ISET 2015
2015.06.04-06
Deagu

Verification Techniques for COTS Dedication of Commercial FPGA Tools

Junbeom Yoo , Eui-Sub Kim , Sejin Jung
Dependable Software Laboratory
KONKUK University

2015.06.04

ISET 2015
2015.06.04-06
Deagu

KU KONKUK UNIVERSITY

Formal Verification Techniques which can be used for COTS SW Dedication of Commercial FPGA Tools used to Develop Safety-Critical Control Software in Nuclear Power Plants

Junbeom Yoo , Eui-Sub Kim , Sejin Jung
Dependable Software Laboratory
KONKUK University

2015.06.04

DEPENDABLE SOFTWARE LABORATORY

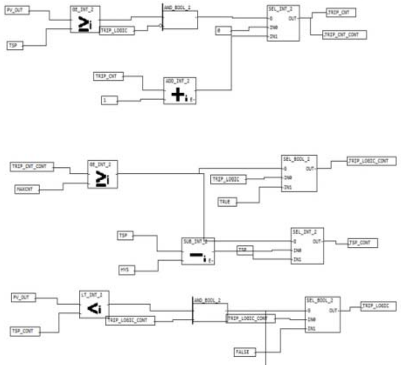
Platform Change from PLC to FPGA

Digital I&C(Instrumentation & Control) in nuclear power plants

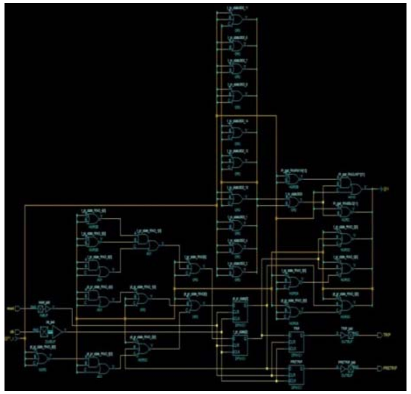
PLC(Programmable Logic Controller) has been used to implement I&Cs for decades
- SW development on industrial computers (CPU & OS)

However, increasing maintenance cost and CCF(Common Cause Fault) problem in security
- Request for alternative implementation platforms

FPGA(Field Programmable Gate Array) is an alternative platform of PLC for I&Cs
- Higher computation performance and stronger security
- HW development

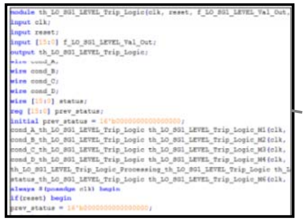


FBD program for PLC

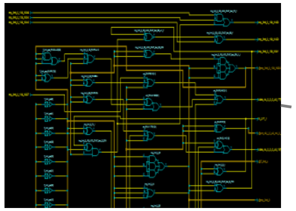


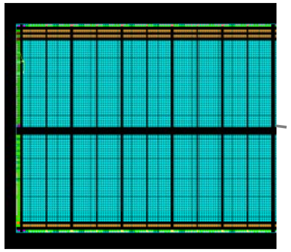
Netlist design for FPGA

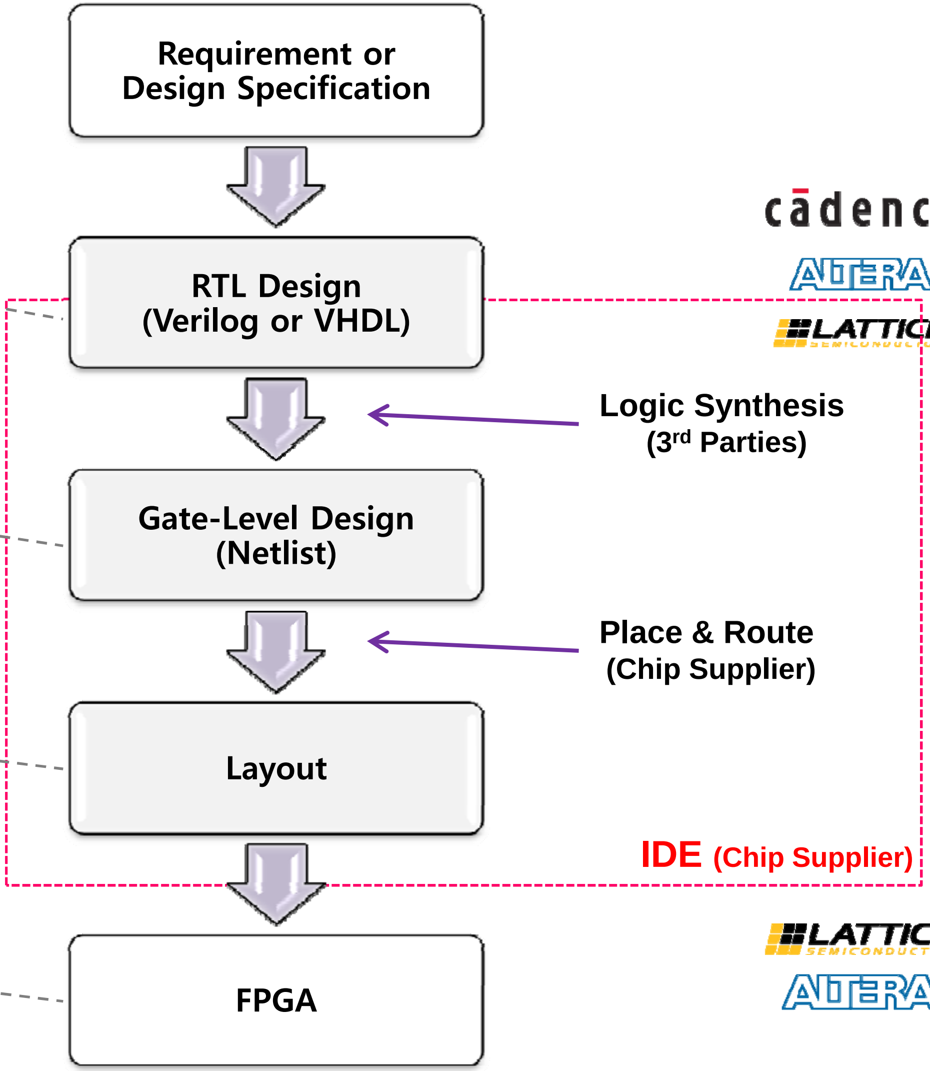
FPGA Development Process



```
module th_LO_SG1_LEVEL_Trip_Logic(clk, reset, f_LO_SG1_LEVEL_Val_Out,
input clk;
input reset;
input [15:0] f_LO_SG1_LEVEL_Val_Out;
output th_LO_SG1_LEVEL_Trip_Logic;
wire cond_A;
wire cond_B;
wire cond_C;
wire cond_D;
wire [15:0] status;
reg [15:0] prev_status;
initial prev_status = 16'b0000000000000000;
cond_A th_LO_SG1_LEVEL_Trip_Logic th_LO_SG1_LEVEL_Trip_Logic_M1(clk,
cond_B th_LO_SG1_LEVEL_Trip_Logic th_LO_SG1_LEVEL_Trip_Logic_M2(clk,
cond_C th_LO_SG1_LEVEL_Trip_Logic th_LO_SG1_LEVEL_Trip_Logic_M3(clk,
cond_D th_LO_SG1_LEVEL_Trip_Logic th_LO_SG1_LEVEL_Trip_Logic_M4(clk,
th_LO_SG1_LEVEL_Trip_Logic_Processing th_LO_SG1_LEVEL_Trip_Logic th_L
status_th_LO_SG1_LEVEL_Trip_Logic th_LO_SG1_LEVEL_Trip_Logic_M6(clk,
always @(posedge clk) begin
if(reset) begin
prev_status = 16'b0000000000000000;
```







