KNS 2016 Spring Meeting
05.11~13 JEJU ICC

KU KONKUK UNIVERSITY

A MC/DC and Toggle Coverage Measurement Tool for FBD Program Simulation

Eui-Sub Kim, Sejin Jung, Jaeyeob Kim, Junbeom Yoo

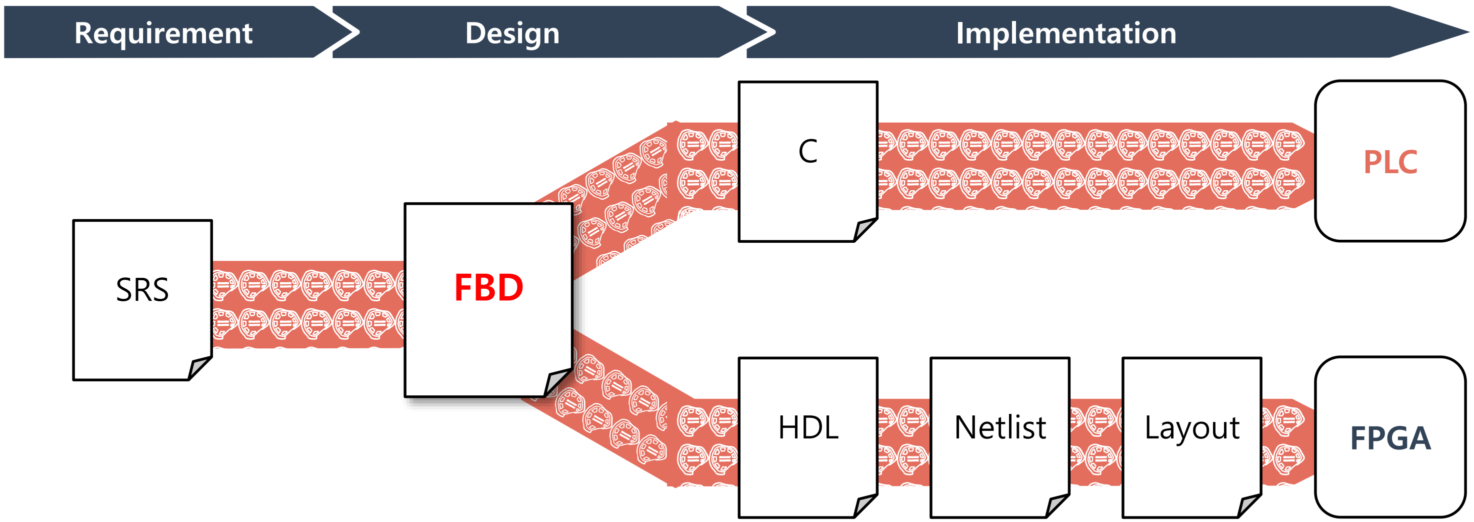
Dependable Software Laboratory
KONKUK University

2016.05.13

DEPENDABLE SOFTWARE LABORATORY

Functional Verification of FBD

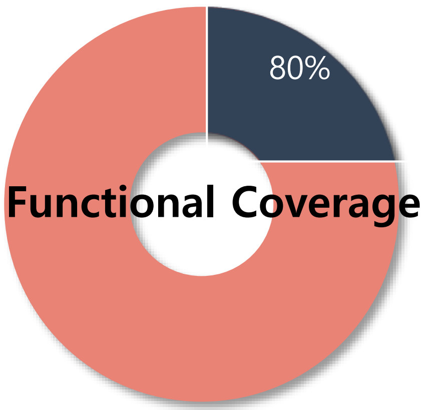
- Functional verification of FBD (Function Block Diagram) is important
  - FBD is a design model for PLC (and FPGA in the NuDE framework)
  - Detection errors early (design phase) ➔ Can reduce costs and increase quality
  - Software design errors are often only detected during final test or after delivery

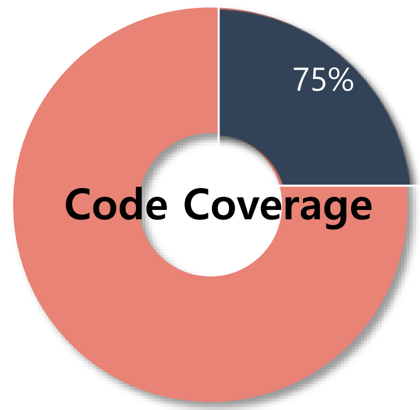


< The NuDE framework >

How Adequately the Testing has been Performed?

"Test Done = Test Plan Executed and All Codes Executed"





Functional Coverage

- = Requirements Coverage
- This coverage will be defined by the user
- User will define the coverage points for the functions to be covered
- 100% of functional coverage is always required

Code Coverage

- = Structural Coverage
- How many lines are executed, how many times expressions, branches executed, etc.
- Code coverage is collected by the simulation/testing tools.
- Users use code coverage to reach those corner cases which are not hit by the test cases.
  - Unfortunately, errors and bugs are often found in the corner cases.
- To assure a high quality of functional verification, code coverage is important as well as functional coverage

Introduction

- **We applied two code coverages to FBDs**
  - (1) Toggle coverage , (2) MC/DC coverage
  - Defined coverage criteria for FBD simulation
  - If the coverages is not 100%, it means that the verification may be insufficient or the FBD may have unintended errors or bugs.

- **We developed a set of supporting CASE tools**
  - Developed two CASE tools 'FBDSim' and 'FBDCover'
  - Can simulate FBDs and measure the code coverages of the FBD simulation
  - Objective : measuring the coverages during simulation (a sequential/continuous operation environment, not a single execution)

Toggle Coverage & MC/DC Coverage

• Toggle Coverage

- One of the oldest measurements of coverage in hardware design
- Measures the bits of logic that have toggled during simulation
- Can be measured in logic simulation
- Ex) 1-to-0 and 0-to-1 → 100% toggle coverage

• MC/DC Coverage

- Control flow-based structural coverage of the most highest level, in practice
- Widely applied to C/Java programs

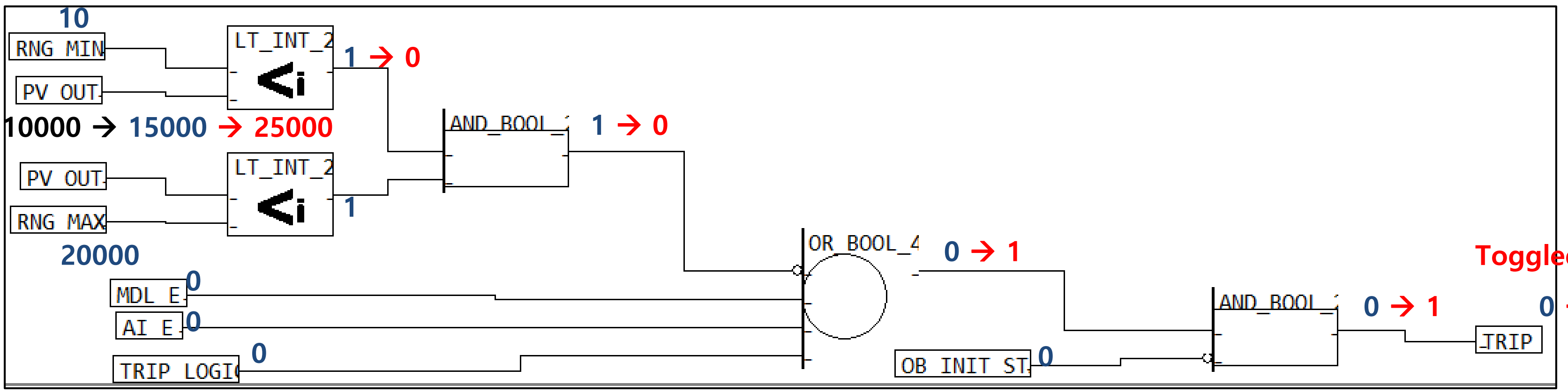
Case #	A	B	OUT	A	B
1	T	T	T	O	O
2	T	F	F		O
3	F	T	F	O	
4	F	F	F		

