

논문 2010-47SD-2-14

기가 스케일 SoC를 위한 통합 설계 방법론 및 검증 플랫폼

(Unified Design Methodology and Verification Platform for Giga-scale System on Chip)

김정훈*

(Jeonghun Kim)

요약

본 논문은 기가 스케일 System on Chip(SoC)를 위한 통합 설계 및 검증 플랫폼을 제안한다. VLSI 집적도의 발달로 그 복잡도가 증가하여 기존의 RTL 설계 방식으로는 그 생산성 차이(Production Gap)를 극복할 수 없게 되었다. 또한, 검증 차이(Verification Gap)의 증가로 검증 방법론에도 커다란 변혁이 필요하게 되었다. 본 플랫폼은 기존의 상위 수준 합성을 포함하며, 그 결과물을 이용하여 저 전력 설계의 전원 인식 검증 플랫폼과 검증 자동화를 개발하였다. 상위 수준 합성 시 사용되는 Control and Data Flow Graph (CDFG)와 그 입력인 상위 수준 언어와 RTL를 기반으로 한 검증 플랫폼 자동화와 전원 인식 검증 방법론을 개발하였다. 검증 플랫폼에는 자동 검사 기능을 포함하고 있으며 Coverage Driven Verification을 채택하고 있다. 특히 전원 인식 검증을 위하여 개발된 조건 랜덤 벡터 생성 알고리즘을 사용하여 랜덤 벡터의 개수를 최소 5.75배 감소시키는 효과를 가져왔고, 전원과 전원 셀에 대한 모델링 기법을 이용하여 일반적인 로직 시뮬레이터 툴을 통해서도 전원 인식 검증을 가능하게 하였다. 이러한 통합된 설계 및 검증 플랫폼은 시스템 수준의 설계에서 검증, 합성에 이르는 전 설계 흐름을 완전 자동화 하여 상위 수준의 설계와 검증을 가능하게 하고 있다.

Abstract

We proposed an unified design methodology and verification platform for giga-scale System on Chip (SoC). According to the growth of VLSI integration, the existing RTL design methodology has a limitation of a production gap because a design complexity increases. A verification methodology need an evolution to overcome a verification gap. The proposed platform includes a high level synthesis, and we develop a power-aware verification platform for low power design and verification automation using it's results. We developed a verification automation and power-aware verification methodology based on control and data flow graph (CDFG) and an abstract level language and RTL. The verification platform includes self-checking and the coverage driven verification methodology. Especially, the number of the random vector decreases minimum 5.75 times with the constrained random vector algorithm which is developed for the power-aware verification. This platform can verify a low power design with a general logic simulator using a power and power cell modeling method. This unified design and verification platform allow automatically to verify, design and synthesis the giga-scale design from the system level to RTL level in the whole design flow.

Keywords : Design Methodology, Verification Automation, High-Level Synthesis (HLS), Power-aware Verification, Constrained Random Vector, Testbench Automation

정회원, 로스앤젤레스 캘리포니아 주립대학교 컴퓨터 과학과

(Department of Computer Science, University of California, Los Angeles)

* 이 논문 또는 저서는 2008년 정부(교육과학기술부)의 재원으로 한국연구재단 (구 한국 학술진흥재단)의 지원을 받아 수행된 연구임.

[이공분야 : KRF-2008-357-D00236].

접수일자: 2009년11월11일, 수정완료일:2010년2월2일

I. 서론

최근 설계 기술의 발달로 기가 스케일의 System on Chip (SOC)과 VLSI 설계의 복잡도가 날로 증가함에 따라 설계 기간 단축과 시스템 설계의 용이성을 위해 상위 수준 합성^[1]과 플랫폼 기반 설계^[2], 통합 검증 방

법론^[3-4] 등에 관한 연구와 다양한 CAD 툴들이 소개되고 있다. 이 중 상위 수준 합성은 1994년 처음으로 소개된 후^[1], 다양한 알고리즘과 툴들이 학계와 산업계에서 개발되고 발표 되었다^[1~2, 5~7]. 이러한 툴들은 추상 및 시스템 수준 언어인 C/C++, SystemC으로 기술된 알고리즘이나 시스템을 직접 Hardware Description Language (HDL) 코드를 생성하여 시스템 수준에서 설계가 가능하게 하고 시스템 레벨에서 하드웨어의 다양한 제한들인 전력 소모와 동작 주파수, 성능 향상, 검증 시간 등을 고려하는 것을 가능하게 하고 있다^[2]. 또한 최근에는 다중 전압과 전원 차단, 다중 임계 전압 등의 VLSI 및 SOC설계에서 사용되고 있는 저 전력 설계 기술을 상위 수준 합성에 도입하여 많은 연구가 이루어지고 있다^[5~7]. 그러나 저 전력 설계 방법론을 이용하여 생성된 결과물에 대한 검증 방법론에 대한 연구는 미비한 상태이다.

상위 수준 합성을 위한 검증 방법에도 많은 연구가 이루어져 정규 검증(Formal Verification)^[8], 순차적 등가 (Sequential Equivalence Checking)^[9] 등을 이용한 연구 및 알고리즘이 소개되었다. 그러나 이러한 검증 방법들은 한 블록 수준의 검증에서 이루어지고 있어 시스템 수준의 검증에 한계를 가지고 있으며 상위 수준 합성에서 수행된 저 전력 설계에 대한 검증은 이루어지고 있지 못하다. 또한, 내부의 Control and Data Flow Graph (CDFG)와 Register Transfer Level (RTL) 생성 이전의 DB에 대한 정규 검증만을 수행하여 최종 결과물인 RTL에 대한 검증이 미흡한 점을 가지고 있다. 이를 보완하기 위하여 상위 수준 합성에서 검증을 위한 함수 모델 (functional model)을 널리 사용되고 있는 Design-For-Verification (DFV) 방법론을 상위 수준 합성에 적용하여 Synthesis For Verification (SFV) 방법론을 사용할 것을 제안하였다^[10]. 그러나 이 논문에서는 그 필요성과 기본 원칙들만을 제시하였다.

