論文98-35C-7-6

면적 제약조건하의 저전력 조합회로 설계를 위한 분할 기반 합성 알고리듬

(A Partitioning-based Synthesis Algorithm for the Design of Low Power Combinational Circuits under Area Constraints)

崔 益 成 *, 金 衡 ***, 黃 善 泳 **

(Ick-Sung Choi, Hyoung Kim, and Sun Young Hwang)

요 약

본 논문에서는 면적 제약조건하의 저전력 조합회로 합성 알고리듬을 제안하였다. 제안된 알고리듬은 주어진 회로를 여러 개로 분할하고 이 중 하나를 선택·수행하여 불필요한 신호 전이를 줄인다. 주어진 회로는 면적 제약조건하에서 전력 소모를 반영하는 비용함수를 사용한 adaptive simulated annealing을 이용하여 여러 개로 분할된다. MCNC 벤치마크 회로에 대한 실험을 통하여 제안된 알고리듬이 sis 1.2로 최적화한 회로와 precomputation 논리에 바탕을 둔 전력 최소화 알고리듬에 비하여 전력 소모가 각각 61.1 % 와 51.1 % 향상된 회로를 생성함을 보였다.

Abstract

In this paper, we propose a synthesis algorithm for the design of low power combinational circuits under area constraints. The proposed algorithm partitions a given circuit into several subcircuits such that only a selected subcircuit is activated at a time, hence reduces unnecessary signal transitions. Partitioning of a given circuit is performed through adaptive simulated annealing algorithm employing the cost function reflecting power consumption under area constraints. Experimental results for the MCNC benchmark circuits show that the proposed algorithm generates the circuits which consume less power by 61.1 % and 51.1 %, when compared to those generated by the sis 1.2 and the precomputation algorithm, respectively.

Ⅰ. 서 론

VLSI 제조 및 설계 기술의 발달과 더불어 최근 휴대용 컴퓨터와 통신 시스템의 수요가 급증함에 따라 회로의 발열 문제와 전력 소모를 개선하기 위하여 칩의 고성능, 고집적화 경향과 함께 저전력 회로의 설계가 점차 중요한 설계 사양으로 등장하게 되었다. 이에 따라 저전력 회로 설계 자동화에 관한 연구가 활발하게 진행되고 있으며, 상위 수준 저전력 합성기, 전력 소모 예측 시스템 등이 개발되어 보고되고 있다[1] [2].

CMOS 소자로 구현된 VLSI 회로는 내부 노드값의 스위칭 활동(switching activity)에 의한 전력 소모가 전체 전력 소모량의 90 % 이상을 차지하므로, 회로의 평균 소모 전력을 줄이기 위하여 스위칭 활동을 줄이는 방안이 다양한 설계 계층에서 제안되었다[3

* 正會員, ** 終身會員, 西江大學校 電子工學科
(Dept. of Electronic Engineering Sogang Univ.)
*** 正會員, 慶畝專門大學校 電子計算學科
(Dept. of Computer Science Kyungmin Univ.)
接受日字:1997年12月4日, 수정완료일:1998年6月1日

] [4]. 저전력 소모를 위한 논리회로 설계는 주로 STG(State Transition Graph) 수준과 논리 게이트 및 F/F 수준에서 이루어지고 있으며, 스위칭 활동을 줄이기 위하여 don't-care optimization, path balancing, factorization, precomputation, 회로 분할 알고리듬 등이 제안되었다[5] [6] [7] [8]. Don't-care optimization은 internal don't-care 중 회로의 중간 노드값에 의해 영향을 받지 않는 출력에 의한 ODC(Observability Don't-Care set)를 이용하여 회로의 신호 전이 확률을 감소시켜 소모 전력을 줄이는 설계 방법이다[5] [9]. Path balancing은 회로의 경로 지연시간을 적절히 조정하여 동적 소모 전력의 10 - 40 %를 차지를 차지하는 불필요한 spurious transition에 의한 소모 전력을 줄이는 설계 방법으로, 서로 다른 경로의 지연시간을 동일하게 맞추기 위하여 적절한 지연시간의 게이트를 사용하거나 게이트 입력에 버퍼를 추가한다[6]. Factorization은 주어진 논리식의 공통 부분을 재사용함으로써 회로의 면적과 지연시간을 개선하는 기술 독립적인 최적화 방법이다. 다단 논리 최적화 과정은 논리식의 공통 부분을 효율적으로 찾기 위하여 리터럴 수를 가장 줄일 수 있는 커널을 사용하며[9], 커널 선택의 비용함수에 리터럴 수와 스위칭 활동을 반영하여 factorization을 수행한다[10]. Precomputation 설계 방법은 입력 조건에 따라 출력에 영향을 미치지 못하는 회로의 영역을 예측하여 이 영역의 상태 전이가 일어나지 않도록 함으로써 전력 소모를 줄이는 알고리듬이다[8]. 회로 분할 알고리듬은 스위칭 활동을 줄이기 위해 회로를 분할한 후 입력 조건에 따라 선택적으로 수행하는 구조의 회로를 생성하며 Shannon expansion을 이용한 분할 알고리듬과 커널을 이용한 분할 알고리듬이 제안되었다[11]. Shannon expansion을 이용한 분할 알고리듬은 전력 소모를 효과적으로 줄이는 입력을 선택하여 회로를 분할하며, 커널을 이용한 분할 알고리듬은 소모 전력 감소에 효과적인 커널을 선택하여 회로를 분할한다.

Path balancing 설계 방법은 버퍼나 게이트 capacitance 증가에 따라 전력 소모가 증가하므로 스위칭 활동의 감소에 의한 소모 전력의 감소 효과가 상쇄되어 효율이 좋지 않으며, precomputation 설계 방법은 추가 회로에 의해 지연시간이 증가하고 적절한 ODC가 존재하지 않는 회로에 대하여 비효율적이므로 일반적인 회로에 적용하기 어려운 단점이 있다. 회로 분할 알고리듬은 일반적인 회로에 적용이 가능하고 전력 소모를 효과적으로 감소시킬 수 있으나 면적이 증가하고 greedy 알고리듬을 사용하므로 지역 최적해(locally optimal solution)를 찾는 단점이 있다. 이에 따라 소모 전력, 면적과 지연시간을 고려하여 회로를 설계할 수 있는 효율적인 회로 합성 알고리듬의 개발이 요구되고 있다.

제안된 회로 분할 알고리듬은 기존 설계 방법의 단점을 보완하여 일반적인 회로에 적용이 가능하고 면적 제약조건하에서 소모 전력을 고려하여 설계를 수행할 수 있으며, 기존의 분할 알고리듬의 단점인 면적 overhead를 줄이고 simulated annealing을 이용하여 전역 최적해(globally optimal solution)를 탐색할 수 있도록 개선하였다. 이를 바탕으로 주어진 회로를 여러 개의 세부 회로로 분할하고 이 중 한 회로를 선택하여 수행하는 구조의 회로를 생성하는 합성 시스템을 개발하였다. 본 논문의 2 장에서는 논리회로의 전력 소모와 기존의 저전력 회로 합성에 관련된 연구를 설명하고, 3 장에서 제안된 simulated annealing을 이용한 회로 분할 알고리듬을 기술한다. 4 장에서 실험 결과를 제시하고, 마지막으로 5 장에서 결론을 맺는다.

