Submitted to IEEE Software 2009
- Special issue on "Software Development for Embedded Systems"

Formal Modeling and Verification of Safety-Critical Software implemented in PLC

JUNBEOM YOO

KONKUK University, Korea

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

Other Authors

Sungdeok Cha
- Professor in Korea University

Enukyoung Jee
- PhD Candidate in KAIST

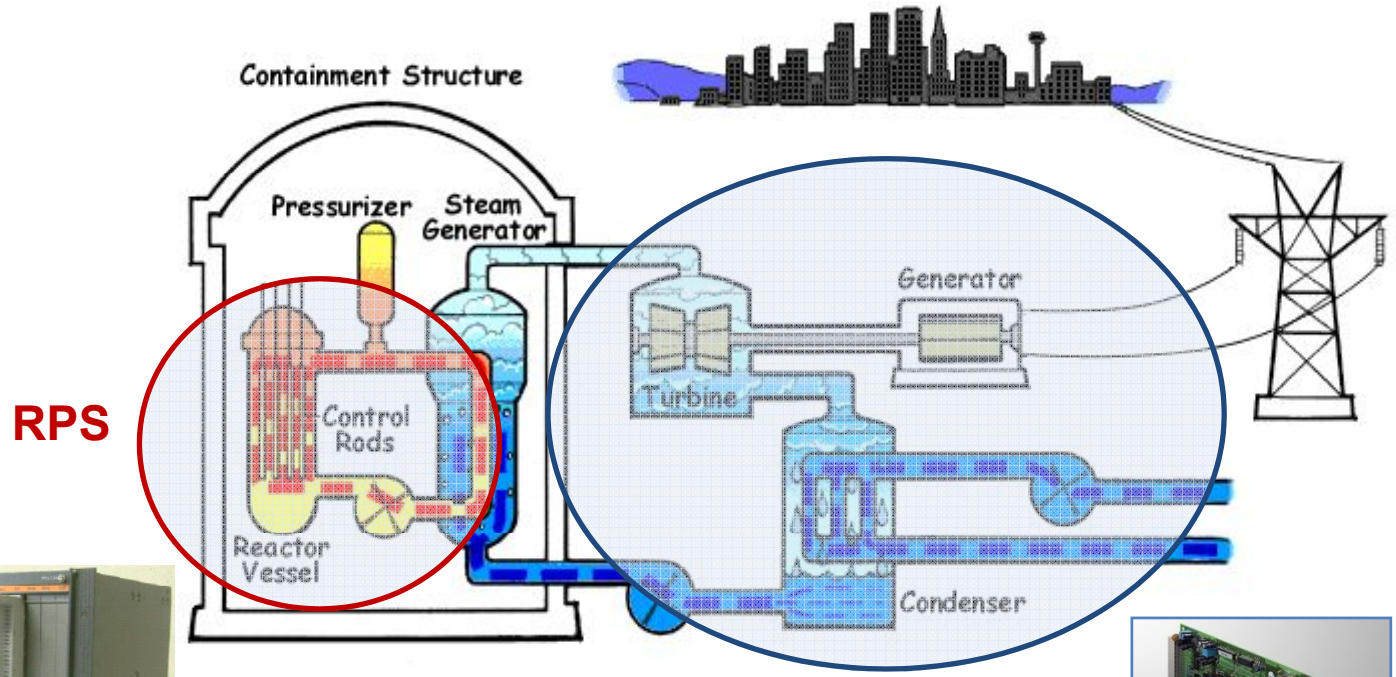
Contents

Introduction

1. Safety-Critical Software in Nuclear Power Plants
2. Software Development Process (Existing vs. Proposed)

Safety-Critical Software in Nuclear Power Plants

- RPS (Reactor Protection System)
- ESF-CCS (Engineering Safety Features Component Control System)

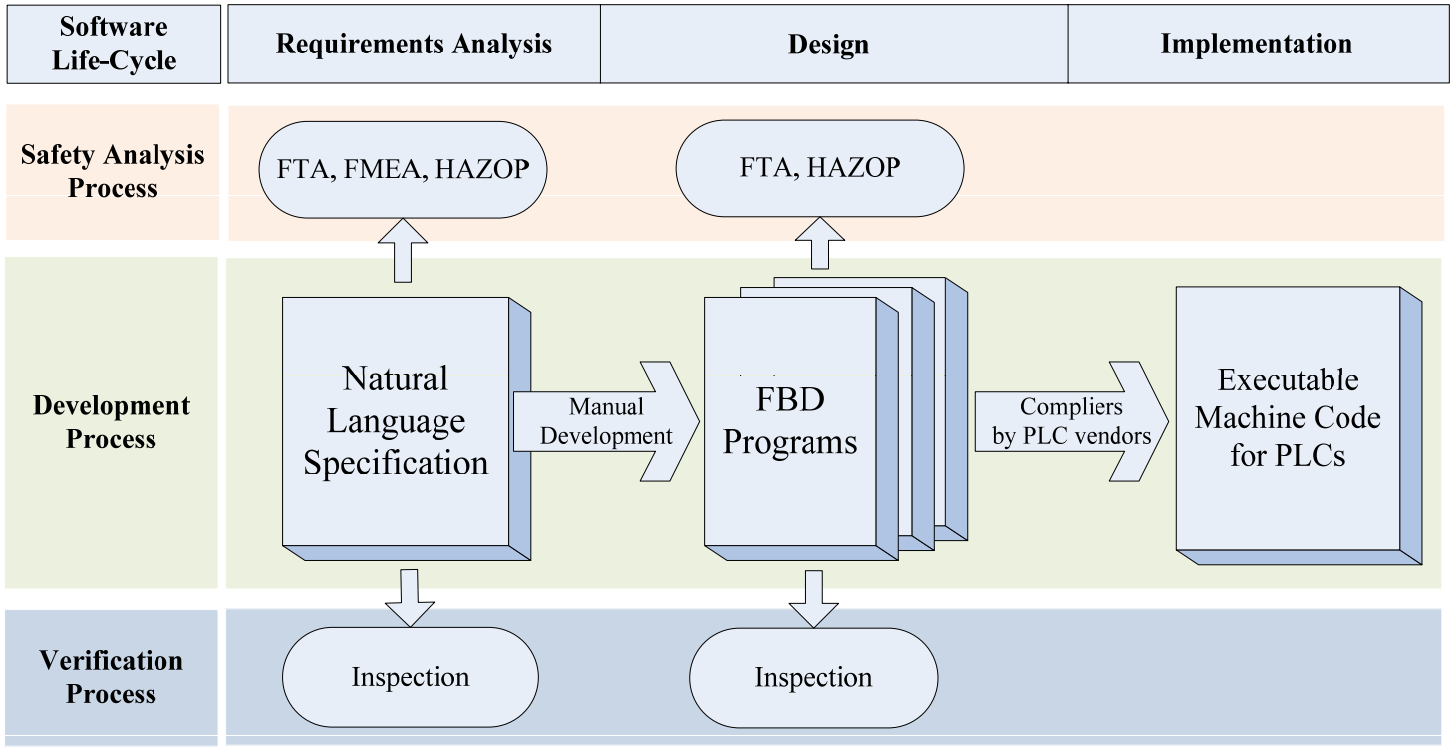


PLC
(Programmable Logic Controller)

DCS
(Distributed Computing System)

