OVP Guide to Using Processor Models

Model Specific Information for variant ARM_Cortex-M4

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_armm_Cortex-M4.pdf
Created	25 August 2015

Copyright Notice

Right to Copy Documentation

Destination Control Statement

Disclaimer

Table of Contents

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

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 implemented. Use parameter override_MPU_TYPE to disable it or change the number of MPU regions if required.
SysTick timer is implemented. Use parameter SysTickPresent to disable it if required.
FPU extension is not implemented. Use parameter override_MVFR0 to enable it if required.
DSP extension is implemented. Use parameter override_InstructionAttributes3 to disable it if required.
Bit-band region is not implemented. Use parameter BitBandPresent to enable it if required.

2.0 Configuration

2.1 Location

The model source and object file is found in the VLNV tree at:
arm.ovpworld.org/processor/armm/1.0

2.2 GDB Path

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

2.3 Semi-Host Library

The default semi-host library file is found in the VLNV tree at :
arm.ovpworld.org/semihosting/armNewlib/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: 0x28

3.0 Other Variants in this Model

Table 1.

Variant
ARMv6-M
ARMv7-M
Cortex-M0
Cortex-M0plus
Cortex-M1
Cortex-M3
Cortex-M4
Cortex-M4F

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
sysResetReq	output
intISS	output
eventOut	output
lockup	output
int	input
reset	input
nmi	input
eventIn	input
int0	input
int1	input
int2	input
int3	input
int4	input
int5	input
int6	input
int7	input

int8	input
int9	input
int10	input
int11	input
int12	input
int13	input
int14	input
int15	input

6.0 FIFO Ports

No FIFO Ports in this model.

7.0 Parameters

Table 4.

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=0 gdb=1 nopBKPT=2
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	Boolean	Specify presence of SysTick timer
BitBandPresent	Boolean	Specify presence of bit-band region
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_InstructionAttributes0	Uns32	Override InstructionAttributes0 register
override_InstructionAttributes1	Uns32	Override InstructionAttributes1 register
override_InstructionAttributes2	Uns32	Override InstructionAttributes2 register
override_InstructionAttributes3	Uns32	Override InstructionAttributes3 register
override_InstructionAttributes4	Uns32	Override InstructionAttributes4 register
override_InstructionAttributes5	Uns32	Override InstructionAttributes5 register
override_MVFR0	Uns32	Override MVFR0 register
override_MVFR1	Uns32	Override MVFR1 register

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 (default is 16)

8.0 Execution Modes

Table 5.

Name	Code
Thread	0
Handler	1

9.0 Exceptions

Table 6.

Name	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

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:

This level in the model hierarchy has 2 commands.

This level in the model hierarchy has 4 register groups:

Table 7.

Group name	Registers
Core	16
Control	7
System	57
Integration_support	2

This level in the model hierarchy has no children.

11.0 Model Commands

11.1 Level 1:

Table 8.

Name	Arguments
isync	specify instruction address range for synchronous execution
itrace	enable or disable instruction tracing

12.0 Registers

12.1 Level 1:

12.1.1 Core

Table 9.

Name	Bits	Initial value (Hex)		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

12.1.2 Control

Table 10.

Name	Bits	Initial value (Hex)		Description
cpsr	32	0	rw	
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

12.1.3 System

Table 11.

Name	Bits	Initial value (Hex)		Description
ICTR	32	0	rw	Address 0xe000e004
ACTLR	32	0	rw	Address 0xe000e008
SYST_CSR	32	4	rw	Address 0xe000e010
SYST_RVR	32	0	rw	Address 0xe000e014
SYST_CVR	32	0	rw	Address 0xe000e018
SYST_CALIB	32	0	rw	Address 0xe000e01c
NVIC_ISER0	32	0	rw	Address 0xe000e100
NVIC_ICER0	32	0	rw	Address 0xe000e180
NVIC_ISPR0	32	0	rw	Address 0xe000e200
NVIC_ICPR0	32	0	rw	Address 0xe000e280
NVIC_IABR0	32	0	r-	Address 0xe000e300
NVIC_IPR0	32	0	rw	Address 0xe000e400
NVIC_IPR1	32	0	rw	Address 0xe000e404
NVIC_IPR2	32	0	rw	Address 0xe000e408
NVIC_IPR3	32	0	rw	Address 0xe000e40c
CPUID	32	410fc241	r-	Address 0xe000ed00
ICSR	32	1000	rw	Address 0xe000ed04
VTOR	32	0	rw	Address 0xe000ed08
AIRCR	32	fa050000	rw	Address 0xe000ed0c
SCR	32	0	rw	Address 0xe000ed10
CCR	32	200	rw	Address 0xe000ed14
SHPR1	32	0	rw	Address 0xe000ed18
SHPR2	32	0	rw	Address 0xe000ed1c
SHPR3	32	0	rw	Address 0xe000ed20
SHCSR	32	0	rw	Address 0xe000ed24
CFSR	32	0	rw	Address 0xe000ed28
HFSR	32	0	rw	Address 0xe000ed2c
MMAR	32	0	rw	Address 0xe000ed34
BFAR	32	0	rw	Address 0xe000ed38
AFSR	32	0	rw	Address 0xe000ed3c
ID_PFR0	32	30	rw	Address 0xe000ed40
ID_PFR1	32	200	rw	Address 0xe000ed44

ID_DFR0	32	0	rw	Address 0xe000ed48
ID_AFR0	32	0	rw	Address 0xe000ed4c
ID_MMFR0	32	30	rw	Address 0xe000ed50
ID_MMFR1	32	0	rw	Address 0xe000ed54
ID_MMFR2	32	0	rw	Address 0xe000ed58
ID_MMFR3	32	0	rw	Address 0xe000ed5c
ID_ISAR0	32	1141110	rw	Address 0xe000ed60
ID_ISAR1	32	2112000	rw	Address 0xe000ed64
ID_ISAR2	32	21232231	rw	Address 0xe000ed68
ID_ISAR3	32	1111131	rw	Address 0xe000ed6c
ID_ISAR4	32	1310132	rw	Address 0xe000ed70
ID_ISAR5	32	0	rw	Address 0xe000ed74
CPACR	32	0	rw	Address 0xe000ed88
MPU_TYPE	32	800	rw	Address 0xe000ed90
MPU_CONTROL	32	0	rw	Address 0xe000ed94
MPU_RNR	32	0	rw	Address 0xe000ed98
MPU_RBAR	32	0	rw	Address 0xe000ed9c
MPU_RASR	32	0	rw	Address 0xe000eda0
MPU_RBAR_A1	32	0	rw	Address 0xe000eda4
MPU_RASR_A1	32	0	rw	Address 0xe000eda8
MPU_RBAR_A2	32	0	rw	Address 0xe000edac
MPU_RASR_A2	32	0	rw	Address 0xe000edb0
MPU_RBAR_A3	32	0	rw	Address 0xe000edb4
MPU_RASR_A3	32	0	rw	Address 0xe000edb8
STIR	32	-	-w	Address 0xe000ef00 (User mode access:-w)

12.1.4 Integration_support

Table 12.

Name	Bits	Initial value (Hex)		Description
executionPri	32	7fffffff	r-	current execution priority level
stackDomain	32	970e918	r-	stack domain for current execution level

#

