1983

A facility layout program using computer graphics

Ronald L. Ketcham
Iowa State University

A facility layout program using computer graphics

by

Ronald L. Ketcham

A Thesis Submitted to the

Graduate Faculty in Partial Fulfillment of the

Requirements for the Degree of

MASTER OF SCIENCE

Co-majors: Industrial Engineering
Economics

Signatures have been redacted for privacy

Iowa State University
Ames, Iowa

1983

TABLE OF CONTENTS

I. STATEMENT OF PROBLEM

This thesis develops a set of computer assisted facility layout programs utilizing the technology of computer graphics. These programs are collectively referred to as FLAG (Facility Layout Algorithm using Graphics). FLAG is primarily a construction layout procedure that enhances both the role of the computer and the engineer in computerized facility design.

Facility layout programs, to date, have been extremely limited in the role they play in the development of the actual plant layout. Most of these programs yield as output, a block diagram that specifies only the desired relative positions of workcenters in relation to each other. The design is still several person-hours from its final configuration.

FLAG programs can be incorporated into the complete facility layout process. They assist the engineer in designing workcenters, establishing aisles, and positioning workcenters within the facility. FLAG layouts include details, such as positions of machines, workbenches, tool cabinets, and operators.

FLAG also gives the engineer an expanded role when opting to use a computer for facility design. Most present computerized plant layout programs are of the batch mode type. They require only an initial input by the user. These programs then generate layout designs by established quantitative decision algorithms, which by nature tend to be oversimplifications of the realistic criteria that is required for good design.

FLAG does use a design algorithm, but to a more limited extent.

The FLAG algorithm places workcenters in the layout one at a time. At each step, the user is asked to adjust the design to reflect qualitative criteria not intrinsic to the design algorithm. To this extent, FLAG is both a construction and improvement procedure.

FLAG requires more input by the user, but results in a more realistic and usable output. Therefore, FLAG is referred to as a computer assisted, as opposed to computerized, facility layout program. FLAG requires the engineer to use more specific design skills on an interactive basis. It is a tool for the design engineer to use in developing the best facility layout.

Using computer graphics, FLAG is able to relax several constraints that have historically limited the utility of previous layout programs. Such constraints have included the following:

1. All departments (or workcenters) have been assumed to be either rectangular or symmetrical. FLAG relaxes both of these requirements. Workcenters may assume any straight line segments. For example, a workcenter with an L-shaped perimeter is allowable.

2. Material flows have been measured between the center-points, or centroids, of departments. FLAG measures material flow from estimated points of departure and entry between departments. This makes the orientation of workcenters to each other a relevant factor.

3. Existing plant layout programs have assumed departmental shapes to be completely flexible. FLAG assumes that departments have an internal skeleton of machines and other elements that determines and fixes their shapes.

4. Departments have been denoted as blocks with no internal details given. FLAG interactively assists the engineer in designing departmental interiors. The final FLAG layout details departmental interiors, as opposed to mere outlines.

5. Aisles external to departments have traditionally been ignored. FLAG permits the user to designate aisles between departments.

6. Other facility layout programs design plants on a grid, with grid blocks of a specified rectangular area. Departmental areas are required to equal some multiple of this grid block size. FLAG designs the layout on a continual plane. Therefore, no restrictions are made on departmental area.

7. Historically, the designer has not been able to examine plant layouts at interim design stages. FLAG permits interim viewing, which enhances the interactive process between computer and user.

The FLAG layout discipline also has some unique features. FLAG recognizes that the facility to be designed may manufacture several products. It further recognizes that the desired layout may change with respect to different proposed product mixes. FLAG permits the user to develop separate layouts to compare effects of different product mixes. FLAG programs are structured as separate task modules which can be accessed independently, providing flexibility of output and easy expandability of the programs.

This thesis describes the development and use of FLAG. An evaluation that compares FLAG to the widely used computerized facility layout programs is also presented. The evaluation is made on the basis of required user input time, optimality criteria, validity of move/cost relationships, and the quality and utility of program outputs.

A variety of literature has been written that documents the operation of existing computerized layout programs. A review of this literature is presented in the following chapter.

II. REVIEW OF RELEVANT LITERATURE

A. Overview

The use of computers to layout industrial facilities is not a new concept. The first computerized facility layout programs were developed in the 1960s, and have since proliferated. The most widely used as listed by Thompkins and Moore [22] are:

CRAFT - Computerized Relative Allocation of Facilities,

COFAD - COmputerized FAcilities Design,

PLANET - Plant Layout ANalysis and Evaluation Technique,

CORELAP - COmputerized RElationship LAyout Planning, and

ALDEP - Automated Layout DEsign Program.

Examples of outputs of the above programs are presented in Appendix I.

A list of some of the less popular programs as surveyed by Moore [15] is presented below.

CASS	LAYOUT
COLO 2	LSP
COMP 2	MAT
COMSBUL	MUSTLAP 2
DOMINO	OFFICE
FRAT	PLAN
GENOPT	PREP
GRASP	RMA
IMAGE	RUGR
KONUVER	SISTAP
LAYADAPT	SUMI
LAYOPT	TSP

There is another class of programs that are altered versions of previously developed programs. For example, Nelson [17] developed an altered version of ALDEP, called OPDEP (OPtimal DEsign Program). Also, CRAFT IV and CRAFT M [8] are altered versions of the original CRAFT program, which was first presented by Buffa, Armour and Vollmann [4].

Before reviewing in detail the significant facility layout programs, it is helpful to define terms. To begin, Francis and White [7] have divided computer algorithms into construction and improvement algorithms. Construction algorithms begin the departmental layout process from scratch. They are primarily used to layout new facilities and represent the most common algorithms. Major examples of construction algorithms include PLANET, CORELAP, and ALDEP.

Improvement algorithms must begin with some form of an initial layout. The algorithm then seeks to improve the layout in terms of some criteria by interchanging departments or activities. CRAFT and COFAD are examples of major improvement algorithms.

This thesis will further divide facility layout programs into the dual classification of computerized facility layout programs and computer assisted facility layout programs. The computerized facility layout program divides the process into two distinct steps. The user initially inputs all data required by the program. The program then does all analysis, with only moderate, if any, additional user input.

Alternatively, the computer assisted facility layout program does not make the above two step distinction in the layout process. The user inputs the data interactively during the course of the layout process. This permits the user to work with the program by making decisions based on qualitative factors not easily programmed.

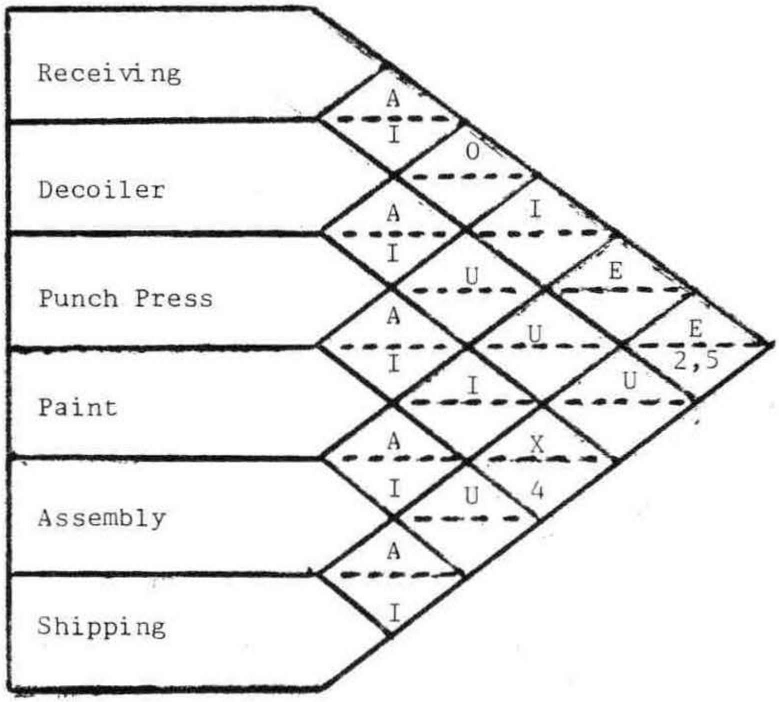
Three other definitions of terminology include activity relationship chart, Muther's six-step priority closeness rating scale, and layout discipline. These terms are discussed separately below.

The activity relationship, or REL chart, details in matrix form the coded input which denotes the relationship that exists between each department (workcenter) pair. Some form of this data is required by all facility layout programs. As stated by Moore [15], "In a problem where more than one activity is to be located, it is impossible ... to conceive of a computer working on the problem without this type of data as input, either in discrete or continuous form."

Muther's six-step priority closeness scale, as defined by Francis and White [7], is named for its creator, Richard Muther. It relies upon a common six letter code (A, E, I, O, U, and X) which is used in the input of REL chart data. The different letters are associated with different desired levels of closeness. The standard code for the six-step priority closeness is found in the rating table illustrated in Figure 2.1. Of note is the special notation of the X code, which indicates that it is undesirable to have two departments located close to each other. The interpretations on the A, E, I, O, U, and X codes are not standard among facility layout programs. Nelson [17] for example, suggests a code interpretation for OPDEP that differs from that in ALDEP.

Finally, the term layout discipline denotes the algorithmic procedure by which the computer makes the basic layout decisions. This refers to what department should be placed in the layout next, and where to put it.

Five major facility layout programs are reviewed in detail in the following sections. Some of the less popular programs that have unique features are also discussed.



Code	Reason
1	Material flows
2	Common personnel
3	Convenience
4	Noise
5	Dock location
6	Floor supports
.	.
.	.
.	.

Rating	Description
A	Absolutely necessary
E	Especially important
I	Important
O	Ordinary closeness, O.K.
U	Unimportant
X	Undesirable

Figure 2.1. Example of activity relationship (REL) chart using Muther's six-step priority closeness rating

B. CRAFT: Computerized Relative Allocation of Facilities Techniques

CRAFT is an improvement algorithm and was developed in 1963 by Elwood S. Buffa, Gordon C. Armour, and Thomas E. Vollmann [4]. The objective was to develop a facility layout approaching minimum transportation cost. This transportation cost is obtained by multiplying the figures from a user input from-to chart by figures in an input move-cost chart. This product is then multiplied by the rectilinear distances between the centroids of plant workcenters.

For example, suppose 500 moves per unit time are made between two workcenters and that each move costs $.50 per foot. In the final layout, suppose the workcenter centroids are ten feet from each other. The transportation costs between these two workcenters are then calculated as:

500 moves/unit time x 10 feet/move x $.50/foot = $2500/unit time

CRAFT attempts to minimize the sum of these transportation costs. Thompkins and Moore [22] have stated that the results from CRAFT, are generally good. These facts may be the reasons that Hicks and Cowan [8] listed CRAFT as the most widely used of all facility layout programs.

CRAFT is limited by several assumptions, as detailed by Thompkins and Moore [22]. Material handling methods must be specified before using CRAFT. Move-costs are assumed to be certain and independent of material handling equipment utilization. Move-costs are also assumed to be linearly related to length of the moves. All material flows are assumed to be between workcenter centroids and rectilinear.

CRAFT inputs consist of a parameter card which specified problem to be solved and any of CRAFT's options the user may elect to use. CRAFT also requires from-to chart and move-cost data. Since CRAFT is an improvement program, an initial layout must also be specified. The CRAFT layout discipline is detailed on the following page:

1. Determine department centroids of current layout.

2. Store rectilinear distances between departments i and j in a distance chart. (D_{ij}).

3. Calculate transportation cost between all departments i and j (TC_{ij}).

$$TC_{ij} = FT_{ij} \times MC_{ij} \times D_{ij} \qquad (2.1)$$

FT_{ij}: From-to chart value for departments i and j.

MC_{ij}: Move-cost chart value for departments i and j.

4. Departments with equal area or common borders are then considered for interchange to see if such a switch will reduce transportation costs.

5. The interchange giving the greatest transportation cost reduction is implemented, and the algorithm returns to step 1. If no interchanges can be found in the layout that reduces the transportation cost, the algorithm is terminated.

CRAFT output includes a reprint of all input values together with a layout pattern for each CRAFT iteration. Each layout pattern includes a total transportation cost and a cost reduction figure. A sample CRAFT output is presented in Appendix I.

Apple [2] has listed several advantages and limitations of the CRAFT program. Some of his remarks have relatively more merit than others. CRAFT's advantages include short CPU time requirements and potential use for office layouts. The disadvantages of CRAFT are more significant. CRAFT may not find the best layout by switching only two or three, departments at a time. It also does not allow preassigned workcenters. Even more limiting is that switched departments must be the same size, adjacent to each other, and border on a common department. Input is cumbersome and limited to 40 departments. Apple also suggests that CRAFT is flawed because the output is not directly usable, and first requires hand adjustment.

C. COFAD: COmputerized FAcilities Design

COFAD is very similar to CRAFT except for one major difference. COFAD allows for different material handling methods in making any particular move. Therefore, not only does COFAD select a plant layout, but it also allows selection of material handling methods. The objective of COFAD is to develop a layout and materials handling system which approaches a minimal material handling cost.

There is a secondary distinction between COFAD and CRAFT is detailed by Thompkins and Moore [22]. COFAD allows the assumption that parts may travel in either rectilinear paths or straight lines. The straight line assumption is more representative of some conveyor type material handling systems.

COFAD's input requirements include a parameter card specifying the problem at hand, any of COFAD's options which are to be used, and a from-to chart for each mobile material handling system alternative. Also required is a description of material handling equipment alternatives available for each move, and an indication whether the move is to be straight line or rectilinear. Finally, cost data for all material handling equipment alternatives and an initial layout are necessary. The COFAD layout discipline is described below:

1. Determine the departmental centroids of current layout.

2. Determine the move-cost (MC) for each move for each material handling alternative. This is dependent on the type of material handling equipment.

   a. fixed path equipment:

   $$MC_{ijk} = VC_{ijk} \times M_{ij} + NVC_{ijk} \qquad (2.2)$$

   VC - variable cost ($/feet)
   M - movelength (feet), calculated by COFAD
   MVC- non-variable cost ($)
   i - from department
   j - to department
   k - material handling type

   b. mobile equipment:

   $$MC_{ijk} = VC_{ijk} \times MT_{ijk} + MVC_{ijk} \times EU_{ijk} \qquad (2.3)$$

   MT - move time (hours)
   EU - equipment utilization for move (%), calculated by COFAD

3. Select material handling equipment for each move. This is the material handling equipment with the smallest move-cost.

4. First improvement phase - this phase interchanges material handling equipment for a move, if such an interchange will improve utilization of an equipment type that has been selected in step 3.

5. Second improvement phase - this phase examines department interchanges that will reduce transportation cost. When an interchange is found that reduces the transportation cost, it is implemented.

6. Test for steady state - COFAD compares new layout with last iteration layout. The cost of materials handling system and the number of changes in material handling equipment assignments may vary by less than an initially specified steady state percentage. If so, the algorithm is terminated. If not terminated, the process returns to Step 1.

7. Sensitivity analysis - after termination of the main algorithm loop, the flow volumes on the from-to charts are varied by some initially set percentage. The main algorithm is restarted. The purpose of this is to verify that the previous steady-state solution is indeed

> correct. The procedure protects against the design of a facility which is not able to handle flow variances.

A sample of a COFAD output is presented in Appendix I. COFAD has some obvious advantages over CRAFT as an improvement program. COFAD considers alternative material handling systems and the choice of using straight line or rectilinear material flows. Obvious disadvantages include the increased complexity, high computation time, and more extensive required inputs.

D. PLANET: Plant Layout ANalysis and Evaluation Technique

PLANET is a construction algorithm which requires the same basic input as CRAFT. According to Thompkins and Moore [22], PLANET is the most flexible of the principal facility layout programs.

PLANET requires material handling systems to be selected prior to utilization of the program. Move-costs are assumed to be independent of material handling system utilization and are linearly related to the length of the moves. As usual, all flows between workcenters are assumed to originate and terminate at the departmental centroids, and direction is not important. Finally, PLANET does not assume a particular building shape, and therefore, layouts often have irregular exterior perimeters.

The user of PLANET gives all departments a placement priority number from one to nine (one is the highest priority). This priority refers to the sequence of placement into the layout location. The user has the option of entering from-to chart data, or designating the relationship between departments by use of a penalty chart. If the user chooses a penalty chart input, the penalties range from -9 to 99. A -9 penalty indicates an undesirable closeness between departments.

With the input of this data, PLANET utilizes one of three placement algorithms:

1. Selection method A:

   a. Review all placement priorities of all departments that have yet to be placed.

   b. Of those departments within highest placement priority group, select the department with the highest move-cost, or penalty, with one other department. Place this department in the layout.

   c. Repeat step a.

   d. Of those unselected departments within the highest remaining placement priority group, select the department with the highest move-cost, or penalty, with one of those departments already placed. Place this newly selected department next to the department with the highest move-cost, or penalty, relationship.

   e. Repeat steps c and d until all departments have been placed in the layout.

2. Selection method B:

   a. Select the first two departments by the method specified in Steps 1a and 1b.

   b. Place the department in the highest placement priority group with the highest sum of move-costs, or penalties, with all other previously placed departments.

   c. Repeat step b, until all departments have been placed in the layout.

3. Selection method C:

   a. Place the department in the layout that is the department in the highest placement priority group with the highest sum of move-costs, or penalties, with all other departments.

   b. Step a is repeated for all unselected departments until all have been placed.

In all three selection methods, the positioning routine is identical.

The first two selected departments are set adjacent to each other in the layout field. Additional departments are positioned at the location that minimizes the increase in material handling costs. The PLANET output consists of a documented version of input, a listing of internally utilized data matrices, and a copy of the layout generated by each of the selection methods. A sample of this output is presented in Appendix I.

Apple [2] has suggested that one of the advantages of PLANET is that it "requires interaction between computer routine and engineer, to exercise judgment." However, this interaction is very limited. PLANET does have the advantage that it is flexible in application to situations where quantifiable relationships exist between activities. It is also flexible in the choice of method for selecting and placing departments.

E. CORELAP: COmputerized RElationship LAyout Planning

CORELAP was developed by Lee and Moore [13], and represents the first effort at development of a construction algorithm. It is also one of the simplest algorithms, and requires the least amount of input data of all those programs discussed thus far. However, the data that that are required are difficult to obtain. A unique feature of CORELAP is that it measures distances between departments at the closest points, as opposed to centroids.

There are also similarities to the programs discussed earlier. The layout shape generated is irregular. The design of the layout is not based on the material handling systems or the relative move-costs. For layout scoring, the flows are assumed to follow the shortest routes between departments.

The main input is the REL chart using the Muther's six-step priority closeness scale (A, I, I, O, U, X). CORELAP assigns the values 6, 5, 4, 3, 2, and 1 to the rating scale (e.g., A=6, E=5, etc.). CORELAP uses this input to calculate the total closeness rating (TCR) for each department. The TCR is the sum of numerical values representing the A, E, I, O, U, X codes. The layouts are scored as specified by Equation 2.4.

$$TCR = \sum_{j=1}^{n} \sum_{i=1}^{n} D_{ij} CR_{ij}; \quad i \neq j \qquad 2.4$$

n = number of departments.

D_{ij} = shortest route between departments, i and j.

CR_{ij} = the numerical representation of the closeness rating (6, 5, 4, 3, 2, 1) between two departments, i and j, from the REL chart.

The layout discipline utilized by CORELAP, as detailed by Thompkins and Moore [22], will now be described.

1. The department with the highest TCR is placed in the center of the layout. If there is a tie, the department with the largest area is selected.

2. Remaining departments are checked to find the highest priority closeness rating with the department just placed. If a tie again occurs, the department with the largest area is selected.

3. Remaining departments are scanned for an "A" priority closeness rating with the first department placed. If none is found, then the remaining departments are searched once again for an "A" priority closeness rating with the second department placed. If none is found, the procedure is repeated again until relationships with the last department placed are examined. If no department has been selected, the process begins again looking for an "E" relationship with the first department place, and so on.

4. When a department has been selected by the method detailed in step 3, it is placed in the layout. The placement of a department is determined by use of a placement rating. The placement rating is the sum of numerical closeness rating with departments on the boundary of the department being placed. A placement rating is calculated for each position available along the perimeter of the existing layout. The department is placed in the position with the highest placement rating.

5. Steps 3 and 4 are repeated until all departments have been placed.

As stated by Thompkins and Moore [22], the output of CORELAP consists of a copy of all input, a listing of the modified input utilized by CORELAP, an order and distance table, and a final layout. CORELAP also has the option of utilizing a CALCOMP plotter for the final version of the layout.

Like the programs discussed so far, CORELAP is primarily a batch mode program. However, there is a newer interactive version of CORELAP which merits some discussion, if for no other reason than it is philosophically similar to the basis for FLAG. Interactive CORELAP is considered a construction program. Regular CORELAP is an improvement algorithm. Interactive CORELAP allows the user to revise the obtained layout shape or solution. This approaches the concept of a computer assisted facility layout program. Interactive CORELAP also allows the user to score a layout at any stage of the program. Department locations may also be pre-assigned to specific locations. The batch version does not have these features.

F. ALDEP: Automated Layout DEsign Program

ALDEP's input is basically identical to that of CORELAP. The ALDEP program, however, is very unique in other aspects. ALDEP generates several hundred layouts by a random placement algorithm. The program scores each layout, allows the designer to compare them, and select the best one. Another distinction of ALDEP is its ability to layout a multi-story plant, up to three floors. This final feature, however, can present difficulties. For example, Muther and McPherson [16] have stated that some activities could be split between floors by chance.

The ALDEP input consists of a departmental listing, related department sizes, a REL chart using Muther's six-step priority closeness rating scale, the number of layouts to be generated, block size, sweep length used in the placing of departments, minimum score required to print out a layout, and two random number seeds. Like several other programs previously discussed, ALDEP allows the user to preassign departments. The A, E, I, O, U, and X ratings differ from those used in CORELAP. The scores 64, 16, 4, 1, 0 and -1024 are assigned respectively, to each of the codes. Like CORELAP, these numbers are used in scoring the layout. The layout discipline is discussed by Thompkins and Moore [22] and is described below:

1. Assign all departments which have been preassigned to the layout blocks.

2. Randomly select first department to be placed in the layout. (This department may be specified, but this is not recommended.)

3. Scan all unselected departments in the highest remaining priority closeness category. If there is more than one department in this category, one of them is randomly selected.

4. Place the selected department, block by block, according to a path determined by the sweep length. (Refer to Figure 2.2.)

5. Repeat steps 3 and 4 until all departments are placed.

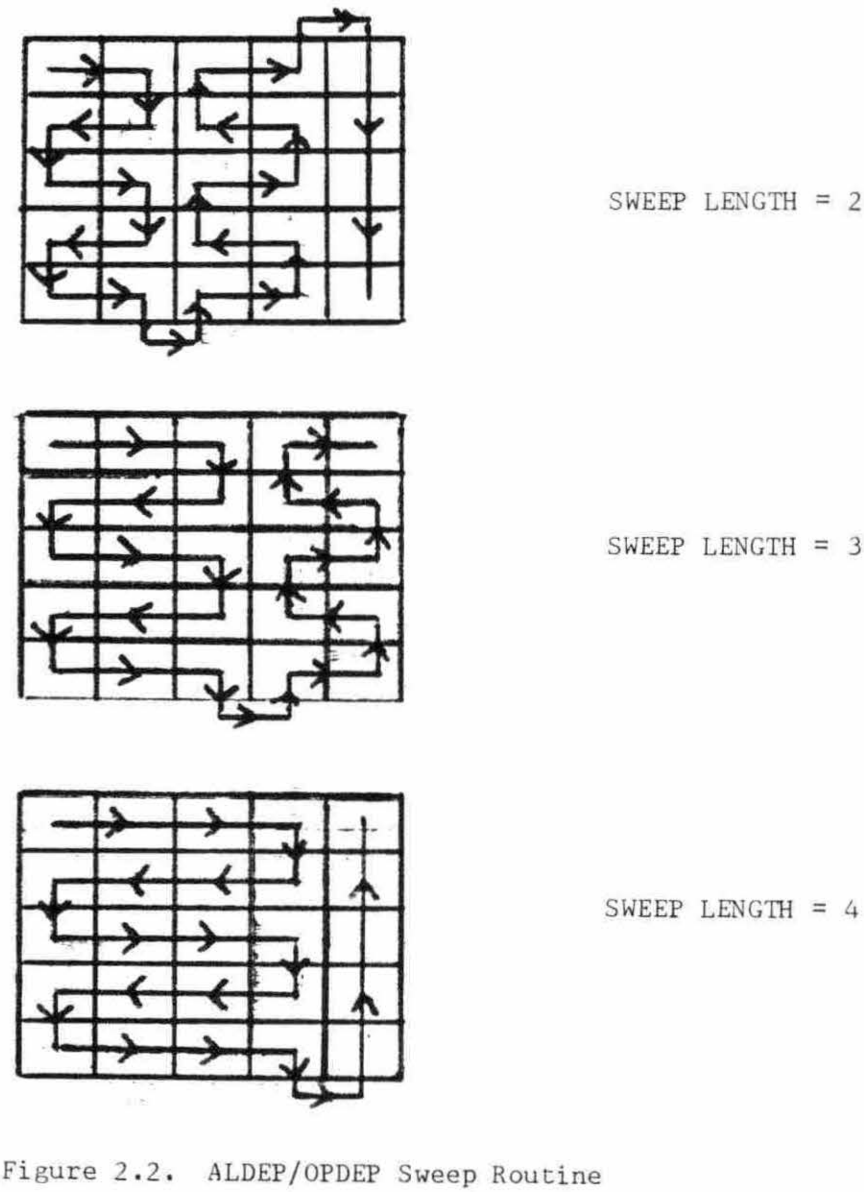


Figure 2.2. ALDEP/OPDEP Sweep Routine

6. The layout is next scored. This is done by reviewing all cells in the REL chart. If two departments are contiguous, the relationship numerical value (A=64, E=16, etc.) is added to the score. If the departments are within one block of each other, then half of the relationship's numerical value is added to the score.

7. If the score is equal to or greater than the score specified by the user, the layout is printed.

8. If the specified number of layouts have been generated, the algorithm ends. If not, the execution returns to step 1.

The layouts generated by ALDEP are highly dependent on the sweep length specified. This is because the sweep length determines the path by which department blocks are placed in the layout grid (see Figure 2.2). Different sweep lengths will generally result in different layout possibilities. Therefore, to assure optimal results, the program should be run at various sweep lengths.

Thompkins and Moore [22] suggest that one reason ALDEP is very useful is that aisles, stairwells, docks, and other layout details not directly associated with workcenters can be placed. However, this claim is an exaggeration as can be seen by examining the sample ALDEP output in Appendix I.

Nelson [17] developed an enhanced version of ALDEP called OPDEP (Optimal Plant DEsign and Evaluation Program). OPDEP makes some simple, but logical, changes in the manner in which the layout discipline scans REL chart data to select departments for placement. Nelsen suggests relating the A, E, I, O, U, and X values to material flow levels of individual facilities. This reduces the arbitrary nature of the choices made by ALDER.

G. Other Facility Layout Programs

There exist a variety of programs which have received less attention in published literature. O'Brien and Barr [18] have used computer graphics to develop an improvement program. The program interchanges departments to see if weighted material handling costs can be reduced.

There are also some theoretical layout disciplines that have been presented in the literature not associated with any particular named program. For example, Bazaraa [3] has developed a layout scheme based on layout techniques have also been discussed by Papineau, Francis and Bartholdi [19] and Thorton, Francis, and Lowe [23].

There is also work being done on using the computer to layout particular sections of an industrial facility. For example, Moore [14] reported the use of a computer in the planning of storage facilities. There has also been work documented on the use of computers in the layout of office facilities as described by Vollmann, Nuggent and Zartler [25].

Table 2.1 describes some of the features of the more unique facility layout programs. This illustration lists whether the program is a construction, or improvement algorithm together with REL chart input requirements. The table also lists appropriate references for each program.

FLAG incorporates and extends many features of the programs that have been described in this chapter. Chapter III describes the use and operation of FLAG in detail.

Table 2.1. Features of computerized facilities design programs

Acronym	Name	C/I[a]	REL Chart Data Required	Uses Muther's Priority Closeness Rating?	S/M[b]	Comment	Reference Number
CRAFT	Computer-ized Relative Allocation of Facilities	I	Yes	No	S		2,4,5,7,8, 9,10,11,12, 15,16,20,22, 24,26
COFAD	COmputer-ized FAcilities Design	I	Yes	No	S		15
PLANET	Plant Layout ANalysis and Evaluation Techniques	C	Yes	No	S		2,7,15,22
CORELAP	COmputer-RElation-ship LAyout Planning	C	Yes	Yes	S		2,5,7,13, 14,16,21, 22,26

ALDEP	Automated Layout DEsign Program	C	Yes	Yes	M	Three floors	1,3,8,14, 16,21,22, 26
OPDEP	Optimal Plant Design and Evaluation Program	C	Yes	Yes	M	Altered Version of ALDEP Three floors	17
LSP	Layout Simulation Program	C	Yes	Yes	M	Rectangular floor output. Limited inter-action between planner and computer. Distinguishes variable, linked, and fixed depts. Combines conceptual characteristics of CRAFT, CORELAP, and ALDEP	7,15

[a]Construction (C)/Improvement (I) Algorithm.

[b]Single Story (S)/Multi-story (M).

Table 2.1. Continued

Acron	Name	C/I	REL Chart Data Required	Uses Muther's Priority Closeness Rating?	S/M	Comment	Reference Number
RUGR		C	Yes			Czechoslovakian program using characteristics of planarity from graph theory as basis for heuristic algorithm. Requires as input the fixing one dimension of rectangular building.	7,15
OFFICE		I	Yes			Specialized layout program in office design.	15,25
RMA Comp I		C	Yes				7,16,26
IMAGE		C	Yes				15

COMPROPLAN		I	Yes		Builds a 3-D relationship diagram rotating it in three dimensions until an acceptable two dimensional view is found.	15
PREP	Plant Relayout and Evaluation Package	I-C	Yes	No	Analyzes all possible departmental relocation combinations. Distances are based on the actual footage transversed by material handling systems. It also considers different floor area requirements.	1,15

Table 2.1. Continued

Acronym	Name	C/I	REL Chart Data Required	Uses Muther's Priority Closeness Rating?	S/M	Comment	Reference Number
MAT	Modular Allocation Technique	C	Yes	No	S	Sub-optimal, max of 40 departments. Allocates facilities on matrix grid. Selects an initial lay-out at random and then attempts to improve by making pairwise interchange.	6,12,26
FRAT	Facilities Relative Allocation Techniques	I	Yes	No	S	Very efficient in computer time. Designed to solve problems of equal area facilities.	12

III. METHOD OF OPERATION

A. Introduction

The purpose of this chapter is to describe the principal operational features of FLAG. Facets of the operation of the FLAG programs are illustrated using a detailed numerical example.

B. Distinguishing Operational Features

FLAG is a series of programs which encompass a large part of the facility layout problem - from developing individual workcenter layouts to the printing of the final layout. Some of the distinguishing features of FLAG include:

- An actual plant layout as opposed to a numerical grid.
- Internal layouts of individual workcenters.
- Programs are adapted to taking into account different product mixes of multi-product facilities.
- Flows are measured from workcenter points of entry and departure.
- A greater degree of user participation.

These distinguishing aspects of FLAG are illustrated in greater detail in Chapter VI.

C. Program Description

FLAG is comprised of four programs which utilize four subroutine libraries. The four FLAG programs are outlined below:

FLOWS - Phase one involves the input of all data detailing the flows between workcenters required for REL chart calculations.

WORKOUT - Phase two of the facility layout procedure which develops the layout of individual workcenters (or departments).

LAYOUT - Phase three generates the proposed facility layout interactively with the engineer.

OUTPUT - Phase four generates various forms of output according to user needs and specifications.

The four subroutine libraries utilized by FLAG are:

IGL - Interactive Graphics Library, a set of PLOT 10 graphics subroutines used by the VAX computer.

TEMPS - A series of subroutines that draw templates of machines, workbenches, etc. This library also contains a directory of templates.

UTIL/A and UTIL/B - Libraries of general purpose FORTRAN subroutines that perform various tasks required throughout FLAG. For example, the subroutine CLEAN erases the graphics display.

The interaction and logical flow of these programs and subroutine libraries are illustrated in the general flow diagram shown in Figure 3.1. Detailed descriptions are presented in the next section using a numerical example to describe the FLAG layout procedure.

D. A Numerical Example

Throughout the remainder of this chapter a numerical example is used to describe the FLAG layout process with clarity. The example is that of a fictitious manufacturing company which produces three products referred to as A, B, and C.

The production of products A, B, and C requires six purchased parts

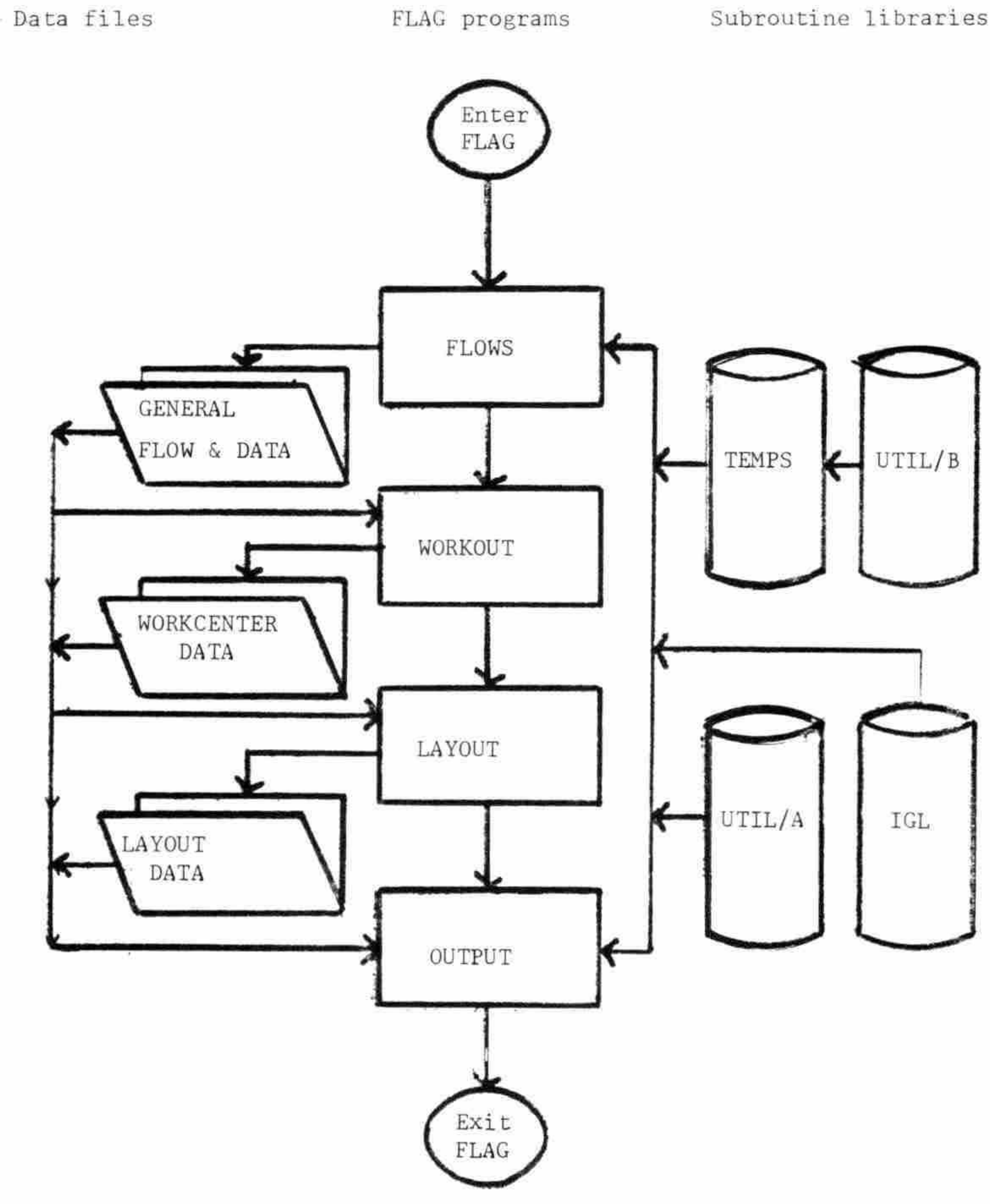


Figure 3.1. FLAG general flow diagram

designated as P1 to P6. It has been determined that five workcenters, or departments, are required to perform all manufacturing processes. These departments are denoted as shipping (S), receiving (R), and workcenters I, II, and III. The purchased part make-up for each product together with process locations for each part are shown in Table 3.1. Partially completed subassemblies of Product A are denoted as A2 and A3. Similar subassemblies for Products B and C are denoted as B2, B3, and C2, respectively. Figure 3.2 illustrates all part flows between departments.

All interdepartmental flows are groups of parts, subassemblies, and final products. The size of these flows is determined by the particular material handling system. This size also establishes the move-cost per foot of each group of parts, subassemblies, and final products. These values are presented in Table 3.2.

1. FLOWS subroutine

The from-to chart can now be developed. All calculations are done by FLOWS after the user inputs the product mix. This from-to chart depicts the number of movements between departments for a predetermined unit of time and product mix. The example from-to chart is based on a period of one year, and the following production mix:

Product A	5000 units
Product B	3000 units
Product C	1500 units

Once this product mix has been input, material flows between departments, are calculated, using data entered by the user. A flow chart outlining FLOWS is presented in Figure 3.3. For example, suppose a purchase part flows from Receiving to Workcenter I. From Table 3.1, it

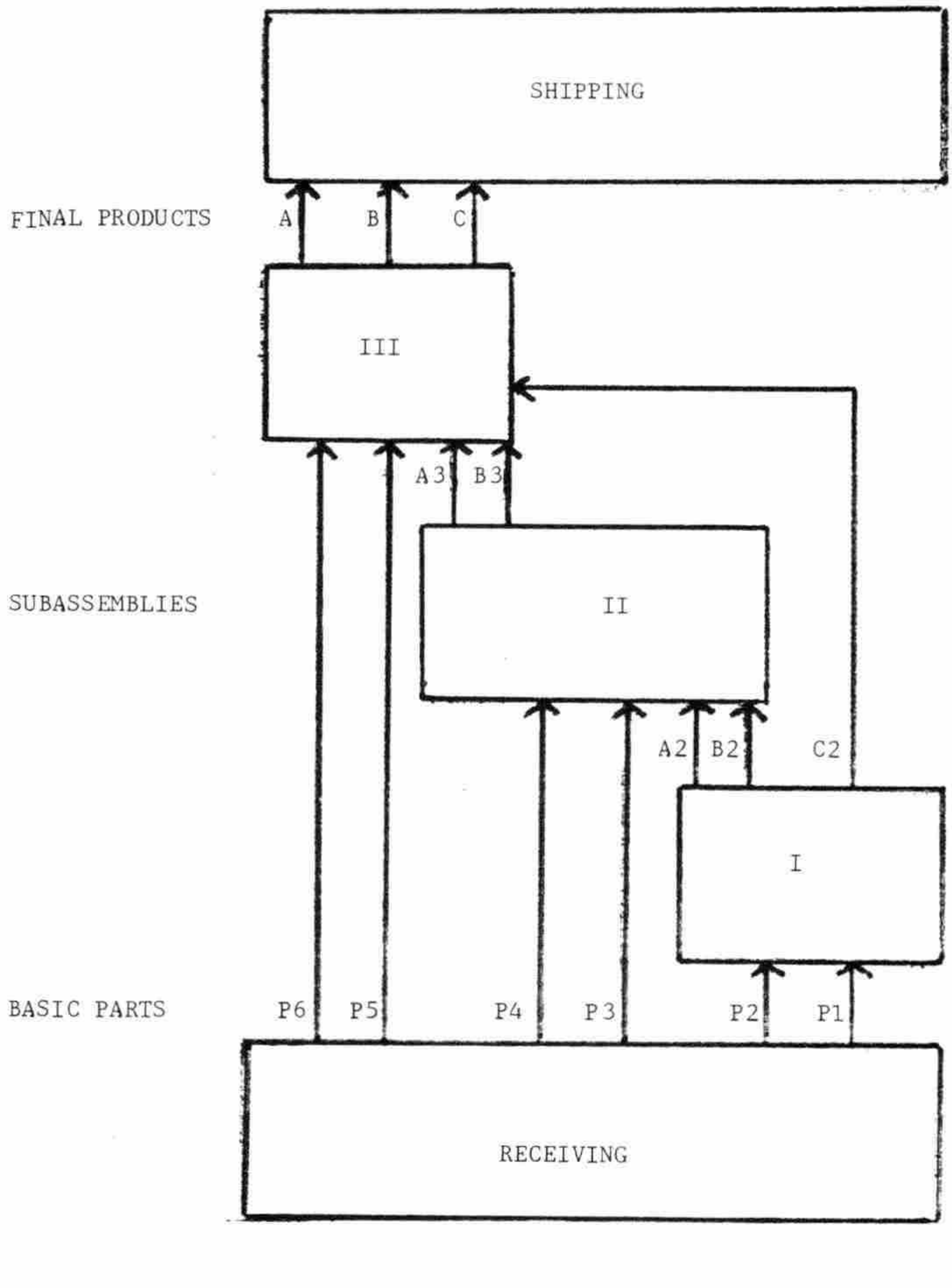


Figure 3.2. Departmental part flow patterns

Table 3.1. Routings and configurations of manufactured parts

PRODUCT	PART	PART QUANTITY	PROCESS LOCATION[a]
A	P1	3	I
	P4	4	II
	P5	1	III
	P6	6	III
B	P1	3	I
	P2	2	I
	P3	4	II
	P5	4	III
C	P1	2	I
	P2	2	I
	P5	6	III
	P6	6	III

[a]Workcenter numbers.

Table 3.2. Parts move data

ITEM	UNITS/MOVE	MOVE-COST/FT.
P1	25	.50
P2	50	.30
P3	30	.50
P4	20	.25
P5	25	.30
P6	50	.40
A2	10	.75
B2	15	.60
C2	25	.70
A3	10	.85
B3	10	.70
A	5	1.25
B	5	1.10
C	2	1.00

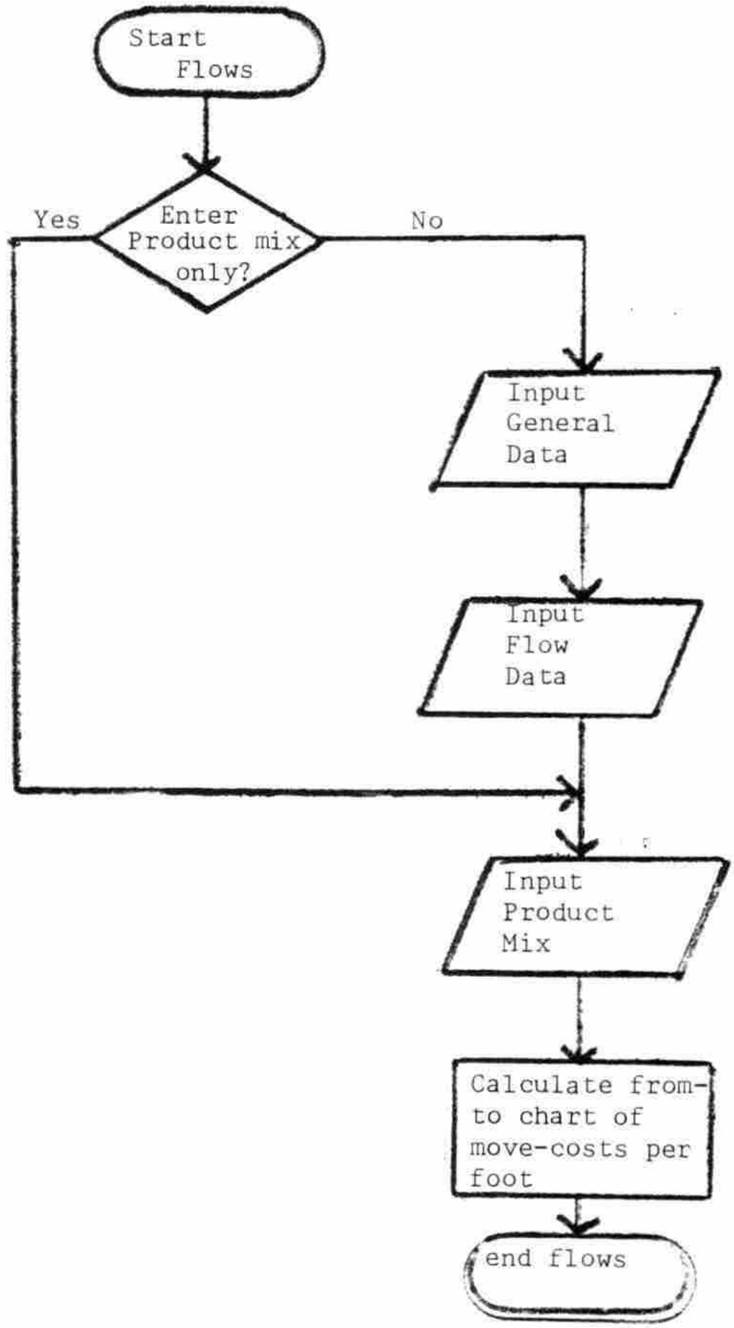


Figure 3.3. General flow chart for FLOWS program

can be seen that product A requires 3 units of P1, product B requires 3 units, and product C requires 2 units. From Table 3.2, P1 moves in groups of 25 units. Therefore, the general equation for the number of moves of P1, referred to as $P1^*$ is:

$$P1^* = (3A + 3B + 2C)/25 \qquad (3.1)$$

Therefore, with the suggested product mix:

$$P1^* = [(3 \times 5000) + (3 \times 3000) + (2 \times 1500)]/25$$

$$P1^* = 27{,}000/25$$

$$P1^* = 1080$$

Similarly, from Table 3.2, subassembly A2 moves in groups of 10 units. Since there is one subassembly for each unit of product A, the general equation for the number of A2 subassemblies, $A2^*$, is:

$$A2^* = A/10$$

$$A2^* = 5000/10$$

$$A2^* = 500$$

Ten represents the number of units moved per group (See Table 3.2). The fraction A/10 is always rounded up to the nearest integer value.

General equations for interdepartmental movements, based on data from Tables 3.1 and 3.2 are presented below. Specific values, based on the above product mix, are also show, rounded up to the nearest integer value.

$$P1^* = (3A + 3B + 2C)/25, \qquad P1^* = 1080$$

$$P2^* = (2B + 2C)/50, \qquad P2^* = 180$$

$$P3^* = 4B/30, \qquad P3^* = 400$$

$$P4^* = 4A/20, \qquad P4^* = 1000$$

$$P5^* = (A + 4B + 6C)/25, \qquad P5^* = 1040$$

$$P6^* = (6A + 6C)/50, \qquad P6^* = 780$$

Table 3.3. From-to chart

FROM \ TO	R	I	II	III	S
R	---	$P1^* + P2^*$	$P3^* + P4^*$	$P5^* + P6^*$	0
I	0	---	$A2^* + B2^*$	$C2^*$	0
II	0	0	---	$A3^* + B3^*$	0
III	0	0	0	---	$A^* + B^* + C^*$
S	0	0	0	0	---

Table 3.4. Move-cost table

FROM \ TO	R	I	II	III	S
R	---	$.5P1^* + .3P2^*$	$.5P3^* + .25P4^*$	$.3P5^* + .4P6^*$	0
I	0	---	$.75A2^* + .6B2^*$	$.7C2^*$	0
II	0	0	---	$.85A3^* + .7B3^*$	0
III	0	0	0	---	$1.25A^* + 1.1B^* + C^*$
S	0	0	0	0	---

Table 3.5. Move-cost table for annual production levels of 5000, 3000, and 1500 for products A, B, C, respectively

FROM \ TO	R	I	II	III	S
R	---	594	450	624	0
I	0	---	495	42	0
II	0	0	---	335	0
III	0	0	0	---	2660
S	0	0	0	0	---

$A2^* = A/10,$	$A2^* = 500$
$B2^* = B/15,$	$B2^* = 200$
$C2^* = C/25,$	$C2^* = 60$
$A3^* = A/10,$	$A3^* = 500$
$B3^* = B/10,$	$B3^* = 300$
$A^* = A/5,$	$A^* = 1000$
$B^* = B/5,$	$B^* = 600$
$C^* = C/2,$	$C^* = 750$

Using Figure 3.2 and the results above, the resultant general from-to chart can be developed and is presented in Table 3.3. The move-cost per foot data from Table 3.2 are then applied to the from-to chart to obtain the move-cost table presented in Table 3.4. The calculations in Table 3.4 are completed in Table 3.5 using the production levels previously suggested. The values in Table 3.5 represent the material handling costs for each foot that separates each department. For example, it will cost \$594 (\$0.50 x 1080 + \$0.30 x 180) in material handling for each foot that separates Receiving from Workcenter I. (Refer to Table 3.5.) The move-cost table represents the REL chart data on which future placement decisions will be based.

2. <u>WORKOUT subroutine</u>

The user begins the layout process with WORKOUT. A flow chart for this subroutine is presented in Figure 3.4. The program starts by asking the user the number of departments in the facility. For this example, four departments are specified, by arbitrarily combining Shipping and Receiving since it is desired to place both areas adjacent to a single set of docks. This combination is not required by WORKOUT, but has been made to simplify example calculations.

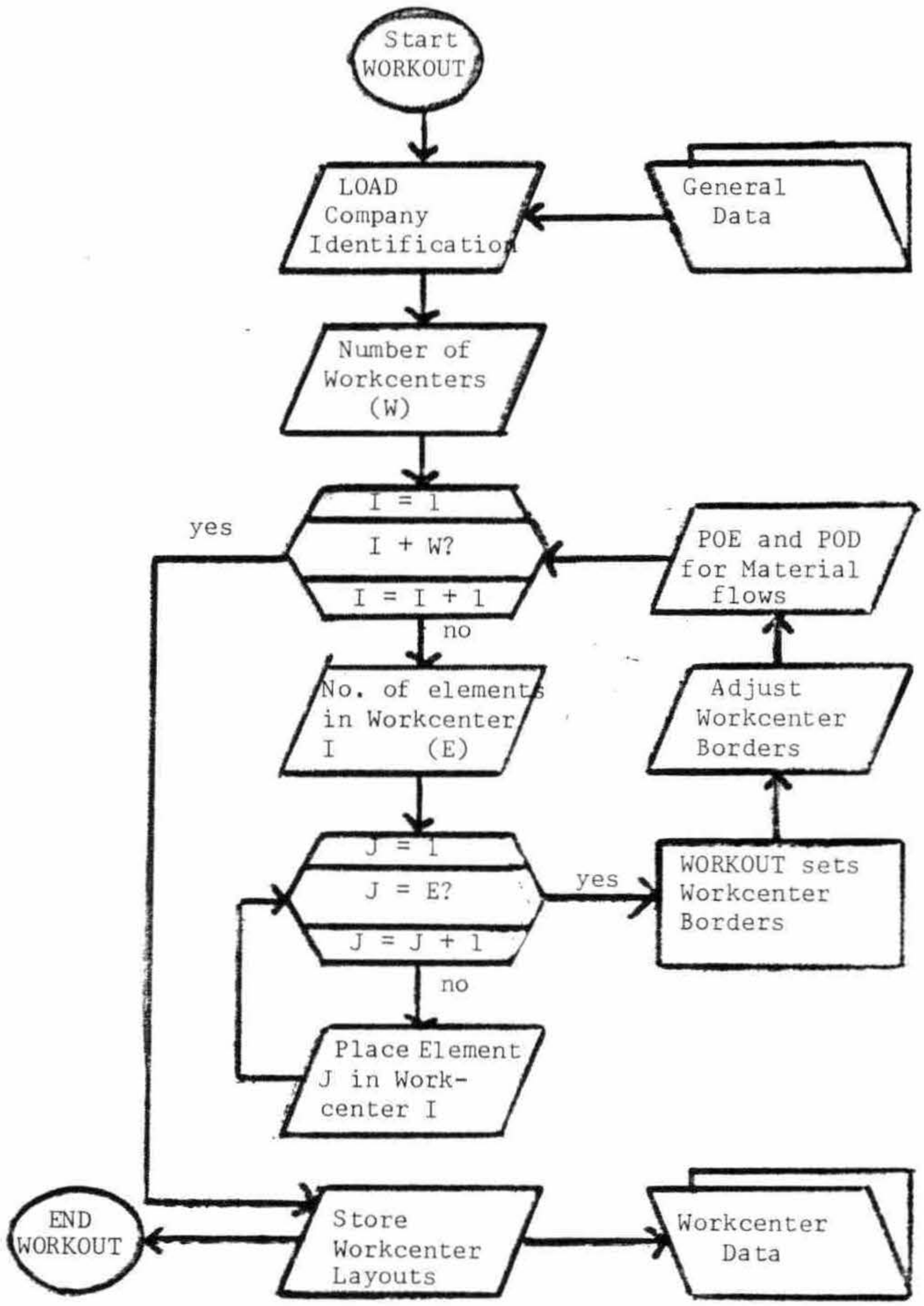


Figure 3.4. WORKOUT general flow diagram

WORKOUT now enters the main program loop which is repeated once for each department. For purposes of this example, it is assumed that WORKOUT is in its third iteration and awaiting input for data on workcenter II. The first question asked of the user is the number of elements in the workcenter. FLAG considers all machines, carts, tool racks, workbenches, pallets, and any other workcenter fixture that should be laid out, as an element. A machine, for example, is considered a primary element. Workcenter II will be assumed to have three elements: a drill, tool rack, and pallet.

After input of the number of elements, the user begins to layout the workcenter. At this point, the workcenter has no exterior border. The user is free to position elements at any point in relation to one another. WORKOUT now enters a secondary loop that is repeated for each element. The initial step is the selection of a template from the template directory. The FLAG template library contains two types of templates. The first are machine templates which are designed by PLANPRINT Inc., a professional plant layout firm. While PLANPRINT designs a template for every machine made, FLAG uses only thirty-three representative templates. For example, a 55 ton Niagara press template is used as generic representative of all 55 ton presses.

The second template type is that of general geometric shapes which the user can utilize and label as required by design needs. For example, a pallet can easily be denoted by use of a square or rectangle.

After selection of the appropriate template, the shape is placed in the workcenter layout by detailing an x, y coordinate selection and rotation factor. For example, the engineer, already having selected the machine template for a drill, places it in the layout by specifying an x coordinate of 50, y coordinate of 53 and a rotation of zero degrees. (See Figure 3.5.) FLAG also generates temporary dashed lines across the CRT screen showing coordinates to assist the user in aligning other templates in the layout. After the initial phase of placing the template, the user is permitted to make necessary adjustments. This procedure is repeated until all elements in the workcenter are placed.

Upon placement of all elements, WORKOUT draws an initial border around the workcenter. This is done by setting the border two feet from each extreme element. (See Figure 3.5.) For example, the extreme right element is the pallet. Therefore, an initial left boundary is established two feet to the right of the pallet. These initial boundaries can then be adjusted by the user. The user may also cut out a notch in any corner of the workcenter. In this example, the engineer may notice an excess of unused space in the upper right hand corner of the workcenter. The user therefore may make the adjustment shown in Figure 3.6.

Before leaving workcenter II, the user must answer questions regarding flows in and out of the department. The user specifies the point at which products enter and depart from the workcenter along the workcenter border. These points are respectively referred to as the point of entry (POE) and point of departure (POD). They are shown in Figure 3.6. This completes the WORKOUT phase of FLAG.

3. LAYOUT subroutine

At this point, all workcenters have been designed and stored in a data file for use by LAYOUT which will position these workcenters in the

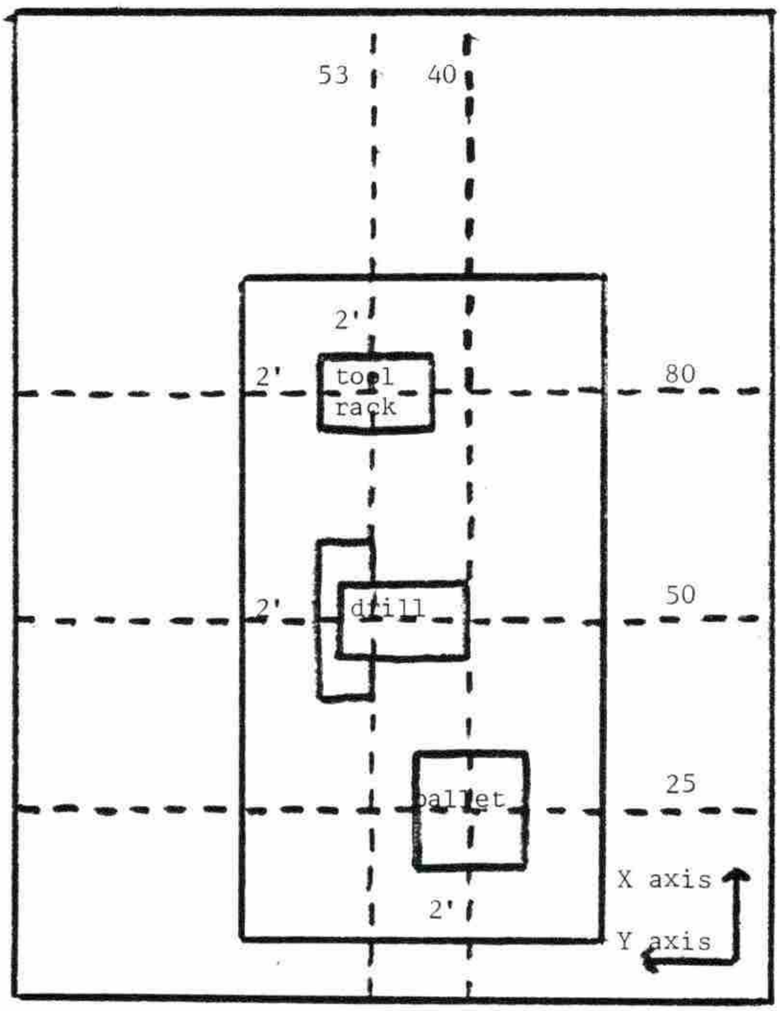


Figure 3.5. Initial layout for Workcenter II

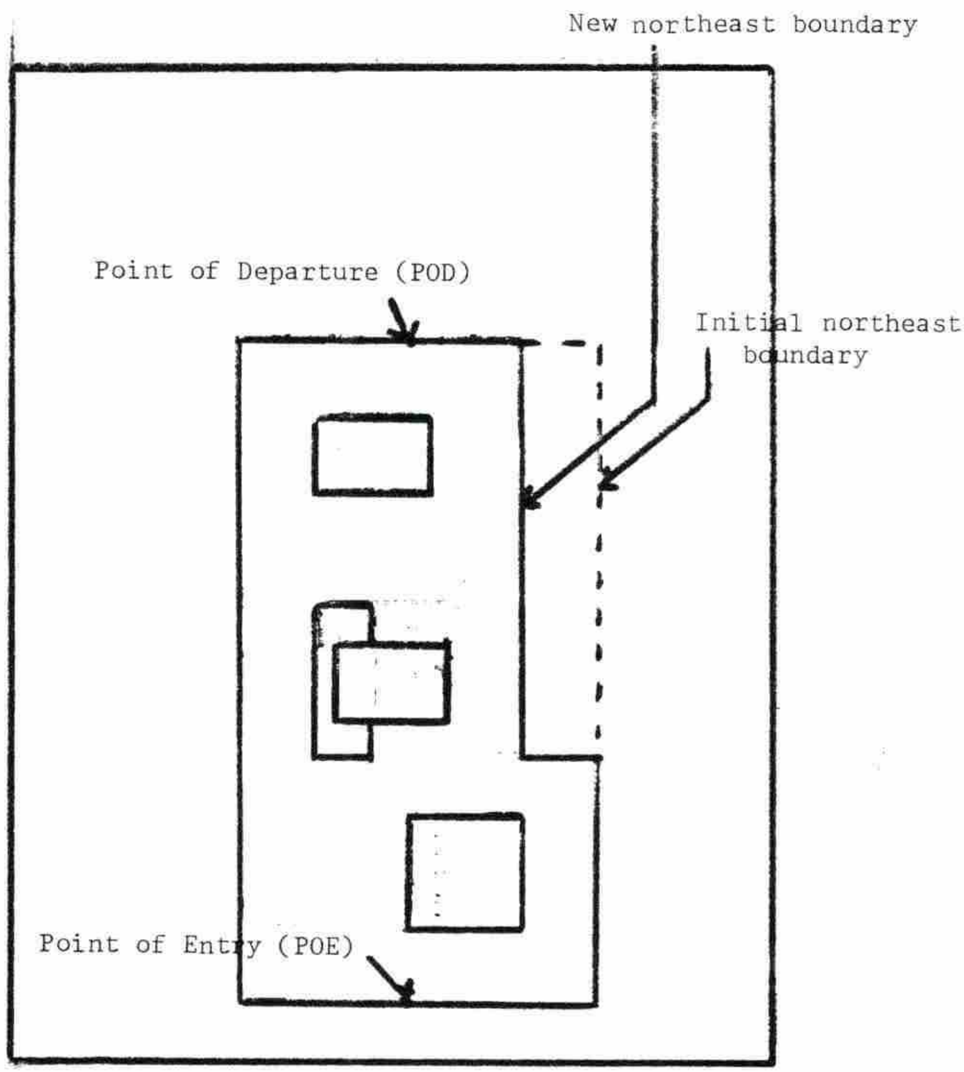


Figure 3.6. Final layout for Workcenter II after notching workcenter perimeter

plant. Also, all material flow relationships have been stored in a data file by FLOWS. The user is now ready to generate the facility layout.