본 논문에서는 기존에 바인딩, 스케줄링, 다중 전원 등을 이용한 다양한 알고리즘을 이용한 xPilot^[2]를 토대로 검증 자동화 (Verification Automation)^[11], 전원 인식 검증 (Power-aware Verification), 조건 랜덤 벡터 생성 (Constrained Random Vector Generation)^[12] 기법을 이용하여 통합 설계 및 검증 플랫폼을 개발 하였다.

본 논문은 II장에서 기존의 상위 수준 합성에 대하여 기술하고, III장에서는 제안된 통합 설계 및 검증 플랫폼의 전체 흐름도와 구조에 대해서 기술하였다. IV장에서는

전원 인식 검증, 조건 랜덤 벡터 생성 방법에 대해서 기술하였다. V장에서는 그 실험 결과, VI장에서 결론을 기술하였다.

II. 기존 상위 수준 합성

기존의 상위 수준 합성 시스템이 그림 1에 나타나 있다^[1~2]. 상위 수준 합성은 크게 스케줄링과 바인딩으로 이루어져 있다. 상위 수준 언어인 C/C++, SystemC를 입력으로 받아 컴파일러가 내부 표현(Internal Representation)으로 변환한 후 설계 제약을 고려하여 바인딩과 스케줄링으로 이루어진 상위 수준 합성을 수행하게 된다. 이후에 RTL 생성기가 ASIC이나 FPGA 단계로 이행할 수 있는 RTL를 생성하게 된다. 이렇게 합성된 설계의 검증은 설계자에 의존된 방식이나, 제한되고 독립적인 검증 방법에 수행되고 있다. 이러한 검증 방법은 상위 수준 합성의 장점인 시스템 수준의 검증과 설계기간의 단축을 방해하는 요소가 되고 있다.

최근 저 전력 설계 방식을 위하여 Unified Power Format (UPF)^[13]을 사용한 방법이 사용 되고 있고, 상위 수준 합성에서도 이러한 저 전력 설계 알고리즘이 제안 되고 있다^[5~7]. VLSI나 SoC 설계 단계에서 저 전력 설계에 대한 검증 방법은 기존의 CAD 툴 업체에서 소개 되고 있다. 그러나 상위 수준 합성에서 설계된 저 전력 설계의 결과물을 위한 검증 방법에 대한 연구는 미미한 상황이다. 본 논문에서는 상위 수준 합성에서 필요한 통합 설계 방법론과 검증 플랫폼의 제안한다. 제안된 설계 및 검증 플랫폼은 검증 자동화, 전원 인식 검증, 조건 랜덤 벡터 생성을 포함하고 있다.

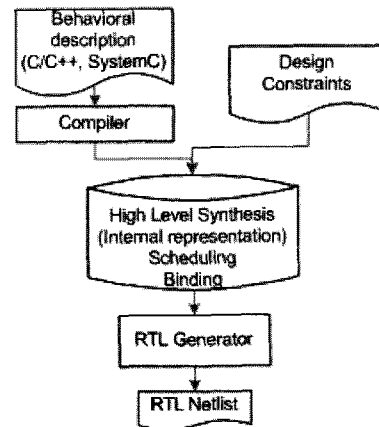


그림 1. 기존 상위 수준 합성 시스템
Fig. 1. The existing high level synthesis system.

III. 통합 설계 및 검증 플랫폼

상위 수준 합성을 포함한 시스템 수준의 통합 설계 및 검증 플랫폼을 그림 2에 나타내고 있다. 시스템 수준 언어인 C/C++를 입력으로 받아 상위 수준 합성 과정을 거쳐 검증 자동화, 전원 모델을 통한 시스템 검증을 가능하게 하고 있다. 본 플랫폼에서는 로직 합성을 위한 기본 스크립트를 생성하여 시스템 레벨에서 로직 레벨에 이르는 통합 설계 방법론을 제시하고 있다. 다음에서는 각각의 과정에서 이루어지는 내용에 대해 자세히 기술하도록 하겠다.

본 플랫폼에서 상위 수준 합성을 위한 단계는 기존에 제안된 xPilot^[2]를 기반으로 하여 구성하였다. 기존에 발표된 우수한 바인딩 및 스케줄링 알고리즘을 그대로 사용하였으며^[5-6], 최근에 저 전력 설계를 위하여 발표된 Unified Power Format (UPF)^[13]를 적용하여 RTL생성과 함께 이루어지고 있다. UPF는 전원 정보를 RTL 설계나 그 이전 단계에서 구현하는 것을 가능하게 해주며 여러 EDA 회사에서 채택하고 있다. 본 논문에서는 RTL과 함께 생성된 UPF를 설계 및 검증 과정에서 사용하고 있다. 상위 수준 합성의 결과물인 RTL과 상위 수준 언어인 C 코드의 기능 검증을 주목적으로 하고 있다. 검증 자동화를 위하여 RTL을 생성하는 단계에서 검증 자동화 단계를 추가하였다. 이 단계에서는 모든 과정은 완전히 자동화되어 C 코드로부터 설계 결과물

은 물론 검증 플랫폼과 테스트벤치를 포함한 랜덤 벡터 입력, 자동 검사(Self Checking)와 실행 스크립트까지 자동적으로 생성하게 된다. 이러한 기본적인 검증 플랫폼은 저 전력 설계 검증과 통합 설계 및 검증 방법론에 기본 요소로 사용된다. 전원 모델 생성은 UPF에 기술된 다중 전원, 전원 차단, 전원 조정 신호, 전원 기초 요소들에 대한 분석 후 시뮬레이션을 위한 모델을 생성하게 된다. 이렇게 생성된 전원 모델과 RTL, 테스트벤치는 검증 단계에서 C코드와 RTL의 등가 검사를 수행하게 된다.

