論 文

論 文 68—5—4—1

組合論理函數의 TANT回路에 依한 合成

(A Synthesis of Combinational Logic with TANT Networks)

高 瓊 植*
(Koh, Kyung Shik)

要 約

TANT回路라함은 眞入力만을 갖는 NAND게이트만으로 構成되는 3段回路를 말한다. 本論文에서는 任意의 Boole函數에 대한 最小數게이트의 TANT回路를 發見하는 方法을 提示한 것이다.

合成節次의 첫 단계는 Quine-McCluskey의 節次 또는 其他方法에 의하여 essential prime implicants(EPI)를 定하고 consensus算法을 適用하여 EPI와 同一한 頭部를 갖는 prime implicants(P I)를 유도하는 것이다. 두째 단계로 同一한 頭部를 갖는 E P I 및 P I를 統合하고 有用한 尾部要素를 發生시키는 것이다. 그 다음에 이들 尾部要素중에서 共通要素를 選定하는데 이 단계는 C-C表를 利用하는 것과 相通한 點이 있다. 마지막 단계로 冗長한 P I를 削除함으로써 入力數를 줄이는 것이다. 이 方法에 의하면 入力數가 5 및 6의 경우에는 手動的으로 容易하게 解를 얻을 수 있다.

ABSTRACT

A TANT network is a three-level network composed solely of NAND gates having only true (i.e. uncomplemented) inputs. The paper presents a technique for finding for any given Boolean function a least-cost TANT network.

The first step of the technique is to determine essential prime implicants(EPI) by Quine-McCluskey procedure or other methods and generate prime implicants(PI) having the same head as any one of EPI by consensus operation. The second step is to combine EPI and PI having the same head and generate usable tail factors. The next step is to select common factors among the usable tail factors. The selection phase is analogous to the use of C-C table. The last step is to minimize inputs by deleting the redundant PI. The technique permits hand solution of typical five- and six-variable problems.

I. 序 論

最近 超小形論理回路의 實用化에 따라 이에 便利한 NAND素子에 의한 論理設計가 問題가 되고 있다. 即 最小數의 NAND素子만을 사용하여 論理回路를 設計한다면 回路의 單純性과 아울러 經濟的이기 때문에 이에 대한 硏究가 進行되고 있으며 이 問題에 관련된 論文도 여러 篇 發表되고 있다(1)-(7). 그러나 아직까지 絕對的으로 最小數의 NAND素子를 사용하여 論理回路를 設計할 수 있는 代數的 方法은 發見되지않고 있다.

本論文에서는 모든 論理函數를 眞入力만을 갖는 NAND素子만으로 構成되는 3段回路 即 TANT回路에 의하여 實現시키는 方法을 論하는 것

*仁荷工科大學 電氣工學科
Dept. of Electrical Eng., Inha Institute of Technology
接受日字：1968年 12月 30日

이다. TANT回路에 의한 論理回路實現은 반드시 最小數의 게이트가 든다는 保證은 없지만 遲延時間이 가장 짧으며 系統的인 解法을 求할 수가 있다. 뿐만 아니라 入力變數가 그리 많지 않을 경우에는 NAND要素만으로 構成되는 最簡型의 3段回路이야 말로 어떤 基準下에서도 가장 要望되는 回路인 것이다.

다음에 本論文에서 사용될 表現法을 實例를 들어 說明하기로 한다. 그림1에 있어서 (a)는 NAND素子의 實地回路, (b)는 이것을 象徵하는 記號이며

$$f=(ab)'$$

를 그 出力으로 한다.

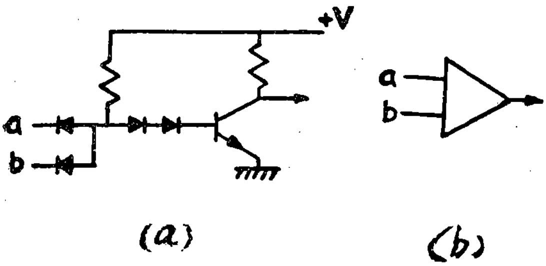


그림1. NAND素子 (a) 實地回路 (b) 記號
Fig. 1. NAND element. (a) practical circuit (b) symbol

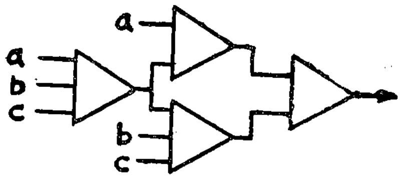


그림2. 論理函數 f=ab′+ac′+a′bc에 대한 最小 TANT回路
Fig. 2. A minimal TANT network for the function f=ab′+ac′+a′bc.

