SOME REMARKS ON APPLICATION OF HIGHER ORDER KARNAUGH MAP TO RELIABILITY ORIENTATION

TRAN VAN DAC

Department of Precision Mechanics and Applied Optics, Technical University, H-1521 Budapest.

Received September 16, 1986
Presented by Prof. Dr. O. Petrik

Abstract

In this paper the Karnaugh map is performed with some practical contributions from the point of engineering applications. The properties of adjacency of Karnaugh maps are more clearly explained by the equivalence existing between its planar representation and its axonometrical one. Some algorithms and examples are presented for a more easy understanding of the method of this map. A few reversion problems are presented and solved which seem important for system reliability evaluations and circuit designs.

To avoid misunderstanding and save time to the reader a short appendix is given.

1. Introduction

The Karnaugh map was introduced in 1953 by M. Karnaugh as a method for synthesis of combinational logic circuits [1]. From a technical view-point the Karnaugh map is probably the simplest and fastest tool for handling a set of engineering tasks connect to some problems of discrete algebraic structures. For instance one may use it to represent the Boolean functions or multiple valued discrete functions which can be well used for system reliability evaluations and network designs [2], [3]. The power of the Karnaugh map lies in its utilization concerning ability of the human mind to perceive patterns in the pictorial representation of data. In a few areas, namely in the lattice theory, as a field of algebra, the main drawback of the Karnaugh map representation for detecting prime implicants and prime implicates comes from the fact that it is practically limited to discrete functions as binary logic functions as well as multiple value ones.

The aim of this paper is, first, to give a short summary of this representation method and, second, to contribute to the expansive application of this method for the larger dimension limit.

Our opinion is that the Karnaugh map is not only a tutorial tool, as some authors sometimes stated, but a very good one for solving concrete technical problems, namely the reliability calculations and evaluations concerning the ability of decomposition of systems of higher complexity [4—7], further on it is a tool to explain pictorially some other methods extended from binary basis for engineers whose mathematical background is not sufficiently modern.

2. The original basic concept of the Karnaugh map

2.1. Karnaugh maps in binary systems

A Karnaugh map structure is an area which is subdivided into 2^n cells—one for each possible input combination for Boolean functions of n variables. Of these cells half are associated with an input value of 1 (e.g. the truth value) for one of the variables, and the other half are associated with an input value of 0 (e.g. the false value) for the same variable. This association of cells is done for each variable, with the splitting of the 2^n cells yielding a different pair of halves for the distinct variable. Namely, if the Boolean function of one variable, is, say x_1, then it would be according to Karnaugh map as represented in Fig. 1a, where $\bar{x}_1$ is NOTx_1 or the negation of x_1. The cells labeled x_1 and $\bar{x}_1$ are the halves that are associated with input value of 1 and 0 for x_1 respectively. The Boolean function of two variables x_1 and x_2 would be according to Karnaugh map shown in Fig. 1b, where one half on cells (the right column) is assigned to the input value of 1 for x_1 and where the left column is the half assigned to the input value of 0 for the x_1. Note that the four cells were split into halves in two different ways, one for each variable. Similarly, a 3-variable Karnaugh map could be like Fig. 1c and a 4-variable map is given in Fig. 1d. A 5-variable Karnaugh map can be represented by two 4-variable Karnaugh maps given in Fig. 1e where one of them is associated with a 0 value for x_5 (the left map in the diagram) and the remaining map (the right one) with value 1 for x_5. The 6-variable map would be two 5-variable maps (see Fig. 1f), and so on. In this way one may see that there are difficulties in case of a function of higher-order due to the complexity of the map. But in the following section with a little modification in representation the problem in fact may become easier.

Before dealing with the multiple valued system let us see a very important problem of Karnaugh map, that is, its adjacent property, as by means of it one can represent the Boolean function or extract the prime implicants and prime implicates of both switching functions and multiple value discrete functions.

Note that the labeling used in Karnaugh maps may appear to be

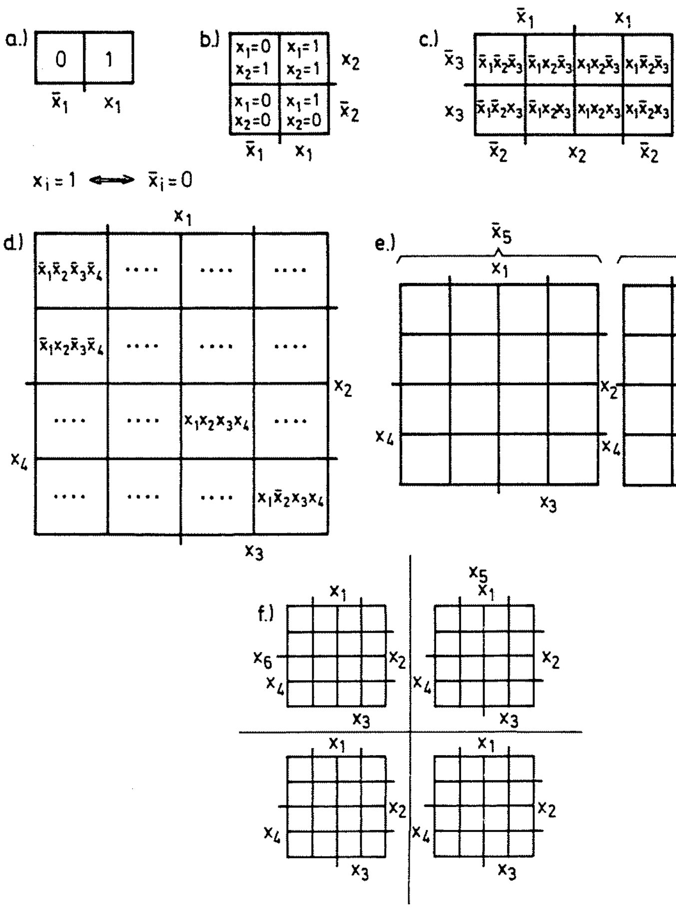


Fig. 1

arbitrary. However, there is one important concept of adjacency involved in these maps, namely, the n-tuples adjacent to one another should also appear in adjacent cells. This arrangement, however, is not always possible due to the planar representation of the Karnaugh maps. One can overcome this difficulty easily if one interprets as adjacent not only the internal cells adjacent to one another, but also the cells on opposite edges. For example the map in Fig. 2a has cell d adjacent to cells c, h, a and k, similarly, the cell h is adjacent to cells d, g, j and e while, in the normal case (at the internal cell) of cell g its adjacent cells are c, f, i and h. For a 5-variable map each of the 4-variable maps is assumed to be connected as earlier described, but also one of the maps is adjacent to the corresponding cell in the other map, namely, in Fig. 2b the cell $\bar{g}$ is also adjacent to g.