100% MC/DC
➔ (T,T), (F,T), (T,F)

Toggle Coverage in FBDs

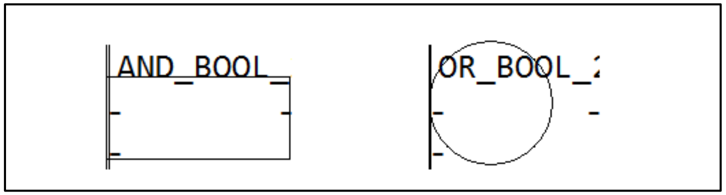
- **Toggle Coverage in the FBD**

- Two application targets : (1) Output toggle, (2) Block toggle
  - (1) Output toggle : an output is toggle during the simulation
  - (2) Block toggle : a function block's output is toggle during the simulation
- Ex) If an output is not toggled, we may doubt that
  - the output variable is not tested → simulation may be insufficient.
  - the output variable is unreachable → the logic may have dead codes → a logic-fix requires



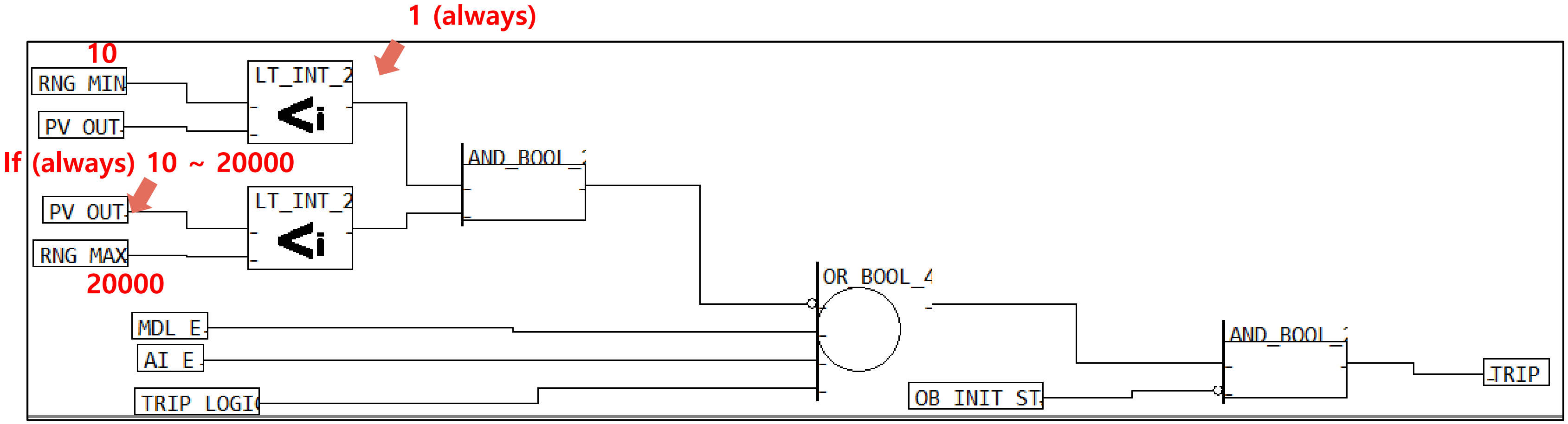
MC/DC Coverage in FBDs

- **MC/DC Coverage in the FBD**
  - Based on the typical MC/DC principle
  - Measure the MC/DC coverage of a function block
  - Ex) If any block does not cover 100% MC/DC coverage, we may doubt that
    - the block is not tested → simulation may be insufficient
    - the block is unreachable → the logic may have dead codes → a logic-fix requires



	Inputs	MC/DC
AND	IN1, IN2	(0,1) (1,0) (1,1)
OR	IN1, IN2	(0,0) (0,1) (1,0)

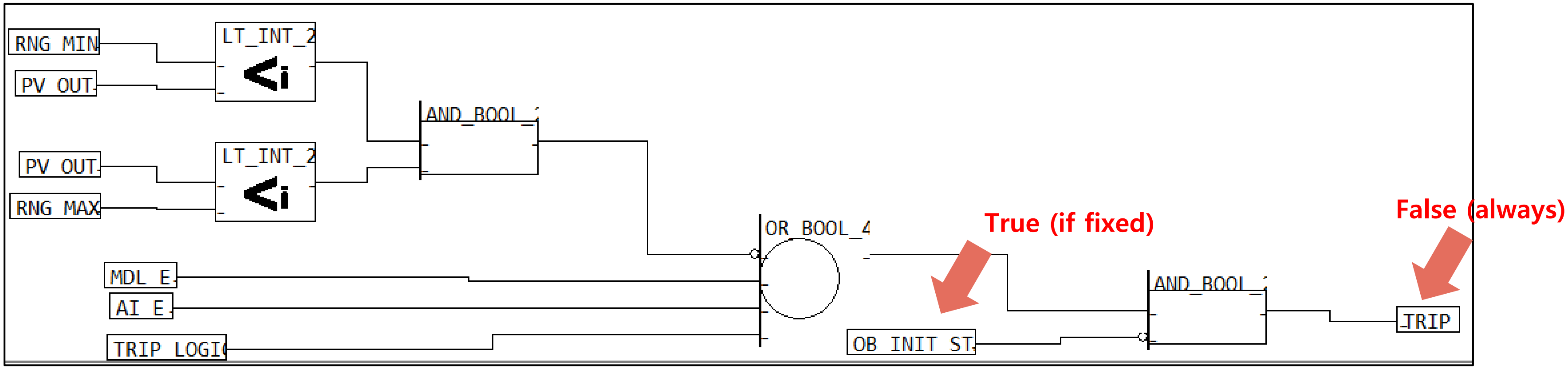
Block Toggle Coverage (An Example of Insufficient Simulation)



- **Insufficient simulation ?**
- **If the variable 'PV_OUT' is always located between MIN and MAX,**
  - The block 'LT_INT_2' is never toggled. → 0% toggle coverage
- **User can add more test cases to toggle the function block**
  - Ex) PV_OUT = 0~9 and next PV_OUT > 10 (again)

