

論 文

CCD를 이용한 多值論理回路의 설계에 관한 Tabular法

正會員 宋 洪 復* 正會員 鄭 萬 永**

Tabular Methods for the Design of Multivalued Logic Circuits Using CCD

Hong Bok SONG*, Man Yung CHUNG** *Regular Members*

要 約 본 논문에서는 Tabular法을 이용한 CCD (charge-coupled device) 4值論理回路를 설계하는 방법을 제시하였다. 첫번째 4值 논리함수를 수산(手算) 및 컴퓨터 프로그래밍에 의해서 분해하고 이것을 기초로하여 Tabular法에 의한 CCD 4值회로를 실현시키는 알고리즘을 유도하였다. 이 알고리즘에 의해서 2變數 4值 논리함수를 분해(分解)해서 4개의 기본게이트에 의해 CCD 회로를 실현시켰다. 본논문의 방법에 의하면 기존방법에 비해 동일한 함수를 실현시키는데 소자수(素子數)와 코스트가 상당히 감소됨이 밝혀졌다.

ABSTRACT This paper offers a method to design CCD four-valued circuits using the tabular method. First, the four-valued logic function is decomposed by hand-calculation or computer program. Next, the algorithm is derived from the tabular method based on the decomposition process to realize the CCD four-valued circuit. According to this algorithm, the two-variable four valued logic function is decomposed and realized by CCD network with four basic gates. The synthesis method in this paper proves that the number of devices and cost is considerably reduced as compared with the existing methods to realize the same logic functions.

I. 서 론

CCD (Charge-Coupled Device)는 1970년 Boyle과 Smith가 처음 고안한 이래 단일 칩으

로 대량의 정보 처리를 할 수 있으며 집적도가 매우 높고 회로가 간단하여 여러 분야에 대한 응용이 활발하게 연구 되어 왔다^{(1), (2), (3)}. Zimmerman, Allen 및 Jacobs⁽⁴⁾는 C Mos, N Mos, I²L, T²L을 2치 논리 함수로 실현하는데 있어서 전력 소모, 지연 시간적(積) 및 칩 면적에 관한 비교를 하고 있다. CCD는 2치 논리연산에 적용되고 있지만 다치(多值) 논리연산 특히 4치 논리연산 회로에 관한 연구가 활발이 이루어지고 있다^{(5), (6)} 다치 논리회로는 2진 회로

* 東義工業專門大學 電子通信科
Dep. of Electronic Communication Engineering,
Dong Eui Technical Junior College
** 東亞大學校 電子工學科
Dep. of Electronic Engineering, Dong-A University
論文番號 : 88-41(接受 1988. 7. 29)

에 비하여 동일 정보량을 처리하는데 상호 연결이 감소되고, 단위 면적당 높은 함수 기능을 갖는 장점이 있다⁽⁷⁾. 최근에 Kerkhoff 및 Tervoert는 4치 논리 함수를 몇 개의 기본 게이트만으로 합성할 수 있음을 밝히고 있다. 이들은 주어진 4치 논리 함수를 몇 개의 부분(副分) 함수로 분해하고 다시 이 부분 함수를 4개의 기본 연산 회로로 실현 시키고 있다. Kerkhoff 및 Tervoert⁽⁸⁾ 등이 제시한 알고리즘은 분해가 한정되어 있고 부분(副分) 함수가 많이 생성되는 결점이 있으므로 본 논문에서는 부분 함수 생성을 줄일 수 있는 방법을 제시하였다. 제 II 장에서는 CCD의 기본 회로를 설명하였으며, 제 III 장에서는 Tabular 법에 의한 분해 알고리즘을 이용하여 승산 회로를 실현하였다.

II. CCD의 기본 회로

CCD는 전하량의 주입에 필요한 입력 다이오드와 입력 게이트가 필요하며 전하이동을 위한 전하이동 게이트와 에너지 우물에 저장된 전하량의 검출을 위한 floating 게이트가 필요하다. 임의의 4치 논리 함수는 4개의 기본 게이트에 의하여 CCD 회로 실현이 가능하며 이 회로 실현을 위한 CCD 기본 게이트는 다음과 같다^{(8), (9)}.

(a) 가산 회로 (Addition)

CCD 게이트 중에서 가장 기본적인 게이트이며 그 구성은 그림 1과 같고 그림 1 (a)는 가산 회로의 기본 구조이고 (b)는 가산 회로의 기호이고 (c)는 가산 회로의 수학적 표현식이다.

이 게이트의 작용은 두개의 입력 합을 만드는 것이며 서로 다른 우물 (1)(2)에 저장되어 있는 전하량 Q_1, Q_2 를 우물 (3)으로 이동시켜 더하는 것이다. 에너지 우물에 저장할 수 있는 전하량은 다음과 같다.

$$\text{여기서 } Q_{\max} = C_{ox} (V_1 - V_2) \cdot A \quad (1)$$

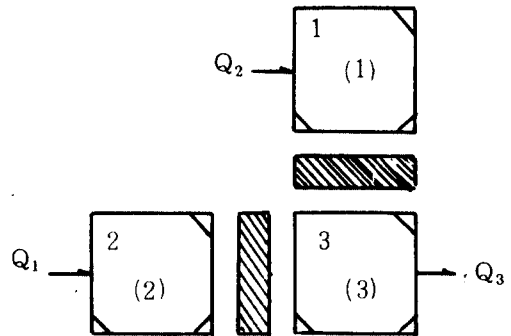
Q_{\max} : 에너지 우물에 저장할 수 있는 최대 전하량

C_{ox} : SiO_2 층에 의한 게이트의 단위체적당 용량

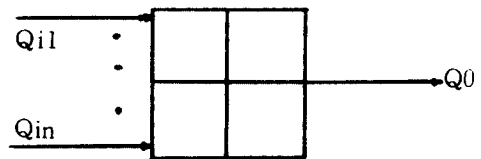
V_1, V_2 : 서로 이웃한 게이트 전압

A : 축적 게이트

여기서 Q_{\max} 를 변화시키는데 A를 변화시킬 수도 있지만 V_1, V_2 중의 어느 하나를 변화시켜 하는 것이 보통이다.



