연구논문집 제11권 제1호 p.15~20, 1980

An Algorithm for Minimization of The Multi-Output Switching Function

Sa Won Jang
Dept. of Computer Science

多出力 논리함수의 최소화에 대한 Computer Algorithm

張 師 元
전산공학과

〈요 약〉

多重出力의 論理函數는 入力變數와 出力函數가 增加함에 따라 複雜해지는데, 本論文에서는 共通된 민텀을 직접 볼 수 있는 交叉表를 만들고 이를 利用하여 價格表를 構成한 다음 各 出力에 대한 Prime implicant를 選擇하고 있기 때문에 設計의 複雜性을 除去할 수 있다. 本 過程은 쉽게 손으로 處理되며 또한 컴퓨터로도 處理될 수 있는데 컴퓨터 프로그램은 FORTRAN IV 言語로 짜서 결과를 확인하였다.

Ⅰ. Introduction

A large number of the established problems in logic design have multioutput switching function. One of the approaching methods would be to implement each function completely indepently by the technique【5】 already developed. For many of the simpler logic networks, such approach is a 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] to share as many logic circuit elements as possible. 〈Goal of each techniques are the minimization of switching functions with the *optimal* 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 this author [5].〉 The optimal set of prime implicants which cover the function is selected from the cyclic chart by using the Cost Table defined in Chapter Ⅱ of this work.

Ⅱ. Cost Table

1. Criterion for Minimality

Before developing a minimization procedure for multiple-output switching circuits, it is necessary to introduce a criterion to measure the network cost. In this work, a criterion for the Minimality among various criteria [6] is selected as follows:

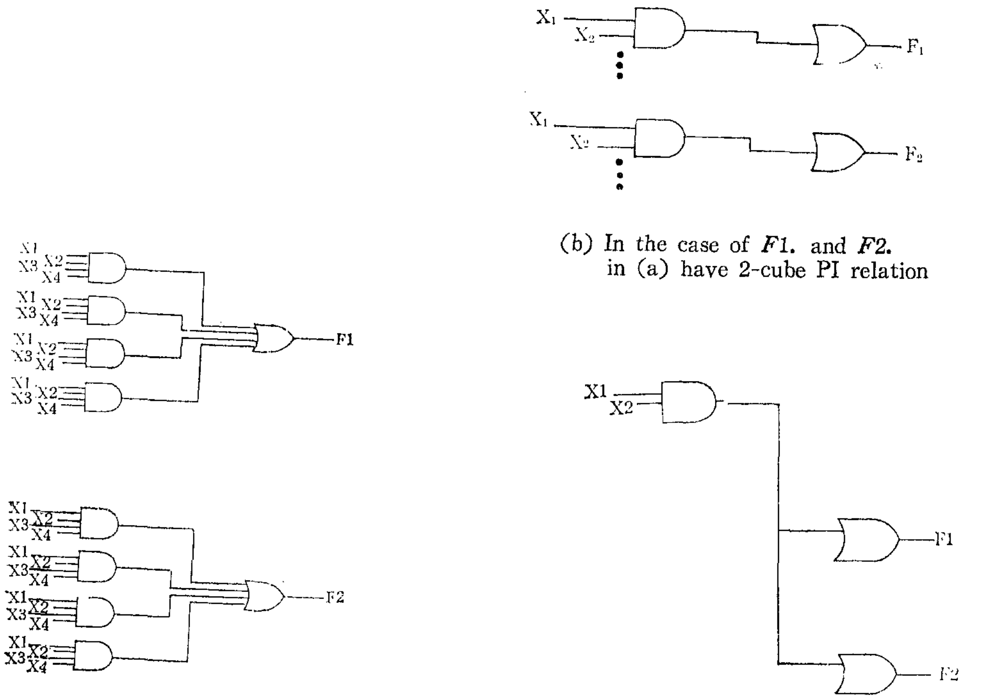
Let $F1, F2, \cdots, Fm$ be the set of normal switching expressions describing a multiple-output switching circuuit and let $t1$, $t2,\cdots$, tp be the set of all distinct terms appearing in the m output expres sions. The cost of the multiple output switching circuit is p, the number of distinct terms in the representation. In the case of that two switching circuits have the same number of distinct terms but different realizations, the number of literals in the term t_j is also considered in the cost evaluation.