The actual facility layout is now generated in a joint effort between the user and the third FLAG program, LAYOUT. A flow chart for this subroutine is illustrated in Figure 3.7. To illustrate this process, the manufacturing example is continued. The five workcenters to be placed by LAYOUT are shown in Figure 3.8.

The initial phase of LAYOUT is the input of some basic data, such as the default aisle width. This default value can be overridden, but will be used when no other value is specified. The default value, 5 feet, has been used for this example.

The next step is the placement of the Shipping and Receiving workcenters. This department is placed first to assure that it will be along the wall of the plant and will thus have access to shipping docks. LAYOUT presents the user with a hypothetical plant wall. The user then places the Shipping/Receiving departments against this wall with the proper orientation. (See Figure 3.9.) Coordinate values and rotation factors are again specified as in WORKOUT.

It is now desired to determine which workcenter is to be placed next. LAYOUT reviews the move-cost table (Table 3.5) for all relationships of the Shipping and/or Receiving departments with all departments yet to be placed and prioritizes them as shown below:

The prioritized values represent the four move-cost relationships between Shipping/Receiving and other workcenters shown in Table 3.5. The user is presented with the above data. Unless he or she chooses to over-

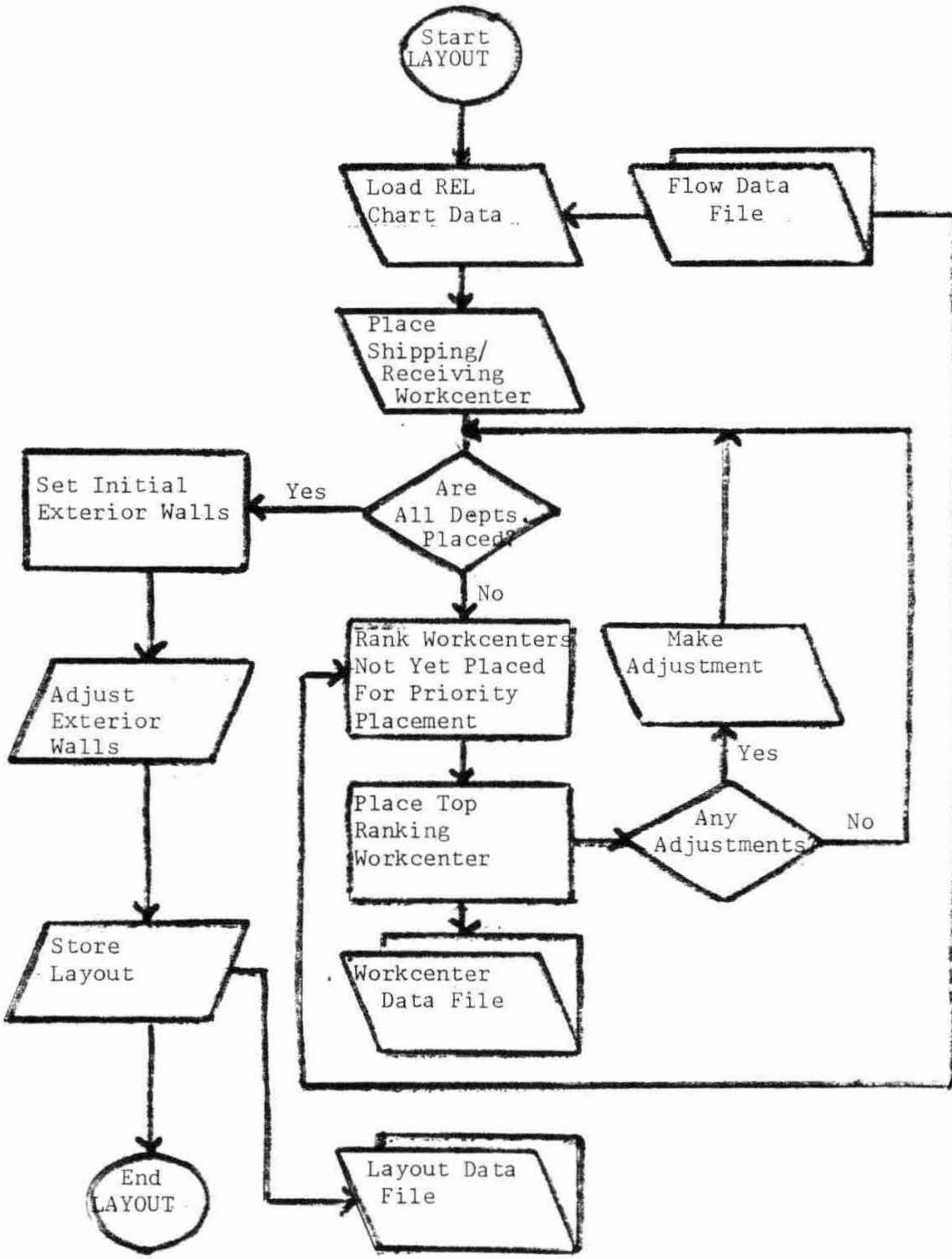


Figure 3.7. LAYOUT general flow diagram

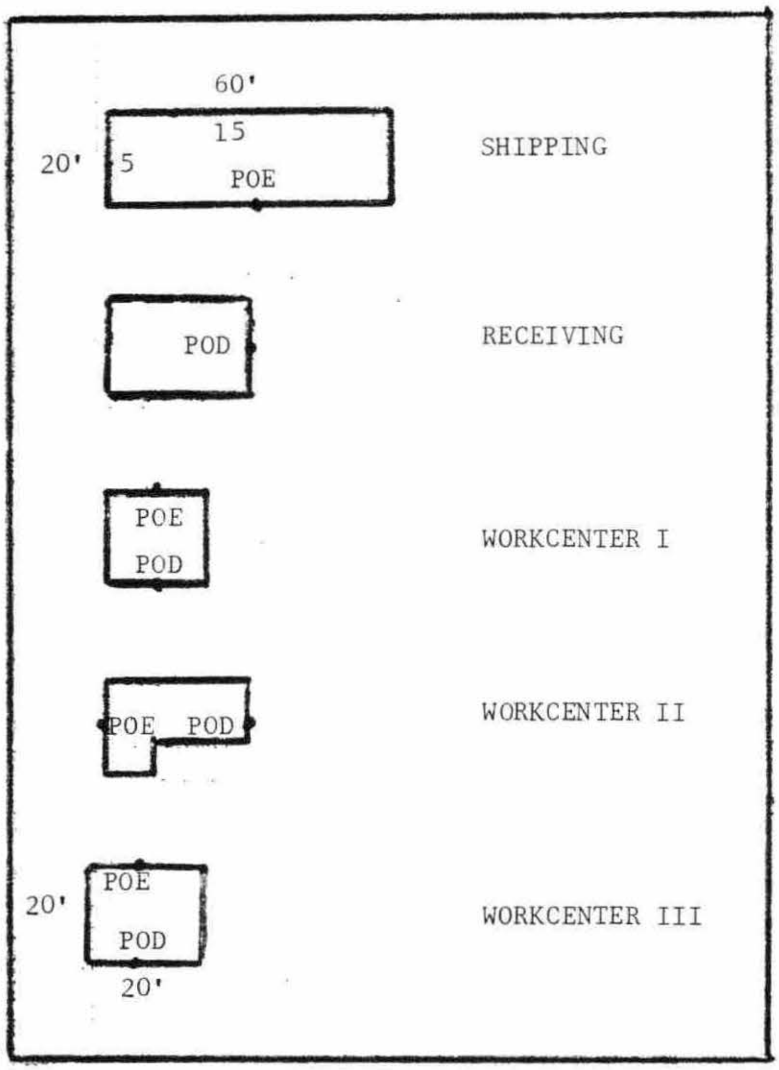


Figure 3.8. Departments prior to placement

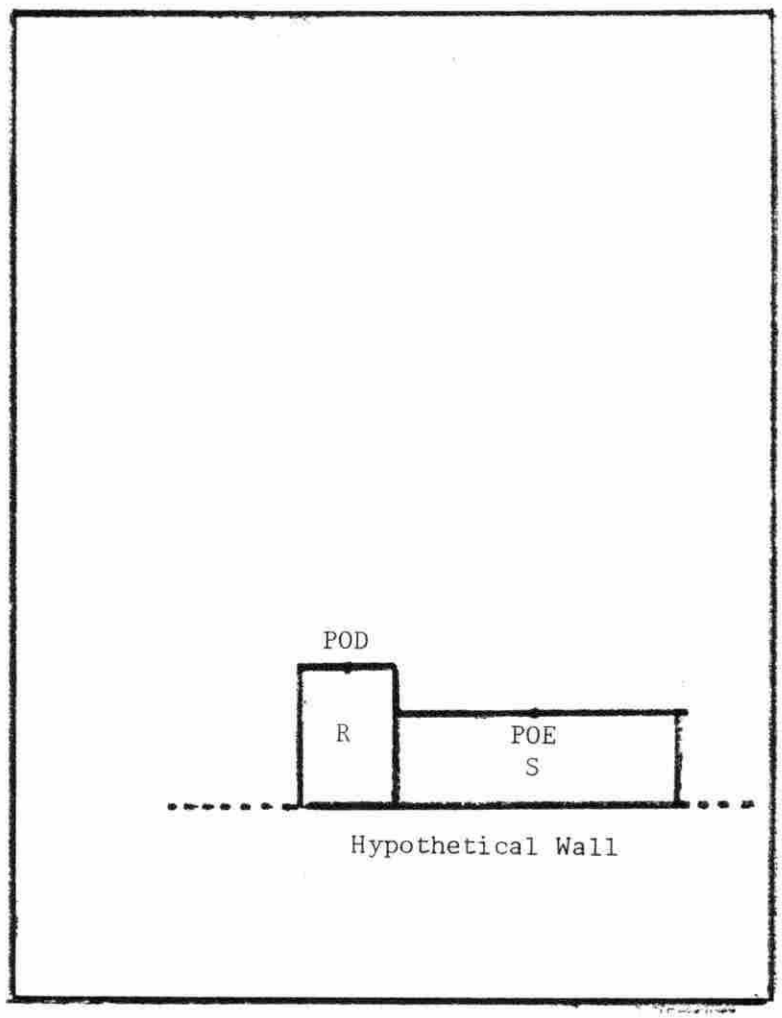


Figure 3.9. Placement of Shipping/Receiving combined department

Table 3.6. First iteration move-cost priorities

Departing	Entering	Move-cost 1 foot
III	S	2660
R	III	624
R	I	594
R	II	450

ride the LAYOUT choice, LAYOUT will satisfy the first priority, by placing Workcenter III's POD across from Shipping's POE. This placement is shown in Figure 3.10. The space separating the two departments is the default aisle width established earlier.

The user is then given the opportunity to make adjustments in the LAYOUT placement if needed or desired. The user may also experiment at this point. For example, after reviewing the above table, the user may think that the second highest move-cost relationship could be reduced by rotating Workcenter III and removing the aisle space. The user knows from the departmental part flow chart (Figure 3.5) that no other workcenter needs direct access to Shipping. This adjustment was made and is shown in Figure 3.11. There is a trade-off in making this change since the POD and POE of Workcenter III and Shipping are now farther apart. This tradeoff is calculated by LAYOUT and shown to the user, as indicated in Table 3.7.

To illustrate the method for Table 3.7, consider a sample calculation for the flow from Workcenter III to shipping. The pre-adjustment distance from POD to POE is 5 feet. The move-cost per foot from Table 3.5 is $2660. Therefore, the total move-cost is 5 x $2660 = $13,300. After the adjust-

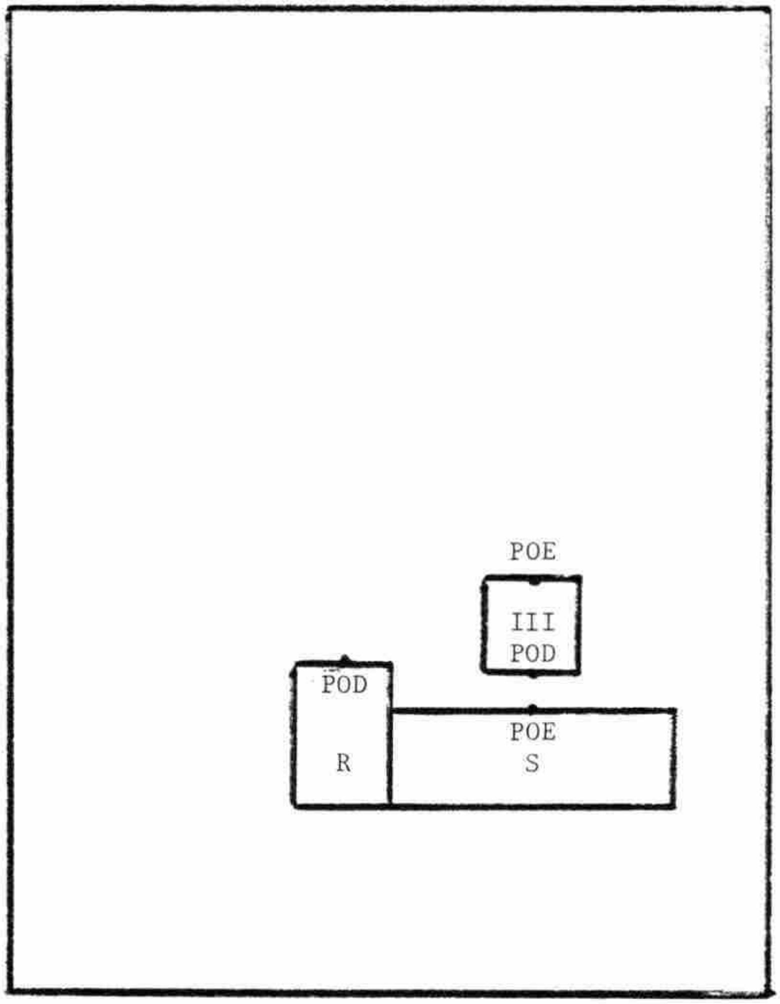


Figure 3.10. First layout iteration - initial placement of Workcenter III

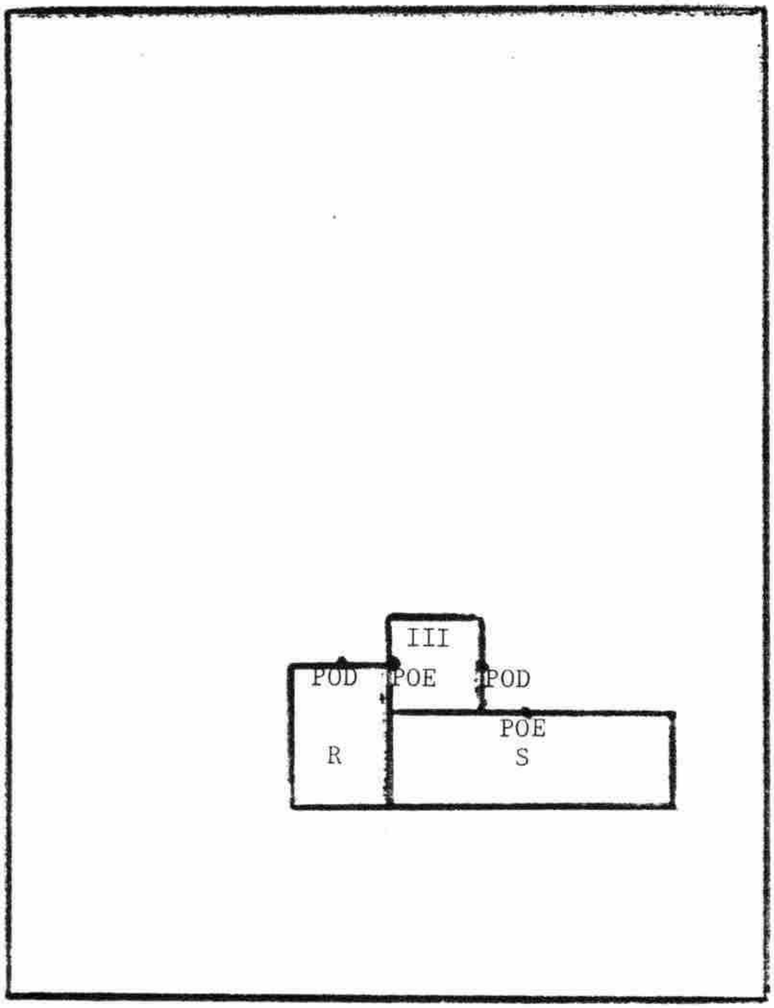


Figure 3.11. First layout iteration - adjustment of Workcenter III

Table 3.7. First iteration adjustment

Dep.	Ent.	Distance (ft)	Move-cost/ft	Total move-cost	Change
Pre-adjustment					
III	S	5	2660	13,300	-
R	III	33.54	624	20,930	-
Adjustment					
III	S	10	2660	26,600	13,300
R	III	10	624	6,240	14,690

Total move-cost change - 1390

ment, the distance doubles to 10 feet, and so does the total move-cost as calculated 10 x \$2660 = \$26,600. Similarly, the move-cost between Workcenter III and receiving is reduced from \$20,930 to \$14,690. The change from the pre-adjustment is given by:

Adjustment Total Move-cost - Pre-adjustment Total Move-cost = Change

26,600 - 13,300 = 13,300

LAYOUT now repeats the placement iteration by again prioritizing the move-cost relationships between those departments already placed with those yet to be placed. These data are presented to the user as shown in Table 3.8. The sum of all change costs represents the total move-cost change, i.e., 13,300 + (-14,690) = -1,390. Therefore, the proposed adjustment results in a move-cost savings of \$1,390 and is therefore finalized by the user.

LAYOUT chooses to place Workcenter I next as shown in Table 3.8. This placement is shown in Figure 3.12. Suppose the user decides to make one adjustment. Since parts will be required to flow to Workcenter II (not yet placed) the user might decide to create an aisle between work-

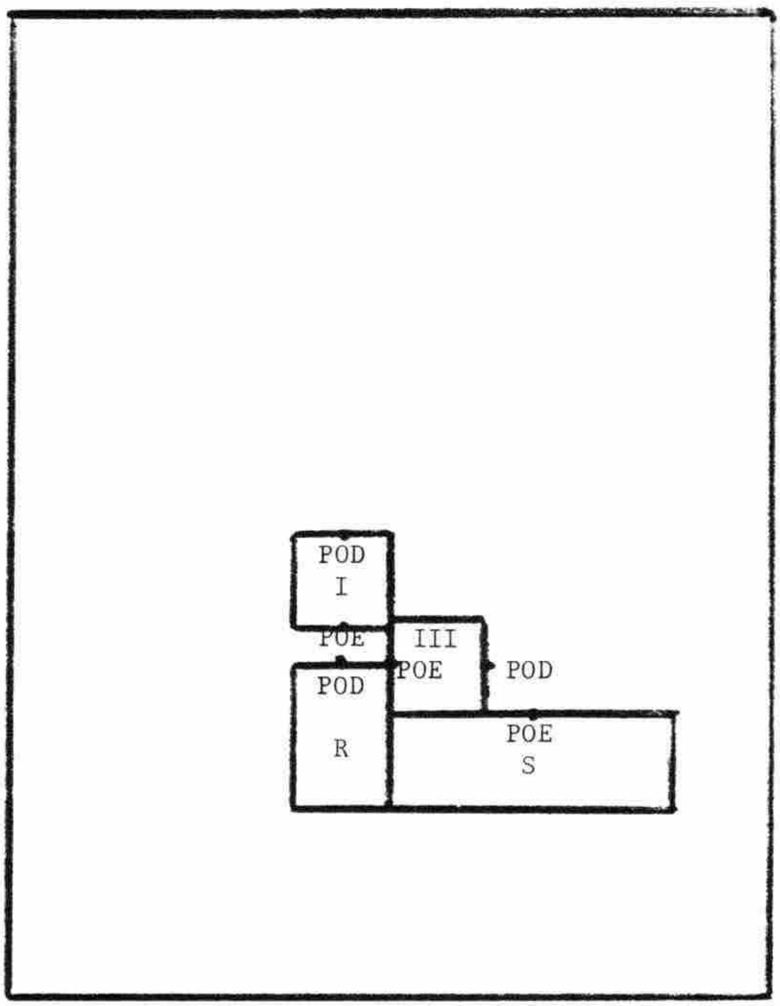


Figure 3.12. Second layout iteration - initial placement of Workcenter I

Table 3.8. Second iteration move-cost priorities

Departing	Entering	Move-cost/ft.
R	I	594
I	II	495
R	II	450
II	III	335
I	III	42

centers I and III. This adjustment is seen in Figure 3.13. After the adjustment is finalized, LAYOUT begins the next placement iteration.

Table 3.9. Final iteration priorities

Departing	Entering	Move-cost/ft.
I	II	495
R	II	450
II	III	335

LAYOUT decides to satisfy the I-II relationship shown in Table 3.9 and does so in Figure 3.14. It is obvious to the user that is might be advantageous to rotate workcenter II in an attempt to better satisfy the II-III relationship. This would bring workcenter II's POD closer to workcenter III's POE without sacrificing closeness between I and II. It would also reduce the required plant area. It is desirable to know whether the savings between II and III are greater than the increase in move-cost relating to the increase in distance between I and II. This experiment is illustrated in Figure 3.15. As before, there is a trade-off is again presented for the user's review as shown in Table 3.10. Since the adjustment results in a savings of $3576, it is finalized by

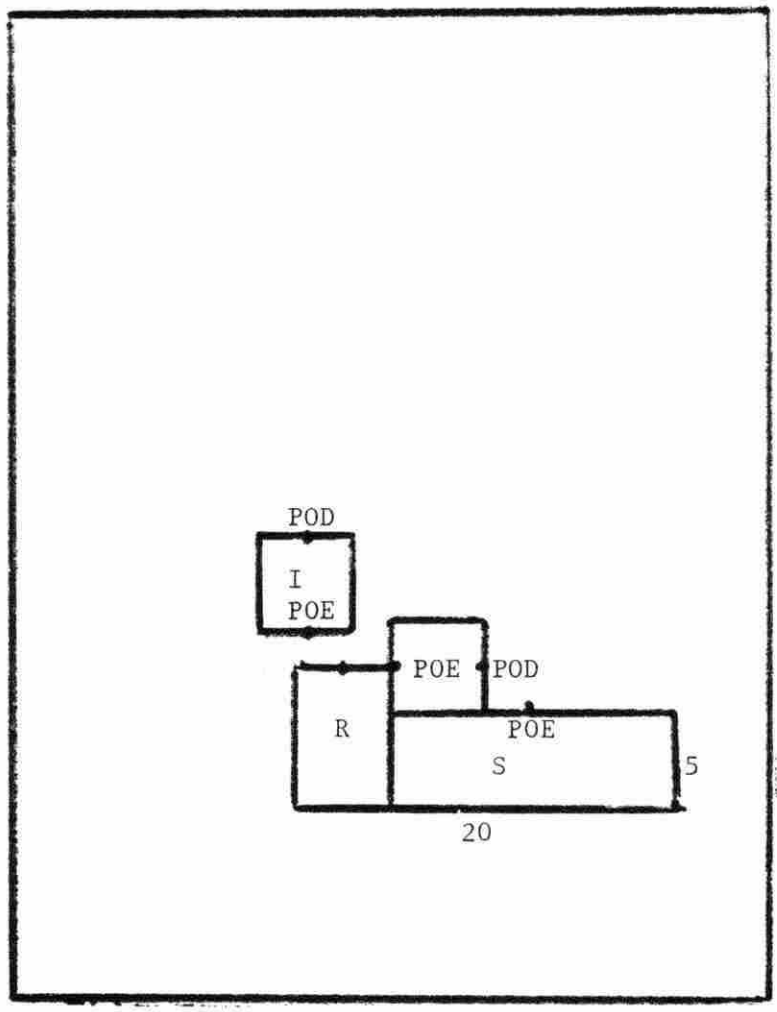


Figure 3.13. Second layout iteration - adjustment of Workcenter I

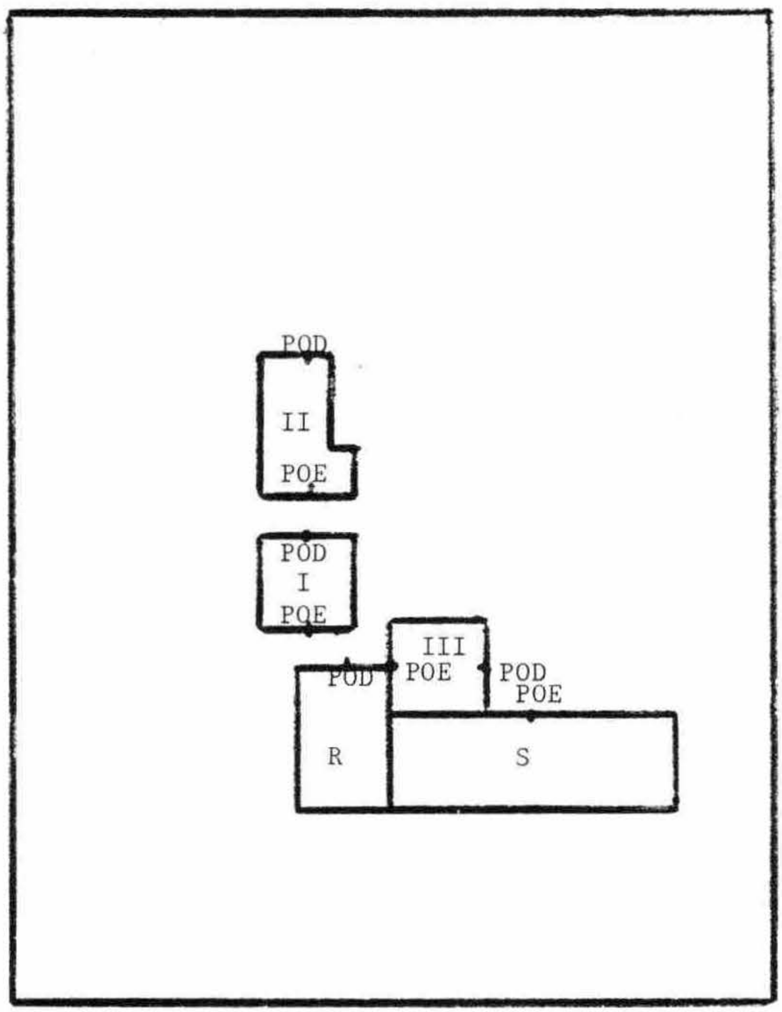


Figure 3.14. Final layout iteration - initial placement of Workcenter II

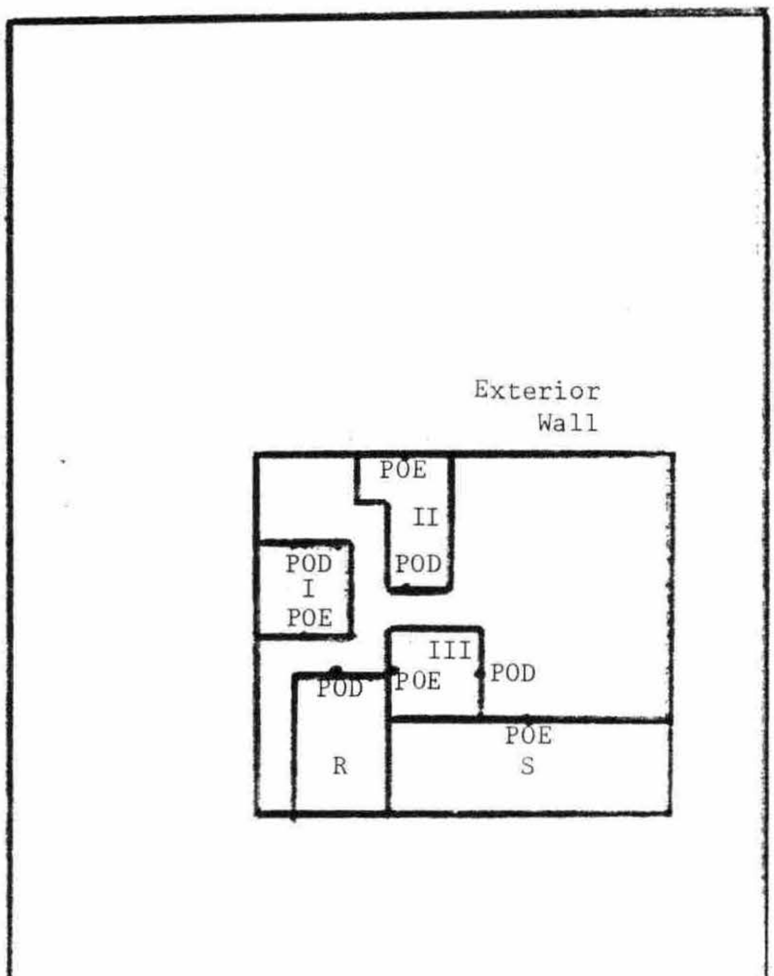


Figure 3.15. Final layout iteration - adjustment of Work-center II/Initial placement of exterior walls

the user, thus completing the placement phase of LAYOUT.

LAYOUT's final task is to place the interior walls. This process is similar to the placement of workcenter border in WORKOUT. The initial borders are automatically determined by the extreme workcenters, as illustrated in Figure 3.15. The user can make horizontal or vertical adjustments. The user can also notch any corner desired as shown in Figure 3.16. After completion of all adjustments, data for the entire layout are stored for later use, thus completing the LAYOUT phase of FLAG.

4. <u>OUTPUT subroutine</u>

The last phase of FLAG is the program OUTPUT. A flow chart for this subroutine is illustrated in Figure 3.17. This program can generate various forms of output specified by the user according to various needs. The possible output alternatives include:

- Full plant layout. This layout can be generated with all templates shown or just workcenter borders. This can be done after only using LAYOUT.
- Workcenter layouts. These drawings can include internal templates, or just borders. They can be used as templates to create a layout by hand. This can be accomplished after only completing WORKOUT.
- Templates. OUTPUT can also be used to make paper copies of any templates in the template library. This can be done at any point in the facility layout process.
- Move-cost rankings. OUTPUT can create a complete listing

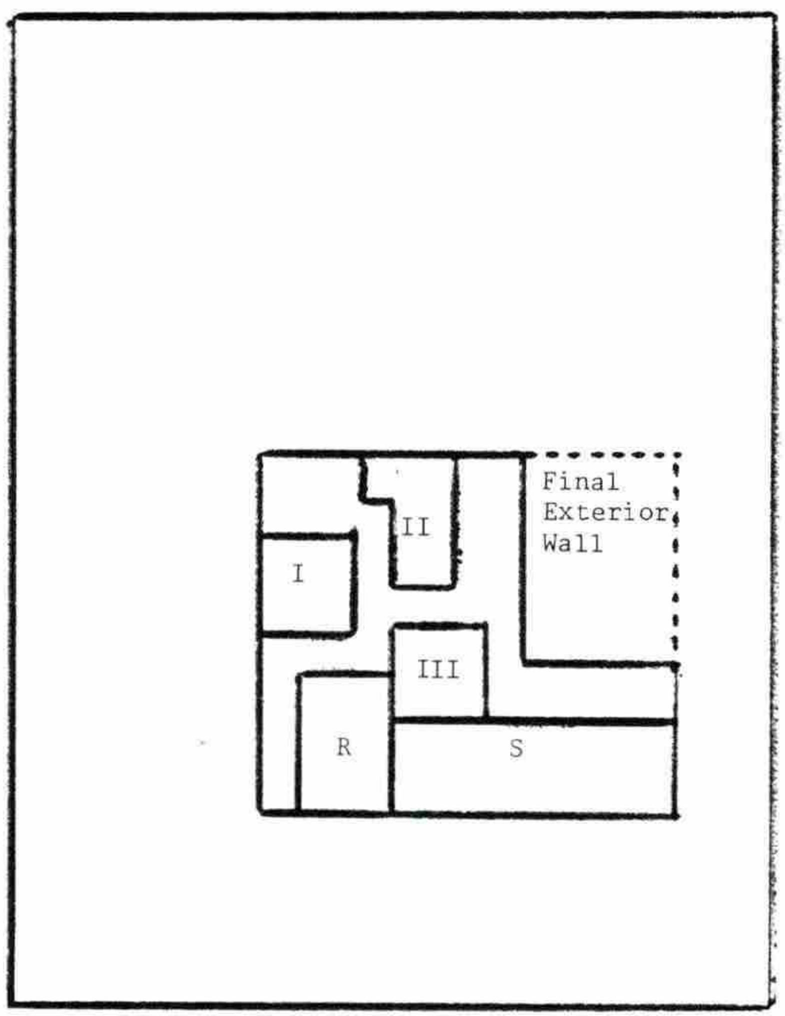


Figure 3.16. Final layout

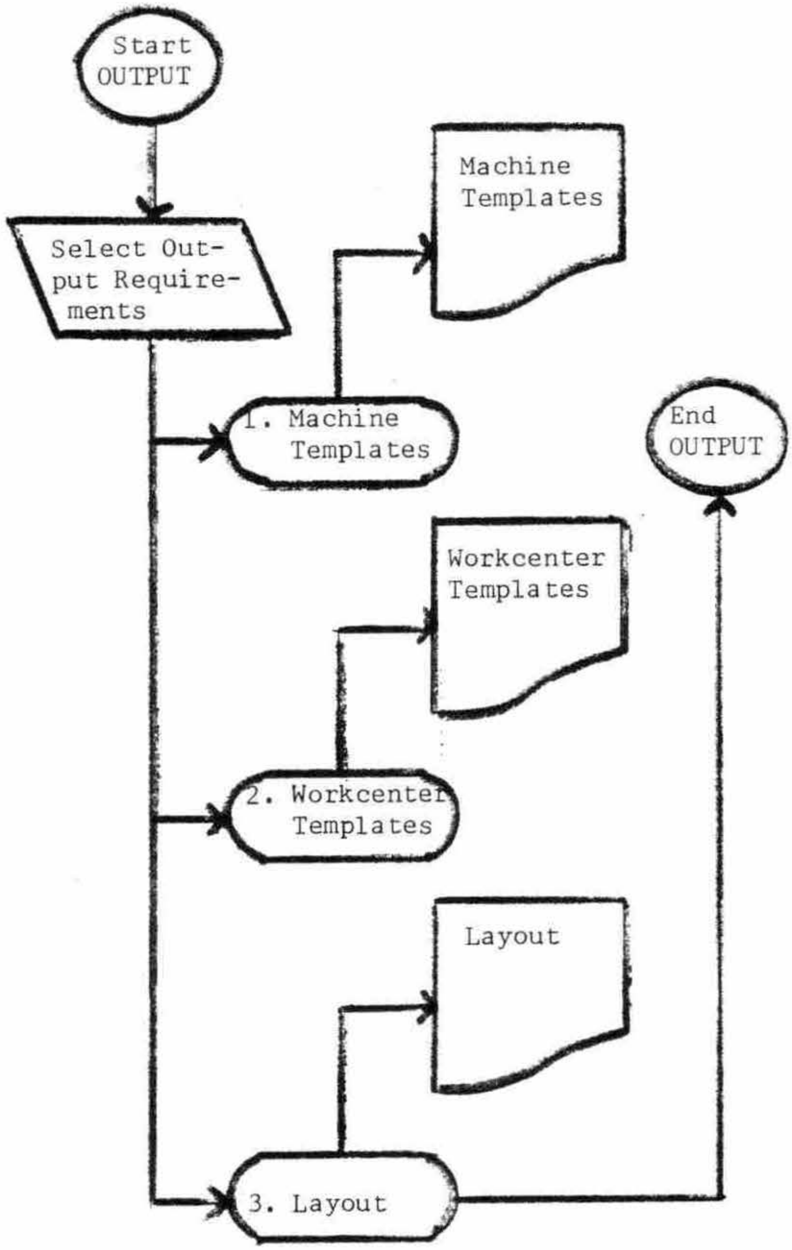


Figure 3.17. OUTPUT general flow diagrams

of all relationships in the move-cost table, ranked from high to low, for any product mix. This feature can only be utilized after running WORKOUT.

This completes the detailed example illustrating the features of FLAG. The following chapter discusses the problem of scale integrity. Using computer graphics presented some special problems of scaling that require detailed discussion.

IV. SCALE

A. Introduction

Scale integrity was one of the more difficult hurdles to overcome in the development of FLAG. Problems of scale were encountered in relation to both hardware and software. Hardware problems of scale occurred because the paper that plotters utilize have fixed dimensions. Finished layouts must be scaled to fall within these dimensions. The program must therefore adapt the layout to the paper. FLAG does this key reducing the scale in increments of 50% until the drawing fits the paper (or display screen).

The FLAG software also must be rather sophisticated to provide standard scales. As will be shown in this chapter, the scale can continually change in the course of operating FLAG. This feature overcomes the difficulty of limited CRT screen space which would otherwise cause problems during the execution of the program.

B. Graphic Display Units (GDUs)

The Graphic Display Unit (GDU) is the measure used to divide the horizontal and vertical axes in the Interactive Graphics Library (IGL) on the VAX II computer. Prior to executing any FLAG graphics commands, the display screen on the Tektronix model 4051 terminal is mapped out into 100 GDUs in the vertical (Y) axis and 130 GDUs on the horizontal (X) axis. This mapping can be changed by two commands.

The first command to control the display screen mapping is the Viewport command. The Viewport specifies which portion of the screen is used as a drawing board. When graphics are used in WORKOUT and LAYOUT, the screen is divided into separate areas as shown in Figure 4.1.

The Viewport command is used to specify a screen section (section A) which is reserved for graphics. Section B is reserved for graphic numerical parameters. This space is used to display information to the user regarding the drawing frame Section C is reserved for interactive questions and answers between FLAG and the user.

The Viewport is always proportional to the dimensions of the full screen. This assures that when OUTPUT expands the picture to the full screen the drawing is not distorted. In other words, OUTPUT proportionally expands pictures to occupy the entire area of Sections A, B, and C of Figure 4.1.

The second mapping command is the Window command. The Window defines the axis of the Viewport. For example, the basic Window command defines the Viewport axis is:

WINDOW (0.0, 130.0, 0.0, 100.0)

The above command sets the X axis range from 0.0 to 130.0 graphic display units. The Y axis is mapped from 0.0 to 100.0 GDUs. Initially, the viewing window is the entire CRT screen. However, after the viewport command has been executed, these coordinates will be mapped into Section A only. Each GDU becomes smaller, proportionally to the reduction in the viewport. This mapping is illustrated in Figure 4.2.

C. Numbered Scales

The Window command described above previously defines dimensions that correspond to Scale no. 1. This scale number appears in Figure 4.2. Scale 1 is defined as the base scale. When expanded to the full screen,

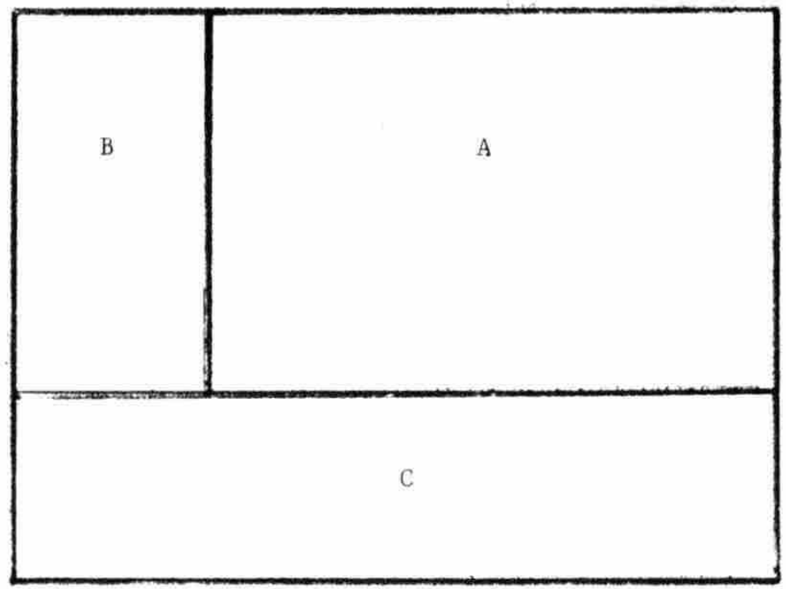


Figure 4.1. The breakdown of the display screen: (A) Drawing frame, (B) Graphics Numerical Parameters, (C) User Interaction

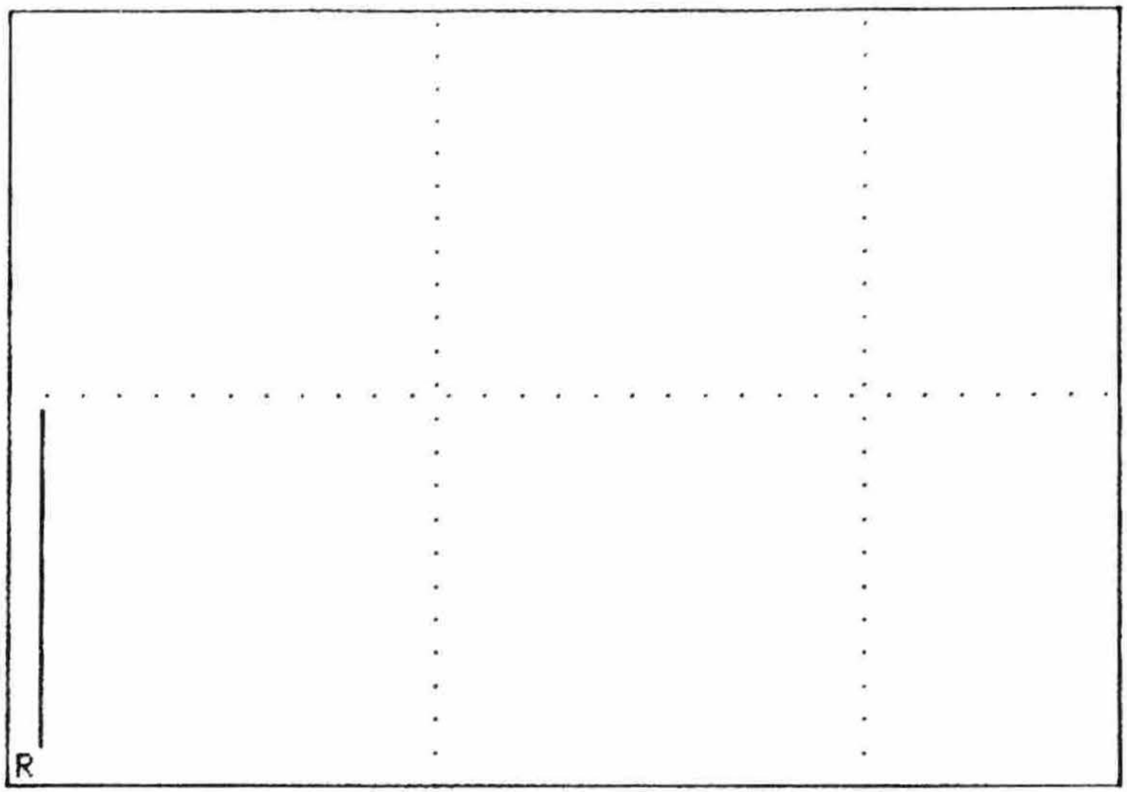
AXIS RANGES:

X AXIS:
0.0 TO 130.0
Y AXIS:
0.0 TO 100.0

DOTTED LINE
INCREMENTS 50.0

SCALE NO.: 1

R LINE = 10 FT.



SELECT TEMPLATE TYPE AND PRESS <RET>.

1. GENERAL SHAPE TEMPLATE.
2. PLANPRINT MACHINE TEMPLATE.

Figure 4.2. Section A viewing window with coordinate grid lines

it corresponds to ¼ inch equals 1 foot.

As additional elements are added to the workcenter, the user will often run out of drawing space (see Figure 4.3.) This difficulty is alleviated by specifying that FLAG jump to the next scale. The window is now doubled and maps the X axis from 0 to 260 CDUs, and the Y axis from 0 to 200 GDUs. This is illustrated in Figure 4.4. In this illustration, each GDU is 50% smaller as is every element in the drawing frame. The scale has now been changed to 1/8" = 1'. This corresponds to scale 2 of FLAG. This is done to maintain scale integrity.

To illustrate this process, consider the following example. Suppose a 3 foot by 3 foot (real length) square is to be used to represent a pallet. To draw this pallet in the viewport, a representative scale is required such as ¼" = 1'. If the scale applies, scaled drawing of the pallet will be represented by a square that measures 3/4" on each side.

To draw this object, the shape must be defined in terms of graphic display units. The conversion formula from real inches to GDUs is:

$$\text{number of GDUs} = C \times \text{number of real feet} \tag{4.1}$$

C is a constant coefficient. It is calculated so that when using Scale number 1 in OUTPUT where the viewport is expanded to the full screen, the real length scale is ¼" = 1'. In Scale number 2, an actual scale of 1/8" = 1' is used. Each scale increase reduces the size of a GDU by 50%.

All elements on the drawing frame are defined in a specified number of GDUs for any FLAG scale. This relationship is defined by Equation 4.1. The size of the constant in the expression is reduced by

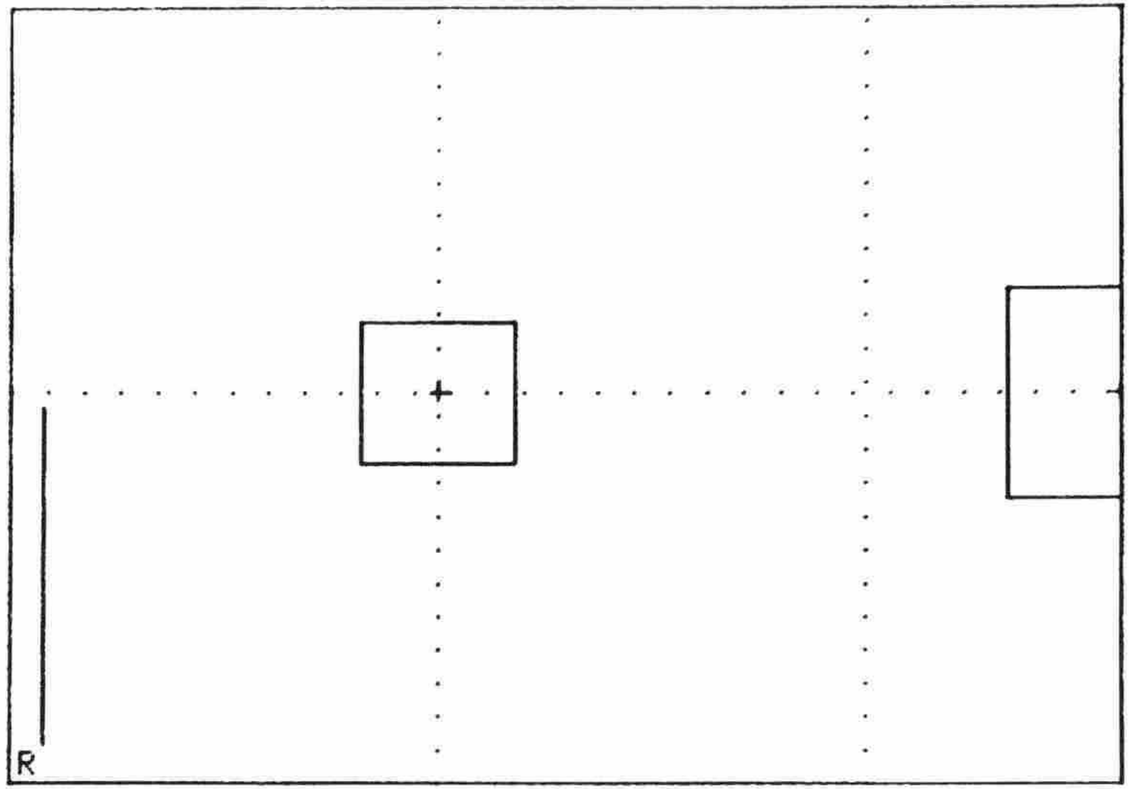
AXIS RANGES:

X AXIS:
0.0 TO 130.0
Y AXIS:
0.0 TO 100.0

DOTTED LINE
INCREMENTS 50.0

SCALE NO.: 1

R LINE = 10 FT.



SELECT OPTION AND PRESS <RET>.

1. PLACEMENT APPROVED.
2. TEMPLATE ADJUSTMENT NEEDED.
3. INCREASE FRAME AREA.

3

Figure 4.3. Display situation where drawing exceeds frame area

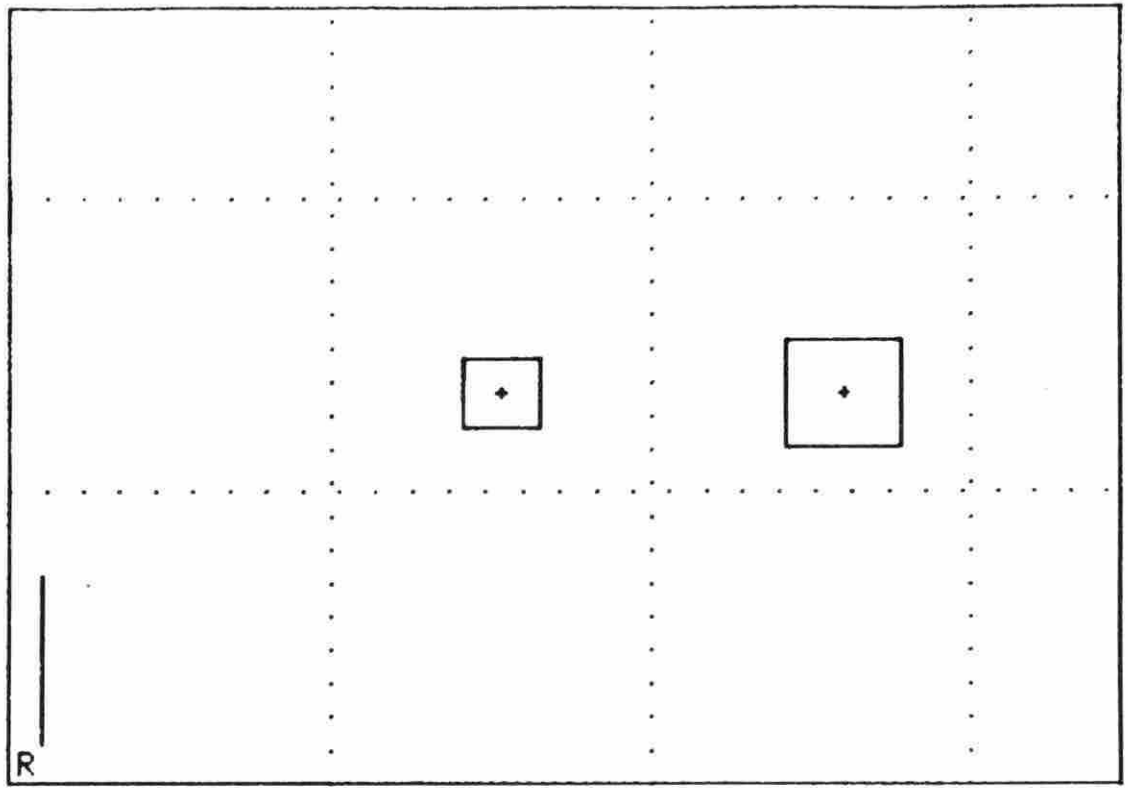
AXIS RANGES:

X AXIS:
0.0 TO 260.0
Y AXIS:
0.0 TO 200.0

DOTTED LINE
INCREMENTS 75.0

SCALE NO.: 2

R LINE = 10 FT.



SELECT OPTION AND PRESS <RET>.

1. PLACEMENT APPROVED.
2. TEMPLATE ADJUSTMENT NEEDED.
3. INCREASE FRAME AREA.

Figure 4.4. Display area scale reduced by 50%

50% for each scale increase. The scale can be increased as many times as necessary.

Table 4.1 lists a summary of scales used by FLAG. The increase of scales allows the user to reduce a scaled layout, regardless of initial size, to fall within the fixed dimensions of a display screen or plotter.

Table 4.1. FLAG scales

Scale Number	OUTPUT real length scale
1	1/4" = 1'
2	1/8" = 1'
3	1/16"= 1'
4	1/32"= 1'
.	.
.	.
.	.

D. Reference Length Indicator

To assist the user in conceptualizing real lengths, a reference length has been provided for use. This reference length will decrease in physical size as the user increments to larger scale numbers. However, the reference indicator always represents a fixed real length, in the 10 feet.

The indicator is positioned along the vertical axis of the screen in the lower right hand corner. It is labeled with the letter "R." The indicator is illustrated in Figures 4.2, 4.3, and 4.4.

The first operational step in the actual use of FLAG begins with the input of FLOWS data. This process is described in the following chapter with the use of a detailed numerical example.

V. FLOW ANALYSIS

A. Introduction

FLOWS is the initial contact between the design engineer and FLAG. It is primarily an input subroutine, where the user defines the basic nature of the facility to be designed. The data are entered in three separate program segments. These are:

- Input of general data.
- Input of flow data.
- Input of product mix and calculation of the from-to chart.

For large facilities, this data entry procedure can prove to be a lengthy task. FLOWS is designed to allow the user to enter data for each of the above segments separately. The user may therefore opt to perform data entry on as many as three separate sessions of terminal use.

A second, larger numerical example is developed in this chapter. The example references a fictitious manufacturing organization, hereafter referred to as the Ronko Manufacturing Company (RMC). The development of the Ronko manufacturing example is continued through chapter VI. In chapter VII, the FLAG layout for RMC is compared with layouts generated by OPDEP and PLANET.

B. General Data Input

General data refers to the basic data defining the nature of the plant, e.g., name and number of workcenters. These data are used several times throughout the remainder of FLAG. The general data required in this segment of FLOWS is summarized below:

1. Input the name of the facility to be designed.
2. Input your class design group number.

3a. Input the number of members in your design group.

b. Enter the name of member number i.

4a. Enter the number of products to be manufactured at the facility under design.

b. Enter the name of product number i.

5. Enter the number of workcenters (or depts.) in the facility under design.

6a. Enter the name of workcenter i

b. Enter a five character or less abbreviated label for workcenter i.

Items two and three of the above listing were incorporated for the purpose of classroom use of FLAG. They are used to label FLAG generated output.

Each workcenter is given two names which correspond to items 6a and b. Abbreviated names are used in cases where display space is limited. Where space permits, longer, unabbreviated terms are used.

The Ronko Manufacturing Company produces three separate products. These include a heater unit, a control unit, and a combination of the two which is referred to as a grain dryer. Twelve workcenters are required to manufacture these products, and are listed in Table 5.1. The RMC general data are summarized on Data Input Worksheet #1. This worksheet is illustrated in Figure 5.1.

FLAGS/FLOWS

Data Input Worksheet #1

Name of Facility (20 Characters): Ronko Mfg. Co.

Design Group No: 3

Members of Design Group (15 characters each):

1. Robert Nutting
2. Dwyla Kennison
3. Sherman Severin
4. ________
5. ________
6. ________

Names of products (10 characters):

1. Heater Unit
2. Control Unit
3. Grain Dryer
4. ________
5. ________

Number of Workcenters: 12

Names of Workcenters (15 characters each)/Abbreviated Names (5 characters each)

1. Shipping/Rec. / S/R
2. Decoiler/Shear / D/S
3. Punch Press A / PPA
4. Runch Press B / PPB
5. Roller / Rllr
6. Flanger / Flgr
7. Seam Welder / SmW
8. Spot Welder / SpW
9. Wire Cutter / WC
10. Packaging / Pkg
11. Assembly / Assem
12. Painting / Ptg

Figure 5.1. Data input worksheet number 1

Table 5.1. RMC workcenters

Dept. No.	Workcenter or Department	Abbreviation
1	Shipping/Receiving	S/R
2	Decoiler/Shear	D/S
3	Punch Press A	PPA
4	Punch Press B	PPB
5	Roller	Rllr
6	Flanger	Flgr
7	Seam Welding	SmW
8	Spot Welding	SpW
9	Wire Cutting	WC
10	Packaging	Pkg
11	Assembly	Assem
12	Painting	Ptg

C. Flow Data

After completion of general data input, the user must enter the flow data. These data are used to calculate the from-to chart.

To enter the flow data, individual flow paths for each product must first be defined. To illustrate this, consider the heating unit. The flow path for the heating unit is depicted in Figure 5.2. The heating unit consists of only two basic elements. Both elements begin flow paths in shipping/receiving.

The heating coils are purchased by RMC and go directly to assembly where they are attached to the fabricated shell. The shell starts

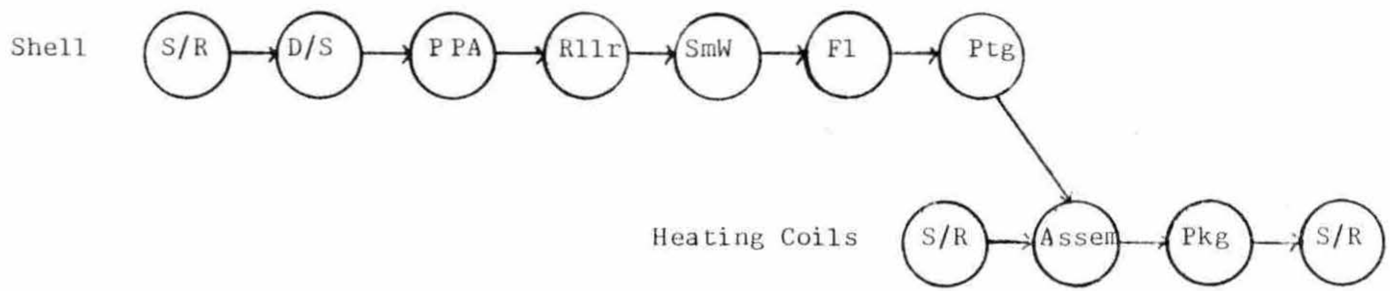


Figure 5.2. Product A: Basic flows for heating unit

in S/R as coiled steel. The steel is transported to the decoiler/shear, where it is cut to length. The steel is then routed through Punch Press A, the Seam Welder, Flanger, Painting, to Assembly. From assembly, the shell, now with heating coils, goes to packaging and shipping/ receiving.

Each circular node in Figure 5.2 represents a workcenter. The connecting lines represent the flows between workcenters. It is these flows that must be defined.

Each product must be analyzed in this fashion. Figure 5.3 illustrates the flow path for the control unit. A control unit is combined with a heating unit to comprise a finished grain dryer. The combined flow path for the grain dryer is illustrated in Figure 5.4.

For each interdepartmental flow, the following data must be entered.

1. Enter the number of interdepartmental flows associated with the product i.

2a. For product/flow j enter the number of units per product.

b. For product/flow j, enter the units per move.

c. For product/flow j, enter the move cost/foot.

d. Enter the number of the workcenter product/flow j departs.

e. Enter the number of the workcenter product/flow j enters.

