최소 자원을 사용하는 저전력 데이터 패스 할당 알고리즘

문 성 필, 김 영 환
포항공과대학교 전자전기공학과
전화 : 054-279-2878 / 핸드폰 : 017-520-9503
E-mail : spmoon@bud.postech.ac.kr

Abstract

This paper presents a new algorithm for allocating the data path to achieve the minimum power consumption under the constraints of minimum hardware resources. In order to minimize the power consumption, the proposed algorithm tries to minimize the input transitions of functional units, unnecessary computations, and size of interconnects in a greedy manner during allocation. Experimental results using benchmarks indicate the proposed algorithm achieves 17.5% power reduction on average, when compared with the *genesis-lp* [1] high-level synthesis system.

Ⅰ. 서론

최근에 시스템의 전력 소모는 VLSI 설계의 중요한 요소가 되었고, 여러 설계 수준에서 전력 소모를 줄이고자 노력하고 있다. 한편 집적회로의 설계과정을 볼 때 상위 수준에서의 설계는 최종 구현된 시스템에 가장 큰 영향을 주기 때문에 상위 수준에서의 전력 최적화는 매우 중요하다.

상위 수준 합성이란 CDFG(Control Data Flow Graph)와 같이 행위 수준에서 기술된 시스템을 레지스터 트랜스퍼 수준으로 변환하는 것을 의미하며, 스케줄링과 할당 단계로 나뉘어진다. 스케줄링 단계에서는 CDFG내의 연산들을 제어 구간(control step)에 대응시키고, 할당 단계에서는 연산과 변수들을 기능 연산자(functional unit)와 레지스터에 할당하는 일을 수행한다. 제안하는 알고리즘은 스케줄링이 끝난 CDFG를 받아들여 할당과정에서 전력 최적화를 수행한다.

전력 최적화를 위한 할당 방법은 현재까지 많은 연구가 이루어졌다. [1,2]는 공유를 통해 기능 연산자나 레지스터의 입력 변수들을 조절하여 시스템 내에 발생하는 스위칭을 줄이고자 하였다. 하지만, 이들은 할당과정에서 기능 연산자의 불필요한 연산을 고려하지 못한 문제점을 지닌다. [3,4]는 [1,2]를 보완하여 레지스터 할당 과정에서 기능 연산자의 불필요한 연산을 없애어 전력 소모를 줄이고자 하였다. 하지만, 이 방법도 연결구조의 전력 소모를 고려하지 못해 데이터 패스 일부에 대한 최적화를 이루는데 그치고 말았다. 본 논문에서는 이러한 문제점들을 반영하여, 데이터 패스에서 발생하는 모든 전력 소모를 고려함으로써 데이터 패스 전체에 대한 전력 최적화를 수행하는 알고리즘을 제안하고자 한다.

Ⅱ. 제안하는 데이터 패스 할당 알고리즘

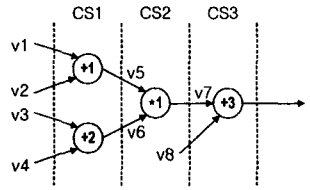
2.1 기본 개념

데이터 패스 할당과정에서 전력 소모를 줄일 수 있는 방법은 다음과 같다. 첫째는 기능 연산자와 레지스터에 들어오는 입력 변수들의 순서를 조절하여 하드웨어 자원에서 발생하는 스위칭을 줄이는 방법이다[1,2].

