

## 상위 수준에서 행해지는 전력 소모 절감의 원리, 도구와 방법

서울대학교 장래혁\*

### 1. 개요

#### 1.1 전력 소모와 저전력 시스템

디지털 시스템의 전력 소모를 줄이려는 노력은 아주 오래 전부터 중요한 설계요소로서 자리를 잡아왔으나, 최근 들어 배터리를 사용하는 시스템이 늘어가면서 전력 소모 절감에 대한 요구가 더욱 절실해졌다. 전력 소모의 절감이란 파워(Power, W)를 줄이는 것과 에너지(Energy, J)를 줄이는 것으로 구분되는데, 많은 경우에 이를 혼용하여 사용하는 것을 볼 수 있다. 엄밀히 말하면, 파워를 줄이는 것은 단위 시간에 소모되는 전력 양을 줄이는 것으로, 전원으로 공급되는 전류를 줄이는 것을 말하고, 에너지를 줄이는 것은 하고자 하는 일을 모두 마치는데 소요되는 총 에너지를 줄이는 것을 말한다.

전력 소모를 줄이는 것은 궁극적으로 배터리의 수명을 연장하기 위한 것이 주된 목적이므로, 파워를 줄이는 것보다는 에너지를 줄이는 것이 직접적인 효과를 가져온다. 그러나, 실제로 파워를 줄이는 것도 직, 간접적으로 영향을 주므로, 배터리의 수명을 연장하는 방법은 간단히 설명되지 않는다. 파워를 줄이면 전력을 소모하는 주체인 각각의 소자들의 발열이 줄어들게 되므로, 열 방출을 위하여 유지되어야 하는 열 용량을 줄이는 것이 가능해 저서 소자의 부피를 줄일 수 있고 결과적으로 비용이 줄어들게 되며 배터리의 방전특성에 따라 보다 많은 전력을 뽑아낼 수 있게 되어 많은 긍정적인 효과를 가져온다. 그러나 일반적으로는 파워를 줄이는 반면 디지털 시스템의 동작속도가 느려지는 효과를 가져와 같은 일을 하는데 걸리는 시간이 늘어나게 되면 궁극적으로 에너지

의 절감을 가져오지 않는 경우가 생길 수 있다. 따라서, 일반적으로 배터리의 수명을 연장하는 방법은 에너지를 줄이는 방법으로 이해하면 큰 무리가 없다. 따라서 본 글에서는 배터리의 수명을 연장하는 방법으로 같은 일을 수행하는데 소요되는 총 에너지를 줄이는 것을 설명한다.

#### 1.2 상위 수준에서 행해지는 전력 소모 절감의 원리

전력 소모를 절감하는 노력은 디지털 시스템을 설계하는 각 단계에서 모두 행해질 수 있다. 소자를 만드는 입장에서 가능한 한 불필요한 전력 소모를 줄이도록 소자를 설계하고, 보다 적은 전력으로 빠른 스위칭 동작을 할 수 있도록 노력해왔다. 실제로 CMOS 소자의 출현은 상당한 기간동안 디지털 시스템을 설계하는 데 있어서 전력 소모에 대한 부담을 덜 수 있게 해주었다.

상위 수준과 하위 수준에서의 디자인이란 상대적인 개념으로 이를 명확하게 나눌 수는 없다. 그렇지만 여기에서는 이미 구현된 라이브러리로 설계하는 반도체나, FPGA를 포함하여 하드웨어 시스템을 설계하는 것, 시스템 소프트웨어와 운영체제 등을 최적화하여 전력 소모를 줄이는 것을 상위 수준의 전력 소모 절감 방법이라 정의하고 설명하기로 한다.

상위 수준에서 행해지는 전력 소모 절감 방법은 이미 만들어진 소자의 전력 소모 특성을 적극적으로 이용하여, 동일한 일을 하는데 소요되는 전력 소모를 줄이는 것이다. 상위 수준에서 정의 되는 일을 하위 수준까지 구체화하게 되면, 그 방법은 한가지 유일한 방법만이 존재하는 것이 아니라 여러 가지 다른 방법을 따를 수 있게 되는 경우가 일반적이므로, 이중에서 가장 전력 소모가 적은 방법을 취하게 되면 되는

\* 중신회원

것이다. 현재까지 일반적으로 시스템의 최적화에서는 전력 소모라는 성능지표가 적용되는 경우가 적었으므로, 기존의 방법을 사용하는 것보다 전력 소모가 적은 경우에는 이를 상위 수준에서 행해지는 전력 소모 절감이라고 말한다. 경우에 따라서는 상위 수준에서 정의 되는 일 자체를 전력 소모가 적은 구체화된 구현 방법이 도출될 수 있도록 하는 노력도 행해지며, 이와 같이 보다 높은 수준에서부터 전력 소모를 의식하고 시스템을 설계하는 경우, 기대할 수 있는 전력 소모의 절감 효과는 더욱 커지게 된다.

이제, 상위 수준에서 행해지는 전력 소모 절감에는 어떠한 정보가 필요한가 생각해 보자. 상위 수준에서 시스템의 설계를 바꾸게 되면, 분명히 하위 수준까지의 구현이 달라지게 되어 시스템의 전력 소모가 달라지게 된다. 그런데 현재까지 대개 전력 소모에 대한 분석은 소자를 만드는 하위 수준에서만 행해졌으므로, 결국 상위 수준의 설계 변경에 따른 전력의 변화 추이를 알려면 하위 수준에서 동작하는 내용을 모두 알아야 된다. 그러나 이렇게 되면 그 복잡도가 다루기 힘든 수준이 되며, 특히 상위 수준에서 설계하는 사람들이 가지고 있지 않은 하위 수준의 지식이 없이는 전력 소모를 설계에 반영하는 것이 불가능해진다. 결과적으로, 상위 수준에서 전력 소모의 절감을 피하려면 상위 수준에서의 설계 변경과 시스템의 전력소모의 변화를 직접적으로 설명해 줄 수 있는 전력소모의 정보가 있어야 한다는 결론을 얻을 수 있다.

