HIDECS 2: A COMPUTER PROGRAM FOR THE HIERARCHICAL DECOMPOSITION OF A SET WITH AN ASSOCIATED LINEAR GRAPH

Christopher Alexander
Marvin L. Manheim

RESEARCH REPORT R62-2

DEPAR™ENT OF CIVIL ENGINEERING

CIVIL ENGINEBRING SYSTEMS LABORATORY

HID S 2:
A COMPUTER PROGRAM FOR THE HIERARCHICAL DECOMPOSITION OF A SET WHICH HAS AN ASSOCIATED LINEAR GRAPH

by

Christopher Alexander,

Society of Fellows, Harvard University,

and

Marvin L. Manheim,

Department of Civil Engineering, M. I. T.

Publication No. 160
June, 1962

Sponsored by: Massachusetts Department of Public Works
In cooperation with: U. S. Bureau of Public Roads
Contract 1017 Mass. HPS 1(16)

School of .Engineering
Massachusetts Institute of Technology
Cambridge 39, Massachusetts

Acknowledgements

The research of which this report is a part was supported by the Bureau of Public Roads of the U.S. Department of Commerce and by the Massachusetts Department of Public Works. The computer and associated facilities were made available by the M.I.T. Computation Center.

The work was performed at the Civil Engineering Systems Laboratory at M. I. T. The authors gratefully acknowledge the suggestions and criticisms freely offered by Profes_sors A. Scheffer Lang, Charles L. Miller, and Paulo. Roberts; however, all statements made herein are the responsibility of the authors.

The diagrams were prepared by Miss Candace Allen• of the Civil Engineering Systems Laboratory.

CONTENTS

FIGURES (Cont)

I. INTRODUCTION

A. ABSTRACT AND INTRODUCTION

The program discussed in this report was developed at the Civil Engineering Systems Laboratory, M.I.T. for application to the analysis of several problems in highway engineering. The nature of the analytical methods and of the specifications of the program allow for broad application to other subjects.

The program is used to analyze the structure of a linear graph, or topological one-complex, as it is also termed. Such a graph consists of just two types of elements: vertices, and non-directed links connecting specified pairs of vertices. The input to this program is the matrix description of the graph. According to a criterion derived from assumptions about the graph structure and its information-theoretic properties, the set of vertices can be divided into two or more subsidiary sets of vertices, called subsets, so that each subset still has an associated graph. The program performs such a partition on the input set of vertices, and then in turn partitions the resulting sets. This process is repeated, successivly decomposing the original vertex set into smaller and smaller subsets, until it has been completely decomposed into its constituent vertices.

The set of sets which results from the successive partitions is ordered naturally as a "tree." This tree specifies the order in which the subsets can be recombined to produce the original graph. In the application for which this program

was developed, the vertices of the graph represent the requirements of a design problem. The tree defined by the program's output specifies an order in which the designer should consider the requirements he tries to meet in the process of evolving a design.

This report describes the program, the algorithms upon which it is based, operational procedures, and certain possible modifications.

B. APPLICATION OF THE PROGRAM

This program was developed for use in the analysis of certain types of design problems, according to the theory put forward by Christopher Alexander, in "NOTES ON THE SYNTHESIS OF FORM" (Ph.D. thesis, Harvard University, 1962). The theory has been applied to two highway engineering problems: the design of a highway interchange, and the selection of a location for a highway.*

All design problems contain two kinds of element:

(1) Requirements

(2) Interactions between requirements.

The requirements which the design has to meet are represented as vertices of a graph. The purpose of the design process is to select a design (in the case of our examples, for a highway or an interchange) which fulfills these requirements. The difficulty of achieving a design which satisfies such a list of requirements is due to the fact that requirements conflict with one another: some requirements place demands upon the design which are contradictory to the demands of other requirements. The presence of these interactions between pairs of requirements in a specific design program is represented by links between the vertices corresponding to the particular requirements. The set of vertices, and the set of links together define a graph;

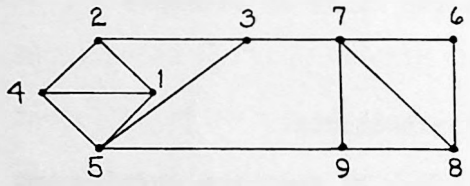
---

*See the reports on these projects by Alexander and Manheim, also issued by the Civil Engineering Systems Laboratory, M.I.T.: THE DESIGN OF HIGHWAY INTERCHANGES: AN EXAMPLE OF A GENERAL METHOD FOR ANALYSING ENGINEERING DESIGN PROBLEMS, and THE USE OF DIAGRAMS IN HIGHWAY ROUTE LOCATION: AN EXPERIMENT.

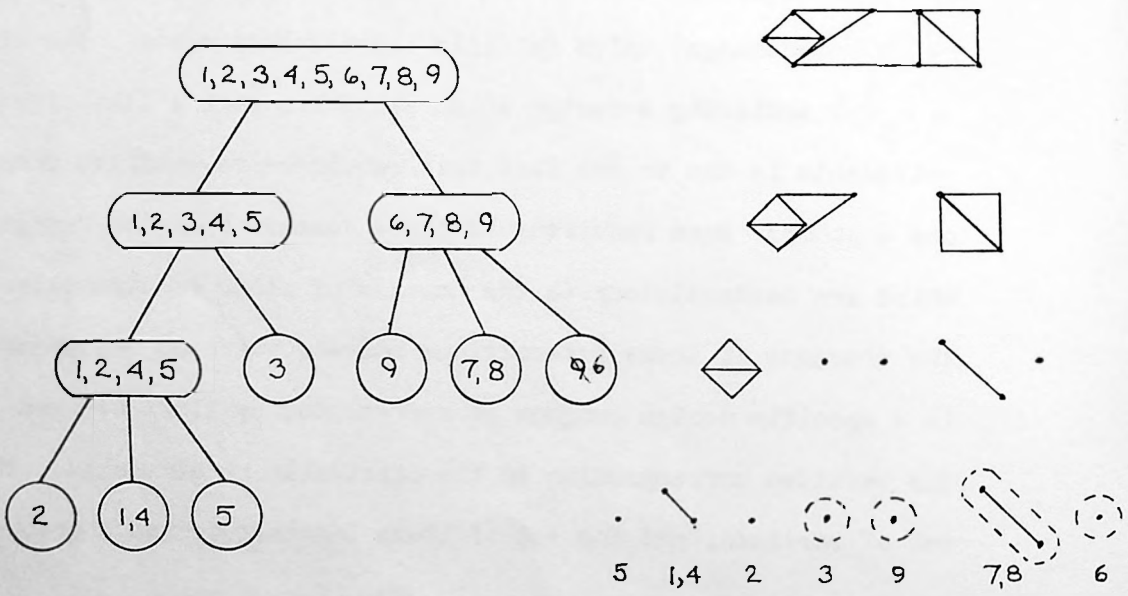
Figure 1

EXAMPLE OF A GRAPH AND ITS TREE

a) A GRAPH



b) ITS TREE



that graph, by virtue of the correspondence between vertices and requirements, and the correspondence between links and interactions of requirements, represents, for the purpose of this analysis, the structure of the design problem.

The input to the program is a graph; the output is a tree, a hierarchical ordering of the graph's vertex set and its partitioned subsets. Because of the correspondence between the graph and the problem, the tree which is obtained by the program provides an orderly scheme for dealing with the requirements posed by a particular problem. The tree specifies which requirements are to be considered together and the order in which different groups of requirements are to be combined and considered. See Figure 1.

C. MACHINE SPECIFICATION

This program has been debugged and run on an IBM 709 at the M.I.T. Computation Center. This machine uses 36-bit words, has a memory capacity of 32,767 words, and has three index registers. The program is designed to be executed under the control of the Fortran Monitor System in use at the Center during the second half of 1961. For information as to peculiarities of the M.I.T. installation which might prove critical in running this program on another machine, see further the M.I.T. Computation Center Procedures Handbook, 1961.

The program has also been used for production runs on an IBM 7090 at the Smithsonian Astrophysical Observatory, Cambridge, under the control of an M.I.T. system tape. No changes in the program were required.

It is not possible to give a rule for estimating the running time required for any specific analysis.

D. REFERENCES

Alexander, Christopher, NOTES ON THE SYNTHESIS OF FORM. Unpublished Ph.D. thesis. Harvard University (1962).

Alexander, Christopher and Marvin L. Manheim, THE DESIGN OF HIGHWAY INTERCHANGES: AN EXAMPLE OF A GENERAL METHOD FOR ANALYSING ENGINEERING DESIGN PROBLEMS. Cambridge, Mass.: Civil Engineering Systems Laboratory, M.I.T. (1962).

_____, THE USE OF DIAGRAMS IN HIGHWAY ROUTE LOCATION: AN EXPERIMENT. Cambridge, Mass.: Civil Engineering Systems Laboratory, M.I.T. (1962).

II. DESCRIPTION OF THE PROGRAM

This description is divided into three major sections. The first describes the operational structure of the program, as comprising a package of subprograms with specific functions. This serves as an introduction to the dictionary of subprograms in Appendix B. The second section of the description discusses the machine representation of a graph. The third section of the description discusses the body of the program, the algorithms actually used in the analysis of graphs. These algorithms are discussed in sufficient detail to introduce the actual program listing, included as Appendix G of the report.

A. GENERAL DESCRIPTION

The program consists of three groups of subprograms. The first, or Preliminary, group prepares for the execution of other subprograms by reading in several parameters, generating other standard parameters, and setting variables which control the execution of particular loops in other subprograms. This group has five subprograms: INPAR, GENER, SET8, SET9, and SET11.

The second, or Data, group of subprograms is concerned with reading in the binary data matrix (representing the graph to be analyzed), and putting it in appropriate form for the analysis. Several different operations are performed by this set of programs: INDAT reads in the data as it is punched on cards, CNDAT converts the inputted data from its input form into the format in which the other subprograms can operate upon it, and SYMET checks the data for inconsistencies which may arise in the pre-computer preparation of the data. With the results of the Preliminary and Data groups of subprograms, the analysis proper can be begun.

The group of subprograms which actually performs the analysis of the graph consists of seven programs actually involved in the analysis, and six which print out the results and comments. LCTRL is the major control program for the Analysis group; it controls the course of the partitioning iterations, the manner of storage of the results, and the selection of one of two sequences of subprograms.

Figure 2

GROUPING OF SUBPROGRAMS

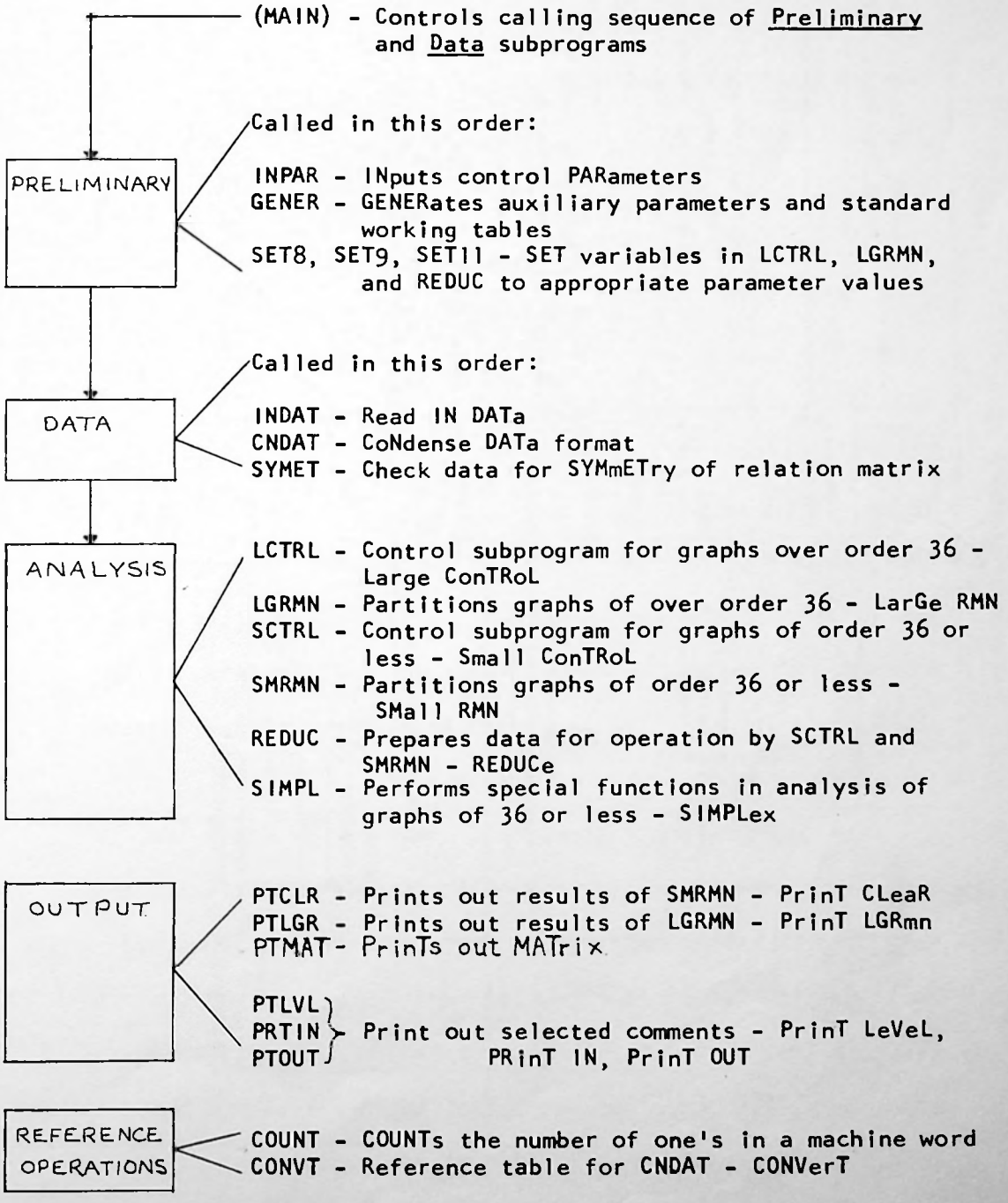


FIGURE 3

SEQUENCE OF DATA OPERATIONS

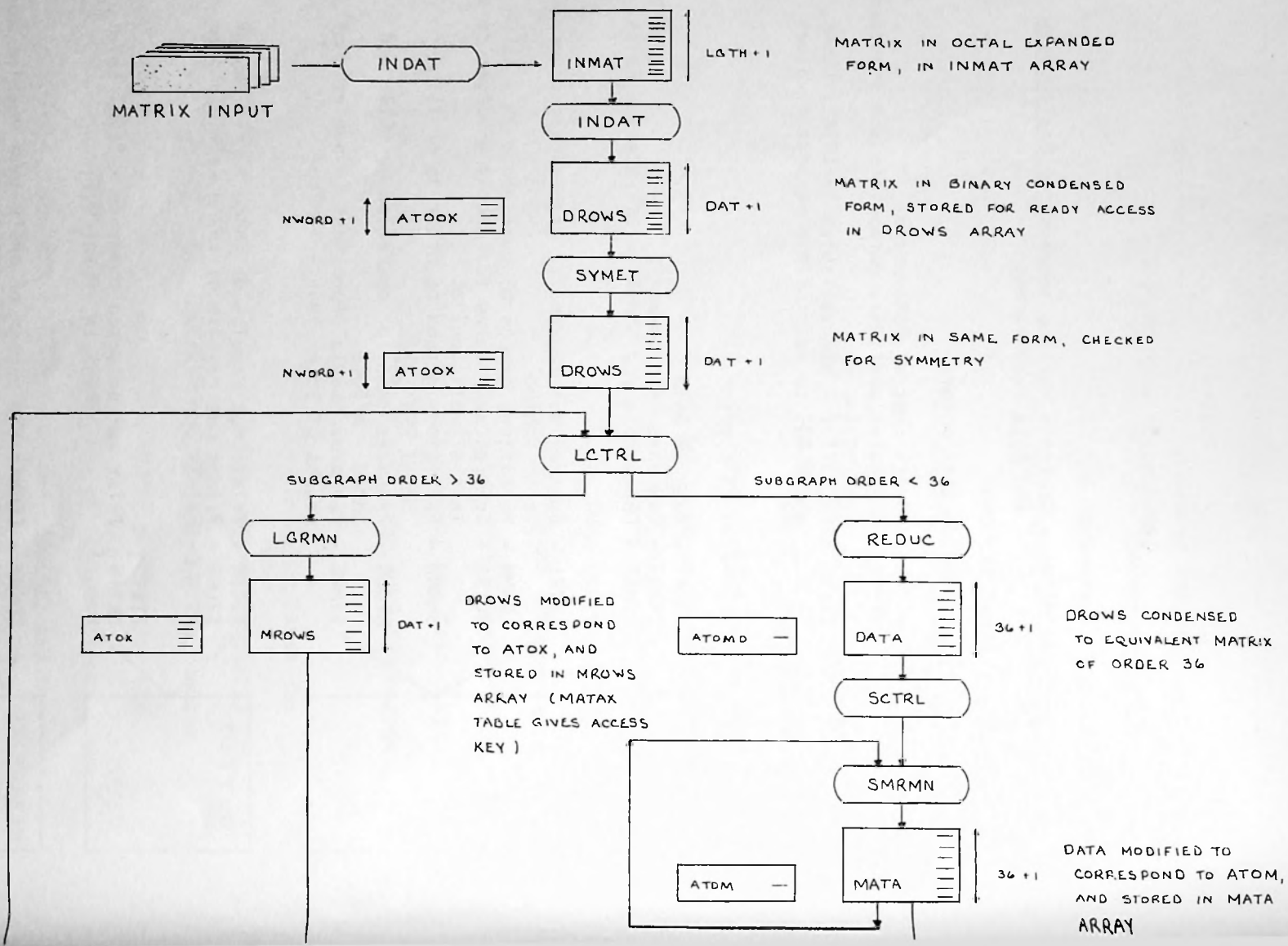
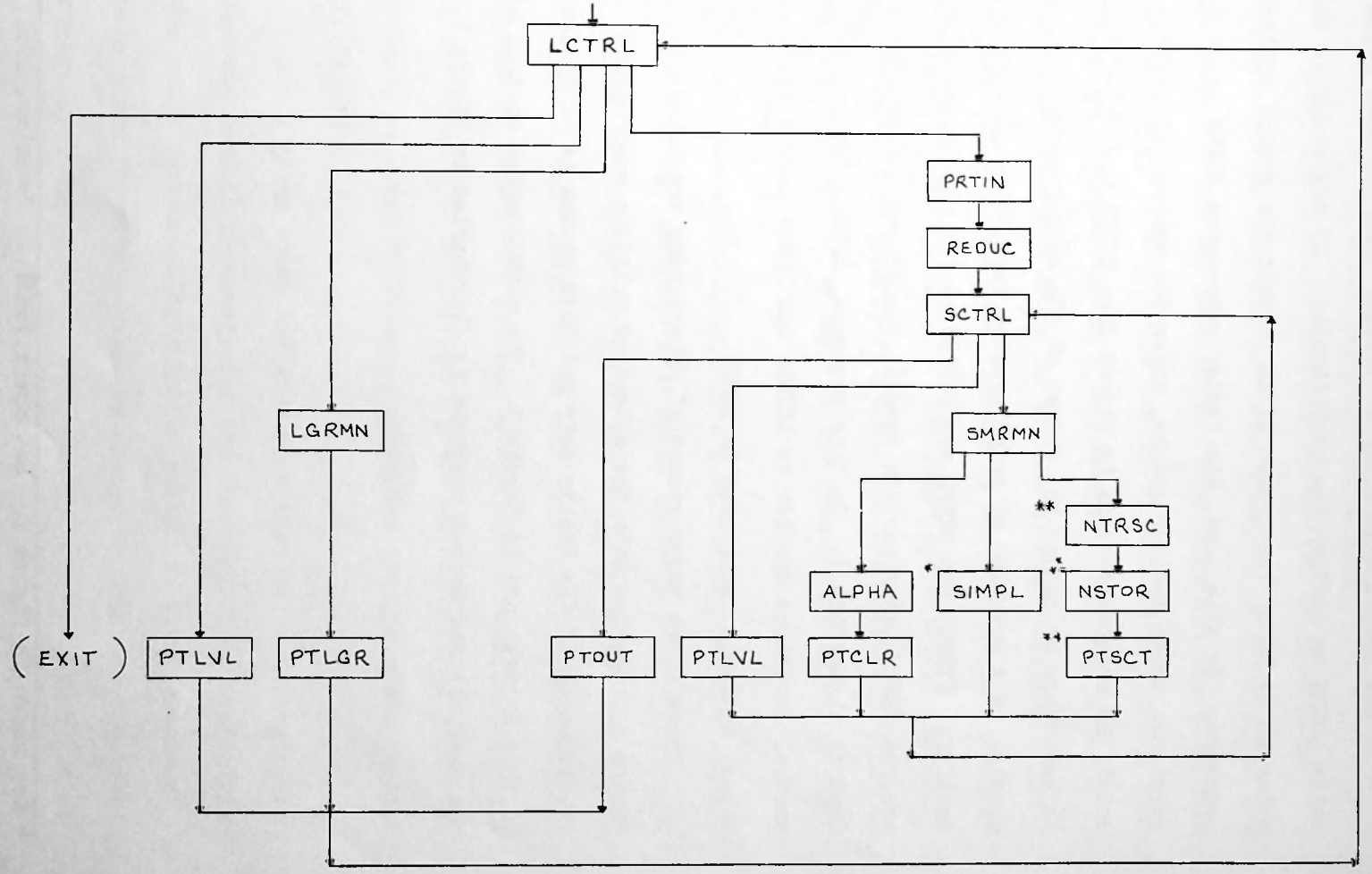


FIGURE 4

THE ANALYSIS SUBPROGRAMS

(ENTRY FROM DATA SUBPROGRAMS)



* = ENTRY POINT IN SCTRL

** = NOT INCORPORATED IN VERSION OF PROGRAM OPERATIONAL IN DECEMBER, 1961

If the subgraph is of order greater than 36, then LCTRL calls LGRMN to perform the partitioning. If the order of the subgraph is 36 or less, then LCTRL selects the second analysis sequence. In this case, the faster subprogram SMRMN is used to partition these smaller graphs, under the control of SCTRL, to which the program control is passed from LCTRL via REDUC. REDUC is necessary to change the format of the data from that corresponding to a subgraph of order greater than 36 to that of one smaller. NTRSC* and SIMPL are called by the partitioning subprogram SMRMN under certain special conditions (Cf. Section 3, Control algorithms.). In this sequence, SCTRL performs the control functions similar to LCTRL, and SMRMN performs the actual analysis, analogous to LGRMN.

These three major groups of subprograms are illustrated in Figure 2, together with the output or printing subprograms. The sequence of the Preliminary and Data groups is as indicated in this Figure, and in Figure 3. The more complex control of the Analysis and Output programs is flowcharted in Figure 4.

* Not shown in Figure 2. Not operational in December, 1961.

B. MACHINE REPRESENTATION

Every graph which contains n vertices is in one-one correspondence with a binary square matrix of order n, in which the i-th row and the i-th column both stand for the i-th vertex of the graph. A one (1) in the ij-th cell of the matrix stands for a link between vertex i and vertex j, and a 0 in the ij-th cell indicates that there is no link between vertices i and j. The principal diagonal of the matrix contains zeros, since no vertex is linked to itself. The links are non-directional, so the matrix is symmetrical about the main diagonal (that is, the entry in the ij-th cell is the same as the entry in the ji-th cell).

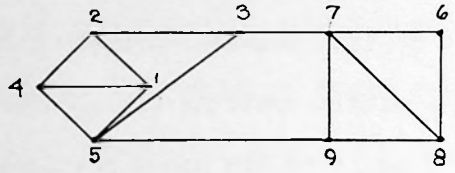
Each row of the matrix is a n-bit binary vector, whose 1's specify the vertices from which there is a link incident on the vertex corresponding to the row. Since the IBM 709 and 7090 have a word length of 36 bits, a single computer word is capable of describing one row of any binary matrix whose order is less than 37. Hence, for a graph which contains 36 vertices or less, an array of 36 words describes the graph completely. Cf. Figure 5.

For a graph which contains more than 36 variables, more than one word is necessary for the description of one matrix row, and a correspondingly larger array, in which blocks of words stand for single rows, is needed to describe the graph completely.

Figure 5

EXAMPLE OF A GRAPH AND ITS MATRIX REPRESENTATION

a) THE GRAPH



b) MATRIX REPRESENTATION

VERTICES:	1	2	3	4	5	6	7	8	9
1	0	1	0	1	1	0	0	0	0
2	1	0	1	1	0	0	0	0	0
3	0	1	0	0	1	0	1	0	0
4	1	1	0	0	1	0	0	0	0
5	1	0	1	1	0	0	0	0	1
6	0	0	0	0	0	0	1	1	0
7	0	0	1	0	0	1	0	1	1
8	0	0	0	0	0	1	1	0	1
9	0	0	0	0	1	0	1	1	0

c) MACHINE REPRESENTATION

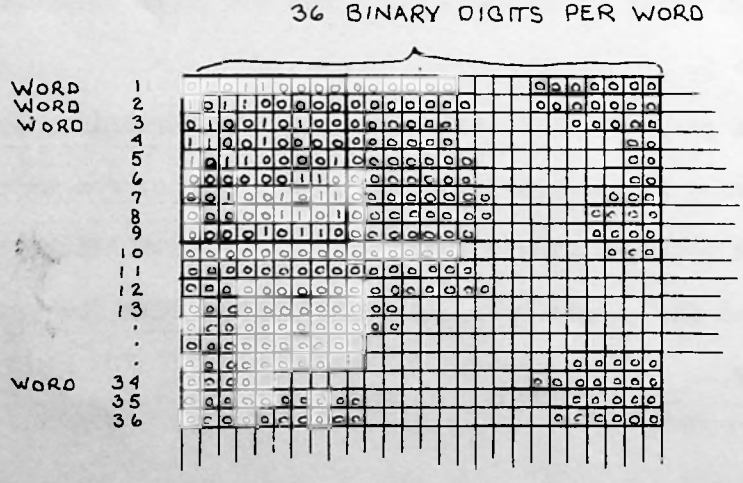
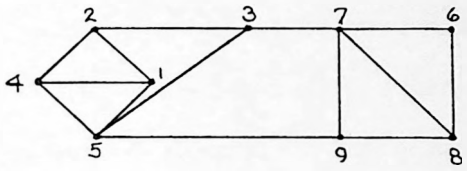


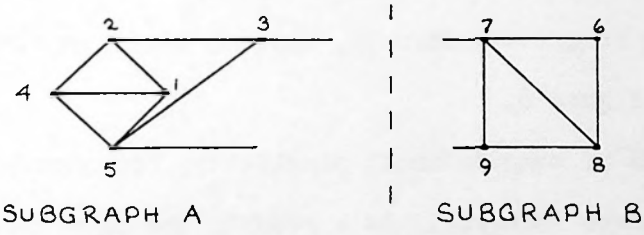
Figure 6

MACHINE REPRESENTATION OF A GRAPH AND ITS SUBGRAPHS

a) THE GRAPH

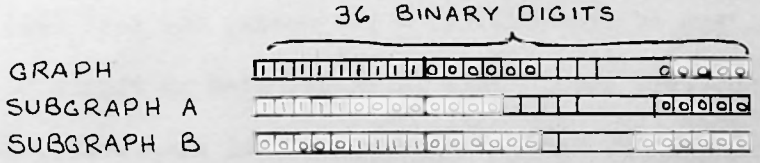


b) A PARTITION OF THE GRAPH INTO TWO SUBGRAPHS



SUBGRAPH A SUBGRAPH B

c) MACHINE REPRESENTATION



NOTE THAT ANY TWO OF THESE THREE WORDS ARE SUFFICIENT TO SPECIFY THE THIRD, SINCE

GRAPH = UNION OF (A, B)

Similarly, when it comes to partitioning the graph, machine words (binary vectors) can be used to describe any given partition. A partition divides the vertex set of a given graph into two or more subsets. If the number of subsets is precisely two (as it is throughout the program*), then the partition is uniquely specified by either of the two subsets, since the other subset is its complement with respect to the set being partitioned. An n-bit binary vector, with 1's indicating the presence of a vertex, and 0's indicating the absence of a vertex, can represent any such set. Again, if the number of vertices in the set being partitioned is not greater than 36, only one machine word is necessary to describe it. If the number of vertices in the set is greater than 36, several words per set are necessary. Cf. Figure 6.

For the sake of computational simplicity, the number of words used is always integral. As a result, the squareness of the matrix means that the matrix storage always contains integral multiples of 36 words. Thus, if the matrix is of order 50, two words per matrix row are used, the last 22 entries in these vectors simply become zeros for the duration of the program, and of the (72).(2) = 144 words, the last (22).(2) = 44 are entirely zero. This is illustrated in Figure 6.

Secondly, again for computational simplicity, and to save computer time, a distinction is made between cases which can

---

*Except for operations with NTRSC and its associated subprograms. Cf. Section 3, Control algorithms.

be dealt with by a single word (i.e. those where the graph contains 36 vertices or less, so that the order of the vectors required is 36 or less), and those cases which cannot be dealt with by single words. Two different sequences of subprograms are used to deal with these two cases and transfer control back and forth between them, as described above.

Finally, to make the use of index registers and indirect addressing as simple as possible, all sequences of stored information are stored backwards, from their symbolic address.

C. DESCRIPTION OF ANALYSIS ALGORITHMS

The algorithms upon which the analysis subprograms are based can be divided into three groups:

1. criterion- the computation of the measure by which the "strength" of a partition is evaluated
2. sampling- the selection of possible partitions to be evaluated
3. control- allocation and record-keeping with regard to storage of partition results; decisions about sequence of partitioning, when to stop partitioning, and printing out of results.

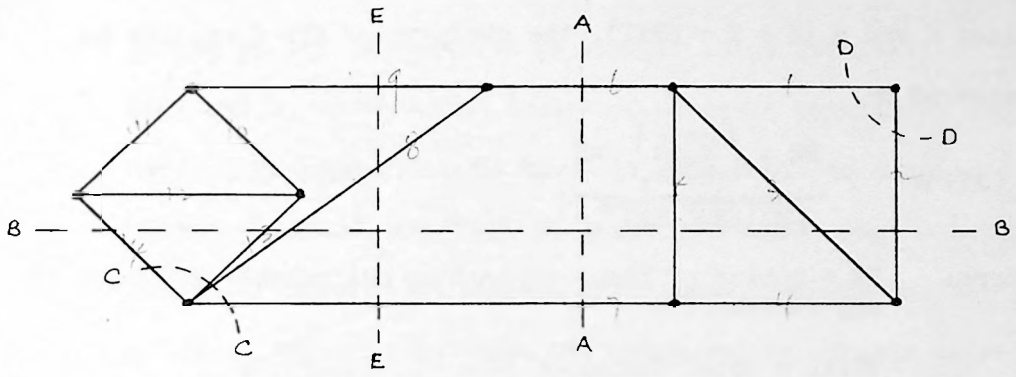
1. The criterion for selecting an optimal partition

The particular criterion upon which this program is based is derived from information - theoretic considerations. The purpose of the criterion is to select one particular decomposition of a graph as best. To discuss the particular criterion, it is necessary to define the basis upon which links are determined.

We assume that each link of the graph represents a statistical correlation between the variables associated with its endpoints (end-vertices). It follows, from considerations of information theory, that the information transmitted from one subset to the other can be used as a criterion for an optimal partition. As shown in Appendix E, it is desired to obtain a partition of the graph's vertex set into two sets which have the least possible information transmitted across the partition.

Figure 7

EXAMPLES OF INFO



NUMBER OF NODES = ORDER = 9

NSQ1 = (9)(8)(1/2) = 36

TOTAL = 14

PARTITION	NO. OF LINKS, RR	NO. OF VERTICES ON EACH SIDE OF PARTITION M	N	INFO	RANKING 1 = BEST
AA	2	5	4	-.104	1
BB	6	3	6	-.003	4
CC	4	1	8	+.004	5
DD	2	1	8	-.006	3
EE	3	4	5	-.071	2

$$STR = \frac{RR - \left(\frac{TOTAL}{NSQ1}\right) MN}{\sqrt{MN\,(NSQ1 - MN)}}$$

$$INFO = (STR)(|STR|)$$

The mathematical expression of this information, and its normalization, as discussed in Appendix E, lead to the following specific measure of the "strength" of any partition: for a partition which divides the set of vertices into two subsets of sizes M and N (M + N = NBIT), the strength of the partition is measured by

$$STR = \frac{RR - \left(\frac{TOTAL}{NSQ1}\right) MN}{\sqrt{MN(NSQ1-MN)}}$$

where: RR = number of links connecting any vertex of M with any vertex of N

TOTAL = total number of links in the graph

NSQ1 = maximum possible number of links in the graph $\left[\frac{NBIT\ (NBIT-1)}{2}\right]$ $\frac{(n)(n-1)}{2}$, n = no of nodes

MN = (M) x (N).

For computational purposes, the actual measure used is INFO, a monotonic function of STR: INFO is the square of STR, but with the sign of STR preserved. Cf. Figure 7.

The program's central algorithm searches for that partition of a graph's vertex set for which INFO is algebraically minimal.*

2. The selection of trial partitions

As shown above, for a given vertex set, a partition is uniquely determined by giving one of its component subsets, since the other subset is always the complement of the first, with respect to the vertex set under consideration. Let INFO be defined for a given subset, as that value of INFO defined for the partition which this subset determines. Then, the task of finding a

*INFO may be negative, and usually is for minimal points.

partition for which INFO is minimum, is the same as the task of finding a subset for which INFO is minimum. However, the number of possible partitions of a vertex set, being $\frac{1}{2}$ the total number of subsets, is, in the case of n vertices, $\frac{1}{2}.2^n = 2^{n-1}$. For graphs of any interest, n is usually large (at least of order 100, say)*, so that the number of possible partitions becomes very large indeed. This makes it impossible to examine every possible subset, and then to select precisely that one for which INFO is minimum. Instead we must somehow sample the set of all possible subsets, and then use these sample subsets as starting points in a hill-climb search procedure.

More precisely, sampling produces a starting trial subset, which is then modified iteratively, by one point at a time, in an attempt to find subsets whose INFO is lower. This continues until no modification of the subset by one vertex improves the value of INFO. The algorithm thus has three components: the choice of a starting subset; its modification under the rule that INFO has to improve as we go along; and the termination of the modification, at that point where no improvement is possible.

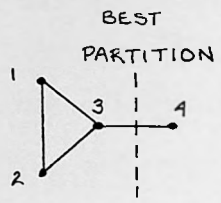
The 2^n possible subsets of the vertex set of n vertices form what is called a lattice. The arrangement of these subsets in the lattice depends upon the simple notion of adjacency between two subsets. Two subsets are called adjacent if one can be made from the other by adding or subtracting a single vertex from it. This is illustrated in Figure 8, where the adjacent

*The analysis of the highway interchange problem used a graph with 112 vertices. Cf. Alexander and Manheim, THE DESIGN OF HIGHWAY INTERCHANGES.

Figure 8

EXAMPLE OF A GRAPH AND ITS LATTICE

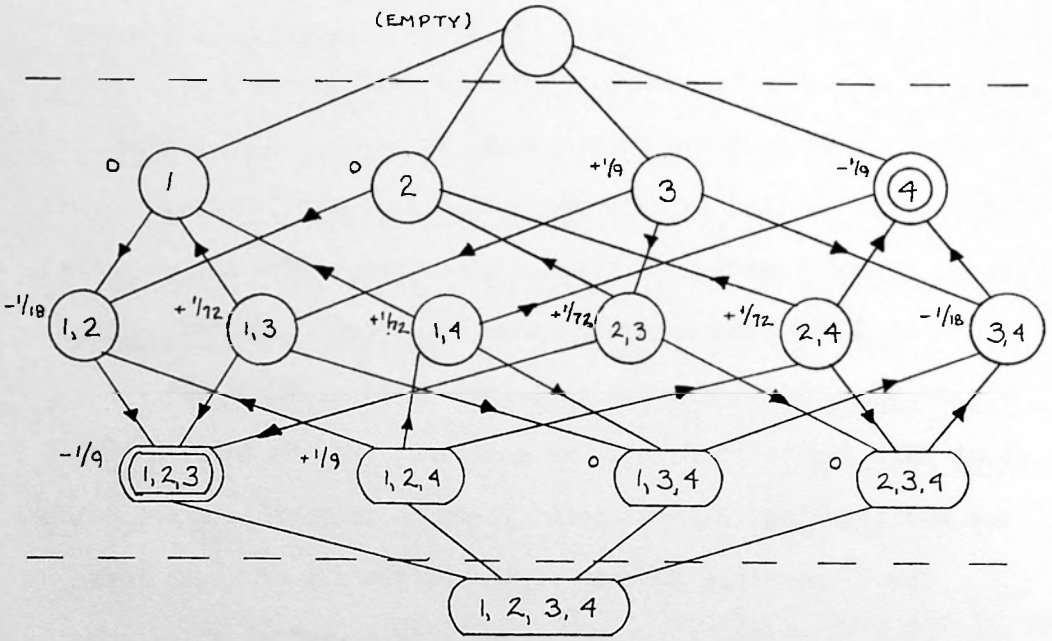
a) GRAPH



b) LATTICE:

(1,2) = SUBGRAPH CONSISTING OF VERTICES 1, 2

= OPTIMAL PARTITION



c) The number beside the subset identification indicates the corresponding value of INFO.
Note that the lattice is the same for all graphs with four vertices. However, the values of INFO (and therefore the arrows) will depend on the specific graph links associated with the four vertices.

subsets are connected by a line. Since every subset has a value of INFO attached to it, we can also associate an arrow with every line in the lattice, showing by its direction which of the two subsets concerned has the lower value of INFO. As a convention, the arrow points toward the subset whose INFO is lower.

The search for better subsets now traces out a path over these lattice lines, always going in the direction of the arrows. There must be subsets which have no arrows leaving them. As soon as the search encounters one of these subsets, it terminates.

There are two points of the lattice which are singular, and must therefore be ignored. These are the full set and the empty set, which both correspond to that imaginary partition that separates the entire set from nothing. Clearly this partition is of no interest. This is expressed mathematically by the fact that for these two subsets, INFO is indeterminate, being 0/0. Arrows cannot be associated with any lattice lines connected to these points; in the program the hill-climbing procedure ignores them.

For the sake of simplicity, we may introduce an analogy, in which subsets are the points of a surface. The altitude of the surface at any point has the value of the INFO of the corresponding subset, and the arrows are always pointing downhill. In this case, the search is equivalent to dropping a ball on the surface, and watching to see where it rolls to.

The analogy makes it clear how critical the choice of starting points can be. The purpose of the search is to find

the very lowest point of the surface. It may well be, however, that the ball finds its way into some valley, not the lowest, but cannot get out again. This is the problem of local minima which occurs in all hill-climbing methods.

In other words the assumption underlying the hill-climbing procedure is that the surface is relatively smooth: that is, the minima whose values are low have correspondingly large "drainage basins," and will therefore be reached from a large number of other points on the surface, while minima whose values are relatively high and undesirable, have relatively small drainage basins. This assumption implies that the best minimum will actually be reached by at least one path, even if the number of starting points is rather small.

There is a difficulty, however. For this procedure to work, the starting points should be equally spaced over the surface. Unfortunately, there is no obvious way of finding points which are equidistantly distributed over a lattice. Finding such a collection of points is equivalent to finding a collection of corners of an n-dimensional cube which are evenly spaced, for edge distance, over the cube. This is a very difficult problem which we have not attempted to solve. Instead, a randomly generated vector is used to select the starting points.*

*The selection of an actual starting point is achieved by taking a given random set of vertices, represented by the octal words RANDM, and adding, each time a new sample starting point is to be generated, another random word, DIFF. The resultant random word(s) are then tested to select those vertices which are in the graph, such that those vertices of the graph which are also in the generated random word(s) become the elements of the starting partition.

The trial subset so generated is called TSET. The path-finding component of the algorithm proceeds by testing each vertex which is not in TSET, adding it to TSET and determining if the set so found would yield a lower INFO. If not, then another vertex is selected and tested for addition. If the tentative modification does achieve a lower INFO, that modification is stored.* All the vertices not already in TSET are tested for addition in this manner, one by one.

Retaining the same TSET, each vertex included in TSET is tested to determine if removing that vertex from TSET would result in a better partition.** The best partition discovered in this subtraction loop is then compared with the best discovered in the addition loop, and the better of these two is compared with the partition represented by TSET. If the addition or subtraction of one vertex results in a partition with a smaller value of the corresponding INFO, that partition replaces TSET, and the procedure is repeated. In this manner, a path through the lattice of possible partitions is traced out by additions or subtractions of one vertex at a time.

---

*Since the addition test selects the vertices in the numerical order of their labels, i.e., 1,2,3,4,... this procedure results in a slight bias towards the lowest-numbered vertices.

**The vertices are considered in order of ascending numerical label in the subtract loop, so that here too there is a bias toward lowest-numbered vertices. Furthermore, a partition found in the subtract loop must be better than one found in the add loop, not just equal, in order to replace it. Therefore, an additional bias exists in favor of partitions which add vertices, at the expense of partitions of equal strength but which are formed from TSET by subtracting vertices.

3. Control algorithms

The control algorithms are concerned with keeping track of the several results of each partition, selecting previously-found partitions for partitioning in their turn, selecting one of the two major partitioning sequences (and the variants thereof), and deciding when to stop partitioning and when to print out the results.

LCTRL is the major control subprogram for the analysis subprograms. The normal flow is from LCTRL to the partitioning subprogram LGRMN and back again. LCTRL selects the subgraph to be partitioned and stores the results; it also contains the decision rules for terminating partitioning and for transferring control to SCTRL.

After SYMET we keep a permanent record of the symmetricised data matrix in DROWS. This matrix describes the entire graph. As successive partitions of the graph are found, each new subgraph of the basic graph must also be stored in matrix form so that we can operate on it. For convenience of operation, and so that its rows are always numbered consistently, this working matrix keeps the original matrix order, but for those vertices which do not appear in the subgraph, the corresponding rows and columns of the matrix are set to 0. Before any particular subgraph of the original graph can itself be partitioned, it is necessary to produce the appropriate modified matrix to represent the subgraph. In the subprogram LGRMN, this is performed with the generation of MROWS from DROWS. Similarly,

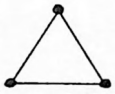
Figure 9

COMPLETE GRAPHS OF ORDERS 1 - 5

N = 1

N = 2

N = 3



N = 4

N = 5

For a complete graph of any order = n, all possible partitions have INFO = 0. The only non-arbitrary decomposition is into n vertices.

in SMRMN, the working area MATA is generated from the semi-permanent storage in DATA.

When LCTRL calls up a candidate for partitioning, it computes the size of that subgraph. If the subgraph is of order greater than 36 the normal sequence through LGRMN is followed. However, if the order of the subgraph is 36 or less, control is shifted to the alternate sequence which utilizes the faster partitioning subprogram SMRMN. (The path of this sequence is through REDUC, necessary to change the format of the data to less-than-36.)

When LCTRL transfers to the REDUC-SCTRL-SMRMN sequence, it keeps track in REDST of the subset which was just small enough to be shifted. Once control is passed to SCTRL, this subset is partitioned and repartitioned until it has been decomposed into its component complete subgraphs. A complete subgraph is one in which every pair of vertices is linked, and there are therefore $n(n-1)/2$ links if there are n vertices. Cf. Figure 9. LCTRL is never again concerned with this subtree of the hierarchy.

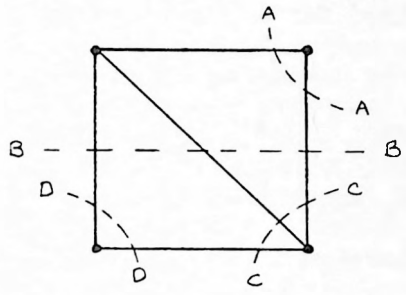
Since each subset passed to SCTRL is thus partitioned to completion, LCTRL can use the record of subsets passed, kept in REDST, as a criterion for halting partitioning: when all the vertices in the original graph (represented by ATOOX) have been included in subsets passed to SCTRL, then the original vertex set has been completely reduced, and partitioning terminates.

Within the less-than-36 sequence, SCTRL is the subprogram that must decide when to stop partitioning the subset it receives as input.

Figure 10

SYMMETRIC GRAPHS

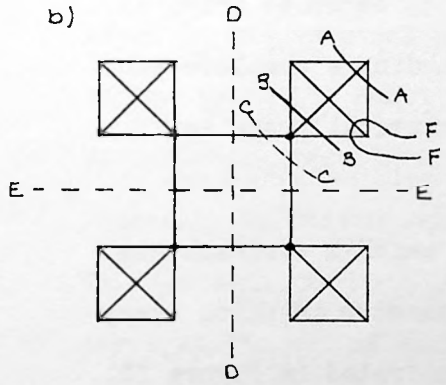
a)



PARTITION	INFO
AA	- 1/36
BB	- 1/72
CC	+ 1/36
DD	- 1/36

This graph is symmetric: partitions AA and DD both have the same (lowest) value of INFO. The intersection procedure (NTRSC) produces this result, a three-way decomposition:

b)



PARTITION	INFO
AA	-.00016
BB	-.0118
CC	-.0195
DD	-.0463
EE	-.0463
FF	-.0016

This graph is symmetric: partitions DD and EE both have the same (lowest) value of INFO. This intersection procedure (NTRSC) produces the following result, a four-way decomposition:

Under certain conditions the subgraph which arrives in SMRMN, from SCTRL, is a complete subgraph. In this case every possible partition of the set of vertices has the same value of INFO, so that it is clearly pointless to try and apply the hill-climb algorithm: the arrows are not defined for any line on the lattice. SMRMN tests for this condition by comparing the number of 1's in the input matrix (TOTAL) with NSQ1 = (n)(n-1)/2. Each time that a complete subgraph is detected, control at once returns to SCTRL, the set of vertices in question is recorded in SIMST, and SCTRL moves on to select another subset for decomposition. As a result, SIMST contains a running record of all sets which can be decomposed no further. When SIMST contains all the vertices of the graph of order 36 or less which was input to SCTRL, then SCTRL stops and returns program control to LCTRL.

This test is made in the less than 36 sequence only, as it is assumed that the probability of finding a complete subgraph of order greater than 36 in any practical graph is negligible.

There is a second kind of symmetry which a subgraph can exhibit, less strong than the perfect symmetry in which every partition has equal value. This is illustrated in Figure 10. Here, though not all partitions are equal, the two best partitions (i.e., lowest INFO value) are equal. To avoid being arbitrary, we must make both these partitions simultaneously,

and thus cut the set into three subsets rather than two. This procedure, brought into action whenever two or more best partitions of equal value occur, is handled by NTRSC* (called by SMRMN). Again, it is assumed that this happens mainly for small graphs, and there is no corresponding procedure for LGRMN.

The development of the hierarchy of subsets begins with the two-way partition of the original vertex set; each of the two subsets (more than two if the NTRSC subprogram is called for) is then partitioned, and these results are in turn partitioned. The control procedure in both LCTRL and SCTRL is to use one index register to keep track of where the results should be stored. Both a subset and its resulting partitions are stored in the same block with this method (MACRO in LCTRL and ATMS in SCTRL), since each result of a partition will in turn become an object to be partitioned.

The hierarchy of subsets has discrete levels; at the first level is the original graph, at the second its two major subsets, at the third the partitions of these subsets, etc.** Since the partitioning results are stored in one area, sequentially, it is necessary to distinquish among these levels of the hierarchy. This is done, in both LCTRL and SCTRL, by storing a word of zeros at the end of each level. When index register 1, which selects subsets to be partitioned from one level, finds one subset which is empty (all zeros), then another empty set

---

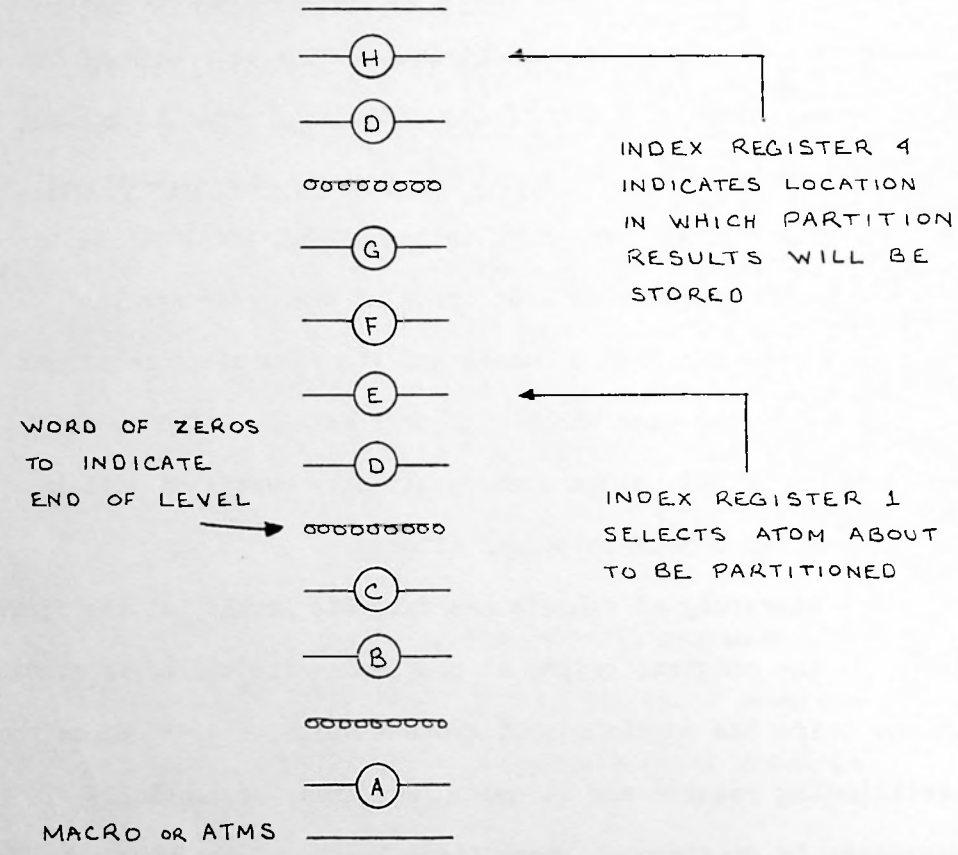
*NTRSC and its associated print-out routine, PTSCT, have not been incorporated in the version of the program operational in December 1961.