(a)



(b)

$$Q_0 = \sum_{m=1}^n Q_{t,m}$$

$$n = 2, 3, \dots$$

(c)

그림 1 (a) 가산 회로의 기본 구조

(b) 가산 회로의 기호

(c) 수학적 표현식

(a) The basic structure of addition gate circuit

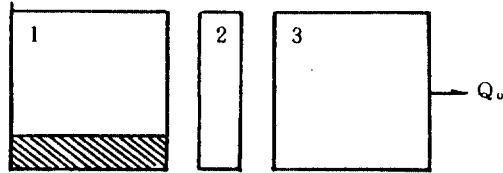
(b) The symbol of addition gate circuit

(c) The expression of mathematics

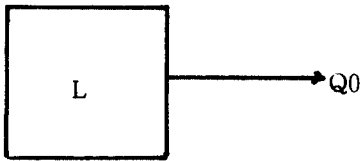
(b) 상수 회로 (Constant)

그림 2는 상수 게이트 구조를 나타낸 것이며 (a) 상수 회로의 기본 구조 (b) 상수 회로의 기호 (c) 수학적 표현식이다. 입력 전하량의 전체량이 우물의 전체량을 초과할 때는 출력측에서 전하의

발생이 생기며 입력 전하량에 따라 출력에 전하량 1, 2, 3의 출력량을 얻게 된다. 공정과 바이어스 변수 이외도 게이트 구성의 입력과 출력에 연관되어 상수는 가산 및 금지 회로와 같이 사용한다. 간혹 분해에서 출력이 사용되지 않는 경우는 회로에서 출력전하를 소모하는 것을 의미한다.



(a)



(b)

$$Q_0 = L$$

$$L = 1, 2, \dots$$

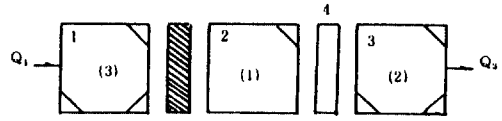
(c)

그림 2 (a) 상수회로의 기본 구조
(b) 상수회로의 기호
(c) 수학적 표현식
(a) The basic structure of constant gate circuit
(b) The symbol of constant gate circuit
(c) The expression of mathematics

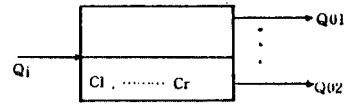
(c) Fixed overflow 회로

그림 3은 Fixed overflow 회로의 구조를 나타낸 것이며 이 회로의 동작은 (a) Fixed overflow의 기본구조 (b) Fixed overflow 기호 (c) 수학적 표현식이다.

우물 2에 전하량이 전달되고 이 우물의 용량이 주어진 전하량보다 작다고 하면 전하량은 인접된 우물 3으로 넘어 흐르게 된다. 여기서 전하처리 용량은 장벽 4의 전압 V_2 와 우물 2의 면적에 의해서 결정된다.



(a)



(b)

$$Q_{0m} = C_m \cdot \left(\sum_{k=1}^m C_k \right)^{Q_1} + \left(Q_1 - \sum_{k=1}^{m-1} C_k \right) \cdot Q_1^{\sum_{k=1}^{m-1} C_k} \cdot \sum_{k=1}^m C_k^l$$

$$m = 1, 2, \dots, r \quad r = 1, 2, \dots \quad C_k = 0, 1, \dots$$

(c)

그림 3 (a) fixed overflow 회로의 기본 구조
(b) Fixed overflow 회로의 기호
(c) 수학적 표현식

(a) The basic structure of fixed overflow gate circuit
(b) The symbol of fixed overflow gate circuit
(c) The expression of mathematics.

(d) 금지회로(Inhibit)

그림 4는 금지회로의 구성도이며 (a) 금지회로의 기본구조 (b) 금지회로의 기호 (c) 수학적 표현식이다.

입력 전하량 Q_{im} 의 제어 전하량 Q_c 에 의해서 제어되며 $Q_c > 0$ 일 때에는 Q_{sm} 에는 각기 Q_{im} 의 전하량이 전달되고 Q_{dm} 은 모두 0이 되지만 $Q_c < 0$ 일 때에는 Q_{sm} 을 모두 0이 되고 Q_{dm} 은 각기 Q_{im} 과 같아야 한다.

