

Microprogrammed 디지털 시스템의 制御 記憶 容量의 最小化 (Minimization of the Capacity of Control Memory in Microprogrammed Digital Systems)

趙 榮 一*, 林 寅 七*
(Young Il Cho and In Chil Lim)

要 約

Microprogrammed 디지털 시스템에서 再프로그램을 위한 可變性を 考慮한 制御記憶裝置의 비트 幅을 最小化시키는 새로운 알고리즘을 提示한다. 本 알고리즘은 비트 幅을 最小化시킴과 同時에 weight가 높은 MOP에 高유영역을 부여함으로써 비트 最小化 過程時에 隨伴되는 可變性的 損失을 補償할 수 있다. 또 本 알고리즘을 프로그램하여 從來의 研究結果와 比較 檢討하여 비트 감소와 可變성이 개선된 것을 증명하였다.

Abstract

This paper suggests a new algorithm which minimizes the bit dimension of control memory considering the flexibility for reprogramming in the microprogrammed digital systems.

The algorithm can not only minimize the bit dimension but also compensate the loss of flexibility by giving a highest MOP the unique field in the process of bit minimization. Also, programming the algorithm, that result and previous works are compared and reviewed. Then the bit reduction and the improvement of flexibility are proved.

I. 序 論

現在 마이크로프로그램을 사용한 디지털시스템의 비트 幅 (bit dimension)을 줄이는 問題는 마이크로프로그램 디지털 시스템의 說計者가 直面하는 主要한 問題가 되어왔다. 즉 마이크로프로그램을 사용한 制御部를 說計할 때 만약 電力(power), 溫度條件(thermal requirement), 핀수, 기판크기(board size), 背後面(back plane) 등과 같은 事情으로 크기가 작은 制御記憶裝置을 選擇한다면 制御部の horizontal format에 의한 마이크로프로그램이 選擇한 制御記憶容量을 超過하는 境遇가 發生한다. 이와같은 問題를 解決하기 위

하여 마이크로프로그램의 비트 幅을 減少시키는 方法이 提案되어 왔다.^[1-10] 이에 關한 研究는 Schwartz^[1]가 처음으로 비트 幅을 最小化시키는 方法을 提示하였으며, 이를 바탕으로 Grasselli와 Montanari^[2]가 스위칭 理論의 implicant type covering 問題를 利用하여 Schwartz의 方法을 改善시켰다. 그러나 cover 알고리즘에 의한 最小化는 再프로그램을 위한 可變성이 制限되는 問題는 解結하지 못하였다. 또한 Alice Parker^[3]는 각 micro-operation(MOP)에 weight를 주어 可變성을 考慮한 autonomy 알고리즘을 提示하였으나 Grasselli와 Montanari의 境遇처럼 最小의 비트 減少를 얻지는 못하였다. 또 Ayakannu Mathiala^[4]는 符號化된 format에서 共通 비트를 找함으로써 비트 數를 減少시키는 方法을 提案하였으나 實際 마이크로프로그램을 사용한 디지털 시스템에 適用시킬 수 있는 境遇는 드

*正會員, 漢陽大學校 工科學 電子工學科
(Dept. of Electronics Eng., Hanyang Univ.)
接受日字: 1984年 1月 12日

문 方法이다.

本 論文에서는 最小의 비트 幅을 얻는 同時에 마이크로 워드를 修正하거나 追加시키는 再프로그램을 위한 可變性を 考慮한 새로운 알고리즘을 提案한다. 또 이 알고리즘을 프로그램하여 從來의 研究課果와 比較檢討하여 最小의 비트 幅을 얻는 동시에 可變성이 개선됨을 確認한다.

II. 비트 最小化 問題 및 MOP'S間的 關係

Horizontal microprogrammed 디지털 시스템에서 制御記憶裝置의 각 워드(microinstruction)는 MOP을 制御하기 위해서 N비트와 1개 以上の MOP로 構成된다. B가 MOP를 나타내기 위해 必要한 비트 數를 標示한다면 비트 減少化의 問題는 $B \leq N$ 인 B의 값을 減少시키는 問題가 된다. 그러나 앞서의 研究^{[1][6]}에서는 B를 減少시킴에 따라 마이크로프로그램의 最大 長点이라 할 수 있는 可變性的 損失을 가져왔고, 可變性を 考慮한 Alice Parker의 연구^[7]는 最大의 비트 減少를 얻지 못하였다. 따라서 本 論文에서는 可變性を 考慮해 주면서 最大의 비트 減少를 얻는 알고리즘을 提示한다.

