저전력설계를 위한 공통 표현의 추출
(Extraction of Common Expressions for Low Power Design)

황 민 † 정 미 경 †† 이 귀 상 †††
(Min Hwang) (Migyoung Jeong) (Gueesang Lee)

요 약 본 논문에서는 논리합성 단계에서의 전력최소화를 위한 새로운 전력소모함수를 제안하고 이의 공통표현추출 과정에의 적용방법에 대해 기술한다. 제안된 새로운 전력소모표현은 노드의 표현 및 구현이 복합게이트(complex gate)로 이루어진다는 가정아래 각 노드에서의 정전용량(capacitance)과 그 스위칭 활동량(switching activity)을 반영하되 정전용량은 노드의 입력 수에 비례한다고 가정한다. 공통 표현 추출, 즉 커널(kernel) 과 큐브(cube) 추출은 사각형 커버링(rectangle covering) 문제로 변환될 수 있으며 본 논문에서는 이러한 과정에서 각 노드의 전력소모 표현을 어떻게 이용하는지 기술하고 실험을 통해 SIS-1.2의 결과와 비교한다.

Abstract In this paper, we propose a new method for power estimation in nodes of multi-level combinational circuits and describe its application to the extraction of common expressions for low power design. Extracting common expressions which is accomplished mostly by the extraction of kernels and common cubes, can be transformed to the problem of rectangle covering. We describe how the newly proposed estimation method can be applied to the rectangle covering problem and show the experimental results with comparisons to the results of SIS-1.2.

1. 서 론

오늘날 저전력화는 거의 모든 전자제품에 공통적으로 필요한 사항이 되어 있으며 그러한 예는 거의 모든 축전지에 의하여 동작하는 기기 들, 즉 손목시계, 계산기, 이동전화기, 보청기, 보병용 휴대용 군사 기기, 뿐만 아니라 각종 컴퓨터시스템을 비롯한 일반 가전제품에 이르기까지 많은 경우에서 찾아볼 수 있다. 더욱이 오늘날과 같이 정보화에 대한 인식과 활용이 보편화되면서 발생하는 개인 휴대용 컴퓨터를 비롯한, 이동 통신이나 개인 휴대통신의 폭발적인 수요를 고려할 때, 관련제품의 경쟁력을 결정하는 소형화, 경량화의 문제를 해결하기 위해서 저전력 설계방법은 반도체 설계 시 필수적인 고려사항으로 인식되고 있다.

이러한 현실적인 요구로 인하여 지난 수년간 저전력 반도체설계방법에 대한 연구가 매우 활발히 진행되었으나 어떤 단 하나의 방법이 전력소모의 문제점을 단번에 해결해 주지는 못하였다. 저전력 설계의 효과를 얻기 위해서는 소자 및 회로 수준, layout 수준, 논리수준 그리고 구조 및 알고리즘 수준의 모든 설계단계에서 전력소모가 고려되어야 하며 본 논문에서는 반도체 설계 계층(design hierarchy)의 여러 단계 중에서 논리합성단계에서의 저전력 설계 방법을 기술하며 특히 논리합성방법의 기초가 되는 불리안 함수에서의 공통표현추출에 대하여 연구한다.

본 논문에서는 공통표현 추출에 있어서 어떻게 전력소모를 비용함수로 적용하는지 분석하여 이를 합성도구로 구현한다. 이러한 공통 표현 추출에는 커널과 큐브 추출이 주된 역할을 하며 구체적으로 SIS-1.2[1]에서의 논리합성방법을 개선하여 그 결과를 기존의 방법과 비교한다. 이러한 커널이나 큐브의 추출은 사각형 커버링(rectangle covering) 문제로 변환되며 본 논문에서는

· "본 연구는 '97년도 교육부 반도체분야 학술연구조성비(ISRC 97-E-2030)에 의하여 연구되었음"
† 비 회 원 : 전남대학교 전산학과, 기초과학연구소
hwang@cad.chonnam.ac.kr
†† 정 회 원 : 전남대학교 전산학과, 기초과학연구소
mgjung@chonnam.chonnam.ac.kr
††† 종신회원 : 전남대학교 전산학과, 기초과학연구소 교수
gslee@chonnam.chonnam.ac.kr
논문접수 : 1999년 5월 13일
심사완료 : 1999년 10월 25일

사각형 커버링에서의 전력소모 구현방법을 제시한다. 일반적인 회로의 구성은 CMOS로 구성된다고 가정하며 이 때 전력소모는 출력노드의 정전용량이고 이것은 스위칭 활동량에 의하여 좌우된다. 전력소모 표현식은 각 노드에서의 정전용량으로 표현되고 그 스위칭 활동량을 반영하되 정전용량은 노드의 입력 수에 비례한다고 가정하였으며 스위칭 활동량은 BDD를 구성하여 계산하는 방법을 사용하였다. 또한 스위칭동작의 분석에 있어서 입력 상호간의 관계(correlation)[13]는 고려하지 않고 단순히 확률에 의한 스위칭 동작 추정방법[2-4,12]을 사용하였다.