또한 본 논문에서 사용되는 수학적 기호법은 다음과 같다.

$$Q^{\leq \alpha} = 1 \quad \text{iff} \quad Q \leq \alpha \quad (2)$$

$$0 \quad \text{otherwise}$$

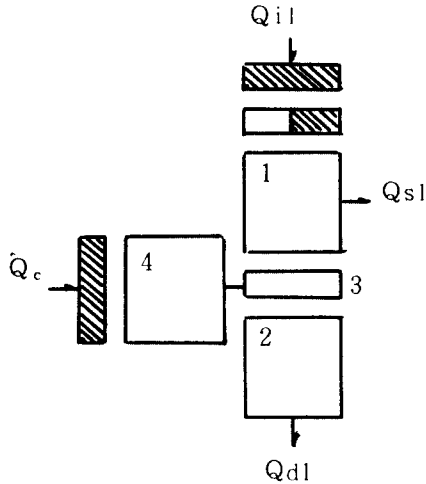
$$Q^{> \alpha} = 1 \quad \text{iff} \quad Q > \alpha$$

$$0 \quad \text{otherwise}$$

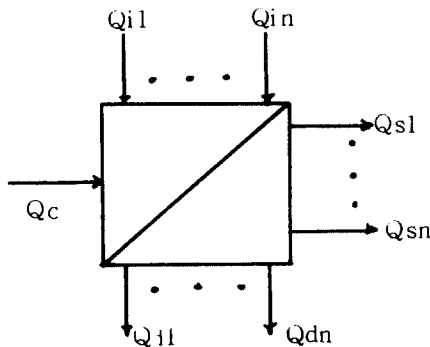
$$Q^{(\alpha, \beta)} = 1 \quad \text{iff} \quad \alpha \leq Q < \beta$$

$$0 \quad \text{otherwise}$$

임을 뜻한다.



(a)



(b)

$$Q_{sm} = Q_{im} \cdot Q_c^{01}$$

$$Q_{am} = Q_{im} \cdot Q_c^{01}$$

$$m = 1, 2, \dots, n \quad n = 1, 2, \dots$$

(c)

그림 4 (a) 금지회로의 기본구조
 (b) 금지회로의 기호
 (c) 수학적 표현식
 (a) The basic structure of Inhibit gate Circuit
 (b) The symbol of Inhibit gate circuit
 (c) The expression of mathematics

III. Tabular법을 이용한 4치 논리함수의 분해

일반적으로 CCD 4치논리함수는 이것을 회로로 실현시키는데 각종 게이트를 다 사용해서 할 수도 있지만 코스트를 고려할때는 칩 면적을 가장 작게 하는 것이 요구된다. 그러기 위해서는 앞에서 설명한 가산회로, 상수회로, Fixed overflow 회로 및 금지회로 4개만으로 구성시키는 것이 가장 적합하다⁽⁸⁾.

그러나 진리표로 주어지는 임의의 4치논리함수는 표 1의 24개의 함수를 제외하고는 직접 이들 4개의 게이트로 실현시킬수는 없으며, 그러기 위해서는 분해 절차가 필요하게 된다.

III-1. 진리표의 분해

본절에서는 주어진 진리치 표에 대하여 분해하는 방법을 논하며, 본 논문에서 사용하는 함수들에 대하여 다음과 같이 정의한다.

[정의 1]⁽⁹⁾

단일변수 4치논리함수의 입력 $\langle 0, 1, 2, 3 \rangle$ 에 대한 출력은 $\langle X_0, X_1, X_2, X_3 \rangle$ 으로 표시한다.

[정의 2]⁽⁹⁾

단일변수 4치논리함수 $\langle X_0, X_1, X_2, X_3 \rangle$ 에 있어서

- a) $X_0 \leq X_1 \leq X_2 \leq X_3$ 일때는 증가함수
- b) $X_0 \geq X_1 \geq X_2 \geq X_3$ 일때는 감소함수
- c) 그 외의 경우에는 혼합함수라고 한다.

예를 들면 $\langle 0, 0, 2, 3 \rangle$, $\langle 3, 2, 0, 0 \rangle$ 및 $\langle 0, 2, 3, 0 \rangle$ 을 각각 3-증가함수, 3-감소함수, 3-혼합함수이다. 여기서 3은 출력치 중에 최대치가 3임을 뜻한다.

Kerkhoff와 Robroek⁽⁸⁾는 단일변수 4치논리함수를 CCD로 실현시키는데 있어서 코스트를 고려한 몇가지 정형을 설정하여 이 정형을 이용하는 것이 효과적이라는 것을 제시하였다.

표 1은 Kerkhoff와 Robroek⁽⁸⁾가 제시한 정형의 코스트와 이를 실현시키는데 필요한 회로를 표시한 것이다.

여기서

$C(L)$: 상수회로에서 출력량 L 을 발생하는 기호

$A(Q_1, Q_2, Q_3)$: 가산회로에서 입력 Q_1, Q_2, Q_3 를 더하는 기호

$I(Q_1, Q_2, Q_3)$: 금지회로에서 입력량 Q_i 를 제어량 Q_c 에 의해서 출력량 Q_o 를 발생하는 기호.

$F(Q_x, C_1, C_2, C_3, Q_1, Q_2, Q_3)$: Fixed overflow 회로에서 입력량 Q_x 에 대한 용량 C_1, C_2, C_3 인 우물의 출력량이 각각 Q_1, Q_2, Q_3 를 발생하는 기호이다.

그러므로 주어진 함수 $f(x)$ 를 효과적으로 분해하여 최종 분해가 표 1에 포함될때까지 분해를 계속하는데 그림 5는 그 과정을 표시한 것이다. 여기서 $R_i(x)$ 는 부분함수이고 $S_i(x)$ 는 최종 함수로서 표 1에 포함됨을 뜻한다.