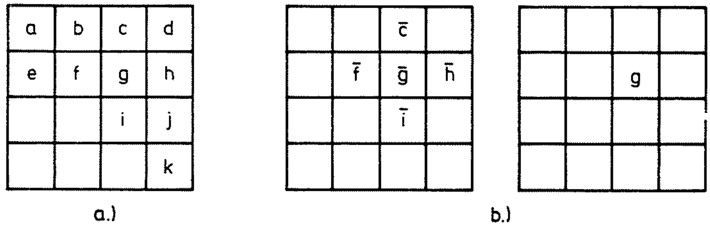


Fig. 2

2.2. Karnaugh maps in multiple-valued (or multistate) systems

Recall that due to the adjacency properties required for extraction of implicants and implicates, the entries are to be arranged so that any pair of entries immediately adjacent to each other (horizontally or vertically) must correspond to a pair of input conditions that are logically adjacent, i. e. that differ by a single unit in one single of their coordinates. This problem will be discussed in the following section for the sake of easier understanding.

By means of Karnaugh map there are no difficulties in the representation of multiple value discrete functions having two multiple value discrete variables. For instance, see the examples given in Fig. 3. Difficulties, however, occur when the number of variables is more than two. By axonometrical representation, as we shall see in the following, this problem may be overcome to some extent.

$x_2 \backslash x_1$	0	1	2	3
0	0	1	0	2
1	0	2	1	1
2	1	1	0	2
3	2	0	1	2

$f: \{0,1,2,3\}^2 \to \{0,1,2\}$

a)

$x_2 \backslash x_1$	0	1	2	3	4
0	0	0	0	1	1
1	0	0	1	1	1
2	1	1	1	2	3
3	1	2	3	4	5

$f: \{0,1,2,3\} \times \{0,1,2,3,4\} \to \{0,1,2,3,4,5\}$

b.)

Fig. 3

3. Representation of Boolean function and extraction of prime implicants and prime implicates

3.1. Representation of the Boolean function [2]

We do not want to discuss here in detail how to represent the Boolean functions by means of Karnaugh maps, or reversely, how to receive the Boolean function from its given Karnaugh map as the given value table. The detailed instruction may read in [2] and [3]. Note, nevertheless that between this map and the Boolean function there exists a one-to-one functional relationship, i. e. the map is reversible.

Via some simple examples using the block concept [2] this can be seen without any difficulty.

Example 1. Given a Karnaugh map of Boolean function $f(x_1, x_2, x_3, x_4)$ (see Fig. 4), let us find the function!

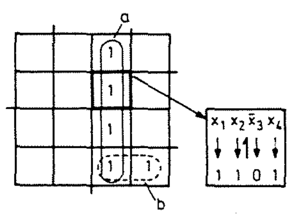


Fig. 4

Solution. First, let, conventionally, the ordered tuple (or string) of variables be $x_1x_2x_3x_4$, in this case we suggest a procedure as follows.

Complete every cell with value 1 (the true value of the Boolean function) with the corresponding tuple of variables, not forgetting the complement property of variables if it really exists. Note that in binary systems beside the weight of every cube is 1 or 0,

$$x_i = \begin{cases} 1 & \text{iff } x_i \text{ is true (or it lies on the true half of Karnaugh map)} \\ 0 & \text{(or false) otherwise} \end{cases}$$

In addition write to the same cell the value of $x_i(i = 1, 2, 3, 4$ or briefly $i = \overline{1,4})$ for the sake of easier handling (see, for example, the framed cells in Fig. 4). In practice the cells are not designated with variable tuples by necessity, and in a concrete case we treat with the above procedure also the cells containing value 1, because otherwise the weight of the cubes equals zero. In the given case there are two cubes, each of which consists of 4 cells in column assigned by pattern a and the consists of two cells in the row assigned by pattern b (with dashed line) as given in the figure. For the purpose of expressing the

Boolean function let us arrange the 4 tuples of numbers in the block of an array form as follows

for block a:
$$\begin{array}{cccc} 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{array}$$

and for block b:
$$\begin{array}{cccc} 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{array}$$

Observe that in the block under consideration the columns consist of 1s and 0s. In any column there may be 3 cases:

(1) If the number of 1s is greater than one of 0s then the corresponding variable in the tuple remains;

(2) If the number of 1s is equal to one of the 0s then the corresponding variable is deleted (or absent);

(3) If the number of 1s is smaller than one of the 0s then the negation of the corresponding variable in the tuple holds.

By means of this procedure the result of block a is x_1x_3 and the one of the block b is $\bar{x}_2x_3x_4$. Thus, the Boolean function to be found is

$$f(x_1x_2x_3x_4) = x_1x_3 + \bar{x}_2x_3x_4 .$$

The truth of the procedure described above is easily seen if we remember the operations holding in the set theory.

Example 2. Given a Boolean function as follows

$$f(x_1x_2x_3x_4) = x_1x_2x_3 + x_2x_3\bar{x}_4 + x_1\bar{x}_2\bar{x}_3x_4 .$$

What is the corresponding Karnaugh map like?

Solution. From the given function the smallest dimension Karnaugh map is one of four-variables. Let us apply the 4-tuple order as used above, i. e., $x_1x_2x_3x_4$. The Boolean function consists of three blocks (with the weight of 1).

The procedure is carried out step by step with the so-called Karnaugh "frame" represented in Fig. 5.

First, let us deal with the first block $x_1x_2x_3$.

Step 1 is to choose a cell of 4-tuples that contains the term in fact(i. e. the $x_1x_2x_3$). There are two such cells, namely, the one 1 1 1 0 ($\Leftrightarrow x_1x_2x_3\bar{x}_4$) and the one 1 1 1 1 ($\Leftrightarrow x_1x_2x_3x_4$). Let us select cell 1 1 1 1.

