



OVP Guide to Using Processor Models

Model specific information for MIPS_5Kf

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



| | |
|----------|--|
| Author | Imperas Software Limited |
| Version | 20210408.0 |
| Filename | OVP_Model_Specific_Information_mips64.r1r5_5Kf.pdf |
| Created | 9 April 2021 |
| Status | OVP Standard Release |

Copyright Notice

Copyright (c) 2021 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 | Limitations | 1 |
| 1.4 | Verification | 2 |
| 1.5 | Features | 2 |
| 2 | Configuration | 3 |
| 2.1 | Location | 3 |
| 2.2 | GDB Path | 3 |
| 2.3 | Semi-Host Library | 3 |
| 2.4 | Processor Endian-ness | 3 |
| 2.5 | QuantumLeap Support | 3 |
| 2.6 | Processor ELF code | 3 |
| 3 | All Variants in this model | 4 |
| 4 | Bus Master Ports | 5 |
| 5 | Bus Slave Ports | 6 |
| 6 | Net Ports | 7 |
| 7 | FIFO Ports | 8 |
| 8 | Formal Parameters | 9 |
| 9 | Execution Modes | 13 |
| 10 | Exceptions | 14 |
| 11 | Hierarchy of the model | 15 |
| 11.1 | Level 1: CPU | 15 |
| 12 | Model Commands | 16 |
| 12.1 | Level 1: CPU | 16 |
| 12.1.1 | isync | 16 |
| 12.1.2 | itrace | 16 |
| 12.1.3 | mipsCOP0 | 16 |

| | | |
|-----------|----------------------|-----------|
| 12.1.4 | mipsCacheDisable | 17 |
| 12.1.4.1 | Argument description | 17 |
| 12.1.5 | mipsCacheEnable | 17 |
| 12.1.6 | mipsCacheRatio | 17 |
| 12.1.7 | mipsCacheReport | 17 |
| 12.1.7.1 | Argument description | 17 |
| 12.1.8 | mipsCacheReset | 17 |
| 12.1.8.1 | Argument description | 17 |
| 12.1.9 | mipsCacheTrace | 17 |
| 12.1.10 | mipsDebugFlags | 18 |
| 12.1.11 | mipsReadRegister | 18 |
| 12.1.12 | mipsReadTLBEntry | 18 |
| 12.1.13 | mipsTLBDump | 18 |
| 12.1.13.1 | Argument description | 18 |
| 12.1.14 | mipsTLBGetPhys | 18 |
| 12.1.15 | mipsWriteRegister | 19 |
| 12.1.16 | mipsWriteTLBEntry | 19 |
| 13 | Registers | 20 |
| 13.1 | Level 1: CPU | 20 |
| 13.1.1 | Core | 20 |
| 13.1.2 | FPU | 21 |
| 13.1.3 | DSP | 21 |
| 13.1.4 | COP0 | 22 |
| 13.1.5 | 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

MIPS64 Configurable Processor Model

1.2 Licensing

Usage of binary model under license governing simulator usage. Source of model available under Imperas Software License Agreement.

1.3 Limitations

Cache model not implemented on mips64 variants

If this model is not part of your installation, then it is available for download from www.OVPworld.org/MIPStuser.

1.4 Verification

Models have been validated correct as part of the MIPS Verified program and run through the MIPS AVP test programs

1.5 Features

only MIPS64 Instruction set implemented

MMU Type: Standard TLB

FPU implemented

Vectored interrupts implemented

Chapter 2

Configuration

2.1 Location

This model's VLVN is `mips.ovpworld.org/processor/mips64_r1r5/1.0`.

The model source is usually at:

`$IMPERAS_HOME/ImperasLib/source/mips.ovpworld.org/processor/mips64_r1r5/1.0`

The model binary is usually at:

`$IMPERAS_HOME/lib/$IMPERAS_ARCH/ImperasLib/mips.ovpworld.org/processor/mips64_r1r5/1.0`

2.2 GDB Path

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

2.3 Semi-Host Library

The default semi-host library file is `mips.ovpworld.org/semihosting/mips64Newlib/1.0`

2.4 Processor Endian-ness

This model can be set to either endian-ness (normally by a pin, or the ELF code).