Requirement or Design Specification

RTL Design (Verilog or VHDL)

Logic Synthesis (3rd Parties)

Gate-Level Design (Netlist)

Place & Route (Chip Supplier)

Layout

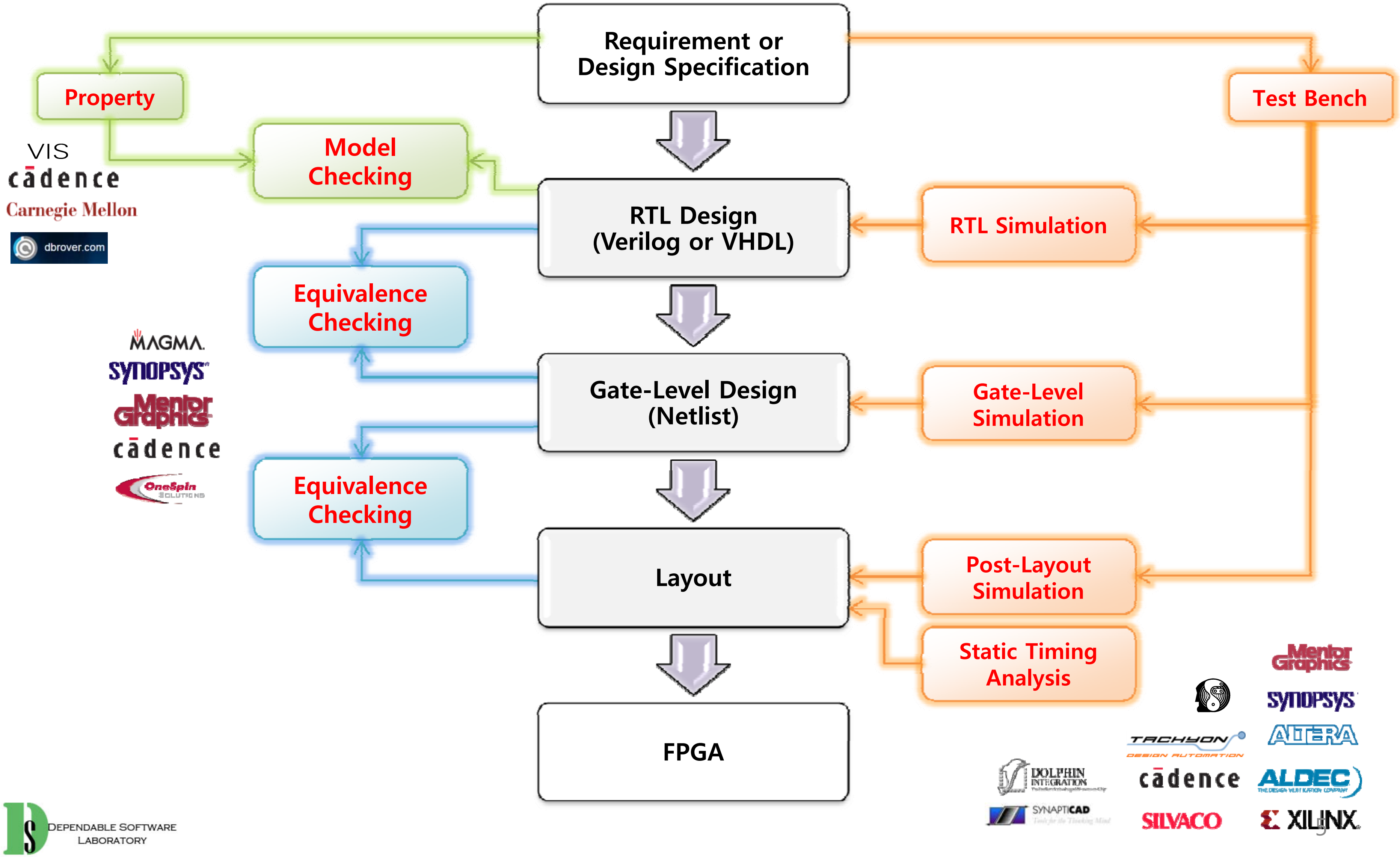
IDE (Chip Supplier)

