ILP Models for the Synthesis of Asynchronous Control Circuits

Josep Carmona and Jordi Cortadella
Universitat Politècnica de Catalunya
Outline

• Synthesis of Async. Circuits: VME example
  – State space explosion problem
  – Structural methods
• Petri net methods. ILP for:
  – State encoding verification
  – Decomposition of initial specification
• Design Flow
• Synthesis Example
Synthesis of Async. Controllers

HDL

CSP, Tangram, Balsa, Verilog...

Graph Model

Petri nets, Automata, ...

Boolean Gates

Complex gates, two-level, ...

This work

CMOS, NMOS, FPGAs, ...

Physical Implementation
Synthesis of Async. Controllers

This thesis
Device

Data Transceiver

Bus

VME Bus Controller

DSr LDS LDTACK DDTACK

Read Cycle
Boolean equations:

\[ \text{LDS} = D \lor \text{csc} \]
\[ \text{DTACK} = D \]
\[ D = \text{LDTACK} \]
\[ \text{csc} = \text{DSr} \]
ILP Models

• Motivation: avoid the computation of the state space for the checking of USC/CSC
  – Linear Algebra (Integer Linear Programming models)
  – USC/CSC fast conservative method
  – Novel method for decomposition
Previous Work

• David Cells [David] [Varshavsky]
• Lock Theory [Vanbekbergen] [Ykman-Couveur, Lin, Goossens, De Man]
• ILP & Implicit places [Colom, García-Vallés]
• Unfoldings & ILP [Khomenko, Koutny, Yakovlev]
• Unfoldings & SAT [Khomenko, Koutny, Yakovlev]
Marking equation

Incidence matrix

\[
\begin{pmatrix}
-1 & 0 & 0 & 0 & 1 & -1 & 0 \\
1 & 0 & -1 & 0 & 0 & 0 & 0 \\
1 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & -1 \\
0 & 0 & 0 & -1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & -1 & 0 & 1 \\
0 & 1 & 0 & 1 & -1 & 0 & 0
\end{pmatrix}
\]
Marking equation

\[ M' = M + Ax \]

### Necessary reachability condition, but not sufficient.
The encoding problem
Checking Unique State Coding

$M_0 \xrightarrow{x} M_1 \xrightarrow{z} M_2$

$z = \{a^+ \ b^+ \ a^- \ b^-\}$

- $M_1$ and $M_2$ have the same binary code
  ($z$ must be a complementary set of transitions)

- $M_1$ and $M_2$ must be different markings
  (they must differ in at least one place)
Checking Unique State Coding

\[ M_0 \xrightarrow{x} M_1 \xrightarrow{z = \{a+ b+ a- b-\}} M_2 \]

ILP formulation:

\[
\begin{align*}
M_1 &= M_0 + Ax \\
M_2 &= M_1 + Az \\
bal(z) \\
M_1 &\neq M_2 \\
x, z, M_1, M_2 &\geq 0
\end{align*}
\]

\[ bal(z) \equiv \forall a: #(a+) = #(a-) \]
### Some experiments (USC)

<table>
<thead>
<tr>
<th>benchmark</th>
<th></th>
<th>P</th>
<th></th>
<th></th>
<th>T</th>
<th></th>
<th></th>
<th>signals</th>
<th></th>
<th>CLP</th>
<th></th>
<th>ILP</th>
</tr>
</thead>
<tbody>
<tr>
<td>PpWk(3,9)</td>
<td>106</td>
<td>56</td>
<td>28</td>
<td>10.53</td>
<td>0.03</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PpWk(3,12)</td>
<td>142</td>
<td>74</td>
<td>37</td>
<td>876.63</td>
<td>0.05</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PpWkCsc(3,9)</td>
<td>108</td>
<td>56</td>
<td>28</td>
<td>2002.29</td>
<td>0.67</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PpWkCsc(3,12)</td>
<td>144</td>
<td>74</td>
<td>37</td>
<td>time</td>
<td>1.17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PpArb(3,9)</td>
<td>128</td>
<td>72</td>
<td>34</td>
<td>0.01</td>
<td>0.06</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PpArb(3,12)</td>
<td>164</td>
<td>90</td>
<td>43</td>
<td>0.00</td>
<td>0.08</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PpArbCsc(3,9)</td>
<td>131</td>
<td>72</td>
<td>34</td>
<td>time</td>
<td>1.05</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PpArbCsc(3,12)</td>
<td>167</td>
<td>90</td>
<td>43</td>
<td>time</td>
<td>1.69</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **CLP** [Khomenko, Koutny, Yakovlev]:
  - Partial order approach (Unfoldings)
  - Integer Linear Programming
Checking Complete State Coding

ILP formulation:

\[ M_1 = M_0 + Ax \]
\[ M_2 = M_1 + Az \]
\[ bal(z) \]
\[ M_1 \in ER(a^*) \]
\[ M_2 \notin ER(a^*) \]
\[ x, z, M_1, M_2 \geq 0 \]

