

TANT 회로망의 계산기 이용 합성에 관한 연구 (A Study on the Computer-Aided Synthesis of TANT Network)

安 光 善* ; 朴 圭 泰**
(Ahn, Gwang Seon and Park, Kue Tae)

要 約

스위칭함수는 만능게이트인 NAND gate를 이용하여 3단논리회로로 구성될 수 있으며, 3단논리회로이면서 眞의 입력값만을 허용하는 회로를 TANT (three-level AND-NOT network with true inputs) 회로망으로 정의하여 사용하고 있다.

본 연구는 TANT 회로망의 설계에 있어 최적화과정의 새 방법을 제안한 것으로 CPPI 혹은 EPPI를 만들면서 C-C 표를 쓰지 않고 직접 최적의 TANT 회로망을 구하는 방법이다.

본 알고리즘은 스위칭함수의 입력변수가 4개 혹은 5개까지 수작업(手作業)으로 가능하지만 그 이상의 것은 컴퓨터에 의해 처리될 수 있으며 이를 위해 CAD (computer aided design) 소프트웨어 패키지를 FORTRAN IV로 작성하였다.

Abstract

Any switching function can be constructed with universal building block of NAND gate. Three-level AND-NOT logic networks with only true inputs are called TANT networks.

Systematic approach to TANT minimization starts from the UF type minterm with the smallest subscript and ends when UF type minterms are all covered. Optimal PEI is composed of CPPI or EPPI without C-C table.

The algorithm in this work is useful in solving TANT optimization problem of four or five variables by hand solution. When variable are six or more, it is required to be solved by computer. A CAD software package of this algorithm with FORTRAN IV language is made to solve such problems.

1. 서 론

논리회로망의 설계에 있어서 가장 중요하게 취급되어야 할 사항은 구성게이트의 수를 최소화시키면서 이들의 상호연결에 관계되는 게이트 입력수를 최소화시키는 것이다.^[1, 2, 3] 이렇게 함으로써 간략화된 회로망은 오동작이 줄어들어 신뢰도가 높아지게 된다. 또한 회로구성의 간략화로 고장검출이 용이해지며 가격도 낮아진다.

이상의 조건외에 일반적으로 취급되어지는 제한은 회로의 전달시간을 줄이기 위한 논리단의 수를 최소로 줄이는 문제이며, 또한 회로에 진값을 가진 입력만을 허용하는 것이다.

어떤 스위칭함수도 만능게이트인 NAND를 이용하여 3단논리회로로 구성할 수 있으며, 3단회로이면서 진의 입력값만을 허용하는 논리회로를 TANT 회로망이라 정의하여 사용한다.^[4-9]

TANT 회로망은 그림 1에서 볼 수 있으며 그림 1(a)는 Karnaugh Map 이고 그림 1(b)는 그의 최적 TANT 회로망이다.

NAND 나 NOR 게이트를 이용한 회로망구성에 대한 연구는 복보수 기법(double complement technique), 부울대수식 변환방법 및 도해법(map factoring)에 기초를 두어 왔다.^[2, 3, 15, 16] 회로망의 간략

* 正會員, 경북대학교 공과대학 전자공학과
(Dept. of Electronics, Kyungpook National Univ.)

** 正會員, 연세대학교 공과대학 전자공학과
(Dept. of Electronics Eng. Yonsei Univ.)

接受日字: 1980年 6月 27日

WX \ Y	00	01	11	10
0	0 ⁰	0 ²	1 ⁶	1 ⁴
1	1 ¹	0 ³	0 ⁷	1 ⁵

(a)

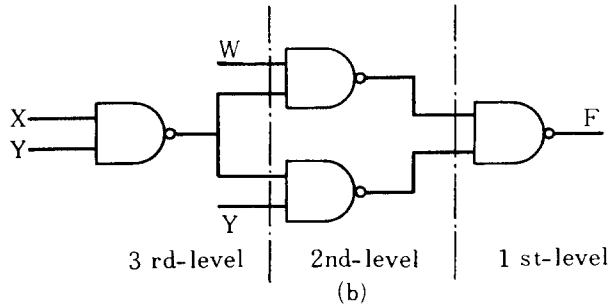


그림 1. 카누 맵(a)과 그의 최적 TANT 회로망(b)

Fig. 1. A Karnaugh map(a) and it's optimal TANT network(b).

