OVP Guide to Using Processor Models

Model Specific Information for variant ARM_ARM1176JZ-S

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_arm_ARM1176JZ-S.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

ARM 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.

Source of model available under separate Imperas Software License Agreement.

1.3 Limitations

Instruction pipelines are not modeled in any way. All instructions are assumed to complete immediately. This means that instruction barrier instructions (e.g. ISB, CP15ISB) are treated

as NOPs, with the exception of any undefined instruction behavior, which is modeled. The model does not implement speculative fetch behavior. The branch cache is not modeled. Caches and write buffers are not modeled in any way. All loads, fetches and stores complete immediately and in order, and are fully synchronous (as if the memory was of Strongly Ordered or Device-nGnRnE type). Data barrier instructions (e.g. DSB, CP15DSB) are treated as NOPs, with the exception of any undefined instruction behavior, which is modeled. Cache manipulation instructions are implemented as NOPs, with the exception of any undefined instruction behavior, which is modeled.
Real-world timing effects are not modeled: all instructions are assumed to complete in a single cycle.
TLBs are architecturally-accurate but not device accurate. This means that all TLB maintenance and address translation operations are fully implemented but the cache is larger than in the real device.

1.4 Verification

Models have been extensively tested by Imperas. ARM11 models have been successfully used by customers to simulate Linux and Nucleus on ArmIntegrator virtual platforms.

1.5 Features

FCSE extension is implemented.
Thumb instructions are supported.
Trivial Jazelle extension is implemented.
Security extensions are implemented (also known as TrustZone). Non-secure accesses can be made visible externally by connecting the processor to a 41-bit physical bus, in which case bits 39..0 give the true physical address and bit 40 is the NS bit.
VMSA secure and non-secure address translation is implemented.
2 ITCMs are implemented.
2 DTCMs are implemented.
Vectored Interrupt Controller Port (VIC port) is implemented.

2.0 Configuration

2.1 Location

The model source and object file is found in the VLNV tree at:
arm.ovpworld.org/processor/arm/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
ARMv4T
ARMv4xM
ARMv4
ARMv4TxM
ARMv5xM
ARMv5
ARMv5TxM
ARMv5T
ARMv5TExP
ARMv5TE
ARMv5TEJ
ARMv6
ARMv6K
ARMv6T2
ARMv6KZ
ARMv7
ARM7TDMI
ARM7EJ-S
ARM720T
ARM920T
ARM922T
ARM926EJ-S
ARM940T
ARM946E
ARM966E
ARM968E-S
ARM1020E
ARM1022E
ARM1026EJ-S
ARM1136J-S
ARM1156T2-S
ARM1176JZ-S
Cortex-R4
Cortex-R4F
Cortex-A5UP
Cortex-A5MPx1
Cortex-A5MPx2
Cortex-A5MPx3
Cortex-A5MPx4
Cortex-A8

Cortex-A9UP
Cortex-A9MPx1
Cortex-A9MPx2
Cortex-A9MPx3
Cortex-A9MPx4
Cortex-A7UP
Cortex-A7MPx1
Cortex-A7MPx2
Cortex-A7MPx3
Cortex-A7MPx4
Cortex-A15UP
Cortex-A15MPx1
Cortex-A15MPx2
Cortex-A15MPx3
Cortex-A15MPx4
Cortex-A17MPx1
Cortex-A17MPx2
Cortex-A17MPx3
Cortex-A17MPx4
AArch32
AArch64
Cortex-A53MPx1
Cortex-A53MPx2
Cortex-A53MPx3
Cortex-A53MPx4
Cortex-A57MPx1
Cortex-A57MPx2
Cortex-A57MPx3
Cortex-A57MPx4

4.0 Bus Ports

Table 2.

Type	Name	Bits	Description
master (initiator)	ITCM0	32	instruction TCM
master (initiator)	ITCM1	32	instruction TCM
master (initiator)	DTCM0	32	data TCM
master (initiator)	DTCM1	32	data TCM
master (initiator)	INSTRUCTION	32	
master (initiator)	DATA	32	

5.0 Net Ports

Table 3.

