

다단 논리합성을 위한 출력 Phase 할당 알고리듬

(Output Phase Assignment Algorithm for Multilevel Logic Synthesis)

李 載 興* 鄭 正 和*

(Jae Heung Lee and Jong Wha Chong)

要 約

본 논문에서는 회로 다단화 과정 전단계에서 면적 최소화를 위한 출력 phase 할당 알고리듬을 제안한다.

주어진 함수 표현식의 문자 사이의 관계 정도를 나타내는 평가 함수를 정의하고, 함수 사이의 공통 부분정도를 나타내는 가중치를 정의함으로써 출력 phase 할당을 위한 공통 부분표현식 그래프를 생성한다. 생성된 그래프로 부터 가중치의 합이 최대가 되도록 출력 phase를 할당하는 휴리스틱 알고리듬을 제안하였다. 또한 MCNC benchmark 데이터를 사용하여 실험하였으며 그 결과 본 논문에서 제안한 방법이 우수함을 알 수 있었다.

Abstract

This paper presents a new output phase assignment algorithm which determines the phases of all the nodes in a given boolean network. An estimation function is defined, which is represented by the relation between the literals in the given function expression. A weight function, WT (f_i, f_j) is defined, which is represented by approximate amount of common subexpression between function f_i and f_j . Common Subexpression Graph (CSG) is generated for phase selection by the weight function between all given functions. We propose a heuristic algorithm finding subgraph of which sum of weights has maximum by assigning phases into the given functions. The experiments with MCNC benchmarks show the efficiency of the proposed method.

I. 서 론

최근 VLSI 제조기술의 발달과 함께 칩의 복잡도 및 집적도가 증가함에 따라 설계 자동화를 위한 CAD 시스템 개발에 대한 요구가 증가하였다. CAD 시스템의 장기적인 목표는 동작기술로 부터 설계자에 의한 사양을 만족하는 최적에 근사한 해를 구하는 실리콘으로의 자동합성에 있다. 이러한 CAD 시스템

의 개발은 ASIC(application specific integrated circuit) 시장의 요구가 빠른 속도로 증가함에 따라 그 중요성이 증대되고 있다.

CAD 시스템에서 논리합성 설계는 면적, 지연시간, 그 밖의 제약 조건을 만족하도록 RT(register transfer) 레벨로 부터 게이트 레벨의 net list 정보를 얻는 과정으로 크게 FSM(finite state machine) 설계와 조합논리 회로 설계로 나누어 활발히 연구가 진행되고 있다.

특히, 조합논리회로 설계에서 면적 및 지연시간에서 효과적인 다단논리회로의 설계 요구가 증대됨에 따라 다단논리합성 툴로써 MIS^[2], BOLD^[3] 등과 같

*正會員, 漢陽大學校 電子工學科
(Dept. of Elec. Eng., Hanyang Univ.)
接受日字 : 1991年 6月 5日

은 우수한 시스템이 개발되었다. 이들은 주어진 자연시간의 제약조건 내에서 최소의 면적으로 함수를 구현하는 데로써 주어진 함수에 대한 boolean cover 형태의 표현식을 입력으로 회로 다단화 과정을 거친 후 다단 구조를 가진 부울회로를 얻는다. 이 부울회로는 불필요한 부분을 제거하는 다단논리 최소화와 technology를 고려한 라이브러리로의 매핑 과정을 거쳐 최종적인 net list를 얻게 되는데, 회로 다단화에서 얻어지는 부울회로의 다단구조는 최종 결과에 중대한 영향을 미친다. 상기 툴들의 회로 다단화 과정에서는 주어진 positive 함수에 대해서만 공통 큐브 및 커넬들을 생성하여 공통 부분 표현식을 대치하였기 때문에 complément 함수에 대한 우수한 공통 큐브 또는 커넬을 조사하지 않고 생략될 가능성이 있다.

그러나 간단히 인버터 하나만을 첨가 시킴으로써 phase를 쉽게 변화시킬 수 있기 때문에 complement 함수에 대한 좋은 대수 공통 큐브나 커넬들의 생략을 방지할 수 있다. 이미, PLA와 다단 논리 최소화에서는 각 출력에 대한 phase를 할당함으로써 최종적인 면적에 협력한 차이가 나타나는 것을 입증하였다.^[4,5,6] 다단 논리 최소화를 위한 출력 phase 할당 방법으로^[16]에서는 이미 할당 전의 최적화된 다단 구조의 회로로부터 각 출력의 phase를 결정하는 방법을 제택하고 있어 주어진 다단 구조 상에서 적당한 phase를 결정함으로써 나타나는 회로 다단화 과정에서 필요한 다양한 공통 부분 표현식을 충분히 반영하고 있지 못하고 있다.

