論 文
28～8～2

論理圖變換의 새로운 技法

New Techniques for the Transformation of the Logic Diagram

趙 東 燮* · 黃 熙 隆**
Dong Sub Cho, Hee Yeung Hwang

Abstract

This paper is concerned with not only the transformation of the logic diagrams to the NAND and the NOR forms but also the inverse transformation deriving the simple Boolean function from a logic diagram. The conversions of the algebraic expression from the AND, OR and NOT operations to the NAND and the NOR operations are usually quite complicated, because they involve a large number of repeated applications of De Morgan's Theorem and the other logic relations.

For the derivation of the Boolean function, it becomes difficult because the Boolean function is determined from the De Morgan's theorem in consecutive order until the output is expressed in terms of input variables (9).

But, these difficulties are avoided by the use of new techniques, called the TWO-NOTs method and the MOVING-NOT method, that are presented in this paper.

I. INTRODUCTION

In the practical cases, any Boolean function can be realized exclusively in terms of NAND or NOR gates. For the two-level forms, the transformation from the AND-OR to the NAND-NAND or from the OR-AND to the NOR-NOR is simply obtained. If the logic networks with more than two levels are used, the situation becomes considerably more complex. For the multi-level forms, the available transformation methods [1]~[9] are developed. And the inverse transformation methods are represented in references [1], [2] and [9]. But, these methods lack the simplicity and the general adaptation to specific functions such as functions including the Exclusive-OR gates.

New techniques are developed to handle the logic diagram simply and directly, and shown in Section II. A theorem is generated from the property of the EXCLUSIVE-OR operation and is applied to its graphic transformations. By applying these techniques shown in Section II, general and simple transformations of logic dagram to the NAND and the NOR forms are presented in Section III and IV, respectively, and the inverse transformation from a logic diagram to a Boolean function is introduced in Section V.

* 正會員 : 서울工大 電子計算機 工學科 大學院
** 正會員 : 〃 〃 〃 副敎授

接受日字 : 1979年 6月 15日

II. NEW TECHNIQUES FOR THE LOGIC GATES

The key of the transformation procedure is based on the De Morgan's Theorem. Shannon suggested De Morgan's theorem in the functional notation as follows:

$$f(X_1, X_2, \dots, X_n, +, .)^1 = f(X_1, X_2, \dots, X_n, ., +). \quad (1)$$

By modifying Eq. (1) to Eq. (2), the De Morgan's theorem becomes

$$f(X_i, +, .)^1 = f(X_i,).(, +), \text{ for } i = 1, 2, \dots, n. \quad (2)$$

For the any number of complementations, Eq. (2) can be further generalized into Eq. (3).

The generalized De Morgan's Theorem:

$$f(X_i,+,.,(\dots))^n = f(X_i^n,+^n,.^n,(\dots)^n)$$
$$= f(X_i,+,.,(\dots)), \text{ when } n \text{ is even.}$$
$$= f(X_i^1,).(,+,(\dots)), \text{ when } n \text{ is odd.} \quad (3)$$

Symbolically the De Morgan's theorem becomes

$$F^n = \begin{cases} F, & \text{when } n \text{ is even} \\ F^1, & \text{when } n \text{ is odd} \end{cases}$$

where n is equal to the number of complementations acting on the variables or operations. This theorem is proved in [9].

From the generalized De Morgan's Theorem, new techniques are derived as follows:

TECHNIQUE 1 (The TWO-NOTs method):

If n is even, the output function F^n is equal to the function F. Therefore two NOT gates can be inserted into the output line of any gate and, conversely, two NOT gates on the same line can be eliminated. These techniques are illustrated by the logic diagram in Fig. 1.

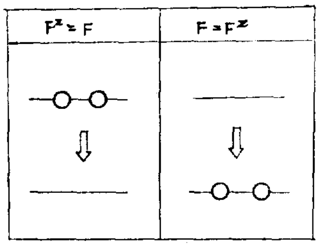


Fig. 1. The graphic representation of the TWO-NOTs method.