Step 2: addition one of its adjacent cells (because of cell 1 1 1 1 being an internal cell, it therefore has four "normal" adjacent cells). The left adjacent cell is necessarily disregarded because tuple 0 1 1 1 would delete the variable x_1. The lower adjacent cell and the right one is also disregarded because their connection will delete x_2 or x_3 respectively from the block in question. Remains thus the only one, the upper adjacent cell, in which 1 1 1 0 exists. We see that an array of block

$$\begin{matrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{matrix}$$

corresponds to the 3-tuple $x_1x_2x_3$.

Note that if the "addition" is false we must repeat Step 2 but for adjacent cells of cell 1 1 1 1 such that these cells, together with cell 1 1 1 1 itself, have to construct a square or a rectangle.

By means of the similar algorithm we receive the corresponding Karnaugh map of block $x_2x_3\bar{x}_4$ without difficulty (assigned by a horizontal solid pattern in Fig. 5).

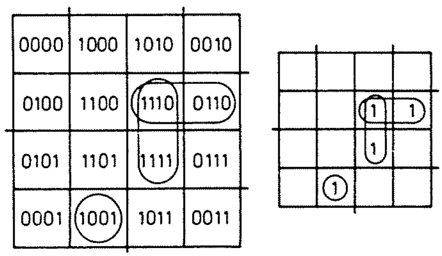


Fig. 5

Block $x_1\bar{x}_2\bar{x}_3x_4$ corresponds to a single cell of 1 0 0 1 which is assigned in the figure by a circle pattern.

With this the problem is solved.

Note that in the Karnaugh diagram, as for n-tuple in number of set $\{0, 1\}$, there are no two similar cells. For the sake of brevity of assignment instead of binary number in very cell let us use the decimal number equivalent to that (if we consider the n-tuples in the number of set $\{0, 1\}$ as the tuples in a binary system). For example 1 0 0 1$\Leftrightarrow$9 (see Fig. 6a).

From the so-called examples we may construct the corresponding algorithm for two problems, one of which is the reverse of the other.

A very interesting feature is that the adjacency of any cell is independent of the arrangement of variables. So, for instance, in the 4-variable maps with the equivalent assignment described above, the adjacent cells of cell 5 always are 4, 1, 7 and 13. Similarly, cell 2 is adjacent to cells 3, 10, 0, 6, and so on (see Fig. 6). Experience suggest that one has to perform the finding procedure in direction such that the give variable always remains in the actual term derived from the preceding result. Furthermore, the cells obtained must form any rectangle or square!

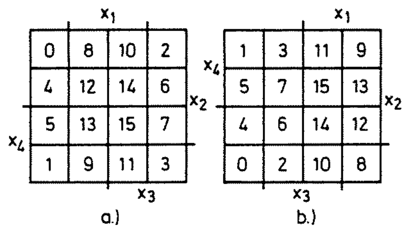


Fig. 6

3.2. *Extraction of prime implicants and prime implicates*

It is well known that when investigating multistate system reliability one of the effectual methods is based on the multiple value discrete function theory. However, one of the ways which gives a discrete function is to find an expression as a disjunction of all its prime implicants or as a conjunction of all its prime implicates. These expressions of the discrete function are not only simpler than other types of lattice expressions but they constitute also canonical forms for discrete functions since they are unique for a given function [3]. By means of Karnaugh map of two-dimensional discrete functions prime implicants and prime implicates may be derived. For the sake of obtaining the adjacency of Karnaugh maps, here we will only deal with the method based on finding the prime blocks and prime convex blocks, because this is the simpler way.

The pattern corresponding to a block of weight l is again a square or a rectangle, grouping entries larger than or equal to l but these rectangles have an additional property: when dealing with blocks it is assumed that the variables are arranged in ring patterns of symmetry so that these entries in a Karnaugh map would be adjacent if the map were drawn on torus. Again the prime blocks correspond to the largest rectangles.

In order to detect prime implicants the following algorithm may be used [3]:

(1) Start with the largest possible pattern, i. e. $S^* = S_1 \times S_0$ and with, for example, the largest weight.

(2) By deleting the smallest possible number of rows and/or columns in the pattern S^* try to obtain a pattern S^{**} such that each of its entries has a weight l; (l, S^*) is then a prime implicant of the function and the patterns smaller than S^{**} are not to be tested for implicants of weight.

(3) Perform the step (2) but by deleting in another manner the rows and/or columns of S^*; explore in this way all the possible patterns and determine all the prime implicants of weight l.

(4) Once the smallest patterns (which are eventually the entries of matrix $S_1 \times S_0$) have been explored, perform the same operation (2) and (3) but for the new weight $l=$(old weight—1).

(5) The algorithm ends once the patterns of weight $l=1$ has been explored.

Example 3. Find prime implicants of function

$$f:\{0, 1, 2, 3, 4\}^2 \to \{0, 1, 2, 3, 4\}$$

represented by the Karnaugh map shown in Fig. 7.

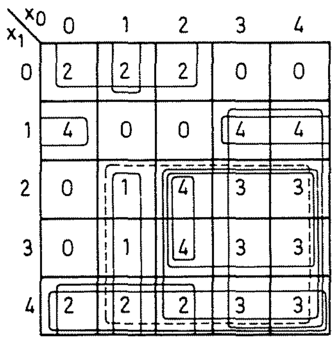


Fig. 7

Solution. Perform the algorithm described above. In our case namely, the largest weight is $l=4$. Therefore, we obtain the prime blocks as follows:

