易 灵 思

Sapphire RISC-V SoC Data Sheet

DS-SAPPHIRE-v5.1
January 2025
www.elitestek.com

Contents

Introduction

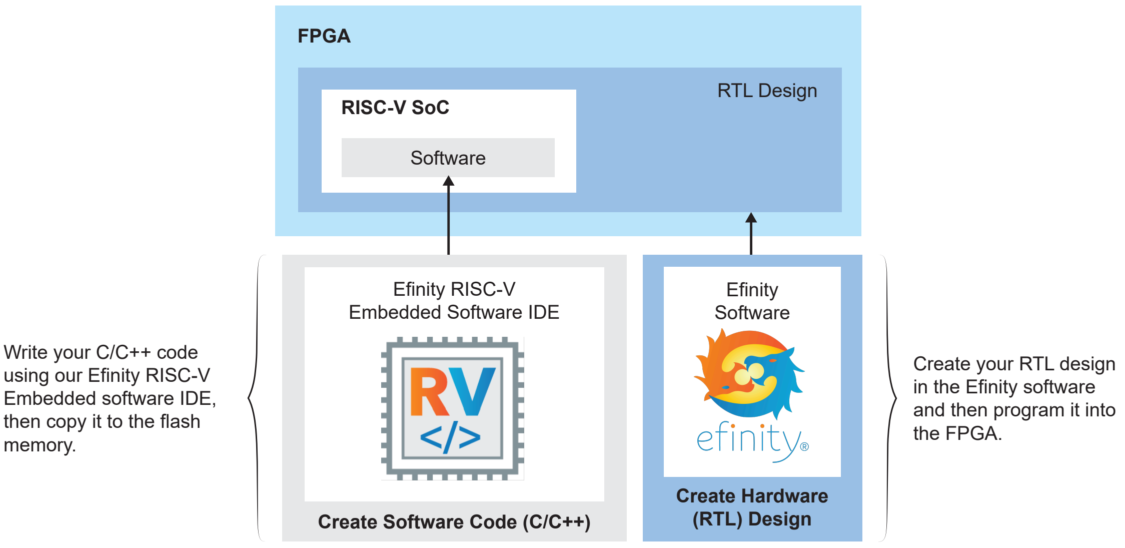
Elitestek provides the configurable, cached soft RISC-V SoC, Sapphire, which optionally includes a memory controller interface. The Sapphire SoC supports a variety of peripherals. You can choose which peripherals you want by configuring the SoC in the Efinity® IP Manager. This core is similar to the open-source SaxonSOC, but it has been optimized for TJ-Series 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


FPGA
RTL Design
RISC-V SoC
Software
Write your C/C++ code using our Efinity RISC-V Embedded software IDE, then copy it to the flash memory.
Efinity RISC-V Embedded Software IDE
RV </>
Create Software Code (C/C++)
Efinity Software
efinity
Create Hardware (RTL) Design
Create your RTL design in the Efinity software and then program it into the FPGA.

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 Sapphire SoC VexRiscv core uses Little-Endian for its memory storage.

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

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

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 DDR3, LPDDR4x 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 (up to 512-bits) or 1 full-duplex AXI4 interface (up to 512-bits) to communicate with the external memory
  - 400 MHz DDR3 clock frequency, 800 Mbps
  - 1089 MHz LPDDR4x clock frequency, 2178 Mbps
  - 250 MHz HyperRAM clock frequency, 500 Mbps
- Up to 2 AXI master channels for user logic, data widths from 32 to 512
- 1 AXI slave channel to user logic
- Includes an optional multi-way instruction and Data Cache
- Includes a floating point unit (FPU)
- Includes an optional Linux memory management unit (MMU)
- Includes an optional custom instruction interface with 1,024 IDs to perform various functions
- Supports optional RISC-V extensions such as atomic and compressed
- APB3 peripherals:
  - Up to 32 GPIOs
  - Up to 3 I^{2}C masters
  - Clint timer
  - Platform-level interrupt controller (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 TJ-Series 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.

TJ-Series Resource Utilization and Performance

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

Table 2: Cacheless SoC with External Memory

Standard option

FPGA	Logic/Adders	FlipFlops	Memory Blocks	DSP48 Blocks	f_{MAX} (MHz)	Efinity Version
Ti60 F225 C4 (typical)	7,035	7,670	44	4	369	2024.2
Ti60 F225 C4 (custom instruction)	7,099	7,716	44	4	388	2024.2

Table 3: Cacheless SoC without External Memory

Standard option

FPGA	Logic/Adders	FlipFlops	Memory Blocks	DSP48 Blocks	f_{MAX} (MHz)	Efinity Version
Ti60 F225 C4 (typical)	4,492	3,110	12	4	386	2024.2
Ti60 F225 C4 (custom instruction)	4,502	3,151	12	4	398	2024.2

Table 4: Cached SoC with External Memory

Standard option

FPGA	Logic/Adders	FlipFlops	Memory Blocks	DSP48 Blocks	f_{MAX} (MHz)	Efinity Version
Ti60 F225 C4 (typical)	7,507	8,110	56	4	427	2024.2
Ti60 F225 C4 (custom instruction)	7,717	8,161	56	4	396	2024.2
Ti60 F225 C4 (FPU)	14,059	12,230	77	13	308	2024.2
Ti60 F225 C4 (2 cores)	14,108	13,181	103	8	328	2024.2
Ti60 F225 C4 (3 cores)	18,328	15,521	127	12	310	2024.2
Ti60 F225 C4 (4 cores)	22,561	17,784	150	16	285	2024.2

Table 5: Cached SoC without External Memory

Standard option

FPGA	Logic/Adders	FlipFlops	Memory Blocks	DSP48 Blocks	f_{MAX} (MHz)	Efinity Version
Ti60 F225 C4 (typical)	4,883	3,529	24	4	400	2024.2
Ti60 F225 C4 (custom instruction)	5,101	3,569	24	4	414	2024.2
Ti60 F225 C4 (FPU)	11,522	7,677	44	13	294	2024.2
Ti60 F225 C4 (2 cores)	11,365	8,302	62	8	334	2024.2
Ti60 F225 C4 (3 cores)	15,642	10,630	87	12	330	2024.2
Ti60 F225 C4 (4 cores)	19,589	12,888	110	16	309	2024.2

Table 6: Cacheless SoC

Lite option

FPGA	Logic/Adders	FlipFlops	Memory Blocks	DSP48 Blocks	f_{MAX} (MHz)	Efinity Version
Ti60 F225 C4 (external memory)	3,297	2,789	14	0	386	2024.2
Ti60 F225 C4 (internal memory)	2,732	1,949	24	0	388	2024.2

Table 7: Cached SoC

Lite option

FPGA	Logic/Adders	FlipFlops	Memory Blocks	DSP48 Blocks	f_{MAX} (MHz)	Efinity Version
Ti60 F225 C4 (external memory)	3,837	2,978	26	0	373	2024.2
Ti60 F225 C4 (internal memory)	3,103	2,124	36	0	396	2024.2

Trion Resource Utilization and Performance

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

Table 8: Cacheless SoC with External Memory

Standard option

FPGA	Logic/Adders	FlipFlops	Memory Blocks	DSP Blocks	f_{MAX} (MHz)	Efinity Version
T120 F324 (typical)	7,144	7,814	48	4	111	2024.2
T120 F324 (custom instruction)	7,098	7,865	48	4	111	2024.2

Table 9: Cacheless SoC without External Memory

Standard option

FPGA	Logic/Adders	FlipFlops	Memory Blocks	DSP Blocks	f_{MAX} (MHz)	Efinity Version
T120 F324 (typical)	4,505	3,257	16	4	114	2024.2
T120 F324 (custom instruction)	4,629	3,300	16	4	116	2024.2

Table 10: Cached SoC with External Memory

Standard option

FPGA	Logic/Adders	FlipFlops	Memory Blocks	DSP Blocks	f_{MAX} (MHz)	Efinity Version
T120 F324 (typical)	7,501	8,258	67	4	116	2024.2
T120 F324 (custom instruction)	7,600	8,300	67	4	119	2024.2
T120 F324 (FPU)	14,556	12,599	80	13	84	2024.2
T120 F324 (2 cores)	14,137	13,456	109	8	92	2024.2
T120 F324 (3 cores)	18,575	15,928	136	12	90	2024.2
T120 F324 (4 cores)	22,917	18,334	162	16	91	2024.2

Table 11: Cached SoC without External Memory

Standard option

FPGA	Logic /Adders	FlipFlops	Memory Blocks	DSP Blocks	f_{MAX} (MHz)	Efinity Version
T120 F324 (typical)	4,882	3,676	35	4	113	2024.2
T120 F324 (custom instruction)	5,135	3,719	35	4	114	2024.2
T120 F324 (FPU)	11,947	8,038	47	13	87	2024.2
T120 F324 (2 cores)	11,579	8,575	68	8	106	2024.2
T120 F324 (3 cores)	16,038	11,035	96	12	98	2024.2
T120 F324 (4 cores)	19,867	13,425	122	16	90	2024.2

Table 12: Cacheless SoC

Lite option

FPGA	Logic/Adders	FlipFlops	Memory Blocks	DSP Blocks	f_{MAX} (MHz)	Efinity Version
T120 F324 (external memory)	3,299	2,805	18	0	112	2024.2
T120 F324 (internal memory)	2,759	1,964	40	0	115	2024.2

Table 13: Cached SoC

Lite option

FPGA	Logic/Adders	FlipFlops	Memory Blocks	DSP Blocks	f_{MAX} (MHz)	Efinity Version
T120 F324 (external memory)	3,754	3,025	37	0	118	2024.2
T120 F324 (internal memory)	3,102	2,170	59	0	121	2024.2

