



OVP Guide to Using Processor Models

Model Specific Information for variant riscv_RV32G

Imperas Software Limited

Imperas Buildings, North Weston
Thame, Oxfordshire, OX9 2HA, UK
docs@imperas.com



Author	Imperas Software Limited
Version	0.5
Filename	OVP_Model_Specific_Information_riscv_RV32G.pdf
Created	27 February 2018
Status	OVP Standard Release

Copyright Notice

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 reader's 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.

Table of Contents

1 Overview.....	4
1.1 Description.....	4
1.2 Licensing.....	4
1.3 Features.....	4
1.4 Interrupts.....	5
1.5 Debug Mask.....	5
1.6 Integration Support.....	5
1.6.1 CSR Register External Implementation.....	5
1.6.2 LR/SC Active Address.....	5
1.7 Limitations.....	6
1.8 Verification.....	6
1.9 References.....	6
2 Configuration.....	6
2.1 Location.....	6
2.2 GDB Path.....	6
2.3 Semi-Host Library.....	6
2.4 Processor Endian-ness.....	7
2.5 QuantumLeap Support.....	7
2.6 Processor ELF Code.....	7
3 Other Variants in this Model.....	7
4 Bus Ports.....	7
5 Net Ports.....	7
6 FIFO Ports.....	8
7 Parameters.....	8
8 Execution Modes.....	9
9 Exceptions.....	9
10 Hierarchy of the model.....	11
10.1 Level 1:.....	11
11 Model Commands.....	12
11.1 Level 1:.....	12
11.1.1 dumpTLB.....	12
11.1.1.1 Argument description.....	12
11.1.2 isync.....	12
11.1.3 itrace.....	12
12 Registers.....	12
12.1 Level 1:.....	12
12.1.1 Core.....	12
12.1.2 Floating_point.....	13
12.1.3 User_Control_and_Status.....	14
12.1.4 Supervisor_Control_and_Status.....	16
12.1.5 Machine_Control_and_Status.....	16
12.1.6 Integration_support.....	21

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 RV32G 32-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 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)

32-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".

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

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 "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.

A 9-bit ASID is implemented. Use parameter "ASID_bits" to specify a different implemented ASID size if required.

This variant supports address translation modes 0 and 1. 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.

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)

2 Configuration

2.1 Location

The model source and object file is found in the VLNV tree at:

`riscv.ovpworld.org/processor/riscv/1.0`

2.2 GDB Path

The default GDB for this model is found at:

`$IMPERAS_HOME/lib/$IMPERAS_ARCH/gdb/riscv64-unknown-elf-gdb`

2.3 Semi-Host Library

The default semi-host library file is found in the VLNV tree at :

`riscv.ovpworld.org/semihosting/riscv32Newlib/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

3 Other Variants in this Model

Table 1. All variants in this model

Variant
RV32I
RV32IM
RV32IMC
RV32IMAC
RV32G
RV32GC
RV32GCN
RV32E
RV32EC
RV64I
RV64IM
RV64IMC
RV64IMAC
RV64G
RV64GC
RV64GCN

4 Bus Ports

Table 2. Bus Ports

Type	Name	min	max	Description
master (initiator)	INSTRUCTION	32	34	Instruction bus
master (initiator)	DATA	32	34	Data bus

5 Net Ports

Table 3. Net Ports

Name	Type	Description
------	------	-------------

reset	input	Reset
nmi	input	NMI
SSWInterrupt	input	Supervisor software interrupt
MSWInterrupt	input	Machine software interrupt
STimerInterrupt	input	Supervisor timer interrupt
MTimerInterrupt	input	Machine timer interrupt
SExternalInterrupt	input	Supervisor external interrupt
MExternalInterrupt	input	Machine external interrupt

6 FIFO Ports

No FIFO Ports in this model.

7 Parameters

Table 4. Parameters that can be set in the model, type:

Name	Type	Description
user_version	Enumeration	Specify required User Architecture version 2.2=0 (User Architecture Version 2.2) 2.3=1 (User Architecture Version 2.3-draft)
priv_version	Enumeration	Specify required Privileged Architecture version 1.10=0 (Privileged Architecture Version 1.10) 1.11=1 (Privileged Architecture Version 1.11-draft)
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
time_undefined	Boolean	Specify that the time 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
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
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)

8 Execution Modes

Table 5. CPU modes implemented in the model, type:

Name	Code	Description
User	0	User mode
Supervisor	1	Supervisor mode
Machine	3	Machine mode

9 Exceptions

Table 6. Exceptions handled by the model, type:

Name	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	33	Supervisor software interrupt
MSWInterrupt	35	Machine software interrupt
STimerInterrupt	37	Supervisor timer interrupt
MTimerInterrupt	39	Machine timer interrupt

SExternalInterrupt	41	Supervisor external interrupt
MExternalInterrupt	43	Machine external interrupt

10 Hierarchy of the model

A CPU core may allow the user to configure it 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.