TECHNIQUE 2 (The MOVING-NOT method for the AND and the OR)

The well-known form of De Morgan's Theorem is a special case of Eq. (3) when n is equal to 1. In this case, the variables and the operations are changed and can be expressed in the algorithmic forms as follows:

$$X_i \longrightarrow X_i^1$$
$$+ \longrightarrow) \cdot (\longrightarrow) ($$
$$\cdot \longrightarrow +.$$

From these relations, new techniques of the logic diagram transformation are developed and used in the following Sections.

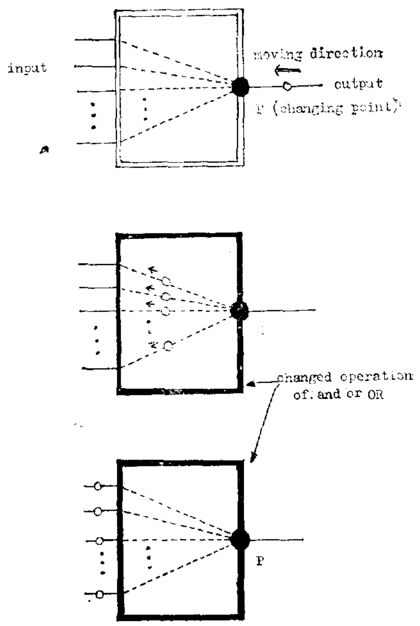
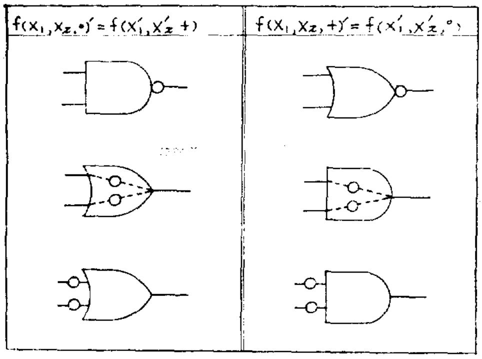


Fig. 2. Graphic representation of the MOVING-NOT method for the AND and the OR operation.

At first, a complementation (NOT operation) on the output of a gate is moved leftward through the gate along all the i branches as shown in Fig. 2. On moving the NOT through the point called the changing point, all the input variables are complemented and the operation is changed to the dual form: for example, from the AND to



$f(x_1,x_2,\cdot)' = f(x_1',x_2',+)$
$f(x_1,x_2,+)' = f(x_1',x_2',\cdot)$

The Expressions of the
Fig. 3. De Morgan's theorem by the MOVING-NOT method

the OR and from the OR to the AND, respectively. This procedure is illustrated in Fig. 2. Therefore, the De Morgan's theorem can be handled by the MOVING-NOT method as shown in Fig. 3.

TECHNIQUE 3 (The MOVING-NOT method for the EXCLUSIVE-OR):

In order to treat the EXCLUSIVE-OR operation in the logic diagram, the property of the EXCLUSIVE-OR operation is given as a Theorem below:

THEOREM-The property of the EXCLUSIVE-OR operation:

$$f(X_i,\oplus)^n = f(X_i(K),\oplus) \cdots\cdots\cdots (4)$$

where n is equal to the number of complementations on the logic output function $f(X_i,\oplus)$, and K (where $X_i(K)$ means that the number of complemented variables of all X_i's is equal to K) is even number iff n is even and K is odd number iff n is odd.

PROOF:

The EXCLUSIVE-OR operation has the following properties: By the definition of the EXCLUSIVE-OR, we obtain

$$X_1 \oplus X_2 = X_1.\bar{X}_2 + \bar{X}_1.X_2 \cdots\cdots\cdots (5)$$

Let X_2 be logical 1 and substituting this value of X_2 into Eq. (5), we find that

$$X_1 \oplus 1 = X_1.0 + \bar{X}_1.1 = \bar{X}_1 \cdots\cdots\cdots (6)$$

Using the associative law of EX-OR and Eq. (6), we obtain

$$\begin{aligned} X_1\oplus 1\oplus 1 &= (X_1\oplus 1)\oplus 1 && \text{(by using associative law)} \\ &= \bar{X}_1 \oplus 1 && \text{(from (6))} \\ &= \bar{\bar{X}}_1 && \text{(from (6))} \\ &= X_1 && \text{(from (3)).} \qquad (7) \end{aligned}$$

