

FAST MINIMIZATION ON THE XIAO MAP USING ROW GROUP STRUCTURE RULES

C.C. Osuagwu, C.D. Anyanwu and J.O. Agada
 Department of Electronic Engineering
 University of Nigeria, Nsukka.

ABSTRACT

Xiao has proposed a new graphical method suitable for the minimisation of logical functions of five or more variable. In this paper, we present a set of rules that simplify minimization using the Xiao map. We also show that the Xiao map technique compares favourable with the Quine -McCluskey algorithmic method.

1. INTRODUCTION

A basic problem in logic design is the minimisation of a given Boolean expression such that the resulting logical function can be implemented using a minimum number of universal gates (example NAND or NOR gates). The objective is to minimise cost and increase reliability by minimizing the gate count. However, the tremendous improvement in integrated circuit technology has made gate cost almost insignificant thereby changing the focus of digital design from gate minimization to package or chip minmisation [1].

Gate level minimization still remains relevant despite the advent of large scale integrated circuit (LSI) and very large scale integrated circuit (VLSI). For example, map entered variable methods are used to reduce the number of multiplexer packages used in the synthesis of logic circuits with multiplelexer [2]. LSI devices like programmable logic arrays (PLA) require minimization of Boolean function if they are to be used efficiently. Gate level minimization is extensively applied in the design of LSI and VLSI circuitry since a saving of one or more gates could translate to substantial savings in silicon real estate. Finally, gate level minimization techniques are irreplaceable vehicles for teaching a systematic approach to logic design. It must be recognized that minimization destroys the regular structure of a logical network and makes it difficult to understand what the system actually does. This is a limitation particularly in the design of large digital systems. The Karnaugh map and the Quine-McCluskey technique [3] are the most widely used gate level minimization procedures. The Karnaugh map exploits the pattern recognition abilities of the designer to achieve fast identification of essential prime implicants. For four variables or less, the Karnaugh map can be used easily and efficiently. However for five or more variables it becomes more difficult to use.

A number of algorithmic techniques, including the

Quine McCluskey method, have been developed for the minimization of functions of five more variables [4]. The Quine McCluskey technique can be computerized. When used manually, the complexity increases rapidly for functions of six or more variables because of the exhaustive search for adjacencies involved in the use of the technique. The technique ultimately guarantees a minimal solution although many redundant terms are generated in the process.

Xiao [5] has proposed a new graphical method for the minimisation of logic functions of five or more variables. In this paper, we evaluate this new technique and present a set of rules which aid minimisation of logic functions using the Xiao map.

2. XIAO MAP

A boolean function consisting of M minterms, where each minterm has n variables is denoted by

$$F(X_1, \dots, X_n) = \sum_i^M K_i \quad (1)$$

This function is mapped into a Xiao map consisting of 2n rows and M columns as shown in figure 1. Each column represents a minterm K_i and the minterms are arranged in the ascending order of magnitude. The map variables (X_1, \dots, X_n) are partition into groups of two variables $X_1X_2, X_3X_4, \dots, X_{n-1} X_n$ starting with the most significant variable. Each group of two variables is represented uniquely on the map by four rows where each row represents a possible binary combination of the two variables. The rows are arranged such that adjacent rows are logically adjacent (00, 01, 11, 10). Thus for n variables there will be 2n rows, We shall refer to each group of four rows as a partition.

The i-th minterm is plotted on the Xiao map by placing dots at the intersection of the i-th column with the rows (one in each partition) that together correspond to the binary representation of

minterm. These intersections are called keypoints. An example will help illustrate the mapping of minterms on a Xiao map.

Consider the four variable function $F(A, B, C, D) = \sum m(6, 7, 13, 11, 14, 9)$ (2)

