OVP Guide to Using Processor Models

Model specific information for MIPS_M6200

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

Author	Imperas Software Limited
Version	20200630.0
Filename	OVP_Model_Specific_Information_mips32_M6200.pdf
Created	2 July 2020
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.

Contents

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

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. MIPS32 used when coming out of reset

MMU Type: Fixed Mapping

Vectored interrupts implemented

MCU ASE implemented

DSP ASE Rev 2 implemented

Chapter 2

Configuration

2.1 Location

This model's VLNV is mips.ovpworld.org/processor/mips32/1.0.
The model source is usually at:
$IMPERAS_HOME/ImperasLib/source/mips.ovpworld.org/processor/mips32/1.0
The model binary is usually at:
$IMPERAS_HOME/lib/$IMPERAS_ARCH/ImperasLib/mips.ovpworld.org/processor/mips32/1.0

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/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.

Chapter 3

All Variants in this model

This model has these variants

Variant	Description
I7200	
M5100	
M5150	
M6200	(described in this document)
M6250	
MIPS32R6	
P5600	

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
softreset	input	optional	Core soft 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
hwint6	input	optional	External interrupt
hwint7	input	optional	External interrupt
nmi	input	optional	Non-maskable external interrupt
EICPresent	input	optional	Input signal SI_EICPresent per VPE
EIC_RIPL	input	optional	External interrupt controller RIPL (alias of hwint0 - 5 or 7)
EIC_EICSS	input	optional	External interrupt controller EICSS
EIC_VectorNum	input	optional	External interrupt controller vector number
EIC_VectorOffset	input	optional	External interrupt controller vector offset
intISS	output	optional	True when interrupt request is serviced
causeTI	output	optional	True when timer interrupt expires
causeIP0	output	optional	Raised for software interrupt request IP0
causeIP1	output	optional	Raised for software interrupt request IP1
si_sleep	output	optional	True when the VPE is in WAIT state
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)
configDS	Boolean	Override Config.DS for cacheless cores. Cosmetic only. Requires custom platform to actually implement the Dual SRAM interface.
MIPS_TRACE	Boolean	Enable MIPS-format trace output
gprNames	Boolean	Disassemble the register names from the default ABI instead of register numbers for 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
perfCounters	Uns32	Performance Counters
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. Ignored if less than two VPEs configured.

VPE1MaxTC	Uns32	Specifies the maximum TCs initially on VPE1. Ignored if less than three VPEs configured.
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
tcDisable	Uns32	Number of disabled TCs
vpeDisable	Uns32	Number of disabled VPEs
mvpconf0tc	Uns32	Override MVPConf0.PTC
mvpconf0pcp	Boolean	Override MVPConf0.PCP
mvpconf0tcp	Boolean	Override MVPConf0.TCP
mvpconf1c1f	Boolean	Override MVPConf.C1F
mvpcontrolPolicyMode	Boolean	Override MVPControl.POLICY_MODE
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 separated 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
FUSA	Boolean	Enable Functional Safety
CPC_FAULT_SUPPORTED	Uns32	Functional Safety Supported
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
usePreciseFpu	Uns32	Use the precise Floating Point emulation
simulateLite	Enumeration	Run Simulation with optimization. There are several optimizations which coule be combined (NONE, FS, MA or FSMA)
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
segcfg0PA	Uns32	Set CFG0.PA field of SegCtl0 register
segcfg1PA	Uns32	Set CFG1.PA field of SegCtl0 register
segcfg2PA	Uns32	Set CFG2.PA field of SegCtl1 register
segcfg3PA	Uns32	Set CFG3.PA field of SegCtl1 register
segcfg4PA	Uns32	Set CFG4.PA field of SegCtl2 register
segcfg5PA	Uns32	Set CFG5.PA field of SegCtl2 register
segcfg0AM	Uns32	Set CFG0.AM field of SegCtl0 register
segcfg1AM	Uns32	Set CFG1.AM field of SegCtl0 register
segcfg2AM	Uns32	Set CFG2.AM field of SegCtl1 register
segcfg3AM	Uns32	Set CFG3.AM field of SegCtl1 register
segcfg4AM	Uns32	Set CFG4.AM field of SegCtl2 register
segcfg5AM	Uns32	Set CFG5.AM field of SegCtl2 register
segcfg0EU	Uns32	Set CFG0.EU field of SegCtl0 register
segcfg1EU	Uns32	Set CFG1.EU field of SegCtl0 register
segcfg2EU	Uns32	Set CFG2.EU field of SegCtl1 register
segcfg3EU	Uns32	Set CFG3.EU field of SegCtl1 register
segcfg4EU	Uns32	Set CFG4.EU field of SegCtl2 register
segcfg5EU	Uns32	Set CFG5.EU field of SegCtl2 register
segcfg0C	Uns32	Set CFG0.C field of SegCtl0 register
segcfg1C	Uns32	Set CFG1.C field of SegCtl0 register
segcfg2C	Uns32	Set CFG2.C field of SegCtl1 register

