

스위칭 동작 최소화를 통한 저 전력 자원할당 알고리즘

신무경, 인치호, 김희석*

sh0909@venus.semyung.ac.kr, ich410@venus.semyung.ac.kr, khs8391@chongju.ac.kr*

세명대 컴퓨터학과, 청주대 전자공학과*

A Low Power Resource Allocation Algorithm Based on Minimizing Switching Activity

Mu-kyoung Shin, Chi-ho Lin, Hi-seok Kim*

smk0909@hanmail.net, ich410@venus.semyung.ac.kr, khs8391@chongju.ac.kr*

Dept. of Computer Science, Semyung University.

Dept. of Electronic Engineering, Chongju University*

Abstract

This paper proposed resource allocation algorithm for the minimum switching activity of functional unit in high level synthesis process as like DSP which is circuit to give many functional unit.

The resource allocation method after scheduling use the power function calculating average hamming distance and switching activity of the between two input. First of all, the switching activity is calculated by the input value after calculating the average hamming distance between operation.

In this paper, the proposed method though high level simulation find switching activity in circuit each functional unit exchange for binary sequence length and value bit are logic one value. To use the switching activity find the allocation with minimal power consumption, the proposed method visits all control steps one by one and determines the allocation with minimal power consumption at each control step.

As the existing method, the execution time can be fast according to use the number of operator and max control step. And it is the reduction effect from 6% to 8%.

I. 서론

1)이전의 상위 수준 합성에 관련된 대부분의 연구는

* 본 연구는 과학기술부·한국 과학재단 지정 청주대학교 정보통신 연구센터의 지원에 의한 것입니다.

VLSI의 면적과 성능을 최적화하는 방법에 대하여 집중적으로 연구 되어왔다[1-2]. 최근 들어, 디지털 시스템의 성능과 복잡도가 증가함에 따라 전력이 중요한 요소가 되기 때문에, 자원 할당 과정에서도 전력을 줄이기 위한 방법들이 제안되고 있다. 저 전력 회로구현은 여러 설계 수준의 범위를 포함해야 하나 특히 회로 설계 초기 단계의 결정은 다음 단계에 큰 영향을 미치므로 상위 수준에서의 최적화는 매우 중요하다. 이전에 제안된 방법으로는 데이터 경로의 병렬화, 파이프라이닝 또는 다중 전압(multiple voltage)을 이용하여 공급전압(supply voltage)을 감소시키거나[3-4], 피 연산자를 공유하는 시블링 연산(sibling operation)을 같은 기능장치에 할당하는 스케줄링(scheduling)과 바인딩(binding) 방법을 제시하였다[5-6].

본 논문에서는 DSP 분야의 회로나 필터를 대상으로 연산자가 소모하는 전력을 최소화 하고자 한다. DSP와 같은 분야에서는 산술 연산을 수행하는 연산자의 수가 많고 곱셈기와 같은 연산자가 소모하는 전력이 훨씬 크다. 따라서 회로전체의 전력 소모를 줄이기 위해 연산자가 소모하는 전력을 우선적으로 최소화하는 것은 큰 의미를 가진다.

본 논문에서는 이미 스케줄링 된 DFG를 대상으로 전력 소모의 원인이 되는 스위칭 동작을 최소화하는 자원 할당 알고리즘을 제안한다. 스케줄링 된 DFG상에 있는 여러 개의 연산은 자원공유를 통하여 같은 기능장치에 구현될 수 있다. 이런 경우 두 개의 연속적인 연산의 실행사이에 각 연산의 입력 변수들이 연속적으로 변화하기 때문에 기능장치의 스위칭동작이 변하게 된다. 이때 자원할당 과정에서 기능장치의 입력 신호들 사이의 스위칭동작과 상관관계를 고려하여 소비전력을 감소시킨다.

II. 전력 소모 모델

CMOS 회로에서의 전력 소모는 동적 전력 소모, 단락 전류 전력 소모, 그리고 누설 전류에 의한 전력 소모로 구성된다. 전형적으로 단락 전류 전력 소모에 의한 전력 소모는 전체 회로에서의 전력 소모 중 약 10% 이하를, 그리고 누설 전류에 의한 전력 소모는 약 5% 이하를 차지하기 때문에, 보통 상위 수준 합성에서는 동적 전력 소모만을 고려한다. CMOS 회로에서의 평균전력은 다음과 같이 표현될 수 있다.

