

UNISIM ARMEmu Simulator Manual

Daniel Gracia Pérez

Contents

1 Simulator technical reference (generated)	2
1.1 Introduction	2
1.2 Licensing	2
1.3 Simulated configuration	2
1.4 Using the UNISIM ARMEmu simulator	3
1.5 Configuration	4
1.6 Statistics	14
1.7 Formulas	16

List of Figures

1 UNISIM ARMEmu simulator schematic.	2
--	---

List of Tables

1 Simulator technical reference (generated)

This documentation has been automatically generated from the simulator UNISIM ARMEmu version 0.7.1 on Jun 26 2013.

1.1 Introduction

UNISIM ARMv5 User Level Simulator.

Section 1.2 gives licensing informations about the simulator. Section 1.3 shows the set of modules and services that compose the simulator. Section 1.4 shows how to invoke the simulator at the command line prompt. Section 1.5 gives the simulator parameters. Section 1.6 gives the simulator statistic counters. Section 1.7 gives the simulator statistic formulas.

1.2 Licensing

UNISIM ARMEmu 0.7.1

Copyright (C) 2007-2010, Commissariat a l'Energie Atomique

License: BSD (See file COPYING)

Authors: Daniel Gracia Perez (daniel.gracia-perez@cea.fr)

1.3 Simulated configuration

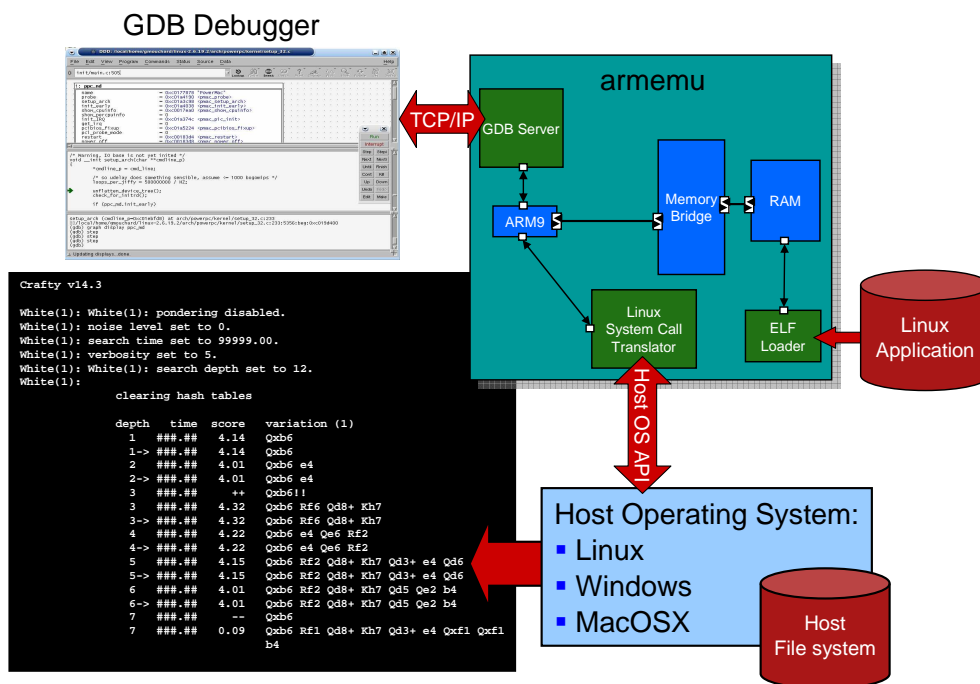


Figure 1: UNISIM ARMEmu simulator schematic.