TP-Series Resource Utilization and Performance

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

Table 14: Cacheless SoC with External Memory

Standard option

FPGA	Logic/Adders	FlipFlops	Memory Blocks	DSP Blocks	f_{MAX} (MHz)	Efinity Version
TP110 J484 C3 (typical)	7,035	7,670	44	4	257	2024.2
TP110 J484 C3 (custom instruction)	7,099	7,716	44	4	260	2024.2

Table 15: Cacheless SoC without External Memory

Standard option

FPGA	Logic/Adders	FlipFlops	Memory Blocks	DSP Blocks	f_{MAX} (MHz)	Efinity Version
TP110 J484 C3 (typical)	4,492	3,110	12	4	266	2024.2
TP110 J484 C3 (custom instruction)	4,502	3,151	12	4	281	2024.2

Table 16: Cached SoC with External Memory

Standard option

FPGA	Logic/Adders	FlipFlops	Memory Blocks	DSP Blocks	f_{MAX} (MHz)	Efinity Version
TP110 J484 C3 (typical)	7,507	8,110	56	4	280	2024.2
TP110 J484 C3 (custom instruction)	7,717	8,161	56	4	282	2024.2
TP110 J484 C3 (FPU)	14,059	12,230	77	13	199	2024.2
TP110 J484 C3 (2 cores)	14,108	13,181	103	8	206	2024.2
TP110 J484 C3 (3 cores)	18,328	15,521	127	12	211	2024.2
TP110 J484 C3 (4 cores)	22,561	17,784	150	16	215	2024.2

Table 17: Cached SoC without External Memory

Standard option

FPGA	Logic /Adders	FlipFlops	Memory Blocks	DSP Blocks	f_{MAX} (MHz)	Efinity Version
TP110J484 C3 (typical)	4,883	3,529	24	4	271	2024.2
TP110 J484 C3 (custom instruction)	5,101	3,569	24	4	285	2024.2
TP110 J484 C3 (FPU)	11,522	7,677	44	13	195	2024.2
TP110 J484 C3 (2 cores)	11,365	8,302	62	8	224	2024.2
TP110 J484 C3 (3 cores)	15,642	10,630	87	12	219	2024.2

FPGA	Logic /Adders	FlipFlops	Memory Blocks	DSP Blocks	f_{MAX} (MHz)	Efinity Version
TP110 J484 C3 (4 cores)	19,589	12,888	110	16	216	2024.2

Table 18: Cacheless SoC

Lite option

FPGA	Logic/Adders	FlipFlops	Memory Blocks	DSP Blocks	f_{MAX} (MHz)	Efinity Version
TP110 J484 C3 (external memory)	3,297	2,789	14	0	253	2024.2
TP110 J484 C3 (internal memory)	2,732	1,949	24	0	277	2024.2

Table 19: Cached SoC

Lite option

FPGA	Logic/Adders	FlipFlops	Memory Blocks	DSP Blocks	f_{MAX} (MHz)	Efinity Version
TP110 J484 C3 (external memory)	3,837	2.978	26	0	273	2024.2
TP110 J484 C3 (internal memory)	3,103	2,124	36	0	273	2024.2

Performance Benchmark

The performance of the CPU can be benchmarked with Dhrystone and Coremark benchmark programs for easier comparison between processors. While the Sapphire SoC uses the same VexRiscv core, the configurations such as memory, cache, and additional extensions would affect the performance of the system. The Dhrystone and CoreMark scores for each of the configurations are tabulated in the table based on Trion T120 F324 Development Board.

Development board: Trion T120 F324

Efinity version 2024.1

Table 20: Performance Benchmark 1

Constants: Frequency @ 50 MHz with External Memory (DDR3), Legend: 1 = Enabled; 0 = Disabled

Option	Cache	Instruction and Data Way (kB)	Instruction and Data Size (kB)	FPU	Atomic	Compressed	Coremark (/MHz)	Dhrystone (/MHz)
Standard	1	1	4	0	0	0	1.74	0.89
Standard	1	4	4	0	0	0	1.85	1.05
Standard	1	8	32	0	0	0	1.90	1.05
Standard	1	1	4	1	0	0	1.77	0.90
Standard	1	4	4	0	1	1	1.84	1.03
Standard	1	4	4	1	0	0	1.86	1.05
Standard	0	1	4	0	0	0	0.23	0.12
Lite	1	1	4	0	0	0	0.61	0.60
Lite	0	1	4	0	0	0	0.13	0.14

Table 21: Performance Benchmark 2

Constants: Frequency @ 50 MHz with Internal Memory, Legend: 1 = Enabled; 0 = Disabled

Option	Cache	Instruction and Data Way (kB)	Instruction and Data Size (kB)	FPU	Atomic	Compressed	Coremark (/MHz)	Dhrystone (/MHz)
Standard	1	1	4	0	0	0	1.81	0.97
Standard	1	4	4	0	0	0	1.88	1.05
Standard	1	8	32	0	0	0	1.90	1.05
Standard	1	1	4	1	0	0	1.84	0.99
Standard	1	4	4	0	1	1	1.82	1.03
Standard	1	4	4	1	0	0	1.89	1.06
Standard	0	1	4	0	0	0	1.69	0.86
Lite	1	1	4	0	0	0	0.62	0.72
Lite	0	1	4	0	0	0	0.60	0.75

Functional Description

The Sapphire 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 1 - 32 KB, a data cache with up to 8 ways and a configurable size of 1 - 32 KB, 1 - 512 KB of on-chip RAM, and a variety of peripherals (including 1 - 5 APB3 slave peripherals and 1 AXI slave). You can configure the operating frequency from 20 - 400 MHz (the actual performance is limited by the design's f_{MAX}). The SoC includes 1 - 3 I^2C peripherals, 1 - 3 UARTs, 1 - 3 user timers, 1 - 8 user interrupts, and 1 - 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 and full-duplex AXI bus to communicate with the Elitestek LPDDR4x 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: Sapphire SoC Block Diagram

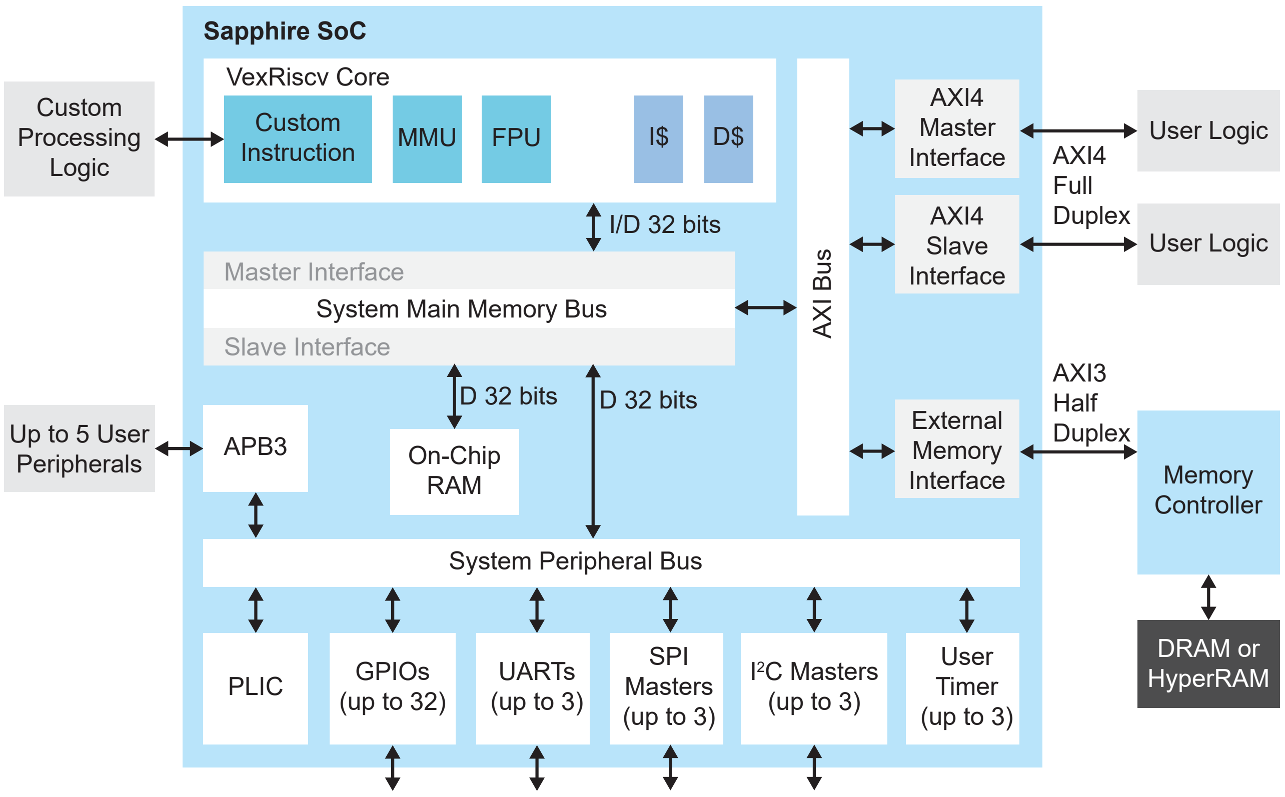
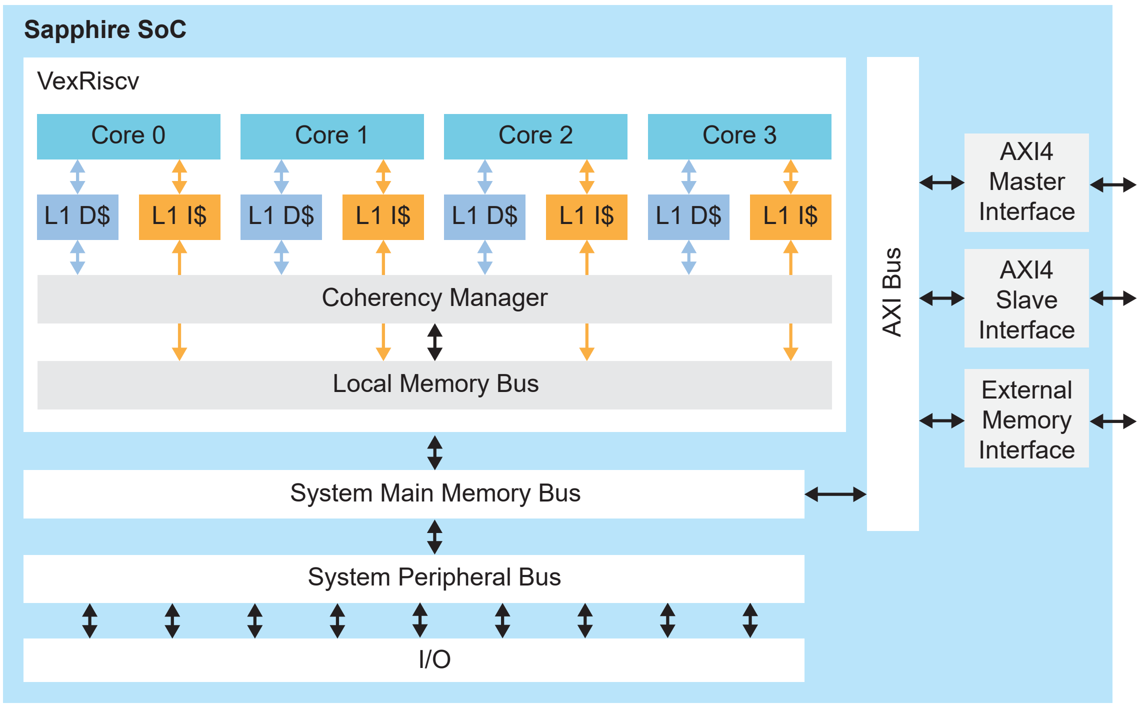