PLA 논리 최소화의 목적 함수는 product term의 수와 문자 수의 최소화이므로 면적, 지연 시간, 그 밖의 제약 조건을 목적 함수로 하는 다단논리회로 설계보다 적용하기 쉽다. 다단논리회로를 위한 phase 할당 적용은 크게 3가지로 분류할 수 있다.

- 1) 회로 다단화 전단계의 출력 phase 할당
 - i) 주어진 부울회로의 모든 노드에 phase 할당^[4]
 - ii) i)과 공통 부분 표현식을 대치할 때 생성되는 중간변수를 포함한 phase 할당
- 2) 다단 논리회로 구성 후, 인버터 수를 줄이기 위한 게이트 phase 할당^[2]
- 3) 다단 논리회로를 라이브러리로의 매핑을 위한 phase 할당

본 논문에서는 회로 다단화 과정 전단계에서 면적 최소화를 위한 출력 phase 할당 알고리듬을 제안한다.

2장에서 주어진 함수 표현식의 문자 사이의 관계 정도를 나타내는 평가 함수를 정의하고, 3장에서 함-

수 사이의 공통 부분 정도를 나타내는 가중치를 정의한다. 4장에서는 출력 phase 할당을 위한 각 함수 사이의 공통 부분 표현식 그래프를 생성하고, 생성된 그래프로부터 가중치의 합이 최대가 되도록 출력 phase를 할당하는 허리스틱 알고리듬을 제안한다. 한편, benchmark 데이터를 사용하여 실험한 결과를 5장에 나타내어 본 논문에서 제안한 방법의 우수성을 보이고, 6장에서 본 논문의 결론과 앞으로의 연구 과제를 제시한다.

II. 부울함수 구조의 평가함수

일반적으로 부분 표현식은 각 문자들 사이에 product와 sum의 표현식으로 나타낸다. 특히, Boolean cover 형태의 표현식은 큐브들의 sum 형태 즉, PLA 표현식으로 AND-OR 평면에서 설계되는 바와 같이, 각 문자들 간의 product와 sum의 관계를 이용하여 PLA 표현식 즉 부울함수의 구조를 평가할 수 있다. 주어진 PLA 표현식에서 자체의 구조를 결정하는 요인을 살펴보면 다음과 같다.

- 1) 함수의 변수 수
 - 2) 함수의 큐브 수
 - 3) 함수의 문자 수
 - 4) 각 문자들 간의 product 또는 sum의 관계
- 본 논문에서는 위의 요인들을 이용하여 주어진 함수 f 의 각 문자들 간의 관계를 그 함수의 평가 함수 $EF(f)$ 라고 정의하는데 이것은 함수 f 의 주어진 표현식의 구조를 평가하는 함수로 $EP(f)$ 와 $ES(f)$ 의 합으로 구성된다. $EP(f)$ 와 $ES(f)$ 는 각각 함수 f 의 문자들 간의 product와 sum 관계에 대한 평가 함수들이며, 주어진 함수 f 에 대한 평가함수 $EF(f)$ 는 다음과 같이 정의한다.

$$EF(f) = \alpha \times EP(f) + \beta \times ES(f)$$

$$\text{여기서, } \alpha = \frac{NL(f) - NP(f)}{NL(f) - 1}, \quad \beta = \frac{NP(f) - 1}{NL(f) - 1}$$

여기서 α, β 는 각각 주어진 함수 표현식에서 전체 오퍼레이션 수에 대한 product와 sum 오퍼레이션 수가 차지하는 비율을 나타내는데 그 함수 f 표현식의 문자 수를 $NL(f)$, 큐브(product term)의 수를 $NP(f)$, 변수의 수를 $NV(f)$ 라고 정의했을 때 전체 오퍼레이션 수는 $NL(f) - 1$ 값이 된다. $EP(f)$ 는 함수 f 의 각 큐브에서 문자들 간의 product 관계에 있는 행렬 표현을 나타내는데, $EP_{ij}(f)$ 는 함수 f 에서 i 와 j 번째 문자 사이의 product 관계의 정도를 나타내며 다음과 같이 정의한다. 여기서, l_m, l_n 는 큐브 표현식에서 m 과 n 번째의 값으로 정의한다.

$$EP_{ij}(f) = \sum_{p=0}^{NP(f)-1} R_{ij}(p), 0 \leq i, j < 2NV(f)$$

여기서, if ($(l \lfloor i/2 \rfloor = i \text{ MOD } 2) \text{ AND } (l \lfloor j/2 \rfloor = j \text{ MOD } 2)$ in p-th product term)
 $R_{ij}(p) = 1$
else
 $R_{ij}(p) = 0$

같은 방법으로, $ES_{ij}(f)$ 는 함수 f의 각 product term 사이에서 각 문자들간의 sum 관계에 있는 행렬 표현을 나타내고, $ES_{ij}(f)$ 는 함수 f에서 i와 j번째 문자 사이의 sum 관계의 정도를 나타내며 다음과 같이 정의 한다.

