

UNISIM

avr32emu Simulator Manual

Gilles Mouchard

1 Simulator technical reference (generated)

This documentation has been automatically generated from the simulator UNISIM AVR32EMU version 0.1 on Dec 12 2014.

1.1 Introduction

UNISIM AVR32EMU, AVR32A 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 AVR32EMU 0.1

Copyright (C) 2014, Commissariat a l'Energie Atomique (CEA)

License: BSD (see file COPYING)

Authors: Julien Lisita <julien.lisita@cea.fr>, Gilles Mouchard <gilles.mouchard@cea.fr>

1.3 Simulated configuration

The UNISIM AVR32EMU simulator is composed of the following modules and services:

- **avr32-t2h-syscalls**
- **cpu**: This module implements an AVR32UC CPU core.
- **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
- **irq-stub0**: An initiator stub
- **irq-stub1**: An initiator stub
- **irq-stub2**: An initiator stub
- **irq-stub3**: An initiator stub
- **loader**: A multi-format loader that supports ELF32, ELF64, S19, COFF and Raw binary files
- **loader.memory-mapper**: A memory mapper

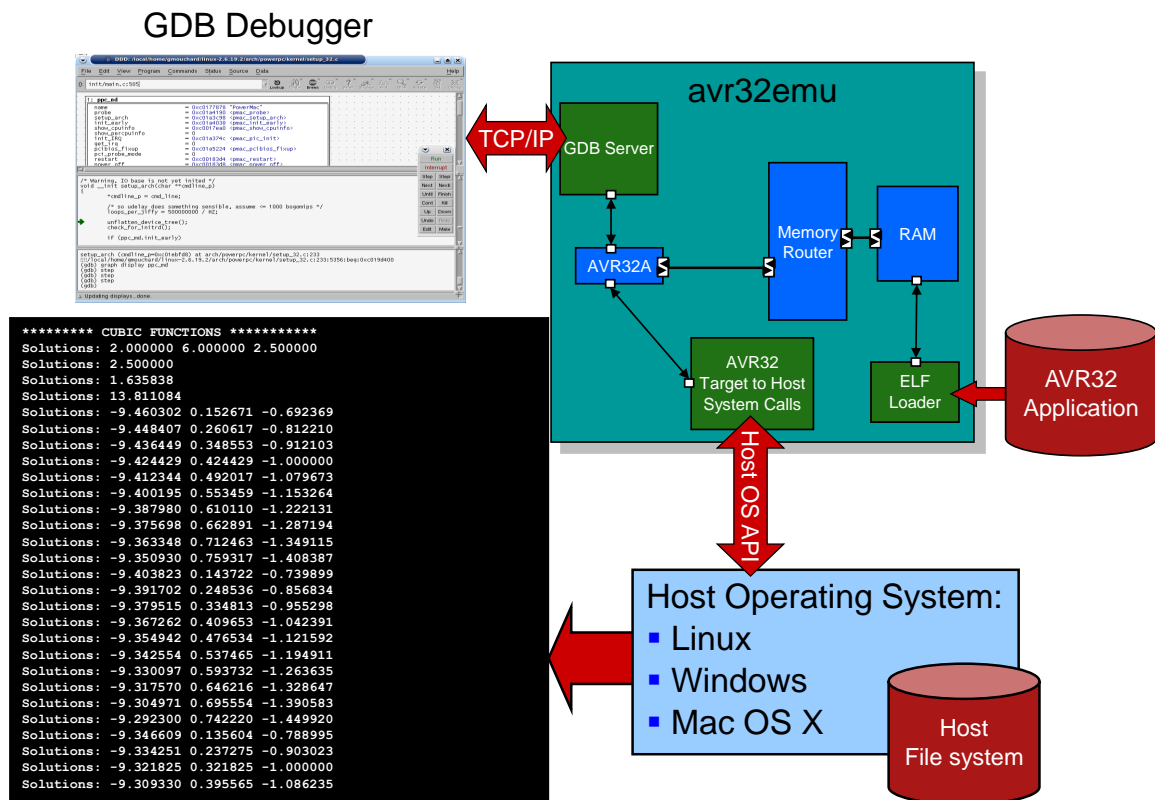


Figure 1: UNISIM AVR32EMU simulator schematic.

