

논문 2015-52-10-7

클럭 게이팅 적용회로의 상위수준 전력 모델링 (High-level Power Modeling of Clock Gated Circuits)

김종규*, 이준환**

(Jonggyu Kim and Joonhwan Yi[Ⓒ])

요약

SoC (System-on-Chip) 설계초기 상위수준에서 성능뿐만 아니라 전력 분석이 중요하다. 본 논문에서는 상위수준에서 전력 분석 정확도가 높은 클럭 게이팅 구동 신호 기반 전력 모델을 제안한다. 클럭 게이팅 구동 신호의 조합으로 전력 상태를 정의하며, 클럭 게이팅 구동 신호를 자동으로 추출하여 전력 모델을 자동으로 생성할 수 있다. 실험 결과 평균 96% 이상의 정확도를 보였으며, 상위수준에서의 전력 분석 속도는 게이트 수준 대비 평균 280배 빠른 속도향상을 보였다.

Abstract

Not only performance analysis but also power analysis at early design stages is important in designing a system-on-chip. We propose a power modeling based on clock gating enable signals that enables accurate power analysis at a high-level. Power state is defined as combinations of the values of the clock gating enable signals and we can extract the clock gating enable signals to generate the power model automatically. Experimental results show that the average power accuracy is about 96% and the speed gain of power analysis at the high-level power is about 280 times compared to that at the gate-level.

Keywords : Clock gating, Automatic power modeling, High-level power analysis, ESL power estimation.

I. 서론

상위 수준 (electronic system-level) 에서의 전력 분

석은 저 전력 SoC (system-on-chip) 설계에 필수적이다. SoC 설계에서 성능과 전력 값의 적절한 균형을 맞추기 위하여 모든 설계와 시뮬레이션 단계에서는 전력 분석이 요구된다. 하위 수준 (gate-level)에서의 전력 분석은 정확도가 매우 높은 반면에 분석 속도가 매우 느리다. SoC 설계는 계속된 제조 공정의 발달과, 설계 복잡도의 향상으로 점점 그 크기가 커지고 있어, 하위수준에서 전력을 측정하기에는 시간과 비용이 너무 크다. 이러한 이유로 전력 최적화의 효과를 높이고, 재설계의 비용을 줄이기 위하여 상위 수준에서의 전력 분석이 필요하다. 상위 수준에서는 추상화 수준이 높고 알고리즘 단위로 수행하기 때문에 SoC 시스템 전체에 대해서 빠르게 시뮬레이션이 가능하다. 또한, 설계 구조를 변경하며 전력과 성능을 분석 비교 할 수 있다.

상위 수준에서의 전력 모델링에는 top-down 기법과

* 학생회원, ** 평생회원, 광운대학교 컴퓨터공학과
(Dept. of Computer Engineering, Kwangwoon University)

Ⓒ Corresponding Author(E-mail: joonhwan.yi@kw.ac.kr)

※ 이 논문은 2013년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (No. NRF-2011-0025385)

※ 이 논문은 2013년도 지식경제부 글로벌전문기술개발사업으로 지원된 연구결과입니다. [10047664, 저전력 설계를 위한 고속 (게이트수준 대비 300배) 정확한 (게이트수준 대비 80%) 전력모델 자동생성 소프트웨어 개발]

※ 이 논문은 IDEC의 지원을 받아 제작되었습니다.

Received ; June 22, 2015 Revised ; August 17, 2015

Accepted ; September 30, 2015

bottom-up 기법이 있다. Top-down 기법은, 합성된 하위 수준 코드(netlist)없이 RTL (register transfer level)에서 관측 가능한 함수의 입출력과 레지스터만으로 조합 논리 회로의 스위칭 확률과 게이트를 예측하는 방법^[1], 그리고 이미 분석되어진 기초 함수들을 맵핑하여 캐패시턴스를 예측하고, RTL 시뮬레이터를 이용하여 스위칭 확률을 측정하는 방법^[2]들이 있다. 이러한 top-down 기법들은 설계의 초기 부분에서 빠르게 전력 분석 예측이 가능하나, 게이트 수준으로의 합성에 따른 설계의 구조 및 저전력 기법들(클럭 게이팅, 데이터 게이팅, 파워 게이팅 등)을 반영하지 못하기 때문에 정확도가 낮다.