그림2는 記號로 표시한 TANT回路의 構成例이며 論理函數

$$f=ab'+ac'+a'bc \tag{1}$$

를 最小數의 게이트를 사용하여 實現시킨 것이다. TANT回路構成에 있어서 모든 NAND gate의 外部入力은 眞入力이고 첫째段은 論理函數중의 相補變數(complemented variables)를 얻기 위한 否定(negation)의 역할을 담당하는 것이고 세째段은 出力을 얻기 위한 單一 NAND素子로 되어 있는 것이다. 그림2의 TANT回路는 4개의 NAND 게이트와 10개의 入力端子로 構成되는데 그 出力을 확인하면

$$\begin{aligned} f&=\{[a(abc)']'[bc(abc)']'\}' \\ &=a(abc)'+bc(abc)' \\ &=a(a'+b'+c')+bc(a'+b'+c') \\ &=ab'+ac'+a'bc \end{aligned} \tag{2}$$

가 된다. 만약 式(1)의 表現을 그대로 채택하여 TANT回路로 實現시킨다면 그림 3과 같이 되며 7개의 NAND 게이트와 13개의 入力端子가 소요된다.

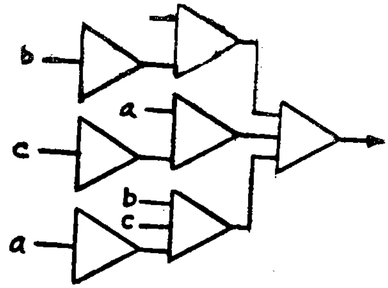


그림3. 論理函數 f=ab′+ac′+a′bc에 대한 TANT 回路例
Fig. 3. A TANT network for the function f=ab′+ac′+a′bc.

本論文의 目的은 위의 說明例에서 보는 바와 같이 任意의 論理函數를 最小數의 게이트를 사용한다는 基準(gate-cost criterion)下에서 TANT回路로 實現시키는 方法을 究明하는데 있으며 아울러 最小數의 入力端子를 사용한다는 基準(input-cost criterion)下에서도 考察하는 것이다.

Ⅱ. 本 論

1) 論理函數의 最小和表現

任意의 論理函數를 最小數의 NAND素子를 사용하여 TANT回路로 實現시키는데 있어서 첫 단계는 그 論理函數를 最簡型의 論理積의 和 即 最小和(minimal sum)의 型式으로 表現하는 것이다. 이것은 그 函數의 EPI를 求하는 題問로 歸着되며 이에 관해서는 여러 論文이 發表되고 있다(8)-(11). 지금 論理函數

$$f=(a,b,c,d,e)=\Sigma(1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 9,\ 11,\ 16,\ 17,\ 18,\ 19,\ 20,\ 21,\ 24,\ 25,\ 26,\ 27,\ 28,\ 29) \quad (3)$$

을 생각할 때 가령 Quine-McCluskey의 節次에 의하여 EPI를 구하여 最小和의 表現을 하면

$$f(a,b,c,d,e)=ac'+ad'+a'b'd+b'cd'+ce' \quad (4)$$

式(4)를 TANT回路로 實現시킨다면 게이트數는 다음 式에 의하여 10임을 알 수 있다. 即

$$E+N+1=5+4+1=10$$

여기서 E는 論理積의 項數, N은 相補變數의 數를 의미한다. 그러나 이 TANT回路는 式(3)의 論理函數를 實現시키는데 있어서 게이트數가 最少인 경우는 아니며 뒤에서 論하는 바에 의하여 8개의 게이트로 足함을 알 수 있다.

다음에 本論文에서 사용될 用語를 Gimpel (7)에 準하여 定義한다.

(定義) 論理表現 $HT'_1T'_2\cdots\cdots T'_n$에 있어서 H는 非相補變數 또는 非相補變數의 積 또는 Boole 常數 1을, 各 T'_i는 相補變數 또는 非相補變數의 積의 相補(또는 否定)를 나타낸다고 한다. 이때 H를 그 論理表現의 頭部(head), $T'_1T'_2\cdots T'_n$를 尾部(tail), 各 T'_i를 尾部要素(tail factor)라고 定義한다.

例컨대 論理表現 abc'(de)'에 있어서 ab는 頭部, c'(de)'는 尾部이고 c', (de)'는 각각 尾部要素이다. 또 論理表現 a'b'(cd)'(ef)'에 있어서 頭部는 1, 尾部는 a'b'(cd)'(ef)'이고 a', b', (cd)', (ef)'는 각각 尾部要素이다.