$$P = \frac{1}{2} C_L V_{DD}^2 f_{CLK} P_{switching} \quad (식 1)$$

여기서 C_L 은 유효 정전용량, V_{DD}^2 공급전압 그리고 f_{CLK} 는 클럭 주파수이고, $P_{switching}$ 는 스위칭동작의 평균값을 나타낸다. 여기서 스위칭동작의 평균값은 $1/f_{CLK}$ 클럭 사이클 마다 출력변화의 평균수이다[7]. 회로 안의 노드에 대해서 C_L , V_{DD}^2 , 그리고 f_{CLK} 는 대부분 주어진다. 그러나, $P_{switching}$ 는 입력패턴과 회로구조 안에서 결정해야한다. 위의 식에서 알 수 있듯이, 유효정전용량과 공급전압, 그리고 클럭 주파수가 주어진다 면, 스위칭동작의 평균값을 최소화시키면 CMOS 회로에서의 평균전력소모는 최소가 된다. 위의 식1에서 스위칭 동작의 평균값인 $P_{switching}$ 을 입력값에 대한 비트패턴을 이용한 식으로 변형하면 다음과 같다.

$$P_{avg} = \frac{C_{fu} V_{clk}^2 switch(i, i-1)}{2 \times Bitwidth} \quad (식2)$$

위의 식2에서 $switch(i, i-1)$ 는 기능장치에 대한 하나의 입력단에 연속적으로 입력되는 입력값에 대한 스위칭 동작을 나타낸다. 즉, 제어단계 i 와 제어단계 $i-1$ 사이의 존재하는 입력신호의 스위칭의 총합이다. 그리고 $Bitwidth$ 는 해당자원에 대한 입력신호의 비트수를 나타낸다.

본 논문에서는 평균 스위칭 동작을 아래의 식3과 같이 구할 수 있다. 먼저 연속되는 입력값에 대해서 신호 확률에 의한 두 입력값 사이의 스위칭 동작을 구하게 된다. 그 다음 연속되는 입력값에 대해서 해밍거리(hamming distance)를 이용하여 평균 스위칭 동작을 구한다. 해밍거리는 두 변수의 2진수를 XOR를 취하여 구해지며, 두 개의 2진수 사이의 bit의 차이 수를 말한다. 즉, 두 입력 변수사이의 해밍거리를 이용하므로 데이터 상관관계를 고려하게 되어 보다 정확한 스위칭 동작을 구할 수 있다. 아래 식3은 하나의 기능장치에 대해 연속적인 입력값에 대한 평균 스위칭 동작을 나타낸다.

$$SW = \frac{\sum_{i=1}^n switch(x_i, x_{i-1}) \times HAD(x_i, x_{i-1})}{Bitwidth} \quad (식 3)$$

위 식3에서 $HAD(x_i, x_{i-1})$ 은 연속되는 두 입력값 사이의 해밍거리를 나타낸다.

III. 신호 확률에 대한 스위칭동작

스위칭동작은 단위 클럭 내에서 신호의 스위칭이 발생할 확률이다. 그리고 그것은 입력값의 신호 확률에 의해 계산되어진다[8]. 예를 들어 어떤 이진 순차입력을 x 라고 할 때 순차입력의 길이를 $length(L)$ 이라 하고, 순차입력 중 '1'인 비트를 S 라 하면 신호 확률은 S/L 이 된다. 그리고 $P(x->1) = S/L$ 로 표시한다. 또한 순차입력 중 '0'인 비트를 G 라 할 때 $G = length - S$ 가 되고, 신호 확률은 G/L 이 되며, $P(x->0) = G/L$ 로 표시된다. 여기서 $P(x->1) + P(x->0) = 1$ 이 된다.

하나의 신호가 클럭에 동조하여 입력된다고 할 때, 입력신호의 변화상태를 표1과 같이 4가지 상태로 표현할 수 있다. 표1에서 A, B는 신호가 '0'에서 '1'로 '1'에서 '0'으로 변화된 상태를 나타낸다.

표 1. 네 개의 동작 신호

x	P(x -> x')	동작
0	0 -> 0	'0'상태유지
A	0 -> 1	0->1변화
B	1 -> 0	1->0변화
1	1 -> 1	'1'상태유지

이전 입력 신호 값과 현재 입력 신호 값에 존재확률을 $P(x^0)$, $P(x^A)$, $P(x^B)$, $P(x^1)$ 로 나타낸다. 동작 확률의 관계에 따라 신호 확률을 나타내면 아래 식과 같다.

$$P(x->0) = P(x^0) + P(x^B) \quad (식 4)$$

$$P(x->1) = P(x^1) + P(x^A) \quad (식 5)$$

$$P(x^0) + P(x^A) + P(x^B) + P(x^1) = 1 \quad (식 6)$$

$$P(x^A) = P(x^B) \quad (식 7)$$

위 식7에서 올라가는 것과 내려가는 것의 변화는 같다. 따라서 $P_{switching}$ 는 다음과 같이 나타낼 수 있다.

