A Verification Framework for FBD based Software in Nuclear Power Plants

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
  – Safety-Critical Software in Nuclear Power Plants
  – Software Development Process

• Background
  – FBD

• Verification Framework
  – VIS Equivalence Checking
  – SMV Model Checking
  – Case Study

• Conclusion & Future Work
Introduction

- Safety-Critical Software in Nuclear Power Plants
  - RPS (Reactor Protection System)
  - ESF-CCS (Engineering Safety Features Component Control System)
Introduction

• A Software Development Process for RPS in KNICS APR-1400 NPP
  - (http://www.knics.re.kr/english/eindex.html)
Background

- **FBD (Function Block Diagram)**
  - IEC 61131-3 standard declared 5 programming languages for PLC (ST, LD, IL, SFC, FBD)
  - KNICS consortium decided to use FBD to program KNICS RPS software
  - Sequential Interconnections between function blocks
Verification Framework

• In design phase,
• Two different formal verifications to verify FBD programs efficiently,
• Based on our experience on KNICS RPS for 7 years
  – Equivalence Checking : VIS verification system (ver.2.0)
  – Model Checking : Cadence SMV model checker
1. SMV Model Checking

- **SMV Model Checking with LTL properties**
  - **Cadence SMV model checker** (http://www.kenmcmil.com/smv.html)
    - An extension of SMV from CMU
    - CTL / LTL model checking
    - Two front-ends
      - SMV input programs (for CTL/LTL properties)
      - Verilog program (for LTL properties)
  - **FBD Verifier 1.0** (http://dslab.konkuk.ac.kr/Nuclear-Design/FBD_Verifier.htm)
    - Translates FBD into Verilog automatically
    - Properties are inserted into Verilog programs (through “assert” statement)
    -Executes Cadence SMV with translated Verilog program seamlessly
1. Read an FBD program in standard XML format
2. Translate the FBD into an equivalent Verilog program
3. Execute Cadence SMV model checker
Case Study

- SMV model checking for KNCIS RPS BP & CP
  - Performed to up-to-date whole KNICS-RPS-SDS231 Rev.02

<table>
<thead>
<tr>
<th>System</th>
<th># of pages of requirements Spec. (Natural lang.)</th>
<th># of function blocks</th>
<th># of variables</th>
<th># of lines in Verilog model</th>
</tr>
</thead>
<tbody>
<tr>
<td>BP</td>
<td>190</td>
<td>1,335</td>
<td>1,038</td>
<td>7,862</td>
</tr>
<tr>
<td>CP</td>
<td>163</td>
<td>1,623</td>
<td>820</td>
<td>3,085</td>
</tr>
</tbody>
</table>

- Found a few, not many important verification results

<table>
<thead>
<tr>
<th>System</th>
<th>BP</th>
<th>CP</th>
</tr>
</thead>
<tbody>
<tr>
<td># of Properties</td>
<td>216</td>
<td>83</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Found Errors</th>
<th>BP</th>
<th>CP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incorrect Logic</td>
<td>14</td>
<td>6</td>
</tr>
<tr>
<td>Omission</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Ambiguous Logic</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>Incorrect FBD</td>
<td>13</td>
<td>5</td>
</tr>
<tr>
<td>Incorrect Design</td>
<td>16</td>
<td>0</td>
</tr>
</tbody>
</table>

| Total # of Errors    | 47 | 13 |
| Distinct # of Errors | 10 | 3  |
2. VIS Equivalence Checking

• Behavioral Equivalence Checking between two FBD programs
  – VIS verification system 2.0 (http://embedded.eecs.berkeley.edu/research/vis/)
    • Widely used in hardware design,
      – Simulation
      – CTL model checking
      – Equivalence checking
      – Etc.
    • Reads Verilog program
    • But, no graphical interface
  – VIS Analyzer 1.0 (http://dslab.konkuk.ac.kr/Nuclear-Design/VIS_Analyzer.htm)
    • Seamless execution of VIS verifications
      – vl2mv → read_blif_mv → flatten_hierarchy → seq_verify, simulate
    • Automatic reorganization of verification result through VIS simulation
1. Read two Verilog programs
2. Execute VIS equivalence checking

3. Execute VIS simulation
4. Display a full trace for counterexample

<table>
<thead>
<tr>
<th># state</th>
<th>Input</th>
<th>File1Output</th>
<th>File2Output</th>
<th>File1State</th>
<th>File2State</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Initial</td>
<td>Initial</td>
<td>Initial</td>
<td>S1 1 T0</td>
<td>S0 1 T0</td>
</tr>
<tr>
<td>1</td>
<td>61</td>
<td>1</td>
<td>1</td>
<td>S1 1 T1</td>
<td>S1 1 T1</td>
</tr>
<tr>
<td>2</td>
<td>61</td>
<td>1</td>
<td>1</td>
<td>S1 1 T2</td>
<td>S1 1 T2</td>
</tr>
<tr>
<td>3</td>
<td>61</td>
<td>1</td>
<td>1</td>
<td>S1 1 T3</td>
<td>S1 1 T3</td>
</tr>
<tr>
<td>4</td>
<td>61</td>
<td>1</td>
<td>1</td>
<td>S1 1 T4</td>
<td>S1 1 T4</td>
</tr>
<tr>
<td>5</td>
<td>61</td>
<td>1</td>
<td>1</td>
<td>S1 1 T5</td>
<td>S1 1 T5</td>
</tr>
<tr>
<td>6</td>
<td>61</td>
<td>0</td>
<td>0</td>
<td>S0 0 T5</td>
<td>S2 0 T5</td>
</tr>
<tr>
<td>7</td>
<td>52</td>
<td>0</td>
<td>0</td>
<td>S0 0 T0</td>
<td>S2 0 T0</td>
</tr>
<tr>
<td>8</td>
<td>52</td>
<td>1</td>
<td>0</td>
<td>Null</td>
<td>Null</td>
</tr>
</tbody>
</table>
Case Study

- VIS equivalence checking for KNCIS RPS BP
  - We didn’t meet the schedule, so a few official verification results are left only.
    - Requirements: KNICS-RPS-SRS101 Rev.00 (prototype)
    - Original FBD: KNICS-RPS-SDS101 Rev.00 (prototype)
    - Compared FBD: Synthesized automatically from the requirements spec.
  - Found several errors

<table>
<thead>
<tr>
<th>Trip Logic</th>
<th>Error Type</th>
<th>Compared FBD (Num. of Errors)</th>
<th>Original FBD (Num. of Errors)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed Set-Point Rising Trip without Operating Bypass</td>
<td>Syntactic</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Logical</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Manual Reset Variable Set-Point Trip without Operating Bypass</td>
<td>Syntactic</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Logical</td>
<td>6</td>
<td>2</td>
</tr>
</tbody>
</table>
Conclusion & Future Work

- We proposed a software verification framework
  - Target: KNICS RPS
    - HW: PLC (Programmable Logic Controller)
    - SW: FBD (Function Block Diagram)
  - Two verification techniques together
    - SMV Model Checking (Cadence SMV + FBD Verifier)
    - VIS Equivalence Checking (VIS 2.0 + VIS Analyzer)
  - They performed the formal verification of KNICS RPS sufficiently.

- We’re planning
  - A combined tool-set (FBD Verifier + VIS analyzer) with enhanced GUIs
  - Enhance through applying to other systems (e.g. ESF-CCS)