User responses to the above statements must be entered for each interdepartmental flow. Interdepartmental flow data for RMC are summarized in Table 5.2. RMC makes three products. Table 5.2 summarizes the flows for the heating and control units. Since the grain dryer is a combination of a heating unit with a control unit, the flows are a duplication

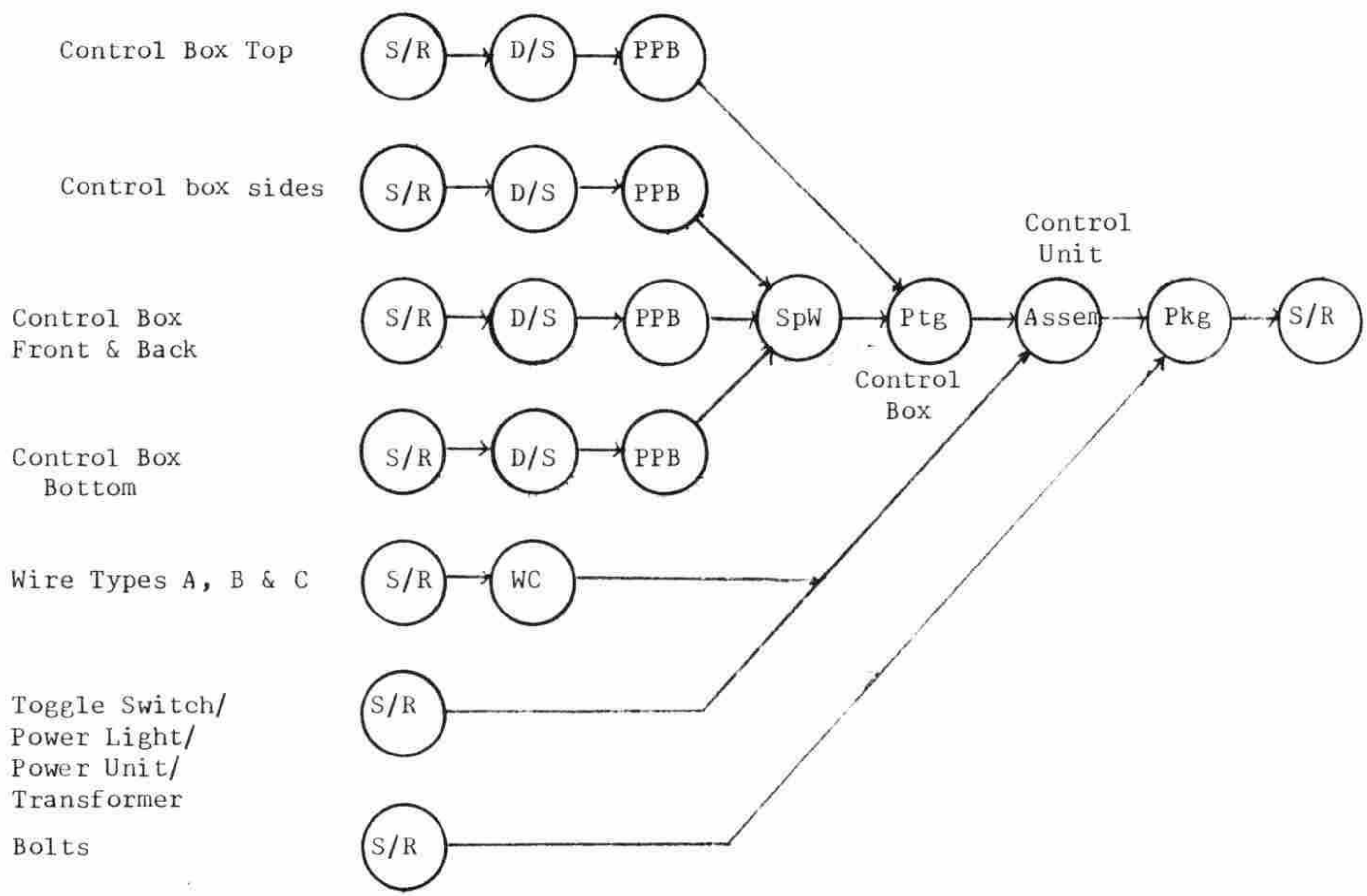


Figure 5.3. Product B: Basic flows for control unit

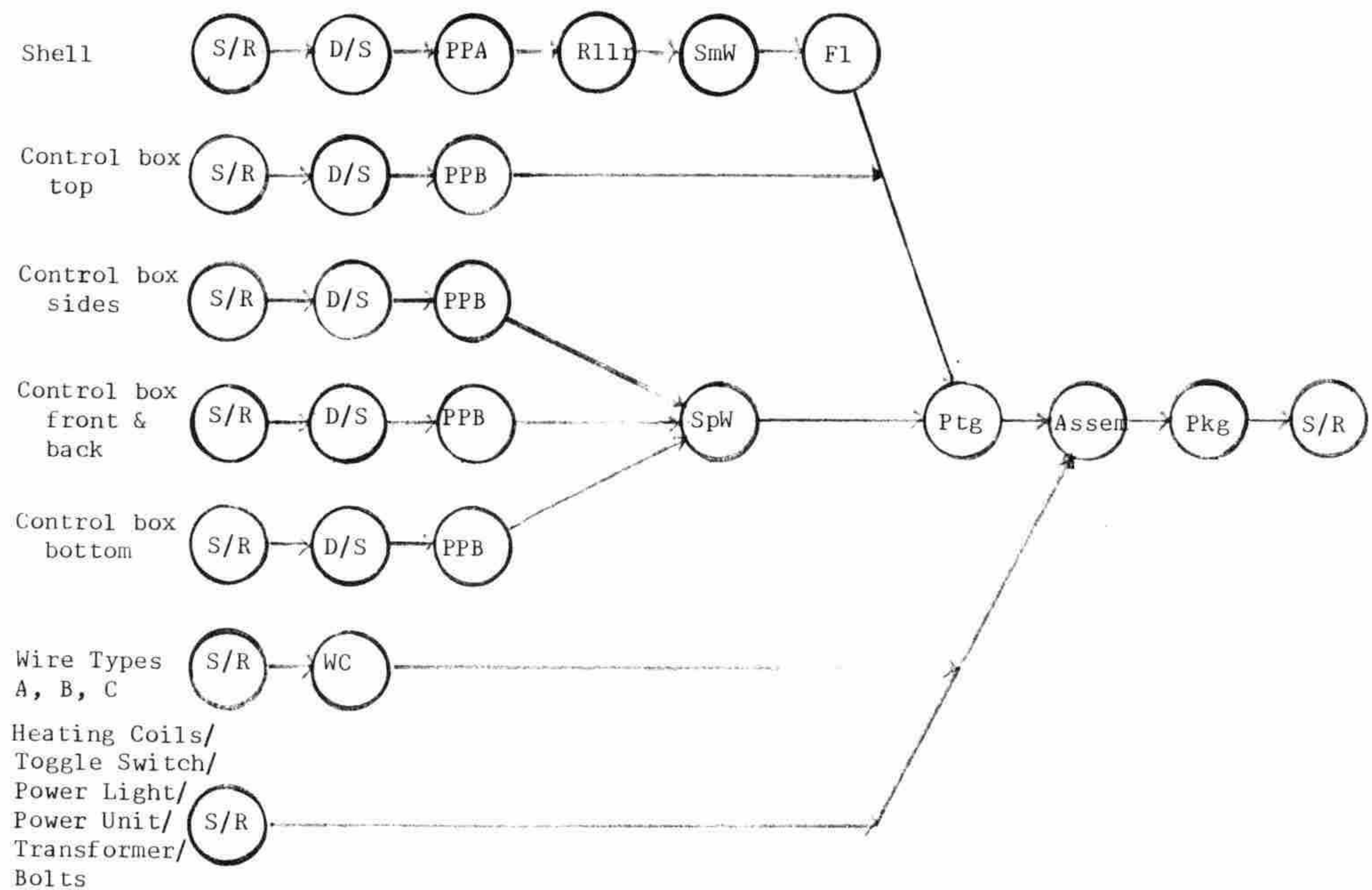


Figure 5.4. Product C: Basic flow for grain dryer

of flows in Table 5.2 with one exception. The bolts for the control unit travel from shipping/receiving to assembly instead of packaging.

D. Calculation of the From-to Chart

The final phase of the FLOWS program is the input of the product mix and subsequent calculation of the from-to chart. As stated in chapter III, the from-to chart is utilized as the REL chart in the FLAG layout discipline. The method of calculation for the from-to chart is also detailed in chapter III.

The from-to chart for RMC is illustrated in Table 5.3. This from-to chart is based on a product mix of 3000 heating units, 3000 control units, and 5000 complete grain dryers.

After the completion of input of the flow data for the facility under design, the user is prepared to enter the second phase of the FLAG routine. This is the design of the facility workcenters and generation of the layout. This process was described in considerable detail in chapter III.

The following chapter continues to illustrate this design procedure using the example initially developed and presented in this chapter.

Table 5.2. Interdepartmental flows - heating unit

Part Flow Segment	Interdepartmental Flow	Flow Number	Number in Final Product	Number of Units per Move	Move/ Cost/ Foot	From		To	
						No.	Abbr.	No.	Abbr.
SHELL (SH)	Coiled Steel type A	1	1	250	1.5	1	S/R	2	D/S
	Sheared Steel	2	1	20	.7	2	D/S	3	PPA
	Punched Steel	3	1	20	.25	3	PPA	5	R11r
	Bended Steel	4	1	3	.25	5	R11r	7	SmW
	Heater Shell	5	1	3	.25	7	SmW	6	Flgr
	Flanged Shell	6	1	3	.25	6	Flgr	12	Ptg
	Painted Shell	7	1	3	.25	12	Ptg	11	Assem
Heating Coils (HC)	Coils	8	3	90	.20	1	S/R	11	Assem
Heating Unit	Assembled Unit	9	1	3	.25	11	Assem	10	Pkg
(Final product A)	Packaged Unit	10	1	2	.25	10	Pkg	1	S/R

Table 5.2. Continued - control unit

Part Flow Segment	Interdepartmental Flow	Flow Number	Number in Final Product	Number of Units per Move	Move/Cost/Foot	From		To	
						No.	Abbr.	No.	Abbr.
Control Box Top	Coiled Steel Type B	1	1	600	1.5	1	S/R	2	D/S
	Sheared Top	2	1	25	.20	2	D/S	4	PPB
	Punched Top	1	1	25	.20	4	PPB	12	Ptg
	Painted Top	4	1	25	.20	12	Ptg	11	Assem
Control Box Sides	Coiled Steel Type B	5	2	600	1.5	1	S/R	2	D/S
	Sheared Sides	6	2	30	.20	2	D/S	4	PPB
	Punched Sides	7	2	30	.20	4	PPB	8	SpW
Control Box Front & Back	Coiled Steel Type B	8	2	500	1.5	1	S/R	2	D/S
	Sheared Parts	9	2	20	.20	1	D/S	4	PPB
	Punched Parts	10	2	20	.20	4	PPB	8	SpW
Control Box Bottom	Coiled Steel Type B	11	2	750	1.5	1	S/R	2	D/S
	Sheared Bottom	12	2	50	.20	2	D/S	4	PPB
	Punched Bottom	13	2	50	.20	4	PPB	8	SpW

Table 5.2. Continued

Part Flow Segment	Interdepartmental Flow	Flow Number	Number in Final Product	Number of Units per Move	Move/ Cost/ Foot	From		To	
						No.	Abbr.	No.	Abbr.
Control Box	Welded Control Box	14	1	20	.25	8	SpW	12	Ptg
	Painted Control Box	15	1	20	.25	12	Ptg	11	Assem
Wire Type A	Rolled Wire Type A	16	6	3000	.25	1	S/R	9	SC
	Cut Wire Type A	17	6	1000	.15	9	WC	11	Assem
Wire Type B	Rolled Wire Type B	18	4	3000	.25	1	S/R	9	WC
	Cut Wire Type B	19	4	1000	.15	9	WC	11	Assem
Wire Type C	Rolled Wire Type C	20	2	3000	.25	1	S/R	9	WC
	Cut Wire Type C	21	2	1000	.15	9	WC	11	Assem
Toggle Switch	Toggle Switch	22	1	500	.10	1	S/R	11	Assem
Power Light	Power Light	23	1	500	.10	1	S/R	11	Assem
Power Unit	Power Unit	24	1	25	.10	1	S/R	11	Assem
Transformer	Transformer	25	1	10	.10	1	S/R	11	Assem
Bolts	For Packaging	26	6	600	.30	1	S/R	10	Pkg
Control Unit (Final product B)	Assembled Unit	27	1	20	.30	11	Assem	PO	Pkg
	Packaged Unit	28	1	10	.20	10	Pkg	1	S/R

Table 5.3. From-to chart for RMC move costs/foot assuming product mix of 2500 heaters, 3000 control units, and 5000 grain dryers

To/ From	1	2	3	4	5	6	7	8	9	10	11	12
1. S/R	0	198	0	0	0	0	0	0	9.25	9.3	181.5	0
2. D/S	0	0	263.9	396.2	0	0	0	0	0	0	0	0
3. PPA	0	0	0	0	263.9	0	0	0	0	0	0	0
4. PPB	0	0	0	0	0	0	0	331.8	0	0	0	64.4
5. Rllr	0	0	0	0	0	0	625.25	0	0	0	0	0
6. FL	0	0	0	0	0	0	0	0	0	0	0	625.25
7. SmW	0	0	0	0	0	625.25	0	0	0	0	0	0
8. SpW	0	0	0	0	0	0	0	0	0	0	0	100.5
9. WC	0	0	0	0	0	0	0	0	0	0	15.3	0
10. Pkg	2248.7	0	0	0	0	0	0	0	0	0	0	0
11. Assem	0	0	0	0	0	0	0	0	0	1509.05	0	0
12. Ptg	0	0	0	0	0	0	0	0	0	0	690.11	0

VI. A FLAG EXAMPLE

A. Introduction

In the preceding chapter, a flow analysis was developed for the Ronko Manufacturing Company. This organization will require a twelve workcenter facility to manufacture grain dryers. The flow data for RMC were entered using the FLOWS subroutine. The user is now ready to begin the design process outlined below:

1. Design of Workcenters - WORKOUT
2. Design of Layout - LAYOUT
3. Improvements on Layout - LAYOUT

A series of illustrations detailing some of the mechanics of working with FLAG is also presented in this chapter.

B. Workcenter Design

1. Introduction

The user begins the design process by developing the final configuration for each workcenter for use by the LAYOUT and OUTPUT programs. This process is demonstrated by illustrating the primary steps taken to develop the decoiler/shear workcenter.

2. Template selection and placement

The workcenter design process begins by selecting the first template for placement. In this case, a template to be selected will represent the shear. The user is first presented with the option of a general shape template or a PLANPRINT machine template. In this instance, the latter is selected. WORKOUT then gives the user a menu of the thirty-three PLANPRINT machine templates available. These templates are reproduced in Appendix III. The user selects the number associated with the shear.

The user is now prepared to place the shear on the drawing frame (see Figure 6.1). The initial placement position of the shear is specified as 50.0 on the X-axis, 50.0 on the Y-axis, with a 90° rotation (see Figure 6.1). WORKOUT then places the shear on the drawing grid according to user specifications as shown in Figure 6.2. The option is then given to either increase the scale or to adjust the position of the template.

When the user accepts the placement of the shear, WORKOUT proceeds with the selection of the second template. This time the user refers to the general template library to select a template to represent a decoiler. The general template menu is then presented with the following options:

1. square
2. rectangle
3. double rectangle

The user, after selecting a rectangle to represent the decoiler, must now specify the dimensions of that rectangle (see Figure 6.3). As shown, the user has specified a width of 70.0 inches and a length of 36 inches.

When using a general template, the user must also specify that template label; to be used in FLAG output. In this instance, the template is labeled 'decoiler'. Again, the user is asked to specify the placement coordinates.

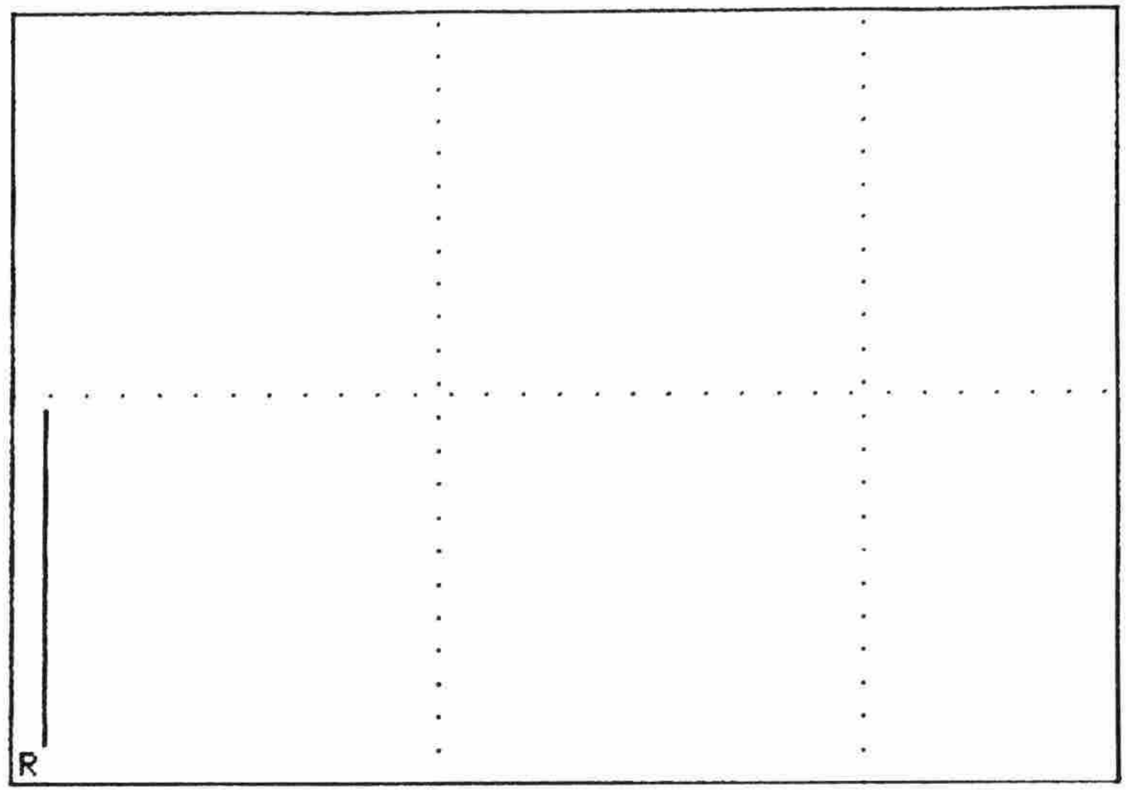
AXIS RANGES:

X AXIS:
0.0 TO 130.0
Y AXIS:
0.0 TO 100.0

DOTTED LINE
INCREMENTS 50.0

SCALE NO.: 1

R LINE = 10 FT.



INPUT X-COORDINATE, Y-COORDINATE, AND
ROTATION FACTOR SEPARATED BY COMMAS.
50.,50.,90.

Figure 6.1. Entering initial placement coordinates for shear

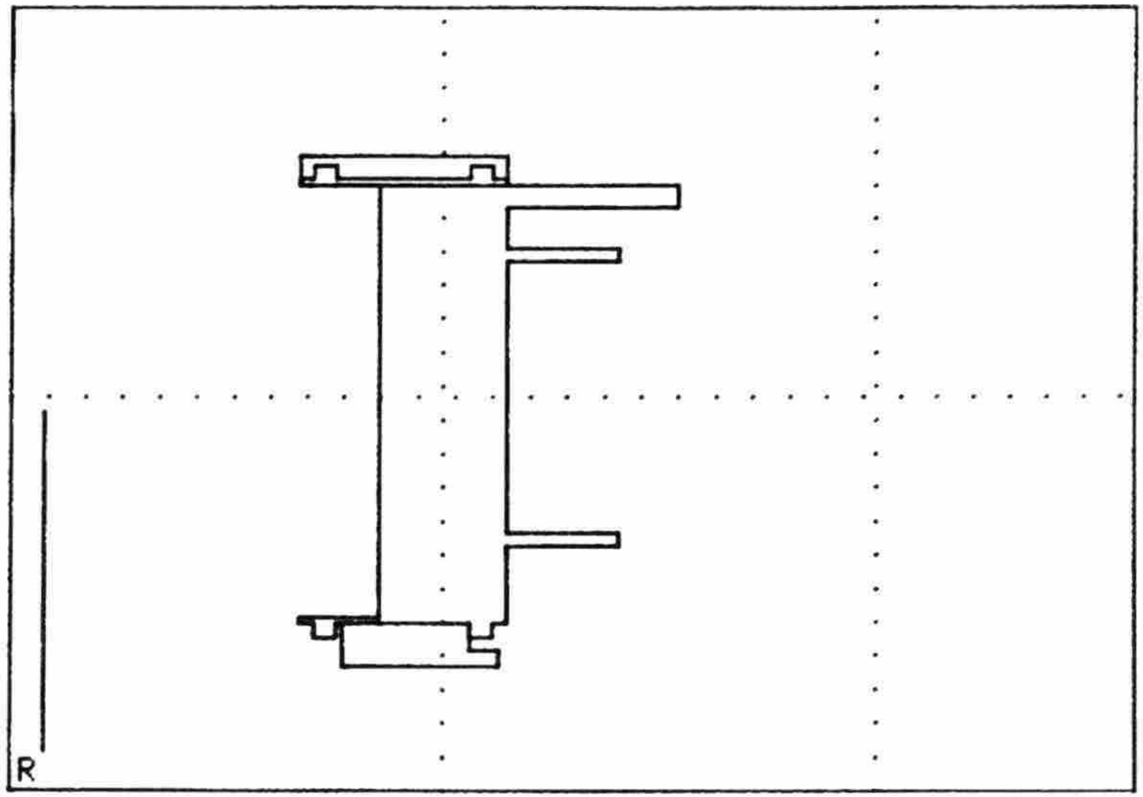
AXIS RANGES:

X AXIS:
0.0 TO 130.0
Y AXIS:
0.0 TO 100.0

DOTTED LINE
INCREMENTS 50.0

SCALE NO.: 1

R LINE = 10 FT.



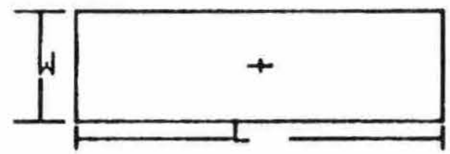
SELECT OPTION AND PRESS <RET>.

1. PLACEMENT APPROVED.
2. TEMPLATE ADJUSTMENT NEEDED.
3. INCREASE FRAME AREA.

1

Figure 6.2. Initial placement of shear

GENERAL TEMPLATE: RECTANGLE




```
INPUT THE WIDTH (W) MEASUREMENT IN TERMS
     OF INCHES.
70.

INPUT THE LENGTH (L) MEASUREMENT IN TERMS
     OF INCHES.
36.

INPUT TEMPLATE LABEL
DECOILER
```

Figure 6.3. Rectangle input parameters

After entering an X coordinate of 100.0, a Y coordinate of 50.0, and a 0.0 degree rotation, WORKOUT draws the initial placement of the decoiler (see Figure 6.4). Following the same routine, the user places two pallets in the workcenter, (see Figure 6.5) completing the element placement phase for this workcenter.

3. Placement of workcenter borders

After the placement of all elements, WORKOUT establishes initial borders (see Figure 6.6). The user has the option to adjust any border, by shifting the borders vertically or horizontally. This phase refers to the pre-notched borders. After finalizing border placement, the user can then notch any of the four corners desired. This process was illustrated in chapter III.

4. Placement of POE and POD

The last phase of workout is the placement of the Point of Entry (POE) and Point of Departure (POD). These points are for all flows in and out of the workcenter. The user begins placement of the POE by specifying the border in which the POE is placed. This is illustrated in Figure 6.7. By specifying the border, the user has specified one of the two coordinates. In this case, the user has established the X coordinate, since it is equal to that of the east border.

The next step is for the specification of the Y coordinate. WORKOUT informs the user of the acceptable range (see Figure 6.8). This is to assure placement on the eastern border. After entering an acceptable Y coordinate value, the POE is placed (see Figure 6.9).

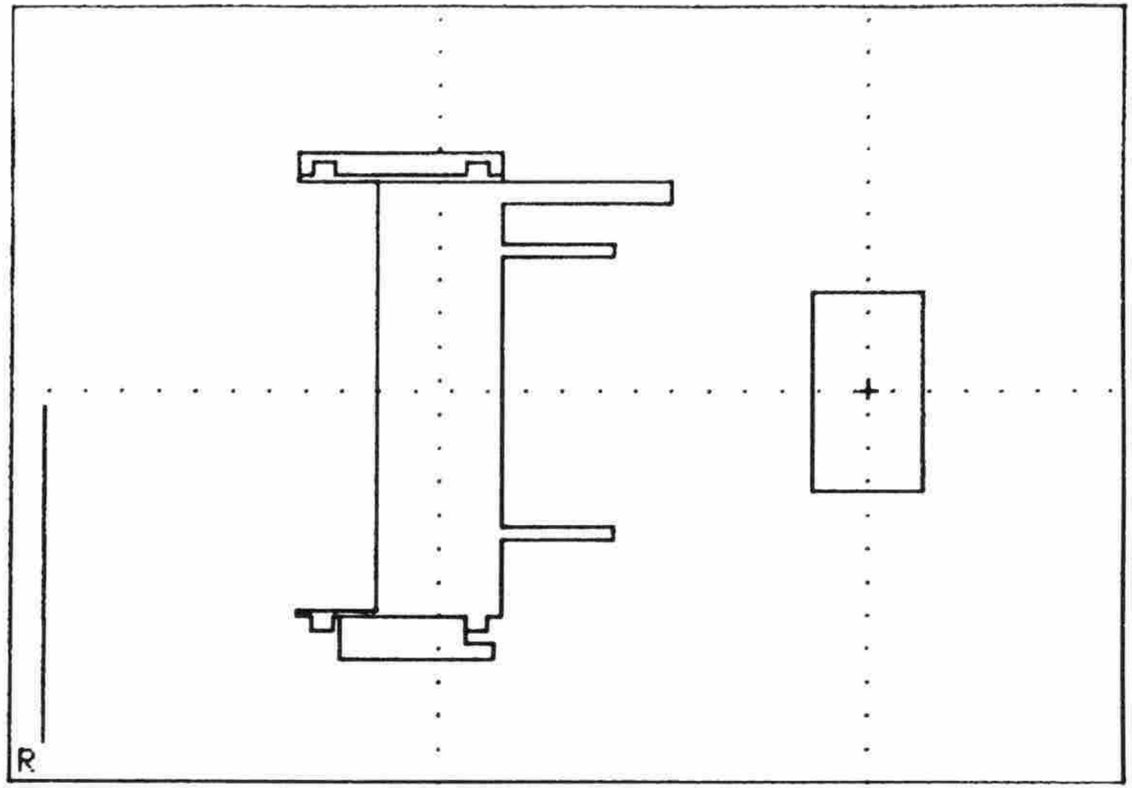
AXIS RANGES:

X AXIS:
0.0 TO 130.0
Y AXIS:
0.0 TO 100.0

DOTTED LINE
INCREMENTS 50.0

SCALE NO.: 1

R LINE = 10 FT.



SELECT OPTION AND PRESS <RET>.

1. PLACEMENT APPROVED.
2. TEMPLATE ADJUSTMENT NEEDED.
3. INCREASE FRAME AREA.

1

Figure 6.4. Decoiler placement

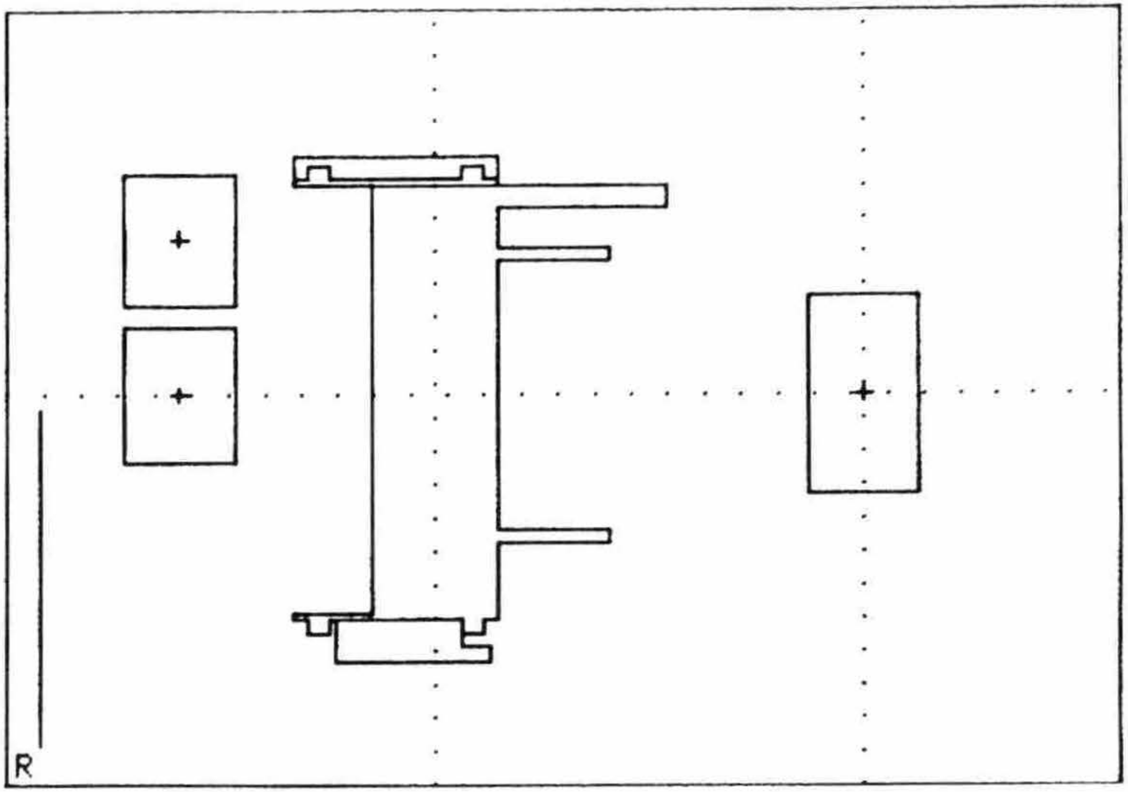
AXIS RANGES:

X AXIS:
0.0 TO 130.0
Y AXIS:
0.0 TO 100.0

DOTTED LINE
INCREMENTS 50.0

SCALE NO.: 1

R LINE = 10 FT.



SELECT OPTION AND PRESS <RET>.

1. PLACEMENT APPROVED.
2. TEMPLATE ADJUSTMENT NEEDED.
3. INCREASE FRAME AREA.

Figure 6.5. Final placement of all elements

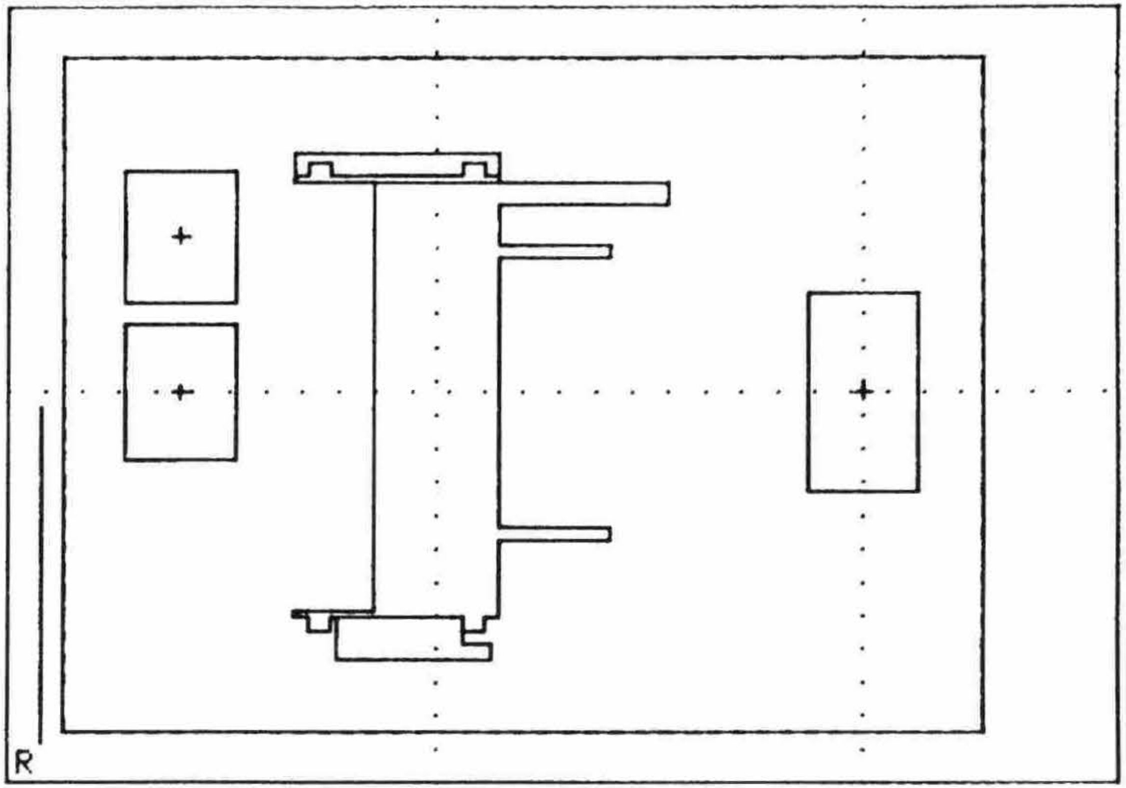
AXIS RANGES:

X AXIS:
0.0 TO 130.0
Y AXIS:
0.0 TO 100.0

DOTTED LINE
INCREMENTS 50.0

SCALE NO.: 1

R LINE = 10 FT.

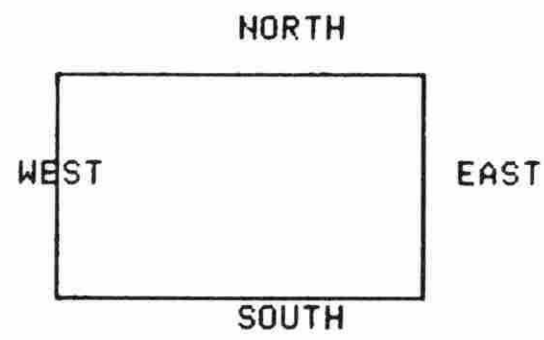


SELECT OPTION AND PRESS <RET>.
1. ACCEPT PRE-NOTCHED BORDERS.
2. ADJUST PRE-NOTCHED BORDERS.
3. INCREASE FRAME AREA.

1

Figure 6.6. Initial borders

INDICATE THE BORDER YOU DESIRE TO PLACE
THE POE.



1. NORTH
2. SOUTH
3. EAST
4. WEST

3

Figure 6.7. Border selection for POE

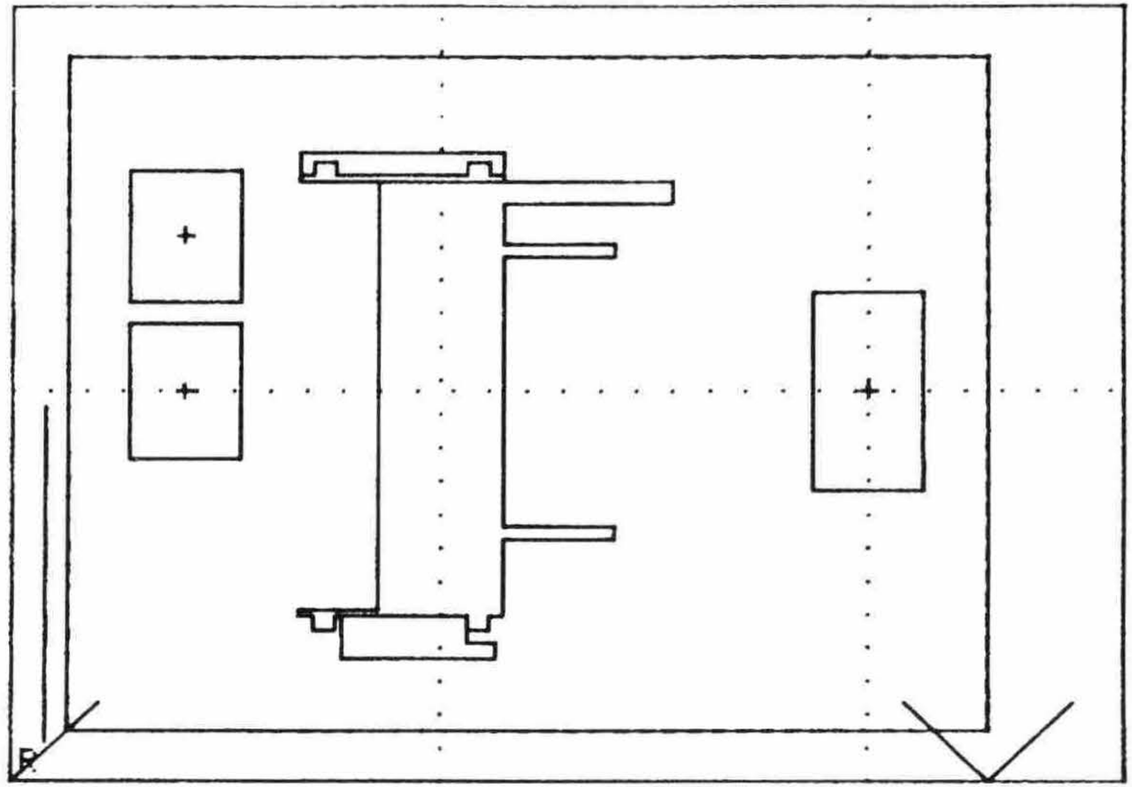
AXIS RANGES:

X AXIS:
0.0 TO 130.0
Y AXIS:
0.0 TO 100.0

DOTTED LINE
INCREMENTS 50.0

SCALE NO.: 1

R LINE = 10 FT.



INDICATE AT WHAT POINT ALONG THE
EAST BORDER YOU DESIRE TO PLACE THE
POE.
ACCEPTABLE RANGE: 6.39 - 93.61

Figure 6.8. Acceptable range for POE placement

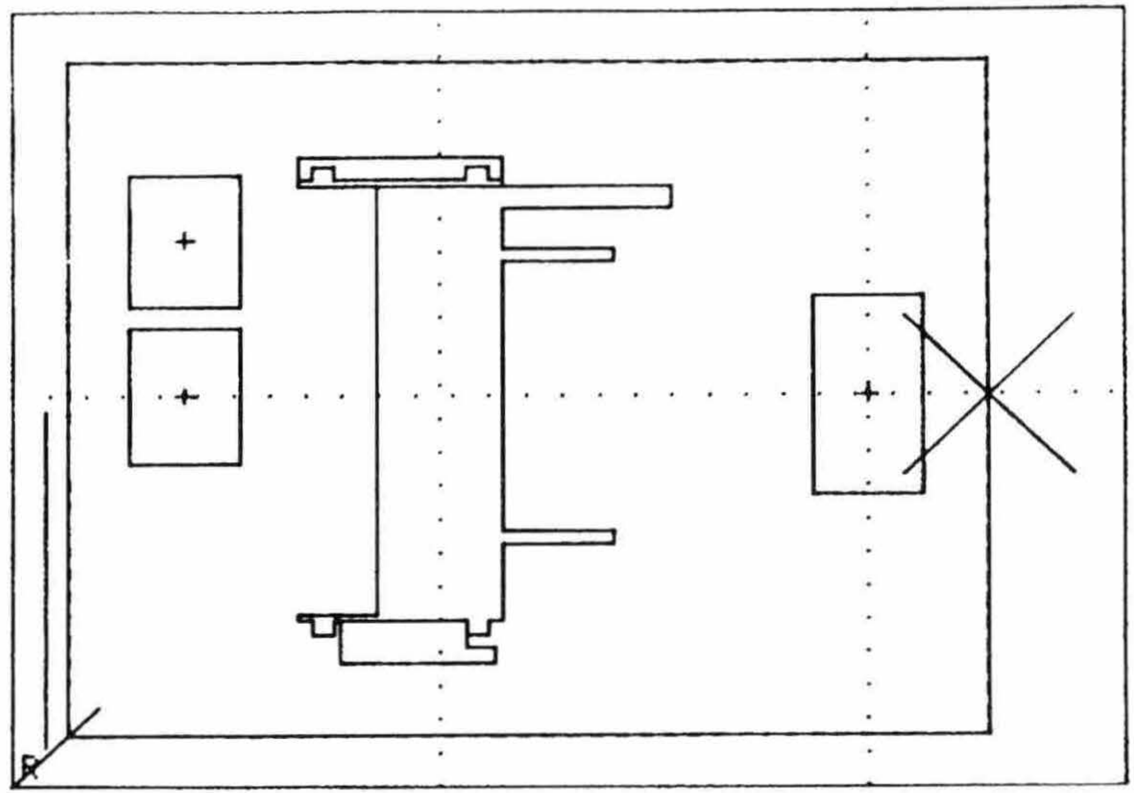
```
AXIS RANGES:

  X AXIS:
    0.0 TO  130.0
  Y AXIS:
    0.0 TO  100.0

DOTTED LINE
  INCREMENTS  50.0

  SCALE NO.:  1

  R LINE = 10 FT.
```




```
INPUT OPTION AND PRESS <RET>.
          1. ACCEPT POE.
          2. ADJUST POE.
```

Figure 6.9. POE placement

The POD is placed in a similar fashion (see Figure 6.10), completing the design of the Decoiler/Shear workcenter. All data on this workcenter are stored for use by LAYOUT and OUTPUT.

5. Design of remaining workcenters

During the execution of WORKOUT, each of the twelve workcenters for the RMC example was designed in the above fashion just described. Final workcenter layouts are reproduced in Figures 6.11 through 6.22. Each workcenter dimensions were measured on the originals of these figures, without duplication reduction. From these measurements, final workcenter areas were calculated. These values are presented in Table 6.1.

C. Layouts

After completion of WORKOUT, the user enters LAYOUT, by executing a VAX run command. This is done to generate a FLAG layout in the manner described in chapter III. The process starts with the user selecting the first department for placement. Although any department may be selected, it is best to select that workcenter which will have most interaction with external elements to the facility.

When executing WORKOUT, it is periodically necessary to increase the scale on the drawing frame. When these increases are made, the placement points of workcenter elements are translated to the center of the drawing fram. This allows the user to place additional elements anywhere around those templates already positioned.

In LAYOUT, there is also a necessity to make periodic scale changes. When these scale changes occur, workcenters are not translated to the

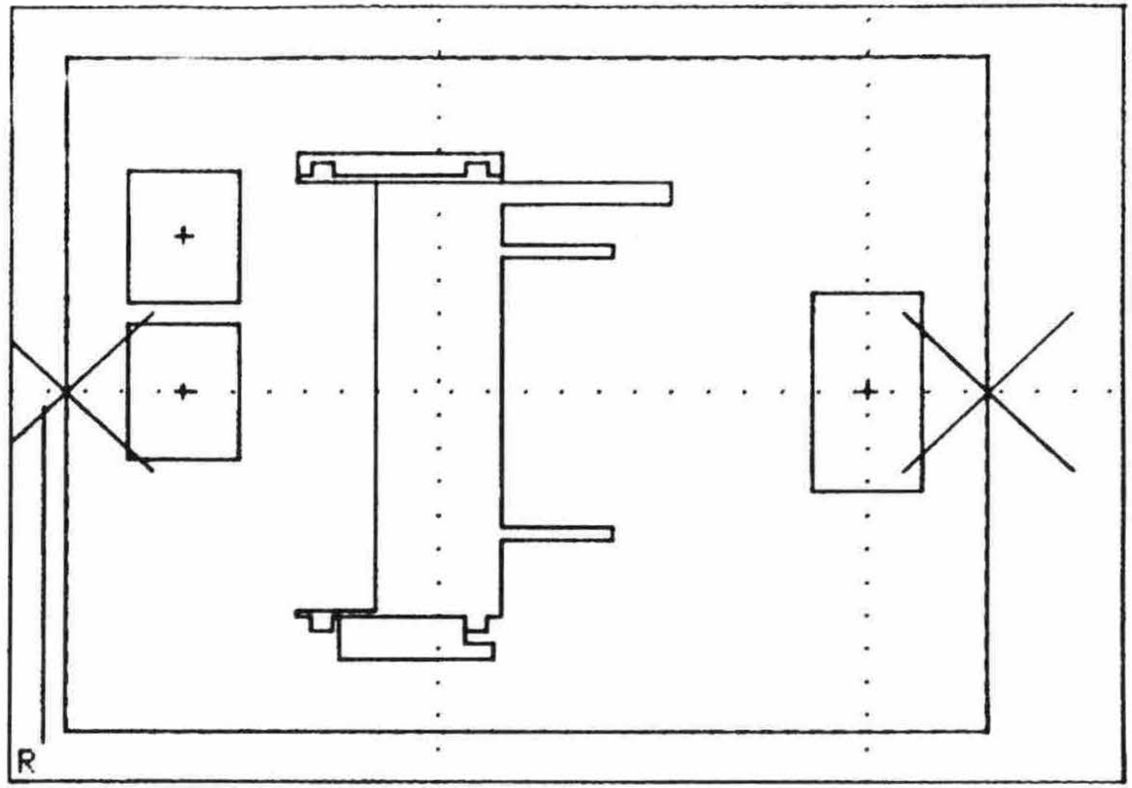
```
AXIS RANGES:

  X AXIS:
    0.0 TO  130.0
  Y AXIS:
    0.0 TO  100.0

DOTTED LINE
  INCREMENTS  50.0

  SCALE NO.:  1

  R LINE = 10 FT.
```




```
INPUT OPTION AND PRESS <RET>.
         1. ACCEPT POD.
         2. ADJUST POD.
1
```

Figure 6.10 Final design of shear/decoder with POE and POD

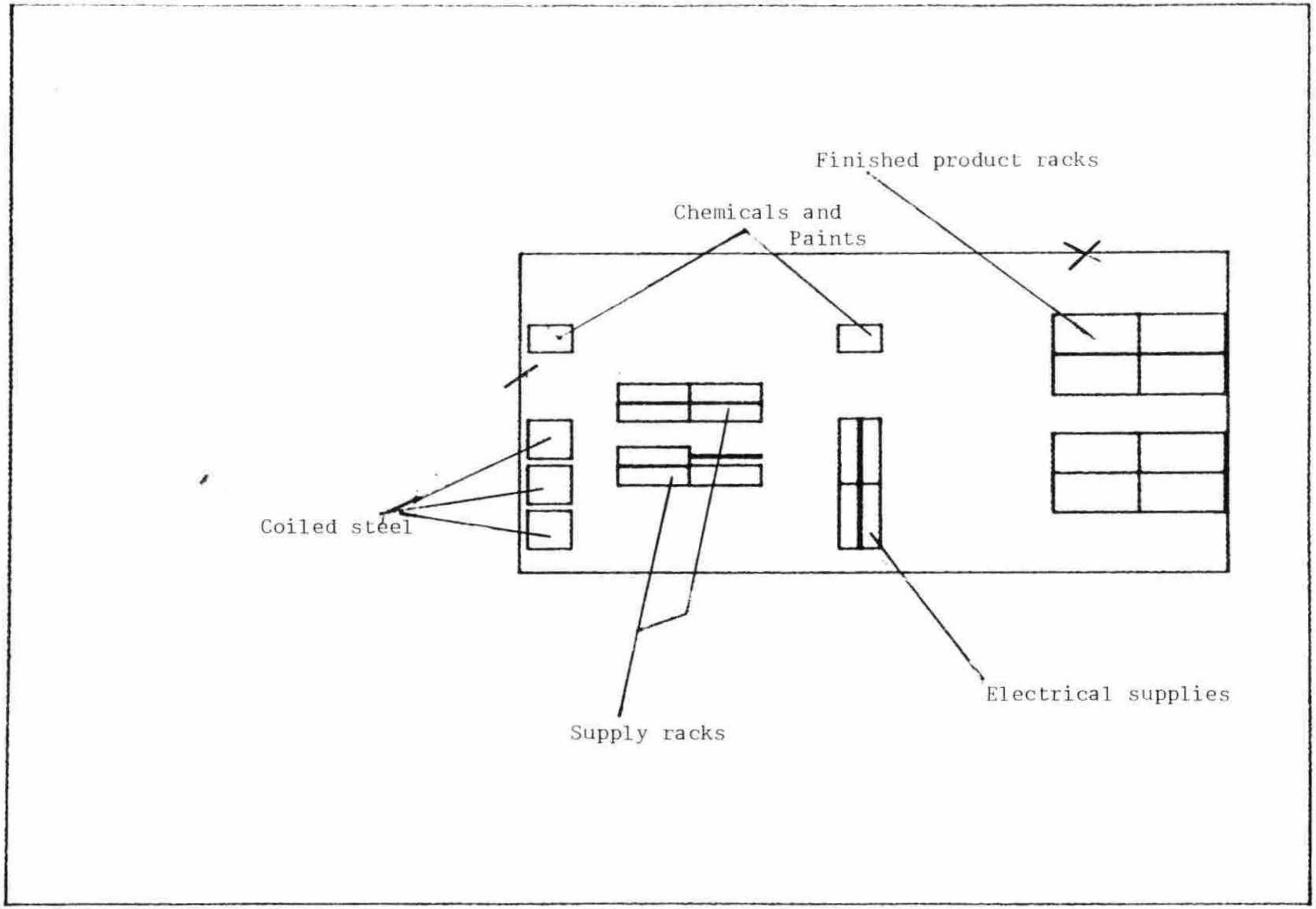


Figure 6.11. Shipping and receiving (scale 5)

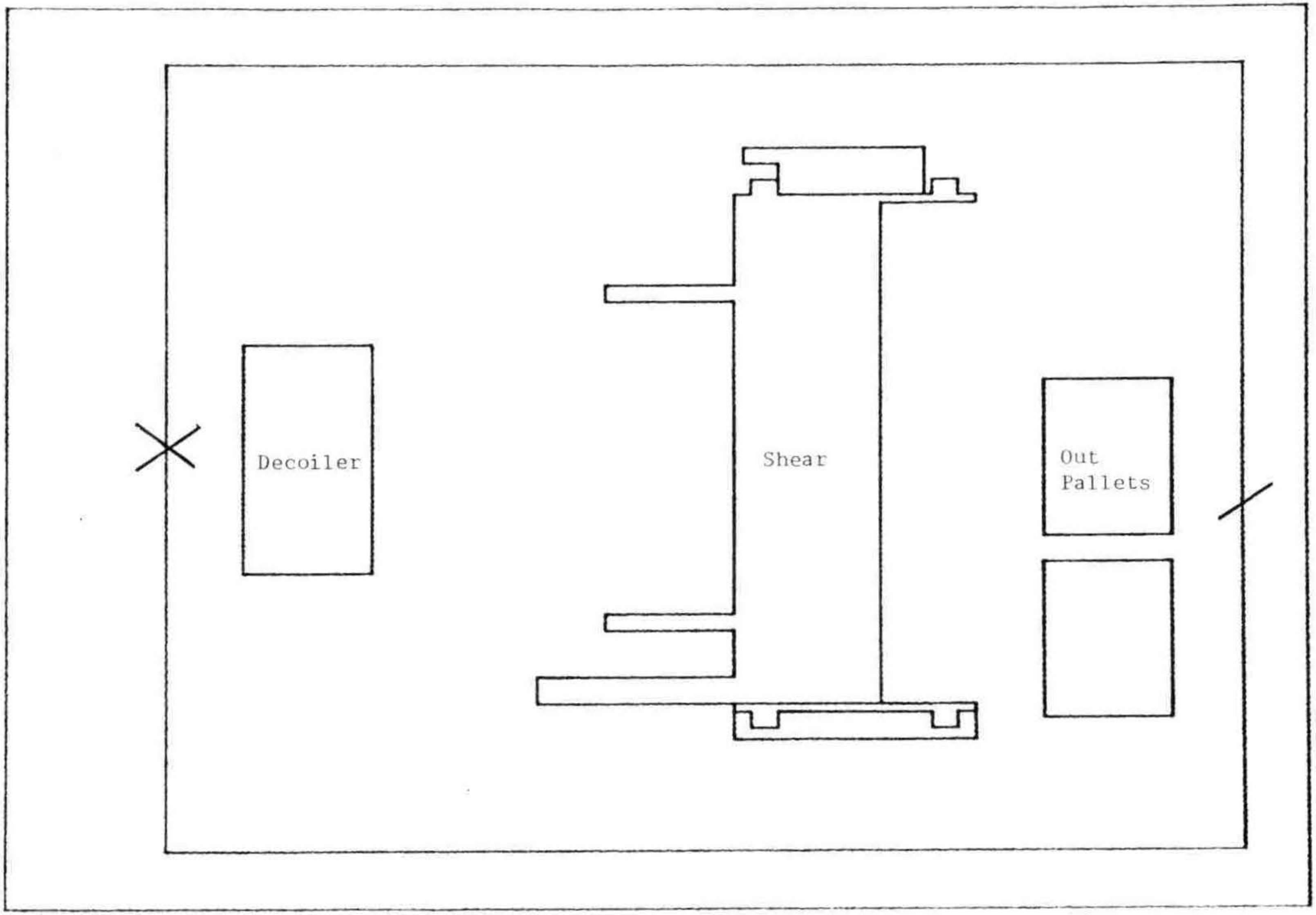


Figure 6.12. Decoiler/shear (scale 1)

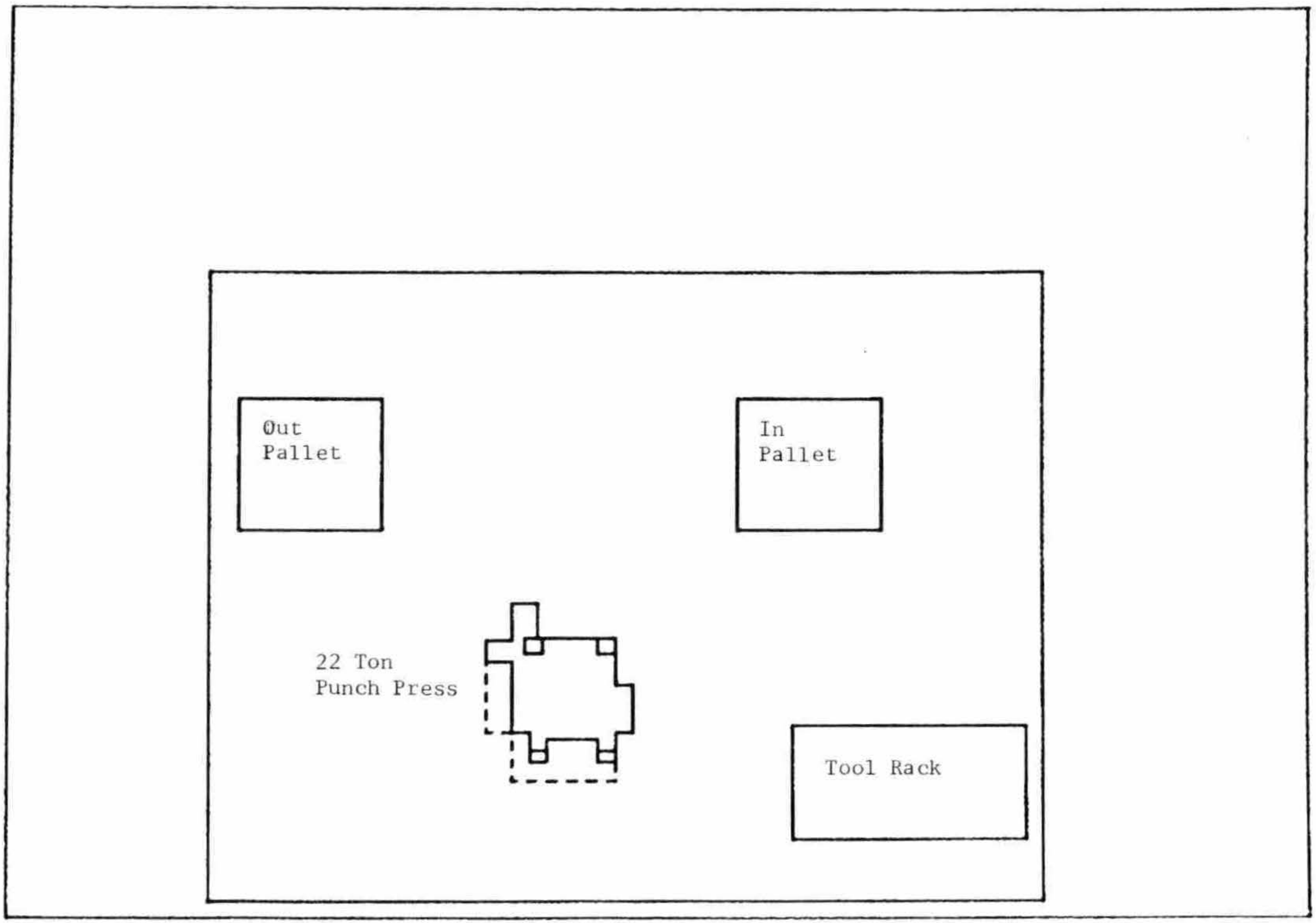


Figure 6.13. Punch press A (scale 1)

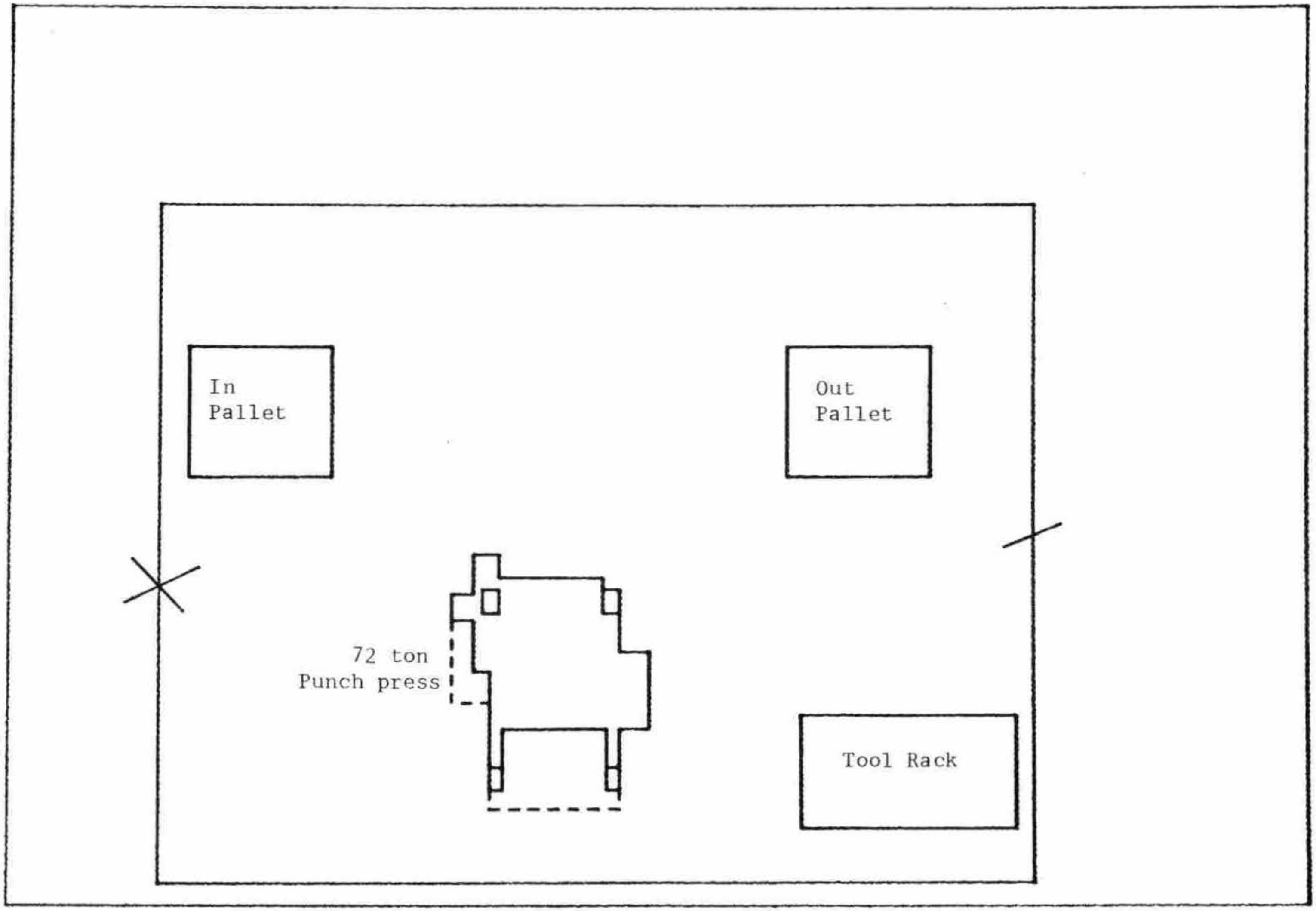


Figure 6.14. Punch press B (scale 1)