**Each new level is indicated by the remark "New level of hierarchy" in the printed output. Cf. p. 37 f.

Figure 11

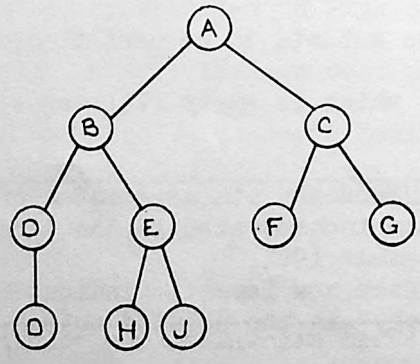
STORAGE CONTROL

STORAGE FORMAT (BACKWARDS FROM MACRO OR ATMS)



CORRESPONDING HIERARCHY

NOTE THAT SUBGRAPH D IS NOT PARTITIONED IN ITS TURN; THEREFORE, IT IS STORED REPEATEDLY, ONCE IN EACH HIERARCHY LEVEL.



(zero word) is stored at the location for partition results specified by index register 4, thus marking the end of the level next lower in the hierarchy. Cf. Figure 11.

Both for the less-than-36 sequence and the larger sequence, the results of the successive partitions are printed out as they are computed.* This is accomplished by calling PTCLR from SCTRL and PTLGR from LCTRL, and PTSCT for those exceptional cases in SCTRL which are referred to the control of NTRSC.** This is discussed further under OUTPUT, Section III, C.

*This eliminates the requirement for erasing the whole of the ATMS block each time that a less-then-36 sequence is completed.

**Cf. note about NTRSC on preceding page.

III. OPERATIONAL DETAILS

A. INPUT

The links of a graph, as used in this program, are non-directional. The matrix representing the links must therefore be symmetrical about its main diagonal, i.e. $m_{ij} = m_{ji}$. Also, since no vertex may be linked to itself, the elements along the main diagonal, m_{ii}, must all be zero.

The matrix is input from punched cards. Since a single error would make the matrix symmetrical, or introduce 1's on the main diagonal, and would thus destroy the conditions necessary for correct operation of the algorithms, the subprogram SYMET is used to check the input and remove errors of the above types. SYMET eradicates 1's on the main diagonal, and replaces both m_{ij} and m_{ji} by $(m_{ij} \cap m_{ji})$ for all i and j, thus leaving a 1 in them only if both are initially 1.

It turns out, incidentally, that SYMET is even more useful than originally intended. For a large design problem of the type which generates our input, it is not only hard to ensure accuracy in punching, but also hard to decide just which point pairs are linked. If the decision for m_{ij} is made independently of the decision for m_{ji}, it is almost impossible, in practice, to make all these decisions consistent with the formal symmetry required. With the program SYMET in operation, however, it is possible to generate the data at the card punch, without hand-checking it, with the assurance that it will be machine checked and that only the "most certain" pairs - those where a link has been defined for both m_{ij} and m_{ji} - will be treated as linked.

Other than the data of the problem - the representation of the graph itself - the program requires as input only three parameters. These are: ORDER, the order or number of rows in the matrix; LATIS, the number of starting sets for the hill-climbing algorithm to be chosen from the lattice; and RANDM, a random word used as a base for calculating the RANDM and DIFF blocks of random words. The larger the value of LATIS selected, the more likely that the sampling procedure will discover the optimal TSET - but as the sample size increases, so does the amount of computer time used.

Input Formats

A. Parameters

1. The parameters follow immediately after the *....DATA card of the Fortran Monitor System.
2. The sequence of the parameters is:

   ORDER

   RANDM

   LATIS
3. Each parameter is entered on a separate card, in octal, in columns 1-12. ORDER and LATIS have their respective numerical values located in the decrement of the corresponding 36-bit word, and are integer-valued. RANDM is any arbitrary octal word.

B. Data

1. The data follow the parameter cards.
2. The matrix is inputted in descending order, by rows; i.e., row one, row two, etc.

3. Each row is broken into units of 72 columns; any fraction of a unit is completed with columns of zeros to form a full 72-column unit.

4. Each 72-column-by-one-row unit is on a separate card, occupying columns 1-72.

5. The binary matrix representing the graph is expressed directly on the card in ones and zeros. (Under the present M.I.T. Fortran monitor system, only the ones need to be punched, since blanks are interpreted as zeros.) Any identification of the card can be punched in columns 73-80.

6. The sequence of the cards for any row is in order of ascending order of the columns: i.e., 1-72, 73-144, 145-216, etc.

7. Example of input sequence:

ORDER = 90

Card 1. Row 1, columns 1-72 of the matrix; punched in card columns 1-72.

Card 2. Row 1, columns 73-90 of the matrix; punched in card columns 1-18 (card columns 19-72 are dummies).

Card 3. Row 2, columns 1-72 of the matrix; punched in card columns 1-72.

Card 4. Row 2, columns 73-90 of the matrix; punched in card columns 1-18.

Card 5. Row 3, columns 1-72 of the matrix; punched in card columns 1-72.

etc.

(Total: 180 cards)

B. SIMPLIFIED OPERATING INSTRUCTIONS

A deck prepared for submission of a problem under the Fortran Monitor System (using the M.I.T. system tape as of December 1, 1961) consists of the following:

(1) identification card

(2) XEQ card

(3) program deck - FAP symbolic deck or relocatable column binary

(4) DATA card

(5) parameters

(6) data deck - representation of the graph in matrix form

(7) Fortran Post-mortem request cards, if desired.

Format of the printed output is under program control.

C. OUTPUT

The output of running the program package on a particular graph is the tree, or hierarchy of subdivisions of the graph. The printed output consists of one sequence ---the subgraphs of order greater than 36--- into which are injected the several sequences for subgraphs of order 36 or less.

When partitioning control passes from LCTRL to SCTRL (i.e., the subgraph to be partitioned is of order 36 or less), the partition of the subgraph proceeds through to completion, before control is passed back to LCTRL and another subgraph (which may be greater or less than 36) is selected. As the partitioning of the less-than-37 subgraph proceeds, each result is written into the output as it is computed, along with the appropriate comments. This results in the complete tree of the partitions of the less-than-37 subgraph being printed as a unit, prefaced by the PRTIN remark, followed by the PRTOUT remark, and with PTLVL remarks at appropriate points.*

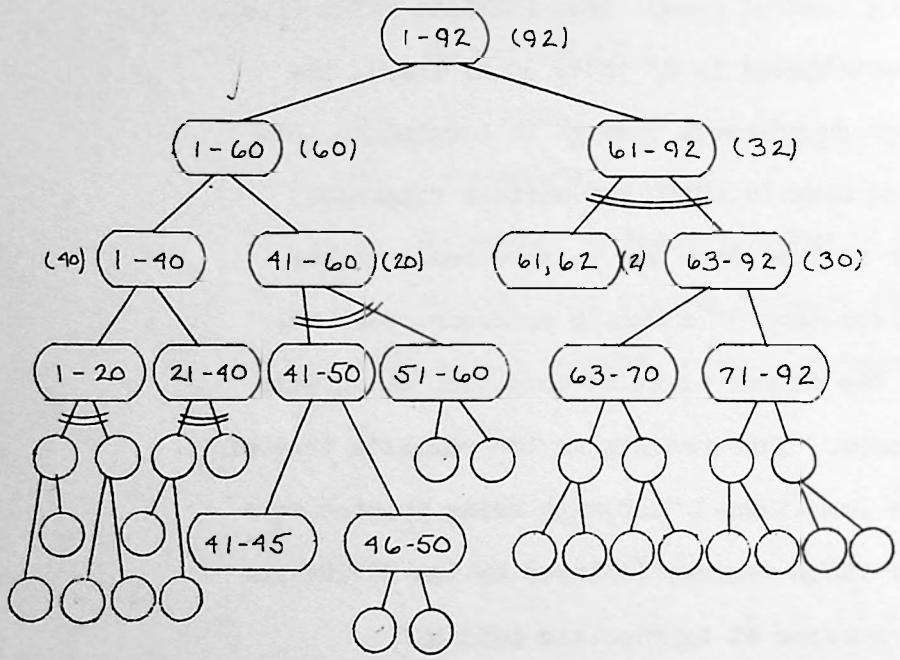
When partitioning control for a particular subgraph remains in LCTRL (i.e., order greater than 36), the partition of the subgraph is only carried through one stage, before the next subgraph is selected by LCTRL for partitioning by LCTRL or SCTRL. Therefore, the major sequence of the output is the partition results produced by LCTRL, printed as they are computed. Whenever the result of a previous partitioning is determined to be of order 36 or less, control of partitioning shifts to SCTRL; at this point, the entire decomposition of this subgraph is computed and printed out, before control returns to LCTRL and

*I.e., "New level of hierarchy." Cf. p. 30.

Figure 12

EXAMPLE OF OUTPUT FORMAT

TREE



KEY: Number in circle identifies the vertices in the subgraph; number in parentheses indicates the number of these vertices included. ⊭ indicates passage to SCTRL from LCTRL.

CORRESPONDING OUTPUT

```
1-92
New level of hierarchy
1-60
61-92
New level of hierarchy
1-40
41-60
Control passed to SCTRL
61-62
63-92
New level of hierarchy
63-70
71-92
New level of hierarchy
------
------
------
------
  .
  .
  .
New level of hierarchy
Control returned to LCTRL
New level of hierarchy
1-20
21-40
Control passed to SCTRL
41-50
51-60
New level of hierarchy
41-45
46-50
------
------
------
  .
  .
  .
New level of hierarchy
Control returned to LCTRL
New level of hierarchy
1-20
21-40
New level of hierarchy
Control passed to SCTRL
  .
  .
  .
New level of hierarchy
Control returned to LCTRL
New level of hierarchy
(END)
```

the printout again follows the major sequence of partitioning results. (Cf. Figure 12, Example of output format, and Appendix D, Typical input and output.)

D. RESTRICTIONS ON MATRIX SIZE

The program as written can operate upon matrices as large as order 252. This limitation is not inherent in the algorithms used, but only depends upon the amount of storage space allocated to the several different tables generated by the program.

Most of the variables for which a block of several words must be reserved do not affect significantly the total amount of storage used. As shown in the program listings, most of these variables, from INDIC to SECTS, inclusive, require a total of only about 350 words. MATAX is also of the order of magnitude of several hundreds of words. It is the other blocks, using several thousands of words each, which are critical.

The length of the ATMS block is determined by the maximum number of subgraphs it is possible to obtain from a graph of order 36 (the maximum size which is handled by SCTRL, the only subprogram operating upon ATMS). It can be shown (see Appendix F) that, for a graph of order n, the maximum number of elements in the tree representing the hierarchical decomposition, the number of ATMS, is $2n - 1$. If we account for spaces between levels in the hierarchy, ATMS, which handles only less-than-37 graphs, cannot require more than $(3)(36) = 108$ words of storage.

This same calculation performed for the size of the MACRO block shows that for a matrix of order 252, MACRO needs, at most, 5300 words of storage.

The length of INMAT is a direct function of the size of the matrix inputted. The length of INMAT is LGTH, a function of ORDER and NWORD. The present size of INMAT is just slightly larger than that necessary for a matrix of order 252. Any increase in ORDER requires an increase in the size of the INMAT block, and the increase is about proportional to the square of the increase in ORDER.

The lengths of DROWS and MROWS are identical, being given by DAT. DAT is a function of NWORD and ORDER, and increases approximately as the square of ORDER. However, DAT is about 40% the magnitude of LOTH for a given ORDER, and so the length of the INMAT block is most critical of all the several large storage areas. The length of MATAX is a direct linear function of ORDER, and so is relatively uncritical for large values of ORDER.

The critical constraints upon the order of matrix, then, are the sizes of the storage areas which must be reserved for DROWS, MROWS, ATMS, and, most critically, MACRO and INMAT. If it is desired to expand the size of graph which can be analyzed, it is necessary to increase the sizes of these blocks appropriately (as well as ATOOX, ATOX, SET, RANDM, DIFF).

There is no real reason for preserving the array stored in INMAT once the Data series of subprograms has been completed, except as a check upon suspicious results. If this check can be dispensed with, or if an appropriate readout subprogram is incorporated, the MACRO block can be written over the INMAT area.

The same block of storage would do double duty, serving as INMAT in the Data stage and as MACRO in the Analysis sequence. This would result in a significant reduction in the amount of storage required, enabling expansion of the program's capacity.

IV. APPENDICES

A. DICTIONARY OF VARIABLES

This alphabetical list of variables includes all those incorporated in the COMMON block for ready access by more than one subprogram, and some others.** All the arguments of the subprograms described in the DICTIONARY OF SUBPROGRAMS are included in this list, as they must be in COMMON. Variables not in COMMON are so described.

The descriptions include where (in which subprogram) the values for the variable are computed or generated, and where the variable is used for additional operations.

For descriptions of some variables not listed, but important in LGRMN or SMRMN, see the descriptions of those subprograms. Those variables marked with an asterisk have reference only to the part of the program which deals with matrices of order less than 37. These names are usually simpler versions of their greater-than-36 counterparts, since this part of the program was developed first.

*ATMS- storage for the results of successive partitions of graphs of size 36 or less. Under the control of SCTRL, the results of successive partitions of a particular subgraph are stored in ATMS. The entire ATMS block is printed out by PTCLR as each element is computed, and before control is returned to LCTRL. The next time that control passes from LCTRL to SCTRL the previous results in ATMS are overwritten. Computed in SCTRL, used in SCTRL.

**Some variables particular to NTRSC are included here, although NTRSC is not incorporated in the program version operational as of December, 1961.

*ATOM- the representation of the vertices of the subgraph (of order 36 or less) which is currently being considered for or undergoing partition. Computed in SCTRL, operated on in SMRMN.

*ATOMO- the vertices of the subgraph with which LCTRL passes control to SCTRL: the first graph of the subtree built up by SCTRL-SMRMN. Computed in REDUC, used in SCTRL, SMRMN.

ATOOX- the graph which is the subject of the analysis. More precisely, the representation of the vertices of the graph. Computed as a function of ORDER in MAIN, used in LCTRL, LGRMN. An arbitrary ATOOX can be inserted in MAIN, if desired.

ATOX- the representation of the subgraph (of order greater than 36) which is currently being considered for or undergoing partition. Computed in LCTRL, used in LCTRL and LGRMN.

COMUN- a table of constants, used in various operations. The complement of UNIT; COMUN-n has a zero bit in the bit position n-th from the left in the 36-bit 709 logical word, and ones elsewhere. Computed in GENER, used throughout.

CONVT- the block in which is stored the key by which the subgraphs of order 36 or less can be identified in the subgraph of greater-than-36 size, while being partitioned under the control of SCTRL. Computed in REDUC, used in REDUC, PTCLR.

DAT- the number of significant words stored in the DROWS or MROWS blocks. DAT = ORDER (NWORD + 1). Computed in GENER, used in CNDAT, SYMET.

*DATA- the data matrix representing the links of the subgraph of order 36 (or less) being partitioned under the control of SCTRL. When control passes to SCTRL, DATA is computed from DROWS in REDUC, and all the successive partitions of the subgraph (under the control of SCTRL) use DATA to compute the appropriate MATA. DATA is similar to DROWS in that, once completed, it becomes permanent storage, and necessary modifications are computed and stored in the working area MATA (counterpart of MROWS). Computed in REDUC, used in SMRMN.

DIFF- the word or block of words used together with RANDM to select a sample of starting points of size LATIS: new RANDM's are generated by the addition of DIFF to the preceding RANDM. Calculated in GENER, used in SMRMN, LGRMN.

DROWS- the data matrix representing the links of the original graph. Once SYMET and the previous sequence of Preliminary and Data subprograms have been completed, DROWS becomes unchangeable, the permanent storage for the original symmetricized matrix. All modifications of this matrix necessary for operation on particular subgraphs are generated when needed, and stored in the local working areas NROWS and DATA - MATA. Computed in final form in SYMET, used throughout.

D36- a standard quantity: one word which contains the integer 36 in its decrement. Computed in GENER, used throughout.

*EQLS- the set of partitions of equal strength at the subdivision of a particular subgraph. Cf. discussion of NTRSC, SECTS. Computed in SMRMN, used in NTRSC.

INFO- the criterion used to determine the better of two partitions. INFO is signed, and has a range over the real numbers. Of two partitions, the one with an algebraically lower value of INFO is the more desirable. Computed and used in SMRMN and in LGRMN. Not in COMMON.

INMAT- temporary storage for data matrix of the original graph, as inputted by INDAT. Format of data changed by CNDAT and new version stored in MROWS. Computed in INDAT, used in CNDAT.

LATIS- the number of times that the lattice of possible partitions is sampled at any point in the tree by generation of a new path beginning; the sample size. Input as parameter, used in LGRMN, SMRMN.

*LATS1- the counter used to control the size of sample for the starting-point algorithm and the associated hill-climbing algorithms. Reset value is LATIS. Computed and used in SMRMN. Not in COMMON.

LGTH- the number of significant words stored in the INMAT block. LGTH = (ORDER) x (NWORD) x (3). Computed in GENER, used in INDAT, CNDAT.

MACRO- the larger-than-36 counterpart of ATMS. Storage for the tree of successive partitions of the graph, computed in LGRMN and LCTRL. Used in LCTRL.

*MATA- the data matrix computed from DATA which represents the links of the subgraph actually being partitioned. Cf. DATA. Computed and used in SMRMN.

MATAX- a key for indirect addressing of MROWS. Cf. Figure 28, Appendix. Computed in GENER, used in all subprograms dealing with graphs of order greater than 36.

MN- for any partition, and the two corresponding subgraphs, the product of the number of vertices in one subgraph and the number in the second. MN represents the maximum number of links possible between the two subgraphs. Computed and used in SMRMN; computed and used similarly in LGRMN. Not in COMMON.

MROWS- the working area for storage of the matrix representing any particular graph of order greater than 36. Derived from DROWS (q.v.). MATAX is indexing key for addresses in MROWS. Also used in Data subprograms as temporary storage. Computed in LGRMN, used in LGRMN, REDUC.

NBIT- the number of vertices in ATOX (under control of LCTRL), or in ATOM (when under control of SCTRL). Computed in SCTRL or in LCTRL.

NDXX- the counter used to control the size of sample for the starting point algorithm and the associated hill-climbing algorithms. Reset value is LATIS. Analogous to LATS1. Computed and used in LGRMN. Not in COMMON.

NN- for any partition and the two corresponding subgraphs, NN is the number of vertices in that subgraph of the two which is described by TSET. Computed and used in SMRMN; computed and used similarly in LGRMN. Not in COMMON.

NSET- for any subgraph represented by ATOM or ATOX, SET is the best partition. More precisely, SET enumerates those points which are all in one of the two subgraphs into which ATOM or ATOX is divided. NSET is the other of the two subgraphs, and is the complement of SET within ATOM (or ATOX). NSET is not actually labelled and used as such in any subprograms, but is defined for clarity in this writeup.

NSQ1- for a subgraph of size N, NSQ1 is the maximum number of links possible - i.e., assuming that each point in the graph is connected to every other point; the number of links in a complete graph. Computed in SMRMN to test whether any subgraph is a complete graph. If so, partitioning is not attempted. NSQ1 = $\frac{N(N-1)}{2}$. Computed in SMRMN, used in SMRMN; computed in LGRMN, used in LGRMN. Not in COMMON.

NWORD- the number of 36-bit words required to represent the graph. Precisely, NWORD is the number of units of 36 in ORDER, with any fraction of a unit considered as a full unit, and any odd number of units rounded to the next highest even number. For example, ORDER = 35, 36, 37 or 72, NWORD = 2; ORDER = 73, 108, or 144, NWORD = 4; etc. Computed in GENER, used in all subprograms operating upon graphs larger than 36.

ONED- a standard quantity; the integer one in the decrement of a word. Computed in GENER, used throughout.

ORDER- the size of the matrix: the highest number assigned to a vertex of the graph. ORDER may be much greater than the number of vertices in the graph - e.g., a three-vertex graph may consist of vertices labelled, 9, 35, and 56, in which case ORDER = 56, while NBIT = 3. Input as parameter, used in almost all subprograms.

RANDM- the word or block of words used to select a pseudo-random sample of starting points for the hill-climbing algorithms of SMRMN and LGRMN. RANDM itself is input as an arbitrary parameter, and used in SMRMN. In GENER, RANDM is used to compute DIFF, and the two blocks behind RANDM and DIFF. The block forms are used in LGRMN, the single words in SMRMN.

REDST- the cumulative record of subsets of vertices which are passed from LCTRL to SCTRL, for partitioning to completion by the less-than-36 sequence. REDST is used by LCTRL as a criterion for halting partitioning: when REDST is identical to ATOOX, partitioning is terminated and control returns to (MAIN) from LCTRL. Computed and used in LCTRL. Not in COMMON.

RR- the number of links connecting the two subgraphs represented by the partition of a graph. Computed and used in SMRMN; computed and used similarly in LGRMN. Not in COMMON.

SET- the representation of one subgraph derived from a partition of the current ATOM or ATOX. If ATOM, then SET proper is the subgraph; if ATOX (order greater than 36), then the subgraph is stored in the block labelled by SET. Computed in LGRMN or SMRMN, used in LCTRL or SCTRL, and PTLGR or PTCLR.

SIMST- used in SCTRL to determine when the particular subgraph being partitioned has been reduced completely to component complete graphs, by successive partitioning. Cf. SCTRL description. Computed and used in SCTRL; not in COMMON.

TSET- a graph and its partition into two subgraphs can be described by three quantities, one of which is redundant: the set of vertices in the graph, the set of vertices in one subgraph, the set of vertices in the other subgraph. In this program, the graph is described by ATOM or ATOX (for less than 36 or greater than, respectively), and TSET specifies one of the two subgraphs. The second subgraph can always be computed from ATOM or ATOX, and TSET. -TSET is generated by the sampling and hill-climbing algorithms. Computed and used in SMRMN; computed and used similarly in LGRMN. Not in COMMON.

UNIT- a table of constants, used in various operations. UNIT-n has a one in the bit position n-th from the left in the 36-bit 709 logical word, and zeros in the remaining 35 positions. Computed in GENER, used throughout.

B. DICTIONARY OF SUBPROGRAMS

1. INPAR
2. GENER
3a. SET8
3b. SET9
4. INDAT
5. CNDAT
6. SYMET
7. PTMAT
8. LCTRL
9. LGRMN
10. REDUC
11a. SCTRL
11b. *ALPHA
11c. *NSTOR
11d. SIMPL
12. SMRMN
13. *NTRSC
14a. COUNT
14b. CNVRT
15a. PRTIN
15b. PTLVL
15c. PTOUT
16. PTLGR
17. PTCLR
18. PTSCT

*These subprograms, concerned with NTRSC and related functions, have not been incorporated in the version of HIDECS 2 operational in December, 1961.

Explanation of format of subprogram writeups

Arguments: key variables whose values are carried from one subprogram into another. All arguments are in COMMON.

Called by: the subprogram from which program control is transferred.

Calls: subroutine into which program control is to be transferred.

Follows: if this subroutine is being called as one of a sequence, this indicates the preceding subprogram in that sequence.

Followed by: next subroutine which will be called by same calling sequence.

1. Subprogram: INPAR
Called by (MAIN)
Calls: ---
Follows: --- (first program called)
Followed by: GENER
Entered with arguments: (none)
Determines as arguments: LATIS
ORDER
RANDM

Description:
INPAR reads in LATIS, the size of sample to be taken; ORDER, the size of the graph with which the analysis commences; RANDM, a random number used to select trial partitions of the graph; Cf. remarks in "Input" section, and Figure 3.

2. Subprogram: GENER
Called by: (MAIN)
Calls: ---
Follows: INPAR
Followed by: SET 8
Entered with arguments: ORDER, RANDM
Determines as arguments: NWORD
DAT
LGTH
ONED
D36
UNIT table
COMUN table
RANDM table
DIFF table
MATAX table

Description:
GENER generates a number of relevant parameters. NWORD, DAT, and LGTH are parameters which describe various dimensions of the data arrays, and are computed as functions of ORDER.
ONED and D36 are standard constants.
The UNIT and COMUN tables are also constant from run to run. Cf. the variable dictionary.
GENER generates the MATAX block, which is the indirect-addressing key to MROWS.

3a. Subprogram: SET 8
Called by: (MAIN)
Calls: ---
Follows: GENER
Followed by: SET9
Entered with arguments: NWORD
ORDER
Determines as arguments: see description

Description: Sets the controls for certain loops in LCTRL as a function of the variable parameters NWORD and ORDER.

3b. Subprogram: SET9
Called by: (MAIN)
Calls: ---
Follows: SET8
Followed by: SET11
Entered with arguments: NWORD
ORDER
Determines as arguments: see description

Description: Sets the controls for certain loops in LGRMN as a function of the variable parameters NWORD and ORDER.

3c. Subprogram: SET11
Called by: (MAIN)
Calls: ---
Follows: SET9
Followed by: INDAT
Entered with arguments: NWORD
ORDER
Determines as arguments: see description

Description: Sets the controls for certain loops in REDUC as a function of the variable parameters NWORD and ORDER.

4. Subprogram: INDAT
Called by: (MAIN)
Calls: ---
Follows: SET11
Followed by: CNDAT
Entered with arguments: ORDER
LGTH
DAT
NWORD
Determines as arguments: INMAT block of data

Description: The Fortran Monitor System (FMS) cannot read in binary information directly. Therefore, INDAT reads in the sequence of data cards with the matrix representation as if they were in octal form, and stores this information in sequence in the storage block INMAT. Cf. section on "Input."

The matrix is represented in binary fashion, by arranging one row to each consecutive group of cards. When read in, however, each binary 1 on the cards is interpreted by FMS as an octal 1 and is allocated to one octal digit, or three binary digits, in storage. Therefore, each card of 72 columns, representing a matrix row with 72 columns, is interpreted and stored as six 36-bit words. Each of these words is the binary representation of the octal number represented by a unit of twelve binary bits in the input card. CNDAT operates to contract these words into their proper form.

5. Subprogram: CNDAT
Called by: (MAIN)
Calls: CNVRT
Follows: INDAT
Followed by: SYMET
Entered with arguments: INMAT block of data
NWORD
DAT
LGTH
Determines as arguments: DROWS block of data
MROWS

Description: Operates upon the expanded form of data matrix, as stored behind INMAT, and condenses it to correct format, loading into MROWS and DROWS blocks. Utilizes table look-up operation in CNVRT.

6. Subprogram: SYMET
Called by: (MAIN)
Calls: ---
Follows: CNDAT
Followed by: PTMAT
Entered with arguments: ORDER
NWORD
DROWS block of data
Determines as arguments: DROWS block (modified)

Description: Checks data matrix, as stored behind DROWS, for symmetry about the diagonal, and replaces all non-symmetric pairs of elements by zeros. That is, SYMET takes the intersection of the two halves of the matrix about the diagonal. Also, replaces all non-zero elements on the diagonal by zeros.

7. Subprogram: PTMAT
Called by: (MAIN)
Calls: ---
Follows: SYMET
Followed by: LCTRL
Entered with arguments: ORDER
NWORD
DAT
DROWS block of data
Determines as arguments: see description

Description: This subprogram prints out the links of the matrix, after SYMET has symmetricised the input data.

8. Subprogram: LCTRL
Called by: (MAIN)
Calls: LGRMN
PTLVL
PRTIN
REDUC
SCTRL
Follows: SYMET
Followed by: PTLGR
Entered with arguments: ATOOX block
DROWS block of data
ORDER
NWORD
LATIS

Determines as arguments: MACRO block of partitions
NBIT
ATOX block (to be partitioned)

Description: This is the control subprogram that analyzes each subgraph and decides whether it should be partitioned further by LGRMN or if it should be passed to REDUC for partition by SCTRL and SMRMN. LCTRL also stores the results of LGRMN and controls printout by PTLGR. Cf. the flow diagram.

9. Subprogram: LGRMN
Called by: LCTRL
Calls: ---
Follows: ---
Followed by: ---

Entered with arguments:	DROWS block of data
	ATOX block
	NWORD
	LATIS
	NBIT
	RANDM block
	DIFF block
Determines as arguments:	SET block (partition results)

Description: This subprogram performs the actual analysis of the graph with which it is entered (ATOX block), and consists of the sampling and criterion algorithms. The output is the best partition discovered, the SET block.

10. Subprogram: REDUC
Called by: LCTRL
Calls: ---
Follows: PRTIN
Followed by: SCTRL

Entered with arguments:	ATOX block
	DROWS block of data
	NWORD
	NBIT
Determines as arguments:	ATOMO
	DATA block
	CONVT block

Description: REDUC is called by LCTRL when the subgraph to be partitioned is of order 36 or less. REDUC prepares the subgraph for analysis by SMRMN under the control of SCTRL by:

1. Condensing the description of the subgraph as recorded by ATOX block into a single word, ATOMO.

2. Condensing the matrix of size ORDER as stored in the DROWS block, into a matrix of order 36 or less, stored in the DATA block.

3. Generating a conversion key, CONVT, for identifying the elements of the condensed format.

The printout routine, PTCLR, utilizes this key.

11a. Subprogram: SCTRL
Called by: LCTRL
Calls: SMRMN
PTLVL
PTCLR
PTSCT
PTOUT
Follows: REDUC
Followed by: ---
Entered with arguments: ATOMO
DATA block
Determines as arguments: ATMS block of partition results
ATOM (to be partitioned)

Description: This is the control subprogram that, analogous to LCTRL, analyzes each subgraph and decides whether it should be partitioned further, by SMRMN, and controls the storage of partition results. Each time control returns to SCTRL from SMRMN, the partition results are printed out by PTCLR.

SCTRL is entered first from REDUC with ATOMO; ATOMO and its successive partitions are stored in ATMS, and partitioning continues under the control of SCTRL until ATOMO has been reduced entirely to its component complete graphs.

Cf. flow diagram, Figure 16.

11b. Subprogram: ALPHA (SCTRL)
Called by: SMRMN
Calls: PTCLR
Follows: *
Followed by: *
Entered with arguments: SET
ATOM
Determines as arguments: see description

*Description: ALPHA is an entry point in SCTRL, to which SMRMN returns control under normal conditions (i.e., ATOM is not a simplex, and the best partition of ATOM is the only one found with the lowest value of RMN). ALPHA stores SET and its complement in ATOM in the ATMS block, and calls PTCLR to print out SET and its complement.

11c. Subprogram: NSTOR (SCTRL)
Called by: NTRSC
Calls: PTSCT
Follows: *
Followed by: *
Entered with arguments: SECTS
Determines as arguments: see description

*Description: NSTOR is an entry point within SCTRL. When more than one "best" partition of ATOM has been found, SMRMN calls NTRSC and then returns control to SCTRL via NSTOR. NSTOR calls PTSCT to print out the partition results of this special case, as entered in SECTS, and stores these results in ATMS block.

11d. Subprogram: SIMPL (SCTRL)
Called by: SMRMN
Calls: *
Follows: *
Followed by: *
Entered with arguments: ATOM
Determines as arguments: see description

*Description: SIMPL is an entry point in SCTRL. Control is returned to SCTRL via the SIMPL entry point by SMRMN when the NSQ1 test indicates that ATOM is a simplex, and therefore partitioning is unnecessary.
ATOM itself is considered to be the result of the partition and is stored in ATMS block as such. Since this entry comes after "Call PTCLR," there is no printout in this case.

12. Subprogram: SMRMN
Called by: SCTRL
Calls: NTRSC, SIMPL
Follows: ---
Followed by: PTCLR
Entered with arguments: ATOM
DATA block
LATIS
RANDM
DIFF
Determines as arguments: SET
NSET

Description: This subprogram performs the actual analysis of the graph with which it is entered (ATOM), of size 36 or less. Analogous to LGRMN for larger graphs, SMRMN consists of sampling and criterion algorithms. The output is the best partition discovered, SET, and its complement in ATOM, NSET.
In addition to these similarities with LGRMN, SCTRL calls two special subroutines not used in the analysis of larger graphs. Cf. NTRSC and SIMPL.

13. Subprogram: NTRSC
Called by: SMRMN
Calls: NSTOR
Follows: ---
Followed by: ---
Entered with arguments: ATOM
EQLS block
Determines as arguments: SECTS

Description: NTRSC is called if and only if the sampling procedure of SMRMN has found two or more non-identical partitions of ATOM which have equal strength. NTRSC takes these sets of equal partition strengths as stored in EQLS, and computes the subsets of vertices which correspond to the intersections of the partition sets. These are stored in SECTS.

14a. Subprogram: COUNT
Called by: SCTRL
LCTRL
SMRMN
LGRMN
GENER
Calls: ---
Follows: ---
Followed by: ---
Entered with arguments: *
Determines as arguments: see description

*Description: COUNT is a service routine, and is used to count the number of bits in one or more words by a CAQ table-look-up operation. The most important use is to examine ATOM or ATOX block and determine NBIT. COUNT also is used to calculate TOTAL, PA, QA, PS, QS, SR, NN, and RR.

14b. Subprogram: CNVRT
Called by: CNDAT
Calls: ---
Follows: ---
Followed by: ---
Enters with arguments: *
Determines as arguments: see description

*Description: CNVRT is a service routine, using a CAQ table-look-up operation to operate on the expanded matrix stored in the INMAT block and condense it, preparatory to storage in DROWS. Used in CNDAT.

15a. Subprogram: PRTIN
Called by: LCTRL
Calls: ---
Follows: ---
Followed by: REDUC
Entered with arguments: *
Determine as arguments: See description.

*Description: Whenever a subgraph is to be partitioned of size 36 or less and therefore control passes from LCTRL to SCTRL, PRTIN places an appropriate comment in the output: "Control passed to SCTRL."

15b. Subprogram: PTLVL
Called by: SCTRL, LCTRL
Calls: ---
Follows: ---
Followed by: ---
Entered with arguments: *
Determines as arguments: See description.

*Description: Whenever SCTRL or LCTRL, in the process of developing the partition tree for some subgraph, determines that the end of a row in the tree has been reached, PTLVL is called to print an appropriate comment in the output: "New level of hierarchy."

15c. Subprogram: PTOUT
Called by: LCTRL
Calls: ---
Follows: SCTRL
Followed by: ---
Entered with arguments: *
Determines as arguments: See description.

*Description: When SCTRL determines that partitioning of a particular subgraph (of size 36 or less) is completed, and control is returned to LCTRL, then PTOUT is called to place an appropriate comment in the output.

16. Subprogram: PTLGR
Called by: LCTRL
Calls: ---
Follows: ---
Followed by: ---
Entered with arguments: MACRO block of partition results (the tree)
Determines as arguments: See description.

Description: Each time that LGRMN produces SET as a result of partitioning ATOX, PTLGR prints out SET and its complement in ATOX, NSET.

17. Subprogram: PTCLR
Called by: ALPHA (SCTRL)
Calls: ---
Follows: SMRMN
Followed by: ---
Entered with arguments: SET
NSET
Determines as arguments: See description.

Description: Each time that SMRMN produces SET as a result of partitioning ATOM, this subprogram prints out SET and its complement in ATOM, NSET.

18. Subprogram: PTSCT
Called by: SCTRL (NSTOR)
Calls: ---
Follows: ---
Followed by: ---
Entered with arguments: SECTS block
Determines as arguments: See description.

Description: Whenever NTRSC has been used, in stead of the usual SMRMN procedure, PTSCT is called to print out the particular results of NTRSC.

C. FLOW DIAGRAMS FOR SELECTED SUBPROGRAMS

Figure 13

GENER

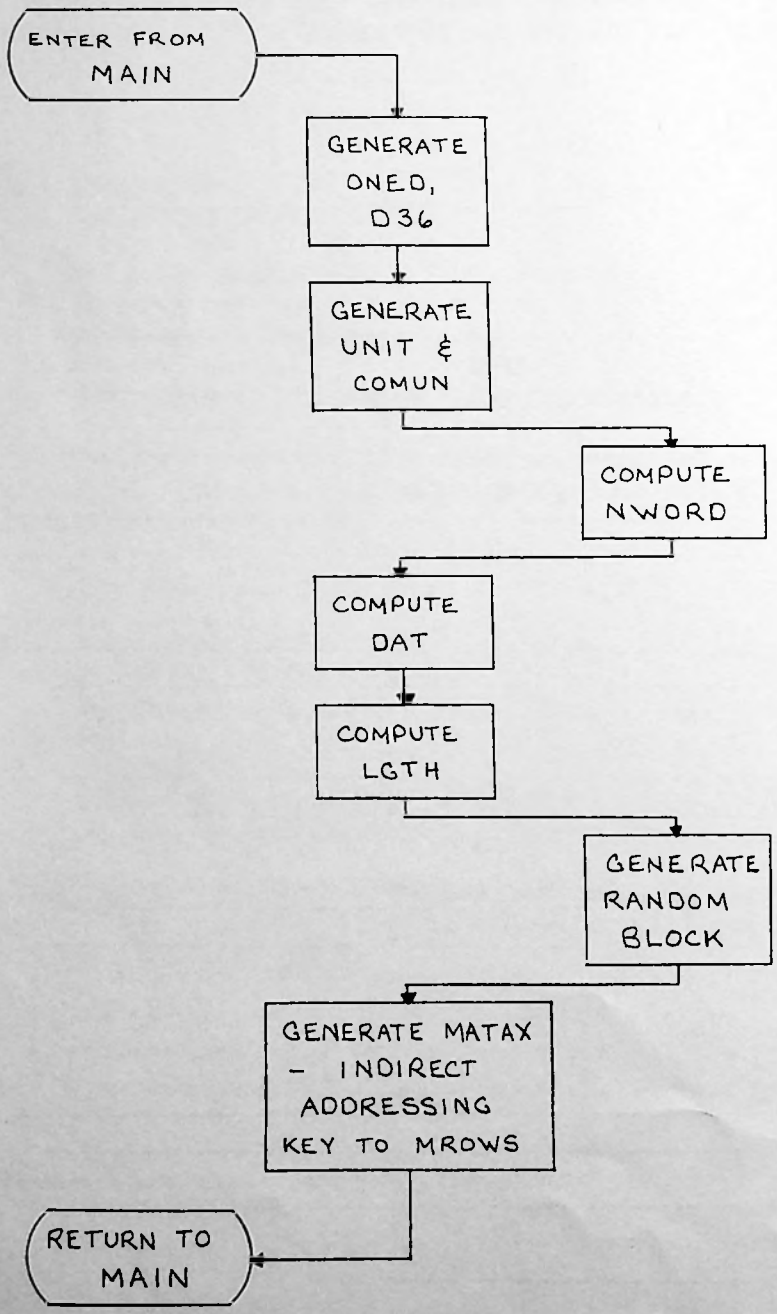


Figure 14

SYMET

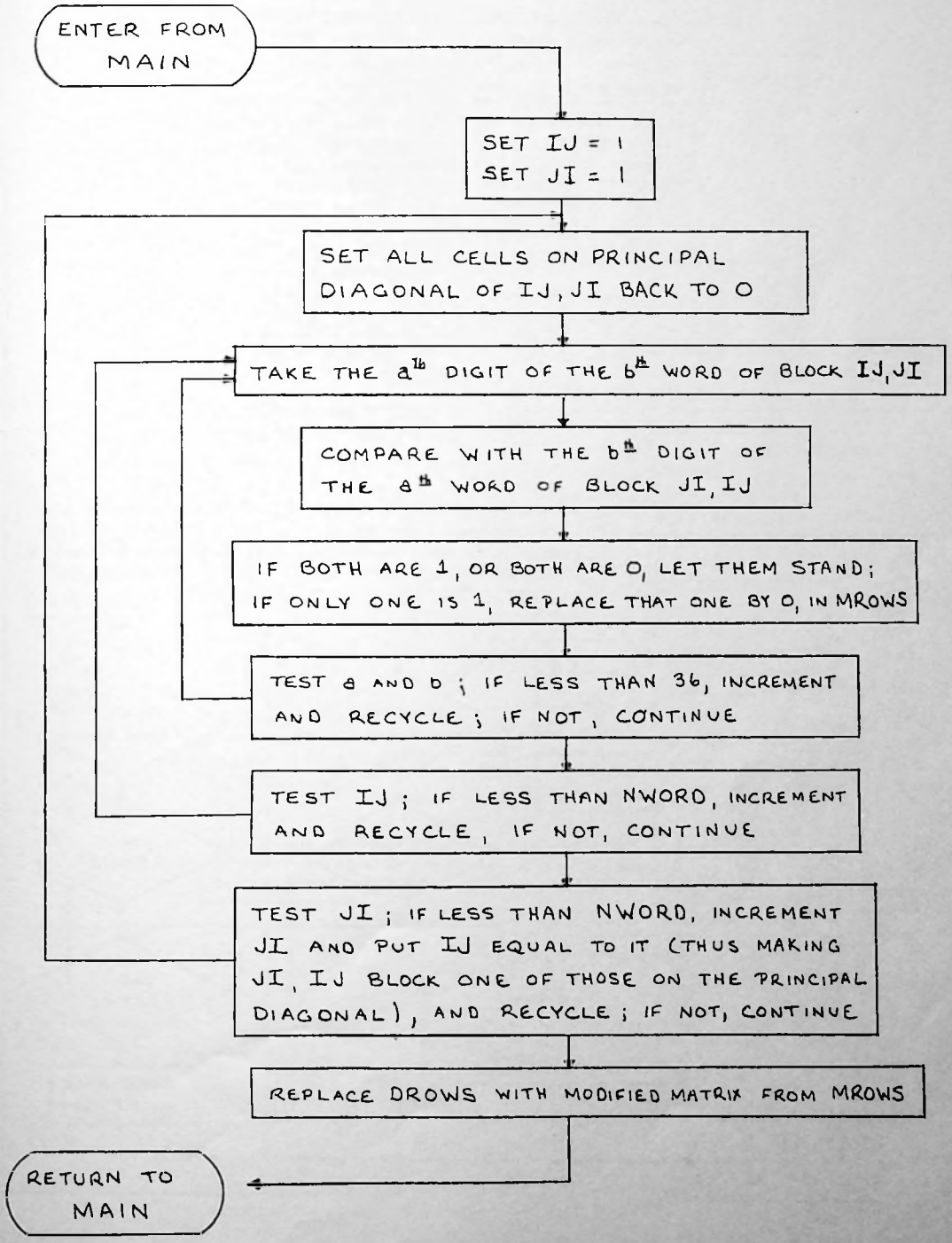


Figure 15

LCTRL

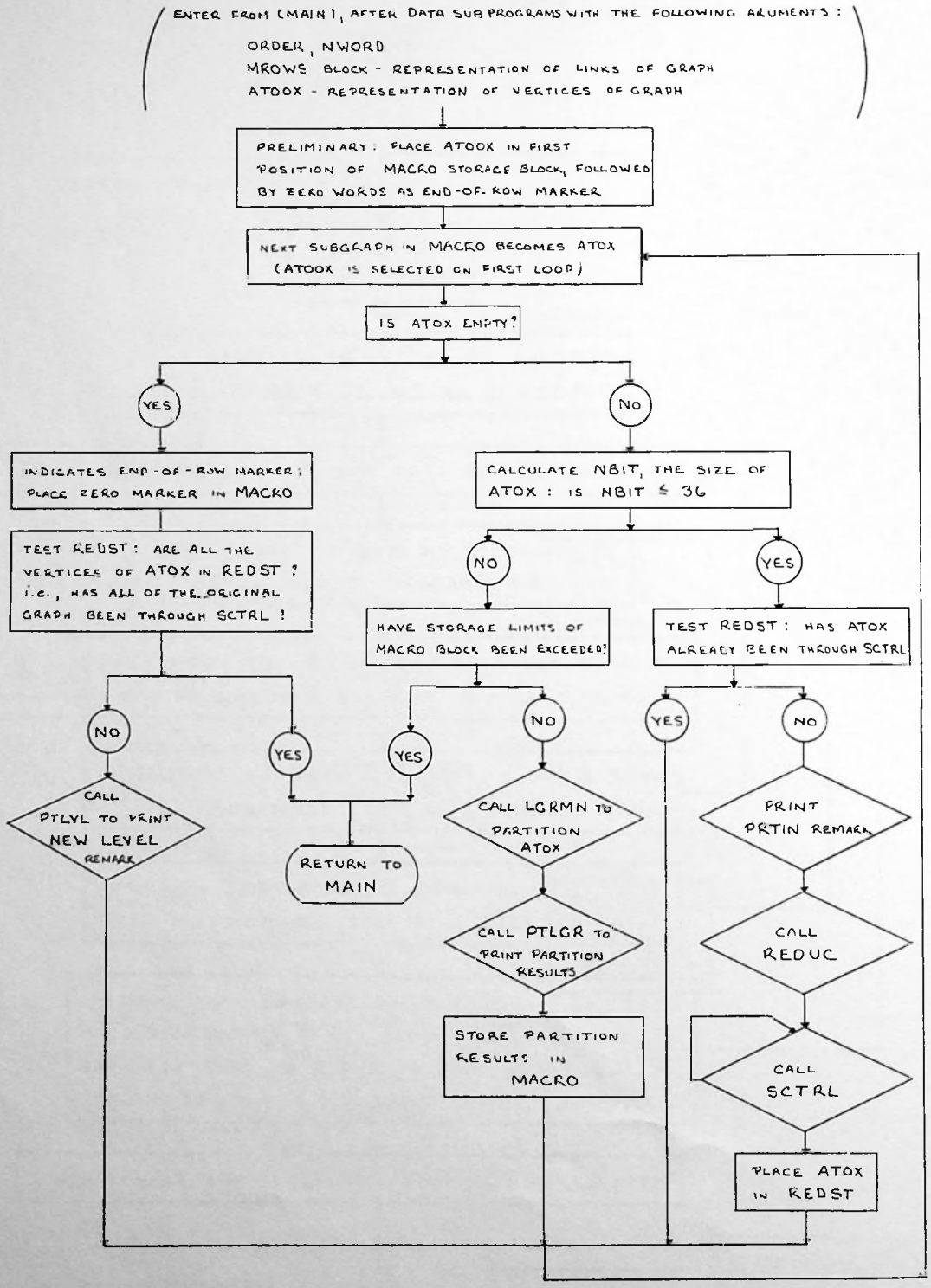


Figure 16

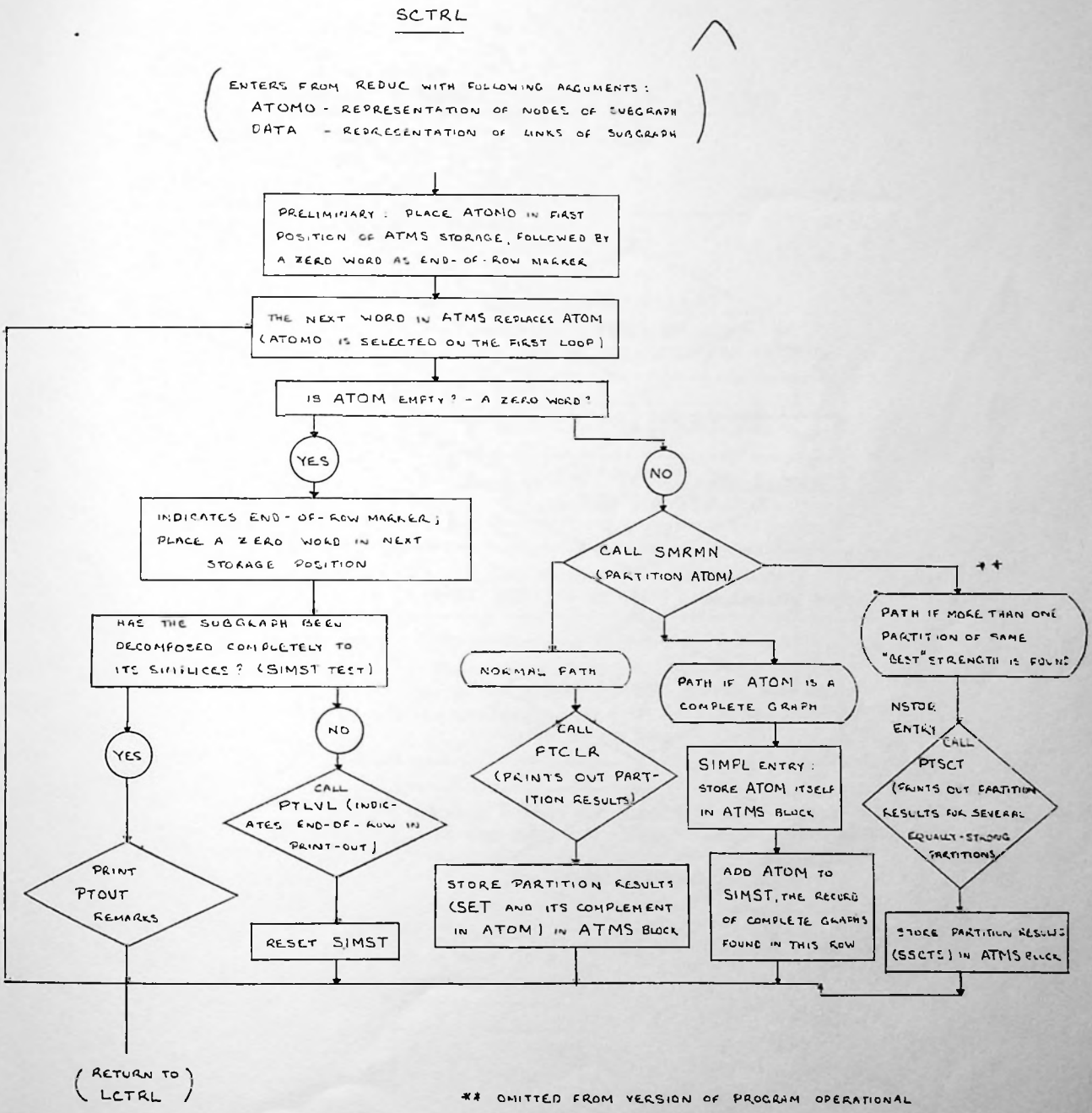
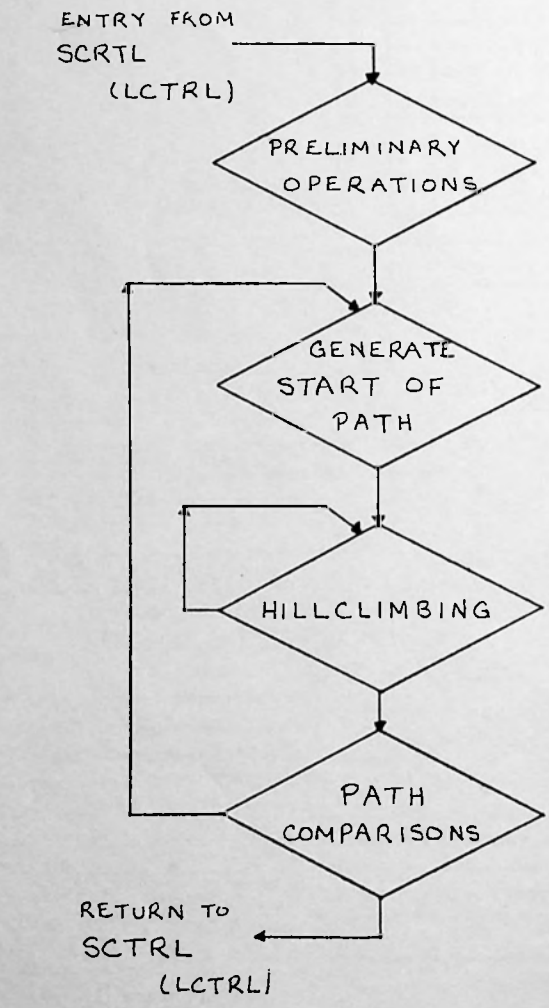
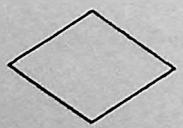

