An Automated Flow to Map Throughput Constrained Applications to a MPSoC

Roel Jordans, Firew Siyoum, Sander Stuijk, Akash Kumar, Henk Corporaal
Electronic Systems Group, Electrical Engineering, Eindhoven University of Technology

Motivation
Applications are becoming more connected, interactive and dynamic. Moreover, multiple time-constrained applications are executed concurrently on a heterogeneous multi-processor platform.

- Timing guarantees for each application
- Resource utilization
- Requires complex programming techniques

Objective
Develop an automated flow that maps a streaming application onto a template-based MPSoC and produces a working FPGA prototype of the system while minimizing resource usage and providing timing guarantees on the produced system.

Application Modelling
Synchronous Data-Flow (SDF) graphs and C code are used to represent applications throughout the flow.

Listing 1: SDF actor code

```c
void actor_A(TypA & TypB & TypC)
{
    // do something
    // write the output token
}
```

Architecture Modelling
The MAMPS hardware platform is automatically constructed from the architecture model and is based on a set of template components with matching parameterized SDF models.

**SDF For Free (SDF³) and MAMPS**

SDF³ offers
- Dataflow transformation and analysis algorithms
- Advanced MP-SoC binding and scheduling functions for dataflow graphs

MAMPS offers
- Predictable template based MP-SoC
- Fully automated synthesis from graph representation to FPGA platform
- Easy to use API for writing actor code

Case study — MJPEG decoder
This case study was used to validate performance as predicted by SDF³ for different forms of interconnect. The case study also shows the effort required to map an application using the automated flow.

Comparing expected throughput and measured throughput shows minimal margin while throughput at the worst-case analysis line is guaranteed by the flow.

**Timeline**

<table>
<thead>
<tr>
<th>Step</th>
<th>Time spent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paralleling the MJPEG code</td>
<td>3 days</td>
</tr>
<tr>
<td>Creating the SDF graph</td>
<td>5 minutes</td>
</tr>
<tr>
<td>Gathering required actor metrics</td>
<td>1 day</td>
</tr>
<tr>
<td>Creating application model</td>
<td>1 hour</td>
</tr>
<tr>
<td>Mapping the design (SDF³)</td>
<td>1 minute</td>
</tr>
<tr>
<td>Generating Xilinx project (MAMPS)</td>
<td>16 seconds</td>
</tr>
<tr>
<td>Synthesis of the system</td>
<td>17 minutes</td>
</tr>
<tr>
<td>Total time spent</td>
<td>4 days</td>
</tr>
</tbody>
</table>

This work was partly funded by the PROGRESS program of the Dutch Technology Foundation STW through the PreMaDoNa project EES. 6390

R. Jordans
Eindhoven University of Technology
r.jordans@tue.nl

Graph showing throughput comparison.