

회로의 대칭성을 이용한 다단계 논리회로 회로에서의 전력 최소화 기법

(Power Minimization Techniques for Logic Circuits Utilizing Circuit Symmetries)

정기석[†] 김태환^{††}
(Ki-Seok Chung) (Taewhan Kim)

요약 논리회로 합성에서 함수의 대칭성을 이용하여 면적이나 시간 지연을 최소화하는 문제는 많은 시간동안 연구되어 왔다. 본 논문은 최근 들어 면적이나 시간지연 보다도 더 중요하게 여겨지는 전력 소모를 최소화하는데, 회로 대칭성이 어떻게 이용되는 지에 대한 연구를 소개한다. 이 논문에서 회로의 대칭성에 대한 폭넓은 정의를 소개하고, 각 대칭성 간의 관계에 대해 논의하며, 각 회로의 대칭성이 어떻게 전력을 줄이는데 유용할 수 있는지에 대해 논의한다. 또한, 회로에 존재하는 주 입력(primary input)과 내부 노드사이에 존재하는 대칭성을 찾아내는 알고리즘을 소개한다. 이 논문에서 소개하는 알고리즘의 특징은 첫째, 면적이나 속도지연의 증가가 거의 없이, 전력 소모를 줄여주는 효과적인 재합성 기법이란 것이다. 둘째, 대부분의 다른 휴리스틱(heuristic) 알고리즘과는 달리, 회로의 스위칭 (switching) 양에 있어 단조 향상(monotonic improvement)을 보장한다. 이미 잘 알려진 바와 같이 CMOS 회로에서는 스위칭 양이 전력소모에 대부분을 차지하므로, 알고리즘의 적용 후에 회로가 전력 소모 면에서 계속적인 향상을 이룰 수 있게 한다는 점에서 매우 효과적이라 하겠다. 알고리즘의 효과를 검증하기 위해서, MCNC 벤치마크 회로를 이용하여 실험을 시행하였고, 실험 결과, 속도나 면적에 대한 오버헤드가 거의 없으면서 평균 12%의 전력 소모를 줄일 수 있었다.

키워드 : CAD, 논리 회로 설계, 저전력 설계

Abstract The property of circuit symmetry has long been applied to the problem of minimizing the area and timing of multi-level logic circuits. In this paper, we focus on another important design objective, power minimization, utilizing circuit symmetries. First, we analyze and establish the relationship between several types of circuit symmetry and their applicability to reducing power consumption of the circuit, proposing a set of re-synthesis techniques utilizing the symmetries. We derive an algorithm for detecting the symmetries (among the internal signals as well as the primary inputs) on a given circuit implementation. We then propose effective transformation algorithms to minimize power consumption using the symmetry information detected from the circuit. Unlike many other approaches, our transformation algorithm guarantees monotonic improvement in terms of switching activities, which is practically useful in that user can check the intermediate re-synthesized designs in terms of the degree of changes of power, area, timing, and the circuit structure. We have carried out experiments on MCNC benchmark circuits to demonstrate the effectiveness of our algorithm. On average we reduced the power consumption of circuits by 12% with relatively little increase of area and timing.

Key words : CAD, logic circuit design, low power design

1. 서론

[†] 비회원 : 홍익대학교 전산학과 교수
kchung@cs.hongik.ac.kr
^{††} 종신회원 : 한국과학기술원 전자전산학과 교수
tkim@cs.kaist.ac.kr
논문접수 : 2003년 2월 19일
심사완료 : 2003년 6월 19일

VLSI 설계에 있어 속도와 면적을 최소화하기 위한 연구는 오랜 동안 계속되었지만, 휴대 장치의 일반화와 빠른 클럭(clock) 속도는 전력 소모의 최소화 문제를 속도나 면적을 최소화하는 문제 이상으로 중요한 설계의 목적의 하나로 만들었다. 반도체 기술의 급속한 발전은 설계안에 엄청난 수의 트랜지스터의 집적을 가능케 하

였고, 이는 곧, 설계의 복잡도가 빠르게 증가한다는 것을 암시한다. 더불어, 현대의 VLSI의 설계의 특징은 매우 빠른, 설계에서 제품까지의 제한 시간(fast time-to-market)으로 볼 수 있으며, 복잡한 설계의 검증 등을 위한 노력을 줄이기 위해, 기존의 설계를 특화된 목적을 위해 점진적 재합성하는 것이 점차 일반화되어 가고 있다. 예를 들어, 하나의 마이크로프로세서가 개발되면, 이 프로세서를 노트북에 사용하기 위해 기본적인 설계는 유지하면서, 설계를 점진적으로 재설계하여, 성능과 전력 소모 면에서 trade-off를 구하는 경우가 일반적이다. 이 논문에서는 바로 이런 점진적인 재설계를 위한 알고리즘을 제시한다. 기본적으로 이 논문은 회로 대칭성(circuit symmetry)을 이용하여, 이미 속도나 면적을 최소화한 회로를 입력으로 받아 들여, 속도나 면적 면에서의 최적화를 크게 훼손하지 않으면서 전력 소모를 최소화하는 방법을 제시한다. 회로의 대칭성을 이용하여 회로를 최적화하려고 할 때, 우리가 해결해야 할 문제는 크게 두 가지로 요약될 수 있다. 첫째는 어떻게 주어진 회로에서 대칭성을 찾을 수 있는가의 문제이고, 둘째는 어떻게 발견된 대칭성을 전력 소모를 줄이는 데 이용할 수 있는가의 문제이다. 이 논문에서는 바로 이 두 문제를 위한 알고리즘들을 제시한다.