다단계회로에서의 저전력 설계를 위한 논리합성에 대한 연구는 지금까지 많은 연구가 이루어져 왔다. 그 중에서 [2]는 각 노드가 factored form으로 구성된다고 가정하고 이러한 가정아래에서 공통표현이 추출되는 노드의 입력에서의 스위칭 활동량의 변화와 내부 노드의 변화를 추정한다. 그러나 이 방법의 결점은 그 노드가 반드시 사용된 Factored form의 형태로 구현되는 것이 아니라는 점이다. 따라서 실제로 노드가 구성되는 형태와 전력소모 예측시의 노드 형태가 다르게 됨으로써 잘못된 계산을 할 가능성이 높은 것이다. 이를 개선하기 위하여 [3,4]에서는 각 노드의 표현이 SOP(Sum of Products) 형태의 이단계(two-level) AND/OR회로로 구현된다고 보는 것이다. 그러나 이 방법은 각 큐브들이 하나 하나의 AND 게이트 노드로 구성된다고 가정함으로써 더욱 간소화된 표현의 가능성을 고려하지 않으므로 각 큐브들로 인하여 과다한 전력소모예측을 하게 된다는 점이다. 예를 들어 4 입력 parity함수를 고려해보자. 이 표현은 EXOR (Exclusive-OR) 게이트를 사용하면 매우 간단하게 표현할 수 있으나 반면에 AND/OR 게이트들을 사용한다면 매우 많은 수의 큐브들을 요구하게 된다. 실제로 두 가지 구현형태에 대한 전력소모 예측은 매우 다르며 대개 기술사상(technology mapping)이후에는 EXOR 게이트 또는 그와 유사한 구조로 구현된다는 점을 고려할 때[5,6], AND 게이트로 이루어진 큐브들을 모두 전력소모예측에 고려하는 것은 정확한 것이 아니라는 것을 알 수 있다. 따라서 본 논문에서는 이러한 형태의 단점을 개선하기 위하여 각 노드의 표현으로 SOP형태를 이용하지만 이 표현이 이단계 AND/OR 회로로 구현되는 것이 아니고 하나의 복합게이트(complex gate)로 이루어진다고 가정한다. 임의의 노드가 하나의 복합게이트로 구성되면 각 노드를 factored 형태로 가정하는 경우[2]나 또는 AND/OR 회로로 구성된다는 가정[3,4]에 의한 경우와 비교하여 위에서 언급한 단점들을 보완할 수 있고 실제로 기술사상이 이루어지면 각 노드는 하나의 논리소자로 구현되므로 이러한 가정은 최종적인 실제 구현과 일치한다

본 논문에서는 global BDD를 이용한 zero delay model 의 전력소모 예측기법을 사용하며 필요에 따라 본 연구결과를 SIS-1.2의 simulation에 의한 real delay model로 확장할 수 있을 것이다. 커널과 공통 큐브를 추출하는 과정에 대한 자세한 기술은 생략한다[8]. 본 논문의 구성은 다음과 같다. 2장에서는 저전력 설계를 위해 이를 어떻게 사각형 커버링에 적용하여 논리합성을 실행하는지를 설명하고 3장에서 이의 실험 결과를 보이며, 마지막으로 결론을 제시한다.

2. 저전력 설계를 위한 공통 표현추출

본 논문에서는 전력소모 예측을 각 노드에서의 정전용량과 스위칭 활동량을 고려하여 수행한다. 본 논문의 초점은 공통표현을 추출하는데 있어서 사각형 커버링을 적용함에 있어서 이러한 정전용량과 스위칭 활동량을 어떻게 이용하는가 이다. 기본적으로 본 논문은 입력변수 load의 변화를 스위칭 활동량과 함께 적용하는 것이다. 임의의 함수 F의 커널 추출 시, 정전용량을 고려하지 않는 경우를 먼저 생각해보면, 전력소모 감소량을 다음과 같은 식으로 예측할 수 있다.

$$\begin{aligned}&(R-1)\sum_{i=0}^{M} t(v_i)CL(v_i,D)+\\&(P-1)\sum_{i=0}^{R}\sum_{j=0}^{M} t(v_i)CL(v_i,q_i)-Rt(D)\end{aligned} \tag{1}$$

여기서 $\{v_1,\cdots,v_M\}$은 입력변수들이고 $D=d_1+\cdots+d_P$는 주어진 논리함수 F을 나누는 커널 이라 하자. 그리고 $Q=\{q_1,\cdots,q_R\}$은 커널 D에 대한 co-kernel 큐브들이며 $CL(v,q)$는 논리식 q의 SOP 표현에서 나타나는 입력 v의 수(cube-load)이다. 임의의 내부노드 n의 신호확률(signal probability)을 $p(n)$이라 할 때, 노드 n의 스위칭 활동량 $t(n)$은 다음과 같이 구해진다.

