SCR1 User Manual

Syntacore, info@syntacore.com

v1.0.0, 2018-05-07

Table of Contents

Revision history

Revision	Date	Description
1.0.0	2018-05-07	Initial version

1. SCR1 overview

SCR1 is an open-source RISC-V compatible MCU core, designed by Syntacore.

1.1. Version of SCR1 Core

The version of SCR1 core corresponds to MIMPID value of 0x18050700.

1.2. Features

- RV32I|E[MC] ISA

- Machine privilege mode

- 2 to 4 stage pipeline

- 32-bit AXI4/AHB-Lite external interface

- Integrated IRQ controller and advanced debug

- Optimized for area and power

- Written in SystemVerilog

- Features a number of configurable parameters

1.3. Block Diagram

The core is load-store architecture, where only load and store instructions access memory and arithmetic instructions only operate on integer registers. The core provides a 32-bit user address space that is byte-addressed and little-endian. The execution environment will define what portions of the address space are legal to access.

Block diagram of the core is shown in Figure 1.
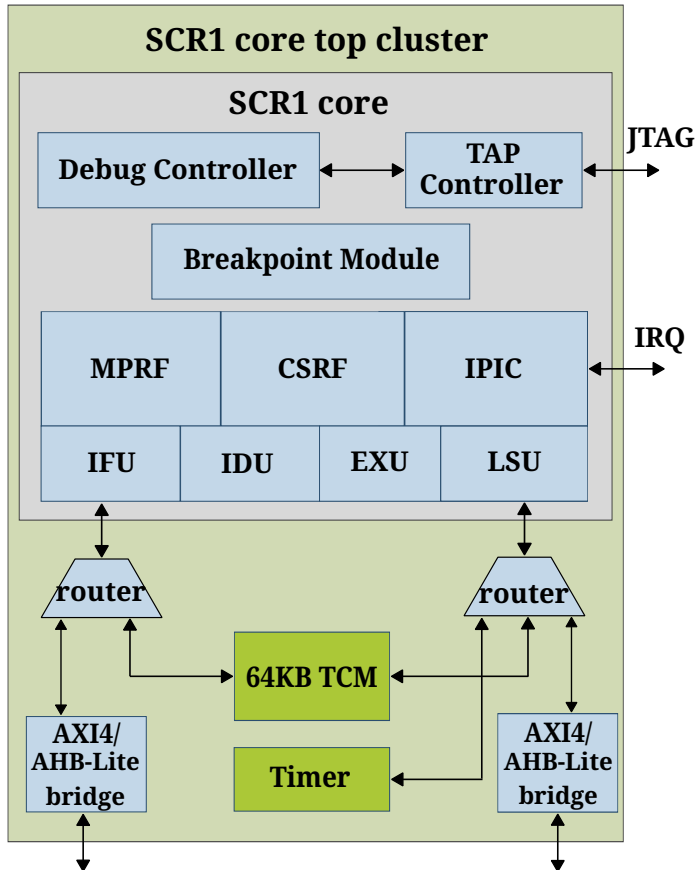
Figure 1: SCR1 Block Diagram

SCR1 core contains:

- Instruction Fetch Unit (IFU)

- Instruction Decode Unit (IDU)

- Execution Unit (incl. integer ALU) (EXU, IALU)

- Load-Store Unit (LSU)

- Multi-port register file (MPRF)

- Control/Status register file (CSRF)

- Integrated programmable interrupt controller (IPIC)

- Hardware Breakpoint Module (BRKM)

- Tightly-coupled memory (TCM)

- External AXI4/AHB-Lite instruction memory interface

- External AXI4/AHB-Lite data memory interface
- Debug Subsystem:
    - Test access point controller (TAPC)
    - Debug Controller (DBGC)

2. Codebase overview

Table 1: Repository contents

Folder	Description
docs	SCR1 documentation
src	SCR1 RTL source and testbench files
tests/common	Common source files for tests
tests/riscv_isa	Common source files for RISC-V ISA tests
tests/benchmarks/dhrystone21	Dhrystone 2.1 source files
tests/vectored_isr_sample	Simple test example for vectored interrupt mode