10.1 Level 1:

This level in the model hierarchy has 3 commands.

This level in the model hierarchy has 6 register groups:

Table 7. Register groups

Group name	Registers
Core	33
Floating_point	32
User_Control_and_Status	67
Supervisor_Control_and_Status	10
Machine_Control_and_Status	134
Integration_support	1

This level in the model hierarchy has no children.

11 Model Commands

11.1 Level 1:

11.1.1 *dumpTLB*

11.1.1.1 *Argument description*

show TLB contents

11.1.2 *isync*

specify instruction address range for synchronous execution

Table 8. *isync* command arguments

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

11.1.3 *itrace*

enable or disable instruction tracing

Table 9. *itrace* command arguments

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

12 Registers

12.1 Level 1:

12.1.1 *Core*

Table 10. Registers at level 1, type: , register group: 'Core'

Name	Bits	Initial value (Hex)		Description
zero	32	0	r-	
ra	32	0	rw	
sp	32	0	rw	stack pointer
gp	32	0	rw	
tp	32	0	rw	
t0	32	0	rw	
t1	32	0	rw	
t2	32	0	rw	
s0	32	0	rw	
s1	32	0	rw	
a0	32	0	rw	
a1	32	0	rw	

a2	32	0	rw	
a3	32	0	rw	
a4	32	0	rw	
a5	32	0	rw	
a6	32	0	rw	
a7	32	0	rw	
s2	32	0	rw	
s3	32	0	rw	
s4	32	0	rw	
s5	32	0	rw	
s6	32	0	rw	
s7	32	0	rw	
s8	32	0	rw	
s9	32	0	rw	
s10	32	0	rw	
s11	32	0	rw	
t3	32	0	rw	
t4	32	0	rw	
t5	32	0	rw	
t6	32	0	rw	
pc	32	0	rw	program counter

12.1.2 Floating_point

Table 11. Registers at level 1, type: , register group: 'Floating_point'

Name	Bits	Initial value (Hex)		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	

12.1.3 User_Control_and_Status

Table 12. Registers at level 1, type: , register group: 'User_Control_and_Status'

Name	Bits	Initial value (Hex)		Description
fflags	32	0	rw	Floating-Point Flags
frm	32	0	rw	Floating-Point Rounding Mode
fcsr	32	0	rw	Floating-Point Control and Status
cycle	32	0	r-	Cycle Counter
time	32	0	r-	Timer
instret	32	0	r-	Instructions Retired
hpmcounter3	32	-	r-	Performance Monitor Counter 3 (hardwired to zero)
hpmcounter4	32	-	r-	Performance Monitor Counter 4 (hardwired to zero)
hpmcounter5	32	-	r-	Performance Monitor Counter 5 (hardwired to zero)
hpmcounter6	32	-	r-	Performance Monitor Counter 6 (hardwired to zero)
hpmcounter7	32	-	r-	Performance Monitor Counter 7 (hardwired to zero)
hpmcounter8	32	-	r-	Performance Monitor Counter 8 (hardwired to zero)
hpmcounter9	32	-	r-	Performance Monitor Counter 9 (hardwired to zero)
hpmcounter10	32	-	r-	Performance Monitor Counter 10 (hardwired to zero)
hpmcounter11	32	-	r-	Performance Monitor Counter 11 (hardwired to zero)
hpmcounter12	32	-	r-	Performance Monitor Counter 12 (hardwired to zero)
hpmcounter13	32	-	r-	Performance Monitor Counter 13 (hardwired to zero)
hpmcounter14	32	-	r-	Performance Monitor Counter 14 (hardwired to zero)
hpmcounter15	32	-	r-	Performance Monitor Counter 15 (hardwired to zero)
hpmcounter16	32	-	r-	Performance Monitor Counter 16 (hardwired to zero)
hpmcounter17	32	-	r-	Performance Monitor Counter 17 (hardwired to zero)
hpmcounter18	32	-	r-	Performance Monitor Counter 18 (hardwired to zero)