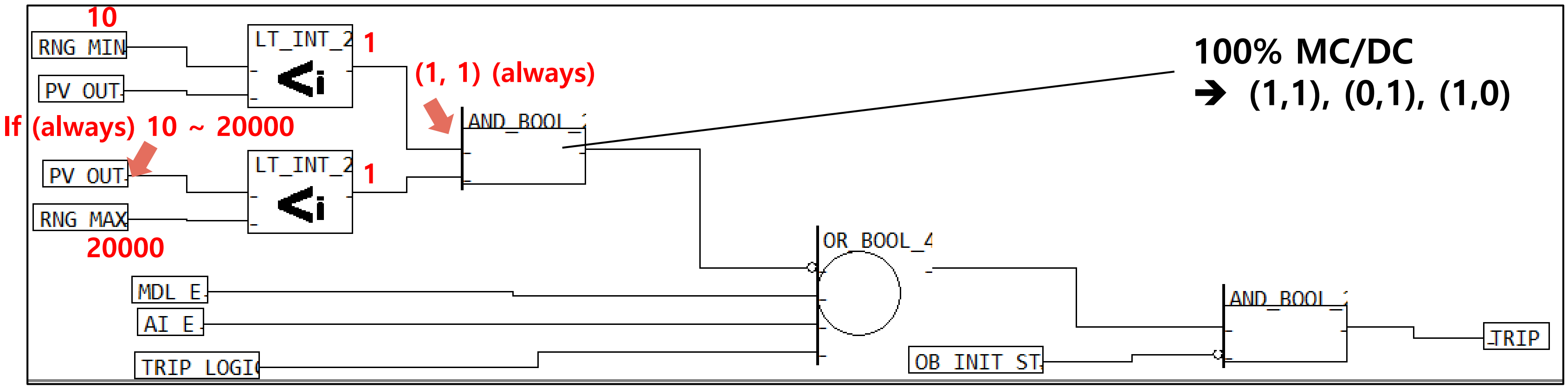
    (0 → 1) (1 → 0)

Output Toggle Coverage (An Example of Unreachable Code)



- **Unreachable ?**
- If the variable 'OB_INT_ST' is always true?
  - The output variable 'TRIP' is never toggled. → 0% toggle coverage
- User can modify the logic
  - Ex) remove 'AND_BOOL' block
  - Ex) change the 'OB_INT_ST' variable (i.e., constant) to an (simulation) input variable

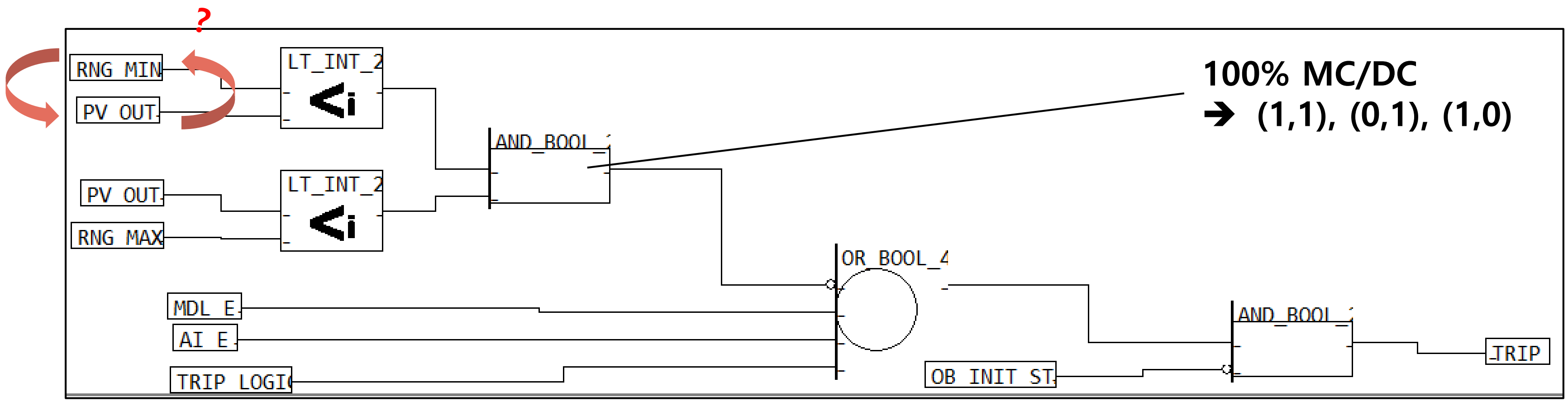
MC/DC Coverage (An Example of Insufficient Simulation)



- **Insufficient simulation ?**
- If the variable 'PV_OUT' is always located between MIN and MAX,
  - The input of 'AND_BOOL' is always (1, 1) → 33% MC/DC coverage
- User can add more test cases to toggle the function block
  - Ex) PV_OUT = 0~9 and PV_OUT = over 20000

    (0, 1) (1, 0)

MC/DC Coverage (An Example of Unreachable Code)

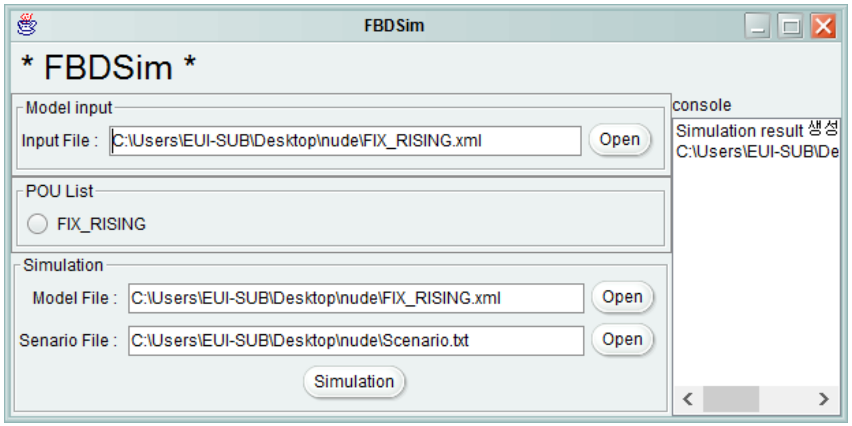


- **Unreachable ?**
- If two inputs of the upper 'LT_INT_2' are exchanged (due to a logic error)
  - It means "PV_OUT < MIN and PV_OUT < MAX"
  - The condition (1, 0) is never generated. → The max MC/DC is 66%
- User may have a chance to identify the (hypothetical) error and fix the logic