Figure 3: Sapphire SoC with Multiple Cores Block Diagram



Address Map

Because the address range might be updated, Elitestek 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 22: 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

When accessing the addresses in the I/O region, type casting the pointer with the keyword **`volatile`**. The compiler recognizes this as a memory-mapped I/O register without optimizing the read/write access. An example of the casting is shown by the following command:

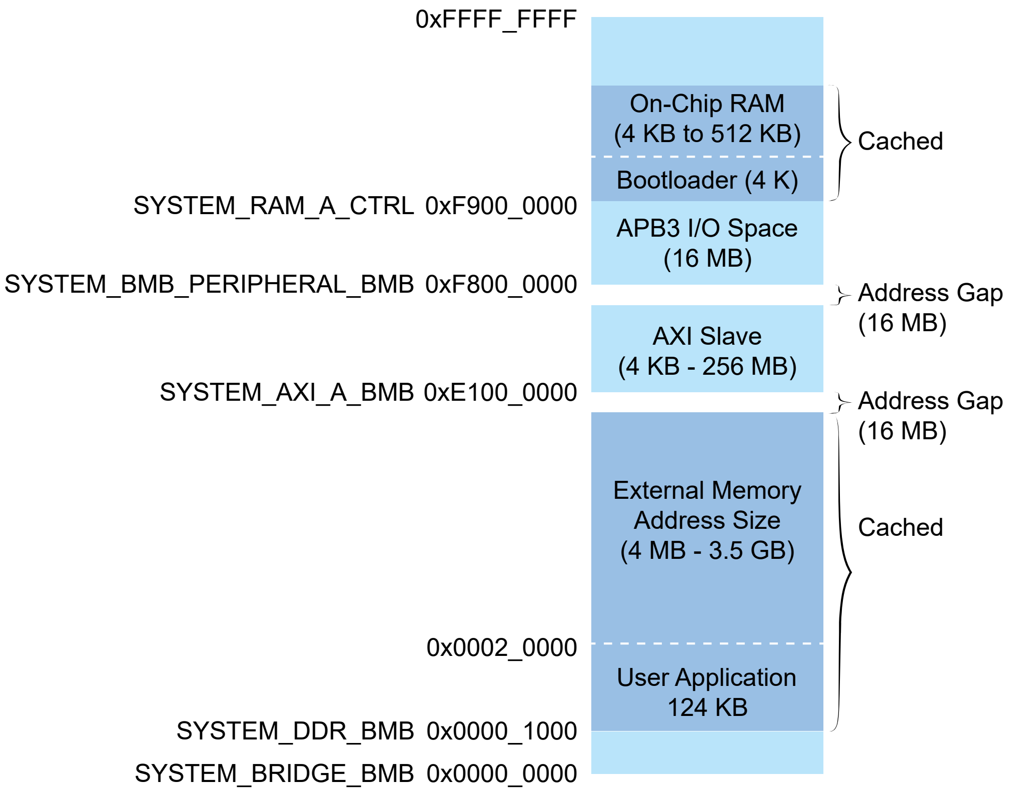
```
*((volatile u32*) address);
```

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.

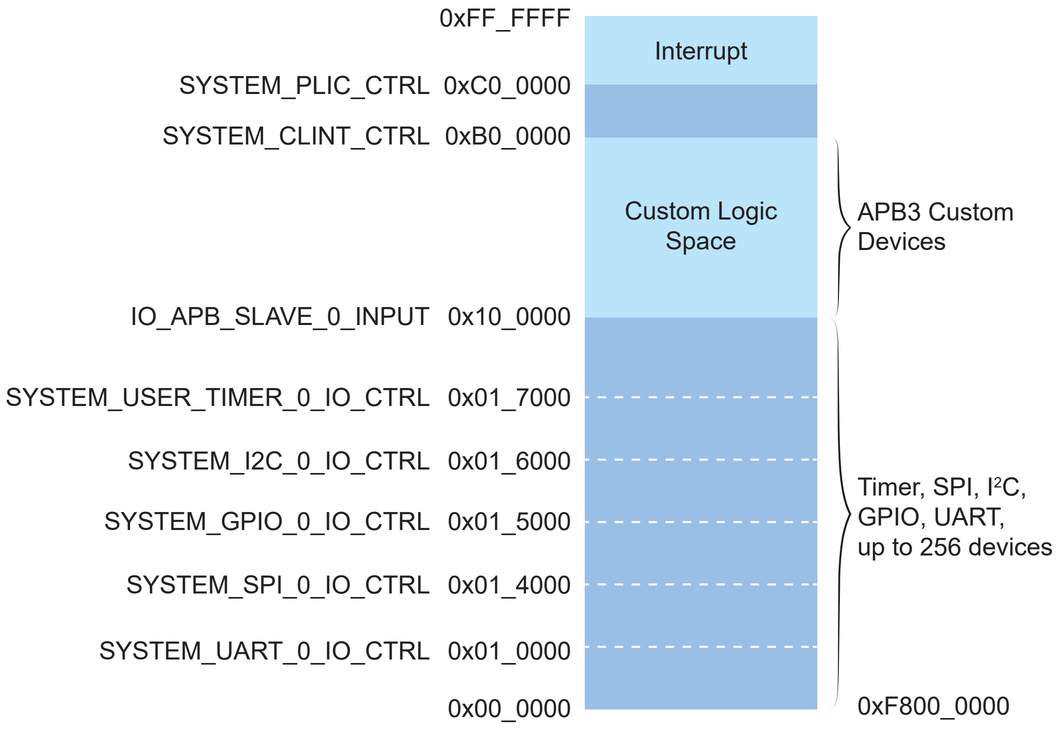
The following figure shows the default address map and the corresponding software parameters for modules in the memory space.

Figure 4: Sapphire Memory Space


0xFFFF_FFFF
On-Chip RAM (4 KB to 512 KB)
Bootloader (4 K)
Cached
SYSTEM_RAM_A_CTRL 0xF900_0000
APB3 I/O Space (16 MB)
SYSTEM_BMB_PERIPHERAL_BMB 0xF800_0000
Address Gap (16 MB)
AXI Slave (4 KB - 256 MB)
SYSTEM_AXI_A_BMB 0xE100_0000
Address Gap (16 MB)
External Memory Address Size (4 MB - 3.5 GB)
Cached
0x0002_0000
User Application 124 KB
SYSTEM_DDR_BMB 0x0000_1000
SYSTEM_BRIDGE_BMB 0x0000_0000

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

Figure 5: Sapphire I/O Space


0xFF_FFFF
Interrupt
SYSTEM_PLIC_CTRL 0xC0_0000
SYSTEM_CLINT_CTRL 0xB0_0000
Custom Logic Space
APB3 Custom Devices
IO_APB_SLAVE_0_INPUT 0x10_0000
SYSTEM_USER_TIMER_0_IO_CTRL 0x01_7000
SYSTEM_I2C_0_IO_CTRL 0x01_6000
SYSTEM_GPIO_0_IO_CTRL 0x01_5000
SYSTEM_SPI_0_IO_CTRL 0x01_4000
SYSTEM_UART_0_IO_CTRL 0x01_0000
Timer, SPI, I²C, GPIO, UART, up to 256 devices
0x00_0000
0xF800_0000

Default I/O address offset: 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 23: 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 24: 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 25: 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 26: 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.
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.

Port	Direction	Description
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 27: 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 28: 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 29: 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 30: 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 31: 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 32: 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 33: 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.

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

Table 34: User Slave Read Address Channel

Port	Direction	Description
axiA_arvalid	Output	User read address valid.
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 35: 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 36: 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.

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

AXI Master Interface

Table 37: 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.
io_ddrMasters_*n*_aw_payload_prot[2:0]	Input	User protection type.

Table 38: 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[*m*: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 39: 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 40: 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.

