FUJITSU-MONAKA™ specification

Ver. 1.0
Dec. 26, 2024

Fujitsu Limited

Copyright© 2024 Fujitsu Limited, 4-1-1 Kamikodanaka, Nakahara-ku, Kawasaki, 211-8588, Japan. All rights reserved.

This product and related documentation are protected by copyright and distributed under licenses restricting their use, copying, distribution, and decompilation. No part of this product or related documentation may be reproduced in any form by any means without prior written authorization of Fujitsu Limited and its licensors, if any.

The product(s) described in this book may be protected by one or more U.S. patents, foreign patents, or pending applications.

TRADEMARKS

Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

Fujitsu and the Fujitsu logo are trademarks of Fujitsu Limited.

This publication is provided "as is" without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or noninfringement.
This publication could include technical inaccuracies or typographical errors. Changes are periodically added to the information herein; these changes will be incorporated in new editions of the publication. Fujitsu Limited may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time.

Revision History

Date	Contents	Page
Dec. 26, 2024	Ver. 1.0 released	

Contents

Preface

This specification document is positioned as a supplement that explains the implementation-dependent and proprietary parts for the Arm architecture of the FUJITSU-MONAKA processor and does not provide a detailed explanation of the Arm architecture itself.

The specification related to the ARM architecture to which this specification refer is the one shown in the following table. Please refer to these documents as needed.

	Title	File Name	Rev.
[1]	Armv9-A arm	DDI0487K_a_a-profile_architecture_reference_manual	K.a
[2]	Exploration Tools	https://developer.arm.com/downloads/-/exploration-tools/feature-names-for-a-profile	
[3]	GICv3/v4	IHI0069H_gic_architecture_specification	H
[4]	MPAM	IHI0099A_a_MPAM_system_component_specification	A.a
[5]	SMMUv3	IHI0070F_a_System_Memory_Management_Unit_Architecture_Specification	F.a
[6]	A-profile registers	https://developer.arm.com/documentation/ddi0601/2024-06	2024-06

1. About FUJITSU-MONAKA processor

1.1. FUJITSU-MONAKA processor summary

The following shows the overview of FUJITSU-MONAKA. See also 1.2 Supported features for more details.

Core features

- Instruction set architecture:

  Armv9.3-A architecture and some of the features introduced in Armv9.4 and Armv9.5.
- Compliant with Arm Confidential Compute Architecture (Arm CCA).
  - Implementation of the FEAT_RME and the FEAT_MEC.
- Implementation of the SVE and the SVE2 with 256-bit vector length.
- 48-bit of Physical Address (PA) and 48-bit Virtual Address (VA) with mixed endian support.
- Support for 4 Kbytes and 64 Kbytes memory granule size at stage 1 and stage 2 address translation.
- Cache line length: 64 bytes.
- AArch64 execution state at EL0 to EL3.

SoC features

- Up to 144 single thread cores per socket.
- Up to 2 sockets configuration.
- SBSA 7.0+, BSA 1.0+ class support.
- Implementation of the GICv4.2 (See the Arm GIC architecture v3/v4 [3].).
- Support for the MPAMv1.1 with RME support (See the Arm MPAM architecture [4].).
- Implementation of the System MMU.
  - Conforms to the SMMUv3.3 (See the Arm SMMU architecture v3 [5].).
- Implementation of the PCIe gen6 controllers.
- Implementation of the DDR5 memory channels.
- Implementation of the Compute Express Link (CXL) 3.0 controllers.
  - Support for Type-3 Devices.
- Implementation of the on-chip system controllers.
  - Controlls the UART, GPIO, SMBUS, and SPI.
- Implementation of the co-packaged micro-controller subsystems.
  - Security subsystem.
  - Management subsystem.

Figure 1-1 shows the overview of the SoC System Architecture.

Note The contents of this section include information currently under developing and are subject to change in the future.

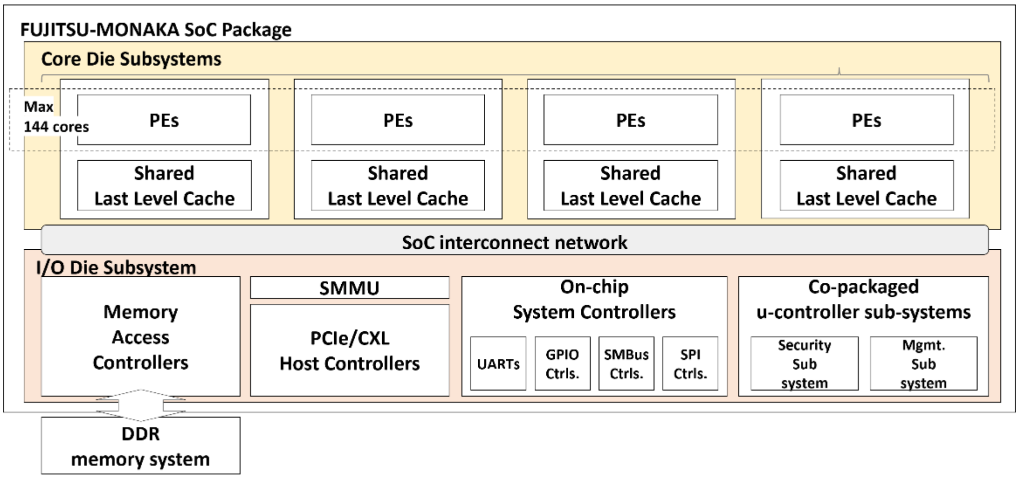


Figure 1-1 SoC System Architecture

1.2. Supported features

FUJITSU-MONAKA is compliant with Armv9.3-A architecture. It also supports some of the features introduced in Armv9.4 and Armv9.5. The following tables show the features that FUJITSU-MONAKA supports for each Armv8-A and Armv9-A architecture versions.

Note The contents of this section include information currently under developing and are subject to change in the future.

