


A Typing Discipline for Hardware Interfaces (Artifact)

Jan de Muijnck-Hughes 

University of Glasgow, UK
Jan.deMuijnck-Hughes@glasgow.ac.uk

Wim Vanderbauwhede 

University of Glasgow, UK
Wim.Vanderbauwhede@glasgow.ac.uk

Abstract

Modern Systems-on-a-Chip (SoC) are constructed by composition of IP (Intellectual Property) Cores with the communication between these IP Cores being governed by well described interaction protocols. However, there is a disconnect between the machine readable specification of these protocols and the verification of their implementation in known hardware description languages. Although tools can be written to address such a separation

of concerns, such tooling is often hand written and used to check hardware designs *a posteriori*. We have developed a dependent type-system and proof-of-concept modelling language to reason about the physical structure of hardware interfaces respective to user provided descriptions. Our type-system provides correct-by-construction guarantees that the interfaces on an IP Core will be well-typed if they adhere to a specified standard.

2012 ACM Subject Classification Theory of computation → Type theory; Hardware → System on a chip; Software and its engineering → System description languages

Keywords and phrases System-on-a-Chip, AXI, Dependent Types, Substructural Typing

Digital Object Identifier 10.4230/DARTS.5.2.14

Funding This work is part of *Border Patrol: Improving Smart Device Security through Type-Aware Systems Design* – EPSRC Project EP/N028201/1.

Related Article Jan de Muijnck-Hughes and Wim Vanderbauwhede, “A Typing Discipline for Hardware Interfaces”, in 33rd European Conference on Object-Oriented Programming (ECOOP 2019), LIPIcs, Vol. 134, pp. 6:1–6:27, 2019.

<https://dx.doi.org/10.4230/LIPIcs.ECOOP.2019.6>

Related Conference 33rd European Conference on Object-Oriented Programming (ECOOP 2019), July 15–19, 2019, London, United Kingdom

1 Scope

CORDIAL is a theoretical framework that provides guarantees that the interfaces on an IP (Intellectual Property) core satisfies a known description. CORDIAL comprises of a system of related statically typed models that reason about the structure of interfaces on a component respective to a given interface specification. We build upon existing work from existing hardware description tooling, and utilise state-of-the-art concepts from programming language theory to provide correct-by-construction guarantees over construction of abstract interface descriptions and the adherence of concrete interfaces to said abstract descriptions. The framework has been realised within the dependently typed language Idris [3] as a proof-of-concept implementation, and it has been used to describe several well known IP communication protocols. Namely: APB from ARM [1]; LocalLink [4]; and AXI4 from ARM [2].



© J. de Muijnck-Hughes and W. Vanderbauwhede;
licensed under Creative Commons Attribution 3.0 Germany (CC BY 3.0 DE)

Dagstuhl Artifacts Series, Vol. 5, Issue 2, Artifact No. 14, pp. 14:1–14:3



DAGSTUHL
ARTIFACTS SERIES

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

14:2 A Typing Discipline for Hardware Interfaces (Artifact)

Our artefact is the Idris implementation of CORDIAL together with the paper's described case studies. We present the artefact within a Virtual Box¹ machine image that contains a working production environment. The machine image has been designed to work with virtual machine management tool Vagrant². We have also provided the source code for CORDIAL alongside the source code for inspection and construction outside the constraints imposed by the virtual machine setup.

2 Content

The artifact itself includes:

- `README.org`
A listing of the archive's contents.
- `ecoop2019.box`
A vagrant box containing the virtual machine that replicates an execution environment for the framework.
- `About-The-Artefact.pdf`
A file containing instructions for accessing the artefacts, including software dependencies, together with more information detailing the artefacts relationship to the paper.
- `code/`
The source files for the framework and dependencies have been provided alongside the virtual machine to enable better viewing of the source. There is a `Makefile` that will install the dependencies, build the framework, and type-check the examples. The listing for `code` is:
 - `cordial/`
This directory contains the source code for CORDIAL. It is comprised of two coalesced Idris packages, one containing the model and the other containing the examples themselves. The following sub-directories are of interest:
 - * `src/`
Contains the source code for the framework and the examples.
 - * `src_html/`
A pretty-printed version of the source rendered as HTML files. The HTML highlighted source code was generated using an external program³.
 - * `example_doc/` & `cordial_doc`
Contains generated `idrisdoc`⁴ documentation.
 - `test/`
A dependency for internal testing.
 - `commons/`
A dependency containing required data structures.
 - `containers/`
A dependency containing required data structures.
 - `edda/`
A dependency used to add rich documentation to specifications.
 - `xml/`
A dependency that is required for `edda`, and will be used in future releases of `cordial`.

¹ <https://www.virtualbox.org/>

² <https://www.vagrantup.com>

³ <https://github.com/david-christiansen/idris-code-highlighter>

⁴ <http://docs.idris-lang.org/en/v1.3.1/reference/documenting.html>

3 Getting the artifact

The artifact endorsed by the Artifact Evaluation Committee is available free of charge on the Dagstuhl Research Online Publication Server (DROPS). In addition, the latest version of CORDIAL has been made available at:

<https://github.com/border-patrol/cordial>.

4 Tested platforms

The implementation of CORDIAL was developed and tested on Ubuntu Mate LTS 18.04.2 with version 1.3.1 of Idris. We have packaged the artefact within a Virtual Box machine image managed by Vagrant to provide a reproducible production environment. Virtual Box and Vagrant are known to work across all the major platforms: Linux, macOS, and Windows. As an alternative, we have also provided the supporting source code. Idris is also known to work with all the major platforms. More information about the machine image, and working with the provided source code, is presented alongside the virtual machine image. We have not tested the framework on other platforms.

5 License

The artifact is available under *The Clear BSD License*.

6 MD5 sum of the artifact

0636fdbc64a9f3308c05e32b1a56eb2e

7 Size of the artifact

1.4 GiB

References

- 1 ARM Limited. *AMBA APB Protocol*, arm ihi 0024c edition, 2010.
- 2 ARM Limited. *AXI and ACE Protocol Specification*, arm ihi 0022f.b edition, 2017.
- 3 Edwin Brady. Idris, a general-purpose dependently typed programming language: Design and implementation. *J. Funct. Program.*, 23(5):552–593, 2013. doi:10.1017/S095679681300018X.
- 4 Xilinx. *LocalLink Interface Specification*, sp006 (v2.0) edition, 2005.