segcfg3C	Uns32	Set CFG3.C field of SegCtl1 register
segcfg4C	Uns32	Set CFG4.C field of SegCtl2 register
segcfg5C	Uns32	Set CFG5.C field of SegCtl2 register
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)
config1MMUSizeM1_VPE1	Uns32	Override Config1.MMUSizeM1 for VPE1
config1MMUSizeM1_VPE2	Uns32	Override Config1.MMUSizeM1 for VPE2
config1MMUSizeM1_VPE3	Uns32	Override Config1.MMUSizeM1 for VPE3
config1WR	Boolean	Override Config1.WR (watchpoint registers present)
config1PC	Boolean	Override Config1.PC (Performance Counters present)
config1C2	Boolean	Override Config1.C2 (Coprocessor 2 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
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
config4KScrExist	Uns32	Override Config4.KScrExist
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)
config5VP	Boolean	Override Config5.VP
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)
config7WR	Boolean	Override Config7[31] bit (Alternative implementation of Watch registers)
config7FPR	Boolean	Override Config7.FPR (one-half FPU clock ratio)
config7USP	Uns32	Override Config7.USP (USPRAM enable)
config7BTLM	Boolean	Override Config7.BTLM bit
config7BusSlp	Boolean	Override Config7.BusSlp bit
config7IVAD	Boolean	Override Config7.IVAD bit
config7RPS	Boolean	Override Config7.RPS bit
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)
srsconf0SRS1	Uns32	Override the SRS1 field in SRSConf0 register
srsconf0SRS2	Uns32	Override the SRS2 field in SRSConf0 register
srsconf0SRS3	Uns32	Override the SRS3 field in SRSConf0 register
wiredLimit	Uns32	Override Limit field of the Wired register
wiredLimitBits	Uns32	Override width of Limit field of the Wired register
wiredWiredBits	Uns32	Override width of Wired field of the Wired register
cdmmBaseCI	Boolean	Override CDMMBase.CI
parityEnable	Uns32	Specify error detection support: 0 - none; 1 - parity; 2 - ECC
useMpTb	Boolean	Override Use of multi-processor test bench
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
EIC_OPTION	Uns32	Override the external interrupt controller EIC_OPTION
ICU_NUMINTERRUPTS	Uns32	ICU system only: override GIC_SH_CONFIG.NUMINTERRUPTS
icuBaseAddress	Uns32	ICU system only: Override icuBaseAddress. This should be configured to enable ICU.
icuPolarity	Uns32	ICU system only: Override icuPolarity
icuTrigger	Uns32	ICU system only: Override icuTrigger
icuDual	Uns32	ICU system only: Override icuDual
hasImpl17	Boolean	Enable read/write of Impl17 bit in Status register
hasImpl16	Boolean	Enable read/write of Impl16 bit in Status 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
USPRAM_SIZE	Uns32	Encoded size of the USPRAM region (log2(<USPRAM size in bytes>) - 11)
USPRAM_BASE	Uns64	Starting physical address of the USPRAM region
USPRAM_ENABLE	Boolean	Set the enable bit of the USPRAM region's tag (used to enable the USPRAM region prior to reset)
USPRAM_FILE	String	Load a MIPS hex file into the USPRAM region prior to reset
misalignedDataException	Enumeration	Select misaligned data access exception signaling: never, checkCCA or always (never, checkCCA or always)
commitTlbwErr	Boolean	Commit TLBWI/TLBRI on ECC; in MIPS_DV_MODE only

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
GE	27
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 12 commands.
This level in the model hierarchy has 5 register groups:

Group name	Registers
Core	65
DSP	9
Shadow	128
COP0	46
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 resource

-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	Int32	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	
-nocached	Boolean	
-nodcache	Boolean	

-noicache	Boolean	
-notrue	Boolean	
-nouncached	Boolean	
-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 mips model debug value

Argument	Type	Description
-value	Uns32	specify mips model debug flags

Table 12.7: mipsDebugFlags command arguments

12.1.11 mipsReadRegister

Read processor register using <resource><offset>

Argument	Type	Description
-offset	Uns32	the register offset
-resource	Uns32	the register resource