FPGA

cadence MAGMA ALTERA SYNOPSYS LATTICE SEMICONDUCTOR Mentor Graphics

XILINX LATTICE SEMICONDUCTOR achronix SEMICONDUCTOR CORPORATION ALTERA Microsemi

FPGA Development Process + Verification



Requirement or Design Specification

Property

Test Bench

Model Checking

RTL Design (Verilog or VHDL)

RTL Simulation

Equivalence Checking

Gate-Level Design (Netlist)

Gate-Level Simulation

Equivalence Checking

Layout

Post-Layout Simulation

Static Timing Analysis

FPGA

COTS SW Dedication

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

- Direct COTS SW : Directly <u>used</u> in an application to perform safety functions
- Indirect COTS SW : Directly <u>produces</u> direct SW (not COTS SW)

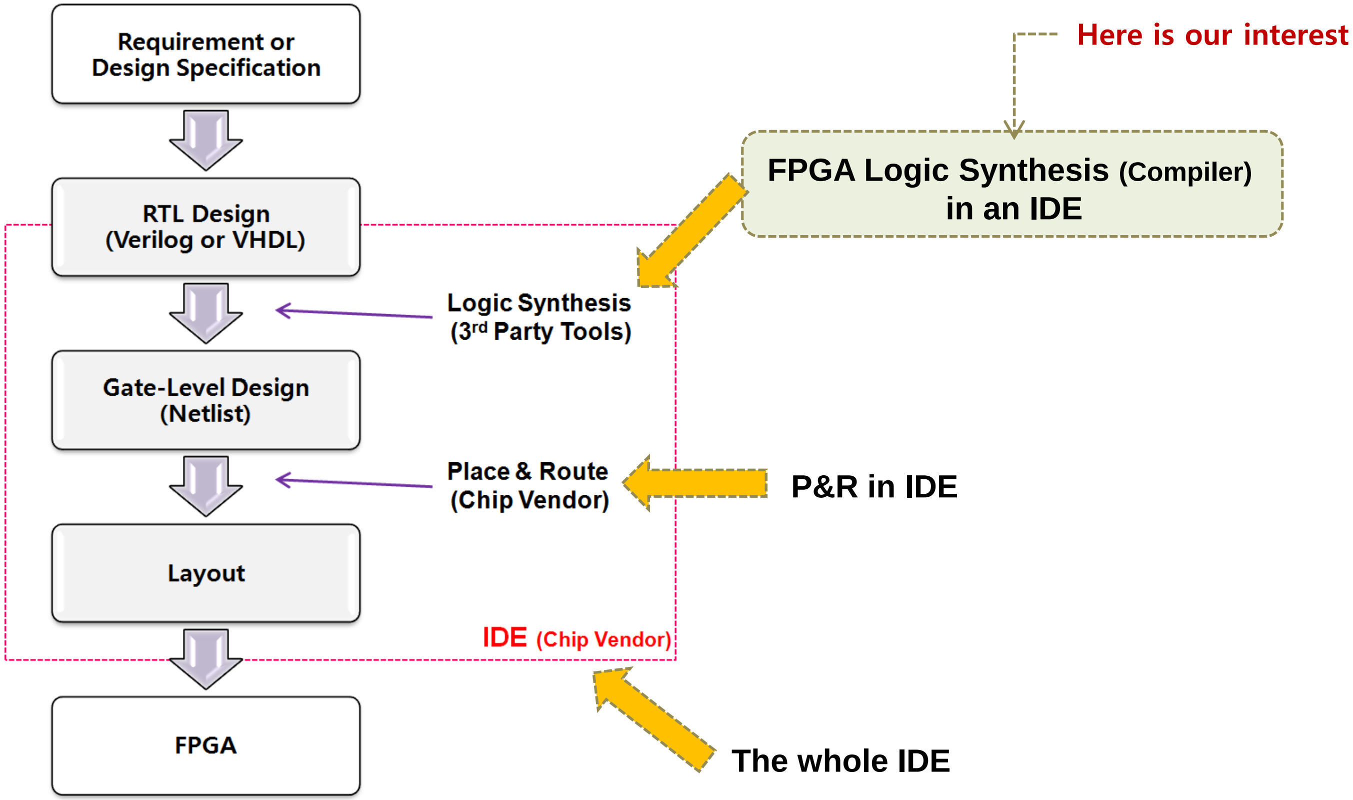
Two international standards to cope with for digital I&Cs in NPP

Standards	Target	Process	Note
EPRI-NP5652 (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	Processes for each safety category	Containing Indirect CGI

COTS SW Dedication for FPGA Development

COTS software such as logic synthesis and IDEs are always used to develop FPGA.

