The Map Method for Synthesis of Combinational Logic Circuits

M. KARNAUGH
NONMEMBER AIEE

THE SEARCH for simple abstract techniques to be applied to the design of switching systems is still, despite some recent advances, in its early stages. The problem in this area which has been attacked most energetically is that of the synthesis of efficient combinational that is, nonsequential, logic circuits.

While this problem is closely related to the classical one of simplifying logical truth functions, there are some significant differences. To each logical truth function, or Boolean algebraic expression, there corresponds a combinational circuit which may be constructed from a given set of appropriate components. However, minimization of the number of appearances of algebraic variables does not necessarily lead to the most economical circuit. Indeed, the criteria of economy and simplicity may vary widely for different types of components. A general approach to circuit synthesis must therefore be highly flexible. What is perhaps most to be desired is a simple and rapid technique for generating a variety of near-minimal algebraic forms for the designer's inspection.

Boolean algebra,[1] or the calculus of propositions, is a basic tool for investigation of circuits constructed from 2-valued devices. Its direct application to synthesis problems is, nevertheless, not completely satisfactory. The designer employing Boolean algebra is in possession of a list of theorems which may be used in simplifying the expression before him; but he may not know which ones to try first, or to which terms to apply them. He is thus forced to consider a very large number of alternative procedures in all but the most trivial cases. It is clear that a method which provides more insight into the structure of each problem is to be preferred. Nevertheless, it will be convenient to describe other methods in terms of Boolean algebra. Whenever the term "algebra" is used in this paper, it will refer to Boolean algebra, where addition corresponds to the logical connective "or," while multiplication corresponds to "and."

The minimizing chart,[2] developed at the Harvard Computation Laboratory, represents a step in the desired direction. It makes possible the fairly rapid derivation of near-minimal 2-stage forms. By a 2-stage form is meant a sum of products of the elementary variables, or else a product of sums of the elementary variables. These expressions may then be further reduced by algebraic factoring. The chief drawback to this method lies in the necessity of writing, and perhaps erasing, on a chart that, for n variables, contains 2^{2^n} entries. Thus, we must keep track of 1,024 entries for five variable problems and 4,096 entries for six variable problems.

E. W. Veitch[3] has suggested a method whereby results similar to those yielded by the minimizing chart can be obtained from an array containing only 2^n entries in a more rapid and elegant manner. The map method, which is explained in this paper, involves a reorganization of Veitch's charts, an extension to the use of 3-dimensional arrays, and some special techniques for diode and relay circuits.

Maps

Let the active and inactive conditions of the inputs to a combinational circuit be designated by assigning the values 1 and 0 respectively to the associated algebraic variables. An assignment of a simultaneous set of values to the n variables for a given problem will be called an input condition. There are 2^n possible input conditions.

For example, with only two variables, there are four input conditions. They may be represented graphically by the four squares in Fig. 1(A). Here, the values of variables A and B have simply been plotted along two perpendicular axes. It should be noted that squares which are adjacent, either horizontally or vertically, differ in the value of only one of the variables.

If Fig. 1(A) is cut along its horizontal midsection and the bottom half is rotated into line with the top, as in Fig. 1(B), then a representation of the input conditions for two variables is obtained along a single axis. Let us consider the squares at opposite ends of the row to be termed adjacent, as if it were inscribed on a cylinder. Then, as before, adjacent squares differ in the value of only one variable. Conversely, if two input conditions differ in the value assigned to just one of the variables, they are represented by adjacent squares.

If one also makes use of the vertical axis, one can represent the input conditions for three variables as in Fig. 2(A), and for four variables as in Fig. 2(B). In the latter case, opposite ends of each row or column should be considered adjacent, as though the figure were inscribed on a torus.

The labels on the diagrams may be simplified as shown in Fig. 3. The rows or columns within a bracket are those in which the designated variable has the value 1, while it is 0 elsewhere.

A combinational circuit of the type under consideration has a 2-valued output which is a function of the input condition. The synthesis problem may be said to begin with the specification of this functional dependence. Such information may be represented on a map as follows: Place a 1 in each square which represents an input condition for which the output is to have the value 1. The other squares may be imagined to contain zeroes.

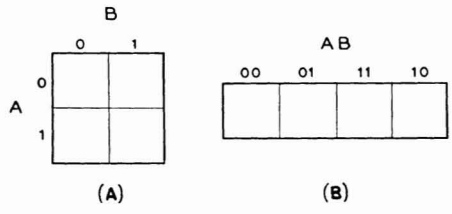


Fig. 1. Graphical representation of the input conditions for two variables

(A) Along two axes
(B) Along a single axis

Synthesis of 2-Stage Forms

Consider the function mapped in Fig. 4(A). Its algebraic realization is the product $A'BC'D$, where the primes indicate negation or complementation, for $A'BC'D=1$ if, and only if, $A=0$, $B=1$, $C=0$, and $D=1$.

Let us define a complete product to be a product in which each of the variables appears as one factor, either primed or

Paper 53-217, recommended by the AIEE Communication Switching Systems Committee and approved by the AIEE Committee on Technical Operations for presentation at the AIEE Summer General Meeting, Atlantic City, N. J., June 15–19, 1953. Manuscript submitted March 17, 1953; made available for printing April 23, 1953.

M. KARNAUGH is with the Bell Telephone Laboratories, Inc., Murray Hill, N. J.

The author wishes to express his indebtedness for many valuable suggestions, help, and encouragement to E. F. Moore, K. Goldschmidt, and W. Keister, all of the Bell Telephone Laboratories.

