論 文
28～12～2

交叉表와 價格表를 利用한 多重出力 論理函數의 最小化

Minimization of the Multi-Output Switching Function by using the Intersection Table and the Cost Table

黃 熙 隆*·金 豪 謙**·朴 永 鐵***·趙 東 燮****
(Hee Yeung Hwang, Ho Kyum Kim, Young Chul Park, Dong Sub Cho)

Abstract

The minimization of the multi-output switching function becomes a difficult task when the input variables and the number of functions increase. This paper describes the optimal selection of prime implicants for the multi-output switching function by using the Intersection Table. This procedure is applicable to both manual and computer programmed realization without complexity. The algorithm is implemented by a computer program in the standard FORTRAN Ⅳ language.

Ⅰ. INTRODUCTION

A large number of the established problems in logic design have multi-output switching function. One of the approaching methods would be to implement each function completely independently by the techniques [5] already developed. For many of the simpler logic networks, such approach is realistic solution to the problem. But, as the function becomes more complex, we frequently wish to implement a number of different functions of the same set of input variables.

This sharing of hardware between various functions affects the overall cost of the network design, so there have been other techniques [1]-[4], [6] and [7] to share as many logic circuit elements as possible. Goal of each techniques is the minimization of switching functions with the minimal cost by retaining as much commonality as possible between switching functions. In this paper an application from the Simple Table and the DA Table [5] is generated. A criterion for minimality is introduced in Chapter Ⅱ. From the Intersection table defined in Chapter Ⅳ, the commonality of minterms between functions are found in Chapter Ⅱ of this work easily. The generation of PI's is treated in Chapter Ⅲ as the previous work of one of these authors. The optimal set of prime implicants which cover the function is selected from the Intersection Table by Using the Cost Table and the Subcost Table defined in Chapter Ⅱ of this work.

* 正會員：서울大 工大 副敎授
** 〃 서울大大學院
*** 〃 〃
**** 〃 〃
接受日字：1979年 8月 31日

Ⅱ. CRITERIA OF COST AND SUBCOST

1) Cost consideration

Before developing a minimization procedure for the multi-output switching function it is necessary to introduce a cirterion to evaluate the network cost.

In this work, a cost criterion for the minimality among various criteria [6] is the number of the connecting lines which were covered by selecting any PI or minterm. Let's examine the following two cases.

Case 1) If any PI covers minterms m_1, m_2, m_3, and m_3 for the function F_a as in Fig. 1, then 4 connecting lines will be covered by selecting this

PI. Therefore, the Cost for this PI is 4.

Case 2) If any minterm m_1 is covered by the function F_a, F_b, F_c and F_d then 4 connecting lines will be covered by selecting m_1, as in the Fig. 2. The Cost for this minterm is 4.

In the above both cases of same cost, being the number of the input lines of the PI fewer than that of the minterm, the selecting of PI is performed earlier than that of minterm in the selection process of PI's.

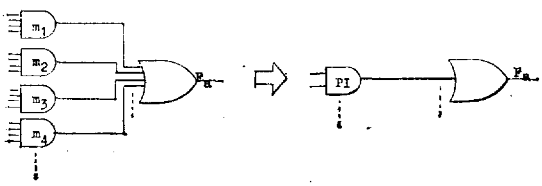

Fig. 1. Definition of the Cost for Case 1.(The Cost is 4)

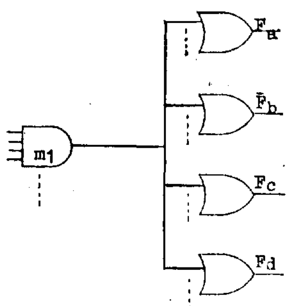

Fig. 2. Definition of the Cost for Case 2.(The Cost is 4)

2) Subcost consideration

If we select any PI in the Intersection Table as in Fig. 11, the number of minterms which were covered by this PI is decided upon the commonality of this PI. Therefore we define the sum of the remaining functions which were not covered for these minterms as the Subcost for this PI.

