Literal과 Test Set를 최소화하는 RMC Form 결정방법

(RMC Forms Determination with Minimal Literals and Test Sets)

白 哲 和*, 金 宗 相**
(Paik, Chul Hwa and Kim, Chong Sang)

要 約

테스트셋트를 적게 가지면서 literal 수도 적게 포함하는 RMC (Reed-Muller Canonical) form 을 찾는 방법을 고안하여 現存하는 方法보다 간단함을 보였다. 더불어 이 方法은 Don't Care 조건을 갖는 함수에도 적용됨을 보였다.

Abstract

A nonexhaustive procedure for obtaining minimally testable Reed-Muller Canonical (RMC) forms with minimal literals of switching functions is presented.

And, it is shown that this procedure allows nonexhaustive and near-optimal handling of functions with Don't Care conditions.

I. 序 論

임의의 스윗칭함수를 RMC (Reed-Muller Canonical) form으로 표현함으로써 구성되는 회로는 보다 쉽게 결함진단을 할 수 있는 성질을 내포하고 있다. 즉, 진단테스트셋트를 보다 적게 갖게 된다. 지금까지 여러 문헌[1~4]에서 RMC form에 대해 연구되어왔다. 이들 문헌에서 n변수 스윗칭함수의 RMC form은 2^n개 존재한다는 것을 보였다.

Marinkovic and Tosic[1]에서는 이런 2^n개 RMC form 중에서 사용되는 literal 수가 가장 적은 RMC form을 찾아내는 알고리즘에 대해 연구하였으며, Kodandapani and Setlur[2]는 문헌[1]의 연구 내용을 변형시켜 2^n개의 RMC form 중에서, 사용되는 項의 수가 가장 적은 RMC form을 찾아내는 알고리즘을 고안해내서 회로를 간단하게 설계하는 방법을 연구했다. 또, Reddy[3]는 보다 적은 적은 진단 테스트셋트를 갖는 회로의 RMC form 은 form 안에 짝수씩 나타나는 변수(n_e)의 갯수를 적게 갖는 것을 뜻한다고 밝혔다.

이 성질을 이용하여 E. W. Page[4]는 2^n RMC form 중에서 n_e가 가장 적은 RMC form을 찾는 알고리즘을 보였다.

그러므로 본 논문에서는 가장 적은 결함진단 테스트셋트를 가질 뿐만 아니라, 회로에 사용되는 literal 수도 가장 적은 RMC form을 2^n RMC form 들을 일일이 유도하지 않고 찾을 수 있는 알고리즘을 고안 했으며 또한, 이 알고리즘으로 임의의 함수가 Don't Care 조건을 지녔을 때도 쉽게 조건을 결정하여 원하는 RMC form을 얻을 수 있음을 보였다.

* 準會員, 서울 大學校 工科大學 電子科
(Dept. of Electronic Engineering, Seoul National University)

** 正會員, 서울 大學校 工科大學 電子計算機工學科
(Dept. of Computer Engineering, Seoul National University)

接受日字：1981年 1月 8日

II. RMC Form을 결정하는 알고리즘

스윗칭함수의 일반 RMC form은 式(1)과 같다.

$$f(x_n, \cdots\cdots, x_2, x_1) = a_0 \oplus a_1 \dot{x}_1 \oplus a_2 \dot{x}_2 \oplus, \cdots\cdots, \oplus a_{2^n-1} \dot{x}_n \cdots\cdots \dot{x}_2 \dot{x}_1 \qquad \cdots\cdots\cdots\cdots (1)$$

이때 a_i는 2進 상수이며 $\dot{x}_i$는 RMC form 안에서 x_i나 $\overline{x}_i$ 둘 중에 하나의 형태를 나타낸다. 그러므로 $\dot{x}_i$에 따라 RMC form은 2^n개 존재하게 된다.

이런 2^n개의 RMC form의 각각을 polarity에 따라 확장벡터(expansion vector) $Q=(q_1, q_2, \cdots q_n)$로 표시하며 이때 q_1은 LSB(least significant bit)이며 q_i는 아래와 같이 表示된다.

$$q_i \quad \begin{cases} 0 : \dot{x}_i = x_i \\ 1 : \dot{x}_i = \overline{x}_i \end{cases} \qquad \cdots\cdots (2)$$

이러한 q_i로 표현한 2^n RMC form을 $p(Q)$로 표시한다. 즉, $p(111)$이면 $p(\overline{x}_1, \overline{x}_2, \overline{x}_3)$로 표시된 RMC form을 뜻한다. 그리고 式(3)과 같이 될 때 이 Q와 Q^i는 i에 관해서 인접된 확장 백터라 한다.

$$Q = (q_1, \cdots\cdots, q_i, \cdots\cdots, q_n)$$
$$Q^i = (q_1, \cdots\cdots, \overline{q}_i, \cdots\cdots, q_n) \qquad \cdots\cdots (3)$$

