

# 명령어 레벨의 비동기식 프로세서 소비 전력 모델

이제훈<sup>1\*</sup>

<sup>1</sup>강원대학교 삼척캠퍼스 전자정보통신공학부

## Instruction-level Power Model for Asynchronous Processor

Je-Hoon Lee<sup>1\*</sup>

<sup>1</sup>Division of Electronics and Information Communication Eng., Samcheok Campus,  
Kangwon National University

**요약** 본 논문은 비동기식 프로세서를 위한 새로운 명령어 레벨 소비 전력 모델을 제안한다. 최근까지 SoC에 내장 되는 프로세서의 소비 전력을 예측하기 위한 다양한 소비 전력 모델들이 제안되었으나, 모두 동기식 프로세서들을 타겟으로 구현되었기 때문에 비동기식 프로세서에 적용할 경우 정확성이 떨어진다. 이러한 문제를 해결하기 위하여 비동기식 회로의 동작 특성을 반영한 새로운 비동기식 프로세서 소비 전력 모델을 제안하였다. 제안된 소비 전력 모델은 비동기식 8051 프로세서, A8051의 소비 전력 특성을 반영하여 구현되었고 게이트 레벨의 합성한 결과를 이용한 소비 전력 예측 결과와 비교하여 성능 평가를 수행하였다. 제안된 소비 전력 모델의 예측 결과는 게이트 레벨의 소비 전력 측정 결과와 비교하여 90.7%의 정확도를 보였고, 1,900 배 이상 시뮬레이션 시간을 단축하였다.

**Abstract** This paper presents the new instruction-level power model for an asynchronous processor. Until now, the various power models for estimating the power dissipation of embedded processor in SoC are proposed. Since all of them are target to the synchronous processors, the accuracy is questionable when we apply those power models to the asynchronous processor in SoC. To solve this problem, we present new power model for an asynchronous processor by reflecting the behavioral features of an asynchronous circuit. The proposed power model is verified using an implementation of asynchronous processor, A8051. The simulation results of the proposed model is compared with the measurement result of gate-level synthesized A8051. The proposed power model shows the accuracy of 90.7% and the simulation time for estimation the power consumption was reduced to 1,900 times.

**Key Words** : Power estimation, Power model, Asynchronous circuit, Instruction-level power model

### 1. 서론

최근 DSM (deep sub-micron) 공정 기술의 발전으로 칩의 복잡도가 크게 증가하고 있다. 시스템 설계 방법도 전체 시스템을 하나의 보드로 구현하는 SoB (system-on-board) 설계 대신 하나의 칩에 집적하는 SoC (system-on-chip) 설계가 일반화되고 있다. 기존 SoC 설계는 전체 시스템을 하드웨어와 소프트웨어 설계로 구분한 후, 하드웨어를 먼저 설계하고 소프트웨어를 설계하였다. SoC 설계 속도를 높이기 위해서 최근 플랫폼 기반

SoC 설계 방법이 일반화되고 있다. 이는 내장형 프로세서와 여러 주변장치들을 하나의 범용 플랫폼으로 구성한 후 하드웨어와 소프트웨어 설계를 동시에 진행하여 설계 속도를 향상시킨다[1, 2]. 내장형 프로세서를 이용한 소프트웨어 설계는 하드웨어 설계에 비해 설계 변경이 쉽고 빠른 시뮬레이션이 가능하다. 최근 전체 SoC 설계에서 소프트웨어 설계가 차지하는 비중이 점차 증가하기 때문에 내장형 프로세서와 SoC의 성능 및 소비 전력간의 연관성도 점점 커지고 있다.

SoC 성능 향상을 위해 높은 클럭 주파수를 갖는 고성

본 연구는 교육과학기술부와 한국연구재단의 지역혁신인력양성사업으로 수행된 연구결과임 (2012H1B8A2026055). 그리고 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(2011-0013219)."

\*Corresponding Author : Je-Hoon Lee

Tel: +82-10-3258-2480 email: jehoon.lee@kangwon.ac.kr

접수일 12년 04월 09일

수정일 12년 06월 19일

게재확정일 12년 07월 12일

능 프로세서를 SoC에 집적하거나, 여러 프로세서들을 하나의 SoC에 집적하는 MPSoC (multi-processor SoC)가 사용된다. SoC의 소비 전력을 줄이기 위해서 클럭 게이팅 (clock gating) 및 DVFS (dynamic voltage frequency scaling)와 같은 저전력 설계 기술이 적용된 프로세서를 사용한다. 또한, 프로세서는 설계 방식에 따라 동기식 설계와 비동기식 설계로 나뉜다. 동기식 프로세서는 클럭을 이용하여 동작의 필요성 여부와 관계없이 동시에 모든 모듈들을 구동하고, 클럭 역시 가장 긴 지연을 갖는 블록의 동작 완료 시점을 기준으로 결정되어, 최악 성능 지연 (worst-case performance)을 갖는다. 반면 비동기식 프로세서는 핸드셰이킹 프로토콜을 이용하여 동작이 필요한 블록만을 선택적으로 구동시킨다. 블록별로 서로 다른 지연 시간으로 구동되어, 평균 성능 지연 (average-case performance)을 갖는다. 소비 전력 측면에서도, 동작이 필요한 블록만을 구동시키고, 동작이 불필요한 블록들은 자동적으로 대기 상태로 유지시켜 소비 전력의 낭비를 줄인다. 그러나, 비동기식 프로세서는 설계가 복잡하고 설계 환경이 부족하기 때문에, 최근까지 널리 사용되지 않고 있다.