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: `0x8`.

Chapter 3

All Variants in this model

This model has these variants

| Variant | Description |
|----------------|------------------------------|
| 5Kf | (described in this document) |
| 5Kc | |
| 5KEf | |
| 5KEc | |

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 | 12 | 36 | mandatory | |
| DATA | 12 | 36 | optional | |

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 | Core reset |
| dint | input | optional | Debug external interrupt |
| hwint0 | input | optional | External interrupt |
| hwint1 | input | optional | External interrupt |
| hwint2 | input | optional | External interrupt |
| hwint3 | input | optional | External interrupt |
| hwint4 | input | optional | External interrupt |
| hwint5 | input | optional | External interrupt |
| nmi | input | optional | Non-maskable external interrupt |
| vc_run | input | optional | Set to force stop of execution on processor VPE (simulation control only) |

Table 6.1: Net Ports

Chapter 7

FIFO Ports

This model has no FIFO ports.

Chapter 8

Formal Parameters

| Name | Type | Description |
|--------------------------|-------------|--|
| variant | Enumeration | Processor variant |
| endian | Endian | Model endian |
| mipsHexFile | String | Load a MIPS hex file (test-mode) |
| IMPERAS_MIPS_AVP_OPCODES | Boolean | Enable MIPS-specific magic Pass/Fail opcodes (specific for AVP test termination) |
| MIPS_TRACE | Boolean | Enable MIPS-format trace output |
| supervisorMode | Boolean | Override whether processor implements supervisor mode |
| busErrors | Boolean | Override bus error exception behavior. When true, accesses of memory not defined by platform will cause bus error exceptions |
| fixedMMU | Boolean | Override the MMU type to fixed mapping when true (sets Config.MT=3, Config.KU/K23=2 and Config1.MMUSizeM1=0) |
| removeDSP | Boolean | Override the DSP-present configuration when true (sets Config3.DSPP/DSP2P=0) |
| removeCMP | Boolean | Override the CMP-Present configuration when true (sets Config3.CMGCR and GCR_BASE to 0) |
| removeFP | Boolean | Override the FP-Present configuration when true (sets Config1.FP to 0) |
| isISA | Boolean | Enable to specify ISA model (reset address from ELF, all coprocessors enabled) |
| hiddenTLBentries | Boolean | Deprecated - Instead set config1MMUSizeM1 to maximum value to improve performance |
| ITCNumEntries | Uns32 | Specify number of ITC cells present (MT cores only) |
| ITCNumFIFO | Uns32 | Specify number of ITC FIFO cells in reference ITC implementation (MT cores only) |
| MTFPU | Uns32 | Enable multi-threaded FPU (1:old mttc1 behavior, 2:new mttc1 behavior) |
| supportDenormals | Boolean | Enable to specify that the FPU supports denormal operands and results |
| VPE0MaxTC | Uns32 | Specifies the maximum TCs initially on VPE0 |
| segBits | Uns32 | Override the number of address bits implemented for 64 bit segments (MIPS64 Only) |
| mpuRegions | Uns32 | Number of regions for memory protection unit |
| mpuType | Uns32 | Type of MPU implementation |
| mpuEnable | Boolean | Enable MPU2 segment control at reset |
| mpuSegment0 | Uns32 | Attributes for segment 0 in MPU2 SegmentControl.0 register |
| mpuSegment1 | Uns32 | Attributes for segment 1 in MPU2 SegmentControl.0 register |