$p(Q)$에서 $p(Q^i)$로 변할 때 이 $p(Q^i)$ 안에서 같은 AND項으로 인해 삭제되는 項을 삭제하지 않을 때 각 변수가 RMC form 안에 나타나는 횟수를 표시하기 위해 패리티 벡터 $V(Q)=(V_1, V_2, \cdots, V_n)$을 도입한다. 이때 V_i는

$$V_i = \begin{cases} \text{“}\odot\text{”} : p(Q)\text{ 안의 AND 項에 } \dot{x}_i\text{가 홀수 개로 나타날때} \\ \text{“E”} : \text{짝수 개로 나타날때} \end{cases} \qquad \cdots\cdots (4)$$

본 논문에서의 “+” 기호는 산술 合을 뜻한다.

(정리) $p(Q)$안에 변수 $\dot{x}_j$가 나타나는 항의 數가 g이고 변수 $\dot{x}_i$와 $\dot{x}_j$가 동시에 나타나는 AND 項의 數가 m이고 $p(Q)$에서 $p(Q^i)$로 변한 후 같은 AND 項이 $p(Q^i)$안에 있을 경우 이 AND 項을 삭제하지 않는다면, 이때 $P(Q^i)$안에 변수 $\dot{x}_i$가 나타나는 총갯수는 $P(Q)$의 $\dot{x}_i$총갯수와 같으며 변수 $\dot{x}_i$가 나타나는 총갯수는 $g+m$ 이다.

(증명) $p(Q)$안의 임의의 AND 項 $p\dot{x}_i$(p는 $\dot{x}_i$를 포함하지 않는 literal 들의 AND이다)을 생각하자. $p(Q)$에서 $p(Q^i)$로 변화할 때 $p\dot{x}_i = p \oplus p\dot{x}_i'$로 바꾸어지기 때문에 이런치환이 $\dot{x}_i$를 포함하는 부수적인 AND 項을 발생시키지 않는다. 더구나 $p(Q)$는 redundant 하지 않기 때문에 $p(Q)$안에 또 다른 $p\dot{x}_i$항이 없으므로 $p(Q^i)$의 $p\dot{x}_i'$는 상쇄되어 없어지지 않는다.

또한 변수 $\dot{x}_i$와 $\dot{x}_j$가 동시에 나타나는 AND項의 총 갯수가 m개 있다는 것은, $p_1\dot{x}_i\dot{x}_j \oplus p_2\dot{x}_i\dot{x}_j \oplus, \cdots\cdots, \oplus p_m\dot{x}_i\dot{x}_j$ 올 뜻하는 데 이것을 $p(Q^i)$로 바꿀 때 $p_1\dot{x}_j \oplus p_1\dot{x}_i'\dot{x}_j \oplus p_2\dot{x}_j \oplus p_2\dot{x}_i'\dot{x}_j \oplus, \cdots\cdots, \oplus p_m\dot{x}_j \oplus p_m\dot{x}_i'\dot{x}_j$로 바뀐다.

같은 AND項이 있는데도 서로를 삭제하지 않을 경우 $p(Q^i)$안에 $\dot{x}_j$가 나타나는 총 갯수는 $k+2m$이다. (이때 k는 $p(Q)$안에서 $\dot{x}_i$를 포함하지 않으면서 $\dot{x}_j$를 가지고 있는 AND項의 갯수라 한다.)

이때 $k+m=g$ 임을 쉽게 알 수 있으므로 $k+2m = k+m+m = g+m$ 이므로 $p(Q^i)$안에 $\dot{x}_j$가 나타나는 AND 項의 총 갯수는 $g+m$이 된다.

정리에 의해 $p(Q^i)$에 나타나는 각 변수들의 총갯수를 구할 수 있으므로 이것으로 패리티 벡터 $V(Q)$를 결정하여 EO(짝수, 홀수) 판정을 할 수 있다. 이때 앞에서 고려 않했던 같은 항으로 인해 서로 삭제되는 $p(Q^i)$안의 AND 項은 짝수개씩 삭제되기 때문에 $p(Q^i)$의 각 변수들의 EO 판정을 하는데 어떠한 영향도 미치지 않는다.

지금까지의 방법으로 정해진 어떤 $p(Q)$와 인접한 $p(Q^i)$들을 일일이 RMC form을 결정하지 않고도 $V(Q)$를 결정할 수 있으므로, n_e가 가장 적은 RMC form을 결정할 수 있다.