By applying Eq. (6) and Eq. (7), the property of the EXCLUSIVE-OR operation is derived as follows:

$$f(X_i,\oplus)^n = (X_1\oplus X_2 \oplus \ldots\ldots \oplus X_i)^r$$

$$= (\ldots\ldots(((X_1\oplus X_2\oplus\ldots\oplus X_i)\underbrace{\oplus 1)\oplus 1)\ldots\ldots\oplus 1)}_{}\text{(from Eq. (6))}$$

The number of the EX-OR operation is n.

$$= \begin{cases} (\ldots(((X_1\oplus X_2\oplus\ldots\oplus X_i)\underbrace{\oplus 1)\oplus 1)\ldots\oplus 1)}_{} \text{ (from Eq.(7))} \\ \text{The number of the EX-OR operation is } K \text{ (even number), if } n \text{ is the even number.} \\ (\ldots(((X_1\oplus X_2\oplus\ldots\oplus X_i)\underbrace{\oplus 1)\oplus 1)\ldots\oplus 1)}_{} \text{ (from Eq. (7))} \\ \text{The number of the EX-OR operation is } K \text{ (odd number), if } n \text{ is the odd number.} \end{cases}$$

$$= f(X_i(K),\oplus) \text{ (from the commutative law and the associative law of the EX-OR)}$$

Q.E.D.

Using the theorem, the MOVING-NOT method is developed for the EXCLUSIVE-OR. The only differences in comparison with the Technique 3 are that no operation change is made on the EXCLUSIVE-OR and one NOT moves along one branch when it is moved through the EXCLUSIVE-OR operator. These are illustrated in Fig. 4. Note that the CHANGING POINT is removed in this case. For the EXCLUSIVE-OR operation. the TWO-NOTs method and the MOVING-NOT method can be used as shown in Fig. 5.

TECHNIQUE 4 (The MOVING NOT method for BRANCH)

In the logic diagrams, gates often have the common input. The MOVING-NOT method is also

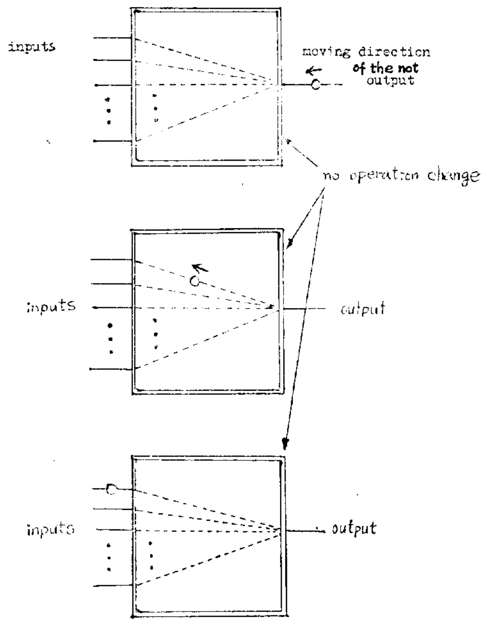


Fig. 4. Graphic representation of the MOVING-NOT method for the EXCLUSIVE-OR operation.

applicable to this case. It is clear that the relationships as shown in Fig. 6 are given.

Using these techniques, the transformations of the AND, OR and EXCLUSIVE-OR gate to NAND and the NOR forms are shown in the the Appendix.

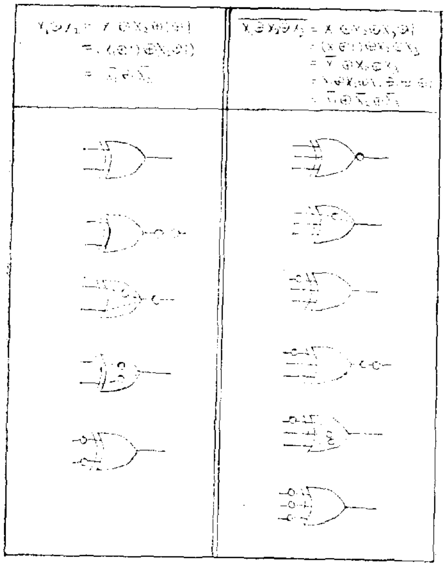


Fig. 5. The TWO-NOTs method and the MOVING-NOT method in the EXCLUSIVE-OR operation.