논리회로에서, 전체 대칭성(total symmetry)과 부분 대칭성(partial symmetry)을 발견하는 문제는 1950년대 초부터 연구되어 왔다[1,2,3]. 논리함수의 전체 대칭성은 polynomial time 알고리즘이 존재한다. 하지만, 부분 대칭성은 발견하기 훨씬 어려운 문제이다. 현실적으로 가능한 모든 경우를 다 따져보는 것은 불가능하기 때문에, 많은 휴리스틱 알고리즘이 제안되었다[4,5,6]. [4]에서는 ROBDD(Reduced Ordered Binary Decision Diagram)를 이용하여 대칭성을 찾는 알고리즘을 제안하였다. 그들은 전처리 단계에서 비대칭성(asymmetry)을 발견함으로써, 대칭성의 과정을 단순하게 하는 알고리즘을 제시했다. [5]에서는 BDD(binary decision diagram)에서 변수들의 순서를 동적으로 재 정렬하여 대칭성을 갖는 변수들이 가능하면 가깝게 배치되면 쉽게 대칭성이 발견된다는 관찰에 의거한 효과적인 알고리즘을 제안하였다. [6]에서는 대칭성을 발견하기 위해 일반화된 Reed-Muller 폼(form)을 이용하였다.

이 논문에서는 과거의 관련 논문들과는 달리 주어진 회로에서 대칭성을 발견하려고 했다. 과거의 논문들이 주 입력 변수(primary input)들 사이의 대칭성을 발견하려는 것과는 달리, 이 논문에서는 입력 변수간의 대칭성은 물론, 회로내부의 신호선들 간 또는 내부 신호선과 입력변수간의 대칭성을 찾는 방법을 제시하였다. 다시 말해, 과거의 논문들이 회로가 정확히 논리게이트에 매

핑(mapping) 되지 않은 technology independent한 수준에서 대칭성을 찾으려고 했다면, 이 논문에서는 technology mapping이 끝난 후에 대칭성을 찾으려고 했다는 점이 다르다고 할 수 있다. 또한, 이 대칭성을 발견한 후, 각 대칭성을 어떤 재합성 방법을 사용하여 전력 소모를 줄일 수 있는지에 대한 방법을 제시한 점도 이 논문의 중요한 기여라고 할 수 있다.

2. 대칭성의 정의

어떤 Bool 함수 $f(x_1, x_2, \dots, x_n)$ 의 x_i 에 대한 cofactor는 $f_{x_i} = f(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n)$ 이고, $f_{\bar{x}_i}$ 에 대한 cofactor는 $f_{\bar{x}_i} = f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n)$ 이다. 모든 n -변수 함수 $f(x_1, x_2, \dots, x_n)$ 에 대한 Shannon expansion은 다음과 같이 정의된다.

$$f(x_1, x_2, \dots, x_n) = x_i f_{x_i} + \bar{x}_i f_{\bar{x}_i} \quad (1)$$

$f(x_1, x_2, \dots, x_n)$ 는 $f_{x_i} \neq f_{\bar{x}_i}$ 인 경우일 때, 또 그 때에만 x_i 에 의존한다고 말한다. Shannon expansion은 여러 개의 변수에 대해서 일반화 될 수 있으며, 만일 두 변수 x_i 와 x_j 에 대한 Shannon Expansion을 구하면 다음과 같다.

$$f(x_1, x_2, \dots, x_n) = x_i x_j f_{x_i x_j} + x_i \bar{x}_j f_{x_i \bar{x}_j} + \bar{x}_i x_j f_{\bar{x}_i x_j} + \bar{x}_i \bar{x}_j f_{\bar{x}_i \bar{x}_j} \quad (2)$$

어떤 함수 $f(x_1, x_2, \dots, x_n)$ 의 값이 두 변수 $\{x_i, x_j\}$ (또는 $\{x_i, \bar{x}_j\}$)에 대해서 만일 두 변수를 서로 맞바꾸어도 함수의 기능이 변하지 않을 때, 이 함수는 이 두 변수에 대해 대칭이라고 한다. 이 두 변수 쌍 $\{x_i, x_j\}$ 을 대칭쌍(symmetry pair)이라고 한다. 어떤 함수 f 의 변수의 집합을 X 라 할 때, 어떤 부분 집합 $Y \subseteq X$ 에 대해, Y 의 어떤 순열(permutation)에 대해서도 함수의 기능이 변하지 않을 때, 이 때 Y 를 f 의 대칭 부분집합(symmetry subset)이라고 한다. [7]에서는 대칭함수의 개념을 다섯 종류의 대칭성으로 확장하여 정의하였다. 하지만, 이 논문에서는 그 중 몇 가지는 실질적으로 같은 종류의 대칭성이라 판단하여, 세 종류의 대칭성으로 재분류하여 논의한다.

2.1 비등가(non-equivalence)와 등가(equivalence) 대칭

어떤 함수 f 가 두 변수 x_i 와 x_j 에 대해서 비등가 대칭일 필요충분조건은 $f_{x_i \bar{x}_j} = f_{\bar{x}_i x_j}$ 이다. 이 것은 위에 언급한 함수 대칭의 원래 개념과 일치한다. 즉, 이 논문에서 별도의 언급이 없이 함수의 대칭을 언급할 때, 이는 비등가 대칭을 의미한다. 어떤 함수 f 가 두 변수 x_i 와

x_j 에 대해서 등가 대칭일 필요충분조건은 $f_{x_i} = f_{\bar{x}_i}$ 이다. 예를 들어, $r(x, y, z) = x\bar{y}z + xy + \bar{x}yz$ 는 $r_{x\bar{y}} = r_{\bar{x}y} = z$ 이기 때문에 x 와 y 에 대해 비등가 대칭이고, $g(x, y, z) = x\bar{y}z + xy + \bar{x}yz + \bar{x}y$ 는 $g_{xy} = g_{\bar{x}y} = 1$ 이라서 x 와 y 에 대해서 등가 대칭이다. 이 논문에서는 함수 f 가 두 변수 x, y 에 대해 (비등가) 대칭일 때, 이를 $f \subset S_{x,y}$ 로 나타내기로 한다. 어떤 함수가 두 변수 x, y 에 대해서 비등가 대칭일 때, 이는 x, \bar{y} (또는 \bar{x}, y)에 대한 등가 대칭이란 말과 일치하기 때문에 이 논문에서는 비등가 대칭만을 고려한다.

