

IP Core library 사용이 상용 합성도구의 COTS SW Dedication 에 미치는 영향 분석

정세진 *, 김의섭 *, 유준범 *, 최종균 **, 김장열 **, 이장수 **

* 건국대학교 컴퓨터공학과

** 한국 원자력 연구원

{Jsjj0728, atang34, jbyoo}@konkuk.ac.kr

{jgchoi, jykim, jslee}@kaeri.re.kr

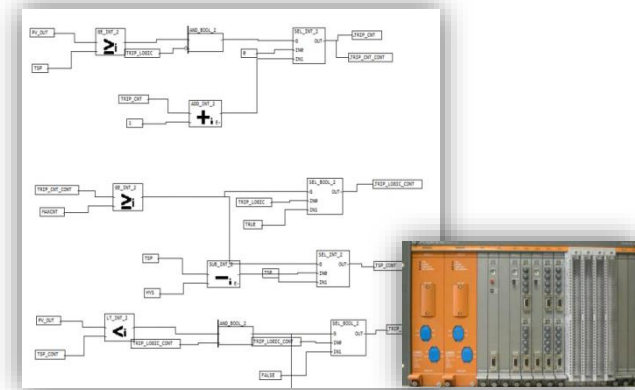
Platform Change from PLC to FPGA

원자력 발전소의 Digital I&C(Instrumentation & Control) 시스템

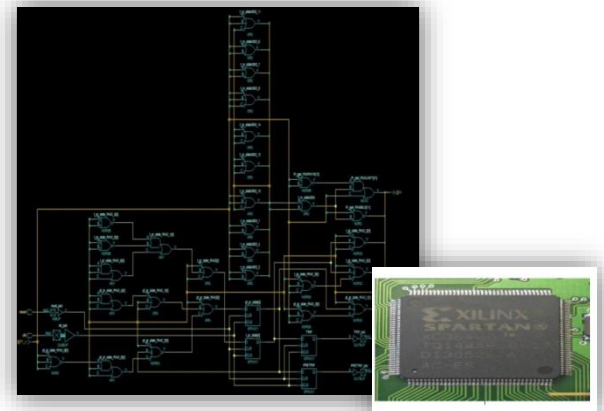
- PLC (Programmable Logic Controller)를 주로 사용
 - 유지 보수 비용의 증가
 - CCF(Common Cause Failure)로 인한 다양성 확보
- SW Development
- 다른 대체 platform에 대한 요구사항 발생



- 높은 성능과 보안성으로 인해, 대체 platform 으로서 **FPGA** 기반 I&C 에 대한 많은 연구가 진행 중
- HW Development