MOP의 集合인 마이크로워드 W_a, W_b, \dots, W_t 를 다음과 같이 定義한다.^[6]

$$\begin{aligned}
 W_a &= \{m_{a_1}, m_{a_2}, \dots, m_{a_n}\}, \\
 W_b &= \{m_{b_1}, m_{b_2}, \dots, m_{b_n}\}, \\
 &\vdots \\
 W_t &= \{m_{t_1}, m_{t_2}, \dots, m_{t_n}\}
 \end{aligned}$$

여기서, m_i 는 마이크로 워드 i에 있는 MOP j를 나타낸다.

(定義 1) 兩立關係(compatibility relation); 任意的 두 MOP m_i, m_j 와 마이크로 워드 W_h 에 대해서 $m_i \in W_h, m_j \in W_h$ 이면 m_i 와 m_j 間에 兩立關係가 成立한다고 定義한다.

(定義 2) 兩立部類(compatibility class:CC); 서로 兩立 關係가 成立하는 MOP'S의 集合을 兩立部類라 한다.

(定義 3) 最大兩立部類(maximal compatibility class; MCC); 兩立部類 C_i 에 어떤 MOP의 追加도 C_i 의 元素 間에 兩立關係를 破壞시킬 때 C_i 를 最大兩立部類라 한다.

(定義 4) 절대최대양립부류(essential MCC) 任意的 MOP가 오직 1個의 MCC에만 속할 때 그러한 MOP을 包含하는 MCC를 절대최대양립부류라 定義한다.

III. 可變性を 考慮한 비트 最小化 알고리즘

本 알고리즘은 兩立部類로 構成된 集合 $\Delta_i = \{C_{i1}, C_{i2}, C_{i3}, \dots, C_{in}\}$ 이 다음 條件을 滿足하는 最小 解를 求하는 알고리즘이다.

(條件 1) 각 MOP은 Δ_i 안에 있는 적어도 하나의 兩立部類안에 속해야 한다.^[6]

(條件 2) $B = B_0 + \sum_{k=1}^n [\log_2 (\#C_{ik} + 1)]$ 의 값이 最小가 되어야 한다.

여기서, B는 가능한 가장 最小의 비트 數를 의미하고 B_0 는 固有領域을 갖는 비트 數, $\#C_{ik}$ 는 C_{ik} 가 cover하는 MOP의 갯수, +1은 No operation을 나타낸다.

可變性を 考慮하여 주기위해 각 MOP에 weight를 다음 定義와 같이 附與한다.

(定義 5) 加重函數(weight function);

$$W_x = N_x \times B_x \text{로 定義한다.}$$

여기서, W_x : MOP x의 加重函數

N_x : MOP x가 속한 각 워드의 數

B_x : MOP x가 속한 word에서 x를 除外한 重複되지 않는 MOP의 數

(定理) 높은 weight의 MOP를 除去하여 固有의 領域을 주면 可變성이 증가한다.

(證明) MOP의 weight가 높다는 것은 그러한 MOP가 많은 워드에 다른 MOP가 함께 存在함을 意味한다(定義 4 參照). 즉 이러한 MOP를 除去하여 固有의 領域을 附與한다면 마이크로프로그램이 變形되거나 또는 새로운 워드가 追加되더라도 각 MOP間에 conflict할 確率이 減少되므로 可變性を 增加시키게 된다.

Q. E. D.

1. 알고리즘 節次

條件 1, 2와 定義와 定理를 바탕으로 비트 幅을 最大限으로 減少시키면서 再프로그램을 위한 可變性を 考慮한 알고리즘 節次는 다음과 같다.

(節次 1) 각 MOP에 대해서 定義 5와 같이 加重函數를 計算한다.

(節次 2) 加重函數의 값이 가장 큰 MOP를 ROM 모델에서 除去하여 固有의 領域을 附與한다.

(節次 3) 나머지 MOP'S에 대하여 最大兩立部類를 求한다.

(節次 4) 節次 3에서 求해진 最大兩立部類에 의해 cover table을 作成한다.

