AN ALGORITHM FOR TANT SYNTHESIS AND ITS SEQUENTIAL APPLICATION

By

JOHN MICHAEL ACKEN

Bachelor of Science in Electrical Engineering

Oklahoma State University

Stillwater, Oklahoma

1976

Submitted to the Faculty of the Graduate College
of the Oklahoma State University
in partial fulfillment of the requirements
for the Degree of
MASTER OF SCIENCE
May, 1978

AN ALGORITHM FOR TANT SYNTHESIS AND

ITS SEQUENTIAL APPLICATION

Thesis Approved:

James R. Rowland

Thesis Adviser

Edward L. Shreve

Charles M. Bacon

Norman N Durham

Dean of Graduate College

ACKNOWLEDGMENT

I wish to express my sincere appreciation to Dr. James R. Rowland for spending many hours in valuable guidance on this thesis research. I wish to thank the other members of my committee, Dr. Charles M. Bacon and Dr. Edward L. Shreve.

I would like to thank the students of ELEN 4253 for their many computer runs.

I wish to thank Velda Davis for typing my thesis.

Finally, I wish to thank my parents for their encouragement, and my family for their love and moral support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Digital logic circuit designers have many computer-aided design (CAD) packages at their disposal. That most current computers were designed by computers proves the practicality of such CAD programs. On a smaller scale, many algorithms and associated software packages solve the gate minimization problem for combinational logic design or select the optimum memory device from among given candidates to yield gate minimization for sequential problems. Digital logic designers choose a memory device, find the resulting logic equations, and minimize the combinational portion of the sequential feedback circuit. Another memory device is utilized only after dissatisfaction with the first has resulted. However, a CAD program can test all types of memory devices and yield a minimum combinational circuit for each device. Thus, the designer can be assured of finding the optimal design for given constraints. The problem investigated in this thesis is the design of optimal sequential circuits using NAND gates for the combinational portion. A list of several familiar memory devices is specified for the sequential portion of the problem. The number of gates and, secondarily, the number of inputs to those gates are minimized for each type of device.

Background

Historically, the problem of finding a minimum logic circuit has been divided into two parts: the combinational logic design problem and the memory selection problem. The usual criteria for minimization are propagation time and package count. In this thesis, the combinational problem under consideration is the design of optimal TANT networks, i.e., three-level NAND-gate circuits with true inputs only (1). Used in the examples as a convenient tool for visualization in combinational problems with few inputs is the circuit-action Karnaugh map. In the sequential problem, some of the memories to choose from are the D flip-flop, RS flip-flop, JK flip-flop, T (or toggle) flip-flop, and, of course, any clocked version of any of these.

The TANT restriction of three levels or less is used to minimize propagation time since any logic function can be realized in three levels, and propagation time is a function of the number of levels through which the signal must pass. Solving the minimum package count is not as straightforward and, in fact, is not the consideration when logic component (chip) design is considered. Therefore, for ease in the first case and correctness in the second, rather than minimum package count, minimum-gate and minimum-input realizations will be the goal. The solution of NAND circuits is easily used in an analogous manner to solve for NOR circuits (1-3). The uncomplemented requirement is typical for inputs to a logic circuit. However, since complemented inputs will be available as outputs from memory devices, this thesis actually deals with a modified TANT problem.

The majority of previous work on the NAND/NOR synthesis problem has been four categories: the double-complement technique (4-6),

decomposition (7-9), the head-tail approach (10-12), the map factoring approach (13-15). Although there has been some work in using integer programming (16-19), it has been of a much more general nature and not explicitly advantageous in the solution of the TANT problem. There also has been work done using exhaustive search (20, 21), but this method was used to generate a table of solutions for all possible small number of input problems and is not very practical for situations involving many input variables.

The double-complement technique, which is solving the two-level AND/OR problem by essentially a Quine-McCluskey approach and then inverting inputs to achieve the necessary negated inputs, is the most straightforward technique because of the equivalence of any two-level NAND structure and its corresponding AND/OR circuit (22). Although straightforward, this procedure usually does not yield an optimal (i.e., minimum-gate, minimum-input) realization for a given problem. A double-complement method was developed early by Gimpel (4) which involved certain extensions involving a prime implicant cover and closure (CC) table. Hohulin and Muroga (5) reported several alternative methods for solving the CC-table in Gimpel's algorithm suitable for computer processing, and these methods are implemented as computer programs. Ellis (6) extended the double-complement technique to NOR gates as well as NAND. Davidson (7) used decomposition, in which a first cut of the double-complement method is made, followed by back-tracking for improvements while considering fan-in, fan-out, and level constraints. Dietmeyer and Su (8) used decomposition with factoring, assuming complemented inputs available and no level constraint. Schneider and Dietmeyer (9) extended decomposition solutions for multiple

outputs. Koh's (10) head-tail approach is based on first obtaining the essential prime implicants and then generating useful prime implicants. By various methods groupings are made to reduce the NAND circuit. Then a table analogous to the CC table is solved. Chakrabarti, Choudhury, and Basu (11) proposed an early head-tail investigation based upon analysis of production at the second-level gate outputs of the complement of the function as well as the desired minterms. Frackowiak (12) presented two approaches, one yielding a quasi-minimal hazardless solution and the other yielding a strictly minimal solution. Maley and Earle (13) initiated map factoring for finding gates from prime implicants by a method analogous to circling n-cubes on a Karnaugh map; hence, the name map factoring is appropriate. Eisenberg's (14) work was extended by Torng (15) to yield a systematic realization procedure for NAND networks by alternately inhibiting 0 and 1 entires in the Karnaugh map as the number of switching levels increases. After this primitive realization is obtained, a level reduction procedure is applied to reduce the number of levels to at most three. A transform technique is used to complete the design procedure. These various procedures provide a wide range of solutions to the TANT problem, with most of them having been implemented on digital computers. None of these procedures attacks the problem of associated sequential problems. Also, none of the publications give very extensive analysis of memory or CPU time requirements for computer implementations.

An Example

To demonstrate the TANT solution of a particular problem using the double-complement technique and subsequent improvements until an optimum

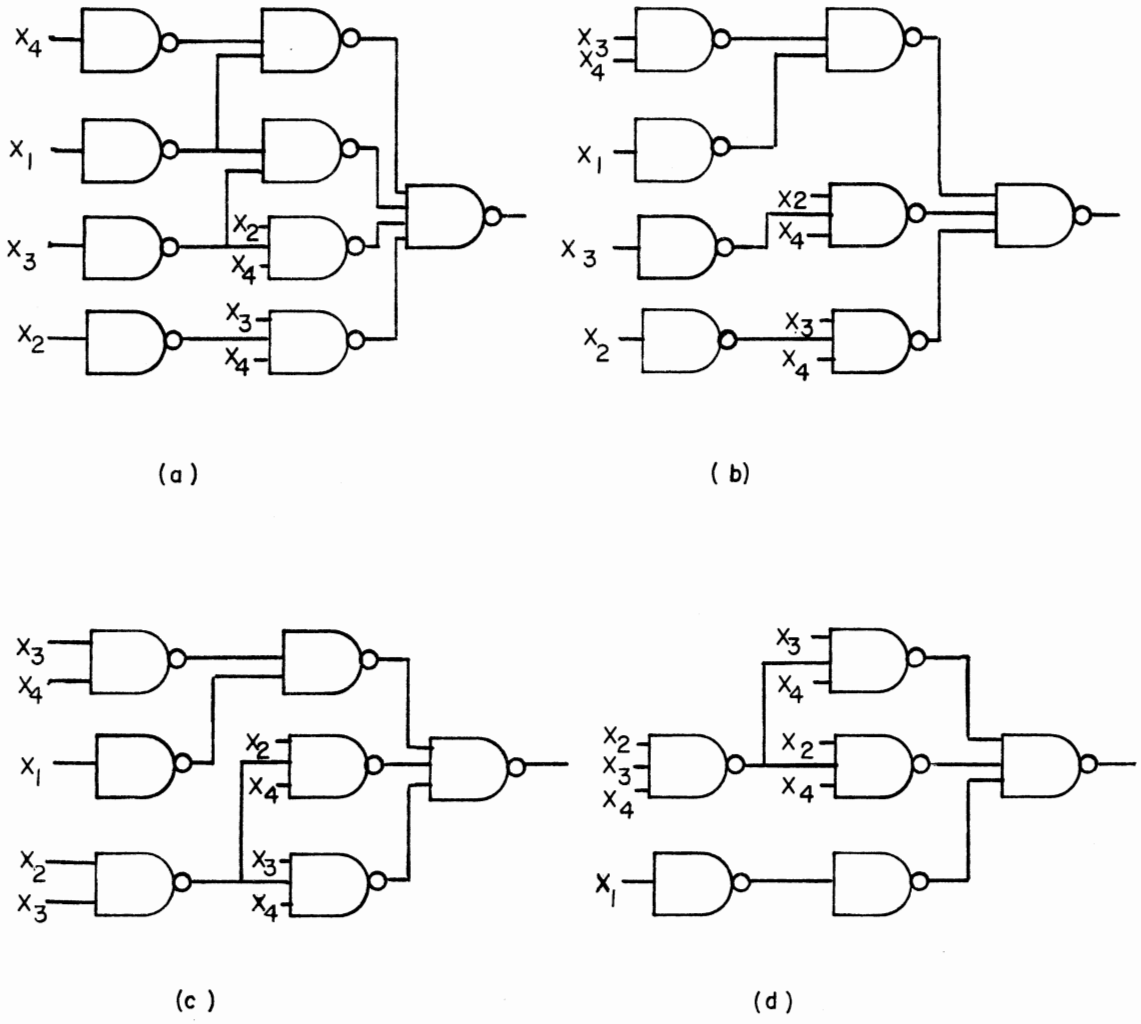


Figure 1. Example NAND Gate Circuit Improvements

solution is found, consider the function

$$F(x_1, x_2, x_3, x_4) = \Sigma m\,(0, 2, 3, 4, 5, 6, 11, 13). \tag{1}$$

The solution corresponding to the optimal AND/OR network is

$$F(x_1, x_2, x_3, x_4) = \bar{x}_1\bar{x}_3 + \bar{x}_2x_3x_4 + x_2\bar{x}_3x_4 + \bar{x}_1\bar{x}_4 \tag{2}$$

which is shown as a circuit using NAND gates in Figure 1(a). Note that single input NAND gates are used as inverters to get the complemented inputs. This realization has 9 gates and 18 inputs to those gates. Now using the first and fourth terms of the right-hand side of (2) we see that

$$\bar{x}_1\bar{x}_3 + \bar{x}_1\bar{x}_4 = \bar{x}_1(\bar{x}_3 + \bar{x}_4), \tag{3}$$

and substituting (3) into (2) the result is

$$F(x_1, x_2, x_3, x_4) = \bar{x}_1(\bar{x}_3 + \bar{x}_4) + \bar{x}_2x_3x_4 + x_2\bar{x}_3x_4. \tag{4}$$

This realization is shown in Figure 1 (b). It has only 8 gates and 16 inputs. Further reduction is seen by ORing zero to the second and third terms of (4), i.e.,

$$\begin{aligned}\bar{x}_2x_3x_4 + x_2\bar{x}_3x_4 &= \bar{x}_2x_3x_4 + \bar{x}_2x_2x_4 + x_2\bar{x}_3x_4 + x_3\bar{x}_3x_4 \\ &= (\bar{x}_2 + \bar{x}_3)(x_2x_4 + x_3x_4) \\ &= (\overline{x_2x_3})(x_2x_4 + x_3x_4).\end{aligned} \tag{5}$$

Now substituing (5) into (4) yields

$$F(x_1, x_2, x_3, x_4) = \bar{x}_1(\bar{x}_3 + \bar{x}_4) + (\overline{x_2x_3})(x_2x_4 + x_3x_4) \tag{6}$$

which is shown as a NAND circuit in Figure 1(c). This circuit has only 7

gates but still 16 inputs. With some more manipulations, the result is

$$F(x_1, x_2, x_3, x_4) = (\overline{x_2x_3x_4})(x_3x_4 + x_2x_4 + \overline{x}_1) \qquad (7)$$

and this result is shown in Figure 1 (d). This is the optimal TANT network for this problem with only 6 gates and 15 inputs. This is the same solution as found by Torng (15) on page 125 by using his level-reduction and gate-reduction technique. Also, this is the solution found using the algorithm developed later in this thesis. This example demonstrates the straightforward initial solution using double-complement method and that the result was not the optimal TANT circuit. With some algebraic manipulations, the optimal solution was found and matched the result using level-reduction gate-reduction technique's solution. This same example will be treated in detail using this thesis' algorithm in the next chapter.

Problem Formulation and Approach

The problem to be investigated in this thesis is the TANT problem in conjunction with memory. The combinational portion of the problem uses three-level NAND-gate circuits with only true externally supplied inputs and both true and complemented inputs available from the memory devices. Also, the combinational portion must cover multiple output cases. The memory, or sequential, portion simply tries different flip-flops and solves the related combinational problem. This combined effort of both the combinational and sequential problem does not appear in the literature.

The algorithm was initially programmed to simply solve the TANT problem with true inputs only. The program was then exercised with many

large examples with a spinoff effort to decide whether minimum gate was a prerequisite for a circuit to have minimum inputs. The program was then exercised with different implementations of arbitrary steps to determine the best on the basis of computer time, core storage requirements, and ease to user. Finally, the TANT synthesis algorithm was matched with the flip-flop selection and implemented as a computer-aided design package.

Thesis Outline

Following this introductory chapter, the TANT synthesis algorithm is rigorously developed and described in detail in Chapter II. The CAD package for the TANT problem, along with several large examples, is presented in Chapter III. Chapter IV describes the sequential problem, along with presenting a key example. Discussion of extensions, improvements, and conclusions are presented in Chapter V.

CHAPTER II

SYNTHESIS OF OPTIMAL TANT NETWORKS

This chapter presents and proves the optimality of an algorithm to solve the TANT problem. A firm theoretical basis is presented in the form of theorems and rigorous proofs, followed by a detailed discussion of the synthesis algorithm. The underlying concept was first worked on by Layton (23) beginning in 1973. The early concepts have been formalized in the form of theorems and some of his third-level gate selections have been modified to allow a form of backtracking. Also, a very extensive gate and input reduction scheme has been added onto the end of the algorithm. One of his examples is used, while two new examples have been added to demonstrate the improvements.

Basic Theorems

This section presents fundamental theorems to be used as a basis for the TANT network synthesis procedure. Notationally, the first-level gate is defined as the gate from which the output is obtained. Second-level gates feed first-level gates and third-level gates feed second-level gates. Additionally, no feedback is allowed whereby lower-level gates feed higher-level gates. Theorems 1 through 4 provide for second-level and third-level gate candidate generation. The completeness of an iterative procedure based on these first four theorems is guaranteed by Theorem 5. Theorems 6 and 7 describe a necessary selection of certain

second-level gate candidates for the optimal network. A key network reduction is indicated by Theorem 8, and Theorems 9 and 10 minimize gate inputs. Finally, Theorem 11 shows that an optimal TANT network is obtained.

Theorem 1

Each 1-set containing the primary cube is a candidate for a second-level gate in a three-level NAND network having only true inputs.

Proof:

Let $F(x_1, x_2, \ldots, x_n)$ be some Boolean switching function with n inputs defined as

$$F(x_1, x_2, \ldots, x_n) = Q + \sum_{j=1}^{m} S_j \tag{8}$$

where m is the number of 1-set terms containing only true inputs, S_j is the product of true inputs for the j^{th} 1-set, and Q is some function covering the remaining 1-sets. Using the standard involution theorem (2) gives

$$F(x_1, x_2, \ldots, x_n) = \overline{\overline{Q + \sum_{j=1}^{m} S_j}} \tag{9}$$

Applying De Morgan's Theorem (2) yields

$$F(x_1, x_2, \ldots, x_n) = \overline{\overline{Q} \cdot \prod_{j=1}^{m} \overline{S_j}} \tag{10}$$

which can be recognized as a two-level NAND network with S_j as inputs to the j^{th} gate at the second level. Moreover, the m outputs from these second-level gates and $\overline{Q}$ are inputs to the first level.

The primary cube (22) is the 0-cube having the set of coordinates (111...1), i.e., all of the true inputs. Thus, any N-cube with only true inputs must contain the primary cube.

Theorem 2

Each 0-set containing the primary cube is a candidate for a third-level gate in a TANT network.

Proof:

Let $F(x_1, x_2, \ldots, x_n)$ be defined as

$$F(x_1, x_2, \ldots, x_n) = R \cdot \prod_{i=1}^{\ell} \left[\sum_{k=1}^{p_i} \bar{x}_{\xi_k} \right] \tag{11}$$

where there are ℓ 0-sets composed of the sums of p_i complemented inputs, $\bar{x}_{\xi_k}$ denotes the k^{th} complemented input in the i^{th} sum, and R is some function covering the remaining 0-sets. For example, one such function might be $(\bar{x}_1 + x_2)\ (\bar{x}_1 + \bar{x}_3)\ (\bar{x}_2 + \bar{x}_3 + \bar{x}_4)\ \bar{x}_5$, where $R = \bar{x}_1 + x_2$. Invoking involution and De Morgan's theorems gives

$$F(x_1, x_2, \ldots, x_n) = \overline{R \cdot \prod_{i=1}^{\ell} \left[\sum_{k=1}^{p_i} \bar{x}_{\xi_k} \right]} \tag{12}$$

$$F(x_1, x_2, \ldots, x_n) = \overline{R \cdot \prod_{i=1}^{\ell} \left[\overline{\prod_{k=1}^{p_i} x_{\xi_k}} \right]} \tag{13}$$

Therefore, the inputs to the third-level gates are the p_i true inputs.

Theorem 3

For a three-level NAND network, the input combinations to a third-level gate place "don't cares" in the 1-sets being considered as

candidates for second-level gates.

Proof:

Let the input set to the combinational network be represented by*

$$T_{3rd} = \sum_{i=1}^{2^n} t_i \tag{14}$$

where n is the number of input variables, t_i is the i^{th} combination of the input variables, and Σ denotes the logical sum of the t_i. The k applied input combinations which yield a "1" and 2^n-k which yield a "0" at the output of the third-level gate may be expressed as

$$T_{3rd} = \sum_{i=1}^{k} t_i + \sum_{i=k+1}^{2^n} t_i \tag{15}$$

Since the output of this third-level gate can be an input to the second level, the total number of inputs to the second level is n+1. The resulting 2^{n+1} combinations of inputs to the second level may be expressed as

$$T_{2nd} = \sum_{i=1}^{2^{n+1}} t_i \tag{16}$$

With Z_{3rd} as the output of the third-level gate,

$$T_{2nd} = \overline{Z}_{3rd} \cdot T_{3rd} + Z_{3rd} \cdot T_{3rd} \tag{17}$$

---

*The summation upper limit 2^n indicates only that there are 2^n distinct values which i can assume over the entire summation. No relation between the binary value of the particular input t_i and i is implied.

Substituting (8) into (10) and simplifying yields

$$T_{2nd} = \bar{Z}_{3rd} \sum_{i=1}^{k} t_i + \bar{Z}_{3rd} \sum_{i=k+1}^{2^n} t_i + Z_{3rd} \sum_{i=1}^{k} t_i + Z_{3rd} \sum_{i=k+1}^{2^n} t_i \tag{18}$$

Observe that $\bar{Z}_{3rd}$ is zero for the $\sum_{i=1}^{k} t_i$ input combinations and Z_{3rd} is zero for the $\sum_{i=k+1}^{2^n} t_i$ input combinations. Therefore, the inputs corresponding to the first and fourth terms in (18) are never applied and may be considered as "don't cares" for any second-level gate into which Z_{3rd} feeds.

Theorem 4

If some second-level gate candidates in a three-level NAND network have been determined, then the corresponding 1-set input combinations may be treated as "don't cares" in the O-sets being considered as candidates for third-level gates.

Proof:

If q members of the 1-set are covered by the second-level gate candidates already determined, then the output of the network will be "1" regardless of whether the remaining switching would result in an output of "1" or "0" for the q members of the 1-set previously covered.

Theorem 5

If a necessary third-level gate candidate is determined but discarded in favor of another candidate (or candidates), then it will be

regenerated subsequently for 0-set coverage at the third level.

Proof:

Suppose a third-level gate candidate is generated but not needed as a "don't care" (Theorem 3) for particular 1-set input combinations being considered for second-level gate candidates. Even though discarded at this stage of generation, the third-level gate candidate may be required subsequently as a "don't care" to feed a second-level gate candidate covering yet another 1-set. Therefore, to cover this other 1-set with a second-level gate, it will be necessary to regenerate the given third-level gate candidate. Furthermore, it is entirely possible that an improved third-level gate candidate having fewer inputs (and containing as a subset the previously discarded gate candidate) will be generated. In the worst case, the discarded gate candidate itself will be regenerated.

Theorem 6

The last second-level gate candidate generated must be selected for the TANT network.

Proof:

Since second-level gate candidates are generated to cover members of the 1-set (Theorem 1), each new candidate generated must cover at least one member of the 1-set not covered by candidates generated previously. The generation of these second-level gate candidates ceases when all members of the 1-set have been covered. Therefore, the last candidate generated must be selected because it covers at least one member of the 1-set not covered by any other candidate.

Theorem 7

If the first stage of generation of second-level gate candidates yields only one candidate, then that candidate must be selected for the TANT network.

Proof:

At least one member of the 1-set covered by the sole second-level gate candidate must be used as a "don't care" (Theorem 4) for the subsequent generation of one or more third-level gate candidates. By Theorem 2, the resulting third-level gate candidate(s) prevent any subsequent second-level gate candidates from covering the particular member(s) of the 1-set being used in Theorem 4.

Theorem 8

If a third-level gate candidate feeds only one second-level gate candidate as the sole input to that gate candidate, then both of these gate candidates may be discarded and the inputs to the original third-level gate candidate fed directly into the first-level gate.

Proof:

Let $F(x_1, x_2, \ldots, x_n)$ be defined as

$$F(x_1, x_2, \ldots, x_n) = \overline{\overline{t_{3rd}} P} \tag{19}$$

where t_{3rd} is the input combination to the third-level gate under consideration and P is some function that covers the remainder of the members of the 1-set. Using the involution theorem yields

$$F(x_1, x_2, \ldots, x_n) = \overline{t_{3rd}\, P} \tag{20}$$

which corresponds to t_{3rd} being fed directly into the first-level gate.

Theorem 9

If any input x_ζ is fed to a particular third-level gate and to each of the second-level gates which that gate feeds, then x_ζ may be removed, as an unnecessary input, from the third-level gate.

Proof:

Let the switching function $F(x_1, x_2, \ldots, x_n)$ be represented by

$$F(x_1, x_2, \ldots, x_n) = \overline{\left(\overline{R_1 x_\zeta \overline{(x_\zeta x_\eta)}}\right)\left(\overline{R_2 x_\zeta \overline{(x_\zeta x_\eta)}}\right)} \tag{21}$$

where x_ζ and x_η are inputs to a third-level gate which feeds two second-level gates. In addition to this input, R_1 feeds one of the second-level gates, R_2 feeds the other, and x_ζ feeds them both directly, where R_1 and R_2 are arbitrary. Using De Morgan's theorem gives

$$F(x_1, x_2, \ldots, x_n) = R_1 x_\zeta(\bar{x}_\zeta + \bar{x}_\eta) + R_2 x_\zeta(\bar{x}_\zeta + \bar{x}_\eta) \tag{22}$$

which may further be expressed as

$$F(x_1, x_2, \ldots, x_n) = R_1 x_\zeta \bar{x}_\zeta + R_1 x_\zeta \bar{x}_\eta + R_2 x_\zeta \bar{x}_\zeta + R_2 x_\zeta \bar{x}_\eta \tag{23}$$

Observe that the first and third terms in (23) are zero and that the remaining terms are due to the appearance of x_ζ as a direct input to the second-level gates. Therefore, the x_ζ input may be omitted from the third-level gate.

Theorem 10

To obtain an optimal network, any inputs which do not affect the output must be removed.

Proof:

If an input is present in Network A and not in Network B with all gates and other inputs being the same, then Network B obviously has fewer inputs. Provided that the two networks have identical outputs under all input conditions, Network A cannot be the optimal network.

Theorem 11

Among the networks generated by using Theorems 1 through 10, there exists an optimal network.

Proof:

Theorems 5 and 6 guarantee that all of the necessary gate candidates have been generated. Using these candidates to cover the 1-set of the desired switching function and applying Theorem 8 yields a minimum number of gates. The fewest number of inputs to these gates are determined by Theorems 9 and 10. Therefore, these exhaustive searches on covering and input reduction yield an optimal network.

The Network Synthesis Procedure

A detailed description of the TANT network synthesis procedure based upon the theorems of the previous section is presented here. A flow chart is given in Figure 2, and a detailed explanation of each step is given below.

Step 1

Determine whether the primary cube contains a "1" or a "0". If the primary cube contains a "1", then go to Step 2. If the primary cube contains a "0", go to Step 4.

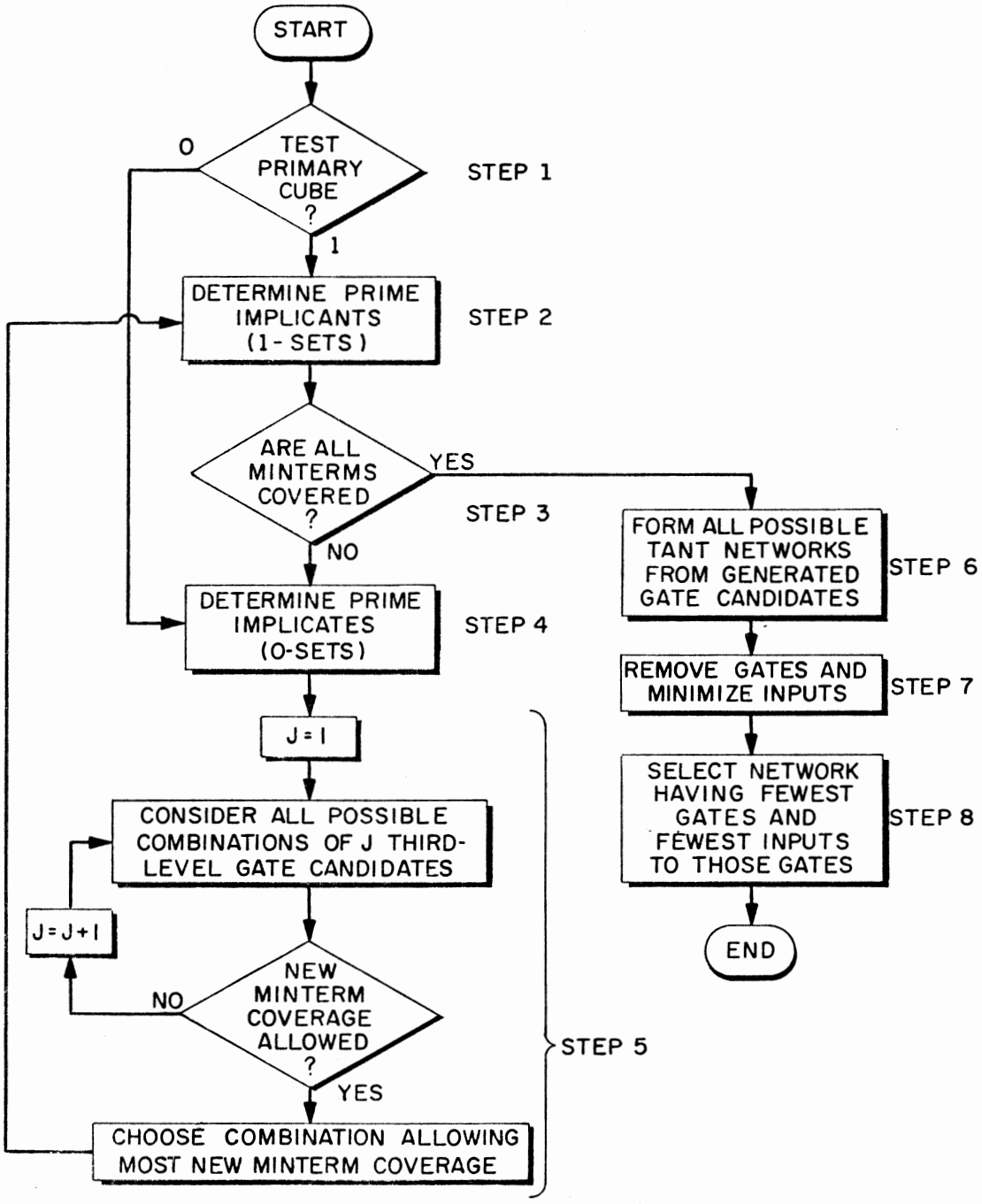


Figure 2. Flow Chart of the TANT Network Synthesis Procedure

Step 2

Determine the prime implicants (largest 1-sets) not requiring complemented inputs. This step uses Theorem 1 to generate second-level gate candidates by forming, with true inputs only, the largest groups of minterms (1-set members).*

Step 3

Test for coverage of the minterms for the desired swithcing function. If all of the minterms are covered, then go to Step 6; otherwise, more gate candidates must be generated, and, therefore, go to Step 4.

Step 4

Using all minterms that have been covered previously as "don't cares" by Theorem 4, determine the prime implicates (largest 0-sets) not requiring complemented inputs. Thus, by Theorem 2, third-level gate candidates are generated from the largest groups of maxterms (0-set members).