$$t(n)=2*p(n)*(1-p(n))$$

식 F을 식으로 표현하면 다음과 같이 쓸 수 있다.

$$\begin{aligned}F &= (d_1+\cdots+d_P)(q_1+\cdots+q_R)+\cdots\\ &= q_1D+q_2D+\cdots+q_RD+\cdots\end{aligned} \tag{2}$$

여기에서 새로운 노드 D를 만든다면 상당히 많은 큐브들이 없어짐을 알 수 있다. 즉, $D=d_1+\cdots+d_P$ 의 큐브들이 전부 없어지며 새로운 노드 D을 만들 때 단 하나만 필요하므로 결과적으로 (R-1) 개의 D 표현이 없

어진다. 또한 $Q=\{q_1,\cdots,q_R\}$ 표현도 (P-1)개가 없어짐을 알 수 있다. 이제 위와 같이 새로운 노드 D을 만들어 이를 식(2)과 같이 대입한다고 가정하면 전력 소모식 (1)에서 첫 항은 커널 D을 도입함으로써 R개의 논리표현 D가 없어지며 새로운 노드(커널에 대한)가 생겨나므로 결과적으로 (R-1)개의 표현이 없어지므로 발생하는 D의 표현에서 일어나는 리터럴들의 스위칭 활동량을 더한 것임을 알 수 있다. 그리고 둘째 항은 없어지는 (P-1)개의 co-kernel 큐브들에 대한 스위칭 활동량의 합임을 알 수 있다. 셋째 항은 새로 도입된 R개의 새로운 노드에 대한 리터럴들의 스위칭 활동량이다. 다음과 같은 예를 들어 보자.

$$F = af + bf + ag + cg + ade + bde + cde$$
$$G = af + bf + ace + bce$$

이 논리식들이 다음과 같이 변화될 수 있다.

$$\begin{aligned} F &= deX + fX + ag + cg + cde \\ G &= ceX + fX \\ X &= a + b \end{aligned} \tag{3}$$

여기에서 커널 X가 추출됨으로써 커널 큐브들의 개수 P=2 이고 co-kernel 큐브들의 개수 R=4이며 $a+b$표현이 (R-1)개 즉 3개가 없어졌으며, co-kernel 큐브들은 P-1=1개가 없어졌음을 알 수 있다.

본 논문에서 제안한 식(1)에 의한 위의 방법을 좀 더 개선하여 없어지는 각 리터럴들에 대하여 단지 스위칭 활동량만을 계산하여 더하지 않고 트랜지스터 사이징을 고려하여[9-11] 이 입력변수들의 정전용량(CMOS 회로에서의 transistor 면적)이 그 노드에서 사용하는 입력변수의 수와 비례한다고 가정하고 이를 각 노드의 무게(weight)로 한다. 즉 임의의 회로에서 트랜지스터 면적과 회로의 동작속도가 비례하므로 하나의 노드에 같은 지연시간을 유지하기 위해서는 입력변수의 수에 따라 트랜지스터 면적을 증가시킨다고 가정하는 것이다. 그러나 어느 정도 트랜지스터 면적이 커지면 더 이상 동작속도는 빨라지지 않으므로 실제 구현에서는 최소크기의 4배 이상은 허용하지 않는다.(편의상 여기서는 설명을 위해 트랜지스터 면적은 입력 수에 정비례한다는 가정 아래 기술함)

위의 예에서는 입력변수 a 의 정전용량이 식 F와 식 X에서 서로 다르게 계산되어야 한다. 예를 들어 식(3)에서 식 F의 무게는 7(입력변수 = $\{a,c,d,e,f,g,X\}$) 인데 반해 식 X의 무게는 2 (입력변수 = $\{a,b\}$)임을 알 수 있다. 실제 회로 설계에서 입력 변수가 많은 경우 이에 의한 트랜지스터(transistor)의 크기가 달라지며 이는 전력소모의 크기에 바로 영향을 미침을 앞에서 알아보았다. 여기서는 이러한 정전용량을 각 노드에서의 입력변수의 크기에 비례한다고 보고 이를 고려하여 전력소모식을 설정한다.

$$\begin{aligned} & R\sum_{i=0}^{M} t(v_i)w(D)CL(v_i,D) + \\ & (P-1)\sum_{i=0}^{R}\sum_{j=0}^{M} t(v_i)w(q_i)CL(v_i,q_i) - \\ & Rw(D)t(D) - \sum_{i=0}^{M} t(v_i)w(D)CL(v_i,D) \end{aligned}$$