2. Cost Consideration

Considerable savings often be achieved by the sharing of hardware among the multi-output functions.

It is clear that minimum total number of gates are required. For the minimum sum of products realization of the given switching function, it is considered that how many gates are eliminated by minimization of the functions.

For example, if two different output functions have a same 2-cube PI(prime implicant) which is reduced one minterm from four minterms, 8 AND gates can be realized to 1 AND gate, therefore 7 AND gates can be saved as shown in Fig. 1. In other words, if we don't restrict the number of inputs of any gate and multi-output functions have n-commonality of the n-cube PI, $(2^n \times m-1)$ AND gates can be saved in comparison to the logic circuit that realized separately. From the above description, we can constuct the Cost Table as shown in Fig. 2, where the cost means the number of AND gates that can be reduced to one AND gate.

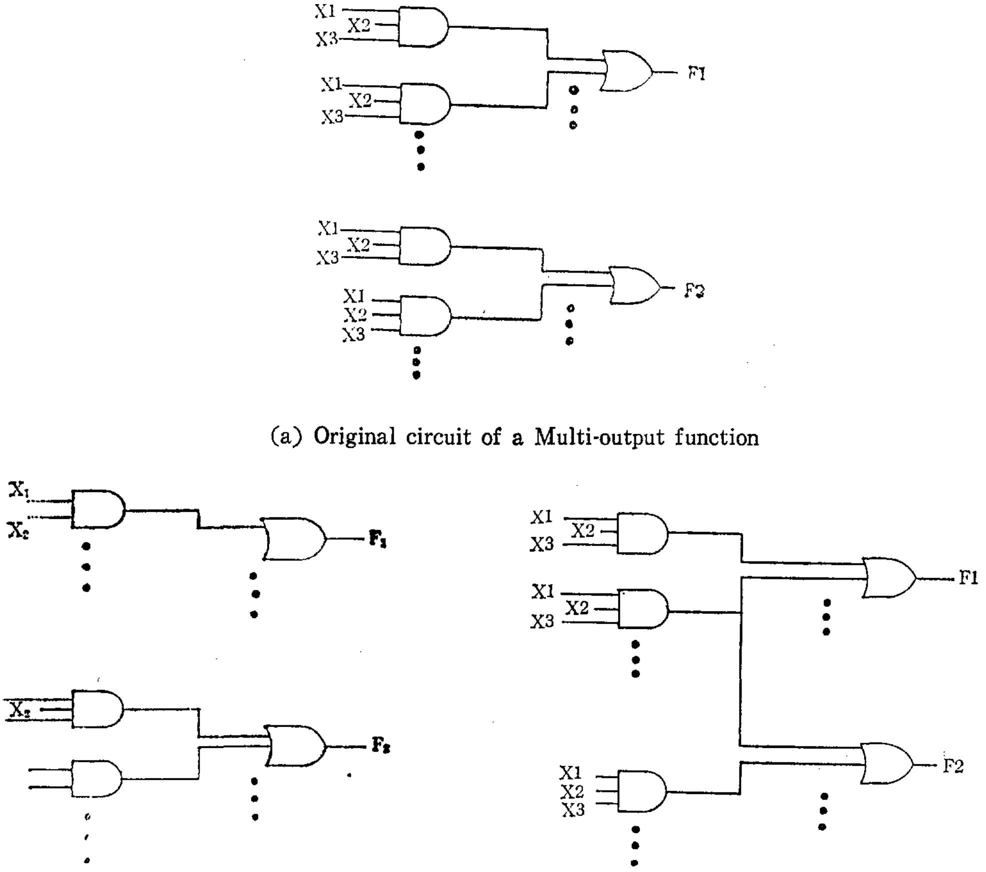


(b) In the case of $F1$. and $F2$. in (a) have 2-cube PI relation

(a) Two different output function

(c) In the case of two functions in (b) have common 2-cube PI

Fig. 1. The reduction of a multi-output function that have 2-cube PI and 2-commonality relation.