| | | |
|------------------|---------|---|
| mpuSegment2 | Uns32 | Attributes for segment 2 in MPU2 SegmentControl_0 register |
| mpuSegment3 | Uns32 | Attributes for segment 3 in MPU2 SegmentControl_0 register |
| mpuSegment4 | Uns32 | Attributes for segment 4 in MPU2 SegmentControl_1 register |
| mpuSegment5 | Uns32 | Attributes for segment 5 in MPU2 SegmentControl_1 register |
| mpuSegment6 | Uns32 | Attributes for segment 6 in MPU2 SegmentControl_1 register |
| mpuSegment7 | Uns32 | Attributes for segment 7 in MPU2 SegmentControl_1 register |
| mpuSegment8 | Uns32 | Attributes for segment 8 in MPU2 SegmentControl_2 register |
| mpuSegment9 | Uns32 | Attributes for segment 9 in MPU2 SegmentControl_2 register |
| mpuSegment10 | Uns32 | Attributes for segment 10 in MPU2 SegmentControl_2 register |
| mpuSegment11 | Uns32 | Attributes for segment 11 in MPU2 SegmentControl_2 register |
| mpuSegment12 | Uns32 | Attributes for segment 12 in MPU2 SegmentControl_3 register |
| mpuSegment13 | Uns32 | Attributes for segment 13 in MPU2 SegmentControl_3 register |
| mpuSegment14 | Uns32 | Attributes for segment 14 in MPU2 SegmentControl_3 register |
| mpuSegment15 | Uns32 | Attributes for segment 15 in MPU2 SegmentControl_3 register |
| mvpconf0vpe | Uns32 | Override MVPConf0.PVPE |
| mvpconf0tc | Uns32 | Override MVPConf0.PTC |
| mvpconf0pcp | Boolean | Override MVPConf0.PCP |
| mvpconf0tcp | Boolean | Override MVPConf0.TCP |
| hasFDC | Uns32 | Specify the size of Fast Debug Channel register block |
| statusFR | Boolean | Override power on value in Status.FR (Floating point register mode) |
| configDSP | Boolean | Override Config.DSP (data scratchpad RAM present) |
| configISP | Boolean | Override Config.ISP (instruction scratchpad RAM present) |
| configK0 | Uns32 | Override power on value of Config.K0 (set Kseg0 cacheability) |
| configKU | Uns32 | Override power on value of Config.KU (set Useg cacheability) |
| configK23 | Uns32 | Override power on value of Config.K23 (set Kseg23 cacheability) |
| configMDU | Boolean | Override Config.MDU (iterative multiply/divide unit) |
| configMM | Boolean | Override Config.MM (merging mode for write) |
| configMT | Uns32 | Override Config.MT |
| configSB | Boolean | Override Config.SB (simple bus transfers only) |
| MIPS16eASE | Boolean | Override Config1.CA (enables the MIPS16e ASE) |
| config1EP | Boolean | Override Config1.EP (EJTag present) |
| config1MMUSizeM1 | Uns32 | Override Config1.MMUSizeM1 (number of MMU entries-1) |
| config1WR | Boolean | Override Config1.WR (watchpoint registers present) |
| config1FP | Boolean | Override Config1.FP (FPU present) |
| config3BI | Boolean | Override Config3.BI |
| config3BP | Boolean | Override Config3.BP |

