DSP를 위한 새로운 저전력 상위 레벨 합성

한 태희, 김 영숙, 인 치호, 김 희석[1]
세명대학교, 전산정보학과
[1]청주대학교, 전자공학과
전화 : 043-649-1272 / e-mail: hanhe2@kebi.com

A New Low Power High Level Synthesis for DSP

Taehee Han, Youngsuk Kim, Chiho Lin
Dept. of Computer Science, Semyung University
Dept. of Electronic Engineering, Chongju University[1]

Abstract

This paper propose that is algorithm of power dissipation reduction in the high level synthesis design for DSP(Digital Signal Processor), as the portable terminal system recently demand high power dissipation.

This paper obtain effect of power dissipation reduction and switching activity that increase correlation of operands as input data of function unit. The algorithm search loop or repeatedly data to the input operands of function unit. That can be reduce the power dissipation using the new low power high level synthesis algorithm. In this paper, scheduling operation search same nodes from input DFG(Data Flow Graph) with correlation coefficient of first input node and among nodes. Function units consist a multiplier, an adder and a register. The power estimation method is added switching activity for each bits of nodes. The power estimation have good efficient using proposed algorithm.

This paper result obtain more power reduction of fifty percents after using a new low power algorithm in a function unit as multiplier.

I. 서론

1990년대 후반부터, 휴대용 전자 제품의 발전으로 시장이 급격하게 성장하고, 많은 기능을 하는 전자제품 즉, PDA, 휴대폰, 노트북 등 고전력 소모에 따른 신뢰도 문제가 발생함에 따라, 저전력을 고려한 VLSI 설계가 점점더 중요해지고 있다.

상위 수준 합성이란 회로의 동작적 기술로부터 레지스터 전송 수준으로의 합성을 의미한다[1-3]. 상위 수준 합성 과정은 크게 스케줄링 과정과 자원 할당 과정, 제어기 합성 과정으로 나눌 수 있다. 스케줄링은 연산자를 제어 구간에 할당하는 것이고, 할당과정은 연산자들과 변수 등을 기능장치(functional unit), 레지스터, 멀티플렉서나 버스와 같은 연결 요소 등에 할당하는 것이다. 스케줄링의 목적은 사용 가능한 제한된 하드웨어를 이용하여 수행 시간을 최소화하는 것이다. 자원 할당은 사용되는 기능장치의 개수를 줄이는 것이다. 또한 자원 할당 과정에서 소비전력을 줄이기 위한 방법들이 제안되고 있다[4-7]. 그리고 data path 설계를 위한 스케줄링과 자원지정 알고리즘은 펑션유닛(덧셈기, 감산기 등)과 레지스터의 입력으로 들어가는 신호에 대해 천이의 수를 최소화시키는 방법이 제안되었다[8-9]. 이러한 방법은 펑션유닛에 입력으로 들어오는 피연산자들의 비트별 스위칭의 변화가 없도록 스케줄링 하는 것이다.

본 논문에서는 DSP를 위한 고정 상수의 입력을 연속적으로 펑션유닛에 할당하는 스케줄링 알고리즘을 제안한다.

II. CMOS 회로에서의 전력 소모 측정

2.1 일반적 전력소모 모델

CMOS 회로에서의 전력소모는 누설전류나 전원에서 연속적으로 공급되는 전류에 의한 정적 전력소모(Static power dissipation)와 스위칭에 과도전류와 부하 캐패시턴스의 충전·방전에 의한 동적 전력소모(Dynamic power dissipation)가 있다.

일반적으로 상위 레벨 합성에서는 동적 전력소모만을 고려한다. 그 이유는 파워소비 식(1)에서 전력 소비에 영향을 주는 요인으로 물리적인 요소, 즉 전원과 클럭주기 캐패시턴스의 정전용량이 있다. 그리고 캐패

시턴스의 정전 용량은 입력 비트에 따라 유동적이기 때문에 상위 레벨 합성에서 스위칭 엑티버티(Switching Activity)를 줄이는 방법이 저전력의 주목적으로 되고 있다. CMOS 인버터에서 출력이나 입력의 상태가 '0'에서 '1'로 '1'에서 '0'으로 변하는 과도상태에서, n형과 p형 트랜지스터 모두가 순간적으로 도통된다. 이 과정에서 전력 소비 현상이 발생하는데 일정한 비트가 들어올 경우 CMOS 인버터의 전력소비는 처음 상태를 유지하여 전력 소비를 줄일 수 있다. 물리적인 요인은 하위설계에서 다루고, 상위레벨에서는 스위칭 엑티버티를 줄임으로써 저전력을 유도한다.

$$P_{dynamic} = C_L \cdot V^2_{DD} \cdot f \quad \text{-- 식(1)}$$
$$C_L = \alpha C_{phy}$$