여기서 $w(K)$는 논리식 K가 속한 노드의 무게(weight)를 말하며 본 논문에서는 이 무게함수로 그 노드의 입력 수를 사용한다. 위의 식에서 첫째항의 경우 D의 표현이 소속된 노드는 새로운 노드를 적용하기 전의 원래의 식이고 네 번째 항의 D는 새로운 노드가 형성된 후의 이 노드의 표현을 말한다. 그리고 둘째 항에서의 $w(q_i)$도 큐브 q_i가 소속된 노드, 즉 원래의 식에서의 무게를 말한다. 세 번째 항은 새로운 노드의 출력변수를 원래의 식에서 입력변수로 사용할 때의 경우를 나타낸다. 그러므로 이 때의 무게도 원래의 식의 무게를 말하며 이러한 무게에 따라 스위칭 활동량의 합계가 달라짐을 알 수 있다. 예를 들어 식(3)에서 식 F의 입력변수의 무게는 7이며, 식 G의 무게는 4, 그리고 식 X의 무게는 2임을 알 수 있다. 물론 여기에서 새로운 노드를 대입한 후의 입력변수의 수는 그 전의 입력변수의 수와는 다르다. 그러므로 여기에서의 문제점은 새로운 노드를 대입하기 전에 새로운 변수를 사용한 노드의 입력 수를 예측하기가 어렵다는 것이다. 이를 정확히 계산하기 위해서는 원래의 식을 새로운 노드(커널)로 나누어야 하며 이는 상당한 cost를 요구할 것으로 생각되며, 본 논문에서는 근사적인 방법으로 새로운 노드를 대입하기전의 노드의 무게를 사용하며 이것이 변화될 때마다 바로 무게함수를 바꾸어 다음에 적용될 때는 새로운 무게값을 사용하도록 한다. 이를 사각형 커버링에 적용하기 위한 방법을 다음과 같이 기술한다.

```
rectangle value를 계산하기 위한 자료구조
1. rectangle 의 각각의 co-kernel cubes에 대해
   row_cost[]    - 리터럴수
   row_weight[] - 큐브를 포함한 노드의 fanin 수
   row_power[]  - switching activity
2. rectangle 의 각각의 kernel cubes에 대하여
   col_cost[]    - 리터럴 수
   col_power[]  - switching activity
```

그림 1 rectangle value를 계산하기 위한 자료구조

임의의 사각형 $R=(R,C)$을 고려해보자. 이 행렬의 각 열은 커널 큐브들을, 각 행은 이 커널에 대한 co-kernel 큐브들에 관련되어 있다. 이 사각형에서 위의 전력소모식을 적용하기 위하여 다음과 같은 리스트들을 사용한다. 기본적으로 각각의 커널, co-kernel 큐브들에 대하여 이 큐브들의 크기뿐 아니라 이 큐브들이 소속된 노드의 무게를 기록하며 또한 그 입력변수들의 스위칭 활동량을 기록한다.

그림 2는 임의의 사각형에 대해 적용되는 자료를 나타낸다. p(k)는 논리식 또는 큐브 k 에 대한 스위칭 활동량을 나타내며 w(k)는 논리표현 k가 포함된 노드의 무게를 나타낸다.

	$p(d_1)$	$p(d_2)$	...	$p(d_P)$
$p(q_1)$	$w(q_1)$			
$p(q_2)$	$w(q_2)$			
...	...			
$p(q_R)$	$w(q_R)$			

그림 2 임의의 사각형에 대해 적용되는 자료

	$p(d_1)$	$p(d_2)$	...	$p(d_P)$
$p(q_1)$	$w(q_1)p(q_1)$, $w(q_1)p(d_1)$	$w(q_1)p(q_1)$, $w(q_1)p(d_2)$		$w(q_1)p(q_1)$, $w(q_1)p(d_P)$
$p(q_2)$	$w(q_2)p(q_2)$, $w(q_2)p(d_1)$	$w(12)p(q_2)$, $w(q_2)p(d_2)$		$w(q_2)p(q_2)$, $w(q_2)p(d_P)$
...		...		
$p(q_R)$	$w(q_R)p(q_R)$, $w(q_2)p(d_1)$	$w(q_R)p(q_R)$, $w(q_2)p(d_2)$		$w(q_R)p(q_R)$, $w(q_2)p(d_P)$

그림 3

그림 3에서 $(d_1+\cdots+d_P)(q_1+\cdots+q_R)=Dq_1+\cdots+Dq_R$ 와 같이 논리식이 변화하므로 없어지는 큐브들의 무게와 스위칭 활동량의 곱은 다음과 같이 나타낼 수 있다.

$$(P-1)\sum_{i=1}^{R} w(q_i)p(q_i)+W\sum_{i=1}^{P} p(d_i),$$
$$W = w(q_1)+\cdots+w(q_R)$$

