

Input Code 를 指定하기 위한 Davis 方法의 算法

An algorithm in Davis method for assigning input codes

論 文

18-3-2

梁 興 錫* 方 勝 楊**
(Yang Heung Suk, Bang Sung Yang)

[Abstract]

A combinational circuit is one of the fundamental elements in digital circuits, containing electronic computers. W.A. Davis presented a new approach to the problem of assigning binary input codes so that the combinational circuits necessary to realize the given output functions may be minimized.

This paper has derived an algorithm for determining the actual cost of a given assignment scheme in Davis method. First, the combinations of partitions realizing the output functions are obtained and then those among them implementing all of the output functions are found by using a table.

In consequence it has been proved that by this algorithm the process of the assignment in a scheme with 16 states and 3 determined partitions can be reduced to $\frac{1}{10^6}$.

1. 緒 論

지나간 20년 동안 電氣工學分野에 있어서의 많은, 重要한 發展中에서 電子計算機만큼 科學的으로나 公共的으로 重要性을 지닌것은 없었을 것이다. 軍事, 科學計算 企業內的 會計問題 또 最近에 이르러서는 工場機械 프로세스 制御에 對한 이 電子計算機의 應用은 技術者가 연달아 効率的으로 그들의 想像을 應用하는가를 證明해 왔다. 아마 다음 10년 동안에는 digital 技術이 巨大하고 活用性이 큰 data-processing system 뿐만 아니라 手工으로 하던 너무나 힘든 일을 다루는 작고 限定된 目的을 가진 機器까지에도 보다 더 널리 利用될 것을 보게 될 것이다.

Switching circuit란 logical function을 遂行하는 回路를 말한다. switching circuit 設計를 logical design이라 한다. switching circuit의 複雜性을 보면 upstairs와 down-

stairs로 制御되는 簡單한 staircase lighting circuit로부터 電子計算機內에서 計算을 행하는 複雜한 回路까지 여러가지가 있다.

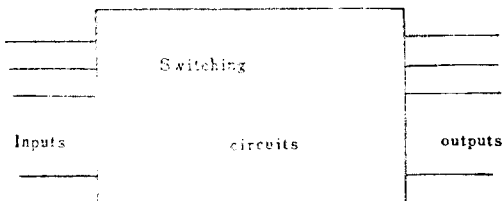
output 狀態가 언제나 그래프의 input 狀態에만 依存해서 決定되는 그림 1과 같은 回路를 combinational circuit라 한다. (1)(2)

表 1
Table 1

Number of Possible Assignments

No. of States q	No. of Variable n	No. of QBP'S	No. of Assignments
1	0	—	—
2	1	1	1
3	2	3	3
4	2	3	3
5	3	15	140
6	3	25	420
7	3	35	840
8	3	35	840
9	4	255	10,810,800
10	4	501	.
11	4	957	.
12	4	1,749	.
13	4	3,003	.
14	4	4,719	.
15	4	6,435	.
16	4	6,435	$\approx 5 \times 10^{10}$
17	5	65,535	$\approx 10^{20}$
q	$n = \lceil \log_2 q \rceil$	$\sum_{i=q-1-2^{n-1}}^{2^{n-1}-1} (q-i)$	$\frac{(2^n-1)!}{(2^n-q)!n!}$

그림 1 (Fig 1)



* 正會員: 서울大學工科大学電氣工學科 教授
* 正會員: " " 助教

주어진 output function을 實現하기 위한 combinational circuit가 最小가 되도록 各 狀態에 binary input code를 指定하는 問題가 Dolotta와 Mc Cluskey(3) 그리고 Dolotta(4)에 의하여 研究 되었다.

表 1에 表示된 것과 같이 assignment 數와 QBP의 數까지도 4 以上の 狀態에서는 아주 커지므로 實際問題에서 高速度計算機를 가지고서도 모든 可能한 assignment를 考慮한다는 것은 不可能하게 되고 만다.

最近 Davis氏는 주어진 output function을 實現하기 위한 combinational circuit가 最小가 되도록 binary input code를 指定하는 問題에 對한 새로운 手法을 提示하였다(5).

이 方法은 binary partition과 set system에 基礎를 두고 있다. 주어진 output function을 實現하는데 必要한 gate數 diode數가 最小이라는 뜻으로 最適의 assignment를 만들어낸다. 그러나 이 方法을 完成시키기 위하여서는 다음의 3가지 問題를 解決해야 한다고 Davis氏가 指摘하고 있다.