FBD program for PLC



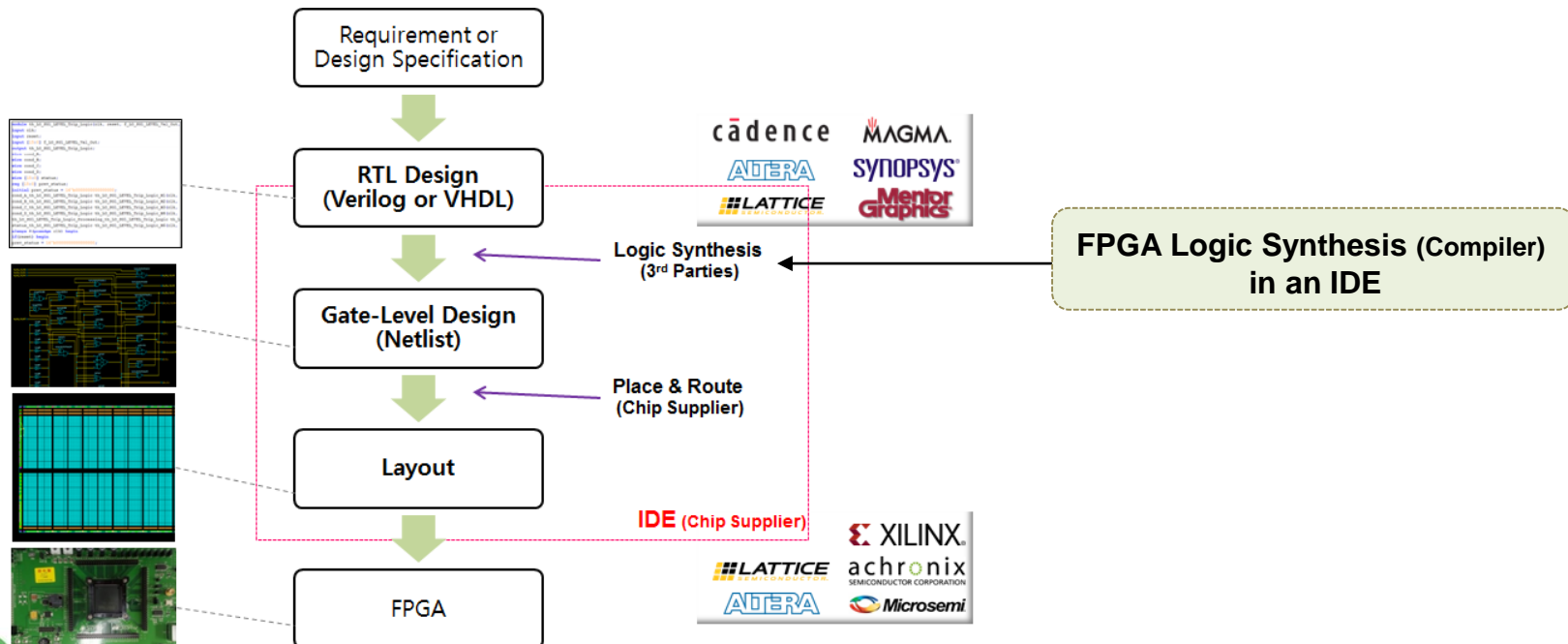
Netlist design for FPGA

FPGA 개발 및 COTS SW Dedication

- FPGA 개발에는 여러 상용 소프트웨어들이 사용됨 (e.g. 합성도구, P&R 도구 등)
- 이 중 합성도구는 RTL design을 gate 로 합성하는 도구로써 컴파일러의 역할 수행
 - 사용되는 소프트웨어들은 적절한 과정을 통해 안전성과 신뢰성 등의 입증 필요

COTS(Commercial-Off The Shelf) dedication : 원자력 발전소에 사용되는 상용 제품들의 인증

- EPRI NP-5652/TR-106439
- NRC NUREG/CR-6421
- EPRI TR-1025243 등 표준이 존재



COTS SW Dedication

A process for demonstrating correctness and safety of commercial software (COTS) used directly or indirectly

COTS SW Dedication 대상

- Direct SW : 직접 사용되는 소프트웨어
- Indirect SW : 개발에 사용되는 소프트웨어 (컴파일러, 개발환경 등) → 상용 합성도구
 - NP-5652 : Indirect SW에 대한 직접적인 정의 없음
 - NUREG/CR-6421 : '컴파일러' 포함
 - TR-1025243 : 조건에 따라 'FPGA 개발환경'도 포함

상용 합성도구의 dedication 시 → Functionality, QA, Operating history등을 고려

Standards	Target	Process	Note
EPRI NP-5652 (EPRI TR-106439)	Commercial Grade Item (CGI) + Software-based equipments	Method 1 ~ 4	Focusing on Direct CGI Base of Korean Std.
NUREG/CR-6421	Direct / Indirect COTS software	Safety category 별 인 증 진행	Indirect 포함
EPRI TR-1025243	Commercial computer program(SW)	Method 1 ~ 4 + 안전성 분류	

IP Core Library 및 사용 방법

IP Core Library

- FPGA 개발에서 설계의 편의성 및 효율성을 위해 미리 정의된 기능과 회로의 라이브러리
- FPGA 벤더 별 제공 혹은 3RD party 개발자 로부터의 제공이 가능

The screenshot shows the IP Core Library browser on the left with 'Basic Blocks' expanded. On the right, a schematic diagram shows three IP cores: 'adder_0', 'decoder_0', and 'mux_0'. 'adder_0' has inputs 'DataA[1:0]' and 'DataB[1:0]' and output 'Sum[1:0]'. 'decoder_0' has inputs 'Data0' and 'Eq[1:0]' and output 'Enable'. 'mux_0' has inputs 'Data0_port', 'Data1_port', 'Data2_port', 'Data3_port', 'Data4_port', 'Sel0', 'Sel1', and 'Sel2', and output 'Result'.