Table 1-1 Armv8.0 architecture extensions

Feature	Implementation	Note
FEAT_AA32EL0, Support for AArch32 at EL0	Not supported	See the Armv9-A arm[1].
FEAT_AA32EL1, Support for AArch32 at EL1	Not supported	See the Armv9-A arm[1].
FEAT_AA32EL2, Support for AArch32 at EL2	Not supported	See the Armv9-A arm[1].
FEAT_AA32EL2, Support for AArch32 at EL2	Not supported	See the Armv9-A arm[1].
FEAT_AA64EL0, Support for AArch64 at EL0	Supported	See the Armv9-A arm[1].
FEAT_AA64EL1, Support for AArch64 at EL1	Supported	See the Armv9-A arm[1].
FEAT_AA64EL2, Support for AArch64 at EL2	Supported	See the Armv9-A arm[1].
FEAT_AA64EL3, Support for AArch64 at EL3	Supported	See the Armv9-A arm[1].
FEAT_AES, Advanced SIMD AES instructions	Supported	See the Armv9-A arm[1].
FEAT_ASID16, 16 bit ASID	Supported	See the Armv9-A arm[1].
FEAT_AdvSIMD, Advanced SIMD Extension	Supported	See the Armv9-A arm[1].
FEAT_CRC32, CRC32 instructions	Supported	See the Armv9-A arm[1].
FEAT_CSV2_1p1, Cache Speculation Variant 2	Supported	See the Armv9-A arm[1].
FEAT_CSV2_1p2, Cache Speculation Variant 2 version 1.2	Supported	See the Armv9-A arm[1].
FEAT_CSV2_2, Cache Speculation Variant 2	Supported	See the Armv9-A arm[1].

version 2		
FEAT_CSV2_3, Cache Speculation Variant 2 version 3	Supported	See the Armv9-A arm[1].
FEAT_Crypto, Cryptographic Extension	Supported	See the Armv9-A arm[1].
FEAT_DoubleLock, Double Lock	Not supported	See the Armv9-A arm[1].
FEAT_EL0, Support for execution at EL0	Supported	See the Armv9-A arm[1].
FEAT_EL1, Support for execution at EL1	Supported	See the Armv9-A arm[1].
FEAT_EL2, Support for execution at EL2	Supported	See the Armv9-A arm[1].
FEAT_EL3, Support for EL3	Supported	See the Armv9-A arm[1].
FEAT_ETMv4, Embedded Trace Macrocell version 4	Not supported	See the Armv9-A arm[1].
FEAT_ETS2, Enhanced Translation Synchronization	Supported	See the Armv9-A arm[1].
FEAT_FP, Floating Point extensions	Supported	See the Armv9-A arm[1].
FEAT_IVIPT, The IVIPT Extension	Supported	See the Armv9-A arm[1].
FEAT_MixedEnd, Mixed-endian support	Supported	See the Armv9-A arm[1].
FEAT_MixedEndEL0, Mixed-endian support at EL0	Supported	See the Armv9-A arm[1].
FEAT_PCSRv8, PC Sample-based Profiling extension	Not supported	See the Armv9-A arm[1].
FEAT_PMULL, Advanced SIMD PMULL instructions	Supported	See the Armv9-A arm[1].
FEAT_PMUv3, PMU extension version 3	Supported	See the Armv9-A arm[1].
FEAT_PMUv3_EXT, External interface to the Performance Monitors	Not supported	See the Armv9-A arm[1].
FEAT_PMUv3_EXT32, 32-bit external interface to the Performance Monitors	Not supported	See the Armv9-A arm[1].
FEAT_SHA1, Advanced SIMD SHA1 instructions	Supported	See the Armv9-A arm[1].
FEAT_SHA256, Advanced SIMD SHA256 instructions	Supported	See the Armv9-A arm[1].
FEAT_SpecSEI, SError interrupt exceptions from speculative reads of memory	Supported	See the Armv9-A arm[1].
FEAT_TGran16K, Support for 16KB memory translation granule size at stage 1	Not supported	See the Armv9-A arm[1].
FEAT_TGran4K, Support for 4KB memory translation granule size at stage 1	Supported	See the Armv9-A arm[1].
FEAT_TGran64K, Support for 64KB memory translation granule size at stage 1	Supported	See the Armv9-A arm[1].
FEAT_TRC_EXT, Trace external registers	Not supported	See the Armv9-A arm[1].
FEAT_TRC_SR, Trace System registers	Supported	See the Armv9-A arm[1].
FEAT_nTLBPA, Intermediate caching of translation table walks	Supported	See the Armv9-A arm[1].

Table 1-2 Armv8.1 architecture extensions

Feature	Implementation	Note
FEAT_Debugv8p1, Debug with VHE	Supported	See the Armv9-A arm[1].
FEAT_HAFDBS, Hardware management of the Access flag and dirty state	Supported	See the Armv9-A arm[1].
FEAT_HPDS, Hierarchical permission disables in translations tables	Supported	See the Armv9-A arm[1].
FEAT_LOR, Limited ordering regions	Supported	See the Armv9-A arm[1].
FEAT_LSE, Large System Extensions	Supported	See the Armv9-A arm[1].
FEAT_PAN, Privileged access never	Supported	See the Armv9-A arm[1].
FEAT_PMUv3p1, Armv8.1 PMU extensions	Supported	See the Armv9-A arm[1].
FEAT_RDM, Advanced SIMD rounding double multiply accumulate instructions	Supported	See the Armv9-A arm[1].
FEAT_VHE, Virtualization Host Extensions	Supported	See the Armv9-A arm[1].
FEAT_VMID16, 16-bit VMID	Supported	See the Armv9-A arm[1].

