OVP Guide to Using Processor Models

Model Specific Information for variant MIPS32_MIPS32R6

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_mips32_MIPS32R6.pdf
Created	23 February 2018
Status	OVP Standard Release

Copyright Notice

Right to Copy Documentation

Destination Control Statement

Disclaimer

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

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

MIPS32 Configurable Processor Model
If you need other variants, these models can be obtained from www.OVPworld.org/MIPSuser.

1.2 Licensing

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

1.3 Limitations

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

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

Both MIPS32 and microMIPS32 Instruction sets implemented
MMU Type: Standard TLB
L1 I and D cache model in either full or tag-only mode implemented (disabled by default)
Vectored interrupts implemented
MCU ASE implemented
DSP ASE Rev 2 implemented

2 Configuration

2.1 Location

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

imgtec.ovpworld.org/processor/mips32/1.0

2.2 GDB Path

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

2.3 Semi-Host Library

The default semi-host library file is found in the VLNV tree at :
mips.ovpworld.org/semihosting/mips32Newlib/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

3 Other Variants in this Model

Table 1. All variants in this model

Variant
M5100
.M5100Guest
M5150
.M5150Guest
M6200
M6250
MIPS32R6
.MIPS32R6Guest
P5600
.P5600Guest

4 Bus Ports

Table 2. Bus Ports

Type	Name	min	max	Description
master (initiator)	ISPRAM	32	32	instruction scratchpad RAM
master (initiator)	DSPRAM	32	32	data scratchpad RAM
master (initiator)	INSTRUCTION	32	36	
master (initiator)	DATA	32	36	

5 Net Ports

Table 3. Net Ports

Name	Type	Description
reset	input	Core reset
dint	input	Debug external interrupt
hwint0	input	External interrupt
hwint1	input	External interrupt
hwint2	input	External interrupt
hwint3	input	External interrupt
hwint4	input	External interrupt
hwint5	input	External interrupt
hwint6	input	External interrupt
hwint7	input	External interrupt
nmi	input	Non-maskable external interrupt
EICPresent	input	Input signal SI_EICPresent per VPE
EIC_RIPL	input	External interrupt controller RIPL (alias of hwint0 - 5 or 7)
EIC_EICSS	input	External interrupt controller EICSS
EIC_VectorNum	input	External interrupt controller vector number
EIC_VectorOffset	input	External interrupt controller vector offset
EIC_GID	input	External interrupt controller guest ID
intISS	output	True when interrupt request is serviced
causeTI	output	True when timer interrupt expires
causeIP0	output	Raised for software interrupt request IP0
causeIP1	output	Raised for software interrupt request IP1
si_sleep	output	True when the VPE is in WAIT state
Guest.EIC_RIPL	input	Guest External interrupt controller RIPL
Guest.EIC_EICSS	input	Guest External interrupt controller EICSS
Guest.EIC_VectorNum	input	Guest External interrupt controller vector number
Guest.EIC_VectorOffset	input	Guest External interrupt controller vector offset
Guest.EIC_GID	input	Guest External interrupt controller guest ID
Guest.intISS	output	True when Guest interrupt request is serviced
Guest.causeTI	output	True when Guest timer interrupt expires
Guest.causeIP0	output	Raised for Guest software interrupt request IP0
Guest.causeIP1	output	Raised for Guest software interrupt request IP1

6 FIFO Ports

No FIFO Ports in this model.

7 Parameters

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

Name	Type	Description
cacheenable	Enumeration	Select cache model mode default=0 tag=1 full=2
cachedebug	Uns32	Cache debug flags
cacheextbiuinfo	Pointer	Pointer to platform-provided BIU cache info structure
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)
cacheIndexBypassTLB	Boolean	When set, cache index ops do not generate TLB exceptions
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)
fixedDbgRegSize	Boolean	Enable applications to debug on P5600 with GDB version 2015.06-05 and prior
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)
removeFTLB	Boolean	Override the FTLBEn configuration when true (disable FTLB)
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
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
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
licenseWarningDays	Uns32	Specify the number of days before a license expires to start issuing a warning. 0 disables warnings.
MIPS_UHI	Boolean	Enable MIPS-Unified Hosting interface
mipsUhiArgs	String	Specifies UHI arguments string seperated by spaces
mipsUhiJail	String	Specifies UHI jailroot
MIPS_DV_MODE	Boolean	Enable Design Verification mode
MIPS_MAGIC_OPCODES	Boolean	Enable MIPS-specific magic Pass/Fail opcodes