(a) Original circuit of a Multi-output function

(b) Reduced circuit when $F1$ of (a) has 1-cube PI.

(c) Reduced circuit when $F1$ and $F2$ of (a) have 0-cube that is common to the two functions.

Fig. 3. An Example of the reduction of multioutput functions that have different hardware, but same cost

Commonality No. of Minterms That can be Reduced to 1 PI	4	3	2	1
8	32	24	16	8
4	16	12	8	4
2	8	6	4	2
1	4	3	2	1

Fig. 2. The Cost Table of a multi-output function that has four outputs and 3-cube relation of the highest.

In the cost table, we can see that if output function $F1$ has a 1-cube PI that is not included in Function $F2$, and the output functions $F1$ and $F2$ have the O-cube PI that is common in two functions, they have differnt hardware realizations, but the costs of the two realized circuits are same.

In other words, 1 AND gate and 1 input can be eliminated in the former case, and only 1 AND gate can be eliminated in the latter case as shown in Fig. 3.

Ⅲ. Generation of Prime Implicants

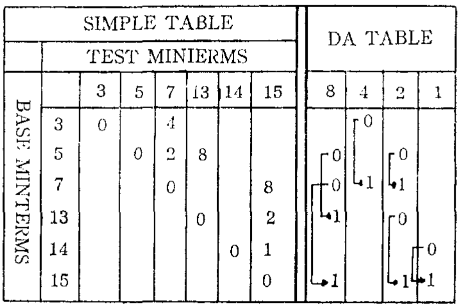
For the generation of PI, the application of the Simple table and DA table [5] to the multi-output function is quite similar to the single output function.

A little difference between the single output funotion and multioutput function is to be in the selection of PI's. In the case of single output function, this procedure is satisfied by only generating PI's, but in the multi-output function the subsets of a PI should be considered, because of the cost consideration. If any subset of a PI has higher commonality than the PI. it should be listed in PI chart.

Let us assume that we wish to find a minimal sum of products realization for three functions represented in Fig. 4 as an example. This example is taken from [4], p152—155.

Minterm	*X*1	*X*2	*X*3	*X*4	*F*1	*F*2	*F*3
0	0	0	0	0	0	0	1
1	0	0	0	1	0	0	0
2	0	0	1	0	0	0	0
3	0	0	1	1	1	0	0
4	0	1	0	0	0	0	1
5	0	1	0	1	1	1	0
6	0	1	1	0	0	0	0
7	0	1	1	1	1	1	0
8	1	0	0	0	0	0	0
9	1	0	0	1	0	0	0
10	1	0	1	0	0	1	1
11	1	0	1	1	0	0	0
12	1	1	0	0	0	0	0
13	1	1	0	1	1	1	0
14	1	1	1	0	1	1	1
15	1	1	1	1	1	1	1

Fig. 4. An Example of truth-table for three output logic network.



SIMPLE TABLE							DA TABLE			
	TEST MINIERMS									
BASE MINTERMS	3	5	7	13	14	15	8	4	2	1
3	0		4					0		
5		0	2	8			0		0	
7			0			8	0	1	1	
13				0		2	1		0	
14					0	1				0
15						0	1		1	1

Fig. 5. The Simple table and the DA table of functionF1 for the generating PI's.

At first, draw the Simple table and DA table for a function *F*1 of the multi-output function Fig. 5.