2.2 다중 형태 대칭(multi-form symmetry)

어떤 함수 f 가 두 변수 x, y 에 대하여 다중 형태 대칭일 필요충분조건은 이 함수가 두 변수에 대해, 비등가 대칭인 동시에 등가 대칭일 때이다. 예를 들어, 함수 $g(x, y, z) = x\bar{y} + xy + \bar{x}yz + \bar{x}y$ 는 $g_{xy} = g_{\bar{x}y} = 1$ 이고 $g_{\bar{x}y} = g_{xy} = z$ 라서 x 와 y 에 대해서 다중 형태의 대칭이다. 함수 f 가 변수 x, y 에 대해 다중 형태 대칭일 때, 이를 $f \subset M_{x,y}$ 로 표시한다.

2.3 단일 변수 대칭(single-variable symmetry)

어떤 함수가 변수 $x=1$ 인 조건에서 변수 y 에 대해 단일 변수 대칭일 필요충분조건은 $f_{x\bar{y}} = f_{xy}$ 이다. 비슷하게, 어떤 함수가 변수 $x=0$ 인 조건하에서 변수 y 에 대해 단일 변수 대칭일 필요충분조건은 $f_{\bar{x}\bar{y}} = f_{\bar{x}y}$ 이다. 함수 f 가 $x=1(x=0)$ 인 조건하에서 변수 y 에 대해 단일 변수 대칭일 때, 이를 $f \subset V_{x \rightarrow y}(V_{\bar{x} \rightarrow y})$ 로 표시한다. 예를 들어, $h(x, y, z) = x\bar{y}z + xy$ 는 $h_{x\bar{y}} = h_{xy} = 0$ 이기 때문에 $x=0$ 하에서 변수 y 에 대하여 단일 변수 대칭이다.

3. 대칭성을 이용한 전력 소모 최소화 기법

이 절에서 먼저 어떻게 회로의 대칭성이 전력 소모를 줄이는데 사용될 수 있는지 소개한다. 이 절에서 소개되는 기법들의 큰 특징은 대부분 면적에 대한 오버헤드는 거의 없고, 속도 지연의 증가를 최소화하면서, 전력소모를 동시에 줄이는 기법들이란 점이다.

3.1 면적의 추가 부담이 없는 재합성 기법: 연결을 맞추기

대칭 부분집합의 정의에 의하면, 함수는 대칭 부분 집합의 변수들의 어떤 순열(permutation)에 대해서도 함수의 기능은 변하지 않는다. 이를 회로의 구현 면에서 보면, 변수들의 순열을 바꾼다는 것은 변수들의 서로간의 연결을 맞바꾼다는 것과 일치한다. 이렇게 서로 맞바꿈으로써, 기능을 변화시키지 않으면서도 전력 소모 면에서 더 좋은 회로를 구현할 수 있다. 또한, 추가의 게

이트를 사용하지 않으므로, 면적의 증가¹⁾는 없다고 볼 수 있다. 예로서, 그림 1-(a)의 회로는 $g \subset S_{e,i}$ 이므로 그림 1-(b)의 회로와 서로 등가이다.

다중 형태의 대칭은 비등가 대칭을 함축하므로, 함수가 어떤 회로의 두 연결 s 와 t 에 대해 다중 형태의 대칭이라면, s 와 t 의 연결을 서로 맞바꿀 수 있다. 더욱이, 다중형태 대칭은 등가 대칭도 함축하므로 회로에 존재한다면 \bar{s} (not s)와 t , 또는 s 와 \bar{t} 의 연결을 맞바꾸어도 회로의 기능은 변하지 않는다. 그림 2에 다중 대칭에 대한 예를 보였다. 회로는 연결 c, e 에 대하여 다중 형태 대칭이므로 세 회로는 모두 등가이다.

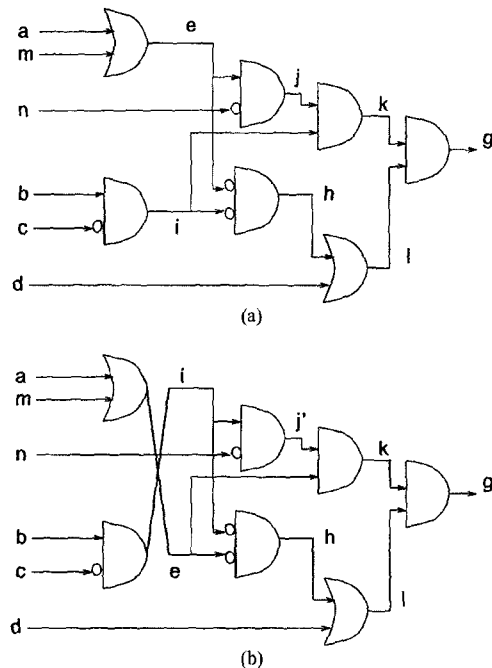


그림 1 (a) 대칭성 $g \subset S_{e,i}$ 과 (b) 연결을 맞바꾸었을 때의 등가 회로

3.2 추가 게이트를 사용하는 재합성 기법: 구조적 변형

우리가 추가의 게이트의 사용을 허용한다면, 다른 종류의 회로 대칭성을 이용해서 더 효과적으로 전력소모를 줄일 수 있다. 단, 추가의 게이트의 사용은 반드시 면적이 늘어나도록 하는 것은 아니며, 경우에 따라 줄어들 수 있다는 것을 유의할 필요가 있다. 앞으로의 설명에 있어 매핑에 사용되는 논리 게이트의 fan-in은 2 이하라고 가정한다. 이는 설명을 간단히 하기 위한 것이

1) 이 논문에서의 면적은 게이트의 수와 신호선의 숫자로 측정된다고 가정한다.