화에 대한 체계적인 시도는 Gimpel^[4]에 의해 시작되었으며 이 방법은 제 2 단계이트가 되는 함수들이 후보로써 미리 정해진 다음 C-C 표 (cover and closure table)에 의해 선정되어 본 함수에 대한 최적 회로망이 구성된다. 여기서 이용된 C-C 표는 스위칭함수의 구성변수가 4개 이상이 될때 매우 방대해지므로 C-C 표를 다른 각도로 구성하는 방법이 Lee^[5]에 의해 연구되었다. Chakrabarti^[6]는 C-C 표 대신 보수함수 (complementary function)를 이용하였으며, Koh^[7]는 EPI(essential prime implicant)를 이용하여 최적회로망을 구하는 방법을 제시하였다. Layton^[9]은 Karnaugh Map을 이용하여 제 2 단 및 제 3 단에 필요한 게이트 후보들을 만들고 최소화이론을 가지고 최적 TANT 회로망을 구성하였다.

위에서 언급된 모든 방법은 원 함수에 관계되는 모든 제 2 단계이트의 집합을 먼저 구하는 과정(확정)과 이와 관련된 제 3 단계이트와 연관지워 최적함수를 만들 수 있는 제 2 단계이트들의 부분집합을 골라 내는 과정(선정)으로 구성되고 있다. 이러한 분리된 방법은 원 함수의 간략화과정을 비효율적으로 만들게 되며, 적당한 방법을 이용하여 최적함수를 구성하는 게이트들의 집합을 직접 선정시킬 수 있는 방법이 필요하게 된다. 이렇게 하여 최적회로망에 사용되지 않는 게이트집단을 만드는 데 필요한 시간과 노력을 절약할 수 있다.

본 연구에서는 Karnaugh Map의 구성형태를 이용하여 진값을 가진 민텀들에 대한 통합개념을 적용하여 최적함수의 구성게이트 집합을 직접 만들 수 있었다. 조직적인 통합은 PSA(position search algorithm)에 의하며 이렇게 하여 전체적인 진행과정을 현저하게 줄일 수 있었다.

2. 최적 TANT 회로망의 구성이론

2-1. TANT 회로망의 해석

· 임의의 스위칭함수는 Quine McCluskey에 의하면

AND/OR 게이트로 구성할 수 있으며 그림 2(a)와 같이 2 단논리회로가 된다. 입력변수가 항상 진값만을 가질때는 그림 2(b)와 같이 3 단회로망이 된다. 디지털회로 구성 기본소자로서 그의 단순성과 확실성 및 생산가격의 저렴성때문에 NAND 혹은 NOR 게이트가 가장 많이 사용되며 그림 2(b)의 회로망에 de Morgan의 정리를 적용하면 다음과 같다.

$$F = WX' + WY' + YX' = \overline{\overline{WX'}} \overline{\overline{WY'}} \overline{\overline{YX'}}$$

이 식에 대한 회로를 NAND 게이트만으로 구성시키면 그림 2(c)가 되며 이것은 진입력만의 3 단논리 회로망인 TANT 회로망이 된다.

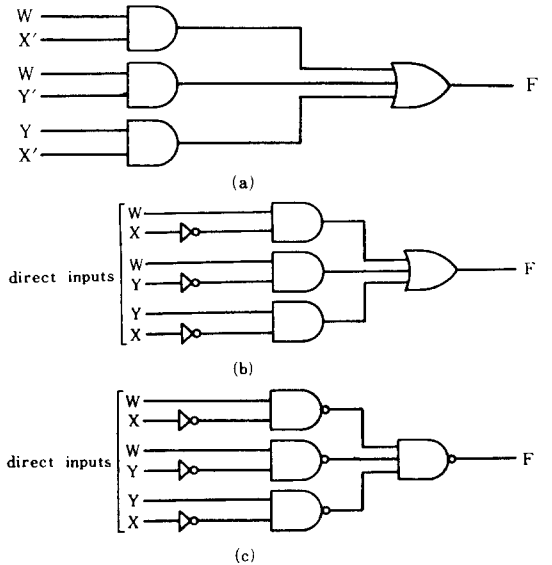


그림 2. 함수 F = WX'+WY'+YX'에 대한 논리회로망

Fig. 2. General configuration of logic circuit for F = WX'+WY'+YX'.

