Sapphire RISC-V SoC Data Sheet

DS-SAPPHIRE-v3.0
August 2022
www.elitestek.com

Contents

Introduction

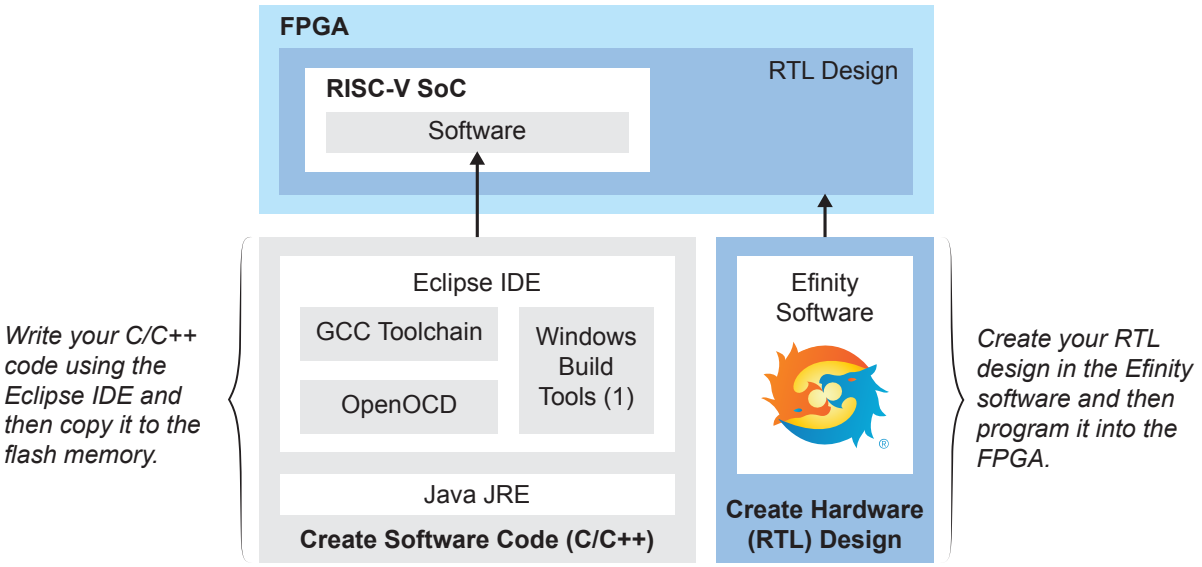
易灵思 provides the heavy-weightlight-weightultra-lightweightconfigurable, cached soft RISC-V SoC, RubyJadeOpalSapphire, which optionally includes a memory controller interface. The Sapphire SoC supports a variety of peripherals. You can choose which peripherals you want by configuring the SoC in the Efinity® IP Manager. This core is similar to the open-source SaxonSOC, but it has been optimized for 钛金系列 and Trion® FPGAs.

Important: You should use the Sapphire SoC for all new designs; the Ruby, Jade, and Opal SoCs are end of life with the Efinity® software v2022.1.

Table 1: SoC Version Compatibility

Efinity Version	SoC Version	Notes
2021.1 (and all patches)	1.x	Initial version with limited feature set. You can continue to use this version with the Efinity software v2021.1.
2021.2 and higher (and all patches)	2.0 and higher	Enhanced version with additional features, such as custom instructions, floating point unit, Linux memory managemet unit, optional RISC-V extensions (atomic and compressed), and up to 3 user timers. This version is not backwards compatible with v1.x. Use this version for all new designs.

Figure 1: Sapphire RISC-V SoC Design Flow



1. Windows build tools required on Windows platforms only.

Learn more: For details on developing RTL designs or creating software, refer to the **Sapphire RISC-V Hardware and Software User Guide**.

VexRiscv RISC-V Core

The Sapphire SoC is based on the VexRiscv core created by Charles Papon. The VexRiscv core is a 32-bit CPU using the ISA RISCV32I with M, A, F, D, and C extensions, has six pipeline stages (fetch, injector, decode, execute, memory, and writeback), and a configurable feature set.

In the Sapphire SoC, the VexRiscv core is user configurable, and can support AXI4 and APB3 bus interfaces and instruction and data caches.

The VexRiscv core won first place in the RISC-V SoftCPU contest in 2018.[1]

Features

- 1 - 4 (user selectable) VexRiscv processor(s) with 6 pipeline stages (fetch, injector, decode, execute, memory, and write back), interrupts and exception handling with machine mode
- 20 - 400 MHz system clock frequency
- 1 - 512 KB on-chip RAM with boot loader for SPI flash
- Memory controller for DDR or HyperRAM memories
  - Supports memory module sizes from 4 MB to 3.5 GB
  - User-configurable external memory bus frequency
  - 1 half duplex AXI3 interface or 1 full duplex AXI4 interface (up to 512-bits) to communicate with the external memory
  - 400 MHz DDR clock frequency, 800 Mbps
  - 200 MHz HyperRAM clock frequency, 400 Mbps
- Up to 2 AXI master channels for user logic, data widths from 32 to 512
- 1 AXI slave channel to user logic
- Includes a floating point unit
- Includes an optional Linux memory management unit
- Includes a custom instruction interface with 1,024 IDs to perform different functions
- Supports optional RISC-V extensions such as atomic and compressed
- APB3 peripherals:
  - Up to 32 GPIOs
  - Up to 3 I^2C masters
  - Clint timer
  - PLIC
  - Up to 3 SPI masters
  - Up to 3 user timers
  - Up to 3 UARTs with 115,200 baud rate
  - Up to 5 slave user peripherals
  - Up to 8 user interrupts

FPGA Support

The Sapphire SoC supports all Trion® FPGAs (except the T4) and all FPGAs, however, you may only be able to use some of the features in certain devices. For example, the DDR controller only works with FPGAs that have a hardened DDR controller block.

[1] https://www.businesswire.com/news/home/20181206005747/en/RISC-V-SoftCPU-Contest-Winners-Demonstrate-Cutting-Edge-RISC-V

Resource Utilization and Performance

The Sapphire is configurable. These tables show the resource usage for various configurations.

Table 2: Cachless SoC with External Memory

FPGA	Logic/ Adders	FlipFlops	Memory Blocks	DSP48 Blocks	f_{MAX} (MHz)	Efinity Version
Ti60 F225 C4 (typical)	7,065	7,563	45	4	317	2022.1
Ti60 F225 C4 (custom instruction)	7,257	7,597	46	4	315	2022.1

Table 3: Cachless SoC without External Memory

FPGA	Logic/ Adders	FlipFlops	Memory Blocks	DSP48 Blocks	f_{MAX} (MHz)	Efinity Version
Ti60 F225 C4 (typical)	4,401	3,055	12	4	317	2022.1
Ti60 F225 C4 (custom instruction)	4,471	3,096	12	4	318	2022.1

Table 4: Cached SoC with External Memory

FPGA	Logic/ Adders	FlipFlops	Memory Blocks	DSP48 Blocks	f_{MAX} (MHz)	Efinity Version
Ti60 F225 C4 (typical)	7,568	8,194	58	4	335	2022.1
Ti60 F225 C4 (custom instruction)	7,732	8,229	58	4	324	2022.1
Ti60 F225 C4 (FPU)	14,627	12,456	79	13	242	2022.1
Ti60 F225 C4 (2 cores)	14,418	13,033	107	8	266	2022.1
Ti60 F225 C4 (3 cores)	18,510	15,362	131	12	251	2022.1
Ti60 F225 C4 (4 cores)	22,740	17,620	154	16	235	2022.1

Table 5: Cached SoC without External Memory

FPGA	Logic/ Adders	FlipFlops	Memory Blocks	DSP48 Blocks	f_{MAX} (MHz)	Efinity Version
Ti60 F225 C4 (typical)	5,015	3,657	24	4	335	2022.1
Ti60 F225 C4 (custom instruction)	5,048	3,691	24	4	332	2022.1
Ti60 F225 C4 (FPU)	11,620	7,889	44	13	230	2022.1
Ti60 F225 C4 (2 cores)	11,551	8,212	64	8	294	2022.1
Ti60 F225 C4 (3 cores)	16,156	10,517	87	12	272	2022.1
Ti60 F225 C4 (4 cores)	20,155	12,759	110	16	266	2022.1

Trion Resource Utilization and Performance

The Sapphire is configurable. These tables show the resource usage for various configurations.

Table 6: Cachless SoC with External Memory

FPGA	Logic Utilization (LUTs)	Memory Blocks	f_{MAX} (MHz)	Efinity Version
T120 F576 (typical)	11,245	50	108	2022.1
T120 F576 (custom instruction)	11,521	50	101	2022.1

Table 7: Cachless SoC without External Memory

FPGA	Logic Utilization (LUTs)	Memory Blocks	f_{MAX} (MHz)	Efinity Version
T120 F576 (typical)	5,995	16	107	2022.1
T120 F576 (custom instruction)	6,194	16	111	2022.1

Table 8: Cached SoC with External Memory

FPGA	Logic Utilization (LUTs)	Memory Blocks	f_{MAX} (MHz)	Efinity Version
T120 F576 (typical)	12,154	69	110	2022.1
T120 F576 (custom instruction)	12,310	69	115	2022.1
T120 F576 (FPU)	21,487	82	92	2022.1
T120 F576 (2 cores)	21,250	113	94	2022.1
T120 F576 (3 cores)	27,218	140	89	2022.1
T120 F576 (4 cores)	32,272	166	84	2022.1

Table 9: Cached SoC without External Memory

FPGA	Logic Utilization (LUTs)	Memory Blocks	f_{MAX} (MHz)	Efinity Version
T120 F576 (typical)	6,914	35	113	2022.1
T120 F576 (custom instruction)	7,050	35	109	2022.1
T120 F576 (FPU)	16,287	47	87	2022.1
T120 F576 (2 cores)	15,687	70	96	2022.1
T120 F576 (3 cores)	21,565	96	98	2022.1
T120 F576 (4 cores)	26,816	122	93	2022.1

Functional Description

The RubyJadeOpalSapphire SoC incorporates 1 to 4 32-bit RISC-V processors that have an instruction cache with up to 8 ways and a configurable size of 4 KB4 KB4 KB1 - 32 KB, a data cache with up to 8 ways and a configurable size of 4 KB4 KB4 KB1 - 32 KB, 4 - 512 KB4 - 512 KB4 - 512 KB1 - 512 KB of on-chip RAM, and a variety of peripherals (including 2111 - 5 APB3 slave peripherals and 1 AXI slave). You can configure the operating frequency from 20 - 350 MHz20 - 350 MHz20 - 300 MHz20 - 400 MHz (the actual performance is limited by the design's f_{MAX}). The SoC includes 3211 - 3 I^2C peripherals, 1 - 3 UARTs, 1 - 3 user timers, 1 - 8 user interrupts, and 321 or 21 - 3 SPI masters. The SoC also features a floating-point unit (FPU) and Linux memory management unit (MMU).

The default configuration has up to a 512-bit half-duplex AXI bus to communicate with the 易灵思 DDR controller or HyperRAM controller.

- DDR controller—This core uses the Trion FPGAs hard DDR DRAM interface to reset an external DRAM module (resets and re-initializes the Trion FPGA's DDR interface as well as the DDR module(s)).
- HyperRAM controller—This core controls HyperRAM memory modules.

You can customize the SoC using the IP Manager in the Efinity® software.