- **loader.tee-backtrace:** This service/client implements a tee ('T'). It unifies the backtrace capability of several services that individually provides their own backtrace capability
- **loader.tee-blob:** This service/client implements a tee ('T'). It unifies the statement lookup capability of several services that individually provides their own statement lookup capability
- **loader.tee-loader:** This service/client implements a tee ('T'). It unifies the loader capability of several services that individually provides their own loader capability
- **loader.tee-stmt-lookup:** This service/client implements a tee ('T'). It unifies the statement lookup capability of several services that individually provides their own statement lookup capability
- **loader.tee-symbol-table-lookup:** This service/client implements a tee ('T'). It unifies the symbol table lookup capability of several services that individually provides their own symbol table lookup capability
- **memory-router:** A memory-mapped router
- **nmreq-stub:** An initiator stub
- **profiler**
- **ram:** this module implements a memory
- **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 AVR32EMU simulator

The UNISIM AVR32EMU simulator has the following command line options:

Usage: `unisim-avr32emu-0.1 [<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>	Type: parameter
Default: <code>true</code>	Data type: boolean

Valid: true, false	
Description: Enable/Disable GDB server instantiation.	
Name: enable-inline-debugger	Type: parameter
Default: true	Data type: boolean
Valid: true, false	
Description: Enable/Disable inline debugger instantiation.	
Name: enable-press-enter-at-exit	Type: parameter
Default: false	Data type: boolean
Valid: true, false	
Description: Enable/Disable pressing key enter at exit.	
Name: kernel_logger.file	Type: parameter
Default: false	Data type: boolean
Valid: true, false	
Description: Keep logger output in a file.	
Name: kernel_logger.filename	Type: parameter
Default: logger_output.txt	Data type: string
Description: Filename to keep logger output _(the option file must be activated).	
Name: kernel_logger.std_err	Type: parameter
Default: true	Data type: boolean
Valid: true, false	
Description: Show logger output through the standard error output.	
Name: kernel_logger.std_err_color	Type: parameter
Default: false	Data type: boolean
Valid: true, false	
Description: Colorize logger output through the standard error output _(only works if std_err is active).	
Name: kernel_logger.std_out	Type: parameter
Default: false	Data type: boolean
Valid: true, false	
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).	
avr32-t2h-syscalls	
Name: avr32-t2h-syscalls.verbose- ↔all Default: false Valid: true, false	Type: parameter Data type: boolean
Description: globally enable/disable verbosity.	
Name: avr32-t2h-syscalls.verbose- ↔syscalls Default: false Valid: true, false	Type: parameter Data type: boolean
Description: enable/disable verbosity while system calls.	
Name: avr32-t2h-syscalls.verbose- ↔setup Default: false Valid: true, false	Type: parameter Data type: boolean
Description: enable/disable verbosity while setup.	

Name: avr32-t2h-syscalls.argc Default: 0	Type: parameter Data type: signed 32-bit integer
Description: Number of program arguments.	
Name: avr32-t2h-syscalls.stdin-pipe- ↔filename Default:	Type: parameter Data type: string
Description: stdin pipe filename.	
Name: avr32-t2h-syscalls.stdout- ↔pipe-filename Default:	Type: parameter Data type: string
Description: stdout pipe filename.	
Name: avr32-t2h-syscalls.stderr- ↔pipe-filename Default:	Type: parameter Data type: string
Description: stderr pipe filename.	
cpu	
Name: cpu.max-inst Default: 18446744073709551615	Type: parameter Data type: unsigned 64-bit integer
Description: maximum number of instructions to simulate.	
Name: cpu.verbose-all Default: false Valid: true, false	Type: parameter Data type: boolean
Description: globally enable/disable verbosity.	
Name: cpu.verbose-setup Default: false Valid: true, false	Type: parameter Data type: boolean
Description: enable/disable verbosity while setup.	
Name: cpu.verbose-interrupt	Type: parameter

Default: false Valid: true, false Description: enable/disable verbosity when handling interrupts.	Data type: boolean
Name: cpu.verbose-step Default: false Valid: true, false Description: enable/disable verbosity when stepping instructions.	Type: parameter Data type: boolean
Name: cpu.trap-on-instruction-counter Default: 18446744073709551615 Description: number of simulated instruction before trapping.	Type: parameter Data type: unsigned 64-bit integer
Name: cpu.enable-trap-on-exception Default: false Valid: true, false Description: enable/disable trap reporting on exception.	Type: parameter Data type: boolean
Name: cpu.halt-on Default: Description: Symbol or address where to stop simulation.	Type: parameter Data type: string
Name: cpu.cpu-cycle-time Default: 14999 ps Description: CPU cycle time.	Type: parameter Data type: sc_time
Name: cpu.hsb-cycle-time Default: 14999 ps Description: HSB cycle time.	Type: parameter Data type: sc_time
Name: cpu.nice-time Default: 1 ms Description: maximum time between synchronizations.	Type: parameter Data type: sc_time