저전력 SoC 설계가 점차 중요해지면서, SoC에 비동기식 프로세서를 집적하려는 노력이 계속되고 있다[3-6]. S. Furber가 제안한 AMULET3i가 상용 SoC에 집적되었고, Handshake Solution사의 비동기식 ARM, ARM966HS가 ARM사의 라이선스 제품군에 추가되었다. 325MHz 클럭을 사용하는 동기식 ARM 프로세서가 350MIPS의 성능과 163mW의 평균 소비 전력을 갖는다. 반면, ARM966HS 프로세서의 동작속도는 83MIP로 느리지만, 평균 소비 전력이 3.6mW로 기존 동기식 프로세서에 비해 45배 이상 낮아진다. 이와 같은 비동기 프로세서의 저전력 특성 때문에 향후 저전력 SoC에 널리 집적될 것으로 예측된다.

SoC 설계 순서는 다음과 같다. 첫 번째로, 구현할 SoC의 스펙을 결정한 후, SystemC와 같은 하이-레벨 언어를 이용하여 기능 모델을 설계하여 알고리즘을 검증한다. 두 번째로, 전체 시스템을 H/W와 S/W로 설계 파트로 나눈 후, H/W 및 S/W 동시 설계를 수행한다. H/W 파트는 상위 수준의 동작 검증 후 RTL 혹은 게이트 레벨 합성을 통해 FPGA 레벨에서 검증을 수행하고, S/W 파트는 컴파일러와 같은 S/W 인프라를 이용하여 검증한다. 세 번째로, H/W와 S/W의 개별 검증이 끝난 후 H/W와 S/W의 통합 시뮬레이션 툴을 이용하여 전체 SoC의 검증을 수행한다. 네 번째로, 통합 검증이 완료된 후 칩 제작과정을 거쳐 최종적으로 칩 테스트를 수행한다.

SoC의 소비 전력을 줄이기 위해서는 다양한 소비 전

력 모델을 이용하여 SoC 설계 단계별로 소비 전력을 예측하고, 예측된 소비 전력이 허용 가능한 소비 전력보다 클 경우 해당 설계 단계 혹은 그 이전 단계로 돌아가 반복적으로 재설계해야 한다. SoC 설계 초반에 정확하게 소비 전력을 예측한다면 SoC 설계 과정 동안 반복되는 재설계 과정을 크게 줄일 수 있다. S/W 파트의 경우 프로세서 소비 전력 모델을 이용하여 프로그램을 실행하는 동안 프로세서가 소비하는 전력을 예측한다. 현재 SoC에서 사용되는 프로세서 소비 전력 모델은 모두 동기식 프로세서를 대상으로 한다. 비동기식 프로세서를 내장한 SoC에 동기식 프로세서의 소비 전력 모델을 적용하면 정확도가 크게 떨어진다. 따라서, 비동기식 프로세서의 소비 전력 모델을 구현하고 이를 이용하여 SoC 설계 초반에 소비 전력 예측할 필요가 있다.

본 논문은 비동기식 프로세서의 동작 특성을 반영한 소비 전력 모델을 개발하였다. 제안된 비동기식 소비 전력 모델은 시스템 레벨과 같은 SoC 설계 초반에 소비 전력을 빠르게 예측할 수 있도록 명령어-레벨 모델로 구성하였다. 성능 평가를 수행하기 위하여 제안된 소비 전력 모델은 미리 개발된 A8051 프로세서에 적용하여 소비 전력을 예측한 후, 게이트 레벨로 합성된 A8051 프로세서와 PrimePower와 같은 게이트-레벨의 상용 소비 전력 예측 툴의 결과와 정확도 및 시뮬레이션 시간을 비교 평가를 수행하였다.

본 논문의 구성은 다음과 같다. 2장은 비동기식 시스템의 동작 특성과 이에 따른 소비 전력 특성에 대해 설명하고, 3장에서는 제안하는 비동기식 프로세서의 명령어 레벨 소비 전력 모델에 대해 설명한다. 4장에서는 모의실험을 통해 제안하는 소비 전력 모델의 성능을 분석하며, 마지막으로 5장에서 결론을 맺는다.

## 2. 비동기식 프로세서의 소비 전력 특성

비동기식 회로는 핸드셰이킹 프로토콜을 이용하여 동작이 필요한 블록만을 구동한다. 핸드셰이킹 프로토콜은 이웃한 블록간에 데이터선외에 요구 (request)와 확인 (acknowledge) 신호를 추가로 사용한다. 이웃한 블록의 동작 개시가 필요할 경우 동작 시작을 알리는 요구 신호를 전송하고, 이를 수신한 블록은 연산을 시작한다. 연산이 완료되면 확인 신호를 요구 신호를 전송한 블록으로 보내 동작이 종료되었음을 지시한다. 비동기식 회로의 핸드셰이킹 프로토콜은 그림 1에 나타난 것처럼, 2상 및 4상 프로토콜로 나뉜다. 그림 1(a)의 2상 프로토콜은 4상 프로토콜과 달리 RTZ (return to zero)가 없기에 동작이

빠르나, 4상 프로토콜에 비해 안정적인 데이터 전송이 어렵다는 단점을 갖는다.