2) Consensus 算法에 의한 有用 PI의 誘導

AND 및 OR 게이트로 構成되는 2段回路에 의하여 論理函數를 實現시키는 경우와 달라서 TANT回路에 의하여 論理函數를 實現시키는데는 最小和表現에 의한 것이 반드시 게이트數가 最少가 되지는 않는다. TANT回路에 의할 때는 EPI와 同一한 頭部를 갖는 PI를 導入하여 頭部가 同一한 것을 統合함으로써 共通要素가 나타나게 되어 게이트數가 節減되는 경우가 생긴다. EPI와 同一한 頭部를 갖는 PI를 유도하는데는 consensus算法을 이용한다.

(定義) 두 論理表現 $P=xy_1^*y_2^*\cdots y_n^*$ 및 $Q=x'z_1^*z_2^*\cdots z_m^*$에 있어서 어떤 i와 j에 대해서는 $y_i^*=z_j^*$는 成立하지만 $y_i^*=(z_j^*)'$는 成立하지 않는다고 한다. 이때 P와 Q의 consensus라 함은 (P*Q로 表記) $y_1^*y_2^*\cdots y_n^*z_1^*z_2^*\cdots z_m^*$을 말한다.

例컨대 $abc'd*a'be'f=bc'de'f$는 可能하지만 $abc'd*a'b'e'f$는 존재할 수 없다.

다음에 式(4)로 表示되는 函數의 各項에 反復的으로 consensus算法을 適用하여 EPI와 同一한 頭部를 갖는 PI를 유도하기로 한다. 여기서 便宜上 EPI를 다음과 같이 나열하고 番號를 붙이기로 한다.

	a	b	c	d	e	
(1)	1	—	0	—	—	
(2)	1	—	—	0	—	
(3)	0	0	—	1	—	
(4)	—	0	1	0	—	
(5)	—	—	0	—	1	(以上은 EPI)
(6)	—	0	0	1	—	(1)*(3)
(7)	0	0	1	—	—	(3)*(4)
(8)	—	0	—	0	1	(4)*(5)
(9)	0	0	—	—	1	(3)*(8)

위의 表에 있어서 橫線을 경계로하여 上半部는 EPI이고 下半部는 consensus 算法에 의하여 얻지는 EPI와 同一한 頭部를 갖는 PI이다. 여기서 (1)*(4)=10—0—는 EPI(1) 및 (2)와 同一한 頭部를 갖기는 하지만 이것은 EPI(1)에 包含되므로 PI가 될 수 없으며 對象에서 除外되는 것이다. 이와같이 consensus 算法에 의하여 얻어지는 結果가 PI임을 確認하여야 한다.

다음 節次는 同一한 頭部를 갖는 EPI와 PI를 統合하는 것이다(附錄1 參照). 위의 例에서는 EPI(1)과 (2), EPI(3)과 PI(6), EPI(4)와 PI(7), EPI(5)와 PI(8)및 (9)가 각각 同一한 頭部를 가지며 統合한 結果 a(cd)', db'(ac)', cb'(ad)', e(bc)'(acd)'를 얻는다. 따라서 式(4)는

$$f=(a,b,c,d,e)=a(cd)'+cb'(ad)'+db'(ac)'+e(bc)'(acd)' \quad (5)$$

로 表示할 수 있다. 式(5)의 表現을 채택하여 TANT回路로 實現시킨다면 $E+N+1=4+6+1=11$ 이 되며 式(4)의 最小和表現의 경우보다 오히려 게이트數가 많아진다. 그러므로 다음 節

에서 說明하는 方法에 의하여 위의 函數表現을 變形하여 共通要素를 發見함으로써 게이트數의 節減을 꽤한다.

3). 共通要素의 發生

두 EPI에 consensus算法을 適用하여 얻는 PI는 그 두 EPI중의 어느 하나와 同一한 頭部를 갖는 경우가 있는데 이들 同一한 頭部를 갖는 EPI와 PI를 統合함으로써 다른 EPI와의 사이에 共通的인 尾部要素를 發生시킬 수 있다(附錄2參照). 지금 共通的인 尾部要素를 最大限으로 發見하기 위하여 式(5)를 基準으로 하여 다음과 같은 表를 만든다.

