A Predictable Multiprocessor Design-Flow for Streaming Applications Presentation

Sander Stuijk, AmirHossein Ghariani, Twan Basten, Marc Geilen, Bart Theelen and Henk Corporaal
1st Workshop on Mapping of Applications to MPSoCs
SDFG-based MP-SoC design-flow

Streaming application SDFG
Throughput constraint
Architecture platform

Predictable design flow

MP-SoC configuration
Streaming application SDFG

Throughput constraint: 0.07 firings / time-unit

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>7</td>
<td>200</td>
<td>200</td>
<td>d_1</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>-</td>
<td>350</td>
<td>-</td>
<td>d_2</td>
</tr>
<tr>
<td>C</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>100</td>
<td>-</td>
</tr>
</tbody>
</table>

Token size memory execution time

P_1 P_2 P_1 P_2

SDFG-based MP-SoC design-flow

Constraint refinement (2 steps)

Tile binding and scheduling (4 steps)

NoC routing and scheduling (3 steps)

MP-SoC configuration

Memory dimensioning and constraint refinement

Throughput constraint: 0.07 firings / time-unit

<table>
<thead>
<tr>
<th></th>
<th>token size</th>
<th>storage-space</th>
<th>latency</th>
<th>bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>d_1</td>
<td>128</td>
<td>4</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>d_2</td>
<td>64</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

Throughput constraint: 0.07 firings / time-unit

Memory dimensioning and constraint refinement

Streamlining application SDFG

Throughput constraint

Architecture platform

Predictable design flow

MP-SoC configuration

SDFG-based MP-SoC design-flow

Streaming application SDFG

Throughput constraint

Architecture platform

Predictable design flow

MP-SoC configuration

SDFG-based MP-SoC design-flow

Streaming application SDFG

Throughput constraint

Architecture platform

Predictable design flow

MP-SoC configuration
**11. SDFG-based MP-SoC design-flow**

- Streaming application SDFG
- Throughput constraint

**Platform graph**

- Memory dimensioning (4 steps)
- Constraint refinement (2 steps)
- Tile binding and scheduling (4 steps)
- Interconnect graph
- NoC routing and scheduling (3 steps)

**MP-SoC configuration**

---

**18. Tile binding**

- Actors sorted on “criticality”
- Related to notion of cycle-mean in HSDF
- Binding considers
  - Processing load
  - Memory load
  - Communication load
  - Communication latency
- Cost function weights alternative tiles
  \[ \text{cost}(t) = c_1 \cdot l_p(t) + c_2 \cdot l_m(t) + c_3 \cdot l_c(t) + c_4 \cdot l_l(t) \]
- Greedy strategy with one optimization pass after initial binding

---

**19. Scheduling**

- Static-order scheduling between actors
  - Order actor firings of an application on a processor
  - List-scheduling algorithm
- TDMA scheduling between applications
  - Provide timing independence between applications
  - Binary search algorithm using fast throughput analysis technique

---

**20. Tile binding and scheduling**

- Find a binding and scheduling of an SDFG onto an MP-SoC that satisfies the throughput constraint

Find a binding and scheduling of an SDFG onto an MP-SoC that satisfies the throughput constraint
Binding-aware SDFG

Tile binding and scheduling

SDFG-based MP-SoC design-flow
Throughput constraint: 0.07 firings / time-unit

100% time wheels allocated gives throughput of 0.07
NoC routing and scheduling

SDFG-based MP-SoC design-flow

Run-time of the design flow
SDF3: SDF For Free

- SDF3 implements the NoC-based MP-SoC design flow
- Input/output of each step is described in XML
- XML can be transformed to HTML
- Command-line tool and C/C++ API available

Conclusions

- MP-SoC design-flow and SDF3 toolkit available at www.es.ele.tue.nl/sdf3
- First design-flow which maps SDFG to NoC-based MP-SoC
- Considers scheduling on processing, storage and communication resources
- Flow based on trade-offs between storage space, latency and bandwidth
- Most of the steps in the design-flow require milliseconds to complete for realistic applications