

전송 수준 시스템 명세를 이용한 공유 버스 구조의 RTL 모델 자동 생성 기법

*방관후, 김 동, 정의영
연세대학교 전기전자공학과

e-mail : *lamar49@yonsei.ac.kr, charikim@yonsei.ac.kr, eychung@yonsei.ac.kr*

An Automatic Generation of Shared Bus RTL Model from Transaction Level System Specification

*Kwan-Hu Bang, Dong Kim, Eui-Young Chung
School of Electrical and Electronic Engineering
Yonsei University

Abstract

In recent complex System-on-Chip(SoC) design, high level system modeling techniques such as Transaction-Level Modeling(TLM) are widely used for fast simulation and verification. The advantage of TLM methods is the fast simulation speed, hence it is appropriate for the exploration of a large design space. However, it is commonly observed that the TLM is not automatically translated into the RTL specification. In other words, the advantage of TLM is depreciated by the manual translation due to the time for the coding and verification in RTL design stage. This paper proposes an automatic generation of synthesizable shared bus RTL model, from the system specification simulated and verified with TLM design. The technique enables fast system design and synthesis only with high level simulation and verification. The experimental result shows that the simulation time of TLM is about 50 times faster than that of RTL model, and the modeling effort of TLM also reduced significantly.

I. 서론

최근의 복잡한 SoC(System-on-Chip) 설계의 빠른 시뮬레이션 및 검증을 위해 전송 수준 모델링(Transaction-Level Modeling)과 같은 상위 수준 모델링 기법이 사용되고 있다[1][2]. 이러한 전송 수준의 설계를 실제 칩으로 구현하기 위한 합성 기법이 연구되고 있으나 모델간의 자동 변환이 이루어지지 못함으로써 정확도 측면에서 RTL(Register Transfer Level) 모델과의 차이가 발생할 수 있는 단점을 가지고 있다.[3] 이러한 차이점은 전송 수준 모델의 구현과 별도로 RTL 모델의 구현을 필요로 하게 된다.

본 논문은 SoC 설계에서 널리 사용되는 공유 버스 구조에 대하여 전송 수준 모델로 검증된 시스템의 명세를 이용하여 XML(Extensible Markup Language)로 표현된 공유 버스 모델로부터 합성 가능한 RTL 모델을 자동으로 생성하는 기법을 제안한다. 이러한 방법을 통하여 상위 수준의 시뮬레이션 및 검증을 통한 빠른 시스템 설계 및 하위 수준 합성이 가능하게 된다.

II. 본론

2.1 RTL 모델의 XML 표현

XML은 텍스트 기반의 자료의 유연한 기술을 가능

하게 해주는 메타 문서 양식이다[4]. XML의 이러한 특징은 RTL 모델 내부의 세부적인 동작에 대한 XML 태그를 이용한 체계적인 기술이 가능하게 해준다. 본 논문에서는 공유 버스 구조의 RTL 모델을 XML 문서로 기술하였다. 이 XML 문서에 전송 수준 모델의 검증을 통해 얻은 시스템 명세를 적용하면 합성 가능한 공유 버스 구조 RTL 모델을 얻게 된다.

2.2 전체 시스템 구조 및 동작

본 논문에서 제안하는 공유 버스 구조의 RTL 모델 자동 생성을 위한 전체 시스템 구조 및 동작은 그림 1과 같다. 먼저 최초의 시스템 명세를 기술한 후 이를 전송 수준 모델로 구현한다. 구현된 전송 수준 모델을 이용하여 빠른 시뮬레이션 및 검증을 함으로써 목표하는 시스템의 명세는 반복적으로 수정될 수 있다. 이런 방법으로 최종적으로 도출된 시스템 명세를 자동 공유 버스 구조 생성기의 입력으로 넣어주면 XML 문서로 기술된 공유 버스 구조를 분석하여 시스템 명세에 적합한 합성 가능한 RTL 모델을 출력하게 된다.

