

# 비동기식 프로세서 A8051의 명령어 레벨 소비 전력 모델

## Instruction-level Power Model for Asynchronous Processor, A8051

이제훈

강원대학교 삼척캠퍼스 공학대학 전자정보통신공학부

Je-Hoon Lee(jehoon.lee@kangwon.ac.kr)

### 요약

본 논문은 비동기식 프로세서, A8051의 명령어 레벨 소비 전력 모델을 제안한다. 제안된 소비 전력 모델은 명령어 레벨로 프로세서가 소비하는 전력을 예측하지만, 프로그램이 실행되는 동안 비동기식 파이프라인의 동작 특성을 반영한다. 따라서, 제안된 방법은 프로세서 소비 전력 모델의 복잡도와 시뮬레이션 시간의 증가 없이 비동기식 임베디드 프로세서 소비 전력 모델의 정확도를 효과적으로 향상시켰다. 제안된 소비 전력 모델은 A8051의 소비 전력 특성을 반영하여 구현되었고 게이트 레벨의 합성한 결과를 이용한 소비 전력 예측 결과와 비교하여 성능 평가를 수행하였다. 제안된 소비 전력 모델은 게이트 레벨의 소비 전력 예측 결과와 비교하여 94%의 정확도를 보였고, 1,600 배 이상 시뮬레이션 시간을 단축하였다.

■ 중심어 : | 비동기식 프로세서 | 소비 전력 모델 | 소비전력 예측 | 시스템온칩 |

### Abstract

This paper presents new instruction-level power model for an asynchronous processor, A8051. Even though the proposed model estimates power consumption as instruction level, this model reflects the behavioral features of asynchronous pipeline during the program is executed. Thus, it can effectively enhance the accuracy of power model for an asynchronous embedded processor without significant complexity of power model as well as the increase of simulation time. The proposed power model is based on the implementation of A8051 to reflect the characteristics of power consumption in A8051. The simulation results of the proposed model is compared with that of gate-level synthesized A8051. The proposed power model shows the accuracy of 94% and the simulation time for estimation the power consumption was reduced to 1,600 times.

■ keyword : | Asynchronous Processor | Power Model | Power Estimation | System-on-chip |

## I. 서론

최근 VLSI 기술 발전으로 하나의 칩에 전체 시스템을 집적하는 SoC (system-on-chip) 설계가 일반화되고 있다. SoC 설계 복잡도는 SoC 설계 능력 향상 속도에 비해 훨씬 빠르게 증가하며 이들간의 설계 생산성

갭 (design productivity gap)도 점점 커지고 있다. 설계 속도를 높이기 위해, 기존 설계 회로를 재활용하는 IP (intellectual property) 기반 SoC 설계 방법과 SoC내에 프로세서와 주변장치를 범용 플랫폼으로 내장하여 하드웨어와 소프트웨어를 동시에 개발하는 플랫폼 기반 SoC 설계 방법이 사용된다[1-3].

\* "본 연구는 교육과학기술부와 한국연구재단의 지역혁신인력양성사업으로 수행된 연구결과임 (2012H1B8A2026055)"

접수번호 : #120419-012

접수일자 : 2012년 04월 19일

심사완료일 : 2012년 06월 08일

교신저자 : 이제훈 : jehoon.lee@kangwon.ac.kr

SoC 설계는 스펙 결정, 시스템 레벨 설계, HW/SW 동시 설계, RTL 혹은 게이트 레벨 설계, 통합 설계 및 검증 순으로 진행된다. SoC의 소비 전력을 줄이기 위해서는 다양한 소비 전력 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 재설계를 해야 하며, 이 과정은 SoC 설계가 끝날 때까지 반복적으로 수행된다. 특히, 시스템 레벨 설계 단계와 같이 SoC 설계 초반에 정확한 소비 전력 예측을 통해 시스템 레벨에서 요구 소비 전력을 만족하는 SoC 설계가 가능할 경우, SoC 설계 과정 동안 반복되는 재설계 과정을 크게 줄일 수 있다.

플랫폼 기반 SoC에서 소프트웨어 설계는 SoC에 집적된 임베디드 프로세서를 통해 구현된다. 고성능 SoC가 요구됨에 따라, 고성능 프로세서를 집적하거나, 하나의 SoC내에 여러 개의 프로세서들을 집적한다. 소프트웨어 설계의 중요성이 증가되면서 SoC내에서 임베디드 프로세서의 소비 전력이 차지하는 비중도 증가하며, 저전력 SoC 구현을 위해서는 프로세서의 소비 전력 감소가 중요하다. 대부분 프로세서는 클럭 신호에 따라 모든 모듈들이 동시에 구동하는 동기식 회로 설계 방법으로 구현된다. 동기식 프로세서는 동작이 불필요한 모듈을 구동하고 클럭 신호를 전역 전송하여 소비 전력을 낭비한다[4][5]. 일례로, StrongARM 및 XScale 프로세서는 PLL을 포함한 클럭 관련 회로가 전체 소비 전력 중에서 각각 26%와 23%의 전력을 소비하며, 최근 프로세서중 IBM사의 zEnterprise™ 프로세서도 클럭 관련 회로가 19%의 전력을 소비한다[6-8].