Figure 2: RubyJadeOpalSapphire SoC Block Diagram

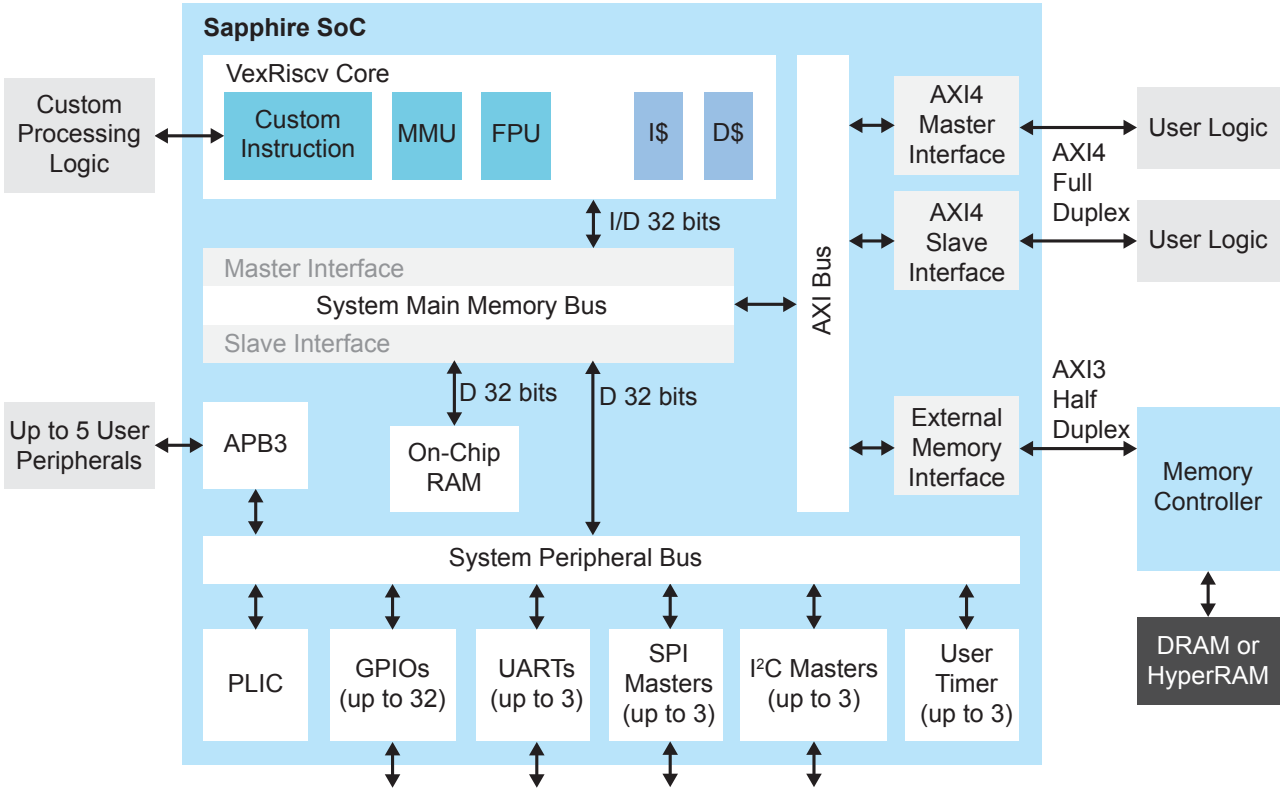
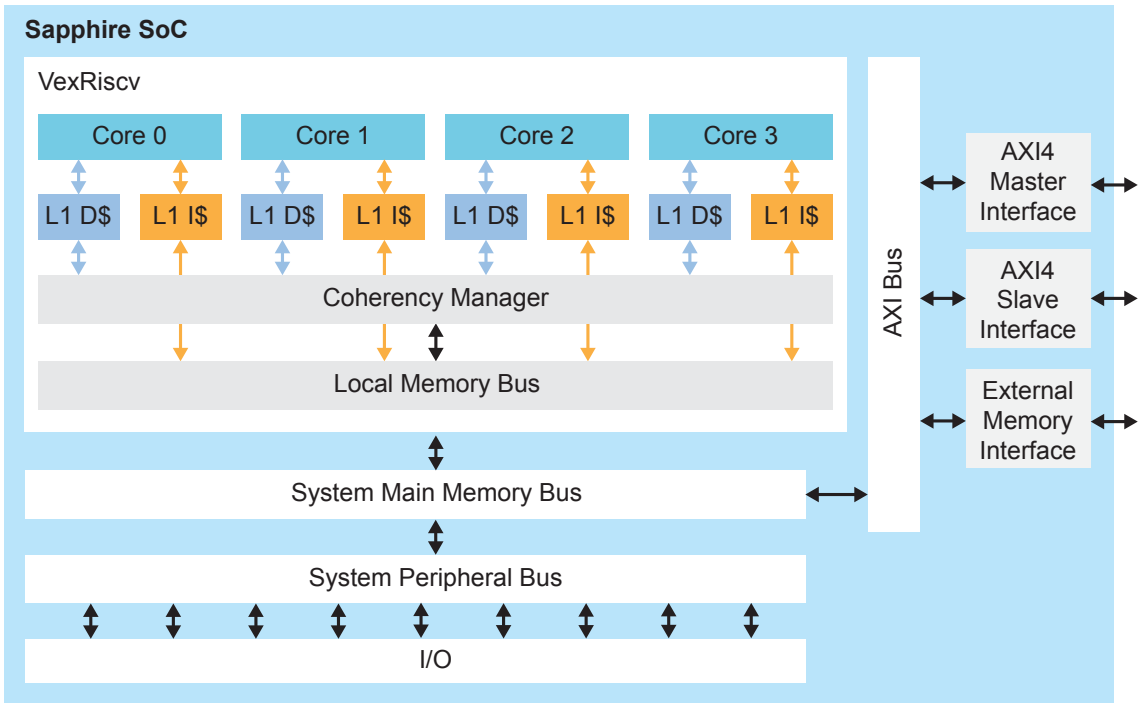


Figure 3: RubyJadeOpalSapphire SoC with Multiple Cores Block Diagram



Address Map

Because the address range might be updated, recommends that you always refer to the parameter name when referencing an address in firmware, not by the actual address. The parameter names and address mappings are defined in **/embedded_sw/**<module>**/bsp/efinix/EfxSapphireSoc/include/soc.h**.

Note: If you need to update the address map, use the IP Configuration wizard to change the addressing and then re-generate the SoC. Using this method keeps the software **soc.h** and FPGA netlist definitions aligned.

Table 10: Default Address Map, Interrupt ID, and Cached Channels

The AXI user slave channel is in a cacheless region (I/O) for compatibility with AXI-Lite.

Device	Parameter	Size	Interrupt ID	Region
Off-chip memory	SYSTEM_DDR_BMB	4 MB to 3.5 GB	–	Cache
GPIO 0	SYSTEM_GPIO_0_IO_CTRL	4 K	[0]: 12 [1]: 13	I/O
GPIO 1	SYSTEM_GPIO_1_IO_CTRL	4 K	[0]: 14 [1]: 15	I/O
I^2C 0	SYSTEM_I2C_0_IO_CTRL	4 K	8	I/O
I^2C 1	SYSTEM_I2C_1_IO_CTRL	4 K	9	I/O
I^2C 2	SYSTEM_I2C_2_IO_CTRL	4 K	10	I/O
Core timer	SYSTEM_CLINT_CTRL	4 K	–	I/O
PLIC	SYSTEM_PLIC_CTRL	4 K	–	I/O
SPI master 0	SYSTEM_SPI_0_IO_CTRL	4 K	4	I/O
SPI master 1	SYSTEM_SPI_1_IO_CTRL	4 K	5	I/O
SPI master 2	SYSTEM_SPI_2_IO_CTRL	4 K	6	I/O
UART 0	SYSTEM_UART_0_IO_CTRL	4 K	1	I/O

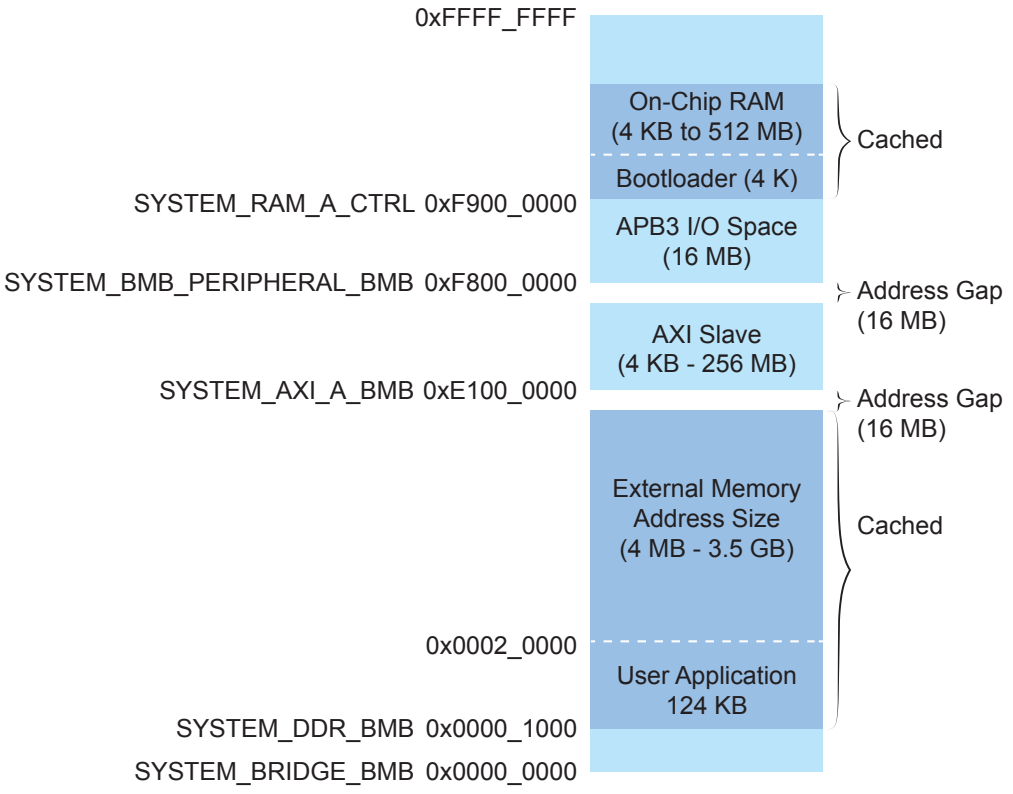
Device	Parameter	Size	Interrupt ID	Region
UART 1	SYSTEM_UART_1_IO_CTRL	4 K	2	I/O
UART 2	SYSTEM_UART_2_IO_CTRL	4 K	3	I/O
User timer 0	SYSTEM_USER_TIMER_0_CTRL	4 K	19	I/O
User timer 1	SYSTEM_USER_TIMER_1_CTRL	4 K	20	I/O
User timer 2	SYSTEM_USER_TIMER_2_CTRL	4 K	21	I/O
User peripheral 0	IO_APB_SLAVE_0_CTRL	4 K to 1 MB	–	I/O
User peripheral 1	IO_APB_SLAVE_1_CTRL	4 K to 1 MB	–	I/O
User peripheral 2	IO_APB_SLAVE_2_CTRL	4 K to 1 MB	–	I/O
User peripheral 3	IO_APB_SLAVE_3_CTRL	4 K to 1 MB	–	I/O
User peripheral 4	IO_APB_SLAVE_4_CTRL	4 K to 1 MB	–	I/O
On-chip BRAM	SYSTEM_RAM_A_BMB	1 - 512 KB	–	Cache
AXI user slave	SYSTEM_AXI_A_BMB	1 K to 256 MB	–	I/O
External interrupt	–	–	[0]: 16 [1]: 17 [2]: 22 [3]: 23 [4]: 24 [5]: 25 [6]: 26 [7]: 27	I/O

For the cached regions, the burst length is equivalent to an AXI burst length of 8. For the I/O region, the burst length is equivalent to an AXI burst length of 1. The AXI user slave is compatible with AXI-Lite by disconnecting unused outputs and driving a constant 1 to the input port.

Note: The RISC-V GCC compiler does not support user address spaces starting at 0x0000_0000.