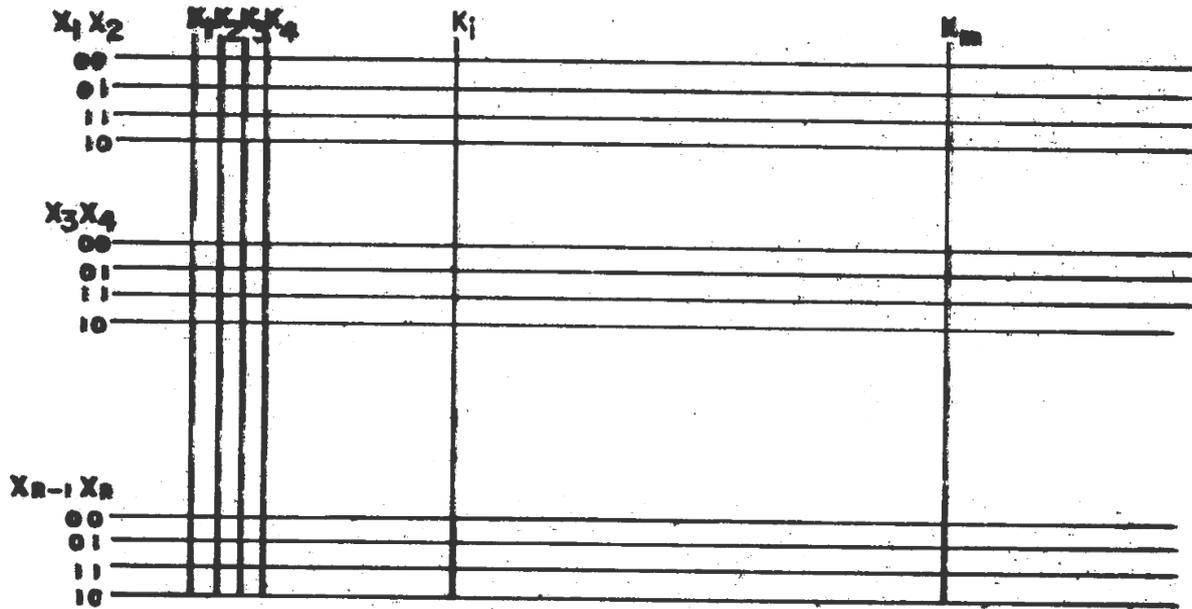


Fig. 1: Structure of a Xiao map.

The number of rows required is $2n$, which is 8 in this case. The eight rows are arranged as two partitions AB and CD. The minterms are arranged in ascending order of magnitude (6, 7, 9, 11, 13, 14) as shown in figure 2. The binary representation of minterm 6 is 0110. In terms of the two partitions, minterm 6 is AB = 01 and CD = 10. To plot this minterm, dots are placed at the intersection of rows AB = 01 and CD = 10 with the first column as shown in figure 2. In the same way, the other minterms are plotted on the Xiao map.

3. MINIMIZATION ON THE XIAO – MAP

All minimisation is based on the concept of adjacency.

$$AB + A\bar{B} = A \tag{3}$$

Therefore, if two minterms are adjacent, a variable (literal) is eliminated. For two minterms to be adjacent on the Xiao map, the following condition must be satisfied simultaneously:

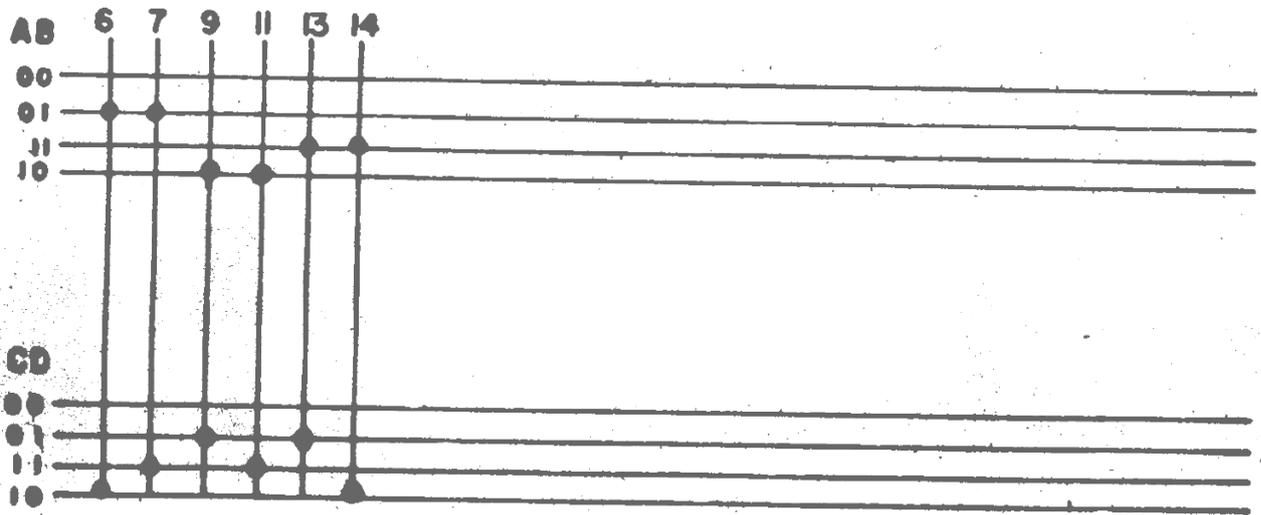


Fig. 2: Xiao map of $F(ABCD) = \sum m(6, 7, 13, 11, 14, 9)$.