비동기식 회로는 클럭을 사용하지 않고, 내부 모듈끼리 핸드셰이킹 프로토콜에 의해 데이터를 동기화하고 전송하는 회로이다. 비동기식 회로로 구현된 프로세서인 비동기식 프로세서는 핸드셰이킹 프로토콜에 의해 동작이 요구되는 모듈만을 선택적으로 구동시키고, 나머지 모듈들은 자동적으로 대기상태로 유지시켜 전력 낭비를 줄인다. 또한, 비동기식 프로세서는 칩 내부를 전역 전송하는 클럭 신호 대신 이웃한 회로간의 로컬 전송하는 핸드셰이킹 신호를 사용한다. 비동기식 프로세서의 핸드셰이킹 회로의 소비 전력은 전체 프로세서

소비 전력의 10.5%에 불과하다[9][10]. 동일한 DSP 프로세서를 동기식 및 비동기식으로 설계하였을 때 비동기식 DSP가 동기식에 비해 30% 소비 전력을 감소시킨다는 연구 결과가 발표되었다[11]. 비동기식 프로세서 중 상용 SoC에 적용된 프로세서는 AMULET3i 프로세서와 Handshake Solution社의 ARM966HS가 대표적이다[12]. [표 1]에서 나타난 것처럼 325MHz 클럭을 사용하는 동기식 ARM 프로세서가 350MIPS의 성능과 163mW의 평균 소비 전력을 갖는데 반해, 비동기식 ARM966HS 프로세서는 83MIP의 동작속도와 3.6mW의 평균 소비 전력을 가져 소비 전력면에서 크게 유리하다. 이러한 이유로, 저전력 SoC에는 비동기식 프로세서가 적합하다.

소비 전력면에서 유리하다는 비동기식 회로의 장점에도 불구하고, 상용 칩 개발이 부족하다. 비동기식 회로에서 핸드셰이킹 프로토콜을 처리하기 위해서는 추가적인 배선과 제어 로직이 요구된다. 따라서, 배선이 복잡해지고, 회로 크기가 증가한다. 또한, 상용화된 CAD 툴의 부재와 회로 설계의 동작 검증을 적용할 수 있는 표준화된 방법이 부족하다는 단점을 갖는다. 현재 SoC에서 사용되는 프로세서의 소비 전력 모델은 동기식 프로세서를 모델로 구성되었기 때문에 이를 비동기식 프로세서에 직접적으로 적용하기 어렵다는 문제를 갖는다.

동기식 프로세서의 소비 전력 모델은 SoC 명령어 레벨, 아키텍처 레벨, 마이크로 아키텍처 레벨등과 같이 서로 다른 추상화 레벨로 구성된다. 특히, 명령어 레벨과 같이 높은 수준의 추상화 레벨의 소비 전력 모델은 소비 전력 모델이 간단하고 쉽게 구현되기 때문에, SoC 설계 초반에 이용할 수 있다. 그러나, 예측된 소비 전력의 정확도가 떨어진다는 단점을 갖는다. 프로세서의 파이프라인 구조, 메모리, 캐쉬 그리고 분기 예측기와 같은 기능 블록별로 세분화된 아키텍처 레벨의 소비 전력 모델은 파이프라인 레벨의 정확도를 갖는다. 따라서, 정확도는 크게 상승하지만 소비 전력 모델 구현을 위한 노력과 예측 시간이 크게 증가하기 때문에 시스템 레벨과 같은 SoC 상위 수준 설계시 사용이 어렵다.

이러한 문제를 해결하기 위하여 본 논문은 비동기식

프로세서, A8051의 소비 전력 모델을 제안하였고 게이트 레벨의 소비 전력과 비교하여 성능을 평가하였다. 제안된 소비 전력 모델은 명령어 레벨의 소비 전력 모델로 시스템 레벨과 같은 SoC 설계 초반에 사용될 수 있다. 또한 비동기식 파이프라인 동작을 고려하여 아키텍처 레벨의 파이프라인 정확도를 갖도록 구성하였다. 제안된 소비 전력 모델은 비동기식 프로세서인 A8051에 적용하여 이를 위한 소비 전력 모델을 구현하고, 이를 이용하여 SoC 설계 초기에 프로세서의 소비 전력을 예측할 수 있어야 한다.

본 논문의 구성은 다음과 같다. II장은 비동기식 시스템의 동작 특성과 이에 따른 소비 전력 특성에 대해 설명하고, III장에서는 제안하는 비동기식 프로세서의 명령어 레벨 소비 전력 모델에 대해 설명한다. IV장에서는 모의실험을 통해 제안하는 소비 전력 모델의 성능을 분석하며, 마지막으로 V장에서 결론을 맺는다.

표 1. 비동기식 및 동기식 임베디드 프로세서 비교

	AMULET3H	ARM966HS	동기식 ARM
연구기관	맨체스터대	Handshake Solution社	ARM社
평균성능	100 MIPS	83 MIPS	350 MIPS
공정	0.35- $\mu$ m	0.13- $\mu$ m	0.13- $\mu$ m
소비전력	0.215mW	0.045mW/MHz	0.5mW/MHz
구조	32-bit RISC구조		

## II. 비동기식 시스템의 동작 및 소비 전력 특성

동기식 회로의 클럭 주기는 모든 블록중 가장 긴 지연시간으로 결정되고 클럭의 상승 천이시 모든 블록이 동시에 구동되고 다음 클럭 천이전에 종료된다. 비동기식 회로의 모든 블록들은 핸드셰이킹 프로토콜에 따라서 서로 다른 동작 시간을 갖는다. 연산 완료 시점을 검출하는 방법은 비동기식 회로가 채택한 지연 모델 (delay model)에 따라 다르다. 지연 모델은 배선 및 소자 지연에 대한 타이밍 가정이며 [표 2]처럼 유한 (bounded)과 미지 지연 (delay insensitive) 모델로 나뉜다.

유한 지연 모델은 소자와 배선 지연을 알고 있다고 가정하는 지연 모델이며, 이를 위해 전체 회로를 게이트

레벨로 합성한 후, 포스트 레이아웃 (post-layout) 시뮬레이션으로 모든 회로의 최대 지연 시간을 추출하고 이보다 큰 지연을 갖는 지연 소자 (delay element)를 생성하고 이를 통해 동작 완료를 검출한다. 반면, 미지 지연 모델은 소자와 배선 지연은 유한하나 상한치를 알 수 없다고 가정하며, 데이터 전송시 타이밍 정보를 실어 전송하고 각 회로의 연산이 끝난 후 동작 완료 검출 회로로 동작 완료 시점을 검출한다.

