



OVP Guide to Using Processor Models

Model specific information for riscv_RV64GC

Imperas Software Limited
Imperas Buildings, North Weston
Thame, Oxfordshire, OX9 2HA, U.K.
docs@imperas.com



| | |
|----------|---|
| Author | Imperas Software Limited |
| Version | 20181114.0 |
| Filename | OVP_Model_Specific_Information_riscv_RV64GC.pdf |
| Created | 21 November 2018 |
| Status | OVP Standard Release |

Copyright Notice

Copyright (c) 2018 Imperas Software Limited. All rights reserved. This software and documentation contain information that is the property of Imperas Software Limited. The software and documentation are furnished under a license agreement and may be used or copied only in accordance with the terms of the license agreement. No part of the software and documentation may be reproduced, transmitted, or translated, in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without prior written permission of Imperas Software Limited, or as expressly provided by the license agreement.

Right to Copy Documentation

The license agreement with Imperas permits licensee to make copies of the documentation for its internal use only. Each copy shall include all copyrights, trademarks, service marks, and proprietary rights notices, if any.

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the readers responsibility to determine the applicable regulations and to comply with them.

Disclaimer

IMPERAS SOFTWARE LIMITED, AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Model Release Status

This model is released as part of OVP releases and is included in OVPworld packages. Please visit OVPworld.org.

Contents

| | | |
|-----------|--|-----------|
| 1 | Overview | 1 |
| 1.1 | Description | 1 |
| 1.2 | Licensing | 1 |
| 1.3 | Features | 1 |
| 1.4 | Interrupts | 3 |
| 1.5 | Debug Mask | 3 |
| 1.6 | Integration Support | 4 |
| 1.6.1 | CSR Register External Implementation | 4 |
| 1.6.2 | LR/SC Active Address | 4 |
| 1.7 | Limitations | 4 |
| 1.8 | Verification | 5 |
| 1.9 | References | 5 |
| 2 | Configuration | 6 |
| 2.1 | Location | 6 |
| 2.2 | GDB Path | 6 |
| 2.3 | Semi-Host Library | 6 |
| 2.4 | Processor Endian-ness | 6 |
| 2.5 | QuantumLeap Support | 6 |
| 2.6 | Processor ELF code | 6 |
| 3 | All Variants in this model | 7 |
| 4 | Bus Master Ports | 8 |
| 5 | Bus Slave Ports | 9 |
| 6 | Net Ports | 10 |
| 7 | FIFO Ports | 11 |
| 8 | Formal Parameters | 12 |
| 8.1 | Parameters with enumerated types | 13 |
| 8.1.1 | Parameter user_version | 13 |
| 8.1.2 | Parameter priv_version | 13 |
| 9 | Execution Modes | 14 |
| 10 | Exceptions | 15 |

| | |
|--|-----------|
| 11 Hierarchy of the model | 16 |
| 11.1 Level 1: Hart | 16 |
| 12 Model Commands | 17 |
| 12.1 Level 1: Hart | 17 |
| 12.1.1 dumpTLB | 17 |
| 12.1.1.1 Argument description | 17 |
| 12.1.2 isync | 17 |
| 12.1.3 itrace | 17 |
| 13 Registers | 18 |
| 13.1 Level 1: Hart | 18 |
| 13.1.1 Core | 18 |
| 13.1.2 Floating_point | 19 |
| 13.1.3 User_Control_and_Status | 19 |
| 13.1.4 Supervisor_Control_and_Status | 20 |
| 13.1.5 Machine_Control_and_Status | 20 |
| 13.1.6 Integration_support | 22 |

Chapter 1

Overview

This document provides the details of an OVP Fast Processor Model variant.

OVP Fast Processor Models are written in C and provide a C API for use in C based platforms. The models also provide a native interface for use in SystemC TLM2 platforms.

The models are written using the OVP VMI API that provides a Virtual Machine Interface that defines the behavior of the processor. The VMI API makes a clear line between model and simulator allowing very good optimization and world class high speed performance. Most models are provided as a binary shared object and also as source. This allows the download and use of the model binary or the use of the source to explore and modify the model.

The models are run through an extensive QA and regression testing process and most model families are validated using technology provided by the processor IP owners. There is a companion document (OVP Guide to Using Processor Models) which explains the general concepts of OVP Fast Processor Models and their use. It is downloadable from the OVPworld website documentation pages.

1.1 Description

RISC-V RV64GC 64-bit processor model

1.2 Licensing

This Model is released under the Open Source Apache 2.0

1.3 Features

The model supports the following architectural features, defined in the misa CSR:

extension A (atomic instructions)

extension C (compressed instructions)

extension D (double-precision floating point)

extension F (single-precision floating point)

RV32I/64I/128I base ISA

extension M (integer multiply/divide instructions)

extension S (Supervisor mode)

extension U (User mode)

64-bit XLEN

If required, supported architectural features may be overridden using parameter “`misa_Extensions`”. Parameter “`misa_Extensions_mask`” can be used to specify which features can be dynamically enabled or disabled by writes to the `misa` register.

