ILP-based Modulo Scheduling for High-level Synthesis

Julian Oppermann  Andreas Koch  Melanie Reuter-Oppermann  Oliver Sinnen
Embedded Systems & Applications Group  Technische Universität Darmstadt, Germany
Discrete Optimization & Logistics Group  Karlsruhe Institute of Technology, Germany
Parallel & Reconfigurable Computing Group  University of Auckland, New Zealand

Loop pipelining
- Start new loop iterations after a fixed number of time steps, called **Initiation Interval (II)**
- Partially overlapping execution of subsequent loop iterations $\rightarrow$ **resource constraints on congruence classes (modulo II)** of time steps
- Primary objective: Find schedule with **smallest feasible II**, subject to resource constraints and inter-iteration dependencies

Approaches (based on Integer Linear Programs)
- Scheduling without resource constraints is easy, can be done in polynomial time with a System of Difference Constraints (SDC)
- Approaches differ in the modelling of resource constraints
  - There are $A_k$ units of resource kind $k$ and II-1 congruence classes
  - Each resource instance can be used at most by one operation in each congruence class

Moovac (novel)
- **Exact** formulation based on an efficient task scheduler
- Uses integer variables to model operations’ start times
- Resource assignment modelled by
  - integer variables
    - $r_i$: resource instance ID
    - $m_i$: congruence class ID
  - binary **overlap** variables
    - $\epsilon_{ij} = 1$ iff. $r_i < r_j$
    - $\mu_{ij} = 1$ iff. $m_i < m_j$
- No resource conflict iff. $\epsilon_{ij} + \epsilon_{ji} + \mu_{ij} + \mu_{ji} \geq 1$

Evaluation
- Schedulers implemented with CPLEX 12.6.3, ran single-threadedly on Intel Xeon E5-2667’s at 3.3 GHz
- Time limit of 5 min or 60 min per candidate II $\rightarrow$ increment II if instance is shown to be infeasible, or no solution was found within time budget
- Attempted to schedule 225 loops from CHStone and MachSuite

Scheduling time - 5 min time limit

<table>
<thead>
<tr>
<th>Graphs</th>
<th>Moovac</th>
<th>Modulo SDC</th>
<th>Eichenberger’s ILP</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td>225</td>
<td>489</td>
<td>96</td>
</tr>
<tr>
<td>small</td>
<td>203</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>large</td>
<td>22</td>
<td>486</td>
<td>96</td>
</tr>
</tbody>
</table>

Moovac is surprisingly fast; Moovac + M. SDC synergistically is even faster: **429 min**

Result quality - 5 min time limit
- Compared to time-limited Moovac, Modulo SDC finds schedules with...
  - the same II for 217 of 225 graphs
  - a worse II for 6 of 225 graphs
  - a better II for 2 of 225 graphs

Fruitless attempts dominate overall time. Heuristic can struggle with small graphs.