• Cover table을 作成하는 方法

i) 첫번째 列에 MOP m_a, m_b, \dots, m_n 을 나열한다.

ii) $m_j \in C_i, C_i \in MCC$ 인 關係가 成立하면 나열된 m_j 아래 C_i 을 記入한다. 이때 cover table의 行은 그 行의 MOP를 cover할 수 있는 MCC의 集合이 된다.

(節次 4) Cover table에서 essential MCC를 包含하는 모든 行을 除去하여 reduced cover table을 作成한다 (이때 除去되지 않은 行의 MOP'S은 아직 cover되지 않은 狀態이다).

(節次 5) 除去되지 않은 行을 cover할 수 있는 MCC集合을 求한다.

(節次 6) Essential MCC와 節次 5에서 求한 MCC集合에 대해 각 境遇의 solution cover table을 作成한다 (solution cover table을 作成하는 方法은 cover table 作成 方法과 同一).

(節次 7) solution cover table에서 essential MCC를 갖는 行만 除去하여 reduced solution cover table을 作成한다.

(節次 8) 除去되지 않은 行을 cover하는 MCC集合을 求한다.

(節次 9) 節次 7의 essential MCC와 節次 8에서 求한 MCC集合에 의해 가장 적은 비트 幅을 갖는 符號化 비트를 計算한다.

(節次 10) 固有의 領域을 갖는 MOP의 비트 數와 符號化된 비트 數를 合한 總 비트 數가 증가하지 않았다면 節次 1~9를 反復한다.

(節次 11) 總 비트 數가 증가 하였다면 증가하기 바로 前의 固有의 領域과 符號化한 format을 最小의 解로 取한다.

IV. 適用例 및 比較檢討

MOP μ -word	MOP							
	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8
1	1	0	1	0	1	0	0	0
2	0	1	0	0	1	0	0	0
3	0	1	0	1	0	0	0	0
4	1	0	1	0	0	0	0	1
5	1	0	0	0	1	0	1	0
6	1	0	0	1	0	1	0	0
7	0	1	0	0	0	0	0	1
8	0	0	0	0	1	1	0	0
9	0	0	0	0	0	1	0	1

그림 1. Dasgupta의 ROM Model^[10]
Fig. 1. Dasgupta's ROM model^[10]

그림 1에서 1열은 microword을 나타내고 1행은 MOP을 나타낸다. Microword에 해당 MOP이 存在하면 "1"로 表示하고 存在하지 않으면 "0"로 表示한다.

反復回數 加重函數	反復回數			
	1	2	3	4
W_1	24	—	—	—
W_2	9	9	6	3
W_3	6	4	2	2
W_4	6	4	4	4
W_5	20	16	—	—
W_6	12	9	6	3
W_7	2	0	—	0
W_8	12	9	9	—

그림 2. Dasgupta의 ROM model^[10]에 대한 加重函數의 計算結果

Fig. 2. Computing result of weight function for Dasgupta's ROM model^[10]

本 알고리즘을 원래의 크기가 9×8 인 Dasgupta의 例 (그림 1)에 適用시켜 본다.

먼저 각 MOP에 대한 加重函數값을 求한다 (그림 2의 1行). 각 MOP의 加重函數의 計算結果 W_i 의 값이 24로 가장 높음을 알 수 있다. 따라서 MOP m_1 을

COVER TABLE

	1	2	3	4	5	6	7	8	9	10
0	1	1	3	5	2	1	5	0	0	
0	2	3	5	0	4	2	6	0	0	
0	0	4	0	0	0	3	0	0	0	
0	0	0	0	0	0	4	0	0	0	
0	0	0	0	0	0	6	0	0	0	
0	0	0	0	0	0	0	0	0	0	

(a)

SOLUTION COVER TABLE

	1	2	3	4	5	6	7	8	9	10
0	1	1	5	5	4	1	5	0	0	
0	0	4	0	0	0	4	0	0	0	
0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	

(b)

- OP(1, 1) = 2
- OP(1, 2) = 3
- OP(1, 3) = 7
- OP(2, 1) = 4
- OP(2, 2) = 5
- OP(2, 3) = 8
- OP(3, 1) = 0

MB = 5

그림 3. (a) m_1 을 除去한 cover table
(b) (a)에 대한 solution cover table
(c) (b)에 대한 符號化 비트

Fig. 3. (a) The cover table removed m_1 .
(b) The solution cover table for (a).
(c) The encoded bit for (b).