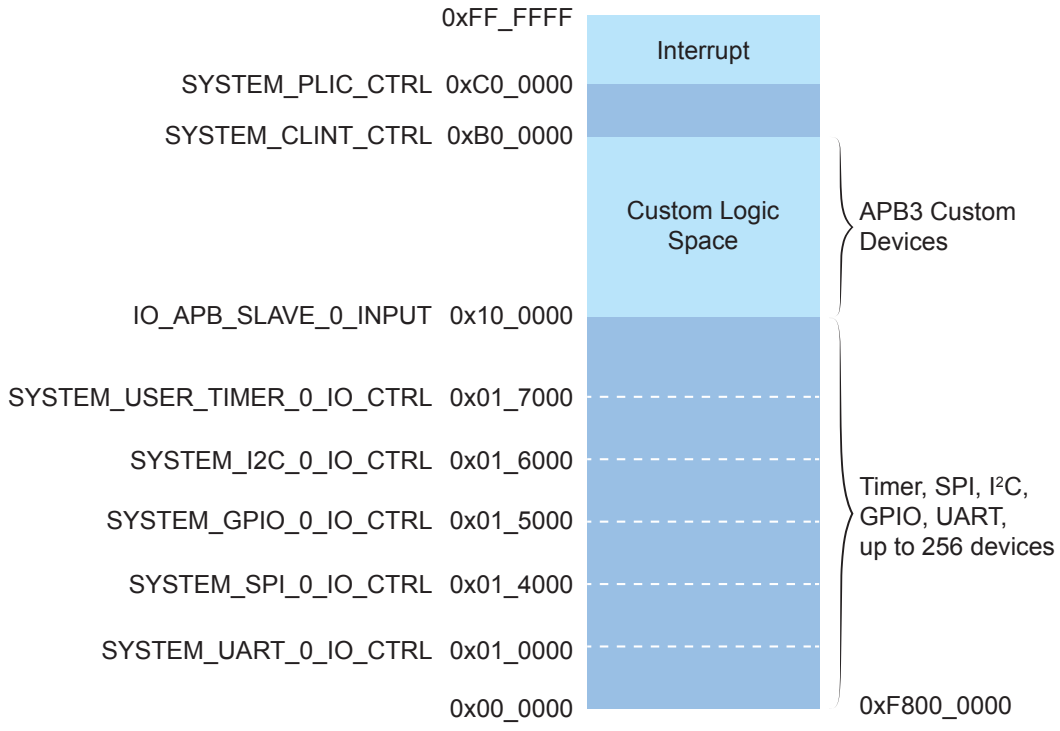
The following figure shows the default address map and the corresponding software parameters for modules in the memory space.

Figure 4: Sapphire Memory Space



The following figure shows the default address map and the corresponding software parameters for I/O.

Figure 5: Sapphire I/O Space



Default I/O address ofset: 0xF800_0000
Total: 16 MB

Flash Address

When the FPGA boots up, the Sapphire SoC copies your binary application file from a SPI flash device to the DDR DRAM module, and then begins execution. The SPI flash binary address starts at 0x0038_0000.

Clocks

Table 11: Clock Ports

Port	Direction	Description
io_systemClock	Input	Provides a 20 - 400 MHz clock for the SoC.
io_peripheralClock	Input	Provides a 20 - 200 MHz clock for the APB3 peripherals and AXI4 slave.
io_memoryClock	Input	Provides a clock for the external memory bus. The frequency is user defined.

Interrupts

Table 12: Interrupt Ports

Port	Direction	Description
userInterruptA userInterruptB userInterruptC userInterruptD userInterruptE userInterruptF userInterruptG userInterruptH	Input	Provides external interrupts.
axiAInterrupt	Input	User AXI slave channel interrupt.

Resets

The Sapphire SoC has as master reset signal, `io_asyncReset` that triggers a system reset. Your RTL design should hold `io_asyncReset` for 10 ns to reset the whole SoC system completely. When you assert `io_asyncReset`, the SoC asserts:

- `io_systemReset`, which resets the RISC-V processor, on-chip memory, and peripherals.
- `io_peripheralReset`, which resets the APB3 peripherals and AXI4 slave.
- `io_memoryReset`, which resets the memory controller, external memory module, I^2C master and slave connected to the memory controller, and any user logic.
- `io_ddrMasters_0_reset`, which responds to the reset for AXI master channel 0 and is synchronized to `io_ddrMasters_0_clk`.
- `io_ddrMasters_1_reset`, which responds to the reset for AXI master channel 1 and is synchronized to `io_ddrMasters_1_clk`.

The SoC asserts the `io_memoryReset`, `io_ddrMaster_0_reset`, and `io_ddrMaster_1_reset` signals at the same time to allow the AXI masters access to the AXI cross bar once the reset completes.

Once `io_systemReset` goes low, the user binary code is executed.

Table 13: Reset Ports

Port	Direction	Description
io_asyncReset	Input	Active-high asynchronous reset for the entire system.
io_systemReset	Output	Synchronous active-high reset for the system clock (io_systemClk).
io_peripheralReset	Output	Synchronous active-high reset for the peripheral clock (io_peripheralClock).
io_memoryReset	Output	External memory reset source from the RISC-V SoC.
io_ddrMasters_0_reset io_ddrMasters_1_reset	Output	Responds to the reset for the AXI master.

AXI Interface

The Sapphire SoC has up to a 512 bit half duplex AXI3 interface or up to a 512 bit full duplex AXI4 interface to communicate to external memory. You configure it on the **Cache/Memory** tab in the IP Configuration wizard.

Additionally it has an optional full duplex AXI4 interface to connect to user logic. You configure it in the **AXI4** tab in the IP Configuration Wizard.

- There is one AXI4 slave interface, which is compatible with AXI-Lite (axlen is always 0.)
- There are two optional full duplex AXI4 master interfaces. You can configure the width as 32, 64, 128, 256, or 512 bits.

Learn more: Refer to the AMBA AXI and ACE Protocol Specification for AXI channel descriptions and handshake information.

AXI Interface to External Memory

Table 14: AXI Slave Full-Duplex Address Channel for Read and Write

Port	Direction	Description
io_ddrA_aw_valid	Output	External memory write address valid.
io_ddrA_aw_ready	Input	External memory write address ready.
io_ddrA_aw_payload_addr[31:0]	Output	External memory write address.
io_ddrA_aw_payload_id[7:0]	Output	External memory write address ID.
io_ddrA_aw_payload_region[3:0]	Output	External memory write region identifier.
io_ddrA_aw_payload_len[7:0]	Output	External memory write burst length.
io_ddrA_aw_payload_size[2:0]	Output	External memory write burst size.
io_ddrA_aw_payload_burst[1:0]	Output	External memory write burst type, INCR only.
io_ddrA_aw_payload_lock	Output	External memory write lock type.
io_ddrA_aw_payload_cache[3:0]	Output	External memory write memory type.
io_ddrA_aw_payload_qos[3:0]	Output	External memory write quality of service.
io_ddrA_aw_payload_prot[2:0]	Output	External memory write protection type.
io_ddrA_ar_valid	Output	External memory read address valid.
io_ddrA_ar_ready	Input	External memory read address ready.
io_ddrA_ar_payload_addr[31:0]	Output	External memory read address.
io_ddrA_ar_payload_id[7:0]	Output	External memory read address ID.
io_ddrA_ar_payload_region[3:0]	Output	External memory read region identifier.

Port	Direction	Description
io_ddrA_ar_payload_len[7:0]	Output	External memory burst length.
io_ddrA_ar_payload_size[2:0]	Output	External memory read burst size.
io_ddrA_ar_payload_burst[1:0]	Output	External memory read burst type, INCR only.
io_ddrA_ar_payload_lock	Output	External memory read lock type.
io_ddrA_ar_payload_cache[3:0]	Output	External memory read memory type.
io_ddrA_ar_payload_qos[3:0]	Output	External memory read quality of service.
io_ddrA_ar_payload_prot[2:0]	Output	External memory read protection type.

Table 15: AXI Slave Half-Duplex Address Channel for Read and Write

Port	Direction	Description
io_ddrA_arw_valid	Output	External memory address valid.
io_ddrA_arw_ready	Input	External memory address ready.
io_ddrA_arw_payload_addr[31:0]	Output	External memory address.
io_ddrA_arw_payload_id[7:0]	Output	External memory address ID.
io_ddrA_arw_payload_region[3:0]	Output	External memory region identifier.
io_ddrA_arw_payload_len[7:0]	Output	External memory burst length.
io_ddrA_arw_payload_size[2:0]	Output	External memory burst size.
io_ddrA_arw_payload_burst[1:0]	Output	External memory burst type, INCR only.
io_ddrA_arw_payload_lock	Output	External memory lock type.
io_ddrA_arw_payload_cache[3:0]	Output	External memory memory type.
io_ddrA_arw_payload_qos[3:0]	Output	External memory quality of service.
io_ddrA_arw_payload_prot[2:0]	Output	External memory protection type.
io_ddrA_arw_payload_write	Output	External memory address read/write selection: 0: Read 1: Write

Table 16: AXI Slave Write Data Channel

Port	Direction	Description
io_ddrA_w_valid	Output	External memory write valid.
io_ddrA_w_ready	Input	External memory write ready.
io_ddrA_w_payload_data[n:0]	Output	External memory write data. n is user configurable up to 256 bits wide.
io_ddrA_w_payload_strb[m:0]	Output	External memory write strobe. m is the width of io_ddrA_w_payload_data[n:0] divided by 8.
io_ddrA_w_payload_last	Output	External memory write last.

Table 17: AXI Slave Write Respond Channel

Port	Direction	Description
io_ddrA_b_valid	Input	External memory write respond valid.
io_ddrA_b_ready	Output	External memory respond ready.
io_ddrA_b_payload_id[7:0]	Input	External memory respond ID.
io_ddrA_b_payload_resp[1:0]	Input	External memory write respond.

Table 18: AXI Slave Read Data Channel

Port	Direction	Description
io_ddrA_r_valid	Input	External memory read valid.
io_ddrA_r_ready	Output	External memory read ready.
io_ddrA_r_payload_data[n:0]	Input	External memory read data. n is user configurable up to 256 bits wide.
io_ddrA_r_payload_id[7:0]	Input	External memory read ID.
io_ddrA_r_payload_resp[1:0]	Input	External memory read respond.
io_ddrA_r_payload_last	Input	External memory read last.

AXI Interface to User Logic

Table 19: User Slave Write Address Channel

Port	Direction	Description
axiA_awvalid	Output	User write address valid.
axiA_awready	Input	User write address ready.
axiA_awaddr[31:0]	Output	User write address.
axiA_awid[7:0]	Output	User write address ID.
axiA_awregion[3:0]	Output	User region identifier.
axiA_awlen[7:0][(2)]	Output	User burst length.
axiA_awsize[2:0]	Output	User burst size.
axiA_awburst[1:0]	Output	User burst type, INCR only.
axiA_awlock	Output	User lock type.
axiA_awcache[3:0]	Output	User memory type.
axiA_awqos[3:0]	Output	User quality of service.
axiA_awprot[2:0]	Output	User protection type.

Table 20: User Slave Write Data Channel

Port	Direction	Description
axiA_wvalid	Output	User write valid.
axiA_wready	Input	User write ready.
axiA_wdata[31:0]	Output	User write data.
axiA_wstrb[3:0]	Output	User write strobe.
axiA_wlast	Output	User write last.

Table 21: User Slave Write Respond Channel

Port	Direction	Description
axiA_bvalid	Input	User write respond valid.
axiA_bready	Output	User respond ready.
axiA_bid[7:0]	Input	User respond ID.
axiA_bresp[1:0]	Input	User write respond.

Table 22: User Slave Read Address Channel

Port	Direction	Description
axiA_arvalid	Output	User read address valid.

(2) axiA_awlen always outputs 0, that is, a burst length of 1. This setting makes the, axiA channel compatible with AXI-Lite.

Port	Direction	Description
axiA_arready	Input	User read address ready.
axiA_araddr[31:0]	Output	User read address.
axiA_arid[7:0]	Output	User read address ID.
axiA_arregion[3:0]	Output	User region identifier.
axiA_arlen[7:0][3]	Output	User burst length.
axiA_arsize[2:0]	Output	User burst size.
axiA_arburst[1:0]	Output	User burst type, INCR only.
axiA_arlock	Output	User lock type.
axiA_arcache[3:0]	Output	User memory type.
axiA_arqos[3:0]	Output	User quality of service.
axiA_arprot[2:0]	Output	User protection type.

Table 23: User Slave Read Data Channel

Port	Direction	Description
axiA_rvalid	Input	User read valid.
axiA_rready	Output	User read ready.
axiA_rdata[31:0]	Input	User read data.
axiA_rid[7:0]	Input	User read ID.
axiA_rresp[1:0]	Input	User read respond.
axiA_rlast	Input	User read last.

Table 24: User Master Clock and Reset

Where n is the channel number.

Port	Direction	Description
io_ddrMasters_n_clk	Input	AXI master clock.
io_ddrMasters_n_reset	Output	AXI master active high reset.

AXI Master Interface

Table 25: User Master Write Address Channel

Where n is the channel number.