enableTrickbox	Boolean	Enable trickbox addresses (specific for AVP)
fpuexcdisable	Boolean	Disable FPU exceptions
TRU_PRESENT	Boolean	Disable or Enable based on TRU presence to control certain fields (e.x.perfCtl.PCTD
ucLLwordsLocked	Uns32	Numbers of words (4 byte) an uncached LL is locking. Maximum: 4K
cop2Bits	Uns32	Specifies width in bits of COP2 registers (32 or 64)
cop2FileName	String	Specifies COP2 dynamically-loaded object (.so/.dll) defining COP2 instructions
udiConfig	Int32	Specifies UDI configuration attribute
udiFileName	String	Specifies UDI dynamically-loaded object (.so/.dll) defining UDI instructions
vectoredinterrupt	Boolean	Enables vectored interrupts (sets Config3 VInt)
externalinterrupt	Boolean	Enables the use of an external interrupt controller (sets Config3 VEIC)
rootFixedMMU	Boolean	Override the root MMU type to fixed mapping when true (sets Config.MT=3 and Config.KU/K23=2)
rootMMUSizeM1	Uns32	Override the root MMUSizeM1 field in Config1 register (number of MMU entries-1)
srsctlHSS	Uns32	Override the HSS field in SRSCtl register (number of shadow register sets)
firPS	Uns32	Override the PS field in FIR register
firHas2008	Uns32	Override the Has2008 field in FIR register
simulateLite	Uns32	Run Simulation with no overhead. By default runs with disabled Forbidden slot for R6 cores when set to 1.
pridCompanyOptions	Uns32	Override the Company Options field in PRId register
pridRevision	Uns32	Override the Revision field in PRId register
globalClusterNum	Uns32	Override the ClusterNum field in GlobalNumber register
intctlIPTI	Uns32	Override the IPTI field in IntCtl register
intctlIPFDC	Uns32	Override the IPFDC field in IntCtl register
intctlIPPCI	Uns32	Override the IPPCI field in IntCtl register
numWatch	Uns32	Specify number of WatchLo/WatchHi register pairs
xconfigSpecified	Boolean	True if the configuration comes from a valid xconfig file
cdmmSize	Uns32	Override the cdmmsize reset value
configAR	Uns32	Enables R6 support
configBM	Uns32	Override the BM field in Config register (burst 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)
configBCP	Boolean	Override Config.BCP (Buffer Cache Present)
MIPS16eASE	Boolean	Override Config1.CA (enables the MIPS16e ASE)
config1DA	Uns32	Override Config1.DA (Dcache associativity)
config1DL	Uns32	Override Config1.DL (Dcache line size)
config1DS	Uns32	Override Config1.DS (Dcache sets per way)
config1EP	Boolean	Override Config1.EP (EJTag present)
config1IA	Uns32	Override Config1.IA (Icache associativity)
config1IL	Uns32	Override Config1.IL (Icache line size)
config1IS	Uns32	Override Config1.IS (Icache sets per way)
config1MMUSizeM1	Uns32	Override Config1.MMUSizeM1 (number of MMU entries-1)
config1WR	Boolean	Override Config1.WR (watchpoint registers present)
config1PC	Boolean	Override Config1.PC (Performance Counters present)
config1C2	Boolean	Override Config1.C2 (Coprocessor 2 present)
config2SU	Uns32	Override the SU field in Config2 register
config2SS	Uns32	Override the SS field in Config2 register
config2SL	Uns32	Override the SL field in Config2 register
config2SA	Uns32	Override the SA field in Config2 register
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
config3VZ	Boolean	Override Config3.VZ
config3MSAP	Boolean	Override Config3.MSAP
config3CMGCR	Boolean	Override the CMGCR field in Config3 register
config3SP	Boolean	Override the SP field in Config3 register
config3TL	Uns32	Override the TL field in Config3 register
config3PW	Boolean	Override the PW field in Config3 register
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
config5LLB	Boolean	Override Config5.LLB (LLAddr supports LLbit)
config5MRP	Boolean	Override Config5.MRP (MaaR Present)
config5NFExists	Boolean	Override Config5.NFExists
mips32Macro	Boolean	Enables the MIPS32 SAVE and RESTORE macro instructions. Ignored if Config5.CA2 is not set)
config5MSAEn	Boolean	Override Config5.MSAEn
config5MVH	Boolean	Override Config5.MVH (enable MTHC0 and MFHC0 instructions)
config5DEC	Boolean	Override Config5.DEC (to test Dual Endian Capability)
config5GI	Uns32	Override Config5.GI (enable GINV)
config5CRCP	Boolean	Override Config5.CRCP (CRCP Present)
config6FTLBEn	Boolean	Override power on value of Config6.FTLBEn
config7AR	Boolean	Override Config7.AR (Alias removed Data cache)
config7DCIDX_MODE	Uns32	Override Config7.DCIDX_MODE
config7HCI	Boolean	Override Config7.HCI (Hardware Cache Initialization)
config7IAR	Boolean	Override Config7.IAR (Alias removed Instruction cache)
config7WII	Boolean	Override Config7.WII (wait IE/IXMT ignore)