On this variant, the Machine trap-vector base-address register (`mtvec`) is writable. It can instead be configured as read-only using parameter “`mtvec_is_ro`”.

Values written to “`mtvec`” are masked using the value `0xffffffffffffd`. A different mask of writable bits may be specified using parameter “`mtvec_mask`” if required. In addition, when Vectored interrupt mode is enabled, parameter “`tvec_align`” may be used to specify additional hardware-enforced base address alignment. In this variant, “`tvec_align`” defaults to 0, implying no alignment constraint.

The initial value of “`mtvec`” is `0x0`. A different value may be specified using parameter “`mtvec`” if required.

Values written to “`stvec`” are masked using the value `0xffffffffffffd`. A different mask of writable bits may be specified using parameter “`stvec_mask`” if required. parameter “`tvec_align`” may be used to specify additional hardware-enforced base address alignment in the same manner as for the “`mtvec`” register, described above.

On reset, the model will restart at address `0x0`. A different reset address may be specified using parameter “`reset_address`” if required.

On an NMI, the model will restart at address `0x0`. A different NMI address may be specified using parameter “`nmi_address`” if required.

WFI will halt the processor until an interrupt occurs. It can instead be configured as a NOP using parameter “`wfi_is_nop`”. WFI timeout wait is implemented with a time limit of 0 (i.e. WFI causes an Illegal Instruction trap in Supervisor mode when `mstatus.TW=1`).

The “`cycle`” CSR is implemented in this variant. Set parameter “`cycle_undefined`” to True to instead specify that “`cycle`” is unimplemented and reads of it should trap to Machine mode.

The “`time`” CSR is implemented in this variant. Set parameter “`time_undefined`” to True to instead specify that “`time`” is unimplemented and reads of it should trap to Machine mode. Usually, the value of the “`time`” CSR should be provided by the platform - see notes below about the artifact “`CSR`” bus for information about how this is done.

The “`instret`” CSR is implemented in this variant. Set parameter “`instret_undefined`” to True to

instead specify that “instret” is unimplemented and reads of it should trap to Machine mode.

A 16-bit ASID is implemented. Use parameter “ASID_bits” to specify a different implemented ASID size if required.

This variant supports address translation modes 0, 8 and 9. Use parameter “Sv_modes” to specify a bit mask of different modes if required.

Unaligned memory accesses are not supported by this variant. Set parameter “unaligned” to “T” to enable such accesses.

16 PMP entries are implemented by this variant. Use parameter “PMP_registers” to specify a different number of PMP entries; set the parameter to 0 to disable the PMP unit.

LR/SC instructions are implemented with a 1-byte reservation granule. A different granule size may be specified using parameter “lr_sc_grain”.

By default, the processor starts with floating-point instructions disabled (mstatus.FS=0). Use parameter “mstatus_FS” to force mstatus.FS to a non-zero value for floating-point to be enabled from the start.

The D extension is enabled in this variant independently of the F extension. Set parameter “d_requires_f” to “T” to specify that the D extension requires the F extension to be enabled.

This variant implements floating point status in mstatus.FS as defined in the Privileged Architecture specification. To specify that a simpler mode supporting only values 0 (Off) and 3 (Dirty) should be used, Set parameter “fs_always_dirty” to “T”. When this simpler mode is used, any write of values 1 (Initial) or 2 (Clean) from privileged code behave as if value 3 was written.

1.4 Interrupts

The “reset” port is an active-high reset input. The processor is halted when “reset” goes high and resumes execution from the reset address specified using the “reset_address” parameter when the signal goes low. The “mcause” register is cleared to zero.

The “nmi” port is an active-high NMI input. The processor is halted when “nmi” goes high and resumes execution from the address specified using the “nmi_address” parameter when the signal goes low. The “mcause” register is cleared to zero.

All other interrupt ports are active high.

1.5 Debug Mask

It is possible to enable model debug messages in various categories. This can be done statically using the “override_debugMask” parameter, or dynamically using the “debugflags” command. Enabled messages are specified using a bitmask value, as follows:

Value 0x002: enable debugging of PMP and virtual memory state;

Value 0x004: enable debugging of interrupt state.

All other bits in the debug bitmask are reserved and must not be set to non-zero values.

1.6 Integration Support

This model implements a number of non-architectural pseudo-registers and other features to facilitate integration.

1.6.1 CSR Register External Implementation

If parameter “enable_CSR_bus” is True, an artifact 16-bit bus “CSR” is enabled. Slave callbacks installed on this bus can be used to implement modified CSR behavior (use `opBusSlaveNew` or `icmMapExternalMemory`, depending on the client API). A CSR with index `0xABC` is mapped on the bus at address `0xABC0`; as a concrete example, implementing CSR “time” (number `0xC01`) externally requires installation of callbacks at address `0xC010` on the CSR bus.

1.6.2 LR/SC Active Address

Artifact register “LRSCAddress” shows the active LR/SC lock address. The register holds all-ones if there is no LR/SC operation active.