Step 5

This step determines which third-level gate candidates to retain. Test whether each gate candidate, considered singly, creates suitably placed "don't cares" (by Theorem 3) to allow additional second-level gate candidates to be generated that cover at least one minterm not

---

*Particularly for TANT networks having relatively few input variables, the Karnaugh map is a convenient tool for forming these largest groups by circling on the minterms in Step 2 and maxterms in Step 4. A computer program should be considered for handling networks with large numbers of input variables.

already covered. If none allow another second-level gate candidate to be generated, try all possible combinations of two, then all combinations of three, etc., until at least one new second-level gate candidate can be generated. When at least one new minterm can be covered by this procedure, select the combination of third-level gate candidates that allows the most new minterms to be covered. For example, let third-level gate candidates considered singly or in combinations of two permit no new second-level gate generation. Furthermore, suppose two groups of these third-level gate candidates considered in combinations of three do allow new second-level gate generations. Therefore, the group consisting of a combination of three third-level gate candidates which yields the second level gate candidate(s) covering the most new minterms retained. By Theorem 5, the discarded combination, if needed for the optimal network, will be generated subsequently. Consider all the implicates covered by these third-level gate candidates to be "don't cares" for subsequent gate candidate generation. Return to Step 2.

Step 6

Create all possible networks that cover the desired switching function by using the second-level and third-level gate candidates generated. Use Theorems 6 and 7, and then form a covering and closure table.

Step 7

Use Theorem 8 to eliminate third-level gates which solely feed second-level gates by feeding the third-level gate inputs directly to the first-level gate. Use Theorems 9 and 10 to eliminate the unnecessary inputs for the remaining second-level and third-level gates.

Step 8

Determine from among the several resulting networks the one(s) having a minimum number of gates and a minimum number of inputs to those gates. By Theorem 11, an optimal network is guaranteed.

The optimality of the resulting TANT network is based upon the procedure's adherence to the theorems of the previous section. Although more efficient steps might possibly be appropriate in selected cases, the synthesis procedure presented here does yield an optimal TANT network. Specifically, the discarding of third-level gate candidates in Step 5 may be avoided by a parallel storage and the subsequent consideration of all possible candidates. The particular selections in this step of the procedure yield an effective solution without unduly large amounts of storage. The directness of this procedure is demonstrated in the following examples.

Examples

Three examples are presented in this section to illustrate the developed network synthesis procedure. The first example appeared in the introduction and is included here for comparison as well as a detailed explanation of the steps of the present algorithm. Example 2 demonstrates the regeneration principle of Step 5 (Theorem 5) and gate selection procedures of Step 6, and Example 3 shows the details of input reduction in Step 7 (Theorems 9 and 10).

Example 1

It is required that an optimal TANT network be designed to realize the combinational switching function given by

$$F(x_1, x_2, x_3, x_4) = \Sigma m(0, 1, 2, 3, 4, 5, 6, 11, 13) \qquad (24)$$

The minterms in (24) are indicated by 1's in the Karnaugh map of Figure 3(a), and 0's are shown in all other map locations. In Step 1 of the synthesis procedure, the primary cube contains a 0, which directs the algorithm to Step 4. Two prime implicates are identified and labeled as T_1 and T_2 in Step 4. In Step 5, results are shown in Figures 3(b) and 3(c) for T_1 and T_2, respectively, by replacing the appropriate 0's by d's, allowing the tentative formation of prime implicants. Since the new 1-set coverage for the prime implicants obtained by using T_1 is greater than T_2, only T_1 is retained. Proceeding to Step 2, the two prime implicants of Figure 3(b) become second-level gate candidates and are labeled S_1 and S_2. In Step 3, Minterms 0, 1, 2, 4, and 6 are not yet covered, and the procedure returns to Step 4. Additional "don't cares" (d's) due to S_1 and S_2 are shown in Figure 3(d), along with the d's due to T_1 determined earlier. The prime implicate labeled T_3 covers the remaining five 0's (Step 4). Since only one prime implicate is generated, proceed past Step 5 directly to Step 2. Replacing the entries in T_3 by d's in Figure 3(e) yields the second-level gate candidate labeled S_3, which is composed of the entire map. Since all minterms are covered (Step 3), proceed to Step 6 to form possible TANT networks. In the second-level, S_3 must be selected, since it was the last second-level gate candidate. Minterms 0 through 6 are covered by S_3, as indicated by checks (✓) in Figure 3(a). Third-level gates whose outputs feed into the NAND-gate with output S_3 have outputs T_1 and T_3. T_3 covers Maxterms 8, 9, 10, 12, 14, and 15. Maxterms 7 and 15 are covered by T_1. Additional second-level gates are needed to cover Minterms 11 and 13. The gate candidate with output S_1 covers Minterms 3 and 11, and

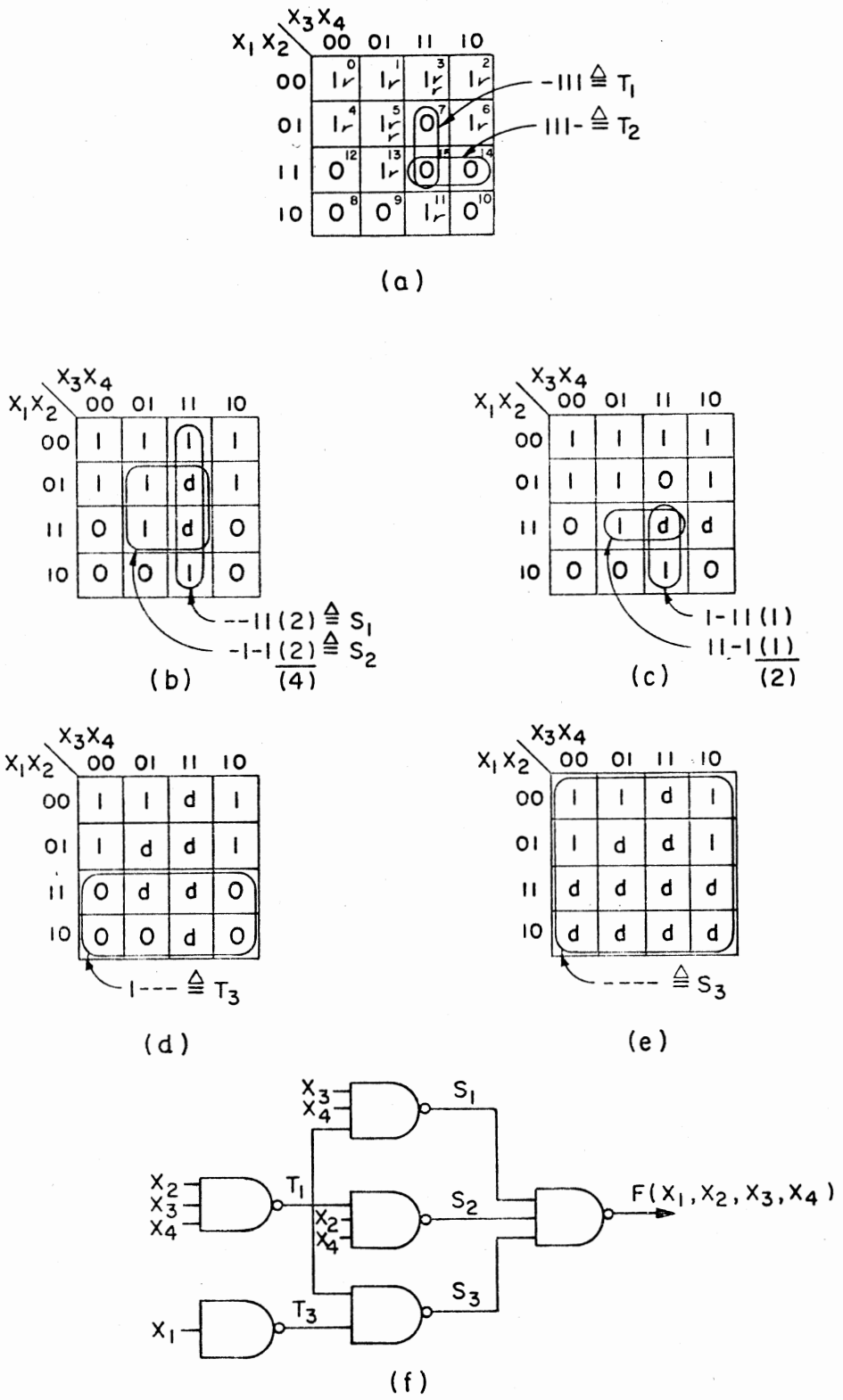


Figure 3. Design Steps and Optimal TANT Network for Example 1

the gate candidate with output S_2 covers Minterms 5 and 13. Both of these gate candidates require T_1 as an input to cover Maxterms 7 and 15. Input variables needed for these second-level and third-level gate candidates are identified in the expressions for S_1, S_2, S_3, T_1, and T_3 in Figures 3(a), 3(b), 3(d), and 3(e). No other possible TANT networks covering the desired minterms can be formed from these gate candidates. Furthermore, no gate or input reduction (Step 7) is possible. Therefore, the combinational switching circuit shown in Figure 3(f) is the optimal TANT network (Step 8).

This example was worked by Torng (15) using Eisenberg's method (14) as Example 8.5 on Pages 118 through 125. The result in Figure 3(f) is identical to Torng's switching circuit realization in Figure 8.20 on Page 125 of (15), which was obtained after applying a level-reduction technique to the primitive realization.

Example 2

As a second example, consider the switching function given by

$$F(x_1, x_2, x_3, x_4) = \Sigma m\ (0, 1, 2, 6, 7, 8, 11, 12, 14) \qquad (25)$$

which is indicated on the Karnaugh map of Figure 4(a). Step 1 directs the algorithm to Step 4, where the single prime implicate T_1 is formed. Passing through Step 5 to Step 2 yields the three prime implicants S_1, S_2, and S_3 in Figure 4(b). Since not all minterms have been covered (Step 3), return to Step 4. The four prime implicates T_2, T_3, T_4, and T_5 are formed in Figure 4(c). None of these prime implicates considered singly allows new minterm coverage in Step 5. Taken two at a time, T_3 with T_4 allows Minterm 8 to be covered and T_2 with T_3 allows Minterm 2

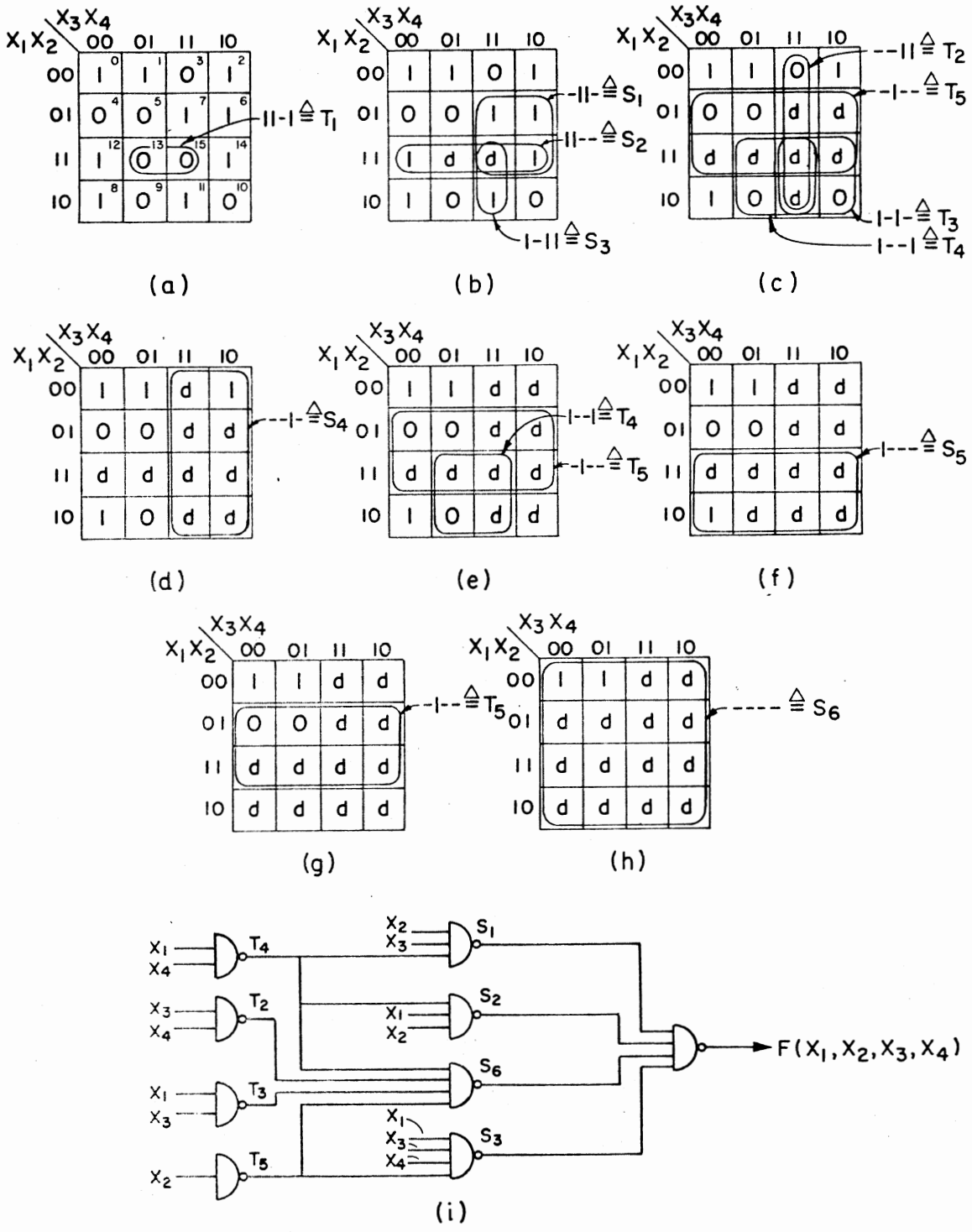


Figure 4. Design Steps and Optimal TANT Network for Example 2

to covered. Arbitrarily choosing the T_2 and T_3 combination yields S_4 in Step 2, as shown in Figure 4(d). Again, return to Step 4. T_4 and T_5 are regenerated (Figure 4(e)), but when considered singly, only T_4 allows new minterm coverage. Returning to Step 2 yields S_5 (Figure 4(f)), which covers Minterm 8. Once again, return to Step 4. T_5 is regenerated (Figure 4(g)), allowing in Step 2 the coverage of the entire map (S_6 in Figure 4(h)). Since all minterms have now been covered (Step 3), proceed to Step 6. Select S_6, which covers Minterms 0, 1, 2, and 8, as the last gate candidate formed. Moreover, select S_1 to cover Minterms 6, 7, and 14, S_2 to cover Minterms 12 and 14, and S_3 to cover Minterm 11. Observe that T_2, T_3, T_4, and T_5 must be used as inputs to S_6 to cover all maxterms. Rather than selecting T_1 to feed S_1, S_2, and/or S_3, T_4 may be used to feed S_1 and S_2, and T_5 to feed S_3, as shown in Figure 4(i). Alternately, S_2 may be used to cover only Minterm 12, since Minterm 14 is already covered by S_1. If both T_3 and T_4 are used to feed S_2, then the x_2 input to S_2 can be removed, and S_2 covers Minterms 8 and 12. Therefore, this input change yields a second optimal TANT network, in addition to the one shown in Figure 4(i).

Example 3

The purpose of this final example is to illustrate input reductions for second-level and third-level gates. Let the switching function to be realized be given by

$$F(x_1, x_2, x_3, x_4) = \Sigma m(0, 2, 3, 4, 5, 6, 7, 8, 10, 12, 15) \qquad (26)$$

Step 1 directs the algorithm to Step 2, where S_1 is formed in Figure 5(a). Passing through Step 3, Prime Implicates T_1 and T_2 are generated in

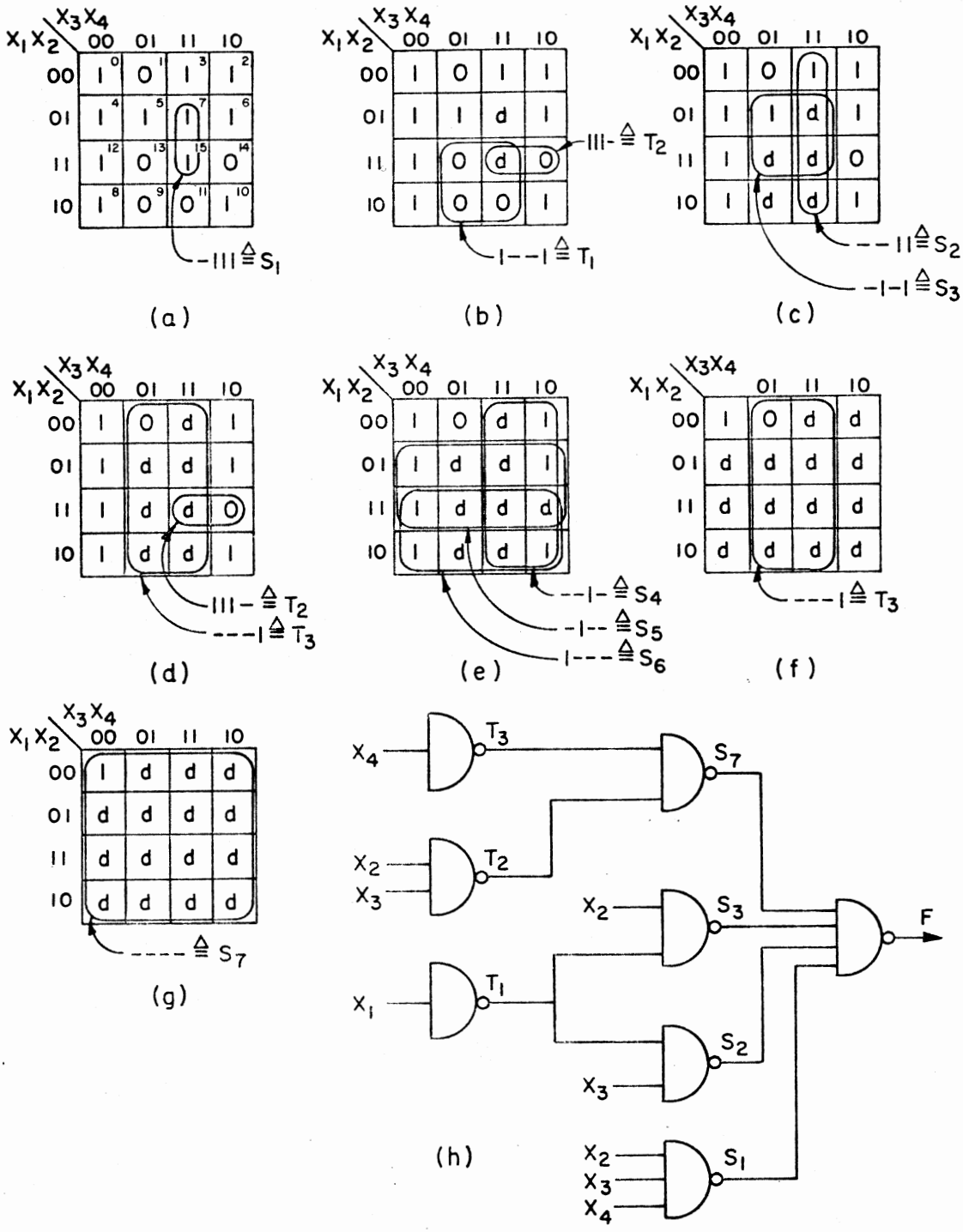


Figure 5. Design Steps and Optimal TANT Network for Example 3

Step 4 (Figure 5(b)). In Step 5, T_1 covers more new minterms, and T_2 is discarded. Returning to Step 2, S_2 and S_3 are generated (Figure 5(c)). In Step 4, T_2 and T_3 are generated, but only T_2 is retained in Step 5 (Figure 5(d)). Figure 5(e) shows the results of forming S_4, S_5, and S_6 in Step 2. Thereafter, T_3 is regenerated in Step 4, allowing S_7 to be formed in Step 2 (Figures 5(f) and 5(g)).

It is the application of Steps 6 and 7 in this example which requires special attention. S_7 is selected, as the last gate candidate generated, for a second-level gate to cover Minterms 0, 2, 4, 6, 8, 10, and 12. Both T_2 and T_3 are needed as inputs to this gate. In particular, T_2 covers Maxterm 14, and T_3 covers Maxterms 1, 9, 11, and 13. The sole first second-level gate candidate generated (S_1) covers Minterms 7 and 15, and no third-level gate outputs are required as added inputs to this gate. Only Minterms 3 and 5 remain to be covered at this point. Minterm 3 can be covered by using either S_2 with inputs x_3, x_4, and T_1 or S_4 with inputs x_3, T_1, and T_2. Minterm 5 can be covered by using either S_3 with inputs x_2, x_4, and T_1 or S_5 with inputs x_2, T_1, and T_2. Therefore, using either S_2 and S_3 or S_4 and S_5 should result in a TANT network covering all the minterms. However, if S_2 and S_3 are used, then the x_4 input to T_1 is redundant and can be removed by Theorem 9. Moreover, the x_4 input can be omitted from both the S_2 and S_3 second-level gates as shown in Figure 5(h), since Minterms 2 and 6 and Minterms 4 and 6 are included in the desired switching function (Theorem 10). Theorem 10 also allows x_1 to be removed as an input to T_2, since Minterm 6 is now covered by both S_2 and S_3. The resulting optimal TANT network is shown in Figure 5(h).

Summary

The theorems provided a firm groundwork for the steps in a network synthesis algorithm for the design of optimal three-level NAND-gate combinatorial switching networks having only uncomplemented inputs (TANT). A detailed description of the procedure has been provided and a flowchart included for easy reference. Three examples have been presented to demonstrate pertinent details of the algorithm. The network synthesis procedure utilizes cyclical gate candidate generation and exhaustive input optimization, which is very amenable to digital computers.

CHAPTER III

THE CAD SOFTWARE PACKAGE

This chapter discusses the computer implementation of the algorithm described in Chapter II. The first section gives a detailed description of the program and some major subroutines. The second section discusses evaluation of the CAD package. The third section of this chapter analyzes alternative methods for Step 5 of the algorithm, which is the choosing of third-level gate candidates. The fourth section describes the slight modification required for the NOR version, and the final section provides a summary.

Program Description

The program, like the algorithm, has two main parts: the part which finds the gate candidates and the part which finds and minimizes circuits using these candidates. Figure 2 (for the algorithm) and Figure 6 (for the program) differ primarily in the part on prime implicant generation. The difference allows more of the interdependence to be seen for Steps 1 through 5 in the program. This interdependence is dictated in part by the choice of a method for Step 5. This first part finds gate candidates by continuing to circle alternately on the 1-sets and 0-sets until all minterms are covered. The second part differs in that the algorithm does all of the work in parallel; whereas, the program enters a loop. In the loop of Part II, some circuit is formed from

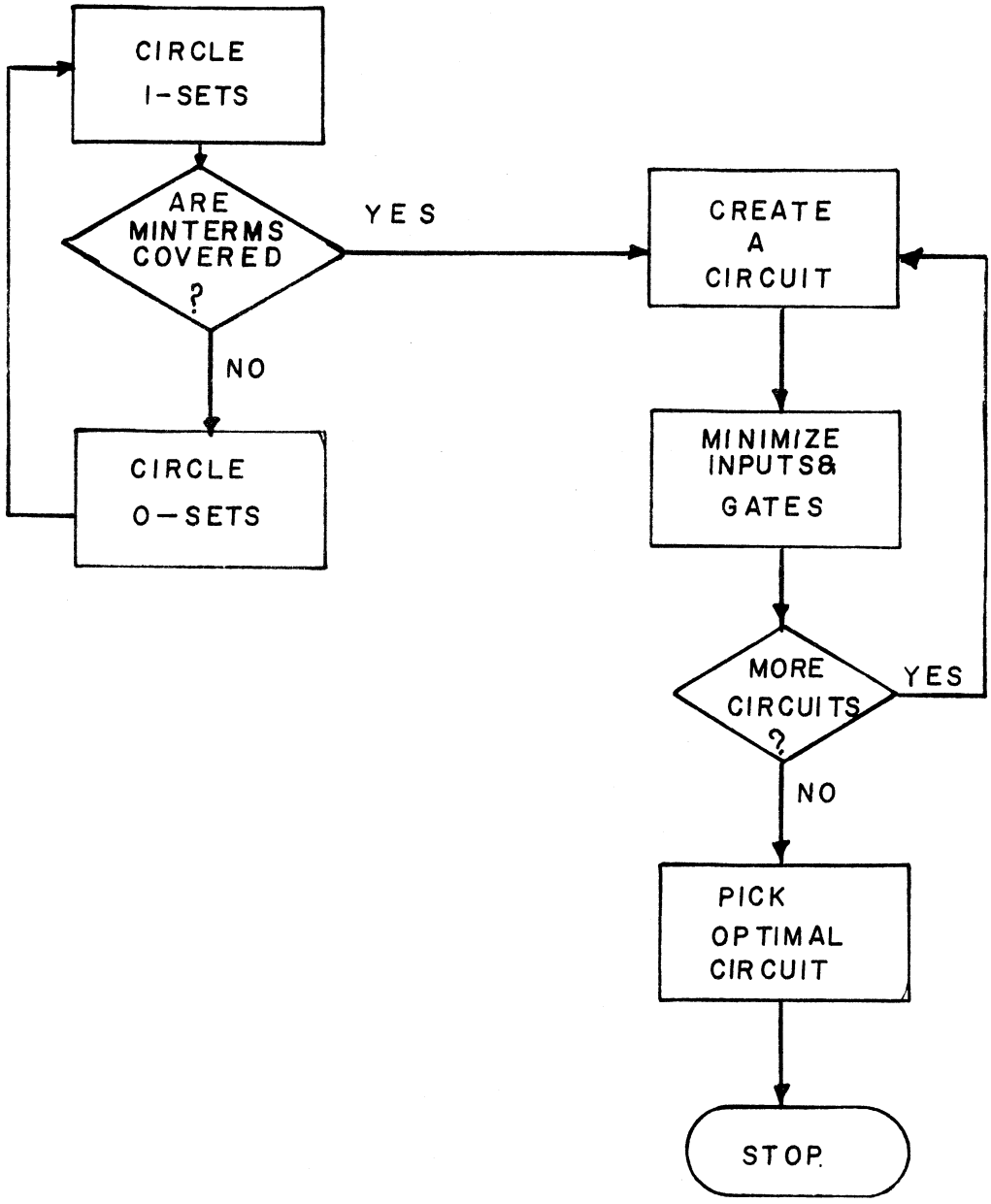


Figure 6. CAD Program Flow Chart

the candidates and then minimized, then another circuit is formed and minimized, and so forth until all possible circuits are formed and minimized. Finally, the best of these circuits is chosen as the solution.

Circling on the 1-sets (Step 2) and circling on the 0-sets (Step 4) are actually performed by the same subroutine with an input parameter indicating whether to circle on 1's or 0's. The subroutine SUBFN simply starts with the lowest minterm (maxterm), checks to determine whether it is a cell which includes the needed cube (the primary cube for TANT problems), then tests whether all members of that cell are minterms (maxterms) and, if so, that cell becomes a candidate. This subroutine is essentially finding implicants (implicates) as gate candidates. All entries within the cell are then reassigned values indicating "don't cares" for subsequent gate candidate generation. Steps 2 (circling the 1-sets), 4 (circling the 0-sets), and 5 (choosing which 0-set to discard) are all contained in a large loop, which uses minterm coverage (Step 3) to terminate the loop. Proof that these steps produce the necessary gates for an optimal network is contained in Theorems 1 through 5.

Subroutine CKFND is used to find circuits, i.e., generate different combinations of second-level gate candidates to cover the minterms. Theorem 6 states that the last candidate generated must be used. If the first second-level candidate were generated solely, it must also be used (Theorem 7). The other candidates are chosen for the circuit by a simple covering and closure table. This routine also eliminates those third-level gates solely feeding second-level gates by feeding their inputs directly into the first-level gate (Theorem 8). Subroutine CKFND corresponds to part of Steps 6 and 7.

Subroutine M3RDI eliminates from the third-level gates any inputs

that are also present in all second-level gates it feeds. This is part of Step 7 and is covered by Theorem 9. Subroutines EL2IN and EL3IN eliminate unnecessary inputs to the second- and third-level gates, respectively. This part of Step 7 uses exhaustive search. An input is removed and, if the resulting output function is unaltered, the change is made permanent; otherwise, the input is replaced. After input reduction, some gates may have identical inputs. Subroutine DUPGT finds these redundancies and eliminates them.

At the beginning of the circuit selection sequence, CKFND found a combination of second-level gates and generated a circuit. This circuit was then minimized. Now, Subroutine ALTCK modifies the feeding third-level gates in an effort to come up with an alternate circuit with the same prime implicants (second-level gate candidates), which is then minimized. ALTCK and CKFND are called until all possible circuits are found and minimized. Step 8, the choosing of the optimal circuit(s), is then executed. The Appendix contains a listing of the computer program.

CAD Evaluation

Three categories for evaluation of a CAD software package are execution time, core storage required, and user convenience. Algorithm correctness to yield an optimal network is necessary before any of these measures even has any significance. The first two categories lend themselves readily to numerical analysis, whereas the third is a rather subjective quality. This section uses tabular and graphical displays of the first two categories.