스위칭 엑티버티(α)는 각 평션유닛의 입력으로 들어오는 데이터 비트의 천이를 나타낸 것이다. C_L은 캐패시턴스의 정전용량을 나타내며, 다시 물리적 정전용량(C_{phy})과 스위칭 엑티버티의 곱으로 나타낼 수 있다. 식(1)에서 전력은 공급전압(V^2_{DD})이 조금만 감소하여도 소비 전력은 크게 감소하게 된다. 하지만 일반적으로 회로에서 영향을 주는 요인이 없다면 V_{DD}의 감소는 클럭 주파수에 의해 제한을 받고, 따라서 속도 제한 조건을 만족시키기 위해서는 유효 정전 용량의 감소를 고려해야만 한다.

2.2 본 논문에서의 전력소모 모델

제안하는 알고리즘에서의 스위칭 엑티버티 계산 공식은 아래와 같다.

α : switching activity = P1 / P2 -- 식(2)

$$\alpha = (P|(X_{in} - X_{in-1})| + P|(X_{in-1} - X_{in-2})| \cdots P|X_{i1} - X_{i0}|) + 1) * 100 / n \quad \text{-- 식(3)}$$

$$\text{Power} = \left(\frac{\sum_{i=0} (P|Xin - Xin-1| + 1(FIBSA)}{n} \right) * C_{phy} \cdot V^2_{DD} \cdot f \quad \text{-- 식(4)}$$

저 전력을 위한 공식(1)은 일반적인 파워소비 공식이고, 스위칭 엑티버디 α는 공유 전 파워소비(P2)로 공유 후 파워소비(P1)를 나누어 얻을 수 있다. 또한 α는 평션유닛의 입력으로 들어오는 비트들 즉, Xin은 평션유닛에 먼저 들어온 n번째 데이터의 i비트를 의미하고, Xin-1은 Xin이후에 들어온 비트를 의미한다. 각각의 스위칭 확률은 먼저 들어온 비트에서 나중 들어온 비트를 뺀 값의 절대값을 취해 모두 더하여 데이터 개수(n)로 나누어 백분율로 나타낼 수 있다. 그리고 식 (4)의 FIBSA(First Input Bit Switching Activity)는 처음 입력으로 들어오는 비트에 대한 스위칭 엑티버티 값을 1로 더해준다. 그 이유는 처음 평션유닛에 값이 입력될 때 스위칭이 일어나기 때문이다.

일반적으로 DSP 방정식은 다음과 같이 나타낼 수 있다.

$$y[n] = \sum_{i=0} hi * X[n-i] \quad \text{-- 식(5)}$$

DSP 방정식 식(5)에서 hi는 상수로서 고정되어질 수 있고, 상수에 지연신호(X[n-i])를 곱하여 모두 더해 결과를 얻는 것이다.

III. 제안하는 알고리즘

본 논문에서 제안하는 알고리즘은 다음과 같다.

```
Searching nodes for loop from input DFG
  loop {
       searching input nodes to multiplication
       shared resource possible for input node
    repeat {  if(static variables exist)
          investigate switching activity each nodes
          power estimation use equation (4)
          call NAFS()
            }
          }
NAFS( ) {
          for all input nodes
          if(V(a,b) for adder_FU ≠ Ø)
          control step   i
          if(same coefficient to node V'(a,b))
          scheduling node V'(a, b) to the control step i'
          each node V'(a,b) is scheduled same
          input node to the control step i
          increase control step i
          }
```

입력으로 들어오는 DFG에서 루프를 위한 노드들을 찾고, 찾아진 루프에 대해 곱셈기를 찾는다. 또한 이것이 공유되어질 수 있는 가를 결정한다. 만약 고정적인 상수가 존재하면 각각의 입력 노드에 대해 식(4)에 의해서 스위칭 엑티버티와 파워소비를 계산하게된다. 그리고 함수 SAFN(Scheduling Algorithm For Nodes)를 호출하여 스위칭이 최소로 발생하고, 값이 변하지 않는 변수를 같은 컨트롤 스텝에 위치시켜 평션유닛에서 발생하는 스위칭 엑티버티를 줄이는 알고리즘을 제안한다. 제안하는 알고리즘의 초기 입력으로 들어오는 DFG는 그림 1 이다.

DSP 방정식의 루프 예

```
for ( i = 0 ; i < n ; i++){
    y[n] = Σ_{i=0} h0*X[n-i]+
           Σ_{i=0} h1*X[n-i];
}
```

위의 루프에서 y[n]에 대한 연산과정은 피연산자 h0 항과 h1항이 각각 교차하며 수행된다. 즉, h0+h1+h0+h1 과 같은 연산을 수행하게 된다. 피연산자의 상관관계를 높이기 위해 h0항 다음의 연산을 h0항으로 스케줄링 하는 것이다.