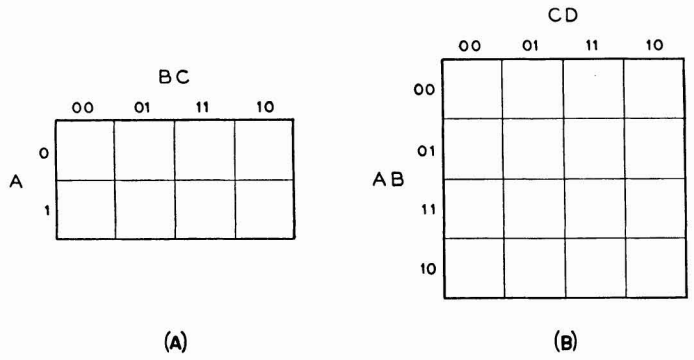


Fig. 2. Graphical representations of the input conditions for three and for four variables

not. Then any function whose map contains a single 1 may be represented by a single complete product. Each factor is primed if, and only if, it has the value 0 at the square in question. Because each square that contains a 1 gives rise to a product, such squares will be called p-squares.

If the map of a function contains k p-squares, then the function may be represented by the logical sum of the corresponding k complete products, each selected by this rule. This form of representation is the complete disjunctive normal form of the calculus of propositions. It is often the starting point for algebraic simplification.

However, it is usually possible to write down a more economical representation than a complete normal form by direct inspection of a map. Consider the function whose map is shown in Fig. 4(B). Its complete disjunctive normal form is $AB+A'B$. This is easily reducible algebraically: $AB+A'B=B(A+A')=B$. Now note that the p-squares on the map are precisely that set for which $B=1$.

Let us define a subcube to be the set of all squares on a map over which certain of the variables have fixed values. A subcube formed entirely of p-squares will be called a p-subcube.

Each p-subcube may be regarded as the map of a product formed according to the rules:

1. The factors of the product are those variables whose values are fixed within the subcube.

2. A factor is primed if, and only if, its value within the subcube is 0.

Fig. 5 shows some typical p-subcubes and the corresponding products. Each p-subcube may be thought of as a simply connected square or rectangular group of p-squares, if it is recalled that opposite ends of columns and rows are adjacent.

If m variables are not fixed in a given subcube, it is said to be m-dimensional, and it contains 2^m squares. A single square is thus a zero-dimensional subcube. Note that the larger p-subcubes correspond to products having fewer factors, since fewer variables are fixed in them.

It is now easy to see how to obtain economical 2-stage forms from maps. The rules are:

1. Choose a set of p-subcubes which includes every p-square at least once. In general, it is desirable to make the selected subcubes as large and as few in number as possible.

2. Write down the sum of the products which correspond to the selected p-subcubes. This gives the desired expression.

As an example of this procedure, we can, for the function mapped in Fig. 6, make the selection

$$f=AC'+A'CD+BCD$$

An alternate procedure is possible that leads to a product of sums, that is, a conjunctive normal form instead of a sum of products. First, this procedure is used to obtain an expression for the negative of the function mapped. This is done by considering the empty squares to be the new p-squares. In the case of Fig. 6

$$f'=A'C'+CD'+AB'C$$

The function desired, which is the negative of this, is now obtained by the simul-