정의 : PE (permissible expression) 는 n ≥ 0 인 조건에서 T₀ T₁ T₂ ... T_n의 형태를 가지며 결과값은 0이 아닌 Boole 식이 되어야 한다. 여기서 T_i는 Boole 상수 1 혹은 진값만을 갖는

변수들의 적(積)으로 표시되며 T_0 는 헤드(head)라 불리우고, T_j '들은 테일(tail)이라 불리운다. PE 형태를 갖는 Boole 함수는 PEI (permissible implicant)로 불리운다.

예를 들면 $X'Y'Z'$, XYZ' , $V'(WXYZ')$ '등은 PEI이며 $X'+YZ'$, $XY(XY)'$, $X(1)'$ '등은 이에 속하지 않는다.

정의: TANT 식은 $E_1 + E_2 + E_3 + \dots + E_n$ 의 형태를 가지며 여기서 E_i 는 PE 형이 되어야 한다.

위에 정의된 TANT 식은 TANT 회로망과 완전한 대응관계를 갖게 된다. TANT 식에서 $T_0 T_1' T_2' \dots T_n'$ 의 형태를 갖는 E_i 는 제2 단계이트 집단을 형성하게 되며, 식 중 헤드인 T_0 는 제2 단계이트에 대한 직입력(直入力) 변수이고 테일인 T_j '들은 제3 단계이트가 되어 그 출력이 2 단계이트의 입력으로 사용된다. 여기서 보는 바와 같이 최적 TANT 회로망에서는 E_i 의 개수를 가능한 한 최소로 하여 제2 단계이트 수를 줄일 수 있으며 제3 단계이트가 되는 T_j '들을 공통으로 이용하여 전체적인 게이트 가격(gate cost)에 대한 최적 TANT 회로망이 구성되어질 수 있다.

2--2. TANT 회로망 구성 알고리즘

앞절에서 정의된 TANT 식은 여러 개의 PEI들로 구성되며 게이트 가격의 제한조건에 의한 최적의 TANT 식을 만들기 위해 E_i 의 수를 줄여야 한다.

정리 1: 같은 헤드(H)를 가진 PEI 들은 Boole 덧셈규칙에 의해 묶인다.

증명: PEI 인 E_1 및 E_2 를 각각 $H\xi_1$ 및 $H\xi_2$ 로 놓으면 $E_1 + E_2 = H(\xi_1 + \xi_2)$ $H\tau$ 이며 PEI의 정의로부터 ξ_1 및 ξ_2 는 보수형 변수이므로 de Morgan의 정리에 의해 복수 보수형 변수(compound complement variable) τ 로 변한다.

Quine McCluskey 방법에 의한 원 함수의 최적 표현식은 PI (prime implicant) 만으로 구성되며, 이러한 PI 들을 정리 1에 의해 재결합함으로써 최적 TANT 회로망을 만들게 된다. 그 결과로 남는 PEI항들은 적어도 한 개의 PI 들로 이루어져야 한다.^[4, 7]

정의: PPI(prime permissible implicant)란 PEI 중에서 임의의 구성 테일변수가 없다면 본 함수에 포함되지 않는 PEI 를 말한다.

정리 2: 모든 PI 는 PPI 이다.

증명: 어떤 함수 F에 대한 PI 는 변수의 적의 항(product term)이며, PI 를 구성하는 임의의 변수를 제거하면 원 함수에 포함되지 않는 새로운 변수적이 나온다. 따라서 모든 PI

는 PPI 가 된다.

PI 를 구하는 과정은 점차 값이 가장 작은 UF 형(uncomplement form)의 민텀이 정점(시작점)으로 선정되며 그것을 포함하는 PI 가 Karnaugh Map 상에서 민텀의 위치에 의해 직접 결정되므로 이러한 알고리즘을 PSA (position-search algorithm) 라고 한다. PSA에(의해서 정해지는 확장과정은 정점)민텀을 중심으로 그 주변의 민텀과 비교되며 입력변수들이 1개 변수만 다를 경우 축소되어지는 원리를 이용한다. 모든 민텀들은 n개의 다른 민텀들과 인접하며 FPD(forming PI direction) 확장방향을 가질 수 있으며 FPD는 다음 성질의 정수 집합

$$\{a_k\} = \left\{ \begin{array}{l} S \times 2^k \quad k = 0, 1, 2, \dots, n-1, S = \begin{array}{l} +1 \\ -1 \end{array} \\ \text{단 } k \text{ 번째 bit} = \begin{array}{l} 0 \\ 1 \end{array} \end{array} \right\}$$

이다.