Library 사용시

- 내부 구조 및 동작에 대한 검증이 필요
- 상용 합성도구의 **dedication** 진행 시 이러한 library 사용에 대한 고려가 필요함

The screenshot shows the IP Core Library browser on the left with 'Memory & Controllers' expanded. On the right, a schematic diagram shows the 'COREDDR_0' core with various inputs and outputs. Inputs include 'AUTO_PCH', 'BLL[1:0]', 'B_SIZE[3:0]', 'CLK_0', 'CLK_1X', 'CLK_1X_SH90', 'CL_HALF', 'CS_N[7:0]', 'COLB[3:0]', 'DATA[16:0]', 'DELAY[15:0]', 'DM_W[0]', 'DOE[3:0]', 'DS[3:0]', 'DS1[0]', 'MRD[2:0]', 'RADR[30:0]', 'RAS[3:0]', 'RAS[0]', 'RAS[1]', 'RAS[2]', 'REF[15:0]', 'REGDIM', 'RESET_N', 'RFCH[0]', 'ROWB[5:0]', 'RR[2:0]', 'RR[1:0]', 'R_REQ', 'SD_INT', 'WR[1:0]', and 'W_REQ'. Outputs include 'BA[1:0]', 'CAS_N', 'CKE', 'CLK_1X', 'CLK_1X_SH90', 'CS_N[7:0]', 'DATAOUT[6:0]', 'DM[3:0]', 'DQ_OE[3:0]', 'D_REQ', 'RAS_N', 'R_VALID', 'SA[13:0]', 'WE_N', and 'W_VALID'.

상용 합성도구의 두 가지 library 사용 방법

- 디자인 단계에서 사용한 Core library 합성
- 도구 임의로 Core library 를 이용한 합성

사용 방법 별 영향 및 분석 방법

디자인 단계에서 사용한 Core library 합성

- 해당 라이브러리의 사용은 디자인에 의존적
- 디자인 대상의 검증에 포함
- 합성도구의 dedication 에 미치는 영향이 없는 것으로 판단

합성도구 임의로 Core library 를 이용한 합성

- 합성도구의 기능과 연관
- 합성도구의 dedication 과정에서 이 부분에 대한 고려가 필요
(Library 에 대한 검증 여부 확인 or 특별 시험을 통한 확인 등)

사용 여부 분석 방법

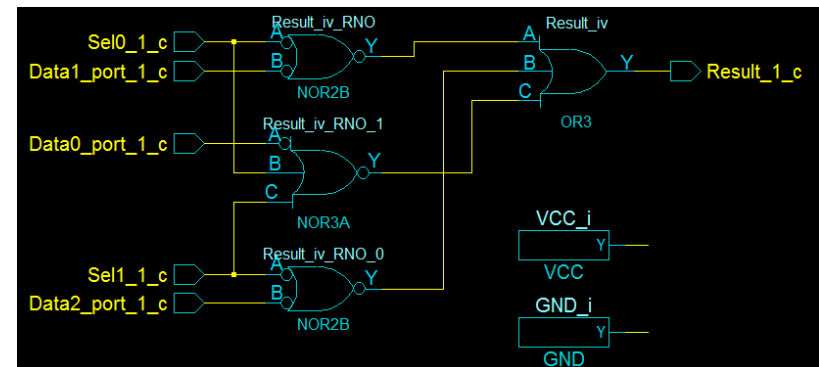
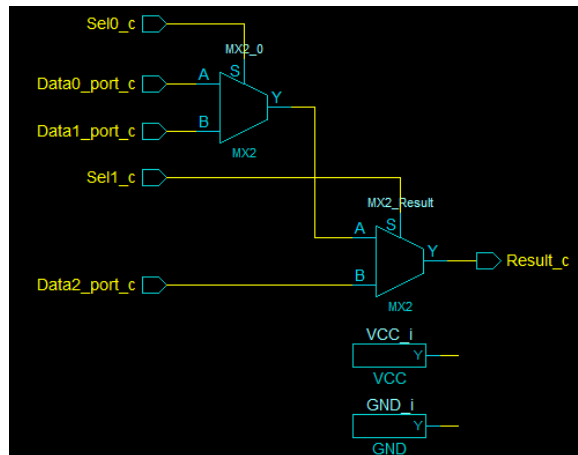
- 라이브러리 사용 유/무 디자인의 합성 결과 비교를 통한 확인
- 실제 예제 비교를 통한 확인