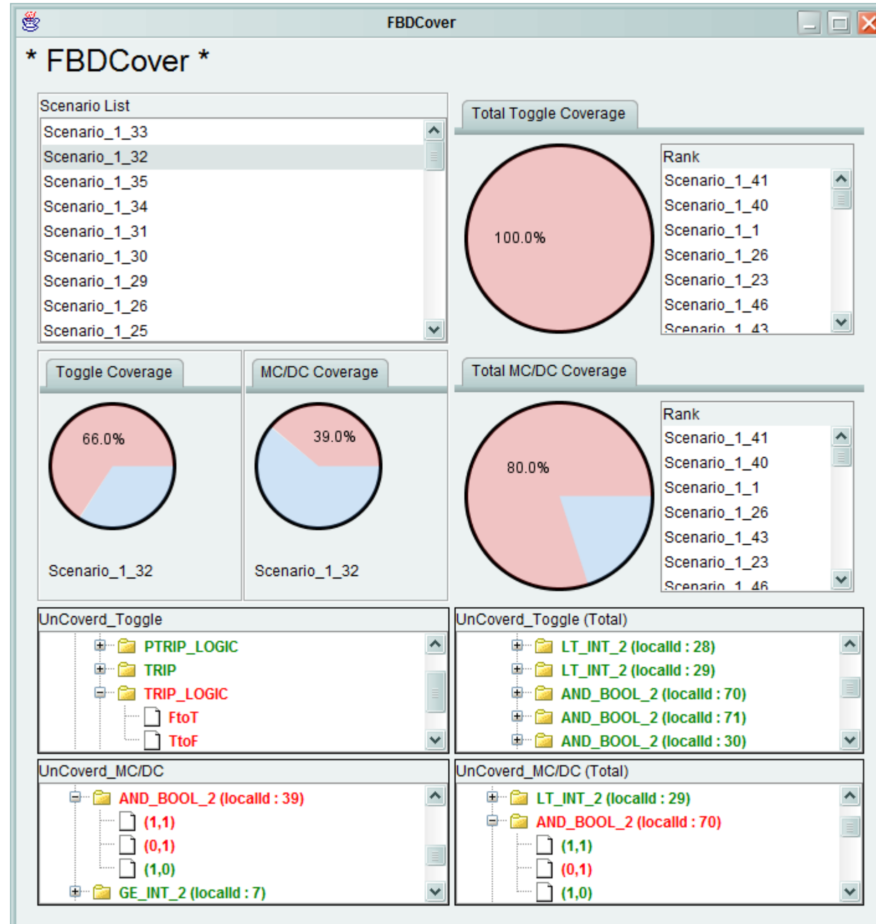
THE TOOL DEVELOPMENT

The Tool Development

- **We develop two tools: (1) FBDSim (2) FBDCover**



FBDSim

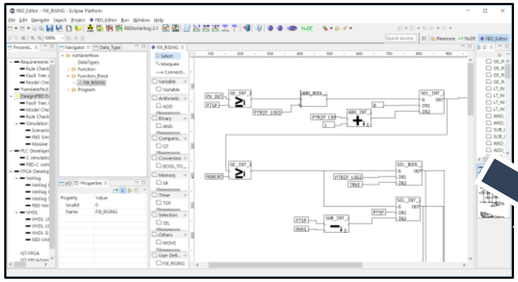


FBDCover

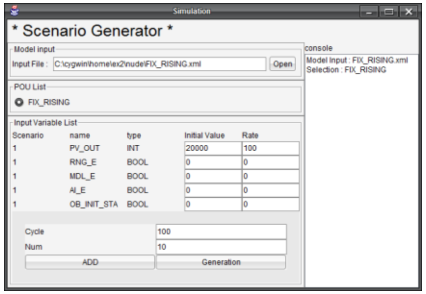
FBDSim

• FBD Simulation Tool

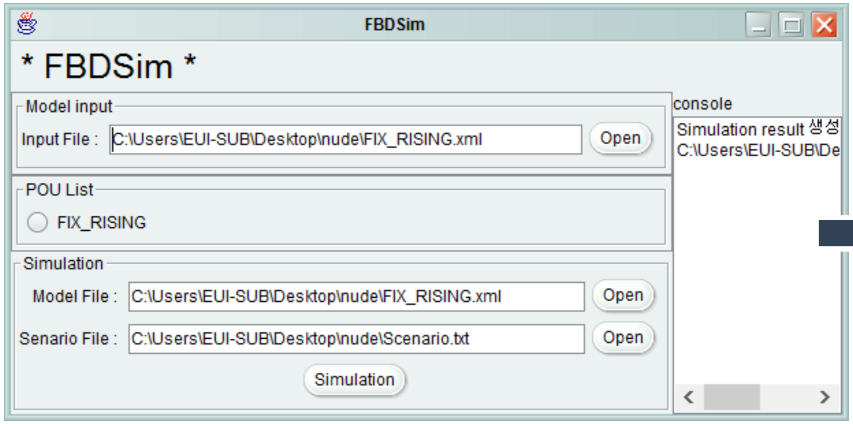
- Input: (1) FBD program in PLCopen TC6 XML format , (2) Simulation scenario
- Output: (1) Simulation result, (2) Coverage information
- Embedded in FBD Editor



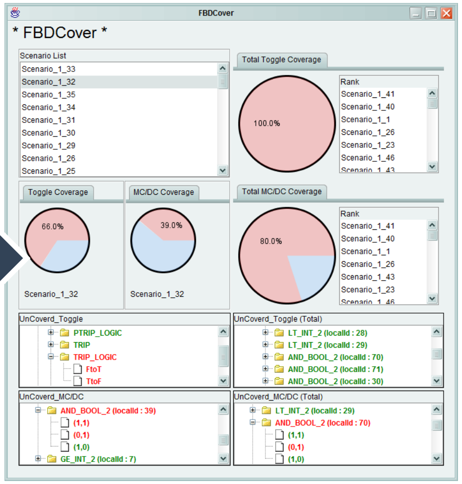
FBD Editor



Scenario Generator



FBDSim



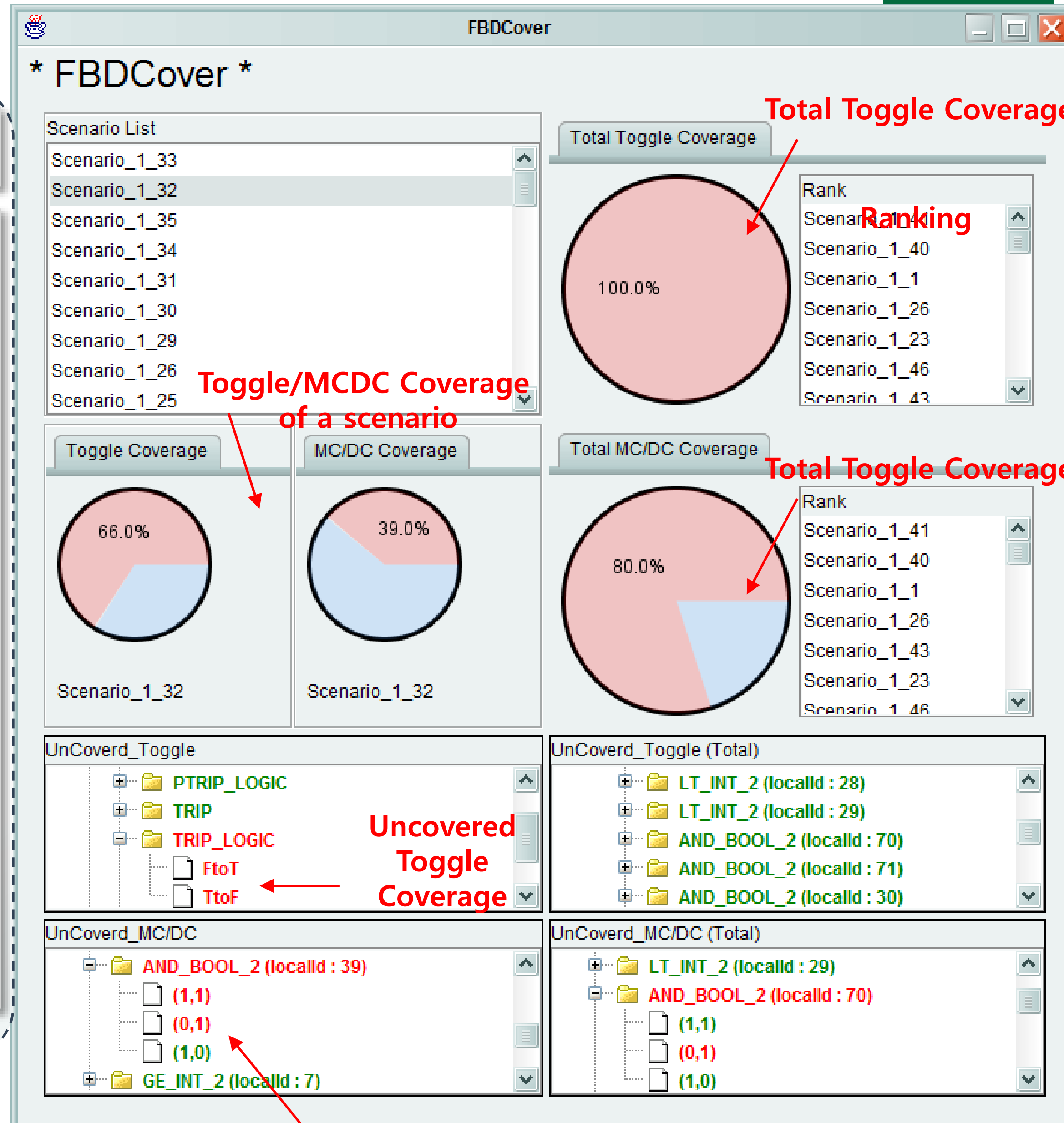
FBDCover

FBDCover

- **Coverage Measurement Tool**

- **Input:**
  - Coverage information from FBDSim
- **Output:**
  - Graphical coverage result
- **Embedded in FBD Editor**