따라서 총 코스트를 계산하는 식은 다음과 같다.

$$C\{f(x)\} = \sum_{j=1}^v C\{S(j)\} + (V-1) \cdot C\{A\} \quad (3)$$

여기서 $C\{S(j)\}$: 최종함수의 코스트

$C\{A\}$: 가산회로의 코스트

V : 최종함수의 수

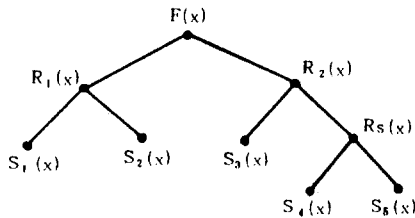
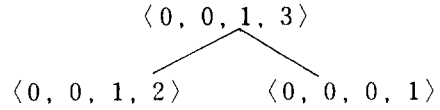


그림 5 함수분해의 예
Decomposition of a function

앞에서 설명한 분해 절차를 증가함수, 감소함수 및 혼합함수의 경우를 예로 들어 설명한다.

예) 증가함수 $\langle 0, 0, 1, 3 \rangle$ 을 분해하면



으로 분해되며 회로 실현을 그림 6과 같다. 이때 총 코스트는 식(3)에 의하여 10이 된다.

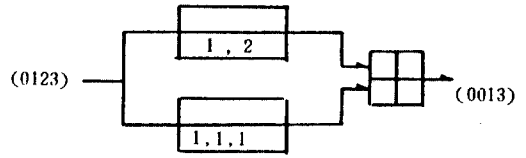
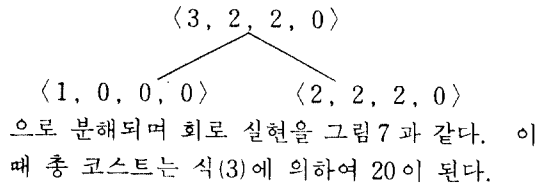


그림 6 증가함수 회로 실현
Circuit realization of increasing function

예) 감소함수 $\langle 3, 2, 2, 0 \rangle$ 을 분해하면



으로 분해되며 회로 실현을 그림 7과 같다. 이때 총 코스트는 식(3)에 의하여 20이 된다.

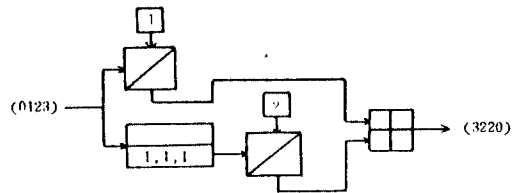
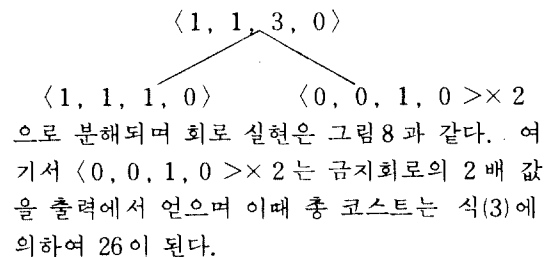


그림 7 감소함수의 회로 실현
Circuit realization of decreasing function

예 3) 혼합함수 $\langle 1, 1, 3, 0 \rangle$ 을 분해하면



으로 분해되며 회로 실현은 그림 8과 같다. 여기서 $\langle 0, 0, 1, 0 \rangle \times 2$ 는 금지회로의 2배 값을 출력에서 얻으며 이때 총 코스트는 식(3)에 의하여 26이 된다.

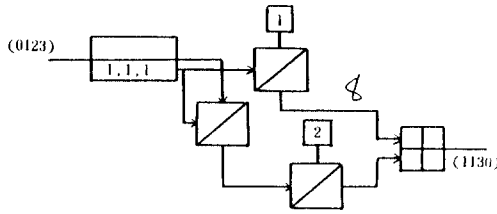


그림 8 혼합함수 회로 실현
Circuit realization of mixed function

위의 예에서 보는바와 같이 주어진 함수를 효과적으로 표 1에 의한 정형으로 분해해야 하는 데 상수함수 $\langle 1, 1, 1, 1 \rangle$, $\langle 2, 2, 2, 2 \rangle$ 및

$\langle 3, 3, 3, 3 \rangle$ 을 표 1에 있으므로 간단히 합성되지만 상수함수가 아닌 함수를 분해하는 방법은 다음과 같다.

1) 증가함수 $f(x)$ 는 부분함수인 $R_1(x)$ 와 $R_2(x)$ 로 분해하며 그 중 하나는 표 1에 포함되는 함수이거나 또는 표 1에 포함되는 함수의 각 항을 정수배한 것이 되게 한다.

2) 최소함수 $f(x)$ 는 $\langle 2, 1, 0, 0 \rangle$ $\langle 2, 1, 1, 0 \rangle$ 및 $\langle 3, 2, 1, 0 \rangle$ 인 함수를 제외하고는 (1)의 방법을 취한다. 이때 $\langle 2, 1, 0, 0 \rangle$ $\langle 2, 1, 1, 0 \rangle$ $\langle 3, 2, 1, 0 \rangle$ 인 함수는 $\langle 2, 0, 0, 0 \rangle$ 이 되게 분해하여야 한다.