The UNISIM ARMEmu simulator is composed of the following modules and services:

- **cpu**

- **cpu.dcache**
- **cpu.icache**
- **debugger**
- **gdb-server**: this service implements the GDB server remote serial protocol over TCP/IP. Standards GDB clients (e.g. gdb, eclipse, ddd) can connect to the simulator to debug the target application that runs within the simulator.
- **host-time**: this service is an abstraction layer for the host machine time
- **inline-debugger**: this service implements a built-in debugger in the terminal console
- **linux-os**
- **memory**: this module implements a memory
- **profiler**
- **tee-memory-access-reporting**
- **tee-memory-access-reporting.tee-memory-access-reporting.control_selector[0]**
- **tee-memory-access-reporting.tee-memory-access-reporting.control_selector[10]**
- **tee-memory-access-reporting.tee-memory-access-reporting.control_selector[11]**
- **tee-memory-access-reporting.tee-memory-access-reporting.control_selector[12]**
- **tee-memory-access-reporting.tee-memory-access-reporting.control_selector[13]**
- **tee-memory-access-reporting.tee-memory-access-reporting.control_selector[14]**
- **tee-memory-access-reporting.tee-memory-access-reporting.control_selector[15]**
- **tee-memory-access-reporting.tee-memory-access-reporting.control_selector[1]**
- **tee-memory-access-reporting.tee-memory-access-reporting.control_selector[2]**
- **tee-memory-access-reporting.tee-memory-access-reporting.control_selector[3]**
- **tee-memory-access-reporting.tee-memory-access-reporting.control_selector[4]**
- **tee-memory-access-reporting.tee-memory-access-reporting.control_selector[5]**
- **tee-memory-access-reporting.tee-memory-access-reporting.control_selector[6]**
- **tee-memory-access-reporting.tee-memory-access-reporting.control_selector[7]**
- **tee-memory-access-reporting.tee-memory-access-reporting.control_selector[8]**
- **tee-memory-access-reporting.tee-memory-access-reporting.control_selector[9]**
- **time**: this service is an abstraction layer for the SystemC kernel time

1.4 Using the UNISIM ARMEMu simulator

The UNISIM ARMEMu simulator has the following command line options:

Usage: `unisim-armemu-0.7.1 [<options>] [...]`

Options:

- `--set <param=value>` or `-s <param=value>`: set value of parameter 'param' to 'value'
- `--config <XML file>` or `-c <XML file>`: configures the simulator with the given XML configuration file
- `--get-config <XML file>` or `-g <XML file>`: get the simulator configuration XML file (you can use it to create your own configuration. This option can be combined with `-c` to get a new configuration file with existing variables from another file)
- `--list` or `-l`: lists all available parameters, their type, and their current value

- `--warn` or `-w`: enable printing of kernel warnings
- `--doc <Latex file>` or `-d <Latex file>`: enable printing a latex documentation
- `--version` or `-v`: displays the program version information
- `--share-path <path>` or `-p <path>`: the path that should be used for the share directory (absolute path)
- `--help` or `-h`: displays this help

1.5 Configuration

Simulator configuration (see below) can be modified using command line Options `--set <param=value>` or `--config <config file>`.

Global	
Name: <code>enable-gdb-server</code> Default: <code>true</code> Valid: <code>true, false</code>	Type: parameter Data type: boolean
Description: Enable GDB server..	
Name: <code>enable-inline-debugger</code> Default: <code>true</code> Valid: <code>true, false</code>	Type: parameter Data type: boolean
Description: Enable inline debugger..	
Name: <code>enable-power-estimation</code> Default: <code>false</code> Valid: <code>true, false</code>	Type: parameter Data type: boolean
Description: Activate caches power estimation..	
Name: <code>enable-press-enter-at-exit</code> Default: <code>false</code> Valid: <code>true, false</code>	Type: parameter Data type: boolean
Description: Enable/Disable pressing key enter at exit.	
Name: <code>kernel_logger.file</code> Default: <code>false</code> Valid: <code>true, false</code>	Type: parameter Data type: boolean
Description: Keep logger output in a file.	
Name: <code>kernel_logger.filename</code> Default: <code>logger_output.txt</code>	Type: parameter Data type: string
Description: Filename to keep logger output _(the option file must be activated).	