1.7 Limitations

Instruction pipelines are not modeled in any way. All instructions are assumed to complete immediately. This means that instruction barrier instructions (e.g. `fence.i`) are treated as NOPs, with the exception of any Illegal Instruction behavior, which is modeled.

Caches and write buffers are not modeled in any way. All loads, fetches and stores complete immediately and in order, and are fully synchronous. Data barrier instructions (e.g. `fence`) are treated as NOPs, with the exception of any Illegal Instruction behavior, which is modeled.

Real-world timing effects are not modeled: all instructions are assumed to complete in a single cycle.

The processor fully supports the architecturally-specified floating-point instructions with the exception of the round-to-nearest, ties-to-max-magnitude rounding mode (RMM) which is supported for `fcvt` instruction variants that convert to long, unsigned long, word, or unsigned word only. In other cases, this rounding mode is treated a round-to-nearest, ties-to-even (RNE). Use of RMM rounding mode in any situation other than rounding to an integral value is dubious because it leads to cumulative bias towards larger-magnitude values.

Hardware Performance Monitor and Debug registers are not implemented and hardwired to zero.

The TLB is architecturally-accurate but not device accurate. This means that all TLB maintenance and address translation operations are fully implemented but the cache is larger than in the real device.

1.8 Verification

All instructions have been extensively tested by Imperas, using tests generated specifically for this model and also reference tests from <https://github.com/riscv/riscv-tests>.

1.9 References

The Model details are based upon the following specifications:

- RISC-V Instruction Set Manual, Volume I: User-Level ISA (User Architecture Version 2.2)
- RISC-V Instruction Set Manual, Volume II: Privileged Architecture (Privileged Architecture Version 1.10)

Chapter 2

Configuration

2.1 Location

This model's VLN is `riscv.ovpworld.org/processor/riscv/1.0`.

The model source is usually at:

`$IMPERAS_HOME/ImperasLib/source/riscv.ovpworld.org/processor/riscv/1.0`

The model binary is usually at:

`$IMPERAS_HOME/lib/$IMPERAS_ARCH/ImperasLib/riscv.ovpworld.org/processor/riscv/1.0`

2.2 GDB Path

The default GDB for this model is: `$IMPERAS_HOME/lib/$IMPERAS_ARCH/gdb/riscv64-unknown-elf-gdb`.

2.3 Semi-Host Library

The default semi-host library file is `riscv.ovpworld.org/semihosting/riscv64Newlib/1.0`

2.4 Processor Endian-ness

This is a LITTLE endian model.

2.5 QuantumLeap Support

This processor is qualified to run in a QuantumLeap enabled simulator.

2.6 Processor ELF code

The ELF code supported by this model is: `0xf3`.

Chapter 3

All Variants in this model

This model has these variants

| Variant | Description |
|----------------|------------------------------|
| RV32I | |
| RV32IM | |
| RV32IMC | |
| RV32IMAC | |
| RV32G | |
| RV32GC | |
| RV32GCN | |
| RV32E | |
| RV32EC | |
| RV64I | |
| RV64IM | |
| RV64IMC | |
| RV64IMAC | |
| RV64G | |
| RV64GC | (described in this document) |
| RV64GCN | |

Table 3.1: All Variants in this model

Chapter 4

Bus Master Ports

This model has these bus master ports.

| Name | min | max | Connect? | Description |
|-------------|-----|-----|-----------|-----------------|
| INSTRUCTION | 32 | 56 | mandatory | Instruction bus |
| DATA | 32 | 56 | optional | Data bus |

Table 4.1: Bus Master Ports

Chapter 5

Bus Slave Ports

This model has no bus slave ports.

Chapter 6

Net Ports

This model has these net ports.

| Name | Type | Connect? | Description |
|--------------------|-------|----------|-------------------------------|
| reset | input | optional | Reset |
| nmi | input | optional | NMI |
| SSWInterrupt | input | optional | Supervisor software interrupt |
| MSWInterrupt | input | optional | Machine software interrupt |
| STimerInterrupt | input | optional | Supervisor timer interrupt |
| MTimerInterrupt | input | optional | Machine timer interrupt |
| SExternalInterrupt | input | optional | Supervisor external interrupt |
| MExternalInterrupt | input | optional | Machine external interrupt |

Table 6.1: Net Ports

Chapter 7

FIFO Ports

This model has no FIFO ports.