Ⅱ. 논리회로의 저전력 설계

1. 논리회로의 전력 소모

CMOS 회로의 전력 소모 P_{avg}는 V_{dd}를 공급 전압, T_c는 클럭 주파수, C는 노드 capacitance, N은 클럭 주기당 노드값의 평균 전이 횟수라고 할 때 식 (1)과 같이 나타낼 수 있다[12].

$$\begin{aligned} P_{avg} &= P_{switching} + P_{short_circuit} + P_{leakage} \\ &= \frac{1}{2T_c} N C V_{dd}^2 + \frac{1}{T_c} N Q_{sc} V_{dd} + I_{leak} V_{dd} \end{aligned} \quad (1)$$

식(1)에서 첫항 $P_{switching}$은 스위칭 활동에 의한 전력 소모를 나타내는 항목으로 내부 노드의 충방전에 의해 소모되는 전력이며, 두번째 항 $P_{short_circuit}$는 노드값의 전이 시 V_{dd}와 접지 사이에 일시적으로 흐르는 전류(short-circuit current)에 의한 소모 전력을 나타내며, Q_{sc}는 short-circuit 전류에 의해 생성되는 전하량을 가리킨다. 마지막 항 $P_{leakage}$는 회로의 누설전류(leakage current) I_{leak}에 의한 전력 소모를 나타

낸다[12] [13]. 회로 내 노드 x 의 신호 확률(signal probability) $p_s(x)$는 여러 클럭 주기 동안 안정된 논리 high 값을 가지는 클럭 주기의 평균 비율이고, 회로 내 노드 x의 전이 확률(transition probability) $p_t(x)$ 는 여러 클럭 주기 중 초기값이 전이되는 클럭 주기의 평균 비율이다[12]. CMOS 회로의 경우 노드 값의 전이에 의해 발생하는 소모 전력이 전체 소모 전력의 대부분을 차지하므로 회로의 전체 소모 전력을 $P_{switching}$으로 가정하였다. 회로의 평균 전력 소모량 P_{avg}는 T_c를 clock period, C_i 를 노드 x_i 의 총 capacitance, V_{dd}를 공급 전압, n 을 회로 내의 노드 수라고 할 때 식 (2)과 같이 나타낼 수 있다.

$$P_{avg} \cong P_{switching} = \frac{1}{2T_c} V_{dd}^2 \sum_{i=1}^{n} C_i \, p_t(x_i) \qquad (2)$$

2. 관련 연구

1) Precomputation 논리에 바탕을 둔 설계 알고리듬

Precomputation 논리에 바탕을 둔 설계 알고리듬은 입력 조건에 따라 출력에 영향을 미치지 못하는 회로의 영역을 예측하여 이 영역의 상태 전이가 일어나지 않도록 함으로써 전력 소모를 줄이는 방법이다. Precomputation 논리는 회로의 입력 조건으로부터 2 개의 세부 회로 중 수행할 하나의 회로를 선택하며, 회로에서 추출한 ODC에 universal quantification 연산을 수행하여 precomputation 함수를 계산할 수 있다[8].

그림 1 (a)는 일반적인 회로로 함수 f를 구현한 회로 C_i와 입력 래치 L_i로 나타낸 회로이다. 그림 1 (b)는 그림 1의 회로를 precomputation 구조로 합성한 회로이며, 여기서 g_1, g_2는 precomputation 함수로 $g_1 = 1$ 이면 $f = 1$ 이고, $g_2 = 1$ 이면 $f = 0$ 의 조건을 만족한다. 만일 클럭 주기 t 에 g_1 또는 g_2가 1이면 래치의 L_i load-enable(LE) 신호가 0 이 되어 전 클럭 주기의 입력 값이 유지되므로 클럭 주기 t+1 동안 회로 블럭 C_i 는 값이 바뀌는 내부 노드가 없어 전력 소모가 감소한다. 전력 소모를 효과적으로 줄일 수 있는 precomputation 함수를 찾는 휴리스틱 알고리듬이 제안되었다[8].

2) Shannon expansion을 이용한 분할 알고리듬

일반적인 논리 함수 f 는 입력 집합이 $I = \{ i_1, i_2, \ldots, i_n \}$ 인 경우 Shannon expansion을 적용하여 식 (3)과 같이 변수 i_j에 대한 cofactor를 이용하여 표현할 수 있다.

$$f = i_j f_{i_j} + \overline{i_j} f_{\overline{i_j}} \qquad (3)$$

그림 1 (c)는 Shannon expansion을 이용한 회로 구조를 나타낸 것이며, 선택된 입력 i_1 의 값에 따라 cofactor f_{i_1}, $f_{\overline{i_1}}$ 를 구현한 2 개의 세부 회로가 배타적으로 선택되어 실행된다. 입력 i_1 의 값이 1 일 때 f_{i_1} 회로가 수행되고 $f_{\overline{i_1}}$ 회로는 disable되며, i_1 의 값이 0 일 때 $f_{\overline{i_1}}$ 블록이 수행되고 f_{i_1}가 disable되는 구조이다[8].

Shannon expansion을 이용한 저전력 논리 합성 과정은 전력 소모를 최대로 줄일 수 있도록 적절한 입력 변수를 선택하여 회로를 분할해야 한다. 최적의 저전력 소모 회로는 모든 변수를 이용하여 합성한 결과 회로의 전력 소모를 예측하고 이 중 가장 소모 전력이 적은 경우를 선택하는 것이 바람직하다. 회로를 분할 시 입력은 합성된 회로의 소모 전력을 효율적으로 줄일 수 있도록 선택되어야 한다. Precomputation 설계 방법은 ODC가 존재하지 않는 회로의 경우 합성의 효율이 좋지 않고 효율적인 precomputation 논리를 구성할 수 있는 회로가 제한되어 있는 반면, Shannon expansion을 이용한 저전력 회로 구조는 보다 일반적인 회로에 적용할 수 있고 면적과 지연시간의 overhead가 적으나 precomputation 설계 방법과 같이 동일한 입력이 중복 사용되므로 입력 래치수의 증가에 따라 면적이 증가하는 단점이 있다.

3) 커널에 바탕을 둔 분할 알고리듬

커널을 이용한 분할 알고리듬은 스위칭 활동을 효과적으로 줄일 수 있는 커널을 이용하여 회로를 분할하며, 이를 이용한 저전력 회로 합성 알고리듬은 입력 조건에 따라 분할된 회로 중 하나를 택일하여 수행하는 구조의 회로를 생성한다. 그림 1 (d)에 커널을 이용한 분할 알고리듬에 의해 생성되는 회로의 구조를 나타내었다. 커널은 회로내에서 공통되는 부분을 나타낼 수 있는 가장 기본적인 단위이므로 단일 출력 함수뿐만 아니라 다출력 함수에서 공유되는 부분을 찾기 위해 자주 사용된다. 커널은 회로를 분할하는 효율적인 divisor로 사용이 가능하므로 적절한 커널을 이용하여 회로를 분할함으로써 전력 소모와 면적을 개선할 수 있다[11]. 커널을 이용한 분할 알고리듬은 합성 과정에서 ODC를 사용하지 않으므로 보다 일반적인 회

로에 적용할 수 있는 장점이 있으나, 커널을 이용하여 다중 분할하는 경우 selection logic의 면적과 지연시간이 증가되어 효율적이지 않은 단점이 있다.

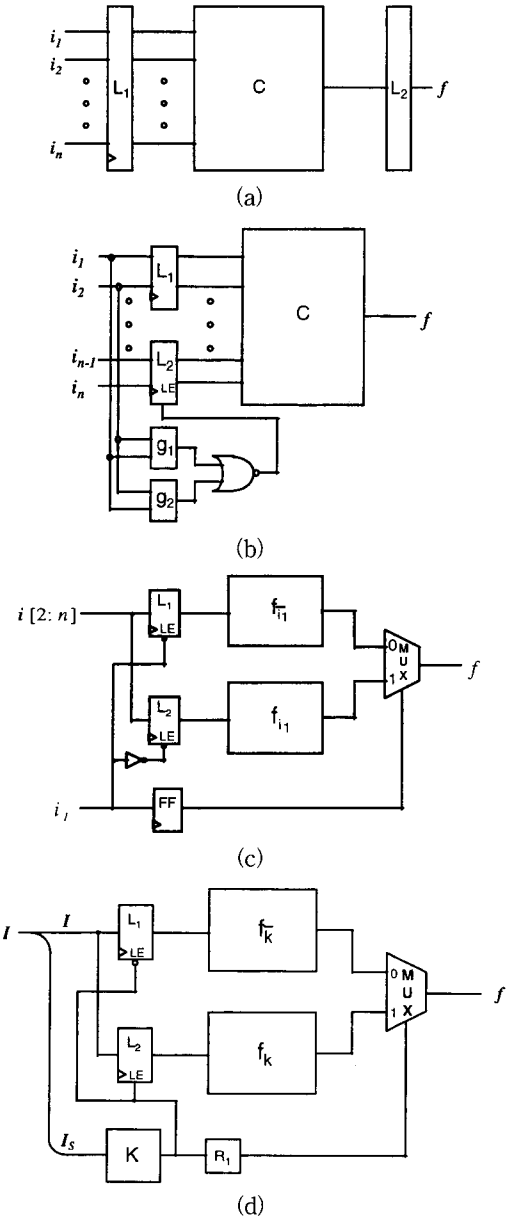


그림 1. 저전력 소모를 위한 여러 가지 회로 구조
(a) 일반적인 조합 회로 (b) Precomputation 논리를 이용한 회로 구조 (c) Shannon expansion을 이용한 회로 구조 (d) 커널을 이용한 회로 구조

Fig. 1. Various circuit structure for low power consumption. (a) A combinational circuit (b) A structure based on precomputation scheme (c) Circuit structure based on Shannon expansion (d) Circuit structure partitioned by a selected kernel

Ⅲ. Simulated Annealing을 이용한 합성 알고리듬

실생활과 이론에서 자주 등장하는 NP-complete 문제에서는 문제의 크기가 증가함에 따라 지수 함수에 비례하는 수행 시간 혹은 기억 장소가 필요하므로 빠른 시간 내에 근사 해를 구하기 위하여 simulated annealing 방식이 제안되었다[14]. Simulated annealing은 물리학의 담금질 과정을 시뮬레이션하여 해를 효율적으로 탐색하는 조합 최적화(combinatorial optimization) 알고리듬으로 구현이 쉬우며 빠른 시간 내에 양질의 해를 얻을 수 있는 장점이 있다.

