Propositional Approximations for Bounded Model Checking of Partial Circuit Designs

Ralf Wimmer

(joint work with Bernd Becker, Marc Herbstritt, Natalia Kalinnik, Matthew Lewis, Juri Lichtner, Tobias Nopper)

Albert-Ludwigs-University
Freiburg im Breisgau, Germany

Outline

1 Preliminaries
   - Bounded Model Checking
   - Relational vs. Functional Representation

2 BMC with Black Boxes
   - Black Boxes
   - Three-valued Logic
   - Impact on 01X-BMC

3 Experimental Results
Preliminaries
Invariant Properties

**Given**
- Sequential circuit \( SK = (x, s, \delta, \lambda) \)
  - \( x \) inputs
  - \( s \) state bits
  - \( \delta \) transition functions
  - \( \lambda \) output functions
- invariant property \( \phi \)

**Question**
- Does \( \phi \) hold in all reachable states?
Bounded Model Checking

Method:

- Formulate the reachability of a state violating the invariant property within $k$ steps as a satisfiability problem:

\[
BMC(k) = I(s^0) \cdot \bigwedge_{i=0}^{k-1} T(s^i, s^{i+1}) \cdot P(s^k)
\]

- $I(s^0) = \text{true}$ iff $s^0$ is the initial state
- $T(s^i, s^{i+1}) = \text{true}$ iff there is a transition from state $s^i$ to $s^{i+1}$.
- $P(s^k) = \text{true}$ iff $s^k$ satisfies the invariant property.
Relational vs. Functional Transition Representation

**Transition Relation**

- **Local** transition relation:

  \[ T_i := (s'_i \equiv \delta_i(s, x)) \]

- **Global** transition relation:

  \[
  T(s, x, s') := \bigwedge_{i=0}^{n-1} T_i(s, x, s'_i)
  = \bigwedge_{i=0}^{n-1} (s'_i \equiv \delta_i(s, x))
  \]
Relational vs. Functional Transition Representation

Transition Function

$$\delta^k_I : B^p \times (B^n)^k \rightarrow B$$

that is inductively defined by:

$$\delta^0_I(s^0) := s^0$$

$$\delta^k_I(s^0, x^0, \ldots, x^{(k-1)}) := \delta_I(\delta^{k-1}_0(s^0, x^0, \ldots, x^{k-2}),$$

$$\ldots, \delta^{(p-1)}_I(s^0, x^0, \ldots, x^{k-2}), x^{k-1})$$,
Relational vs. Functional Transition Representation

**Relational transition representation**

\[
\bigwedge_{i=0}^{n-1} (s'_i \equiv \delta_i(s, x))
\]

\[
\begin{align*}
\delta(s^0, x^0) & \equiv s^1 \\
\delta(s^1, x^1) & \equiv s^2 \\
\delta(s^{k-1}, x^{k-1}) & \equiv s^k
\end{align*}
\]

**Functional transition representation**

\[
\delta_f(\delta^{k-1}_0, \ldots, \delta^{k-1}_{(p-1)}, x^{k-1})
\]

\[
\begin{align*}
\delta(s^0, x^0) & \rightarrow \delta(s^1, x^1) \\
\delta(s^{k-1}, x^{k-1})
\end{align*}
\]
Relational vs. Functional Transition Representation

Bounded Model Checking

- **Relational TR:**
  \[ \text{BMC}^{\text{rel}}(k) := I(s^0) \cdot T^k(s^0, x^0, s^1, \ldots, x^{k-1}, s^k) \cdot \overline{P(s^k)} \]

- **Functional TR:**
  \[ \text{BMC}^{\text{func}}(k) := I(s^0) \cdot \overline{P(\delta^k(s^0, x^0, \ldots, x^{k-1}))} \]

Both formulae are satisfiability equivalent for circuits.
BMC with Black Boxes
What are Black Boxes?

- Parts of a digital system are not available (yet):
  - design not finished yet
  - irrelevant parts removed for efficiency reasons
  - fault localization

- Outputs of a blackbox have an unknown value \( (X) \)
  \( \Rightarrow \) Three-valued logic
Most commonly used encoding [Jain et al., 2000]:

\[
\begin{array}{c|ccc}
 a & b & 0 & 1 & X \\
\hline
 0 & 0 & 0 & 0 \\
 1 & 0 & 1 & X \\
 X & 0 & X & X \\
\end{array}
\]

\[
\begin{align*}
0_{01X} & := (1, 0) \\
1_{01X} & := (0, 1) \\
X_{01X} & := (0, 0)
\end{align*}
\]

\[
\begin{align*}
\text{AND}_{01X} ((a_0, a_1), (b_0, b_1)) & := (a_0 + b_0, a_1 \cdot b_1) \\
\text{OR}_{01X} ((a_0, a_1), (b_0, b_1)) & := (a_0 \cdot b_0, a_1 + b_1) \\
\text{NOT}_{01X} ((a_0, a_1), (b_0, b_1)) & := (a_1, a_0)
\end{align*}
\]

