2=0.  
GO TO 180  

120 CONTINUE  

C
C C PAPER SIZE TOO SMALL
   MCOUNT=999
   RETURN
130 CONTINUE
C C OTHER FRAMING PLAN SIDE BY SIDE
C ORGY=0
SIZEX=SPX1-SPX
MCOUNT=7
GO TO 140
C C TRY TO LOCATE BELOW
C 140 CONTINUE
ADIM(ICOUNT)=MARG(9)+MARG(1)+SPX1-SPX
IF ADIM(ICOUNT).GT.SIDE4 GO TO 150
BDIM(ICOUNT)=BDIM(KCOUNT)+BDIM(ICOUNT)
IF FRDIM(ICOUNT).GT.SIDE3 GO TO 150
KCOUNT=ICOUNT
ORGX=MARG(9)+MARG(1)-ORGX2
ORY=(MARG(10)+MARG(2)+(STPY-STPY)/SCALE)-1.
SIZEY=SPX1-SPY
MCOUNT=0
GO TO 160
150 CONTINUE
C C NEW PAGE
C MCOUNT=999
GO TO 115
180 CONTINUE
ADIM1=ADIM(ICOUNT)
BDIM1=BDIM(ICOUNT)
IF MCOUNT.LE.998 GO TO 185
ADIM1=ADIM(ICOUNT-1)
BDIM1=BDIM(ICOUNT-1)
185 CONTINUE
XLEFT=SIDE1IPS1-ADIM1*SPX-4.5
YLEFT=SIDE1IPS1-BDIM1
IF MCOUNT.LE.998 SIZEX1=SIZEX
ORGX=XLEFT+SIZEX1+4.5
ORY=STPY/SCALE-YLEFT
IF (DEBUG.EQ.53)
IPRINT=999
MCOUNT=MCOUNT+KCOUNT+ADIM(ICOUNT)+BDIM(ICOUNT)
2STPY/SCALE,ORGX,ORY,ORGX1,ORY1,ORGX2,XLEFT,YLEFT
RETURN
<table>
<thead>
<tr>
<th>NAME</th>
<th>ADDRESS</th>
<th>BLOCK</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRUN</td>
<td>121R</td>
<td></td>
</tr>
<tr>
<td>ISOVL</td>
<td>3R</td>
<td></td>
</tr>
<tr>
<td>ITCI</td>
<td>475P</td>
<td></td>
</tr>
<tr>
<td>ITEST</td>
<td>120B</td>
<td></td>
</tr>
<tr>
<td>IV</td>
<td>6</td>
<td>DUMMY-AF</td>
</tr>
<tr>
<td>JBR</td>
<td>52602P</td>
<td>/PLOT1/</td>
</tr>
<tr>
<td>JFT</td>
<td>724R</td>
<td></td>
</tr>
<tr>
<td>JFL</td>
<td>42038</td>
<td>/BMFL4/</td>
</tr>
<tr>
<td>JSTOD</td>
<td>111B</td>
<td></td>
</tr>
<tr>
<td>KA</td>
<td>4</td>
<td>DUMMY-AP</td>
</tr>
<tr>
<td>KAK</td>
<td>124B</td>
<td></td>
</tr>
<tr>
<td>KAI</td>
<td>1017P</td>
<td></td>
</tr>
<tr>
<td>KB</td>
<td>5</td>
<td>DUMMY-AF</td>
</tr>
<tr>
<td>KPI</td>
<td>1028A</td>
<td></td>
</tr>
<tr>
<td>KCOUNT</td>
<td>777B</td>
<td></td>
</tr>
<tr>
<td>KET</td>
<td>736B</td>
<td></td>
</tr>
<tr>
<td>KK</td>
<td>1021B</td>
<td></td>
</tr>
<tr>
<td>KKPLT</td>
<td>3154P</td>
<td>/BMFL4/</td>
</tr>
<tr>
<td>KST</td>
<td>13370R</td>
<td>/PLOT1/</td>
</tr>
<tr>
<td>LFT</td>
<td>762B</td>
<td></td>
</tr>
<tr>
<td>LEV</td>
<td>2B</td>
<td></td>
</tr>
<tr>
<td>LIST</td>
<td>410B</td>
<td></td>
</tr>
<tr>
<td>LN</td>
<td>774B</td>
<td></td>
</tr>
<tr>
<td>LNFN</td>
<td>5B</td>
<td></td>
</tr>
<tr>
<td>LMAX</td>
<td>7B</td>
<td></td>
</tr>
<tr>
<td>LN</td>
<td>123B</td>
<td>DUMMY-AP</td>
</tr>
<tr>
<td>LNR1</td>
<td>777B</td>
<td></td>
</tr>
<tr>
<td>LNR2</td>
<td>776B</td>
<td></td>
</tr>
<tr>
<td>LON</td>
<td>666B</td>
<td></td>
</tr>
<tr>
<td>LVFIN</td>
<td>4B</td>
<td></td>
</tr>
<tr>
<td>LVNAX</td>
<td>6B</td>
<td></td>
</tr>
<tr>
<td>LV4</td>
<td>7</td>
<td>DUMMY-AP</td>
</tr>
<tr>
<td>LV1</td>
<td>1023B</td>
<td></td>
</tr>
<tr>
<td>LV2</td>
<td>1228</td>
<td></td>
</tr>
<tr>
<td>MARG</td>
<td>3402B</td>
<td>/BMFL4/</td>
</tr>
<tr>
<td>MAX</td>
<td>0B</td>
<td></td>
</tr>
<tr>
<td>MAXF</td>
<td>117R</td>
<td></td>
</tr>
<tr>
<td>MP</td>
<td>1014B</td>
<td></td>
</tr>
<tr>
<td>MR1</td>
<td>4B</td>
<td>/BMFL4/</td>
</tr>
<tr>
<td>MCOUNT</td>
<td>3</td>
<td>DUMMY-AR</td>
</tr>
<tr>
<td>MR</td>
<td>425B</td>
<td></td>
</tr>
<tr>
<td>MEF</td>
<td>1B</td>
<td></td>
</tr>
<tr>
<td>MK</td>
<td>1822B</td>
<td></td>
</tr>
<tr>
<td>MR</td>
<td>7520B</td>
<td>/PLOT1/</td>
</tr>
<tr>
<td>MKC</td>
<td>1737B</td>
<td>/BMFL4/</td>
</tr>
<tr>
<td>NCOL</td>
<td>1735B</td>
<td>/BMFL4/</td>
</tr>
<tr>
<td>MEZ</td>
<td>476e</td>
<td></td>
</tr>
<tr>
<td>WL</td>
<td>1B</td>
<td></td>
</tr>
<tr>
<td>WME</td>
<td>14B</td>
<td></td>
</tr>
<tr>
<td>WSTOP</td>
<td>112B</td>
<td></td>
</tr>
<tr>
<td>OREG</td>
<td>9</td>
<td>DUMMY-AP</td>
</tr>
</tbody>
</table>
SUBROUTINE GRIOL(MK, IW, K, I1, IGR3, IGR, LVV, LVM)
COMMON M, MK, LEV, ISOLV, LVFIN, LNFIN, LVMAX, NV, IFLEV, IFRR
COMMON ZZZZ, NME, DAT(5), IDOC(6), IDOS(6), IDAT, ICRIS(6), ICRIC(6)
COMMON JSSTOP, NSTOP, IPRT, ILINK, EC1, EC5, MAXFR, ITEST, IPRUN, LVU, LNU
COMMON KAK, LIST(10), QOF(2), MD(20), UF(20)
COMMON ITC, NE2(90)
COMMON HSF(10), BON(20), LCN(20)
COMMON XH(10), FT(10), KF(20), LET(20)

INTEGER KAK(90, 1), MME(11, 3)
COMMON PLOT1/ESTIF(980, 2), RLEN(980, 2), WKR(980, 2), XST(980, 2),
  IIFR(980, 2), CSTF(920, 2), CLE'(920, 2), MX(920, 2), CZ(920, 2),
  ZINWAV(2), PMIC(920, 2), OVR(920, 2), TRANS(90, 2), C11(2), C22(2),
  3JFR(980, 2)
COMMON CLFL2/LX(140), JX(140), LY(140), JY(140), CLFL(140, 9)
COMMON BMFL2/SPAN(50, 20), MBM(30, 20, 4), LRM(30, 20, 4), JMY(30)
COMMON BMFL4/STP(50, 20, 4), STYP(50, 20, 4), MB(12), IFR(2), CORD(140, 2), COL1
  1, ICF(140), IFEM(2, GRD(140, 2), WCOL(2), WC(140, 2),
  4, CLE(90), IBUSG, STZ(1), STZ(1), IFLAG(140), IFLAG(140), JKFL(140)
  5, SIDE(10), MARG, IPLT(25, 15), JPL, JLRD(140), JGRID(140, 2)
REAL MARG(10)

INTEGER COL1(140, 2)

C
C
C
IGR3 = IGR
MKX = MK
IW2 = IVER(IW)
MB = MB1(IW)
IFR = IFR2(IW2)
IFR1 = IFR2(IW2)
KA = K
KAX = K1
JX1 = JY(MK)
IF (IW < EO. 1) GO TO 160
KA = K1
KB = K
JX1 = JX(MK)
10 CONTINUE
IF (JX1.EQ.1) GO TO 150
110 CONTINUE
DO 150 IF = 1, IFR1
KA = COL1(IF, IW2)
IF (IF.LT. IFR1) KB = COL1(IF + 1, IW2) - 1
IF (IF.LT. IFR1) KB = MB1(IW2)
DO 120 K1 = KA, KB
MK1 = MNC1(K1, IW2)
IF (MK1 .EQ. KA) GO TO 120
IF (MK1 .NE. MKX) GO TO 120
K2 = K1 - 1
MK = MNC2(K2, IW2)
IF (IF .EQ. 140)
GO TO 140
120 CONTINUE
130 CONTINUE
140 CONTINUE
DO 145 IF = 1, IFR1
KA = COL1(IF, IW2)
SUBROUTINE GRIDL1/2

DO 145 K3=KAA+KKB
MK3=MKC(K3,1W)
IF(MK2.NE.MK3)GO TO 145
XX1=KAA
XX=XX1

IF(IW.EQ.2)MX1=K3
IF(IW.EQ.2)XX=KBB
GO TO 146

145 CONTINUE

146 CONTINUE

DO 148 MX1=XX1,XX
K2=K1

IF(IW.EQ.2)MX2=MX1+XX1
LVN=1K2-1)+LM+LVV
IF(CX1LVN+1W).EQ.999.AND.PZ(LVN+1W).EQ.999)GO TO 148
MK2=MKC(K2,1W)
LX1=LX(MK2)
JX1=JX(MK2)

IF(IW.EQ.1)LY1=LY(MK2)
IF(IW.EQ.1)JY1=JY(MK2)
IF(JX1.NE.0.AND.SPAN(LX1+JX1)+LE.0)GO TO 148
IF(K2.EQ.MKX.AND.IW.EQ.1)GO TO 149
IF(K2.EQ.MKX.AND.IW.EQ.2)GO TO 149
IF(IW.EQ.1)IGRY=IGRID1(MK2)/100
IGR1=MOD(IGRID1(MK2),100)
IF(IER1.EQ.0)GO TO 148
IF(IW.EQ.2)IGRY=MAXD(MCOL(1),MCOL(2))-.IFR2(1)+IGR1
GO TO 148

148 CONTINUE

149 CONTINUE

MKX=MK2
IF(IIF1.EQ.2)GO TO 300
GO TO 110

150 CONTINUE

DO 200 K=KAA,KB
K1=K

IF(IW.EQ.2)K1=K+KAA
LVN=(K1-1)+LM+LVV
IF(CX1LVN+1W).EQ.999.AND.PZ(LVN+1W).EQ.999)GO TO 200
IF(K1.EQ.KA.AND.IW.EQ.1)GO TO 200
IF(K1.EQ.KB.AND.IW.EQ.2)GO TO 200
MKX=MKC(K1,1W)
JX1=JX(MKX)
IF(IW.EQ.1)JX1=JY(MKX)
IF(JX1.EQ.1)GO TO 200
GO TO 110

200 CONTINUE

300 CONTINUE

RETURN
END
SUBROUTINE GRID5(LVM*)
COMMON MAT, MP, LEV, ISOL, LVP, IN, LINF, LVP, MAX, NL, IFLEV, IFMR
COMMON ZZZZ, NME, DAT(5), IDOC(6), IDCS(6), ICPP(5), ICPRC(6)
COMMON JSTOP, NSTOP, IPRINT, LIN, EC, EC5, MAXFRI, ITEST, IRUN, LVP, LNU
COMMON KAK, LLIST(10), I09, CODF(2), MD(20), UF(20)
COMMON ITCI, NE7(90)
COMMON USF(10), BON(20), LON(20)
COMMON TAK(110), IFT(10), KET(20), LET(20)

INTEGER KAK(20, 2), NME(11, 3)
COMMON/PLT1/ PSTIF(90, 2), BLEN(90, 2), MOP(90, 2), KST(90, 2)
COMMON/STP1/ STIF(90, 2), CLEN(90, 2), CY(90, 2), C2(90, 2)
COMMON/SPX/ STP1(1, 2), STPY(1, 2), MB1(2), IFP2(2), CORD(140, 2)
COMMON/COL/ IF1(140), IFBM, GRID1(140, 2), MC1(140, 2)
COMMON/CLE1/ IRUE(2), ISR(1, 2), STZZI, STZZ, IFLAG(140), IFLAG(140), KFLAG(20)
COMMON/SIDE/ SS(10), IARG(2), IPILOT(25, 15), JPL, IGRID1(140), GRID(140, 2)

REAL MARG(10)

CALL ZERO(GRID1, 122)
CALL ZERO(GRID1, 20)

DO 100 LW = 1, 2

MAX = MB1(TW)
IFRI = IFR2(TW)
IW2 = INVER(TW)
MAX1 = MB1(IW2)
IFR3 = IFR3(IW2)

DO 100 IS = 1, IFP1

KA = COL1(IS + IW)
IF(IS + L1 + I) KB = COL1(IS + 1, IW) - 1
IF(IS + EQ + I) KB = MAX
KFL & EQ = 0

DO 100  K = KA, KP

100 RETURN

DO 100 IS = 1, IFR3
KA = COL1(IS + IW2)
IF(IS + LT + I) KB1 = COL1(IS + 1, IW2) - 1
IF(IS + EQ + I) KB1 = MAX

DO 60 K = KA, KB

IF(MK = EQ = MK1) GO TO 60
MK1 = MKC1(K + IW)
IFLA1 = 0

DO 60 IS1 = 1, IFR3
KA = COL1(IS1 + IW2)
IF(IS1 + LT + I) KB1 = COL1(IS1 + 1, IW2) - 1
IF(IS1 + EQ + I) KB1 = MAX

DO 60 K1 = KA, KB1
MK1 = MKC1(K1 + IW2)
IF(MK = EQ = MK1) GO TO 60

60 CONTINUE

IFLA1 = 1

70 CONTINUE

IF(IFLA1 = EQ = 1) GRID(MK, IW) = 7777, 777
IF(IFLA1 = EQ = 1) GRID(MK, IW2) = 7777, 777
IF(GRID(MK, IW) = EQ = 7777, 777) GO TO 90
IF(GRID(MK, IW2) = EQ = 7777, 777) GO TO 90
IF(KFLAG = EQ = 1) GO TO 80
GRID(MK, IW) = CORD(MK, IW)
GRID(MK, IW2) = CORD(MK, IW2)
SUBROUTINE GRIDS/2

KFLAG=1
GO TO 80

80 CONTINUE
GRID(MK,IW2)=77777.777
GRID(MK,IW)=CORD(MK,IW)

90 CONTINUE
KFLAG=1
IF(IPEG.EQ.53)PRINT 997,IW,IS,MK,K,KFLAG,CSTIF(LYN,IW).
GRID(MK,IW2),GRID(MK,IW),CORD(MK,IW)
997 FORMAT* 9 GRID5 *515,E12.5*5F20.5)
NCOL(IW)=NCOL(IW)+1

100 CONTINUE
DO 240 IW=1,2
NCOL1=NCOL(IW)
IW2=INVER(IW)
NCOL2=NCOL(IW2)
NCOL1=AMAX(0,NCOL1,NCOL2)
DO 210 I=1,NCOL1
GNUM=GRID(I,IW)
IF(GNUM.EQ.-9999;++DR,GNUM,EQ.77777.777)GO TO 210

DO 200 I=1,NCOL1
IF(ABS(GNUM-GRID(I,IW))GE.5)GO TO 200
IF(I1.EQ.1)GO TO 200
GRID(I1,IW)=77777.777

200 CONTINUE

210 CONTINUE
DO 230 I=1,NCOL1
GNUM=GRID(I,IW)
DO 220 I=1,NCOL1
IF(IPEG.EQ.53)PRINT 998,IW,I,II,NCOL1,
++GNUM,GRID(I,IW),GRID(I1,IW)
998 FORMAT* 9 GRID5 *515,E12.5*5F20.5)

IF(GRID(I1,IW)-GNUM).LT.0. AND.IW.EQ.1)GNUM=GRID(I1,IW)
IF(II.GNUM,GT.0. AND.IW.EQ.2)GNUM=GRID(I1,IW)
IF(ABS(GNUM-GRID(I1,IW))LE.5)GO TO 210

215 CONTINUE
COL1=GRID(I1,IW)
GRID(I1,IW)=GNUM
GRID(I1,IW)=COL1

220 CONTINUE

230 CONTINUE

240 CONTINUE
DO 250 IW=1,2
NCOL1=NCOL(IW)
IW2=INVER(IW)
NCOL2=NCOL(IW2)
NCOL1=AMAX(0,NCOL1,NCOL2)
DO 250 I=1,NCOL1
GRID1(IW,IW)=GRID(IW,IW)

250 CONTINUE
CALL ZERO(IGRID1,148)
CALL ZERO(GRID,288)
DO 249 IW=1,2
MAX=MB1(IW)
IFRIA=IFRIA(IW)
IW2=INVER(IW)
SUBROUTINE GRIDS/3 74/74 OPT=1

NCOL1=NCOL(IW2)
DO 249 IFR1=1
KA=COL1(IF+IW)
IF(IF.LT.IFR1)KB=COL1(IF+1+IW)-1
IF(IF.EQ.IFR1)KB=MAX
DO 246 KK=KA+KB
KK1=KK+KB
IF(IW.EQ.1)KK1=MK
LVM=(KK1-1)*LVM+INOS(3)+1
MK2=MKC(KK1,IW)
IF(IW.EQ.0.2)MK1=MKC(KK1,IW)
IF(CYL1<IF(LVM+IW.LE.1.E-10))GO TO 246
GRID1(IF+IW2)=CORD(MK1,IW2)
GO TO 247

246 CONTINUE
247 CONTINUE
248 CONTINUE
DO 252 IW=1,2
MAX=MB1(IW)
IFR1=IFR2(IW)
IW=INVER(IW)
NCOL1=NCOL(IW2)
NCOL2=NCOL(IW)
NCOL2=AMAX0(NCOL1,NCOL2)
DO 252 IF=1,IFR1
KA=COL1(IF+IW)
IF(IF.LT.IFR1)KB=COL1(IF+1+IW)-1
IF(IF.EQ.IFR1)KB=MAX
DO 252 MK=KA+KB
MK2=MKC(KK1,IW)
MK1=MKC(KK1,IW)
MK2=MKC(KK1,IW)
MN=MK
IF(IW.EQ.2)MN=MK
DO 254 I=1,NCOL1
IF(ISGRD1(I+IW2)=EQ.77777.777.AND.I.EQ.NCOL1)GO TO 251
IF(ISGRD1(I+IW2)=EQ.77777.777)GO TO 254
DO 248 TI=1,IFR1
IF(IBUS.EQ.53)
WRITE(*,'**(\"IGRIDS=1\"+\"816\"+\"3F10.5\")')
*IW1+I1+I3+MK+MN+GRID1(MK),GRID1(MK),GRID1(I1,IW2),
*GRID1(I1,IW2)+CORD(MK,IW2)
I3=IFRI-1+1
IF(AARS1(ISGRD1(I1,IW2))..GT..116)GO TO 246
IF(AARS1(ISGRD1(I1+IW2)-CORD(MK2,IW2))..GT..1)GO TO 254
IF(IW.EQ.1)IGRID1(MK)=1100+IGRID1(MK)
IF(IW.EQ.2)IGRID1(MK)=IGRID1(MK)+100+13
GO TO 251