config7ES	Uns32	Override the ES field in Config7 register (Externalize sync)
statusFR	Boolean	Override power on value in Status.FR (Floating point register mode)
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)
numMaarRegs	Uns32	Override number of MAAR registers (must be even)
wiredLimit	Uns32	Override Limit field of the Wired register
cdmmBaseCI	Boolean	Override CDMMBase.CI
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
firstBEVExceptionBaseMaskBit	Uns32	Specify LSB position of GCR_Cx_RESET_EXT_BASE.BEVExceptionBaseMask field. Only used when SegCtl present
EVAReset	Boolean	Set to one to reset into non-legacy address map and BEV location. Only used when non-CMP and SegCtl present
ExceptionBaseMask	Uns32	Specify the ExceptionBaseMask value used for bits [27:firstBEVExceptionBaseMaskBit]. Only used when non-CMP and SegCtl present
ExceptionBasePA	Uns32	Bits [35:29] of the physical address for the BEV overlays. Only used when non-CMP and SegCtl present
l1BufferCache	Boolean	L1 Buffer Cache
EIC_OPTION	Uns32	Override the external interrupt controller EIC_OPTION
guestCtl0RI	Uns32	Override the RI field in GuestCtl0 register
guestCtl0MC	Uns32	Override the MC field in GuestCtl0 register
guestCtl0CP0	Uns32	Override the CP0 field in GuestCtl0 register
guestCtl0AT	Uns32	Override the AT field in GuestCtl0 register
guestCtl0GT	Uns32	Override the GT field in GuestCtl0 register
guestCtl0CG	Uns32	Override the CG field in GuestCtl0 register
guestCtl0CF	Uns32	Override the CF field in GuestCtl0 register
guestCtl0G1	Uns32	Override the G1 field in GuestCtl0 register
guestCtl0RAD	Uns32	Override the RAD field in GuestCtl0 register
guestCtl0DRG	Uns32	Override the DRG field in GuestCtl0 register
hasImpl17	Boolean	Enable read/write of Impl17 bit in Status register

hasImpl16	Boolean	Enable read/write of Impl16 bit in Status register
guestintctlIPTI	Uns32	Override the Guest IPTI field in IntCtl register
guestintctlIPFDC	Uns32	Override the Guest IPFDC field in IntCtl register
guestintctlIPPCI	Uns32	Override the Guest IPPCI field in IntCtl register
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)
DSPRAM_PRESENT	Boolean	DSPRAM is present with SAAR
misalignedDataException	Enumeration	Select misaligned data access exception signaling: never, checkCCA or always never=0 checkCCA=1 always=2
commitTlbwErr	Boolean	Commit TLBWI/TLBRI on ECC; in MIPS_DV_MODE only

8 Execution Modes

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

Name	Code
KERNEL	0
DEBUG	1
SUPERVISOR	2
USER	3
GUEST_KERNEL	4
GUEST_SUPERVISOR	5
GUEST_USER	6

9 Exceptions

Table 6. Exceptions handled by the model, type: CPU

Name	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
GE	27
Prot	29
CacheErr	30

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: CPU

This level in the model hierarchy has 20 commands.

