FBDtoVerilog 변환기의 Correctness 를 검증하기 위한 자동화된 시나리오 생성기 구현

김의섭, 이동아, 유준범*
건국대학교 컴퓨터 정보통신공학과
e-mail : {atang34, ldalove, jbyoo*}@konkuk.ac.kr

The Scenario Generator for Verifying the Correctness of FBDtoVerilog Translator

Eui-Sub Kim, Dong-Ah Lee, Junbeom Yoo*
Division of Computer Science and Engineering, Konkuk University

요 약

본 논문은 FBDtoVerilog 변환기의 correctness 검증을 지원하는 시나리오 생성기에 대해 소개한다. 현재 원자력 발전소의 제어기는 PLC 를 이용하여 개발되고 있지만, 최근 FPGA 를 이용한 제어기 개발의 필요성이 증가하고 있다. 우리는 이를 지원하기 위해 PLC 개발에 사용되는 언어인 FBD 를 FPGA 에 사용되는 언어인 Verilog 로 자동 변환하는 변환기 FBDtoVerilog 를 개발 하였다. 하지만 원자력 발전소와 같은 안전 필수 시스템은 철저하고 엄격한 검증 과정이 필수 이기 때문에, 우리는 FBDtoVerilog 를 검증할 수 있는 Co-Simulation 환경을 구축하여 검증할 계획을 가지고 있다. Co-Simulation 환경을 위한 첫 번째 단계로 자동화된 시나리오 생성기를 개발 하였다. 개발된 시나리오 생성기는 도메인 특징을 반영한 시나리오를 생성할 수 있고, 무한한 개수의 시나리오를 자동으로 생성할 수 있는 장점을 가지고 있다.

1. 서론

현대 소프트웨어 개발에서 변환기의 역할은 점점 중요해 지고 있으며, 적용되는 산업 분야 또한 작은 규모의 임베디드 소프트웨어 개발에서부터 안전성이 최우선 되는 안전 필수 시스템에 이르기 까지 다양한 곳에서 사용되고 있다. 하지만 대부분 변환기의 검증은 벤더(Vendor) 내부적인 검증에 의존하고 있기 때문에 신뢰성 부분에서 문제를 가지고 있다. 만약 신뢰성이 보장되지 않는다면, 이런 변환기를 이용하여 개발한 소프트웨어 또한 신뢰성을 보장할 수 없는 것은 당연하다.

과거 변환기 검증을 위해 위해 많은 방법들이 제시되어 왔다. [1]. 하지만 소프트웨어의 규모가 점점 커지고 복잡해지면서 변환기 자체를 직접 검증하는 일은 많은 시간과 노력이 필요하게 되었다. 따라서 이런 시간과 노력을 줄이고자 변환기를 간접적으로 검증하는 방법들이 연구되어 왔다. 변환기를 간접적으로 검증할 수 있는 방법 중 하나로 Co-Simulation 을 통해 변환되기 전 프로그램과 변환 후 프로그램의 correctness 확인을 통해 검증하는 방법이 있다. Co-Simulation 을 통한 correctness 확인은 적어도 입력된 프로그램이 이와 동일한 기능을 하는 프로그램으로 변환이 되었다는 것을 보장 할 수 있게 된다.

우리는 과거 연구를 통해 FBDtoVerilog 2.0 라는 변환기를 구현하였다[2]. 현재 원자력 발전소의 RPS (Reactor Protection System)는 PLC (Programmable Logic Controller)로 구현되고 있는데, 공통 원인 고장이나 (Common Cues Failure), 새로운 소프트웨어 개발의 복잡성, 유지보수 비용의 증가, 보안성(Security) 과 같은 이유 때문에 FPGA (Filed-Programmable Gate Array)로의 플랫폼 전환이 요구되고 있다. 하지만 플랫폼이 소프트웨어 기반의 PLC 에서 하드웨어 기반의 FPGA 로 변하게 되면 여러 위험요소를 가지게 된다[3]. 예를 들면 과거 축척 된 기술이나 안전성 분석, 인증, 노하우 등을 버려야 하고, 개발자는 FPGA 를 위한 새로운 언어인 HDL (Hardware Description Language)를 배워야 한다. 따라서 이런 위험 요소를 제거하고 새로운 언어에 대한 지원을 하기 위해, 우리는 PLC 개발에 사용되는 디자인 언어인 FBD (Function Block Diagram)를 FPGA 를 개발에 사용되는 HDL 언어인 Verilog 로 자동 변환해 주는 FBDtoVerilog 2.0 을 개발하였다.