1. 검증 자동화

검증 자동화는 Coverage Driven Verification (CDV)^[15-16]와 랜덤 벡터, 자동 검사 기능을 생성하는 부분과 실행 스크립트를 생성하는 부분으로 나누어져 있다. 이러한 순서도는 그림 3에 잘 나타나 있다. 우선 HLS로부터 C 코드와 내부 표현인 System-level Synthesis Data Model (SSDM)이나 CDFG를 입력으로 받아 Design Under Verification (DUV)에 대한 입력 출력 정보와 인터페이스 정보, 각종 검증에 필요한 정보를 수집하여 테스트벤치 내부에 드라이버, 랜덤 벡터 생성부, Coverage 검사부, 등가 검사 부를 생성하게 된다^[14]. 또한 이러한 검증 플랫폼을 Synopsys VCS 시뮬레이션 환경에서 실행할 수 있는 스크립트를 자동으로

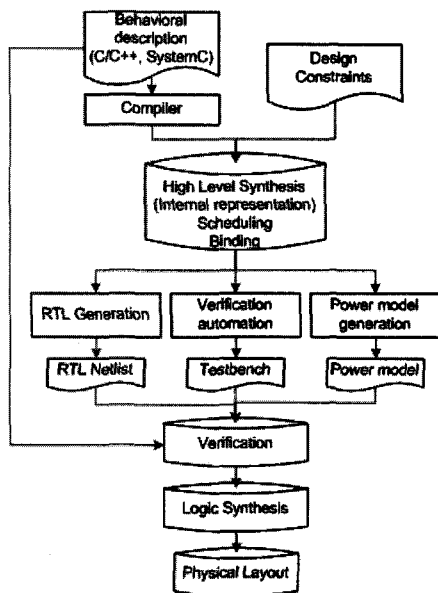


그림 2. 통합 설계 및 검증 플랫폼
Fig. 2. Unified Design and Verification Platform.

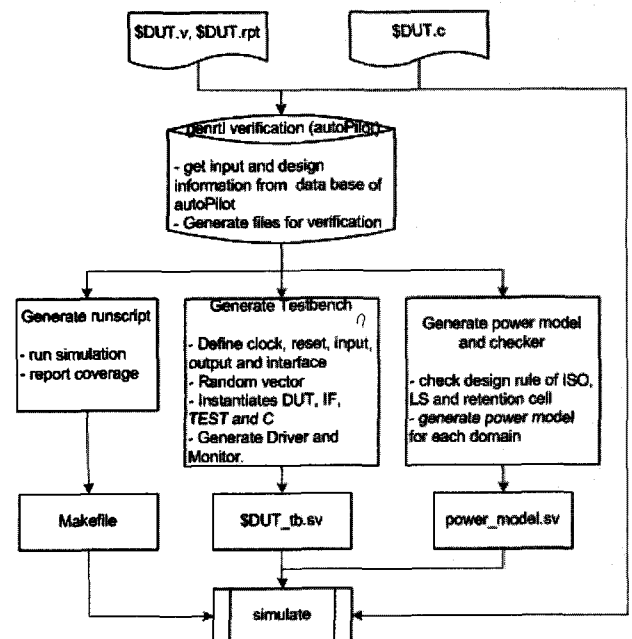


그림 3. 검증 자동화 흐름도
Fig. 3. The verification automation flow.

생성한다.

전원 인식 검증은 전원 모델을 생성하는 부분과 Functional Coverage Group, 조건 랜덤 벡터를 생성하는 부분을 가지고 있다. Functional Coverage Group은 상태와 상태 전이 두 부분으로 하여 UPF 파일에 있는 모든 전원 모드와 각 모드에서 전이 가능한 상태(State)를 Coverage 항목으로 가지게 된다. 이러한 기능 Coverage 그룹은 RTL 시뮬레이션 시에 검사되며 목표 Coverage에 도달할 때까지 조건 랜덤 벡터를 계속 생성하며 반복 수행하게 되고 검증 결과에 대한 평가가 이루어져 검증에 대한 신뢰성과 추가 검증에 대한 검토에 기본 자료로 활용 될 수 있다.

2. 검증 플랫폼

검증 플랫폼은 그림 4와 같이 DUT를 중심으로 데이터 입출력을 위한 기본 인터페이스를 가지고 있는 Transactor와 랜덤 벡터를 생성 하는 드라이버, C 코드와의 등가를 위한 자동 검사부(Self-Checker) 로 이루어져 있다^[14]. 또한 모니터(Monitor)에서는 CDV 방법론을 사용하여 DUT에 대한 Coverage를 검사하여 코드 Coverage (Code Coverage)가 100%를 만족할 때까지 반복 실행하며, 매 벡터에 대한 등가 검사를 실행하게 된다. 이러한 플랫폼은 SystemVerilog로 기술되며, Makefile기반의 실행 script도 함께 생성되어 사용자가 간편하게 설계 결과물에 대한 검증을 수행할 수 있다.

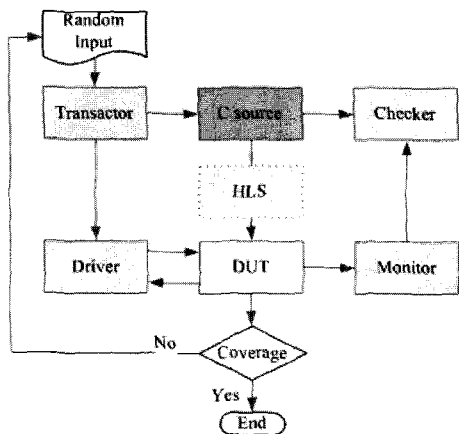


그림 4. 검증 플랫폼
Fig. 4. Verification Platform.