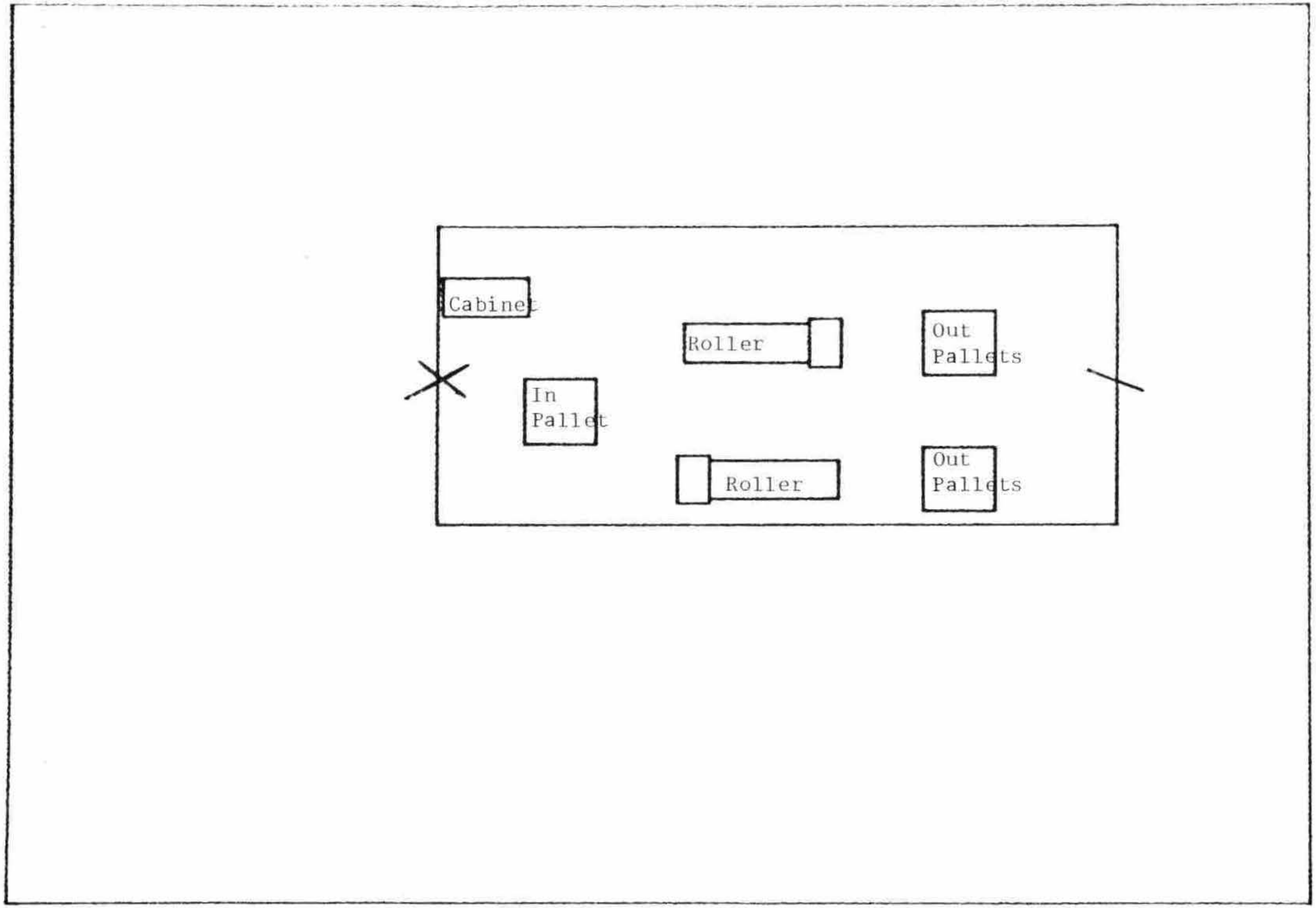


Figure 6.15. Roller (scale 2)

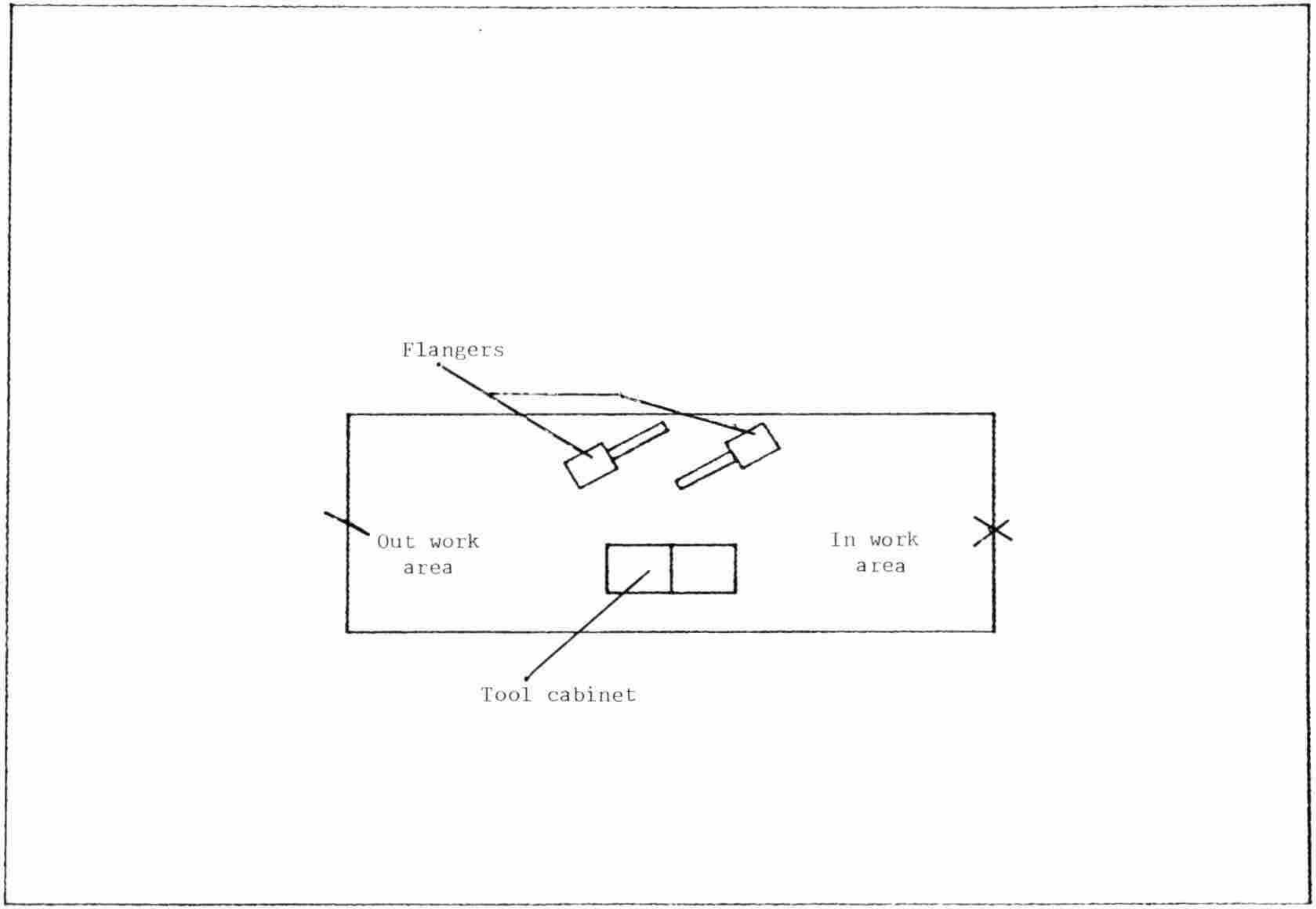


Figure 6.16. Flanger (scale 2)

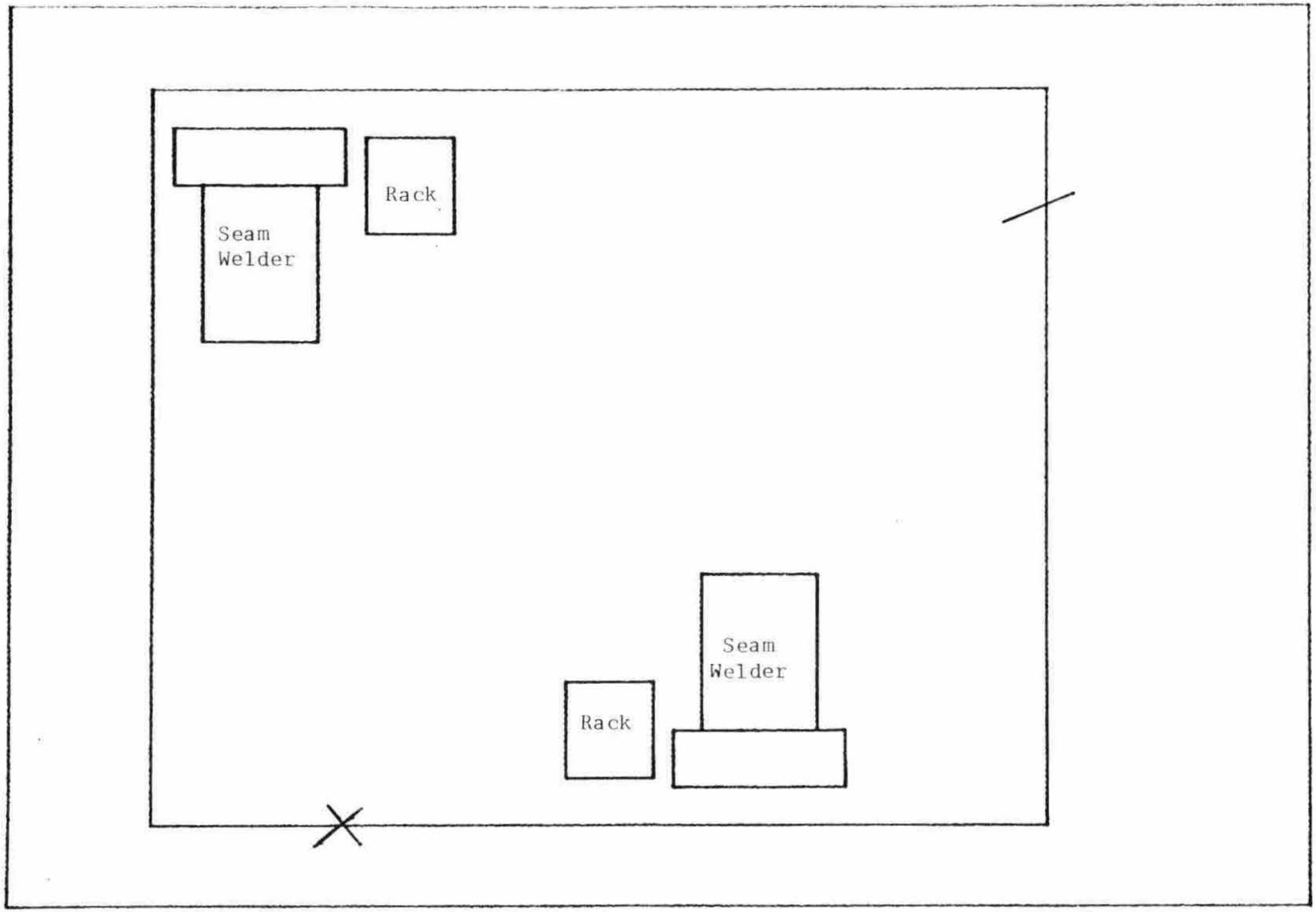


Figure 6.17. Seam welder (scale 1)

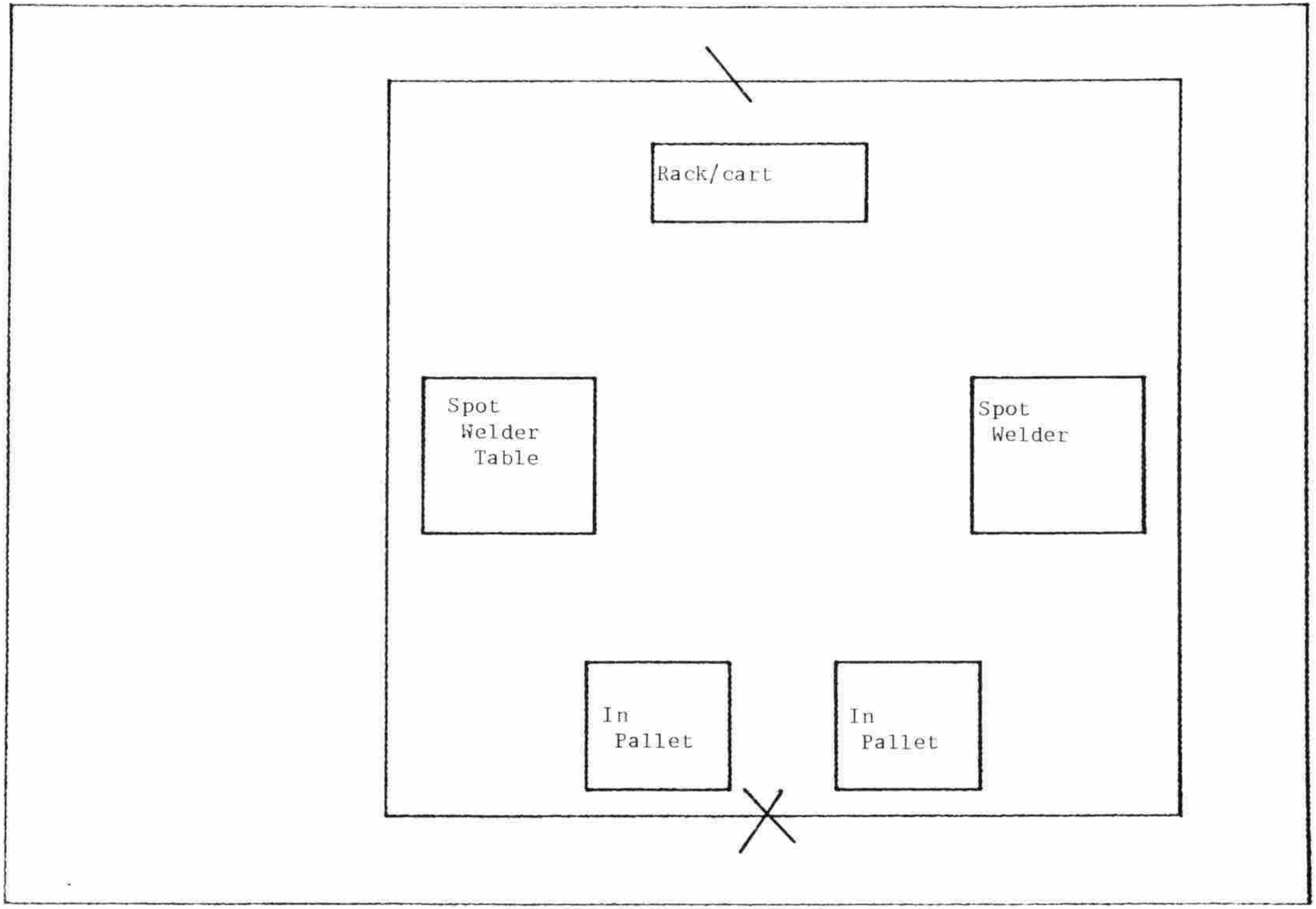


Figure 6.18. Spot welder (scale 1)

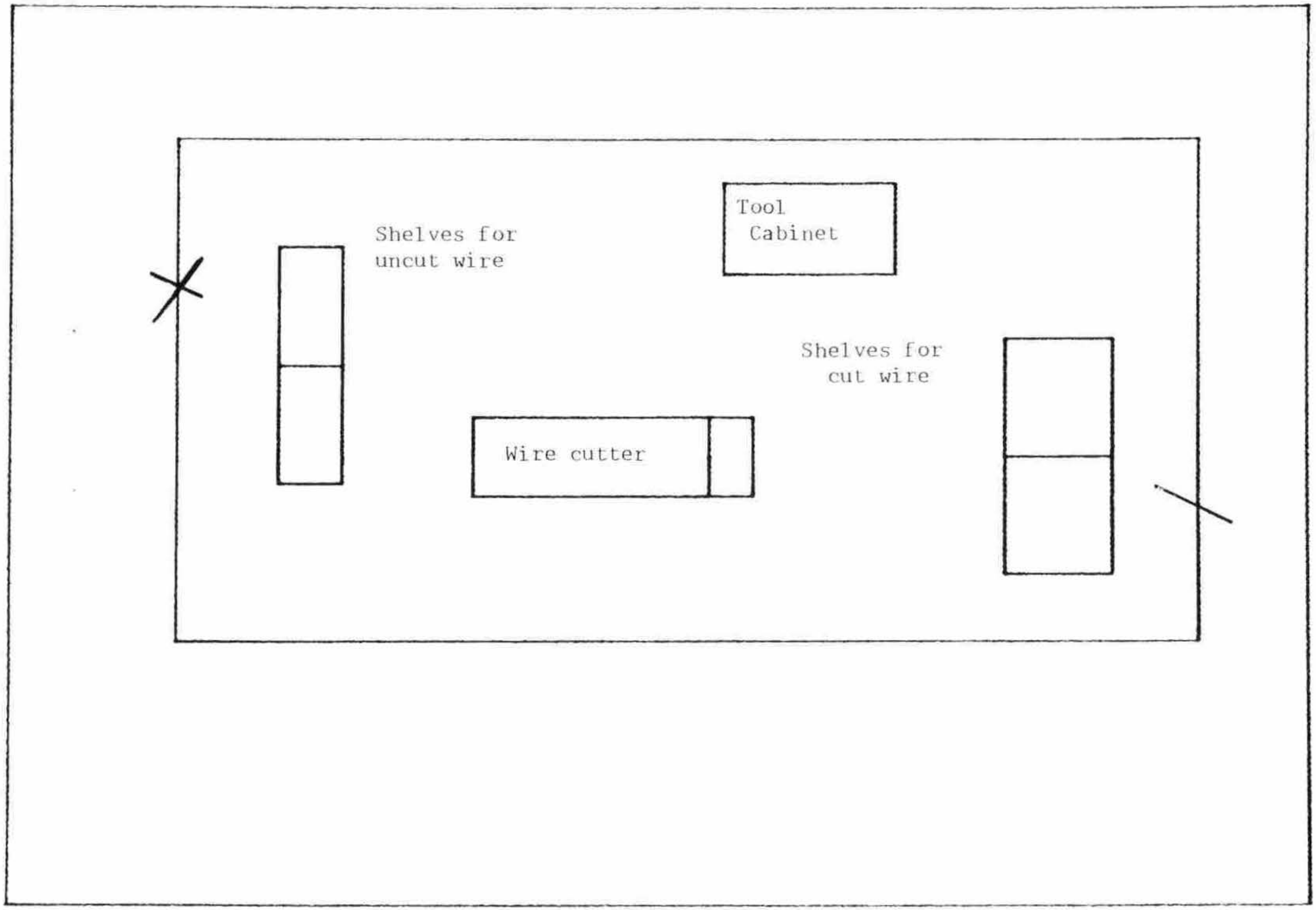


Figure 6.19. Wire cutter (scale 1)

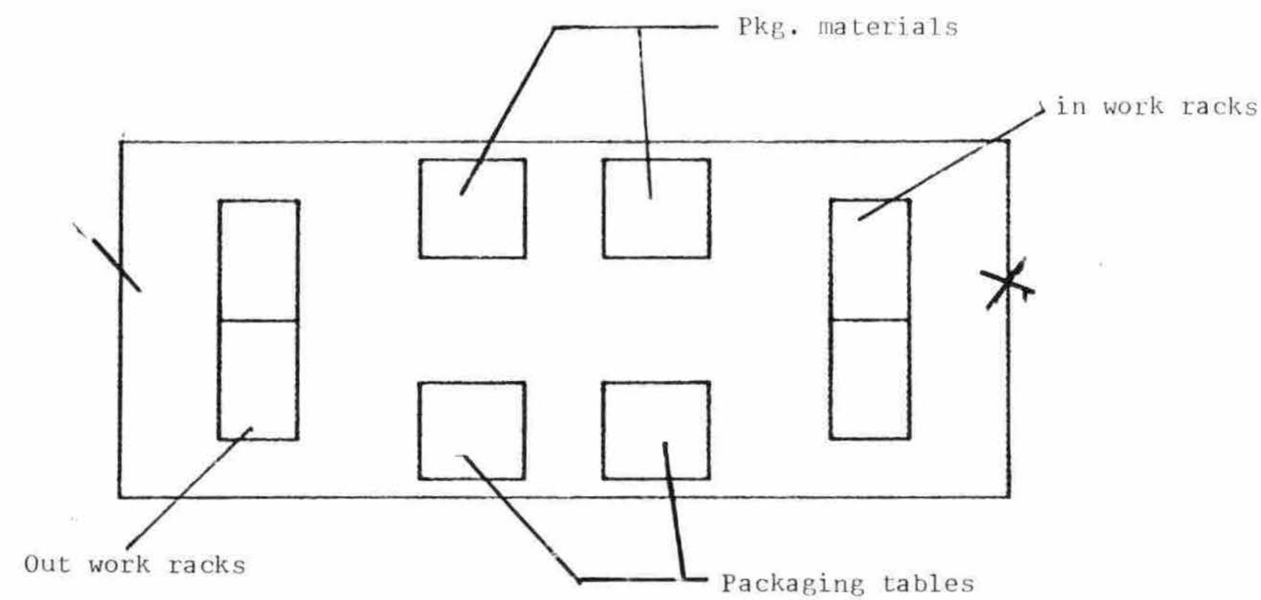


Figure 6.20. Packaging (scale 2)

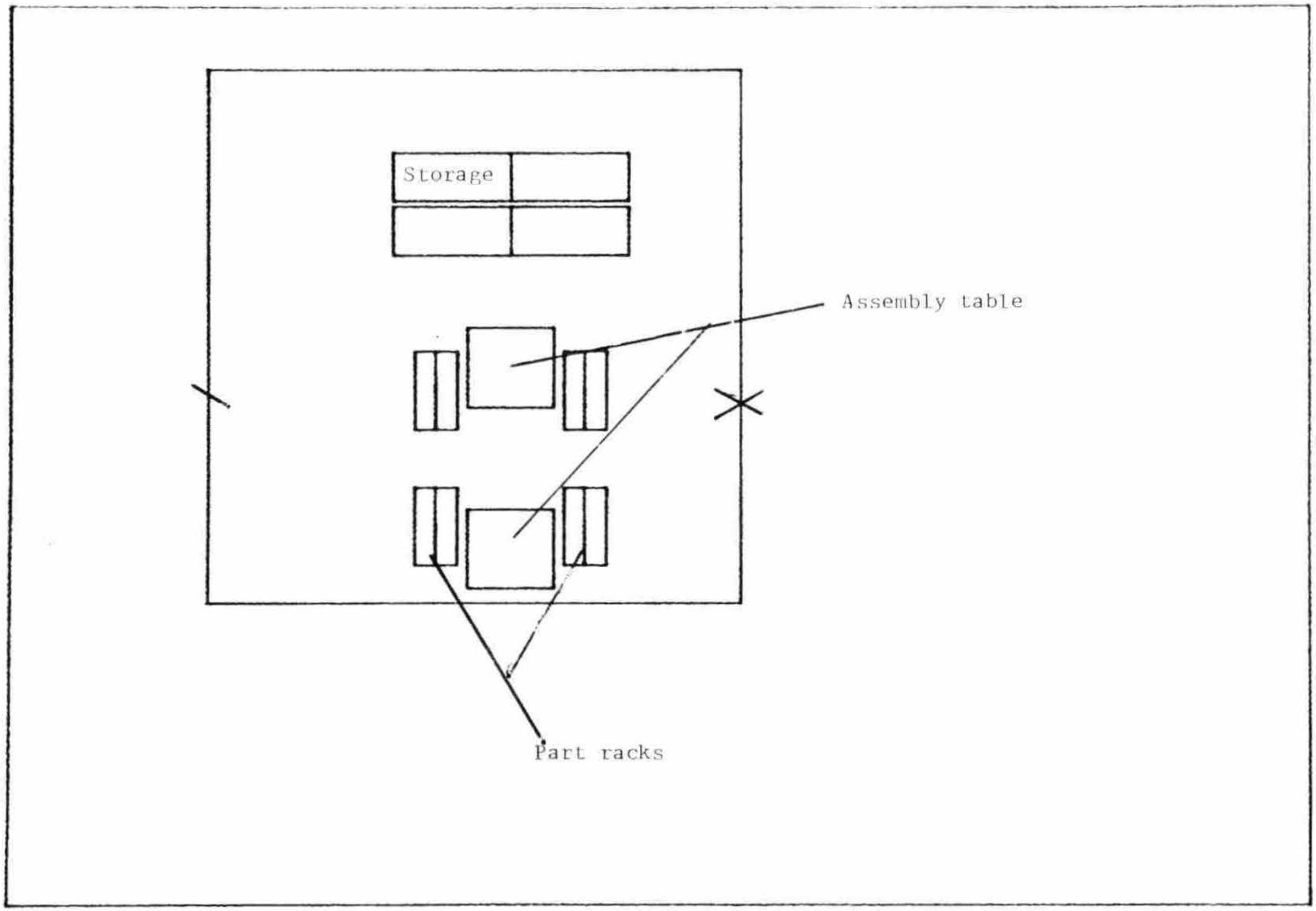


Figure 6.21. Assembly (scale 2)

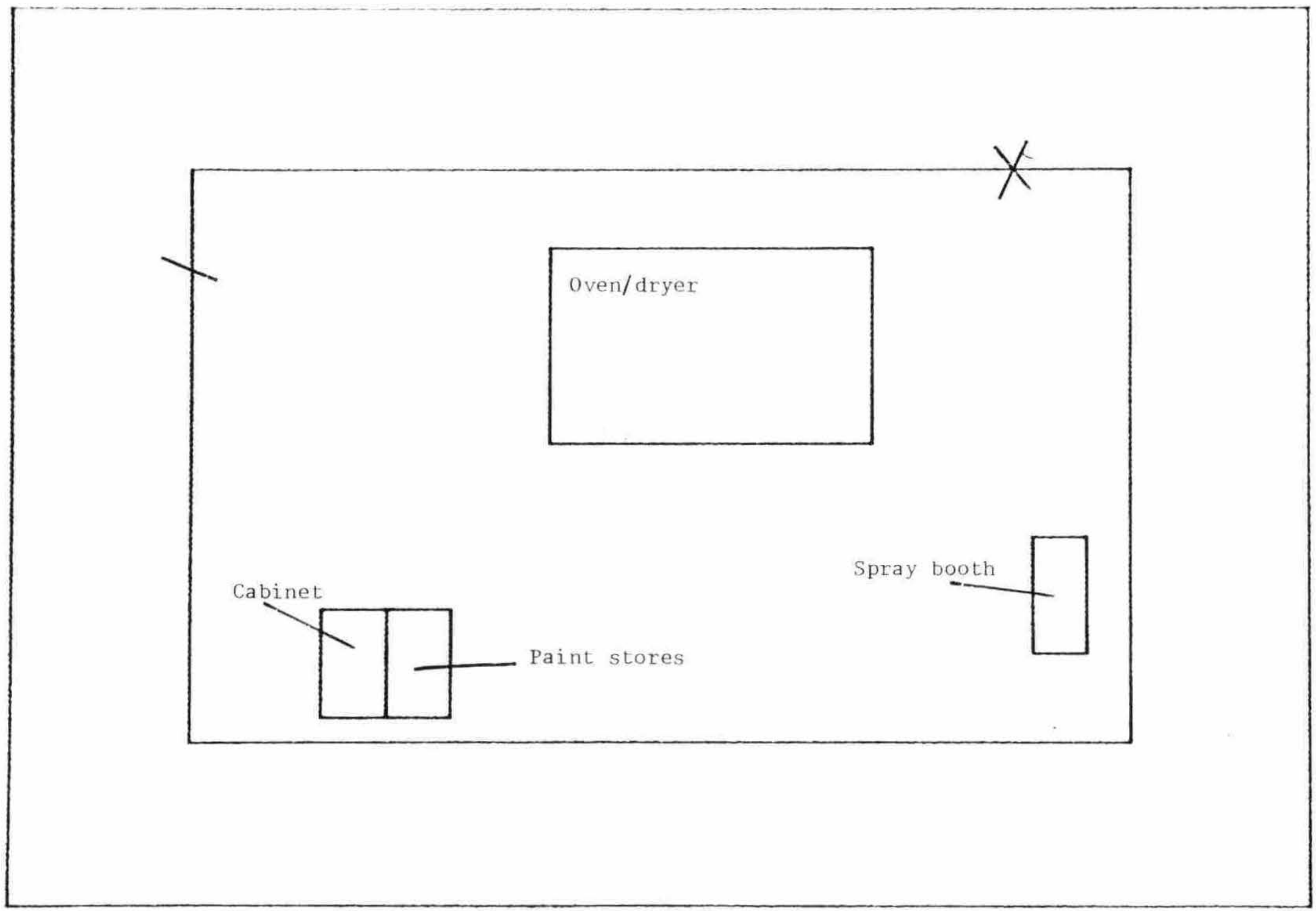


Figure 6.22. Painting (scale 2)

Table 6.1. Workcenter areas

Workcenter Number	Workcenter Name	Width (ft)	Length (ft)	Area (sq ft)
1	Shipping/ Receiving	34.2	62.1	2183.82
2	Decoiler/ Shear	15.3	20.7	316.71
3	Punch Press A	13.95	16.2	218.7
4	Punch Press B	14.5	16.57	240.26
5	Roller	11.52	26.1	300.672
6	Flanger	9.36	22.95	214.81
7	Seam Welder	15.75	16.02	252.32
8	Spot Welder	14.4	15.3	220.32
9	Wire Cutter	10.4	22.95	238.68
10	Packaging	11.16	27.9	311.36
11	Assembly	20.25	20.25	410.06
12	Painting	21.6	36	777.6

center of the drawing frame. Workcenters maintain their relative positions on the screen as the scale is increased. To facilitate the layout process, it is desirable to start at the Southwest corner of the drawing frame. Workcenters are placed first when they require access to an outside wall (S). Shipping/receiving is normally a department requiring such access and is usually placed first.

Placement of subsequent workcenters will be ranked by FLAG. Selected workcenters will initially placed to minimize move costs. The user will be required to perform subsequent placement adjustments during the layout process. Such adjustments will require moving workcenters that are partially superimposed upon one another. Additional adjustments are required as the user lays out the workcenters outward from the Southwest corner of the drawing frame. It is necessary to start in this corner since subsequent scale increases will not move workcenters to the drawing frame's center. This makes it difficult to position subsequent workcenters in the Southwest section of the drawing frame.

In this example, Shipping/receiving is placed first. The initial placement by FLAG is shown in Figure 6.23. The cross represents the POE, and the slash represents the POD.

LAYOUT next ranks all flow relationships between all workcenters placed (i.e., Shipping/receiving) and those departments yet to be placed. This ranking (see Figure 6.24) establishes packaging as the best choice to be placed. The user could override the program's choice, but in this case, does not.

The initial placement of the packaging workcenter is shown in

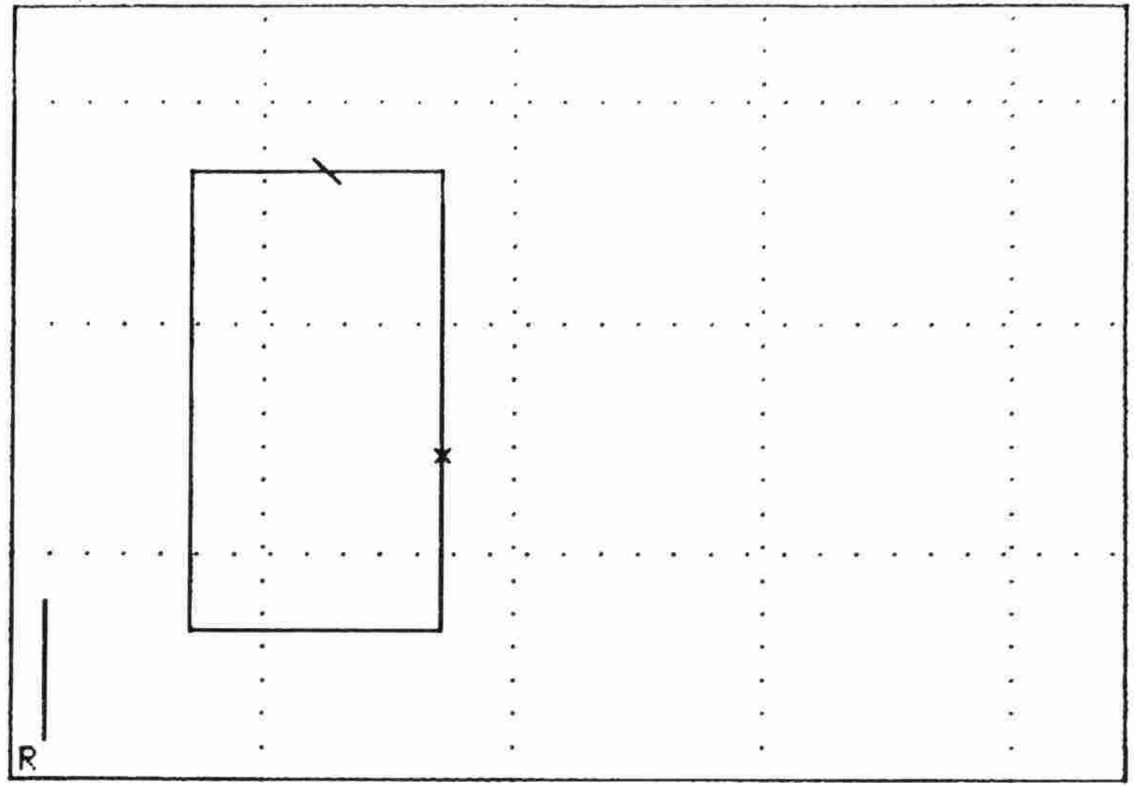
AXIS RANGES:

X AXIS:
0.0 TO 780.0
Y AXIS:
0.0 TO 600.0

DOTTED LINE
INCREMENTS 175.0

SCALE NO.: 6

R LINE = 25 FT.
270.0000



SELECT OPTION AND PRESS <RET>:

1. ACCEPT PLACEMENT
2. INCREASE FRAME AREA

Figure 6.23. Initial placement of shipping/receiving

```
RONKO MFG. CO.                                              FLAG
PHASE: LAYOUT

                     RANKING OF UNPLACED WORKCENTERS

    1      10    1     PACKAGING              2248.700

    2       1    2     DECOILER/SHEAR          198.000
    3       1   11     ASSEMBLY                181.500
    4       1   10     PACKAGING                 9.300
    5       1    9     WIRE CUTTER               9.250

FLAG HAS SELECTED THE FIRST-RANKED DEPARTMENT
FOR PLACEMENT IN THE LAYOUT. DO YOU ACCEPT THIS
SELECTION?

                1. YES
                2. NO
1
```

Figure 6.24. Ranking of unplaced workcenters

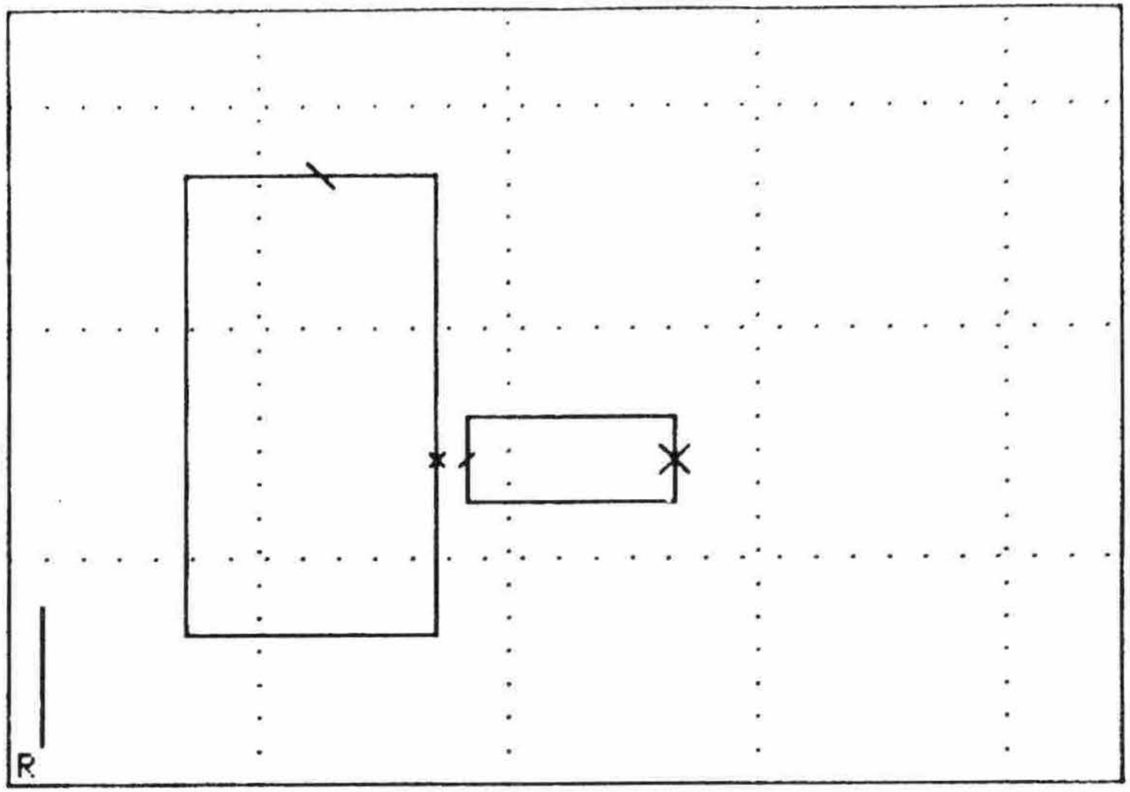
AXIS RANGES:

X AXIS:
0.0 TO 780.0
Y AXIS:
0.0 TO 600.0

DOTTED LINE
INCREMENTS 175.0

SCALE NO.: 6

R LINE = 25 FT.



SELECT OPTION AND PRESS RETURN

1. SHIFT	5. RANKING
2. ROTATE	6. LAYOUT
3. POINTS	7. INCREASE AREA
4. LIST	8. ACCEPT

Figure 6.25. Initial placement of packaging

Figure 6.25. Note that the packaging POD was placed, by LAYOUT, directly across the POE of shipping/receiving, separated by the default aisle length of five feet. The user, has made the judgment that this aisle is unnecessary opts to adjust the workcenter to the west by five feet. This completed adjustment is illustrated in Figure 6.26.

This process is completed for the next 10 workcenters. After accepting placement of the 12th workcenter, the LAYOUT program establishes initial walls. These walls can be adjusted in the same way as borders of workcenters in the WORKOUT routine.

The final result of this process is the actual facility layout, produced by OUTPUT and illustrated in Figure 6.27. This result represents a good layout from the stand point of flow patterns of flow patterns (discussed in the following chapter). However, a better layout can be obtained by bringing down the uppermost workcenters into the open spaces in the middle. Therefore, LAYOUT was run again to generate the improved layout shown in Figure 6.28. Measurements illustrated the improved layout represents a reduction from 7715 square feet to 6052 square feet. This improvement is a representation of the benefits of user interaction with FLAG.

This improved version of the FLAG layout for the RMC facility (Figure 6.28) is analyzed in chapter VII. Similar layouts generated by the OPDEP and PLANET plant layout programs provide a basis of comparison for the three facilities design programs.

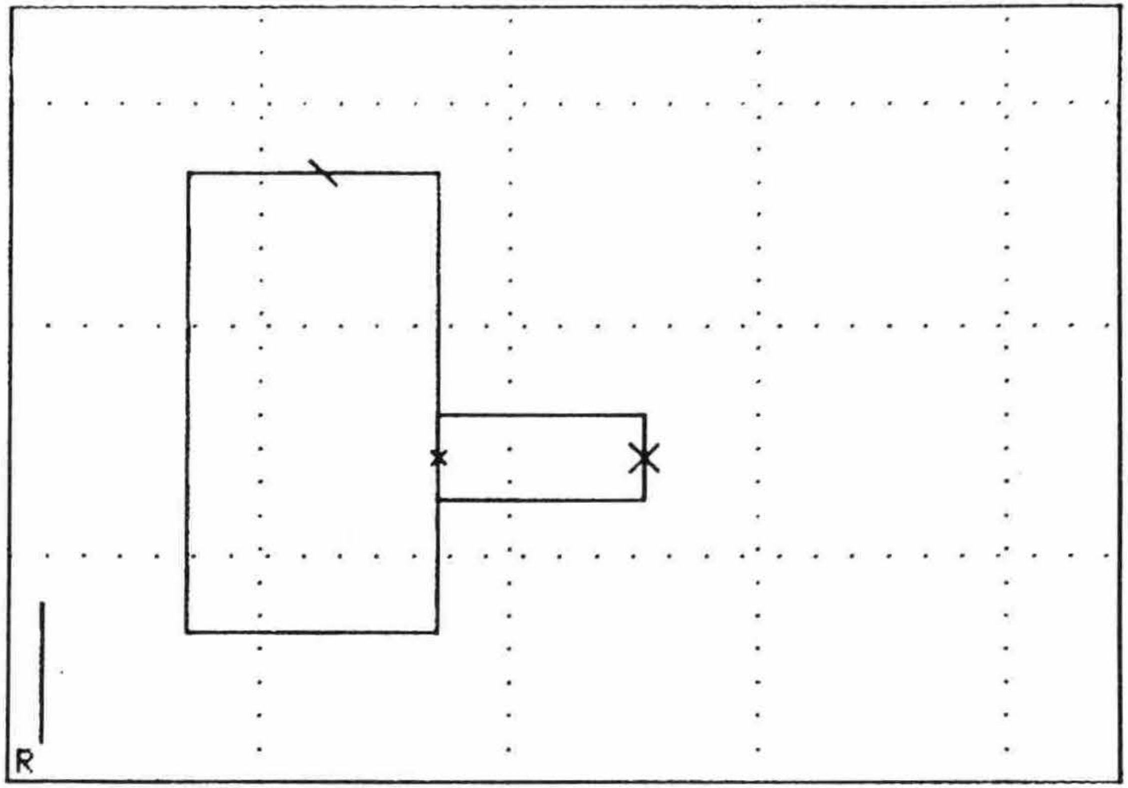
AXIS RANGES:

X AXIS:
0.0 TO 780.0
Y AXIS:
0.0 TO 600.0

DOTTED LINE
INCREMENTS 175.0

SCALE NO.: 6

R LINE = 25 FT.



SELECT OPTION AND PRESS RETURN

1. SHIFT
2. ROTATE
3. POINTS
4. LIST
5. RANKING
6. LAYOUT
7. INCREASE AREA
8. ACCEPT

Figure 6.26. Adjusted placement of packaging

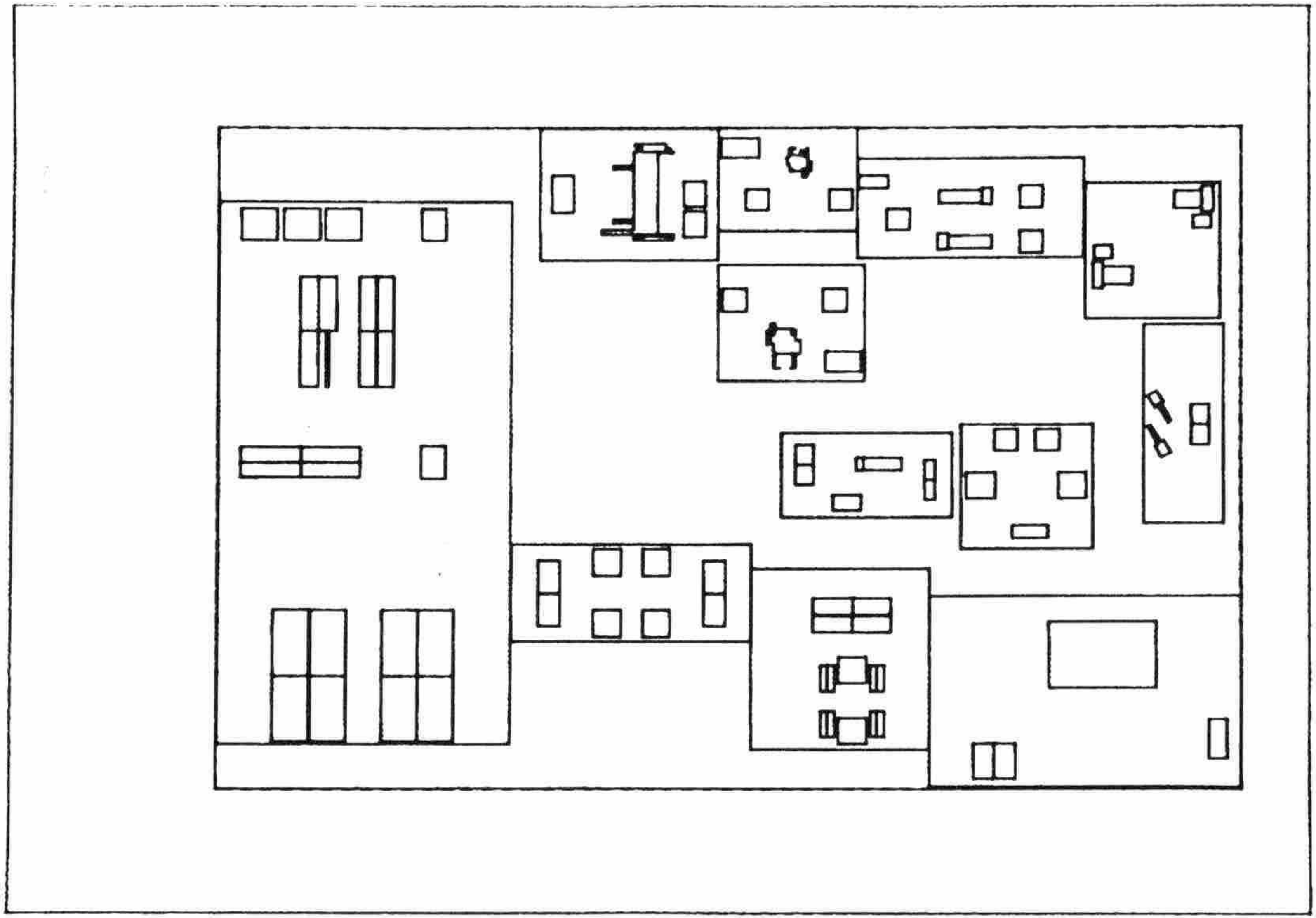


Figure 6.27. Ronko Manufacturing Company FLAG layout (Scale 6)

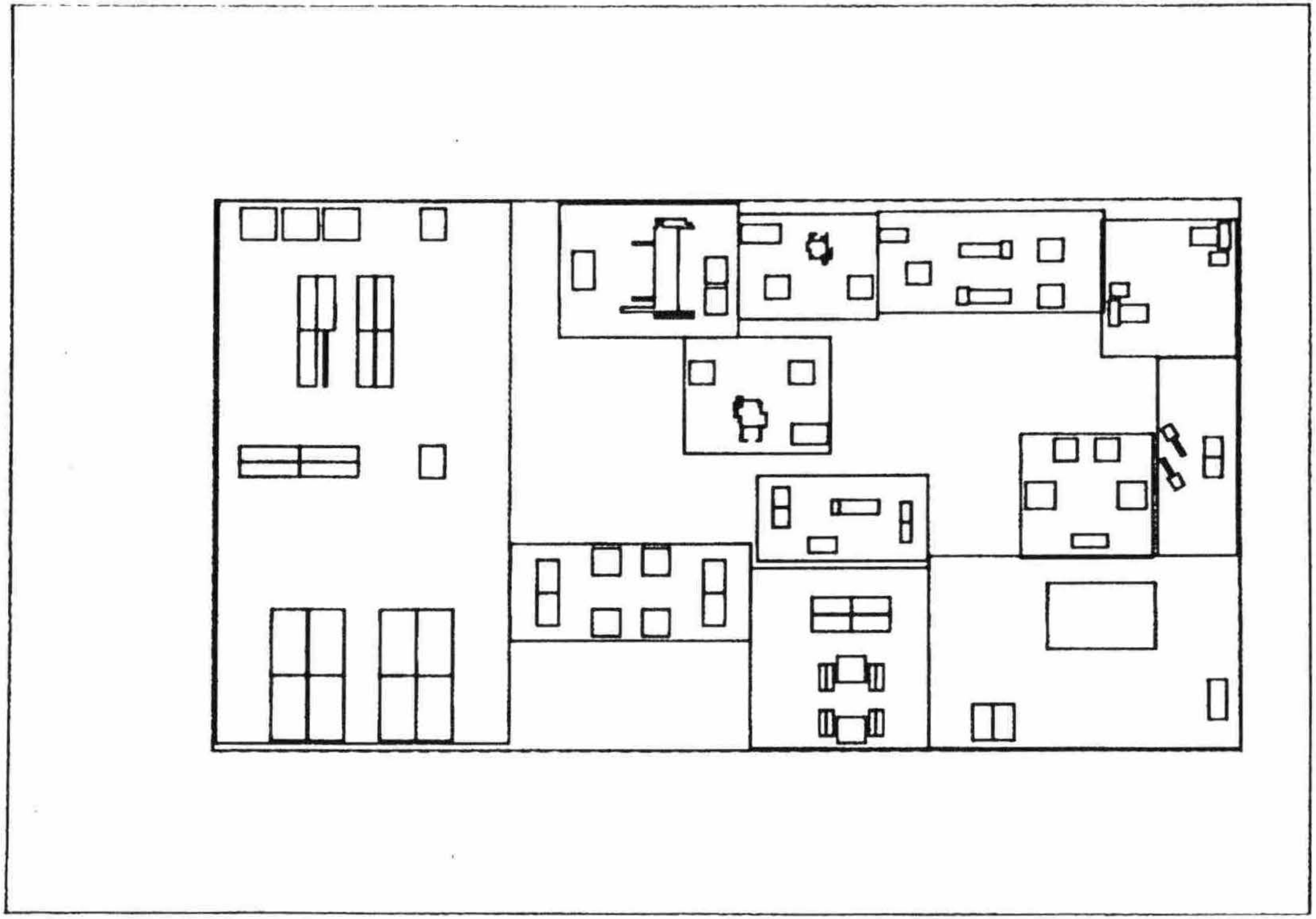


Figure 6.28. Ronko Manufacturing Company FLAY layout - improved version (scale 6)

VII. COMPARATIVE ANALYSIS

A. Introduction

This chapter describes a comparative analysis between the FLAG, OPDEP, and PLANET facility layout programs. Data for the Ronko Manufacturing Company, described in chapter VI, were used to execute all programs. The FLAG output from this exercise was presented in chapter VI. Copies of the results from OPDEP and PLANET are presented in Appendix II.

The comparison will be made through the entire process on the basis of utilizing a computer program for facilities design. The comparison considers all steps from the preparation required before executing each program to the analysis and utility of the resulting output obtained in each case.

It is difficult to directly compare FLAG with PLANET and OPDEP. FLAG has distinct features that distinguish it from other layout programs. OPDEP and PLANET claim only to specify the relative positions of workcenters in relation to each other. FLAG gives the user a preliminary layout. However, by making a few minor assumptions, some useful comparisons can be made regarding the utility of each layout developed.

B. Preparation

Each program requires some preparation in terms of collecting and formulating the necessary data. This must be accomplished before the actual execution of the program. The preparation required for each program is summarized below:

1. FLAG - Requires collection of data in all interdepartmental flows for each product manufactured. This includes move-cost/foot, quantity in product, and units per move.
   - Requires establishment of a product mix.
   - Requires information about workcenter in terms of machines, cabinets, racks, etc.
2. OPDEP - Requires tabulation of a from-to chart relating moves between workcenters.
   - Establishment of a product mix is implicit in the tabulation of the from-to chart.
   - Requires estimates of workcenter sizes.
3. PLANET - Requires collection of all interdepartmental flows, including frequency of moves and moves cost per foot.
   - Establishment of product mix is implicit in the input of interdepartmental hours.
   - Requires estimates of workcenter sizes.

When the facility manufactures only one product, there is little difference in the amount of preparation required by the three programs. OPDEP requires the user to tabulate the from-to chart from interdepartmental flow data, while both FLAG and PLANET do this tabulation for the user. FLAG requires more specific knowledge about workcenter components. However, some of this information must be conceptualized for the workcenter area estimates required by OPDEP and PLANET.

A significant difference does arise when different product mixes, i.e. production levels for each product, are considered by the user. The FLAG user need only specify the new product mix to the FLOWS program. This is because the initial input of interdepartmental flow data was independent of quantity by product type. In OPDEP, the product mix is implicit in the tabulation of the from-to chart. The user will therefore, be required to tabulate a new from-to chart if the product mix changes. PLANET interdepartmental flow data are dependent on product quantities. Consequently, these data will also have to be reformulated and re-entered into the program if the product mix changes.

C. Program Execution

There can be no question that FLAG requires more user effort during the program execution period. The twelve workcenter, Ronko Manufacturing example required approximately seven hours of terminal time to develop the layout from FLOWS to OUTPUT. The combined time for running both PLANET and OPDEP was approximately three and one-half hours. This time might have been lower if the writer was as familiar with the operation of OPDEP and PLANET as with FLAG.

There are two primary reasons for the difference in program execution times. First of all, PLANET and OPDEP are primarily batch mode programs. After the input of initial data files, there is no user contact with the program.

FLAG is an interactive time-sharing mode program. The user is continually asked to participate in the decisionmaking process of the program. Also, FLAG requires a larger quantity of input data. The

8	8	12	12	12	12	1	1	1	1		
8	8	12	6	12	12	1	1	1	1		
4	4	6	6	12	12	1	1	1	1	9	9
4	4	6	7	12	12	1	1	1	1	9	9
2	2	7	7	12	12	1	1	1	1	1	1
2	2	7	7	11	11	1	1	1	1	1	1
2	2	5	5	11	11	10	10	1	1	1	1
3	3	5	5	11	11	10	10	1	1	1	1
3	3	5	5	11	11	10	10	1	1	1	1

Score = 880

Figure 7.1. Best scoring OPDEP layout

best example of this is in the internal design of workcenters. FLAG is the only program of the three which addresses the internal workcenter design process.

D. Output

1. FLAG

The FLAG final output for the Ronko Manufacturing Company is illustrated in Figure 6.28. The layout depicts all workcenters with their internal elements. The final output also details external walls and all unassigned spaces.

2. OPDEP

The OPDEP output prints all layouts that score above a specified level. The complete OPDEP printout is presented in Appendix II. The highest scoring layout has been selected and is presented in Figure 7.1. The output is in the form of rectangular grid blocks. No internal details of workcenters are presented. Workcenter shapes random and irregular. Workcenters are represented by their number, and each grid block represents fifty square feet of floor area.

One point should be made regarding the layout scoring. The high scoring layout for sweep lengths 2, 3, and 4 was 978. However, the same printout lists the theoretical perfect score as 880. To resolve this discrepancy, the OPDEP high scoring layout was scored manually using the OPDEP scoring procedure. The actual score was 880. The OPDEP scoring method will be described in detail in the next section. For the analysis in this chapter, the 880 score is considered accurate. The erroneous score is attributed to an error in the OPDEP program.

							09	09											
							09	09											
		10	10	10	11	11	11	11											
01	01	10	10	10	11	11	11	11											
01	01	01	01	01	01	12	12	12											
01	01	01	01	01	01	12	12	12		06		07	05	06					
01	01	01	01	01	01	12	12	12	06	06	07	07	05	05	03	03		08	08
01	01	01	C1	01	01	12	12	12	12	06	07	07	05	05	03	03	02	08	08
01	01	01	01	01	01	01									02	02	02	04	04
01	01	01	01	01	01	01										02	02	04	04

Score = 28

Figure 7.2. Best scoring PLANET layout

This value only implies a perfect score according to the OPDEP scoring procedure, and not necessarily a great, or even a good, layout.

3. PLANET

The PLANET output is essentially the same as the OPDEP output in nature. The best PLANET layout is presented in Figure 7.2. The complete PLANET output is presented in Appendix II. The one significant difference between the PLANET output and the OPDEP output is the shape of the PLANET layout. While the OPDEP layout is rectangular, the PLANET layout takes the shape of an irregular polygon.

E. OPDEP Scoring

1. Introduction

Each of the three layouts discussed in this chapter was scored by two different methods. The first method is the OPDEP scoring method. This method rewards layouts for placing workcenters close together if they have a high frequency of moves between them. The second method might be considered the FLAG scoring method since it rewards layouts which minimize total move-cost. This is an implicit criterion of the FLAG layout discipline.

2. Numerical value assignments

The OPDEP scoring mechanism is based on the Muther's six-step priority scale and on assignment of values to the A,E,I,O,U, and X ratings. Such assignments are based on the relative frequency of moves between workcenters. The move frequencies for the Ronko Manufacturing Company from-to chart are presented in Table 7.1. The move frequencies are ranked in descending order in Table 7.2. Large values are assigned

Table 7.1. Frequency of moves between workcenters

	1	2	3	4	5	6	7	8	9	10	11	12
1 S/R		80							33	30	1453	
2 D/S			375	1974								
3 PPA					375							
4 PPB								1654				320
5 Rllr							2501					
6 Phl												2501
7 SmW						2501						
8 SpW												400
9 WC											96	
10 Pkg	6525											
11 Assem											4318	
12 Pzg												3221

Table 7.2. Ranked frequency of moves between workcenters

From	To	Frequency of Moves	Muther's Rating
10	1	6525	A
11	10	4318	A
12	11	3221	E
7	6	2501	E
6	12	2501	E
5	7	2501	E
2	4	1974	I
4	8	1654	I
1	11	1453	I
8	12	400	O
3	5	375	O
4	12	320	O
9	11	96	U
1	2	80	U
1	9	33	U
1	10	30	U

an 'A' rating. Workcenters given 'A' ratings are given first priority for placement. Lower frequencies are assigned other letter codes as indicated in Table 7.2. While the breakpoints for these ratings are somewhat arbitrary, numerical gaps were observed and used to make reasonable assignments of ratings. The breakpoints for the ratings are summarized in Table 7.3.

The OPDEP user must next assign a numerical value to each rating. These assigned values are again arbitrary. However, they are structured to be relatively proportional to the frequency of move breakpoints. (See Table 7.3.)

3. Scoring procedure

These ratings and rating values are now used to score the layouts. To illustrate the method described by Nelson [17], an example, shown in Figure 7.3 is used. For simplicity workcenter 01 has an 'A' relationship with workcenters 06, 11, and 12. The 'A' rating value is assumed to be 90. Ninety points are therefore added to the total score for each relationship that is 'fully' satisfied. Relationships are assumed to be fully satisfied if any part of the workcenter perimeters touch each other. Workcenters 12 and 01 meet this requirement. Half of the score, 45 points, is given for each relationship that is 'half' satisfied. "Half-satisfied" relationships exist when the workcenters are separated by one grid block and are grid block only. Workcenters 01 and 11 meet this requirement. Workcenters separated by more than one grid block are given no points.

Each workcenter is scored in this manner, and the sum of these

Table 7.3. Breakdown of OPDEP ratings with assigned scoring values

Rating	Frequency of Moves	Full values	Half values
A	fm $\geq$ 4000	90	45
E	4000 > fm $\geq$ 2000	45	22.5
I	2000 > fm $\geq$ 1000	20	10
O	1000 > fm $\geq$ 250	5	2.5
U	250 > fm $\geq$ 0	0	0
X	not used in this example	0	0

scores represents the total layout score. By scoring each workcenter in this manner, it should be noted that each flow is created twice in the OPDEP scoring method.

	01	01	
	01	01	
12	12	12	
12	12	11	11
	06	06	11
	06	06	

01 - 12	full score	90 points
01 - 11	half score	45 points
01 - 06	no score	0 points

Figure 7.3. Scoring procedure for workcenter

F. Obtained Layout Scores by the OPDEP Method

1. OPDEP layout score

A summary of scores for each of the three layouts by the OPDEP method is presented in Table 7.4. The best OPDEP layout got a theoretically perfect score of 880. This was due in part to the fact that workcenter shapes are assumed to be flexible and can thus be fitted into odd positions.

To illustrate this point, the best OPDEP layout is reproduced in Figure 7.4. The clearest example of the benefit of variable workcenter shapes is between workcenters 4 (Punch press B) and 12 (painting). The flows between these two workcenters were given an 'O' rating worth 5 points if fully satisfied. The relationship is fully satisfied, but only because of the irregular shape of the painting workcenter. This shape is not realistic to the internal requirements of the painting workcenter, but is allowed by OPDEP.

