

FBDtoVHDL: FPGA 개발을 위한 FBD로부터 VHDL 자동 변환 도구

김재엽¹, 김의섭¹, 유준범¹, 이영준², 최종균²

¹건국대학교 컴퓨터공학부, ²한국원자력연구원

{ksjy1990, atang34, jbyoo}@konkuk.ac.kr, {yjlee426, choijg}@kaeri.re.kr

FBDtoVHDL: An automatic translation from FBD into VHDL for FPGA development

Jae-Yeob Kim¹, Eui-Sub Kim¹, Junbeom Yoo¹, Young Jun Lee², Jong-Gyun Choi²

¹Division of Computer Science and Engineering, Konkuk University

²Korea Atomic Energy Research Institute

요 약

PLC (Programmable Logic Controller)는 원자력 발전소의 디지털 제어시스템의 개발을 위해 널리 사용되어왔지만 복잡성의 증가와 유지보수 비용 등의 문제로 인해 FPGA (Field Programmable Gate Array) 기반 제어시스템이 대안으로 떠오르고 있다. 하지만 FPGA 기반 제어시스템을 개발하기 위해서는 PLC 개발자가 새로운 언어를 사용해야 하며, PLC 개발에 사용된 노하우 및 지식의 재사용이 어렵다는 문제가 있다. 본 논문에서는 PLC 소프트웨어 개발을 위한 언어 중 하나인 FBD (Function Block Diagram)를 FPGA 개발을 위한 하드웨어 기술 언어 중 하나인 VHDL로의 자동 변환 방법을 소개하고 이를 기반으로 개발한 자동 변환 도구인 FBDtoVHDL을 소개한다. 그리고 KNICS RPS BP의 로직을 이용한 실험을 통해 FBDtoVHDL를 이용해 변환한 VHDL이 FBD와 동일한 기능을 수행 하는지 확인하였다.

1. 서 론

원자력 발전소의 안전계통이 아날로그 기반의 I&C (Instrumentation and Control) 시스템에서 디지털 기반의 I&C로 전환되면서 소프트웨어와 네트워크가 시스템의 일부가 되었다. 디지털 시스템으로의 전환을 통해 신뢰도와 더 나은 성능을 제공하게 되었지만[1] 소프트웨어 등의 디지털 기술이 사용되면서 공통원인고장의 발생 가능성과 유지보수 비용, 복잡성의 증가가 문제로 떠올랐다.

하드웨어 기반의 FPGA (Field Programmable Gate Array)를 이용한 다양성 확보는 공통원인고장으로부터 RPS를 보호하기 위한 방법 중 하나이다[2]. FPGA는 하드웨어 기술 언어를 사용해 시스템을 개발해야 하지만 PLC 기반 언어와는 다르고 새로운 개발 프로세스를 따라 개발해야 한다. 따라서 PLC 기반의 시스템을 개발하던 엔지니어가 FPGA 기반 시스템을 개발하는 것은 새로운 언어를 습득해야 하는 어려움과 PLC 기반의 경험과 지식 등을 이용하지 못한다는 문제점이 있다.

본 논문에서는 PLC 소프트웨어 개발을 위한 언어 중 하나인 FBD로부터 하드웨어 기술 언어 중 하나인 VHDL로의 변환을 소개한다. FBD를 VHDL로 변환함으로써 PLC 개발자는 하드웨어 기술 언어에 대한 지식이 없이도 FPGA를 개발하는 것이 가능할 것이다. 또한 PLC 소프트웨어에 적용되던 모든 V&V와 안전성 분석이 FPGA 개발에서도 유효하게 될 것이다. 우리는 변환 규칙을 기반으로 구현한 자동 변환 도구인 FBDtoVHDL을 소개하고 이를 통해 변환한 VHDL이 FBD와 같은 기능을 수행하는지 확인하기 위해 KNICS RPS BP 로직을 이용한 실험을 수행하였다. 추가로 이후 FPGA 개발프로세스

에 적용 가능한 VHDL인지 확인하기 위해 합성가능 여부도 확인하여보았다.

2. 배경 지식

2.1 FBD (Function Block Diagram)

FBD는 IEC 61131-3 표준[3]에 정의된 5가지 PLC 프로그래밍 언어 중 하나이다. FBD는 그래픽 기반의 표현이 가능한 언어로 각각의 기능을 수행하는 FB(Function Block)들과 이들의 연결로 표현이 된다. FB는 사각형으로 묘사되고 입력/출력 값과 연결되며 각 FB는 산술연산, 논리연산, 비교연산, 선택연산, 시간연산 등의 연산을 수행한다. 각 FB는 실행 순서가 부여되고 선으로 연결되어 절차적인 흐름을 표현하게 된다.