Name	Type	Description
reset	input	Processor reset, active high
fiq	input	FIQ interrupt, active high (negation of nFIQ)
irq	input	IRQ interrupt, active high (negation of nIRQ)
AXI_SLVERR	input	AXI external abort type (DECERR=0, SLVERR=1)
CP15SDISABLE	input	CP15SDISABLE (active high)
DMAIRQ	output	DMA nonsecure interrupt (active high)
DMASIRQ	output	DMA secure interrupt (active high)
DMAEXTERRIRQ	output	DMA error (active high)
PMUIRQ	output	Performance monitor event (active high)
VALRESET	output	Validation reset interrupt (active high)
VALIRQ	output	Validation IRQ interrupt (active high)
VALFIQ	output	Validation FIQ interrupt (active high)
VALDEBUG	output	Validation Debug interrupt (active high)
VICACK	output	VIC Port acknowledge (active high)
VICADDR	input	VIC Port Address (32 bit value)

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
compatibility	Enumeration	Specify compatibility mode ISA=0 gdb=1 nopSVC=2
override_debugMask	Uns32	Specifies debug mask, enabling debug output for model components
override_fcsePresent	Boolean	Specifies that FCSE is present (if true)
override_physicalBits	Uns32	Specifies the implemented physical bus bits (defaults to connected physical bus width)
override_SCTLR_V	Boolean	Override SCTLR.V with the passed value (enables high vectors)
override_SCTLR_CP15BEN_Present	Boolean	Enable ARMv7 SCTLR.CP15BEN bit (CP15 barrier enable)
override_MIDR	Uns32	Override MIDR register
override_CTR	Uns32	Override CTR register
override_TLBTR	Uns32	Override TLBTR register
override_CLIDR	Uns32	Override CLIDR register
override_AIDR	Uns32	Override AIDR register

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_ERG	Uns32	Specifies exclusive reservation granule
override_STRoffsetPC12	Boolean	Specifies that STR/STR of PC should do so with 12:byte offset from the current instruction (if true), otherwise an 8:byte offset is used
override_fcseRequiresMMU	Boolean	Specifies that FCSE is active only when MMU is enabled (if true)
override_ignoreBadCp15	Boolean	Specifies whether invalid coprocessor 15 access should be ignored (if true) or cause Invalid Instruction exceptions (if false)
override_SGIDisable	Boolean	Override whether GIC SGIs may be disabled (if true) or are permanently enabled (if false)
override_condUndefined	Boolean	Force undefined instructions to take Undefined Instruction exception even if they are conditional
override_deviceStrongAligned	Boolean	Force accesses to Device and Strongly Ordered regions to be aligned
override_Control_V	Boolean	Override SCTLR.V with the passed value (deprecated, use override_SCTLR_V)
override_MainId	Uns32	Override MIDR register (deprecated, use override_MIDR)
override_CacheType	Uns32	Override CTR register (deprecated, use override_CTR)
override_TLBType	Uns32	Override TLBTR register (deprecated, use override_TLBTR)
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)

8.0 Execution Modes

Table 5.

Name	Code
User	16
FIQ	17
IRQ	18
Supervisor	19
Monitor	22
Abort	23
Undefined	27
System	31

9.0 Exceptions

Table 6.

Name	Code
Reset	0
Undefined	1
SupervisorCall	2
SecureMonitorCall	3
PrefetchAbort	5
DataAbort	6
IRQ	8
FIQ	9

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 4 commands.

This level in the model hierarchy has 14 register groups:

Table 7.

Group name	Registers
Core	16
Control	3
User	7
FIQ	8
IRQ	3
Supervisor	3
Monitor	3
Undefined	3
Abort	3
Coprocessor_32_bit	124
Coprocessor_32_bit_secure	19
Coprocessor_32_bit_non_secure	19
Coprocessor_64_bit	4
Integration_support	2

This level in the model hierarchy has no children.

11.0 Model Commands

11.1 Level 1: CPU

Table 8.

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

12.0 Registers

12.1 Level 1: CPU

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
fps	32	0	rw	archaic FPSCR view (for gdb)
cpsr	32	1d3	rw	
spsr	32	0	rw	

12.1.3 User

Table 11.

Name	Bits	Initial value (Hex)		Description
r8_usr	32	0	rw	
r9_usr	32	0	rw	
r10_usr	32	0	rw	
r11_usr	32	0	rw	
r12_usr	32	0	rw	
sp_usr	32	0	rw	
lr_usr	32	0	rw	

12.1.4 FIQ

Table 12.