ROM model에서 除去하여 固有의 領域을 附與하고 나머지 MOP에 節次 3~節次 9를 遂行하면 그림 3의 結果를 얻는다.

그러므로 固有領域, 符號化領域, 總 비트 數는 다음과 같다.

$$\{m_1\}, \quad \{m_2, m_3, m_7\}, \quad \{m_4, m_5, m_6\}, \quad \{m_8\}$$

(固有領域) (符號化領域)

總 비트數 = 1 + 2 + 2 + 1 = 6

원래의 비트數 8보다 減少하였으므로 다시 節次 1, 2를 遂行하여 각 MOP'S의 加重函數의 計算結果(그림 2의 2行) W_5 가 가장 높으므로 m_5 을 ROM model에서 除去하여 固有領域을 附與하고 나머지 MOP'S에 대하여 節次 3~節次 9를 遂行하면 그림 4의 結果를 얻는다.

COVER TABLE

1	2	3	4	5	6	7	8	9	10
0	1	1	3	0	2	1	5	0	0
0	2	3	5	0	4	2	0	0	0
0	0	4	0	0	0	3	0	0	0
0	0	0	0	0	0	4	0	0	0
0	0	0	0	0	0	5	0	0	0
0	0	0	0	0	0	0	0	0	0

(a)

SOLUTION COVER TABLE

1	2	3	4	5	6	7	8	9	10
0	1	1	5	0	4	1	5	0	0
0	0	4	0	0	0	4	0	0	0
0	0	0	0	0	0	5	0	0	0
0	0	0	0	0	0	0	0	0	0

(b)

OF(1, 1) = 2
 OF(1, 2) = 3
 OF(1, 3) = 6
 OF(2, 1) = 4
 OF(2, 2) = 7
 OF(2, 3) = 8

(c)

MR = 4

그림 4. (a) m_5 을 除去한 cover table
 (b) (a)에 대한 solution cover table
 (c) (b)에 대한 符號化 비트

Fig. 4. (a) The cover table removed m_5 .
 (b) The solution cover table for (a).
 (c) The encoded bit for (b).

$$\{m_1\}, \quad \{m_3\}, \quad \{m_2, m_3, m_6\}, \quad \{m_4, m_7, m_8\} \quad (1)$$

(固有領域) (符號化 領域)

總 비트數 = 1 + 1 + 2 + 2 = 6

總 비트數가 증가하지 않았으므로 위의 節次를 反復하면 m_5 이 固有領域을 갖게 되고 固有領域, 符號化領域, 總 비트數는 아래와 같다.

$$\{m_1\}, \quad \{m_3\}, \quad \{m_6\}, \quad \{m_2, m_3, m_6\}, \quad \{m_4, m_7\}$$

總 비트數 = 1 + 1 + 1 + 2 + 2 = 7

總 비트數가 증가하였으므로 節次 11에 의해 바로

前 過程의 結果를 最適의 解로 取한다. 그러므로 本 알고리즘에 의한 最適의 解는 (1)과 같다.

Dasguta의 例를 cover 알고리즘과 autonomy 알고리즘에 適用시켜 얻는 結果는 다음과 같다.

Cover 알고리즘 ;

$$\{m_1, m_2\}, \quad \{m_3, m_6, m_7\}, \quad \{m_4, m_5, m_8\} \quad (2)$$

總 비트數 = 2 + 2 + 2 = 6

Autonomy 알고리즘 ;

$$\{m_1\}, \quad \{m_4\}, \quad \{m_5\}, \quad \{m_6\}, \quad \{m_2, m_3/m_6, m_7\}$$

總 비트數 = 4 + 2 + 1

(固有領域) (opset) (비트選擇領域)

本 알고리즘과 cover 알고리즘을 比較하면 cover 알고리즘에서 求한 結果 (2)는 可變性을 전혀 考慮하지 않은 반면 本 알고리즘의 結果 (1)은 總 비트數 6을 얻음과 同時에 높은 weight을 갖는 MOP m_1, m_2 에 固有領域을 附與하여 可變性이 증가함을 알 수 있다. 또

MOP μ -word	MOP										
	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	m_9	m_{10}	m_{11}
1	1	1	1	1	1	1	0	0	0	0	0
2	0	0	1	0	0	0	1	1	1	0	0
3	1	1	0	0	0	0	0	1	1	1	0
4	0	0	0	1	0	0	0	1	0	0	1
5	0	0	0	0	0	1	0	1	0	0	0

그림 5. Schwartz의 ROM model
 Fig. 5. Schwartz's ROM model.