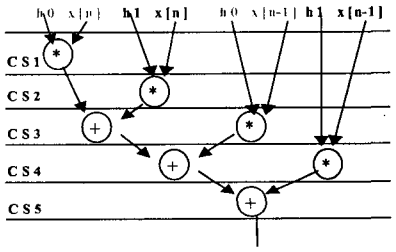

그림 1. 알고리즘 적용 전 DFG

각각의 데이터는 곱셈 연산 후 덧셈기의 입력으로 들어간다. 노드 A와 B는 각각의 곱셈 연산을 의미하고 교환 법칙에 의해 A노드와 B노드의 위치가 교환되었음을 알 수 있다. 즉 각각의 곱셈 연산이 스케줄일 될 때 상관관계가 높은 피연산자를 연속적으로 평션유닛의 입력으로 들어와도 같은 결과를 가지는 것을 의미한다. 알고리즘 적용 후 CDFG는 그림 2 와 같다.

```
A + B = B + A                    -- 식(6)
=> (h1 * x[n-1])  = A
=> (h0 * x[n-1])  = B

Out1 = (h0 * x[n]) +(h1 * x[n]) +
       (h0 * x[n-1]) + (h1 * x[n-1])
Out2 = (h0 * x[n]) + (h0 * x[n-1]) +
       (h1 * x[n]) + (h1 * x[n-1])
Out1 = Out2
```

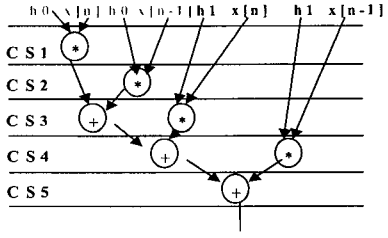

그림 2. 알고리즘 적용 후 CDFG

그림 3 은 알고리즘 적용 후 생성된 구조적 레벨을 의미하고 여기서 제약 조건은 곱셈기 한 개와 덧셈기 한 개를 사용하였다. 식(5)에서 *hi*에 대해 변하는 값 지연신호(X[n-i])를 계속적으로 곱해 가는 구조가 그림 3 이다. 하나의 곱셈기에 입력 데이터가 계속 적으로 들어오고, 곱셈한 결과는 반복적으로 더해져 결과를 얻는다.

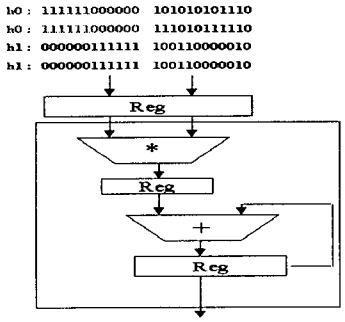

그림 3. 피연산자 스케줄링 후 입력

그림 4 는 스케줄링 후 평션유닛(곱셈기)에 입력으로 들어오는 데이터들의 비트별 스위칭을 나타내었다. 각 데이터는 초기 입력과 중간의 한번만 스위칭이 나타나는 것을 볼 수 있다.

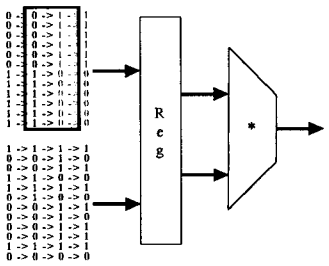

그림 4. 평션유닛에 12비트 데이터 입력 예

스케줄링 목적은 평션유닛에 대한 입력 변수의 재사용의 가능성을 최대로 하는 것이다. 이것은 두 개의 연산자를 같은 기능장치에 연속적으로 수행시킴으로서 평션유닛 한 개의 입력 데이터를 고정시켜 스위칭 변화가 일어나지 않도록 스케줄링 한다.

제안하는 알고리즘을 적용할 경우 일반적인 계산에서 연속적인 값을 사용할 경우 상당한 효과를 기대할 수 있다. 스위칭이 일어나지 않는 계수를 목표로 삼아 효율적인 스케줄링을 한다. 본 알고리즘에서 입력의 비트수가 많을수록 효과적이고 수행 속도 또한 빠르다. 주요 목적인 저전력 설계면에서 스위칭을 최소로 하였기 때문에 전력 소비는 최소화될 수 있다.

IV. 제안하는 알고리즘을 이용한 스위칭 확률 계산

입력 피연산자가 12비트 데이터일 경우 평션유닛에서의 스위칭 확률계산은 식(3)을 적용하여 아래와 같다.