라이브러리 사용 유/무 디자인 비교를 통한 확인

실험 방법

- Libero SoC에서 지원하는 Smart Design을 이용해 library 사용한 디자인 생성 하여 합성
- 위 방법에서 생성한 동일한 기능을 하는 Verilog 디자인을 이용하여 합성
- 두 디자인의 합성 결과를 비교
- adder, subtractor, counter, incremter, comparator, multiplexer, multiplier를 대상으로 실험

3to1 mux 실험 결과 다른 모양을 보임



실험 디자인 비교 결과 예시

디자인 비교 결과 (4종류 예시)

- 합성 결과 gate-design 이 다름을 확인
- 사용하는 block의 종류와 개수가 다름을 확인

실험적인 디자인에서는 임의로 라이브러리를 사용하지 않는 것을 확인 가능

Logic	Smart Design (사용)	Verilog (미사용)
3 to 1 MUX	2개 (2 to 1 MUX 2)	9개 (NOT 5, NOR3 1, NOR2 2, OR3 1)
Decoder	6개 (NOT2, AND 3 NOR 1)	8개 (NOT 4, NOR 4)
4 bit Adder	11개 (AND 6 XOR3 3, XOR2 1)	15개 (AND 6, OR 2, XOR3 3, XOR2 1, NOT 2, NOR1)
Incrementer	6개 (AND3 1, AND 1, XOR 3, NOT 1)	9개 (AND3 1, XOR 3, NOT 3, NOR 2)

예제 비교를 통한 확인

실험 방법

- Verilog 로 작성된 18개 RPS BP logic의 프로토타입 합성 결과를 확인
- Manual 한 방법으로 확인 결과 Smart Design을 통해 사용할 수 있는 라이브러리를 이용하지 않음을 확인

Logic	IP Core library 사용 여부	Logic	IP Core library 사용 여부
HI_CPRS_NR	X	LO_RC1_FLW	X
HI_LOG_POW	X	LO_RC2_FLW	X
HI_LOG_POW	X	LO_SGL1_WR	X
HI_PPRS_RPS	X	LO_SGL2_WR	X
HI_SGL1_RPS	X	LOLO_SGL1_WR	X
HI_SGP2_RPS	X	LO_SGL1_WR	X
HIHI_CPRS_NR	X	LOLO_SGL1_WR	X
LO_DNBR_RPS	X	LO_SGL1_WR	X
LO_PZR_RPS	X	VAR_OVR_PWR	X

결론 및 향후 연구

결론

- 'FPGA 기반 디지털 I&C 개발' 연구에 사용된 상용 합성도구인 Synplify Pro 에 대해 Core library 사용 여부 확인
- Library 사용 여부에 따른 COTS SW dedication에의 영향 분석
- 'Synplify Pro'에 대해서는 임의적으로 사용하지 않음을 확인

향후 계획

- 합성도구의 dedication 수행 시 optimization이 미치는 영향에 대한 분석 및 연구
- 합성도구 이외에 FPGA 개발에 사용되는 소프트웨어들의 dedication 진행 시 Core library에 대해 고려할 점에 대한 연구