Name: kernel_logger.std_err Default: true Valid: true, false	Type: parameter Data type: boolean
Description: Show logger output through the standard error output.	
Name: kernel_logger.std_err_color Default: true Valid: true, false	Type: parameter Data type: boolean
Description: Colorize logger output through the standard error output _(only works if std_err is active).	
Name: kernel_logger.std_out Default: false Valid: true, false	Type: parameter Data type: boolean
Description: Show logger output through the standard output.	
Name: kernel_logger.std_out_color Default: false Valid: true, false	Type: parameter Data type: boolean
Description: Colorize logger output through the standard output _(only works if std_out is active).	
Name: kernel_logger.xml_file Default: false Valid: true, false	Type: parameter Data type: boolean
Description: Keep logger output in a file xml formatted.	
Name: kernel_logger.xml_file_gzipped Default: false Valid: true, false	Type: parameter Data type: boolean
Description: If the xml_file option is active, the output file will be compressed (a .gz extension will be automatically added to the xml_filename option.	
Name: kernel_logger.xml_filename Default: logger_output.xml	Type: parameter Data type: string
Description: Filename to keep logger xml output _(the option xml_file must be activated).	
cpu	
Name: cpu.default-endianness Default: little-endian	Type: parameter Data type: string

Description: The processor default/boot endianness. Available values are: little-endian and big-endian..	
Name: cpu.voltage Default: 1800	Type: parameter Data type: unsigned 64-bit integer
Description: The processor voltage in mV..	
Name: cpu.verbose Default: 0x00000000	Type: parameter Data type: unsigned 32-bit integer
Description: Activate the verbose system (0 = inactive, different than 0 = active)..	
Name: cpu.trap-on-instruction-counter Default: 0	Type: parameter Data type: unsigned 64-bit integer
Description: Produce a trap when the given instruction count is reached..	
Name: cpu.cpu-cycle-time Default: 31250 ps	Type: parameter Data type: sc_time
Description: The processor cycle time..	
Name: cpu.bus-cycle-time Default: 31250 ps	Type: parameter Data type: sc_time
Description: The processor bus cycle time..	
Name: cpu.nice-time Default: 1 ms	Type: parameter Data type: sc_time
Description: Maximum time between SystemC waits..	
Name: cpu.ipc Default: 1	Type: parameter Data type: double precision floating-point
Description: Instructions per cycle performance..	
Name: cpu.verbose-tlm	Type: parameter

Default: false	Data type: boolean
Valid: true, false	
Description: Display TLM information.	
cpu.dcache	
Name: cpu.dcache.size	Type: parameter
Default: 131072	Data type: unsigned 32-bit integer
Description: Size of the cache in bytes. Available sizes are 4KB, 8KB, 16KB, 32KB, 64KB and 128KB. The cache can be deactivated setting this value to 0..	
cpu.icache	
Name: cpu.icache.size	Type: parameter
Default: 131072	Data type: unsigned 32-bit integer
Description: Size of the cache in bytes. Available sizes are 4KB, 8KB, 16KB, 32KB, 64KB and 128KB. The cache can be deactivated setting this value to 0..	
debugger	
Name: debugger.verbose	Type: parameter
Default: false	Data type: boolean
Valid: true, false	
Description: Enable/Disable verbosity.	
Name: debugger.dwarf-to-html-output- ↔directory	Type: parameter
Default:	Data type: string
Description: DWARF v2/v3 to HTML output directory.	
Name: debugger.dwarf-register-number- ↔mapping-filename	Type: parameter
Default: arm_eabi_dwarf_register_ ↔number_mapping.xml	Data type: string
Description: DWARF register number mapping filename.	
Name: debugger.parse-dwarf	Type: parameter
Default: true	Data type: boolean
Valid: true, false	