Table I shows the measurements on many problems (including several from the literature) allowing a very thorough evaluation of this

TABLE I

TIME REQUIREMENTS FOR SEVERAL TANT SOLUTIONS

Case Number	Order	Number of Minterms	Number of Don't Cares	Execution Time (sec)	Number of Gates	Number of Inputs
1	4	11	0	1.23	8	17
2	4	9	0	0.56	6	15
3	4	9	0	1.18	9	25
4	4	11	0	0.68	6	17
5	4	1	0	0.56	6	9
6	4	8	0	1.78	8	19
7	4	1	1	0.58	6	9
8	4	3	3	0.51	5	8
9	4	3	0	0.61	7	13
10	4	7	0	0.89	6	15
11	3	5	0	0.49	3	5
12	4	7	0	0.50	5	11
13	6	20	0	1.54	9	23
14	4	7	0	0.61	7	19
15	4	6	0	0.74	6	15
16	4	7	0	0.59	6	16
17	4	7	0	0.92	7	15
18	6	10	0	1.11	8	18
19	7	15	0	3.21	8	18
20	4	6	0	0.60	7	21
21	4	9	0	1.00	8	20

package. The predominance of fourth-order problems is due both to the ease of hand analysis and to their presence in most of the literature. The timing analysis is, therefore, very accurate for the fourth-order problem, but not quite as complete for the higher-order problems. The times for the fourth-order problem range (from 0.50 seconds to 1.78 seconds. The time for the third-order problem was 0.49 seconds. The average for the fourth-order problems was 0.78 seconds. These are shown in Table I. Figure 7(a) shows a graph of the timing data. This data seems to point out a steep (perhaps exponential) rise of execution time with increased order.

The storage analysis is shorter primarily because the number of runs to find out the required storage is one, that is one fourth-order problem will take as much storage as any other fourth-order problem. The graph in Figure 7(b) displays the storage requirements for various order problems. As one might expect, the storage requirement is relatively constant over the range of lower order problems because program statement storage predominates. As the order increases, the array storage begins to dominate and the storage requirement begins to soar. The information in this section is sufficient for a complete numerical evaluation of the CAD package.

Alternate Procedures for Choosing Prime Implicates as Third-Level Gate Candidates

The algorithm describes a Step 5 procedure which chooses an 0-set as a gate candidate based upon how many new minterms that particular 0-set allows new 1-sets to cover. This procedure is not intrinsic to the algorithm, and, consequently, other methods are possible. The

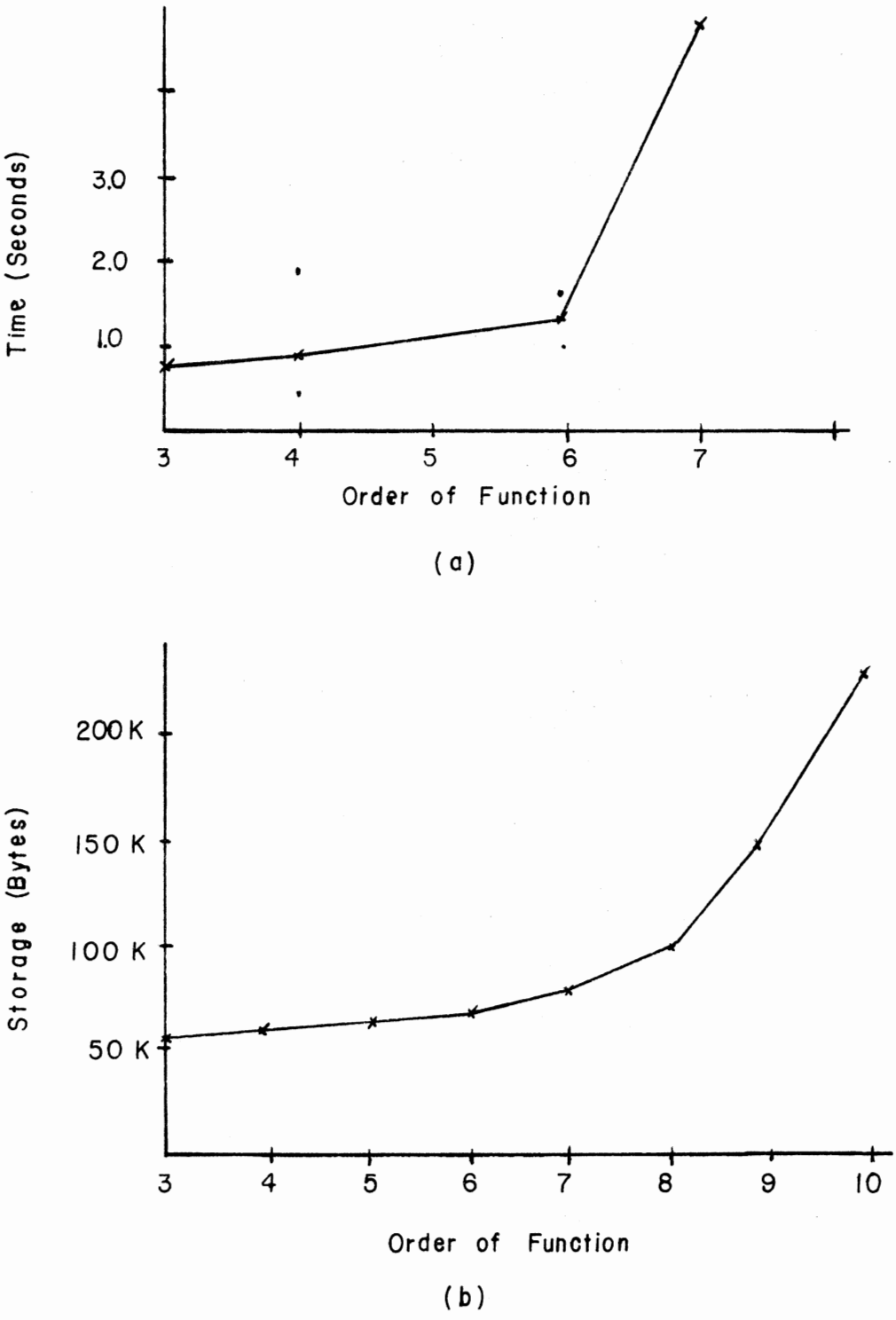


Figure 7. Timing and Storage Analysis of the CAD Program

choice could be based upon the number of 1-sets allowed to be generated, number of 0's covered, minterms in "hard" places, maxterms in "hard" places, or the first candidate which allows generation of a new 1-set.

Since the operation of circling on 0's is to allow second-level gate candidates to be generated, the method of choosing the 0-set which allows the most 1-sets has merit. The disadvantage of this method comes in the form of extra or unnecessary candidates being generated in some cases. The method of finding the most new 0's follows from the fact that 0's must be covered to allow second-level gate candidate generation. However, it is obviously not the purpose of this algorithm to cover all of the maxterms, or even as many as possible. A "hard" places criterion can be defined as those terms farthest from the primary cube. This would generate larger cells and fewer inputs to the candidates. The method chosen does not affect the optimality of the solution, as regeneration comes into play, but different methods result in faster or slower execution times. For example, the method of choosing the first 0-set allowing new minterm coverage is faster in the choosing stage but may be slower in the long run by taking more time to generate all of the second-level candidates. Saving all of the candidates and solving in parallel, thus avoiding any choices, would not only increase time requirements, because of having to solve for all possible candidates, but would increase storage space in memory.

Table II shows the execution time and storage requirements for some of the choices for Step 5. The execution times are for an ensemble of 20 problems of various orders. The storage requirement is for up to an eighth order problem. Almost immediately apparent is the very small difference in core storage requirements. This is due to the fact that

TABLE II

TIMING AND STORAGE COMPARISONS FOR METHODS
OF CHOOSING PRIME IMPLICATES

Method	Execution Time	Storage
Original Step 5	11.56	103k Bytes
Maximum 1-sets	11.60	103k Bytes
Maximum maxterms	11.85	103k Bytes
Minterms in "hard" places	10.90	104k Bytes
Maxterms in "hard" places	11.77	104k Bytes
Simplistic	8.79	100k Bytes

much of the storage requirement is due to the input and gate optimization portion of the program.

The original Step 5 chose third-level gate candidates based upon the number of new minterms allowed to be covered. The other methods are self-explanatory. All of the sophisticated methods, i.e., the original, the maximum 1-set, the maximum minterms, the maximum maxterms, minterms in hard places, and maxterms in hard places, seemed justifiable. Actually, these methods took more time and storage than the simplistic method of choosing the first 0-set or sets to allow minterm coverage to be chosen as third-level gate candidates. Regeneration of necessary gate candidates guaranteed the optimality of any of the methods. The difference of about 4K bytes between the sophisticated methods and the simplistic method was the subroutine to determine gate candidate choice. More significant, the simplistic method used only 8.79 seconds of CPU time to solve the examples, compared to 10.9 to 11.85 seconds for the sophisticated approaches.

NOR Example

This section describes how the NAND algorithm can be used to find a NOR circuit. In fact, Theorem 12 proves that any NAND circuit finding algorithm can be used to find a NOR circuit, or vice versa. The procedure is to complement the function, then complement the inputs, and solve as if solving for a NAND circuit. The resulting circuit is merely implemented using NOR gates to realize the original function. This procedure is based upon the complemented equivalency of NAND and NOR circuits as proven by Theorem 12.

Theorem 12

When every NAND gate in a circuit is replaced in that same circuit with NOR gates, the new circuit is the realization for the complement of the original switching function with complemented inputs.

Proof:

Given a switching function

$$F(x_1, x_2, x_3) = \overline{x_1 \cdot \overline{(x_2 \cdot \bar{x}_3)}} \tag{27}$$

the NAND circuit can be readily recognized. Replacing the NAND gates with NOR gates yields

$$\begin{aligned} F'(x_1, x_2, x_3) &= \overline{x_1 + \overline{(x_2 + \bar{x}_3)}} \\ &= \overline{x_1 + (\bar{x}_2 \cdot x_3)} \\ &= \bar{x}_1 \cdot \overline{(\bar{x}_2 \cdot x_3)} \\ &= \overline{\bar{x}_1 \cdot \overline{(\bar{x}_2 \cdot x_3)}} \end{aligned} \tag{28}$$

which is the complement of F with complemented inputs.

Example

Let the combinational function to be realized be given by

$$F(x_1, x_2, x_3, x_4) = \Pi M(0, 3, 4, 7, 8, 11, 12, 13, 15). \tag{29}$$

Figure 8(a) shows the complement of this function, which is just replacing the maxterms with minterms, and vice versa. Complementing the inputs essentially moves the minterms around; e.g., 0 becomes 1 and 13 becomes 2, which is illustrated in Figure 8(b). Step 1 indicates that the

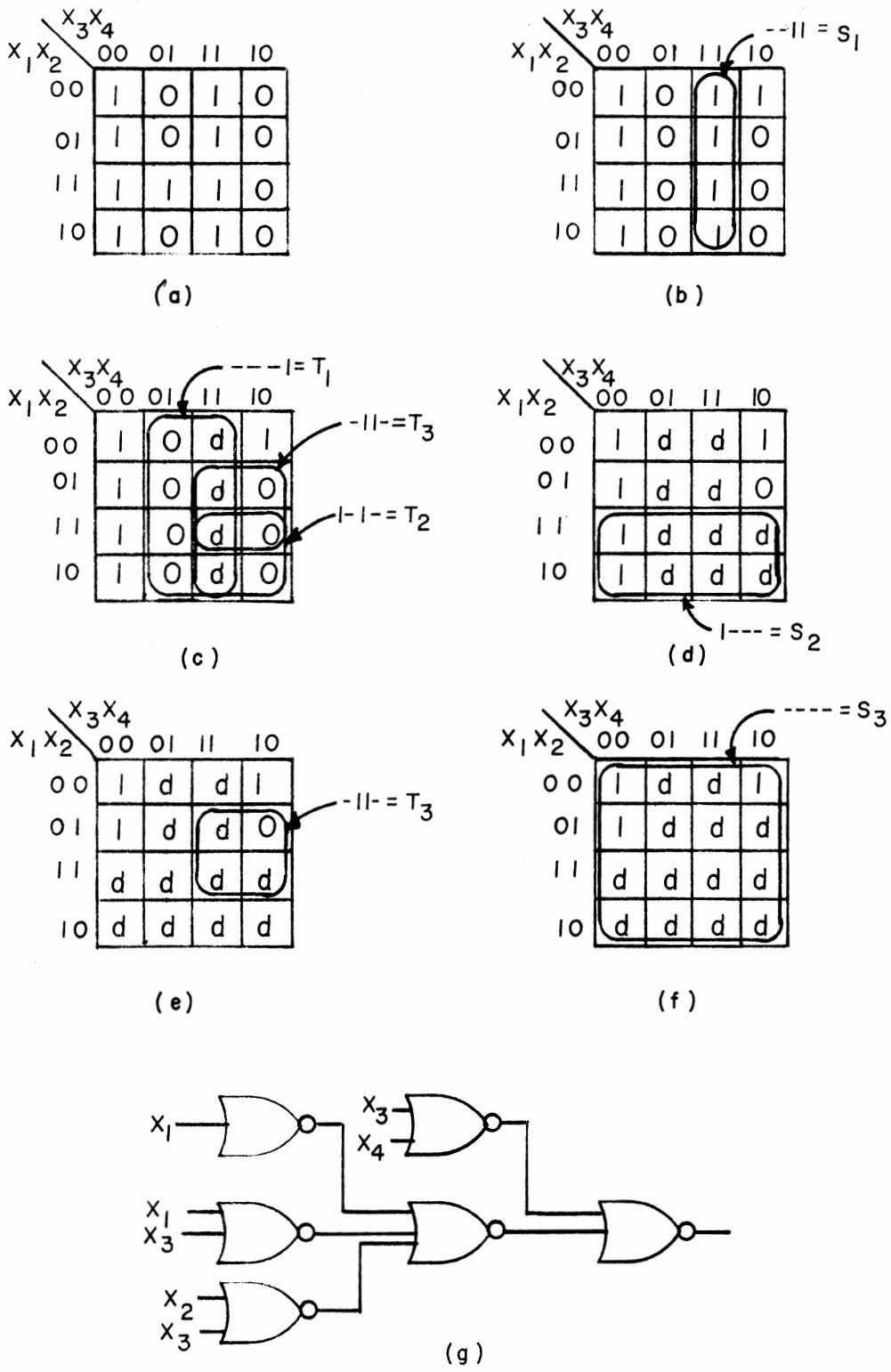


Figure 8. Design Steps and Optimal Network for NOR Example

primary cube is a 1; therefore, circling on the 1's yields S1 as shown also in Figure 8(b). Step 5 then circles on the 0's and T1 and T2 are choosen as candidates from the 0-sets generated, as shown in Figure 8(c). Returning to Step 2, S2 is generated as is shown in Figure 8(d) Figure 8(e) shows T3 which was actually regenerated by Step 5. Step 2 now generates S1 which covers the whole map and is shown in Figure 8(f). Step 3 exits the candidate-generation loop, as all of the minterms have been covered. Step 6 finds that only S1 and S3 are needed to cover all of the minterms, and that T1, T2, and T3 must feed S3. Steps 7 and 8 find no minimization or alternate circuits. The resulting NOR circuit is shown in Figure 8(g).

Summary

This chapter has described the computer implementation of the algorithm of Chapter II, and the slight differences required. A detailed description of the program and some major subroutines has been given. The CAD package has been evaluated based upon time and storage requirements. Alternative methods for choosing from among 0-sets for third-level gate candidates in Step 5 have been discussed. Finally, the adaptation to the NOR problem has been presented and proved.

CHAPTER IV

SEQUENTIAL DESIGN WITH TANT NETWORKS

This chapter describes the solution of sequential problems which use the TANT criteria for the combinational portion. The first section formulates the problem requiring sequential machines with TANT combinational logic. The second section describes the modifications required for the algorithm in Chapter II to solve the sequential problem. The third section describes the computer modifications for solving the generalized excitation table. A sequential example is given in the fourth section, and the final section provides a summary.

Problem Description

In general, a sequential problem is composed of two portions, a combinational logic circuit and some memory devices, as shown in Figure 9. The primary inputs are the n x's, and the m z's are the output. The j memory devices are fed by the excitation variables represented by E's, and feedback y's to the combinational portion. The sequential portion of the problem proceeds from the primitive flow table to the generalized excitation table. The generalized excitation table is used to start the TANT solution for the combinational portion of the problem. The sequential design portion is well documented, but the application of the TANT algorithm of Chapter II to the generalized excitation table is a new concept.

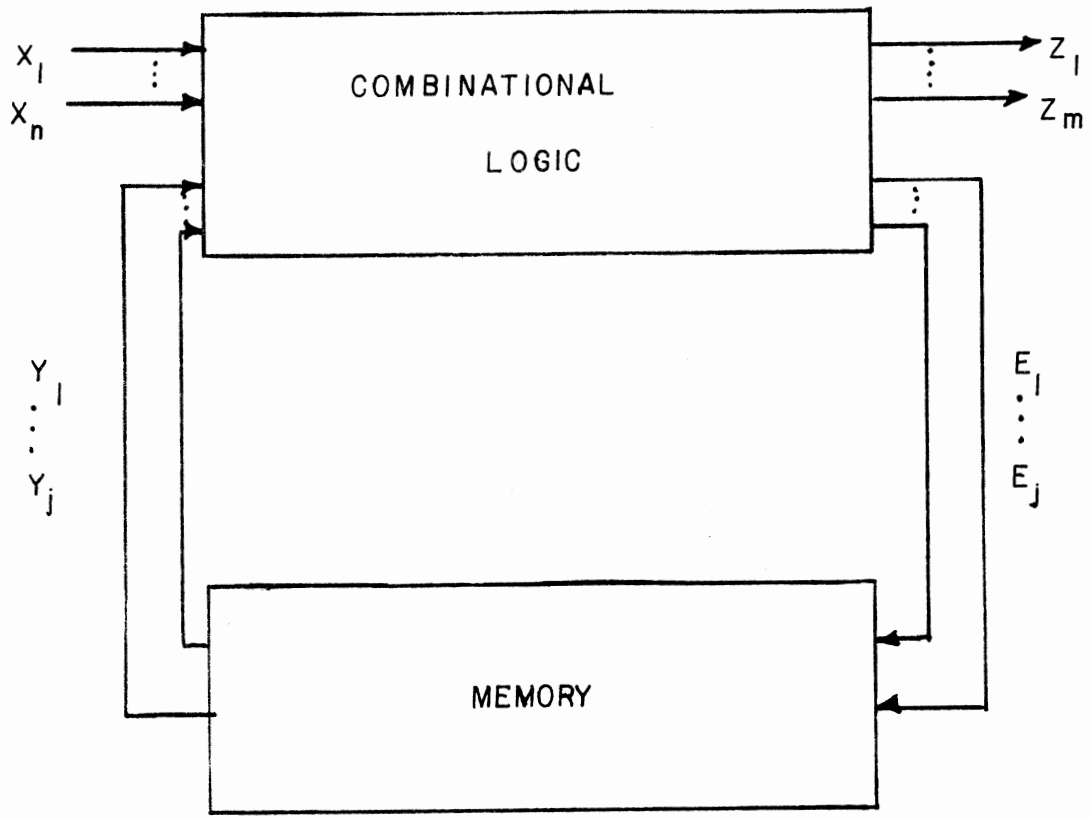


Figure 9. The General Sequential Circuit

The sequential problem begins by converting the word statement of the problem into a primitive flow table; i.e., a flow table with only one stable state per row. The next step is to use state reduction techniques to find a minimal row flow table. Next, state assignments are made and the transition table is generated. From the transition table a generalized excitation table is generated. This is a key step, in that one does not need to choose a particular type of flip-flop yet. In many design procedures the type of flip-flop is chosen at this point and excitation tables are used to produce the Boolean equations for the combinational portion of the problem. By using the generalized excitation tables, the choice of a particular flip-flop may be postponed until the optimal combinational circuit is found for each option and, thus, the overall optimal sequential machine is determined. The combinational portion of the problem proceeds as in the algorithm of Chapter II for the solving of a TANT problem.

Modifications for Sequential Applications

The basic TANT algorithm requires two modifications to be used effectively for sequential design problems. Since the state values from the memory are available in both complemented and uncomplemented form, the TANT restriction of true inputs only must be relaxed for the optimal combinational logic design problem. Secondly, the memory requires excitation values which must be supplied by the combinational portion of the circuit. These outputs, as well as the total network's output require that the TANT algorithm be capable of simultaneous optimization to provide multiple outputs. These modifications are useful as well for other applications. For example, many problems in combinational

logic require multiple outputs, and often some inputs will be available in uncomplemented form.

The presence of some uncomplemented inputs relaxes the restriction that the primary cube (111...1) be contained within every cell used as a gate input combination. The primary cube becomes a primary cell, with the requirement that every implicant or implicate used as a gate candidate must contain at least one member of the primary cell. The primary cell does contain the primary cube. The extended TANT algorithm will solve regular TANT problems, as they are a special case of the set of problems requiring a primary cell (i.e., TANT has a primary cell of dimension 1).

The multiple output case for TANT problems proceeds similar to multiple output AND/OR problems. The minterms are "tagged" to indicate for which function (outputs or excitations) the minterm is intended. As the prime implicants are generated the "tags" are carried along, and any implicant with all minterms having the same flag(s) becomes "tagged" for the same function(s) as the minterms. The final count can, thus, be minimized by using some of the same gates for two (or more) functions. The circling on the O's need not be tagged, as any function can use any third-level gate it might need. The multiple output extension is merely a bookkeeping procedure, but does result in the overall optimal result.

The Generalized Excitation Table

An excitation table shows what the next state will be given the current state and the input, a generalized excitation table (25) shows what action will be taken when given the current state and the applied inputs. Each type of flip-flop has a unique excitation table for a

given problem. However, a generalized excitation is applicable to any flip-flop. The computer program uses the generalized excitation table to generate the excitation for each of the types of flip-flops within its library. The TANT algorithm will solve the combinational circuit for each excitation table and print the optimal result. The computer program merely uses the generalized excitation to postpone the selection of a particular flip-flop.

The entries in a generalized excitation table are: a 1, which indicates that the current state is 1 and the next state is 1; a 0, which indicates the current state is 0 and the next state is 0; a Θ, which represents the transition from a current state of 1 to a next state of 0, and, finally, an I, which represents a current state of 0 and a next state of 1. Table III shows how each of these entries relates to some common flip-flop devices. Table III is essentially the library that the computer uses for converting the generalized excitation table into a list of minterms for input to the TANT algorithm.

Although there are many design steps leading to the formulation of the generalized excitation table, the table is merely the beginning point for the combinational circuit which will provide the excitation values and the outputs.

A Sequential Example

This example demonstrates the solution of a sequential problem with TANT circuitry for the combinational portion. The major emphasis will be the TANT solutions. A word statement is presented with its associated primitive flow table. The minimal row table is given without showing explicitly the row reduction. The state assignments are made and the

TABLE III

TRANSLATION OF GENERALIZED EXCITATION TABLE ENTRIES FOR SOME COMMON TYPES OF FLIP-FLOPS

		R	S	J	K	T	D
I	0 → 1	0	1	1	-	1	1
Θ	1 → 0	1	0	-	1	1	0
1	1 → 1	0	-	-	0	0	1
0	0 → 0	-	0	0	-	0	-
-	-	-	-	-	-	-	-

transition table presented, which readily yields the generalized excitation table. The solution for T flip-flops is shown in detail, and some other flip-flop type solutions are presented. Finally, an optimal sequential circuit with TANT is shown in a diagram.

The desired sequential circuit is to respond to a series of input which will produce an output to trigger the release of a combination lock. The circuit is to operate in fundamental mode, have inputs of x_1 and x_2, and produce outputs of z_1 and z_2. To open the lock an output of $z_1z_2 = 11$ is to occur after the x_1x_2 input sequence of: 01, 00, 10, 11. The circuit is initialized to $y_1y_2x_1x_2 = 0001$, and only single bit changes in the inputs will be allowed. While the correct combination is being entered the output z_1z_2 shall be 10. Upon any deviation from the correct sequence the output will become 00, which will trigger external gates to bring the system back to the 0100 initial conditions.

The primitive flow table and reduced flow table are shown in Figure 10. Using the adjacency diagram of Figure 11(a) and the state assignment map of Figure 11(b), the state assignments of: $\alpha = 00$, $\beta = 01$, $\gamma = 11$, $\delta = 10$ are found. Figure 11(x) displays the transition table, and Figure 11(d) shows the generalized excitation table.

The generalized excitation table must be coverted to an excitation table for a particular type of flip-flop to begin the modified TANT design. Figure 12(a) shows this conversion for T flip-flops. ET will represent the excitation inputs for T flip-flops. The tags are A for ET_1, B for ET_2, C for z_1, and D for z_2. The primary cell in this case is $y_1y_2x_1x_2 = (--11)$. The "don't cares" in the output subtable are due to the unstable states. Step 1 of the algorithm from Chapter II shows 1's present in the primary cell, so proceeding to Step 2 (circling the 1-sets)

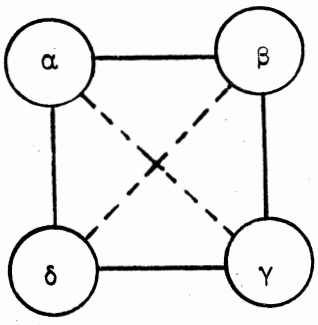
	NEXT STATE				OUTPUT Z_1Z_2			
	INPUT X_1X_2				INPUT X_1X_2			
	00	01	11	10	00	01	11	10
A	B	(A)	G	—	10	10	-0	—
B	(B)	A	-	C	10	10	—	10
C	E	—	D	(C)	-0	—	1-	10
D	-	A	(D)	F	—	1-	11	—
E	(E)	A	—	F	00	-0	—	-0
F	E	—	G	(F)	00	—	00	00
G	—	A	(G)	F	—	-0	00	00

(a)

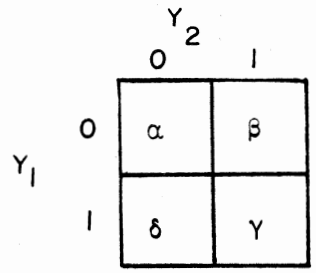
	NEXT STATE				OUTPUT Z_1Z_2			
	INPUT X_1X_2				INPUT X_1X_2			
	00	01	11	10	00	01	11	10
AB α	(α)	α	δ	β	10	10	-0	10
C β	δ	—	γ	(β)	-0	—	1-	10
D γ	—	α	(γ)	δ	—	1-	11	—
EFG δ	(δ)	α	(δ)	(δ)	00	10	00	00

(b)

Figure 10. The Primitive and Reduced Flow Tables



(a)



(b)

Y_1Y_2 \ X_1X_2	00	01	11	10
00	00	00	10	01
01	11	—	11	01
11	10	10	11	10
10	10	00	10	10

(c)

Y_1Y_2 \ X_1X_2	00	01	11	10
00	00	00	I0	0 I
01	I1	—	I1	0 1
11	1θ	1θ	11	1θ
10	10	θ0	10	10

(d)

Figure 11. Sequential Design Tables

generates $S_1 - S_4$ as shown in Figure 12(a). S_1 is tagged with D for z_2, S_2, and S_3 are tagged with a B, and S_4 is tagged with an A. The Step 3 test indicates that not all minterms are covered; therefore, proceed to Step 4. Figure 12(b) shows the cells generated by Step 4. T_1 allows the coverage of more new minterms; therefore, it is kept: Note that the T's need not be tagged as they are available to feed any second level gate. Step 3 indicates not all minterms are covered, and Step 4 regenerates T_2. Step 5 passes this sole O-set to Step 2 which generates S_9, as in Figure 12(d). All minterms are now covered, so Step 6 is now executed. Only one possible circuit is found and no minimizations are possible. The resulting circuit with 13 gates and 33 inputs and/or interconnections is shown in Figure 13.

The input and gate counts for the combinations of flip-flops is shown in Table IV for both the TANT algorithm and the double-complement algorithm. Due to the state assignment, only 2 flip-flops are required in any of the circuits. The T,D flip-flop version (i.e., using a D flip-flop for memory device 1, and a T flip-flop for memory device 2) is shown in Figure 14. The circuit using only T's has the same number of gates as the D,T version, but more inputs. The D,T version has the interesting phenomena in that a third-level gate for one function directly feeds the first-level gate of another function. The result is that a particular gate is both a second- and a third-level gate at the same time.