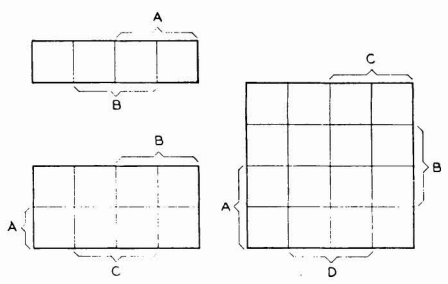


Fig. 3. Input representations with simplified labels

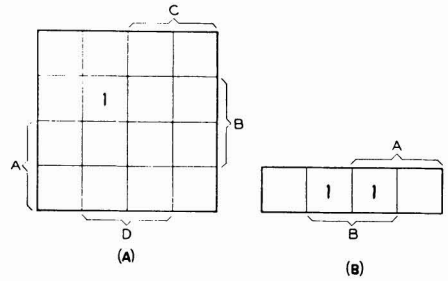


Fig. 4. Maps of two functions

(A) $f=A'BC'D$

(B) $f=A'B+AB=B(A'+A)=B$

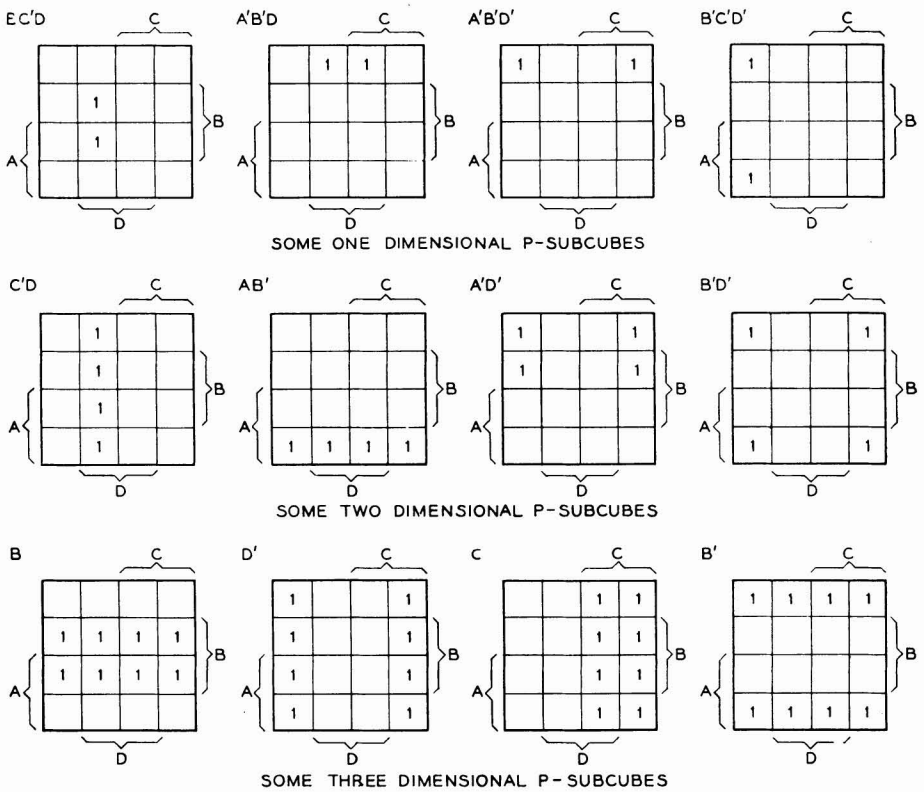


Fig. 5. A number of typical p-subcubes and the corresponding algebraic products

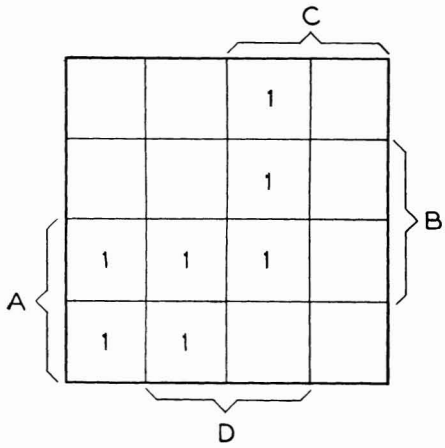


Fig. 6. Map of a function

taneous interchange of primes and non-primes, and of multiplication and addition signs.

Thus

$$f=(A+C)(C'+D)(A'+B+C')$$

Both of these procedures have been proposed by Veitch.[3]

Minimal 2-Stage Forms

In combinational diode circuits, there is usually one diode per input lead to every stage. For 2-stage circuits, this means one diode per appearance of each algebraic variable plus one diode per product, or per sum, of these variables. It is often a simple matter to minimize rigorously the number of diodes used in such a circuit.

Consider Fig. 7(A). The dotted lines correspond to the choice of p-subcubes.

$$f=B+AC$$

Now note that asterisks are placed in two of the p-squares, so chosen that no single p-subcube includes both of them. Hence at least two p-subcubes are required. Furthermore, the selected p-subcube containing each asterisk is of maximum possible dimensionality. Hence each of the corresponding products contains the minimum number of factors.

The same kind of proof must be carried out for the alternate procedure, as illustrated in Fig. 7(B). Here we have

$$f'=A'B'+B'C'$$
$$f=(A+B)(B+C)$$

This is not as good, however, as the previous result, which we have now proved to be minimal in

1. Number of terms
2. Appearances of the variables
3. Diodes

This proof depends upon the fact that no one pair of asterisks lies in the same p-subcube. In some cases it may be found that only k asterisks can be placed on a map in this manner, and yet more than k terms are required to represent the function. When this occurs, a proof that at least $k+1$ terms are necessary can be carried through by contradiction. When the attempt to associate a p-subcube with each asterisk is made, it will be found impossible to include all p-squares in the k p-subcubes so selected.

Factoring by Inspection

When circuits are not restricted to the 2-stage variety, it is sometimes advantageous to reduce further the 2-stage forms by algebraic factoring. It is of some importance to show that factoring may also be carried out directly by inspection of a map.

For example, the function mapped in Fig. 8 is

$$f=A'B'+B'C=B'(A'+C)$$

Since both the chosen p-subcubes lies within subcube B, the presence of the common factor is established by inspection.

Occasionally, observation of the possi bilities for factoring will determine the selection of subcubes and lead to a better circuit than would otherwise be obtained. In the case of Fig. 6, the choices

$$f=AC'+A'CD+BCD=AC'+CD(A'+B)$$
$$f=AC'+A'CD+ABD=A(C'+BD)+A'CD$$
$$\text{or}\qquad =AC'+D(A'C+AB)$$

lead to equally good 2-stage forms; but the former yields the best factored form. Inspection of the map indicates that p-subcube BCD lies in CD along with $A'CD$, thus providing two common factors, while the alternative choice of ABD will give only a single common factor in either of two ways. When inspecting the map, it is not necessary to think of these subcubes by name as we must in the text, but merely to observe their relations, as sets of p-squares.

Even more extensive use of the set theoretic union (our $+$) and intersection (our $\cdot$) relations is possible. Consider Fig. 9. Algebraically, we get

$$f=A'B'C'D+A'B'CD'+ABC'D+ABCD'$$
$$=A'B'(C'D+CD')+AB(C'D+CD')$$
$$=(A'B'+AB)(C'D+CD')$$

But it can be seen directly that the four p-squares form the set which is the intersection of the union of $A'B'$ and AB and the union of $C'D$ and CD'. Thus $f=(A'B'+AB)(C'D+CD')$, as illustrated by the dotted lines.