비동기식 시스템은 동기식 시스템과 다른 소비 전력 특성을 갖는다. 첫째, 비동기식 시스템은 동작이 불필요한 데이터패스는 휴지 (idle) 상태를 유지시키기 때문에 평균 소비 전력이 감소한다. 특히, 계층 구조 캐쉬 메모리, 복잡한 파이프라인 구조, 중첩을 이용한 병렬 처리처럼 프로세서가 복잡할수록 동작이 불필요한 블록의 수가 증가한다. 따라서, 비동기식 프로세서는 데이터패스를 더욱 세분화하여 명령어에 따라 실행이 요구되는 데이터패스의 길이를 줄여 소비 전력을 줄일 수 있다.

표 2. 비동기 회로의 지연 모델

지연 모델	게이트 지연	배선 지연
Bounded	bounded	bounded
Speed independent(SI)	unbounded	zero
Delay insensitive(DI)	unbounded	unbounded
Quasi DI (QDI)	unbounded	isochronic fork
Scalable DI(SDI)	unbounded	all except bounded

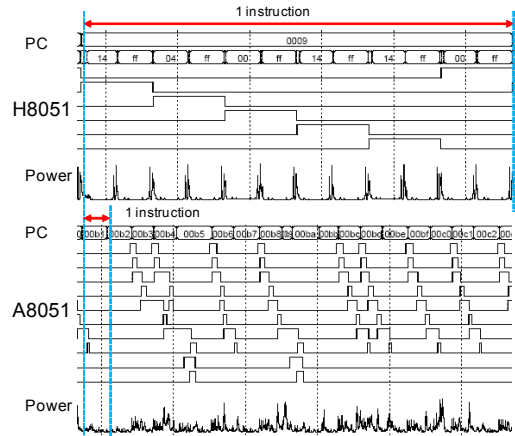


그림 1. 동기식 프로세서 (H8051)과 비동기식 프로세서 (A8051)의 소비 전력 특성 비교

[그림 1]은 동기식 8051 프로세서인 H8051과 비동기식 버전인 A8051 프로세서에 동일한 명령어를 실행시킨 결과를 나타낸다. [그림 1]처럼 동기식 8051은 12 클럭동안 하나의 명령어를 실행하고, 클럭 상승 천이시 모든 블록이 구동되어 임펄스적인 소비 전력을 갖는 반면 비동기식 8051은 파이프라인 구조를 갖기 때문에 실행시간이 크게 감소된다. 또한 각 모듈이 동시에 구동되지 않기 때문에 소비 전력이 전체 시간대역으로 분산되어 순간 소비 전력이 작다. 비동기식 프로세서의 소비 전력 특성은 동기식과 다르기 때문에, 동기식 프로세서의 소비 전력 모델을 이용하여 비동기식 프로세서의 소비 전력을 예측할 경우 정확도가 크게 떨어진다.

### III. 제안된 비동기식 프로세서의 명령어 레벨 소비 전력 모델

본 논문에서는 비동기식 프로세서의 소비 전력 예측 모델을 구현하기 위해 인텔 8051 프로세서의 비동기식 버전인 A8051 프로세서를 이용하였다. A8051 프로세서는 메모리를 제외한 전체 회로는 미지 지연 모델을 사용하였다. A8051의 명령어 집합은 인텔 8051 프로세서와 호환되는 255개의 명령어로 구성되었고, 각 명령어는 인텔 8051 프로세서와 마찬가지로 1, 2, 그리고 3 바이트의 서로 다른 명령어 길이를 갖는다[13-15]. 그러나 A8051 프로세서의 파이프라인 구조와 명령어 실행 스킴은 동기식 8051 프로세서와 크게 다르다.

인텔 8051은 6개의 내부 상태로 구성된 머신 사이클을 갖는다. 동작이 불필요한 내부 상태도 명령어 실행의 규칙성을 유지하기 위해 실행시키고, 이를 버블(bubble)이라 한다. 모든 명령어의 명령어 실행 사이클은 머신 사이클의 배수로 구성되며, 다중 사이클 명령어로 인해 선형 파이프라인 구조를 채택하기 어렵다. A8051 프로세서는 일반적인 RISC 프로세서처럼 명령어 페치(IF), 디코드(ID), 오퍼랜드 페치(OF), 실행(EX), 그리고 저장(WB)으로 구성된 파이프라인 구조를 갖는다. A8051의 모든 명령어는 하나의 명령어 실행 사이클로 구동되나, [표 3]에 나타낸 것처럼 7개의 서로 다른 길이를 갖는다. 그룹 1부터 그룹 4의 명령어들은 전체

명령어의 85%를 차지하고, 선형 파이프라인으로 실행된다. 나머지 명령어들은 OF와 EX의 부분적인 반복이 필요하다. A8051은 버블 상태를 실행시키지 않고 건너뛰는 스테이지-스킵핑(stage-skipping)과 다중 사이클 명령어를 위한 국부적인 스테이트 반복을 위한 멀티-루핑(multi-looping)을 허용한다. A8051은 파이프라인 채택으로 성능이 향상되고, 버블 상태의 기능 블록을 대기 상태에 유지시켜 소비 낭비를 막는다.