SCTRL
ENTERS FROM REDUC WITH FOLLOWING ARGUMENTS:
ATOMO - REPRESENTATION OF NODES OF SUBGRAPH
DATA - REPRESENTATION OF LINKS OF SUBGRAPH
PRELIMINARY: PLACE ATOMO IN FIRST POSITION OF ATMS STORAGE, FOLLOWED BY A ZERO WORD AS END-OF-ROW MARKER
THE NEXT WORD IN ATMS REPLACES ATOM (ATOMO IS SELECTED ON THE FIRST LOOP)
IS ATOM EMPTY? - A ZERO WORD?
YES
NO
INDICATES END-OF-ROW MARKER; PLACE A ZERO WORD IN NEXT STORAGE POSITION
CALL SMRMN (PARTITION ATOM)
++
PATH IF MORE THAN ONE PARTITION OF SAME "BEST" STRENGTH IS FOUND
HAS THE SUBGRAPH BEEN DECOMPOSED COMPLETELY TO ITS SIMPLICES? (SIMST TEST)
NORMAL PATH
PATH IF ATOM IS A COMPLETE GRAPH
NSTOR ENTRY
YES
NO
CALL PTLVL (INDICATES END-OF-ROW IN PRINT-OUT)
CALL PTCLR (PRINTS OUT PARTITION RESULTS)
SIMPL ENTRY: STORE ATOM ITSELF IN ATMS BLOCK
CALL PTSCT (PRINTS OUT PARTITION RESULTS FOR SEVERAL EQUALLY-STRONG PARTITIONS)
PRINT PTOUT REMARKS
RESET SIMST
STORE PARTITION RESULTS (SET AND ITS COMPLEMENT IN ATOM) IN ATMS BLOCK
ADD ATOM TO SIMST, THE RECORD OF COMPLETE GRAPHS FOUND IN THIS ROW
STORE PARTITION RESULTS (SSCTS) IN ATMS BLOCK
RETURN TO LCTRL
** OMITTED FROM VERSION OF PROGRAM OPERATIONAL AS OF DECEMBER 1, 1961

Figure 17

SMRMN* (LGRMN) : MAJOR UNITS



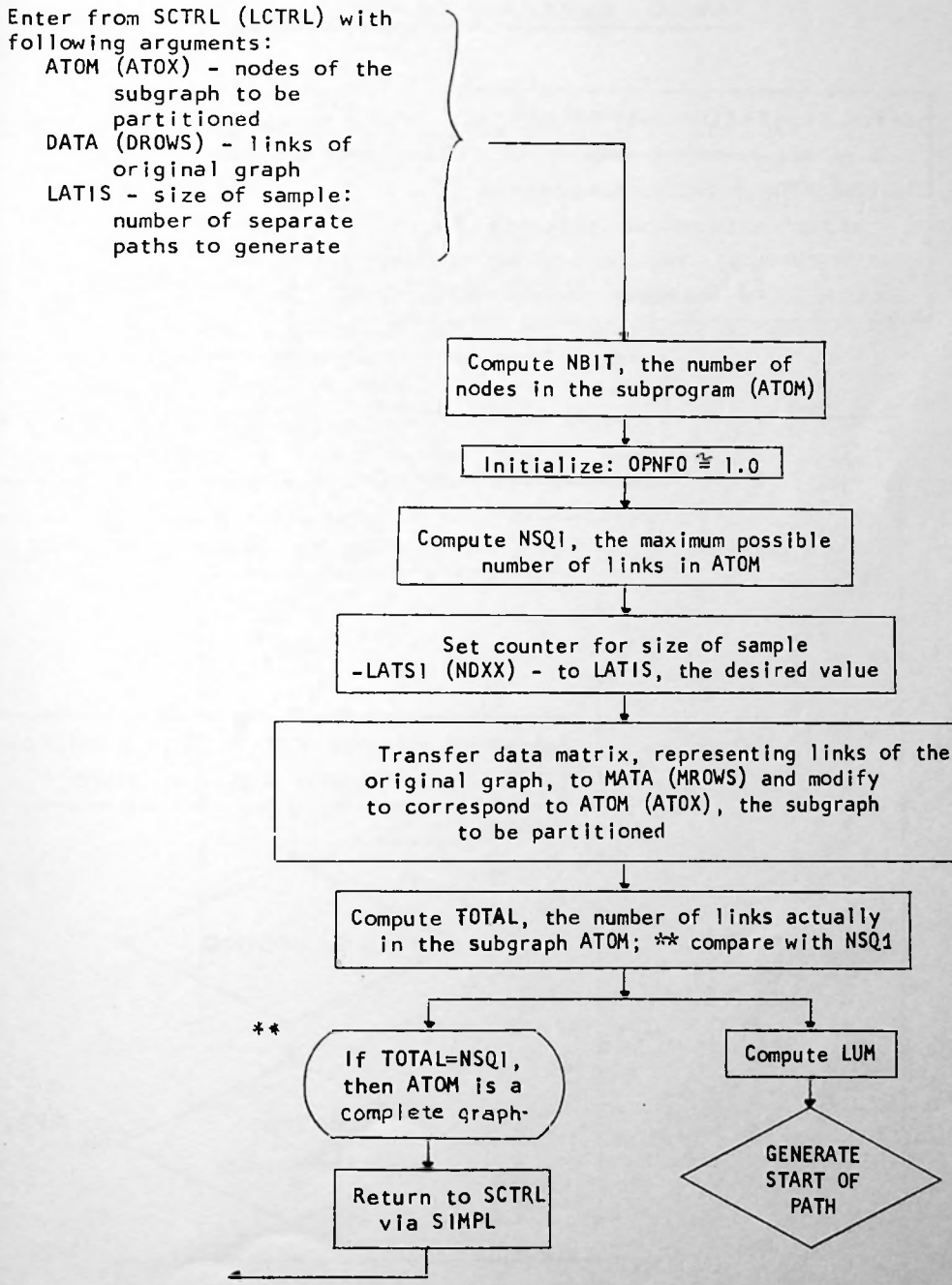


= DETAILS IN FOLLOWING FIGURE

* SMRMN AND LGRMN ARE SIMILAR IN STRUCTURE. ONLY SMRMN IS DIAGRAMMED IN THIS AND THE FOLLOWING FIGURES.

Figure 18

SMRMN*: PRELIMINARY OPERATIONS



* SMRMN and LGRMN are similar in structural details
** In SMRMN only - not included in LGRMN

Figure 19

SMRMN: GENERATE START OF PATH

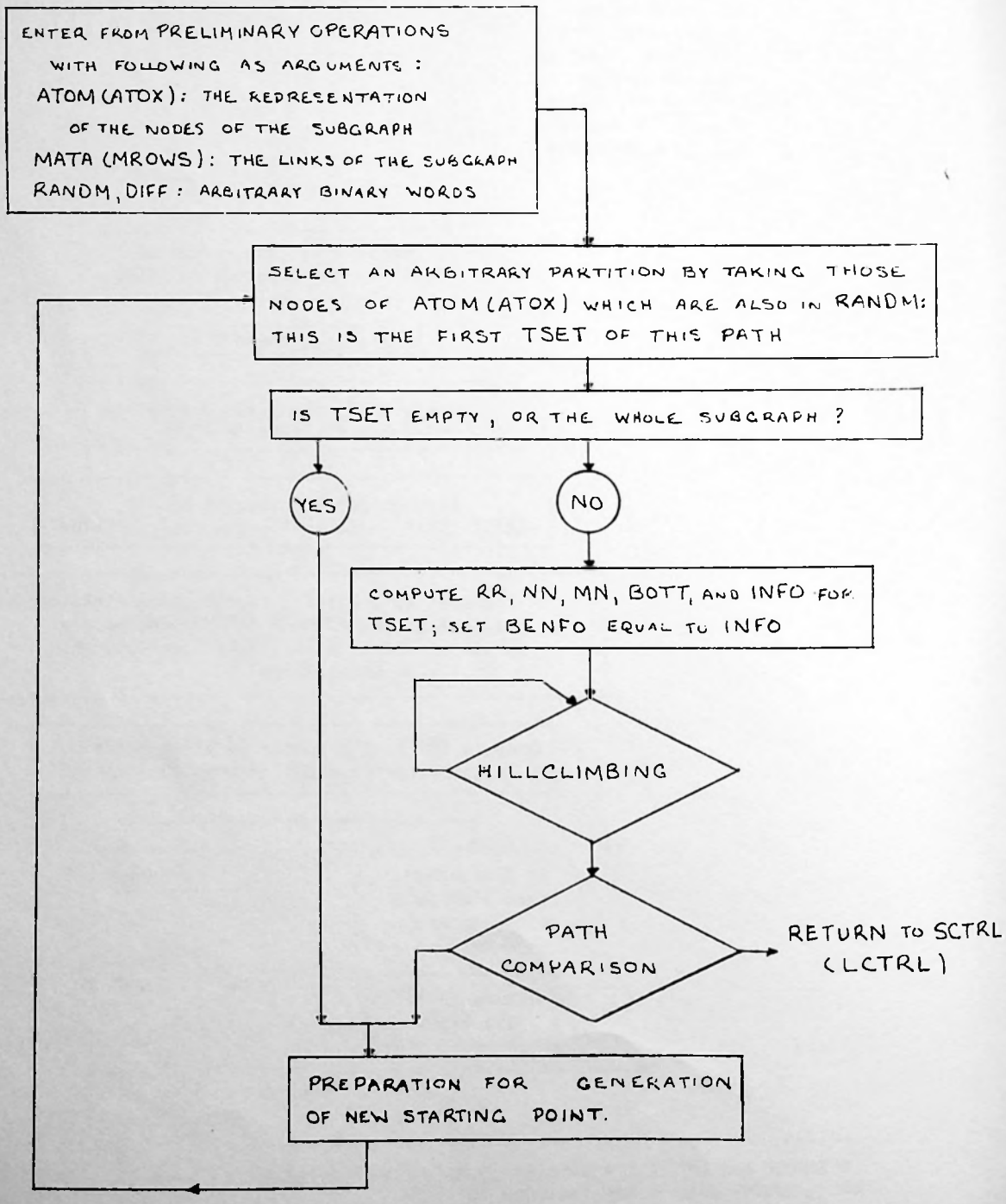
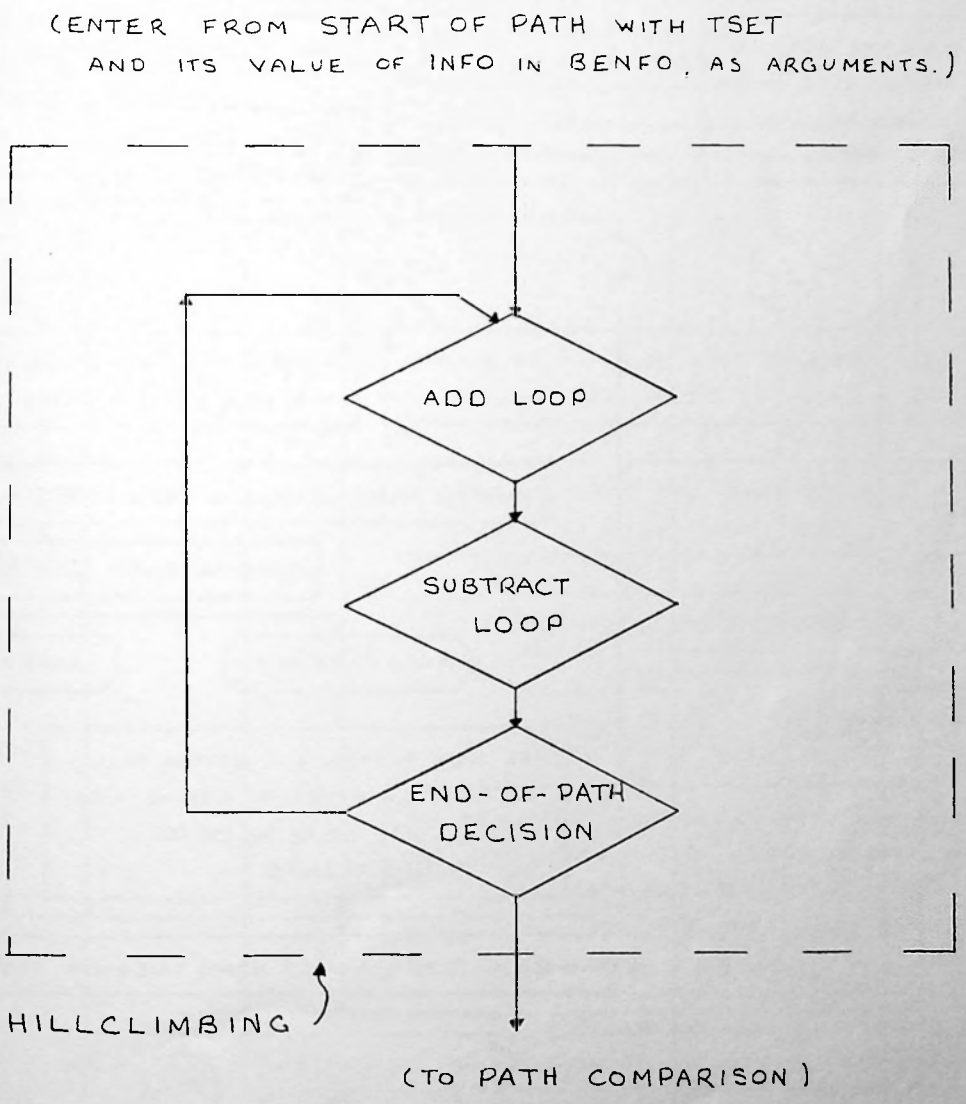


Figure 20

SMRMN*: HILLCLIMBING - (1) OUTLINE



* LGRMN IS SIMILAR IN STRUCTURAL DETAILS TO SMRMN

Figure 21

SMRMN : HILLCLIMBING - (2) DETAIL OF ADD LOOP

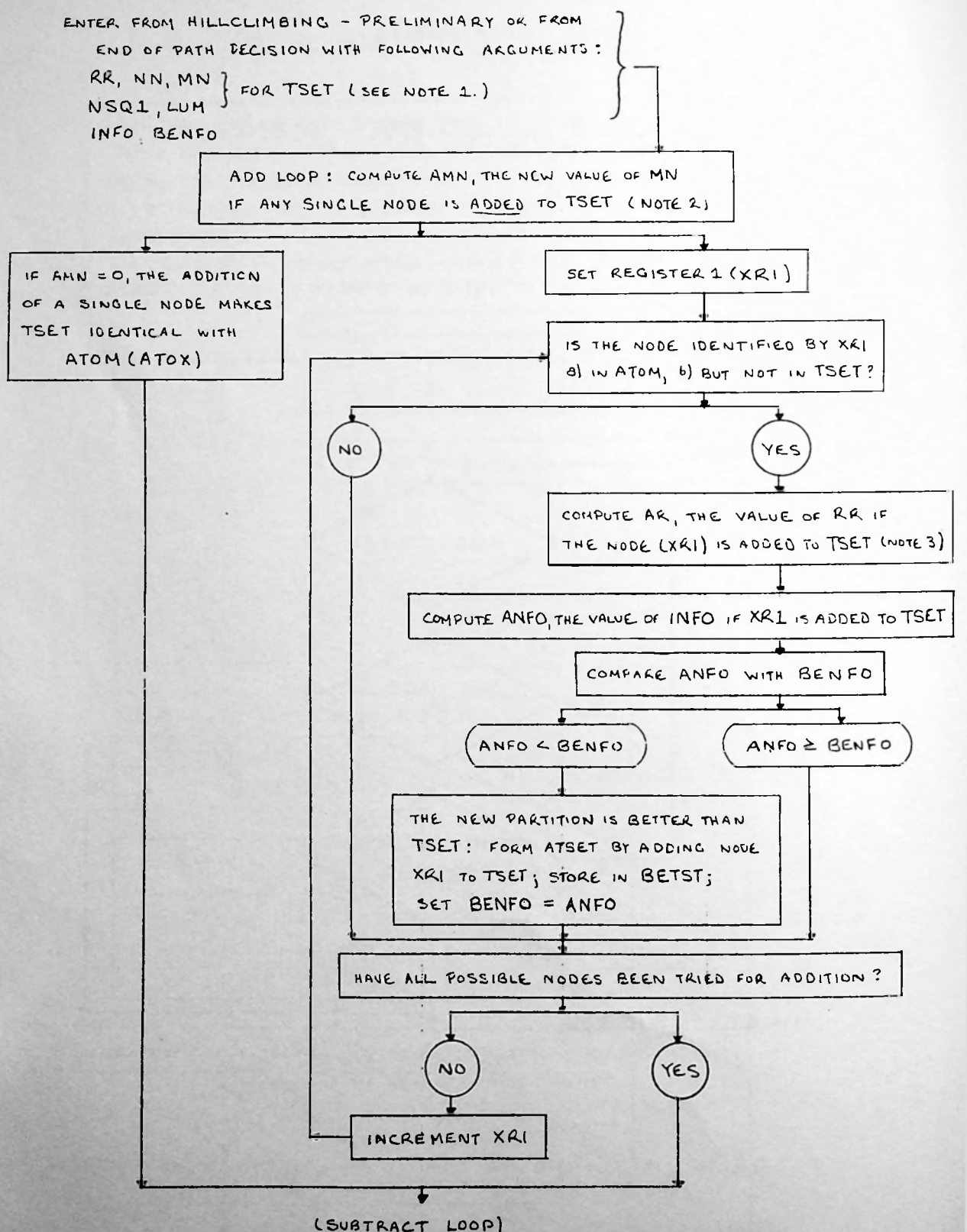


Figure 22

SMRMN : HILLCLIMBING -(3) DETAIL OF SUBTRACT LOOP

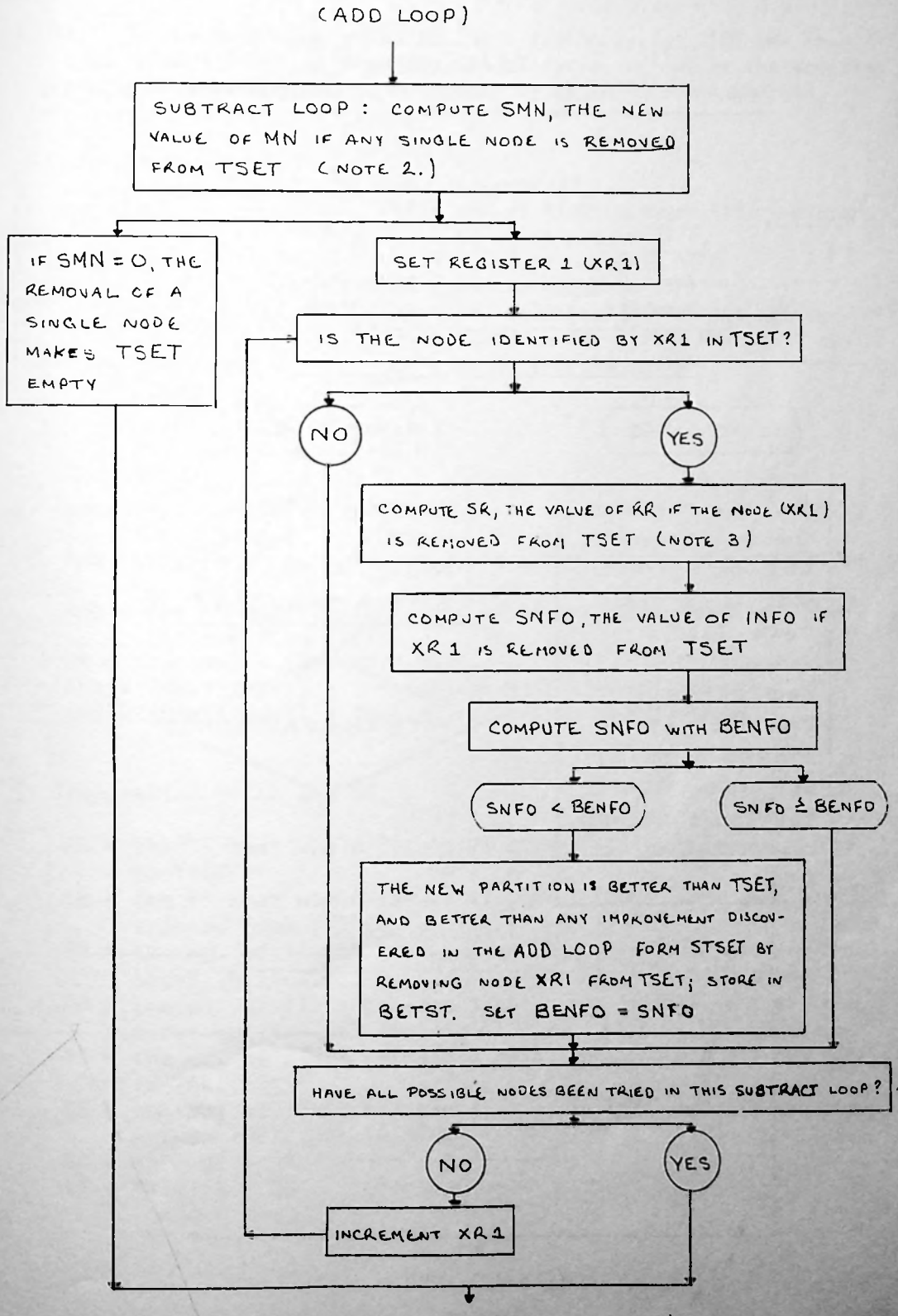


Figure 23

SMRMN : HILLCLIMBING - (4) END OF PATH DECISION

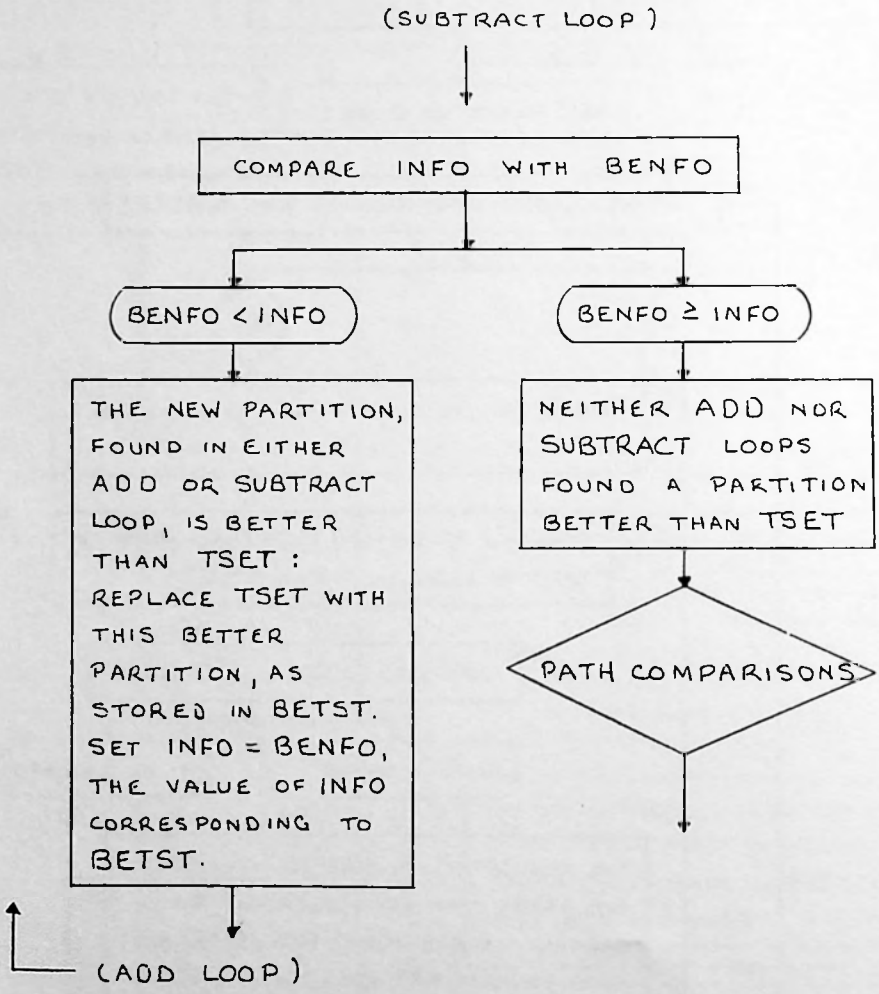
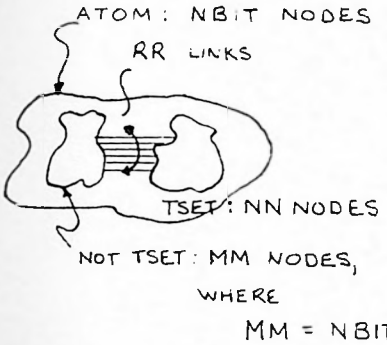


Figure 24

SMRMN : HILLCLIMBING - (5) NOTES

1. TSET is the starting point for this search cycle; TSET may have been generated by a previous search cycle, or may be the starting point of a search path, generated by GENERATE START OF PATH.



RR = no. of links between TSET and nodes of graph not in TSET
NBIT = no. of nodes in ATOM
NN = no. of nodes in TSET
MM = no. of nodes in ATOM but not in TSET MM = (NBIT-NN)
MN = product of NN and MM

2. Computation of AMN and SMN:

AMN = the MN that would result if a(any) single node were added to TSET.
SMN = the MN that would result if a(any) single node were subtracted from TSET.

$MN = (MM)(NN) = (NN)(NBIT-NN)$
$AMN = (MM-1)(NN+1) = (MM)(NN)-NN+MM-1 = MN+NBIT-NN-NN+1$
$SMN = (MM+1)(NN-1) = (MM)(NN)+NN-MM-1 = MN-NBIT+NN+NN-1$

3. Computation of AR and SR:

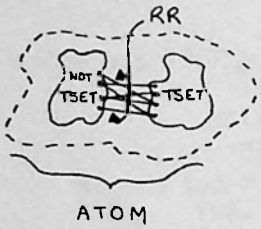
AR = the RR that would result if a(any) single node were added to TSET.
SR = the RR that would result if a(any) single node were subtracted from TSET.
PA = the no. of links between a node, not in TSET, and all the nodes in TSET.
QA = the no. of links between a node, not in TSET, and all the nodes <u>not</u> in TSET (but in ATOM).
PS = the no. of links between a node, in TSET, and all the nodes in TSET.
QS = the no. of links between a node, in TSET, and all the nodes <u>not</u> in TSET (but in ATOM).

AR = RR + QA - PA
SR = RR + PS - QS

Figure 25

SMRMN : HILLCLIMBING -(6) ALGORITHM FOR COMPUTING RR

RR IS THE NUMBER OF LINKS CONNECTING NODES OF TSET WITH NODES OF ATOM WHICH ARE NOT IN TSET.



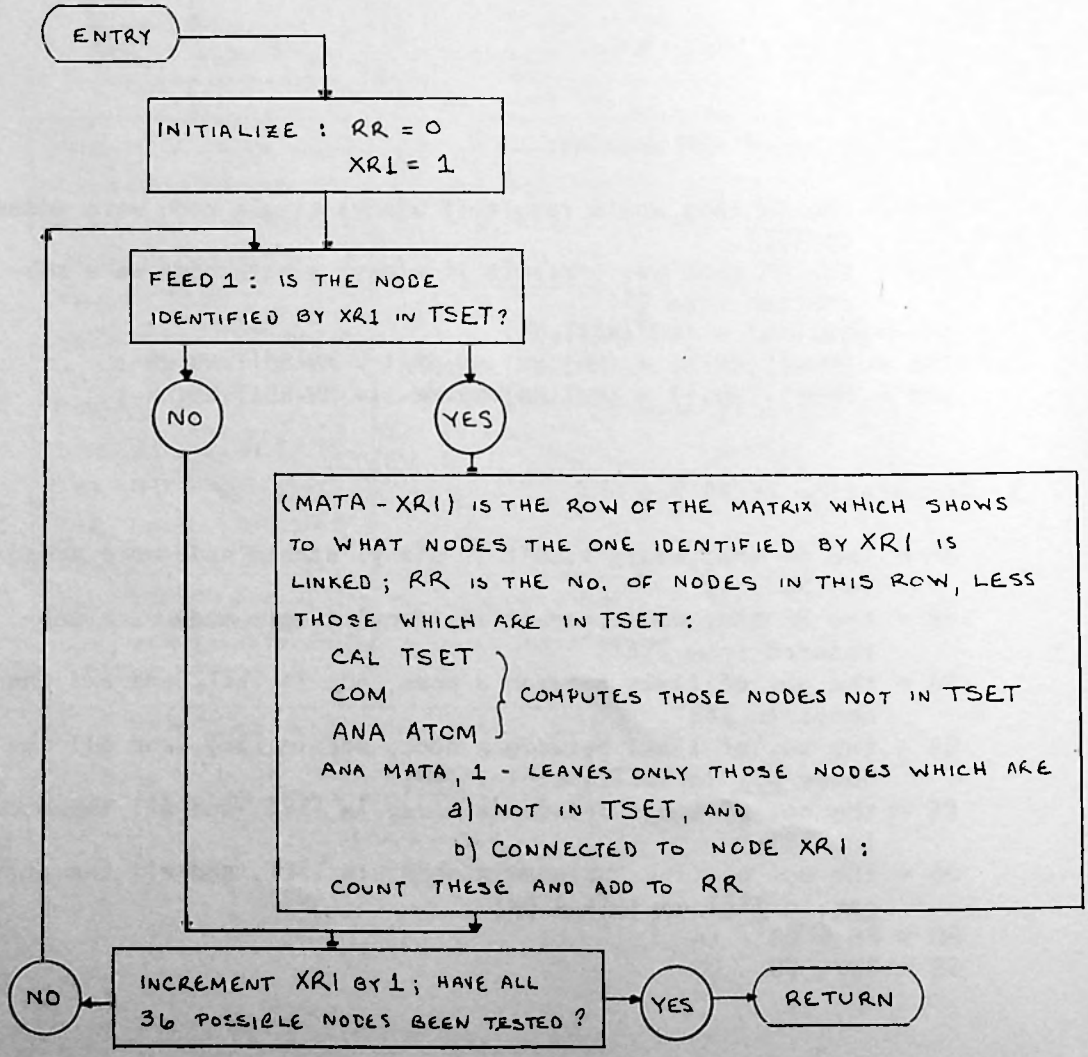


Figure 26

SMRMN: HILLCLIMBING - (7) ALTERNATIVE OUTLINE

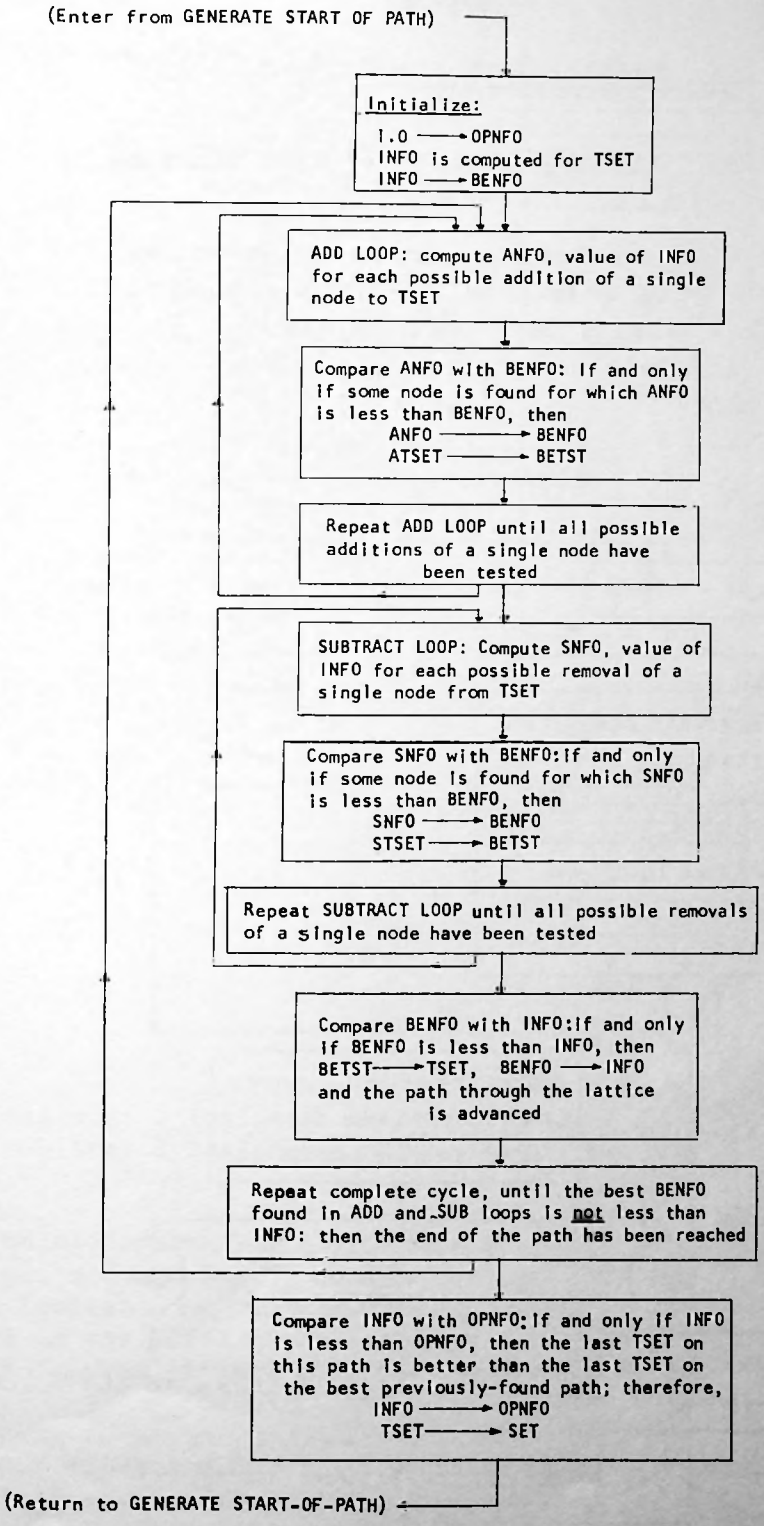


Figure 27

SMRMN : PATH COMPARISONS

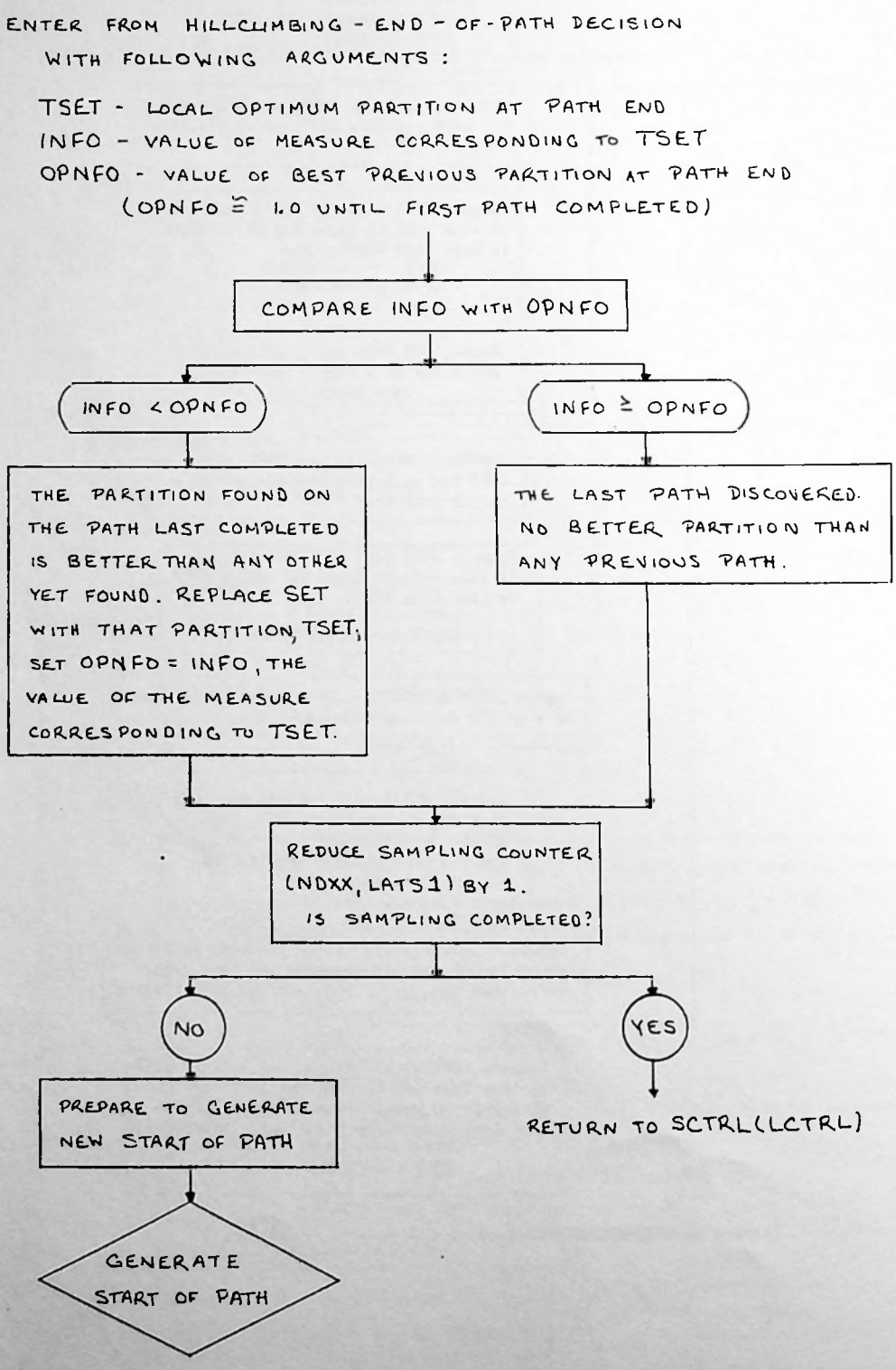
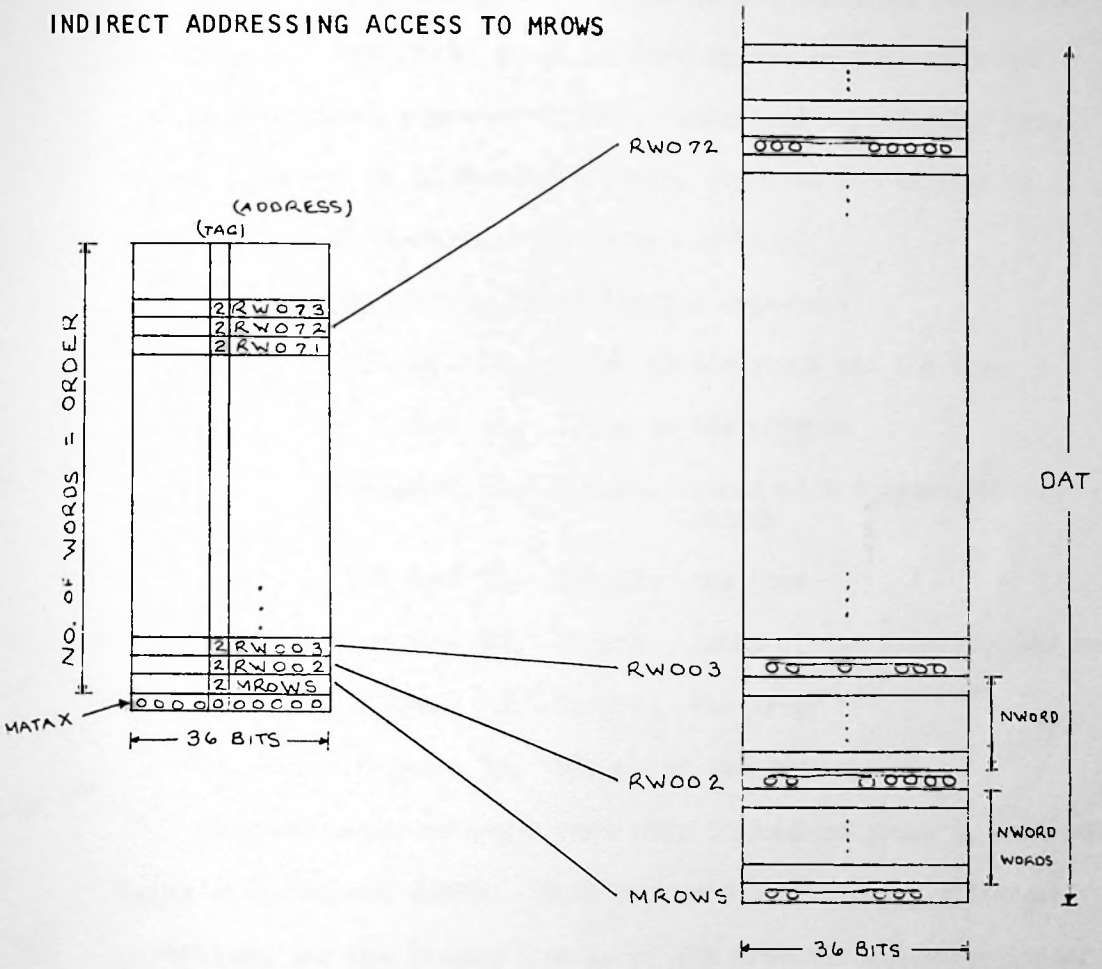


Figure 28

INDIRECT ADDRESSING ACCESS TO MROWS



XR1, Index register 1 indicates the row of matrix.
XR2, Index register 2 designates which word of that row.

For example:

CLA* MATAX,1 with XR1 = 72, XR2 = 2

will clear and add Row 72 word 2:
the (*) indicates indirect addressing and the tag of 2 in the MATAX block indicates that XR2 is used to determine the second-level address.

DAT, length of MROWS and DROWS, is ORDER units of (NWORD + 1) words each unit

D. TYPICAL INPUT AND OUTPUT

In the following pages we present the input and output for two graphs. The first graph we made up for testing purposes; the second graph represents the structure of a particular problem, reported on elsewhere.* (This graph is too complex to draw, and so a picture of it is not included.)

The material is in the following sequence:

Graph A: Figure 29, Sketch of the graph and its tree

Figure 30, Input to the program

Figure 31, Output: links of the symmetricised matrix

Figure 32, Output: the tree

Graph B: Figure 33, Output: links of the symmetricised matrix

Figure 34, Output: the tree

Figure 35, Sketch of the tree

Several computer runs were made to analyse graph B, each run using a different RANDM. Each run produced slightly different partitions at the lowest levels of the tree.** Investigation of the structures of the appropriate subgraphs indicated that this was due to the presence of two or more partitions of equal strength. On one run, the search procedure selected one such partition; on another run a second partition was selected. The subprogram NTRSC is designed to resolve the problem of several partitions of equal strength; however, NTRSC was not available at the time the analysis was performed. Therefore, this problem was resolved manually.

---

*Cf. Alexander and Manheim, THE DESIGN OF HIGHWAY INTERCHANGES.

**Figure 34 is, of course, the result of only one run. Therefore, it differs from Figure 35 in the way described.

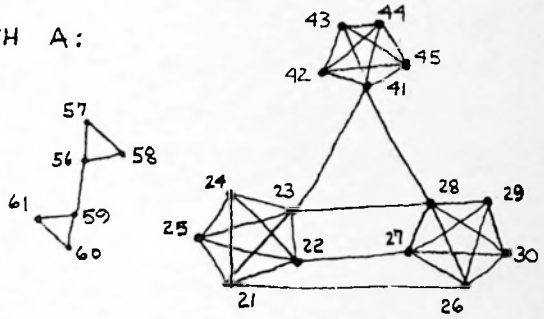
The procedure is illustrated in Figures 36 and 37. (All these graphs are subgraphs of Graph B.) In Figure 36, note that if two additional links were added, (73-79) and (72-81), this subgraph would be a complete graph. As it stands now, partitions I-I and II-II are of equal strength, the only non-arbitrary procedure is to take the intersection of the several equal partitions, as illustrated.

In Figure 37, note how the presence of equal-strength partitions can affect several levels of the tree.

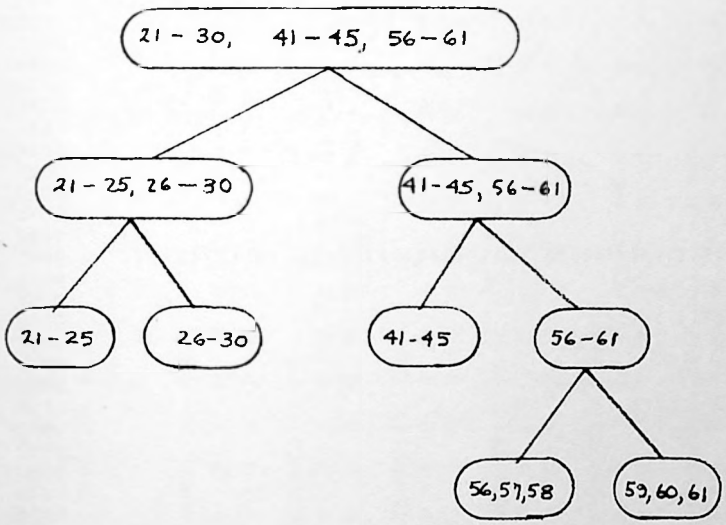
Figure 29.