- 1) 最小實現數를 얻기위한 nonenumerative한 方法.
 - 2) 주어진 assignment scheme에 있어서 實際實現費用을 計算하는 算法
 - 3) 큰 output table을 取扱하는 computer program
- 이 論文에서는 Davis 方法에 對한 몇가지 言及과 더불어 주어진 assignment scheme에 있어서의 實際實現費用을 決定하기위한 한 算法이 提示되고 論議된다.

2. 算法의 誘導

이 算法은 原理的으로는 定理 1에 依存한다.

定理 1: q개의 狀態를 갖는 集合 S에 對해서 n개의 binary 變數 y_1, y_2, \dots, y_n 를 갖는 assignment가 有効하기 위한 必要充分條件은 만들어질 n개의 BP가 mutually consistent 되도록 各 y_i 가 S에 있어서 各 binary partition(B P)를 規定하는 것이다.

이 證明은 Brzozowski와 Davis의 文獻에 있다(7).

補助定理 1, 만일 P_{i+1} 이 決定해야 할 BP이고 P_1, P_2, \dots, P_i 이 決定된 partition이라면 $P_1 P_2 \dots P_i P_{i+1}$ 은 mutually consistent 이다.

補助定理 2, 만일 $P_{i+1}, P_{i+2}, \dots, P_n$ 이 나머지 決定해야 할 BP이고 P_1, P_2, \dots, P_i 이 決定된 partition이라면 $P_1 P_2 \dots P_n = 0$.

이들 두 補助定理는 定理 1에서 明確하다.

따라서 補助定理를 滿足하고 補助定理 1로 연결되는 partition 組合란이 可能性이 있음을 알 수 있다.

그런데 어떤 output function은 最小實現數 $m(f)$ 로 實現할 수 있다.

補助定理 3, P_1, P_2, \dots, P_j 는 決定된 partition 이고 $P_1 P_2$

$P_j = \{b_{j1}; b_{j2}; \dots; b_{jk}\}$ 그리고 $p_{j+1}, p_{j+2}, \dots, p_m$ 은 決定된 partition 이라고하면 $f_n = b_{ji} (i=1, 2, \dots, k)$ 인 output function f_n 는 決定된 $p_{j+1}, p_{j+2}, \dots, p_m$ 에 不拘하고 언제든지 $m(f_n)$ 로 實現 할 수 있다.

가령 $q=15, n=4$ 인 output table을 생각해 본다. $p_{21} p_{22}$

表 2
Table II

$g(f)$ and $d(f)$ for $q=3, 4, \dots, 16$

q	n	#(b)	#(b)	d(f)	g(f)	q	n	#(b)	#(b)	d(f)	g(f)
3	2	1	2	2	1	13	4	4	8	2	1
		5	7	4	2						
		6	6	4	2						
4	2	1	3	2	1	14	4	1	12	2	1
		2	2	6	3						
		3	3	4	2						
5	3	1	5	2	1	15	4	1	14	3	1
		2	4	2	1						
		3	3	4	2						
7	3	1	6	2	1	16	4	1	15	4	1
		2	5	2	1						
		3	4	4	2						
8	3	1	7	3	1	11	4	2	14	3	1
		2	6	2	1						
		3	5	4	2						
9	4	1	8	2	1	12	4	2	10	4	2
		2	7	2	1						
		3	6	2	1						
10	4	1	9	2	1	13	4	3	12	2	1
		2	8	2	1						
		3	7	2	1						
11	4	1	10	2	1	14	4	4	11	2	1
		2	9	2	1						
		3	8	2	1						
12	4	1	11	2	1	15	4	5	11	7	3
		2	10	2	1						
		3	9	2	1						

= $\{b_{21}; b_{22}; b_{23}; b_{24}\}$ 인 決定된 2 partition P_{21}, P_{22} 를 가진 scheme에 있어서 (b_{2i})는 다만 3 또는 4 狀態 만을 包含한다. ($i=1, 2, \dots, 4$). 한편 $q=15$ 일 때 $\#(b)=3$ 또는 4를 實現하는데 必要한 $d(f), g(f)$ 는 表 2로부터 各 各 2, 1이다. 따라서 만일 $f_i = b_i (i=1, 2, 3, 4)$ 인 어떤 한 output function f_j 를 實現 할 수 있다면 앞으로 決定해야 할 2 partition 에도 不拘하고 그러한 f_j 는 언제든지 2 diode 1 gate 로 實現 할 수 있다. 또 $p_{31} p_{32} p_{33} = \{b_{31}; b_{32}; \dots; b_{38}\}$ 인 決定된 partition p_{31}, p_{32}, p_{33} 를 가진 scheme에 있어서 各 block은 1 또는 2 狀態만을 包含한다. 이 경우 表 3에서 $\#(b)=1, 2$ 인 $d(f), g(f)$ 는 各 各 3, 1이다. 이 補助定理는 q가 3으로부터 16까지의 모든 경우에 있어서 適用 할 수 있음을 알았다. 그러므로 그러한 output function은 그 scheme에 있어서 檢討할 必要가 없는 것이다.

