# Generation of Prime Implicants of A Logic Function by Multibranch Expansion Method

Kimio GOTO*, Masao MENJO***, Takashi ITO**,
Kee-seng CHIN**, Xiao-ping LING*
and Hisayuki TATSUMI*

**Abstract**

For generating the prime implicants, we developed the Multibranch Expansion method. This method divides a given function into several subfunctions by the binary values of several consecutive variables in order to generate a number of prime implicants more speedily than the usual methods. The C language program using this method was run on a SPARC station 5 for the different numbers of consecutive variables and compared with each other in terms of the computation times. As a result, it was found that the case using a larger number of consecutive bits is better than the case using a smaller number of ones.

## 1. Introduction

The generation of prime implicants of a logic function has been studied by authors ([1] through [3]). In the same way as these papers, this paper also deals with the problem of comparing the computation times, especially for a larger number of variables and a number of inputs. The method of this paper, the Multibranch Expansion method uses the following techniques : First, the given original function is divided into the $2^{n_b}$ subfunctions by the binary values of the $n_b$ consecutive variables. Second, other new functions are made by combining the adjacent functions included in these $2^{n_b}$ divided functions. Then, the same procedures of divisions and combinations are repeated at each expansion level until all prime implicants are obtained. This method has a tendency that the case using the long consecutive bits for division is better than the case using the short one in terms of the computation time.

## 2. Definitions, Rules and Theorems

In this section, the definitions, the rules and the theorems used in the Multibranch Expansion method are explained.

**Rule 1** When the original logic function $f$ having $n$ variables is represented by the sum-of-products form, this form is rewritten to the sum-of-minterms form.

Received, 1995. 9. 16
* 情報工学科
** 情報工学科専攻修士学生
*** 日立製作所（情報工学科卒業生）

**Rule 2** The original function $f$ is rewritten as the form of the set $S_f^1$ consisting of the minterm numbers. And, for any sets generated by divisions, combinations, or the others repeated in order to obtain the prime implicants since this set $S_f^1$ has been generated, the minterm numbers, the elements of these sets, must be arranged in the ascending order as well.

**Definition 1** The set whose elements, the minterm numbers, have been arranged in the ascending order is called the ordered set.

**Definition 2** All the minterm numbers of the ordered sets $S_f^1$ are divided into the $2^{n_b}$ ordered sets $G_{j's}$ $(0 \leq j \leq 2^{n_b}-1)$ by the binary values of the $n_b$ consecutive variables. The set constructed by these $2^{n_b}$ ordered sets is called the division set (at the $i$'th expansion) $T^i$. And the $n_b$ consecutive variables are called the $n_b$ division bits.

**Definition 3** For the subscripts $j's$ $(0 \leq j \leq 2^{n_b}-1)$ of the ordered sets $G_{j's}$ which are the elements of the division set $T^i$, the $2^p$ (where $1 \leq p \leq 2^{n_b}$, $p$ is the integer) subscripts adjacent to each other are selected and put into the same group. The set consisting of these groups is called the adjacency set (at the $i$'th expansion) $R^i$.

**Rule 3** Several ordered sets $G_{j's}$ included in the division set $T^i$ at the $i$'th expansion, whose subscripts $j's$ belong to the same group set up by the adjacency set $R^i$, generate the new ordered sets of minterm numbers by the following procedures:

When some group of the adjacency set $R^i$ has the $2^q$ $(1 \leq q \leq 2^{n_b}$, $q$ is the integer) adjacent subscripts, it is supposed that they construct a kind of ordered set $(j_1, j_2, \cdots, j_{2^{q-1}}\ j_{2^{q-1}+1}, j_{2^{q-1}+2}, \cdots, j_{2^q})$ and that $j_r$ is adjacent to $j_{2^{q-1}+r}$ (where $0 \leq r \leq 2^{q-1}$) in this ordered set. Then, for each $r$, the minterm number $m_{2r} = m_{1r} + 2^s$, (where $s = (n - n_b) \times 2^{q-1}$) in the group $G_{j,2^{q-1}+r}$ is chosen for the minterm number $m_{1r}$ in the group $G_{j,r}$. For each $r$, these two minterm numbers $m_{1r}$ and $m_{2r}$ become the elements of the new minterm number group. For each $r$, the same way is applied to all the other minterm numbers belonging to the groups $G_{j,r}$ and $G_{j,2^{q-1}+r}$, as well. Of course, these new minterm number groups must be arranged as the new ordered set.

**Theorem 1** The number of new minterm number ordered sets generated by Rule 3 is equal to the number of elements included in the adjacency set $R^i$.
(**Proof**) It is very easily proved.

**Definition 4** Each new minterm number ordered set generated by Rule 3 becomes each element of the new set. This new set is called the consensus set (at the $i$'th expansion) $K^i$.