- i. Their keypoints must lie on the same row within all the partitions except one.
- ii. In the partition in which the positions of the keypoints differ, the key points must lie on logically adjacent rows.

In the Xiao map of figure 2, for example, minterms 6 and 7 are adjacent because their keypoints lie on the same row in the AB partition while in the CD partition, their keypoints lie on logically adjacent rows (11, 10). Minterms 13 and 14 are not adjacent because in the CD partition in which the position of their keypoint differ, the keypoints do not lie not satisfy the conditions for adjacency because their keypoints differ in both the AB and CD partitions.

When two adjacent minterms in a Xiao map are combined, a variable is eliminated in the partition in which the positions of their keypoints differ. The variable remaining is indicated with a line segment joining the two rows on which the keypoints lie. This line segment is called a keyline.

In figure 3, when minterms 6 and 7 are combined, the variable D is eliminated. This is denoted by the keyline joining rows 11 and 10 in the CD partition resulting in term 1^1 in the first class. The first class consists of terms in which one variable has been eliminated. The 0 – class consists of the minterm list. Unlike the Karnaugh map technique, minimisation on a Xiao map is not a one step procedure. Adjacent terms in the 0 – class combine to form terms in the

second class. (The second class consists of term in which two variables have been eliminated). In general, the procedure is repeated with adjacent terms in the i-th class combined to form terms in the (j + 1)th class. However, since the terms in this class may not cover the minterm list, a final class is formed from those terms in other classes which are needed to cover the minterm list.

3.1 ADJACENCY OF TERMS IN FIRST AND HIGHER CLASSES

Two terms in the first (or higher) class are adjacent if and only if

- i. They differ in only one partition
- ii. In the partition in which they differ
 - a. Their keylines are adjacent or
 - b. Their keypoints are adjacent.

Adjacent keyline eliminate the same variable in a partition. For example, the keylines in columns 1 and 2 of figure 4 (a) are adjacent because they both eliminate the variable A and when combined, eliminate variable B i. e. $B + \bar{B} = 1$.

Similarly, the keylines in figure 4(b) are adjacent and when combined, eliminate the variable A. Clearly then, the combination of two adjacent keyline results in the elimination of two variables.

3.2 THEOREM

If two terms have a keypoint and a keyline in the same partition, they cannot be adjacent.

Proof:

The presence of a keypoint and a keyline in one partition shows that the two terms have

different variables eliminated hence they cannot be adjacent.

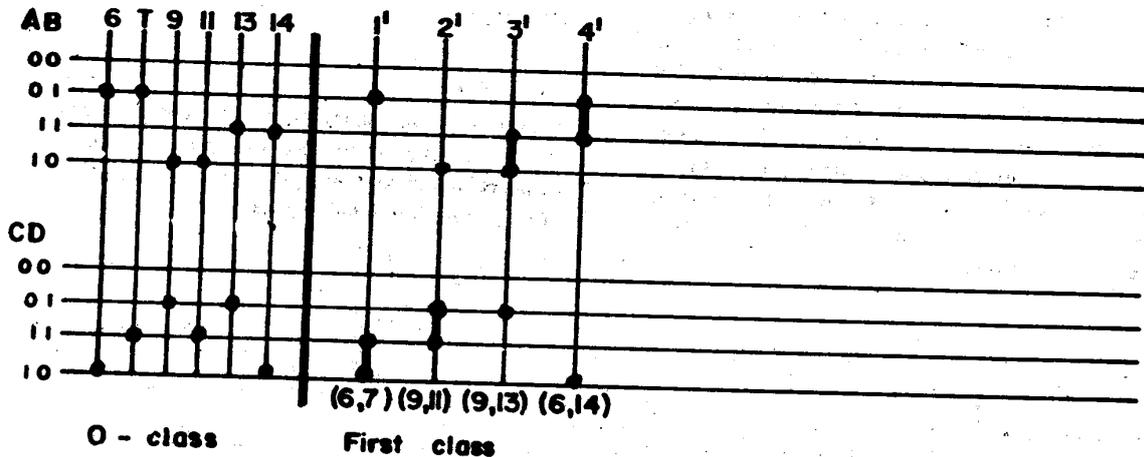


Fig. 3: Generation of first class terms from 0-class minterms

3.3- MINIMISATION PROCEDURE

Minimisation of a logical function on the Xiao map uses the following three steps:

STEP1: Generate all possible classes from the minterm list.

STEP 2: Eliminate any redundant term using the frequency of occurrence of minterms as criteria. (Xiao calls this elimination step emergence of minterms).