Table 1-3 Armv8.2 architecture extensions

Feature	Implementation	Note
FEAT_AA32HPD, AArch32 hierarchical permission disables	Not supported	See the Armv9-A arm[1].
FEAT_AA32I8MM, AArch32 Int8 matrix multiplication instructions	Not supported	See the Armv9-A arm[1].
FEAT_ASMv8p2, Armv8.2 changes to the A64 ISA	Supported	See the Armv9-A arm[1].
FEAT_DPB, DC CVAP instruction	Supported	See the Armv9-A arm[1].
FEAT_Debugv8p2, Debug v8.2	Supported	See the Armv9-A arm[1].
FEAT_F32MM, Single-precision Matrix Multiplication	Not supported	See the Armv9-A arm[1].
FEAT_F64MM, Double-precision Matrix Multiplication	Not supported	See the Armv9-A arm[1].
FEAT_FP16, Half-precision floating-point data processing	Supported	See the Armv9-A arm[1].
FEAT_HPDS2, Hierarchical permission disables	Supported	See the Armv9-A arm[1].
FEAT_I8MM, AArch64 Int8 matrix multiplication instructions	Supported	See the Armv9-A arm[1].
FEAT_IESB, Implicit Error Synchronization event	Supported	See the Armv9-A arm[1].
FEAT_LPA, Large PA and IPA support	Not supported	See the Armv9-A arm[1].
FEAT_LSMAOC, AArch32 Load/Store Multiple instruction atomicity and ordering controls	Not supported	See the Armv9-A arm[1].
FEAT_LVA, Large VA support	Not supported	See the Armv9-A arm[1].
FEAT_PAN2, AT S1E1R and AT S1E1W instruction variants affected by PSTATE.PAN	Supported	See the Armv9-A arm[1].
FEAT_PCSRv8p2, PC Sample-based Profiling Extension	Not supported	See the Armv9-A arm[1].
FEAT_RAS, Reliability, Availability and Serviceability (RAS) Extension	Supported	See the Armv9-A arm[1].
FEAT_RASSAv1, RAS System Architecture version 1	Supported	See the Armv9-A arm[1].
FEAT_SHA3, Advanced SIMD SHA3 instructions	Supported	See the Armv9-A arm[1].
FEAT_SHA512, Advanced SIMD SHA512 instructions	Supported	See the Armv9-A arm[1].
FEAT_SM3, Advanced SIMD SM3 instructions	Supported	See the Armv9-A arm[1].
FEAT_SM4, Advanced SIMD SM4 instructions	Supported	See the Armv9-A arm[1].
FEAT_SPE, Statistical Profiling Extension	Not supported	See the Armv9-A arm[1].
FEAT_SVE, Scalable Vector Extension	Supported	See the Armv9-A arm[1].
FEAT_TTCNP, Translation table Common not private translations	Supported	See the Armv9-A arm[1].
FEAT_UAO, Unprivileged Access Override control	Supported	See the Armv9-A arm[1].
FEAT_XNX, Translation table stage 2 Unprivileged Execute-never	Supported	See the Armv9-A arm[1].

Table 1-4 Armv8.3 architecture extensions

Feature	Implementation	Note
FEAT_CCIDX, Extended cache index	Supported	See the Armv9-A arm[1].
FEAT_CONSTPACFIELD, PAC algorithm enhancement	Supported	See the Armv9-A arm[1].
FEAT_DoPD, Debug over Powerdown	Supported	See the Armv9-A arm[1].
FEAT_EPAC, Enhanced pointer authentication	Not supported	See the Armv9-A arm[1].
FEAT_FCMA, Floating-point complex number instructions	Supported	See the Armv9-A arm[1].
FEAT_FPAC, Faulting on AUT* instructions	Supported	See the Armv9-A arm[1].
FEAT_FPACCOMBINE, Faulting on combined pointer authentication instructions	Supported	See the Armv9-A arm[1].

FEAT_FPACC_SPEC, Faulting on combined pointer authentication instructions	Supported	See the Armv9-A arm[1].
FEAT_JSCVT, JavaScript conversion instructions	Supported	See the Armv9-A arm[1].
FEAT_LRCPC, Load-acquire RCpc instructions	Supported	See the Armv9-A arm[1].
FEAT_NV, Nested virtualization support	Supported	See the Armv9-A arm[1].
FEAT_PACIMP, Pointer authentication - IMPLEMENTATION DEFINED algorithm	Not supported	See the Armv9-A arm[1].
FEAT_PACQARMA3, Pointer authentication - QARMA3 algorithm	Supported	See the Armv9-A arm[1].
FEAT_PACQARMA5, Pointer authentication - QARMA5 algorithm	Not supported	See the Armv9-A arm[1].
FEAT_PAuth, Pointer authentication	Supported	See the Armv9-A arm[1].
FEAT_SPEv1p1, Statistical Profiling Extension version 1	Not supported	See the Armv9-A arm[1].

Table 1-5 Armv8.4 architecture extensions

Feature	Implementation	Note
FEAT_AMU_EXT, External Activity Monitors	Not supported	See the Armv9-A arm[1].
FEAT_AMU_EXT32, AArch32 External Activity Monitors	Not supported	See the Armv9-A arm[1].
FEAT_AMUv1, Activity Monitors Extension version 1	Supported	See the Armv9-A arm[1].
FEAT_BBM, Translation table break-before-make levels	Supported, Level 2	See the Armv9-A arm[1].
FEAT_CNTSC, Generic Counter Scaling	Not supported	See the Armv9-A arm[1].
FEAT_DIT, Data Independent Timing instructions	Supported	See the Armv9-A arm[1].
FEAT_Debugv8p4, Debug v8.4	Supported	See the Armv9-A arm[1].
FEAT_DotProd, Advanced SIMD dot product instructions	Supported	See the Armv9-A arm[1].
FEAT_DoubleFault, Double Fault Extension	Supported	See the Armv9-A arm[1].
FEAT_FHM, Floating-point half-precision to single-precision multiply-add instructions	Supported	See the Armv9-A arm[1].
FEAT_FlagM, Condition flag manipulation instructions	Supported	See the Armv9-A arm[1].
FEAT_IDST, ID space trap handling	Supported	See the Armv9-A arm[1].
FEAT_LRCPC2, Load-acquire RCpc instructions version 2	Supported	See the Armv9-A arm[1].
FEAT_LSE2, Large System Extensions version 2	Supported	See the Armv9-A arm[1].
FEAT_MPAM, Memory Partitioning and Monitoring Extension	Supported	See the Armv9-A arm[1].
FEAT_NV2, Enhanced nested virtualization support	Supported	See the Armv9-A arm[1].
FEAT_PMUv3p4, Arm8.4 PMU extensions	Supported	See the Armv9-A arm[1].
FEAT_RASSAv1p1, RAS version 1.1 System Architecture	Supported	See the Armv9-A arm[1].
FEAT_RASv1p1, RAS extension v1.1	Supported	See the Armv9-A arm[1].
FEAT_S2FWB, Stage 2 forced Write-Back	Supported	See the Armv9-A arm[1].
FEAT_SEL2, Secure EL2	Supported	See the Armv9-A arm[1].
FEAT_TLBIOS, TLB invalidate instructions in Outer Shareable domain	Supported	See the Armv9-A arm[1].
FEAT_TLBIRANGE, TLB invalidate range instructions	Supported	See the Armv9-A arm[1].
FEAT_TRF, Self-hosted Trace Extensions	Supported	See the Armv9-A arm[1].
FEAT_TTL, Translation Table Level	Supported	See the Armv9-A arm[1].
FEAT_TTST, Small translation tables	Supported	See the Armv9-A arm[1].

