



## OVP Guide to Using Processor Models

### Model specific information for ARM\_Cortex-M7

Imperas Software Limited  
Imperas Buildings, North Weston  
Thame, Oxfordshire, OX9 2HA, U.K.  
[docs@imperas.com](mailto:docs@imperas.com)



|          |   |
|----------|---|
| Author   | Imperas Software Limited                          |
| Version  | 20240902.0  |
| Filename | OVP_Model_Specific_Information_armm_Cortex-M7.pdf |
| Created  | 2 September 2024                                  |
| Status   | OVP Standard Release                              |

## Copyright Notice

Copyright (c) 2024 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](http://OVPworld.org).

# Contents

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Overview</b>                                    | <b>1</b>  |
| 1.1      | Description . . . . .                              | 1         |
| 1.2      | Licensing . . . . .                                | 1         |
| 1.3      | Limitations . . . . .                              | 2         |
| 1.4      | Verification . . . . .                             | 2         |
| 1.5      | Features . . . . .                                 | 2         |
| 1.6      | Unpredictable Behavior . . . . .                   | 3         |
| 1.6.1    | Equal Target Registers . . . . .                   | 3         |
| 1.6.2    | Floating Point Load/Store Multiple Lists . . . . . | 3         |
| 1.6.3    | If-Then (IT) Block Constraints . . . . .           | 4         |
| 1.6.4    | Use of R13 . . . . .                               | 4         |
| 1.6.5    | Use of R15 . . . . .                               | 4         |
| 1.7      | Integration Support . . . . .                      | 4         |
| 1.7.1    | Memory Transaction Query . . . . .                 | 4         |
| 1.7.2    | Halt Reason Introspection . . . . .                | 4         |
| <b>2</b> | <b>Configuration</b>                               | <b>6</b>  |
| 2.1      | Location . . . . .                                 | 6         |
| 2.2      | GDB Path . . . . .                                 | 6         |
| 2.3      | Semi-Host Library . . . . .                        | 6         |
| 2.4      | Processor Endian-ness . . . . .                    | 6         |
| 2.5      | QuantumLeap Support . . . . .                      | 6         |
| 2.6      | Processor ELF code . . . . .                       | 6         |
| <b>3</b> | <b>All Variants in this model</b>                  | <b>7</b>  |
| <b>4</b> | <b>Bus Master Ports</b>                            | <b>8</b>  |
| <b>5</b> | <b>Bus Slave Ports</b>                             | <b>9</b>  |
| <b>6</b> | <b>Net Ports</b>                                   | <b>10</b> |
| <b>7</b> | <b>FIFO Ports</b>                                  | <b>12</b> |
| <b>8</b> | <b>Formal Parameters</b>                           | <b>13</b> |
| 8.1      | Parameter values and limits . . . . .              | 14        |
| <b>9</b> | <b>Execution Modes</b>                             | <b>17</b> |

|                                      |           |
|--------------------------------------|-----------|
| <b>10 Exceptions</b>                 | <b>18</b> |
| <b>11 Hierarchy of the model</b>     | <b>19</b> |
| 11.1 Level 1 . . . . .               | 19        |
| <b>12 Model Commands</b>             | <b>20</b> |
| 12.1 Level 1 . . . . .               | 20        |
| 12.1.1 debugflags . . . . .          | 20        |
| 12.1.2 isync . . . . .               | 20        |
| 12.1.3 itrace . . . . .              | 20        |
| <b>13 Registers</b>                  | <b>22</b> |
| 13.1 Level 1 . . . . .               | 22        |
| 13.1.1 Core . . . . .                | 22        |
| 13.1.2 Control . . . . .             | 22        |
| 13.1.3 System . . . . .              | 23        |
| 13.1.4 Integration_support . . . . . | 24        |

# 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

ARMM Processor Model

### 1.2 Licensing

Usage of binary model under license governing simulator usage.

Note that for models of ARM CPUs the license includes the following terms:

Licensee is granted a non-exclusive, worldwide, non-transferable, revocable licence to:

If no source is being provided to the Licensee: use and copy only (no modifications rights are granted) the model for the sole purpose of designing, developing, analyzing, debugging, testing, verifying, validating and optimizing software which: (a) (i) is for ARM based systems; and (ii) does not incorporate the ARM Models or any part thereof; and (b) such ARM Models may not be used

to emulate an ARM based system to run application software in a production or live environment.

If source code is being provided to the Licensee: use, copy and modify the model for the sole purpose of designing, developing, analyzing, debugging, testing, verifying, validating and optimizing software which: (a) (i) is for ARM based systems; and (ii) does not incorporate the ARM Models or any part thereof; and (b) such ARM Models may not be used to emulate an ARM based system to run application software in a production or live environment.

In the case of any Licensee who is either or both an academic or educational institution the purposes shall be limited to internal use.

Except to the extent that such activity is permitted by applicable law, Licensee shall not reverse engineer, decompile, or disassemble this model. If this model was provided to Licensee in Europe, Licensee shall not reverse engineer, decompile or disassemble the Model for the purposes of error correction.