GRAPH A: THE GRAPH AND ITS TREE

a) GRAPH A:



b) ITS TREE:



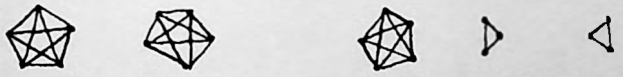


FIGURE 30, GRAPH A: INPUT TO THE PROGRAM

```
*     DATA
000110000000
402530215320
000030000000
```

```
GRAPH A
ORDER
 RANDM
LATIS
```

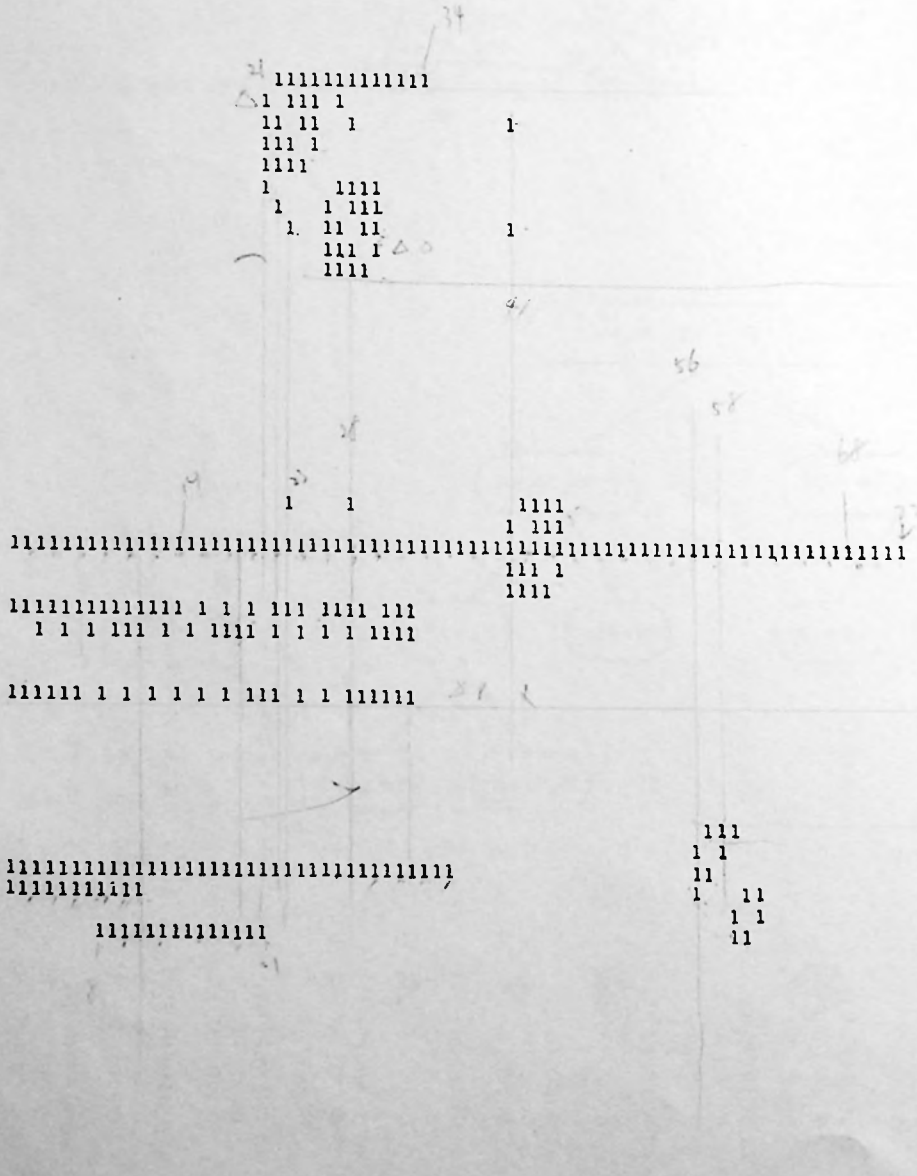


FIGURE 31, GRAPH A: OUTPUT - THE LINKS OF THE SYMMETRICISED MATRIX

REQUIREMENT 001 IS CONNECTED TO--
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

REQUIREMENT 002 IS CONNECTED TO--
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

REQUIREMENT 003 IS CONNECTED TO--
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

REQUIREMENT 004 IS CONNECTED TO--
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

REQUIREMENT 005 IS CONNECTED TO--
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

REQUIREMENT 006 IS CONNECTED TO--
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

REQUIREMENT 007 IS CONNECTED TO--
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

REQUIREMENT 008 IS CONNECTED TO--
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

REQUIREMENT 009 IS CONNECTED TO--
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

REQUIREMENT 010 IS CONNECTED TO--
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

REQUIREMENT 011 IS CONNECTED TO--
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

REQUIREMENT 012 IS CONNECTED TO--
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

REQUIREMENT 013 IS CONNECTED TO--
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

FIGURE 31-2

```
REQUIREMENT     014 IS CONNECTED TO--
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT     015 IS CONNECTED TO--
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT     016 IS CONNECTED TO--
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT     017 IS CONNECTED TO--
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT     018 IS CONNECTED TO--
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT     019 IS CONNECTED TO--
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT     020 IS CONNECTED TO--
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT     021 IS CONNECTED TO--
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 22 23 24 25 26  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT     022 IS CONNECTED TO--
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 21  0 23 24 25  0 27  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT     023 IS CONNECTED TO--
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 21 22  0 24 25  0  0 28  0  0  0  0  0  0  0  0
  0  0  0  0 41  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT     024 IS CONNECTED TO--
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 21 22 23  0 25  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT     025 IS CONNECTED TO--
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 21 22 23 24  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT     026 IS CONNECTED TO--
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 21  0  0  0  0  0 27 28 29 30  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

FIGURE 31-3

```
REQUIREMENT      027 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 22  0  0  0 26  0 28 29 30  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

REQUIREMENT      028 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 23  0  0 26 27  0 29 30  0  0  0  0  0  0
 0  0  0  0 41  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

REQUIREMENT      029 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 26 27 28  0 30  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

REQUIREMENT      030 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 26 27 28 29  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

REQUIREMENT      031 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

REQUIREMENT      032 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

REQUIREMENT      033 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

REQUIREMENT      034 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

REQUIREMENT      035 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

REQUIREMENT      036 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

REQUIREMENT      037 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

REQUIREMENT      038 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

REQUIREMENT      039 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

FIGURE 31-4

```
REQUIREMENT     040 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT     041 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 23  0  0  0  0 28  0  0  0  0  0  0  0  0
 0  0  0  0  0 42 43 44 45  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT     042 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0 41  0 43 44 45  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT     043 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0 41 42  0 44 45  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT     044 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0 41 42 43  0 45  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT     045 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0 41 42 43 44  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT     046 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT     047 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT     048 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT     049 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT     050 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT     051 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT     052 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

FIGURE 31 - 5

```
REQUIREMENT    053 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT    054 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT    055 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT    056 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 57 58 59  0  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT    057 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 56  0 58  0  0  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT    058 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 56 57  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT    059 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 56  0  0  0 60 61  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT    060 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 59  0 61  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT    061 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 59 60  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT    062 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT    063 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT    064 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT    065 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

FIGURE 31-6

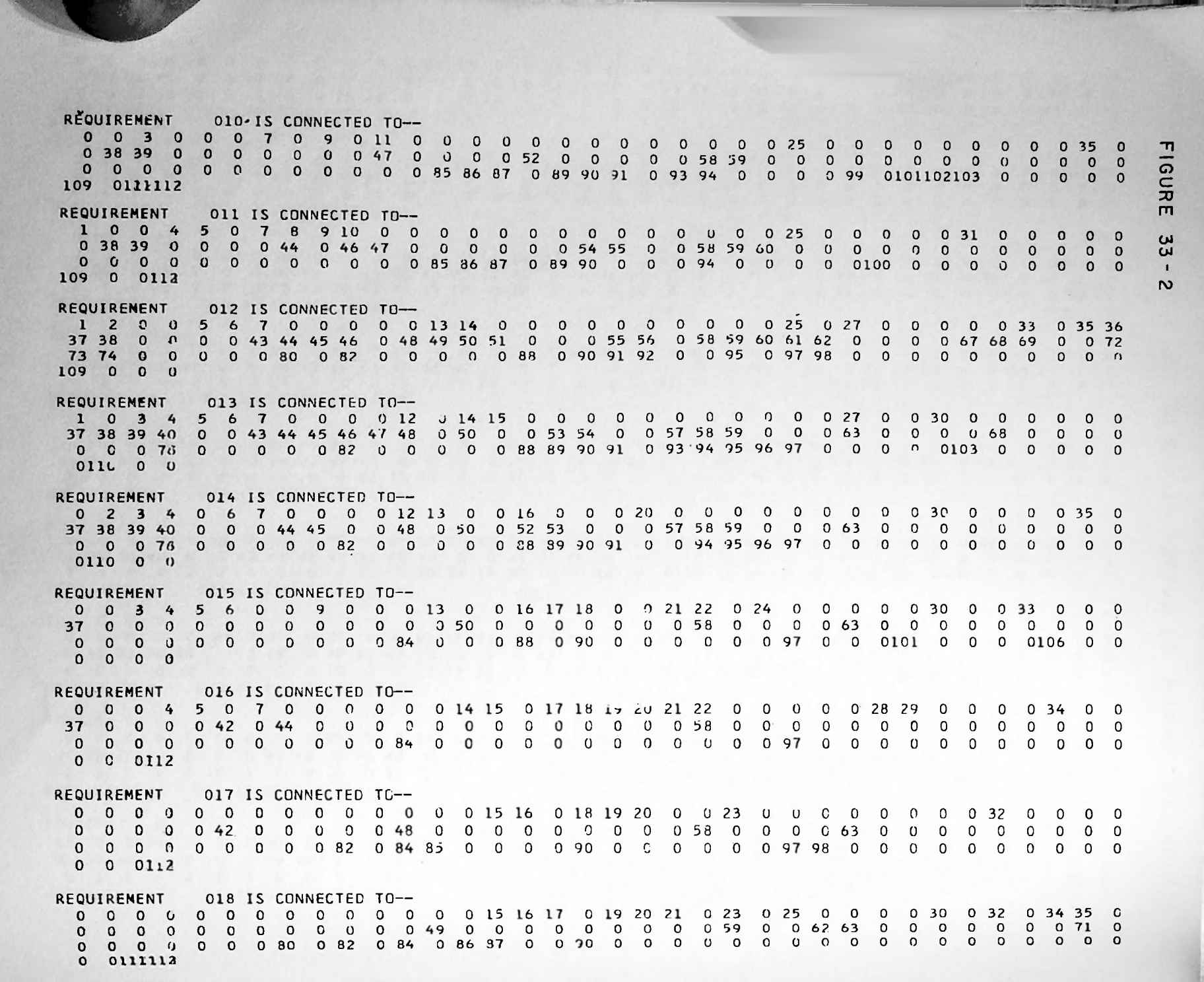
```
REQUIREMENT     066 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT     067 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT     068 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT     069 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT     070 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT     071 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
REQUIREMENT     072 IS CONNECTED TO--
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

FIGURE 32, GRAPH A: OUTPUT - THE TREE

```
CONTROL PASSED TO SCTRL
21 22 23 24 25 26 27 28 29 30  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0 41 42 43 44 45 56 57 58 59 60 61  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

NEW LEVEL OF HIERARCHY
 0  0  0  0  0 26 27 28 29 30  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
21 22 23 24 25  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0 41 42 43 44 45  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 56 57 58 59 60 61  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

NEW LEVEL OF HIERARCHY
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 56 57 58  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 59 60 61  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

NEW LEVEL OF HIERARCHY
CONTROL RETURNED TO LCTRL
```

FIGURE 33, GRAPH B: OUTPUT - LINKS OF THE SYMMETRICISED MATRIX

```
REQUIREMENT    001 IS CONNECTED TO--
  0  2  0  4   5  6  7  8  0  0 11 12 13  0  0  0  0  0  0  0  0  0  0 23  0  0 26 27  0  0 30  0  0 33  0 35  0
 37 38 39  0   0  0  0 44 45  0  0 48  0 50  0 52 53  0  0  0 57 58  0 60  0  0 65 66  0  0 69  0 71  0
 73  0  0 76   0  0  0 80  0 82  0  0 85  0 87 88  0  0  0  0 94 95 96 97  0  0  0  0  0  0 105  0 107108
109110  0  0

REQUIREMENT    002 IS CONNECTED TO--
  1  0  0  4   5  6  7  0  0  0 12  0 14  0  0  0  0  0  0  0 22 23 24  0  0 27  0  0 30  0  0  0  0 35  0
 37 38 39 40   0  0  0 44 45  0 47 48  0 50 51  0  0  0 56 57 58 59 60  0  0  0 65 66  0 68  0 70  0  0
  0  0  0  0   0  0  0  0  0 82  0  0 85  0 87  0  0 90  0  0  0 94 95 96 97  0  0100  0102  0 0105  0 0108
109110  0  0

REQUIREMENT    003 IS CONNECTED TO--
  0  0  0  0   5  0  0  0  9 10  0  0 13 14 15  0  0  0 19 20  0 22  0 24  0  0  0 29  0  0  0  0 35  0
  0 38 39 40  41  0 43 44  0 46 47 48  0  0  0 53  0  0  0  0 59  0  0  0 65  0  0  0  0  0  0  0  0
  0  0  0  0   0  0  0 80  0 82  0 84 85 86 87  0 89 90 91  0 93 94  0 96  0  0 99 C101102103  0  0  0  0
109  0  0  0

REQUIREMENT    004 IS CONNECTED TO--
  1  2  0  0   5  6  7  0  0  0 11  0 13 14 15 16  0  0 19  0  0 22  0 24  0  0  0 28 29  0  0  0  0 35  0
 37 38 39 40   0  0 43 44 45 46 47 48 49  0  0 52 53 54 55  0  0  0 59 60  0  0  0  0  0  0  0  0  0  0
  0  0  0  0   0  0  0 80  0 82  0 84 85 86 87 88 89 90 91  0  0 94 95 96 97  0  0  0102 0104105  0  0  0
109  0111  0

REQUIREMENT    005 IS CONNECTED TO--
  1  2  3  4   0  6  7  0  0  0 11 12 13  0 15 16  0  0  0  0  0  0  0 23  0  0  0  0 26 27 28  0  0  0 33  0 35 36
 37 38 39  0   0  0  0 44 45 46 47  0  0 50  0 52  0  0 54 55 56 57  0 59 60 61  0 63  0 65 66  0 68 69  0 71  0
  0  0  0 76   0  0  0 80  0 82  0  0  0  0 83  0  0 89 90 91  0 93  0 95 96 97 98  0  0  0  0  0  0 107  0
  0110  0  0

REQUIREMENT    006 IS CONNECTED TO--
  1  2  0  4   5  0  7  0  0  0  0 12 13 14 15  0  0  0  0 20 21  0 23  0  0  0  0  0  0  0 26 27 28  0  0  0 33 34  0 36
 37 38  0 40  41  0  0 44 45 46 47 48 49 50 51  0  0  0 55  0  0  0 59 60  0  0  0  0  0  0  0 62 63 64 65 66  0  0 69 70 71  0
  0  0  0  0   0  0  0 80  0 82  0 84  0  0  0 88  0 90 91  0  0 94  0 96  0  0  0  0  0102 0104105  0107  0
109  0 0112

REQUIREMENT    007 IS CONNECTED TO--
  1  2  0  4   5  6  0  0  9 10 11 12 13 14  0 16  0  0  0 20  0  0 23  0  0  0 27  0  0 30  0  0  0  0 35 36
 37 38  0 40   0  0  0 44 45  0 47 48 49 50 51 52  0 54 55 56  0 58 59 60 61  0 63  0 65 66  0 68  0 70 71  0
  0  0  0  0   0  0  0 80  0 82  0  0  0 86 87 88 89  0 91  0 93 94 95 96 97  0  0  0  0  0  0 107  0
  0110  0  0

REQUIREMENT    008 IS CONNECTED TO--
  1  0  0  0   0  0  0  0  0  0 11  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0 39  0   0  0  0  0 44  0  0 47  0  0  0  0  0  0  0  0  0  0  0 59  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0 76  77 78 79  0  0  0  0  0  0 86 87  0 89  0  0 91  0  0  0  0  0  0  0  0100  0  0  0  0  0 0107108
109  0  0  0

REQUIREMENT    009 IS CONNECTED TO--
  0  0  3  0   0  0  7  0  0 10 11  0  0  0  0 15  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0 38 39 40   0  0 43 44  0  0 47  0  0  0  0  0 53  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0   0  0  0  0  0  0  0  0  0 85 86  0  0 89 90 91  0 93  0  0  0  0  0  0  0  0  0  0  0  0  0  0
109  0111112
```



REQUIREMENT 010 IS CONNECTED TO--
0 0 3 0 0 0 7 0 9 0 11 0 0 0 0 0 0 0 0 0 0 0 0 0 25 0 0 0 0 0 0 0 0 0 35 0
0 38 39 0 0 0 0 0 0 0 47 0 0 0 0 52 0 0 0 0 0 58 59 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 85 86 87 0 89 90 91 0 93 94 0 0 0 0 99 0101102103 0 0 0 0 0
109 0111112

REQUIREMENT 011 IS CONNECTED TO--
1 0 0 4 5 0 7 8 9 10 0 0 0 0 0 0 0 0 0 0 0 0 0 25 0 0 0 0 0 31 0 0 0 0 0
0 38 39 0 0 0 0 44 0 46 47 0 0 0 0 0 0 54 55 0 0 58 59 60 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 85 86 87 0 89 90 0 0 0 94 0 0 0 0 0100 0 0 0 0 0 0 0 0
109 0 0112

REQUIREMENT 012 IS CONNECTED TO--
1 2 0 0 5 6 7 0 0 0 0 0 13 14 0 0 0 0 0 0 0 0 0 0 25 0 27 0 0 0 0 0 33 0 35 36
37 38 0 0 0 0 43 44 45 46 0 48 49 50 51 0 0 0 55 56 0 58 59 60 61 62 0 0 0 0 67 68 69 0 0 72
73 74 0 0 0 0 0 80 0 82 0 0 0 0 0 88 0 90 91 92 0 0 95 0 97 98 0 0 0 0 0 0 0 0 0 0
109 0 0 0

REQUIREMENT 013 IS CONNECTED TO--
1 0 3 4 5 6 7 0 0 0 0 12 0 14 15 0 0 0 0 0 0 0 0 0 0 0 27 0 0 30 0 0 0 0 0 0
37 38 39 40 0 0 43 44 45 46 47 48 0 50 0 0 53 54 0 0 57 58 59 0 0 0 63 0 0 0 0 68 0 0 0 0
0 0 0 76 0 0 0 0 0 82 0 0 0 0 0 88 89 90 91 0 93 94 95 96 97 0 0 0 0 0103 0 0 0 0 0
0110 0 0

REQUIREMENT 014 IS CONNECTED TO--
0 2 3 4 0 6 7 0 0 0 0 12 13 0 0 16 0 0 0 20 0 0 0 0 0 0 0 0 0 30 0 0 0 0 35 0
37 38 39 40 0 0 0 44 45 0 0 48 0 50 0 52 53 0 0 0 57 58 59 0 0 0 63 0 0 0 0 0 0 0 0 0
0 0 0 76 0 0 0 0 0 82 0 0 0 0 0 88 89 90 91 0 0 94 95 96 97 0 0 0 0 0 0 0 0 0 0 0
0110 0 0

REQUIREMENT 015 IS CONNECTED TO--
0 0 3 4 5 6 0 0 9 0 0 0 13 0 0 16 17 18 0 0 21 22 0 24 0 0 0 0 0 30 0 0 33 0 0 0
37 0 0 0 0 0 0 0 0 0 0 0 0 50 0 0 0 0 0 0 0 58 0 0 0 0 63 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 84 0 0 0 88 0 90 0 0 0 0 0 0 97 0 0 0101 0 0 0 0106 0 0
0 0 0 0

REQUIREMENT 016 IS CONNECTED TO--
0 0 0 4 5 0 7 0 0 0 0 0 0 14 15 0 17 18 19 20 21 22 0 0 0 0 0 28 29 0 0 0 0 34 0 0
37 0 0 0 0 42 0 44 0 0 0 0 0 0 0 0 0 0 0 0 0 58 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 84 0 0 0 0 0 0 0 0 0 0 0 0 97 0 0 0 0 0 0 0 0 0 0 0
0 0 0112

REQUIREMENT 017 IS CONNECTED TO--
0 0 0 0 0 0 0 0 0 0 0 0 0 0 15 16 0 18 19 20 0 0 23 0 0 0 0 0 0 0 0 32 0 0 0 0
0 0 0 0 0 42 0 0 0 0 0 48 0 0 0 0 0 0 0 0 0 58 0 0 0 0 63 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 82 0 84 85 0 0 0 0 90 0 0 0 0 0 0 97 98 0 0 0 0 0 0 0 0 0 0
0 0 0112

REQUIREMENT 018 IS CONNECTED TO--
0 0 0 0 0 0 0 0 0 0 0 0 0 0 15 16 17 0 19 20 21 0 23 0 25 0 0 0 0 30 0 32 0 34 35 0
0 0 0 0 0 0 0 0 0 0 0 0 49 0 0 0 0 0 0 0 0 0 59 0 0 0 62 63 0 0 0 0 0 0 0 71 0
0 0 0 0 0 0 0 80 0 82 0 84 0 86 87 0 0 90 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0111112

FIGURE 33 - 2

```
REQUIREMENT    019 IS CONNECTED TO--
  0  0  3  4  0  0  0  0  0  0  0  0  0  0  0 16 17 18  0 20  0  0  0  0  0  0  0  0  0  0 32  0  0  0  0
  0  0  0  0  0 42  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 58  0  0  0 62 63  0  0  0  0  0  0 71  0
  0  0  0  0  0  0  0  0  0 82  0 84 85  0  0  0  0 90  0 92  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0111112

REQUIREMENT    020 IS CONNECTED TO--
  0  0  3  0  0  6  7  0  0  0  0  0  0 14  0 16 17 18 19  0 21  0 23  0 25  0  0  0  0 30  0 32  0  0  0  0
 37  0  0  0  0 42  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 63  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0 80  0 32  0 84  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0110  0112

REQUIREMENT    021 IS CONNECTED TO--
  0  0  0  0  0  6  0  0  0  0  0  0  0  0 15 16  0 18  0 20  0  0  0  0  0  0 27 28  0  0  0  0  0  0  0  0
  0  0  0  0  0  0 43  0  0  0  0  0 49  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0 79 80  0  0  0 84  0  0 87  0 89  0  0  0  0  0  0  0  0  0 99  0  0  0  0  0  0  0  0
  0  0  0112

REQUIREMENT    022 IS CONNECTED TO--
  0  2  3  4  0  0  0  0  0  0  0  0  0 15 16  0  0  0  0  0  0 23 24  0  0  0  0 29  0 31 32  0  0  0 36
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 54  0  0  0  0  0  0 62  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 89  0  0  0  0  0  0  0 98  0  0  0  0  0  0  0  0  0  0  0
109110  0  0

REQUIREMENT    023 IS CONNECTED TO--
  1  2  0  0  5  6  7  0  0  0  0  0  0  0  0  0 17 18  0 20  0 22  0 24 25  0  0  0 29 30  0 32  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 63  0  0  0  0 68  0  0  0  0
  0  0  0  0  0  0  0  0  0 82  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0105  0  0  0
109110111112

REQUIREMENT    024 IS CONNECTED TO--
  0  2  3  4  0  0  0  0  0  0  0  0  0 15  0  0  0  0  0  0 22 23  0 25  0  0 28 29 30  0 32  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 63  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0 82  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 98  0  0  0  0  0105  0  0  0
109  0  0  0

REQUIREMENT    025 IS CONNECTED TO--
  0  0  0  0  0  0  0  0  9 10 11 12  0  0  0  0  0 18  0 20  0  0 23 24  0  0  0  0  0 30 31 32  0  0  0  0
 37 38 39 40 41  0  0  0 45 46 47 48 49 50  0  0 53 54 55  0  0 58 59 60  0 62  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0 84 85 86 87 88 89 90 91 92 93 94  0 96  0 98 99100101  0103  0105  0  0108
  0110111112

REQUIREMENT    026 IS CONNECTED TO--
  1  0  0  0  0  6  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 27 28  0  0  0  0 33 34  0 36
  0  0  0  0  0 42  0 44  0  0  0 48  0  0  0  0  0  0  0  0  0  0  0 60  0  0 63  0  0  0  0  0 69 70 71 72
 73  0  0  0  0  0  0 80 81  0  0 84 85  0  0  0  0  0  0 91  0  0  0 95 96  0  0  0  0  0  0  0  0  0106107  0
  0  0111112

REQUIREMENT    027 IS CONNECTED TO--
  1  2  0  0  0  6  7  0  0  0  0 12 13  0  0  0  0  0  0  0 21  0  0  0  0 26  0 28  0  0  0  0 33 34  0  0
  0  0 39 40  0  0  0 44 45  0 47 48  0  0 51  0 53  0  0  0  0  0  0 59 60 61  0 63  0 65 66  0 68  0  0 71 72
 73  0  0  0  0  0  0 80  0 82  0  0  0 86  0 88  0 90 91  0 93 94 95 96  0  0  0  0  0  0 0104  0  0  0108
  0  0  0  0
```

```
REQUIREMENT    028 IS CONNECTED TO--
  0  0  0  4  0  6  0  0  0  0  0  0  0  0  0 16  0  0  0  0 21  0  0 24  0 26 27  0  0 30 31  0  0  0  0  0
  0  0  0  0  0  0  0 44  0 46 47  0  0 50  0  0  0 54 55  0  0  0 59 60  0  0 63  0 65  0  0 68  0  0 71  0
  0  0  0  0  0  0 79  0  0 82  0 84  0  0  0 88  0 90 91  0  0  0  0 96  0  0  0  0  0  0  0  0  0 0107  0
  0  0  0  0

REQUIREMENT    029 IS CONNECTED TO--
  0  0  3  4  0  0  0  0  9  0  0  0  0  0  0 16  0  0  0  0  0 22 23 24  0  0  0  0  0  0  0  0  0  0  0
 37  0  0  0  0  0  0 44  0 46  0  0  0  0  0  0  0 54 55  0  0  0  0 60  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0 79  0  0  0  0 84  0 86  0 88  0  0 91  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0

REQUIREMENT    030 IS CONNECTED TO--
  1  2  0  0  5  0  7  0  0  0  0  0 13 14 15  0  0 18  0 20  0  0 23 24 25  0  0 28  0  0  0 32  0  0  0 36
  0 38 39 40  0  0 43  0  0 46 47 48  0  0  0  0  0 54  0  0 57  0 59 60  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0 79 80  0  0  0  0 85  0  0  0 89  0  0  0  0  0  0 96  0 98  0  0  0  0  0  0  0  0  0
109110111  0

REQUIREMENT    031 IS CONNECTED TO--
  0  0  0  0  0  0  0  0  0  0 11  0  0  0  0  0  0  0  0  0  0 22  0  0 25  0  0 28  0  0  0 32  0  0  0  0
  0  0 39  0  0  0  0  0  0 46  0  0 49  0  0 52  0 54  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0 81  0  0 84  0 86 87  0 89  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
109110  0  0

REQUIREMENT    032 IS CONNECTED TO--
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 17 18 19 20  0 22 23 24 25  0  0  0  0 30 31  0  0  0  0  0
 37 38 39  0  0  0 43  0 45 46 47 48 49  0  0  0  0  0 55  0  0  0  0  0  0 62 63  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0 80  0 82  0  0  0 86 87  0  0  0 91  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
109110  0  0

REQUIREMENT    033 IS CONNECTED TO--
  1  0  0  0  5  6  0  0  0  0  0 12  0  0 15  0  0  0  0  0  0  0  0  0  0  0 26 27  0  0  0  0  0  0 35  0
 37 38 39 40  0  0 43  0 45  0  0 48 49 50 51  0  0  0  0  0  0 58 59 60 61 62 63 64 65 66  0 68 69 70 71 72
 73 74  0 76 77 78  0 80 81 82 83  0  0 86  0 88  0 90 91  0 93 94 95 96  0  0  0  0 0102  0  0  0  0  0  0
  0 0111  0

REQUIREMENT    034 IS CONNECTED TO--
  0  0  0  0  0  6  0  0  9  0  0  0  0  0  0 16  0 18  0  0  0  0  0  0  0 26 27  0  0  0  0  0  0  0  0
  0  0  0 40  0  0  0  0 45  0  0  0 49  0  0  0 53  0  0  0  0  0  0  0  0  0  0  0 66  0  0  0  0 71  0
  0  0  0  0  0  0  0 80  0  0  0  0  0  0  0 88  0 90  0  0 93  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0

REQUIREMENT    035 IS CONNECTED TO--
  1  2  3  4  5  0  7  0  9 10  0 12  0 14  0  0  0 18  0  0  0  0  0  0  0  0  0  0  0  0  0 33  0  0  0
 37 38 39 40 41  0 43  0 45  0 47 48 49 50 51  0 53  0  0  0  0  0 59 60 61  0 63  0  0  0  0 68  0  0  0  0
  0 74  0 76  0  0  0  0  0  0  0  0  0  0  0 88 89 90 91 92  0  0 95 96  0  0  0  0  0  0  0  0  0  0  0
  0 0111112

REQUIREMENT    036 IS CONNECTED TO--
  0  0  0  0  5  6  7  0  0  0  0 12  0  0  0  0  0  0  0  0  0 22  0  0  0 26  0  0  0 30  0  0  0  0  0  0
  0  0  0  0  0  0  0 44 45  0  0  0  0 50  0  0  0  0  0  0  0  0  0  0  0  0  0 64 65 66  0  0  0 70 71 72
  0  0  0  0  0  0  0  0  0 82 83  0  0  0  0  0  0  0  0  0  0  0  0 96 97  0  0  0  0  0  0  0  0  0  0
```

FIGURE 33-4

```
REQUIREMENT     037 IS CONNECTED TO--
  1  2  0  4  5  6  7  0  0  0  0 12 13 14 15 16  0  0  0 20  0  0  0  0 25  0  0  0 29  0  0 32 33  0 35  0
  0 38  0 40 41  0 43 44 45 46 47 48  0 50  0  0 53  0  0 56  0  0 59  0  0  0 63 64 65 66  0 68 69 70 71 72
  0  0  0  0  0  0  0  0 81 82 83  0  0 86  0 88 89 90 91  0  0  0 95 96 97 98  0  0  0  0  0  0  0 0107  0
109  0111112

REQUIREMENT     038 IS CONNECTED TO--
  1  2  3  4  5  6  7  0  9 10 11 12 13 14  0  0  0  0  0  0  0  0  0  0 25  0  0  0  0 30  0 32 33  0 35  0
 37  0 39  0  0  0 43 44 45 46 47 48 49 50 51  0 53 54  0  0  0 58  0 60 61  0  0 64 65 66  0  0 69  0  0 72
  0 74  0  0  0  0  0  0 81 82 83  0  0 86 87 88 89 90 91 92 93 94 95 96 97 98 99  0101  0103104  0 0107  0
109  0111112

REQUIREMENT     039 IS CONNECTED TO--
  1  2  3  4  5  6  0  8  9 10 11  0 13 14  0  0  0  0  0  0  0  0  0  0 25  0 27  0  0 30 31 32 33  0 35  0
  0 38  0 40 41  0 43 44 45 46 47 48 49 50 51 52 53 54 55  0 57 58 59  0  0 62  0  0 65 66  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0 82  0  0 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99  0101  0103  0105  0107  0
1091I0  0  0

REQUIREMENT     040 IS CONNECTED TO--
  0  2  3  4  0  6  7  0  9  0  0  0 13 14  0  0  0  0  0  0  0  0  0  0 25  0 27  0  0 30  0  0 33 34 35  0
 37  0 39  0 41  0  0  0 45  0 47 48 49  0  0 52 53 54  0  0 57  0 59 60  0 62 63  0 65 66  0  0  0 70  0  0
  0  0  0  0  0  0  0 80  0 82  0  0  0 86 87 88 89 90 91  0 93 94 95 96  0  0 99  0101  0  0  0105  0107  0
109  0111112

REQUIREMENT     041 IS CONNECTED TO--
  0  0  3  0  0  6  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 25  0  0  0  0  0  0  0  0  0 35  0
 37  0 39 40  0 42 43  0  0  0  0  0  0  0  0  0 53  0  0  0  0 58 59  0  0  0  0  0 65  0 67  0  0  0  0  0
  0 74  0 76 77 78  0 80 81  0  0  0  0 86 87  0 89  0 91 92 93 94  0 96  0  0  0  0  0102  0  0  0  0  0  0
  0  0  0  0

REQUIREMENT     042 IS CONNECTED TO--
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 16 17  0 19 20  0  0  0  0  0 26  0  0  0  0  0  0  0  0  0  0
  0  0  0  0 41  0  0  0  0 46  0  0  0  0  0  0  0  0  0 56  0 58 59  0  0 62  0 64  0 66 67  0  0  0 71  0
  0 74 75 76 77 78  0 80 81 82  0  0  0  0  0  0  0  0 91  0  0  0  0 96  0  0  0  0  0102  0  0  0  0  0  0
  0  0  0  0

REQUIREMENT     043 IS CONNECTED TO--
  0  0  3  4  0  0  0  0  9  0  0 12 13  0  0  0  0  0  0  0 21  0  0  0  0  0  0  0  0 30  0 32 33  0 35  0
 37 38 39  0 41  0  0 44  0  0  0 48  0 50  0 52 53  0  0  0 57  0 59  0  0 62 63  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0 82 83  0  0 86 87 88 89  0 91  0  0  0  0 96 97 98  0  0  0  0  0  0  0  0  0  0
109  0111112

REQUIREMENT     044 IS CONNECTED TO--
  1  2  3  4  5  6  7  8  9  0 11 12 13 14  0 16  0  0  0  0  0  0  0  0  0 26 27 28 29  0  0  0  0  0  0 36
 37 38 39  0  0  0 43  0 45  0 47 48  0 50  0 52 53 54 55 56  0  0 59  0  0  0 63  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0 80  0 82  0  0  0 86  0  0 89 90 91  0 93 94 95 96 97 98  0100101  0103  0  0 0107108
1091I0I2I1112

REQUIREMENT     045 IS CONNECTED TO--
  1  2  0  4  5  6  7  0  0  0  0 12 13 14  0  0  0  0  0  0  0  0  0  0 25  0 27  0  0  0  0 32 33 34 35 36
 37 38 39 40  0  0  0 44  0  0 47 48 49 50  0  0 53 54 55 56  0  0 59 60 61  0 63 64 65 66  0  0  0  0 71 72
 73  0  0  0  0  0  0 80  0 82  0  0  0 86 87 88 89 90 91  0 93 94 95 96 97 98  0  0  0  0103104  0 0107108
109  0  0112
```

FIGURE 33-5

```
REQUIREMENT    046 IS CONNECTED TO--
  0  0  3  4  5  6  0  0  0  0 11 12 13  0  0  0  0  0  0  0  0  0  0  0 25  0  0 28 29 30 31 32  0  0  0  0
 37 38 39  0  0 42  0  0  0  0 47 48 49  0  0  0 53 54  0  0  0 58 59 60 61  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0 80  0 82  0 84 85 86 87 88 89 90 91  0 93 94 95 96  0  0 99100101  0  0  0  0  0 0108
  0110  0112

REQUIREMENT    047 IS CONNECTED TO--
  0  2  3  4  5  6  7  8  9 10 11  0 13  0  0  0  0  0  0  0  0  0  0  0 25  0 27 28  0 30  0 32  0  0 35  0
 37 38 39 40  0  0  0 44 45 46  0  0  0 51  0 53 54 55  0  0  0 59  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0 80  0 82  0  0 85 86 87 88 89  0 91  0  0 94  0  0  0  0 99100101102  0  0  0  0 0108
  0110  0  0

REQUIREMENT    048 IS CONNECTED TO--
  1  2  3  4  0  6  7  0  0  0  0 12 13 14  0  0 17  0  0  0  0  0  0  0  0 25 26 27  0  0 30  0 32 33  0 35  0
 37 38 39 40  0  0 43 44 45 46  0  0 49 50 51 52 53  0  0  0  0  0 59 60 61  0 63  0 65 66  0 68 69  0  0 72
 73 74  0  0  0 78  0 80  0 82  0  0  0 86  0 88 89 90 91  0  0  0 95 96 97  0  0  0101  0  0  0  0  0  0  0
109  0  0112

REQUIREMENT    049 IS CONNECTED TO--
  0  0  0  4  0  6  7  0  0  0  0 12  0  0  0  0  0 18  0  0 21  0  0  0 25  0  0  0  0  0 31 32 33 34 35  0
  0 38 39 40  0  0  0  0 45 46  0 48  0  0 51  0 53  0  0  0 57  0 59 60 61 62 63  0  0 66 67  0  0  0  0  0
  0  0  0  0  0  0  0  0 81 82 83 84  0 86 87 88 89 90  0 92 93 94 95 96  0  0  0  0  0  0  0  0  0  0  0  0
  01101111112

REQUIREMENT    050 IS CONNECTED TO--
  1  2  0  0  5  6  7  0  0  0  0 12 13 14 15  0  0  0  0  0  0  0  0  0 25  0  0 28  0  0  0  0 33  0 35 36
 37 38 39  0  0  0 43 44 45  0  0 48  0  0  0 52  0 54  0  0  0  0  0  0 61  0 63  0  0  0  0  0  0 70  0  0
 73  0  0  0  0  0  0  0  0 82  0  0  0  0  0 88 89 90 91 92  0  0 95 96 97 98 99  0101102103  0105  0  0  0
109  0  0112

REQUIREMENT    051 IS CONNECTED TO--
  0  2  0  0  0  6  7  0  0  0  0 12  0  0  0  0  0  0  0  0  0  0  0  0  0  0 27  0  0  0  0  0 33  0 35  0
  0 38 39  0  0  0  0  0  0  0 47 48 49  0  0  0 53  0  0  0 57  0 59 60 61  0 63  0 65  0  0  0 69  0  0  0
  0 74  0  0  0  0  0  0 81  0  0  0  0  0  0 88  0 90 91  0  0  0  0 96  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0112

REQUIREMENT    052 IS CONNECTED TO--
  1  0  0  4  5  0  7  0  0 10  0  0  0 14  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 31  0  0  0  0  0
  0  0 39 40  0  0 43 44  0  0  0 48  0 50  0  0 53 54  0  0 57  0 59 60  0 62 63  0 65  0  0  0  0 70  0  0
  0  0  0  0  0  0  0  0  0 82  0  0  0 86 87 88 89 90  0  0 93 94  0 96 97 98  0  0101  0  0  0  0  0  0  0
109110111112

REQUIREMENT    053 IS CONNECTED TO--
  1  0  3  4  0  0  0  0  9  0  0  0 13 14  0  0  0  0  0  0  0  0  0  0 25  0 27  0  0  0  0  0  0 34 35  0
 37 38 39 40 41  0 43 44 45 46 47 48 49  0 51 52  0 54  0 56 57 58 59 60 61 62  0  0 65 66  0 68 69  0  0  0
  0  0  0  0  0  0  0  0  0 82 83  0  0 86 87 88 89 90 91 92 93 94 95  0  0  0 99100101  0103  0  0  0  0  0
109  0111112

REQUIREMENT    054 IS CONNECTED TO--
  0  0  0  4  5  0  7  0  0 11  0 13  0  0  0  0  0  0  0  0  0 22  0  0 25  0  0 28 29 30 31  0  0  0  0  0
  0 38 39 40  0  0  0 44 45 46 47  0  0 50  0 52 53  0 55  0 57  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0 80  0  0  0  0  0 86 87 88 89  0 91  0 93 94  0  0  0  0  0100  0  0  0  0  0  0  0  0
  01101111112
```

```
REQUIREMENT     055 IS CONNECTED TO--
  0   0   0   4   5   6   7   0   0   0  11  12   0   0   0   0   0   0   0   0   0   0   0   0  25   0   0  28  29   0   0  32   0   0   0   0
  0   0  39   0   0   0   0  44  45   0  47   0   0   0   0   0   0  54   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0  80   0  82   0   0   0  86  87  88  89   0   0   0   0   0   0   0   0   0   0 99100   0   0   0   0   0   0   0
  0110   0112

REQUIREMENT     056 IS CONNECTED TO--
  0   2   0   0   5   0   7   0   0   0   0  12   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 37   0   0   0   0  42   0  44  45   0   0   0   0   0   0   0  53   0   0   0   0  58  59   0   0   0   0  64  65  66   0  68  69   0   0  72
 73   0   0   0   0  78   0   0  81   0   0   0   0   0   0   0   0   0  91   0   0  94   0   0   0   0 0100   0   0   0   0   0   0   0   0
109   0   0   0

REQUIREMENT     057 IS CONNECTED TO--
  1   2   0   0   5   0   0   0   9   0   0   0  13  14   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  30   0   0   0   0   0   0
  0   0  39  40   0   0  43   0   0   0   0   0  49   0  51  52  53  54   0   0   0  58  59   0   0  62   0   0   0   0   0   0   0   0   0   0
  0   0   0   0  77   0   0   0   0  82   0   0   0  86  87  88   0  90   0  92  93  94   0  96  97   0 99100101   0103   0   0   0   0   0
  0110   0   0

REQUIREMENT     058 IS CONNECTED TO--
  1   2   0   0   0   0   7   0   9  10  11  12  13  14  15  16  17   0  19   0   0   0   0   0  25   0   0   0   0   0   0   0  33   0   0   0
  0  38  39   0  41  42   0   0   0  46   0   0   0   0   0   0  53   0   0  56  57   0   0   0   0   0   0  64   0  66  67   0  69   0   0  72
 73  74  75  76  77  78  79  80  81  82   0  84  85   0   0   0   0  90  91   0  93   0  95  96  97   0  99   0101   0   0104   0106107108
109   0   0112

REQUIREMENT     059 IS CONNECTED TO--
  0   2   3   4   5   6   7   8   9  10  11  12  13  14   0   0   0  18   0   0   0   0   0   0  25   0  27  28   0  30   0   0  33   0  35   0
 37   0  39  40  41  42  43  44  45  46  47  48  49   0  51  52  53   0   0  56  57   0   0  60  61   0   0  64  65  66   0  68  69   0  71  72
  0  74  75   0  77  78  79  80  81  82   0   0   0  86  87  88  89  90  91  92  93  94   0  96   0   0 99100101102103104105   0   0108
  0110111112

REQUIREMENT     060 IS CONNECTED TO--
  1   2   0   4   5   6   7   0   9   0  11  12   0   0   0   0   0   0   0   0   0   0   0   0  25  26  27  28  29  30   0   0  33   0  35   0
  0  38   0  40   0   0   0   0  45  46   0  48  49   0  51  52  53   0   0   0   0   0   0  59   0  61   0   0   0  65  66   0  68   0   0   0
  0   0   0   0   0   0   0   0   0  82   0   0   0  86  87  88  89  90  91  92   0  94  95  96  97   0  99   0101   0103   0   0   0   0   0
109   0111   0

REQUIREMENT     061 IS CONNECTED TO--
  0   0   0   0   5   0   7   0   0   0   0  12   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  27   0   0   0   0   0  33   0  35   0
  0  38   0   0   0   0   0   0  45  46   0  48  49  50  51   0  53   0   0   0   0   0  59  60   0   0   0   0   0   0   0  68   0  70   0   0
  0   0   0   0   0   0   0   0  81  82   0   0  85   0  87  88   0   0  91   0   0   0  95  96   0   0   0   0   0   0   0   0   0   0   0   0
109   0   0112

REQUIREMENT     062 IS CONNECTED TO--
  0   0   0   0   0   6   0   0   0   0   0  12   0   0   0   0   0  18  19   0   0  22   0   0  25   0   0   0   0   0   0  32  33   0   0   0
  0   0  39  40   0  42  43   0   0   0   0   0  49   0   0  52  53   0   0   0  57   0   0   0   0   0  63  64  65  66  67   0  69   0   0   0
  0  74  75  76   0   0   0  80   0  82   0   0  85   0   0   0  89  90  91   0   0   0   0  96   0  98   0   0   0   0   0   0   0   0   0   0
  0110   0   0

REQUIREMENT     063 IS CONNECTED TO--
  0   0   0   0   5   6   7   0   0   0   0   0  13  14  15   0  17  18  19  20   0   0  23  24   0  26  27  28   0   0   0  32  33   0  35   0
 37   0   0  40   0   0  43  44  45   0   0  48  49  50  51  52   0   0   0   0   0   0   0   0   0  62   0   0   0  66   0  68   0   0   0   0
  0   0   0   0  77   0   0  80   0  82   0   0   0   0   0   0   0   0  91  92   0   0  95  96  97  98   0   0   0   0   0 0105   0   0   0
      0   0112
```

FIGURE 33-8

```
REQUIREMENT     064 IS CONNECTED TO--
  0  0  0  0  0  6  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 33  0  0 36
 37 38  0  0  0 42  0  0 45  0  0  0  0  0  0  0  0  0  0 56  0 58 59  0  0 62  0  0 65 66 67 68 69 70 71 72
 73  0  0  0  0  0  0  0  0  0 83  0 85  0  0  0  0  0 91  0  0 94 95  0  0  0  0  0  0  0104105106107108
  0  0  0  0

REQUIREMENT     065 IS CONNECTED TO--
  1  2  3  4  5  6  7  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 27 28  0  0  0  0 33  0  0 36
 37 38 39 40 41  0  0  0 45  0  0 48  0  0 51 52 53  0  0 56  0  0 59 60  0 62  0 64  0 66  0  0 69 70 71  0
  0  0  0  0  0  0  0  0  0  0 83  0  0 86  0 88  0  0  0  0 93 94 95  0  0  0  0  0  0  0104105  0  0  0
109  0  0112

REQUIREMENT     066 IS CONNECTED TO--
  1  2  0  0  5  6  7  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 27  0  0  0  0  0 33 34  0 36
 37 38 39 40  0 42  0  0 45  0  0 48 49  0  0  0 53  0  0 56  0 58 59 60  0 62 63 64 65  0  0 68 69  0 71  0
  0  0  0  0  0  0  0  0  0  0 83  0 85 86  0 88  0  0  0  0 93 94 95  0  0  0 99  0  0  0  0104  0  0  0  0
109  0  0112

REQUIREMENT     067 IS CONNECTED TO--
  0  0  0  0  0  0  0  0  0  0  0 12  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0 41 42  0  0  0  0  0  0 49  0  0  0  0  0  0  0  0 58  0  0  0 62  0 64  0  0  0  0  0  0 71 72
 73 74 75 76 77 78 79  0 81 82  0  0 85  0  0  0  0  0  0  0 93  0  0 96  0  0  0100101  0  0  0  0  0  0  0
  0  0  0  0

REQUIREMENT     068 IS CONNECTED TO--
  0  2  0  0  5  0  7  0  0  0  0 12 13  0  0  0  0  0  0  0  0  0 23  0  0  0 27 28  0  0  0  0 33  0 35  0
 37  0  0  0  0  0  0  0  0  0  0 48  0  0  0  0 53  0  0 56  0  0 59 60 61  0 63 64  0 66  0  0 69 70 71 72
  0 74 75  0 77 78  0  0  0 82  0  0  0  0  0  0  0  0  0  0 93  0  0  0  0  0  0  0  0101  0  0104  0  0  0  0
  0  0  0  0

REQUIREMENT     069 IS CONNECTED TO--
  1  0  0  4  5  6  0  0  0  0  0 12  0  0  0  0  0  0  0  0  0  0  0  0  0 26  0  0  0  0  0  0 33  0  0  0
 37 38  0  0  0  0  0  0  0  0  0 48  0  0 51  0 53  0  0 56  0 58 59  0  0 62  0 64 65 66  0 68  0  0  0  0
 73  0  0  0 77 78  0 80  0 82  0  0  0  0  0  0  0  0 91  0  0 94  0 96  0  0  0  0101  0  0104  0  0  0  0
  0  0  0  0

REQUIREMENT     070 IS CONNECTED TO--
  0  2  0  0  0  6  7  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 26  0  0  0  0  0  0 33  0  0 36
 37  0  0 40  0  0  0  0  0  0  0  0  0 50  0 52  0  0  0  0  0  0  0  0 61  0  0 64 65  0  0 68  0  0  0  0
  0  0  0  0  0 78 79  0 81 82  0 84  0  0  0  0  0  0  0  0 93  0 95  0  0  0  0  0  0102  0  0  0  0  0108
109  0  0112

REQUIREMENT     071 IS CONNECTED TO--
  1  0  0  0  5  6  7  0  0  0  0  0  0  0  0  0  0 18 19  0  0  0  0  0  0 26 27 28  0  0  0  0 33 34  0 36
 37  0  0  0  0 42  0  0 45  0  0  0  0  0  0  0  0  0  0  0  0  0 59  0  0  0  0 64 65 66 67 68  0  0  0 72
 73  0 75 76  0  0  0 80  0 82  0  0 85  0  0 88  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
109  0  0  0