FBDtoVerilog 의 타겟은 안전 필수 시스템인 원자력 발전소의 소프트웨어이기 때문에 철저하고 엄격한 검증이 필요 하다. 따라서 우리는 FBDtoVerilog 의 검증을 위해 Co-Simulation 환경 구축을 계획하고 있다. 하지만 Co-Simulation 의 결과는 상당부분 시나리오의 개수나 커버리지에 등에 의해 결정될 수 있기 때문에, Co-Simulation 환경에서 시나리오는 중요한 부분을 차지하게 된다. 우리는 이를 지원하기 위해 자동으로 시나리오를 생성해 주는 시나리오 생성기를 구현 하였다.

본 논문에서 구현한 시나리오 생성기는 다음과 같은 장점을 가지고 있다. 무한한 수의 시나리오를 자동으로 생성 할 수 있다. 도메인의 특징을 반영한 적절한 시나리오를 생성 할 수 있다. FBD 시뮬레이션을 위한 시나리오뿐만 아니라, Verilog 시뮬레이션을 위한 ModelSim 의 test bench 를 동시에 생성해 테스터의 추가적인 수고를 줄여 준다.

2. 관련 연구

2.1 FBD (Function Block Diagram)

FBD 는 PLC (하드웨어 입출력 장치를 실사간으로 제어하기 위한 제어 시스템)의 개발을 위해 IEC-61131-1 [4] 표준에 정의된 5 가지 언어 중에 하나이다. 시스템의 행위를 입력과 출력 사이의 블록(Block)과 블록 사이의 연결을 통해 표현하는 그래픽 기반의 언어로써, 전자회로 다이어그램(electrical circuit diagram)과 유사한 형태의 그래픽컬한 다이어그램이다. FBD 는 그래픽컬한 다이어그램은 데이터의 흐름을 쉽게 나타낼 수 있다는 장점이 있어 PLC 를 프로그래밍 하는데 널리 쓰이는 언어이다.

2.3 Verilog HDL

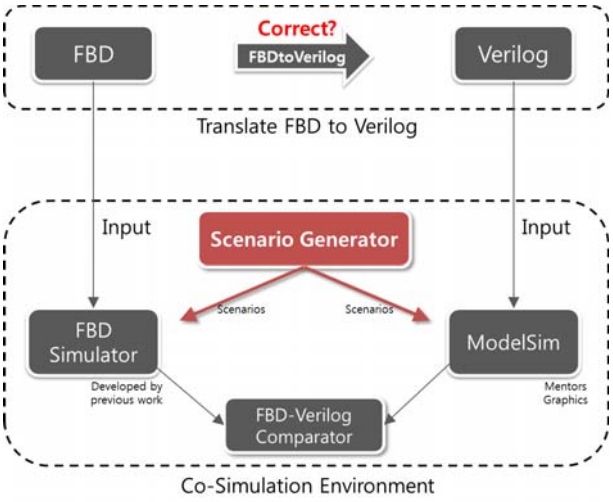
Verilog [5]는 FPGA 를 개발하기 위한 HDL 언어 중에 하나로써, 회로 설계 및 검증, 구현 등 다양한 용도로 사용되고 있다. 다양한 검증 및 분석 도구 들이 Verilog 를 입력 언어로 지원하고 있고, 대부분의 대중적인 논리합성 도구들이 Verilog 를 지원하고 있기 때문에 널리 사용되고 있는 언어 중 하나이다. 또한 다양한 Vendor 와 EDA(Electronic Design Automation) 도구들이 게이트 수준의 합성 과정을 지원해 주기 때문에, Verilog 를 이용한 개발은 FPGA 디바이스 또는 반도체 칩 제조 과정 등과는 무관하게, 회로 설계 및 디자인에만 집중하여 개발 할 수 있다는 장점이 있다.