Description: Enable/Disable parsing of DWARF debugging informations.	
Name: debugger.debug-dwarf	Type: parameter
Default: false	Data type: boolean
Valid: true, false	
Description: Enable/Disable debugging of DWARF.	
gdb-server	
Name: gdb-server.memory-atom-size	Type: parameter
Default: 0x00000001	Data type: unsigned 32-bit integer
Description: size of the smallest addressable element in memory.	
Name: gdb-server.tcp-port	Type: parameter
Default: 12345	Data type: signed 32-bit integer
Description: TCP/IP port to listen waiting for a GDB client connection.	
Name: gdb-server.architecture-description	Type: parameter
↔filename	
Default: gdb_armv5l.xml	Data type: string
Description: filename of a XML description of the connected processor.	
Name: gdb-server.verbose	Type: parameter
Default: false	Data type: boolean
Valid: true, false	
Description: Enable/Disable verbosity.	
inline-debugger	
Name: inline-debugger.memory-atom-size	Type: parameter
↔size	
Default: 0x00000001	Data type: unsigned 32-bit integer
Description: size of the smallest addressable element in memory.	
Name: inline-debugger.search-path	Type: parameter
Default:	Data type: string
Description: Search path for source (separated by ';').	

Name: inline-debugger.init-macro Default:	Type: parameter Data type: string
Description: path to initial macro to run when debugger starts.	
Name: inline-debugger.output Default:	Type: parameter Data type: string
Description: path to output file where to redirect the debugger outputs.	
linux-os	
Name: linux-os.verbose Default: false Valid: true, false	Type: parameter Data type: boolean
Name: linux-os.parse-dwarf Default: false Valid: true, false	Type: parameter Data type: boolean
Name: linux-os.debug-dwarf Default: false Valid: true, false	Type: parameter Data type: boolean
Name: linux-os.dwarf-to-html-output- ↔directory Default:	Type: parameter Data type: string
Name: linux-os.dwarf-to-xml-output- ↔filename Default:	Type: parameter Data type: string
Name: linux-os.system Default: arm-eabi	Type: parameter Data type: string
Description: Emulated system architecture available values are "arm", "arm-eabi" and "powerpc".	
Name: linux-os.endianness Default: little-endian Valid: little-endian, big-endian	Type: parameter Data type: endianness
Description: The endianness of the binary loaded. Available values are: little-endian and big-endian..	
Name: linux-os.memory-page-size Default: 0x00001000	Type: parameter Data type: unsigned 32-bit integer
Name: linux-os.stack-base Default: 0x40000000	Type: parameter Data type: unsigned 32-bit integer

Description:

The stack base address used for the load and execution of the linux application.

Name: linux-os.binary

Type: parameter

Default:

Data type: string

Description:

The binary to execute on the target simulator. Usually it is the same value than the argv[1] parameter..

Name: linux-os.argc

Type: parameter

Default: 0

Data type: unsigned 32-bit integer

Description:

Number of commands in the program execution line (usually at least one which is the name of the program executed). The different tokens can be set up with the parameters argv[<n>] where <n> can go up to argc - 1..

Name: linux-os.apply-host-environment

Type: parameter

Default: false

Data type: boolean

Valid: true, false

Description:

Whether to apply the host environment on the target simulator or use the provided envc and envp..

Name: linux-os.envc

Type: parameter

Default: 0x00000000

Data type: unsigned 32-bit integer

Description:

Number of environment variables defined for the program execution. The different variables can be set up with the parameters envp[<n>] where <n> can go up to envc - 1..

Name: linux-os.utsname-sysname

Type: parameter

Default: Linux

Data type: string

Description:

The value that the uname system call should return. As this service is providing linux emulation support its value should be 'Linux', so you should not modify it..

Name: linux-os.utsname-nodename

Type: parameter

Default: localhost

Data type: string

Description:

The network node hostname that the uname system call should return. Default value is localhost, but you could write whatever name you want..

