# 2개의 곱항에서 공통인수를 이용한 논리 분해식 산출

권오형
한서대학교 인터넷공학과
e-mail : ohkwon@*hanseo.ac.kr*

# Boolean Factorization Technique Using Two-cube Terms

Oh-Hyeong Kwon
Dept. of Internet Engineering
Hanseo University

## Abstract

A factorization is an extremely important part of multi-level logic synthesis. The number of literals in a factored form is a good estimate of the complexity of a logic function, and can be translated directly into the number of transistors required for implementation. Factored forms are described as either algebraic or Boolean, according to the trade-off between run-time and optimization. A Boolean factored form contains fewer number of literals than an algebraic factored form. In this paper, we present a new method for a Boolean factorization. The key idea is to identify two-cube Boolean subexpression pairs from given expression. Experimental results on various benchmark circuits show the improvements in literal counts over the algebraic factorization based on Brayton's co-kernel cube matrix.

## I. 서론

논리함수는 리터럴(literal) 개수가 다른 여러 형태의 논리식으로 표현될 수 있다. MOS 회로의 경우 트랜지스터의 개수는 논리식의 리터럴 개수와 비례한다. 즉, 논리식의 리터럴 개수를 최소화하면 회로의 트랜지스터 개수를 최소화할 수 있게 된다. 따라서, 논리식 최적화의 한 분야가 최소 개수의 리터럴을 갖는 논리식을 산출하는 것이다. 일반적으로, 분해식(factored form)은 단순식(sum-of-products form)보다 적은 리터럴 개수로 동일한 논리함수를 표현할 수 있는 수단이다. Lawler[1]는 최적 분해식을 산출할 수 있는 알고리즘을 제안하였다. 이 알고리즘은 주어진 논리함수에 대하여 가능한 모든 분해식을 산출하고 그 중에서 가장 적은 리터럴 개수를 갖는 분해식을 산출하는 전수탐색(exhaustive search) 방법을 사용한다. 따라서, 매우 작은 논리함수 또는 회로에 대한 분해식을 산출하는 데를 제외하고는 장시간의 수행시간이 소요되기 때문에 실용성이 없다. 따라서, 분해식 산출에 대해서는 주로 선험 방법(heuristic method)이 적용되고 있다. 분해식 산출의 선험 방법은 분해식 산출 수행 시간과 리터럴 개수의 최적화 중에 어디에 더 중점을 두느냐에 따라 대수 분해 방법과 부울 분해 방법으로 구분한다. 대수 분해 방법에 있어서는 논리함수를 대수 다항식으로 간주한다. Brayton 등[2-5]은 코커널/커널을 이용하여 대수 분해식을 산출하는 방법 즉, 커널 기반 방법(kernel-based method)을 제안하였다. 이 분해 방법은 분해식 산출 속도를 빠르게 향상시켰으나, 때때로 최소 개수의 리터럴을 갖는 분해식을 산출할 수 없다. 그러나, 부울 분해 방법은 대수 분해 방법과 비교하면 보다 적은 개수의 리터럴을 갖는 분해식을 산출할 수 있는 장점을 갖는다. 다음은 최근의 부울 분해 방법들이다. Liao 등[6]은 다치 함수(multi-valued function)와 분지 및 한계 커버 방법(branch-and-bound covering)을 이용한 부울 분해 방법을 제안하였다. Stanion 등[7]은 Liao의 방법이 대수 분해 방법에 비해 리터럴 개수를 줄이는 데 효과가 매우 작다고 지적하였다. 그래서, 효과를 높이기 위해서 Stanion 등은 부울 분해식을 산출하는 데 Binary

Decision Diagram(BDD)를 이용하였다. 또한, 이들은 부울 나눗셈과 부울 분해를 수행하기 위해서 BDD를 활용하는 방법을 제시하였다. 그러나, 이 방법은 때때로 대수 분해에 의한 분해식보다 리터럴 개수가 늘어나는 결과를 초래한다. 가장 최근의 연구 결과로 Kwon 등[8]은 코커널 큐브 행렬을 확장하여 부울 분해식 산출 방법을 제시하였다. 이 방법은 대수 분해식을 산출하기 위한 SIS에서 사용하는 행렬을 포함하기 때문에 대수 분해식보다 좋은 결과를 산출할 수 있다. 그러나, 주어진 논리식에서 커널 집합을 구하고, 다시 커널-커널 집합을 찾는 과정 때문에 수행 시간이 늘어나는 단점을 갖는다. 이처럼 최근의 연구들을 볼 때, 최소 개수의 리터럴을 갖는 부울 분해식 산출에 대한 연구가 더 필요하다.