hpmcounter19	32	-	r-	Performance Monitor Counter 19 (hardwired to zero)
hpmcounter20	32	-	r-	Performance Monitor Counter 20 (hardwired to zero)
hpmcounter21	32	-	r-	Performance Monitor Counter 21 (hardwired to zero)
hpmcounter22	32	-	r-	Performance Monitor Counter 22 (hardwired to zero)
hpmcounter23	32	-	r-	Performance Monitor Counter 23 (hardwired to zero)
hpmcounter24	32	-	r-	Performance Monitor Counter 24 (hardwired to zero)
hpmcounter25	32	-	r-	Performance Monitor Counter 25 (hardwired to zero)
hpmcounter26	32	-	r-	Performance Monitor Counter 26 (hardwired to zero)
hpmcounter27	32	-	r-	Performance Monitor Counter 27 (hardwired to zero)
hpmcounter28	32	-	r-	Performance Monitor Counter 28 (hardwired to zero)
hpmcounter29	32	-	r-	Performance Monitor Counter 29 (hardwired to zero)
hpmcounter30	32	-	r-	Performance Monitor Counter 30 (hardwired to zero)
hpmcounter31	32	-	r-	Performance Monitor Counter 31 (hardwired to zero)
cycleh	32	0	r-	Cycle Counter High
timeh	32	0	r-	Timer High
instreth	32	0	r-	Instructions Retired High
hpmcounterh3	32	-	r-	Performance Monitor High 3 (hardwired to zero)
hpmcounterh4	32	-	r-	Performance Monitor High 4 (hardwired to zero)
hpmcounterh5	32	-	r-	Performance Monitor High 5 (hardwired to zero)
hpmcounterh6	32	-	r-	Performance Monitor High 6 (hardwired to zero)
hpmcounterh7	32	-	r-	Performance Monitor High 7 (hardwired to zero)
hpmcounterh8	32	-	r-	Performance Monitor High 8 (hardwired to zero)
hpmcounterh9	32	-	r-	Performance Monitor High 9 (hardwired to zero)
hpmcounterh10	32	-	r-	Performance Monitor High 10 (hardwired to zero)
hpmcounterh11	32	-	r-	Performance Monitor High 11 (hardwired to zero)
hpmcounterh12	32	-	r-	Performance Monitor High 12 (hardwired to zero)
hpmcounterh13	32	-	r-	Performance Monitor High 13 (hardwired to zero)
hpmcounterh14	32	-	r-	Performance Monitor High 14 (hardwired to zero)
hpmcounterh15	32	-	r-	Performance Monitor High 15 (hardwired to zero)
hpmcounterh16	32	-	r-	Performance Monitor High 16 (hardwired to zero)
hpmcounterh17	32	-	r-	Performance Monitor High 17 (hardwired to zero)
hpmcounterh18	32	-	r-	Performance Monitor High 18 (hardwired to zero)
hpmcounterh19	32	-	r-	Performance Monitor High 19 (hardwired to zero)
hpmcounterh20	32	-	r-	Performance Monitor High 20 (hardwired to zero)
hpmcounterh21	32	-	r-	Performance Monitor High 21 (hardwired to zero)
hpmcounterh22	32	-	r-	Performance Monitor High 22 (hardwired to zero)
hpmcounterh23	32	-	r-	Performance Monitor High 23 (hardwired to zero)
hpmcounterh24	32	-	r-	Performance Monitor High 24 (hardwired to zero)
hpmcounterh25	32	-	r-	Performance Monitor High 25 (hardwired to zero)
hpmcounterh26	32	-	r-	Performance Monitor High 26 (hardwired to zero)
hpmcounterh27	32	-	r-	Performance Monitor High 27 (hardwired to zero)
hpmcounterh28	32	-	r-	Performance Monitor High 28 (hardwired to zero)

hpmcounterh29	32	-	r-	Performance Monitor High 29 (hardwired to zero)
hpmcounterh30	32	-	r-	Performance Monitor High 30 (hardwired to zero)
hpmcounterh31	32	-	r-	Performance Monitor High 31 (hardwired to zero)

12.1.4 Supervisor_Control_and_Status

Table 13. Registers at level 1, type: , register group: 'Supervisor_Control_and_Status'