| | | |
|--------------------|---------|---|
| config3CDMM | Boolean | Override Config3.CDMM |
| config3CTXTC | Boolean | Override Config3.CTXTC |
| config3DSPP | Boolean | Override Config3.DSPP |
| config3DSP2P | Boolean | Override Config3.DSP2P |
| config3IPLW | Uns32 | Override Config3.IPLW |
| config3ISA | Uns32 | Override Config3.ISA |
| config3ISAOnExc | Boolean | Override Config3.ISAOnExc |
| config3ITL | Boolean | Override Config3.ITL |
| config3LPA | Boolean | Override Config3.LPA |
| config3MCU | Boolean | Override Config3.MCU |
| config3MMAR | Uns32 | Override Config3.MMAR |
| config3RXI | Boolean | Override Config3.RXI |
| config3SC | Boolean | Override Config3.SC |
| config3ULRI | Boolean | Override Config3.ULRI |
| externalinterrupt | Boolean | Override Config3.VEIC (enables the use of an external interrupt controller) |
| vectoredinterrupt | Boolean | Override Config3.VInt (enables vectored interrupts) |
| config3VZ | Boolean | Override Config3.VZ |
| config4AE | Boolean | Override Config4.AE |
| config4IE | Uns32 | Override Config4.IE |
| config4MMUConfig | Uns32 | Override Config4.MMUConfig field (interpretation depends on MMUExtDef value) |
| config4MMUExtDef | Uns32 | Override Config4.MMUExtDef |
| config4VTLBSizeExt | Uns32 | Override Config4.VTLBSizeExt |
| config5EVA | Boolean | Override Config5.EVA |
| config5NFExists | Boolean | Override Config5.NFExists |
| config5MSAEn | Boolean | Override Config5.MSAEn |
| config6FTLBEEn | Boolean | Override power on value of Config6.FTLBEEn |
| config7DCIDX_MODE | Uns32 | Override Config7.DCIDX_MODE |
| config7WII | Boolean | Override Config7.WII (wait IE/IXMT ignore) |
| fcsrABS2008 | Boolean | Override FCSR.ABS2008 (ABS/NEG compliant with IEEE 754-2008) |
| fcsrNAN2008 | Boolean | Override FCSR.NAN2008 (QNaN/SNaN encodings match IEEE 754-2008 recommendation) |
| firPS | Boolean | Override FIR.PS (PS floating point type implemented) |
| firHas2008 | Boolean | Override FIR.Has2008 (one or more IEEE 754-2008 features present) |
| intctlIPFDC | Uns32 | Override IntCtl.IPFDC |
| intctlIPTI | Uns32 | Override IntCtl.IPTI |
| pridRevision | Uns32 | Override PRId.Revision |
| srscctlHSS | Uns32 | Override SRSCtl.HSS (number of shadow register sets) |
| ExceptionBase | Uns32 | Specify the BEV Exception Base address. (use GCR_Cx_RESET_BASE on CMP processors) |
| UseExceptionBase | Boolean | Set to one to use ExceptionBase[29:12] as the corresponding BEV address bits |
| EIC_OPTION | Uns32 | Override the external interrupt controller EIC_OPTION |
| ISPRAM_SIZE | Uns32 | Encoded size of the ISPRAM region (log2(<ISPRAM size in bytes>) - 11) |
| ISPRAM_BASE | Uns64 | Starting physical address of the ISPRAM region |
| ISPRAM_ENABLE | Boolean | Set the enable bit of the ISPRAM region's tag (used to enable the ISPRAM region prior to reset) |
| ISPRAM_FILE | String | Load a MIPS hex file into the ISPRAM region prior to reset |
| DSPRAM_SIZE | Uns32 | Encoded size of the DSPRAM region (log2(<DSPRAM size in bytes>) - 11) |
| DSPRAM_BASE | Uns64 | Starting physical address of the DSPRAM region |

| | | |
|---------------|---------|---|
| DSPRAM_ENABLE | Boolean | Set the enable bit of the DSPRAM region's tag (used to enable the DSPRAM region prior to reset) |
|---------------|---------|---|

Table 8.1: Parameters that can be set in: CPU

Chapter 9

Execution Modes

| Mode | Code |
|------------|------|
| KERNEL | 0 |
| DEBUG | 1 |
| SUPERVISOR | 2 |
| USER | 3 |

Table 9.1: Modes implemented in: CPU

Chapter 10

Exceptions

| Exception | Code |
|-----------|------|
| Int | 0 |
| Mod | 1 |
| TLBL | 2 |
| TLBS | 3 |
| AdEL | 4 |
| AdES | 5 |
| IBE | 6 |
| DBE | 7 |
| Sys | 8 |
| Bp | 9 |
| RI | 10 |
| CpU | 11 |
| Ov | 12 |
| Tr | 13 |
| FPE | 15 |
| Impl1 | 16 |
| Impl2 | 17 |
| C2E | 18 |
| TLBRI | 19 |
| TLBXI | 20 |
| MDMX | 22 |
| WATCH | 23 |
| MCheck | 24 |
| Thread | 25 |
| DSPDis | 26 |
| Prot | 29 |
| CacheErr | 30 |