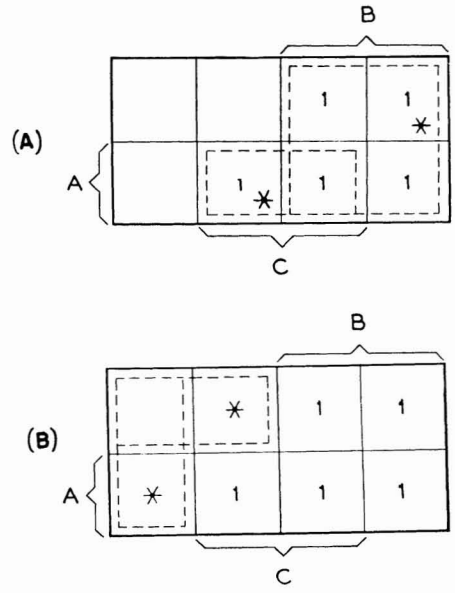


Fig. 7. Maps used to minimize a diode circuit

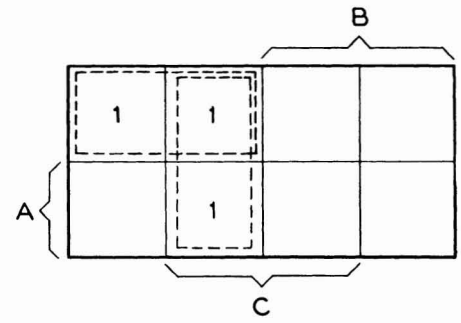


Fig. 8. Map of a factorable function

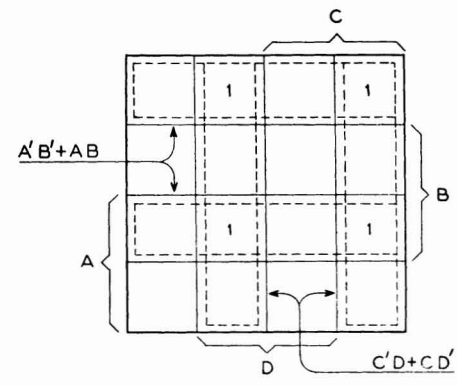


Fig. 9. Set theoretic interpretation of a map

"Don't-Care" Conditions

Very often, the output of a circuit is subject to less rigid restriction than the assignment of a definite value, 0 or 1, for some input conditions. The simplest such case is that of no restriction at all This may occur because the input conditions in question never are realized in practice, or because the output has no effect in those cases. We shall designate

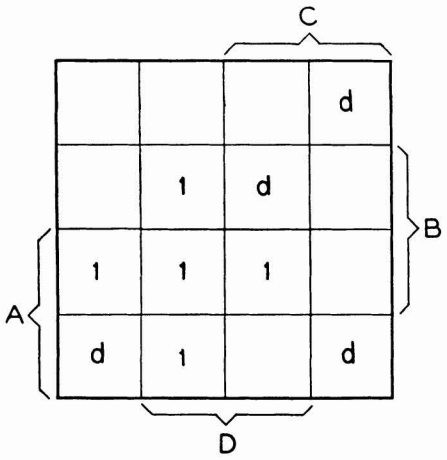


Fig. 10. Map of an incompletely specified function

such don't-care conditions by placing the symbol d in the appropriate squares.

It is usually quite simple to make an economical assignment of values to the d-squares by inspection of a map. Since these are at the disposal of the designer, it is to his advantage to employ them so as to simplify the resulting circuit.

The best 2-stage form for the function in Fig. 10 is

$$f = AC' + BD$$

obtained by setting the two d's on the right

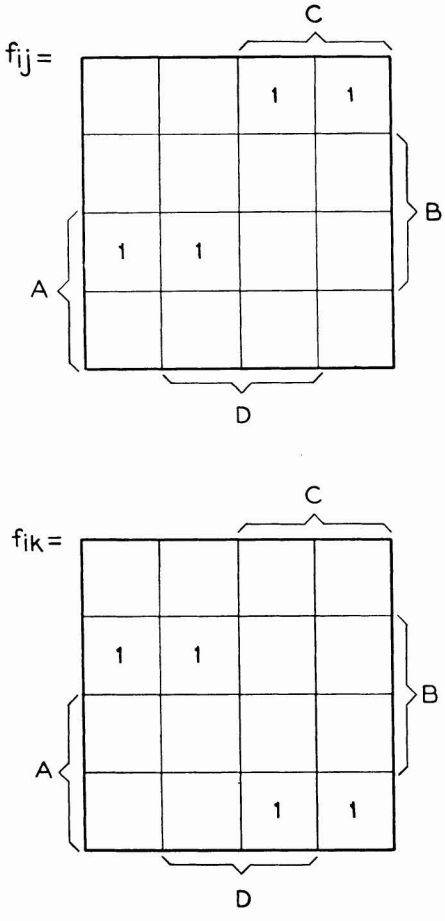
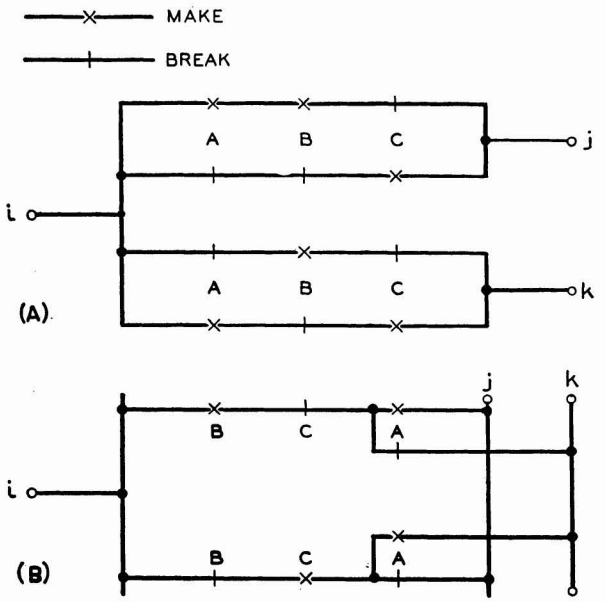


Fig. 11. A 2-output problem

Fig. 12 (right). Synthesis of a 2-output circuit

(A) Two separate contact networks

(B) Networks disjunctively combined



equal to 0, and the other two equal to 1.

