D 3.3 Hardware implementation of the self-timed NOC

Version 1.0
31 May 2013
Final

Public Distribution

Technical University of Denmark

Project Partners: AbsInt Angewandte Informatik, Eindhoven University of Technology, GMVIS
Skysoft, Intecs, Technical University of Denmark, The Open Group, University of York, Vienna University of Technology

Every effort has been made to ensure that all statements and information contained herein are accurate, however the Partners accept no liability for any error or omission in the same.

© 2013 Copyright in this document remains vested in the T-CREST Project Partners.
## Project Partner Contact Information

<table>
<thead>
<tr>
<th>Company</th>
<th>Contact Information</th>
</tr>
</thead>
</table>
| **AbsInt Angewandte Informatik** | Christian Ferdinand  
Science Park 1  
66123 Saarbrücken, Germany  
Tel: +49 681 383600  
Fax: +49 681 3836020  
E-mail: ferdinand@absint.com |
| **Eindhoven University of Technology** | Kees Goossens  
Potentiaal PT 9.34  
Den Dolech 2  
5612 AZ Eindhoven, The Netherlands  
E-mail: k.g.w.goossens@tue.nl |
| **GMVIS Skysoft** | João Baptista  
Av. D. Joao II, Torre Fernao Magalhaes, 7  
1998-025 Lisbon, Portugal  
Tel: +351 21 382 9366  
E-mail: joao.baptista@gmv.com |
| **Intecs** | Silvia Mazzini  
Via Fori trav. A5 Ospedaletto  
56121 Pisa, Italy  
Tel: +39 050 965 7513  
E-mail: silvia.mazzini@intecs.it |
| **Technical University of Denmark** | Martin Schoeberl  
Richard Petersens Plads  
2800 Lyngby, Denmark  
Tel: +45 45 25 37 43  
Fax: +45 45 93 00 74  
E-mail: masca@imm.dtu.dk |
| **The Open Group** | Scott Hansen  
Avenue du Parc de Woluwe 56  
1160 Brussels, Belgium  
Tel: +32 2 675 1136  
Fax: +32 2 675 7721  
E-mail: s.hansen@opengroup.org |
| **University of York** | Neil Audsley  
Deramore Lane  
York YO10 5GH, United Kingdom  
Tel: +44 1904 325 500  
E-mail: Neil.Audsley@cs.york.ac.uk |
| **Vienna University of Technology** | Peter Puschner  
Treitlstrasse 3  
1040 Vienna, Austria  
Tel: +43 1 58801 18227  
Fax: +43 1 58801 918227  
E-mail: peter@vmars.tuwien.ac.at |
## Document Control

<table>
<thead>
<tr>
<th>Version</th>
<th>Status</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>First draft</td>
<td>24 May 2013</td>
</tr>
<tr>
<td>1.0</td>
<td>Final draft</td>
<td>31 May 2013</td>
</tr>
</tbody>
</table>
1 Introduction

Deliverable D3.3 is the hardware implementation of the self-timed NOC. It involves a gate netlist implementation of the NIs and asynchronous network of routers. The NOC follows the simulation model as it was presented in deliverable D3.2, now using a self-timed router design. The whole NOC has been synthesized in a 65nm STMicroelectronics technology library. The purpose of this document is to provide a brief overview of the design and the related implementation and simulation files. A report providing detailed description will be delivered later as D3.4.

2 Gate Netlist Implementation

For the self-timed network, a new asynchronous router design was used. The router design is based on the synchronous 3-stage pipeline router that was presented in deliverable D3.2 report [2]. A 2-phase bundled data protocol is used for synchronization as presented in [3]. To synchronize neighboring stages MOUSERTRAP controllers[4] are used, extended with the "clock" gating scheme shown in [3] to save power consumption.