- Indirect COTS SW & Category B
- Should take the COTS SW dedication process according to the standards


Requirement or Design Specification
RTL Design (Verilog or VHDL)
Gate-Level Design (Netlist)
Layout
FPGA
Logic Synthesis (3rd Party Tools)
Place & Route (Chip Vendor)
IDE (Chip Vendor)
Here is our interest
FPGA Logic Synthesis (Compiler) in an IDE
P&R in IDE
The whole IDE

COTS SW Dedication : EPRI NP-5652

NP-5652 suggests 4 methods

Method 1 : Special Test and Inspection
- Verifying important functionalities

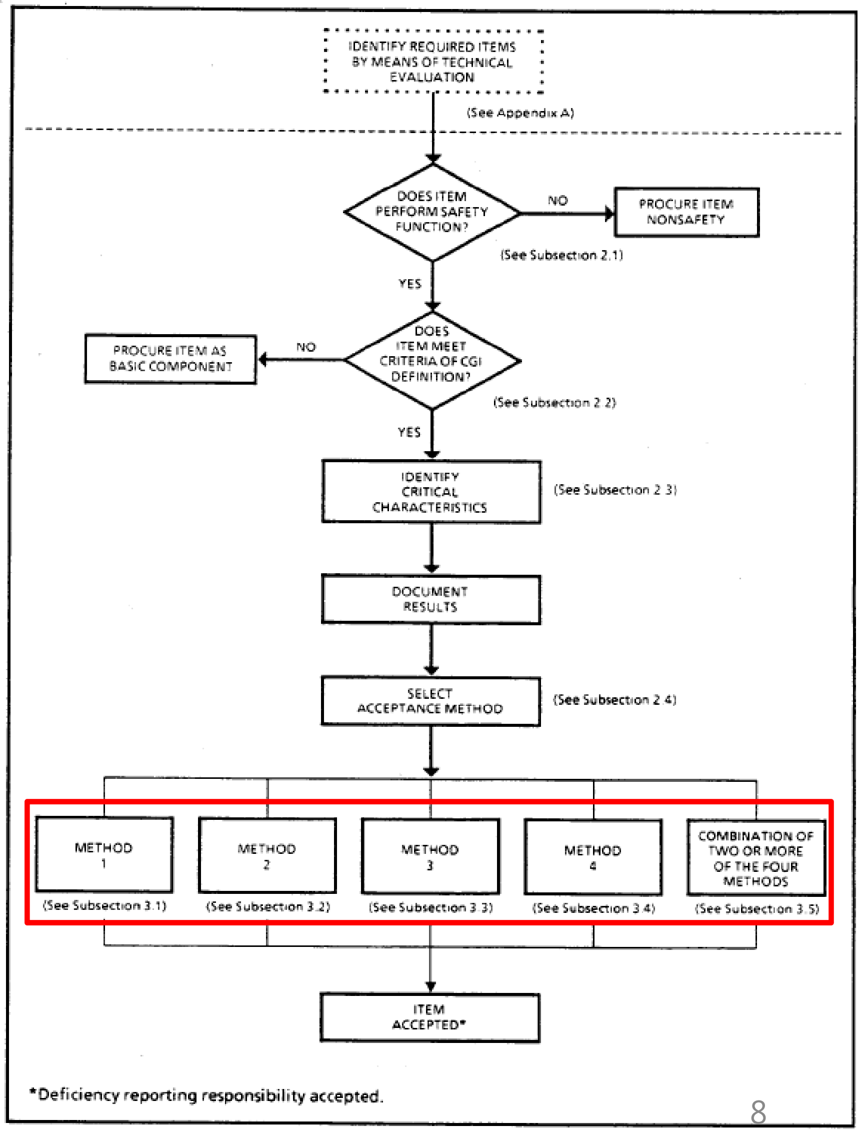
Method 2 : Commercial-Grade Survey
- Confirming and evaluating QA program of suppliers

Method 3 : Source Verification
- Verifying critical characteristics at the supplier’s facility (often impossible)

Method 4 : Item/Supplier Performance Record
- Verifying acceptability through documented items or supplier’s performance records

Method 1 is important for logic synthesis
- Functionality to verify : correct synthesis
- Direct compiler verification techniques can’t be used
- It is a commercial compiler (No source code opened)

Indirect verification is required
- Logic Equivalence Checking(LEC) for specific inputs

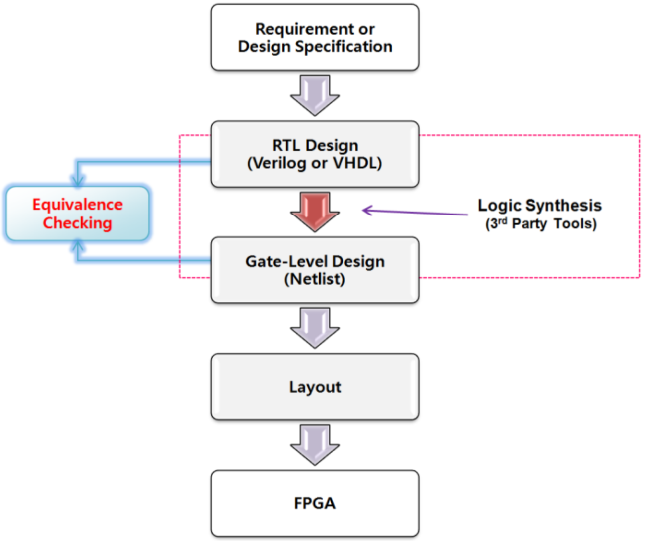


*Deficiency reporting responsibility accepted.

Logic Equivalence Checking

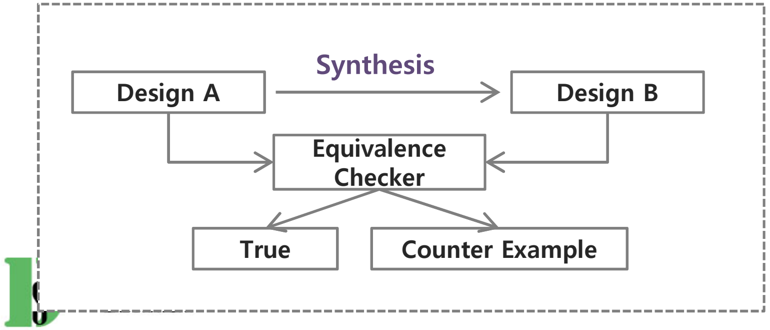
Formally verify(prove) that

- for a specific input, the output always shows the same behavior with the input


Requirement or Design Specification
RTL Design (Verilog or VHDL)
Equivalence Checking
Logic Synthesis (3^{rd} Party Tools)
Gate-Level Design (Netlist)
Layout
FPGA

