VIS Analyzer: Visual Assistant for VIS Verification and Analysis

Sehun Jeong¹, Junbeom Yoo², Sungdeok Cha¹
¹Korea University, Republic of Korea
²Konkuk University, Republic of Korea

*Photo was taken at Real Alcazar in Seville
Outline

• Motivation
• Our Solution
• Comparisons
• Conclusions & Future Work
KNICS Project

- Development of software for nuclear power plant reactor protection system (2000 ~ 2007)
- To be deployed in Shin-Uljin NPP in Korea
Goals

• Use proven-effective formal techniques when feasible to demonstrate high reliability

• Domain experts should NOT be left alone in the dark
In implementation, code optimization is unavoidable
- Demonstration of behavioral equivalence is critical
- Testing is not sufficient
  - VIS formally checks behavior equivalence
- Counterexample is provided when targets are inequivalent

Ver. 1 \rightarrow Ver. 2 \rightarrow \cdots \rightarrow Ver. n

Behavior Equivalence Checking
KNICS and VIS

- KNICS uses **FBD** as implementation language
- VIS can accept **Verilog** code as an input
- FBD and Verilog have similar semantics

➡ Synthesis of **Verilog from FBD** is straightforward*

➡ VIS can perform **CTL & LTL model checking**

✓ I can say that VIS is worth for “★★★★☆☆”

Practical Limitations on VIS

- Unfamiliar user interface
- Overly detailed verification process

```bash
vis> read_blif_mv ../.../example/mc/RPS/test.mv
Warning: Some variables are unused in model SEL.
Warning: Some variables are unused in model MUX_INT.
vis> model_check ../.../example/mc/RPS/property.ctl
There is no network. Use flatten_hierarchy.
vis> flatten_hierarchy
vis> model_check ../.../example/mc/RPS/property.ctl
Network has no partition. Cannot create FSM.
vis> build_partition_mdds
The MDD variables have not been ordered. Use static_order.
vis> static_order
vis> build_partition_mdds
vis> model_check ../.../example/mc/RPS/property.ctl
# MC: formula failed --- AG <<< BB.BB.AAA.timer<0>=1 * BB.BB.AAA.timer<1>=0 > B BB.AAA.timer<2>=1 -> AX BB.DD.th_Prev_X_Pretrip=0 >>>
```
**Information partialities** in counterexamples decrease readability

- Unchanged input & state values are not shown
- Output values are not shown in equivalence checking

<table>
<thead>
<tr>
<th>...</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>--Goes to state 8: BB.BB.AAA.timer&lt;0&gt;:0 BB.BB.AAA.timer&lt;1&gt;:1</td>
<td>--Goes to state 10: BB.BB.AAA.timer&lt;0&gt;:0 BB.BB.AAA.timer&lt;1&gt;:0 BB.BB.AAA.timer&lt;2&gt;:1</td>
</tr>
<tr>
<td>--On input: f_X_Raw&lt;4&gt;:0 f_X_Raw&lt;5&gt;:0</td>
<td>--On input: &lt;Unchanged&gt;</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
• **Textual display** is not adequate for counterexample representation.
Outline

• Motivation
• Our Solution
• Comparisons
• Conclusions & Future Work
VIS Analyzer 3.0

• Graphic user interface
• Automation of VIS verification features
  - Equivalence checking
  - Model checking
  - Simulation
• Verification Results without partialities
• Visualization of the results
Side-by-side code comparing with syntax highlighting
Intuitive counterexample visualization
Flexible display of verification result

<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>f_X:61</td>
<td>1</td>
<td>1</td>
<td>S1 1 T1</td>
<td>S1 1 T1</td>
</tr>
<tr>
<td>2</td>
<td>f_X:61</td>
<td>1</td>
<td>1</td>
<td>S1 1 T2</td>
<td>S1 1 T2</td>
</tr>
<tr>
<td>3</td>
<td>f_X:61</td>
<td>1</td>
<td>1</td>
<td>S1 1 T3</td>
<td>S1 1 T3</td>
</tr>
<tr>
<td>4</td>
<td>f_X:61</td>
<td>1</td>
<td>1</td>
<td>S1 1 T4</td>
<td>S1 1 T4</td>
</tr>
<tr>
<td>5</td>
<td>f_X:61</td>
<td>1</td>
<td>1</td>
<td>S1 1 T5</td>
<td>S1 1 T5</td>
</tr>
<tr>
<td>6</td>
<td>f_X:61</td>
<td>0</td>
<td>0</td>
<td>S0 0 T5</td>
<td>S2 0 T5</td>
</tr>
<tr>
<td>7</td>
<td>f_X:52</td>
<td>0</td>
<td>0</td>
<td>S0 0 T0</td>
<td>S2 0 T0</td>
</tr>
<tr>
<td>8</td>
<td>f_X:52</td>
<td>1</td>
<td>0</td>
<td>Null</td>
<td>Null</td>
</tr>
</tbody>
</table>

