論 文

M-AND, M-OR, NOT 연산을 이용한 다치 논리함수의 간단화에 관한 연구

正會員 宋 洪 復* 正會員 金 永 珍* 正會員 金 明 起**

A Study on Minimization of Multiple-Valued Logic Functions using M-AND, M-OR and NOT Operators

Hong Bok Song*, Young Jin Kim*, Myong Ki Kim** *Regular Members*

要 約

본 논문에서는 Lukasiewicz가 제시한 M-AND, M-OR, NOT연산을 기본으로 하는 다치(Multiple-Valued)논리함수의 간단화 방법을 제시하였다.

먼저 간단화를 행하기 위해서는 Cube를 나열하는 방법에 의해서 그 결과가 틀리기 때문에 가장 효과적인 인접항을 찾는 방법은 간단화에서 무엇보다도 중요하다.

이 방법에 의하여 진리표에 주어진 2변수 다치논리함수를 분해하고 이함수로부터 적항수의 갯수를 비교하였다.

본 논문의 방법에 의하면 기존방법[3]에 비해 동일한 함수를 실현시키는데 소자수 및 코스트가 상당히 감소됨이 밝혀졌다.

ABSTRACT

This paper offers the simplification method of Multiple-Valued logic function based on M-AND, M-OR, NOT operation presented by Lukasiewicz. First in performing the simplification the result is different by the method to arrange Cube, the method to find the most effective adjacent term if, most of all, important in simplification. According to this method, the two-variable multiple-valued logic function given by thruth table is decomposed.

The simplification method in this paper proves that the number of devices and cost is considerably reduced comparing with the existing method[4] to realize the same logic functions.

I. 서 론

논리 레벨을 다치(Multiple-vaiued)화 하는 것에 의해 IC핀의수와 배선수를 줄일수 있고 또한 동작의 고속화 특히 직렬 전송의 고속화등이 기대 된다. 그러나 2치 논리의 경우와 같이 통일적인 논리계가 확립되지 않는것 등이 실용화를 할때에 문제점으로 되고 있다. 본 논문에서 Lukasiewicz에 의해 얻은 연산(문헌(3)에 의해서 M-AND, M-OR, NOT연산이라 한다)을 기본 연산으로 하는 다치 논리 함수의 간단화

*東義大學校 電子工學科
Dept. of Electronics Engineering Dong Eui Univ.
**東亞大學校 電子工學科
Dept. of Electronic Engineering, Dong-A University
論文番號 : 92-59(接受1991. 12. 26)

방법을 제안한다. 본 방법은 문헌[3],[4]에 표시된 표준형 및 Cube 표현을 이용해서 간단화를 행하는 것이지만 문헌[4]에 표시한 방법과 비교해서 특수한 경우를 제외하고 간단화를 행할 수 있다.

Ⅱ. p치 논리함수의 수학적 배경

$L=\{0, 1, 2, 3, \cdots, p-1\}$, p치 변수 X_i, $X_j \in L(1 \le i, j \le n)$이라 하고 p치 n변수 논리 함수 $f(x_1, x_2, \cdots, x_n)$를 $L^n \to L$라고 정의한다.

[정의1] p치 변수 x_i, $x_j \in L$에 대해서 M-AND연산 $x_i x_j$를 다음과 같은 형태로 정의한다.

$$x_i x_j \begin{cases} 0 & ;\ x_i+x_j \le p-1 \\ x_i+x_j-(p-1) & ;\ x_i+x_j > p-1 \end{cases} \tag{1}$$

[정의2] p치 변수 x_i, $x_j \in L$에 대해서 M-OR연산 $x_i \vee x_j$를 다음과 같이 정의한다.

$$x_i \vee x_j \begin{cases} p-1 & ;\ x_i+x_j \ge p-1 \\ x_i+x_j & ;\ x_i+x_j < p-1 \end{cases} \tag{2}$$

[정의3] p치 변수 $x_i \in L$에 대해서 NOT 연산$\bar{x}_i$를 다음과 같이 정의한다.

$$\bar{x}_i=(p-1)-x_i \tag{3}$$

단, +, -는 보통 산술연산을 나타낸다.