2^n RMC form을 일일이 유도하여 패리티 벡터 $V(Q)$를 구하지 않고 2^n 확장 벡터들 중에서 임의의 패리티 Q를 정하여 그것과 거리가 1인 벡터들로 서브셋트 $I_1, I_2, \cdots, I_k, \cdots, I_m$을 구성한다. I_k에서 k는 $1 \le k \le m$이다. 이때 Q는 이런 서브셋트 I_k를 최소로 갖도록 정해야 한다. 이런 각 서브셋트의 첫 번째 확장 벡터를 c_k로 표시한다. 예로써 $n=3$이면 두개의 서브셋트 I_1, I_2가 존재하며 아래 式(5)와 같다.

$$I_1 = (000, 001, 010, 100)$$
$$I_2 = (111, 110, 101, 011) \qquad \cdots\cdots (5)$$

이처럼 RMC form으로 유도해야 할 polarity는 $c_1=(000)$, $c_2=(111)$ 뿐으로 이것들의 RMC form $p(000)$, $p(111)$로써 2^3개 가능한 모든 RMC form의 패리티 벡터 $V(Q)$를 정리의 방법으로 결정할 수 있다.

예로써 $n=5$일때의 c_k는 式(6)과 같다.

$$\begin{array}{ll} c_1 = (00000) & c_5 = (00111) \\ c_2 = (01000) & c_6 = (01111) \\ c_3 = (10000) & c_7 = (10111) \\ c_4 = (11000) & c_8 = (11111) \end{array} \qquad \cdots\cdots (6)$$

(1) 패리티 벡터 $V(Q)$를 결정하는 알고리즘

$p(Q)$안에 변수 $\dot{x}_i$와 $\dot{x}_j$가 동시에 나타나는 AND 項의 數가 m이므로 이때 이 m들을 표시하는데 다음과 같은 行列, M_k를 도입한다. 이 M_k 行列은

行列 바깥편에 行列을 따라 literal $\dot{x}_1\dot{x}_2\cdots\dot{x}_n$ 을 표시하고 요소 m_{ij}는 $\dot{x}_i\dot{x}_j$ 雙을 포함하는 AND 項의 合을 의미한다.

M_k를 표현하기 위한 예로써,

Example 1 : $f(x_5,x_4,x_3,x_2,x_1)$의 RMC form은

$$P(c_k)=1\oplus\bar{x}_1\oplus\bar{x}_1x_5\oplus x_2x_3\oplus x_3x_4\oplus x_3x_5\oplus\bar{x}_1x_3x_5 \quad \cdots\cdots (7)$$

$$M_k=\begin{array}{c|ccccc|} & \dot{x}_1 & \dot{x}_2 & \dot{x}_3 & \dot{x}_4 & \dot{x}_5 \\ \dot{x}_1 & \phi & 0 & 1 & 0 & 2 \\ \dot{x}_2 & 0 & 0 & 1 & 0 & 0 \\ \dot{x}_3 & 1 & 1 & 0 & 1 & 2 \\ \dot{x}_4 & 0 & 0 & 1 & 0 & 0 \\ \dot{x}_5 & 2 & 0 & 2 & 0 & 0 \end{array} \quad \cdots\cdots (8)$$

이때 $c_k=(10000)$ 이다.

또, $i=j$ 일때 m_{ij}는 0으로 규정하는데 나중에 literal의 총수를 계산하는 데 사용하기 위해 $i=j$ 인 항이 있을 때는 ϕ로 표시한다. 그리고 $p(c_k)$안에 $\dot{x}_j$가 나타나는 총수, g는 각 변수가 나타나는 횟수를 표시하는 벡터 G_k로 표시한다.

이때 위의 예의 G_k는,

$$G_k=(\overset{x_1}{3},\ 1,\ 4,\ 1,\ 3) \quad \cdots\cdots (9)$$

이렇게 구한 벡터 M_k, G_k로서 $p(c_k)$에 인접한 $p(c_k^i)$ ($i=1,2,\cdots,n$)를 다음과 같이 구한다.

$$\begin{bmatrix}\dot{V}(c_k^1)\\ \dot{V}(c_k^2)\\ \vdots \\ \dot{V}(c_k^n)\end{bmatrix}=M_k+\dot{G}_k \quad \cdots\cdots (10)$$

이때 I_k는 式(11)과 같이 된다.

$$I_k=(10000,00000,11000,10100,10010,10001) \quad \cdots\cdots (11)$$

그리고 $\dot{G}_k$는 벡터 G_k를 M_k와 行의 次元을 같게 한 벡터이다. 또 $\dot{V}(c_k^i)$는 $P(c_k^i)$ 안에 변수 $\dot{x}_i$와 $\dot{x}_j$의 총수를 나타내는 벡터이다.

$$\begin{bmatrix}\dot{V}(c_k^1)\\ \dot{V}(c_k^2)\\ \dot{V}(c_k^3)\\ \dot{V}(c_k^4)\\ \dot{V}(c_k^5)\end{bmatrix}=\begin{bmatrix}0&0&1&0&2\\0&0&1&0&0\\1&1&0&1&2\\0&0&1&0&0\\2&0&2&0&0\end{bmatrix}+\begin{bmatrix}3&1&4&1&3\\3&1&4&1&3\\3&1&4&1&3\\3&1&4&1&3\\3&1&4&1&3\end{bmatrix}$$

$$=\begin{bmatrix}3&1&5&1&5\\3&1&5&1&3\\4&2&4&2&5\\3&1&5&1&3\\5&1&6&1&3\end{bmatrix} \quad \cdots\cdots (12)$$

