A Fast Yet Accurate Message-level Communication Bus Model for Timing Prediction of SDFGs on MPSoC

Hai-Dang Vu¹, Sébastien Le Nours¹, Sébastien Pillement¹, Ralf Stemmer² and Kim Grüettner³

¹University of Nantes, France, IETR, UMR CNRS 6164
²University of Oldenburg, Germany
³OFFIS e.V., Germany

ASP-DAC’21, Japan, Jan.18-21, 2021
Biography

Hai-Dang VU (07/09/1993) is a PhD student at the University of Nantes, France. My research interest is on probabilistic modeling and timing properties analysis of multi-processor system-on-chip (MPSoCs).

Project fundings:

- CampusFrance with funds from the French ministry of Europe and Foreign Affairs (MEAE) and by the French ministry for Higher Education, Research and Innovation (MESRI)
- DAAD with funds from the Federal Ministry of Education and Research (BMBF)
Outline

Introduction

Message-level communication model

Evaluation of the proposed approach

Conclusion and future work
Introduction

Message-level communication model

Evaluation of the proposed approach

Conclusion and future work
Motivation
Propose a fast and accurate approach for design space exploration of MPSoCs
Proposed approach

Implementation
- MoC
- MoA
- FPGA

Characterization & Simulation
- Measurement
- Probabilistic Computation
  Time Model
- Message level
  Communication Time Model
- Simulation
  Model

Evaluation
- Observed
  Iterations
- Comparison
- Analysis
  results

Characterization & Simulation
- Probabilistic Computation
  Time Model
- Message level
  Communication Time Model
- Simulation
  Model
Establish and evaluate a fast yet accurate message level communication model

- Scalable for design space exploration
Outline

Introduction

Message-level communication model

Evaluation of the proposed approach

Conclusion and future work
Model of Computation

Channel

Actor

Token rate

GP

GY

GX

ABS

Actor-Execution

ReadTokens | Compute | WriteTokens
Model of Architecture

Tile 0
- Private mem.
- PE

... (repeated for n-1 tiles)

Tile n
- Private mem.
- PE

Shared memory

Shared interconnect
Model of Architecture

- **Actor-execution**:
  - **Tile 0**
    - Private mem.
    - PE
  - **Tile n**
    - Private mem.
    - PE

- **Shared interconnect**
  - No caches
  - 1 token = 1 data word on bus

- **FIFO buffer**
  - Shared memory
  - 1 shared FCFS bus

- **Compute**
  - ReadTokens
  - WriteTokens

- **Shared Memory**

11/27
Measurement infrastructure

![Diagram of measurement infrastructure]

- **GP**
- **GX**
- **GY**
- **ABS**
- **Tile 0**
  - Private mem.
  - PE
- **Tile 1**
  - Private mem.
  - PE
- **Shared memory**
- **Shared interconnect**
- **TMC**
- **TMU**
- **SystemILA**
- **UART**
- **JTAG**

Components and connections are illustrated with arrows and labels, indicating flow and interaction within the measurement infrastructure.
Computation characterization

```
ReadTokens(C1);
start();
ComputeGX(&C1,&C3);
stop();
WriteTokens(C3);
```
Characterize the elementary delays of ReadTokens/WriteTokens

An analytical model formulates the communication time using Timed Petri net (TPN)
Principle of the ML communication model

Transaction level
Message level
Abstraction level

Function called
Ready to write
Function ended

- **initializing**
- **polling**
- **inter-polling**
- **pre-writing**
- **writing**
- **inter-writing**
- **post-writing**
- **managing**

- **waiting**
- **empty event**
- **full event**
- **/ComputeCommTime()**

WriteTokens

- init.
- polling
- pre-writing
- writing
- post-writing
- managing

buffer not available
index != n
index += 1
index == n
Computation of the communication time

- **UpdateStatus()** updates the contention context
- **ComputeCommTime()** computes the communication time
Outline

Introduction

Message-level communication model

Evaluation of the proposed approach

Conclusion and future work
Case studies
7 MicroBlazes share a bus AXI4LITE and a BRAM memory on Xilinx ZC702
Experiments