Table 12.8: mipsReadRegister command arguments

12.1.12 mipsWriteRegister

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

Argument	Type	Description
-offset	Uns32	the register offset
-resource	Uns32	the register resource
-value	Uns64	the value to write to register

Table 12.9: mipsWriteRegister 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	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
r0	32	0	r-	constant zero

r1	32	0	rw	
r2	32	0	rw	
r3	32	0	rw	
r4	32	0	rw	
r5	32	0	rw	
r6	32	0	rw	
r7	32	0	rw	
r8	32	0	rw	
r9	32	0	rw	
r10	32	0	rw	
r11	32	0	rw	
r12	32	0	rw	
r13	32	0	rw	
r14	32	0	rw	
r15	32	0	rw	
r16	32	0	rw	
r17	32	0	rw	
r18	32	0	rw	
r19	32	0	rw	
r20	32	0	rw	
r21	32	0	rw	
r22	32	0	rw	
r23	32	0	rw	
r24	32	0	rw	
r25	32	0	rw	
r26	32	0	rw	
r27	32	0	rw	
r28	32	0	rw	
r29	32	0	rw	stack pointer
r30	32	0	rw	frame pointer
r31	32	0	rw	

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

13.1.2 DSP

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

Name	Bits	Initial-Hex	RW	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

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

13.1.3 Shadow

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

Name	Bits	Initial-Hex	RW	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	
r0[0]	32	0	r-	constant zero
r1[0]	32	0	rw	
r2[0]	32	0	rw	
r3[0]	32	0	rw	
r4[0]	32	0	rw	
r5[0]	32	0	rw	
r6[0]	32	0	rw	
r7[0]	32	0	rw	
r8[0]	32	0	rw	
r9[0]	32	0	rw	
r10[0]	32	0	rw	
r11[0]	32	0	rw	
r12[0]	32	0	rw	
r13[0]	32	0	rw	
r14[0]	32	0	rw	
r15[0]	32	0	rw	
r16[0]	32	0	rw	
r17[0]	32	0	rw	
r18[0]	32	0	rw	
r19[0]	32	0	rw	
r20[0]	32	0	rw	
r21[0]	32	0	rw	
r22[0]	32	0	rw	
r23[0]	32	0	rw	
r24[0]	32	0	rw	
r25[0]	32	0	rw	
r26[0]	32	0	rw	
r27[0]	32	0	rw	
r28[0]	32	0	rw	
r29[0]	32	0	rw	stack pointer
r30[0]	32	0	rw	frame pointer
r31[0]	32	0	rw	
r0[1]	32	0	r-	constant zero
r1[1]	32	0	rw	
r2[1]	32	0	rw	
r3[1]	32	0	rw	
r4[1]	32	0	rw	
r5[1]	32	0	rw	
r6[1]	32	0	rw	
r7[1]	32	0	rw	
r8[1]	32	0	rw	
r9[1]	32	0	rw	
r10[1]	32	0	rw	
r11[1]	32	0	rw	
r12[1]	32	0	rw	
r13[1]	32	0	rw	
r14[1]	32	0	rw	

r15[1]	32	0	rw	
r16[1]	32	0	rw	
r17[1]	32	0	rw	
r18[1]	32	0	rw	
r19[1]	32	0	rw	
r20[1]	32	0	rw	
r21[1]	32	0	rw	
r22[1]	32	0	rw	
r23[1]	32	0	rw	
r24[1]	32	0	rw	
r25[1]	32	0	rw	
r26[1]	32	0	rw	
r27[1]	32	0	rw	
r28[1]	32	0	rw	
r29[1]	32	0	rw	stack pointer
r30[1]	32	0	rw	frame pointer
r31[1]	32	0	rw	

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

13.1.4 COP0

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

Name	Bits	Initial-Hex	RW	Description
sr	32	400004	rw	CP0 register 12/0 (status)
bad	32	0	rw	CP0 register 8/0 (badvaddr)
cause	32	0	rw	CP0 register 13/0 (cause)
userlocal	32	0	rw	CP0 register 4/2
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
compare	32	0	rw	CP0 register 11/0
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
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	1aa20	rw	CP0 register 15/0
ebase	32	80000000	rw	CP0 register 15/1
config	32	a4008982	rw	CP0 register 16/0
config1	32	80000002	rw	CP0 register 16/1
config2	32	80000000	rw	CP0 register 16/2
config3	32	8c2aac20	rw	CP0 register 16/3
config4	32	80fc0000	rw	CP0 register 16/4
config5	32	2011	rw	CP0 register 16/5
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
l23taglo	32	0	rw	CP0 register 28/4
l23datalo	32	0	rw	CP0 register 28/5
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

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