# 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<br>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 fgmp public-domain multi-precision arithmetic library.

## 2. File listing

The mpintpack distribution includes the following files:

| | |
|---|---|
| /mpintpack | Top-level directory |
| LICENSE | The modified BSD license governs mpintpack. |
| README.rst | This file. |
| README.html | HTML version of README. |
| README.pdf | PDF version of README. |
| rst2docs.sh | Bash script for generating the HTML and PDF versions. |
| /bench/vhdl | Testbench source code directory for the package |
| gmpfact.vhd | Multi-precision factorial. |
| gmpfact_slv.vhd | Multi-precision factorial using std_logic_vector |
| gmpfibo.vhd | Multi-precision factorial. |
| gmpfibo_slv.vhd | Multi-precision factorial using std_logic_vector |
| /rtl/vhdl | RTL source code directory for the package |
| mpintpack.vhd | The rational arithmetic package. |
| /sim/rtl_sim | RTL simulation files directory |
| /sim/rtl_sim/bin | RTL simulation makefiles directory |
| test.mk | GNU Makefile for running GHDL simulations. |
| /sim/rtl_sim/out | RTL simulation output files directory |
| gmpfact_results.txt | Output generated by corresponding test. |
| gmpfact_slv_results.txt \| Output generated by corresponding test. | |
| gmpfibo_results.txt | Output generated by corresponding test. |
| gmpfibo_slv_results.txt \| Output generated by corresponding test. | |
| /sim/rtl_sim/run | RTL simulation run scripts directory |
| clean.sh | A bash script for cleaning simulation artifacts. |
| test.sh | 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
$ ./ghdl.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.