Verilog4VIS-EC: VIS의 동치성 검사를 위한 Verilog의 정제된 포맷

김의섭 ○, 정세진, 김재엽, 유준범, 장천현

건국대학교 컴퓨터공학부

{atang34, jsjj0728, ksjy1990, jbyoo, chchang}@konkuk.ac.kr

Verilog4VIS-EC: A Manipulated Verilog Format for VIS Equivalence Checking

Eui-Sub Kim○, Sejin Jung, Jaeyeob Kim, Junbeom Yoo, Chun-Hyon Chang

Division of Computer Science and Engineering, Konkuk University

요 약

최근 원자력발전소의 계측제어 시스템을 PLC에서 FPGA로 대체하는 연구가 진행되면서, FPGA 개발에 사용된 합성도구와 같은 COTS 도구의 정확성 및 안전성 증명이 필요해졌다. 이를 위해 우리는 VIS의 동치성 검사를 통해 이를 증명하고자 한다. 하지만 VIS가 Verilog의 모든 표현 형식을 지원하지 않는 문제가 있었고, 이를 위해 본 논문은 VIS의 동치성 검사를 지원하는 Verilog의 정제된 형태인 Verilog4VIS-EC를 제안한다. 우리는 본 논문에서 제시하는 Verilog4VIS-EC와 VIS의 동치성 검사를 이용해 FPGA 개발에 사용된 합성 도구의 정확성을 증명할 수 있을 것으로 기대하고 있다.

1. 서 론

원자력발전소의 디지털 계측제어 시스템(Instrumentation and Controller: I&C)은 높은 수준의 안전성과 신뢰성이 요구되는 안전필수시스템 (Safety-Critical System)으로서 제어기에 사용되는 소프트웨어 역시 높은 수준의 품질이 요구된다. 이를 보장하기 위해서는 소프트웨어의 자체적인 검증뿐 아니라 안전성과 신뢰성이 충분히 검증된 COTS 소프트웨어를 이용하여 개발해야 한다[1][2].

최근 원자력발전소의 계측제어 시스템을 PLC (Programmable Logic Controller)기반에서 FPGA(Field-programmable gate array)기반으로 대체하고자 하는 연구가 활발히 진행 중이다[3][4]. 일반적으로 FPGA를 개발하기 위해서는 합성(synthesis) 도구와 배선 및 배치(P&R: Place and Route) 도구 같은 다양한 COTS 소프트웨어가 사용 되는데, 원자력발전소 분야에 사용하기 위해서는 해당 COTS 소프트웨어들에 대한 인증 프로세스(COTS Dedication)를 진행하여 신뢰성 및 안전성을 충분히 보장할 필요가 있다.

특히, 우리는 합성도구의 기능적 정확성을 검증하고자 합성 전 프로그램인 Verilog와 합성 후 프로그램인 EDIF간의 일치성(Correctness) 증명을 통해 정확성을 간접적으로 증명하고자 계획하고 있다. 이를 위해 정형검증 도구 중 하나인 VIS를 사용할 계획이고, 특히 VIS의 기능 중 동치성 검사(Equivalence Checking)를 이용해 검증을 수행할 계획이다. 하지만 VIS가 모든 Verilog의 표현을 지원하고 있지 않기 때문에 Verilog의 적절한 정제가 필요한 상황이다.

이를 위해 본 논문은 VIS의 동치성 검사를 지원하는 Verilog의 정제된 포맷인 Verilog4VIS-EC를 제안한다. Verilog4VIS-EC는 VIS의 동치성 검사를 위해 제안한 포맷으로서 기본적으로 Verilog 문법을 따르며 VIS의 동치성 검사를 위한 제약조건 및 가정(Assumption)을 포함하고 있다. 본 논문은 해당 제약조건 및 가정을 제시하고 있다. 우리는 Verilog4VIS-EC와 VIS의 동치성 검사를 이용해 합성 전 프로그램인 Verilog와 합성 후 프로그램인 EDIF간의 일치성을 증명할 계획이고, 이를 통해 합성도구의 정확성을 간접적으로 증명할 수 있을 것으로 기대하고 있다.