This example shows (a) the TANT algorithm of Chapter II, when slightly modified, yields better (fewer gates and interconnections) sequential circuits than the double-complement method, and (b) allowing multiple types of flip-flops in the same circuit can at least result in

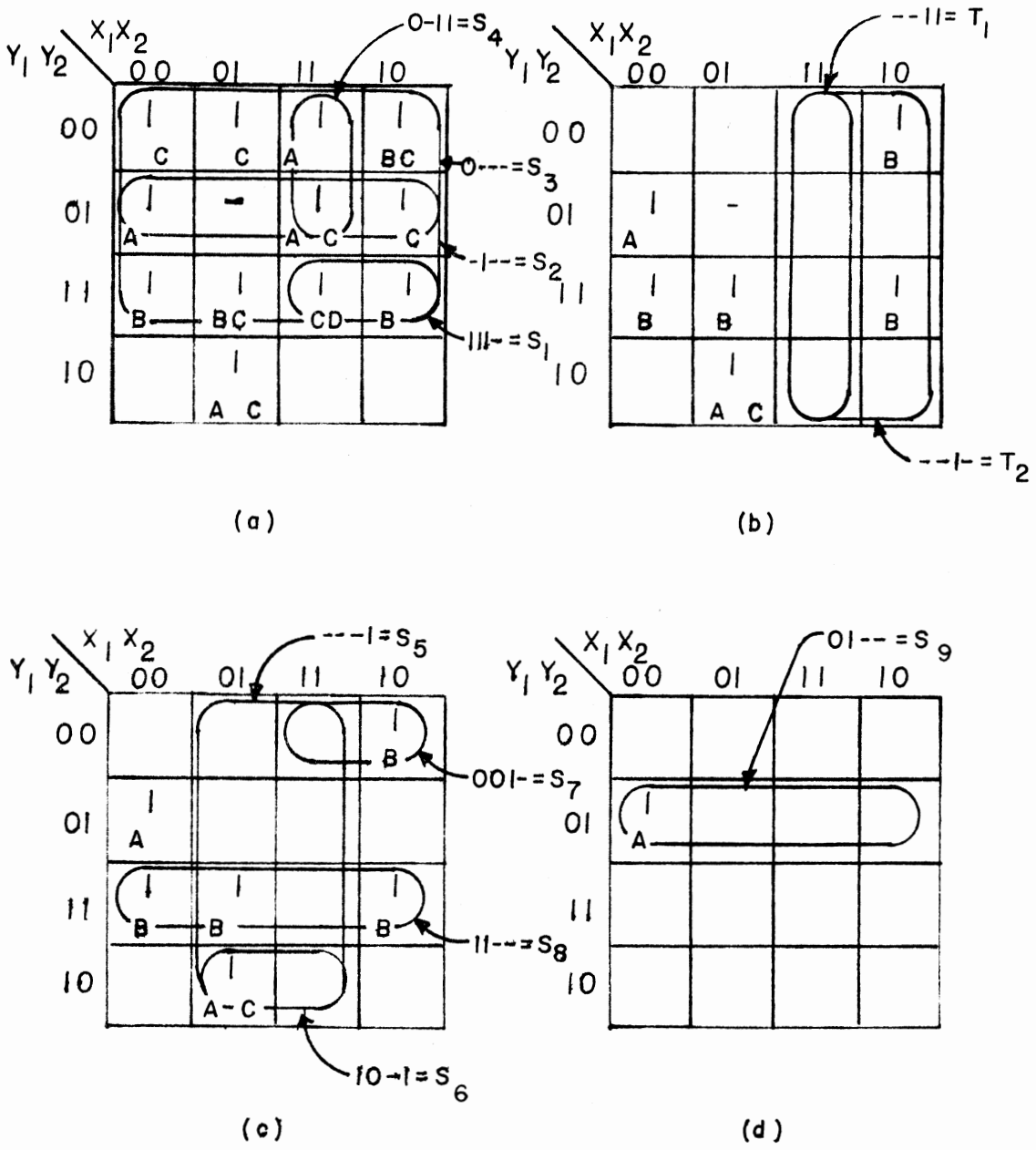


Figure 12. TANT Solution for Toggle Flip-Flops

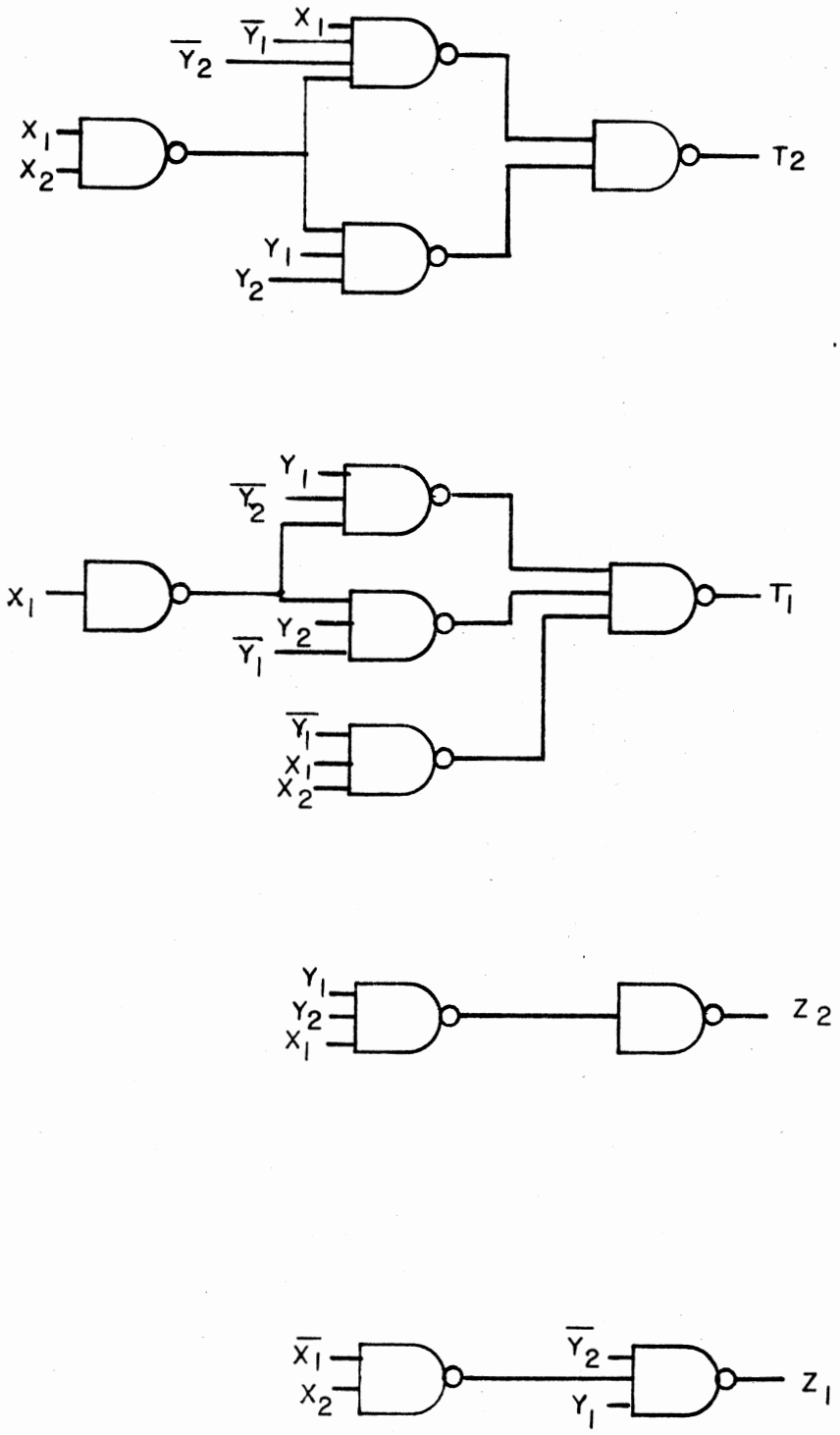


Figure 13. The Toggle-Toggle Flip-Flop Solution

TABLE IV

INPUT AND GATE COUNTS FOR SOLUTIONS TO THE SEQUENTIAL EXAMPLE

Flip-Flop Types		Using Thesis Algorithm		Using Double-Complement	
FF1	FF2	Number of Gates	Number of Inputs	Number of Gates	Number of Inputs
T	T	13	33	14	37
D	D	14	31	14	31
RS	RS	14	28	16	31
JK	JK	14	28	16	31
T	D	14	35	14	35
T	RS	14	33	15	34
T	JK	14	33	15	34
D	T	13	29	14	32
D	RS	14	29	15	30
D	JK	14	29	15	30
RS	T	13	30	15	34
RS	D	14	31	15	32
RS	JK	14	28	16	31
JK	T	13	30	15	34
JK	D	14	31	15	32
JK	RS	14	28	16	31

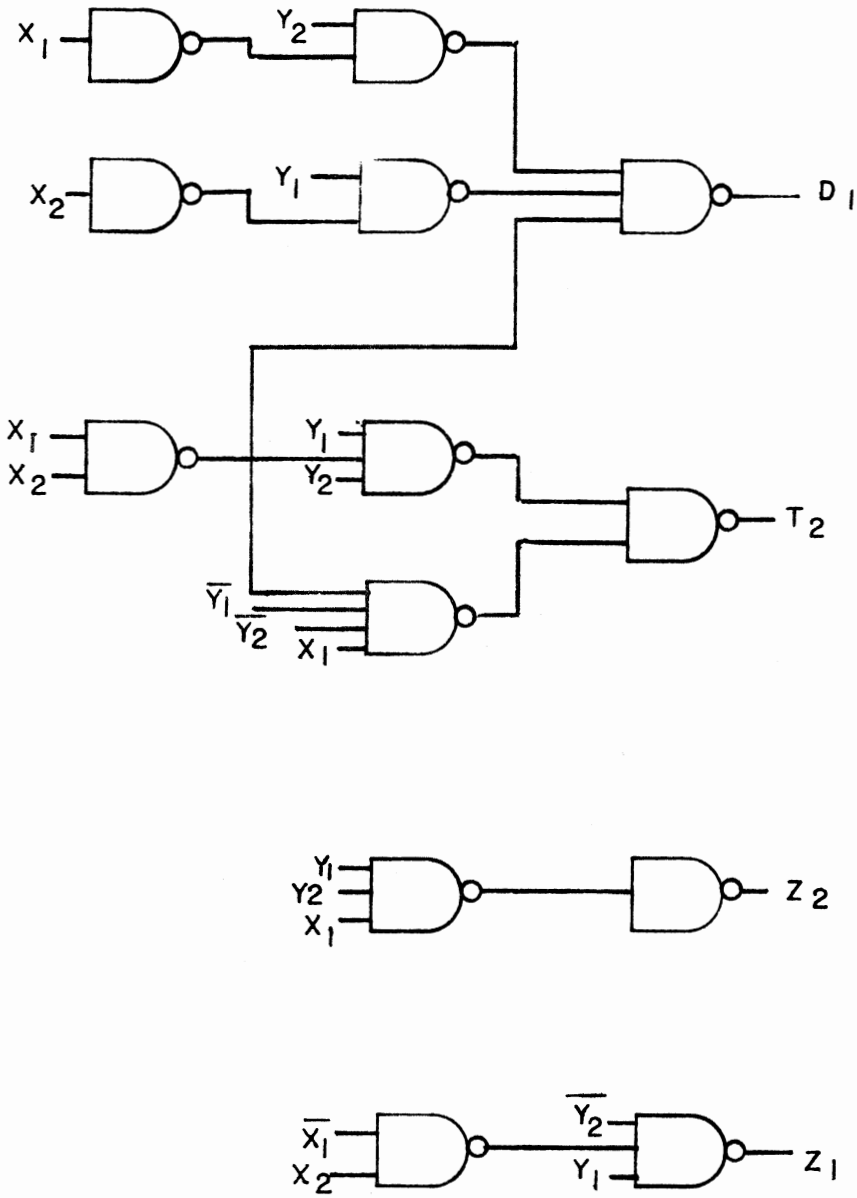


Figure 14. The Toggle-Delay Flip-Flop Solution

fewer interconnections.

Summary

This chapter described the general problem for sequential design with NAND gate combinational logic, and only true external inputs available. The modifications for the algorithm of Chapter II to handle the special requirements of primary cell and multiple output were described. The use of the generalized excitation table as the starting point for the computer program was presented. Finally, an example with detailed explanation for some cases was presented.

CHAPTER V

CONCLUSIONS AND RECOMMENDATIONS

This thesis applied a new map-factoring type of algorithm to the solution of three-level NAND-gate problems with only uncomplemented inputs allowed. A set of theorems proved the optimality of the solution found by the basic algorithm. The theorems formed a constructive proof, and the steps of the procedure for the algorithm followed the theorems closely. Expansion of the basic algorithm allowed solutions to problems with sequential applications.

Conclusions

Third-level and second-level gates correspond to prime implicates and prime implicants, respectively. Not all of the prime implicants for a given function are required for the minimum realization of that function. These points with some input reduction allow the generation of an optimal circuit realization without using an exhaustive search on all possible combinations of gates. The procedure is readily implemented on a computer. While the algorithm considered many situations in parallel, the computer, being essentially a sequential machine, required that many steps be converted to loops. The NOR problem is easily handled by the algorithm due to the complemented function with complemented inputs relationship between NAND and NOR gates. The overall procedure produces an optimal result independent of the particular method of implementation of

some of the steps, particularly the choice of prime implicates for third-level gate candidates.

The basic TANT problem is a subset of combinational problems which allow some complemented inputs. The extension of the basic algorithm to solve problems with some complemented inputs involved expanding the primary 1-cube concept to a primary r-cube. Multiple outputs are required in many combinational problems. The multiple output procedure entailed the tagging of minterms and generated prime implicants. The extensions of some uncomplemented inputs and multiple outputs allows sequential problems to be solved.

Recommendations

The creation of hazardless networks being a desirable goal, the investigation of covering minterms more than once by adding selected gates to the optimal circuit is suggested. Work on fault analysis is suggested to both determine a circuit's sensitivity to a particular fault as well as fault detection and correction. One possible approach to this problem is the generation of input test sequences, i.e., a series of inputs with known expected outputs which yields the most information with the least effort. Further investigation of fan-in and fan-out restrictions would increase the applicability of the algorithm to common design problems. A concentrated effort is needed to create a sequential CAD package to utilize fully the TANT algorithm in sequential applications.

The generality of the concept behind the algorithm could be investigated by considering more complex units than NAND gates as the basic building blocks. The basic objective is to search for an optimal circuit

without total exhaustive search upon the set of all possible circuits. This concept could be used for NAND-trees, minimizing package counts in microprocessor circuits, of a general method of attack on logic circuits. Since much of today's logic is implemented in software, the use of this algorithm to minimize programming steps in microprocessors should be investigated.

SELECTED BIBLIOGRAPHY

(1) McCluskey, E. J. "Logical Design Theory of NOR-gate Networks With No Complemented Inputs." 1963 Proc. 4th Annual Symp. on Switching Circuit Theory and Logical Design, IEEE Special Publication S-156 (1963), pp. 139-148.

(2) Nagle, H. T., Jr., Carroll, B. D., Irwin, J. D. An Introduction to Computer Logic. Englewood-Cliffs, N. J.: Prentice-Hall, 1975.

(3) Bradley, D. B. "A Survey of Boolean Function Realization Using NAND and NOR Logic." (Unpub. M. S. Thesis, Auburn University, 1970.)

(4) Gimpel, J. F. "The Minimization of TANT Networks." IEEE Trans. Electron. Comput., EC-16 (1967), pp. 18-38.

(5) Hohulin, K. R., Muroga, S. "Alternative Methods for Solving the CC-table in Gimpel's algorithm for Synthesizing Optimal Three-Level NAND Networks." (Abstract.) Computer (Dec., 1975), p. 95.

(6) Ellis, D. T. "A Synthesis of Combinational Logic With NAND or NOR Elements." IEEE Trans. Electron. Comput., EC-14 (1965), pp. 701-705.

(7) Davidson, E. S. "An Algorithm for NAND Decomposition Under Network Constraints." IEEE Trans. Comput., C-18 (1969), pp. 1098-1109.

(8) Dietmeyer, D. L., Su, Y. H. "Logic Design Automation of Fan-in Limited NAND Networks." IEEE Trans. Comput., C-18 (1969), pp. 11-22.

(9) Schneider, P. R., Dietmeyer, D. L. "An Algorithm for Synthesis of Multiple-Output Combinational Logic." IEEE Trans. Comput., C-17 (1968), pp. 117-128.

(10) Chakrabarti, K. K., Choudhury, A. K., Basu, M. S. "Complementary Function Approach to the Synthesis of Three-Level NAND Networks." IEEE Trans. Comput., C-19 (1970), pp. 509-514.

(11) Koh, K. S. "A Minimization Technique for TANT Networks." IEEE Trans. Comput., C-20 (1971), pp. 105-107.

(12) Frackowiak, J. "The Synthesis of Minimal Hazardless TANT Networks." IEEE Trans. Comput., C-21 (1972), pp. 1099-1108.

(13) Maley, G. A., Earle, J. The Logical Design of Transistor Digital Computers. Englewood-Cliffs, N. J.: Prentice-Hall, 1963.

(14) Eisenberg, H. "An Algorithm for the NAND-Gate Realization of Switching Functions." (Unpub. M. S. Thesis, Cornell University, 1969.)

(15) Torng, H. C. Switching Circuits Theory and Logic Design. Reading, Mass.: Addison-Wesley, 1972.

(16) Baugh, C. S., Chandersekaran, C. S., Swee, R. S., Muroga, S. "Optimal Networks of NOR-OR Gates for Functions of Three Variables." IEEE Trans. Comput., C-21 (1972), pp. 153-160.

(17) Breuer, M. A. "Implementation of Threshold Nets by Integer Linear Programming." IEEE Trans. Electron. Comput., EC-14 (1965), pp. 950-952.

(18) Cameron, S. H. "The Generation of Minimal Threshold Nets by an Integer Program." IEEE Trans. Electron. Comput., EC-13 (1964), pp. 299-302.

(19) Muroga, S. "Logical Design of Optimal Digital Networks by Integer Programming." Advances in Information Systems Science. Ed. J. T. Tou. New York: Plenum, 1970, pp. 283-384.

(20) Hellerman, L. "A Catalogue of Three-Variable OR-Invert and AND-Invert Logic Circuits." IEEE Trans. Electron. Comput., EC-12 (1963), pp. 198-223.

(21) Smith, R. A. "Minimal Three-Variable NOR and NAND Logic Circuits. IEEE Trans. Electron. Comput., EC-14 (1965), pp. 79-81.

(22) Ramamoorthy, C. V. "New Synthesis Techniques for Special Logical Circuits." (Internal Memo.) Needham, Massachusetts: Honeywell, Inc., 1962.

(23) Layton, J. E., Rowland, J. R. "Direct Map Factoring TANT Network Synthesis." (Report for National Science Foundation Research.) Stillwater, Oklahoma: Department of Electrical Engineering, 1975.

(24) Givone, D. D. Introduction to Switching Theory. New York, N. Y.: McGraw-Hill, 1970