The License agreement does not entitle Licensee to manufacture in silicon any product based on this model.

The License agreement does not entitle Licensee to use this model for evaluating the validity of any ARM patent.

The License agreement does not entitle Licensee to use the model to emulate an ARM based system to run application software in a production or live environment.

Source of model available under separate Imperas Software License Agreement.

## 1.3 Limitations

Performance Monitors are not implemented.

Debug Extension and related blocks are not implemented.

## 1.4 Verification

Models have been extensively tested by Imperas. ARM Cortex-M models have been successfully used by customers to simulate the Micrium uC/OS-II kernel and FreeRTOS.

## 1.5 Features

The model is configured with 16 interrupts and 3 priority bits (use `override_numInterrupts` and `override_priorityBits` parameters to change these).

Thumb-2 instructions are supported.

MPU is present. Use parameter `override_MPU_TYPE` to disable it or change the number of MPU regions if required.

SysTick timer is present. Use parameter `SysTickPresent` to disable it if required.

FPU extension is not present. Use parameter `override_MVFR0` to enable it if required.

DSP extension is present. Use parameter `override_InstructionAttributes3` to disable it if required.

Bit-band region is not present. Use parameter `BitBandPresent` to enable it if required.

TCMs are present (ITCM is 16384 bytes, DTCM is 16384 bytes). Use parameters `override_ITCMSize` and `override_DTCMSize` to set TCM sizes if required (note that these parameters specify size codes in `CFGITCMSIZE/CFGDTCMSIZE` format, not byte sizes).

When TCMs are present, bus ports called ITCM and DTCM are created so that TCM contents may be viewed or modified externally by connecting to these ports. Parameter `useInternalTCMs` specifies whether TCM memory is modeled internally or externally. If modeled externally, the TCMs must be implemented on a bus which is then connected to the TCM bus ports listed above.

A 20-bit PPB bus port can be connected. If it is, then any loads or stores to address range `0xE0040000:0xE00FFFFF` will be visible on this bus at address range `0x00040000:0x000FFFFF`.

An AHB peripheral interface port is present. If this port is unconnected, all AHB region accesses are routed via the data bus. If the port is connected, then all data accesses to the System region `0xE0100000:0xFFFFFFFF` will be routed to that port. Additionally data accesses to addresses in the Peripheral region starting at address `0x40000000` will be routed to that port if enabled by `CM7_AHBPCR`.

An AHB region of size 536870912 bytes is present. Use parameter `override_AHBPSize` set a different size if required, or to disable the region if required using size zero. Note that this parameter specifies size codes in `CM7_AHBPCR` format, not byte size.

If the AHB region is present, use parameter `override_AHBPEnable` or configuration `net initahbp` to specify whether the AHB region should be enabled or disabled at reset.

## 1.6 Unpredictable Behavior

Many instruction behaviors are described in the ARM ARM as **CONSTRAINED UNPREDICTABLE**. This section describes how such situations are handled by this model.

### 1.6.1 Equal Target Registers

Some instructions allow the specification of two target registers (for example, double-width `SMULL`, or some `VMOV` variants), and such instructions are **CONSTRAINED UNPREDICTABLE** if the same target register is specified in both positions. In this model, such instructions are treated as **UNDEFINED**.

### 1.6.2 Floating Point Load/Store Multiple Lists

Instructions that load or store a list of floating point registers (e.g. `VSTM`, `VLDM`, `VPUSH`, `VPOP`) are **CONSTRAINED UNPREDICTABLE** if either the uppermost register in the specified range is greater than 32 or (for 64-bit registers) if more than 16 registers are specified. In this model, such instructions are treated as **UNDEFINED**.

### 1.6.3 If-Then (IT) Block Constraints

Where the behavior of an instruction in an if-then (IT) block is described as CONSTRAINED UNPREDICTABLE, this model treats that instruction as UNDEFINED.

### 1.6.4 Use of R13

Use of R13 is described as CONSTRAINED UNPREDICTABLE in many circumstances. This model allows R13 to be used like any other GPR.

### 1.6.5 Use of R15

Use of R15 is described as CONSTRAINED UNPREDICTABLE in many circumstances. This model allows such use to be configured using the parameter “unpredictableR15” as follows:

Value “undefined”: any reference to R15 in such a situation is treated as UNDEFINED;

Value “nop”: any reference to R15 in such a situation causes the instruction to be treated as a NOP;

Value “raz\_wi”: any reference to R15 in such a situation causes the instruction to be treated as a RAZ/WI (that is, R15 is read as zero and write-ignored);

Value “execute”: any reference to R15 in such a situation is executed using the current value of R15 on read, and writes to R15 are allowed.

Value “assert”: any reference to R15 in such a situation causes the simulation to halt with an assertion message (allowing any such unpredictable uses to be easily identified).

In this variant, the default value of “unpredictableR15” is “execute”.

## 1.7 Integration Support

This model implements a number of non-architectural pseudo-registers and other features to facilitate integration.