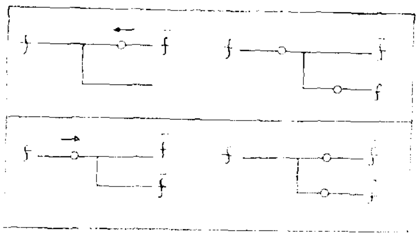


Fig. 6. The MOVING-NOT method for the branching.

Ⅲ. TRANSFORMATION TO NAND FORMS

Combinational logic circuits are more frequently constructed with the NAND and the NOR gates. Because of the prominence of the NAND and the NOR gates in the logic design, the transformation techniques have been developed [1]~[9]. Existing transformation techniques lack the general adaptation and the simplicity. New techniques for the NAND forms are presented in this Section and for the NOR forms in the next Section.

The conversion of a given logic diagram to the NAND forms is achieved by following steps:

Step 1 (TWO-NOTs method): Insert two NOT's into the output of the OR gates.

Step 2 (MOVING-NOT method): Using Technique 2. in the Section II, move one NOT leftward through the OR gates and obtain the NAND forms.

Step 3 (TWO-NOTs method): If the AND still remains in logic diagram, insert two NOT's into the output of the AND gate.

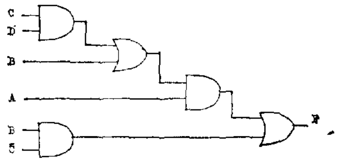
Otherwise, go to the Step 4.

Step 4 : NAND realization.

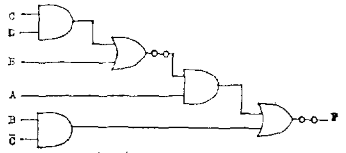
EXAMPLE 1 :

This procedure is illustrated in Fig. 7. for the function that is taken from reference [2] :

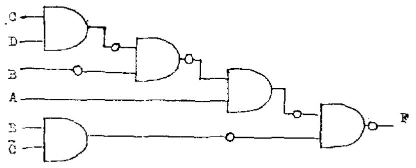
$$F=(CD+B)A+B\bar{C}$$



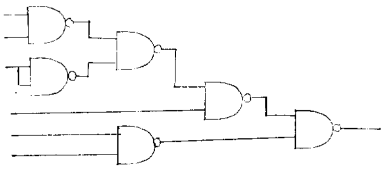
(a) The AND/OR forms of a function $F=(CD+B)\ A+B\bar{C}$



(b) Applying the TWO-NOTs method to fig. 7-(a)



(c) Applying the MOVING-NOT method to fig. 7-(b)



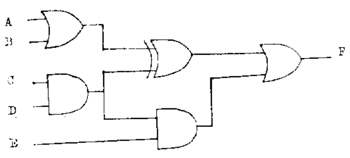
(d) The NAND realization from fig. 7-(c)

Fig. 7. Transformation of $F=(CD+B)\ A+B\bar{C}$ to NAND forms. (Example 1)

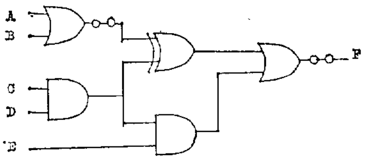
EXAMPLE 2 :

For a specific function $F=(A+B)\oplus CD+CDE$, the NAND realization is achieved as illustrated in Fig. 8.

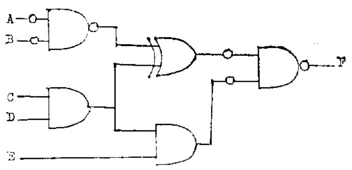
In Fig. 8-(D), a EXCLUSIVE-OR gate can be transformed to four NAND forms as in the Appendix.