조합 최적화 알고리듬은 제약조건하에서 정보를 나열함으로써 해를 구성하며 해공간, 비용함수, 이웃 구조 등이 명확하게 정의되어야 한다. 해공간은 주어진 문제의 모든 가능한 유한 개의 해 집합을 나타내며 비용함수는 해를 평가하기 위해 사용된다. 이웃 구조는 하나의 해를 변형시켜 그 해에 근접한 이웃해의 집합을 구할 수 있는 구조로 해공간을 모두 탐색 가능해야 하며 방대한 해공간을 탐색하기 위해 이웃 구조 생성 과정의 시간 복잡도(time complexity)가 낮아야 한다. Simulated annealing의 온도 냉각 방법은 수행 시간에 많은 영향을 미치므로 방대한 해공간을 빠른 시간 내에 탐색하기 위하여 효율적인 온도 냉각 방법의 사용이 필수적이다. 일차원 공간의 해를 효율적으로 찾기 위해 시간 t에 비례하여 일정한 간격으로 온도를 냉각시키는 방법, 온도를 1/log(t)의 간격으로 일정하게 냉각시키는 방법, Boltzmann annealing, simulated quenching, fast annealing 등의 온도 냉각 방법이 제안되었다[14] [15]. 다수의 변수를 사용하는 실생활의 문제들을 빠른 시간내에 해결하기 위해 다차원 공간을 효율적으로 탐색하기 위한 온도 냉각 방법의 개발이 필요하게 되었다. 기존의 온도 냉각 방법들은 모든 변수에 대하여 동일한 annealing schedule을 적용하는 반면, 각 변수마다 효과적인 annealing schedule을 적용함으로써 탐색 시간을 효과적으로 개선한 adaptive simulated annealing 방법이 제안되었다[16] Adaptive simulated annealing 방법은 reannealing과 simulated quentching 기법을 사용하여 전역해를 찾는 시간을 현저히 개선하였으며, reannealing는 일정한 기간 동안 온도 변화에 따른 변수의 sensitity를 계산한 후 민감하지 않은 변수의

탐색 구간을 annealing time을 조정하여 적절히 확장함으로써 전역 해를 빠른 시간 내에 찾을 확률을 증가시키는 방법이며, simulated quentching은 시간에 따른 비용함수의 변화가 적을 경우 온도를 급격하게 내림으로써 해의 탐색 시간을 줄이는 방법이다. 제안된 알고리듬은 다차원 공간의 분할 해를 효율적으로 탐색하기 위해 adaptive simulated annealing 방법을 사용하였다[16].

기존의 분할 알고리듬은 전력 소모를 효율적으로 줄일 수 있으나 면적과 지연시간의 overhead가 큰 단점이 있으며, 제안된 분할 알고리듬은 면적 제약조건하에서 전력 소모를 고려한 비용함수를 사용하므로 면적과 지연시간의 overhead를 줄일 수 있다. Simulated annealing은 확률적으로 local optima에서 벗어날 수 있는 가능성을 제공하여 보다 효율적인 회로를 설계할 수 있다. Simulated annealing 알고리듬을 이용하여 문제를 해결할 경우 해의 코딩 방법, 전체 해공간, 해를 평가하는 비용함수, 이웃해를 구하는 move 방법, 온도 냉각 방법을 적절히 설계하여야 하며, 제안된 알고리듬의 해 표현 방법, 비용함수, 이웃해를 구하는 방법을 기술하였다.

1. 해의 표현

논리회로는 최대 $2^{\#inputs-1}$ 개로 분할이 가능하므로 $O(2^{\#inputs})$의 해공간이 존재한다. 제안된 알고리듬은 분할된 회로를 나타내는 해를 체계적으로 표현하기 위하여 분할트리를 사용하였다. 분할트리의 노드는 회로 분할에 사용된 변수를 나타내며 leaf 노드를 제외한 모든 노드는 두 자손을 가진다. 노드의 자손들은 노드에 해당하는 변수를 이용하여 분할된 두 cofactor 회로를 나타내며, 분할된 회로들은 트리의 leaf 노드 집합으로 나타낼 수 있다. 그림 2에 MCNC 벤치마크 조합회로 'b1'을 보였으며, $bc, b\bar{c}, \bar{b}$ 의 입력으로 cofactor된 3 개의 세부 회로로 나누어진 해를 나타내는 분할 트리를 그림 3에 나타내었다. 그림 2의 예제 회로를 그림 3의 분할 트리에 해당하는 회로로 변환할 경우 그림 4의 결과 회로를 얻을 수 있다.

제안된 알고리듬은 한번 방문한 해를 중복 방문하지 않기 위하여 트리의 각 노드마다 과거에 분할을 시도한 변수의 집합을 유지하며, 이를 이용하여 시행 착오와 탐색 시간을 줄일 수 있다. 모든 노드에 대하여 분할에 사용된 변수 집합을 저장할 경우 메모리 사용이 증가하는 overhead가 있으므로 이를 개선하기 위하여 변수 집합을 비트 단위로 packing하여 관리하였다.

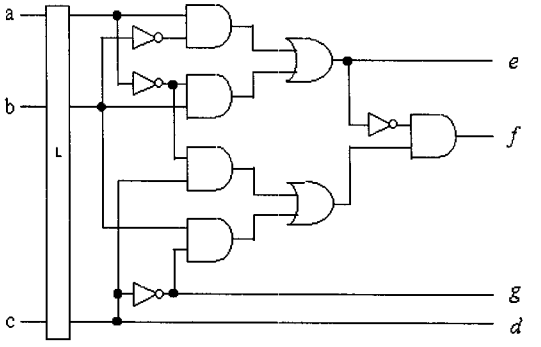


그림 2. 'b1' 벤치마크 회로.
Fig. 2. Benchmark circuit 'b1'.

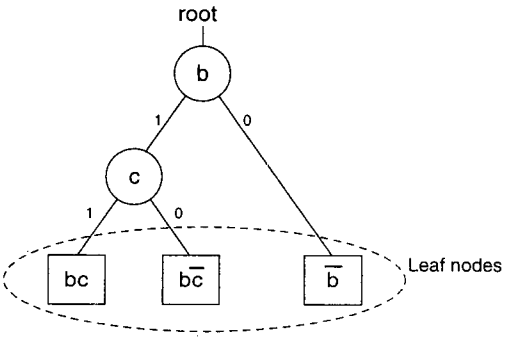


그림 3. 'b1' 회로의 분할트리
Fig. 3. The partition tree for the circuit 'b1'.

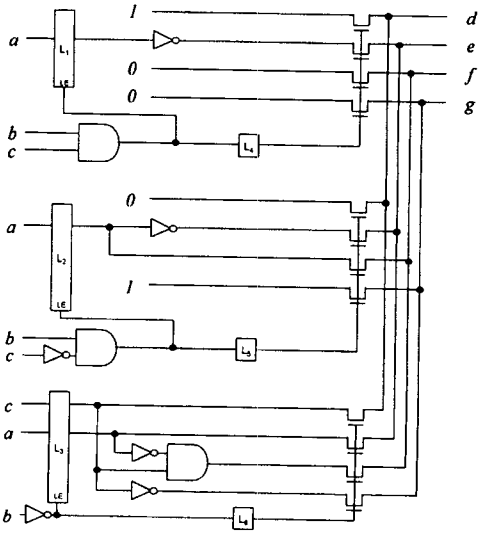


그림 4. 제안된 구조로 합성한 'b1' 회로.
Fig. 4. The circuit 'b1' transformed into the proposed scheme.

2. 이웃해의 생성

Move는 한 해로부터 이웃해를 구하기 위해 분할된 회로를 임의로 하나 선택하여 분할 혹은 병합 과정을 거쳐 새로운 분할 집합을 구하는 과정이다. 방대한 해공간을 빠른 시간 내에 탐색하기 위하여 이웃해를 구하는 과정의 시간 복잡도는 낮아야 하며 전체 해공간의 탐색이 가능해야 한다.