와 같이 변화하므로 그리고 새로 생겨나는 노드와 리터럴들의 스위칭 활동량과 무게의 곱은 새로 생겨나는 커널에 대한 노드와 이 노드의 입력에서 생기는 전력소모를 고려할 수 있다. 그러므로 이는 다음과 같이 나타낼 수 있다.

$$w(D)\sum_{i=1}^{R} p(d_i)+W\,p(D)$$

여기서 첫째 항은 커널의 입력에 대한 전력소모로서 커널의 무게에 각 큐브의 스위칭 활동량을 곱한것이며, 둘째 항은 새로 도입된 노드의 리터럴이 각각의 co-kernel 에 대해 적용되므로 새 노드인 커널 큐브들에 대해 커널의 무게를 곱한 것이다. 그러므로 다음과 같이 최종적인 전력소모식을 만들 수 있다.

$$(P-1)\sum_{i=1}^{R} w(q_i)p(q_i)+W\sum_{i=1}^{P} p(d_i)-$$
$$w(D)\sum_{i=1}^{R} p(d_i)-W\,p(D)$$

위의 자료구조와 전력소모식을 이용하여 사각형 커버링에 본 논문의 전력소모식을 적용하는 알고리즘은 그림 4와 같다.

```
/*  커널 추출을 위한 rectangle 값의 계산,
                        rect 는 커널을 나타낸다. */
value(rect)  /* rect is a rectangle */
{
  R=rect->rows->length; /* no. of cokernels */
  P=rect->cols->length; /* no. of kernels */

  /* calculate switching activity & weight in co-kernel cubes */
  cks_p = sum_node_weights = 0;
  foreach_col_element(p = rect->rows) {
            weight = row_weight[p->row_num];
            cks_p += row_power[p->row_num] * weight;
            sum_node_weights += weight;
  }
  /* calculate switching activity in kernel cubes */
  ks_p = 0;
  foreach_row_element(p = rect->cols) {
            ks_p += col_power[p->col_num];
  }
  /* compute the real rectangle value */
  kernel_power = node_power(rect->kernel);
  if(weight is not considered ) {
    rect-> value =
            (R-1) * ks_p + (P-1) * cks_p - R * kernel_power;
  } else {
            /* signal prob.  of the kernel */
            v1 = sum_node_weights * ks_p ;
            v2 = (P-1) * cks_p;
            v3 = sum_node_weights * kernel_power;
            v4 = kernel_weight * ks_p ;
            rect->value = v1 + v2 - v3 -v4;
  }
  return rect->value;
}
```

그림 4 커널 추출 알고리즘

전력소모 감소를 위한 커널 추출은 다음과 같이 진행된다.

1. 모든 커널 set K를 계산한다.
2. K를 이용한 모든 커널 intersection D를 계산한다.

3. 여기서 최대 전력감소를 가져올 부분 표현 $D_i \in D$ 을 선택한다.

3. 공통 큐브 추출

이제 큐브(곱항) 추출에 대해서 알아보자. 임의의 다출력함수 $F=(f_1,\cdots,f_L)$과 그 큐브세트 $(C_1,\cdots,C_N)$ 그리고 입력 세트 $(v_1,\cdots v_M)$에 대해 $C=c_i \cap c_j$ 는 T개의 리터럴을 갖고 R(>1)개의 큐브들에 공통으로 사용된다고 하자. 여기서 첫째 항은 큐브를 반복할 필요가 없음으로 인한 리터럴 수 감소이고 두 번째 항은 새로운 게이트의 도입으로 인한 리터럴 수 감소이다. 이 게이트의 load를 CL(f,C)라 하고, 이를 이용하여 다음과 같은 전력소모식을 만들어 볼 수 있다[4].

$$\left((R-1)\sum_{i=0}^{M} t(v_i)CL(v_i,C)\right)-Rt(C)$$

본 논문에서는 커널 추출과 마찬가지로 무게함수와 스위칭 활동량을 적용하기 위해 다음과 같은 전력소모식을 제안한다. 먼저 다음과 같이 무게함수를 사용한 식을 쓸 수 있다.

$$\left(R\sum_{i=0}^{M} w(v_i)t(v_i)CL(v_i,C)\right)-$$
$$Rw(C)t(C)-\sum_{i=0}^{M} w(v_i)t(v_i)CL(v_i,C)$$

따라서 위 식은 무게함수를 다음과 같이 더하여서 사용할 수 있다. 여기서도 마찬가지로 큐브들의 부분큐브 중에서 가장 전력감소가 큰 것을 선택한다. 여기서 W는 이 공통 큐브들이 포함된 노드들의 무게를 더한 것이며 $w(C)$는 공통큐브의 리터럴 수이다.

$$\left(W\sum_{i=0}^{M} t(v_i)CL(v_i,C)\right)- \quad (4)$$
$$Wt(C)-w(C)\sum_{i=0}^{M} t(v_i)CL(v_i,C)$$

위의 전력소모식을 사각형에 적용하는 것은 앞에서 제시한 커널추출과 유사하므로 자세한 설명은 생략하기로 한다. 위 식의 적용알고리즘은 그림 5와 같다.