그렇다면 이러한 정보는 어디에서 얻을 수 있으며, 누가 만드는가 생각해 볼 필요가 있다. 이를 위하여 자동차를 예로 들어 본다. 연료비가 상대적으로 비싼 우리나라에서는 이왕이면 연비가 좋은 자동차를 유지하는 것이 유리하다. 고 연비를 실현하려면 일단 연비가 좋은 차를 구매하여야 한다. 자동차 메이커에서는 이미 연비가 경쟁력이라는 것을 오래 전부터 인식하였으므로 연비가 좋은 자동차를 생산하려고 많은 노력을 들이고 있다. 일단 연비가 좋은 자동차를 구매하고 나면, 이제는 운전자의 책임이다. 운전자는 연비가 좋게 되는 운전 방법을 개발하여 이를 실천하여야 한다. 연비가 좋은 운전 방법이란 무엇인가? 급가속을 적게 하고 불필요한 짐을 싣지 말고 필요 없이 고속으로 달리지 말아야 한다. 이러한 일반론은 어느 자동차에나 적용될 수 있으나, 엄밀히 말하면 성능의 저하를 가져온다는 것을 알 수 있다. 보다 무거운 짐을 싣고도 보다 민첩하게 차가 움직여주고,

필요에 따라서 고속 주행도 가능한 것이 성능이 좋은 차이다. 연비를 좋게 하기 위하여 차의 성능을 무척 대고 저하한다면, 애초부터 성능이 낮은 차를 사고 그 차액으로 연료비를 사용하는 것이 더 경제적인 해결책이 될 수 있다. 즉, 연비를 좋게 하려고 출퇴근이 20분이 더 걸린다면, 이것이 과연 올바른 방법인가는 선불리 판단하기 어려운 일이라는 것이다.

이러한 관점에서 보면 진정한 연비의 절감이란 먼저 만족해야 할 최소의 성능, 예를 들면 출퇴근에 총 걸리는 시간의 최대값이 주어진 경우에 이를 만족하는 한에서 가장 높은 연비를 실현하는 것이라고 보면 된다. 총 20킬로미터의 거리에, 중간에 신호등이 10개가 있으며, 길이가 각각 500미터 800미터인 언덕이 두개가 있고, 언덕의 상승 경사는 3%, 5%에 하강 경사는 2%, 5%라는 등등, 기타 여러 가지 주행하여야 하는 길의 정보의 분석이 먼저 필요하다. 신호에 많이 걸리게 되면 총 주행에 걸리는 시간이 많이 걸려 나머지 구간에서 속도를 높여야 하므로 결국 고 연비를 실행할 수 없다. 그러므로 일부 구간에서는 신호에 걸리지 않도록 가속을 많이 하여 고속 주행을 하는 것이 최적의 답을 가져올 수도 있는 것이다.

연비가 같다고 주장하는 두개의 차종에서도 엔진의 토크 특성, 변속기의 각 단의 기어비, 타이어의 지름, 브레이크의 효율 등이 당연히 다를 수밖에 없다. 그러므로 이제 연비가 같다는 것은 어느 특정한 운전 조건 하에서의 이야기이고, 내가 주행해야 하는 20킬로미터의 코스를 최적으로 주파하는데 필요한 운전 방법은 전혀 다를 수 있다는 것을 알 수 있다. 그렇다면 이제 독자는 최적의 연비를 실현하기 위하여 필요한 정보가 엄청나다는 것을 짐작할 수 있을 것이다. 먼저 주행해야 하는 코스의 완벽한 분석, 신호등의 주기 등, 경우에 따라서는 시간에 따른 차량의 흐름과 분포를 모두 알아야 한다. 경우에 따라서는 사전에 아는 것이 불가능한 정보도 많다. 이는 내가 수행해야 하는 프로그램의 수행시간을 예측하는 것과 비슷한 상황이다. 따라서 사전에 알 수 있는 정보는 최대한 수집하고, 결국 실제와 다른 경우에는 이에 대신할 수 있는 차선의 정보를 활용하여야 한다. 두 번째로는 운전하는 상황에 따른 차량의 연비에 대한 정보를 정확히 알아야 한다. 차량의 가속도에 따른 연료 소비를 정확히 알기 위해서는 차량의 무게, 승객 및 짐의 무게, 경사도, 노면에 따라 바퀴에 걸리는 부하, 바람에 따른 차체에 걸리는 부하, 가속을 시작하

는 시점에서의 속도, 엔진 회전수, 변속기의 변속 상태 등의 다양한 변수에 따른 연료 소비 정보를 얻어 내야 한다. 이는, 현재 동작하는 마이크로프로세서의 상황에 따른 전력 소모의 변화를 아는 것에 비유할 수 있다.

그렇다면 위의 정보를 자동차의 메이커에서 얻을 수 있는가? 아마 이와 같은 정보를 알려줄 수 있는 자동차 메이커는 없을 것이며, 이는 결국 자동차의 운전자가 많은 실험을 통하여 뽑아내야 하는 정보이다. 다시 말해서, 고 연비의 자동차(결국 일반적으로 고 연비를 실현할 가능성이 있는 자동차)를 개발하기 위해서 소요되는 정보와, 특정한 코스를 주파하기 위한 고 연비의 운전 방법을 개발하기 위하여 필요한 정보는 무척 다르다는 것이다. 마찬가지로 우리가 구매한 부품의 전력 소모에 대한 이러한 정보도 메이커에서 공급해 주기를 바라는 것은 불가능한 실정이다. 결국, 이는 상위 수준에서 전력 소모를 절감하려고 하는 설계자가 구해야 하는 정보이다.