We can generate all the PI's of *F*1 from the above table. And now all that we must do is to choose the minimum number of PI's that can cover the entire function *F*1. Same procedures for the rest functions *F*2 and *F*3 must be done similary.

Ⅳ. Intersection Table and Selection of the Prime Implicants

As described previously, we can find the minimal set of sum of products by selecting the larger cost PI earlier. To find the commonality of each PI, we draw the Intersection table (as shown in Fig. 6.) of given functions from the truth-table shown in Fig. 4.

Minterms	*F*1	*F*2	*F*3
0			V
3	V		
4			V
5	V	V	
7	V	V	
10		V	V
13	V	V	
14	V	V	V
15	V	V	V

Fig. 6. Intersection table of An Example in Fig. 4.

By locating this Intersection table between DA table and cyclic chart, we can easily know the commonality of the minterms that can be reduced to one minterm(or the selected prime implicants), Now, for the selection of PI's that cover the entire function *F*1 draw the cyclic chart as shown in Fig. 7 as an Example

		15	14	13	7	5	3		Minterms **	
		V	V	V	V	V	V	*F1*	Intersection Table	
		V	V	V	V	V		*F2*		
		V	V					*F3*		
P.I								No. of Minterms covered by PI	Commonality	Cost
PI (1)	5,7,13,15(2,8)	V		V	V	V		4	*F*1, *F*2	6*
PI (2)	3,7(4)				V		V	2	*F*1	2
PI (3)	14,15(1)	V	V					2	*F*1, *F*2, *F*3	6
PI (4)	5,7(2)				V	V		2	*F*1, *F*2	4
PI (5)	13,15(2)	V		V				2	*F*1, *F*2	4
PI (6)	5,13(8)			V		V		2	*F*1, *F*2	4
PI (7)	7,15(8)	V			V			2	*F*1, *F*2	4
PI (8)	3						V	1	*F*1	1
PI (9)	5					V		1	*F*1, *F*2	2
PI(10)	7				V			1	*F*1, *F*2	2
PI(11)	13			V				1	*F*1, *F*2	2
PI(12)	14		V					1	*F*1, *F*2, *F*3	3
PI(13)	15	V						1	*F*1, *F*2, *F*3	2
		V		V	V	V				

*PI chosen in the Cyclic chart.
**Minterms covered by the function *F*1.

Fig. 7. Cyclic chart (first step) for the selection of PI's of F1.

PI's	Minterms	15	14	13	7	5	3	No. of minterms remained	Commonality	Cost **
PI (1)	5,7,13,15(2,8)	V		V	V	V				
PI (2)	3,7(4)				V		V	1	*F*1	1
PI (3)	14,15(1)	V	V					1	*F*1, *F*2, *F*3	3 *
PI (4)	5,7(2)				V	V				
PI (5)	13,15(2)	V		V						
PI (6)	5,13(8)			V		V				
PI (7)	7,15(8)	V			V					
PI (8)	3						V	1	*F*1	1
PI (9)	5					V				
PI(10)	7				V					
PI(11)	13			V						
PI(12)	14		V					1	*F*1, *F*2, *F*3	3
PI(13)	15	V								
		V	V	V	V	V				

*PI chosen in the Cyclic chart. **Minterms covered by the function *F*1.

Fig. 8. Cyclic chart (second step) **for the selection of PI's of F1.** (continued from Fig. 7.)

In this cyclic chart, all the PI's of $F1$ are listed and their cost are also computed from the Cost Table and recorded. From this cyclic chart, at first we choose the largest cost PI and eliminated the corresponding minterms as shown in Fig. 7.

Next, compute the cost again for the remaining PI's and choose the largest PI and eliminate the corresppnding minterms as shown in Fig. 8 This procedure continues until the entire function is covered. The minimized result for a function $F1$ is as follows. $F1=\overline{X}_1X3X4+X1X2X3+X2X4$

Similarly, the minimized results for the $F2$ and $F3$ are obtained as follows:

$$F2=X2X4+X1X3\overline{X}4$$

$$F3=X1X2X3+X1X3\overline{X}4+\overline{X}1\overline{X}3\overline{X}4$$

The realization of the three functions given in Fig. 4 is shown in Fig. 9.