Commercial LEC tools

- FormalPro (Mentor Graphics)
- Formality (Synopsys)
- Encounter Conformal EC (Cadence)
- Jasper Gold (Cadence)
- Quartz Formal (Magma Design Automation)
- 360 EC (OneSpin Solutions)



Applicability of LECs

Applicability depends on the tool combinations

- LEC x Logic Synthesis x IDEs

No applicable LEC for Synopsys Synplify Pro (in Actel Libero IDE)

- In this case, we need to develop a customized LEC

Logic Synthesis	IDE	Mentor Graphics FormalPro	Cadence Encounter Conformal EC	Synopsys Formality
Mentor Graphics Precision RTL	Xilinx ISE	O		
	Actel Libero Soc	O		
Synopsys Synplify Pro	Xilinx ISE	O	O	
	Actel Libero Soc			
	Altera Quartus Ⅱ		O	
Xilinx XST	Xilinx ISE		O	O
Synopsys DC Ultra	-			O

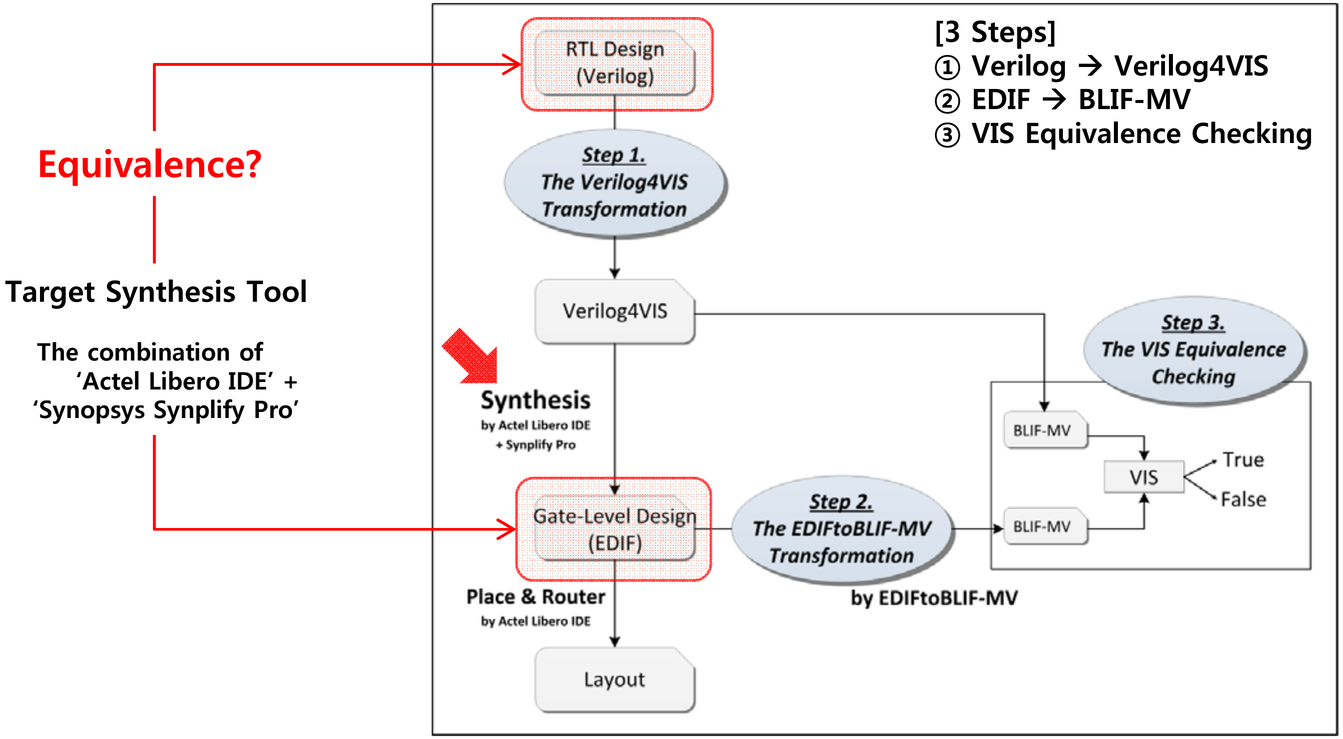
No LEC available

A New Customized LEC : CVEC (A Customized VIS based Equivalence Checking)

A VIS based solution (VIS : Verification Interacting with Synthesis)

It can verify the combination of 'Synopsys Synplify Pro' with 'Actel Libero SoC'

- An open-sourced formal verification tool, VIS
- Translators requires (step1,2) to use the VIS
- Verification performance is up to the VIS



[3 Steps]
① Verilog → Verilog4VIS
② EDIF → BLIF-MV
③ VIS Equivalence Checking

Summary

FPGA is receiving international attention as an alternative platform of digital I&Cs in NPPs.

We should do the COTS SW dedication to demonstrate correctness and safety of commercial software(COTS) used indirectly, such as FPGA logic synthesis and IDEs, according to international standards.

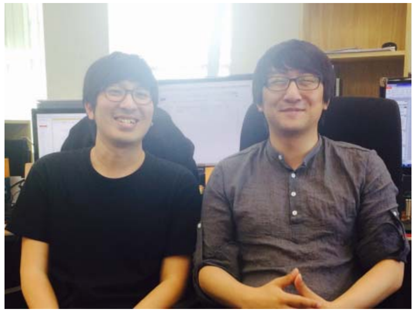
LEC(Logic Equivalence Checking) is strongly suggested as a means of the special test (Method 1).