IV. 전원 인식 검증

모든 RTL 및 로직 시뮬레이션은 전원에 대한 정보를

가지고 있지 못하여, 일반적으로 전원 차단이나 다중 전원 설계에 있어서 그 검증은 또 다른 틀이나, Back-end 작업이후의 Gate Level이나 Physical Level에서 수행하게 된다. 이에 따라 설계의 오류를 후 공정에서 발견하게 되었을 때, 예러 보완이나 수정을 위한 기회비용과 설계 기간의 손실을 가지고 오게 된다. VLSI의 저 전력 설계의 검증을 위하여 스펙을 근거로 하여 해당하는 입력을 발견적 방법에 따라 작성하게 된다. 이러한 방법의 각 모드에 대한 누락이나 실수를 발생하게 된다. 본 연구에서는 이러한 전원과 전원 셀에 대한 모델을 HLS에서 생성된 UPF, CDFG으로부터 생성하게 된다. 또한 저 전력 설계에 대한 검증을 위하여 기능 Coverage 그룹(Functional Coverage Group)과 랜덤 벡터의 조건을 자동으로 생성하여 각 전원 모드에 대해, 기본 검증 플랫폼을 이용하여 검증하게 된다.

1. 전원 인식 검증 흐름도

그림 5는 전원 인식 검증 흐름도를 나타낸다. UPF 파일과 CDFG, RTL로부터 전원과 전원 셀, 전원 모드, 전원 제어 신호 정보를 추출한다. 이러한 정보를 토대로 전원과 그 셀을 위한 모델을 SystemVerilog을 이용하여 생성하게 된다. 이후 Power Functional Coverage Group과 조건 랜덤 벡터를 기본 검증 플랫폼에 추가

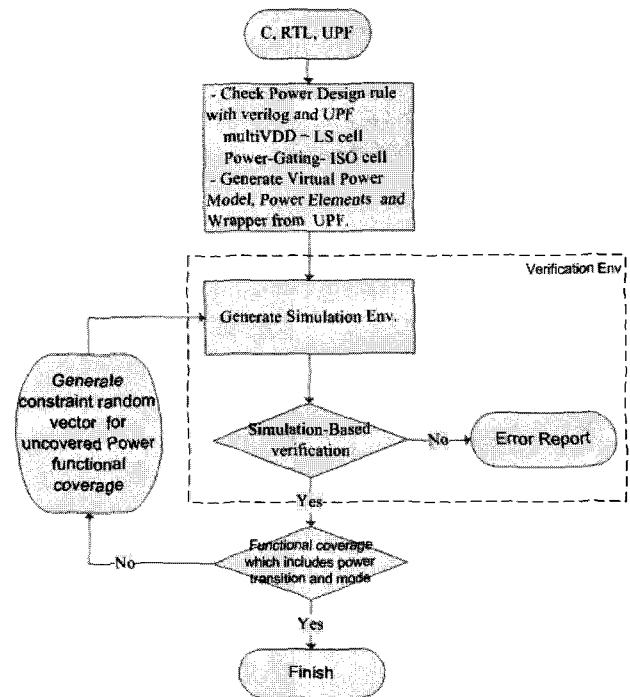


그림 5. 전원-인식 검증 흐름도
Fig. 5. Power-aware Verification flow.

하여, Functional Coverage와 코드 Coverage를 일반 랜덤 벡터와 조건 랜덤 벡터를 이용하여 모든 전원 모드에 대해 기능과 저 전력 설계 기능에 대한 검증을 수행한다.

2. 전원 및 전원 셀 모델링

전원과 전원 셀에 대한 모델은 UPF 파일로부터 전원과 각 전원 모드, 전원 제어 신호, 전원 분리를 위해 필요한 전원 기본 요소의 정보를 수집하여 RTL 시뮬레이션에 필요한 모델을 자동으로 생성한다. 전원 셀은 Level Shifter와 Isolation Cell, Retention Cell, Power Down Block에 대한 전원 모델 등으로 이루어진다. 이 중 Isolation Cell에 대한 개념도가 그림 6에 나타나 있다. 이 모델은 전원 차단 블록의 전원 차단 시 모든 출력에 대해 미지 상태 (Unknown State)를 발생시키면 이러한 미지 값이 다른 블록으로 전파되는 것을 Isolation cell로 방지되는 기능을 모델링하여 전원 셀이 적절히 설계되었는지를 검증하게 된다. Level Shifter의 경우는 다중 전원을 사용하는 블록 간에 연결되는 신호에 Isolation cell과 같은 방법의 기능을 추가하여 구현하였다. Retention Cell의 경우에는 전원 차단 블록의 레지스터에 가상의 레지스터를 추가하여 전원의 On/Off시에 데이터를 저장 및 출력하도록 모델링하였다. 이러한 모델링의 추가는 UPF를 기반으로 하여 생성되어 상위 수준 합성에서 생성된 UPF에 잘 못된 저

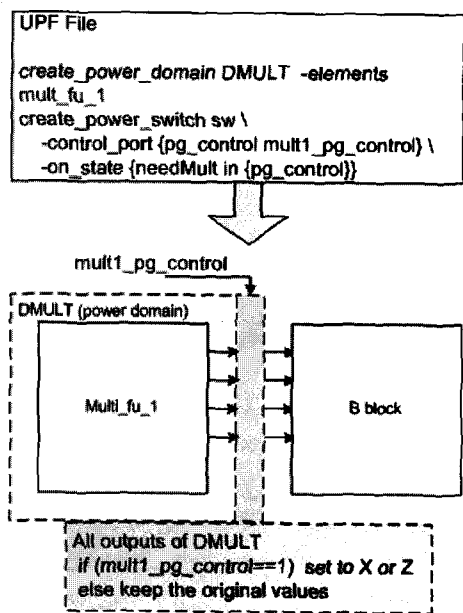


그림 6. 전원모델의 개념
Fig. 6. The concept of power model.