Bottom-up 기법은 모든 하위 수준의 특징뿐만 아니라 저전력 기법들을 반영할 수 있어 재사용성이 높은 설계에 대해서 활용할 수 있다. 다양한 파라미터를 기반으로 하위수준에서 분석한 전력을 매크로 모델로 생성하고, 상위수준에서 이 파라미터를 기반으로 매크로 모델의 테이블을 참조하여 전력을 빠르게 분석^[3]한다. 기존의 연구들은 설계자가 내부 구조를 파악해서 매크로 모델을 생성하거나 작은 조합 논리 회로에서만 적용이 가능하였다. 본 논문에서는 새로운 bottom-up 기술로서 클럭 게이팅 구동신호를 파라미터로 한 기법을 제안한다. 클럭 게이팅 기술이 적용된 회로에 대해서 자동으로 파라미터를 추출함으로써 설계자의 내부 구조 분석 없이 전력 모델을 자동 생성할 수 있으며, 크기가 큰 순차회로에 적용이 가능한 기술이다. 본 연구에 의한 실험 결과 평균 96% 이상의 정확도를 보였으며, 상위수준에서의 전력 분석으로 게이트 수준 대비 전력 분석 속도는 평균 280배 빠른 속도향상을 보였다.

나머지 장의 구성은, II 본문에서는 기존 전력 모델링 관련 연구를 살펴보고, 클럭 게이팅 기법에 대한 설명과 이를 이용한 본 논문에서 제안하는 전력 모델링 방법을 설명한다. 그리고 상위수준에서 전력을 분석하는 방법에 대해서 알아본다. 클럭 게이팅 구동 신호 기반 전력 모델링 기술의 상위수준 전력 분석 정확도와 속도를 III 실험에서 게이트 수준과 비교해 본다. IV 결론에서는 고찰을 다룬다.

II. 본 론

1. 관련 연구

최근의 전력 모델링 기법^[4~10]으로는 동작 상태 혹은 내부 블록의 동작 횟수 등으로 전력 값들을 분류하여 전력 모델을 생성하는 방법이 제안되었다. 이러한 기법들은 내부 블록의 동작상태, 구조를 파악함으로써 효과적인 전력 모델 생성이 가능하다. 다목적 프로세서나 버스, 메모리의 경우에는 이미 많은 전력 모델 방법들이 제안되었다^[4]. 프로세서는 입력되는 명령어, 사칙연산, 메모리 접근, 분기 등에 따라서, 또는 캐시 접근 비율, 메모리 접근 비율, 동작 주파수 등에 따라서 전력 모델링^[5]을 생성한다. 버스는 데이터 전송 혹은 유희상태에 따라서, 그리고 SDRAM의 경우에는 다섯 가지 동작 모드(precharge, activate, read, write, refresh)와 전력을 적게 소모하는 power down 모드, 그리고 클럭이 공급되지 않더라도 데이터를 유지하기 위한 self-refresh 모드에 따라서 전력 값을 분류^[6]하였다. 이러한 재사용성이 높은 블록 들은 동작이 분명하고 일정한 패턴으로 정해져있어 내부 동작을 쉽게 파악할 수 있다. 때문에 정확하고 효율적인 전력 모델 생성이 가능하다. 하지만 제안된 기술들은 모두 설계자가 회로 내부를 분석하고 그에 맞게 전력 모델링을 수행함으로써 설계자에 따라서 전력 모델의 질이 달라 질 수 있다.

내부 구조를 파악하지 않고 입력 신호 또는 출력 신호를 이용하는 기법들도 제안되었다. 이전 입력 값을 행으로, 현재 입력 값을 열로 가지고, 각 항목에는 전력 값을 가진 입력 벡터를 이용한 테이블을 사용하는 기법이 있다. 하지만 입력 벡터의 수가 많아지면 테이블의 크기가 4^n (n :입력 비트 수) 지수 적으로 증가하기 때문에 크기가 큰 회로에서는 사용할 수 가 없다. 이러한 테이블의 크기를 줄이기 위하여 회로를 분할하거나^[7], 비슷한 전력 값들을 묶는 클러스터링 기법^[8], 또는 입력 벡터의 hamming distance를 파라미터로 사용하는 기법^[9], 그리고 네 개의 파라미터 (average input signal probability, average input transition density, average spatial correlation coefficient, 그리고 average output zero-delay transition density)를 사용하여 4-D 테이블을 사용한 기법^[10]들이 제안되었다. 하지만 입출력에 의존하기 때문에 내부 동작에 따른 변화를 추적하기 어려워 작은 크기의 조합 논리 회로에서만 가능하다.