본 논문에서는 선험 부울 분해 방법을 제안한 것으로, SIS에서 사용되는 커널 기반 방법(즉, 대수 분해 방법)과는 달리 커널을 찾지 않고 2개의 큐브들 간에 공통인수를 추출하여 부울 분해식을 산출한다.

## II. 부울 분해 방법

본 논문에서 제시하는 분해 방법을 서술하는 데 기본이 되는 용어에 대하여 서술한다. 다음 커널/커널과 확장된 코커널 큐브 행렬에 대하여 소개한다. 확장된 코커널 큐브 행렬을 이용하여 부울 분해식을 산출하는 알고리즘을 제시한다.

**정의 1: 변수**(variable)는 부울 공간(Boolean space)에서 한 좌표를 나타내는 문자다. **리터럴**(literal)은 변수 그 자체 또는 그의 보수(complement)다. **큐브**(cube)는 리터럴들의 집합으로 만일 리터럴 $a$가 존재하면, 그의 보수 리터럴 $a'$을 포함하지 않는다. **단순식**(expression 또는 sum-of-products(SOP) form)은 큐브들의 집합이다.

**예 1:** 문자 $a$는 변수다. $a$와 $a'$은 리터럴이다. 리터럴 집합 $a, b$ 는 큐브, 그러나 $a, a'$ 은 큐브가 아니다. $a, b'$ , $b, c$ 는 단순식이다.

본 논문에서는 큐브와 단순식을 표현하는 경우 집합 표기와 보편적으로 사용되는 수식 표기를 모두 사용한다. 따라서 큐브 $a, b$ 는 $ab$와 동일한 표현이며, 단순식 $a, b'$ , $b, c$ 는 $ab'+bc$와 동일한 표현이다.

**정의 2: 분해식**(factored form)은 단순식들이 합과 곱으로 표현된 것이다. 구체적인 정의는 다음과 같다.

1) 리터럴은 분해식이다.
2) 분해식들의 곱은 분해식이다.
3) 분해식들의 합은 분해식이다.

분해식은 단순식들이 합과 곱으로 반복해서 표현된 논리식이다.

**정의 3:** 등멱법칙( $a \cdot a = a$)과 보수법칙( $a \cdot a' = 0$)을 적용할 필요없이 분해식을 구성하는 단순식들을 곱할 수 있는 경우 **대수 분해식**(algebraic factored form)이라 한다. 대수 분해식 이외의 경우 **부울 분해식**(Boolean factored form)이라 한다.

**예 2:** $F = bc'd'e + ab'c + ab'e + ac'd'$와 $F = a(b'(c+e) + c'd') + bc'd'e$ 모두 대수 분해식이다. $F = (a+be)(b'(c+e) + c'd')$는 부울 분해식이다.

### 2.1 공통인수 쌍 추출

주어진 단순형의 논리식에서 2개의 큐브를 선택하고 이 2개의 큐브들에서 공통 인수, 즉 공통 큐브를 찾는다. 이 때, 공통 큐브가 제수고 이 제수로 2개의 큐브를 나눈 것이 몫이 된다. $C$를 제수 집합, $Q$를 2개의 큐브로 구성된 몫 집합이라 하자. 표기상 제수/몫 쌍을 괄호를 이용하여 표현하고, $c_i \in C$ $c_j \in C$ $q_i \in Q$ $q_i \in Q$이고 $i \neq j$라 하자. 그러면, $(c_i, q_i)$, $(c_j, q_j)$는 대수 나눗셈에 의한 제수/몫 쌍을 표현한 것이다. 만일 $c_i \in q_j$, $c_j \in q_i$ 이고 $q_i q_j$ 가 주어진 논리식에 포함되면, $(q_i, q_j)$는 주어진 논리식에 포함되는 부분 부울 부울식이라 한다.