For the weight $l=4$:

$$4x_0^{[3,0]}x_1^{(1)} \quad \text{and} \quad 4x_0^{(2)}x_1^{(2,3)}$$

For the weight $l=3$:

$$3x_0^{(3,4)}x_1^{[1,4]} \quad \text{and} \quad 3x_0^{(2,3)}x_1^{[2,4]}$$

For the weight $l=2$:

$$2x_0^{[0,2]}x_1^{[4,0]} \quad \text{and} \quad 2x_0^{[2,4]}x_1^{[2,4]}$$

For the weight $l=1$:

$$1x_0^{(1)}x_1^{[2,0]} \quad \text{and} \quad 1x_0^{[1,4]}x_1^{[2,4]}$$

From these the given discrete function in a lattice expression as a disjunction of all its prime implicants is

$$f(x_0, x_1)=4x_0^{[3,0]}x_1^{(1)} \vee 4x_0^{(2)}x_1^{(2,3)} \vee 3x_0^{(3,4)}x_1^{[1,4]} \vee 3x_0^{(2,3)}x_1^{[2,4]} \vee$$
$$\vee\, 2x_0^{[0,2]}x_1^{[4,0]} \vee 2x_0^{[2,4]}x_1^{[2,4]} \vee x_0^{(1)} \vee x_1^{[2,0]} \vee$$
$$\vee\, 1x_0^{[1,4]}x_1^{[2,4]}.$$

One may check the rightness of this result by concrete values of x_0 and x_1 noting that the assignment of lattice operation was deleted from expression of blocks, i. e.,

$1x_i^{[\cdot]}x_k^{[\cdot]} \Leftrightarrow 1 \wedge x_i^{[\cdot]} \wedge x_k^{[\cdot]}$, $i \neq k$, i and k are positive integer. For example, if $x_0 = 2$, $x_1 = 2$ then

$$f(2,2) = 4 \quad 0 \quad 0 \vee 4 \quad 4 \quad 4 \vee 3 \quad 0 \quad 0 \vee 3 \quad 4 \quad 4 \vee 2 \quad 4 \quad 0 \vee 2 \quad 4 \quad 4$$
$$\vee 1 \quad 0 \quad 4 \vee 1 \quad 4 \quad 4 = 4 \vee 3 \vee 2 \vee 1 = 4$$

The result thus is true (see Fig. 7).

If $x_0 = 1$, $x_1 = 3$, the result must be 1. Let us check!

$$f(1,3) = 4 \quad 0 \quad 0 \vee 4 \quad 0 \quad 4 \vee 3 \quad 0 \quad 4 \vee 3 \quad 0 \quad 4 \vee 2 \quad 4 \quad 0 \vee 2 \quad 0 \quad 4$$
$$\vee 1 \quad 4 \quad 4 \vee 1 \quad 4 \quad 4 = 1 \vee 1 = 1,$$

therefore our statement is true!

Remember that from the normal form of function in fact all its implicants are perceived.

As for the inversion problem in multiple value functions (i. e., to find the Karnaugh map from a given function in the form of prime implicant disjunctions) difficulties crop up from time to time we are not sure whether the given function is in a normal form or not. To answer this problem we may use a check-up algorithm as follows:

(1) Let us start with the prime of greatest weight. From the lattice exponentiations we know what cells contain the (actual) value of l, not forgeting to order the limit elements in an exponential set of power (formed by consecutive numbers in the ring of integers modulo $l_{max} + 1$).

(2) Continue the procedure described above with the new weight $l =$ (old weight—1) for the purpose of realizing cells of corresponding squares and/or rectangles. If any cell here is occupied, disregard it!

(3) The procedure stops when $l = 0$.

(4) Check if in the obtained Karnaugh map of the given function there are any more prime blocks or false blocks which cause the equivocal fact. If the answer is "NO" then the function in question really has a prime implicant disjunction form. Otherwise, as for the normal form, the statement is false.

In an other way we may check the truth of the mentioned statement by means of finding the contradiction in the concrete value of any cell, and calculate the value of the given function in every cell. If there is no contradiction in either cell then the function is given in a normal form. If there exists a cell in the Karnaugh map which has two or more distinct function values then the statement is false.

Note that this check-up method is very favourable for computer use, but it may be time-consuming. Of course, the check-up procedure is stopped at the first contradiction and one may draw the necessary conclusions!

Example 4. There is a function

$$f:\{0, 1, 2, 3\}^2 \rightarrow \{0, 1, 2\}$$

given in the lattice expression form as follows

$$f(x_0, x_1) = 2x_0^{[0,2]}x_1^{(3)} \vee 2x_0^{[3,1]}x_1^{[0,1]} \vee 1x_0^{(2)}x_1^{(2,3)} \vee 1x_0^{[2,0]}x_1^{(2,3)} \vee 1x_0^{[0,1]}x_1^{(3)}.$$

Is the function in a disjunctive prime implicant form?

Solution. Draw a square consisting of 4^2 cells as shown in Fig. 8. Draw the corresponding pattern of the block of weight $l=2$, not forgetting to write the magnitude of the actual weight to the corresponding cell (or cells), at present: 2. In Fig. 8a the pattern a corresponds to the first block. Similarly we can get the remaining patterns for the remaining blocks 2, 3, 4 and 5 respectively.

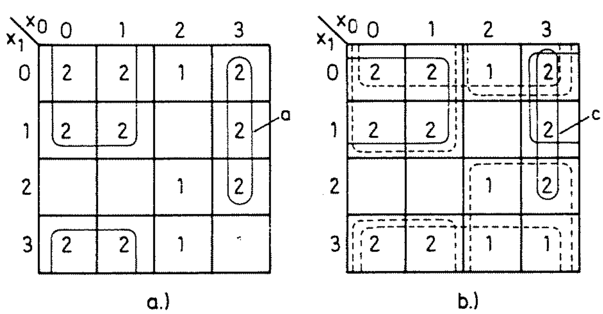


Fig. 8

One can perceive that the given function has only one "good" prime block as shown in Fig. 8b (see pattern c drawn by a solid line).

After all the given function has no a one-to-one correspondance mapping $\{0, 1, 2, 3\}$ into $\{0, 1, 2\}$! It means that this function has no a normal form.

As for finding the prime implicates of any discrete function we may use the theorem as follows [3]

THEOREM. The prime implicates (resp. prime implicants) of discrete function f are the negation of the prime implicants (resp. prime implicates) of $\bar{f}$.

One may perform $\bar{f}$ in Karnaugh map of f with the rule that

"The new value" = "The maximum value" − "The old value"

For instance see Figs 9a, b, where Fig. 9a is the Karnaugh map of f and Fig. 9b is the one of $\bar{f}$.