The rule for making such choices is as follows: Assign values to the d's which enlarge and combine the necessary p-subcubes as much as possible but do not make necessary the selection of any additional subcubes.

The ease with which don't-cares can be properly evaluated is one of the major advantages shared by the minimizing chart, Veitch chart, and map methods in varying degree.

Disjunctive Combination in Relay Nets

The map method, inasmuch as it yields expressions in Boolean algebra, can be used to design 2-terminal, series-parallel relay contact networks, but not bridge-type 2-terminal networks. Hence, many 2-terminal contact networks designed by means of the map method will not be minimal in contacts or springs. This will be true, in particular, of the symmetric circuits.[4]

However, in the case of complicated, multioutput networks, the map method may be a very effective tool. Suppose that terminal i is a ground, to be connected through networks f_{ij} and f_{ik} to the output terminals j and k respectively. The specifications for f_{ij} and f_{ik}, which are networks on the contacts of relays A, B, C, D, are mapped in Fig. 11. If each net is synthesized separately, there results the circuit of Fig. 12(A). In Fig. 12(B), it is shown how, with a slight rearrangement, parts of the upper paths to j and k can be combined, as can parts of the lower paths. This results in a saving of four contacts.

The second circuit is completely equivalent to the first, for the transfers on relay A prevent any sneak paths between terminals j and k. While disjunctive combinations of this sort are certainly not new to the relay art,[5] this section is included to show how they may easily be recognized on maps, and hence how they play a part in the selection of subcubes.

Note that the paths ABC' and $A'BC'$, which give rise to one of the combinations, differ by only a prime on A. The corresponding subcubes in Fig. 11 are seen to be related by a simple displacement. The same is true for the other pair of p-subcubes.

A little practice will enable the designer to evaluate the various possibilities for factoring and disjunctive combination by inspection of the maps. It will then be a simple task to make a good choice of p-subcubes.

Unnecessary Contacts

It is of interest to note that for any given function some of the variables or their primes may be unnecessary. That is, it is possible to find an algebraic representation of the function in which these variables, or negated variables, do not appear. Hence the corresponding relay

Table I. Specifications for a Coded Decimal Digit Translator

Digit	1	2	4	5	Z	O	T	F	S
0	0	0	0	0	0	0	0	1	1
1	1	0	0	0	1	1	0	0	0
2	0	1	0	0	1	0	1	0	0
3	1	1	0	0	0	1	1	0	0
4	0	0	1	0	1	0	0	1	0
5	0	0	0	1	0	1	0	1	0
6	1	0	0	1	0	0	1	1	0
7	0	1	0	1	1	0	0	0	1
8	1	1	0	1	0	1	0	0	1
9	0	0	1	1	0	0	1	0	1