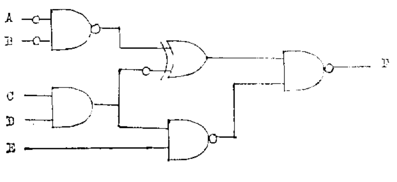
(a) AND/OR/EXCLUSIVE-OR forms of $F=(A+B)\oplus CD+CDE$



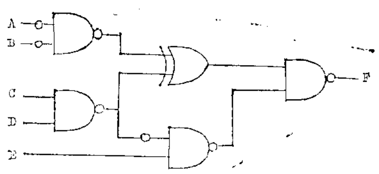
(b) Applying the TWO-NOT method to Fig. 8-(a)



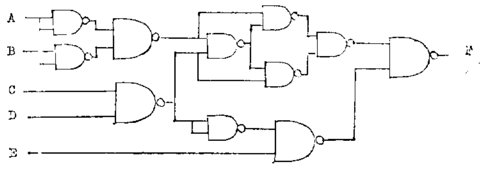
(c) Applying the MOVING-NOT method to Fig. 8-(b)



(d) Applying the MOVING-NOT method for EXCLUSIVE-OR



(e) Applying the MOVING-NOT method for Branch to Fig. 8-(d)



(f) NAND realization by substituting four NANDs for the EXCLUSIVE-OR

Fig. 8. Tranformation of $F=(A+B)\oplus CD+CDE$ to NAND forms. (Example 2)

IV. TRANSFORMATION TO THE NOR FORMS

Since the NOR operation is the dual of the NAND, the procedure for the NOR forms is the dual of that for the NAND forms.

The procedure consists of following steps:

Step 1 (TWO-NOTs method): Insert two NOT's into the output of the AND gates.

Step 2 (MOVING NOT method): Using Technique 2. in Section II, move one NOT leftward through the AND gates and obtain the NAND forms.

Step 3 (TWO-NOTs method): If the OR still remains in logic diagram, insert two NOT's into the output of the OR gate.

Otherwise, go to the Step 4.

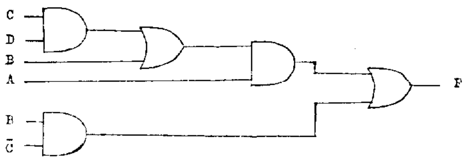
Step 4 : NOR realization.

EXAMPLE 3 :

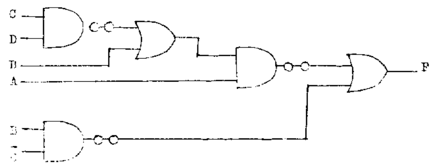
This procedure is illustrated in Fig. 9. for the same function of the EXAMPLE 1.

EXAMPLE 4 :

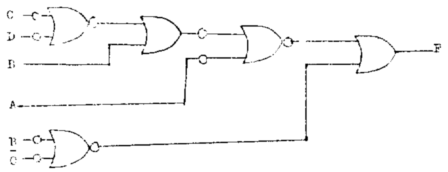
For a specific function that is the same as that given for EXAMPLE 2, the NOR transform is illustrated in Fig. 10.



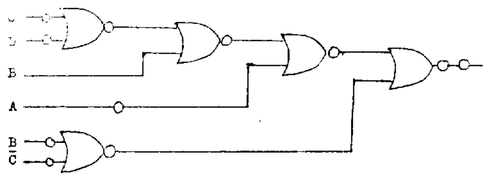
(a) The AND/OR forms of a function $F=(CD+B)A+B\bar{C}$



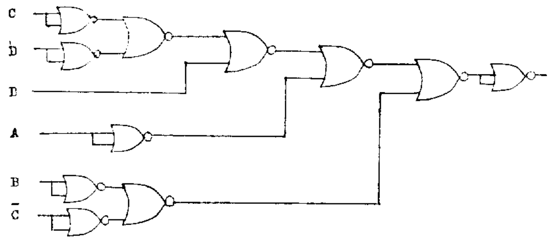
(b) Applying the TWO-NOSs method to Fig. 9-(a)