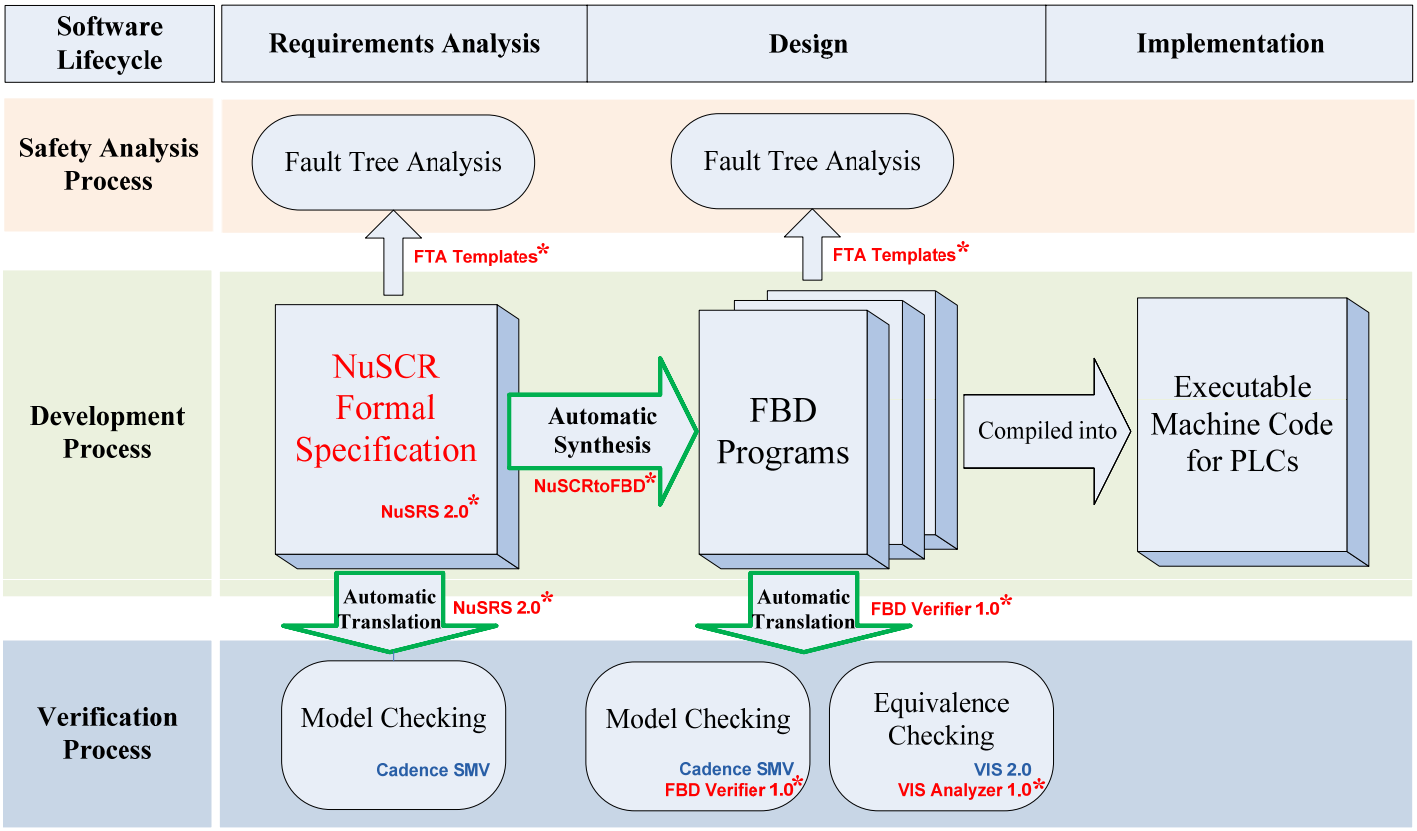
Existing Software Development Process

- For Most NPPs in Korea (e.g. Wolsung NPP)



Proposed Software Development Process

- For KNICS RPS for APR-1400 [1] (http://www.knics.re.kr)
  - APR-1400 : Next generation nuclear reactor being developed in Korea



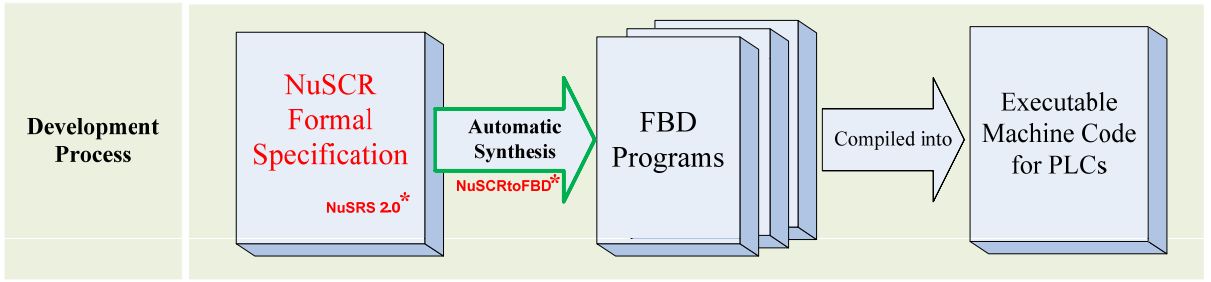
* : Our group's effort

Software Development Process for NPPs

1. Development Process
2. Verification Process
3. Safety-Analysis Process

Development Process

1. Formal Requirements Specification
2. Automatic Design Synthesis

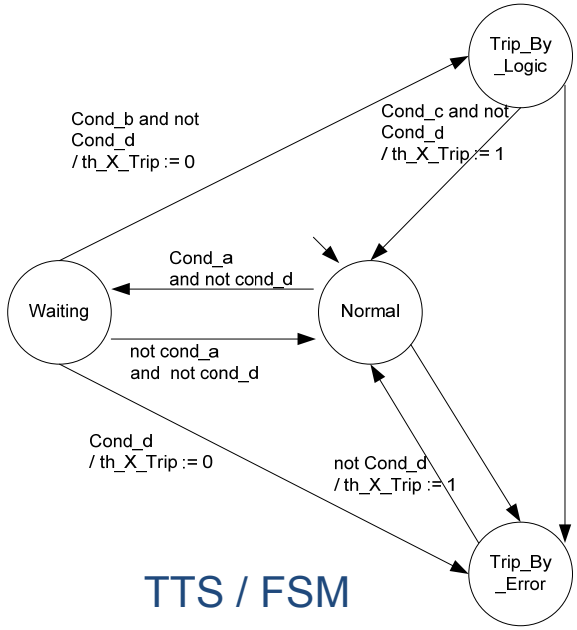


1. Formal Requirements Specification

- NuSCR [3]
  - Formal requirements specification language
  - Customized SCR [2] for nuclear applications
    - Listened to opinions offered by domain experts
  - 4 constructs
    - SDT (Structured Decision Table)
    - FSM (Finite State Machine)
    - TTS (Timed Transition System)
    - FOD (Function Overview Diagram)

Conditions		
k_X_MIN <= f_X <= k_X_MAX	T	F
Actions		
f_X_Valid := 0	X	
f_X_Valid := 1		X

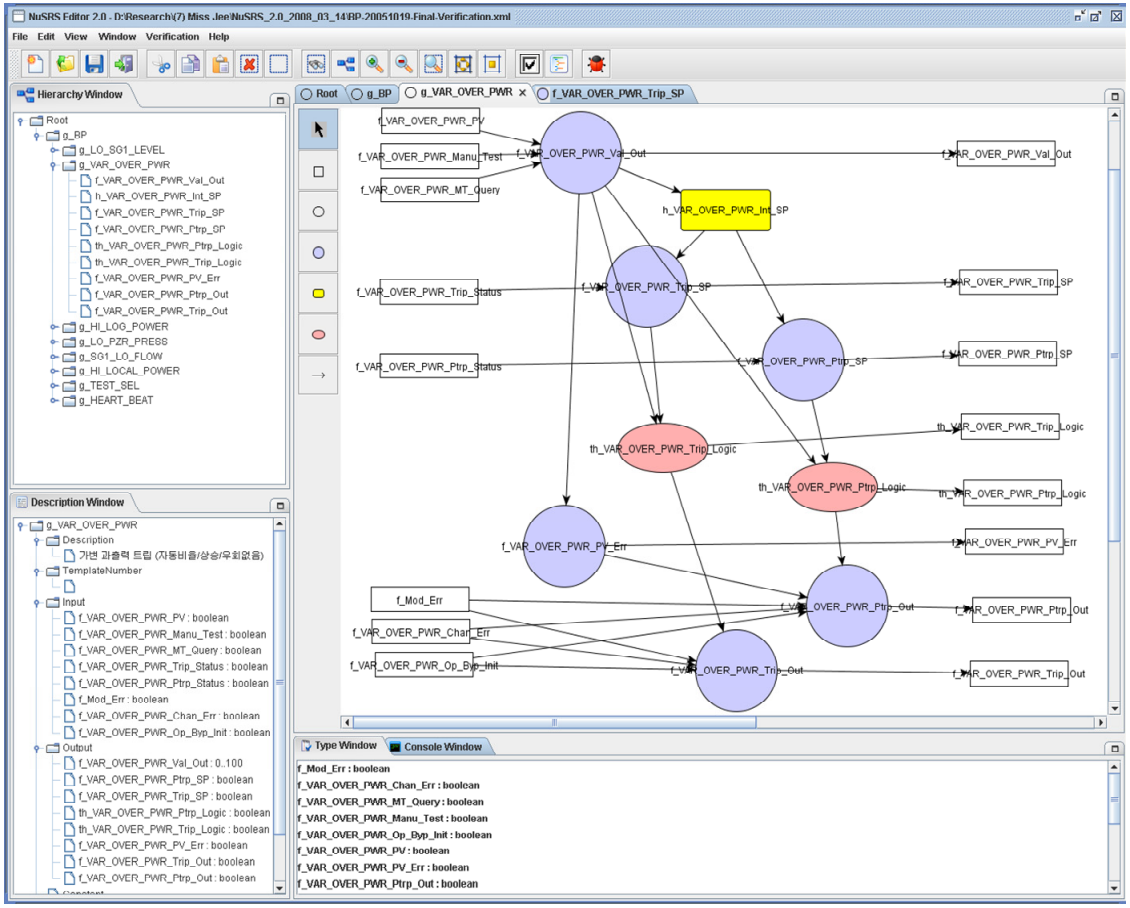
SDT



TTS / FSM

1. Formal Requirements Specification

- NuSRS (ver 2.0)
  - CASE tool supporting
    - NuSCR specification
    - Self-Checking (on-going)
    - SMV program translation
      (NuSCR → SMV)
    - SMV verification
      (CTL Model Checking)
  - Case Study
    - KNICS-RPS-SRS101, Rev,00, 2003.
      (by NuSRS 1.0)
    - KNICS-RPS-SVR131-01, Rev.00, 2005.
      (by NuSRS 2.0)