### 1.7.1 Memory Transaction Query

Two registers are intended for use within memory callback functions to provide additional information about the current memory access. Register `executionPri` indicates the current processor execution priority. Register `atomicType` indicates whether the current instruction has any special atomic constraints (0 indicates no constraint, 1 indicates atomic, 2 indicates exclusive access).

### 1.7.2 Halt Reason Introspection

An artifact register `HaltReason` can be read to determine the reason or reasons that a processor is halted. This register is a bitfield, with the following encoding: bit 0 indicates the processor has



executed a wait-for-event (WFE) instruction; bit 1 indicates the processor has executed a wait-for-interrupt (WFI) instruction; bit 2 indicates the processor has entered lockup state; and bit 3 indicates the processor is held in reset.

# Chapter 2

## Configuration

### 2.1 Location

This model's VLVN is [arm.ovpworld.org/processor/armmm/1.0](http://arm.ovpworld.org/processor/armmm/1.0).

The model source is usually at:

`$IMPERAS_HOME/ImperasLib/source/arm.ovpworld.org/processor/armmm/1.0`

The model binary is usually at:

`$IMPERAS_HOME/lib/$IMPERAS_ARCH/ImperasLib/arm.ovpworld.org/processor/armmm/1.0`

### 2.2 GDB Path

The default GDB for this model is: `$IMPERAS_HOME/lib/$IMPERAS_ARCH/gdb/arm-none-eabi-gdb`.

### 2.3 Semi-Host Library

The default semi-host library file is [arm.ovpworld.org/semihosting/armNewlib/1.0](http://arm.ovpworld.org/semihosting/armNewlib/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: 0x28.

## Chapter 3

# All Variants in this model

This model has these variants

| Variant       | Description                  |
|---------------|------------------------------|
| ARMv6-M       |                              |
| ARMv7-M       |                              |
| Cortex-M0     |                              |
| Cortex-M0plus |                              |
| Cortex-M1     |                              |
| Cortex-M3     |                              |
| Cortex-M4     |                              |
| Cortex-M4F    |                              |
| Cortex-M7     | (described in this document) |
| Cortex-M7F    |                              |
| Cortex-M23    |                              |
| Cortex-M33    |                              |
| Cortex-M33F   |                              |

Table 3.1: All Variants in this model

## Chapter 4

# Bus Master Ports

This model has these bus master ports.

| <b>Name</b> | min | max | Connect?  | Description               |
|-------------|-----|-----|-----------|---------------------------|
| INSTRUCTION | 32  | 32  | mandatory |                           |
| DATA        | 32  | 32  | optional  |                           |
| PPB         | 20  | 20  | optional  | PPB interface             |
| AHB         | 32  | 32  | optional  | AHB interface             |
| ITCM        | 14  | 14  | optional  | instruction TCM interface |
| DTCM        | 14  | 14  | optional  | data TCM interface        |

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                                     |
|-------------|--------|----------|---|
| sysResetReq | output | optional | Indicates that a reset is required              |
| intISS      | output | optional | Indicates that an interrupt service has started |
| eventOut    | output | optional | EVENTO output signal (SEV)                      |
| lockup      | output | optional | Indicates fatal lockup state entered            |
| haltReason  | output | optional | Indicates why core is halted                    |
| int         | input  | optional | Raise or lower interrupt by index               |
| reset       | input  | optional | RESET (active high)                             |
| cpuWait     | input  | optional | CPUWAIT (active high at reset only)             |
| nmi         | input  | optional | NMI (posedge-triggered)                         |
| eventIn     | input  | optional | EVENTI input (posedge-triggered)                |
| initvtor    | input  | optional | VTOR reset configuration                        |
| inititcm    | input  | optional | Enable ITCM at reset (if high)                  |
| initdtcm    | input  | optional | Enable DTCM at reset (if high)                  |
| initahbp    | input  | optional | Enable AHBP at reset (if high)                  |
| int0        | input  | optional | Scalar interrupt input 0                        |
| int1        | input  | optional | Scalar interrupt input 1                        |
| int2        | input  | optional | Scalar interrupt input 2                        |
| int3        | input  | optional | Scalar interrupt input 3                        |
| int4        | input  | optional | Scalar interrupt input 4                        |
| int5        | input  | optional | Scalar interrupt input 5                        |
| int6        | input  | optional | Scalar interrupt input 6                        |
| int7        | input  | optional | Scalar interrupt input 7                        |
| int8        | input  | optional | Scalar interrupt input 8                        |
| int9        | input  | optional | Scalar interrupt input 9                        |
| int10       | input  | optional | Scalar interrupt input 10                       |
| int11       | input  | optional | Scalar interrupt input 11                       |
| int12       | input  | optional | Scalar interrupt input 12                       |
| int13       | input  | optional | Scalar interrupt input 13                       |
| int14       | input  | optional | Scalar interrupt input 14                       |
| int15       | input  | optional | Scalar interrupt input 15                       |

---

Table 6.1: Net Ports

## Chapter 7

# FIFO Ports

This model has no FIFO ports.