論理積	被覆되어야할尾部要素	可能한 對象尾部要素	
a(cd)′	(cd)′	(cd)′, (acd)′	(第1行)
cb′(ad)′	b′	b′, (bc)′	(第2行)
	(ad)′	(ad)′, (acd)′	(第3行)
db′(ac)′	b′	b′, (bd)′,	(第4行)
	(ac)′	(ac)′ (acd)′	(第5行)
e(bc)′(acd)′	(bc)′	(bc)′, (bce)′	(第6行)
	(acd)′	(acd)′, (acde)′	(第7行)

위의 表에서 第3列의 可能한 對象尾部要素라 함은 그 中의 어느 것을 尾部要素로 취해도 무방하다는 것이다. 가령 第1行의 경우 (ad)′ 대신에 (acd)′를 취할 때는 그 頭部와의 論理積을 생각하면

$$a(acd)'=a[a'+(cd)']=aa'+a(cd)'=a(cd)'$$

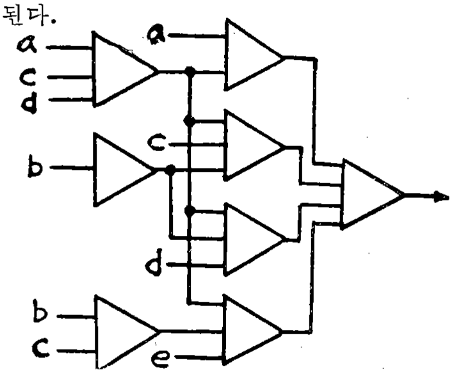
가 되어 本來의 論理積을 얻는다. 그러므로 위의 表를 利用하여 가장 많은 論理積에 共通的으로 들어갈 수 있는 尾部要素를 對象尾部要素중에서 選擇하면 게이트數를 節減할 수 있는 것이다. 觀察에 의하여 우선 (acd)′가 선택되며 따라서 第1行에서 (cd)′, 第3行에서 (ad)′, 第5行에서 (ac)′, 第7行에서 (acde)′가 除去된다. 나머지 行을 고찰할 때 第2行에서 b′, 第4行에서 b′, 第6行에서 (bc)′를 선택할 수도 있고 또는 第2行에서 (bc)′, 第4行에서 b′, 第6行에서 (bc)′를 선택할 수도 있다. 따라서 TANT表現으로서는

$$f=a(acd)'+cb'(acd)'+db'(acd)'+e(bc)'(acd)' \quad (6)$$

또는

$$f=a(acd)'+c(bc)'(acd)'+db'(acd)'+e(bc)'(acd)' \quad (7)$$

의 두 式中 어느것을 취해도 무방하며 다 같이 게이트數는 $E+N+1=4+3+1=8$이 된다. 따라서 式(4)의 最小和表現의 경우보다 게이트가 2개 節減되며 TANT回路로 實現시키면 그림4(a) 및 (b)와 같이 되며 總入力數는 어느 경우에도 21이 된다.



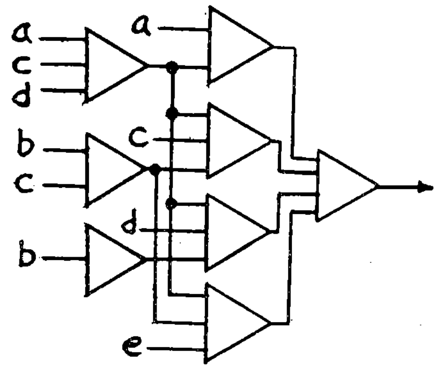


그림4. 本文式(6) 및 (7)의 TANT回路實現

Fig. 4. TANT networks for the functions shown in expression (6) and (7)

共通尾部要素의 選定에 있어서 頭部가 1인 統合表現에 單一變數尾部要素가 있을 때는 이 尾部要素는 最小數게이트 TANT表現에 반드시 包含된다 (7). 그러므로 이 點을 고려할 때는 選定節次가 상당히 간단해지는데 다음에 그 例를 들기로 한다.

$$f(a,b,c,d)=a'b'c'+a'b'd'+b'c'd'+bca'+abd'+acdb' \quad (8)$$

와 같은 最小和表現의 論理函數에 consensus 算

法을 適用하여 EPI와 同一한 頭部를 갖는 PI를 求하면 다음과 같다.

	a	b	c	d	
(1)	0	0	0	—	
(2)	0	0	—	0	
(3)	—	0	0	0	
(4)	0	1	1	—	
(5)	1	1	—	0	
(6)	1	0	1	1	(以上은 EPI)
(7)	—	1	1	0	(4)*(5)

EPI (1), (2), (3)을 統合하면 b'(ac)'(ad)'(cd)', EPI(4)와 PI(7)을 統合하면 bc(ad)'를 얻으며 다음表에 의하여 共通素子를 구한다.

