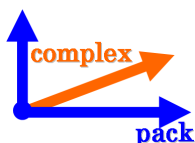


complexpack user manual



Title	complexpack (VHDL complex arithmetic package).
Author	Nikolaos Kavvadias 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017
Contact	nikolaos.kavvadias@gmail.com
Website	http://www.nkavvadias.com
Release Date	16 June 2017
Version	0.2.3
Rev. history	
v0.2.3	2017-06-16 Add complex reciprocal.
v0.2.2	2017-06-10 Add complex mac.
v0.2.1	2017-06-07 Add arg (argument).
v0.2.0	2017-06-05 Add polar form, to_polar, to_cartesian, exp, log, pow, sqrt, sin, cos, tan, arcsin, arccos, arctan.
v0.1.5	2016-23-07 Use sim/rtl_sim/bin dir, ignores for git, clean script
v0.1.4	2016-03-13 Update date information.
v0.1.3	2014-11-30 Added project logo to README.rst.
v0.1.2	2014-06-17 Changed README to README.rst.
v0.1.1	2014-03-04 Added support for the "abs" and the negation ("-") operators as those are needed by the complexarrpack package project.
v0.1.0	2014-03-03 Added get_real, get_imaginary, magnitude, and operators: lt, gt, le, ge, eq, ne. New library dependency: IEEE.math_real.

v0.0.1	2014-02-24 Changed documentation format to RestructuredText. Code has been reorganized into new directory structure.
v0.0.0	2009-10-02 First public release.

1. Introduction

`complexpack` is a simple complex arithmetic package written in VHDL. It is based on a code example present in the RASSP series of VHDL lectures. Compared to the RASSP version, the following have been added:

- conjugate function.
- magnitude function.
- comparison operators.
- get real and get imaginary part functions.

A complex number is defined by the pair (real-part, imaginary-part) where both items of the pair are numbers. A common algebraic representation for complex numbers is:

$$z = a + i*b,$$

where:

- z is the resulting complex number
- a is the real part of the number also written as $a = \text{Re}(z)$
- b is the imaginary part of the number also written as $b = \text{Im}(z)$
- i is the imaginary unit and has the value of $\sqrt{-1}$.

Currently, the `complexpack` package implements the following:

- the constants `re` and `im`, which specify addresses for an array-based representation of a complex number
- type definition for a complex number
- interface and implementation for complex arithmetic functionality

1.1. Implemented functions and operators

This is a summary of the currently supported functions, procedures and operators by the `complexpack` package.

- `to_complex(real, real)`: form a complex number
- `+`: add two complex numbers

- `-`: subtract one complex number from another
- `-`: negate a complex number
- `*`: multiply two complex numbers
- `/`: divide two complex numbers
- `mac`: multiply-and accumulate (`a * b + c`)
- `reciprocal`: the reciprocal of a complex number
- `conjugate (complex)`: return the conjugate of the given complex number
- `exp (complex)`: return the complex exponent
- `log (complex)`: return the complex logarithm
- `pow (complex, complex)`: return the complex power
- `sqrt (complex)`: return the complex square root
- `sin (complex)`: return the complex sine
- `cos (complex)`: return the complex cosine
- `tan (complex)`: return the complex tangent
- `arcsin (complex)`: return the complex arcsine
- `arccos (complex)`: return the complex arccosine
- `arctan (complex)`: return the complex arctangent
- `to_cartesian (polar)`: convert from polar form to Cartesian

All functions above return an item of the complex data type.

- `get_real (complex)`: get the real part of a complex number
- `get_imaginary (complex)`: get the imaginary part of a complex number
- `magnitude (complex)`: return the magnitude (distance from point 0,0) of the complex number
- `arg (complex)`: return the argument (phase) of the complex number
- `abs (complex)`: alias for magnitude

All functions above return an item of the real data type (a scalar quantity).

- `<`: less than comparison for two complex numbers
- `>`: greater than comparison for two complex numbers
- `<=`: less than or equal comparison for two complex numbers
- `>=`: greater than or equal comparison for two complex numbers
- `=`: equality comparison for two complex numbers

- `/=`: non-equality comparison for two complex numbers

All functions above return an item of the boolean data type (TRUE or FALSE).

- `to_polar(complex)`: convert the complex (Cartesian) to polar coordinates

The function above returns a complex number in polar form.

The definition of `magnitude` requires a square root computation. For this task, a call to the `sqrt` function found in the `IEEE.math_real` library is used.

`complexpack` is distributed along with a simple VHDL testbench exercising basic functionalities.

2. File listing

The `complexpack` distribution includes the following files:

<code>/complexpack</code>	Top-level directory
<code>ChangeLog</code>	A log for code changes.
<code>LICENSE</code>	The modified BSD license governs <code>complexpack</code> since version 0.2.0.
<code>README.rst</code>	This file.
<code>README.html</code>	HTML version of <code>README.rst</code> .
<code>README.pdf</code>	PDF version of <code>README.rst</code> .
<code>VERSION</code>	Current version of the project sources.
<code>complexpack.png</code>	PNG image for the <code>complexpack</code> project logo.
<code>rst2docs.sh</code>	Bash script for generating the HTML and PDF versions.
<code>/bench/vhdl</code>	Benchmarks VHDL directory
<code>complexpack_tb.vhd</code>	A simple testbench.
<code>/doc</code>	Documentation directory
<code>/rtl/vhdl</code>	RTL source code directory for the package
<code>complexpack.vhd</code>	The complex arithmetic package.
<code>/sim/rtl_sim</code>	RTL simulation files directory
<code>/sim/rtl_sim/bin</code>	RTL simulation makefiles directory
<code>complexpack.mk</code>	GNU Makefile for running GHDL simulations.
<code>/sim/rtl_sim/out</code>	RTL simulation output files directory
<code>complexpack_results-.txt</code>	Output generated by the <code>complexpack_tb.vhd</code> test.
<code>/sim/rtl_sim/run</code>	RTL simulation run scripts directory
<code>clean.sh</code>	A bash script for cleaning simulation artifacts.
<code>run.sh</code>	A bash script for running the GNU Makefile for GHDL.

3. complexpack usage

The `complexpack` package test script can be used as follows:

```
$ ./run.sh
```

as run from within the `./sim/rtl_sim/run` subdirectory. The run script expects that the GHDL simulator is installed and its `bin` directory is in the `$PATH`.

After this process, the `complexpack_results.txt` file is generated containing simulation results.

A reference `complexpack_results.txt` is kept under `./sim/rtl_sim/out` for comparison.

To clean up afterwards, use:

```
$ ./clean.sh
```

4. Prerequisites

- Standard UNIX-based tools (tested on cygwin/x86 and MinGW/x86 and MinGW/x64)
 - make
 - bash
- GHDL simulator (<http://ghdl.free.fr>)

Provides the "ghdl" executable and corresponding simulation environment. Versions throughout 0.26 to 0.33 have been used for testing.