2. 제안 동기

본 논문에서는 전력 모델을 자동으로 생성할 수 있는

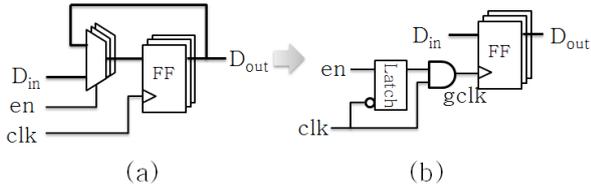


그림 1. 클럭 게이팅 셀이 (a) 삽입되지 않은 경우와 (b) 삽입된 경우

Fig. 1. A gate-level implementation (a) without clock gating and (b) with clock gating

기술을 제안한다. 클럭 게이팅 기술은 상용 합성 툴들을 이용하여 게이트 수준으로 합성 시 자동으로 삽입된다. 이러한 점을 이용하여 클럭 게이팅 구동 신호를 자동으로 추출할 수 있게 되고, 전력 모델을 자동 생성할 수 있다. 클럭 게이팅 구동 신호를 자동으로 추출할 수 있어 회로의 내부구조를 파악할 필요가 없으므로 누구나 사용가능하며, 또한 누가 만들어도 일정한 정확도를 유지한다. 클럭 게이팅 기술은 플립플롭뿐만 아니라 클럭 트리 네트워크^[13]에 의해 전력 소모량이 감소한다. 클럭 게이팅 기술로 최대 70%까지 동적 전력 소모량을 줄일 수 있음이 여러 논문에 보고되어 있다.^[12~14] 이처럼, 동적 전력 소모량은 클럭 게이팅 셀의 on/off에 따라 크게 차이가 발생하기 때문에, 본 논문에서는 클럭 게이팅 구동 신호를 파라미터로 한 전력 모델링 기법을 제안한다.

3. 클럭 게이팅

클럭 게이팅 기술은 특정 회로의 동작이 필요하지 않을 경우 회로에 클럭을 공급하지 않음으로써 동적 전력 소모를 줄이는 최적화 기술이다.^[11~12] 일반적인 소비 전력 식은 식 1과 같으며, 동적 전력과 정적 전력 수식은 식 2와 식 3에 각각 보여준다. 동적 전력 식 2에서 α 는 게이트의 스위칭 횟수를 나타내고, C_L 은 게이트의 부하 용량, V_{dd} 는 공급 전압, 그리고 f 는 클럭 동작 주파수를 나타낸다. 클럭 게이팅을 적용하여 클럭을 차단하면 플립플롭들의 f 는 0이 되고, α 도 0이 된다. 또한 플립플롭들과 연결된 조합 논리회로의 게이트들의 α 는 크게 감소하게 되며, 해당 영역의 게이트의 스위칭이 감소하므로 short-circuit 전류(I_{sc}) 또한 감소한다. 결국 동적 전력 소모는 감소하며 정적 전력(P_{static}) 소모가 증가한다. 반대로 클럭을 통과시키면 플립플롭과 그와 연결된 게이트들의 스위칭 확률이 높아지면서 동적 전

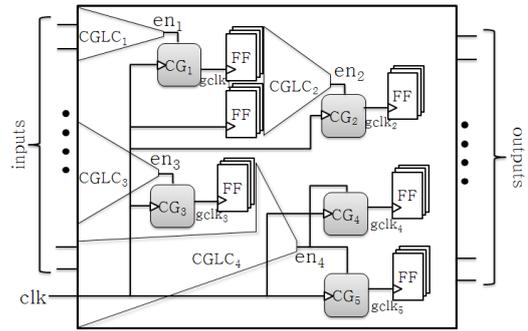


그림 2. 예제 회로 C_1

Fig. 2. Example circuit C_1 .