위의 M-AND, M-OR, NOT연산은 교환법칙, 결합법칙, 보상법칙, 분배법칙, De-Morgan정리를 따른다. 또한 $a_i x_i$의 m번째 M-OR 의미에서 화 $\{(a_i x_i) \vee (a_i x_i) \vee \cdots \vee (a_i x_i)\}$를 $(a_i x_i)$으로 나타낸다. 정의 1,2로 부터 $(px)_{p-1}=p-1$이다.

Ⅲ. p치 논리함수의 표준형

문헌[3],[4]에 의해 p치 논리함수 표준형은 다음과 같이 된다.

$$f(x_1,x_2,\cdots,x_n)=\sum_{\alpha \in L^n} f(\alpha)\prod_{i=1}^{n}((p-a_i)x_i)_{p-1}((a_i+1)\bar{x}_i)_{p-1} \tag{4}$$

단, $\alpha=(a_1,a_2,\cdots,a_n) \in L^n$, $1 \le i \le n$이고 $\sum$는 보든 α에 대해서 M-OR의미에서 총합을 표시하며 π는 M-AND의미에서 승적을 표시한다. 또 $a_i \le a_i'$라고 하면

$$((p-a_i)x_i)_{p-1}((a_i'+1)\bar{x}_i)_{p-1}= \begin{cases} p-1 & ;\ a_i \le x_i \le a_i' \\ 0 & ;\ 1\ \text{이외일때} \end{cases} \tag{5}$$

가 성립한다. 식(5)는 $a_i=a_i'$일때 식(4)의 적항과 일치한다.

Ⅳ. Cube 표현

간단화를 행할려고 할때에 함수를 보다 효율적으로 표현하는 방법으로서 문헌[4]에 표시되고 있는것처럼 Cube 표현을 이용한다. Cube표현은 적항

$$\Pi((p-a_i)x_i)_{p-1}(a_i'+1)\bar{x}_i)_{p-1} \tag{6}$$

에 1대1 형태로 대응해 있으며 다음과 같이 정의된다.

[정의4] 함수값 $k(1 \le k \le p-1)$를 갖는 항의 적항 $\prod_{i=1}^{n}((p-a_i))x_i)_{p-1}((a_i'+1)\bar{x}_i)_{p-1}$에 대해서 Cube $\mathbf{C}_k^+$은 $\mathbf{C}_k^+=\mathbf{C}_1\mathbf{C}_2\cdots\mathbf{C}_i\cdots\mathbf{C}_n$ $(1 \le i \le n)$ $\mathbf{C}_i$를 좌표i 벡터라고 한다. 여기서 $\mathbf{C}_i=(c_{i0}, c_{i1}, \cdots, c_{ij}, \cdots, c_{ip-1})$ $(0 \le j \le p-1)$이고 c_{ij}는 $a_i \le j \le a_i'$일때 1, 그 이외의 값일때 0를 갖는다. 다음으로 Cube간에 있어서 다음 처럼 OR연산에대한 산술합을 정의한다.

[정의5] Cube간의 OR연산 $\mathbf{C}^* \cup \mathbf{C}^{*\prime}$, 성분간 산술화 $\mathbf{C}^*+\mathbf{C}^{*\prime}$를 아래와 같이 정의한다. $\mathbf{C}^* \cup \mathbf{C}^{*\prime}=\mathbf{C}_1\mathbf{C}_2\cdots\mathbf{C}_i\cdots\mathbf{C}_n \cup \mathbf{C}_1'\mathbf{C}_2'\cdots\mathbf{C}_i'\cdots\mathbf{C}_n'=(\mathbf{C}_1 \cup \mathbf{C}_1')\cdots(\mathbf{C}_i \cup \mathbf{C}_i')\cdots(\mathbf{C}_n \cup \mathbf{C}_n')$ 단, $\mathbf{C}_i \cup \mathbf{C}_i'=(\max(c_{i0},c_{i0}'), \max(c_{i1},c_{i1}'),\cdots, \max(c_{ip-1}, c_{ip-1}'))$라고 한다. $\mathbf{C}^*+\mathbf{C}^{*\prime}=\mathbf{C}_1\mathbf{C}_2\cdots\mathbf{C}_i\cdots\mathbf{C}_n+\mathbf{C}_1'\mathbf{C}_2'\cdots\mathbf{C}_i'\cdots\mathbf{C}_n'=(\mathbf{C}_1+\mathbf{C}_1')\cdots(\mathbf{C}_i+\mathbf{C}_i')\cdots(\mathbf{C}_n+\mathbf{C}_n')$ 단, +는 보통 산술적인 가산을 의미한다. [예1] 그림 1에 표시되는 진리치표에 의해 주어진 4치(four-valued) 2변수의 경우에 있어서 1값을 갖는 셀(Cell)에 대응하는 Cube 표현은 (1000)(0010)이고, 2값을 갖는 셀에 대응하는 Cube 표현은 (0100)(0100), 3값을 갖는 셀에 대응하는 Cube 표현은 (0010)(0100)으로 된다. 또한 2값을 갖는 셀에 대응하는 Cube와 3값을 갖는 셀에 대응하는 Cube와 OR연산을 하면 (0110)(0100)으로 되고 산술합은 (0110)