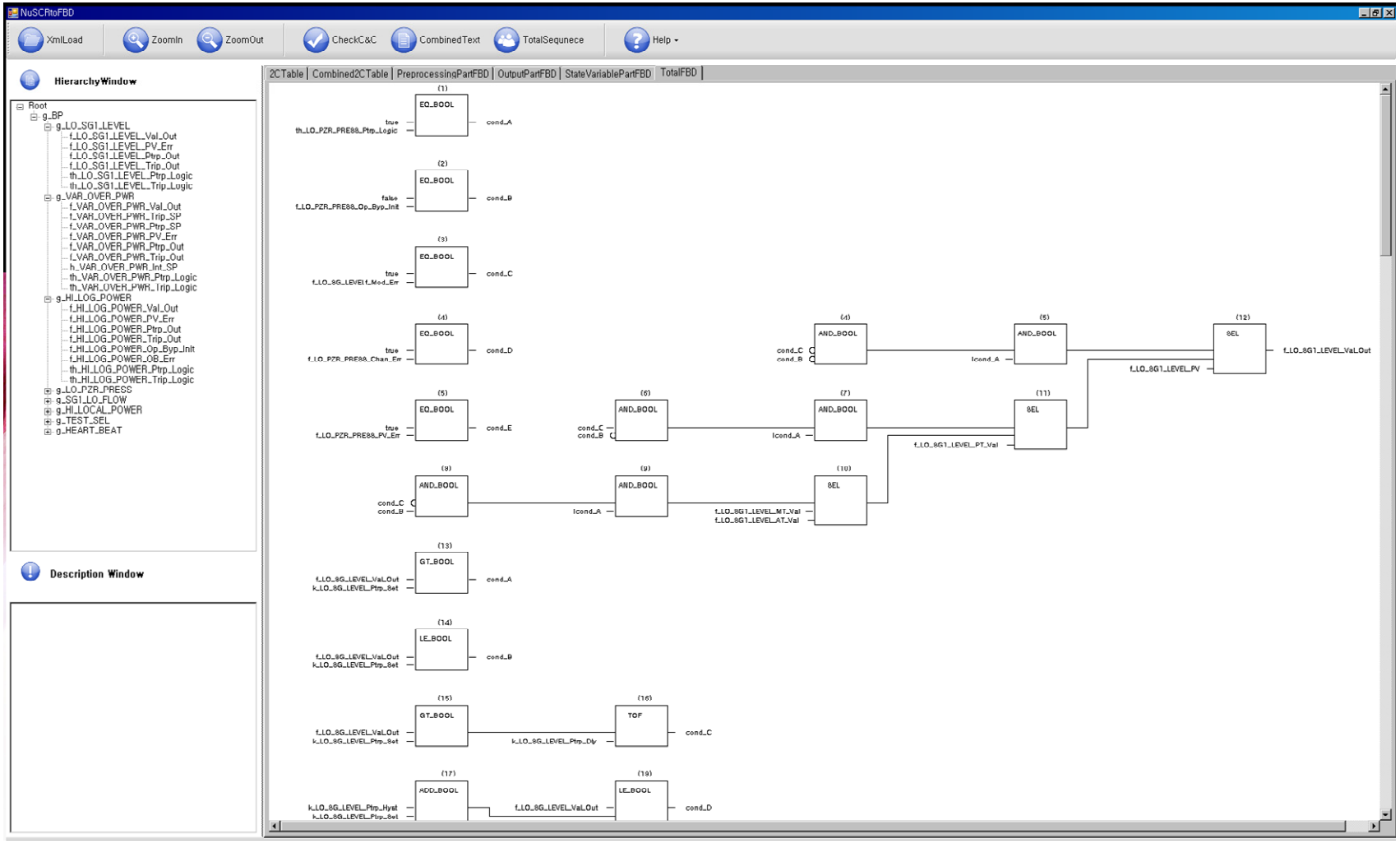


NuSRS (ver. 2.0)

2. Automatic Design Synthesis

- NuSCRtoFBD Synthesis Procedure [8]
  - Synthesizes FBD programs from NuSCR specification automatically
    - More than twice FBD blocks than manually coded and optimized ones
  - Unused in the project, because unable to develop CASE tools in advance
  - However, can be used as a baseline for FBD programming in design phase

- NuSCRtoFBD (ver 1.0)
  - CASE tool supporting
    - Automatic FBD synthesis from NuSCR
      - Reads NuSCR specification in XML format
      - Stores FBD programs in standard XML format (on-going)
    - Algorithm is being optimized

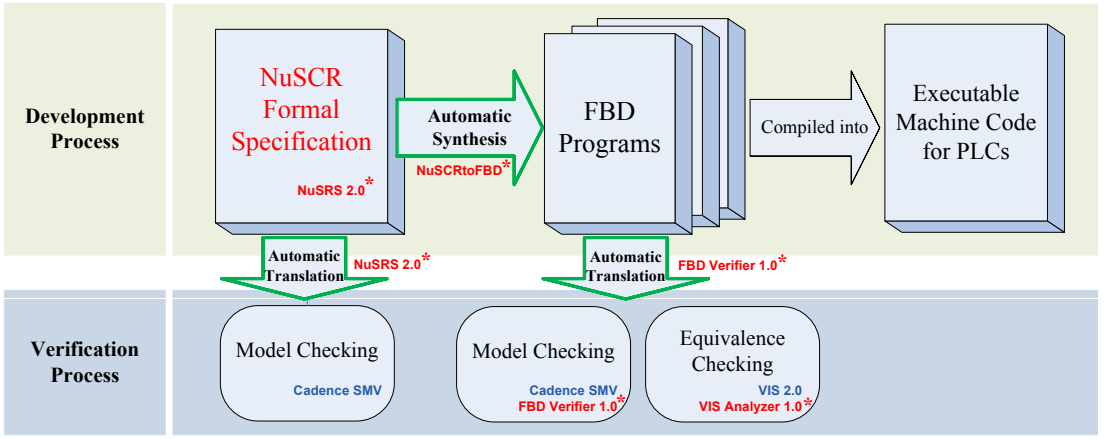


NuSCRtoFBD (ver. 1.0)

- Synthesized from KNICS RPS BP SRS (KNICS-RPS-SVR131-01, Rev.00, 2005)