력 소모가 증가하고 정적 전력 소모는 감소한다. 본 논문에서 사용하는 전력 모델은 하나의 전력 상태에 대해서 동적 전력과 정적 전력을 합친 총 전력 값을 사용한다. 전력 상태는 IV장에서 설명한다.

$$P_{total} = P_{dynamic} + P_{static} \quad (1)$$

$$P_{dynamic} = \sum_i^N \alpha_i C_{L_i} V_{dd}^2 f + I_{sc} V_{dd} \quad (2)$$

$$P_{static} = I_{leakage} V_{dd} \quad (3)$$

그림 1에 클럭 게이팅 셀이 삽입되고 어떻게 동작하는지를 보여준다. 그림 1a 클럭 게이팅 셀이 삽입되지 않은 레지스터의 경우는 en 값에 따라 D_{in} 또는 D_{out} 의 값이 플립플롭에 입력으로 들어간다. 그림 1b 클럭 게이팅이 삽입된 레지스터의 경우는 클럭 게이팅 구동 신호(en)가 1인 경우에만 클럭(clk)이 게이팅된 클럭(gclk)으로 플립플롭에 전달되고, 0일 때는 gclk이 0으로 안정되어 플립플롭에 클럭이 전달되지 않는다.

4. 전력 상태

전력 모델에서 클럭 게이팅 구동 신호들의 on/off 상태의 조합으로 전력 상태를 정의한다. 어떤 C 라는 회로가 있으면, 이 회로가 n 개의 클럭 게이팅 구동 신호 $EN_C = \{en_1, en_2, \dots, en_n\}$ 를 가지고 있을 때, 클럭 게이팅 구동신호 EN_C 의 조합으로 생성될 수 있는 최대 전력 상태 수는 2^n 개이다. 예를 들어, 그림 2의 C_1 회로와 같이 4개의 클럭 게이팅 구동 신호 $EN_{C_1} = \{en_1, en_2, en_3, en_4\}$ 가 있을 때, 생성될 수 있는 16개의 전력 상태 $PS_{C_1} = \{ps_1, ps_2, \dots, ps_{16}\}$ 을

표 1. 회로 $C1$ 의 전력 상태 PS_{C1} 와 전력 값 P_{C1}
Table 1. Power states PS_{C1} of circuit $C1$ and their corresponding power values

Power state	en_1	en_2	en_3	en_4	Power value
ps_1	0	0	0	0	p_1
ps_2	0	0	0	1	p_2
ps_3	0	0	1	0	p_3
...
ps_{15}	1	1	1	0	p_{15}
ps_{16}	1	1	1	1	p_{16}

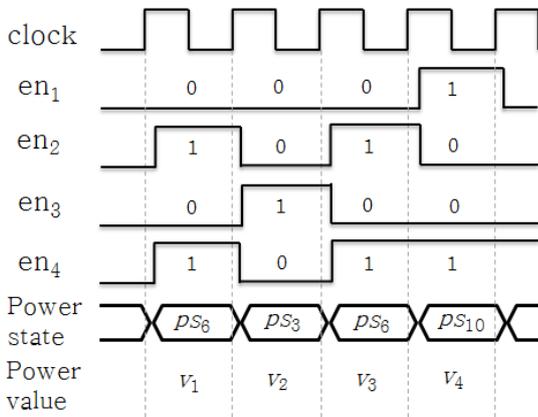


그림 3. Characterized 전력 상태와 전력 값
Fig. 3. Characterization of power state and power values.

표 1에서 보여준다. 표 1에는 각 전력 상태마다 전력 값을 가지고 있다. 이 전력 값은 회로 $C1$ 이 ps_i 일 때 소모되는 전체 회로의 전력 값 p_i 를 나타낸다. 그림 3처럼 회로 $C1$ 의 매 clock cycle별 클럭 게이팅 구동 신호를 확인하여 $\{ps_3, ps_6, ps_{10}\}$ 을 생성하고, 그 때에 소모되는 전력 값 $\{v_1, v_2, v_3, v_4\}$ 를 계산한다. 이 때 첫 번째 사이클과 세 번째 사이클에서 전력 상태 ps_6 으로 같은 전력 상태를 가지므로, 이 두 곳의 전력 값 v_1 과 v_3 를 평균 $(v_1 + v_3)/2$ 하여 ps_6 의 전력 값으로 계산한다. 이렇게 전력 상태와 전력 값을 가진 표를 전력 상태 테이블이라고 한다.