## Chapter 8

# Formal Parameters

| Name               | Type        | Description  |
|--------------------|-------------|--|
| verbose            | Boolean     | Specify verbosity of output  |
| showHiddenRegs     | Boolean     | Show hidden registers during register tracing  |
| UAL                | Boolean     | Disassemble using UAL syntax   |
| enableVFPAtReset   | Boolean     | Enable vector floating point (SIMD and VFP) instructions at reset. (Enables cp10/11 in CPACR and sets FPEXC.EN)  |
| compatibility      | Enumeration | Specify compatibility mode   |
|                    | ISA         |  |
|                    | gdb         |  |
|                    | nopBKPT     |  |
| unpredictableR15   | Enumeration | Specify behavior for UNPREDICTABLE uses of R15   |
|                    | undefined   |  |
|                    | nop         |  |
|                    | raz_wi      |  |
|                    | execute     |  |
|                    | assert      |  |
| override_debugMask | Uns32       | Specifies debug mask, enabling debug output for model components   |
| instructionEndian  | Endian      | The architecture specifies that instruction fetch is always little endian; this attribute allows the defined instruction endianness to be overridden if required |
| resetAtTime0       | Boolean     | Reset the model at time=0 (default=1)  |
| SysTickPresent     | Uns32       | Specify number of SysTick timers present   |
| BitBandPresent     | Boolean     | Specify presence of bit-band region  |
| useInternalTCMs    | Boolean     | Specify that configured TCMs should be modeled internally  |
| override_PFR0      | Uns32       | Override ID_PFR0 register  |
| override_PFR1      | Uns32       | Override ID_PFR1 register  |
| override_DFR0      | Uns32       | Override ID_DFR0 register  |
| override_AFR0      | Uns32       | Override ID_AFR0 register  |
| override_MMFR0     | Uns32       | Override ID_MMFR0 register   |
| override_MMFR1     | Uns32       | Override ID_MMFR1 register   |
| override_MMFR2     | Uns32       | Override ID_MMFR2 register   |
| override_MMFR3     | Uns32       | Override ID_MMFR3 register   |
| override_ISAR0     | Uns32       | Override ID_ISAR0 register   |
| override_ISAR1     | Uns32       | Override ID_ISAR1 register   |
| override_ISAR2     | Uns32       | Override ID_ISAR2 register   |
| override_ISAR3     | Uns32       | Override ID_ISAR3 register   |
| override_ISAR4     | Uns32       | Override ID_ISAR4 register   |
| override_ISAR5     | Uns32       | Override ID_ISAR5 register   |
| override_MVFR0     | Uns32       | Override ID_MVFR0 register   |
| override_MVFR1     | Uns32       | Override ID_MVFR1 register   |

|                                 |         |   |
|---------------------------------|---------|---|
| override_MVFR2                  | Uns32   | Override ID_MVFR2 register  |
| override_ACTLR                  | Uns32   | Override system ACTLR register  |
| override_CPUID                  | Uns32   | Override system CPUID register  |
| override_MPU_TYPE               | Uns32   | Override system MPU_TYPE register   |
| override_VTOR                   | Uns32   | Override VTOR register reset value  |
| override_CCSIDR_1I              | Uns32   | Override CCSIDR (level 1 instruction)   |
| override_CCSIDR_1D              | Uns32   | Override CCSIDR (level 1 data)  |
| override_CCSIDR_2I              | Uns32   | Override CCSIDR (level 2 instruction)   |
| override_CCSIDR_2D              | Uns32   | Override CCSIDR (level 2 data)  |
| override_CCSIDR_3I              | Uns32   | Override CCSIDR (level 3 instruction)   |
| override_CCSIDR_3D              | Uns32   | Override CCSIDR (level 3 data)  |
| override_CCSIDR_4I              | Uns32   | Override CCSIDR (level 4 instruction)   |
| override_CCSIDR_4D              | Uns32   | Override CCSIDR (level 4 data)  |
| override_CCSIDR_5I              | Uns32   | Override CCSIDR (level 5 instruction)   |
| override_CCSIDR_5D              | Uns32   | Override CCSIDR (level 5 data)  |
| override_CCSIDR_6I              | Uns32   | Override CCSIDR (level 6 instruction)   |
| override_CCSIDR_6D              | Uns32   | Override CCSIDR (level 6 data)  |
| override_CCSIDR_7I              | Uns32   | Override CCSIDR (level 7 instruction)   |
| override_CCSIDR_7D              | Uns32   | Override CCSIDR (level 7 data)  |
| override_deviceStrongAligned    | Boolean | Force accesses to Device and Strongly Ordered regions to be aligned   |
| override_STROffsetPC12          | Uns32   | Specifies that STR/STR of PC should do so with 12:byte offset from the current instruction (if 1), otherwise an 8:byte offset is used |
| override_ERG                    | Uns32   | Specifies exclusive reservation granule   |
| override_priorityBits           | Uns32   | Specifies number of priority bits in BASEPRI etc (1-8, default is 3)  |
| override_numInterrupts          | Uns32   | Specifies number of external interrupt lines  |
| override_ITCMSIZE               | Uns32   | Specifies ITCM size from 0KB to 16MB (CFGITCMSIZE value format)   |
| override_DTCMSIZE               | Uns32   | Specifies DTCM size from 0KB to 16MB (CFGDTCMSIZE value format)   |
| override_AHBPSIZE               | Uns32   | Specifies AHBPS size from 0MB to 512MB (CM7_AHBPCR value format)  |
| override_ITCMEnable             | Boolean | Specifies initial ITCM enabled state  |
| override_DTCMEnable             | Boolean | Specifies initial DTCM enabled state  |
| override_AHBPEEnable            | Boolean | Specifies initial AHBPE enabled state   |
| override_InstructionAttributes0 | Uns32   | Override ID_ISAR0 register (deprecated, use override_ISAR0)   |
| override_InstructionAttributes1 | Uns32   | Override ID_ISAR1 register (deprecated, use override_ISAR1)   |
| override_InstructionAttributes2 | Uns32   | Override ID_ISAR2 register (deprecated, use override_ISAR2)   |
| override_InstructionAttributes3 | Uns32   | Override ID_ISAR3 register (deprecated, use override_ISAR3)   |
| override_InstructionAttributes4 | Uns32   | Override ID_ISAR4 register (deprecated, use override_ISAR4)   |
| override_InstructionAttributes5 | Uns32   | Override ID_ISAR5 register (deprecated, use override_ISAR5)   |