Chapter 8

Formal Parameters

| Name | Type | Description |
|----------------------|-------------|---|
| variant | Enumeration | Selects variant (either a generic UISA or a specific model) |
| user_version | Enumeration | Specify required User Architecture version(2.2 or 2.3) |
| priv_version | Enumeration | Specify required Privileged Architecture version(1.10 or 1.11) |
| verbose | Boolean | Specify verbose output messages |
| updatePTEA | Boolean | Specify whether hardware update of PTE A bit is supported |
| updatePTED | Boolean | Specify whether hardware update of PTE D bit is supported |
| unaligned | Boolean | Specify whether the processor supports unaligned memory accesses |
| wfi_is_nop | Boolean | Specify whether WFI should be treated as a NOP (if not, halt while waiting for interrupts) |
| mtvec_is_ro | Boolean | Specify whether mtvec CSR is read-only |
| tvec_align | Uns32 | Specify hardware-enforced alignment of mtvec/stvec/utvec when Vectored interrupt mode enabled |
| mtvec_mask | Uns64 | Specify hardware-enforced mask of writable bits in mtvec register |
| stvec_mask | Uns64 | Specify hardware-enforced mask of writable bits in stvec register |
| tval_ii_code | Boolean | Specify whether mtval/stval contain faulting instruction bits on illegal instruction exception |
| cycle_undefined | Boolean | Specify that the cycle CSR is undefined (reads to it are emulated by a Machine mode trap) |
| time_undefined | Boolean | Specify that the time CSR is undefined (reads to it are emulated by a Machine mode trap) |
| instret_undefined | Boolean | Specify that the instret CSR is undefined (reads to it are emulated by a Machine mode trap) |
| enable_CSR_bus | Boolean | Add artifact CSR bus port, allowing CSR registers to be externally implemented |
| d_requires_f | Boolean | If D and F extensions are separately enabled in the misa CSR, whether D is enabled only if F is enabled |
| fs_always_dirty | Boolean | When FPU is enabled, whether mstatus.FS is always 3 (indicating dirty) |
| ASID_bits | Uns32 | Specify the number of implemented ASID bits |
| lr_sc_grain | Uns32 | Specify byte granularity of ll/sc lock region (constrained to a power of two) |
| reset_address | Uns64 | Override reset vector address |
| nmi_address | Uns64 | Override NMI vector address |
| PMP_registers | Uns32 | Specify the number of implemented PMP address registers |
| Sv_modes | Uns32 | Specify bit mask of implemented Sv modes (e.g. 1<<8 is Sv39) |
| local_int_num | Uns32 | Specify number of supplemental local interrupts |
| endian | Endian | Model endian |
| misa_MXL | Uns32 | Override default value of misa.MXL |
| misa_MXL_mask | Uns32 | Override mask of writable bits in misa.MXL |
| misa_Extensions | Uns32 | Override default value of misa.Extensions |
| misa_Extensions_mask | Uns32 | Override mask of writable bits in misa.Extensions |
| mvendorid | Uns64 | Override mvendorid register |

| | | |
|------------|-------|---|
| marchid | Uns64 | Override marchid register |
| mimpid | Uns64 | Override mimpid register |
| mhartid | Uns64 | Override mhartid register |
| mtvec | Uns64 | Override mtvec register |
| mstatus_FS | Uns32 | Override default value of mstatus.FS (initial state of floating point unit) |

Table 8.1: Parameters that can be set in: Hart

8.1 Parameters with enumerated types

8.1.1 Parameter user_version

| Set to this value | Description |
|-------------------|-------------------------------------|
| 2.2 | User Architecture Version 2.2 |
| 2.3 | User Architecture Version 2.3-draft |

Table 8.2: Values for Parameter user_version

8.1.2 Parameter priv_version

| Set to this value | Description |
|-------------------|--|
| 1.10 | Privileged Architecture Version 1.10 |
| 1.11 | Privileged Architecture Version 1.11-draft |

Table 8.3: Values for Parameter priv_version

Chapter 9

Execution Modes

| Mode | Code | Description |
|-------------|------|-----------------|
| User | 0 | User mode |
| Supervisor | 1 | Supervisor mode |
| Machine | 3 | Machine mode |

Table 9.1: Modes implemented in: Hart

Chapter 10

Exceptions

| Exception | Code | Description |
|------------------------------|------|--|
| InstructionAddressMisaligned | 0 | Fetch from unaligned address |
| InstructionAccessFault | 1 | No access permission for fetch |
| IllegalInstruction | 2 | Undecoded, unimplemented or disabled instruction |
| Breakpoint | 3 | EBREAK instruction executed |
| LoadAddressMisaligned | 4 | Load from unaligned address |
| LoadAccessFault | 5 | No access permission for load |
| StoreAMOAddressMisaligned | 6 | Store/atomic memory operation at unaligned address |
| StoreAMOAccessFault | 7 | No access permission for store/atomic memory operation |
| EnvironmentCallFromUMode | 8 | ECALL instruction executed in User mode |
| EnvironmentCallFromSMode | 9 | ECALL instruction executed in Supervisor mode |
| EnvironmentCallFromMMode | 11 | ECALL instruction executed in Machine mode |
| InstructionPageFault | 12 | Page fault at fetch address |
| LoadPageFault | 13 | Page fault at load address |
| StoreAMOPageFault | 15 | Page fault at store/atomic memory operation address |
| SSWInterrupt | 65 | Supervisor software interrupt |
| MSWInterrupt | 67 | Machine software interrupt |
| STimerInterrupt | 69 | Supervisor timer interrupt |
| MTimerInterrupt | 71 | Machine timer interrupt |
| SExternalInterrupt | 73 | Supervisor external interrupt |
| MExternalInterrupt | 75 | Machine external interrupt |

Table 10.1: Exceptions implemented in: Hart

Chapter 11

Hierarchy of the model

A CPU core may be configured to instance many processors of a Symmetrical Multi Processor (SMP). A CPU core may also have sub elements within a processor, for example hardware threading blocks.