Table 7.4. Summary of layout scores by OPDEP method

	Score	% of Perfect Score
OPDEP	880	100
PLANET	860	97.7
FLAG	790	89.8

Also assisting OPDEP in achieving a perfect score, is the zero value the user arbitrarily assigned to the 'U' rating. For example, consider the flow of cut wires from the wire cutting workcenter to

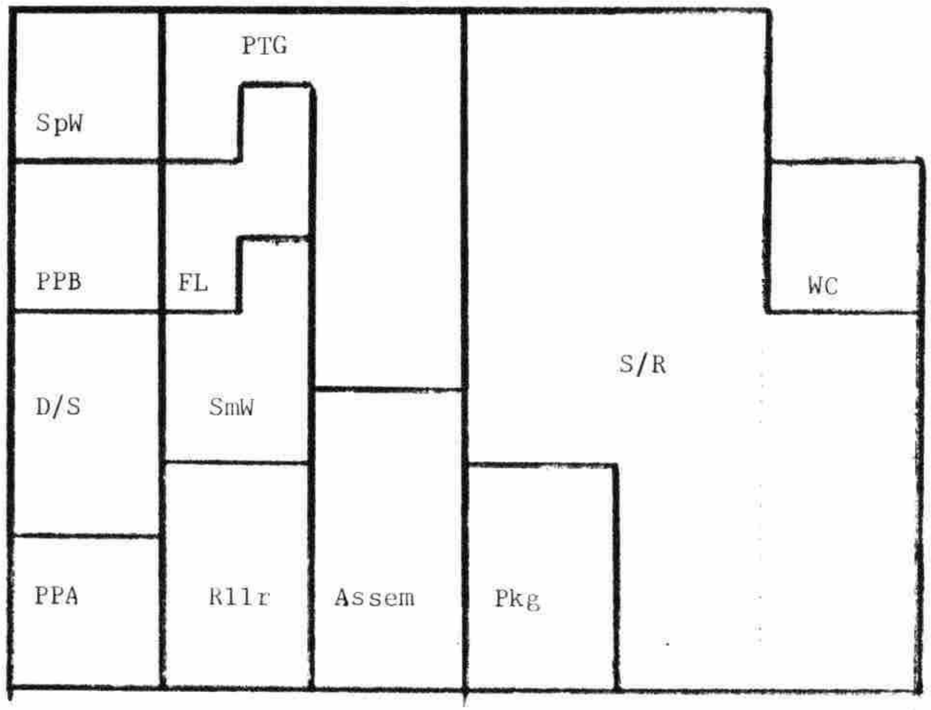


Figure 7.4. OPDEP layout

assembly. This flow occurs 96 fimes per unit time (see Table 7.1). Yet, in relation to some of the other flows, this number is insignificant. Therefore, a 'U' rating was given to this relationship, and to all others with less than 250 moves per unit of time.

Since 'U' was given a zero value, it does not enter into the scoring. If the 'U' rating was given a value of one, the theoretical perfect would be increased by one point for each relationship assigned a 'U' rating. Therefore, the satisfaction or non-satisfaction of the relationships now become relevant in terms of the actual layout score to the theoretical perfect score. In the present OPDEP example, the layout would then fall short of a perfect score since the relationship between assembly and wire cutting is not satisfied.

2. PLANET layout score

The PLANET layout also fared well by OPDEP scoring method, obtaining 97.7% of a perfect score. The reasons discussed above for OPDEP also assisted the PLANET layout.

The PLANET layout is shown in Figure 7.5. As can be seen from this illustration, the layout has an irregular external perimeter. Since the program is out bound by rectangular form constants, the layout had more alternatives by which a high numerical score could be obtained.

The PLANET layout is numerically enhanced by the fact that move-costs are ignored by the OPDEP scoring method. A move of bulky and heavy material is considered equal to that of small lightweight matierial. Therefore, the separation the decoil/shear and receiving by a large distance (see Figure 7.5) has no detrimental effects. This

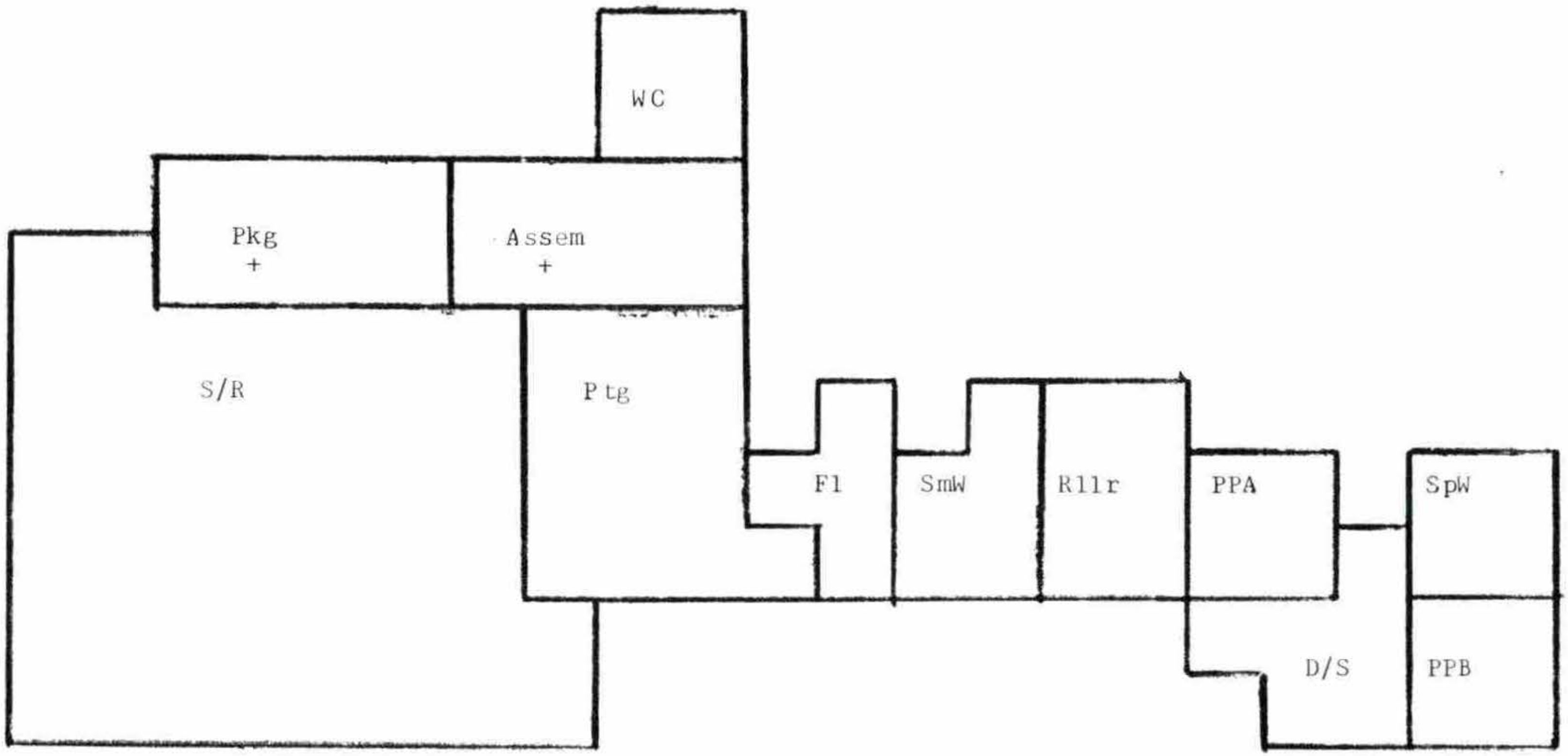


Figure 7.5. Representation of PLANET layout

is because the number of moves falls under a 'U' rating, despite the fact that steel coils are one of the most expensive items to move.

3. FLAG layout score

The OPDEP scoring procedure cannot be directly applied to the FLAG output. This is because the FLAG layout is not in the form of grid blocks. To apply the OPDEP scoring procedure to FLAG, the following procedure was adopted to translate the FLAG output into an equivalent management of grid blocks:

1. A grid of nine rows and twelve columns (the same number as the OPDEP layout) was drawn over the FLAG layout.
2. Each grid block was assigned to the appropriate workcenter or marked as being empty.
3. Each grid block was assigned to the workcenter occupying the most physical space in the grid block.
4. If no workcenter fell into the grid block it was marked empty.

The resultant layout from the above procedure is shown in Figure 7.6. The shaded areas represent open spaces. The OPDEP scoring method can now be applied to the FLAG output.

Although the FLAG program did not do as well under OPDEP scoring as the other two programs, it still received a respectable 89.8% of a perfect score. The pressing reason for this third place performance is that FLAG workcenters have fixed shapes. These shapes were not changed to fit into smaller irregular spaces. Good design judgment dictated final workcenter shapes and placement. For example, suppose

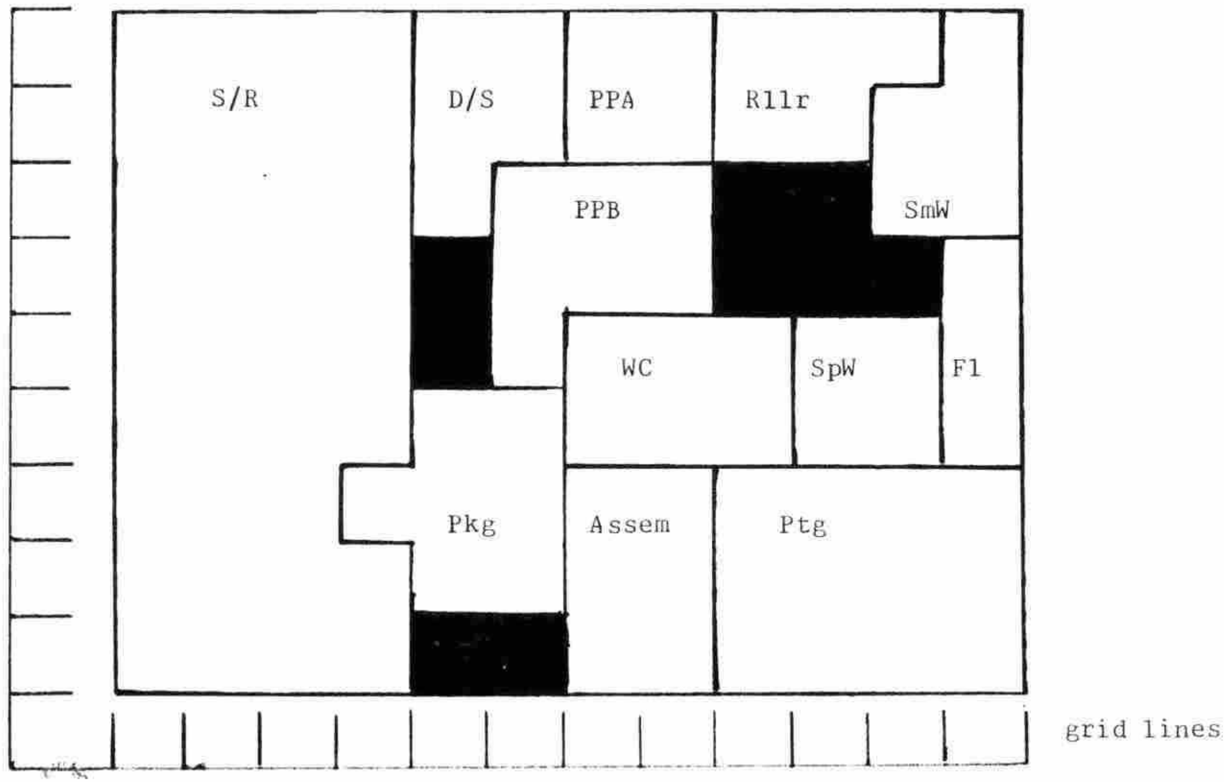


Figure 7.6. FLAG grid block layout

the spot welding workcenter could be configured like an "L" instead of a square. If so, it could be placed around the wire cutting workcenter to correct with punch press B. This could be accomplished without losing contact with the painting workcenter. This would result in full satisfaction of both proximity relationships. Instead, due to its fixed shape, only one relationship was satisfied. This fact prevented a higher score from being obtained under the OPDEP scoring procedure.

G. Material Flow Comparison

There is another point of discussion regarding the three layouts. Only FLAG recognizes shipping/receiving as having two geographically distinct functions. That is, receiving occupies the north half of the workcenter, while shipping occupies the lower half. OPDEP and PLANET do not make this distinction. FLAG does consider this feature through points of entry and departure. Using the OPDEP scoring method, no penalties were assessed from failing to correct either shipping or receiving with appropriate adjacent workcenters. This feature of FLAG yields a logical material flow pattern.

Materials flow out of the north part (receiving) of S/R. Material generally travels in a U or V flow pattern to return to shipping. (See Figure 7.7.) The corresponding flow pattern for the PLANET layout appears in Figure 7.8. The material flows demonstrated here are less structured and assume many parallel paths. The corresponding flow paths in the OPDEP layout are illustrated in Figure 7.9.

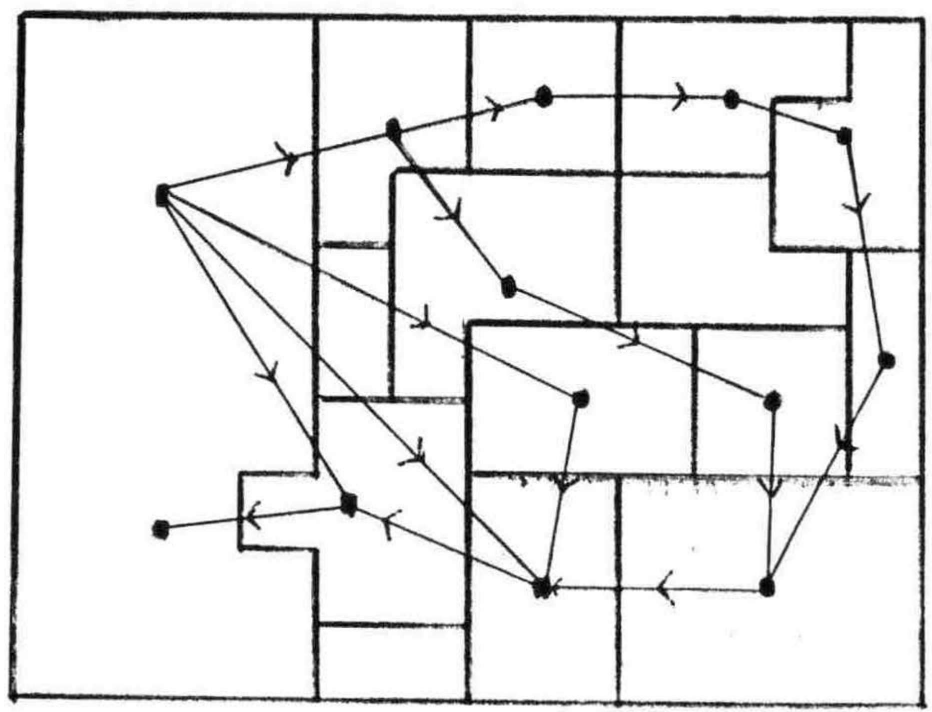


Figure 7.7. Flow through FLAG layout

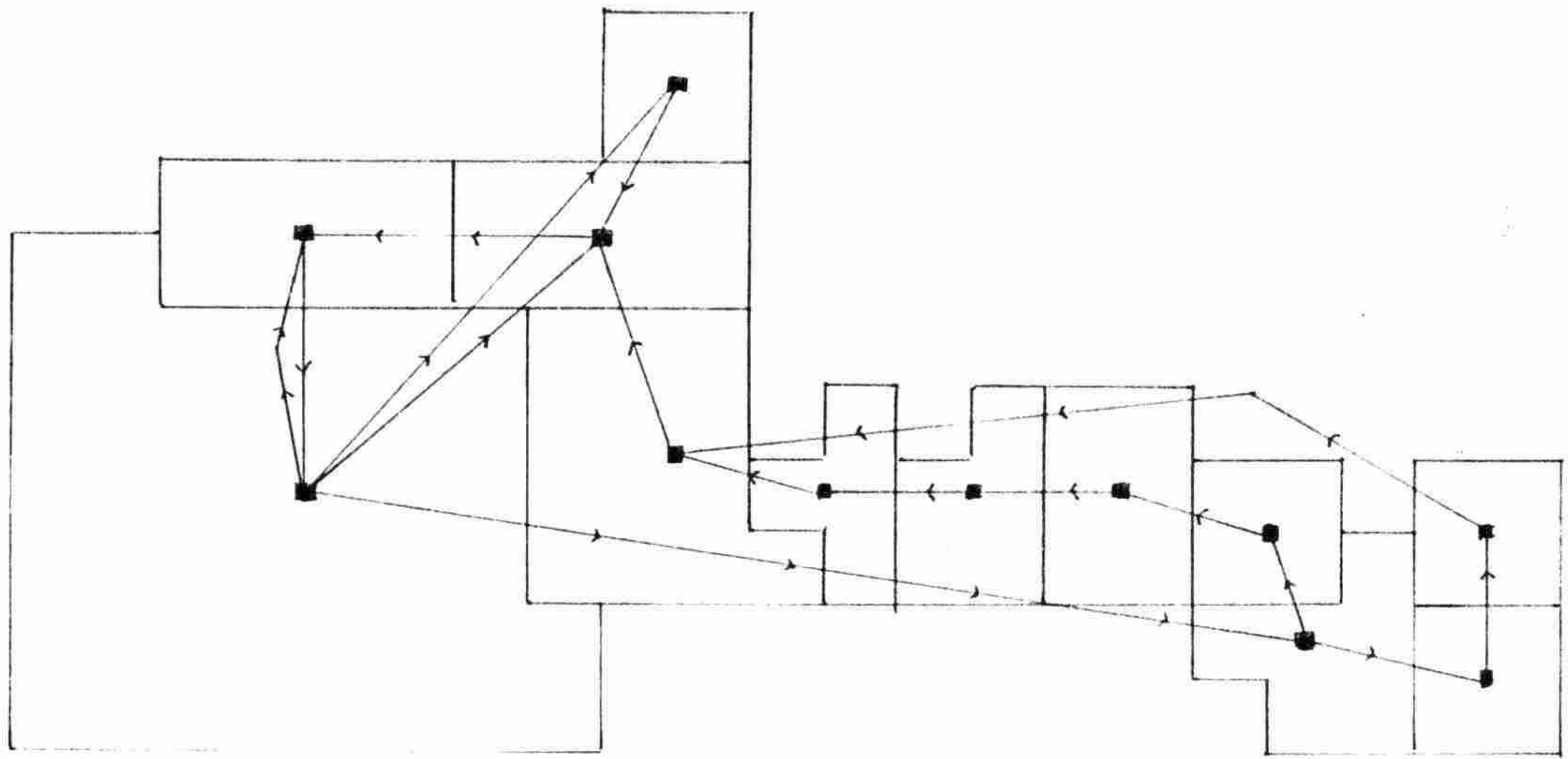


Figure 7.8. Flow through PLANET layout

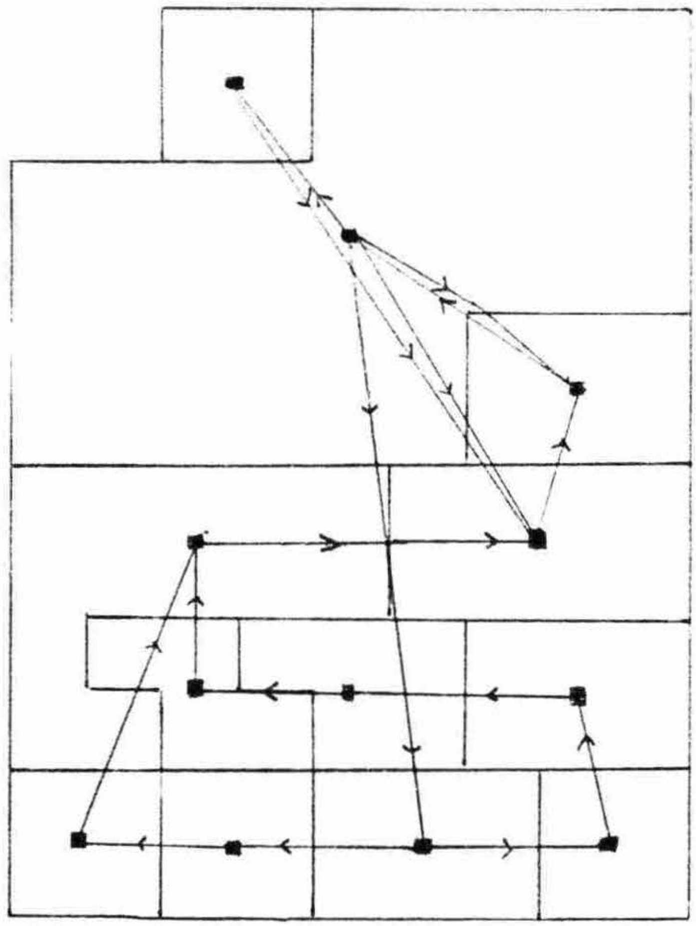


Figure 7.9. Flows through OPDEP layout

H. Scoring with Move Cost

1. Introduction

Scoring with move-cost is a simpler and better method of scoring layouts since it recognizes that various material flows have different move costs. The resultant score using this method is a total materials handling cost, which can be traced to the individual layout. Since the scores are costs, lower scores indicate better layouts.

To illustrate this scoring method, assume there are N flows between workcenters. This method scores each flow only once. The formula for material handling costs becomes:

$$\text{M.H.C.} = \sum_{i=1}^{n} \sum_{j=1}^{m} f_{jab} D_{ab} MC/F_{jab} \qquad (7.1)$$

m - number of different type flows from workcenter a to workcenter b.

f_{ab} - frequency of move for flow j from workcenter a to workcenter b.

D_{jab} - distance between centroids of workcenter a and workcenter b in feet.

MC/F_{jab} - move cost per foot for flow j from workcenter a to workcenter b.

FLAG measures the distance between workcenters as the distance between the POEs and PODs. OPDEP and PLANET use centroids of workcenters to define the parameter. Since neither PLANET and OPDEP have POEs and PODs, centroids measurement was applied to all three layouts for scoring purposes.

A summary of layout scores by move-cost method is presented in Table 7.5.

Table 7.5. Scores of layouts using move costs

Program	Material Handling Cost
OPDEP	159,049.49
FLAG	160,759.74
PLANET	175,275.78

2. OPDEP layout score

OPDEP achieved the highest score, but not by a significant amount. (See Table 7.5.) Again, the fact that OPDEP was able to mold workcenters into small, irregular area shapes helped the program achieve this relatively low total move cost.

3. FLAG layout score

The FLAG score was slightly higher than the OPDEP score. The FLAG layout score was increased by the need in some cases for material to travel open, unused, areas that do not exist in either of the other two layouts. This illustrates the disadvantage of realism by the fixed workcenter shapes, resulting from internal requirements.

4. PLANET layout score

The PLANET layout scored badly despite the fact that the PLANET algorithm takes into account move cost factors. The scoring problem results partially from the program's feature to confine the overall layout to a rectangular area. PLANET lays out one workcenter at a time allowing the layout to explode outward. This promotes large distances between workcenters. For example, large distances exist between receiving and the decoiled shear. Coiled steel must traverse this distance at high move cost per foot. (See Figure 7.5.)

I. Post Program Tasks

There exists a variety of tasks to be completed after running each facility layout program. In this area, the FLAG program distinguishes itself from both PLANET and OPDEP. All that PLANET and OPDEP claim to do is give the user a relative first positioning of workcenters with regard to each other. The user must still design workcenters and then develop a first layout - not necessarily identical or similar to the PLANET or OPDEP results.

FLAG ends when it gives the user the first draft of a plant layout.

This layout includes machines, cabinets, aisles, and other details. Workcenter have already been designed, and templates can be made to manually adjust the initial layout to obtain an even better further refinements and improvements. For the increased time investment during the program execution, the FLAG user will realize considerable time savings at the end of the layout process.

J. Summary of Comparisons

The results of these comparisons in this chapter are summarized below:

1. FLAG, OPDEP, and PLANET show relatively no difference in the preparation stage unless multiple product output from the facility is considered. In the latter case, FLAG involves much less user work.

2. FLAG requires significantly more care and terminal operation time than OPDEP and PLANET.
3. FLAG output is the only output of the three programs that gives internal workcenter detail. Also, workcenter templates are available only by using FLAG. Workcenter shapes are fixed and realistic to the needs of each department.
4. The OPDEP scoring method ranks both OPDEP and PLANET above the FLAG layout. However, the FLAG layout obtained a respectable 89.8% of a perfect score.
5. The move-cost method of scoring ranks FLAG a close second to the OPDEP layout. The insignificant difference can be disregarded because of differences in the nature of FLAG and OPDEP layouts. FLAG scored much better than PLANET using this method.
6. After completion of FLAG, the user if much further along the layout process than with either OPDEP and PLANET.

The next chapter summarizes the work presented in this thesis. Some suggestions for directly extending this research are proposed and evaluated.

VIII. CONCLUSIONS AND SUGGESTIONS FOR FURTHER RESEARCH

As illustrated in the previous chapter, FLAG does have certain disadvantages over other facilities design programs. One of these is additional demands placed upon the user. Like other facility layout programs, FLAG requires the user to collect and enter initial data regarding flow between workcenters. However, the user contribution to a FLAG generated layout does not end at this point. The user is called upon to interact with the program throughout the layout process. This interaction allows the user to participate in the design of all facets of the facility.

The FLAG output reflects this increased user involvement in two related ways. First, the output results in a more realistic relative positioning of workcenters. The user can make necessary adjustments for special requirements, such as larger aisles. The relative positioning of workcenters is also enhanced, in terms of realism, by fixed workcenter shapes. These shapes are determined by the departments' internal requirements. The orientation of workcenters is aimed at reducing distances between points of entry and departure.

Not only are FLAG outputs more realistic, they are also more usable. FLAG can be considered as a multipurpose tool in the plant layout design process. First of all, FLAG can generate a crude layout. Alternatively, the user can generate workcenter templates to use as jigsaw puzzle pieces for developing an improved layout manually. The user can also use FLAG to make element templates for manual workcenter design.

Finally, FLAG is a beginning. Four programs have been described in this thesis that presently constitute the FLAG layout procedure. These programs can be considered a basis for future work in this area. For example, several overlay programs can be developed to design the electrical wiring paths, network configuration, plumbing lines, etc. By developing an element code that designates what elements (or machines) require compressed air, a program can be developed to layout paths for air lines using network analysis.

These extensions will enhance FLAG's existing potential as a viable, general purpose layout tool for the industrial engineer.

IX. BIBLIOGRAPHY

1. Anderson, David M. "New Plant Layout Information System." Industrial Engineering, 5, 3 (1973):32-37.

2. Apple, J. M. Plant Layout and Material Handling. New York: John Wiley and Sons. 1978.

3. Bazaraa, M. S. "Computerized Layout Design: A Branch and Bound Approach." AIEE Transactions, 7, 4 (1975):432-438.

4. Buffa, E. S., Armour, G. C., and Vollmann, T. E. "Allocating Facilities with CRAFT." Harvard Business Review, 42, 2 (1964): 136-59.

5. Denholm, D. H. and Brooks, G. H. "A Comparison of Three Computer Assisted Plant Layout Techniques." Proceedings, AIIE, 21st Annual Convention, Cleveland, 1970. Pp. 77-84.

6. Edwards, H. K., Gillet, B. E., and Hale, M. E. "Modular Allocation Technique (MAT)." Management Science, 17, 3 (1971):161-167.

7. Francis, R. L., and White, J. A. Facility Layout and Location: An Analytical Approach. Englewood Cliffs, New Jersey: Prentice Hall. 1974.

8. Hicks, P. E. and Cowan, T. W. "CRAFT-M for Layout Rearrangement." Industrial Engineering, 8, 5 (1976):30-35.

9. Hiller, F. C., and Connors, M. M. "Quadratic Assignment Problem Algorithms and Location of Indivisible Facilities." Management Science, 13, 1 (1966):42-57.

10. Johnson, Rover V. "SPACE CRAFT for multi-floor layout planning." Management Science, 28, 7 (April 1982):407-417.

11. Kaiman, L. "Computer Programs for Architects and Layout Planning." Proceedings, AIIE, 22nd Annual Convention, Boston, 1971. Pp. 139-147.

12. Khalil, T. M. "Facilities Relative Allocation Technique (FRAT)." International Journal of Production Research, 11, 2 (1973): 183-194.

13. Lee, R. C., and Moore, J. M. "CORELAP - COmputerized RElationship LAyout Planning." Journal of Industrial Engineering, 28, 2 (1967):195-207.

14. Moore, J. M. "Computer Evaluates Layout Alternatives." Industrial Engineering, 3, 1 (1971):19-25.

15. Moore, J. M. "Computer Aided Facilities Design: An International Survey." International Journal of Production Research, 12, 1 (1974):21-44.

16. Muther, R., and McPherson, K. "Four Approaches to Computerized Layout Planning." Industrial Engineering, 2, 2 (1970):39-46.

17. Nelson, A. A New Computer Program for Plant Layout - OPDEP: Optimal Plant Design and Evaluation Program. Master's thesis. Iowa State University, Ames, Iowa, 1980.

18. O'Brien, C. and A. Barr. "An Interactive Approach to Computer Aided Facility Layout." International Journal of Production Research, 18, 3 (1980):74-78.

19. Papineau, R. L., Francis, R. L., and Bartholdi, John J. "A Minimax Facility Layout Problem Involving Distances Between and Within Facilities." AIIE Transactions, 7, 4 (1975):345-355.

20. Ritzman, L. P. "The Efficiency of Computer Algorithms for Plant Layout." Management Science, 18, 5 (1972):240-248.

21. Seehof, J. M., and Evans, W. "Automated Layout DEsign Program." Journal of Industrial Engineering, 28, 1 (1967):69-75.

22. Thompkins, J. A., and Moore, J. M. Computer Aided Layout: A User's Guide." AIIE, Facilities Planning and Design Monograph Series No. 1, 1978.

23. Thorton, V. D., Francis, R. L., and Lowe, T. J. "Rectangular Layout Problems with Worst-case Distance Measures." AIIE Transactions, 11, 1 (1979):2-11.

24. Vollmann, T. E. and Buffa, E. C. "The Facilities Layout Problem in Perspective." Management Science, 12, 10 (1965):B450-B466.

25. Vollmann, T. E., Nuggent, C. E., and Zartler, J. "A Computerized Model for Office Layout." Journal of Industrial Engineering, 19, 3 (1968):321-30.

26. Zoller, K. and Adendorff, K. "Layout Planning by Computer Simulation." AIIE Transactions, 4 (1972):116-125.

X. APPENDIX I: SAMPLE OUTPUTS FROM MAJOR FACILITY LAYOUT PROGRAMS

A. CRAFT Output

INTER-DEPT PRODUCT FLOW

	A	B	C	D	E	F	G	H
A	0.	45.	15.	25.	10.	5.	0.	0.
B	0.	0.	0.	30.	25.	15.	0.	0.
C	0.	0.	0.	0.	5.	10.	0.	0.
D	0.	20.	0.	0.	35.	0.	0.	0.
E	0.	0.	0.	0.	0.	65.	35.	0.
F	0.	5.	0.	0.	25.	0.	65.	0.
G	0.	0.	0.	0.	0.	0.	0.	0.
H	0.	0.	0.	0.	0.	0.	0.	0.

(a)

INTER-DEPT MOVE COST PER UNIT LOAD PER UNIT DISTANCE

	A	B	C	D	E	F	G	H
A	1.	1.	1.	1.	1.	1.	1.	1.
B	1.	1.	1.	1.	1.	1.	1.	1.
C	1.	1.	1.	1.	1.	1.	1.	1.
D	1.	1.	1.	1.	1.	1.	1.	1.
E	1.	1.	1.	1.	1.	1.	1.	1.
F	1.	1.	1.	1.	1.	1.	1.	1.
G	1.	1.	1.	1.	1.	1.	1.	1.
H	1.	1.	1.	1.	1.	1.	1.	1.

(b)

COVOL=(MOVE COST/LOAD) X (NO. OF LOADS)

	A	B	C	D	E	F	G	H
A	0.	45.	15.	25.	10.	5.	0.	0.
B	0.	0.	0.	30.	25.	15.	0.	0.
C	0.	0.	0.	0.	5.	10.	0.	0.
D	0.	20.	0.	0.	35.	0.	0.	0.
E	0.	0.	0.	0.	0.	65.	35.	0.
F	0.	5.	0.	0.	25.	0.	65.	0.
G	0.	0.	0.	0.	0.	0.	0.	0.
H	0.	0.	0.	0.	0.	0.	0.	0.

(c)

LOCATION PATTERN

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	A	A	A	A	A	A	A	A	A	A	G	G	G	G	G	G	G	G
2	A									A	G							G
3	A	A	A	A	A	A	A	A	A	A	G	G	G					G
4	B	B	B	B	B	C	C	C	C	C	E	E	G	G	G	G	G	G
5	B				B	C				C	E	E	E	E	E	E	E	E
6	B			–	B	C	C	C	C	C	E	E	E	E	E	E	E	E
7	B	B	B	B	B	D	D	D	D	F	F	F	F	F	F	F	F	F
8	D	D	D	D	D	D			D	F						F	F	F
9	D							D	D	F	F	F.	F	F				F
10	D	D	D	D	D	D	D	D	H	H	H	H	H	F	F	F	F	F

LOCATION PATTERN

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	A	A	A	A	A	A	A	A	A	A	G	G	G	G	G	G	G	G
2	A									A	G							G
3	A	A	A	A	A	A	A	A	A	A	G	G	G					G
4	B	B	B	B	B	C	C	C	C	C	F	F	G	G	G	G	G	G
5	B				B	C				C	F	F	F	F	F	F	F	F
6	B				B	C	C	C	C	C	F	F	F	F	F	F		F
7	B	B	B	B	B	D	D	D	D	E	E	E	E	E	E	F		F
8	D	D	D	D	D	D			D	E					E	F		F
9	D							D	D	E	E	E	E	E	E	F		F
10	D	D	D	D	D	D	D	D	H	H	H	H	H	E	E	F	F	F

Total Cost 2952.99 Est. Cost Reduction 202.00 Move A E Move B F

Move C Iteration 1

(a)

LOCATION PATTERN

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	A	A	A	A	A	A	A	A	A	A	G	G	G	G	G	G	G	G
2	A									A	G							G
3	A	A	A	A	A	A	A	A	A	A	G	G	G					G
4	C	C	C	B	B	B	B	B	B	B	F	F	G	G	G	G	G	G
5	C		C	C	B					B	F	F	F	F	F	F	F	F
6	C			C	B	B	B	B	B	B	F	F	F	F	F	F		F
7	C	C	C	C	B	D	D	D	C	E	E	E	E	E	E	F		F
8	D	D	D	D	D	D			D	E					E	F		F
9	D							D	D	E	E	E	E	E	E	F		F
10	D	D	D	D	D	D	D	D	H	H	H	H	H	E	E	F	F	F

Total Cost 2833.50 Est. Cost Reduction 95.00 Move A C Move B B

Move C Iteration 2

B. COFAD Output

Location Pattern

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	A	A	A	A	A	A	A	A	A	A	G	G	G	G	G	G	G	G
2	A									A	G							G
3	A	A	A	A	A	A	A	A	A	A	G	G	G					G
4	C	C	C	B	B	B	B	B	B	B	F	F	G	G	G	G	G	G
5	C		C	C	B					B	F	F	F	F	F	F	F	F
6	C			C	B	B	B	B	B	B	F	F	F	F	F	F		F
7	C	C	C	C	B	D	D	D	D	E	E	E	E	E	E	F		F
8	D	D	D	D	D	D			D	E					E	F		F
9	D							D	D	E	E	E	E	E	E	F		F
10	D	D	D	D	D	D	D	D	H	H	H	H	H	E	E	F	F	F

Total Cost 4.47 Est. Cost Reduction 0.0 Move A H Move B D Move C C
********** End Iteration Number 5 **********

```
Equipment Type 2 to Perform Move Number  1 at a Cost of 2566.289
Equipment Type 2 to Perform Move Number  2 at a Cost of 1246.893
Equipment Type 2 to Perform Move Number  3 at a Cost of 2055.423
Equipment Type 2 to Perform Move Number  4 at a Cost of 1250.478
Equipment Type 2 to Perform Move Number  5 at a Cost of 1038.483
Equipment Type 2 to Perform Move Number  6 at a Cost of 2309.[illegible]95
Equipment Type 2 to Perform Move Number  7 at a Cost of 2572.013
Equipment Type 2 to Perform Move Number  8 at a Cost of 1794.702
Equipment Type 2 to Perform Move Number  9 at a Cost of  708.364
Equipment Type 2 to Perform Move Number 10 at a Cost of 1555.284
Equipment Type 2 to Perform Move Number 11 at a Cost of 1518.371
Equipment Type 2 to Perform Move Number 12 at a Cost of 3286.887
Equipment Type 2 to Perform Move Number 13 at a Cost of 3635.783
Equipment Type 2 to Perform Move Number 14 at a Cost of 3226.139
Equipment Type 2 to Perform Move Number 15 at a Cost of  798.013
Equipment Type 2 to Perform Move Number 16 at a Cost of 1595.786
Equipment Type 2 to Perform Move Number 17 at a Cost of 3959.788

Lowest Total Cost System Found =   35118.066
Number of Equipment Changes   0

Equipment Type 1                       Equipment Type 2
****************                       ****************
Number of Moves Included  0            Number of Moves Included   17
Sum of ALCOTK   0.0                    Sum of ALCOTK   35118.07
Number of Pieces of Equipment          Number of Pieces of Equipment
Required   0.0                         Required   2.000
```

C. CORELAP Output

DO YOU WISH A SCORE FOR THIS LAYOUT? 1

THE TOTAL SCORE FOR THIS LAYOUT IS 165
DO YOU WISH TO MAKE CHANGES? Ø

WHICH PARTIAL LAYOUT SHOULD BE PRINTED NEXT? 99

CORELAP BLOCK PLAN LAYOUT

	1	2	3	4	5	6	7	8	9	1Ø	11	12	13	14	15	16
1																
2																
3																
4																
5																
6	Ø	Ø	Ø	Ø	Ø	Ø	Ø	12	12	18	18	15	16	2Ø	2Ø	2Ø
7	Ø	Ø	Ø	Ø	Ø	Ø	Ø	17	12	18	18	16	16	11	2Ø	Ø
8	Ø	Ø	Ø	Ø	Ø	Ø	19	19	19	19	14	16	16	2Ø	2Ø	Ø
9	Ø	Ø	Ø	Ø	Ø	Ø	19	19	19	19	13	13	13	2Ø	2Ø	2Ø
1Ø	Ø	Ø	Ø	Ø	Ø	Ø	19	19	19	19	19	19	13	2Ø	2Ø	2Ø
11																
12																
13																
14																
15																
16																

PROGRAM STOP AT Ø

USED 11.34 UNITS
GOODBYE
ØØ16.43 CRU ØØØØ.74 TCH ØØ11.57 KC

OFF AT 112ØEDT 1Ø/13/7Ø

D. PLANET Output

NUMBER OF DEPARTMENTS = 7 EXAMPLE PROBLEM

UNIT BLOCK SIZE = 400.00

INPUT DATA IS IN THE FORM OF A FROM-TO CHART.

THE TYPE OF SELECTION METHOD USED:
TYPE 1 A LAYOUT WILL BE PRINTED ONLY AFTER THE LAST ITERATION.

TYPE 2 A LAYOUT WILL BE PRINTED ONLY AFTER THE LAST ITERATION.

TYPE 3 A LAYOUT WILL BE PRINTED ONLY AFTER THE LAST ITERATION.

(a)

INPUT DATA FOR DEPARTMENT
BLOCK ALLOCATIONS

DEPARTMENT SYMBOL	REQUIRED AREA	NUMBER OF BLOCKS	PRIORITY	REMARKS
A	12000.	30	2	
B	8000.	20	1	
C	6000.	15	1	
D	12000.	30	1	
E	8000.	20	1	
F	12000.	30	1	
G	12000.	30	2	

THERE ARE 7 DEPARTMENTS AVAILABLE FOR ARRANGEMENT.

NORMALIZED FROM-TO CHART 0.6500000E 02

	A	B	C	D	E	F	G
A	0.0	0.23077	0.69231	0.38462	0.15385	0.07692	0.0
B	0.0	0.0	0.0	0.46154	0.38462	0.23077	0.0
C	0.0	0.0	0.0	0.0	0.07692	0.15385	0.0
D	0.0	0.30769	0.0	0.0	0.53846	0.0	0.0
E	0.0	0.0	0.0	0.0	0.0	1.00000	0.53846
F	0.0	0.07692	0.0	0.0	0.38462	0.0	1.00000
G	0.0	0.0	0.0	0.0	0.0	0.0	0.0

(a)

NORMALIZED FLOW-BETWEEN COST CHART 0.6500000E 02

	A	B	C	D	E	F	G
A	0.0	0.2308	0.6923	0.3846	0.1538	0.0769	0.0
B	0.2308	0.0	0.0	0.7692	0.3846	0.3077	0.0
C	0.6923	0.0	0.0	0.0	0.0769	0.1538	0.0
D	0.3846	0.7692	0.0	0.0	0.5385	0.0	0.0
E	0.1538	0.3846	0.0769	0.5385	0.0	1.3846	0.5385
F	0.0769	0.3077	0.1538	0.0	1.3846	0.0	1.0000
G	0.0	0.0	0.0	0.0	0.5385	1.0000	0.0

LAYOUT COST 47.

(a)

```
            A  A  A
   A  A  A  A  A  A
   A  A  A  A  A  A  C  C  C
   A  A  A  A  A  C  C  C  C
   A  A  A  A  A  C  C  C  C
   A  A  A  A  A  C  C  C  C
E  E  E  E  E  F  F  F  F  F  F  G  G  G  G  G
E  E  E  E  E  F  F  F  F  F  F  G  G  G  G  G
E  E  E  E  E  F  F  F  F  F  F  G  G  G  G  G
E  E  E  E  E  F  F  F  F  F  F  G  G  G  G  G
D  D  D  D  D  F  F  F  F  F  F  G  G  G  G  G
D  D  D  D  D  B  B  B  B        G  G  G  G  G
D  D  D  D  D  B  B  B  B
D  D  D  D  D  B  B  B  B
D  D  D  D  D  B  B  B  B
D  D  D  D  D  B  B  B  B
```

THE ORDER OF PLACEMENT WAS E F D B C G A

LAYOUT COST 48.

(b)

```
A  A  A  A  A  A
A  A  A  A  A  A     C  C  C
A  A  A  A  A  A  C  C  C  C
A  A  A  A  A  A  C  C  C  C
A  A  A  A  A  A  C  C  C  C
E  E  E  E  E  F  F  F  F  F  F
E  E  E  E  E  F  F  F  F  F  F
E  E  E  E  E  F  F  F  F  F  F
E  E  E  E  E  F  F  F  F  F  F
B  B  B  B  B  F  F  F  F  F  F
B  B  B  B  B  G  G  G  G  G
B  B  B  B  B  G  G  G  G  G
B  B  B  B  B  G  G  G  G  G
D  D  D  D  D  G  G  G  G  G
D  D  D  D  D  G  G  G  G  G
D  D  D  D  D  G  G  G  G  G
D  D  D  D  D
D  D  D  D  D
D  D  D  D  D
```

THE ORDER OF PLACEMENT WAS E F B D C G A

LAYOUT COST 48.

(c)

```
A  A  A  A  A  A
A  A  A  A  A  A     C  C  C
A  A  A  A  A  A  C  C  C  C
A  A  A  A  A  A  C  C  C  C
A  A  A  A  A  A  C  C  C  C
E  E  E  E  E  F  F  F  F  F  F
E  E  E  E  E  F  F  F  F  F  F
E  E  E  E  E  F  F  F  F  F  F
E  E  E  E  E  F  F  F  F  F  F
B  B  B  B  B  F  F  F  F  F  F
B  B  B  B  B  G  G  G  G  G
B  B  B  B  B  G  G  G  G  G
B  B  B  B  B  G  G  G  G  G
D  D  D  D  D  G  G  G  G  G
D  D  D  D  D  G  G  G  G  G
D  D  D  D  D  G  G  G  G  G
D  D  D  D  D
D  D  D  D  D
D  D  D  D  D
```

THE ORDER OF PLACEMENT WAS E F B D C G A

E. ALDEP Output

INPUT MATRIX

```
111    S
112    E S
113    O U S
114    I E U S
115    O I U I S
116    U I O U A S
117    U U U U I F S
```

(a)

DECODE MATRIX

```
                           1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3
         1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0
111       016 1 4 1 0 0
112      16 0 016 4 4 0
113       1 0 0 0 0 1 0
114       416 0 0 4 0 0
115       1 4 0 4 064 4
116       0 4 1 064 016
117       0 0 0 0 416 0
```

(b)

DEPARTMENT	REQUIRES AREA	NO. SQUARES
111	12000.000	30
112	8000.000	20
113	6000.000	15
114	11000.000	30
115	8000.000	20
116	12000.000	30
117	12000.000	30

DEPTS. AVAILABLE FOR RANDOM PLACEMENT = 7

(c)

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 1 2 2 2 2 4 4 5 5 6 6 6 6 7 7 7 7 0
0 1 1 2 2 2 2 4 4 5 5 6 6 6 6 7 7 7 7 0
0 1 1 2 2 2 2 4 4 5 5 6 6 6 6 7 7 7 7 0
0 1 1 2 2 2 2 4 4 5 5 6 6 6 6 7 7 7 7 0
0 1 1 2 2 2 2 4 4 5 5 6 6 6 6 7 7 7 7 0
0 1 1 1 1 4 4 4 4 5 5 6 6 3 3 7 7 7 7 0
0 1 1 1 1 4 4 4 4 5 5 6 6 3 3 7 7 7 7 0
0 1 1 1 1 4 4 4 4 5 5 6 6 3 3 3 7 0 7 0
0 1 1 1 1 4 4 4 4 5 5 6 6 3 3 3 3 0 0 0
0 1 1 1 1 4 4 4 4 5 5 6 6 3 3 3 3 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

GROUND FLOOR

```
0 0
0 0
```

TERRACE LEVEL

UNDER THE RULES OF FOR EVALUATION, THIS LAYOUT DOES NOT MEET THE NECESSARY RELATIONSHIPS.

```
111   113
111   115
112   115
112   116
113   111
115   111
115   112
115   117
116   112
117   115
```

XI. APPENDIX II: PLANET AND OPDEP PRINTOUTS FOR RONKO MANUFACTURING EXAMPLE

A. PLANET Output for Ronko Manufacturing Example

```
1[illegible] AUG 82
                                        RONKO MFG CO.

NUMBER OF DEPARTMENTS = 12

UNIT BLOCK SIZE =      50.00

INPUT DATA IS IN THE FORM OF A PARTS LIST.

THE TYPE OF SELECTION METHOD USED'

      TYPE 1   A LAYOUT WILL BE PRINTED ONLY AFTER THE LAST ITERATION.

      TYPE 2'  A LAYOUT WILL BE PRINTED ONLY AFTER THE LAST ITERATION.

      TYPE 3   A LAYOUT WILL BE PRINTED ONLY AFTER THE LAST ITERATION.
```

INPUT DATA FOR DEPARTMENT
BLOCK ALLOCATIONS

DEPARTMENT SYMBOL	REQUIRED AREA	NUMBER OF BLOCKS	PRIORITY	REMARKS
01	2000.	40	2	SHIPREC
02	300.	6	1	SHEAR
03	200.	4	1	PRESS1
04	200.	4	1	PRESS2
05	300.	6	1	ROLLER
06	200.	4	1	RWLDER
07	250.	5	1	SMWELD
08	200.	4	1	SPWELD
09	200.	4	1	WIREOUT
10	300.	6	1	PACKG
11	400.	8	1	ASSEMBLY
12	650.	13	1	PAINTG

THERE ARE 12 DEPARTMENTS AVAILABLE FOR ARRANGEMENT.

INPUT DATA FOR PARTS LIST

PART NO	FREQUENCY OF MOVE	COST/MOVE PER 100 FT.	MOVE SEQUENCE
BEDD	30	0.1500E+01	01 02
AAAB	375	0.7000E+00	02 03 05
AAAC	950	0.2500E+00	05 07 06 12 11
AAAD	950	0.2500E+00	05 07 06 12 11
AAAE	600	0.2500E+00	05 07 06 12 11
AAAF	250	0.2000E+00	01 11
AAAG	950	0.2500E+00	11 10
AAAH	600	0.2500E+00	11 10
AAAI	950	0.2500E+00	10 01
AAAJ	950	0.2500E+00	10 01
AAAK	950	0.2500E+00	10 01
AAAL	325	0.2500E+00	10 01
AAAM	14	0.1500E+01	01 02
AAAN	320	0.2000E+00	02 04 12 11
ANNO	27	0.1500E+01	01 02
AAAP	534	0.2000E+00	02 04 03
AAAQ	32	0.1500E+01	01 02
AAAR	300	0.2000E+00	02 04 03
AAAS	22	0.1500E+01	01 02
AAAT	320	0.2000E+00	02 04 03

PART NO.	FREQUENCY OF MOVE	COST/MOVE PER 100 FT.	MOVE SEQUENCE
AAAU	400	0.2500E+00	08 10 11
AAAV	16	0.2500E+00	01 07
AAAW	48	0.1500E+00	09 11
AAAX	11	0.2500E+00	01 07
AAAY	32	0.1500E+00	09 11
AAAZ	6	0.2500E+00	01 02
AABA	16	0.1500E+00	09 11
AABB	950	0.1000E+00	01 11
AABC	234	0.1000E+00	01 11
AABD	150	0.3000E+00	11 10
AABE	300	0.2000E+00	10 01
AABF	30	0.3000E+00	01 10
AABG	50	0.3000E+00	01 11
AABH	750	0.5000E+00	11 10
AABI	722	0.5000E+00	11 10
AABJ	950	0.5000E+00	11 01
AABK	950	0.5000E+00	11 01
AABL	600	0.5000E+00	11 01

NORMALIZED FROM-TO CHART [illegible]

	01	02	03	04	05	06	07	08	09	10	11	12
01	0.00000	0.14899	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00532	[illegible]	[illegible]	[illegible]
02	0.00000	0.00000	0.20694	0.31123	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
03	0.00000	0.00000	0.00000	0.00000	0.20694	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
04	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.23078	0.00000	0.00000	0.00000	[illegible]
05	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.49271	0.00000	0.00000	0.00000	0.00000	0.00000
06	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.49271
07	0.00000	0.00000	0.00000	0.00000	0.00000	0.49271	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
08	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	[illegible]
09	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.31125	[illegible]
10	0.77158	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
11	[illegible]	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	1.00000	0.00000	0.00000
12	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	[illegible]	0.00000

NORMALIZED FLOW-BETWEEN COST CHART 0.1248500E+04

	01	02	03	04	05	06	07	08	09	10	11	12
01	0.0000	0.1490	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	[illegible]	[illegible]	[illegible]	[illegible]
02	0.1490	0.0000	0.2069	0.3112	0.0000	0.0000	0.0000	0.0000	[illegible]	[illegible]	[illegible]	0.0000
03	[illegible]	0.2069	0.0000	0.0000	0.2069	0.0000	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
04	0.0000	0.3112	0.0000	0.0000	0.0000	0.0000	[illegible]	0.2608	[illegible]	[illegible]	[illegible]	0.2505
05	0.0000	0.0000	0.2069	0.0000	0.0000	0.0000	0.4927	0.0000	[illegible]	[illegible]	0.0000	[illegible]
06	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.4927	0.0000	0.0000	0.0000	0.0000	0.4927
07	0.0000	0.0000	0.0000	0.0000	0.4927	0.4927	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
08	0.0000	0.0000	0.0000	0.2608	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0788
09	0.0053	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0114	0.0000
10	0.7737	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	[illegible]	[illegible]
11	[illegible]	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0114	1.0000	0.0000	[illegible]
12	[illegible]	0.0000	0.0000	0.3505	0.0000	0.4927	0.0000	0.0788	0.0000	0.0000	[illegible]	[illegible]

```
                LAYOUT          COST        28.

                                         09 09
                                         09 09
          10 10 10 11 11 11 11
    01 01 10 10 10 11 11 11 11
    01 01 01 01 01 01 12 12 12
    01 01 01 01 01 01 12 12 12    06    07 05 05
    01 01 01 01 01 01 12 12 12 06 06 07 07 05 05 03 03    08 08
    01 01 01 01 01 01 12 12 12 12 06 07 07 05 05 03 03 02 08 08
    01 01 01 01 01 01 01                         02 02 02 04 04
    01 01 01 01 01 01 01                            02 02 04 04
THE ORDER OF PLACEMENT WAS 10 11 12 06 07 05 03 02 04 08 09 01
```

```
LAYOUT          COST        28.

                     09 09
                     09 09
      10 10 10 11 11 11 11
01 01 10 10 10 11 11 11 11
01 01 01 01 01 01 12 12 12
01 01 01 01 01 01 12 12 12    06    07 05 05
01 01 01 01 01 01 12 12 12 06 06 07 07 05 05 03 03    08 08
01 01 01 01 01 01 12 12 12 12 06 07 07 05 05 03 03 02 08 08
01 01 01 01 01 01 01                         02 02 02 04 04
01 01 01 01 01 01 01                            02 02 04 04
```

THE ORDER OF PLACEMENT WAS 10 11 12 06 07 05 03 02 04 08 09 01

```
LAYOUT          COST        31.

                              01 01 01 01 01 01 01
                                 01 01 01 01 01 01
                                 01 01 01 01 01 01
                                 01 01 01 01 01 01
                        09 09 01 01 01 01 01 01
                        09 09 01 01 01 01 01 01
                  11 11 11 11 10 10 10 01 01 01
                  11 11 11 11 10 10 10
                     12 12 12    05 05 05 03 03
                     12 12 12 07 07 05 05 03 03
                     12 12 12 07 07 06 06 03 02
                     12 12 12 12 07 06 06 02 02
                                    08 02 02 02
                                    08 08 04 04
                                       08 04 04
THE ORDER OF PLACEMENT WAS 11 10 12 07 06 05 02 04 03 08 09 01
```

B. OPDEP Output for Ronko Manufacturing Example

```
    FACILITIES LAYOUT - DESIGN PROGRAM-OPDEPMOD
STARTING NEW JOB, RUN CODE =RONKO MFG. COMPANY       DATE      AUGUST 15, 1982
RANDOM NO. SEEDS ARE 3321 8547
SQUARE SIZE IS  50.0
NUMBER OF LAYOUTS REQUESTED IS   50
MIN ACCEPTABLE SCORE IS  900
VARIABLE FORMAT FOR THE LAYOUT MATRIX IS  (20I3)  (1A3)  (20I2)
SWEEP LENGTH FOR TOP FLOOR     0
SWEEP LENGTH FOR MAIN FLOOR    2
SWEEP LENGTH FOR BASEMENT      0
NO FIRST DEPT TO PLACE SPECIFIED.
```

INPUT MATRIX

```
101  S
102  U S
103  U A S
104  U I A S
105  U U O U S
106  U U U A U S
107  U U U U I E S
108  U U U U I U U S
109  U U U U U U U U S
110  I U U U U U U U U S
111  U U U U U U U U U A S
112  U U U U U E U O U U E S
```

EXPLANATION OF PREFERENCES

A-REPRESENTS THE HIGHEST CATEGORY OF M/H ACTIVITY
E-REPRESENTS A LARGE AMOUNT OF M/H ACTIVITY
I-REPRESENTS A MEDIUM AMOUNT OF M/H ACTIVITY
O-REPRESENTS A SMALL AMOUNT OF M/H ACTIVITY
U-REPRESENTS NO MATERIAL HANDLING ACTIVITY
X-REPRESENTS NO M/H ACTIVITY

NUMERICAL VALUES OF PREFERENCES

A = 90

E = 45

I = 20

O = 5

U = 0

X = 0

SCORING OF LAYOUTS

IF TWO DEPARTMENTS ARE LAID OUT ADJACENT, EITHER DIAGONALLY OR PERPENDICULAR TO EACH OTHER, THE VALUE OF THEIR PREFERENCE IS ADDED TO THE LAYOUT SCORE. IF TWO DEPTS ARE SEPARATED BY ONE SQUARE DIAGONALLY OR PERPENDICULAR, HALF OF THEIR PREFERENCE IS ADDED TO THE SCORE. THOSE DEPTS SEPARATED BY MORE THAN ONE SQUARE HAVE NONE OF THEIR PREFERENCE ADDED TO THE SCORE.

DECODE MATRIX

```
                           1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5 5 5 5 5 6 6 6 6
       1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
101    0 0 0 0 0 0 0 0 09020 0
102    0 0 520 0 0 0 0 0 0 0 0
103    0 5 0 0 5 0 0 0 0 0 0 0
104    020 0 0 0 0 020 0 0 0 5
105    0 0 5 0 0 045 0 0 0 0 0
106    0 0 0 0 0 045 0 0 0 045
107    0 0 0 04545 0 0 0 0 0 0
108    0 0 020 0 0 0 0 0 0 0 5
109    0 0 0 0 0 0 0 0 0 0 0 0
110   90 0 0 0 0 0 0 0 0 090 0
111   20 0 0 0 0 0 0 0 090 045
112    0 0 0 5 045 0 5 0 045 0
```

```
TOTAL NUMBER OF SQUARES IN BLDG.= 104
      TOP FLOOR SQUARES=   0
      MAIN FLOOR SQS. =  104
      BASEMENT  SQUARES=   0

TOTAL SQUARES AVAILABLE TO LAYOUT= 104
      TOP FLOOR SQUARES=   0
      MAIN FLOOR SQS. =  104
      BASEMENT  SQUARES=   0

TOTAL PREASSIGNED SQS ALL FLOORS=     0
      TOP FLOOR SQUARES=   0
      MAIN FLOOR SQS. =    0
      BASEMENT  SQUARES=   0

TOTAL SQS REQD. FOR ALL DEPTS=     104

TOTAL BUFFER SQUARES AVAILABLE=      0

NO. DEPTS TO BE PLACED BY THE DESIGN PROGRAM=  12

NO. SQS TO BE PLACED BY THE DESIGN PROGRAM  = 104
```

DEPARTMENT	REQUIRED AREA	NO. SQUARES	NAME
101	2000.000	40	SHIPPING/RECEIVING
102	300.000	6	DECOILER/SHEAR
103	200.000	4	PUNCH PRESS A
104	200.000	4	PUNCH PRESS B
105	300.000	6	ROLLER
106	200.000	4	PHLANGER
107	250.000	5	SEAM WELDER
108	200.000	4	SPOT WELDER
109	200.000	4	WIRE CUTTER
110	300.000	6	PACKAGING
111	400.000	8	ASSEMBLY
112	450.000	[illegible]	PAINTING

TRIAL LAYOUT 25 RONKO MFG. COMPANY SCORE= 930 EFFICIENCY= 105.[illegible] %
PERFECT SCORE = 880

TOP FLOOR

0	0
0	0

MAIN FLOOR

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	8	8	12	12	12	12	1	1	1	1	0	0	0
0	8	8	12	[illegible]	12	12	1	1	1	1	0	0	0
0	4	4	6	6	12	12	1	1	1	1	9	9	0
0	4	4	6	6	12	12	1	1	1	1	9	9	0
0	3	3	7	7	12	12	1	1	1	1	1	1	0
0	3	3	7	7	11	11	1	1	1	1	1	1	0
0	2	2	5	5	11	11	10	10	1	1	1	1	0
0	2	2	5	5	11	11	10	10	1	1	1	1	0
0	2	2	5	5	11	11	10	10	1	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

BASEMENT FLOOR

0	0
0	0

ACCORDING TO THE PREFERENCES INPUT FOR THIS JOB, THIS LAYOUT SATISFIES ALL PREFERENCES THAT ARE GREATER THAN OR EQUAL TO 90.

** NOTE ** THE ABOVE LIST DOES NOT INCLUDE 4 PREFERENCES HALF SATISFIED BECAUSE OF A SQUARE BETWEEN THEM, WHICH INCREASED THE SCORE OF THIS LAYOUT BY 50. IT IS POSSIBLE TO SATISFY ALL PREFERENCES BY THESE CRITERIA AND NOT GET A PERFECT SCORE

THIS LAYOUT SATISFIES 4 OF 4 PREFERENCES INPUT THAT ARE > OR = 90.

LEGEND OF DEPARTMENTS USED IN LAYOUT

1 SHIPPING/RECEIVING	2 DECOILER/SHEAR	3 PUNCH PRESS A
4 PUNCH PRESS B	5 ROLLER	6 FLANGER
7 SEAM WELDER	8 SPOT WELDER	9 WIRE CUTTER
10 PACKAGING	11 ASSEMBLY	12 PAINTING

```
TRIAL LAYOUT  31 RONKO MFG. COMPANY    SCORE=   928  EFFICIENCY= 105.5 %
                                                     PERFECT SCORE =   880

                    TOP FLOOR
0  0
0  0

                    MAIN FLOOR
0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  3  3 12 12 12 12  1  1  1  1  0  0  0
0  3  3 12  6 12 12  1  1  1  1  0  0  0
0  4  4  6  6 12 12  1  1  1  1  9  9  0
0  4  4  6  7 12 12  1  1  1  1  9  9  0
0  2  2  7  7 12 12  1  1  1  1  8  8  0
0  2  2  7  7 11 11  1  1  1  1  8  8  0
0  2  2  5  5 11 11 10 10  1  1  8  8  0
0  2  2  5  5 11 11 10 10  1  1  8  8  0
0  2  2  5  5 11 11 10 10  1  1  8  8  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0

                    BASEMENT FLOOR
0  0
0  0
```

ACCORDING TO THE PREFERENCES INPUT FOR THIS JOB,
THIS LAYOUT SATISFIES ALL PREFERENCES THAT ARE
GREATER THAN OR EQUAL TO 90.

NOTE ## THE ABOVE LIST DOES NOT INCLUDE 4 PREFERENCES HALF SATISFIED BECAUSE OF A SQUARE BETWEEN THEM, WHICH INCREASED THE SCORE OF THIS LAYOUT BY 48. IT IS POSSIBLE TO SATISFY ALL PREFERENCES BY THESE CRITERIA AND NOT GET A PERFECT SCORE

THIS LAYOUT SATISFIES 4 OF 4 PREFERENCES INPUT THAT ARE > OR = 90.