Name: <code>cpu.ipc</code> Default: 1	Type: parameter Data type: double precision floating-point
Description: maximum instructions per cycle (should be ≤ 1.0).	
Name: <code>cpu.enable-host-idle</code> Default: false Valid: true, false	Type: parameter Data type: boolean
Description: Enable/Disable host idle periods when target is idle.	
Name: <code>cpu.enable-dmi</code> Default: true Valid: true, false	Type: parameter Data type: boolean
Description: Enable/Disable TLM 2.0 DMI (Direct Memory Access) to speed-up simulation.	
Name: <code>cpu.debug-dmi</code> Default: false Valid: true, false	Type: parameter Data type: boolean
Description: Enable/Disable debugging of DMI (Direct Memory Access).	
debugger	
Name: <code>debugger.verbose</code> Default: false Valid: true, false	Type: parameter Data type: boolean
Description: Enable/Disable verbosity.	
Name: <code>debugger.dwarf-to-html-output- ↔directory</code> Default:	Type: parameter Data type: string
Description: DWARF v2/v3 to HTML output directory.	
Name: <code>debugger.dwarf-register-number- ↔mapping-filename</code> Default: <code>avr32_dwarf_register_number_ ↔mapping.xml</code>	Type: parameter Data type: string
Description: DWARF register number mapping filename.	

Name: debugger.parse-dwarf Default: true Valid: true, false	Type: parameter Data type: boolean
Description: Enable/Disable parsing of DWARF debugging informations.	
Name: debugger.debug-dwarf Default: false Valid: true, false	Type: parameter Data type: boolean
Description: Enable/Disable debugging of DWARF.	
gdb-server	
Name: gdb-server.memory-atom-size Default: 0x00000001	Type: parameter Data type: unsigned 32-bit integer
Description: size of the smallest addressable element in memory.	
Name: gdb-server.tcp-port Default: 0	Type: parameter Data type: signed 32-bit integer
Description: TCP/IP port to listen waiting for a GDB client connection.	
Name: gdb-server.architecture-description ↔filename Default: gdb_avr32.xml	Type: parameter Data type: string
Description: filename of a XML description of the connected processor.	
Name: gdb-server.verbose Default: false Valid: true, false	Type: parameter Data type: boolean
Description: Enable/Disable verbosity.	
inline-debugger	
Name: inline-debugger.memory-atom-size ↔size Default: 0x00000001	Type: parameter 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	Type: parameter
Default:	Data type: string
Description: path to initial macro to run when debugger starts.	
Name: inline-debugger.output	Type: parameter
Default:	Data type: string
Description: path to output file where to redirect the debugger outputs.	
irq-stub0	
Name: irq-stub0.enable	Type: parameter
Default: true	Data type: boolean
Valid: true, false	
Description: Enable/Disable a lazy implementation of TLM 2.0 method interface.	
Name: irq-stub0.verbose	Type: parameter
Default: false	Data type: boolean
Valid: true, false	
Description: Enable/Disable verbosity.	
irq-stub1	
Name: irq-stub1.enable	Type: parameter
Default: true	Data type: boolean
Valid: true, false	
Description: Enable/Disable a lazy implementation of TLM 2.0 method interface.	
Name: irq-stub1.verbose	Type: parameter
Default: false	Data type: boolean
Valid: true, false	
Description: Enable/Disable verbosity.	
irq-stub2	
Name: irq-stub2.enable	Type: parameter
Default: true	Data type: boolean
Valid: true, false	

Description: Enable/Disable a lazy implementation of TLM 2.0 method interface.	
Name: irq-stub2.verbose	Type: parameter
Default: false	Data type: boolean
Valid: true, false	
Description: Enable/Disable verbosity.	
irq-stub3	
Name: irq-stub3.enable	Type: parameter
Default: true	Data type: boolean
Valid: true, false	
Description: Enable/Disable a lazy implementation of TLM 2.0 method interface.	
Name: irq-stub3.verbose	Type: parameter
Default: false	Data type: boolean
Valid: true, false	
Description: Enable/Disable verbosity.	
loader	
Name: loader.verbose	Type: parameter
Default: false	Data type: boolean
Valid: true, false	
Description: Enable/Disable verbosity.	
Name: loader.verbose-parser	Type: parameter
Default: false	Data type: boolean
Valid: true, false	
Description: Enable/Disable verbosity of parser.	
Name: loader.filename	Type: parameter
Default:	Data type: string
Description: List of files to load. Syntax: [[filename=]<filename1>[:[format=]<format1>]], [filename=]<filename2>[:[form	
(e.g. boot.bin:raw,app.elf).	
loader.memory-mapper	
Name: loader.memory-mapper.verbose	Type: parameter
Default: false	Data type: boolean
Valid: true, false	

