

mpintpack user manual

Title	mpintpack (VHDL multi-precision integer arithmetic package).
Author	Nikolaos Kavvadias 2012-2020
Contact	nikolaos.kavvadias@gmail.com
Website	http://www.nkavvadias.com
Release Date	27 December 2020
Version	0.1.0
Rev. history	
v0.1.0	2020-12-27 First public release

1. Introduction

mpintpack is a multi-precision integer arithmetic package written in VHDL. The multi-precision integer numbers may take up to MAXDIGITS and the underlying arithmetic is implemented in base-256 as each chunk is an unsigned integer of 8 bits wide.

Two types are provided for working with the multi-precision integer API:

- `MP_INT_ARR` where the underlying storage is an integer range 0 to 255
- `MP_SLV_ARR` where the underlying storage is a `std_logic_vector` of the corresponding width

Currently, the mpintpack package implements the following:

- the `MP_INT_ARR` and `MP_SLV_ARR` data types.
- `mpz_init`: initialize the storage for the number
- `mpz_set_ui`: assign the MP number from an unsigned integer or a `std_logic_vector` respectively
- `mpz_set`: assign the MP number from another
- `mpz_add`: multi-precision addition
- `mpz_add_ui`: add an unsigned integer or a `std_logic_vector` to an MP number
- `mpz_sub`: multi-precision subtraction

- `mpz_sub_ui`: subtract an unsigned integer or a `std_logic_vector` from an MP number
- `mpz_mul`: multi-precision multiplication
- `mpz_mul_ui`: multiply an unsigned integer or a `std_logic_vector` with an MP number
- `mpz_get_str`: convert the MP number to a string
- `mpz_printh`: print the MP number in hexadecimal

The API is loosely based (or loosely follows) the conventions set by the `fmpz` public-domain multi-precision arithmetic library.

2. File listing

The `mpintpack` distribution includes the following files:

<code>/mpintpack</code>	Top-level directory
<code>LICENSE</code>	The modified BSD license governs <code>mpintpack</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>rst2docs.sh</code>	Bash script for generating the HTML and PDF versions.
<code>/bench/vhdl</code>	Testbench source code directory for the package
<code>gmpfact.vhd</code>	Multi-precision factorial.
<code>gmpfact_slv.vhd</code>	Multi-precision factorial using <code>std_logic_vector</code> .
<code>gmpfibo.vhd</code>	Multi-precision Fibonacci series generator.
<code>gmpfibo_slv.vhd</code>	Multi-precision Fibonacci series generator using <code>std_logic_vector</code>
<code>/rtl/vhdl</code>	RTL source code directory for the package
<code>mpintpack.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>test.mk</code>	GNU Makefile for running GHDL simulations.
<code>/sim/rtl_sim/out</code>	RTL simulation output files directory
<code>gmpfact_results.txt</code>	Output generated by corresponding test.
<code>gmpfact_slv_results.txt</code>	Output generated by corresponding test.
<code>gmpfibo_results.txt</code>	Output generated by corresponding test.
<code>gmpfibo_slv_results.txt</code>	Output generated by corresponding test.
<code>/sim/rtl_sim/run</code>	RTL simulation run scripts directory
<code>clean.sh</code>	A bash script for cleaning simulation artifacts.
<code>test.sh</code>	A bash script for running the GNU Makefile for GHDL.

3. mpintpack usage

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

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

```
$ ./test.sh <test case>
```

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

After this process, the `<test>_results.txt` file is generated containing simulation results.

Here follows a simple usage example of this bash script:

Compile the `mpintpack` package and generate factorials from 1 to 20.

```
$ cd sim/rtl_sim/run
$ ./test.sh gmpfact
```

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

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

4. Prerequisites

- Standard UNIX-based tools

- make
- bash
- tee

- GHDL simulator (<http://ghdl.free.fr>)

Provides the "ghdl" executable and corresponding simulation environment.