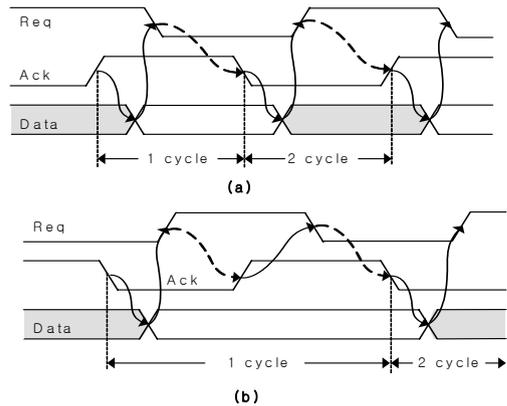
비동기식 회로의 각 모듈들은 요구 신호를 수신한 후 각자의 연산 지연 시간에 따라 구동되기 때문에 연산 완료 시점이 모두 다르다. 각 모듈은 연산 완료 시점을 스스로 검출해야 한다. 이를 위해 비동기식 회로는 특별한 타이밍 가정을 갖고, 이를 지연 모델 (delay model)이라 한다. 표 1처럼, 비동기식 회로의 지연 모델은 유한 지연 (bounded delay)과 미지 지연 (delay insensitive) 모델로 나뉜다. 유한 지연 모델은 소자와 배선 지연이 유한하며, 지연 시간을 모두 알고 있다고 가정한다. 유한 지연 모델을 사용하는 비동기식 회로는 모듈들을 게이트 레벨로 합성한 후, 정밀한 포스트 레이아웃 (post-layout) 시뮬레이션을 통해 모든 모듈의 최대 동작 시간을 추출하고, 이보다 약간 큰 지연을 갖는 지연 소자 (delay element)를 생성한다. 지연소자의 통과 지연 시간은 모듈의 동작 시간보다 크며, 이를 통해 각 모듈의 동작 완료 시점을 검출한다.

미지 지연 모델을 갖는 비동기식 회로는 소자나 배선의 지연은 유한하나 상한치를 알 수 없다고 가정한다. 미지 지연 모델을 갖는 비동기식 회로는 데이터 전송시 특별한 인코딩 기법을 이용하여 타이밍 정보를 데이터에 싣고 동시에 전송한다. 지연소자를 구성하는 대신 연산 타이밍을 인코딩을 통해 데이터와 같이 전송하기 때문에 추가적인 전송선과 동작 완료를 검출하는 회로가 필요하다는 단점을 갖는다.

비동기식 프로세서는 동기식 프로세서와 구분되는 소비 전력 특성을 갖는다. 가장 큰 차이는 비동기식 프로세서는 동작이 필요한 데이터패스를 실행시켜 전력을 소비하고, 동작이 불필요한 데이터패스는 대기 상태로 유지하여 소비 전력 낭비를 막는다. 최근 프로세서는 성능 향상을 위해 계층 구조 캐쉬, 복잡한 파이프라인 구조 채용으로 프로세서 복잡도가 크게 증가하고 있다. 프로세서의 복잡도가 증가할수록 프로그램 실행중에 동작이 불필요한 모듈의 수가 증가한다. 따라서, 동작이 불필요한 블록을 대기 상태로 유지시켜 대기 전력만을 소비할 경우 전력 낭비를 크게 줄일 수 있다. 더 나아가 비동기식 프로세서의 주요 블록들을 더욱 세분화하여 명령어에 따라 실행되는 데이터패스를 분리할 경우 소비 전력을 크게 낮출 수 있다.

그림 2는 동기식 프로세서와 비동기식 프로세서의 소비 전력 특성을 나타낸다. 동기식 프로세서는 시스템 클럭의 상승 천이시 모든 블록이 구동되기 때문에 임펄스적인 소비 전력 특성을 갖는다. 비동기식 프로세서는 모든 모듈이 자기동기식으로 구동되어 각 모듈의 구동 시

간이 서로 달라 소비 전력도 분산된다. 따라서, 평균 소비 전력이 크더라도 최대 소비 전력은 오히려 동기식 프로세서보다 작다. 비동기식 프로세서의 소비 전력 특성은 동기식 프로세서와 크게 달라서, 동기식 프로세서의 소비 전력 모델을 비동기식 프로세서에 적용하면 예측된 소비 전력의 정확도가 크게 떨어진다.