<그림 1>은 FBD로 구현된 KNICS RPS BP의 FIX_FALLING [4]의 로직 중 일부로 이전에 연산된 cond_final_(P)TRIP_LOGIC의 값과 에러 값과 상태 값을 고려해서 출력 값인 (P)TRIP을 결정하는 기능을 수행한다. 블록 위 괄호 안의 숫자는 블록의 실행 순서로 낮은 숫자부터 높은 숫자 순으로 실행된다. 31번째 실행 순서를 가지는 AND_BOOL_2 블록은 29번째와 30번째 실행순서를 가지는 LT_INT_2가 실행되면 그 출력을 입력으로 하여 연산을 수행하는 것이다.

2.2 VHDL (VHSIC Hardware Description Language)

VHDL은 IC(Integrated Circuit) 개발자들에게 널리 사용되는 하드웨어 기술 언어 중 하나로 IEEE 표준 언어이다[5]. VHDL

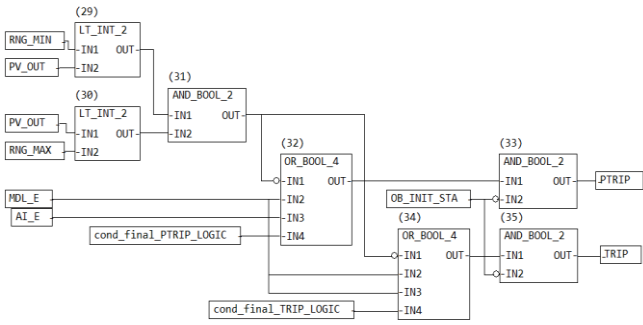


그림 1. FBD로 구현된 FIX_FALLING 로직의 일부

을 이용한 설계는 시스템 수준부터 게이트 수준까지 표현하는 것이 가능하기 때문에 넓은 범위의 설계가 가능하다는 장점이 있다. 또한 VHDL을 이용한 하향식 설계가 가능하기 때문에 설계에서 발생할 수 있는 에러를 초기에 발견하는 것이 가능하고 이를 이용하여 설계에 필요한 비용을 최소화할 수 있다.

3. FBDtoVHDL

FBDtoVHDL은 기존의 PLC 기반 제어를 개발하기 위해 사용된 언어 중 하나인 FBD를 FPGA 개발을 위한 VHDL로 자동 변환하는 도구이다. 우리는 이를 위해서 FBD를 VHDL로 변환하는 규칙을 정의하였고, 이를 토대로 자동 변환 도구를 구현하였다.

3.1 변환 규칙

FBD를 VHDL로 변환하는 규칙은 <그림 2>와 같다. 1~3번 규칙은 POU의 인터페이스 부분과 4~7번 규칙은 바디 부분과 관련된 규칙이다. 규칙 1의 경우 변환 중인 FBD의 입출력 값을 VHDL 형태로 선언하는 규칙으로 FBD에서 값의 이름과 데이터 타입을 이용하여 규칙과 같이 입력, 출력, 피드백인 경우를 구분하여 선언한다.