output function을 實現하기 위해서는 두가지 方法이 있다. 하나는 그 function의 1을 實現하는 것이며 또 하나는 0단을 實現하는 것이다. 이때 complementation에는 費用이 안든다는 條件이 있다. 그리고 또 output function을 實現하는데 두가지의 2-level logic circuit가 있다. 그 하나는 AND-OR(minterm) 型에 의한 것이고 또 하나는 OR-AND(maxterm) 型에 의한 것이다

補助定理 4, 만일 한 output function의 1들이 몇개의 diode와 gate로 어느쪽 한型에 의하여 實現 할 수 있다면 그 output function의 0들은 또 하나의 型으로 그것도 똑같은 數의 diode, gate로 實現 할 수 있다.

證明, duality 原理로부터 明確하다(11).

補助定理 4는 어떤 output function을 네번 檢討 할 必要는 없고 그 output function의 1과 0을 한型으로 또는 1이나 0 어느쪽 하나를 두型으로 檢討하면 充分함을 말한다. 나아가서는 1과 0을 어느 한가지 型으로 檢討하는 것이 便利함을 곧 알 수 있다. 여기서는 AND-OR 型을 採用한다. 그리고 $n=2, 3$ 일때의 각 $d(f), g(f)$ 에 對한 實現 form이 表 3에 주어져 있다. 表 3을 만들

表 3
Table III
AND-OR Logics for $n=2$ and 3

n	$d(f)$	$g(f)$	logic
2	2	1	$\bar{b}_i + \bar{b}_j$
	4	2	$b_i + b_j \bar{b}_j$
	6	3	$b_i b_j + \bar{b}_i \bar{b}_j$
3	2	1	$\bar{b}_i + \bar{b}_j, \bar{b}_i \bar{b}_j$
	3	1	$\bar{b}_i + \bar{b}_j + \bar{b}_k, \bar{b}_i \bar{b}_j \bar{b}_k$
	4	2	$\bar{b}_i + b_j \bar{b}_k$
	6	3	$b_i \bar{b}_j + \bar{b}_i \bar{b}_k, b_i b_j + \bar{b}_i \bar{b}_j$
	7	3	$b_i b_j + \bar{b}_i \bar{b}_j \bar{b}_k, b_i + b_j b_k + \bar{b}_i \bar{b}_k$
	8	3	$b_i b_j b_k + \bar{b}_i \bar{b}_j \bar{b}_k, b_i b_j b_k + \bar{b}_i \bar{b}_j b_k, b_i b_j b_k + b_i \bar{b}_j \bar{b}_k$
	9	4	$b_i b_j + \bar{b}_i \bar{b}_k + \bar{b}_j \bar{b}_k, b_i b_j + b_j b_k + \bar{b}_i \bar{b}_j, b_i b_j + b_j b_k + b_i \bar{b}_k$
	10	4	$b_i b_j b_k + \bar{b}_i \bar{b}_k + \bar{b}_j \bar{b}_k, b_i \bar{b}_j \bar{b}_k + \bar{b}_i \bar{b}_j + \bar{b}_j \bar{b}_k$
13	5	$b_i b_j b_k + \bar{b}_i \bar{b}_j + \bar{b}_i \bar{b}_k + \bar{b}_j \bar{b}_k$	
16	5	$b_i b_j b_k + b_i \bar{b}_j \bar{b}_k + \bar{b}_i b_j \bar{b}_k + \bar{b}_i \bar{b}_j b_k$	

where b_x implies \bar{b}_x and \bar{b}_x does the complement of b_x .
기 위하여서는 다음의 補助定理가 도움이 된다.

補助定理 5. 만일 output function이

$$\bar{b}_z = \bar{b}_{111} + \bar{b}_{112} + \dots + \bar{b}_{11j} + \bar{b}_{211} \bar{b}_{221} + \dots + \bar{b}_{21k} \bar{b}_{22k}$$

$$+ \bar{b}_{311} \bar{b}_{321} \bar{b}_{331} + \dots + \bar{b}_{31l} \bar{b}_{32l} \bar{b}_{33l} + \dots$$