표 1 4차회로 합성에 사용되는 코스트 표
Cost table used in the Synthesis Techniques.

No.	Function	Cost	Realization
1	1111	1	C(1)
2	2222	1	C(2)
3	3333	1	C(3)
4	1233	3	A(1, Q _x)
5	2333	3	A(2, Q _x)
6	0001	4	F(Q _x , 1, 1, 1, Q _s)
7	0011	4	F(Q _x , 1, 1, Q ₂)
8	0111	4	F(Q _x , 1, Q ₁)
9	0012	4	F(Q _x , 1, 2, Q ₂)
10	0122	4	F(Q _x , 2, Q ₁)
11	0112	6	F(Q _x , 1, 1, 1, Q ₁ , Q ₃); A(Q ₁ , Q ₃)
12	0133	6	F(Q _x , 1, 2, Q ₂); A(Q ₂ , Q _x)
13	0233	6	F(Q _x , 1, Q ₁); A(Q ₁ , Q _x)
14	1000	7	I(1, Q _x , Q _d)
15	2000	7	I(2, Q _x , Q _d)
16	0222	7	I(2, Q _x , Q _s)
17	0010	10	F(Q _x , 1, 1, 1, Q ₂ , Q ₃); I(Q ₂ , Q ₃ , Q _d)
18	0100	10	F(Q _x , 1, 1, Q ₁ , Q ₂); I(Q ₁ , Q ₂ , Q _d)
19	0110	10	F(Q _x , 1, 1, 1, Q ₁ , Q ₃); I(Q ₁ , Q ₃ , Q _d)
20	1120	10	F(Q _x , 2, 1, Q ₁ , Q ₂); I(Q ₁ , Q ₂ , Q _d)
21	1100	11	F(Q _x , 1, 1, Q ₂); I(1, Q ₂ , Q _d)
22	1110	11	F(Q _x , 1, 1, 1, Q ₃); I(1, Q ₃ , Q _d)
23	2200	11	F(Q _x , 1, 1, Q ₂); I(2, Q ₂ , Q _d)
24	2220	11	F(Q _x , 1, 1, 1, Q ₃); I(2, Q ₃ , Q _d)

3) 혼합함수의 경우에는 0 항의 위치에 따라 분해방법이 달라진다. 일반적으로 증가함수는 감소함수보다 코스트가 낮고 감소함수의 코스트는 혼합함수의 코스트보다 낮다.

그 이유는 증가함수는 Fixed overflow 회로만으로 실현이 가능하나 감소함수의 경우는 이에 더하여 가산회로 및 금지회로가 더 필요하다. 혼합함수에 있어서는 부분함수가 생성될 때마다 금지회로가 필요하게 되므로 코스트가 높아지는 것이다.

III - 2. Tabular 법에 의한 분해

본 절에서는 단일변수 4 치논리함수를 분해하는데 있어서 Kerkhoff 등이 제시한 알고리즘은 분해가 한정되어 있고 부분함수가 많이 생성되는 결점이 있으므로 본 논문에서는 부분함수 생성을 줄일 수 있는 방법을 제시한다.

이에 앞서 회로 실현에 있어서 증가함수는 fixed overflow 회로만으로 실현이 가능하고 감소함수는 이에 더하여 가산회로와 금지회로가 필요하다. 또한 혼합함수는 부분함수가 생성시마다 금지회로가 필요하게 되므로 함수를 분해하는데 있어서 코스트가 낮게 분해하여야 한다. 그러므로 적은 코스트를 알 수 있는 변환수 (Transition Count : TC) 를 다음과 같이 정의한다.

[정의 3] ⁽⁹⁾

변환수 (TC)라 함은 단일변수 출력에 있어서 증가에서 감소하고 다시 증가하는 회로수를 말한다. 여기서 처음 증가하는 것에 대해서는 계산하지 않는다. 예로서 $\langle 0, 0, 2, 3 \rangle$, $\langle 3, 2, 0, 0 \rangle$, $\langle 0, 2, 3, 0 \rangle$, $\langle 0, 2, 0, 3 \rangle$, $\langle 2, 0, 3, 0 \rangle$ 의 TC는 각각 0, 1, 1, 2, 3이다.

[Tabular 법에 의한 분해 알고리즘]

1) 부분함수 $R_1(x)$ 및 $R_2(x)$ 로 분해할때 이들의 TC는 각각 原함수 $f(x)$ 의 TC보다 작거나 최소한 같도록 분해한다.

2) 原함수 $f(x)$ 에 원소치가 3을 포함할때는 $R_1(x)$ 및 $R_2(x)$ 의 TC는 2 또는 1이 되거나 작도록 분해한다.

3) 부분함수 $R_1(x)$ 및 $R_2(x)$ 는 각 原함수의 총 수가 최소가 되도록 분해한다.

4) 단일변수 $\langle X_0, X_1, X_2, X_3 \rangle$ 에 있어서 $X_0 = 0, X_1 \leq X_2, X_3 = 0, X_1, X_2 \geq 2$ 일때는 $\langle 0, 1, 1, 0 \rangle$ 의 2배와 $\langle 0, X_1 - 2, X_2 - 2, 0 \rangle$ 로 분해한다.