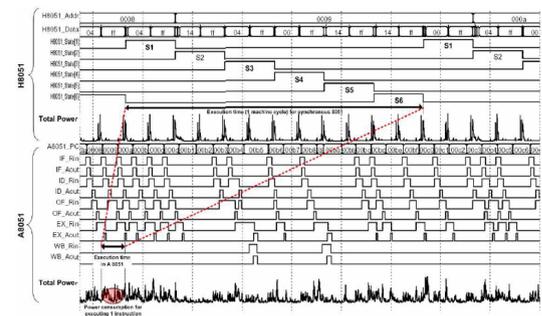
[그림 1] 핸드셰이킹 프로토콜: (a) 2상 프로토콜, (b) 4상 프로토콜

[Fig. 1] Handshaking protocol. (a) 2-phase protocol and (b) 4-phase protocol

[표 1] 지연 모델

[Table 1] Delay model

지연 모델	게이트 지연	배선 지연
Bounded	유한	유한
Speed independent(SI)	무한	0
Delay insensitive(DI)	무한	무한
Quasi delay insensitive(QDI)	무한	isochronic fork
Scalable DI(SDI)	무한	-



[그림 2] 동기식 프로세서 (H8051)와 자기동기식 프로세서 (A8051)의 소비 전력 분석

[Fig. 2] Power dissipation comparison between synchronous 8051 (H8051) and asynchronous 8051 (A8051)

### 3. 제안된 비동기식 소비 전력 모델

#### 3.1 A8051 프로세서의 동작 특성

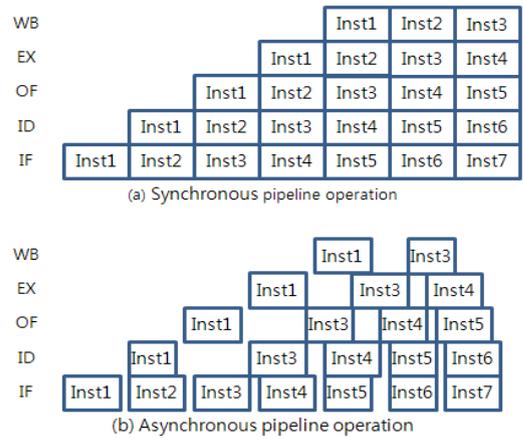
본 논문에서는 비동기식 프로세서의 소비 전력 예측 모델을 구성하기 위해 동기식 인텔 8051 프로세서의 비동기식 버전인 A8051 프로세서를 이용하였다. A8051의 명령어 집합은 인텔 8051과 호환되는 255개의 명령어로 구성되었고, 각 명령어는 인텔 8051과 마찬가지로 1, 2, 그리고 3 바이트의 서로 다른 명령어 길이를 갖는다[5-6].

A8051 프로세서의 명령어 실행 사이클은 인텔 8051과 크게 다르다. 인텔 8051의 명령어 실행 사이클은 머신 사이클을 기반으로 제어된다. 인텔 8051 프로세서의 모든 명령어들은 1, 2, 그리고 4 머신 사이클로 구동된다. 즉 인텔 8051의 모든 명령어 실행 사이클은 머신 사이클의 배수로 구성되어 프로세서 제어가 단순해진다. 반면에, 명령어별로 서로 다른 명령어 실행 사이클 때문에 파이프라인 구조를 채택하기 어렵고, 동작이 불필요한 내부 상태를 포함시켜 속도가 느리고 소비 전력을 낭비된다는 단점이 있다.

반면, A8051의 모든 명령어들은 RISC 프로세서처럼 모든 명령어가 하나의 명령어 실행 사이클을 갖는다. 대신, A8051 프로세서는 명령어마다 머신 사이클의 길이가 다르다. A8051 프로세서의 명령어 실행 사이클은 일반적인 RISC 프로세서처럼 명령어 페치(IF), 디코드(ID), 오퍼랜드 페치(OF), 실행(EX), 그리고 저장(WB)의 5단 파이프라인 구조에서 실행된다. 5단 파이프라인 구조를 갖는 일반적인 RISC 프로세서는 그림 3(a)에서 나타난 것처럼, IF 스테이지부터 WB 스테이지까지 순차적으로 실행되는 선형 파이프라인 실행 구조를 갖고, 모든 명령어 실행 사이클은 5개의 파이프라인 스테이지 동작을 포함한다.

A8051 프로세서의 5단 파이프라인을 이용한 명령어 실행 사이클은 일반적인 RISC 프로세서의 파이프라인 실행 사이클과 다르다. A8051의 명령어 실행 사이클에서 동작이 불필요한 버블 스테이트를 명령어 실행 사이클에서 제거하였다. 또한, A8051 프로세서의 명령어 실행 사이클은 WB 스테이지까지 구동하지 않더라도 중간에 명령어 실행이 종료되는 경우 이른 종료 (early termination)을 허용한다. 마지막으로, 모든 명령어가 선형 파이프라인 실행 구조를 갖지 않고, 일부 명령어는 지역적인 파이프라인 반복 실행을 허용한다. 결론적으로 A8051 프로세서의 모든 명령어가 하나의 명령어 실행 사이클로 구동되나, 그림 3에 나타난 것처럼, 모든 명령어는 서로 다른 길이의 명령어 실행 사이클을 갖는 7개의 명령어 그룹으

로 구성된다. A8051의 명령어 실행 사이클은 동작이 불필요한 버블 스테이지를 포함하지 않는다. 그룹 1, 2, 3 그리고 5의 경우 WB 스테이지전에 이른 종료를 허용한다. 그룹 1부터 그룹 4의 명령어들은 선형 파이프라인 구조로 실행되는 반면, 그룹 5에서 그룹 7의 명령어들은 선형 파이프라인 실행 대신 OF와 EX의 부분적인 반복을 허용하는 비선형 파이프라인 실행을 갖는다.