<table>
<thead>
<tr>
<th>Exp. → Actor ↓</th>
<th>Jpeg1</th>
<th>Jpeg3</th>
<th>Jpeg7</th>
<th>Exp. → Actor ↓</th>
<th>Sobel1</th>
<th>Sobel2</th>
<th>Sobel4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get MCU</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>GP</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>IQ&lt;sub&gt;Y&lt;/sub&gt;</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>GX</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>IQ&lt;sub&gt;Cr&lt;/sub&gt;</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>GY</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>IQ&lt;sub&gt;Cb&lt;/sub&gt;</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>ABS</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>IDCT&lt;sub&gt;Y&lt;/sub&gt;</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IDCT&lt;sub&gt;Cr&lt;/sub&gt;</td>
<td>0</td>
<td>4</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IDCT&lt;sub&gt;Cb&lt;/sub&gt;</td>
<td>0</td>
<td>4</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>YCrCb RGB</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Use injected data for the computation time
- Run 1000000 iterations splitted into 20 processes on dedicated processor E5-2630 v4 (2.20 GHz) (at https://ccipl.univ-nantes.fr)
**Evaluation: Accuracy**

<table>
<thead>
<tr>
<th>Exp.</th>
<th>Measured</th>
<th>TL model</th>
<th>ML model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sobel1</td>
<td>3690</td>
<td>3821.6 (3.57%)</td>
<td>3756.6 (1.80%)</td>
</tr>
<tr>
<td>Sobel2</td>
<td>2902.5</td>
<td>2960.6 (2.00%)</td>
<td>2936.6 (1.17%)</td>
</tr>
<tr>
<td>Sobel4</td>
<td>3097.4</td>
<td>2859.7 (-7.70%)</td>
<td>3153.6 (1.81%)</td>
</tr>
<tr>
<td>Jpeg1</td>
<td>2385860</td>
<td>2387757.6 (0.08%)</td>
<td>2387489.6 (0.07%)</td>
</tr>
<tr>
<td>Jpeg3</td>
<td>940836</td>
<td>940174.7 (-0.07%)</td>
<td>941445.3 (0.06%)</td>
</tr>
<tr>
<td>Jpeg7</td>
<td>941059</td>
<td>925969.7 (-1.61%)</td>
<td>941518.1 (0.05%)</td>
</tr>
</tbody>
</table>

- ML models showed better accuracy than the TL models.
- The ML errors are less than 2% for Sobel filter and 1% for JPEG decoder.
<table>
<thead>
<tr>
<th>Exp.</th>
<th>Measured</th>
<th>TL model</th>
<th>ML model</th>
<th>Speed-up</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sobel1</td>
<td>0:07:23</td>
<td>0:00:08</td>
<td>0:00:03</td>
<td>2.67x</td>
</tr>
<tr>
<td>Sobel2</td>
<td>0:07:03</td>
<td>0:00:11</td>
<td>0:00:03</td>
<td>3.67x</td>
</tr>
<tr>
<td>Sobel4</td>
<td>0:07:13</td>
<td>0:00:13</td>
<td>0:00:02</td>
<td>6.5x</td>
</tr>
<tr>
<td>Jpeg1</td>
<td>13:14:31</td>
<td>0:00:30</td>
<td>0:00:06</td>
<td>5x</td>
</tr>
<tr>
<td>Jpeg3</td>
<td>5:12:58</td>
<td>0:40:09</td>
<td>0:00:04</td>
<td>602.25x</td>
</tr>
<tr>
<td>Jpeg7</td>
<td>5:13:02</td>
<td>1:55:25</td>
<td>0:00:04</td>
<td>1731.25x</td>
</tr>
</tbody>
</table>

- ML models showed faster simulation time up to 1731.25x than the TL models.
- The simulation time for ML models is several seconds.
Introduction

Message-level communication model

 Evaluation of the proposed approach

Conclusion and future work
Conclusion

Advantages

- Fast and accurate communication model
- ML model is composable and can be applied for different mappings, applications

Limits

- Simple FCFS bus arbitration policy AXI4LITE.
Future work

- Consider private and shared caches.
- Consider COTS MPSoC platforms.
- Use probabilistic computation time model.
- Consider a more complex bus (DMA transfers and split-burst transactions).
- Apply statistical model checking analysis approach.
Thank you for your attention!
Questions?