(c) Applying the MOVING-NOT method to Fig. 9-(b)

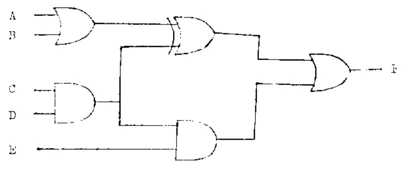


(d) Applying the TWO-NOTs method to Fig. 9-(c)

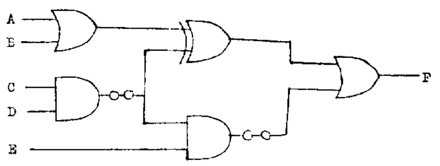


(e) NOR realzation.

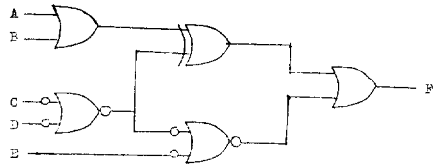
Fig. 9. Transformation of $F=(CD+B)\ A+B\bar{C}$ to NOR forms (Example 3)



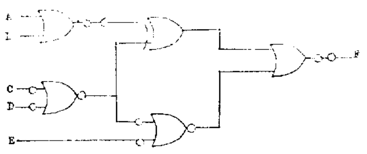
(a) The AND/OR forms of a function $F=(A+B)\oplus CD+CDE$



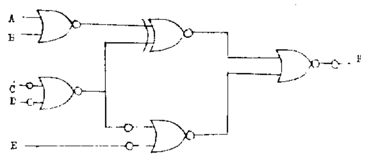
(b) Applying the TWO-NOTs method to Fig.10-(a)



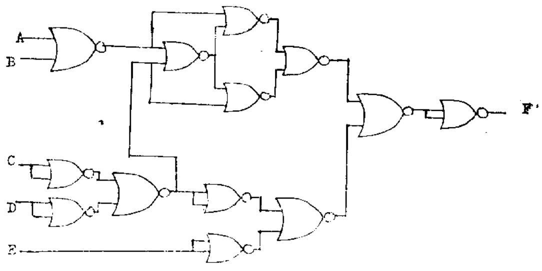
(c) Applying the MOVING-NOT method to Fig. 10-(b)



(d) Applying the TWO-NOTs method to Fig. 10-(c)



(e) Applying the MOVING-NOT method for the EXCLUSIVE-OR



(f) NOR realization by substituting the EXCLUSIVE-NOR to four NOR forms

Fig. 10. Transformation of $F=(A+B)\oplus CD+CDE$ to the NOR forms. (Example 4)

V. INVERSE TRANSFORMATION FROM A LOGIC DIAGAM TO BOOLEAN FUNCTION

It is more complex to find the sum-of-products terms from the logic diagram realized with NAND and NOR forms.

The conversion of the NAND and the NOR forms to the AND/OR/NOT forms is directly achieved by applying the MOVING-NOT method and the TWO-NOTs method.

The inverse transformation procedure is as follows.

Step 1 (MOVING-NOT method): Move leftward the rightmost NOT through the gate.

Step 2 (TWO-NOTs method): If there exist two NOT's on the same line, they are eliminated. If the NOT re mains in logic diagram, go to the Step 1.

Otherwise, go to the Step 3.

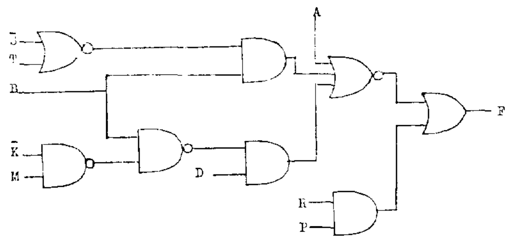
Step 3 : Find the sum-of-products terms.

EXAMPLE 5

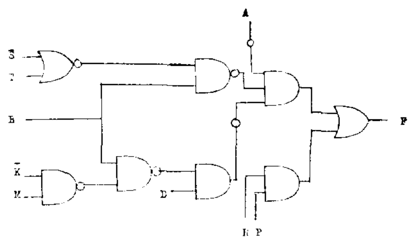
The procedure for logic diagram of $F=\overline{(A+B(\overline{S}+T)+\overline{D(B(\overline{MK}))})}+RP$ is illustrated in Fig. 11. This example is taken from [9].