OVP processor models can be written to include SMP blocks and to have many levels of hierarchy. Some OVP CPU models may have a fixed hierarchy, and some may be configured by settings in a configuration register. Please see the register definitions of this model.

This model documentation shows the settings and hierarchy of the default settings for this model variant.

11.1 Level 1: Hart

This level in the model hierarchy has 3 commands.

This level in the model hierarchy has 6 register groups:

| Group name | Registers |
|-------------------------------|-----------|
| Core | 33 |
| Floating_point | 32 |
| User_Control_and_Status | 35 |
| Supervisor_Control_and_Status | 10 |
| Machine_Control_and_Status | 101 |
| Integration_support | 1 |

Table 11.1: Register groups

This level in the model hierarchy has no children.

Chapter 12

Model Commands

A Processor model can implement one or more **Model Commands** available to be invoked from the simulator command line, from the OP API or from the Imperas Multiprocessor Debugger.

12.1 Level 1: Hart

12.1.1 dumpTLB

12.1.1.1 Argument description

show TLB contents

12.1.2 isync

specify instruction address range for synchronous execution

| Argument | Type | Description |
|------------|-------|--|
| -addresshi | Uns64 | end address of synchronous execution range |
| -addresslo | Uns64 | start address of synchronous execution range |

Table 12.1: isync command arguments

12.1.3 itrace

enable or disable instruction tracing

| Argument | Type | Description |
|-------------------|---------|--|
| -after | Uns64 | apply after this many instructions |
| -enable | Boolean | enable instruction tracing |
| -instructioncount | Boolean | include the instruction number in each trace |
| -off | Boolean | disable instruction tracing |
| -on | Boolean | enable instruction tracing |
| -registerchange | Boolean | show registers changed by this instruction |
| -registers | Boolean | show registers after each trace |

Table 12.2: itrace command arguments

Chapter 13

Registers

13.1 Level 1: Hart

13.1.1 Core

Registers at level:1, type:Hart group:Core

| Name | Bits | Initial-Hex | RW | Description |
|------|------|-------------|----|-----------------|
| zero | 64 | 0 | r- | |
| ra | 64 | 0 | rw | |
| sp | 64 | 0 | rw | stack pointer |
| gp | 64 | 0 | rw | |
| tp | 64 | 0 | rw | |
| t0 | 64 | 0 | rw | |
| t1 | 64 | 0 | rw | |
| t2 | 64 | 0 | rw | |
| s0 | 64 | 0 | rw | |
| s1 | 64 | 0 | rw | |
| a0 | 64 | 0 | rw | |
| a1 | 64 | 0 | rw | |
| a2 | 64 | 0 | rw | |
| a3 | 64 | 0 | rw | |
| a4 | 64 | 0 | rw | |
| a5 | 64 | 0 | rw | |
| a6 | 64 | 0 | rw | |
| a7 | 64 | 0 | rw | |
| s2 | 64 | 0 | rw | |
| s3 | 64 | 0 | rw | |
| s4 | 64 | 0 | rw | |
| s5 | 64 | 0 | rw | |
| s6 | 64 | 0 | rw | |
| s7 | 64 | 0 | rw | |
| s8 | 64 | 0 | rw | |
| s9 | 64 | 0 | rw | |
| s10 | 64 | 0 | rw | |
| s11 | 64 | 0 | rw | |
| t3 | 64 | 0 | rw | |
| t4 | 64 | 0 | rw | |
| t5 | 64 | 0 | rw | |
| t6 | 64 | 0 | rw | |
| pc | 64 | 0 | rw | program counter |

Table 13.1: Registers at level 1, type:Hart group:Core

13.1.2 Floating_point

Registers at level:1, type:Hart group:Floating_point

| Name | Bits | Initial-Hex | RW | Description |
|------|------|-------------|----|-------------|
| ft0 | 64 | 0 | rw | |
| ft1 | 64 | 0 | rw | |
| ft2 | 64 | 0 | rw | |
| ft3 | 64 | 0 | rw | |
| ft4 | 64 | 0 | rw | |
| ft5 | 64 | 0 | rw | |
| ft6 | 64 | 0 | rw | |
| ft7 | 64 | 0 | rw | |
| fs0 | 64 | 0 | rw | |
| fs1 | 64 | 0 | rw | |
| fa0 | 64 | 0 | rw | |
| fa1 | 64 | 0 | rw | |
| fa2 | 64 | 0 | rw | |
| fa3 | 64 | 0 | rw | |
| fa4 | 64 | 0 | rw | |
| fa5 | 64 | 0 | rw | |
| fa6 | 64 | 0 | rw | |
| fa7 | 64 | 0 | rw | |
| fs2 | 64 | 0 | rw | |
| fs3 | 64 | 0 | rw | |
| fs4 | 64 | 0 | rw | |
| fs5 | 64 | 0 | rw | |
| fs6 | 64 | 0 | rw | |
| fs7 | 64 | 0 | rw | |
| fs8 | 64 | 0 | rw | |
| fs9 | 64 | 0 | rw | |
| fs10 | 64 | 0 | rw | |
| fs11 | 64 | 0 | rw | |
| ft8 | 64 | 0 | rw | |
| ft9 | 64 | 0 | rw | |
| ft10 | 64 | 0 | rw | |
| ft11 | 64 | 0 | rw | |