(1, 1) illegal value
**Impact on 01X-BMC**

- Initial state: $s_0 = 0$, $s_1 = 0$
- Invariant property: $AG(\neg s_0 \land \neg s_1)$, i.e., $\overline{P(s_0, s_1)} = s_0 \lor s_1$
- Transition functions:
  
  $\delta_0(s_0, s_1, x) = X \lor s_1$
  
  $\delta_1(s_0, s_1, x) = 1$
Impact on 01X-BMC

Using 01X-encoding, we obtain:

\[ \text{BMC}^{f,\text{enc}}(1) = (s_{0,1}^0 + s_{1,1}^0, s_{0,0}^0 \cdot s_{1,0}^0) \equiv 1_{01X} = (0, 1) \]
Impact on 01X-BMC

Using 01X-encoding, we obtain:

\[ \text{BMC}^{r,\text{enc}}(1) = (s_{0,1}^{0} + s_{1,1}^{0} + s_{0,0}^{1} \cdot s_{0,1}^{1} + s_{1,0}^{1}, \ s_{0,0}^{0} \cdot s_{1,0}^{0} \cdot s_{0,1}^{0} \cdot s_{1,1}^{1} \cdot s_{1,1}^{1}) \overset{1}{=} 1_{01X} = (0, 1) \]
Relational vs. Functional BMC

### Functional TR

\[
BMC_{f,enc}^f(1) = (s_{0,1}^0 + s_{1,1}^0, s_{0,0}^0 \cdot s_{1,0}^0) \overset{!}{=} (0, 1)
\]

Solution found!

\[s_{0,0} = s_{1,0} = 1, \quad s_{0,1} = s_{1,1} = 0, \text{ i.e. } s_0 = 0, \quad s_1 = 0.\]

### Relational TR

\[
BMC_{r,enc}^r(1) = (s_{0,1}^0 + s_{1,1}^0 + s_{0,0}^1 \cdot s_{0,1}^1 + s_{1,0}^1,
\]

\[
s_{0,0}^0 \cdot s_{1,0}^0 \cdot s_{0,1}^1 \cdot s_{1,1}^0 \cdot s_{1,1}^1) \overset{!}{=} (0, 1)
\]

No solution!

In the first part, we would have to set \(s_{1,1}^0 = 0\) and in the second part, \(s_{1,1}^1 = 1\).
Relational vs. Functional BMC

**Functional TR**

\[ BMC^{f, enc}(1) = (s^{0}_{0,1} + s^{0}_{1,1}, s^{0}_{0,0} \cdot s^{0}_{1,0}) \Downarrow (0, 1) \]

Solution found!

\( s_{0,0} = s_{1,0} = 1, s_{0,1} = s_{1,1} = 0, \text{ i.e. } s_0 = 0, s_1 = 0. \)

**Relational TR**

\[ BMC^{r, enc}(1) = (s^{0}_{0,1} + s^{0}_{1,1} + s^{1}_{0,0} \cdot s^{1}_{0,1} + s^{1}_{1,0}, s^{0}_{0,0} \cdot s^{1}_{1,0} \cdot s^{0}_{0,1} \cdot s^{1}_{1,1}) \Downarrow (0, 1) \]

No solution!

In the first part, we would have to set \( s^{0}_{1,1} = 0 \) and in the second part, \( s^{0}_{1,1} = 1. \)
The Cause of the Effect (1)

Transition relation:

\[
\bigwedge_{i=0}^{n-1} (s'_i \equiv \delta_i(s, x))
\]

\[X_{01X} \equiv X_{01X}?\]

Consider: \(x \equiv y\) (short for: \(s'_i \equiv \delta_i(s, x)\))

Two-valued encoding:

\[
((x_0 \cdot x_1 + x_0 \cdot y_1 + x_1 \cdot y_0 + y_0 \cdot y_1), (x_0 \cdot y_0 + x_1 \cdot y_1))
\]

\((*)\)

For \(x = y = X_{01X} \Rightarrow (x_0, x_1) = (y_0, y_1) = (0, 0): (*)\) results in \((0, 0) = X_{01X}\).

Observation

*Abuse of the equivalence operator \(\equiv\) (i.e., \(\oplus\)) disables propagation of \(X_{01X}\) for latch values.
"Fixed edges":
Solid (black) edges exist independently of the content of the blackbox.

"Possible edges":
Dashed (blue) edges are an over-approximation of the edges which may exist depending on the blackbox implementation.
The Cause of the Effect (3)

- Relational TR uses only fixed (= solid) edges.
- **No counterexample found!**

- Functional TR uses all edges.
- Counterexample:
  \[ x^0 = 1, \, x^1 = 0, \, x^2 = 1. \]

\[
\begin{align*}
00 & \rightarrow \{10, 11\} \rightarrow \{01, 10\} \rightarrow 11 \\
\end{align*}
\]

“Uniform counterexample”
The Cause of the Effect (3)

- Relational TR uses only fixed (= solid) edges.
- No counterexample found!