This level in the model hierarchy has 6 register groups:

Table 7. Register groups

Group name	Registers
Core	33
DSP	9
Shadow	64
COP0	138
SPRAM	15
Integration_support	1

This level in the model hierarchy has no children.

11 Model Commands

11.1 Level 1: CPU

11.1.1 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

11.1.2 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

11.1.3 mipsCOP0

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

Table 10. mipsCOP0 command arguments

Argument	Type	Description
-register	Int32	specify the COP0 register resource
-select	Int32	specify the COP0 register select

11.1.4 mipsCacheDisable

11.1.4.1 Argument description

Disables tag or full cache model

11.1.5 mipsCacheEnable

enable tag or full cache model

Table 11. mipsCacheEnable command arguments

Argument	Type	Description

-debug	Int32	set cache model debug flags
-full	Boolean	enable full cache model
-tag	Boolean	enable cache tag line only model

11.1.6 mipsCacheRatio

Report current hit ratio for selected cache

Table 12. mipsCacheRatio command arguments

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

11.1.7 mipsCacheReport

11.1.7.1 Argument description

Report current cache statistics

11.1.8 mipsCacheReset

11.1.8.1 Argument description

reset the cache model

11.1.9 mipsCacheTrace

Control the tracing of cache accesses

Table 13. mipsCacheTrace command arguments

Argument	Type	Description
-noartifact	Boolean	
-nocached	Boolean	
-nodcache	Boolean	
-noicache	Boolean	
-notrue	Boolean	
-nounchched	Boolean	
-off	Boolean	turn off the cache tracing
-on	Boolean	turn on the cache tracing

11.1.10 mipsDebugFlags

Set the mips model debug value

Table 14. mipsDebugFlags command arguments

Argument	Type	Description
-value	Int32	specify mips model debug flags

11.1.11 mipsReadRegister

Read processor register using <resource> <offset>

Table 15. mipsReadRegister command arguments

Argument	Type	Description
-offset	Int32	the register offset
-resource	Int32	the register resource

11.1.12 mipsReadTLBEntry

read a TLB entry specified by the index

Table 16. mipsReadTLBEntry command arguments

Argument	Type	Description
-index	Uns64	select the TLB entry

11.1.13 mipsTLBDump

11.1.13.1 Argument description

Dumps the current contents of the TLB

11.1.14 mipsTLBDumpGuest

11.1.14.1 Argument description

Dumps the current contents of the Guest TLB

11.1.15 mipsTLBDumpRoot

11.1.15.1 Argument description

Dumps the current contents of the Root TLB

11.1.16 mipsTLBGetPhys

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

Table 17. mipsTLBGetPhys command arguments

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

11.1.17 mipsTraceGuest

control tracing of guest

Table 18. mipsTraceGuest command arguments

Argument	Type	Description
-off	Boolean	stop tracing
-on	Boolean	start tracing

11.1.18 mipsTraceRoot

control tracing on root processor

Table 19. mipsTraceRoot command arguments

Argument	Type	Description
-off	Boolean	stop tracing
-on	Boolean	start tracing

11.1.19 mipsWriteRegister

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

Table 20. mipsWriteRegister command arguments

Argument	Type	Description
-offset	Int32	the register offset
-resource	Int32	the register resource
-value	Uns64	the value to write to register

11.1.20 mipsWriteTLBEntry

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

Table 21. mipsWriteTLBEntry command arguments

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

12 Registers

12.1 Level 1: CPU

12.1.1 Core

Table 22. Registers at level 1, type: CPU, register group: 'Core'

Name	Bits	Initial value (Hex)		Description
zero	32	0	r-	constant zero
at	32	0	rw	
v0	32	0	rw	
v1	32	0	rw	
a0	32	0	rw	
a1	32	0	rw	

a2	32	0	rw	
a3	32	0	rw	
t0	32	0	rw	
t1	32	0	rw	
t2	32	0	rw	
t3	32	0	rw	
t4	32	0	rw	
t5	32	0	rw	
t6	32	0	rw	
t7	32	0	rw	
s0	32	0	rw	
s1	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	
t8	32	0	rw	
t9	32	0	rw	
k0	32	0	rw	
k1	32	0	rw	
gp	32	0	rw	
sp	32	0	rw	stack pointer
s8	32	0	rw	frame pointer
ra	32	0	rw	
pc	32	bfc00000	rw	program counter