Verification and model check Ready
Raw data for reference

```plaintext
-- On input:
<Unchanged>

-- Goes to state 5:
BB.BB.AAA.timer<0>: 0
BB.BB.AAA.timer<1>: 0
BB.BB.AAA.timer<2>: 1
-- On input:
<Unchanged>

-- Goes to state 6:
BB.BB.AAA.timer<0>: 1
-- On input:
<Unchanged>

Counter example ends

# MC: formula passed --- AG(((BB.BB.AAA.timer<0>=1 * BB.BB.AAA.timer<1>=0) * BB.BB.AAA.timer<2>=1) -> EX(BB.DD.th_Pretrip=0)))
# MC: formula passed --- AG(((BB.BB.AAA.timer<0>=1 * BB.BB.AAA.timer<1>=0) * BB.BB.AAA.timer<2>=1) -> E(((BB.BB.AAA.timer<0>=1 * BB.BB.AAA.timer<1>=0) * BB.BB.AAA.timer<2>=1) U BB.DD.th_Pretrip=0)))
```
3-in-1 Model checking window

function [0:1] MUX_INT_1;
  input [0:1] in1;
  input [0:1] in2;
  input [0:1] in3;
  input [0:1] in4;
begin
  MUX_INT_1
  end
endfunction

//SPEC AG(BB.BB.AAA.timer<0>=1 -> AX
BB.DD.th_Prev_X_PRETRIP=0);

VIS configuration
Verilog source code path
C:\cygwin\home\Sehun.Jeong\example\mc\RPS\test.v
Open
Model check specification file path
C:\cygwin\home\Sehun.Jeong\example\mc\RPS\property.ctl
Open
Model check fairness property file path
C:\cygwin\home\Sehun.Jeong\wis-2.0\examples\ctlp3\ctlp3.fair
Open

Accept  Cancel
Outline

• Motivation
• Our Solution
• Comparisons
• Conclusions & Future Work
Comparison 1

• Equivalence checking process

```
c:\>vl2mv th_X_Pretrip_Manual.v
c:\>vl2mv th_X_Pretrip_Mach.v
vis release 2.0 (compiled Sat Jun 14 12: ... )
vis> read_blif_mv th_X_Pretrip_Manual.mv
vis> flatten_hierarchy
vis> seq_verify th_X_Pretrip_Mech.mv
--State 0:
state$NTK2:S1
...```
Comparison 1

- Equivalence checking process

```plaintext
c:\>vl2mv th_X_Pretrip_Manual.v
C:\>vl2mv th_X_Pretrip_Mach.v

vis release 2.0 (compiled Sat Jun 14 12: ... )

vis> read_blif_mv th_X_Pretrip_Manual.mv
vis> flatten_hierarchy
vis> seq_verify th_X_Pretrip_Mech.mv