$f(x_0, x_1)$

$x_1 \backslash x_0$	0	1	2	3	4
0	0	1	2	4	5
1	1	0	3	5	4
2	1	2	4	0	4
3	0	3	0	2	2
4	2	1	1	3	0

a)

$\bar{f}(x_0, x_1)$

$x_1 \backslash x_0$	0	1	2	3	4
0	5	4	3	1	0
1	4	5	2	0	1
2	4	3	1	5	1
3	5	2	5	3	3
4	3	4	4	2	5

b)

Fig. 9

4. Axonometrical representation of Karnaugh maps and the adjacency problem

In this section we consider a possibility of the representation of Karnaugh map and deal with the adjacency problem in a new situation.

4.1. For binary state systems

It is well known that in a binary state system with planar representation by means of Karnaugh map one may handle the Boolean function with a variable number more than 2. Furthermore, Karnaugh map of $(n+1)$-dimension can always be represented by two corresponding Karnaugh maps of n-dimension (see point 2.1).

Now, besides a planar representation we represent the same function by an axonometrical map extending step by step. Thus let us first, consider the case of 3-variables, then, the case of 4-variables and finally that of 5-variables. Of course, this procedure may be continued with a higher number of variables too.

For the sake of brevity we always place the corresponding axonometrical representation beside the planar one and give some remarks for a better understanding.

Fig. 10 shows the case of 3-variables with concrete values of function. In principle the cubes are 3-dimensional, but corresponding to the displacement

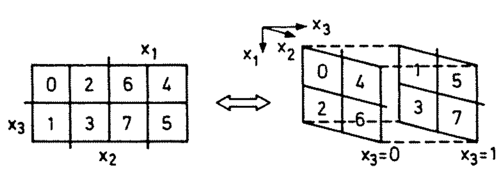


Fig. 10

of the function values in Karnaugh map there may exist a certain number of degenerated cubes and their dimension is 2.

Fig. 11 and Fig. 12 show the case of 4- and 5-variables, respectively.

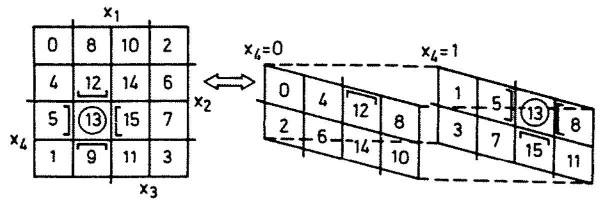


Fig. 11

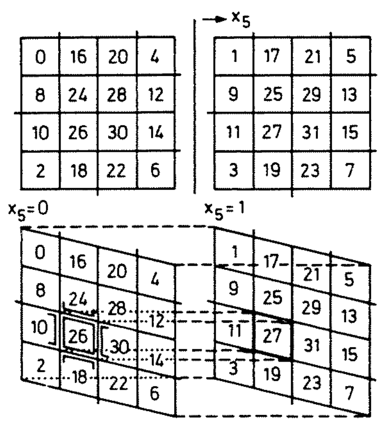


Fig. 12

As for the adjacency problem one can state that in an n-dimension case:

(1) Any cell has n adjacent cells (where n is the number of variables).

(2) If in the plane of a certain cell one found that the number of its adjacent cells is less than n then the absent cells must necessarily be found in the other corresponding planes.

(3) In principle any edge of any cell is a cut between two regions so that moving from the one to the other the value of any variable is changed. Therefore, at the higher order map by means of the concept based on the "torus idea" one may refer to concentric tori (having concentric creative circles) and furthermore, to hyper-concentric tori which, from the view-point adjacency, may be taken as a sort of Riemann surface. This Riemann surface has similar cuts as the ones of Karnaugh map. This concept gives a tool to find the adjacent cells in higher order map. For example, see Fig. 13, adjacent

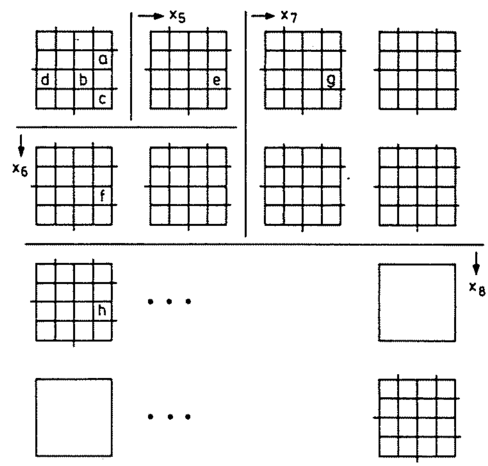


Fig. 13

cells of the shaded cell

—in 4-dimension are cells *a*, *b*, *c* and *d*,
—in 5-dimension are cells *a*, *b*, *c*, *d* and *e*,
—in 6-dimension are cells *a*, *b*, *c*, *d*, *e* and *f*,
—in 7-dimension are cells *a*, *b*, *c*, *d*, *e*, *f* and *g*,
—in 8-dimension are cells *a*, *b*, *c*, *d*, *e*, *f*, *g* and *h*

. . ., and so on.

In general, the Hamming distance between any cell and any of its adjacent cell is 1 when cells are designated with a string of binary numbers as in Sec. 3.

PROPOSITION. The Hamming distance between two adjacent cells is 1 in the Karnaugh map representation. In other words, two cells may be adjacent if between which the Hamming distance is 1.

Proof. The proof is based on two facts:

(1) In the Karnaugh map representation there are no two cells having a similar n-tuple of binary numbers.

(2) Because of the nature of the representation method itself, running from one cell to another cell along a horizontal or a vertical direction, only one letter of the mentioned n-tuple changes.

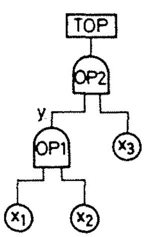
4.2. For multistate systems

In the domain of multistate systems, a planar representation of Karnaugh map may be used only for two variable functions. However, by means of the axonometrical representation one can extend this method for the extraction of prime implicants (or prime implicates) in case of three variables [6]. It must be stated that this extension is certainly sufficient, because by the maximum rectangular covering performed in Sec. 3 combined with a corresponding step by step licitation we may handle a higher multiple value discrete function. Furthermore, for the same purpose one can use the decomposition method because with more complex systems this ability has a great probability.

