OVP Guide to Using Processor Models

Model Specific Information for variant ARM_AArch64

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_AArch64.pdf
Created	25 August 2015

Copyright Notice

Right to Copy Documentation

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

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.

ARMv8 architecture models additionally require a run time model license - contact Imperas for more information.

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.

Performance Monitors are implemented as a register interface only.

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.

Debug registers are implemented but non-functional (which is sufficient to allow operating systems such as Linux to boot). Debug state is not implemented.

This initial ARMv8 model also has the following specific limitations, which will be rectified shortly:

The model currently implements only a memory-mapped GICv2, not a register-accessible GICv3.

The optional CRC32 instructions are not supported.

The optional SIMD Cryptographic Extension instructions are not supported.

1.4 Verification

Models have been extensively tested by Imperas. ARM Cortex-A models have been successfully used by customers to simulate SMP Linux, Ubuntu Desktop, VxWorks and ThreadX on Xilinx Zynq virtual platforms.

1.5 Features

AArch64 is implemented at EL3, EL2, EL1 and EL0.

SIMD, VFP and LPA (large physical address extension) are implemented as standard in ARMv8.

Security extensions are implemented (also known as TrustZone). To make non-secure accesses visible externally, override ID_AA64MMFR0_EL1.PARange to specify the required physical bus size (32, 36, 40, 42, 44 or 48 bits) and connect the processor to a bus one bit wider (33, 37, 41, 43, 45 or 49 bits, respectively). The extra most-significant bit is the NS bit, indicating a non-secure access. If non-secure accesses are not required to be made visible externally, connect the processor to a bus of exactly the size implied by ID_AA64MMFR0_EL1.PARange.

VMSA EL1, EL2 and EL3 stage 1 address translation is implemented. VMSA stage 2 address translation is implemented.

Generic Timer is present. Use parameter override_timerScaleFactor to specify the counter rate as a fraction of the processor MIPS rate (e.g. 10 implies Generic Timer counters increment once every 10 processor instructions).

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/aarch64-none-elf-gdb

2.3 Semi-Host Library

The default semi-host library file is found in the VLNV tree at :

arm.ovpworld.org/semihosting/armAngel/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: 0xb7

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
master (initiator)	INSTRUCTION	32
master (initiator)	DATA	32

5.0 Net Ports

Table 3.

Name	Type	Description
CNTVIRQ	output	Virtual timer event (active high)
CNTPSIRQ	output	Secure physical timer event (active high)
CNTPNSIRQ	output	Non-secure physical timer event (active high)
CNTPHPIRQ	output	Hypervisor physical timer event (active high)
VINITHI	input	Configure HIVECS mode (SCTLR.V)
CFGEND	input	Configure exception endianness (SCTLR.EE)
CFGTE	input	Configure exception state at reset (SCTLR.TE)
reset	input	Processor reset, active high
fiq	input	FIQ interrupt, active high (negation of nFIQ)
irq	input	IRQ interrupt, active high (negation of nIRQ)
vfiq	input	Virtual FIQ interrupt, active high (negation of nVFIQ)
virq	input	Virtual IRQ interrupt, active high (negation of nVIRQ)
AXI_SLVERR	input	AXI external abort type (DECERR=0, SLVERR=1)
CP15SDISABLE	input	CP15SDISABLE (active high)