From Fig. 11-(d), the sum-of-products terms are obtained,

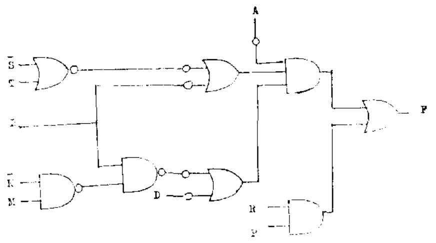
$$\begin{aligned}F&=(\bar{S}+T+\bar{B})\bar{A}((K+\bar{M})B+\bar{D})+RP\\&=(\bar{S}+T+\bar{B})\bar{A}(KB+\bar{M}B+\bar{D})+RP\\&=(\bar{S}\bar{A}+T\bar{A}+\bar{B}\bar{A})(KB+\bar{M}B+\bar{D})+RP\\&=\bar{A}BK\bar{S}+\bar{A}B\bar{M}\bar{S}+\bar{A}\bar{D}\bar{S}+\bar{A}BKT+\bar{A}B\bar{M}T+\\&\quad\bar{A}\bar{D}T+\bar{A}\bar{B}\bar{D}+RP.\end{aligned}$$



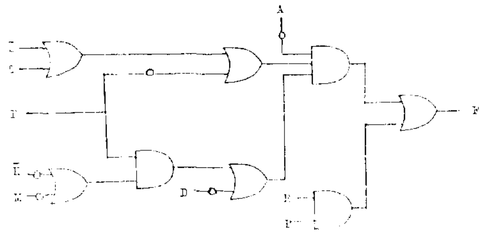
(a) Given logic diagram.



(b) Applying the MOVING-NOT method to the rightmost NOT.



(c) Applying the MOVING-NOT method to Fig. 11-(b)



(d) Applying the TWO-NOTs method and the MOVING-NOT method to Fig. 11-(c)

Fig. 11. Inverse transformation of the Example 5.

CONCLUSIONS

New techniques presented in the section II are very convenient and simple to transfer logic diagrams to the NAND and the NOR forms. These techniques require two methods, the TWO-NOTs method and the MOVING-NOT method, is generated in the Section II and used in every transformation. For a specific logic diagram including the EXCLUSIVE-OR gates, these methods are also applicable from the theorem presented in the Section II. In general, conclusions are summarized as follows.

1) When the logic diagram has many levels, techniques using the TWO-NOTs method and the MOVING-NOT method are more efficient than existing methods [1]~[9].

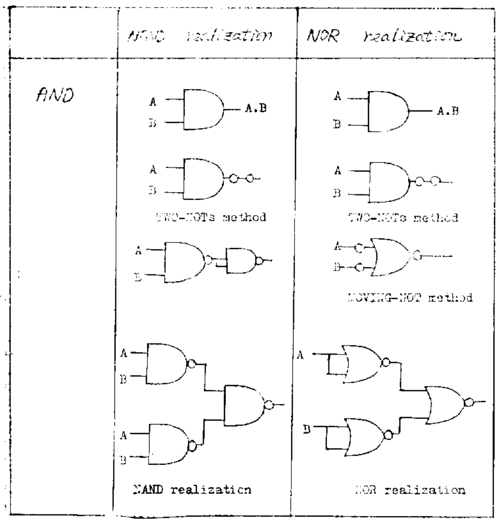
2) For the logic diagrams including the EXCLUSIVE-OR operation, two methods are also applied without difficulty