가. Characterized & uncharacterized 전력 상태

본 논문에서는 전력 상태를 characterized 전력 상태 (CPS)와 uncharacterized 전력 상태 (UPS)의 두 가지로 분류한다. 전력 상태를 추출하기 위하여 게이트 수

준에서 학습 시나리오를 이용하여 한번 시뮬레이션을 수행한다. 수행된 시뮬레이션 결과로부터 해당 회로 내부의 클럭 게이팅 구동 신호들의 매 클럭 사이클별 on/off를 추출하게 되고, 이들의 조합으로 전력 상태 PS_C 를 도출한다. 즉, CPS는 학습 시나리오로부터 도출된 전력 상태이다. 이와 반대로 UPS는 CPS에 도출되지 않았지만 잠정적으로 해당 회로가 학습 시나리오와 다른 시나리오에 의해서 도출될 수 있는 전력 상태이다. CPS는 게이트 수준에서 학습 시나리오로 전력 분석을 수행하여서 각 CPS마다 전력 값을 가지고 있다. 하지만 UPS는 전력 값도 가지고 있지 않으며, 뿐만 아니라 다른 시나리오에서 클럭 게이팅 구동 신호들이 어떠한 조합으로 도출 되는지를 알 수가 없다. 이러한 이유로 본 논문에서는 UPS가 도출 되더라도, 이들에 해당하는 전력 값을 예측하기 위하여 hamming distance를 사용한다.

나. Uncharacterized 전력 상태의 전력 값 예측

UPS의 전력 값을 예측하기 위해서 본 논문에서는 hamming distance (HD)의 2진 코드 기법을 사용하였다. CPS들 중에서 UPS와 HD가 가장 작은 CPS를 선택하여 해당 전력 값을 UPS의 전력 값으로 선택한다. 만약 HD가 가장 작은 CPS가 두 개 이상일 경우, 해당하는 CPS들의 전력 값의 평균을 계산하여 UPS의 전력 값으로 선택한다.

그림 4에 CPS들 중에서 하나의 CPS를 UPS로 간주

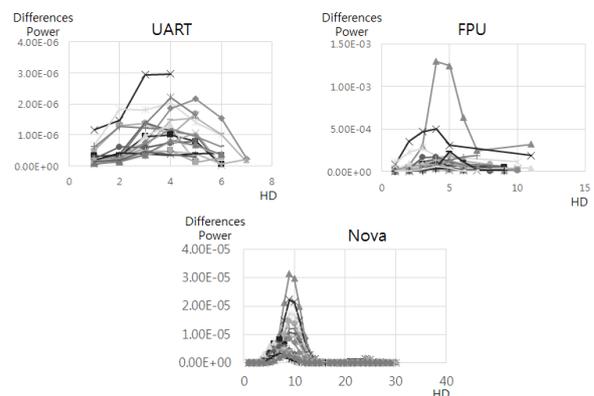


그림 4. UPS의 전력 값과 UPS로 부터 HD 값에 따른 CPS들의 전력 값 차이 그래프

Fig. 4. The graph of power differences between the power value of the CPS according to the HD value from UPS and the power value of UPS.

하였을 때, 자기 자신을 제외한 나머지 CPS들과의 HD 값에 따른 전력 값의 절대 값 차이를 보여주고 있다. 그래프를 살펴보면 HD값이 가장 적은 1 부근에서 전력 값의 차이가 제일 적게 나타나는 것을 알 수 있다. 그래프에서 중간 지점에서 가장 크게 전력 값의 차이가 나는 이유는, 예를 들어, 클럭 게이팅 구동 신호가 4개인 UPS가 0011일 때, CPS가 1111 또는 0000의 HD가 2일 때 가장 크게 전력 값의 차이가 발생하기 때문이다. 그리고 HD의 값이 커질수록 다시 전력 값의 차이가 줄어드는 이유는 CPS가 1100이면 HD가 4이지만 클럭 게이팅 구동 신호의 on된 개수가 비슷해져서, 비슷한 전력 값을 소모할 수 있기 때문이다. 하지만 가장 큰 HD값을 가지는 CPS를 선택하지 않는 이유는 0011의 UPS에 대해서 1100의 CPS가 없거나, 항상 UPS와 비슷한 값을 가진다고 보장할 수 없기 때문이다.