$$+ \bar{b}_{w1y} \bar{b}_{w2y} \dots \bar{b}_{wxy}$$

로 表現되고 그때 diode 數가 最小이면

$$\bar{b}_{11i} \neq \bar{b}_{rst} \quad i=1, 2, \dots, j$$

$$r=1, 2, \dots, w$$

$$s=1, 2, \dots, x$$

$$t=1, 2, \dots, y$$

證明 위의 實現에서 $\bar{b}_{11i} = \bar{b}_{rst}$ 이면

$$\bar{b}_z = \bar{b}_{111} + \bar{b}_{112} + \dots + \bar{b}_{11(i-1)} + \bar{b}_{11(i+1)} + \dots + \bar{b}_{11j}$$

$$+ \bar{b}_{211} \bar{b}_{221} + \dots + \bar{b}_{21k} \bar{b}_{22k} + \bar{b}_{311} \bar{b}_{321} \bar{b}_{331} + \dots$$

$$+ \bar{b}_{31l} \bar{b}_{32l} \bar{b}_{33l} + \dots + \bar{b}_{w1y} \bar{b}_{w2y} \dots \bar{b}_{wxy}$$

$$+ \bar{b}_{11i} (1 + \bar{b}_{r1} \bar{b}_{r2} \dots \bar{b}_{r(s-1)} \bar{b}_{r(s+1)} \dots \bar{b}_{rt})$$

따라서 最後項은 \bar{b}_{11i} 가 되고 그 結果는 이 實現이 最小이라는 假定에 어긋난다.

그러므로

$$\bar{b}_{11i} \neq \bar{b}_{rst}$$

補助定理 6. b_n 이 어떤 output function의 한 表現이지만 決定된 partition과는 다르고 b_1, b_2, \dots, b_m 은 BP 이라고하고 만일 \bar{b}_n 이 $\bar{b}_1, \bar{b}_2, \dots, \bar{b}_m$ 를 갖고 AND-OR 型으로 實現할 수 있다면

$$\bar{b}_n \supset \bar{b}_i \quad i=1, 2,$$

證明 b_n 이 $\bar{b}_1, \bar{b}_2, \dots, \bar{b}_n$ 를 가지고 다음과 같이 實現된 다면

$$\bar{b}_n = \bar{b}_{111} + \bar{b}_{112} + \dots + \bar{b}_{11j} + \bar{b}_{211} \bar{b}_{221} + \dots + \bar{b}_{21k} \bar{b}_{22k}$$

$$+ \bar{b}_{311} \bar{b}_{321} \bar{b}_{331} + \dots + \bar{b}_{31l} \bar{b}_{32l} \bar{b}_{33l} + \dots + \bar{b}_{w1y} \bar{b}_{w2y}$$

$$\dots \bar{b}_{wzy}$$

그 때

$$\bar{b}_n \supset \bar{b}_{r1} \bar{b}_{r2} \dots \bar{b}_{rut}$$

여기서 等記號는 처음부터 $\bar{b}_{r1} \bar{b}_{r2} \dots \bar{b}_{rut}$ 項만일 때 滿足된다.

$$\therefore \bar{b}_n \supset \bar{b}_{rst} \quad s=1, 2, \dots, u$$

그러나 one literal로 實現되는 \bar{b}_n 은 없으므로 等記號는 없어야 한다.

$$\therefore \bar{b}_n \supset \bar{b}_{rst}$$

3. 算 法

過程에 있어서 表 6과 같은 表方法이 採用된다.

step 1. 그 scheme에 있어서 p_1, p_2, \dots, p_j 를 決定된 partition $p_{j+1}, p_{j+2}, \dots, p_m$ 을 決定될 partition이라고하면 먼저 $p_1 p_2 \dots p_j p_{j+1, k}$ 가 mutually consistent 되는 $p_{j+1, 1}, p_{j+1, 2}, \dots, p_{j+1, r}$ 를 찾아내고 다음에 각 $p_{j+1, k}$ 에 대해서 $p_1 p_2 \dots p_j p_{j+1, k} p_{j+2, l}$ 이 mutually consistent 되는 $p_{j+2, k1}, p_{j+2, k2}, \dots, p_{j+2, kt}$ 를 골라낸다. 그러한 式으로 p_m 까지 계속한다. 이렇게 해서 $p_1 p_2 \dots p_m = 0$ 이 되는 $p_{j+1}, p_{j+2}, \dots, p_m$ 에 대한 全組合을 찾아낸다.

Step 2. 決定된 partition의 b 와 \bar{b} 그리고 Step 1에서 얻어진 組合을 表의 最上行에 넣고 決定된 partition의 \bar{b} 와 各組合의 \bar{b} 를 組合하고 表 3에 나타나 있는 것中에서 다음 Scheme 보다 크지않은 $d(f), g(f)$ 에 該當

하는 全 minterm 을 만든다. 그 minterm 을 各該當 組合 옆에 넣는다.