REQUIREMENT     072 IS CONNECTED TO--
  0  0  0  0  0  0  0  0  0  0  0 12  0  0  0  0  0  0  0  0  0  0  0  0  0 26 27  0  0  0  0  0 33  0  0 36
 37 38  0  0  0  0  0  0 45  0  0 48  0  0  0  0  0  0  0 56  0 58 59  0  0  0  0 64  0  0 67 68  0  0 71  0
 73 74 75 76 77 78 79 80  0 82  0  0  0  0  0  0  0  0  0  0  0  0 95  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0111  0
```

```
REQUIREMENT     073 IS CONNECTED TO--
  1  0  0  0  0  0  0  0  0  0  0 12  0  0  0  0  0  0  0  0  0  0  0  0  0 26 27  0  0  0  0  0 33  0  0  0
  0  0  0  0  0  0  0  0 45  0  0 48  0 50  0  0  0  0  0 56  0 58  0  0  0  0  0 64  0  0 67  0 69  0 71 72
  0 74 75 76 77 78  0  0 81 82  0  0  0  0  0  0  0  0  0  0  0  0 95  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0

REQUIREMENT     074 IS CONNECTED TO--
  0  0  0  0  0  0  0  0  0  0  0 12  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 33  0 35  0
  0 38  0  0 41 42  0  0  0  0  0 48  0  0 51  0  0  0  0  0  0 58 59  0  0 62  0  0  0  0 67 68  0  0 72
 73  0 75 76 77 78 79 80 81 82  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
109  0  0112

REQUIREMENT     075 IS CONNECTED TO--
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0 42  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 58 59  0  0 62  0  0  0  0 67 68  0  0 71 72
 73 74  0 76 77 78 79  0 81 82  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0104  0  0  0  0
  0  0  0  0

REQUIREMENT     076 IS CONNECTED TO--
  1  0  0  0  5  0  0  8  0  0  0  0 13 14  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 33  0 35  0
  0  0  0  0 41 42  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 58  0  0  0 62  0  0  0  0 67  0  0  0 71 72
 73 74 75  0 77 78 79  0 81 82  0  0  0  0  0  0  0  0  0  0  0  0  0 95  0 97  0 99100  0  0  0  0  0  0  0  0
109  0111  0

REQUIREMENT     077 IS CONNECTED TO--
  0  0  0  0  0  0  0  8  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 33  0  0  0
  0  0  0  0 41 42  0  0  0  0  0  0  0  0  0  0  0  0  0  0 57 58 59  0  0  0 63  0  0  0 67 68 69  0  0 72
 73 74 75 76  0 78 79  0 81 82  0  0 85  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0103  0  0  0107  0
  0  0  0112

REQUIREMENT     078 IS CONNECTED TO--
  0  0  0  0  0  0  0  8  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 33  0  0  0
  0  0  0  0 41 42  0  0  0  0  0 48  0  0  0  0  0  0  0 56  0 58 59  0  0  0  0  0  0  0 67 68 69 70  0 72
 73 74 75 76 77  0 79  0 81  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0

REQUIREMENT     079 IS CONNECTED TO--
  0  0  0  0  0  0  0  8  0  0  0  0  0  0  0  0  0  0  0  0 21  0  0  0  0  0  0 28 29 30  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 58 59  0  0  0  0  0  0  0 67  0  0 70  0 72
  0 74 75 76 77 78  0  0 81  0  0 84  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0

REQUIREMENT     080 IS CONNECTED TO--
  1  0  3  4  5  6  7  0  0  0  0 12  0  0  0  0  0 18  0 20 21  0  0  0  0 26 27  0  0 30  0 32 33 34  0  0
  0  0  0 40 41 42  0 44 45 46 47 48  0  0  0  0  0 54 55  0  0 58 59  0  0 62 63  0  0  0  0  0 69  0 71 72
  0 74  0  0  0  0  0  0 81 82 83  0  0  0 87 88 89  0 91  0 93 94  0 96 97  0 99100101102  0  0  0  0  0108
109110  0112

REQUIREMENT     081 IS CONNECTED TO--
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 26  0  0  0  0 31  0 33  0  0  0
 37 38  0  0 41 42  0  0  0  0  0  0 49  0 51  0  0  0  0 56  0 58 59  0 61  0  0  0  0  0 67  0  0 70  0  0
 73 74 75 76 77 78 79 80  0 82 83  0 85  0  0  0  0 90 91 92 93 94 95 96  0  0  0  0  0  0  0  0  0  0  0108
109  0111112
```

FIGURE 33-9

```
REQUIREMENT     082 IS CONNECTED TO--
  1  2  3  4  5  6  7  0  0  0  0 12 13 14  0  0 17 18 19 20  0  0 23 24  0  0 27 28  0  0  0 32 33  0  0 36
 37 38 39 40  0 42 43 44 45 46 47 48 49 50  0 52 53  0 55  0 57 58 59 60 61 62 63  0  0  0 67 68 69 70 71 72
 73 74 75 76 77  0  0 80 81  0 83  0 85 86  0  0 89 90 91 92 93 94 95 96 97 98  0  0101  0103  0  0  0  0  0
109110111112

REQUIREMENT     083 IS CONNECTED TO--
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 33  0  0 36
 37 38  0  0  0  0 43  0  0  0  0  0 49  0  0  0 53  0  0  0  0  0  0  0  0  0  0 64 65 66  0  0  0  0  0  0
  0  0  0  0  0  0  0 80 81 82  0  0  0  0 87  0  0 90 91 92  0  0  0 96  0  0  0  0  0  0  0  0  0  0  0  0
109  0111112

REQUIREMENT     084 IS CONNECTED TO--
  0  0  3  4  0  6  0  0  0  0  0  0  0  0 15 16 17 18 19 20 21  0  0  0 25 26  0 28 29  0 31  0  0  0  0  0
  0  0  0  0  0  0  0  0  0 46  0  0 49  0  0  0  0  0  0  0  0 58  0  0  0  0  0  0  0  0  0  0  0 70  0  0
  0  0  0  0  0  0 79  0  0  0  0  0  0 86 87 88 89  0  0  0  0 94  0 96  0  0  0  0  0  0  0  0  0  0107  0
 0110  0112

REQUIREMENT     085 IS CONNECTED TO--
  1  2  3  4  0  0  0  0  9 10 11  0  0  0  0  0 17  0 19  0  0  0  0  0 25 26  0  0  0 30  0  0  0  0  0  0
  0  0 39  0  0  0  0  0  0 46 47  0  0  0  0  0  0  0  0  0  0 58  0  0 61 62  0 64  0 66 67  0  0  0 71  0
  0  0  0  0 77  0  0  0 81 82  0  0  0 86 87 88 89 90  0  0 93 94  0  0  0  0 99100  0  0  0  0  0  0  0  0
109110  0112

REQUIREMENT     086 IS CONNECTED TO--
  0  0  3  4  0  0  7  8  9 10 11  0  0  0  0  0  0 18  0  0  0  0  0  0 25  0 27  0 29  0 31 32 33  0  0  0
 37 38 39 40 41  0 43 44 45 46 47 48 49  0  0 52 53 54 55  0 57  0 59 60  0  0  0  0 65 66  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0 82  0 84 85  0 87 88 89 90 91  0 93 94  0 96  0  0 99100101  0103104105106107108
109110111112

REQUIREMENT     087 IS CONNECTED TO--
  1  2  3  4  0  0  7  8  0 10 11  0  0  0  0  0  0 18  0  0 21  0  0  0 25  0  0  0  0  0 31 32  0  0  0  0
  0 38 39 40 41  0 43  0 45 46 47  0 49  0  0 52 53 54 55  0 57  0 59 60 61  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0 80  0  0 83 84 85 86  0 88 89 90 91  0 93 94 95 96  0  0  0100101  0103104105106107108
 0110111112

REQUIREMENT     088 IS CONNECTED TO--
  1  0  0  4  5  6  7  0  0  0  0 12 13 14 15  0  0  0  0  0  0  0  0  0 25  0 27 28 29  0  0  0 33 34 35  0
 37 38 39 40  0  0 43  0 45 46 47 48 49 50 51 52 53 54 55  0 57  0 59 60 61  0  0  0 65 66  0  0  0  0 71  0
  0  0  0  0  0  0  0 80  0  0  0 84 85 86 87  0 89 90 91  0  0 94 95  0 97 98 99100101  0103104  0  0  0  0
109  0111112

REQUIREMENT     089 IS CONNECTED TO--
  0  0  3  4  0  0  7  8  9 10 11  0 13 14  0  0  0  0  0  0 21 22  0  0 25  0  0  0  0 30 31  0  0  0 35  0
 37 38 39 40 41  0 43 44 45 46 47 48 49 50  0 52 53 54 55  0  0  0 59 60  0 62  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0 80  0 82  0 84 85 86 87 88  0 90 91  0 93 94 95  0  0 98 99100  0102103  0105  0  0  0
 0110  0  0

REQUIREMENT     090 IS CONNECTED TO--
  0  2  3  4  0  6  0  0  9 10 11 12 13 14 15  0 17 18 19  0  0  0  0  0 25  0 27 28  0  0  0  0 33 34 35  0
 37 38 39 40  0  0  0 44 45 46  0 48 49 50 51 52 53  0  0  0 57 58 59 60  0 62  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0 81 82 83  0 85 86 87 88 89  0 91 92 93 94  0 96  0  0  0  0101  0103  0  0  0  0  0
109110111112
```

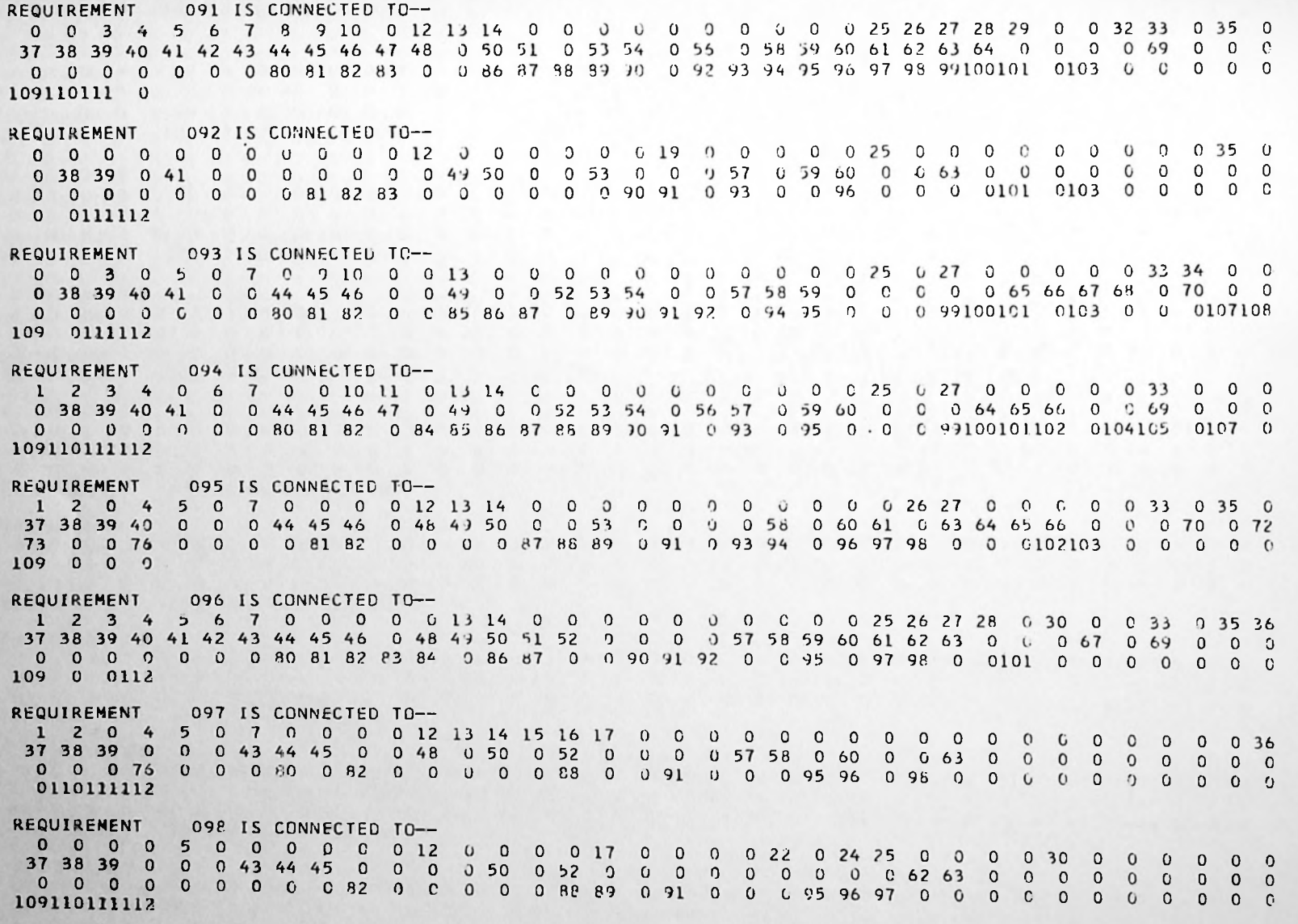

REQUIREMENT 091 IS CONNECTED TO--
0 0 3 4 5 6 7 8 9 10 0 12 13 14 0 0 0 0 0 0 0 0 0 0 25 26 27 28 29 0 0 32 33 0 35 0
37 38 39 40 41 42 43 44 45 46 47 48 0 50 51 0 53 54 0 56 0 58 59 60 61 62 63 64 0 0 0 0 69 0 0 0
0 0 0 0 0 0 0 80 81 82 83 0 0 86 87 98 89 90 0 92 93 94 95 96 97 98 99100101 0103 0 0 0 0 0
109110111 0
REQUIREMENT 092 IS CONNECTED TO--
0 0 0 0 0 0 0 0 0 0 0 12 0 0 0 0 0 0 19 0 0 0 0 0 25 0 0 0 0 0 0 0 0 0 35 0
0 38 39 0 41 0 0 0 0 0 0 0 49 50 0 0 53 0 0 0 57 0 59 60 0 0 63 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 81 82 83 0 0 0 0 0 0 90 91 0 93 0 0 96 0 0 0 0101 0103 0 0 0 0 0
0 0111112
REQUIREMENT 093 IS CONNECTED TO--
0 0 3 0 5 0 7 0 9 10 0 0 13 0 0 0 0 0 0 0 0 0 0 0 25 0 27 0 0 0 0 0 33 34 0 0
0 38 39 40 41 0 0 44 45 46 0 0 49 0 0 52 53 54 0 0 57 58 59 0 0 0 0 0 65 66 67 68 0 70 0 0
0 0 0 0 0 0 0 80 81 82 0 0 85 86 87 0 89 90 91 92 0 94 95 0 0 0 99100101 0103 0 0 0107108
109 0111112
REQUIREMENT 094 IS CONNECTED TO--
1 2 3 4 0 6 7 0 0 10 11 0 13 14 0 0 0 0 0 0 0 0 0 0 25 0 27 0 0 0 0 0 33 0 0 0
0 38 39 40 41 0 0 44 45 46 47 0 49 0 0 52 53 54 0 56 57 0 59 60 0 0 0 64 65 66 0 0 69 0 0 0
0 0 0 0 0 0 0 80 81 82 0 84 85 86 87 88 89 90 91 0 93 0 95 0 0 0 99100101102 0104105 0107 0
109110111112
REQUIREMENT 095 IS CONNECTED TO--
1 2 0 4 5 0 7 0 0 0 0 12 13 14 0 0 0 0 0 0 0 0 0 0 0 26 27 0 0 0 0 0 33 0 35 0
37 38 39 40 0 0 0 44 45 46 0 48 49 50 0 0 53 0 0 0 0 58 0 60 61 0 63 64 65 66 0 0 0 70 0 72
73 0 0 76 0 0 0 0 81 82 0 0 0 0 87 88 89 0 91 0 93 94 0 96 97 98 0 0 0102103 0 0 0 0 0
109 0 0 0
REQUIREMENT 096 IS CONNECTED TO--
1 2 3 4 5 6 7 0 0 0 0 13 14 0 0 0 0 0 0 0 0 0 0 25 26 27 28 0 30 0 0 33 0 35 36
37 38 39 40 41 42 43 44 45 46 0 48 49 50 51 52 0 0 0 0 57 58 59 60 61 62 63 0 0 0 67 0 69 0 0 0
0 0 0 0 0 0 0 80 81 82 83 84 0 86 87 0 0 90 91 92 0 0 95 0 97 98 0 0101 0 0 0 0 0 0 0
109 0 0112
REQUIREMENT 097 IS CONNECTED TO--
1 2 0 4 5 0 7 0 0 0 0 12 13 14 15 16 17 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 36
37 38 39 0 0 0 43 44 45 0 0 48 0 50 0 52 0 0 0 0 57 58 0 60 0 0 63 0 0 0 0 0 0 0 0 0
0 0 0 76 0 0 0 80 0 82 0 0 0 0 0 88 0 0 91 0 0 0 95 96 0 98 0 0 0 0 0 0 0 0 0 0
0110111112
REQUIREMENT 098 IS CONNECTED TO--
0 0 0 0 5 0 0 0 0 0 0 12 0 0 0 0 17 0 0 0 0 22 0 24 25 0 0 0 0 30 0 0 0 0 0 0
37 38 39 0 0 0 43 44 45 0 0 0 0 50 0 52 0 0 0 0 0 0 0 0 0 0 62 63 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 82 0 0 0 0 0 88 89 0 91 0 0 0 95 96 97 0 0 0 0 0 0 0 0 0 0 0
109110111112
REQUIREMENT 099 IS CONNECTED TO--
0 0 3 0 0 0 0 0 9 10 0 0 0 0 0 0 0 0 0 0 21 0 0 0 25 0 0 0 0 0 0 0 0 0 0 0
0 38 39 40 0 0 0 0 0 46 47 0 0 50 0 0 53 0 55 0 57 58 59 60 0 0 0 0 0 66 0 0 0 0 0 0
0 0 0 76 0 0 0 80 0 0 0 0 85 86 0 88 89 0 91 0 93 94 0 0 0 0 0100101102103 0105 0 0 0
0110 0112

FIGURE 33 - 11

```
REQUIREMENT    100 IS CONNECTED TO--
  0  2  0  0  0  0  0  8  9  0 11  0  0  0  0  0  0  0  0  0  0  0  0  0 25  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0 44  0 46 47  0  0  0  0  0 53 54 55 56 57  0 59  0  0  0  0  0  0  0 67  0  0  0  0  0
  0  0  0 76  0  0  0 80  0  0  0  0 85 86 87 88 89  0 91  0 93 94  0  0  0  0 99  0101102103  0105  0  0  0
  0110  0  0

REQUIREMENT    101 IS CONNECTED TO--
  0  0  3  0  0  0  0  0  9 10  0  0  0  0 15  0  0  0  0  0  0  0  0  0 25  0  0  0  0  0  0  0  0  0  0
  0 38 39 40  0  0  0 44  0 46 47 48  0 50  0 52 53  0  0  0 57 58 59 60  0  0  0  0  0  0 67 68 69  0  0  0
  0  0  0  0  0  0  0 80  0 82  0  0  0 86 87 88  0 90 91 92 93 94  0 96  0  0 99100  0102103104105  0  0  0
  0110111112

REQUIREMENT    102 IS CONNECTED TO--
  0  2  3  4  0  6  0  0  0 10  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 33  0  0  0
  0  0  0  0 41 42  0  0  0  0 47  0  0 50  0  0  0  0  0  0  0  0 59  0  0  0  0  0  0  0  0  0 70  0  0
  0  0  0  0  0  0  0 80  0  0  0  0  0  0  0  0 89  0  0  0  0 94 95  0  0  0 99100101  0103  0105  0  0  0
109  0  0  0

REQUIREMENT    103 IS CONNECTED TO--
  0  0  3  0  0  0  0  0  9 10  0  0 13  0  0  0  0  0  0  0  0  0  0  0 25  0  0  0  0  0  0  0  0  0  0
  0 38 39  0  0  0  0 44 45  0  0  0  0 50  0  0 53  0  0  0 57  0 59 60  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0 77  0  0  0  0 82  0  0  0 86 87 88 89 90 91 92 93  0 95  0  0  0 99100101102  0  0  0  0  0  0
109110  0  0

REQUIREMENT    104 IS CONNECTED TO--
  0  0  0  4  0  6  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 27  0  0  0  0  0  0  0  0
  0 38  0  0  0  0  0  0 45  0  0  0  0  0  0  0  0  0  0  0  0 58 59  0  0  0  0 64 65 66  0 68 69  0  0  0
  0  0 75  0  0  0  0  0  0  0  0  0  0 86 87 88  0  0  0  0  0 94  0  0  0  0  0  0101  0  0  0105  0107  0
109  0  0  0

REQUIREMENT    105 IS CONNECTED TO--
  1  2  0  4  0  6  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 23 24 25  0  0  0  0  0  0  0  0  0
  0  0 39 40  0  0  0  0  0  0  0  0  0 50  0  0  0  0  0  0  0  0  0 59  0  0  0 63 64 65  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0 86 87  0 89  0  0  0  0  0 94  0  0  0  0 99100101102  0104  0  0  0108
  0  0  0  0

REQUIREMENT    106 IS CONNECTED TO--
  0  0  0  0  0  0  0  0  0  0  0  0  0  0 15  0  0  0  0  0  0  0  0  0  0 26  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 58  0  0  0  0  0 64  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0 86 87  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0108
109  0  0112

REQUIREMENT    107 IS CONNECTED TO--
  1  0  0  0  5  6  7  8  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 26  0 28  0  0  0  0  0  0  0
 37 38 39 40  0  0  0 44 45  0  0  0  0  0  0  0  0  0  0  0  0 58  0  0  0  0  0 64  0  0  0  0  0  0  0
  0  0  0  0 77  0  0  0  0  0  0 84  0 86 87  0  0  0  0  0 93 94  0  0  0  0  0  0  0  0104  0  0  0108
109  0  0  0

REQUIREMENT    108 IS CONNECTED TO--
  1  2  0  0  0  0  0  8  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 25  0 27  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0 44 45 46 47  0  0  0  0  0  0  0  0  0  0 58 59  0  0  0  0 64  0  0  0  0  0 70  0  0
  0  0  0  0  0  0  0 80 81  0  0  0  0 86 87  0  0  0  0  0 93  0  0  0  0  0  0  0  0  0  0 0105106107  0
109  0111  0
```

```
REQUIREMENT     109 IS CONNECTED TO--
  1  2  3  4  0  6  0  8  9 10 11 12  0  0  0  0  0  0  0  0  0 22 23 24  0  0  0  0  0 30 31 32  0  0  0  0
 37 38 39 40  0  0 43 44 45  0  0 48  0 50  0 52 53  0  0 56  0 58  0 60 61  0 63  0 65 66  0  0  0 70 71  0
  0 74  0 76  0  0  0 80 81 82 83  0 85 86  0 88  0 90 91  0 93 94 95 96  0 98  0  0  0102103104  0106107108
  0110111112

REQUIREMENT     110 IS CONNECTED TO--
  1  2  0  0  5  0  7  0  0  0  0  0 13 14  0  0  0  0  0 20  0 22 23  0 25  0  0  0  0 30 31 32  0  0  0  0
  0  0 39  0  0  0  0 44  0 46 47  0 49  0  0 52  0 54 55  0 57  0 59  0  0 62  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0 80  0 82  0 84 85 86 87  0 89 90 91  0  0 94  0  0 97 98 99100101  0103  0  0  0  0  0
109  0111112

REQUIREMENT     111 IS CONNECTED TO--
  0  0  0  4  0  0  0  0  9 10  0  0  0  0  0  0  0 18 19  0  0  0 23  0 25 26  0  0  0 30  0  0 33  0 35  0
 37 38  0 40  0  0 43 44  0  0  0  0 49  0  0 52 53 54  0  0  0  0 59 60  0  0  0  0  0  0  0  0  0  0  0 72
  0  0  0 76  0  0  0  0 81 82 83  0  0 86 87 88  0 90 91 92 93 94  0  0 97 98  0 0101  0  0  0  0  0 0108
109110  0112

REQUIREMENT     112 IS CONNECTED TO--
  0  0  0  0  0  6  0  0  9 10 11  0  0  0  0 16 17 18 19 20 21  0 23  0 25 26  0  0  0  0  0  0  0  0 35  0
 37 38  0 40  0  0 43 44 45 46  0 48 49 50 51 52 53 54 55  0  0 58 59  0 61  0 63  0 65 66  0  0  0 70  0  0
  0 74  0  0 77  0  0 80 81 82 83 84 85 86 87 88  0 90  0 92 93 94  0 96 97 98 99  0101  0  0  0 0106  0  0
109110111  0
```

FIGURE 33-13

```
  1  2  3  4  5  6  7  0  9 10 11 12 13 14  0  0  0  0  0  0  0  0  0  0 25  0 27  0  0 30  0 32 33  0 35  0
 37 38 39 40 41  0 43 44 45 46 47 48 49 50 51 52 53 54 55  0 57  0 59 60 61 62 63  0 65 66  0  0 69  0  0  0
  0  0  0  0  0  0  0 80  0 82 83  0 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99100101102103  0105  0  0  0
109110111112

  0  0  0  0  0  0  0  8  0  0  0  0  0  0 15 16 17 18 19 20 21 22 23 24  0 26  0 28 29  0 31  0  0 34  0 36
  0  0  0  0  0 42  0  0  0  0  0  0  0  0  0  0  0  0  0 56  0 58  0  0  0  0  0 64  0  0 67 68  0 70 71 72
 73 74 75 76 77 78 79  0 81  0  0 84  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0104  0106107108
  0  0  0  0

NEW LEVEL OF HIERARCHY
  1  2  0  4  5  6  7  0  0  0  0 12 13 14  0  0  0  0  0  0  0  0  0  0  0  0 27  0  0  0  0  0 33  0 35  0
 37 38 39 40  0  0 43 44 45  0  0 48 49 50 51 52 53  0  0  0  0  0 59 60 61 62 63  0 65 66  0  0 69  0  0  0
  0  0  0  0  0  0  0  0  0 82 83  0  0  0  0 88  0 90 91 92  0  0 95 96 97 98  0  0  0  0  0  0  0  0  0  0
109  0111112

  0  0  3  0  0  0  0  0  9 10 11  0  0  0  0  0  0  0  0  0  0  0  0  0 25  0  0  0  0 30  0 32  0  0  0  0
  0  0  0  0 41  0  0  0  0 46 47  0  0  0  0  0  0 54 55  0 57  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0 80  0  0  0  0 85 86 87  0 89  0  0  0 93 94  0  0  0  0 99100101102103  0105  0  0  0
  0110  0  0

  0  0  0  0  0  0  0  8  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 26  0  0  0  0  0  0  0  0 36
  0  0  0  0  0 42  0  0  0  0  0  0  0  0  0  0  0  0  0 56  0 58  0  0  0  0  0 64  0  0 67 68  0 70 71 72
 73 74 75 76 77 78 79  0 81  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0104  0106107108
  0  0  0  0

  0  0  0  0  0  0  0  0  0  0  0  0  0  0 15 16 17 18 19 20 21 22 23 24  0  0  0 28 29  0 31  0  0 34  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0 84  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0

NEW LEVEL OF HIERARCHY
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0 43  0  0  0  0  0  0  0  0 52  0  0  0  0  0  0  0  0  0 62 63  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0 83  0  0  0  0  0  0  0  0 92  0  0  0  0 97 98  0  0  0  0  0  0  0  0  0  0
  0  0111112

  1  2  0  4  5  6  7  0  0  0  0 12 13 14  0  0  0  0  0  0  0  0  0  0  0  0 27  0  0  0  0  0 33  0 35  0
 37 38 39 40  0  0  0 44 45  0  0 48 49 50 51  0 53  0  0  0  0  0 59 60 61  0  0  0 65 66  0  0 69  0  0  0
  0  0  0  0  0  0  0  0  0 82  0  0  0  0  0 88  0 90 91  0  0  0 95 96  0  0  0  0  0  0  0  0  0  0  0  0
109  0  0  0

 CONTROL PASSED TO SCTRL
  0  0  0  0  0 30 32 41  0  0  0 55  0 80  0  0  0  0  0  0  0  0  0  0102  0105  0  0  0  0  0  0  0  0  0  0
  3  9 10 11 25  0  0  0 46 47 54  0 57  0 85 86 87 89 93 94 99100101  0103  0110  0  0  0  0  0  0  0  0  0  0
```

FIGURE 34, GRAPH B: OUTPUT - THE TREE

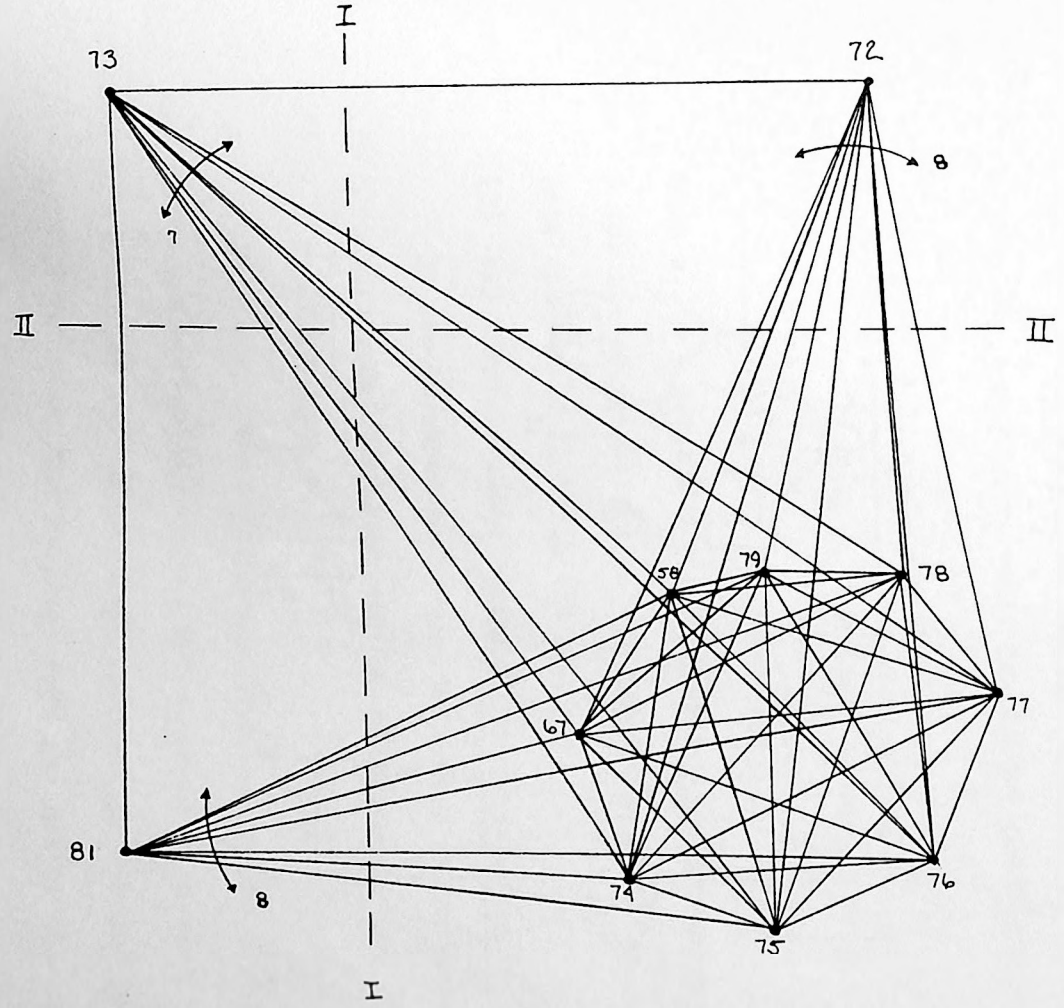
```
NEW LEVEL OF HIERARCHY
 0  0  0  0  0  0  0 41  0  0  0  0  0  0  0  0  0  0  0  0  0  0 0102  0105  0  0  0  0  0  0  0  0  0
 0  0  0  0  0 30 32  0  0  0  0 55  0 80  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 3  9 10  0  0  0  0  0  0  0  0  0 57  0  0  0  0  0  0 93  0 99  0101  0103  0  0  0  0  0  0  0  0  0
 0  0  0 11 25  0  0  0 46 47 54  0  0  0 85 86 87 89  0 94  0100  0  0  0  0110  0  0  0  0  0  0  0  0

NEW LEVEL OF HIERARCHY
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 0102  0105  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0 41  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0 55  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0 30 32  0  0  0  0  0  0 80  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 3  9 10  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 93  0 99  0101  0103  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0 57  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0 25  0  0  0 46 47 54  0  0  0  0 86 87 89  0 94  0100  0  0  0  0110  0  0  0  0  0  0  0  0
 0  0  0 11  0  0  0  0  0  0  0  0  0  0 85  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

NEW LEVEL OF HIERARCHY
CONTROL RETURNED TO LCTRL


CONTROL PASSED TO SCTRL
 0  0  0 42 56 58  0 67 68  0 71 72 73 74 75 76 77 78 79 81  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 8 26 36  0  0  0 64  0  0 70  0  0  0  0  0  0  0  0  0  0104106107108  0  0  0  0  0  0  0  0  0  0  0  0

NEW LEVEL OF HIERARCHY
 0  0  0  0 56  0  0  0 68  0 71  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0 42  0 58  0 67  0  0  0 72 73 74 75 76 77 78 79 81  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0 26 36  0  0  0 64  0  0 70  0  0  0  0  0  0  0  0  0  0  0106  0108  0  0  0  0  0  0  0  0  0  0  0
 8  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0104  0107  0  0  0  0  0  0  0  0  0  0  0

NEW LEVEL OF HIERARCHY
 0  0  0  0  0  0  0  0 68  0 71  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0 56  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0 58  0 67  0  0  0 72 73 74 75 76 77 78 79 81  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0 42  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0 64  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0106  0108  0  0  0  0  0  0  0  0  0  0
 0 26 36  0  0  0  0  0  0 70  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0104  0107  0  0  0  0  0  0  0  0  0  0  0
 8  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

NEW LEVEL OF HIERARCHY
 0  0  0  0  0  0  0  0 73  0  0  0  0  0  0 81  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0 58  0 67  0  0  0 72  0 74 75 76 77 78 79  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

NEW LEVEL OF HIERARCHY
CONTROL RETURNED TO LCTRL


CONTROL PASSED TO SCTRL
 0  0  0  0  0  0  0 22 23 24 28 29 31  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
15 16 17 18 19 20 21  0  0  0  0  0  0 34 84  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

```
NEW LEVEL OF HIERARCHY
 0  0  0  0  0  0  0  0  0 28  0 31  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0 22 23 24  0 29  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0 34  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
15 16 17 18 19 20 21  0  0  0  0  0  0  0 84  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

NEW LEVEL OF HIERARCHY
15  0  0  0  0  0 21  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0 16 17 18 19 20  0  0  0  0  0  0  0  0 84  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

NEW LEVEL OF HIERARCHY
CONTROL RETURNED TO LCTRL

NEW LEVEL OF HIERARCHY


CONTROL PASSED TO SCTRL
 0  0  0  0 83 92  0  0111112  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
43 52 62 63  0  0 97 98  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

NEW LEVEL OF HIERARCHY
43 52 62 63  0  0  0 98  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0 97  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

NEW LEVEL OF HIERARCHY
CONTROL RETURNED TO LCTRL


CONTROL PASSED TO SCTRL
 1  2  4  5  6  7 12 13 14  0  0 35 37 38 39  0 44 45 48  0 50  0  0  0  0  0  0  0  0 82  0 90 91 95 96109
 0  0  0  0  0  0  0  0  0 27 33  0  0  0  0 40  0  0  0 49  0 51 53 59 60 61 65 66 69  0 88  0  0  0  0  0

NEW LEVEL OF HIERARCHY
 0  0  4  0  6  0  0 13 14  0  0  0  0 38 39  0 44 45 48  0  0  0  0  0  0  0  0  0  0 82  0 90 91  0 96109
 1  2  0  5  0  7 12  0  0  0  0 35 37  0  0  0  0  0  0  0 50  0  0  0  0  0  0  0  0  0  0  0  0 95  0  0
 0  0  0  0  0  0  0  0  0  0 33  0  0  0  0  0  0  0  0 49  0 51  0  0  0 61  0  0 69  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0 27  0  0  0  0  0 40  0  0  0  0  0  0 53 59 60  0 65 66  0  0 88  0  0  0  0  0

NEW LEVEL OF HIERARCHY
 0  0  4  0  6  0  0 13 14  0  0  0  0 38 39  0 44 45 48  0  0  0  0  0  0  0  0  0  0 82  0 90 91  0 96  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 0109
 0  0  0  0  0  0  0  0  0  0 33  0  0  0  0  0  0  0  0 49  0 51  0  0  0 61  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 69  0  0  0  0  0  0  0  0

NEW LEVEL OF HIERARCHY
CONTROL RETURNED TO LCTRL
```

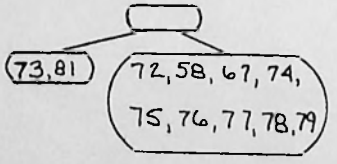
FIGURE 34 - 3

FIGURE 35, GRAPH B: THE TREE

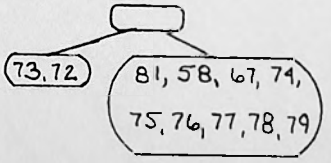
FIG. 36. A SUBGRAPH WITH TWO EQUAL PARTITIONS



RUN I



RUN II



RESOLUTION

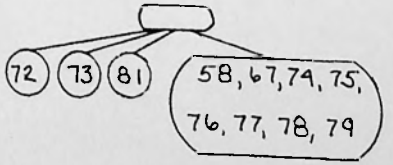
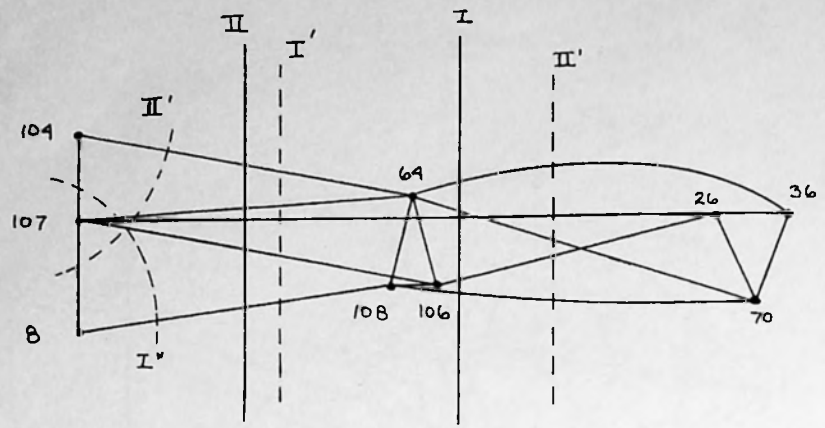
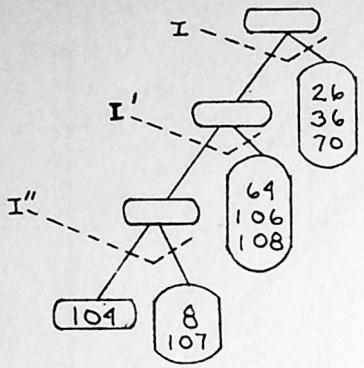


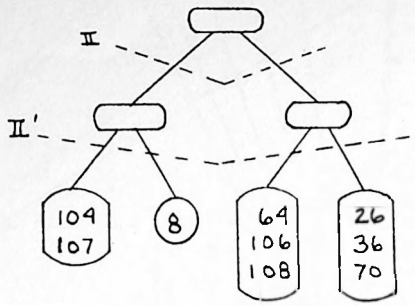
FIG. 37. A SECOND SUBGRAPH WITH EQUAL PARTITIONS



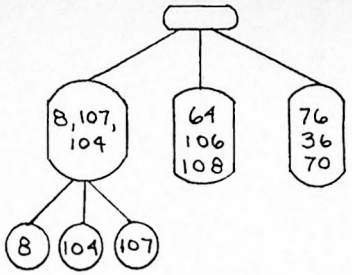
RUN I



RUN II



RESOLUTION



D36

E. THE INFORMATION THEORETIC CRITERION

The decomposition of a set of n vertices (NBIT = n) is based on the following assumptions:

(1) Each vertex has associated with it a stochastic binary variable, with $p(x_i = 0) = p(x_i = 1) = \frac{1}{2}$, i = 1,2,...n.

(2) A link between two vertices of the graph indicates a fixed small correlation between the two variables x_i and x_j corresponding to those vertices: that is, if the correlation is e_{ij},

$$e_{ij} = \frac{\left[p(x_i = 0,\ x_j=0)\ p(x_i=1,\ x_j=1)\right] - \left[p(x_i=1,\ x_j=0)\ p(x_i=0, x_j=1)\right]}{\left[p(x_i = 0)\ p\ (x_i = 1)\ p(x_j = 0)\ p(x_j = 1)\right]^{\frac{1}{2}}}$$

$$= 4\left[p(00)\ p\ (11) - p(10)\ p\ (01)\right],$$

then $e_{ij} = e$ for all i,j for which a link in the graph exists and 0 for all other pairs i,j, and $(e)(TOTAL) < 1$, where TOTAL is the total number of links in the graph.

(3) All three-variable correlations are assumed to be zero.

It can be shown that these three conditions uniquely define a probability distribution over the 2^n states of the system of n = NBIT variables.*

To measure the information transfer across a partition between a subset of M vertices and a subset of N vertices (M + N = n = NBIT), we take the average information carried by the system of NBIT vertices, and subtract from the sum of the two average informations carried by the system of M vertices and the system of N vertices. This measure of the redundancy between the two

*Alexander, NOTES, Appendix 2.

subsets of variables can be shown to be $\frac{(RR)e}{2}$, a constant multiple of (RR), the number of links between the two subsets. In decomposing the set of vertices, we wish to find a partition for which this redundancy is minimal.

As the measure stands, however, it is biased toward strongly assymmetrical partitions, in which the product MN is small - e.g., where M is small and N large. We normalize the measure by subtracting the expected value of RR and dividing by the square root of its variance. The normalized redundancy is

$$STR = \frac{(RR) - \left(\frac{TOTAL}{NSQ1}\right)(MN)}{\left[(MN)(NSQ1 - MN)\right]^{\frac{1}{2}}}$$

where TOTAL is again the total number of links in the graph, and NSQ1 is the maximum possible number of links (the total number of distinct unordered vertex pairs) - i.e., $NSQ1 = \frac{(NBIT)(NBIT - 1)}{2}$.

To simplify computation, we square this function but preserve its sign, giving

$$INFO = \frac{\left[RR - \left(\frac{TOTAL}{NSQ1}\right) MN\right] \left|\left[RR - \left(\frac{TOTAL}{NSQ1}\right) MN\right]\right|}{MN\ (NSQ1 - MN)}$$

INFO is the measure used in the algorithms of SMRMN and LGRMN, in the version of HIDECS-2 operational in December, 1961. That partition of a set of vertices is best for which the value of INFO is minimal.

F. THE MAXIMUM NUMBER OF SUBGRAPHS OF A GRAPH OF GIVEN ORDER

Assume first that each successive partition divides the appropriate subgraph into two and only two partitions. This assumption will be shown below to be conservative.

Let N_m be the number of elements (the original graph and each of its respective subgraphs) in the tree, where the subscript m indicates that there are m nodes in the original graph. If the original graph is decomposed into segments of x nodes and y nodes ($x + y = m$), then the number of elements in the subtree at the head of which is the x-nodes graph, is N_x, and the other subtree, headed by y, has N_y elements. The number of elements in the total tree is therefore $N_m = N_x + N_y + 1$, where the one is added to account for the original graph. This recurrence relationship requires that $N_n = 2n - 1$, which holds for n = 1,2,3, for example, and therefore by the recurrence relationship must hold for any n:

$$N_m = N_x + N_y + 1 = (2x - 1) + (2y - 1) + 1 = 2(x + y) - 1 = 2m - 1$$

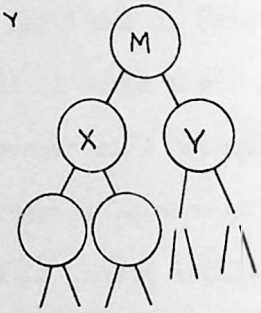
The assumption that each partition has only two subgraphs is conservative, because if it has more than two, say three, then the corresponding subtrees can be moved up into the tree, requiring correspondingly fewer elements. Cf. Figure 38, Trees and subtrees.

