On the Minimization of the Multi-output Switching Function by Using the Intersection Table

Hee Yeung Hwang, Dong Sub Cho, Ho Kyum Kim

COLLEGE OF ENGINEERING, SEOUL NATIONAL UNIVERSITY

Abstract : The optimal selection of Prime Implicants for the multi-output switching function is a difficult task, as the input variables increase. This paper is concerned with the technique for the minimization of the multi-output switching function using the intersection table (5). This procedure is applicable to both manual and computer-programmed realization without complexity.

I. INTRODUCTION

A large number of an established problem in logic design have multi-output switching functions. One of the approaching method would be to design each function separately. For many of the simpler logic networks, this approach is a realistic solution to the problem. But, as the function becomes more complex, we often find that we can share a certain part of the logic network.
This sharing of the network affect the overall cost of the network design, so there are other techniques to share as many logic circuit elements as possible. Goals of each techniques are minimization of switching function with the optimal cost by retaining as much commonality as possible between switching functions. In this paper an application from the simple table and Intersection table is generated.
From the intersection table, the commonality of minterms are found easily and prime implicants are selected optimally.

II. APPLICATION OF THE SIMPLE TABLE METHOD TO MULTI-OUTPUT FUNCTION

The adaptation the Simple Table Method (5) to Multi-Output function is quite similar to the Single out-put function.
Considerable savings can often be achieved by the sharing of hardware among the multi-out-put functions. It is clear that minimum total number of gates are required.
To find the minimal sum of product realization for each function, at first draw the intersection table (as shown in Fig. 1.) of a given functions.
Then locate this intersection table between the Simple Table and the Check List of each function for corresponding column minterms.
Next make the check list, just as in the single output Simple Table Method. Only differnences in that are (1) we select the common letter from the intersection table

in checking minterms which are reduced to one prime implicant and write in corresponding row and (2) in selecting prime implicants. The following procedure is used to guarantee an optimum realization (minimum total cost).

Step 1. Select the prime implicants group containing the same number of the common letters.

Step 2. Obtain the prime implicants from the selected group just as in the single output function.

Step 3. If all the minterms are covered, go to Step 5. Otherwise go to Step 4.

Step 4. Select the next group which has the lower common letters, and go to step 2.

Step 5. Find the dominated prime implicants and eliminate them.

Step 6. Function realization.

An example for this case is solved as follows.
This example is taken from (1), pp. 161–166.
For a given multiple output function, first the intersection table is constructed as in Fig. 1. and then from this intersection table we can obtain each another intesecti table that inserted between the Simple Table and the Check List.
The Simple Table and the intersection table and the Check List are given Fig. 2–Fig. 4 for each function.
Using this Step we can select the minimal sum of products for each function.

REFERENCES

(1) F.J. Hill and G.R. Peterson, Introduction to switching theory and Logical design, Wiley, New York, 1974, pp. 97–174.

(2) Taylor L. Booth, Digital Network and Computer Systems, Wiley, New York, 1971, pp. 93–157.

(3) John B. Peatman, The Design of Digital System, McGraw Hill, New York, 1972, pp. 56–116.

(4) V.T. Rhyne, Fundamentals of Digital Systems Design, Englewood Cliffs, NJ: Prentic-Hall, 1973, pp. 163–165.

(5) Hee Yeung Hwang, "A New Approach to the Minimization of Switching Function by the Simple Table Method," 대한전기학회지, 제 28권 제 6호 pp. 61–77, 1979.

Output Functions		Minterms								
	F	1	2	3	4	5	10	11	12	13
	a		a		a		a	a	a	a
	F_b				b	b	b	b		b
	F_c	c	c	c			c	c	c	

Fa=$\sum$m(2, 4, 10, 11, 12, 13)
Fb=$\sum$m(4, 5, 10, 11, 13)
Fc=$\sum$m(1, 2, 3, 10, 11, 12)

Fig. 1 The Intersection Table.

	2	4	10	11	12	13	
2	0	2	5	9	10	11	
4		0	6	7	8	9	
10			0	1	2	3	
11				0	1	2	
12					0	1	
13						0	
F_a	a	a	a	a	a	a	
F_b		b	b	b		b	
F_c	c		c	c	c		
10(8)	▼		▼				ac *
12(8)		▼			▼		a
11(1)			▼	▼			abc *
13(1)					▼	▼	a *
						▼	ab
					▼		ac
				▼			abc
			▼				abc
		▼					ab *
	▼						ac
	▼	▼	▼	▼	▼	▼	

$Fa=X_2X_3X_4+X_1X_2X_3+X_1X_2X_3+X_1X_2X_3X_4$

Fig. 2 The Multi-output function.

	4	5	10	11	13	
4	0	1	6	7	9	
5		0	5	6	8	
10			0	1	3	
11				0	2	
13					0	
F_a	a		a	a	a	
F_b	b	b	b	b	b	
F_c			c	c		
5(1)	▼	▼				b
11(1)			▼	▼		abc *
13(8)		▼			▼	b *
					▼	ab
				▼		abc
			▼			abc
		▼				b
	▼					ab *
	▼	▼	▼	▼	▼	

$Fb=X_1X_2X_3+X_2X_3X_4+X_1X_2X_3X_4$

Fig. 3 The Multi-output function

	1	2	3	10	11	12	
1	0	1	2	9	10	11	
2		0	1	8	9	10	
3			0	7	8	9	
10				0	1	2	
11					0	1	
12						0	
Fa		a		a	a	a	
Fb				b	b		
Fc	c	c	c	c	c	c	
11(8,1)		▼	▼	▼	▼		c
11(1)				▼	▼		abc *
3(2)	▼		▼				c *
3(1)		▼	▼				c
10(8)		▼		▼			ac *
12						▼	ab *
11					▼		abc
10				▼			abc
3			▼				c
2		▼					ac
1	▼						c

$Fc=X_1X_2X_3+X_1X_2X_4+X_2X_3X_4+X_1X_2X_3X_4$

Fig. 4 The Multi-output function.