[그림 2]는 “Add A, #data” 명령어의 동기식 8051과 A8051 프로세서의 명령어 실행 사이클을 나타낸다. 동기식 8051에서 “Add A, #data”는 6 개의 내부 상태로 구성된 1 머신 사이클에 동작한다. 동기식 8051은 한 번에 1 바이트 메모리 액세스를 허용하며, 해당 명령어는 명령어 길이가 2 바이트로, 두 번의 메모리 액세스가 필요하다. 또한 명령어 디코드, 실행 단계가 필요하며 S3와 S6의 두 상태는 동작이 불필요한 버블이나 규칙성을 유지하기 위해 포함된다. 반면 A8051의 명령어 실행 사이클은 해당 버블 상태의 동작을 건너뛰며, A8051의 “Add A, #data” 명령어 실행 사이클은 IF, ID, OF 및 EX의 4개의 스테이지로 구성된다.

표 3. A8051 명령어 실행 스킴

	명령어의 수	부분 반복의 수	명령어 실행 스킴
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

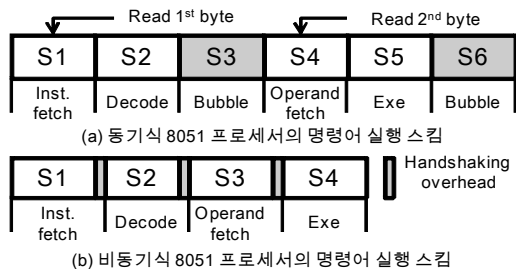


그림 2. 2바이트, 1사이클 명령어 ADD A, #data의 동기식과 비동기식 8051의 명령어 실행 사이클 비교

프로세서의 소비 전력 모델은 추상화 레벨, 정확도, 시뮬레이션 시간, 그리고 개발 비용간의 트레이드-오프로 결정된다. 가장 먼저 제안된 V. Tiwari의 소비 전력 모델은 프로세서가 프로그램을 구동하는 동안 실행되는 명령어를 통해 전력을 예측하는 소비 전력 모델이다 [16]. Tiwari의 소비 전력 모델은 프로세서가 지원하는 모든 명령어별로 소비 전력을 예측한 후, 타겟 프로그램이 실행되는 동안 실행된 명령어들의 소비 전력의 합으로 소비 전력을 구한다. 이러한 명령어 레벨 소비 전력 모델은 간단한 구성과 빠른 예측이 가능하나 파이프라인, 캐쉬 메모리, 분기 예측기를 갖는 복잡한 프로세서의 소비 전력 예측시 정확도가 크게 떨어진다. 소비 전력 예측의 정확도를 높이기 위해 프로세서의 아키텍처 혹은 마이크로 아키텍처 레벨의 소비 전력 모델처럼 복잡한 파라미터들로 구성된 소비 전력 모델들이 제안되었으나, 시스템 레벨 설계와 같은 SoC 초기 설계 단계에 적용하기에는 시뮬레이션 속도가 느리다[17-20].

V. Tiwari의 명령어 레벨 모델은 식 (1)처럼 나타난다. 벤치마크 프로그램의 전체 명령어 갯수를  $N_i$ 로 그리고 각 인접한 명령어의 수는  $N_{i,j}$ 로 가정한다. 이 때 전체 프로그램을 실행하는 동안 프로세서가 소비하는 전체 에너지,  $E_p$ 는 프로그램내에 포함된 각 명령어들이 실행하는 기본 에너지 값,  $B_i$ 와 명령어의 개수,  $N_i$ 를 곱한 결과와 연속된 명령어 각각의 에너지 오버헤드,  $O_{i,j}$ 와 인접한 명령어의 수,  $N_{i,j}$ 의 곱, 그리고 파이프라인 스톱과 캐쉬 미스와 같이 내부 명령어 실행에 따른 에너지 소비,  $E_k$ 의 합으로 결정된다.

$$E_p = \sum_i (B_i \times N_i) + \sum_{i,j} (O_{i,j} \times N_{i,j}) + \sum_k E_k \quad (1)$$

동기식 프로세서의 경우 프로세서가 명령어를 실행할 때, 모든 명령어들은 동일한 명령어 실행 사이클에 따라 구동되어, 식 (1)과 같이 소비 전력을 예측할 수 있다. 그러나, 비동기식 프로세서는 명령어 실행 스킴이 가변적이고, 동작이 불필요한 기능 블록을 구동시키지 않기 때문에 Tiwari의 명령어 레벨 소비 전력 모델을 직접 비동기식 프로세서에 적용할 경우 정확도가 크게 떨어진다. 이러한 문제를 해결하기 위해, 본 논문에서는

명령어 실행 스킴과 파이프라인 동작 특성을 반영하는 새로운 명령어 레벨의 소비 전력 모델을 제안한다.

제안된 비동기식 프로세서의 소비 전력 모델은 명령어 기반 소비 전력 모델임에도 불구하고, 명령어 실행 스킴과 파이프라인 동작 특성을 분석하여 소비 전력 모델의 정확도를 크게 향상시켰다. 또한, 제안된 비동기식 프로세서 소비 전력 모델은 명령어 레벨로 소비 전력 예측이 가능하기 때문에, SoC의 시스템 레벨 설계와 같은 SoC 초기 설계 단계에 쉽게 이용할 수 있다.

본 논문에서 제안한 비동기식 프로세서의 명령어 레벨 소비 전력 모델은 [그림 3]과 같이 크게 세 단계로 구성된다. 첫 단계는 A8051의 명령어 실행 스킴에서 동작이 필요한 파이프라인 스테이지와 버블 스테이지를 구분한다. 두 번째 단계는 A8051 5단 파이프라인 구조에서 동시에 실행될 수 있는 모든 경우를 A8051의 ISS에 입력하여 실제 파이프라인에서 구동되는 환경을 에뮬레이션한다. 세 번째 단계에서는 A8051 프로세서를 타겟 라이브러리를 이용하여 합성한 후, 모든 가능한 명령어 조합의 소비 전력을 프로파일로 구성한다.