Figure 38

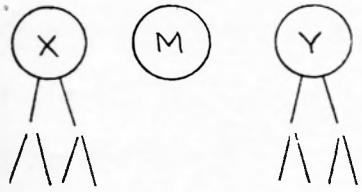
TREES AND SUBTREES

a) TWO - WAY PARTITIONS

$M = X + Y$



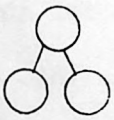
$N_M = N_X + N_Y + 1$



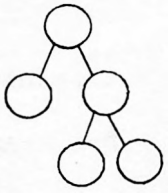
$N_M = 2M - 1$:

$M = 1,\ N_M = 1$

$M = 2,\ N_M = 3$

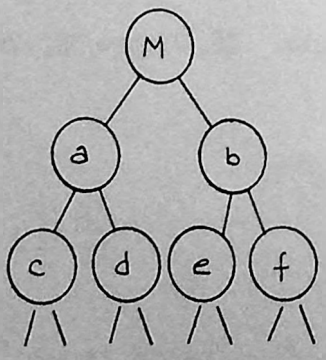


$M = 3,\ N_M = 5$



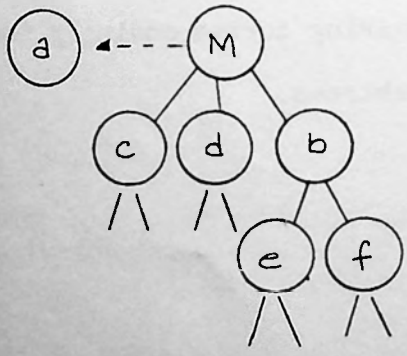
b) r - FOLD PARTITIONS

2 - FOLD : N_M

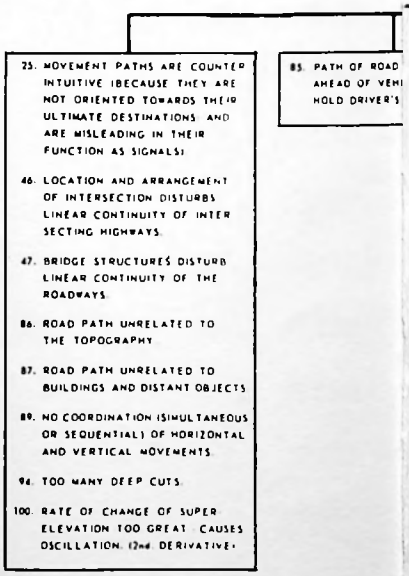


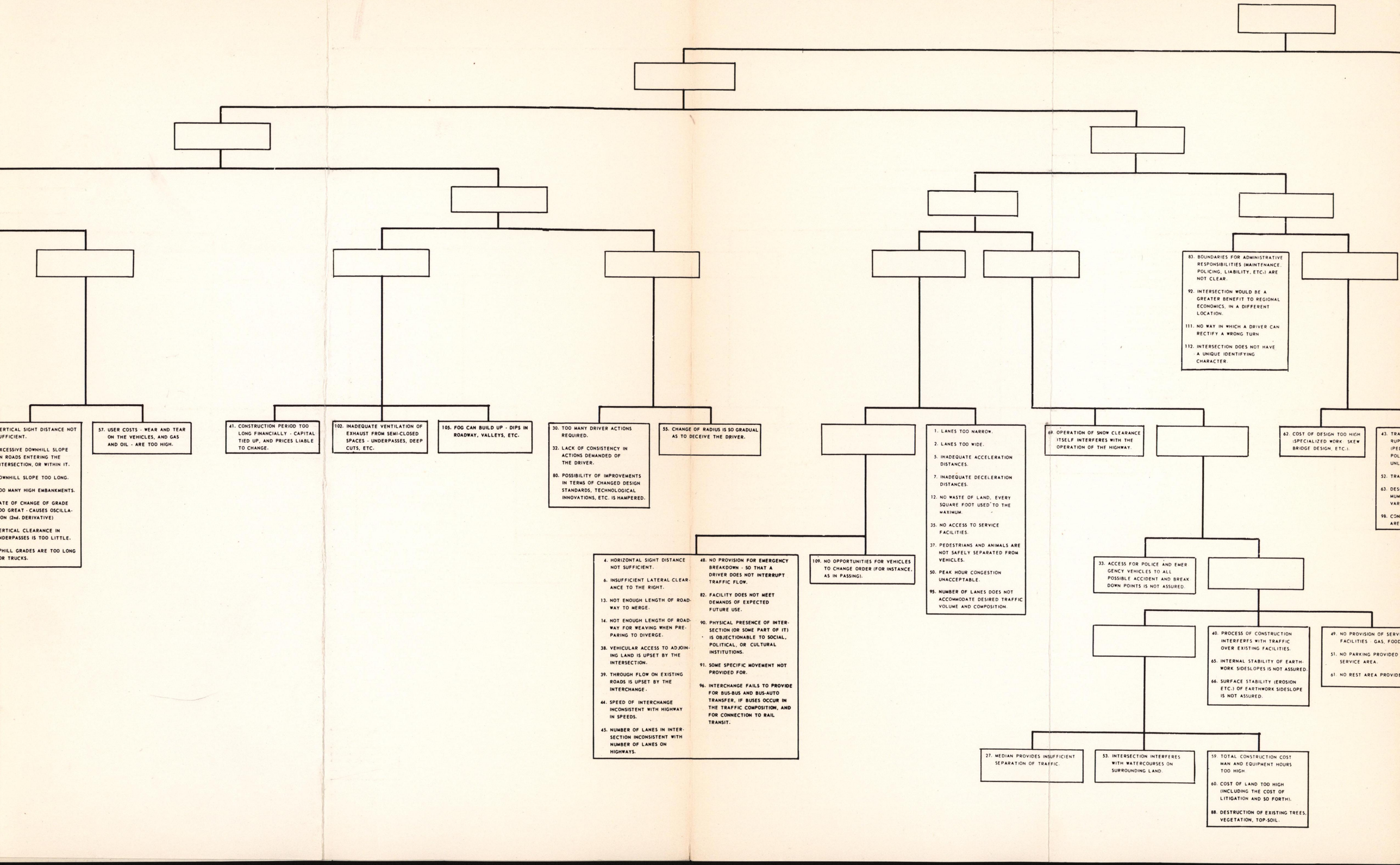
$N_M = N_a + N_b + 1 = N_c + N_d + N_b + 2$

3 - FOLD : N_M'



$N_M' = N_c + N_d + N_b + 1 < N_M$


25. MOVEMENT PATHS ARE COUNTER INTUITIVE (BECAUSE THEY ARE NOT ORIENTED TOWARDS THEIR ULTIMATE DESTINATIONS, AND ARE MISLEADING IN THEIR FUNCTION AS SIGNALS).
46. LOCATION AND ARRANGEMENT OF INTERSECTION DISTURBS LINEAR CONTINUITY OF INTER SECTING HIGHWAYS.
47. BRIDGE STRUCTURES DISTURB LINEAR CONTINUITY OF THE ROADWAYS.
86. ROAD PATH UNRELATED TO THE TOPOGRAPHY.
87. ROAD PATH UNRELATED TO BUILDINGS AND DISTANT OBJECTS.
89. NO COORDINATION (SIMULTANEOUS OR SEQUENTIAL) OF HORIZONTAL AND VERTICAL MOVEMENTS.
94. TOO MANY DEEP CUTS.
100. RATE OF CHANGE OF SUPER ELEVATION TOO GREAT. CAUSES OSCILLATION. (2nd DERIVATIVE).
85. PATH OF ROAD
AHEAD OF VEHI
HOLD DRIVER'S


ERTICAL SIGHT DISTANCE NOT UFFICIENT.
XCESSIVE DOWNHILL SLOPE N ROADS ENTERING THE NTERSECTION, OR WITHIN IT.
OWNHILL SLOPE TOO LONG.
OO MANY HIGH EMBANKMENTS.
ATE OF CHANGE OF GRADE OO GREAT - CAUSES OSCILLA- ON (2nd. DERIVATIVE)
ERTICAL CLEARANCE IN NDERPASSES IS TOO LITTLE.
PHILL GRADES ARE TOO LONG OR TRUCKS.
57. USER COSTS - WEAR AND TEAR ON THE VEHICLES, AND GAS AND OIL - ARE TOO HIGH.
41. CONSTRUCTION PERIOD TOO LONG FINANCIALLY - CAPITAL TIED UP, AND PRICES LIABLE TO CHANGE.
102. INADEQUATE VENTILATION OF EXHAUST FROM SEMI-CLOSED SPACES - UNDERPASSES, DEEP CUTS, ETC.
105. FOG CAN BUILD UP - DIPS IN ROADWAY, VALLEYS, ETC.
30. TOO MANY DRIVER ACTIONS REQUIRED.
32. LACK OF CONSISTENCY IN ACTIONS DEMANDED OF THE DRIVER.
80. POSSIBILITY OF IMPROVEMENTS IN TERMS OF CHANGED DESIGN STANDARDS, TECHNOLOGICAL INNOVATIONS, ETC. IS HAMPERED.
55. CHANGE OF RADIUS IS SO GRADUAL AS TO DECEIVE THE DRIVER.
4. HORIZONTAL SIGHT DISTANCE NOT SUFFICIENT.
6. INSUFFICIENT LATERAL CLEARANCE TO THE RIGHT.
13. NOT ENOUGH LENGTH OF ROADWAY TO MERGE.
14. NOT ENOUGH LENGTH OF ROADWAY FOR WEAVING WHEN PREPARING TO DIVERGE.
38. VEHICULAR ACCESS TO ADJOINING LAND IS UPSET BY THE INTERSECTION.
39. THROUGH FLOW ON EXISTING ROADS IS UPSET BY THE INTERCHANGE.
44. SPEED OF INTERCHANGE INCONSISTENT WITH HIGHWAY IN SPEEDS.
45. NUMBER OF LANES IN INTERSECTION INCONSISTENT WITH NUMBER OF LANES ON HIGHWAYS.
48. NO PROVISION FOR EMERGENCY BREAKDOWN - SO THAT A DRIVER DOES NOT INTERRUPT TRAFFIC FLOW.
82. FACILITY DOES NOT MEET DEMANDS OF EXPECTED FUTURE USE.
90. PHYSICAL PRESENCE OF INTERSECTION (OR SOME PART OF IT) IS OBJECTIONABLE TO SOCIAL, POLITICAL, OR CULTURAL INSTITUTIONS.
91. SOME SPECIFIC MOVEMENT NOT PROVIDED FOR.
96. INTERCHANGE FAILS TO PROVIDE FOR BUS-BUS AND BUS-AUTO TRANSFER, IF BUSES OCCUR IN THE TRAFFIC COMPOSITION, AND FOR CONNECTION TO RAIL TRANSIT.
109. NO OPPORTUNITIES FOR VEHICLES TO CHANGE ORDER (FOR INSTANCE, AS IN PASSING).
1. LANES TOO NARROW.
2. LANES TOO WIDE.
5. INADEQUATE ACCELERATION DISTANCES.
7. INADEQUATE DECELERATION DISTANCES.
12. NO WASTE OF LAND; EVERY SQUARE FOOT USED TO THE MAXIMUM.
35. NO ACCESS TO SERVICE FACILITIES.
37. PEDESTRIANS AND ANIMALS ARE NOT SAFELY SEPARATED FROM VEHICLES.
50. PEAK HOUR CONGESTION UNACCEPTABLE.
95. NUMBER OF LANES DOES NOT ACCOMMODATE DESIRED TRAFFIC VOLUME AND COMPOSITION.
83. BOUNDARIES FOR ADMINISTRATIVE RESPONSIBILITIES (MAINTENANCE, POLICING, LIABILITY, ETC.) ARE NOT CLEAR.
92. INTERSECTION WOULD BE A GREATER BENEFIT TO REGIONAL ECONOMICS, IN A DIFFERENT LOCATION.
111. NO WAY IN WHICH A DRIVER CAN RECTIFY A WRONG TURN
112. INTERSECTION DOES NOT HAVE A UNIQUE IDENTIFYING CHARACTER.
69. OPERATION OF SNOW CLEARANCE ITSELF INTERFERES WITH THE OPERATION OF THE HIGHWAY.
62. COST OF DESIGN TOO HIGH (SPECIALIZED WORK SKEW BRIDGE DESIGN, ETC.).
43. TRA
RUP
(PE
POL
UNL
52. TRA
63. DES
MUM
VAR
98. CON
ARE
33. ACCESS FOR POLICE AND EMERGENCY VEHICLES TO ALL POSSIBLE ACCIDENT AND BREAK DOWN POINTS IS NOT ASSURED.
40. PROCESS OF CONSTRUCTION INTERFERFS WITH TRAFFIC OVER EXISTING FACILITIES.
65. INTERNAL STABILITY OF EARTHWORK SIDESLOPES IS NOT ASSURED.
66. SURFACE STABILITY (EROSION ETC.) OF EARTHWORK SIDESLOPE IS NOT ASSURED.
49. NO PROVISION OF SERV
FACILITIES GAS, FOOD
51. NO PARKING PROVIDED
SERVICE AREA.
61. NO REST AREA PROVIDE
27. MEDIAN PROVIDES INSUFFICIENT SEPARATION OF TRAFFIC.
53. INTERSECTION INTERFERES WITH WATERCOURSES ON SURROUNDING LAND.
59. TOTAL CONSTRUCTION COST MAN AND EQUIPMENT HOURS TOO HIGH.
60. COST OF LAND TOO HIGH (INCLUDING THE COST OF LITIGATION AND SO FORTH).
88. DESTRUCTION OF EXISTING TREES, VEGETATION, TOP-SOIL.

28. EXCESSIVE HEADLIGHT GLARE FOR COMFORT AND SAFETY.

31. TOO FEW DRIVER ACTIONS REQUIRED - MONOTONY DANGEROUS.

22. DRIVER DOES NOT KNOW WHAT OTHER DRIVERS IN HIS OWN TRAFFIC STREAM ARE DOING. (MOVEMENT PATTERNS INADEQUATELY DEFINED AND CONTROLLED).

23. DRIVER DOES NOT KNOW WHAT COURSE OF ACTION HE SHOULD TAKE (MOVEMENT PATTERNS INADEQUATELY DEFINED AND CONTROLLED).

24. DRIVER DOES NOT KNOW WHAT DRIVERS IN OTHER TRAFFIC STREAMS ARE DOING. (MOVEMENT PATTERNS INADEQUATELY DEFINED AND CONTROLLED).

29. SUN IN YOUR EYES.

34. POWER LINE EMERGENCY TELEPHONE, ETC. CONNECTIONS ARE NOT ACCOMMODATED.

FLOW GETS INTER-
BY REGULAR EVENTS
IANS, TRAFFIC SIGNALS,
N, SWIVEL BRIDGES,
D ACCESS).

IME TOO GREAT.

DES NOT PERMIT MAXI-
OF FACILITY UNDER
TRAFFIC CONDITIONS.

ING MOVEMENTS
IBLE.

97. MANEUVER AREAS DO NOT ACCOMMODATE DESIRED VOLUME AT DESIGN SPEED.

15. VISIBILITY AT POINT OF ENTRY IS DANGEROUSLY LOW.

21. TOO MUCH STIMULATION (ADVERTISING, FLICKERING LIGHTS, VISUAL NOISE).

16. INSUFFICIENT ILLUMINATION.

17. LACK OF CONSISTENCY IN SIGNS.

18. DRIVERS DO NOT RECEIVE ADEQUATE INFORMATION AS TO HOW THEY SHOULD PROCEED TO DESTINATION.

19. TOO MUCH INFORMATION FOR THE DRIVER TO TAKE IN (NUMBER OF SIGNS OR INFORMATION PER SIGN).

20. DISTANCE BETWEEN SUCCESSIVE SIGNS TOO SMALL FOR ADEQUATE REACTIONS.

84. TOO FEW VISUAL STIMULI ALONG ROADSIDE TO AVOID MONOTONY.

68. COST (TO MAINTENANCE DEPARTMENT) OF CLEARING SNOW, VEGETATION, GARBAGE, ETC. IS TOO HIGH.

71. MAINTENANCE OF PAINT AND PAVEMENT MARKINGS (CHANNEL MARKINGS), VEGETATION CONTROL, ETC. SIGN CHANGES AND REPAINTING OBSTRUCTS TRAFFIC.

56. WATER FALLING ONTO ROADWAY CANNOT DRAIN.

42. DESIGN OF INTERCHANGE IS NOT STANDARDIZED AND SO DOES NOT ALLOW THE USE OF STANDARD OR PREFABRICATED COMPONENTS.

72. TIME FOR WHICH ROAD IS CLOSED FOR PAVEMENT RENEWAL IS EXCESSIVE.

73. PROCESS OF PAVEMENT RENEWAL ITSELF INTERFERES TOO MUCH WITH OPERATION OF THE HIGHWAY.

81. EXPECTED LIFE IS TOO LONG FOR ITS ROLE IN REGIONAL DEVELOPMENT.

58. TOTAL COST OF MATERIAL TOO HIGH.

67. COST OF REPLACEMENT AND RENEWAL (PAVEMENT MATERIALS), ETC.) IS TOO HIGH.

74. ORIGINAL LOAD-CARRYING CAPACITY TOO LOW.

75. WEAR AND TEAR ON STRUCTURE DUE TO WEATHERING (HEAT, WATER, ICE, COLD, WIND, ETC.) REDUCES ORIGINAL LOAD-CARRYING CAPACITY TOO QUICKLY.

76. LOADS CARRIED THEMSELVES REDUCE ORIGINAL LOAD-CARRYING CAPACITY TOO QUICKLY.

77. SURFACE TOO ROUGH FOR COMFORT.

78. SURFACE SUCH THAT IT RETAINS AN EXCESSIVE FILM OF WATER.

79. MATERIAL OF SURFACE TOO HIGHLY REFLECTANT - CAUSES SUN AND HEADLIGHT GLARE.

64. WIND, WATER, AND GRAVITY-BORNE DEBRIS CAN GET ONTO THE ROADWAY.

106. WIND-BORNE SMOKE CAN INTERFERE WITH DRIVER VISION.

108. HIGH CROSS WINDS A NUISANCE.

26. MEDIAN ACTS AS LATERAL OBSTRUCTION (PSYCHOLOGICAL CROWDING).

36. PATHS FOR PEDESTRIAN MOVEMENT THROUGH INTERCHANGE AREA ARE NOT DEFINED.

70. SNOW REMOVAL IS TOO SLOW AND LEAVES HIGHWAY OBSTRUCTED FOR LONG PERIODS.

8. SUPERELEVATION, RADIUS, AND DESIGN SPEED AND MATERIAL ARE NOT CONSISTENT WITH SAFETY.

104. SNOW DRIFTS DANGEROUSLY ONTO ROADWAY.

107. SUDDEN CHANGES IN WIND PRESSURE ARE DANGEROUS - ESPECIALLY AT HIGH SPEED.

G. LISTINGS OF SUBPROGRAMS

1. INPAR
2. GENER
3. SET8, SET9, SET11 *
4. INDAT
5. CNDAT
6. SYMET
7. PTMAT
8. LCTRL
9. LGRMN
10. REDUC
11. SCTRL, SIMPL, ALPHA**, NSTOR**
12. SMRMN ***
13. NTRSC **
14. COUNT, CNVRT
15. PRTIN, PTLVL, PTOUT
16. PTLGR
17. PTCLR
18. PTSCT **

* SET8 is physically located in LCTRL, SET9 in LGRMN, and SET11 in REDUC.

** These subprograms, concerned with NTRSC and related functions, have not been incorporated into the version of HIDECS 2 operational in December, 1961.

*** The listing for SMRMN, designated as HIDECS 2A here, is slightly different from that described in the text and shown in the figures, due to discovery of a program error as this manuscript was going to the printer.

DATE OF THIS DECK 3/1/62. VERSION OPERATIONAL AS OF 12/1/61.

```
                00004          ENTRY   INPAR

  TRANSFER VECTOR
00000  746362303460      (TSH)
00001  745163453460      (RTN)

  LINKAGE DIRECTOR
00002  000000000000
00003  314547215160


00004 -0634 00 4 00021    INPAR SXD IR4,4
                         *THIS SEQUENCING READS IN  ORDER,RANDM, AND LATIS IN THAT ORDER
00005 -0500 00 0 00022          CAL TAPE4
00006  0074 00 4 00000          TSX $(TSH),4
00007  0 00000 0 00024          PZE INFMT
00010 -1 00000 0 00000          STR
00011 -0600 00 0 77461          STQ ORDER
00012 -1 00000 0 00000          STR
00013 -0600 00 0 77404          STQ RANDM
00014 -1 00000 0 00000          STR
00015 -0600 00 0 77455          STQ LATIS
00016  0074 00 4 00001          TSX $(RTN),4
00017 -0534 00 4 00021          LXD IR4,4
00020  0020 00 4 00001          TRA 1,4
00021  0 00000 0 00000    IR4
00022  0 00004 0 00000    TAPE4 PZE 0,0,4
00023  060067346060             BCI 1,60X)
00024  740146010273       INFMT BCI 1,(1012,
                         * COMMON BLOCK - REVISED  13 SEPTEMBER 1961
                  77462          COMMON -1
                  77462    INDIC COMMON 1
                  77461    ORDER COMMON 1
                  77460    NWORD COMMON 1
                  77457    DAT   COMMON 1
                  77456    LGTH  COMMON 1
                  77455    LATIS COMMON 1
                  77454    NBITH COMMON 1
                  77453    NBITL COMMON 1
                  77452    NBIT1 COMMON 1
                  77451    NBIT  COMMON 1
                  77450    NSQ1  COMMON 1
                  77447    OPRMN COMMON 1
                  77446    ATOMO COMMON 1
                  77445    ATOM  COMMON 1
                  77444    ONED  COMMON 1
                  77443    D36   COMMON 1
                  77442    ATOOX COMMON 10
                  77430    ATOX  COMMON 10
                  77416    SET   COMMON 10
                  77404    RANDM COMMON 10
                  77372    DIFF  COMMON 10
```

linkage

linkage

```
77360   CONVT COMMON 40
77310   DATA  COMMON 40
77240   MATA  COMMON 40
77170   UNIT  COMMON 40
77120   COMUN COMMON 40
77050   EQLS  COMMON 20
77024   SECTS COMMON 50
76742   MATAX COMMON 260
76336   DROWS COMMON 2100
72252   MROWS COMMON 2100
66166   INMAT COMMON 5400
53536   ATMS  COMMON 2000
47616   MACRO COMMON 7000
              END

00025 IS THE FIRST LOCATION NOT USED BY THIS PROGRAM


   NO ERROR IN ABOVE ASSEMBLY
* DATE AND TIME NOW      4/12 1739.8
```

```
                    00002          ENTRY   GENER

  LINKAGE DIRECTOR
00000   000000000000
00001   272545255160

00002   0020 00 0 00003   GENER TRA *+1
00003  -0634 00 4 00105         SXD IR4,4
                        *               GENERATE ONED,D36,UNIT,COMON
00004   0500 00 0 00107         CLA =01000000
00005   0622 00 0 77444         STD ONED
00006   0500 00 0 00110         CLA =044000000
00007   0622 00 0 77443         STD D36
00010   0774 00 1 00044   START AXT 36,1
00011  -0500 00 0 00106         CAL =1
00012   0602 00 1 77170   DEF1  SLW UNIT,1
00013   0760 00 0 00006         COM
00014   0602 00 1 77120         SLW COMUN,1
00015   0760 00 0 00006         COM
00016   0767 00 0 00001         ALS 1
00017   2 00001 1 00012         TIX DEF1,1,1
                        *               GENERATE STORAGE PARAMETERS (NWORD,DAT,LGTH) FROM ORDER
00020   0774 00 1 00000         AXT 0,1
00021   0500 00 0 77461         CLA ORDER
00022   0402 00 0 77443         SUB D36
00023  -0120 00 0 00026         TMI MINUS
00024   0100 00 0 00026         TZE MINUS
00025   1 00001 1 00022         TXI *-3,1,1
00026   0754 00 1 00000   MINUS PXA 0,1
00027   0760 00 0 00001         LBT
00030   0400 00 0 00106         ADD =1
00031   0400 00 0 00106         ADD =1
00032   0767 00 0 00022         ALS 18
00033   0622 00 0 77460         STD NWORD
00034   0500 00 0 77460         CLA NWORD
00035   0400 00 0 77444         ADD ONED
00036   0131 00 0 00000         XCA
00037   0200 00 0 77461         MPY ORDER
00040   0767 00 0 00021         ALS 17
00041   0622 00 0 77457         STD DAT
00042   0500 00 0 77460         CLA NWORD
00043   0400 00 0 77460         ADD NWORD
00044   0400 00 0 77460         ADD NWORD
00045   0131 00 0 00000         XCA
00046   0200 00 0 77461         MPY ORDER
00047   0767 00 0 00021         ALS 17
00050   0622 00 0 77456         STD LGTH
                        *               GENERATES RANDM BLOCK FROM SINGLE INPUT WORD, RANDM.
00051   0560 00 0 77404         LDQ RANDM
00052  -0534 00 2 77460         LXD NWORD,2
00053  -0773 00 0 00013         RQL 11
```

```
00054 -0600 00 2 77404          STQ RANDM,2
00055  2 00001 2 00053          TIX *-2,2,1
00056 -0600 00 0 77372          STQ DIFF
00057 -0534 00 2 77460          LXD NWORD,2
00060 -0773 00 0 00013          RQL 11
00061 -0600 00 2 77372          STQ DIFF,2
00062  2 00001 2 00060          TIX *-2,2,1
                        * GENERATES  INDIRECT ADDRESSING KEY (BEHIND MATAX)  - GENERZALIZED
00063  0500 00 0 00103          CLA MXM1
00064  0602 00 0 76741          SLW MATAX-1
00065  0500 00 0 77461          CLA ORDER
00066  0622 00 0 00100          STD LOC
00067  0500 00 0 77460          CLA NWORD
00070  0400 00 0 77444          ADD ONED
00071  0771 00 0 00022          ARS 18
00072  0601 00 0 00104          STO DIFSP
00073  0774 00 1 00001          AXT 1,1
00074  0500 00 1 76742          CLA MATAX,1
00075  0402 00 0 00104          SUB DIFSP
00076  1 00001 1 00077          TXI *+1,1,1
00077  0602 00 1 76742          SLW MATAX,1
00100 -3 00000 1 00075  LOC     TXL *-3,1,**
                        * END OF THIS BLOCK --(MUST COME AFTER DEFINITION OF ONED)
00101 -0534 00 4 00105          LXD IR4,4
00102  0020 00 4 00001          TRA 1,4
00103  000000272252     MXM1   VFD 18/0,03/2,15/MROWS
00104  0 00000 0 00000  DIFSP  PZE
00105  0 00000 0 00000  IR4
                        * COMMON BLOCK - REVISED  13 SEPTEMBER 1961
                 77462          COMMON -1
                 77462  INDIC   COMMON 1
                 77461  ORDER   COMMON 1
                 77460  NWORD   COMMON 1
                 77457  DAT     COMMON 1
                 77456  LGTH    COMMON 1
                 77455  LATIS   COMMON 1
                 77454  NBITH   COMMON 1
                 77453  NBITL   COMMON 1
                 77452  NBIT1   COMMON 1
                 77451  NBIT    COMMON 1
                 77450  NSQ1    COMMON 1
                 77447  OPRMN   COMMON 1
                 77446  ATOMO   COMMON 1
                 77445  ATOM    COMMON 1
                 77444  ONED    COMMON 1
                 77443  D36     COMMON 1
                 77442  ATOOX   COMMON 10
                 77430  ATOX    COMMON 10
                 77416  SET     COMMON 10
                 77404  RANDM   COMMON 10
                 77372  DIFF    COMMON 10
                 77360  CONVT   COMMON 40
                 77310  DATA    COMMON 40
```

```
77240    MATA   COMMON 40
77170    UNIT   COMMON 40
77120    COMUN  COMMON 40
77050    EQLS   COMMON 20
77024    SECTS  COMMON 50
76742    MATAX  COMMON 260
76336    DROWS  COMMON 2100
72252    MROWS  COMMON 2100
66166    INMAT  COMMON 5400
53536    ATMS   COMMON 2000
47616    MACRO  COMMON 7000
                END

  LITERALS
00106   000000000001
00107   000001000000
00110   000044000000
```

```
                  00004          ENTRY  INDAT

  TRANSFER VECTOR
00000  746362303460      (TSH)
00001  745163453460      (RTN)

  LINKAGE DIRECTOR
00002  000000000000
00003  314524216360


00004 -0634 00 4 00030    INDAT SXD IR4,4
00005  0500 00 0 77456          CLA LGTH
00006  0622 00 0 00027          STD LENTH
00007 -0500 00 0 00025          CAL TAPE4
00010  0074 00 4 00000          TSX $(TSH),4
00011  0 00000 0 00026          PZE INFMT
00012 -0500 00 0 00027          CAL LENTH
00013  0622 00 0 00020          STD TXL2
00014  0774 00 1 00001          AXT 1,1
00015 -1 00000 0 00000    LST2  STR
00016 -0600 00 1 66166          STQ INMAT,1
00017  1 00001 1 00020          TXI *+1,1,1
00020 -3 00000 1 00015    TXL2  TXL LST2,1,**
00021  0074 00 4 00001          TSX $(RTN),4
00022  1 00001 2 00023          TXI *+1,2,1
00023 -0534 00 4 00030          LXD IR4,4
00024  0020 00 4 00001          TRA 1,4
00025  0 00004 0 00000    TAPE4 PZE 0,0,4
00026  740646010234       INFMT BCI 1,(6012)
00027  0 00000 0 00000    LENTH PZE 0,0,0
00030  0 00000 0 00000    IR4   PZE
                         * COMMON BLOCK - REVISED  13 SEPTEMBER 1961
                  77462          COMMON -1
                  77462    INDIC COMMON 1
                  77461    ORDER COMMON 1
                  77460    NWORD COMMON 1
                  77457    DAT   COMMON 1
                  77456    LGTH  COMMON 1
                  77455    LATIS COMMON 1
                  77454    NBITH COMMON 1
                  77453    NBITL COMMON 1
                  77452    NBIT1 COMMON 1
                  77451    NBIT  COMMON 1
                  77450    NSQ1  COMMON 1
                  77447    OPRMN COMMON 1
                  77446    ATOMO COMMON 1
                  77445    ATOM  COMMON 1
                  77444    ONED  COMMON 1
                  77443    D36   COMMON 1
                  77442    ATOOX COMMON 10
                  77430    ATOX  COMMON 10
```

```
77416   SET    COMMON 10
77404   RANDM  COMMON 10
77372   DIFF   COMMON 10
77360   CONVT  COMMON 40
77310   DATA   COMMON 40
77240   MATA   COMMON 40
77170   UNIT   COMMON 40
77120   COMUN  COMMON 40
77050   EQLS   COMMON 20
77024   SECTS  COMMON 50
76742   MATAX  COMMON 260
76336   DROWS  COMMON 2100
72252   MROWS  COMMON 2100
66166   INMAT  COMMON 5400
53536   ATMS   COMMON 2000
47616   MACRO  COMMON 7000
               END

00031 IS THE FIRST LOCATION NOT USED BY THIS PROGRAM


   NO ERROR IN ABOVE ASSEMBLY
 * DATE AND TIME NOW      4/12 1739.9
```

```
                00010          ENTRY  CNDAT

  TRANSFER VECTOR
00000  234565516360      CNVRT

  LINKAGE DIRECTOR
00001  000000000000
00002  234524216360


                              *              THIS SUBPROGRAM CONVERTS THE MATRIX, AS STORED BEHIND
                              *              INMAT, FROM OCTAL TO BINARY FORMAT, BY COLLAPSING EACH
                              *              3 BINARY POSITIONS INTO ONE THROUGH REFERENCE TO
                              *              CNVRT TABLE.
00003   0 00000 0 00000    IR4
00004   0 00000 0 00000    HOLDR PZE
00005   0 00000 0 00000    SXRA
00006   0 00000 0 00000    SXRB
00007   0 00000 0 00000    SXRD
00010 -0634 00 4 00003     CNDAT SXD IR4,4
                              *                    PRELIMINARY OPERATION
00011   0500 00 0 77461           CLA ORDER
00012   0622 00 0 00056           STD ROWSS
00013   0500 00 0 77460           CLA NWORD
00014   0622 00 0 00054           STD TTT
                              *              THE WORDS OF INMAT ARE CALLED UP IN GROUPS OF THREE,
                              *              EACH WORD IS CONVERTED TO CONDENSED FORMAT BY REFERENCE
                              *              TO CNVRT, AND ADDED TO HOLDR. AT THE COMPLETION OF EACH
                              *              UNIT OF THREE 36-BIT BINARY WORDS, EACH REPRESENTING A
                              *              12-DIGIT OCTAL WORD, THE RESULT IN HOLDR IS A SINGLE
                              *              36-BIT BINARY WORD CORRESPONDING TO THE MATRIX BEFORE
                              *              INPUT.
00015   0774 00 1 00001           AXT 1,1
00016   0774 00 4 00001           AXT 1,4
00017   0774 00 2 00001    CAQ2   AXT 1,2
00020 -0500 00 0 00061     CAQ1   CAL =0
00021   0602 00 0 00004           SLW HOLDR
00022   0560 00 4 66166           LDQ INMAT,4
00023   0500 00 0 00061           CLA =0
00024   0522 60 0 00000           XEC* $CNVRT
00025 -0320 00 0 00062            ANA =0777700000000
00026   0361 00 0 00004           ACL HOLDR
00027   0602 00 0 00004           SLW HOLDR
00030   1 00001 4 00031           TXI *+1,4,1
00031   0560 00 4 66166           LDQ INMAT,4
00032   0500 00 0 00061           CLA =0
00033   0522 60 0 00000           XEC* $CNVRT
00034 -0320 00 0 00062            ANA =0777700000000
00035   0771 00 0 00014           ARS 12
00036   0361 00 0 00004           ACL HOLDR
00037   0602 00 0 00004           SLW HOLDR
00040   1 00001 4 00041           TXI *+1,4,1
```

```
00041   0560 00 4 66166          LDQ INMAT,4
00042   0500 00 0 00061          CLA =0
00043   0522 60 0 00000          XEC* $CNVRT
00044  -0320 00 0 00062          ANA =0777700000000.
00045   0771 00 0 00030          ARS 24
00046   0361 00 0 00004          ACL HOLDR
00047   0602 00 0 00004          SLW HOLDR
00050   1 00001 4 00051          TXI *+1,4,1
                        *                STORE THE RESULT IN ITS APPROPRIATE PLACE IN MROWS.
00051  -0500 00 0 00004          CAL HOLDR
00052   0602 60 1 76742          SLW* MATAX,1
00053   1 00001 2 00054          TXI *+1,2,1
00054  -3 00000 2 00020   TTT    TXL CAQ1,2,0
00055   1 00001 1 00056          TXI *+1,1,1
00056  -3 00000 1 00017   ROWSS  TXL CAQ2,1,0
00057  -0534 00 4 00003          LXD IR4,4
00060   0020 00 4 00001          TRA 1,4
                        * COMMON BLOCK - REVISED  13 SEPTEMBER 1961
                  77462          COMMON -1
                  77462   INDIC  COMMON 1
                  77461   ORDER  COMMON 1
                  77460   NWORD  COMMON 1
                  77457   DAT    COMMON 1
                  77456   LGTH   COMMON 1
                  77455   LATIS  COMMON 1
                  77454   NBITH  COMMON 1
                  77453   NBITL  COMMON 1
                  77452   NBIT1  COMMON 1
                  77451   NBIT   COMMON 1
                  77450   NSQ1   COMMON 1
                  77447   OPRMN  COMMON 1
                  77446   ATOMO  COMMON 1
                  77445   ATOM   COMMON 1
                  77444   ONED   COMMON 1
                  77443   D36    COMMON 1
                  77442   ATOOX  COMMON 10
                  77430   ATOX   COMMON 10
                  77416   SET    COMMON 10
                  77404   RANDM  COMMON 10
                  77372   DIFF   COMMON 10
                  77360   CONVT  COMMON 40
                  77310   DATA   COMMON 40
                  77240   MATA   COMMON 40
                  77170   UNIT   COMMON 40
                  77120   COMUN  COMMON 40
                  77050   EQLS   COMMON 20
                  77024   SECTS  COMMON 50
                  76742   MATAX  COMMON 260
                  76336   DROWS  COMMON 2100
                  72252   MROWS  COMMON 2100
                  66166   INMAT  COMMON 5400
                  53536   ATMS   COMMON 2000
                  47616   MACRO  COMMON 7000
                          END
```

```
  LITERALS
00061   000000000000
00062   777700000000

00063 IS THE FIRST LOCATION NOT USED BY THIS PROGRAM
```

```
                    00002          ENTRY  SYMET .

 LINKAGE DIRECTOR
00000  000000000000
00001  627044256360


00002 -0634 00 4 00106   SYMET SXD IR4,4
                       *              INITIALIZE
00003  0500 00 0 77444          CLA ONED
00004  0622 00 0 00105          STD IJ
00005  0622 00 0 00104          STD JI
00006 -0534 00 2 00105   DIACK LXD IJ,2
00007  0774 00 1 00044          AXT 36,1
00010 -0500 00 1 77120          CAL COMUN,1
00011  0320 60 1 76742   FF     ANS* MATAX,1
00012  2 00001 1 00010          TIX *-2,1,1
00013 -0534 00 2 00104   BEGIN LXD JI,2
00014  0774 00 4 00044   AA     AXT 36,4
00015  0560 60 4 76742   BB     LDQ* MATAX,4
00016  0774 00 1 00001          AXT 1,1
00017  0162 00 0 00030   TET    TQP ZO
00020 -0500 00 4 77170          CAL UNIT,4
00021 -0534 00 2 00105          LXD IJ,2
00022 -0320 60 1 76742   CC     ANA* MATAX,1
00023 -0534 00 2 00104          LXD JI,2
00024 -0100 00 0 00034          TNZ RQL
                       *                THIS BLOCK SELECTS THE INTERSECTION
00025 -0500 00 1 77120          CAL COMUN,1
00026  0320 60 4 76742   DD     ANS* MATAX,4
00027  0020 00 0 00034          TRA RQL
00030 -0500 00 4 77120   ZO     CAL COMUN,4
00031 -0534 00 2 00105          LXD IJ,2
00032  0320 60 1 76742   EE     ANS* MATAX,1
00033 -0534 00 2 00104          LXD JI,2
00034 -0773 00 0 00001   RQL    RQL 1
00035  1 00001 1 00036          TXI *+1,1,1
                       *                TEST A,B
00036 -3 00044 1 00017          TXL TET,1,36
00037  2 00001 4 00015          TIX TET-2,4,1
00040  0020 00 0 00041          TRA NTROL
                       *                TEST IJ
00041  0500 00 0 00105   NTROL CLA IJ
00042  0340 00 0 77460          CAS NWORD
00043  0761 00 0 00000          NOP
00044  0020 00 0 00055          TRA DIAG
00045  0400 00 0 77444          ADD ONED
00046  0622 00 0 00105          STD IJ
00047  0500 00 0 00015          CLA BB
00050  0760 00 0 00003          SSP
00051  0402 00 0 00107          SUB =36
00052  0621 00 0 00015          STA BB
```

```
00053  0621 00 0 00026          STA DD
00054  0020 00 0 00013          TRA BEGIN
                       *                TEST JI
00055  0500 00 0 00104   DIAG   CLA JI
00056  0340 00 0 77460          CAS NWORD
00057  0761 00 0 00000          NOP
00060  0020 00 0 00075          TRA FINIS
00061  0400 00 0 77444          ADD ONED
00062  0622 00 0 00104          STD JI
00063  0622 00 0 00105          STD IJ
00064  0500 00 0 00022          CLA CC
00065  0760 00 0 00003          SSP
00066  0402 00 0 00107          SUB =36
00067  0621 00 0 00015          STA BB
00070  0621 00 0 00022          STA CC
00071  0621 00 0 00026          STA DD
00072  0621 00 0 00032          STA EE
00073  0621 00 0 00011          STA FF
00074  0020 00 0 00006          TRA DIACK
00075  0020 00 0 00076   FINIS  TRA *+1
                       *                REPLACE DROWS WITH MODIFIED MATRIX FROM MROWS.
00076 -0534 00 4 77457          LXD DAT,4
00077 -0500 00 4 72252          CAL MROWS,4
00100  0602 00 4 76336          SLW DROWS,4
00101  2 00001 4 00077          TIX *-2,4,1
00102 -0534 00 4 00106          LXD IR4,4
00103  0020 00 4 00001          TRA 1,4
00104  0 00000 0 00000   JI
00105  0 00000 0 00000   IJ
00106  0 00000 0 00000   IR4    PZE
                        * COMMON BLOCK - REVISED  13 SEPTEMBER 1961
                  77462         COMMON -1
                  77462   INDIC COMMON 1
                  77461   ORDER COMMON 1
                  77460   NWORD COMMON 1
                  77457   DAT   COMMON 1
                  77456   LGTH  COMMON 1
                  77455   LATIS COMMON 1
                  77454   NBITH COMMON 1
                  77453   NBITL COMMON 1
                  77452   NBIT1 COMMON 1
                  77451   NBIT  COMMON 1
                  77450   NSQ1  COMMON 1
                  77447   OPRMN COMMON 1
                  77446   ATOMO COMMON 1
                  77445   ATOM  COMMON 1
                  77444   ONED  COMMON 1
                  77443   D36   COMMON 1
                  77442   ATOOX COMMON 10
                  77430   ATOX  COMMON 10
                  77416   SET   COMMON 10
                  77404   RANDM COMMON 10
                  77372   DIFF  COMMON 10
```

```
77360   CONVT COMMON 40
77310   DATA  COMMON 40
77240   MATA  COMMON 40
77170   UNIT  COMMON 40
77120   COMUN COMMON 40
77050   EQLS  COMMON 20
77024   SECTS COMMON 50
76742   MATAX COMMON 260
76336   DROWS COMMON 2100
72252   MROWS COMMON 2100
66166   INMAT COMMON 5400
53536   ATMS  COMMON 2000
47616   MACRO COMMON 7000
        END

  LITERALS
00107  000000000044

00110 IS THE FIRST LOCATION NOT USED BY THIS PROGRAM


   NO ERROR IN ABOVE ASSEMBLY
* DATE AND TIME NOW      4/12 1740.2
```