Table 10.1: Exceptions implemented in: CPU

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: CPU

This level in the model hierarchy has 16 commands.

This level in the model hierarchy has 5 register groups:

| Group name | Registers |
|---------------------|-----------|
| Core | 33 |
| FPU | 34 |
| DSP | 9 |
| COP0 | 32 |
| 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: CPU

12.1.1 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

12.1.2 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

12.1.3 mipsCOP0

query a COP0 register value using <register><select>

| Argument | Type | Description |
|-----------|-------|----------------------------------|
| -register | Uns32 | specify the COP0 register number |

| | | |
|---------|-------|----------------------------------|
| -select | Uns32 | specify the COP0 register select |
|---------|-------|----------------------------------|

Table 12.3: mipsCOP0 command arguments

12.1.4 mipsCacheDisable

12.1.4.1 Argument description

Disables tag or full cache model

12.1.5 mipsCacheEnable

enable tag or full cache model

| Argument | Type | Description |
|----------|---------|----------------------------------|
| -debug | Uns32 | set cache model debug flags |
| -full | Boolean | enable full cache model |
| -tag | Boolean | enable cache tag line only model |

Table 12.4: mipsCacheEnable command arguments

12.1.6 mipsCacheRatio

Report current hit ratio for selected cache

| Argument | Type | Description |
|----------|---------|-----------------------------|
| -dcache | Boolean | report hit ratio for dcache |
| -icache | Boolean | report hit ratio for icache |

Table 12.5: mipsCacheRatio command arguments

12.1.7 mipsCacheReport

12.1.7.1 Argument description

Report current cache statistics

12.1.8 mipsCacheReset

12.1.8.1 Argument description

reset the cache model

12.1.9 mipsCacheTrace

Control the tracing of cache accesses

| Argument | Type | Description |
|-------------|---------|--------------------------|
| -noartifact | Boolean | filter artifact accesses |
| -nocached | Boolean | filter cached accesses |
| -nodcache | Boolean | filter dcache accesses |

| | | |
|-------------|---------|----------------------------|
| -noicache | Boolean | filter icache accesses |
| -notrue | Boolean | filter true accesses |
| -nouncached | Boolean | filter uncached accesses |
| -off | Boolean | turn off the cache tracing |
| -on | Boolean | turn on the cache tracing |

Table 12.6: mipsCacheTrace command arguments

12.1.10 mipsDebugFlags

Set the processor model debug flags to <value>

| Argument | Type | Description |
|----------|-------|---------------------------|
| -value | Uns32 | specify model debug flags |

Table 12.7: mipsDebugFlags command arguments

12.1.11 mipsReadRegister

Read a processor register using <resource><offset>

| Argument | Type | Description |
|-----------|-------|--|
| -offset | Uns32 | the processor register offset |
| -resource | Uns32 | the processor register resource number |

Table 12.8: mipsReadRegister command arguments

12.1.12 mipsReadTLBEntry

read a TLB entry specified by the index

| Argument | Type | Description |
|----------|-------|----------------------|
| -index | Uns64 | select the TLB entry |

Table 12.9: mipsReadTLBEntry command arguments

12.1.13 mipsTLBDump

12.1.13.1 Argument description

Dumps the current contents of the TLB

12.1.14 mipsTLBGetPhys

Reports the entry(s) in the TLB that match the given virtual address and ASID

| Argument | Type | Description |
|----------|-------|-----------------|
| -asid | Uns64 | ASID |
| -va | Uns64 | virtual address |