$$ES_{ij}(f) = \sum_{q=0}^{NP(f)-1} \sum_{p=q}^{NP(f)-1} R_{ij}(p, q) \times \text{sum_scale}$$

여기서, if ($p = q$) {
sum_scale = 1
if ($i = j \text{ AND } (l \lfloor i/2 \rfloor = i \text{ MOD } 2)$)
 $R_{ij}(p, q) = 1$
else
 $R_{ij}(p, q) = 0$
} else {
sum_scale = $1 / (|C_{pq}| \times |C_{qp}|)$
if ($(l \lfloor i/2 \rfloor \text{ in } C_{pq}) = i \text{ MOD } 2 \text{ AND } (l \lfloor j/2 \rfloor \text{ in } C_{qp}) = j \text{ MOD } 2$)
 $R_{ij}(p, q) = 1$
else
 $R_{ij}(p, q) = 0$
}}

여기서, C_p 와 C_q 가 각각 p와 q번째 큐브라고 할 때 C_p & C_q 는 C_p 와 C_q 의 공통 인수를 의미한다. ‘&’는 각 AND 비트 오퍼레이션을 나타내며, 큐브 C_p 와 C_q 로부터 공통 인수를 제외한 큐브 C_{pq} 와 C_{qp} 는 각각 $C_{pq} = C_p \# (C_p \& C_q)$ 와 $C_{qp} = C_q \# (C_q \& C_q)$ 로 정의한다. ‘#’은 sharp-product^[7]인데, i # j는 큐브 j에 있는 문자에서 큐브 i & j에 있는 문자를 제외시킨 큐브를 의미한다. sum_scale 값을 곱한 의미는 예를 들어, $a + bc$ 에서 문자 a와 b의 sum 관계는 $(a+b)(a+c)$ 에서 1/2의 확률과 같은 의미로 문자 i와 j가 sum의 관계로 나타날 확률을 의미한다. 즉, 공통 인수를 제외한 큐브 사이의 sum 오퍼레이션의 확률을 의미하게 된다. 위의 관계식에서 함수 f의 문자 수 $NL(f)$ 는 다음과 같다.

$$NL(f) = \sum_{i=0}^{2NV(f)-1} ES_{ii}(f) = \sum_{i=0}^{2NV(f)-1} EP_{ii}(f)$$

| | \bar{a} | a | \bar{b} | b | \bar{c} | c | \bar{d} | d | \bar{e} | e | \bar{f} | f |
|-------------|-----------|-----|-----------|-----|-----------|-----|-----------|-----|-----------|-----|-----------|-----|
| $\bar{a}0$ | 2 | 0 | 2 | 0 | 0 | 2 | 0 | 1 | 0 | 1 | 1 | 0 |
| $a1$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $\bar{b}2$ | 2 | 0 | 2 | 0 | 0 | 2 | 0 | 1 | 0 | 1 | 1 | 0 |
| $b3$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $\bar{c}4$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $c5$ | 2 | 0 | 2 | 0 | 0 | 2 | 0 | 1 | 0 | 1 | 1 | 0 |
| $\bar{d}6$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $d7$ | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| $\bar{e}8$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $e9$ | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| $\bar{f}10$ | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| $f11$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

(a) 행렬 표현식 EP

| | \bar{a} | a | \bar{b} | b | \bar{c} | c | \bar{d} | d | \bar{e} | e | \bar{f} | f |
|-------------|-----------|-----|-----------|-----|-----------|-----|-----------|-----|-----------|-----|-----------|-----|
| $\bar{a}0$ | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $a1$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $\bar{b}2$ | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $b3$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $\bar{c}4$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $c5$ | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| $\bar{d}6$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $d7$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1/2 | 0 |
| $\bar{e}8$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $e9$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1/2 | 0 |
| $\bar{f}10$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1/2 | 0 | 1/2 | 1 | 0 |
| $f11$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

(b) 행렬 표현식 ES

| | \bar{a} | a | \bar{b} | b | \bar{c} | c | \bar{d} | d | \bar{e} | e | \bar{f} | f |
|-------------|-----------|-----|-----------|-----|-----------|-----|-----------|------|-----------|------|-----------|-----|
| $\bar{a}0$ | 2 | 0 | 7/4 | 0 | 0 | 7/4 | 0 | 7/8 | 0 | 7/8 | 7/8 | 0 |
| $a1$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $\bar{b}2$ | 7/4 | 0 | 2 | 0 | 0 | 7/4 | 0 | 7/8 | 0 | 7/8 | 7/8 | 0 |
| $b3$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $\bar{c}4$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $c5$ | 7/4 | 0 | 7/4 | 0 | 0 | 2 | 0 | 7/8 | 0 | 7/8 | 7/8 | 0 |
| $\bar{d}6$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $d7$ | 7/8 | 0 | 7/8 | 0 | 0 | 7/8 | 0 | 1 | 0 | 7/8 | 1/16 | 0 |
| $\bar{e}8$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $e9$ | 7/8 | 0 | 7/8 | 0 | 0 | 7/8 | 0 | 7/8 | 0 | 1 | 1/16 | 0 |
| $\bar{f}10$ | 7/8 | 0 | 7/8 | 0 | 0 | 7/8 | 0 | 1/16 | 0 | 1/16 | 1 | 0 |
| $f11$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