Name: linux-os.utsname-release Default: 2.6.27.35	Type: parameter Data type: string
Description: The kernel release information that the uname system call should return. This should usually match the linux-kernel parameter..	
Name: linux-os.utsname-version Default: #UNISIM SMP Fri Mar 12 05:23:09 ↔UTC 2010	Type: parameter Data type: string
Description: The kernel version information that the uname system call should return..	
Name: linux-os.utsname-machine Default: armv5	Type: parameter Data type: string
Description: The machine information that the uname system call should return. This should be one of the supported architectures (the system parameter, that is, arm or powerpc) or a specific model derived from it (i.e., arm926ejs)..	
Name: linux-os.utsname-domainname Default: localhost	Type: parameter Data type: string
Description: The domain name information that the uname system call should return..	
Name: linux-os.hwcap Default: swp half fastmult	Type: parameter Data type: string
Description: CPU Hardware capabilities to enable (e.g. "swp thumb fastmult vfp"..	
memory	
Name: memory.org Default: 0x00000000	Type: parameter Data type: unsigned 32-bit integer
Description: memory origin/base address.	
Name: memory.bytesize Default: 4294967295	Type: parameter Data type: unsigned 32-bit integer
Description: memory size in bytes.	

Name: memory.initial-byte-value Default: 0x00	Type: parameter Data type: unsigned 8-bit integer
Name: memory.cycle-time Default: 31250 ps	Type: parameter Data type: sc_time
Description: memory cycle time.	
Name: memory.read-latency Default: 31250 ps	Type: parameter Data type: sc_time
Description: memory read latency.	
Name: memory.write-latency Default: 0 s	Type: parameter Data type: sc_time
Description: memory write latency.	
Name: memory.verbose Default: false Valid: true, false	Type: parameter Data type: boolean
Description: enable/disable verbosity.	
profiler	
Name: profiler.min-data-read-prof- ↔addr Default: 0x00000000	Type: parameter Data type: unsigned 32-bit integer
Description: Minimum address for data read profiling.	
Name: profiler.max-data-read-prof- ↔addr Default: 0xffffffff	Type: parameter Data type: unsigned 32-bit integer
Description: Maximum address for data read profiling.	
Name: profiler.min-data-write-prof- ↔addr Default: 0x00000000	Type: parameter Data type: unsigned 32-bit integer

Description: Minimum address for data write profiling.	
Name: profiler.max-data-write-prof- ↔addr	Type: parameter
Default: 0xffffffff	Data type: unsigned 32-bit integer
Description: Maximum address for data write profiling.	
Name: profiler.min-insn-fetch-prof- ↔addr	Type: parameter
Default: 0x00000000	Data type: unsigned 32-bit integer
Description: Minimum address for instruction fetch profiling.	
Name: profiler.max-insn-fetch-prof- ↔addr	Type: parameter
Default: 0xffffffff	Data type: unsigned 32-bit integer
Description: Maximum address for instruction fetch profiling.	
Name: profiler.min-insn-exec-prof- ↔addr	Type: parameter
Default: 0x00000000	Data type: unsigned 32-bit integer
Description: Minimum address for instruction execution profiling.	
Name: profiler.max-insn-exec-prof- ↔addr	Type: parameter
Default: 0xffffffff	Data type: unsigned 32-bit integer
Description: Maximum address for instruction execution profiling.	
Name: profiler.enable-data-read- ↔prof	Type: parameter
Default: false	Data type: boolean
Valid: true, false	
Description: Enable/Disable data read profiling.	
Name: profiler.enable-data-write- ↔prof	Type: parameter
Default: false	Data type: boolean

Valid: true, false	
Description: Enable/Disable data write profiling.	
Name: profiler.enable-insn-fetch- ↔prof	Type: parameter
Default: false	Data type: boolean
Valid: true, false	
Description: Enable/Disable instruction fetch profiling.	
Name: profiler.enable-insn-exec- ↔prof	Type: parameter
Default: false	Data type: boolean
Valid: true, false	
Description: Enable/Disable instruction execution profiling.	
Name: profiler.verbose	Type: parameter
Default: false	Data type: boolean
Valid: true, false	
Description: Enable/Disable verbosity.	