LEGEND OF DEPARTMENTS USED IN LAYOUT

1	SHIPPING/RECEIVING	2	DECOILER/SHEAR	3	PUNCH PRESS A
4	PUNCH PRESS B	5	ROLLER	6	FLANGER
7	SEAM WELDER	8	SPOT WELDER	9	WIRE CUTTER
10	PACKAGING	11	ASSEMBLY	12	PAINTING

TRIAL LAYOUT 42 RONKO MFG. COMPANY SCORE= 779 EFFICIENCY= [illegible] %
PERFECT SCORE = [illegible]

TOP FLOOR

0	0
0	0

MAIN FLOOR

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	3	3	12	12	12	12	1	1	1	1	0	0	0
0	3	3	12	6	12	12	1	1	1	1	0	0	0
0	4	4	6	6	12	12	1	1	1	1	9	9	0
0	4	4	6	7	12	12	1	1	1	1	9	9	0
0	2	2	7	7	12	12	1	1	1	1	1	1	0
0	2	2	7	7	11	11	1	1	1	1	1	1	0
0	2	2	5	5	11	11	1	1	1	1	1	1	0
0	2	2	5	5	11	11	10	10	1	1	1	1	0
0	3	3	5	5	11	11	10	10	1	1	1	1	0
0	4	4	5	5	11	11	10	10	1	1	1	1	0
0	4	3	5	5	11	11	10	10	1	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

BASEMENT FLOOR

0	0
0	0

ACCORDING TO THE PREFERENCES INPUT FOR THIS JOB,
THIS LAYOUT SATISFIES ALL PREFERENCES THAT ARE
GREATER THAN OR EQUAL TO 90.

** NOTE ** THE ABOVE LIST DOES NOT INCLUDE 4 PREFERENCES HALF SATISFIED
BECAUSE OF A SQUARE BETWEEN THEM, WHICH INCREASED THE SCORE OF THIS LAYOUT
BY 90. IT IS POSSIBLE TO SATISFY ALL PREFERENCES BY THESE CRITERIA AND NOT
GET A PERFECT SCORE

THIS LAYOUT SATISFIES 4 OF 4 PREFERENCES INPUT THAT ARE > OR = 90.

LEGEND OF DEPARTMENTS USED IN LAYOUT

1	SHIPPING/RECEIVING	2	DECOILER/SHEAR	3	PUNCH PRESS A
4	PUNCH PRESS B	5	ROLLER	6	FLANGER
7	SEAM WELDER	8	SPOT WELDER	9	WIRE CUTTER
10	PACKAGING	11	ASSEMBLY	12	PAINTING

XII. APPENDIX III: FLAG PROGRAMS

A. FLOWS Source Program

```
C
C
C
C
C
      CHARACTER *20 NFAC
      CHARACTER *15 NMGRP(5), NMWTR(25)
      CHARACTER *10 NMPRDT(5)
      CHARACTER *5  ADNMW(25)
      CHARACTER *7 PGM(4)
      INTEGER NGRP, GRPN, NWCTR, NPRDT
      INTEGER NFLOWS(5), NEL(25)
      INTEGER PHASE, STAGE
      INTEGER PQTY,UPM
      INTEGER FWC,TWC,MIX(5)
      INTEGER SQTY,QTY,NMVES
      REAL MCF,A(25,25),D(25,25)
      REAL COST
      PHASE=0
      IERR=2
      STAGE = 1
      PGM(1)='FLOWS'
C
C
C
      CALL GRSTRT(4051,1)
      CALL CMCLOS
C
C
C
C
C
    1 FORMAT(A20)
    2 FORMAT(I2)
    3 FORMAT(I1)
    4 FORMAT(A15)
    5 FORMAT(A10)
    6 FORMAT(I3)
    7 FORMAT(A5)
    8 FORMAT(I7)
    9 FORMAT(F7.3)
   26 FORMAT(1X,A20)
   27 FORMAT(1X,I2,5X,I1)
   28 FORMAT(1X,A15)
   29 FORMAT(1X,I1)
   30 FORMAT(1X,A10)
   31 FORMAT(1X,I2)
   32 FORMAT(1X,A15,5X,A5)
   33 FORMAT(1X,A10,5X,I3)
```

```
 34    FORMAT(1X,2(I7,2X),F7.3,2(2X,I2))
 35    FORMAT(1X,25(F10.3,2X))
 36    FORMAT(1X,I7)
 37    FORMAT(1X,F10.3)
C
C
C
C
C
100    CALL CLEAR
101    PRINT *
       PRINT *
       PRINT *
       PRINT *,'SELECT TASK FROM THE FOLLOWING:'
       PRINT *
       PRINT *,'      1. INPUT GENERAL DATA - WORKSHEET FLOWS 1
       PRINT *,'      2. INPUT FLOWS DATA - WORKSHEET FLOWS 2'
       PRINT *,'      3. CALCULATE FROM/TO CHART'
       PRINT *,'      4. EXIT FLOWS'
       PRINT *
       PRINT *,'INPUT SELECTION NUMBER AND PRESS <RETURN>'
       READ (*,3) MS
       IF (MS .EQ. 1) GO TO 105
       IF (MS .EQ. 2) GO TO 300
       IF (MS .EQ. 3) GO TO 500
       IF (MS .EQ. 4) GO TO 600
       CALL CLEAR
       PRINT *
       PRINT *
       CALL WARN(IERR)
       PRINT *,'ERROR: IMPROPER SELECTION NUMBER ENTERED!!!!'
       GO TO 101
105    CALL CLEAR
       CALL TRISPAC
       PRINT *,'INPUT THE NAME OF THE FACILITY TO BE DESIGNEI
       PRINT *,'      AND PRESS <RETURN>.'
       CALL DUSPAC
       PRINT *,'NOTE: LIMIT OF 20 CHARACTERS.'
       READ (*,1) NFAC
110    CALL HDGA(STAGE,NFAC,PGM)
       CALL DUSPAC
       PRINT *,'INPUT YOUR CLASS DESIGN GROUP NUMBER'
       PRINT *,'      AND PRESS <RETURN>.'
       READ (*,2) GRPN
115    CALL HDGA(STAGE,NFAC,PGM)
       CALL TRISPAC
       PRINT *,'ENTER THE NUMBER OF MEMBERS IN YOUR'
       PRINT *,'      DESIGN GROUP AND PRESS <RETURN>.'
       READ (*,3)NGRP
116    DO 117 I=1,NGRP
       CALL HDGA(STAGE,NFAC,PGM)
```

```
      CALL TRISPAC
118   FORMAT(1X,'ENTER THE NAME OF GROUP MEMBER NO. ',I1)
      PRINT 118,I
      PRINT *,'      AND PRESS <RETURN>.'
      CALL DUSPAC
      PRINT *,'NOTE: LIMIT OF 15 CHARACTERS.'
      READ (*,4) NMGRP(I)
117   CONTINUE
120   CALL HDGA(STAGE,NFAC,PGM)
      CALL TRISPAC
      PRINT *,'ENTER THE NUMBER OF PRODUCTS TO BE'
      PRINT *,'      MANUFACTURED AT THE FACILITY UNDER'
      PRINT *,'      DESIGN AND PRESS <RETURN>.'
      READ (*,3) NPRDT
121   DO 123 I = 1,NPRDT
      CALL HDGA(STAGE,NFAC,PGM)
      CALL TRISPAC
122   FORMAT(1X,'ENTER THE NAME OF PRODUCT NO. ',I1)
      PRINT 122,I
      PRINT *,'      AND PRESS <RETURN>.'
      CALL DUSPAC
      PRINT *,'NOTE: LIMIT OF 10 CHARACTERS.'
      READ (*,5) NMPRDT(I)
123   CONTINUE
130   CALL HDGA(STAGE,NFAC,PGM)
      CALL TRISPAC
      PRINT *,'ENTER THE NUMBER OF WORKCENTERS (OR DEPTS.)'
      PRINT *,'      IN THE FACILITY UNDER DESIGN AND PRESS'
      PRINT *,'      <RETURN>.'
      READ (*,2) NWCTR
135   DO 136 I = 1,NWCTR
      CALL HDGA(STAGE,NFAC,PGM)
      CALL DUSPAC
137   FORMAT(1X,'ENTER THE NAME OF WORKCENTER NO. ',I2)
      PRINT 137,I
      PRINT *,'      AND PRESS <RETURN>.'
      CALL DUSPAC
      PRINT *,'NOTE: LIMIT OF 15 CHARACTERS.'
      READ (*,4) NMWTR(I)
      CALL TRISPAC
      PRINT *,'ENTER A FIVE CHARACTER, OR LESS, ABBREVIATED'
      PRINT *,'      LABEL FOR THE ABOVE WORKCENTER AND'
      PRINT *,'      PRESS <RETURN>.'
      READ (*,7) ABNMW(I)
136   CONTINUE
      PHASE=1
      OPEN (UNIT=1,FILE='GEN.DAT',TYPE='NEW',DISP='KEEP',
     1 FORM='FORMATTED')
      WRITE (1,26) NFAC
      WRITE (1,27) GRPN, NGRP
      DO 151 I= 1,NGRP
```

```
      WRITE (1,28) NMGRP(I)
151   CONTINUE
      WRITE (1,29) NPRDT
      DO 152 I=1,NPRDT
      WRITE(1,30) NMPRDT(I)
152   CONTINUE
      WRITE (1,31) NWCTR
      DO 153 I = 1,NWCTR
      WRITE (1,32) NMWTR(I),ABNMW(I)
153   CONTINUE
      CLOSE(UNIT=1)
      GO TO 100
300   OPEN(UNIT=2,TYPE='NEW',DISP='KEEP',FILE='FLOWS.DAT',
     1 FORM='FORMATTED')
      DO 310 I = 1,NPRDT
      CALL HDGA(STAGE,NFAC,PGM)
      CALL DUSPAC
301   FORMAT(1X,'PRODUCT: ',A10)
      PRINT 301,NMPRDT(I)
      CALL DUSPAC
      PRINT *,'ENTER THE NO. OF INTERDEPARMENTAL FLOWS'
      PRINT *,'      ASSOCIATED WITH THE ABOVE PRODUCT'
      PRINT *,'      AND PRESS <RETURN>.'
      READ (*,6) NFLOWS(I)
      WRITE (2,33) NMPRDT(I),NFLOWS(I)
310   CONTINUE
      DO 320 I = 1, NPRDT
      DO 325 J = 1, NFLOWS(I)
      CALL HDGA(STAGE,NFAC,PGM)
      CALL DUSPAC
321   FORMAT(3X,'PRODUCT:                          ',A10)
322   FORMAT(3X,'INTERDEPARTMENTAL FLOW NO.:  ',I3)
      CALL CMOPEN
      CALL MOVE(0.0,68.0)
      CALL DRAW(130.0,68.0)
      CALL DRAW(130.0,76.0)
      CALL DRAW(0.0,76.0)
      CALL DRAW(0.0,68.0)
      CALL HOME
      CALL CMCLOS
      CALL TRISPAC
      CALL TRISPAC
      CALL DUSPAC
      PRINT 321,NMPRDT(I)
      PRINT 322,J
      CALL DUSPAC
      PRINT *,'FOR THE ABOVE PRODUCT/FLOW ENTER THE '
      PRINT *,'      NUMBER OF UNITS PER PRODUCT.'
      READ (*,8) PQTY
      PRINT *
      PRINT *,'FOR THE ABOVE PRODUCT/FLOW ENTER THE'
```

```
      PRINT *,'      UNITS PER MOVE.'
      READ (*,8) UPM
      PRINT *
      PRINT *,'FOR THE ABOVE PRODUCT/FLOW ENTER THE'
      PRINT *,'      MOVE-COST PER FOOT.'
      READ (*,9) MCF
      CALL HDGA(STAGE,NFAC,PGM)
      CALL DUSPAC
      CALL CMOPEN
      CALL MOVE(0.0,68.0)
      CALL DRAW(130.0,68.0)
      CALL DRAW(130.0,76.0)
      CALL DRAW(0.0,76.0)
      CALL DRAW(0.0,68.0)
      CALL HOME
      CALL CMCLOS
      CALL TRISPAC
      CALL TRISPAC
      CALL DUSPAC
      PRINT 321,NMPRDT(I)
      PRINT 322,J
      CALL TRISPAC
      PRINT *,'ENTER THE NUMBER OF THE WORKCENTER'
      PRINT *,'      THE ABOVE PRODUCT/FLOW DEPARTS.'
      READ (*,2) FWC
      CALL TRISPAC
      PRINT *,'ENTER THE NUMBER OF THE WORKCENTER'
      PRINT *,'      THE ABOVE PRODUCT/FLOW ENTERS.'
      READ (*,2) TWC
      WRITE (2,34) PQTY,UPM,MCF,FWC,TWC
325   CONTINUE
320   CONTINUE
      CLOSE(UNIT=2)
      PHASE=2
      GO TO 100
500   OPEN(UNIT=1,FILE='GEN.DAT',TYPE='OLD',DISP='KEEP',
     1 FORM='FORMATTED')
      OPEN(UNIT=2,FILE='FLOWS.DAT',TYPE='OLD',DISP='KEEP',
     1 FORM='FORMATTED')
      OPEN(UNIT=3,FILE='FROMTO.DAT',TYPE='NEW',DISP='KEEP',
     1 FORM='FORMATTED')
      READ (1,26) NFAC
      READ (1,27) GRPN,NGRP
      DO 501 I=1,NGRP
501   READ(1,28)NMGRP(I)
      READ(1,29)NPRDT
      DO 502 I=1,NPRDT
      READ(1,30)NMPRDT(I)
502   READ(2,33)NMPRDT(I),NFLOWS(I)
      READ(1,31)NWCTR
      DO 503 I=1,NWCTR
```

```
503     READ(1,32)NMWTR(I),ABNMW(I)
        CLOSE (UNIT=1)
        DO 504 I=1,25
        DO 505 J=1,25
        A(I,J)=0.0
        B(I,J)=0.0
505     CONTINUE
504     CONTINUE
        CALL HDGA(STAGE,NFAC,PGM)
        CALL DUSPAC
506     FORMAT(1X,'ENTER REQUIRED NUMBER OF ',A10)
507     FORMAT(6X,'TO BE MANUFACTURED.')
        DO 508 I=1,NPRDT
        PRINT 506,NMPRDT(I)
        PRINT 507
        READ (*,8) MIX(I)
        WRITE(3,36)MIX(I)
508     CONTINUE
        CALL CLEAR
        DO 510 I=1,NPRDT
        DO 515 J=1,NFLOWS(I)
        READ(2,34)PQTY,UPM,MCF,FWC,TWC
        QTY = PQTY * MIX(I)
        NMVES= INT(QTY/UPM)+1
        COST= NMVES * MCF
        A(FWC,TWC)=A(FWC,TWC)+COST
        B(FWC,TWC)=B(FWC,TWC)+COST
 515    CONTINUE
 510    CONTINUE
 511    DO 527 I=1,NWCTR
        DO 526 K=1,NWCTR
 526    WRITE (3,37) B(I,K)
 527    CONTINUE
        CLOSE(UNIT=2)
        CLOSE(UNIT=3)
600     END
$
```

B. WORKOUT Source Program

```
C
C
C       DECLARATIONS
C
        REAL PAR(10)
        REAL PARP(10)
        REAL BINT(4), BORD(4)
        REAL NOTCH(4,4),AREA(3)
        REAL CEN(2),PO(2,3),FT(25,25)
        REAL CF(2),NF(4,4),PF(2,3),BF(4)
        REAL OLD(2,2)
        INTEGER GRPN,NGRP,NPRDT,NWCTR,NEL
        INTEGER TN,TY,DIR
        INTEGER TNP,TYP,TPLAC,PH
        INTEGER NBPO(2)
        INTEGER SCREF
        CHARACTER * 20 NFAC
        CHARACTER * 15 NMGRP(5), NMWTR(25)
        CHARACTER * 10 NMPRDT(5), LBL
        CHARACTER *10 LBLP
        CHARACTER * 7 PGM(4)
        CHARACTER * 5 ABNMW(25),WAL(4)
        CHARACTER * 3 POL(2)
        WAL(1)='NORTH'
        WAL(2)='SOUTH'
        WAL(3)='EAST'
        WAL(4)='WEST'
        POL(1)='POE'
        POL(2)='POD'
        PH=2
        IERR=2
        MAXSC=1
        ISTART=1
        ISEL=0
        JPLAC=0
        SINCH=.3585771658
        CALL GRSTRT(4051,1)
        CALL CMCLOS
C
C
C       FORMATS
C
    1   FORMAT(1X,A20)
    2   FORMAT(1X,I2,5X,I1)
    3   FORMAT(1X,A15)
    4   FORMAT(1X,I1)
    5   FORMAT(1X,A10)
    6   FORMAT(1X,I2)
```

```
    7     FORMAT(1X,A15,5X,A5)
    8     FORMAT(I2)
    9     FORMAT(1X,I2,5X,A15,5X,I2)
   10     FORMAT(I1)
   11     FORMAT(1X,I1,5X,I2,5X,A10)
   12     FORMAT(1X,2(F10.3,5X),F10.3)
   13     FORMAT(1X,10F10.3)
   14     FORMAT(2F10.3,F5.2)
   15     FORMAT(1X,F4.1)
   16     FORMAT(F7.3)
   17     FORMAT(1X,25(F10.3,2X))
   18     FORMAT(1X,4(F10.3,2X))
   19     FORMAT(1X,2(F10.3,2X))
   20     FORMAT(1X,2(I1,2X))
   21     FORMAT(1X,3(F10.3,2X))
   22     FORMAT(1X,F10.3)
          GO TO 89
C
C
C         INITIALIZE PROGRAM
C
   65     OPEN(UNIT=1,FILE='GEN.DAT',TYPE='OLD',
     1    DISP='KEEP',FORM='FORMATTED')
          OPEN(UNIT=2,FILE='WCEN.DAT',TYPE='NEW',
     1    DISP='KEEP',FORM='FORMATTED')
          OPEN(UNIT=3,FILE='BORD.DAT',TYPE='NEW',
     1    DISP='KEEP',FORM='FORMATTED')
          OPEN(UNIT=13,FILE='FROMTO.DAT',TYPE='OLD',
     1    DISP='KEEP',FORM='FORMATTED')
          OPEN(UNIT=8,FILE='WORKOUT.DAT',TYPE='NEW',
     1    DISP='KEEP',FORM='FORMATTED')
   70     READ (1,1) NFAC
          READ (1,2)GRPN,NGRP
          DO 76 I = 1,NGRP
   76     READ (1,3) NMGRP(I)
          READ (1,4) NPRDT
          DO 77 I = 1,NPRDT
   77     READ (1,5) NMPRDT(I)
          READ (1,6) NWCTR
          DO 78 I = 1,NWCTR
   78     READ (1,7) NMWTR(I),ABNMW(I)
          CLOSE(UNIT=1)
          KNUM=NPRDT+(NPRDT*NWCTR*NWCTR)
          DO 79 KKO = 1,NPRDT
   79     READ (13,5) LBLP
          DO 80 KKI = 1,NWCTR
          DO 85 KKIP= 1,NWCTR
          READ (13,22) FT(KKI,KKIP)
   85     CONTINUE
   80     CONTINUE
          CLOSE(UNIT=13)
```

```
      GO TO 101
C
C
C
C
 89   CALL CLEAR
      CALL TRISPAC
      PRINT *,'INPUT THE NUMBER OF PREVIOUSLY DESIGNED'
      PRINT *,'        WORKCENTERS AND PRESS <RET>'
      READ *,NPD
      IF (NPD .EQ. 0) GO TO 65
      ISTART = NPD + 1
      OPEN(UNIT=2,FILE='WCEN.DAT',TYPE='OLD',
     1 DISP='KEEP',FORM='FORMATTED')
      OPEN(UNIT=8,FILE='WORKOUT.DAT',TYPE='OLD',
     1 DISP='KEEP',FORM='FORMATTED')
      OPEN(UNIT=3,FILE='BORD.DAT',TYPE='OLD',
     1 DISP='KEEP',FORM='FORMATTED')
      OPEN(UNIT=1,FILE='GEN.DAT',TYPE='OLD',
     1 DISP='KEEP',FORM='FORMATTED')
      OPEN(UNIT=13,FILE='FROMTO.DAT',TYPE='OLD',
     1 DISP='KEEP',FORM='FORMATTED')
      DO 95 I = 1,NPD
      READ (2,9) KK,NMWTR(KK),NEL
      READ (3,9) KK,NMWTR(KK),NEL
      READ (8,6) NEL
      READ (2,6) IS
      READ (3,6) IS
      IF (IS .GT. MAXSC) MAXSC = IS
      NREC = NEL * 3
      NRECA = NEL * 4
      DO 91 II = 1,NREC
 91   READ(2,5) LBLP
      DO 92 II = 1,NRECA
 92   READ(8,5) LBLP
      DO 93 II = 1,19
 93   READ (3,5) LBLP
 95   CONTINUE
      GO TO 70
C
C
C     MAIN WORKCENTER LOOP
C
 101  DO 100 I = ISTART,NWCTR
      SCREF=1
      CALL CLEAR
      CALL HDGA(PH,NFAC,PGM)
      CALL DUSPAC
      PRINT 111,I ,NMWTR(I)
 111  FORMAT(1X,'WORKCENTER NO.: ',1X,I2,2X,'-',2X,A15
      CALL DUSPAC
```

```
      PRINT *,' INPUT THE NUMBER OF ELEMENTS TO'
      PRINT *,' BE PLACED IN THE ABOVE WORKCENTER'
      PRINT *,' AND PRESS <RETURN>.'
      READ (*,8)NEL
      WRITE(2,9)I,NMWTR(I),NEL
      WRITE(3,9)I,NMWTR(I),NEL
      IPLAC=0
      JPLAC=0
      TPLAC=0
      DO 150 JKI = 1,4
      DO 151 IJK = 1,4
 151  NOTCH(IJK,JKI) = 0.0
 150  CONTINUE
      DO 152 JKI=1,3
 152  AREA(JKI)=0.0
      OPEN(UNIT=11,FILE='T1.DAT',TYPE='NEW',
     1 DISP='KEEP',FORM='FORMATTED')
      CLOSE(UNIT=11)
C
C
C     BEGIN ELEMENT LOOP
C
      DO 350 J = 1,NEL
C
C
C     ELEMENT LOOP MAIN BOTTLENECK
C
 205  IF (IPLAC .EQ. 0) GO TO 210
      IF (IPLAC .EQ. 1) GO TO 240
      IF (IPLAC .EQ. 2) GO TO 220
      IF (IPLAC .GE. 3) GO TO 240
C
C
C     FIRST TEMP /REDRAW BYPASS
C
 210  CALL FRAME(SCREF,XA,YA,STEP)
      CALL FRAMEA(SCREF,XA,YA,STEP)
      IF (IPLAC .EQ. 1) GO TO 250
      GO TO 223
C
C
C     REDRAW ROUTINE
C
 220  OPEN(UNIT=11,FILE='T1.DAT',TYPE='OLD',
     1 DISP='KEEP',FORM='FORMATTED')
      CALL FRAME(SCREF,XA,YA,STEP)
      CALL FRAMEA(SCREF,XA,YA,STEP)
      DO 221 JJ = 1,TPLAC
      READ(11,11) TYP,TNP,LBLP
      READ(11,12) XP,YP,RP
      READ(11,13)(PARP(JK),JK=1,10)
```

```
      CALL RDRAW(XA,YA,TYP,TNP,XP,YP,RP,PARP,LBLP,SINC
  221 CONTINUE
      CLOSE(UNIT=11)
      IF(JPLAC .GT. 0) GO TO 500
      IF(IPLAC .EQ. 2) GO TO 223
      IF(IPLAC .EQ. 3) GO TO 242
      IF(IPLAC .EQ. 4) GO TO 250
C
C
C     SELECT TEMPLATE TYPE
C
  223 CALL FSPAC
  224 PRINT *,'SELECT TEMPLATE TYPE AND PRESS <RET>.'
      PRINT *
      PRINT *,'        1. GENERAL SHAPE TEMPLATE.'
      PRINT *,'        2. PLANPRINT MACHINE TEMPLATE.'
      READ (*,10) TY
      IF (TY .EQ. 1) GO TO 225
      IF (TY .EQ. 2) GO TO 233
      CALL CLEAR
      CALL WARN(IERR)
      PRINT *,'ERROR: MUST ANSWER 1 OR 2.'
      CALL TRISPAC
      GO TO 224
C
C
C     MENUA - GENERAL SHAPE TEMPLATES
C
  225 CALL MENUA(PGM)
      READ(*,8) TN
      IF(IPLAC .EQ. 0)GO TO 227
      IPLAC = 3
  228 IF(TN .EQ. 1) GO TO 226
      IF(TN .EQ. 2) CALL RECTA(PAR,LBL)
      IF(TN .EQ. 3) CALL DRECTA(PAR,LBL)
CR
CR
      GO TO 205
  226 CALL SQUAREA(PAR,LBL)
      GO TO 205
CR
CR
CR
CR
  227 IPLAC =1
      GO TO 228
C
C
C     MENUB - MACHINE TEMPLATES
C
  233 CALL MENUB(PGM)
```

```
      READ (*,8) TN
      IF (IPLAC .EQ. 0) GO TO 229
      IPLAC = 3
      GO TO 205
 229  IPLAC = 1
      GO TO 205
C
C
C     PLACE NEW TEMPLATE
C
 240  IF (IPLAC .GE. 2) GO TO 220
      CALL FRAME(SCREF,XA,YA,STEP)
      CALL FRAMEA(SCREF,XA,YA,STEP)
 242  CALL FSPAC
      PRINT *,'INPUT X-COORDINATE, Y-COORDINATE, AND'
      PRINT *,'ROTATION FACTOR SEPARATED BY COMMAS.'
      READ(*,14) X,Y,R
      IF (IPLAC .EQ. 1) GO TO 210
      IPLAC = 4
      GO TO 205
C
C
C     END LOOP OPTIONS
C
 250  CALL RDRAW(XA,YA,TY,TN,X,Y,R,PAR,LBL,SINCH)
 251  CALL FSPAC
      PRINT *,'SELECT OPTION AND PRESS <RET>.'
      PRINT *
      PRINT *,'      1. PLACEMENT APPROVED.'
      PRINT *,'      2. TEMPLATE ADJUSTMENT NEEDED.'
      PRINT *,'      3. INCREASE FRAME AREA.'
      READ(*,10)ISEL
      IF (ISEL .EQ. 1) GO TO 260
      IF (ISEL .EQ. 2) GO TO 270
      IF (ISEL .EQ. 3) GO TO 280
      CALL CLEAR
      CALL WARN(IERR)
      PRINT *,'ERROR: MUST SELECT 1, 2, OR 3.'
      GO TO 251
C
C
C     PLACEMENT APPROVED
C
 260  ISEL = 0
      ISKIM = 3*IPLAC
      OPEN(UNIT=11,FILE='T1.DAT',TYPE='OLD',
     1 DISP='KEEP',FORM='FORMATTED')
      IF (IPLAC .LE. 1) GO TO 269
      DO 261 IS =1,ISKIM
 261  READ(11,5)LBLP
 269  WRITE(11,11) TY,TN,LBL
```

```
      WRITE(11,12) X,Y,R
      WRITE(11,13) (PAR(JK),JK=1,10)
      CLOSE(UNIT=11)
      IPLAC = 2
      TPLAC =TPLAC +1
      GO TO 350
C
C
C     REPLACE TEMPLATE
C
270   ISEL = 0
      IPLAC=3
      IF (IPLAC .EQ.. 1) GO TO 240
      GO TO 240
C
C
C     DOUBLE FRAME AREA
C
280   ISEL = 0
      IOLD =SCREF
      SCREF = SCREF +1
      XCP=(130.0*SCREF)/2.0
      XOLD=(130.0*IOLD)/2.0
      YCP=(100.0*SCREF)/2.0
      YOLD=(100.0*IOLD)/2.0
      DX=XCP-XOLD
      DY=YCP-YOLD
      X=X+DX
      Y=Y+DY
      IF(JPLAC .EQ. 0) GO TO 286
      BORD(1) = BORD(1)+DY
      BORD(2) = BORD(2)+DY
      BORD(3) = BORD(3)+DX
      BORD(4) = BORD(4)+DX
      IF (NOTCH(1,1) .EQ. 0) GO TO 283
      NOTCH(2,1) = NOTCH(2,1)+DX
      NOTCH(3,1) = NOTCH(3,1)+DY
283   IF (NOTCH(1,2) .EQ. 0) GO TO 284
      NOTCH(2,2) = NOTCH(2,2)+DX
      IF (IPLAC .EQ. 1) GO TO 210
      NOTCH(3,2) = NOTCH(3,2)+DY
284   IF (NOTCH(1,3) .EQ. 0) GO TO 285
      NOTCH(2,3) = NOTCH(2,3)+DX
      NOTCH(3,3) = NOTCH(3,3)+DY
285   IF (NOTCH(1,4) .EQ. 0) GO TO 286
      NOTCH(2,4) = NOTCH(2,4)+DX
      NOTCH(3,4) = NOTCH(3,4)+DY
286   OPEN(UNIT=11,FILE='T1.DAT',TYPE='OLD',
     1 DISP='KEEP',FORM='FORMATTED')
      OPEN(UNIT=12,FILE='T2.DAT',TYPE='NEW',
     1 DISP='KEEP',FORM='FORMATTED')
```

```
      DO 281 KK=1,TPLAC
      READ(11,11)TYP,TNP,LBLP
      READ(11,12)XP,YP,RP
      READ(11,13)(PARP(JK),JK=1,10)
      XP=XP+DX
      YP=YP+DY
      PARP(7) = PARP(7) + DY
      PARP(8) = PARP(8) + DY
      PARP(9) = PARP(9) + DX
      PARP(10) = PARP(10) + DX
      WRITE(12,11)TYP,TNP,LBLP
      WRITE(12,12)XP,YP,RP
      WRITE(12,13)(PARP(JK),JK=1,10)
281   CONTINUE
      CLOSE (UNIT=11)
      CLOSE (UNIT=12)
      OPEN(UNIT=11,FILE='T1.DAT',TYPE='NEW',
     1 DISP='KEEP',FORM='FORMATTED')
      OPEN(UNIT=12,FILE='T2.DAT',TYPE='OLD',
     1 DISP='DELETE',FORM='FORMATTED')
      DO 282 KK =1,TPLAC
      READ(12,11)TYP,TNP,LBLP
      READ(12,12)XP,YP,RP
      READ(12,13)(PARP(JK),JK=1,10)
      WRITE(11,11)TYP,TNP,LBLP
      WRITE(11,12)XP,YP,RP
      WRITE(11,13)(PARP(JK),JK=1,10)
282   CONTINUE
      CLOSE(UNIT=11)
      CLOSE(UNIT=12)
      IF (JPLAC .GT. 0) GO TO 500
      GO TO 205
C
C
C     END OF ELEMENT LOOP
C
350   CONTINUE
      WRITE(2,6) SCREF
      WRITE(3,6) SCREF
      OPEN(UNIT=11,FILE='T1.DAT',TYPE='OLD',
     1 DISP='KEEP',FORM='FORMATTED')
      DO 360 KK = 1,NEL
      READ(11,11) TYP,TNP,LBLP
      READ(11,12)XP,YP,RP
      READ(11,13)(PARP(JK),JK=1,10)
      WRITE(2,11)TYP,TNP,LBLP
      WRITE(2,12)XP,YP,RP
      WRITE(2,13)(PARP(JK),JK=1,10)
360   CONTINUE
      CLOSE(UNIT=11)
C
```

```
C
C
C       START BORDERS
C
C
        JPLAC = 1
        TPLAC = NEL
        OPEN(UNIT=11,FILE='T1.DAT',TYPE='OLD',
     1  DISP='KEEP',FORM='FORMATTED')
        READ(11,11)TYP,TNP,LBLP
        READ(11,12)XP,YP,RP
        READ(11,13)(PARP(J),J=1,6),(BORD(IJ),IJ=1,4)
        IF (NEL .EQ. 1) GO TO 499
        DO 475 JJ = 2,NEL
        READ(11,11) TYP,TNP,LBLP
        READ(11,12) XP,YP,RP
        READ(11,13) (PARP(J),J=1,6),(BINT(II),II=1,4)
        IF (BINT(1) .LE. BORD(1)) GO TO 470
        BORD(1) = BINT(1)
  470   IF (BINT(2) .GE. BORD(2)) GO TO 471
        BORD(2) = BINT(2)
  471   IF (BINT(3) .LE. BORD(3)) GO TO 472
        BORD(3) = BINT(3)
  472   IF (BINT(4) .GE. BORD(4)) GO TO 475
        BORD(4) = BINT(4)
  475   CONTINUE
  499   CLOSE(UNIT=11)
C
C
C       BORDER SEGMENT EXCHANGE
C
  500   IF (JPLAC .EQ. 1) GO TO 525
        IF (JPLAC .EQ. 2) GO TO 530
        IF (JPLAC .EQ. 3) GO TO 546
        IF (JPLAC .EQ. 4) GO TO 547
        IF (JPLAC .EQ. 5) GO TO 575
        IF (JPLAC .EQ. 6) GO TO 580
        IF (JPLAC .EQ. 7) GO TO 586
        IF (JPLAC .EQ. 8) GO TO 600
        IF (JPLAC .EQ. 9) GO TO 625
        IF (JPLAC .EQ. 10) GO TO 635
        IF (JPLAC .EQ. 11) GO TO 650
        IF (JPLAC .EQ. 12) GO TO 675
C
C
C       INITIAL BORDER OPTIONS
C
  525   JPLAC=2
        GO TO 220
  530   CALL BORDER(XA,YA,BORD,NOTCH,AREA,SINCH,PO)
        CALL FSPAC
```

```
  531   PRINT *,'SELECT OPTION AND PRESS <RET>.'
        PRINT *,'           1. ACCEPT PRE-NOTCHED BORDERS.'
        PRINT *,'           2. ADJUST PRE-NOTCHED BORDERS.'
        PRINT *,'           3. INCREASE FRAME AREA.'
        READ(*,10) ISEL
        IF (ISEL .EQ. 1) GO TO 535
        IF (ISEL .EQ. 2) GO TO 545
        IF (ISEL .EQ. 3) GO TO 565
        CALL CLEAR
        CALL WARN(IERR)
        PRINT *,'ERROR: MUST INPUT 1, 2, OR 3.'
        CALL WARN(IERR)
        PRINT *
        GO TO 531
  535   JPLAC = 5
        GO TO 500
C
C
C       IDO:  ADJUST BORDER
C
  545   JPLAC=3
        CALL CLEAR
        CALL HDGA(PH,NFAC,PGM)
        CALL TRISPAC
        PRINT *,'SPECIFY WHICH BORDER IS TO BE ADJUSTED.'
        PRINT *
        PRINT *,'           1. NORTH       (TOP)'
        PRINT *,'           2. SOUTH       (BOTTOM)'
        PRINT *,'           3. EAST        (RIGHT)'
        PRINT *,'           4. WEST        (LEFT)'
        READ(*,10)NBADJ
        GO TO 500
  546   JPLAC = 4
        GO TO 220
  547   JPLAC = 1
        CALL BORDER(XA,YA,BORD,NOTCH,AREA,SINCH,PO)
        CALL FSPAC
        IF (NBADJ .GE. 3) GO TO 552
        PRINT *,'SELECT THE DIRECTION THE BORDER IS'
        PRINT *,'TO BE ADJUSTED:'
        PRINT *,'        1. UP'
        PRINT *,'        2. DOWN'
        READ (*,10) DIR
        IF (DIR .EQ. 1) GO TO 548
        IF (DIR .EQ. 2) GO TO 550
        CALL WARN(IERR)
        CALL WARN(IERR)
        GO TO 546
  548   PRINT *,'INPUT THE AMOUNT OF UPWARD ADJUSTMENT'
        PRINT *,'IN REAL INCHES AND PRESS <RET>.'
        READ(*,16)ADJ
```

```
      SADJ=ADJ*SINCH
      BORD(NBADJ)=BORD(NBADJ) + SADJ
      GO TO 500
550   PRINT *,'INPUT THE AMOUNT OF DOWNWARD ADJUSTMENT'
      PRINT *,'IN REAL INCHES AND PRESS <RET>.'
      READ (*,16) ADJ
      SADJ =ADJ * SINCH
      BORD(NBADJ)=BORD(NBADJ)-SADJ
      GO TO 500
552   PRINT *,'SELECT THE DIRECTION THE BORDER IS'
      PRINT *,'TO BE ADJUSTED.'
      PRINT *,'         1. RIGHT.'
      PRINT *,'         2. LEFT.'
      READ (*,10) DIR
      IF(DIR .EQ. 1) GO TO 553
      IF (DIR .EQ. 2) GO TO 554
      CALL WARN(IERR)
      CALL WARN(IERR)
      GO TO 546
553   PRINT *,'INPUT THE AMOUNT OF RIGHT ADJUSTMENT'
      PRINT *,'IN REAL INCHES AND PRESS <RET>.'
      READ (*,16) ADJ
      SADJ = ADJ * SINCH
      BORD(NBADJ) = BORD(NBADJ) + SADJ
      GO TO 500
554   PRINT *,'INPUT THE AMOUNT OF LEFT ADJUSTMENT '
      PRINT *,'IN REAL INCHES AD PRESS <RET>.'
      READ(*,16) ADJ
      SADJ = ADJ * SINCH
      BORD(NBADJ)=BORD(NBADJ)-SADJ
      GO TO 500
C
C
C     IBO:  DOUBLE FRAME
C
565   JPLAC = 1
      GO TO 280
C
C
C     NOTCH OPTIONS    (NO)
C
575   JPLAC = 6
      GO TO 220
580   CALL BORDER(XA,YA,BORD,NOTCH,AREA,SINCH,PO)
      CALL FSPAC
581   PRINT *,'DO YO WISH TO NOTCH A CORNER?'
      PRINT *
      PRINT *,'         1. YES'
      PRINT *,'         2. NO - DELETE A NOTCH'
      PRINT *,'         3. NO - ACCEPT PRESENT BORDERS'
      READ(*,10) NCC
```

```
      IF(NCE .EQ. 1) GO TO 585
      IF (NCE .EQ. 2) GO TO 590
      IF (NCE .EQ. 3) GO TO 595
      CALL CLEAR
      CALL WARN(IERR)
      PRINT *,'ERROR: MUST SELECT 1, 2, OR 3.'
      PRINT *
      GO TO 581
C
C
C     NO: MAKE NOTCH
C
 585  JPLAC = 7
      CALL CLEAR
      CALL HDGA(PH,NFAC,PGM)
      CALL TRISPAC
      PRINT *,'ENTER CORNER NUMBER TO BE NOTCHED'
      CALL DUSPAC
      PRINT *,'         1. NORTHWEST     (UPPER-LEFT)'
      PRINT *,'         2. NORTHEAST     (UPPER-RIGHT)'
      PRINT *,'         3. SOUTHEAST     (LOWER-RIGHT)'
      PRINT *,'         4. SOUTHWEST     (LOWER-LEFT)'
      READ (*,10) NCORN
      GO TO 220
 586  JPLAC = 5
      CALL BORDER(XA,YA,BORD,NOTCH,AREA,SINCH,PO)
      CALL FSPAC
      PRINT *,'ENTER X-COORDINATE OF NOTCHPOINT'
      PRINT*,'AND PRESS <RET>.'
      READ(*,16)NOTCH(2,NCORN)
      PRINT *,'ENTER Y-COORDINATE OF NOTCHPOINT'
      PRINT *,'AND PRESS <RET>.'
      READ(*,16) NOTCH(3,NCORN)
      NOTCH(1,NCORN)=1.0
      GO TO 500
C
C
C     NO:  DELETE NOTCH
C
 590  JPLAC =6
      CALL CLEAR
      CALL HDGA(PH,NFAC,PGM)
      CALL TRISPAC
      PRINT *,'ENTER THE NOTCH YOU DESIRE TO DELETE.'
      CALL DUSPAC
      PRINT *,'         1. NORTHWEST     (UPPER-LEFT)'
      PRINT *,'         2. NORTHEAST     (UPPER-RIGHT)'
      PRINT *,'         3  SOUTHEAST     (LOWER-RIGHT)'
      PRINT *,'         4. SOUTHWEST     (LOWER-LEFT)'
      PRINT *,'         5. NO DELETION'
      READ (*,10) NCORN
```

```
      IF (NCORN .EQ. 5) GO TO 575
      IF (NOTCH(1,NCORN) .EQ. 0.0) GO TO 593
      NOTCH(1,NCORN)= 0.0
      NOTCH(2,NCORN)= 0.0
      NOTCH(3,NCORN)= 0.0
      NOTCH(4,NCORN)= 0.0
      GO TO 500
 593  CALL CLEAR
      CALL WARN(IERR)
      PRINT*,'ERROR: THE CORNER SPECIFIED HAS NO NOTCH.'
      CALL DUSPAC
      CALL WARN(IERR)
      GO TO 581
C
C
C     NO:  FINALIZE BORDERS
C
 595  JPLAC = 8
      CEN(1) = BORD(3) - BORD(4)
      CEN(2) = BORD(1) - BORD(2)
      GO TO 500
C
C
C     POE AND POD OPTIONS   (PO)
C
 600  JPLAC=9
      NBPO(1)=0
      NBPO(2)=0
      PO(1,1)=0.0
      DO 605 KKI = 1,NWCTR
      IF (FT(KKI,I) .GT. 0.0) PO(1,1) = 1.0
 605  CONTINUE
      PO(2,1)=0.0
      DO 606 KKI = 1,NWCTR
      IF (FT(I,KKI) .GT. 0.0) PO(2,1) = 1.0
 606  CONTINUE
C
C
C     PO:   POE/POD LOOP
C
 625  DO 699 NP=1,2
 626  JPLAC = 10
      IF (PO(NP,1) .EQ. 0.0) GO TO 699
      CALL CLEAR
 627  FORMAT(1X,'INDICATE THE BORDER YOU DESIRE TO PLACE
 628  FORMAT(1X,'THE',1X,A3,'.')
 629  PRINT 627
      PRINT 628, POL(NP)
      CALL WALL
      READ(*,10) NB
      NBPO(NP)=NB
```

```
      IF(NB .EQ. 1) GO TO 630
      IF(NB .EQ. 2) GO TO 630
      IF(NB .EQ. 3) GO TO 631
      IF(NB .EQ. 4) GO TO 631
      CALL CLEAR
      CALL WARN(IERR)
      PRINT *,'ERROR: MUST SELECT 1, 2, 3, OR 4'
      CALL WARN(IERR)
      GO TO 629
C
  630 PO(NP,3)=BORD(NB)
```

```
      KP=2
      GO TO 635
  631 PO(NP,2)=BORD(NB)
      KP=3
      GO TO 635
C
  635 JPLAC = 11
      GO TO 220
C
  650 JPLAC = 12
      CALL BORDER(XA,YA,BORD,NOTCH,AREA,SINCH,PO)
      CALL FSPAC
  655 FORMAT(1X,'INDICATE AT WHAT POINT ALONG THE')
  656 FORMAT(1X,A5,1X,'BORDER YOU DESIRE TO PLACE THE')
  657 FORMAT(1X,A3,'.')
  658 FORMAT(5X,'ACCEPTABLE RANGE:',1X,F8.2,1X,'-',1X,F8.2)
  659 PRINT 655
      PRINT 656, WAL(NB)
      PRINT 657, POL(NP)
      IF(NB .GT. 2) GO TO 660
      PRINT 658,BORD(4),BORD(3)
      READ(*,16) PO(NP,KP)
      IF (PO(NP,KP) .GT. BORD(3)) GO TO 665
      IF (PO(NP,KP) .LT. BORD(4)) GO TO 666
      GO TO 220
C
  660 PRINT 658,BORD(2),BORD(1)
      READ (*,16) PO(NP,KP)
      IF (PO(NP,KP) .GT. BORD(1)) GO TO 665
      IF (PO(NP,KP) .LT. BORD(2)) GO TO 666
      GO TO 220
C
  665 CALL CLEAR
      CALL WARN(IERR)
```

```
      PRINT *,'VALUE INPUT EXCEEDS PERMISSABLE RANGE.'
      PRINT *,'TRY AGAIN, YOU JERK'
      CALL DUSPAC
      GO TO 659
C
  666 CALL CLEAR
      CALL WARN(IERR)
      PRINT *,'VALUE INPUT FALLS BELOW PERMISSABLE RANGE.'
      PRINT *,'TRY AGAIN, YOU JERK'
      CALL DUSPAC
      GO TO 659
C
  675 CALL BORDER(XA,YA,BORD,NOTCH,AREA,SINCH,PO)
      CALL FSPAC
  676 PRINT *,'INPUT OPTION AND PRESS <RET>.'
  680 FORMAT(10X,'1. ACCEPT',1X,A3,'.')
  681 FORMAT(10X,'2. ADJUST',1X,A3,'.')
      PRINT 680, POL(NP)
      PRINT 681, POL(NP)
      READ (*,10) ISEL
      IF (ISEL .EQ. 1) GO TO 699
      IF (ISEL .EQ. 2) GO TO 626
      CALL CLEAR
      CALL WARN(IERR)
      PRINT *,'ERROR: MUST SELECT 1 OR 2.'
      CALL WARN(IERR)
      CALL DUSPAC
      GO TO 676
C
C
C     END OF POE/POD LOOP
C
  699 CONTINUE
C
```

```
C
C        STORE BORDER DATA
C
  850    CALL PSTORE(BORD,CEN,PO,NOTCH,NBPO,OLD)
         OPEN(UNIT=11,FILE='T1.DAT',TYPE='OLD',
     1   DISP='DELETE',FORM='FORMATTED')
         WRITE(8,6) NEL
         DO 860 IKP=1,NEL
         READ(11,11) TY,TN,LBL
         WRITE(8,11) TY,TN,LBL
         READ(11,12) X,Y,R
         XPE=X-OLD(1,1)
         YPE=Y-OLD(1,2)
         WRITE(8,12) XPE,YPE,R
         XPE=X-OLD(2,1)
         YPE=Y-OLD(2,2)
         WRITE(8,12) XPE,YPE,R
         READ(11,13) (PAR(JKK),JKK=1,10)
         WRITE(8,13) (PAR(JKK),JKK=1,10)
  860    CONTINUE
         CLOSE(UNIT=11)
         IF (SCREF .GT. MAXSC) MAXSC = SCREF
C
C
C
C
C
C
C
C
C
C
C
C
  100    CONTINUE
         CLOSE(UNIT=8)
         OPEN(UNIT=7,FILE='SCALE.DAT',TYPE='NEW',
     1   DISP='KEEP',FORM='FORMATTED')
         WRITE(7,6) MAXSC
         CLOSE(UNIT=7)
         CALL GRSTOP
         END
```

