

ratpack user manual



Title	ratpack (VHDL rational arithmetic package).
Author	Nikolaos Kavvadias 2009-2020
Contact	nikolaos.kavvadias@gmail.com
Website	http://www.nkavvadias.com
Release Date	23 July 2016
Version	0.3.5
Rev. history	
v0.3.5	2016-23-07 Use sim/rtl_sim/bin dir, ignores for git, clean script
v0.3.4	2016-03-12 Updates dates, Changelog.
v0.3.3	2014-11-29 Added project logo in README.
v0.3.2	2014-10-14 Replaced COPYING by LICENSE.
v0.3.1	2014-09-22 New features and changes: <ul style="list-style-type: none">• Testing of max, rmin (min), rmax (max).• Automatic end of simulation (via forced assert).• Renamed run.sh to ghdl.sh.• Added mti.sh and ratpack_tb1/tb2.do for Modelsim.• README update.
v0.3.0	2014-09-22 Updated for github (contents of /doc moved to top-level directory, minor documentation changes).
v0.2.0	2014-02-21 Changed documentation format to RestructuredText. Code has been reorganized into new directory structure.

v0.1.3	2010-11-17 Added max, min.
v0.1.2	2010-11-17 Added gcditer (iterative GCD using rational numbers).
v0.1.1	2010-06-07 Minor update in documentation (README).
v0.1.0	2010-05-14 First public release.

1. Introduction

`ratpack` is a rational arithmetic package written in VHDL.

Currently, the `ratpack` package implements the following:

- the `RATIONAL` data type.
- `to_rational`: construction function of a rational given two integers (numerator and denominator).
- `int2rat`: conversion function of an integer to its rational representation.
- `numerator`: extracts the numerator of a rational number.
- `denominator`: extracts the denominator of a rational number.
- `"+", "-", "*", "/"`: implementation of the basic arithmetic operations for rationals.
- `abs`: extracts the absolute value of a given rational number.
- `max`: extracts the maximum of two rationals.
- `min`: extracts the minimum of two rationals.
- `">", "<", ">=", "<=", "=", "/="`: overload comparison operators for rationals.
- `gcd`: computes the greatest common divisor of two integers (positive, covers the pathological case of division by zero).
- `mediant`: computes the mediant rational of two given rationals.

`ratpack` is distributed along with two VHDL testbenches: a simple one (`ratpack_tb1.vhd`) and a testbench generating the Farey series of orders 1 to 12 (`ratpack_tb2.vhd`). An exemplary rational arithmetic ALU has also been included but it is currently left untested (not testbench for it).

The `ratpack` project can be download from the following OpenCores website:
<http://opencores.org/project,ratpack>

An up-to-date version of the `ratpack` code base is also maintained on Github:
<http://github.com/nkkav/ratpack.git>

2. File listing

The `ratpack` distribution includes the following files:

<code>/ratpack</code>	Top-level directory
<code>AUTHORS</code>	List of <code>ratpack</code> authors.
<code>LICENSE</code>	The modified BSD license governs <code>ratpack</code> .
<code>README.rst</code>	This file.
<code>README.html</code>	HTML version of <code>README</code> .
<code>README.pdf</code>	PDF version of <code>README</code> .
<code>ratpack.png</code>	PNG image for the <code>ratpack</code> project logo.
<code>rst2docs.sh</code>	Bash script for generating the HTML and PDF versions.
<code>VERSION</code>	Current version of the project sources.
<code>/bench/vhdl</code>	Testbench source code directory for the package
<code>ratpack_tb1.vhd</code>	A simple testbench.
<code>ratpack_tb2.vhd</code>	Testbench generating the Farey series (orders 1-12).
<code>/rtl/vhdl</code>	RTL source code directory for the package
<code>ratalu.vhd</code>	Implementation of a rational arithmetic ALU.
<code>ratpack.vhd</code>	The rational arithmetic package.
<code>/sim/rtl_sim</code>	RTL simulation files directory
<code>/sim/rtl_sim/bin</code>	RTL simulation makefiles directory
<code>ratpack.mk</code>	GNU Makefile for running GHDL simulations.
<code>ratpack_tb1.do</code>	Modelsim <code>.do</code> script for running the simulation for testbench <code>ratpack_tb1.vhd</code> .
<code>ratpack_tb2.do</code>	Modelsim <code>.do</code> script for running the simulation for testbench <code>ratpack_tb2.vhd</code> .
<code>/sim/rtl_sim/out</code>	RTL simulation output files directory
<code>ratpack_results1.txt</code>	Output generated by the <code>ratpack_tb1.vhd</code> tests.
<code>ratpack_results2.txt</code>	Output generated by the <code>ratpack_tb2.vhd</code> tests.
<code>/sim/rtl_sim/run</code>	RTL simulation run scripts directory
<code>clean.sh</code>	A bash script for cleaning simulation artifacts.
<code>ghdl.sh</code>	A bash script for running the GNU Makefile for GHDL.
<code>mti.sh</code>	A bash script for running the Mentor Modelsim simulation for either testbench.

3. `ratpack` usage

Contents of the `ratpack` distribution can be simulated using either GHDL or Mentor Modelsim using the provided scripts.

3.1. GHDL

The `ratpack` package test script for GHDL can be used as follows:

```
$. /ghdl.sh <package/entity name> <test case>
```

from within directory `ratpack/sim/rtl_sim/run`.

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

Here follow some simple usage examples of this bash script.

1. Compile the `ratpack` package and do a simple test.

```
$ cd sim/rtl_sim/run
$ ./ghdl.sh ratpack 1
```

The default results for comparison can be found as `sim/rtl_sim/out/ratpack_results1.txt`

2. Compile the `ratpack` package and generate the Farey series.

```
$ ./ghdl.sh ratpack 2
```

The default results for comparison can be found as `sim/rtl_sim/out/ratpack_results2.txt`

The run script expects that the GHDL simulator is installed and its `bin` directory is in the `$PATH`.

3.2 Modelsim

For running the Modelsim simulation, the `ratpack` package test script for can be used as follows:

```
$. /mti.sh <package/entity name> <test case>
```

from within directory `ratpack/sim/rtl_sim/run`.

1. Compile the `ratpack` package and do a simple test.

```
$ cd sim/rtl_sim/run
$ ./mti.sh ratpack 1
```

As in the GHDL case, a diagnostic text file is produced. The default results for comparison can be found as `sim/rtl_sim/out/ratpack_results1.txt`

2. Compile the `ratpack` package and generate the Farey series.

```
$ ./mti.sh ratpack 2
```

The default results for comparison can be found as `sim/rtl_sim/out/ratpack_results2.txt`

4. Prerequisites

- Standard UNIX-based tools (tested on cygwin/x86 and MinGW/x64)
 - `make`

– bash

- GHDL simulator (<http://ghdl.free.fr>)
Provides the "ghdl" executable and corresponding simulation environment.
- Alternatively to GHDL, the Mentor Modelsim simulator (<http://www.mentor.com>)