We give a short summary of the axonometrical method, then, by means of an illustrative example the problem becomes more clear as seen in the following. We would like to emphasize that the concept of maximum rectangular covering always holds in prime implicant detecting. Adjacent property is lost when the number of variables is more than two in planar representation. Axonometry however helps us to ameliorate the use of Karnaugh map to some extent. Furthermore, by means of axonometry one may pictorially explain (with the full induction) some other representation methods of discrete functions whose variable number, in principle, is not limited, for instance the NELSON algorithm [4] and the generalized QUINE method of consensus [5]. These methods are also based on the maximum rectangular covering idea!

Note that if the number of variables is 3, Karnaugh map remains the best method as we may see in a concrete example. It shows how to perform the correspondance between the planar representation and the axonometrical one. The result is received as two equivalent representation types.

Example 5. Let us describe the discrete function determined by the given structure and value table shown in Fig. 14.



OP 1:

$x_2 \backslash x_1$	0	1	2
0	0	0	1
1	0	1	2
2	1	2	2

OP 2:

$y \backslash x_3$	0	1	2	3
0	0	0	0	1
1	1	1	2	2
2	1	2	2	2

Fig. 14

Solution. There are two operations ***OP1*** and ***OP2*** on three variables x_1, x_2 and x_3, namely, ***OP1*** holds on x_1 and x_2, and ***OP2*** holds on (x_1***OP1***x_2) and x_3.

First, let us draw axonometrically n corresponding square maps having $(\# x_1) \times (\# x_2)$ cells, where $n = \# x_3$, the number of states of variable x_3 (see Fig. 13b and imagine the cells to be empty or numberless).

The numbering of the cells, or rather, how to give the function value to the cells, may be performed as follows.

See the value table of ***OP2***, step by step from $x_3=0$ to $x_3=3$ via $x_3=1$ and $x_3=2$. Concretely, at $x_3=0$ the function value equals to 0 when product $(x_1, x_2)=(x_1\mathrm{OP1}x_2)=0$ and equals to 1 when product $(x_1, x_2)=1$ or $(x_1, x_2)=2$. Therefore, the planar cells or value 0 correspond to axonometrical cells of value 0, the planar cells of value 1 or value 2 correspond to the axonometrical cells of value 1. The result may be seen in the "axonometrical value table" beside $x_3=0$.

For the sake of a better practice of this procedure let us also see the case of $x_3=1$. One may see that when $x_3=1$ function $f(x_1, x_2, x_3)$ gives—following each other—value 0, 1 and 2, corresponding to $(x_1, x_2)=0$, 1 and 2 respectively. Thus, every planar cell of value 0 gives the mapped cells of value 0; the planar cells of value 1 give the mapped cells of value 1 and, finally, the planar cells of value 2 give the mapped cells of value 2. The result may be seen in the mapped plane of $x_3=1$ in Fig. 15.

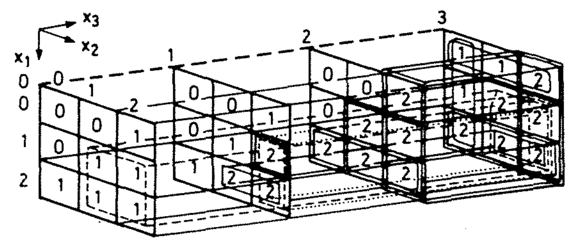


Fig. 15

Continue this procedure until $x_3=3$, and the Karnaugh map (in axonometry) of the given function will be completed.

After receiving the axonometrical Karnaugh map, also by means of the maximum rectangular covering principle, the normal form of

$$f(x_1, x_2, x_3)=2x_1^{(1,2)}x_2^{(1,2)}x_3^{(2,3)} \vee 2x_1^{(2)}x_3^{(2,3)} \vee 2x_2^{(2)}x_3^{(2,3)}$$
$$\vee\, 2x_1^{(1,2)}x_2^{(2)}x_3^{[1,3]} \vee 2x_1^{(2)}x_2^{(1,2)}x_3^{[1,3]}$$
$$\vee\, 1x_1^{(1,2)}x_2^{(1,2)} \vee 1x_1^{(2)} \vee 1x_2^{(2)} \vee 1x_1^{(0)}x_3^{(3)} \vee 1x_2^{(0)}x_3^{(3)}$$

from which prime implicants may be extracted without difficulty.

Recall that in axonometrical planes the magnitude of the number placed in the cells is the value of the function!

In a certain sense one may consider the axonometrical map as a sort of decomposition of the given function.

As for the truth of the received function, for control, let us calculate its value, for example, at some value of its variables as follows

$$x_1=1, \quad x_2=2, \quad x_3=1$$

$$f(1,2,1)=2\ \ 2\ \ 2\ \ 0\vee 2\ \ 0\ \ 0\vee 2\ \ 2\ \ 0\vee 2\ \ 2\ \ 2\ \ 2\vee 2\ \ 0\ \ 2\ \ 2$$
$$\vee 1\ \ 2\ \ 2\vee 1\ \ 0\vee 1\ \ 2\vee 1\ \ 0\ \ 0\vee 1\ \ 0\ \ 0$$
$$=0\vee 0\vee 0\vee 2\vee 0\vee 1\vee 0\vee 1\vee 0\vee 0=2\vee 1\vee 1=2\,.$$

It is true (see the thickly drawn mapped cell in plane of $x_3=1$).

From the axonometrical Karnaugh map one may easily extract the minimal upper vectors (or maximum lower vectors) of the system described by the given function for every corresponding level. For example, in the present case the minimal upper vectors of the system at level 1 and level 2 are:

minimal upper vectors for level 1

		x_1	x_2	x_3
vector	1	0	0	3
	2	1	1	0
	3	0	2	0
	4	2	0	0

minimal upper vectors for level 2

		x_1	x_2	x_3
vector	1	2	2	1
	2	0	1	1
	3	1	0	1
	4	0	2	2
	5	1	1	2
	6	2	0	2

After receiving the minimal upper vectors probability calculation of top event (e. g. system failure) is continued but not dealt here because of lack of space.

Conclusion and discussion

Following the aim mentioned in the introduction the engineering practice of Karnaugh map is performed.

In the sense of lattice theory between Karnaugh map and the normal form of any discrete function there is a one-to-one correspondance. Thus, some reversion problems are dealt with and, in this respect, some algorithms are presented.