제안된 시스템은 회로의 분할과 병합을 위한 두 종류의 move를 사용한다. 이웃해는 난수를 이용하여 트리의 leaf를 선택하고 이를 병합 혹은 분할하여 생성된다. 트리상에서 leaf 노드를 선택하는 방법은 루트 노드로부터 leaf 노드까지의 경로를 결정하는 방법과 leaf 노드에 유일한 정수를 할당한 후 이 중 하나를 선택하는 방법이 있다. 전자는 루트로부터 leaf 노드의 경로를 난수를 이용하여 결정하는 방법으로 그림 3의 예에서 난수 11은 bc로 나눈 회로, 난수 10은 회로를 $b\bar{c}$로 나눈 회로, 그 외의 난수는 $\bar{b}$로 나눈 회로를 선택한다. 이 방법은 각 leaf가 선택될 확률이 2^{depth}로 depth가 깊은 leaf 노드일수록 선택될 확률이 적어지므로 트리의 좌우가 균형있게 분할되는 성질이 있어 회로를 균일한 크기로 나눌 수 있는 장점이 있다. 반면, 이 방법은 과거에 선택한 leaf 노드를 다시 중복 선택하는 횟수가 많은 단점이 있다. 후자의 leaf 노드 선택 방법은 트리의 모든 leaf 노드에 유일한 정수를 할당한 후 난수를 이용하여 이 중 한 leaf 노드를 선택하며, 그림 3의 leaf 노드인 $bc, b\bar{c}, \bar{b}$ 중 하나를 임의로 선택하는 방법이다. 이 방법은 난수를 이용한 leaf 노드의 선택시 중복 선택되는 경우가 적은 장점이 있지만 트리의 depth에 영향을 받지 않으므로 분할되는 회로의 크기가 고르지 못하고 트리가 수정될 때마다 leaf 노드에 유일한 정수를 재할당해야 하는 단점이 있다.

제안된 알고리듬은 수행 시간을 줄이기 위하여 만일 선택된 leaf 노드에 해당하는 회로가 과거에 사용한 변수를 이용하여 분할을 시도하는 경우에는 분할 과정을 거치지 않도록 설계하였다. 제안된 이웃해를 구하는 알고리듬을 그림 5에 나타내었다. 알고리듬에서 partition_circuit은 주어진 회로를 변수에 대한 두 cofactor 회로로 분할하는 과정이며, Merge_circuit는 두 cofactor 회로를 분할하기 전 상태의 회로로 변환하는 과정이다. Partition과 Merge는 임의로 선택하여 수행된다.

```
/*
  T:  partition tree whose leaf node represents a circuit partition.
*/
move (T, mode)
{
   leaf = get_random_leaf(T);      /* Select a subcircuit randomly. */
   if (mode == Partition)  {
      /* Select an input variable which has not been tried. */
      variable = select_random_input(leaf);
      Partition_circuit(leaf, variable);
   } else {   /* mode == Merge */
      if (leaf == root_node)  return FAIL;
      Merge_circuits(leaf->parent->left,  leaf->parent->right);
   }
   return SUCCESS;
}
```

그림 5. 제안된 이웃해를 구하는 알고리듬
Fig. 5. Move procedure in the proposed algorithm.

3. 해의 평가를 위한 비용함수

해의 평가는 비용함수를 통하여 이루어지며, 주어진 제약조건하에서 해의 비용이 최소가 되는 해를 최적의 해로 간주한다. 제안된 분할 알고리듬은 회로의 면적 제약조건을 만족하는 경우 분할된 회로의 전력 소모 감소량을 비용함수로 사용하며, 합성된 결과 회로가 제약조건을 만족하지 않는 경우 비용함수는 가장 큰 실수값을 출력한다. 제약조건을 만족하는 경우 제안된 알고리듬의 비용함수를 식 (4)에 나타내었다.

비용함수=합성된 결과 회로의 전력 소모-합성전 회로의 전력 소모 (4)

제안된 알고리듬은 adaptive simulated annealing을 사용하였으며, 이 알고리듬은 탐색 공간이 상대적으로 큰 다차원 공간을 효율적으로 탐색할 수 있는 장점이 있다[16].

그림 6에 제안된 회로 분할 알고리듬에 의해 생성되는 회로의 구조를 나타내었으며, 전체 입력 집합 I = { i_1, i_2, ... , i_n } 의 부분 집합인 I_s의 값에 따라 분할된 회로 중 하나의 회로가 선택하여 수행한다. 출력단에 사용된 mux는 각 출력단에 transmission gate 하나를 추가하여 구현이 가능하다. 제안된 알고리듬은 분할된 회로들의 소모 전력의 편차가 적은 경향이 있으므로 입력 값의 변화에 따라 회로의 소모 전력의 변화가 작은 회로를 합성할 수 있는 장점이 있다.

그림 2에 보인 조합회로 예제는 sis 1.2 에서 script.rugged 회로를 이용하여 최적화한 MCNC 벤치마크의 'b1' 회로로, 입력의 signal probability를

0.5 로 가정할 때 44.8 μW 의 전력을 소모하고 리터럴의 수는 16 개다[17]. 예제 회로를 입력 집합 선택 알고리듬을 사용하여 precomputation 구조로 합성한 회로를 그림 7 (a)에 보인다. 선택한 입력 집합이 { b, c } 이고 precomputation 함수가 g = b′ c + b c′ 일 때 전력 소모는 40.2 μW 이고 리터럴 수는 19 개이다. 이 예제 회로를 입력 c 를 선택하여 Shannon expansion 구조로 합성한 경우의 회로를 그림 7 (b)에 보인다. 이 때 소모 전력은 22.8 μW 이며 리터럴의 수는 18 개이다. 커널에 바탕을 둔 구조로 예제 회로를 합성한 경우 전력 소모가 26.6 μW 이고 리터럴의 수가 18 개인 그림 7 (c)의 회로를 얻을 수 있다. 제안된 회로 분할 알고리듬을 사용하여 그림 2의 예제 회로를 합성할 경우 그림 4와 같이 소모 전력이 4.2 μW 로 크게 감소하고 리터럴의 수가 21 개인 회로를 얻었다.

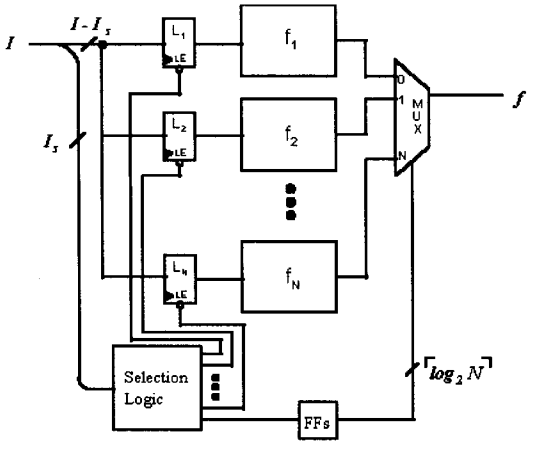
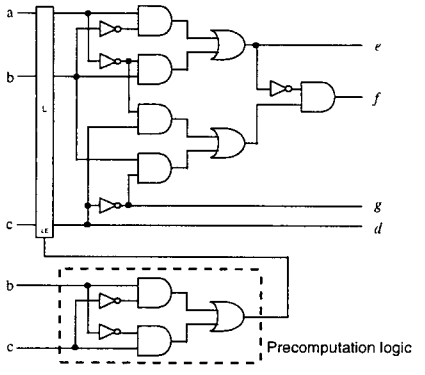
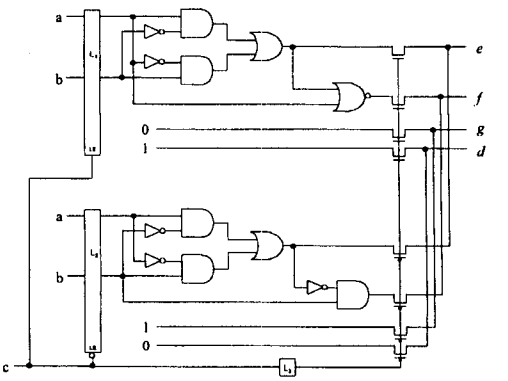
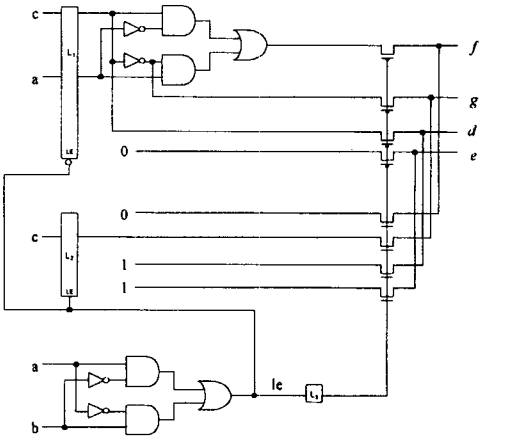


그림 6. 제안된 알고리듬에 의해 생성된 회로 구조
Fig. 6. The circuit structure obtained by the proposed algorithm.