Table 13.2: Registers at level 1, type:Hart group:Floating_point

13.1.3 User_Control_and_Status

Registers at level:1, type:Hart group:User_Control_and_Status

| Name | Bits | Initial-Hex | RW | Description |
|-------------|------|-------------|----|-----------------------------------|
| fflags | 64 | 0 | rw | Floating-Point Flags |
| frm | 64 | 0 | rw | Floating-Point Rounding Mode |
| fcsr | 64 | 0 | rw | Floating-Point Control and Status |
| cycle | 64 | 0 | r- | Cycle Counter |
| time | 64 | 0 | r- | Timer |
| instret | 64 | 0 | r- | Instructions Retired |
| hpmcounter3 | 64 | 0 | r- | Performance Monitor Counter 3 |

| | | | | |
|--------------|----|---|----|--------------------------------|
| hpmcounter4 | 64 | 0 | r- | Performance Monitor Counter 4 |
| hpmcounter5 | 64 | 0 | r- | Performance Monitor Counter 5 |
| hpmcounter6 | 64 | 0 | r- | Performance Monitor Counter 6 |
| hpmcounter7 | 64 | 0 | r- | Performance Monitor Counter 7 |
| hpmcounter8 | 64 | 0 | r- | Performance Monitor Counter 8 |
| hpmcounter9 | 64 | 0 | r- | Performance Monitor Counter 9 |
| hpmcounter10 | 64 | 0 | r- | Performance Monitor Counter 10 |
| hpmcounter11 | 64 | 0 | r- | Performance Monitor Counter 11 |
| hpmcounter12 | 64 | 0 | r- | Performance Monitor Counter 12 |
| hpmcounter13 | 64 | 0 | r- | Performance Monitor Counter 13 |
| hpmcounter14 | 64 | 0 | r- | Performance Monitor Counter 14 |
| hpmcounter15 | 64 | 0 | r- | Performance Monitor Counter 15 |
| hpmcounter16 | 64 | 0 | r- | Performance Monitor Counter 16 |
| hpmcounter17 | 64 | 0 | r- | Performance Monitor Counter 17 |
| hpmcounter18 | 64 | 0 | r- | Performance Monitor Counter 18 |
| hpmcounter19 | 64 | 0 | r- | Performance Monitor Counter 19 |
| hpmcounter20 | 64 | 0 | r- | Performance Monitor Counter 20 |
| hpmcounter21 | 64 | 0 | r- | Performance Monitor Counter 21 |
| hpmcounter22 | 64 | 0 | r- | Performance Monitor Counter 22 |
| hpmcounter23 | 64 | 0 | r- | Performance Monitor Counter 23 |
| hpmcounter24 | 64 | 0 | r- | Performance Monitor Counter 24 |
| hpmcounter25 | 64 | 0 | r- | Performance Monitor Counter 25 |
| hpmcounter26 | 64 | 0 | r- | Performance Monitor Counter 26 |
| hpmcounter27 | 64 | 0 | r- | Performance Monitor Counter 27 |
| hpmcounter28 | 64 | 0 | r- | Performance Monitor Counter 28 |
| hpmcounter29 | 64 | 0 | r- | Performance Monitor Counter 29 |
| hpmcounter30 | 64 | 0 | r- | Performance Monitor Counter 30 |
| hpmcounter31 | 64 | 0 | r- | Performance Monitor Counter 31 |

Table 13.3: Registers at level 1, type:Hart group:User_Control_and_Status

13.1.4 Supervisor_Control_and_Status

Registers at level:1, type:Hart group:Supervisor_Control_and_Status

| Name | Bits | Initial-Hex | RW | Description |
|------------|------|-------------|----|---|
| sstatus | 64 | 2 00000000 | rw | Supervisor Status |
| sie | 64 | 0 | rw | Supervisor Interrupt Enable |
| stvec | 64 | 0 | rw | Supervisor Trap-Vector Base-Address |
| scounteren | 64 | 0 | rw | Supervisor Counter Enable |
| sscratch | 64 | 0 | rw | Supervisor Scratch |
| sepc | 64 | 0 | rw | Supervisor Exception Program Counter |
| scause | 64 | 0 | rw | Supervisor Cause |
| stval | 64 | 0 | rw | Supervisor Trap Value |
| sip | 64 | 0 | rw | Supervisor Interrupt Pending |
| satp | 64 | 0 | rw | Supervisor Address Translation and Protection |