2.4 FBDtoVerilog 2.0

우리는 과거 연구를 통해 FBDtoVerilog 2.0 라는 변환기를 구현하였다[2]. 원자력 발전소의 플랫폼 전환(PLC→FPGA)을 지원하기 위해 개발한 변환기로써 FBD 를 Verilog 로 변환해 주는 변환기 이다. FBD 의 block 과 Verilog 의 module 간에 1 대 1 변환 룰을 가지고 있고, Verilog 의 module 은 전문가에 의해 작성된 module 을 이용하여 변환한다. in/output 변수는 reg 로 변환, Connector & continuation 변수는 wire 로 변환, feedback 변수는 scan time 시 할당되게 변환된다.

2.5 Co-Simulation 환경

FBDtoVerilog 의 correctness 를 검증을 위한 Co-Simulation 환경을 그림 1 에서 확인 할 수 있다. 시나리오 생성기를 통해 시나리오를 생성하면, FBD Simulator 를 이용해 FBD 를 시뮬레이션 하고, ModelSim[6]을 통해 Verilog 를 시뮬레이션 할 수 있다. FBD-Verilog Comparator 를 이용해 두 시뮬레이션의 결과를 비교할 수 있고, 두 시뮬레이션 결과가 일치한다면 두 프로그램은 동일한 기능을 수행하는 것으로 추측할 수 있고, 이를 통해 FBDtoVerilog 가 변환을 correct 하게 했다는 것을 간접적으로 검증할 수 있다.



(그림 1) FBDtoVerilog 변환기의 Correctness 를 증명하기 위한 Co-Simulation 환경

3. 시나리오 생성기

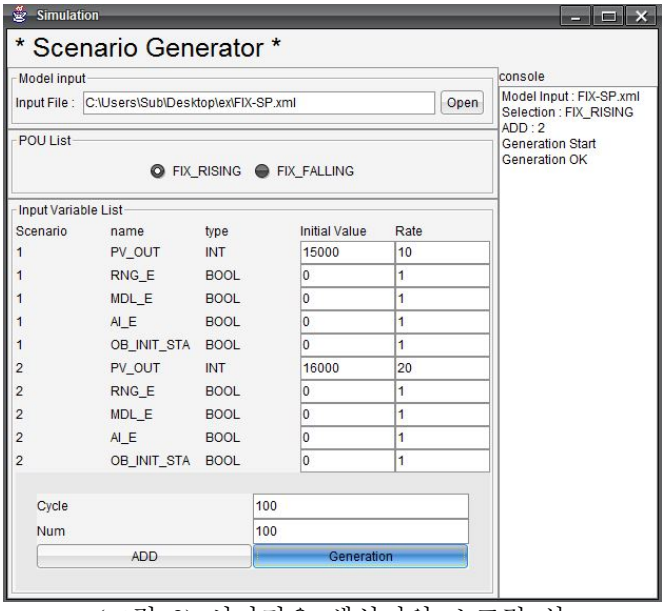
Co-Simulation 환경 에서 가장 중요한 부분은 시뮬레이션을 하기 위한 시나리오의 생성이다. 테스팅과 마찬가지로 시뮬레이션을 통한 검증은 모든 경우에 대해서 확인 할 수 없다는 단점이 있다. 따라서 Co-Simulation 결과의 신뢰성을 높이려면 시나리오의 개수나 커버리지 등이 중요한 요소이다. 우리는 이를 지원하기 무한한 수의 시나리오를 생성할 수 잇고, 도메인의 특징을 반영하여 적절한 시나리오 생성할 수 있는 시나리오 생성기를 구현 하였다.