둘째는 레지스터 할당으로 발생하는 기능 연산자의 불필요한 연산을 줄이는 방법이다[3,4]. 여기서 기능 연산자의 불필요한 연산이란 기능 연산자의 idle time, 즉 DFG 내의 연산을 수행하지 않는 시간에서 레지스터 할당으로 인해 기능 연산자의 입력이 바뀌어 발생하는 것으로 시스템의 올바른 동작과는 무관한 연산을 의미한다. 그림 1은 레지스터 할당에 따른 기능 연산자의 불필요한 연산이 발생하는 예를 보여준다. 그림 (c)는 그림 (a)의 DFG에서 연산 +1, +3이 ADD1에, +2가 ADD2에, 그리고 *1이 MUL1에 할당되었다고 가정했을 때, 그림 (b)의 레지스터 할당에 따른 기능 연산자의 동작과정을 나타내었다. 그림 (c),

(d)에서 색칠된 블록의 U표시는 불필요한 연산을 나타낸다. 그림 1에서 볼 수 있듯이 경우 1에서 3번, 경우 2에서는 2번의 불필요한 연산이 발생하므로, 전력 소모 면에서는 경우 1보다 경우 2의 할당이 유리하다.

세 번째는 할당에 따라 변화하는 기능 연산자와 레지스터 사이의 연결구조를 줄이는 방법이다. 제안하는 알고리즘은 전술한 3가지 방법을 모두 고려하여 데이터 패스 전체에 대한 전력 최적화를 수행한다.

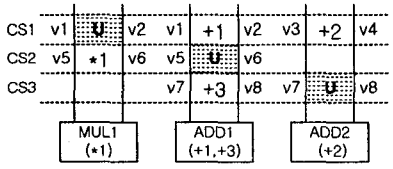


(a) 예제 CDFG

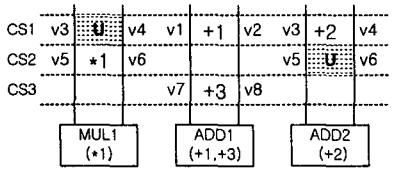
Register allocation 경우 1	
Reg 1	v1, v5
Reg 2	v2, v6
Reg 3	v3, v7
Reg 4	v4, v8

Register allocation 경우 2	
Reg 1	v1, v7
Reg 2	v2, v8
Reg 3	v3, v5
Reg 4	v4, v6

(b) 레지스터 할당의 2가지 경우



(c) 경우 1에 따른 기능 연산자의 동작



(d) 경우 2에 따른 기능 연산자의 동작

그림1. 레지스터 할당에 따른 기능 연산자의 동작

데이터 패스 할당은 기능 연산자 할당과 레지스터 할당으로 나누어진다. 일반적으로 둘을 동시에 최적화 하는 것은 복잡도가 너무 높기 때문에 대부분의 할당 알고리즘은 순차적으로 최적화를 수행한다. 제안하는 알고리즘은 기능 연산자의 전력 소모량이 레지스터 보다 크다는 점을 감안하여 기능 연산자 할당을 먼저 수행하고, 이후 레지스터 할당을 수행한다.

본 논문에서는 기능 연산자와 레지스터의 연결구조로 멀티플렉서를 사용하였으며, 멀티플렉서를 기능 연산자의 입력에 존재하는 입력 멀티플렉서와 출력에 존재하는 출력 멀티플렉서로 구분하였다. 표 1은 데이터 패스에서 발생하는 전력 소모 요소를 하드웨어 별로 구분하였으며, 기능 연산자 할당과 레지스터 할당과의 의존성을 나타내었다. 표 1에서 볼 수 있듯이 기능 연산자의 불필요한 연산과 입력 멀티플렉서는 기능 연산자 할당과 레지스터 할당의 복합효과에 의해 결정된다. 따라서 제안하는 알고리즘은, 먼저 기능 연산자 할당에서 기능 연산자의 필수 연산을 최적화하고, 이후 레지스터 할당에서 기능 연산자의 불필요한 연산, 레지스터, 그리고 멀티플렉서에 의한 전력 소모를 최적화 하게 된다.

제안하는 알고리즘은 그래프를 기반으로 하고, greedy 방법을 적용해 공유를 이루어 나간다. 그리고 기능 연산자와 레지스터는 할당에 사용하는 그래프만 다르고, 기본적으로 할당 과정은 동일하다.