## 2. 디지털 시스템의 전력 소모

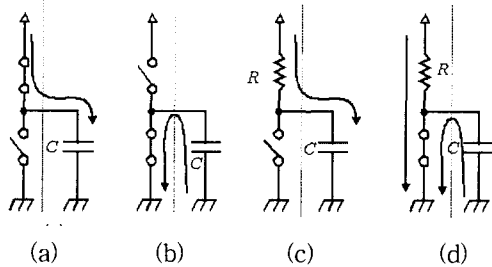


그림 1 스위칭 동작에 따른 전력 소모

디지털 소자가 on/off 동작을 할 때에 소모하는 전력은 회로의 구성에 따라서 여러 가지 다양한 형태로 나타난다. 이 중에서 가장 대표적인 것은 회로 내부에 존재하는 기생 캐패시턴스를 충전/방전할 때 소요되는 전력이다(그림 1, (a) (b)). 회로 내부의 실제 구성을 고려하지 않는 상위 수준에서의 전력 소모 모델로 가장 흔히 사용되는 것이 이 모델이기 때문이다. 실제로 정적(Static) CMOS 소자로 구성된 회로에서는 최근 들어 공급 전압과 천이 전압(Threshold voltage)이 극단적으로 낮아지기 전에는 단지 스위칭 동작만을 고려함으로써 회로의 전력 소모에 대한 충분한 설명이 가능했다. 정적 CMOS 회로에서는 회로

의 실제 구성을 따지지 않더라도, 데이터가 이동하는 경로(RTL, Register Transfer Level)의 스위칭 동작으로 시스템의 전력 소모를 어느 정도 합리적으로 설명하는 것이 가능하므로 대부분의 상위 수준의 전력 소모 기술이 이를 토대로 행해졌다. 그런데 실제 시스템을 구성하는 경우에는, 정적 CMOS로 구성된 칩 내부의 전력소모 외에도 보드(Board, 인쇄회로기판) 상에서 구성되는 신호, 동적(Dynamic) CMOS로 구성된 소자, CMOS가 아닌 양극성(Bipolar) 소자 그리고 저항(Passive pull up) 등에서 소모되는 전력이 상당히 큰 비중을 보이는 경우가 흔하다. 이 경우의 전력 소모 모델은 그림 1의 (c) (d)와 같이 설명되는 것이 타당하며, 스위칭 동작이 곧 전력 소모를 설명해 주지 않는다.

상위 수준의 전력 소모 모델에서는 실제 회로가 어떻게 구성되었는지, 어느 소자로 구성되었는지, 저항을 이용한 풀업이 되어 있는지에 관계하지 않고 전력 소모를 표현하기를 원한다. 이는 이러한 것이 상위 수준의 개념에 맞는 것으로 설명될 수 있으나, 일반적으로 상위 수준에서 전력 소모를 다루고 싶은 사람들이 실제 회로의 구성에 대해서 알지 못하며, 알고 싶어하지 않는다는 것으로도 설명될 수도 있다. 그러나 상위 수준에서도 실제 회로가 어떻게 구성되었는지 전혀 알지 못하면서 단지 스위칭 동작으로 전력 소모를 설명하려고 한다면, 많은 경우에 전력 소모를 제대로 설명하지 못하게 되고, 결과적으로 고안된 전력 소모 절감의 노력이 실제 응용에서는 제대로 효과를 발휘하지 못하는 결과를 낳을 수도 있다.

### 2.1 동적 전력 소모

그림 1의 (a)는 디지털 소자가 0에서 1로 출력이 전환되는 경우의 전력 소모를 설명해준다. 이전에 출력은 0의 값을 가지고 있었으므로 캐패시터 C는 완전히 방전된 상태로 있으며, 출력이 1로 전환되는 순간부터 공급전압(VDD)만큼으로 충전되게 된다. 결과적으로 스위칭이 완전히 끝나고 나면 캐패시터에는  $1/2 C VDD^2$ 의 전력이 저장되고, 같은 양의 전력이 손실로 소요된다. 반대로 1에서 0으로 출력이 바뀌는 경우에는 캐패시터에 충전되어 있던  $1/2 C VDD^2$ 의 전력이 손실로 없어지게 된다. 이와 같은 에너지는 회로의 출력이 변화되는 경우에만 소모되므로 이를 동적 전력(Dynamic energy) 소모라 한다.

이렇게 정적 CMOS로 구성된 경우의 동적 전력 소모는, 회로가 동작할 때 새로운 출력 값이 이전 출력 값과 다른 경우에만 전력이 소모되고, 새로 계산된 결과가 이전 값과 같은 경우에는 전력이 소모되지 않는다. 따라서 전력 소모는 신호의 해밍거리(Hamming Distance)에 비례하게 된다.

최근에는 이와 같은 정적 CMOS를 많이 사용하나, 아직도 상당한 경우에 동적 CMOS를 사용함으로써 소자의 속도를 높이고, 전력 소모의 절감을 꾀하며 트랜지스터의 개수를 줄이는 설계를 많이 한다. 동적 CMOS는 항상 충전(Precharge)을 한 후에 계산(Evaluation)을 하는데, 회로의 구성에 따라서 출력이 1이거나 0이면 계산 시에 방전이 일어난다. 즉, 출력이 1인 경우에 방전이 일어나게끔 구성된 동적 CMOS 회로에서는 출력이 계속해서 1이면 회로는 계속 전력을 소모하게 된다. 만일 같은 동작을 정적 CMOS에서 수행하게 되면 출력이 계속해서 1인 경우에는 전력을 소모하지 않으므로, 동적 CMOS나 정적 CMOS냐에 따라서 동적 전력 소모의 양상은 크게 달라지게 된다. 대개의 CMOS 회로에서는 출력의 0과 1의 값 차이가 VDD와 GND사이의 차와 같으므로 전력 소모는 VDD<sup>2</sup>에 관계하게 된다. 만일 출력 값이 VDD와 GND의 사이에서 변한다면 전력 소모는 보다 복잡한 형태를 띠게 된다.<sup>1)</sup>

## 2.2 정적 전력 소모

양극성 소자의 경우에는 출력이 0인 경우에 소자에서 공급전압(VCC)<sup>2)</sup>에서 공통접지(GND)로 지속적으로 전류가 흐르게 된다. 이 전류는 회로의 스위칭 동작에는 무관하며, 출력이 0으로 유지되는 한 계속 흐르게 된다. 따라서 회로의 스위칭 동작이 없더라도 출력이 계속해서 0으로 유지되면 최대한의 정적 전력을 소모하게 된다. 또, 보드 상에서 구현되는 많은 신호가 고 임피던스(High impedance) 상태에서 신호가 개방(Floating)되는 것을 막거나, 보드상의 신호선의 임피던스를 맞추어 주기 위해서 저항을 신호선과 공급전압<sup>3)</sup> 사이에 삽입하는 경우가 흔하

1) VD대신 (VOH-VOL)/VDD에 비례하게 된다. VOH와 VOL은 출력이 1인 경우와 0인 경우의 전압이다.  
 2) FET로 구성된 회로의 경우 공급전압을 VDD라 표시하고 양극성 회로에서는 공급전압을 VCC라 표시한다.  
 3) 일부 회로에서는 공급 전압 대신에 일정한 DC 전압원에 연결하기도 한다.