**Definition 5** For the elements included in either division set $T^i$ or consensus set $K^i$, when one element $e_1$ is covered by the other element $e_2$, that is, all minterm numbers included in one element are covered by some minterm numbers included in the other element, this element $e_1$ is deleted. This action is called the absorption, and we say that the element $e_1$ is absorbed by the element $e_2$.

**Theorem 2** Any element of either division set $T^i$ or consensus set $K^i$ consisting of the ordered set of minterm numbers becomes the candidate of prime implicant when the following conditions are sufficed [2] :

1. This element contains $2^l$ minterms (where $l$ is the integer).
2. Two sequences having $2^{l-1}$ consecutive minterm numbers obtained by extracting $2^p$ (where $p=0, 1, 2, \cdots, l-1$) consecutive minterm numbers every $2^p$ consecutive minterm numbers from one sequence having $2^l$ consecutive minterm numbers corresponding to the ordered set which is the element of either sets $T^i$ or sets $K^i$ and concatenating them, are adjacent to each other for all values of $p$.

**(Proof)** It is proved easily.

**Theorem 3** The candidate of prime implicant left without being absorbed by any other candidates of prime implicants becomes the prime implicant.
**(Proof)** This is proved very easily.

**Theorem 4** If any element correspondent to any lower group $G_R^i$ in the adjacency set $R^i$, included by the consensus set $K^i$, is absorbed by any other elements, it is absorbed by the elements correspondent to the groups having more subscripts than the $G_R^i$'s subscripts in the adjacency set $R^i$, included by the same consensus set $K^i$.
**(Proof)** It is omitted.

**Definition 6** The absorption described in Theorem 4 is called the first absorption.

**Theorem 5** If any element of the division set $T^i$ is absorbed by the other element, it is absorbed by the elements (generated by using this division set $T^i$ as shown in Definition 3, Rule 3 and Definition 4) in the consensus set $K^i$.
**(Proof)** It is proved easily.

**Definition 7** The absorption described in Theorem 5 is called the second absorption.

**Definition 8** When some candidates of prime implicants are absorbed by the other candidates of prime implicants, this absorption is called the third absorption.

## 3. Algorithm of Multibranch Expansion Method

In this section the algorithm of Multibranch Expansion method is described as follows :

**Step 1** For the given function $f$ having $n$ variables, the ordered set $S_f^1$ (or $S_f^i$) is set up according to Rules 1 and 2. The minterm numbers included by the function $f$ are shown by the correspondent decimal numbers.

**Step 2** The length $n_b$ of division bits is determined for the ordered set $S_f^1$ (or $S_f^i$). If the length $n_b$ of remaining bits on the way of the divisions, becomes shorter than the length $n_b$, the new length for division bits is shortened.

**Step 3** The top $n_b$ consecutive bits including the MSB are extracted from the binary numbers consisting of $n$ bits. The binary value combinations of these $n_b$ division bits divide the decimal number elements of the set $S_f^1$ (or $S_f^i$) into the $2^{n_b}$ different groups $G_{j's}$ (where $j = 0, 1, 2, \cdots, 2^{n_b}-1$).

**Step 4** For all subscripts $j's$ of groups $G_{j's}$ in the division set $T^i$, all the adjacent groups are generated. Then, they become all the elements of the adjacency set $R^i$ (in general at the $i$'th expansion).

**Step 5** For the division set $T^i$ generated at Step 3, the consensus sets $K^i$ are generated by referring to the adjacency set $R^i$.

**Step 6** The first absorptions are made between all elements included in the consensus set $K^i$. Then, the second absorptions are performed between the division set $T^i$ and the consensus set $K^i$.

**Step 7** For all the elements remaining in either consensus set $K^i$ or division set $T^i$ after all absorptions at Step 6, if they are the candidates of prime implicants, the third absorptions are performed by some candidates of prime implicants already obtained at the preceding expansion levels.

**Step 8** For all the elements remaining in either sets $K^i$ or sets $T^i$ after step 7, it is decided whether they are the prime implicants by the adjacency decisions (Theorem 2). The elements decided as the prime implicants are excluded after reserved. Then, the remaining elements are united and they replace the new set $S_f^{i+1}$ for the new set for the $(i+1)$th expansion.

**Step 9** When any elements which are neither the candidates of prime implicants nor absorbed by the other elements remain in sets $K^i$ or sets $T^i$ after step 8, the next expansion is performed by returning back to Step 2. When either of the two following conditions is sufficed after Step 8, this algorithm finishes:

1. There are no elements in both sets $K^i$ and sets $T^i$ for the next expansion.
2. There are no division bits because the expansions for all $n$ bits have been completed.

## 4. Example of Multibranch Expansion Method

In this section, an example is described for the algorithm of the Multibranch Expansion method.

