



OVP Guide to Using Processor Models

Model specific information for
TexasInstruments_generic

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

The Imperas logo consists of the word 'imperas' in a bold, lowercase, blue sans-serif font. The letter 'i' has a small orange square above it.

Author	Imperas Software Limited
Version	20220722.0
Filename	OVP_Model_Specific_Information_tms320c3x_generic.pdf
Created	27 July 2022
Status	OVP Standard Release

Copyright Notice

Copyright (c) 2022 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
2	Configuration	2
2.1	Location	2
2.2	GDB Path	2
2.3	Semi-Host Library	2
2.4	Processor Endian-ness	2
2.5	QuantumLeap Support	2
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
8.1	Parameter values	9
9	Execution Modes	10
10	Exceptions	11
11	Hierarchy of the model	12
11.1	Level 1	12
12	Model Commands	13
12.1	Level 1	13
12.1.1	debugflags	13
12.1.2	isync	13
12.1.3	itrace	13
13	Registers	15
13.1	Level 1	15
13.1.1	GPR	15

13.1.2 Status	16
-------------------------	----

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.

Chapter 2

Configuration

2.1 Location

This model's VLVN is ti.ovpworld.org/processor/tms320c3x/1.0.

The model source is usually at:

`$IMPERAS_HOME/ImperasLib/source/ti.ovpworld.org/processor/tms320c3x/1.0`

The model binary is usually at:

`$IMPERAS_HOME/lib/$IMPERAS_ARCH/ImperasLib/ti.ovpworld.org/processor/tms320c3x/1.0`

2.2 GDB Path

This model has no GDB available in this package.

2.3 Semi-Host Library

The default semi-host library file is ti.ovpworld.org/semitesting/tms320c3xNewlib/1.0

2.4 Processor Endian-ness

This is a LITTLE endian model.

2.5 QuantumLeap Support

This processor is qualified to run in a QuantumLeap enabled simulator.

2.6 Processor ELF code

The ELF code supported by this model is: 0x93.

Chapter 3

All Variants in this model

This model has these variants

Variant	Description
generic	Single default variant

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	32	32	mandatory	Used to fetch code for execution
DATA	32	32	optional	Used to read & write data

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
int0	input	optional	
tint0	input	optional	
tint1	input	optional	

Table 6.1: Net Ports

Chapter 7

FIFO Ports

This model has no FIFO ports.

Chapter 8

Formal Parameters

Name	Type	Description
verbose	Boolean	Turn on model messages
variant	Enumeration	Processor variant

Table 8.1: Parameters

8.1 Parameter values

These are the current parameter values.

Name	Value
(Others)	
verbose	F
variant	generic

Table 8.2: Parameter values

Chapter 9

Execution Modes

This model does not have different execution modes.

Chapter 10

Exceptions

Exception	Code	Description
RESERVE0	0	Reserved 0
INT0	1	External Interrupt 0
INT1	2	External Interrupt 1
INT2	3	External Interrupt 2
INT3	4	External Interrupt 3
XINT0	5	Serial Port 0 tx
RINT0	6	Serial Port 0 rx
RESERVE7	7	Reserved 7
RESERVE8	8	Reserved 8
TINT0	9	Timer 0
TINT1	10	Timer 1
DINT0	11	DMA0 channel
DINT1	12	DMA1 chaennl

Table 10.1: Exceptions implemented by this processor

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

This level in the model hierarchy has 3 commands.

This level in the model hierarchy has 2 register groups:

Group name	Registers
GPR	29
Status	0

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

12.1.1 debugflags

show or modify the processor debug flags

Argument	Type	Description
-get	Boolean	print current processor flags value
-mask	Boolean	print valid debug flag bits
-set	Int32	new processor flags (only flags 0x00000002 can be modified)

Table 12.1: debugflags command arguments

12.1.2 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 12.2: isync command arguments

12.1.3 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
-memory	String	show memory accesses by this instruction. Argument can be any combination of X (execute), A (load or store access) and S (system)
-mode	Boolean	show processor mode changes
-off	Boolean	disable instruction tracing
-on	Boolean	enable instruction tracing
-processorname	Boolean	Include processor name in all trace lines
-registerchange	Boolean	show registers changed by this instruction
-registers	Boolean	show registers after each trace

Table 12.3: itrace command arguments

Chapter 13

Registers

13.1 Level 1

13.1.1 GPR

Registers at level:1, group:GPR

Name	Bits	Initial-Hex	RW	Description
R0	64	0	rw	
R1	64	0	rw	
R2	64	0	rw	
R3	64	0	rw	
R4	64	0	rw	
R5	64	0	rw	
R6	64	0	rw	
R7	64	0	rw	
AR0	32	0	rw	
AR1	32	0	rw	
AR2	32	0	rw	
AR3	32	0	rw	
AR4	32	0	rw	
AR5	32	0	rw	
AR6	32	0	rw	
AR7	32	0	rw	
DP	32	0	rw	
IR0	32	0	rw	
IR1	32	0	rw	
BK	32	0	rw	
SP	32	0	rw	stack pointer
ST	32	0	rw	
IE	32	0	rw	
IF	32	0	rw	
IOF	32	0	rw	
RS	32	0	rw	
RE	32	0	rw	
RC	32	0	rw	
PC	32	0	rw	program counter

Table 13.1: Registers at level 1, group:GPR

13.1.2 Status

Registers at level:1, group:Status

Name	Bits	Initial-Hex	RW	Description
------	------	-------------	----	-------------

Table 13.2: Registers at level 1, group:Status