248 CONTINUE
254 CONTINUE
IF(IW.EQ.1)IGRID1(MK)=IGRID1(MK)
IF(IW.EQ.2)IGRID1(MK)=IGRID1(MK)+100+IGRID1(MK)

251 CONTINUE
IF(IBUS.EQ.53)
1PRINT 996+IBUS +IW+NCOL1+MK+MN,
1GRID1(MK),GRID1(MN)

996 FORMA**.1*IGRIDS=220*.718
IF(TIUG6.0.53)
WRITE(*,6)**IGRID05-2**NCO1, MK, MK
*GRID1(MK), GRID2(MK)
252 CONTINUE
IF(TIUG6.0.53)GO TO 270
DO 260 IV = 1, 2
NCO1 = NCO1(IW)
IW = IW + 1
IF2 = IW + 1
NCO2 = NCO2(IW2)
NCO1 = MAXD(NCO1, NCO1, NCO2)
DO 260 IN = 1, NCO1
PRINT 999, IW, IN, GRID1(IN, IW)
999 FORMAT(* GRIDS-1*215, F14.3)
260 CONTINUE
270 CONTINUE
RETURN
END

LE MAP--(LOC4)
-ADDRESS--BLOCK-----PROPERTIES-------TYPE-------SIZE--NAME---ADDRESS---BLOCK---

| 3650B /PLOT1/ | REAL 1960 IFLA1 11330 |
| 642B // | REAL 20 IFLH 118 // |
| 0B /PLOT1/ | REAL 1960 IFR1 11160 |
| 270100 /PLOT1/ | REAL 1984 IFR2 68 /BMFL4/ |
| 23678 /BMFL4/ | REAL 90 IFR3 11218 |
| 4238 // | REAL 2 IGRID1 42040 /BMFL4/ |
| 11508 | REAL IFLN 1140 // |
| 440B /BMFL4/ | INTEGER 280 IN 11518 |
| 108 /BMFL4/ | REAL 280 INWAY 425100 /PLOT1/ |
| 231108 /PLOT1/ | REAL 1984 I09 4228 // |
| 327108 /PLOT1/ | REAL 1984 IPL0T 34140 /BMFL4/ |
| 366108 /PLOT1/ | REAL 1984 IPRINT 1138 // |
| 525718 /PLOT1/ | REAL 2 IRUN 1218 // |
| 526000 /PLOT1/ | REAL 2 I3 11220 |
| 558 // | REAL 3 ISOLV 30 // |
| 464128 /PLOT1/ | REAL 1984 IT1 11348 |
| 1158 // | REAL ITC1 4758 // |
| 1168 // | REAL ITEST 1208 // |
| 11448 | REAL IW 11140 |
| 4420B /BMFL4/ | REAL 280 IV2 11178 |
| 13058 /BMFL4/ | REAL 280 I1 11458 |
| 11438 | INTEGER I3 11638 |
| 172498 /PLOT1/ | INTEGER 1960 JBR 526008 /PLOT1/ |
| 25218 /BMFL4/ | INTEGER JET 7248 // |
| 10799 /BMFL4/ | INTEGER 140 JPL 42930 /BMFL4/ |
| 1038 // | INTEGER 6 JSTOP 1120 // |
| 758 // | INTEGER 6 K 11270 |
| 748 // | INTEGER KA 11249 |
| 68 // | INTEGER 6 KAK 1248 // |
| 66B // | INTEGER 6 KAI 11350 |
| 1152B | INTEGER KB 11258 |
| 1384B /BMFL4/ | INTEGER KB1 11328 |
| 120 // | INTEGER KET 7358 // |
| 27408 /BMFL4/ | INTEGER 140 KFLA6 11268 |
| 25328 /BMFL4/ | INTEGER 140 KK 11538 |
ENCODE(7.81,1WD1)ICOLF+ISIGN
ICHRI=2
GO TO 5
20 CONTINUE
ENCODE(7.82,1WD1)ICOLF+ISIGN
ICHRI=3
GO TO 50
30 CONTINUE
ENCODE(4.83,1WD1)ICOLF+ISIGN
ICHRI=4
GO TO 50
40 CONTINUE
ENCODE(4.84,1WD1)ICOLF+ICOL
ICHRI=5
50 CONTINUE
IF (ICHRI.EQ.0) ICMR2=1
IF (ICMR2.EQ.1) GO TO 60
ENCODE(3.82,1WD2)ICOL,ICLAP
ICMR2=3
GO TO 100
60 CONTINUE
ENCODE(2.81,1WD2)ICOL,ICLAP
ICMR2=2
GO TO 100
70 CONTINUE
IF (ICHRI.EQ.0) GO TO 150
IF (ICMR2.EQ.1) GO TO 80
ENCODE(3.82,1WD2)ICOL,ICLAP
ICMR2=3
GO TO 120
80 CONTINUE
ENCODE(2.81,1WD2)ICOL,ICLAP
ICMR2=2
100 CONTINUE
IF (ICOLD.EQ.0) GO TO 150
GO TO (110,120,110,120,110,120,110,120,110,120,120,120,120,120)*120,150,140,ICOLD
CALL GOTOER
110 CONTINUE
ICMR3=4
ENCODE(4.85,1WD3)IDI (ICOLD)
GO TO 150
120 CONTINUE
ICMR3=5
ENCODE(3.86,1WD3)IDI (ICOLD)
GO TO 150
130 CONTINUE
ICMR3=5
ENCODE(5.87,1WD3)IDI (ICOLD)
GO TO 150
140 ICOL1=ICOLI+1
ICOLD=9
ICOLII=ICOLI/12
ICOLI=MOD (ICOLI,12)
ICOLF=ICOLF+ICOLIII
GO TO 50
150 CONTINUE
ICHR1-ICHR2-ICHR3
GO TO 190

190 CONTINUE

WRITE(*,*) 190 CONTINUE

WRITE(*,*) 190 CONTINUE

*PRINT 998, ICHR1, ICHR2, ICHR3, DIMEN, IWDF, IWD2, IWD3
998 FORMAT(*, DIM5*, 9I4, 9E2.3, 9X, A01)

IF (LEVEL.EQ.0) ANGL=0
IF (LEVEL.EQ.1) ANGL=0.5
IF (LEVEL.EQ.2) ANGL=0.8
GO TO 194

194 CONTINUE

CIN=C1X-.03+ICHR*HGT
C2Z=C12+.847
ANGL=0.
GO TO 194

193 C2Z=C12-ICHR*HGT*.5
194 CONTINUE

CIN, SYMBOL(C1X, C2Z, HGT, IWDF, ANGL, ICHR1)
IF (ICHR1.EQ.0) GO TO 600
IF (LEVEL.EQ.0) GO TO 195
C1X=C1X+ICHR1*HGT
GO TO 196

195 C2Z=C12+ICHR1*HGT
196 CONTINUE
CALL SYMOL(C1X,C2Z,HGT,IWD2,ANGLE,ICH2)
IF(ICH2.EQ.(1+ICH3).OR.ICH2.EQ.(ICH3+ICH2)) GO TO 197
IF(ILEV.EQ.0) GO TO 198
C1X=C1X+ICH2*HGT
GO TO 198
197 C2Z=C2Z+ICH2*HGT
198 CONTINUE
CALL SYMOL(C1X,C2Z,HGT,IWD3,ANGLE,ICH3)
GO TO 630
200 CONTINUE
IF(ILEV.EQ.0) GO TO 203
C1X=C1X+.093-ICH3*HGT
C2Z=C2Z+.047
ANGL=0.
GO TO 204
203 C2Z=C2Z-ICH3*HGT*.5
204 CONTINUE
CALL NUMBER(C1X,C2Z,HGT,FLOAT(IWD1),ANGLE=1)
GO TO 600
210 CONTINUE
C2=C2+.047
FACTOR=.1.
ELEN=211/SCALE
HGT=HGT*FACTOR
WORD=ICH3*HGT
IF(ELEN.GT.WORD.AND.ICS.EQ.4) GO TO 214
IF(ELEN.GT.WORD) GO TO 212
FACTOR=.75
HGT=FACTOR*HGT
WORD=ICH3*HGT
IF(ELEN.GT.WORD.AND.ICS.EQ.4) GO TO 214
IF(ELEN.GT.WORD) GO TO 212
IF(IWAR.EQ.1) GO TO 211
C2=C2+.125
IWAR=1
GO TO 215
211 CONTINUE
C2=C2+.094
212 CONTINUE
IF(ICS.EQ.4) FAC1=.75
IF(ICS.EQ.5) PRINT 997, ICHR, ICHR3, DIMEN, HGT
997 FORMAT(1X,'DIM5=1+215+2F20.5')
C1=C1+DIMEN*.5*UZ11/SCALE-ICH3*.5*HGT+FAC1
IF(IWAR.EQ.2) GO TO 220
CALL SYMOL(C1,C2,HGT,IWD1,ANGLE,ICH1)
IF(ICHR.EQ.ICH1+ICH2) GO TO 600
C1=C1+ICH1*HGT
CALL SYMOL(C1,C2,HGT,IWD2,ANGLE,ICH2)
IF(ICHR.EQ.ICH1+ICH2) GO TO 600
C1=C1+ICH2*HGT
CALL SYMOL(C1,C2,HGT,IWD3,ANGLE,ICH3)
GO TO 600
SUBROUTINE IFIM5

C CONTINUE

WRITE(*,'(*,I8*1,IS*15..F15.5*)')

C *IND,ICHR,C1,C2,HGT,ANGLE
CALL NUMBER(C1,C2,HGT,FLOAT(IWD1),ANGLE,-1)
GO TO 600

250 CONTINUE
C1=C17=0.47
FACTOR=.75

ELEN=DIMEN*UZ11/SCALE
HGT=HGT*FACTOR
WORD=ICHR*HGT
IF(ELEN,GT,WORD.AND.IS,LT,EQ,4.)60 TO 254
IF(ELEN,GT,WORD)60 TO 252
FACTOR=.75

HGT=FACTOR*HGT,
WORD=ICHR*HGT
IF(ELEN,GT,WORD.AND.IS,LT,EQ,100)60 TO 254
IF(ELEN,GT,WORD)60 TO 252
IF(IWOR.EQ,11)60 TO 251
C2=C2+.125

IWOR=1
GO TO 255

251 CONTINUE
C2=C2-.394

252 CONTINUE
IF(K,LT,10)C1=C17=.422

254 CONTINUE
IWOR=0

255 CONTINUE
FAC1=.1
IF(ICHR3,LT,0)FAC1=.75
C2=C1*DIMEN*.5*UZ11/SCALE-ICHR*.5*HGT*FAC1

IF(KUR,LT,0)2160 TO 280
CALL SYMBOL(C1,C2,HGT,IWD1,ANGLE,ICHR1)
IF(ICHR,LT,ICHR1)60 TO 300
C2=C2+ICHR1*HGT
CALL SYMBOL(C1,C2,HGT,IWD2,ANGLE,ICHR2)
IF(ICHR,LT,ICHR1+ICHR2)60 TO 300
C2=C2+ICHR2*HGT
CALL SYMBOL(C1,C2,HGT,IWD3,ANGLE,ICHR3)
GO TO 390

280 CONTINUE
CALL NUMBER(C1,C2,HGT,FLOAT(IWD1),ANGLE,-1)
300 CONTINUE

600 CONTINUE
RETURN
END

IF MAP=(LOGA)

-ADDRESS----BLOCK------PROPERTIES--------TYPE---------SIZE----NAME-----ADDRESS--BLOCK----
FUNCTION FXRM5(L,J,K).
INTEGER XBM

COMMON/BMFL2/SPAN(30,20),XBM(30,20,4),LBM(30,20,4),JMX(30)

C
C THIS FUNCTION COMPUTES THE DISTANCE BETWEEN
C SECONDARY BEAMS
C K IS THE NO. OF SECONDARY BEAMS PER MAIN BEAMS
C MAX. 8
C
IF(K.GT.4)GO TO 2
FXRM5=MOD(XBM(L,J+K)*100000)
FXRM5=FXRM5*.1
RETURN
2 CONTINUE
FXBML=FXRM(L,J,K-4)/100000
FXRM5=FXRMS*.1
RETURN
END

LE MAP--(LO=A)
-ADDRESS--BLOCK------PROPERTIES-------TYPE----------SIZE

NONE
2 DUMMY-ARG
12430B /BMFL2/
3 DUMMY-ARG

-NAME-----ADDRESS---BLOCK-----

REAL
 integer
  L  1 DUMMY-AP
 LBM 5670B /BMFL2/
 SPAN 0R /BMFL2/
 XBM 1130P /BMFL2/

USES--(LO=A)
-----TYPE-----------ARGS--------CLASS---------

 GENERIC  2 INTRINSIC

ENT LABELS--(LO=A)
ADDRESS-----PROPERTIES----DEF

22B 14

POINTS--(LO=A)
-ADDRESS--ARGS------

4B 3

TICS--

M-UNIT LENGTH 360 = 30
ELLED COMMON LENGTH 12466B = 5430
RAGE USED 2000000 = 65536
E TIME 0.148 SECONDS
FUNCTION JLBM5
COMMON /BMWFL2/ SPAN(30*20), XBM(30*20*4), LBM(30*20*4), JMX(30)

C
C THIS FUNCTION LOOKS FOR THE MARK OF THE SECONDARY BEAM
C
IF (K .GT. 4) GO TO 2
JLBM5 = J0 + (LBM(L+J*K) * 10000)
RETURN
2 CONTINUE
JLBM5 = LCM(L+J*K-4) * 10000
RETURN
END

LE MAP -- (LO=A)
ADDRESS -- BLOCK ---- PROPERTIES ------ TYPE ------ SIZE
-------- NAME ------- ADDRESS ------- BLOCK ------
2 DUMMY-ARG
NONE INTEGER L 1 DUMMY-ARG
12430B /BMWFL2/ INTEGER LBM 5670B /BMWFL2/
3 DUMMY-ARG
INTEGER SPAN 0B /BMWFL2/
INTEGER XBM 1130B /BMWFL2/

UDES -- (LO=A)
----- TYPE -------- ARGS ------ CLASS ------

GENERIC 2 INTRINSIC

ENT LABELS -- (LO=A)
ADDRESS ---- PROPERTIES ---- DEF

218

POINTS -- (LO=A)
ADDRESS -- ARGS ---

4B 3

TICS--

M-UNIT LENGTH 33P = 27
ELLED COMMON LENGTH 12466B = 5438
RAGE USED 280000B = 65536
E TIME 0.132 SECONDS
SUBROUTINE LABEL5(IW,JW,C1,C2,KP,LVMA,IW1,LN2,SCALE)

COMMON/PILOT,STIF(800,2),MEN(800,2),MKB(800,2),KST(900,2)
1
I800,2)CSTIF(992,2)CLE1(992,2)CX(992,2)CZ(992,2)
2
IWAVY(2)PHI(992,2)DV(992,2)TRANS(900,2)C11(2)C22(2)
3
J800,2)

COMMON/ALFA,SINAS,COSA,ID3

COMMON/THETF,DOM(180)

DIMENSION I800,2501,X81(250),Y81(250),MRB(250),PWSH(21)

1
BIMSH(10,20),P82(250),RIM(240)

EQUIVALENCE (DUM(1),IF81(1)),(DUM(251),MRB(1)),(DUM(501),YP1(1))

EQUIVALENCE (DUM(751),MKB(1))

EQUIVALENCE (DUM(1001),BWSH(1)),(DUM(1022),RIMSH(1))

EQUIVALENCE (DUM(1622),Q82(1)),(DUM(172),THUM)

EQUIVALENCE (DUM(211),RIM(1))

COMMON/F8A,FLA,STPX1,STPX,TSTY1,STTP,MB(12),IFR(2),C0RD(140,2)
1
COL,ICFL(140),IFBM,GRD1(140,2),NCOL(2),MKC(140)

C8E1,992,I8U6,STZ1,STZ2,IFLAG(140),IFLA6(140)

5
KKPLT(140),SIDE(18),MARG,PLT(140,15),JPL,GRD1(140)

6
GRD1(140,2),DUMMY(240)

REAL MARK(I)

INTEGER COL(140,2)

DIMENSION LETER(26)

DATA LETER/1HA,1NB,1HC,1HD,1HE,1HF,1HG,1HM,1HI,1HJ,1HK,
1HL,1HM,1HN,1HO,1HP,1HG,1HR,1HS,1HT,1HU,1HV,1HY,1HZ/

C

COMPUTE HEIGHT AND CLEARANCE FOR MARKS AND TITLES

C

IM1 BM MARKS (ELEV.)
C

2 COL MARKS (ELEV.)
C

3 COL MARKS (PLAN)
C

4 BM MARKS (PLAN)

5 SEC. BM MARKS (PLAN)
C

6 PLAN VIEW TITLES

C

FACTOR=.75

HGT=.125*FACTOR

HGT2=HGT/FACTOR

COSA1=COSA

SINA1=SINA

HGT1=HGT

IFIF.EQ.5)HGT1=.75*HGT

IFIF.EQ.5)SINA1=0.

IFIF.EQ.5)COSA1=1.