Port	Direction	Description
io_ddrMasters_n_aw_valid	Input	User write address valid.
io_ddrMasters_n_aw_ready	Output	User write address ready.
io_ddrMasters_n_aw_payload_addr[31:0]	Input	User write address.
io_ddrMasters_n_aw_payload_id[7:0]	Input	User write address ID.
io_ddrMasters_n_aw_payload_region[3:0]	Input	User region identifier.
io_ddrMasters_n_aw_payload_len[7:0]	Input	User burst length.
io_ddrMasters_n_aw_payload_size[2:0]	Input	User burst size.
io_ddrMasters_n_aw_payload_burst[1:0]	Input	User burst type, INCR only.
io_ddrMasters_n_aw_payload_lock	Input	User lock type.
io_ddrMasters_n_aw_payload_cache[3:0]	Input	User memory type.
io_ddrMasters_n_aw_payload_qos[3:0]	Input	User quality of service.

[3] axiA_arlen always outputs 0, that is, a burst length of 1.This setting makes the, axiA channel compatible with AXI-Lite.

Port	Direction	Description
io_ddrMasters_n_aw_payload_prot[2:0]	Input	User protection type.

Table 26: User Master Write Data Channel

Where n is the channel number.

Port	Direction	Description
io_ddrMasters_n_w_valid	Input	User write valid.
io_ddrMasters_n_w_ready	Output	User write ready.
io_ddrMasters_n_w_payload_data[127m:0]	Input	User write data. m is 31, 63, or 127.
io_ddrMasters_n_w_payload_strb[15:0]	Input	User write strobe.
io_ddrMasters_n_w_payload_last	Input	User write last.

Table 27: User Master Write Respond Channel

Where n is channel number.

Port	Direction	Description
io_ddrMasters_n_b_valid	Output	User write respond valid.
io_ddrMasters_n_b_ready	Input	User respond ready.
io_ddrMasters_n_b_payload_id[7:0]	Output	User respond ID.
io_ddrMasters_n_b_payload_resp[1:0]	Output	User write respond.

Table 28: User Master Read Address Channel

Where n is the channel number.

Port	Direction	Description
io_ddrMasters_n_ar_valid	Input	User read address valid.
io_ddrMasters_n_ar_ready	Output	User read address ready.
io_ddrMasters_n_ar_payload_addr[31:0]	Input	User read address.
io_ddrMasters_n_ar_payload_id[7:0]	Input	User read address ID.
io_ddrMasters_n_ar_payload_region[3:0]	Input	User region identifier.
io_ddrMasters_n_ar_payload_len[7:0]	Input	User burst length.
io_ddrMasters_n_ar_payload_size[2:0]	Input	User burst size.
io_ddrMasters_n_ar_payload_burst[1:0]	Input	User burst type, INCR only.
io_ddrMasters_n_ar_payload_lock	Input	User lock type.
io_ddrMasters_n_ar_payload_cache[3:0]	Input	User memory type.
io_ddrMasters_n_ar_payload_qos[3:0]	Input	User quality of service.
io_ddrMasters_n_ar_payload_prot[2:0]	Input	User protection type.

Table 29: User Master Read Data Channel

Where n is the channel number.

Port	Direction	Description
io_ddrMasters_n_r_valid	Output	User read valid.
io_ddrMasters_n_r_ready	Input	External memory read ready.
io_ddrMasters_n_r_payload_data[127m:0]	Output	External memory read data. m is 31, 63, or 127.
io_ddrMasters_n_r_payload_id[7:0]	Output	External memory read ID.
io_ddrMasters_n_r_payload_resp[1:0]	Output	External memory read respond.

Port	Direction	Description
io_ddrMasters_n_r_payload_last	Output	External memory read last.

APB3 Interface

The following table shows the ports for the APB3 user slave peripheral. Refer to the AMBA APB Protocol Specification for APB port descriptions and handshake information.

Table 30: APB3 Ports

Where n is 0, 1, 2, 3, or 4

Port	Direction	Description
io_apbSlave_n_PADDR[15:0]	Output	User address.
io_apbSlave_n_PSEL	Output	User select.
io_apbSlave_n_PENABLE	Output	User enable.
io_apbSlave_n_PREADY	Input	User ready.
io_apbSlave_n_PWRITE	Output	User direction.
io_apbSlave_n_PWDATA[31:0]	Output	User write data.
io_apbSlave_n_PRDATA[31:0]	Input	User read data.
io_apbSlave_n_PSLVERROR	Input	User transfer failure.

JTAG Interface

The Sapphire SoC uses the JTAG User TAP interface block to communicate with the OpenOCD debugger.

Table 31: JTAG Ports

Port	Direction	Description
jtagCtrl_enable	Input	Indicates that the user instruction is active for the interface.
jtagCtrl_capture	Input	TAP controller is in the capture state.
jtagCtrl_shift	Input	TAP controller is in the shift state.
jtagCtrl_update	Input	TAP controller in the update state.
jtagCtrl_reset	Input	TAP controller is in the reset state.
jtagCtrl_tdi	Input	JTAG TDI for debugging.
jtagCtrl_tdo	Output	JTAG TDO for debugging.
jtagCtrl_tck	Input	JTAG TCK for debugging.

Custom Instruction Interface

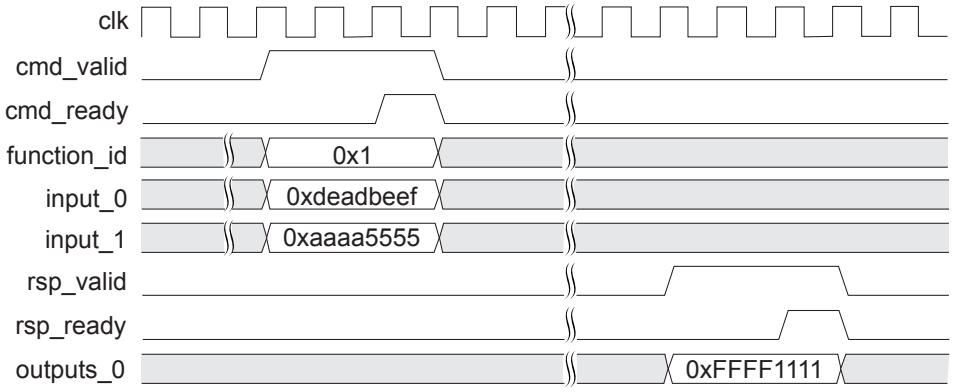
The Sapphire SoC supports a custom instruction interface so you can accelerate software functions with custom hardware logic. The custom instruction supports R-type instructions, which provides two registers (`rs1` and `rs2`) to custom instruction processing logic and up to 1,024 IDs to perform different functions.

Table 32: Custom Instruction Ports

Where n is the core number (0, 1, 2, or 3).

Port	Direction	Description
cpun_customInstruction_cmd_valid	Output	Indicates that registers rs1 and rs2 are present and ready for processing.
cpun_customInstruction_cmd_ready	Input	Indicates that the custom processing logic is ready to process register rs1 and rs2 from the CPU.
cpun_customInstruction_function_id[9:0]	Output	Function id for the custom instruction.
cpun_customInstruction_inputs_0[31:0]	Output	Register rs1 for the custom instruction.
cpun_customInstruction_inputs_1[31:0]	Output	Register rs2 for the custom instruction.
cpun_customInstruction_rsp_valid	Input	Indicates that the custom instruction result is available.
cpun_customInstruction_rsp_ready	Output	Indicates that the CPU is ready to accept the custom instruction result.
cpun_customInstruction_outputs_0[31:0]	Input	Result of the custom instruction.

Figure 6: Custom Instruction Waveform



GPIO Peripheral Interface

Use the `SYSTEM_GPIO_0_IO_CTRL` or `SYSTEM_GPIO_1_IO_CTRL` parameter to reference the GPIO interface.

Table 33: GPIO Ports

Where n is 0 or 1, and where BIT can be configured to as 1, 2, 4, 8, or 16 bits.

Port	Direction	Description
system_gpio_n_io_read[BIT-1:0]	Input	GPIO input.
system_gpio_n_io_write[BIT-1:0]	Output	GPIO output.
system_gpio_n_io_writeEnable[BIT-1:0]	Output	GPIO output enable.

Table 34: GPIO Register Map

Address Offset	Register Name	Privilege	Width
0x0000_0000	Input	Read	32
0x0000_0004	Output	Read/Write	32
0x0000_0008	Output Enable	Read/Write	32
0x0000_0020	Interrupt Rise Enable	Read/Write	32
0x0000_0024	Interrupt Fall Enable	Read/Write	32
0x0000_0028	Interrupt High Enable	Read/Write	32
0x0000_002C	Interrupt Low Enable	Read/Write	32

Input Register: 0x0000_0000

31 … 16	15 … 0
Reserved	Input

Bits	Field	Description	Privilege
0-(BIT-1)	Input	Input state of the GPIO pin (up to 16 pins). 1'b1: GPIO in high state. 1'b0: GPIO in low state.	Read
BIT-31	Reserved	Reserved.	N/A

Output Register: 0x0000_0004

31 … 16	15 … 0
Reserved	Output

Bits	Field	Description	Privilege
0-(BIT-1)	Output	Output state of the GPIO pin (up to 16 pins). 1'b1: Configure GPIO as high. 1'b0: Configure GPIO as low.	Read/Write
BIT-31	Reserved	Reserved.	N/A

Output Enable Register: 0x0000_0008

31	16	15	0
Reserved		OE	

Bits	Field	Description	Privilege
0-(BIT-1)	OE	Enable GPIO output pin (up to 16 pins). 1'b1: Configure GPIO as output. 1'b0: Configure GPIO as input.	Read/Write
BIT-31	Reserved	Reserved.	N/A

Interrupt Rise Enable Register: 0x0000_0020

31	2	1	0
Reserved		IntRiseEn	

Bits	Field	Description	Privilege
0-1	IntRiseEn	Enable a rise interrupt on GPIO pins 0 and 1.	Read/Write
2-31	Reserved	Reserved.	N/A

Interrupt Fall Enable Register: 0x0000_0024

31	2	1	0
Reserved		IntFallEn	

Bits	Field	Description	Privilege
0-1	IntFallEn	Enable a fall interrupt on GPIO pins 0 and 1.	Read/Write
2-31	Reserved	Reserved.	N/A

Interrupt High Enable Register: 0x0000_0028

31	2	1	0
Reserved		IntHighEn	

Bits	Field	Description	Privilege
0-1	IntHighEn	Enable a high interrupt on GPIO pins 0 and 1.	Read/Write
2-31	Reserved	Reserved.	N/A

Interrupt Low Enable Register: 0x0000_002C

31	2	1	0
Reserved		IntLowEn	

Bits	Field	Description	Privilege
0-1	IntLowEn	Enable a low interrupt on GPIO pins 0 and 1.	Read/Write
2-31	Reserved	Reserved.	N/A

I²C Peripheral Interface

The Sapphire SoC has up to 3 I²C master/slave peripherals. You use the `system_i2c_2*` ports to calibrate the DDR DRAM memory; if you do not want to perform calibration, you can use this peripheral for your own purposes. Use these parameters to reference the interface:

Table 35: I²C Peripheral Ports (User)

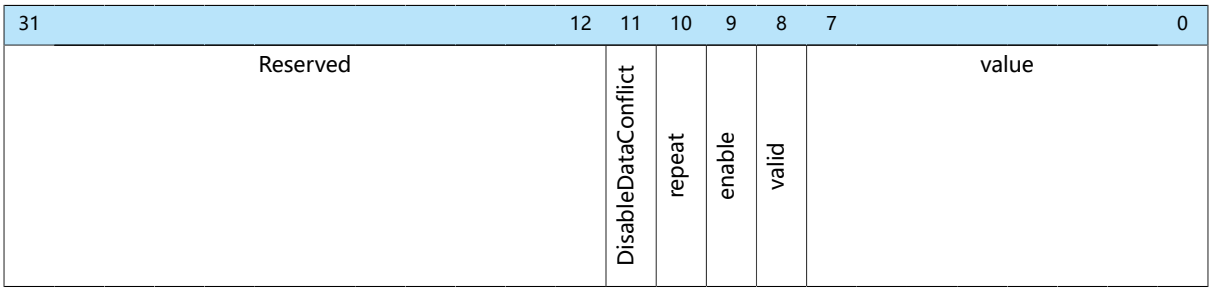
Where n is 0, 1, or 2.