論理積	被覆되어야할尾部要素	可能한 對象尾部要素
b'(ac)'(ad)'(cd)'	b'	b'
	(ac)'	(ac)'
	(ad)'	(ad)'
	(cd)'	(cd)'
bc(ad)'	(ad)'	(ad)', (abd)', (acd)'. (abcd)'
abd'	d'	d', (ad)', (bd)', (abd)'
acdb'	b'	b'

위의 表에서 論理積 acdb'의 可能한 對象尾部要素로서는 b'外에 (ab)', (bc)', (db)', (abc)', (abd)', (bcd)', (abcd)'도 있을 수 있지만 頭部가 1인 統合表現 b'(ac)'(ad)'(cd)'에 單一變數尾部要素b'가 있으므로 b'만을 對象尾部要素로 취한 것이다. 最大限으로 共通的인 尾部要素를 順次的으로 선정하면 (ad)', b', (ac)', (cd)'를 얻으며 다음과 같은 TANT表現을 얻는다.

$$f(a,b,c,d)=b'(ac)'(ad)'(cd)'+bc(ad)'+ab(ad)'+acdb' \qquad (9)$$

4). 入力數에 對한 考察

다음에 入力數를 節減할 수 있는지의 如否에 대하여 고찰한다. 式(6)및 式(7)의 두 TANT表現에 있어서 尾部要素b'는 不可避한 것이므로 가령 式(6)의 경우를 생각할 때 尾部要素(bc)'를 보면 (bc)'=b'+c'로 b'를 包含하므로 다음과 같이 b'를 摘出하는 방향으로 유도하면

$$e(bc)'(acd)'=eb'(acd)'+ec'(acd)'$$
$$=eb'(acd)'+ec' \qquad (10)$$

그런데 e를 頭部로 하는 EPI는 ec'뿐이며 이것은 式(10)의 第2項과 같으며 따라서 第1項은 冗長한 (redundant) 것이므로 이를 버리면 式(6)은 다음과 같이 修正된다.

$$f=a(acd)'+cb'(acd)'+db'(acd)'+ec' \qquad (11)$$

式(11)을 TANT回路로 實現시키면 그림5와 같으며 게이트數는 8이고 入力數는 19로 되어 그림4의 경우보다 入力數가 2개 節減된다.

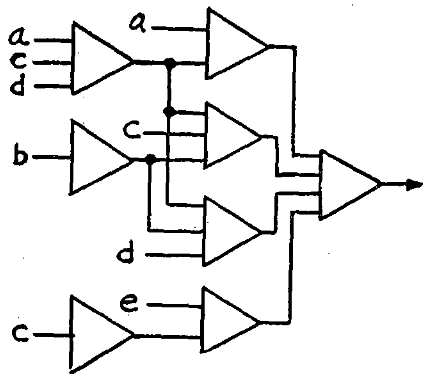


그림5. 本文式(3)에 대한 最小 TANT回路
Fig.5. A minimal TANT network for the expression(3)

또 다른 例로서 다음과 같은 EPI에 의한 最小和表現의 論理函數를 생각한다.

$$f(a,b,c,d,e)=ac'+ad'+bd'+cb'e' \qquad (12)$$

Consensus算法을 적용하여 이들 EPI와 同一한 頭部를 갖는 PI를 유도하면 ab'e', cd'e'를 얻는다. 다음에 同一한 頭部를 갖는 것들을 統合하여 共通的인 尾部要素를 선정하여 整理하면 다음과 같다.

$$f=a(bcd)'(cde)'+b(bd)'+ce'(bd)' \qquad (13)$$

또는 $f=a(bcd)'(cde)'+bd'+ce'(bcd)' \qquad (14)$

위의 두 表現은 다 같이 게이트數는 8이고 總入力數는 式(13)의 경우에는 20, 式(14)의 경우에는 19이다. 그러나 어느 경우에도 尾部要素e'는 不可避한 것이므로 가령 式(13)을 생각할 때 第1項에서 e'를 摘出해 본다.

$$a(bcd)'(cde)'=a(bcd)'[(cd)'+e']$$
$$=a(cd)'+ab'e' \qquad (15)$$

그런데 a를 頭部로 하는 EPI는 ac'및 ad'이며

이것을 統合하면 $ac'+ad'=a(cd)'$이며 따라서 式(15)의 第2項은 冗長한 것이다. 이것을 버리면 式(13)은 다음과 같이 된다.

$$f(a,b,c,d,e)=a(cd)'+b(bd)'+ce'(bd)' \quad (16)$$

式(16)을 TANT回路로 實現시키면 그림6과 같으며 게이트數는 7이고 入力數는 15가된다.