[그림 3] 동기식과 비동기식 파이프라인 동작 비교  
[Fig. 3] Comparison between synchronous and asynchronous pipeline operation

	Number of Instructions	Number of partial iterations	Instruction execution scheme
Group 1	22	0	IF
Group 2	13	0	IF-ID-EX
Group 3	93	0	IF-ID-OF-EX
Group 4	89	0	IF-ID-OF-EX-WB
Group 5	4	1	IF-ID-OF-EX-OF-EX
Group 6	6	1	IF-ID-OF-EX-OF-EX-WB
Group 7	28	2	IF-ID-OF-EX-OF-EX-OF-EX-WB

[그림 4] A8051 명령어 실행 사이클  
[Fig. 4] Instruction execution cycle of an A8051

#### 3.2 A8051 프로세서의 명령어 레벨 소비 전력 모델

최근 복잡한 모바일용 SoC 시스템에서의 타이트한 소비 전력 예산을 만족하는 S/W 설계를 위해서 SoC 설계 초반에 S/W 실행시 프로세서의 소비 전력을 예측하여 주어진 소비 전력 범위를 넘지 않게 설계해야 한다. 이를 위해 다양한 프로세서의 소비 전력 모델들이 개발되었다. V. Tiwari가 처음으로 프로그램이 실행되는 동안 프로세

서가 소비하는 전력을 예측할 수 있는 소비 전력 모델을 개발하였다. 이는 명령어 레벨의 소비 전력 모델로 프로그램이 실행되는 동안 그 안에 포함된 명령어의 평균 소비 전력과 연속된 명령어간의 에너지 오버헤드 그리고 프로세서의 파이프라인 스톱 및 캐쉬 미스로 인한 에너지 소비를 합하여 프로세서의 소비 전력을 예측한다. 명령어 레벨의 소비 전력 모델은 구성이 단순하고 소비 전력 예측을 위한 시뮬레이션 시간이 적어 SoC 설계 초기에 소비 전력을 쉽게 예측할 수 있는 반면 정확도가 높지 않다. 소비 전력 모델의 정확도를 높이기 위해서 아키텍처 레벨 혹은 마이크로 아키텍처 레벨과 같이 프로세서 구조를 세분화하여 소비 전력 파라미터를 추출하고 이를 통해 소비 전력 모델을 구성한다[7-11]. 그러나 소비 전력 모델의 파라미터의 수가 증가할수록 예측된 소비 전력의 정확도는 증가하는 반면, 소비 전력 예측에 걸리는 시간은 급격히 증가한다. 따라서, 프로세서의 소비 전력 모델은 추상화 레벨, 정확도, 시뮬레이션 시간, 그리고 개발 비용간의 트레이드-오프로 결정해야만 한다. SoC 시스템 설계 초반에 소비 전력을 예측하고 이 결과를 SoC 설계에 적용하기 위해서는 정확도를 손해 보더라도 빠른 소비 전력 예측이 중요하다.

이러한 이유로, 본 논문에서는 명령어 레벨의 소비 전력 예측 방법을 채택하였다. 제안된 프로세서 소비 전력 모델은 비동기식 프로세서의 동작 특성에 따라 소비 전력을 예측하여 비동기식 프로세서에 적합하게 구성하였고, 이를 프로그램이 실행되는 동안 실행되는 명령어에 따라 프로세서의 소비 전력을 즉시 예측하여 SoC 설계 초반에 S/W 파트에 의해 소비되는 전력을 예측하도록 하였다. 제안된 소비 전력 모델은 다음 두 단계로 구성되었다. 첫 번째 단계는 A8051 프로세서의 명령어 실행 사이클을 기반으로 명령어별로 실행 상태를 프로파일링한다. 두 번째 단계는 표준 셀 라이브러리 공정을 이용하여 A8051 프로세서를 합성한 후, Synopsys사의 PrimePower 툴을 이용하여 개별 명령어별로 5개의 내부 상태에서의 평균, 최저, 그리고 최대 소비 전력을 구하고, 이전 단계에서 결정된 프로파일과 대응시켜 소비 전력 모델을 완성한다. 자세한 단계별 소비 전력 모델 구현 방법은 아래와 같다.

명령어별 실행 상태 프로파일링 단계에서는 A8051 프로세서의 명령어별로 명령어 실행 사이클에 따라 명령어 실행 사이클에 포함된 내부 상태들의 동작 특성을 프로파일링한다. A8051 프로세서는 그림 4에 나타난 것처럼 IF, ID, OF, EX, 그리고 WB의 5개의 내부 상태를 갖고 이를 기반으로 모든 명령어가 하나의 명령어 실행 사이클로 구동된다. 그러나, 명령어별로 실행이 반드시 필요

한 내부 상태만을 구동시키기 때문에 명령어 실행 사이클에 포함된 내부 상태의 수가 서로 다르며 이는 명령어 별로 소비 전력이 차이가 나는 가장 큰 원인이 된다. 첫 번째 단계에서는 각 명령어별로 5개의 내부 상태들을 활성 상태 (active state, AS), 데이터 전송 동작만이 요구되는 버블 상태 (bubble state, BS), 마지막으로 이른 종료에 따른 대기 상태 (idle state, IS)로 구분한다. 활성 상태는 요구 신호를 수신한 후 해당 기능 블록들을 구동하여 정상적으로 전력을 소비한다. 버블 상태는 연산이 불필요하며 다음 상태로 필요한 데이터를 전송하는 동작만을 수행하며 이에 따라 소비 전력도 크게 낮아진다. 마지막으로 대기 상태의 경우 해당 명령어 실행이 종료되어 동작이 필요하지 않고 대기 상태를 유지하며 최소한의 대기 전력만을 소비한다. 따라서, 그림 4에서 보인 A8051 프로세서의 명령어 실행 사이클에 따라 각 명령어는 서로 다른 개수의 활성 상태, 버블 상태, 그리고 대기 상태를 갖고 식 (1)처럼 소비 전력 프로파일이 얻어진다.