4. 실험결과

본 논문의 실험을 위하여 위에 제시한 방법들을 SIS-1.2에 구현하여 그 결과를 비교하였다. 주로 본 논문에서 제안한 전력소모식을 SIS-1.2에 적용하고 이를 위해 사각형 커버링의 전력소모 계산방법을 구현하였다. 실험에는 MCNC benchmark의 조합회로들을 이용하였으며 먼저 pla 형식으로 읽어서 이를 gkx 또는 gcx 등의 명령에 적용하였다. 실험결과의 전력소모는 SIS-1.2에서의 20MHz clock, Vdd=5V를 가정하고 마이크로 와트(μW)로 계산하였다.

```
/*  큐브추출을 위한 rectangle 의 값 계산. */
value(rect) /* rect 는 common cube를 나타냄 */
{
    /* calculate sum of weights */
    sum_node_weights = 0;
    foreach_col_element(p = rect->rows) {
            weight = row_weight[p->row_num];
            sum_node_weights += weight;
    }
    /* calculate switching activity in cube literals */
    cube_p = 0;
    foreach_row_element(p = rect->cols) {
            cube_p += col_power[p->col_num];
    }
    /* compute the real rectangle value of the cube */
    cube_power = node_power(rect->cube);
    /* signal prob. of the common cube */
    v1 = sum_node_weights * cube_p ;
    v2 = sum_node_weights * cube_power;
    v3 = cube_weight * cube_p ;
    /* cube_weight = node_num_fanin(cube) */
    rect->value = v1 - v2 - v3;
    return rect->value;
}
```

그림 5 공통 큐브 추출 알고리즘

다음은 커널 추출을 위해 사용한 script는 그림 6에서 보여주며 여기에서 사용하는 다른 방법들은 gkx의 option으로 구현하여 서로 구분하였다.

```
read_pla a.pla
sim2    /* simplify SOP */
gkx -b /* kernel 추출, 본 논문의 경우 -p option 으로 구분함 */
ps      /* 리터럴 수의 계산 */
pe -S   /* power estimate */
```

그림 6 커널 추출을 위한 SIS-1.2 script

표 1은 커널 추출에 있어서의 본 논문의 구현결과를 보여준다. "*1" 은 본 논문에서 제시한 방법이며 "sis"는 Sis-1.2의 결과를 나타낸다. 전체적으로 리터럴수의 총합은 약 1% 정도 증가한 반면, 전력소모는 약 8% 정도 감소하였음을 보여준다. 그러나 연산시간은 아직까지는 매우 많음을 알 수 있다. 이것은 스위칭 활동량 계산을 위해 전체 회로의 BDD(Binary Decision Diagram)을 계산 시마다 매번 계산함으로 인한 것이다. 본 논문

에서는 이러한 BDD를 사용함에 있어서의 연산시간감소보다는 적절한 전력소모식의 제시와 저전력 설계를 위한 논리합성에 주안점을 두었으므로 상대적으로 연산시간감소에 주력하지 않았다. 저전력 설계에 있어서의 연산시간감소와 사용공간감소의 분야는 또 다른 연구분야로서 많은 연구가 진행되고 있으며 본 연구의 방법에 대한 연산시간은 추후 연구에 의해 상당히 줄어들 수 있을 것으로 예상된다.

표 1 저전력설계를 위한 커널 추출에서의 리터럴과 전력소모 증감

name	#literals			Power Consumption			Time(sec)	
	sis	*1	비율	sis	*1	비율	sis	*1
5xp1	155	157	1.01	775.2	767.2	0.98	1.0	5.7
apex6	1172	1181	1.00	5569.2	5151.6	0.92	14.6	245.6
apex7	363	372	1.02	1683.6	1578.9	0.93	10.8	42.6
b1	12	12	1.00	40.3	33.4	0.82	0.3	0.4
b9	163	164	1.00	721.9	697.2	0.96	1.4	7.5
bw	187	187	1.00	868.5	826.3	0.95	3.2	6.9
c8	189	190	1.00	926.2	827.7	0.89	0.8	6.5
cht	230	230	1.00	1234.7	937.2	0.75	1.0	6.8
cm138a	31	31	1.00	87.2	63.6	0.72	0.6	1.7
cm151a	30	30	1.00	141.6	132.8	0.93	0.4	0.9
cm152a	28	28	1.00	137.5	122.5	0.89	0.4	0.8
cm42a	34	34	1.00	110.7	80.7	0.72	0.6	1.7
cm82a	34	34	1.00	160.6	123.4	0.76	0.4	1.4
cm85a	65	65	1.00	279.3	272.1	0.97	0.6	3.7
con1	23	23	1.00	127.7	107.1	0.83	0.4	0.6
rd73	117	119	1.01	528.6	494.9	0.93	1.3	20.6
sao2	180	193	1.07	821.3	804.3	0.97	1.2	13.2
vg2	99	99	1.00	465.5	446.4	0.95	1.3	2.9
Sum	3112	3149	1.01	14679.6	13467.3	0.91	40.3	369.5