이것으로 $V(c_k^i)$를 결정한다.

$$\begin{bmatrix}V(C_k^1)\\ V(C_k^2)\\ V(C_k^3)\\ V(C_k^4)\\ V(C_k^5)\end{bmatrix}=\begin{bmatrix}\mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}\\ \mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}\\ \mathrm{E}&\mathrm{E}&\mathrm{E}&\mathrm{E}&\mathrm{O}\\ \mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}\\ \mathrm{O}&\mathrm{O}&\mathrm{E}&\mathrm{O}&\mathrm{O}\end{bmatrix} \quad \cdots\cdots (13)$$

$V(c_k^i)$ ($i=1,2,\cdots,n$)이 결정됨으로써 각각의 C_k에 대해 반복하여서 EO 판정을 할 수 있다. 위의 경우에 $C_k=(10000)$이므로 $C_k^1=(00000)$, $C_k^2=(11000)$, $C_k^4=(10010)$인 RMC form의 $V(00000)$, $V(11000)$, $V(10010)$가 전부 $n_e=0$를 내포함으로써 예측한대로 테스트셋트를 적게 갖게 된다. 즉, $V(c_k^3)=V(10100)$의 전부 n_e는 4이므로 $P(c_k^3)$의 테스트셋트는 $n+4+2n_e=17$인데 비해 $P(c_k^1)$, $P(c_k^2)$, $P(c_k^4)$의 테스트셋트는 $n+4=9$이다. (이때 Reddy[3]에 의해 RMC form으로 구성된 회로의 테스트셋트는 $n+4+2n_e$로 알려져 있다.)

이때 $P(C_k^1)=P(00000)$을 구해보면, RMC form 안의 $\bar{x}_1$ 대신 $1\oplus x_1$을 대입하면 얻을 수 있다.

즉, $P(C_k^1)=x_1\oplus x_5\oplus x_1x_5\oplus x_2x_3\oplus x_3x_4\oplus x_1x_3x_5 \quad \cdots\cdots (14)$

여기서 $V(C_k^1)=(\mathrm{O},\mathrm{O},\mathrm{O},\mathrm{O},\mathrm{O})$이다. 결국 같은 결과를 얻는다. 다음은 이 알고리즘과 병행하여 RMC form의 전체 literal 數를 구하는 알고리즘을 설명한다.

(2) Literal 個數 $T(Q)$를 결정하는 알고리즘. 지금까지 구한 $V(C_k^i)$로 n_e의 數가 가장적은 RMC form을 찾아냈다. 단지 이 $V(C_k^i)$로는 EO판정을 할 수 있었다. 그러나 $\dot{V}(C_k^i)$로 $T(Q)=T(C_k^i)$을 결정할 수는 없다. 왜냐하면 식(10)에 사용된 G_k의 도입은 정리에서 알 수 있듯이 $P(C_k)$에서 $P(C_k^i)$로 변할때 같은 AND項을 삭제하지 않았기 때문이다. 그러므로 여기서 임의의 $T(C_k^i)$를 얻는 방법을 설명한다. 삭제되거나 늘어나는 literal들을 각 변수들의 weight, w_i라하며 이것을 weight 벡터, $W(C_k^i)$로 표시한다. 우선 이 $W(C_k^i)$를 구하기 위해, 고려하는 RMC form $P(C_k)$안에 상수항 a_0가 없을 때와 있을 때를 나누어 생각한다.

(2−a) 상수항 a_0가 없을 때의 $W(C_k^i)$는

$$W(C_k^i) = W_a - 2W_b \quad \cdots\cdots (15)$$

이 식(15) 처럼 두개의 벡터로 구성된다.

첫째 : W_a는, 이미 M_k 벡터에서 $i=j$일때 m_{ii}를 ϕ로 표시했는데 이 ϕ가 있는 변수 $\dot{x}_i$의 위치에 상수1을 갖는 行 벡터로 구성된다. 이 W_a가 존재하는 이유는 m_{ii} 요소가 $\dot{P}(C_k^i)$로 변화할 때 이 RMC form 안의 $P\dot{x}_i$ (P=1이므로)는, $\dot{x}_i = 1 \oplus \dot{x}_i'$ 로 치환되기 때문에 이 상수항 1이란 literal이 하나씩 늘어나기 때문에 W_a 벡터를 도입함으로써 늘어나는 literal 수를 계산할 수 있게 된다.

$$M_k = \begin{bmatrix} \phi & & & & \\ & 0 & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \cdot \\ & & & & & \phi \end{bmatrix} \text{이면, } W_a = \begin{bmatrix} 1 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{bmatrix} \text{이다.}$$

둘째 : W_b는 삭제되는 literal 수를 표시하는 벡터이다. 이때 식(10)에서 구한 $\dot{V}(C_k^i)$들의 변수갯수들을 전부 더한 것을 $S(C_k^i)$로 표시한다. 즉,

$$S(C_k^i) = \Sigma\ \dot{V}(C_k^i) \quad \cdots\cdots (16)$$

이다. 삭제되는 literal 들은 짝수개씩 삭제되므로 삭제되는 변수를 찾아서 2배하면 삭제되는 총 literal 수가 된다.