Table 13.4: Registers at level 1, type:Hart group:Supervisor_Control_and_Status

13.1.5 Machine_Control_and_Status

Registers at level:1, type:Hart group:Machine_Control_and_Status

| Name | Bits | Initial-Hex | RW | Description |
|---------|------|-------------|----|----------------|
| mstatus | 64 | a 00000000 | rw | Machine Status |

| | | | | |
|-------------|----|----------------------|----|---|
| misa | 64 | 80000000 0014112d | rw | ISA and Extensions |
| medeleg | 64 | 0 | rw | Machine Exception Delegation |
| mideleg | 64 | 0 | rw | Machine Interrupt Delegation |
| mie | 64 | 0 | rw | Machine Interrupt Enable |
| mtvec | 64 | 0 | rw | Machine Trap-Vector Base-Address |
| mcounteren | 64 | 0 | rw | Machine Counter Enable |
| mhpmevent3 | 64 | 0 | rw | Machine Performance Monitor Event Select 3 |
| mhpmevent4 | 64 | 0 | rw | Machine Performance Monitor Event Select 4 |
| mhpmevent5 | 64 | 0 | rw | Machine Performance Monitor Event Select 5 |
| mhpmevent6 | 64 | 0 | rw | Machine Performance Monitor Event Select 6 |
| mhpmevent7 | 64 | 0 | rw | Machine Performance Monitor Event Select 7 |
| mhpmevent8 | 64 | 0 | rw | Machine Performance Monitor Event Select 8 |
| mhpmevent9 | 64 | 0 | rw | Machine Performance Monitor Event Select 9 |
| mhpmevent10 | 64 | 0 | rw | Machine Performance Monitor Event Select 10 |
| mhpmevent11 | 64 | 0 | rw | Machine Performance Monitor Event Select 11 |
| mhpmevent12 | 64 | 0 | rw | Machine Performance Monitor Event Select 12 |
| mhpmevent13 | 64 | 0 | rw | Machine Performance Monitor Event Select 13 |
| mhpmevent14 | 64 | 0 | rw | Machine Performance Monitor Event Select 14 |
| mhpmevent15 | 64 | 0 | rw | Machine Performance Monitor Event Select 15 |
| mhpmevent16 | 64 | 0 | rw | Machine Performance Monitor Event Select 16 |
| mhpmevent17 | 64 | 0 | rw | Machine Performance Monitor Event Select 17 |
| mhpmevent18 | 64 | 0 | rw | Machine Performance Monitor Event Select 18 |
| mhpmevent19 | 64 | 0 | rw | Machine Performance Monitor Event Select 19 |
| mhpmevent20 | 64 | 0 | rw | Machine Performance Monitor Event Select 20 |
| mhpmevent21 | 64 | 0 | rw | Machine Performance Monitor Event Select 21 |
| mhpmevent22 | 64 | 0 | rw | Machine Performance Monitor Event Select 22 |
| mhpmevent23 | 64 | 0 | rw | Machine Performance Monitor Event Select 23 |
| mhpmevent24 | 64 | 0 | rw | Machine Performance Monitor Event Select 24 |
| mhpmevent25 | 64 | 0 | rw | Machine Performance Monitor Event Select 25 |
| mhpmevent26 | 64 | 0 | rw | Machine Performance Monitor Event Select 26 |
| mhpmevent27 | 64 | 0 | rw | Machine Performance Monitor Event Select 27 |
| mhpmevent28 | 64 | 0 | rw | Machine Performance Monitor Event Select 28 |
| mhpmevent29 | 64 | 0 | rw | Machine Performance Monitor Event Select 29 |
| mhpmevent30 | 64 | 0 | rw | Machine Performance Monitor Event Select 30 |
| mhpmevent31 | 64 | 0 | rw | Machine Performance Monitor Event Select 31 |
| mscratch | 64 | 0 | rw | Machine Scratch |
| mepc | 64 | 0 | rw | Machine Exception Program Counter |
| mcause | 64 | 0 | rw | Machine Cause |
| mtval | 64 | 0 | rw | Machine Trap Value |
| mip | 64 | 0 | rw | Machine Interrupt Pending |
| pmpcfg0 | 64 | 0 | rw | Physical Memory Protection Configuration 0 |
| pmpcfg2 | 64 | 0 | rw | Physical Memory Protection Configuration 2 |
| pmpaddr0 | 64 | 0 | rw | Physical Memory Protection Address 0 |
| pmpaddr1 | 64 | 0 | rw | Physical Memory Protection Address 1 |
| pmpaddr2 | 64 | 0 | rw | Physical Memory Protection Address 2 |
| pmpaddr3 | 64 | 0 | rw | Physical Memory Protection Address 3 |
| pmpaddr4 | 64 | 0 | rw | Physical Memory Protection Address 4 |
| pmpaddr5 | 64 | 0 | rw | Physical Memory Protection Address 5 |
| pmpaddr6 | 64 | 0 | rw | Physical Memory Protection Address 6 |
| pmpaddr7 | 64 | 0 | rw | Physical Memory Protection Address 7 |
| pmpaddr8 | 64 | 0 | rw | Physical Memory Protection Address 8 |
| pmpaddr9 | 64 | 0 | rw | Physical Memory Protection Address 9 |
| pmpaddr10 | 64 | 0 | rw | Physical Memory Protection Address 10 |
| pmpaddr11 | 64 | 0 | rw | Physical Memory Protection Address 11 |
| pmpaddr12 | 64 | 0 | rw | Physical Memory Protection Address 12 |