加重函數	反復回數					
	1	2	3	4	5	6
W_1	16	14	-	-	-	-
W_2	16	14	12	-	-	-
W_3	16	14	12	10	-	-
W_4	14	12	10	8	6	-
W_5	5	5	4	3	2	1
W_6	12	10	8	6	4	2
W_7	3	2	2	2	1	1
W_8	36	-	-	-	-	-
W_9	12	10	8	6	4	4
W_{10}	4	3	2	1	1	1
W_{11}	2	1	1	1	1	1

그림 6. Schwartz ROM model 에 대한 加重函數의 計算結果

Fig. 6. Computing result of weight function for Schwartz's ROM model

autonomy 알고리즘 (3)과 比較하면 weight가 높은 M-OP m_1, m_8 에 固有領域을 附與하여 可變性이 높으면서 autonomy 알고리즘보다는 비트數 1개를 더 줄일 수 있음을 알 수 있다.

本 알고리즘을 schwartz 例에 適用시켜 본다.

m_8 을 除去한 境遇 나머지 MOP' S에 대한 符號化 비트는 그림 7과 같다.

1	2	3	4	5	6	7	8	9	10	11
1	2	3	4	6	8	1	0	5	3	1
0	0	0	5	7	9	2	0	7	4	2
0	0	0	0	0	0	4	0	9	6	3
0	0	0	0	0	0	0	0	0	8	6
0	0	0	0	0	0	8	0	0	0	7
0	0	0	0	0	0	0	0	0	0	8
0	0	0	0	0	0	0	0	0	0	9
0	0	0	0	0	0	0	0	0	0	0

1	2	3	4	5	6	7	8	9	10	11
1	2	3	4	6	9	1	0	9	3	1
0	0	0	0	0	0	2	0	0	4	2
0	0	0	0	0	0	4	0	0	6	3
0	0	0	0	0	0	6	0	0	0	6
0	0	0	0	0	0	0	0	0	0	9
0	0	0	0	0	0	0	0	0	0	0

- OF (1, 1) = 1
 - OF (2, 1) = 2
 - OF (3, 1) = 3
 - OF (4, 1) = 4
 - OF (5, 1) = 5
 - OF (5, 2) = 7
 - OF (5, 3) = 10
 - OF (6, 1) = 6
 - OF (6, 2) = 9
 - OF (6, 3) = 11
- MB = 8

그림 7. (a) m_8 을 除去한 cover table
(b) (a)에 대한 solution cover table
(c) (b)에 대한 符號化 비트

Fig. 7. (a) The cover table removed m_8 .
(b) The solution cover table for (a).
(c) The encoded bit for (b).

$\{m_8\}, \{m_1\}, \{m_2\}, \{m_3\}, \{m_4\}, \{m_5, m_7, m_{10}\}, \{m_6, m_9, m_{11}\}$

總 비트數 = 9

m_1 을 除去한 境遇는 그림 8과 같다.

$\{m_8\}, \{m_1\}, \{m_2\}, \{m_3\}, \{m_4, m_7, m_{10}\}, \{m_5, m_9, m_{11}\}, \{m_6\}$

總 비트數 = 9

m_2 을 除去한 境遇는 그림 9와 같다.

$\{m_8\}, \{m_1\}, \{m_2\}, \{m_3\}, \{m_4, m_7, m_{10}\}, \{m_5, m_9, m_{11}\}, \{m_6\}$

總 비트數 = 9

m_3 을 除去한 境遇는 그림 10과 같다.

m_3 을 除去한 境遇 essential MCC가 存在하지 않으므로 cover table과 solution cover table은 同一하다.