Our target (Current working set) - the combination of Actel Libero Soc with Synopsys Synplify Pro has no LEC solution applicable.

In this case, we may need to develop a new customized solution.

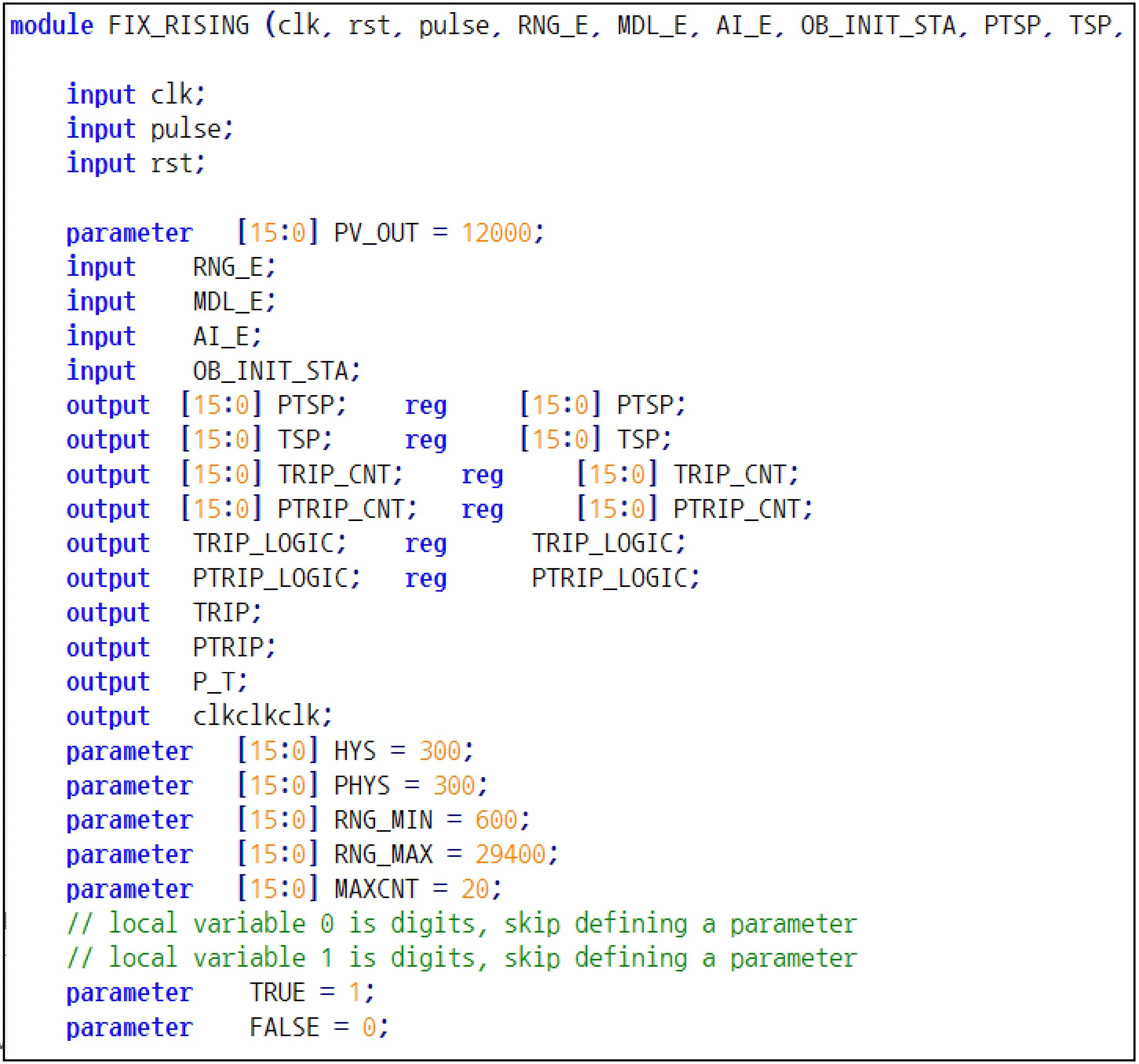
COTS SW dedication of indirect SW involves an in-depth analysis on the target's functionality and the techniques used to verify the functionality.