	의존성	
	기능 연산자 할당	레지스터 할당
기능 연산자의 필수 연산	○	×
기능 연산자의 불필요한 연산	○	○
레지스터	×	○
입력 멀티플렉서	○	○
출력 멀티플렉서	×	○

표 1. 데이터 패스의 전력 소모 요소 구분

2.2 상위 수준에서의 전력 소모 예측

제안하는 알고리즘은 할당을 수행하는 도중 반복해서 시스템의 전력 소모를 예측하고, 이를 기반으로 전력 최적화를 수행한다. 따라서 할당의 효율성을 높이기 위해서는 상위 수준에서의 정확하고 빠른 전력 소모 예측이 필요하다. 일반적으로 상위 수준에서의 전력 소모 예측은 행위 시뮬레이션을 통해 얻은 스위칭 정보와 전력 모델을 이용해 수행한다[5].

행위 시뮬레이션 과정에서는 모든 변수와 변수 쌍간의 스위칭 정보(signal probability, switching activity)를 기록하며, 스위칭 정보가 0.1%의 허용 오차 내로 수렴할 때까지 시뮬레이션을 수행하였다.

제안하는 방법에서는 기능 연산자와 멀티플렉서의 전력 모델로 *BPCM*(Backward Propagated Capacitance Model)[6]을 사용하였다. 그리고 레지스터는 전력 소모가 입력의 스위칭 횟수에 정비례함을 이용해 다음 식과 같이 레지스터의 전력 소모를 모델링하여 사용하였다.

$$P_{register} = 1.53 \times SW + 1.24 \quad [\mu W]$$

여기서 SW는 신호가 천이할 확률을 의미한다. 위의 레지스터 전력 모델은 클락 주기가 10 ns이고, 공급전압이 5V일 때 게이트 수준 시뮬레이션으로 추출되었고, 식에서 상수항은 클락에 의해 일정 소모되는 전력을 나타낸다.

2.3 그래프 구성 및 할당 알고리즘

제안하는 알고리즘에서 데이터 패스 할당을 위한 그래프는 스케줄링된 DFG를 해석하여 만든다. 이때 기능 연산자 할당에 필요한 그래프의 노드는 연산으로, 레지스터 할당에 필요한 그래프의 노드는 변수로 구성된다. 그리고 기능 연산자 할당의 경우에는 동일 연산에 대해서만 공유를 이루어야 하므로, 연산의 종류별로 그래프를 구성한다. 이처럼 각 할당 과정에서 사용되는 그래프는 다르지만, 기본적으로 그래프를 구성하는 과정은 동일하다.

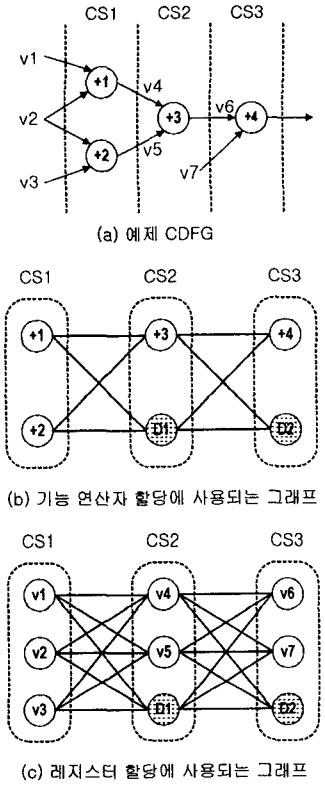
그래프를 만드는 과정은 다음과 같다. 먼저 제어 구간이 동일한 연산 또는 변수들을 노드로 하는 집합들을 형성한다. 그리고 집합들 중 가장 많은 노드의 수를 S_{max}라 할 때, 모든 집합내의 노드 수가 S_{max}가 되도록 더미노드를 추가한다. 여기서 S_{max}는 할당에 필요한 최소 자원의 개수가 된다. 마지막으로, 제어 구간이 인접한 두 집합간에 완전한 두갈래 그래프(complete bipartite graph)를 형성한다. 여기서 완전한 두갈래 그래프란 두 개의 노드 집합에 대해 같은 집합내의 노드간에는 edge로 연결되지 않고, 다른 집합의 노드간에는 모두 edge로 연결되는 그래프를 의미한다. 그림 2에서는 기능 연산자와 레지스터 할당에 사용하는 그래프의 예를 보여준다. 그림 (b), (c)는 그림 (a)의 DFG로부터 생성된 그래프이며, 여기서 색칠된 노드는 추가된 더미노드를 의미한다.