다. 이러한 저항은 역시 출력 값이 0인 경우에 많은 정적 전류가 공급전압으로부터 흐르도록 하는 경로를 제공한다.

전력 소모 면에서 비교적 최적화가 잘 되어 있는 회로에서는 소모되는 전력이 주로 동적 전력 소모가 된다. 이와는 반대로 고성능을 위한 시스템에서는 정적 전력 소모가 동적 전력 소모를 능가하는 경우도 많이 볼 수 있다. 최근 들어서는 전력 소모를 줄이기 위해서 지속적으로 공급전압을 낮추는 노력의 부작용으로 신호의 값에 관계없이 서브스트레이트(Substrate)로 소모되는 전력이 급격히 증가하고 있다. 이러한 손실(Leakage) 전력은 소자의 게이트가 얇아지면서 이곳을 통하여 흐르기도 한다. 손실 전력은 상수 값을 가지고 꾸준히 소모되며 칩이 전체적으로 수면(Idle 또는 Power down) 상태로 들어가는 경우에 이를 줄이는 설계가 많이 행해진다.

- HDD(Hamming-Distance dependent Dynamic)energy
  - 스위칭 빈도
  - 정적 CMOS
  - 기생 캐패시턴스
- WDD(Weight dependent Dynamic)energy
  - 출력 값
  - 동적 CMOS
- WDS(Weight dependent Static)energy
  - 출력 값
  - 양극성 소자 또는 풀업 저항
- Leakage energy
  - 반도체 내부에서 고정적으로 흐르는 형태

그림 2 상위 수준에서 본 전력 소모의 유형

## 3. 전력 소모 예측을 위한 도구

### 3.1 디지털 소자의 전력 소모

하위 수준에서 전력 소모를 절감하는 연구는 그림 2에 설명된 개개의 전력 소모 요소의 값 자체를 줄이는 것을 목적으로 한다. 예를 들면 HDD를 줄이기 위해서는 기생 캐패시턴스의 값을 줄이거나 공급 전압을 낮춤으로써 같은 동작을 하는데 소요되는 HDD를 줄이게 된다. 이에 반하여, 상위 수준에서 행해지는 전력 소모 절감의 노력은 HDD와 같은 값 자체를 줄이는 것이 아니라 대신 스위칭 빈도 자체를 줄여서 HDD 전력을 줄이는 방법이 된다. 마찬가지로, 같은 의미의 동작을 수행하면서도 WDD와 WDS의 값이

적어지도록 논리 값을 할당함으로써 WDD나 WDS를 줄인다. 손실 전력은 불필요하게 디바이스가 활성화되지 않도록 함으로써 최소화할 수 있다.

예 1) 어떤 변수  $x$ 가 있다.  $x$ 는 처음에는 0이 할당되고, 이어서 주로 1씩 증가하는 형태를 보인다. 일반적으로  $x$ 는 이진(Binary) 인코딩되는 것이 일반적이거나, 회로가 정적 CMOS로 구성된 경우, 이를 그레이(Gray) 인코딩하면, 같은 동작을 수행하는 경우 많은 HDD 전력을 절감할 수 있다.

예 2) 어떤 신호  $x$ 는 외부로부터 요청이 들어오면 활성화 된다. 만일 외부로부터 요청이 평소에 거의 들어오지 않는 상황이고, 이 신호가 출력이 풀업 저항이 있는 양극성 소자로 구성되어 있다면, 신호  $x$ 는 활성화된 경우의 값을 0으로 함으로써, 1로 된 경우보다 많은 WDS 전력을 줄일 수 있다.

### 3.2 전력 소모 예측을 위한 도구

소자 수준에서 전력 소모의 예측과 분석은 회로의 정밀한 모의 실험이나 측정을 통하여 얻을 수 있다. 하위 수준에서의 전력 소모 분석을 위한 도구는 매우 널리 존재하며 상당한 정확도를 가지고 소자의 전력 소모를 분석하여 준다. 이에 반하여 상위 수준의 동작 기술은 시스템 전체의 전력 추이를 설명하기에는 너무 정보가 부족하기 때문에 상위 수준의 동작을 토대로 시스템의 전력 소모를 설명하기란 매우 어렵다. 그렇지만 상위 수준에서 전력 소모를 최적화 하기 위해서 소자 수준의 전력 소모 정보를 제공하는 도구를 사용하는 것은 복잡도 면에서 바람직하지 못한 방법이 된다.

따라서 상위 수준에서 전력 소모를 기술하고, 이를 토대로 다른 상황에서의 전력 소모를 예측하기 위한 노력의 결실로 여러 가지 도구가 개발 되었다. 상위 수준에서 전력 소모를 예측하기 위해서는 모의 실험이나 측정에 의존하거나, 필요 시 이들을 혼용하기도 한다. 대표적인 시스템 수준의 전력 소모 예측 도구 중에서 모의 실험에 의존하는 것으로는 Simple Power [1, 2]와 Wattch [3]라는 도구를 들 수 있다. 이 두 가지 도구는 모두 컴퓨터 구조 실험에 널리 사용되는 SimpleScalar라는 모의 실험 도구를 토대로 개발되었다. SimpleScalar는 RTL 수준에서 마이크로프로세서와 메모리 계층을 모의 실험 해준다. 여기에 각 동작에 소요되는 전력의 값을 대응 시켜서 시

스템의 전력 소모를 예측한다. 이들 도구의 특징은 가상의 RISC 프로세서의 시뮬레이터에 독자적으로 스위칭 캐패시턴스를 입력하여 전력소모를 유추한다는 점이다. 이러한 도구의 장점으로는 100% 시뮬레이션에 의한 도구이므로 사용이 편리하고, 보급이 용이하며, 사용자가 어렵지 않게 시뮬레이터를 수정할 수 있는 등 일반적인 시뮬레이션 도구의 장점을 보여 준다. 그러나, 가상의 프로세서를 기초로 한 것과, 현실과 거리가 있는 RTL 수준에서의 부분별 캐패시턴스에 의한 전력 소모 유추로 인하여 부적절한 결과를 유도한다는 것이 단점이다.