Name	Bits	Initial value (Hex)		Description
sstatus	32	0	rw	Supervisor Status
sie	32	0	rw	Supervisor Interrupt Enable
stvec	32	0	rw	Supervisor Trap-Vector Base-Address
scounteren	32	0	rw	Supervisor Counter Enable
sscratch	32	0	rw	Supervisor Scratch
sepc	32	0	rw	Supervisor Exception Program Counter
scause	32	0	rw	Supervisor Cause
stval	32	0	rw	Supervisor Trap Value
sip	32	0	rw	Supervisor Interrupt Pending
satp	32	0	rw	Supervisor Address Translation and Protection

12.1.5 Machine_Control_and_Status

Table 14. Registers at level 1, type: , register group: 'Machine_Control_and_Status'

Name	Bits	Initial value (Hex)		Description
mvendorid	32	0	r-	Vendor ID
marchid	32	0	r-	Architecture ID
mimpid	32	0	r-	Implementation ID
mhartid	32	0	r-	Hardware Thread ID
mstatus	32	0	rw	Machine Status
misa	32	40141129	rw	ISA and Extensions
medeleg	32	0	rw	Machine Exception Delegation
mideleg	32	0	rw	Machine Interrupt Delegation
mie	32	0	rw	Machine Interrupt Enable
mtvec	32	0	rw	Machine Trap-Vector Base-Address
mcounteren	32	0	rw	Machine Counter Enable
mscratch	32	0	rw	Machine Scratch
mepc	32	0	rw	Machine Exception Program Counter
mcause	32	0	rw	Machine Cause
mtval	32	0	rw	Machine Trap Value
mip	32	0	rw	Machine Interrupt Pending
pmpcfg0	32	0	rw	Physical Memory Protection Configuration 0
pmpcfg1	32	0	rw	Physical Memory Protection Configuration 1
pmpcfg2	32	0	rw	Physical Memory Protection Configuration 2

pmpcfg3	32	0	rw	Physical Memory Protection Configuration 3
pmpaddr0	32	0	rw	Physical Memory Protection Address 0
pmpaddr1	32	0	rw	Physical Memory Protection Address 1
pmpaddr2	32	0	rw	Physical Memory Protection Address 2
pmpaddr3	32	0	rw	Physical Memory Protection Address 3
pmpaddr4	32	0	rw	Physical Memory Protection Address 4
pmpaddr5	32	0	rw	Physical Memory Protection Address 5
pmpaddr6	32	0	rw	Physical Memory Protection Address 6
pmpaddr7	32	0	rw	Physical Memory Protection Address 7
pmpaddr8	32	0	rw	Physical Memory Protection Address 8
pmpaddr9	32	0	rw	Physical Memory Protection Address 9
pmpaddr10	32	0	rw	Physical Memory Protection Address 10
pmpaddr11	32	0	rw	Physical Memory Protection Address 11
pmpaddr12	32	0	rw	Physical Memory Protection Address 12
pmpaddr13	32	0	rw	Physical Memory Protection Address 13
pmpaddr14	32	0	rw	Physical Memory Protection Address 14
pmpaddr15	32	0	rw	Physical Memory Protection Address 15
mcycle	32	0	rw	Machine Cycle Counter
minstret	32	0	rw	Machine Instructions Retired
mhpmcounter3	32	-	rw	Machine Performance Monitor Counter 3 (hardwired to zero)
mhpmcounter4	32	-	rw	Machine Performance Monitor Counter 4 (hardwired to zero)
mhpmcounter5	32	-	rw	Machine Performance Monitor Counter 5 (hardwired to zero)
mhpmcounter6	32	-	rw	Machine Performance Monitor Counter 6 (hardwired to zero)
mhpmcounter7	32	-	rw	Machine Performance Monitor Counter 7 (hardwired to zero)
mhpmcounter8	32	-	rw	Machine Performance Monitor Counter 8 (hardwired to zero)
mhpmcounter9	32	-	rw	Machine Performance Monitor Counter 9 (hardwired to zero)
mhpmcounter10	32	-	rw	Machine Performance Monitor Counter 10 (hardwired to zero)
mhpmcounter11	32	-	rw	Machine Performance Monitor Counter 11 (hardwired to zero)
mhpmcounter12	32	-	rw	Machine Performance Monitor Counter 12 (hardwired to zero)
mhpmcounter13	32	-	rw	Machine Performance Monitor Counter 13 (hardwired to zero)
mhpmcounter14	32	-	rw	Machine Performance Monitor Counter 14 (hardwired to zero)
mhpmcounter15	32	-	rw	Machine Performance Monitor Counter 15 (hardwired to zero)