Table 8.1: Parameters

## 8.1 Parameter values and limits

These are the formal parameter limits and actual parameter values

| Name     | Min | Max | Default | Actual    |
|----------|-----|-----|---------|-----------|
| (Others) |     |     |         |           |
| variant  |     |     |         | Cortex-M7 |
| verbose  |     |     | t       | t         |

|                    |   |            |            |            |
|--------------------|---|------------|------------|------------|
| showHiddenRegs     |   |            | f          | f          |
| UAL                |   |            | t          | t          |
| enableVFPAtReset   |   |            | f          | f          |
| compatibility      |   |            | ISA        | ISA        |
| unpredictableR15   |   |            | execute    | execute    |
| override_debugMask | 0 | 0xffffffff | 0          | 0          |
| endian             |   |            | none       | none       |
| instructionEndian  |   |            | none       | none       |
| resetAtTime0       |   |            | t          | t          |
| SysTickPresent     | 0 | 0          | 1          | 1          |
| BitBandPresent     |   |            | f          | f          |
| useInternalTCMs    |   |            | f          | f          |
| override_PFR0      | 0 | 0xffffffff | 48         | 48         |
| override_PFR1      | 0 | 0xffffffff | 0x200      | 0x200      |
| override_DFR0      | 0 | 0xffffffff | 0          | 0          |
| override_AFR0      | 0 | 0xffffffff | 0          | 0          |
| override_MMFR0     | 0 | 0xffffffff | 0x100030   | 0x100030   |
| override_MMFR1     | 0 | 0xffffffff | 0          | 0          |
| override_MMFR2     | 0 | 0xffffffff | 0x1000000  | 0x1000000  |
| override_MMFR3     | 0 | 0xffffffff | 0          | 0          |
| override_ISAR0     | 0 | 0xffffffff | 0x1101110  | 0x1101110  |
| override_ISAR1     | 0 | 0xffffffff | 0x2112000  | 0x2112000  |
| override_ISAR2     | 0 | 0xffffffff | 0x20232231 | 0x20232231 |
| override_ISAR3     | 0 | 0xffffffff | 0x1111131  | 0x1111131  |
| override_ISAR4     | 0 | 0xffffffff | 0x1310132  | 0x1310132  |
| override_ISAR5     | 0 | 0xffffffff | 0          | 0          |
| override_MVFR0     | 0 | 0xffffffff | 0          | 0          |
| override_MVFR1     | 0 | 0xffffffff | 0          | 0          |
| override_MVFR2     | 0 | 0xffffffff | 0          | 0          |
| override_ACTLR     | 0 | 0xffffffff | 0          | 0          |
| override_CPUID     | 0 | 0xffffffff | 0x410fc272 | 0x410fc272 |
| override_MPU_TYPE  | 0 | 0xffffffff | 0x800      | 0x800      |
| override_VTOR      | 0 | 0xffffffff | 0          | 0          |
| override_CCSIDR_1I | 0 | 0xffffffff | 0xf03fe019 | 0xf03fe019 |
| override_CCSIDR_1D | 0 | 0xffffffff | 0xf03fe019 | 0xf03fe019 |
| override_CCSIDR_2I | 0 | 0xffffffff | 0          | 0          |
| override_CCSIDR_2D | 0 | 0xffffffff | 0          | 0          |
| override_CCSIDR_3I | 0 | 0xffffffff | 0          | 0          |
| override_CCSIDR_3D | 0 | 0xffffffff | 0          | 0          |
| override_CCSIDR_4I | 0 | 0xffffffff | 0          | 0          |
| override_CCSIDR_4D | 0 | 0xffffffff | 0          | 0          |
| override_CCSIDR_5I | 0 | 0xffffffff | 0          | 0          |
| override_CCSIDR_5D | 0 | 0xffffffff | 0          | 0          |
| override_CCSIDR_6I | 0 | 0xffffffff | 0          | 0          |
| override_CCSIDR_6D | 0 | 0xffffffff | 0          | 0          |