12.1.2 DSP

Table 23. Registers at level 1, type: CPU, register group: 'DSP'

Name	Bits	Initial value (Hex)		Description
lo	32	0	rw	
hi	32	0	rw	
lo1	32	0	rw	
hi1	32	0	rw	
lo2	32	0	rw	
hi2	32	0	rw	
lo3	32	0	rw	
hi3	32	0	rw	
dspctl	32	0	rw	DSP control

12.1.3 Shadow

Table 24. Registers at level 1, type: CPU, register group: 'Shadow'

Name	Bits	Initial value (Hex)		Description
zero[0]	32	0	r-	constant zero
at[0]	32	0	rw	
v0[0]	32	0	rw	
v1[0]	32	0	rw	
a0[0]	32	0	rw	
a1[0]	32	0	rw	
a2[0]	32	0	rw	
a3[0]	32	0	rw	
t0[0]	32	0	rw	
t1[0]	32	0	rw	
t2[0]	32	0	rw	
t3[0]	32	0	rw	
t4[0]	32	0	rw	
t5[0]	32	0	rw	
t6[0]	32	0	rw	
t7[0]	32	0	rw	
s0[0]	32	0	rw	
s1[0]	32	0	rw	
s2[0]	32	0	rw	
s3[0]	32	0	rw	
s4[0]	32	0	rw	
s5[0]	32	0	rw	
s6[0]	32	0	rw	
s7[0]	32	0	rw	
t8[0]	32	0	rw	
t9[0]	32	0	rw	
k0[0]	32	0	rw	
k1[0]	32	0	rw	
gp[0]	32	0	rw	
sp[0]	32	0	rw	stack pointer
s8[0]	32	0	rw	frame pointer
ra[0]	32	0	rw	
zero[1]	32	0	r-	constant zero
at[1]	32	0	rw	
v0[1]	32	0	rw	
v1[1]	32	0	rw	
a0[1]	32	0	rw	
a1[1]	32	0	rw	

a2[1]	32	0	rw	
a3[1]	32	0	rw	
t0[1]	32	0	rw	
t1[1]	32	0	rw	
t2[1]	32	0	rw	
t3[1]	32	0	rw	
t4[1]	32	0	rw	
t5[1]	32	0	rw	
t6[1]	32	0	rw	
t7[1]	32	0	rw	
s0[1]	32	0	rw	
s1[1]	32	0	rw	
s2[1]	32	0	rw	
s3[1]	32	0	rw	
s4[1]	32	0	rw	
s5[1]	32	0	rw	
s6[1]	32	0	rw	
s7[1]	32	0	rw	
t8[1]	32	0	rw	
t9[1]	32	0	rw	
k0[1]	32	0	rw	
k1[1]	32	0	rw	
gp[1]	32	0	rw	
sp[1]	32	0	rw	stack pointer
s8[1]	32	0	rw	frame pointer
ra[1]	32	0	rw	

12.1.4 COP0

Table 25. Registers at level 1, type: CPU, register group: 'COP0'

Name	Bits	Initial value (Hex)		Description
sr	32	400004	rw	CP0 register 12/0
bad	32	0	rw	CP0 register 8/0
cause	32	0	rw	CP0 register 13/0
index	32	0	rw	CP0 register 0/0
random	32	0	rw	CP0 register 1/0
entrylo0	32	0	rw	CP0 register 2/0
entrylo1	32	0	rw	CP0 register 3/0
context	32	0	rw	CP0 register 4/0
userlocal	32	0	rw	CP0 register 4/2
pagemask	32	0	rw	CP0 register 5/0
pagegrain	32	c8000000	rw	CP0 register 5/1