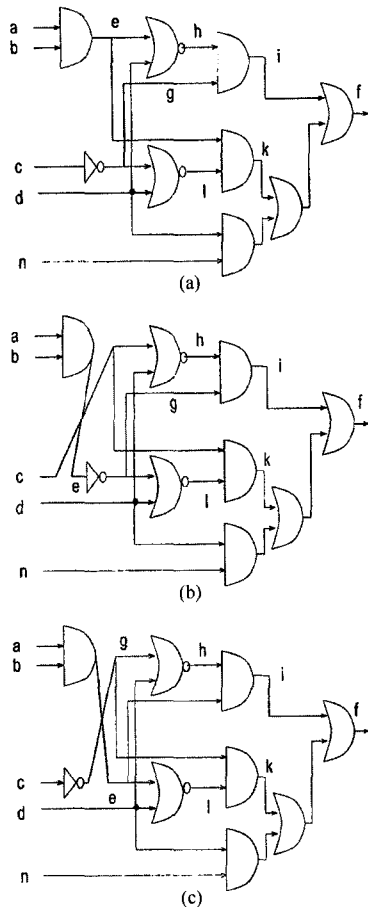


그림 2 (a) 다중 형태 대칭성 $f \subset M_{c,e}$ 과 (b) 비등가 대칭을 이용한 연결을 맞추었을 때의 등가 회로 (c) 등가 대칭을 이용하여 연결을 맞추었을 때의 등가회로

며, 다중 fan-in에 대한 일반화는 간단히 이루어질 수 있다.

3.2.1 다중 형태의 대칭성을 이용한 기법

만일 함수 f 가 두 변수 s 와 t 에 대해 다중 형태 대칭이라고 하면, 어떤 클럭 사이클에서, $s=t=0(s=t=1)$ 이고 다음 클럭 사이클에서 $s=t=1(s=t=0)$ 이라고 하면, $f_{st}=f_{\bar{s}\bar{t}}$ 이기 때문에 함수의 값은 변하지 않는다. 즉, s 와 t 에서의 상태 전이는 불필요한 것이다. 비슷하게, 어떤 클럭 사이클에서 $s=0$ 이고 $t=1$ 이었다가 다음 사이클에서 $s=1$ 이고 $t=0$ 이면 $f_{st}=f_{\bar{s}\bar{t}}$ 이기 때문에 s 와 t 에서의 상태 전이에 상관없이 f 의 상태는 변하지 않는다. 결국, 전력 소모의 감소는 바로 이런 불필요한 상태 전이를 최대한 없애 줌으로써 이룩된다. s 와 t 에 대해 f 가 다중 형태 대칭이라면, 불필요한 상태 전이를 줄이기 위

해서는 s 와 t 의 XOR(exclusive-OR)를 계산하여, s 와 t 가 같은 값을 갖느냐 다른 값을 갖느냐에 따라서만 f 에 영향을 주도록 하면 된다. 즉, 재합성 기법은 다음과 같이 기술될 수 있다.

(1) t 의 위치에 Vdd(상수 신호 1)이나 GND(상수 신호 0)을 연결한다. 0과 1의 선택은 t 의 fanout에 있는 게이트의 타입에 따라 결정한다.

(2) t 에 0(1)을 연결했으면, 2-입력 XOR(XNOR) 게이트를 삽입하고, s 와 t 를 XOR(XNOR)의 입력으로 연결하라. 그리고 XOR(XNOR)의 출력을 원래의 s 의 fanout에 연결하라.

그림 3(a)에서는 상수 0이 t 에 연결되고, 따라서 XOR가 재합성에 쓰인 경우를 보여준다. 여기서 주의 깊게 봐야할 것은, 이 변형에 있어 면적의 증가는 기껏해야 XOR게이트(g_0) 하나이나, t 에 상수 신호를 연결함으로써, 원래 t 의 fanout에 있던 두 게이트(g_1, g_2)가 변형 후에 없어짐을 알 수 있다.

3.2.2 단일 변수 대칭

함수 f 가 $s=1$ 인 조건하에서 변수 t 에 대한 단일 변수 대칭이 존재하면, $f_{st}=f_{\bar{s}t}$ 이다. 이는 다시말해 $s=1$ 이면, 함수 f 는 t 의 값에 의존하지 않는다는 말이다. 즉 이를 전력 소모를 줄이는 측면에서 얘기하면, 단일 변수 대칭이 존재하면, t 의 값의 변화, 즉 상태 전이에 따른 fanout에서의 상태전이를 없애 줌으로써, t 의 상태 변화에 따른 t 의 transitive fanout에서의 전력 소모를 줄일 수 있다는 말이다. 이는 그림 3-(b)에서 보여준 바와 같이 게이트 하나를 더함으로써 구현될 수 있다. 더해지는 게이트의 종류는 $V_{s,t}$ 의 경우 OR게이트이고, $V_{\bar{s},t}$ 의 경우 AND게이트이다. $s=1(s=0)$ 일 때, OR(AND)게이트를 사용함으로써, t 의 상태 전이가 차단되는 것을 쉽게 이해할 수 있다.