Name	Bits	Initial value (Hex)		Description
r8_fiq	32	0	rw	
r9_fiq	32	0	rw	
r10_fiq	32	0	rw	
r11_fiq	32	0	rw	
r12_fiq	32	0	rw	
sp_fiq	32	0	rw	
lr_fiq	32	0	rw	
spsr_fiq	32	0	rw	

12.1.5 IRQ

Table 13.

Name	Bits	Initial value (Hex)		Description
sp_irq	32	0	rw	
lr_irq	32	0	rw	
spsr_irq	32	0	rw	

12.1.6 Supervisor

Table 14.

Name	Bits	Initial value (Hex)		Description
sp_svc	32	0	rw	
lr_svc	32	0	rw	
spsr_svc	32	0	rw	

12.1.7 Monitor

Table 15.

Name	Bits	Initial value (Hex)		Description
sp_mon	32	0	rw	
lr_mon	32	0	rw	
spsr_mon	32	0	rw	

12.1.8 Undefined

Table 16.

Name	Bits	Initial value (Hex)		Description
sp_undef	32	0	rw	
lr_undef	32	0	rw	
spsr_undef	32	0	rw	

12.1.9 Abort

Table 17.

Name	Bits	Initial value (Hex)		Description
sp_abt	32	0	rw	
lr_abt	32	0	rw	
spsr_abt	32	0	rw	

12.1.10 Coprocessor_32_bit

Table 18.

Name	Bits	Initial value (Hex)		Description
ACTLR	32	7	rw	Auxiliary Control
ATS1CPR	32	-	-w	Address Translate Stage 1 Current State EL1 Read
ATS1CPW	32	-	-w	Address Translate Stage 1 Current State EL1 Write
ATS1CUR	32	-	-w	Address Translate Stage 1 Current State Unprivileged Read
ATS1CUW	32	-	-w	Address Translate Stage 1 Current State Unprivileged Write
ATS12NSOPR	32	-	-w	Address Translate Stages 1 and 2 Non-Secure Only EL1 Read
ATS12NSOPW	32	-	-w	Address Translate Stages 1 and 2 Non-Secure Only EL1 Write
ATS12NSOUR	32	-	-w	Address Translate Stages 1 and 2 Non-Secure Only Unprivileged Read
ATS12NSOUW	32	-	-w	Address Translate Stages 1 and 2 Non-Secure Only Unprivileged Write

CBOR	32	0	rw	Cache Behavior Override
CCNT	32	0	rw	Cycle Counter
CONTEXTIDR	32	0	rw	Context ID
CPACR	32	0	rw	Coprocessor Access Control
CTR	32	1d152152	r-	Cache Type
CleanDCache	32	-	-w	Clean Data Cache
CleanDCacheLineMVA	32	-	-w	Data Cache Line Clean by VA
CleanDCacheLineSW	32	-	-w	Data Cache Line Clean by Set/Way
CleanInvalDCache	32	-	-w	Data Cache Clean and Invalidate
CleanInvalDCacheLineMVA	32	-	-w	Data Cache Line Clean and Invalidate by VA
CleanInvalDCacheLineSW	32	-	-w	Data Cache Line Clean and Invalidate by Set/Way
DACR	32	0	rw	Domain Access Control
DBGDIDR	32	0	r-	Debug ID
DCLR	32	fffffff0	rw	Data Cache Lockdown
DCacheMasterValid	32	0	rw	Data Cache Master Valid
DFAR	32	0	rw	Data Fault Address
DFSR	32	0	rw	Data Fault Status
DMAChannel	32	0	rw	DMA Channel
DMAClear	32	-	-w	DMA Clear
DMAContextID	32	0	rw	DMA Context ID
DMAControl	32	0	rw	DMA Control
DMAExternalStart	32	0	rw	DMA External Start Address
DMAInternalEnd	32	0	rw	DMA Internal End Address
DMAInternalStart	32	0	rw	DMA Internal Start Address
DMAInterrupting	32	0	r-	DMA Interrupting
DMAPresent	32	3	r-	DMA Present
DMAQueued	32	0	r-	DMA Queued
DMARunning	32	0	r-	DMA Running
DMAStart	32	-	-w	DMA Start
DMAStatus	32	0	r-	DMA Status
DMAStop	32	-	-w	DMA Stop
DMAUserAccessibility	32	0	rw	DMA User Accessibility
DTCMNSCAR	32	0	rw	DTCM Non-Secure Access
DTCMRR	32	10	rw	DTCM Region
DTLBIALL	32	-	-w	Invalidate Entire Data TLB
DTLBIASID	32	-	-w	Invalidate Data TLB by ASID
DTLBIMVA	32	-	-w	Invalidate Data TLB by VA
DTLBLR	32	0	rw	TLB Lockdown
DataMemoryBarrier	32	-	-w	Data Memory Barrier
DataSynchBarrier	32	-	-w	Data Synchronization Barrier
DirtyStatus	32	0	r-	Dirty Status