Table 1-6 Armv8.5 architecture extensions

Feature	Implementation	Note

FEAT_BTI, Branch Target Identification	Supported	See the Armv9-A arm[1].
FEAT_CSV2, Cache Speculation Variant 2	Supported	See the Armv9-A arm[1].
FEAT_CSV3, Cache Speculation Variant 3	Supported	See the Armv9-A arm[1].
FEAT_DPB2, DC CVADP instruction	Supported	See the Armv9-A arm[1].
FEAT_E0PD, Preventing EL0 access to halves of address maps	Supported	See the Armv9-A arm[1].
FEAT_EVT, Enhanced Virtualization Traps	Supported	See the Armv9-A arm[1].
FEAT_ExS, Context synchronization and exception handling	Supported	See the Armv9-A arm[1].
FEAT_FRINTTS, Floating-point to integer instructions	Supported	See the Armv9-A arm[1].
FEAT_FlagM2, Enhancements to flag manipulation instructions	Supported	See the Armv9-A arm[1].
FEAT_GTG, Guest translation granule size	Supported	See the Armv9-A arm[1].
FEAT_MTE, Memory Tagging Extension	Not supported	See the Armv9-A arm[1].
FEAT_MTE2, Memory Tagging Extension	Not supported	See the Armv9-A arm[1].
FEAT_PMUv3p5, Arm8.5 PMU extensions	Supported	See the Armv9-A arm[1].
FEAT_RNG, Random number generator	Supported	See the Armv9-A arm[1].
FEAT_RNG_TRAP, Trapping support for RNDR/RNDRRS	Supported	See the Armv9-A arm[1].
FEAT_S2TGran16K, Support for 16KB memory translation granule size at stage 2	Not supported	See the Armv9-A arm[1].
FEAT_S2TGran4K, Support for 4KB memory translation granule size at stage 2	Supported	See the Armv9-A arm[1].
FEAT_S2TGran64K, Support for 64KB memory translation granule size at stage 2	Supported	See the Armv9-A arm[1].
FEAT_SB, Speculation Barrier	Supported	See the Armv9-A arm[1].
FEAT_SPECRES, Speculation restriction instructions	Supported	See the Armv9-A arm[1].
FEAT_SSBS, Speculative Store Bypass Safe	Supported	See the Armv9-A arm[1].
FEAT_SSBS2, MRS and MSR instructions for SSBS version 2	Supported	See the Armv9-A arm[1].

Table 1-7 Armv8.6 architecture extensions

Feature	Implementation	Note
FEAT_AA32BF16, AArch32 BFloat16 instructions	Not supported	See the Armv9-A arm[1].
FEAT_AMUv1p1, Activity Monitors Extension version 1.1	Supported	See the Armv9-A arm[1].
FEAT_BF16, AArch64 BFloat16 instructions	Supported	See the Armv9-A arm[1].
FEAT_CP15SDISABLE2, CP15SDISABLE2	Not supported	See the Armv9-A arm[1].
FEAT_DGH, Data Gathering Hint	Not supported	See the Armv9-A arm[1].
FEAT_ECV, Enhanced Counter Virtualization	Supported	See the Armv9-A arm[1].
FEAT_FGT, Fine Grain Traps	Supported	
FEAT_HPMN0, Setting of MDCR_EL2.HPMN to zero	Supported	See the Armv9-A arm[1].
FEAT_MPAMv0p1, Memory Partitioning and Monitoring version 0.1	Not supported	See the Armv9-A arm[1].
FEAT_MPAMv1p1, Memory Partitioning and Monitoring version 1.1	Supported	See the Armv9-A arm[1].
FEAT_MTPMU, Multi-threaded PMU extensions	Not supported	See the Armv9-A arm[1].
FEAT_PAuth2, Enhancements to pointer authentication	Supported	See the Armv9-A arm[1].
FEAT_TWED, Delayed Trapping of WFE	Supported	See the Armv9-A arm[1].

Table 1-8 Armv8.7 architecture extensions

Feature	Implementation	Note
FEAT_AFP, Alternate floating-point behavior	Supported	See the Armv9-A arm[1].
FEAT_EBF16, AArch64 Extended BFloat16	Not supported	See the Armv9-A arm[1].

instructions		
FEAT_HCX, Support for the HCRX_EL2 register	Supported	See the Armv9-A arm[1].
FEAT_LPA2, Larger physical address for 4KB and 16KB translation granules	Not supported	See the Armv9-A arm[1].
FEAT_LS64, Support for 64-byte loads and stores without status	Supported	See the Armv9-A arm[1].
FEAT_LS64_ACCDATA, Support for 64-byte EL0 stores with status	Supported	See the Armv9-A arm[1].
FEAT_LS64_V, Support for 64-byte stores with status	Supported	See the Armv9-A arm[1].
FEAT_MTE3, MTE Asymmetric Fault Handling	Not supported	See the Armv9-A arm[1].
FEAT_MTE_ASYM_FAULT, Memory tagging asymmetric faults	Not supported	See the Armv9-A arm[1].
FEAT_PAN3, Support for SCTLR_ELx.EPAN	Supported	See the Armv9-A arm[1].
FEAT_PMUv3p7, Armv8.7 PMU extensions	Supported	See the Armv9-A arm[1].
FEAT_RPRES, Increased precision of FRECPE and FRSQRTE	Not supported	See the Armv9-A arm[1].
FEAT_SPEv1p2, Statistical Profiling Extensions version 1.2	Not supported	See the Armv9-A arm[1].
FEAT_WFxT, WFE and WFI instructions with timeout	Supported	See the Armv9-A arm[1].
FEAT_XS, XS attribute	Supported	See the Armv9-A arm[1].