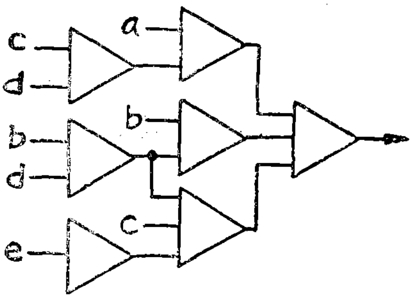


그림6. 本文式(12)에 대한 最小 TANT回路
Fig. 6. A minimal TANT network for the expression (12)

5). TANT回路의 實現節次에 대한 要約

(a). Quine-McCluskey의 節次 또는 其他方法에 의하여 EPI를 결정한다

(b). 反復的으로 consensus算法을 適用하여 EPI와 同一한 頭部를 갖는 PI를 유도한다.

(c). 同一한 頭部를 갖는 EPI및 PI를 統合한다. 이때 統合된 表現이 EPI및 PI를 統合하지 않고 그대로 사용할때 보다 素子數가 적게 드는가를 確認한다.

(d). 被覆되어야 할 尾部要素 및 可能한 對象尾部要素를 本文의 要領에 의하여 表記한다. 이때 頭部가 1인 統合表現의 單一變數尾部要素는 반드시 最小數게이트 TANT回路에 채택된다.

(e). 對象尾部要素中에서 最大限으로 共通的인 要素를 順次的으로 選定한다. 選定된 尾部要素中에 他尾部要素를 包含하는 것이 있을 때는 이것을 分離하여 冗長한 部分이 發見되면 이를 除去함으로써 入力數를 節减하는 方向으로 유도한다.

(f). TANT表現式을 完成하고 實地回路로 實現시킨다.

Ⅲ. 結　論

單一出力의 任意의 組合論理函數를 最小數의 NAND게이트를 사용하며 TANT回路로 實現시키는 方法을 考察하였다. 本論文의 方法에 의하면 入力變數가 5내지 6程度의 경우에는 그리 큰 手苦를 함이 없이 手動的으로 正確한 解를 얻을 수 있다. 入力數가 많아짐에 따라 手動的으로 처리하기에는 번잡성을 면할수 없으며, 電子計算機를 사용할 경우 本論文의 節次에 의하여 programing이 可能한지의 如否에 대해서는 앞으로의 檢討가 필요할 것으로 생각한다.

附　錄

1) 同一한 頭部를 갖는 PI의 統合

同一한 頭部를 갖는 여러 PI의 統合 節次를 實例를 들어 考察한다. 頭部를 除外한 尾部가 다음과 같은 一般型의 경우를 우선 생각한다.

a	b	c	d	e	f
—	0	0	—	—	0
0	—	0	0	—	0
0	—	—	0	0	—
0	0	—	—	0	0

처음 두 項을 統合하면

$$c'f'(b'+a'd')=c'f'(ab)'(bd)'$$

끝 두 項을 統合하면

$$a'e'(d'+b'f')=a'e'(bd)'(df)'$$

위의 두 式을 다시 統合하면

$$\begin{aligned}&c'f'(ab)'(bd)'+a'e'(bd)'(df)'\\&=(bd)'[c'f'(ab)'(df)'+a'e'(df)'(ab)']\\&=(bd)'(ab)'(df)'(c'f'+a'e')\\&=(bd)'(ab)'(df)'(ac)'(af)'(ce)'(ef)'\end{aligned}$$

다음에 特殊型의 경우를 생각한다. 여기서 中間計算을 略하고 結果만 쓰기로 한다.

(a) 한 bit만 0이고 나머지 bit는 무두 —인 n개의 尾部의 경우

$$\left.\begin{array}{cccccc}x_1 & x_2 & x_3\cdots\cdots\cdots\cdots & & x_{n\ 1} & x_n\\ 0 & — & —\cdots\cdots\cdots\cdots & & — & —\\ — & 0 & —\cdots\cdots\cdots\cdots & & — & —\\ — & — & 0\cdots\cdots\cdots\cdots & & — & —\\ \cdots & \cdots & \cdots\cdots\cdots\cdots & & \cdots & \cdots\\ — & — & —\cdots\cdots\cdots\cdots & & 0 & —\\ — & — & —\cdots\cdots\cdots\cdots & & — & 0\end{array}\right\} n\text{개}$$

이 n項은 $(x_1\,x_2\,x_3\cdots\cdots x_{n-1}\,x_n)'$와 같이 한 項으로 統合된다.