(b)



(c)

그림 7. 저전력 조합회로의 설계를 위한 다양한 회로 구조
(a) Precomputation 구조로 합성한 회로
(b) Shannon expansion 구조로 합성한 회로
(c) 커널에 바탕을 둔 구조로 합성한 회로
Fig. 7. Various schemes for low power design of combinational circuits.
(a) Circuit obtained by the precomputation scheme (b) Circuit obtained by the Shannon expansion scheme (c) Circuit obtained by the kernel-based scheme

Ⅳ. 실험 결과

제안된 알고리듬의 성능 평가를 위해 MCNC 벤치마크 회로에 대해 실험을 수행하였다. MCNC 벤치마크 회로와 이를 sis 1.2의 script.rugged를 이용하여 면적을 최적화한 회로를 표 1에 나타내었다[17]. 실험에서 전력 소모량은 sis 1.2를 이용하여 5 V 공급 전압과 20 MHz 클럭 주파수에서 zero-delay model을 사용하여 측정하였으며, 지연시간을 비교하기 위해 회

로를 MCNC 라이브러리로 기술 매핑한 후 측정한 레벨을 나타내었다.

표 1. 제안된 알고리듬의 성능평가에 사용된 벤치마크 회로

Table 1. Benchmark circuits used for the performance evaluation of the proposed algorithm.

Circuits	Benchmark circuits					The circuits optimized by the sis 1.2 script.rugged		
	# ins	#outs	Area (# lits)	Delay (# levels)	Power (uW)	Area (# lits)	Delay (# levels)	Power (uW)
b1	3	4	23	3	81	16	3	45
cc	21	20	136	4	424	99	7	183
cm42a	4	10	43	4	116	42	5	111
cm82a	5	3	36	6	123	34	6	101
cm138	6	8	57	4	151	43	5	87
cm150	21	1	119	11	418	94	7	212
cm162	14	5	86	5	288	74	7	298
cmb	16	4	94	7	273	84	5	228
count	35	16	244	19	600	213	20	588
cordic	23	2	240	9	530	110	7	344
cu	14	11	118	6	414	86	7	221
c8	28	18	286	5	1,558	193	11	538
decode	5	16	78	3	317	87	3	328
duke2	22	29	982	12	10,934	458	12	906
e64	65	65	2,274	6	10,714	383	63	348
majority	5	1	21	4	88	20	4	64
misex2	25	18	214	6	935	156	7	380
mux	21	1	134	10	699	157	7	380
f51m	8	8	185	7	1,713	130	25	716
pcler8	27	17	149	16	272	147	12	284
sao2	10	4	156	16	633	156	18	643
sct	19	15	202	6	1,117	146	33	292
tcon	17	16	74	2	221	66	3	135
x2	10	7	91	5	346	67	8	203
z4ml	7	4	91	8	1,361	55	10	208
9symml	9	1	295	11	1,264	178	11	1,032
C1908	33	25	1,497	40	3,425	540	18	1,783
i9	88	63	1453	7	5,068	592	6	2,215
dalu	75	16	4541	24	9,808	964	14	5,366
Average	17	12	247	8	1,369	186	12	629

표 2에서 MCNC 벤치마크 회로와 제안된 알고리듬에 의해 생성된 결과 회로의 전력 소모를 나타내었다. 표 3에서 제안된 분할 알고리듬과 기존의 precomptuation 알고리듬[8]의 실험 결과를 비교하였다. Precomputation 알고리듬과 제안된 알고리듬에 의해 생성된 회로의 소모 전력과 면적의 차이를 sis 1.2의 script.rugged를 이용하여 최적화한 벤치마크 회로에 대하여 비교한 값을 백분율로 나타내었다. 제약조건이 주어지지 않은 경우 제안된 알고리듬으로 합성한 회로의 전력 소모가 sis 1.2와 precomputation 알고리듬에 비해 각각 61.1 %, 51.1 % 감소하였다.

표 2. 제안된 방식에 의해 생성된 회로의 전력 소모

Table 2. Power consumption of the circuits produced by the proposed scheme.

Circuits	Benchmark circuits					The proposed algorithm					
						Area		Delay		Power	
	#ins	#outs	Area (#lits)	Delay (#levels)	Powe (uW)	#lits	비교(%)	#levels	비교(%)	uW	비교(%)
b1	3	4	23	3	81	21	- 8.7	3	0.0	4	- 94.7
cc	21	20	136	4	424	129	- 5.1	3	- 25.0	90	- 78.8
cm42a	4	10	43	4	116	56	+ 30.2	4	0.0	34	- 70.3
cm82a	5	3	36	6	123	75	+ 108.3	5	- 16.7	49	- 60.1
cm138	6	8	57	4	151	93	+ 63.2	4	0.0	29	- 81.0
cm150	21	1	119	11	418	98	- 17.6	6	- 45.5	103	- 75.4
cm162	14	5	86	5	288	117	+ 105.8	7	+ 40.0	41	- 85.7
cmb	16	4	94	7	273	110	+ 17.0	4	- 42.9	59	- 78.5
count	35	16	244	19	600	203	- 16.8	20	+ 5.3	224	- 62.7
cordic	23	2	240	9	530	131	- 45.4	4	- 55.6	176	- 66.8
cu	14	11	118	6	414	97	- 17.8	7	+ 16.7	94	- 77.4
c8	28	18	286	5	1,558	244	- 14.7	11	+ 120.0	285	- 81.7
decode	5	16	78	3	317	95	+ 21.8	3	0.0	62	- 80.3
duke2	22	29	982	12	10,934	518	- 47.3	13	+ 8.3	425	- 96.1
e64	65	65	2,274	6	10,714	387	- 83.0	54	+ 800.0	99	- 99.1
majority	5	1	21	4	88	48	+ 128.6	3	- 25.0	13	- 84.9
misex2	25	18	214	6	935	169	- 21.6	7	+ 16.7	173	- 81.5
mux	21	1	134	10	699	126	- 6.0	5	- 50.0	81	- 88.4
f51m	8	8	185	7	1,713	209	+ 13.0	14	+ 100.0	245	- 85.7
pcler8	27	17	149	16	272	128	- 14.1	14	- 12.5	103	- 61.9
sao2	10	4	156	16	633	154	- 1.3	17	+ 6.3	334	- 47.3
sct	19	15	202	6	1,117	197	- 2.5	29	+ 383.3	130	- 88.4
tcon	17	16	74	2	221	56	- 24.3	1	- 50.0	0	- 100.0
x2	10	7	91	5	346	84	- 7.7	9	+ 80.0	100	- 71.1
z4ml	7	4	91	8	1,361	179	+ 96.7	11	+ 37.5	109	- 92.0
9symml	9	1	295	11	1,264	332	+ 12.5	10	- 9.1	342	- 72.9
C1908	33	25	1,497	40	3,425	996	- 33.5	26	- 35.0	1,421	- 58.5
i9	88	63	1453	7	5,068	1,533	+ 5.5	10	+ 42.9	2,202	- 56.6
dalu	75	16	4541	24	9,808	2,898	- 36.2	15	- 37.5	1,150	- 88.3
Average	17	12	247	8	1,369	329	+ 6.9	11	- 43.2	282	- 78.1

실험 회로 중 C1908, i9, dalu 회로는 precomputaion 알고리듬의 수행시간과 메모리 사용량이 방대하여 결과회로를 생성하지 못하여, 실험결과에 -로 나타내었다. 제안된 구조에서 cofactor 회로들은 원래 회로에 비해 회로의 depth가 감소되는 특징이 있으며 selection logic에 의한 지연시간이 큰 영향을 미치는

일부 작은 회로 외에는 지연 시간이 증가하지 않으므로 지연시간 제약조건에 영향을 거의 받지 않음을 알 수 있다.

표 3. Precomputation 설계 방식과 제안된 방식의 전력 소모

Table 3. Power consumption of the circuits produced by the precomputation-based scheme and the proposed scheme.