(0200)으로 된다.

$x_1 \backslash x_2$	0	1	2	3
0	0	0	1	0
1	0	2	0	0
2	0	3	0	0
3	0	0	0	0

그림 1.

V. 간단화 방법

본 방법에서 식(4)의 표준형 각항을 Cube에 의해서 표현하고 Cube간 인접관계를 조사함으로써 간단화를 행한다. 인접관계는 Cube간 OR연산인산술화를 구하고 그 결과에 의해서 판단한다. 구체적으로 아래와 같은 조건을 만족하는 경우에 두개의 Cube (즉, 그들의 Cube에 의해서 표현되는 항)는 인접해 있다고 한다.

조건1 : 2개의 Cube에서 OR연산에 의해서 얻어진 Cube 좌표 i벡터 $\mathbf{C}_i$성분에 …10…01… 같은 배열이 존재하지 않을것. 조건2 : $\mathbf{C}_i+\mathbf{C}_i' \in \{0, 1\}^P$를 만족하는 i $(1 \le n)$이 유일하게 한개 존재하고 $\mathbf{C}_j+\mathbf{C}_j' \in \{0, 2\}^P$ $(p \le j \le n,\ i \ne j)$가 성립할것. [예2] 그림1에 표시한 함수의 경우에 있어서 $\mathbf{C}_1^* \cup \mathbf{C}_2^{*\prime}=(1100)(0110)$, $\mathbf{C}_1^*+\mathbf{C}_2^*=(1100)(0110)$ 따라서 조건2를 만족하고 있지 않기 때문에 1항과 2항은 인접하고 있지 않다. 또 $\mathbf{C}_2^* \cup \mathbf{C}_3^*=(0110)(0100)$, $\mathbf{C}_2^*+\mathbf{C}_3^{*\prime}=(0110)(0200)$. 위의 조건 1,2를 만족하기 때문에 2항과 3항은 인접해 있다.

식(4)의 표준형 적항 $\Pi((p-a_i)x_i)_{p-1}((a_i+1)\bar{x}_i)_{p-1}$은 식(5)로부터 $\alpha=(a_1, a_2, \cdots, a_n) \in L^n$에 의해서 표시 되는 셀 0이외 값을 갖고 있는 경우에 p−1로 되고 그 이외 셀을 표시하는 값이 0일 때에는 0으로 된다.

따라서 식(4)에 표시하는 표준형은 α에 의해서 표시된 셀이 0이외의 값을 표시할 때에는

$$\sum_{\{\alpha|\alpha\in L^n\}} f(\alpha)\prod_{i=1}^{n}((p-a_i)a_i)_{p-1}((a_i+1)\bar{a}_i)_{p-1}$$
$$=0\vee0\vee\cdots\vee f(\alpha)((p-a_i)x_i)_{p-1}((a_i+1)\bar{a}_i)_{p-1}\vee\cdots\vee0$$
$$=f(\alpha)((p-a_i)a_i)_{p-1}((a_i+1)\bar{a}_i)_{p-1}=f(\alpha) \qquad (7)$$

