Semi-formal Reformulation of Requirements for Formal Property Verification

Katharina Ceesay-Seitz, CERN
Hamza Boukabache, CERN
Daniel Perrin, CERN
Agenda

• Motivation & Introduction

• Natural Language Properties (NLPs)

• Examples

• Results

• Conclusion
Motivating Example

• 1999: Loss of NASA’s Mars Climate Orbiter space craft
  – Onboard software and ground software used different measurement units

• 2018 + 2019: Crash of 2 Boeing 737 MAX 8 planes
  – Multiple reasons (sensor failure, lack of pilot training, safety features as add-ons to purchase, ...)

• In both cases:
  – Lack of communication
  – Unvalidated assumptions
Motivation

• Design specification is a refinement of requirements specification
  – Different levels of abstraction
  – Often written by different persons
  – Ambiguities → misinterpretations

• Misunderstandings enter the design specification
  – Can have life threatening consequences (safety)
  – Can cause unnecessary iterations (expensive)
Motivation & Introduction

• Validation of the design (specification) necessary
  – Proposed technique: Natural Language Properties (NLPs)
  – Validating the design specification through verification

• Formal Property Verification
  – Exhaustive proof of formal properties
  – Bounded model checking
  – Property languages: SystemVerilog, PSL, ...
Natural Language Properties (NLPs)

• Formal properties expressed in natural language
• Defined “n:1” mapping from NLP snippets to SystemVerilog

• Review of NLPs (and by that the formal properties) by requirements engineer
  – Validates the verification
  – Implicitly validates the specification when verification passes
Workflow

- Requirements Engineer
  - Requirements Specification
    - Review Meeting
      - Translation of NLPs into SystemVerilog

- Verification Engineer
  - Review of Requirements & Natural Language Property Specification
    - Time
Life Cycle

System Safety Requirements

System Architecture Specification

FPGA/Software Safety Requirements

FPGA/Software System Design

Module Design Specification

Code Implementation

Module/Unit Testing

Module Integration Testing

FPGA/Software System Validation

Programmable Electronics Integration Testing

System Safety Validation

Validation

Verification

NLP

Time Flow, Trace
Radiation Protection at CERN
CERN RadiatiOn Monitoring Electronics (CROME)

Supervision

~150 configuration parameters
~60 measurements

Integration Block (radiation dose):
7 conditions at 2 real-time cycles
 determine alarm state → 2^14 possibilities
max. integration period: 10 years
large bit widths: 48-64 bit
# Natural Language Property Mapping

<table>
<thead>
<tr>
<th>Natural Language</th>
<th>SystemVerilog</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expr1 equals Expr2</td>
<td>Expr1 == Expr2</td>
</tr>
<tr>
<td>Expr1 and Expr2</td>
<td>Expr1 &amp;&amp; Expr2</td>
</tr>
<tr>
<td>Expr implies that: Seq</td>
<td>Expr \rightarrow\text{Seq}</td>
</tr>
<tr>
<td>Every time when Expr: Seq</td>
<td>Expr \rightarrow\text{Seq}</td>
</tr>
<tr>
<td>TimeSeq after Expr: Seq</td>
<td>Expr ##\text{TimeSeq Seq}</td>
</tr>
<tr>
<td>(Expr)</td>
<td>(Expr)</td>
</tr>
<tr>
<td>(“”)</td>
<td></td>
</tr>
<tr>
<td>Natural Language</td>
<td>SystemVerilog</td>
</tr>
<tr>
<td>---------------------------------------------------------------------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>Cycle is the start of a MC</td>
<td>$rose(mtValidxDI)</td>
</tr>
<tr>
<td>The time passed since integration start</td>
<td>== etCount</td>
</tr>
<tr>
<td>At the previous MC</td>
<td>signalNameLastMC</td>
</tr>
<tr>
<td>At integration end time</td>
<td>elapsedTimexDO &gt;= integralTimexDI</td>
</tr>
<tr>
<td>Once the calculation is ready</td>
<td>`nrCUntilCalcRdy</td>
</tr>
<tr>
<td>Alarm function is/was activated</td>
<td>alarmActivexDI == 1</td>
</tr>
<tr>
<td>Integral value</td>
<td>signed'(integralxDO)</td>
</tr>
</tbody>
</table>

MC = Measurement Cycle
Example 1

- **Requirement:**
  "A manual zeroing of the integrated value shall be possible."

- **SystemVerilog property:**
  ```
  property pIntManualResetNextMT();
  ($rose(mtValidxDI) && intResLastMC == 1) |
  ##`nrCUntilCalcRdy integralxDI == integralTB;
  endproperty
  ```
Example 1

property pIntManualResetNextMT();
    ($rose(mtValidxDI) && intResLastMC == 1)
    |->
    ##`nrCUntilCalcRdy integralxDO == integralTB;
endproperty

• Natural Language Property:

Cycle is the start of a MC and integral reset at previous MC equaled 1 implies that:

once the calculation is ready, integral value equals the testbench internal integral value ("integrated since last MC")
Example 2

• Requirement:
• "It shall be possible to manually trigger a reset of an integration alarm through the supervision software."

• Natural language property:
  "(Cycle is no MC and (alarm was configured as latched at the previous MC) and alarm reset equals 1 and (integral value is less than (threshold at previous MC) or alarm function was deactivated at previous MC)) implies that: (in one clock cycle, alarm is off)"

MC = Measurement Cycle
Example 2

"(Cycle is no MC and (alarm was configured as latched at the previous MC) and alarm reset equals 1 and (integral value is less than (threshold at previous MC) or alarm function was deactivated at previous MC)) implies that:(in one clock cycle, alarm is off)"

• **SystemVerilog property:**

```
property pIntAlarmResetBetweenMT1();
  (mtValidxDI == 0 && latchedLastMC == 1 &&
   integralAlarmResetxDI == 1 &&
   (signed'(integralxDO) < signed'(thresholdLastMc) ||
    alarmActiveLastMc == 0))
  |
  #1 (ALARMxDO == 0);
endproperty
```

**MC = Measurement Cycle**
Results for CERN RadiatiOn Monitoring Electronics (CROME)

• Found a severe safety-critical fault
  – Design and verification engineer had same assumptions
  – Requirements engineer intended different meaning
  – Review of NLPs discovered the discrepancy

• Review of NLPs for integration dose block took only 1 hour
Conclusion & Outlook

• Natural Language Properties (NLPs) & workflow

• Main goal:
  – Avoid misunderstandings
    • Reduce risk of faults
    • Reduce number of iterations during development

• Adds to V-Model:
  – Specification of NLPs
  – Review of NLPs by requirements engineer → validation
Conclusion & Outlook

• Found a safety-critical fault in the CERN RadiatiOn Monitoring Electronics (CROME)

• NLPs will be used for further verification of CROME

• Possibility for automated translation of NLPs into SystemVerilog
Thank You!

Questions?