00004 ENTRY PTMAT

```
  TRANSFER VECTOR
00000  746263303460     (STH)
00001  742631433460     (FIL)

  LINKAGE DIRECTOR
00002  000000000000
00003  476344216360


                        * THIS PROGRAM READS OUT THE MATRIX. GOOD FOR ORDER UP TO 400. IF LARGER, CHNAGE
                        *      BES FOR TABLE AND ROWS, AND CHANGE HNDRD BLOCK.
00004 -0634 00 4 01005   PTMAT SXD IR4,4
                        *INITIALIZE CONTROL PARAMETERS
00005  0500 00 0 77461         CLA ORDER
00006  0622 00 0 00063         STD TXL1
00007  0622 00 0 00137         STD TXL2
00010  0622 00 0 00144         STD TXL4
00011  0622 00 0 00035         STD TXL5
00012  0500 00 0 77460         CLA NWORD
00013  0622 00 0 00126         STD TXL3
                        *TRANSFER MATRIX TO WORKING AREA
00014 -0534 00 4 77457         LXD DAT,4
00015 -0500 00 4 76336         CAL DROWS,4
00016  0602 00 4 72252         SLW MROWS,4
00017  2 00001 4 00015         TIX *-2,4,1
                        *GENERATE LIST OF ROW NUMBERS FOR HEADINGS IN PRINTOUT
00020  0600 00 0 01012         STZ DECML
00021  0500 00 0 77444         CLA ONED
00022  0622 00 0 01010         STD RGSTR
00023  0600 00 0 01011         STZ OTHER
00024  0774 00 1 00000         AXT 0,1    ROWS
00025  0774 00 4 00000         AXT 0,4    TENS
00026  0774 00 2 00000         AXT 0,2    UNITS
00027 -0500 00 2 01645   CYCLE CAL UNITS,2
00030  0400 00 0 01012         ADD DECML
00031  0400 00 0 01011         ADD OTHER
00032  0602 00 1 01633         SLW ROWS,1
00033  1 00001 2 00034         TXI *+1,2,1
00034  1 00001 1 00035         TXI *+1,1,1
00035  3 00000 1 00053   TXL5  TXH FIN,1,**
00036 -3 00011 2 00027         TXL CYCLE,2,9
00037  1 00001 4 00040         TXI *+1,4,1
00040  3 00011 4 00044         TXH BELOW,4,9
00041  0500 00 4 01660         CLA TENS,4
00042  0601 00 0 01012         STO DECML
00043  0020 00 0 00026         TRA CYCLE-1
00044 -0534 00 4 01010   BELOW LXD RGSTR,4
00045  0500 00 4 01666         CLA HNDRD,4
00046  0601 00 0 01011         STO OTHER
00047  1 00001 4 00050         TXI *+1,4,1
```

```
00050 -0634 00 4 01010          SXD RGSTR,4
00051  0600 00 0 01012          STZ DECML
00052  0020 00 0 00025          TRA CYCLE-2
00053  0020 00 0 00054   FIN    TRA *+1
                        *BEGIN SELECTING ROWS
00054  0774 00 4 00001          AXT 1,4
00055 -0634 00 4 01006   START  SXD ROWNO,4
                        *GENERATE TABLE EACH TIME
00056  0774 00 1 00001          AXT 1,1
00057  0500 00 0 77444          CLA ONED
00060  0622 00 1 01004          STD TABLE,1
00061  0400 00 0 77444          ADD ONED
00062  1 00001 1 00063          TXI *+1,1,1
00063 -3 00000 1 00060   TXL1   TXL *-3,1,**
00064  0020 00 0 00066          TRA *+2
                        *INSERT ROW NUMBER IN HEADING
00065  0 00000 0 00156          PZE MMARK
00066 -0534 00 4 01006   PTMTX  LXD ROWNO,4
00067  0500 00 4 01633          CLA ROWS,4
00070  0601 00 0 00160          STO MMARK+2
                        *PRINT HEADING
00071  0500 00 0 00065          CLA PTMTX-1
00072  0621 00 0 00101   STRT2  STA LST1
00073 -0500 00 0 00147          CAL TAPE2
00074  0074 00 4 00000          TSX $(STH),4
00075  0 00000 0 00155          PZE LELFT
00076 -0500 00 0 00150          CAL NUM
00077  0622 00 0 00104          STD TXH1
00100  0774 00 1 00000          AXT 0,1
00101  0560 00 1 00000   LST1   LDQ **,1
00102 -1 00000 0 00000          STR
00103  1 77777 1 00104          TXI *+1,1,-1
00104  3 00000 1 00101   TXH1   TXH LST1,1,**
00105  0074 00 4 00001          TSX $(FIL),4
                        *MODIFY TABLE TO INDICATE LINKS OF ROWNO ROW
00106  0774 00 4 00001          AXT 1,4
00107  0774 00 2 00001          AXT 1,2
00110  0774 00 1 00001   AXT1   AXT 1,1
00111 -0634 00 4 01007          SXD COLNO,4
00112 -0534 00 4 01006          LXD ROWNO,4
00113  0560 60 4 76742          LDQ* MATAX,4
00114 -0634 00 4 01006          SXD ROWNO,4
00115 -0534 00 4 01007          LXD COLNO,4
00116  0162 00 0 00120          TQP *+2
00117  0020 00 0 00121          TRA *+2
00120  0600 00 4 01004   STORE  STZ TABLE,4
00121 -0773 00 0 00001          RQL 1
00122  1 00001 1 00123          TXI *+1,1,1
00123  1 00001 4 00124          TXI *+1,4,1
00124 -3 00044 1 00116          TXL STORE-2,1,36
00125  1 00001 2 00126          TXI *+1,2,1
00126 -3 00000 2 00110   TXL3   TXL AXT1,2,**
00127  0020 00 0 00130          TRA OUT
```

```
                                    *PRINT MODIFIED TABLE
00130 -0500 00 0 00151   OUT    CAL NN
00131  0074 00 4 00000          TSX $(STH),4
00132  0 00000 0 00153          PZE FMT
00133  0774 00 1 00001          AXT 1,1
00134  0560 00 1 01004   LST2   LDQ TABLE,1
00135 -1 00000 0 00000          STR
00136  1 00001 1 00137          TXI *+1,1,1
00137 -3 00000 1 00134   TXL2   TXL LST2,1,**
00140  0074 00 4 00001          TSX $(FIL),4
00141  0020 00 0 00142          TRA NEXT
00142 -0534 00 4 01006   NEXT   LXD ROWNO,4
00143  1 00001 4 00144          TXI *+1,4,1
00144 -3 00000 4 00055   TXL4   TXL START,4,**
                                    *FINISH
00145 -0534 00 4 01005          LXD IR4,4
00146  0020 00 4 00001          TRA 1,4
00147  0 00002 0 00000   TAPE2  PZE 0,0,2
00150  0 77772 0 00000   NUM    PZE 0,0,-6
00151  0 00002 0 00000   NN     PZE 0,0,2
00152  063103346060             BCI 1,6I3)
00153  740130607303      FMT    BCI 1,(1H ,3
00154  210634606060             BCI 1,A6)
00155  740130007306      LELFT  BCI 1,(1HO,6
00156  512550643151      MMARK  BCI 6,REQUIREMENT        IS CONNECTED TO--
00157  254425456360
00160  606060606060
00161  603162602346
00162  454525236325
00163  246063464040
01004                           BES 400
01004  0 00000 0 00000   TABLE
01005  0 00000 0 00000   IR4
01006  0 00000 0 00000   ROWNO
01007  0 00000 0 00000   COLNO
01010  0 00000 0 00000   RGSTR  PZE
01011  0 00000 0 00000   OTHER  PZE
01012  0 00000 0 00000   DECML  PZE
01633                    ROWS   BES 400
01633 -377777777777             OCT 777777777777
01634  606060000011             BCI 1,   009
01635  606060000010             BCI 1,   008
01636  606060000007             BCI 1,   007
01637  606060000006             BCI 1,   006
01640  606060000005             BCI 1,   005
01641  606060000004             BCI 1,   004
01642  606060000003             BCI 1,   003
01643  606060000002             BCI 1,   002
01644  606060000001             BCI 1,   001
01645  606060000000      UNITS  BCI 1,   000
01646 -377777777777             OCT 777777777777
01647  000000001100             BCI 1,000090
01650  000000001000             BCI 1,000080
```

```
01651  000000000700            BCI 1,000070
01652  000000000600            BCI 1,000060
01653  000000000500            BCI 1,000050
01654  000000000400            BCI 1,000040
01655  000000000300            BCI 1,000030
01656  000000000200            BCI 1,000020
01657  000000000100            BCI 1,000010
01660  000000000000      TENS  BCI 1,000000
01661 -377777777777            OCT 777777777777
01662  000000040000            BCI 1,000400
01663  000000030000            BCI 1,000300
01664  000000020000            BCI 1,000200
01665  000000010000            BCI 1,000100
01666  000000000000      HNDRD BCI 1,000000
                        * COMMON BLOCK - REVISED 13 SEPTEMBER 1961
                 77462          COMMON -1
                 77462    INDIC COMMON 1
                 77461    ORDER COMMON 1
                 77460    NWORD COMMON 1
                 77457    DAT   COMMON 1
                 77456    LGTH  COMMON 1
                 77455    LATIS COMMON 1
                 77454    NBITH COMMON 1
                 77453    NBITL COMMON 1
                 77452    NBIT1 COMMON 1
                 77451    NBIT  COMMON 1
                 77450    NSQ1  COMMON 1
                 77447    OPRMN COMMON 1
                 77446    ATOMO COMMON 1
                 77445    ATOM  COMMON 1
                 77444    ONED  COMMON 1
                 77443    D36   COMMON 1
                 77442    ATOOX COMMON 10
                 77430    ATOX  COMMON 10
                 77416    SET   COMMON 10
                 77404    RANDM COMMON 10
                 77372    DIFF  COMMON 10
                 77360    CONVT COMMON 40
                 77310    DATA  COMMON 40
                 77240    MATA  COMMON 40
                 77170    UNIT  COMMON 40
                 77120    COMUN COMMON 40
                 77050    EQLS  COMMON 20
                 77024    SECTS COMMON 50
                 76742    MATAX COMMON 260
                 76336    DROWS COMMON 2100
                 72252    MROWS COMMON 2100
                 66166    INMAT COMMON 5400
                 53536    ATMS  COMMON 2000
                 47616    MACRO COMMON 7000
                          END
```

```
01667 IS THE FIRST LOCATION NOT USED BY THIS PROGRAM

   NO ERROR IN ABOVE ASSEMBLY
* DATE AND TIME NOW      4/12 1741.3
```

```
                00030          ENTRY   LCTRL
                00011          ENTRY   SET8

  TRANSFER VECTOR
00000   476343654360      PTLVL
00001   234664456360      COUNT
00002   475163314560      PRTIN
00003   512524642360      REDUC
00004   622363514360      SCTRL
00005   432751444560      LGRMN
00006   476343275160      PTLGR

  LINKAGE DIRECTOR
00007   000000000000
00010   432363514360


                          *              SET8 IS CALLED FROM MAIN, TO SET CONTROL PARAMETERS
00011 -0634 00 4 00237    SET8  SXD IR4,4
00012  0500 00 0 77460          CLA NWORD
00013 -0534 00 1 00027          LXD NUMB4,1
00014  0622 60 1 00026          STD* LOCS4,1
00015  2 00001 1 00014          TIX *-1,1,1
00016 -0534 00 4 00237          LXD IR4,4
00017  0020 00 4 00001          TRA 1,4
00020  0 00000 0 00000          PZE
00021  0 00000 0 00154          PZE M805
00022  0 00000 0 00212          PZE M804
00023  0 00000 0 00202          PZE M803
00024  0 00000 0 00055          PZE M802
00025  0 00000 0 00042          PZE M801
00026  0 00000 0 00000    LOCS4 PZE
00027  0 00005 0 00000    NUMB4 PZE 0,0,5
                          *              ENTER LCTRL
00030 -0634 00 4 00237    LCTRL SXD IR4,4
                          *              INITIALIZE REDST
00031 -0534 00 2 77460          LXD NWORD,2
00032  0600 00 2 00232          STZ REDST,2
00033  2 00001 2 00032          TIX *-1,2,1
00034  0774 00 4 00001          AXT 1,4
00035  0774 00 2 00001          AXT 1,2
                          *              PLACE ATOOX IN FIRST POSITION OF MACRO, THEN ZERO
                          *              END-OF-ROW MARKER
00036 -0500 00 2 77442          CAL ATOOX,2
00037  0602 00 4 47616          SLW MACRO,4
00040  1 00001 4 00041          TXI *+1,4,1
00041  1 00001 2 00042          TXI *+1,2,1
00042 -3 00000 2 00036    M801  TXL *-4,2,**
00043 -0534 00 2 77460          LXD NWORD,2
00044  0600 00 4 47616          STZ MACRO,4
00045  1 00001 4 00046          TXI *+1,4,1
00046  2 00001 2 00044          TIX *-2,2,1
```

```
00047  0774 00 1 00001          AXT 1,1
                        *              XR1 SELECTS SUBGRAPH FROM MACRO TO BECOME ATOX
00050  0774 00 2 00001   ARCHX  AXT 1,2
00051 -0500 00 1 47616          CAL MACRO,1
00052  0602 00 2 77430          SLW ATOX,2
00053  1 00001 1 00054          TXI *+1,1,1
00054  1 00001 2 00055          TXI *+1,2,1
00055 -3 00000 2 00051   M802   TXL *-4,2,**
00056  0760 00 0 00000          CLM
00057 -0534 00 2 77460          LXD NWORD,2
00060 -0501 00 2 77430          ORA ATOX,2
00061  2 00001 2 00060          TIX *-1,2,1
00062 -0100 00 0 00111          TNZ NZATM
                        *              IF ATOX IS EMPTY, PLACE ZERO MARKER IN MACRO
00063 -0534 00 2 77460          LXD NWORD,2
00064  0600 00 4 47616          STZ MACRO,4
00065  1 00001 4 00066          TXI *+1,4,1
00066  2 00001 2 00064          TIX *-2,2,1
                        *              --AND TEST REDST.WHETHER ALL OF ATOOX HAS BEEN THRU SCTRL
                        *              REDST IS CUMULATIVE RECORD OF VERTICES PASSED THRU SCTRL
00067 -0534 00 2 77460          LXD NWORD,2
00070 -0500 00 2 77442          CAL ATOOX,2
00071  0322 00 2 00232          ERA REDST,2
00072 -0602 00 0 00233          ORS TESST
00073  2 00001 2 00070          TIX *-3,2,1
00074 -0500 00 0 00233          CAL TESST
00075  0100 00 0 00214          TZE OUT
00076  0600 00 0 00233          STZ TESST
00077 -0634 00 1 00234          SXD SXRA,1
00100 -0634 00 2 00235          SXD SXRB,2
00101 -0634 00 4 00236          SXD SXRD,4
00102  0074 00 4 00000          CALL PTLVL
00103  1 00000 0 00105
00104  0 00146 0 00007
00105 -0534 00 1 00234          LXD SXRA,1
00106 -0534 00 2 00235          LXD SXRB,2
00107 -0534 00 4 00236          LXD SXRD,4
00110  0020 00 0 00050          TRA ARCHX
                        *              IF ATOX IS NOT EMPTY, CALCULATE NBIT, COMPARE WITH CHECK
00111  0600 00 0 77451   NZATM  STZ NBIT
00112 -0534 00 2 77460          LXD NWORD,2
00113  0500 00 0 77451          CLA NBIT
00114  0560 00 2 77430          LDQ ATOX,2
00115  0522 60 0 00001          XEC* $COUNT
00116  0622 00 0 77451          STD NBIT
00117  2 00001 2 00113          TIX *-4,2,1
00120  0500 00 0 77451          CLA NBIT
00121  0340 00 0 00232          CAS CHECK
00122  0020 00 0 00162          TRA MORE
00123  0020 00 0 00125          TRA LESS
00124  0020 00 0 00125          TRA LESS
                        *              IF NBIT LESS THAN OR EQUAL TO CHECK, USE SCTRL,SHRMN
00125 -0534 00 2 77460   LESS   LXD NWORD,2
```

```
00126 -0500 00 2 00232          CAL REDST,2
00127 -0320 00 2 77430          ANA ATOX,2
00130 -0100 00 0 00147          TNZ AFTER
00131  2 00001 2 00126          TIX *-3,2,1
                       *                IF ATOX ALREADY IN REDST, IT HAS BEEN THRU SCTRL ALREADY
                       *                IF NOT, PASS TO SCTRL
00132 -0634 00 4 00236          SXD SXRD,4
00133 -0634 00 1 00234          SXD SXRA,1
                       *                --PRINTING APPROPRIATE COMMENT IN OUTPUT
00134  0074 00 4 00002          CALL PRTIN
00135  1 00000 0 00137
00136  0 00206 0 00007
00137  0074 00 4 00003          CALL REDUC
00140  1 00000 0 00142
00141  0 00211 0 00007
00142  0074 00 4 00004          CALL SCTRL
00143  1 00000 0 00145
00144  0 00216 0 00007
                       *                ALL SMRMN RESULTS ARE PRINTED UNDER CONTROL OF SCTRL
00145 -0534 00 1 00234          LXD SXRA,1
00146 -0534 00 4 00236          LXD SXRD,4
00147  0774 00 2 00001   AFTER  AXT 1,2
00150 -0500 00 2 77430          CAL ATOX,2
00151  0602 00 4 47616          SLW MACRO,4
00152  1 00001 2 00153          TXI *+1,2,1
00153  1 00001 4 00154          TXI *+1,4,1
00154 -3 00000 2 00150   M805   TXL *-4,2,**
                       *                PLACE ATOX IN REDST
00155 -0534 00 2 77460          LXD NWORD,2
00156 -0500 00 2 77430          CAL ATOX,2
00157 -0602 00 2 00232          ORS REDST,2
00160  2 00001 2 00156          TIX *-2,2,1
00161  0020 00 0 00050          TRA ARCHX
                       *                IF NBIT GREATER THAN CHECK, USE LGRMN
00162  3 11610 4 00214   MORE   TXH OUT,4,5000
00163 -0634 00 1 00234          SXD SXRA,1
00164 -0634 00 4 00236          SXD SXRD,4
00165  0074 00 4 00005          CALL LGRMN
00166  1 00000 0 00170
00167  0 00245 0 00007
00170  0074 00 4 00006          CALL PTLGR
00171  1 00000 0 00173
00172  0 00252 0 00007
00173 -0534 00 1 00234          LXD SXRA,1
00174 -0534 00 4 00236          LXD SXRD,4
                       *                STORE PARTITION RESULTS IN MACRO
00175  0774 00 2 00001   ALPHX  AXT 1,2
00176 -0500 00 2 77416          CAL SET,2
00177  0602 00 4 47616          SLW MACRO,4
00200  1 00001 2 00201          TXI *+1,2,1
00201  1 00001 4 00202          TXI *+1,4,1
00202 -3 00000 2 00176   M803   TXL *-4,2,**
00203  0774 00 2 00001          AXT 1,2
```

```
00204 -0500 00 2 77416          CAL SET,2
00205  0760 00 0 00006          COM
00206 -0320 00 2 77430          ANA ATOX,2
00207  0602 00 4 47616          SLW MACRO,4
00210  1 00001 2 00211          TXI *+1,2,1
00211  1 00001 4 00212          TXI *+1,4,1
00212 -3 00000 2 00204   M804   TXL *-6,2,**
00213  0020 00 0 00050          TRA ARCHX
                        *                IF ALL OF ATOOX HAS BEEN DECOMPOSED, RETURN TO MAIN
00214 -0534 00 4 00237   OUT    LXD IR4,4
00215  0020 00 4 00001          TRA 1,4
00216  0 00000 0 00000          PZE
00217  0 00000 0 00000          PZE
00232                    REDST BES 10
00232 +000044000000      CHECK OCT 000044000000
00233  0 00000 0 00000   TESST PZE
00234  0 00000 0 00000   SXRA
00235  0 00000 0 00000   SXRB  PZE
00236  0 00000 0 00000   SXRD  PZE
00237  0 00000 0 00000   IR4   PZE
                        * COMMON BLOCK - REVISED  13 SEPTEMBER 1961
                  77462         COMMON -1
                  77462   INDIC COMMON 1
                  77461   ORDER COMMON 1
                  77460   NWORD COMMON 1
                  77457   DAT   COMMON 1
                  77456   LGTH  COMMON 1
                  77455   LATIS COMMON 1
                  77454   NBITH COMMON 1
                  77453   NBITL COMMON 1
                  77452   NBIT1 COMMON 1
                  77451   NBIT  COMMON 1
                  77450   NSQ1  COMMON 1
                  77447   OPRMN COMMON 1
                  77446   ATOMO COMMON 1
                  77445   ATOM  COMMON 1
                  77444   ONED  COMMON 1
                  77443   D36   COMMON 1
                  77442   ATOOX COMMON 10
                  77430   ATOX  COMMON 10
                  77416   SET   COMMON 10
                  77404   RANDM COMMON 10
                  77372   DIFF  COMMON 10
                  77360   CONVT COMMON 40
                  77310   DATA  COMMON 40
                  77240   MATA  COMMON 40
                  77170   UNIT  COMMON 40
                  77120   COMUN COMMON 40
                  77050   EQLS  COMMON 20
                  77024   SECTS COMMON 50
                  76742   MATAX COMMON 260
                  76336   DROWS COMMON 2100
                  72252   MROWS COMMON 2100
```

```
66166    INMAT COMMON 5400
53536    ATMS  COMMON 2000
47616    MACRO COMMON 7000
               END

00240 IS THE FIRST LOCATION NOT USED BY THIS PROGRAM

   NO ERROR IN ABOVE ASSEMBLY
* DATE AND TIME NOW      4/12 1740.4
```

```
                00020          ENTRY   LGRMN
                00003          ENTRY   SET9

  TRANSFER VECTOR
00000  234664456360      COUNT

  LINKAGE DIRECTOR
00001  000000000000
00002  432751444560


                              *              SET9 IS CALLED FROM MAIN, TO SET CONTROL PARAMETERS
00003 -0634 00 4 00635    SET9  SXD IR4,4
00004  0500 00 0 77461          CLA ORDER
00005  0774 00 1 00004          AXT 4,1
00006  0622 60 1 00017          STD* LOCS3,1
00007  2 00001 1 00006          TIX *-1,1,1
00010 -0534 00 4 00635          LXD IR4,4
00011  0020 00 4 00001          TRA 1,4
00012  0 00000 0 00000          PZE
00013  0 00000 0 00503          PZE S904
00014  0 00000 0 00360          PZE S903
00015  0 00000 0 00166          PZE S902
00016  0 00000 0 00060          PZE S901
00017  0 00000 0 00000    LOCS3 PZE
                              *                BEGIN PRELIMINARY OPERATIONS
00020 -0634 00 4 00635    LGRMN SXD IR4,4
00021 -0534 00 1 77457          LXD DAT,1
00022 -0500 00 1 76336          CAL DROWS,1
00023  0602 00 1 72252          SLW MROWS,1
00024  2 00001 1 00022          TIX *-2,1,1
                              *                COMPUTE NBIT
00025 -0534 00 2 77460          LXD NWORD,2
00026  0600 00 0 77451          STZ NBIT
00027  0500 00 0 77451          CLA NBIT
00030  0560 00 2 77430          LDQ ATOX,2
00031  0522 60 0 00000          XEC* $COUNT
00032  0622 00 0 77451          STD NBIT
00033  2 00001 2 00027          TIX *-4,2,1
                              *                JOSIE BLOCK FORMS NEW MATRIX FROM DROWS AND
                              *                STORES IT IN MROWS
00034 -0534 00 2 77460          LXD NWORD,2
00035 -0500 00 2 77430          CAL ATOX,2
00036 -0534 00 1 77461          LXD ORDER,1
00037  0320 60 1 76742          ANS* MATAX,1
00040  2 00001 1 00037          TIX *-1,1,1
00041  2 00001 2 00035          TIX *-4,2,1
00042  0774 00 4 00001          AXT 1,4
00043  0774 00 1 00001          AXT 1,1
00044 -0500 00 4 77430    CALAT CAL ATOX,4
00045 -0634 00 4 00640          SXD SXRD,4
00046  0774 00 4 00001          AXT 1,4
```

```
00047 -0760 00 0 00001   PBTT  PBT
00050  0020 00 0 00052         TRA *+2
00051  0020 00 0 00055         TRA SHIFT
00052 -0534 00 2 77460         LXD NWORD,2
00053  0600 60 1 76742         STZ* MATAX,1
00054  2 00001 2 00053         TIX *-1,2,1
00055  0767 00 0 00001   SHIFT ALS 1
00056  1 00001 1 00057         TXI *+1,1,1
00057  1 00001 4 00060         TXI *+1,4,1
00060  3 00000 1 00064   S901  TXH PREPR,1,**
00061 -3 00044 4 00047         TXL PBTT,4,36
00062 -0534 00 4 00640         LXD SXRD,4
00063  1 00001 4 00044         TXI CALAT,4,1
                       *               INITIALIZE OPNFO
00064  0500 00 0 00642   PREPR CLA =0377777777777
00065  0601 00 0 00556         STO OPNFO
                       *               COMPUTE NSQ1
00066  0500 00 0 77451         CLA NBIT
00067  0402 00 0 77444         SUB ONED
00070 -0765 00 0 00044         LGR 36
00071  0200 00 0 77451         MPY NBIT
00072  0767 00 0 00020         ALS 16
00073  0601 00 0 77450         STO NSQ1
                       *               COMPUTE TOTAL
00074 -0534 00 1 77457         LXD DAT,1
00075  0600 00 0 00555         STZ TOTAL
00076  0500 00 0 00555         CLA TOTAL
00077  0560 00 1 72252         LDQ MROWS,1
00100  0522 60 0 00000         XEC* $COUNT
00101  0622 00 0 00555         STD TOTAL
00102  2 00001 1 00076         TIX *-4,1,1
00103  0500 00 0 00555         CLA TOTAL
00104  0771 00 0 00001         ARS 1
00105  0622 00 0 00555         STD TOTAL
                       *               COMPUTE LUM
00106  0560 00 0 00641         LDQ =0
00107  0220 00 0 77450         DVH NSQ1
00110  0760 00 0 00000         CLM
00111 -0765 00 0 00021         LGR 17
00112 -0600 00 0 00565         STQ LUM
                       *               SET NDXX
00113  0500 00 0 77455         CLA LATIS
00114  0622 00 0 00612         STD NDXX
                       *               GENERATE START OF PATH, TSET, AND TEST IF EMPTY
00115 -0534 00 2 77460   JULX  LXD NWORD,2
00116 -0500 00 2 77404         CAL RANDM,2
00117 -0320 00 2 77430         ANA ATOX,2
00120  0602 00 2 00634         SLW TSET,2
00121  2 00001 2 00116         TIX *-3,2,1
00122 -0534 00 2 77460         LXD NWORD,2
00123 -0500 00 2 00634         CAL TSET,2
00124 -0100 00 0 00127         TNZ E1
00125  2 00001 2 00123         TIX *-2,2,1
00126  0020 00 0 00540         TRA XOOL
```

```
                                *              IF TSET NOT EMPTY, TEST IF TSET=ATOX
00127 -0534 00 2 77460   E1     LXD NWORD,2
00130 -0500 00 2 00634          CAL TSET,2
00131  0760 00 0 00006          COM
00132 -0320 00 2 77430          ANA ATOX,2
00133 -0100 00 0 00136          TNZ D1
00134  2 00001 2 00130          TIX *-4,2,1
00135  0020 00 0 00540          TRA XOOL
                                *              IF TSET NOT EMPTY AND NOT ATOM, COMPUTE INFO
00136  0020 00 0 00137   D1     TRA *+1
00137  0600 00 0 00603          STZ RR                                  NEW BLOCK
                                *              COMPUTE RR
00140  0774 00 4 00001          AXT 1,4
00141  0774 00 1 00001          AXT 1,1
00142  0560 00 4 00634   LDTST  LDQ TSET,4
00143 -0634 00 4 00640          SXD SXRD,4
00144  0774 00 4 00001          AXT 1,4
00145  0162 00 0 00163   TQPP   TQP SKIP
00146 -0600 00 0 00611          STQ WAIT
00147 -0534 00 2 77460   AROW   LXD NWORD,2
00150 -0500 00 2 00634          CAL TSET,2
00151  0760 00 0 00006          COM
00152 -0320 00 2 77430          ANA ATOX,2
00153 -0320 60 1 76742          ANA* MATAX,1
00154  0100 00 0 00161          TZE *+5
00155 -0765 00 0 00044          LGR 36
00156  0500 00 0 00603          CLA RR
00157  0522 60 0 00000          XEC* $COUNT
00160  0622 00 0 00603          STD RR
00161  2 00001 2 00150          TIX AROW+1,2,1
00162  0560 00 0 00611          LDQ WAIT
00163 -0773 00 0 00001   SKIP   RQL 1
00164  1 00001 1 00165          TXI *+1,1,1
00165  1 00001 4 00166          TXI *+1,4,1
00166  3 00000 1 00172   S902   TXH RRRR,1,**
00167 -3 00044 4 00145          TXL TQPP,4,36
00170 -0534 00 4 00640          LXD SXRD,4
00171  1 00001 4 00142          TXI LDTST,4,1
                                *              COMPUTE NN,MN,BOTT
00172  0500 00 0 00603   RRRR   CLA RR
00173  0600 00 0 00570          STZ NN
00174 -0534 00 2 77460          LXD NWORD,2
00175  0500 00 0 00570          CLA NN
00176  0560 00 2 00634          LDQ TSET,2
00177  0522 60 0 00000          XEC* $COUNT
00200  0622 00 0 00570          STD NN
00201  2 00001 2 00175          TIX *-4,2,1
00202  0500 00 0 77451          CLA NBIT
00203  0402 00 0 00570          SUB NN
00204 -0765 00 0 00044          LGR 36
00205  0200 00 0 00570          MPY NN
00206  0767 00 0 00021          ALS 17
00207  0622 00 0 00571          STD MN
```

```
00210   0500 00 0 77450          CLA NSQ1
00211   0402 00 0 00571          SUB MN
00212   0765 00 0 00043          LRS 35
00213   0200 00 0 00571          MPY MN
00214   0771 00 0 00001          ARS 1
00215   0601 00 0 00563          STQ BOTT
                        *                COMPUTE INFO FOR TSET.
00216   0500 00 0 00571          CLA MN
00217   0765 00 0 00043          LRS 35
00220   0200 00 0 00565          MPY LUM
00221   0763 00 0 00021          LLS 17
00222   0402 00 0 00603   SAD    SUB RR
00223   0760 00 0 00002          CHS
00224   0601 00 0 00567          STO MULT
00225   0120 00 0 00227          TPL *+2
00226   0760 00 0 00002          CHS
00227   0765 00 0 00043          LRS 35
00230   0200 00 0 00567          MPY MULT
00231   0765 00 0 00001          LRS 1
00232   0220 00 0 00563          DVH BOTT
00233  -0600 00 0 00560          STQ INFO
00234  -0600 00 0 00557          STQ BENFO
00235   0020 00 0 00236          TRA LOOPX
                        *                BEGIN HILL CLIMBING. ADD LOOP
00236   0500 00 0 00571   LOOPX  CLA MN
00237   0400 00 0 77451          ADD NBIT
00240   0402 00 0 00570          SUB NN
00241   0402 00 0 00570          SUB NN
00242   0402 00 0 77444          SUB ONED
00243   0601 00 0 00574          STO AMN
00244  -0520 00 0 00574          NZT AMN
                        *                IF AMN=0, THEN MODIFIED TSET IS ATOM--SKIP ADD LOOP
00245   0020 00 0 00363          TRA SLOOP
00246   0774 00 1 00001   ADDX   AXT 1,1
00247   0774 00 4 00001          AXT 1,4
00250   0774 00 2 00001          AXT 1,2
00251   0600 00 0 00605          STZ PA
00252   0600 00 0 00607          STZ QA
00253  -0500 00 2 77170   ADD1X  CAL UNIT,2
00254  -0320 00 4 77430          ANA ATOX,4
00255  -0634 00 2 00637          SXD SXRB,2
00256   0100 00 0 00355          TZE CUTAX
00257  -0320 00 4 00634          ANA TSET,4
00260  -0100 00 0 00355          TNZ CUTAX
                        * COMPUTE AR IF VERTEX XR1 IS IN ATOM BUT NOT IN TSET
00261  -0534 00 2 77460          LXD NWORD,2
00262  -0500 60 1 76742   IN2    CAL* MATAX,1
00263  -0320 00 2 00634          ANA TSET,2
00264  -0130 00 0 00000          XCL
00265   0500 00 0 00605          CLA PA
00266   0522 60 0 00000          XEC* $COUNT
00267   0622 00 0 00605          STD PA
00270  -0500 00 2 00634          CAL TSET,2
```

```
00271  0760 00 0 00006          COM
00272 -0320 00 2 77430          ANA ATOX,2
00273 -0320 60 1 76742   IN3    ANA* MATAX,1
00274 -0130 00 0 00000          XCL
00275  0500 00 0 00607          CLA QA
00276  0522 60 0 00000          XEC* $COUNT
00277  0622 00 0 00607          STD QA
00300  2 00001 2 00262          TIX IN2,2,1
00301  0500 00 0 00603          CLA RR
00302  0400 00 0 00607          ADD QA
00303  0402 00 0 00605          SUB PA
00304  0601 00 0 00601          STO AR
                        *               COMPUTE NEW BOTT CORRESPONDING TO MODIFIED
                        *               (BY ADDITION) TSET
00305  0500 00 0 77450          CLA NSQ1
00306  0402 00 0 00574          SUB AMN
00307  0765 00 0 00043          LRS 35
00310  0200 00 0 00574          MPY AMN
00311  0771 00 0 00001          ARS 1
00312  0601 00 0 00563          STO BOTT
                        *               COMPUTE ANFO, VALUE OF INFO FOR MODIFIED TSET
00313  0500 00 0 00574          CLA AMN
00314  0765 00 0 00043          LRS 35
00315  0200 00 0 00565          MPY LUM
00316  0763 00 0 00021          LLS 17
00317  0402 00 0 00601   ASAD   SUB AR
00320  0760 00 0 00002          CHS
00321  0601 00 0 00567          STO MULT
00322  0120 00 0 00324          TPL *+2
00323  0760 00 0 00002          CHS
00324  0765 00 0 00043          LRS 35
00325  0200 00 0 00567          MPY MULT
00326  0765 00 0 00001          LRS 1
00327  0220 00 0 00563          DVH BOTT
00330 -0600 00 0 00561          STQ ANFO
                        *               COMPARE ANFO, BENFO
00331  0500 00 0 00561          CLA ANFO
00332  0340 00 0 00557          CAS BENFO
00333  0020 00 0 00355          TRA CUTAX
00334  0020 00 0 00355          TRA CUTAX
                        *               IF ANFO LESS THAN BENFO, PLACE MODIFIED TSET IN BETST
00335  0601 00 0 00557          STO BENFO
00336  0500 00 0 00570          CLA NN
00337  0400 00 0 77444          ADD ONED
00340  0601 00 0 00576          STO NEWN
00341  0500 00 0 00574          CLA AMN
00342  0601 00 0 00600          STO NEWMN
00343  0500 00 0 00601          CLA AR
00344  0601 00 0 00577          STO NEWR
00345 -0534 00 2 77460          LXD NWORD,2
00346 -0500 00 2 00634          CAL TSET,2
00347  0602 00 2 00623          SLW BETST,2
00350  2 00001 2 00346          TIX *-2,2,1
```

```
00351 -0534 00 2 00637          LXD SXRB,2
00352 -0500 00 2 77170          CAL UNIT,2
00353 -0501 00 4 00634          ORA TSET,4
00354  0602 00 4 00623          SLW BETST,4
00355 -0534 00 2 00637   CUTAX  LXD SXRB,2
00356  1 00001 1 00357          TXI *+1,1,1
00357  1 00001 2 00360          TXI *+1,2,1
                       *                IF ALL POSSIBLE VERTICES HAVE BEEN TESTED FOR ADDITION
                       *                TO TSET, TRANSFER TO SUBTRACT LOOP.
00360  3 00000 1 00363   S903   TXH SLOOP,1,**
00361 -3 00044 2 00251          TXL ADDX+3,2,36
00362  1 00001 4 00250          TXI ADDX+2,4,1
                       *                SUBTRACT LOOP
00363  0500 00 0 00571   SLOOP  CLA MN
00364  0400 00 0 00570          ADD NN
00365  0400 00 0 00570          ADD NN
00366  0402 00 0 77451          SUB NBIT
00367  0402 00 0 77444          SUB ONED
00370  0601 00 0 00575          STO SMN
00371 -0520 00 0 00575          NZT SMN
                       *                IF SMN=0, MODIFIED TSET IS EMPTY, SKIP SUBTRACT LOOP.
00372  0020 00 0 00506          TRA COMPX
00373  0774 00 1 00001   SUBX   AXT 1,1
00374  0774 00 4 00001          AXT 1,4
00375  0774 00 2 00001          AXT 1,2
00376  0600 00 0 00606          STZ PS
00377  0600 00 0 00610          STZ QS
00400 -0500 00 2 77170   SUB1X  CAL UNIT,2
00401 -0320 00 4 00634          ANA TSET,4
00402 -0634 00 2 00637          SXD SXRB,2
00403  0100 00 0 00500          TZE CUTSX
                       *                IF VERTEX XR2 IS IN TSET, COMPUTE SR.
00404 -0534 00 2 77460          LXD NWORD,2
00405 -0500 60 1 76742   IN4    CAL* MATAX,1
00406 -0320 00 2 00634          ANA TSET,2
00407 -0130 00 0 00000          XCL
00410  0500 00 0 00606          CLA PS
00411  0522 60 0 00000          XEC* $COUNT
00412  0622 00 0 00606          STO PS
00413 -0500 00 2 00634          CAL TSET,2
00414  0760 00 0 00006          COM
00415 -0320 00 2 77430          ANA ATOX,2
00416 -0320 60 1 76742   IN5    ANA* MATAX,1
00417 -0130 00 0 00000          XCL
00420  0500 00 0 00610          CLA QS
00421  0522 60 0 00000          XEC* $COUNT
00422  0622 00 0 00610          STO QS
00423  2 00001 2 00405          TIX IN4,2,1
00424  0500 00 0 00603          CLA RR
00425  0400 00 0 00606          ADD PS
00426  0402 00 0 00610          SUB QS
00427  0601 00 0 00602          STO SR
                       *                COMPUTE BOTT FOR MODIFIED (BY SUBTRACTION) TSET
```

```
00430  0500 00 0 77450          CLA NSQ1
00431  0402 00 0 00575          SUB SMN
00432  0765 00 0 00043          LRS 35
00433  0200 00 0 00575          MPY SMN
00434  0771 00 0 00001          ARS 1
00435  0601 00 0 00563          STO BOTT
                       *                COMPUTE SNFO, VALUE OF INFO FOR MODIFIED
                       *                (BY SUBTRACTION) TSET
00436  0500 00 0 00575          CLA SMN
00437  0765 00 0 00043          LRS 35
00440  0200 00 0 00565          MPY LUM
00441  0763 00 0 00021          LLS 17
00442  0402 00 0 00602   SSAD   SUB SR
00443  0760 00 0 00002          CHS
00444  0601 00 0 00567          STO MULT
00445  0120 00 0 00447          TPL *+2
00446  0760 00 0 00002          CHS
00447  0765 00 0 00043          LRS 35
00450  0200 00 0 00567          MPY MULT
00451  0765 00 0 00001          LRS 1
00452  0220 00 0 00563          DVH BOTT
00453 -0600 00 0 00562          STQ SNFO
                       *                COMPARE SNFO, BENFO
00454  0500 00 0 00562          CLA SNFO
00455  0340 00 0 00557          CAS BENFO
00456  0020 00 0 00500          TRA CUTSX
00457  0020 00 0 00500          TRA CUTSX
                       *                IF SNFO LESS THAN BENFO, PLACE MODIFIED
                       *                (BY SUBTRACTION) TSET IN BETST
00460  0601 00 0 00557          STO BENFO
00461  0500 00 0 00570          CLA NN
00462  0402 00 0 77444          SUB ONED
00463  0601 00 0 00576          STO NEWN
00464  0500 00 0 00575          CLA SMN
00465  0601 00 0 00600          STO NEWMN
00466  0500 00 0 00602          CLA SR
00467  0601 00 0 00577          STO NEWR
00470 -0534 00 2 77460          LXD NWORD,2
00471 -0500 00 2 00634          CAL TSET,2
00472  0602 00 2 00623          SLW BETST,2
00473  2 00001 2 00471          TIX *-2,2,1
00474 -0534 00 2 00637          LXD SXRB,2
00475 -0500 00 2 77120          CAL COMUN,2
00476 -0320 00 4 00634          ANA TSET,4
00477  0602 00 4 00623          SLW BETST,4
00500 -0534 00 2 00637   CUTSX LXD SXRB,2
                       *                IF ALL POSSIBLE MODES HAVE BEEN TESTED FOR SUBTRACTION,
                       *                TRANSFER TO END-OF-PATH
00501  1 00001 1 00502          TXI *+1,1,1
00502  1 00001 2 00503          TXI *+1,2,1
00503  3 00000 1 00506   S904   TXH COMPX,1,**
00504 -3 00044 2 00376          TXL SUBX+3,2,36
00505  1 00001 4 00375          TXI SUBX+2,4,1
                       *                END-OF-PATH DECISION. COMPARE INFO WITH BENFO
```

```
00506  0500 00 0 00557   COMPX CLA BENFO
00507  0340 00 0 00560         CAS INFO
00510  0020 00 0 00526         TRA PT
00511  0020 00 0 00526         TRA PT
                       *              IF BENFO NOT LESS THAN INFO, THEN TSET IS END OF PATH,
                       *              AND TRANSFER TO PATH COMPARISONS
                       *              IF BENFO LESS THAN INFO, REPLACE TSET WITH ITS
                       *              MODIFICATION, BETST
00512  0601 00 0 00560         STO INFO
00513  0500 00 0 00600         CLA NEWMN
00514  0601 00 0 00571         STO MN
00515  0500 00 0 00576         CLA NEWN
00516  0601 00 0 00570         STO NN
00517  0500 00 0 00577         CLA NEWR
00520  0601 00 0 00603         STO RR                                                  L5
00521 -0534 00 2 77460         LXD NWORD,2
00522 -0500 00 2 00623         CAL BETST,2
00523  0602 00 2 00634         SLW TSET,2
00524  2 00001 2 00522         TIX *-2,2,1
00525  0020 00 0 00236         TRA LOOPX
                       *              RETURN TO ADD LOOP, TO TEST MODIFICATION OF NEW TSET
                       *              PATH COMPARISONS. TEST OPNFO TO SEE WHETHER THIS
                       *              END-OF-PATH IS BETTER THAN BEST PREVIOUS ONE.
00526  0500 00 0 00560   PT    CLA INFO
00527  0340 00 0 00556         CAS OPNFO
00530  0020 00 0 00540         TRA XOOL
00531  0020 00 0 00540         TRA XOOL
00532  0020 00 0 00533         TRA KING                                                MK5
00533  0601 00 0 00556   KING  STO OPNFO
00534 -0534 00 2 77460         LXD NWORD,2
00535 -0500 00 2 00634         CAL TSET,2
00536  0602 00 2 77416         SLW SET,2
00537  2 00001 2 00535         TIX *-2,2,1
00540  0500 00 0 00612   XOOL  CLA NDXX
00541  0402 00 0 77444         SUB ONED
00542  0622 00 0 00612         STD NDXX
00543 -0520 00 0 00612         NZT NDXX
00544  0020 00 0 00553         TRA OUT
                       *              PREPARE TO GENERATE NEW START OF PATH.
00545 -0534 00 2 77460         LXD NWORD,2
00546 -0500 00 2 77404         CAL RANDM,2
00547  0361 00 2 77372         ACL DIFF,2
00550  0602 00 2 77404         SLW RANDM,2
00551  2 00001 2 00546         TIX *-3,2,1
00552  0020 00 0 00115         TRA JULX
00553 -0534 00 4 00635   OUT   LXD IR4,4
00554  0020 00 4 00001         TRA 1,4
00555  0 00000 0 00000   TOTAL
00556  0 00000 0 00000   OPNFO
00557  0 00000 0 00000   BENFO
00560  0 00000 0 00000   INFO
00561  0 00000 0 00000   ANFO
00562  0 00000 0 00000   SNFO
```

```
00563  0 00000 0 00000   BOTT
00564  0 00000 0 00000   TOP
00565  0 00000 0 00000   LUM
00566  0 00000 0 00000   SIGN
00567  0 00000 0 00000   MULT
00570  0 00000 0 00000   NN
00571  0 00000 0 00000   MN
00572  0 00000 0 00000   RMN
00573  0 00000 0 00000   BERMN
00574  0 00000 0 00000   AMN
00575  0 00000 0 00000   SMN
00576  0 00000 0 00000   NEWN
00577  0 00000 0 00000   NEWR
00600  0 00000 0 00000   NEWMN
00601  0 00000 0 00000   AR
00602  0 00000 0 00000   SR
00603  0 00000 0 00000   RR
00604  0 00000 0 00000   BETRR
00605  0 00000 0 00000   PA
00606  0 00000 0 00000   PS
00607  0 00000 0 00000   QA
00610  0 00000 0 00000   QS
00611  0 00000 0 00000   WAIT
00612  0 00000 0 00000   NDXX
00623                          BES 8
00623  0 00000 0 00000   BETST
00634                          BES 8
00634  0 00000 0 00000   TSET
00635  0 00000 0 00000   IR4
00636  0 00000 0 00000   SXRA  PZE
00637  0 00000 0 00000   SXRB  PZE
00640  0 00000 0 00000   SXRD  PZE
                         *COMMON BLOCK
                         * COMMON BLOCK - REVISED  13 SEPTEMBER 1961
                  77462          COMMON -1
                  77462   INDIC COMMON 1
                  77461   ORDER COMMON 1
                  77460   NWORD COMMON 1
                  77457   DAT   COMMON 1
                  77456   LGTH  COMMON 1
                  77455   LATIS COMMON 1
                  77454   NBITH COMMON 1
                  77453   NBITL COMMON 1
                  77452   NBIT1 COMMON 1
                  77451   NBIT  COMMON 1
                  77450   NSQ1  COMMON 1
                  77447   OPRMN COMMON 1
                  77446   ATOMO COMMON 1
                  77445   ATOM  COMMON 1
                  77444   ONED  COMMON 1
                  77443   D36   COMMON 1
                  77442   ATOOX COMMON 10
                  77430   ATOX  COMMON 10
```

```
77416   SET    COMMON 10
77404   RANDM  COMMON 10
77372   DIFF   COMMON 10
77360   CONVT  COMMON 40
77310   DATA   COMMON 40
77240   MATA   COMMON 40
77170   UNIT   COMMON 40
77120   COMUN  COMMON 40
77050   EQLS   COMMON 20
77024   SECTS  COMMON 50
76742   MATAX  COMMON 260
76336   DROWS  COMMON 2100
72252   MROWS  COMMON 2100
66166   INMAT  COMMON 5400
53536   ATMS   COMMON 2000
47616   MACRO  COMMON 7000
               END

  LITERALS
00641  000000000000
00642  377777777777

00643 IS THE FIRST LOCATION NOT USED BY THIS PROGRAM


   NO ERROR IN ABOVE ASSEMBLY
* DATE AND TIME NOW      4/12 1740.6
```