(c) 행렬 표현식 EF

그림 1. 주어진 함수 f에 대한 행렬 표현식 EP, ES, EF

Fig. 1. The matrice EP, ES and EF for the given function f.

결국, EP(f)와 ES(f)의 대각선 행렬 값의 합은 그 함수 f의 문자 수와 같다.

따라서 함수 f의 평균 함수 EF(f)을 구함으로써 그 함수 f의 구조를 어느정도 예측할 수 있고 함수와

함수 사이에 공통 부분 표현식의 정도를 나타내는 가중치를 추출하는데 이 평가 함수를 이용하게 된다.
[예제 1] $f = abcde + \bar{a}\bar{b}cf$ 에 대한 큐브 표현식은

$$\begin{array}{c} l_0 \quad l_1 \quad l_2 \quad l_3 \quad l_4 \quad l_5 \\ \left[\begin{array}{cccccc} P_0 : & 0 & 0 & 1 & 1 & 1 & - \\ P_1 : & 0 & 0 & 1 & - & - & 1 \end{array} \right] \end{array}$$

로 주어지며 이 표현식에 대한 행렬 표현식 EP, ES, EF를 구하면 그림1과 같다.

예로써, 행렬 표현식 EP에서 $EP_{2,7}=1$ 인 이유는 다음과 같다.

P_0 에서 $R_{2,7}=1$:

$$(l_2/2) = l_1 = 2 \text{ MOD } 2 = 0 \text{ AND}$$

$$(l_7/2) = l_3 = 7 \text{ MOD } 2 = 1$$

P_1 에서 $R_{2,7}=0$:

$$(l_2/2) = l_1 = 2 \text{ MOD } 2 = 0 \text{ AND}$$

$$(l_7/2) = l_3 \neq 7 \text{ MOD } 2$$

행렬 표현식 ES에서 $ES_{7,10}=1/2$ 인 이유는 다음과 같다.

$$p = P_0, q = P_1, C_{pq}' = de, C_{qp}' = f, \text{sum_scale} = 1/2$$

$$R_{7,10}(p, q) = 1 :$$

$$(C_{pq}' \text{에서 } l_7/2 = l_3 = 7 \text{ MOD } 2 = 1) \text{ AND}$$

$$(C_{qp}' \text{에서 } l_1/2 = l_6 = 10 \text{ MOD } 2 = 0)$$

따라서, $R_{7,10}(p, q) \times \text{sum_scale} = 1/2$

행렬 표현식 EF는 $NL(f) = 9$ 이고, $NP(f) = 2$ 이므로 $\alpha = 7/8$ 이고, $\beta = 1/8$ 이다. 따라서, $EF = 7/8 \times EP + 1/8 \times ES$ 에 의해서 얻어진다.

III. 함수들간의 가중치 함수

2단 논리회로에서 다단 논리회로의 합성은 각 함수들 간의 공통 부분 표현식을 추출하여 대치하는 작업을 반복 수행하게 되는데, 각 함수 표현식의 구조는 그 함수에서 문자들 사이의 관계 정도를 평가 함수로 정의함에 따라 어느 정도 예측할 수 있는 것을 앞장에서 보였다.

본 장에서는 2함수사이의 공통 부분 정도를 나타내는 가중치 함수를 정의한다. 함수 f_p 와 f_q 사이의 가중치에 대한 행렬 표현식은 $W(f_p, f_q)$ 로 나타내며, $W_{ij}(f_p, f_q)$ 는 함수 f_p 와 f_q 사이에서 i번째 문자와 j번째 문자와의 관계정도를 나타내는 가중치를 의미하고 다음과 같이 정의한다.

$$W_{ij}(f_p, f_q) = \min(EF_{ij}(f_p), EF_{ij}(f_q)) \quad 0 \leq i, j < NV(f)$$

위 표현식에서 평가 함수 $EF_{ij}(f_p)$ 와 $EF_{ij}(f_q)$ 는 f_p 와 f_q 에서 각각 i와 j번째 문자 사이의 관계 정도를 나타내고 있으며 함수 f_p 와 f_q 사이에서 i와 j번째 문