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 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_fpexcDexPresent	Boolean	Specifies that the FPEXC.DEX register field is implemented (if true)
override_advSIMDPresent	Boolean	Specifies that Advanced SIMD extensions are present (if true)
override_vfpPresent	Boolean	Specifies that VFP extensions are 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_PMCR	Uns32	Override PMCR register (not functionally significant in the model)
override_PMCEID0	Uns32	Override PMCEID0 register (not functionally significant in the model)
override_PMCEID1	Uns32	Override PMCEID1 register (not functionally significant in the model)
override_FPSID	Uns32	Override SIMD/VFP FPSID register
override_MVFR0	Uns32	Override SIMD/VFP MVFR0 register
override_MVFR1	Uns32	Override SIMD/VFP MVFR1 register
override_MVFR2	Uns32	Override SIMD/VFP MVFR2 register
override_FPEXC	Uns32	Override SIMD/VFP FPEXC register
override_ERG	Uns32	Specifies exclusive reservation granule
override_RMR	Uns32	Override RMR register
override_RVBAR	Uns64	Override RVBAR register
override_AA64PFR0_EL1	Uns64	Override ID_AA64PFR0_EL1 register
override_AA64PFR1_EL1	Uns64	Override ID_AA64PFR1_EL1 register
override_AA64DFR0_EL1	Uns64	Override ID_AA64DFR0_EL1 register
override_AA64DFR1_EL1	Uns64	Override ID_AA64DFR1_EL1 register
override_AA64AFR0_EL1	Uns64	Override ID_AA64AFR0_EL1 register
override_AA64AFR1_EL1	Uns64	Override ID_AA64AFR1_EL1 register

override_AA64ISAR0_EL1	Uns64	Override ID_AA64ISAR0_EL1 register
override_AA64ISAR1_EL1	Uns64	Override ID_AA64ISAR1_EL1 register
override_AA64MMFR0_EL1	Uns64	Override ID_AA64MMFR0_EL1 register
override_AA64MMFR1_EL1	Uns64	Override ID_AA64MMFR1_EL1 register
override_DCZID_EL0	Uns32	Override DCZID_EL0 register
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**

EL0t	0
EL1t	4
EL1h	5
EL2t	8
EL2h	9
EL3t	12
EL3h	13

9.0 Exceptions

Table 6.

Name	**Code**
Reset	0
Undefined	1
SupervisorCall	2
SecureMonitorCall	3
HypervisorCall	4
PrefetchAbort	5
DataAbort	6
HypervisorTrap	7
IRQ	8
FIQ	9
IllegalState	10
MisalignedPC	11
MisalignedSP	12
SError	13

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 5 register groups:

Table 7.

Group name	**Registers**
Core_AArch64	33
SIMD_FP_AArch64	32
AArch64_system	242
AArch64_system_artifact	1
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_AArch64

Table 9.

Name	Bits	Initial value (Hex)		Description
x0	64	0	rw	
x1	64	0	rw	
x2	64	0	rw	
x3	64	0	rw	
x4	64	0	rw	
x5	64	0	rw	
x6	64	0	rw	
x7	64	0	rw	
x8	64	0	rw	
x9	64	0	rw	
x10	64	0	rw	
x11	64	0	rw	
x12	64	0	rw	
x13	64	0	rw	
x14	64	0	rw	
x15	64	0	rw	
x16	64	0	rw	
x17	64	0	rw	
x18	64	0	rw	
x19	64	0	rw	
x20	64	0	rw	
x21	64	0	rw	
x22	64	0	rw	
x23	64	0	rw	
x24	64	0	rw	
x25	64	0	rw	
x26	64	0	rw	

x27	64	0	rw	
x28	64	0	rw	
x29	64	0	rw	frame pointer
x30	64	0	rw	
sp	64	0	rw	stack pointer
pc	64	0	rw	program counter

12.1.2 SIMD_FP_AArch64

Table 10.

Name	Bits	Initial value (Hex)		Description
v0	128	-	rw	
v1	128	-	rw	
v2	128	-	rw	
v3	128	-	rw	
v4	128	-	rw	
v5	128	-	rw	
v6	128	-	rw	
v7	128	-	rw	
v8	128	-	rw	
v9	128	-	rw	
v10	128	-	rw	
v11	128	-	rw	
v12	128	-	rw	
v13	128	-	rw	
v14	128	-	rw	
v15	128	-	rw	
v16	128	-	rw	
v17	128	-	rw	
v18	128	-	rw	
v19	128	-	rw	
v20	128	-	rw	
v21	128	-	rw	
v22	128	-	rw	
v23	128	-	rw	
v24	128	-	rw	
v25	128	-	rw	
v26	128	-	rw	
v27	128	-	rw	
v28	128	-	rw	
v29	128	-	rw	
v30	128	-	rw	
v31	128	-	rw	

12.1.3 AArch64_system

Table 11.