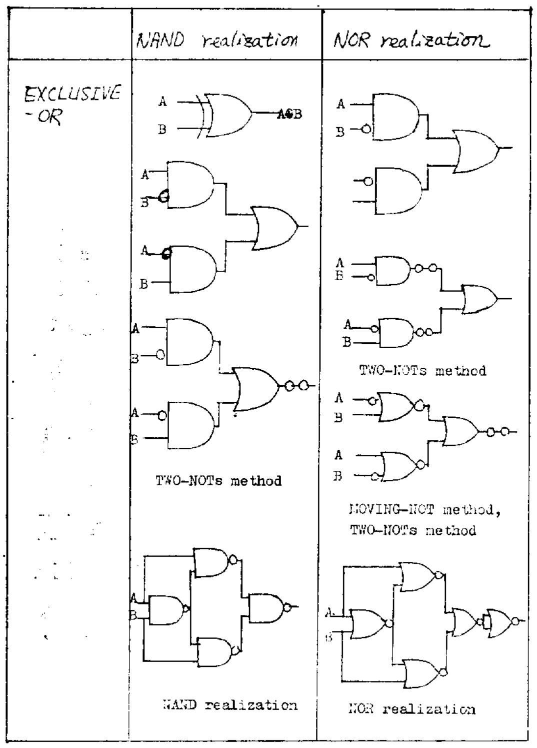
3) De Morgan's theorem is simply applied to the logic diagram.

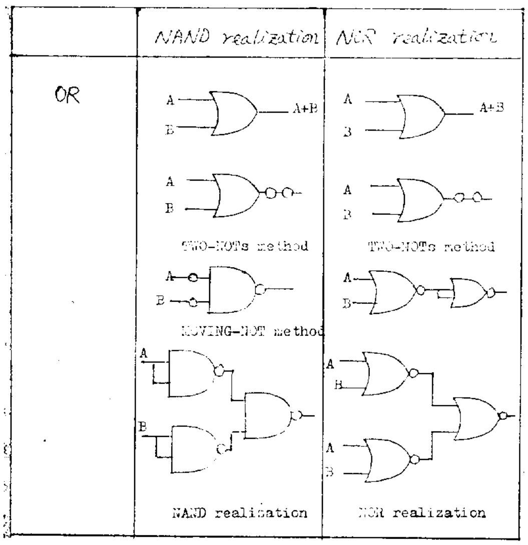
4) Because of the graphic transformation, it is very obvious that the procedure is simple and straightforward.

APPENDIX

The transformations of AND, OR and EXCLUSIVE-OR operations to the NAND and the NOR forms are given as follows.







REFERENCES

[1] F.J. Hill and G.R. Peterson, Introduction to Switching Theory and Logical design, Wiley, New York, 1974, pp. 194~199.

[2] M.M. Mano, Computer Logic Design, Englewood Cliffs, N.J.: Prentice-Hall, 1972, pp. 120~160.

[3] Douglas Lewin, Logical Design of Switching Circuits, Nelson, London, 1974, pp. 221~229.

[4] Taylor L. Booth, Digital Network and Computer Systems, McGraw Hill, New York, 1971,

pp. 104~107.

〔5〕 John. B Peatman, The Design of Digital System, McGraw Hill, New York, 1972. pp. 76~82.

〔6〕 V.T. Rhyne, Fundamentals of Digital Systems Design, Englewood Cliffs, N.J.: Prentice-Hall, 1973.

〔7〕 H.V. Malmstadt and C.G. Enke, Digital Electronics for Scientists, Benjamin, New York, 1969, pp. 168~173.

〔8〕 D.D. Givone, Introduction to Switching Circuit Theory, McGraw Hill, New York, 1970, pp. 206~216.

〔9〕 H. Christian Wehrfritz, "Techniques for the Transformation of Logic Equations," IEEE Trans. Comput. Vol. C-23, pp. 477~480, May, 1974.

訂　　　正

第28卷 第6號 ('79. 6) p. 62

Simple Table 方法에 의한 論理函數 最小化의 新方法 중에서……

PROOF:

There are only two cases satisfying $Td-Bd=2^i$ $(i=0, 1, 2, \cdots\cdots, n-1)$ among minterms without loss of generality.

의 部分을 아래와 같이 訂正합니다.

PROOF:

There are only two cases satisfying two conditions that $Td-Bd=2^i$ $(i=0, 1, 2, \cdots\cdots, n-1)$ and the distance between the Td and the Bd is not more than 2 when they are chosen arbitrarily among the given minterms without loss of generality.