$$P_{switching} = P(x^A) + P(x^B) = 2P(x^A) = 2P(x^B) \quad (식 8)$$

두 개의 클럭 사이클 안에 신호값이 서로 독립되었다고 가정하면

$$P_{switching} = P(x^A) + P(x^B) = 2P(x->1) * P(x->0) \quad (식 9)$$

위 식9와 같이 나타낼 수 있다. 그리고 순차입력 비트의 길이를 L 이라 하고 순차입력 중 '1'인 비트를 S 라 하면 S 에 대해 스위칭 동작은 아래 식10과 같다.

$$P_{switching} = 2 \left(\frac{S}{L} \right) \cdot \left(\frac{L-S}{L} \right) \quad (식 10)$$

IV. 제안하는 자원할당 알고리즘

본 논문에서 제안하는 자원할당 알고리즘은 먼저 각 제어 단계마다 자원공유 가능한 모든 경우에 대해서 연속적으로 입력되는 입력변수에 대한 할당 가능한 경우의 수를 구한다. 그 다음 식3에 의해 평균 스위칭동작이

스위칭 동작 최소화를 통한 저 전력 자원할당 알고리즘

최소가 되는 입력 변수를 할당하게 된다. 이렇게 구해진 입력변수를 입력으로 하는 공유 가능한 연산자를 할당하게 된다. 연산자 할당 방법은 할당 가능한 모든 경우에 대해 식3에 의해 평균 스위칭동작을 구한 후, 식2에 의해 소비전력을 측정하게 된다. 그리고 각 제어 단계에서 전력 소모가 최소가 되는 경우를 찾아서 할당하므로 연산자가 소모하는 전력을 줄일 수 있다.

아래 그림1은 회로 중 곱셈기의 DFG만을 나타낸 것으로 4단계로 이미 스케줄링 되어 있다고 가정한다. 그림1에서 주어진 DFG는 2개의 곱셈기만을 사용할 수 있는 자원 제약 하에 자원 할당과정을 거쳐 DFG내의 연산들을 2개의 곱셈기에 할당할 것이다.

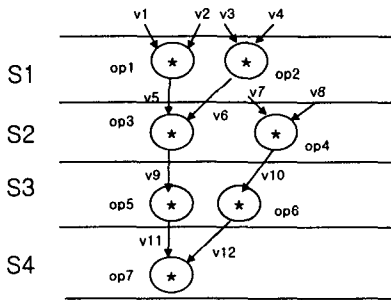


그림 1. 스케줄링 된 DFG

예를 들어 그림1에서 제어 단계 1과 제어 단계 2사이에서 할당할 수 있는 경우의 수는 다음과 같이 두 가지 경우가 있다.

표 2. 공유 가능 연산자에 대한 입력 가능변수와 선택 입력변수에 대한 공유 연산자 선택

	공유가능한 연산자	연산자에 대한 입력 가능 변수	선택된 입력 변수	선택된 공유 연산자
경우1	op1, op3	v1-v5, v2-v6	v1-v5, v2-v6	경우1의 선택 (op1, op3) (op2, op4)
		v1-v6, v2-v5		
	op2, op4	v3-v7, v4-v8	v3-v7, v4-v8	
		v3-v8, v4-v7		
경우2	op1, op4	v1-v7, v2-v8	v1-v7, v2-v8	
		v1-v8, v2-v7		
	op2, op3	v3-v5, v4-v6	v3-v6, v4-v5	
		v3-v6, v4-v5		

위 표는 두 제어단계 사이의 공유 가능한 연산자의 두 가지 경우에 대한 각 연산자의 입력 가능한 변수 쌍을 나타내었다. 먼저 연속적으로 입력되는 이들 변수 쌍 중 스위칭 동작이 최소가 되는 입력 쌍을 선택하게 된다. 그 다음 선택된 입력 변수 쌍을 입력으로 하는 연산자에 대한 전력 소모가 최소가 되는 경우를 선택하여 기능장치에 할당하게 된다. 이와 같은 방법으로 제어단계를 제어 단계를 한 단계씩 증가 시키면서 할당 가능한 경우에 대해 입력변수와 연산자를 입력 비트패턴과 해밍거리를 이용하여 스위칭 동작을 최소화하여 자원을 할당한다.