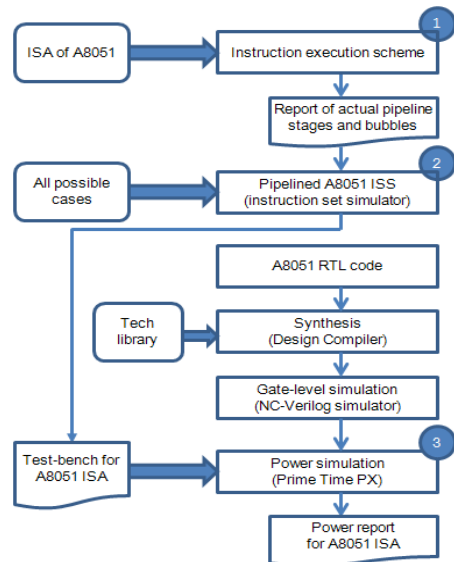


그림 3. 제안된 비동기식 프로세서를 위한 명령어 레벨 소비 전력 모델 구성 순서도

제안된 비동기식 프로세서의 명령어 레벨 소비 전력

모델을 구성하기 위해 단계별 세부 수행 내용은 아래와 같다. 첫 번째 단계는 A8051 프로세서의 명령어 실행 스킴에 따라 명령어별로 동작이 요구되는 스테이지와 버블 스테이지를 구분한다. 비동기식 프로세서의 경우 동작이 요구되는 파이프라인 스테이지는 실행에 따라 전력을 소비하지만, 버블 스테이지의 경우 누설 전류에 따른 대기 전력만을 소비한다. 정확한 소비 전력 예측을 위해서는 파이프라인 구동중에 파이프라인 버블을 유발하는 명령어들을 구분하여야 한다. 따라서, A8051 프로세서의 전체 명령어에 대해 각 명령어별로 명령어 실행 스킴과 프로세서의 파이프라인 구조를 대응하여 시켜 식 (2)와 같이 이들을 구분한다.

$$I_k = \sum_i^n AS_i + \sum_j^m BS_j \quad (2)$$

K-번째 명령어,  $I_K$ 는 5개의 파이프라인 스테이지중에서 각 명령어 실행 스킴에 따라 동작이 요구되는  $n$ 개의 활성 파이프라인 스테이지 (AS, active stage)들의 집합,  $\sum AS_i$ 와  $m$ 개의 동작이 불필요한 버블 스테이지 (BS, bubble stage)들의 집합,  $\sum BS_j$ 로 구성된다. 전체 명령어의 활성 파이프라인 스테이지와 버블 스테이지의 종류를 결정하고 LUT (look-up table)에 저장한다.

두 번째 단계는 프로그램이 실행될 때, A8051의 ISS (instruction set simulator)를 이용하여 파이프라인 동작을 에뮬레이션한다. A8051은 5단 파이프라인 구조를 갖기 때문에, 동시에 최대 5개의 명령어가 실행되며, A8051의 파이프라인을 통해 발생 가능한 명령어 조합의 수는  $255^5$ 이 된다. A8051 프로세서는 동작이 불필요한 파이프라인 스테이지를 건너뛰는 스테이지-스킵핑, 연산 종료 후 다음 파이프라인 스테이지가 사용 가능할 때, 파이프라인 래치를 투명하게 만드는 파이프라인 병합, 그리고 지역적인 파이프라인 스테이지들의 멀티-루핑을 허용한다. 이전에 파이프라인에 진입한 명령어가 멀티-루핑을 할 경우 다음 명령어는 해당 파이프라인 스테이지가 사용가능할 때까지 기다리는 파이프라인 스톱이 발생된다. 2단계에서는 발생 가능한 명령어 조합을 ISS에 입력하여 각 파이프라인 스테이지 동작을

에뮬레이션하고 입력된 모든 명령어 조합별로 각 파이프라인 스테이지 동작 결과를 LUT에 저장한다.

세 번째 단계에서는 RT-레벨의 A8051 프로세서를 타겟 라이브러리를 이용하여 합성하고 게이트 레벨의 시뮬레이션을 수행한다. 2단계에서 얻어진 파이프라인에서 동시에 구동될 수 있는 모든 명령어 조합을 테스트벤치로 구성한 후 Synopsis사의 PrimePower툴을 이용하여 각 명령어 조합별로 프로세서의 모든 기능 유닛들이 소비하는 전력을 구분하여 LUT에 저장한다.

제안된 비동기식 프로세서의 소비 전력 모델은 각 단계별로 얻어진 결과를 이용하여 프로그램내의 실행되는 명령어를 추적하여 얻는다. 제안된 소비 전력 모델은  $N$ 개의 명령어로 구성된 프로그램의 각 명령어들의 파이프라인 스테이지의 소비 전력,  $E_{inst}$ , 파이프라인 래치 및 제어 회로의 소비 전력,  $E_{ctrl}$ , 그리고 명령어 캐쉬와 분기 예측기와 같은 추가적인 회로에서의 소비 전력,  $E_{others}$ 를 식 (3)과 같이 모두 더해 전체 프로그램의 소비 전력,  $E_p$ 를 구할 수 있다.

$$E_p = (E_{inst} + E_{ctrl} + E_{others}) \times N_i \quad (3)$$

제안된 소비 전력 모델을 이용하여 소비 전력을 예측할 때, ISS를 이용하여 프로그램이 실행될 때, 파이프라인 스테이지 및 메모리의 동작 여부를 추적한다. 각 명령어가 소비하는 에너지,  $E_{inst}$ 는 프로그램 실행시 실제로 구동되는 명령어들을 순서대로 트레이스하고, 각 명령어 실행 사이클에 포함된 파이프라인 스테이지들의 소비 전력과 대기 상태에 머무는 버블 스테이지의 소비 전력을 더하여 구한다. 따라서, 식 (2)와 같이 각 명령어 실행 사이클의 활성 스테이지 (active stage, AS)와 버블 스테이지 (bubble stage, BS)에서의 대기 전력을 구분하여 식 (4)와 같이 구한다.

