

fixed_extensions user manual

Title	fixed_extensions (VHDL fixed-point arithmetic extensions package)
Author	Nikolaos Kavvadias 2011, 2012, 2013, 2014
Contact	nikos@nkavvadias.com
Website	http://www.nkavvadias.com
Release Date	26 September 2014
Version	0.1.1
Rev. history	
v0.1.1	26-09-2014 Updated README and file organization for Github. Re-named COPYING.BSD to LICENSE.
v0.1.0	21-02-2014 Changed documentation format to RestructuredText.
v0.0.5	25-07-2011 First public release.

1. Introduction

`fixed_extensions_pkg` is a fixed-point arithmetic package written in VHDL according to the VHDL-2008 update of the standard. It uses VHDL-2008 back-compatible libraries (by David Bishop) that are included in this distribution for the sake of completeness.

Currently, the `fixed_extensions_pkg` package implements the following:

- ceil:** round towards plus infinity.
- fix:** round towards zero.
- floor:** round towards minus infinity.
- round:** round to nearest; ties to greatest absolute value.
- nearest:** round to nearest; ties to plus infinity.
- convergent:** round to nearest; ties to closest even.
- bitinsert:** bit-field insertion to word
- bitextract:** bit-field extraction from word

`fixed_extensions` is distributed along with a tool (`gentestround`) to generate customized VHDL test designs.

The `fixed_extensions` project can be downloaded either from the following OpenCores website: http://opencores.org/project,fixed_extensions or from its corresponding Github repository: http://github.com/nkkav/fixed_extensions

2. File listing

The `fixed_extensions` distribution includes the following files:

<code>/fixed_extensions</code>	Top-level directory
<code>AUTHORS</code>	List of authors.
<code>BUGS</code>	Bug list.
<code>ChangeLog</code>	A log for code changes.
<code>COPYING.BSD</code>	The modified BSD license.
<code>README</code>	This file.
<code>README.html</code>	HTML version of README.
<code>README.pdf</code>	PDF version of README.
<code>THANKS</code>	Acknowledgements.
<code>TODO</code>	A list of future enhancements.
<code>VERSION</code>	Current version of the project sources.
<code>rst2docs.sh</code>	Bash script for generating the HTML and PDF versions.
<code>/bench/vhdl</code>	Benchmarks VHDL directory
<code>testrounding_tb.vhd</code>	Standard testbench file.
<code>/gen/vhdl</code>	Generated RTL VHDL code directory.
<code>testroundings.vhd</code>	Auto-generated test file for sfixed arithmetic.
<code>testroundingu.vhd</code>	Auto-generated test file for ufixed arithmetic.
<code>/rtl/vhdl</code>	RTL source code directory for the package
<code>fixed_extensions_pkg-_sim.vhd</code>	The VHDL package for simulation-oriented use.
<code>/sim/rtl_sim</code>	RTL simulation files directory
<code>/sim/rtl_sim/bin</code>	RTL simulation scripts directory
<code>run.sh</code>	A bash script for testing the package.
<code>testroundings.do</code>	Modelsim macro script for testing sfixed arithmetic.
<code>testroundings.sh</code>	Bash script for running an sfixed simulation.
<code>testroundingu.do</code>	Modelsim macro script for testing ufixed arithmetic.
<code>testroundingu.sh</code>	Bash script for running an ufixed simulation.
<code>/sim/rtl_sim/src</code>	Various source files for running RTL simulations
<code>fixed_float_types_custom.vhd</code>	VHDL package with definitions for fixed-point arithmetic.
<code>fixed_pkg_c.vhd</code>	VHDL package implementing fixed-point arithmetic (VHDL'93 version of the VHDL-2008 package as found http://www.eda.org/fphdl/).

math_real.vhd	VHDL package with some real arithmetic functions (also part of the IEEE 1076 standard for VHDL).
/sw	Software utilities
Makefile	Makefile for compiling the test design generator.
gentestround.c	Test design generator written in ANSI C.

3. fixed_extensions usage

The fixed_extensions package can be used as follows. Assuming that the user has changed directory to ./fixed_extensions, the following can be used:

```
$ cd sim/rtl_sim/bin
$ ./run.sh
```

Alternatively, the user can only generate and run some tests for solely the signed fixed-point and unsigned fixed-point data types. This is correspondingly performed as:

```
$ ./testroundings.sh
```

and

```
$ ./testroundingu.sh
```

4. Prerequisites

- Standard UNIX-based tools (tested on cygwin/x86)
 - make
 - bash
- [optional] Mentor Modelsim (mt i) from <http://www.model.com>
Provides a simulation environment to run the tests.