Name	**Bits**	**Initial value (Hex)**		**Description**
ACTLR_EL1	32	0	rw	Auxiliary Control (EL1)
ACTLR_EL2	32	0	rw	Auxiliary Control (EL2)
ACTLR_EL3	32	0	rw	Auxiliary Control (EL3)
AFSR0_EL1	32	0	rw	Auxiliary Fault Status 0 (EL1)
AFSR0_EL2	32	0	rw	Auxiliary Fault Status 0 (EL2)
AFSR0_EL3	32	0	rw	Auxiliary Fault Status 0 (EL3)
AFSR1_EL1	32	0	rw	Auxiliary Fault Status 1 (EL1)
AFSR1_EL2	32	0	rw	Auxiliary Fault Status 1 (EL2)
AFSR1_EL3	32	0	rw	Auxiliary Fault Status 1 (EL3)
AIDR_EL1	32	0	r-	Auxiliary ID
AMAIR_EL1	64	0	rw	Auxiliary Memory Attribute Indirection (EL1)
AMAIR_EL2	64	0	rw	Auxiliary Memory Attribute Indirection (EL2)
AMAIR_EL3	64	0	rw	Auxiliary Memory Attribute Indirection (EL3)
CCSIDR_EL1	32	701fe00a	r-	Current Cache Size ID
CLIDR_EL1	32	a000023	r-	Cache Level ID
CNTFRQ_EL0	32	4c4b40	rw	Counter-Timer Frequency
CNTHCTL_EL2	32	3	rw	Counter-Timer Hypervisor Control
CNTHP_CTL_EL2	32	0	rw	Counter-Timer Hypervisor Physical Timer Control
CNTHP_CVAL_EL2	64	0	rw	Counter-Timer Hypervisor Physical Timer CompareValue
CNTHP_TVAL_EL2	32	0	rw	Counter-Timer Hypervisor Physical Timer TimerValue
CNTKCTL_EL1	32	0	rw	Counter-Timer Kernel Control
CNTPCT_EL0	64	0	r-	Counter-Timer Physical Count
CNTPS_CTL_EL1	32	0	rw	Counter-Timer Physical Secure Timer Control
CNTPS_CVAL_EL1	64	0	rw	Counter-Timer Physical Secure Timer CompareValue
CNTPS_TVAL_EL1	32	0	rw	Counter-Timer Physical Secure Timer TimerValue
CNTP_CTL_EL0	32	0	rw	Counter-Timer Physical Timer Control
CNTP_CVAL_EL0	64	0	rw	Counter-Timer Physical Timer CompareValue
CNTP_TVAL_EL0	32	0	rw	Counter-Timer Physical Timer TimerValue