Port	Direction	Description
system_i2c_n_io_sda_write	Output	SDA output for user device.
system_i2c_n_io_sda_read	Input	SDA input for user device.
system_i2c_n_io_scl_write	Output	SCL output for user device.
system_i2c_n_io_scl_read	Input	SCL input for user device.

Table 36: I²C Register Map

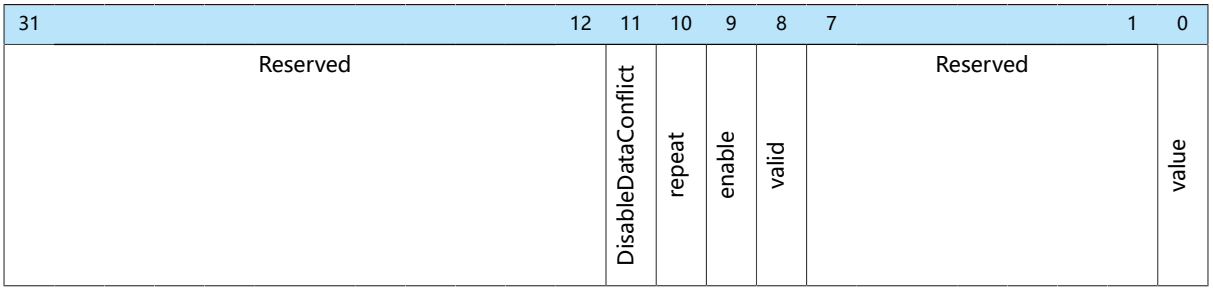
Address Offset	Register Name	Privilege	Width
0x0000_0000	txData	Read/Write	32
0x0000_0004	txAck	Read/Write	32
0x0000_0008	rxData	Read/Write	32
0x0000_000C	rxAck	Read/Write	32
0x0000_0020	Interrupt	Read/Write	32
0x0000_0024	Interrupt Clears	Read/Write	32
0x0000_0028	Sampling Clock Divider	Write	32
0x0000_002C	Timeout	Write	32
0x0000_0030	tsuData	Write	32
0x0000_0040	Master Status	Read/Write	32
0x0000_0050	tlow	Write	32
0x0000_0054	tHigh	Write	32
0x0000_0058	tBuf	Write	32
0x0000_0080	Filtering Status	Read	32
0x0000_0084	Hit Context	Read	32
0x0000_0088	Filtering Configuration 0	Write	32
0x0000_008C	Filtering Configuration 1	Write	32

txData Register: 0x0000_0000



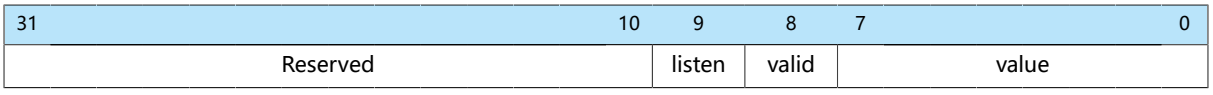
Bits	Field	Description	Privilege
0-7	value	Transmit data value.	Write
8	valid	Transmit data valid bit.	Read/Write
9	enable	Transmit data enable.	Read/Write
10	repeat	Transmit data repeat bit.	Write
11	DisableDataConflict	Disable transmit data conflict.	Write
12-31	Reserved	Reserved.	N/A

txAck Register: 0x0000_0004



Bits	Field	Description	Privilege
0	value	Transmit acknowledge bit.	Write
1-7	Reserved	Reserved.	N/A
8	valid	Transmit acknowledge valid bit.	Read/Write
9	enable	Transmit acknowledge enable.	Read/Write
10	repeat	Transmit acknowledge repeat bit.	Write
11	DisableDataConflict	Disable transmit acknowledge conflict.	Write
12-31	Reserved	Reserved.	N/A

rxData Register: 0x0000_0008



Bits	Field	Description	Privilege
0-7	value	Received data.	Read
8	valid	Receive data valid.	Read
9	listen	Start listen data.	Write
10-31	Reserved	Reserved.	N/A

rxAck Register: 0x0000_000C

31	10	9	8	7	1	0
Reserved		listen	valid	Reserved		value

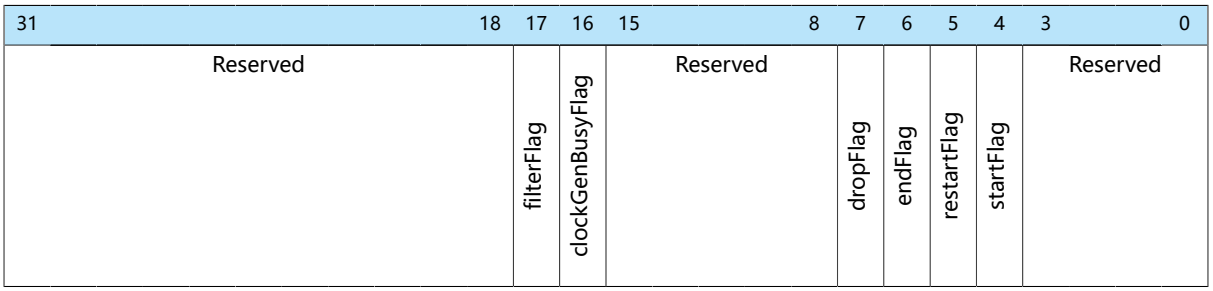
Bits	Field	Description	Privilege
0	value	Received acknowledge.	Read
1-7	Reserved	Reserved.	N/A
8	valid	Receive acknowledge valid.	Read
9	listen	Start listen acknowledge.	Write
10-31	Reserved	Reserved.	N/A

Interrupt Register: 0x0000_0020

31	18	17	16	15	8	7	6	5	4	3	2	1	0
Reserved		filterEnable	clockGenBusyEnable	Reserved		dropEnable	endEnable	restartEnable	startEnable	txAckEnable	txDataEnable	rxAckEnable	rxDataEnable

Bits	Field	Description	Privilege
0	rxDataEnable	Receive data interrupt enable	Read/Write
1	rxAckEnable	Receive acknowledge interrupt enable	Read/Write
2	txDataEnable	Transmit data interrupt enable	Read/Write
3	txAckEnable	Transmit acknowledge interrupt enable	Read/Write
4	startEnable	Start interrupt enable	Read/Write
5	restartEnable	Restart interrupt enable	Read/Write
6	endEnable	End interrupt enable	Read/Write
7	dropEnable	Drop interrupt enable	Read/Write
8-15	Reserved	Reserved.	N/A
16	clockGenBusyEnable	Master clock generation interrupt enable.	Read/Write
17	filterEnable	Slave address filtering hit interrupt enable	Read/Write
18-31	Reserved	Reserved.	N/A

Interrupt Clears Register: 0x0000_0024

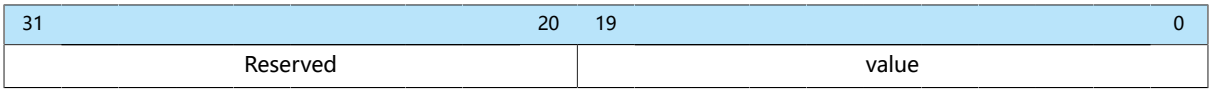


Bits	Field	Description	Privilege
0-3	Reserved	Reserved.	N/A
4	startFlag	Start interrupt flag.	Read/Write
5	restartFlag	Restart interrupt flag.	Read/Write
6	endFlag	End interrupt flag.	Read/Write
7	dropFlag	Drop interrupt flag.	Read/Write
8 - 15	Reserved	Reserved.	N/A
16	clockGenBusyFlag	Master clock generation interrupt flag.	Write
17	filterFlag	Slave address filtering hit interrupt	Write
18-31	Reserved	Reserved.	N/A

Sampling Clock Divider Register: 0x0000_0028

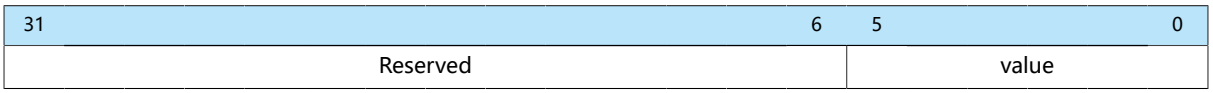
Bits	Field	Description	Privilege
0-9	samplingClockDividerWidth	Clock divider width. Controls the rate at which the I^2C controller reads SCL and SDA.	Write
10-31	Reserved	Reserved.	N/A

Timeout Register: 0x0000_002C



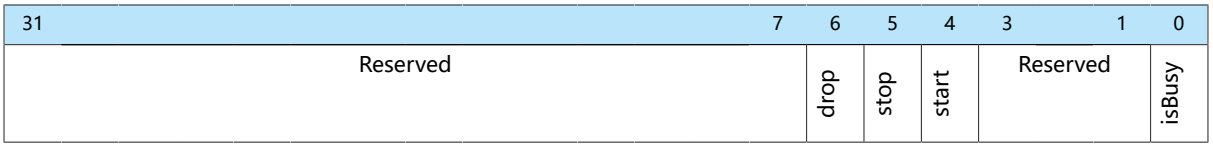
Bits	Field	Description	Privilege
0-19	value	Inactive timeout setting.	Write
20-31	Reserved	Reserved.	N/A

tsuData Register: 0x0000_0030



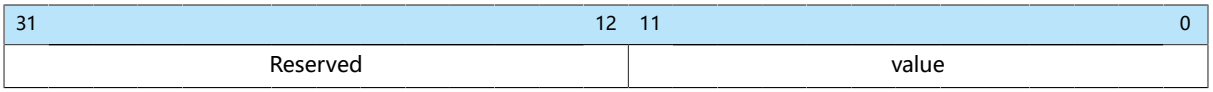
Bits	Field	Description	Privilege
0-5	value	Data setup time.	Write
6-31	Reserved	Reserved.	N/A

Master Status Register: 0x0000_0040



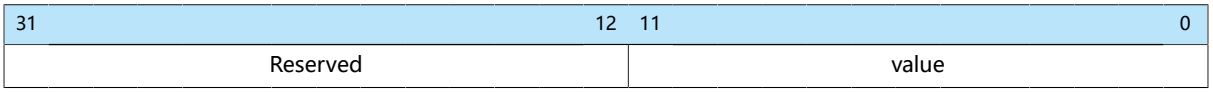
Bits	Field	Description	Privilege
0	isBusy	Master busy.	Read
1-3	Reserved	Reserved.	N/A
4	start	Master start.	Read/Write
5	stop	Master stop.	Read/Write
6	drop	Master drop.	Read/Write
6-31	Reserved	Reserved.	N/A

tLow Register: 0x0000_0050



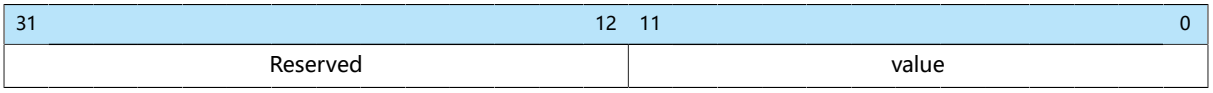
Bits	Field	Description	Privilege
0-11	value	SCL low period.	Write
12-31	Reserved	Reserved.	N/A

tHigh Register: 0x0000_0054



Bits	Field	Description	Privilege
0-11	value	SCL high period.	Write
12-31	Reserved	Reserved.	N/A

tBuf Register: 0x0000_0058



Bits	Field	Description	Privilege
0-11	value	Start and stop bus free time.	Write
12-31	Reserved	Reserved.	N/A

Filtering Status Register: 0x0000_0080

Bits	Field	Description	Privilege
0	hit_0	Filtering hit bit 0.	Read
1	hit_1	Filtering hit bit 1.	Read
2-32	Reserved	Reserved.	N/A

Hit Context Register: 0x0000_0084

31 … 1	0
Reserved	rw

Bits	Field	Description	Privilege
0	rw	Hit context read.	Read
1-31	Reserved	Reserved.	N/A

Filtering Configuration 0 Register: 0x0000_0088

31 … 15	14	13 … 10	9 … 0
	enable	Reserved	value
	10 bit address		

Bits	Field	Description	Privilege
0-9	value	Set the filter 0 value.	Write
10-13	Reserved	Reserved.	N/A
14	10 bit address	Set the filter address to 10 bits wide.	Write
15	enable	Enable filter address 0.	Write
1-31	Reserved	Reserved.	N/A

Filtering Configuration 1 Register: 0x0000_008C

31 … 15	14	13 … 10	9 … 0
	enable	Reserved	value
	10 bit address		

Bits	Field	Description	Privilege
0-9	value	Set the filter 1 value.	Write
10-13	Reserved	Reserved.	N/A
14	10 bit address	Set the filter address to 10 bits wide.	Write
15	enable	Enable filter address 1.	Write
1-31	Reserved	Reserved.	N/A

PLIC Peripheral Interface

Use the `SYSTEM_PLIC_CTRL` parameter to reference the interface PLIC interface.