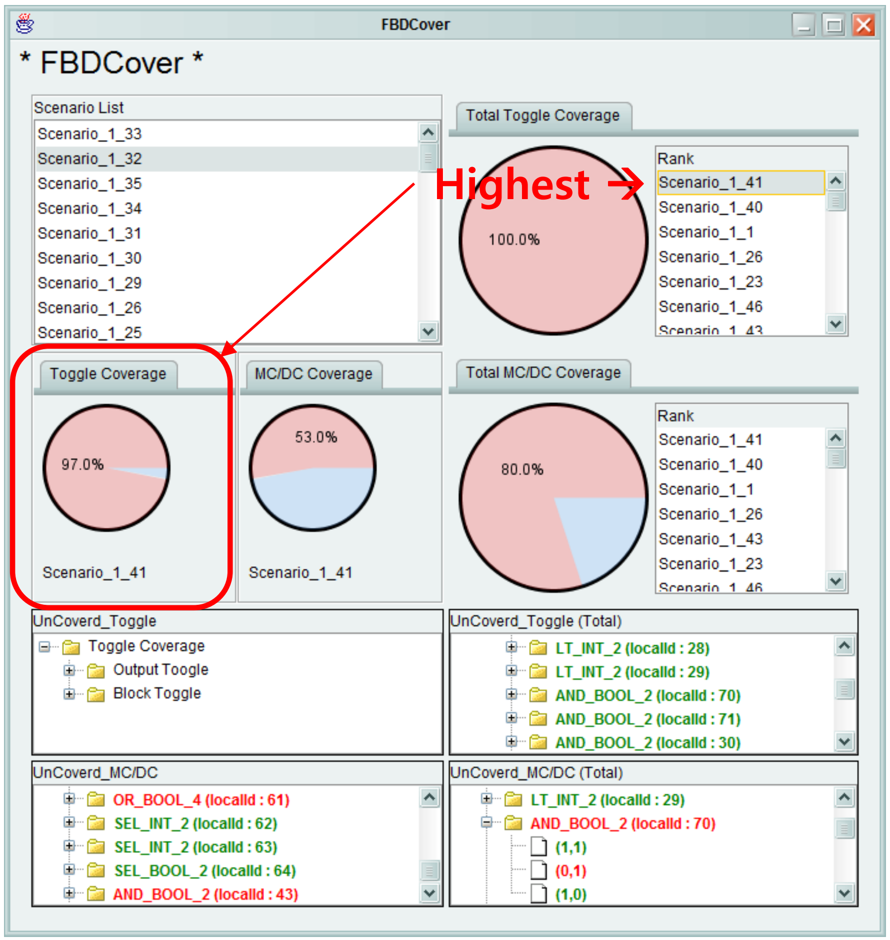
- **Notifies ranks of scenarios**
- **Notifies uncovered elements**