STEP 3: Obtain the reduced logical expression from the Xiao map

We shall now apply the minimisation procedure to the following minterms lists:

$$F_1(A,B,C,D,E,F) = \sum M(0,2,5,6,8,10,14,16,17,18,21,22,24,26,30,34,37,38,42,46,49,50,53,54,58,62)$$

(4)

and

$$F_2(A,B,C,D,E) = \sum m(0,6,8,10,12,14,17,19,20,22,25,27,28,30).$$

In applying the procedure, we shall proposed rules designed to enhance the ease of use of the xiao map in minimization of logical functions.

4. FAST MINIMIZATION USING ROW GROUP STRUCTURE RULES

Figures 5 and 6 show the Xiao map (STEP 1 of procedure) for the minimization of F1 and F2 using the adjacency rule and theorems of sections 3.1 and 3.2 respectively and the following additional explanations and rules.

4.1 ROW GROUP STRUCTURE

The partition of the map variables into two or more sets imposes a row group structure on the most significant partition of the 0-class. The most significant partition contains the two most significant variables. Such a row group structure may also be evident in the first or higher classes. For example, in figure 5 minterms (0,2,5,6,8,10,14) form a row group because they all have a keypoint lying on the same row in the most significant partition AB.

Similarly minterms 16 - 30, 34- 46 and 49 -62 form other row groups. Note the existence of a row group structure on the most, class consisting of four groups as follows:

group 1(1¹ - 8¹); group 2(9¹ - 17¹); group 3 (18¹ - 22) and group 4 (23¹ - 27¹).

Using this row group structure, two types of combinations between adjacent terms may be identified: intra-group, where a term is combined with a term in the same row group; and inter-group, where a term is combined with a term from another group It is of interest to formulate rules that exploit this row group structure to achieve fast minimisation.

When a row group structure is evident in the most significant partition of any class exhaustive intra-group combination of all adjacent terms; in the group should be carried out to generate terms in the next (or succeeding) class.

Application of this rule to the 0-class of figure 5 results in the 27 terms of the first class with

the exception of 5^1 and 20^1 , Figure 7 of the appendix shows that exhaustive combination of the 0 – Cubes in the Quine-McCluskey method yields 48 term in the 1st cubes.

Note that, minterms 5, 21, 37 and 53 do not have adjacent intra –group terms. We apply rule 2 in this case.

4.3 RULE 2:

If any term cannot be combined within its row group, then it should be combined with any adjacent out – of – group term if the minterm (s) covered by that term is not covered by second class terms in the succeeding class.

Application of Rule 2

Referring to figure 5 the 1st class terms 5^1 and 20^1 are generated from 0-class terms 5, 21 and 37, 53 respectively since 5 and 37 have no adjacent intra-group terms. The first class term 9^1 is not adjacent to any other first class term but its component minterms are covered by second class terms 4^2 and 5^2 .

4.4. RULE 3:

If a term cannot be combined within or outside its group and the minterm(s) it covers do not appear as components of any term in the next class, a search should be made in the preceding class to generate new terms that can combine with it or can combine with other terms so as to cover the component minterms of the original term.

Application of Rule 3

In minimising the function F_2 (see figure 6), this rule is used to generate the first class term $1^1(22, 30)$ which can then be combined with $2^1(6,14)$ so as to generate the 2nd class term $1^2(2^1,11^1) = (6,14,22,30)$ needed to cover minterm 6.

4.5 RULE 4:

If the search of rules is unsuccessful, then the term is a prime implicant.

For Instance 3^2 and 5^2 are prime implicants of F_1 ; so also is 11^1 in figure 6. Prime implicants are indicated in figures 5 and 6 with asterisks.

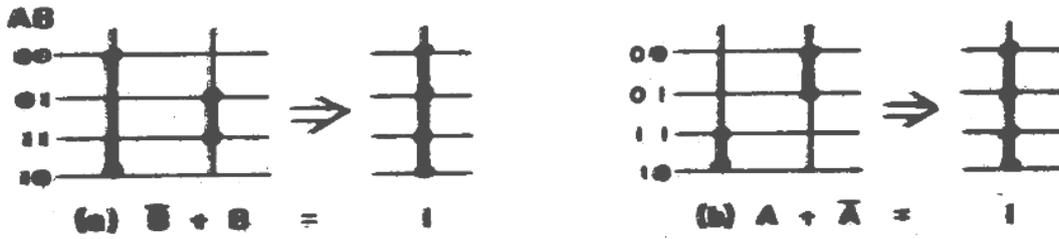


Fig. 4: Adjacent keylines.