Description: Enable/Disable verbosity.	
Name: loader.memory-mapper.verbose- ↔parser	Type: parameter
Default: false	Data type: boolean
Valid: true, false	
Description: Enable/Disable verbosity of parser.	
Name: loader.memory-mapper.mapping	Type: parameter
Default: ram:0x0-0xffffffff	Data type: string
Description: Memory mapping. Syntax: [[(memory=]<memory1>[:[range=]<low1-high1>]],[(memory=]<memory2>[:[range=]<low2-high2>]]... (e.g. ram:0x0-0x00ffff,rom:0xff0000-0xffff).	
memory-router	
Name: memory-router.cycle_time	Type: parameter
Default: 14999 ps	Data type: sc_time
Description: Time to process a request/response by the router.	
Name: memory-router.port_buffer_ ↔size	Type: parameter
Default: 0	Data type: unsigned 32-bit integer
Description: Defines the size of the buffer for incoming requests in each of the input ports (0 = infinite).	
Name: memory-router.mapping_0	Type: parameter
Default: range_start="0x0" range_ ↔end="0xffffffff" output_ ↔port="0" translation="0x0"	Data type: unisim::component::t1m2::interconnect: ↔:generic_router::Mapping
Description: Defined a mapping of the router with format "[range_start]", "[range_end]", "[outport_index]" where [range_start], [range_end] and [outport_index] are to be replaced with the initial address, end address (= range_start + range_size - 1) and the output port index respectively.	
nmireq-stub	
Name: nmireq-stub.enable	Type: parameter
Default: true	Data type: boolean
Valid: true, false	

Description: Enable/Disable a lazy implementation of TLM 2.0 method interface.	
Name: nmireq-stub.verbose	Type: parameter
Default: false	Data type: boolean
Valid: true, false	
Description: Enable/Disable verbosity.	
profiler	
Name: profiler.min-data-read-prof- ↔addr	Type: parameter
Default: 0x00000000	Data type: unsigned 32-bit integer
Description: Minimum address for data read profiling.	
Name: profiler.max-data-read-prof- ↔addr	Type: parameter
Default: 0xffffffff	Data type: unsigned 32-bit integer
Description: Maximum address for data read profiling.	
Name: profiler.min-data-write-prof- ↔addr	Type: parameter
Default: 0x00000000	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 Default: false Valid: true, false Description: Enable/Disable verbosity.	Type: parameter Data type: boolean
ram	
Name: ram.org Default: 0x00000000 Description: memory origin/base address.	Type: parameter Data type: unsigned 32-bit integer
Name: ram.bytesize Default: 0 Description: memory size in bytes.	Type: parameter Data type: unsigned 32-bit integer
Name: ram.initial-byte-value Default: 0x00 Description: memory cycle time.	Type: parameter Data type: unsigned 8-bit integer
Name: ram.cycle-time Default: 14999 ps Description: memory read latency.	Type: parameter Data type: sc_time
Name: ram.read-latency Default: 14999 ps Description: memory write latency.	Type: parameter Data type: sc_time
Name: ram.write-latency Default: 0 s Description: enable/disable verbosity.	Type: parameter Data type: sc_time
Name: ram.verbose Default: false Valid: true, false Description: enable/disable verbosity.	Type: parameter Data type: boolean

1.6 Statistics

Simulation statistic counters are listed below:

cpu	
Name: <code>cpu.instruction-counter</code>	Type: statistic Data type: unsigned 64-bit integer
Description: number of simulated instructions.	
Name: <code>cpu.run-time</code>	Type: statistic Data type: <code>sc_time</code>
Description: run time.	
Name: <code>cpu.idle-time</code>	Type: statistic Data type: <code>sc_time</code>
Description: idle time.	
ram	
Name: <code>ram.memory-usage</code>	Type: statistic Data type: unsigned 32-bit integer
Description: target memory usage in bytes (page granularity of 1048576 bytes).	
Name: <code>ram.read-counter</code>	Type: statistic Data type: unsigned 64-bit integer
Description: read access counter (not accurate when using SystemC TLM 2.0 DMI).	
Name: <code>ram.write-counter</code>	Type: statistic Data type: unsigned 64-bit integer
Description: write access counter (not accurate when using SystemC TLM 2.0 DMI).	

1.7 Formulas

Simulation statistic formulas are listed below:

cpu	
Name: <code>cpu.idle-rate</code>	Type: formula
Formula: <code>cpu.idle-time / cpu.run-time</code>	Data type: double precision floating-point

Description:

idle rate.

Name: `cpu.load-rate`**Type:** `formula`**Formula:** `1 - cpu.idle-time / cpu.`**Data type:**`double precision``↔run-time``floating-point`**Description:**

load rate.