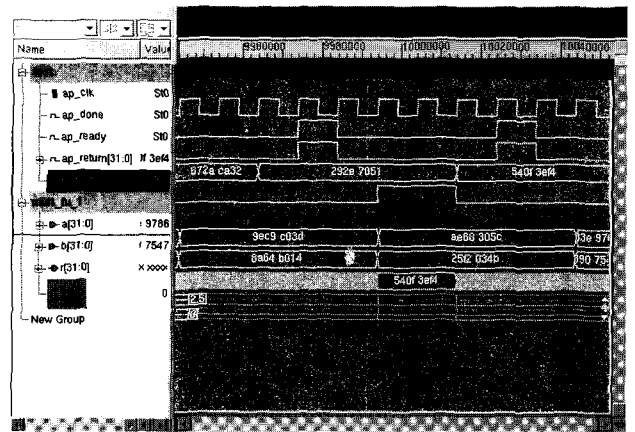


그림 7. 전원 모델의 simulation 결과
Fig. 7. The simulation result of power-model.

전력 설계 시에 전원 인식 검증 단계에서 등가 검사를 통하여 그 에러 유무를 검사할 수 있다.

이러한 전원 및 전원 셀의 모델과 전원 차단 블록에 대한 시뮬레이션 결과가 그림 7에 나타나 있다. 해당 파형에서 전원 정보인 VDD와 전원 제어 신호에 따른 각 신호들의 동작들이 나타나 있다. 전원 제어 신호인 "pwr_ctl" 신호가 Low일 시에 전원 차단 블록의 VDD가 0V를 나타내며 동시에 모든 출력이 미지 값을 출력하여 전원 차단 블록과 연결된 다른 블록 사이에 Isolation Cell이 UPF에 기술되어 있지 않을 시 미지 값이 전파되어 전체 설계의 결과가 미지 값을 얻게 되어 등가 검사가 실패하게 된다. 또한 전원 차단 모드 (Power Down Mode)에서 정상 모드(Normal Mode)로 전환 시 전원 차단 블록의 동작을 기술하여 전원 차단과 다중 전원을 사용한 저 전력 설계에 대한 검증이 가능하다.

조건 추출 알고리즘을 구현하기 위해서 내부 표현으로 사용하고 있는 CDFG를 사용하였다^[11]. CDFG는 다음 같은 표현으로 나타내어진다. G 는 주어진 CDFG의 집합이고 신호 v 에 해당하는 CDFG의 집합은 $G(v)$ 로 나타내어진다. 여기서 CDFG는 각종 신호(V)와 조건문(C), 함수 표현(E), 할당(Assignment: A)을 포함한다. V 는 설계의 신호 집합, 이러한 신호는 그 성질에 따라 V_i 는 내부 신호 i 의 집합, V_o 는 입력 포트 신호 o 의 집합, V_{pw} 는 전원 제어 신호 pw 의 집합으로 나타낸다.

P 는 CDFG의 제어 흐름 path (p)의 집합이다, 여기서 제어 흐름 path p , $p \in P$ 는 신호 v , $v \in V$ 의 순서이다. A 는 CDFG에 해당하는 프로세스안의 모든 할당의 집합이다. C 는 CDFG 내부의 조건의 집합이고, 신호 v

에 해당하는 조건은 $C(v)$ 로 표현된다. $E(c)$ 는 c 조건에
 술어부에 해당되어 되돌려지는 함수 표현이고 $Var(E)$
 는 함수 표현 E 에 포함된 신호의 집합이다. 이러한 정
 의를 통하여 CDFG내부에서 조건을 가지고 있지 않은
 신호의 함수 표현은 식 (1)로 나타내어지고,

$$u = A(v) \quad (v, u \in V) \quad (1)$$

조건을 가지는 신호에 대한 표현은 식 (2)와 같이 표현
 될 수 있다.

$$\text{if } (C(v)) \ u = E_v(c) \quad (v, u \in V) \quad (2)$$

상기에 정의를 바탕으로 그림 8에서 제안된 조건 랜
 덤 벡터 알고리즘을 설명하고 있다. 본 알고리즘에 의
 해 찾아진 제약 조건 C (Constraint)의 집합은 C_c 로 표
 현한다.

이상에서 설명한 알고리즘에 의해 각각의 전원 블록
 을 제어하는 신호를 발생시키기 위한 조건을 추출하는
 것이 가능하다. 이러한 조건의 랜덤 벡터 생성기의 조
 건 성분(Constraint)로 추가 되어 효율적인 랜덤 벡터
 입력을 발생하는 것을 가능하게 하여 준다. 다음 그림
 9는 추출된 조건 C_{C1} 과 C_{C2} , C_{C3} 을 랜덤 벡터 발생기에
 추가된 예제 파일이다. a와 b는 데이터 중심의 프로그
 램을 위한 변수 이고 sel은 제어를 위한 변수로 제어 신

호의 입력으로 사용된다. 각 조건의 랜덤 변수를 균일
 하게 생성하기 위해 “dist1” 변수가 추가 적으로 사용되
 였고, “dist1” 변수의 값에 따라 C_{C1} 과 C_{C2} , C_{C3} 조건을
 만족하는 “sel”값이 랜덤하게 생성되게 된다.
 “constraint order”는 변수 생성 시 그 분포에 대한 의
 존성을 나타내고 있다. 각 조건이 다른 개수를 포함하
 여 다른 확률을 가지고 있어도 “dist1”에 대한 랜덤 확
 률을 우선적으로 수행하여 적절한 랜덤 벡터를 생성하도
 록 해 준다.