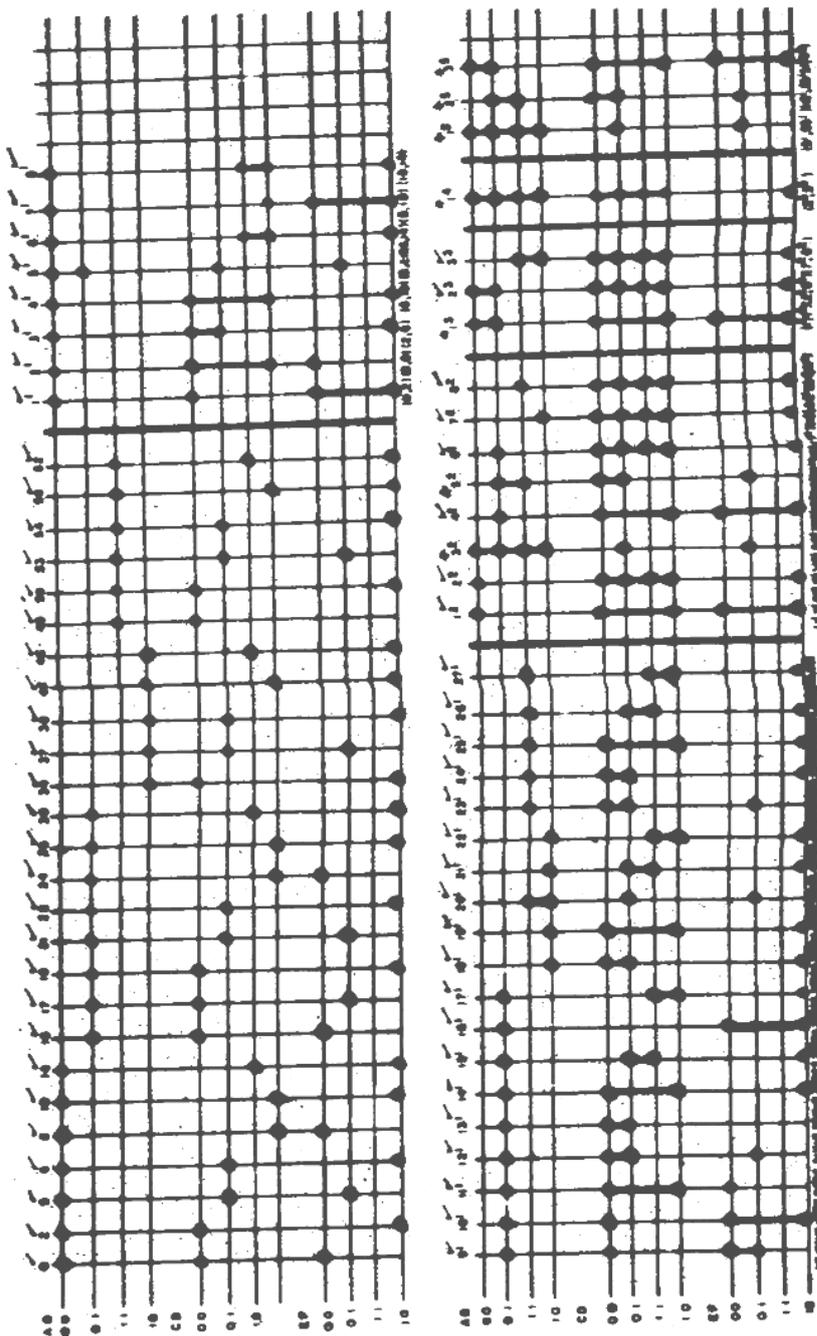
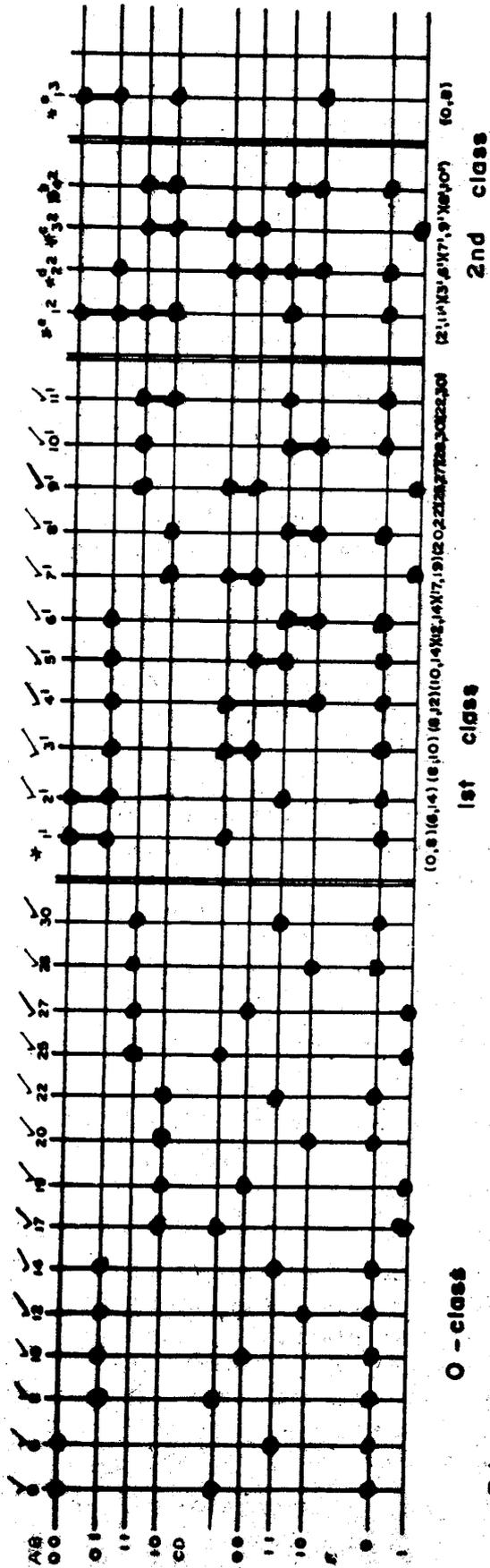