便宜上 決定된 partition 만으로 이루어진 column 群을 0 다음 組合에 對應하는 column 群을 1……式으로 이름 부르기로 한다.

step 3. 實現해야 할 \bar{b}_i 를 첫 column 에 넣고 \bar{b}_i 에 包含되는 column 을 點檢한다.

step 4. 點檢한 그 column 中에서 表 3 에 나타나있는 AND-OR logic 으로 實現 할 수 있도록 column 의 組合를 찾아내고 그때 使用된 column 의 群番號를 記入한다. 但 0와 다른 番號의 column 으로 이루어졌을 때에는 0 아닌 番號를 記入하고 0에 該當하는 column 만으로 이루어졌을 때에만 0를 記入한다.

step 5. 이러한 方法으로 全 function 을 檢討한 후 全行에 나타난 番號에 該當하는 partition 이 assignment 를 決定하고 同時에 그 assignment 에 의한 實際實現費用이 얼어진다.

表 4
Table IV
Example 1

S	f_1	f_2	f_3	f_4
A	0	1	0	1
B	1	0	1	0
C	1	1	1	0
D	1	0	1	1
E	0	1	0	0
F	0	1	1	0
G	0	0	0	0

表 5
Table V
 $g(f)$ and $d(f)$ for Example 1

f	f_1	f_2	f_3	f_4
$g(f)$	2	2	2	1
$d(f)$	4	4	4	2

4. 例

具體的으로 例를 들어 보면 明確해 질 것이다. 表 4⁽³⁾ 가 이 算法으로 檢討되고 算出結果는 表 6 에 주어진다.

step 1. 이 경우 決定해야 할 partition 은 하나이다. 따라서 $p_2 p_3 p_5 = 0$ 를 滿足하는 p_5 로는 8 set $\{\overline{A}, \overline{B}, \overline{C}, \overline{D}, \overline{E}, \overline{F}, \overline{G}\}, \{A, B, C, G; D, E, F\} \dots \{A, D, F, G; \overline{B}, \overline{C}, \overline{E}\}$ 가 있다.

step 2. 決定된 partition p_2, p_3 의 \bar{b} 들과 8 set 의 \bar{b} 들을 表 最上行에 넣고 決定된 partition 으로 주어진 \bar{b}

群을 0, 各 set \bar{b} 群을 1, 2, ..., 8 群이라고 이름 부른다.

그런데 이 경우 diode 數가 8가지의 表 3 에 表示되어 있는 모든 AND-OR logic 즉 1, 2, 3 literal 까지의 minterm 을 考慮해야한다.

왜냐하면

$$m \text{ (다음 scheme)} - d(f_4) = 10 - 2 = 8$$

0 群안의 \bar{b} 로 構成된 2, 3 literal 의 minterm 즉 $b_2 b_3 = (CF)$ $b_2 \bar{b}_3 = (AE)$ $\bar{b}_2 \bar{b}_3 = (G)$ $\bar{b}_2 b_3 = (BD)$ 를 0 群에 追加한다. 또 0 群 1 群 안의 b 로 構成된 2, 3 literal minterm 을 1 群에 追加한다. 이러한 方法으로 8 群까지 繼續한다.

step 3. 實現해야할 output function f_1 의 1 즉 $b_1 = (BCD)$ 를 包含하는 $b_1 \supset \bar{b}_i$ 인 全 \bar{b}_i 를 찾아서 點檢한다. 가령 0 群안에서는 그러한 column 은 $\bar{b}_2 b_3 = (BD)$ 하나 뿐이다.

step 4. 먼저 表 3 에 있는 4 diode AND-OR logic $\bar{b}_i + \bar{b}_j \bar{b}_k$ 를 檢討해 보면 이 logic 으로서는 check 가 있는 column 을 어떻게 組合하더라도 (BCD)는 얻어지지 않는다.

step 5. 그러나 f_1 의 0 즉 $\bar{b}_1 = (AEFG)$ 는 이 logic 으로 8 가지 方法으로 實現할 수 있다. 그것들은 表에서 1, 2, ..., 8 의 數字로 表示되어 있다.