이러한 랜덤 벡터의 조건 들이 어떠한 효과가 있는
 지 살펴보도록 하겠다. 가능한 입력의 모든 경우를 r
 회, 주어진 조건 C_{C1} 이 만족하는 경우를 $n1$ 회, C_{C2} 를
 만족하는 경우를 $n2$ 회, C_{C3} 이 만족하는 경우를 $n3$ 회,
 각각의 조건에 대한 확률을 rP_{n1} 과 rP_{n2} , rP_{n3} 라고 할
 때 필요한 벡터가 생성될 확률은 다음의 식(3)과 같이
 나타낼 수 있다.

$$P(n) = rP_{n1} * rP_{n2} * rP_{n3} \quad (3)$$

이러한 확률은 벡터의 분포가 균일하다는 가정 하에 매
 번 같은 입력의 경우의 수 n 회를 가질 때를 의미한다.
 일반 랜덤 벡터는 그 분포가 균일 하지 않기 때문에 실
 제로는 같은 경우가 두 번 이상 발생할 가능성이 있어
 모든 조건에 해당되는 입력을 얻기 위해서는 그이상의
 랜덤 벡터가 필요하게 된다. 벡터의 분포가 균일하여
 벡터 생성 시 각각의 값이 1회씩만 사용되는 경우에는,

```

Input: CDFG G, 전원 제어 신호  $V_{pw}$ 
Output:  $C_c$  //  $V_{pw}$ 신호를 만족하는  $n$ 개의 Constraint들
SearchConst(신호  $V_{pw}$ , CDFG G)
for ( $G(V_{pw}) \in G$ ) //각 신호  $pw$ 를 가지는 모든 G
  for ( $p \in P$ ) //  $pw$ 의 G내부의 모든 path
    if ( $A(V_{pw}) = Constant$ ) //  $pw$ 의 할당이 상수
       $C_c = C_c + Null$ ;
    else //  $A(V_{pw})$ 이 다른 신호의 할당이면
      SearchConst(Var( $A(V_{pw})$ ), G(Var( $A(V_{pw})$ )));
      // Var( $A(V_{pw})$ )가 Null인 경우 G(Var( $A(V_{pw})$ ))
      // 가 존재하지 않아 return된다.
  for ( $C(V_{pw}) \in C$ ) //주어진  $pw$ 의 모든 조건 C
    if (Var( $C_{pw}$ )  $\in V_c$ ) //  $C_{pw}$ 의 모든 변수가
      // 포트 신호 o 내부에 존재하면
       $C_c = C(V_o) + C_c$ ;  $n=n+1$ ;
    else SearchConst(Var( $c$ ), G(Var( $c$ )))
return 0;
    
```

그림 8. 조건 랜덤 벡터 생성 알고리즘
 Fig. 8. The constrained random vector generation algorithm.

```

class random_data
  rand int a;
  rand int b;
  rand char sel;
  rand int dist1;
  constraint dist_dist1 {
    dist1 inside {0,1,2};
  }
  constraint range {
    if (dist1==0)      sel =  $C_{C1}$ ;
    else if (dist1==1) sel =  $C_{C2}$ ;
    else                sel =  $C_{C3}$ ;
  }
  constraint order { solve dist1 before sel; }
endclass
    
```

그림 9. 조건 랜덤 벡터 예제
 Fig. 9. The constrained random vector example.

매회 벡터의 생성 시 이전의 벡터를 제외하여 다음 식 (4)와 같이 나타낼 수 있다.

$$P(n) = rP_{n1} * r-1P_{n2} * r-2P_{n3} \quad (4)$$

결과 적으로 식 (3)의 경우는 최대 $r^3/(n1*n2*n3)$ 의 횟수만큼의 벡터가 필요하게 되고, 식 (4)의 경우에는 $(r*(r-1)*(r-2))/(n1*n2*n3)$ 만큼의 벡터가 필요하게 된다. 그러나 랜덤 벡터에 조건을 추가하여 각 조건에 해당하는 경우에 수와 무관하게 각각의 확률을 $1/(조건\의\ 개수=3)$ 로 증가시켜 랜덤 벡터의 개수를 줄일 수 있다.

V. 실험 결과

전원 인식 검증을 포함한 통합 설계 및 검증 플랫폼은 기존의 상위 수준 합성 툴인 xPilot^[2]에 추가 및 구현 되었다. RTL 생성을 위해 사용된 내부 표현인 CDFG를 기본 입력으로 사용하였고, 추가로 UPF와 조건 추출 알고리즘을 사용하고 있다. 이러한 입력으로 가지고 테스트 벤치와 랜덤 벡터, 랜덤 벡터의 조건, Coverage 모니터 등을 생성함으로써 RTL 생성에서 검증에 이르는 일련의 설계 과정을 자동적으로 수행한다. 이러한 플랫폼을 검증하기 위하여 다양한 예제들을 가지고 실험을 수행하였다. 우선 기본적인 검증 자동화 플랫폼을 위하여 신호 및 영상 처리를 위한 데이터 중심의 상위 수준 프로그램을 이용하였다.

표 1은 상위 수준 합성의 결과물을 검증 자동화를 통하여 생성된 검증 플랫폼에서 수행된 결과로 각각의 Coverage 결과를 기존의 랜덤 벡터 방법^[6]을 사용한

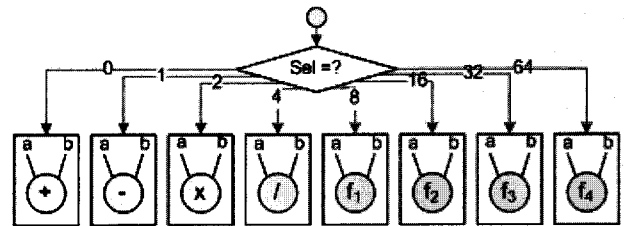
표 1. 설계의 Coverage 결과
Table 1. Coverage result of designs.