확장과정 중 j개의 FPD 방향을 가지면 최대PI는 j-cube가 되어 2^j개의 민텀들이 묶이게 된다. 모든 FPD에 의한 결합이 전개된 후 최대민텀수를 포함하는 PI가 선정된다.

정의: 도미넌트 PPI(P_0)란 PPI(P_1)에 대하여 $P_0 P_1 = 0$ 및 $P_0 \geq P_1$ 의 성질을 갖는 PPI를 말한다.

정의: LPPI (large prime permissible implicant)란 정점이 되는 민텀을 포함하는 PPI 중에서 도미넌트 PPI를 말한다.

정리 3: 스위칭함수 F에 있어서 LPPI는 어떤 정점을 중심으로 같은 헤드를 가진 PPI들의 합으로 구해진다.

증명: $(\phi_1 + \phi_2 + \dots + \phi_r)(\psi_1 + \psi_2 + \dots + \psi_s) = 0$ 으로 놓고 $P = \phi_1 + \phi_2 + \dots + \phi_r$ 로 놓으면(단 $\phi_1, \phi_2, \dots, \phi_r$ 은 헤드가 H인 PPI이고 $\psi_1, \psi_2, \dots, \psi_s$ 는 헤드가 H가 아닌 PPI이다) P는 헤드가 H인 PPI이다. P_1 을 헤드가 H₁인 다른형의 PPI인 $P_1 = H_1 S$ 로 놓으면 S는 $\{\phi_1\}$ 나 $\{\psi_2\}$ 중의 어느 하나에 포함되기 때문에 전자의 경우는 $P_1 \leq P$ 가 되며 후자의 경우는 $P_1 P = 0$ 가 되어 P가 P_1 에 대해 도미넌트하다. 따라서 정의에 의해 P는 LPPI가 된다.

그림 1의 예에서는 3개의 PI 즉 YX' , WY' 및 WX' 이 존재하며 헤드는 Y 및 W 2종이므로 2개의 도미넌트 LPPI인 YX' 및 $W(XY)'$ 이 존재한다.

정의: CPPI (core prime permissible implicant)란 정점 민텀을 포함하는 LPPI가 하나뿐일때 이

LPPI를 말하며, EPPI(essential prime per-missible implicant)는 정점 민텀을 포함하는 LPPI 중에서 구성변수의 성질에 의해 선택된 1개의 LPPI를 말한다.

구성변수의 성질은 LPPI들의 입력수, 헤드 및 테일의 성질을 말하며 열 도미넌스(row dominance) 및 포함(covering)에 의하여 최적의 EPPI는 결정된다.

정리 4 : 모든 CPPI와 EPPI만으로 게이트값의 제한에 대한 최적 TANT 회로가 구성된다.

증명 : CPPI나 EPPI의 정의에 따라 위 정리는 자명해진다.

2-3. 최적 TANT 회로망의 구성

최적 TANT 회로망 구성에 있어 PEI의 탐색은 정점 민텀을 중심으로 시작되며 정점 민텀보다 높은 첨자를 가진 민텀들과 쌍을 이루어 간략화된다. 이 때문에 최적 TANT 회로에 대한 LPPI의 선정과정은 첨자의 값이 작은 UF형의 민텀부터 시작하여야 한다. 이렇게 하여 묶이는 LPPI는 헤드의 구성개수가 작으므로 여러개의 PI항들이 서로 합쳐서 많은 수의 UF형 민텀들을 포함시킬 수 있다. 또한 테일의 개수가 많으므로 그후에 만들어지는 LPPI를 처리할 때 EPPI를 발생시켜야 할 경우 많은 도움이 된다. 이렇게 하여 만들어지는 CPPI 혹은 EPPI는 최적 TANT 회로망의 구성게이트로 되며 이에 의해 포함된 민텀들은 모두 UF형에서 XF(don't care form)형으로 바뀌어진 후 다음 과정으로 넘어간다.