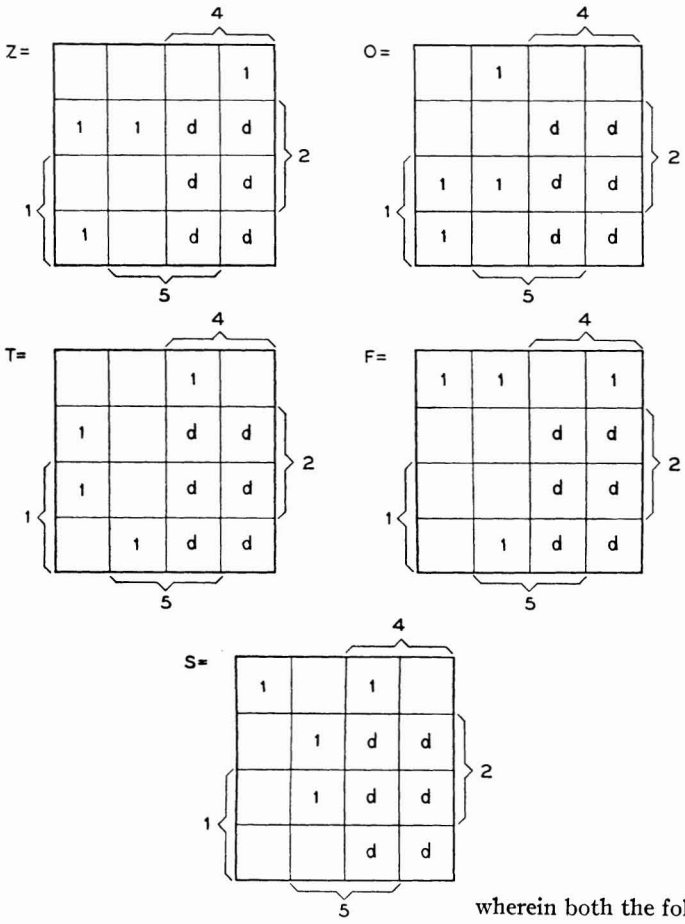


Fig. 13. A translator problem

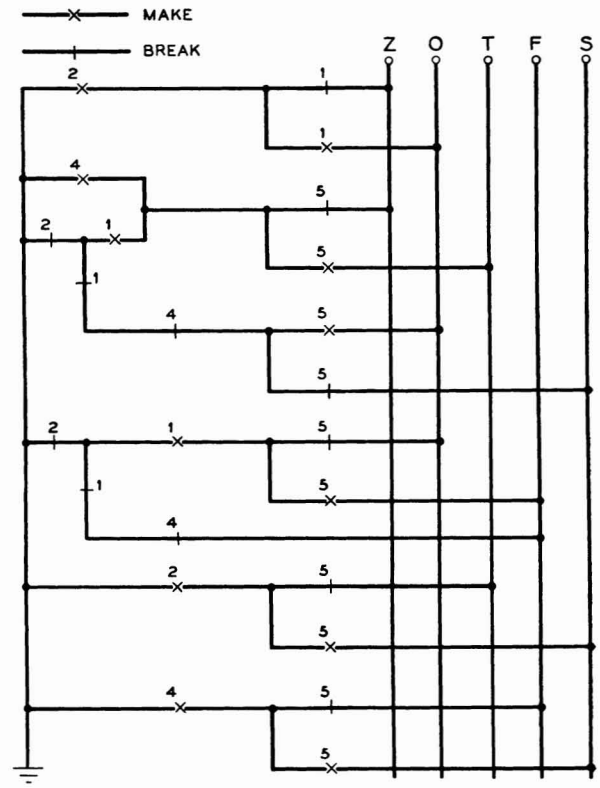


Fig. 15. The finished translator network

contact network will not contain make-contacts, or break-contacts, on some of the relays.

For example, the functions in Fig. 11 are shown on four-variable maps, but they may be realized in terms of only three variables, as in Fig. 12. Neither D nor D' is necessary.

In this case, it can be seen at a glance that the patterns appearing in the D and D' subcubes in both maps are identical. Therefore the output is independent of the value assigned to D. This is a case wherein both the following rules hold:

1. A function may be represented without the appearance of an unprimed variable, say D if, and only if, to each p-square in subcube D there corresponds an adjacent p-square in subcube D'.

2. A function may be represented without the appearance of D' if, and only if, to each p-square in subcube D' there corresponds an adjacent p-square in D.

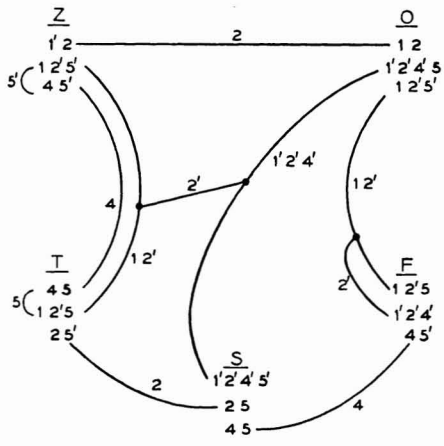


Fig. 14. Work sheet for synthesis of the translator

Illustrative Example: A Relay Translator

Suppose it is desired to find a relay contact network to translate coded decimal digits from a 1-2-4-5 code to 2-out-of-5 code. The five outputs will operate the relays Z (zero), O (one), T (two), F (four), and S (seven). The required translation properties are listed in Table I. The unarithmetic representation for zero is standard in the 2-out-of-5 code.

The remaining six input conditions for the 1-2-4-5 relays are unused or don't-care conditions. However, it is required that none of these conditions results in operation of zero or two of the five output relays. From these specifications, one obtains the five maps in Fig. 13.

At this point, p-subcubes must be selected, and the desirability kept in mind of factoring and disjunctive combinations. The chosen p-subcubes are listed in Table II, where the numbers in parenthesis indicate the order in which they were selected. This should be followed on the maps in order to see how the terms will combine.

A check on the six d-squares now shows that each of them has been taken = 1 on at least three of the maps. Hence the restriction on unused conditions has been satisfied, and no changes need be made in Table II.

The worksheet on which the network is planned is shown in Fig. 14. The lines drawn between terms designate disjunctive combinations or factoring; and the symbols adjacent to the lines indicate which contacts are shared in each case. A careful comparison of this worksheet with the resulting network, shown in Fig. 15, will enable the reader to understand both.

Table II. A List of Selected p-Subcubes

Z		T		S		O		F
(1) 45′	(2)	45	(10)	1′2′4′5′	(6)	12	(8)	12′5
(3) 12′5′	(4)	12′5	(12)	25	(7)	12′5′	(13)	45′
(5) 1′2	(11)	25′	(14)	45	(9)	1′2′4′5	(15)	1′2′4

Three-Dimensional Maps

Up to this point, we have discussed functions of no more than four variables. If it is desired to increase the number of variables on a map, two possibilities suggest themselves:

1. Increase the number of variables plotted on each axis.
2. Use three mutually perpendicular axes instead of two.

Both methods are feasible. If method 2 is employed, then for (even) n variables, we will have $n/2$ on each axis. This means an array of $2^{n/2}$ by $2^{n/2}$ squares. However, with more than two variables on an axis, the definition of adjacence must be extended rather tenuously and subcubes become more difficult to recognize. This scheme is like the one originally suggested by Veitch.[3]

We have chosen method 2, which allows a 50-per-cent increase in the number of variables without any extension of the rules. Thus, for six variables, the methods we have described still apply, but in three dimensions.

A suitable framework is shown in Fig. 16. It consists of four 6-inch square plexiglass sheets supported at 1-1/2-inch intervals by rods of the same material. The rods and sheets are glued together. The author has been told that the 3-dimensional ticktacktoe boards sold at some toy shops under various names are satisfactory.