wired	32	0	rw	CP0 register 6/0
hwrena	32	0	rw	CP0 register 7/0
badvaddr	32	0	rw	CP0 register 8/0
badinstr	32	0	rw	CP0 register 8/1
badinstrp	32	0	rw	CP0 register 8/2
count	32	0	rw	CP0 register 9/0
entryhi	32	0	rw	CP0 register 10/0
guestctl1	32	0	rw	CP0 register 10/4
guestctl2	32	0	rw	CP0 register 10/5
guestctl3	32	0	rw	CP0 register 10/6
compare	32	0	rw	CP0 register 11/0
guestctl0ext	32	40	rw	CP0 register 11/4
status	32	400004	rw	CP0 register 12/0
intctl	32	e0030000	rw	CP0 register 12/1
srsctl	32	4000000	rw	CP0 register 12/2
srsmap	32	0	rw	CP0 register 12/3
viewipl	32	0	rw	CP0 register 12/4
srsmap2	32	0	rw	CP0 register 12/5
guestctl0	32	c4c00fc	rw	CP0 register 12/6
gtoffset	32	0	rw	CP0 register 12/7
viewripl	32	0	rw	CP0 register 13/4
nestedexc	32	0	rw	CP0 register 13/5
epc	32	0	rw	CP0 register 14/0
nestedepc	32	0	rw	CP0 register 14/2
prid	32	1ad20	rw	CP0 register 15/0
ebase	32	80000000	rw	CP0 register 15/1
cdmmbase	32	0	rw	CP0 register 15/2
config	32	81a08882	rw	CP0 register 16/0
config1	32	bea95482	rw	CP0 register 16/1
config2	32	80000000	rw	CP0 register 16/2
config3	32	8caabc28	rw	CP0 register 16/3
config4	32	c0fc0000	rw	CP0 register 16/4
config5	32	2011	rw	CP0 register 16/5
config6	32	0	rw	CP0 register 16/6
config7	32	80000000	rw	CP0 register 16/7
lladdr	32	0	rw	CP0 register 17/0
debug	32	2028000	rw	CP0 register 23/0
depc	32	0	rw	CP0 register 24/0
errctl	32	0	rw	CP0 register 26/0
itaglo	32	0	rw	CP0 register 28/0
idatalo	32	0	rw	CP0 register 28/1
dtaglo	32	0	rw	CP0 register 28/2

ddatalo	32	0	rw	CP0 register 28/3
l23taglo	32	0	rw	CP0 register 28/4
l23datalo	32	0	rw	CP0 register 28/5
itaghi	32	0	rw	CP0 register 29/0
idatahi	32	0	rw	CP0 register 29/1
dtaghi	32	0	rw	CP0 register 29/2
ddatahi	32	0	rw	CP0 register 29/3
l23taghi	32	0	rw	CP0 register 29/4
l23datahi	32	0	rw	CP0 register 29/5
errorepc	32	0	rw	CP0 register 30/0
desave	32	0	rw	CP0 register 31/0
kscratch1	32	0	rw	CP0 register 31/2
kscratch2	32	0	rw	CP0 register 31/3
kscratch3	32	0	rw	CP0 register 31/4
kscratch4	32	0	rw	CP0 register 31/5
kscratch5	32	0	rw	CP0 register 31/6
kscratch6	32	0	rw	CP0 register 31/7
guestindex	32	0	rw	CP0 guest register 0/0
guestrandom	32	0	rw	CP0 guest register 1/0
guestentrylo0	32	0	rw	CP0 guest register 2/0
guestentrylo1	32	0	rw	CP0 guest register 3/0
guestcontext	32	0	rw	CP0 guest register 4/0
guestuserlocal	32	0	rw	CP0 guest register 4/2
guestpagemask	32	0	rw	CP0 guest register 5/0
guestpagegrain	32	c8000000	rw	CP0 guest register 5/1
guestwired	32	0	rw	CP0 guest register 6/0
guesthwrena	32	0	rw	CP0 guest register 7/0
guestbadvaddr	32	0	rw	CP0 guest register 8/0
guestbadinstr	32	0	rw	CP0 guest register 8/1
guestbadinstrp	32	0	rw	CP0 guest register 8/2
guestcount	32	0	rw	CP0 guest register 9/0
guestentryhi	32	0	rw	CP0 guest register 10/0
guestguestctl1	32	0	rw	CP0 guest register 10/4
guestguestctl2	32	0	rw	CP0 guest register 10/5
guestguestctl3	32	0	rw	CP0 guest register 10/6
guestcompare	32	0	rw	CP0 guest register 11/0
guestguestctl0ext	32	0	rw	CP0 guest register 11/4
gueststatus	32	400004	rw	CP0 guest register 12/0
guestintctl	32	e0030000	rw	CP0 guest register 12/1
guestsrsctl	32	4000000	rw	CP0 guest register 12/2
guestsrsmap	32	0	rw	CP0 guest register 12/3
guestviewipl	32	0	rw	CP0 guest register 12/4