다음 과정의 정점이 되는 민텀은 남아있는 UF형 중 가장 작은 값의 첨자를 갖는 민텀이며 위와 같은 과정을 거쳐 CPPI 혹은 EPPI를 구성한 후 그곳에 포함된 UF형의 민텀들은 모두 XF형으로 바뀐다. 남아있는 UF형의 민텀이 없어질 때까지 이 과정이 반복되며 최종적으로 구해지는 CPPI 및 EPPI의 집단중에서 같은 헤드를 가진 것이 있으면 정리 1에 의해 함께 묶이어서 전체 게이트 수를 줄이게 된다.

CPPI와 EPPI는 변수의 구성에 따라서 여러개의 테일항을 가질 수 있다. 어떤 항이 n개의 헤드변수 및 m개의 테일변수로 구성된 PEI라면, 같은 값을 갖는 PEI는 2^m 개 만큼 생기며 이것들을 ATS(available tail set)로 정한다. CPPI 및 EPPI들은 각각 ATS를 가지며 각 항에 대한 최적 테일을 선택하여 게이트가격에 대한 최적의 TANT 회로망을 구성한다.

이상의 과정에서 최적 TANT 회로망을 얻을 수 있는 알고리즘을 정리하면 다음과 같다.

- 1) 가장 작은 첨자값을 가진 UF형의 민텀을 선택
- 2) 위의 민텀을 정점으로 { LPPI } 발생
- 3) { LPPI } 중 CPPI 혹은 EPPI의 선정
- 4) 앞의 3) 과정에 포함되는 UF형의 민텀을 모두 XF형으로 바꾼 후 1) 과정으로 돌아감. 모든 UF형의 민텀이 포함되면 5) 과정으로 넘어감
- 5) 선정된 항 중 같은 헤드를 가진 항들을 묶은 후 각 항들의 ATS로부터 최적 TANT 회로망을 만든다

3. CAD의 처리과정

앞에 서술된 TANT 회로망 구성 알고리즘은 스위칭함수의 구성변수가 6개 이상이 되면 컴퓨터를 이용한 설계가 필연적으로 요구된다.

최적 TANT 회로망을 설계하는 전체 처리과정은 그림 3에 표시하였으며 전 과정은 다음과 같이 크게 4 단계로 나눈다.

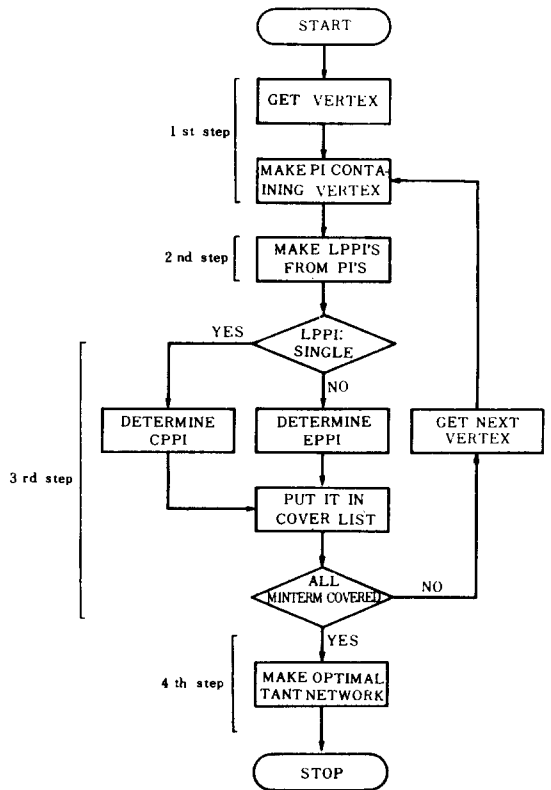


그림 3. 최적 TANT 회로망의 구성 과정에 대한 일반 유통도

Fig. 3. General flow chart of optimal TANT network synthesis procedure.

제 1 단계 : 스위칭함수의 UF 형을 갖는 민텀중에서 첨자의 값이 가장 적은 것부터 정점으로 선택되어 P SA 알고리즘에 의해 FPD 방향의 민텀들과 결합되어 확장된다. 확장과정 중 CF (complement form) 형의 민텀이 포함될 때 PI가 결정되며 모든 FPD방향의 민텀에 대한 탐색이 끝나면 정점을 포함하는 PI의 집단이 만들어진다.