이와는 반대로, 직접 측정에 의한 도구를 사용하게 되면 비교적 신뢰성 있는 결과를 얻을 수 있다. 이러한 목적으로 개발된 도구로는 컴패사에서 만든 Itsy라는 포켓 컴퓨터에 테스트 포인트를 뽑아내어 범용 측정도구로 전력 소모를 측정할 수 있도록 만든 것이 있다[4]. 이들의 단점은 범용 도구로는 원하는 수준으로 전력 소모에 대한 정보를 추출하기 어렵다는 것이다.

### 3.3 SNU에서 개발한 전력 측정 기법

서울대학교 컴퓨터공학부의 연구실에서는 최근 수년간, 시스템 수준의 전력 소모 절감 연구를 위하여 전력 소모 분석 도구를 개발하는데 주력하여왔다. 이 도구는 특별히 개발된 방법에 의한 사이클별 전력 소모 측정과, 정적 전력 소모와 동적 전력 소모를 구분하여 동작하는 유한 상태 기계(FSM, Finite State Machine)에 기반 한 전력 소모 모델, C 언어로 작성된, 메모리와 주변장치의 동작 및 전력 소모 파악을 위한 시뮬레이션 도구로 구성되어 시스템 수준에서 각 부분의 사이클별 전력 소모를 매우 정확하게 분석하여 준다.

#### 3.3.1 사이클 별 전력 소모 측정 회로

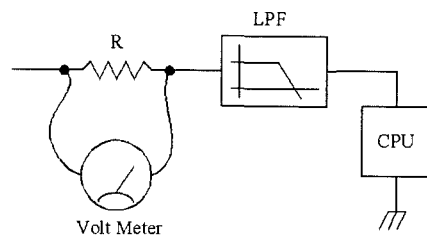


그림 3 멀티미터를 사용한 고전적인 전력 측정 방법

그림 3은 멀티미터를 사용하여 소비전력을 측정하는 고전적인 테스트 셋업을 보여준다. 작은 직렬 저항 R을 전원라인에 삽입하고 여기에 발생하는 전압차를 읽음으로써 소비전력을 구한다. 보통 멀티미터의 동작 한계 주파수는 수십 KHz를 넘지 못하므로, 정확한 측정을 위해서는 측정하고자 하는 회로(여기서는 CPU)와 전원 공급선 사이에 저역 대역 통과 필터(LPF, Low-pass Filter)를 달아주어야 한다. 사실 이와 같은 측정 셋업은 피측정 장치의 시간에 따른 소비 전류의 변화가 거의 없는 경우에만 실용적으로 사용이 가능하다. 그러나 디지털 회로와 같은 경우에는 동작하는 때 클럭 사이클별로 매우 다른 양의 전력을 소모하게 되므로 일반적으로는 측정이 불가능하다. 관련 연구에서는 CPU로 하여금 매우 긴 길이의 무한 루프를 수행하게 하고 루프 안에는 동일한 명령어로 채움으로써 해당 명령어를 수행하는 동안 소요되는 전력을 측정하였다[5].

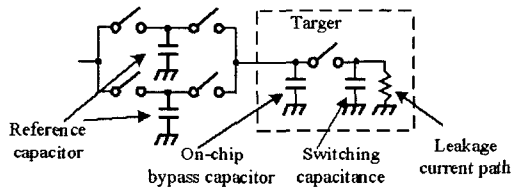


그림 4 사이클별 전력소모 측정 회로

그림 4에서는 서울대학교에서 고안한 사이클별 전력소모 측정 회로를 도시한다[6, 7]. 그림 3에서 직렬 저항을 삽입한 대신에 두개의 스위치를 삽입하고 그 사이에 기준이 되는 캐패시터를 병렬로 연결해 둔다. 먼저 뒤쪽 스위치를 열어둔 채로 앞쪽 스위치를 닫아 캐패시터를 충전한다. 이 동안에는 아래쪽 캐패시터가 타겟 회로에 전원을 공급하도록 되어 있다. 충전한 후에는 충전된 전압을 측정한다. 이론적으로는 당연히 VDD만큼으로 충전될 것이나 스위치의 내부저항이나 기생 인덕턴스 등으로 실제로 충전된 전압과 VDD와는 차이가 있다. 다음 사이클에서는 앞 스위치를 개방과 동시에 뒤 스위치를 연결하여 타겟 회로를 충전된 캐패시터로 동작시킨다. 동작이 완료가 되면 캐패시터의 전압 측정을 통해 남아 있는 전하량과 사용된 전력을 계산할 수 있다. 즉, 사용된 전력은 충전된 전력에서 남아 있는 전력을 빼면 된다. 충전된 전압을  $V_0$ , 방전된 전압을  $V_1$ . 기준이 되는 레퍼런스

캐패시터의 용량을  $C_R$ 이라고 하면,

$$E = \frac{1}{2} C_R (V_0^2 - V_1^2)$$

이 성립하게 된다.

그런데 실제로 이 회로를 적용하려면, 반도체 내부에 존재하는 바이패스 캐패시터의 용량을 고려하여 이를 보정해 주어야 한다. 우리가 구하고자 하는 전력 소모는 곧 내부의 스위칭 캐패시터의 용량을 구하는 것이 된다. 구하고자 하는 스위칭 캐패시터의 용량을  $C_L$ 이라고 하고, 바이패스 캐패시터 용량을  $C_B$ 라고 하자. 스위치를 연결하는 순간 타겟 회로가 동작하지 않더라도  $C_B$ 와  $C_R$ 이 같은 전압을 유지하기 위해서 전하가 이동하게 된다. 따라서  $C_R$ 에 전압  $V_0$ 는 순간적으로 약간 감소하게 되고, 이를  $V_{01}$ 이라고 하고  $C_B$ 가  $V_{01}$ 이 되기 이전 즉 이전 스위칭에서 방전된 상태에의  $C_B$ 의 전압을  $V_1$ 라 하면(이 때  $V_1$ 은 이전 주기에서 방전된 상태에서의 반대쪽  $C_R$ 의 전압과 같다). 전하량 보존의 법칙에 의하여,

$$C_B(V_{01} - V_1) = C_R(V_0 - V_{01})$$

이 성립한다. 따라서 에 의하여 소모된 전력은

$$E = \frac{1}{2} (C_R + C_B) (V_{01}^2 - V_1^2)$$

로 정확하게 계산된다.