IFIF.EQ.4.AND.IF8M.EQ.2)HGT1=.75*HGT1

IFIF.EQ.5.AND.IF8M.EQ.2)HGT1=.75*HGT1

C

WRITE BEAM MARKS

C

IF(IFM.EQ.2)GO TO 150

IF(IFM.EQ.3)GO TO 450

IF(IFM.EQ.6)GO TO 470

IF(IFM(140,JPL).EQ.0)GO TO 110

IF(IFM.EQ.1)MEN=MKB(LN2,TW),10000

IF(IFM.EQ.2)MEN=MKB(TW),10000

111=MEN/100

REN=3.*HGT1

C

150

C

450

C

470

C

110

C

10000

C
IF(L11. GT. 9) BLEN=4.*HGT1
C3=C1-1.5*HGT1
C2B=C2-5.*HGT1/3.
IF(IN. EQ.1) GO TO 60
IF(IN. EQ.2) GO TO 20
C2B=C2-DELEN*5.*SINA1=(. C97+HGT1)*COSA1
C3=C1+(. 047+HGT1)*SINA1=DELEN*.5*COSA1
GO TO 2C

20 CONTINUE
C2B=C2+(. 047+HGT1)*SINA1=DELEN*.5*COSA1
C3=C1+(. 047+HGT1)*COSA1=DELEN*.5*SINA1

60 CONTINUE
ALFA=0.
ALFA=ASIN(SINA1)*57.29570
ANGLE=0.
IF(IN. EQ.5) ITEMP=LN2
IF(IN. EQ.1) GO TO 90
ANGLE=ALFA
IF(IN. EQ.2) ANGLE=90+ALFA

90 CONTINUE
IF(IVUS. EQ. 5)
1PRINT 999, IM, IV, ITEMP, IFBM, C3, C2B, ALFA, ANGLE, STM, HGT
999 FORMAT(* ** DEBUG BEAM MARKS ***, I6, 2F6.2, 4F8.4)
IF(IPLLOT. EQ.0) GO TO 110
IF(IN. EQ. 4) AND IFBM=NE 3) GO TO 100
IF(IN. EQ. 5) AND IFBM=NE 3) GO TO 100
IF(IN. EQ.1) GO TO 100
C3=C1-1.5*HGT1
C2B=C2-.047-.5*HGT1
IF(IN. EQ.1) GO TO 95
C3=C1+.047+.5*HGT1
C2B=C2-.5*HGT1

95 CONTINUE
CALL SYMBOL(C3, C2B, HGT1, ****, ANGLE, 1)
GO TO 110
100 CALL NUMBER(C3, C2B, HGT1, FLOAT(ITEM), ANGLE, -1)
110 CONTINUE
IF(IVUS. EQ. 53) PRINT 998, LN2, 1W, C3, C2B.
998 FORMAT(* SECMARK*2I6, 2F8.2)
IF(IN. EQ. 4) GO TO 460
RETURN

C COLUMN MARKS **************

150 CONTINUE
DO 200 I=KA+1B
LN2=(I-1)+LVM+1
C1X=CX(LN2+IV)/SCALE+HGT
C2Z=2Z(LN2+IV)/SCALE+HGT
IF(IPLLOT. EQ.0) GO TO 200
CALL SYMBOL(C1X, C1Z, HGT, C=8, 11)
C1X=C1X+HGT
ITEM=MKC(I+IV)
CALL NUMBER(C1X, C1Z, HGT, FLOAT(ITEM)+8, -1)

200 CONTINUE

C LEVEL NUMBERS **************
SUBROUTINE LAB5/3

LEVEL=LV

IF(PUG.EQ.53)
1PRINT 990

990 FORMAT(* ** DEBUG LAB ** FIRST ST LAST ST (LVXM,LEV,LV COOR*)
DO 220 LL=KA,KR
   KK=LL
   210 LVXM=(KK-1)*LV+LEVEL
   IF(CSTF(LVX,IVX)*LE.1.E-20)GO TO 215
   GO TO 230
   215 CONTINUE
   IF(KK.EQ.KR)LEVEL=LEVEL+1
   IF(KK.EQ.KA)GO TO 220
   KK=KK+1
   GO TO 210
   220 CONTINUE
   230 CONTINUE
   DO 350 I=2,LEVEL
   IF(PUG.EQ.53)
   1PRINT 991,KA,KR,LM4,1,CLFV(I)

991 FORMAT(* ** DEBUG LAB ** A18,FL*2)
   CIX=0.
   C1Z=CLFV(I)/SCALE
   ELEN6=HGT
   IF(I.GT.9)ELEN6=2.*HGT
   C1=CIX/SCALE-.125
   C2=C1Z+.0938
   C3=C1-1.265
   IF(1.PLOT(12,JPL) .EQ. 0)GO TO 350
   CALL PLOT(C1,C1Z+3)
   CALL PLOT(C3,C1Z+29)
   NTEM=I-1
   C4=C1-.5625-ELEN6*5
   CALL NUMBER(C4,C2,HGT,FLOAT(NTEM),0..-1)
   350 CONTINUE
   C1=C4-.2*.HGT
   C2=C1+.75*.HGT
   IF(1.PLOT(12,JPL) .EQ. 0)GO TO 360
   CALL SYMBOL(C1,C2,HGT,LEVELS*60.,8)
   360 CONTINUE
   LN1=(KA-1)*LV+1
   LN3=(KR-1)*LV+1
   C1=CX(LN1+1)/SCALE
   C2=CX(LN3+1)/SCALE
   C3=STZ2/SCLAE-.18*HGT2
   C44=C1+C2+.5-16*HGT2
   C5=1.*C1+C2+.5
   C5=1.*C3-.2*.HGT2
   C=NEUPEN(14N)
   CALL SYMBOL(C4,C3,1.6*HGT2,*FRAMING ELEVATION **9,18)
C44=C4+.2*8*.HGT2
K=0
IVZ=INVER(IW)
NCL1=NCL1(IW2)
MK=MARKA(IW)
IG3=IGRID1(MK)*100
IF(IVZ .EQ. 2)IG2=IGRID1(MK)+180
}
DO 370 II=1,NCOL1
  IF(IIPUR.EQ.53) PRINT 993,K;1W2;II;NCOL1,
  *CORD(MK;IW2);GRID1(II;TW2)
  593 FORMAT(* LAR5=TI*14;2F20=5)
  IF(GRID1(II;IW2);EQ.;77777;777760;GO TO 370
  K=1
  IF(APS(CORD(MK;IW2);GRID(II;IW2);LY*1)GO TO 390
  CONTINUE
  380 CONTINUE
  K=16*R
  IF(IIPUE.1F.15) PRINT 992;NCOL1;II;K;IW
  NTERM=LETTER(K)
  992 FORMAT(* LAR5=TI*155)
  CALL SYMBOL(C44*C33;1.6*HGT2;*LINE *D*E)
  RAD=15
  C44=C44+15*R*HGT2
  C33=C33+8*HGT2
  CALL ARC(C44*C33;RAD;0*360.)
  IC=1
  IF(INW1Y(IW1);EQ.;1) GO TO 390
  C44=C44+5*HGT
  C33=C33+5*HGT
  CALL SYMBOL(C44*C33;HGT;NTERM;0*IC)
  CALL SYMBOL(CSX;CSY;HGT2;SCALF 1 = **R*10)
  CSX= CSX+10*HGT2
  NTERM= PLOT12(1;1)
  CALL NUMBER(CSX;CSY;HGT2;FLOAT(NTERM);0..-1)
  GO TO 410
  390 CONTINUE
  NTERM=K
  IC=1
  IF(NTERM.HT.9.AND.NTERM.LE.99) IC=2
  C44=C44+5*HGT*IC
  C33=C33+5*HGT
  CALL NUMBER(C44*C33;HGT;FLOAT(NTERM);0..-1)
  410 CONTINUE
  CALL NEWPEN(1)
  RETURN

450 CONTINUE

C PLAN VIEW COLUMN MARKS
C
C    IF (IFLAG1(LN2) .NE. 0) GO TO 460
C    IFLAG1(LN2) = 1
    C1X=C1
    C1Z=C2
    CALL NUMBER(C1X;C1Z;HGT;FLOAT(LN2);0..-1)

460 CONTINUE
RETURN

C PLAN VIEW TITLE ***************
C
470 CONTINUE
C    C2X=C1-17.8*HGT2
C    C2Y=C2-10.8*HGT2
C    CALL NEWPEN(11)
    CALL SYMBOL(C2X;C2Y);6*HGT2;#FRAMING PLAN LEVEL **0..28)
SUBROUTINE LABEL5/5 7/74 OPT=1  FTN 5.1*582  84

NTEM=LN2
CIX=C2X+32.*HGT2
CALL NUMBERIC1X,C2Y,1.6*HGT2,FLOAT(NTEM),0.,-1)  
CIX= C1-5.56*HGT2  
C2Y= C2Y-2.*HGT2  
NTEM= I.PLOT(20,J)  
CALL SYMBOL(C1X,C2Y,HGT2,*SCALE 1 = *,0.,10)  
CIX=C1X+10.*HGT2
CALL NUMBERIC1X,C2Y,HGT2,FLOAT(NTEM),0.,-1)  
CALL NEWPEN(1)  
RETURN
END

LF MAP--((LC=4))
ADDRESS--BLOCK------PROPERTIES-------TYPE--------SIZE-------NAME-----ADDRESS--BLOCK-----

| 13428 | 13439 | REAL | HGT2 | 1351F |
| 50508 | /BMFL4/ | REAL | I | 13458 |
| 17758 | /THEFL/ | REAL | IBUG | 1726B |
| 36508 | /PLOT1/ | REAL | IC | 1317F |
| 13578 | /FMFL4/ | REAL | ICFL | 1707B |
| 17508 | /THEFL/ | REAL | ID3 | 2B |
| 27010 | /PLOT1/ | REAL | IFBB | 1799F |
| 23678 | /BMFL4/ | REAL | IFLAG | 2748B |
| 4400 | /BMFL4/ | INTEGER | IFLAG1 | 1764B |
| 109 | /BMFL4/ | REAL | IFR2 | 2B |
| 18 | /ALFA/ | REAL | IGRID | 4B |
| 13328 | /PLOT1/ | REAL | IGRID1 | 1394B |
| 23110 | /PLOT1/ | REAL | IGR2 | 1372B |
| 13649 | /PLOT1/ | REAL | II | 1373B |
| 13658 | /PLOT1/ | REAL | IM | 1 |
| 397108 | /PLOT1/ | REAL | INUMX | 3517B |
| 366108 | /PLOT1/ | REAL | INWAY | 42510B |
| 3 | DUMMY-ARG | REAL | IPILOT | 3414B |
| 13778 | REAL | TW | 1 |
| 13508 | /PLOT1/ | REAL | J | 2 |
| 525768 | /PLOT1/ | REAL | JBR | 52602B |
| 4 | DUMMY-ARG | REAL | JPL | 4B |
| 13418 | | REAL | K | 1366B |
| 19488 | REAL | KA | 5 |
| 546488 | /PLOT1/ | REAL | KB | 6 |
| 13488 | | REAL | KK | 1353B |
| 13628 | | REAL | KPLT | 3154B |
| 13570 | | REAL | KST | 13370B |
| 13630 | REAL | LETTER | 1275B |
| 28 | /THEFL/ | REAL | LEVEL | 1351B |
| 50508 | /BMFL4/ | REAL | LI1 | 1336B |
| 464128 | /PLOT1/ | REAL | LL | 1352B |
| 13568 | | REAL | LN1 | 1369B |
| 13278 | | REAL | LN2 | 9 |
| 13658 | /BMFL4/ | REAL | LN3 | 1362B |
| 13388 | REAL | LVA | 7 |
| 13348 | REAL | LVN | 13548 |
SUBROUTINE LINES5(S,KA,KF,LYN,126,1W,EXD,SCALE)
COMMON PAT,MF,LEV,ISOLY,VFIN,LFTH,LYMAX,LHMAX,NL,IFLEV,IFRM
COMMON 22Z22,ZS,JDOC,DAT(3),IDOS(6),IDAT,ICRIST(6),ICR(6)
COMMON JSTOP,NSTOP,PRINT,SLINK,EC1,EC5,MAXFP,TEST,IRUN,LVU,LMU
COMMON MK,LIST(10),TOC,CODE(2),MN(20),UF(20)
COMMON ITC,ME(909)
COMMON USF(12),BOM(20),LON(20)
COMMON TAK(10),JFT(10),KET(20),MET(20)

INTEGER WAK(40,2),NHE(1,13)
COMMON/PLT1/RSTIF(980,2),RLN(980,2),MKB(980,2),KST(980,2)
COMMON/E0(980,2),STIF(992,2),CLE(1492,2),CX(992,2),CZ(992,2)
COMMON/INWAY(19992,2),DI(992,2),TRANS(90,2),C#(112,2),C2(26)
COMMON/J18(4,2)
COMMON/CEFL2/LX(140),LY(140),ST(140),YT(140),FCL(140,9)
COMMON/CEFL3/X(140),Y(140),VMAX(140),VMIN(140),XSC(140)
COMMON/FLP(140),STPX1,STPY1,MVX(1,2),CIP(140,2),C1(140,2),C2(140,2)
COMMON/INWER(140),INC1(140),INC2(140),NCCOL(2),MK(140,2)
COMMON/M(140),TRR(140),TSTZ2(140),IFLAG1(140),IFLAG2(140),IKX(140)

REAL X,REAL(140,2)

INTEGER COL1(140,2)
DIMENSION LETER(26)
DATA LETER`1HA,1HD,1HC,1HE,1HF,1HE,1HI,1HM,1HK`
*1LM,1M,1NH,1NO,1NP,1HQ,1HP,1H1,1HT,1MU,1HY,1M,1MY,1HY,1H7/
C
IFLCT=0
IFLC1=0
IWE=INVER(IW)
N#C=NCOL(I12)
IF(11E.5E260 TO 200)
C1=CEFL(140)+SCALE
C2=CEFL(140)+SCALE-.125
C21=.5
C24=0.625
C23=.9375
EXD1=EXD+.5625
MK=MK(C1A4K,IV)
IF(IW.7.0.EQ.1)IGR2=IGR1(MK)/100
IF(IW.7.0.EQ.2)IGR2=MOD(IGR1(MK),100)
CONTINUE
IF(IFLOT1.7.0.EQ.3)EXD1=EXD+.71875
IF(EXD1.LT.560 TO 30)
IF(EXD1.GT.5 AND EXD1.LE.7360 TO 30)
IFLOT1=2
GO TO 40
30 CONTINUE
IFLOT1=1
40 CONTINUE
IF(IFLOT1.7.0.EQ.3)60 TO 50
EN2
50 CONTINUE
DO 60 I=1,NK
CALL PLOT(C1+C2,3)
C2=C2+C21
CALL PLOT(C1+C2,2)
C2=C2+C24
CALL PLOT(C1+C2,3)
60 CONTINUE
C2=C2+C23
CALL PLOT(C1,C2+2)

C2=C2+C24

60 CONTINUE
IF(IFPLOT.EQ.2)60 TO 70
CALL PLOT(C1,C2+3)
C2=C2+EW1
CALL PLOT(C1,C2+2)

GO TO 60

70 CONTINUE
EXD=EXD1
IFPLOT1=3
K=8
K
GO TO 10

80 CONTINUE
C2=C2+.5*C21
RAD=25
CALL ARC(C1,C2,RAD,0..360.)
NCOL1=NCOL(IW)
NCOL1=AMAX1(NCOL1,NCOL2)
K=0
DO 90 I=1,NCOL1
IF(GRID1(I,IG).EQ.7777777777)60 TO 90
K=K+1
IF(IPUG.EQ.53)PRINT 998,I,IV,MK,K,NCOL1,
+GRID1(I,IG),GRID1(I,IV)

998 FORMAT(*,LINES=1*,G14.6,F10.0)
IF(ABS(CORD(MK,IV)-GRID1(I,IV)).LT.1)60 TO 92
90 CONTINUE
92 CONTINUE
IF(IPUG.EQ.53)PRINT 999,IV,I,1,K,NCOL1
IC=1

999 FORMAT(*,LINES=5*,515)
IF(IV.EQ.1)60 TO 94
K=IGR2
NTEM=K
IC=1
IF(NTEM. GT. 9. AND. NTEM.LE.99)IC=2

94 CONTINUE
HGT=.125
C1=C1-.5*HGT+IC
C2=C2-.5*HGT
IF(IV.EQ.2)60 TO 97
NTEM=LETER(K)
CALL SYMBOL(C1,C2,HGT,NTEM,0..IC)
GO TO 100

97 CONTINUE
CALL NUMBER(C1,C2,HGT,FLOAT(NTEM),0..1)
90 CONTINUE
GO TO 60

200 CONTINUE
C
C FLOOR PLAN GRID LINES
C
IW=INVER(IV)
IGRX=HA
IGRY=KB
MK=LVM
IF(IPUG E 9.5) WRITE (6,*) (*LINES-1)*,414,7F10.3*)
  *MK+1W,IGRY,GRID1(IGRX,1W),GRID1(IGRY,1W),
  *GRID1(IGRX,1W2),GRID1(IGRY,1W),EXD,FCL(MK,4),FCL(MK,5)
IF(IW.EQ.2) GO TO 210
IF(IW.EQ.2) GO TO 207
DIM1=Y(MK)+5/(12.*SCALE)
GO TO 207

207 CONTINUE
DIM1=1.5*FCL(MK,4)/(24.*SCALE)
208 CONTINUE
GRAY=GRAY1GRID1(IGRX,1W),CORD(MK,1W))/SCALE
EXD=EXD-GRAY*DIN1
C1=GRAY-.125*DIM1
C2=GRID1(IGRY,1W2)/SCALE
EXDI=AP*EXD+.5625
C21=*.5
C24=.0625
C23=.09375
GO TO 220

210 CONTINUE
IF(IAT.EQ.2) GO TO 215
DIM1=Y(MK)+5/(12.*SCALE)
IF(KAS(MK),EQ.2) DIM1=X(MK)+5/(12.*SCALE)
GO TO 217

215 CONTINUE
DIM1=1.5*FCL(MK,5)/(24.*SCALE)
217 CONTINUE
GRAY=AMAX1GRID1(IGRY,1W),CORD(MK,1W))/SCALE
EXD=EXD-GRAY*DIN1
C1=GRAY1GRID1(IGRY,1W2)/SCALE
C2=GRAY-.125*DIM1
EXDI=EXD-.5625
C21=*.5
C24=.0625
C23=.09375

220 CONTINUE
IF(IPUG.EQ.53) WRITE (6,*) (*LINES-2)*,514,6F15.3*)
  *MK+1W,IFPLOT1,IFPLOT*KN,EXD
  IF(IFPLOT1.EQ.3) EXDI=ARS(EXD)-.71875
  IF(EXDI.LE.5) GO TO 230
  IF(EXDI.GT.5) AND EXDI.LE.73) GO TO 230
  IFPLOT=2
  GO TO 240

230 CONTINUE
IFPLOT=1
240 CONTINUE
IF(IFPLOT1.EQ.3) GO TO 250

245 CONTINUE
IF(IW.EQ.1) GO TO 290
DO 260 T=1,KN
CALL PLOT(C1,C2,3)
C2=C2+C1
CALL PLOT(C1,C2,2)
C2=C2+C1
CALL PLOT(C1,C2,3)
C2=C2+C2
CALL PLOT(C1,C2,2)
C2=C2+C2
260 CONTINUE
IF(IFPLOT.EQ.2)G0 TO 270
265 CALL PLOT(C1,C2,3)
C2=C2+EXD1
CALL PLOT(C1,C2,2)
GO TO 240
270 CONTINUE
EXD=EXD1
IF(IFPLOT1.EQ.3)
KN=1
GO TO 220
280 CONTINUE
C2=C2+.5*C21
I2=I1X
I3=I2X
GO TO 330
290 CONTINUE
DO 300 I=1,KN
CALL PLOT(C1,C2,3)
C1=C1+C21
CALL PLOT(C1,C2,2)
C1=C1+C24
CALL PLOT(C1,C2,3)
C1=C1+C23
CALL PLOT(C1,C2,2)
C1=C1+C24
300 CONTINUE
IF(IFPLOT.EQ.2)G0 TO 310
305 CALL PLOT(C1,C2,3)
C1=C1-EXD1
CALL PLOT(C1,C2,2)
GO TO 320
310 CONTINUE
EXD=EXD1
IF(IFPLOT1.EQ.3)
KN=1
GO TO 220
320 CONTINUE
C1=C1+.5*C21
330 CONTINUE
RAD=.25
CALL ARC(C1,C2,RAD,0.360)
IFR2=GRID1(MK)/100
IF(IFW.EQ.2)GRID2=MOD(GRID1(MK),100)
IF(IFUB.EQ.53)
WRITE(6,105)WLINE5-1,418,IV,NK,IGR2,GRID1(MK)
105 FMT=125
IF(IFW.EQ.1)G0 TO 360
C1=C1-.5*MT
C2=C2-.5*MT
K=IGR2
NTEM=LETER(K)
IGC1
IF(IFUB.EQ.53)WRITE(6,105)WLINE5-5,414,1X,A2,3F15.5)
**MKIV, IGR2, IC, NTEM, C1, C2, HGT**

Call Symbol(C1, C2, HGT, NTEM, IC)

60 TO 600

360 CONTINUE

IC = 1

IF (NTEM.GT.9 AND .NOT. NTEM.LE.99) IC = C1 = C2 - 5 * HGT + IC

C2 = C2 - 5 * HGT

IF (IC.EQ.5) WRITE (6, (* LTNE5*5**5, 15, 3F15, 3))

**MKIV, IGR2, IC, NTEM, C1, C2, HGT**

CALL NUMBER(C1, C2, HGT, FLOAT(ITEM), 0.0, 1)