guestsrsmap2	32	0	rw	CP0 guest register 12/5
guestguestctl0	32	0	rw	CP0 guest register 12/6
guestgtoffset	32	0	rw	CP0 guest register 12/7
guestcause	32	0	rw	CP0 guest register 13/0
guestviewripl	32	0	rw	CP0 guest register 13/4
guestnestedexc	32	0	rw	CP0 guest register 13/5
guestepc	32	0	rw	CP0 guest register 14/0
guestnestedepc	32	0	rw	CP0 guest register 14/2
guestprid	32	0	rw	CP0 guest register 15/0
guestebase	32	80000000	rw	CP0 guest register 15/1
guestcdmmbase	32	0	rw	CP0 guest register 15/2
guestconfig	32	80048882	rw	CP0 guest register 16/0
guestconfig1	32	bea95482	rw	CP0 guest register 16/1
guestconfig2	32	80000000	rw	CP0 guest register 16/2
guestconfig3	32	8c2abc28	rw	CP0 guest register 16/3
guestconfig4	32	c0fc0000	rw	CP0 guest register 16/4
guestconfig5	32	11	rw	CP0 guest register 16/5
guestconfig6	32	0	rw	CP0 guest register 16/6
guestconfig7	32	0	rw	CP0 guest register 16/7
guestlladdr	32	0	rw	CP0 guest register 17/0
guestdebug	32	0	rw	CP0 guest register 23/0
guestdepc	32	0	rw	CP0 guest register 24/0
guesterrctl	32	0	rw	CP0 guest register 26/0
guestitaglo	32	0	rw	CP0 guest register 28/0
guestidatalo	32	0	rw	CP0 guest register 28/1
guestdtaglo	32	0	rw	CP0 guest register 28/2
guestddatalo	32	0	rw	CP0 guest register 28/3
guestl23taglo	32	0	rw	CP0 guest register 28/4
guestl23datalo	32	0	rw	CP0 guest register 28/5
guestitaghi	32	0	rw	CP0 guest register 29/0
guestidatahi	32	0	rw	CP0 guest register 29/1
guestdtaghi	32	0	rw	CP0 guest register 29/2
guestddatahi	32	0	rw	CP0 guest register 29/3
guestl23taghi	32	0	rw	CP0 guest register 29/4
guestl23datahi	32	0	rw	CP0 guest register 29/5
guesterrorepc	32	0	rw	CP0 guest register 30/0
guestdesave	32	0	rw	CP0 guest register 31/0
guestkscratch1	32	0	rw	CP0 guest register 31/2
guestkscratch2	32	0	rw	CP0 guest register 31/3
guestkscratch3	32	0	rw	CP0 guest register 31/4
guestkscratch4	32	0	rw	CP0 guest register 31/5
guestkscratch5	32	0	rw	CP0 guest register 31/6

guestkscratch6	32	0	rw	CP0 guest register 31/7

12.1.5 SPRAM

Table 26. Registers at level 1, type: CPU, register group: 'SPRAM'

Name	Bits	Initial value (Hex)		Description
ISPRAM_INDEX	8	0	rw	
ISPRAM_ENABLE	8	0	rw	
ISPRAM_SIZE	8	0	rw	
ISPRAM_BASE	32	0	rw	
ISPRAM_OFFSET	32	0	rw	
ISPRAM_FILE	32	-	-w	
ISPRAM_READ	32	-	-w	
ISPRAM_WRITE	32	-	-w	
DSPRAM_INDEX	8	0	rw	
DSPRAM_ENABLE	8	0	rw	
DSPRAM_SIZE	8	0	rw	
DSPRAM_BASE	32	0	rw	
DSPRAM_OFFSET	32	0	rw	
DSPRAM_READ	32	-	-w	
DSPRAM_WRITE	32	-	-w	

12.1.6 Integration_support

Table 27. Registers at level 1, type: CPU, register group: 'Integration_support'

Name	Bits	Initial value (Hex)		Description
stop	32	0	rw	write with non-zero to stop processor

#