$$I_{\alpha} = \sum_i^l AS_i + \sum_j^m BS_j + \sum_k^n IS_k \quad (1)$$

$\alpha$ -번째 명령어,  $I_{\alpha}$ 는 5개로 구분된 내부 상태중에서 해당 명령어의 명령어 실행 사이클에 따라, 동작이 요구되는  $l$  개의 활성 상태 집합,  $\sum AS_i$ ,  $m$  개의 버블 상태 집합,  $\sum BS_j$ , 그리고  $n$ 개의 대기 상태 집합,  $\sum IS_k$ 의 합으로 구성된다.

두 번째 단계인 소비 전력 모델 구현 단계에서는 명령어별 실행 상태 프로파일에 포함된 각 내부 상태에 실제 프로세서를 구동하여 얻어진 소비 전력을 대응시켜 소비 전력 모델을 구현한다. 이를 위해서는 RTL (register transfer level)로 구현된 A8051 프로세서에 0.35- $\mu$ m Hynix 표준 셀 라이브러리를 이용하여 합성하였다. Synopsys사의 PrimePower와 같은 게이트 레벨의 소비 전력 측정 툴을 이용하여 A8051 프로세서의 모든 명령어를 순차적으로 실행시키고 각 명령어 실행할 때 내부 상태의 소비 전력을 측정한다. 이 때 각 내부 상태에 포함된 모든 기능 블록의 소비 전력을 더하여 내부 상태별로 소비 전력을 구한다. 이 때, 각 내부 상태별로 활성 상태에서의 소비 전력 (PAS, power for active state), 버블 상태에서의 소비 전력 (PBS, power for bubble state), 그리고 대기 상태에서의 소비 전력 (PIS, power for idle state)를 각각 구하고 이를 이전 단계에서 구한 실행 모델의 내부 상태별로 각각 대응시켜 식 (2)와 같이 A8051 프로세서의 명령어 레벨 소비 전력 모델을 구현한다. 또한 각 소비 전력은 최대, 최소 그리고 평균 소비 전력을 각각 구하여 대응시킨다.

$$P_{\alpha} = \sum_{i=1}^l PAS_i + \sum_{j=1}^m PBS_j + \sum_{k=1}^n PIS_k \quad (2)$$

$\alpha$ -번째 명령어의 소비 전력,  $P_{\alpha}$ 는 5개의 내부 상태별로  $l$ 개의 활성 상태의 소비 전력의 합,  $\Sigma PAS_i$ ,  $m$ 개의 버블 상태에서의 소비 전력의 합,  $\Sigma PBS_j$  그리고  $n$ 개의 대기 상태에서의 소비 전력의 합,  $\Sigma PIS_k$ 를 모두 더해 얻어진다. 최종적으로 SoC 설계 초기에 S/W 설계의 소비 전력을 예측하기 위해 제안된 명령어 레벨의 프로세서 소비 전력 모델은 식 (3)과 같이 구현된다.

$$P_{Program} = \sum_{\alpha=1}^{N_p} [\sum_{i=1}^l PAS_i + \sum_{j=1}^m PBS_j + \sum_{k=1}^n PIS_k] \quad (3)$$

$N_p$ 는 프로그램이 실행되는 동안 실행되는 명령어의 수이며, 프로그램이 실행될 때, 프로세서가 소비하는 전력,  $P_{Program}$ 은 프로세서로 패치된 명령어의 소비 전력을 모두 합하여 프로세서의 소비 전력을 구한다. 또한 각 내부 상태의 소비 전력 프로파일을 최소, 최대 그리고 평균 소비 전력을 구하여 식 (2)에 대응시킬 경우 프로그램을 실행하는 동안 프로세서의 최소, 최대, 그리고 평균 소비 전력을 각각 예측할 수 있다.

#### 4. 실험 및 결과 분석

제안된 명령어 레벨의 소비 전력 모델을 검증하기 위해 다음과 같은 실험 환경을 구축하였다. VHDL로 설계된 A8051 프로세서를 Hynix 0.35- $\mu$ m 표준 셀 라이브러리를 이용하여 합성한 후, A8051의 모든 명령어를 포함하는 테스트벤치를 구성하였다. Synopsys사의 PrimePower 설계 툴을 이용하여 A8051 프로세서의 기능 블록별로 최소, 최대, 그리고 평균 소비 전력을 구한 후, 내부 상태별로 기능 블록들의 소비 전력을 합하여, 내부 상태별 소비 전력을 구하였다. 각 내부 상태는 명령어별로 활성, 버블, 그리고 대기 상태를 가질 수 있고 이를 구분하여 명령어 실행 프로파일링 결과와 매핑하였다. 마지막으로, 입력에 따라 소비 전력 예측 프로그램을 구성하였다. Dhrystone V2.1과 JPEG 테스트벤치를 입력하여 소비 전력을 예측하고 이 결과를 Synopsys사의 PrimePower 툴을 이용한 게이트 레벨의 소비 전력 측정 결과와 비교하여 성능을 평가하였다.