(b) 서로 다른 組合의 두 bit만 0이고 나머지

$n-2$ bit는 모두 一인 $_nC_2$개의 尾部의 경우

$$\left.\begin{array}{cccccc}
x_1 & x_2 & x_3 & \cdots\cdots & x_{n-1} & x_n \\
0 & 0 & - & \cdots\cdots & - & - \\
0 & - & 0 & \cdots\cdots & - & - \\
\cdots & \cdots & \cdots & \cdots\cdots & \cdots & \cdots \\
0 & - & - & \cdots\cdots & 0 & - \\
0 & - & - & \cdots\cdots & - & 0 \\
- & 0 & 0 & \cdots\cdots & - & - \\
\cdots & \cdots & \cdots & \cdots\cdots & \cdots & \cdots \\
- & 0 & - & \cdots\cdots & 0 & - \\
- & 0 & - & \cdots\cdots & - & 0 \\
\cdots & \cdots & \cdots & \cdots\cdots & \cdots & \cdots \\
- & - & - & \cdots\cdots & 0 & 0
\end{array}\right\} nC_2\text{개}$$

이 $_nC_2$개의 項은

$(x_1x_2x_3\cdots x_{n-1})'$ (x_n을 포함하지 않음) $(x_1x_2x_3\cdots x_n)'\cdots\cdots$ (x_{n-1}을 포함하지 않음) $(x_2x_3\cdots x_{-n}x_n)'$ (x_1을 포함하지 않음)

$$=\frac{n}{\Pi}(n-1\text{개의 變數의 論理積})'$$

와 같이 n項의 積으로 統合되며 各項은 n개의 變數中 서로 다른 組合의 n−1개의 論理積의 否定의 型을 갖는다.

(c) 서로 다른 組合의 m bit가 0이고 나머지 n−m bit는 一인 $_nC_m$개의 尾部의 경우

앞의 두 경우를 擴張한 것에 해당하며 다음과 같이 $_nC_{m-1}$項의 論理積으로 統合된다.

$$\frac{nCm\text{-}1}{\Pi}(n-m+1\text{개의 變數의 論理積})'$$

여기서 各項은 n개의 變數中 서로 다른 組合의 n−m+1개의 變數의 論理積의 否定의 型을 갖는다.

同一한 頭部를 갖는 여러 PI를 統合함에 있어서 留意할 점은 TANT回路實現에 있어서 統合表現이 개개의 PI를 그대로 사용할 때 보다 반드시 素子數가 節減되지 않는다. 가령

$$g=a'b'c'+d'e'f'$$

를 例로 들면 그대로 사용할때는 素子數는 $E+N+1=2+6+1=9$인데 반하여 統合表現을 생각하면

$$g=(ad)'(ae)'(af)'(bd)'(be)'(bf)'(cd)'(ce)'(cf)'$$

와 같이 되며 素子數는 $E+N+1=1+9+1=11$이 된다.

2). Consensus算法에 의한 共通要素의 發生.

Consensus算法에 의하여 EPI와 同一한 頭部를 갖는 PI를 얻으려면 두 EPI는 다음과 같은 一般型을 가져야 한다.

$$KMa'b'\cdots\cdots\cdots\cdots\cdots\cdots k'$$
$$Kap'q'\cdots\cdots\cdots\cdots\cdots\cdots s'$$

여기서 K는 두 EPI의 共通的인 bit를 대표하고 M은 眞入力을 대표하는 것이다. 이 두 EPI에 consensus 算法을 적용하면 $KMb'\cdots k'p'q'\cdots s'$를 얻으며 이것은 첫 EPI와 同一한 頭部를 갖는다. 다음에 同一한 頭部를 갖는 EPI와 PI를 統合함으로써 共通要素가 發生하는 경우를 고찰한다.

(a). (1) $KMa'b'\cdots\cdots k'$

(2) $Kap'q'\cdots\cdots s'$ (以上은 EPI)

---

(3) $KMb'\cdots\cdots k'p'q'\cdots s'$ (1)*(2)

EPI(1)과 PI(3)을 統合하면

$$KMb'\cdots\cdots k'(a'+p'q'\cdots\cdots s')=KMb'\cdots\cdots k'(ap)'(aq)'\cdots\cdots(as)'$$

일방 EPI(2)를 變形하면

$$Kap'q'\cdots\cdots s'=Ka(ap)'(aq)'\cdots\cdots(as)'$$

가 되어 共通要素$(ap)'$, $(aq)'$, ……, $(as)'$가 나타나며 素子數가 한 개 節減된다.