CNTVCT_EL0	64	0	r-	Counter-Timer Virtual Count
CNTVOFF_EL2	64	0	rw	Counter-Timer Virtual Offset
CNTV_CTL_EL0	32	0	rw	Counter-Timer Virtual Timer Control
CNTV_CVAL_EL0	64	0	rw	Counter-Timer Virtual Timer CompareValue
CNTV_TVAL_EL0	32	0	rw	Counter-Timer Virtual Timer TimerValue
CONTEXTIDR_EL1	32	0	rw	Context ID
CPACR_EL1	32	0	rw	Architectural Feature Access Control
CPTR_EL2	32	33ff	rw	Architectural Feature Trap (EL2)
CPTR_EL3	32	0	rw	Architectural Feature Trap (EL3)
CSSELR_EL1	32	0	rw	Current Size Selection
CTR_EL0	32	8404c004	r-	Cache Type
CurrentEL	32	c	r-	Current Exception Level
DAIF	32	3c0	rw	Interrupt Mask Bits
DBGAUTHSTATUS_EL1	32	aa	r-	Debug Authentication Status
DBGBCR0_EL1	32	0	rw	Debug Breakpoint Control 0
DBGBCR1_EL1	32	0	rw	Debug Breakpoint Control 1
DBGBCR2_EL1	32	0	rw	Debug Breakpoint Control 2
DBGBCR3_EL1	32	0	rw	Debug Breakpoint Control 3
DBGBCR4_EL1	32	0	rw	Debug Breakpoint Control 4
DBGBCR5_EL1	32	0	rw	Debug Breakpoint Control 5
DBGBCR6_EL1	32	0	rw	Debug Breakpoint Control 6
DBGBCR7_EL1	32	0	rw	Debug Breakpoint Control 7
DBGBCR8_EL1	32	0	rw	Debug Breakpoint Control 8
DBGBCR9_EL1	32	0	rw	Debug Breakpoint Control 9
DBGBCR10_EL1	32	0	rw	Debug Breakpoint Control 10
DBGBCR11_EL1	32	0	rw	Debug Breakpoint Control 11
DBGBCR12_EL1	32	0	rw	Debug Breakpoint Control 12
DBGBCR13_EL1	32	0	rw	Debug Breakpoint Control 13
DBGBCR14_EL1	32	0	rw	Debug Breakpoint Control 14
DBGBCR15_EL1	32	0	rw	Debug Breakpoint Control 15
DBGBVR0_EL1	64	0	rw	Debug Breakpoint Value 0
DBGBVR1_EL1	64	0	rw	Debug Breakpoint Value 1
DBGBVR2_EL1	64	0	rw	Debug Breakpoint Value 2
DBGBVR3_EL1	64	0	rw	Debug Breakpoint Value 3
DBGBVR4_EL1	64	0	rw	Debug Breakpoint Value 4
DBGBVR5_EL1	64	0	rw	Debug Breakpoint Value 5
DBGBVR6_EL1	64	0	rw	Debug Breakpoint Value 6
DBGBVR7_EL1	64	0	rw	Debug Breakpoint Value 7
DBGBVR8_EL1	64	0	rw	Debug Breakpoint Value 8
DBGBVR9_EL1	64	0	rw	Debug Breakpoint Value 9

DBGBVR10_EL1	64	0	rw	Debug Breakpoint Value 10
DBGBVR11_EL1	64	0	rw	Debug Breakpoint Value 11
DBGBVR12_EL1	64	0	rw	Debug Breakpoint Value 12
DBGBVR13_EL1	64	0	rw	Debug Breakpoint Value 13
DBGBVR14_EL1	64	0	rw	Debug Breakpoint Value 14
DBGBVR15_EL1	64	0	rw	Debug Breakpoint Value 15
DBGCLAIMCLR_EL1	32	0	rw	Debug Claim Tag Clear
DBGCLAIMSET_EL1	32	0	rw	Debug Claim Tag Set
DBGDTRTRX_EL0	32	0	rw	Debug Data Transfer, Transmit/ Receive
DBGDTR_EL0	64	0	rw	Debug Data Transfer
DBGPRCR_EL1	32	0	rw	Debug Power Control
DBGVCR32_EL2	32	0	rw	Debug Vector Catch
DBGWCR0_EL1	32	0	rw	Debug Watchpoint Control 0
DBGWCR1_EL1	32	0	rw	Debug Watchpoint Control 1
DBGWCR2_EL1	32	0	rw	Debug Watchpoint Control 2
DBGWCR3_EL1	32	0	rw	Debug Watchpoint Control 3
DBGWCR4_EL1	32	0	rw	Debug Watchpoint Control 4
DBGWCR5_EL1	32	0	rw	Debug Watchpoint Control 5
DBGWCR6_EL1	32	0	rw	Debug Watchpoint Control 6
DBGWCR7_EL1	32	0	rw	Debug Watchpoint Control 7
DBGWCR8_EL1	32	0	rw	Debug Watchpoint Control 8
DBGWCR9_EL1	32	0	rw	Debug Watchpoint Control 9
DBGWCR10_EL1	32	0	rw	Debug Watchpoint Control 10
DBGWCR11_EL1	32	0	rw	Debug Watchpoint Control 11
DBGWCR12_EL1	32	0	rw	Debug Watchpoint Control 12
DBGWCR13_EL1	32	0	rw	Debug Watchpoint Control 13
DBGWCR14_EL1	32	0	rw	Debug Watchpoint Control 14
DBGWCR15_EL1	32	0	rw	Debug Watchpoint Control 15
DBGWVR0_EL1	64	0	rw	Debug Watchpoint Value 0
DBGWVR1_EL1	64	0	rw	Debug Watchpoint Value 1
DBGWVR2_EL1	64	0	rw	Debug Watchpoint Value 2
DBGWVR3_EL1	64	0	rw	Debug Watchpoint Value 3
DBGWVR4_EL1	64	0	rw	Debug Watchpoint Value 4
DBGWVR5_EL1	64	0	rw	Debug Watchpoint Value 5
DBGWVR6_EL1	64	0	rw	Debug Watchpoint Value 6
DBGWVR7_EL1	64	0	rw	Debug Watchpoint Value 7
DBGWVR8_EL1	64	0	rw	Debug Watchpoint Value 8
DBGWVR9_EL1	64	0	rw	Debug Watchpoint Value 9
DBGWVR10_EL1	64	0	rw	Debug Watchpoint Value 10
DBGWVR11_EL1	64	0	rw	Debug Watchpoint Value 11
DBGWVR12_EL1	64	0	rw	Debug Watchpoint Value 12