| | | | | |
|---------------|----|---|----|---|
| pmpaddr13 | 64 | 0 | rw | Physical Memory Protection Address 13 |
| pmpaddr14 | 64 | 0 | rw | Physical Memory Protection Address 14 |
| pmpaddr15 | 64 | 0 | rw | Physical Memory Protection Address 15 |
| tselect | 64 | - | rw | Debug/Trace Trigger Register Select (not implemented) |
| tdata1 | 64 | - | rw | Debug/Trace Trigger Data 1 (not implemented) |
| tdata2 | 64 | - | rw | Debug/Trace Trigger Data 2 (not implemented) |
| tdata3 | 64 | - | rw | Debug/Trace Trigger Data 3 (not implemented) |
| dcsr | 64 | - | rw | Debug Control and Status (not implemented) |
| dpc | 64 | - | rw | Debug PC (not implemented) |
| dscratch | 64 | - | rw | Debug Scratch (not implemented) |
| mcycle | 64 | 0 | rw | Machine Cycle Counter |
| minstret | 64 | 0 | rw | Machine Instructions Retired |
| mhpmcounter3 | 64 | 0 | rw | Machine Performance Monitor Counter 3 |
| mhpmcounter4 | 64 | 0 | rw | Machine Performance Monitor Counter 4 |
| mhpmcounter5 | 64 | 0 | rw | Machine Performance Monitor Counter 5 |
| mhpmcounter6 | 64 | 0 | rw | Machine Performance Monitor Counter 6 |
| mhpmcounter7 | 64 | 0 | rw | Machine Performance Monitor Counter 7 |
| mhpmcounter8 | 64 | 0 | rw | Machine Performance Monitor Counter 8 |
| mhpmcounter9 | 64 | 0 | rw | Machine Performance Monitor Counter 9 |
| mhpmcounter10 | 64 | 0 | rw | Machine Performance Monitor Counter 10 |
| mhpmcounter11 | 64 | 0 | rw | Machine Performance Monitor Counter 11 |
| mhpmcounter12 | 64 | 0 | rw | Machine Performance Monitor Counter 12 |
| mhpmcounter13 | 64 | 0 | rw | Machine Performance Monitor Counter 13 |
| mhpmcounter14 | 64 | 0 | rw | Machine Performance Monitor Counter 14 |
| mhpmcounter15 | 64 | 0 | rw | Machine Performance Monitor Counter 15 |
| mhpmcounter16 | 64 | 0 | rw | Machine Performance Monitor Counter 16 |
| mhpmcounter17 | 64 | 0 | rw | Machine Performance Monitor Counter 17 |
| mhpmcounter18 | 64 | 0 | rw | Machine Performance Monitor Counter 18 |
| mhpmcounter19 | 64 | 0 | rw | Machine Performance Monitor Counter 19 |
| mhpmcounter20 | 64 | 0 | rw | Machine Performance Monitor Counter 20 |
| mhpmcounter21 | 64 | 0 | rw | Machine Performance Monitor Counter 21 |
| mhpmcounter22 | 64 | 0 | rw | Machine Performance Monitor Counter 22 |
| mhpmcounter23 | 64 | 0 | rw | Machine Performance Monitor Counter 23 |
| mhpmcounter24 | 64 | 0 | rw | Machine Performance Monitor Counter 24 |
| mhpmcounter25 | 64 | 0 | rw | Machine Performance Monitor Counter 25 |
| mhpmcounter26 | 64 | 0 | rw | Machine Performance Monitor Counter 26 |
| mhpmcounter27 | 64 | 0 | rw | Machine Performance Monitor Counter 27 |
| mhpmcounter28 | 64 | 0 | rw | Machine Performance Monitor Counter 28 |
| mhpmcounter29 | 64 | 0 | rw | Machine Performance Monitor Counter 29 |
| mhpmcounter30 | 64 | 0 | rw | Machine Performance Monitor Counter 30 |
| mhpmcounter31 | 64 | 0 | rw | Machine Performance Monitor Counter 31 |
| mvendorid | 64 | 0 | r- | Vendor ID |
| marchid | 64 | 0 | r- | Architecture ID |
| mimpid | 64 | 0 | r- | Implementation ID |
| mhartid | 64 | 0 | r- | Hardware Thread ID |

Table 13.5: Registers at level 1, type:Hart group:Machine_Control_and_Status

13.1.6 Integration_support

Registers at level:1, type:Hart group:Integration_support

| Name | Bits | Initial-Hex | RW | Description |
|-------------|------|----------------|----|---------------------------|
| LRSCAddress | 64 | ffffff fffffff | rw | LR/SC active lock address |

Table 13.6: Registers at level 1, type:Hart group:Integration_support