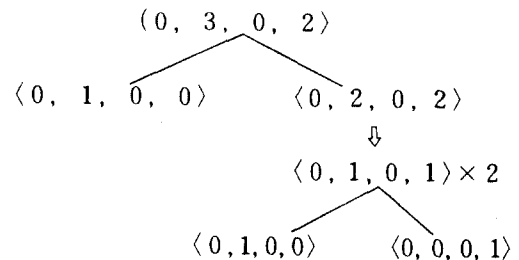
5) $X_0 = X_2 = 0, X_1, X_3 \geq 2$ 일 때는 $\langle 0, 1, 0, 1 \rangle$ 의 2배와 $\langle 0, X_1 - 2, 0, X_3 - 2 \rangle$ 로 분해한다.

6) $X_0, X_2 \geq 2, X_1 = 1, X_3 = 0$ 이거나 $X_1 = 0, X_3 = 1$ 일때는 $\langle 1, 0, 1, 0 \rangle$ 의 2배와 $\langle X_0 - 2, 1, X_2 - 2, 0 \rangle$ 또는 $\langle X_0 - 2, 0, X_2 - 2, 1 \rangle$ 로 분해한다.

7) $X_0 \geq 2$ 이고 X_1, X_2, X_3 중 원소치 2를 취하는 것이 1 또는 2 개이고 나머지 변수 중 원소치 0 일때는 0을 취하는 자리를 제외한 나머지 변수의 원소치는 모두 1인 원소치의 negation의 2배와 다른 함수로 분해한다.

본 논문의 알고리즘을 사용하여 예를 들면 다음과 같다.

예 1)



으로 분해되며 여기서 $\langle 0, 1, 0, 1 \rangle \times 2$ 는 금지회로에서 2배의 값을 출력 Q_3 에서 얻으며 회로 실현은 그림 9와 같으며 이 함수의 코스트는 식(3)에 의하여 21이다.

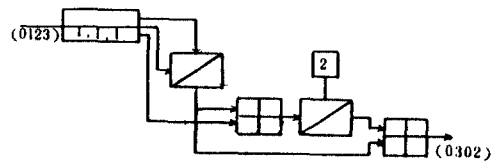
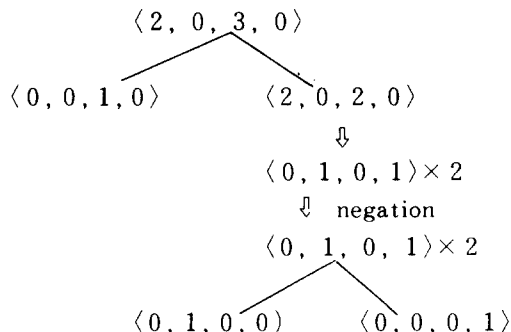


그림 9 회로 실현 예

An example of a circuit realization

예 2)



으로 분해되며 여기서 $\langle 1, 0, 1, 0 \rangle \times 2$ 를 negation 취하면 $\langle 0, 1, 0, 1 \rangle \times 2$ 로 되며 $\langle 0, 1, 0, 1 \rangle \times 2$ 는 금지회로에서 2 배의 값을 출력 Qd에서 얻으며 회로실현은 그림10과 같으며 이 함수의 코스트는 식(3)에 의해서 28이다.

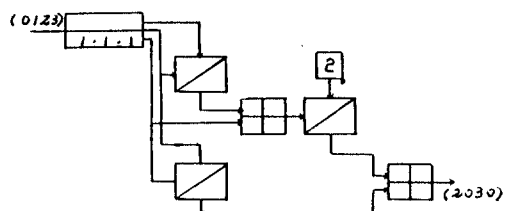


그림10 회로 실현 예
An example of circuit realization

본 논문에서 논한 Tabular법에 의한 분해 알고리즘을 이용하여 총 256개의 단일변수 4치 논리함수를 컴퓨터 시뮬레이션 하였으며 이 결과를 Kerkhoff 및 Robroek⁽⁸⁾의 처리 결과와 Lee⁽¹⁰⁾의 처리 결과와 비교하였다. 표2는 본 논문의 결과를 기존의 방법과 비교하여 코스트가 감소하는 함수들만을 나타내었다. 그리고 다른 함수에 대해서는 같은 코스트를 얻을 수 있었다. 이 표에 의하면 Kerkhoff 및 Robroek⁽⁸⁾과는 106개 함수에 대하여, Lee⁽¹⁰⁾경우와는 8개 함수에 대하여 코스트가 감소하였으며 평균 코스트는 Kerkhoff 및 Robroek⁽⁸⁾보다 2.79, Lee⁽¹⁰⁾보다는 0.45가 감소함을 알 수 있다.

IV. 적용예

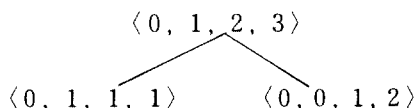
III-2에서는 단일변수 4치논리함수의 회로실현에 대하여 논하였으며 본장에서는 이 결과를 2변수 4치논리함수의 회로실현에 확장하는 절차에 대해서 논하였다.

예로서 그림11의 진리표(multiple 회로)로 주어지는 2변수 4치논리함수를 합성할 때 이것을 단일변수 4치의 합으로 본다. 여기서 X는 일정하고 Y가 변한다고 볼 때는 진리표를 縱으로 보기 때문에 縱축 처리를 하게 하는 것이고 Y가 일정하고 X가 변한다고 볼 때는 橫축 처리를 하게 된다. 縱축 및 橫축 중에서 총 코스트가 작은 쪽으로 제어 입력을 취하는데 이 진리표에서는 縱, 橫이 같으므로 橫축(y변수)를 제어 입력으로 취한다.