Port	Direction	Description
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 41: 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[*m*: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.
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 42: 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 43: 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

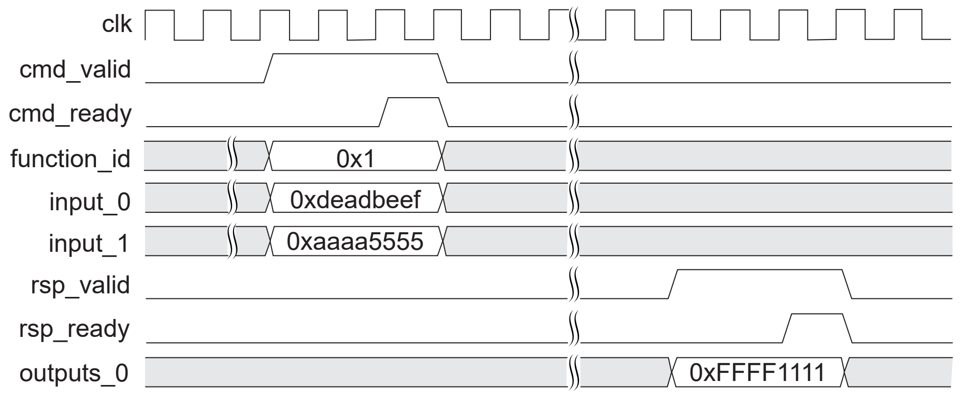
The Sapphire SoC supports a custom instruction interface so you can accelerate software functions with custom hardware logic. The custom instruction supports R-type instructions, which provides two registers (`rs1` and `rs2`) to custom instruction processing logic and up to 1,024 IDs to perform different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

Table 44: Custom Instruction Ports

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

Port	Direction	Description
cpu*n*_customInstruction_cmd_valid	Output	Indicates that registers rs1 and rs2 are present and ready for processing.
cpu*n*_customInstruction_cmd_ready	Input	Indicates that the custom processing logic is ready to process register rs1 and rs2 from the CPU.
cpu*n*_customInstruction_function_id[9:0]	Output	Function id for the custom instruction.
cpu*n*_customInstruction_inputs_0[31:0]	Output	Register rs1 for the custom instruction.
cpu*n*_customInstruction_inputs_1[31:0]	Output	Register rs2 for the custom instruction.
cpu*n*_customInstruction_rsp_valid	Input	Indicates that the custom instruction result is available.
cpu*n*_customInstruction_rsp_ready	Output	Indicates that the CPU is ready to accept the custom instruction result.
cpu*n*_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 45: 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 46: 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 32 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 32 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 32 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. 1'b1: Enable GPIO rising edge interrupt 1'b0: Disable GPIO rising edge interrupt	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. 1'b1: Enable GPIO falling edge interrupt 1'b0: Disable GPIO falling edge interrupt	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. 1'b1: Enable GPIO high level interrupt 1'b0: Disable GPIO high level interrupt	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. 1'b1: Enable GPIO low level interrupt 1'b0: Disable GPIO low level interrupt	Read/Write
2-31	Reserved	Reserved.	N/A

Watchdog Timer Interface

Use the `SYSTEM_WATCHDOG_LOGIC_CTRL` parameter to reference the watchdog timer.

Table 47: Watchdog Timer I/O Ports

Port	Direction	Description
system_watchdog_hardPanic	Output	Indicates that the watchdog timer counter 1 has reached its limit. Active high. 0: Counter 1 has not reached its limit. 1: Counter 1 has reached its limit.

Table 48: Watchdog Timer Register Map

Address Offset	Register Name	Privilege	Width
0x0000_0000	WATCHDOG_HEARTBEAT	Write	32
0x0000_0004	WATCHDOG_ENABLE	Write	32
0x0000_0040	WATCHDOG_PRESCALER	Write	32
0x0000_0080	WATCHDOG_COUNTER_LIMIT 0	Write	32
0x0000_0084	WATCHDOG_COUNTER_LIMIT 1	Write	32
0x0000_00C0	WATCHDOG_COUNTER_VALUE 0	Read	32
0x0000_00C4	WATCHDOG_COUNTER_VALUE 1	Read	32

Heartbeat Register: 0x0000_0000

31 ... 0
Heartbeat value

Bits	Field	Description	Privilege
0-31	Heartbeat value	Value for the timer to generate a detection pulse. Write 0xAD68E70D into this register to reset the watchdog timer.	Write

Enable Register: 0x0000_0004

31 ... 2	1	0
Reserved	Enable Counter 1	Enable Counter 0

Bits	Field	Description	Privilege
0	Enable Counter 0	1'b1: Enable for counter 0 1'b0: Disable for counter 0	Write
1	Enable Counter 1	1'b1: Enable for counter 1 1'b0: Disable for counter 1	Write
2-31	Reserved	Reserved.	N/A

Prescaler Register: 0x0000_0040

31 ... 16	15 ... 0
Reserved	Prescaler value

Bits	Field	Description	Privilege
0-15	Prescaler value	The clock divider ratio. Example: 16'd0: divide by 1 16'd1: divide by 2 ... 16'd65534: divide by 65535 16'd65535: divide by 65536	Write
16-31	Reserved	Reserved.	-

Counter Limit 0 Register: 0x0000_0080

31 ... 0
Limit 0 value

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

Counter Limit 1 Register: 0x0000_0084

31 ... 0
Limit 1 value

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

Counter Value 0 Register: 0x0000_00C0

31	0
Value 0	

Bits	Field	Description	Privilege
0-31	Value 0	Value of the increment counter 0.	Read

Counter Value 1 Register: 0x0000_00C4

31	0
Value 1	

Bits	Field	Description	Privilege
0-31	Value 1	Value of the increment counter 1.	Read

I^{2}C Peripheral Interface

The Sapphire SoC has up to 3 I^{2}C 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 49: I^{2}C Peripheral Ports (User)

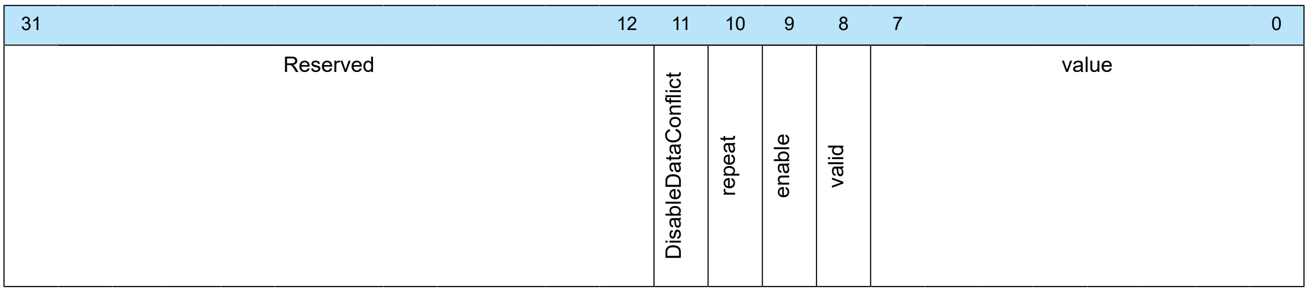
Where *n* is 0, 1, or 2.

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 50: I^{2}C Register Map

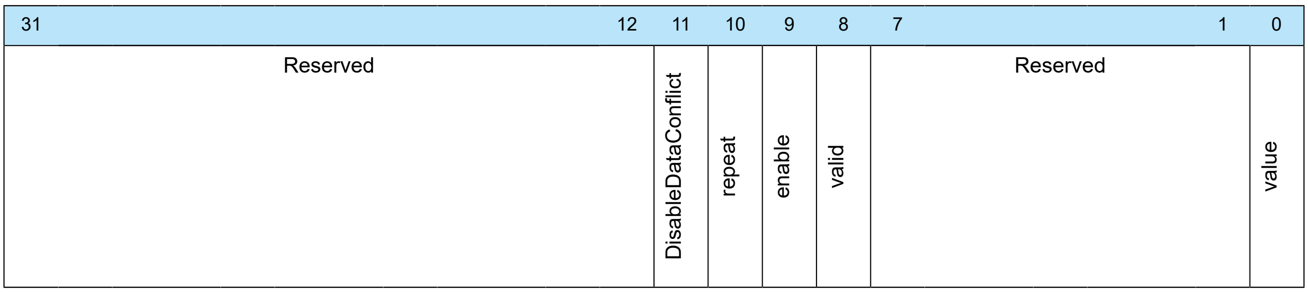
Address Offset	Register Name	Privilege	Width
0x0000_0000	txData	Read/Write	32
0x0000_0004	txAck	Read/Write	32
0x0000_0008	rxData	Read/Write	32
0x0000_000C	rxAck	Read/Write	32
0x0000_0020	Interrupt	Read/Write	32
0x0000_0024	Interrupt Clears	Read/Write	32
0x0000_0028	Sampling Clock Divider	Write	32
0x0000_002C	Timeout	Write	32
0x0000_0030	tsuData	Write	32
0x0000_0040	Master Status	Read/Write	32
0x0000_0044	Slave Status	Read	32
0x0000_0048	Slave Override	Read/Write	32
0x0000_0050	tlow	Write	32
0x0000_0054	tHigh	Write	32
0x0000_0058	tBuf	Write	32
0x0000_0080	Hit Context	Read	32
0x0000_0084	Filtering Status	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	Data field that stores transmitted data.	Write
8	valid	Write 1'b1 to indicate valid data for available transmission in the data field.	Read/Write
9	enable	Write 1'b1 to enable data transmission operation.	Read/Write
10	repeat	Write 1'b1 to enable data transmission operation in repeat mode.	Write
11	DisableDataConflict	Write 1'b1 to stop data transmission if your address negotiation had conflicted with another master.	Write
12-31	Reserved	Reserved.	N/A

txAck Register: 0x0000_0004



Bits	Field	Description	Privilege
0	value	Transmit acknowledge bit. 1'b1: Nack 1'b0: Ack	Write
1-7	Reserved	Reserved.	N/A
8	valid	Write 1'b1 to indicate available valid acknowledge bit to transmit.	Read/Write
9	enable	Write 1'b1 to enable the transmit operation acknowledge bit.	Read/Write
10	repeat	Write 1'b1 to enable the transmit operation acknowledge bit in repeat mode.	Write
11	DisableDataConflict	Write 1'b1 to stop the transmit acknowledge bit if your negotiation address is in conflict with another master.	Write
12-31	Reserved	Reserved.	N/A

rxData Register: 0x0000_0008

31 10	9	8	7 0
Reserved	listen	valid	value

Bits	Field	Description	Privilege
0-7	value	Data field that stores received data.	Read
8	valid	Indicates the received data is ready.	Read
9	listen	Write 1'b1 to start listening data from respondents.	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	Acknowledge bit to the respondent. 1'b1: Nack 1'b0: Ack	Read
1-7	Reserved	Reserved.	N/A
8	valid	Indicates received acknowledge bit is ready.	Read
9	listen	Write 1'b1 to start listening to the acknowledge bit from respondent.	Write
10-31	Reserved	Reserved.	N/A