Design	Coverage (covered/total)				
	Branch	Condition	FSM Tr	Line	Reg bits
air	666/666	39/39	7/12	891/891	3663/4075
chem	407/407	19/19	6/10	521/521	3754/4004
dir	266/266	42/42	6/10	381/381	1390/1579
feig_dct	759/759	3/3	6/10	801/801	6321/6511
honda	235/235	9/9	15/28	338/338	1134/1279
lee	113/113	3/3	5/8	156/156	698/711
mcm	99/99	3/3	4/6	119/119	650/692
pr	44/44	3/3	3/4	55/55	290/293
u5ml	834/834	340/340	58/114	1244/1244	3658/3933
wang	49/49	3/3	4/6	61/61	200/203
total	100%	100%	61.4%	100%	94.0%

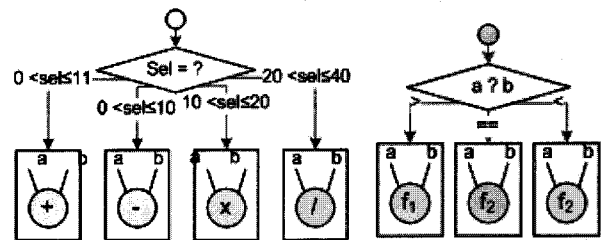
것과 비교하여 나타나고 있다. 표에 나타나있는 Coverage를 만족하며 RTL과 C 코드의 결과가 같은 값을 나타내었다. FSM Transition (Tr) Coverage와 Reg bits Coverage를 제외한 Coverage 들은 100%를 만족하고 있다. 그 외의 Coverage 들은 61.4%와 94%를 나타내고 있다. 이 두 Coverage가 100%를 만족하지 못하는 이유는 데이터 중심의 프로그램에서 RTL을 생성하는 과정에서 결정된 Finite State Machine (FSM)이 고정된 흐름도를 가지고 있어 도달하기 불가능한 FSM과 레지스터 값을 가지고 있기 때문이다.

저 전력 설계에 대해서는 전원 인식 검증 방법이 사용되었고 이러한 설계를 실험하기 위하여 제어 및 데이터 모두를 가지는 상위 수준 프로그램을 사용하였다. 본 실험에서 사용된 CDFG 예제를 그림 9에 나타내고 있다. 제어 흐름도를 가지는 프로그램에서 주로 사용하는 비트 제어 신호(그림 10(a)), 범위 구분 제어(그림 10(b)), 입력 값 비교(그림 10(c))와 같은 예제를 이용하였다.

비트 제어 신호 예제의 경우 8bits 입력 제어 신호의 각각의 비트에 함수를 적용하여 8개의 함수를 가지도록 하였다. 범위 구분 제어 예제의 경우에는 입력 제어 신호의 범위에 따라 4개의 함수가 수행하는 프로그램이며 입력 값 비교 예제의 경우는 두 개의 입력을 비교하여 두 개의 값이 같은 값과 적은 값, 큰 값에 따라 함수



(a) bitset 예제
(a) bitset example



(b) range 예제
(b) range example
(c) comp 예제
(c) comp example

그림 10. 실험을 위해 사용된 CDFG 예제들
Fig. 10. CDFG examples for experiments.

표 2. 기존 방법과 제안된 랜덤 벡터의 개수와 Coverage 비교

Table 2. Comparison of the vector number and coverage between the existing and proposed method.

Design	General Random			Constrained Random		
	Line	fsm tr	vector	Line	fsm tr	vector #
bitsel	191/191	34/50	581	191/191	34/50	20 (29x)
range	215/215	39/62	92	215/215	39/62	16 (5.75x)
comp	71/79	7/14	8332	79/79	10/14	6(<<1000x)

를 수행하고 있다. 이러한 CDFG를 가지는 상위 수준언어를 상위 수준 합성과 검증 자동화를 거쳐 RTL과 UPF, 조건 랜덤 벡터 생성기, 자동 검사기능을 포함한 검증 플랫폼을 생성하고 그 결과를 일반 로직 시뮬레이터 (Synopsys VCS)를 이용하여 실행하였다.

표 2는 상위 수준 합성을 통하여 저 전력 설계가 이루어진 설계에 대한 검증 결과인 벡터의 개수와 Coverage를 비교한 결과 이다. 일반 랜덤 입력과 조건 랜덤 입력을 사용한 경우를 비교하고 있다. 전원 인식 검증 플랫폼을 통해 생성된 입력을 사용한 경우 5.75배 이상 더 적은 입력을 가지고 같은 수준의 Coverage 결과를 얻을 수 있었다. 예제 comp의 경우에는 조건 랜덤 벡터를 사용하는 경우에 1000배 이상의 벡터의 수를 감소시킬 수 있었다. 이는 4장 3절에서 기술한 것과 같이 적은 확률을 가지는 입력 조건을 가지는 경우 일반 랜덤 벡터에서 생성될 가능성이 매우 적은 데서 기인하고 있다. 이 경우 매우 많은 벡터를 사용하여도 Coverage가 90%, 71%만을 만족하는 것을 알 수 있다.

VI. 결 론

본 논문에서는 설계 및 검증 자동화와 저 전력 설계를 위한 검증 방법론 등을 포함하는 통합 설계 방법론 및 검증 플랫폼을 제안 및 개발하였다. VLSI의 저 전력 설계의 검증을 위하여 스펙을 근거로 하여 해당하는 입력을 발견적 방법에 따라 작성하게 된다. 이러한 방법의 각 모드에 대한 누락이나 실수를 발생하게 된다. 본 논문에서는 전원 및 전원 셀 모델링 기법을 이용하여 전원 인식 검증을 가능하게 하였고 각 전원 모드에 대한 조건을 자동적으로 생성하여 효과적인 벡터를 생성하였다. 제안된 검증 자동화 플랫폼을 이용하여 상위 수준 합성에서 생성된 RTL과 UPF등의 결과물을 CDV, 테스트 벤치 자동화 등을 이용하여 설계의 검증을 효과

적으로 수행할 수 있었다. 또한 그 설계에 대한 우수한 Coverage 결과를 실험을 통하여 확인하였다.

