T-CREST

TIME-PREDICTABLE MULTI-CORE ARCHITECTURE
FOR EMBEDDED SYSTEMS

Project Number 288008

D 8.5 Open Source Reference Implementation

Version 1.0
25 September 2014
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**

Project Partner Contact Information

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 José Neves Av. D. João II, Torre Fernão de Magalhães, 7 1998-025 Lisbon, Portugal Tel: +351 21 382 9366 E-mail: jose.neves@gmv.com	**Intecs** Silvia Mazzini Via Forti 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

Contents

Document Control

Version	Status	Date
0.1	First draft	1 September 2014
0.9	Version for partner review	24 September 2014
1.0	Final version	25 September 2014

Executive Summary

This document describes the deliverable *D 8.5 Open Source Reference Implementation* of work package 8 of the T-CREST project, due 36 months after project start as stated in the Description of Work. This document describes the open-source T-CREST components and where they can be found.

1 Introduction

Most components of T-CREST are made available in open source. The components are licensed under the industry friendly "Simplified BSD License". The license text is as follows:

The T-CREST compiler is based on LLVM and therefore licensed under the University of Illinois/NCSA Open Source License.[1]

2 Source Access

T-CREST components are hosted on GitHub under the Organization `t-crest`:

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

[1] `http://en.wikipedia.org/wiki/University_of_Illinois/NCSA_Open_Source_License`

In the following we list the individual components in two categories: core components and secondary components. The core components are direct results of the T-CREST project. Secondary components are additional sub-projects not directly connected to the EC funded project T-CREST (e.g., master thesis work or work funded by another research project (RTEMP)).

2.1 Core Components

Core components are part of the EC funded project T-CREST.

2.1.1 Patmos

Patmos [12] is the processor of the T-CREST platform:

`https://github.com/t-crest/patmos`

The `patmos` repository contains the hardware description of the processor, a software simulator, and a small assembler. Furthermore, the repository contains two memory controllers for SRAM and SSRAM memories as they are available on the Altera DE2 boards. Several cores can share that memory controller via a round-robin memory arbiter or a distributed TDM based memory arbiter [11].

2.1.2 Compiler, Linker, and Library

The T-CREST compiler [8], linker, and libraries are distributed over several repositories.

Port of the LLVM compiler infrastructure to the time-predictable processor Patmos:

`https://github.com/t-crest/patmos-llvm`

Port of the clang front-end to the time-predictable processor Patmos:

`https://github.com/t-crest/patmos-clang`

Port of LLVM's compiler library:

`https://github.com/t-crest/patmos-compiler-rt`

Port of the binutils gold linker to Patmos:

`https://github.com/t-crest/patmos-gold`

Newlib port for Patmos:

`https://github.com/t-crest/patmos-newlib`

2.1.3 RTEMS

Port of the RTEMS real-time operating system to the time-predictable processor Patmos.

`https://github.com/t-crest/rtems`

2.1.4 Argo

Argo is a time-division multiplexing network-on-chip [2] with synchronous and asynchronous routers [4, 3] and an area-efficient network interface [15].

t-crest:tvlsi2014

`https://github.com/t-crest/argo`

2.1.5 Memory Tree

The memory tree is the memory interconnect designed for the T-CREST project such that multiple Patmos processors have a time-predictable way to access memory [7, 1].

`https://github.com/t-crest/bluetree`

2.1.6 S4NOC

A statically-scheduled TDM network-on-chip for real-time systems. The initial version of the T-CREST NoC [10] including a minimalistic network interface [13].

`https://github.com/t-crest/s4noc`

2.1.7 Benchmarks

A collection of embedded and WCET benchmarks and tests for the Patmos processor and compiler.

`https://github.com/t-crest/patmos-benchmarks`

2.1.8 Misc

Config files, scripts, and other stuff for Patmos.

`https://github.com/t-crest/patmos-misc`

2.1.9 Aegean

Configuration framework for the T-CREST platform (Aegean).

`https://github.com/t-crest/aegean`

2.1.10 SDRAM controller model

RTMemController is an open-source tool for evaluating the worst-case and average-case execution time of memory transactions of the dynamic memory controller, described in [6].

`http://www.es.ele.tue.nl/rtmemcontroller/`

2.2 Secondary Components

Secondary components are additional contributions, often results of a master thesis, to T-CREST.

2.2.1 SDRAM controller

A simple and static SDRAM controller for time-predictable main memory access [5].

`https://github.com/t-crest/sdram`

2.2.2 ospat

A real-time operating system for Patmos [16].

`https://github.com/t-crest/ospat`

2.2.3 Poseidon

Static NoC TDM scheduler [14].

`https://github.com/t-crest/poseidon`

2.2.4 Debugger

Port of the LLDB debugger for Patmos.

`https://github.com/t-crest/patmos-lldb`

2.2.5 OTAWA support

Files for OTAWA Patmos port.

`https://github.com/t-crest/patmos-otawa`

2.2.6 RTEMP GUI

Graphical Configuration for the RTEMP Project.

`https://github.com/t-crest/rtemp-gui`

3 Retrieving the Source Code and Build Instructions

Here we list very briefly the instructions to build the T-CREST platform version consisting of open-source components. Detailed instructions can be found in the Patmos reference handbook [9].