The implementation of the NI is based on the VHDL simulation model that was presented in Deliverable 3.2 [2] and published in [5]. This design was extended to accommodate a phase delay of either 0, 1 or 2 cycles in the incoming and outgoing packets from and to the network. The reason for this is that phit scheduling is followed in this design. Phit scheduling is needed since packets are organized in sizes of 3 flits, while routers cannot store one whole packet(3 phits). Thus, a fine-grained scheduling is needed, at the level of phits, resulting in packets scheduled with possible gaps of 0, 1 or 2 cycles inbetween. To integrate this functionality, an additional 2-bit field was added to the Slot Table in the NI to encode the potential phase delay and control functionality to delay the outgoing packet based on this phase delay. Control functionality was added to detect phase delay in the incoming packets as well.

The implemented netlist was derived through synthesis of the parameterized VHDL model of NIs and asynchronous routers. The synthesized netlist implements a 2x2 bi-torus network. In total 4 NIs and 4 asynchronous MOUSERTRAP routers were synthesized in a 65nm STMicroelectronics technology library.

The NOC was verified using post-synthesis simulation. To obtain schedules on the phit level a dedicated phit scheduler was used and can be found in [1]. The initialization part of the simulation environment was modified to use the new phit-level scheduling. The test environment is initializing the SPM data, the Slot Table with the schedule provided and configuring the DMA with the communication channels. The SPM memories are simulated in the environment.

This deliverable includes: (i) source code describing the noc and the environment behaviour, (ii) a gate netlist implementation, (iii) the synthesis scripts that were used for the implementation and (iv) simulation scripts for testing. The source code involves the VHDL description of the NI, the asynchronous routers, the 2x2 bi-torus network and the simulated environment (simulated processor interface, SPM memories). The synthesized gate netlist(.v) is accompanied by the corresponding timing constraints(.sdc) and cell delay annotation(.sdf) files. For synthesizing the design “tcl” scripts for synthesis are included in the deliverable. They define the synthesis parameters to optimize the
implemented netlist for performance and area. Simulation scripts for the simulation model as well as for the synthesized netlist are provided along with schedule files with one-to-all and all-to-all communication.

3 Accessing the Source Code

The noc code and scripts are provided through the t-crest repository via the git source code management tool:

https://github.com/t-crest/t-crest-noc

The self-timed noc source code along with a simulation environment, the gate netlist and synthesis scripts can be found under the directory /async_noc.

3.1 Requirements

To compile and simulate of the t-crest NOC, the VHDL design as well as the gate netlist, the following tools are needed:

- A Unix like environment with git, make, and a C/C++ compiler, such as: Linux, Mac OS X, or cygwin/Windows
- A recent version of cmake
- ModelSim for simulation (the free version from Altera is good enough)

To synthesize the gate netlist of the t-crest NOC the following tools are needed:

- A Unix like environment with git, and a C/C++ compiler, such as: Linux, Mac OS X, or cygwin/Windows.
- DesignCompiler from Synopsys.
- STMicroelectronics Technology Libraries.

3.2 Retrieving and Running the Source Code

The VHDL description model of the T-CREST NOC can be retrieved as follows:

`git clone git://github.com/t-crest/noc.git`

or downloaded as .zip file from GitHub:

https://github.com/t-crest/t-crest-noc/zipball/master

The simulation of the t-crest NOC is make based.

A plain, and simple

https://www.altera.com/download/software/modelsim-starter
make

will: (1) compile the VHDL design or the gate netlist, (2) simulate the model/netlist along with the environment, (3) generate a waveform with the signals showing the operation.

Following make targets are available to simulate the different test-cases as presented in Deliverable 3.2 [2] for the VHDL description model as well as for the gate netlist and clean up from temporary files:

make test0  simulate test-case0
make test1  simulate test-case1
make test2  simulate test-case2
make netlist0 simulate test-case0 for the gate netlist
make netlist1 simulate test-case1 for the gate netlist
make netlist2 simulate test-case2 for the gate netlist
make clean  remove (most) temporary files

The Makefile is intended to support: Linux, a cygwin environment under Windows, and Mac OSX. Under Mac OSX the Windows version of ModelSim is supported via wine.
References