mhpmcounter16	32	-	rw	Machine Performance Monitor Counter 16 (hardwired to zero)
mhpmcounter17	32	-	rw	Machine Performance Monitor Counter 17 (hardwired to zero)
mhpmcounter18	32	-	rw	Machine Performance Monitor Counter 18 (hardwired to zero)
mhpmcounter19	32	-	rw	Machine Performance Monitor Counter 19 (hardwired to zero)
mhpmcounter20	32	-	rw	Machine Performance Monitor Counter 20 (hardwired to zero)
mhpmcounter21	32	-	rw	Machine Performance Monitor Counter 21 (hardwired to zero)
mhpmcounter22	32	-	rw	Machine Performance Monitor Counter 22 (hardwired to zero)
mhpmcounter23	32	-	rw	Machine Performance Monitor Counter 23 (hardwired to zero)
mhpmcounter24	32	-	rw	Machine Performance Monitor Counter 24 (hardwired to zero)
mhpmcounter25	32	-	rw	Machine Performance Monitor Counter 25 (hardwired to zero)
mhpmcounter26	32	-	rw	Machine Performance Monitor Counter 26 (hardwired to zero)
mhpmcounter27	32	-	rw	Machine Performance Monitor Counter 27 (hardwired to zero)
mhpmcounter28	32	-	rw	Machine Performance Monitor Counter 28 (hardwired to zero)
mhpmcounter29	32	-	rw	Machine Performance Monitor Counter 29 (hardwired to zero)
mhpmcounter30	32	-	rw	Machine Performance Monitor Counter 30 (hardwired to zero)
mhpmcounter31	32	-	rw	Machine Performance Monitor Counter 31 (hardwired to zero)
mcycleh	32	0	rw	Machine Cycle Counter High
minstreth	32	0	rw	Machine Instructions Retired High
mhpmcounterh3	32	-	rw	Machine Performance Monitor Counter High 3 (hardwired to zero)
mhpmcounterh4	32	-	rw	Machine Performance Monitor Counter High 4 (hardwired to zero)
mhpmcounterh5	32	-	rw	Machine Performance Monitor Counter High 5 (hardwired to zero)
mhpmcounterh6	32	-	rw	Machine Performance Monitor Counter High 6 (hardwired to zero)
mhpmcounterh7	32	-	rw	Machine Performance Monitor Counter High 7 (hardwired to zero)
mhpmcounterh8	32	-	rw	Machine Performance Monitor Counter High 8 (hardwired to zero)

mhpmcounterh9	32	-	rw	Machine Performance Monitor Counter High 9 (hardwired to zero)
mhpmcounterh10	32	-	rw	Machine Performance Monitor Counter High 10 (hardwired to zero)
mhpmcounterh11	32	-	rw	Machine Performance Monitor Counter High 11 (hardwired to zero)
mhpmcounterh12	32	-	rw	Machine Performance Monitor Counter High 12 (hardwired to zero)
mhpmcounterh13	32	-	rw	Machine Performance Monitor Counter High 13 (hardwired to zero)
mhpmcounterh14	32	-	rw	Machine Performance Monitor Counter High 14 (hardwired to zero)
mhpmcounterh15	32	-	rw	Machine Performance Monitor Counter High 15 (hardwired to zero)
mhpmcounterh16	32	-	rw	Machine Performance Monitor Counter High 16 (hardwired to zero)
mhpmcounterh17	32	-	rw	Machine Performance Monitor Counter High 17 (hardwired to zero)
mhpmcounterh18	32	-	rw	Machine Performance Monitor Counter High 18 (hardwired to zero)
mhpmcounterh19	32	-	rw	Machine Performance Monitor Counter High 19 (hardwired to zero)
mhpmcounterh20	32	-	rw	Machine Performance Monitor Counter High 20 (hardwired to zero)
mhpmcounterh21	32	-	rw	Machine Performance Monitor Counter High 21 (hardwired to zero)
mhpmcounterh22	32	-	rw	Machine Performance Monitor Counter High 22 (hardwired to zero)
mhpmcounterh23	32	-	rw	Machine Performance Monitor Counter High 23 (hardwired to zero)
mhpmcounterh24	32	-	rw	Machine Performance Monitor Counter High 24 (hardwired to zero)
mhpmcounterh25	32	-	rw	Machine Performance Monitor Counter High 25 (hardwired to zero)
mhpmcounterh26	32	-	rw	Machine Performance Monitor Counter High 26 (hardwired to zero)
mhpmcounterh27	32	-	rw	Machine Performance Monitor Counter High 27 (hardwired to zero)
mhpmcounterh28	32	-	rw	Machine Performance Monitor Counter High 28 (hardwired to zero)
mhpmcounterh29	32	-	rw	Machine Performance Monitor Counter High 29 (hardwired to zero)
mhpmcounterh30	32	-	rw	Machine Performance Monitor Counter High 30 (hardwired to zero)
mhpmcounterh31	32	-	rw	Machine Performance Monitor Counter High 31 (hardwired to zero)