600 CONTINUE

RETURN

END

---

**LE MAP**

---

**ADDRESS** | **BLOCK** | **PROPERTIES** | **TYPE** | **SIZE** | **NAME** | **ADDRESS** | **BLOCK**
--- | --- | --- | --- | --- | --- | --- | ---
36500 | /PLOT1/ | REAL | 1960 | IDAT | 74B | //
642P | // | REAL | 20 | IDOC | 60B | //
00 | /PLOT1/ | REAL | 1960 | IDOS | 66B | //
270100 | /PLOT1/ | REAL | 1984 | *IFBM* | 1304B | /RMFL4/
2367B | /RMFL4/ | REAL | 90 | IFFRM | 128B | //
423B | // | REAL | 2 | IFLAG | 274B | /RMFL4/
4408 | /BMFL4/ | INTEGER | 280 | IFLAG1 | 252B | //
108 | /BMFL4/ | REAL | 280 | IFLF | 11B | //
231108 | /PLOT1/ | REAL | 1984 | IFLAG | 1112B | //
327108 | /PLOT1/ | REAL | 1984 | IPLOT1 | 1113B | //
366108 | /PLOT1/ | REAL | 1984 | IFR2 | 68B | /BMFL4/
1116B | // | REAL | IGRD | 442B | /RMFL4/ | //
525760 | /PLOT1/ | REAL | 2 | IGRID1 | 4204B | /BMFL4/
1117B | REAL | IGRX | 114B | //
1120B | REAL | IRGY | 114B | //
526800 | /PLOT1/ | REAL | 2 | IR2 | 1125B | //
1122B | // | REAL | ILINK | 114B | //
1124B | REAL | INWY | 425B | //
550 | // | REAL | 3 | I09 | 422B | //
1142B | // | REAL | IPRINT | 341B | /BMFL4/ | //
46442B | /PLOT1/ | REAL | 1984 | IRUN | 121B | //
1150 | // | REAL | IS | 1 | DUMMY-AR | //
1160 | // | REAL | ISDY | 3B | //
7 | DUMMY-ARG | REAL | ITIC | 475B | //
1123B | // | REAL | 1260 | ITEST | 120B | //
1060B | /CLFL2/ | REAL | 280 | IW | 6 | DUMMY-AR | //
1395B | /BMFL4/ | REAL |IW2 | 111B | //
1145B | REAL | IX | 114B | //
1144B | REAL | IX | 114B | //
1147B | REAL | I2 | 114B | //
1127B | // | INTEGER | I26 | 5 | DUMMY-AR | //
17240B | /PLOT1/ | INTEGER | 1960 | I2X | 115B | //
25218 | /BMFL4/ | INTEGER | I3 | 111B | //
1135B | INTEGER | JBR | 526B | //
1077B | /BMFL4/ | INTEGER | 140 | JET | 72B | //
750 | // | INTEGER | JPL | 42B | //
75B | // | INTEGER | 6 | JSTOP | 111B | //
SUBROUTINE LOGOS/1

COMMON M4, M7, M8, M14, IFOL3, IOSR, ORG, ORGY, SCALE
COMMON Z2Z2, ZTME, DAT(5), IDOC(6), IDOS(6), IDAT, ICRI(6), IERIC(6)
COMMON JSTOP, NSTOP, TPRINT, NLINK, EC1, EC5, MAXF, ITEST, IURM, LVU, LNU
COMMON KAK, LIST(10), IOD, CODE(3), MD(20), UF(20)
COMMON ITA1, NE2(40), SMD(20), LON(20)
COMMON TAK, (10), JT(10), KET(20), LET(20)

INTEGER KAK(40, 2), ZTME(11, 3)
DIMENSION ICC(140, 2), LOGO(5)
DATA LOGO, MAXPDO, MAXMD, MDFRAN, MDFSCALE, MDAT, XZERO, SIDE(IPS+1) = 0.03

REAL MARG(10)
INTEGER COL1(140, 2)

CALL PLOT(XLOGO, YZERO, 3)
CALL PLOT(XLOGO, YLOGO2, 2)
CALL PLOT(XLOGO, YLOGO3, 3)
CALL PLOT(XZERO, YLOGO3, 2)
CALL PLOT(XLOGO, YLOGO3)

CALL PLOT(XZERO, YLOGO2, 2)
CALL PLOT(XLOGO, YLOGO, 5, 3)
CALL PLOT(XZERO, YZERO, 5, 2)
CALL PLOT(XLOGO, YZERO, 1.7, 3)
CALL PLOT(XZERO, YZERO, 1.7, 3)
CALL PLOT(XLOGO, YZERO, 4, 3, 2)
CALL PLOT(XLOGO, YZERO, 6, 4, 3)
CALL PLOT(XZERO, YZERO, 6, 4, 3, 2)
CALL PLOT(XZERO, YZERO, 2.75, YZERO, 1.7, 3)
CALL PLOT(XZERO, YZERO, 2.75, YZERO, 2)
CALL PLOT(XZERO, YZERO, 1.7, 3)
CALL PLOT(XZERO - 2, YZERO, 0.25, 2)
CALL PLOT(XZERO - 1, YZERO, 0.17, 3)
CALL PLOT(XZERO - 1, YZERO, 0.2)
CALL PLOT(XZERO - 2.75, YZERO, 0.8, 3)
CALL PLOT(XZERO, YZERO, 0.82)
CALL PLOT(XZERO, YZERO - 2.75, YZERO - 0.1, 3)
CALL PLOT(XZERO, YZERO + 1.7, 3)
CALL PLOT(XZERO, YZERO + 0.25, 2)
CALL PLOT(XZERO - 1, YZERO + 1.7, 3)
CALL PLOT(XZERO - 1, YZERO + 0.2)
CALL PLOT(XZERO - 2.75, YZERO + 0.8, 3)
CALL PLOT(XZERO, YZERO + 0.82)
CALL PLOT(XZERO - 2.75, YZERO + 0.1, 3)
CALL PLOT(XZERO, YZERO + 1.1, 2)
CALL PLOT(XZERO - 2.75, YZERO + 1.4, 3)
CALL PLOT(XZERO, YZERO + 1.4, 2)
CALL NEWPEN(11)
IF(III.EQ.1) GO TO 80
IF(III.EQ.2) GO TO 90
MIV = .2
IC=10
XWR=YLLOGO+IC*.5*HGT+.5+.5
YWR=YZERO+.7315
CALL SYMBOL(XWR,YWR,HGT,*,CIVIL ENGINEERING DEPT.*,0,IC)
HGT=.15
IC=2
XWR=YLLOGO+.5*.5=IC*HGT*.5
YWR=YZERO+.955
CALL SYMBOL(XWR,YWR,HGT,*,ENGINEERING CORPORATION*,0,IC)
80 CONTINUE
HGT=.2
IC=4
XWR=YLLOGO+IC*.5*HGT+.5+.5
YWR=YZERO+.7315
CALL SYMBOL(XWR,YWR,HGT,*,PAINT*,0,IC)
HGT=.15
IC=23
XWR=YLLOGO+.5*.5=IC*HGT+.5
YWR=YZERO+.955
CALL SYMBOL(XWR,YWR,HGT,*,ENGINEERING CORPORATION*,0,IC)
90 CONTINUE
HGT=.15
XWR=YLLOGO+1.875
YWR=YLLOGO+.30
CALL SYMBOL(XWR,YWR,HGT,*,NOTES*,0,IC)
HGT=.0625
XWR=YLLOGO+HGT
YWR=YZERO+.375
CALL SYMBOL(XWR,YWR,HGT,*,PROJECT NO.*011)
XWR=YZERO-.25+HGT
CALL SYMBOL(XWR,YWR,HGT,*,DRAWING NO.*011)
XWR=ZERO-1.5+HGT
CALL SYMBOL(XWR,YWR,HGT,*,REVISION*0,9)
HGT=.25
XWR=YZERO-.25-.5+HGT
AINC=0
XWR=YZERO-.5+AISC-HGT*.5
FIELD=DATE()
IF(DAT(1),EQ.0) READ(FIELD,*,(A10,*)DAT(1)
DO 100 ILO=1,5
ILO=4
IF(ILO.EQ.3 OR ILO.EQ.4) ILO=5
NTEM=LOGO(ILO)
IF(ILO.EQ.5) XWR=YZERO-1.5+25+HGT
IF(ILO.EQ.5) YWR=YZERO-1.55-HGT*.5
CALL SYMBOL(XWR,YWR,HGT,NTEM,0,1,0)
IF(ILO.LE.3) GO TO 95
XWR=YZERO-1.5+HGT
CALL SYMBOL(XWR,YWR,HGT,*,DAT(1),8,10)
95 CONTINUE
AINC=.3
XWR=YWR+AINC
100 CONTINUE
NTEM=SCALE
IC=1
IF(NTM.GT.9 AND NTEM.LE.99) IC=2
IF(NTE,GT,90.)AND.,NTE,LT,.999)IC=3
HWR=2*EO,6+1*IC,5*HGT
YWR=YZERO,5+5*HGT*5
CALL SYMBOL(XWR,YWR,HGT,+AS NOTED,0,5,8)
HGT=,125
C
C NME(I,11) I=0,11 KEEPS JOB NO. AND IF USER PUTS AN SLASH
C FOLLOWED BY SS IF STEEL OR SC IF CONCRETE
C NME(I,13) I=0,11 KEEPS REVISION NO.
C
DO 200 I=1,3,2
ILRA=0
DO 120 I=9,11
IF(NME(I,1),EQ,6H)AND,(I,EQ,9)GO TO 140
IF(NME(I,3),EQ,6H)AND,(I,EQ,9)GO TO 260
IF(NME(I,11),EQ,6H)ILRA=I-1
IF(ILRA,NE,0)GO TO 130
120 CONTINUE
ILRA=-10
130 CONTINUE
IF(ILRA,NE,9)ILRA=1
IF(ILRA,EQ,10)ILRA=2
IF(ILRA,EQ,11)ILRA=3
DO 132 I6=1,4P
ICCI(I6)=10H
132 CONTINUE
NN1=1
NN2=6
ILRA=ILRA+P
DO 134 I11=9*ILRA2
DECODE(16,887*NME(I11,N)+ICCI(KK),KK)NN1,NN2
IF(I11,NE,ILRA2)GO TO 133
NN1=NN1+6
NN2=NN1+5
134 CONTINUE
133 CONTINUE
967 FORMAT(6A1)
IF(NME,EQ,3)GO TO 142
N3=ILRA+6
DO 135 I=1,N3
IF(ICCI(I),EQ,1H)GO TO 142
135 CONTINUE
142 CONTINUE
IF(NME,EQ,1)GO TO 146
C
C REVISION NO.
C
HGT=125
I2=I-1
I9=1
DO 143 I1=1,1P
IF(ICCI(I1),EQ,1H)GO TO 143
I3=I1
143 CONTINUE
I2=I3
XWR=YZERO,5-HGT*I2,5
YWR=YZERO,-1875
DO 144 I3=1,I2
NTM=ICC(I1)
CALL SYMBOL(XWR,YWR,HGT,TEM,0..1)
XWR=XWR+HGT
144 CONTINUE
GO TO 270
C
C JOB NO.
C
C 146 CONTINUE
IC=I-1
IO=I
HGT=1.E5/IC
IF(HGT.GT.12)HGT=12
XWR=XLOGO+1.75*IC-HGT*.5
YWR=YZERO+1.875
DO 150 I=1,IC
NTM=ICC(I)
CALL SYMBOL(XWR,YWR,HGT,TEM,0..1)
XWR=XWR+HGT
150 CONTINUE
160 CONTINUE
HGT=12
NWOR=3HSC=
IF(MAT,EQ.1)NWOR=HSC=
XWR=YZERO+1.875-HGT*1.5
YWR=YZERO+1.875
CALL SYMBOL(XWR,YWR,HGT,NWOR,0..3)
IC=1
IF(IFPLT.GT.9)IC=2
NTM=IFPL
CALL NUMBER(XWR,YWR,HGT,FLOAT(NTM),0..1)
200 CONTINUE
C
C CLIENT NAME NME(I+2)=1!*8
C IF SLASH IT IS ENTERED WRITE IN 2 LINES
C
HGT=2
DO 260 N2=1,3
IBLA=0
DO 255 I=1,N
IF(N2.EQ.3.AND.HGT1.EQ.1.5)HGT=1.5
IF(HGT1.LT.15.AND.N2.EQ.3)HGT=HGT1
IF(NME(I+1).EQ.6H .AND.I.EQ.1)60 TO 260
IF(NME(I+2).EQ.6H .AND.I.EQ.1)60 TO 260
IF(NME(I+N2).EQ.6H )IBLA=I-1
IF(IBLA.NE.0)60 TO 257
255 CONTINUE
IBLA=8
257 CONTINUE
DO 236 I6=1,18
ICC(I6)=10H
236 CONTINUE
NN1=1
NN2=6
DO 241 III=1,IBLA

DECODE(6*RR7*NME(311*N2))*(ICC1(KK)*KK=NN1, NN2)
IF(III*EQ.1.1L1)GO TO 239
NN1=NN1+6
NN2=NN1+5
241 CONTINUE
239 CONTINUE
W1=IPLA*6
IPLAI=0
DO 238 I=1,N1
IF(ICC1(I)*EQ.1H)GO TO 242
238 CONTINUE
DO 240 KK=1,N1
IF(ICC1(KK)*EQ.1H)GO TO 247
IC=KK
240 CONTINUE
IPLAI=1
GO TO 243
242 CONTINUE
IC=I-1
243 CONTINUE
IC1=N1
IF(ICC1.LE.0)IC1=IC
HGT=4./IC
IF(N2.EQ.3.AND.HGT.GT.HGT1)HGT=HGT1
IF(HGT.GT.2)HGT=2
K1=IC
IF(ICC1*EQ.N1)K1=IC-1
IF(N2.EQ.3)HGT1=HGT
XWR=XL060*4.5*5-K1*5*HGT
YWR1=5.65
IF(N2.EQ.1)YWR1=3.73
IF(N2.EQ.3)YWR1=2.75
IF(N2.EQ.3.AND.(IC-I-1).EQ.IC)YWR1=YWR1-0.15
IF(N2.EQ.1.AND.(I-1).EQ.IC)YWR1=YWR1-0.1
IF(N2.EQ.2.AND.(IC-I-1).EQ.IC)YWR1=5.43
YWR=YZERO+YWR1+HGT*5
DO 245 I=1,K1
NTEM=ICC1(I)
CALL SYMBOL(XWR,YWR,HGT,NTEM,0,1)
XWR=XWR+HGT
245 CONTINUE
IF(ICC1.EQ.N1)GO TO 250
IF(IPLAI.EQ.1)GO TO 250
K1=IC+2
K2=IC1
DO 247 KK=K1,K2
IF(ICC1(KK)*EQ.1H)GO TO 247
K3=KK
247 CONTINUE
K3=K2
YWR1=5.15
IF(N2.EQ.1)YWR1=3.53
IF(N2.EQ.3)YWR1=2.63
XWR=XL060*4.5*5-(K2-K1)*5*HGT
YWR=YZERO+YWR1+HGT*5
DO 248 I=K1,K2
NTEM=ICC1(I)
248 CONTINUE
CALL SYMBOL(YWR,YWR,HGT,KTEM,0..1)
  XWR=YWR+HGT
  248 CONTINUE
  250 CONTINUE
  250 CONTINUE
  MHT=15
  IC=18
  IF(IS,EQ,2)IC=E
  XWR=XFEC+4.0*E-IC*5+HHT
  YWR=YZERO+2.03
  NTEM=10+ELEVATIONS
  IF(IS,EQ,2)NTFM=50+LANS
  CALL SYMBOL(YWR,YWR,HGT,FREMING,0..0)
  IC=10
  IF(IS,EQ,2)IC=5
  XWR=XWR+P+HGT
  CALL SYMBOL(YWR,YWR,HGT,KTEM,0..IC)
  CALL NEWPN(1)
  RETURN
END

<table>
<thead>
<tr>
<th>LE MAP (LOC=A)</th>
<th>ADDRESS--BLOCK----PROPERTIES-----TYPE------SIZE</th>
<th>NAME</th>
<th>ADDRESS--BLOCK---</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>REAL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1778A</td>
<td>REAL</td>
<td>IFPL3</td>
<td>5 DUMMY-AR</td>
</tr>
<tr>
<td>642P</td>
<td>REAL</td>
<td>IFR2</td>
<td>6P</td>
</tr>
<tr>
<td>2367P /BMFL4/</td>
<td>REAL</td>
<td>IGRID</td>
<td>44286</td>
</tr>
<tr>
<td>425B</td>
<td>REAL</td>
<td>IGRID</td>
<td>42048</td>
</tr>
<tr>
<td>440A /BMFL4/</td>
<td>INTEGER</td>
<td>I11</td>
<td>2004B</td>
</tr>
<tr>
<td>10B /BMFL4/</td>
<td>REAL</td>
<td>ILINK</td>
<td>114B</td>
</tr>
<tr>
<td>55A</td>
<td>REAL</td>
<td>ILO</td>
<td>1771P</td>
</tr>
<tr>
<td>115B</td>
<td>REAL</td>
<td>ILOI</td>
<td>1772B</td>
</tr>
<tr>
<td>116B</td>
<td>REAL</td>
<td>IIM</td>
<td>1 DUMMY-AR</td>
</tr>
<tr>
<td>1666A</td>
<td>CHAP+10</td>
<td>ILO</td>
<td>422B</td>
</tr>
<tr>
<td>13058 /BMFLA/</td>
<td>REAL</td>
<td>IPLOT</td>
<td>3414P</td>
</tr>
<tr>
<td>1764B</td>
<td>REAL</td>
<td>IPRT</td>
<td>113R</td>
</tr>
<tr>
<td>2021A</td>
<td>REAL</td>
<td>IPS</td>
<td>6 DUMMY-AR</td>
</tr>
<tr>
<td>17778</td>
<td>INTEGER</td>
<td>I0</td>
<td>2012B</td>
</tr>
<tr>
<td>17760</td>
<td>INTEGER</td>
<td>IROUN</td>
<td>121B</td>
</tr>
<tr>
<td>2025B</td>
<td>INTEGER</td>
<td>IS</td>
<td>2 DUMMY-AR</td>
</tr>
<tr>
<td>2003B</td>
<td>INTEGER</td>
<td>ISOLV</td>
<td>3R</td>
</tr>
<tr>
<td>25218 /BMFL4/</td>
<td>INTEGER</td>
<td>ICT1</td>
<td>475B</td>
</tr>
<tr>
<td>17658</td>
<td>INTEGER</td>
<td>ICT</td>
<td>128B</td>
</tr>
<tr>
<td>16670</td>
<td>INTEGER</td>
<td>I4</td>
<td>6 DUMMY-AR</td>
</tr>
<tr>
<td>10700 /BMFL4/</td>
<td>INTEGER</td>
<td>I10</td>
<td>2013B</td>
</tr>
<tr>
<td>103B</td>
<td>INTEGER</td>
<td>I11</td>
<td>2011B</td>
</tr>
<tr>
<td>75B</td>
<td>INTEGER</td>
<td>I13</td>
<td>2014B</td>
</tr>
<tr>
<td>2026B</td>
<td>INTEGER</td>
<td>I6</td>
<td>2800B</td>
</tr>
<tr>
<td>74A</td>
<td>INTEGER</td>
<td>JET</td>
<td>724B</td>
</tr>
<tr>
<td>60B</td>
<td>INTEGER</td>
<td>JPL</td>
<td>4203R</td>
</tr>
<tr>
<td>66B</td>
<td>INTEGER</td>
<td>JSTOP</td>
<td>111B</td>
</tr>
<tr>
<td>13046 /BMFL4/</td>
<td>INTEGER</td>
<td>KAK</td>
<td>120B</td>
</tr>
<tr>
<td>12B</td>
<td>INTEGER</td>
<td>KET</td>
<td>736B</td>
</tr>
<tr>
<td>27408 /BMFL4/</td>
<td>INTEGER</td>
<td>KK</td>
<td>2800B</td>
</tr>
<tr>
<td>25248 /BMFL4/</td>
<td>INTEGER</td>
<td>KKPCT</td>
<td>3154B</td>
</tr>
<tr>
<td>11B</td>
<td>INTEGER</td>
<td>K1</td>
<td>2027B</td>
</tr>
</tbody>
</table>
SUPROUTINE POS1 /1