|                                 |   |            |            |            |
|---------------------------------|---|------------|------------|------------|
| override_CCSIDR_7I              | 0 | 0xffffffff | 0          | 0          |
| override_CCSIDR_7D              | 0 | 0xffffffff | 0          | 0          |
| override_deviceStrongAligned    |   |            | f          | f          |
| override_STRoffsetPC12          | 0 | 1          | 0          | 0          |
| override_ERG                    | 0 | 0x400      | 0          | 0          |
| override_priorityBits           | 1 | 8          | 3          | 3          |
| override_numInterrupts          | 0 | 0x1f0      | 16         | 16         |
| override_ITCMSize               | 0 | 15         | 5          | 5          |
| override_DTCMSize               | 0 | 15         | 5          | 5          |
| override_AHBPSize               | 0 | 4          | 4          | 4          |
| override_ITCMEnable             |   |            | f          | f          |
| override_DTCMEnable             |   |            | f          | f          |
| override_AHBPEnable             |   |            | f          | f          |
| override_InstructionAttributes0 | 0 | 0xffffffff | 0x1101110  | 0x1101110  |
| override_InstructionAttributes1 | 0 | 0xffffffff | 0x2112000  | 0x2112000  |
| override_InstructionAttributes2 | 0 | 0xffffffff | 0x20232231 | 0x20232231 |
| override_InstructionAttributes3 | 0 | 0xffffffff | 0x1111131  | 0x1111131  |
| override_InstructionAttributes4 | 0 | 0xffffffff | 0x1310132  | 0x1310132  |
| override_InstructionAttributes5 | 0 | 0xffffffff | 0          | 0          |

Table 8.2: Parameter values and limits

## Chapter 9

# Execution Modes

| Mode    | Code |
|---------|------|
| Thread  | 0    |
| Handler | 1    |

Table 9.1: Modes implemented in this processor

## Chapter 10

# Exceptions

| Exception      | Code |
|----------------|------|
| None           | 0    |
| Reset          | 1    |
| NMI            | 2    |
| HardFault      | 3    |
| MemManage      | 4    |
| BusFault       | 5    |
| UsageFault     | 6    |
| SVCall         | 11   |
| DebugMonitor   | 12   |
| PendSV         | 14   |
| SysTick        | 15   |
| ExternalInt000 | 16   |
| ExternalInt001 | 17   |
| ExternalInt002 | 18   |
| ExternalInt003 | 19   |
| ExternalInt004 | 20   |
| ExternalInt005 | 21   |
| ExternalInt006 | 22   |
| ExternalInt007 | 23   |
| ExternalInt008 | 24   |
| ExternalInt009 | 25   |
| ExternalInt00a | 26   |
| ExternalInt00b | 27   |
| ExternalInt00c | 28   |
| ExternalInt00d | 29   |
| ExternalInt00e | 30   |
| ExternalInt00f | 31   |

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 4 register groups:

| Group name          | Registers |
|---------------------|-----------|
| Core                | 16        |
| Control             | 7         |
| System              | 83        |
| Integration_support | 4         |

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 0x0000008c 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   |
|----------|--------|---|
| -access  | String | show memory accesses by this instruction. Argument can be any combination of X (execute), A (load or store access) and S (system) |
| -after   | Uns64  | apply after this many instructions  |



|                   |         |   |
|-------------------|---------|---|
| -enable           | Boolean | enable instruction tracing  |
| -full             | Boolean | turn on all trace features  |
| -instructioncount | Boolean | include the instruction number in each trace  |
| -memory           | String  | (Alias for access). 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 Core

Registers at level:1, group:Core

| Name | Bits | Initial-Hex | RW | Description     |
|------|------|-------------|----|-----------------|
| r0   | 32   | 0           | rw |                 |
| 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 | frame pointer   |
| r12  | 32   | 0           | rw |                 |
| sp   | 32   | 0           | rw | stack pointer   |
| lr   | 32   | 0           | rw |                 |
| pc   | 32   | 0           | rw | program counter |

Table 13.1: Registers at level 1, group:Core

#### 13.1.2 Control

Registers at level:1, group:Control

| Name       | Bits | Initial-Hex | RW | Description                                 |
|------------|------|-------------|----|---|
| cpsr       | 32   | 0           | rw | xPSR register. Includes APSR, IPSR and EPSR |
| control    | 32   | 0           | rw |   |
| primask    | 32   | 0           | rw |   |
| faultmask  | 32   | 0           | rw |   |
| basepri    | 32   | 0           | rw |   |
| sp_process | 32   | 0           | rw | stack pointer                               |
| sp_main    | 32   | 0           | rw | stack pointer                               |