mhpmevent3	32	-	rw	Machine Performance Monitor Event Select 3 (hardwired to zero)
mhpmevent4	32	-	rw	Machine Performance Monitor Event Select 4 (hardwired to zero)
mhpmevent5	32	-	rw	Machine Performance Monitor Event Select 5 (hardwired to zero)
mhpmevent6	32	-	rw	Machine Performance Monitor Event Select 6 (hardwired to zero)
mhpmevent7	32	-	rw	Machine Performance Monitor Event Select 7 (hardwired to zero)
mhpmevent8	32	-	rw	Machine Performance Monitor Event Select 8 (hardwired to zero)
mhpmevent9	32	-	rw	Machine Performance Monitor Event Select 9 (hardwired to zero)
mhpmevent10	32	-	rw	Machine Performance Monitor Event Select 10 (hardwired to zero)
mhpmevent11	32	-	rw	Machine Performance Monitor Event Select 11 (hardwired to zero)
mhpmevent12	32	-	rw	Machine Performance Monitor Event Select 12 (hardwired to zero)
mhpmevent13	32	-	rw	Machine Performance Monitor Event Select 13 (hardwired to zero)
mhpmevent14	32	-	rw	Machine Performance Monitor Event Select 14 (hardwired to zero)
mhpmevent15	32	-	rw	Machine Performance Monitor Event Select 15 (hardwired to zero)
mhpmevent16	32	-	rw	Machine Performance Monitor Event Select 16 (hardwired to zero)
mhpmevent17	32	-	rw	Machine Performance Monitor Event Select 17 (hardwired to zero)
mhpmevent18	32	-	rw	Machine Performance Monitor Event Select 18 (hardwired to zero)
mhpmevent19	32	-	rw	Machine Performance Monitor Event Select 19 (hardwired to zero)
mhpmevent20	32	-	rw	Machine Performance Monitor Event Select 20 (hardwired to zero)
mhpmevent21	32	-	rw	Machine Performance Monitor Event Select 21 (hardwired to zero)
mhpmevent22	32	-	rw	Machine Performance Monitor Event Select 22 (hardwired to zero)
mhpmevent23	32	-	rw	Machine Performance Monitor Event Select 23 (hardwired to zero)
mhpmevent24	32	-	rw	Machine Performance Monitor Event Select 24 (hardwired to zero)
mhpmevent25	32	-	rw	Machine Performance Monitor Event Select 25 (hardwired to zero)

mhpmevent26	32	-	rw	Machine Performance Monitor Event Select 26 (hardwired to zero)
mhpmevent27	32	-	rw	Machine Performance Monitor Event Select 27 (hardwired to zero)
mhpmevent28	32	-	rw	Machine Performance Monitor Event Select 28 (hardwired to zero)
mhpmevent29	32	-	rw	Machine Performance Monitor Event Select 29 (hardwired to zero)
mhpmevent30	32	-	rw	Machine Performance Monitor Event Select 30 (hardwired to zero)
mhpmevent31	32	-	rw	Machine Performance Monitor Event Select 31 (hardwired to zero)
tselect	32	-	rw	Debug/Trace Trigger Register Select (not implemented)
tdata1	32	-	rw	Debug/Trace Trigger Data 1 (not implemented)
tdata2	32	-	rw	Debug/Trace Trigger Data 2 (not implemented)
tdata3	32	-	rw	Debug/Trace Trigger Data 3 (not implemented)
dcsr	32	-	rw	Debug Control and Status (not implemented)
dpc	32	-	rw	Debug PC (not implemented)
dscratch	32	-	rw	Debug Scratch (not implemented)

12.1.6 Integration_support

Table 15. Registers at level 1, type: , register group: 'Integration_support'

Name	Bits	Initial value (Hex)		Description
LRSCAddress	32	fffffff	rw	LR/SC active lock address

#