COMMON MAT, MEM, LLEV, ISOLV, LVFIN, LNFIN, LPMAX, LMANAX, NLIFLEV, IFRM
COMMON ZZ, ZMEM, DAT(3), TDC(6), IDOS(6), IDAT, IRC(4), ICRGE(6)
COMMON JSTOP, NSTOP, IPRINT, JLINF, EC, EC5, MAXFR, ITEST, IRUN, LVU, LAU
COMMON KAX, LIST(10), I109, CDDF(2), MN(20), UF(20)
COMMON ICTI, NE7(90)
COMMON JUSF(10), BDN(20), LCM(20)
COMMON TAKI(10), JET(10), KET(20), LET(20)
INTEGER KAK(90, 2), NME(11, 3)
COMMON/IPL(980, 2), BLEN(980, 2), MKR(980, 2), KST(980, 2),
1/BR(980, 2), CSTI(982, 2), CLNII(982, 2), CXX(952, 2), CZZ(952, 2),
2/INWAY(2), PHP(992, 2), DVI(992, 2), TRANS(90, 2), C11(2), C22(2),
3/BR(993, 2)
COMMON/FLA/STPX, STPY, STPY1, M3(2), IFA(2), CORD(14, 2), CCL
1/ICFL(140), IFRM, GRID1(140, 2), NCOL(2), MKC(140, 2)
4/CLEV(90), IPUG, STZ1, STZ2, IFLAG1(140), IFLAG(140), KPL(140)
5/SIDE(10), MARG, IPILOT(25, 15), JPL, IGRID(140), IGRID(140, 2)
REAL MARG(10)
INTEGER COL(140, 2)
COMMON/ALPHA/ALFA, COSA, ID3
COMMON/MFL2/SPAN(30, 20), XEM(30, 20, 4), LR(30, 20, 4), JX(30)
COMMON/BMFL1/WIDTH(30, 20), DEPTH(30, 20)
COMMON/MFL3/EYE(30, 20), ILTV(30)
COMMON/CLFL2/LX(140), JX(140), LY(140), JY(140), FCL(140, 9)
COMMON/CLFL3/YX(140), Y(140), Z(140), XMAX(140), YMAX(140), KSC(140)

1KOC(140)
IFMB = 0
IWM = INVER(IW)
FACTOR = .75
MGT = 12 * FACTOR
IRUG = LET(11)
CWEB = 1.5 * FCL(MK, 4) / (24 * SCALE)
IF(MAT * EQ. 1) CWEB = 1.5 * MW(K) / (24 * SCALE)
IF(MAT = EQ. 1 AND KSC(MK) EQ. 1) CWEB = 1.5 * MW(K) / (24 * SCALE)
IF(IBUG EQ. 53)  
1PRINT 990, JIW, MK, K, JXM, X1, C3, C5L, CA0, CIW, CW 
990 FORMAT(* TO POSTTI * 61, 6, 53)
CLEG = HGT
IF(MK, GT, 9) CLEG = 2 * HGT
IF(MK, GT, 99) CLEG = 3 * HGT
C5L = C3L + CLEN
IF(IW, EQ. 1) MARK = MOD(MK, LVNC(14)), 10000
IF(IW, EQ. 2) MARK = MKR(LVNC(14))/1000
L12 = MARK/100
L12 = MARK-L12/100
KASE = 1
BLEN = 3 * HGT
IF(MARK, GT, 920) BLEN = 4 * HGT
BLEN = CIW - BLEN + 5 - HGT/3.
L12 = LY(MK)
LY = JY(MK)
LY = IF(IW, EQ. 1) 60 TO 20
L12 = L12
JY = IS2
L12 = XN(MK)
IS2 = JX(MK)
IV3 = IV
20 CONTINUE

IF (L12 .EQ. 0 .AND. L2 .EQ. 0) GO TO 510
SPAN = SPAN(L12, IS2) / (12 * SCALE)
IF (W .EQ. 0) SPAN = SPAN(L1, J1) / (12 * SCALE)
IF (PLENG .EQ. 0) SPAN = LT .EQ. 75 * PLEN
IF (Z).EQ. 0) PLEN = 4
IF (FLANGE(L12, IS2) .EQ. 0) GO TO 50
IF (PLOT(14, J1) .EQ. 0) GO TO 50
IF (L12 .EQ. 0) GO TO 70
IF (SPAN(L12, IS2) .EQ. 0) GO TO 70
IF (WIDTH(L12, IS2) .EQ. 0) GO TO 70
IF (TRUE .EQ. 0) PRINT 992, CASE, L12, IS2, MARK, WIDTH(L12, IS2)

* EYELIY1, J1

IF (TRUE .EQ. 53) PRINT 992, CASE, MK, IV, MK, C4L, BLENG1, C3L, C4L, BLENG1
92 FORMAT (POSIT, 4I6, 5F8.3)
TFC = C3L, BLENG1, AND, ALFA, EQ, 0.1, GO TO 100
70 IF (J1 .EQ. 0) GO TO 500
IF (SPAN(L1, J1) .EQ. 0) GO TO 500
IF (EYELIY1, J1-1 .EQ. 0) GO TO 500
PLENY = 0.5HET
IF (Y1 .EQ. 0) BLENGY = 0.5HET
PLENG = (CORD(MK, IV3) - SPAN(L1, J1-1, 24) / SCALE + BLENGY + 5HET / 3.
IF (TRUE .EQ. 53) PRINT 992, CASE, MK, IV, MARK, C4L, BLENG1
IF (C4L .EQ. 0, PLEN5) GO TO 500
C TRY UPPER RIGHT CORNER
C
100 CONTINUE

MARK = J1 = 100, J1
C1L = C3L
C2L = 2 * CORD(MK, IV3) / SCALE - C4L = HET
IF (W .EQ. 0) C2L = 2 * CORD(MK, IV) / SCALE - C4L = HET
C3L = C2L = HET
IF (PLOT(18, J1) .EQ. 1) GO TO 140
KASE = 2
IF (J1 .EQ. IS2) GO TO 500
IF (SPAN(LY1, J1) .EQ. 0) GO TO 500
IF (EYELIY1, J1-1 .EQ. 0) GO TO 500
PLENY = 0.5HET
IF (Y1 .EQ. 0) BLENGY = 0.5HET
BLENGY = (CORD(MK, IV3) + SPAN(LY1, J1) / 24) / SCALE - 5HET - HET / 3.
IF (TRUE .EQ. 53) PRINT 992, CASE, MK, IV3, MARK, C5L, BLENGY
IF (C5L .EQ. 0, BLENGY) GO TO 500
C TRY LOWER LEFT CORNER
C
140 CONTINUE

KASE = 3
C1L = 2 * CORD(MK, IV) / SCALE - C3L - CLEN6 .75
C2L = C4L
IF (W .EQ. 0) GO TO 145
C1L = 2 * CORD(MK, IV) / SCALE - C3L - CLEN6 .75
145 IF (K .EQ. KA) GO TO 500
SPAN1=0.
IF(12.EQ.0.AND.IPL0T(18.J).EQ.0)GO TO 500
IF(L12.EQ.0)GO TO 150
SPAN1=SPAN(L12.IS2=1)
IF(SPAN1.EQ.0.AND.IPL0T(18.J).EQ.0)GO TO 500
150 CMW=(CORD(MK,IW)-SPAN1/24)*SCALF+.5*BLENG-HGT/3.
IF(IPL0T(18.J).EQ.0)GO TO 155
IF(CM14.EQ.C4W.AND.12L.EQ.BLEN6)GO TO 500
155 C2L=2.*CORD(MK,IW3)/SCALE-C4L-HGT
IF(IW.EQ.2)C2L=2.*CORD(MK,IW)/SCALE-C3L-HGT
IF(IPL0T(18.J).EQ.0)GO TO 510
C SPAN1=SPAN(L12.IS2)/(12.*SCALE)
C IF(SPAN1.LE.BLEN6)GO TO 160
C GO TO 500
C C REDUCE BEAM MARK SIZE TO 3/4 OF HGT
C C
160 CONTINUE
KASE=4
C4L=C4L+BLEN6
BLEN6=.75*BLEN6
BLEN61=C1V-BLEN6+.5-HGT/4.
IF(C4L.LT.BLEN61)GO TO 500
200 CONTINUE
C C WRITE FOOT NOTE
C C
IFRM=3
KASE=5
500 CONTINUE
IF(KASE.EQ.4)IFRM=2
IF(KASE.EQ.1)GO TO 510
C3L=C1L
C4L=C2L
510 CONTINUE
IF(I.executeQuery33(EQ.53)) PRINT 991,JMK,MKB,LVNC,IW,IFRM,KASE,C3L,C4L,C1V,C2V
PRINT 991,FORMAT* IQ POSIT5 *>519,5FP.3)
RETURN
END

LF MAP-----LG=A-----
ADDRESS--BLOCK-----PROPERTIES--------TYPE---------SIZE
--NAME--ADDRESS--BLOCK--

0B /ALFA/
36500 /PLOT1/
6430 REAL 1968 CLEV 2367/R /NMFL4/
6590 REAL 238 CODE 4238 //

1 6550 REAL COSA 18 /ALFA/
6440 REAL CSTIF 23110R /PLOT1/
6510 REAL CWED 634R
6420 REAL 20 C1 39710R /PLOT1/
0P /PLOT1/
6350 REAL 1960 C2 36610R /PLOT1/
27010B /PLOT1/
1804 C1L 6530
SUBROUTINE R3/1

C COPYRIGHT (C) 1983 AMECO COMPUTING SYSTEMS INC.

C INTEGER XR

C COMMON MATEM,LEV,SOLV,LVFIN,LNFIN,LVMAX,LMAX,NL,TL,FLEV,IFRR
C COMMON ZZZZZ,NME,DAT(3),IDOC(6),ICRIS(6),ICRIS(6),ICRIS(6)
C COMMON USTOP,USTOP,PRINT,NLNK,ECSC,MAXT,TEST,IRUN,LVLU,LWU
C COMMON IAK,LIST(10),IO9,COF(2),M0(20),UF(20),ITC1
C COMMON KE2,IPF,USF(19),EON(20),LON(20)
C COMMON TAKI(10)
C COMMON NME(11,3)
C COMMON IAK(190,2),NEQ

C COMMON /BMFL1/WIDTH(30,20)*DEPTH(30,20)
C COMMON /BMFL2/SPAN(30,20)*XN(30,20,4)*LRM(30,20,4)*JNM(30)
C COMMON /BMFL3/EYE(30,20)*ILM(30)
C COMMON /BMFL4/PIN(30,21,4)
C COMMON/MLN1/FCL1*FY
C COMMON /RIGI/I1(30,20)
C COMMON /BRACE/IBRA(30,20)*LAR(9)
C COMMON /CFL1/FCS,FY5*,CVRS,FKC(9)
C COMMON /CFL2/LX(140),JY(140),LY(140),JY(140),FCL(140,9)
C COMMON /CFL3/X(140,Y(140),Z(140),XMAX(140),YMAX(140),KSC(140)
C COMMON /KDC(140)
C COMMON /CLFL6/INO,NEQ(140),MAX(140)
C COMMON /CLFL5/KBAR(140,2)
C COMMON /CLFL7/MCD(140),XY(9),STUB(9),FCL(9),XX(9),YY(9),YK(9),INX(9)
C COMMON /CLFL8/TOR(140,2),TORM(21)
C COMMON /THFL/THEFL(30,21)*THELL(30,21,2)
C COMMON /RIBS/IRIB2(IRIB(8)
C COMMON/NOVE/NOVX,XM(21,10),PM(21,10),MSTART(10,10)
C COMMON/MEND(10,10),TREV(10,NWH(10),PACT(10),NIT(10)
C COMMON /ALS/ALFC(9),ALFY(9)
C COMMON /BRCs/IRBC(10)
C COMMON/GEN/ESTI(12,4),JYPE(30,2),KANT(30,2)
C COMMON/CPR2/CPR2(140)
C COMMON/CONS/CONS(10),IEOF,ACC(3)

IF (I60.EQ.2) GO TO 100
IF (AXC(3).EQ.2) GO TO 12

READ (LU) LIG,JIG,NIG,(P,WN(LM+1),L+1,4)*THEFL(L,P)
1 THEMELC(I,L1,2,N,1,JIG,L=1,LI)
2 (TORA(N),TORA(N),N,1,NIG)*FC1

12 CONTINUE
READ (LU) LIG,JIG,NIG,(LX(N),N,1,NIG),(JX(N),N,1,NIG)*LY(N),N=1

IF (LINK,JE=3) READ (LU) KBAR
READ (LU) LVR,FCS,FY5,CF5,CFCS,FKCL,LMFL,NOMNO,(X(N),N,1,NIG),(Y(N),N,
A1,N19),(Z(N),N,1,NIG),(XMAX(N),N,1,NIG),(YMAX(N),N,1,NIG),(KSC(N)
, BN1,NIG),(KDC(1),N,1,NIG),(FCL(1),N,1,NIG),(M1,9),(MCD(N),N,1
C1,N13,MLEL,5),XV,STUB,IF0,XX,YY,INX

C FCL(V),N,1,NIG),FETL,IRUN,IRB,
C (JYPE,LIN),KANT(LM),L=1,LIG,N=1,2,LON,BON;

C
C((SPAN(M,N),M=1,L16),N=1,J16),(LPM(M,N),M=1,L16),N=1,J16),K=1
C,(X),(SPAN(M,N),M=1,L16),(FLY(M,N),M=1,L16),N=1,J16),K=1
C(DEPTH(M,N),M=1,L16),N=1,J16),(JMVY(M,N),M=1,L16),N=1,J16),K=1
C(((I1G(M,N),M=1,L16),N=1,J16))
C(IPRA(M,N),M=1,L16),N=1,J16)
C((J=1,J9R16),(N=1,J16),N=1,J16)
C(ALFY,ALFY,IPRAC,EBRC
C)
C
IF(MVNC,GT,0) READ(LU),(PACT(J),IT(J),J=1,MOVNO)
C
1(MSTART(K,J),K=1,10),(MNED(K,J),K=1,10),J=1,MOVNO), (IREVE(J),J=1
C,2MOVNO), (NBW(J),J=1,MOVNO)
C)
C
RETURN
1
C
100 CONTINUE
C
C
RETURN
END

ILE MAP=(LO=A)
ADDRESS=BLOCK=INSERTIONS=TYPE=SIZE=NAME=ADDRESS=BLOCK=

| 13B | /CONSTS/ | REAL | 3 | IFO | 6668 | #CLFL7/ |
| 0B | /ALS/ | REAL | 4 | IF1 | 68 | #RIP1/ |
| 4B | /ALS/ | REAL | 4 | IFO | 1 | DUMMY-AF |
| 642B | / | REAL | 20 | ILINE | 11308 | #RNFL3/ |
| 425B | / | REAL | 2 | ILINK | 1148 | # |
| 6B | /CONSTS/ | REAL | 10 | INX | 7213 | #CLFL7/ |
| 6B | /CPROS/ | REAL | 140 | IO9 | 4228 | # |
| 2B | /CLFL1/ | REAL | 3 | IPRINT | 1138 | # |
| 55B | / | REAL | 600 | IREVE | 11558 | #MOV/ |
| 11508 | /BMFL/ | REAL | 10 | IRIP | 88 | #RIP/ |
| 12B | /PRCS/ | REAL | 10 | IRUN | 1218 | # |
| 115B | / | REAL | 10 | ISDLV | 38 | # |
| 116B | /REAL | REAL | 48 | ITEST | 1208 | # |
| 0B | /GEN1/ | REAL | 600 | J | 12738 | # |
| 1060B | /CLFL2/ | REAL | 1260 | JG6 | 6 | DUMMY-AF |
| 0B | /BMFL1/ | REAL | 124308 | JM6 | #BMFL2/ |
| 6B | /CLFL1/ | REAL | 1118 | JSTOP | 1118 | # |
| 3B | /CLFL1/ | REAL | 9 | JX | 2149 | #CLFL2/ |
| 1B | /BMFL1/ | REAL | 6448 | JY | #CLFL2/ |
| 1B | /CLFL1/ | REAL | 688 | JYPE | #GEN1/ |
| 1227B | / | INTEGER | K | 12528 | # |
| 0B | /BRACE/ | INTEGER | 600 | KAR | 1248 | # |
| 0B | /PRCS/ | INTEGER | 18 | KANT | 1548 | #GEN1/ |
| 103B | / | INTEGER | 6 | KBAR | 08 | #CLFL5/ |
| 7B | / | INTEGER | 6 | KDC | 15108 | #CLFL3/ |
| 74B | / | INTEGER | 6 | KSC | 12748 | #CLFL3/ |
| 6B | / | INTEGER | 6 | L | 12238 | # |
| 66B | / | INTEGER | 6 | LAR | 11508 | #BRACE/ |
| 12B | /CONSTS/ | INTEGER | 6 | LBM | 5678 | #BMFL2/ |
| 12B | / | INTEGER | 6 | LEV | 28 | # |
| 11B | / | INTEGER | 6 | LIG | 3 | DUMMY-AF |
SUBROUTINE SECBM1(I, J, IVMX, LIG, MIG, LV2, LVM)

COMMON "STATE", LEY, ISOL, LIV, LIVIN, MF, LN, LVAP, LNM, LIF, LVE, IFFR"

COMMON IZ2, IKM5, IDAT6, IDOC6, IDOS6, IDAT9, ICRI5, ICRI6, ICRI9

COMMON JSTOP, NSTOP, PFIN, PRINT, IIN, ICRI1, ICRI3, ICRI6, ICRI9, ICRI19

COMMON K1K, LIST10, 1D9, LODF, Z10, MD20, UF20

COMMON ITC1, NE7(90)

COMMON USF(20), PON(20), LIN(20)

COMMON TAKI(10), JET(10), KET(20), LET(20)

INTEGER KAK(90, 2), NME(11, 3)

COMMON/FLT1/BSTIF(986, 2), BL(986, 2), KRE(986, 2),
   IPR(986, 2), CUTF(992, 2), CLE(992, 2), CXY(992, 2),
   2INMAY(2), PHTC(992, 2), DVF(992, 2), TRANS(195, 2), C11(2), C22(2),
   3JBM(992, 2)

COMMON/ALFA/ALFA, ALFA, ALFA, ALFA, ALFA, ALFA, ALFA, ALFA, ALFA, ALFA

COMMON/HFL1/WIDTH(30, 20), DEPTH(30, 20)

COMMON/HFL2/SPAN(30, 20), WB(30, 20, 4), LBH(30, 20, 4), JMX(30)

COMMON/CFL2/LX1(140), JX(140), LY(140), JY(140), FC1(140),
   COMMON/STPK/LXPK1, STPK, STPY1, STPY2, STPY3, STPY4, MR(12),
   IFR(2), CORD(140, 2),
   14COL1, 1ICFL(140), IFBM, GRID1(140, 2), NCOL(2), MCR(140, 2),
   2CLE(140), TRUE(922), 5ST2(922), ST2Z(922), IFLAG1(140),
   6IFLAG(140)

COMMON/REIZ/UXZ1, UXZ2, UXZ3, UXZ4, UXZ5, UXZ6, UXZ7, UXZ8, UXZ9, UXZ10, UXZ11,
   UXZ12, UXZ13, UXZ14, UXZ15, UXZ16, UXZ17, UXZ18

REAL MARG(10)

INTEGER COL1, COL2(140, 2)

DIMENSION CC(18)

COMMON/THET/LBMW(1490)

DIMENSION IRB(250), JB(250), YE1(250), MKP(250), BM(21)

!BMW=30, 20!, OB2(250), BM(240)

EQUIVALENCE (DUM1, IFBB(1)), (DUM251, XPI11), (DUM450, YE111)

EQUIVALENCE (DUM4751, MKP111)

EQUIVALENCE (DUM41801, BM=111), (DUM41022, BMW=111)

EQUIVALENCE (DUM41622, ORP211), (DUM41872, INUMX)

EQUIVALENCE (DUMMY11, IRP11)

IBUG=LET(11)

CALL NEWMEN(11)

SCALE=1/PLT(28, 1)

GAP=0.047

IF(140, EQ=1) GAP=0.

IF(ISCALE, EQ=0) SCAL=10.