APPENDIX

COMPUTER PROGRAM LISTING

```
                                                                        00000010
                                                                        00000020
                                                                        00000030
    M A I N .                                                           00000040
                                                                        00000050
VARIABLES:                                                              00000060
IN-LOGICAL UNIT NUMBER FOR THE INPUT DEVICE.                            00000070
IOUT-LOGICAL UNIT NUMBER FOR OUTPUT DEVICE.                             00000080
LP1-LEVEL PLUS 1.                                                       00000090
LM1-LEVEL MINUS 1.                                                      00000100
NCTST- ARRAY OF VALUES FOR TESTING CRITERIA                             00000101
       FOR STEP 5 CHOICES OF PRIME IMPLICATES.                          00000102
JCRIT- INDEX FOR TYPE OF STEP 5 BEING USED:                             00000103
     =1 MOST 1-SETS GENERATED.                                          00000104
     =2 MOST NEW MINTERMS COVERED.                                      00000105
     =3 MOST NEW MAXTERMS COVERED.                                      00000106
     =4 FIRST 0-SET GENERATED ALLOWING NEW MINTERM                      00000107
        COVERAGE.                                                       00000108
     =5 MAXTERMS IN HARD PLACES COVERED.                                00000109
     =6 MINTERMS IN HARD PLACES COVERED.                                00000110
NITMP-NUMBER OF NEW SUBFUNCTIONS(TEMPORARY).                            00000120
ITTMP-NEW SUBFUNCTIONS(TEMPORARY).                                      00000130
JMA,JMA2-LOOP INDEXING VARIABLES.                                       00000140
                                                                        00000150
   IPRIM-THE PRIMARY CUBE.                                              00000160
   NCTRM-NUMBER OF MINTERMS INPUT.                                      00000170
   ICRDR-ORDER OF FUNCTION TO BE MINIMIZED.                             00000180
   NDC  -NUMBER OF DONT CARES READ IN.                                  00000190
   ISET-TYPE OF SET TO BE PROCESSED(1-SET OR 0-SET).                    00000200
MCCEL-NUMBER OF MAXTERM CELLS FOUND.                                    00000210
MCELL-LIST OF MAXTERM CELLS FOUND.                                      00000220
MXTRM-LIST OF MAXTERMS.                                                 00000230
MTRMS-LIST OF MINTERMS.                                                 00000240
ICUBE-0-SET CUBES NOW BEING CONSIDERED.                                 00000250
NSTCP-NUMBER OF 0-SET CUBES NOW BEING CONSIDERED.                       00000260
NCMX-NUMBER OF MAXTERMS.                                                00000270
NTMP-PARTICULAR 0-SET CUBE NOW BEING CONSIDERED.                        00000280
IS-LIST OF SUBFUNCTIONS.                                                00000290
NCSF-NUMBER OF SUBFUNCTIONS.                                            00000300
IT-LIST OF RESIDUAL FUNCTIONS.                                          00000310
NCTF-NUMBER OF RESIDUAL FUNCTIONS.                                      00000320
IFUNC-FUNCTION VALUE AT VARIOUS LEVELS OF SOLUTION.                     00000330
LEVEL-CURRENT LEVEL OF FUNCTION.                                        00000340
ICCV-COVERING TABLE WHERE ROW IS SUBFUNCTION AND COLUMN IS MINTERM      00000350
NEPIS-NUMBER OF ESSENTIAL PI'S.                                         00000360
IEPIS-LIST OF ESSENTIAL PI'S.                                           00000370
IFST-FIRST LEVEL DIRECT INPUT CUBE MINIMUM.                             00000380
NCKT-NUMBER OF CIRCUITS.                                                00000390
NGATS-NUMBER OF GATES FOR A CIRCUIT.                                    00000400
NINPT-NUMBER OF INPUTS FOR A CIRCUIT.                                   00000410
ISWIC-SWITCH FOR DEBUG PRINT, IF ON(= TO IDBUG) THEN PRINT ALL          00000420
     DEBUG PRINTS.                                                      00000430
IPRT-LOGICAL UNIT NUMBER FOR LINE PRINTER.                              00000440
IDBUG-TEST VALUE FOR DEBUG SWITCH.                                      00000450
IDUM-LOGICAL UNIT NUMBER FOR NOWHERE.                                   00000460
                                                                        00000470
                                                                        00000480
VARIABLES INDEXING KEYS:                                                00000490
IT(JQ,1)=THE MINIMUM OF THE INPUT CELL FOR THE JQTH 0-SET.              00000500
IS(JR,1)=THE MINIMUM OF THE INPUT CELL FOR THE JRTH 1-SET.              00000510
IS(JR,2)=THE # OF IT'S BEFORE THE JRTH 1-SET.                           00000520
```

```
C IS(JR,3)=THE LEVEL OF IFUNC FOR THE FUNCTION VALUES FOR THE JRTH 100000530
C I2ND(JS)=THE JRTH 1-SET FOR THE JSTH 2ND LEVEL GATE.              00000540
C I3RD(JT)=THE JQTH 0-SET FOR THE JTTH 3RD LEVEL GATE.              00000550
C IFEED(JU,1)=# OF FEEDING GATES FOR THE I2ND(JU)TH GATE.           00000560
C IFEED(JU,JV)=THE JVTH FEEDING JQ FOR THE I2ND(JU)TH GATE.         00000570
C MASK2(JW,JTERM)=VALUE OF THE SECOND LEVEL GATE AT JTERM FOR THE   00000580
C                 I2ND(JW)TH GATE.                                  00000590
C MASK3(JX,JTERM)=VALUE OF THE THIRD LEVEL GATE AT JTERM FOR THE    00000600
C                   I3RD(JX)TH GATE.                                00000610
C ICONV(JQ)=JX           USED FOR INDEX CONVERSION.                 00000620
C                                                                   00000630
C                                                                   00000640
C   <<<INPUT CARDS>>>                                               00000650
C                                                                   00000660
C THERE IS A SET OF CARDS FOR EACH PROBLEM                          00000670
C   TO BE SOLVED.  A BLANK TRAILER CARD                             00000680
C   FOLLOWS THE LAST SET AND TERMINATES                             00000690
C   EXECUTION.                                                      00000700
C                                                                   00000710
C                                                                   00000720
C EACH SET CONTAINS THESE CARDS:                                    00000730
C      CARD 1- TITLE OR NOTE(MUST NOT BE BLANK).                    00000740
C      CARD 2-IN 4I5 FORMAT                                         00000750
C            COLS   1-5  ORDER,I.E. NUMBER OF INPUTS                00000760
C            COLS  6-10  NUMBER OF MINTERMS TO BE READ              00000770
C            COLS 11-15  NUMBER OF DON'T CARES                      00000780
C            COLS 16-20  EXTRA DEBUG OUTPUT DUMP SWITCH,            00000790
C                        IF 1 IN COL 20 DUMP IS ON,                 00000800
C                        IF 0 OR BLANK IN COL 20 ONLY SOLUTION PRINT00000810
C         COLS 21-25  NOR INDICATOR(IF BLANK SOLVE A NAND           00000820
C                        CIRCUIT, BUT IF A POSITIVE VALUE THEN      00000830
C                        SOLVE FOR A NOR CIRCUIT).                  00000840
C                                                                   00000850
C      CARD 3(AND MORE IF NECESSARY)-CONTAINS THE MINTERMS          00000860
C               IN 16I5 FORMAT,I.E. EACH MINTERM IS RIGHT           00000870
C               JUSTIFIED IN A FIELD OF 5 COLS.                     00000880
C                                                                   00000890
C                                                                   00000900
C                                                                   00000910
      DIMENSION NGATS(30),NINPT(30)                                 00000920
      DIMENSION NCTST(10)                                           00000921
      DIMENSION MCKTS(30)                                           00000930
      DIMENSION IEPIS(30)                                           00000940
      DIMENSION I3RD(30),IFEED(30,10)                               00000950
      DIMENSION MCELL(20)                                           00000960
      DIMENSION MXTRM(256)                                          00000970
      DIMENSION IFUNC(40,256)                                       00000980
      DIMENSION MTRMS(256),IDC(256),ICELL(20)                       00000990
      DIMENSION ICUBE(20)                                           00001000
      DIMENSION IS(30,3)                                            00001010
      DIMENSION IT(30,2)                                            00001020
      DIMENSION ISTMP(20),ITTMP(20)                                 00001030
      DIMENSION ICOV(20,256)                                        00001040
      DIMENSION LPTST(20)                                           00001050
      DIMENSION LPINE(20),I2ND(20),ICOVT(20,256)                    00001060
      DIMENSION TITLE(20)                                           00001070
      DIMENSION ITT(30)                                             00001080
      DIMENSION MASK3(32,256),MASK2(32,256)                         00001090
      COMMON IN,IOUT,IPRIM,ISET,IFUNC,LEVEL,                        00001100
     *ICONV(30),MASK3,MASK2                                         00001110
      DATA BLANK/'    '/                                            00001120
      DATA IBLNK/' '/                                               00001130
      DATA IPLUS/'+'/                                               00001140
      DATA MINUS/'-'/                                               00001150
                                                                    00001160
      IN=5                                                          00001170
```

```
                                                                      00001180
                      INITIALIZE IDBUG.  IF IDBUG IS EQUAL            00001190
                       TO ISWTC(ON THE INPUT CARDS) THEN              00001200
                      THE SWITCH IS TURNED ON AND ALL                 00001210
                      DEBUG OUTPUT WILL BE PRINTED.                   00001220
                                                                      00001230
      IDBUG=1                                                         00001240
                                                                      00001250
      IPRT12=12                                                       00001260
      IDUM=13                                                         00001270
      IPRT=6                                                          00001280
      ICUT=6                                                          00001290
      INRUN=4                                                         00001291
                                                                      00001292
                              INITIALIZE AND PRINT RUN COUNT.         00001293
                                                                      00001294
      READ(INRUN,20)JRUN,JCRIT,IDBUG,NCASES                           00001295
   20 FORMAT(I10,I1,2I2)                                              00001296
      JRUN=JRUN+1                                                     00001297
      REWIND INRUN                                                    00001298
      WRITE(INRUN,20)JRUN,JCRIT,IDBUG,NCASES                          00001299
      WRITE(IOUT,30)JRUN,JCRIT,IDBUG,NCASES                           00001300
   30 FORMAT(' JRUN=',I10,10X,'JCRIT=',I2,10X,'IDBUG=',I3,10X,        00001301
     * 'NCASES=',I3)                                                 00001302
                                                                      00001303
                              INITIALIZE CASE COUNT.                  00001310
                                                                      00001320
      JCASES=0                                                        00001330
                                                                      00001350
                                                                      00001360
              I N P U T                                               00001370
                                                                      00001380
                              FIRST CARD HAS TITLE.                   00001390
                                                                      00001400
   90 ICUT=IPRT                                                       00001410
      READ(IN,91)TITLE                                                00001420
   91 FORMAT(20A4)                                                    00001430
                                                                      00001440
                              TEST FOR BLANK CARD(IF BLANK ALL DONE).  00001450
                                (FIRST 72 COLS BLANK)                 00001451
                                                                      00001452
                                                                      00001460
      DO 92 JMA=1,18                                                  00001470
      IF(TITLE(JMA)-BLANK)93,92,93                                    00001480
   92 CONTINUE                                                        00001490
                                                                      00001500
      GO TO 9000                                                      00001510
   93 CONTINUE                                                        00001520
                                                                      00001530
                             STOP IF NCASES HAVE BEEN PROCESSED.      00001540
                                                                      00001550
      IF(JCASES-NCASES)94,8990,8990                                   00001560
   94 CONTINUE                                                        00001570
      JCASES=JCASES+1                                                 00001580
                                                                      00001590
                              PRINT TITLE.                            00001600
                                                                      00001610
                                                                      00001620
      WRITE(IOUT,96)TITLE                                             00001630
   96 FORMAT('1',20A4)                                                00001640
                                                                      00001650
                              SECOND CARD HAS FUNCTION PARAMETERS     00001660
                              ON I5 FORMAT.                           00001670
                                                                      00001680
      READ(IN,100)IORDR,NOTRM,NODC,ISWTC,NOR                          00001690
  100 FORMAT(16I5)                                                    00001700
```

```
      NCKT=0                                                            00001710
      LEVEL=1                                                           00001720
      NCTF=0                                                            00001730
      MCOV=0                                                            00001740
      NISF=0                                                            00001750
      NCSF=0                                                            00001760
      JSTOP=2**IORDR                                                    00001770
      IPRIM=JSTOP-1                                                     00001780
      JP1=JSTOP+1                                                       00001790
      DO 107 JMA=1,JP1                                                  00001800
      IFUNC(1,JMA)=0                                                    00001810
      DO 107 JMA2=1,10                                                  00001820
      ICOV(JMA2,JMA)=IBLNK                                              00001830
  107 CONTINUE                                                          00001840
                                                                        00001850
      WRITE(IOUT,110)IORDR,NOTRM,NODC,ISWTC,IDBUG                       00001860
  110 FORMAT('0 FUNCTION OF ORDER:',I5/' WITH',I5,' MINTERMS AND ',I5,' 00001870
     *DON''T CARES'/' ISWTC=',I5,'  IDBUG=',I5)                         00001880
                                                                        00001890
                              READ MINTERMS AND SET FUNCTION VALUE.     00001900
                                                                        00001910
      READ(IN,100)(MTRMS(JMA),JMA=1,NOTRM)                             00001920
      WRITE(IOUT,111)(MTRMS(JMA),JMA=1,NOTRM)                           00001930
  111 FORMAT('0 MINTERMS ARE:',50(/16I5))                               00001940
                                                                        00001950
      DO 120 JMA=1,NOTRM                                                00001960
      JTMP=MTRMS(JMA)+1                                                 00001970
      IF(JTMP-JSTOP)115,115,113                                         00001980
  113 WRITE(IOUT,114)MTRMS(JMA),IORDR                                   00001990
  114 FORMAT('0*****GOOF***** YOU HAVE ENTERED',I10,                    00002000
     # '  WHICH IS TOO BIG AS A MINTERM IN A FUNCTION OF ORDER',        00002010
     # I2,'   *****')                                                  00002020
  115 CONTINUE                                                          00002030
      IFUNC(1,JTMP)=1                                                   00002040
  120 CONTINUE                                                          00002050
                                                                        00002060
                                                                        00002070
                              READ DON'T CARES AND FILL FUNCTION VALUES 00002080
                                                                        00002090
      IF(NODC)141,141,130                                               00002100
  130 READ(IN,100)(IDC(JMA),JMA=1,NODC)                                 00002110
      WRITE(IOUT,131)(IDC(JMA),JMA=1,NODC)                              00002120
  131 FORMAT('0 DONT CARES ARE:',50(/16I5))                             00002130
      DO 140 JMA=1,NODC                                                 00002140
      JTMP=IDC(JMA)+1                                                   00002150
      IFUNC(1,JTMP)=2                                                   00002160
  140 CONTINUE                                                          00002170
  141 CONTINUE                                                          00002171
      IF(NOR)150,150,142                                                00002172
  142 CONTINUE                                                          00002173
                                                                        00002180
                                  COMPLEMENT INPUTS USING IFUNC(2,J) AS 00002190
                                   A SCRATCH PAD.                       00002200
                                                                        00002210
      DO 143 JMA=1,JSTOP                                                00002220
      JTMP=JSTOP-JMA+1                                                  00002230
      IFUNC(2,JTMP)=IFUNC(1,JMA)                                        00002240
  143 CONTINUE                                                          00002250
                                                                        00002260
                                  COMPLEMENT FUNCTION AND FIND NEW      00002270
                                  WORKING MINTERMS. RESET # OF WORKING  00002280
                                  MINTERMS.                             00002290
                                                                        00002300
      NCTRM=0                                                           00002310
      DO 146 JMA=1,JSTOP                                                00002320
      JTMP=IFUNC(2,JMA)                                                 00002330
```

```
      IF(JTMP-1)144,146,147                                          00002340
  144 NOTRM=NOTRM+1                                                  00002350
      MTRMS(NOTRM)=JMA                                               00002360
  146 IFUNC(2,JMA)=1-JTMP                                            00002370
  147 CONTINUE                                                       00002380
                                                                     00002390
  150 CONTINUE                                                       00002400
                                                                     00002410
                              TEST FOR DEBUG AND SET SWITCH.         00002420
                                                                     00002430
      IF(ISWTC-IDBUG)152,153,152                                     00002440
  152 ISWTC=IDUM                                                     00002450
      GO TO 154                                                      00002460
  153 ISWTC=IPRT                                                     00002470
  154 IOUT=ISWTC                                                     00002480
                                                                     00002490
                                                                     00002500
                              SET MAXTERMS.                          00002510
                                                                     00002520
      JMAII=1                                                        00002530
      NCMX=JSTOP-NOOC-NOTRM                                          00002540
      JTMP=NCMX                                                      00002550
      DO 160 JMA=1,JSTOP                                             00002560
      IF(JTMP)161,161,155                                            00002570
  155 IF(IFUNC(1,JMA))160,156,160                                    00002580
  156 MXTRM(JMAII)=JMA-1                                             00002590
      JMAII=JMAII+1                                                  00002600
      JTMP=JTMP-1                                                    00002610
  160 CONTINUE                                                       00002620
  161 CONTINUE                                                       00002630
                                                                     00002640
                                                                     00002650
                              TEST FOR 1 OR 0 SET.                   00002660
                                                                     00002670
      IF(IFUNC(1,JSTOP))170,180,170                                  00002680
                                                                     00002690
                                                                     00002700
            F I N D  1  S E T .                                      00002710
                                                                     00002720
  170 ISET=1                                                         00002730
      CALL SUBFN(NOTRM,MTRMS,NOCEL,ICELL,IS,NOSF)                    00002740
                                                                     00002750
      IF(NOCEL)300,300,171                                           00002760
                                                                     00002770
                              NOTE: AT THIS POINT NOCEL SHOULD NEVER BE 00002780
                                                                     00002790
  171 CONTINUE                                                       00002800
      IF(NISF)172,172,173                                            00002810
                                                                     00002820
               NISF-NUMBER OF INITIAL SUBFUNCTIONS.                  00002830
                                                                     00002840
  172 CONTINUE                                                       00002850
      NISF=NOCEL                                                     00002860
  173 CONTINUE                                                       00002870
      CALL CLPRT(IORDR,NOCEL,ICELL)                                  00002880
*                                                                    00002890
*                             DEBUG STATEMENT NUMBER PRINT.          00002900
*                                                                    00002910
      ISTMT=173                                                      00002920
      WRITE(IOUT,700)ISTMT                                           00002930
*                                                                    00002940
*                                                                    00002950
                                                                     00002960
                              INITIALIZE NEXT LEVEL OF IFUNC.        00002970
                                                                     00002980
      LP1=LEVEL+1                                                    00002990
```

```
      DO 174 JMA=1,JSTOP                                              00003000
      IFUNC(LP1,JMA)=IFUNC(LEVEL,JMA)                                 00003010
  174 CONTINUE                                                        00003020
                                                                      00003030
      LEVEL=LP1                                                       00003040
                                                                      00003050
                              REPLACE 1'S WITH R'S.                   00003060
                                                                      00003070
      DO 178 JMA=1,NOCEL                                              00003080
        NOSF=NOSF+1                                                   00003090
        LOW=ICELL(JMA)                                                00003100
        IS(NOSF,1)=LOW                                                00003110
        IS(NOSF,2)=NOTF                                               00003120
         IS(NOSF,3)=LEVEL                                             00003130
                                                                      00003140
        DO 178 JMA2=1,NOTRM                                           00003150
          IF(MTRMS(JMA2)-LOW)178,175,175                              00003160
  175     JTMP=MTRMS(JMA2)+1                                          00003170
          IF(IFUNC(LEVEL,JTMP))178,176,176                            00003180
  176     ITEST=IAND(LOW,MTRMS(JMA2))                                 00003190
          IF(ITEST-LOW)178,177,178                                    00003200
  177      IFUNC(LEVEL,JTMP)=3                                        00003210
          ICOV(NOSF,JTMP)=IPLUS                                       00003220
  178 CONTINUE                                                        00003230
      WRITE(IOUT,7777)(IFUNC(LEVEL,JKK),JKK=1,JSTOP),LEVEL            00003240
*                                                                     00003250
*                             DEBUG STATEMENT NUMBER PRINT.           00003260
*                                                                     00003270
      ISTMT=178                                                       00003280
      WRITE(IOUT,700)ISTMT                                            00003290
*                                                                     00003300
                                                                      00003310
                                                                      00003320
                                                                      00003330
                              CHECK MINTERM COVERAGE.                 00003340
                                                                      00003350
      DO 179 JMA=1,NOTRM                                              00003360
      JMA2=MTRMS(JMA)+1                                               00003370
      IF(IFUNC(LEVEL,JMA2)-1)179,180,179                              00003380
  179 CONTINUE                                                        00003390
                                                                      00003400
      GO TO 500                                                       00003410
                                                                      00003420
                                                                      00003430
           F I N D   O   S E T .                                      00003440
                                                                      00003450
  180 ISET=0                                                          00003460
      CALL SUBFN(NCMX,MXTRM,MOCEL,MCELL,IT,NOTF)                      00003470
      CALL CLPRT(IORDR,MOCEL,MCELL)                                   00003480
*                                                                     00003490
*                             DEBUG STATEMENT NUMBER PRINT.           00003500
*                                                                     00003510
      ISTMT=180                                                       00003520
      WRITE(IOUT,700)ISTMT                                            00003530
*                                                                     00003540
                                                                      00003550
                              INITIALIZE NEXT LEVEL OF IFUNC.         00003560
      LM1=LEVEL                                                       00003570
      LEVEL=LEVEL+1                                                   00003580
                                                                      00003590
      DO 181 JMA=1,JSTOP                                              00003600
      IFUNC(LEVEL,JMA)=IFUNC(LM1,JMA)                                 00003610
  181 CONTINUE                                                        00003620
      NCTST(1)=0                                                      00003630
      NCTST(2)=0                                                      00003631
      NCTST(3)=0                                                      00003632
```

```
      NCTST(4)=0                                                        00003633
      NCTST(5)=0                                                        00003634
      NCTST(6)=0                                                        00003635
      NTMP=0                                                            00003640
      ICUBE(1)=1                                                        00003650
      NSTOP=1                                                           00003660
  182 NTMP=1                                                            00003670
      NEWMAX=0                                                          00003671
      MAXDST=0                                                          00003672
  183 LOW=MCELL(ICUBE(NTMP))                                            00003680
                                                                        00003681
                              SUM UP DISTANCES FOR MAXTERMS AS          00003682
                              HARD PLACES CRITERIA.                     00003683
                                                                        00003684
      MAXDST=MAXDST+IDIST(LOW,IORDR)                                    00003685
                                                                        00003686
                                                                        00003690
                               FILL IN D*S(HERE USE A 2).               00003700
                                                                        00003710
      JSTRT=LOW+1                                                       00003720
                                                                        00003730
      DO 190 JMA=JSTRT,JSTOP                                            00003740
        IF(IFUNC(LEVEL,JMA)-2)186,190,186                               00003750
  186   JTMP=JMA-1                                                      00003760
        ITEST=IAND(LOW,JTMP)                                            00003770
        IF(ITEST-LOW)190,187,190                                        00003780
  187   IFUNC(LEVEL,JMA)=2                                              00003790
                                                                        00003791
                              INCREMENT NEW MAXTERM COVERED COUNT.      00003792
                                                                        00003793
      NEWMAX=NEWMAX+1                                                   00003794
                                                                        00003795
  190 CONTINUE                                                          00003800
                                                                        00003810
      IF(NTMP-NSTOP)191,193,193                                         00003820
  191 NTMP=NTMP+1                                                       00003830
      GO TO 183                                                         00003840
  193 CONTINUE                                                          00003850
      WRITE(IOUT,7777)(IFUNC(LEVEL,JKK),JKK=1,JSTOP)                    00003860
 7777 FORMAT(1H ,40I3)                                                  00003870
*                                                                       00003880
*                              DEBUG STATEMENT NUMBER PRINT.            00003890
*                                                                       00003900
      ISTMT=193                                                         00003910
      WRITE(IOUT,700)ISTMT                                              00003920
*                                                                       00003930
                                                                        00003940
      ISET=1                                                            00003950
      CALL SUBFN(NOTRM,MTRMS,NOCEL,ICELL,IS,NOSF)                       00003960
                                                                        00003970
                                                                        00003990
      IF(NOCEL)195,195,194                                              00004000
  194 CONTINUE                                                          00004010
                                                                        00004011
                             COUNT NEW MINTERMS AND DISTANCES           00004012
                              TO MINTERMS IN HARD PLACES.               00004013
                                                                        00004014
      NEWMIN=0                                                          00004015
      MINDST=0                                                          00004016
      DO 192 JMA=1,NOCEL                                                00004017
        LOW=ICELL(JMA)                                                  00004018
        MINDST=MINDST+IDIST(LOW,IORDR)                                  00004019
        DO 192 JMA2=1,NOTRM                                             00004020
          IF(MTRMS(JMA).LT.LOW) GO TO 192                               00004021
          JTMP=MTRMS(JMA)+1                                             00004022
          IF(IFUNC(LEVEL,JTMP).NE.1) GO TO 192                          00004023
```

```
      ITEST=IAND(LOW,MTRMS(JMA2))                                       00004024
      IF(ITEST.NE.LOW) GO TO 192                                        00004025
      IFUNC(LEVEL,JTMP)=3                                               00004026
      NEWMIN=NEWMIN+1                                                   00004027
192 CONTINUE                                                            00004028
                                                                        00004029
                          CALL STEP5 FOR TESTING FOR WHICH              00004030
                           PRIME IMPLICATE TO KEEP AS A                 00004031
                          THIRD LEVEL GATE CANDIDATE                    00004032
                                                                        00004033
  CALL STEP5(JCRIT,NCTST,NOCEL,NITMP,NSTOP,ITTMP,MCELL,                 00004034
 * ICUBE,ISTMP,ICELL,NEWMAX,NEWMIN,MAXDST,MINDST)                       00004035
                                                                        00004041
195 CONTINUE                                                            00004130
                                                                        00004140
                          RESET IFUNC.                                  00004150
                                                                        00004160
  DO 196 JMA2=1,JSTOP                                                   00004170
  IFUNC(LEVEL,JMA2)=IFUNC(LM1,JMA2)                                     00004180
196 CONTINUE                                                            00004190
                                                                        00004200
                         IF THE CRITERIA IS 4, USE THESE PRIME          00004210
                          IMPLICATES AS GATE CANDIDATES.                00004220
                                                                        00004230
  IF(JCRIT.EQ.4) GO TO 201                                              00004240
                                                                        00004241
                          TEST FOR END OF THIS SERIES OF 0 SETS.        00004242
                                                                        00004243
  JMA=NSTOP                                                             00004244
  NTEST=MOCEL                                                           00004245
197 IF(ICUBE(JMA)-NTEST)198,199,199                                     00004250
198 ICUBE(JMA)=ICUBE(JMA)+1                                             00004260
  GO TO 182                                                             00004270
199 IF(JMA-1)201,201,200                                                00004280
200 JMA=JMA-1                                                           00004290
  NTEST=NTEST-1                                                         00004300
  GO TO 197                                                             00004310
                                                                        00004320
                          THAT NUMBER OF 0-SETS HAS BEEN COMPLETELY S00004330
                                                                        00004340
201 CONTINUE                                                            00004350
  IF(NCTST(1))210,210,230                                               00004360
                                                                        00004370
                          THESE 0-SETS DID NOT INCREASE 1-SET COVERAG00004380
                                                                        00004390
210 IF(NSTOP-MOCEL)220,9010,9010                                        00004400
220 NSTOP=NSTOP+1                                                       00004410
                                                                        00004420
                          REINITIALIZE ICUBE.                           00004430
                                                                        00004440
  DO 225 JMA=1,NSTOP                                                    00004450
  ICUBE(JMA)=JMA                                                        00004460
225 CONTINUE                                                            00004470
                                                                        00004480
  GO TO 182                                                             00004490
230 CONTINUE                                                            00004500
  WRITE(IOUT,235)                                                       00004510
235 FORMAT('0 ENVELOPE PLEASE, THE CHOSEN 0-SET CELL IS:')              00004520
  MCCEL=NITMP                                                           00004530
                                                                        00004540
                          MORE MINTERMS WERE COVERED,MAKE FINDINGS FI00004550
                                                                        00004560
  DO 252 JMA=1,NITMP                                                    00004570
  NOTF=NOTF+1                                                           00004580
  LOW=ITTMP(JMA)                                                        00004590
  MCELL(JMA)=LOW                                                        00004600
```

```
      IT(NOTF,1)=LOW                                                    00004610
      JSTRT=LOW+1                                                       00004620
                                                                        00004630
                              FILL IN WITH D'S(HERE WITH -NOTF AS D'S.  00004640
                                                                        00004650
      DO 250 JMA2=JSTRT,JSTOP                                           00004660
  246       JTMP=JMA2-1                                                 00004670
            ITEST=IAND(LOW,JTMP)                                        00004680
            IF(ITEST-LOW)250,248,250                                    00004690
  248        IFUNC(LEVEL,JMA2)=-NOTF                                    00004700
                                                                        00004710
  250 CONTINUE                                                          00004720
                                                                        00004730
  252 CONTINUE                                                          00004740
                                                                        00004750
      ISET=0                                                            00004760
      CALL CLPRT(IORDR,NOCEL,MCELL)                                     00004770
      WRITE(IOUT,7777)(IFUNC(LEVEL,JKK),JKK=1,JSTOP),LEVEL              00004780
*                                                                       00004790
*                             DEBUG STATEMENT NUMBER PRINT.             00004800
*                                                                       00004810
      ISTMT=252                                                         00004820
      WRITE(IOUT,700)ISTMT                                              00004830
*                                                                       00004840
      ISET=1                                                            00004850
                                                                        00004860
                                                                        00004870
                              GO TO 1-SET HANDLING OF R'S.              00004880
                                                                        00004890
      NCCEL=NCTST(1)                                                    00004900
      NCTEMP=NCTST(1)                                                   00004901
      DO 280 JMA=1,NCTEMP                                               00004910
      ICELL(JMA)=ISTMP(JMA)                                             00004920
  280 CONTINUE                                                          00004930
      GO TO 171                                                         00004940
                                                                        00004950
                                                                        00004960
        C U T P U T .                                                   00004970
                                                                        00004980
  300 CONTINUE                                                          00004990
      CALL CLPRT(IORDR,NOCEL,ICELL)                                     00005000
*                                                                       00005010
*                             DEBUG STATEMENT NUMBER PRINT.             00005020
*                                                                       00005030
      ISTMT=300                                                         00005040
      WRITE(IOUT,700)ISTMT                                              00005050
*                                                                       00005060
      WRITE(IOUT,400)                                                   00005070
  400 FORMAT(1H0)                                                       00005080
      GO TO 90                                                          00005090
                                                                        00005100
                         PRINT HEADINGS AND THEN S AND T FUNCTIONS.     00005110
                                                                        00005120
  500 CONTINUE                                                          00005130
      WRITE(IOUT,505)                                                   00005140
  505 FORMAT('0  LEVEL 3     LEVEL2')                                   00005150
*                                                                       00005160
*                             DEBUG STATEMENT NUMBER PRINT.             00005170
*                                                                       00005180
      ISTMT=500                                                         00005190
      WRITE(IOUT,700)ISTMT                                              00005200
*                                                                       00005210
                                                                        00005220
     VARIABLES.                                                         00005230
      ISNOW-THE 1-SET NOW BEING CONSIDERED.                             00005240
      ITFIN-THE 0-SET LAST FINISHED.                                    00005250
```

```
      ISNOW=1                                                          00005260
      ITFIN=0                                                          00005270
      IF(NOSF)550,550,510                                              00005280
  510 CONTINUE                                                         00005290
      IF(IS(ISNOW,2)-ITFIN)520,520,530                                 00005300
  520 ISET=1                                                           00005310
      CALL CLPR2(IORDR,IS(ISNOW,1))                                    00005320
      ISNOW=ISNOW+1                                                    00005330
      IF(ISNOW-NOSF)510,510,550                                        00005340
  530 ISET=0                                                           00005350
      ITFIN=ITFIN+1                                                    00005360
      CALL CLPR2(IORDR,IT(ITFIN,1))                                    00005370
      GO TO 510                                                        00005380
                                                                       00005390
  550 CONTINUE                                                         00005400
                                                                       00005410
                              PRINT INITIAL COVERING TABLE.            00005420
                                                                       00005430