- Functional TR uses all edges.
- Counterexample: \( x^0 = 1, x^1 = 0, x^2 = 1. \)

\[
\begin{align*}
00 & \xrightarrow{1} \{10, 11\} \xrightarrow{0} \{01, 10\} \xrightarrow{1} 11
\end{align*}
\]

“Uniform counterexample”
Experimental Results
Implementation

- Implementation in C++
- And-Inverter-Graphs (AIGs) for composition of transition functions / relations
- Experiments performed on AMD Opteron Dual Processor, 2.6 GHz, 4 GB main memory
## Experimental Results (1)

<table>
<thead>
<tr>
<th>prop.</th>
<th>#ex.</th>
<th>CPU time</th>
<th>#CE</th>
<th>AIG size</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>rel.</td>
<td>func.</td>
<td>rel.</td>
<td>func.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MiniSAT</td>
<td>Tot.</td>
<td>MiniSAT</td>
<td>Tot.</td>
</tr>
<tr>
<td>1</td>
<td>260</td>
<td>2.01</td>
<td>39.40</td>
<td>0.20</td>
<td>9.83</td>
</tr>
<tr>
<td>2</td>
<td>450</td>
<td>1.97</td>
<td>41.95</td>
<td>0.10</td>
<td>8.19</td>
</tr>
<tr>
<td>3</td>
<td>300</td>
<td>0.85</td>
<td>20.85</td>
<td>0.05</td>
<td>4.27</td>
</tr>
<tr>
<td>4</td>
<td>340</td>
<td>1.32</td>
<td>26.59</td>
<td>0.07</td>
<td>5.96</td>
</tr>
<tr>
<td>5</td>
<td>650</td>
<td>2.96</td>
<td>64.92</td>
<td>0.17</td>
<td>13.12</td>
</tr>
</tbody>
</table>

- Functional approach considerably faster
- Functional approach detects more counterexamples
- CNF of functional approach is much smaller:
  - Cone-of-influence reduction performed for free.
Experimental Results

VLIW ALU

<table>
<thead>
<tr>
<th>width</th>
<th>#ex.</th>
<th>rel. MiniSAT</th>
<th>CPU time rel.</th>
<th>func. MiniSAT</th>
<th>Tot.</th>
<th>CPU time func. Tot.</th>
<th>#CE rel.</th>
<th>func.</th>
<th>AIG size rel.</th>
<th>func.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4</td>
<td>0.003</td>
<td>0.05</td>
<td>0.003</td>
<td>0.04</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>7,954</td>
<td>5,032</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>0.006</td>
<td>0.09</td>
<td>0.009</td>
<td>0.08</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13,566</td>
<td>8,712</td>
</tr>
<tr>
<td>16</td>
<td>4</td>
<td>0.089</td>
<td>0.55</td>
<td>0.265</td>
<td>0.75</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>47,238</td>
<td>30,792</td>
</tr>
<tr>
<td>24</td>
<td>4</td>
<td>0.070</td>
<td>0.90</td>
<td>0.466</td>
<td>1.45</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>69,686</td>
<td>45,512</td>
</tr>
<tr>
<td>32</td>
<td>4</td>
<td>0.112</td>
<td>1.48</td>
<td>0.938</td>
<td>2.39</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>92,134</td>
<td>60,232</td>
</tr>
<tr>
<td>40</td>
<td>4</td>
<td>0.275</td>
<td>2.02</td>
<td>5.469</td>
<td>7.59</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>114,582</td>
<td>74,952</td>
</tr>
<tr>
<td>48</td>
<td>4</td>
<td>0.340</td>
<td>2.66</td>
<td>5.734</td>
<td>8.56</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>137,030</td>
<td>89,672</td>
</tr>
<tr>
<td>64</td>
<td>4</td>
<td>0.674</td>
<td>3.16</td>
<td>16.489</td>
<td>18.63</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>181,926</td>
<td>119,112</td>
</tr>
</tbody>
</table>
VLIW ALU: Results

- Both approaches detect a counterexample:
  - Counterexample uses only fixed edges.
- Relational approach faster:
  - Functional approach gets stuck on paths with possible edges.
- Difference in size is smaller since the property depends on all outputs.
Conclusions

- **VIS-benchmarks:**
  - Functional TR more accurate (210%) than relational TR
  - Functional TR faster (390%) than relational TR

- **VLIW ALU:**
  - Relational TR accurate enough (no advantage for functional TR)
  - Relational TR faster (300%) than functional TR

- Relational and functional TR are “orthogonal” \( \Rightarrow \) toolbox for 01X-BMC

- Functional TR inbetween relational TR and QBF-approach wrt. accuracy