Table 1-9 Armv8.8 architecture extensions

Feature	Implementation	Note
FEAT_CMOW, Control for cache maintenance permission	Supported	See the Armv9-A arm[1].
FEAT_Debugv8p8, Debug v8.8	Supported	See the Armv9-A arm[1].
FEAT_HBC, Hinted conditional branches	Supported	See the Armv9-A arm[1].
FEAT_MOPS, Standardization of memory operations	Supported	See the Armv9-A arm[1].
FEAT_NMI, Non-maskable Interrupts	Supported	See the Armv9-A arm[1].
FEAT_PMUv3_EXT64, 64-bit external interface to the Performance Monitors	Not supported	See the Armv9-A arm[1].
FEAT_PMUv3_TH, Event counting threshold	Supported	See the Armv9-A arm[1].
FEAT_PMUv3p8, Armv8.8 PMU extensions	Supported	See the Armv9-A arm[1].
FEAT_SCTLR2, Extension to SCTLR_ELx	Supported	See the Armv9-A arm[1].
FEAT_SPEv1p3, Statistical Profiling Extensions version 1.3	Not supported	See the Armv9-A arm[1].
FEAT_TCR2, Support for TCR2_ELx	Supported	See the Armv9-A arm[1].
FEAT_TIDCP1, EL0 use of IMPLEMENTATION DEFINED functionality	Supported	See the Armv9-A arm[1].

Table 1-10 Armv8.9 architecture extensions

Feature	Implementation	Note
FEAT_ADERR, Asynchronous Device Error Exceptions	Not supported	See the Armv9-A arm[1].
FEAT_AIE, Memory Attribute Index Enhancement	Not supported	See the Armv9-A arm[1].
FEAT_AMU_EXT64, the 64-bit external Activity Monitors extension	Not supported	See the Armv9-A arm[1].
FEAT_ANERR, Asynchronous Normal Error Exceptions	Not supported	See the Armv9-A arm[1].
FEAT_ATS1A, Address Translation operations that ignore stage 1 permissions	Not supported	See the Armv9-A arm[1].
FEAT_CLRBHB, Support for Clear Branch History instruction	Supported	See the Armv9-A arm[1].
FEAT_CSSC, Common Short Sequence Compression instructions	Not supported	See the Armv9-A arm[1].
FEAT_Debugv8p9, Debug v8.9	Not supported	See the Armv9-A arm[1].

FEAT_DoubleFault2, Double Fault Extension v2	Not supported	See the Armv9-A arm[1].
FEAT_ECBHB, Exploitative control using branch history information	Supported	See the Armv9-A arm[1].
FEAT_EDHSR, Support for EDHSR	Not supported	See the Armv9-A arm[1].
FEAT_FGT2, Fine-grained traps 2	Not supported	See the Armv9-A arm[1].
FEAT_HAFT, Hardware managed Access Flag for Table descriptors	Not supported	See the Armv9-A arm[1].
FEAT_LRCPC3, Load-Acquire RCpc instructions version 3	Not supported	See the Armv9-A arm[1].
FEAT_MTE4, Enhanced Memory Tagging Extension	Not supported	See the Armv9-A arm[1].
FEAT_MTE_ASYNC, Asynchronous reporting of Tag Check Fault	Not supported	See the Armv9-A arm[1].
FEAT_MTE_CANONICAL_TAGS, Canonical Tag checking for Untagged memory	Not supported	See the Armv9-A arm[1].
FEAT_MTE_NO_ADDRESS_TAGS, Memory tagging with Address tagging disabled	Not supported	See the Armv9-A arm[1].
FEAT_MTE_PERM, Allocation tag access permission	Not supported	See the Armv9-A arm[1].
FEAT_MTE_STORE_ONLY, Store-only Tag Checking	Not supported	See the Armv9-A arm[1].
FEAT_MTE_TAGGED_FAR, FAR_ELx on a Tag Check Fault	Not supported	See the Armv9-A arm[1].
FEAT_PCSRv8p9, Armv8.9 PC Sample-based Profiling Extension	Not supported	See the Armv9-A arm[1].
FEAT_PFAR, Physical Fault Address Register Extension	Not supported	See the Armv9-A arm[1].
FEAT_PMUv3_EDGE, PMU event edge detection	Not supported	See the Armv9-A arm[1].
FEAT_PMUv3_ICNTR, Fixed-function instruction counter	Not supported	See the Armv9-A arm[1].
FEAT_PMUv3_SS, PMU Snapshot extension	Not supported	See the Armv9-A arm[1].
FEAT_PMUv3p9, Armv8.9 PMU extensions	Not supported	See the Armv9-A arm[1].
FEAT_PRFMSLC, SLC target support for PRFM instructions	Not supported	See the Armv9-A arm[1].
FEAT_RASSAv2, RAS System Architecture Extension v2	Not supported	See the Armv9-A arm[1].
FEAT_RASv2, RAS Extension v2	Not supported	See the Armv9-A arm[1].
FEAT_RPRFM, Support for Range Prefetch Memory instruction	Not supported	See the Armv9-A arm[1].
FEAT_S1PIE, Stage 1 permission indirections	Not supported	See the Armv9-A arm[1].
FEAT_S1POE, Stage 1 permission overlays	Not supported	See the Armv9-A arm[1].
FEAT_S2PIE, Stage 2 permission indirections	Not supported	See the Armv9-A arm[1].
FEAT_S2POE, Stage 1 permission overlays	Not supported	See the Armv9-A arm[1].
FEAT_SPECRES2, Enhanced speculation restriction instructions	Supported	See the Armv9-A arm[1].
FEAT_SPE_CRR, Call Return Branch Records	Not supported	See the Armv9-A arm[1].
FEAT_SPE_DPFZS, Disable Cycle Counter on SPE Freeze	Not supported	See the Armv9-A arm[1].
FEAT_SPE_FDS, Data Source Filtering	Not supported	See the Armv9-A arm[1].
FEAT_SPEv1p4, Statistical Profiling Extension version 1.4	Not supported	See the Armv9-A arm[1].
FEAT_SPMU, System Performance Monitors Extension	Not supported	See the Armv9-A arm[1].
FEAT_THE, Translation Hardening Extension	Not supported	See the Armv9-A arm[1].

Table 1−11 Armv9.0 architecture extensions

Feature	Implementation	Note
FEAT_Armv9_Crypto, Armv9 Cryptographic Extension	Supported	See the Armv9-A arm[1].
FEAT_ETE, Embedded Trace Extension	Supported	See the Armv9-A arm[1].
FEAT_SVE2, Scalable Vector Extension version 2	Supported	See the Armv9-A arm[1].
FEAT_SVE_AES, Scalable Vector AES instructions	Supported	See the Armv9-A arm[1].
FEAT_SVE_BitPerm, Scalable Vector Bit Permutes instructions	Supported	See the Armv9-A arm[1].
FEAT_SVE_PMULL128, Scalable Vector PMULL instructions	Supported	See the Armv9-A arm[1].
FEAT_SVE_SHA3, Scalable Vector SHA3 instructions	Supported	See the Armv9-A arm[1].
FEAT_SVE_SM4, Scalable Vector SM4 instructions	Supported	See the Armv9-A arm[1].
FEAT_TME, Transactional Memory Extension	Not supported	See the Armv9-A arm[1].
FEAT_TRBE, Trace Buffer Extension	Supported	See the Armv9-A arm[1].