If PI$_a$ and PI$_b$ are given, the Cost of two PI's are same and the Subcost of PI$_a$ is lower than PI$_b$, we must select PI$_a$ at first in the PI's selection.

For example, let minterm m_2 is covered by functions F_a and F_c, m_3 by F_c, m_{10} by F_a, F_b and F_c, m_{11} by F_a, F_b and F_c as in Fig. 3, and PI$_a$ covers m_2, m_3, m_{10} and m_{11} for F_c, and PI$_b$ covers m_2 and m_{10} for F_a and F_c commonly.

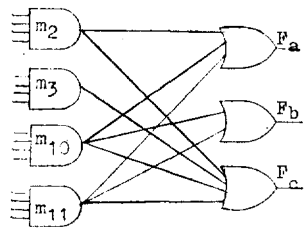

Fig. 3. Connections for the given switching network.

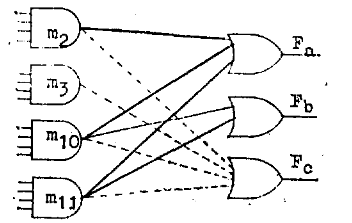

Fig. 4. Remaining connections by selecting PI$_a$. Cost=4, Subcost=5.

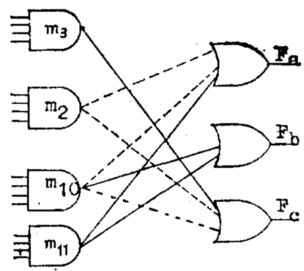

Fig. 5. Remaining connections by selecting PI$_b$. Cost=4, Subcost=1.

From Fig. 3, by selecting PI$_a$, whose Cost is 4, the number of the remaining connections is 5. Therefore the Subcost is 5 and it's graphic representation is shown as in Fig. 4. At Fig. 3, by selecting PI$_b$, the number of the remaining connection is 1. Therefore the Subcost is 1 and it's graphic representation is shown in as Fig. 5. From the above, it would be better to select PI$_a$ than PI$_b$. Therefore, if the Cost of PI's are same, selecting the PI whose Subcost is lower will be better. And if the Cost and the Subcost of PI's are same, selecting any PI becomes an optional case.

Ⅲ. GENERATION OF PRIME IMPLICANTS

For the generation of PI's, the application of the Simple Table and DA Table [5] to the multi-output function is quite similar to the single output function.

A little difference between the single output function and the multi-output function is in the selection of PI's. In the case of single output function the dominated PI's which were included in a larger PI were not considered, but in the multi-output function all the dominated PI's should be considered since the dominated PI's can have higher commonality. If any dominated PI has higher commonality than the PI, it should be listered in PI table.

Let's find a minimal sum-of-prodcts realization for three functions represented in Fig. 7 as an example, which was taken from [1], pp. 161~166.

$F_a = \sum m(2,4,10,11,12,13)$

$F_b = \sum m(4,5,10,11,13)$

$F_c = \sum m(1,2,3,10,11,12)$

a) Three output logic functions.

Minterm	F_a	F_b	F_c
1			1
2	1		1
3			1
4	1	1	
5		1	
10	1	1	1
11	1	1	1
12	1		1
13	1	1	

b) Commonality Table.

Fig. 6. An example for three output logic network.

From Fig. 6 we can show the Simple Table and the DA Table for the function F_a as Fig. 7 [6]. And then all the PI's for F_a can be generated.

Same procedures for the rest of functions F_b, and F_c, must be done similary. By this way we can construct the PI Table for the above example as Fig. 8. From Fig. 8, by deleting PI's which are same with the others, the Reduced PI table is constructed as Fig. 9 easily.

SIMPLE TABLE								DA TABLE			
		TEST MINTERMS									
BASE MINTERMS		2	4	10	11	12	13	8	4	2	1
	2	0		8				0			
	4		0			8		0			
	10			0	1			1			0
	11				0						1
	12					0		1			0
	13						0				1

Fig. 7. The Simple Table and the DA Table of function F_a for the generation of PI's

Function	PI	Commonality of PI's
F_a	2,10 (8)	F_a, F_c
	10,11 (1)	F_a, F_b, F_c
	4,12 (8)	F_a
	12,13 (1)	F_a
F_b	4,5 (1)	F_b
	10,11 (1)	F_a, F_b, F_c
	5,13 (8)	F_b
F_c	2,3,10,11(1,8)	F_c
	2,10 (8)	F_c, F_a
	1,3 (2)	F_c
	10,11(1)	F_a, F_b, F_c

Fig. 8. PI table.