이 때 W_b의 요소는 $P(C_k)$안의 $P\dot{x}_i$ 항(이때 P는 x_i 와 무관한 AND 項이며 이경우 $P \neq 1$ 이다.)에 대해 이 form 안에 P 항이 존재하는가 살펴보아 그 P가 구성하는 변수갯수들의 총합이 W_b를 형성한다. 다시 말하면 이것은 $P\dot{x}_i = P \oplus P\dot{x}_i'$ 로 치환할때 $P(C_k)$에 P 항이 있으면 P 항이 구성하는 변수갯수의 2배의 literal 수가 없어지는 것을 뜻한다. 이때 W_b 벡터도 역시 行 벡터이다.

前例에서 $\bar{x}_1$ 의 경우 $x_3 x_5$가 삭제되고
x_3 의 경우 $\bar{x}_1 x_5$가 삭제되고
x_5 의 경우 $\bar{x}_1$가 삭제된다.

그러므로 W_b는 式(17)과 같이 된다.

$$W_b = \begin{bmatrix} 2 \\ 0 \\ 2 \\ 0 \\ 1 \end{bmatrix} \quad \cdots\cdots (17)$$

(2−b) 상수항 a_0가 있을 때의 $W(C_k^i)$는

$$W(C_k^i) = W_a - 2W_b \quad \cdots\cdots (18)$$

이식(18)의 W_a는, M_k 벡터안에 m_{ii}가 있고, 이것이 $P(C_k)$에서 $P(C_k^i)$로 변화할 때 $1 \oplus \dot{x}_i = \dot{x}_i'$로 치환되기 때문에 literal 수의 계산에 새로 추가될 literal 수가 없으며 이것은 m_{ii}가 ϕ인 위치에 0이 들어가는 것을 의미한다. 또 m_{ii}가 0인 위치에는 1이 들어간다. 이것은 원래의 RMC form의 a_0상수항을 literal 수의 계산에 첨가해야 하기 때문이다.

그러므로, 이때 $\dot{W}_a$는

$$[W_a]^T \times [\dot{W}_a] = [0] \quad \cdots\cdots (19)$$

이때 W_a는

$$M_k = \begin{bmatrix} \phi & & & & \\ & 0 & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \cdot \\ & & & & & \phi \end{bmatrix} \text{이면, } W_a = \begin{bmatrix} 0 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix}$$

임을 알 수 있다.

이렇게 $W(C_k^i)$를 구한 후, 전체 literal 수, $T(C_k^i)$는

$$T(C_k^i) = S(C_k^i) + W(C_k^i) \quad \cdots\cdots (20)$$

앞의 예의 $T(C_k^i)$는,

$$S(C_k^i) = \begin{bmatrix} S(C_k^1) \\ S(C_k^2) \\ S(C_k^3) \\ S(C_k^4) \\ S(C_k^5) \end{bmatrix} = \begin{bmatrix} 15 \\ 13 \\ 17 \\ 13 \\ 16 \end{bmatrix} \quad \cdots\cdots (21)$$

$$W(C_k^i) = \dot{W}_a - 2W_b = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} - 2 \cdot \begin{bmatrix} 2 \\ 0 \\ 2 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -4 \\ 1 \\ -3 \\ 1 \\ -1 \end{bmatrix} \quad (22)$$

이 $S(C_k^i)$와 $W(C_k^i)$를 식(20)에 대입하여 $T(C_k^i)$를 구한 결과가 式(23)이다.

$$T(C_k^i) = S(C_k^i) + W(C_k^i) = \begin{bmatrix} 15 \\ 13 \\ 17 \\ 13 \\ 16 \end{bmatrix} + \begin{bmatrix} -4 \\ 1 \\ -3 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 11 \\ 14 \\ 14 \\ 14 \\ 15 \end{bmatrix} \quad \cdots\cdots (23)$$

이때 위의 $T(C_k^i)$에서 literal 수를 가장 적게 포함하는 RMC form은 $T(C_k^i)=11$로써, 앞에서 구한 변수가 홀수 패리티인 $V(C_k^1)$, $V(C_k^2)$, $V(C_k^4)$ 중에서 두조건, 즉, $n_e=0$과 가장 적은 literal 수를, 만족하

는 polarity는 $Q=(00000)$임을 알 수 있다.