Table 12.10: mipsTLBGetPhys command arguments

12.1.15 mipsWriteRegister

Write to a processor register using <resource><offset><value>

| Argument | Type | Description |
|-----------|-------|----------------------------------|
| -offset | Uns32 | the register offset number |
| -resource | Uns32 | the register resource number |
| -value | Uns64 | the register value to be written |

Table 12.11: mipsWriteRegister command arguments

12.1.16 mipsWriteTLBEntry

Writes values to a TLB entry using the index, lo0, lo1, hi0 and mask fields

| Argument | Type | Description |
|----------|-------|-----------------------------|
| -hi0 | Uns64 | the TLB entry high address |
| -index | Uns64 | the TLB entry index |
| -lo0 | Uns64 | the TLB entry low address 0 |
| -lo1 | Uns64 | the TLB entry low address 1 |
| -mask | Uns64 | the TLB entry mask |

Table 12.12: mipsWriteTLBEntry command arguments

Chapter 13

Registers

13.1 Level 1: CPU

13.1.1 Core

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

| Name | Bits | Initial-Hex | RW | Description |
|------|------|--------------------|----|-----------------|
| zero | 64 | 0 | r- | constant zero |
| at | 64 | 0 | rw | |
| v0 | 64 | 0 | rw | |
| v1 | 64 | 0 | rw | |
| a0 | 64 | 0 | rw | |
| a1 | 64 | 0 | rw | |
| a2 | 64 | 0 | rw | |
| a3 | 64 | 0 | rw | |
| t0 | 64 | 0 | rw | |
| t1 | 64 | 0 | rw | |
| t2 | 64 | 0 | rw | |
| t3 | 64 | 0 | rw | |
| t4 | 64 | 0 | rw | |
| t5 | 64 | 0 | rw | |
| t6 | 64 | 0 | rw | |
| t7 | 64 | 0 | rw | |
| s0 | 64 | 0 | rw | |
| s1 | 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 | |
| t8 | 64 | 0 | rw | |
| t9 | 64 | 0 | rw | |
| k0 | 64 | 0 | rw | |
| k1 | 64 | 0 | rw | |
| gp | 64 | 0 | rw | |
| sp | 64 | 0 | rw | stack pointer |
| s8 | 64 | 0 | rw | frame pointer |
| ra | 64 | 0 | rw | |
| pc | 64 | ffffff bfc00000 | rw | program counter |

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

13.1.2 FPU

Registers at level:1, type:CPU group:FPU

| Name | Bits | Initial-Hex | RW | Description |
|------|------|-------------|----|----------------------------|
| f0 | 64 | 0 | rw | |
| f1 | 64 | 0 | rw | |
| f2 | 64 | 0 | rw | |
| f3 | 64 | 0 | rw | |
| f4 | 64 | 0 | rw | |
| f5 | 64 | 0 | rw | |
| f6 | 64 | 0 | rw | |
| f7 | 64 | 0 | rw | |
| f8 | 64 | 0 | rw | |
| f9 | 64 | 0 | rw | |
| f10 | 64 | 0 | rw | |
| f11 | 64 | 0 | rw | |
| f12 | 64 | 0 | rw | |
| f13 | 64 | 0 | rw | |
| f14 | 64 | 0 | rw | |
| f15 | 64 | 0 | rw | |
| f16 | 64 | 0 | rw | |
| f17 | 64 | 0 | rw | |
| f18 | 64 | 0 | rw | |
| f19 | 64 | 0 | rw | |
| f20 | 64 | 0 | rw | |
| f21 | 64 | 0 | rw | |
| f22 | 64 | 0 | rw | |
| f23 | 64 | 0 | rw | |
| f24 | 64 | 0 | rw | |
| f25 | 64 | 0 | rw | |
| f26 | 64 | 0 | rw | |
| f27 | 64 | 0 | rw | |
| f28 | 64 | 0 | rw | |
| f29 | 64 | 0 | rw | |
| f30 | 64 | 0 | rw | |
| f31 | 64 | 0 | rw | |
| fsr | 64 | 0 | rw | floating point status |
| fir | 64 | 739300 | r- | floating point information |