3.1 시나리오 생성기 특징

구현된 시나리오 생성기는 다음과 같은 특징을 가지고 있다. 1) FBD 로 작성된 프로그램의 input 에 입력받은 변화율을 바탕으로 랜덤 값을 생성한다. 2) 무한한 수의 시나리오를 생성 할 수 있다. 3) 도메인의 특징을 반영한 적절한 시나리오를 생성한다. 4) FBD 를 위한 시나리오뿐만 아니라, ModelSim 을 위한 test bench 를 동시에 생성해 테스터의 수고를 줄여준다.

3.2 시나리오 생성기 구성

구현된 시나리오 생성기는 크게 4 개의 파트로 구성되어 있다. 그림 2 를 통해 이를 확인 할 수 있다. 1) Model Input 파트: 사용자로부터 시뮬레이션 하길 원하는 FBD 를 입력 받는다. 2) POU List 파트: 입력된 FBD 로부터 pou (Program of Unit)를 분류하여 사용자에게 시뮬레이션 하길 원하는 pou 를 입력 받는다. 3) Input Variable List 파트: 사용자로부터 input 의 initial 값, 변화율, cycle 수 등을 입력 받는다. 4) Console 파트: 시나리오 생성기의 현재 진행 상태를 알려준다.



(그림 2) 시나리오 생성기의 스크린 샷

3.3 시나리오 생성기의 입, 출력

시나리오 생성기는 입력으로 FBD(PLCopen Standard[7])를 받아 FBD Simulator 를 위한 시나리오를 생성한다. 또한, 이와 동일한 내용의 ModelSim 을 위한 test bench 를 동시에 생성한다. 그림 3 은 FBD Simulator 를 위한 시나리오를 보여주고 그림 4 는 동일한 내용을 가지고 있는 ModelSim 을 위한 test bench 를 보여준다. 그림 3 의 PV_OUT 값에 15000, 14998, 15001, …… 가 차례로 생성된 것을 볼 수 있고, 그림 4 를 통해 test bench 에도 PV_OUT 에 15000, 14998, 15001, …… 이 차례로 생성된 것을 볼 수 있다.

FBD

```
Name begin
text1_1_0
end

Inputs begin
PV_OUT
end

Simulation begin
15000   14998   14997   15001   14996   15000   15000   14995   14997   14992
14990   14986   14977   14975   14969   14967   14963   14968   14961   14964
14963   14954   14953   14961   14958   14954   14956   14953   14946   14952
14946   14955   14962   14956   14962   14966   14965   14963   14962   14971
14967   14963   14959   14953   14950   14941   14932   14929   14924   14929
14928   14934   14927   14924   14918   14917   14917   14924   14927   14922
14918   14918   14919   14923   14918   14913   14920   14927   14922   14919
14920   14918   14909   14900   14901   14902   14908   14913   14913   14911
14915   14912   14912   14912   14907   14913   14904   14898   14890   14889
14891   14891   14890   14898   14895   14904   14903   14897   14894   14901

end
```

(그림 3) FBD Simulation 을 위한 scenario

Verilog

```
initial
begin
    #(SYSCLK_PERIOD * 10 )
        NSYSRESET = 1'b1;
    #(SYSCLK_PERIOD * 10 )
        NSYSRESET = 1'b0;
    #(SYSCLK_PERIOD * 50 )  pulse = 1;  PV_OUT = 15000; #(SYSCLK_PERIOD / 2.0 ) pulse = 0;
    #(SYSCLK_PERIOD * 50 )  pulse = 1;  PV_OUT = 14998; #(SYSCLK_PERIOD / 2.0 ) pulse = 0;
    #(SYSCLK_PERIOD * 50 )  pulse = 1;  PV_OUT = 14997; #(SYSCLK_PERIOD / 2.0 ) pulse = 0;
    #(SYSCLK_PERIOD * 50 )  pulse = 1;  PV_OUT = 15001; #(SYSCLK_PERIOD / 2.0 ) pulse = 0;
    #(SYSCLK_PERIOD * 50 )  pulse = 1;  PV_OUT = 14996; #(SYSCLK_PERIOD / 2.0 ) pulse = 0;
    #(SYSCLK_PERIOD * 50 )  pulse = 1;  PV_OUT = 15000; #(SYSCLK_PERIOD / 2.0 ) pulse = 0;
    #(SYSCLK_PERIOD * 50 )  pulse = 1;  PV_OUT = 15000; #(SYSCLK_PERIOD / 2.0 ) pulse = 0;
    #(SYSCLK_PERIOD * 50 )  pulse = 1;  PV_OUT = 14995; #(SYSCLK_PERIOD / 2.0 ) pulse = 0;
```

