OpenCPI Verification White Paper

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **By** | **Notes** |
| 0.00 | 2010-11-01 | Shepard Siegel | Creation |
| 0.01 | 2011-02-15 | Shepard Siegel | Update |

# Introduction and Overview

This document explains how industry-standard verification and test practices are used specifically in the context of OpenCPI.

# Application Worker Verification Methodology

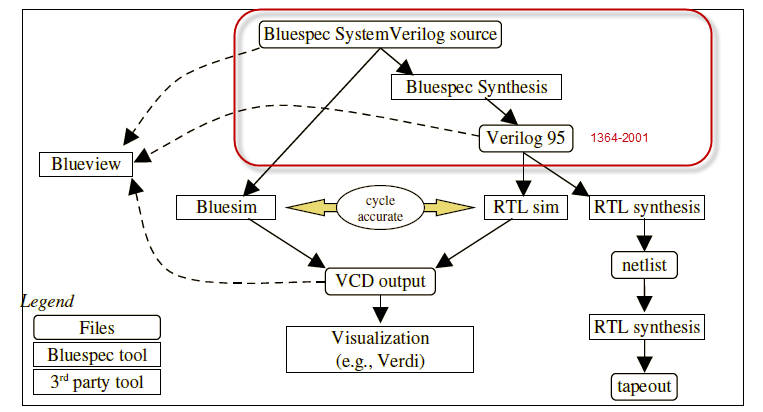
This section details the test cases provided that are used to verify an application worker’s function and performance. In this context, the worker is sometimes referred to as the Device-Under-Test (DUT).

A universe of methodologies exists for verification of RTL Application Workers. This section of the verification user doc describes one possible approach among many. It is specifically our intent not to dictate a particular methodology; but to document and demonstrate our process.

The practice described herein is based in concepts in the text “*Writing Testbenches using SystemVerilog*” by Janick Bergeron (2006, Springer). This text followed two editions of an earlier text called “*Writing Testbenches, Functional Verification of HDL Models*”, by Janick Bergeron (2000, 2003, KAP). Where the first two editions provided examples with implementation languages of VHDL and Verilog; the latest edition uses SystemVerilog.

With OpenCPI, we have tried to be as agnostic to implementation language as possible; and instead focus on the interaction patterns that are agnostic to the language. One common denominator between languages we use is IEEE Verilog 1364-2001. Because of its ubiquity, we refer to this just as “RTL”.

The diagram below, taken from the document $OCPI/doc/bsv-verification-guide.pdf, shows how still another implementation language, Bluespec SystemVerilog (BSV) is compiled to RTL.



Technologies such “C to Gates”, “MATLAB to Gates” and “OpenCL to Gates” will mature, and almost certainly have a path for their compiler output to VHDL or Verilog RTL.

## Specific Test Cases

The table below shows at a glance the attributes of specific test cases. The source code to a particular test cases is given as TB<x> and can be found in $OCPI/bsv/tst/TB<x>.bsv .

|  |  |  |
| --- | --- | --- |
| **ID** | **Worker** | **Notes** |
| 1 | BiasWorker | Procedural source/sink in TB1; with Protocol Monitors in TB11 |
| 2 | DelayWorker | Procedural source/sink in TB9 |
| 3 | FFTWorker | Array source, Procedural Sink in TB10 |

# Reference Documents

Table - list of Reference Documents

|  |  |  |
| --- | --- | --- |
| **ID** | **Standard** | **Link** |
| 1 | OCP 3.0 Specification | [http://www.ocpip.org](http://www.ocpip.org/) |
| 2 | OpenCPI WIP Specification | [http://www.opencpi.org](http://www.opencpi.org/) |
| 3 | ARM AMBA 4.0 AXI-4 Protocol Specification |  |
| 4 | IEEE 1364-2001 – “Verilog 2001” |  |

Table - List of Abbreviations

|  |  |
| --- | --- |
| AFRL | Air Force Research Lab |
| ALT | Altera |
| AR | Atomic Rules |
| AXI | Advanced Extendable Interconnect |
| BSV | Bluespec SystemVerilog |
| FFT | Fast Fourier Transform |
| FPGA | Field Programmable Gate Array |
| GIT | A distributed version control system |
| HDL | Hardware Description Language (VHDL or Verilog) |
| IP | Intellectual Property (core, worker) |
| MFS | Mercury Federal Systems |
| OCP | Open Core Protocol |
| OCDP | OpenCPI Data Plane |
| OCPI | Open Component Portability Infrastructure |
| POP | Period of Performance |
| PCI | Peripheral Component Interconnect |
| PCIe | PCI Express |
| SOW | Statement of Work |
| TRL | Technology Readiness Level |
| WCI | Worker Control Interface |
| WIP | Worker Interface Profiles |
| WSI | Worker Streaming Interface |
| XIL | Xilinx |