THANK YOU



Sejin Jung & Eui-Sub Kim

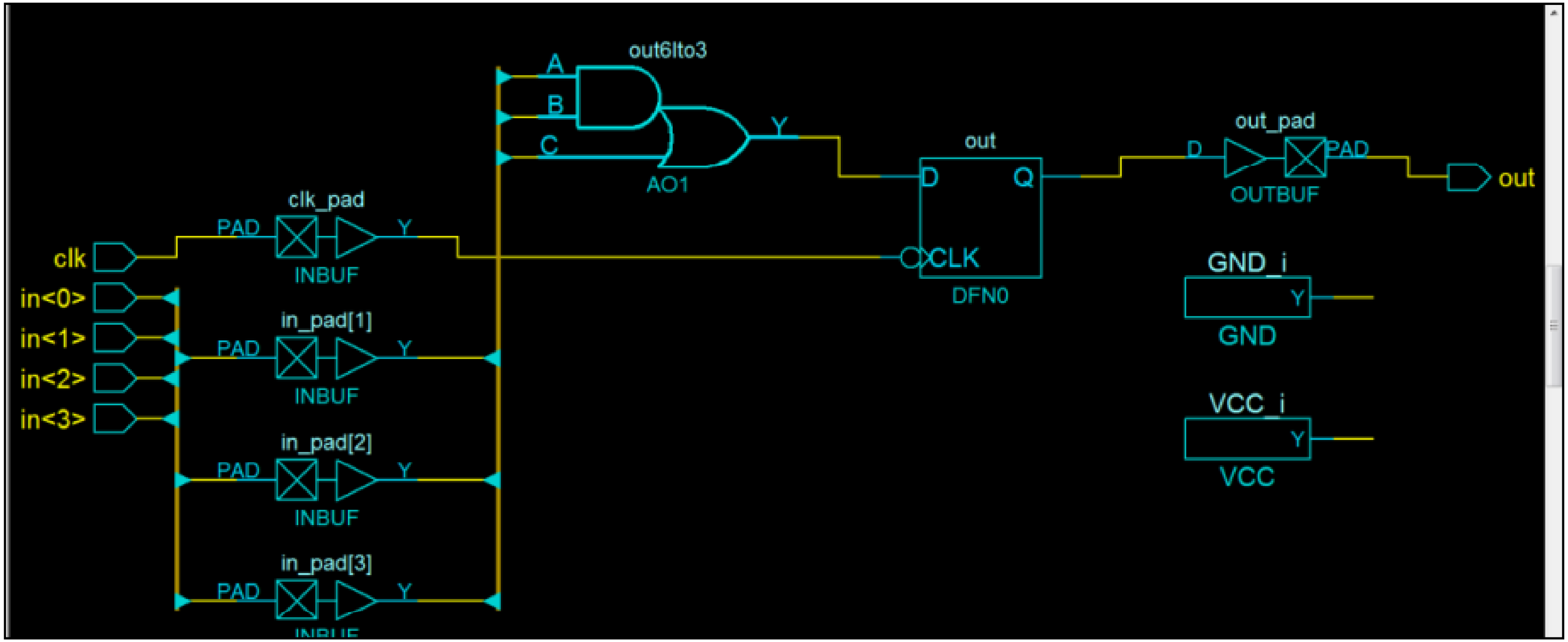
http://dslab.konkuk.ac.kr
jbyoo@konkuk.ac.kr