여기서 $f(\alpha)=p-1$인 때에는 식(7)은 M-OR 정의로 부터

$$(p-1)\prod_{i=1}^{n}((p-a_i)x_i)_{p-1}((a_i+1)\bar{x}_i)_{p-1}$$
$$=1((p-a_i)x_i)_{p-1}((a_i+1)\bar{x}_i)_{p-1}\vee(p-2)\prod_{i=2}^{n}((p-a_i)x_i)_{p-1}((a_i+1)\bar{x}_i)_{p-1}$$
$$=1((p-a_i)x_i)_{p-1}((a_i+1)\bar{x}_i)_{p-1}\vee(p-2)\prod_{i=2}^{n}((p-a_i)x_i)_{p-1}((a_i+1)\bar{x}_i)_{p-1}$$
$$=1((p-a_i)x_i)_{p-1}((a_i+1)\bar{x}_i)_{p-1}\vee1((p-a_i)x_i)_{p-1}((a_i+1)\bar{x}_i)_{p-1}\vee\cdots\vee 1((p-a_i)x_i)_{p-1}((a_i+1)x_i)_{p-1} \qquad (8)$$

로 고쳐 쓸수가 있다. 또 M-AND 정의로 부터 $(p-1)X=X$로 되므로 적항중에 $(p-a_i)$와 (a_i+1)이 p−1로 될때에는 정수 p−1을 제거할 수 있다. 따라서 논리 함수 표현에 사용하는 정수를 줄일수 가 있으며 실제로 어떤 회로를 설계할때 논리 게이트의 정수 입력도 감소 시키는 것이 가능하기 때문에 배선수의 감소를 도모할 수 있다. 예를들면 $a_i=1$ 일때 $((p-a_i)x_i)_{p-1}((a_i+1)x_i)_{p-1}=((p-1)x_i)_{p-1}((2\bar{x}_i)_{p-1}=(x_i)_{p-1}(2x_i)_{p-1}$으로 된다. 이 경우 얻어진 결과는 원래의 적항과 비교해서 정수 p−1이 생략 가능하는 것을 안다. 또한 식(8)로 부터 중복되는 동일형의 Cube가 n개 존재할 경우 그들을 모아서 n×(대응하는 적항)로 표현하는것에 의해서 논리 함수중 적항의 갯수를 감소시킬 수 있다. 그림1 함수의 2셀은 Cube 표현으로는 (0100)(0100), (0100)(0100)과 2개의 Cube로 표현되는 적항에 의해서 표현할 경우 본 방법에서는 $2(3x_1)_3(2\bar{x}_1)_3(3x_2)_3(2\bar{x}_2)_3=2(x_1)_3(2\bar{x}_1)_3(x_2)_3(2\bar{x}_2)_3$으로 표현되며 문헌[4]의 표현 $(x_1)_3(2\bar{x}_1)_3(x_2)_3(2\bar{x}_2)_3\vee(x_1)_3(2\bar{x}_1)_3(x_2)_3(2\bar{x}_2)_3$으로 부터 적항의 갯수를 감소 시킬수 있다. 이상으로부터 표준형에서 주어진 논리함수 적항의 갯수 감소와 정수의 감소를 행해서 간단화된 함수를 얻는다. 다음에 구체적인 순서를 제시한다.

[순서 1] : 주어진 p치 논리 함수를 식(4)에 따라서 전개한다.

[순서 2] : 순서 1에서 얻어진 표준형의 각항중에 $f(\alpha)=p-1$로 되는 항을 식(8)에 따라서 전개한다.

[순서 3] : 각항을 정의4에 따라서 Cube 표현한다. r=1로 한다.

[순서 4] : Cube를 $f(\alpha)<p-2$ Cube와 $f(\alpha)=p-2$ Cube를 구별해서 세로로 줄지어서 쓰고 제r열로 한다.

[순서 5] : 제r열 $f(\alpha)<p-2$ Cube들과 $f(\alpha)=p-2$ Cube중에서 각각 인접 관계를 조사한다. 인접한 Cube가 존재하지 않는 경우는 순서 7 상태로 간다. 인접한 Cube가 존재할때에는 OR연산에 의해서 얻어진 Cube를 제 $(r+1)$열로서 $f(\alpha)<p-2$ Cube와 $f(\alpha)=p-2$ Cube를 구별해서 세로로 나열해서 쓴다. 이때 제r열 2개의 Cube에 표시를 해둔다. 이상의 조작을 인접하는 Cube가 존재하지 않을때까지 행한다.

[순서 6] : $r \leftarrow r+1$로 하고 순서 4 상태로 간다.

[순서 7] : 표시되어 있지 않는 Cube를 식(6)에 따라서 적항으로 전개한다.