3.2.3 단일 변수 대칭과 비등가 대칭이 동시에 존재

함수 f 가 단일 변수 대칭과 비등가 대칭이 s 와 t 사이에 동시에 존재하면, 이는 $f_{st}=f_{\bar{s}t}=f_{s\bar{t}}$ 를 의미한다. 이는 s 와 t 의 값이 연속된 클럭 사이클에 걸쳐 (1,1), (0,1) 또는 (1,0)인 상태로 있는 한 f 의 값이 변하지 않는다는 말이며, 이는 s 와 t 값이 위 값들을 갖고면서 변하는 동안 생기는 상태 전이는 불필요한 것을 의미한다. 즉, 전력을 줄이기 위해서는 제거되어야 한다. 이를 제거하기 위해서는 s 와 t 의 값이 (0,0)인 경우와 나머지 조합의 경우만 구별해주면 된다. 이는 OR 게이트를 하나 추가함으로써 쉽게 구별된다. 그림 3-(c)를 보면, OR 게이트(g_0)가 더해지고, t 대신에 t 의 fanout에 상수 신호인 0(1)이 연결되면 된다. 0과 1의 선택은 s 와 t 의 값에 있어 (1,1), (0,1), (1,0)의 경우 어떤 값을 세 조합의 대

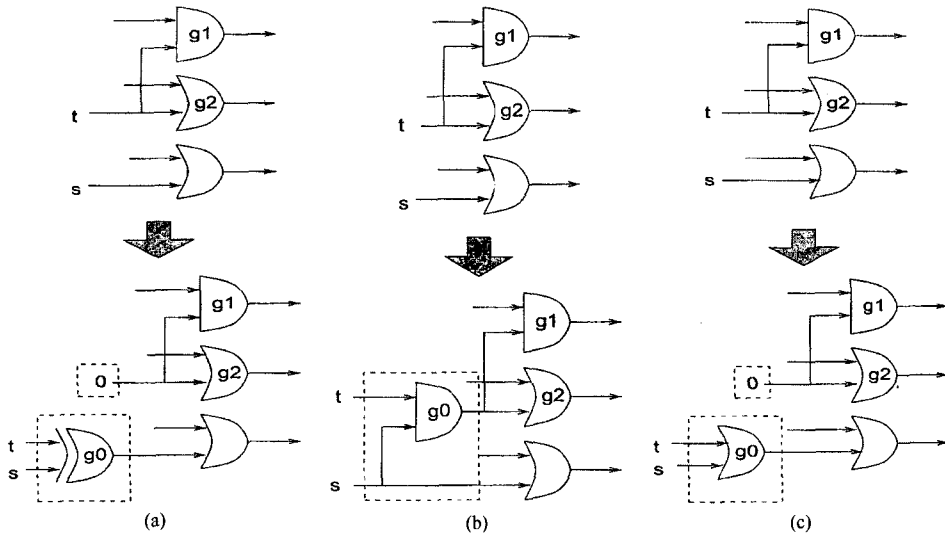


그림 3 대칭을 이용한 구조적 변형의 예: (a) 다중 형태 대칭성 $f_{M_{s,t}}$ 과 (b) 단일 변수 대칭성 (c) 등가 대칭을 이용하여 연결을 맞바꾸었을 때의 등가회로

표 값으로 쓰는가를 결정하는 것이며, t 의 fanout에 따라 결정된다. $f_{s,t} = f_{s,t} = f_{s,t}$ 의 경우에는 추가로 AND 게이트를 이용한다. 이 경우의 재합성기법에 대해 요약 하면:

- (1) t 대신에 상수 신호인 0(1)을 연결한다. 0과 1의 선택은 t 의 fanout의 게이트 타입에 따라 결정된다.
- (2) 2 입력 OR (AND) 게이트를 더하고, s 와 t 를 입력으로 만들어라. 원래 s 의 fanout이 이 OR(AND)게이트의 출력에 연결된다.

4. 대칭성을 발견하는 알고리즘

이 절에서는 회로에서 주 입력(primary input)을 포함한 모든 게이트의 입력선들 사이에서 존재하는 대칭성을 발견하는 알고리즘에 대해 설명한다. 이 알고리즘은 게이트의 타입이나 게이트의 입력 수에 상관없이 동작하나, 설명의 편의를 위하여, 여기서는 2입력 게이트만 고려한다.

먼저, $FI(v)$ 와 $FO(u)$ 를 각각 노드 v 의 fan-in과 u 의 fanout의 집합으로 표시한다. 그리고 $C(v)$ 를 노드 v 로부터 시작하는 논리 cone이 나타내는 함수라고 한다. 그리고 $TFI(v)$ 와 $TFO(u)$ 를 v 와 u 의 transitive fan-in과 transitive fanout을 각각 나타내며, 의미는 재귀적으로 다음과 같이 정의 된다.

$$TFI(v) = FI(v) \cup \left(\bigcup_{u \in FI(v)} TFI(u) \right) TFO(u) \quad (3)$$

$$= FO(u) \cup \left(\bigcup_{v \in FO(u)} TFO(v) \right)$$

또 $Level(v)$ 는 v 가 primary input일 경우에는 0이며, 주 입력이 아니면, 주 입력으로부터 v 까지의 최대 거리를 나타내며, 즉, $Level(v) = \max_{u \in FI(v)} \{Level(u) + 1\}$ 이다. 회로에서 대칭성을 발견하는 일은 매우 어려운 문제 이므로, 우리는 다음의 관찰에 의거한 휴리스틱 알고리즘을 제시한다.

관찰: f 가 단일 출력 함수라고 하자. 우리가 f 를 구현한 회로가 내부 선인 s 와 t 에 대해 대칭성이 있는가를 검사하려고 한다. 만일 s 와 t 로부터 주 출력(primary output)에 이르는 경로에 다중 fanout이 없다고 가정하자. 이때 어떤 내부 선 $c \in (TFO(s) \cap TFO(t))$ 에 대해서 c 가 s 와 t 에 대해서 대칭이라면(그림 4 참조) f 도 s 와 t 에 대해서 대칭이다.