(그림 4) 그림 3 과 동일한 내용의 ModelSim 을 위한 Test bench

3.4 시나리오 생성의 기본 원리

시나리오 생성기는 FBD 로 작성된 프로그램의 input 에 사용자로부터 입력 받은 초기값과, 변화율, cycle 등을 바탕으로 랜덤 값을 생성하여 시나리오를 생성한다. 초기값을 이용해 시나리오의 처음 cycle 의 값을 생성하고, 이를 바탕으로 입력 받은 변화율 내에 랜덤한 정수 하나를 선정, 이를 현재 값에 더하여 다음 cycle 을 위한 값을 생성한다. 입력 받은 cycle 을 바탕으로 위 연산 과정을 계속하여 하나의 시나리오를 완성한다.

그림 2 를 보면 초기값이 15000 이고 변화율이 10 인 것을 확인 할 수 있다. 생성된 시나리오(그림 3)을 보면 첫 번째 cycle 의 값에 입력 받은 초기값인 15000 이 생성되었고, 다음 cycle 값으로 14998 이 생성된 것을 확인 할 수 있다. 이는 변화율 10 내에서 -2 가 선택되어, 이전 cycle 값인 15000 에 -2 를 한 값인 14998 이 생성 된 것이다. 입력 받은 cycle 수가 100 이였기 때문에 총 100 개의 값이 생성된 것을 확인 할 수 있다.

3.5 생성된 시나리오의 문법

생성된 시나리오는 특정한 문법을 가지고 있다. 이런 문법의 사용은 추후 FBD Simulator 와 연동하는데 크게 도움이 될 것으로 생각된다. 기본적으로 문법은 크게 3 개의 파트로 구성되어 있다. 1) Name: 생성된 시나리오를 식별하는 이름을 적는 파트이다. 2) Inputs: pou 내 존재하는 입력 변수들은 나타낸다. 3) Simulation: 랜덤 함수를 이용하여 생성한 값들을 나타낸다. 각 값들은 한 cycle 을 위한 값들로써 추후 FBD Simulator 는 해당 값들을 parsing 하여 입력으로 활용 될 것이다. 모든 파트는 begin 으로 시작하고 end 로 끝을 내는 식별자를 가지고 있다.

3.6 도메인 특징을 반영한 시나리오 생성

시나리오 생성기는 도메인의 특징을 반영한 적절한 시나리오를 생성한다. 구현한 시나리오 생성기의 도메인은 RPS(Reactor Protection System) BP(Bistable Process) 로써, BP 의 input 인 공정변수는 기본적으로 연속적인 값을 가진다는 특징을 가지고 있다. 다시 말해, 현재 100 이였던 공정 변수가 다음 cycle 에서 10000 으로 급격하게 변하지 않는다는 특징을 가지고 있다. 이를 위해 시나리오 생성기는 사용자로부터 해당 변수가 한 cycle 동안 변할 수 있는 값인 변화율을 입력 받는다. 해당 변화율을 바탕으로 시나리오 생성기는 변화율 안에서 랜덤 값을 생성하여 입력 값이 급격하게 변하지 않는 시나리오를 생성하게 된다. 또한 공정변수는 항상 초기값을 가지고 있어야 하기 때문에, 이를 위해 시나리오 생성기는 사용자로부터 초기값을 입력 받아 이를 반영한 적절한 시나리오를 생성 할 수 있게 된다.