Interrupt Register: 0x0000_0020

31 18	17	16	15	14 8	7	6	5	4	3	2	1	0
Reserved	filterEnable	clockGenEnterEnable	clockGenExitEnable	Reserved	dropEnable	endEnable	restartEnable	startEnable	txAckEnable	txDataEnable	rxAckEnable	rxDataEnable

Bits	Field	Description	Privilege
0	rxDataEnable	Write 1'b1 to enable interrupt when valid data is received.	Read/Write
1	rxAckEnable	Write 1'b1 to enable interrupt when valid acknowledge bit is received.	Read/Write
2	txDataEnable	Write 1'b1 to enable interrupt when valid data is transmitted.	Read/Write
3	txAckEnable	Write 1'b1 to enable interrupt when valid acknowledge bit is transmitted.	Read/Write
4	startEnable	Write 1'b1 to enable interrupt when a transfer is in START state.	Read/Write
5	restartEnable	Write 1'b1 to enable interrupt when a transfer is restarted.	Read/Write
6	endEnable	Write 1'b1 to enable interrupt when a transfer is in STOP state.	Read/Write
7	dropEnable	Write 1'b1 to enable interrupt when a transfer is dropped due to confilct or timeout.	Read/Write
8-14	Reserved	Reserved.	N/A
15	clockGenExitEnable	Write 1'b1 to enable interrupt when the controller stops generating clock output.	Read/Write
16	clockGenEnterEnable	Write 1'b1 to enable interrupt when the controller starts generating clock input.	Read/Write
17	filterEnable	Write 1'b1 to enable interrupt when the controller acts as slave and its address filter is triggered.	Read/Write
18-31	Reserved	Reserved.	N/A

Interrupt Clears Register: 0x0000_0024

31 18	17	16	15	14 8	7	6	5	4	3 0
Reserved	filterFlag	clockGenEnterFlag	clockGenExitFlag	Reserved	dropFlag	endFlag	restartFlag	startFlag	Reserved

Bits	Field	Description	Privilege
0-3	Reserved	Reserved.	N/A
4	startFlag	Start interrupt status. 1'b1: A transfer is in START state Write 1'b1 to clear the flag.	Read/Write
5	restartFlag	Restart interrupt status. 1'b1: A transfer is restarted Write 1'b1 to clear the flag.	Read/Write
6	endFlag	End interrupt status. 1'b1: A transfer is in STOP state Write 1'b1 to clear the flag.	Read/Write
7	dropFlag	Drop interrupt status. 1'b1: A transfer is dropped Write 1'b1 to clear the flag.	Read/Write
8 - 14	Reserved	Reserved.	N/A
15	clockGenExitFlag	Master clock generation exit interrupt status. 1'b1: Controller stops generating output clock. Write 1'b1 to clear the flag.	Read/Write
16	clockGenEnterFlag	Master clock generation enterinterrupt status. 1'b1: Controller starts generating output clock. Write 1'b1 to clear the flag.	Read/Write
17	filterFlag	Address filter interrupt status. 1'b1: Controller as slave and its address filter is triggered Write 1'b1 to clear the flag.	Read/Write
18-31	Reserved	Reserved.	N/A

Sampling Clock Divider Register: 0x0000_0028

31 10	9 0
Reserved	samplingClockDivider

Bits	Field	Description	Privilege
0-9	samplingClockDivider	Sampling rate = (FCLK/(samplingClockDivider + 1) Controls the rate at which the I2C controller samples SCL and SDA. 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
10-31	Reserved	Reserved.	N/A

Timeout Register: 0x0000_002C

31 … 20	19 … 0
Reserved	value

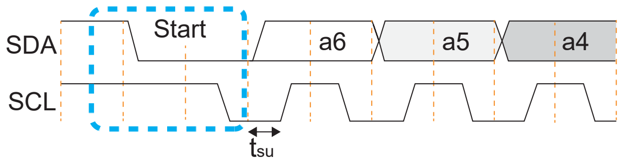
Bits	Field	Description	Privilege
0-19	value	Inactive timeout clock cycle. The controller will drop the transfer when the value of the timeout is reached or exceeded. Setting the timeout value to zero will disable the timeout feature. The clock cycle refers to FCLK.	Write
20-31	Reserved	Reserved.	N/A

tsuData Register: 0x0000_0030

31 … 6	5 … 0
Reserved	value

Bits	Field	Description	Privilege
0-5	value	Data setup time. The number of clock cycles should SDA hold its state before the rising edge of SCL. The clock cycle refers to FCLK.	Write
6-31	Reserved	Reserved.	N/A

Figure 7: tsuData Register Waveform



Master Status Register: 0x0000_0040

31	12	11	10	9	8	7	6	5	4	3	1	0
Reserved		recoverDropped	stopDropped	startDropped	Reserved	Recover	drop	stop	start	Reserved		isBusy

Bits	Field	Description	Privilege
0	isBusy	The busy status of the controller. 1'b1: Busy 1'b0: Idle	Read
1-3	Reserved	Reserved.	N/A
4	start	Master issues the START bit.	Read/Write
5	stop	Master issues the STOP bit.	Read/Write
6	drop	Master drops the transfer due to address confilct.	Read/Write
7	Recover	Master issues recover sequence by toggling the SCL line multiple times to recover the slave. 1'b1: Trigger recover sequence. 1'b0: Idle	Read/Write
8	Reserved	Reserved	N/A
9	startDropped	Indicates timeout occurred during the start transaction. 1'b1: Timeout occurred. 1'b0: No timeout occurred. Write 1'b1 to reset this flag.	Read/Write
10	stopDropped	Indicates timeout occurred during the stop transaction. 1'b1: Timeout occurred. 1'b0: No timeout occurred. Write 1'b1 to reset this flag.	Read/Write
11	recoverDropped	Indicates timeout occurred during the recover transaction. 1'b1: Timeout occurred. 1'b0: No timeout occurred. Write 1'b1 to reset this flag.	Read/Write
12-31	Reserved	Reserved	N/A

Slave Status Register: 0x0000_0044

31 … 3	2	1	0
Reserved	Scl	Sda	inFrame

Bits	Field	Description	Privilege
0	inFrame	Indicates whether the controller is inFrame, where the transaction is in progress. The controller is expected to be inFrame after a successful start is sent out. the controller would exit the inFrame after a successful stop is sent out. 1'b1: InFrame 1'b0: Idle	Read
1	Sda	Current state of SDA line. 1'b1: High state 1'b0: Low state	Read
2	Scl	Current state of SCL line. 1'b1: High state 1'b0: Low state	Read
3-31	Reserved	Reserved	N/A

Slave Override Register: 0x0000_0048

31 … 3	2	1	0
Reserved	Scl	Sda	Reserved

Bits	Field	Description	Privilege
0	Reserved	Reserved	N/A
1	Sda	Force SDA line. 1'b1: release SDA bus 1'b0: Force SDA bus to pull low	Read/Write
2	Scl	Force SCL line. 1'b1: Release SCL bus 1'b0: Force SCL bus to pull low	Read/Write
3-31	Reserved	Reserved	N/A

tLow Register: 0x0000_0050

31 … 12	11 … 0
Reserved	value

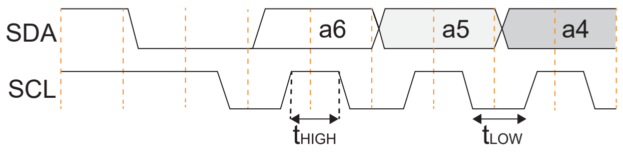
Bits	Field	Description	Privilege
0-11	value	The number of clock cycles of SCL in LOW state. The clock cycle refers to FCLK.	Write
12-31	Reserved	Reserved.	N/A

tHigh Register: 0x0000_0054

31 12	11 0
Reserved	value

Bits	Field	Description	Privilege
0-11	value	The number of clock cycles of SCL in HIGH state. The clock cycle refers to FCLK.	Write
12-31	Reserved	Reserved.	N/A

Figure 8: tHigh and tLow Register Waveform

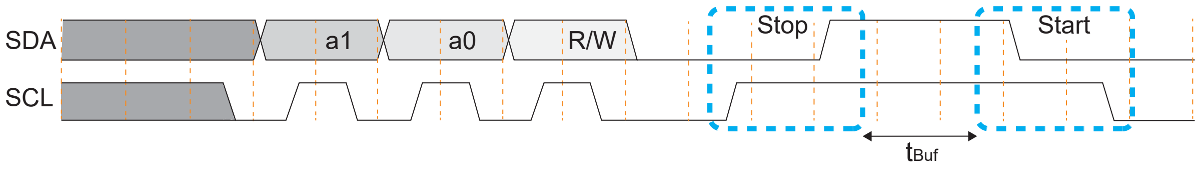


tBuf Register: 0x0000_0058

31 12	11 0
Reserved	value

Bits	Field	Description	Privilege
0-11	value	The number of clock cycles delay before master can initiate a START bit after a STOP bit is issued. The clock cycle refers to FCLK.	Write
12-31	Reserved	Reserved.	N/A