Table 37: RISC-V PLIC Operation Parameters

Defines	Description
Interrupt priorities registers	The interrupt priority for each interrupt source.
Interrupt pending bits registers	The interrupt pending status of each interrupt source.
Interrupt enables registers	Enables the interrupt source of each context.
Priority thresholds registers	The interrupt priority threshold of each context.
Interrupt claim registers	The register to acquire interrupt source ID of each context.
Interrupt completion registers	The register to send interrupt completion message to the associated gateway.

The **soc.h** file contains a number of PLIC parameters to specify the interrupt ID for the various peripherals.

Table 38: PLIC Interrupt ID Parameters

Where n is the peripheral number and m is the interrupt ID.

Parameter	Refer to
SYSTEM_PLIC_SYSTEM_I2C_n_IO_INTERRUPT m	**Interrupt Register: 0x0000_0020** on page 24 **Interrupt Clears Register: 0x0000_0024** on page 25
SYSTEM_PLIC_SYSTEM_GPIO_n_IO_INTERRUPTS_0 m	**Interrupt Low Enable Register: 0x0000_002C** on page 21 **Interrupt High Enable Register: 0x0000_0028** on page 21 **Interrupt Fall Enable Register: 0x0000_0024** on page 21 **Interrupt Rise Enable Register: 0x0000_0020** on page 21
SYSTEM_PLIC_SYSTEM_AXI_A_INTERRUPT	**Interrupts** on page 12
SYSTEM_PLIC_SYSTEM_SPI_n_IO_INTERRUPT m	**Interrupt Register: 0x0000_000C** on page 31
SYSTEM_PLIC_SYSTEM_UART_n_IO_INTERRUPT m	**Status Register: 0x0000_0004** on page 34
SYSTEM_PLIC_USER_INTERRUPT_A_INTERRUPT SYSTEM_PLIC_USER_INTERRUPT_B_INTERRUPT SYSTEM_PLIC_USER_INTERRUPT_C_INTERRUPT SYSTEM_PLIC_USER_INTERRUPT_D_INTERRUPT SYSTEM_PLIC_USER_INTERRUPT_E_INTERRUPT SYSTEM_PLIC_USER_INTERRUPT_F_INTERRUPT SYSTEM_PLIC_USER_INTERRUPT_G_INTERRUPT SYSTEM_PLIC_USER_INTERRUPT_H_INTERRUPT	**Interrupts** on page 12
SYSTEM_PLIC_SYSTEM_USER_TIMER_n_INTERRUPTS_m	**Timer Limit Register: 0x0000_00044**

SPI Master Peripheral Interface

The SPI master peripheral interface supports traditional dual-line full-duplex mode as well as half-duplex mode in 2 and 4-wire SPI. The SPI data width is configurable up to 16 bits. Half-duplex mode is only available when the SPI data width is configured as 8 or 16. When implementing the SPI peripheral in traditional dual-line mode, use the `data_0` ports as MOSI and and the `data_1` ports as MISO.

Use these parameters to reference the interface:

- SPI master 0—SYSTEM_SPI_0_IO_CTRL
- SPI master 1—SYSTEM_SPI_1_IO_CTRL
- SPI master 2—SYSTEM_SPI_2_IO_CTRL

Table 39: SPI Master Ports

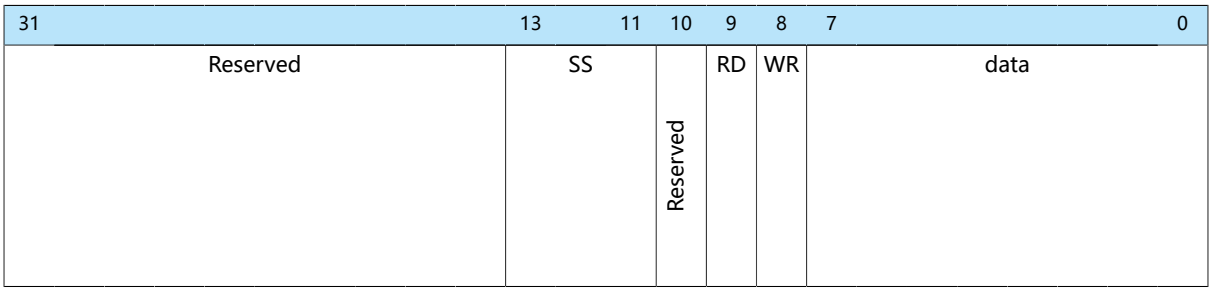
Where n is 0, 1, or 2

Port	Direction	Description
system_spi_n_io_sclk_write	Output	SPI SCK.
system_spi_n_io_data_0_writeEnable	Output	SPI output enable for data 0.
system_spi_n_io_data_0_read	Input	SPI input for data 0.
system_spi_n_io_data_0_write	Output	SPI output for data 0.
system_spi_n_io_data_1_writeEnable	Output	SPI output enable for data 1.
system_spi_n_io_data_1_read	Input	SPI input for data 1.
system_spi_n_io_data_1_write	Output	SPI output for data 1.
system_spi_n_io_data_2_writeEnable	Output	SPI output enable for data 2.
system_spi_n_io_data_2_read	Input	SPI input for data 2.
system_spi_n_io_data_2_write	Output	SPI output for data 2.
system_spi_n_io_data_3_read	Input	SPI input for data 3.
system_spi_n_io_data_3_write	Output	SPI output for data 3.
system_spi_n_io_data_3_writeEnable	Output	SPI output enable for data 3.
system_spi_n_io_ss	Output	SPI SS.

Table 40: SPI Master Register Map

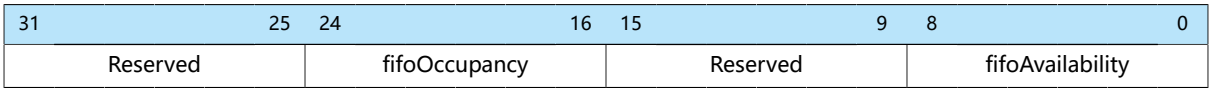
Address Offset	Register Name	Privilege	Width
0x0000_0000	Cmd	Read/Write	32
0x0000_0004	RSP	Read	32
0x0000_0008	Config	Write	32
0x0000_000C	Interrupt	Read/Write	32
0x0000_0020	ClockDivider	Write	32
0x0000_0024	ssSetup	Write	32
0x0000_0028	ssHold	Write	32
0x0000_002C	ssDisable	Write	32
0x0000_0030	ssActiveHigh	Write	32
0x0000_0050	cmd32_0	Write	32
0x0000_0054	cmd32_1	Write	32
0x0000_0058	rsp32	Read	32

Cmd Register: 0x0000_0000



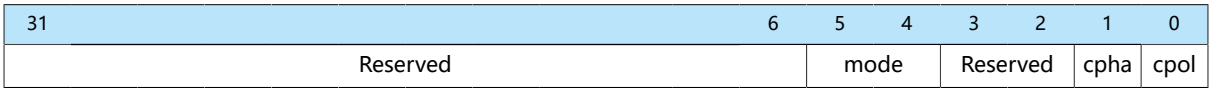
Bits	Field	Description	Privilege
0-7	data	FIFO data value transmit/receive.	Read/Write
8	WR	Write trigger.	Write
9	RD	Read trigger.	Write
10	Reserved	Reserved.	N/A
11-13	SS	SPI chip select. Configure up to 8 chip select ID numbers.	Read/Write
14-31	Reserved	Reserved.	N/A

RSP Register: 0x0000_0004



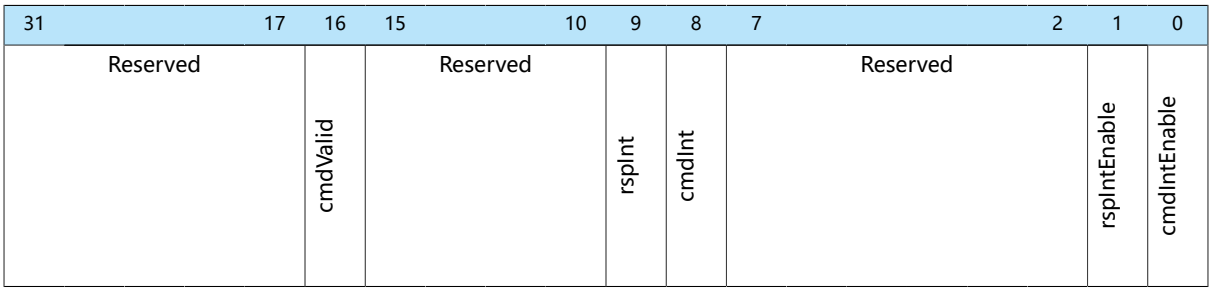
Bits	Field	Description	Privilege
0-8	fifoAvailability	FIFO Availability.	Read
9-15	Reserved	Reserved.	N/A
16-24	fifoOccupancy	FIFO Occupancy.	Read
25-31	Reserved	Reserved.	N/A

Config Register: 0x0000_0008



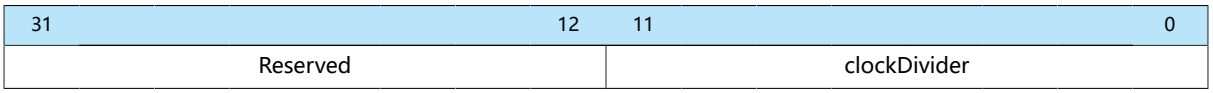
Bits	Field	Description	Privilege
0	cpol	Clock polarity setting.	Write
1	cpha	Clock phase setting.	Write
2-3	Reserved	Reserved.	N/A
4-5	mode	0: Full duplex dual line. 1: Half-duplex dual line. 2: Half-duplex quad line.	Write
6-31	Reserved	Reserved.	N/A

Interrupt Register: 0x0000_000C



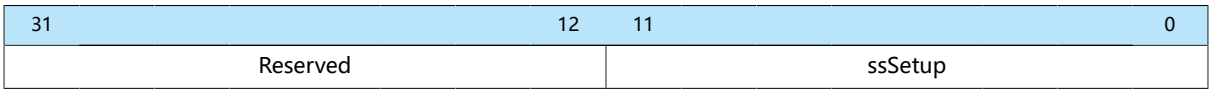
Bits	Field	Description	Privilege
0	cmdIntEnable	Command FIFO empty interrupt enable.	Read/Write
1	rspIntEnable	Read FIFO not empty interrupt enable.	Read/Write
2-7	Reserved	Reserved.	N/A
8	cmdInt	Command FIFO empty interrupt pending.	Read/Write
9	rspInt	Read FIFO not empty interrupt pending.	Read/Write
10-15	Reserved	Reserved.	N/A
16	cmdValid	Command FIFO is not empty.	Read
17-31	Reserved	Reserved.	N/A

clockDivider Register: 0x0000_0020



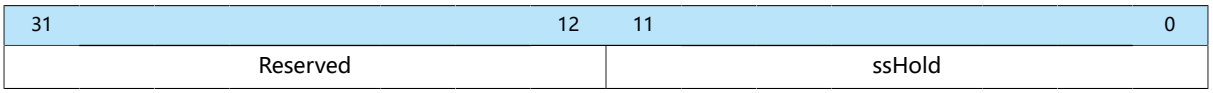
Bits	Field	Description	Privilege
0-11	clockDivider	SPI frequency = FCLK / ((clockDivider+1)*2 FCLK is the system clock (io_systemClk) to the SoC. If you enable the peripheral clock, then FCLK is driven by the peripheral clock (io_peripheralClk) instead.	Write
12-31	Reserved	Reserved.	N/A

ssSetup Register: 0x0000_0024



Bits	Field	Description	Privilege
0-11	ssSetup	Time between the chip select enable and the next byte.	Write
12-31	Reserved	Reserved.	N/A

ssHold Register: 0x0000_0028



Bits	Field	Description	Privilege
0-11	ssHold	Time between the last byte transmission and the chip select disable.	Write
12-31	Reserved	Reserved.	N/A

ssDisable Register: 0x0000_002C

31	12	11	0
Reserved		ssDisable	

Bits	Field	Description	Privilege
0-11	ssDisable	Time between the chip select disable and the chip select enable.	Write
12-31	Reserved	Rserved.	N/A

ssActiveHigh Register: 0x0000_0030

31	8	7	0
Reserved		ssActiveHigh	

Bits	Field	Description	Privilege
0-7	ssActiveHigh	These bits correspond to the hardware SPI chip select. 0: Chip select is active low. 1: Chip select is active high.	Write
8-31	Reserved	Rsreved.	N/A

cmd32_0 Register: 0x0000_0050

31	16	15	0
Reserved		data	

Bits	Field	Description	Privilege
0-15	data	Trigger to transmit data. The data width is configurable up to 16 bits. It should be more than 8 bits.	Write
16-31	Reserved	Reserved.	N/A

cmd32_1 Register: 0x0000_0054

31	16	15	0
Reserved		data	

Bits	Field	Description	Privilege
0-15	data	Trigger to transmit data. The data width is configurable up to 16 bits. It should be more than 8 bits.	Write
16-31	Reserved	Reserved.	N/A

rsp32 Register: 0x0000_0058

31	16	15	0
Reserved		data	

Bits	Field	Description	Privilege
0-15	data	Trigger to receive data. The data width is configurable up to 16 bits. It should be more than 8 bits.	Write
16-31	Reserved	Reserved.	N/A