FACTOR=.75

MGT=125*FACTOR

IF(I=1, EQ=2) 60 TO 600

INU=0

IF(I=1, EQ=3) 16 TO 20

IF (PLT18, JX=1, 10) AND (PLT19, JX=0) 60 TO 30

DO 10 LR=1, 16

CALL RMOV(12, BMW, 2, 1)

10 CONTINUE

DO 10 CONTINUE

GO TO 30
20 CONTINUE
CALL ENCO(LIG)

30 CONTINUE
IF(IPLOT(I6, J).EQ.0) GO TO 700
DO 320 I=1, IWMX
MB=MR1(IW)
IFR=IFR2(IW)
IW=INVR4(IW)
DO 320 I=1, IFR
KA=COL1(I, IW)
IF(I.LT.IFR) MA=COL1(I+1, IW)-1
IF(I.EQ.IFR) MA=MR
DO 320 K=KA, KB
LN=(K-1) * LVMA + LV2
MK=MC(K, IW)
MK2=MK2(K, IW)
IF(0 STIF(LN, IW).LE.1.E-15.AND.IPLOT(9, J).EQ.0) GO TO 320
IF(IW.EQ.1) MARK=MOD(MK2, 1.0000)
IF(IW.EQ.2) MARK=MK2/10000
L11=MARK/100
IS1=MARK-L11+100
DPMI=DM
RATIO=.5
IF(MAT.EQ.2) GO TO 40
IF(DEPTH(L11, IS1).EQ.0.) GO TO 40
RATIO=WIDTH(L11, IS1)/DEPTH(L11, IS1)

40 CONTINUE
IF(IPUG.EQ.53)
IPRINT 852, K, MK, MK2, IW, MA, MK, L11, IS1, IFBB(L11)
F 2 FORMAT(1X, 18E12.5)
IF(IFBB(L11).EQ.2) GO TO 300
DO 280 KN=1, K
IJB=IJB5(L11, IS1, KN)
IF(IPUG.EQ.53) PRINT 852, KN, L11, IS1, IJB, IFBB(L11), INU
IF(IJB.EQ.0) GO TO 300
L12=IJB/100
IS2=IJB-L12+100
DM=EXR=5(L11, IS1, KN)/12.
DBM=DBM1+DBM
IF(WIDTH(L12, IS2).LE.1.) GO TO 280
IF(SYX(L12, IS2).EQ.0.) GO TO 280
INU=INU+1
MARK=INU*L11+IW+10000
CALL UNP(MKST(LN, IW), A1, A2, B)
MGT=125*FACTOR
BM=SPAN(L12, IS2)/(12.*SCALE)
BM1=SPAN(L11, IS1)/(12.*SCALE)
GAP2=0.
IF(NAT.EQ.2) GO TO 60
GAP=WIDTH(L12, IS2)/(12.*SCALE) + .023
GAP2=WIDTH(L12, IS2)/(12.*SCALE)
IF(IPUG.EQ.53) WR TE(6, * * * TSEC81 = * * * T$15...2F18.3*)
MARK=L11+IS1+L12+IS2+GAP1+120.0 UZ10+GAP2+120.0 UZ10
60 CONTINUE
IF(IW.EQ.2) GO TO 65

C MAIN BEAM IN X DIRECTION
SUBROUTINE SCM9/3  74/74  OPT=1  FTN 5.1+557

C

133 FORMAT(10A1)
1
DECODE(10,333,PIMSH(LT1:LSI))XCC
2
DO 70 IC=1,10
3
  IF(CC IC).EQ.1M8 GO TO 80
4
70 CONTINUE
5
80 CONTINUE
6
IC1=IC-1
7
SLEN=IC1*HGT
8
XR(INU)=DB1*COSA/SCALE+CORD(MK,1W)/SCALE
9
YR(INU)=CORD(MK,1W)/SCALE-DBM1*ALFA/(SCALE)
10
C=XR(INU)
11
C2=XR(INU)+GAP1
12
C5=C1
13
C6=RP*YR(INU)
14
BLENG=CORD(MK,1W)/SCALE+PM1*.5-.5*SLEN-HGT/3.
15
PLENG2=BLENG1+SLEN+2.*HGT/3.
16
IF(IPLOT(18J).EQ.0) GO TO 90
17
IF(IF(TF(IN1,IM)).EQ.10) GO TO 90
18
IF(IF(TF(IN1,IM)).EQ.15) GO TO 90
19
IF(RATIC.GE.5.*AND.MAT.EQ.1) GO TO 90
20
IF(C1.LT.BLENG1) GO TO 90
21
IF(C1.GT.BLENG2) GO TO 90
22
C2=C2+HGT*.047
23
GO TO 90

C

MAIN BEAM IN Y DIRECTION

85 CONTINUE

XR(INU)=CORD(MK,1W)/SCALE-DBM1*ALFA/SCALE
1
YR(INU)=DBM1*COSA/SCALE+CORD(MK,1W)/SCALE
2
BLENG=2.*HGT
3
IF(LI1.GT.9) BLENG=4.*HGT
4
C1=XR(INU)+GAP1
5
C2=XR(INU)
6
C5=XR(INU)+BM
7
C6=XR(INU)
8
BLENG1=CORD(MK,1W)/SCALE+PM1*.5-BLENG*.5-HGT/3.
9
BLENG2=BLENG1+BLENG2*.5-HGT/3.
IF(IPLOT(14J).EQ.0) GO TO 90
IF(C1.LT.BLENG1) GO TO 90
IF(C2.GT.BLENG2) GO TO 90
C1=C1+HGT*.047

90 CONTINUE

C

FIND RIGHT END OF SEC. BM
C
CHECK IF IT IS SUPP. IN MAIN BM OR IN SECOND BM.

I1L=I1L+1
1
DO 160 I1=1,1116
2
I3=I1G+I1+1
3
JMXT=JMXT(I3)-1
4
IF(JMXT.LT.0) GO TO 160
5
DO 140 I2=1,JMXT
6
CALL UNSQZHO(2.,IFBB(I3),I2,IER)
7
DO 120 KNK=1,9
8
I1L2=JLPH5(I3,I2,KNK)
9
100 CONTINUE
110 CONTINUE
120 CONTINUE
130 CONTINUE
140 CONTINUE
150 CONTINUE
160 CONTINUE

SUBROUTINE SECBM/4  74/74  OPT=1

IF(IWL2.EQ.0)GO TO 140
IF(IWL1.NE.IWL2)GO TO 120
IF(IPI.NE.I)GO TO 130
DO 100 N1=1,N1G
   HGT1=HGT
   LX1=LX(N1)
   JX1=JX(N1)
IF(IW.EQ.2)JX1=1&5(N1)
IF(IFUG.EQ.53)PRINT 858,N1,T3,I2,LX1,JX1
   858 FORMAT(* IG7 ,* BI8*)
   IF(LX1.NE.I3)GO TO 100
   IF(JX1.NE.I2)GO TO 100
   IF(N1.NE.MM)GO TO 140
   BLENG=3.*HGT
   IF(LX1.GT.9)BLENG=4.*HGT
   IF(IW.EQ.1)GO TO 9R
   DECODE(10,333,PRMSH(13,I2))CC
   DO 95 IC=1,10
      IF(CCC(IC).EQ.1H)GO TO 96
         95 CONTINUE
         96 CONTINUE
      IC1=IC-1
      BLENG=IC1*HGT1
      GO TO 137
   98 CONTINUE
   PRINT 840
   840 FORMAT( 3H***,**MESSAGE 1 SEC. BEAMS *,3H***, * ERRDR-2)
STOP
120 CONTINUE
GO TO 140

130 CONTINUE
   HGT1=.75*HGT
   BLENG=3.*HGT1
   IF(I3.GT.9)BLENG=4.*HGT1
   IF(IW.EQ.1)GO TO 138
   DECODE(10,333,PRMSH(13,I2))CC
   DO 132 IC=1,10
      IF(CCC(IC).EQ.1H)GO TO 134
132 CONTINUE
134 CONTINUE
   IC1=IC-1
   BLENG=IC1*HGT1
   GO TO 138

137 CONTINUE
   BLENG3=(CORD(N1,IW)+SPAN(I3,I2)*.5/I2)*SCALE-.5*BLENG-HGT1/3.
   GO TO 139
138 CONTINUE
   BLENG3=(CORD(MK,IW)+SPAN(I3,I2)*.5/I2)*SCALE-.5*BLENG-HGT1/3.
   139 CONTINUE
      BLENG4=BLENG3+BLENG+2.*HGT1/3.
      GO TO 190
140 CONTINUE
160 CONTINUE
180 CONTINUE
IF(MAT.EQ.2)GO TO 190
SUBROUTINE $FCBm$ / 74/74 OPT=1  FTN E:1:552  8

GAP1=WIDTH(L1+L2)/((2*L+SCALF)*L2)
GAP2=WIDTH(L1+L2)/(L2+SCALF)

190 CONTINUE
IF(IW.EQ.2160 TO 220
  C6=C6+GAP1
IF(IIFLOT(9+J*EQ.0 AND .ST. IF(LN+IW).LE.1.F-15) GO TO 240
IF(IIFLOT(9+J*EQ.0 AND .AND. (WIDTH(L1+L2)*LT.?. AND. MAT. EQ.1) )GO TO 240
IF(IIFLOT(9+J*EQ.0 AND IBM. EQ.1) GO TO 240
IF(CLE. GT. BLENG3) GO TO 240
IF(CLE. GT. BLENG4) GO TO 240
C6=C6-HGT1-.047
GO TO 240

220 CONTINUE

C
C Y DIRECTION (MAIN BEAM)
C
IF(DEPTH(L1+L2).EQ.133.) GO TO 240
IF(DEPTH(L1+L2).EQ.0.) GO TO 240
C5=C6-GAP1
IF(RACIC. EQ.5.* AND. MAT. EQ.1) GO TO 240
IF(WIDTH(L1+L2)*LT.?. AND. MAT. EQ.11) GO TO 240
IF(IIFLOT(18+J*EQ.0 AND IBM. EQ.1) GO TO 240
IF(IIFLOT(19+J*EQ.0 AND IBM. EQ.1) GO TO 240
IF(CLE. LT. BLENG3) GO TO 240
IF(CLE. GT. BLENG4) GO TO 240
C5=C5-HGT1-.047
240 CONTINUE
IF(IIFUSE.EQ.53)
  IPRINT 860,MARK,IMJ.11.11C1,C2,C5,C6,BLENG1,BLENG2,BLENG3,BLENG4
  B60 FORMAT(18B4,16B8B3)
  IF(MAT.EQ.2) GO TO 260
IF(IIFUSE.EQ.57) WRITE(6.6(*16SECB2,*5T5,*2F10.3))
MARK,L1+L2,13,12,GAP1*120,JUZ10,GAP2*120,JUZ10
C11(1)=C1-6AP2
C11(2)=C5-6AP2
C22(1)=C2
C22(2)=C6
IF(IW.EQ.1) GO TO 245
C11(1)=C1
C11(2)=C5
C22(1)=C2-6AP2
C22(2)=C6-6AP2

245 CONTINUE
CALL DASH(C11=C2,C2+1.,1.)
IF(IW.EQ.2) GO TO 247
C11(1)=C1+6AP2
C11(2)=C5+6AP2
GO TO 250

247 CONTINUE
C22(1)=C2+6AP2
C22(2)=C6+6AP2
250 CONTINUE
CALL DASH(C11=C2,C2+1.,1.)
GO TO 265

260 CONTINUE
CALL PLOT(C1,C2,3)
CALL PLT(C5,C6,C2)
26 CONTINUE

CALL SQ6BITS(2,IFRB(I2),IS2+1,IER)
OR(INU)=C2
P(INU)=C6-C2
IF(TXL.EQ.1)GO TO 270
OR(INU)=C1
P(INU)=C5-C1

270 CONTINUE
280 CONTINUE
300 CONTINUE
320 CONTINUE

DO 520 I=1,LIG
M(KK)=MKK1=0

DO 550 I=1,LIG
I13=LIG-I1+1
JM(K)=JM(K)(I13)-1
IF(KM(I12+1,LT)0)GO TO 550
DO 480 I=1,JM(K)
CALL UNSG2B(2,IFRB(I13),I12+1,IER)
IF(IUBE.EQ.53)1PRINT 855,I13,LIG,JM(K),I12+1,INU
IF(IBI.EQ.0)160 TO 500
IF(IBI.EQ.9)160 TO 480
NST1=NST+1
DN=0

DO 460 K=1,N
IF(K.EQ.1)K=1
K=K-1
IF(K.EQ.1)K=K-1

C FIND SECOND BEAM SUPPORTED IN BM (SECONDARY) MKB2
C DIRECTION IN2 OF BM MKB2
C

IF(IUBE.EQ.53)1PRINT 855,I13,JM(K),I12+1,IP1,1JL
855 FORMAT(* IQ 4*RIP)
IF(IP1.EQ.0)160 TO 480
L=I1-L1/100

IS=I1-L1-L1*100
CALL SQ6BITS(2,IFRB(I13),I12+2,IER)
CALL UNSG2B(2,IFRB(I13),I12+1,IER)
IF(IUBE.EQ.53)1PRINT 855,I1L,IBI1,L1,L1,IS1,INN,I12,IBI1
IF(IBI1.EQ.0.AND.INN.EQ.1)160 TO 480
DN=DN+1(MK112+KNN112)+12

DBM=DN=DBM
MKK1=M112+100+I12
IF(KK1.EQ.0)MKK11KK1=0
IF(ISPAN(I1,I11,LE.,E.)KNN1=1
IF(ISPAN(I1,I11,E.O.)160 TO 460
IF(ISPAN(I11,E.F.)160 TO 460

IF(KK1.EQ.0)160 TO 380
MKK=M113+100+112
DO 360 K=1,INU
MKB=MOD(MKB1+KK1,1000)
K2=MK114(KK1)/1000

IF(KK1.EQ.53)1PRINT 856,KK,INU,MKK,MKB2,MKB1(KK1),I1L
856 FORMAT(* IQ 6*B18)
390 CONTINUE

XR1(INU)=XR1(KK)*DBM1/SCALE
IF(SPAWalkerEQ.0) GO TO 360
IF(WIDTH(LI1+1,IS1)*LT.0) GO TO 440
C1=XR1(INU)+GAP1
C2=XR1(INU)
C5=XR1(INU)+PM
C6=XR1(INU)
PLENG=3.*HGT1
IF(IT13.GT.0.1) PLENG=4.*HGT1
PLENG1=XR1(KK)+EM1*5.5+BLENH-HGT1/3.
PLENG2=PLENG1+PLENG+2.*HGT1/3.
IF(IPLT(IJ1,J1),EQ.0.0) GO TO 410
IF(IPLT(IJ1,J1),EQ.0.0) GO TO 410
IF(C2+GTRUBLW2) GO TO 410
C1=C1+HGT1+0.47

410 CONTINUE

C FIND RIGHT SUPPORT (IJL1) OF FEAM IJL

IJL1=IJL+1
DO 430 I6=1,LIG
I3=I6-I6+1
JMX2=JMX(13)-1
IF(JMX2.LT.0.0) GO TO 430
DO 429 I2=1,JMX2
CALL UMSQZB2(2,IFRB(13),I2,I2,I2,I2)
DO 427 KKN=1,A
IJL2=JLM5(I3,I2,KKN)
IF(IJL2.EQ.0.0) GO TO 426
IF(IJL1.NE.IJL2) GO TO 426
BM1=SPAN(I3,I2)-I3-1,BLENH
RATIO=5
IF(MAT.EQ.2.0) GO TO 415
IF(DEP(13,I2).EQ.0.0) GO TO 415
RATIO=WIDTH(I3,I2)/DEP(13,I2)

415 CONTINUE

IF(IPI.EQ.1.0) GO TO 423
DO 422 I1=1,LIG
HGT1=HGT
LX1=LX(I1)
JX1=JX(I1)
IF(IW2.EQ.2.0) LY1=LY(I1)
IF(IW2.EQ.2.0) JX1=JY(I1)
IF(IBUG.EQ.53) PRINT 858,N1,I3,I2,LX1,JX1,KKN1,W2,I3,I2
IF(ITA.EQ.2.0) GO TO 417
GAP1=WIDIT(I3,I2)/24.*SCALE)+.23

417 CONTINUE

IF(LX1.NE.I3) GO TO 422
IF(JX1.NE.I1) GO TO 422
IF(W3.EQ.2.0) GO TO 422
DECODE(10,333,B12H(I3,I2))CC
DO 418 IC=1,10
IF(ICC(I1+IC).EQ.1) GO TO 420

418 CONTINUE
420 CONTINUE
  IC1=IC-1
  PLEN6=IC1*HGT1
  PLEN9=IC1*HGT1
  PLEN3=SCALE+PM1+5*PLEN9
  PLEN4=PLEN3+PLEN2*HGT1/3
  GO TO 432

422 CONTINUE
  PLEN1=IC1*HGT1
  IF(12.0.9)PLEN5=IC1*HGT1
  PLEN5=IC1*HGT1
  PLEN3=SCALE+PM1+5*PLEN5
  PLEN4=PLEN3+PLEN2*HGT1/3
  GO TO 434

423 CONTINUE
  HGT1=IC1*HGT
  IF(12.5.2)GO TO 427
  IF(NAME(43.133.DCIN1.1))CC
  GO 424. IC1=1.10
  IF(NAME(10.1)EQ.1H1)GO TO 425

424 CONTINUE

425 CONTINUE
  IC1=IC-1
  PLEN3=IC1*HGT1
  PLEN6=IC1*HGT1
  PLEN3=IC1*HGT1
  PLEN4=PLEN3+PLEN2*HGT1/3
  GO TO 432

427 CONTINUE
  PLEN6=IC1*HGT1
  IF(13<.0.9)PLEN6=IC1*HGT1
  PLEN3=IC1*HGT1
  PLEN4=PLEN3+PLEN2*HGT1/3
  GO TO 434

426 CONTINUE
  IF(NAME.10.EQ.1)GO TO 432
  GAP1=WIDTH(13.12)/ SCALE+823

412 CONTINUE

428 CONTINUE

430 CONTINUE

432 CONTINUE
  CS=CS-GAP1
  IF(NAME.10.EQ.1)GO TO 432
  IF(NAME.10.EQ.1)GO TO 433
  IF(NAME.10.EQ.1)GO TO 433
  IF(NAME.10.EQ.1)GO TO 433
IF(C.LT.BLEN9)GO TO 434
IF(C.GT.BLEN9)GO TO 435
C5=C5+HCT1+.047
GO TO 438
C YDIRECTION NEW BEAM RIGHT SUPPORT
C
434 CONTINUE
C6=C6+GAP1
IF(IPLLOT(4,J)*EQ.0.AND.WDTH(T+12)*LT.20.0.AND.MAT.EQ.1)GO TO 436
IF(IPLLOT(14,J)*EQ.0.AND.IBI.EQ.1)GO TO 436
IF(IPLLOT(15,J)*EQ.0.AND.IBI.NE.1)GO TO 438
IF(C.GT.BLEN9)GO TO 430
IF(C5.GT.BLEN9)GO TO 430
C6=C6+HCT1+.047
438 CONTINUE
IF(IUN.EQ.53)
IPRINT =4*KM1+II1+II2*JMN1*IPI*W2*INU*IJL1*IJL2
IC1=2*C5=C6+RM1*BLEN9*BLEN9*BLEN6
854 FORMAT(1X,*10.14F8.2)
KM1(INU)=JL1+IW3+10000
IF(MAT.EQ.2)GO TO 450
C11(1)=C1-GAP2
C11(2)=C5-GAP2
C22(1)=C2
C22(2)=C6
IF(IUN.EQ.53)WRITE(6,**(*10SEC83.6I5.2F10.3*))
IW3=(KM1+II1+II2+GAP1+GAP2+UZ10+GAPF2+120.+UZ10
IF(IUN.EQ.2)GO TO 442
C11(1)=C1
C11(2)=C5
C22(1)=C2-GAP2
C22(2)=C6-GAP2
442 CONTINUE
CALL DASH(C11,C22,2,1,1
IF(IUN.EQ.1)GO TO 446
C11(1)=C1+GAP2
C11(2)=C5+GAP2
GO TO 448
446 CONTINUE
C22(1)=C2+GAP2
C22(2)=C6+GAP2
448 CONTINUE
CALL DASH(C11,C22,2,1,1
GO TO 452
450 CONTINUE
CALL PLOT(C1,C2,3)
CALL PLOT(C5,C6,2)
452 CONTINUE
CALL SQ2BITS(2,IFRB(111),IS1,2,IFER)
BIN(INU)=C6-C2
OB2(INU)=C2
IF(IUN.EQ.2)GO TO 440
OB2(INU)=C1
BIN(INU)=C5-C1
440 CONTINUE
460 CONTINUE
SUBROUTINE SFCPM/11, 7A/7A, OPT=1

