



OVP Guide to Using Processor Models

Model Specific Information for variant MIPS32_microAptivCF

Imperas Software Limited

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



Author	Imperas Software Limited
Version	0.4
Filename	OVP_Model_Specific_Information_mips32_microAptivCF.pdf
Created	25 August 2015

Copyright Notice

Copyright © 2015 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 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.

Table of Contents

1.0 Overview.....	4
1.1 Description.....	4
1.2 Licensing.....	4
1.3 Limitations.....	4
1.4 Verification.....	4
1.5 Features.....	4
2.0 Configuration.....	4
2.1 Location.....	4
2.2 GDB Path.....	4
2.3 Semi-Host Library.....	4
2.4 Processor Endian-ness.....	4
2.5 QuantumLeap Support.....	4
2.6 Processor ELF Code.....	4
3.0 Other Variants in this Model.....	5
4.0 Bus Ports.....	5
5.0 Net Ports.....	5
6.0 FIFO Ports.....	6
7.0 Parameters.....	6
8.0 Execution Modes.....	11
9.0 Exceptions.....	11
10.0 Hierarchy of the model.....	13
10.1 Level 1: CPU.....	13
11.0 Model Commands.....	14
11.1 Level 1: CPU.....	14
12.0 Registers.....	14
12.1 Level 1: CPU.....	14
12.1.1 Core.....	14
12.1.2 FPU.....	15
12.1.3 DSP.....	16
12.1.4 COP0.....	16
12.1.5 Integration_support.....	17

1.0 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

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: Fixed Mapping

FPU implemented

Vectored interrupts implemented

MCU ASE implemented

DSP ASE Rev 2 implemented

2.0 Configuration

2.1 Location

The model source and object file is found in the VLNV tree at:
mips.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/mips32SDE/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

3.0 Other Variants in this Model

Table 1.

Variant
ISA
M14K
M14KcTLB
M14KcFMM
4KEc
4KEm
4KEp
M4K
4Kc
4Km
4Kp
24Kc
24Kf
24KEc
24KEf
34Kc
34Kf
34Kn
74Kc
74Kf
1004Kc
1004Kf
1074Kc
1074Kf
microAptivC
microAptivP
microAptivCF
interAptiv
interAptivUP
proAptiv

4.0 Bus Ports

Table 2.

Type	Name	Bits
master (initiator)	INSTRUCTION	32
master (initiator)	DATA	32

5.0 Net Ports

Table 3.

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

6.0 FIFO Ports

No FIFO Ports in this model.

7.0 Parameters

Table 4.

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)
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
mpuRegions	Uns32	Number of regions for memory protection unit
mvpconf0vpe	Uns32	Override MVPConf0.PVPE
mvpconf0tc	Uns32	Override MVPConf0.PTC
mvpconf0pcp	Boolean	Override MVPConf0.PCP
mvpconf0tcp	Boolean	Override MVPConf0.TCP
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)
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)
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
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
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)

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
srsctlHSS	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
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
GIC_EX	Boolean	CMP system only: GIC unit present
CPC_EX	Boolean	CMP system only: CPC unit present
TIMER_ROUTABLE	Boolean	CMP system only: cpu timer interrupt routable within cluster
SWINT_ROUTABLE	Boolean	CMP system only: software interrupt routable within cluster
GCR_PCORES	Uns32	CMP system only: override GCR_CONFIG.PCORES (number of cores-1)
GCR_BASE	Uns32	CMP system only: override GCR_BASE.GCR_BASE (default GCR register address)
GCR_MINOR_REV	Uns32	CMP system only: override GCR_REV.MINOR_REV
GCR_MAJOR_REV	Uns32	CMP system only: override GCR_REV.MAJOR_REV