**예 3:** 논리식 $F = bc'd'e + ab'c + ab'e + ac'd'$으로부터 표1과 같이 각 큐브에 양의 값을 갖는 인덱스를 부여한다. 다음 2개의 큐브들 사이의 공통인수를 추출하여 산출한 대수 분해식을 표2에 나열한다. 표 2에서는 분해식을 제수와 몫을 열(column)로 구분해서 표시한다. 표 2의 제 1열은 설명을 쉽게 하기 위해 부여한 행 번호다.

표 1: 큐브 인덱스

| 큐브 $C_i$ | $bc'd'e$ | $ab'c$ | $ab'e$ | $ac'd'$ |
|---|---|---|---|---|
| 인덱스 $index(C_i)$ | 1 | 2 | 3 | 4 |

표 2: 부분 분해식

| 행 | 선택된 큐브 | 분해식 | |
|---|---|---|---|
| | | 제수 | 몫 |
| 1 | $C_1 \cap C_2$ | $\varnothing$ | $\varnothing$ |
| 2 | $C_1 \cap C_3$ | $e$ | $bc'd'+ab'$ |
| 3 | $C_1 \cap C_4$ | $c'd$ | $be+a$ |
| 4 | $C_2 \cap C_3$ | $ab'$ | $c+e$ |
| 5 | $C_2 \cap C_4$ | $a$ | $b'c+c'd$ |
| 6 | $C_3 \cap C_4$ | $a$ | $b'e+c'd$ |

표 2로부터 행2과 행4의 분해식을 보면 행2의 제수가 행4의 몫에 포함되고, 다시 행4의 제수가 행2의 몫에 포함된다. 또한 행2와 행4의 분해식의 곱(ANDing)을 하면 주어진 논리식에 포함되며, 이는 부울 분해식이 된다. 또 행3과 행5에서, 행3과 행6에서 같은 결과를 얻는다. 이를 정리하면 표 3과 같다.

표 3: 부분 부울 분해식

| 선택된 2개 행 | 몫과 몫 곱 |
|---|---|
| 행2, 행4 | $(bc'd'+ab')(c+e)$ |
| 행3, 행4 | $(be+a)(b'c+c'd)$ |
| 행3, 행5 | $(be+a)(b'e+c'd)$ |

## 2.2 분해식 산출 행렬

단순식 $F$가 주어졌을 때, 분해식 산출 행렬 $M$은 2개의 큐브로부터 표2와 같이 제수와 몫으로 구성된 쌍과 표3과 같은 몫과 몫의 쌍을 이용한다. 행렬 $M$의 행은 제수와 몫의 쌍의 제수들과 몫과 몫의 쌍에 포함되는 모든 큐브로 구성된다. 행렬 $M$의 행은 제수와 몫의 쌍에서 몫을 구성하는 모든 큐브에 열이 배당된다. $C$를 제수들의 집합, $CQ$를 행렬 $M$의 열에 해당하는 큐브 집합으로 표기하고자 한다. $\alpha_i$, $\beta_j$를 각각 $M$의 $i$번째 행, $j$번째 열을 나타낸다고 하면 $\alpha_i \in C$, $\beta_j \in CQ$이다. 이 때, 행렬 $M$의 원소 $M(\alpha_i, \beta_j)$는 다음과 같은 값을 갖는다.

$$M(\alpha_i, \beta_j) = \begin{cases} index(\alpha_i \beta_j) & \text{if } \alpha_i \in C,\ \beta_j\text{는 } \alpha_i\text{로나눈 몫에 속하는 큐브} \\ index(c), & \text{if } \alpha_i\text{와 } \beta_j\text{는 몫과 몫 쌍에 포함되는 큐브,} \\ & \quad c = \alpha_i \beta_j \neq 0 \\ * & \text{if } \alpha_i \beta_j = 0, \\ & \quad \alpha_i\text{와 } \beta_j\text{는 몫과 몫 쌍의 큐브} \\ 0 & \text{if 그 외의 경우} \end{cases}$$