1	2	3	4	5	6	7	8	9	10	11
0	1	2	3	5	7	1	0	4	2	1
0	0	0	4	6	8	3	0	6	3	2
0	0	0	0	0	0	5	0	8	5	5
0	0	0	0	0	0	7	0	0	7	6
0	0	0	0	0	0	0	0	0	0	7
0	0	0	0	0	0	0	0	0	0	8
0	0	0	0	0	0	0	0	0	0	0

1	2	3	4	5	6	7	8	9	10	11
0	1	2	3	6	7	1	0	6	2	1
0	0	0	0	0	0	3	0	0	3	2
0	0	0	0	0	0	7	0	0	7	6
0	0	0	0	0	0	0	0	0	0	7
0	0	0	0	0	0	0	0	0	0	0

- OF (1, 1) = 2
 - OF (2, 1) = 3
 - OF (3, 1) = 4
 - OF (3, 2) = 7
 - OF (3, 3) = 10
 - OF (4, 1) = 5
 - OF (4, 2) = 9
 - OF (4, 3) = 11
 - OF (5, 1) = 6
- MB = 7

그림 8. (a) m_1 을 除去한 cover table
(b) (a)에 대한 solution cover table
(c) (b)에 대한 符號化 비트

Fig. 8. (a) The cover table removed m_1 .
(b) The solution cover table for (a).
(c) The encoded bit for (b).

1	2	3	4	5	6	7	8	9	10	11
0	0	1	2	4	6	2	0	3	1	1
0	0	0	3	5	7	4	0	5	2	4
0	0	0	0	0	0	6	0	7	4	5
0	0	0	0	0	0	0	0	0	6	6
0	0	0	0	0	0	0	0	0	0	7
0	0	0	0	0	0	0	0	0	0	0

1	2	3	4	5	6	7	8	9	10	11
0	0	1	2	5	6	2	0	5	1	1
0	0	0	0	0	0	6	0	0	2	5
0	0	0	0	0	0	0	0	0	0	6
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

- OF (1, 1) = 3
 - OF (2, 1) = 4
 - OF (2, 2) = 7
 - OF (2, 3) = 10
 - OF (3, 1) = 5
 - OF (3, 2) = 9
 - OF (3, 3) = 11
 - OF (4, 1) = 6
- MB = 6

그림 9. (a) m_2 를 除去한 cover table
(b) (a)에 대한 solution cover table
(c) (b)에 대한 符號化 비트

Fig. 9. (a) The cover table removed m_2 .
(b) The solution cover table for (a).
(c) The encoded bit for (b).

COVER TABLE

1	2	3	4	5	6	7	8	9	10	11
0	0	0	1	3	5	1	0	2	1	3
0	0	0	2	4	6	3	0	4	3	4
0	0	0	0	0	0	5	0	6	5	5
0	0	0	0	0	0	0	0	0	0	6
0	0	0	0	0	0	0	0	0	0	0

(a)

DP(1, 1)= 4
DP(1, 2)= 7
DP(1, 3)=10
DP(2, 1)= 5
DP(2, 2)= 9
DP(2, 3)=11
DP(3, 1)= 6

(b)

MR= 5

그림10. (a) m_3 을 除去한 cover table
(b) (a)에 대한 符號化 비트

Fig. 10. (a) The cover table removed m_3 .
(b) The encoded bit.

$\{m_6\}, \{m_1\}, \{m_2\}, \{m_3\}, \{m_4, m_7, m_{10}\}, \{m_5, m_8, m_{11}\}, \{m_9\}$

總 비트數 = 1 9

m_4 을 除去한 境遇는 그림 11과 같다.

COVER TABLE

1	2	3	4	5	6	7	8	9	10	11
0	0	0	0	1	3	1	0	2	1	1
0	0	0	0	2	4	3	0	4	3	2
0	0	0	0	0	0	0	0	0	0	3
0	0	0	0	0	0	0	0	0	0	4
0	0	0	0	0	0	0	0	0	0	0

(a)

DP(1, 1)= 5
DP(1, 2)= 9
DP(1, 3)=11
DP(2, 1)= 6
DP(2, 2)= 7
DP(2, 3)=10

(b)

MR= 4

그림11. (a) m_4 을 除去한 cover table
(b) (a)에 대한 符號化 비트

Fig.11. (a) The cover table removed m_4
(b) The encoded bit for (a).

m_4 을 除去한 境遇도 essential MCC가 存在하지 않으므로 cover table과 solution cover table은 同一하다.

$\{m_6\}, \{m_1\}, \{m_2\}, \{m_3\}, \{m_4\}, \{m_5, m_8, m_{11}\}, \{m_9, m_7, m_{10}\}$ (1)

總 비트數 = 9

m_5 을 除去한 경우 總 비트數가 10이므로 바로 前過程의 結果 (1) 를 最適의 解로 取한다.