$$E_{inst} = \sum_{i=1}^N [\sum_{j=1}^5 (N_A \times EAS_j + EBS_j)] \quad (4)$$

식 (4)에서  $E_{inst}$ 는 명령어별로 파이프라인 스테이지에서 소비하는 에너지를 나타낸다.  $N$ 은 프로그램의 실행 명령어의 수를 나타낸다.  $j$ -번째 파이프라인 스테이지가 실행될 때의 평균 소비 전력은  $EAS_j$  그리고 버블

스테이지로 동작될 때의 대기 전력은  $EBS$ 로 나타낸다.  $N_A$ 는 파이프라인 스테이지중에 각 명령어의 명령어 실행 사이클에 포함된 횟수를 나타낸다. A8051은 명령어에 따라 로컬 파이프라인 스테이지 반복을 허용한다. 따라서 하나의 명령어 실행 사이클에 포함된 파이프라인 스테이지의 실행 횟수가 다르며, 제안된 소비 전력 모델에서  $N_A$ 로 나타낸다. 명령어별로 명령어 실행 스킴에 포함된 활성 파이프라인 스테이지의 소비 전력과 버블 스테이지에서의 대기 전력을 더하여 명령어별로 소비하는 전력을 구한다. 전체 프로그램에서 실행된 모든 명령어에 대해 이 과정을 반복하여 전체 프로그램에서 파이프라인 스테이지들이 소비한 전력을 구한다.

두 번째로 ISS 에뮬레이션 단계에서 A8051 파이프라인을 통해 동시에 구동할 수 있는 모든 가능한 경우를 테스트-벤치로 구동하여 파이프라인 래치를 포함한 제어 회로에서의 소비 전력,  $E_a$ 를 식 (5)와 같이 구할 수 있다. IF, ID, OF, EX, 그리고 WB으로 구성된 A8051의 5단 파이프라인 구조에서 동시에 최대 5개의 명령어들을 동시에 실행시킬 때, 각각의 파이프라인 래치 및 제어 회로에서 발생하는 에너지 오버헤드를 각각  $E_{IFID}$ ,  $E_{IDOF}$ ,  $E_{OFEX}$ ,  $E_{EXWB}$ 으로 나타내고, 이들의 합은 파이프라인 구동에 따른 에너지 오버헤드가 된다.

$$E_a = E_{IFID} + E_{IDOF} + E_{OFEX} + E_{EXWB} \quad (5)$$

마지막으로, A8051의 분기 예측기와 명령어 캐쉬에 의한 추가적인 에너지 소비량,  $E_{others}$ 를 식 (5)처럼 구한다. 식 (5)에서,  $E_{b,t}$ 와  $E_{b,nt}$ 는 분기 예측이 맞은 경우 분기 예측기의 소비 전력과 분기 예측이 틀린 경우 되돌리기 위해 필요한 소비 전력을 각각 나타낸다.  $E_{c,hit}$ 와  $E_{c,miss}$ 는 명령어 캐쉬가 히트될 때와 미스되었을 때 명령어 캐쉬의 소비 에너지를 나타낸다. 마지막으로 제안된 비동기식 프로세서의 소비 전력 모델은 식 (3), 식 (4) 및 식 (5)의 결과를 합하여 식 (6)과 같이 구현된다.

$$E_{others} = E_{b,t} + E_{b,nt} + E_{c,hit} + E_{c,miss} \quad (6)$$

#### IV. 실험 및 결과분석

본 논문에서 제안한 명령어 레벨의 소비 전력 모델을 검증하기 위해 다음과 같은 실험 환경을 구축하였다. 첫 번째로, A8051 ISS는 프로그램이 실행되는 동안 실행되는 명령어를 추적하고, 파이프라인 스테이지 각각의 동작을 에뮬레이션한다. ISS를 이용하여 입력된 A8051 명령어에 따라 파이프라인 동작을 추적하며, 버블 스테이지의 대기 유지, 지역적인 멀티투핑으로 인한 파이프라인 스톨, 그리고 파이프라인 스테이지 병합을 구분한다. 두 번째로, A8051 프로세서는 VHDL을 이용하여 RTL 레벨의 동작을 검증한 후 Hynix 0.35- $\mu$ m 표준 셀 라이브러리를 이용하여 합성하였다. 합성된 A8051은 Synopsys사의 PrimePower를 이용하여 입력 패턴에 따른 평균, 최대 그리고 최소 소비 전력을 구할 수 있다.

표 4. ISS을 이용한 A8051의 명령어 실행 스킴 특징

종류	버블스테이트 제거율	파이프라인 병합 발생율	파이프라인 스톨 발생율
A8051 ISA	16.2%	13.7%	14.7%

첫 번째 실험은 A8051의 모든 명령어의 가능한 조합을 구성한 후 ISS에 입력하고 파이프라인 동작을 에뮬레이션하였다. [표 4]에 나타난 것처럼, 전체 동작이 필요한 파이프라인 스테이지 중에서 16.2%의 파이프라인 스테이지를 구동하지 않고 건너뛴다. 또한 명령어 조합에 따라 파이프라인 스테이지중 다음 파이프라인 사용이 가능한 경우 그 사이의 파이프라인 래치를 투명하게 만들어 두 파이프라인 스테이지를 병합한다. [표 3]의 그룹 1에서 그룹 3의 명령어들은 버블 스테이지 제거로 빠른 종료로 파이프라인 병합이 발생된다. 에뮬레이션을 통해 병렬 처리 가능한 모든 명령어 조합을 트레이스한 결과 13.7%의 파이프라인 병합율을 갖는다. 전체 명령어중 15%에 해당하는 명령어들이 파이프라인 실행시 OF 및 EX의 부분적인 반복을 허용하며, 비선형 파이프라인 구동으로 인해 다음에 폐지된 명령어의 파이프라인 진행이 스톨된다. 에뮬레이션 결과 A8051 프로세서의 14.7%의 파이프라인 스톨이 발생하였다.