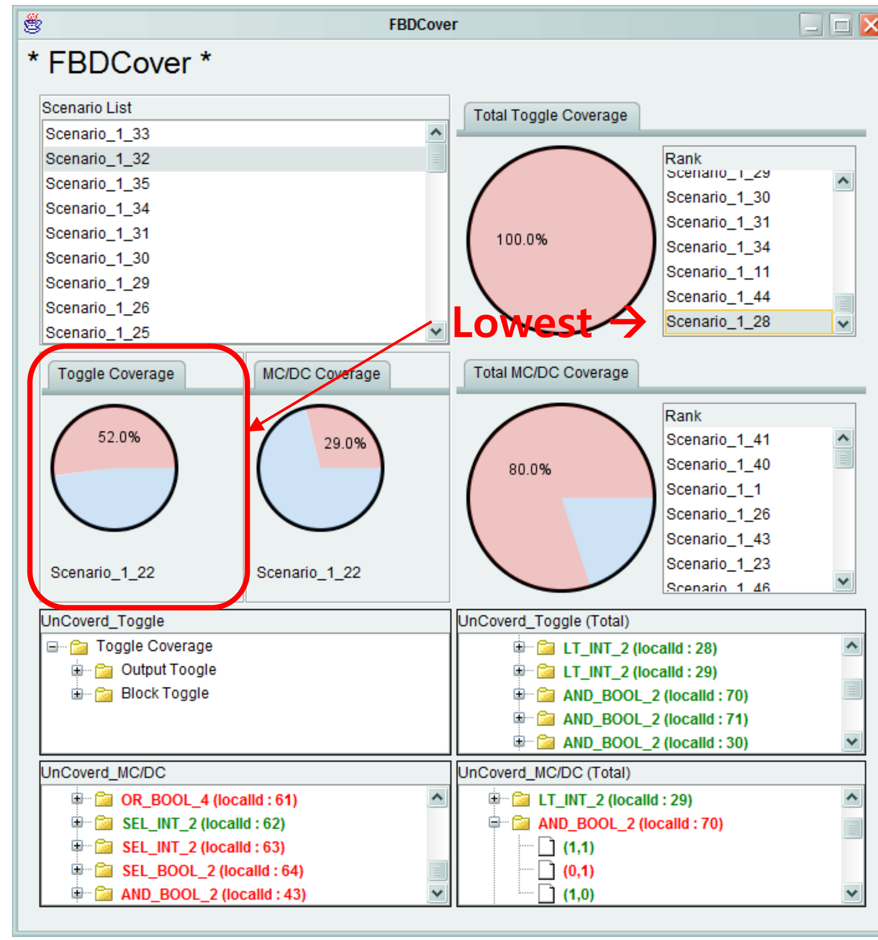


Uncovered MC/DC condition

Ranks of FBDCover

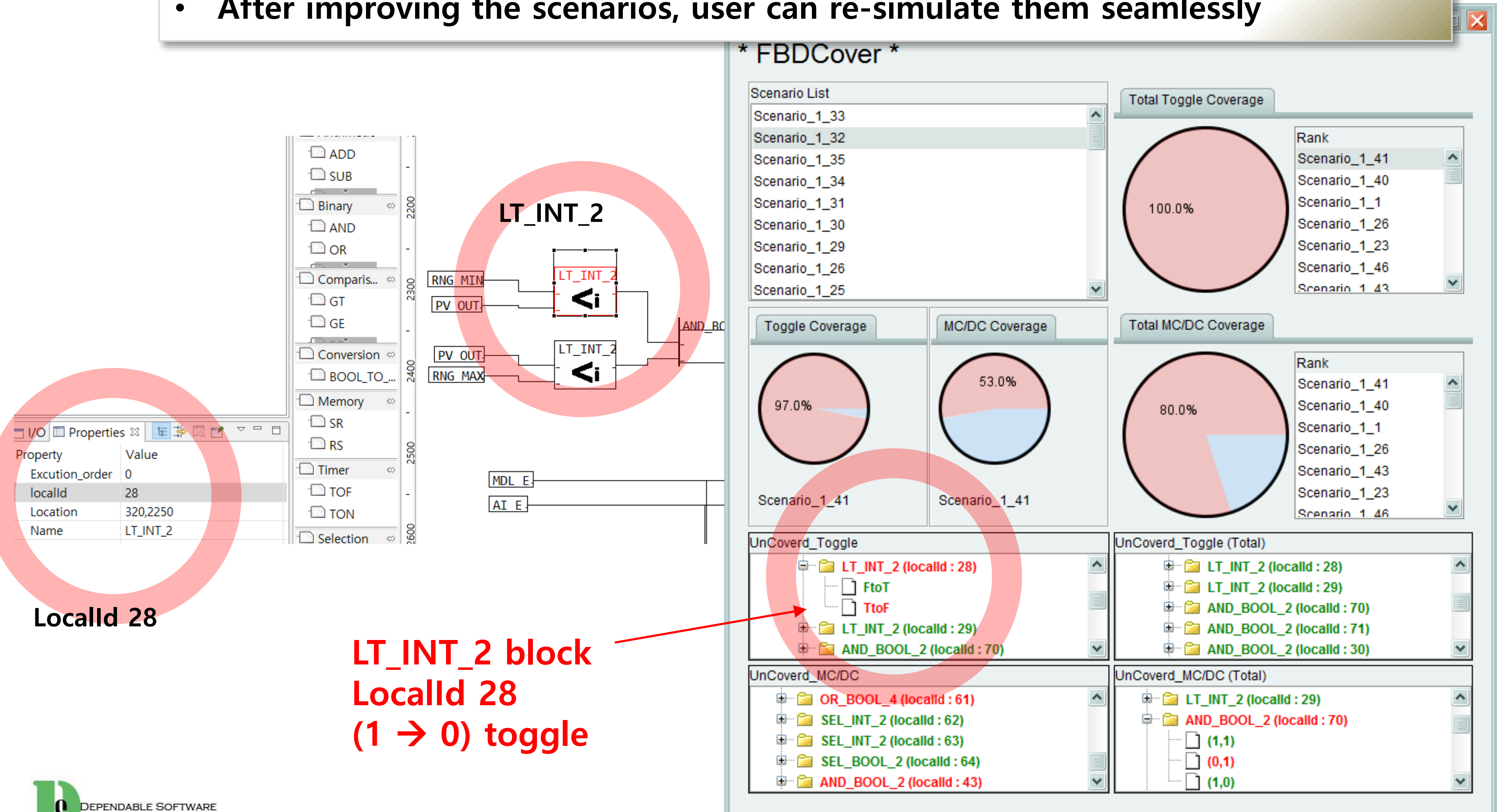
- **Highest rank scenario vs. Lowest rank scenario of toggle coverage**
  - **Provide valuable information to improve simulation scenarios**





Uncovered Elements of FBDCover

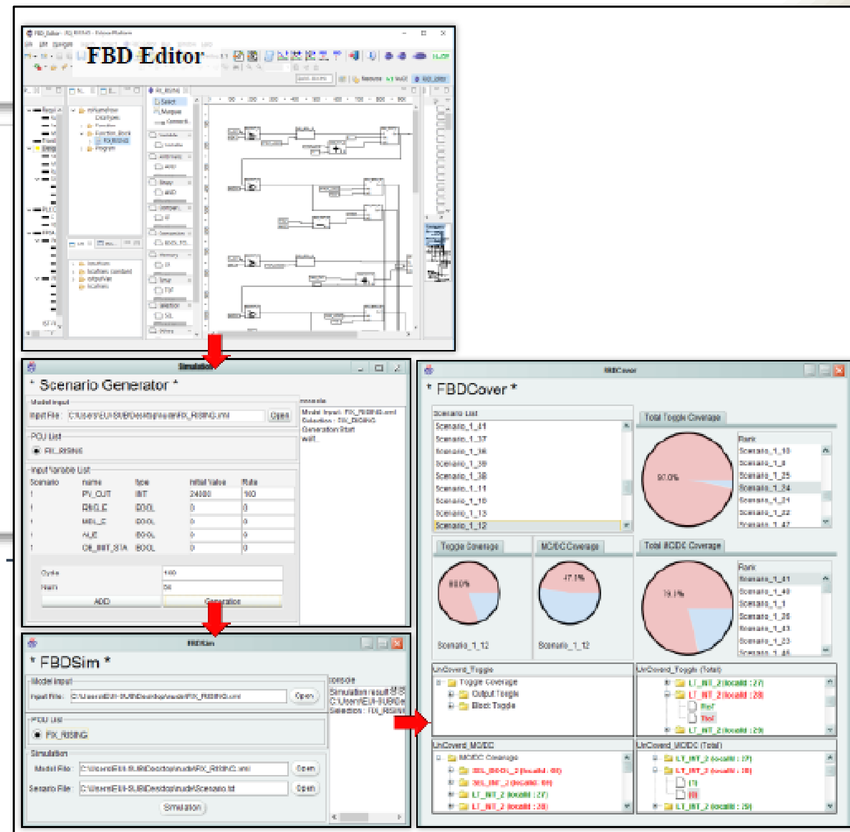
- **Notify elements which are not simulated**
  - **After improving the scenarios, user can re-simulate them seamlessly**