Circuits	Precomputation algorithm						The proposed algorithm					
	Area		Delay		Power		Area		Delay		Power	
	#lits	비교(%)	#levels	비교(%)	(uW)	비교(%)	# lits	비교(%)	#levels	비교(%)	(uW)	비교(%)
b1	19	+ 18.8	4	+ 33.3	40	- 11.1	21	+ 31.3	3	0.0	4	- 90.4
cc	123	+ 24.2	10	+ 42.9	182	- 0.5	129	+119.2	3	- 57.1	90	- 51.0
cm42a	46	+ 9.5	6	+ 20.0	108	- 2.4	56	+ 33.3	4	- 20.0	34	- 69.0
cm82a	38	+ 11.8	7	+ 16.7	112	+ 11.1	75	+120.6	5	- 16.7	49	- 51.7
cm138	51	+ 18.6	7	+ 40.0	57	- 34.6	93	+116.3	4	- 20.0	29	- 67.3
cm150	117	+ 24.5	10	+ 42.9	121	- 42.9	98	+ 4.3	6	- 14.3	103	- 51.4
cm162	106	+ 43.2	9	+ 28.6	159	- 46.6	117	+139.2	7	0.0	41	- 86.2
cmb	56	- 33.3	8	+ 60.0	174	- 23.7	110	+ 31.0	4	- 20.0	59	- 74.2
count	293	+ 37.6	21	+ 5.0	606	+ 3.0	203	- 4.7	20	0.0	224	- 62.0
cordic	175	+ 59.1	11	+ 57.1	299	- 13.0	131	+ 19.1	4	-42.9	176	- 48.8
cu	89	+ 3.5	12	+ 71.4	150	- 32.0	97	+ 12.8	7	0.0	94	- 57.6
c8	209	+ 8.3	14	+ 27.3	613	+ 13.9	244	+ 26.4	11	0.0	285	- 47.1
decode	68	- 21.8	5	+ 66.7	101	- 69.3	95	- 9.2	3	0.0	62	- 81.0
duke2	704	+ 53.7	12	0.0	1,031	+ 13.8	518	+ 13.1	13	8.3	425	- 53.0
e64	388	+ 1.3	66	+ 4.8	513	+ 47.4	387	+ 1.0	54	- 14.3	99	- 71.4
majority	24	+ 20.0	6	- 50.0	39	- 39.1	48	+140.0	3	- 25.0	13	- 79.2
misex2	225	+ 44.2	7	0.0	433	+ 13.7	169	+ 8.3	7	0.0	173	- 54.6
mux	135	- 14.0	10	+ 42.9	261	- 31.3	126	- 19.7	5	- 28.6	81	- 78.7
f51m	207	+ 59.2	26	+ 4.0	468	- 34.7	209	+ 60.8	14	- 44.0	245	- 65.8
pcler8	189	+ 28.6	14	- 16.7	182	- 36.0	128	- 12.9	14	+ 16.7	103	- 63.6
sao2	242	+ 55.1	18	0.0	402	- 37.5	154	- 1.3	17	- 5.6	334	- 48.1
sct	122	- 16.4	35	+ 6.1	297	+ 1.6	197	+ 34.9	29	- 12.1	130	- 55.5
tcon	66	0.0	3	0.0	125	- 7.1	56	- 15.2	1	- 66.7	0	-100.0
x2	71	6.0	13	- 62.5	232	+ 14.3	84	+ 25.4	9	+ 12.5	100	- 50.8
z4ml	59	7.3	11	+ 10.0	214	+ 2.8	179	+225.5	11	+ 10.0	109	- 47.6
9symml	211	+ 18.5	20	+ 81.8	1,312	+ 27.1	332	+ 86.5	10	- 9.1	342	- 66.9
C1908	-	-	-	-	-	-	996	- 84.4	26	+ 44.4	1,421	- 20.3
i9	-	-	-	-	-	-	1,533	+159.0	10	+ 66.7	2,202	- 0.6
dalu	-	-	-	-	-	-	2,898	+200.6	15	+ 7.1	1,150	- 78.6
Average	155	+ 18.0	14	- 30.4	317	- 12.0	329	+ 53.8	11	- 7.9	282	- 61.1

* Experimental results are compared with the circuits optimized by sis 1.2.

실험 결과에서 제안된 알고리듬으로 합성한 회로의 면적은 sis 1.2와 precomputation 알고리듬에 비해 평균적으로 각각 53.8 %, 23.2 %가 증가되었으며, 이는 분할된 cofactor 회로에 중복되는 부분에 의한 것이다. sis 1.2로 최적화한 회로보다 면적이 증가하지 않는 제약조건하에서 제안된 알고리듬으로 합성한 결과를 표 4 (a)에 나타내었으며 sis 1.2에 비해 전력 소모가 평균 18.4 % 감소하였다.

표 4. 면적 제약조건하에서 제안된 알고리듬에 의해 생성된 회로의 실험 결과 (a) sis 1.2로 최적화한 회로의 면적을 증가시키지 않는 제약조건하에서 제안된 알고리듬에 의해 합성된 회로의 실험 결과 (b) 면적 제약조건하에서 제안된 알고리듬에 의해 합성된 회로의 실험결과

Table 4. Experimental results for the circuits obtained by the proposed algorithm under area constraints. (a) Experimental results for the circuits synthesized by the proposed algorithm where the area constraints are given as those of the sis 1.2-optimized circuits (b) Experimental results for the circuit synthesized by the proposed algorithm under various values of area constraints.

(a)

Circuits	Area		Delay		Power	
	#lits	비교(%)	#levels	비교(%)	uW	비교(%)
b1	16	0.0	3	0.0	45	0.0
cc	99	0.0	7	0.0	183	0.0
cm42a	42	0.0	5	0.0	111	0.0
cm82a	34	0.0	6	0.0	101	0.0
cm138	43	0.0	5	0.0	87	0.0
cm150	94	0.0	7	0.0	212	0.0
cm162	74	0.0	7	0.0	298	0.0
cmb	84	0.0	5	0.0	228	0.0
count	203	- 4.7	20	0.0	224	- 62.0
cordic	110	0.0	7	0.0	344	0.0
cu	86	0.0	7	0.0	221	0.0
c8	193	0.0	11	0.0	538	0.0
decode	86	- 1.1	4	+ 33.3	72	- 78.1
duke2	458	0.0	12	0.0	906	0.0
e64	383	0.0	63	0.0	348	0.0
majority	20	0.0	4	0.0	64	0.0
misex2	156	0.0	7	0.0	380	0.0
mux	126	- 19.7	5	- 28.6	81	- 78.6
f51m	130	0.0	25	0.0	716	0.0
pcler8	128	- 12.9	14	+ 16.7	103	- 63.6
sao2	154	- 1.3	17	0.0	334	- 48.1
sct	131	- 10.3	33	0.0	150	- 48.6
tcon	56	- 15.2	1	- 66.7	0	-100.0
x2	67	0.0	8	0.0	203	0.0
z4ml	55	0.0	10	0.0	208	0.0
9symml	178	0.0	11	0.0	1,032	0.0
C1908	540	0.0	18	0.0	1,783	0.0
i9	592	0.0	6	0.0	2,215	0.0
dalu	964	0.0	14	0.0	5,366	0.0
Average	123	- 2.5	12	- 2.0	277	- 18.4

(b)