자의 공통 부분 정도를 가중치로 정의하기 때문에 최소값을 선택하였다. 따라서 함수 f_p 와 f_q 사이의 전체 가중치 $WT(f_p, f_q)$ 는 모든 문자에 대한 가중치의 합으로써 다음과 같이 정의한다.

$$WT(f_p, f_q) = \sum_{i=0}^{2NV(f)-1} \sum_{j=1}^{2NV(f)-1} W_{ij}(f_p, f_q) - \gamma_p \times DFT(f_p) - \gamma_q \times DFT(f_q)$$

$$\text{where, } DFT(f_p) = NL(f_p) - \min(NL(f_p), NL(\bar{f}_p))$$

$$DFT(f_q) = NL(f_q) - \min(NL(f_q), NL(\bar{f}_q))$$

$WT(f_p, f_q)$ 은 i와 j번째 문자 사이의 가중치, $W_{ij}(f_p, f_q)$ 의 총합을 의미하며 DFT 함수는 주어진 함수의 positive와 negative 함수 사이의 문자 수 차이를 나타내는데, 이는 함수 f의 문자 수와 그의 보수인 f 의 문자 수와의 차이를 의미한다. γ_p 와 γ_q 는 각각 함수 f_p, f_q 에서 문자1개의 증가에 따라 전체 가중치에 영향을 주는 문자 규모 인수값을 나타내는데 이는 각 함수의 문자 수와 product term의 수에 따라 변하는 값이 된다. 그런데 본 논문에서 이미 각 함수에 대한 EF(f)를 계산할 때 필요한 규모 인수 α, β 을 적용하였기 때문에 $WT(f_p, f_q)$ 에 크게 영향을 주지 못하므로 실현적으로 상수 값 4.0을 적용하였다. 물론 γ_p 와 γ_q 에 매우 큰 값을 적용할 경우 오직 함수 f와 그의 보수 f 중에서 문자 수가 작은 경우만을 택하는 결과를 초래하였다.

IV. 다단 논리회로를 위한 Phase 할당

주어진 각 함수에 적당한 phase를 할당 함으로써 다단 논리합성에서 하나의 목적 함수인 전체 침면적을 줄이고자 한다. 주어진 부울회로에서 가능한 공통 부분 표현식이 많도록 각 함수의 phase를 결정함으로써 논리회로를 다단화 과정 수행 후 최소의 면적을 갖는 다단 논리회로 구조의 부울회로를 얻고자 한다.

다단 논리 최소화 단계에서는 주어진 다단 논리회로 구조인 부울 회로에서 불필요한 부분을 제거하는 단계이기 때문에 주어진 다단논리회로 구조의 구성에 따라 중대한 영향을 미치게 된다. 물론 부분적으로 collapsing하여 부분적으로 재구성화(restructuring)^[10] 시키지만 전체의 회로 구조들을 바꾸지는 못하므로 주어진 다단 논리회로 구조에 따라 결과에 미치는 영향은 대단히 크다.

1. 공통부분 표현식 그래프

본 논문에서는 가능한 공통 부분표현식이 많도록 phase를 할당하기 때문에 앞에서 정의한 가중치가 최대가 되도록 전체 함수의 phase를 결정하는 문제

로 귀착된다. 이 문제를 해결하기 위하여 주어진 positive 다출력 함수 $f = \{f_1, f_2, \dots, f_n\}$ 로부터 positive 와 negative를 포함한 모든 함수 $F = \{F_1, F_2, \dots, F_n\}$ 사이에서 가중치를 구함으로써 각 함수 사이의 공통 부분 정도를 나타내는 공통 부분표현식 그래프 G_F 다음과 같이 얻을 수 있다.

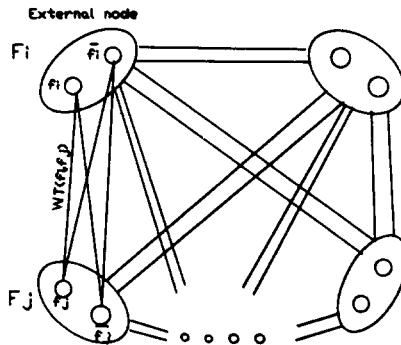


그림 2. 함수 $F = \{F_1, F_2, \dots, F_n\}$ 에 대한 공통 부분표현식 그래프 G_F

Fig. 2. Common subexpression graph G_F of $F = \{F_1, F_2, \dots, F_n\}$.