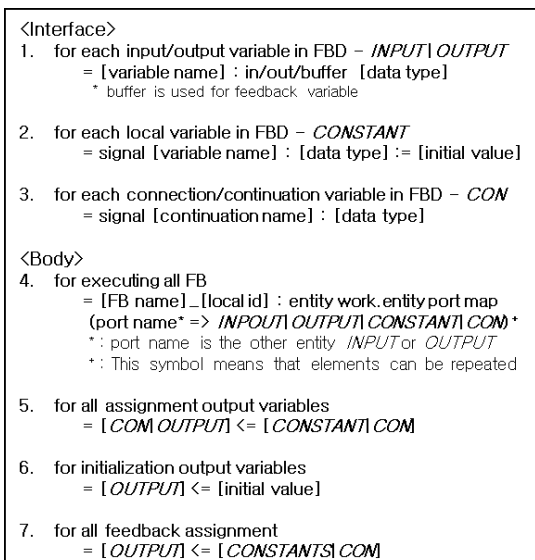


그림 2. FBD를 VHDL로의 변환 규칙

<그림 3>은 <그림 1>의 FBD를 대상으로 변환 규칙을 적용하여 변환을 수행한 결과 중 29번째 실행순서를 가지는

LT_INT_2와 관련된 부분과 특징적인 부분만을 변환 결과 프레임에 적용한 결과이다. 변환 결과 프레임은 굵게 표현된 부분으로 VHDL의 기본 구조를 표현하기 위한 부분이다. 11번째 줄은 입력으로 사용되는 PV_OUT을 선언하는 부분이고 16번째 줄은 입력 값과 비교를 위해 사용되는 지역 변수인 RNG_MIN을 선언하는 부분이다. 18번째 줄은 LT_INT_2_29의 출력을 다음 FB에 연결하기 위해 사용할 시그널인 LT_INT_2_wire_29_OUT을 선언하는 부분이다. 20~22번째 줄은 LT_INT_2_29를 선언하고 연산을 수행하는 부분이다. 사전에 만들어 둔 LT_INT_2의 입력 포트에 RNG_MIN과 PV_OUT을 연결하고 출력 포트에 LT_INT_2_wire_29_OUT을 연결하여 29번째 실행 순서를 가지는 FB의 변환이 완료된다. FBD의 출력 중 하나인 TRIP에 값을 할당하는 부분은 24번째 줄로 35번째 실행 순서를 가지는 AND_BOOL_2의 출력 값을 TRIP에 할당하게 된다. 26~32번째 줄은 프로세스로 구현한 부분으로 rst의 값을 확인하여 1이면 출력 값을 초기화하고 pulse 값을 확인하여 입력으로 할당하는 연산을 수행하게 된다.

```

1: entity FIX_FALLING is (
2:   generic (
3:     INT_HI : integer := 8388607;
4:     INT_LO : integer := -8388608
5:   );
6:   port (
7:     clk : in std_logic;
8:     rst : in std_logic;
9:     pulse : in std_logic;
10:
11:     PV_OUT : in integer range INT_LO to INT_HI;
12:
13:   end FIX_FALLING;
14:
15:   architecture Behavioral of FIX_FALLING is
16:     signal RNG_MIN: integer range INT_LO to INT_HI := 600;
17:
18:     signal LT_INT_2_wire_29_OUT : std_logic;
19:
20:     LT_INT_2_29 : entity work.LT_INT_2 port map
21:       (clk => clk, rst => rst, A_i => RNG_MIN,
22:        B_i => PV_OUT, R_o=>LT_INT_2_wire_29_OUT);
23:
24:     TRIP <= AND_BOOL_2_wire_35_OUT;
25:
26:   process(clk, rst) begin
27:     if(rst = '1') then
28:       TRIP_LOGIC <= '0';
29:     elseif(pulse = '1') then
30:       TRIP_LOGIC <= SEL_BOOL_2_wire_27;
31:     end if;
32:   end process;
33: end Behavioral;
    
```

그림 3. FIX_FALLING의 일부(그림 1)를 변환한 결과

3.2 자동 변환 도구 구현 (FBDtoVHDL)

우리는 3.1장에서 설명한 변환 규칙을 이용하여 FBD를 FPGA 개발을 위한 VHDL로의 자동 변환을 수행하는 도구인 FBDtoVHDL을 구현하였다. 도구는 크게 두 부분으로 FBD를 읽고 변환이 가능한 형태인지 검증하는 부분과 검증을 통과한 FBD를 VHDL로의 변환을 수행하는 부분으로 이루어져 있다.

FBDtoVHDL에서는 PLCopen TC6 XML 버전 2.01 스키마 [6]를 만족하는 FBD를 대상으로 변환을 수행한다. 스키마를

만족하는 FBD가 입력일 경우 변환 규칙에 따른 변환을 수행하며 <그림 3>와 같은 형태의 '*.vhd'의 확장자를 가지는 VHDL 파일의 형태로 지정한 폴더에 저장된다.