Circuits	Circuits generated under the constraints of 120 % of sis-1.2 circuits.						Circuits generated under the constraints of 140 % of sis-1.2 circuits.					
	Area		Delay		Power		Area		Delay		Power	
	# lits	비교(%	#levels	비교(%	(uW)	비교(%	#level	비교(%)	Depth	비교(%)	(uW)	비교(%)
b1	16	0.0	3	0.0	45	0.0	21	+31.3	3	0.0	4	- 90.4
cc	99	0.0	7	0.0	183	0.0	99	0.0	7	0.0	183	0.0
cm42a	50	+19.0	4	- 20.0	41	- 62.8	56	+33.3	4	-20.0	34	- 69.0
cm82a	34	0.0	6	0.0	101	0.0	34	0.0	3	-50.0	101	0.0
cm138	43	0.0	5	0.0	87	0.0	43	0.0	5	0.0	87	0.0
cm150	98	+ 4.3	6	- 14.3	103	- 51.4	98	+ 4.3	6	-14.3	103	- 51.4
cm162	87	+17.6	8	+ 14.3	82	- 72.5	99	+33.8	8	+14.3	47	- 84.3
cmb	84	0.0	5	0.0	228	0.0	110	+31.0	4	-20.0	59	- 74.2
count	203	- 4.7	20	0.0	224	- 62.0	203	- 4.7	20	0.0	224	- 62.0
cordic	131	+19.1	4	- 42.9	176	- 48.8	131	+19.1	4	-42.9	176	- 48.8
cu	97	+12.8	7	0.0	94	- 57.6	97	+12.8	7	0.0	94	- 57.6
c8	193	0.0	11	0.0	538	0.0	244	+26.4	11	0.0	285	- 47.1
decode	95	+ 9.2	3	0.0	62	- 81.0	95	+ 9.2	3	0.0	62	- 81.0
duke2	518	+13.1	13	8.3	425	- 53.0	518	+13.1	13	8.3	425	- 53.0
e64	384	+ 1.0	54	- 14.3	99	- 71.6	387	+ 1.0	54	-14.3	99	- 71.4
majority	20	+ 0.0	4	0.0	64	0.0	20	0.0	4	0.0	64	0.0
misex2	169	+ 8.3	7	0.0	60	- 84.2	169	+ 8.3	7	0.0	173	- 54.6
mux	126	-19.7	5	- 28.6	81	- 78.7	126	-19.7	5	-28.6	81	- 78.7
f51m	130	0.0	25	0.0	716	0.0	175	+34.6	19	-24.0	388	- 45.9
pcler8	128	-12.9	14	+ 16.7	104	- 63.6	128	-12.9	14	+16.7	103	- 63.6
sao2	154	- 1.3	17	- 5.6	334	- 48.1	154	- 1.3	17	- 5.6	334	- 48.1
sct	131	-10.3	33	0.0	150	- 48.7	131	-10.3	33	0.0	150	- 48.7
tcon	56	-15.2	1	- 66.7	0	-100.0	56	-15.2	1	-66.7	0	-100.0
x2	67	0.0	8	0.0	203	0.0	84	+25.4	9	+12.5	100	- 50.8
z4ml	55	0.0	10	0.0	208	0.0	55	0.0	6	-40.0	208	0.0
9symml	207	+16.3	9	- 18.2	489	- 52.6	207	+16.3	9	-18.2	489	- 52.6
C1908	540	0.0	18	0.0	1,783	0.0	540	0.0	18	0.0	1,783	0.0
i9	592	0.0	6	0.0	2,215	0.0	592	0.0	6	0.0	2,215	0.0
dalu	964	0.0	14	0.0	5,366	0.0	964	0.0	14	0.0	5,366	0.0
Average	133	- 8.2	11	- 18.0	199	- 42.4	139	- 2.1	12	-19.1	167	- 56.1

* Experimental results are compared with the circuits optimized by sis 1.2.

면적 제약조건에 따라 합성된 회로의 전력 소모를 관찰하기 위하여 sis 1.2로 합성한 회로보다 면적이 120 %, 140 % 증가된 제약조건하에서 실험을 수행하였으며, 그 합성 결과를 표 4 (b)에 나타내었다. 실험 결과에서 전력소모는 원래 회로의 면적을 증가시키지 않는 면적 제약조건하에서 18.4 %, 원래 회로 면적의 120 % 제약조건하에서 생성한 회로는 42.3 %, 원래 회로 면적의 140 % 제약조건하에서 생성한 회로는 56.1 %, 제약조건이 없이 생성된 회로는 61.1 % 전력 소모가 감소된 결과를 얻어, 면적 제약조건이 커짐에 따라 전력 소모가 적은 회로를 얻을 수 있음을 확인할 수 있었다. 제안된 알고리듬과 기존의 precomputation 알고리듬의 수행시간을 표 5에 나타내었으며 평균 수행 시간이 95.7 % 감소되었다. Precomputation 알고리듬은 크기가 작은 회로의 경우 수행시간이 제안된 알고리듬보다 적었으나, 회로의 크기가 커짐에 따라 수행시간이 방대하게 소요되는 결과를 얻었다.

표 5. Precomputation 설계 방식과 제안된 방식의 수행 시간 비교

Table 5. Runtime consumption of the precomputation-based scheme and the proposed scheme.

Circuits	Precomputation algorithm (sec.)	The proposed algorithm (sec.)	비교 (%)
b1	0	21	+ 20,900.0
cc	43,201	214	- 99.5
cm42a	4	36	+ 928.6
cm82a	0	28	+ 27,900.0
cm138	0	299	+ 149,400.0
cm150	117	110	- 6.0
cm162	1	97	+ 13,757.1
cmb	2	81	+ 4,400.0
count	68	288	+ 323.5
cordic	71	373	+ 428.3
cu	84	1,291	+ 1,446.1
c8	84,364	277	- 99.7
decode	106	91	- 13.8
duke2	36,275	2,173	- 94.0
e64	82,585	2,363	- 97.1
majority	0	76	+ 75,900.0
misex2	17,582	1,086	- 93.8
mux	1	592	+ 84,471.4
f51m	3	57	+ 1627.3
pcler8	5,957	936	- 84.3
sao2	0	373	+ 93,150.0
sct	904	95	- 89.5
tcon	4,808	280	- 94.2
x2	3	59	+ 1,934.5
z4ml	0	416	+ 103,900.0
9symml	1	272	+ 24,627.3
C1908	-	531	-
i9	-	5640	-
dalu	-	7080	-
Average	10,621	461	- 95.7

CMOS 회로의 경우 동작하는 래치의 전력 소모에 비해 disable된 래치의 전력 소모는 무시할 수 있으며, 제안된 알고리듬에 의해 원래 벤치마크 회로와 합성된 회로의 동작하는 래치수는 각각 (분할된 회로의 평균 입력 래치 개수 + 2) 와 (입력 래치의 개수) 로

대부분의 경우 합성된 회로의 래치에 의한 전력 소모가 적으므로 실험에서 회로의 래치 소모 전력은 생략하였다. 보다 정확한 소모 전력을 예측하기 위하여 기술 매핑(technology mapping) 후 전력을 예측하는 방법이 바람직하지만 기술 매핑 방법에 따라 회로의 전력 소모는 평균 18 %, 사용되는 라이브러리에 따라 9 % - 38 %의 편차가 있으며, 기술 매핑 후의 신호 전파 경로의 지연 시간 차이에 따라 15 % - 20 %의 glitch에 의한 전력 소모의 차이가 있다고 알려져 있다[2]. 제안된 회로 분할 알고리듬은 기술 독립 수준에서 노드의 전이 활동을 줄임으로써 전력 소모를 줄이는 알고리듬이므로, 전이 확률 감소에 따른 소모 전력 감소 효과를 가장 잘 나타낼 수 있는 기술 매핑 전 zero-delay 모델 환경에서 실험 결과를 비교하였다. 제안된 합성 알고리듬은 기술 독립적인 레벨에서 노드 전이 횟수를 줄이는 알고리듬으로 게이트의 지연 시간에 대한 정보가 없어 glitch나 hazard등의 spurious transition에 의한 전력 소모를 고려하기 어려우며, spurious transition에 의한 전력 소모는 기술 매핑 과정 후 path balancing 등의 최적화 과정을 통하여 줄일 수 있으며 이를 위한 연구가 필요하다.

실험 결과에서 벤치마크 회로 중 b1 회로, cm162 회로, decod 회로, majority 회로는 precomputation 설계 방식에 비하여 제안된 방식에 의해 생성된 회로의 전력 소모가 79 - 90 % 로 크게 감소되었다. 이것은 이 회로에 제안된 알고리듬을 적용할 때 회로의 스위칭 활동에 영향을 많이 미치는 입력을 이용하여 회로의 소모 전력이 크게 감소된 반면, precomputation 설계 방식을 적용할 경우 스위칭 활동을 줄일 수 있는 입력의 수가 적거나 precomputation 논리회로에 의한 추가 소모 전력이 스위칭 활동 감소에 따른 소모 전력보다 큰 경우에 전체 소모 전력이 크게 감소되지 않기 때문이다. 다수개의 출력에 공통적으로 사용되는 입력이 존재하는 회로는 입력 래치의 추가에 의한 면적 증가량보다 조합회로의 면적 감소량이 크므로 전체적인 면적이 감소되었으며, 분할된 회로에 중복되는 입력 래치가 많은 cm162, cm82a, z4ml 등의 회로는 면적이 증가하였다. z4ml 등의 회로는 입력이 스위칭 활동에 많은 영향을 미치지 못하는 구조이므로 전력 소모 감소량이 상대적으로 적었다. tcon 회로는 분할된 cofactor 회로가 항등 함수(identity function)이므로 조합회로의 전력소모가 0 μW 인 결과를 얻었다. Precomputation 알고리듬으로 합성한 회로는 precomputataion 회로의 추가에 따라 지연시간이 증가하는 단점이 있으나 제안된 알고리듬에 의해 생성된 회로는 분할된 cofactor 회로로 구성되어 있어 대부분의 경우 원래 회로에 비해 지연시간이 증가하지 않는 장점이 있다. MCNC 벤치마크 회로에 대한 실험을 통하여 제안된 회로 분할을 이용한 합성 알고리듬이 저전력 회로 설계에 보다 효율적임을 확인하였다. 한편, 제안된 알고리듬의 수행시간은 실행할 때마다 일정하지 않아 Ultra Sparc I에서 수행한 시간 측정치의 평균값을 나타낸다.