4. 적용 사례

우리는 KNICS[8] RPS BP 중 FIX-RISING 과 FIX-FALLING 에 이용하여 시나리오 생성기를 적용해 보았다. 나리오 생성기를 이용하여 시나리오를 생성하였고 Co-Simulation 환경(prototype)을 이용하여 FBD 와 Verilog 간의 Correctness 를 확인 하였다. 총 2000 개의 시나리오를 자동 생성 하였고, Co-Simulation 을 수행하였다. 결과는 모두 Correct 한 것으로 나왔고, 이로 미루어 보아 FBDtoVerilog 가 변환을 correct 하게 했다는 것을 간접적으로 증명 할 수 있었다. 표 1 을 통해 생성된 시나리오의 초기값과 변화율, 이를 통해 수행한 Co-Simulation 의 결과를 확인 할 수 있다.

<표 1> 시나리오 생성기를 이용한 Co-Simulation 결과

Name of Logic	Scenarios	Initial Values	Rate of Change	Cycles
FIX-RISING	1,000	27,000 – 28,000 (Stepwise: 100)	10 – 100 (Stepwise: 10)	100
FIX-FALLING	1,000	12,000 – 13,000 (Stepwise: 100)	10 – 100 (Stepwise: 10)	100
Total	2,000	All correct		

5. 결론 및 향후 연구

본 논문은 FBDtoVerilog 변환기의 Correctness 를 검증하기 위한 Co-Simulation 환경을 지원하는 도구인 시나리오 생성기를 구현 하였다. 구현한 시나리오 생성기는 FBD 로 작성된 프로그램을 입력으로 받아 FBD Simulator 를 위한 시나리오와 동일한 내용의 ModelSim 을 위한 test bench 를 자동 생성해 준다. 생성된 시나리오는 변화율과 초기값을 바탕으로 도메인의 특성을 반영한 적절한 시나리오를 생성할 수 있고, 무한한 수의 시나리오를 생성 할 수 있는 장점이 있다. 또한 FBD 를 위한 시나리오뿐만 아니라 ModelSim 을 위한 test bench 를 동시에 생성은 사용자의 편의를 도모 할 수 있다는 장점이 있다.

Co-Simulation 의 결과는 시나리오의 질과 양에 의존하게 된다는 점을 해결하기 시나리오 생성기를 구현 하게 되었고, 구현된 시나리오 생성기를 이용한 Co-Simulation 은 FBDtoVerilog 변환기의 correctness 검증을 보다 신뢰성 있게 수행할 수 있을 것으로 기대하고 있다. 추후 보다 높은 수준의 신뢰성 보장을 위해, 시나리오 생성시 특정 커버리지를 만족하는 시나리오 또는 경계 값을 중심으로 생성된 시나리오 등 보다 시나리오의 질을 높일 수 있는 연구를 진행 할 예정이다.

사 사

본 연구는 한국원자력연구원 주요사업 “원자력계측제어 적합성평가, 감시 및 대응 체계 구축” 사업과 창의연구사업 “FPGA-기반 제어기 통합개발환경을 위한 핵심 소프트웨어 기술 개발” 사업의 지원으로 연구한 결과입니다.

참고문헌

[1] 김의섭, 이동아, 유준범, "번역기, 코드 생성기 및 컴파일러를 위한 검증기법 조사," 2013 한국소프트웨어공학학술대회 (KCSE 2013), pp.43-51, 2013.

[2] Dong-Ah Lee, Eui-Sub Kim, Junbeom Yoo, “FBDtoVerilog 2.0: An automatic translation of FBD into Verilog to develop FPGA,” International Conference on Information Science & Applications (ICISA2014), (in press)

[3] Junbeom Yoo, Jong-Hoon Lee, 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] IEC. “IEC 61131-3, International standard for Programmable controllers - Part 3: Programming languages.” 2013.

[5] IEEE Computer Society. “IEEE Std 1364-2005,” IEEE Standard Verilog Hardware Description Language, 2006.

[6] Mentors Graphics, ModelSim “http://www.mentor.com/products/fpga/model”

[7] PLCopen. PLCopen for efficiency in automation. “http://www.plcopen.org.”

[8] KAERI. “KNICS-RPS-SRS101 Rev.00.”, 2003.