C. LAYOUT Source Program

```
C
C
C         DECLARATIONS
C
C
          REAL FT(25,25),PF(2,2),OLD(2,2),BF(2,4)
          REAL CF(2,2),NF(2,4,2),PLOT(25,5),A(2)
          REAL WALLS(4),TWALLS(4)
          REAL RVAL(10)
          REAL NCORN(4,2)
          REAL DIST(25,25),BP(2)
          INTEGER FROM,TO,START,SBORD
          INTEGER BCORN(4)
          INTEGER GRPN, MIX(5),PT(2),NPT(4),NBPO(2)
          INTEGER LNBPO(2),PLACE(25),SCALE,PH,PIV
          INTEGER WDIR
          INTEGER NBPOP(2),IPLOT(25,5)
          INTEGER RLIST(10,2)
          INTEGER CLINE
          CHARACTER * 20 NFAC
          CHARACTER * 15 NMGRP(5),NMWTR(25)
          CHARACTER *10 NMPRDT(5)
          CHARACTER * 5 ABNMW(25)
          CHARACTER * 7 PGM(4)
C
C
C         FORMATS
C
C
    1     FORMAT(I2)
    2     FORMAT(F10.3)
    6     FORMAT(1X,I2)
C
C
C         INITILAL VALUES
C
C
          LPLAC=1
          SCALE=6
          NPLAC=0
          PH=3
          IERR=2
          CLINE=0
          NSEL=0
          SINCH=.3585771658
          ADEF=0.0
          FX=300.0
          FY=250.0
          IPE=0
          DO 10 II=1,4
   10     BCORN (II) = 0
C
C
C         INITIALIZATION
C
C
          CALL GRSTRT(4051,1)
          CALL CMCLOS
C
C
C         READ WORK.DAT/FROMTO.DAT
C
C
          CALL RDGEN(NFAC,GRPN,NGRP,NMGRP,NPRDT,NMPRDT,
     1    NWCTR,NMWTR,ABNMW)
          CALL RDFT(NPRDT,NWCTR,FT,MIX)
          DO 75 I = 1,4
          DO 70 J = 1,NWCTR
   70     PLOT(J,I)=0.0
```

```
   75    CONTINUE
C
C
C        EXCHANGE
C
  100    IF (LPLAC .EQ. 1) GO TO 105
         IF (LPLAC .EQ. 2) GO TO 110
         IF (LPLAC .EQ. 3) GO TO 115
         IF (LPLAC .EQ. 4) GO TO 120
         IF (LPLAC .EQ. 5) GO TO 140
         IF (LPLAC .EQ. 6) GO TO 200
         IF (LPLAC .EQ. 7) GO TO 600
         IF (LPPLAC .EQ. 8) GO TO 999
C
C
C        SELECTING FIRST WORKCENTER
C
  105    LPLAC = 2
         CALL HDGA(PH,NFAC,PGM)
         PRINT *
         CALL LIST(NWCTR,NMWTR,PLACE)
         CALL FSPAC
         PRINT *,'ENTER THE NUMBER OF THE WORKCENTER TO BE'
         PRINT *,'PLACED FIRST. NOTE: THIS SHOULD BE THE'
         PRINT *,'DEPARTMENT WITH THE MOST OUTSIDE CONTACT'
         PRINT *,'(e.g. SHIPPING AND/OR RECEIVING)'
         READ(*,1) NSEL
         CALL LOAD(NSEL,BF,CF,PF,NF,NBPO,OLD,PT,NPT)
         CALL NPOINT(PT,NP,W)
         GO TO 100
C
C
  110    CALL FRAME(SCALE,XA,YA,STEP)
         CALL FRAMEA(SCALE,XA,YA,STEP)
         IF (NP .EQ. 1) LPLAC = 3
         IF (NP .EQ. 2) LPLAC = 4
         GO TO 100
C
C
C        INITIAL DRAW ROTATION/ 1 POINT
C
  115    LPLAC = 5
         IF (NBPO(W) .EQ. 1) RTE = 270.
         IF (NBPO(W) .EQ. 2) RTE = 90.
         IF (NBPO(W) .EQ. 3) RTE = 0.0
         IF (NBPO(W) .EQ. 4) RTE = 180.
         PIV = W
         W=0
         GO TO 100
C
C
C        INITIAL DRAW ROTATION/2 POINTS
C
  120    LPLAC = 5
         IF (NBPO(1) .EQ. 3) GO TO 125
         IF (NBPO(2) .EQ. 3) GO TO 126
         IF (NBPO(1) .EQ. 1) GO TO 130
         IF (NBPO(2) .EQ. 1) GO TO 131
         IF (NBPO(1) .EQ. 4) GO TO 135
         IF (NBPO(2) .EQ. 4) GO TO 136
         RTE = 90.
         PIV = 1
         GO TO 100
C
C
```

```
  125 W=2
      PIV=1
      GO TO 127
  126 W=1
      PIV=2
  127 IF(NBPO(W) .EQ. 4) RTE = 90.
      IF(NBPO(W) .NE. 4) RTE = 0.0
      GO TO 100
C
C
  130 W=2
      PIV=1
      GO TO 132
  131 W=1
      PIV=2
  132 IF(NBPO(W) .EQ. 4) RTE = 270.
      IF(NBPO(W) .NE. 4) RTE = 0.0
      GO TO 100
C
C
  135 PIV = 1
      GO TO 137
  136 PIV = 2
  137 RTE = 180.
      GO TO 100
C
C
C     DRAW FIRST WORKCENTER
C
  140 LPLAC = 6
      NPLAC = NPLAC +1
      A(1) = FX
      A(2) = FY
      PLACE(NSEL)=1
      PLOT(NSEL,5) = RTE
      IPLOT(NSEL,1)=PIV
      IPLOT(NSEL,4)=PT(1)
      IPLOT(NSEL,5)=PT(2)
      CALL ACCEPT(RTE,BF,NBPO,PIV,PT,NBPOP,WALLS)
      IPLOT(NSEL,2)=NBPOP(1)
      IPLOT(NSEL,3)=NBPOP(2)
      WALLS(1) = WALLS(1) + FY
      WALLS(2) = WALLS(2) + FY
      WALLS(3) = WALLS(3) + FX
      WALLS(4) = WALLS(4) + FX
      PRINT *,RTE
      IF(PT(1) .EQ. 0) GO TO 150
      IF(PIV .EQ. 1) GO TO 145
      PLOT(NSEL,1)=PF(1,1) +FX
      PLOT(NSEL,2)=PF(1,2) +FY
      GO TO 150
  145 PLOT(NSEL,1)=FX
      PLOT(NSEL,2)=FY
  150 IF (PT(2) .EQ. 0) GO TO 160
      IF (PIV .EQ. 2) GO TO 155
      PLOT(NSEL,3)=PF(2,1)+FX
      PLOT(NSEL,4)=PF(2,2)+FY
      GO TO 160
  155 PLOT(NSEL,3)=FX
      PLOT(NSEL,4)=FY
  160 CALL DWCTR(XA,YA,BF,NF,PF,PT,NPT,A,PIV,RTE,IPE)
      CALL FSPAC
  165 PRINT *,'SELECT OPTION AND PRESS (RET):'
      PRINT *
      PRINT *,'         1. ACCEPT PLACEMENT'
```

```
          PRINT *,'            2. INCREASE FRAME AREA'
          READ(*,1) IOPT
          IF (IOPT .EQ. 1) GO TO 100
          IF (IOPT .EQ. 2) GO TO 175
          CALL CLEAR
          CALL WARN(IERR)
          PRINT *,'ERROR: MUST SELECT ONE OR TWO.'
          CALL DUSPAC
          GO TO 165
C
C
C         DOUBLE FRAME AREA
C
  175     CALL DOUBLE(NPLAC,SCALE,XA,YA,A,PLACE,PLOT,IPLOT,
     1    FX,FY,NSE,WALLS,TWALLS)
          IF (PIV .EQ. 2) GO TO 176
          A(1) = PLOT(NSEL,1)
          A(2) = PLOT(NSEL,2)
          GO TO 177
  176     A(1) = PLOT(NSEL,3)
          A(2) = PLOT(NSEL,4)
  177     CALL CLEAR
          CALL FRAME(SCALE,XA,YA,STEP)
          CALL FRAMEA(SCALE,XA,YA,STEP)
          CALL DWCTR(XA,YA,BF,NF,PF,PT,NPT,A,PIV,RTE,IPE)
          CALL FSPAC
          GO TO 165
C
C
C         MAIN LOOP/ LAYOUT 2+ WORKCENTERS
C
  200     LPLAC = 7
          DO 550 I = 2,NWCTR
          CALL RANK(PLACE,FT,RLIST,RVAL,NWCTR)
          IAA=RLIST(1,1)
          IBB=RLIST(1,2)
          IF (PLACE(IAA) .NE. 1) NLAY = RLIST(1,1)
          IF (PLACE(IBB) .NE. 1) NLAY = RLIST(1,2)
          CALL HDGA(PH,NFAC,PGM)
          CALL RLISTG(PLACE,RLIST,RVAL,NMWTR)
  205     PRINT *
          PRINT *,'FLAG HAS SELECTED THE FIRST-RANKED DEPARTMENT'
          PRINT *,'FOR PLACEMENT IN THE LAYOUT. DO YOU ACCEPT THIS'
          PRINT *,'SELECTION?'
          PRINT *
          PRINT *,'                     1. YES'
          PRINT *,'                     2. NO'
          READ(*,1)IANS
          IF (IANS .EQ. 1) GO TO 250
          IF (IANS .EQ. 2) GO TO 220
          CALL CLEAR
          CALL WARN(IERR)
          PRINT *,'ERROR: MUST SELECT 1 OR 2'
          PRINT *
          CALL RANK(PLACE,FT,RLIST,RVAL,NWCTR)
          CALL RLISTG(PLACE,RLIST,RVAL,NMWTR)
          GO TO 205
C
C
  220     CALL HDGA(PH,NFAC,PGM)
  225     CALL LIST(NWCTR,NMWTR,PLACE)
          PRINT *
          PRINT *,'SELECT WORKCENTER TO BE PLACED NEXT. NOTE:'
          PRINT *,'DEPARTMENTS MARKED WITH X ARE NOT ELIGIBLE'
          PRINT *,'FOR SELECTION.'
```

```
                READ (*,1) NLAYT
                IF (PLACE(NLAYT) .EQ. 0) GO TO 230
                CALL CLEAR
                CALL WARN(IERR)
                PRINT *,'ERROR: THE WORKCENTER SELECTED HAS ALREADY'
                PRINT *,'       BEEN PLACED. TRY AGAIN - I HAVE ALL'
                PRINT *,'       DAY.'
                GO TO 225
C
C
C               SELECT ALTERNATIVE DEPT
C
        230     NLAY = NLAYT
C
C
C
        250     FROM = RLIST(1,1)
                TO   = RLIST(1,2)
                IF (PLACE(FROM) .EQ. 1) START = FROM
                IF (PLACE(TO) .EQ. 1) START = TO
                IF (START .EQ. FROM) PIV = 1
                IF (START .NE. FROM) PIV = 2
                IPLOT(NLAY,1) = PIV
                CALL LOAD(NLAY,BF,CF,PF,NF,NBPO,
           1    OLD,PT,NPT)
                IPLOT(NLAY,4) = PT(1)
                IPLOT(NLAY,5) = PT(2)
                IF (FROM .EQ. START) BP(1) = PLOT(START,3)
                IF (FROM .EQ. START) BP(2) = PLOT(START,4)
                IF (TO .EQ. START) BP(1) = PLOT(START,1)
                IF (TO .EQ. START) BP(2) = PLOT(START,2)
                RGARD= ADEF * SINCH
                IF (FROM .EQ. START) SIV = 2
                IF (TO .EQ. START) SIV =1
                IF (SIV .EQ. 1) SBORD=IPLOT(START,2)
                IF (SIV .EQ. 2) SBORD=IPLOT(START,3)
                IF (SBORD .EQ. 1) GO TO 255
                IF (SBORD .EQ. 2) GO TO 260
                IF (SBORD .EQ. 3) GO TO 265
                A(1) = BP(1) - RGARD
                A(2) = BP(2)
                GO TO 270
C
C
        255     A(1) = BP(1)
                A(2) = BP(2) + RGARD
                GO TO 270
C
C
        260     A(1) =BP(1)
                A(2) =BP(2) -RGARD
                GO TO 270
C
C
        265     A(1) =BP(1)+RGARD
                A(2) = BP(2)
C
C
        270     IF (SIV .EQ. 1) NBS=IPLOT(START,2)
                IF (SIV .EQ. 2) NBS=IPLOT(START,3)
                IF (NBPO(PIV) .EQ. 1) GO TO 275
                IF (NBPO(PIV) .EQ. 2) GO TO 280
                IF (NBPO(PIV) .EQ. 3) GO TO 285
                IF (NBS .EQ. 3) RTE = 0.0
                IF (NBS .EQ. 1) RTE = 90.0
```

```
      IF (NBS .EQ. 4) RTE = 180.0
      IF (NBS .EQ. 2) RTE = 270.0
      GO TO 290
C
C
  275 IF (NBS .EQ. 2) RTE = 0.0
      IF (NBS .EQ. 3) RTE = 90.0
      IF (NBS .EQ. 1) RTE = 180.0
      IF (NBS .EQ. 4) RTE = 270.0
      GO TO 290
C
  280 IF (NBS .EQ. 1) RTE = 0.0
      IF (NBS .EQ. 4) RTE = 90.0
      IF (NBS .EQ. 2) RTE = 180.0
      IF (NBS .EQ. 3) RTE = 270.0
      GO TO 290
C
C
  285 IF (NBS .EQ. 4) RTE = 0.0
      IF (NBS .EQ. 2) RTE = 90.0
      IF (NBS .EQ. 3) RTE = 180.0
      IF (NBS .EQ. 1) RTE = 270.0
C
C
  290 PLOT(NLAY,5) = RTE
      CALL ACCEPT(RTE,BF,NBPO,PIV,PT,NBPOP,TWALLS)
      IPLOT(NLAY,2) = NBPOP(1)
      IPLOT(NLAY,3) = NBPOP(2)
      TWALLS(1) = TWALLS(1) +A(2)
      TWALLS(2) = TWALLS(2) +A(2)
      TWALLS(3) = TWALLS(3) +A(1)
      TWALLS(4) = TWALLS(4) +A(1)
      IF (PT(1) .EQ. 0) GO TO 300
      IF (PIV .EQ. 1) GO TO 295
      PLOT(NLAY,1) = PF(1,1) + A(1)
      PLOT(NLAY,2) = PF(1,2) + A(2)
      GO TO 300
C
C
  295 PLOT(NLAY,1) = A(1)
      PLOT(NLAY,2) = A(2)
C
C
  300 IF (PT(2) .EQ. 0) GO TO 310
      IF (PIV .EQ. 2) GO TO 305
      PLOT(NLAY,3) = PF(2,1) + A(1)
      PLOT(NLAY,4) = PF(2,2) + A(2)
      GO TO 310
C
C.
  305 PLOT(NLAY,3) = A(1)
      PLOT(NLAY,4) = A(2)
C
C
C
C     REDRAW
C
  310 CALL FRAME(SCALE,XA,YA,STEP)
      CALL FRAMEA(SCALE,XA,YA,STEP)
      NPLAC=NPLAC +1
      PLACE(NLAY) = 1
      CALL RDEPT(PLACE,NWCTR,PLOT,IPLOT,XA,YA,IPE)
      CALL OPTIONS(NOPT)
  320 IF(NOPT .EQ. 1) GO TO 325
      IF(NOPT .EQ. 2) GO TO 350
```

```
      IF(NOPT .EQ. 3) GO TO 375
      IF(NOPT .EQ. 4) GO TO 400
      IF(NOPT .EQ. 5) GO TO 425
      IF(NOPT .EQ. 6) GO TO 450
      IF(NOPT .EQ. 7) GO TO 475
      IF(NOPT .EQ. 8) GO TO 500
      CALL CLEAR
      CALL WARN(IERR)
      PRINT *,'ERROR: MUST INPUT 1, 2, 3, 4, 5, 6, 7, OR 8.'
      CALL TRISPAC
      CALL OPTIONS(NOPT)
      GO TO 320
C
C
C     OPTION 1: SHIFT
C
  325 CALL HDGA(PH,NFAC,PGM)
      CALL COST(PLACE,PLOT,BCOST,FT)
      CALL WALL
      CALL FSPAC
  326 PRINT *,'INPUT DIRECTION OF DESIRED SHIFT: '
      PRINT *,'          1. NORTH (TOP)'
      PRINT *,'          2. SOUTH (BOTTOM)'
      CALL TOP
      CALL FSPAC
      PRINT *
      PRINT *,'                                         3. EAST  (RIGHT)'
      PRINT *,'                                         4. WEST  (LEFT)'
      PRINT *,'                              5. CANCEL SHIFT'
      READ (*,1) IDIR
      IF (IDIR .EQ. 1) GO TO 330
      IF (IDIR .EQ. 2) GO TO 330
      IF (IDIR .EQ. 3) GO TO 335
      IF (IDIR .EQ. 4) GO TO 335
      IF (IDIR .EQ. 5) GO TO 450
      CALL CLEAR
      CALL WARN(IERR)
      PRINT *,'ERROR: MUST SELECT 1, 2, 3, 4, OR 5 '
      CALL TRISPAC
      GO TO 326
C
C
  330 CALL FRAME(SCALE,XA,YA,STEP)
      CALL FRAMEA(SCALE,XA,YA,STEP)
      CALL RDEPT(PLACE,NWCTR,PLOT,IPLOT,XA,YA,IPE)
      CALL FSPAC
      IF (NOPT .EQ. 2) GO TO 351
      PRINT *,'INPUT IN INCHES THE SHIFT DESIRED IN THE'
      PRINT *,'PRE - SELECTED DIRECTION.'
      READ (*,2) RSHIFT
      IF (IDIR .GT. 2) GO TO 336
      IF (IDIR .EQ. 2) RSHIFT = RSHIFT * -1.0
      RSHIFT = RSHIFT * SINCH
      IF (IPLOT(NLAY,4) .EQ. 0) GO TO 331
      PLOT(NLAY,2) = PLOT(NLAY,2) + RSHIFT
  331 IF (IPLOT(NLAY,5) .EQ. 0) GO TO 332
      PLOT(NLAY,4) = PLOT(NLAY,4) + RSHIFT
  332 TWALLS(1) = TWALLS(1) + RSHIFT
      TWALLS(2) = TWALLS(2) + RSHIFT
      CLINE=1
      CALL COST(PLACE,PLOT,ACOST,FT)
      RNET=ACOST-BCOST
      GO TO 450
C
C
```

```
      335   GO TO 330
      336   IF (IDIR .EQ. 4) RSHIFT =RSHIFT * -1.0
            RSHIFT = RSHIFT * SINCH
            IF (IPLOT(NLAY,4) .EQ. 0) GO TO 337
            PLOT(NLAY,1) = PLOT(NLAY,1) + RSHIFT
      337   IF (IPLOT(NLAY,5) .EQ. 0) GO TO 338
            PLOT(NLAY,3) = PLOT(NLAY,3) + RSHIFT
      338   TWALLS(3) = TWALLS(3) + RSHIFT
            TWALLS(4) = TWALLS(4) + RSHIFT
            GO TO 450
C
C
C           OPTION 2: ROTATE
C
      350   CALL HDGA(PH,NFAC,PGM)
            CALL TOP
            CALL COST(PLACE,PLOT,BCOST,FT)
            CALL TRISPAC
            CALL TRISPAC
      351   PRINT *,'SELECT DESIRED ROTATION:'
            PRINT *,'          1. 90 DEG.'
            PRINT *,'          2. 180 DEG.'
            CALL TOP
            CALL TRISPAC
            CALL TRISPAC
            PRINT *
            PRINT *,'                                      3. 270 DEG.'
            PRINT *,'                                      4. CANCEL ROTATION.'
            READ (*,1) IRTE
            IF (IRTE .EQ. 1) GO TO 355
            IF (IRTE .EQ. 2) GO TO 358
            IF (IRTE .EQ. 3) GO TO 365
            IF (IRTE .EQ. 4) GO TO 450
            CALL CLEAR
            CALL WARN(IERR)
            PRINT *,'ERROR: MUST SELECT 1, 2, 3, OR 4.'
            GO TO 351
C
C           90 DEGREES
C
      355   IF (PLOT(NLAY,5) .EQ. 270.) RTEP = 0.0
            IF (PLOT(NLAY,5) .NE. 270.) RTEP = PLOT(NLAY,5) + 90.
            GO TO 370
C
C           180 DEGREES
C
      358   IF (PLOT(NLAY,5) .EQ. 180.) RTEP = 0.0
            IF (PLOT(NLAY,5) .EQ. 270.) RTEP = 90.
            IF (PLOT(NLAY,5) .EQ. 0.0) RTEP = 180.
            IF (PLOT(NLAY,5) .EQ. 90.) RTEP = 270.
            GO TO 370
C
C           270 DEGREES
C
      365   IF (PLOT(NLAY,5) .EQ. 0.) RTEP = 270.
            IF (PLOT(NLAY,5) .EQ. 90.) RTEP = 0.0
            IF (PLOT(NLAY,5) .EQ. 180.) RTEP = 90.
            IF (PLOT(NLAY,5) .EQ. 270.) RTEP = 180.
      370   PIV = IPLOT(NLAY,1)
            CALL LOAD(NLAY,BF,CF,PF,NF,NBPO,OLD,PT,NPT)
            CALL ACCEPT(RTEP,BF,NBPO,PIV,PT,NBPOP,TWALLS)
            PLOT(NLAY,5) = RTEP
            IPLOT(NLAY,2) = NBPOP(1)
            IPLOT(NLAY,3) = NBPOP(2)
```

```
      CLINE =1
      CALL  COST(PLACE,PLOT,ACOST,FT)
      RNET=ACOST-BCOST
      GO TO 450
C
C
C     OPTION THREE: POINTS
C
  375 CALL  HDGA(PH,NFAC,PGM)
      CALL  POINTS(PLOT,NMWTR,PLACE)
      CALL  OPTIONS(NOPT)
      GO TO 320
C
C
C     OPTION FOUR: LIST
C
  400 CALL  HDGA(PH,NFAC,PGM)
      CALL  LIST(NWCTR,NMWTR,PLACE)
      CALL  OPTIONS(NOPT)
      GO TO 320
C
C
C     OPTION FIVE: RANKING
C
  425 CALL  HDGA(PH,NFAC,PGM)
      CALL  RANK(PLACE,FT,RLIST,RVAL,NWCTR)
      CALL  RLISTG(PLACE,RLIST,RVAL,NMWTR)
      CALL  OPTIONS(NOPT)
      GO TO 320
C
C
C     OPTION SIX: LAYOUT
C
  450 CALL  FRAME(SCALE,XA,YA,STEP)
      CALL  FRAMEA(SCALE,XA,YA,STEP)
      IF  (CLINE .EQ. 1)  CALL  RCOST(RNET)
      CLINE=0
      CALL  RDEPT(PLACE,NWCTR,PLOT,IPLOT,XA,YA,IPE)
      CALL  OPTIONS(NOPT)
      GO TO 320
C
C
C     OPTION SEVEN: INCREASE FRAME AREA
C
  475 CALL  DOUBLE(NPLAC,SCALE,XA,YA,A,PLACE,PLOT,
     1IPLOT,FX,FY,NSEL,WALLS,TWALLS)
      GO TO 450
C
C
C     OPTION EIGHT: ACCEPT
C
  500 IF  (TWALLS(1)  .GT.  WALLS(1))  WALLS(1)  =  TWALLS(1)
      IF  (TWALLS(2)  .LT.  WALLS(2))  WALLS(2)  =  TWALLS(2)
      IF  (TWALLS(3)  .GT.  WALLS(3))  WALLS(3)  =  TWALLS(3)
      IF  (TWALLS(4)  .LT.  WALLS(4))  WALLS(4)  =  TWALLS(4)
  550 CONTINUE
C
C
C     BEGIN WALLS
C
  600 LPLAC  =  8
      IPE  =  1
      CALL  FRAME(SCALE,XA,YA,STEP)
      CALL  FRAMEA(SCALE,XA,YA,STEP)
      CALL  RDEPT(PLACE,NWCTR,PLOT,IPLOT,XA,YA,IPE)
```

```
      CALL DWALLS(XA,YA,WALLS,NCORN,BCORN)
      CALL FSPAC
      PRINT *,'SELECT OPTION AND PRESS (RET):'
      PRINT *
      PRINT *,'          1. ACCEPT PRE-NOTCHED WALLS'
      PRINT *,'          2. ADJUST PRE-NOTCHED WALLS'
      READ (*,1) IDS
      IF (IDS .EQ. 1) GO TO 650
      IF (IDS .EQ. 2) GO TO 625
625   CALL CLEAR
      CALL HDGA(PH,NFAC,PGM)
      CALL TRISPAC
      CALL WALL
      READ (*,1) NWADJ
      CALL FRAME(SCALE,XA,YA,STEP)
      CALL FRAMEA(SCALEA,XA,YA,STEP)
      CALL RDEPT(PLACE,NWCTR,PLOT,IPLOT,XA,YA,IPE)
      CALL DWALLS(XA,YA,WALLS,NCORN,BCORN)
      CALL FSPAC
      PRINT *,'SELECT THE DIRECTION THE WALL IS TO '
      PRINT *,'ADJUSTED'
      IF (NWADJ .GT.2) GO TO 630
      PRINT *,' 1.UP'
      PRINT *,' 2. DOWN'
      READ (*,1) WDIR
      IF (WDIR .EQ. 1) FACT = 1.0
      IF (WDIR .EQ. 2) FACT = -1.0
      GO TO 635
630   PRINT *,'          1.RIGHT'
      PRINT *,'          2. LEFT'
      READ (*,1) WDIR
      IF (WDIR .EQ. 1) FACT = 1.0
      IF (WDIR .EQ. 2) FACT = -1.0
635   CALL FRAME(SCALE,XA,YA,STEP)
      CALL FRAMEA(SCALE,XA,YA,STEP)
      CALL RDEPT(PLACE,NWCTR,PLOT,IPLOT,XA,YA,IPE)
      CALL DWALLS(XA,YA,WALLS,NCORN,BCORN)
      CALL FSPAC
      PRINT  *,'INPUT THE NUMBER OF INCHES TO ADJUST.'
      READ (*,2) ADJ
      ADJ = ADJ*FACT
      WALLS(NWADJ)=WALLS(NWADJ) + ADJ
      GO TO 600
650   LPLAC = 9
      CALL FRAME(SCALE,XA,YA,STEP)
      CALL FRAMEA(SCALE,XA,YA,STEP)
      CALL RDEPT(PLACE,NWCTR,PLOT,IPLOT,XA,YA,IPE)
      CALL DWALLS(XA,YA,WALLS,NCORN,BCORN)
      CALL FSPAC
      PRINT *,'DO YO WISH TO NOTCH A BORDER?'
      PRINT *,'          1. YES'
      PRINT *,'          2. NO'
      READ (*,1) IOPT
      IF (IOPT .EQ. 2) GO TO 999
      CALL HDGA(PH,NFAC,PGM)
      CALL TRISPAC
      PRINT *,'INPUT CORNER NUMBER AND PRESS (RET).'
      PRINT *,'          1. NORTHWEST  -  UPPER LEFT'
      PRINT *,'          2. NORTHEAST  -  UPPER RIGHT'
      PRINT *,'          3. SOUTHEAST  - LOWER RIGHT'
      PRINT *,'          4. SOUTHWEST  -  LOWER LEFT'
      READ(*,1) IC
      CALL FRAME(SCALE,XA,YA,STEP)
      CALL FRAMEA(SCALE,XA,YA,STEP)
      CALL RDEPT(PLACE,NWCTR,PLOT,IPLOT,XA,YA,IPE)
```

```
      CALL DWALLS(XA,YA,WALLS,NCORN,BCORN)
      CALL FSPAC
      PRINT *,'INPUT X AND Y COORDINATES OF NOTHPOINT'
      PRINT *,'SEPARATED BY A COMMA.'
      READ *,NCORN(IC,1),NCORN(IC,2)
      BCORN(IC) = 1
      GO TO 650
C
C
C
C
C
C
C
C
C
C
C
999   CALL SLAYT(NWCTR,PLOT,IPLOT,NCORN,BCORN,WALLS)
      OPEN(UNIT=7,FILE='SCALE.DAT',TYPE='OLD',
     1 DISP='KEEP',FORM='FORMATTED')
      READ (7,6) IFEE
      WRITE(7,6) SCALE
      CLOSE(UNIT=7)
      END
```

D. OUTPUT Source Program

```
C
C
C       DECLARATIONS
C
        REAL PF(2,2),OLD(2,2),BF(2,4),CF(2,2)
        REAL NCORN(4,2)
        REAL NF(2,4,2),PAR(10),PLOT(25,5)
        REAL A(2),WALLS(4)
        INTEGER GRPN,PH,DEVICE,CODE,SCALE,SCREF
        INTEGER PLACE(25)
        INTEGER BCORN(4)
        INTEGER PT(2),NPT(2),NBPO(2),TN,TY,IPLOT(25,5)
        INTEGER PIV
        CHARACTER * 20  NFAC
        CHARACTER * 15  NMGRP(5),NMWTR(25)
        CHARACTER * 10  NMPRDT(5),LBL
        CHARACTER * 5   ABNMW(25)
        CHARACTER * 7   PGM(4)
        CHARACTER * 79  Z
C
C
C       INITIAL VALUES
C
        PH=4
        SINCH =.3585771658
        IERR=2
        IPE = 1
C
C
C       FORMATS
C
    1   FORMAT(I4)
    2   FORMAT(1X,I2)
    3   FORMAT(1X,I2,5X,A15,5X,I2)
    4   FORMAT(1X,I2)
    5   FORMAT(1X,I1,5X,I2,5X,A10)
    6   FORMAT(1X,2(F10.3,5X),F10.3)
    7   FORMAT(1X,10F10.3)
    8   FORMAT(1X,A79)
C
C
C       PGM INITIALIZATION
C
        CALL RDGEN(NFAC,GRPN,NGRP,NMGRP,NPRDT,NMPRDT,
     1  NWCTR,NMWTR,ABNMW)
        CALL RELAYT(NWCTR,PLOT,IPLOT,NCORN,BCORN,WALLS)
        CALL GRSTRT(4051,1)
        CALL CMCLOS
```

```
C
C
C       OUTPUT MENU
C
   50   CALL HDGA(PH,NFAC,PGM)
   55   CALL TRISPAC
        PRINT *,'                             OUTPUT MENU'
        CALL DUSPAC
        PRINT *,'          1. MACHINE TEMPLATE'
        PRINT *,'          2. WORKCENTER LAYOUT'
        PRINT *,'          3. PLANT LAYOUT'
        PRINT *,'          4. EXIT OUTPUT'
        READ (*,1) IOPT
        IF (IOPT .EQ. 1) GO TO 400
        IF (IOPT .EQ. 2) GO TO 200
        IF (IOPT .EQ. 3) GO TO 300
        IF (IOPT .EQ. 4) GO TO 999
        CALL CLEAR
        CALL WARN(IERR)
        PRINT *,'ERROR: MUST SELECT 1, 2, 3, OR 4:'
        GO TO 55
C
C
C       WORHCENTER LAYOUTS
C
  199   READ (*,1) IDS
  200   CALL HDGA(PH,NFAC,PGM)
        CALL TRISPAC
        PRINT *,'                             WORKCENTER LAYOUTS'
        CALL DUSPAC
        PRINT *,'READ ALL INSTRUCTIONS BEFORE DOING ANYTHING'
        PRINT *
        PRINT *,'          TO MAKE WORKCENTER LAYOUT:'
        PRINT *
        PRINT *,'                    A. ENTER WORKCENTER NUMBER.'
        PRINT *,'                    B. WAIT UNTIL DRAWING ON DISPLAY'
        PRINT *,'                       IS COMPLETED.'
        PRINT *,'                    C. PRESS MAKE COPY.'
        PRINT *,'                    D. AFTER HARDCOPY COMPLETE INPUT'
        PRINT *,'                       ANY DIGIT AND PRESS <RET>.'
        CALL DUSPAC
        PRINT *,'          TO RETURN TO OUTPUT MENU INPUT 99'
        PRINT *,'                    AND PRESS <RET>.'
        READ (*,1) NW
        IF (NW .GT. NWCTR) GO TO 50
        OPEN(UNIT=7,FILE='SCALE.DAT',TYPE='OLD',
     1  DISP='KEEP',FORM='FORMATTED')
        READ (7,2) MSCALE
        CLOSE(UNIT=7)
        CALL FRAMEW(MSCALE,XA,YA,STEP)
        SCALE=MSCALE
```

```
      OPEN(UNIT=8,FILE='WORKOUT.DAT',TYPE='OLD',
     1 DISP='KEEP',FORM='FORMATTED')
      PIV=IPLOT(NW,1)
      CALL LOAD(NW,BF,CF,PF,NF,NBPO,OLD,PT,NPT)
      A(1)=OLD(PIV,1)
      A(2)=OLD(PIV,2)
      IF (NW .EQ. 1) GO TO 210
      NK=NW-1
      DO 205 IKK = 1,NK
      READ(8,4) NELP
      DO 204 IKA = 1,NELP
      DO 203 IKB = 1,4
203   READ(8,8) Z
204   CONTINUE
205   CONTINUE
210   READ(8,4) NEL
      DO 230 ILL=1,NEL
      READ (8,5) TY,TN,LBL
      IF (PIV .EQ. 2) GO TO 220
      READ (8,6) X,Y,R
      READ (8,8) Z
      GO TO 225
220   READ (8,8) Z
      READ (8,6) X,Y,R
225   READ (8,7) (PAR(IAS),IAS=1,10)
      CALL TRANSL(A(1),A(2))
      CALL RDRAWO(XA,YA,TY,TN,X,Y,R,PAR,LBL,SINCH)
230   CONTINUE
      CLOSE(UNIT=8)
      RTE=0.0
      CALL TRANSL(A(1),A(2))
      CALL DWCTRO(XA,YA,BF,NF,PF,PT,NPT,A,PIV,RTE,IPE)
      READ *,IDS
      GO TO 199
C
C
300   CALL HDGA(PH,NFAC,PGM)
      CALL DUSPAC
      CALL CMOPEN
      CALL GRSTOP
      PRINT *,'                              PLANT LAYOUT'
      CALL DUSPAC
      PRINT *,'INPUT FOUR DIGIT DEVICE NUMBER'
      READ (*,1) DEVICE
      CALL DUSPAC
      PRINT *,'INPUT OPTION CODE'
      READ (*,1) CODE
      CALL GRSTRT(DEVICE,CODE)
      CALL CMCLOS
      OPEN(UNIT=7,FILE='SCALE.DAT',TYPE='OLD',
     1 DISP='KEEP',FORM='FORMATTED')
```

```
      OPEN(UNIT=8,FILE='WORKOUT.DAT',TYPE='OLD',
     1  DISP='KEEP',FORM='FORMATTED')
      READ(7,2) SCALE
      READ(7,2) SCALE
      CLOSE(UNIT=7)
      DO 305 IJK=1,25
305   PLACE(IJK)=1
      CALL FRAMEW(SCALE,XA,YA,STEP)
      DO 350 IJK =1,NWCTR
      RTEC=PLOT(IJK,5)
      IPIV=IPLOT(IJK,1)
      CALL LOAD(IJK,BF,CF,PF,NF,NBPO,OLD,
     1  PT,NPT)
      IF (IPIV .EQ. 2) GO TO 310
      A(1)=PLOT(IJK,1)
      A(2)=PLOT(IJK,2)
      GO TO 315
310   A(1)=PLOT(IJK,3)
      A(2)=PLOT(IJK,4)
315   CALL TRANSL(A(1),A(2))
      CALL ROTATE(RTEC,RTEC)
      CALL DWCTRO(XA,YA,BF,NF,PF,PT,NPT,A,IPIV,RTEC,IPE)
      READ(8,4) NEL
      DO 340 ILL=1,NEL
      READ(8,5) TY,TN,LBL
      IF (IPIV .EQ. 2) GO TO 320
      READ (8,6) X,Y,R
      READ (8,8) Z
      GO TO 325
320   READ (8,8) Z
      READ (8,6) X,Y,R
325   READ (8,7) (PAR(ILP),ILP=1,10)
      CALL TRANSL(A(1),A(2))
      CALL ROTATE(RTEC,RTEC)
      CALL RDRAWO(XA,YA,TY,TN,X,Y,R,PAR,LBL,SINCH)
340   CONTINUE
350   CONTINUE
      CALL DWALLSO(XA,YA,WALLS,NCORN,BCORN)
      READ (*,1)IDS
      CLOSE(UNIT=8)
      GO TO 50
C
C
C
C
400   CALL HDGA(PH,NFAC,PGM)
      IPH=1
      CALL TRISPAC
      PRINT *,'                          TEMPLATES'
      CALL DUSPAC
      PRINT *,'SELECT OPTION AND PRESS <RET>:'
```

```
      CALL DUSPAC
      PRINT *,'                    1. GENERAL SHAPE TEMPLATE'
      PRINT *,'                    2. PLANPRINT MACHINE TEMPLATE'
      PRINT *,'                    3. RETURN TO OUTPUT MENU'
      READ *,IOPT
      TY = IOPT
      IF (IOPT .EQ. 1) GO TO 425
      IF (IOPT .EQ. 2) GO TO 450
      GO TO 50
C
C
  425 CALL MENUA(IPH)
      READ *,TN
      IF (TN .EQ. 1) GO TQ 426
      IF(TN .EQ. 2) CALL RECTA(PAR,LBL)
      IF(TN .EQ. 3) CALL DRECTA(PAR,LBL)
CR
CR
      GO TO 500
  426 CALL SQUAREA(PAR,LBL)
      GO TO 500
C
C
C
  450 CALL MENUB(IPH)
      READ *,TN
      GO TO 500
C
C
C
  500 PH=4
      X=65.0
      Y=50.0
      R=0.0
      SCALE = 1
      CALL FRAMEW(SCALE,XA,YA,STEP)
      CALL RDRAWO(XA,YA,TY,TN,X,Y,R,PAR,LBL,SINCH)
      CALL FSPAC
      CALL TRISPAC
      PRINT *,LBL
      PRINT *,'INPUT ANY DIGIT TO CLEAR DISPLAY'
      READ *,IJK
      GO TO 400
C
C
C
  999 CALL CMOPEN
      CALL GRSTOP
      END
$
```