ARC CONTINUE

5 CONTINUE

DO 510 I3=1,L10
   JMY1=JMY(I3)-1
   DO 510 I2=1,JMY1
   CALL UNSQ2P(T1*IFPR(I3)+I2,1D0,IFR)
   IF(IF1),F0=0 GO TO 520

10 CONTINUE

GO TO 510

20 CONTINUE

30 CONTINUE

INUMX=INU

GO TO 700

400 CONTINUE

CALL NEWPN(1)
C SECONDARY BEAM MARKS
C

MT1=75+HGT

640 DO IMX=1,INUMX
   IFRM=M
   MKK=MCD(MKP1(IMX)+10000)
   I2=MKP1(IMX)/10000
   L1=MKK/100
   IS1=MKK-L1+100
   IF(IXM,E0,INUMX)+605
   MKK=MCD(MKP1(IMX+1)+10000)
   I2=MKP1(IMX+1)/10000
   L1=MKK/100
   IS2=MKK-L1+100
   605 CONTINUE

SLEN=3.*40+HGT1

450 IF(L1+GT.9) SLEN=4.*47+HGT1
   GAP=0.
   IF(MAT,E0,1)+60.GAP=WIDTH(L1+IS1)/(24.*SCALE)
   POS1=L1+IS1+5
   C3=OR2(IMX)*POS1
   C4=OR1(IMX)+GAP1
   IF(IXM,E0,INUMX)+605
   C5=OR1(IMX+1)-OR1(IMX)+2.*HGT1*0.47+GAP1

460 CONTINUE

IF(L2+E0,1)+620
   C3=PX1(IMX)+GAP1
   C4=OB1(IMX)+POS1
   IF(IXM,E0,INUMX)+620
   C5=PB1(IMX+1)-PB1(IMX)+2.*HGT1*0.47+GAP1

620 CONTINUE

IF(PUG,E0,53) WRITE(6,1)-*IQSEB4**,6I4,6F10.3 *)
   M84=K1+L11,IS1,IMX+INUMX*21(IMX)+81(IMX)+1,YB1(IMX)
   V+YB1(IMX+1)*GAP1+C5
   GAP+HGT1*0.47+GAP1
   IF(IXM,E0,INUMX)+605
   IF(AR1+82(IMX+1)-OB2(IMX)+GAP1)+625
   IF(C5,LE0.*,AND.IV3,EQ.IW2)+600

625 CONTINUE

IF(SLEN1,GT,AR1(IMX)+1) IFRM=2
   IF(.75+SLEN1,GT,81(IMX)) IFRM=3
```
SUBROUTINE SECBM/12 74/74  OPT=1  FN=5.1+552

IF(IP5.GE.63)PRINT 997,IMX,IMK,IM2,IFBM,C3,C4,X1(1,IMX)
1,0B2(IMX),BM1(IMX),PS1,SLIG1

997 FORMAT(4,4T6,4F8.2)
1,IF(IFBM,NE.3)GO TO 630
2,C3=PI1(IMX)*GAP1
3,C4=OR2(IMX)*IM1(IMX)*.5
4,IF(IM2.GT.2)GO TO 630
5,CA1=PI1(IMX)-GAP1
630 CONTINUE
7,CALL LABEL655,C3,C4,KX,KZ,LVM,K1,IM2,IMX,SCALE1)
8,640 CONTINUE
9,700 CONTINUE
10,CALL NEPEN(1)
11,RETURN
12,END

'LE WAF=(LC=A)
ADDRESS--BLOCK---------PROPERTIES---------TYPE--------SIZE--------NAME--------ADDRESS--------BLOCK-----

<table>
<thead>
<tr>
<th>OP</th>
<th>/ALFA/</th>
<th>REAL</th>
<th>DBM</th>
<th>3221B</th>
</tr>
</thead>
<tbody>
<tr>
<td>3222B</td>
<td><em>S</em></td>
<td>REAL</td>
<td>DRM1</td>
<td>3213B</td>
</tr>
<tr>
<td>3223B</td>
<td><em>S</em></td>
<td>REAL</td>
<td>DEPTH</td>
<td>1130B</td>
</tr>
<tr>
<td>3224B</td>
<td><em>S</em></td>
<td>REAL</td>
<td>DUM</td>
<td>1036</td>
</tr>
<tr>
<td>2585B</td>
<td>/BMFL4/</td>
<td>EQV</td>
<td>REAL</td>
<td>240</td>
</tr>
<tr>
<td>1775P</td>
<td>/THETF1/</td>
<td>EQV</td>
<td>REAL</td>
<td>693</td>
</tr>
<tr>
<td>3658B</td>
<td>/PLOT1/</td>
<td>REL</td>
<td>REAL</td>
<td>693</td>
</tr>
<tr>
<td>241B</td>
<td><em>S</em></td>
<td>REAL</td>
<td>EC5</td>
<td>116B</td>
</tr>
<tr>
<td>3237B</td>
<td>REAL</td>
<td>FACTOR</td>
<td>3162B</td>
<td></td>
</tr>
<tr>
<td>3240B</td>
<td>REAL</td>
<td>FCL</td>
<td>1060B</td>
<td></td>
</tr>
<tr>
<td>3262B</td>
<td>REAL</td>
<td>GAP</td>
<td>3320B</td>
<td></td>
</tr>
<tr>
<td>3263B</td>
<td>REAL</td>
<td>GAP1</td>
<td>3161B</td>
<td></td>
</tr>
<tr>
<td>3225B</td>
<td>REAL</td>
<td>GAP2</td>
<td>3227B</td>
<td></td>
</tr>
<tr>
<td>1750P</td>
<td>/THETF1/</td>
<td>EQV</td>
<td>REAL</td>
<td>21</td>
</tr>
<tr>
<td>3226B</td>
<td>REAL</td>
<td>GRID1</td>
<td>1305B</td>
<td></td>
</tr>
<tr>
<td>642B</td>
<td>/PLOT1/</td>
<td>REAL</td>
<td>REAL</td>
<td>20</td>
</tr>
<tr>
<td>3146B</td>
<td>REAL</td>
<td>HGT</td>
<td>3257B</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>/ALFA/</td>
<td>REAL</td>
<td>18</td>
<td>IPI</td>
</tr>
<tr>
<td>2701B</td>
<td>/PLOT1/</td>
<td>REAL</td>
<td>1984</td>
<td>IB11</td>
</tr>
<tr>
<td>2367B</td>
<td>/BMFL4/</td>
<td>REAL</td>
<td>90</td>
<td>IBR</td>
</tr>
<tr>
<td>423B</td>
<td>/BMFL4/</td>
<td>REAL</td>
<td>2</td>
<td>IBU6</td>
</tr>
<tr>
<td>448B</td>
<td>/BMFL4/</td>
<td>INTEGER</td>
<td>280</td>
<td>IC</td>
</tr>
<tr>
<td>10B</td>
<td>/BMFL4/</td>
<td>REAL</td>
<td>220</td>
<td>ICLF</td>
</tr>
<tr>
<td>18</td>
<td>/ALFA/</td>
<td>REAL</td>
<td>ICRCI</td>
<td>103B</td>
</tr>
<tr>
<td>23110B</td>
<td>/PLOT1/</td>
<td>REAL</td>
<td>1984</td>
<td>ICRI</td>
</tr>
<tr>
<td>32710B</td>
<td>/PLOT1/</td>
<td>REAL</td>
<td>1984</td>
<td>IC1</td>
</tr>
<tr>
<td>36610B</td>
<td>/PLOT1/</td>
<td>REAL</td>
<td>1984</td>
<td>ICAT</td>
</tr>
<tr>
<td>32330B</td>
<td>REAL</td>
<td>IODG</td>
<td>60B</td>
<td></td>
</tr>
<tr>
<td>52576B</td>
<td>/PLOT1/</td>
<td>REAL</td>
<td>2</td>
<td>IONS</td>
</tr>
<tr>
<td>3240B</td>
<td>REAL</td>
<td>IOD</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>52600B</td>
<td>/PLOT1/</td>
<td>REAL</td>
<td>2</td>
<td>IER</td>
</tr>
<tr>
<td>3316B</td>
<td>REAL</td>
<td>IFBB</td>
<td>88</td>
<td></td>
</tr>
<tr>
<td>3317B</td>
<td>REAL</td>
<td>IFBM</td>
<td>1304B</td>
<td></td>
</tr>
<tr>
<td>3239B</td>
<td>REAL</td>
<td>IFIRM</td>
<td>12B</td>
<td></td>
</tr>
<tr>
<td>3234B</td>
<td>REAL</td>
<td>IFLAG</td>
<td>2740B</td>
<td></td>
</tr>
<tr>
<td>99B</td>
<td>/PLOT1/</td>
<td>REAL</td>
<td>3</td>
<td>IFLA61</td>
</tr>
</tbody>
</table>
```
SUBROUTINE SIZES(IS, J, IW, C1, L, C2, LVNC, SCALF)
COMMON MAT, MEM, LEV, ISOLV, LV, LIN, INF, NLMAX, LMA, IC, LTA, TELEV, IFRM
COMMON ZZ, ZN, DAT(3), DOC(6), IDOS(6), IDAT, ICRT(6), ICRTC(6)
COMMON JSTOP, STOP, STOPT, STOPD, LINK, EC1, EC5, MAXR, TEST, IRUM, LVU, LNU
COMMON KAK, LIST(10), IO9, CODE(21), MO(29), UK(20)
COMMON TAIK(10), JET(10), XMT(20), LET(20)
INTEGER KAK(900, 2), NHE(11, 3)
COMMON/PLOT/ PSTIF(980, 2), PLEN(980, 2), MKP(900, 2), XMT(900, 2)
1 PRR(900, 2), PSTIF(992, 2), CLEN(992, 2), CX(992, 2), CZ(992, 2)
2 INWAY(2), PHIC(900, 2), DVI(992, 2), TRNS(900, 2), CI(112, 2), CI(225)
JAR(992, 1)
COMMON/ALFA/ ALFA, COS2, ID2
COMMON/THID/ MTOPS, MIND(991)
COMMON /LADG/ LIND(171), CPJ(1L), CDAT(10, 6)
COMMON/PLICE/ ANSPL(140)
COMMON/RFL/SPAN(30, 20), XB(30, 20, 4), LRM(30, 20, 4), JMX(30)
COMMON/CFL/LY(140), JX(140), LX(140), JY(140), FCL(140, 9)
COMMON/RFL1/STP1, STP2, STP3, STP4, STP5, MB1(2), MFB1(2), CYB1(140, 2),
1 COL(1), CFL(140), IFRM, GFRI(140, 2), NCOL(2), MKC(140, 2)
4 CLEV(90), IBUG, STZZ1, STZZ, IFLAG1(140), IFLAG1(140)
5 MKPLOT(140), SIDE(10), MARG, IPILOT(25, 15), JPL, IGRI(140)
6/16 IDI(140, 2), DUMMY(240)
COMMON/RFL1/WIDTH(30, 20), DEPTH(30, 20),
REAL MARG(110)

INTEGER COL(140, 2)
DIMENSION CC(10), LEV(21)
COMMON/THETFL/DUM(1490)
DIMENSION IFR(250), XB(250), B1(250), MKP(1250), RMSh(21)
1 RIMS(30, 20), O82(250), RIMC(240)
EQUIVALENCE (DUM(1), IFR(1)), (DUM(251), XB1(1)), (DUM(501), YB1(1))
EQUIVALENCE (DUM(151), MKR(101))
EQUIVALENCE (DUM(1001), RMSh(1)), (DUM(1022), RMSH(1))
EQUIVALENCE (DUM(1622), O82(1)), (DUM(1872), INUMX)
EQUIVALENCE (DUMMY(1)), RMSh(1)
IFRM = 0
FACTOR = .75

HGT = 12 / FACTOR
IF(IS, EQ, 2) 60 TO 150
IF(IS, EQ, 3) 60 TO 500

C MAIN BEAM SHAPES

RM = 1, PBLEN(LVNC, IW) = SCALF
IF(IW, EQ, 1) MARK = MOD(MKP(LVNC, IW), 1000)
IF(IW, EQ, 2) MARK = MKP(LVNC, IW) / 1000
L11 = MARK / 100
IS1 = MARK - L11 * 100
WIDTH = 8

IF(MAT, EQ, 1) WDT1 = WIDTH(L11, IS1) / (24 * SCALF)
DECODE(19, 333, BUM(11, IS1)) CC
DO 100 IC = 1, 110
100 CONTINUE
110 CONTINUE
IC1 = IC - 1
SLEN=IC1+HGT
IF(SLEN+GT+R)1FR=2

IF(.75 SLEN+GT+R)1FBM=3
IF(IFBM.EQ.3)GO TO 140
C1=C1W*SLEN+.5*ALFA-.047*COSA
C2=C2W-.047*ALFA+SLEN+.5*COSA
WIDT7=WIDT1+.25*+120.
IF(IPU.EQ.5) PRINTF(A,15197)

MARK=W1+L11;IS1;WIDT2;ALFA;COSA;C1W;C2W
IF(W1.EQ.2.AND.IS.EQ.1)GO TO 120
C1=C1W-.047*ALFA-SLEN+.5*COSA
C2=C2W+SLEN+.5*ALFA-.047*COSA
120 CONTINUE
HGT1=HGT
IF(IFBM.EQ.2)HGT1=.75*HGT
ANGLE=0.
IF(IS.EQ.4)GO TO 130
ANGLE=ASIN(ALFA)*57.2958.
IF(IF1.EQ.2)ANGLE=ANGLE+9C.
130 CONTINUE
C1=BIMSH(L11,IS1)
CALL SYMBOL(C1,C2,HGT1,C1,ANGLE,IC1)
GO TO 470
140 CONTINUE
C1=C1W+.5*HGT
C2=C2W
IF(W1.EQ.2)GO TO 145
C1=C1W
C2=C2W+.5*HGT1
145 CONTINUE
CALL SYMBOL(C1,C2,HGT,***,ANGLE,1)
GO TO 470
335 FORMAT(10A1)
C
C SECONDARY BEAM SHAPES
C
150 CONTINUE
DO 460 IMX=1,1,NUMX
IFBM=0
MKK=MOD(MKB1(IMX),10000)
IM2=MKB1(IMX)/10000
L11=MKK/100
IS1=MKK-L11*100
WIDT1=.0
IF(W1.EQ.1)WIDT1=WIDT1+WIDT(L11,IS1)/.124*SCALE)
DECOD(10,335,BIMSH(L11,IS1))CC
DO 170 IC=1,10
IF(CC(IC).EQ.1M )GO TO 1=0
170 CONTINUE
180 CONTINUE
IC1=IC-1
HGT1=.75*HGT
SLEN1=IC1+HGT1
IF(SLEN1.8T.BIM1(IMX))1FRM=2
IF(.75 SLEN1.8T.BIM1(IMX))1FBM=3
POS1=BIM1(IMX)-SLEN1+.5
POS2=.047
C3=OF2(WX)+POS1
C4=EL1(WX)+POS2
C4=C4+W1DT1
ANGLE=0.
IF(W2==Q1+1) GO TO 190
C3=XP1(WX)-POS2
C4=OF2(WX)+POS1
C3=C3+W1DT1
ANGLE=180.
190 CONTINUE
IF(IFM>=EQ.7) GO TO 210
IF(IFM<EQ.2) GOTO 270
CALL SYGOL(C3+C4, MGT1, CC1, ANGLE, IC1)
GO TO 400
210 CONTINUE
C3=EX1(WX)-.047
C4=OP2(WX)+.5*TR(WX)-.5*MGT1
IF(W2==Q2+2) GO TO 220
C3=OP2(WX)+FM(WX)+.5-.5*MGT1
C4=YB1(WX)+.047
220 CONTINUE
CALL SYGOL(C3+C4, MGT1, ***, ANGLE+1)
450 CONTINUE
460 CONTINUE
470 CONTINUE
GO TO 1000
500 CONTINUE
C COLUMN PROFILES
C
LOAD=0
KA=C1W
KB=C2W
LVM4=LVM4
C FIND HIGHEST LEVEL PER COLUMN STACK
C
N=0
DO 520 K=KA, KP
N=N+1
DO 510 L=2, LVM4
L1=LVM4-L+2
LVM=K-1)*LVM4+L1
LEVEL(N)=L1
IF(CXILVM4+IW).EQ.999.8AND.CXILVM+W).EQ.999.) GO TO 510
GO TO 520
510 CONTINUE
520 CONTINUE
DO 590 L=2, LVM4
L1=LVM4-L+2
N=0
DO 590 K=KA, KB
N=N+1
LVM=(K-1)*LVM4+L1
MK=KC(K,IM)
CJP(2)=6N
IF(L1,C,LEV(N)) GO TO 540
IF(U.EQ.L1) LEAVE TO 530
CALL STINDX(12,MIND2,91,91)
IF(MIND2(L1),LE,0) GO TO 530
CALL READMS(12,LIND2,171,L1-1)
CALL STINDX(12,LIND2,171,0)
IF(LIND2(MK+1),LE,0) GO TO 532
LDASD=1
530 CONTINUE
CALL READMS(12,CJP,2,MK+30)
C
532 CONTINUE
IF(L1,EQ.LEV(N)) GO TO 540
IF(MSPL(MK),EQ,0) GO TO 540
IF(MSPL(MK),GT,0) GO TO 533
NSPL1=APS(NSPL(MK))
MULT=MSPL1(L1-1,NSPL1)
IF(MULT,GT,1) GO TO 580
GO TO 540
535 CONTINUE
CALL UKSQ2R(1,NSPL(MK),L1-1,I,W1,IER)
IF(I=W1,GT,0) GO TO 580
540 CONTINUE
IF(INUE,EQ,55)PRINT 990,MK,L1+1,I,IP1,NSPL(MK),CJP(2)
990 FORMAT(* SIZEF=1,9,1E8,120,1X,10) 
COL=2
C1=CX(LVN,IV)/SCALE
C2=CZ(LVN,IV)/SCALE
IF(CLEN1(LVN,IV),EQ,3,15) GO TO 580
COL=1/CLEN1(LVN,IV)
IF(CLEN1(LVN,IV),LE,1E-20,GT,COL,EQ,9) GO TO 580
IF(LEV(N),EQ,1) GO TO 550
C1X=C1-.047
C17=C2+.2/SCALE
CALL PLOT(C1X,C17,3)
C2X=C1+.047
CALL PLOT(C2X,C17,2)
550 CONTINUE
DECODE(10,333,CJP(2)) IC
DO 555 IC=1,10
IF(IC(IC),EQ,1) 160 TO 560
555 CONTINUE
560 CONTINUE
IC=IC+1
ANGLE=90
CLEN6=IC1+HGT
DIMEN=ABS(CZ(LVN,IV),CZ(LVN,1+IV))/SCALE
IF(CLEN6,GT,30),CLEN6=IC1+75+HGT
CC=CJP(2)
C1X=C1-.047
C1Z=CZ-COL/62,=SCALE)-CLEN6-.5
IF(CLEN6,GT,60) GO TO 570
CALL SYMBOL(C1X,C1Z,HGT,CC1,ANGLE,IC1)
GO TO 580
570 CONTINUE
CALL SYMBOL(C1X,C1Z,HGT,++,-,ANGLE+1)
580 CONTINUE
### SUBROUTINE UNP53(I,A1,A2,P)

**COMMON/A1FA/SINA*COSA**

- **SINA** = 0
- **COSA** = 1

```plaintext
C AABBBBCCC
11 = I/1000
12 = I1+1000
E = (11-12)* .01
13 = I1/1000
14 = I1+1000
A2 = (I1-I4)* .01
15 = I3/1000
16 = I3+1000
A1 = (12-16)* .01
IF (I5 .EQ. 0) GO TO 10
SINA = -(I5-10000)* .0001
IF (I5 .LT. 10000) SINA = I5* .0001
Y = ASIN(SINA)
COSA = COS(Y)
10 CONTINUE
RETURN
END
```

---

### DATA (LO=A)

- **ADDRESS** = BLOCK
- **PROPERTY** = TYPE
- **ADDRESS** = BLOCK

<table>
<thead>
<tr>
<th>Type</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>REAL</td>
<td>I2</td>
</tr>
<tr>
<td>REAL</td>
<td>I3</td>
</tr>
<tr>
<td>REAL</td>
<td>I4</td>
</tr>
<tr>
<td>REAL</td>
<td>I5</td>
</tr>
<tr>
<td>INTEGER</td>
<td>I6</td>
</tr>
<tr>
<td>INTEGER</td>
<td>SINA</td>
</tr>
<tr>
<td>INTEGER</td>
<td>Y</td>
</tr>
</tbody>
</table>

### URES (LO=A)

- **TYPE** = ARGS
- **CLASS**

<table>
<thead>
<tr>
<th>Generic</th>
<th>Intrinsinc</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Intrinsinc</td>
</tr>
</tbody>
</table>

### FENT LABELS (LO=A)

- **ADDRESS** = PROPERTIES

<table>
<thead>
<tr>
<th>Label</th>
<th>37B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Def</td>
<td>20</td>
</tr>
</tbody>
</table>

### POINTS (LO=A)

- **ADDRESS** = ARG

<table>
<thead>
<tr>
<th>Label</th>
<th>38</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arg</td>
<td>4</td>
</tr>
</tbody>
</table>
ANGLE2=70.