Fig. 5: X100 map minimization of F1



Prime implicants: $a \equiv (0,8)$, $b \equiv (20,22,28,30)$, $c \equiv (17,19,25,27)$, $d \equiv (8,10,12,14)$, $e \equiv (6,14,22,30)$

(a) XIAO map

(b) Emergence table

Minterm	0	6	8	10	12	14	17	19	20	22	25	27	28	30
Emergence	1	1	2	1	1	2	1	1	1	2	1	1	1	2

Essential prime implicants: a, b, c, d, e ,
 $F = \bar{A}\bar{B}E + A\bar{C}E + ACE + CDE + ACDE$

Fig. 6 Minimisation of F2 using a

MINTERM 0 2 5 6 8 10 14 16 11 18 21 22 24 26 30 34 37 38 42 46 49 50 53 54 58 62
 EMERGENCE 1 3 1 2 1 2 1 1 1 4 2 .3 1 3 2 1 11 2 1 1 1 3 2 2

Figure 8: Emergency Table for Function F₁

4.6 RULE 5

If two or more combinations in one class yield the same pattern in the succeeding class, the second and subsequent combination are not drawn on the Xiao map; but terms in the combinations are ticked off as all the minterms they cover (i.e. the minterms which combine to yield these terms) would have been covered by the first combination.

Application of Rule 5

Such a situation arises in the generation of 12 of figure 5 where both (1¹, 7¹) and (2¹, 4¹) lead to the same pattern. It also arises in figure 6 where the combination of (3¹, 6¹) and (4¹, 5¹) lead to the same pattern.

4.7 RULE 6

Where a row group structure cannot be identified, minimal combination of terms of that class that assures cover of the minterm list is carried out.

Application of Rule 6

This rule is applied in figure 5 where for example, in the generation of the 3rd class the combination (6² 8²) is not used. This is because 6² and 8² are already covered by 2³(2², 6²) and 3³ (7² 8²) respectively.

It is clear that the generation of classes terminates after the 4th class for function F₁. Prime implicant 3² 5² and 1² are put into the 5th and final class.

These and 1⁴ form a subset of the prime implicants containing all the essential prime implicants. For F₂, the generation of classes terminates with the 2nd class with the prime implicants (1³, 1², 2², 3², and 4²) being generated.

4.8. STEP2

For the function, F₁ the identified prime implicants are:

- 1⁴ (2, 6, 10, 14, 18, 22, 26, 30, 34, 38, 42, 46, 50, 54, 58, 62)
- 1⁵ (5, 21, 37, 53)
- 2⁵ (17, 21, 49, 53)
- 3⁵ (0, 2, 8, 10, 16, 18, 24, 26).

Using this list, an emergence table is drawn for F₁ as follows

- (i) List all the minterms of the function
- (ii) Under each minterm, write the total prime implicants in which the minterm appears. For instance minterm 53 appears in two prime implicants while minterm 17 occurs in only one.

Figure 8 shows the resulting emergence table for F₁

Prime implicants which contain minterms with only one emergence, that is, they occur only in one prime implicant are essential prime implicants. From figure 8, the prime implicants

minterms 0, 5, 8, 14, 16, 17, 24, 17, 42, 46 and 49 are essential prime implicants. Thus for the function of F₁, all the prime implicants are essential and together ensure a covering of the minterm list.

4.9 STEP 3

The minimised function can be read off from the Xiao map representation of the prime implicant as follows (see figure 5).