E. FLAG Subroutines

```
      SUBROUTINE BORDER(XA,YA,B,N,A,S,P)
      REAL N(4,4),B(4),A(3)
      REAL P(2,3)
      CALL CMOPEN
      CALL VWPORT(39.0,130.0,30.0,100.0)
      CALL WINDOW(0.0,XA,0.0,YA)
C
C     CORNER ONE  (NW)
C
      IF (N(1,1) .EQ. 1.0) GO TO 10
      CALL MOVE(B(4),B(1))
      GO TO 20
   10 CALL MOVE(N(2,1),N(3,1))
      CALL DRAW(N(2,1),B(1))
C
C     CORNER TWO   (NE)
C
   20 IF(N(1,2) .EQ. 1.0) GO TO 30
      CALL DRAW(B(3),B(1))
      GO TO 40
   30 CALL DRAW(N(2,2),B(1))
      CALL DRAW(N(2,2),N(3,2))
      CALL DRAW(B(3),N(3,2))
C
C     CORNER THREE   (SE)
C
   40 IF (N(1,3) .EQ. 1.0) GO TO 50
      CALL DRAW(B(3),B(2))
      GO TO 60
   50 CALL DRAW(B(3),N(3,3))
      CALL DRAW(N(2,3),N(3,3))
      CALL DRAW(N(2,3),B(2))
C
C     CORNER FOUR   (SW)
C
   60 IF (N(1,4) .EQ. 1.0) GO TO 70
      CALL DRAW(B(4),B(2))
      GO TO 80
   70 CALL DRAW(N(2,4),B(2))
      CALL DRAW(N(2,4),N(3,4))
      CALL DRAW(B(4),N(3,4))
C
C     CLOSING THE BORDER
C
```

```
   80   IF (N(1,1) .EQ. 1.0) GO TO 90
        CALL DRAW(B(4),B(1))
        GO TO 100
   90   CALL DRAW(B(4),N(3,1))
        CALL DRAW(N(2,1),N(3,1))
C
C       DASHLINE
C
  100   CALL DASHPT(3)
        CALL MOVE(B(4),B(1))
        CALL DRAW(B(3),B(1))
        CALL DRAW(B(3),B(2))
        CALL DRAW(B(4),B(2))
        CALL DRAW(B(4),B(1))
        CALL DASHPT(0)
C
C
C       POINTS
C
        IF (P(1,1) .EQ. 0.0) GO TO 200
        AN=1.0
        CALL CROSS(P(1,2),P(1,3),AN)
  200   IF (P(1,2) .EQ. 0.0) GO TO 210
        AN=2.0
        CALL CROSS(P(2,2),P(2,3),AN)
C
C       CALCULATE FULLBLOCK AREA
C
  210   ZW=B(1)-B(2)
        ZL=B(3)-B(4)
        CALL BAREA(ZW,ZL,S,ZA)
        A(2)=ZA
C
C       SUBTRACT NOTCH AREAS
C
        IF (N(1,1) .EQ. 0.0) GO TO 110
        ZW=B(1)-N(3,1)
        ZL=N(2,1)-B(4)
        CALL BAREA(ZW,ZL,S,ZA)
        N(4,1)=ZA
        GO TO 115
  110   N(3,1)=0.0
        N(4,1)=0.0
        N(2,1)=0.0
  115   IF(N(1,2) .EQ. 0.0) GO TO 120
        ZW =B(1)-N(3,2)
        ZL =B(3)-N(2,2)
        CALL BAREA(ZW,ZL,S,ZA)
        N(4,2)=ZA
        GO TO 125
  120   N(4,2)=0.0
```

```
      N(3,2)=0.0
      N(2,2)=0.0
125   IF (N(1,3) .EQ. 0.0) GO TO 130
      ZW=N(3,3)-B(2)
      ZL=B(3)-N(2,3)
      CALL BAREA(ZW,ZL,S,ZA)
      N(4,3)=ZA
      GO TO 135
130   N(4,3)=0.0
      N(3,3)=0.0
      N(2,3)=0.0
135   IF (N(1,4) .EQ. 0.0) GO TO 140
      ZW=N(3,4)-B(2)
      ZL=N(2,4)-B(4)
      CALL BAREA(ZW,ZL,S,ZA)
      N(4,4)=ZA
      GO TO 145
140   N(4,4)=0.0
      N(3,4)=0.0
      N(2,4)=0.0
145   DO 150  JKJ = 1,4
150   A(2)=A(2)-N(4,JKJ)
      CALL CMCLOS
      RETURN
      END
      SUBROUTINE COST(PLACE,PLOT,CST,FT)
      REAL PLOT(25,5),FT(25,25)
      INTEGER PLACE(25)
      REAL INDIST,INDISTX,INDISTY,INCOST
      CST=0
      DO 100 IFIR=1,25.
      DO 50  ISEC=1,25
      IF (PLACE(IFIR) .EQ. 0) GO TO 100
      IF (FT(IFIR,ISEC) .EQ. 0.0) GO TO 50
      INDISTX=ABS(PLOT(IFIR,3)-PLOT(ISEC,1))**2
      INDISTY=ABS(PLOT(IFIR,4)-PLOT(ISEC,2))**2
      INDIST=SQRT(INDISTX + INDISTY)/.3585771638
      INCOST=INDIST*(FT(IFIR,ISEC)/12.0)
      CST=CST+INCOST
50    CONTINUE
100   CONTINUE
      RETURN
      END
      SUBROUTINE FRAME(SCREF,XA,YA,STEP)
      INTEGER SCREF
      CALL CMOPEN
      CALL NEWPAG
      XA=130.0*SCREF
      YA=100.0*SCREF
      CALL VWPORT(30.0,130.0,30.0,100.0)
      CALL WINDOW(0.0,XA,0.0,YA)
```

```
      CALL MOVE(0.0,0.0)
      CALL DRAW(0.0,YA)
      CALL DRAW(XA,YA)
      CALL DRAW(XA,0.0)
      CALL DRAW(0.0,0.0)
      CALL DASHPT(9)
      STEP=25.0+(25.0*SCREF)
      YLIN=0.0
10    YLIN=YLIN+STEP
      IF (YLIN .GE. YA) GO TO 20
      CALL MOVE(0.0,YLIN)
      CALL DRAW(XA,YLIN)
      GO TO 10
20    XLIN=0.0
30    XLIN=XLIN+STEP
      IF (XLIN .GE. XA) GO TO 40
      CALL MOVE(XLIN,0.0)
      CALL DRAW(XLIN,YA)
      GO TO 30
40    CALL DASHPT(0)
      IF (SCREF .LT. 6) REF = 120.0 * .35875771658
      IF (SCREF .GE. 6) REF = 300.0 * .35875771658
      AXE = .03 * XA
      AYE = .05 * YA
      AYP = AYE + REF
      AXM = .01 * XA
      AYM = .01 * YA
      CALL MOVE(AXE,AYE)
      CALL DRAW(AXE,AYP)
      CALL MOVE(AXM,AYM)
      CALL TXAM
      CALL TEXT(1,'R')
      CALL CMCLOS
      RETURN
      END
      SUBROUTINE FRAMEA(SCREF,XA,YA,STEP)
      INTEGER SCREF
      CALL TOP
      PRINT *,'AXIS RANGES:'
      PRINT *
      PRINT *,'  X AXIS:'
 5    FORMAT(3X,'  0.0 TO',1X,F6.1)
 6    FORMAT(3X,'INCREMENTS',1X,F5.1)
      PRINT 5, XA
      PRINT *,'  Y AXIS:'
      PRINT 5, YA
      PRINT *
      PRINT *,'DOTTED LINE'
      PRINT 6, STEP
      PRINT *
      PRINT 7,SCREF
```

```
  7     FORMAT(3X,'SCALE NO.:',1X,I2)
        PRINT *
        IF (SCREF .GE. 6) GO TO 10
        PRINT 8
  8     FORMAT(3X,'R LINE = 10 FT.')
        GO TO 99
 10     PRINT 9
  9     FORMAT(3X,'R LINE = 25 FT.')
 99     RETURN
        END
        SUBROUTINE FRAMEW(SCREF,XA,YA,STEP)
        INTEGER SCREF
        CALL CMOPEN
        CALL NEWPAG
        XA=130.0*SCREF
        YA=100.0*SCREF
        CALL VWPORT(0.0,130.0,0.0,100.0)
        CALL WINDOW(0.0,XA,0.0,YA)
        CALL MOVE(0.0,0.0)
        CALL DRAW(0.0,YA)
        CALL DRAW(XA,YA)
        CALL DRAW(XA,0.0)
        CALL DRAW(0.0,0.0)
        STEP=25.0+(25.0*SCREF)
        CALL CMCLOS
        RETURN
        END
%TYPE-W-OPENIN, error opening DISKDOE:[ED01602JJHEAD
-RMS-E-FNF, file not found
        SUBROUTINE LIST(NWCTR,NMWTR,PLACE)
        INTEGER PLACE(25)
        CHARACTER *15 NMWTR(25)
  1     FORMAT(30X,'WORKCENTER LISTING')
  2     FORMAT(27X,'X = WORKCENTER IS PLACED')
  3     FORMAT(1X,I2,'.',1X,A15,3X,'X')
 14     FORMAT(1X,I2,'.',1X,A15)
  5     FORMAT(29X,I2,'.',A15,3X,'X')
  6     FORMAT(29X,I2,'.',A15)
        PRINT *
        PRINT 1
        PRINT *
        PRINT 2
        CALL BUSPAC
        DO 100 KAB=1,20
        IF (KAB .GT. NWCTR) GO TO 300
        IF (PLACE(KAB) .EQ. 1) GO TO 50
        PRINT 14,KAB,NMWTR(KAB)
        GO TO 100
 50     PRINT 3,KAB,NMWTR(KAB)
100     CONTINUE
        IF (NWCTR .LE. 20) GO TO 300
```

```
      CALL TOP
      CALL TRISPAC
      CALL DUSPAC
      DO 200 KAB=21,NWCTR
      IF (PLACE(KAB) .EQ. 1) GO TO 150
      PRINT 6,KAB,NMWTR(KAB)
      GO TO 200
150   PRINT 5,KAB,NMWTR(KAB)
200   CONTINUE
300   RETURN
      END
      SUBROUTINE MENUA(PGM)
      CHARACTER * 20 ENAM(4)
      INTEGER PGM
      CALL CMOPEN
 21   FORMAT(5X,I1,'.',2X,A20)
 16   FORMAT(1X,' ')
      ENAM(1) = 'SQUARE'
      ENAM(2) = 'RECTANGLE'
      ENAM(3) = 'DOUBLE RECTANGLE'
      ENAM(4) = 'CIRCLE'
      IF (PGM .EQ. 4) GO TO 10
      CALL CLEAR
      PRINT 15
      CALL TRISPAC
      DO 20 KK = 1,4
 20   PRINT 21, KK,ENAM(KK)
      CALL DUSPAC
      PRINT *,'SELECT THE DESIRED TEMPLATE NUMBER'
      PRINT *,'      AND PRESS <RETURN>.'
      GO TO 50
 10   OPEN(UNIT=10,FILE='FLAG010.DAT',TYPE='NEW',
  1   DISP='KEEP',FORM='FORMATTED')
 15   FORMAT(10X,'GENERAL SHAPE TEMPLATES')
      WRITE (10,15)
      WRITE (10,16)
      WRITE (10,16)
      DO 30 KK =1,4
 30   WRITE (10,21) KK,ENAM(KK)
      WRITE (10,16)
      WRITE (10,16)
      CLOSE(UNIT=10)
 50   CALL CMCLOS
      RETURN
      END
      SUBROUTINE MENUB(PGM)
      INTEGER PGM
      CHARACTER * 1 ZZ
      CHARACTER * 25 ENAMB(40)
 10   FORMAT(14X,'MACHINE TEMPLATES')
 11   FORMAT(1X,I2,'.',2X,A25)
```

```
12    FORMAT(5X,I2,'.',2X,A25)
13    FORMAT(1X,' ')
14    FORMAT(1X,A7)
      ENAMB(1)='PUNCH PRESS 22TN'
      ENAMB(2)='PUNCH PRESS 40TN'
      ENAMB(3)='PUNCH PRESS 65TN'
      ENAMB(4)='PUNCH PRESS 72TN'
      ENAMB(5)='PUNCH PRESS 90TN'
      ENAMB(6)='PUNCH PRESS 110TN'
      ENAMB(7)='AUTOMATIC LATHE'
      ENAMB(8)='LATHE'
      ENAMB(9)='BENCH LATHE'
      ENAMB(10)='ENG. LATHE'
      ENAMB(11)='TURRET LATHE'
      ENAMB(12)='SHEAR'
      IF (PGM .EQ. 4) GO TO 25
      CALL CLEAR
      PRINT 10
      CALL DUSPAC
      DO 20 KK = 1,12
      PRINT 12,KK,ENAMB(KK)
20    CONTINUE
      CALL DUSPAC
      PRINT *,'SELECT THE DESIRED TEMPLATE NUMBER'
      PRINT *,'      AND PRESS <RETURN>'
      GO TO 50
25    OPEN(UNIT=10,FILE='FLAG010.DAT',TYPE='OLD',
  1   DISP='KEEP',FORM='FORMATTED')
      DO 26 KK = 1,9
26    READ(10,14) ZZ
      WRITE (10,10)
      WRITE (10,13)
      WRITE (10,13)
      DO  28 KK =1,7
28    WRITE (10,11)KK,ENAMB(KK)
      CLOSE(UNIT=10)
50    RETURN
      END
      SUBROUTINE OPTIONS(NOPT)
 1    FORMAT(I1)
      CALL FSPAC
      PRINT *,'SELECT OPTION AND PRESS RETURN'
      PRINT *,'        1. SHIFT'
      PRINT *,'        2. ROTATE'
      PRINT *,'        3. POINTS'
      PRINT *,'        4. LIST'
      CALL TOP
      CALL FSPAC
      PRINT *
      PRINT *,'                                5. RANKING'
      PRINT *,'                                6. LAYOUT'
```

```
      PRINT *,                                    7. INCREASE AREA'
      PRINT *,'                                   8. ACCEPT'
      READ (*,1) NOPT
      RETURN
      END
      SUBROUTINE RCOST(RNET)
      CALL TOP
      DO 10 ITIR=1,15
      PRINT *
 10   CONTINUE
      PRINT 9
      PRINT 11,RNET
 11   FORMAT(5X,F10.2)
 9    FORMAT(3X,'NET COST:')
      RETURN
      END
      SUBROUTINE RDEPT(PLACE,NWCTR,PLOT,IPLOT,XA,YA,IFC)
      REAL PLOT(25,5),AC(2),BFC(2,4),CFC(2,2)
      REAL PFC(2,2),NFC(2,4,2),OLDC(2,2)
      INTEGER PLACE(25),IPLOT(25,5),PIVC
      INTEGER NBPOC(2),PTC(2),NPTC(4)
      DO 50 KAA = 1,NWCTR
      IF (PLACE(KAA) .EQ. 0) GO TO 50
      RTEC = PLOT(KAA,5)
      PIVC = IPLOT(KAA,1)
      IF (PIVC .EQ. 2) GO TO 15
      AC(1) = PLOT(KAA,1)
      AC(2) = PLOT(KAA,2)
      GO TO 20
 15   AC(1) = PLOT(KAA,3)
      AC(2) = PLOT(KAA,4)
 20   CALL LOAD(KAA,BFC,CFC,PFC,NFC,NBPOC,
     1 OLDC,PTC,NPTC)
      CALL DWCTR(XA,YA,BFC,NFC,PFC,PTC,NPTC,AC,PIVC,RTEC,IFC)
 50   CONTINUE
      RETURN
      END
      SUBROUTINE RDRAW(XA,YA,TYR,TNR,XR,YR,RR,PARR,LBLR,SINCH)
      INTEGER TYR,TNR
      REAL PARR(10)
      CHARACTER *10 LBLR
      IF (TYR .EQ. 2) GO TO 100
      IF (TNR .EQ. 1) GO TO 110
      IF (TNR .EQ. 2) GO TO 120
      IF (TNR .EQ. 3) GO TO 130
      IF (TNR .EQ. 4) GO TO 140
 100  IF (TNR .EQ. 7) GO TO 260
      IF (TNR .EQ. 1) GO TO 261
      IF(TNR .EQ. 12) GO TO 262
      IF (TNR .EQ. 4) GO TO 263
```

```
C
C        GEN SHAPE TEMPS DRAWN
C
 110     CALL SQUARE(XA,YA,XR,YR,RR,PARR,LBLR,SINCH)
         GO TO 999
 120     CALL RECT(XA,YA,XR,YR,RR,PARR,LBLR,SINCH)
         GO TO 999
 130     CALL DRECT(XA,YA,XR,YR,RR,PARR,LBLR,SINCH)
         GO TO 999
 140     GO TO 999
 260     CALL AULTH(XA,YA,XR,YR,RR,PARR,LBLR,SINCH)
         GO TO 999
 261     CALL PP22TN(XA,YA,XR,YR,RR,PARR,LBLR,SINCH)
         GO TO 999
 262     CALL SHEAR(XA,YA,XR,YR,RR,PARR,LBLR,SINCH)
         GO TO 999
 263     CALL PP72TN(XA,YA,XR,YR,RR,PARR,LBLR,SINCH)
         GO TO 999
 999     RETURN
         END
         SUBROUTINE RDRAWO(XA,YA,TYR,TNR,XR,YR,RR,PARR,LBLR,SINCH)
         INTEGER TYR,TNR
         REAL PARR(10)
         CHARACTER *10 LBLR
         IF (TYR .EQ. 2) GO TO 100
         IF (TNR .EQ. 1) GO TO 110
         IF (TNR .EQ. 2) GO TO 120
         IF (TNR .EQ. 3) GO TO 130
         IF (TNR .EQ. 4) GO TO 140
 100     IF (TNR .EQ. 7) GO TO 260
         IF (TNR .EQ. 1) GO TO 261
         IF(TNR .EQ. 12) GO TO 262
         IF (TNR .EQ. 4) GO TO 263
C
C
C        GEN SHAPE TEMPS DRAWN
C
 110     CALL SQUAREO(XA,YA,XR,YR,RR,PARR,LBLR,SINCH)
         GO TO 999
 120     CALL RECTO(XA,YA,XR,YR,RR,PARR,LBLR,SINCH)
         GO TO 999
 130     CALL DRECTO(XA,YA,XR,YR,RR,PARR,LBLR,SINCH)
         GO TO 999
 140     GO TO 999
 260     CALL AULTHO(XA,YA,XR,YR,RR,PARR,LBLR,SINCH)
         GO TO 999
 261     CALL PP22TNO(XA,YA,XR,YR,RR,PARR,LBLR,SINCH)
         GO TO 999
 262     CALL SHEARO(XA,YA,XR,YR,RR,PARR,LBLR,SINCH)
         GO TO 999
 263     CALL PP72TNO(XA,YA,XR,YR,RR,PARR,LBLR,SINCH)
```

```
      GO TO 999
999   RETURN
      END
      SUBROUTINE TITLE
      CALL CMOPEN
      CALL TXAM
      CALL PIVOT(65.0,50.0)
      CALL NEWPAG
      CALL BELL
      CALL MOVE(55.0,50.0)
      CALL DRAW(55.0,30.0)
      CALL MOVE(85.0,50.0)
      CALL DRAW(95.0,50.0)
      CALL MOVE(80.0,60.0)
      CALL DRAW(70.0,80.0)
      CALL MOVE(25.0,40.0)
      CALL DRAW(30.0,40.0)
      CALL MOVE(20.0,30.0)
      CALL DRAW(10.0,40.0)
      CALL MOVE(40.0,70.0)
      CALL DRAW(50.0,70.0)
      CALL MOVE(80.0,30.0)
      CALL DRAW(80.0,50.0)
      CALL MOVE(50.0,50.0)
      CALL DRAW(40.0,50.0)
      CALL MOVE(65.0,80.0)
      CALL DRAW(65.0,60.0)
      CALL MOVE(105.0,40.0)
      CALL DRAW(110.0,50.0)
      CALL MOVE(25.0,50.0)
      CALL DRAW(25.0,40.0)
      CALL MOVE(25.0,30.0)
      CALL DRAW(25.0,40.0)
      CALL MOVE(40.0,70.0)
      CALL DRAW(40.0,80.0)
      CALL DRAW(50.0,80.0)
      CALL MOVE(20.0,50.0)
      CALL DRAW(10.0,40.0)
      CALL DRAW(10.0,50.0)
      CALL MOVE(100.0,50.0)
      CALL DRAW(100.0,30.0)
      CALL MOVE(80.0,60.0)
      CALL DRAW(80.0,70.0)
      CALL MOVE(55.0,60.0)
      CALL DRAW(65.0,60.0)
      CALL MOVE(65.0,50.0)
      CALL DRAW(55.0,50.0)
      CALL MOVE(85.0,40.0)
      CALL DRAW(95.0,40.0)
      CALL MOVE(70.0,60.0)
      CALL DRAW(70.0,70.0)
```

```
CALL MOVE(40.0,70.0)
CALL DRAW(50.0,60.0)
CALL MOVE(110.0,30.0)
CALL DRAW(110.0,50.0)
CALL MOVE(95.0,50.0)
CALL DRAW(95.0,30.0)
CALL MOVE(80.0,40.0)
CALL DRAW(70.0,40.0)
CALL MOVE(55.0,80.0)
CALL DRAW(65.0,80.0)
CALL MOVE(35.0,50.0)
CALL DRAW(25.0,50.0)
CALL MOVE(55.0,30.0)
CALL DRAW(65.0,30.0)
CALL MOVE(85.0,50.0)
CALL DRAW(85.0,30.0)
CALL MOVE(55.0,60.0)
CALL DRAW(55.0,70.0)
CALL MOVE(55.0,80.0)
CALL DRAW(55.0,70.0)
CALL MOVE(35.0,30.0)
CALL DRAW(25.0,30.0)
CALL MOVE(70.0,50.0)
CALL DRAW(70.0,30.0)
CALL MOVE(40.0,60.0)
CALL DRAW(40.0,70.0)
CALL MOVE(50.0,70.0)
CALL DRAW(50.0,80.0)
CALL MOVE(70.0,80.0)
CALL DRAW(70.0,70.0)
CALL MOVE(10.0,30.0)
CALL DRAW(10.0,40.0)
CALL MOVE(45.0,50.0)
CALL DRAW(45.0,30.0)
CALL MOVE(105.0,40.0)
CALL DRAW(100.0,50.0)
CALL MOVE(110.0,30.0)
CALL DRAW(110.0,50.0)
CALL MOVE(80.0,80.0)
CALL DRAW(80.0,70.0)
CALL BELL
CALL BELL
CALL MOVE(60.0,85.0)
CALL TEXT(1,'A')
CALL BELL
CALL MOVE(51.0,20.0)
CALL TEXT(10,'PRODUCTION')
CALL BELL
CALL BELL
CALL MOVE(0.0,0.0)
CALL CMCLOS
```

```
      RETURN
      END
      SUBROUTINE WALL
      CALL CMOPEN
      CALL TRIDNT(.TRUE.)
      CALL MOVE(50.0,50.0)
      CALL DRAW(50.0,70.0)
      CALL DRAW(80.0,70.0)
      CALL DRAW(80.0,50.0)
      CALL DRAW(50.0,50.0)
      CALL MOVE(65.0,73.0)
      CALL TXAM
      CALL TEXT(5,'NORTH')
      CALL MOVE(47.0,60.0)
      CALL TXAM
      CALL TEXT(4,'WEST')
      CALL MOVE(83.0,60.0)
      CALL TXAM
      CALL TEXT(4,'EAST')
      CALL MOVE(65.0,47.0)
      CALL TXAM
      CALL TEXT(5,'SOUTH')
      CALL HOME
      CALL CMCLOS
      DO  10 KKI = 1,20
10    PRINT *
      PRINT *,'          1. NORTH'
      PRINT *,'          2. SOUTH'
      PRINT *,'          3. EAST'
      PRINT *,'          4. WEST'
      RETURN
      END
      SUBROUTINE AULTH(XA,YA,XD,YD,RD,PARD,LBLD,G)
      REAL PARD(10)
      REAL BOR
      REAL XA,YA,XD,YD,RD,G
      REAL XAA,XAB,XAC,XAD,YAA,YAB,YAC
      REAL XDA,YDA
      CHARACTER * 10 LBLD
      CALL CMOPEN
      CALL VWPORT(39.0,130.0,30.0,100.0)
      CALL WINDOW(0.0,XA,0.0,YA)
      LBLD='AUTO LATHE'
      CALL TRANSL(XD,YD)
      CALL PIVOT(0.0,0.0)
      CALL ROTATE(RD,RD)
      RPD=360.0-RD
      XAA=-36.0*G
      XAB=36.0*G
      XAC=-7.2*G
      XAD=-16.8*G
```

```
YAA=-31.5*G
YAB=31.5*G
YAC=38.7*G
XDA=XD*-1.0
YDA=YD*-1.0
CALL MOVE(XAA,YAA)
CALL DRAW(XAA,YAB)
CALL DRAW(XAB,YAB)
CALL DRAW(XAB,YAA)
CALL DRAW(XAA,YAA)
CALL MOVE(XAC,YAB)
CALL DRAW(XAC,YAC)
CALL DRAW(XAD,YAC)
CALL DRAW(XAD,YAB)
BOR=48.0*G
PARD(7)=YD+BOR
PARD(8)=YD-BOR
PARD(9)=XD+BOR
PARD(10)=XD-BOR
CALL ROTATE(RPD,RPD)
CALL TRIDNT(.FALSE.)
CALL CMCLOS
RETURN
END
SUBROUTINE AULTHO(XA,YA,XD,YD,RD,PARD,LBLD,G)
REAL PARD(10)
REAL BOR
REAL XA,YA,XD,YD,RD,G
REAL XAA,XAB,XAC,XAD,YAA,YAB,YAC
REAL XDA,YDA
CHARACTER * 10 LBLD
CALL CMOPEN
LBLD='AUTO LATHE'
CALL TRANSL(XD,YD)
CALL PIVOT(0.0,0.0)
CALL ROTATE(RD,RD)
RPD=360.0-RD
XAA=-36.0*G
XAB=36.0*G
XAC=-7.2*G
XAD=-16.8*G
YAA=-31.5*G
YAB=31.5*G
YAC=38.7*G
XDA=XD*-1.0
YDA=YD*-1.0
CALL MOVE(XAA,YAA)
CALL DRAW(XAA,YAB)
CALL DRAW(XAB,YAB)
CALL DRAW(XAB,YAA)
CALL DRAW(XAA,YAA)
```

```
      CALL MOVE(XAC,YAB)
      CALL DRAW(XAC,YAC)
      CALL DRAW(XAD,YAC)
      CALL DRAW(XAD,YAB)
      BOR=48.0*G
      PARD(7)=YD+BOR
      PARD(8)=YD-BOR
      PARD(9)=XD+BOR
      PARD(10)=XD-BOR
      CALL ROTATE(RPD,RPD)
      CALL TRIDNT(.FALSE.)
      CALL CMCLOS
      RETURN
      END
       SUBROUTINE DRECT(XA,YA,XD,YD,RD,PARD,LBLD,S)
       CHARACTER * 10 LBLD
       REAL PARD(10)
       CALL CMOPEN
       CALL VWPORT(39.0,130.0,30.0,100.0)
       CALL WINDOW(0.0,XA,0.0,YA)
       CALL TRANSL(XD,YD)
       CALL PIVOT(0.0,0.0)
       CALL ROTATE(RD,RD)
       WH=PARD(1) * .5 *S
       RL=PARD(2) * S
       WIH=PARD(3) * .5 *S
       RIL=PARD(4) * S
       WHN=WH*-1.0
       RLN=RL*-1.0
       WIHN=WIH*-1.0
       CALL MOVE(0.0,0.0)
       CALL DRAW(0.0,WHN)
       CALL DRAW(RLN,WHN)
       CALL DRAW(RLN,WH)
       CALL DRAW(0.0,WH)
       CALL DRAW(0.0,0.0)
       CALL DRAW(0.0,WIHN)
       CALL DRAW(RIL,WIHN)
       CALL DRAW(RIL,WIH)
       CALL DRAW(0.0,WIH)
       CALL DRAW(0.0,0.0)
       CALL CMARK(0.0,0.0)
       PPA=WH**2.0+RL**2.0
       PPB=WIH**2.0+RLI**2.0
       PR = SQRT(PPA)
       PPC=SQRT(PPB)
       IF (PPC .GT. PR) PR = PPC
       PARD(7)=YD+PR
       PARD(8)=YD-PR
       PARD(9)=XD+PR
       PARD(10)=XD-PR
```

```
      CALL TRIDNT(.FALSE.)
      CALL CMCLOS
      RETURN
      END
      SUBROUTINE DRECTA(PARD,LBLD)
      CHARACTER * 10 LBLD
      REAL PARD(10)
1     FORMAT(A10)
      KLP=1
50    CALL CLEAR
      PRINT *,'GENERAL TEMPLATE: RECTANGLE'
      CX=65.0
      CY=50.0
      CALL CMOPEN
      CALL VWPORT(0.0,130.0,50.0,100.0)
      CALL WINDOW(0.0,130.0,0.0,100.0)
      CALL CMARK(CX,CY)
      CALL MOVE(40.,30.)
      CALL DRAW(40.,70.)
      CALL DRAW(65.,70.)
      CALL DRAW(65.,30.)
      CALL DRAW(40.,30.)
      CALL MOVE(65.,60.)
      CALL DRAW(95.,60.)
      CALL DRAW(95.,40.)
      CALL DRAW(65.,40.)
      CALL MOVE(30.,30.)
      CALL DRAW(35.,30.)
      CALL MOVE(33.5,30.)
      CALL DRAW(33.5,47.)
      CALL MOVE(33.5,53.)
      CALL DRAW(33.5,70.)
      CALL MOVE(30.0,70.0)
      CALL DRAW(35.0,70.0)
      CALL MOVE(40.,20.)
      CALL DRAW(40.,25.)
      CALL MOVE(40.,22.5)
      CALL DRAW(48.5,22.5)
      CALL MOVE(56.5,22.5)
      CALL DRAW(65.,22.5)
      CALL MOVE(65.,20.)
      CALL DRAW(65.,25.)
      CALL MOVE(65.,30.)
      CALL DRAW(65.,35.)
      CALL MOVE(65.,33.5)
      CALL DRAW(76.5,33.5)
      CALL MOVE(83.5,33.5)
      CALL DRAW(95.0,33.5)
      CALL MOVE(95.0,30.0)
      CALL DRAW(95.,35.)
      CALL MOVE(100.,40.)
```

```
      CALL DRAW(105.,40.)
      CALL MOVE(102.5,40.)
      CALL DRAW(102.5,47.)
      CALL MOVE(102.5,53.)
      CALL DRAW(102.5,60.)
      CALL MOVE(100.,60.)
      CALL DRAW(105.,60.)
      CALL MOVE(32.5,47.5)
      CALL TXAM
      CALL TEXT(2,'W1')
      CALL MOVE(48.5,21.)
      CALL TXAM
      CALL TEXT(2,'L1')
      CALL MOVE(77.,31.)
      CALL TXAM
      CALL TEXT(2,'L2')
      CALL MOVE(101.,48.)
      CALL TXAM
      CALL TEXT(2,'W2')
      CALL MOVE(42.,48.)
      CALL TXAM
      CALL TEXT(7,'RECT. 1')
      CALL MOVE(67.,48.)
      CALL TXAM
      CALL TEXT(7,'RECT. 2')
      CALL MOVE(0.0,30.0)
      CALL CMCLOS
      IF (KLP .EQ. 2) GO TO 150
      PRINT *
      PRINT *
      PRINT *,'INPUT IN INCHES WIDTH OF RECTANGLE 1 (W1)'
      READ *,PARD(1)
      PRINT *
      PRINT *,'INPUT IN INCHES LENGHT OF RECTANGLE 1, (L1)'
      READ *,PARD(2)
      KLP=2
      GO TO 50
150   PRINT *
      PRINT *
      PRINT *,'INPUT IN INCHES WIDTH OF RECTANGLE 2, (W2)'
      READ *,PARD(3)
      PRINT *
      PRINT *,'INPUT IN INCHES LENGHT OF RECTANGLE 2, (L2)'
      READ *,PARD(4)
      PRINT *
      PRINT *,'INPUT TEMPLATE LABEL'
      READ (*,1) LBLD
      CALL CLEAR
      RETURN
      END
      SUBROUTINE DRECTO(XA,YA,XD,YD,RD,PARD,LBLD,S)
```

```
CHARACTER * 10 LBLD
REAL PARD(10)
CALL CMOPEN
CALL TRANSL(XD,YD)
CALL PIVOT(0.0,0.0)
CALL ROTATE(RD,RD)
WH=PARD(1) * .5 *S
RL=PARD(2) * S
WIH=PARD(3) * .5 *S
RIL=PARD(4) * S
WHN=WH*-1.0
RLN=RL*-1.0
WIHN=WIH*-1.0
CALL MOVE(0.0,0.0)
CALL DRAW(0.0,WHN)
CALL DRAW(RLN,WHN)
CALL DRAW(RLN,WH)
CALL DRAW(0.0,WH)
CALL DRAW(0.0,0.0)
CALL DRAW(0.0,WIHN)
CALL DRAW(RIL,WIHN)
CALL DRAW(RIL,WIH)
CALL DRAW(0.0,WIH)
CALL DRAW(0.0,0.0)
PPA=WH**2.0+RL**2.0
PPB=WIH**2.0+RLI**2.0
PR = SQRT(PPA)
PPC=SQRT(PPB)
IF (PPC .GT. PR) PR = PPC
PARD(7)=YD+PR
PARD(8)=YD-PR
PARD(9)=XD+PR
PARD(10)=XD-PR
CALL TRIDNT(.FALSE.)
CALL CMCLOS
RETURN
END
SUBROUTINE PP22TN(XA,YA,XD,YD,RD,PARD,LBLD,G)
REAL PARD(10)
REAL BOR
REAL XA,YA,XD,YD,RD,G
CHARACTER * 10 LBLD
CALL CMOPEN
CALL VWPORT(39.0,130.0,30.0,100.0)
CALL WINDOW(0.0,XA,0.0,YA)
LBLD='22TN PRESS'
P=G*48.0
PLUS=(.425**2)+(.575**2)
BOR=SQRT(PLUS)
BOR = BOR * P
CALL TRANSL(XD,YD)
```

```
CALL PIVOT(0.0,0.0)
CALL ROTATE(RD,RD)
XAA=-.325*P
XAB=-.425*P
XAC=-.225*P
XAD=.075*P
XAE=.175*P
XAF=.275*P
XAG=.425*P
XAH=.125*P
XAI=.1*P
XAJ=.2*P
YAA=-.35*P
YAB=-.05*P
YAC=.25*P
YAD=.45*P
YAE=.37*P
YAF=.3*P
YAG=-.2*P
YAH=-.335*P
YAI=-.575*P
YAJ=-.25*P
YAK=.575*P
CALL MOVE(XAA,YAA)
CALL DRAW(XAA,YAB)
CALL DRAW(XAB,YAB)
CALL DRAW(XAB,YAC)
CALL DRAW(XAA,YAC)
CALL DRAW(XAA,YAD)
CALL DRAW(XAC,YAD)
CALL DRAW(XAC,YAE)
CALL DRAW(XAA,YAE)
CALL MOVE(XAC,YAE)
CALL DRAW(XAC,YAF)
CALL DRAW(XAD,YAF)
CALL DRAW(XAD,YAD)
CALL DRAW(XAE,YAD)
CALL DRAW(XAE,YAE)
CALL DRAW(XAD,YAE)
CALL MOVE(XAE,YAE)
CALL DRAW(XAE,YAC)
CALL DRAW(XAF,YAC)
CALL DRAW(XAF,YAG)
CALL DRAW(XAG,YAG)
CALL DRAW(XAG,YAH)
CALL DRAW(XAF,YAH)
CALL DRAW(XAF,YAI)
CALL DRAW(XAH,YAI)
CALL DRAW(XAH,YAA)
CALL DRAW(XAI,YAA)
CALL DRAW(XAI,YAJ)
```

```
CALL DRAW(XAJ,YAJ)
CALL DRAW(XAJ,YAA)
CALL DRAW(XAA,YAA)
CALL MOVE(XAA,YAJ)
CALL DRAW(XAC,YAJ)
CALL DRAW(XAC,YAA)
CALL MOVE(XAA,YAD)
CALL DASHPT(3)
CALL DRAW(XAA,YAK)
CALL DRAW(XAF,YAK)
CALL DRAW(XAF,YAC)
CALL DRAW(XAG,YAC)
CALL DRAW(XAG,YAH)
CALL DASHPT(0)
PARD(7)=YD+BOR
PARD(8)=YD-BOR
PARD(9)=XD+BOR
PARD(10)=XD-BOR
CALL TRIDNT(.FALSE.)
CALL CMCLOS
RETURN
END
SUBROUTINE PP22TNO(XA,YA,XD,YD,RD,PARD,LBLD,G)
REAL PARD(10)
REAL BOR
REAL XA,YA,XD,YD,RD,G
CHARACTER * 10 LBLD
CALL CMOPEN
LBLD='22TN PRESS'
P=G*48.0
PLUS=(.425**2)+(.575**2)
BOR=SQRT(PLUS)
BOR=BOR*P
CALL TRANSL(XD,YD)
CALL PIVOT(0.0,0.0)
CALL ROTATE(RD,RD)
XAA=-.325*P
XAB=-.425*P
XAC=-.225*P
XAD=.075*P
XAE=.175*P
XAF=.275*P
XAG=.425*P
XAH=.125*P
XAI=.1*P
XAJ=.2*P
YAA=-.35*P
YAB=-.05*P
YAC=.25*P
YAD=.45*P
YAE=.37*P
```

```
YAF=.3*P
YAG=-.2*P
YAH=-.335*P
YAI=-.575*P
YAJ=-.25*P
YAK=.575*P
CALL MOVE(XAA,YAA)
CALL DRAW(XAA,YAB)
CALL DRAW(XAB,YAB)
CALL DRAW(XAB,YAC)
CALL DRAW(XAA,YAC)
CALL DRAW(XAA,YAD)
CALL DRAW(XAC,YAD)
CALL DRAW(XAC,YAE)
CALL DRAW(XAA,YAE)
CALL MOVE(XAC,YAE)
CALL DRAW(XAC,YAF)
CALL DRAW(XAD,YAF)
CALL DRAW(XAD,YAD)
CALL DRAW(XAE,YAD)
CALL DRAW(XAE,YAE)
CALL DRAW(XAD,YAE)
CALL MOVE(XAE,YAE)
CALL DRAW(XAE,YAC)
CALL DRAW(XAF,YAC)
CALL DRAW(XAF,YAG)
CALL DRAW(XAG,YAG)
CALL DRAW(XAG,YAH)
CALL DRAW(XAF,YAH)
CALL DRAW(XAF,YAI)
CALL DRAW(XAH,YAI)
CALL DRAW(XAH,YAA)
CALL DRAW(XAI,YAA)
CALL DRAW(XAI,YAJ)
CALL DRAW(XAJ,YAJ)
CALL DRAW(XAJ,YAA)
CALL DRAW(XAA,YAA)
CALL MOVE(XAA,YAJ)
CALL DRAW(XAC,YAJ)
CALL DRAW(XAC,YAA)
CALL MOVE(XAA,YAD)
CALL DASHPT(3)
CALL DRAW(XAA,YAK)
CALL DRAW(XAF,YAK)
CALL DRAW(XAF,YAC)
CALL DRAW(XAG,YAC)
CALL DRAW(XAG,YAH)
CALL DASHPT(0)
PARD(7)=YD+BOR
PARD(8)=YD-BOR
PARD(9)=XD+BOR
```

```
PARD(10)=XD-BOR
CALL TRIDNT(.FALSE.)
CALL CMCLOS
RETURN
END
SUBROUTINE PP72TN(XA,YA,XD,YD,RD,PARD,LBLD,G)
REAL PARD(10)
REAL BOR
REAL XA,YA,XD,YD,RD,G
CHARACTER * 10 LBLD
CALL CMOPEN
CALL VWPORT(39.0,130.0,30.0,100.0)
CALL WINDOW(0.0,XA,0.0,YA)
LBLD='72TN PRESS'
P=G*48.0
PLUS=(.625**2)+(.825**2)
BOR=SQRT(PLUS)
BOR = BOR * P
CALL TRANSL(XD,YD)
CALL PIVOT(0.0,0.0)
CALL ROTATE(RD,RD)
XAA=.35*P
XAB=.5*P
XAC=.625*P
XAD=.4*P
XAE=.325*P
XAF=-.275*P
XAG=-.35*P
XAH=-.525*P
XAI=-.25*P
XAJ=.45*P
YAA=-.675*P
YAB=-.825*P
YAC=-.575*P
YAD=-.4*P
YAE=-.075*P
YAF=.7*P
YAG=.55*P
YAH=.3*P
YAI=-.2*P
YAJ=-.6*P
YAK=-.45*P
YAL=.125*P
YAM=.825*P
CALL MOVE(0.0,YAA)
CALL DRAW(XAA,YAA)
CALL DRAW(XAA,YAB)
CALL DRAW(XAB,YAB)
CALL DRAW(XAB,YAC)
CALL DRAW(XAC,YAC)
CALL DRAW(XAC,YAB)
```

```
CALL DRAW(XAB,YAD)
CALL DRAW(XAB,YAE)
CALL DRAW(XAD,YAE)
CALL DRAW(XAD,YAF)
CALL DRAW(XAE,YAF)
CALL DRAW(XAE,YAG)
CALL DRAW(XAD,YAG)
CALL DRAW(XAE,YAG)
CALL DRAW(XAE,YAH)
CALL DRAW(XAF,YAH)
CALL DRAW(XAF,YAF)
CALL DRAW(XAG,YAF)
CALL DRAW(XAG,YAG)
CALL DRAW(XAF,YAG)
CALL DRAW(XAG,YAG)
CALL DRAW(XAG,YAH)
CALL DRAW(XAH,YAH)
CALL DRAW(XAH,YAI)
CALL DRAW(XAG,YAI)
CALL DRAW(XAG,YAJ)
CALL DRAW(XAI,YAJ)
CALL DRAW(XAI,YAK)
CALL DRAW(XAG,YAK)
CALL MOVE(XAI,YAJ)
CALL DRAW(XAI,YAA)
CALL DRAW(0.0,YAA)
CALL MOVE(XAA,YAJ)
CALL DRAW(XAJ,YAJ)
CALL DRAW(XAJ,YAK)
CALL DRAW(XAA,YAK)
CALL DRAW(XAA,YAJ)
CALL DASHPT(3)
CALL MOVE(XAC,YAB)
CALL DRAW(XAC,YAL)
CALL DRAW(XAD,YAL)
CALL MOVE(XAD,YAF)
CALL DRAW(XAD,YAM)
CALL DRAW(XAC,YAM)
CALL DRAW(XAC,YAF)
CALL DASHPT(0)
PARD(7)=YD+BOR
PARD(8)=YD-BOR
PARD(9)=XD+BOR
PARD(10)=XD-BOR
CALL TRIDNT(.FALSE.)
CALL CMCLOS
RETURN
END
SUBROUTINE PP72TNO(XA,YA,XD,YD,RD,PARD,LDLD,B)
REAL PARD(10)
REAL BOR
```

```
REAL XA,YA,XD,YD,RD,G
CHARACTER * 10 LBLD
CALL CMOPEN
LBLD='72TN PRESS'
P=G*48.0
PLUS=(.625**2)+(.825**2)
BOR=SQRT(PLUS)
BOR = BOR * P
CALL TRANSL(XD,YD)
CALL PIVOT(0.0,0.0)
CALL ROTATE(RD,RD)
XAA=.35*P
XAB=.5*P
XAC=.625*P
XAD=.4*P
XAE=.325*P
XAF=-.275*P
XAG=-.35*P
XAH=-.525*P
XAI=-.25*P
XAJ=.45*P
YAA=-.675*P
YAB=-.825*P
YAC=-.575*P
YAD=-.4*P
YAE=-.075*P
YAF=.7*P
YAG=.55*P
YAH=.3*P
YAI=-.2*P
YAJ=-.6*P
YAK=-.45*P
YAL=.125*P
YAM=.825*P
CALL MOVE(0.0,YAA)
CALL DRAW(XAA,YAA)
CALL DRAW(XAA,YAB)
CALL DRAW(XAB,YAB)
CALL DRAW(XAB,YAC)
CALL DRAW(XAC,YAC)
CALL DRAW(XAC,YAD)
CALL DRAW(XAB,YAD)
CALL DRAW(XAB,YAE)
CALL DRAW(XAD,YAE)
CALL DRAW(XAD,YAF)
CALL DRAW(XAE,YAF)
CALL DRAW(XAE,YAG)
CALL DRAW(XAD,YAG)
CALL DRAW(XAE,YAG)
CALL DRAW(XAE,YAH)
CALL DRAW(XAF,YAH)
```

```
      CALL DRAW(XAF,YAF)
      CALL DRAW(XAG,YAF)
      CALL DRAW(XAG,YAG)
      CALL DRAW(XAF,YAG)
      CALL DRAW(XAG,YAG)
      CALL DRAW(XAG,YAH)
      CALL DRAW(XAH,YAH)
      CALL DRAW(XAH,YAI)
      CALL DRAW(XAG,YAI)
      CALL DRAW(XAG,YAJ)
      CALL DRAW(XAI,YAJ)
      CALL DRAW(XAI,YAK)
      CALL DRAW(XAG,YAK)
      CALL MOVE(XAI,YAJ)
      CALL DRAW(XAI,YAA)
      CALL DRAW(0.0,YAA)
      CALL MOVE(XAA,YAJ)
      CALL DRAW(XAJ,YAJ)
      CALL DRAKXAJ,YAK)
      CALL DRAW(XAA,YAK)
      CALL DRAW(XAA,YAJ)
      CALL DASHPT(3)
      CALL MOVE(XAC,YAD)
      CALL DRAW(XAC,YAL)
      CALL DRAW(XAD,YAL)
      CALL MOVE(XAD,YAF)
      CALL DRAW(XAD,YAM)
      CALL DRAW(XAG,YAM)
      CALL DRAW(XAG,YAF)
      CALL DASHPT(0)
      PARD(7)=YD+BOR
      PARD(8)=YD-BOR
      PARD(9)=XD+BOR
      PARD(10)=XD-BOR
      CALL TRIDNT(.FALSE.)
      CALL CMCLOS
      RETURN
      END
      SUBROUTINE RECTA(PARD,LBLD)
      CHARACTER * 10 LBLD
      REAL PARD(10)
1     FORMAT(A10)
      CALL CLEAR
      PRINT *,'GENERAL TEMPLATE: RECTANGLE'
      CX=65.0
      CY=50.0
      CALL CMOPEN
      CALL VWPORT(0.0,130.0,50.0,100.0)
      CALL WINDOW(0.0,130.0,0.0,100.0)
      CALL CMARK(CX,CY)
      CALL MOVE(50.0,40.0)
```

```
CALL DRAW(80.0,40.0)
CALL DRAW(80.0,60.0)
CALL DRAW(50.0,60.0)
CALL DRAW(50.0,40.0)
CALL MOVE(45.0,40.0)
CALL DRAW(49.0,40.0)
CALL MOVE(47.0,40.0)
CALL DRAW(47.0,47.5)
CALL DRAW(47.0,47.5)
CALL MOVE(47.0,52.5)
CALL DRAW(47.0,60.0)
CALL MOVE(45.0,60.0)
CALL DRAW(49.0,60.0)
CALL MOVE(50.0,35.0)
CALL DRAW(50.0,39.0)
CALL MOVE(50.0,37.0)
CALL DRAW(62.5,37.0)
CALL MOVE(67.5,37.0)
CALL DRAW(80.0,37.0)
CALL MOVE(80.0,35.0)
CALL DRAW(80.0,39.0)
CALL MOVE(47.0,48.0)
CALL TXAM
CALL TEXT(1,'W')
CALL MOVE(63.0,36.0)
CALL TXAM
CALL TEXT(1,'L')
CALL MOVE(0.0,40.0)
CALL CMCLOS
PRINT *
PRINT *
PRINT *,'INPUT THE WIDTH (W) MEASUREMENT IN TI
PRINT *,'      OF INCHES.'
READ *,PARD(1)
PARD(1) = PARD(1) * .5
PRINT *
PRINT *,'INPUT THE LENGHT (L) MEASUREMENT IN
PRINT *,'      OF INCHES.'
READ *,PARD(2)
PARD(2)=PARD(2)*.5
PRINT *
PRINT *,'INPUT TEMPLATE LABEL'
READ (*,1) LBLD
CALL CLEAR
RETURN
END
SUBROUTINE RECT(XA,YA,XD,YD,RD,PARD,LBLD,S)
CHARACTER * 10 LBLD
REAL PARD(10)
CALL CMOPEN
CALL VWPORT(39.0,130.0,30.0,100.0)
```

```
CALL WINDOW(0.0,XA,0.0,YA)
CALL TRANSL(XD,YD)
CALL PIVOT(0.0,0.0)
CALL ROTATE(RD,RD)
WH = PARD(1) *S
RL = PARD(2) *S
WHN=WH*-1.0
RLN=RL*-1.0
CALL MOVE(RLN,WHN)
CALL DRAW(RLN,WH)
CALL DRAW(RL,WH)
CALL DRAW(RL,WHN)
CALL DRAW(RLN,WHN)
CALL CMARK(0.0,0.0)
PP = WH **2.0 + RL **2.0
PR = SQRT(PP)
PARD(7)=YD+PR
PARD(8)=YD-PR
PARD(9)=XD+PR
PARD(10)=XD-PR
CALL TRIDNT(.FALSE.)
CALL CMCLOS
RETURN
END
SUBROUTINE RECTO(XA,YA,XD,YD,RD,PARD,LBLD,S)
CHARACTER * 10 LBLD
REAL PARD(10)
CALL CMOPEN
CALL TRANSL(XD,YD)
CALL PIVOT(0.0,0.0)
CALL ROTATE(RD,RD)
WH = PARD(1) *S
RL = PARD(2) *S
WHN=WH*-1.0
RLN=RL*-1.0
CALL MOVE(RLN,WHN)
CALL DRAW(RLN,WH)
CALL DRAW(RL,WH)
CALL DRAW(RL,WHN)
CALL DRAW(RLN,WHN)
PP = WH **2.0 + RL **2.0
PR = SQRT(PP)
PARD(7)=YD+PR
PARD(8)=YD-PR
PARD(9)=XD+PR
PARD(10)=XD-PR
CALL TRIDNT(.FALSE.)
CALL CMCLOS
RETURN
END
SUBROUTINE SHEAR(XA,YA,XD,YD,RD,PARD,LBLD,S)
```

```
REAL PARD(10)
REAL BOR
REAL XA,YA,XD,YD,RD,G
CHARACTER * 10 LBLD
CALL CMOPEN
CALL VWPORT(39.0,130.0,30.0,100.0)
CALL WINDOW(0.0,XA,0.0,YA)
LBLD='SHEAR'
P=G*48.0
PLUS=(1.985**2.0) + (1.575**2.0)
BOR=SQRT(PLUS)
BOR = BOR * P
CALL TRANSL(XD,YD)
CALL PIVOT(0.0,0.0)
CALL ROTATE(RD,RD)
XAA=1.0*P
XAB=1.1*P
XAC=1.4*P
XAD=1.575*P
XAE=1.625*P
XAF=1.725*P
XAG=1.8*P
XAH=-1.625*P
XAI=-1.675*P
XAJ=-1.775*P
XAK=-1.875*P
XAL=-1.985*P
XAM=-1.1*P
XAN=-1.0*P
YAA=-.425*P
YAB=-1.175*P
YAC=-1.575*P
YAD=.975*P
YAE=.875*P
YAF=.725*P
YAG=-.175*P
YAH=-.325*P
YAI=.425*P
YAJ=-.375*P
YAK=.675*P
CALL MOVE(0.0,YAA)
CALL DRAW(XAA,YAA)
CALL DRAW(XAA,YAB)
CALL DRAW(XAB,YAB)
CALL DRAW(XAB,YAA)
CALL DRAW(XAC,YAA)
CALL DRAW(XAC,YAC)
CALL DRAW(XAD,YAC)
CALL DRAW(XAD,YAD)
CALL DRAW(XAE,YAD)
CALL DRAW(XAE,YAE)
```

```
CALL DRAW(XAF,YAE)
CALL DRAW(XAF,YAF)
CALL DRAW(XAE,YAF)
CALL DRAW(XAE,YAG)
CALL DRAW(XAF,YAG)
CALL DRAW(XAF,YAH)
CALL DRAW(XAE,YAH)
CALL DRAW(XAE,YAA)
CALL DRAW(XAD,YAA)
CALL DRAW(XAC,YAA)
CALL DRAW(XAC,YAD)
CALL DRAW(XAD,YAD)
CALL DRAW(XAD,YAI)
CALL DRAW(XAH,YAI)
CALL DRAW(XAH,YAD)
CALL DRAW(XAI,YAD)
CALL DRAW(XAI,YAE)
CALL DRAW(XAJ,YAE)
CALL DRAW(XAJ,YAF)
CALL DRAW(XAI,YAF)
CALL DRAW(XAI,YAG)
CALL DRAW(XAJ,YAG)
CALL DRAW(XAJ,YAH)
CALL DRAW(XAI,YAH)
CALL DRAW(XAI,YAA)
CALL DRAW(XAH,YAA)
CALL MOVE(XAI,YAG)
CALL DRAW(XAK,YAG)
CALL DRAW(XAK,YAJ)
CALL DRAW(XAL,YAJ)
CALL DRAW(XAL,YAK)
CALL DRAW(XAI,YAK)
CALL MOVE(XAH,YAA)
CALL DRAW(XAM,YAA)
CALL DRAW(XAM,YAB)
CALL DRAW(XAN,YAB)
CALL DRAW(XAN,YAA)
CALL DRAW(0.0,YAA)
PARD(7)=YD+BOR
PARD(8)=YD-BOR
PARD(9)=XD+BOR
PARD(10)=XD-BOR
CALL TRIDNT(.FALSE.)
CALL CMCLOS
RETURN
END
SUBROUTINE SHEARO(XA,YA,XD,YD,RD,PARD)
REAL PARD(10)
REAL BOR
REAL XA,YA,XD,YD,RD,C
CHARACTER * 10 LBLD
```

```
CALL CHOPEN
LBLD='SHEAR'
P=G*40.0
PLUS=(1.785**2.0) + (1.575**2.0)
BOR=SQRT(PLUS)
BOR = BOR * P
CALL TRANSL(XD,YD)
CALL PIVOT(0.0,0.0)
CALL ROTATE(RD,RD)
XAA=1.0*P
XAB=1.1*P
XAC=1.4*P
XAD=1.575*P
XAE=1.625*P
XAF=1.725*P
XAG=1.0*P
XAH=-1.625*P
XAI=-1.675*P
XAJ=-1.775*P
XAK=-1.875*P
XAL=-1.785*P
XAM=-1.1*P
XAN=-1.0*P
YAA=-.425*P
YAB=-1.175*P
YAC=-1.575*P
YAD=.975*P
YAE=.875*P
YAF=.725*P
YAG=-.175*P
YAH=-.325*P
YAI=.425*P
YAJ=-.375*P
YAK=.675*P
CALL MOVE(0.0,YAA)
CALL DRAW(XAA,YAA)
CALL DRAW(XAA,YAB)
CALL DRAW(XAB,YAB)
CALL DRAW(XAB,YAA)
CALL DRAW(XAC,YAA)
CALL DRAW(XAC,YAC)
CALL DRAW(XAD,YAC)
CALL DRAW(XAD,YAD)
CALL DRAW(XAE,YAD)
CALL DRAW(XAE,YAE)
CALL DRAW(XAF,YAE)
CALL DRAW(XAF,YAF)
CALL DRAW(XAE,YAF)
CALL DRAW(XAE,YAG)
CALL DRAW(XAF,YAG)
CALL DRAW(XAF,YAH)
```

```
CALL DRAW(XAE,YAH)
CALL DRAW(XAE,YAA)
CALL DRAW(XAD,YAA)
CALL DRAW(XAG,YAA)
CALL DRAW(XAG,YAD)
CALL DRAW(XAD,YAB)
CALL DRAW(XAD,YAI)
CALL DRAW(XAH,YAI)
CALL DRAW(XAH,YAD)
CALL DRAW(XAI,YAD)
CALL DRAW(XAI,YAE)
CALL DRAW(XAJ,YAE)
CALL DRAW(XAJ,YAF)
CALL DRAW(XAI,YAF)
CALL DRAW(XAI,YAG)
CALL DRAW(XAJ,YAG)
CALL DRAW(XAJ,YAH)
CALL DRAW(XAI,YAH)
CALL DRAW(XAI,YAA)
CALL DRAW(XAH,YAA)
CALL MOVE(XAI,YAG)
CALL DRAW(XAK,YAG)
CALL DRAW(XAK,YAJ)
CALL DRAW(XAL,YAJ)
CALL DRAW(XAL,YAK)
CALL DRAW(XAI,YAK)
CALL MOVE(XAH,YAA)
CALL DRAW(XAM,YAA)
CALL DRAW(XAM,YAD)
CALL DRAW(XAN,YAD)
CALL DRAW(XAN,YAA)
CALL DRAW(0.0,YAA)
PARD(7)=YD+DOR
PARD(8)=YD-DOR
PARD(9)=XD+DOR
PARD(10)=XD-DOR
CALL TRIDHT(.FALSE.)
CALL CMCLOS
RETURN
END
SUBROUTINE SQUARE(XA,YA,XD,YD,RD..
CHARACTER * 10 LBLD
REAL PARD(10)
CALL CMOPEN
CALL VWPORT(37.0,130.0,30.0,100.0
CALL WINDOW(0.0,XA,0.0,YA)
CALL TRANSL(XD,YD)
CALL PIVOT(0.0,0.0)
CALL ROTATE(RD,RD)
SH=PARD(1)*S
SX=SH*-1.0
```

```
      CALL MOVE(SN,SN)
      CALL DRAW(SN,SH)
      CALL DRAW(SH,SH)
      CALL DRAW(SH,SN)
      CALL DRAW(SN,SN)
      CALL CMARK(0.0,0.0)
      PP = SH **2.0 + SH **2.0
      PR = SQRT(PP)
      PARD(7)=YD+PR
      PARD(8)=YD-PR
      PARD(9)=XD+PR
      PARD(10)=XD-PR
      CALL TRIDNT(.FALSE.)
      CALL CMCLOS
      RETURN
      END
      SUBROUTINE SQUAREA(PARD,LBLD)
      CHARACTER * 10 LBLD
      REAL PARD(10)
1     FORMAT(A10)
      CALL CLEAR
      PRINT *,'GENERAL TEMPLATE: SQUARE'
      CX=65.0
      CY=50.0
      CALL CMOPEN
      CALL VWPORT(0.0,130.0,50.0,100.0)
      CALL WINDOW(0.0,130.0,0.0,100.0)
      CALL CMARK(CX,CY)
      CALL MOVE(55.0,30.0)
      CALL DRAW(75.0,30.0)
      CALL DRAW(75.0,70.0)
      CALL DRAW(55.0,70.0)
      CALL DRAW(55.0,30.0)
      CALL MOVE(50.0,30.0)
      CALL DRAW(45.0,30.0)
      CALL MOVE(47.5,30.0)
      CALL DRAW(47.5,47.5)
      CALL MOVE(47.5,52.5)
      CALL DRAW(47.5,75.0)
      CALL MOVE(45.0,75.0)
      CALL DRAW(50.0,75.0)
      CALL MOVE(47.5,45.0)
      CALL TXAM
      CALL TEXT(1,'S')
      CALL MOVE(0.0,0.0)
      CALL CMCLOS
      PRINT *,'INPUT THE SIDE (S) MEASUREMENT IN TERMS
      PRINT *,'      OF INCHES.'
      PRINT *
      PRINT *,'EXAMPLE: INPUT 39 FOR 3FT, 3IN.'
      READ *, PARD(1)
```

```
PARD(1)=PARD(1)*.5
PRINT *
PRINT *,'INPUT TEMPLATE LABEL.'
READ (*,1)LBLD
CALL CLEAR
RETURN
END
SUBROUTINE SQUARED(XA,YA,XD,YD,RD,PARD,LBLD,S)
CHARACTER * 10 LBLD
REAL PARD(10)
CALL CMOPEN
CALL TRANSL(XD,YD)
CALL PIVOT(0.0,0.0)
CALL ROTATE(RD,RD)
SH=PARD(1)*S
SN =SH* -1.0
CALL MOVE(SN,SN)
CALL DRAW(SN,SH)
CALL DRAW(SH,SH)
CALL DRAW(SH,SN)
CALL DRAW(SN,SN)
PP = SH **2.0 + SH **2.0
PR = SQRT(PP)
PARD(7)=YD+PR
PARD(8)=YD-PR
PARD(9)=XD+PR
PARD(10)=XD-PR
CALL TRIDNT(.FALSE.)
CALL CMCLOS
RETURN
END
SUBROUTINE ACCEPT(RTE,BF,NBPO,PIV,PTS,NDPOP,DFF)
REAL BF(2,4),DFF(4)
INTEGER PIV,NBPO(2),RND(4,4),NDPOP(2)
INTEGER PTS(2),RDF(4,4)
RND(1,1)=4
RND(1,2)=2
RND(1,3)=3
RND(2,1)=3
RND(2,2)=1
RND(2,3)=4
RND(3,1)=1
RND(3,2)=1
RND(3,3)=2
RND(4,1)=2
RND(4,2)=3
RND(4,3)=1
RND(1,4)=1
RND(2,4)=2
RND(3,4)=3
RND(4,4)=4
```

```
      IF(RTE .EQ. 90.) IR=1
      IF(RTE .EQ. 180.) IR=2
      IF(RTE .EQ. 270.) IR=3
      IF (RTE .EQ. 0.0) IR = 4
      IF (RTE .EQ. 0.0) GO TO 25
      IF (PTS(1) .EQ. 0) GO TO 10
      NBPOP(1) = RND(NBPO(1),IR)
      GO TO 35
10    NBPOP(1) = 0
      GO TO 35
25    NBPOP(1) = NBPO(1)
35    IF (RTE .EQ. 0.0) GO TO 50
      IF (PTS(2) .EQ. 0) GO TO 45
      NBPOP(2) = RND(NBPO(2),IR)
      GO TO 60
45    NBPOP(2) = 0
      GO TO 60
50    NBPOP(1)=NBPO(1)
60    RBF(1,1)=3
      RBF(1,2)=4
      RBF(1,3)=2
      RBF(1,4)=1
      RBF(2,1)=2
      RBF(2,2)=1
      RBF(2,3)=4
      RBF(2,4)=3
      RBF(3,1)=4
      RBF(3,2)=3
      RBF(3,3)=1
      RBF(3,4)=2
      RBF(4,1)=1
      RBF(4,2)=2
      RBF(4,3)=3
      RBF(4,4)=4
      DO 70 JAB=1,4
70    BFF(JAB) = BF(PIV,RBF(IR,JAB))
      RETURN
      END
      SUBROUTINE BAREA(ZW,ZL,S,ZA)
      ZWW=ZW/S
      ZLL=ZL/S
      ZWWW=ZWW/12.0
      ZLLL=ZLL/12.0
      ZA=ZWWW*ZLLL
      RETURN
      END
      SUBROUTINE CLEAR
      CALL CMOPEN
      CALL NEWPAG
      CALL CMCLOS
      RETURN
```

```
      END
      SUBROUTINE CMARK(CX,CY)
      CXP=CX+1.0
      CXM=CX-1.0
      CYP=CY+1.0
      CYM=CY-1.0
      CALL MOVE(CXM,CY)
      CALL DRAW(CXP,CY)
      CALL MOVE(CX,CYM)
      CALL DRAW(CX,CYP)
      RETURN
      END
      SUBROUTINE CROSS(XS,YS,AN)
      CALL CMOPEN
      D=10.0
      CALL MOVE(XS-D,YS+D)
      CALL DRAW(XS+D,YS-D)
      CALL MOVE(XS+D,YS+D)
      CALL DRAW(XS-D,YS-D)
      CALL MOVE(XS+D,YS+D)
      IF (AN .EQ. 2.0) GO TO 10
      CALL TXAM(1,'E')
      GO TO 20
 10   CALL TXAM(1,'D')
 20   CALL CMCLOS
      RETURN
      END
      SUBROUTINE DOUBLE(NPLAC,SCALE,XA,YA,A,PLACE,PLOT,
    1 IPLOT,FX,FY,NSEL,WALLS,TWALLS)
      REAL PLOT(25,5),PLACE(25),WALLS(4),TWALLS(4)
      INTEGER NPLAC,SCALE,IPLOT(25,5)
      IOLD=SCALE
      SCALE=SCALE+1
      XOLD = FX
      YOLD = FY
      FXP=FX+25.0
      FYP = (FY*SCALE)/2.0
      DX=FXP-XOLD
      DY=FYP-YOLD
      DO 125 JBA=1,25
      IF (PLACE(JBA) .EQ. 0) GO TO 125
      IF (IPLOT(JBA,4) .EQ. 4) GO TO 25
      PLOT(JBA,1) = PLOT(JBA,1) +DX
      PLOT(JBA,2) = PLOT(JBA,2) +DY
 25   IF (IPLOT(JBA,5) .EQ. 0) GO TO 125
      PLOT(JBA,3) = PLOT(JBA,3) +DX
      PLOT(JBA,4) = PLOT(JBA,4) +DY
 125  CONTINUE
      IF (NPLAC .EQ. 1) GO TO 300
      TWALLS(1) = TWALLS(1) +DY
      TWALLS(2) = TWALLS(2) +DY
```

```
      TWALLS(3) = TWALLS(3) +DX
      TWALLS(4) = TWALLS(4) +DX
300   WALLS(1) = WALLS(1) +DY
      WALLS(2) = WALLS(2) +DY
      WALLS(3) = WALLS(3) +DX
      WALLS(4) = WALLS(4) +DX
      RETURN
      END
      SUBROUTINE DUSPAC
      PRINT *
      PRINT *
      RETURN
      END
      SUBROUTINE TRISPAC
      PRINT *
      PRINT *
      PRINT *
      RETURN
      END
      SUBROUTINE DWALLS(XA,YA,B,N,NPTS)
      REAL B(4),N(4,2)
      INTEGER NPTS(4)
      CALL CMOPEN
      CALL VWPORT(30.0,130.0,30.0,100.)
      CALL WINDOW(0.0,XA,0.0,YA)
      IF (NPTS(1) .EQ. 1) GO TO 10
      CALL MOVE(B(4),B(1))
      GO TO 20
10    CALL MOVE(N(1,1),N(1,2))
      CALL DRAW(N(1,1),B(1))
C

20    IF (NPTS(2) .EQ. 1) GO TO 30
      CALL DRAW(B(3),B(1))
      GO TO 40
30    CALL DRAW(N(2,1),B(1))
      CALL DRAW(N(2,1),N(2,2))
      CALL DRAW(B(3),N(2,2))
C
C
40    IF (NPTS(3) .EQ. 1) GO TO 50
      CALL DRAW(B(3),B(2))
      GO  TO 60
50    CALL DRAW(B(3),N(3,2))
      CALL DRAW(N(3,1),N(3,2))
      CALL DRAW(N(3,1),B(2))
C
C
60    IF (NPTS(4) .EQ. 1) GO TO 70
      CALL DRAW(B(1),B(2))
      GO TO 80
```

```
 70    CALL DRAW(N(4,1),B(2))
       CALL DRAW(N(4,1),N(4,2))
       CALL DRAW(B(4),N(4,2))
C
C
 80    IF (NPTS(1) .EQ. 1) GO TO 90
       CALL DRAW(B(4),B(1))
       GO TO 100
 90    CALL DRAW(B(4),N(4,2))
       CALL DRAW(N(4,1),N(4,2))
C
C
100    CALL DASHPT(3)
       CALL MOVE(B(4),B(1))
       CALL DRAW(B(3),B(1))
       CALL DRAW(B(3),B(2))
       CALL DRAW(B(4),B(2))
       CALL DRAW(B(4),B(1))
       CALL DASHPT(0)
C
       CALL CMCLOS
       RETURN
       END
       SUBROUTINE DWALLSO(XA,YA,B,N,NPTS)
       REAL B(4),N(4,2)
       INTEGER NPTS(4)
       CALL CMOPEN
       CALL VWPORT(0.0,130.0,0.0,100.)
       CALL WINDOW(0.0,XA,0.0,YA)
       IF (NPTS(1) .EQ. 1) GO TO 10
       CALL MOVE(B(4),B(1))
       GO TO 20
 10    CALL MOVE(N(1,1),N(1,2))
       CALL DRAW(N(1,1),B(1))


 20    IF (NPTS(2) .EQ. 1) GO TO 30
       CALL DRAW(B(3),B(1))
       GO TO 40
 30    CALL DRAW(N(2,1),B(1))
       CALL DRAW(N(2,1),N(2,2))
       CALL DRAW(B(3),N(2,2))


 40    IF (NPTS(3) .EQ. 1) GO TO 50
       CALL DRAW(B(3),B(2))
       GO  TO 60
 50    CALL DRAW(B(3),N(3,2))
       CALL DRAW(N(3,1),N(3,2))
       CALL DRAW(N(3,1),B(2))
C
```

```
C
   60   IF (NPTS(4) .EQ. 1) GO TO 70
        CALL DRAW(B(4),B(2))
        GO TO 80
   70   CALL DRAW(N(4,1),B(2))
        CALL DRAW(N(4,1),N(4,2))
        CALL DRAW(B(4),N(4,2))
C
C
   80   IF (NPTS(1) .EQ. 1) GO TO 90
        CALL DRAW(B(4),B(1))
        GO TO 100
   90   CALL DRAW(B(4),N(4,2))
        CALL DRAW(N(4,1),N(4,2))
C
C
  100   CALL DASHPT(3)
        CALL MOVE(B(4),B(1))
        CALL DRAW(B(3),B(1))
        CALL DRAW(B(3),B(2))
        CALL DRAW(B(4),B(2))
        CALL DRAW(B(4),B(1))
        CALL DASHPT(0)
C
        CALL CMCLOS
        RETURN
        END
        SUBROUTINE DWCTR(XA,YA,B,N,P,PTS,NPTS,A,PV,RTE,IPE)
        REAL A(2),B(2,4),N(2,4,2),P(2,2)
        INTEGER PTS(2),NPTS(4),PV
        IAZ = 0
        CALL CMOPEN
        CALL VWPORT(39.0,130.0,30.0,100.)
        CALL WINDOW(0.0,XA,0.0,YA)
        CALL TRANSL(A(1),A(2))
        CALL ROTATE(RTE,RTE)
        IF (NPTS(1) .EQ. 1) GO TO 10
        CALL MOVE(B(PV,4),B(PV,1))
        GO TO 20
   10   CALL MOVE(N(PV,1,1),N(PV,1,2))
        CALL DRAW(N(PV,1,1),B(PV,1))
C
C
   20   IF (NPTS(2) .EQ. 1) GO TO 30
        CALL DRAW(B(PV,3),B(PV,1))
        GO TO 40
   30   CALL DRAW(N(PV,2,1),B(PV,1))
        CALL DRAW(N(PV,2,1),N(PV,2,2))
        CALL DRAW(B(PV,3),N(PV,2,2))
C
C
```

```
   40   IF (NPTS(3) .EQ. 1) GO TO 50
        CALL DRAW(B(PV,3),B(PV,2))
        GO  TO 60
   50   CALL DRAW(B(PV,3),N(PV,3,2))
        CALL DRAW(N(PV,3,1),N(PV,3,2))
        CALL DRAW(N(PV,3,1),B(PV,2))
C
C
   60   IF (NPTS(4) .EQ. 1) GO TO 70
        CALL DRAW(B(PV,4),B(PV,2))
        GO TO 80
   70   CALL DRAW(N(PV,4,1),B(PV,2))
        CALL DRAW(N(PV,4,1),N(PV,4,2))
        CALL DRAW(B(PV,4),N(PV,4,2))
C
C
   80   IF (NPTS(1) .EQ. 1) GO TO 90
        CALL DRAW(B(PV,4),B(PV,1))
        GO TO 100
   90   CALL DRAW(B(PV,4),N(PV,4,2))
        CALL DRAW(N(PV,4,1),N(PV,4,2))
C
C
  100   DO 95 IAX =1,4
   95   IF (NPTS(IAX) .EQ. 1) IAZ=1
        IF (IAZ .EQ. 0) GO TO 888
        CALL DASHPT(3)
        CALL MOVE(B(PV,4),B(PV,1))
        CALL DRAW(B(PV,3),B(PV,1))
        CALL DRAW(B(PV,3),B(PV,2))
        CALL DRAW(B(PV,4),B(PV,2))
        CALL DRAW(B(PV,4),B(PV,1))
        CALL DASHPT(0)
C
C
  888   IF (IPE .GT. 0) GO TO 999
        IF (PTS(1) .EQ. 0) GO TO 120
        IF (PV .EQ. 1) GO TO 110
        CALL MOVE(P(1,1)+10.,P(1,2)+10.)
        CALL DRAW(P(1,1)-10.,P(1,2)-10.)
        CALL MOVE(P(1,1)+10.,P(1,2)-10.)
        CALL DRAW(P(1,1)-10.,P(1,2)+10.)
        GO TO 120
  110   CALL MOVE(5.,5.)
        CALL DRAW(-5.,-5.)
        CALL MOVE(5.0,-5.)
        CALL DRAW(-5.,5.0)
C
C
  120   IF(PTS(2) .EQ. 0) GO TO 140
        IF(PV .EQ. 2) GO TO 150
```

```
      CALL MOVE(P(2,1)+10.,P(2,2)+10.)
      CALL DRAW(P(2,1)-10.,P(2,2)-10.)
      GO TO 140
 150  CALL MOVE(5.,5.)
      CALL DRAW(-5.,-5.)
 140  CALL TRIDNT(.FALSE.)
      CALL CMCLOS
 999  RETURN
      END
      SUBROUTINE DWCTRO(XA,YA,B,N,P,PTS,NPTS,A,PV,RTE,IPE)
      REAL A(2),B(2,4),N(2,4,2),P(2,2)
      INTEGER PTS(2),NPTS(4),PV
      IAZ = 0
      CALL CMOPEN
      IF (NPTS(1) .EQ. 1) GO TO 10
      CALL MOVE(B(PV,4),B(PV,1))
      GO TO 20
  10  CALL MOVE(N(PV,1,1),N(PV,1,2))
      CALL DRAW(N(PV,1,1),B(PV,1))
C
C
  20  IF (NPTS(2) .EQ. 1) GO TO 30
      CALL DRAW(B(PV,3),B(PV,1))
      GO TO 40
  30  CALL DRAW(N(PV,2,1),B(PV,1))
      CALL DRAW(N(PV,2,1),N(PV,2,2))
      CALL DRAW(B(PV,3),N(PV,2,2))
C
C
  40  IF (NPTS(3) .EQ. 1) GO TO 50
      CALL DRAW(B(PV,3),B(PV,2))
      GO TO 60
  50  CALL DRAW(B(PV,3),N(PV,3,2))
      CALL DRAW(N(PV,3,1),N(PV,3,2))
      CALL DRAW(N(PV,3,1),B(PV,2))
C
C
  60  IF (NPTS(4) .EQ. 1) GO TO 70
      CALL DRAW(B(PV,4),B(PV,2))
      GO TO 80
  70  CALL DRAW(N(PV,4,1),B(PV,2))
      CALL DRAW(N(PV,4,1),N(PV,4,2))
      CALL DRAW(B(PV,4),N(PV,4,2))
C
C
  80  IF (NPTS(1) .EQ. 1) GO TO 90
      CALL DRAW(B(PV,4),B(PV,1))
      GO TO 100
  90  CALL DRAW(B(PV,4),N(PV,4,2))
      CALL DRAW(N(PV,4,1),N(PV,4,2))
C
```

```
C
  100   DO 75 IAX =1,4
   75   IF (NPTS(IAX) .EQ. 1) IAZ=1
        IF (IAZ .EQ. 0) GO TO 888
        CALL DASHPT(3)
        CALL MOVE(B(PV,4),B(PV,1))
        CALL DRAW(B(PV,3),B(PV,1))
        CALL DRAW(B(PV,3),B(PV,2))
        CALL DRAW(B(PV,4),B(PV,2))
        CALL DRAW(B(PV,4),B(PV,1))
        CALL DASHPT(0)
C
C
  888   IF (IPE .GT. 0) GO TO 999
        IF (PTS(1) .EQ. 0) GO TO 120
        IF (PV .EQ. 1) GO TO 110
        CALL MOVE(P(1,1)+10.,P(1,2)+10.)
        CALL DRAW(P(1,1)-10.,P(1,2)-10.)
        CALL MOVE(P(1,1)+10.,P(1,2)-10.)
        CALL DRAW(P(1,1)-10.,P(1,2)+10.)
        GO TO 120
  110   CALL MOVE(5.,5.)
        CALL DRAW(-5.,-5.)
        CALL MOVE(5.0,-5.)
        CALL DRAW(-5.,5.0)
C
C
  120   IF(PTS(2) .EQ. 0) GO TO 999
        IF(PV .EQ. 2) GO TO 150
        CALL MOVE(P(2,1)+10.,P(2,2)+10.)
        CALL DRAW(P(2,1)-10.,P(2,2)-10.)
        GO TO 999
  150   CALL MOVE(5.,5.)
        CALL DRAW(-5.,-5.)
  999   CALL TRIDNT(.FALSE.)
        CALL CMCLOS
        RETURN
        END
        SUBROUTINE FSPAC
        CALL CMOPEN
        CALL HOME
        CALL CMCLOS
        DO 10 II = 1,25
        PRINT *
   10   CONTINUE
        RETURN
        END
        SUBROUTINE HDGA(PH,NFAC,PSM)
        INTEGER PH
        CHARACTER *20 NFAC
        CHARACTER *7 PSM(4)
```

```
      PGM(2)='WORKOUT'
      PGM(1)='FLOWS'
      PGM(3)='LAYOUT'
      PGM(4)='OUTPUT'
      CALL CLEAR
      CALL CMOPEN
      CALL TRIDNT(.TRUE.)
      CALL MOVE(0.0,90.0)
      CALL DRAW(130.0,90.0)
      CALL DRAW(130.0,100.0)
      CALL DRAW(0.0,100.0)
      CALL DRAW(0.0,90.0)
      CALL MOVE (0.0,130.0)
      CALL HOME
      CALL CMCLOS
 1    FORMAT(3X,A20,40X,'FLAG')
 2    FORMAT(3X,'PHASE: ',A7)
      PRINT 1,NFAC
      PRINT 2,PGM(PH)
      RETURN
      END
      SUBROUTINE LOAD(NL,BFA,CFA,PFA,NFA,NBPO,
     1 OLD,PTS,NPTS)
      REAL PFA(2,2),OLD(2,2),BFA(2,4),CFA(2,2)
      REAL NFA(2,4,2)
      INTEGER PTS(2),NPTS(4),NBPO(2)
      CHARACTER *79 Z
 1    FORMAT(1X,A79)
 10   FORMAT(1X,2(I2,2X))
 11   FORMAT(1X,2(F10.3,2X))
 12   FORMAT(1X,4(F10.3,2X))
 13   FORMAT(1X,4(I2,2X))
      OPEN(UNIT=3,FILE='BORD.DAT',TYPE='OLD',
     1 DISP='KEEP',FORM='FORMATTED')
      IF (NL .EQ. 1) GO TO 50
      NPASS=NL-1
      NREC= (NPASS*21)
      DO 45 JAAA=1,NREC
 45   READ (3,1) Z
 50   READ (3,1) Z
      READ(3,1) Z
      READ(3,10) (PTS(JA),JA=1,2)
      READ(3,10) (NBPO(JA),JA=1,2)
      DO 150 JA=1,2
 150  READ(3,11) (OLD(JA,JB),JB=1,2)
      DO 160 JA=1,2
 160  READ(3,11) (PFA(JA,JB),JB=1,2)
      DO 165 JA = 1,2
 165  READ(3,12) (BFA(JA,JB),JB=1,4)
      DO 170 JA = 1,2
 170  READ(3,11) (CFA(JA,JB),JB=1,2)
```

```
      READ(3,13) (NPTS(JA),JA=1,4)
      DO 180 JA=1,2
      DO 175 JB=1,4
175   READ(3,11) (NFA(JA,JB,JC),JC=1,2)
180   CONTINUE
      CLOSE (UNIT=3)
      RETURN
      END
      SUBROUTINE NPOINT(PTSS,NPAA,WHICH)
      INTEGER PTSS(2) ,NPAA,WHICH
      WHICH=0
      NPAA=0
      IF (PTSS(1) .EQ. 1) NPAA =NPAA+1
      IF (PTSS(2) .EQ. 1) NPAA = NPAA+1
      IF (NPAA .EQ. 2) GO TO 100
      IF (PTSS(1) .EQ. 1) WHICH = 1
      IF (PTSS(2) .EQ. 2) WHICH = 2
100   RETURN
      END
      SUBROUTINE POINTS(PLOT,NMWTR,PLACE)
      CHARACTER * 15 NMWTR(25)
      REAL PLOT(25,5)
      INTEGER PLACE(25)
 1    FORMAT(25X,'POD AND POE POSITIONS')
 2    FORMAT(1X,'NO.',5X,'DEPT. NAME',9X,'POD-X',10X,
  1   'POD-Y',10X,'POE-X',10X,'POE-Y')
 3    FORMAT(1X,I2,3X,A15,2(3X,F10.3,2X,F10.3))
      PRINT *
      PRINT 1
      PRINT *
      PRINT 2
      DO 100 KAB = 1,25
      IF(PLACE(KAB) .EQ. 0) GO TO 100
      PRINT 3,KAB,NMWTR(KAB),PLOT(KAB,3),PLOT(KAB,4),
  1   PLOT(KAB,1),PLOT(KAB,2)
100   CONTINUE
      RETURN
      END
      SUBROUTINE PSTORE(BF,CF,PF,NF,NBPG,OLD)
      REAL BF(4)
      REAL CF(2), PF(2,3),NF(4,4)
      REAL PFA(2,2), OLD(2,2), BFA(2,4)
      REAL CFA(2,2), NFA(2,4,2)
      INTEGER POINTS(2),NPOINT(4),NBPG(2)
 10   FORMAT(1X,2(I2,2X))
 11   FORMAT(1X,2(F10.3,2X))
 12   FORMAT(1X,4(F10.3,2X))
 13   FORMAT(1X,4(I2,2X))
```

```
      OLD(1,1)=PF(1,2)
      OLD(1,2)=PF(1,3)
      OLD(2,1)=PF(2,2)
      OLD(2,2)=PF(2,3)
      POINTS(1) = 0
      POINTS(2) = 0
      IF (PF(1,1) .EQ. 1.0) POINTS(1) = 1
      IF (PF(2,1) .EQ. 1.0) POINTS(2) = 1
      DO 50 JA =1,2
      DO 45 JB =1,2
45    PFA(JA,JB) = 0.0
50    CONTINUE
      IF (POINTS(1) .EQ. 0) GO TO 55
      IF (POINTS(2) .EQ. 0) GO TO 55
      PFA(1,1) = OLD(1,1) - OLD(2,1)
      PFA(1,2) = OLD(1,2) - OLD(2,2)
      PFA(2,1) = OLD(2,1) - OLD(1,1)
      PFA(2,2) = OLD(2,2) - OLD(1,2)
55    DO 70 JA = 1,2
      DO 65 JB = 1,4
65    BFA(JA,JB) = 0.0
      DO 67 JB = 1,2
67    CFA(JA,JB) =0.0
70    CONTINUE
      IF (POINTS(1) .EQ. 0) GO TO 80
      BFA(1,1) = BF(1) - OLD(1,2)
      BFA(1,2) = BF(2) - OLD(1,2)
      BFA(1,3) = BF(3) - OLD(1,1)
      BFA(1,4) = BF(4) - OLD(1,1)
      CFA(1,1) = CF(1) - OLD(1,1)
      CFA(1,2) = CF(2) - OLD(1,2)
80    IF (POINTS(2) .EQ. 0) GO TO 85
      BFA(2,1) = BF(1) - OLD(2,2)
      BFA(2,2) = BF(2) - OLD(2,2)
      BFA(2,3) = BF(3) - OLD(2,1)
      BFA(2,4) = BF(4) - OLD(2,1)
      CFA(2,1) = CF(1) - OLD(2,1)
      CFA(2,2) = CF(2) - OLD(2,2)
85    DO 90 JA=1,4
90    NPOINT(JA) = 0
      DO 95 JA=1,4
95    IF (NF(1,JA) .EQ. 1.0) NPOINT(JA) = 1
      DO 105 JA=1,4
      DO 100 JB=1,2
      DO 99 LC=1,2
99    NFA(LC,JA,JB) = 0.0
100   CONTINUE
105   CONTINUE
      IF (POINTS(1) .EQ. 0) GO TO 125
      DO 120 JA=1,4
      IF (NPOINT(JA) .EQ. 0) GO TO 120
```

```
      NFA(1,JA,1)=NF(2,JA)-OLD(1,1)
      NFA(1,JA,2)=NF(3,JA)-OLD(1,2)
120   CONTINUE
125   IF (POINTS(2) .EQ. 0) GO TO 140
      DO 130 JA = 1,4
      IF (NPOINT(JA) .EQ. 0) GO TO 130
      NFA(2,JA,1) = NF(2,JA) - OLD(2,1)
      NFA(2,JA,2) = NF(3,JA) - OLD(2,2)
130   CONTINUE
140   WRITE(3,10) (POINTS(JA),JA=1,2)
      WRITE(3,10) (NBPO(JA),JA=1,2)
      DO 150 JA =1,2
150   WRITE(3,11) (OLD(JA,JB),JB=1,2)
      DO 160 JA = 1,2
160   WRITE(3,11) (PFA(JA,JB),JB=1,2)
      DO 165 JA = 1,2
165   WRITE(3,12) (BFA(JA,JB),JB=1,4)
      DO 170 JA = 1,2
170   WRITE(3,11) (CFA(JA,JB),JB=1,2)
      WRITE(3,13) (NPOINT(JA),JA=1,4)
      DO 180 JA = 1,2
      DO 175 JB = 1,4
175   WRITE(3,11) (NFA(JA,JB,JC),JC=1,2)
180   CONTINUE
      RETURN
      END
      SUBROUTINE RANK(PLACE,FT,RLIST,RVAL,NWCTR)
      INTEGER PLACE(25)
      INTEGER RLIST(10,2),TLIST(10,2),RPLAC(10,25)
      REAL FT(25,25),RVAL(25),TVAL(25)
      DO 10 IAB=1,10
      RVAL(IAB)=0.0
      TVAL(IAB)=0.0
      DO 4 IAC =1,25
  4   RPLAC(IAB,IAC)=0
      DO 5 IAC=1,2
      RLIST(IAB,IAC)=0
  5   TLIST(IAB,IAC)=0
 10   CONTINUE
      NRANK=0
      DO 150 IAB=1,NWCTR
      DO 145 IAC=1,NWCTR
      IF (FT(IAB,IAC) .EQ. 0.0) GO TO 145
      IF (IAB .EQ. IAC) GO TO 145
      IF (RPLAC(IAB,IAC) .EQ. 1) GO TO 145
      IF (PLACE(IAB) .EQ. 0) GO TO 150
      IF (PLACE(IAC) .EQ. 1) GO TO 145
      DO 25 IAD = 1,10
      IF (FT(IAB,IAC) .GT. RVAL(IAD)) GO TO 155
 25   CONTINUE
145   CONTINUE
```

```
150     CONTINUE
        GO TO 400
155     IAT=1
        IAR=1
160     IF (IAT .EQ. IAD) GO TO 175
        TLIST(IAT,1) = RLIST(IAR,1)
        TLIST(IAT,2) = RLIST(IAR,2)
        TVAL(IAT) = RVAL(IAR)
        IAT = IAT +1
        IAR = IAR +1
        IF (IAT .EQ. 11) GO TO 195
        GO TO 160
175     TLIST(IAT,1)=IAB
        TLIST(IAT,2)=IAC
        TVAL(IAT) = FT(IAB,IAC)
        RPLAC(IAB,IAC) = 1
        IAT = IAT +1
        IF (IAT .EQ. 11) GO TO 195
        GO TO 160
195     DO 200 IAK = 1,10
        RLIST(IAK,1)=TLIST(IAK,1)
        RLIST(IAK,2)=TLIST(IAK,2)
        RVAL(IAK)=TVAL(IAK)
        TLIST(IAK,1) = 0
        TLIST(IAK,2) = 0
        TVAL(IAK) = 0.0
200     CONTINUE
        GO TO 145
400     DO 650 IAC = 1,NWCTR
        DO 600 IAB = 1,NWCTR
        IF (PLACE(IAC) .EQ. 0) GO TO 650
        IF (PLACE(IAB) .EQ. 1) GO TO 600
        IF (FT(IAB,IAC) .EQ. 0.0) GO TO 600
        IF(IAB .EQ. IAC) GO TO 600
        IF (RPLAC(IAB,IAC) .EQ. 1) GO TO 600
        DO 425 IAD = 1,10
        IF(FT(IAB,IAC) .GT. RVAL(IAD)) GO TO 450
425     CONTINUE
600     CONTINUE
650     CONTINUE
        GO TO 700
450     IAT = 1
        IAR = 1
460     IF (IAT .EQ. IAD) GO TO 500
        TLIST(IAT,1) = RLIST(IAR,1)
        TLIST(IAT,2) = RLIST(IAR,2)
        TVAL(IAT) =RVAL(IAR)
        IAT = IAT +1
        IAR = IAR +1
        IF (IAT .EQ. 11) GO TO 525
        GO TO 460
```

```
500   TLIST(IAT,1) = IAB
      TLIST(IAT,2) = IAC
      TVAL(IAT) = FT(IAB,IAC)
      RPLAC(IAB,IAC) = 1
      IAT =IAT+1
      IF (IAT .EQ. 11) GO TO 525
      GO TO 460
525   DO 550 IAK = 1,10
      RLIST(IAK,1) = TLIST(IAK,1)
      RLIST(IAK,2) = TLIST(IAK,2)
      RVAL(IAK)=TVAL(IAK)
      TLIST(IAK,1) = 0
      TLIST(IAK,2) = 0
550   TVAL(IAK) = 0.0
      GO TO 600
700   RETURN
      END
      SUBROUTINE RDFT(NPRDT,NWCTR,FT,MIX)
      INTEGER NMPRDT,NWCTR,MIX(5)
      REAL FT(25,25)
 25   FORMAT(1X,F10.3)
 35   FORMAT(1X,I7)
      OPEN(UNIT=3,FILE='FROMTO.DAT',TYPE='OLD',
    1 DISP='KEEP',FORM='FORMATTED')
      DO 50 IA=1,NPRDT
 50   READ(3,35) MIX(IA)
      DO 90 IA=1,NWCTR
      DO 85 IB=1,NWCTR
 85   READ(3,25) FT(IA,IB)
 90   CONTINUE
      CLOSE(UNIT=3)
      RETURN
      END
      SUBROUTINE RDGEN(NFAC,GRPN,NGRP,NMGRP,NPRDT,
    1 NMPRDT,NWCTR,NMWTR,ABNMW)
      INTEGER GRPN,NGRP,NPRDT,NWCTR
      CHARACTER * 20 NFAC
      CHARACTER * 15 NMGRP(5),NMWTR(25)
      CHARACTER * 10 NMPRDT(5)
      CHARACTER *  5 ABNMW(25)
      OPEN(UNIT=1,FILE='GEN.DAT',TYPE='OLD',
    1 DISP='KEEP',FORM='FORMATTED')
    1 FORMAT(1X,A20)
    2 FORMAT(1X,I2,5X,I1)
    3 FORMAT(1X,A15)
    4 FORMAT(1X,I1)
    5 FORMAT(1X,A10)
    6 FORMAT(1X,I2)
    7 FORMAT(1X,A15,5X,A5)
      READ(1,1) NFAC
      READ(1,2) GRPN,NGRP
```

```
      DO 10 IA=1,NGRP
 10   READ(1,3) NMGRP(IA)
      READ(1,4) NPRDT
      DO 15 IA=1,NPRDT
 15   READ(1,5) NMPRDT(IA)
      READ(1,6)  NWCTR
      DO 20 IA=1,NWCTR
 20   READ(1,7) NMWTR(IA),ABNMW(IA)
      CLOSE(UNIT=1)
      RETURN
      END
      SUBROUTINE RELAYT(NWCTR,PLOT,IPLOT,NCORN,BCORN,WALLS)
      REAL PLOT(25,5),NCORN(4,2),WALLS(4)
      INTEGER IPLOT(25,5),BCORN(4)
  1   FORMAT(1X,5(2X,F10.3))
  2   FORMAT(1X,5(2X,I2))
  3   FORMAT(1X,4(2X,F10.3))
  4   FORMAT(1X,4(2X,I2))
  5   FORMAT(1X,2(2X,F10.3))
      OPEN(UNIT =9,FILE='LAYOUT.DAT',TYPE='OLD',
     1 DISP='KEEP',FORM='FORMATTED')
      DO 100 I=1,NWCTR
100   READ(9,1) (PLOT(I,J),J=1,5)
      DO 200 I=1,NWCTR
200   READ(9,2) (IPLOT(I,J),J=1,5)
      READ(9,3) (WALLS(I),I=1,4)
      READ(9,4) (BCORN(I),I=1,4)
      DO 300 I=1,4
300   READ(9,5) (NCORN(I,J),J=1,2)
      CLOSE(UNIT = 9)
      RETURN
      END
      SUBROUTINE RLISTS(PLACE,RLIST,RVAL,NMWTR)
      INTEGER PLACE(25), RLIST(10,2)
      REAL RVAL(10)
      CHARACTER * 15 NMWTR(25)
  1   FORMAT(5X,I2,5X,2(I2,2X),2X,A15,3X,F10.3)
  2   FORMAT(25X,'RANKING OF UNPLACED WORKCENTERS')
      PRINT *
      PRINT 2
      PRINT *
      IAK=1
      IF (PLACE(RLIST(1,2)) .EQ. 1) GO TO 15
      PRINT 1,IAK,RLIST(1,1),RLIST(1,2),NMWTR(RLIST(1,2)),
     1 RVAL(1)
      GO TO 20
 15   PRINT 1,IAK,RLIST(1,1),RLIST(1,2),NMWTR(RLIST(1,1)),
     1 RVAL(1)
 20   PRINT *
      DO 50 IAJ = 2,10
      IF (RVAL(IAJ) .EQ. 0.0) GO TO 999
```

```
      IF (PLACE(RLIST(IAJ,1)) .EQ. 1) GO TO 25
      PRINT 1,IAJ,RLIST(IAJ,1),RLIST(IAJ,2),
    1 NMWTR(RLIST(IAJ,1)),RVAL(IAJ)
      GO TO 50
 25   PRINT 1,IAJ,RLIST(IAJ,1),RLIST(IAJ,2),
    1 NMWTR(RLIST(IAJ,2)),RVAL(IAJ)
 50   CONTINUE
999   RETURN
      END
      SUBROUTINE SLAYT(NWCTR,PLOT,IPLOT,NCORN,BCORN,WALLS)
      REAL PLOT(25,5),NCORN(4,2),WALLS(4)
      INTEGER IPLOT(25,5),BCORN(4)
    1 FORMAT(1X,5(2X,F10.3))
    2 FORMAT(1X,5(2X,I2))
    3 FORMAT(1X,4(2X,F10.3))
    4 FORMAT(1X,4(2X,I2))
    5 FORMAT(1X,2(2X,F10.3))
      OPEN(UNIT =9,FILE='LAYOUT.DAT',TYPE='NEW',
    1 DISP='KEEP',FORM='FORMATTED')
      DO 100 I=1,NWCTR
100   WRITE(9,1) (PLOT(I,J),J=1,5)
      DO 200 I=1,NWCTR
200   WRITE(9,2) (IPLOT(I,J),J=1,5)
      WRITE(9,3) (WALLS(I),I=1,4)
      WRITE(9,4) (BCORN(I),I=1,4)
      DO 300 I=1,4
300   WRITE(9,5) (NCORN(I,J),J=1,2)
      CLOSE(UNIT = 9)
      RETURN
      END
      SUBROUTINE TOP
      CALL CMOPEN
      CALL HOME
      CALL CMCLOS
      RETURN
      END
      SUBROUTINE TRISPAC
      PRINT *
      PRINT *
      PRINT *
      RETURN
      END
      SUBROUTINE WARN(N)
      CALL CMOPEN
      DO 10 I = 1,N
      CALL BELL
10    CONTINUE
      CALL CMCLOS
      RETURN
      END
```

XIII. ACKNOWLEDGEMENTS

I wish to express my appreciation to my major professor Dr. Eric Malstrom for all of the guidance he has given me in completing this project. I also would like to thank the other members of my committee, Dr. Roger Berger, Dr. Keith McRoberts, and Dr. Dennis Starleaf for their time and counsel.

I am also grateful to Robert Nutting and Rebecca Shivvers, who assisted me greatly in completing my work in absentia.

Finally, I wish to thank my father, Wm. W. Ketcham, for his continued support throughout my college years.