UART Peripheral Interface

You can configure up to 3 UART peripherals. Each UART peripheral runs at 115200 baud and supports 8 data bits, no parity, and 1 stop bit. Use these parameters to reference the interface:

- UART 0—`SYSTEM_UART_0_IO_CTRL`
- UART 1—`SYSTEM_UART_1_IO_CTRL`
- UART 1—`SYSTEM_UART_2_IO_CTRL`

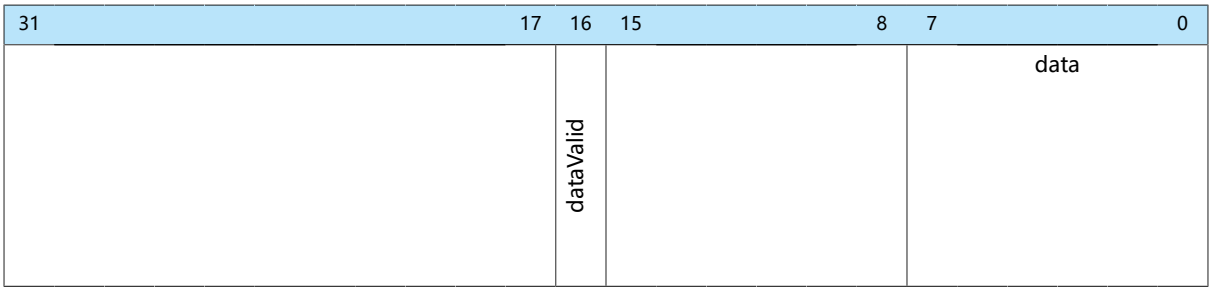
Table 41: UART Ports

Port	Direction	Description
system_uart_0_io_txd	Output	UART 0 transmit.
system_uart_0_io_rxd	Input	UART 0 receive.
system_uart_1_io_txd	Output	UART 1 transmit.
system_uart_1_io_rxd	Input	UART 1 receive.
system_uart_2_io_txd	Output	UART 2 transmit.
system_uart_2_io_rxd	Input	UART 2 receive.

Table 42: UART Register Map

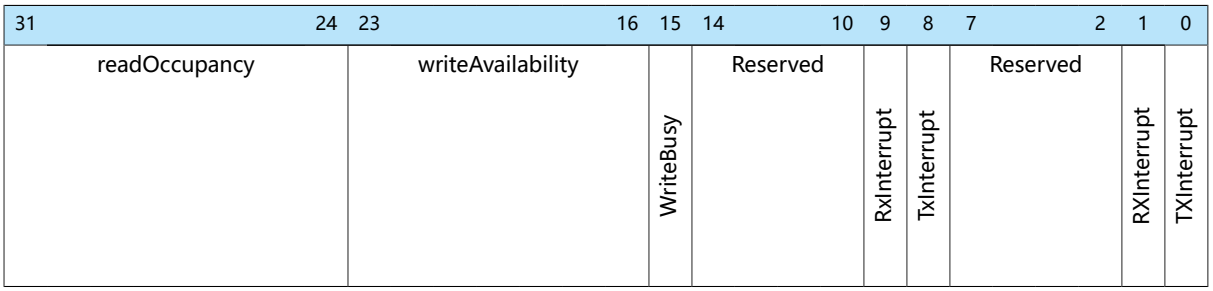
Address Offset	Register Name	Privilege	Width
0x0000_0000	Data	Read/Write	32
0x0000_0004	Status	Read/Write	32
0x0000_0008	Clock divider	Write	32
0x0000_000C	Config register	Write	32
0x0000_0010	Error break	Read/Write	32

Data Register: 0x0000_0000



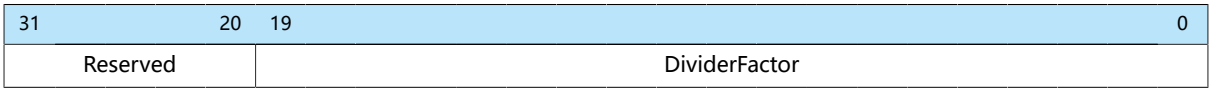
Bits	Field	Description	Privilege
0-7	data	Stores data that is transmitted or received.	Read/Write
8-15	Reserved	Reserved.	N/A
16	dataValid	Read data is valid.	Read
17-31	Reserved	Reserved.	N/A

Status Register: 0x0000_0004



Bits	Field	Description	Privilege
0	TXInterruptEnable	Interrupt when TX FIFO is empty.	Read/Write
1	RXInterruptEnable	Interrupt when RX FIFO is not empty.	Read/Write
2-7	Reserved	Reserved.	N/A
8	TxInterrupt	Interrupt when the TX FIFO is empty.	Read
9	RxInterrupt	Interrupt when the RX FIFO is not empty.	Read
10-14	Reserved	Reserved.	N/A
15	WriteBusy	Write operation in progress.	Read
16-23	writeAvailability	UART FIFO availablity.	Read
24-31	readOccupancy	UART FIFO occupancy.	Read

Clock Divider Register: 0x0000_0008



Bits	Field	Description	Privilege
0-19	DividerFactor	Divider factor for the UART baud rate. Baudrate = io_systemClk/(Data Length * DividerFactor)	Write
20-31	Reserved	Reserved.	N/A

Config Register: 0x0000_000C

31 … 17	16	15 … 10	9 … 8	7 … 3	2 … 0
Reserved	Stop	Reserved	Parity	Reserved	DataLength

Bits	Field	Description	Privilege
0-2	DataLength	Data length.	Write
3-7	Reserved	Reserved.	???
8-9	Parity	Parity bit number. 0: None 1: Even 2: Odd	Write
10-15	Reserved	Reserved.	
16	Stop	Stop bit number.	Write
17-31	Reserved	Reserved.	N/A

Error Break Register: 0x0000_0010

31 … 12	11	10	9	8	7 … 2	1	0
Reserved	DisableBreak	EnableBreak	BreakDetect	ReadBreak	Reserved	ReadOverFlow	ReadError

Bits	Field	Description	Privilege
0	ReadError	Error occurred during UART read operation. Write 1’ b1 to reset the error.	Read/Write
1	ReadOverFlow	RX FIFO overflow error occurred. Write 1’ b1 to reset the error.	Read/Write
2-7	Reserved	Reserved.	N/A
8	ReadBreak	Read break occurred	Read
9	BreakDetect	Break detected during read operation. Write 1’ b1 to reset the error.	Read/Write
10	EnableBreak	Enable break.	Write
11	DisableBreak	Disable break.	Write
12-31	Reserved	Reserved.	N/A

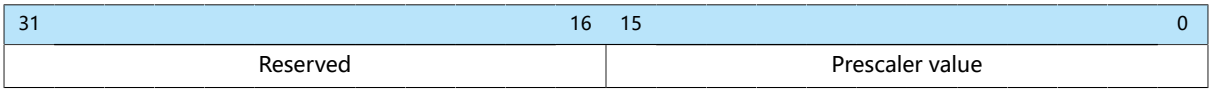
User Timer

You can configure up to three user timers so you can perform actions such as timestamp and interrupts without using the core timer. You can adjust the interval period to generate a timer tick pulse by setting the prescaler register, based on the system clock or peripheral clock (if enabled).

Table 43: User Timer Register Map

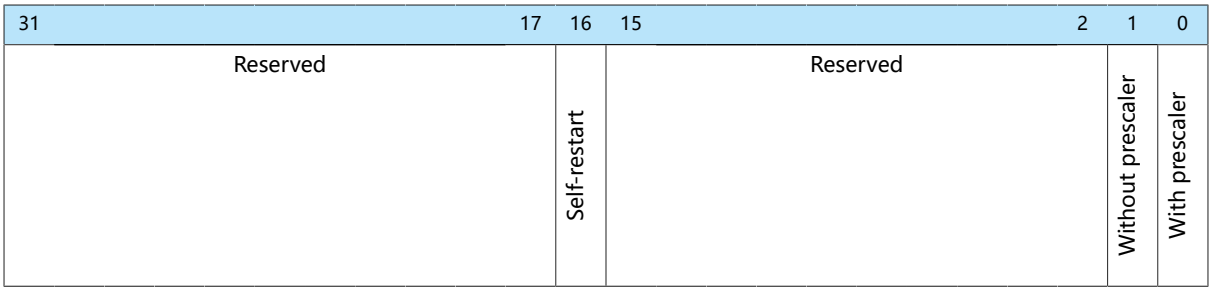
Address Offset	Register Name	Privilege	Width
0x0000_0000	Prescaler	Read/Write	32
0x0000_0040	Timer configuration	Read/Write	32
0x0000_0044	Timer limit	Read/Write	32
0x0000_0048	Timer value	Read	32

Prescaler Register: 0x0000_0000



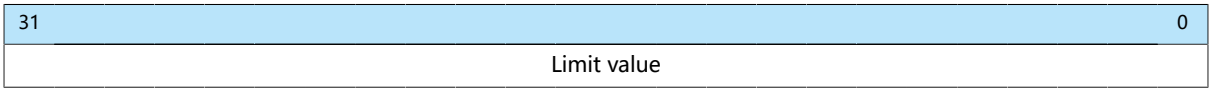
Bits	Field	Description	Privilege
0-15	Prescaler value	The clock divider ratio.	Read/Write
16-31	Reserved	Reserved.	-

Timer Configuration Register: 0x0000_0040



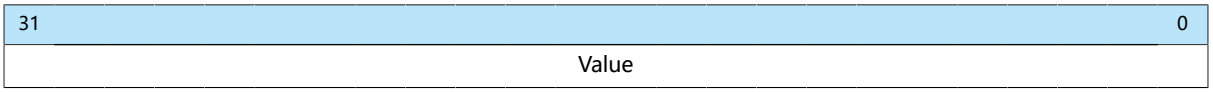
Bits	Field	Description	Privilege
0	With prescaler	Run timer with prescaler	Read/Write
1	Without prescaler	Run timer without prescaler	Read/Write
2-15	Reserved	Reserved.	N/A
16	Self-restart	self-restart when reach timer limit.	Read/Write
17-31	Reserved	Reserved.	N/A

Timer Limit Register: 0x0000_0044



Bits	Field	Description	Privilege
0-31	Limit value	Value for the timer to an generate output pulse. The final value with the prescaler enabled is: (limit value + 1) * (prescaler value + 1)	Read/Write

Timer Value Register: 0x0000_0048



Bits	Field	Description	Privilege
0-31	Value	Value of the increment counter when it detects a timer tick.	Read

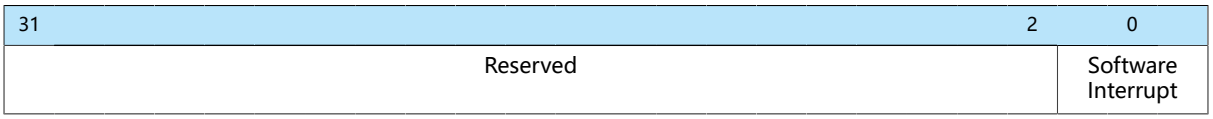
Clint

The core local interrupt (clint) consists of a 64-bit realtime counter, which is driven by `io_systemClk` or `io_peripheralClk` (if enabled). The clint counter value increases monotonically. Clint is also responsible for handling the control and status via software interrupt.