Table 2: SCR1 RTL source and testbench files

Path	Description
SCR1 header files	
includes/scr1_ahb.svh	AHB header file
includes/scr1_arch_description.svh	Architecture description file
includes/scr1_arch_types.svh	Pipeline types description file
includes/scr1_brkm.svh	BRKM header file
includes/scr1_csr.svh	CSR mapping/description file
includes/scr1_dbgc.svh	DBGC header file
includes/scr1_ipic.svh	IPIC header file
includes/scr1_memif.svh	Memory interface definitions file
includes/scr1_riscv_isa_decoding.svh	RISC-V ISA definitions file
includes/scr1_search_ms1.svh	Most significant one search function
includes/scr1_tapc.svh	TAPC header file
SCR1 pipeline source files	
pipeline/scr1_brkm_matcher.sv	Matcher of the BRKM
pipeline/scr1_ipic.sv	Integrated Programmable Interrupt Controller (IPIC)
pipeline/scr1_pipe_brkm.sv	Hardware Breakpoint Module (BRKM)
pipeline/scr1_pipe_csr.sv	Control Status Registers (CSR)
pipeline/scr1_pipe_dbga.sv	Debug Agent (DBGA)
pipeline/scr1_pipe_exu.sv	Execution Unit (EXU)
pipeline/scr1_pipe_ialu.sv	Integer Arithmetic Logic Unit (IALU)
pipeline/scr1_pipe_idu.sv	Instruction Decoder Unit (IDU)
pipeline/scr1_pipe_ifu.sv	Instruction Fetch Unit (IFU)
pipeline/scr1_pipe_lsu.sv	Load/Store Unit (LSU)
pipeline/scr1_pipe_mprf.sv	Multi Port Register File (MPRF)

Path	Description
pipeline/scr1_pipe_top.sv	SCR1 pipeline top
pipeline/scr1_tracelog.sv	Core tracelog module
SCR1 top source files	
core/primitives/scr1_cg.sv	SCR1 clock gate primitive
core/scr1_clk_ctrl.sv	SCR1 clock control
core/scr1_core_top.sv	SCR1 core top
core/scr1_dbgc.sv	Debug Controller (DBGC)
core/scr1_sync_rstn.sv	Synchronizer for rst_n with test_mode
core/scr1_tapc.sv	TAP Controller (TAPC)
core/scr1_tapc_data_reg.sv	TAPC data register
core/scr1_tapc_shift_reg.sv	TAPC shift register
core/scr1_tapc_synchronizer.sv	TAPC clock domain crossing synchronizer
SCR1 top cluster source files	
top/scr1_dmem_ahb.sv	Data memory AHB bridge
top/scr1_dmem_router.sv	Data memory router
top/scr1_dp_memory.sv	Dual-port synchronous memory with byte enable inputs
top/scr1_imem_ahb.sv	Instruction memory AHB bridge
top/scr1_imem_router.sv	Instruction memory router
top/scr1_mem_axi.sv	Memory AXI bridge
top/scr1_tcm.sv	Tightly-Coupled Memory (TCM)
top/scr1_timer.sv	Memory-mapped Timer
top/scr1_top_ahb.sv	SCR1 AHB top
top/scr1_top_axi.sv	SCR1 AXI top
Testbench files	
tb/scr1_memory_tb_ahb.sv	AHB memory testbench
tb/scr1_memory_tb_axi.sv	AXI memory testbench
tb/scr1_top_tb_ahb.sv	SCR1 top testbench AHB
tb/scr1_top_tb_axi.sv	SCR1 top testbench AXI

3. Recommended configurations

The table below shows recommended SCR1 configurations for typical use cases. These configurations can be easily enabled in **scr1_arch_description.svh** file, section "Recommended configurations".

Table 3: SCR1 recommended configurations

Architecture	RV32EC	RV32IC	RV32IMC	RV32IMC
Pipeline stages	3	2	2	2
GPRs	16	32	32	32
Hardware multiplier	-	-	+	+
Fast multiplier	-	-	-	+
Compressed instructions	+	+	+	+
IRQ lines	1	16	1	16
Debug	-	+	-	+
HW breakpoints	0	2	0	4
Coremark/MHz	1.00	1.27	1.60	2.95
DMIPS/MHz	1.54	1.90	1.89	1.89
Area, 100MHz @90nm_LP, kgates	10.8	24.4	18.7	33.8
Max frequency @90nm_LP, MHz	400	340	255	214
Config option	SCR1_CFG_RV32EC	SCR1_CFG_RV32IC	SCR1_CFG_RV32IMC_SLOW	SCR1_CFG_RV32IMC_FAST

4. Configurable options

SCR1 has a total of 33 configurable options, described below.