Verification Process

1. Model Checking Requirements
2. Model Checking Design
3. Equivalence Checking Designs

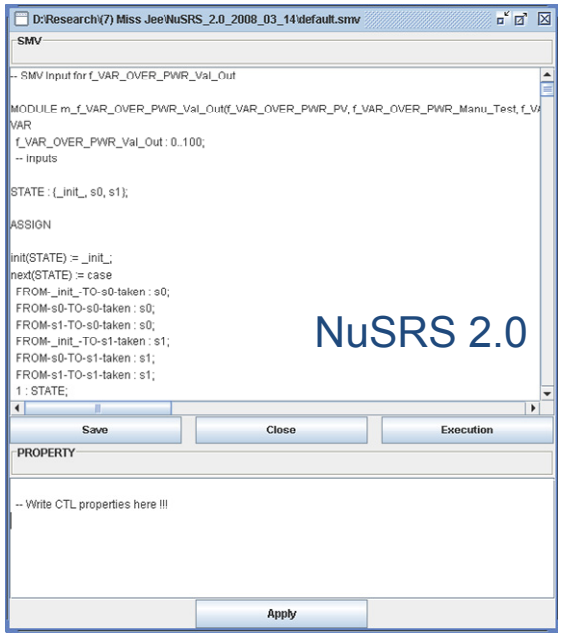


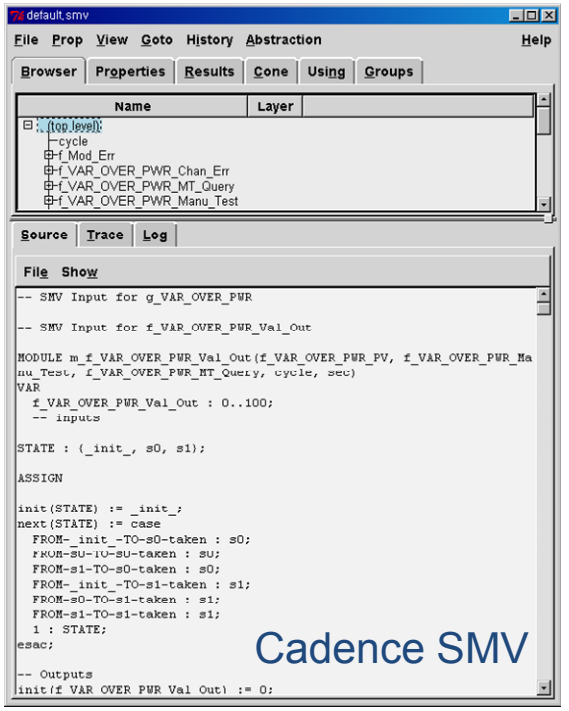
1. Model Checking Requirements

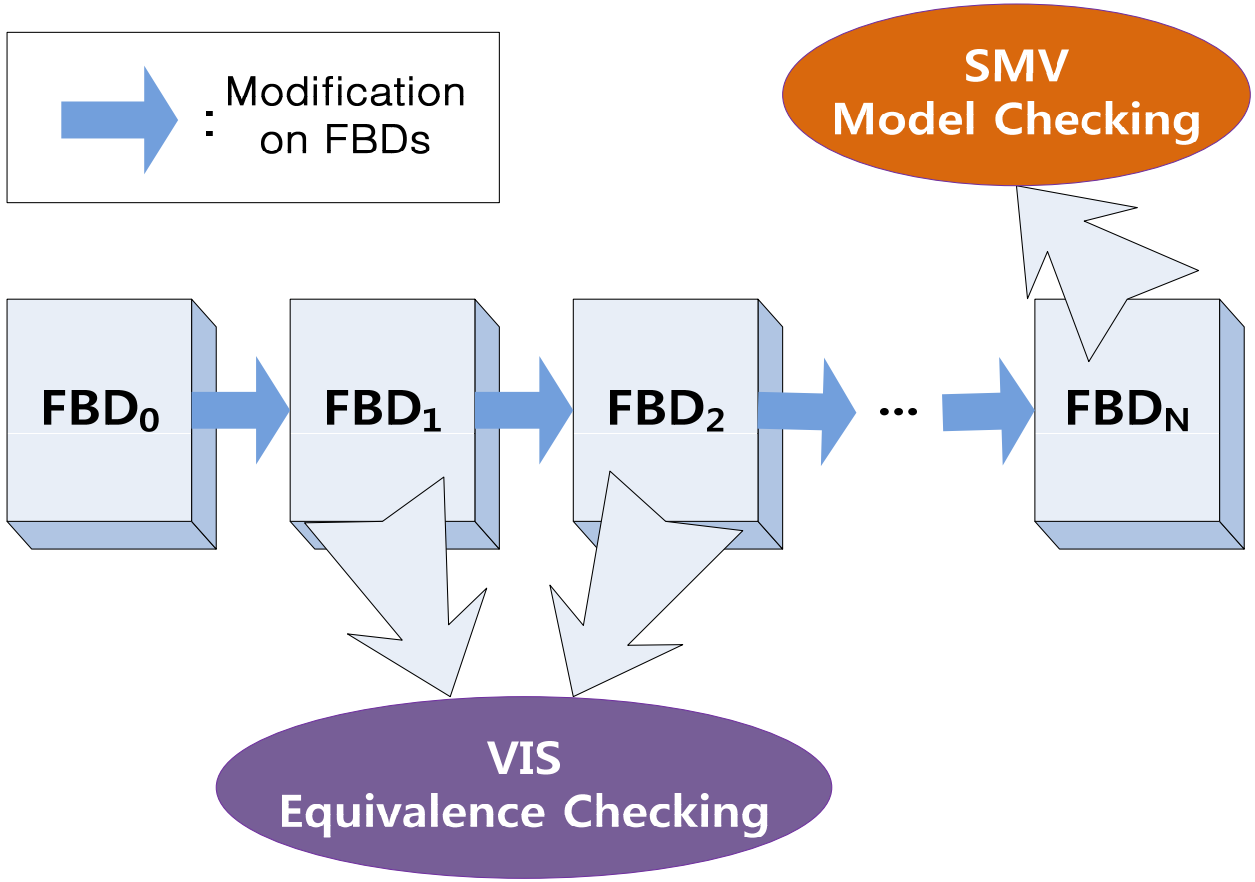
- Formal verification for requirements specification
  - Target : NuSCR formal specification
  - Tool : Cadence SMV [5]
  - Technique : CTL model checking

- NuSRS (ver. 2.0)
  - Automatic translation
    from NuSCR into SMV programs [10]
  - Seamless execution of SMV

  - Case Study
    - KNICS-RPS-SVR131-01, Rev.00, 2005
    - Found 157 errors (25 critical)







FBD Verification using

- SMV model checking & VIS Equivalence checking

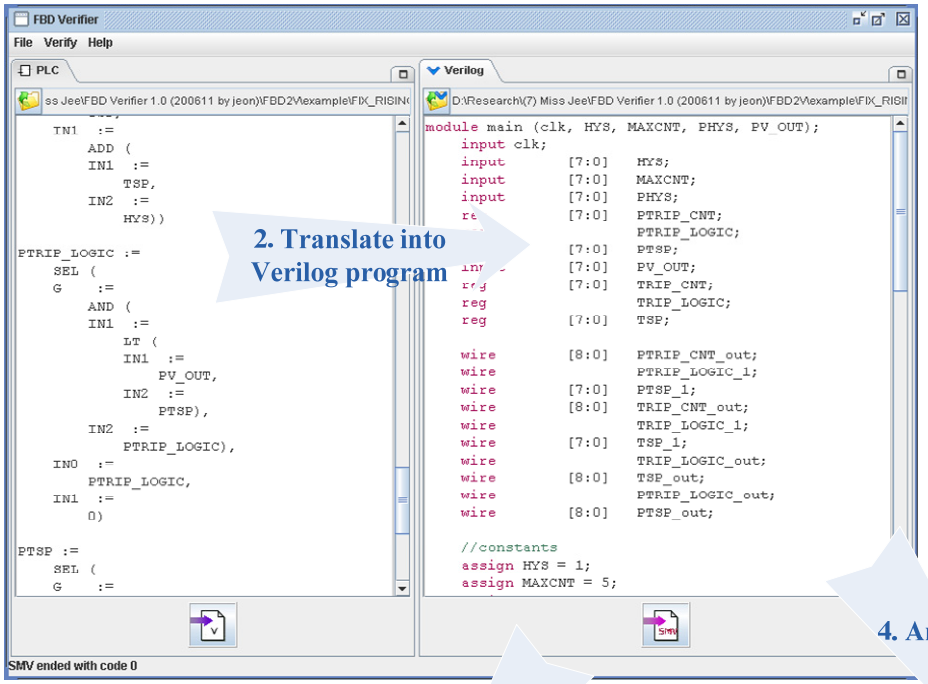
2. Model Checking Design

- Formal verification for design specification
  - Target : FBD program
  - Tool : Cadence SMV [5]
  - Technique : LTL model checking

- FBD Verifier (ver. 1.0 / 2.0)
  - Automatic translation from FBD programs into Verilog programs [11]
  - Seamless execution of SMV

  - Case Study
    - KNICS-RPS-SDS231, Rev.01, 2006
    - Found 60 errors (13 critical)