이때의 RMC form $P(00000)$는,

$$P(00000)=x_1\oplus x_5\oplus x_1x_5\oplus x_2x_3\oplus x_3x_4\oplus x_1x_3x_5$$

로써 $T(00000)=11$, 테스트 셋트$=9$

$$V(00000)=(\odot,\odot,\odot,\odot,\odot)$$

이고, 이렇게 식(10)과 (20)을 각 C_k에 따라 반복하여 만족하는 RMC form을 찾을 수 있다. 앞에서 $n=5$일때의 서브셋트에 의해 $P(10000)=P(C_3)$임을 알 수 있다. 그러므로 $P(C_1)=P(C_3^1)=P(00000)$이고, 다른 RMC form중에서 $n_e=0$를 만족하는 form은,

$$\left.\begin{array}{ll} P(01010)=P(C_2^4) \text{ 는 } & T(C_2^4)=11 \\ P(11001)=P(C_4^5) \text{ 는 } & T(C_4^5)=14 \\ P(00011)=P(C_5^3) \text{ 는 } & T(C_5^3)=14 \\ P(10011)=P(C_7^3) \text{ 는 } & T(C_7^3)=14 \end{array}\right\} \cdots\cdots(24)$$

이때 literal 수까지 고려하면,

$$P(01010)=x_1\oplus x_5\oplus x_1x_5\oplus x_2x_3\oplus x_3x_4\oplus x_1x_3x_5$$

가 만족하는 또 다른 RMC form이 된다.

결국, 원하는 RMC form으로 $P(00000)$와 $P(01010)$을 채택하면 된다.

지금까지의 (1)(2)알고리즘을 종합한

(알고리즘)

Step 1 : 모든 패리티 Q를 서브셋트 벡터 I_k로 나누고 이것들의 C_k를 정한다.

Step 2 : 각 C_k에 대해 RMC form $P(C_k)$를 결정하고, 각 C_k에 대해 G_k를 결정한다.

Step 3 : 각 C_k에 대해 M_k를 결정한다.

Step 4 : 식(10)를 사용하여 각 C_k에 대해 $V(C_k^i)$를 구하여 여기서 $V(C_k^i)$를 정한다.

Step 5 : 각 C_k의 RMC form에 상수항 a_0가 있는지의 여부에 따라 식(15)나 (18)를 적용하여 각 C_k의 $W(C_k^i)$를 구한다.

Step 6 : 식(20)에 의해 각 C_k에 따른 $T(C_k^i)$를 구한다.

Step 7 : $V(C_k^i)$, $T(C_k^i)$에서 짝수 패리티가 적으면서 literal 수를 적게 갖는 RMC form을 찾는다.

이 알고리즘은 특히 Don't Care 조건을 가진 함수에도 적용되는 것을 다음 절에서 설명한다.

Ⅲ. Don't Care 조건 함수인 例

Don't Care 조건을 가진 함수인 경우 앞에서 제시한 알고리즘으로 이 Don't Care 조건을 결정할 수 있으며 원하는 RMC form을 구할 수 있음을 보인다.

Example 2 :

$$f(x_3,x_2,x_1)=\Sigma m(1,4,5)+\Sigma d(3,7) \quad\cdots\cdots(25)$$

Step 1 : 2^3 확장 벡터를 서브셋트 I_1, I_2로 나눈다. 즉,

$$C_1=(000),\ I_1=(000,001,010,100)$$
$$C_2=(111),\ I_2=(111,110,101,011)\cdots(26)$$

Step 2 : 우리는 RMC form과 G_k를 결정한다.

$$\begin{bmatrix} a_0\\ a_1\\ a_2\\ a_3\\ a_4\\ a_5\\ a_6\\ a_7 \end{bmatrix}=\begin{bmatrix} 10000000\\ 11000000\\ 10100000\\ 11110000\\ 10001000\\ 11001100\\ 10101010\\ 11111111 \end{bmatrix}\cdot\begin{bmatrix} 0\\ 1\\ 0\\ d_1\\ 1\\ 1\\ 0\\ d_2 \end{bmatrix}=\begin{bmatrix} 0\\ 1\\ 0\\ \overline{d}_1\\ 1\\ 1\\ 1\\ \overline{d}_1\oplus d_2 \end{bmatrix} \quad (27)$$

이때 $d(3)=d_1$, $d(7)=d_2$, $d_1\oplus d_2=b$ 이다.