본 논문의 구성은 다음과 같다. 2장에서는 배경지식을 설명한다. 3장 에서는 VIS의 동치성 검사를 위한 Verilog의 정제된 포멧인 Verilog4VIS-EC를 위한 제약사항 및 가정에 대해 설명하고 마지막으로 4장에서 결론 및 향후 연구를 설명한다.

2. 배경지식

VIS[5]는 정형검증, 논리합성, 클럭 기반의 시뮬레이션, CTL(Computational Tree Logic) 모델 체킹, 순차적 동시성 검증 (sequential equivalence checking) 기능, 조합적 동치성 검증(combinational equivalence checking) 기능 등을 제공하는 통합 개발 및 검증 도구이다. VIS는 Verilog를 입력언어로 받아들이며, 내부적으로 vl2mv[6] 라는 변환기를 이용하여 VIS 내부 포맷인 BLIF-MV[7] 파일로 변환한 후 기능을 수행한다.

동치성 검사는 두 프로그램이 동일한 입력에 대해 동일한 출력을 내보내는지 확인하는 정형 기법으로 모든 입력 조합을 이용하여 출력을 확인한다. 내부적으로 회로를 이진 결정 다이어그램(Binary Decision Diagram)을 생성하여 수행을 하며, 이 이진 다이어그램의 크기에 따라 검증 속도 차이 나게 된다

3. Verilog4VIS-EC를 위한 제약사항

Verilog4VIS-EC는 VIS의 동치성 검사를 위한 Verilog의 정제된 포맷이다. VIS는 대부분의 Verilog 표현을 지원하고 있지만 동치성 검사를 위해서는 특정 부분을 수정할 필요가 있다. 정상적인 동치성 검사를 진행하기 위한 제약사항이 필요하고 해당 제약사항을 바탕으로 Verilog를 작성할 필요가 있다. 아래 내용은 Verilog4VIS-EC를 위한 제약사항 및 가정이다.

(1) 클럭 관련 제약사항
- always 구문 안에는 오직 하나의 클럭(clk)만 사용
- negedge에 동기화된 클럭의 사용 불가
- always 구문 안에 딜레이와 관련된 구문 사용 불가

(2) 비결정성 관련 제약사항
- non-blocking 구문 사용 불가
- register 변수 사용 시 0으로 초기화

(3) 구문 오류 관련 제약사항
- integer형 변수의 register 사용 불가
- parameter 변수 사용 시 bits 의 크기 지정 불가

3.1. (1) 클럭 관련 제약사항

VIS는 기본적으로 하나의 글로벌 클럭에 의해 동작하는 것을 전제로 하고 있는 시스템이다. 따라서 이런 개념에 어긋난 구문은 변경해 주어야 한다. 클럭 관련 제약사항은 다음과 같다.

• 먼저 always 구문 안에는 오직 하나의 클럭(clk)만을 사용해야 한다. VIS는 Verilog 모델을 하나의 내부적인 클럭을 가지는 이산시간 유한 상태 머신(discrete-time finite state machine)으로 여긴다. 따라서 특정 의도를 가진 하나 이상의 추가적인 클럭 사용을 VIS는 허용하지 않는다. 또한 always 구문에 사용된 변수들을 모두 클럭으로 취급하기 때문에 always 구문에서 클럭 이외의 변수를 사용할 경우, 추가적인 클럭을 사용하는 것으로 인식해 구문 오류 메시지를 내보낸다. 따라서 동치성 검사를 위해 always 구문에는 하나의 클럭만을 사용하여 Verilog를 작성해야 한다.

• Verilog 작성시 negedge를 이용한 표현은 허용되지 않는다. VIS의 내부 변환기인 vl2mv는 클럭과 관련된 모든 구문은 제거하기 때문에 posedge 클럭이나 negedge 클럭이나 모두 동일하게 변환 된다. 반면 합성도구는 posedge와 negedge의 기능을 나누어 변환하기 때문에 만약 negedge를 이용한 클럭을 사용한다면 동치성 검사 시 일치하지 않는다는 결과를 얻게 될 것이다. 이를 피하기 위해 Verilog 작성시 negedge를 이용한 표현은 동치성 검사를 위해 사용하면 안 된다.