예를 들어 공유 가능한 두 가지 경우에 대해서 각 기능장치에 입력되는 입력값이 아래 그림2와 같다고 가정하면 자원할당방법은 다음과 같다. 먼저 첫 번째 경우에서

와 같이 fu1과 fu2에 입력되는 순차입력들에 대해서 입력비트의 길이와 '1'인 비트의 수를 구하여 식10에 의해 스위칭동작을 구한 후, 연속적인 입력 비트에 대한 해밍 거리를 구하여 식3에 의해 평균 스위칭동작을 구한다.

각 연산자에 대한 평균 스위칭동작이 최소가 되는 입력 변수를 선택하게 된다. 그런 다음 선택된 입력변수에 대하여 위와 같은 방법에 의해 스위칭동작을 구한 후, 식1에 의해 각 기능장치가 소모하는 전력을 계산하게 된다. 이렇게 구해진 각 기능장치에 대한 전력소모를 비교하여 전력소모가 최소가 되는 경우를 할당하게 된다.

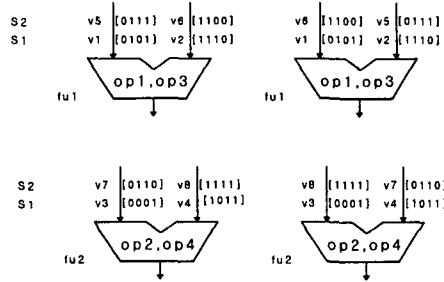


그림 2. 2가지 경우 중 첫 번째 경우의 fu1과 fu2에 입력되는 비트

이와 같은 방법으로 제어 단계를 다음단계로 증가하면서 전력소모가 최소가 되는 공유 가능한 연산자를 기능장치에 할당하게 된다. 만약 연산자가 할당되지 않은 제어 단계가 존재한다면 이러한 제어 단계에서는 실제적으로 연산자가 전력을 소모하지 않기 때문에 스위칭 동작을 구할 때는 신호의 개수로 (L-1)를 사용한다. 그러나 만약 신호의 길이가 무한히 클 경우에는 L로 간주한다.

마지막 제어 단계에서 연산자를 할당할 때는 다른 제어 단계와는 다른 방법으로 스위칭동작을 구해야 한다. 마지막 제어 단계와 첫 번째 제어 단계사이의 스위칭도 고려하여 스위칭동작을 구해야 한다. 본 논문에서 제안하는 알고리즘은 다음과 같다.

```

consider scheduled DFG is given
loop control step i
  Shared resource possible case construction j ;
  Shared resource possible about input values construction I ;

  loop construction j
    Sum_power = 0;
    loop functional unit same type k
      loop input values I;
      SW =Calculate using equation(3);
      select input values with minimum switching activity;

      increment I;
    until empty
    if ( selected input values == minimum switching activity)
      storage construction I ;
      Fu_power = 0;
      Fu_power=Calculate using equation(2);
      Sum_power=Sum_power + Fu_power;
    until empty
    if ( construction j == minimum power)
      storage construction j and Sum_power;
    increment j;
  until empty
  increment i;
until empty
    
```

그림 3. 제안하는 알고리즘

V. 실험 및 고찰

본 논문에서 제안하는 자원할당 알고리즘의 수행시간은 제어 단계 수에 비례하고 하나의 제어 단계에서 할당 가능한 경우의 수에 비례한다. 본 논문에서는 저 전력을 고려한 스케줄링 과정을 거친 후의 DFG를 가지고 자원할당을 함으로 연산자의 개수는 고정되어 있기 때문에, 수행시간 계산에서 입력신호의 개수와 제어단계 수에 대해서만 고려한다. 그러므로 본 논문에서 제안하는 알고리즘은 그래프를 이용한 알고리즘과 비교할 때 시간 복잡도가 높지 않다. 또한 입력에 대한 상관 관계까지 고려하여 좀 더 정확한 전력측정을 할 수 있다.

본 논문이 제안하는 자원할당 방법에 대한 실험은 Ultra-sparc 5에서 수행하였다. 그리고 성능평가를 위해 volterra필터와 wavelet필터, 그리고 하나의 예로 테스트회로에 대해 덧셈기와 곱셈기의 개수에 제한을 두어 실험을 하였다. 회로에 대한 스케줄링은 저 전력을 고려한 리스트 스케줄링 방법을 사용하였다.

본 논문이 제안하는 자원할당 방법에서 한번에 할당하는 제어단계의 개수를 한 개로 하여 전력 소모측정을 한 결과 volterra필터와 wavelet필터에서는 각각 6.78%와 8.25%의 전력 감소율을 보이며, IIR필터는 6.21%의 감소율을 보인다. 그리고 실험에 사용한 테스트회로에서는 8.5%의 전력 감소율을 보이고 있다.