$$\begin{aligned} P(C_1)&=x_1\oplus\overline{d}_1x_2x_1\oplus x_3\oplus x_3x_1\oplus x_3x_2\oplus(\overline{d}_1\oplus d_2)x_3x_2x_1\\ &=x_1\oplus\overline{d}_1x_2x_1\oplus x_3\oplus x_3x_1\oplus x_3x_2\oplus\overline{b}x_3x_2x_1 \end{aligned}$$

$$P(C_2)=d_2\oplus d_2\overline{x}_1\oplus\overline{d}_2\overline{x}_2\oplus d_2\overline{x}_2\overline{x}_1\oplus b\overline{x}_3\oplus b\overline{x}_3\overline{x}_1\oplus b\overline{x}_3\overline{x}_2\oplus\overline{b}\,\overline{x}_3\overline{x}_2\overline{x}_1 \quad\cdots\cdots(28)$$

$$G_1=(2+\overline{b}+\overline{d}_1,\ 1+\overline{b}+\overline{d}_1,\ 3+\overline{b})$$
$$G_2=(1+2d_2,\ 2,\ 1+2b)$$

Step 3 : 여기의 M_1, M_2는

$$M_1=\begin{bmatrix} \phi & \overline{b}+\overline{d}_1 & 1+\overline{b}\\ \overline{b}+\overline{d}_1 & 0 & 1+\overline{b}\\ 1+\overline{b} & 1+\overline{b} & \phi \end{bmatrix};\ M_2=\begin{bmatrix} \cancel{d}_2 & \overline{b}+d_2 & 1\\ \overline{b}+d_2 & \cancel{d}_2 & 1\\ 1 & 1 & \cancel{b} \end{bmatrix} \quad\cdots\cdots(29)$$

Step 4 : 구한 M_1, M_2, G_1, G_2를 식(10)에 대입

$$\dot{V}(C_1^i)=\begin{bmatrix} 2+\overline{b}+\overline{d}_1 & 1+2\overline{b}+2\overline{d}_1 & 4+2\overline{b}\\ 2+2\overline{b}+2\overline{d}_1 & 1+\overline{b}+\overline{d}_1 & 4+2\overline{b}\\ 3+2b+d_1 & 2+2b+d_1 & 3+b \end{bmatrix};$$

$$V(C_1^i)=\begin{bmatrix} ? & \odot & E\\ E & ? & E\\ \odot+d_1 & E+d_1 & 3+b \end{bmatrix}$$

$$\dot{V}(C_2^i)=\begin{bmatrix} 1+2d_2 & 2+\overline{b}+d_2 & 2+2b\\ 1+\overline{b}+3d_2 & 2 & 2+2b\\ 2+2d_2 & 3 & 1+2b \end{bmatrix};$$

$$V(C_2^i)=\begin{bmatrix} \odot & ? & E\\ ? & E & E\\ E & \odot & \odot \end{bmatrix} \quad\cdots\cdots(30)$$

이때 패리티 벡터를 살펴보면, 전변수가 홀수 패리

티인 polarity가 없으므로 $n_e=1$인 polarity를 구한다.

이경우,

$d_2=1$ 일때 $V(C_1^1)$

$b=1$ 일때 $V(C_1^3)$

$d_1=0$ 일때 $V(C_2^1)$와 $V(C_2^3)$이다.

Step 5 : $W(C_k^i)$를 구한다.

$W(C_1^i)$일때 : $W(C_2^i)$ 이고

$d_2=1$ 일때 ;

$$W_a=\begin{bmatrix}1\\0\\1\end{bmatrix}\ W_b=\begin{bmatrix}1+2\bar{b}\\1+2\bar{b}+\bar{d}_1\\1+2\bar{b}\cdot\bar{d}_1\end{bmatrix}\ ;\ \dot{W}_a=\begin{bmatrix}0\\1\\d_1\end{bmatrix}\ ;$$

$d_2=0$ 일때

$$W_b=\begin{bmatrix}\bar{d}_1\\1+\bar{d}_1\\1\end{bmatrix}\ ;\ \dot{W}_a=\begin{bmatrix}0\\1\\d_1\end{bmatrix}\ W_b=\begin{bmatrix}d_1\\d_1\\d_1\end{bmatrix}\ \cdots(31)$$

Step 6 ; 식(20)에 의해 $T(C_k^i)$를 구한다.

$$T(C_1^i)=\begin{bmatrix}6+\bar{b}+3\bar{d}_1\\5+\bar{b}+\bar{d}_1\\7+5\bar{b}+3\bar{d}_1-4\bar{b}\cdot\bar{d}_1\end{bmatrix}\ ;$$

$$\overset{d_2=1}{T(C_2^i)}=\begin{bmatrix}9-\bar{d}_1\\8-\bar{d}_1\\7+\bar{d}_1\end{bmatrix}\ ;\ \overset{d_2=0}{T(C_2^i)}=\begin{bmatrix}6-d_1\\7-d_1\\6+d_1\end{bmatrix}\cdots\cdots(32)$$

특히 $W(C_2^i)$의 $\dot{W}_a$는 M_2의 m_{ii} 요소들이 $\dot{W}_a$ 안에서 반대로 나타나므로 $\dot{W}_a=(\bar{d}_2, d_2, \bar{b})$가 되며 $\dot{V}(C_2^i)$를 만들때 계산에 넣지 않는다. 지금까지 step 4에서 정해진 Don't Care 조건을 step6의 벡터 $T(C_k^i)$에 대입하여 literal 수를 적게 갖는 RMC form을 찾는다.