다음에 나머지 output function f_4 의 b_4, \bar{b}_4 을 같은 方法으로 檢討해 간다. 이 양쪽은 $d(f_4) = 2, 3$ 으로서도 그리고 b_4 는 $d(f_4) = 4$ 으로서도 實現할 수 없지만 \bar{b}_4 는 $d(f) = 4$ 으로서 實現할 수 있고 表에서는 5, 7로 表示되어 있다.

step 6. 結局 表에서 5와 7이 $f_1 f_4$ 양쪽에 나타났으므로 더 以上의 調査는 必要 없는 것이다. 實際費用은 表로부터 自動的으로 計算되고 이 경우 5, 7 어느쪽에 의하여도 8 diode 4 gate 가 必要함을 알 수 있다. 또 assignment 와 그 實現은 5에 의하면

$$p_5 = \{A, C, D; B, E, F, G\}$$

$$\bar{b}_1 = \bar{b}_3 + b_2 \bar{b}_5$$

$$\bar{b}_4 = \bar{b}_5 + b_2 b_3$$

7에 의하면

$$p_5 = \{A, D, F; \overline{B}, \overline{C}, \overline{E}, G\}$$

$$\bar{b}_1 = \bar{b}_3 + b_2 b_5$$

$$\bar{b}_4 = \bar{b}_5 + b_2 b_3$$

으로 된다.

5. 檢 討

1. Davis 方法은 2-level unshared logic 을 使用하는 實用性에 있어서 不利한 前提를 갖고 있다. output table 이 2-level shared logic 을 使用하면 얼마나 經濟的으로 實現되느냐에 관해서는 McCluskey 의 文獻에 나타나 있

3																
Name of Group minterms	b_5 ABF CDEF	b_2b_5 AF	b_3b_5 B	b_3b_5 BF	b_3b_5 A	b_2b_5 CE	b_2b_5 DG	b_2b_5 CD	b_2b_5 EG	$b_2b_5b_5$ F	$b_2b_5b_5$ B	$b_2b_5b_5$ A	$b_2b_5b_5$ C	$b_2b_5b_5$ E	$b_2b_5b_5$ D	$b_2b_5b_5$ G
$b_1(BCD) \supset$		V						V			V		V			V
$d(f)=4$																
$b_1(AEFG) \supset$			V						V		V		V			V
$d(f)=4$	1,2 3,4 5,6 7,8	3														
$\bar{b}_1 + \bar{b}_2 \bar{b}_k$																
$b_1(AD) \supset$					V							V				V
$d(f)=2$																
$d(f)=3$																
$d(f)=4$			V	V	V				V		V		V			V
$b_1(BCEFG) \supset$																
$d(f)=2$																
$d(f)=3$																
$d(f)=4$	5 7															
4																
Name of Group minterms	b_5 ABFG CDE	b_2b_5 AF	b_2b_5 BG	b_2b_5 BF	b_2b_5 AG	b_2b_5 CE	b_2b_5 D	b_2b_5 CD	b_2b_5 E	$b_2b_5b_5$ F	$b_2b_5b_5$ B	$b_2b_5b_5$ A	$b_2b_5b_5$ C	$b_2b_5b_5$ G	$b_2b_5b_5$ E	$b_2b_5b_5$ D
$b_1(BCD) \supset$							V	V			V		V			V
$d(f)=4$																
$\bar{b}_1(AEFG) \supset$					V				V		V		V			V
$d(f)=4$	1,2 3,4 5,6 7,8															
$\bar{b}_1 + \bar{b}_2 \bar{b}_k$																
$b_1(AD) \supset$							V				V					V
$d(f)=2$																
$d(f)=3$																
$d(f)=4$	5 7		V	V	V				V		V		V			V
$b_1(BCEFG) \supset$																
$d(f)=2$																
$d(f)=3$																
$d(f)=4$	5 7															