FBD Verifier 1.0

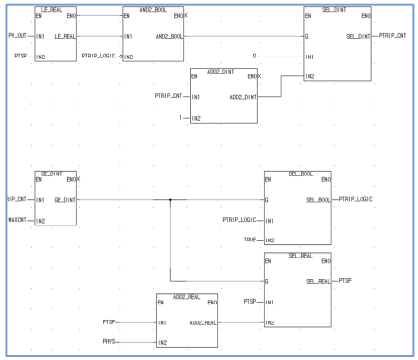


2. Translate into Verilog program

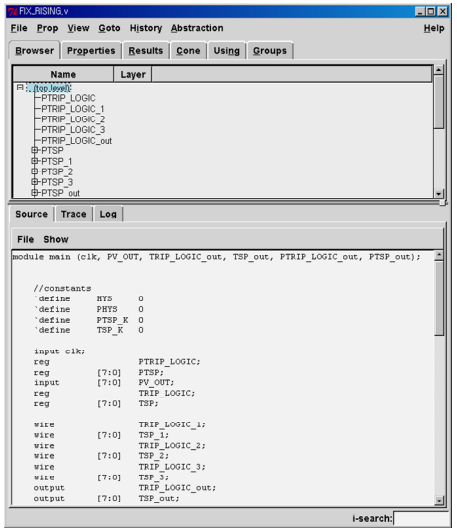
1. Read FBD programs in XML format

3. Perform SMV model checking

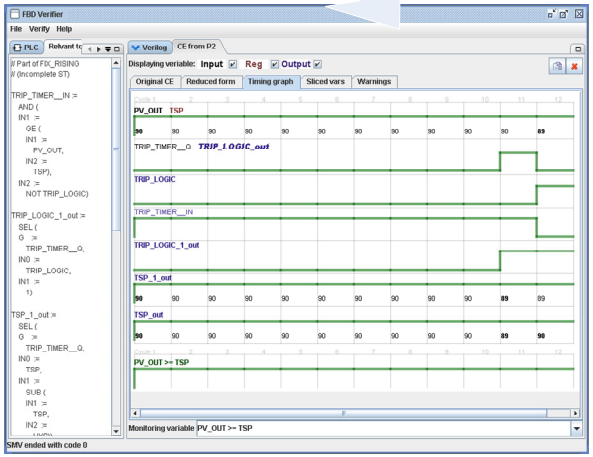
4. Analyze verification result



Engineering Tools by PLC vendors



Cadence SMV



Counterexample viewer in FBD Verifier 1.0

3. Equivalence Checking Designs

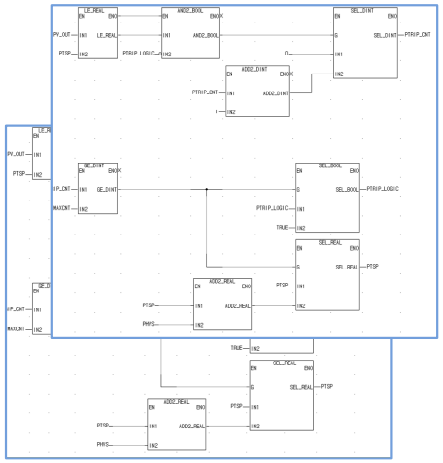
- Formal verification for design specifications
  - Target : Two FBD programs
  - Tool : VIS Verification System [4]
  - Technique : Equivalence checking, Simulation

- VIS Analyzer (ver. 1.0)
  - Seamless execution of VIS (VIS has no GUI)
  - Visualization of VIS's process and verification results [12]
  - Unused in the project, because unable to develop CASE tools in advance

  - Case Study
    - KNICS-RPS-SDS101, Rev.00, 2005
    - No official result

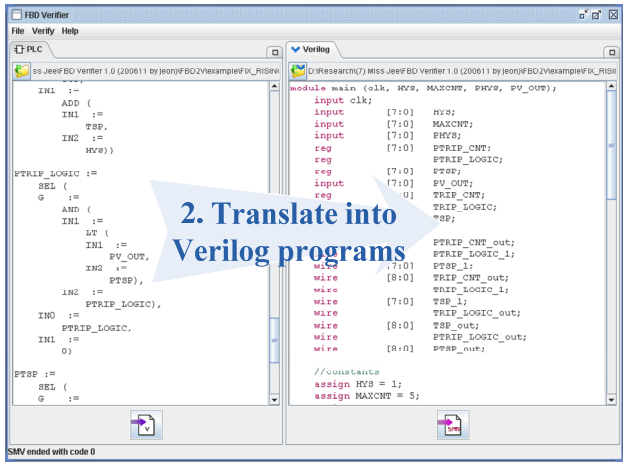
Trip Logic	Error Type	Compared FBD (Num. of Errors)	Original FBD (Num. of Errors)
Fixed Set-Point Rising Trip without Operating Bypass	Syntactic Logical	0 0	0 1
Manual Reset Variable Set-Point Trip without Operating Bypass	Syntactic Logical	0 6	3 2

Engineering Tools
by PLC vendors



1. Read two FBD programs
in XML format

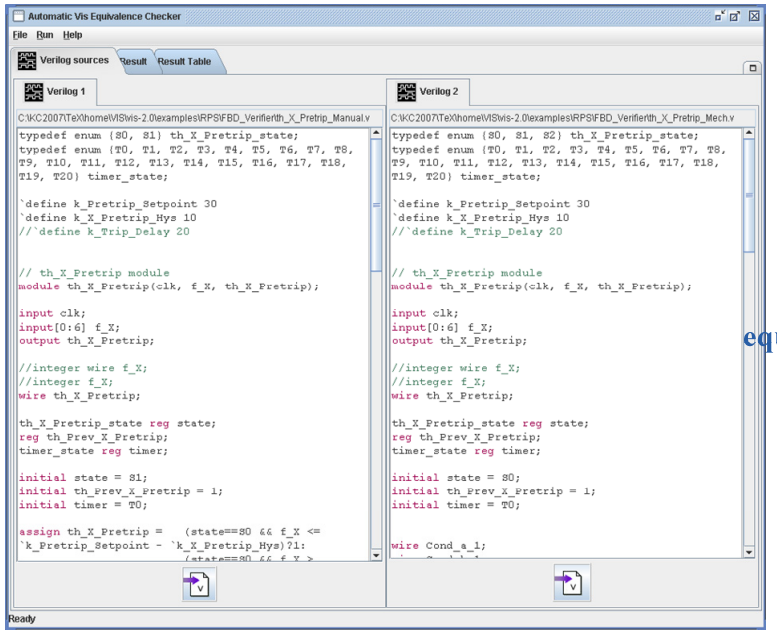
FBD Verifier 1.0



2. Translate into
Verilog programs

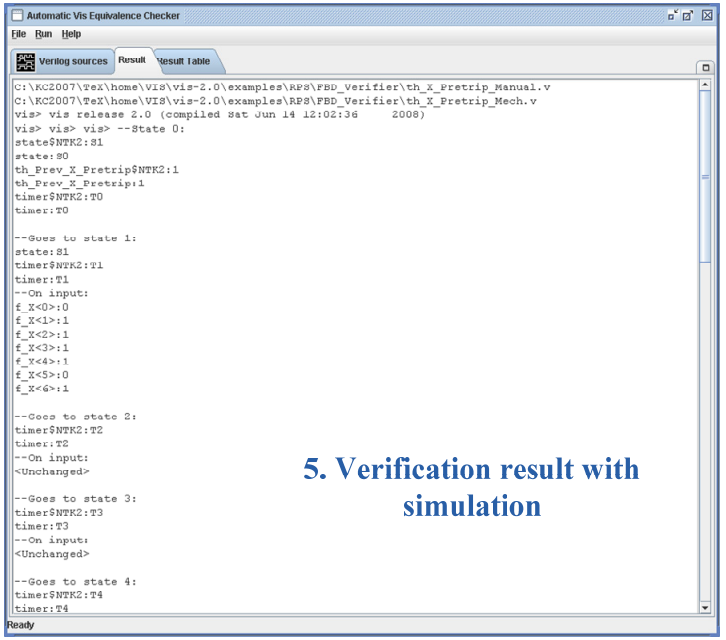
3. Read two Verilog programs

VIS Analyzer 1.0 (Source)