기능 연산자 할당과 레지스터 할당에 사용되는 그래프의 edge weight, W_{FU}와 W_{REG}는 다음과 같이 정의한다.

$$W_{FU} = PD_{USEFUL}$$

$$W_{REG} = PD_{REG} + PD_{USELESS} + PD_{MUX}$$

여기서 PD_{USEFUL}은 두 연산을 공유하기 전과 후의 기능 연산자의 필수 연산에 의한 전력 소모의 차이를 의미한다. 그리고 PD_{REG}, $PD_{USELESS}$, PD_{MUX}은 두 변수를 공유하기 전과 후의 레지스터, 기능 연산자의 불필요한 연산, 멀티플렉서에 의한 전력 소모의 차이를 의미한다. 더미 노드와의 edge weight는 더미노드를 더미노드와 이웃하는 실제 연산 또는 변수로 간주하고 값을 구한다.



(a) 예제 CDFG

(b) 기능 연산자 할당에 사용되는 그래프

(c) 레지스터 할당에 사용되는 그래프

그림 2. 그래프 구성 예제

정의한 edge weight는 자원의 공유에 따라서 값이 변하므로 할당 과정 중에 계속 업데이트하여야 한다.

그래프에서 최대 공유는 모든 제어 구간 내의 노드를 한번씩 거치며 서로 교차하지 않는 disjoint path들을 찾아내는 것과 동일하다. 이때 disjoint path의 개수는 한 제어구간 내에 있는 노드의 수인 S_{max}이며, 각 path는 실제 할당될 하드웨어 자원을 의미한다. 만약 i번째 disjoint path가 P_i이고, P_i를 형성하는 edge weight의 총합을 W_i로 할 경우 비용($Cost$)은 다음과 같이 정의한다.

$$Cost = \sum_{i=1}^{S_{max}} W_i$$

전력 최적화를 위한 할당은 정의한 비용이 최대가 되도록 S_{max}개의 disjoint path를 형성해 나가는 문제로 정형화 될 수 있다. 제안한 알고리즘에서는 문제 해결을 위해 greedy 방법을 적용하였다. 즉, 공유하는 각 단계에서 disjoint path를 형성하는데 아무 문제가 없는 edge들 중 weight가 가장 큰 것을 선택하여 공유를 이루어 나간다. 그리고 할당은 공유를 하지 않은 상태에서 S_{max}개의 disjoint path가 형성될 때까지 수행한다.

	No sharing		*Genesis-lp*		Proposed algorithm		Power reduction [%]	
	Power [μW]	*#_resources*	Power [μW]	*#_resources*	Power [μW]	*#_resources*	Compared with no sharing	Compared with *genesis-lp*
Diffeq	247.6	+8(4), *8(5), R(15)	347.9	+8(1), *8(2), R(8), M(16)	233.9	+8(1), *8(2), R(8), M(13)	5.5	32.8
Elliptic	628.1	+8(26), *8(8), R(50)	557.6	+8(3), *8(3), R(11), M(72)	562.1	+8(3), *8(3), R(11), M(56)	10.5	- 0.1
Volterra	493.6	+16(10), *8(6), *16(11), R(43)	485.7	+16(2), *8(2), *16(3), R(14), M(50)	385.3	+16(2), *8(2), *16(3), R(14), M(35)	22.1	20.7
Wavelet	659.8	+16(13),*16(14), R(55)	706.4	+16(3), *16(3), R(13), M(63)	577.6	+16(3), *16(3), R(13), M(44)	12.4	18.2
Example1	317.3	+8(4), *8(6), R(15)	286.0	+8(2), *8(2), R(6), M(15)	238.1	+8(2), *8(2), R(6), M(14)	35.8	16.8
Example2	335.8	+8(4), *8(4), R(18)	227.1	+8(1), *8(1), R(5), M(21)	189.6	+8(1), *8(1), R(5), M(14)	43.5	16.5
Average power reduction [%]							21.6	17.5