ISS를 통한 명령어 실행 분석을 수행한 후, RTL 레벨의 A8051 프로세서 코어를 Hynix 0.35- $\mu\text{m}$  공정을 이용하여 합성한 후, Synopsis사의 PrimePower를 이용하여 소비 전력 시뮬레이션을 수행하였다. 모든 가능한 명령어 조합을 포함한 테스트벤치를 입력하여 A8051내의 각각의 파이프라인 스테이지 및 메모리, 파이프라인 컨트롤로직의 소비 전력을 각각 구한 후 파이프라인 내의 병렬 처리되는 명령어 조합별로 소비 전력 프로파일을 저장한다. 그 후 SoC의 S/W 컴포넌트를 A8051 프로세서로 구동할 때 프로세서가 소비하는 전력을 예측한다. [표 5]는 사이클 수에 따른 프로세서의 컴포넌트별 소비 전력을 나타내며, 110 사이클 동안 전체 명령어의 평균 소비 전력은 24.7mW를 소비한다. 각 컴포넌트별로 소비 전력을 나누면 IF, ID, OF, EX 및 WB 스테이지는 각각 전체 소비 전력의 32%, 20%, 11%, 13% 그리고 9%를 나타낸다. IF 스테이지는 분기 명령어를 위한 분기 예측을 수행하기 때문에 소비 전력의 양이 가장 크다. A8051 프로세서는 프로그램 메모리와 데이터 메모리가 분리되었고, 데이터 메모리로 액세스가 필요할 때만 전력을 소비하기 때문에 8%의 적은 소비량을 갖는다. 마지막으로, 핸드셰이킹 및 파이프라인 컨트롤 회로가 전체 소비 전력의 7%를 차지한다. 클럭 및 관련 회로가 전체 소비 전력의 20% 이상을 차지하는 동기식 프로세서보다 크게 적다.

표 5. 소비 전력 시뮬레이션 결과

Cycle	Components							Total (mW)
	IF	ID	OF	EX	WB	Mem.	Ctl.	
10	8.4	2.6	1.1	2.1	1.1	1.2	0.5	18.0
20	6.5	3.6	1.7	2.2	1.5	3.1	2.5	21.1
30	8.2	4.4	1.4	3.8	3.2	1.1	1.2	23.3
40	7.9	5.8	3.1	2.9	1.1	2.3	1.8	24.9
50	7.8	4.7	2.1	3.1	3.5	2.1	1.9	25.2
60	7.1	5.8	4.1	4.5	1.6	0.9	2.3	26.3
70	7.8	5.5	2.5	4.0	1.4	3.1	1.4	25.7
80	7.7	5.9	4.1	2.8	2.1	1.8	2.3	26.7
90	7.9	7.1	2.9	3.2	1.9	2.2	1.8	27.0
100	8.0	5.1	4.0	4.5	2.5	1.8	2.1	28.0
110	10.2	4.5	2.1	2.8	3.2	1.7	2.1	26.6
Ave.	8.0	5.0	2.6	3.3	2.1	1.9	1.8	24.7

본 논문에서는 사이클-레벨 정확도 (cycle-level accuracy)를 갖는 프로세서 소비 전력 모델을 구성하였고, 상용 테스트벤치인 Dhrystone V2.1 프로그램을 이용하여 소비 전력을 예측하였다. 성능 평가를 위해 A8051 프로세서 코어를 Hynix 0.35- $\mu\text{m}$  표준 셀 라이브러리 공정을 이용하여 게이트 레벨로 합성한 후, Dhrystone V2.1 테스트벤치 프로그램을 입력하여 소비 전력을 측정된 결과와 본 논문에서 제안한 소비 전력 모델을 이용하여 소비 전력을 예측한 결과를 비교하였고, 그 결과는 [그림 4]와 같이 나타난다. 제안된 프로세서의 소비 전력 예측 결과 게이트 레벨의 소비 전력 예측 결과와 비교하여 94%의 정확도를 가짐을 확인하였다. 일반적으로 명령어 레벨의 동기식 소비 전력 모델의 정확도가 90% 내외이며, 보다 정밀한 아키텍처 레벨의 소비 전력 모델이 일반적으로 95% 내외의 정확도를 가짐과 비교하여 제안된 소비 전력 모델은 명령어 레벨임에도 불구하고 94%의 높은 정확도를 갖는다.

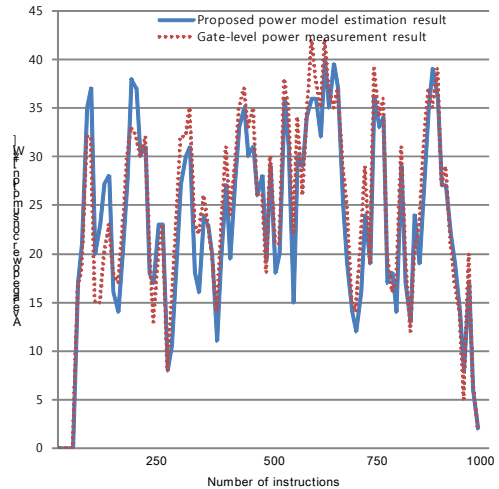


그림 4. 제안된 A8051의 소비 전력 모델과 A8051 게이트 레벨의 소비 전력 예측을 통한 정확도 비교

표 6. 소비 전력 예측 시간 비교

	PrimePower Gate-level CPU time	Proposed power model CPU time	Speedup (x times)
Dhrystone V2.1	5.6 Hrs	12.5 sec	1,613
JPEG	3.8 Hrs	7.9 sec	1,732



본 논문에서 제안한 소비 전력 모델을 SoC 시스템 레벨 설계에 적용하여 소비 전력을 예측하는데 걸리는 시간과 Synopsys사의 PrimePower를 이용하여 게이트 레벨에서의 소비 전력을 예측하는데 걸리는 시간을 비교하여 [표 6]에 나타내었다. 테스트벤치로 Dhrystone V2.1과 JPEG을 이용하였고, 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 1,613배 그리고 JPEG의 경우 1,732배 빠르게 예측된다.