Figure 9: tBuf Register Waveform



Hit Context Register: 0x0000_0080

31 2	1	0
Reserved	hit_1	hit_0

Bits	Field	Description	Privilege
0	hit_0	Address filter 0 status. 1'b1: Request address from master which is matched using the value set by **Filtering Configuration 0 Register**	Read
1	hit_1	Address filter 1 status. 1'b1: Request address from master which is matched using the value set by **Filtering Configuration 1 Register**	Read
2-31	Reserved	Reserved.	N/A

Filtering Status Register: 0x0000_0084

31	1	0
Reserved		rw

Bits	Field	Description	Privilege
0	rw	Read/Write attributes of requested address from master. 1'b1: Read operation 1'b0: Write operation	Read
1-31	Reserved	Reserved.	N/A

Filtering Configuration 0 Register: 0x0000_0088

31	16	15	14	13	10	9	0
Reserved		enable	10 bit address	Reserved		value	

Bits	Field	Description	Privilege
0-9	value	Set target address to be filtered.	Write
10-13	Reserved	Reserved.	N/A
14	10 bit address	Set the target address bits width. 1'b1: 10 bits 1'b0: 7 bits	Write
15	enable	Enable address filter 0.	Write
16-31	Reserved	Reserved.	N/A

Filtering Configuration 1 Register: 0x0000_008C

31	16	15	14	13	10	9	0
		enable	10 bit address	Reserved		value	

Bits	Field	Description	Privilege
0-9	value	Set target address to be filtered.	Write
10-13	Reserved	Reserved.	N/A
14	10 bit address	Set the target address bits width. 1'b1: 10 bits 1'b0: 7 bits	Write
15	enable	Enable address filter 1.	Write
16-31	Reserved	Reserved.	N/A

PLIC Peripheral Interface

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

Table 51: 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 52: 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 34 **Interrupt Clears Register: 0x0000_0024** on page 35
SYSTEM_PLIC_SYSTEM_GPIO_*n*_IO_INTERRUPTS_0 *m*	**Interrupt Low Enable Register: 0x0000_002C** on page 27 **Interrupt High Enable Register: 0x0000_0028** on page 26 **Interrupt Fall Enable Register: 0x0000_0024** on page 26 **Interrupt Rise Enable Register: 0x0000_0020** on page 26
SYSTEM_PLIC_SYSTEM_AXI_A_INTERRUPT	**Interrupts** on page 16
SYSTEM_PLIC_SYSTEM_SPI_*n*_IO_INTERRUPT *m*	**Interrupt Register: 0x0000_000C** on page 44
SYSTEM_PLIC_SYSTEM_UART_*n*_IO_INTERRUPT *m*	**Status Register: 0x0000_0004** on page 49
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 16
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 53: 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 54: 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	cmd_writeLarge	Write	32
0x0000_0054	cmd_readWriteLarge	Write	32
0x0000_0058	cmd_readLarge	Read	32

Cmd Register: 0x0000_0000

31	30 12	11	10	9	8	7 0
Read Invalid	Reserved	SS	Reserved	RD	WR	data

Bits	Field	Description	Privilege
0-7	data	Data field that stores data that is transmitted or received, or chip select ID. For chip-select enablement, you should write 1'b1 to bit[7].	Read/Write
8	WR	Write 1'b1 to trigger a SPI write command. Data in the data field of this register will be transmitted.	Write
9	RD	Write 1'b1 to trigger a SPI read command. The read data will be stored in the data field of this register after the SPI read command is processed.	Write
10	Reserved	Reserved.	N/A
11	SS	Write 1'b1 to indicate the operation is for chip-select enablement.	Write
12-30	Reserved	Reserved.	N/A
31	Read invalid	Indicates whether the read data in the data field is invalid. 1: Read data is invalid 0: Read data is valid	Read

RSP Register: 0x0000_0004

31 25	24 16	15 9	8 0
Reserved	RspFifoOccupancy	Reserved	CmdFifoAvailability

Bits	Field	Description	Privilege
0-8	CmdFifoAvailability	Shows the number of operation command that is available in Cmd Fifo. The command includes write, read, and chip select operation.	Read
9-15	Reserved	Reserved.	N/A
16-24	RspFifoOccupancy	Shows the occupancy in RSP FIFO, indicates the number of valid data to be read.	Read
25-31	Reserved	Reserved.	N/A

Config Register: 0x0000_0008

31	6	5	4	3	2	1	0
Reserved		mode		Reserved		cpha	cpol

Bits	Field	Description	Privilege
0	cpol	Clock polarity during idle state. 1'b1: High 1'b0: Low	Write
1	cpha	Clock phase setting. 1'b1: Data sampled at falling edge 1'b0: Data sampled at rising edge	Write
2-3	Reserved	Reserved.	N/A
4-5	mode	0: Full-duplex dual line 1: Half-duplex dual line (Available only when data width is configured as 8 or 16) 2: Half-duplex quad line (Available only when data width is configured as 8 or 16)	Write
6-31	Reserved	Reserved.	N/A

Interrupt Register: 0x0000_000C

31	17	16	15	10	9	8	7	2	1	0
Reserved		cmdValid	Reserved		rspInt	cmdInt	Reserved		rspIntEnable	cmdIntEnable

Bits	Field	Description	Privilege
0	cmdIntEnable	Flag interrupt when Cmd FIFO is empty. 1'b1: Enable 1'b0: Disable	Read/Write
1	rspIntEnable	Flag interrupt when Rsp FIFO not empty interrupt enable. 1'b1: Enable 1'b0: Disable	Read/Write
2-7	Reserved	Reserved.	N/A
8	cmdInt	Cmd FIFO interrupt status. 1'b1: Interrupt flagged 1'b0: Interrupt not flagged	Read
9	rspInt	Rsp FIFO interrupt status. 1'b1: Interrupt flagged 1'b0: Interrupt not flagged	Read
10-15	Reserved	Reserved.	N/A
16	cmdValid	Cmd FIFO is not empty.	Read
17-31	Reserved	Reserved.	N/A

clockDivider Register: 0x0000_0020

31 12	11 0
Reserved	clockDivider

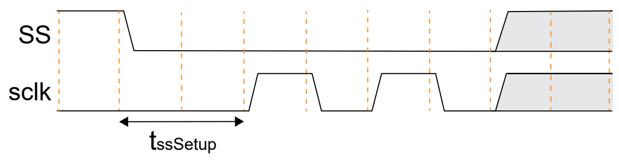
Bits	Field	Description	Privilege
0-11	clockDivider	SPI frequency = FCLK/((clockDivider+1)*2) FCLK is the system clock (io_systemClk) to the SoC. If you enable the peripheral clock, then FCLK is driven by the peripheral clock (io_peripheralClk) instead.	Write
12-31	Reserved	Reserved.	N/A

ssSetup Register: 0x0000_0024

31 12	11 0
Reserved	ssSetup

Bits	Field	Description	Privilege
0-11	ssSetup	Clock cycle between activated chip-select and first rising-edge of SCLK. Clock cycle refers to FCLK.	Write
12-31	Reserved	Reserved.	N/A

Figure 10: ssSetup Register Waveform

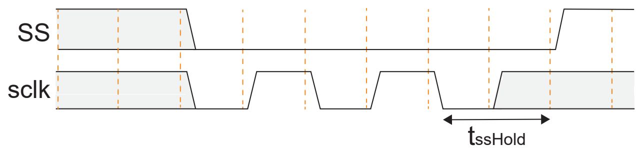


ssHold Register: 0x0000_0028

31 12	11 0
Reserved	ssHold

Bits	Field	Description	Privilege
0-11	ssHold	Clock cycle between last falling-edge and deactivated chip-select is activated. Clock cycle refers to FCLK.	Write
12-31	Reserved	Reserved.	N/A

Figure 11: ssHold Register Waveform

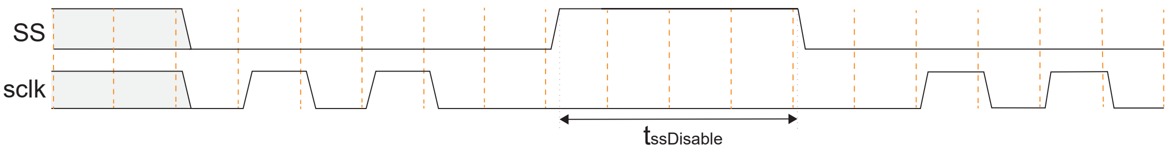


ssDisable Register: 0x0000_002C

31 12	11 0
Reserved	ssDisable

Bits	Field	Description	Privilege
0-11	ssDisable	Clock cycle delay refers to the system clock (io_systemClk) before the next chip select can be activated. If you enable the peripheral clock, the clock cycle will refer to the peripheral clock (io_peripheralClk).	Write
12-31	Reserved	Reserved.	N/A

Figure 12: ssDisable Register Waveform



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. 1'b1: Chip select is active high 1'b0: Chip select is active low	Write
8-31	Reserved	Reserved.	N/A

cmd_writeLarge Register: 0x0000_0050

31 16	15 0
Reserved	data

Bits	Field	Description	Privilege
0-15	data	Data field for write operation that is more than 8 data bits. The write trigger will be issued automatically when you write value to this register.	Write
16-31	Reserved	Reserved.	N/A

cmd_readWriteLarge Register: 0x0000_0054

31 16	15 0
Reserved	data

Bits	Field	Description	Privilege
0-15	data	Data field for write, then read operation that is more than 8 data bits. The write trigger will be issued automatically when you write value to this register.	Write
16-31	Reserved	Reserved.	N/A

cmd_readLarge Register: 0x0000_0058

31 ... 16	15 ... 0
Reserved	data

Bits	Field	Description	Privilege
0-15	data	Data field that stores received data that is more than 8 data bits.	Read
16-31	Reserved	Reserved.	N/A

Note: To trigger a SPI read command for large data (more than 8 bits), just trigger a regular SPI read command from **Cmd Register (0x0)**.