CASE STUDY

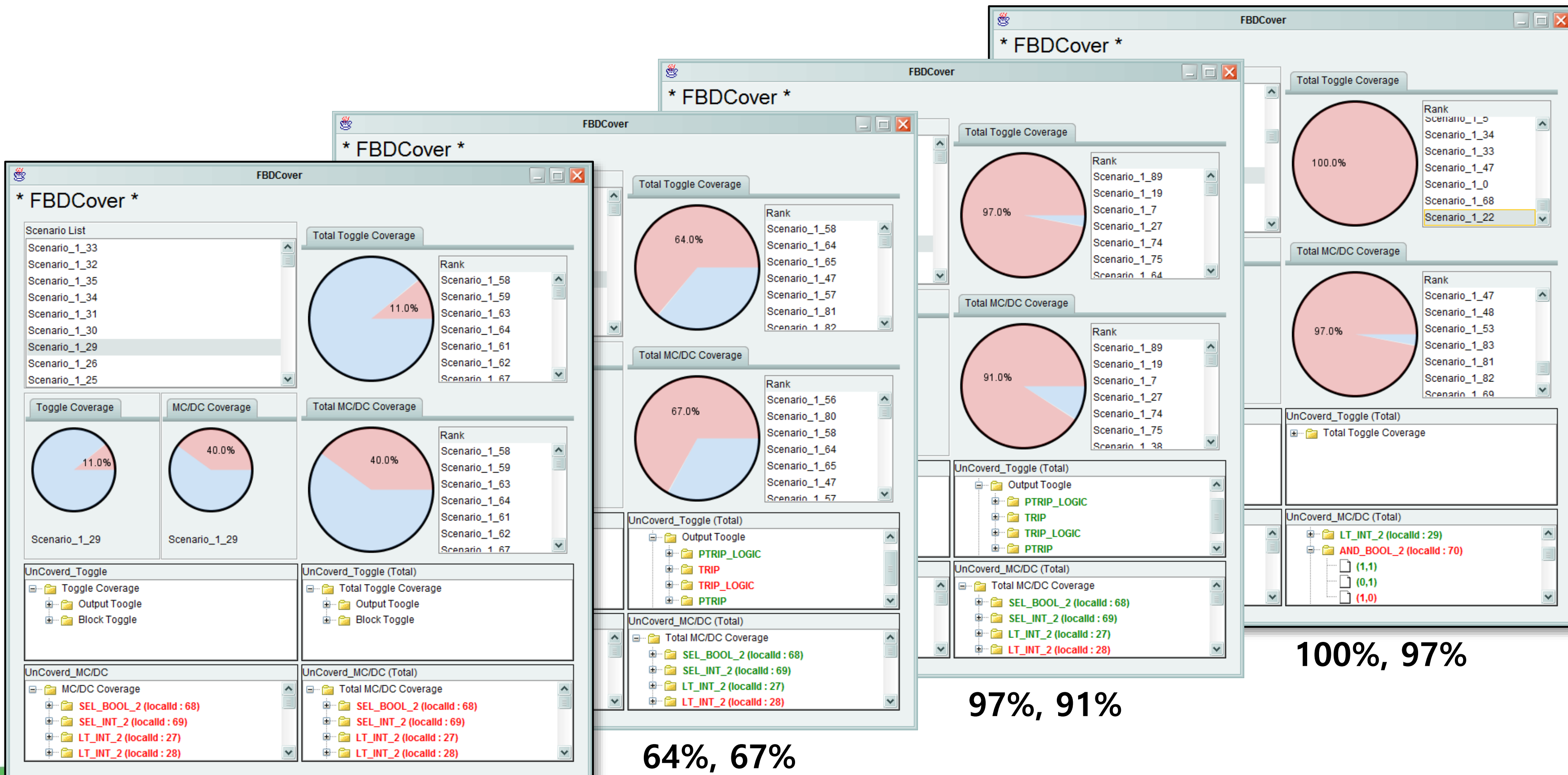
Case Study

- **We performed a case study with an example replicating a KNICS APR-1400 RPS BP**
  - ‘FBDSim’ automatically simulates a set of FBD scenarios and checks toggle and MC/DC coverage
  - We used our tool-set of
    - FBD Editor
    - Scenario Generator
    - FBDSim
    - FBDCover



Case Study

- We found uncovered elements and improved the scenarios and then re-simulated with the scenarios.



11%, 48%

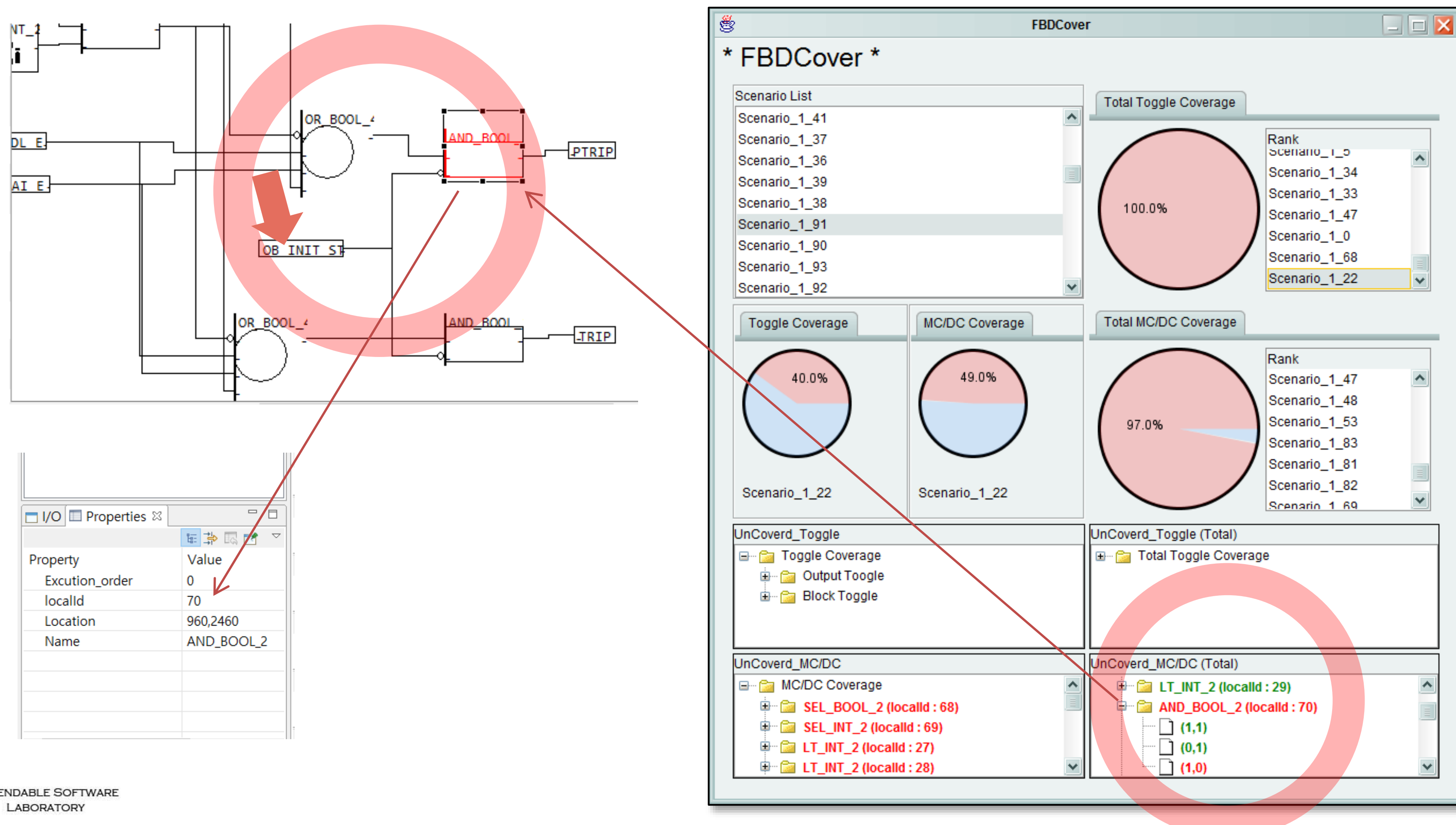
64%, 67%

97%, 91%

100%, 97%

Case Study (Example)