Table 1−12 Armv9.1 architecture extensions

Feature	Implementation	Note
FEAT_ETEv1p1, Embedded Trace Extension	Supported	See the Armv9-A arm[1].

Table 1−13 Armv9.2 architecture extensions

Feature	Implementation	Note
FEAT_BRBE, Branch Record Buffer Extension	Not supported	See the Armv9-A arm[1].
FEAT_ETEv1p2, Embedded Trace Extension	Supported	See the Armv9-A arm[1].
FEAT_RME, Realm Management Extension	Supported	See the Armv9-A arm[1].
FEAT_SME, Scalable Matrix Extension	Not supported	See the Armv9-A arm[1].
FEAT_SME_F64F64, Double-precision floating-point outer product instructions	Not supported	See the Armv9-A arm[1].
FEAT_SME_FA64, Full A64 instruction set support in Streaming SVE mode	Not supported	See the Armv9-A arm[1].
FEAT_SME_I16I64, 16-bit to 64-bit integer widening outer product instructions	Not supported	See the Armv9-A arm[1].

Table 1−14 Armv9.3 architecture extensions

Feature	Implementation	Note
FEAT_BRBEv1p1, Branch Record Buffer Extension version 1.1	Not supported	See the Armv9-A arm[1].
FEAT_MEC, Memory Encryption Contexts	Supported	See the Armv9-A arm[1].
FEAT_SME2, Scalable Matrix Extensions version 2	Not supported	See the Armv9-A arm[1].

Table 1−15 Armv9.4 architecture extensions

Feature	Implementation	Note
FEAT_ABLE, Address Breakpoint Linking Extension	Not supported	See the Armv9-A arm[1].
FEAT_BWE, Breakpoint and watchpoint enhancements	Not supported	See the Armv9-A arm[1].

FEAT_CHK, Check Feature Status	Not supported	See the Armv9-A arm[1].
FEAT_D128, 128-bit Translation Tables, 56 bit PA	Not supported	See the Armv9-A arm[1].
FEAT_EBEP, Exception-based Event Profiling	Not supported	See the Armv9-A arm[1].
FEAT_ETEv1p3, Embedded Trace Extension version 1.3	Not supported	See the Armv9-A arm[1].
FEAT_GCS, Guarded Control Stack Extension	Not supported	See the Armv9-A arm[1].
FEAT_ITE, Instrumentation Trace Extension	Not supported	See the Armv9-A arm[1].
FEAT_LSE128, 128-bit Atomics	Not supported	See the Armv9-A arm[1].
FEAT_LVA3, 56-bit VA	Not supported	See the Armv9-A arm[1].
FEAT_SEBEP, Synchronous Exception-based Event Profiling	Not supported	See the Armv9-A arm[1].
FEAT_SME2p1, Scalable Matrix Extension version 2.1	Not supported	See the Armv9-A arm[1].
FEAT_SME_F16F16, Non-widening half-precision FP16 to FP16 arithmetic for SME2	Not supported	See the Armv9-A arm[1].
FEAT_SVE2p1, Scalable Vector Extensions version 2.1	Not supported	See the Armv9-A arm[1].
FEAT_SVE_B16B16, Non-widening BFloat16 to BFloat16 arithmetic for SVE2 and SME2	Not supported	See the Armv9-A arm[1].
FEAT_SYSINSTR128, 128-bit System instructions	Not supported	See the Armv9-A arm[1].
FEAT_SYSREG128, 128-bit System registers	Not supported	See the Armv9-A arm[1].
FEAT_TRBE_EXT, Trace Buffer external mode	Not supported	See the Armv9-A arm[1].
FEAT_TRBE_MPAM, Trace Buffer MPAM extensions	Not supported	See the Armv9-A arm[1].

Table 1-16 Armv9.5 architecture extensions

Feature	Implementation	Note
FEAT_ASID2, Support for concurrent use of two ASIDs	Not supported	See the Exploration Tools[2].
FEAT_BWE2, Breakpoint and watchpoint enhancements 2	Not supported	See the Exploration Tools[2].
FEAT_CPA, Instruction-only Checked Pointer Arithmetic	Not supported	See the Exploration Tools[2].
FEAT_CPA2, Checked Pointer Arithmetic	Not supported	See the Exploration Tools[2].
FEAT_E2H0, Programming of HCR_EL2.E2H	Not supported	See the Exploration Tools[2].
FEAT_E3DSE, Delegated SError exception injection	Not supported	See the Exploration Tools[2].
FEAT_FAMINMAX, Floating-point maximum and minimum absolute value instructions	Supported	See the Exploration Tools[2].
FEAT_FGWTE3, Fine-Grained Write Trap EL3	Not supported	See the Exploration Tools[2].
FEAT_FP8, FP8 convert instructions	Supported	See the Exploration Tools[2].
FEAT_FP8DOT2	Supported	See the Exploration Tools[2].
FEAT_SSVE_FP8DOT2	Not supported	See the Exploration Tools[2].
FEAT_FP8DOT4	Supported	See the Exploration Tools[2].
FEAT_SSVE_FP8DOT4	Not supported	See the Exploration Tools[2].
FEAT_FP8FMA	Supported	See the Exploration Tools[2].
FEAT_SSVE_FP8FMA	Not supported	See the Exploration Tools[2].
FEAT_FPMR, Floating-point Mode Register	Supported	See the Exploration Tools[2].
FEAT_HACDBS, Hardware accelerator for cleaning Dirty state	Not supported	See the Exploration Tools[2].
FEAT_HDBSS, Hardware Dirty state tracking Structure	Not supported	See the Exploration Tools[2].
FEAT_LUT, Lookup table instructions with 2-bit and 4-bit indices	Supported	See the Exploration Tools[2].
FEAT_LUTv2	Not supported	See the Exploration Tools[2].
FEAT_PAuth_LR	Not supported	See the Exploration Tools[2].
FEAT_PMUv3_SME	Not supported	See the Exploration Tools[2].