FCSEIDR	32	0	rw	FCSE Process ID
FlushBranchTargetCache	32	-	-w	Flush Branch Target Cache
FlushBranchTargetEntry	32	-	-w	Flush Branch Target Entry
FlushPrefetchBuffer	32	-	-w	Flush Prefetch Buffer
ICLR	32	fffffff0	rw	Instruction Cache Lockdown
ICacheMasterValid	32	0	rw	Instruction Cache Master Valid
ID_AFR0	32	0	r-	Auxiliary Feature 0
ID_DFR0	32	0	r-	Debug Feature 0
ID_ISAR0	32	140011	r-	Instruction Set Attribute 0
ID_ISAR1	32	12002111	r-	Instruction Set Attribute 1
ID_ISAR2	32	11231121	r-	Instruction Set Attribute 2
ID_ISAR3	32	1102131	r-	Instruction Set Attribute 3
ID_ISAR4	32	1141	r-	Instruction Set Attribute 4
ID_ISAR5	32	0	r-	Instruction Set Attribute 5
ID_MMFR0	32	1130003	r-	Memory Model Feature 0
ID_MMFR1	32	10030302	r-	Memory Model Feature 1
ID_MMFR2	32	1222100	r-	Memory Model Feature 2
ID_MMFR3	32	0	r-	Memory Model Feature 3
ID_PFR0	32	111	r-	Processor Feature 0
ID_PFR1	32	11	r-	Processor Feature 1
IFAR	32	0	rw	Instruction Fault Address
IFSR	32	0	rw	Instruction Fault Status
ISR	32	0	r-	Interrupt Status
ITCMNSCAR	32	0	rw	ITCM Non-Secure Access
ITCMRR	32	10	rw	ITCM Region
ITLBIALL	32	-	-w	Invalidate Entire Instruction TLB
ITLBIASID	32	-	-w	Invalidate Instruction TLB by ASID
ITLBIMVA	32	-	-w	Invalidate Instruction TLB by VA
InvalDCache	32	-	-w	Invalidate Data Cache
InvalDCacheLineMVA	32	-	-w	Invalidate Data Cache Line by VA
InvalDCacheLineSW	32	-	-w	Invalidate Data Cache Line by Set/Way
InvalICache	32	-	-w	Invalidate Instruction Cache
InvalICacheLineMVA	32	-	-w	Invalidate Instruction Cache Line by VA
InvalICacheLineSW	32	-	-w	Invalidate Instruction Cache Line by Set/Way
InvalUnified	32	-	-w	Invalidate Unified Cache
JIDR	32	0	rw	Jazelle ID
JMCR	32	0	rw	Jazelle Main Configuration
JOSCR	32	0	rw	Jazelle OS Control
MIDR	32	410fb764	r-	Main ID
MVBAR	32	0	rw	Monitor Vector Base Address
NMRR	32	44e048e0	rw	Normal Memory Remap
NSACR	32	0	rw	Non-Secure Access Control