By convenient definitions and assignments the properties of adjacency were more clearly explained, helping engineering applications. Furthermore axonometrical representation, combined with the principle of maximum rectangular covering, also simplifies the engineering use, mainly in the field of both reliability evaluation of multistate systems and logic circuit designs.

In reliability orientation extraction of prime implicants and prime implicates is very important. Indeed, by means of prime implicants (or prime implicates) of functions describing the multistate system one may determine the minimal upper vectors (or maximum lower vectors) which play an important role of min pathset and min cutset determination. Recall that min pathset and min cutset are the key problems of system reliability evaluations.

Acknowledgment

My thanks are due to Prof. Olivér Petrik, head od Department of Precision Mechanics and Applied Optics (BME) for the valuable time he spent to read and discuss this paper.

Appendix

1) Symbols

$\vee$	= disjunction, a binary operation in lattice theory.
$\wedge$	= or not symbol = conjunction, a binary op. in lattice theory
$\cup$	= union of sets
$\cap$	= intersection of sets
$\setminus$	= substraction of set
$\subseteq$	= inclusion of a set in another set
$\in$	= is (are) element (s) of
$\#$	= number of elements (cardinality of a finite set)
$\Rightarrow$	= implication
$\Leftrightarrow$	= equivalence
$\bar{a}$	= negation of a
$\oplus$	= addition modulo an integer
$\times$	= operation in Cartesian product
$\forall$	= for every ...
$\exists$	= it exists
$\overline{1, n}$	$= 1, 2, \ldots, n$
$[a, b]$	$= (a, a\oplus 1, a\oplus 2, \ldots, a\oplus m = b), b\oplus 1 = a$
$[c, d]$	$= (c, c\oplus 1, \ldots, d), a < c < d < b$
$[d, c]$	$= (d, d\oplus 1, \ldots, b, a, a\oplus 1, \ldots, c)$
$\bigtimes_{i=1}^{n}$	= multiple operation in Cartesian product ($n = 1, n$)
$\bigvee_{i=n-1}^{0}$	= disjunction of n elements (or terms or sets)
$\bigwedge_{i=n-1}^{0}$	= conjunction of n elements (or terms or sets)

2) Nomenclature, theorems, lemmas and notes

Sup-semilattice: A sup-semilattice is an ordered set S in which every pair of elements admits a supremum (in sets of finite cardinality $\sup\{.\} \Leftrightarrow \max\{.\}$); S is a set.

Inf-semilattice: An inf-semilattice is an ordered set $\langle S \leq \rangle$ in which every pair of elements has an infimum (in finite cardinality sets $\inf\{.\} \Leftrightarrow \min\{.\}$)

Lattice operation $\vee$: A binary operation is a lattice operation if it has the three following properties

(a) idempotence: $s \vee s = s, \forall s \in S$

(b) associativity: $s \vee (t \vee v) = (s \vee t) \vee v, \forall s, t, v \in S$

(c) commutativity: $s \vee t = t \vee s, \forall s, t \in S$

Theorem 1 A set S is sup-semilattice if one can define a lattice operation $\vee$ on S.

Theorem 2 A set S is inf-semilattice if one can define a lattice operation $\wedge$ on S.

$\vee$*-generating system* G: Let $\langle S, \vee \rangle$ be a sup-semilattice. A subset $G \subset S$ is $\vee$-generating system if every element $s \in S$ can be written in at least one way under the form

$$s = g_1 \vee \ldots \vee g_k$$

where the g_i's are well chosen elements of G.

Cartesian product: Given n sets $A_1, \ldots, A_n$, the Cartesian product

$$A_1 \times A_2 \times \ldots \times A_n = \bigtimes_{i=1}^{n} A_i$$

is the set consisting of the ordered n-tuples $(a_1, \ldots, a_n)$ for every $a_1 \in A_1, \ldots, a_n \in A_n$:

$$\bigtimes_{i=1}^{n} A_i = \{a_1, \ldots, a_n | a_i \in A_i, i = \overline{1, n}\}$$

Set of consecutive in ring $C = \{a, a \oplus 1, \ldots, a \oplus h = b\}$, $h > 0$ C is a set.

Cube function is a discrete function $c(\mathbf{x})$ of the form

$$c(\mathbf{x}) = l \wedge \bigwedge_{i=n-1}^{0} x_i^{(C_i)}, \quad l \in L \qquad (L \text{ is a set})$$

where $C_i \subset S_i$ and l will be called the weight of the cube.

Block function: a block function is a cube function of the form

$$b(\mathbf{x}) = l \wedge \bigwedge_{i=n-1}^{0} x_i^{[a_i, b_i]}, \quad l \in L; \quad a_i, b_i \in S_i \forall i.$$

Convex block function: A block function is a convex block function if $a_i \leq b_i \forall i$.

Lattice exponentiation: Given a lattice $\{L^m, \vee, \wedge, \mathbf{0}, \mathbf{r-1}\}$ where 0 and $(r-1)$ mean m-tuples of 0 and $(r-1)$ respectively, the lattice exponentiation is then defined as follows

$$x_i^{(C_i)} = \begin{matrix} r-1 & \text{if } x_i \in C_i \\ 0 & \text{otherwise} \end{matrix}$$

Anticube function is a discrete function $\mathrm{d}(\mathbf{x})$ of the form

$$\mathrm{d}(\mathbf{x}) = l \vee \bigvee_{i=n-1}^{0} x_i^{(D_i)}$$

where $D_i \subseteq S_i \forall i$; l will be called the weight of the anticube. The anticube $\mathrm{d}(\mathbf{x})$ takes thus the value $r-1$ if $\exists i: x_i \in D_i$ and takes the value l otherwise.

Antiblock function is an anticube function of the form

$$a(\mathbf{x}) = l \vee \bigvee_{i=n-1}^{0} x_i^{[a_i, b_i]}, \quad l \in L; \quad a_i, b_i \in S_i \forall i$$

Convex antiblock function: An antiblock function is a convex antiblock function if either $a_1 > b_i$ or $a_i = 0$ or $a_i = m_i - 1$, $\forall i$.