To read 1⁴, we observe that the variable in the AB and CD partitions have been eliminated. The keypoint in the E¹F¹ partition corresponds to E¹F¹. Therefore the essential prime implicant 1⁴ = E¹F¹.

To read 2⁵ we note that in the AB partition A has been eliminated and B = 1; in the CD partition, D has been eliminated and C = 0; while in the EF partition, the keypoint corresponds to E¹F¹.

Thus the essential prime implicant 2⁵ = B C E F

Similarly 1⁵ = B C E F and 3⁵ = A D F.

The required minimised function is

$$F_1 = E^1F^1 + A^1D^1F^1 + B^1C^1E^1F^1 + C^1D^1E^1F^1.$$

Similarly the minimised function from the Xiao map of figure 6 is

$$F_2 = A^1B^1E^1 + A^1C^1E^1 + A^1C^1E^1 + C^1D^1E^1 + A^1C^1D^1E^1.$$

In the examples we have used, the essential implicants covered all the minterms of the function. It should be noted that when the prime implicants is made from the non-essential prime implicant to cover the remaining minterms at minimal costs [6].

5 COMPARISON OF XIAO MAP AND QUINE McCLUSKEY METHOD

Though essentially different methods (one graphical, the other algorithmic), some similarities exist between the Xiao map and Quine McCluskey techniques such as:

- (1) group structuring - classes in Xiao map and cubes in Quine McCluskey.
- (ii) Use of a selection mechanism to identify essential prime implicants - emergence table in the Xiao map and prime implicant table in the Quine McCluskey method.

Infact, the Xiao map can be said to be an approximate graphical representation of the Quine McCluskey method. The fundamental difference between the two techniques is in the generation of intermediate product terms. While the Quine McCluskey method requires the formation of all possible product terms (48 1-cube terms in the example of figure 7), the Xiao map uses a reduced set of these terms to cover the function, (27 1st class terms as shown in figure 5). This is a significant reduction. This trend continues in subsequent corresponding sets of product terms,

This reduction in the number of intermediate

product terms makes the Xiao map faster to use than the Quine McCluskey method for functions of two or more variables. The Quine McCluskey method is algorithmic and so can be easily automated whereas the Xiao map technique is essentially heuristic and therefore not easily automated. Consequently, the advantage the Xiao map has over the Quine McCluskey with respect to speed of minimisation and ease of use is valid only in the manual mode of minimisation.

6. CONCLUSION

We have presented an adjacency theorem and a set of rules which exploit the row group structure of the Xiao map to effect fast minimisation of Boolean functions of five or more variables using the Xiao map. These rules make minimisation of the Xiao map more systematic. The Xiao map resembles in many ways, a graphical representation of the Quine McCluskey technique.

Our evaluation is that the Xiao map is a useful new graphical technique for the synthesis of logic circuit using gates. It should serve as a useful complement to the familiar Karnaugh map method

for functions of five or more variables

REFERENCES

1. BLAKESLEE T.R., *Digital Design with Standard MSI & LSI*, John. Wiley, New York 1979, 2nd edition.
2. BENNETH, L.A.M., The application of map-entered variables to the use of multiplexers in the synthesis of logic functions, *International J. of Electronics*. Vol.45, 1978, pp373-379.
3. LEWIN.D., *Logical Design of Switching Circuits* Thomas Nelson, London, 1914, 2nd edition
4. MANO, M.M., *Digital Logic and Computer Design* Prentice-Hall, Inc. New Jersey, 1979.
5. YONG-XIN, XIAO., Xiao. map for minimization of Boolean expressions, *International J. of Electronics*, Vol. 63, 1987; pp,353 – 358
6. Hill, F J., and PETERSON, G.R., *Introduction to Switching Theory and Logical Design* John Wiley, New York, 1968.

APPENDIX

F(A,B,C,D,E,F) = m(0,2,5,11,14,16,17,18,21,22,24,26,30,34,37,38,42,46,49,50,53,54,58,62)

0-Cubes:

- 0
- 2
- 8
- 16
- 5
- 6
- 10
- 17
- 18
- 24
- 34
- 14
- 21
- 22
- 26
- 37
- 38
- 42
- 49
- 50
- 30
- 46
- 53
- 54
- 58
- 62

1-Cubes:

- 0,2 (2)
- 0,8 (8)
- 0, 16 (16)
- 2, 6 (4)
- 2, 10 (8)
- 2, 18 (16)
- 2, 34 (32)
- 8, 10 (2)
- 2, 24 (16)
- (e) 16,17 (1)*
- 16, 18 (2)
- 16, 24 (8)
- 5, 21 (16)
- 5, 37 (32)
- 6, 14 (8)
- 6, 22 (16)
- 6, 38 (32)
- 10, 14 (4)
- 10, 26 (16)
- 10, 42 (32)
- 17, 21 (4)
- 17, 49 (32)
- 18, 22 (4)
- 18, 26 (8)
- 18, 50 (32)
- 24, 26 (2)
- 34, 38 (4)
- 34, 42 (8)
- 34, 50 (16)
- 14, 30 (16)
- 14, 46 (32)
- 14, 46 (32)
- 21, 53 (32)
- 22, 30 (8)
- 22, 54 (32)
- 26, 30 (4)
- 26, 58 (32)
- 37, 53 (16)
- 38, 46 (8)
- 38, 54 (16)
- 42, 46 (4)
- 42, 58 (16)
- 49, 53 (4)
- 50, 54 (4)
- 50, 58 (8)
- 30, 62 (32)
- 46, 62 (16)
- 54, 62 (8)
- 58, 62 (4)

2-Cubes:

- 0,2,8,10 (2,8)
- 0, 8, 16, 24 (8,16)
- 2, 6, 10, 14 (4,8)
- 2, 6, 18, 22 (4,16)
- 2, 6, 34, 38 (4,32)
- 2, 10, 18, 26 (8,16)
- 2, 10, 34, 42 (8,32)
- 2,18,34,50 (16,32)
- 8, 10, 24, 26 (2,16)
- 16, 18, 24, 26 (2,8)
- (d) 5, 21, 37, 53 (16,32)*
- 6, 14, 22, 30 (8,16)
- 6, 14, 38, 46 (8,32)
- 6, 22, 38, 54 (16,32)
- 10, 14, 26, 30 (4,16)
- 10, 14, 42, 46 (4,32)
- 10, 26, 42, 58 (16,32)
- (c) 17, 21, 49, 53 (4,32)*
- 18, 22, 26, 30 (4,8)
- 18, 22, 50, 54 (4,32)
- 18, 26, 50, 58 (8,32)
- 34, 38, 42, 46 (4,8)
- 34, 38, 50, 54 (4,16)
- 34, 38, 42, 46 (4,8)
- 34, 42, 50, 58 (8,16)
- 14, 30, 46, 62 (16,32)
- 22, 30, 54, 62 (8,32)
- 26, 30, 58, 62 (4,32)
- 38, 46, 54, 62 (8,16)
- 50, 54, 58, 62 (4,8)

3-Cubes:

- 0, 2, 8, 10, 16, 18, 24, 26 (2,8,16)*
- (b) 2, 6, 10, 14, 18, 22, 26, 30 (4,8,16)
- 2, 6, 18, 22, 34, 38, 50, 54 (4,16,32)
- 2, 6, 34, 38, 10, 14, 42, 46 (4,32,8)
- 2, 10, 18, 26, 6, 14, 22, 30 (8,16,4)
- 2, 10, 18, 26, 34, 42, 50, 58 (8,16,32)
- 6, 14, 22, 30, 38, 46, 54, 62 (8,16,32)
- 10, 26, 42, 58, 14, 30, 46, 62 (16,32,4)
- 18, 22, 26, 30, 50, 54, 58, 62 (4,8,32)
- 34, 38, 42, 46, 50, 54, 58, 62 (4,8,16)

4-Cubes:

- (a) *2,6,10,14,18,22,26,30,34,38,42,46,50,54,58,62(4,8,16,32)*

	0	2	5	6	8	10	14	16	17	18	21	22	24	26	30	34	37	38	42	46	49	50	53	54	58	62
a		✓		✓		✓	✓			✓		✓		✓	✓	✓		✓	✓	✓		✓		✓	✓	✓
b	✓	✓			✓	✓		✓		✓		✓		✓		✓										
c								✓			✓										✓		✓			
d			✓								✓						✓									
e								✓	✓																	
	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Figure 7: Minimisation using the QUINE- McCLUSKEY method.

The essential prime implicants are therefore

a, b, c, d

a - 2,6,10,14,18,22,26,30,34,38,42,46,50,54,58,62

(4,8,16,32)

A B C D E F

X X X X 1 0

a = $E\bar{F}$

b - 0,2,8,10,16,18,24,26 (2,8,16)

A B C D E F

0 X X 0 X 0

b = $\bar{A} \bar{D} \bar{F}$

c - 17,21,49,53 (4,32)

A B C D E F

X 1 0 X 0 1

c = $B \bar{C} \bar{E} \bar{F}$

d - 5,21,37,53 (16,32)

A B C D E F

X X 0 1 0 1

d = $\bar{C} D \bar{E} F$

and $f(A, B, C, D, E, F) = E\bar{F} + \bar{A} \bar{D} \bar{F} + B\bar{C}\bar{E}\bar{F} + \bar{C} D \bar{E} F$