GCR_CACHE_MINOR_REV	Uns32	CMP system only: override GCR_CACHE_REV.MINOR_REV
GCR_CACHE_MAJOR_REV	Uns32	CMP system only: override GCR_CACHE_REV.MAJOR_REV
GCR_IOC1_MINOR_REV	Uns32	CMP system only: override GCR_IOC1_REV.MINOR_REV
GCR_IOC1_MAJOR_REV	Uns32	CMP system only: override GCR_IOC1_REV.MAJOR_REV
GIC_NUMINTERRUPTS	Uns32	CMP system only: override GIC_SH_CONFIG.NUMINTERRUPTS
GIC_COUNTBITS	Uns32	CMP system only: override GIC_SH_CONFIG.COUNTBITS
GIC_MINOR_REV	Uns32	CMP system only: override GIC_SH_REVISION.MINOR_REV
GIC_MAJOR_REV	Uns32	CMP system only: override GIC_SH_REVISION.MAJOR_REV
GIC_PVPES	Uns32	CMP system only: override GIC_SH_CONFIG.PVPE
CPC_MICROSTEP	Uns32	CMP system only: override CPC_SEQDEL.MICROSTEP
CPC_RAILDELAY	Uns32	CMP system only: override CPC_RAIL.RAILDELAY
CPC_RESETLEN	Uns32	CMP system only: override CPC_RESETLEN.RESETLEN
CPC_MINOR_REV	Uns32	CMP system only: override CPC_REVISION.MINOR_REV
CPC_MAJOR_REV	Uns32	CMP system only: override CPC_REVISION.MAJOR_REV
GCR_C0_RESET_BASE	Uns32	CMP system only: GCR_CL_RESET_BASE for core 0
GCR_C1_RESET_BASE	Uns32	CMP system only: GCR_CL_RESET_BASE for core 1
GCR_C2_RESET_BASE	Uns32	CMP system only: GCR_CL_RESET_BASE for core 2
GCR_C3_RESET_BASE	Uns32	CMP system only: GCR_CL_RESET_BASE for core 3
GCR_C0_RESET_EXT_BASE	Uns32	CMP system only: GCR_CL_RESET_EXT_BASE for core 0. Only used when SegCtl present
GCR_C1_RESET_EXT_BASE	Uns32	CMP system only: GCR_CL_RESET_EXT_BASE for core 1. Only used when SegCtl present
GCR_C2_RESET_EXT_BASE	Uns32	CMP system only: GCR_CL_RESET_EXT_BASE for core 2. Only used when SegCtl present

GCR_C3_RESET_EXT_BASE	Uns32	CMP system only: GCR_CL_RESET_EXT_BASE for core 3. Only used when SegCtl present
EIC_OPTION	Uns32	Override the external interrupt controller EIC_OPTION

8.0 Execution Modes

Table 5.

Name	Code
KERNEL	0
DEBUG	1
SUPERVISOR	2
USER	3

9.0 Exceptions

Table 6.

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
Prot	29
CacheErr	30

10.0 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 12 commands.

This level in the model hierarchy has 5 register groups:

Table 7.

Group name	Registers
Core	33
FPU	34
DSP	9
COP0	26
Integration_support	1

This level in the model hierarchy has no children.

11.0 Model Commands

11.1 Level 1: CPU

Table 8.

Name	Arguments
isync	specify instruction address range for synchronous execution
itrace	enable or disable instruction tracing
mipsCOP0	<register> <select>
mipsCacheDisable	
mipsCacheEnable	-tag -full
mipsCacheRatio	-icache -dcache
mipsCacheReport	
mipsCacheReset	
mipsCacheTrace	-on -off [-nocached -nouncached] [-noicache -nodcache] [-noartifact -notrue]
mipsDebugFlags	<value>
mipsReadRegister	<resource> <offset>
mipsWriteRegister	<resource> <offset> <value>

12.0 Registers

12.1 Level 1: CPU

12.1.1 Core

Table 9.

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 FPU

Table 10.

Name	Bits	Initial value (Hex)		Description
f0	32	0	rw	
f1	32	0	rw	
f2	32	0	rw	
f3	32	0	rw	
f4	32	0	rw	
f5	32	0	rw	
f6	32	0	rw	
f7	32	0	rw	
f8	32	0	rw	
f9	32	0	rw	
f10	32	0	rw	
f11	32	0	rw	
f12	32	0	rw	
f13	32	0	rw	
f14	32	0	rw	
f15	32	0	rw	
f16	32	0	rw	
f17	32	0	rw	

f18	32	0	rw	
f19	32	0	rw	
f20	32	0	rw	
f21	32	0	rw	
f22	32	0	rw	
f23	32	0	rw	
f24	32	0	rw	
f25	32	0	rw	
f26	32	0	rw	
f27	32	0	rw	
f28	32	0	rw	
f29	32	0	rw	
f30	32	0	rw	
f31	32	0	rw	
fsr	32	10c0000	rw	floating point status
fir	32	1f39d20	r-	floating point information

12.1.3 DSP

Table 11.

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

12.1.4 COP0

Table 12.

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
hwrena	32	0	rw	CP0 register 7/0
badvaddr	32	0	rw	CP0 register 8/0

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	0	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
epc	32	0	rw	CP0 register 14/0
prid	32	19d00	rw	CP0 register 15/0
ebase	32	80000000	rw	CP0 register 15/1
config	32	a4008582	rw	CP0 register 16/0
config1	32	80000003	rw	CP0 register 16/1
config2	32	80000000	rw	CP0 register 16/2
config3	32	228c20	rw	CP0 register 16/3
config7	32	80000000	rw	CP0 register 16/7
debug	32	2028000	rw	CP0 register 23/0
depc	32	0	rw	CP0 register 24/0
errorepc	32	0	rw	CP0 register 30/0
desave	32	0	rw	CP0 register 31/0

12.1.5 Integration_support

Table 13.

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

#