그림2에서 보는 바와 같이 각 외부노드 F_i 는 2개 phase의 내부 노드 f_i 와 \bar{f}_i (positive와 negative 함수)로 정의하고, 같은 외부 노드에 있는 내부 노드를 제외한 모든 노드와 예지가 연결되어 있고 각 예지에 가중치가 부여되어 있다. 외부노드 F_i 와 F_j 사이의 예지 수는 4개 이므로 가중치 그래프 G_F 에서 예지의 총수는 $4 \times n(n-1)/2$ 이다. G_F 에서 n개의 노드를 갖는 완전 그래프의 총수는 2^n 개 이므로 최대의 가중치를 갖는 G_F 의 서브 그래프를 찾기 위해서는 2^n 의 경우 수에 해당하므로 NP문제이다. 결국 phase 할당 문제는 모든 외부노드에서 오직 하나의 내부 노드를 포함하고 가중치의 합이 최대가 되는 완전 그래프를 찾는 문제가 된다.

2. Phase 할당 알고리듬

주어진 다출력 함수의 phase 할당 알고리듬의 전반적인 구성은 그림3과 같다. 그림3에서 먼저 $Weight_G(f)$ 는 주어진 다출력 positive 함수 ($f = \{f_1, f_2, \dots, f_n\}$)로부터 각 함수 사이의 공통 부분표현식 그래프 G_F 를 구성하는 함수이다.

```

Phase_assignment(f) {
    /*computation of weights between functions */
    G_F=Weight_G(f);
    while (all external nodes are not selected) {
        (x, y)=Phase_selection(G_F);
        update_function(f, x, y);
        update_graph((G_F, x, y));
    }
}

```

그림 3. 전반적인 phase 할당 알고리듬

Fig. 3. Overall phase assignment algorithm.

$Phase_selection(G_F)$ 는 공통부분 표현식 그래프 G_F 로부터 가중치의 합이 최대가 되며 n개의 내부노드를 포함하는 완전 그래프를 구하는 함수이다. 먼저 각 노드에서 가중치의 합이 최대인 n개의 노드를 선택하고, 선택된 n개의 노드로 구성된 서브 그래프에서 이미 선택되지 않은 최대값의 예지를 선택하여 그 노드에 해당하는 phase를 할당하는 알고리듬으로 그림 4와 같다.

```

Phase_selection(G_F) {
    for(each internal node i in G_F)
        for (each external node j in G_F)
            weight_sum(f_i) += max(WT(f_i, f_j), WT(f_i, \bar{f}_j));
    NList=nsort_list(weight_sum);/*sort n function lists
                                in decreasing order */
    do
        (x, y)=max_edge(NList);
        if while(phase_assigned(x)&& phase_assigned(y));
    return(x, y);
}

```

그림 4. Phase 선택 알고리듬

Fig. 4. Phase selection algorithm.

위의 알고리듬에서 각 내부 노드 f_i 에 대해서

$$weight_sum(f_i) = \sum_{j=1}^n \max(WT(f_i, f_j), WT(f_i, \bar{f}_j))$$

를 수행하는데 $\max(WT(f_i, f_j), WT(f_i, \bar{f}_j))$ 를 합하는 이유는 내부 노드 f_i 에 대해서 한 외부 노드 F_j 의 f_j , \bar{f}_j 중 오직 하나의 phase만이 실제로 필요하기 때문에 가중치의 최대값을 선택하였다. 또한 함수 $nsort_list$

()에서 내림 차순으로 배열하여 n 개의 노드를 갖는 서브 그래프를 구성하는데 각 노드의 가중치의 합으로 n 개를 선택한 이유는 총체적으로 문제를 해결하고자 하였고 n 개의 완전 그래프를 구하기 위함이다. 구해진 서브 그래프로 부터 함수 `max_edge()`를 거쳐 이미 할당되지 않은 노드를 포함하는 최대치의 예지를 얻게 됨으로 하나의 `phase_selection()` 함수가 마치게 된다.

그림3에서 함수 `update_function()`은 선택된 예지에 대응하는 노드의 phase로 주어진 함수를 수정하고 함수 `update_graph()`는 한 외부 노드에서 이미 할당된 내부 노드에 대응하는 다른 내부 노드를 제거하는 함수이다.

이와 같은 반복된 작업이 모든 외부 노드의 수 만큼 선택될 때까지 되풀이 함으로써 알고리듬을 마치게 된다.

3. 예제

주 출력의 갯수가 3인 benchmark 데이터 RD53에 대한 phase 할당의 간단한 예이다. 주어진 positive

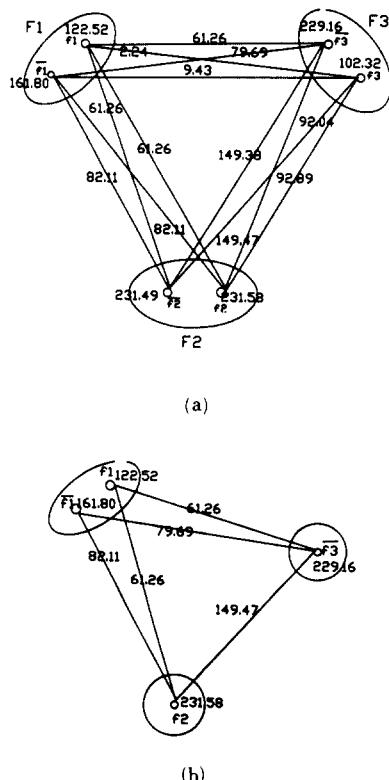


그림 5. RD53에 대한 공통 부분표현식 그래프
Fig. 5. Common subexpression graph of RD53.

