The impact of EFSM Composition on Functional ATPG

Davide Bresolin, Giuseppe Di Guglielmo, Franco Fummi, Graziano Pravadelli, Tiziano Villa

University of Verona
Department of Computer Science
Outline
Introduction
Laerte++: a functional ATPG

- Extended Event FSM (EEFSM) to model the DUT
- Constraint Logic Programming (CLP) engine to generate test vectors
- Two step algorithm: random walk + backjumping
- Transition and fault coverage to evaluate the quality of the test vectors
The EEFSM model

• EEFSM are I/O FSM augmented by a set of *registers* and of I/O *events*
• I/O events are used to model the clock and the *sensitivity list* construct of HDL
• Activation of a transition depends on the *state*, on *inputs*, on *events*, and on the value of the *registers*. 
An EEFSM example

\[ M : \text{process}(i) \]
\[ \text{begin} \]
\[ \text{case state is} \]
\[ \text{when } s_0 => \]
\[ \text{if } \alpha(i) \text{ then} \]
\[ A_1(o); \text{state := } s_1; \]
\[ \text{else} \]
\[ A_2(o); \text{state := } s_0; \]
\[ \text{end if;} \]
\[ \text{when } s_1 => \]
\[ \text{if } \beta(i) \text{ then} \]
\[ B_1(o); \text{state := } s_0; \]
\[ \text{else} \]
\[ B_2(o); \text{state := } s_1; \]
\[ \text{end if;} \]
\[ \text{end case;} \]
\[ \text{end process;} \]
Advantages of EEFSM

• They allow for more compact representations (no space state explosion)
• The event-based semantics make the asynchronous composition cleaner
• Events are used to activate only the components that should trigger a transition
Hard and Easy transitions

• Transitions that depends only on primary inputs are Easy-to-traverse (ETT) transitions.

• Transition that depends on the value of the registers are Hard-to-traverse (HTT) and should be treated with special care.
Laerte++ engine

Laerte++
Learning

Random walk

Backjumping

DUT HDL model

EFSM extraction

To fire easy-to-traverse (ETT) transitions

To fire hard-to-traverse (HTT) transitions
Bottlenecks

• With multi-process design, scheduling can be problematic
• DUTs with a large number of HTT have a low transition coverage
• Invocation of the CLP engine is time consuming
A possible solution

- Compose the processes into a single EEFSM
- Scheduling is simplified
- Some HTT become Easy
- Less CLP invocations
- Higher transition coverage
Serial composition

• Outputs of $\mathcal{M}_1$ are inputs of $\mathcal{M}_2$
• We do not allow the two EEFSM to update R simultaneously
• $\mathcal{M}_2$ fires a transition only if F is in its sensitivity list
Parallel composition

- $M_1$ and $M_2$ share the same inputs
- $R$ cannot be updated simultaneously
- Outputs are computed in parallel
- Transitions are not necessarily synchronized
Feedback composition

- Some outputs of $M_1$ are inputs of $M_2$ (and vice versa)

- This composition is well-defined only if there are no algebraic loops in the dependencies
Hard transitions became Easy!

\[ M_1 ; M_2 \]

\[ t_1 \]

\[ t_{12} \]

\[ t_{13} \]

\[ I \neq 10 / S \leq I ; \]

\[ I = 100 / ... ; \]

\[ I = 10 \& I = 100 / S \leq I ; ... ; \]

\[ (I \neq 10) \& (I = 0) / S \leq I ; ... \]
Composing Processes
Experimental results

- Three industrial benchmarks:
  - *Vr01*: module of a face recognition system
  - *Ecc1, Ecc2*: ECC code of a 16bit page of data

<table>
<thead>
<tr>
<th>DUT</th>
<th>PI</th>
<th>PO</th>
<th>P</th>
<th>FF</th>
<th>Gate</th>
<th>Trn</th>
<th>t(s)</th>
<th>BC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vr01</td>
<td>65</td>
<td>16</td>
<td>8</td>
<td>73</td>
<td>615</td>
<td>69</td>
<td>7.1</td>
<td>2168</td>
</tr>
<tr>
<td>Ecc1</td>
<td>25</td>
<td>32</td>
<td>9</td>
<td>79</td>
<td>703</td>
<td>17</td>
<td>1.7</td>
<td>1022</td>
</tr>
<tr>
<td>Ecc2</td>
<td>55</td>
<td>32</td>
<td>7</td>
<td>88</td>
<td>832</td>
<td>24</td>
<td>2.4</td>
<td>1032</td>
</tr>
</tbody>
</table>
### Experimental Results: vr01

<table>
<thead>
<tr>
<th>D</th>
<th>T%</th>
<th>F%</th>
<th>t(s.)</th>
<th>TV</th>
<th>CSI</th>
<th>CST(s.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>80</td>
<td>54.10</td>
<td>228.38</td>
<td>15000</td>
<td>584935</td>
<td>182.7</td>
</tr>
<tr>
<td>1</td>
<td>80</td>
<td>54.10</td>
<td>201.35</td>
<td>15000</td>
<td>542523</td>
<td>161.1</td>
</tr>
<tr>
<td>2</td>
<td>80</td>
<td>54.10</td>
<td>187.34</td>
<td>15000</td>
<td>489031</td>
<td>159.3</td>
</tr>
<tr>
<td>3</td>
<td>93</td>
<td>71.80</td>
<td>134.92</td>
<td>15000</td>
<td>439413</td>
<td>103.4</td>
</tr>
<tr>
<td>4</td>
<td>93</td>
<td>71.80</td>
<td>132.33</td>
<td>15000</td>
<td>419321</td>
<td>100.0</td>
</tr>
<tr>
<td>5</td>
<td>93</td>
<td>71.80</td>
<td>103.43</td>
<td>15000</td>
<td>371391</td>
<td>73.1</td>
</tr>
<tr>
<td>6</td>
<td>100</td>
<td>98.90</td>
<td>13.93</td>
<td>256</td>
<td>16129</td>
<td>10.4</td>
</tr>
<tr>
<td>7</td>
<td>100</td>
<td>98.90</td>
<td>13.59</td>
<td>263</td>
<td>16129</td>
<td>10.1</td>
</tr>
</tbody>
</table>
## Experimental Results: *ecc1, ecc2*

<table>
<thead>
<tr>
<th>D</th>
<th>T%</th>
<th>F%</th>
<th>t(s.)</th>
<th>TV</th>
<th>CSI</th>
<th>CST(s.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>100</td>
<td>87.7</td>
<td>4.852</td>
<td>884</td>
<td>11680</td>
<td>3.785</td>
</tr>
<tr>
<td>8</td>
<td>100</td>
<td>87.7</td>
<td>3.612</td>
<td>973</td>
<td>5162</td>
<td>2.998</td>
</tr>
</tbody>
</table>

**Ecc01**

<table>
<thead>
<tr>
<th>D</th>
<th>T%</th>
<th>F%</th>
<th>t(s.)</th>
<th>TV</th>
<th>CSI</th>
<th>CST(s.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>71</td>
<td>32.1</td>
<td>312.32</td>
<td>15000</td>
<td>612042</td>
<td>291.30</td>
</tr>
<tr>
<td>8</td>
<td>100</td>
<td>97.4</td>
<td>17.13</td>
<td>315</td>
<td>48984</td>
<td>16.98</td>
</tr>
</tbody>
</table>

**Ecc02**
Conclusions and Future Work

• EFSM composition has proved to be a valuable approach
• We are testing this approach on more complex case studies