		5															
Name of Group	minterms	b_5 ACD BEFG	$b_5\bar{b}_5$ AC	$b_5\bar{b}_5$ D	$b_5\bar{b}_5$ CD	$b_5\bar{b}_5$ A	$b_5\bar{b}_5$ EF	$b_5\bar{b}_5$ BG	$b_5\bar{b}_5$ BF	$b_5\bar{b}_5$ EG	$b_5\bar{b}_5$ C	$b_5\bar{b}_5$ D	$b_5\bar{b}_5$ A	$b_5\bar{b}_5$ F	$b_5\bar{b}_5$ E	$b_5\bar{b}_5$ B	$b_5\bar{b}_5$ G
$b_1(BCD) \supset$			V	V							V	V					
$d(f) = 4$				V	V					V		V	V	V	V	V	V
$\bar{b}_1(AEFG) \supset$																	
$d(f) = 4$	1.2 3.4 5.6 7.8						5										
$\bar{b}_1 + \bar{b}_1\bar{b}_k$																	
$b_1(AD) \supset$			V	V							V	V					
$d(f) = 2$																	
$d(f) = 3$																	
$d(f) = 4$																	
$b_1(BCEFG) \supset$				V	V	V	V	V	V	V	V	V	V	V	V	V	V
$d(f) = 2$																	
$d(f) = 3$																	
$d(f) = 4$	5 7	5															
		6															
Name of Group	minterms	b_5 ACDG BEF	$b_5\bar{b}_5$ AC	$b_5\bar{b}_5$ DG	$b_5\bar{b}_5$ CD	$b_5\bar{b}_5$ AG	$b_5\bar{b}_5$ EF	$b_5\bar{b}_5$ B	$b_5\bar{b}_5$ BF	$b_5\bar{b}_5$ E	$b_5\bar{b}_5$ C	$b_5\bar{b}_5$ D	$b_5\bar{b}_5$ A	$b_5\bar{b}_5$ F	$b_5\bar{b}_5$ G	$b_5\bar{b}_5$ B	$b_5\bar{b}_5$ E
$b_1(BCD) \supset$			V	V							V	V					
$d(f) = 4$																	
$\bar{b}_1(AEFG) \supset$										V		V	V	V	V	V	V
$d(f) = 4$	1.2 3.4 5.6 7.8						6										
$\bar{b}_1 + \bar{b}_1\bar{b}_k$																	
$b_1(AD) \supset$																	
$d(f) = 2$																	
$d(f) = 3$																	
$d(f) = 4$																	
$\bar{b}_1(BCEFG) \subset$																	
$d(f) = 2$																	
$d(f) = 3$																	
$d(f) = 4$	5 7	5															

		7															
Name of Group	minterms	b_5 ADF BCEG	b_4b_5 AF	b_3b_5 D	b_2b_5 DF	b_1b_5 A	b_3b_5 CE	b_2b_5 BG	b_1b_5 BC	b_1b_5 EG	$b_2b_5b_5$ F	$b_3b_5b_5$ D	$b_4b_5b_5$ A	$b_5b_5b_5$ C	$b_2b_5b_5$ E	$b_3b_5b_5$ B	$b_4b_5b_5$ G
$b_1(BCD) \supset$			V						V		V	V	V				V
$d(f)=4$			V		V				V		V	V	V				V
$b_1(AEFG) \supset$																	
$d(f)=4$	1,2 3,4 5,6 7,8		7									V	V				
$b_1+ b_j b_k$				V													
$b_1(AD) \supset$						V											
$d(f)=2$																	
$d(f)=3$																	
$d(f)=4$							V	V	V	V	V	V	V	V	V	V	V
$b_1(BCEFG) \supset$		V															
$d(f)=2$																	
$d(f)=3$																	
$d(f)=4$	5 7	5															
		8															
Name of Group	minterms	b_5 ADFG BCE	b_4b_5 AF	b_3b_5 DG	b_2b_5 DF	b_1b_5 AG	b_3b_5 CE	b_2b_5 B	b_1b_5 BC	b_1b_5 E	$b_2b_5b_5$ F	$b_3b_5b_5$ D	$b_4b_5b_5$ A	$b_5b_5b_5$ C	$b_2b_5b_5$ G	$b_3b_5b_5$ E	$b_4b_5b_5$ B
$b_1(BCD) \supset$										V	V	V	V	V			V
$d(f)=4$																	
$b_1(ABCG) \supset$			V			V				V	V	V	V	V	V	V	V
$d(f)=4$	1,2 3,4 5,6 7,8																
$b_1+ b_j b_k$																	
$b_1(AD) \supset$																	
$d(f)=2$																	
$d(f)=3$																	
$d(f)=4$		V					V	V	V	V	V	V	V	V	V	V	V
$b_1(BCEFG) \supset$																	
$d(f)=2$																	
$d(f)=3$																	
$d(f)=4$	5 7	5															

다(12). 그러므로 이 點을 改善할 수만 있다면 주어진 output function을 實現하기 위해서 必要로하는 gate, diode數가 最小라는 點만 注意할 必要가 없다.

2. 決定된 partition數가 같고 또 $m(S)$ 가 같은 scheme間的 順序問題는 아주 重要하고 어려운 것 같다. 왜냐하면 이 算法에 있어서는 scheme마다 다른 表가 必要하게 되기 때문이다. 이 問題에 對해서도 다만 다음과 같은 추측을 할 수 있다. 즉 確率로 보아서 各 變數의 state數가 均等하게 分配되어 있는 partition을 많이 갖는 scheme이 보다 可能性이 있다. 왜냐하면 partition이 state를 均等하게 나누면 그만큼 算法의 群數가 증가되고 따라서 똑같은 $m(S)$ 로서 보다 많은 output function을 實現할 수 있는 까닭이다.