Table 44: Clint Register Map

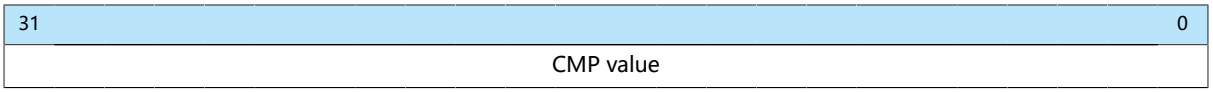
Address Offset	Register Name	Privilege	Width
0x0000_0000	PIP	Read/Write	32
0x0000_4000	MTIMECMP (LO)	Write	32
0x0000_4004	MTIMECMP (HI)	Write	32
0x0000_BFF8	MTIME (LO)	Read	32
0x0000_BFFC	MTIME (HI)	Read	32

PIP Register: 0x0000_0000



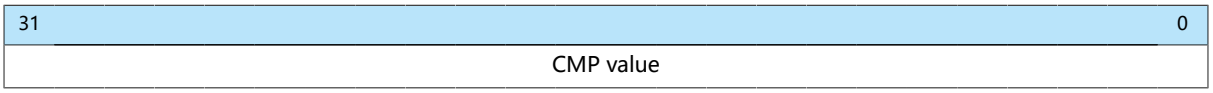
Bits	Field	Description	Privilege
0	Software Interrupt	Machine mode software interrupt.	Read/Write
2-31	Reserved	Reserved.	-

MTIMECMP Register (LO): 0x0000_4000



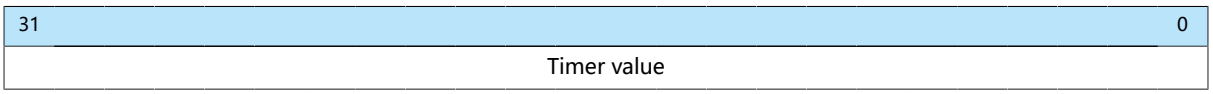
Bits	Field	Description	Privilege
0-31	CMP value	Timer interrupt trigger value (low 32 bits).	Write

MTIMECMP Register (HI): 0x0000_4004



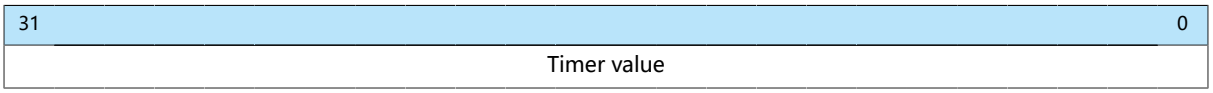
Bits	Field	Description	Privilege
0-31	CMP value	Timer interrupt trigger value (high 32 bits).	Write

MTIME Register (LO): 0x0000_BFF8



Bits	Field	Description	Privilege
0-31	Timer value	Value of increment counter (low 32 bits).	Read

MTIME Register (HI): 0x0000_BFFC



Bits	Field	Description	Privilege
0-31	Timer value	Value of increment counter (high 32 bits).	Read

Control and Status Registers

The following tables show the machine-level CSR implementation.

Table 45: Machine Information Register

Address	Register Name	Privilege	Description	Width
0xF14	mhartid	Read	Hardware thread ID.	32

Table 46: Machine Trap Registers

Address	Register Name	Privilege	Description	Width
0x300	mstatus	Read/Write	Machine status register.	13
0x304	mie	Read/Write	Machine interrupt enable register.	12
0x305	mtvec	Read/Write	Machine trap handler base address.	32

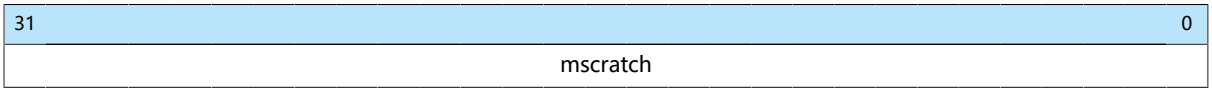
Table 47: Machine Trap Handling Registers

Address	Register Name	Privilege	Description	Width
0x340	mscratch	Read/Write	Scratch register for machine trap handlers.	32
0x341	mpec	Read/Write	Machine exception program counter.	32
0x342	mcause	Read	Machine trap cause.	32
0x343	mtval	Read	Machine bad address or instruction.	32
0x344	mip	Read/Write	Machine interrupt pending.	12

Machine-Level ISA

Machine Scratch Register (mscratch): 0x340

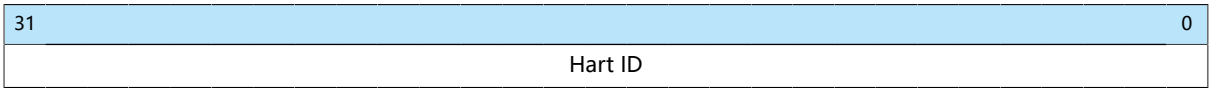
The `mscratch` register is a 32-bit read/write register dedicated for use by machine mode. Typically, it is used to hold a pointer to a machine-mode hart-local context space and swapped with a user register upon entry to an M-mode trap handler.



Bits	Field	Description	Privilege
0-31	mscratch	A temporary scratch space that can used by machine-mode software.	Read/Write

Hart ID Register (mhartid): 0xF14

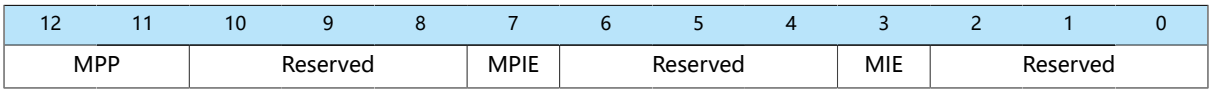
The `mhartid` CSR is a 32-bit read-only register containing the integer ID of the hardware thread running the code. This register must be readable in any implementation. Hart IDs might not necessarily be numbered contiguously in a multiprocessor system, but at least one hart must have a hart ID of zero. Hart IDs must be unique.



Bits	Field	Description	Privilege
0-31	Hart ID	Hardware thread ID.	Read

Machine Status Register (mstatus): 0x300

The `mstatus` register is a 13-bits read/write register formatted. The `mstatus` register keeps track of and controls the hart' s current operating state. Restricted views of the `mstatus` register appear as the `sstatus` and `ustatus` registers in the S-level and U-level ISAs, respectively.



Bits	Field	Description	Privilege
0-2	Reserved	Reserved.	N/A
3	MIE	Machine interrupt enable register.	Read/Write
4-6	Reserved	Reserved.	N/A
7	MPIE	Machine previous interrupt enable.	Read/Write
8-10	Reserved	Reserved.	N/A
11-12	MPP	Machine Previous privilege mode.	Read/Write

Machine Trap-Vector Base-Address Register (mtvec): 0x305

The `mtvec` register is a 32-bit read/write register that holds trap vector configuration, consisting of a vector base address (base) and a vector mode (mode).

Bits	Field	Description	Privilege
0-1	mode	Vector mode. 0: Direct. All exceptions set pc to BASE 1: Vectored. Asynchronous interrupts set pc to BASE + 4xcause ≥ 2: Reserved	Read/Write
2-31	base	Vector base address.	Read/Write

Machine Interrupt Enable Register (mie): 0x304

The `mie` register is a 12-bit read/write register containing interrupt enable bits.

11	10	9	8	7	6	5	4	3	2	1	0
MEIE	Reserved			MTIE	Reserved			MSIE	Reserved		

Bits	Field	Description	Privilege
0-2	Reserved	Reserved.	N/A
3	MSIE	Machine software interrupt enable.	Read/Write
4-6	Reserved	Reserved.	N/A
7	MTIE	Machine timer interrupt enable.	Read
8-10	Reserved	Reserved.	N/A
11	MEIE	Machine external interrupt enable.	Read

Machine Exception Program Counter (mepc): 0x341

`mepc` is a 32-bit read/write register. The low bit of `mepc` (`mepc[0]`) is always zero. On implementations that support only IALIGN=32, the two low bits (`mepc[1:0]`) are always zero.

31	0
mepc	

Bits	Field	Description	Privilege
0-31	mepc	Machine exception program counter.	Read/Write

Machine Cause Register (mcause): 0x342

The `mcause` register is a 32-bit read-write register. When a trap is taken into M-mode, `mcause` is written with a code indicating the event that caused the trap. Otherwise, `mcause` is never written by the implementation, though it may be explicitly written by software.

31	30	0
Interrupt	Exception Code	

Bits	Field	Description	Privilege
0-30	Exception code	See **Table 48: Machine Cause Register (mcause) Values after Trap** on page 41.	Read
31	Interrupt	mcause interrupt bit.	Read

Table 48: Machine Cause Register (mcause) Values after Trap

Interrupt	Exception Code	Description
1	0	Reserved.
1	1	Supervisor software interrupt.
1	2	Reserved.
1	3	Machine software interrupt.
1	4	User timer interrupt.
1	5	Supervisor timer interrupt.
1	6	Reserved.
1	7	Machine timer interrupt.
1	8	User external interrupt.

Interrupt	Exception Code	Description
1	9	Supervisor external interrupt.
1	10	Reserved.
1	11	Machine external interrupt.
1	≥12	Reserved.
0	0	Instruction address misaligned.
0	1	Instruction access fault.
0	2	Illegal instruction.
0	3	Breakpoint.
0	4	Load address misaligned.
0	5	Load access fault.
0	6	Store/AMO address misaligned.
0	7	Store/AMO access fault.
0	8	Reserved.
0	9	Reserved.
0	10	Reserved.
0	11	Environment call from M-mode.
0	12	Instruction page fault.
0	13	Load page fault.
0	14	Reserved.
0	15	Store/AMO page fault.
0	≥16	Reserved.

Machine Trap Value Register (mtval): 0x343

The `mtval` register is a 32-bit register. When a trap is taken into M-mode, `mtval` is either set to zero or written with exception-specific information to assist software in handling the trap. Otherwise, `mtval` is never written by the implementation, though it may be explicitly written by software. The hardware platform will specify which excep-tions must set `mtval` informatively and which may unconditionally set it to zero.

Bits	Field	Description	Privilege
0-31	mtval	Machine trap value register bit.	Read/Write

Machine Interrupt Pending Register (mip): 0x344

The `mip` register is a 12-bit read/write register containing information on pending interrupts.

11	10	9	8	7	6	5	4	3	2	1	0
MEIP	Reserved			MTIP	Reserved			MSIP	Reserved		

Bits	Field	Description	Privilege
0-2	Reserved	Reserved.	N/A
3	MSIP	Machine software interrupt pending.	Read/Write
4-6	Reserved	Reserved.	N/A
7	MTIP	Machine timer interrupt pending.	Read
8-10	Reserved	Reserved.	N/A
11	MEIP	Machine external interrupt pending.	Read

Revision History

Table 49: Revision History

Date	Version	Description
August 2022	3.0	Updated to add multi-core support specifications. Added clint description and registers. Added mscratch register.
June 2022	2.2	The VexRiscv core used in the Sapphire SoC has six pipeline stages.
February 2022	2.1	Updated the Config Register for the SPI Master Peripheral Interface. (DOC-692)
December 2021	2.0	The SoC now supports a floating point unit, Linux memory management unit, custom instruction, and optional RISC-V extensions such as atomic and compressed. The SoC has a core timer and up to 3 user timers. The machine timer is removed. The SoC has an optional I/O peripheral clock and reset for clocking the APB3 peripherals. The address map parameters have changed. Clarified AXI interface description. (DOC-633)
September 2021	1.1	The SoC minimum frequency changed to 20 MHz. (DOC-544) Updated resource utilization and performance. (DOC-544) The APB slave size is configurable. (DOC-544) The AXI slave size is 256 MB maximum. (DOC-544)
July 2021	1.0	Initial release.