• 추가적으로 always 구문 안에 인위적인 시간 딜레이의 사용은 불가하다. VIS는 always 구문 안의 모든 동작을 하나의 클럭에 동기화 되어 동작하는 것으로 취급하고 있다. 따라서 인위적인 시간 딜레이의 사용은 내부적인 글로벌 클럭과 모순되기 때문에 사용하면 안 된다.

3.2. (2) 비결정성 관련 제약사항

Verilog는 비결정성을 가질 수 있는 언어이다. VIS 역시 이런 비결정성을 지원하지만 동치성 검사에서는 이런 비결정성이 문제가 된다. 비결정성이란 값을 결정할 수 없다는 뜻인데, 다시 말하면 비교할 값이 없다는 뜻이다. 비교할 값이 정해지지 않았기 때문에 결국 동치성 검사를 할 수 없는 것은 당연하다. 따라서 사용자는 이런 비결정성을 피해 Verilog를 작성할 필요가 있다. 비결정성에 관련된 제약사항은 다음과 같다.

• Non-blocking 구문의 사용은 허용되지 않는다. Verilog는 기본적으로 register에 값을 할당할 때 blocking('=')과 non-blocking('<=')을 모두 사용할 수 있다. Blocking 구문은 우측 변수가 좌측 변수로 대입되기 전에는 다음 문장이 실행되지 않는 반면, non-blocking 구문은 우측 변수의 값들을 다음 클럭 시 동시에 모두 좌측 변수로 각각 대입한다. 하지만 non-blocking 구문을 사용할 경우 의도하지 않는 동작이 이루어질 수 있다. 예를 들면 두 개의 프로시저에 의해 동시에 동일한 register에 non-blocking으로 변수가 할당될 경우 어떤 프로시저의 값이 최종적으로 할당될지 결정할 수 없는 비결정(non-deterministic)상황이 벌어진다. VIS는 이런 비결정성을 가진 구문 자체는 지원하지만, 동치성 검사 때 해당 구문은 문제가 된다. 값이 정해지지 않았기에 비교를 할 수 있는 값 존재하게 되어 동치성 검사를 수행할 수 없게 된다. VIS는 비결정성 구문이 사용되었을 경우 값이 결정되지 않아 비교를 할 수 없다는 에러 메세지를 내보낸다. 따라서 동치성 검사를 위해서는 non-blocking 구문을 피해 Verilog를 작성해야 한다.

• Register 변수를 사용할 경우 반드시 0으로 초기화 해주어야 한다. VIS는 동치성 검사를 시작할 때 모든 register 변수에 대해 초기값을 요구하고 있다. 초기값이 지정되지 않는 경우 값이 결정되지 않았다는 에러 메세지를 내보내고 있다. 따라서 초기값을 필수적으로 정해주어야 하고, 초기값으로는 어떤 값이든 상관 없다. 하지만 상용 합성도구의 경우 모든 register에

일괄적으로 0을 초기화 하기 때문에, 동치성 검사를 위해 VIS에도 합성도구와 같은 초기값인 0을 설정해 주어야 정확한 동치성 결과값을 얻을 수 있다.

3.3. (3) 구문 오류 관련 제약사항

• Integer형 변수를 register의 용도로 사용할 수 없다. 일반적으로 Verilog 에서 integer형은 register로 사용된다. 차이점은 register는 크기를 갖는 부호 없는 정수이고 integer형은 부호가 있는 32bits 크기의 정수를 의미한다. 따라서 합성도구에서는 integer형의 변수를 사용해도 무방하다. 하지만 VIS는 Integer를 register로 인식하지 않고 있기 때문에 register를 표현하기 위한 용도로 integer를 사용해서는 안 된다.

• Parameter 변수 사용 시 bits의 사이즈를 지정해 주면 안 된다. VIS는 Parameter 변수를 사용할 때 bits의 사이즈를 명시하면 구문 오류 메시지를 출력하고 있다. 따라서 bits의 사이즈를 명시하지 않고 Verilog를 작성해야 한다.