**예 4:** 예3의 $F = bc'd'e + ab'c + ab'e + ac'd$에 대한 표1, 표2와 표3을 이용해서 분해식 산출 행렬 $M$을 만든다. 행렬의 행들은 표2의 제수에 해당하는 큐브와 표3의 몫과 몫의 쌍에 포함되는 큐브들에 대응하도록 한다. 다음 열은 표2의 몫에 포함되는 큐브들에 대응되도록 한다. 그러면, 행렬의 행은 집합 $\{e, c'd', ab', a, bc'd', c, be, b'c, b'e\}$의 각 원소에 대응된다. 행렬의 열은 집합 $\{bc'd', ab', be, a, c, e, b'c, c'd, b'e\}$의 각 원소에 대응된다. 산출된 분해식 산출 행렬은 표 4와 같다. 표 4의 행렬에서 첫 번째 행과 첫 번째 열에 해당하는 $M(1,1) = M(e, bc'd')$의 원소 값은 $index(bc'd'e) = 1$가 된다. 또한 여섯 번째 행과 첫 번째 열에 해당하는 $M(6,1) = M(c, bc'd')$의 원소 값은 $c \cdot bc'd' = 0$가 되기 때문에 $M(6,1) = *$ 가 된다. 나머지 원소들에 대해서도 같은 방법으로 행렬에 값이 할당된다.

표 4: 분해식 산출 행렬

| 행 \ 열 | $bc'd'$ | $ab'$ | $be$ | $a$ | $c$ | $e$ | $b'c$ | $c'd$ | $b'e$ |
|---|---|---|---|---|---|---|---|---|---|
| $e$ | 1 | 3 | | | | | | | |
| $c'd'$ | | | 1 | 4 | | | | | |
| $ab'$ | | | | | 2 | 3 | | | |
| $a$ | | | | | | | 2 | 4 | 3 |
| $bc'd'$ | | | | | * | 1 | | | |
| $c$ | * | 2 | | | | | | | |
| $be$ | | | | | | | * | 1 | * |
| $b'c$ | | | * | 2 | | | | | |
| $b'e$ | | | * | 3 | | | | | |

## 2.3 분해식 산출 행렬 커버와 분해식

행렬 커버는 Brayton 등이 제시한 사각형 커버링 방법과 유사하다. 먼저 사각형에 대한 정의를 하고 커버링 방법에 대하여 설명한다.

**정의 4:** 행렬 $M$에서 **사각형**은 행과 열의 부분 집합 $(R, C)$이며 다음 조건을 갖는다. $\alpha, \gamma \in R$ 및 $\beta, \delta \in C$에 대하여 $M(\alpha, \beta) \neq 0$이며 $\alpha \neq \gamma$이고 $\beta \neq \delta$인 경우 $M(\alpha, \beta) > 0$이고 $M(\gamma, \delta) > 0$이면 $M(\alpha, \beta) \neq M(\gamma, \delta)$. **프라임 사각형**은 다른 사각형에 포함되지 않는 사각형이다. 사각형 $(R, C)$의 **사각형 비용**은 행 $R$과 열 $C$에 포함되는 리터럴들의 개수로 정의한다.

위의 정의는 사각형 또는 프라임 사각형에 포함되는 원소들은 서로 다른 양의 정수 값과 다수의 *(don't-care)를 포함할 수 있음을 의미한다.

**예 5:** 표4와 같은 분해식 산출 행렬 $M$이 주어졌다고 하자. $(a, b'c)$는 사각형이다. 그러나, $(a, b'c)$은

( $a, be$ , $b'c, b'e, c'd$ )에 포함되기 때문에 프라임 사각형은 아니다. 반면에, ( $a, be$ , $b'c, b'e, c'd$ )을 포함하는 사각형이 없기 때문에 프라임 사각형이라 한다. ( $a$ , $b'c, b'e, c'd, c$ )는 $M(a,c)=0$를 포함하기 때문에 사각형이 아니다. 사각형 ( $a, be$ , $b'c, b'e, c'd$ )의 사각형 비용은 행과 열을 나타내는 리터럴의 개수 합으로 9가 된다.