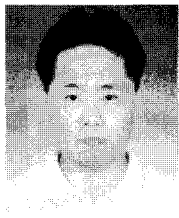
제안된 저 전력 설계의 검증을 위한 조건 랜덤 벡터를 사용하여 기존의 일반 랜덤 벡터를 사용한 경우보다 최소 5.75배 감소한 입력 개수를 사용하여도 같은 수준의 Coverage를 얻을 수 있었다. 또한 이러한 검증 자동화를 통하여 설계 기간의 단축과 더불어, 시스템 수준의 검증이 이루어져 상위 수준 합성이 가지고 있는 장점을 증대시킬 수 있다. 시스템 수준에서 다양한 설계의 검증과 가능성을 시도함으로써 좋은 성능과 저 전력 설계를 가능하게 해 주고 있다. 이러한 통합된 설계 및 검증 플랫폼은 시스템 수준의 설계에서 검증, 합성에 이르는 전 설계 흐름을 완전 자동화를 가능하게 하고 있다.

참 고 문 헌

- [1] Daniel D. Gajski, Loganath Ramachandran, "Introduction to High-Level Synthesis," *IEEE Design and Test of Computers*, vol. 11, no. 4, pp. 44-54, Oct. 1994.
- [2] J. Cong, Y. Fan, G. Han, W. Jiang, and Z. Zhang, "Platform-Based Behavior-Level and System-Level Synthesis," *Proceedings of IEEE International SOC Conference*, pp. 199-202, Austin, Texas, USA, Sept. 2006.
- [3] 김규홍, "통합된 검증 방법론" *전자공학회지*, 제30권 SD편, 제9호, 70-85쪽, 2003년 9월
- [4] 김남섭, 조원경, "임베디드 프로세서와 재구성 가능한 구조를 이용한 SoC 테스트와 검증의 통합," *대한전자공학회논문지*, 제43권 SD편 제7호, pp. 38-49 2006년. 7월
- [5] D. Chen, J. Cong, and J. Xu, "Optimal Module and Voltage Assignment for Low-Power," *Proceedings of the Asia South Pacific Design Automation Conference*, pp. 51-59, January 2005.
- [6] D. Chen, J. Cong, Y. Fan and J. Xu, "Optimality Study of Resource Binding with Multi-Vdds," *Proceedings of the Design Automation Conference*, pp. 580-585, San Francisco, CA, USA, July 2006.
- [7] 정우성, 신현철 "에너지 소모 최소화를 위한 다중전압 스케줄링 기법," *전자공학회논문지*, 제46권 SD편, 제9호, 49-57쪽, 2009년 9월
- [8] A. Maalej and P.Y. Martinez, "Formal proof between two designs having different level of abstraction," *3rd International Conference on Design and Technology of Integrated Systems in Nanoscale Era*, pp. 1-4, Mar. 2008.

- [9] P. Urard, A. Maalej, R. Guizzetti, N. Chawla, and V. Krishnaswamy, "Leveraging sequential equivalence checking to enable system-level to RTL flows", *Proceedings of Design Automation Conference*, pp. 816 - 821, San Francisco, CA, USA, Jun. 2008.
- [10] Malay K. Ganai, Akira Mukaiyama, Aarti Gupta and Kazutoshi Wakabayashi "Synthesizing "Verification Aware" Models: Why and How?," *20th International Conference on VLSI Design*, 2007 pp. 50 - 56 , Jan, 2007.
- [11] Shireesh Verma, I.G Harris and Kiran Ramineni, "Automatic Generation of Functional Coverage Models from Behavioral Verilog Descriptions," *Proceedings of Design, Automation & Test in Europe Conference & Exhibition*, pp. 1-6 April 2007.
- [12] P. Mishra and N. Dutt, "Functional coverage driven test generation for validation of pipelined processors," *Proceedings of Design, Automation and Test in Europe*, pp. 678-683 Vol. 2 Jun. 2006.
- [13] Unified Power Format (UPF) v1.0 Standard, <http://www.unifiedpowerformat.com/>
- [14] Janick Bergeron, Eduard Cerny, Alan Hunter, and Andy Nightingale, "Verification Methodology Manual for SystemVerilog," *Springer*, pp. 1-288, 2005.
- [15] Kelvin Ng, "Challenges in using system-level models for RTL verification," *Proceedings of Design Automation Conference*, pp 812-815, San Francisco, CA, USA, Jun. 2008.
- [16] Y. Lahbib, O. Missaoui, M. Heckel, D. Lahbib, B. Mohamed-Yosri and R. Tourki, "Verification flow optimization using an automatic coverage driven testing policy" *International Conference on Design and Test of Integrated Systems in Nanoscale Technology*, pp. 94 - 99, Sep. 2006.

 저 자 소 개



김 정 훈(정회원)

1993년 인하대학교 전자재료
공학과 학사

2003년 고려대학교 전기전자
공학과 석사

2008년 고려대학교 전자컴퓨터
공학과 박사

1993년~2005년 삼성전자 책임연구원

2005년~2006년 뮤텔테크놀로지 수석연구원

2007년~2008년 매그너칩반도체(주) 수석연구원

2008년~현재 UCLA 연구 조교수

<주관심분야 : VLSI 설계 및 검증 자동화, 서보
제어 및 아키텍처, SoC 설계 및 아키텍처>