표 2 에 UPS의 전력 값을 구하는 알고리즘을 보여주고 있다. CPS는 학습 시나리오로 부터 총 7개가 도출되었으며, 다른 시나리오로부터 두 개의 UPS (ps_8, ps_{11})가 더 도출된 상황에서 hamming distance (HD)를 이용하여 전력 값을 예측한다. ps_8 의 경우 CPS들 중 ps_6 과의 hamming distance가 1로 가장 작으므로 p_6 을 ps_8 의 전력 값 p_8' 으로 선택한다. ps_{11} 은 hamming distance가 가장 작은 1인 CPS가 3개 (ps_3, ps_9, ps_{12})

표 2. Un-characterized 전력 상태의 전력 값 예측 예제
Table 2. Example of estimated power value of un-characterized power state.

Characterized power states				
Power state	Power value	HD by ps_8	HD by ps_{11}	
ps_1	0000	p_1	3	2
ps_2	0001	p_2	2	3
ps_3	0010	p_3	2	1
ps_6	0101	p_6	1	4
ps_9	1000	p_9	4	1
ps_{10}	1001	p_{10}	3	2
ps_{12}	1011	p_{12}	2	1
Un-characterized power states				
Power state	Estimated power value			
ps_8	0111	$p_8' = p_6$		
ps_{11}	1010	$p_{11}' = (p_3 + p_9 + p_{12})/3$		

존재하므로, 이들의 전력 값 (p_3, p_9, p_{12})들의 평균 값 ($(p_3 + p_9 + p_{12})/3$)을 계산하여 ps_{11} 의 전력 값 p_{11}' 으로 결정한다.

III. 실험

본 논문에서 제안한 클럭 게이팅 구동 신호 기반 전력 모델링 기법의 정확도와 속도를 평가하기 위하여, 다음의 3개의 회로(출처: Opencores.org)를 사용하였다. uart: 범용 비동기화 송수신기 버스, fpu: 64bit 부동 소수점 사칙 연산기, nova: H.264/AVC baseline 모바일 어플용 디코더. 각 회로들은 모두 TSMC 0.13um 공정 라이브러리를 사용하여 Synopsys사의 DesignCompiler로 합성하였으며, 동사의 PowerCompiler를 이용하여 클럭 게이팅 셀을 자동 삽입 하였다. 게이트 수준 전력 분석은 동사의 PrimeTime-PX를 사용하였다.

표 3에 게이트 수준 합성 정보를 보여준다. Uart, fpu, 그리고 nova는 각각 클럭 게이팅 구동 신호가 16개, 13개, 그리고 148개씩 삽입되었으며, CPS는 22개, 26개, 그리고 2086개씩 추출되었다.

각 회로에 대해서 사용한 시나리오는 표 4와 같다.

표 3. uart, fpu, 그리고 nova 회로들의 정보
Table 3. Circuit information of uart, fpu, and nova.

Circuit	Area (gate count)	Gated flipflop ratio	Number of CGEN	Number of CPS
uart	1.8K	81.9%	16	22
fpu	63K	89.7%	13	26
nova	305K	99.3%	148	2086

표 4. 학습 시나리오와 테스트 시나리오들
Table 4. Description of the characterization and test scenarios.

Circuit	Characterization scenario	Test scenarios		
		TS1	TS2	TS3
uart	1 write/ 1 read	20 writes	20 reads	20 random
fpu	50 add/sub/ mul/div	20 add/sub	20 mul/div	95 add/sub/ mul/div
nova	Carphone	Bridge	Hall	Foreman

각 첫 번째 열에 적힌 시나리오가 전력 모델을 생성하는 데 사용한 학습 시나리오이고, 나머지 세 개는 테스트 시나리오이다. Uart는 한번의 쓰기와 한번의 읽기를 학습 시나리오로 사용하였으며, 테스트 시나리오로 20번의 쓰기, 20번의 읽기, 그리고 20번의 랜덤 패턴을 사용하였다. Fpu는 64bit 부동 소수점의 사칙연산에 대해 순서에 상관없이 총 50번의 수행을 학습 시나리오로 사용하였으며, 테스트 시나리오로 50번의 덧셈과 뺄셈의 조합, 50번의 곱셈과 나눗셈의 조합, 그리고 95번의 사칙연산을 사용하였다. Nova는 carphone 이미지 3프레임 디코딩을 학습 시나리오로 사용하였으며, bridge, hall, 그리고 foreman의 각 4프레임 디코딩을 테스트 시나리오로 사용하였다. 상위수준 전력 분석은 Carbon사의 SoC Designer에 게이트 수준 시뮬레이션과 동일한 시스템 환경을 구성하고 수행하였다.