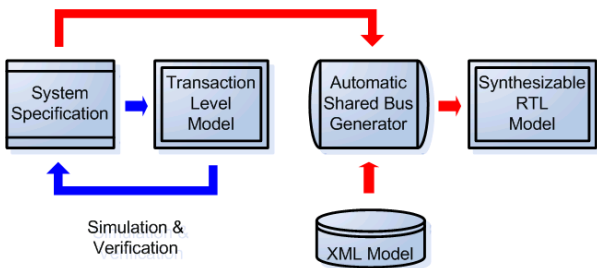


그림 1. 전체 시스템 구조

III. 구현

공유 버스 구조에 사용되는 다양한 프로토콜 중에서 본 논문에서는 AMBA 2.0 AHB 프로토콜을 이용하여 공유 버스 구조의 XML 문서를 작성하였다[5]. 또한 XML 문서를 위한 파서는 Java 언어에서 제공되는 API 클래스들을 토대로 구현하였다[6]. 마지막으로 전송 수준 모델의 시뮬레이션 및 검증을 위한 환경으로 Realview SoC Designer를 사용하였다[7].

전송 수준 모델을 통해 검증된 시스템 명세를 사용하여 얻어낸 RTL 모델을 시뮬레이션 한 결과 파형은 전송 수준 모델에서의 결과 파형과 사이클 단위로 일치하는 것을 확인할 수 있었다. 그리고, 전송 수준 모델을 시뮬레이션 하는데 걸리는 시간은 RTL 모델을 시뮬레이션 하는데 걸리는 시간에 비해 크게 단축되었

다. 표 1은 두 가지 모델의 시뮬레이션 속도를 보여준다. 전송 수준 모델을 사용하면 시뮬레이션 속도 면에서 50배 이상 빠르게 시뮬레이션을 수행할 수 있었다. 표 2에서는 두 가지 모델을 구현하는데 걸린 시간을 나타내었다. 전송 수준 모델을 구현하는데 걸리는 시간이 RTL 모델을 구현하는데 걸리는 시간보다 훨씬 적은 것을 알 수 있다. 마지막으로, 결과물인 RTL 모델은 합성 툴을 사용하여 쉽게 합성할 수 있었다.

표 1. 전송 수준 모델과 RTL 모델의 시뮬레이션 속도

Modeling	Simulation Speed
TLM	1032.824 Kcycles/sec
RTL	19.991 Kcycles/sec

표 2. 전송 수준 모델과 RTL 모델 구현에 걸린 시간

Modeling	Modeling Effort
TLM	2 Weeks
RTL	8 Weeks

IV. 결론 및 향후 연구 방향

본 논문에서 제안된 방법을 사용하면 전송 수준 모델을 이용하여 빠른 동작의 검증이 가능하다. 이러한 전송 수준 모델의 장점을 이용하여 시스템 명세를 결정하면, 이 시스템 명세에 맞는 합성 가능한 RTL 모델을 곧바로 자동 생성해줌으로써 전체 설계 시간을 상당히 단축시킬 수 있다. 이러한 방법을 통해 속도와 정확성을 동시에 만족시키는 설계가 가능하게 된다.

참고문헌

- [1] M. Caldari, M. Conti, M. Coppola, S. Curaba, L. Pieralisi, and C. Turchetti, Transaction-Level Models for AMBA Bus Architecture Using SystemC 2.0, DATE '03, 2003
- [2] Thorsten Grötter, Stan Liao, Grant Martin, and Stuart Swan, System Design with SystemC, Kluwer Academic Publishers, 2002.
- [3] G. Schirner and R. Dömer, Quantitative Analysis of Transaction Level Models for the AMBA Bus, DATE '06, 2006.
- [4] www.w3.org/XML/
- [5] www.arm.com, AMBA Specification Rev 2.0, ARM, 1999.
- [6] java.sun.com
- [7] www.arm.com/products/DevTools/MaxSim.html