Cover 알고리즘;

$\{m_1\}, \{m_2\}, \{m_3\}, \{m_4, m_7, m_{10}\}, \{m_5, m_8\}, \{m_6, m_9, m_{11}\}$

總 비트數 = 9

Autonomy 알고리즘;

$\{m_1\}, \{m_2\}, \{m_3\}, \{m_4\}, \{m_5\}, \{m_7, m_8, m_{10}/m_9, m_6, m_{11}\}$
總 비트數 = 9 (비트 選擇領域 = 1)

Schwartz ROM model에 대해서 本 알고리즘의 結果와 cover 알고리즘, autonomy 알고리즘과 比較하면 비트 減少면에는 3개 알고리즘이 同一한 結果를 가져왔으나 cover 알고리즘보다는 可變性이 改善되었음을 알 수 있다.

IV. 結 論

本 論文에서는 비트 幅을 最大限으로 減少시키면서 비트 減少에 依한 可變性의 損失을 最大限으로 防止할 수 있는 새로운 알고리즘을 提示하였다.

앞서의 例에서 나타난 바와 같이 cover 알고리즘의 短點인 可變性의 損失을 補賞하고 또한 autonomy 알고리즘보다는 비트 數가 減少된 結果를 나타냈다.

1) Schwartz의 ROM 모델과

2) Dasguta의 ROM model에 대해 本 論文에서 提案한 알고리즘에 適用시켜 cover 알고리즘 및 autonomy 알고리즘과를 比較한 結果 1)에 대해서는 3개의 알고리즘이 비트 減少 면에서 同一한 結果를 가져 왔으나 cover 알고리즘보다는 可變性이 改善되었다. 2)에 대해서는 autonomy 알고리즘 보다는 비트 幅이 더욱 減少되었으나 cover 알고리즘보다는 可變性이 改善되었고 또한 符號器의 數를 減少시켰다. 따라서 本 알고리즘은 實在 마이크로프로그램 디지털 시스템에 適用시키면 앞서의 cover 알고리즘과 autonomy 알고리즘 보다 비트 數가 더욱 減少 될 것이 豫想되며 또한 可變性의 改善이 確實視된다.

參 考 文 獻

- [1] T. Agerwala, "Microprogram optimization: A survey," *IEEE Trans. Computers*, vol. C-25, pp.962-973, Oct. 1976.
- [2] S.R. Das, D.K. Banerji, and A. Chattopadhyay, "On control memory minimization in microprogrammed digital computers," *IEEE Trans. Computers*, vol. C-22, pp.845-848, Sep. 1973.
- [3] Jean-Loup Baer and Barbara Koyama, "On the minimization of the width of the control memory of microprogrammed processors," *IEEE Trans. Computers*, vol. C-28, pp.310-316, Apr. 1979.
- [4] M. Andrews, *Principles of Firmware Engineering in Microprogram Control*. Ch.4, Computer Science Press, 1980.
- [5] S.J. Schwartz, "An algorithm for minimi-

- zing read-only memories for machine control," *Proc. IEEE 9th Annual Symp. Switching and Automata Theory*, pp. 28-33, 1968.
- [6] A. Grasselli, and U. Montanari, "On the minimization of read-only memories in microprogrammed digital computers," *IEEE Trans. Computers*, pp.1111-1114, Nov. 1970.
- [7] Andrew W. Nagle, Richard Cloutier and Alice C. Parker, "Synthesis of hardware for the control of digital systems," *IEEE Trans. CAD of I.C. and Systems*, vol. CAD-1, pp.201-212, Oct. 1982.
- [8] Ayakannu Mathialagan and Nripendra N. Biswas, "Bit steering in the minimization of control memory in microprogrammed digital computers," *IEEE Trans. Computers*, vol. C-30, pp.144-147, Feb. 1981.
- [9] Totabri Jayasri and Dhruba Basu, "An approach to organizing microinstructions which minimizes the width of control store words," *IEEE Trans. Computers*, pp.514-521, May 1976.
- [10] Jorge Francisco Martinez-Carballido and V. Michael Powers, "General microprogram width reduction using generator sets," *IEEE Microprogramming Conf.*, pp.144-153, 1981.
-