최소 비용을 갖는 사각형의 집합을 산출한 후, 사각형에 대응되는 분해식을 산출한다. 예 5의 프라임 사각형 ( $a, be$ , $b'c, b'e, c'd$ )에 대응되는 분해식가 $F=(a+be)(b'c+c'd+b'e)$가 산출되고, $b'c+c'd+b'e$에 대하여 다시 분해식 산출 행렬을 만들고 사각형 커버를 찾으면, 최종적으로 리터럴 개수가 8개인 분해식 $F=(a+be)(b'(c+e)+c'd)$가 산출된다. 반면에 대수적 분해식 방법으로는 $F=a(b'(c+e)+c'd)+bc'de$가 산출되며 리터럴 개수는 10개가 된다.

## III. 실험 결과

본 논문에서 제안한 방법을 PLA 형의 여러 MCNC (Microelectronics Center of North Carolina) 벤치마크 회로(benchmark circuit)에 대하여 테스트하였고, 그 결과를 SIS 1.2[5]의 출력과 비교하였다. 널리 사용되는 SIS 1.2는 커널 기반 대수 분해 방법으로 대수 분해식을 산출한다. 본 실험에서 분해식의 리터럴 개수만을 비교할 것이기 때문에, SIS 1.2에서는 분해식 산출 명령 "factor -g *"를 각 벤치마크 회로에 적용하였다. 표5는 실험결과를 SIS 1.2와 비교한 것이다.

표 5: 실험 결과

| 회로 | 입력수 | 출력수 | SIS 1.2 | 제안 방법 |
|---|---|---|---|---|
| rd53 | 5 | 3 | 71 | 69 |
| squar5 | 5 | 8 | 127 | 91 |
| f51m | 8 | 8 | 168 | 165 |
| z4m1 | 7 | 4 | 69 | 66 |
| mux | 5 | 1 | 79 | 72 |
| sct | 19 | 15 | 138 | 137 |
| tcon | 17 | 16 | 48 | 40 |
| cmb | 16 | 4 | 98 | 82 |

## IV. 결론

본 논문은 커널 산출 없이 부울 분해식을 산출할 수 방법을 제시하였다. 주어진 논리식에서 두 개의 큐브를 선택하고 두 개의 큐브로 구성된 2-큐브 쌍 분해식을 산출하는 방법이 커널 산출 방법보다 간단하면서 부울 분해식 산출 가능하게 하는 행렬을 만들 수 있는 기반을 제공한다.

## 참고문헌


[1] E. Lawler, "An Approach to Multilevel Boolean Minimization," *Journal of ACM*, Vol. 11, No. 3, pp. 283-295, July, 1964.

[2] R. K. Brayton and C. McMullen, "The Decomposition and Factorization of Boolean Epressions," *Proc. ISCAS*, pp. 49-54, 1982.

[3] R. K. Brayton, R. Rudell, A. Sangiovanni-Vincentelli, and A. R. Wang, "MIS: A Multiple-Level Logic Optimization System," *IEEE Trans. CAD*, Vol 6, No. 6, pp. 1062-1081, June, 1987.

[4] R. K. Brayton, R. Rudell, A. Sangiovanni-Vincentelli, and A. Wang, "Multi-Level Logic Optimization and the Rectangle Covering Problem," *Proc. ICCAD*, pp. 66-69, 1987.

[5] E. M. Sentovich, K. J. Singh, C. Moon, H. Savoj, R. K. Brayton, R. K., and A. Sangiovanni-Vincentelli, "Sequential Circuit Design Using Synthesis and Optimization," *Proc. ICCD*, pp. 328-333, 1992.

[6] S. Liao, S. Devadas, and A. Ghosh, "Boolean Factoring Using Multiple-Valued Minimization," *Proc. ICCAD*, pp. 606-611, 1993.

[7] T. Stanion and C. Sechen, "Boolean Division and Factorization Using Binary Decision Diagrams," *IEEE Trans. CAD*, Vol. 13, No. 9, pp. 1179-1184, Sep., 1994.

[8] O.-H. Kwon, S. J. Hong, and J. Kim, " A Boolean Factorization Using and Extended Boolean Matrix," *IEICE Trans. Inf. and Sys.*, Vol. E81-D, No. 12, pp. 1466-1472, Dec., 1998.