그리고 마지막으로, 일부 고성능 회로에서는 회로 내부의 스위칭에 관계없이 일정한 량의 전류를 흘리도록 하는 설계를 채택하기도 한다. 이 전력은 VDD에서 GND로 연결된 저항으로 모델링되는데, 이는 회로의 동작이 끝나더라도 의 전압을 계속적으로 하강시킨다. 시간에 따르는 전압 하강을 측정하게 되면 이와 같은 전력도 측정할 수 있다.

### 3.3.2 자동 측정 장치

앞 장에서 소개한 사이클별 전력 측정 회로를 사용하여 실제 빠른 속도로 동작하는 디지털 소자의 전력 소모를 측정하려면, 그림 4의 회로를 자동으로 제어하고, 캐패시터의 전압을 자동으로 측정하여 저장하는 부가 회로가 있어야 한다. 따라서 타겟 회로와 동기되는 클럭을 이용하여 스위치를 제어하고 고속의 A/D 변환기를 부가하여 캐패시터의 전압을 측정 한 후에, 이를 고속 메모리에 저장하여 거의 실시간으로 측정할 수 있도록 한다. 서울대학교에서는 고속 FPGA(Field Programmable Gate Array)를 사용하여 스위치와 A/D 변환기, 메모리를 자동 제어하는 회로를 구성하고 측정 도구의 성격에 맞는 통신 기능

을 부가한다. 이러한 구성을 통하여 모든 측정과정을 컴퓨터가 주관하고 측정 후에 손쉽게 결과를 분석할 수 있는 기능을 추가한 다양한 형태의 측정 도구를 개발하였다.

### 3.4 SNU에서 개발한 전력 측정 도구

#### 3.4.1 SEC(SNU Energy Characterizer)

SEC(SNU Energy Characterizer)는 칩(Chip) 단위로 전력 소모를 측정하는 도구이다. 칩 외부에서 제어 신호로 칩을 적절히 제어하면서 클럭 사이클별로 전력을 측정하여 향후 이를 모델링 할 수 있도록 하는 도구이다.

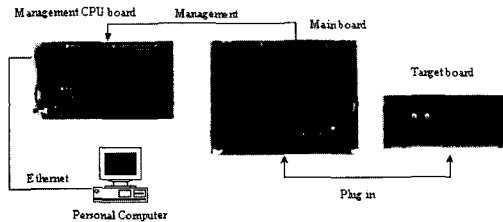


그림 5 SEC의 구성도

그림 5는 SEC의 구성도를 보여준다. SEC는 여러 가지 종류의 타겟을 지원하기 위해서 타겟 보드가 메인보드의 커넥터에 삽입되는 구조를 하고 있다. 원활하고 능률적인 측정을 위해서 범용 CPU 보드를 연결하여 제어 보드를 관리하고, 고속의 네트워크 연결을 통하여 PC와 통신을 하도록 되어 있어, 다량의 측정 데이터를 고속으로 전달할 수 있다.

그림 6의 소프트웨어는 자동 측정을 담당하는 소프트웨어가 ARM7 RISC 프로세서에서의 전력 소모 분석을 수행하는 과정을 보여준다[7, 8, 9]. 먼저 어셈블리 프로그램을 가지고 이진 코드를 얻는다. 이 이진 코드가 타겟으로 설정된 ARM7 프로세서가 수행할 명령어 세트이며 이를 먼저 측정 시스템에 다운로드 한다. 다음으로는 실제로 ARM7 프로세서를 동작시키면서 측정을 하는데, 검은 바탕의 창에 보이는 파형이 측정된 캐패시터의 전압이다. 마지막으로 이를 PC로 업로드한 후에 전압 정보를 전력 정보로 변환하고 원래의 벡터와 대조 시켜서 명령어와 전력 소모를 연관시키는 작업을 한다. 마지막으로 이를 적절히 도시화하여 연관성을 찾는 것이 마지막 윈도우의 화면이다.

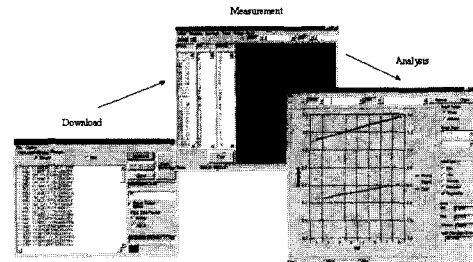


그림 6 SEC 소프트웨어

#### 3.4.2 SES(SNU Energy Scanner)

SEC가 하드웨어 수준에서 디바이스의 전력 소모를 모델링 하는데 필요한 도구라면, SES는 이를 ARM7 프로세서에 특화 시키고 사용을 보다 편리하게 만들어서 소프트웨어 설계자가 전력 소모에 대한 정보를 활용하고 싶을 때 유용한 도구이다[10].

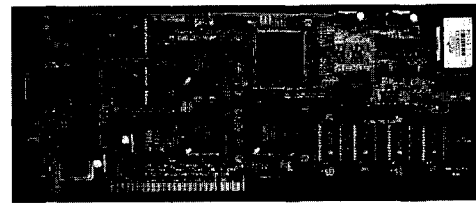


그림 7 SES 도구

그림 7과 같이 SES는 PC에 삽입되는 카드 형태이며, 카드를 삽입함으로써 하드웨어 설치는 모두 끝난다. 이 하드웨어에는 16MB 용량의 벡터 메모리가 있어서 실제로 커다란 크기의 응용프로그램을 그대로 수행시키면서 전력 소모를 분석할 수 있다. 방대한 데이터를 얻는 관계로 보다 빠른 통신 경로가 필요하여 PCI 버스 형식의 카드로 구현되었다. SES는 Linux가 설치된 PC에서 동작하며, 표준 형식의 디버거와 응용프로그램에 삽입하여 원하는 부분의 전력만 측정할 수 있도록 유용한 시스템 함수를 제공한다.