\( n \) ILP problems must be solved
(\( n \) is the number of transitions with label \( a^* \))
### Some experiments (CSC)

| benchmark       | |P| | |T| | |signals| | |CLP| | |SAT| | |ILP| |
|-----------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Tangram(3,2)    | 142 | 92 | 38 | 0.01 | 0.01 | 1.08 |
| Tangram(4,3)    | 321 | 202 | 83 | 0.06 | 0.04 | 9.00 |
| Art(10,9)       | 216 | 198 | 99 | 0.00 | 0.42 | 0.06 |
| Art(20,9)       | 436 | 398 | 199 | 5.00 | 10.35 | 0.24 |
| Art(30,9)       | 656 | 598 | 299 | 38.02 | 81.82 | 0.56 |
| Art(40,9)       | 876 | 798 | 399 | 138.04 | 264.57 | 0.92 |
| Art(50,9)       | 1096 | 998 | 499 | 377.00 | 630.41 | 1.46 |
| ArtCsc(10,9)    | 752 | 630 | 315 | 14 m | 3 m |
| ArtCsc(20,9)    | 1532 | 1270 | 635 | time | mem | 27 m |
| ArtCsc(30,9)    | 2312 | 1910 | 955 | time | mem | 1.5 h |
| ArtCsc(40,9)    | 3092 | 2550 | 1275 | time | mem | 3.5 h |
| ArtCsc(50,9)    | 3872 | 3190 | 1595 | time | mem | 7 h |

- **SAT (Khomenko et al)**:
  - Partial order approach (Unfoldings)
  - Satisfiability solver
Checking the CSC support for a signal

Let $\Sigma$ be the set of signals and $\Sigma'$ a potential support for $a$. Let $z'$ be the projection of $z$ onto $\Sigma'$. $\Sigma'$ is a valid support for $a$ if the following model has no solution:

\[
\begin{align*}
M_1 &= M_0 + Ax \\
M_2 &= M_1 + Az \\
\text{bal}(z') \\
M_1 &\in \text{ER}(a^*) \\
M_2 &\not\in \text{ER}(a^*) \\
x, z, M_1, M_2 &\geq 0
\end{align*}
\]
Algorithm to find the support

\[ z' := \{a\} \cup \{\text{trigger signals of } a\}; \]

\text{forever}

\[ z'' := \text{ILP\_check\_support} (\text{STG, } a, z'); \]

\text{if } z'' = 0 \text{ then return } z';

\[ z' := z' \cup \{\text{unbalanced signals in } z''\}; \]

\text{end \text{forever}}
CSC support for $x_4$

- **Support for $x_4$**
- **Trigger signals. ILP Model feasible => No CSC**
- **$y_4$ unbalanced -> balanced. ILP model infeasible => CSC!**
CSC support for $x_4$

Synthesis of $x_4$:
$x_4, x_3, x_5, z, y_4$
Synthesis Flow

• Motivation: complete design flow for the synthesis of asynchronous control circuits
  – Linear Algebra & Graph theory methods
  – Specially suited for well-structured specifications
  – Conservative approach
  – Results comparable to state-based methods
Synthesis Flow

STG

structural encoding

STG with CSC

structural transformations

optimized STG

support for a

support for b

... support for z

STG for a

STG for b

STG for z

projection

logic synthesis (petrify)

circuit for a

circuit for b

circuit for z
Synthesis example
Experiments (Support + Synthesis)

- Recent work on synthesis with unfoldings and SAT [Khomenko, Koutny, Yakovlev]

| benchmark          | States | |P| |T| |signals| | Literals Petrify | ILP | CPU Petrify | ILP |
|--------------------|--------|----------|----------|----------|----------|----------|----------|----------|----------|
| PpWkCsc(2,6)       | 8192   | 47       | 26       | 19       | 57       | 57       | 5        | 1        |
| PpWkCsc(2,9)       | 524288 | 71       | 38       | 19       | 87       | 87       | 49       | 2        |
| PpWkCsc(3,9)       | 2.7 x 10E7 | 106 | 56       | 28       | ?        | 130      | mem      | 3        |
| PpWkCsc(3,12)      | 2.2 x 10E11 | 142 | 74       | 37       | ?        | 117      | time     | 3        |
| PpArbCsc(2,6)      | 61440  | 62       | 36       | 17       | 77       | 77       | 21       | 83       |
| PpArbCsc(2,9)      | 3.9 x 10E6 | 110 | 60       | 29       | 107      | 107      | 185      | 59       |
| PpArbCsc(3,9)      | 3.3 x 10E9 | 131 | 72       | 34       | 163      | 165      | 10336    | 289      |
| PpArbCsc(3,12)     | 1.7 x 10E12 | 167 | 90       | 43       | ?        | 210      | time     | 608      |
| TangramCsc(3,2)    | 426    | 142      | 92       | 38       | 97       | 103      | 56       | 146      |
| TangramCsc(4,3)    | 9258   | 321      | 202      | 83       | ?        | 247      | mem      | 2 h      |
Conclusions

• CAD tool with a sound theoretical basis for the synthesis of asynchronous circuits.
• Most of the techniques can be included in other approaches for synthesis/verification.
• Techniques specially suited for well-structured specifications (ideally obtained from HDL programs)
• Experimental results show good performance and quality even for large systems