Table 4: SCR1 configurable options

Name	Description
Architectural options	
SCR1_RVE_EXT	Enable RV32E base integer instruction set; when this option is disabled, RV32I base is used
SCR1_RVM_EXT	Enable M extension (hardware multiplication and division)
SCR1_RVC_EXT	Enable C extension
SCR1_DBGC_EN	Enable debug controller
SCR1_BRKM_EN	Enable breakpoint controller
SCR1_BRKM_BRKPT_NUMBER	Number of hardware breakpoints
SCR1_IPIC_EN	Enable interrupt controller
SCR1_VECT_IRQ_EN	Enable vectored mode (see MTVEC [0x305] in **docs/scr1_eas.pdf**)
SCR1_CSR_MCOUNTEN_EN	Enable counter control CSR (see MCOUNTEN [0x7E0] in **docs/scr1_eas.pdf**)
SCR1_CSR_MTVEC_BASE_RW_BITS	Number of writable bits in MTVEC BASE field (see MTVEC [0x305] in **docs/scr1_eas.pdf**)
SCR1_ARCH_RST_VECTOR	User-defined reset vector
SCR1_ARCH_CSR_MTVEC_BASE_RST_VAL	MTVEC BASE field reset value, or constant value for MTVEC BASE bits that are hardwired
Core options	
SCR1_IFU_QUEUE_BYPASS	Pipeline bypass after IFU (see "Pipeline configurations" in **docs/scr1_eas.pdf**)
SCR1_EXU_STAGE_BYPASS	Pipeline bypass before EXU (see "Pipeline configurations" in **docs/scr1_eas.pdf**)
SCR1_FAST_MUL	Enable fast one-cycle multiplication; when this option is disabled, multiplication takes 32 cycles
SCR1_CLKCTRL_EN	Enable global clock gating; please note that for synthesis, code in **scr1_cg.sv** should be replaced with implementation-specific clock gate cod
SCR1_IPIC_SYNC_EN	Enable 2-stage input synchronizer for IRQ lines
Uncore options	
SCR1_TCM_EN	Enable tightly-coupled memory, default size is 64K
SCR1_TCM_ADDR_MASK	Set TCM mask and size; size in bytes is two's complement of the mask value
SCR1_TCM_ADDR_PATTERN	Set TCM address match pattern

Name	Description
SCR1_TIMER_ADDR_MASK	Set timer mask (should be 0xFFFFFFE0)
SCR1_TIMER_ADDR_PATTERN	Set timer address match pattern
SCR1_IMEM_AHB_IN_BP	Enable bypass on instruction memory AHB bridge inputs
SCR1_IMEM_AHB_OUT_BP	Enable bypass on instruction memory AHB bridge outputs
SCR1_DMEM_AHB_IN_BP	Enable bypass on data memory AHB bridge inputs
SCR1_DMEM_AHB_OUT_BP	Enable bypass on data memory AHB bridge outputs
SCR1_IMEM_AXI_REQ_BP	Enable bypass on instruction memory AXI bridge request
SCR1_IMEM_AXI_RESP_BP	Enable bypass on instruction memory AXI bridge response
SCR1_DMEM_AXI_REQ_BP	Enable bypass on data memory AXI bridge request
SCR1_DMEM_AXI_RESP_BP	Enable bypass on data memory AXI bridge response
Simulation options	
SCR1_SIM_ENV	Enable simulation code: SVA, trace log (see Simulation code)
SCR1_TRACE_LOG_EN	Enable trace log (see Trace log)
SCR1_TRACE_LOG_FULL	Enable full trace log (see Trace log)

5. Simulation environment

5.1. Prerequisites

RISC-V GCC toolchain is required to compile the software. You can use pre-built binaries or build the toolchain from scratch.

5.1.1. Using pre-built binary tools

Pre-built RISC-V GCC toolchain and OpenOCD binaries are available to download from http://syntacore.com/page/products/sw-tools. Download the archive (**.tar.gz** for Linux, **.zip** for Windows) for your platform, extract the archive to your preferred directory and update the PATH environment variable as described in **Set environment variables** section.