DBGWVR13_EL1	64	0	rw	Debug Watchpoint Value 13
DBGWVR14_EL1	64	0	rw	Debug Watchpoint Value 14
DBGWVR15_EL1	64	0	rw	Debug Watchpoint Value 15
DCZID_EL0	32	4	r-	Data Cache Zero ID
DLR_EL0	64	0	rw	Debug Link
DSPSR_EL0	32	0	rw	Debug Saved Program Status
ELR_EL1	64	0	rw	Exception Link (EL1)
ELR_EL2	64	0	rw	Exception Link (EL2)
ELR_EL3	64	0	rw	Exception Link (EL3)
ESR_EL1	32	0	rw	Exception Syndrome (EL1)
ESR_EL2	32	0	rw	Exception Syndrome (EL2)
ESR_EL3	32	0	rw	Exception Syndrome (EL3)
FAR_EL1	64	0	rw	Fault Address (EL1)
FAR_EL2	64	0	rw	Fault Address (EL2)
FAR_EL3	64	0	rw	Fault Address (EL3)
FPCR	32	0	rw	Floating Point Control
FPSR	32	0	rw	Floating Point Status
HACR_EL2	32	0	rw	Hypervisor Auxiliary Control
HCR_EL2	64	0	rw	Hypervisor Configuration
HPFAR_EL2	64	0	rw	Hypervisor IPA Fault Address
HSTR_EL2	32	0	rw	Hypervisor System Trap
ID_AA64AFR0_EL1	64	0	r-	AArch64 Auxiliary Feature 0
ID_AA64AFR1_EL1	64	0	r-	AArch64 Auxiliary Feature 1
ID_AA64DFR0_EL1	64	10305106	r-	AArch64 Processor Feature 2
ID_AA64DFR1_EL1	64	0	r-	AArch64 Debug Feature 0
ID_AA64ISAR0_EL1	64	0	r-	AArch64 Instruction Set Attribute 0
ID_AA64ISAR1_EL1	64	0	r-	AArch64 Instruction Set Attribute 1
ID_AA64MMFR0_EL1	64	101124	r-	AArch64 Memory Model Feature 0
ID_AA64MMFR1_EL1	64	0	r-	AArch64 Memory Model Feature 1
ID_AA64PFR0_EL1	64	1111	r-	AArch64 Processor Feature 0
ID_AA64PFR1_EL1	64	0	r-	AArch64 Processor Feature 1
ID_AFR0_EL1	32	0	r-	Auxiliary Feature 0
ID_DFR0_EL1	32	0	r-	Debug Feature 0
ID_ISAR0_EL1	32	0	r-	Instruction Set Attribute 0
ID_ISAR1_EL1	32	0	r-	Instruction Set Attribute 1
ID_ISAR2_EL1	32	0	r-	Instruction Set Attribute 2
ID_ISAR3_EL1	32	0	r-	Instruction Set Attribute 3
ID_ISAR4_EL1	32	0	r-	Instruction Set Attribute 4
ID_ISAR5_EL1	32	0	r-	Instruction Set Attribute 5
ID_MMFR0_EL1	32	0	r-	Memory Model Feature 0
ID_MMFR1_EL1	32	0	r-	Memory Model Feature 1
ID_MMFR2_EL1	32	0	r-	Memory Model Feature 2