PI		Commonality of PI's
$A=1,3(2)$	00×1	F_c
$B=2,3,10,11(1,8)$	×01×	F_c
$C=2,10(8)$	×010	F_a, F_c
$D=4,12(8)$	×100	F_a
$E=4,5(1)$	010×	F_b
$F=5,13(8)$	×101	F_b
$G=10,11(1)$	101×	F_a, F_b, F_c
$H=12,13(1)$	110×	F_a

Fig. 9. Reduced PI table.

Ⅳ. PI's SELECTION FOR THE MULTI-OUTPUT SWITCHING FUNCTION

1) Intersection Table

The optimal selecting of the PI's for the multi-output switching function can be implemented by using the Intersection Table which is composed of the Reduced PI table obtained in Chapter Ⅲ and the Commonality table for all minterms. The calculation of the Cost and the Subcost in this table was implemented as in Chapter Ⅲ.

2) Selection of the PI's

The procedure of selecting the PI's and the minterms in the Intersection Table is as follows:

(a) Select the PI or the minterm which has maximum Cost.

(b) If the Costs of the PI and the minterm which satisfy (a) are same, select the PI.

(c) Select the PI which has minimum Subcost among the PI's whose Costs are same.

(d) If the PI's or the minterms which satisfy (a),(b) and (c) are more than one, select any of them optionally.

(e) Repeat (a) through (d) until all the Costs are 0.

(f) Arrange the chosen PI's and minterms.

(g) Delete the chosen PI's and minterms which are included to the others.

(h) Hardware implementation by the result of (g)

By this procedure the hardware for the minimal cost can be constructed. A standard formulation of the Intersection Table was shown in Fig. 10 for the above example.

For the simple explanation, let us select the PI's in the Intersection Table of Fig. 10 for the given example.

From Fig. 10, it is clear that the Cost of G is maximum. Therefore we must select G first of all.

Since G cover minterm 10 and 11 for the function F_a, F_b and F_c commonly, minterms 10 and 11 were perfectly covered, so we delete the "1" for two columns.

By the above step (step 1), we update the Intersection Table as next step (step 2) by replacing the Cost and the Subcost for all minterms and PI's.

To show easily, next step was shown in Fig. 11. In Fig. 10 and Fig. 11, the subscript number represents the deleting order for the "1", that is, the step number and the asterisked number shows the order of the chosen PI or minterm.

At the step 2 it is clear that PI's, A and C, have same maximum cost of 2 and minimum subcost of 0. Selecting any of them is an optional case, so we select A arbitrary.

At the next step 3 we select C.

At the step 4 we can arbitrary select E or F. Therfore we select E . By selecting E, E

		All Minterms									Commonality of PI's			Cost(Subcost) of PI's
		13	12	11	10	5	4	3	2	1	Fa	Fb	Fc	
PI	A							1		1			1	2(0)
	B			1_1	1_1			1	1				1	4(5)
	C				1_1				1		1		1	4(1)
	D		1				1				1			2(2)
	E					1	1					1		2(1)
	F	1				1						1		2(1)
*1	G			1_1	1_1						1	1	1	6(0)
	H	1	1								1			2(2)
Commonality of Minterms	Fa	1	1	1_1	1_1		1		1					
	Fb	1		1_1	1_1	1	1							
	Fc		1	1_1	1_1			1	1	1				
Cost of minterms		2	2	3	3	1	2	1	2	1				

"1" means the checking symbol.

Subscript number represents the step number.

Fig. 10. Intersection Table (step 1).