이 관찰은 다음과 같은 의미를 내포한다. 먼저, 만일 $Level(c) \ll Level(f)$ 라면, c 에서 대칭성을 검사하는 것이 f 에서 검사하는 것보다 훨씬 용이하다. 둘째로, c 에서의 s 와 t 에 대한 대칭성은 C_s 와 C_t (그림 4에서 회색부분)에 의존하지 않는다. 세 번째로, $TFO(c)$ 에 속한 어떠한 신호선이나 cone C_c 의 범위 밖에 있는 어떠한 신호선에 대해서도, s 와 t 에 대한 대칭 재합성후에도 상태 전이의 양은 변하지 않는다. 반면, cone C_c 안의 신호선들의 활동영역은 재합성에 의해 변할 수 있으며, 이를 이용하면, cone안에서의 상태전이의 빈도수를 줄임으로써 전력 소모를 줄일 수 있다. 이러한 성질은 바로 대칭을 이용한 재합성 기법이 회로의 상태 전이를 최소화하는데, 단조 향상(monotonic improvement)을 할 수 있도록 가

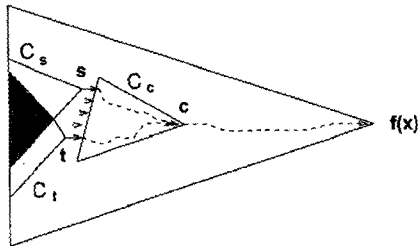


그림 4 회로상에서 대칭성을 발견하기 위해 중요한 관찰

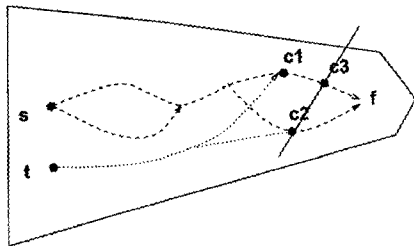


그림 5 다중 fanout을 갖는 회로에서의 대칭성의 발견

능하게 해준다. 즉, cone C_c 안에서 상태전이의 감소는 회로 전체로 보았을 때, cone 밖의 신호에 대한 상태 전이를 변하지 않게 함으로, 전체적으로 상태 전이의 빈도수가 감소됨을 의미한다고 하겠다.

위의 경우를 일반화할 때, 만일 s 와 t 로부터의 경로에 다중 출력(multiple fanout)과 재 수렴하는 출력(reconvergent fanout)이 존재한다면, $TFO(s) \cap TFO(t)$ 은 일반적으로 어떤 신호 선의 집합이 된다. 이 집합의 신호선들은 서로 독립적일 수 있다(그림 5 참조). 이 경우에 대칭성은 각 s 와 t 로부터의 각 fanout 경로에 있어 어떤 $c \in (TFO(s) \cap TFO(t))$ 가 있어 이 c 에 대해 s 와 t 가 대칭일 때 발견된다. 앞에서와 마찬가지로, c 는 s 와 t 로부터 가까울수록 좋다. 그림 5는 예를 보여준다. 만일 $c1$ 에서 s 와 t 에 대한 대칭이 아니라면, $c3$ 에서 대칭을 검사해보고, 만일 $c3$ 와 $c2$ 에서 s 와 t 에 대해 대칭이라면, 전체 회로도 s 와 t 에 대해 대칭이 된다.

이 논문의 대칭성 발견 알고리즘은 다음의 pseudo code에 의해 요약된다. 알고리즘은 먼저, 재합성을 시도하고자 하는 신호선 하나를 정하고, 그와 다른 신호선과의 세 가지 대칭성을 발견하려고 한다. 이때 다른 신호선은 t 와 서로 인과관계가 없는 독립적인 선이며, $Level(t)$ 와 크게 다르지 않은 신호선에 한 해서 대칭성이 검색된다. 레벨이 크게 다른 경우를 제외하는 이유는 레벨 차이가 클 경우, 재합성후의 시간 지연에 문제가 발생할 지 모르기 때문에 제외된다. 레벨 차이는 설계자가 정하는 패러미터이며, 여기서는 편의상 lev_max 로 표시한다. 두 신호선에 대한 대칭성을 검사하기 위한 공

통의 transitive fanout 신호선들의 집합을 구하고 이를 check point(CP)라고 부른다. 이 집합의 신호들은 서로 독립적이며, 되도록, 두 신호 s 와 t 에 가장 가까운 신호선들이다. 이 알고리즘에서 가장 시간 복잡도가 높은 부분은 check point와 s, t 등으로 이루어진 콘(cone)으로부터 세 가지 타입의 대칭성을 검사하는 부분으로, 구현된 대칭성 검사의 방법은 [4]에서 제시한 방법을 이용하였으며, [4]에 의하면 시간 복잡도는 $O(nG)$ 이다. 이 때, n 은 콘의 변수의 개수이며, G 는 콘을 ROBDD(Reduced Ordered Binary Decision Diagram)로 표현했을 때의 노드의 개수이나, 이 논문에서의 경우 콘의 크기를 작게 유지하기 때문에 빠른 실행시간을 보인다.

알고리즘 1: 신호선 t 에 대한 대칭성의 발견

```

/* loop 1 */  $t$ 와 독립적이면서, 레벨차이가 lev_max보다 작거나 같은 모든  $s$ 에 대해서  $s$ 와  $t$  사이에 check point(CP)를 구한다.
/* loop 2 */ 모든  $c \in CP$ 에 대해서  $c$ 와  $s, t$ 로 이루어진 최소한의 크기의 cone을 결정한다.  $c$ 에서 세가지 타입의 대칭성이  $s$ 와  $t$  사이에 존재하는지 검사
/* end of loop 2 */
/* end of loop 1 */
    
```