[순서 8] : 순서 8 상태에서 얻어진 적항을 식(4)에 따라서 전개한다. 이때 식(4)의 $f(\alpha)$값은 $f(\alpha)=p-2$ 경우에는 $p-2$로 되고 그 이상의 경우에는 동일 Cube 갯수로 된다. 여기서 동일 Cube란 각각 Cube 좌표 i벡터 성분의 0.1 배열이 같다는 것을 말한다. 이상의 처리를 행함으로서 논리함수의 간단화를 행할 수 있다.

Ⅵ. 적용예

그림2에 나타난 것처럼 4치 2변수 논리 함수 $f(x_1, x_2)$에 대해서 고찰해 보자.

순서 1 로 부터 : $f(x_1,x_2)=1(2x_1)_3\ (3\bar{x}_1)_3\ (3x_2)_3\ (2x_2)_3 \vee 2(4x_1)_3\ (1\bar{x}_1)_3\ (2x_2)_3\ (3\bar{x}_2)_3 \vee 2(3\bar{x}_1)_3\ (2\bar{x}_1)_3\ (2x_2)_3\ (3\bar{x}_2)_3 \vee 3(3x_1)_3\ (2\bar{x}_1)_3\ (3x_2)_3\ (3x_2)_3$

순서 2 로 부터 : $f(x_1,x_2)=1(2x_1)_3\ (3\bar{x}_1)_3\ (3x_2)_3\ (2\bar{x}_2)_3 \vee 2(4x_1)_3\ (1\bar{x}_1)_3\ (2x_2)_3\ (3\bar{x}_2)_3 \vee 2(3x_1)_3\ (2\bar{x}_1)_3\ (2x_2)_3\ (3x_2)_3 \vee 1(3x_1)_3\ (2\bar{x}_1)_3\ (3x_2)_3\ (2\bar{x}_2)_3 \vee 2(3x_1)_3\ (2\bar{x}_1)_3\ (3x_2)_3\ (2\bar{x}_2)_3$

순서 3과 순서 4에서 : 각 항의 Cube표현은 표1의 제1열과 같이 된다.

순서 4와 순서 6에서 : 제1열 (0010)(0100)과 (0100)(0100), (1000)(0010)과 (0100)(0010)은 각각 인접하고 있기 때문에 Cube간 OR연산의 결과를 제2열에 쓴다. 제2열에는 인접하는 Cube는 존재하지 않는다.

$x_1 \backslash x_2$	0	1	2	3
0	0	0	2	0
1	0	3	2	0
2	0	1	0	0
3	0	0	0	0

그림 2.

표 1.

	제 1 열	제 2 열
$f(\alpha)=1$	* (0010)(0100) * (0100)(0100)	(0110)(0100)
$f(\alpha)=2$	* (1000)(0010) * (0100)(0010) (0100)(0100)	(1100)(0010)

순서 7과 순서 8에서 : $f(x_1,x_2)=1(3x_1)_3\ (3\overline{x_1})_3\ (3x_2)_3\ (2\bar{x}_2)_3 \vee 2(4x_1)_3\ (2\bar{x}_1)_3\ (3x_2)_3\ (2\bar{x}_2)_3 \vee 2(3x_1)_3\ (2\bar{x}_1)_3\ (3x_2)_3\ (2\bar{x}_2)_3 = 1(x_1)_3\ (\bar{x}_1)_3\ (x_2)_3\ (2\bar{x}_2)_3 \vee 2(2\bar{x}_1)_3\ (x_2)_3\ (2\bar{x}_2)_3 \vee 2(x_1)_3\ (2\bar{x}_1)_3\ (x_2)_3\ (2x_2)_3.$

이 예의경우 표준형과 비교해서 항의수도 감소하고 또 식중의 정수도, 표준형이 1,2,3,4를 사용하는것에 대해서 1과 2만이 감소하고 있다. 따라서 명확히 간단화되고 있음을 알 수 있다.