제 2 단계 : 제 1 단계에서 구해진 {PI}는 여러 종류의 헤드를 가질 수 있으며 최적의 제 2 단계이트 구성에 쓰여 질 항을 구하기 위해 같은 헤드를 가진 PI들을 묶어 LPPI를 만들어 낸다. 묶는 방법은 정리 1을 이용하며 결과항은 항상 PE형이 되어야 한다. 따라서 남은 LPPI들은 모두 다른 헤드를 가진 PE형의 항들이다.

제 3 단계 : 앞 단계에서는 정점에 대한 {LPPI}를 만들었으며 이 중에서 최적 TANT 회로망에 쓰여 질 어느 하나를 선택하여야 한다. 정점에 대하여 LPPI가 하나뿐 일 때 이것은 CPPI가 된다. 만약 여러개

의 LPPI가 존재한다면 그 중에서 가장 적합한 LPPI를 선택하여야 한다. 선택된 LPPI를 EPPI로 정의하였다.

이 단계에서 구해진 CPPI나 EPPI에 포함된 모든 UF형의 민텀들은 XF형으로 바뀐 후 다시 제 1 단계로 넘어간다. 이런 과정은 원 함수의 UF형 민텀이 모두 처리 될 때까지 계속한다.

제 4 단계 : 정리 4에 의하여 제 1, 2, 3 단계를 통하여 만들어진 {CPPI}와 {EPPI}로 원 함수를 구성한다. 또한 이항들의 테일들을 재 조정하여 제 3 단 구성게이트의 개수를 줄일 수 있다.

4. 최적 TANT 회로망의 적용예

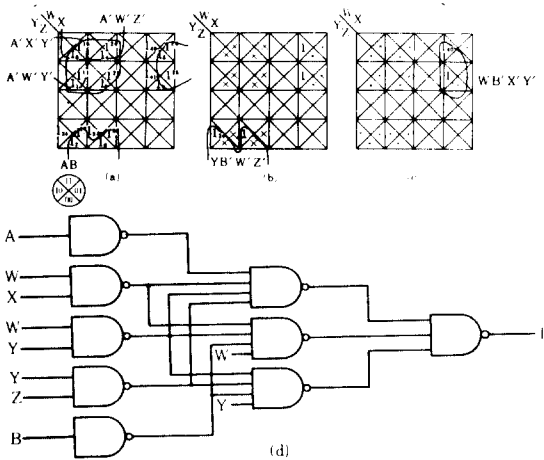
4-1. 예 제

어떤 스위칭함수 F가 아래식으로 구성된다.

$$F(A, B, W, X, Y, Z) = \sum m(0, 1, 2, 4, 5, 6, 8, 9, 16, 17, 18, 20, 21, 22, 24, 25, 34, 38, 40, 41)$$

그림 4(a)에서 초기 정점은 m_0 각 되며 PI들은 {A'W'Y', A'X'Y', A'W'Z'}이다. PI를 전부 더하면 LPPI인 A'(WX)'(WY)'(YZ)'을 얻을 수 있으며 이것이 CPPI로 된다. 여기에 포함되는 민텀들은 모두 XF형으로 변환되어 다음 과정으로 넘어간다. 다음 정점은 그림 4(b)에서와 같이 m_{34} 이며 PI는 YB'W'Z'이며 이것이 바로 CPPI이다. 마찬가지로 m_{40} 에 대한 PI는 WB'X'Y'이며 동시에 CPPI이다. 이상에 의해 UF형의 모든 민텀이 포함됐기 때문에 다음과 같은 TANT 식이 구성된다.

$$F = A'(WX)'(WY)'(YZ)' + YB'(WY)'(YZ)' + WB'X'Y'$$



EXAMPLE 4-1

A OPTIMAL TANT NETWORK WILL BE FOUND FOR THE FUNCTION
 $F(A, B, W, X, Y, Z) = \sum m(0, 1, 2, 4, 5, 6, 8, 9, 16, 17, 18, 20, 21, 22, 24, 25, 34, 38, 40, 41)$