5.1.2. Building tools from source

You can build the RISC-V toolchain from sources.

Build procedure is verified at the Ubuntu 14.04 LTS and Ubuntu 16.04 LTS distributions.

```
sudo apt-get install autoconf automake libmpc-dev libmpfr-dev libgmp-dev gawk bison
flex texinfo libtool make g++ pkg-config libexpat1-dev zlib1g-dev
git clone https://github.com/riscv/riscv-gnu-toolchain.git
cd riscv-gnu-toolchain
git checkout a71fc539850f8dacf232fc580743b946c376014b
git submodule update --init --recursive
./configure --prefix=<YOUR_INSTALL_PATH> --enable-multilib
make
```

More detailed instructions on how to prepare and build the toolchain can be found in https://github.com/riscv/riscv-tools/blob/master/README.md.

5.1.3. Set environment variables

Add the <YOUR_INSTALL_PATH>/bin folder to the PATH environment variable:

```
export PATH=$PATH:<YOUR_INSTALL_PATH>/bin
```

5.2. Clone and prepare the RISC-V ISA tests

```
git clone https://github.com/riscv/riscv-tests
cd riscv-tests
git checkout a9433c4daa287fbe101025f2a079261a10149225
```

Set the $RISCV_TESTS environment variable accordingly:

```
export RISCV_TESTS=<PATH TO RISCV-TESTS>
```

5.3. Build RTL, compile and run tests

`make run_<SIMULATOR> BUS=<AHB, AXI>` will build RTL and tests, then run all tests with default parameters.

Currently supported options:

- run_modelsim
- run_vcs
- run_ncsim

Please note that RTL simulator executables should be in your PATH variable.

To run an arbitrary subset of tests, edit the **tests** target in the top Makefile, or the **rv32_tests.inc** in riscv_isa subfolder. After all the tests have finished, the results can be found in **build/test_results.txt** (default location).

- Test build and run parameters can be configured in the **Makefile**
- SCR1 core parameters can be configured in **src/includes/scr1_arch_description.svh**

Please make sure that architectural config selected for the SCR1 RTL matches the one used for tests compilation.

5.4. Simulation code

It is possible to add useful information about the simulation process: assertions and trace log. Parameter SCR1_SIM_ENV must be defined in src/includes/scr1_arch_description.svh to enable simulation code (disabled in default core configuration).

5.4.1. Trace log

During the simulation, the data from General-purpose Integer Registers and Control and Status Registers will be written to a special files in build directory.

- File trace_mprf_<HARTID>.log contains full trace log: time, delay, PC, values of all GPRs. Available if SCR1_TRACE_LOG_FULL is defined in src/includes/scr1_arch_description.svh (enabled by default).
- File trace_mprf_diff_<HARTID>.log contains compact trace log which only includes GPR value changes: time, PC, value of changed GPR. Available if SCR1_TRACE_LOG_FULL is not defined in src/includes/scr1_arch_description.svh.
- File trace_csr_<HARTID>.log contains trace log for each change in CSRs: time, MSTATUS, MTVEC, MIE, MIP, MEPC, MCAUSE, MTVAL.

Parameter SCR1_TRACE_LOG_EN and SCR1_SIM_ENV must be defined in src/includes/scr1_arch_description.svh to enable trace log.

6. SDK information

SCR1 SDKs are located in https://github.com/syntacore/scr1-sdk. The table below gives some basic information on SDKs.

Table 5: SCR1 SDKs

SDK name	Digilent Arty	Terasic DE10-Lite	Arria V GX Starter
Default architecture	RV32IMC	RV32IMC	RV32IMC
FPGA vendor	Xilinx	Altera	Altera
FPGA part	XC7A35T	10M50DAF484C7G	5AGXFB3H4F35C4N
Required software	Vivado	Quartus	Quartus
Frequency, MHz	25	20	30
Resources	TCM, SRAM, UART	TCM, SDRAM, UART	TCM, DDR3, UART
User Guide link	https://github.com/syntacore/scr1-sdk/blob/master/docs/arty_scr1_guide_en.pdf	https://github.com/syntacore/scr1-sdk/blob/master/docs/de10lite_scr1_guide_en.pdf	https://github.com/syntacore/scr1-sdk/blob/master/docs/a5_scr1_guide_en.pdf

7. Support

For more information on SCR1 core, please write to scr1@syntacore.com.