3. 結局 이 算法은 따져보면 基本的으로는 모든 QBP中에서 모든 QBP의 qualified set를 생각하여야 하므로 별다른 idea는 없는 것이다. 다만 scheme의 決定된 partition數가 커지면 커질수록 考慮對象이 될 QBP數가 적어진다는 것 뿐이다. 故로 이 algorithm은 많은 output function이 있는 큰 output table, 그것도 決定된 partition이 많이 나올때 有力한 것이다. 가령 $q=16$ 인 table의 경우 이 algorithm을 利用하면 고려대상인 assignment數는 그대로 하는 것보다 決定된 partition이 3개일 때에는 $\frac{1}{10^6}$ 로, 2개일 때에는 $\frac{1}{10^4}$ 로 줄일 수 있음을 計算할 수 있다.

4. 그러나 上述한바와 같이 決定된 partition이 2개일때의 數만 하더라도 널리 使用되어있는 IBM 7090 같은 計算機로서도 취급하지 못함을 생각할때 根本적인 改善이 必要하게 된다. 그러기 위해서도 Davis approach에 對한 보다 깊은 研究가 必要할 것이다. 다시말하면 最小實現數를 얻기위한 nonenumerative method가 먼저 研究되어야하고 그 結果를 土臺로하여 전혀 다른 方法을 모색하여야 할 것이다.

5. 勿論 이 algorithm은 간단히 don't care를 포함한 table에도 適用할 수 있다. 단지 Lemma 4가 반드시 成立하지 않으므로 (13) 各 output function을 實現하는 때는 minterm (AND-OR)型 뿐만 아니라 이때도 maxterm (OR-AND)型도 고려해야한다는 點을 注意하여야 한다.

6. 結 論

本 論文에서는 Davis 方法에 있어서 주어진 assignment scheme의 實際費用을 決定하기위한 algorithm에 對한 試圖를 하였다. 먼저 各 output function을 實現하는 partition 組合을 얻은 다음 이들 中에서 모든 output function을 滿足하는 것들을 表를 利用해서 찾아냈다. 이 algorithm은 아직 研究가 未備하여 直接的이고 힘드

는 것이라고 볼 수 있다. 그러나 보다 便利한 algorithm을 얻기 위해서는 最小實現數를 얻는 nonenumerative method를 解決하여야 될 것 같다. 그리고 앞으로 이 algorithm에 對한 computer program도 研究課題로 남아 있다. (1969년 4월 13일 받음)

參 考 文 獻

- (1) M.P. Marcus, "Switching circuits for engineers," chap. 3, pp. 33, Prentice Hall Inc., N.J., 1962.
- (2) M.C. Torng, "Introduction to the logical design of switching systems," chap. 3, pp. 27, Addison Wesley Publishing Company, Inc., 1964.
- (3) T.A. Dolotta and E.J. McCluskey, Jr., "Encoding of incompletely specified Boolean matrices," Western Joint Computer Conf., AFIPS Proc., vol. 17, pp. 231-238, May 1960.
- (4) T.A. Dolotta, "The coding problem in the design of switching circuits," Ph. D. dissertation, Princeton University, Princeton, N.J., May 1961.
- (5) W.A. Davis, "An approach to the assignment of input codes," IEEE Trans. Electronic Computers, vol. EC-16, No. 4, pp. 435-442, August 1967.
- (6) T.A. Dolotta and E.J. McCluskey, Jr., "The coding of internal states of sequential circuits," IEEE Trans. Electronic Computers, vol. 13, pp. 549-562, October 1964.
- (7) J.A. Brozowski and W.A. Davis, "On the linearity of autonomous sequential machines," IEEE Trans. Electronic Computers, vol. EC-13, pp. 673-699, December 1964.
- (8) T.A. Dolotta, "Minimal p-point functions on n-cubes," Princeton University, Princeton, N.J., Digital Systems Lab., Dept. Elec. Engrg., Tech. Rept. 2, November 1960.
- (9) M.P. Marcus, "Switching circuits for engineers," chap. 8, pp. 101-103, Prentice Hall Inc., N.J., 1962.
- (10) M. Phister, Jr., "Logical design of digital computers," chap. 3, pp. 47-53, John Wiley & Sons, Inc., New York, 1958.
- (11) K. Udagawa, "Logic and digital Circuits," chap. 1, pp. 29-30, Asakura-Shoten Company, Tokyo, 1964.
- (12) E.J. McCluskey, "Introduction to the theory of Switching circuits," chap. 4, pp. 11-179, McGraw-Hill Book Company, New York, 1965.
- (13) M.P. Marcus, "Switching circuits for engineers," chap. 6, pp. 82, Prentice Hall, Inc., N.J., 1962.