		*6	*7	All Minterms							Commonality of PI's			Cost(subcost) of PI's					
		13	12	11	10	5	4	3	2	1	Fa	Fb	Fc	step2	step3	step4	step 5	step6	step7
*2	A							1_2		1_2			1	2(0)	0	0	0	0	0
	B							1_2	1_3				1	2(1)	1(1)	0	0	0	0
PI *3	C								1_3		1		1	2(0)	2(0)	0	0	0	0
*5	D		1_5				1_5				1			2(2)	2(2)	2(2)	2(1)	0	0
*4	E					1_4	1_4					1		2(1)	2(1)	2(1)	0	0	0
	F	1_6				1_4						1		2(1)	2(1)	2(1)	1(1)	1(1)	0
*1	G										1	1	1	0	0	0	0	0	0
	H	1_6	1_5								1			2(2)	2(2)	2(2)	2(2)	1(1)	0
Commonality of Minterms	Fa	1_6	1_5				1_5		1_3										
	Fb	1_6				1_4	1_4												
	Fc		1_7					1_2	1_3	1_2									
Cost of Minterms	step 2	2	2	0	0	1	2	1	2	1									
	step 3	2	2	0	0	1	2	0	2	0									
	step 4	2	2	0	0	1	2	0	0	0									
	step 5	2	2	0	0	0	1	0	0	0									
	step 6	2	1	0	0	0	0	0	0	0									
	step 7	0	1	0	0	0	0	0	0	0									
		0	0	0	0	0	0	0	0	0									

Fig. 11. Reduced Intersection Table

covers 4 and 5 only for F_b. So we can't cover the "1" which is located 4 column and D row, 4 column and F_a row.

For next steps, D and 13 are selected in order. At the step 7 we select 12 only for the function F_c. By step 7 because all costs are set to 0, we can arrange the chosen PI's as in Fig. 12.

Chosen PI's and minterms	Commonality		
	F_a	F_b	F_c
G	1	1	1
A			1
C	1		1
E		1	
D	1		
13	1	1	
12			1

Fig. 12. Chosen PI's and minterms table.

From Fig. 12, any of the PI's and the minterms were not included to the others. Therefore the answer is as follows and it's hardware realization is represented in Fig. 13.

$$F_a=C+D+G+13=\bar{X}_2X_3\bar{X}_4+X_2\bar{X}_3\bar{X}_4+X_1\bar{X}_2X_3+X_1X_2\bar{X}_3X_4$$

$$F_b=E+G+13=\bar{X}_1X_2\bar{X}_3+X_1\bar{X}_2X_3+X_1X_2\bar{X}_3X_4$$

$$F_c=A+C+G+12=\bar{X}_1\bar{X}_2X_4+\bar{X}_2X_3\bar{X}_4+X_1\bar{X}_2X_3+X_1X_2\bar{X}_3\bar{X}_4$$

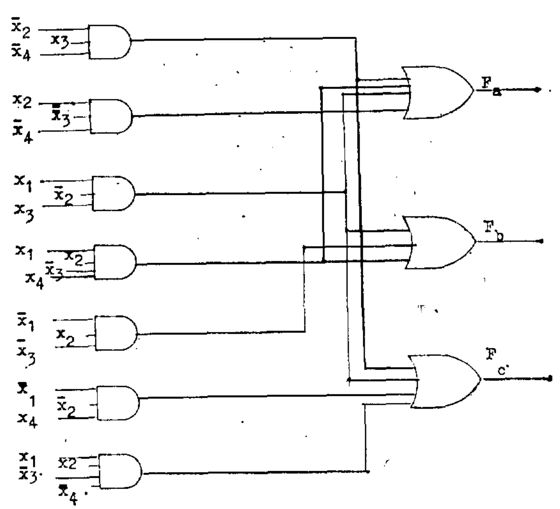

Fig. 13. Reduced multi-output logic network realization.

V. DON'T CARE CONSIDERATION

For the functions including don't care minterms, additional techniques are needed.

$F_a=\sum m(2,10,11,12,13)+d(4)$

$F_b=\sum m(4,5,10)+d(11,13)$

$F_c=\sum m(2,3,10,12)+d(1,11)$

a) Three output logic functions with don't care minterms.