--State 0:
state$NTK2:S1
```

...
• Equivalence checking result

--Goes to state 6:
state$NTK2:S0
state:S2
th_Prev_X_Pretrip$NTK2:0
th_Prev_X_Pretrip:0
--On input:
<Unchanged>

--Goes to state 7:
timer$NTK2:T0
timer:T0
--On input:
f_X<3>:0
f_X<6>:0
Equivalence checking result

--Goes to state 6:
state$NTK2:S0
state:S2
th_Previous_X_Pretrip$NTK2:0
th_Previous_X_Pretrip:0
--On input:
<Unchanged>

--Goes to state 7:
timer$NTK2:T0
timer:T0
--On input:
f_X<3>:0
f_X<6>:0
### Equivalence checking result representation

<table>
<thead>
<tr>
<th>$f_X&lt;0&gt;$: 0</th>
<th><strong>&lt;State 6&gt;</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_X&lt;1&gt;$: 1</td>
<td>state$NTK2:S0$</td>
</tr>
<tr>
<td>$f_X&lt;2&gt;$: 1</td>
<td>state:S2</td>
</tr>
<tr>
<td>$f_X&lt;3&gt;$: 1</td>
<td>th$NTK2$ _Pretrip: 0</td>
</tr>
<tr>
<td>$f_X&lt;4&gt;$: 1</td>
<td>th$NTK2$ _Pretrip: 0</td>
</tr>
<tr>
<td>$f_X&lt;5&gt;$: 0</td>
<td>timer$NTK2$: T5</td>
</tr>
<tr>
<td>$f_X&lt;6&gt;$: 1</td>
<td>timer: T5</td>
</tr>
</tbody>
</table>

th$X$ _Pretrip: 0

<table>
<thead>
<tr>
<th>$f_X&lt;0&gt;$: 0</th>
<th><strong>&lt;State 7&gt;</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_X&lt;1&gt;$: 1</td>
<td>state$NTK2:S0$</td>
</tr>
<tr>
<td>$f_X&lt;2&gt;$: 1</td>
<td>state:S2</td>
</tr>
<tr>
<td>$f_X&lt;3&gt;$: 0</td>
<td>th$NTK2$ _Pretrip: 0</td>
</tr>
<tr>
<td>$f_X&lt;4&gt;$: 1</td>
<td>th$NTK2$ _Pretrip: 0</td>
</tr>
<tr>
<td>$f_X&lt;5&gt;$: 0</td>
<td>timer$NTK2$: T0</td>
</tr>
<tr>
<td>$f_X&lt;6&gt;$: 0</td>
<td>timer: T0</td>
</tr>
</tbody>
</table>

th$X$ _Pretrip: 0
### Equivalence checking result representation

<table>
<thead>
<tr>
<th>$f_X&lt;0&gt;$</th>
<th>0</th>
<th>&lt;State 6&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_X&lt;1&gt;$</td>
<td>1</td>
<td>state $\text{NTK2:S0}$</td>
</tr>
<tr>
<td>$f_X&lt;2&gt;$</td>
<td>1</td>
<td>state: S2</td>
</tr>
<tr>
<td>$f_X&lt;3&gt;$</td>
<td>1</td>
<td>th_Prev_X_Pretrip $\text{NTK2:0}$</td>
</tr>
<tr>
<td>$f_X&lt;4&gt;$</td>
<td>1</td>
<td>th_Prev_X_Pretrip: 0</td>
</tr>
<tr>
<td>$f_X&lt;5&gt;$</td>
<td>0</td>
<td>timer $\text{NTK2:T5}$</td>
</tr>
<tr>
<td>$f_X&lt;6&gt;$</td>
<td>1</td>
<td>timer: T5</td>
</tr>
<tr>
<td>th_X_Pretrip</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$f_X&lt;0&gt;$</th>
<th>0</th>
<th>&lt;State 7&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_X&lt;1&gt;$</td>
<td>1</td>
<td>state $\text{NTK2:S0}$</td>
</tr>
<tr>
<td>$f_X&lt;2&gt;$</td>
<td>1</td>
<td>state: S2</td>
</tr>
<tr>
<td>$f_X&lt;3&gt;$</td>
<td>0</td>
<td>th_Prev_X_Pretrip $\text{NTK2:0}$</td>
</tr>
<tr>
<td>$f_X&lt;4&gt;$</td>
<td>1</td>
<td>th_Prev_X_Pretrip: 0</td>
</tr>
<tr>
<td>$f_X&lt;5&gt;$</td>
<td>0</td>
<td>timer $\text{NTK2:T0}$</td>
</tr>
<tr>
<td>$f_X&lt;6&gt;$</td>
<td>0</td>
<td>timer: T0</td>
</tr>
<tr>
<td>th_X_Pretrip</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
Comparison II

- Model checking result representation
  - Similar with equivalence checking

```plaintext
vis release 2.0 (compiled Sat Jun 14 12: ...)
vis> read_blif_mv test.mv
vis> flatten_hierarchy
vis> static_order
vis> build_partition_mdds
vis> model_check -d 2 -i property.ctl
# MC: formula failed --- AG (((BB.BB.AAA.timer<0>=1 *
...
```
• Model checking result representation
  - Similar with equivalence checking

--Goes to state 7:
AA.f_X_Prev<0>:1
AA.f_X_Prev<1>:1
BB.BB.AAA.timer<0>:1
BB.EE.status<0>:1
BB.Prev_status<0>:1
--On input:
<Unchanged>

--Goes to state 8:
BB.BB.AAA.timer<0>:0
BB.BB.AAA.timer<1>:1
--On input:
f_X_Raw<4>:0
f_X_Raw<5>:0
VIS Analyzer really works well?

- Did not apply on a real project
  - The KNICS project is ended before the completion of the VIS Analyzer development
- But appears promising
- Any suggestions will be welcomed
Outline

• Motivation
• Our Solution
• Comparisons
• Conclusions & Future Work
Conclusions & Future Work

• Assistant tool for domain experts

• VIS Analyzer enhances usability of the VIS with:
  - GUI
  - Automation of verification process
  - Visualization of full verification result

• Currently focusing on more intuitive & informative display methods
Thank you