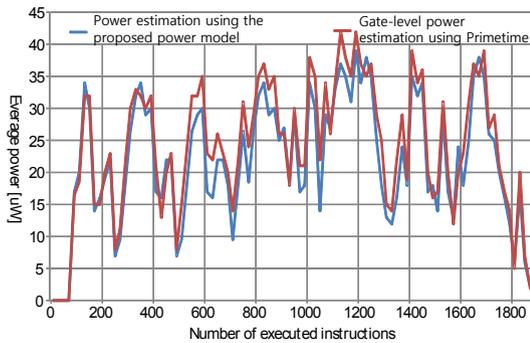
Dhrystone V2.1 벤치 프로그램을 실행하는 동안, 제안된 소비 전력 모델의 예측 결과와 게이트 레벨의 소비 전력 측정 결과와 비교한 결과는 그림 5에 나타내었다. 제

안된 소비 전력 예측 결과는 게이트 레벨의 소비 전력 측정 결과와 비교하여 90.3%의 정확도를 갖는다. 일반적인 동기식 프로세서의 명령어 레벨 소비 전력 모델들이 10%의 오차율을 갖는다. 따라서, 제안된 비동기식 프로세서를 위한 명령어 레벨 소비 전력 모델은 동기식 프로세서의 소비 전력 모델과 비슷한 정확도를 가짐을 알 수 있다.

본 논문에서 제안한 비동기식 프로세서의 소비 전력 모델의 시뮬레이션 시간과 Synopsys사의 PrimePower를 이용하여 게이트 레벨에서의 소비 전력을 예측 시뮬레이션 시간을 비교하여 표 2에 나타내었다. 테스트벤치 프로그램으로 Dhrystone V2.1과 JPEG 프로그램을 사용하였다. 소비 전력 예측은 시스템 레벨과 게이트 레벨 설계에서 각각 수행되었다. 제안된 소비 전력 예측 방법은 게이트 레벨의 소비 전력 측정 시간과 비교하여 Dhrystone V2.1의 경우 약 1,976배 그리고 JPEG의 경우 1,900배 빠르게 예측된다.

제안된 소비 전력 모델을 통해 각 테스트벤치별로 최소, 최대, 그리고 평균 소비 전력을 구한 후, 각각의 편차를 구하였다. Dhrystone V2.1 테스트 벤치의 경우 최소, 최대 그리고 평균 예측 소비 전력은 표 3에 나타난 것처럼 각각 108.6W, 122.5W, 그리고 116.0W로 나타났다. 게이트 레벨의 평균 소비 전력과 비교하여 최소, 최대, 그리고 평균 예측 소비 전력은 과 최대 예측 소비 전력은 각각 7.3%, 24.5%, 그리고 9.0%의 오차율을 보였다. 반면에 JPEG 테스트 벤치의 경우 최소, 최대 그리고 평균 예측 소비 전력은 게이트 레벨의 평균 소비 전력과 비교하여 각각 14.6%, 9.7%, 그리고 3.6%의 오차율을 보였다.

특히, PrimePower를 이용하여 게이트 레벨의 A8051 프로세서의 순간 최대 소비 전력과 순간 최소 소비 전력을 구한 결과와 제안된 소비 전력 모델을 이용하여 최대 순간 소비 전력과 순간 최소 소비 전력과 비교한 결과 5%내외의 오차율을 보였다. 따라서, 평균 소비 전력 예측뿐만 아니라 제안된 모델을 이용하여 순간 최대 소비 전력과 순간 최소 소비 전력을 구할 경우 프로세서의 피크 소비 전력을 5%내외의 오차율로 구할 수 있음을 확인하였다. Park가 제안한 명령어 레벨의 기능 블록 레벨의 소비 전력 모델과 ISA (instruction set architecture) 시뮬레이션을 지원하는 아키텍처 레벨의 동기식 프로세서를 위한 소비 전력 모델[8]은 각각 오차율이 21% 및 13%의 오차율을 갖는다. 이 결과와 비교하여 제안된 비동기식 소비 전력 모델은 명령어 레벨로 구현되어 동작 시간이 빠름에도 불구하고 높은 정확도를 가짐을 알 수 있다.



[그림 5] 제안된 소비 전력 모델 예측값과 게이트 레벨의 소비 전력 측정값의 정확도 비교

[Fig. 5] The accuracy comparison between the estimated power from the proposed power model and the gate-level power measurement result

[표 2] 소비 전력 예측 시간 비교

[Table 2] Power estimation time comparison

	PrimePower Gate-level CPU time	Proposed power model CPU time	Speedup (×times)
DhrystoneV2.1	5.6 Hrs	20.2 sec	998
JPEG	3.8 Hrs	10.2 sec	1,341

[표 3] 최소, 최대, 그리고 평균 소비 전력 예측의 정확도 비교

[Table 3] Accuracy comparison between minimum, maximum, and average power estimation results