다음은 공통 큐브를 위해 사용한 script는 그림 7과 같다. 그리고 표 2는 큐브 추출에 있어서의 실험결과이다. 여기에서도 커널 추출과 비슷하게 리터럴 수의 총합은 약 1%정도 증가한 반면, 전력소모는 약 8%정도 감소하였음을 보여준다.

```
collapse
sim2
gkx -b
gcx   /* 본 논문의 경우 -p option으로 구분함 */
ps
pe -S
```

그림 7 공통 큐브 추출을 위한 Sis-1.2 script

표 2 공통큐브 추출에서의 리터럴과 전력소모 증감

name	# literals			Power Consumptions(uW)			Time(sec)	
	sis	*1	비율	sis	*1	비율	sis	*1
5xp1	165	165	1.00	812.4	716.9	0.88	1.1	3.9
apex6	928	956	1.03	3761.7	3641.3	0.96	16.0	80.8
apex7	355	357	1.00	1573.4	1434.3	0.91	13.0	20.3
b1	12	12	1.00	40.3	40.3	1.00	0.3	0.4
b9	170	170	1.00	738.7	670.8	0.90	1.5	3.8
bw	182	183	1.00	814.0	749.5	0.92	3.2	6.1
c8	149	149	1.00	614.3	562.9	0.91	0.8	2.3
cht	165	165	1.00	757.7	757.7	1.00	1.0	2.1
cm138a	31	31	1.00	87.2	87.2	1.00	0.6	0.7
cm151a	34	34	1.00	180.9	150.9	0.83	0.4	0.5
cm152a	32	32	1.00	180.0	148.8	0.82	0.4	0.5
cm42a	34	34	1.00	110.7	110.7	1.00	0.6	0.7
cm82a	36	36	1.00	170.3	170.0	0.99	0.4	0.5
cm85a	72	72	1.00	304.2	280.2	0.92	0.6	1.1
con1	23	23	1.00	127.7	120.0	0.93	0.4	0.4
rd73	113	113	1.00	467.2	445.8	0.95	1.2	2.9
sao2	198	206	1.04	734.4	688.6	0.93	1.2	5.9
vg2	102	102	1.00	464.0	413.3	0.89	1.3	3.5
sum	2801	2840	1.01	11939.1	11189.2	0.93	44	136.4

표 3 커널 추출에서의 리터럴과 전력소모 증감의 기존 결과와의 비교

name	#literals		Power Consumption		Time(sec)	
	[4]	*1	[4]	*1	[4]	*1
5xp1	163	157	834.6	767.2	13.0	5.7
apex6	1215	1181	5863.4	5151.6	586.0	245.6
apex7	373	372	1796.4	1578.9	121.8	42.6
b1	12	12	40.3	33.4	0.4	0.4
b9	169	164	774.8	697.2	19.5	7.5
bw	190	187	907.1	826.3	6.9	6.9
c8	193	190	980.2	827.7	8.5	6.5
cht	231	230	1244.1	937.2	5.3	6.8
cm138a	34	31	116.5	63.6	2.0	1.7
cm151a	34	30	180.9	132.8	1.0	0.9
cm152a	32	28	180.0	122.5	0.8	0.8
cm42a	35	34	101.9	80.7	2.4	1.7
cm82a	36	34	175.9	123.4	2.7	1.4
cm85a	67	65	289.2	272.1	19.6	3.7
con1	23	23	127.7	107.1	0.6	0.6
rd73	122	119	578.6	494.9	177.9	20.6
sao2	192	193	857.5	804.3	42.0	13.2
vg2	103	99	506.9	446.4	3.3	2.9
Sum	3,224	3,149	15,556	13,467.3	1,013.7	369.5

*1 : 본 연구에서 제안한 방법

표 3 에서는 기존의 방법[4]과의 커널 추출시의 비교를 제시하였다. [4]의 방법은 그 기본 가정이 본 논문의 방법과 다르므로 매우 좋지 않은 결과를 보였다. 그러나 큐브추출에 대하여는 매우 비슷한 결과를 내었다. 그 이유는 큐브의 추출에 대해서는 [4]의 방법과 본 논문의 방법이 식 (4)에서 보듯이 기존의 방법과 크게 차이가 나지는 않기 때문으로 보인다.