\( C1 = X_{10} + H1 \times 45^\circ \)

\( C3 = Y10 \)

IF (IEND.EQ.4) \( C1 = X_{10} + H1 \times \text{COS}A \times \text{COS}A / 3.5 \)

IF (IEND.EQ.4) \( C3 = Y_{10} + H1 \times \text{COS}A \times \text{SIN}A / 3.5 \)

\( C2 = Y_{10} + H1 \)

IF (IEND.EQ.4) \( C1 = X_{10} - H1 \times \text{COS}A \times \text{COS}A / 3.5 \)

IF (IEND.EQ.4) \( C3 = Y_{10} + H1 \times \text{COS}A \times \text{COS}A / 3.5 \)

ALFA=ASIN(SINA)*57.2957

IF (IEND.EQ.4) \( \text{ANGLE}3 = -90^\circ \times \text{ALFA} \)

IF (IEND.EQ.4) \( \text{ANGLE}3 = \text{ALFA} \)

IF (IEND.EQ.4) \( \text{ANGLE}3 \text{=} 90 \text{ TO } 20 \)

IF (IEND.EQ.4) \( 160 \text{ TO } 10 \)

\( \text{ANGLE}1 = 30^\circ \)

\( \text{ANGLE}2 = -90^\circ \)

\( \text{ANGLE}3 = 30^\circ \)

IF (IEND.EQ.5) \( \text{ANGLE}3 = 90^\circ \times \text{ALFA} \)

IF (IEND.EQ.5) \( \text{ANGLE}3 = -180^\circ \times \text{ALFA} \)

C1 = X_{10} - H1 / 3.5

\( C3 = Y10 \)

IF (IEND.EQ.5) \( C1 = X_{10} + H1 \times \text{COS}A \times \text{COS}A / 3.5 \)

IF (IEND.EQ.5) \( C3 = Y_{10} + H1 \times \text{COS}A \times \text{SIN}A / 3.5 \)

\( C2 = X10 + H1 \)

IF (IEND.EQ.5) \( C1 = X_{10} + H1 \times \text{COS}A \times \text{SIN}A / 3.5 \)

IF (IEND.EQ.5) \( C3 = Y_{10} - H1 \times \text{COS}A \times \text{COS}A / 3.5 \)

10 CONTINUE

IF (IEND.EQ.1) \( 160 \text{ TO } 20 \)

C

C ** HINGED CONNECTION *************

CALL ARC(C2,Y10,RAD,ANGLE1,ANGLE2)

GO TO 20

20 CONTINUE

C

C ** RIGID CONNECTION **

C FACTOR=60

HFT=FACTOR*125

CALL SYMBOLE(C1+C3+H1+2+ANGLE3+1)

H1=5*HEI

CALL SYMBOLE(C1+C3+H1+2+ANGLE3+1)

30 CONTINUE

RETURN

50 CONTINUE

C

C ** KSC(MK)=1 SQ X(MK) **

C ** KSC(MK)=2 ROUND X(MK) **

C ** 3 RECT X(MK) Y(MK) **

C ** 9 DUMMY DRAW AN X **

C ** 10 PILASTER **

C ** 11 BETA 0 WER PARALLEL TO X FCL(MK,4) FCL(MK,5) **

C ** 12 BETA 90 WER PERPENDICULAR TO X FCL(MK,5) FCL(MK,4) **

C FACT =1.5 MAGNIFICATION FACTOR

C FACT=1.5

IF (MAT.EQ.1) FACT=1.
MK = MCC(MK*1)
ALC = ALC(MK)
COSAL = COS(ALC)
SINAL = SINAL(ALC)
IF(IEUG*EQ.*5) PRINT 998*MK,IFLAG(MK),KSC(MK)
GO TO 100
 FORMATT( * DEBUG SYMPE *** 4 )
 IF(IFLAG(MK)*NE.0) GO TO 100
 IF(KSC(MK)*EQ.10) GO TO 100
 MK = KSC(MK)
 IF(MK*EQ.*1) GO TO 90
 IF(MK*EQ.*2) GO TO 90
 IF(MK*EQ.*3) GO TO 90
 IF(KSC(MK)*EQ.9) GO TO 70
 IF(MK*EQ.*10) GO TO 70
 IF(KSC(MK)*EQ.11) GO TO 60
 C
 KSC(MK) EQ 12
 C
 FLAN = FACT*FLM(MK*5)/(12.*SCALE)
 DEP = FACT*FLM(MK*4)/(12.*SCALE)
 X1 = X10 + DEP*5*SINAL + FLAN*5*COSAL
 Y1 = Y10 + DEP*5*COSAL + FLAN*5*SINAL
 X2 = X1 + FLAN*COSAL
 Y2 = Y1 + FLAN*SINAL
 X4 = X10 + DEP*5*SINAL + FLAN*5*COSAL
 Y4 = Y10 + DEP*5*COSAL + FLAN*5*SINAL
 X3 = X4 + FLAN*COSAL
 Y3 = Y4 + FLAN*SINAL
 X5 = X10 + DEP*5*SINAL
 Y5 = Y10 + DEP*5*COSAL
 X6 = X10 + DEP*5*SINAL
 Y6 = Y10 + DEP*5*COSAL
 CALL PLOT(X1, Y1)
 CALL PLOT(X2, Y2)
 CALL PLOT(X3, Y3)
 CALL PLOT(X4, Y4)
 CALL PLOT(X5, Y5)
 CALL PLOT(X6, Y6)
 IF(IEUG*EQ.5) AND MK*EQ.1
 PRINT 998*MK, KSC(MK), X10, Y10, DEP*FLAN*COSAL, SINAL + ALFAC(MK)
 2*X1 + Y1 + X2 + Y2
 996 FORMATT( * DEBUG SYMP *** 2 )
 IFLAG(MK) = 1
 GO TO 100
 C
 KSC(MK) EQ 11
 C
 CONTINUE
 DEP = FACT*FLM(MK*4)/(12.*SCALE)
 FLAN = FACT*FLM(MK*5)/(12.*SCALE)
 X1 = X10 + DEP*5*COSAL + FLAN*5*SINAL
 Y1 = Y10 + DEP*5*SINAL + FLAN*5*COSAL
 X2 = X1 + FLAN*SINAL
 Y2 = Y1 + FLAN*COSAL
 X4 = X10 + DEP*5*COSAL + FLAN*5*SINAL
 Y4 = Y10 + DEP*5*SINAL + FLAN*5*COSAL
 X5 = X4 + FLAN*COSAL
 Y3 = Y4 + FLAN*COSAL
SUBROUTINE SVM5/4 74/7A OPT=1

X5=X10-DEP.*5*COSAL
X2=X10-TP.*5*SINAL
X6=X10-DEP.*5*COSAL
Y6=Y10+DEP.*5*SINAL
CALL PLOT(X1,Y10,3)
CALL PLOT(X2,Y2,2)
CALL PLOT(X5,Y5,3)
CALL PLOT(X6,Y6,2)
CALL PLOT(X1,Y1,2)
CALL PLOT(X4,Y4,2)
IF(INU.EQ.5.AND.MK.EQ.6)
PRINT 288,MK,KSC(MK),X10,Y10,DEP,FLAN,COSAL,SINAL,ALFAC(MK),
Y1,Y2,Y3,Y4,Y5,Y6
IFLAG(MK)=1
GO TO 100

C DUMMY

70 CONTINUE
HGT=YSIZE/(100.*SCALE)
CALL SYMBOL(C1,C2,A*HGT,20.*-1)
IFLAG(MK)=1
GO TO 100
70 CONTINUE

C PILASTERS

IF(Y(MK).EQ.0)GO TO 85
GO TO 65

C ROUND CONCRETE COLUMN X(MK)

85 CONTINUE
RAD=FACT*X(MK)/(24.*SCALE)
CALL ARC(X10,Y10,RAD,0.*360.,)
IFLAG(MK)=1
GO TO 100

C RECT. CONCRETE COLUMN Y(MK) Y(MK)
C SQ CONCRETE COLUMN X(MK) Y(MK)
C
90 CONTINUE
FLAN=FACT*Y(MK)/(12.*SCALE)
DEP=FACT*X(MK)/(12.*SCALE)
X1=X10-DEP.*5*COSAL-FLAN.*5*SINAL
Y1=Y10-DEP.*5*SINAL+FLAN.*5*COSAL
X1=X1+FLAN*SINAL
Y2=Y1-FLAN*COSAL
X4=X10+DEP.*5*COSAL+FLAN.*5*SINAL
Y4=Y10+DEP.*5*SINAL-FLAN.*5*COSAL
X3=X4-FLAN*SINAL
Y3=Y4+FLAN*COSAL
X5=X10-DEP.*5*COSAL
Y5=Y10-DEP.*5*SINAL
X6=X10+DEP.*5*COSAL
Y6=Y10+DEP.*5*SINAL
CALL PLOT(X1,Y1,3)
**SUPROUTINE SYMPS**

```
CALL PLOT(X@Y@P@2)
CALL PLOT(X@Y@Y@2)
CALL PLOT(Y@Y@Y@2)
IFLAG(W@1) = 1
GO TO 1°0
C
C CODE RETURN
C
100 CONTINUE
END
```

<table>
<thead>
<tr>
<th>ADDRESS</th>
<th>BLOCK</th>
<th>PROPERTIES</th>
<th>TYPE</th>
<th>SIZE</th>
<th>NAME</th>
<th>ADDRESS</th>
<th>BLOCK</th>
</tr>
</thead>
<tbody>
<tr>
<td>1E</td>
<td>/CONSTS/</td>
<td>REAL</td>
<td>1</td>
<td>FCL</td>
<td>10600</td>
<td>/CLFL2/</td>
<td></td>
</tr>
<tr>
<td>726P</td>
<td>/PMFL1/</td>
<td>REAL</td>
<td>0P</td>
<td>FC1</td>
<td>00</td>
<td>/CLFL1/</td>
<td></td>
</tr>
<tr>
<td>722P</td>
<td>/PMFL1/</td>
<td>REAL</td>
<td>0</td>
<td>FCE</td>
<td>00</td>
<td>/CLFL1/</td>
<td></td>
</tr>
<tr>
<td>09</td>
<td>/PMFL1/</td>
<td>REAL</td>
<td>140</td>
<td>GMC</td>
<td>38</td>
<td>/CLFL1/</td>
<td></td>
</tr>
<tr>
<td>0P</td>
<td>/PMFL1/</td>
<td>REAL</td>
<td>4</td>
<td>FLAM</td>
<td>1720</td>
<td>/PMFL1/</td>
<td></td>
</tr>
<tr>
<td>4P</td>
<td>/PMFL1/</td>
<td>REAL</td>
<td>4</td>
<td>FLY</td>
<td>19</td>
<td>/PMFL1/</td>
<td></td>
</tr>
<tr>
<td>714P</td>
<td>/PMFL1/</td>
<td>REAL</td>
<td>18</td>
<td>25210</td>
<td>/PMFL1/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>715P</td>
<td>/PMFL1/</td>
<td>REAL</td>
<td>10</td>
<td>GRID1</td>
<td>13590</td>
<td>/PMFL1/</td>
<td></td>
</tr>
<tr>
<td>716P</td>
<td>/PMFL1/</td>
<td>REAL</td>
<td>10</td>
<td>HR1</td>
<td>1720</td>
<td>/PMFL1/</td>
<td></td>
</tr>
<tr>
<td>3650B</td>
<td>/PMFL1/</td>
<td>REAL</td>
<td>1960</td>
<td>MCL</td>
<td>711P</td>
<td>/PMFL1/</td>
<td></td>
</tr>
<tr>
<td>642B</td>
<td>//</td>
<td>REAL</td>
<td>2P</td>
<td>IEB</td>
<td>17240</td>
<td>/PMFL1/</td>
<td></td>
</tr>
<tr>
<td>720P</td>
<td>/PMFL1/</td>
<td>REAL</td>
<td>1960</td>
<td>IRA</td>
<td>0P</td>
<td>/PMFL1/</td>
<td></td>
</tr>
<tr>
<td>27010P</td>
<td>/PMFL1/</td>
<td>REAL</td>
<td>1984</td>
<td>IRCP</td>
<td>08</td>
<td>/PMFL1/</td>
<td></td>
</tr>
<tr>
<td>2367B</td>
<td>/PMFL1/</td>
<td>REAL</td>
<td>1984</td>
<td>IRUG</td>
<td>00</td>
<td>/PMFL1/</td>
<td></td>
</tr>
<tr>
<td>423P</td>
<td>/PMFL1/</td>
<td>REAL</td>
<td>2</td>
<td>ICFL</td>
<td>10700</td>
<td>/PMFL1/</td>
<td></td>
</tr>
<tr>
<td>448B</td>
<td>/PMFL1/</td>
<td>REAL</td>
<td>2</td>
<td>ICRC</td>
<td>10390</td>
<td>/PMFL1/</td>
<td></td>
</tr>
<tr>
<td>0P</td>
<td>/PMFL1/</td>
<td>REAL</td>
<td>2</td>
<td>ICRI</td>
<td>75</td>
<td>/PMFL1/</td>
<td></td>
</tr>
<tr>
<td>10P</td>
<td>/PMFL1/</td>
<td>REAL</td>
<td>2</td>
<td>INAT</td>
<td>8</td>
<td>/PMFL1/</td>
<td></td>
</tr>
<tr>
<td>1R</td>
<td>/PMFL1/</td>
<td>REAL</td>
<td>2</td>
<td>IPDC</td>
<td>600</td>
<td>/PMFL1/</td>
<td></td>
</tr>
<tr>
<td>727B</td>
<td>/PMFL1/</td>
<td>REAL</td>
<td>2</td>
<td>IPDS</td>
<td>660</td>
<td>/PMFL1/</td>
<td></td>
</tr>
<tr>
<td>0P</td>
<td>/PMFL1/</td>
<td>REAL</td>
<td>140</td>
<td>IPD</td>
<td>20</td>
<td>/PMFL1/</td>
<td></td>
</tr>
<tr>
<td>23110B</td>
<td>/PMFL1/</td>
<td>REAL</td>
<td>1984</td>
<td>IEND</td>
<td>1</td>
<td>DUMMY-AR</td>
<td></td>
</tr>
<tr>
<td>2R</td>
<td>/PMFL1/</td>
<td>REAL</td>
<td>128</td>
<td>IEOF</td>
<td>/PMFL1/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>32710B</td>
<td>/PMFL1/</td>
<td>REAL</td>
<td>1984</td>
<td>IFRM</td>
<td>13040</td>
<td>/PMFL1/</td>
<td></td>
</tr>
<tr>
<td>36610B</td>
<td>/PMFL1/</td>
<td>REAL</td>
<td>1984</td>
<td>IFRM</td>
<td>128</td>
<td>/PMFL1/</td>
<td></td>
</tr>
<tr>
<td>717B</td>
<td>/PMFL1/</td>
<td>REAL</td>
<td>2</td>
<td>IFLAG1</td>
<td>25240</td>
<td>/PMFL1/</td>
<td></td>
</tr>
<tr>
<td>52576B</td>
<td>/PMFL1/</td>
<td>REAL</td>
<td>2</td>
<td>IFLAG1</td>
<td>25240</td>
<td>/PMFL1/</td>
<td></td>
</tr>
<tr>
<td>721B</td>
<td>/PMFL1/</td>
<td>REAL</td>
<td>2</td>
<td>IFLAG2</td>
<td>/PMFL1/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5280B</td>
<td>/PMFL1/</td>
<td>REAL</td>
<td>2</td>
<td>IFO</td>
<td>6660</td>
<td>/PMFL1/</td>
<td></td>
</tr>
<tr>
<td>720B</td>
<td>/PMFL1/</td>
<td>REAL</td>
<td>2</td>
<td>IFR2</td>
<td>6</td>
<td>/PMFL1/</td>
<td></td>
</tr>
<tr>
<td>550</td>
<td>/PMFL1/</td>
<td>REAL</td>
<td>3</td>
<td>IGI</td>
<td>00</td>
<td>/PMFL1/</td>
<td></td>
</tr>
<tr>
<td>733B</td>
<td>/PMFL1/</td>
<td>REAL</td>
<td>2</td>
<td>IGRID</td>
<td>44200</td>
<td>/PMFL1/</td>
<td></td>
</tr>
<tr>
<td>1138B</td>
<td>/PMFL1/</td>
<td>REAL</td>
<td>600</td>
<td>IGRID1</td>
<td>42048</td>
<td>/PMFL1/</td>
<td></td>
</tr>
<tr>
<td>46412B</td>
<td>/PMFL1/</td>
<td>REAL</td>
<td>1984</td>
<td>ILINE</td>
<td>11300</td>
<td>/PMFL1/</td>
<td></td>
</tr>
<tr>
<td>12R</td>
<td>/PMFL1/</td>
<td>REAL</td>
<td>10</td>
<td>ILINK</td>
<td>118</td>
<td>/PMFL1/</td>
<td></td>
</tr>
<tr>
<td>115B</td>
<td>//</td>
<td>REAL</td>
<td>2</td>
<td>INWAY</td>
<td>25100</td>
<td>/PMFL1/</td>
<td></td>
</tr>
<tr>
<td>116B</td>
<td>//</td>
<td>REAL</td>
<td>3</td>
<td>INX</td>
<td>7218</td>
<td>/PMFL1/</td>
<td></td>
</tr>
<tr>
<td>0R</td>
<td>/PMFL1/</td>
<td>REAL</td>
<td>4</td>
<td>IO9</td>
<td>228</td>
<td>/PMFL1/</td>
<td></td>
</tr>
<tr>
<td>0R</td>
<td>/PMFL1/</td>
<td>REAL</td>
<td>600</td>
<td>IPRINT</td>
<td>11550</td>
<td>/PMFL1/</td>
<td></td>
</tr>
<tr>
<td>723B</td>
<td>REAL</td>
<td>2</td>
<td>IREVE</td>
<td>11550</td>
<td>/PMFL1/</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
END
120686
FIN