V. 결 론

제안된 논문은 면적 제약조건하에서 저전력 소모 회로 설계를 위한 회로 분할 알고리듬을 제안하고 구현하였다. 제안된 회로 분할 알고리듬은 동작에 불필요한 신호값의 전이를 감소시켜 소모 전력을 줄일 수 있도록 입력 조건에 따라 수행할 세부 회로를 결정하고, 이에 따라 다음 클럭에 여러 개의 세부 회로 중 하나가 선택되어 수행되는 구조의 회로를 합성한다. 제안된 알고리듬은 전력 소모를 크게 줄일 수 있으나 면적의 증가를 초래한다. 구현된 시스템은 면적 제약조건하에서 소모 전력이 최소인 회로를 얻을 수 있도록 한다. 제안된 알고리듬은 adaptive simulated annealing 알고리듬을 이용하여 가능한 분할을 탐색하고 입력을 선택한다. 탐색 과정 중 분할된 상태는 트리로 나타내었으며 탐색 공간과 메모리 사용을 줄이기 위하여 중복된 분할을 시도하지 않도록 설계하였다. 제안된 알고리듬에 의해 생성된 회로는 분할된 cofactor 회로의 depth가 합성전 회로보다 감소하여 일부 selection logic이 큰 영향을 미치는 작은 회로 외에는 지연시간이 증가하지 않는 장점이 있으며, 합성된 결과 회로의 래치수는 증가하지만 동작하는 래치수는 원래 회로보다 적으므로 래치에 의한 소모 전력이 감소하는 특징이 있다. 제안된 시스템은 다수의 출력 함수에 영향을 미치는 입력 집합이 존재하는 경우 더욱 효율적인 결과를 얻었으며, 회로의 입력이 대칭적이거나 반복되는 구조의 경우 선택된 입력 집합에 영향을 받는 회로 영역이 적어 전력 감소량이 상대적으로 적게 나타난다. MCNC 벤치마크 회로에 대한 실험을 통하여 제안된 저전력 회로 합성 알고리듬이 기존의

precomputation 구조에 비하여 소모 전력이 적은 회로를 합성함을 확인하였다. 추후 과제로 수행 시간을 개선하기 위하여 대칭 입력(symmetric inputs)을 이용하여 분할된 회로들은 동일한 구조를 가지므로 대칭 입력을 동일한 입력으로 간주하여 탐색 공간을 줄이는 방법과 다중 분할시 회로의 중복되는 영역을 줄임으로써 면적이 증가하는 단점을 개선하기 위한 연구가 필요하다.

감사의 글

※ 본 논문은 1997년도 서강대학교 교내 연구비 지원에 의해 이루어졌음.

참 고 문 헌

[1] S. Devadas, S. Malik, "A Survey of Optimization Techniques Targeting Low Power VLSI Circuits," in Proc. 32nd DAC, pp. 242-247, June 1995.

[2] A. Chandrakasan, T. Sheng, and R. Brodersen, "Low Power CMOS Digital Design," Journal of Solid State Circuits, vol. 27, no. 4, pp. 473-484, April 1992.

[3] J. Monteiro, S. Devadas, and B. Lin, "A Methodology for Efficient Estimation of Switching Activity in Sequential Logic Circuits," in Proc. 31st DAC, pp. 12-17, June 1994.

[4] R. Mehra, J. Rabaey, "Behavioral Level Power Estimation and Exploration," in Proc.Int'l Workshop on Low Power Design, pp. 197-204, April 1994.

[5] A. Shen, S. Devadas, A. Ghosh, and K. Keutzer, "On Average Power Dissipation and Random Pattern Testability of Combinational Logic Circuits," in Proc. ICCAD, pp. 402-407, Nov. 1992.

[6] C. Lemonds, S. Shetti, "A Low Power 16 by 16 Multiplier Using Transition Reduction Circuitry," in Proc. Int'l Workshop on Low Power Design, pp. 139-142, April 1994.

[7] K. Roy, S. Prasad, "SYCLOP: Synthesis of CMOS Logic for Low Power Applications," in Proc. ICCD, pp. 464-467, Oct. 1992.

[8] M. Alidina, J. Monteiro, S. Devadas, and A. Ghosh, "Precomputation-Based Logic Optimization for Low Power," in Proc. ICCAD, pp. 74-81, Nov. 1994.

[9] S. Iman, M. Pedram, "Multi-Level Network Optimization for Low Power," in Proc. ICCAD, pp. 371-377, Nov. 1994.

[10] S. Iman, M. Pedram, "Logic Extraction and Factorization for Low Power," in Proc. 32nd DAC, pp. 248-253, June 1995.

[11] I. S. Choi, H. Kim, D. W. Seo, and S. Y. Hwang, "Kernel-based Precomputation Scheme for the Design of Low Power Combinational Circuits," Electronics Letters, IEE, vol. 32, no. 14, pp. 1281-1283, July 1996.

[12] F. Najm, "Power Estimation Techniques for Integrated Circuits," in Proc. ICCAD, pp. 492-499, Nov. 1995.

[13] A. Ghosh, S. Devadas, K. Keutzer, and J. White, "Estimation of Average Switching Activity in Combinational and Sequential Circuits," in Proc. 29th DAC, pp. 253-259, June 1992.

[14] S. Kirkpatrick, C.Gelatt, and M. Vecchi, "Optimization by Simulated Annealing," *Science,* vol. 220, pp. 671-680, May 1983.

[15] B. Rosen, "Function Optimization Based on Advanced Simulated Annealing," in Proc. IEEE workshop on Physics and Computation, pp. 289-293, 1992.

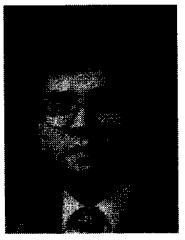
[16] L. Ingber, 'Adaptive simulated annealing (ASA): Lessons learned,' Journal of Control and Cybernetics, vol 25, pp. 33-54, Jan. 1996.

[17] E. Sentovich, H. Savoj, R. Brayton, and A. Sangiovanni-Vincentelli, 'SIS: A System for Sequential Circuit Synthesis,' Memorandum No. UCB/ERL M92/41, Electronic Research Laboratory, University of California, Berkeley, May 1992.

저 자 소 개

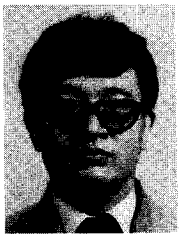
崔 益 成(正會員)

1992년 2월 서강대학교 전자공학과 졸업. 1994년 2월 서강대학교 전자공학과 공학 석사 취득. 1994년 2월 ~ 현재 동 대학원 박사과정 재학 중. 주관심분야는 CAD 시스템, Synthesis for lowpower, Computer Architecture 및 VLSI Testability 등임



黃 善 泳(終身會員)

1976년 2월 서울대학교 전자공학과 졸업. 1978년 2월 한국과학원 전기 및 전자공학과 공학석사 취득. 1986년 10월 미국 Stanford 대학 전자전산공학 박사 학위 취득. 1976년 ~ 1981년 삼성반도체 주식회사 연구원, 팀장. 1986년 ~ 1989년 Stanford 대학 Center for Integrated Systems 연구소 책임연구원. Fairchild Semiconductor Palo Alto Research Center 기술자문. 1989년 ~ 1992년 삼성전자(주) 반도체 기술자문. 1989년 3월 ~ 현재 서강대학교 전자공학과 교수. 주관심분야는 CAD 시스템, Computer Architecture 및 DSP Systems Design, VLSI 설계 등임



金 衡(正會員)

1979년 2월 서강대학교 전자공학과 졸업. 1981년 2월 서강대학교 전자공학과 공학석사 취득. 1997년 8월 서강대학교 전자공학과 공학박사 학위 취득. 1983년 11월 럭키 금성 그룹 기획 조정실 근무. 1985년 1월 럭키 엔지니어링 근무. 1987년 3월 금서 소프트웨어 근무. 1992년 3월 ~ 현재 경민 전문대학교 전자계산학과 조교수