Each sheet is ruled at 1-1/2-inch intervals parallel to both pairs of edges. Thus we have a 4-by-4 array of squares on every sheet. The plexiglass framework enables us to do away with the writing and erasing which would be necessary when dealing with similar problems by other methods. In using it, we employ movable markers, such as 7/8-inch plastic roulette chips. The following scheme is suggested:

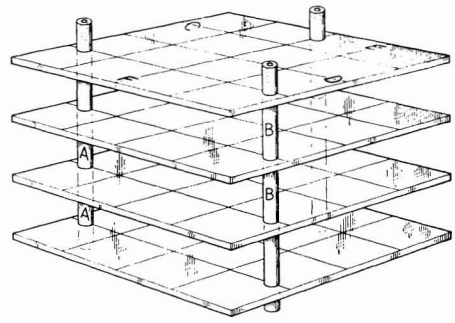


Fig. 16. The cube: a 3-dimensional plastic framework for maps

1. Mark all p-squares with white chips.
2. Mark all d-squares with black chips.
3. As subcubes are selected, mark each one with a set of distinctively colored chips.

Chips of eight or nine different colors are usually sufficient to make all the selected subcubes easily distinguishable. The corresponding products are then found by means of labels on the edges of the plastic cube.

One satisfactory labeling scheme is shown in Fig. 16. The two bottom planes are A, while the middle two are B. The variables C, D, E, and F are arranged on each plane as on the top, each letter serving to label two rows or columns. Opposite ends of any row, column, or vertical on the cube must be considered adjacent. Then every subcube may be thought of as a rectangular parallelepiped with edges 1, 2, or 4 units long. For multioutput problems, it is best to have a set of cubes, one per output.

The extension to seven variables is probably best accomplished by placing two cubes side by side. Corresponding squares in the two cubes must be considered adjacent when looking for p-subcubes. Eight variables can be handled with a set of four cubes, and nine variables require eight cubes. In the latter case, it is convenient to make them so as to stack easily into two layers of four each. Beyond nine variables, the mental gymnastics required for synthesis will, in general, be formidable. Other methods are even more limited in this respect. Outstanding exceptions to this limitation are the symmetric and positional circuits, discussed by Keister, Ritchie, and Washburn.[4]

Conclusions

Employment of the map method seems to be profitable when nontrivial problems in combinational circuit synthesis arise. Its most important advantages appear to be flexibility and speed. Further, if such problems arise frequently, it is advantageous to have a method, such as this, which can be learned and used effectively in a short time by designers new to the field.

References

1. The Design of Switching Circuits (book), **William Keister, A. E. Ritchie, S. H. Washburn.** D. Van Nostrand Company, New York, N. Y., 1951, chap. 5.
2. Synthesis of Electronic Computing and Control Circuits (book), **Staff of the Harvard Computation Laboratory.** Harvard University Press, Cambridge, Mass., 1951, chap. 5.
3. A Chart Method for Simplifying Truth Functions, **E. W. Veitch.** *Proceedings*, Association for Computing Machinery, Pittsburgh, Pa., May 2, 3, 1952.
4. See reference 1, pp. 55–64.
5. See reference 1, pp. 295–297.

Discussion

S. H. Caldwell (Massachusetts Institute of Technology, Cambridge, Mass.): When Shannon published his classic paper on analyzing relay and switching circuits,[1] the engineer was given a powerful method for the solution of many problems in the field of switching circuits. Unfortunately, when one attempted to use the method, there arose a peculiar sort of frustration. Given a circuit which had been designed by the methods of trial and error prevalent at the time, it was readily possible to use Shannon's techniques to investigate alternative forms. In particular, the switching algebra could be used directly for the simplification of contact networks. But the situation was different with respect to the synthesis of a network (unless it could be described by a symmetric function). In the general case, it was necessary to resort to a word statement of the required circuit characteristics and then convert this to an algebraic statement.

For simple problems, and especially those which involved a small number of variables, no difficulty was encountered, but because of their very simplicity such problems rarely needed the algebraic approach. When problems of any magnitude were attempted, the method broke down both because of the difficulty of writing word statements and because of the difficulty of converting bulky word statements into algebraic expressions.

These difficulties were resolved by the adaptation of the logical truth table into the familiar table of combinations (see ref. 1 of the paper). This mechanism enabled the designer to state his requirements in an orderly manner, and gave him a systematic means for checking the completeness of his reasoning. Moreover, the transition from the table of combinations to an equivalent algebraic statement became almost a matter of routine, depending on individual preference for simplifying the algebraic expression by inspection of the table or by algebraic manipulation.

The arrays described by Veitch (see ref. 3 of the paper) and by Mr. Karnaugh represent further development of the table of combinations into forms which are more compact, and which also have the property of making more evident the ways in which the algebraic expression of a switching function can be simplified. Of course, the end result desired in all cases is a minimization of the required circuit, whatever we mean by the word "minimization."

The problem of manipulating functions of many variables is much like the problem the physicist had in his development of mathematical models of atomic structure. Over a period of years he succeeded in

getting better and better mathematical solutions for the hydrogen atom, but none of his methods really worked when he tried to add just one more electron. Similarly, in these various methods for reducing switching functions to minimal forms we seem to be producing better and better ways for reducing functions of four variables, but we are still rather unhappy about five and six variables. The author's plastic cube for the treatment of six variables is an ingenious extension of his four-variable array, and it certainly has the reduction properties he ascribes to it. It does not, however, have the neatness of display which is a feature of the plane map; groupings of variables are not as immediately evident, and alternative groupings are even less apparent.