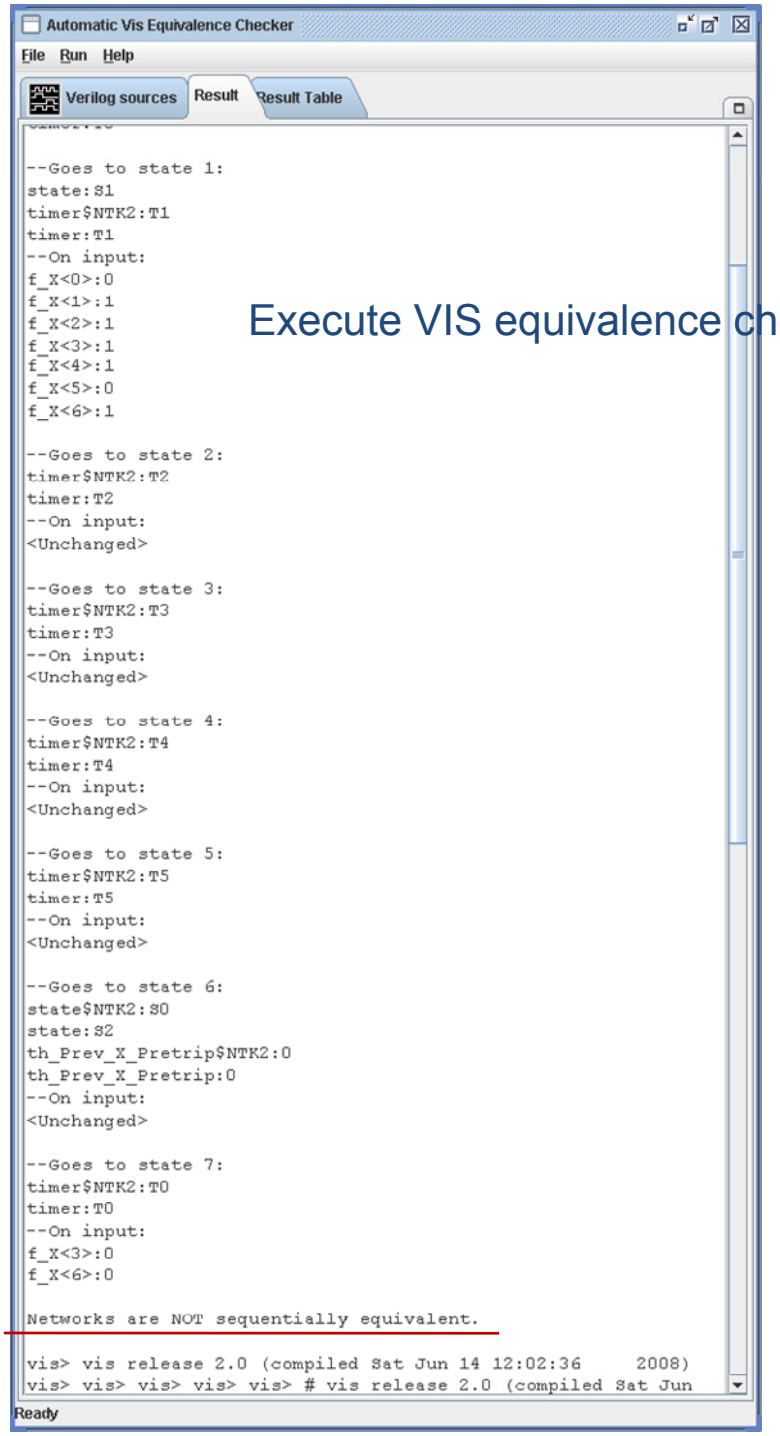


4. Perform VIS
equivalence checking

VIS Analyzer 1.0 (Result)

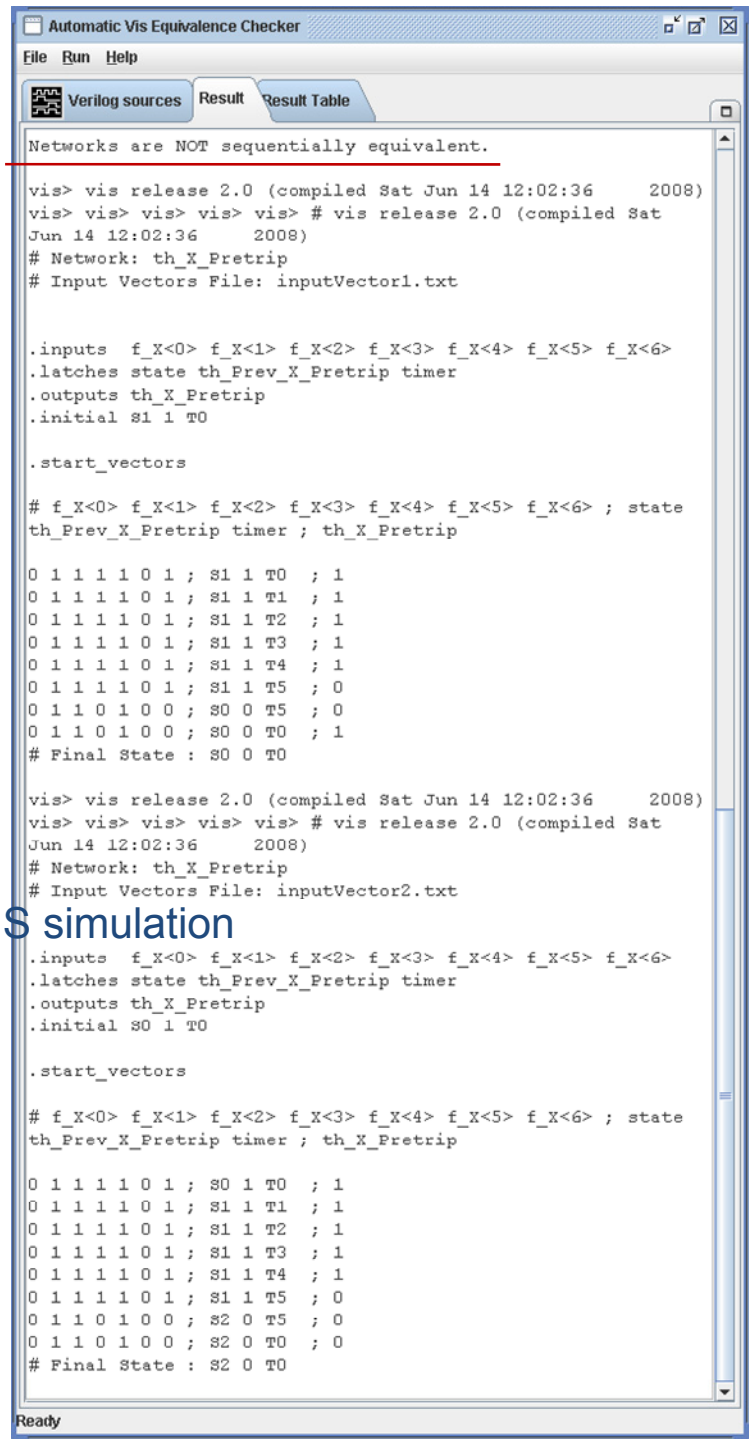


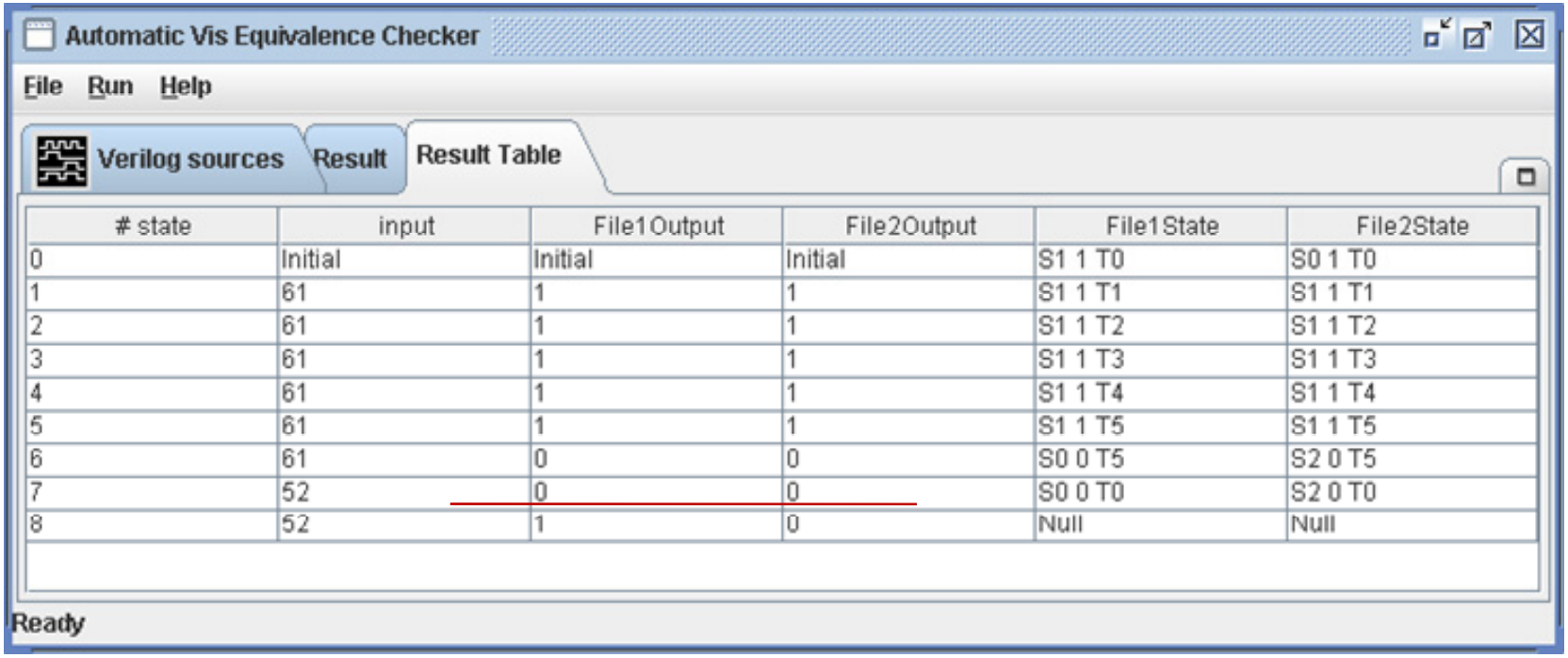
5. Verification result with
simulation



Execute VIS equivalence checking

Execute VIS simulation





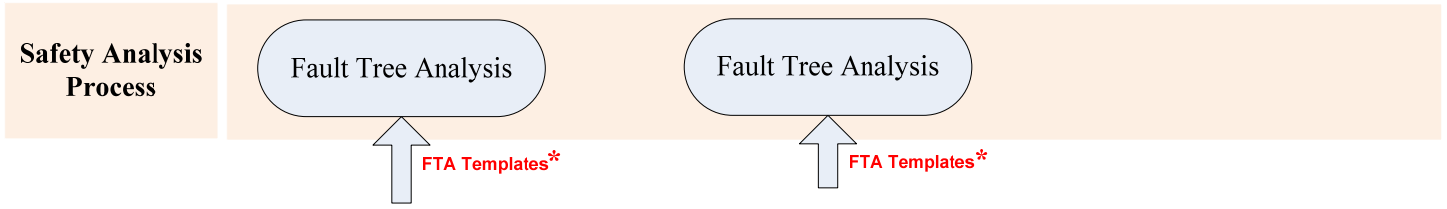
# state	input	File1Output	File2Output	File1State	File2State
0	Initial	Initial	Initial	S1 1 T0	S0 1 T0
1	61	1	1	S1 1 T1	S1 1 T1
2	61	1	1	S1 1 T2	S1 1 T2
3	61	1	1	S1 1 T3	S1 1 T3
4	61	1	1	S1 1 T4	S1 1 T4
5	61	1	1	S1 1 T5	S1 1 T5
6	61	0	0	S0 0 T5	S2 0 T5
7	52	0	0	S0 0 T0	S2 0 T0
8	52	1	0	Null	Null