		X			
	Y	0	1	2	3
0	0	0	0	0	0
1	0	1	2	3	
2	0	2	0	2	
3	0	3	2	1	

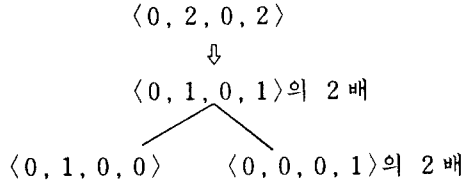
그림11 2변수 4치논리 승산회로의 진리표
Truth table of 2-variable function (multiplier)

1) Y = 1 일 때는 출력함수는 $\langle 0, 1, 2, 3 \rangle$ 이므로 이것을 4개의 기본 게이트로 실현시키기 위해서는 분해를 해야 하는데 표1에 의하여 직접 다음과 같이 분해된다.



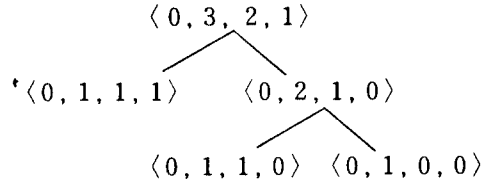
이 분해 결과 코스트는 식(3)에 의해 10이지만 본 논문에서 정의된 TC에 의해서 함수 $\langle 0, 1, 2, 3 \rangle$ 은 TC가 0이므로 코스트는 0이다.

2) $Y = 2$ 일때는 출력함수는 $\langle 0, 2, 0, 2 \rangle$ 이며 알고리즘을 적용하면



로 분해되며 표 1에 포함되는 함수로 분해할 수 있다. 이 분해의 결과는 금지 회로의 Q_5 출력의 2 배로 얻는다.

3) $Y = 3$ 일때의 출력함수는 $\langle 0, 3, 2, 1 \rangle$ 이며



으로 분해되며 TC 및 코스트는 부록에 나타내었다. 따라서 이 분해 결과를 합성하여 회로를 실현하면 그림 12와 같다. 그림 12의 승산회로를 Lee 및 Butler의 승산회로와 비교할 때 훨씬 코스트가 경감됨을 알 수 있다.

표 2

		Lee ⁽¹⁰⁾	Ker and Rob ⁽⁹⁾ 의 개선된 코스트	본논문의 코스트
0302	$F(Q_x, 1, 1, 1, Q_1, Q_2, Q_3); I(Q_1, Q_2, Qd_1); A_1(Qd_1, Q_3); I(2, A_1, Qs_2); A_2(Qs_2, Qd_1)$	27	31	21
0331	$F(Q_x, 1, 1, 1, Q_1, Q_3); I(Q_1, Q_3, Qd_1); I(2, Qd_1, Qs_2); A(Q_1, Qs_2)$	21	23	19
1030	$F(Q_x, 1, 1, 1, Q_2, Q_3); I(Q_2, Q_3, Qd_1); I(2, Qd_1, Qs_2); A_1(Qd_1, Qs_2); I(1, Q_x, Qd_3); A_2(A_1, Qd_3)$	28	38	28
2003	$F(Q_x, 1, 1, 1, Q_1, Q_3); I(Q_1, Q_3, Qd_1); I(2, Qd_1, Qd_2); A(Q_3, Qd_2)$	22	25	19
2030	$F(Q_x, 1, 1, 1, Q_1, Q_2, Q_3); I(Q_1, Q_2, Qd_1); A_1(Qd_1, Q_3); I(2, A_1, Qd_2); I(Q_2, Q_3, Qd_3); A(Qd_2, Qd_3)$	31	38	28
2031	$F(Q_x, 1, 1, 1, Q_1, Q_2, Q_3); I(Q_1, Q_2, Qd_1); A_1(Qd_1, Q_3); I(2, A_1, Qd_2); A_2(Qd_2, Q_2)$	25	32	21
2033	$F(Q_x, 1, 1, Q_1, Q_2); I(Q_1, Q_2, Qd_1); I(2, Qd_1, Qd_2); A(Qd_2, Q_2)$	22	25	19
3032	$F(Q_x, 1, 1, 1, Q_1, Q_2, Q_3); I(Q_1, Q_2, Qd_1); I(2, Qd_1, Qd_2); I(Q_2, Q_3, Qd_3); A_1(Qd_2, Qd_3); I(1, Q_x, Qd_4); A_2(A_1, Qd_4)$	36	40	34
3130	$F(Q_x, 1, 1, 1, Q_1, Q_2, Q_3); I(Q_1, Q_2, Qd_1); A_1(Qd_1, Q_3); I(2, A_1, Qd_2); I(1, Q_3, Qd_3); A_2(Qd_2, Qd_3)$	32	35	28
3303	$F(Q_x, 1, 1, 1, Q_2, Q_3); I(1, Q_2, Qd_1); A_1(Qd_1, Q_3); I(2, A_1, Qd_2); A_2(A_1, Qd_2)$	22	31	22

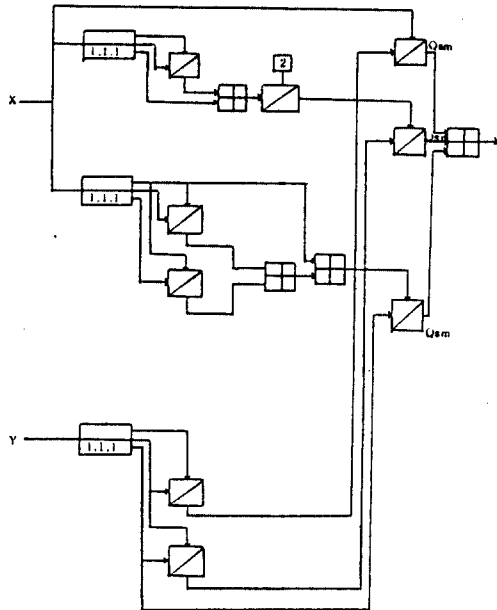


그림12 2 변수 4치 승산회로실현
Circuit realization of Two-variable Four-valued multiplier.