(b). (1) $KMa'T_1'$

(2) KaT_2'

(3) KaT_3' (以上은 EPI)

---

(4) $KMT_1'T_3'$ (1)*(2)

(5) $KMT_1'T_2'$ (1)*(3)

여기서는 便宜上 각각 서로 다른 尾部要素를 T_i'로 대표시켰다.

(1), (4), (5)를 統合하면 $KMT_1'(aT_2T_3)'$

(2), (3)을 統合하면 $Ka(T_2T_3)'=Ka(aT_2T_3)'$

즉 共通要素 $(aT_2T_3)'$가 나타나며 素子가 한 개 節減된다.

(c). (1) $KaT_1'T_4'$

(2) KaT_2'

(3) $KMa'T_3'$

(4) KMT_4' (以上은 EPI)

---

(5) $KMT_2'T_3'$ (2)*(3)

(1), (2)를 統合하면

$$Ka(T_1T_2)'(T_2T_4)'=Ka(T_1T_2)'(aT_2T_4)'$$

(3), (4), (5)를 統合하면

$$KM(T_3T_4)'(aT_2T_4)'$$

즉 共通要素 $(aT_2T_4)'$가 나타나며 素子가 한

개 節減된다.

(d). (1) $K a T_1'$

(2) $K a T_2'$

(3) $K M a' T_3'$

(4) $K M T_4'$ (以上은 EPI)

$K M T_1' T_3'$ (1)*(3)

$K M T_2 T_3'$ (2)*(4)

(1), (2)를 統合하면 $K a (T_1 T_2)'$

(3), (4), (5), (6)을 統合하면

$K M (T_3 T_4)' (a T_1 T_2 T_4)'$

이 경우에는 共通要素가 나타나지 않는다.

参考文獻

(1) Hellerman, L., "A catalog of three-variable OR-invert and AND-invert logical circuits", IEEE Trans. on Electronic Computers, vol. EC—12, pp. 198-233, June 1963.

(2) G. A. Maley and S. Ogden, "Minimal three-level NAND circuitry", IBM Product Development Lab., Data Systems Division, Poughkeepsie, N. Y., Tech. Rept. TR 00. 933, November 7, 1962.

(3) G. A. Maley and J. Earle, "The logical design of transistor digital computers", Englewood Cliffs, N. J. Prentice-Hall, 1963.

(4) D. T. Ellis, "A synthesis of combinational logic with NAND or NOR elements", IEEE Trans. on Electronic Computers, vol. EC-14, pp. 701-705, October 1965.

(5) R. E. Burke and J. G. van Bosse, "NAND-AND circuits", IEEE Trans. on Electronic Computers, vol. EC-14, pp. 63-65, February 1965.

(6) R. A. Smith, "Minimal three-variable NOR and NAND logic circuits", IEEE Trans. on Electronic Computers, vol. EC-14, pp. 79-81, February 1965.

(7) J. F. Gimpel, "The minimization of TANT networks", IEEE Trans. on Electronic Computers, vol. EC-16, pp. 18-38, February 1967.

(8) E. J. McCluskey, "Minimization of Boolean functions", Bell Sys. Tech. J., vol. 35, pp. 1417-1444, November 1956.

(9) I. B. Pyne and E. J. McCluskey, "The reduction of redundancy in solving prime implicant tables", IRE Trans. on Electronic Computers, vol. EC-11, pp. 473-482, August 1962.

(10) J. F. Gimpel, "A reduction technique for prime implicant tables", IEEE Trans. on Electronic Computers, vol. EC-14, pp. 535-541, August 1965.

(11) F. Luccio, "A method for the selection of prime implicants". IEEE Trans. on Electronic Computers, vol. EC-15, pp. 205-212, April 1966.

(12) P. Tison, "Generalization of consensus theory and application to the minimization of Boolean function", vol. EC-16, pp. 446-456, August 1967.

(13) A. R. Meo, "Modular tree structure", IEEE Trans. on Computers, vol. EC-17, pp. 432-442, May 1968.

(14) E. J. McCluskey, "Introduction to the theory of switching circuits, p. 165-174, McGraw-Hill Book Co. N. Y., 1965.