5. 결 론

본 논문에서는 논리합성의 중요한 연산인 공통 표현 추출에 있어서의 저전력 설계를 위하여 전력소모 예측 함수의 설정, 공통표현 추출방법과 전력소모 예측함수의 적용방법을 제시하고 이를 구현하여 그 실험결과를 알아보았다. 본 논문의 요점은 공통표현 추출의 수학적 방법인 사각형 커버링에 어떤 전력소모 예측 방법을 어떻게 효과적으로 적용할 수 있는가 이다. 일반적인 회로의 구성은 CMOS로 구성된다고 가정하였으며 따라서 전력소모는 출력노드의 정전용량과 스위칭 활동량에 의하여 좌우된다.

본 논문에서는 이 두 가지의 요소가 적절히 반영되도록 무게함수를 이용하여 스위칭 활동량과 함께 전력소모식을 제시하였으며 이를 SIS-1.2에서의 커널과 큐브 추출에 사용하는 사각형 커버링에 적용하는 방법을 제시하였다. 본 논문에서의 전력소모표현은 각 노드에서의 정전용량과 그 스위칭 활동량을 반영하되 노드의 표현 및 구현이 SOP(Sum of Products) 표현에 의한 complex gate로 이루어진다고 가정한다. 이러한 가정아래 정전용량은 노드의 입력 수에 비례한다고 가정하였으며 스위칭 활동량은 BDD를 구성하여 계산하는 방법을 사용하였다. 실험결과 비교적 적은 수(약 1%)의 리터럴 증가에 비하여 많은 전력소모 감소(약 8%)를 이루었으며 본 논문결과를 토대로 ESOP(Exclusive Sum of Products)등 새로운 표현을 도입하면 더욱 발전될 수 있을 것으로 예상된다.

참 고 문 헌

[1] "SIS: A system for sequential circuit synthesis," Report M92/41, UC Berkeley, 1992.

[2] K.Roy, S.C.Prasad,"Circuit Activity Based Logic Synthesis for Low Power Reliable Operations," IEEE Transactions of CAD, Dec. 1993, pp.503-513.

[3] S.man, M.Pedram, "Multi-level Network Optimization for Low Power," ICCAD, 1994, pp. 372-377.

[4] S.Iman, M.Pedram, "Logic Extraction and Factorization for Low Power," DAC, 1994, pp.248-253.

[5] Rohfleisch, et. al, "Reducing Power Dissipation after Technology Mapping by Structural Transformations," DAC, 1996, pp.789-794.

[6] S.Iman, M.Pedram, "POSE: Power Optimization and Synthesis Environment," DAC, 1996, pp.21-26.

[7] R.K.Brayton et. al, "MIS: A Multiple-Level Optimization System," IEEE Transactions on CAD, vol. CAD-6, no. 6, Nov., 1987, pp.1062-1080.

[8] R.K.Brayton, G.D.Hachtel, A.L.Sangiovanni-Vincentelli, "Multilevel Logic Synthesis," Proceedings of IEE, vol. 78, no. 2, 1990, pp.264-300.

[9] J.Rabaey and M.Pedram, Low Power Design Methodologies, Kluwer Academics Publishers, 1996, p.10, p.26

[10] J.Fishburn et. al, TILOS : A Posynomial Programming Approach to Transistor Sizing, ICCAD, 1985 pp.326-328

[11] M.Borah et. al, Transistor Sizing for Low Power CMOS Circuits, IEEE TCAD, Nov, 1996, pp.665-671.

[12] S.Devadas and S.Malik, "A Survey of Optimization Techniques Targeting Low Power VLSI Circuits," DAC, pp. 242-247, 1995.

[13] R.Marculescu, D.Marculesc and M.Pedram, Switching Activity Analysis Considering Spatiotemporal Correlations, ICCAD, 1994.

황 민

1986년 전남대학교 전산통계학과 졸업(학사). 1988년 중앙대학교 대학원 전산계산학과 석사. 1995년 ~ 현재 전남대학교 대학원 전산통계학과 박사수료. 관심분야는 VLSI/CAD, 논리합성, 인공지능

정 미 경

1987년 전남대학교 전산통계학과 졸업(학사). 1989년 전남대학교 대학원 전산통계학과 석사. 1995년 ~ 현재 전남대학교 대학원 전산통계학과 박사수료. 관심분야는 VLSI/CAD, 논리합성, 인공지능

이 귀 상

1980년 서울대 공대 전기공학과 졸업(학사). 1982년 서울대 대학원 전자계산기공학과 석사. 1982년 금성통신 연구소 연구원. 1991년 Pennsylvania 주립대학 박사. 1984년 ~ 현재 전남대 전산학과 부교수. 관심분야는 VLSI/CAD, 멀티미디어 시스템, 테스팅, 논리합성