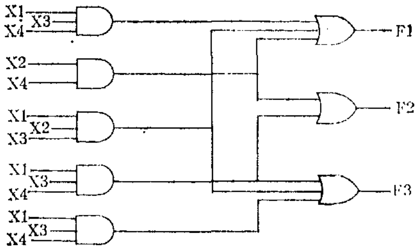


Fig. 9. Reduced multi-output logic network realization.

It is clear that as this procedure treats all the functions and cost numerically and the selection step is also straightforword, this procedure is suitable for the minimization by using the computer. Therefore, this procedure is implemented computer, and the overall flow chart is shown in Fig. 10.

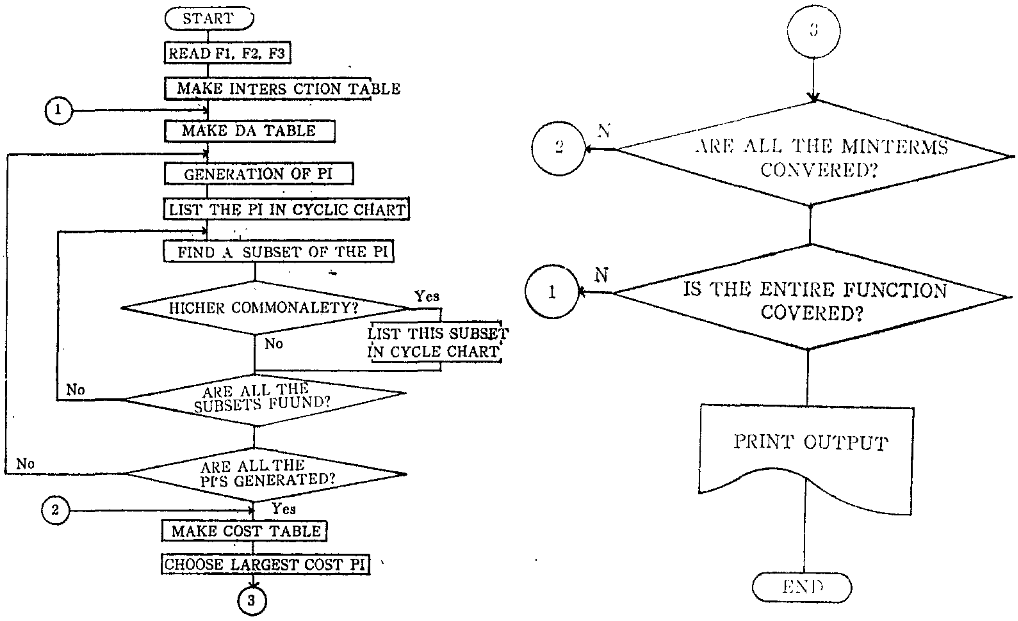


Fig. 10. The overall flow chart for the computer program.

Ⅴ. Conclusion

This paper is concerned with the minimization of the multi-output switching functions by the Intersection Table and the Cost Table presented here. Owing to the Simple algorithm of both generating and selecting the PI's and numerically treated cost, it can be easily programmed on the digital computer. This computer algorithm depends on the constraint given by the cost table, which is useful for the two level (AND/OR) realization regardless of fan-in of gate. But, this disadvantage will overcomes if we use PLA chips.

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

1) Because of the simplicity of table and Intersection Table, the minimization of the multi-output function under the given criterion becomes easier evidently then ealiers works [1]-[4] and [6].
2) This method will be extended to the minimization of the multi-output function under other criteria.
3) In Geneating the multi-output function prime Implicants is obtained easily from the DA table as ealier work [5] done by this author.
4) The experiments with the various examples showed that the intersection table and the cost table method is efficient.