5. 재합성의 선택을 위한 알고리즘

하나의 회로에서 하나의 신호선 t 에 대해서 다양한 종류의 대칭성이 존재 할 수 있다. 대칭성이 발견되었을 때, 발견된 대칭성에 적용할 수 있는 재합성 기법을 다 적용해보고, 어떤 것이 좋은지를 선택하는 것은 현실적으로 불가능하다. 그리하여, 대칭성을 발견되었을 때, 적용할 수 있는 기법의 선택이 매우 중요하다. 이는 재합성을 하는 것이 좋은지 아닌지를 판단하는 것과 또 여러 가지의 재합성이 가능할 때, 어떤 재합성을 적용하는 것이 효과적인가에 대한 판단이 가능해야 한다는 것을 의미한다. 우리는 이 논문에서 재합성의 효과는 신호들의 상호 연관성(correlation)과 밀접한 관계가 있음을 밝힌다. 신호들의 상호연관성은 크게 시간적 연관성(temporal correlation)과 공간적 연관성(spatial correlation)으로 나뉘며, 주어진 회로에서 이를 분석해내는 기법은 여러 논문에서 소개된 바 있다. 이 논문에서는 [8]에서 제시된 방법을 이용하여 두 신호 x, y 의 시간적-공간적 상호 연관성(spatio-temporal correlation)을 계산한다. [8]에서의 시간적 공간적 연관성을 계산하는 총 알고리즘의 시간 복잡도는 $O(n^2NP)$ 이다. 여기서, n 은 변수의 개수, N 은 OBDD에서의 노드의 개수, P 는 OBDD의 경로(path)의 개수이다. OBDD의 크기가 최악의 경우

지수적으로 증가할 수 있으므로, polynomial time 알고리즘이라고 하기는 어려우나 대부분의 실제 경우에 있어 지수적으로 증가하는 경우는 드물기 때문에 본 논문에서의 실험에서는 매우 빠른 실행시간을 보였다.

O_{ab} 를 x 가 a 값을 가질 때, y 는 b 값을 갖게 될 확률(occurrence probability)라고 하자. 이는 엄밀한 의미의 공간적 연관성이라고 볼 수는 없으나, 상당히 spatial correlation에 근접한 확률 값을 갖는다. 또, $T_{ab,cd}$ 를 x 가 a 에서 c 로 값이 바뀔 때, 동시에 y 는 b 에서 d 로 값이 바뀌는 전이 확률(시간적 연관성의 근사치)이라고 하자. 이때 표 1은 이들 값과 재합성기법의 효율성에 대한 연관성을 요약해서 보여준다. 각 재합성 기법에 대해 ↑는 해당 값이 상대적으로 높게 나올 때 효과적이란 의미이고, 반대로 ↓는 낮을 때 효과적이란 의미이다. 재합성기법은 해당 대칭성과 깊은 관련이 있으므로, 표기의 단순화를 위하여 표에서는 대칭성의 타입을 표시하였다. 예를 들면, 어떤 회로가 신호서 x 와 y 에 대해 비등가 대칭일 때, 만일 x 는 어떤 AND게이트에, 또 y 는 어떤 OR게이트에 입력이라면, O_{10} 의 값이 높을 때, 두 연결을 맞바꾸는 재합성기법이 매우 효과적임을 의미한다.

표 1 재합성이 전력을 줄이는데 효과적이기 위한 바람직한 상관 관계

Correlation	Type of symmetry
$O_{01} \uparrow$ or $O_{10} \uparrow$	$S_{x,y}$
$O_{00} \uparrow$ or $O_{11} \uparrow$	$S_{x,\bar{y}}$ or $S_{\bar{x},y}$
$T_{00,11} \uparrow \wedge T_{01,10} \uparrow$	$M_{x,y}$
$O_{11} \downarrow$	$V_{\bar{x} \rightarrow y} \wedge S_{x,y}$ (or $V_{\bar{y} \rightarrow x} \wedge S_{x,y}$)
$O_{00} \downarrow$	$V_{x \rightarrow y} \wedge S_{x,y}$ (or $V_{y \rightarrow x} \wedge S_{x,y}$)
$O_{10} \downarrow$	$V_{y \rightarrow x} \wedge S_{x,\bar{y}}$ (or $V_{\bar{x} \rightarrow y} \wedge S_{x,\bar{y}}$)
$O_{01} \downarrow$	$V_{x \rightarrow y} \wedge S_{x,\bar{y}}$ (or $V_{\bar{y} \rightarrow x} \wedge S_{x,\bar{y}}$)
$T_{01,00} \uparrow$	$V_{\bar{x} \rightarrow y}$
$T_{00,10} \uparrow$	$V_{\bar{y} \rightarrow x}$
$T_{01,11} \uparrow$	$V_{y \rightarrow x}$
$T_{10,11} \uparrow$	$V_{x \rightarrow y}$

6. 실험 결과

회로의 대칭성을 발견하고, 그에 따른 재합성 기법을 적용하여 전력 소모를 줄이는 알고리즘은 C++ 프로그램으로 구현되었고, Linux를 탑재한 Pentium 4를 사용하여 MCNC 벤치마크 회로들을 실행하면서 테스트하였다. 본 논문은 대칭성을 이용한 재합성의 효과를 보이기 위한 것이므로, MCNC 벤치마크의 회로들 중, 실험 결과를 위해 사용된 회로들은, 이 논문의 주요 관심인 회로상의 대칭성이 비교적 많이 나타나는 회로들을 선택하였다. 과정은 SIS[9]을 이용하여, 회로를 2-입력 AND와 OR로 분해하고, mcnc.genlib을 이용하여, 면적과 속도를 최적화 목표로 하는 technology mapping을 수행하였다. 이 회로는 이 논문의 실험의 입력으로 사용되어, 면적과 속도 면에서의 성능의 희생 없이 전력 면에서 더 향상시키고자 하였다. 이를 위하여 다음 단계를 실행하였다. 먼저, 각 회로에서 상태 전이 활동이 가장 활발한 신호선 들을 구한다. 이는 SIS의 power_estimate과 power_print 명령어를 이용하여 구하였다. 그 다음 이 신호선 들과 회로상의 다른 신호선 들과 어떤 종류의 대칭성이 존재하는가를 제한된 알고리즘을 실행함으로써 구했다. 상태 전이가 활발한 선들은 가장 활발한 신호선들 중 서로 서로가 transitive fanout 이 아닌 신호선 들로 약 20%정도의 신호선 들을 고려하여, 대칭성을 발견했다. 또한 신호선들 간의 공간적 시간적 연관성(spatio-temporal correlation)을 구하였다.