FEAT_PMUv3_TH2	Not supported	See the Exploration Tools[2].
FEAT_RME_GPC2	Not supported	See the Exploration Tools[2].
FEAT_SME_F8F16	Not supported	See the Exploration Tools[2].
FEAT_SME_F8F32	Not supported	See the Exploration Tools[2].
FEAT_RPZ	Not supported	See the Exploration Tools[2].
FEAT_SPE_ALTCLK	Not supported	See the Exploration Tools[2].
FEAT_SPE_EFT	Not supported	See the Exploration Tools[2].
FEAT_SPE_FPF	Not supported	See the Exploration Tools[2].
FEAT_SPE_SME	Not supported	See the Exploration Tools[2].
FEAT_STEP2	Not supported	See the Exploration Tools[2].
FEAT_TLBIW	Not supported	See the Exploration Tools[2].

2. Identification registers description

This section describes the identification registers that are implemented in the FUJITSU-MONAKA processor.

Refer to the Arm® Architecture Reference Manual for A-profile architecture[1].

2.1. Main ID Register EL1 (MIDR_EL1)

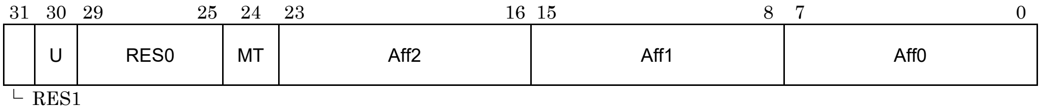
63 32
RES0

31 24	23 20	19 16	15 4	3 0
Implementer	Variant	Architecture	PartNum	Revision

Bits	Name	Value	Function
[63:32]	-	0x00000000	Reserved, RES0
[31:24]	Implementer	0x46	Indicate the implementer code. 0x46 Fujitsu Limited
[23:20]	Variant	0x*	Variant number. Indicate the major revision of FUJITSU-MONAKA processor core die.
[19:16]	Architecture	0xF	Architectural features are individually identified in the ID_* registers.
[15:4]	Primary part number	0x003	Indicate primary part number. 0x003 FUJITSU-MONAKA processor
[3:0]	Revision	0x*	Revision number. Indicate the minor revision of FUJITSU-MONAKA processor core die.

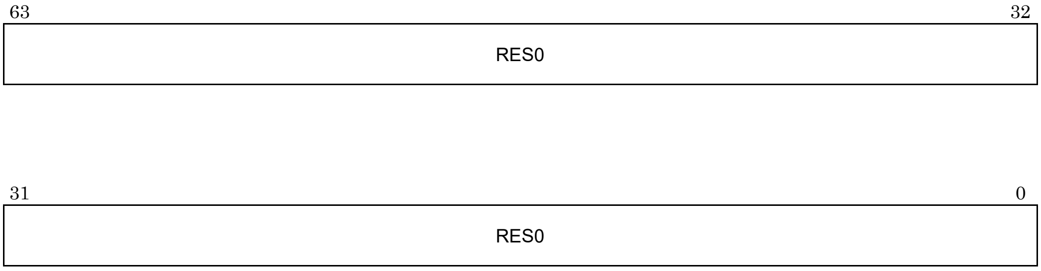
2.2. Multiprocessor Affinity Register EL1 (MPIDR_EL1)

63 40	39 32
RES0	Aff3



Bits	Name	Value	Function
[63:40]	-	`0x000000`	Reserved, RES0
[39:32]	Aff3	`0x00`	Affinity level 3.
[31]	-	`0x1`	RES1
[30]	U	`0x0`	All PEs are part of a multiprocessor system.
[29:25]	-	`0x0`	Reserved, RES0
[24]	MT	`0x0`	The multithreading mechanism is not implementing.
[23:19]	Aff2[7:3]	`0x0`	Reserved, RES0
[18]	Aff2[2]		Indicate the socket number of the logical PE.
[17:16]	Aff2[1:0]		Indicate the COD number of the logical PE.
[15:14]	Aff1[7:6]	`0x0`	Reserved, RES0
[13:8]	Aff1[5:0]		Indicate the physical core number of the logical PE.
[7:0]	Aff0[7:0]	`0x0`	Reserved, RES0

2.3. Revision ID Register EL1 (REVIDR_EL1)



Bits	Name	Value	Function
[63:0]	-	`0x00000000_00000000`	Reserved, RES0

2.4. AArch64 Processor Feature Register 0 EL1 (ID_AA64PFR0_EL1)

63 60	59 56	55 52	51 48	47 44	43 40	39 36	35 32
CSV3	CSV2	RME	DIT	AMU	MPAM	SEL2	SVE

31 28	27 24	23 20	19 16	15 12	11 8	7 4	3 0
RAS	GIC	AdvSIMD	FP	EL3	EL2	EL1	EL0

Bits	**Name**	**Value**	**Function**
[63:60]	CSV3	0x1	FEAT_CSV3 is implemented.
[59:56]	CSV2	0x3	FEAT_CSV2_3 is implemented.
[55:52]	RME	0x1	FEAT_RME is implemented.
[51:48]	DIT	0x1	FEAT_DIT is implemented.
[47:44]	AMU	0x2	FEAT_AMUv1p1 is implemented.
[43:40]	MPAM	0x1	FEAT_MPAMv1p1 is implemented.
[39:36]	SEL2	0x1	FEAT_SEL2 is implemented.
[35:32]	SVE	0x1	FEAT_SVE is implemented.
[31:28]	RAS	0x2	FEAT_RASv1p1 and FEAT_DoubleFault are implemented.
[27:24]	GIC	0x3	System register interface to version 4.1 of the GIC CPU interface is supported.
[23:20]	AdvSIMD	0x1	Advanced SIMD is implemented, including support for half-precision floating-point arithmetic.
[19:16]	FP	0x1	Floating-point is implemented, including support for half-precision floating-point arithmetic.
[15:12]	EL3	0x1	EL3 can be executed in AArch64 state only.
[11:8]	EL2	0x1	EL2 can be executed in AArch64 state only.
[7:4]	EL1	0x1	EL1 can be executed in AArch64 state only.
[3:0]	EL0	0x1	EL0 can be executed in AArch64 state only.