ID_MMFR3_EL1	32	0	r-	Memory Model Feature 3
ID_PFR0_EL1	32	0	r-	Processor Feature 0
ID_PFR1_EL1	32	0	r-	Processor Feature 1
ISR_EL1	32	0	r-	Interrupt Status
MAIR_EL1	64	44e048e000098aa4	rw	Memory Attribute Indirection (EL1)
MAIR_EL2	64	0	rw	Memory Attribute Indirection (EL2)
MAIR_EL3	64	44e048e000098aa4	rw	Memory Attribute Indirection (EL3)
MDCCINT_EL1	32	0	rw	Monitor DCC Interrupt Enable
MDCCSR_EL0	32	0	r-	Monitor DCC Status
MDCR_EL2	32	6	rw	Monitor Debug Configuration (EL2)
MDCR_EL3	32	0	rw	Monitor Debug Configuration (EL3)
MDRAR_EL1	64	0	r-	Monitor Debug ROM Address
MDSCR_EL1	32	0	rw	Monitor Debug System Control
MIDR_EL1	32	410fd000	r-	Main ID
MPIDR_EL1	64	c0000000	r-	Multiprocessor Affinity
MVFR0_EL1	32	0	r-	Media and VFP Feature 0
MVFR1_EL1	32	0	r-	Media and VFP Feature 1
MVFR2_EL1	32	0	r-	Media and VFP Feature 2
NZCV	32	0	rw	Condition Flags
OSDLR_EL1	32	0	rw	OS Double Lock
OSDTRRX_EL1	32	0	rw	OS Lock Data Transfer, Receive
OSDTRTX_EL1	32	0	rw	OS Lock Data Transfer, Transmit
OSECCR_EL1	32	0	rw	OS Lock Exception Catch Control
OSLAR_EL1	32	-	-w	OS Lock Access
OSLSR_EL1	32	a	r-	OS Lock Status
PAR_EL1	64	0	rw	Physical Address
PMCCFILTR_EL0	32	0	rw	Performance Monitors Cycle Count Filter
PMCCNTR_EL0	64	0	rw	Performance Monitors Cycle Count
PMCEID0_EL0	32	3fff0f3f	r-	Performance Monitors Common Event ID 0
PMCEID1_EL0	32	0	r-	Performance Monitors Common Event ID 1
PMCNTENCLR_EL0	32	0	rw	Performance Monitors Count Enable Clear
PMCNTENSET_EL0	32	0	rw	Performance Monitors Count Enable Set
PMCR_EL0	32	410f3000	rw	Performance Monitors Control
PMEVCNTR0_EL0	32	0	rw	Performance Monitors Event Count 0
PMEVCNTR1_EL0	32	0	rw	Performance Monitors Event Count 1
PMEVCNTR2_EL0	32	0	rw	Performance Monitors Event Count 2