Minterm	F_a	F_b	F_c
1			d
2	1		1
3			1
4	d	1	
5		1	
10	1	1	1
11	1	d	d
12	1		1
13	1	d	

b) Commonality table.

Fig. 14. An example for three output logic network.

In the PI's selection, the don't care minterms are treated as normal minterms. If the commonality of any PI consistsof only don't care minterms, that PI must be deleted.

In the Intersection Table, don't care minterms are represented as "d" only in the commonality of minterms

In the evaluation of the Cost and the Subcost, only normal minterms must be considered. By the above consideration, let's examine next example shown in Fig. 14.

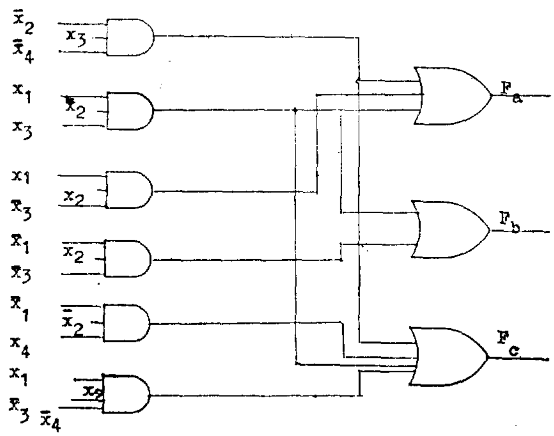

Fig. 15. Reduced multi-output logic network realization.

		All Minterms									Commonality of PI's			Cost(Subcost) Of PI's						
		13	12 (*6)	11	10	5	4	3	2	1	Fa	Fb	Fc	step1	step2	step3	step4	step5	step6	step7
PI'S *5	A							1_5		1			1	1(0)	1(0)	1(0)	1(0)	1(0)	0	0
	B			1	1			1_5	1_2				1	3(4)	2(1)	1(0)	1(0)	1(0)	0	0
*2	C				1				1_2		1		1	4(1)	2(0)	0	0	0	0	0
	D		1				1				1			1(2)	1(2)	1(2)	1(1)	0	0	0
*3	E					1_3	1_3					1		2(0)	2(0)	2(0)	0	0	0	0
	F	1				1						1		1(1)	1(1)	1(1)	1(1)	0	0	0
*1	G			1_1	1_1						1	1	1	4(0)	0	0	0	0	0	0
*4	H	1_4	1_4								1			2(1)	2(1)	2(1)	2(1)	0	0	0
Commonality of Minterms	Fa	1_4	1_4	1_1	1_1		d		1_2											
	Fb	d		d	1_1	1_3	1_3													
	Fc		1_6		1_1			1_5	1_2	d										
Cost of Minterms	step1	1	2	1	3	1	1	1	2	0										
	step2	1	2	0	0	1	1	1	2	0										
	step3	1	2	0	0	1	1	1	0	0										
	step4	1	2	0	0	0	0	1	0	0										
	step5	0	1	0	0	0	0	1	0	0										
	step6	0	1	0	0	0	0	0	0	0										
	step7	0	0	0	0	0	0	0	0	0										

Fig. 16. Intersection Table.

It is obvious that as this procedure treats all the cost and the subcost numerically, and straight-forward, this procedure is applicable to both manual and computer programmed realization by using the computer. The overall flowchart is shown in Fig. 17.

From Fig. 14, we can generate the PI by using the Simple Table and the DA Table. Examining Fig. 12 and Fig. 15, we can see that two Intersection Tables are same except that later table includes don't care representation in the commonality of minterms.