V. 결 론

본 논문에서는 CCD 회로를 사용한 4值 논리함수의 설계 방법을 제시하였다. Tabular 법에 의하여 회로를 실현하는 알고리즘을 구하고 이 알고리즘에 4值 논리함수를 手算 또는 컴퓨터 프로그램에 의해서 분해한 후 합성한 결과 K Kerkhoff, Robroek⁽⁸⁾와 Lee⁽¹⁰⁾에 의한 결과와 비교하였다. 그결과 총 256개의 4值 논리함수에서 Kerkhoff와 Robroek의 방법에 의한 것과 비교할때는 106개가, Lee에 의한결과와 비교할때는 8개의 함수에 대하여 코스트가 감소하였으며 종합적으로 총 평균 코스트는 Kerkhoff와 Robroek의 것보다 2.79, Lee의 것보다는 0.45의 감소를 보였다. 그러나 금지 회로의 입력에 원소치 3의 전하량이 들어올 때 코스트값을 줄이는 문제에 대해서는 아직 결론을 얻지 못하였다.

본논문에서는 잡음문제를 고려하지 않고 회로 소자수 및 코스트값을 줄여 실현하는 방법만을 논하였다.

參 考 文 獻

1. M.J. Howes and D.V. Morgan, "Charge-Coupled devices and system", Wiley, New York, 1979.
2. Y. Gamou, M. Yamada, K. Fujishima, T. Tada and S. Takano, "All TTL Compatible CCD memory with CCD clock, generator", IEEE J. Solid-state circuits, Vol.SC-15, Oct. 1980, pp.881-886.
3. L.M. Terman, Y.S. Yee, R.B. Merrill, L.G. Heller and M.R. Pettigrew, "CCD memory using multilevel storage", IEEE J. Solid-state circuits, Vol.SC-16, Oct. 1981, pp.472-485.
4. T.A. Zimmerman, R.A. Allen and R.W. Jacobs, "Digital Couple Logic(DCCL)", IEEE J. Solid-state circuits, Vol.SC-12, Oct. 1977, pp. 473-485.
5. H.G. Kerkhoff and M.L. Tervoert, "The implementation of multiple-valued function using charge-coupled devices", Proc. 10th International symposium on Multiple-valued logic, Evanston, IL, June 1980, pp.6-15.
6. H.G. Kerkhoff, M.L. Tervoert and Harrie A.C. Tilmans, "design consideration and measurement results of multiple-valued logic CCD'S", Proc. 11th International Symposium on Multiple-Valued Logic, Oklahoma, U.S.A., May 1981, pp.205-211.
7. T.T. Dao, E.J. McCluskey and L.K. ressell, "Multi-valued integrated injection logic", IEEE Trans. on comput., Vol.C-26, Dec. 1977, pp.1233-1241.
8. H.G. Kerkhoff, H.A.J. Robroek, "The Logic Design of Multiple-Valued Logic Function using Charge-Coupled Devices", Proc. 1982 Int. Symp. Multiple-Valued Logic, Paris, France, May 1982, pp.35-44.
9. J.K. Lee, J.T. Butler, "Tabular method for the design of CCD multiple-valued circuits", The thirteenth international symposium on Multiple-Valued Logic, May 1983, Kyoto, pp.162-170.
10. J.K. Lee, "Synthesis Techniques for Four-Valued Logic CCDS Circuits", M.S. Thesis, Northwestern University, Evanston, IL, Aug. 1982.
11. H.A.J. Robroek, "The synthesis of MVL-CCD circuits", M. Sc. report No.1217.3936, Twente University to technology, enschede, Dec. 1981.
12. T.T. Dao, E.J. McCluskey and L.K. Lassel, "Multi-Valued intergrated injection Logic". IEEE Trans. Computer, Vol.C-26, Dec. 1977, pp.1233-1241.



宋洪復(Hong Bok SONG) 正會員
1956年4月14日生
1983年2月：光云大學校 電子通信 工學
科(工學士)
1985年2月：仁荷大學校 電子工學科(工
學碩士)
1985年9月～現在：東義工業專門大學電
子通信科 專任講師
1986年3月：東亞大學校 電子工學科 博
士 過程



鄭萬永(Man Yung CHUNG) 正會員
1926年11月1日生
1948年8月：서울大學校 予科理甲卒
1952年3月：大阪大學通信工學科(工學士)
1960年10月：上記大學(工學博士)
1961年1月：遞信部工務局傳送課長
1962年8月：原子力研究所電子工學室長
1967年4月：科學技術研究所研究室副
所長
1977年12月：電子·通信研究所長
1984年9月～現在：東亞大學校 教授