함수 $f = \{f_1, f_2, f_3\}$ 로 부터 `Weight_G(f)`를 수행하여 얻어진 공통 부분표현식 그래프 G_F 는 그림5와 같다.

먼저 주어진 다출력 함수에 대한 가중치를 구하여 얻어진 공통 부분표현식 그래프 G_F 로 부터 각 f_i 노드에서 모든 f_j 에 대한 $\max(WT(f_i, f_j), WT(f_j, f_i))$ 의 합을 구한다. 그 값은 그림5에서 각 노드 옆에 나타내었다. 얻어진 가중치의 합에 대해서 함수 `nsort_list()`를 수행한 결과 외부 노드의 갯수만큼 내림 차순으로 배열한 함수 리스트를 얻는데 그 리스트는 (f_2, f_2, f_3) 이다. `Phase_selection()`에서는 (f_2, f_2, f_3) 의 노드로 구성된 G_F 의 서브 그래프으로 부터 149.47 값을 갖는 최대 예지 (f_2, f_3) 를 선택하여 이에 대한 phase를 할당하고, `update_function()`에서는 할당된 phase에 따라 positive 또는 그의 complement 함수 값을 구하여 주어진 함수를 대체 시킨다. 선택된 외부 노드에서 할당된 노드에 대응되는 노드를 가중치 그래프 G_F 로부터 제거시키게 되는데 `update_graph()`에서 이를 수행하고 그 결과는 그림 5(b)에 나타내고 있다. 이와 같은 작업이 외부 노드의 갯수 만큼 할당 될 때까지 반복 작업이 수행된다. 마찬가지로 그림 5(b)로 부터 배열된 함수 리스트 (f_2, f_3, f_1) 가 얻어지고 최대값의 예지 (f_1, f_2) 가 선택되어 이미 할당되지 않은 외부 노드 F_1 에서 f_1 를 할당함으로써 최종 phase 할당 결과인 (f_1, f_2, f_3) 를 얻는다.

V. 실험 및 고찰

본 논문에서 제안한 출력 phase 할당 알고리듬의 효율성을 입증하기 위해서 MCNC benchmark 데이터에 대한 실험 결과를 표1에 보인다. 먼저 주어진 부울회로에서 각 노드에 phase를 할당한 후, 문자 수를 표1의 2번째 행에 보였고, 3번째 행에서는 phase 할당에 소요된 CPU 실행 시간을 나타내고 있으며, phase가 할당된 함수에 대하여 회로 단순화 후, 문자 수는 4번째 행에 나타내고 있다. 얻어진 다단 논리회로에 대한 다단 논리 최소화의 결과는 5번째 행에 나타냈는데 다단 논리 최소화 알고리듬은 BOLD 시스템의 ESPRESSO_MLT를 이용한 결과와, BOLD와 MIS를 이용한 결과이다.

표1에서 보는 바와 같이 BOLD를 이용한 결과에서 출력 phase를 할당한 결과 면적 이용률을 문자수로 비교할 때 약 16.87%가 감소하였으며 BOLD와 MIS를 이용한 결과와 비교해 볼 때, 출력 phase를 할당한 결과가 12.37% 감소함을 나타내어 출력 phase를 할당한 결과가 전체 면적 이용률에서 매우 효율적임을 나타내고 있다. 본 알고리듬은 C언어로 SUN 4 SPARC 스테이션 상에서 실행한 결과이다.

표 1. MCNC benchmark 데이터에 대한 결과 비교표

Table 1. The comparison of results for MCNC benchmark data.