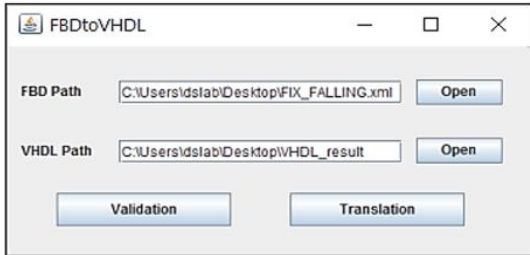


그림 2. FBDtoVHDL 자동 변환 도구

4. 사례 연구

우리는 KNICS RPS BP의 FIX_RISING과 FIX_FALLING을 대상으로 FBDtoVHDL을 통한 변환을 수행하였다. 각 로직은 4개의 입력과 8개의 출력을 가지고 33개의 FB를 포함한다. FBDtoVHDL을 통한 변환 후의 VHDL은 기본적인 입력 3개 (clk, rst, pulse)를 제외하면 FBD와 같은 입출력을 가진다. 또한 33번의 entity 호출이 같은 수의 FB와 동일한 기능을 수행하는 것을 확인하였다. 변환 전후를 비교하면 1:1 수준의 변환이 이루어졌다고 할 수 있다. <표 1>은 FBD와 변환 후의 VHDL을 분석한 결과이다.

표 1. FBD와 FBDtoVHDL을 통한 변환 후의 VHDL 비교

	FBD	VHDL
Input Variable	4	4
Output Variable	8	8 (out: 2, buffer: 6)
Local Variable	11	11
Function	33 (Function Block)	33 (entity call)
Initialization	Correct	

우리는 FBDtoVHDL을 통한 변환이 제대로 이뤄졌는지 확인하기 위해 시뮬레이션 결과를 통한 행위의 일치성을 확인하였다. 일치성 확인을 위해서 IST-FPGA[7]의 Scenario Generator를 이용해 시나리오와 테스트벤치를 생성하였으며 FBD Simulator와 ModelSim[8]으로 시뮬레이션을 수행하였다. <표 2>는 테스트 벤치의 구성과 비교 결과이다. 시뮬레이션에는 합리적인 범위 내에서 무작위로 생성된 100개의 입력을 가지는 1000개의 시나리오가 사용되었다. 시뮬레이션 분석 결과 모든 출력에서 동일한 값이 생성되는 것을 확인하였으며, 이는 FBD와 변환을 통해 생성된 VHDL이 동일한 기능을 수행한다고 할 수 있다.

표 2. FBDtoVHDL의 결과를 대상으로 수행한 실험 결과

	FIX_RISING	FIX_FALLING
시나리오 수	1000	1000
초기 값	27,500	12,500
변화율	10-100 (10 씩 증가)	10-100 (10 씩 증가)
입력 횟수	100	100
출력 비교	모든 출력이 일치	

5. 결론 및 향후 연구

본 논문에서는 FBD를 VHDL로 자동 변환 규칙과 이를 기반으로 구현한 자동 변환 도구인 FBDtoVHDL을 소개하였다. 또한 KNICS RPS BP의 두 개의 로직을 대상으로 변환을 수행하여 FBD로부터 동일한 기능을 수행하는 VHDL이 생성되는 것을 확인하였다. 이를 통해 PLC 개발자가 하드웨어 기술 언어에 대한 지식 없이도 FPGA를 개발할 수 있으며 기존의 V&V 및 안전성 분석 등의 기술을 재사용할 수 있게 될 것이라 기대한다. 앞으로 우리는 변환한 VHDL을 이용하여 FPGA 개발 프로세스를 수행하고 실제 하드웨어에서 문제가 없는 설계가 생성되는지 확인하는 연구를 수행 할 예정이다.

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 Atomic Energy Agency, “Instrumentation and control (I&C) systems in nuclear power plants: A time of transition,” 2008. [Online]. Available: <http://www.iaea.org/About/Policy/GC/GC52/GC52InfDocuments/English/gc52inf-3-att5 en.pdf>
- [2] Jong-Gyun Choi, et al., “Survey of the CPLD/FPGA technology for application to NPP digital I&C system,” Korea Atomic Energy Research Institute (KAERI), Tech. Rep., 2009.
- [3] “International standard for programmable controller – Part 3: Programming languages,” IEC 61131-3 ed3/0, 2013.
- [4] KAERI, “KNICS-RPS-SRS101 Rev.00.,” 2003.
- [5] Standard VHDL Language Reference Manual. The Institute of Electrical and Electronics Engineers, Mar. 1988.
- [6] XML Formats for IEC 61131-3, PLCopen Technical Committee 6, 2009. [Online]. Available: <http://www.plcopen.org>
- [7] Jaeyeob Kim, Eui-Sub Kim, Junbeom Yoo, Young Jun Lee, Jong-Gyun Choi, "An Integrated Software Testing Framework for FPGA-based Controllers in Nuclear Power Plants", submitted to Nuclear Engineering and Technology, 2015.
- [8] Mentor Graphics, ModelSim, <http://www.mentor.com/products/fpga/simulation/modelsim>.