As explained earlier, by step 1 through step 7, PI's and minterms can be easily selected. From the above result, the answer is as follows and it's graphical representation is shown in Fig. 16.

$$F_a = C+G+H = \bar{X}_2 X_3 \bar{X}_4 + X_1 \bar{X}_2 X_3 + X_1 X_2 \bar{X}_3$$

$$F_b = E+G = \bar{X}_1 X_2 \bar{X}_3 + X_1 \bar{X}_2 X_3$$

$$F_c = A+C+G+12 = \bar{X}_1 \bar{X}_2 X_4 + \bar{X}_2 X_3 \bar{X}_4 + X_1 \bar{X}_2 X_3 + X_1 X_2 \bar{X}_3 \bar{X}_4$$

Ⅵ. FLOW CHART

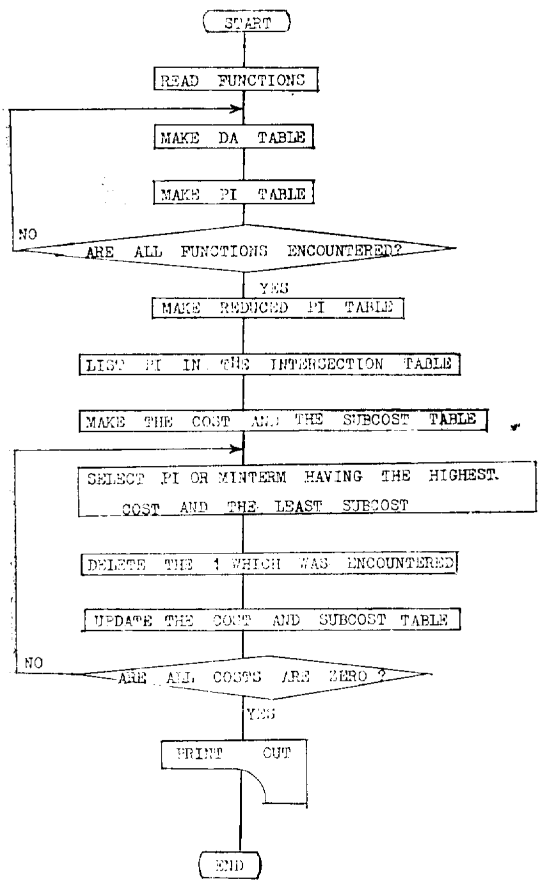

Fig. 17. The overall flow chart for the computer program.

Ⅶ. CONCLUSIONS

This paper is concerned with the minimization of the multi-output switching functions by the Intersection Table presented here. Owing to the Simple algorithm of both generating and selecting the PI's and numerically treated Cost and subcost, it can be easily programmed on the digital computer. This computer algorithm depends on the constraints given by the Cost Table and the Subcost Table, which is useful for the two level (AND/OR) realization regardless of fan-in of gate.

In general, the conclusions of this work can be summarized as follows:

1) Because of the simplicity of the Intersection Table, the minimization of the multi-output function under the given criterion becomes more easier than ealier works [1]-[4], [6] and [7].

2) This method will be extended to the minimization of the multi-output function under other criterion.

3) In treating the multi-output function prime Implicants are obtained easily from the DA table as ealier work [5] done by one of these authors.

4) The experiments with various examples show that the Intersection Table method is efficient.

ACKNOWLEDGEMENT

These authors wish to acknowledge their grateful thanks to Prof. Yang Heung Suk for his valuable suggestion and encouragement, and to a senior student Dept. of Electrical Eng., College of Eng., S.N.U. Lee Gyeu Sang for his participation in program work.

REFERENCES

[1]. F.J. Hill and G.R. Perterson: Introduction to switching theory and logical design, Wiley, New York, 1974.

[2]. V.T. Rhyne, Fundamentals of digital system, Prentice-Hall, Englewood Cliffs, NJ. 1973.

【3】. M.M. Mano,Computer logic design, Prentice-Hall, Englewood Cliffs, NJ. 1972.

【4】. T.L. Booth, Digital networks and computer systems, John Wiley, 1972.

【5】. Hee Yeung Hwang, "A new approach to the minimization of switching functilons by the Simpe Table method," 대한전기 학회지, 제28권 제 6호, pp. 451∼467, 1979.

【6】. Givone, Introduction to switching circuit theory, McGraw Hill, New York, 1970.

【7】. Peter R. Schneider and Donald L.Deitmeyer, "An Algorithm for Synthesis of multiple-output Combinational Logic": IEEE Transactions on Computer. Vol. C-17, No. 2, February 1968.