      LAST=MTRMS(NOTRM)+2                                              00005440
      MIN1=MTRMS(1)+1                                                  00005450
      ICOV(NOSF,LAST)=MINUS                                            00005460
      N=NOSF                                                           00005470
      WRITE(IOUT,555)                                                  00005480
  555 FORMAT('0INITIAL COVERING TABLE:')                               00005490
*                                                                      00005500
*                             DEBUG STATEMENT NUMBER PRINT.            00005510
*                                                                      00005520
      ISTMT=555                                                        00005530
      WRITE(IOUT,700)ISTMT                                             00005540
*                                                                      00005550
      WRITE(IOUT,560)(MTRMS(JMA),JMA=1,NOTRM)                          00005560
  560 FORMAT(1HO,4X,32I3)                                              00005570
      DO 580 JP=1,N                                                    00005580
      WRITE(IOUT,570) JP,(ICOV(JP,MTRMS(JMA)+1),JMA=1,NOTRM)           00005590
  570 FORMAT(I6,32(1X,A1,1X))                                          00005600
                                                                       00005610
  580 CONTINUE                                                         00005620
      NP1=NOSF+1                                                       00005630
      NEPIS=1                                                          00005640
      IEPIS(1)=NOSF                                                    00005650
                                                                       00005660
                              PUT LAST SUBFUNCTION INTO IP1 ROW OF ICOV. 00005670
                                                                       00005680
      DO 590 JMA=1,NOTRM                                               00005690
      JTMP=MTRMS(JMA)+1                                                00005700
      ICOV(NP1,JTMP)=ICOV(NOSF,JTMP)                                   00005710
  590 CONTINUE                                                         00005720
                                                                       00005730
                                                                       00005740
                              TEST FOR ONLY ONE SUBFUNCTION FOUND DURING 00005750
                              FIRST 1-SET SEARCH.                      00005760
                                                                       00005770
      JSTOR=0                                                          00005780
      IF(NISF-1)300,600,630                                            00005790
  600 CONTINUE                                                         00005800
                                                                       00005810
                              CHECK FOR A ONE IN THE PRIMARY CUBE.     00005820
                                                                       00005830
      IF(IPRIM-1)630,620,630                                           00005840
  620 CONTINUE                                                         00005850
      JSTOR=1                                                          00005860
  630 CONTINUE                                                         00005870
      CALL       REDPI(NEPIS,JSTOR,ICOV,NOTRM,MTRMS,MCOV,NOSF,IEPIS,NP1)00005880
*                                                                      00005890
*                             DEBUG STATEMENT NUMBER PRINT.            00005900
*                                                                      00005910
```

```
      ISTMT=630                                                          00005920
      WRITE(IOUT,700)ISTMT                                               00005930
  700 FORMAT(1H+,125X,I5)                                                00005940
*                                                                        00005950
      IF(MCOV)2000,2000,1000                                             00005960
 1000 CONTINUE                                                           00005970
      CALL       RESET(JSTOP,NEPIS,IEPIS,IS,IT,NOTRM,MTRMS)              00005980
                                                                         00005990
      CALL CKFND(NEPIS,IEPIS,IS,IT,I3RD,ICROR,IFEED,N3RD,IFST)           00006000
                                                                         00006010
 1002 CONTINUE                                                           00006020
                                                                         00006030
      CALL M3RDI(NEPIS,IEPIS,N3RD,I3RD,IS,IT,IFEED,ISWTC)                00006040
                                                                         00006050
      CALL EL2IN(NEPIS,IEPIS,IS,IT,IFEED,JSTOP,IORDR)                    00006060
                                                                         00006070
      CALL EL3IN(N3RD,I3RD,IT,NEPIS,IEPIS,IS,IFEED,NOTRM,MTRMS,IORDR,    00006080
     $JSTOP,IFST)                                                        00006090
                                                                         00006100
      CALL DUPGT(N3RD,I3RD,NEPIS,IT,ISWTC,IFEED)                         00006110
                                                                         00006120
      ICUT=IPRT                                                          00006130
                                                                         00006140
                              OUTPUT THIS CIRCUIT.                       00006150
                                                                         00006160
      JCKT=NCKT+1                                                        00006170
      WRITE(IOUT,2205)JCKT                                               00006180
                                                                         00006190
      CALL CKPRT(NEPIS,I3RD,N3RD,IEPIS,IFEED,IS,IT,IORDR,IFST,NCKT,NGATS00006200
     *,NINPT)                                                            00006210
                                                                         00006220
                              ATTEMPT TO FIND ALTERNATE                  00006230
                              CIRCUIT.                                   00006240
                                                                         00006250
      ICUT=ISWTC                                                         00006260
      N2ND=NEPIS                                                         00006270
      WRITE(IOUT,1010)(IEPIS(JMA),JMA=1,NEPIS)                           00006280
 1010 FORMAT(' THE ESSENTIAL PI''S ARE',2X,10I5/                         00006290
     *(25X,10I5/))                                                       00006300
                                                                         00006310
      CALL ALTCKT(IEPIS,IS,IT,I3RD,IFEED,IFLAG,JSTOP,N3RD,               00006320
     *N2ND,NEPIS,IORDR)                                                  00006330
                                                                         00006340
      IF(IFLAG)2400,2400,1002                                            00006350
                                                                         00006360
 2000 CONTINUE                                                           00006370
                              INITIALIZE I2ND'S FIRST PLACES TO ESSENTIAL00006380
                              PI'S.                                      00006390
                                                                         00006400
      DO 2010 JMA=1,NEPIS                                                00006410
      I2ND(JMA)=IEPIS(JMA)                                               00006420
 2010 CONTINUE                                                           00006430
                                                                         00006440
                              INITIALIZE LPINE=LIST OF PRIME IMPLICANTS,N00006450
                              ESSENTIAL.                                 00006460
                                                                         00006470
      NLPI=0                                                             00006480
      DO 2030 JMA=1,NOSF                                                 00006490
      IF(ICOV(JMA,LAST)-MINUS)2020,2030,2020                             00006500
 2020 CONTINUE                                                           00006510
      NLPI=NLPI+1                                                        00006520
      LPINE(NLPI)=JMA                                                    00006530
 2030 CONTINUE                                                           00006540
                                                                         00006550
C                                                                        00006560
                              INITIALIZE CONSTANTS.                      00006570
```

```
                                                                        00006580
      NMBR-NUMBER OF PI'S CURRENTLY BEING CONSIDERED.                   00006590
      NSTOP-LAST NONESSENTIAL PI TO BE CONSIDERED.                      00006600
      LPTST-LIST OF NONESSENTIAL PI'S CURRENTLY BEING CONSIDERED.       00006610
                                                                        00006620
      NMBR=1                                                            00006630
      JNMBR=NLPI                                                        00006640
      LPTST(1)=1                                                        00006650
      ISTOP=NLPI                                                        00006660
                                                                        00006670
 2040 DO 2050 JMA=1,NP1                                                 00006680
      DO 2050 JMA2=1,LAST                                               00006690
      ICOVT(JMA,JMA2)=ICOV(JMA,JMA2)                                    00006700
 2050 CONTINUE                                                          00006710
      NTMP=1                                                            00006720
 2060 N2NDT=NEPIS+NTMP                                                  00006730
 2065 I2ND(N2NDT)=LPINE(LPTST(NTMP))                                    00006740
      JSTOP=LPINE(LPTST(NTMP))                                          00006750
                                                                        00006760
                              N2NDT REDUCED BECAUSE REDPI INCREMENTS IT. 00006770
                                                                        00006780
      N2NDT=N2NDT-1                                                     00006790
      CALL REDPI(N2NDT,JSTOP,ICOVT,NOTRM,MTRMS,MCOV,NOSF,I2ND,NP1)      00006800
      WRITE(IOUT,2222)JNMBR,NMBR,NTMP,ISTOP,LPTST(NTMP)                 00006810
     *,LPINE(LPTST(NTMP)),N2NDT                                         00006820
*                                                                       00006830
*                             DEBUG STATEMENT NUMBER PRINT.             00006840
*                                                                       00006850
      ISTMT=2065                                                        00006860
      WRITE(IOUT,700)ISTMT                                              00006870
*                                                                       00006880
      IF(NTMP-NMBR)2070,2080,2080                                       00006890
 2070 NTMP=NTMP+1                                                       00006900
 2074 CONTINUE                                                          00006910
      IF(ICOVT(LPINE(LPTST(NTMP)),LAST)-MINUS)2060,2075,2060            00006920
 2075 LPTST(NTMP)=LPTST(NTMP)+1                                         00006930
      IF(LPTST(NTMP)-ISTOP)2074,2060,2210                               00006940
 2080 WRITE(IOUT,2090)(I2ND(JMA),JMA=1,N2NDT)                           00006950
 2090 FORMAT('ONOW CONSIDERING',2015)                                   00006960
*                                                                       00006970
*                             DEBUG STATEMENT NUMBER PRINT.             00006980
*                                                                       00006990
      ISTMT=2080                                                        00007000
      WRITE(IOUT,700)ISTMT                                              00007010
*                                                                       00007020
      WRITE(IOUT,2222)JNMBR,NMBR,NTMP,ISTOP,LPTST(NTMP)                 00007030
     *,LPINE(LPTST(NTMP)),N2NDT                                         00007040
      IF(MCOV)2300,2300,2200                                            00007050
 2200 CONTINUE                                                          00007060
                                                                        00007070
      CALL RESET(JSTOP,N2NDT,I2ND,IS,IT,NOTRM,MTRMS)                    00007080
      WRITE(IOUT,2222)JNMBR,NMBR,NTMP,ISTOP,LPTST(NTMP)                 00007090
     *,LPINE(LPTST(NTMP)),N2NDT                                         00007100
*                                                                       00007110
*                             DEBUG STATEMENT NUMBER PRINT.             00007120
*                                                                       00007130
      ISTMT=2200                                                        00007140
      WRITE(IOUT,700)ISTMT                                              00007150
*                                                                       00007160
      CALL CKFND(N2NDT,I2ND,IS,IT,I3RD,IORDR,IFEED,N3RD,IFST)           00007170
                                                                        00007180
                              FILL ITT, A TEMPORARY IT.                 00007190
                                                                        00007200
      DO 2203 JMA=1,NOTF                                                00007210
      ITT(JMA)=IT(JMA,1)                                                00007220
 2203 CONTINUE                                                          00007230
```

```
                                                                      00007240
2204 CONTINUE                                                         00007250
     JCKT=NCKT+1                                                      00007260
                                                                      00007270
     CALL M3RDI(N2NDT,I2ND,N3RD,I3RD,IS,ITT,IFEED,ISWTC)              00007280
                                                                      00007290
     CALL EL2IN(N2NDT,I2ND,IS,ITT,IFEED,JSTOP,IORDR)                  00007300
      CALL EL3IN(N3RD,I3RD,ITT,N2NDT,I2ND,IS,IFEED,NOTRM,MTRMS,IORDR,  00007310
    *JSTOP,IFST)                                                      00007320
     N3RDT=N3RD                                                       00007330
     CALL DUPGT(N3RD,I3RD,N2NDT,ITT,ISWTC,IFEED)                      00007340
     ICUT=IPRT                                                        00007350
                                                                      00007360
     WRITE(IOUT,2205)JCKT                                             00007370
2205 FORMAT('0THIS IS CKT #',I2)                                      00007380
     CALL CKPRT(N2NDT,I3RD,N3RD,I2ND,IFEED,IS,ITT,IORDR,IFST,NCKT,NGATS00007390
    *,NINPT)                                                          00007400
                                                                      00007410
                              ATTEMPT TO FIND ALTERNATE               00007420
                              CIRCUIT.                                00007430
                                                                      00007440
     ICUT=ISWTC                                                       00007450
     CALL ALTCKT(I2ND,IS,ITT,I3RD,IFEED,IFLAG,JSTOP,N3RD,             00007460
    *N2NDT,NEPIS,IORDR)                                               00007470
                                                                      00007480
     IF(IFLAG)2207,2207,2204                                          00007490
2207 CONTINUE                                                         00007500
     N3RD=N3RDT                                                       00007510
     IOUT=ISWTC                                                       00007520
     WRITE(IOUT,2222)JNMBR,NMBR,NTMP,ISTOP,LPTST(NTMP)                00007530
    *,LPINE(LPTST(NTMP)),N2NDT                                        00007540
     MCOV=0                                                           00007550
     IF(NMBR-JNMBR)2230,2230,2210                                     00007560
2210 CONTINUE                                                         00007570
                                                                      00007580
                              HAVING FOUND A COVERAGE, TRY NEW COMBINATI00007590
                                                                      00007600
     IF(JNMBR-1)2400,2400,2220                                        00007610
2220 JNMBR=JNMBR-1                                                    00007620
     NTMP=1                                                           00007630
     NMBR=1                                                           00007640
     LPTST(1)=LPTST(1)+1                                              00007650
     WRITE(IOUT,2222)JNMBR,NMBR,NTMP,ISTOP,LPTST(NTMP)                00007660
    *,LPINE(LPTST(NTMP)),N2NDT                                        00007670
2222 FORMAT('0JNMBR=',I3,' NMBR=',I3,' NTMP=',I3,' ISTOP=',I3,' LPTST='00007680
    *,I3,' LPINE=',I3,' N2NDT=',I3)                                   00007690
*                                                                     00007700
*                             DEBUG STATEMENT NUMBER PRINT.           00007710
*                                                                     00007720
     ISTMT=2220                                                       00007730
     WRITE(IOUT,700)ISTMT                                             00007740
*                                                                     00007750
     IF(LPTST(1)-ISTOP)2040,2040,2400                                 00007760
2230 CONTINUE                                                         00007770
     LPTST(NTMP)=LPTST(NTMP)+1                                        00007780
     WRITE(IOUT,2222)JNMBR,NMBR,NTMP,ISTOP,LPTST(NTMP)                00007790
    *,LPINE(LPTST(NTMP)),N2NDT                                        00007800
*                                                                     00007810
*                             DEBUG STATEMENT NUMBER PRINT.           00007820
*                                                                     00007830
     ISTMT=2230                                                       00007840
     WRITE(IOUT,700)ISTMT                                             00007850
*                                                                     00007860
     IF(LPTST(NTMP)-ISTOP)2040,2040,2240                              00007870
2240 NTMP=NTMP-1                                                      00007880
     IF(NTMP)2400,2400,2245                                           00007890
```

```
2245 CONTINUE                                                          00007900
     NMBR=NMBR-1                                                       00007910
     LPTST(NTMP)=LPTST(NTMP)+1                                         00007920
     WRITE(IOUT,2222)JNMBR,NMBR,NTMP,ISTOP,LPTST(NTMP)                 00007930
    *,LPINE(LPTST(NTMP)),N2NDT                                         00007940
*                                                                      00007950
*                          DEBUG STATEMENT NUMBER PRINT.               00007960
*                                                                      00007970
     ISTMT=2240                                                        00007980
     WRITE(IOUT,700)ISTMT                                              00007990
*                                                                      00008000
2250 IF(NMBR)2210,2210,2040                                            00008010
                                                                       00008020
2300 CONTINUE                                                          00008030
                                                                       00008040
                          MINTERMS HAVE NOT BEEN COVERED, INCREASE     00008050
                          NUMBER OF NONESSENTIAL PI'S TO TRY.          00008060
                                                                       00008070
     NMBR=NMBR+1                                                       00008080
     IF(NMBR-JNMBR)2310,2310,2210                                      00008090
2310 CONTINUE                                                          00008100
     NTMP=NTMP+1                                                       00008110
     LPTST(NTMP)=LPTST(NTMP-1)+1                                       00008120
2320 CONTINUE                                                          00008130
     IF(LPTST(NTMP)-ISTOP)2330,2330,2210                               00008140
2330 CONTINUE                                                          00008150
     WRITE(IOUT,2222)JNMBR,NMBR,NTMP,ISTOP,LPTST(NTMP)                 00008160
    *,LPINE(LPTST(NTMP)),N2NDT                                         00008170
     IF(ICOVT(LPINE(LPTST(NTMP)),LAST)-MINUS)2060,2340,2060            00008180
2340 CONTINUE                                                          00008190
     LPTST(NTMP)=LPTST(NTMP)+1                                         00008200
     GO TO 2320                                                        00008210
                                                                       00008220
                          PRINT OUT CKT ANALYSIS.                      00008230
                                                                       00008240
2400 ICUT=IPRT                                                         00008250
     WRITE(IOUT,2410)                                                  00008260
2410 FORMAT('0CKT#    # OF GATES    # OF INPUTS')                      00008270
     MING=2000                                                         00008280
     DO 2490 JMA=1,NCKT                                                00008290
     WRITE(IOUT,2420)JMA,NGATS(JMA),NINPT(JMA)                         00008300
2420 FORMAT(I5,2I13)                                                   00008310
     IF(NGATS(JMA)-MING)2470,2430,2490                                 00008320
2430 IF(NINPT(JMA)-MININ)2480,2485,2490                                00008330
2470 MING=NGATS(JMA)                                                   00008340
2480 MININ=NINPT(JMA)                                                  00008350
     MCKT=JMA                                                          00008360
     NMCKT=0                                                           00008370
2485 NMCKT=NMCKT+1                                                     00008380
     MCKTS(NMCKT)=JMA                                                  00008390
2490 CONTINUE                                                          00008400
     IF(NMCKT-1)2495,2495,2510                                         00008410
2495 CONTINUE                                                          00008420
     WRITE(IOUT,2500)MCKT,MING,MININ                                   00008430
2500 FORMAT('0OBVIOUSLY CKT#',I5,' IS THE MINIMUM WITH ONLY',I5,' GATES00008440
    * AND ONLY',I5,' INPUTS')                                          00008450
     GO TO 90                                                          00008460
2510 WRITE(IOUT,2520)MING,MININ,(MCKTS(JMA),JMA=1,NMCKT)               00008470
2520 FORMAT('0A TIE AMONG CIRCUITS WITH',I3,' GATES AND',              00008480
    *I3,' INPUTS'/' THE WINNING CIRCUITS ARE:',10I5)                   00008490
     GO TO 90                                                          00008500
                                                                       00008510
                                                                       00008520
                          PRINT NUMBER OF CASES PROCESSED.             00008530
                                                                       00008540
8990 WRITE(IOUT,8991)NCASES                                            00008550
```

```
8991 FORMAT(' FINISHED PROCESSING',I5,' TEST CASES OR EXAMPLES')        00008560
                                                                        00008570
9000 WRITE(IOUT,9001)JRUN                                               00008580
     WRITE(INRUN,9001)JRUN                                              00008581
9001 FORMAT('1 ALL DONE',10X,'JRUN=',I10)                               00008590
     STOP                                                               00008600
                                                                        00008610
                                                                        00008620
9010 WRITE(IOUT,9011)NSTOP,MOCEL                                        00008630
                                                                        00008640
9011 FORMAT('1TERMINATE**ZERO SET EXHAUSTED BUT NO MORE 1-SET CELLS     00008650
    *FOUND******'//' **NSTOP=',I5,'    MOCEL=',I5)                     00008660
                                                                        00008670
                                                                        00008680
                                                                        00008690
     STOP                                                               00008700
     END                                                                00008710
                                                                        00008720
                                                                        00008730
                                                                        00008740
                                                                        00008750
                                                                        00008760
/ / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / 00008770
                                                                        00008780
                                                                        00008790
     SUBROUTINE CLPR2(ICRDR,MIN)                                        00008800
                                                                        00008810
       SUBROUTINE TO PRINT OUT THE BINARY REP. OF A CELL.               00008820
                                                                        00008830
                                                                        00008840
     DIMENSION IBIN(20)                                                 00008850
     COMMON IN,IOUT,MAX,ISET                                            00008860
                                                                        00008870
   VARIABLES:                                                           00008880
     ICRDR-ORDER OF THE FUNCTION WITH THIS CELL.                        00008890
     IN-LOGICAL UNIT NUMBER FOR THE INPUT DEVICE.                       00008900
     IOUT-LOGICAL UNIT NUMBER FOR THE OUTPUT DEVICE.                    00008910
     MIN-MINIMUM VALUE OF THIS CELL.                                    00008920
     IC-VALUE OF THE JMA2TH BIT OF MIN.                                 00008930
     MAX-MAXIMUM VALUE OF THIS CELL.                                    00008940
     IP-VALUE OF THE JMA2TH BIT OF MAX.                                 00008950
     IBIN-ARRAY CONTAINING BINARY REPRESENTATION OF THIS CELL.          00008960
                                                                        00008970
     DATA IONE,IZERO,IDASH/'1','0','-'/                                 00008980
                                                                        00008990
     DO 100 JMA=1,ICRDR                                                 00009000
          ITMP=2**(JMA-1)                                               00009010
          JMA2=ICRDR+1-JMA                                              00009020
          IC=IAND(MIN,ITMP)                                             00009030
          IP=IAND(MAX,ITMP)                                             00009040
                                                                        00009050
          IF(IC+IP-ITMP)50,60,70                                        00009060
  50      IBIN(JMA2)=IZERO                                              00009070
          GO TO 100                                                     00009080
  60      IBIN(JMA2)=IDASH                                              00009090
          GO TO 100                                                     00009100
  70      IBIN(JMA2)=IONE                                               00009110
 100 CONTINUE                                                           00009120
                                                                        00009130
     IF(ISET)200,220,200                                                00009140
 200 WRITE(IOUT,210)(IBIN(JMA),JMA=1,ICRDR)                             00009150
 210 FORMAT(15X,20A1)                                                   00009160
     RETURN                                                             00009170
 220 WRITE(IOUT,230)(IBIN(JMA),JMA=1,ICRDR)                             00009180
 230 FORMAT(3X,20A1)                                                    00009190
     RETURN                                                             00009200
```

```
      END                                                                00009210
                                                                         00009220
                                                                         00009230
                                                                         00009240
C/////////// ////////////////////////////////////////////////////////00009250
                                                                         00009260
                                                                         00009270
      SUBROUTINE SUBFN(N,LIST,NOCEL,ICELL,IP,NIP)                        00009280
                                                                         00009290
                                                                         00009300
     THIS SUBROUTINE FINDS THE MEMBERS OF THE X-SET.                     00009310
                                                                         00009320
      N-NUMBER OF VALUES IN LIST.                                        00009330
      LIST-ARRAY OF VALUES(MINTERMS OR MAXTERMS) TO BE CIRCLED ON.       00009340
      VARIABLES:                                                         00009350
                                                                         00009360
      IP-PREVIOUSLY CREATED CELLS.                                       00009370
      IN-LOGICAL UNIT NUMBER FOR THE INPUT DEVICE.                       00009380
      IOUT-LOGICAL UNIT NUMBER FOR THE OUTPUT DEVICE.                    00009390
      NIP-NUMBER OF PREVIOUS CELLS.                                      00009400
         IPRIM-THE PRIMARY CUBE.                                         00009410
         ISET-TYPE OF SET TO BE PROCESSED(1-SET OR 0-SET).               00009420
                                                                         00009430
      JANUARY 16, 1977                                                   00009440
                                                                         00009450
      DIMENSION IP(30,2)                                                 00009460
      DIMENSION IFUNC(40,256)                                            00009470
      DIMENSION LIST(1),ICELL(1)                                         00009480
      COMMON IN,IOUT,IPRIM,ISET,IFUNC,LEVEL                              00009490
                                                                         00009500
      NOT=1-ISET                                                         00009510
      NOCEL=0                                                            00009520
                                                                         00009530
      DO 100 JMA=1,N                                                     00009540
      LOW=LIST(JMA)                                                      00009550
      LOW=IAND(LOW,IPRIM)                                                00009560
                                                                         00009570
                              TEST IF ALL VERTICES ARE IN X-SET.         00009580
                                                                         00009590
      JTEMP=LOW                                                          00009600
   30 IF(JTEMP-IPRIM)40,40,100                                           00009610
   40 ITEST=IAND(JTEMP,LOW)                                              00009620
      IF(ITEST-LOW)60,50,60                                              00009630
   50 IF(IFUNC(LEVEL,JTEMP+1)-NOT)60,1000,60                             00009640
   60 JTEMP=JTEMP+1                                                      00009650
      GO TO 30                                                           00009660
  100 CONTINUE                                                           00009670
                                                                         00009680
                              TEST IF THIS CELL IS COVERED.              00009690
                                                                         00009700
      IF(NOCEL)110,210,110                                               00009710
  110 CONTINUE                                                           00009720
      DO 200 JTEMP=1,NOCEL                                               00009730
      ITEST=IAND(ICELL(JTEMP),LOW)                                       00009740
      IF(ITEST-ICELL(JTEMP))200,1000,200                                 00009750
  200 CONTINUE                                                           00009760
                                                                         00009770
                              TEST IF THIS CELL WAS PREVIOUSLY CREATED.  00009780
                                                                         00009790
  210 CONTINUE                                                           00009800
      IF(NIP)250,250,220                                                 00009810
  220 DO 240 JTEMP=1,NIP                                                 00009820
      ITEST=IAND(LOW,IP(JTEMP,1))                                        00009830
      IF(ITEST-IP(JTEMP,1))240,1000,240                                  00009840
  240 CONTINUE                                                           00009850
                                                                         00009860
```

```
 250 CONTINUE                                                          00009870
     NOCEL=NOCEL+1                                                     00009880
                                                                       00009890
     ICELL(NOCEL)=LOW                                                  00009900
1000 CONTINUE                                                          00009910
                                                                       00009920
     RETURN                                                            00009930
     END                                                               00009940
                                                                       00009950
                                                                       00009960
                          E N D   O F   S U B F N .                    00009970
                                                                       00009980
/ / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / 00009990
                                                                       00010000
                                                                       00010010
                                                                       00010020
                                                                       00010030
     SUBROUTINE CLPRT(IORDR,NOCEL,ICELL)                               00010040
                                                                       00010050
      THIS SUBROUTINE PRINTS A LIST OF THE CELLS FOUND.                00010060
                                                                       00010070
     JANUARY 16, 1977                                                  00010080
                                                                       00010090
     DIMENSION IFUNC(40,512)                                           00010100
     DIMENSION ICELL(1),IBIN(12)                                       00010110
     COMMON IN,IOUT,IPRIM,ISET,IFUNC,LEVEL                             00010120
     DATA IONE,IZERO,IDASH/'1','0','-'/                                00010130
     VARIABLES:                                                        00010140
        IPRIM-THE PRIMARY CUBE.                                        00010150
        ISET-TYPE OF SET TO BE PROCESSED(1-SET OR 0-SET).              00010160
     IOUT-LOGICAL UNIT NUMBER FOR THE OUTPUT DEVICE.                   00010170
     IN-LOGICAL UNIT NUMBER FOR THE INPUT DEVICE.                      00010180
                                                                       00010190
     WRITE(IOUT,10)ISET                                                00010200
  10 FORMAT('0 THE',I2,'-SET IMPLICANTS ARE:'/ ' CELL MIN  CELL MAX    00010210
    * BINARY FORM')                                                    00010220
     IF(NOCEL)20,20,40                                                 00010230
  20 WRITE(IOUT,30)                                                    00010240
  30 FORMAT('0 I DID NOT WISH TO PRINT ANYTHING ANYWAY#@*&')           00010250
     RETURN                                                            00010260
  40 CONTINUE                                                          00010270
                                                                       00010280
     DO 110 JMA=1,NOCEL                                                00010290
                                                                       00010300
     DO 90 JTEMP=1,IORDR                                               00010310
     ITMP=2**(JTEMP-1)                                                 00010320
     JMAII=IORDR+1-JTEMP                                               00010330
     IC=IAND(ICELL(JMA),ITMP)                                          00010340
     IP=IAND(IPRIM,ITMP)                                               00010350
     IF(IC+IP-ITMP)50,60,70                                            00010360
  50 IBIN(JMAII)=IZERO                                                 00010370
     GO TO 90                                                          00010380
  60 IBIN(JMAII)=IDASH                                                 00010390
     GO TO 90                                                          00010400
  70 IBIN(JMAII)=IONE                                                  00010410
  90 CONTINUE                                                          00010420
     WRITE(IOUT,100)ICELL(JMA),IPRIM,(IBIN(J),J=1,IORDR)               00010430
 100 FORMAT(1H ,I7,I10,6X,12A1)                                        00010440
 110 CONTINUE                                                          00010450
     RETURN                                                            00010460
     END                                                               00010470
                                                                       00010480
                                                                       00010490
/ / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / 00010500
                                                                       00010510
                                                                       00010520
```

```
                                                                      00010530
                                                                      00010540
      SUBROUTINE CELL(MIN,MAX,IBIN,NSIZE,IORDR)                       00010550
      DIMENSION IBIN(1)                                               00010560
      DATA IONE,IZERO,IDASH/'1','0','-'/                              00010570
     VARIABLES.                                                       00010580
      MIN-CELL MINIMUM.                                               00010590
      MAX-CELL MAXIMUM.                                               00010600
      IBIN-ALPHAMERIC ARRAY OF THE BINARY REPRESENTATION OF THE CELL. 00010610
      NSIZE-SIZE OF THE CELL(NUMBER OF DASHES).                       00010620
      IORDR-ORDER OF THE CELL(NUMBER OF DIGITS).                      00010630
                                                                      00010640
      NSIZE=0                                                         00010650
      DO 100 JMA=1,IORDR                                              00010660
      ITMP=2**(JMA-1)                                                 00010670
      JMA2=IORDR+1-JMA                                                00010680
      IC=IAND(MIN,ITMP)                                               00010690
      IP=IAND(MAX,ITMP)                                               00010700
      IF(IC+IP-ITMP)50,60,70                                          00010710
   50 IBIN(JMA2)=IZERO                                                00010720
      GO TO 100                                                       00010730
   60 IBIN(JMA2)=IDASH                                                00010740
      NSIZE=NSIZE+1                                                   00010750
      GO TO 100                                                       00010760
   70 IBIN(JMA2)=IONE                                                 00010770
  100 CONTINUE                                                        00010780
      RETURN                                                          00010790
      END                                                             00010800
                                                                      00010810
                                                                      00010820
                                                                      00010830
                                                                      00010840
/ / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / 00010850
                                                                      00010860
                                                                      00010870
                                                                      00010880
      SUBROUTINE CKFND(N2ND,I2ND,IS,IT,I3RD,IORDR,IFEED,N3RD,IFST)    00010890
                                                                      00010900
   VARIABLES:                                                         00010910
      IWUSE-I WILL USE THIS LIST AS 3RD LEVEL GATES.                  00010920
      NWUSE-NUMBER THAT WILL BE USED.                                 00010930
      N2ND-NUMBER OF 2ND LEVEL GATES.                                 00010940
      I2ND-LIST OF SECOND LEVEL GATES.                                00010950
      IS-LIST OF SUBFUNCTIONS.                                        00010960
      IT-LIST OF RESIDUAL FUNCTIONS.                                  00010970
      I3RD-LIST OF 3RD LEVEL GATES.                                   00010980
      N3RD-NUMBER OF 3RD LEVEL GATES.                                 00010990
      IFEED-2 DIMENSIONAL LIST OF 3RD LEVELS FEEDING 2ND LEVEL.       00011000
      IGATE-2ND LEVEL GATE NOW BEING CONSIDERED.                      00011010
      IGAT3-3RD LEVEL GATE BEING CONSIDERED.                          00011020
      MIN-LOWER BOUND OF CELL FOR CURRENT GATE.                       00011030
      N2T-NUMBER OF 2ND LEVEL GATES TEMPORARY.                        00011040
      IREF-REFERENCE FOR REDUCED CIRCUIT.                             00011050
      IFST-MIN FOR DIRECT INPUT CUBE INTO FIRST LEVEL GATE.           00011060
                                                                      00011070
      DIMENSION I2ND(1),I3RD(1),IS(30,1),IFEED(30,1),IT(30,1)         00011080
      DIMENSION IFUNC(40,256)                                         00011090
      DIMENSION NOUSE(20),IWUSE(20)                                   00011100
      COMMON IN,IOUT,IPRIM,ISET,IFUNC,LEVEL                           00011110
      IREF=0                                                          00011120
      N3RD=0                                                          00011130
      N2T=N2ND                                                        00011140
      IFST=-1                                                         00011150
      DO 500 JMA=1,N2ND                                               00011160
      NWUSE=0                                                         00011170
      IFEED(JMA-IREF,1)=0                                             00011180
```

```
      IGATE=I2ND(JMA)                                                   00011190
      MIN=IS(IGATE,1)                                                   00011200
      JLEVL=IS(IGATE,3)                                                 00011210
                                                                        00011220
                              FIND FEEDING 3RD LEVEL GATES FOR THIS GATE.00011230
                                                                        00011240
      IBEGIN=MIN+1                                                      00011250
      ISTOP=IPRIM+1                                                     00011260
      DO 200 JMA2=IBEGIN,ISTOP                                          00011270
      JTMP=JMA2-1                                                       00011280
      ITEST=IAND(JTMP,MIN)                                              00011290
      IF(ITEST-MIN)200,10,200                                           00011300
   10 CONTINUE                                                          00011310
      IF(IFUNC(1,JMA2))15,15,200                                        00011320
   15 CONTINUE                                                          00011330
      IGAT3=IFUNC(JLEVL,JMA2)                                           00011340
      IF(IGAT3)20,200,200                                               00011350
   20 CONTINUE                                                          00011360
                                                                        00011370
                              TEST AND ADD THIS GATE TO 3RD LEVEL LIST. 00011380
                                                                        00011390
                                                                        00011400
                              CHECK IF THIS GATE HAS BEEN USED, AND ADD I00011410
                              TO IWUSE IF IT HASN'T BEEN USED.          00011420
                                                                        00011430
   45 IF(NWUSE)70,70,50                                                 00011440