ONLY TAIL

THE FUNCTION AND TAIL REALIZED BY THE SECOND-LEVEL GATES ARE

CPPI (OR EPPI) A I S
 (A)'(YZ)'(WY)'(WX)'
 Y(Z)'(WY)'(YZ)'
 W(X)'(WY)'(YZ)'
 W(X)'(WY)'(YZ)'

**** SOLUTION ****
 $F = A'(WX)'(WY)'(YZ)' + YB'(WY)'(YZ)' + WB'X'Y'$

NUMBER OF GATE INPUTS 23
 NUMBER OF GATES 9

ALL DONE TOTAL EXEC. TIME 5283 MILLISECOND

(e)

그림 4. 회로망 예제에 대한 설계과정
 Fig.4. Design steps for example .

및 식에서 최적테일항들을 선정하면 다음과 같은 최적 TANT 식이 된다.

$$F = A'(WX)'(WY)'(ZY)' + YB'(WY)'(YZ)' + WB'(WX)'(WY)'$$

이에 대한 회로망은 그림 4(d)와 같으며 그림 4(e)는 본 연구에 의한 CAD 패키지 적용 결과를 나타낸다.

4.2. 결과 및 고찰

최적 TANT 회로망 설계에서 입력변수의 개수가 2개까지는 쉽게 수작업으로 이루어질 수 있으며, 컴퓨터에 의한 수행은 기억용량의 제한때문에 3개부터 6개까지로 하였다. 함수의 입력변수가 7개일 때 함수의 구성가능한 수식의 조합은 2^{128} 가지이며 각 수식에 대한 실행속도의 비교는 방대한 것이다. 본 실험은 몇 가지의 데이터를 선정하여 실행하였으며, 설계상의 모든 조건을 검사할 수 있는 4개의 변수로 구성된 함수를 주로 다루었고, 표 1에서 볼 수 있는 바와 같이 몇개의 예에 대하여 Layton에 의한 CAD 패키지의 적용 결과와 비교하였다. 표 1에서 보는 바와 같이 구성 게이트 수 및 총 게이트 입력수는 같았으며 실행시간은 3배 이상으로 빨라짐을 볼 수 있었다. 이것은 설계알고리즘의 차이로 생겨나며 Layton에 의한 게이트 후보들의 확정없이 직접 최적게이트집단을 선정하기 때문이다.

실행시간은 입력정보를 읽는 시간, 배열들을 발생

시키는 시간 및 2진명령어코드문을 저장하는 시간을 포함하기 때문에 정확한 실행시간은 총 시간보다 적게 든다. 또한 이 실행 시간은 원 함수의 복잡성에 비례하여 증가되며, 체계적인 수학적 연구가 이루어지지 않았지만 구성 변수의 개수와 UF형 민텀의 밀도가 실행시간을 결정하는 2개의 중요한 요소가 됨을 알 수 있다.

5. 결론

본 연구는 진값의 입력만을 가진 AND-NOT 게이트로 구성된 3단논리회로인 TANT 회로망에 대한 최적설계알고리즘을 게이트가격의 제한아래서 제시하였다.

먼저 TANT 회로의 일반식을 서술하였으며 회로의 최적화를 위해서 우선 제 2 단계이트집합을 구하였다. 필요한 제 2 단계이트의 탐색과정은 작은 첨자값을 가진 민텀에서 부터 시작하여 모든 UF형의 민텀이 처리 될 때까지 계속되며, 정점을 중심으로 하여 {PI}가 구해진 후 정점을 포함하는 최대 PE식이 되는 C PPI 및 EPPI가 선정된다.

본 연구는 타 논문과 비교하여 최적화에 필요한 알고리즘이 간략화되었으며 이로 인해 컴퓨터의 처리시간 및 기억장소가 절약될 수 있었다. 아래에 대표적인 Gimpel 및 Layton의 이론과 비교하여 표를 만들었다.

표 1. 패키지 비교표
Table I. Comparison table of CAD packages.