PMEVCNTR3_EL0	32	0	rw	Performance Monitors Event Count 3
PMEVCNTR4_EL0	32	0	rw	Performance Monitors Event Count 4
PMEVCNTR5_EL0	32	0	rw	Performance Monitors Event Count 5
PMEVTYPER0_EL0	32	0	rw	Performance Monitors Event Type 0
PMEVTYPER1_EL0	32	0	rw	Performance Monitors Event Type 1
PMEVTYPER2_EL0	32	0	rw	Performance Monitors Event Type 2
PMEVTYPER3_EL0	32	0	rw	Performance Monitors Event Type 3
PMEVTYPER4_EL0	32	0	rw	Performance Monitors Event Type 4
PMEVTYPER5_EL0	32	0	rw	Performance Monitors Event Type 5
PMINTENCLR_EL1	32	0	rw	Performance Monitors Interrupt Enable Clear
PMINTENSET_EL1	32	0	rw	Performance Monitors Interrupt Enable Set
PMOVSCLR_EL0	32	0	rw	Performance Monitors Overflow Flag Status Clear
PMOVSSET_EL0	32	0	rw	Performance Monitors Overflow Flag Status Set
PMSELR_EL0	32	0	rw	Performance Monitors Event Counter Selection
PMSWINC_EL0	32	-	-w	Performance Monitors Software Increment
PMUSERENR_EL0	32	0	rw	Performance Monitors User Enable
PMXEVCNTR_EL0	32	0	rw	Performance Monitors Selected Event Count
PMXEVTYPER_EL0	32	0	rw	Performance Monitors Selected Event Type
REVIDR_EL1	32	0	r-	Revision ID
RVBAR_EL3	64	0	r-	Reset Vector Base Address (EL3)
SCR_EL3	32	30	rw	Secure Configuration
SCTLR_EL1	32	30c50838	rw	System Control Register (EL1)
SCTLR_EL2	32	30c50838	rw	System Control Register (EL2)
SCTLR_EL3	32	30c50838	rw	System Control (EL3)
SPSR_EL1	32	0	rw	Saved Program Status (EL1)
SPSR_EL2	32	0	rw	Saved Program Status (EL2)
SPSR_EL3	32	0	rw	Saved Program Status (EL3)
SPSR_abt	32	0	rw	Saved Program Status (Abort Mode)
SPSR_fiq	32	0	rw	Saved Program Status (FIQ Mode)
SPSR_irq	32	0	rw	Saved Program Status (IRQ Mode)
SPSR_und	32	0	rw	Saved Program Status (Undefined Mode)
SPSel	32	1	rw	Stack Pointer Select
SP_EL0	64	0	rw	Stack Pointer (EL0)

SP_EL1	64	0	rw	Stack Pointer (EL1)
SP_EL2	64	0	rw	Stack Pointer (EL2)
SP_EL3	64	0	rw	Stack Pointer (EL3)
TCR_EL1	64	0	rw	Translation Control (EL1)
TCR_EL2	32	80800000	rw	Translation Control (EL2)
TCR_EL3	32	80800000	rw	Translation Control (EL3)
TPIDRRO_EL0	64	0	rw	Thread Pointer/ID, Read-Only (EL0)
TPIDR_EL0	64	0	rw	Thread Pointer/ID (EL0)
TPIDR_EL1	64	0	rw	Thread Pointer/ID (EL1)
TPIDR_EL2	64	0	rw	Thread Pointer/ID (EL2)
TPIDR_EL3	64	0	rw	Thread Pointer/ID (EL3)
TTBR0_EL1	64	0	rw	Translation Table Base 0 (EL1)
TTBR0_EL2	64	0	rw	Translation Table Base 0 (EL2)
TTBR0_EL3	64	0	rw	Translation Table Base 0 (EL3)
TTBR1_EL1	64	0	rw	Translation Table Base 1
VBAR_EL1	64	0	rw	Vector Base Address (EL1)
VBAR_EL2	64	0	rw	Vector Base Address (EL2)
VBAR_EL3	64	0	rw	Vector Base Address (EL3)
VMPIDR_EL2	64	c0000000	rw	Virtualization Multiprocessor ID
VPIDR_EL2	32	410fd000	rw	Virtualization Processor ID
VTCR_EL2	32	80000000	rw	Virtualization Translation Control
VTTBR_EL2	64	0	rw	Virtualization Translation Table Base

12.1.4 AArch64_system_artifact

Table 12.

Name	**Bits**	**Initial value (Hex)**		**Description**
cpsr	32	3cd	rw	

12.1.5 Integration_support

Table 13.

Name	**Bits**	**Initial value (Hex)**		**Description**
transactPL	32	2	r-	privilege level of current memory transaction
transactAT	32	0	r-	current memory transaction type: PA=1, VA=0

#