                                                                        00011450
   50 DO 60 JMA3=1,NWUSE                                                00011460
      IF(IWUSE(JMA3)-IGAT3)60,200,60                                    00011470
   60 CONTINUE                                                          00011480
                                                                        00011490
   70 NWUSE=NWUSE+1                                                     00011500
      IWUSE(NWUSE)=IGAT3                                                00011510
  200 CONTINUE                                                          00011520
                                                                        00011530
                              IF MIN=0 AND NWUSE=1 THIS IS REDUCIBLE.   00011540
                                                                        00011550
      IF(MIN)205,204,205                                                00011560
  204 IF(NWUSE-1)205,350,205                                            00011570
  205 CONTINUE                                                          00011580
                                                                        00011590
      IF(NWUSE)500,500,210                                              00011600
  210 CONTINUE                                                          00011610
      ITEMP=1                                                           00011620
      DO 300 JMA2=1,NWUSE                                               00011630
      IFEED(JMA-IREF,1)=ITEMP                                           00011640
      ITEMP=ITEMP+1                                                     00011650
      IGAT3=-IWUSE(JMA2)                                                00011660
      IFEED(JMA-IREF,ITEMP)=IGAT3                                       00011670
      IF(N3RD)290,290,240                                               00011680
  240 CONTINUE                                                          00011690
      DO 280 JMA3=1,N3RD                                                00011700
      IF(IGAT3-I3RD(JMA3))280,300,280                                   00011710
  280 CONTINUE                                                          00011720
  290 CONTINUE                                                          00011730
      N3RD=N3RD+1                                                       00011740
      I3RD(N3RD)=IGAT3                                                  00011750
  300 CONTINUE                                                          00011760
      GO TO 500                                                         00011770
  350 CONTINUE                                                          00011780
      IFST=IT(-IWUSE(1),1)                                              00011790
      N2T=N2T-1                                                         00011800
      IREF=IREF+1                                                       00011810
      IF(JMA-N2T)360,360,380                                            00011820
                                                                        00011830
                              MOVE 2ND LEVEL GATES UP TO COVER ELIMINATED00011840
```

```
                              GATE.                                          00011850
                                                                             00011860
360 DO 370 JMA2=JMA,N2T                                                      00011870
    I2ND(JMA2)=I2ND(JMA2+1)                                                  00011880
370 CONTINUE                                                                 00011890
                                                                             00011900
380 CONTINUE                                                                 00011910
500 CONTINUE                                                                 00011920
                                                                             00011930
    N2ND=N2T                                                                 00011940
    RETURN                                                                   00011950
                                                                             00011960
    END                                                                      00011970
                                                                             00011980
                                                                             00011990
/ / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / 00012000
                                                                             00012010
                                                                             00012020
    SUBROUTINE CKPRT(NEPIS,I3RD,N3RD,I2ND,IFEED,IS,IT,IORDR,IFST,NCKT,00012030
   *NGATS,NINPT)                                                             00012040
                                                                             00012050
        VARIABLES:                                                           00012060
    NEPIS-NUMBER OF ESSENTIAL PI'S=ALSO NUMBER OF 2ND LEVEL GATES.           00012070
    I2ND-LIST OF 2ND LEVEL GATES=IEPIS IN MAIN PGM.                          00012080
    IFEED-LIST OF 3RD LEVEL GATES FEEDING 2ND LEVEL GATES.                   00012090
    N3RD-NUMBER OF THIRD LEVEL GATES.                                        00012100
    I3RD-LIST OF 3RD LEVEL GATES.                                            00012110
    IS-LIST OF SUBFUNCTIONS.                                                 00012120
    NCKT-NUMBER OR CIRCUITS.                                                 00012130
    NGATS-NUMBER OF GATES FOR THIS CIRCUIT.                                  00012140
    NINPT-NUMBER OF INPUTS FOR THIS CIRCUIT.                                 00012150
                                                                             00012160
    DIMENSION NGATS(1),NINPT(1)                                              00012170
    DIMENSION I3RD(1),I2ND(1),IFEED(30,1),IS(30,1),IT(30,1)                  00012180
    DIMENSION IBIN3(10),IBIN2(10)                                            00012190
    COMMON IN,IOUT,IPRIM                                                     00012200
    DATA ICOMT/',T'/                                                         00012210
                                                                             00012220
    NCKT=NCKT+1                                                              00012230
    NGATS(NCKT)=N3RD+NEPIS+1                                                 00012240
    NINPT(NCKT)=NEPIS                                                        00012250
    MAX=IPRIM                                                                00012260
    LOOP=NEPIS                                                               00012270
    IF(LOOP-N3RD)10,20,20                                                    00012280
 10 LOOP=N3RD                                                                00012290
 20 CONTINUE                                                                 00012300
                                                                             00012310
                              WRITE HEADINGS.                                00012320
                                                                             00012330
    WRITE(IOUT,50)                                                           00012340
 50 FORMAT('03RD LEVEL GATES',10X,' 2ND LEVEL GATES,AND INPUTS FROM 3R00012350
   *D LEVEL',10X,'DIRECT INPUTS TO FIRST LEVEL')                             00012360
                                                                             00012370
    DO 500 JMA=1,LOOP                                                        00012380
                                                                             00012390
                              PRINT 3RD LEVEL.                               00012400
                                                                             00012410
    IF(N3RD-JMA)240,100,100                                                  00012420
100 I13=I3RD(JMA)                                                            00012430
    CALL CELL(IT(I13,1),MAX,IBIN3,NSIZE,ICRDR)                               00012440
    NINPT(NCKT)=NINPT(NCKT)+(IORDR-NSIZE)                                    00012450
    WRITE(IOUT,120)IT3,(IBIN3(JMA2),JMA2=1,IORDR)                            00012460
120 FORMAT(' T',I2,1X,10A1)                                                  00012470
    GO TO 250                                                                00012480
240 WRITE(IOUT,241)                                                          00012490
241 FORMAT(' ')                                                              00012500
```

```
  250 CONTINUE                                                          00012510
                                                                        00012520
                              PRINT 2ND LEVEL.                          00012530
                                                                        00012540
      IF(NEPIS-JMA)300,260,260                                          00012550
  260 CONTINUE                                                          00012560
      CALL CELL(IS(I2ND(JMA),1),IPRIM,IBIN2,NSIZE,IORDR)                00012570
      NTS=IFEED(JMA,1)                                                  00012580
      NINPT(NCKT)=NINPT(NCKT)+NTS+(IORDR-NSIZE)                         00012590
      WRITE(IOUT,270)(IBIN2(JMA2),JMA2=1,IORDR)                         00012600
  270 FORMAT(1H+,27X,10A1)                                              00012610
      IF(NTS)300,300,280                                                00012620
  280 NTS=NTS+1                                                         00012630
      WRITE(IOUT,290)(ICOMT,IFEED(JMA,JMA2),JMA2=2,NTS)                 00012640
  290 FORMAT(1H+,(37X,9(A2,I2)/))                                       00012650
                                                                        00012660
                              PRINT OUT FIRST LEVEL.                    00012670
                                                                        00012680
  300 CONTINUE                                                          00012690
      IF(1-JMA)500,310,310                                              00012700
  310 CONTINUE                                                          00012710
      IF(IFST)500,320,320                                               00012720
  320 CALL CELL(IFST,IPRIM,IBIN2,NSIZE,IORDR)                           00012730
      NINPT(NCKT)=NINPT(NCKT)+(IORDR-NSIZE)                             00012740
      WRITE(IOUT,330)(IBIN2(JMA2),JMA2=1,IORDR)                         00012750
  330 FORMAT(1H+,80X,10A1)                                              00012760
  500 CONTINUE                                                          00012770
                                                                        00012780
      RETURN                                                            00012790
      END                                                               00012800
                                                                        00012810
                                                                        00012820
  / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / /   00012830
                                                                        00012840
                                                                        00012850
                                                                        00012860
      SUBROUTINE RESET(JSTOP,NEPIS,IEPIS,IS,IT,NOTRM,MTRMS)             00012870
                                                                        00012880
      DIMENSION IS(30,1),IT(30,1)                                       00012890
      DIMENSION IFUNC(40,256)                                           00012900
      DIMENSION ICOV(20,256),IEPIS(1),MTRMS(1)                          00012910
      COMMON IN,IOUT,IPRIM,ISET,IFUNC,LEVEL                             00012920
                                                                        00012930
      ISTRT=1                                                           00012940
                                                                        00012950
                              SET SUBFUNCTION LEVEL INDICES.            00012960
                                                                        00012970
      IF(IS(1,3)-2)1010,1010,1020                                       00012980
 1010 LSTMP=2                                                           00012990
      GO TO 1030                                                        00013000
 1020 LSTMP=3                                                           00013010
 1030 CONTINUE                                                          00013020
                                                                        00013030
                              REGENERATE IFUNC VALUES.                  00013040
C                                                                       00013050
      DO 1300 JMA=2,LEVEL                                               00013060
      LM1=JMA-1                                                         00013070
      IF(JMA-LSTMP)1040,1070,1040                                       00013080
                                                                        00013090
                              RESER IFUNC VALUE FOR RESIDUALS.          00013100
                                                                        00013110
 1040 DO 1060 JMA2=1,JSTOP                                              00013120
      IF(IFUNC(JMA,JMA2))1060,1050,1050                                 00013130
 1050 IFUNC(JMA,JMA2)=IFUNC(LM1,JMA2)                                   00013140
 1060 CONTINUE                                                          00013150
      GO TO 1300                                                        00013160
```

```
                                                                      00013170
                          RESET FUNC VALUES FOR SUBFUNCTIONS.         00013180
                                                                      00013190
1070 CONTINUE                                                         00013200
                                                                      00013210
                          INITIALIZE IFUNC TO LEVEL BELOW.            00013220
                                                                      00013230
     DO 1080 JMA2=1,JSTOP                                             00013240
     IFUNC(JMA,JMA2)=IFUNC(LM1,JMA2)                                  00013250
1080 CONTINUE                                                         00013260
                                                                      00013270
                          REPLACE 1'S WITH R'S.                       00013280
     DO 1200 K=ISTRT,NEPIS                                            00013290
     JMA2=IEPIS(K)                                                    00013300
     IF(IS(JMA2,3)-JMA)1200,1100,1090                                 00013310
1090 ITMP=K                                                           00013320
     GO TO 1220                                                       00013330
1100 CONTINUE                                                         00013340
     LOW=IS(JMA2,2)                                                   00013350
                                                                      00013360
     DO 1140 JMA3=1,NOTRM                                             00013370
     IF(MTRMS(JMA3)-LOW)1140,1110,1110                                00013380
1110 JTMP=MTRMS(JMA3)+1                                               00013390
     IF(IFUNC(JMA,JTMP))1140,1120,1120                                00013400
1120 ITEST=IAND(LOW,MTRMS(JMA3))                                      00013410
     IF(ITEST-LOW)1140,1130,1140                                      00013420
1130 IFUNC(JMA,JTMP)=3                                                00013430
1140 CONTINUE                                                         00013440
1200 CONTINUE                                                         00013450
1220 CONTINUE                                                         00013460
     ISTRT=ITMP                                                       00013470
     LSTMP=LSTMP+2                                                    00013480
1300 CONTINUE                                                         00013490
     RETURN                                                           00013500
     END                                                              00013510
                                                                      00013520
                                                                      00013530
/ / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / 00013540
                                                                      00013550
                                                                      00013560
                                                                      00013570
     SUBROUTINE REDPI(NEPIS,JSTOR,ICOV,NOTRM,MTRMS,MCOV,NOSF,IEPIS,NP1)00013580
     DIMENSION ICOV(20,256),IEPIS(1),MTRMS(1)                         00013590
     COMMON IN,IOUT,IPRIM                                             00013600
     DATA IPLUS,MINUS/'+','-'/                                        00013610
     LAST=MTRMS(NOTRM)+2                                              00013620
     IF(JSTOR)630,630,600                                             00013630
 600 CONTINUE                                                         00013640
                                                                      00013650
                          PUT SUBFUNCTION IN ROW NP1 OF ICOV ALSO.    00013660
                          THAT IS, IT IS ESSENTIAL AND IP1 ROW IS     00013670
                          FOR COVERING TESTS.                         00013680
                                                                      00013690
     NEPIS=NEPIS+1                                                    00013700
     IEPIS(NEPIS)=JSTOR                                               00013710
     ICOV(JSTOR,LAST)=MINUS                                           00013720
                                                                      00013730
     DO 620 JMA=1,NOTRM                                               00013740
     JTMP=MTRMS(JMA)+1                                                00013750
     IF(ICOV(JSTOR,JTMP)-IPLUS)620,610,620                            00013760
 610 CONTINUE                                                         00013770
     ICOV(NP1,JTMP)=IPLUS                                             00013780
 620 CONTINUE                                                         00013790
                                                                      00013800
                          CHECK FOR MINTERM COVERAGE.                 00013810
                                                                      00013820
```

```
      DO 625 JMA=1,NOTRM                                                00013830
      IF(ICOV(NP1,MTRMS(JMA)+1)-IPLUS)630,625,630                       00013840
  625 CONTINUE                                                          00013850
      MCOV=1                                                            00013860
      GO TO 700                                                         00013870
  630 CONTINUE                                                          00013880
                                                                        00013890
                                                                        00013900
                          FIND ESSENTIAL PI'S.                          00013910
                                                                        00013920
      DO 680 JMA=1,NOTRM                                                00013930
      JTMP=MTRMS(JMA)+1                                                 00013940
      IF(ICOV(NP1,JTMP)-IPLUS)640,680,640                               00013950
  640 CONTINUE                                                          00013960
      I=0                                                               00013970
      DO 660 JMA2=1,NOSF                                                00013980
      IF(ICOV(JMA2,JTMP)-IPLUS)660,650,660                              00013990
  650 CONTINUE                                                          00014000
      JSTOR=JMA2                                                        00014010
      I=I+1                                                             00014020
  660 CONTINUE                                                          00014030
      IF(I-1)680,600,680                                                00014040
  680 CONTINUE                                                          00014050
                                                                        00014060
                                                                        00014070
                          ELIMINATE COVERED SUBFUNCTIONS.               00014080
                                                                        00014090
  700 CONTINUE                                                          00014100
      DO 810 JMA=1,NOSF                                                 00014110
      IF(ICOV(JMA,LAST)-MINUS)780,810,780                               00014120
  780 CONTINUE                                                          00014130
      DO 800 JMA2=1,NOTRM                                               00014140
      JTMP=MTRMS(JMA2)+1                                                00014150
      IF(ICOV(NP1,JTMP)-IPLUS)790,800,790                               00014160
  790 CONTINUE                                                          00014170
      IF(ICOV(JMA,JTMP)-IPLUS)800,810,800                               00014180
  800 CONTINUE                                                          00014190
      ICOV(JMA,LAST)=MINUS                                              00014200
  810 CONTINUE                                                          00014210
                                                                        00014220
                          PRINT REDUCED PI TABLE.                       00014230
                                                                        00014240
      WRITE(IOUT,820)                                                   00014250
  820 FORMAT('OREDUCED TABLE.')                                         00014260
                                                                        00014270
      WRITE(IOUT,830)(MTRMS(JMA),JMA=1,NOTRM)                           00014280
  830 FORMAT(1HO,4X,32I3)                                               00014290
      DO 900 JP=1,NP1                                                   00014300
      IF(ICOV(JP,LAST)-MINUS)850,900,850                                00014310
  850 CONTINUE                                                          00014320
      WRITE(IOUT,860)JP,(ICOV(JP,MTRMS(JMA)+1),JMA=1,NOTRM)             00014330
  860 FORMAT(I6,32(1X,A1,1X))                                           00014340
  900 CONTINUE                                                          00014350
      IF(MCOV)905,905,1000                                              00014360
  905 CONTINUE                                                          00014370
                                                                        00014380
                          CHECK FOR MINTERM COVERAGE.                   00014390
                                                                        00014400
      DO 910 JMA=1,NOTRM                                                00014410
      IF(ICOV(NP1,MTRMS(JMA)+1)-IPLUS)2000,910,2000                     00014420
  910 CONTINUE                                                          00014430
      MCOV=1                                                            00014440
                                                                        00014450
 1000 CONTINUE                                                          00014460
 2000 CONTINUE                                                          00014470
      RETURN                                                            00014480
```

```
      END                                                               00014490
                                                                        00014500
                                                                        00014510
/ / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / 00014520
                                                                        00014530
                                                                        00014540
      MINIMIZE 3RD LEVEL INPUTS.                                        00014550
      SUBROUTINE M3RDI(N2ND,I2ND,N3RD,I3RD,IS,IT,IFEED,ICUT)           00014560
                                                                        00014570
      I2ND-LIST OF 2ND LEVEL GATES.                                     00014580
      I3RD-LIST OF 3RD LEVEL GATES.                                     00014590
      N2ND-NUMBER OF 2ND LEVEL GATES.                                   00014600
      N3RD-NUMBER OF 3RD LEVEL GATES.                                   00014610
      IFEED-FEEDING 3RD LEVEL GATES TO A PARTICULAR 2ND LEVEL GATE.     00014620
      IS- CELL MINIMUMS FOR 2ND LEVEL INPUTS.                           00014630
      IT- CELL MINIMUMS FOR 3RD LEVEL INPUTS.                           00014640
      IELIM-LIST OF INPUTS TO BE ELIMINATED.                            00014650
      JMA- LOOP INDEX.                                                  00014660
      JMA2-INTERIOR LOOP INDEX.                                         00014670
      ICUT-OUTPUT DEVICE LOGICAL UNIT NUMBER                            00014680
      DIMENSION IT(30,1),IS(30,1)                                       00014690
      DIMENSION IFEED(30,1)                                             00014700
      DIMENSION IELIM(30)                                               00014710
      DIMENSION I2ND(1),I3RD(1)                                         00014720
                                                                        00014730
      IF(N3RD)1000,1000,10                                              00014740
   10 IF(N2ND)1000,1000,20                                              00014750
   20 CONTINUE                                                          00014760
                                                                        00014770
                              INITIALIZE POTENTIAL INPUT ELIMINATIONS.  00014780
                                                                        00014790
      DO 40 JMA=1,N3RD                                                  00014800
      IG=I3RD(JMA)                                                      00014810
      IELIM(IG)=IT(IG,1)                                                00014820
      WRITE(IOUT,30)JMA,IG,IELIM(IG)                                    00014830
   30 FORMAT('0JMA=',I4,' IG=',I4,'  IELIM(IG)=',I5)                    00014840
   40 CONTINUE                                                          00014850
                                                                        00014860
                              AND EACH 3RD LEVEL POTENTIAL INPUT ELIMINAT00014870
                              WITH THE 2ND LEVEL GATES IT FEEDS TO      00014880
                              FIND INPUTS TO BE ELIMINATED.             00014890
                                                                        00014900
      DO 80 JMA=1,N2ND                                                  00014910
      IF(IFEED(JMA,1))80,80,50                                          00014920
   50 ITEMP=IFEED(JMA,1)+1                                              00014930
      MIN=IS(I2ND(JMA),1)                                               00014940
      DO 60 JMA2=2,ITEMP                                                00014950
      IG=IFEED(JMA,JMA2)                                                00014960
      IELIM(IG)=IAND(MIN,IELIM(IG))                                     00014970
      WRITE(IOUT,55)JMA,JMA2,IG,IELIM(IG),MIN                           00014980
   55 FORMAT(' JMA=',I4,' JMA2=',I4,' IG=', I4,' IELIM=',I4,' MIN=',I4) 00014990
   60 CONTINUE                                                          00015000
   80 CONTINUE                                                          00015010
                                                                        00015020
                              ELIMINATE THE INPUTS BY EXCLUSIVE ORING.  00015030
                                                                        00015040
      DO 100 JMA=1,N3RD                                                 00015050
      IG=I3RD(JMA)                                                      00015060
      IT(IG,1)=IXOR(IT(IG,1),IELIM(IG))                                 00015070
      WRITE(IOUT,30)JMA,IG,IELIM(IG)                                    00015080
  100 CONTINUE                                                          00015090
 1000 CONTINUE                                                          00015100
                                                                        00015110
      RETURN                                                            00015120
      END                                                               00015130
                                                                        00015140
```

```
                                                                      00015150
/ / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / 00015160
                                                                      00015170
                                                                      00015180
                                                                      00015190
   ELIMINATE 2ND LEVEL INPUTS.                                        00015200
    SUBROUTINE EL2IN(N2ND,I2ND,IS,IT,IFEED,JSTOP,IORDR)               00015210
                                                                      00015220
    VARAIBLES:                                                        00015230
    N2ND-NUMBER OF SECCND LEVEL GATES.                                00015240
    I2ND-LIST OF SECOND LEVEL GATES.                                  00015250
    IS -CELL MINIMUMS FOR SECOND LEVEL GATES.                         00015260
    IT -CELL MINIMUMS FOR THIRD LEVEL GATES.                          00015270
    IFEED-LIST OF 3ND LEVEL GATES FEEDING 2ND LEVEL GATES.            00015280
    J2ND CURRENT 2ND LEVEL GATE BEING CONSIDERED.                     00015290
    ICLT-OLTPUT LCGICAL UNIT NUMBER.                                  00015300
    MIN-CELL MINIMUM FOR CURRENT 2ND LEVEL GATE BEING CONSIDERED.     00015310
    MAX-PRIMARY CELL(MAX FOR ANY CELL).                               00015320
    MAXC-MAXIMUM FOR CELL=MAX IF NO FEEDING 3RD LEVEL;=COMPLIMENT     00015330
          OF 3RD CELL MINIMUM IF ONLY ONE FEEDING 3RD LEVEL GATE.     00015340
    MINF3-MINIMUM FOR FEEDING 3RD LEVEL CELL.                         00015350
    BITNC-BIT NUMBER IN CELL BEING PROCESSED.                         00015360
    MINT-MINIMUM TEMPCRARY(FOR TESTING FOR INPUT REMOVAL).            00015370
    IRMCV-INPUT TO BE REMOVED(MAYBE?).                                00015380
    JBEG-LOWER BOUND FOR INDEXING MINTERMS IN CELL.                   00015390
    JEND-UPPER BOUND FOR INDEXING MINTERMS IN THIS CELL.              00015400
    JTERM-TERM BEING CONSIDERED.                                      00015410
                                                                      00015420
    DIMENSION I2ND(1),IFEED(30,1)                                     00015430
    DIMENSION IFUNC(40,256)                                           00015440
    DIMENSION MASK3(512)                                              00015450
    DIMENSION IBIN(10)                                                00015460
    DIMENSION IS(30,1),IT(30,1)                                       00015470
    DIMENSION ICONV(30)                                               00015480
                                                                      00015490
    COMMCN IN,IOUT,MAX,ISET,IFUNC,LEVEL,ICONV,MASK3,MASK2             00015500
    DATA JONE/'1'/                                                    00015510
                                                                      00015520
                                                                      00015530
    DO 1000 JMA=1,N2ND                                                00015540
    J2NC=I2ND(JMA)                                                    00015550
    MIN=IS(J2ND,1)                                                    00015560
                           CAN'T ELIMINATE INPUTS IF THERE AREN'T ANY.00015570
                                                                      00015580
    IF(MIN)1000,1000,100                                              00015590
100 CCNTINUE                                                          00015600
    NFEEC=IFEED(JMA,1)                                                00015610
                           TEST NUMBER OF FEEDING 3RD LEVEL GATES;    00015620
                           DIFFERENT CASES FOR ONE FEED, ZERO FEEDS,  00015630
                           AND TWO CR MORE FEEDS.                     00015640
                                                                      00015650
    IF(NFEED-1)120,300,300                                            00015660
                                                                      00015670
                                                                      00015680
                           ZERO FEEDERS SET MAXIMUM FOR CELL(MAXC) TO 00015690
                           PRIMARY CUBE.                              00015700
                                                                      00015710
120 CCNTINUE                                                          00015720
    MAXC=MAX                                                          00015730
130 CCNTINUE                                                          00015740
                                                                      00015750
                           FIND FUNCTION CELL.                        00015760
                                                                      00015770
    CALL CELL(MIN,MAXC,IBIN,NSIZE,IORDR)                              00015780
    WRITE(IOUT,140)MIN,MAXC,(IBIN(J),J=1,IORDR)                       00015790
140 FCRMAT('0MIN=',I4,' MAXC=',I4,'  CELL=',10A1)                     00015800
```

```
                                                                      00015810
                          ATTEMPT TO REMOVE 2ND LEVEL INPUTS.         00015820
                                                                      00015830
                                                                      00015840
      JENC=MAXC+1                                                     00015850
                                                                      00015860
      DO 250 JMA2=1,IORDR                                             00015870
      IF(IBIN(JMA2)-JONE)250,150,250                                  00015880
  150 CONTINUE                                                        00015890
      IRMOV=2**(IORDR-JMA2)                                           00015900
      MINT=MIN-IRMOV                                                  00015910
      IF(MINT)250,155,155                                             00015920
  155 CONTINUE                                                        00015930
      JBEG=MINT+1                                                     00015940
                                                                      00015950
                          TEST IF ALL THE ENTRIES IN THE NEW CELL     00015960
                          ARE MINTERMS OR DON'T CARES.                00015970
                                                                      00015980
      DO 200 JMA3=JBEG,JEND                                           00015990
                                                                      00016000
                          TEST IF THIS TERMS VALUE IS A  MINTERM OR D00016010
                                                                      00016020
      IF(IFUNC(1,JMA3))160,160,200                                    00016030
  160 CONTINUE                                                        00016040
                                                                      00016050
                          TEST IF THIS TERM IS THE NEW CELL.          00016060
                                                                      00016070
      JTERM=JMA3-1                                                    00016080
      JTEST=IAND(JTERM,MINT)                                          00016090
      IF(JTEST-MINT)200,170,200                                       00016100
  170 CONTINUE                                                        00016110
      JTEST=IAND(JTERM,MAXC)                                          00016120
      IF(JTEST-JTERM)200,180,200                                      00016130
  180 CONTINUE                                                        00016140
                                                                      00016150
                          THIS TERM IS IN THE CELL BUT IS NOT A VALUE00016160
                          OF ONE OR A D.C. IN THE ORIGINAL FUNCTION. 00016170
                          DO NOT ELIMINATE IRMOV, JUMP TO NEXT TRY.  00016180
                                                                      00016190
      GO TO 250                                                       00016200
  200 CONTINUE                                                        00016210
                                                                      00016220
                          HAVING CHECKED ALL VALUES IN THE NEW CELL   00016230
                          AND FOUND THAT THEY ARE 1'S OR DC'S, ELIMIN00016240
                          INPUT IRMOV BY SETTING MIN TO MINT.         00016250
                                                                      00016260
      MIN=MINT                                                        00016270
  250 CONTINUE                                                        00016280
                                                                      00016290
                          MAKE NEW CELL MIN PERMANENT.                00016300
                                                                      00016310
      IS(J2ND,1)=MIN                                                  00016320
      GO TO 1000                                                      00016330
                                                                      00016340
  300 CONTINUE                                                        00016350
                                                                      00016360
                          MORE THAN ONE FEEDING 3RD LEVEL GATE SPECIA00016370
                                                                      00016380
                                                                      00016390
                          INITIALIZE INHIBITING 3RD LEVEL MASK.       00016400
                                                                      00016410
                          SET ALL VALUES TO ONES.                     00016420
                                                                      00016430
      DO 350 JMA2=1,JSTOP                                             00016440
      MASK3(JMA2)=1                                                   00016450
  350 CONTINUE                                                        00016460
```

```
                                                                      00016470
                          PLACE ZEROES IN TERMS COVERED BY 3RD LEVEL  00016480
                                                                      00016490
      NFEED=NFEED+1                                                   00016500
      DO 400 JMA2=2,NFEED                                             00016510
      MINF3=IT(IFEED(JMA,JMA2),1)                                     00016520
      JBEG=MINF3+1                                                    00016530
      DO 380 JMA3=JBEG,JSTOP                                          00016540
      JTEST=JMA3-1                                                    00016550
      JTEST=IAND(JTEST,MINF3)                                         00016560
      IF(JTEST-MINF3)380,360,380                                      00016570
  360 MASK3(JMA3)=0                                                   00016580
  380 CONTINUE                                                        00016590
                                                                      00016600
  400 CONTINUE                                                        00016610
                                                                      00016620
                                                                      00016630
                          FIND SECOND LEVEL CELL.                     00016640
                                                                      00016650
      CALL CELL(MIN,MAX,IBIN,NSIZE,IORDR)                             00016660
                                                                      00016670
                          BEGIN REMOVING INPUTS.                      00016680
                                                                      00016690
      DO 600 JMA2=1,IORDR                                             00016700
      IF(IBIN(JMA2)-JONE)600,410,600                                  00016710
  410 CONTINUE                                                        00016720
      IRMOV=2**(IORDR-JMA2)                                           00016730
      MINT=MIN-IRMOV                                                  00016740
      JBEG=MINT+1                                                     00016750
                                                                      00016760
                          TEST IF ALL NEW ENTRIES ARE MINTERMS OR D.C00016770
                                                                      00016780
      DO 500 JMA3=JBEG,JSTOP                                          00016790
      IF(IFUNC(1,JMA3))450,450,500                                    00016800
                                                                      00016810
                          TEST IF THIRD LEVEL MASK INHIBITS THIS TERM00016820
                                                                      00016830
  450 IF(MASK3(JMA3))500,500,460                                      00016840
  460 CONTINUE                                                        00016850
                                                                      00016860
                          TEST IF THIS TERM IS IN THE CELL.           00016870
                                                                      00016880
      JTEST=JMA3-1                                                    00016890
      JTEST=IAND(JTEST,MINT)                                          00016900
      IF(JTEST-MINT)500,600,500                                       00016910
  500 CONTINUE                                                        00016920
                                                                      00016930
                          HAVING CHECKED ALL NEW TERMS AND FOUND THEM00016940
                          MINTERMS OR DC'S, ELIMINATE THIS INPUT.     00016950
                                                                      00016960
      MIN=MINT                                                        00016970
  600 CONTINUE                                                        00016980
                                                                      00016990
                          MAKE ELIMINATIONS PERMANENT.                00017000
                                                                      00017010
      IS(J2ND,1)=MIN                                                  00017020
                                                                      00017030
 1000 CONTINUE                                                        00017040
      RETURN                                                          00017050
      END                                                             00017060
                                                                      00017070
                                                                      00017080
 / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / 00017090
                                                                      00017100
                                                                      00017110
                                                                      00017120
```

```
      SUBROUTINE EL3IN(N3RD,I3RD,IT,N2ND,I2ND,IS,IFEED,NOTRM,MTRMS,IORDRO0017130
     $,JSTOP,IFST)                                                      00017140
                                                                        00017150
                                                                        00017160
                                                                        00017170
      J3RD-THIRD LEVEL GATE TO BE PUT INTO MASK3.                       00017180
      J2ND-SECOND LEVEL GATE TO BE PUT ONTO MASK2.                      00017190
      N2ND-NUMBER OF SECOND LEVEL GATES.                                00017200
      I2ND-LIST OF SECOND LEVEL GATES.                                  00017210
      IS-LIST OF SECOND LEVEL GATE CELL MINIMUMS.                       00017220
      N3RD-NUMBER OF 3RD LEVEL GATES.                                   00017230
      I3RD-LIST OF THIRD LEVEL GATES.                                   00017240
      IT-LIST OF THIRD LEVEL GATE INPUT CELL MINIMUMS.                  00017250
      IFEED-LIST OF FEEDING 3RD LEVELS TO 2ND LEVEL.                    00017260
      MASK2-MASK OF VALUES FOR SECOND LEVEL GATES.                      00017270
      MASK3-MASK OV VALUES FOR 3RD LEVEL GATES.                         00017280
                                                                        00017290
                                                                        00017300
      DIMENSION IFUNC(40,256)                                           00017310
      DIMENSION LFST(256)                                               00017320
      DIMENSION MTRMS(1)                                                00017330
      DIMENSION IT(30,1),IS(30,1),IFEED(30,1)                           00017340
      DIMENSION I3RD(1),I2ND(1)                                         00017350
      DIMENSION IBIN(10)                                                00017360
      DIMENSION MASK3(32,256)                                           00017370
      DIMENSION MASK2(32,256)                                           00017380
      DIMENSION ICONV(30)                                               00017390
      COMMON IN,IOUT,MAX,JSET,IFUNC,LEVEL,ICONV,MASK3,MASK2             00017400
      DATA JONE/'1'/                                                    00017410
                                                                        00017420
                                                                        00017430
                              INITIALIZE CONVERSION INDEX.  THIS CONVERTS00017440
                              THE VALUE OF IFEED,WHICH IS AN INDEX FOR IT00017450