VIS Analyzer (ver. 1.0)

- Visualized and reorganized result - counterexample

Safety Analysis Process

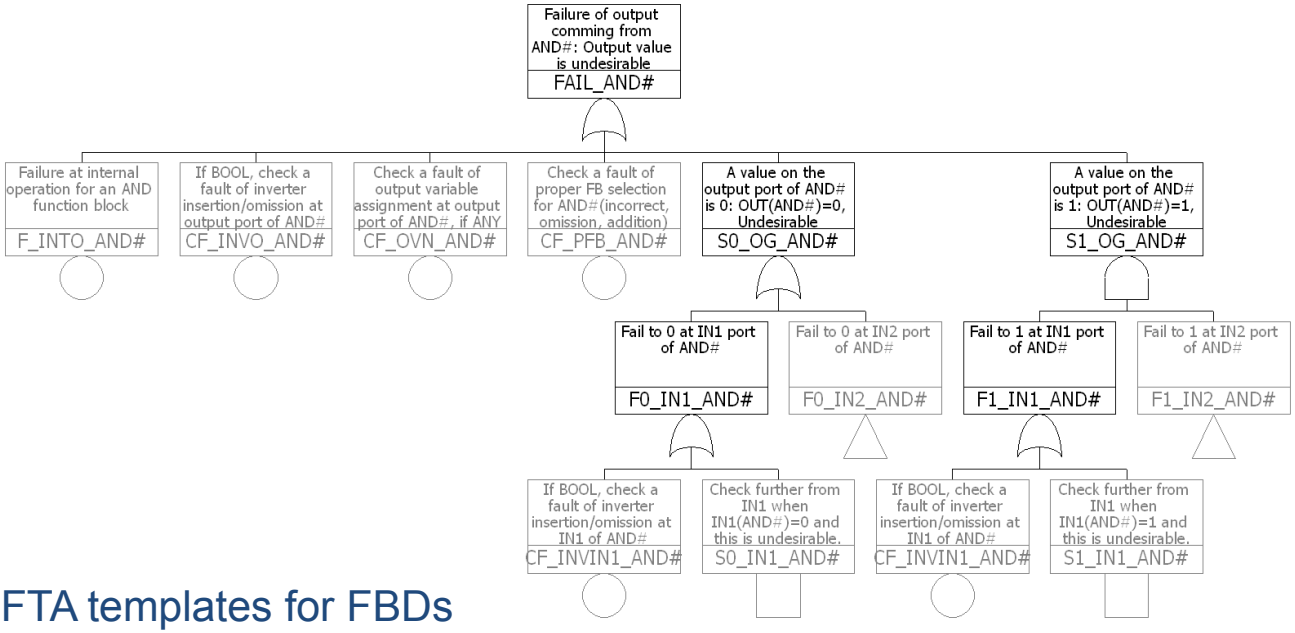
1. Fault Tree Analysis for Requirements
2. Fault Tree Analysis for Design



1. Fault Tree Analysis for Requirements
2. Fault Tree Analysis for Design

- Fault Tree Analysis
  - performed manually
  - Totally depends on analyst's experience and ability

- We provided FTA templates for NuSCR [13] and FBD [15]

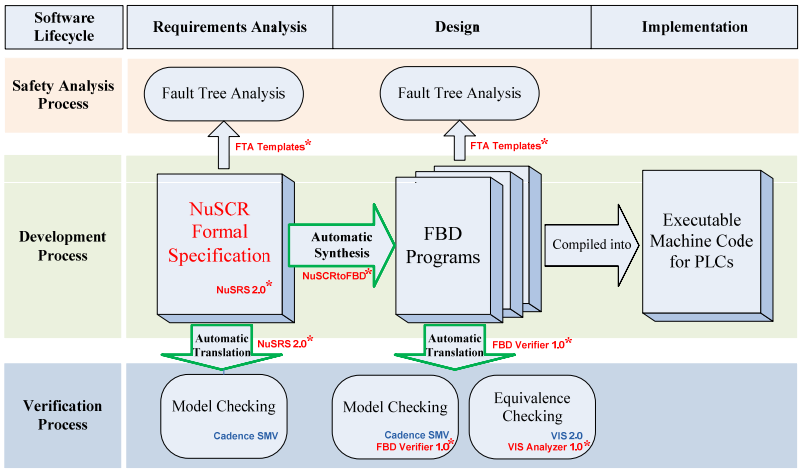


FTA templates for FBDs

Conclusion and Future Work

Conclusion

- We proposed software development processes using formal methods
  - Target: KNICS RPS for APR-1400
  - Development process
    - NuSCR formal requirements specification
    - Automatic FBD design synthesis
  - Verification process
    - Model checking NuSCR requirements
    - Model checking FBD design
    - Equivalence checking FBD designs
  - Safety analysis process
    - FTA templates for NuSCR requirements
    - FTA templates for FBD programs
  - Case Study
    - KNICS-RPS-SVR131-01, Rev.00, 2005
    - KNICS-RPS-SDS231, Rev.01, 2006



Future Work

1. Integrated Tool-set

2. Tool Enhancement
   - Self-checking : completeness & consistency (NuSRS)
   - Synchronous Verilog issue in model checking FBD programs using SMV (FBD Verifier)
   - Optimization of FBD synthesis algorithm (NuSCRtoFBD)
   - Add other functions to VIS Analyzer (VIS Analyzer)

3. Traceability Analysis
   - From requirements to design
   - From requirements' FTA to design's FTA

4. FBD Testing
   - Measures (coverage criteria)
   - Testing tool support

5. Application to Other Domains

References

[1] KNICS (Korea Nuclear Instrumentation & Control System R&D Center). http://www.knics.re.kr.

[2] Kathryn L. Heninger, “Specifying Software Requirements for Complex Systems: New Techniques and Their Application,” *IEEE Trans actions on Software Engineering*, SE Vol.6, No.1, pp2-13, 1980.

[3] Junbeom Yoo, Taihyo Kim, Sungdeok Cha, Jang-Su Lee, Han Seong Son, “A Formal Software Requirements Specification Method f or Digital Nuclear Plants Protection Systems,” *Journal of Systems and Software*, Vol.74, No.1, pp.73-83, 2005.

[4] VIS (Verification Interacting with Synthesis), http:// http://embedded.eecs.berkeley.edu/research/vis.

[5] SMV (Symbolic Model Verifier), http://www.kenmcmil.com/smv.html.

[6] Sungdeok Cha, “Pet Formalisms versus Industry-Proven Survivors: Issues on Formal Methods Education,” *Journal of Research and Practice in Information Technology*, Vol.32, No.1, pp39-46, 2000.

[7] Mats P.E. Heimdahl and Nancy G. Leveson, “Completeness and Consistency in Hierarchical State-Based Requirements,” IEEE Tran sactions on Software Engineering, Vol.22, No.6, pp363-377, 1996.

[8] Junbeom Yoo, Sungdeok Cha, Chang Hwoi Kim, Duck Yong Song, “Synthesis of FBD-based PLC Design from NuSCR Formal Speci fication,” *Reliability Engineering and System Safety*, Vol.87, No.2, pp287-294, 2005.

[9] US NRC, Digital Instrumentation and Control Systems in Nuclear Power Plants: Safety and Reliability Issues, *National Academy Pre ss*, 1997

[10] Jaemyung Cho, Junbeom Yoo, Sungdeok Cha, “NuEditor – A Tool Suite for Specification and Verification of NuSCR,” In proceeding of *Second ACIS International Conference on Software Engineering Research, Management and Applications (SERA2004)*, pp298-30 4, LA, USA, May 5-7, 2004.