항목	예제 구분	EX 4-1		EX 4-2		EX 4-3		EX 4-4	
		AHN	LAYTON	AHN	LAYTON	AHN	LAYTON	AHN	LAYTON
게이트 수		6	6	9	9	7	7	6	6
입력 수		16	16	23	23	21	21	15	15
실행 시간(sec)		0.789	2.701	1.277	9.317	0.726	2.606	0.658	2.189

EX 4-1: $F(W, X, Y, Z) = \sum m(4, 5, 6, 9, 12, 13, 15)$

EX 4-2: $F(A, B, W, X, Y, Z) = \sum m(0, 1, 2, 4, 5, 6, 8, 9, 16, 17, 18, 20, 21, 22, 24, 25, 34, 38, 40, 41)$

EX 4-3: $F(W, X, Y, Z) = \sum m(3, 5, 11, 12, 14, 15)$

EX 4-4: $F(W, X, Y, Z) = \sum m(0, 1, 2, 3, 4, 5, 6, 11, 13)$

실행시간은 HP 3000 컴퓨터 시스템에 의한 것이다.

표 2. 알고리즘 비교표
Table II. Comparison table of algorithm.

구분	본 연구	GIMPEL (4)	LAYTON (9)
중간 과정	UF 항의 민텀을 포함하는 {PI}를 구한 후 {LPPI}를 만든다	회로망에 포함되는 PPI로부터 모든 major PPI를 만든다	기점을 중심한 모든 헤드 및 테일 후보들의 집합을 구한다
최종 과정	{LPPI}로부터 CPPI 및 EPPI를 구하여 직접 최적항을 만든다	major PPI 중에서 C-C 표를 이용하여 최적항을 선정한다	C-C 표와 유사한 방법으로 헤드와 테일 후보들의 집합에서 최적항을 선정한다.

본 연구에서 개발된 알고리즘은 최적의 제 2 단구성 게이트를 확정과정없이 직접 선정하게 되며 이 항들에 대한 유용한 테일집합으로부터 포함표를 이용하여 최적의 테일들로 구성된 최적 TANT 회로망 설계를 마치게 된다.

참 고 문 헌

1. F. J. Hill, G. R. Peterson, *Introduction to switching function theory and logic design*, New York : Wiley International Edition, 1968.
2. Z. Kohavi, *Switching and finite automata theory*, New York : McGraw-Hill, 1970.
3. D. Lewin, *Computer aided design of digital system*, New York : Crane, Russak & Company, 1977.
4. J. F. Gimpel, "The minimization of TANT network," IEEE Trans. on Electronic Computers, Vol. EC-16, pp. 18-38, February, 1967.
5. H. P. S. Lee, "An algorithm for minimal TANT network generation," IEEE Trans. on Computers, Vol. C-27, pp. 1202-1206, December 1978.
6. K. K. Chakrabarti, A. K. Choudhury, and M. S. Basu, "Complementary function approach to the synthesis of three level NAND network," IEEE Trans. on Computers, Vol. C-19, pp. 509-514, June 1970.
7. K. S. Koh, "A minimization technique for TANT networks," IEEE Trans. on Computers, Vol. C-20, pp. 105-107, January, 1971.
8. J. Frackowiak, "the synthesis of minimal hazardless TANT network." IEEE Trans. on Computers, Vol. C-21, pp. 1099-1108, October 1972.
9. J. E. Layton, J. R. Rowland, J. M. Acken, "Synthesis of optimal TANT network," Computer and Electrical Engineering, Vol. 1, pp. 123-134, 1978.
10. M. M. Mano, *Computer logic Design*. New Jersey : Prentice-Hall, 1972.
11. S. U. Robinson, R. W. House, "Gimpel's reduction technique extended to the covering problem with cost," IEEE Trans. on Computers, Vol. EC-16, pp. 509-514, August, 1967.
12. R. W. House, D. W. Steven, "A new rule for reducing CC tables. IEEE Trans. on Computers, Vol. C-19, pp. 1108-1111, November 1970.
13. H. A. Curtis, "The further reduction of C-C tables," IEEE Trans. on Computers, Vol. C-20, pp. 454-456, April, 1971.
14. H. A. Vink, "Reduction of table using multiple using multiple implication, IEEE Trans. on Computers, Vol. C-27, pp. 961-966, October, 1978.
15. V. T. Rhyne, "A new technique for the fast minimization of switching functions," IEEE Trans. on Computers, Vol. C-26, August 1977.
16. J. R. Slagle, C. L. Chang, R. C. T. Lee, "A new algorithm for generating prime implicants," IEEE Trans. on Computers, Vol. C-19, pp. 304-310, April, 1970.