Ⅶ. 결 론

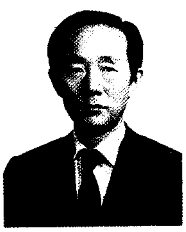
본 논문에서는 Lukasiewicz가 제시한 논리합, 논리적, 부정 연산을 기본 연산으로 하는 p치 논리 함수의 간단화 방법에 대해서 표시했다. 우선 문헌[4]에 표시되고 있는 방법에 비교해서도 명확하듯이 더욱 간단화를 행할 수가 있다. 또, 논리 함수중에서 사용하는 정수가 감소하고 있다고 하는것은 실제의 배선을 할때에 유리하다고 생각된다. 그러나 간단화를 할때에 Cube를 나열하는 방법에 의해서 결과가 틀린다는 것을 고려하기 때문에 간단화에 가장 효과적인 인접항을 찾는 방법등을 잊어버린다. 또 감소시키는 것이

가능한 정수의 수가 p의 값에 관계 없이 일정하므로 p가 크게되면 그다지 큰 장점등이 없다는 것등도 금후의 과제로 남는다.

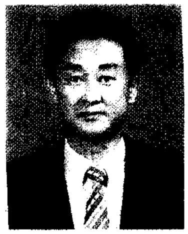
참 고 문 헌

1. M. Helliwell and M. Perkowski, "A fast algorithm to minimize multi-output mixed-polarity generalized Reed-Muller forms," 25th DAC, pp. 427-432, 1988.
2. T. Sasao, "Multiple-valued decomposition of generalized Boolean functions and the Complexity of programmable eagic arrays," IEEE Trans. on Comput., Vol.C-30, No.9, pp.635-643, Sept.1981.
3. 藤田, 佐藤, 小高 "M-AND, M-OR, 及び NOT 演算による p値論理關數の 標準形" 信學論(D)J67-D, No.10., pp.1266-1267.
4. 藤田, 佐藤, 小高 "p値論理關數の一簡素化" 信學論(D)J69-D, No.3, pp.447-479.
5. S. Muroga, Logic design and Switching Theory, Wiley-Interscience publication, 1979.
6. G. papakonstantinou, "Minimization of Modulo-2 sum of products," IEEE Trans. Conput., C-28, pp.163-167, 1979.
7. S. Jo Hong, D. L. Ostapko, "FITPLA : A Programmable logic anay for function independent testing," 10th Int. Snymp. I. C. D., pp. 131-136, TUNe 1980.
8. E. J. McClusky. Jr, "minimization of boolean function," Bell Syst. Tech. J. vol.35, pp. 1417-1444(Apr. 1957).

宋 洪 復(Hong Bok Song) 정회원
1956년 4월 14일생
1983년 2월 : 光云大學校 電子通信工學科 卒業(工學士)
1985년 2월 : 仁荷大學校 大學院 電子工學科 卒業(工學碩士)
1985년~1990년 : 東義工業專門大學 電子通信科(助敎授)
1988년~1989년 : 日本 九州工大 情報工學部(客員硏究員)
1990년 8월 : 東亞大學校 大學院 電子工學科(工學博士)
1991년 3월 : 東義大學校 工科大學 電子工學科(專任講師)
※주관심분야 : 多値論理理論, 函數構成 및 回路設計, 컴퓨터構造및 VLSI設計 등임



金 明 起(Myong Ki Kim) 正會員
1930年 2月 19日生
1951年 8月 : 海軍士官學校 卒業(理學士)
1958年 6月 : 美國海軍工科大學 電子工學科 卒業(工學士)
1966年 2月 : 서울大學校 大學院 電子工學科 卒業(工學碩士)
1976年 2月 : 東亞大學校 大學院(工學博士)
1977年 3月~現在 : 東亞大學校 工科大學 電子工學科敎授



金 永 珍(Young Jin Kim) 정회원
1945년 1월 9일생
1968년 2월 : 광운대학교 응용전자공학과 졸업(공학사)
1980년 10월 : 건국대학원 전자공학과 졸업(공학석사)
1967년 11월~1968년 3월 : 체신부 서울초단파(M/W) 건설국
1968년 3월~1971년 6월 : ROTC 6기(통신)
1971년 9월~1972년 9월 : 주월 한국군 방송국(베트남)
1972년 9월~1972년 12월 : 월남 보건성 전자의료팀
1976년 3월~1977년 2월 : 동의공전 통신과 전임강사
1977년 3월~1980년 2월 : 경동공전 전자과 조교수
1980년 3월~현재 : 동의대학교 전자공학고 부교수
1991년 3월~현재 : 경남대학교 대학원 전자공학과 박사과정 재학중
※주관심분야 : 회로및 시스템, 의용전자