표 1. 최악의 경우 피연산자 입력 데이터 예

h0 :	1 1 1 1 1 1 0 0 0 0 0 0
	▼ ▼ ▼ ▼ ▼ ▼ ▼ ▼ ▼ ▼ ▼ ▼
h1 :	0 0 0 0 0 0 1 1 1 1 1 1
	▼ ▼ ▼ ▼ ▼ ▼ ▼ ▼ ▼ ▼ ▼ ▼
h0 :	1 1 1 1 1 1 0 0 0 0 0 0
	▼ ▼ ▼ ▼ ▼ ▼ ▼ ▼ ▼ ▼ ▼ ▼
h1 :	0 0 0 0 0 0 1 1 1 1 1 1

평션유닛에 첫 번째 데이터 h0가 입력으로 들어오고, 두 번째 데이터 h1이 입력으로 들어오면 h0에 대응되는 각각의 데이터 비트는 100% 스위칭이 일어나는 것을 알 수 있다.

표 2. 최악의 경우 피연산자 입력 데이터를 알고리즘 적용 후의 입력 데이터 예

h0 :	1 1 1 1 1 1 0 0 0 0 0 0
	▼ ▼ ▼ ▼ ▼ ▼ ▼ ▼ ▼ ▼ ▼ ▼
h0 :	1 1 1 1 1 1 0 0 0 0 0 0
	▼ ▼ ▼ ▼ ▼ ▼ ▼ ▼ ▼ ▼ ▼ ▼
h1 :	1 1 1 1 1 1 0 0 0 0 0 0
	▼ ▼ ▼ ▼ ▼ ▼ ▼ ▼ ▼ ▼ ▼ ▼
h1 :	1 1 1 1 1 1 0 0 0 0 0 0

최악의 경우 입력 데이터에 대한 스위칭 엑티버티 계산

$$\alpha = (P| 1(h0)-0(h1) | + P| 0(h1)-1(h0) | + P| 1(h0)-0(h1) | + 1) * 100 / n$$
$$= (1+1+1+1)*100/4 = 100\%$$

최악의 경우 입력 데이터에 대한 알고리즘 적용 후 스위칭 엑티버티 계산

$$\alpha = (P| 1(h0)-1(h1) | + P| 1(h1)-0(h0) | + P| 0(h0)-0(h1) | + 1) * 100 / n$$
$$= (0+1+0+1)*100/4 = 50\%$$

위의 예에서 4개의 입력 데이터를 예로 들었다. 만약 더 많은 데이터가 입력으로 들어올 경우 제안하는 알고리즘을 적용 후 스위칭 확률은 줄어들고, 그에 따른 전력 소비도 감소될 것이다.

V. 결과

본 논문에서는 현대의 전자 제품에 필수 요소인 DSP를 위한 저전력 상위 레벨 합성 과정에서 새로운 알고리즘을 제안하였다. DSP는 산술연산을 수행하는 평션유닛의 수가 많고, 곱셈기와 같은 유닛이 소모하는 전력이 전체회로의 전력소비의 큰 부분을 차지한다. 전체 전력 소모를 줄이기 위해 평션유닛이 소모하는 전력을 줄이는 것은 큰 효과가 있다.

본 논문에서 제안하는 알고리즘을 적용한 실험은 일반적인 파워공식을 제외한 스위칭 엑티버티에 의한 실험만 하였다. 본 논문의 예를 설명한 것과 같이 최악의 데이터 입력을 제안하는 알고리즘을 통하여 개선된 결과를 보였다.

본 논문은 하나의 평션유닛에 입력으로 들어오는 피연산자가 스케줄링 되기 전의 파워 소비와 스케줄링 후 파워 소비를 측정한 결과 최고 50%이상의 전력 감소효과를 가진다.

참고문헌

[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 Yourk: Mc-Graw Hill, Inc., 1994.

[3] Sriram Govindarajan,"Scheduling algorithms high level synthesis," Term Paper ECE 834, Course : Digital Design Environments

[4] A. P. Chandrakasan and R. W. Brodersen, "Minimizing power consumption in digital CMOS circuits," Procs. of the IEEE, vol. 83, pp. 498-523, Apr. 1995

[5] M. C. Mcfarland, A. C. Parker, R. Camposano, "The high level synthesis of digital systems," Procs of the IEEE. Vol 78. No 2, Fed., 1990

[6] J. M. Chand and M. Pedram, "Register allocation and binding for low power," in Procs. of the ACM/IEEE Design Automation Conference, pp. 29-35, June 1995.

[7] 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.

[8] E. Musoll and J. Cortadella, "Scheduling and resource binding for low power," in Procs. ISSS, pp.104-109, 1995

[9] E. Musoll and J. Cortadella, "High-level synthesis technique for reducing the activity of functional units," in Procs. Int. Symp. on Low Power Design, pp. 99-104, 1995.