                              TO AN INDEX FOR MASK3.                    00017460
                                                                        00017470
      DO 20 JMA=1,N3RD                                                  00017480
      JTMP=I3RD(JMA)                                                    00017490
      ICONV(JTMP)=JMA                                                   00017500
   20 CONTINUE                                                          00017510
                                                                        00017520
      JFILL=0                                                           00017530
      IF(IFST)50,40,40                                                  00017540
   40 JFILL=1                                                           00017550
   50 CONTINUE                                                          00017560
                                                                        00017570
                              CLEAR THE MASKS.                          00017580
                                                                        00017590
      DO 100 JMA=1,JSTOP                                                00017600
      DO 80 JMA2=1,N3RD                                                 00017610
      MASK3(JMA2,JMA)=1                                                 00017620
   80 CONTINUE                                                          00017630
                                                                        00017640
      DO 90 JMA2=1,N2ND                                                 00017650
      MASK2(JMA2,JMA)=0                                                 00017660
   90 CONTINUE                                                          00017670
                                                                        00017680
                              INITIALIZE FIRST LEVEL VECTOR.            00017690
                                                                        00017700
      LFST(JMA)=JFILL                                                   00017710
  100 CONTINUE                                                          00017720
                                                                        00017730
      WRITE(IOUT,110)                                                   00017740
  110 FORMAT('0MASK3',120X,'EL3IN')                                     00017750
                                                                        00017760
                              FILL MASK3 WITH ACTUAL VALUES.            00017770
                                                                        00017780
```

```
      DO 250 JMA=1,N3RD                                                 00017790
      J3RD=I3RD(JMA)                                                    00017800
      MIN=IT(J3RD,1)                                                    00017810
      JBEG=MIN+1                                                        00017820
      DO 200 JMA2=JBEG,JSTOP                                            00017830
      JTMP=JMA2-1                                                       00017840
      JTST=IAND(JTMP,MIN)                                               00017850
      IF(JTST-MIN)200,190,200                                           00017860
  190 CONTINUE                                                          00017870
      MASK3(JMA,JMA2)=0                                                 00017880
  200 CONTINUE                                                          00017890
                                                                        00017900
      WRITE(IOUT,210)JMA,(MASK3(JMA,JMA2),JMA2=1,JSTOP)                 00017910
  210 FORMAT(1X,I5,',',16I6)                                            00017920
  250 CONTINUE                                                          00017930
                                                                        00017940
                            FILL FIRST LEVEL VECTOR IF INPUTS ARE       00017950
                            AT THE FIRST LEVEL GATE.                    00017960
                                                                        00017970
      IF(IFST)280,260,260                                               00017980
  260 MIN=IFST                                                          00017990
      JBEG=MIN+1                                                        00018000
      DO 270 JMA=JBEG,JSTOP                                             00018010
      JTST=JMA-1                                                        00018020
      JTST=IAND(JTST,MIN)                                               00018030
      IF(JTST-MIN)270,265,270                                           00018040
  265 LFST(JMA)=0                                                       00018050
  270 CONTINUE                                                          00018060
                                                                        00018070
  280 CONTINUE                                                          00018080
                                                                        00018090
      WRITE(IOUT,285)(LFST(JMA),JMA=1,JSTOP)                            00018100
  285 FORMAT('0LFST  ',16I6)                                            00018110
                                                                        00018120
                            FILL MASK2.                                 00018130
                                                                        00018140
      DO 340 JMA=1,N2ND                                                 00018150
      J2ND=I2ND(JMA)                                                    00018160
      MIN=IS(J2ND,1)                                                    00018170
      JBEG=MIN+1                                                        00018180
      DO 300 JMA2=JBEG,JSTOP                                            00018190
      JTMP=JMA2-1                                                       00018200
      JTST=IAND(JTMP,MIN)                                               00018210
      IF(JTST-MIN)300,290,300                                           00018220
  290 MASK2(JMA,JMA2)=1                                                 00018230
  300 CONTINUE                                                          00018240
                                                                        00018250
      WRITE(IOUT,210)JMA,(MASK2(JMA,JMA2),JMA2=1,JSTOP)                 00018260
  340 CONTINUE                                                          00018270
                                                                        00018280
                            ELIMINATE INPUTS TO 3RD LEVEL GATE BY GATE.00018290
                                                                        00018300
      DO 700 JMA=1,N3RD                                                 00018310
      J3RD=I3RD(JMA)                                                    00018320
      MIN=IT(J3RD,1)                                                    00018330
      CALL CELL(MIN,MAX,IBIN,NSIZE,IORDR)                               00018340
                                                                        00018350
      WRITE(IOUT,345)(IBIN(JMA2),JMA2=1,IORDR)                          00018360
  345 FORMAT(1H0,10A1)                                                  00018370
                                                                        00018380
      IF(IORDR-NSIZE-1)700,700,350                                      00018390
  350 CONTINUE                                                          00018400
                                                                        00018410
                            ELIMINATE INPUTS ONE AT A TIME.             00018420
                                                                        00018430
      DO 600 JMA2=1,IORDR                                               00018440
```

```
    IF(IBIN(JMA2)-JONE)600,360,600                                  00018450
360 CONTINUE                                                        00018460
    IRMOV=2**(IORDR-JMA2)                                           00018470
    MINT=MIN-IRMOV                                                  00018480
    JBEG=MINT+1                                                     00018490
                                                                    00018500
                          REVALUE MASK3.                            00018510
                                                                    00018520
    DO 380 JMA3=JBEG,JSTOP                                          00018530
    JTST=JMA3-1                                                     00018540
    JTST=IAND(JTST,MINT)                                            00018550
    IF(JTST-MINT)380,370,380                                        00018560
370 CONTINUE                                                        00018570
    MASK3(JMA,JMA3)=0                                               00018580
380 CONTINUE                                                        00018590
                                                                    00018600
    WRITE(IOUT,381)(MASK3(JMA,JMA3),JMA3=1,JSTOP)                   00018610
381 FORMAT('OMASK3 ',16I6)                                          00018620
                                                                    00018630
                          TEST FOR MINTERM COVERAGE.                00018640
                                                                    00018650
    DO 500 JMA3=1,NOTRM                                             00018660
    MTERM=MTRMS(JMA3)+1                                             00018670
                                                                    00018680
    IF(LFST(MTERM))390,390,500                                      00018690
390 CONTINUE                                                        00018700
                                                                    00018710
                          TEST EACH SECOND LEVEL GATE FOR A MASK2   00018720
                          VALUE OF 1.                               00018730
                                                                    00018740
    DO 480 JMA4=1,N2ND                                              00018750
    IF(MASK2(JMA4,MTERM))480,480,400                                00018760
400 CONTINUE                                                        00018770
    NFEED=IFEED(JMA4,1)                                             00018780
    IF(NFEED)500,500,410                                            00018790
410 NFEED=NFEED+1                                                   00018800
                                                                    00018810
                          TEST EACH THIRD LEVEL GATE MASK3 FOR A    00018820
                                                                    00018830
                          VALUE OF 1.                               00018840
                                                                    00018850
    DO 470 JMA5=2,NFEED                                             00018860
    JTMP=MASK3(ICONV(IFEED(JMA4,JMA5)),MTERM)                       00018870
    IF(JTMP)480,480,470                                             00018880
470 CONTINUE                                                        00018890
                                                                    00018900
                          THIS GATE COVERS A ONE, GO TO NEXT        00018910
                          MINTERM FOR TESTING.                      00018920
                                                                    00018930
    GO TO 500                                                       00018940
480 CONTINUE                                                        00018950
                                                                    00018960
                                                                    00018970
                          THIS MINTERM WAS NOT COVERED, DO NOT      00018980
                          ELIMINATE THIS INPUT.  REINITIALIZE THE MAS00018990
                                                                    00019000
    GO TO 510                                                       00019010
                                                                    00019020
500 CONTINUE                                                        00019030
                                                                    00019040
                          MINTERMS ARE COVERED MAKE THIS NEW MIN.   00019050
                                                                    00019060
    MIN=MINT                                                        00019070
    GO TO 600                                                       00019080
                                                                    00019090
                          SOME MINTERM NOT COVERED,REINITIALIZE MASK300019100
```

```
                                                                      00019110
510 CONTINUE                                                          00019120
                                                                      00019130
    DO 550 JMA3=JBEG,MAX                                              00019140
    JTST=JMA3-1                                                       00019150
    JTST=IAND(JTST,MIN)                                               00019160
    IF(JTST-MIN)540,550,540                                           00019170
540 MASK3(JMA,JMA3)=1                                                 00019180
550 CONTINUE                                                          00019190
                                                                      00019200
600 CONTINUE                                                          00019210
                                                                      00019220
                          MAKE INPUT REDUCTIONS PERMANENT.            00019230
                                                                      00019240
    IT(J3RD,1)=MIN                                                    00019250
700 CONTINUE                                                          00019260
                                                                      00019270
    RETURN                                                            00019280
    END                                                               00019290
                                                                      00019300
                                                                      00019310
/ / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / 00019320
                                                                      00019330
                                                                      00019340
                                                                      00019350
   SUB.   DUPGT.                                                      00019360
    SUBROUTINE DUPGT(N3RD,I3RD,N2ND,IT,IOUT,IFEED)                    00019370
                                                                      00019380
    SUBROUTINE TO ELIMINATE DUPLICATE GATES.                          00019390
                                                                      00019400
    DIMENSION IFEED(30,1)                                             00019410
    DIMENSION IT(30,1)                                                00019420
    DIMENSION I3RD(1)                                                 00019430
    DIMENSION IDUP(30)                                                00019440
                                                                      00019450
                          INITIALIZE IDUP AND THE FLAG.               00019460
                                                                      00019470
    IFLAG=0                                                           00019480
    DO 10 JMA=1,30                                                    00019490
    IDUP(JMA)=JMA                                                     00019500
 10 CONTINUE                                                          00019510
                                                                      00019520
    JMA=1                                                             00019530
                                                                      00019540
                          LOOP THROUGH AND FIND DUPLICATE GATES.      00019550
                                                                      00019560
 15 IF(JMA-N3RD)20,400,400                                            00019570
 20 JMA2=JMA+1                                                        00019580
    I3JMA=I3RD(JMA)                                                   00019590
                                                                      00019600
 25 IF(JMA2-N3RD)30,30,300                                            00019610
 30 IF(IT(I3JMA,1)-IT(I3RD(JMA2),1))200,50,200                        00019620
                                                                      00019630
                          REMOVE THE JMA2ND GATE MARK ITS REPLACEMENT00019640
                          AND SET THE FLAG.                           00019650
                                                                      00019660
 50 IDUP(I3RD(JMA2))=I3JMA                                            00019670
    WRITE(IOUT,55)I3JMA,JMA,JMA2,N3RD                                 00019680
 55 FORMAT('0I3JMA= ',I5,' JMA=',I5,' JMA2=',I5,' N3RD=',I5)          00019690
    IFLAG=1                                                           00019700
    N3RD=N3RD-1                                                       00019710
    IF(JMA2-N3RD)60,500,500                                           00019720
 60 CONTINUE                                                          00019730
                                                                      00019740
                          MOVE REMAINING GATES UP IN THE LIST.        00019750
                                                                      00019760
```

```
      DO 100 JMA3=JMA2,N3RD                                             00019770
      I3RD(JMA3)=I3RD(JMA3+1)                                           00019780
  100 CONTINUE                                                          00019790
  200 JMA2=JMA2+1                                                       00019800
      GO TO 25                                                          00019810
  300 JMA=JMA+1                                                         00019820
      GO TO 15                                                          00019830
                                                                        00019840
                            FINISHED SEARCHING FOR DUPLICATES.          00019850
                            TEST FLAG TO SEE IF ANY DUPLICATES WERE FOU 00019860
                                                                        00019870
  400 IF(IFLAG)500,700,500                                              00019880
                                                                        00019890
                            CORRECT IFEED FOR REMOVED GATES.            00019900
                                                                        00019910
  500 CONTINUE                                                          00019920
                                                                        00019930
      DO 600 JMA=1,N2ND                                                 00019940
      NFEED=IFEED(JMA,1)                                                00019950
      IF(NFEED)600,600,510                                              00019960
  510 NFEED=NFEED+1                                                     00019970
      WRITE(IOUT,520)JMA,(IFEED(JMA,JMA2),JMA2=2,NFEED)                 00019980
  520 FORMAT('0JMA=',I5,' IFEED=',(T21,16I5/))                          00019990
      DO 580 JMA2=2,NFEED                                               00020000
      IFEED(JMA,JMA2)=IDUP(IFEED(JMA,JMA2))                             00020010
  580 CONTINUE                                                          00020020
                                                                        00020030
      WRITE(IOUT,520)JMA,(IFEED(JMA,JMA2),JMA2=2,NFEED)                 00020040
  600 CONTINUE                                                          00020050
                                                                        00020060
  700 CONTINUE                                                          00020070
      RETURN                                                            00020080
      END                                                               00020090
                                                                        00020100
                                                                        00020110
/ / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / 00020120
                                                                        00020130
                                                                        00020140
                                                                        00020150
      SUBROUTINE ALTCKT(I2ND,IS,IT,I3RD,IFEED,IFLAG,                    00020160
     1JSTOP,N3RD,N2ND,NEPIS,IORDR)                                      00020170
                                                                        00020180
                     THIS ROUTINE ATTEMPTS TO MODIFY A CIRCUIT          00020190
                      BY REPLACING FEEDING THIRD LEVEL GATES.           00020200
                                                                        00020210
      NOV 24, 1977                                                      00020220
      VARIABLES:                                                        00020230
                  JMA,JMA2,JMA3,JMA4-LOOP INDICES.                      00020240
                  IFLAG-FLAG WHICH INDICATES ALTERED CIRCUIT.           00020250
                       =0 MEANS NO ALTERATIONS.                         00020260
                       =+ MEANS SOME ALTERATIONS.                       00020270
                  NRMOV-NUMBER OF REMOVED 3RD LEVEL FEEDING GATES.      00020280
                  MIN2 -MINIMUM OF CELL FOR SECOND LEVEL GATE.          00020290
                  MFEED-MINIMUM FOR CELL OF 3RD LEVEL FEEDING GATE.     00020300
                  IGATE-FEEDING 3RD LEVEL GATE.                         00020310
                  IGTMP-TEMP 3RD LEVEL GATE TO ALTER CIRCUIT.           00020320
                                                                        00020330
                                                                        00020340
                                                                        00020350
      DIMENSION I2ND(1),I3RD(1),MASK2(32,256)                           00020360
      DIMENSION MASK3(32,256),IS(30,1),IT(30,1)                         00020370
      DIMENSION IFEED(30,1),IFUNC(40,256)                               00020380
                                                                        00020390
      COMMON IN,IOUT,IPRIM,ISET,IFUNC,LEVEL,                            00020400
     1  ICONV(30),MASK3,MASK2                                           00020410
                                                                        00020420
```

```
      IFLAG=0                                                          00020430
      NRMOV=0                                                          00020440
                                                                       00020450
       I N I T I A L I Z E   M A S K S .                               00020460
                                                                       00020470
                               CLEAR MASKS.                            00020480
                                                                       00020490
      DO 100 JMA=1,JSTOP                                               00020500
      DO 80 JMA2=1,N3RD                                                00020510
      MASK3(JMA2,JMA)=1                                                00020520
   80 CONTINUE                                                         00020530
                                                                       00020540
      DO 90 JMA2=1,N2ND                                                00020550
      MASK2(JMA2,JMA)=0                                                00020560
   90 CONTINUE                                                         00020570
                                                                       00020580
  100 CONTINUE                                                         00020590
                                                                       00020600
                               FILL MASK3 WITH ACTUAL VALUES.          00020610
                                                                       00020620
      DO 250 JMA=1,N3RD                                                00020630
      J3RD=I3RD(JMA)                                                   00020640
      MIN=IT(J3RD,1)                                                   00020650
      JBEG=MIN+1                                                       00020660
                                                                       00020670
      DO 200 JMA2=JBEG,JSTOP                                           00020680
      JTMP=JMA2-1                                                      00020690
      JTST=IAND(JTMP,MIN)                                              00020700
      IF(JTST-MIN)200,190,200                                          00020710
  190 CONTINUE                                                         00020720
                                                                       00020730
      MASK3(JMA,JMA2)=0                                                00020740
  200 CONTINUE                                                         00020750
                                                                       00020760
      WRITE(IOUT,210)JMA,(MASK3(JMA,JMA2),JMA2=1,JSTOP)                00020770
  210 FORMAT(' ALTCKT ',I5/(13X,16I6))                                 00020780
  250 CONTINUE                                                         00020790
                                                                       00020800
                               FILL MASK2 WITH ACTUAL VALUES.          00020810
                                                                       00020820
      DO 340 JMA=1,N2ND                                                00020830
      J2ND=I2ND(JMA)                                                   00020840
      MIN=IS(J2ND,1)                                                   00020850
      JBEG=MIN+1                                                       00020860
      WRITE(IOUT,260)JMA,J2ND,MIN                                      00020870
  260 FORMAT('0 GATE=',I5,'   ONE-SET=',I5,'   CELLMIN=',              00020880
     +I5)                                                              00020890
                                                                       00020900
      DO 300 JMA2=JBEG,JSTOP                                           00020910
      JTMP=JMA2-1                                                      00020920
      JTST=IAND(JTMP,MIN)                                              00020930
      IF(JTST-MIN)300,290,300                                          00020940
  290 MASK2(JMA,JMA2)=1                                                00020950
  300 CONTINUE                                                         00020960
                                                                       00020970
      WRITE(IOUT,210)JMA,(MASK2(JMA,JMA2),JMA2=1,JSTOP)                00020980
  340 CONTINUE                                                         00020990
                                                                       00021000
                                                                       00021010
      C R E A T E   A L T E R N A T E   C I R C U I T S .              00021020
                                                                       00021030
                                                                       00021040
                                                                       00021050
                               LOOP THROUGH ALL SECOND LEVEL GATES(EXCEPT00021060
                               LAST ONE GENERATED).                    00021070
                                                                       00021080
```

```
      DO 500 JMA=2,N2ND                                                 00021090
      NFEED=IFEED(JMA,1)                                                00021100
                                                                        00021110
                                                                        00021120
                            LOOP THROUGH FEEDING THIRD LEVEL GATES TO   00021130
                            SEE IF ANY CAN BEREPLACED.                  00021140
                                                                        00021150
      IF(NFEED)500,500,350                                              00021160
  350 CONTINUE                                                          00021170
      NFEED=NFEED+1                                                     00021180
      J2ND=I2ND(JMA)                                                    00021190
      MIN2=IS(J2ND,1)                                                   00021200
      JBEG2=MIN2+1                                                      00021210
                                                                        00021220
                                                                        00021230
      DO 450 JMA2=2,NFEED                                               00021240
      IGATE=IFEED(JMA,JMA2)                                             00021250
      MFEED=IT(IGATE,1)                                                 00021260
                                                                        00021270
                                                                        00021280
                            LOOP THROUGH POSSIBLE REPLACEMENTS.         00021290
                                                                        00021300
      DO 4CC JMA3=1,N3RD                                                00021310
      IF(IGATE-I3RD(JMA3))360,400,400                                   00021320
  360 IGTMP=I3RD(JMA3)                                                  00021330
                                                                        00021340
                            LOOP THROUGH TERMS TO CHECK FOR MAXTERM     00021350
                            COVERAGE AND AVOID MINTERM INHIBITION.      00021360
                                                                        00021370
      DO 380 JMA4=JBEG2,JSTOP                                           00021380
                                                                        00021390
                            TEST IF WE ARE IN THE CELL.                 00021400
                                                                        00021410
      JTEMP=JMA4-1                                                      00021420
      JTST=IAND(JTEMP,MIN2)                                             00021430
      IF(JTST-MIN2)380,362,380                                          00021440
  362 CONTINUE                                                          00021450
                                                                        00021460
                          TEST FOR EFFECT OF OTHER                      00021470
                           FEEDING THIRD LEVEL GATES.                   00021480
                                                                        00021490
      IF(NFEED-2)366,366,363                                            00021500
  363 CONTINUE                                                          00021510
                                                                        00021520
      DO 365 JMA5=2,NFEED                                               00021530
      IF(JMA5-JMA2)364,365,364                                          00021540
  364 IF(MASK3(ICONV(IFEED(JMA,JMA5)),JMA4))365,380,365                 00021550
  365 CONTINUE                                                          00021560
                                                                        00021570
  366 CONTINUE                                                          00021580
      JTMP=MASK3(ICONV(IGTMP),JMA4)                                     00021590
                                                                        00021600
                            TEST FOR FUNCTIONAL EQUALITY.               00021610
                                                                        00021620
                                                                        00021630
      IF(JTMP*MASK2(JMA,JMA4)-IFUNC(1,JMA4))400,380,400                 00021640
  380 CONTINUE                                                          00021650
                                                                        00021660
      IFLAG=IFLAG+1                                                     00021670
      IFEED(JMA,JMA2)=IGTMP                                             00021680
      GO TO 450                                                         00021690
  400 CONTINUE                                                          00021700
                                                                        00021710
  450 CONTINUE                                                          00021720
                                                                        00021730
  500 CONTINUE                                                          00021740
```

```
                                                                      00021750
                                                                      00021760
                                                                      00021770
      U P D A T E   I 3 R D   I N   T H E   E V E N T   O F   A       00021780
        G A T E   H A V I N G   B E E N   R E M O V E D.              00021790
                                                                      00021800
                                                                      00021810
                                  CHECK WHETHER ANY 3RD LEVEL GATE    00021820
                                  HAS BEEN COMPLETELY REPLACED IN     00021830
                                  ALL CASES.                          00021840
                                                                      00021850
                                                                      00021860
                                  FIRST TEST IF ANY CHANGES WERE MADE. 00021870
                                                                      00021880
      IF(IFLAG)510,510,550                                            00021890
  510 WRITE(ICUT,520)                                                 00021900
  520 FORMAT(' ALTCKT FOUND NO CHANGES FOR FEEDING GATES')            00021910
      RETURN                                                          00021920
                                                                      00021930
                                  LOOP THROUGH I3RD.                  00021940
                                                                      00021950
  550 JMA=1                                                           00021960
  560 IF(JMA-N3RD)570,570,1010                                        00021970
                                                                      00021980
  570 WRITE(IOUT,580)JMA,I3RD(JMA)                                    00021990
  580 FORMAT(' JMA=',I5,'   NOW CONSIDERING REMOVING GATE',           00022000
     *I5)                                                             00022010
                                                                      00022020
                                  LOOP THROUGH IFEED.                 00022030
                                                                      00022040
      DO 800 JMA2=1,N2ND                                              00022050
      NFEED=IFEED(JMA2,1)                                             00022060
                                                                      00022070
                                  TEST FOR ANY FEEDING 3RD LEVEL GATES.00022080
                                                                      00022090
      IF(NFEED)800,800,600                                            00022100
  600 NFEED=NFEED+1                                                   00022110
                                                                      00022120
      DO 800 JMA3=2,NFEED                                             00022130
      IF(IFEED(JMA2,JMA3)-I3RD(JMA))800,1000,800                      00022140
  800 CONTINUE                                                        00022150
                                                                      00022160
      WRITE(IOUT,810)NRMOV                                            00022170
  810 FORMAT(1H+,40X,'THIS IS THE',I3,'TH GATE REMOVED.')             00022180
                                                                      00022190
      NRMOV=NRMOV+1                                                   00022200
      N3RD=N3RD-1                                                     00022210
      DO 850 JMA3=JMA,N3RD                                            00022220
      I3RD(JMA3)=I3RD(JMA3+1)                                         00022230
  850 CONTINUE                                                        00022240
                                                                      00022250
      GO TO 560                                                       00022260
 1000 JMA=JMA+1                                                       00022270
      GO TO 560                                                       00022280
                                                                      00022290
 1010 CONTINUE                                                        00022300
                                                                      00022310
      WRITE(IOUT,1020)IFLAG,NRMOV                                     00022320
 1020 FORMAT(' ALTCKT HAS MADE ',I5,' CHANGES AND',I5,' 3RD LEVEL '   00022330
     *,'GATE REMOVALS')                                               00022340
      RETURN                                                          00022350
      END                                                             00022360
      FUNCTION IDIST(MIN,IORDR)                                       00022361
                                                                      00022362
       THIS FUNCTION FINDS THE DISTANCE FROM THE                      00022363
       PRIMARY CUBE IN A SYSTEM OF ORDER IORDR AND                    00022364
```

```
      SOME VALUE MIN.                                                   00022365
                                                                        00022366
      J=0                                                               00022367
      DO 100 JMA=1,IORDR                                                00022368
      ITEST=2**(JMA-1)                                                  00022369
      ITEST=IAND(MIN,ITEST)                                             00022370
      IF(ITEST)100,50,100                                               00022371
   50 J=J+1                                                             00022372
  100 CONTINUE                                                          00022373
      IDIST=J                                                           00022374
      RETURN                                                            00022375
      END                                                               00022376
      SUBROUTINE STEP5(JCRIT,NCTST,NOCEL,NITMP,NSTOP,ITTMP,MCELL,       00022377
     * ICUBE,ISTMP,ICELL,NEWMAX,NEWMIN,MAXDST,MINDST)                   00022378
                                                                        00022379
        THIS SUBROUTINE DOES THE STEP5 CHOOSING FOR SVAING              00022380
        PRIME IMPLICANTS AS GATE CANDIDATES.                            00022381
                                                                        00022382
      DIMENSION NCTST(1),ITTMP(1),MCELL(1),ICUBE(1),ISTMP(1)            00022383
      DIMENSION ICELL(1),IFUNC(40,256)                                  00022384
      COMMON IN,IOUT,IPRIM,ISET,IFUNC,LEVEL                             00022385
                                                                        00022386
                          GO TO THE CRITERIA FOR THIS METHOD.           00022387
                                                                        00022388
      GO TO(50,10,20,190,30,40),JCRIT                                   00022389
                                                                        00022390
                     TEST IF THIS COUNT OF NEW MINTERMS IS GREATER      00022391
                     THAN PREVIOUSLY COVERED.                           00022392
                                                                        00022393
   10 IF(NEWMIN.LE.NCTST(2)) RETURN                                     00022394
      NCTST(2)=NEWMIN                                                   00022395
      GO TO 190                                                         00022396
                                                                        00022397
                     TEST IF THIS COUNT OF NEW MAXTERMS IS              00022398
                     GREATER THAN PREVIOUSLY COVERED.                   00022399
                                                                        00022400
   20 IF(NEWMAX.LE.NCTST(3)) RETURN                                     00022401
      NCTST(3)=NEWMAX                                                   00022402
      GO TO 190                                                         00022403
                                                                        00022404
                   TEST FOR MAXTERMS IN HARD PLACES(TOTAL DISTANCE      00022405
                         FORM PRIMARY CUBE.).                           00022406
   30 IF(MAXDST.LE.NCTST(5)) RETURN                                     00022407
      NCTST(5)=MAXDST                                                   00022408
      GO TO 190                                                         00022409
                                                                        00022410
                       TEST FOR MINTERMS IN HARD PLACES.                00022411
                                                                        00022412
   40 IF(MINDST.LE.NCTST(6)) RETURN                                     00022413
      NCTST(6)=MINDST                                                   00022414
      GO TO 190                                                         00022415
                                                                        00022416
                        TEST FOR MOST SUBFUNCTIONS(1-SETS) GENERATED.   00022417
                                                                        00022418
   50 IF(NOCEL.LE.NCTST(1)) RETURN                                      00022419
  190 NCTST(1)=NOCEL                                                    00022420
      NITMP=NSTOP                                                       00022421
                                                                        00022422
      DO 192 JMA=1,NSTOP                                                00022423
      ITTMP(JMA)=MCELL(ICUBE(JMA))                                      00022425
  192 CONTINUE                                                          00022426
                                                                        00022427
      NCTEMP=NCTST(1)                                                   00022428
      DO 202 JMA=1,NCTEMP                                               00022429
      ISTMP(JMA)=ICELL(JMA)                                             00022430
  202 CONTINUE                                                          00022431
```

```
      EXAMPLE 2 FIGURE 3 PAGE 52  IN PAPER                                0000000

 FUNCTION OF ORDER:     4
WITH     7 MINTERMS AND      0 DON'T CARES
ISWTC=     1  IDBUG=     2

 MINTERMS ARE:
   3    8   10   11   12   13   15

THIS IS CKT # 1

3RD LEVEL GATES            2ND LEVEL GATES,AND INPUTS FROM 3RD LEVEL          DIRECT INPUTS TO FIRST LEVEL
T 2 ---1                    1---       ,T 2,T 1
T 1 -11-                    --11       ,T 1
                            1-11
                            11-1

THIS IS CKT # 2

3RD LEVEL GATES            2ND LEVEL GATES,AND INPUTS FROM 3RD LEVEL          DIRECT INPUTS TO FIRST LEVEL
T 2 ---1                    1---       ,T 2,T 1
T 1 -11-                    --11       ,T 1
                            1-11
                            11--       ,T 1

THIS IS CKT # 3

3RD LEVEL GATES            2ND LEVEL GATES,AND INPUTS FROM 3RD LEVEL          DIRECT INPUTS TO FIRST LEVEL
T 2 ---1                    1---       ,T 2,T 1
T 1 -11-                    --11       ,T 1
                            11-1

CKT#    # OF GATES    # OF INPUTS
   1             7             19
   2             7             19
   3             6             15

OBVIOUSLY CKT#     3 IS THE MINIMUM WITH ONLY     6 GATES AND ONLY    15 INPUTS
```

```
H. P. LEE   EG. 6   PG 22                                                    0000000

 FUNCTION OF ORDER:     6
WITH   20 MINTERMS AND      0 DON'T CARES
ISWTC=     1  IOBUG=     2

 MINTERMS ARE:
   0     1     2     4     5     6     8     9    16    17    18    20    21    22    24    25
  34    38    40    41

THIS IS CKT # 1

3RD LEVEL GATES           2ND LEVEL GATES,AND INPUTS FROM 3RD LEVEL          DIRECT INPUTS TO FIRST LEVEL
T 1 ----11                 ------     ,T 1,T 4,T 2,T 5
T 4 --1-1-                 ----1-     ,T 1,T 4,T 3
T 2 --11--                 --1---     ,T 4,T 2,T 3
T 5 1-----
T 3 -1----

CKT#    # OF GATES    # OF INPUTS
   1             9             23

OBVIOUSLY CKT#     1 IS THE MINIMUM WITH ONLY     9 GATES AND ONLY    23 INPUTS
```

```
     EXAMPLE 2 FIGURE 3 PAGE 16  IN PROPOSAL                              0000000

 FUNCTION OF ORDER:     3
WITH     5 MINTERMS AND      0 DON'T CARES
ISWTC=     1  IDBUG=     2

 MINTERMS ARE:
   0     1     2     3     5

THIS IS CKT # 1

3RD LEVEL GATES            2ND LEVEL GATES,AND INPUTS FROM 3RD LEVEL          DIRECT INPUTS TO FIRST LEVEL
T 1 -1-                     --1       ,T 1                                        1--

CKT#    # OF GATES    # OF INPUTS
   1              3              5

OBVIOUSLY CKT#     1 IS THE MINIMUM WITH ONLY     3 GATES AND ONLY     5 INPUTS
```

VITA

John Michael Acken

Candidate for the Degree of

Master of Science

Thesis: AN ALGORITHM FOR TANT SYNTHESIS AND ITS SEQUENTIAL APPLICATION

Major Field: Electrical Engineering

Biographical:

Personal Data: Born in Kansas City, Missouri, September 27, 1951, the son of Mr. and Mrs. John H. Acken, Jr.

Education: Graduated from Nathan Hale High School, Tulsa, Oklahoma, in May, 1969; received the Bachelor of Science in Electrical Engineering degree from Oklahoma State University, Stillwater, Oklahoma, in December, 1976; completed the requirements for the Master of Science degree in May, 1978.

Professional Experience: Private Computer Programming Consultant from September, 1975 to August, 1976; employed by Oklahoma State University as a Graduate Assistant from February, 1977 to May, 1978.

Professional Organizations: Member of the Institute of Electrical and Electronics Engineers, Member of the Association for Computing Machinery.