#### 3.4.3 SEE(SNU Energy Explorer)

SEE(SNU Energy Explorer)는 SES 하드웨어와 PC 상에서 운용되는 시스템 수준의 에너지 시뮬레이터, SEC가 결합된 형태의 도구로, 시스템 전체의 전력 소모를 분석하는데 사용된다. 먼저, 분석하고자 하는 시스템을 구성하는 하드웨어 소자의 전력 소모 모델을 SEC로 측정한다. 추출된 전력 정보는 C언어로 작성된 디바이스의 동작을 유한상태기계로 모델

링 한 함수에 삽입되어 시뮬레이터와 결합된다. 이 상태로 SEE의 준비가 완료된다. 새로운 구성의 시스템이나 새로운 소자가 추가되면 먼저 SEC로 모델링한 후에 시뮬레이터를 수정하여 대응한다. 이 상태에서 측정하고자 하는 응용프로그램을 컴파일하여 SES에 다운로드하여 수행시키면, SES에서는 프로그램이 수행되는 동안 발생하는 프로세서와 L1캐쉬 사이의 모든 메모리 트랜잭션을 저장하여 PC로 전송한다. 물론 ARM7 프로세서가 소모한 전력의 정보도 전송된다. SES에서 보내주는 전력 정보는 클럭당 전력 정보이므로 여기에 동작 주파수를 곱함으로써 다양한 클럭 주파수에서의 소비 전류를 계산할 수 있다. 현재 탑재된 프로세서는 ARM7이나 다른 구조의 RISC 프로세서라 하더라도, 비트폭이 32비트로 같고 코드의 밀도가 비슷하고 레지스터의 개수가 비슷하다면 결국 보여주는 메모리 트랜잭션은 거의 같다고 할 수 있으므로, 이 결과를 쉽게 다른 프로세서의 결과로 가공할 수 있다. 이 경우에는 마이크로프로세서의 자체 전력 소모는 데이터시트를 참조한 평균전력으로 계산한다. SEE는 시스템 수준의 전력 소모를 분석하는데 목적이 있으므로, 실제로 10% 안팎의 전력을 차지하는 마이크로프로세서의 코어를 평균하여 사용한다고 하더라도 주어진 목적을 달성하는 데는 문제가 없다. 두 번째 단계로 획득한 메모리 트레이스를 캐쉬 시뮬레이터 모듈을 통과시킨다. 캐쉬 시뮬레이터는 다양한 캐쉬 구성을 모두 지원하므로, 설계 임의의 구성을 실험할 수 있으며 동시에 어떠한 구성이 적절한가를 판단하는 실험도 용이하게 할 수 있다. 그 다음 단계로는 메모리 버스와 메모리 디바이스의 시뮬레이터를 통과시켜서 이들의 전력 소모를 자세히 분석해 낸다. 아울러 필요 시 각종 주변장치의 전력 소모도 같은 방법으로 분석한다. 그림 8은 SEE의 구성도를 보여준다.

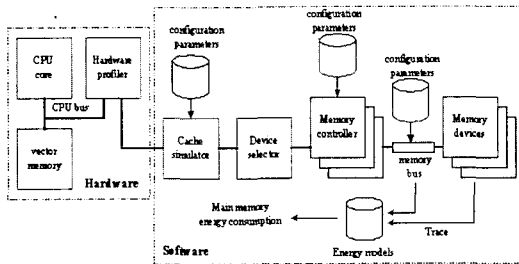


그림 8 SEE의 구성도

#### 4. 상위 수준의 전력 소모 절감

서울대학교에서 개발한 도구로 수행할 수 있는 시스템 수준의 전력 소모 절감은 특별히 언급할 수 없을 만큼 다양하며 제약이 없으나 여기서는 이해를 돕기 위하여 간단한 예를 든다.

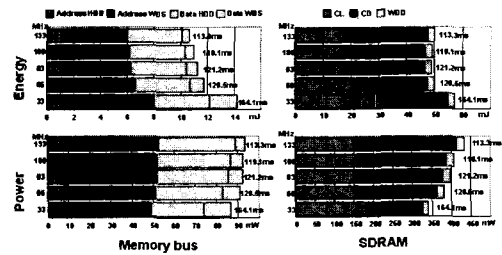


그림 9 SDRAM 메모리 시스템의 전력 소모 분석

그림 9에서는 SEE의 도구를 사용하여 SDRAM으로 구성된 메모리 시스템의 전력 소모를 분석한 결과의 일부이다[11, 12, 13]. 여기서는 206MHz로 동작하는 32비트 RISC(StrongArm) 프로세서, 8KB의 독립된 Instruction 캐쉬와 Data 캐쉬 메모리, LVT 시스템 버스로 구성된 시스템 상에서 디지털 카메라 응용인 JPEG 압축 프로그램을 수행시키는 경우에 적절한 메모리 클럭 주파수를 찾아보는 실험이다. 메모리 클럭의 주파수가 높을수록 버스에서 소모되는 시간당 소비전력은 증가하나 결국 전체 프로그램을 수행하는데 드는 전력은 감소되는 것을 보여준다. 그런데 SDRAM 소자 자체는 83MHz에서 최적의 전력 소모를 보여준다. 버스에서 소모되는 전력은 SDRAM 소자에서 소모되는 전력의 약 25% 정도가 되므로 이를 적절히 고려하여 100MHz로 메모리 버스 클럭을 설정하는 것이 좋을 것이라는 결론을 얻을 수 있다.

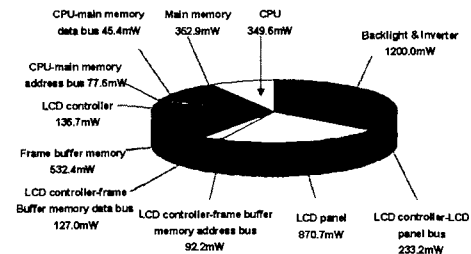


그림 10 PDA에서의 전력 소모 분포

그림 10에서는 흔히 사용하는 PDA에서의 전력 소



모 분포를 보여준다[14]. 전력 소모의 분포는 어떠한 응용프로그램을 수행하는가에 매우 관계가 깊으며, 평균적으로 무엇이 얼마나 소모한다는 등의 정보는 크게 도움이 되지 않는다. 그림 10은 실제로 MPEG4의 디코더에서 비디오 부분만을 수행시키는 경우의 부분별 평균 전력 소모를 자세히 보여준다. 이와 같은 정보는 실제로 측정을 통해서도 매우 구하기 어려운 것이나, SEE를 통하여 매우 쉽게 이러한 정보를 획득할 수 있어 저전력 시스템을 연구하는데 큰 경쟁력을 얻을 수 있다.