| | phase 할당 후 초기 문자수 | Output phase 할당 | | | No output phase 할당* | | | |
|-------|----------------------|-----------------|-------------------|--------------------------|---------------------|-----|------|---------------|
| | | CPU run time | 회로 다단화 수행후 문자수 | 논리 최소화후 BOLD/MIS&BOLD | YSC | MIS | BOLD | MI S& BOLD |
| 5XP1 | 299 | 6.5s | 129 | 86/83 | 113 | 129 | 93 | 91 |
| 9SYM | 544 | 4.3s | 82 | 60/60 | 236 | 260 | 209 | 223 |
| BW | 327 | 24.2s | 199 | 159/143 | 203 | 209 | 169 | 156 |
| CLIP | 751 | 20.7s | 203 | 117/116 | — | — | — | — |
| RD53 | 140 | 0.9s | 58 | 34/33 | 39 | 40 | 36 | 37 |
| RD73 | 876 | 11.6s | 133 | 77/75 | 115 | 143 | 81 | 86 |
| RD84 | 2342 | 56.3s | 480 | 159/134 | 136 | 268 | 159 | 134 |
| SAO2 | 393 | 6.5s | 200 | 148/143 | 158 | 150 | 154 | 131 |
| DUKE2 | 1686 | 521.1s | 520 | 397/378 | 477 | 478 | 360 | — |
| F51M | 325 | 3.6s | 137 | 98/— | — | — | 102 | — |

(※ : 참고문헌[3]의 데이터 참조)

VI. 결 론

본 논문에서는 회로 다단화 과정 전단계에서 면적 최소화를 위한 출력 phase 할당 알고리듬을 제안하였다.

주어진 함수 표현식의 문자 사이의 관계정도를 나타내는 평가 함수 $EF(f)$ 를 정의 하였고, 함수 사이의 공통 부분 정도를 나타내는 가중치에 따라 공통 부분표현식 그래프를 생성하였다. 생성된 그래프로부터 가중치의 합이 최대가 되도록 출력 phase를 할당하는 휴리스틱 알고리듬을 제안하였고 여러 실험 결과 본 논문에서 제안한 알고리듬이 면적 이용면에서 효율적임을 보였다.

앞으로의 연구 과제로는 회로 다단화 과정중에 생성되는 중간 변수에 까지 phase 할당 범위를 확장하는 연구가 요구된다.

參 考 文 獻

- [1] G. De Micheli, A. Sangiovanni-Vincentelli, "Design systems for VLSI circuits logic synthesis and silicon compilation," *Martinus Nijhoff Publishers*, 1987.
- [2] R. Rudell, R.K. Brayton, A. Sangiovanni-Vincentelli and A. Wang, "MIS: A Multilevel Logic Optimization System," *IEEE Transactions on Computer Aided Design*, vol. CAD-6, pp. 1062-1081, Nov. 1987.
- [3] D.G. Bostick, G.D. Hachtel, R.M. Jacoby, M.R. Lightner, P.H. Moceyunas, C.R. Morrison, and D. Ravenscroft, "The Boulder Optimal Logic Design System," in Proc. *IEEE International Conf. on Computer-Aided Design*, 1987.
- [4] C-L. Wey, T-Y. Chang, "An Efficient Output Phase Assignment for PLA Minimization," *IEEE Transactions on Computer Aided Design*, vol. CAD-9, pp. 1-7, Jan. 1990.
- [5] T. Sasao, "Input Variable Assignment and Output Phase Optimization of PLA's," *IEEE Transactions on Computer*, vol. C-33, pp. 879-894, 1984.
- [6] C-L Wey, S-M Chang, J-Y Jou, "OPAM: An Efficient Output Phase Assignment for Multilevel Logic Minimization," in Proc. *IEEE International Conf. on Computer Design*, pp. 270-273, 1989.
- [7] Melvin A. Breuer, "Design Automation of Digital System," *Prentice-Hall, Inc.*, 1972.
- [8] R.K. Brayton and C.T. McMullen, "The decomposition and factorization of boolean expressions," in *Proc. Int. Symp. on Circuits and Systems*, pp. 49-54, 1982.
- [9] R.K. Brayton, G.D. Hachtel, C. McMullen, and A. Sangiovanni-Vincentelli, "Logic minimization algorithms for VLSI synthesis," *Boston: Kluwer Academic*, 1984.
- [10] R.K. Brayton, A. Sangiovanni-Vincentelli, K. Singh, A. Wang, "Timing Optimization of Combinational Logic," in *Proc., IEEE International Conf. on Computer-Aided Design*, Nov. 1988.

著者紹介



李 載 興(正會員)

1959年 12月 28日生. 1983年 한
양대학교 전자공학과 졸업. 1985
年 2月 한양대학교 대학원 전자
공학과 졸업 공학석사 학위취득.
1991年 2月 한양대학교 대학원
전자공학과 박사과정 수료. 1989
年 7月~현재 대전공업대학 전자계산학과 조교수
재직중. 주관심분야는 VLSI CAD 특히 Logic syn-
thesis 등임.



鄭 正 和(正會員)

1950年 3月 10日生. 1975年 한
양대학교 전자공학과 졸업. 1981
年 3月 일본 와세다대학 박사학
위 취득. 일본 NEC(주) 중앙연
구소 연구원. KIET 위촉 연구원.
University of California, Berkely
교환교수. 주관심분야는 VLSI CAD 특히 High-
level Synthesis, Logic synthesis, Layout 등임.