표 5는 게이트 수준 대비 본 논문에서 제안한 전력 모델의 정확도를 보여준다. 시나리오가 N 클럭 사이클 동안 동작한다고 했을 때, $p_i, (i = 1, 2, \dots, N)$ 는 게이트 수준에서 i 번째 클럭 사이클의 전력 분석 값이고, p'_i 는 ESL 수준에서 본 논문에서 제안한 기술을 적용하여 예측한 i 번째 클럭 사이클의 전력 예측 값이다. 표 5의 각 항의 첫 번째 값은 전체 평균 오차(Avg. Error)를 나타낸다.

$$Avg. Error = \frac{|p_{avg} - p'_{avg}|}{p_{avg}} \cdot 100\% \quad (4)$$

표 5의 괄호안의 값은 사이클별 오차의 평균(AACPE:average absolute cycle per error)을 나타낸다.

표 5. 게이트 수준 대비 전력 모델의 오차율 (매 클럭 사이클별 오차 값의 평균)

Table 5. Absolute error rates of power model compared to the gate-level power. (Average absolute cycle power error).

Circuit	TS1	TS2	TS3	Average
uart	0.57% (6.1%)	0.65% (7.7%)	0.07% (6.3%)	0.43% (6.7%)
fpu	25% (26%)	0.67% (2.9%)	0.14% (2.4%)	8.62% (10.4%)
nova	0.76% (0.9%)	0.73% (1%)	0.02% (1.5%)	0.51% (1.13%)
Average				3.18% (6.1%)

$$AACPE = \frac{1}{N} \left(\sum_i^N \frac{|p_i - p'_i|}{p_i} \cdot 100\% \right) \quad (5)$$

Uart의 테스트 시나리오에 대한 전체 평균 정확도는 99.57%, fpu의 평균 정확도는 91.38%, 그리고 nova의 평균 정확도는 99.49%로 매우 정확한 결과를 보였다. 전체 평균 AACPE는 6.1%를 보였다. 실험 결과 중에 fpu의 테스트 시나리오 1의 결과가 오차가 크게 발생하였다. 그 원인으로서는 UPS의 출현 빈도가 fpu의 테스트 시나리오 1 (TS1)을 제외한 모든 경우에서는 2% 이하이고, fpu의 TS1만 99.88%로 매우 높기 때문이다. 현재의 hamming distance가 UPS의 전력 값을 만족할만하게 예측하지 못함을 나타내고 있다. 이는 추후 CPS를 기반으로 통계적인 예측 기술을 적용하여 정확도를 높일 예정이다.

표 6에서 상위수준 전력 모델의 전력 분석 속도 향상률을 보여주고 있다. 상위수준에서의 전력 분석 속도 향상은 uart는 평균 30배, fpu는 452배, 그리고 nova는 356배의 속도향상을 보였다. 상위수준에서의 속도 향상은 회로의 크기가 커질수록, 그리고 추출된 전력 상태의 개수가 적을수록 더 높은 속도 향상률을 보였다. 추가적으로 상위수준에서 전력 분석에 따른 오버헤드는 uart가 1.3배, fpu가 1.5배, 그리고 nova는 2.3배의 속도 저하가 발생한다. 전력 모델을 생성하는 시간은 회로의 크기와 전력상태의 개수, 그리고 학습 시나리오의 길이

표 6. 게이트 수준 대비 전력 분석 속도 향상
Table 6. Speed gains compared to the gate-level power analysis.

Circuit	TS1	TS2	TS3	Average
uart	29	32	30	30
fpu	486	539	331	452
nova	334	344	391	356
Average				280

표 7. 전력 모델 생성 시간 (초)

Table 7. Time for power model generation (In seconds).

Circuit	gate level simulation	gate level power analysis	Power modeling tool	Total
uart	4.7	72	10.5	87.2
fpu	1.4	53	2.2	56.6
nova	94.8	2724	296	3114.8

에 비례한다. 표 7에서 가장 큰 nova의 경우 전력 모델 생성이 약 1시간정도 소요되어 생성할 수 있다.