(+8: 8비트 덧셈기, +16: 16비트 덧셈기, *8: 8비트 곱셈기, *16: 16비트 곱셈기, R: register, M: MUX)

표 2. 벤치마크 회로에 대한 실험결과

Ⅲ. 실험 결과

본 장에서는 실험을 통해서 제안하는 데이터 패스 할당 알고리즘의 성능을 평가하였다. 실험에 사용된 회로는 표준 상위 수준 합성 벤치마크 회로인 difference equation solver, elliptic, volterra, wavelet filter와 2개의 예제 회로이다. 모든 회로는 100 M*Hz*, 5V 공급 전압에서 동작시켰으며, *Synopsys*사의 *DesignPower*를 사용해 게이트 수준에서 전력 소모를 예측하였다.

표 2에서는 제안한 알고리즘을 공유를 하지 않은 경우와 *genesis-lp*의 할당 결과와 비교하였다. *Genesis-lp*는 *Princeton University*의 저전력을 고려한 상위 수준 합성기로 최대 공유를 하며 전력 최적화를 이룬다는 점에서 제안하는 알고리즘과 목적이 유사하다. 표 2에서 볼 수 있듯이 제안하는 알고리즘은 공유를 하지 않은 경우에 비해 평균 21.6%, *genesis-lp*에 비해 평균 17.5%의 전력 감소율을 보인다. 할당에 사용된 자원의 수를 살펴보면, 제안하는 알고리즘은 공유를 이루지 않은 경우에 비해 멀티플렉서의 수는 많지만 기능 연산자와 레지스터의 수가 아주 작음을 볼 수 있다. 또한 *genesis-lp*와 비교했을 때, 기능 연산자와 레지스터의 수는 동일하지만 멀티플렉서의 수가 작은 것을 볼 수 있다.

Ⅳ. 결론

본 논문에서는 하드웨어 비용이 최소가 되도록 데이터 패스 할당과정에서 자원의 최대공유를 이루는 가운데 전력을 최소화하는 알고리즘을 제안하였다. 또한 벤치마크 회로에 대한 실험 결과는 제안한 알고리즘이 기존의 저전력을 고려한 상위 수준 합성기인 *genesis-lp*보다 평균 17.5%율 전력 감소율을 보인다. 제안하는 알고리즘은 DSP나 micro-controller와 같이 데이터 패스가 큰 비중을 차지하는 시스템을 설계하는데 효과적으로 사용할 수 있을 것으로 생각된다.

Acknowledgement

본 연구는 BK21, IDEC 및 삼성전자(주)의 지원으로 이루어졌음.

Reference

[1] A. Raghunathan and N. K. Jha, "Behavioral synthesis for low power," *Int. Conf. on Computer Design*, pp. 318-322, Oct. 1994

[2] A. Raghunathan and N. K. Jha, "ILP formulation for low power based on minimizing switching capacitance during data path allocation," *Int. Symp. on Circuit & System*, 1995

[3] E. Musoll and J. Cortadella, "High level synthesis techniques for reducing the activity of functional units," *Int. Symp. on Low Power Electronics and Design*, pp. 99-104, 1995

[4] A. Raghunathan, S. Dey and N. K. Jha, "Power management methodology for high-level synthesis," *Int. Conf. on VLSI Design*, pp. 24-29, 1998

[5] E. Macii, M. Pedram and F.Somenzi, "High-level power modeling, estimation, and optimization," *IEEE Trans. on Computer-Aided Design*, pp. 1061-1079, Nov. 1998

[6] J. Y. Choi and Y. H. Kim, "Register-transfer level power modeling for the efficient power estimation of VLSI systems," *IEEK CAD & VLSI Design Conference*, pp. 135-137, May. 1999