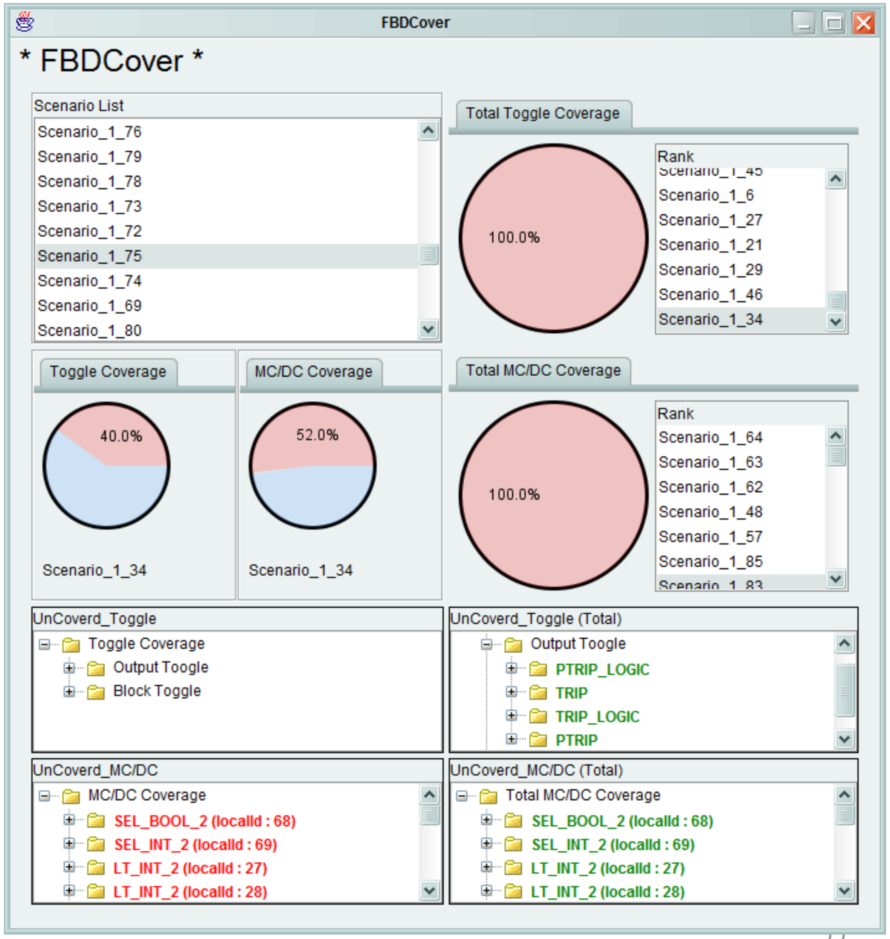
- **We found that we missed to simulate the bypass, with the MC/DC coverage.**



Case Study (Example)

- **Finally, we were able to get 100% toggle and MC/DC coverage.**

- Of course, it is not sufficient to assure that the program is free from bug or error.
- It is possible to fail with 100% code coverage.
- However, we always try to improve on the quality of verification with every possible means.
- The tool is helpful because it notify engineers about that there are uncovered elements.
  - The uncovered elements imply that the simulation is not sufficient or the FBD has unintended errors or bugs.



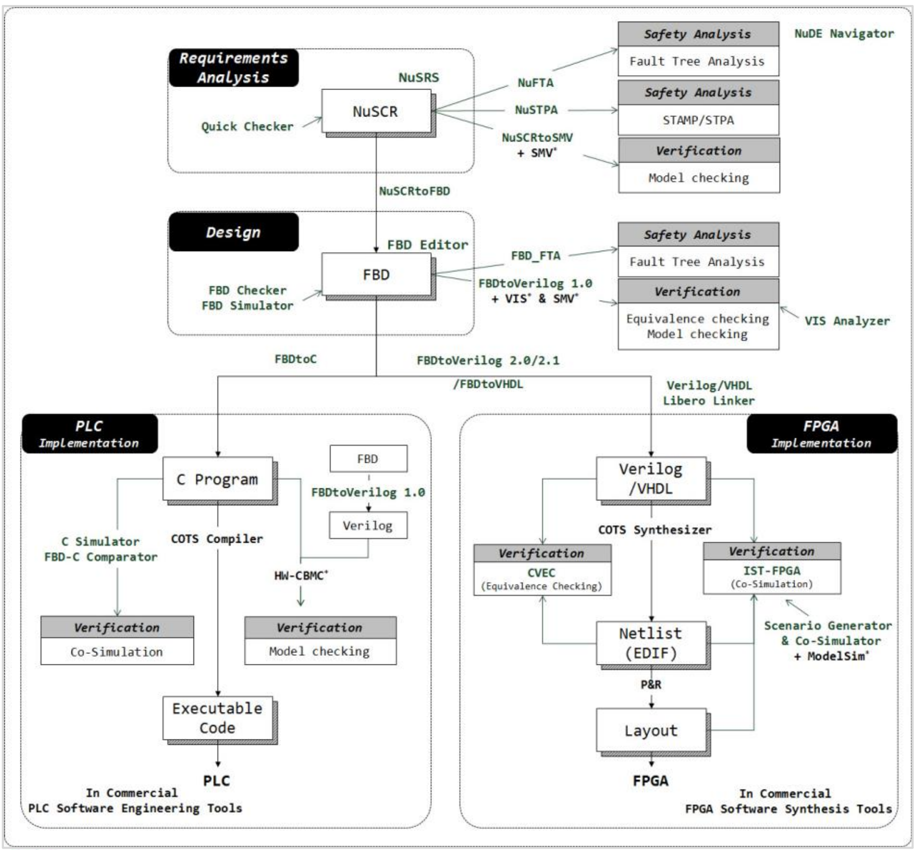
100%, 100%

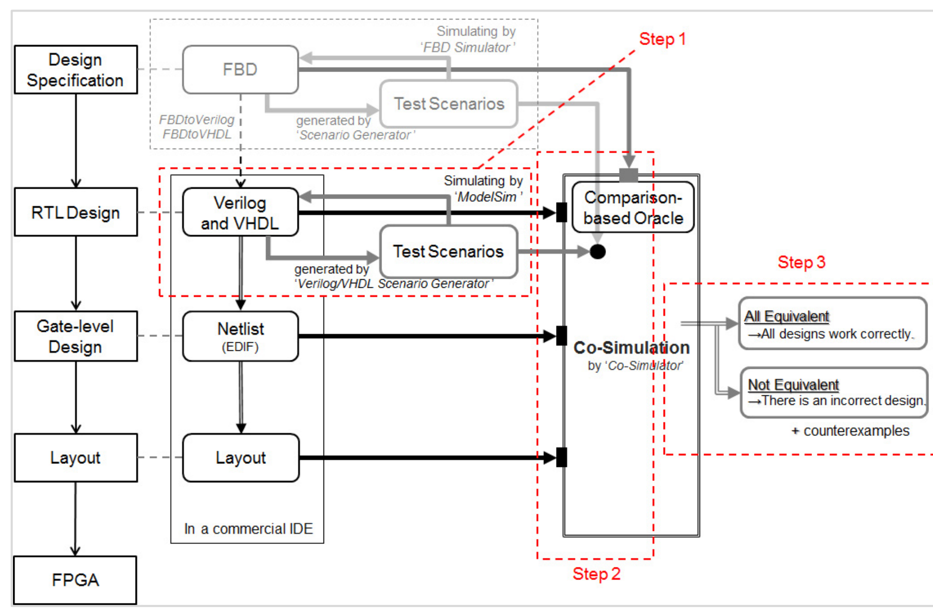
Conclusions and Future Work

- **We applied toggle and MC/DC coverage to the FBD.**
  - If the coverages are not 100%, user should analyze whether it is reasonable.
  - If it is not reasonable, it means that the simulation may be insufficient or the logic may have unintended errors or bugs.
  - We are trying to evaluate the efficiency/applicability of the coverages proposed.
  - All condition coverage is also applicable.
- **We developed two CASE tools.**
  - We developed two CASE tools 'FBDSim' and 'FBDCover'
  - We can simulate the FBD and measure the coverages of the simulation
  - It produces a rank of scenarios and uncovered elements.

Conclusions and Future Work

- **We are now planning to extend the coverage technique and tools to develop a full coverage-based scenario generation tool.**
  - **NuDE 2.0**
  - **IST-FPGA**





Jaeyeob Kim, Eui-Sub Kim, Junbeom Yoo, Young Jun Lee and Jong-Gyun Choi,
"An Integrated Software Testing Framework for FPGA-based Controllers in Nuclear Power Plants,"
Nuclear Engineering and Technology, Vol.48, No.2, pp.470-481, 2016.

THANK YOU

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