Hamming distance: Let $\mathbf{x} = \langle x_1, x_2, \ldots, x_n \rangle$ and $\mathbf{y} = \langle y_1, y_2, \ldots, y_n \rangle$ be n-tuples representing messages $x_1 x_2 \ldots x_n$ and $y_1 y_2 \ldots y_n$ respectively, where $x_i, y_i \in \{0, 1\}$ for all i. The Hamming distance between $\mathbf{x}$ and $\mathbf{y}$, denoted by $H(\mathbf{x}, \mathbf{y})$ is the number of coordinates for which all x_i and y_i are different.

Higher ordered map is a map of functions having a higher number of variables.

Implicant, implicate: Given a lattice $\langle L, \vee, \wedge \rangle$ and a $\vee$-generating system G of L, any element g of G that is smaller than or equals to some element l of L is called an implicant of l. In other words at present if $g(\mathbf{x}) \leq f(\mathbf{x})$, i. e. $g(\mathbf{x}) \geq k$ implies $f(\mathbf{x}) > k$ for all $\mathbf{x}$, then $g(\mathbf{x})$ is an implicant of $f(\mathbf{x})$.

If $g(\mathbf{x}) \geq f(\mathbf{x})$, i. e. $g(\mathbf{x}) \geq k$ implies $f(\mathbf{x}) \geq k$ for all $\mathbf{x}$ then $g(\mathbf{x})$ is an implicate of $f(\mathbf{x})$.

Prime implicant, prime implicate:

—The maximal implicant $g(\mathbf{x})$ of G is prime implicant.

—The minimal implicate $g(\mathbf{x})$ of G is a prime implicate.

Karnaugh's frame is a diagram which consists of a number of square and/or rectangles, in a planar or axonometrical representation, building from their cells in a way that in any cell one may assign a string in respecting to certain entries of the corresponding system.

Upper vector for level j: Given a monotone increasing function $f(\mathbf{x})$, if a state vector $\mathbf{a}$ satisfies that $f(\mathbf{a}) \geq j$, then $\mathbf{a}$ is an upper vector for level j.

Minimal upper vector for level j: If an upper vector for level j, $\mathbf{a}$, satisfies that for any $\mathbf{a}' > \mathbf{a}, f(\mathbf{a}') < j$, then $\mathbf{a}$ is a minimal upper vector for level j.

(Minimal upper vectors for level j, $j = \overline{0, m}$, play a role of min pathset).

Lower vector for level j: Given a monotone increasing function $f(\mathbf{x})$, if state vector $\mathbf{b}$ satisfies that $f(\mathbf{b}) \leq j$ then $\mathbf{b}$ is a lower vector for level j.

Maximum lower vector for level j: If a lower vector for level j, say $\mathbf{b}$, satisfies that for any $\mathbf{b}' > \mathbf{b}, f(\mathbf{b}) > j$, then $\mathbf{b}$ is a maximum lower vector for level j.

(Maximum lower vector for level j, $j = \overline{1, m}$, play a role of min cutset)

Lemma 1 The cubes, blocks and convex blocks smaller than a discrete function f are implicants of f. These implicants will be called cubes of f, blocks of f and convex blocks of f respectively.

Lemma 2 The anticubes, antiblocks and convex antiblocks greater than a discrete function f are implicates of f. These implicates will be called anticubes of f, antiblocks of f and convex antiblock of f respectively.

Theorem 3

(a) Any discrete function f is the disjunction of all its prime cubes.

(b) Any discrete function f is the disjunction of all its prime blocks.

(c) Any discrete function f is the disjunction of all its prime convex blocks.

Theorem 4

(a) Any discrete function is the conjunction of all its prime anticubes.

(b) Any discrete function is the conjunction of all its prime antiblocks.

(c) Any discrete function is the conjunction of all its prime convex antiblocks.

Note 1: Conjunction of two cubes. Given two cubes

$$1_0 \wedge \bigwedge_{i=n-1}^{0} x_i^{(C_{i0})} \quad \text{and} \quad 1_1 \wedge \bigwedge_{i=n-1}^{0} x_i^{(C_{i1})}, \quad 1_0, 1_1 \in L;$$

the conjunction of the two cubes is a cube having the expression as follows

$$1_0 \wedge 1_1 \wedge \bigwedge_{i=n-1}^{0} x_i^{(C_{i0} \cap C_{i1})}$$

Note 2: Disjunction of two anticubes. Given two anticubes

$$1_0 \vee \bigvee_{i=n-1}^{0} x_i^{(D_{i0})} \quad \text{and} \quad 1_1 \vee \bigvee_{i=n-1}^{0} x_i^{(D_{i1})}, \quad 1_0, 1_1 \in L;$$

the disjunction of the two anticubes is an anticube having the expression as follows

$$1_0 \vee 1_1 \vee \bigvee_{i=n-1}^{0} x_i^{(D_{i0} \cup D_{i1})}.$$

Expression form of prime implicants: Prime implicants are product terms of the form

$$\bigwedge_{i=n-1}^{0} x_i^{C_i} \qquad C_i \subseteq S_i \forall i$$

Normal forms of discrete functions: If a discrete function is expressed as the disjunction of cube functions it is called a *disjunctive normal form.* A discrete function expressed as conjunction of anticubes functions it is called a *conjuctive normal form.*

References

1. KARNAUGH, M.: The map method of combinational logic circuits. Trans. AIEE Part I. Communication and electronics, 72, 1953, pp 593—599.
2. TREMBLAY, J. P.–MANOHAR, R.: Discrete mathematical structures with applications to computer science. McGraw-Hill Book Company, 1975.
3. DAVIO, M.–DESCHAMPS, J. P.–THAYSE, A.: Discrete and switching functions. McGraw-Hill International Book Company, 1978
4. NELSON, J. R.: Simplest normal truth functions. J. Symbolic Logic, vol. 20, 1954, pp 105—108.
5. QUINE, W. V.: A way to simplify truth functions. American Math. Monthly, vol. 62, 1955, pp 627—631.
6. GARRIBA, S.–GAGNINI, E.–MUSSIO, P.: Multiple-valued logic trees: Meaning and prime implicants. IEEE Trans. Reliability, vol. R-34. No. 5, 1985 December, pp 463—472.
7. WINFRID, G. S.: Fault-tree analysis using a binary decision tree. IEEE Trans. Reliability, vol. R-34. No. 5, 1985 December, pp 453—457.

Tran VAN DAC H-1521 Budapest