To build T-CREST and develop for T-CREST only open-source or free tools are needed (except an FPGA board and a PC):

- A Unix like environment with `git`, `make`, and a C/C++ compiler, such as: Linux, Mac OSX, or cygwin/Windows
- A recent version of `cmake` and `boost` for the assembler
- ModelSim for simulation (the free version from Altera is good enough) [2]
- Altera Quartus for synthesizing for an FPGA (the free Web edition is good enough) [3]
- An FPGA board, the default target is currently the Terasic DE2-115 board [4]
- A serial interface on a PC to read the program output

We assume that the T-CREST project will live in `$HOME/t-crest`. The source code of T-CREST can be retrieved with as follows:

```
mkdir ~/t-crest
cd ~/t-crest
git clone https://github.com/t-crest/patmos-misc.git misc
./misc/build.sh
```

The `build.sh` script will clone all other needed repositories and then builds the compiler, the Patmos simulator, and the Patmos hardware emulator.

To generate a 2x2 bi-torus T-CREST platform, synthesize it, and configure the FPGA execute following commands:

[2] `https://www.altera.com/download/software/modelsim-starter`
[3] `https://www.altera.com/download/software/quartus-ii-we`
[4] `https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=139&No=502`

```
cd aegean
make platform
make synth
make config
```

The default board for the open-source version of T-CREST is the Altera DE2-115 FPGA board. The target board can be changed by setting the `BOARD` variable in the above make commands.

References

[1] Jamie Garside and Neil C Audsley. WCET Preserving Hardware Prefetch for Many-Core Real-Time Systems. In *RTNS 2014*, page To Appear, Versailles, 2014.

[2] Evangelia Kasapaki, Martin Schoeberl, Rasmus Bo Sørensen, Christian T. Müller, Kees Goossens, and Jens Sparsø. Argo: A Real-Time Network-on-Chip Architecture with an Efficient GALS Implementation. 2014. submitted.

[3] Evangelia Kasapaki and Jens Sparsø. Argo: A Time-Elastic Time-Division-Multiplexed NOC Using Asynchronous Routers. In *Proc. IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*, pages 45–52, May 2014.

[4] Evangelia Kasapaki, Jens Sparsø, Rasmus Bo Sørensen, and Kees Goossens. Router designs for an asynchronous time-division-multiplexed network-on-chip. In *Digital System Design (DSD), 2013 Euromicro Conference on*, pages 319–326. IEEE, 2013.

[5] Edgar Lakis and Martin Schoeberl. An SDRAM controller for real-time systems. In *Proceedings of the 9th Workshop on Software Technologies for Embedded and Ubiquitous Systems*, 2013.

[6] Yonghui Li, Benny Akesson, and Kees Goossens. Dynamic command scheduling for real-time memory controllers. In *Proceedings of the 13th Euromicro Conference on Real-Time Systems (ECRTS)*, July 2014.

[7] Gary Plumbridge, Jack Whitham, and Neil Audsley. Blueshell : A Platform for Rapid Prototyping of Multiprocessor NoCs and Accelerators. In *Proceedings HEART Workshop*. University of York, 2013.

[8] Peter Puschner, Daniel Prokesch, Benedikt Huber, Jens Knoop, Stefan Hepp, and Gernot Gebhard. The T-CREST approach of compiler and WCET-analysis integration. In *9th Workshop on Software Technologies for Future Embedded and Ubiquitious Systems (SEUS 2013)*, pages 33–40, 2013.

[9] Martin Schoeberl, Florian Brandner, Stefan Hepp, Wolfgang Puffitsch, and Daniel Prokesch. Patmos reference handbook. Technical report, 2014.

[10] Martin Schoeberl, Florian Brandner, Jens Sparsø, and Evangelia Kasapaki. A statically scheduled time-division-multiplexed network-on-chip for real-time systems. In *Proceedings of the 6th International Symposium on Networks-on-Chip (NOCS)*, pages 152–160, Lyngby, Denmark, May 2012. IEEE.

[11] Martin Schoeberl, David VH Chong, Wolfgang Puffitsch, and Jens Sparsø. A time-predictable memory network-on-chip. In *Proceedings of the 14th International Workshop on Worst-Case Execution Time Analysis (WCET 2014)*, 2014.

[12] Martin Schoeberl, Pascal Schleuniger, Wolfgang Puffitsch, Florian Brandner, Christian W. Probst, Sven Karlsson, and Tommy Thorn. Towards a time-predictable dual-issue microprocessor: The Patmos approach. In *First Workshop on Bringing Theory to Practice: Predictability and Performance in Embedded Systems (PPES 2011)*, pages 11–20, Grenoble, France, March 2011.

[13] Rasmus Bo Sørensen, Martin Schoeberl, and Jens Sparsø. A light-weight statically scheduled network-on-chip. In *Proceedings of the 29th Norchip Conference*, Copenhagen, November 2012. IEEE.

[14] Rasmus Bo Sørensen, Jens Sparsø, Mark Ruvald Pedersen, and Jaspur Højgaard. A metaheuristic scheduler for time division multiplexed network-on-chip. In *Software Technologies for Future Embedded and Ubiquitous Systems (SEUS), 2014*. IEEE, 2014.

[15] Jens Sparsø, Evangelia Kasapaki, and Martin Schoeberl. An area-efficient network interface for a TDM-based network-on-chip. In *Proceedings of the Conference on Design, Automation and Test in Europe*, DATE '13, pages 1044–1047, San Jose, CA, USA, 2013. EDA Consortium.

[16] Marco Ziccardi. A time-composable operating system for the Patmos processor. Master's thesis, Technical University of Denmark, 2013.