즉, $V(C_1^1)$: step 4에서 $d_2=1$이고 step 6에서 $d_1=1$인 조건을 구할 수 있다. 또 $T(C_1^1)=6$이고 $V(C_1^1)=(\odot,\odot,E)$ 이다.

$V(C_1^3)$; Step 4에서 $b=1$이고 step 6에서 $d_1=1$이므로 $d_2=0$이 된다. 이때 $T(C_1^3)=6$이고 $V(C_1^3)=(\odot,E,\odot)$ 이다.

$V(C_2^1)$; Step 4에서 $d_1=0$이고 step 6에서 $d_2=0$을 얻는다. 이때 $T(C_2^1)=6$이고 $V(C_2^1)=(\odot,\odot,E)$ 이다.

$V(C_2^3)$; Step 7에서 $d_1=0$, $d_2=0$를 얻고 이때 $T(C_2^3)=6$, $V(C_2^3)=(E,\odot,\odot)$이다.

그러나 처음에 시작한 RMC form P(111)의 경우 역시 $d_1=0$, $d_2=0$을 갖게 되면 $G_2=(\odot,E,\odot)$가 되며 $T(111)=4$뿐으로 가장 바람직한 RMC form이 되지만 여기서 例를 위하여 다른 $n_e=1$이 되는 polarity를 구해봤다. 이때 $n_e=1$이 되는 RMC form은,

$$\begin{aligned}
&P(111)=\bar{x}_2\oplus\bar{x}_3\bar{x}_2\bar{x}_1 && (d_1=0,\ d_2=0)\\
&P(011)=\bar{x}_2\oplus\bar{x}_3\bar{x}_2\oplus\bar{x}_3\bar{x}_2x_1 && (d_1=0,\ d_2=0)\\
&P(110)=\bar{x}_2\oplus\bar{x}_2\bar{x}_1\oplus x_3\bar{x}_2\bar{x}_1 && (d_1=0,\ d_2=0)\\
&P(100)=1\oplus\bar{x}_1\oplus x_3\bar{x}_1\oplus x_3x_2\bar{x}_1 && (d_1=1,\ d_2=1)\\
&P(001)=1\oplus x_2\oplus\bar{x}_3\oplus\bar{x}_3x_1\oplus\bar{x}_3x_2 && (d_1=1,\ d_2=0)\\
&P(000)=x_1\oplus x_3\oplus x_2x_1\oplus x_3x_1\oplus x_3x_2 && (d_1=0,\ d_2=1)\\
&\qquad\ =x_1\oplus x_3\oplus x_3x_1\oplus x_3x_2 && (d_1=1,\ d_2=0)
\end{aligned}$$

이와 같이 Don't Care 조건을 결정하여 n_e가 가장 적을 뿐만 아니라 $T(C_k^i)$값 역시 가장 적은 RMC form을 찾아낼 수가 있다. 이 경우 $n_e=1$인 RMC form 중에서 $T(C_k^i)$가 가장 적은 RMC form은,

$d_1=0$, $d_2=0$ 일때

$Q=(111)$

$P(111)=\bar{x}_2\oplus\bar{x}_3\bar{x}_2\bar{x}_1$

$V(111)=(\odot,E,\odot)$

$T(111)=4$ 이다.

이처럼 Don't Care 조건을 갖는 함수는 조건에 따라 32개 RMC form을 구해야 하나 이 알고리즘을 적용함으로써 간단히 원하는 RMC form을 얻을 수 있다.

Ⅳ. 결 론

본 논문에서는 RMC form안에 literal 수가 적고 짝수개씩 나타나는 변수의 갯수가 적은 RMC form을 동시에 찾는 알고리즘을 고안함으로써 테스트셋트의 크기를 줄이고 회로의 복잡성도 줄일 수 있음을 보였다. 또한, Don't Care 조건을 갖는 함수의 RMC form 결정도 이 알고리즘을 적용해서 조건을 결정함과 동시에 RMC form을 얻을 수가 있다.

參 考 文 獻

1. S. B. Marinkovic and Z. Tosic, "Algorithm for minimal polarized polynomial form determination," IEEE Trans. Comput., vol. C-23, pp. 1313-1315, Dec. 1974.
2. K. L. Kodandapani and R. V. Setlur, "A note on minimal Reed-Muller canonical forms of switching functions," IEEE Trans. Comput., vol. C-26, pp. 310-313, Mar. 1977.
3. S. M. Reddy, "Easily testable realizations for logic functions," IEEE Trans. Comput., vol. C-21, pp. 1183-1188, Nov. 1972.
4. E. W. Page, "Minimally Testable Reed-Muller canonical Forms," IEEE Trans. Comput., vol C-29, pp. 746-750, Aug. 1980.