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 55: 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 56: 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

31 17	16	15 8	7 0
Reserved	dataValid	Reserved	data

Bits	Field	Description	Privilege
0-7	data	Data field which stores data that is transmitted or received.	Read/Write
8-15	Reserved	Reserved.	N/A
16	dataValid	Indicates read data that is available. 1'b1: Valid read data is available 1'b0: No data is being received	Read
17-31	Reserved	Reserved.	N/A

Status Register: 0x0000_0004

31 24	23 16	15	14 10	9	8	7 2	1	0
readOccupancy	writeAvailability	WriteBusy	Reserved	RxInterrupt	TxInterrupt	Reserved	RXInterrupt	TXInterrupt

Bits	Field	Description	Privilege
0	TXInterruptEnable	Flag interrupt when write or TX FIFO is empty. 1'b1: Enable 1'b0: Disable	Read/Write
1	RXInterruptEnable	Flag interrupt when read or RX FIFO is not empty. 1'b1: Enable 1'b0: Disable	Read/Write
2-7	Reserved	Reserved.	N/A
8	TxInterrupt	Write or TX FIFO interrupt status. 1'b1: Interrupt flagged 1'b0: Interrupt not flagged	Read
9	RxInterrupt	Read or RX FIFO interrupt status. 1'b1: Interrupt flagged 1'b0: Interrupt not flagged	Read
10-14	Reserved	Reserved.	N/A
15	WriteBusy	Write or transmit operation in progress. 1'b1: Busy 1'b0: Not busy	Read
16-23	writeAvailability	Shows the number of buffer that is available in write or TX FIFO.	Read
24-31	readOccupancy	Shows the occupancy in read or RX FIFO. Indicates the number of valid data to be read.	Read

Clock Divider Register: 0x0000_0008

31 20	19 0
Reserved	DividerFactor

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. 3'b000 = 1 bit 3'b001 = 2 bits 3'b010 = 3 bits 3'b011 = 4 bits 3'b100 = 5 bits 3'b101 = 6 bits 3'b110 = 7 bits 3'b111 = 8 bits	Write
3-7	Reserved	Reserved.	N/A
8-9	Parity	Parity bit. 2'b00: None 2'b01: Even 2'b10: Odd	Write
10-15	Reserved	Reserved.	
16	Stop	Stop bit number. 1'b1 = 2 stop bits 1'b0 = 1 stop bit	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	Read(RX) FIFO overflow error occurred. Write 1’b1 to reset the error.	Read/Write
2-7	Reserved	Reserved.	N/A
8	ReadBreak	Read break status. 1'b1: Break occurred 1'b0: Break not occurred	Read
9	BreakDetect	Break detected during read operation. Write 1’b1 to reset the error.	Read/Write
10	EnableBreak	Write 1'b1 to enable break detect.	Write
11	DisableBreak	Write 1'b1 to disable break detect.	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 57: 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

31 … 16	15 … 0
Reserved	Prescaler value

Bits	Field	Description	Privilege
0-15	Prescaler value	The clock divider ratio. Example: 16'd0: divide by 1 16'd1: divide by 2 ... 16'd65534: divide by 65535 16'd65535: divide by 65536	Read/Write
16-31	Reserved	Reserved.	-

Timer Configuration Register: 0x0000_0040

31 … 17	16	15 … 2	1	0
Reserved	Self-restart	Reserved	With prescaler	Without prescaler

Bits	Field	Description	Privilege
0	Without prescaler	Write 1'b1 to run timer without prescaler.	Read/Write
1	With prescaler	Write 1'b1 to run timer with prescaler.	Read/Write
2-15	Reserved	Reserved.	N/A
16	Self-restart	Write 1'b1 to enable self-restart when reach timer limit.	Read/Write
17-31	Reserved	Reserved.	N/A

Timer Limit Register: 0x0000_0044

31	0
Limit value	

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

Timer Value Register: 0x0000_0048

31	0
Value	

Bits	Field	Description	Privilege
0-31	Value	Value of the increment counter.	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 58: 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

31 2	0
Reserved	Software Interrupt

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

MTIMECMP Register (LO): 0x0000_4000

31 0
CMP value

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

MTIMECMP Register (HI): 0x0000_4004

31 0
CMP value

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

MTIME Register (LO): 0x0000_BFF8

31 0
Timer value

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

MTIME Register (HI): 0x0000_BFFC

31	0
Timer value	

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 59: Machine Information Register

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

Table 60: 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 61: 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 CSR

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.

31	30		20	19	18	17	16	15	14	13	12	11	10	8	7	6	4	3	2	1	0
SD		Reserved		MXR	SUM	MPRV	Reserved		FS		MPP		Reserved		MPIE	Reserved		MIE	Reserved	SIE	Reserved

Bits	Field	Description	Single/Multi-Core	w/FPU	w/MMU
0	Reserved	Reserved.	N/A	N/A	N/A
1	SIE	Machine global interrupt enable register.	N/A	N/A	Read/Write
2	Reserved	Reserved.	N/A	N/A	N/A
3	MIE	Machine interrupt enable register.	Read/Write	Read/Write	Read/Write
4-6	Reserved	Reserved.	N/A	N/A	N/A
7	MPIE	Machine previous interrupt enable.	Read/Write	Read/Write	Read/Write
8-10	Reserved	Reserved.	N/A	N/A	N/A
11-12	MPP	Machine previous privilege mode.	Read/Write	Read/Write	Read/Write
13-14	FS	Status of the floating-point unit. 2'b00: Off 2'b01: Initial 2'b10: Clean 2'b11: Dirty	N/A	Read	N/A
15-16	Reserved	Reserved.	N/A	N/A	N/A
17	MPRV	Modifies the privilege level that loads and stores the executables. 1'b1: Load and store memory address are translated and protected 1'b0: Normal mode	N/A	N/A	Read/Write
18	SUM	Modifies the privilege with which S-mode loads and stores access virtual memory. 1'b1: Access permitted 1'b0: S-mode memory accesses to pages that are accessible by U-mode will fault	N/A	N/A	Read/Write

Bits	Field	Description	Single/Multi-Core	w/FPU	w/MMU
19	MXR	Modifies the privilege that loads access virtual memory. 1'b1: Loads from pages marked with either readable or executable will succeed 1'b0: Only loads from pages marked readable will succeed	N/A	N/A	Read/Write
20-30	Reserved	Reserved.	N/A	N/A	N/A
31	SD	Indicates the presence of FS field with dirty state that requires saving extended user context to memory.	N/A	Read	N/A

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	SEIE	Reserved	MTIE	Reserved	STIE	Reserved	MSIE	Reserved	SSIE	Reserved

Bits	Field	Description	Single/Multi-Core	w/FPU	w/MMU
0	Reserved	Reserved.	N/A	N/A	N/A
1	SSIE	Supervisor mode software interrupt.	N/A	N/A	Read/Write
2	Reserved	Reserved.	N/A	N/A	N/A
3	MSIE	Machine software interrupt enable.	Read/Write	Read/Write	Read/Write
4	Reserved	Reserved.	N/A	N/A	N/A
5	STIE	Supervisor mode timer interrupt enable.	N/A	N/A	N/A
6	Reserved	Reserved.	N/A	N/A	N/A
7	MTIE	Machine timer interrupt enable.	Read/Write	Read/Write	Read/Write
8	Reserved	Reserved.	N/A	N/A	N/A
9	SEIE	Supervisor mode external interrupt enable.	N/A	N/A	Read/Write
10	Reserved	Reserved.	N/A	N/A	N/A
11	MEIE	Machine external interrupt enable.	Read/Write	Read/Write	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).

31	2	1	0
base		Reserved	

Bits	Field	Description	Single/ Multi-Core	w/FPU	w/MMU
0-1	Reserved	N/A	N/A	N/A	N/A
2-31	base	Vector base address.	Read/Write	Read/Write	Read/Write

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.

31	0
mscratch	

Bits	Field	Description	Single/ Multi-Core	w/FPU	w/MMU
0-31	mscratch	A temporary scratch space that can be used by machine mode software.	Read/Write	Read/Write	Read/Write

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	Single/Multi-Core	w/FPU	w/MMU
0-31	mepc	Machine exception program counter.	Read/Write	Read/Write	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	Single/Multi-Core	w/FPU	w/MMU
0-30	Exception code	See **Table 62: Machine Cause Register (mcause) Values after Trap** on page 60.	Read	Read	Read
31	Interrupt	mcause interrupt bit.	Read	Read	Read

Table 62: 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.
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 exceptions must set `mtval` informatively and which may unconditionally set it to zero.

31 ... 0
mtval

Bits	Field	Description	Single/Multi Core	w/FPU	w/MMU
0-31	mtval	Machine trap value register bit.	Read/Write	Read/Write	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	SEIP	Reserved	MTIP	Reserved	STIP	Reserved	MSIP	Reserved	SSIP	Reserved

Bits	Field	Description	Single/Multi-Core	w/FPU	w/MMU
0	Reserved	Reserved.	N/A	N/A	N/A
1	SSIP	Supervisor software interrupt pending.	N/A	N/A	Read/Write
2	Reserved	Reserved.	N/A	N/A	N/A
3	MSIP	Machine software interrupt pending.	Read/Write	Read/Write	Read/Write
4	Reserved	Reserved.	N/A	N/A	N/A
5	STIP	Supervisor timer interrupt pending.	N/A	N/A	R/W
6	Reserved	Reserved.	N/A	N/A	N/A
7	MTIP	Machine timer interrupt pending.	Read	Read	Read
8	Reserved	Reserved.	N/A	N/A	N/A
9	SEIP	Supervisor external interrupt pending.	N/A	N/A	Read/Write
10	Reserved	Reserved.	N/A	N/A	N/A
11	MEIP	Machine external interrupt pending.	Read	Read	Read

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 continuously in a multiprocessor system, but at least one hart must have a hart ID of zero. Hart IDs must be unique.