1.6 Statistics

Simulation statistic counters are listed below:

cpu	
Name: cpu.instruction-counter	Type: statistic Data type: unsigned 64-bit integer
Description: Number of instructions executed..	
Name: cpu.cpu-time	Type: statistic Data type: sc_time
Description: The processor time.	
cpu.dcache	
Name: cpu.dcache.read-accesses	Type: statistic Data type: unsigned 32-bit integer
Description: Number of read accesses to the cache..	
Name: cpu.dcache.write-accesses	Type: statistic Data type: unsigned 32-bit integer

Description: Number of write accesses to the cache..	
Name: <code>cpu.dcache.prefetch-accesses</code>	Type: statistic Data type: unsigned 32-bit integer
Description: Number of prefetch accesses to the cache..	
Name: <code>cpu.dcache.read-hits</code>	Type: statistic Data type: unsigned 32-bit integer
Description: Number of read hit accesses to the cache..	
Name: <code>cpu.dcache.write-hits</code>	Type: statistic Data type: unsigned 32-bit integer
Description: Number of write hit accesses to the cache..	
Name: <code>cpu.dcache.prefetch-hits</code>	Type: statistic Data type: unsigned 32-bit integer
Description: Number of prefetch hit accesses to the cache..	
cpu.icache	
Name: <code>cpu.icache.read-accesses</code>	Type: statistic Data type: unsigned 32-bit integer
Description: Number of read accesses to the cache..	
Name: <code>cpu.icache.write-accesses</code>	Type: statistic Data type: unsigned 32-bit integer
Description: Number of write accesses to the cache..	
Name: <code>cpu.icache.prefetch-accesses</code>	Type: statistic Data type: unsigned 32-bit integer
Description: Number of prefetch accesses to the cache..	

Name: <code>cpu.icache.read-hits</code>	Type: statistic Data type: unsigned 32-bit integer
Description: Number of read hit accesses to the cache..	
Name: <code>cpu.icache.write-hits</code>	Type: statistic Data type: unsigned 32-bit integer
Description: Number of write hit accesses to the cache..	
Name: <code>cpu.icache.prefetch-hits</code>	Type: statistic Data type: unsigned 32-bit integer
Description: Number of prefetch hit accesses to the cache..	
memory	
Name: <code>memory.memory-usage</code>	Type: statistic Data type: unsigned 32-bit integer
Description: target memory usage in bytes (page granularity of 1048576 bytes).	
Name: <code>memory.read-counter</code>	Type: statistic Data type: unsigned 64-bit integer
Description: read counter.	
Name: <code>memory.write-counter</code>	Type: statistic Data type: unsigned 64-bit integer
Description: write counter.	

1.7 Formulas

Simulation statistic formulas are listed below:

cpu.dcache

Name: cpu.dcache.accesses

Type: formula

Formula: cpu.dcache.read-accesses

Data type: signed 32-bit integer

↪+ cpu.dcache.write-accesses

Description:

Number of accesses to the cache..

Name: cpu.dcache.hits

Type: formula

Formula: cpu.dcache.read-hits + cpu.

Data type: signed 32-bit integer

↪dcache.write-hits

Description:

Number of hit accesses to the cache..

Name: `cpu.dcache.hit-rate`

Type: formula

Formula: `cpu.dcache.read-hits + cpu.`

Data type:

double precision

`↔dcache.write-hits / cpu.`

floating-point

`↔dcache.read-accesses + cpu.`

`↔dcache.write-accesses`

Description:

Cache hit rate..

cpu.icache

Name: `cpu.icache.accesses`

Type: formula

Formula: `cpu.icache.read-accesses`

Data type: signed 32-bit integer

`↔+ cpu.icache.write-accesses`

Description:

Number of accesses to the cache..

Name: `cpu.icache.hits`

Type: formula

Formula: `cpu.icache.read-hits + cpu.`

Data type: signed 32-bit integer

`↔icache.write-hits`

Description:

Number of hit accesses to the cache..

Name: `cpu.icache.hit-rate`

Type: formula

Formula: `cpu.icache.read-hits + cpu.`

Data type:

double precision

`↔icache.write-hits / cpu.`

floating-point

`↔icache.read-accesses + cpu.`

`↔icache.write-accesses`

Description:

Cache hit rate..