Table 13.2: Registers at level 1, group:Control

### 13.1.3 System

Registers at level:1, group:System

| Name        | Bits | Initial-Hex | RW | Description   |
|-------------|------|-------------|----|---|
| ICTR        | 32   | 0           | rw | 0xe000e004: Interrupt Controller Type               |
| ACTLR       | 32   | 0           | rw | 0xe000e008: Auxiliary Control                       |
| SYST_CSR    | 32   | 4           | rw | 0xe000e010: SysTick Control and Status              |
| SYST_RVR    | 32   | 0           | rw | 0xe000e014: SysTick Reload Value                    |
| SYST_CVR    | 32   | 0           | rw | 0xe000e018: SysTick Current Value                   |
| SYST_CALIB  | 32   | 0           | rw | 0xe000e01c: SysTick Calibration Value               |
| NVIC_ISER0  | 32   | 0           | rw | 0xe000e100: Interrupt Set Enable 0                  |
| NVIC_ICER0  | 32   | 0           | rw | 0xe000e180: Interrupt Clear Enable 0                |
| NVIC_ISPR0  | 32   | 0           | rw | 0xe000e200: Interrupt Set Pending 0                 |
| NVIC_ICPR0  | 32   | 0           | rw | 0xe000e280: Interrupt Clear Pending 0               |
| NVIC_IABR0  | 32   | 0           | r- | 0xe000e300: Interrupt Active Bit 0                  |
| NVIC_IPR0   | 32   | 0           | rw | 0xe000e400: Interrupt Priority 0                    |
| NVIC_IPR1   | 32   | 0           | rw | 0xe000e404: Interrupt Priority 1                    |
| NVIC_IPR2   | 32   | 0           | rw | 0xe000e408: Interrupt Priority 2                    |
| NVIC_IPR3   | 32   | 0           | rw | 0xe000e40c: Interrupt Priority 3                    |
| CPUID       | 32   | 410fc272    | r- | 0xe000ed00: CPUID Base                              |
| ICSR        | 32   | 1000        | rw | 0xe000ed04: Interrupt Control and State             |
| VTOR        | 32   | 0           | rw | 0xe000ed08: Vector Table Offset                     |
| AIRCR       | 32   | fa050000    | rw | 0xe000ed0c: Application Interrupt and Reset Control |
| SCR         | 32   | 0           | rw | 0xe000ed10: System Control                          |
| CCR         | 32   | 40200       | rw | 0xe000ed14: Configuration and Control               |
| SHPR1       | 32   | 0           | rw | 0xe000ed18: System Handler Priority 1               |
| SHPR2       | 32   | 0           | rw | 0xe000ed1c: System Handler Priority 2               |
| SHPR3       | 32   | 0           | rw | 0xe000ed20: System Handler Priority 3               |
| SHCSR       | 32   | 0           | rw | 0xe000ed24: System Handler Control and State        |
| CFSR        | 32   | 0           | rw | 0xe000ed28: Configurable Fault Status               |
| HFSR        | 32   | 0           | rw | 0xe000ed2c: HardFault Status                        |
| DFSR        | 32   | 0           | rw | 0xe000ed30: Debug Fault Status Register             |
| MMAR        | 32   | 0           | rw | 0xe000ed34: MemManage Fault Address                 |
| BFAR        | 32   | 0           | rw | 0xe000ed38: BusFault Address                        |
| AFSR        | 32   | 0           | rw | 0xe000ed3c: Auxiliary Fault Status                  |
| ID_PFR0     | 32   | 30          | rw | 0xe000ed40: Processor Feature 0                     |
| ID_PFR1     | 32   | 200         | rw | 0xe000ed44: Processor Feature 1                     |
| ID_DFR0     | 32   | 0           | rw | 0xe000ed48: Debug Feature 0                         |
| ID_AFR0     | 32   | 0           | rw | 0xe000ed4c: Auxiliary Feature 0                     |
| ID_MMFR0    | 32   | 100030      | rw | 0xe000ed50: Memory Model Feature 0                  |
| ID_MMFR1    | 32   | 0           | rw | 0xe000ed54: Memory Model Feature 1                  |
| ID_MMFR2    | 32   | 1000000     | rw | 0xe000ed58: Memory Model Feature 2                  |
| ID_MMFR3    | 32   | 0           | rw | 0xe000ed5c: Memory Model Feature 3                  |
| ID_ISAR0    | 32   | 1101110     | rw | 0xe000ed60: Instruction Set Attributes 0            |
| ID_ISAR1    | 32   | 2112000     | rw | 0xe000ed64: Instruction Set Attributes 1            |
| ID_ISAR2    | 32   | 20232231    | rw | 0xe000ed68: Instruction Set Attributes 2            |
| ID_ISAR3    | 32   | 1111131     | rw | 0xe000ed6c: Instruction Set Attributes 3            |
| ID_ISAR4    | 32   | 1310132     | rw | 0xe000ed70: Instruction Set Attributes 4            |
| ID_ISAR5    | 32   | 0           | rw | 0xe000ed74: Instruction Set Attributes 5            |
| CLIDR       | 32   | 9000003     | r- | 0xe000ed78: Cache Level ID                          |
| CTR         | 32   | 8003c003    | r- | 0xe000ed7c: Cache Type                              |
| CCSIDR      | 32   | f03fe019    | r- | 0xe000ed80: Cache Size ID                           |
| CSSELR      | 32   | 0           | rw | 0xe000ed84: Cache Size Selection                    |
| CPACR       | 32   | 0           | rw | 0xe000ed88: Coprocessor Access Control              |
| MPU_TYPE    | 32   | 800         | rw | 0xe000ed90: MPU Type                                |
| MPU_CONTROL | 32   | 0           | rw | 0xe000ed94: MPU Control                             |