IV. 결 론

본 논문에서는 클럭 게이팅 구동 신호 기반 상위수준 전력 모델 자동 생성 기법을 제안하였다. 전력 모델을 설계자의 회로 내부 구조 분석 없이 자동으로 생성할 수 있어 상용 툴로 개발이 가능하다. 차후 연구로 UPS의 예측 정확도를 높이기 위해서 CPS를 기반으로 통계적 분석을 통한 예측을 수행 할 것이다. 본 논문에서는 3개의 회로만을 이용하여 전력 모델의 정확도와 속도를 분석하였는데, 회로의 크기가 더욱 큰 모델에 대해서 실험을 통하여 제안된 기술의 효과를 검증할 것이다.

REFERENCES

- [1] S. Sambamurthy, J. A. Abraham, and R. S. Tupuri, "A robust top-down dynamic power estimation methodology for delay constrained register transfer level sequential circuits," *21st International Conference on VLSI Design*, pp. 521-526, Jan. 2008.
- [2] R. P. Llopis and K. Goossens, "The petrol approach to high-level power estimation," in *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 130-132, 1998.
- [3] Y. Samei, R. Domer, "Automated Estimation of Power Consumption for Rapid System Level Design," IEEE International Performance Computing and Communications Conference, Dec, 2014.
- [4] S. Reda and A. N. Nowroz, "Power Modeling and Characterization of Computing Devices: A Survey," *Foundations and Trends in Electronic Design Automation*, vol. 6, no. 2, pp. 121-216, Feb. 2012.
- [5] S. K. Rethinagiri, O. Palomar, O. Unsal, A. Cristal, R. B. Atitallah, and S. Niar, "PETS: Power and Energy Estimation Tool at System-Level," in *Quality Electronic Design (ISQED)*, 15th International Symposium on, pp. 535-542, Mar. 2014.
- [6] K. Chandrasekar, B. Akesson and K. Goossens, "Improved Power Modeling of DDR SDRAMs," *14th Euromicro Conference on Digital System Design*, pp. 99-108, Oulu, Finland, Sep 2011.
- [7] R. Y. Chen, M. J. Irwin, and R. S. Bajwa, "Architecture-level power estimation and design experiments," *ACM Transactions on Design Automation of Electronic Systems*, vol. 6, no. 1, pp. 50-55, Jan. 2001.
- [8] H. Mehta, R. M. Owens, and M. J. Irwin, "Energy characterization based on clustering," in *Design Automation Conference*, pp. 702-707, Jun. 1996.
- [9] R. Graybill, and R. Melhem, "Power aware computing," *Springer US*, pp. 317-337, 2002.
- [10] S. Gupta, and F. N. Najm, "Power modeling for high-level power estimation," *Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 8, no. 1, pp. 18-29, Feb. 2000.
- [11] M. Powell, A. Biswas, J. Emer, S. Mukherjee, B. Sheikh, and S. Yardi, "CAMP: A Technique to Estimate Per-Structure Power at Run-time using a Few Simple Parameters," *International Symposium on High Performance Computer Architecture*, pp.289-300, Raleigh, Feb. 2009.
- [12] R. Fraer, G. Kamhi, and M. K. Mhameed, "A new paradigm for synthesis and propagation of clock gating conditions," in *Proc. of Design Automation Conference*, pp. 658-663, Jun. 2008.
- [13] Jan Rabaey, "Low Power Design Essentials," *Springer*, pp. 209-214, 327-329, 2009.
- [14] M. Mueller, A. Wortmann, S. Simon, M. Kugel, and T. Schoenauer, "The impact of clock gating schemes on the power dissipation of synthesizable register files," in *Proc. of International Symposium on Circuits and Systems*, vol. 2, pp. II - 609-612, May. 2004.

— 저 자 소 개 —



김 종 규(학생회원)
2011년 광운대학교 컴퓨터공학과
학사 졸업.
2011년~현재 광운대학교 컴퓨터
공학과 석박사통합과정.
<주관심분야 : 상위수준 전력모델,
SoC 구조설계, 저 전력 설계>



이 준 환(평생회원)
1991년 연세대학교 전자공학과
학사졸업.
1998년 Univ. of Michigan, EECS
석사졸업.
2002년 Univ. of Michigan, EECS
박사졸업.
1991년~1995년 삼성전자 시스템LSI 연구원.
2003년~2008년 삼성전자 통신연구소 수석연구
원.
2008년~현재 광운대학교 컴퓨터공학과 부교수.
<주관심분야 : SoC 구조설계, 저전력 설계, 반도체
설계, Computer Vision>