DATE OF THIS DECK 3/1/62. VERSION OPERATIONAL AS OF 12/1/61. 4/12 1740.7

```
                00015          ENTRY   REDUC
                00002          ENTRY   SET11

  LINKAGE DIRECTOR
00000  000000000000
00001  512524642360


                                *               SET 11 IS CALLED FROM MAIN, SETS CONTROL PARAMETERS
00002 -0634 00 4 00116   SET11 SXD IR4,4
00003  0774 00 1 00002         AXT 2,1
00004  0500 00 0 77460         CLA NWORD
00005  0622 60 1 00014         STD* LOCS2,1
00006  2 00001 1 00005         TIX *-1,1,1
00007 -0534 00 4 00116         LXD IR4,4
00010  0020 00 4 00001         TRA 1,4
00011  0 00000 0 00000         PZE
00012  0 00000 0 00104         PZE R1102
00013  0 00000 0 00051         PZE R1101
00014  0 00000 0 00000   LOCS2 PZE
00015  0020 00 0 00016   REDUC TRA *+1
00016 -0634 00 4 00116         SXD IR4,4
                                *               PRELIMINARY. CLEAR CONVT,DATA,AND TRANSFER DROWS TO MROWS
00017  0774 00 1 00044         AXT 36,1
00020  0600 00 1 77360         STZ CONVT,1
00021  2 00001 1 00020         TIX *-1,1,1
00022  0774 00 1 00044         AXT 36,1
00023  0600 00 1 77310         STZ DATA,1
00024  2 00001 1 00023         TIX *-1,1,1
00025 -0534 00 4 77457         LXD DAT,4
00026 -0500 00 4 76336         CAL DROWS,4
00027  0602 00 4 72252         SLW MROWS,4
00030  2 00001 4 00026         TIX *-2,4,1
                                *               STORE IN CONVT THE IDENTIFICATION OF EACH VERTEX IN ATOX
00031  0774 00 4 00001         AXT 1,4
00032  0774 00 1 00001         AXT 1,1
00033  0774 00 2 00001         AXT 1,2                                 VARIABLE
00034  0560 00 2 77430   BB    LDQ ATOX,2
00035 -0634 00 2 00120         SXD SXRB,2
00036  0774 00 2 00001         AXT 1,2
00037  0162 00 0 00043   AA    TQP *+4
00040 -0754 00 1 00000         PXD 0,1
00041  0622 00 4 77360         STD CONVT,4
00042  1 00001 4 00043         TXI *+1,4,1
00043 -0773 00 0 00001         RQL 1
00044  1 00001 1 00045         TXI *+1,1,1
00045  1 00001 2 00046         TXI *+1,2,1
00046 -3 00044 2 00037         TXL AA,2,36
00047 -0534 00 2 00120         LXD SXRB,2
00050  1 00001 2 00051         TXI *+1,2,1
00051 -3 00000 2 00034   R1101 TXL BB,2,**
```

```
                                *               GENERATE DATA AS THE CONDENSED VERSION OF MROWS
                                *               CORRESPONDING TO ATOX
00052  0774 00 2 00001          AXT 1,2
00053  0600 00 0 00114          STZ MASK
00054  0560 00 2 77430   GG     LDQ ATOX,2
00055  0774 00 1 00001          AXT 1,1
00056  0162 00 0 00100   EE     TQP NOBIT
00057 -0500 00 0 00114          CAL MASK
00060  0771 00 0 00001          ARS 1
00061  0602 00 0 00114          SLW MASK
00062 -0520 00 0 00114          NZT MASK
00063 -0500 00 0 77167          CAL UNIT-1
00064  0602 00 0 00114          SLW MASK
00065 -0534 00 4 77451          LXD NBIT,4
00066  0500 00 4 77360   DD     CLA CONVT,4
00067 -0634 00 4 00115          SXD HOLD,4
00070 -0734 00 4 00000          PDX 0,4
00071 -0500 60 4 76742   FF     CAL* MATAX,4
00072 -0320 00 1 77170          ANA UNIT,1
00073 -0534 00 4 00115          LXD HOLD,4
00074  0100 00 0 00077          TZE JUMP
00075 -0500 00 0 00114          CAL MASK
00076 -0602 00 4 77310          ORS DATA,4
00077  2 00001 4 00066   JUMP   TIX DD,4,1
00100 -0773 00 0 00001   NOBIT  RQL 1
00101  1 00001 1 00102          TXI *+1,1,1
00102 -3 00044 1 00056          TXL EE,1,36
00103  1 00001 2 00104          TXI *+1,2,1
00104 -3 00000 2 00054   R1102  TXL GG,2,**
                                *               GENERATE ATOMO CORRESPONDING TO THE SIZE OF ATOX.
00105  0760 00 0 00000          CLM
00106 -0534 00 1 77451          LXD NBIT,1
00107 -0501 00 1 77170          ORA UNIT,1
00110  2 00001 1 00107          TIX *-1,1,1
00111  0602 00 0 77446          SLW ATOMO
00112 -0534 00 4 00116          LXD IR4,4
00113  0020 00 4 00001          TRA 1,4
00114  0 00000 0 00000   MASK   PZE
00115  0 00000 0 00000   HOLD   PZE
00116  0 00000 0 00000   IR4
00117  0 00000 0 00000   SXRA   PZE
00120  0 00000 0 00000   SXRB   PZE
00121  0 00000 0 00000   SXRD   PZE
                                * COMMON BLOCK - REVISED  13 SEPTEMBER 1961
                  77462         COMMON -1
                  77462   INDIC COMMON 1
                  77461   ORDER COMMON 1
                  77460   NWORD COMMON 1
                  77457   DAT   COMMON 1
                  77456   LGTH  COMMON 1
                  77455   LATIS COMMON 1
                  77454   NBITH COMMON 1
                  77453   NBITL COMMON 1
```

```
77452   NBIT1 COMMON 1
77451   NBIT  COMMON 1
77450   NSQ1  COMMON 1
77447   OPRMN COMMON 1
77446   ATOMO COMMON 1
77445   ATOM  COMMON 1
77444   ONED  COMMON 1
77443   D36   COMMON 1
77442   ATOOX COMMON 10
77430   ATOX  COMMON 10
77416   SET   COMMON 10
77404   RANDM COMMON 10
77372   DIFF  COMMON 10
77360   CONVT COMMON 40
77310   DATA  COMMON 40
77240   MATA  COMMON 40
77170   UNIT  COMMON 40
77120   COMUN COMMON 40
77050   EQLS  COMMON 20
77024   SECTS COMMON 50
76742   MATAX COMMON 260
76336   DROWS COMMON 2100
72252   MROWS COMMON 2100
66166   INMAT COMMON 5400
53536   ATMS  COMMON 2000
47616   MACRO COMMON 7000
        END

00122 IS THE FIRST LOCATION NOT USED BY THIS PROGRAM


   NO ERROR IN ABOVE ASSEMBLY
* DATE AND TIME NOW      4/12 1740.8
```

```
                00006          ENTRY   SCTRL
                00071          ENTRY   SIMPL

  TRANSFER VECTOR
00000   476343654360      PTLVL
00001   624451444560      SMRMN
00002   476323435160      PTCLR
00003   476346646360      PTOUT

  LINKAGE DIRECTOR
00004   000000000000
00005   622363514360


00006   0020 00 0 00007    SCTRL TRA *+1
00007  -0634 00 4 00112          SXD IR4,4
00010   0774 00 1 00000          AXT 0,1
00011   0774 00 2 00000          AXT 0,2
00012   0774 00 4 00000          AXT 0,4
                          *              PLACE ATOMO IN FIRST POSITION OF ATMS, THEN ZERO
                          *              END-OF-ROW MARKER.
00013  -0500 00 0 77446          CAL ATOMO
00014   0602 00 4 53536          SLW ATMS,4
00015   1 00001 4 00016          TXI *+1,4,1
00016   0600 00 4 53536          STZ ATMS,4
                          *              XR1 SELECTS SUBGRAPH FROM ATMS TO BECOME ATOM
00017  -0500 00 1 53536    ARCHE CAL ATMS,1
00020   0602 00 0 77445          SLW ATOM
00021   0100 00 0 00023          TZE *+2
00022   0020 00 0 00043          TRA PARTN
                          *              IF ATOM IS EMPTY, PLACE ZERO MARKER IN ATMS
00023   1 00001 4 00024          TXI *+1,4,1
00024   0600 00 4 53536          STZ ATMS,4
                          *              --AND TEST SIMST. WHETHER ALL OF ATOMO HAS BEEN REDUCED
                          *              TO COMPLETE GRAPHS.
                          *              SIMST IS RECORD OF ALL DECOMPOSITION PRODUCTS OF ATOMO
                          *              WHICH ARE COMPLETE GRAPHS,
                          *              AS DETERMINED BY NSQ1 TEST IN SMRMN.
00025  -0500 00 0 77446          CAL ATOMO
00026   0322 00 0 00106          ERA SIMST
00027   0100 00 0 00101          TZE OUT
00030  -0634 00 1 00107          SXD SXRA,1
00031  -0634 00 2 00110          SXD SXRB,2
00032  -0634 00 4 00111          SXD SXRD,4
00033   0074 00 4 00000          CALL PTLVL
00034   1 00000 0 00036
00035   0 00041 0 00004
00036  -0534 00 1 00107          LXD SXRA,1
00037  -0534 00 2 00110          LXD SXRB,2
00040  -0534 00 4 00111          LXD SXRD,4
                          *              BEGIN SIMST RECORD AGAIN WITH EACH NEW ROW OF THE
                          *              HIERARCHY.
```

```
00041  0600 00 0 00106          STZ SIMST
00042  1 00001 1 00017          TXI ARCHE,1,1
                       *              IF ATOM IS NOT EMPTY, PASS TO SMRM FOR PARTITIONING
                       *              AND IMMEDIATE PRINTOUT OF RESULTS
00043 -0634 00 1 00107  PARTN SXD SXRA,1
00044 -0634 00 2 00110         SXD SXRB,2
00045 -0634 00 4 00111         SXD SXRD,4
00046  0074 00 4 00001         CALL SMRMN
00047  1 00000 0 00051
00050  0 00056 0 00004
                       *              NORMAL PATH-SMRMN RETURNS CONTROL TO THIS POINT.
                       *              IF SMRMN DETERMINES THAT ATOM IS A COMPLETE GRAPH,
                       *              RETURN  TO SCTRL IS THROUGH SIMPL ENTRY POINT, BELOW.
00051  0074 00 4 00002         CALL PTCLR
00052  1 00000 0 00054
00053  0 00063 0 00004
00054 -0534 00 1 00107         LXD SXRA,1
00055 -0534 00 2 00110         LXD SXRB,2
00056 -0534 00 4 00111         LXD SXRD,4
00057  0020 00 0 00060         TRA ALPHA                                              K09
                       *              STORE PARTITION RESULTS IN ATMS
00060  1 00001 4 00061  ALPHA TXI *+1,4,1
00061 -0500 00 0 77416         CAL SET
00062  0602 00 4 53536         SLW ATMS,4
00063  1 00001 4 00064         TXI *+1,4,1
00064 -0500 00 0 77416         CAL SET
00065  0760 00 0 00006         COM
00066 -0320 00 0 77445         ANA ATOM
00067  0602 00 4 53536         SLW ATMS,4
00070  1 00001 1 00017         TXI ARCHE,1,1
                       *              ENTRY POINT FOR RETURN FROM SMRMN, IF ATOM IS A COMPLETE
                       *              GRAPH
00071 -0534 00 1 00107  SIMPL LXD SXRA,1
00072 -0534 00 2 00110         LXD SXRB,2
00073 -0534 00 4 00111         LXD SXRD,4
00074  1 00001 4 00075  STOR  TXI *+1,4,1
00075 -0500 00 0 77445         CAL ATOM
00076  0602 00 4 53536         SLW ATMS,4
                       *              ---FACT DULY NOTED BY RECORD IN SIMST
00077 -0602 00 0 00106         ORS SIMST
00100  1 00001 1 00017         TXI ARCHE,1,1
                       *              IF ATOMO HAS BEEN DECOMPOSED ENTIRELY INTO COMPLETE
                       *              GRAPHS, RETURN TO LCTRL.
00101  0074 00 4 00003  OUT    CALL PTOUT
00102  1 00000 0 00104
00103  0 00145 0 00004
00104 -0534 00 4 00112         LXD IR4,4
00105  0020 00 4 00001         TRA 1,4
00106  0 00000 0 00000  SIMST
00107  0 00000 0 00000  SXRA   PZE
00110  0 00000 0 00000  SXRB   PZE
00111  0 00000 0 00000  SXRD   PZE
00112  0 00000 0 00000  IR4
```

```
          * COMMON BLOCK - REVISED  13 SEPTEMBER 1961
77462             COMMON -1
77462    INDIC    COMMON 1
77461    ORDER    COMMON 1
77460    NWORD    COMMON 1
77457    DAT      COMMON 1
77456    LGTH     COMMON 1
77455    LATIS    COMMON 1
77454    NBITH    COMMON 1
77453    NBITL    COMMON 1
77452    NBIT1    COMMON 1
77451    NBIT     COMMON 1
77450    NSQ1     COMMON 1
77447    OPRMN    COMMON 1
77446    ATOMO    COMMON 1
77445    ATOM     COMMON 1
77444    ONED     COMMON 1
77443    D36      COMMON 1
77442    ATOOX    COMMON 10
77430    ATOX     COMMON 10
77416    SET      COMMON 10
77404    RANDM    COMMON 10
77372    DIFF     COMMON 10
77360    CONVT    COMMON 40
77310    DATA     COMMON 40
77240    MATA     COMMON 40
77170    UNIT     COMMON 40
77120    COMUN    COMMON 40
77050    EQLS     COMMON 20
77024    SECTS    COMMON 50
76742    MATAX    COMMON 260
76336    DROWS    COMMON 2100
72252    MROWS    COMMON 2100
66166    INMAT    COMMON 5400
53536    ATMS     COMMON 2000
47616    MACRO    COMMON 7000
                  END

00113 IS THE FIRST LOCATION NOT USED BY THIS PROGRAM


   NO ERROR IN ABOVE ASSEMBLY
* DATE AND TIME NOW      4/12 1740.9
```

```
                    * SMRMN. INFO AS MEASURE. PREPARED 12/1/62. DATE THIS CARD 3/1/62.
              00005       ENTRY   SMRMN

  TRANSFER VECTOR
00000  234664456360     COUNT
00001  623144474360     SIMPL
00002  266345474460     FTNPM

  LINKAGE DIRECTOR
00003  000000000000
00004  624451444560

00005  0020 00 0 00006  SMRMN TRA *+1
00006 -0634 00 4 00512       SXD IR4,4
                     *            BEGIN PRELIMINARY OPERATIONS
                     *            COMPUTE NBIT
00007  0760 00 0 00000       CLM
00010  0560 00 0 77445       LDQ ATOM                    PREPARE TO COUNT BITS IN ATOM
00011  0522 60 0 00000       XEC* $COUNT
00012  0622 00 0 77451       STD NBIT
                     *            INITIALIZE OPNFO
00013  0500 00 0 00514       CLA =0377777777777
00014  0601 00 0 00500       STO OPNFO
                     *            COMPUTE NSQ1
00015  0500 00 0 77451       CLA NBIT
00016  0402 00 0 77444       SUB ONED .
00017 -0765 00 0 00044       LGR 36
00020  0200 00 0 77451       MPY NBIT
00021  0767 00 0 00020       ALS 16
00022  0601 00 0 77450       STO NSQ1
                     *            SET LATS1
00023  0500 00 0 77455       CLA LATIS
00024  0601 00 0 00456       STO LATS1
                     *            JOSIE BLOCK FORMS NEW MATRIX FROM DATA AND STORES IT IN
                     *            MATA BLOCK
00025  0774 00 1 00044  JOSIE AXT 36,1
00026 -0500 00 1 77310       CAL DATA,1
00027  0602 00 1 77240       SLW MATA,1
00030  2 00001 1 00026       TIX *-2,1,1
00031  0774 00 1 00044       AXT 36,1
00032 -0500 00 0 77445       CAL ATOM
00033  0320 00 1 77240       ANS MATA,1
00034  2 00001 1 00033       TIX *-1,1,1
00035  0774 00 1 00044       AXT 36,1
00036  0760 00 0 00001       LBT
00037  0600 00 1 77240       STZ MATA,1
00040  0771 00 0 00001       ARS 1
00041  2 00001 1 00036       TIX *-3,1,1
                     *            COMPUTE TOTAL AND COMPARE WITH NSQ1
00042  0774 00 1 00044       AXT 36,1
00043  0600 00 0 00457       STZ TOTAL
```

```
00044   0500 00 0 00457          CLA TOTAL
00045   0560 00 1 77240          LDQ MATA,1
00046   0522 60 0 00000          XEC* $COUNT
00047   0622 00 0 00457          STD TOTAL
00050   2 00001 1 00044          TIX *-4,1,1
00051   0500 00 0 00457          CLA TOTAL
00052   0771 00 0 00001          ARS 1
00053   0622 00 0 00457          STD TOTAL
00054   0100 00 0 00061          TZE JAKE
                        *                 IF TOTAL=0, THEN THERE ARE NO LINKS AMONG THE VERTICES
                        *                 OF ATOM. RETURN TO SCTRL VIA THE SIMPL ENTRY POINT.
00055   0340 00 0 77450          CAS NSQ1
00056   0000 00 0 00435          HTR EX
00057   0020 00 0 00061          TRA *+2
00060   0020 00 0 00064          TRA HAPPY
                        *             IF TOTAL=NSQ1, THEN ATOM IS A COMPLETE GRAPH, RETURN TO
                        *             SCTRL VIA SIMPL ENTRY POINT.
00061   0074 00 4 00001   JAKE   CALL SIMPL
00062   1 00000 0 00064
00063   0 00411 0 00003
                        *                 COMPUTE LUM
00064   0560 00 0 00513   HAPPY LDQ =0
00065   0220 00 0 77450          DVH NSQ1
00066   0760 00 0 00000          CLM
00067  -0765 00 0 00021          LGR 17
00070  -0600 00 0 00507          STQ LUM
                        *                 GENERATE START OF PATH, TSET, AND TEST IF EMPTY.
00071  -0500 00 0 77404   JULY   CAL RANDM
00072  -0320 00 0 77445          ANA ATOM
00073   0602 00 0 00460          SLW TSET
00074   0100 00 0 00421          TZE KOOL
                        *                 IF TSET NOT EMPTY, TEST IF TSET=ATOM
00075   0760 00 0 00006          COM
00076  -0320 00 0 77445          ANA ATOM
00077   0100 00 0 00421          TZE KOOL
                        * BLOCK D CALCULATES R/MN FOR TSET
                        *                 IF TSET NOT EMPTY AND NOT ATOM, COMPUTE INFO--
00100   0560 00 0 00460   D1     LDQ TSET                                                  D1
                        *                 COMPUTE RR
00101   0600 00 0 00461          STZ RR
00102   0774 00 1 00001          AXT 1,1
00103   0162 00 0 00116   FEED4  TQP SKIP                                                 D4    570
00104  -0500 00 0 00460          CAL TSET                                                  D5    580
00105   0760 00 0 00006          COM                                                       D6    590
00106  -0320 00 0 77445          ANA ATOM
00107  -0320 00 1 77240          ANA MATA,1
00110  -0600 00 0 00450          STQ HOLD
00111  -0130 00 0 00000          XCL
00112   0500 00 0 00461          CLA RR
00113   0522 60 0 00000          XEC* $COUNT
00114   0622 00 0 00461          STD RR
00115   0560 00 0 00450          LDQ HOLD
00116  -0773 00 0 00001   SKIP   RQL 1                                                    D14   670
```

```
00117   1 00001 1 00120          TXI FEED3,1,1                                  D15  680
00120  -3 00044 1 00103   FEED3  TXL FEED4,1,36                                 D16  690
                        *                COMPUTE NN,MN,BOTT
00121   0760 00 0 00000          CLM
00122   0560 00 0 00460          LDQ TSET
00123   0522 60 0 00000          XEC* $COUNT
00124   0622 00 0 00463          STD NN
00125   0500 00 0 77451          CLA NBIT
00126   0402 00 0 00463          SUB NN
00127  -0765 00 0 00044          LGR 36
00130   0200 00 0 00463          MPY NN
00131   0767 00 0 00021          ALS 17
00132   0622 00 0 00462          STD MN
00133   0500 00 0 77450          CLA NSQ1
00134   0402 00 0 00462          SUB MN
00135   0765 00 0 00043          LRS 35
00136   0200 00 0 00462          MPY MN
00137   0771 00 0 00001          ARS 1
00140   0601 00 0 00505          STO BOTT
                        *                COMPUTE INFO FOR TSET.
00141   0500 00 0 00462          CLA MN
00142   0765 00 0 00043          LRS 35
00143   0200 00 0 00507          MPY LUM
00144   0763 00 0 00021          LLS 17
00145   0402 00 0 00461   SAD    SUB RR
00146   0760 00 0 00002          CHS
00147   0601 00 0 00511          STO MULT
00150   0120 00 0 00152          TPL *+2
00151   0760 00 0 00002          CHS
00152   0765 00 0 00043          LRS 35
00153   0200 00 0 00511          MPY MULT
00154   0765 00 0 00001          LRS 1
00155   0220 00 0 00505          DVH BOTT
00156  -0600 00 0 00502          STQ INFO
00157  -0600 00 0 00501          STQ BENFO
00160   0020 00 0 00161          TRA LOOP
                        *                BEGIN HILLCLIMBING
                        *                ADD LOOP
00161   0774 00 1 00044   LOOP   AXT 36,1
00162   0500 00 0 00462          CLA MN
00163   0400 00 0 77451          ADD NBIT
00164   0402 00 0 00463          SUB NN
00165   0402 00 0 00463          SUB NN
00166   0402 00 0 77444          SUB ONED
00167   0601 00 0 00464          STO AMN
00170  -0520 00 0 00464          NZT AMN
                        *                IF AMN=0, THEN MODIFIED TSET IS ATOM-SKIP ADD LOOP.
00171   0020 00 0 00266          TRA SUB
00172  -0500 00 1 77170   ADD1   CAL UNIT,1                                     E2  1140
00173  -0320 00 0 77445          ANA ATOM
00174   0100 00 0 00264          TZE CUTA
00175  -0320 00 0 00460          ANA TSET                                       E3  1150
00176  -0100 00 0 00264          TNZ CUTA                                       E4  1160
```

```
                                *              COMPUTE AR, IF VERTEX XR1 IS IN ATOM BUT NOT IN TSET
00177 -0500 00 1 77240          CAL MATA,1
00200 -0320 00 0 00460          ANA TSET                                                    E6  1180
00201 -0765 00 0 00044          LGR 36
00202  0500 00 0 00513          CLA =0
00203  0522 60 0 00000          XEC* $COUNT
00204  0622 00 0 00467          STO PA
00205 -0500 00 0 00460          CAL TSET                                                    E15 1270
00206  0760 00 0 00006          COM                                                         E16 1280
00207 -0320 00 0 77445          ANA ATOM
00210 -0320 00 1 77240          ANA MATA,1
00211 -0765 00 0 00044          LGR 36
00212  0500 00 0 00513          CLA =0
00213  0522 60 0 00000          XEC* $COUNT
00214  0622 00 0 00470          STO QA
00215  0500 00 0 00461          CLA RR                                                      E26 1380
00216  0400 00 0 00470          ADD QA                                                      E27 1390
00217  0402 00 0 00467          SUB PA                                                      E28 1400
00220  0601 00 0 00471          STO AR
                                *              COMPUTE NEW BOTT CORRESPONDING TO MODIFIED
                                *              (BY ADDITION) TSET
00221  0500 00 0 77450          CLA NSQ1
00222  0402 00 0 00464          SUB AMN
00223  0765 00 0 00043          LRS 35
00224  0200 00 0 00464          MPY AMN
00225  0771 00 0 00001          ARS 1
00226  0601 00 0 00505          STO BOTT
                                *              COMPUTE ANFO, VALUE OF INFO FOR MODIFIED TSET
00227  0500 00 0 00464          CLA AMN
00230  0765 00 0 00043          LRS 35
00231  0200 00 0 00507          MPY LUM
00232  0763 00 0 00021          LLS 17
00233  0402 00 0 00471   ASAD   SUB AR
00234  0760 00 0 00002          CHS
00235  0601 00 0 00511          STO MULT
00236  0120 00 0 00240          TPL *+2
00237  0760 00 0 00002          CHS
00240  0765 00 0 00043          LRS 35
00241  0200 00 0 00511          MPY MULT
00242  0765 00 0 00001          LRS 1
00243  0220 00 0 00505          DVH BOTT
00244 -0600 00 0 00503          STQ ANFO
                                *              COMPARE ANFO, BENFO
00245  0500 00 0 00503          CLA ANFO
00246  0340 00 0 00501          CAS BENFO
00247  0020 00 0 00264          TRA CUTA                                                    F5
00250  0020 00 0 00264          TRA CUTA                                                    F6
                                *              IF ANFO LESS THAN BENFO, PLACE MODIFIED TSET IN BETST
00251  0601 00 0 00501          STO BENFO
00252  0500 00 0 00463          CLA NN
00253  0400 00 0 77444          ADD ONED
00254  0601 00 0 00474          STO NEWN
00255  0500 00 0 00464          CLA AMN
```

```
00256  0601 00 0 00472          STO NEWMN
00257  0500 00 0 00471          CLA AR
00260  0601 00 0 00473          STO NEWR
00261 -0500 00 1 77170          CAL UNIT,1
00262 -0501 00 0 00460          ORA TSET                                        F8
00263  0602 00 0 00451          SLW BETST
00264  2 00001 1 00172  CUTA    TIX ADD1,1,1                                    F10
                        *             IF ALL POSSIBLE VERTICES HAVE BEEN TESTED FOR ADDITION
                        *             TO TSET, TRANSFER TO SUBTRACT LOOP.
00265  0020 00 0 00266          TRA SUB
                        *             SUBTRACT LOOP.
00266  0774 00 1 00044  SUB     AXT 36,1
00267  0500 00 0 00462          CLA MN
00270  0400 00 0 00463          ADD NN
00271  0400 00 0 00463          ADD NN
00272  0402 00 0 77451          SUB NBIT
00273  0402 00 0 77444          SUB ONED
00274  0601 00 0 00455          STO SMN
00275 -0520 00 0 00455          NZT SMN
                        *             IF SMN=0, MODIFIED TSET IS EMPTY--SKIP SUBTRACT LOOP
00276  0020 00 0 00372          TRA COMPA
00277 -0500 00 1 77170  SUB1    CAL UNIT,1
00300 -0320 00 0 00460          ANA TSET                                        G2  1760
00301  0100 00 0 00370          TZE CUTS                                        G3  1770
                        *             IF VERTEX XR1 IS IN TSET, COMPUTE SR.     G4  1780
00302 -0500 00 1 77240          CAL MATA,1
00303 -0320 00 0 00460          ANA TSET
00304 -0765 00 0 00044          LGR 36                                          G6  1800
00305  0500 00 0 00513          CLA =0
00306  0522 60 0 00000          XEC* $COUNT
00307  0622 00 0 00452          STD PS
00310 -0500 00 0 00460          CAL TSET
00311  0760 00 0 00006          COM                                             G15 1890
00312 -0320 00 0 77445          ANA ATOM                                        G16 1900
00313 -0320 00 1 77240          ANA MATA,1
00314 -0765 00 0 00044          LGR 36
00315  0500 00 0 00513          CLA =0
00316  0522 60 0 00000          XEC* $COUNT
00317  0622 00 0 00453          STD QS
00320  0500 00 0 00461          CLA RR
00321  0400 00 0 00452          ADD PS                                          G26 2000
00322  0402 00 0 00453          SUB QS                                          G27 2010
00323  0601 00 0 00454          STO SR                                          G28 2020
                        *             COMPUTE BOTT FOR MODIFIED (BY SUBTRACTION) TSET
00324  0500 00 0 77450          CLA NSQ1
00325  0402 00 0 00455          SUB SMN
00326  0765 00 0 00043          LRS 35
00327  0200 00 0 00455          MPY SMN
00330  0771 00 0 00001          ARS 1
00331  0601 00 0 00505          STO BOTT
                        *             COMPUTE SNFO, VALUE OF INFO FOR MODIFIED
                        *             (BY SUBTRACTION) TSET
```

```
00332  0500 00 0 00455          CLA SMN
00333  0765 00 0 00043          LRS 35
00334  0200 00 0 00507          MPY LUM
00335  0763 00 0 00021          LLS 17
00336  0402 00 0 00454   SSAD   SUB SR
00337  0760 00 0 00002          CHS
00340  0601 00 0 00511          STO MULT
00341  0120 00 0 00343          TPL *+2
00342  0760 00 0 00002          CHS
00343  0765 00 0 00043          LRS 35
00344  0200 00 0 00511          MPY MULT
00345  0765 00 0 00001          LRS 1
00346  0220 00 0 00505          DVH BOTT
00347 -0600 00 0 00504          STQ SNFO
                        *                COMPARE SNFO, BENFO
00350  0500 00 0 00504          CLA SNFO
00351  0340 00 0 00501          CAS BENFO
00352  0020 00 0 00370          TRA CUTS                                        H5
00353  0020 00 0 00370          TRA CUTS                                        H6
                        *                IF SNFO LESS THAN BENFO, PLACE MODIFIED TSET IN BETST
00354  0601 00 0 00501          STO BENFO
00355  0500 00 0 00463          CLA NN
00356  0402 00 0 77444          SUB ONED
00357  0601 00 0 00474          STO NEWN
00360  0500 00 0 00455          CLA SMN
00361  0601 00 0 00472          STO NEWMN
00362  0500 00 0 00454          CLA SR
00363  0601 00 0 00473          STO NEWR
00364 -0500 00 1 77170          CAL UNIT,1                                      H8
00365  0760 00 0 00006          COM                                             H9
00366 -0320 00 0 00460          ANA TSET
00367  0602 00 0 00451          SLW BETST
00370  2 00001 1 00277   CUTS   TIX SUB1,1,1                                    H11
                        *                IF ALL POSSIBLE MODES HAVE BEEN TESTED FOR SUBTRACTION
                        *                FROM TSET, TRANSFER TO END-OF-PATH
00371  0020 00 0 00372          TRA COMPA
00372  0500 00 0 00501   COMPA  CLA BENFO
                        *                END-OF-PATH DECISION. COMPARE INFO WITH BENFO
00373  0340 00 0 00502          CAS INFO
00374  0020 00 0 00410          TRA PRINT                                       L3
00375  0020 00 0 00410          TRA PRINT                                       L4
                        *                IF BENFO NOT LESS THAN INFO, THEN TSET IS END OF PATH,
                        *                AND TRANSFER TO PATH COMPARISONS
                        *                IF BENFO LESS THAN INFO, REPLACE TSET WITH ITS
                        *                MODIFICATION BETST
00376  0601 00 0 00502          STO INFO
00377  0500 00 0 00472          CLA NEWMN
00400  0601 00 0 00462          STO MN
00401  0500 00 0 00474          CLA NEWN
00402  0601 00 0 00463          STO NN
00403  0500 00 0 00473          CLA NEWR
00404  0601 00 0 00461          STO RR                                          L5
00405 -0500 00 0 00451          CAL BETST                                       L6
00406  0602 00 0 00460          SLW TSET                                        L7
```

```
                                *            RETURN TO ADD LOOP, TO TEST MODIFICATIONS OF NEW TSET
00407  0020 00 0 00161                TRA LOOP
                                *                                                              L8
                                *            PATH COMPARISONS. TEST OPNFO TO SEE WHETHER THIS
                                *            END-OF-PATH IS BETTER THAN BEST PREVIOUS ONE
00410  0500 00 0 00502  PRINT  CLA INFO
00411  0340 00 0 00500         CAS OPNFO
00412  0020 00 0 00421         TRA KOOL
00413  0020 00 0 00421         TRA KOOL                                                       MK3
00414  0020 00 0 00415         TRA KING                    REFINE
00415  0601 00 0 00500  KING   STO OPNFO                                                      MK5
00416 -0500 00 0 00460         CAL TSET
00417  0602 00 0 77416         SLW SET                                                        MK7
00420  0020 00 0 00421         TRA KOOL
00421  0500 00 0 00456  KOOL   CLA LATS1                                                     MK9
00422  0402 00 0 77444         SUB ONED                                                       K01
00423  0601 00 0 00456         STO LATS1
                                *                                                              K03
                                *            STOP-SAMPLING DECISION. IF NO. OF PATHS EQUALS
                                *            LATIS, THEN RETURN TO SCTRL.
00424 -0520 00 0 00456         NZT LATS1
00425  0020 00 0 00432         TRA CLR
                                *            PREPARE TO GENERATE NEW START OF PATH
00426 -0500 00 0 77404         CAL RANDM
00427  0361 00 0 77372         ACL DIFF
00430  0602 00 0 77404         SLW RANDM
00431  0020 00 0 00071         TRA JULY
00432  0020 00 0 00433  CLR    TRA *+1                                                        DOGE 3
00433 -0534 00 4 00512         LXD IR4,4
00434  0020 00 4 00001         TRA 1,4
00435  0074 00 4 00002  EX     CALL FTNPM,LAST
00436  0074 00 0 00443
00437  1 00000 0 00441
00440  0 21005 0 00003
00441 -0534 00 4 00512         LXD IR4,4
00442  0020 00 4 00001         TRA 1,4
00443  744421314534    LAST   BCI 4,(MAIN)(/1,/77777)
00444  746101736107
00445  070707073460
00446  606060606060
00447 -377777777777
00450  0 00000 0 00000         OCT 777777777777
00451  0 00000 0 00000  HOLD
00452  0 00000 0 00000  BETST
00453  0 00000 0 00000  PS                                                                    A13*
00454  0 00000 0 00000  QS                                                                    A10  390
00455  0 00000 0 00000  SR                                                                    A11  400
00456  0 00000 0 00000  SMN
00457  0 00000 0 00000  LATS1
00460  0 00000 0 00000  TOTAL
00461  0 00000 0 00000  TSET
00462  0 00000 0 00000  RR     PZE
00463  0 00000 0 00000  MN
00464  0 00000 0 00000  NN
00465  0 00000 0 00000  AMN
                        BERMN
```

```
00466  0 00000 0 00000   RMN
00467  0 00000 0 00000   PA                                                A8   370
00470  0 00000 0 00000   QA                                                A9   380
00471  0 00000 0 00000   AR
00472  0 00000 0 00000   NEWMN
00473  0 00000 0 00000   NEWR
00474  0 00000 0 00000   NEWN
00475  0 00000 0 00000   SXRA
00476  0 00000 0 00000   SXRB   PZE
00477  0 00000 0 00000   SXRD   PZE
00500  0 00000 0 00000   OPNFO
00501  0 00000 0 00000   BENFO
00502  0 00000 0 00000   INFO
00503  0 00000 0 00000   ANFO
00504  0 00000 0 00000   SNFO
00505  0 00000 0 00000   BOTT
00506  0 00000 0 00000   TOP
00507  0 00000 0 00000   LUM
00510  0 00000 0 00000   SIGN
00511  0 00000 0 00000   MULT
00512  0 00000 0 00000   IR4
                        * COMMON BLOCK - REVISED  13 SEPTEMBER 1961
                  77462         COMMON -1
                  77462   INDIC COMMON 1
                  77461   ORDER COMMON 1
                  77460   NWORD COMMON 1
                  77457   DAT   COMMON 1
                  77456   LGTH  COMMON 1
                  77455   LATIS COMMON 1
                  77454   NBITH COMMON 1
                  77453   NBITL COMMON 1
                  77452   NBIT1 COMMON 1
                  77451   NBIT  COMMON 1
                  77450   NSQ1  COMMON 1
                  77447   OPRMN COMMON 1
                  77446   ATOMO COMMON 1
                  77445   ATOM  COMMON 1
                  77444   ONED  COMMON 1
                  77443   D36   COMMON 1
                  77442   ATOOX COMMON 10
                  77430   ATOX  COMMON 10
                  77416   SET   COMMON 10
                  77404   RANDM COMMON 10
                  77372   DIFF  COMMON 10
                  77360   CONVT COMMON 40
                  77310   DATA  COMMON 40
                  77240   MATA  COMMON 40
                  77170   UNIT  COMMON 40
                  77120   COMUN COMMON 40
                  77050   EQLS  COMMON 20
                  77024   SECTS COMMON 50
                  76742   MATAX COMMON 260
```

```
72252   MROWS COMMON 2100
66166   INMAT COMMON 5400
53536   ATMS  COMMON 2000
47616   MACRO COMMON 7000
              END

  LITERALS
00513  000000000000
00514  377777777777
```

```
                00103          ENTRY   CNVRT
                00002          ENTRY   COUNT

  LINKAGE DIRECTOR
00000  000000000000
00001  234565516360


00002 -0114 06 0 00003    COUNT CAQ BITAB,,6
00003   0 00000 0 00003    BITAB PZE     BITAB,,0          OCTAL 0
00004   0 00001 0 00003          PZE     BITAB,,1          1
00005   0 00001 0 00003          PZE     BITAB,,1          2
00006   0 00002 0 00003          PZE     BITAB,,2          3
00007   0 00001 0 00003          PZE     BITAB,,1          4
00010   0 00002 0 00003          PZE     BITAB,,2          5
00011   0 00002 0 00003          PZE     BITAB,,2          6
00012   0 00003 0 00003          PZE     BITAB,,3          7
00013   0 00001 0 00003          PZE     BITAB,,1          10
00014   0 00002 0 00003          PZE     BITAB,,2          11
00015   0 00002 0 00003          PZE     BITAB,,2          12
00016   0 00003 0 00003          PZE     BITAB,,3          13
00017   0 00002 0 00003          PZE     BITAB,,2          14
00020   0 00003 0 00003          PZE     BITAB,,3          15
00021   0 00003 0 00003          PZE     BITAB,,3          16
00022   0 00004 0 00003          PZE     BITAB,,4          17
00023   0 00001 0 00003          PZE     BITAB,,1          20
00024   0 00002 0 00003          PZE     BITAB,,2          21
00025   0 00002 0 00003          PZE     BITAB,,2          22
00026   0 00003 0 00003          PZE     BITAB,,3          23
00027   0 00002 0 00003          PZE     BITAB,,2          24
00030   0 00003 0 00003          PZE     BITAB,,3          25
00031   0 00003 0 00003          PZE     BITAB,,3          26
00032   0 00004 0 00003          PZE     BITAB,,4          27
00033   0 00002 0 00003          PZE     BITAB,,2          30
00034   0 00003 0 00003          PZE     BITAB,,3          31
00035   0 00003 0 00003          PZE     BITAB,,3          32
00036   0 00004 0 00003          PZE     BITAB,,4          33
00037   0 00003 0 00003          PZE     BITAB,,3          34
00040   0 00004 0 00003          PZE     BITAB,,4          35
00041   0 00004 0 00003          PZE     BITAB,,4          36
00042   0 00005 0 00003          PZE     BITAB,,5          37
00043   0 00001 0 00003          PZE     BITAB,,1          40
00044   0 00002 0 00003          PZE     BITAB,,2          41
00045   0 00002 0 00003          PZE     BITAB,,2          42
00046   0 00003 0 00003          PZE     BITAB,,3          43
00047   0 00002 0 00003          PZE     BITAB,,2          44
00050   0 00003 0 00003          PZE     BITAB,,3          45
00051   0 00003 0 00003          PZE     BITAB,,3          46
00052   0 00004 0 00003          PZE     BITAB,,4          47
00053   0 00002 0 00003          PZE     BITAB,,2          50
00054   0 00003 0 00003          PZE     BITAB,,3          51
00055   0 00003 0 00003          PZE     BITAB,,3          52
```

```
00056  0 00004 0 00003          PZE     BITAB,,4            53
00057  0 00003 0 00003          PZE     BITAB,,3            54
00060  0 00004 0 00003          PZE     BITAB,,4            55
00061  0 00004 0 00003          PZE     BITAB,,4            56
00062  0 00005 0 00003          PZE     BITAB,,5            57
00063  0 00002 0 00003          PZE     BITAB,,2            60
00064  0 00003 0 00003          PZE     BITAB,,3            61
00065  0 00003 0 00003          PZE     BITAB,,3            62
00066  0 00004 0 00003          PZE     BITAB,,4            63
00067  0 00003 0 00003          PZE     BITAB,,3            64
00070  0 00004 0 00003          PZE     BITAB,,4            65
00071  0 00004 0 00003          PZE     BITAB,,4            66
00072  0 00005 0 00003          PZE     BITAB,,5            67
00073  0 00003 0 00003          PZE     BITAB,,3            70
00074  0 00004 0 00003          PZE     BITAB,,4            71
00075  0 00004 0 00003          PZE     BITAB,,4            72
00076  0 00005 0 00003          PZE     BITAB,,5            73
00077  0 00004 0 00003          PZE     BITAB,,4            74
00100  0 00005 0 00003          PZE     BITAB,,5            75
00101  0 00005 0 00003          PZE     BITAB,,5            76
00102  0 00006 0 00003          PZE     BITAB,,6            77
00103 -0114 06 0 00104   CNVRT  CAQ     AA,0,6
00104  000000000116      AA     VFD     24/0,12/BB
00105  200000000116      AA1    VFD     03/2,21/0,12/BB
00106  0 00000 0 00000   AA2    PZE
00107  0 00000 0 00000   AA3    PZE
00110  0 00000 0 00000   AA4    PZE
00111  0 00000 0 00000   AA5    PZE
00112  0 00000 0 00000   AA6    PZE
00113  0 00000 0 00000   AA7    PZE
00114  400000000116      AA8    VFD     03/4,21/0,12/BB
00115  600000000116      AA9    VFD     03/6,21/0,12/BB
00116  000000000130      BB     VFD     24/0,12/CC
00117  040000000130      BB1    VFD     06/4,18/0,12/CC
00120  0 00000 0 00000   BB2    PZE
00121  0 00000 0 00000   BB3    PZE
00122  0 00000 0 00000   BB4    PZE
00123  0 00000 0 00000   BB5    PZE
00124  0 00000 0 00000   BB6    PZE
00125  0 00000 0 00000   BB7    PZE
00126  100000000130      BB8    VFD     06/10,18/0,12/CC
00127  140000000130      BB9    VFD     06/14,18/0,12/CC
00130  000000000142      CC     VFD     24/0,12/DD
00131  010000000142      CC1    VFD     06/01,18/0,12/DD
00132  0 00000 0 00000   CC2    PZE
00133  0 00000 0 00000   CC3    PZE
00134  0 00000 0 00000   CC4    PZE
00135  0 00000 0 00000   CC5    PZE
00136  0 00000 0 00000   CC6    PZE
00137  0 00000 0 00000   CC7    PZE
00140  020000000142      CC8    VFD     06/2,18/0,12/DD
00141  030000000142             VFD     06/3,18/0,12/DD
00142  000000000154      DD     VFD     24/0,12/EE
```

```
00143  002000000154        DD1   VFD 09/2,15/0,12/EE
00144  0 00000 0 00000     DD2   PZE
00145  0 00000 0 00000     DD3   PZE
00146  0 00000 0 00000     DD4   PZE
00147  0 00000 0 00000     DD5   PZE
00150  0 00000 0 00000     DD6   PZE
00151  0 00000 0 00000     DD7   PZE
00152  004000000154        DD8   VFD 09/4,15/0,12/EE
00153  006000000154        DD9   VFD 09/6,15/0,12/EE
00154  000000000166        EE    VFD 24/0,12/FF
00155  000400000166        EE1   VFD 012/4,12/0,12/FF
00156  0 00000 0 00000     EE2   PZE
00157  0 00000 0 00000     EE3   PZE
00160  0 00000 0 00000     EE4   PZE
00161  0 00000 0 00000     EE5   PZE
00162  0 00000 0 00000     EE6   PZE
00163  0 00000 0 00000     EE7   PZE
00164  001000000166        EE8   VFD 012/10,12/0,12/FF
00165  001400000166        EE9   VFD 012/14,12/0,12/FF
00166  000000000000        FF    VFD 24/0,12/0
00167  000100000000        FF1   VFD 012/1,12/0,12/0
00170  0 00000 0 00000     FF2   PZE
00171  0 00000 0 00000     FF3   PZE
00172  0 00000 0 00000     FF4   PZE
00173  0 00000 0 00000     FF5   PZE
00174  0 00000 0 00000     FF6   PZE
00175  0 00000 0 00000     FF7   PZE
00176  000200000000        FF8   VFD 012/2,12/0,12/0
00177  000300000000        FF9   VFD 012/3,12/0,12/0
                         * COMMON BLOCK - REVISED  13 SEPTEMBER 19
                  77462          COMMON -1
                  77462   INDIC  COMMON 1
                  77461   ORDER  COMMON 1
                  77460   NWORD  COMMON 1
                  77457   DAT    COMMON 1
                  77456   LGTH   COMMON 1
                  77455   LATIS  COMMON 1
                  77454   NBITH  COMMON 1
                  77453   NBITL  COMMON 1
                  77452   NBIT1  COMMON 1
                  77451   NBIT   COMMON 1
                  77450   NSQ1   COMMON 1
                  77447   OPRMN  COMMON 1
                  77446   ATOMO  COMMON 1
                  77445   ATOM   COMMON 1
                  77444   ONED   COMMON 1
                  77443   D36    COMMON 1
                  77442   ATOOX  COMMON 10
                  77430   ATOX   COMMON 10
                  77416   SET    COMMON 10
                  77404   RANDM  COMMON 10
                  77372   DIFF   COMMON 10
                  77360   CONVI  COMMON 10
```

```
77310   DATA  COMMON 40
77240   MATA  COMMON 40
77170   UNIT  COMMON 40
77120   COMUN COMMON 40
77050   EQLS  COMMON 20
77024   SECTS COMMON 50
76742   MATAX COMMON 260
76336   DROWS COMMON 2100
72252   MROWS COMMON 2100
66166   INMAT COMMON 5400
53536   ATMS  COMMON 2000
47616   MACRO COMMON 7000
              END
00200 IS THE FIRST LOCATION NOT USED BY THIS PROGRAM
```

```
                00021          ENTRY   PTLVL
                00024          ENTRY   PRTIN
                00027          ENTRY   PTOUT
                00005          ENTRY   PAGE
                00010          ENTRY   SPACE
                00013          ENTRY   DBL
                00016          ENTRY   TRPL

  TRANSFER VECTOR
00000  746263303460      (STH)
00001  742631433460      (FIL)

  LINKAGE DIRECTOR
00002  000000000000
00003  476343654360


                               *              THIS SUBPROGRAM IS USED TO PRINT SPECIFIC COMMENTS IN
                               *              THE OUTPUT AS DESIRED
                               * ALSO USED FOR SPACING PRINTED OUTPUT.
00004  0 00000 0 00037          PZE PAGE1
00005  0500 00 0 00004   PAGE   CLA *-1
00006  0020 00 0 00103          TRA START
00007  0 00000 0 00045          PZE SPC1
00010  0500 00 0 00007   SPACE  CLA *-1
00011  0020 00 0 00103          TRA START
00012  0 00000 0 00031          PZE DBL1
00013  0500 00 0 00012   DBL    CLA *-1
00014  0020 00 0 00103          TRA START
00015  0 00000 0 00053          PZE TRPL1
00016  0500 00 0 00015   TRPL   CLA *-1
00017  0020 00 0 00103          TRA START
00020  0 00000 0 00061          PZE RMARK
00021  0500 00 0 00020   PTLVL  CLA *-1
00022  0020 00 0 00103          TRA START
00023  0 00000 0 00067          PZE SMARK
00024  0500 00 0 00023   PRTIN  CLA *-1
00025  0020 00 0 00103          TRA START
00026  0 00000 0 00075          PZE TMARK
00027  0500 00 0 00026   PTOUT  CLA *-1
00030  0020 00 0 00103          TRA START
00031  006060606060      DBL1   BCI 6,0
00032  606060606060
00033  606060606060
00034  606060606060
00035  606060606060
00036  606060606060
00037  016060606060      PAGE1  BCI 6,1
00040  606060606060
00041  606060606060
00042  606060606060
00043  606060606060
```

```
00044  606060606060
00045  606060606060        SPC1  BCI 6,
00046  606060606060
00047  606060606060
00050  606060606060
00051  606060606060
00052  606060606060
00053  406060606060        TRPL1 BCI 6,-
00054  606060606060
00055  606060606060
00056  606060606060
00057  606060606060
00060  606060606060
00061  004525666043        RMARK BCI 6,ONEW LEVEL OF HIERARCHY
00062  256525436046
00063  266030312551
00064  215123307060
00065  606060606060
00066  606060606060
00067  402346456351        SMARK BCI 6,-CONTROL PASSED TO SCTRL
00070  464360472162
00071  622524606346
00072  606223635143
00073  606060606060
00074  606060606060
00075  602346456351        TMARK BCI 6, CONTROL RETURNED TO LCTRL
00076  464360512563
00077  645145252460
00100  634660432363
00101  514360606060
00102  606060606060
00103   0621 00 0 00113    START STA LST1
00104  -0634 00 4 00125          SXD IR4,4
00105  -0500 00 0 00122          CAL TAPE2
00106   0074 00 4 00000          TSX $(STH),4
00107   0 00000 0 00123          PZE LELFT
00110  -0500 00 0 00124          CAL NUM
00111   0622 00 0 00116          STD TXH1
00112   0774 00 1 00000          AXT 0,1
00113   0560 00 1 00000    LST1  LDQ **,1
00114  -1 00000 0 00000          STR
00115   1 77777 1 00116          TXI *+1,1,-1
00116   3 00000 1 00113    TXH1  TXH LST1,1,**
00117   0074 00 4 00001          TSX $(FIL),4
00120  -0534 00 4 00125          LXD IR4,4
00121   0020 00 4 00001          TRA 1,4
00122   0 00002 0 00000    TAPE2 PZE 0,0,2
00123  740621063460        LELFT BCI 1,(6A6)
00124   0 77772 0 00000    NUM   PZE 0,0,-6
00125   0 00000 0 00000    IR4
                         * COMMON BLOCK - REVISED  13 SEPTEMBER 1961
                   77462         COMMON -1
                   77462   INDIC COMMON 1
```

```
77461   ORDER COMMON 1
77460   NWORD COMMON 1
77457   DAT   COMMON 1
77456   LGTH  COMMON 1
77455   LATIS COMMON 1
77454   NBITH COMMON 1
77453   NBITL COMMON 1
77452   NBITI COMMON 1
77451   NBIT  COMMON 1
77450   NSQI  COMMON 1
77447   OPRMN COMMON 1
77446   ATOMO COMMON 1
77445   ATOM  COMMON 1
77444   ONED  COMMON 1
77443   D36   COMMON 1
77442   ATOOX COMMON 10
77430   ATOX  COMMON 10
77416   SET   COMMON 10
77404   RANDM COMMON 10
77372   DIFF  COMMON 10
77360   CONVT COMMON 40
77310   DATA  COMMON 40
77240   MATA  COMMON 40
77170   UNIT  COMMON 40
77120   COMUN COMMON 40
77050   EQLS  COMMON 20
77024   SECTS COMMON 50
76742   MATAX COMMON 260
76336   DROWS COMMON 2100
72252   MROWS COMMON 2100
66166   INMAT COMMON 5400
53536   ATMS  COMMON 2000
47616   MACRO COMMON 7000
        END

00126 IS THE FIRST LOCATION NOT USED BY THIS PROGRAM


   NO ERROR IN ABOVE ASSEMBLY
* DATE AND TIME NOW      4/ 5 1518.8
```

```
                    * MODIFICATION OF LINE SPACING CONTROL IN FMT - 3/22/62
              00005       ENTRY PTLGR

  TRANSFER VECTOR
00000  746263303460     (STH)
00001  742631433460     (FIL)
00002  624721232560     SPACE

  LINKAGE DIRECTOR
00003  000000000000
00004  476343275160


00005 -0634 00 4 00715  PTLGR SXD IR4,4
                       *              PRELIMINARY
00006  0500 00 0 77461         CLA ORDER
00007  0622 00 0 00022         STD TXL1
00010  0622 00 0 00050         STD TXL2
00011  0500 00 0 77460         CLA NWORD
00012  0622 00 0 00037         STD TXL3
00013  0774 00 2 00002         AXT 2,2
00014 -0634 00 2 00716  START  SXD TWO,2
                       *              GENERATE TABLE OF NUMBERS ONE THROUGH ORDER
00015  0774 00 1 00001         AXT 1,1
00016  0500 00 0 77444         CLA ONED
00017  0622 00 1 00714         STD TABLE,1
00020  0400 00 0 77444         ADD ONED
00021  1 00001 1 00022         TXI *+1,1,1
00022 -3 00000 1 00017  TXL1   TXL *-3,1,**
                       *              MODIFY TABLE TO INDICATE THE CONSTITUENT ELEMENTS OF SET
00023  0774 00 2 00001         AXT 1,2
00024  0774 00 4 00001         AXT 1,4
00025  0774 00 1 00001  AXT1   AXT 1,1
00026  0560 00 2 77416         LDQ SET,2
00027  0162 00 0 00031         TQP *+2
00030  0020 00 0 00032         TRA *+2
00031  0600 00 4 00714  STORE  STZ TABLE,4
00032 -0773 00 0 00001         RQL 1
00033  1 00001 1 00034         TXI *+1,1,1
00034  1 00001 4 00035         TXI *+1,4,1
00035 -3 00044 1 00027         TXL STORE-2,1,36
00036  1 00001 2 00037         TXI *+1,2,1
00037 -3 00000 2 00025  TXL3   TXL AXT1,2,**
00040  0020 00 0 00041         TRA OUT
                       *              PRINT OUT MODIFIED TABLE.
00041 -0500 00 0 00071  OUT    CAL NN
00042  0074 00 4 00000         TSX $(STH),4
00043  0 00000 0 00073         PZE FMT
00044  0774 00 1 00001         AXT 1,1
00045  0560 00 1 00714  LST2   LDQ TABLE,1
00046 -1 00000 0 00000         STR
00047  1 00001 1 00050         TXI *+1,1,1
```

```
00050 -3 00000 1 00045   TXL2  TXL LST2,1,**
00051  0074 00 4 00001         TSX $(FIL),4
00052  0020 00 0 00053         TRA COMP
                       *               REPLACE SET WITH ITS COMPLEMENT IN ATOX.
00053 -0534 00 2 77460   COMP  LXD NWORD,2
00054 -0500 00 2 77416         CAL SET,2
00055  0760 00 0 00006         COM
00056 -0320 00 2 77430         ANA ATOX,2
00057  0602 00 2 77416         SLW SET,2
00060  2 00001 2 00054         TIX *-4,2,1
00061 -0534 00 2 00716         LXD TWO,2
00062  0074 00 4 00002         CALL SPACE
00063  1 00000 0 00065
00064  0 00076 0 00003
00065  0020 00 0 00066         TRA FINIS
                       *               IF BOTH SET AND ITS COMPLEMENT HAVE BEEN PRINTED,
                       *               RETURN TO LCTRL.
00066  2 00001 2 00014   FINIS TIX START,2,1
00067 -0534 00 4 00715         LXD IR4,4
00070  0020 00 4 00001         TRA 1,4
00071  0 00002 0 00000   NN    PZE 0,0,2
00072  063103346060            BCI 1,6I3)
00073  740130607303      FMT   BCI 1,(1H ,3
00714                          BES 400
00714  0 00000 0 00000   TABLE
00715  0 00000 0 00000   IR4
00716  0 00000 0 00000   TWO
                       * COMMON BLOCK - REVISED  13 SEPTEMBER 1961
                 77462         COMMON -1
                 77462   INDIC COMMON 1
                 77461   ORDER COMMON 1
                 77460   NWORD COMMON 1
                 77457   DAT   COMMON 1
                 77456   LGTH  COMMON 1
                 77455   LATIS COMMON 1
                 77454   NBITH COMMON 1
                 77453   NBITL COMMON 1
                 77452   NBIT1 COMMON 1
                 77451   NBIT  COMMON 1
                 77450   NSQ1  COMMON 1
                 77447   OPRMN COMMON 1
                 77446   ATOMO COMMON 1
                 77445   ATOM  COMMON 1
                 77444   ONED  COMMON 1
                 77443   D36   COMMON 1
                 77442   ATOOX COMMON 10
                 77430   ATOX  COMMON 10
                 77416   SET   COMMON 10
                 77404   RANDM COMMON 10
                 77372   DIFF  COMMON 10
                 77360   CONVT COMMON 40
                 77310   DATA  COMMON 40
                 77240   MATA  COMMON 40
```

```
77170   UNIT  COMMON 40
77120   COMUN COMMON 40
77050   EQLS  COMMON 20
77024   SECTS COMMON 50
76742   MATAX COMMON 260
76336   DROWS COMMON 2100
72252   MROWS COMMON 2100
66166   INMAT COMMON 5400
53536   ATMS  COMMON 2000
47616   MACRO COMMON 7000
        END

00717 IS THE FIRST LOCATION NOT USED BY THIS PROGRAM
```

```
                00004          ENTRY  PTCLR

  TRANSFER VECTOR
00000  746263303460      (STH)
00001  742631433460      (FIL)

  LINKAGE DIRECTOR
00002  000000000000
00003  476323435160


00004 -0634 00 4 00123   PTCLR SXD IR4,4
                        *              PRELIMINARY
00005  0774 00 2 00002          AXT 2,2
00006 -0634 00 2 00122   START SXD TWO,2
                        *              GENERATE TABLE OF NUMBERS ONE THROUGH 36
00007  0774 00 1 00044          AXT 36,1
00010  0500 00 1 77360          CLA CONVT,1
00011  0622 00 1 00121          STD INTGR,1
00012  2 00001 1 00010          TIX *-2,1,1
                        *              MODIFY TABLE TO INDICATE THE CONSTITUENT ELEMENTS OF SET
00013  0774 00 1 00001          AXT 1,1
00014  0560 00 0 77416          LDQ SET
00015  0162 00 0 00017          TQP *+2
00016  0020 00 0 00020          TRA *+2
00017  0600 00 1 00121   STORE  STZ INTGR,1
00020 -0773 00 0 00001          RQL 1
00021  1 00001 1 00022          TXI *+1,1,1
00022 -3 00044 1 00015          TXL STORE-2,1,36
00023  0020 00 0 00024          TRA OUT
                        *              PRINT OUT MODIFIED TABLE
00024 -0500 00 0 00047   OUT    CAL NN
00025  0074 00 4 00000          TSX $(STH),4
00026  0 00000 0 00050          PZE FMT
00027  0774 00 1 00001          AXT 1,1
00030  0560 00 1 00121   LST2   LDQ INTGR,1
00031 -1 00000 0 00000          STR
00032  1 00001 1 00033          TXI *+1,1,1
00033 -3 00044 1 00030   TXL2   TXL LST2,1,36
00034  0074 00 4 00001          TSX $(FIL),4
00035  0020 00 0 00036          TRA COMP
                        *              REPLACE SET WITH ITS COMPLEMENT IN ATOM
00036 -0500 00 0 77416   COMP   CAL SET
00037  0760 00 0 00006          COM
00040 -0320 00 0 77445          ANA ATOM
00041  0602 00 0 77416          SLW SET
00042 -0534 00 2 00122          LXD TWO,2
00043  0020 00 0 00044          TRA FINIS
                        *              IF BOTH SET AND ITS COMPLEMENT HAVE BEEN PRINTED,
                        *              RETURN TO SCTRL.
00044  2 00001 2 00006   FINIS  TIX START,2,1
00045 -0534 00 4 00123          LXD IR4,4
```

```
00046  0020 00 4 00001          TRA 1,4
00047  0 00002 0 00000    NN    PZE 0,0,2
00050  740306310334       FMT   BCI 1,(36I3)
00121                            BES 40
00121  0 00000 0 00000    INTGR
00122  0 00000 0 00000    TWO
00123  0 00000 0 00000    IR4
                        * COMMON BLOCK - REVISED  13 SEPTEMBER 1961
                  77462          COMMON -1
                  77462    INDIC COMMON 1
                  77461    ORDER COMMON 1
                  77460    NWORD COMMON 1
                  77457    DAT   COMMON 1
                  77456    LGTH  COMMON 1
                  77455    LATIS COMMON 1
                  77454    NBITH COMMON 1
                  77453    NBITL COMMON 1
                  77452    NBIT1 COMMON 1
                  77451    NBIT  COMMON 1
                  77450    NSQ1  COMMON 1
                  77447    OPRMN COMMON 1
                  77446    ATOMO COMMON 1
                  77445    ATOM  COMMON 1
                  77444    ONED  COMMON 1
                  77443    D36   COMMON 1
                  77442    ATOOX COMMON 10
                  77430    ATOX  COMMON 10
                  77416    SET   COMMON 10
                  77404    RANDM COMMON 10
                  77372    DIFF  COMMON 10
                  77360    CONVT COMMON 40
                  77310    DATA  COMMON 40
                  77240    MATA  COMMON 40
                  77170    UNIT  COMMON 40
                  77120    COMUN COMMON 40
                  77050    EQLS  COMMON 20
                  77024    SECTS COMMON 50
                  76742    MATAX COMMON 260
                  76336    DROWS COMMON 2100
                  72252    MROWS COMMON 2100
                  66166    INMAT COMMON 5400
                  53536    ATMS  COMMON 2000
                  47616    MACRO COMMON 7000
                           END

00124 IS THE FIRST LOCATION NOT USED BY THIS PROGRAM


   NO ERROR IN ABOVE ASSEMBLY
* DATE AND TIME NOW      4/12 1741.6
```

Date Due

~~JUN 1 5 1967~~ JUN 0 1972			
~~JUL 1 0 1967~~		SEP 2 0 1972	
~~SEP 2 7 1967~~			
~~OCT 2 5 1967~~			
~~NOV 2 1 1967~~		JUN 2 5 1973	
~~FEB 5 1968~~			
~~FEB 2 0 1968~~			
~~MAR 2 0 1968~~			
~~APR 1 2 1968~~			
~~APR 2 6 1968~~			
~~MAY 1 0 1968~~			
~~MAY 2 4 1968~~			
~~AUG 2 1 1968~~			
~~JAN 8 1969~~			
~~OCT 0 1969~~			
Oct. 24, 69			
MAY 2 4 1971			
NOV 8 1971			
JAN 4 1972			

Demco-293