발견된 대칭성들은 연관성들을 고려하여, 재합성되었고, 이 재합성된 회로는 다시 SIS에 의해 최적화되었다. 특히나 이 단계에서의 최적화는 재합성과정에서 상수 0(GND)과 1(Vdd)로 연결된 부분에 대한 최적화를 이루기 위해 SIS의 sweep과 eliminate명령을 사용하였다. 그리고 20MHz 클럭(clock)을 사용하여, 최종회로에 대해 전력소모 및 속도, 면적을 측정하였다.

표 2는 결과를 요약하고 있다. 열 #PI와 #PO는 각각

표 2 제시된 알고리즘에 의한 회로 재합성의 결과

Circuit	#PI	#PO	Size	Before			After			% Power
				Power	Delay	Area	Power	Delay	Area	
cm162a	14	5	83	203	16.7	78	182	13.7	78	10.7
pml	13	10	93	299.2	14.5	109	277	15	114	7.4
cu	14	11	94	357.7	13	109	321.3	11.6	114	10.2
cm152a	11	1	36	119	11.3	32	82.4	11	43	30.7
pcler8	27	17	94	224.1	27.7	165	200.1	33.5	144	10.7
i2	201	1	233	730.9	18	306	630.4	22.5	317	13.7
cm85a	11	3	76	123.7	21	72	100.3	26.7	84	18.9
9symml	9	1	243	1000.2	28.3	353	973.6	28.7	355	2.7
b9	41	17	219	514.0	16.8	191	394	16.7	198	7.7
cc	21	13	103	367.8	13.5	113	339.0	13.6	107	7.8
c8	28	17	315	924.6	18.1	375	822.5	18.8	378	11.0
avg.							-12%	+5.0%	+4.7%	

주 입력과 주 출력의 수를 나타내며, Size열은 기술적 분해(technology decomposition) 후의 게이트의 수를 나타낸다. Before와 After는 대칭성을 이용한 최적화가 적용되기 전과 후의 전력 소모, 면적, 속도를 측정된 것을 비교한 것이다. %power는 전력 소모의 감소분을 나타낸다. 이 실험에서 우리는 평균 12% 전력소모를 줄일 수 있었으며, 이에 따른 면적의 증가는 5%, 속도의 증가는 4.7%이었다. 이미 최적화된 회로로부터 12%의 향상을 보였다는 것은 상당히 주목할 만한 일로, 우리가 제시한 알고리즘이 면적과 속도의 희생은 거의 없이 상당한 양의 전력 소모를 줄일 수 있다는 것을 분명히 보여준다고 하겠다.

7. 결 론

대칭성을 이용하여 저전력 소모를 위한 재합성기법의 적용은 상당히 효과적으로 면적과 속도의 희생없이 빠른 시간 안에 전력을 줄일 수 있다는 것을 연구하였다. 많은 국지적 재합성 기법이 최적과는 거리가 있는 결과를 내거나 한 가지 면에서 설계를 향상시키려다, 다른 면의 설계의 질을 떨어뜨리는 문제점을 낳는다. 우리가 이 논문에서 제시한 기법은 기본적인 국지적 재합성기법의 단점을 완전히 극복하지는 못했다 할지라도, 이 논문에서 소개한 기법은 최소한의 오버헤드로 아주 효과적인 저전력 소모를 이룩할 수 있는 기법이다. 또한 이 논문의 재합성 기법은 재합성에 의한 상태전이의 변화 영역을 알 수 있기 때문에, 단조적인 향상을 보장한다. 이 모든 기법들은 실제적인 회로를 통하여, 시뮬레이션에 의해 검증되었다.

참 고 문 헌

- [1] S. R. Das and C. L. Sheng, "On detecting total and partial symmetry of switching functions," *IEEE Transactions on Computers*, 20, pp.352-255, 1971.
- [2] D. L. Dietmeyer and P. R. Shneider, "Identification of symmetry, redundancy and equivalence of Boolean functions," *IEEE Transactions on Electronic Computers*, EC-16, pp.804-817, 1967.
- [3] Z. Kohavi, *Switching and Finite Automata Theory*, 2nd edition, London, McGraw-Hill, 1978.
- [4] D. Möller, J. Mohnke, and M. Weber, "Detection of symmetry of Boolean functions represented by ROBDDs," *Proceedings of International Conference on Computer-Aided Design*, pp.680-684, 1993.
- [5] S. Panda, F. Somenzi, and B. F. Plessier, "Symmetry detectoin and dynamic variable ordering of design diagrams," *Proceedings of International Conference on Computer-Aided Design*,

pp.628-631, 1994.

- [6] C. C. Tsai and M. Marek-Sadowska, "Generalized Reed-Muller forms as a tool to detect symmetries", *IEEE Transactions on Computers*, 45, pp.33-40, 1996.
- [7] C. R. Edwards and S. L. Hurst, "A digital synthesis procedure under function symmetries and mapping methods," *IEEE Transactions on Computers*, C-27, pp.985-997, 1978.
- [8] R. Marculescu, D. Marculescu, and M. Pedram, "Switching activity analysis considering spatio-temporal correlations," *Proceedings of International Conference on Computer-Aided Design*, pp.294-299, 1994.
- [9] E. Sentovich, et al. "SIS: A system for sequential circuit synthesis," Report M92/41, University of California Berkeley, 1992.

정 기 석

정보과학회 논문지 : 시스템 및 이론
제 30 권 제 5 호 참조

김 태 환

정보과학회 논문지 : 시스템 및 이론
제 30 권 제 5 호 참조