As Step 1, $S_f^1$ for the function $f$ having five variables ($n=5$) is as follows:

$$S_f^1 = (0, 1, 3, 4, 5, 9, 11, 12, 13, 17, 18, 19, 20, 22, 23, 24, 26, 29, 30, 31). \tag{1}$$

As Step 2, the $n_b(=2)$ bits are set up. As Step 3, the two consecutive bits $b_4$ and $b_3$ are extracted. For four binary values of $b_4b_3$, namely 00, 01, 10 and 11, the four groups $G_0$, $G_1$, $G_2$ and $G_3$ are grouped respectively, and collected together as the expansion set $T^1$, as shown by the following equation:

$$T^1=(G_0, G_1, G_2, G_3) \quad (2)$$
$$=((0, 1, 3, 4, 5), (9, 11, 12, 13), (17, 18, 19, 20, 22, 23), (24, 26, 29, 30, 31)).$$

As Step 4, for the subscripts $j's$ $(=0, 1, 2$ and $3)$ of group $G_{j's}$, the adjacent set $R^1$ is obtained as follows:

$$R^1=((0, 1), (2, 3), (0, 2), (1, 3), (0, 1, 2, 3)). \quad (3)$$

As Step 5, some elements of the division set $T^1$ are combined by referring to the adjacent set $R^1$ of equation (3), and then the consensus set $K^1$ is obtained as follows:

$$K^1=((1, 3, 4, 5, 9, 11, 12, 13), (18, 22, 23, 26, 30, 31), (1, 3, 4, 17, 19, 20), (13, 29)). \quad (4)$$

As Step 6, the first and second absorptions are performed. There are no elements absorbed in this cases. As Step 7, the third absorptions are not required. As Step 8, after the adjacency decision for all elements in the division set $T^1$ and the consensus set $K^1$, the element (13, 29) in $K^1$ is decided as the candidate of prime implicant. This is memorized. The remaining elements are united as the new set $S_f^2$ as follows:

$$S_f^2=((1, 3, 4, 5, 9, 11, 12, 13), (18, 22, 23, 26, 30, 31), (1, 3, 4, 17, 19, 20), (0, 1, 3, 4, 5), (9, 11, 12, 13), (17, 18, 19, 20, 22, 23), (24, 26, 29, 30, 31)). \quad (5)$$

As Steps 2 and 3, the division set at the second expansion $T^2$ is generated by the $n_b(=2)$ bits.

$$T^2=((1, 9), (3, 11), (4, 5, 12, 13), (18, 26), (22, 23, 30, 31), (1, 17), (3, 19), (4, 20), (0, 1), (18, 19), (24), (29)). \quad (6)$$

As Step 4, the adjacent set $R^2$ equals $R^1$. As Step 5, the consensus set $K^2$ is obtained from the division set $T^2$ by referring to the adjacent set $R^2$. Then, as Step 6, after the first and second absorptions, the following equations of $K^2$ and $T^2$ are obtained:

$$K^2=((1, 3, 9, 11), (1, 5, 9, 13), (18, 22, 26, 30), (1, 3, 17, 19), (0, 1, 4, 5), (20, 22), (18, 19, 22, 23), (24, 26), (29, 31)). \quad (7)$$

$$T^2=((4, 5, 12, 13), (22, 23, 30, 31), (4, 20), (30, 31)). \quad (8)$$

As Step 7, the third absorptions are not efficient. As Step 8, the adjacency decisions using Theorem 1 are performed for all remaining elements in the expansion set $T^2$ and the consensus set $K^2$. As Step 9, all prime implicants are as follows:

$$S_f^2=((1, 3, 9, 11), (4, 5, 12, 13), (1, 5, 9, 13), (18, 22, 26, 30), (22, 23, 30, 31), (1, 3, 17, 19), (4, 20), (13, 29), (18, 19, 22, 23), (0, 1, 4, 5), (20, 22), (24, 26), (29, 31)). \quad (9)$$

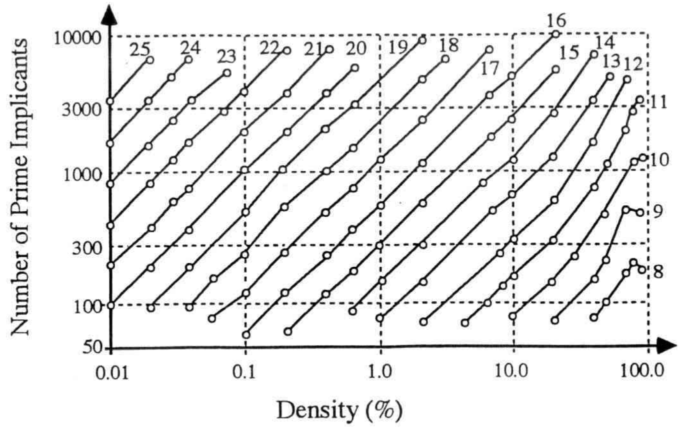


Fig. 1. Relation between Density and Number of Prime Implicants for Random Functions. (Variable Range : 8 through 25. $n_b = 1, 2$ or 3)