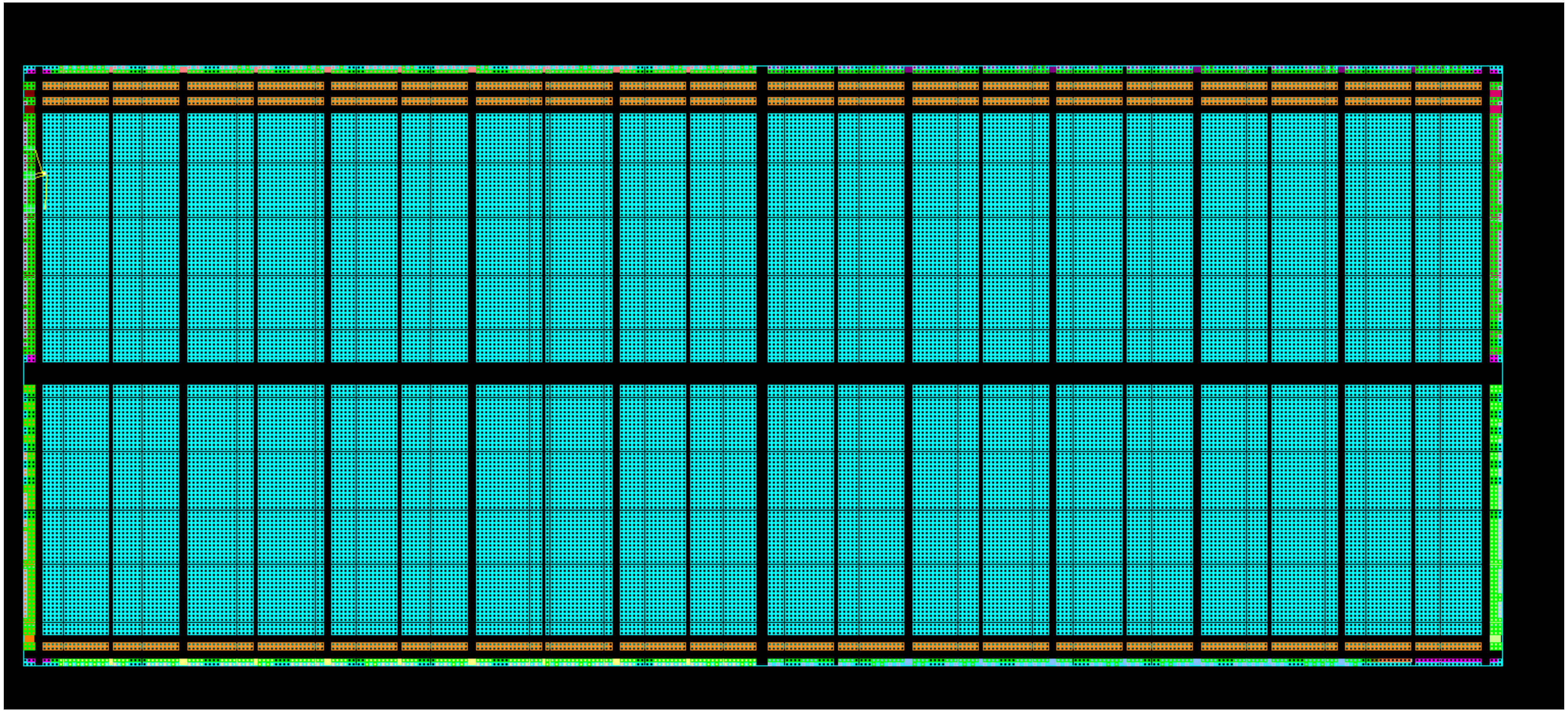


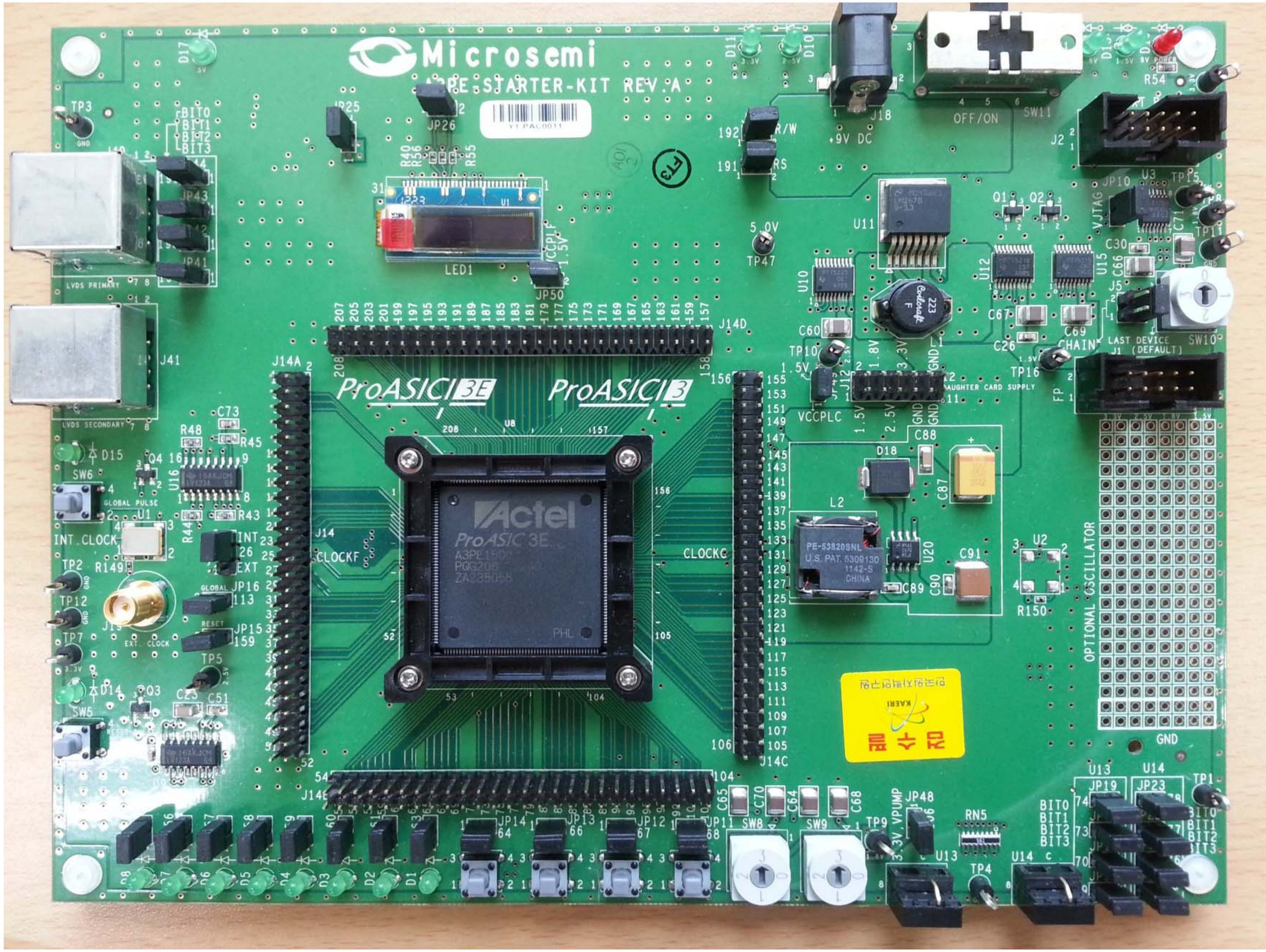
```
module FIX_RISING (clk, rst, pulse, RNG_E, MDL_E, AI_E, OB_INIT_STA, PTSP, TSP,

    input clk;
    input pulse;
    input rst;

    parameter   [15:0] PV_OUT = 12000;
    input    RNG_E;
    input    MDL_E;
    input    AI_E;
    input    OB_INIT_STA;
    output  [15:0] PTSP;    reg     [15:0] PTSP;
    output  [15:0] TSP;     reg     [15:0] TSP;
    output  [15:0] TRIP_CNT;    reg     [15:0] TRIP_CNT;
    output  [15:0] PTRIP_CNT;   reg     [15:0] PTRIP_CNT;
    output   TRIP_LOGIC;    reg      TRIP_LOGIC;
    output   PTRIP_LOGIC;   reg      PTRIP_LOGIC;
    output   TRIP;
    output   PTRIP;
    output   P_T;
    output   clkclkclk;
    parameter   [15:0] HYS = 300;
    parameter   [15:0] PHYS = 300;
    parameter   [15:0] RNG_MIN = 600;
    parameter   [15:0] RNG_MAX = 29400;
    parameter   [15:0] MAXCNT = 20;
    // local variable 0 is digits, skip defining a parameter
    // local variable 1 is digits, skip defining a parameter
    parameter    TRUE = 1;
    parameter    FALSE = 0;
```


out6lto3
A
B
C
AO1
Y
out
D
Q
CLK
DFN0
out_pad
D
PAD
OUTBUF
out
clk_pad
PAD
Y
INBUF
clk
in<0>
in<1>
in<2>
in<3>
in_pad[1]
PAD
Y
INBUF
in_pad[2]
PAD
Y
INBUF
in_pad[3]
PAD
Y
INBUF
GND_i
Y
GND
VCC_i
Y
VCC




Microsemi
A3PE STARTER-KIT REV.A
ProASIC3E
ProASIC3
Actel
ProASIC3E
LED1
INT.CLOCK
OPTIONAL OSCILLATOR
OFF/ON
9V DC
CLOCKC
CLOCKF
J14A
J14C
J14D
J14E
SW5
SW6
SW8
SW9
SW10
SW11