하지만 VIS는 parameter 변수에 대해 특정상황에서 상용 합성도구와 다른 해석을 하는 위험 요소가 있다. VIS는 parameter로 선언된 변수 두 개를 연산할 때 오버플로우가 생긴 값을 버리고 있다. 예를 들면, VIS는 두 변수를 이용해 덧셈을 수행할 경우, 덧셈 결과에 필요한 bits 사이즈를 사용된 두 변수 중 큰 수에 따라 할당한 다음 덧셈한 값이 할당된 bits 보다 크다면 상위 bit를 버리고 있다. 다시 말해, parameter a,b 가 각각 3으로 할당되어 있을 경우 (a = 3; b = 3;) 두 변수의 덧셈 값이 3 (11) + 3 (11) = 6 (110) 이 되는 것이 아니라 2bits 이상의 값을 버림 하여 3 (11) + 3 (11) = 2 (10)로 덧셈 연산이 이루어진다. 따라서 VIS의 동치성 검사를 위해서 해당 제약사항의 준수하여 위험 요소를 피해 Verilog를 작성할 필요가 있다.

지금까지 Verilog4VIS-EC를 위한 제약사항을 살펴보았다. 본 논문에서 제시한 제약사항 이외에 추가적인 제약사항이 존재 할 수 있다. 앞으로 그런 제약사항들을 찾아 보다 정확하고 정형적으로 제약사항을 만드는 작업이 필요할 것이다.

4. 결 론

FPGA를 개발하기 위해서는 다양한 상용 도구가 사용된다. 그 중 합성도구는 복잡한 과정과 고도의 기술이 집약되어있는 도구로서 원자력발전소에 사용되기 위해서는 정확성과 신뢰성 대한 검증이 필요하다. 이를 위해 우리는 VIS의 동치성 검사를 통해 이를 간접적으로 증명하고자 한다. 하지만 VIS가 모든 Verilog의 표현을 지원하고 있지 않고 있어는 문제가 있어, 본 논문에서 VIS의 동치성 검증을 위한 Verilog의 정제된 형태인 Verilgo4VIS를 제안하였다. 앞으로 우리는 Verilgo4VIS-EC와 VIS의 동치성 검사를 이용해 합성 전 프로그램인 Verilog와 합성 후 프로그램인 EDIF간의 일치성을 증명을 통해 합성도구의 간접적인 검증을 수행할 계획이다.

사 사

본 연구는 한국원자력연구원의 “FPGA-기반 제어기 통합개발환경을 위한 핵심 소프트웨어 기술 개발” 사업과 “원자력 계측제어 계통 안전 적합성 평가체계” 사업의 지원으로 연구한 결과입니다.

참 고 문 헌

[1] International Electrotechnical Commission, IEC 1226, “Nuclear Power Plants - Instrumentation And Control Systems Important For Safety - Classification,” 1994

[2] Nuclear Regulatory Commission, NuREG/CR-6421, “A Proposed Acceptance Process for Commercial Off-the-Self (COTS) Software in Reactor Applications,” 1996

[3] Junbeom Yoo, Jong-Hoon Lee and Jang-Soo Lee, “A Research on Seamless Platform Change of Reactor Protection System from PLC to FPGA,” Nuclear Engineering and Technology, Vol.45, No.4, pp.477-488, 2013.

[4] Jong Gyun Choi and Dong Young Lee, “DEVELOPMENT OF RPS TRIP LOGIC BASED ON PLD TECHNOLOGY,” Nuclear Engineering and Technology, vol. 44, no. 6, pp.697-708, 2012

[5] The VIS Group, VIS: Verification Interacting with Synthesis, http://vlsi.colorado.edu/~vis/

[6] Clarke, Edmund M and Grumberg, Orna and McMillan, Kenneth L and Zhao, Xudong, “Efficient generation of counterexamples and witnesses in symbolic model checking,” In Proc. 32nd Design Automat. Conf., pages 427-432, June 1995.

[7] Brayton, Robert King, “BLIF-MV: An interchange format for design verification and synthesis," Electronics Research Laboratory, College of Engineering, University of California, 1991.