## V. 결론

본 논문은 비동기식 프로세서, A8051을 SoC에 내장할 때, 프로세서가 소비하는 전력을 예측하기 위한 소비 전력 모델을 제안하였다. 제안된 소비 전력 모델은 프로그램이 실행되는 동안 프로세서로 폐치되는 명령어별로 소비 전력을 예측하고 실행된 모든 명령어의 소비 전력을 합하여 프로세서의 소비 전력을 예측하는 명령어-레벨의 소비 전력 모델이며, 동작이 필요한 모듈만 구동시키는 비동기식 회로의 동작 특성을 소비 전력 예측에 반영하였다. 타겟 프로세서인 비동기식 프로세서, A8051에서 사용된 비동기식 적응형 파이프라인에서 채용한 스테이지 스킵핑, 스테이지 병합 그리고 멀티-루핑으로 인한 파이프라인 스톨 등의 동작 특성에 따른 소비 전력 변화를 제안된 소비 전력 모델에 반영하였다. 성능 평가 결과 제안된 소비 전력 모델은 시스템-레벨에서 소비 전력을 예측 가능하고, 시뮬레이션 시간이 게이트-레벨에서의 소비 전력 예측 방법에 비해 약 1,600배 이상 빠르게 실행이 가능하며, 6%의 오차를 보였다. 제안된 비동기식 소비 전력 모델을 이용할 경우 SoC 시스템 레벨에서의 소비 전력 예측에 활용할 수 있다. 따라서, 제안된 소비 전력 모델은 A8051 프로세서뿐만 아니라 SoC에 내장 가능한 다른 다양한 비동기식 프로세서에 활용할 수 있을 것으로 기대된다.

## 참고 문헌

- [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 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] 류제천, 이제훈, 조경록, "멀티미디어 SoC용 시스템 버스의 소비 전력 모델링 및 해석", 한국콘텐츠학회논문지, 제7권, 제11호, pp.84-93, 2007(11).
- [4] V. Tiwari, D. Sinho, S. Rajgopal, G. Mehta, P. Patel, and F. Baez, "Reducing power in high-performance microprocessor," Proc. of the DAC 1998, pp.732-737, 1998.
- [5] A. Efthymiou, *Asynchronous techniques for power-adaptive processing*, Ph.D Thesis, 2002.
- [6] S. B. Furber, D. A. Edwards, and J. d. Garside, "AMULET3: a 100MIPS asynchronous embedded microprocessor," Proc. of ICCD 2000, pp.329-334, 2000.
- [7] L. Clark, E. Hoffman, J. Miller, M. Biyani, Y. Liao, S. Strazdus, M. Morrow, K. Velarde, and M. Yarch, "An embedded 32-b microprocessor core for low-power and high-performance applications," IEEE Journal of Solid-State Circuits, Vol.36, No.11, pp.1599-1608, 2001(11).
- [8] J. Warmock, "A 5.2GHz microprocessor chip for the IBM zEnterprise™ system," Proc. of ISSCC 2011, pp.69-71, 2011.
- [9] A. Efthymiou, J. D. Garside, and S. Temple, "A comparative power analysis of an asynchronous processor," Proc. of PATMOS'01, pp.1-10, 2001.
- [10] Y. Shi, B. Gwee, and J. Cang, "Asynchronous DSP for low-power energy-efficient embedded systems," Microprocessors and Microsystems,

Vol.35, No.3, pp.318-328, 2011(3).

[11] H. Jacobson, A. Buyuktosunoglu, P. Bose, E. Acar, and R. Eickemeyer, "Abstraction and microarchitecture scaling in early-stage power modeling," Proc. of HPCA2011, pp.394-405, 2011.

[12] A. Bink and R. York, "ARM966HS: The first licensable, clockless 32-bit processor core," IEEE Micro, Vol.27, No.2, pp.58-68, 2007(3).

[13] 이제훈, 조경록, "CISC 임베디드 컨트롤러를 위한 새로운 비동기 파이프라인 아키텍처, A8051," 대한전자공학회 논문지, 제40권 SD편, 제4호, pp.85-94, 2003(4).

[14] J. H. Lee, Y. H. Kim, and K. R. Cho, "Design of a fast asynchronous embedded CISC microprocessor, A8051," IEICE Trans. on Elec., Vol.E87-C, No.4, pp.527-534, 2004(4).

[15] J. H. Lee, Y. H. Kim, and K. R. Cho, "A low-power implementation of asynchronous 8051 employing adaptive pipeline structure," IEEE T. on Circuits and Systems II-Express Briefs, Vol.55, No.7, pp.673-677, 2008(7).

[16] 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(4).

[17] 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. of CODES+ISSS 2008, pp.255-260, 2008.

[18] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: A framework for architectural-level power analysis and optimizations," Proc. of ISCA, pp.83-94, 2000.

[19] 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.

[20] 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, 1998.

### 저 자 소 개

이 제 훈(Je-Hoon Lee)

정희원



- 1998년 8월 : 충북대학교 정보통신공학과(공학사)
- 2001년 2월 : 충북대학교 정보통신공학과 통신회로및시스템공학(공학석사)

- 2005년 2월 : 충북대학교 정보통신공학과 통신회로및시스템공학(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 공학대학 전자정보통신공학부 조교수
- <관심분야> : 임베디드프로세서, 디지털 회로 설계, SoC, 임베디드 시스템 개발