표3. 전력 소모 비교

단위 : mW

	기존의 자원 할당방법	제안한 자원 할당방법	전력감소 (%)
Volterra	11.211	10.45	6.78
Wavelet	11.976	10.987	8.25
IIR	11.050	10.980	6.21
Example	5.98	5.128	8.5

아래 표4는 DFG내의 곱셈기에 대해 자원할당을 수행하는데 걸리는 시간을 나타낸다. DFG내의 곱셈기에 대해 수행시간을 비교해 보면 volterra필터와 wavelet필터에서 많은 속도 향상이 있다. 본 논문에서 제안한 자원할당 방법은 각 제어 단계에서 할당 가능한 경우의 수만 고려하여 수행하게 된다. 그러나 기존의 자원할당 방법은 전역 기술방법을 사용하므로 최대 제어단계 모두에 대한 할당 가능한 경우의 수를 고려하므로 많은 경우의 수가 생기게 되어 수행속도가 본 논문에서 제안한 방법보다 길어지게 된다. 따라서 본 논문에서 제안하는 자원할당 방법을 이용할 경우 연산자의 수가 많아지거나 제어단계가 길어질수록 전역 기술방법을 사용할 경우와 비교할 때 많은 수행 시간의 차이를 보인다.

표4. 곱셈기 DFG에 대한 수행 시간 비교

단위 : sec

	기존의 자원 할당방법	제안한 자원 할당방법
Volterra	180606.96	15.32
Wavelet	93843.75	14.48
IIR	87394.64	17.23
Example	61.71	5.657

VI. 결론 및 연구과제

본 논문에서 제안하는 방법은 제어 단계를 한 단계씩 증가시키면서 입력값의 길이와 비트가 '1'인 값을 구하여 스위칭동작을 구한 후, 연속적인 입력변수에 대하여 해밍거리를 구하고 이들의 곱을 이용하여 평균 스위칭동작을 구하게 된다. 그리고 스위칭동작이 최소가 되는 입력변수를 선택하게 된다. 이렇게 선택된 입력변수를 이용하여 기능장치에 공유 가능한 연산자에 대한 전력소모를 계산하게 되고, 이 전력 소모를 이용하여 전력 소모를 최소화할 수 있는 경우를 선택하여 할당하게 된다.

본 논문에서 제안하는 방법을 이용하여 자원할당을 할 경우 기존 방법과 비교했을 때 그 수행속도는 사용하는 연산자의 개수와 최대 제어 단계에 따라서 빨라 질 수 있다. 그리고 소모하는 전력의 경우, 작게는 6%에서 8%까지 감소효과가 있다.

본 논문은 상위 레벨 합성과정 중 자원할당 과정에서 소모하는 전력을 최소화하는 것이 목적이다. 하지만 상위 레벨 합성과정에는 레지스터할당, 버스, 그리고 MUX 할당 등의 과정이 있다. 앞으로 이러한 과정에서도 본 논문의 알고리즘이 적용되어 전력소모를 줄일 수 있는 연구를 할 필요가 있다.

본 연구는 반도체설계연구센터(IDEC)의 지원에 의한 것입니다.

참고 문헌

- [1] D. Gajski and N. Dutt, High-level Synthesis: Introduction to Chip and System Design. Kluwer Academic Publishers, 1992.
- [2] G. D. Micheli, Synthesis and Optimization of Digital Circuits. New York: Mc-Graw Hill, Inc., 1994.
- [3] W. T. Shiue, C. Chakrabarti, "Low-Power Scheduling with Resources Operating at Multiple Voltages", IEEE Transactions on Circuits and Systems, vol.47, no.6, pp.536-543, 2000.
- [4] Y.-R. Lin, C.-T. Hwang, and A. C.-h. Wu, "Scheduling techniques for variable voltage Low Power Design", ACM Trans. Design Automat. Electro. Syst., vol. 2, no. 2, pp. 81-97, April 1997.
- [5] E. Musoll and J.Cortadella,"Scheduling and Resource Binding for Low Power", in Proceedings of International Symposium on System Synthesis, pp.104-109, Apr. 1995.
- [6] Y. Fang and A. Albicki,"Joint Scheduling and allocation for Low Power", Int'l Symp. on Circuits & Systems, pp. 556-559, May. 1996.
- [7] M. Pedram, "Power minimization in IC Design: Principles and applications," Transactions of ACM, vol.1,no. 1, pp.1-58, March, 1996.
- [8] F. N. Najm, " A survey of power estimation techniques in VLSI circuits," IEEE Transactions on VLSI Systems, vol.2 , no.4, pp.446-455, 1995.