## 5. 결 론

시스템 수준에서 이루어지는 전력 소모 절감 연구는 응용에 따라서 그 효과가 매우 크므로 많은 사람들이 관심을 가지고 있다. 그러나, 현존하는 범용 도구로는 원하는 연구를 수행할 수 있는 정보를 얻기가 쉽지 않아, 몇몇 연구기관에서 각자 고안된 툴을 사용하고 이의 일부만 공개되는 실정이다. 따라서 서울대학교에서는 기존의 어떤 도구보다 정확한 데이터를 생산해 내는 동시에, 빠르고 손쉬운 실험을 할 수 있는 여러 가지 도구를 개발하였다. 현재 일부 도구는 공개 단계에 있으므로, 저전력 설계에 관심이 있는 분들은 <http://power-reduction.snu.ac.kr> 또는 <http://www.power-reduction.com>, <http://cselab.snu.ac.kr/member/naehyuck>을 참조하거나, naehyuck@snu.ac.kr로 문의하면 된다.

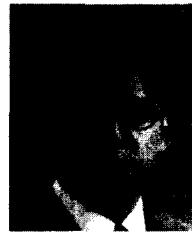
## 참고문헌

- [1] Ye, W., Vijaykrishnan, N., Kandemir, M. and Irwin, M.J., "The design and use of simplePower: a cycle-accurate energy estimation tool," in Proceedings of Design Automation Conference 2000, pp. 340-345, 2000.
- [2] Vijaykrishnan, N., Kandemir, M., Irwin, M.J., Kim, H.S. and Ye, W., "Energy-driven integrated hardware-software optimizations using SimplePower," in Proceedings of the 27th International Symposium on Computer Architecture, pp. 95-106, 2000.
- [3] Brooks, D., Tiwari, V. and Martonosi, M., "Wattch: a framework for architectural-level power analysis and optimizations," in Proceedings of the 27th International Symposium on Computer Architecture, pp. 83-94, 2000.
- [4] K.I.Farkas, J.Flinn, G.Back, D.Grunwald, and J.M.Anderson, "Quantifying the Energy Consumption of a Pocket Computer and a Java Virtual Machine," in Proceedings of SIGMETRICS, pp. 252-263, 2000.
- [5] V. Tiwari, S. Malik, and A. Wolfe, "Power analysis of embedded software: a first step towards software power minimization," in Proceedings of 1994 IEEE/ACM international conference on Computer-aided design, pp. 384-390, 1994.
- [6] Naehyuck Chang and Kwanho Kim, "Real-time per-cycle energy consumption measurement of digital systems," IEE Electronics Letters, vol. 36, no. 13, pp. 1169-1171, June, 2000.
- [7] Naehyuck Chang, Kwanho Kim, and Hyun Gyu Lee, "Cycle-Accurate Energy Consumption Measurement and Analysis: Case Study of ARM7TDMI," in Proceedings of ISPLED 2000, pp. 185 - 190, July, 2000.
- [8] Naehyuck Chang, Kwanho Kim, and Hyun Gyu Lee, "Cycle-Accurate Energy Consumption Measurement and Analysis: Case Study of ARM7TDMI," IEEE Transactions on VLSI Systems, Vol. 10, pp. 146-154, April, 2002.
- [9] Sheayun Lee, Andreas Ermedahl, Sang Lyul Min, and Naehyuck Chang, "An Accurate Instruction-Level Energy Consumption Model for Embedded RISC Processors," in Proceedings of LCTES 2001, pp. 1-10, Snowbird, UT, June, 2001.
- [10] Dongkun Shin, Hojun Shim, Yongsoo Joo, Han-Same Yun, Jihong Kim and Naehyuck Chang, "SES: A highly integrated energy monitoring tool for low-power embedded programs," accepted for publication in IEEE Design and Test of Computers, 2002.
- [11] Yongsoo Joo, Yong Seok Choi, Hojun Shim, Hyung Gyu Lee and Naehyuck Chang, "Energy Exploration and Reduction of

- SDRAM Memory Systems,” in Proceedings of DAC 2002, pp. 892–897, New Orleans, June, 2002.
- [12] Hojun Shim, Yongsoo Joo, Yongseok Choi, Hyung Gyu Lee, and Naehyuck Chang, “Low-Energy Off-Chip SDRAM Memory Systems for Embedded Applications,” accepted for publication of the ACM Transactions on Embedded Computing Systems, Special Issue on Memory Systems, 2002.
- [13] Naehyuck Chang, Kwanho Kim, Heonshik Shin, and Jinsung Cho, “Bus Encoding for Low-Power High-Performance Memory Systems,” in Proceedings of DAC2000, pp. 800–805, June, 2000, Los Angeles, USA.
- [14] In Seok Choi, Hojun Shim and Naehyuck Chang, “LowPower Color TFT LCD Display for HandHeld Embedded Systems,” in Proceedings of ISLPED 2002, pp. 112–117, August 2002, Monterey, California, USA.

---

### 장 래 혁



1996 서울대학교 제어계측공학과 학사,  
석사, 박사  
1997 미시간 대학교 (Ann Arbor 소재)  
Research Fellow  
서울대학교 컴퓨터공학부 조교수  
관심분야: 저전력 시스템, 내장형 시스템,  
컴퓨터 하드웨어, 디지털 시스템  
E-mail: naehyuck@snu.ac.kr

---

### ● 제29회 정기총회 및 추계학술발표회 ●

- 개최일자 : 2002년 10월 25일(금) ~ 26일(토)
- 개최장소 : 수원대학교
- 상세장소 : www.kiss.or.kr
- 문 의 처 : 학회 사무국 Tel. 02-588-9246/7