|             |    |    |    |  |
|-------------|----|----|----|--|
| MPU_RNR     | 32 | 0  | rw | 0xe000ed98: MPU Region Number                                    |
| MPU_RBAR    | 32 | 0  | rw | 0xe000ed9c: MPU Region Base Address                              |
| MPU_RASR    | 32 | 0  | rw | 0xe000eda0: MPU Region Attribute and Size                        |
| MPU_RBAR_A1 | 32 | 0  | rw | 0xe000eda4: MPU Region Base Address Alias 1                      |
| MPU_RASR_A1 | 32 | 0  | rw | 0xe000eda8: MPU Region Attribute and Size Alias 1                |
| MPU_RBAR_A2 | 32 | 0  | rw | 0xe000edac: MPU Region Base Address Alias 2                      |
| MPU_RASR_A2 | 32 | 0  | rw | 0xe000edb0: MPU Region Attribute and Size Alias 2                |
| MPU_RBAR_A3 | 32 | 0  | rw | 0xe000edb4: MPU Region Base Address Alias 3                      |
| MPU_RASR_A3 | 32 | 0  | rw | 0xe000edb8: MPU Region Attribute and Size Alias 3                |
| DEMCR       | 32 | 0  | rw | 0xe000edfc: Debug Exception and Monitor Control                  |
| STIR        | 32 | -  | -w | 0xe000ef00: Software Triggered Interrupt                         |
| ICIALLU     | 32 | -  | -w | 0xe000ef50: Instruction Cache Invalidate All to PoU              |
| ICIMVAU     | 32 | -  | -w | 0xe000ef58: Instruction Cache Invalidate by Address to PoU       |
| DCIMVAC     | 32 | -  | -w | 0xe000ef5c: Data Cache Invalidate by Address to PoC              |
| DCISW       | 32 | -  | -w | 0xe000ef60: Data Cache Invalidate by Set/Way                     |
| DCCMVAU     | 32 | -  | -w | 0xe000ef64: Data Cache Invalidate by Address to PoU              |
| DCCMVAC     | 32 | -  | -w | 0xe000ef68: Data Cache Clean by Address to PoC                   |
| DCCSW       | 32 | -  | -w | 0xe000ef6c: Data Cache Clean by Set/Way                          |
| DCCIMVAC    | 32 | -  | -w | 0xe000ef70: Data Cache Clean and Invalidate by Address to PoC    |
| DCCISW      | 32 | -  | -w | 0xe000ef74: Data Cache Clean and Invalidate by Set/Way           |
| BPIALL      | 32 | -  | -w | 0xe000ef78: Branch Predictor Invalidate All                      |
| CM7_ITCMCR  | 32 | 28 | rw | 0xe000ef90: Instruction Tightly-Coupled memory Control           |
| CM7_DTCMCR  | 32 | 28 | rw | 0xe000ef94: Data Tightly-Coupled memory Control                  |
| CM7_AHBPCR  | 32 | 8  | rw | 0xe000ef98: AHB Control  |
| CM7_CACR    | 32 | 0  | rw | 0xe000ef9c: L1 Cache Control (register interface only)           |
| CM7_AHBSCR  | 32 | 0  | rw | 0xe000efa0: AHB Slave Control (register interface only)          |
| CM7_ABFSR   | 32 | 0  | rw | 0xe000efa8: Auxiliary Bus Fault Status (register interface only) |
| IEBR0       | 32 | 0  | rw | 0xe000efb0: Instruction Error Bank 0 (RAZ/WI)                    |
| IEBR1       | 32 | 0  | rw | 0xe000efb4: Instruction Error Bank 1 (RAZ/WI)                    |
| DEBR0       | 32 | 0  | rw | 0xe000efb8: Data Error Bank 0 (RAZ/WI)                           |
| DEBR1       | 32 | 0  | rw | 0xe000efbc: Data Error Bank 1 (RAZ/WI)                           |

Table 13.3: Registers at level 1, group:System

### 13.1.4 Integration support

Registers at level:1, group:Integration\_support

| Name         | Bits | Initial-Hex      | RW | Description   |
|--------------|------|------------------|----|---|
| executionPri | 32   | 7fffffff         | r- | current execution priority level                        |
| stackDomain  | 64   | 7f6f<br>b1ce5ac8 | r- | stack domain for current execution level                |
| HaltReason   | 8    | 0                | r- | bit field indicating halt reason                        |
| atomicType   | 8    | 0                | r- | current atomic instruction type (1:atomic, 2:exclusive) |

Table 13.4: Registers at level 1, group:Integration\_support