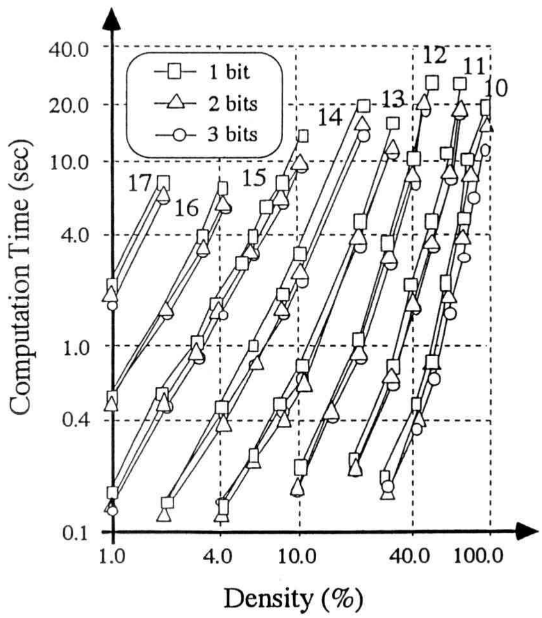


Fig. 2. Relation between Density and Computation Time for Different Values $n_b$. (Variable Range : 10 through 17. $n_b = 1, 2$ or 3)

## 5. Results and Discussions

For the Multibranch Expansion method, the C language program was prepared. By running this program on the SPARC station 5 (made by SUN-Microsystems), both the number of generated prime implicants and the computation times were measured for the different lengths $n_b$ of division bits (that is, $n_b = 1, 2$ and 3) and then compared. These results are shown in figures 1 and 2. In these measurements, the number of variables $n$ given in the original logic function were intended for ten variables through the twenty three or more variables. This method is applicable to any sum-of-products form of logic functions, and the decimal minterm numbers generated in the form of random numbers were applied to computer inputs. The abscissas of figures 1 and 2 show the minterm densities which mean the ratios of the number of minterms to the number of all possible minterms $2^n$. After these measurements were repeated ten times to an input consisting of minterm numbers corresponding to each minterm density and each variable, the results obtained were averaged.

From Fig. 1, it is proved that there are no differences between the three cases (of $n_b=$ 1, 2 and 3) concerning the number of prime implicants generated.

And then, this result shows that this method can endure to generate a number of prime implicants, 7,000 or more prime implicants at twenty four or more variables.

From Fig. 2, the following results are obtained for the computation time:

1. The computation time for $n_b=1$ bit is longer than the computation time for $n_b=2$ bits or $n_b=3$ bits.
2. For the smaller variables below seventeen, the computation times for $n_b=2$ bits have only few differences between the computation times for $n_b=3$ bits.
3. The computation time for $n_b=2$ or 3 bits is about 72% of the computation time for $n_b=1$ bit.

In addition, from figures 1 and 2, as for a larger number of variables more than eighteen, the computation time is determined by the number of generated prime implicants independent on the number of variables.

As for a smaller number of variables than eighteen, for the same number of prime implicants, the smaller the number of variables, the larger the computation time.

## 6. Conclusions

The prime implicants of a logic function were generated on the computer by using the Multibranch Expansion method of this paper. As a result, the computation time for the method using the division bits of $n_b=2$ or 3 bits is better than that the computation time for the method using the division bits $n_b=1$ bit, that is, the usual method such as the consensus method. And this method is completely suitable to a larger number of variables (for example more than twenty variables) and a larger number of prime implicants (for example more than 2,000 prime implicants).

## Acknowledgments

We wish to thank Mr. S. Shiohara for their efforts payed to measure a lot of data.

## References

[1] Goto, K. : "Five Methods for Simplification of Logic Function and Comparison of their Characteristics", Proc. of 1990 IEEE International Symposium on Circuits and Systems, Vol. 2, p. 1122, May, 1990.

[2] Goto, K., Tatsumi, H. and Kobayashi, S. : "Generation of Prime Implicants of Logic Function by Multi-Level-Division-Synthesis Method Including Improved-Consensus-Expansion Method", Proc. of JTC-CSCC '93, Vol. 1, p. 409, July, 1993.

[3] Goto, K., Menjo, M., Tatsumi, H., Ling, X.P. and Takahashi, S. : "Improvement of Prime Implicants Generation of Logic Function by Multi-Level-Synthesis Method", Proc. of the IEEE TENCON '94, Vol. 2, p. 938, Aug, 1994.