	제안된 소비 전력 모델을 이용한 예측 소비 전력			게이트레벨 측정소비전력
	최소	최대	평균	
DhrystoneV2.1	98.6W	132.5W	116.0W	106.4W
JPEG	76.4W	98.2W	86.7W	89.5W

## 5. 결론

본 논문은 비동기식 임베디드 프로세서를 내장한 SoC에 S/W 설계시 프로세서가 소비하는 전력을 예측하기 위한 명령어 레벨의 소비 전력 모델을 제안하였다. 제안된 소비 전력 모델은 프로그램이 실행되는 동안 프로세서로 폐지되는 명령어별로 소비 전력을 예측하고 실행된 모든 명령어의 소비 전력을 합하여 프로세서의 소비 전력을 예측한다. 비동기식 프로세서는 동작이 필요한 내부 블록만을 구동시키는 동작적 특성을 가지고 있기 때문에, 본 논문은 비동기식 프로세서의 명령어 실행 사이클을 분석

하고, 내부 상태를 활성, 버블, 그리고 이른 종료에 따른 대기 상태 유지로 각각 구분하여 소비 전력 모델을 구현하였다. 제안된 소비 전력 모델은 실제 구현된 비동기식 프로세서인 A8051을 이용하여 성능을 검증하였다. 게이트 레벨로 합성한 후 소비 전력을 측정된 결과와 비교한 결과 제안된 소비 전력 모델은 SoC의 시스템 설계 레벨에서 프로그램을 프로세서에서 구동할 때 소비 전력을 예측할 수 있음을 확인하였다. 또한, 소비 전력 예측을 기존 게이트 레벨에 비해 약 1,900배 이상 빠르게 수행할 수 있고, 9.7%의 오차율을 가짐을 확인하였다. 본 논문에서 제안된 소비 전력 모델은 A8051뿐만 아니라 다른 비동기식 프로세서에 적용 가능할 뿐만 아니라 프로세서가 아닌 다른 비동기식 시스템에 이용할 수 있을 것으로 기대된다.

## References

- [1] C. Y. Huang, Y. F. Yin and C. J. Hsu, "SoC HW/SW verification and validation", Proc. of ASP-DAC 2011, pp. 297-300, 2011.
- [2] M. H. Wu, W. C. Lee, C. Y. Chuang and R. S. Tsay, "Automatic generation of software TLM in multiple abstraction layers for efficient HW/SW co-simulation", Proc. of DATE 2010, pp. 1177-1182, 2010.
- [3] S. B. Furber, D. A. Edwards and J. d. Garside, "AMULET3: a 100MIPS asynchronous embedded microprocessor", Proc. of ICCD 2000, pp. 329-334, September 2000.
- [4] A. Bink and R. York, "ARM966HS: The first licensable, clockless 32-bit processor core", IEEE Micro, vol. 27, no. 2, pp. 58-68, March 2007.
- [5] J. H. Lee, Y. Kim and K. R. Cho, "Design of a fast asynchronous embedded CISC microprocessor, A8051", IEICE Trans. on Electronics, vol. E87-C, no. 4, pp. 527-534, April 2004.
- [6] J. H. Lee, Y. H. Kim and K. R. Cho, "A low-power implementation of asynchronous 8051 employing adaptive pipeline structure", IEEE Trans. on Circuits and Systems II - Express Briefs, vol. 55, no. 7, pp. 673-677, July 2008.
- [7] V. Tiwari, S. Malik and A. Wolfe, "Power analysis of embedded system; A first step towards software power minimization," IEEE Trans. on VLSI, vol. 2, no. 4, pp. 437-445, 1994.
- [8] Y. H. Park, S. Pasricha, F. Kurdahi and N. Dutt, "Methodology for multi-granularity embedded processor

- power model generation for an ESL design flow”, Proc. CODES+ISSS 2008, pp. 255-260, 2008.
- [9] D. Brooks, V. Tiwari, and M. Martonosi, “Wattch: A framework for architectural-level power analysis and optimizations”, Proc. of ISCA, pp. 83-94, 2000.
- [10] W. Ye, N. Vijaykrishnan, M. Kandemir and M. J. Irwin, “The design and use of SimplePower: A cycle-accurate energy estimation tool”, Proc. of DAC 2000, pp. 340-345, 2000.
- [11] C. H. Geotys and R. J. Gebotysu, “An empirical comparison of algorithmic, instruction, and architectural power prediction model for high-performance embedded DSP processors”, Proc. ISLPED, pp. 121-123, Aug. 1998.

---

## 이 제 훈(Je-Hoon Lee)

[정회원]



- 1998년 8월 : 충북대학교 정보통신공학과 (공학사)
- 2001년 2월 : 충북대학교 정보통신공학과 통신회로 및 시스템공학 (공학석사)
- 2005년 2월 : 충북대학교 정보통신공학과 통신회로 및 시스템공학 (공학박사)
- 2005년 4월 ~ 2006년 4월 : Univ. of Southern California Viterbi School 박사후연구원
- 2006년 8월 ~ 2009년 8월 : 충북대학교 BK21 충북정보기술사업단 초빙조교수
- 2009년 8월 ~ 현재 : 강원대학교 삼척캠퍼스 전자정보통신공학부 조교수

<관심분야>

임베디드프로세서, 디지털 회로 설계, SoC 설계, 임베디드시스템 개발