Mr. Karnaugh rightly points out that the search represented by this paper is in its early stages. It should be added that the need for better methods for handling the problem in more than four variables will become acute, and it is a problem worthy of the best thinking. Recent developments in the synthesis of sequential circuits show that the end result of a sequential synthesis is a combinational problem. It is a multiple-output problem in many variables, and has ramifications which will tax the best efforts of the circuit designer. Among the possibilities for meeting this problem is that of mechanizing the process involved in the map method.

Incidentally, I am not impressed by the drawbacks attributed to the Harvard Computation Laboratory minimizing chart. The large number of entries involved is no drawback in these days of cheap duplication processes. Keeping track of the entries is really a simple routine. In using the chart for the realization of six-variable functions with don't-care conditions, I find that one rarely has to complete the vertical ruling of the entire chart because the required conditions are usually satisfied with terms at the left-hand side of the chart. In some cases one finds a condition which is satisfied by only one possible minimal term, where the acceptance of that term in turn specifies the nature of one or more don't-care conditions. Of course, the six-variable cube inherently contains the same information, but it is doubtful that its display gives the designer quite as much immediate guidance as he gets from the minimizing chart.

REFERENCE

1. A SYMBOLIC ANALYSIS OF RELAY AND SWITCHING CIRCUITS, **Claude E Shannon.** *AIEE Transactions*, vol. 57, 1938, pp 713–23.

M. Karnaugh: In view of Professor Caldwell's remarks about mechanization, it appears to be desirable to restate the reasons for presenting this paper.

The map method, in its present form, is likely to be useful in two ways: as a pedagogic device, for the introduction of ideas about logic circuits and their synthesis, and also as a desk-top aid to the working engineer.

In making full use of the human faculty for recognizing geometric patterns at a glance, the map method supplies a number of short cuts to synthesis that are not as easily found by other methods. On the other hand, the development of machine which can recognize such relationships has only begun. If one mechanizes the map method in a more conventional way, using a repetitive scanning technique, then the result is similar to a mechanization of the Harvard minimizing charts and no special advantages are expected.

The minimizing charts, which represent one of the first significant advances over purely algebraic manipulation, have proven their usefulness in practice and will undoubtedly do so even more convincingly when machines are programmed to work along the same lines. However, it has been the author's experience that maps present the specifications for a logic circuit in a form more easily used by the human operator. Here, habit and taste enter the picture and it would be unwise to dwell on this point.

For those who are relatively new to the problem under discussion, it is suggested that a number of problems be worked by both methods. It is of interest to see how they are related, and each will throw some light on the operation of the other.

The Use of Steel Sheet for the Construction of Shielded Rooms

A. M. INTRATOR
ASSOCIATE MEMBER AIEE

LOW-LEVEL electronic or electrical measurements are particularly susceptible to errors introduced by external electromagnetic influences. The coupling of spurious electromagnetic energy into a measuring system not only may result in the receipt of false information but also can sometimes cause the complete masking of the desired data as well. For these reasons, many low-level measurements, such as the determination of crystal characteristics, filter insertion loss, noise measurements, and the like, must be made in a location as free as possible from such interference. In a laboratory, such isolation from interference is usually achieved by completely enclosing an area in copper or bronze screening. By shielding off a region relatively free of external interference in this way, a working area is provided within which sensitive electronic measurements can be made.

Certain instrumentation requires a much higher degree of freedom from extraneous influences than can be obtained in screened enclosures. A reduction in the shielding efficiency of screened booths occurs at the lower frequencies because of practical limitations in wire size and at higher frequencies because the wave lengths begin to approach the dimensions of the mesh openings. Fig. 1 shows a typical attenuation curve of a screened room. Because sheet metal presents neither of these difficulties, it is often used instead of screening, to enclose those areas in which a high degree of shielding is required.

Copper sheet has ordinarily been used for this purpose, although copper-clad steel has been used in some cases. Such rooms are usually double-walled; the inner and outer sheet-metal walls are spaced about 4 inches apart and are insulated from each other except at the point where the power line enters the room. Ordinarily the walls are hung upon a kiln-dried, wax-impregnated wood frame and all seams and mounting nails or bolts are completely soldered over to reduce the possibility of energy leakage into the room. Special seals are used to insure continuous metal-to-metal contact around the periphery of the door. Air is introduced through "wave guide below cutoff" vents, the cutoff frequency being determined by the expected top operating frequencies in the room. All power lines entering the room are filtered.

These rooms are very expensive, having ranged in cost from about $10,000 for small rooms to $100,000 for much larger

Paper **53-197,** recommended by the AIEE Instruments and Measurements Committee and approved by the AIEE Committee on Technical Operations for presentation at the AIEE Summer General Meeting, Atlantic City, N. J., June 15–19, 1953. Manuscript submitted February 11, 1953; made available for printing March 30, 1953.

A. M. INTRATOR was formerly with the United States Naval Civil Engineering Research and Evaluation Laboratory, Port Hueneme, Calif., and is now with the General Electric Company, Syracuse, N.Y.

The author wishes to express his appreciation to the members of the staff of the Stanford Research Institute Engineering Division, who performed most of the work described in this paper; to Dr. D. L. Benedict for his efforts in directing the Stanford Research Institute program; and to Dr. C. R. Freberg, formerly of the United States Naval Civil Engineering Research and Evaluation Laboratory, for the suggestions which led to this investigation.