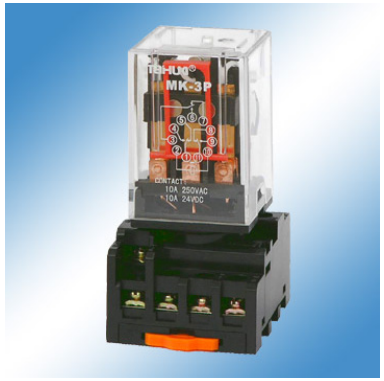
[11] Junbeom Yoo, Sungdeok Cha, and Eunkyoung Jee, “A Verification Framework for FBD based Software in Nuclear Power Plants,” I n the proceeding of *15th Asia Pacific Software Engineering Conference (APSEC)*, pp.385-392, Beijing, China, Dec. 3-5, 2008.

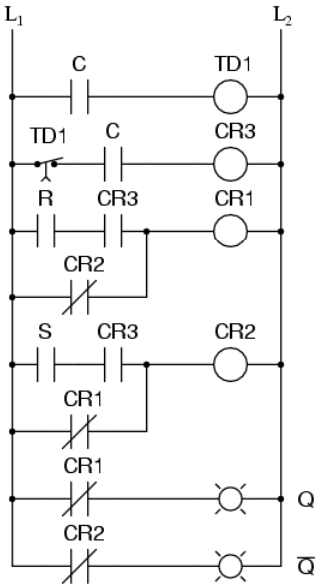
[12] Junbeom Yoo, Sungdeok Cha, and Eunkyoung Jee, “Verification of PLC Programs written in FBD with VIS,” *Nuclear Engineering a nd Technology*, Vol.41, No.2, 2009, to be published.

[13] Taeho Kim, Junbeom Yoo, Sungdeok Cha, “A Synthesis Method of Software Fault Tree from NuSCR Formal Specification using Te mplates,” *Journal of Korea Institute of Information Scientists and Engineers (in Korean)*, SE Vol.32, No.12, pp1178-1192, 2005.

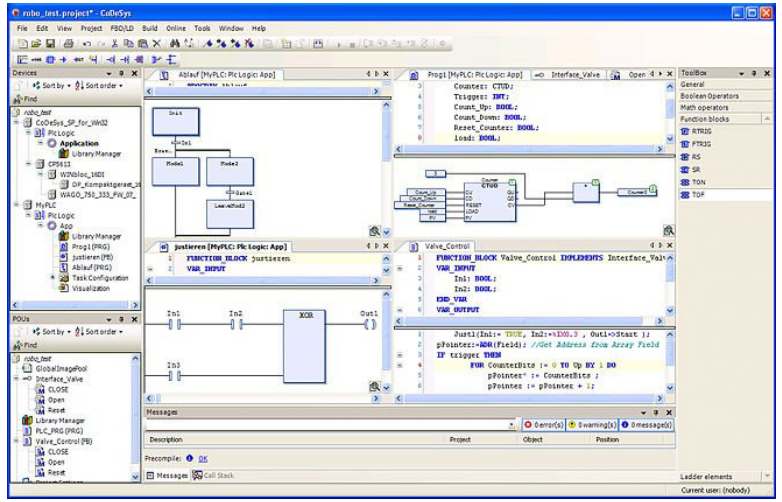
[14] Younju Oh, Junbeom Yoo, Sungdeok Cha, Han Seong Son, "Software Safety Analysis of Function Block Diagrams using Fault Tree s," *Reliability Engineering and System Safety*, Vol.88, No.3, pp215-228, 2005.

[15] Gee-Yong Park, Kwang Yong Koh, Eunkyoung Jee, Poong Hyun Seong, Kee-Choon Kwon and Dae Hyung Lee, “Fault Tree Analys is of KNICS RPS Software,” *Nuclear Engineering and Technology*, Vol.40, No.5, pp397-408, 2008.



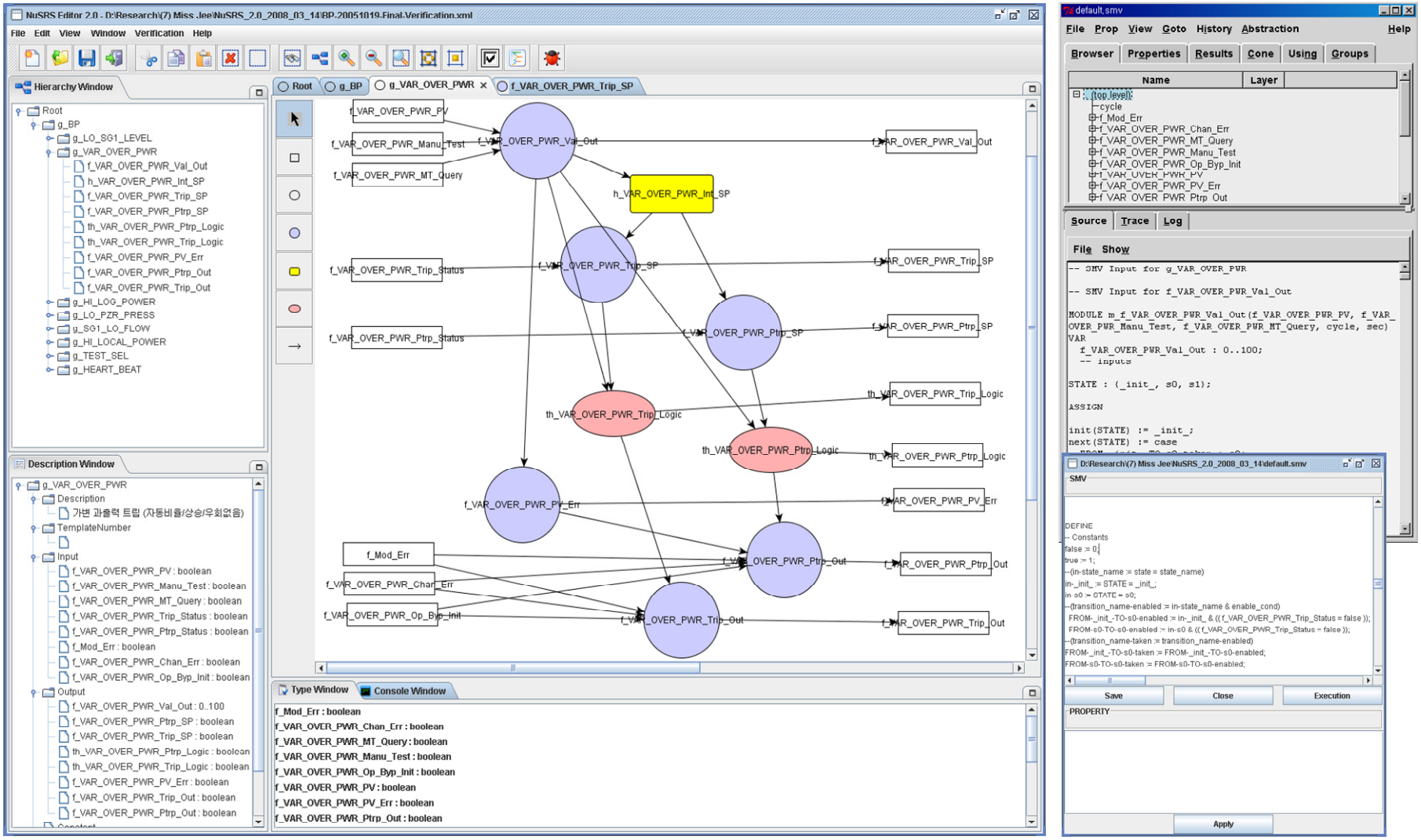


C	S	R	Q	$\overline{Q}$
⎍	0	0	latch	latch
⎍	0	1	0	1
⎍	1	0	1	0
⎍	1	1	0	0
x	0	0	latch	latch
x	0	1	latch	latch
x	1	0	latch	latch
x	1	1	latch	latch



Relay-based Analog System

PLC-based Digital System



NuSRS 2.0

- Full specification for KNICS RPS BP SRS (KNICS-RPS-SVR131-01, Rev.00, 2005)

Target subsystems		BP	CP
System Information	#pages of natural lang. spec.	60	46
	#group nodes of NuSCR spec.	8	10
	#nodes of NuSCR spec.	81	111
Verification Information	#properties	124	207
Detected Errors	Incorrect specification	1	0
	Omission	12	6
	Logic check required	6	0
	Incorrect formal specification	2	2
	Syntax error of formal specification	83	45
Total #errors		104	53
Primary #errors		19	6

SMV Verification Result

- KNICS RPS BP & CP SRS (KNICS-RPS-SVR131-01, Rev.00, 2005)

Target subsystems		BP	CP
System Information	#pages of natural lang. spec.	190	163
	#function blocks	1,335	1,623
	#variables	1,038	820
	#lines of Verilog model	7,862	3,085
Verification Information	#properties	216	83
Detected Errors	Incorrect logic	14	6
	Omission	0	2
	Incorrect in certain condition	4	0
	Incorrect FBD	13	5
	Incorrect SDS	16	0
Total #errors		47	13
Distinct #errors		10	3

SMV Verification Result

- KNICS RPS BP & CP SDS (KNICS-RPS-SDS231, Rev.01, 2006)