PAR	32	0	rw	Physical Address
PMN0	32	0	rw	Count 0
PMN1	32	0	rw	Count 1
PMNC	32	0	rw	Performance Monitor Control
PRRR	32	98aa4	rw	Primary Region Remap
PrefetchICacheLine	32	-	-w	Prefetch Instruction Cache Line
SCR	32	0	rw	Secure Configuration
SCTLR	32	50078	rw	System Control
SDER	32	0	rw	Secure Debug Enable
SUNSAVC	32	0	rw	Secure User and Non-Secure Access Validation
TCMSR	32	0	rw	TCM Select
TCMTR	32	20002	r-	TCM Type
TLBIALL	32	-	-w	Invalidate Entire Unified TLB
TLBIASID	32	-	-w	Invalidate Unified TLB by ASID
TLBIMVA	32	-	-w	Invalidate Unified TLB by VA
TLBLDATTR	32	0	rw	TLB Lockdown Attributes
TLBLDI	32	0	rw	TLB Lockdown Index
TLBLDPA	32	0	rw	TLB Lockdown PA
TLBLDVA	32	0	rw	TLB Lockdown VA
TLBTR	32	800	r-	TLB Type
TPIDRPRW	32	0	rw	PL0 Read/Write Software Thread ID
TPIDRURO	32	0	rw	PL0 Read-Only Software Thread ID
TPIDRURW	32	0	rw	PL1 Software Thread ID
TTBCR	32	0	rw	Translation Table Base Control
TTBR0	32	0	rw	Translation Table Base 0
TTBR1	32	0	rw	Translation Table Base 1
VALCACHESIZEMASK	32	5555	rw	System Validation Cache Size Mask
VALDEBUGCNT	32	0	rw	External Debug Request Counter
VALOPERATIONS	32	0	rw	System Validation Operations
VBAR	32	0	rw	Vector Base Address
WFAR	32	0	rw	Watchpoint Fault Address
WaitForInterrupt	32	-	-w	Wait For Interrupt

12.1.11 Coprocessor_32_bit_secure

Table 19.

Name	**Bits**	**Initial value (Hex)**		**Description**
CONTEXTIDR_S	32	0	rw	Context ID
DACR_S	32	0	rw	Domain Access Control
DFAR_S	32	0	rw	Data Fault Address
DFSR_S	32	0	rw	Data Fault Status

FCSEIDR_S	32	0	rw	FCSE Process ID
IFAR_S	32	0	rw	Instruction Fault Address
IFSR_S	32	0	rw	Instruction Fault Status
NMRR_S	32	44e048e0	rw	Normal Memory Remap
PAR_S	32	0	rw	Physical Address
PRRR_S	32	98aa4	rw	Primary Region Remap
SCTLR_S	32	50078	rw	System Control
TCMSR_S	32	0	rw	TCM Select
TPIDRPRW_S	32	0	rw	PL0 Read/Write Software Thread ID
TPIDRURO_S	32	0	rw	PL0 Read-Only Software Thread ID
TPIDRURW_S	32	0	rw	PL1 Software Thread ID
TTBCR_S	32	0	rw	Translation Table Base Control
TTBR0_S	32	0	rw	Translation Table Base 0
TTBR1_S	32	0	rw	Translation Table Base 1
VBAR_S	32	0	rw	Vector Base Address

12.1.12 Coprocessor_32_bit_non_secure

Table 20.

Name	**Bits**	**Initial value (Hex)**		**Description**
CONTEXTIDR_NS	32	0	rw	Context ID
DACR_NS	32	0	rw	Domain Access Control
DFAR_NS	32	0	rw	Data Fault Address
DFSR_NS	32	0	rw	Data Fault Status
FCSEIDR_NS	32	0	rw	FCSE Process ID
IFAR_NS	32	0	rw	Instruction Fault Address
IFSR_NS	32	0	rw	Instruction Fault Status
NMRR_NS	32	44e048e0	rw	Normal Memory Remap
PAR_NS	32	0	rw	Physical Address
PRRR_NS	32	98aa4	rw	Primary Region Remap
SCTLR_NS	32	50078	rw	System Control
TCMSR_NS	32	0	rw	TCM Select
TPIDRPRW_NS	32	0	rw	PL0 Read/Write Software Thread ID
TPIDRURO_NS	32	0	rw	PL0 Read-Only Software Thread ID
TPIDRURW_NS	32	0	rw	PL1 Software Thread ID
TTBCR_NS	32	0	rw	Translation Table Base Control
TTBR0_NS	32	0	rw	Translation Table Base 0
TTBR1_NS	32	0	rw	Translation Table Base 1
VBAR_NS	32	0	rw	Vector Base Address

12.1.13 Coprocessor_64_bit

Table 21.

Name	Bits	Initial value (Hex)		Description
CleanDCacheRange	64	-	-w	Clean Data Cache Range
CleanInvalDCacheRange	64	-	-w	Clean and Invalidate Data Cache Range
InvalDCacheRange	64	-	-w	Invalidate Data Cache Range
InvalICacheRange	64	-	-w	Invalidate Instruction Cache Range

12.1.14 Integration_support

Table 22.

Name	Bits	Initial value (Hex)		Description
transactPL	32	1	r-	privilege level of current memory transaction
transactAT	32	0	r-	current memory transaction type: PA=1, VA=0

#