31	0
Hart ID	

Bits	Field	Description	Single / Multi-Core	w/FPU	w/MMU
0-31	Hart ID	Hardware thread ID.	Read	Read	Read

Timer Related CSR

CSR	Name	Description	Single / Multi-Core	w/FPU	w/MMU
0xC00	cycle	Cycle counter for **RDCYCLE** instruction.	N/A	N/A	Read
0xC01	time	Timer for **RDTIME** instruction.	N/A	N/A	Read
0xC02	timeh	Instructions-retired counter for **RDINSTRET** instruction.	N/A	N/A	Read

Floating-Point Related CSR

CSR	Name	Description	Single / Multi-Core	w/FPU	w/MMU
0x001	fflags	Floating-point accrued exceptions.	N/A	Read/Write	N/A
0x002	frm	Floating-point dynamic rounding mode.	N/A	Read/Write	N/A
0x003	fcsr	Floating-point control and status register (frm + fflags).	N/A	Read/Write	N/A

Supervisor-Level CSR

The supervisor should only view CSR state that should be visible to a supervisor-level operating system. There is no information about the existence (or non-existence) of higher privilege levels (machine level or other) visible in the CSRs accessible by the supervisor. Many supervisor CSRs are a subset of the equivalent machine mode CSR.

Supervisor Status Register (sstatus): 0x100

the sstatus register is a subset of the mstatus register.

31	30 20	19	18	17	16 15	14 13	12 9	8	7 6	5	4 2	1	0
SD	Reserved	MXR	SUM	MPRV	Reserved	FS	Reserved	SPP	Reserved	SPIE	Reserved	SIE	Reserved

Bits	Field	Description	Single/Multi-Core	w/FPU	w/MMU
0	Reserved	Reserved.	N/A	N/A	N/A
1	SIE	Supervisor global interrupt enable register.	N/A	N/A	Read/Write
2-4	Reserved	Reserved.	N/A	N/A	N/A
5	SPIE	Supervisor previous interrupt enable register.	N/A	N/A	Read/Write
6-7	Reserved	Reserved.	N/A	N/A	N/A
8	SPP	Supervisor previous privilege mode.	N/A	N/A	Read/Write
9-12	Reserved	Reserved.	N/A	N/A	N/A
13-14	FS	Status of the floating-point unit. 2'b00: Off 2'b01: Initial 2'b10: Clean 2'b11: Dirty	N/A	Read/Write	N/A
15-16	Reserved	Reserved	N/A	N/A	N/A
17	MPRV	Modifies the privilege level at which loads and stores the executables. 1'b1: Load and store memory address are translated and protected 1'b0: Normal mode	N/A	N/A	Read/Write
18	SUM	Modifies the privilege with which S-mode loads and stores access virtual memory. 1'b1: Access permitted 1'b0: S-mode memory accesses to pages that are accessible by U-mode will fault	N/A	N/A	Read/Write

Bits	Field	Description	Single/Multi-Core	w/FPU	w/MMU
19	MXR	Modifies the privilege that loads access virtual memory. 1'b1: loads from pages marked with either readable or executable will succeed 1'b0: only loads from page marked readable will succeed	N/A	N/A	Read/Write
20-30	Reserved	Reserved.	N/A	N/A	N/A
31	SD	Indicates the presence of FS field with dirty state that requires saving extended user context to memory.	N/A	Read/Write	N/A

Supervisor Interrupt Enable Register (sie): 0x104

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

31	10	9	8	7	6	5	4		2	1	0
Reserved		SEIE	Reserved			STIE	Reserved			SSIE	Reserved

Bits	Field	Description	Single/Multi-Core	w/FPU	w/MMU
0	Reserved	Reserved.	N/A	N/A	N/A
1	SSIE	Supervisor mode software interrupt.	N/A	N/A	Read/Write
2-4	Reserved	Reserved.	N/A	N/A	N/A
5	STIE	Supervisor mode timer interrupt enable.	N/A	N/A	Read/Write
6-8	Reserved	Reserved.	N/A	N/A	N/A
9	SEIE	Supervisor mode external interrupt enable.	N/A	N/A	Read/Write
10-31	Reserved	Reserved.	N/A	N/A	N/A

Supervisor Trap-Vector Base-Address Register (stvec): 0x305

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

31	2	1	0
base		Reserved	

Bits	Field	Description	Single/ Multi-Core	w/FPU	w/MMU
0-1	Reserved	N/A	N/A	N/A	N/A
2-31	base	Vector base address.	N/A	N/A	Read/Write

Supervisor Scratch Register (sscratch): 0x140

The `sscratch` register is a 32-bit read/write register dedicated for use by machine mode. Typically, `sscratch` is used to hold a pointer to the hart-local supervisor context while the hart is executing user code. At the beginning of a trap handler, `sscratch` is swapped with a user register to provide an initial working register.

31 … 0
sscratch

Bits	Field	Description	Single/ Multi-Core	w/FPU	w/MMU
0-31	sscratch	A temporary scratch space that can be used by supervisor mode software.	N/A	N/A	Read/Write

Supervisor Exception Program Counter (sepc): 0x141

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

31 … 0
sepc

Bits	Field	Description	Single/Multi-Core	w/FPU	w/MMU
0-31	sepc	Supervisor exception program counter.	N/A	N/A	Read/Write

Supervisor Cause Register (scause): 0x142

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

31	30 … 0
Interrupt	Exception Code

Bits	Field	Description	Single/Multi-Core	w/FPU	w/MMU
0-30	Exception code	See **Table 63: Machine Cause Register (scause) Values after Trap** on page 65.	N/A	N/A	Read
31	Interrupt	scause interrupt bit.	N/A	N/A	Read

Table 63: Machine Cause Register (scause) Values after Trap

Interrupt	Exception Code	Description
1	0	Reserved.
1	1	Supervisor software interrupt.
1	2-4	Reserved.
1	5	Supervisor timer interrupt.
1	6-8	Reserved.
1	9	Supervisor external interrupt.
1	10-15	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	Environment call from U-mode.
0	9	Environment call from S-mode.
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.

Supervisor Trap Value Register (stval): 0x143

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

31	0
stval	

Bits	Field	Description	Single/Multi-Core	w/FPU	w/MMU
0-31	stval	Supervisor trap value register bit.	N/A	N/A	Read/Write

Supervisor Interrupt Pending Register (sip): 0x144

The `sip` 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
Reserved		SEIP	Reserved			STIP	Reserved			SSIP	Reserved

Bits	Field	Description	Single/Multi-Core	w/FPU	w/MMU
0	Reserved	Reserved.	N/A	N/A	N/A
1	SSIP	Supervisor software interrupt pending.	N/A	N/A	Read/Write
2-4	Reserved	Reserved.	N/A	N/A	N/A
5	STIP	Supervisor timer interrupt pending.	N/A	N/A	Read/Write
6-8	Reserved	Reserved.	N/A	N/A	N/A
9	SEIP	Supervisor external interrupt pending.	N/A	N/A	Read/Write
10-11	Reserved	Reserved.	N/A	N/A	N/A

Supervisor Address Translation Protection Register (satp): 0x180

The `satp` register is a 32-bit register, which controls supervisor mode address translation and protection. This register holds the physical page number (PPN) of the root page table. For example, its supervisor physical address divided by 4KB; an address space identifier (ASID) that facilitates address translation fences on a per-address-space basis; and the `MODE` field that elects the current address translation scheme.

31	30	22	21	0
MODE	ASID		PPN	

Bits	Field	Description	Single/Multi-Core	w/FPU	w/MMU
0-21	PPN	Physical page number.	N/A	N/A	Read/Write
22-30	ASID	Address space identifier.	N/A	N/A	Read/Write
31	MODE	1'b1: Page-based 32-bits virtual processing. 1'b0: No translation or protection.	N/A	N/A	Read/Write

Revision History

Table 64: Revision History

Date	Version	Description
January 2025	5.1	Updated number of pins to 32 from 16 for GPIO, Input Register: 0x0000_0000, Output Register: 0x0000_0004, and Output Enable Register: 0x0000_0008. (DOC-2295)
December 2024	5.0	Updated TJ-Series and Trion and adding TP-Series in Resource Utilization and Performance and Performance Benchmark in Features. (DOC-2098) Added Watchdog Timer Register Map. Removed vector mode from both Machine Trap-Vector Base-Address Register (mtvec): 0x305 and Supervisor Trap-Vector Base Address Register (stvec): 0x305.
June 2024	4.1	Updated TJ-Series and Resource Utilization and Performance and Performance Benchmark in Features. (DOC-1790)
December 2023	4.0	Updated TJ-Series and Resource Utilization and Performance and Performance Benchmark in Features. (DOC-1533)
July 2023	3.3	Updated Little-Endian in the topic VexRiscv RISC-V Core. (DOC-1380)
June 2023	3.2	Updated the following sections: (DOC-1253) Trion Resource Utilization and Performance Performance Benchmark 1 and 2 tables. Interrupt Register: 0x0000_0020 Interrupt Clears Register: 0x0000_0024 Master Status Register: 0x0000_0040 Timeout Register: 0x0000_002C Added new topics: Slave Status Register: 0x0000_0044 Slave Override Register: 0x0000_0048
January 2023	3.1	Updated of interface details in Features topic and Sapphire RISC-V SoC Design Flow figure. Added new section in Features topic: Performance Benchmark. Updated the tables in UART Peripheral Interface → Status Register: 0x0000_0004, Config Register: 0x0000_000C, Error Break Register: 0x0000_0010. Updated tables in User Timer → Prescaler Register: 0x0000_0000 and Timer Configuration Register: 0x0000_0040. Changed main topic title of Machine-Level ISA to Machine-Level CSR. Updated all topics in Machine-Level CSR. Added new main topic Supervisor-Level CSR.
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.