Table 13.2: Registers at level 1, type:CPU group:FPU

13.1.3 DSP

Registers at level:1, type:CPU group:DSP

| Name | Bits | Initial-Hex | RW | Description |
|------|------|-------------|----|-------------|
| lo | 64 | 0 | rw | |
| hi | 64 | 0 | rw | |
| lo1 | 64 | 0 | rw | |
| hi1 | 64 | 0 | rw | |
| lo2 | 64 | 0 | rw | |

| | | | | |
|--------|----|---|----|-------------|
| hi2 | 64 | 0 | rw | |
| lo3 | 64 | 0 | rw | |
| hi3 | 64 | 0 | rw | |
| dspctl | 64 | 0 | rw | DSP control |

Table 13.3: Registers at level 1, type:CPU group:DSP

13.1.4 COP0

Registers at level:1, type:CPU group:COP0

| Name | Bits | Initial-Hex | RW | Description |
|----------|------|--------------------|----|------------------------------|
| sr | 64 | 400004 | rw | CP0 register 12/0 (status) |
| bad | 64 | 0 | rw | CP0 register 8/0 (badvaaddr) |
| cause | 64 | 0 | rw | CP0 register 13/0 (cause) |
| index | 64 | 0 | rw | CP0 register 0/0 |
| random | 64 | 0 | rw | CP0 register 1/0 |
| entrylo0 | 64 | 0 | rw | CP0 register 2/0 |
| entrylo1 | 64 | 0 | rw | CP0 register 3/0 |
| context | 64 | 0 | rw | CP0 register 4/0 |
| pagemask | 64 | 0 | rw | CP0 register 5/0 |
| wired | 64 | 0 | rw | CP0 register 6/0 |
| hwrena | 64 | 0 | rw | CP0 register 7/0 |
| badvaddr | 64 | 0 | rw | CP0 register 8/0 |
| count | 64 | 0 | rw | CP0 register 9/0 |
| entryhi | 64 | 0 | rw | CP0 register 10/0 |
| compare | 64 | 0 | rw | CP0 register 11/0 |
| status | 64 | 400004 | rw | CP0 register 12/0 |
| intctl | 64 | fc000000 | rw | CP0 register 12/1 |
| srctl | 64 | 0 | rw | CP0 register 12/2 |
| srsmap | 64 | 0 | rw | CP0 register 12/3 |
| epc | 64 | 0 | rw | CP0 register 14/0 |
| prid | 64 | 18100 | rw | CP0 register 15/0 |
| ebase | 64 | ffffff 80000000 | rw | CP0 register 15/1 |
| config | 64 | b600c483 | rw | CP0 register 16/0 |
| config1 | 64 | dec37183 | rw | CP0 register 16/1 |
| config2 | 64 | 80000000 | rw | CP0 register 16/2 |
| config3 | 64 | 20 | rw | CP0 register 16/3 |
| lladdr | 64 | 0 | rw | CP0 register 17/0 |
| xcontext | 64 | 0 | rw | CP0 register 20/0 |
| debug | 64 | 2010000 | rw | CP0 register 23/0 |
| depc | 64 | 0 | rw | CP0 register 24/0 |
| errorepc | 64 | 0 | rw | CP0 register 30/0 |
| desave | 64 | 0 | rw | CP0 register 31/0 |

Table 13.4: Registers at level 1, type:CPU group:COP0

13.1.5 Integration_support

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

| Name | Bits | Initial-Hex | RW | Description |
|------|------|-------------|----|---------------------------------------|
| stop | 32 | 0 | rw | write with non-zero to stop processor |

Table 13.5: Registers at level 1, type:CPU group:Integration_support