2.5. AArch64 Processor Feature Register 1 EL1 (ID_AA64PFR1_EL1)

63 60	59 56	55 52	51 48	47 44	43 40	39 36	35 32
PFAR	DF2	MTEX	THE	GCS	MTE_frac	NMI	CSV2_frac

31 28	27 24	23 20	19 16	15 12	11 8	7 4	3 0
RNDR_trap	SME	RES0	MPAM_frac	RAS_frac	MTE	SSBS	BT

Bits	**Name**	**Value**	**Function**
[63:60]	PFAR	0x0	FEAT_PFAR is not implemented.
[59:56]	DF2	0x0	FEAT_DoubleFault2 is not implemented.
[55:52]	MTEX	0x0	Memory Tagging Extension is not implemented.
[51:48]	THE	0x0	Translation Hardening Extension is not implemented.
[47:44]	GCS	0x0	Guarded Control Stack is not implemented.
[43:40]	MTE_frac	0x0	Memory Tagging Extension is not implemented.
[39:36]	NMI	0x1	FEAT_NMI is implemented.
[35:32]	CSV2_frac	0x0	FEAT_CSV2_3 is implemented, therefore FEAT_CSV2_1p2 is not implemented.
[31:28]	RNDR_trap	0x1	FEAT_RNG_TRAP is implemented.
[27:24]	SME	0x0	Scalable Matrix Extension is not implemented.
[23:20]	-	0x0	Reserved, RES0
[19:16]	MPAM_frac	0x1	FEAT_MPAMv1p1 is implemented.
[15:12]	RAS_frac	0x0	FEAT_RASv1p1 and FEAT_DoubleFault are implemented.
[11:8]	MTE	0x0	Memory Tagging Extension is not implemented.
[7:4]	SSBS	0x2	FEAT_SSBS and FEAT_SSBS2 are implemented.
[3:0]	BT	0x1	FEAT_BTI is implemented.

2.6. AArch64 Processor Feature Register 2 EL1 (ID_AA64PFR2_EL1)

63 36	35 32
RES0	FPMR

31 12	11 8	7 4	3 0
RES0	MTEFAR	MTESTOREONLY	MTEPERM

Bits	Name	Value	Function
[63:36]	-	0x0000000	Reserved, RES0
[35:32]	FPMR	0x1	FEAT_FPMR is implemented. See Arm A-profile Architecture Registers [6].
[31:12]	-	0x00000	Reserved, RES0
[11:8]	MTEFAR	0x0	Memory Tagging Extension is not implemented.
[7:4]	MTESTOREONLY	0x0	Memory Tagging Extension is not implemented.
[3:0]	MTEPERM	0x0	Memory Tagging Extension is not implemented.

2.7. AArch64 Debug Feature Register 0 EL1 (ID_AA64DFR0_EL1)

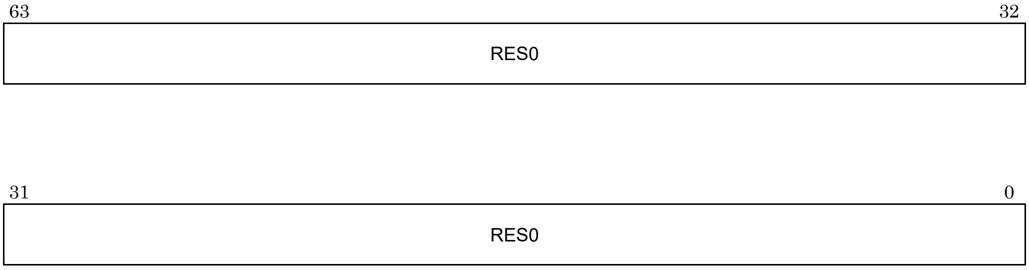
63 60	59 56	55 52	51 48	47 44	43 40	39 36	35 32
HPMN0	ExtTrcBuff	BRBE	MTPMU	TraceBuffer	TraceFilt	DoubleLock	PMSVer

31 28	27 24	23 20	19 16	15 12	11 8	7 4	3 0
CTX_CMPs	SEBEP	WRPs	PMSS	BRPs	PMUVer	TraceVer	DebugVer

Bits	Name	Value	Function
[63:60]	HPMN0	0x1	FEAT_HPMN0 is implemented.
[59:56]	ExtTrcBuff	0x0	FEAT_TRBE_EXT is not implemented.
[55:52]	BRBE	0x0	Branch Record Buffer Extension is not implemented.
[51:48]	MTPMU	0xf	FEAT_MTPMU not implemented. PMEVTYPER<n>_EL0.MT and PMEVTYPER<n>.MT are RES0.
[47:44]	TraceBuffer	0x1	FEAT_TRBE is implemented.
[43:40]	TraceFilt	0x1	FEAT_TRF is implemented.
[39:36]	DoubleLock	0xf	FEAT_DoubleLock is not implemented.
[35:32]	PMSVer	0x0	Statistical Profiling Extension is not implemented.
[31:28]	CTX_CMPs	0x1	Two context-aware breakpoints are implemented.
[27:24]	SEBEP	0x0	FEAT_SEBEP is not implemented.
[23:20]	WRPs	0x3	Four watchpoints are implemented.
[19:16]	PMSS	0x0	FEAT_PMUv3_SS is not implemented.
[15:12]	BRPs	0x5	Six breakpoints are implemented.
[11:8]	PMUVer	0x8	FEAT_PMUv3p8 is implemented.

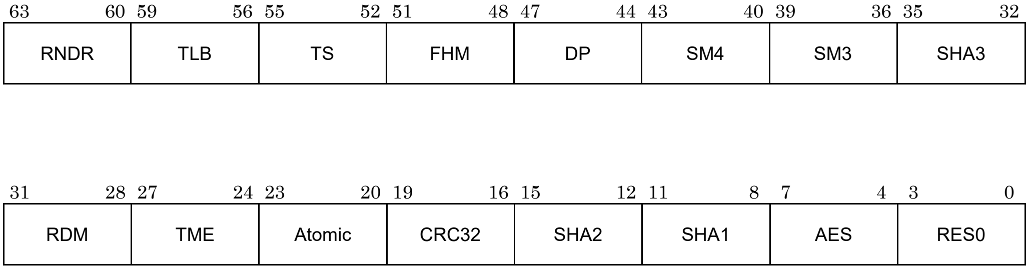
[7:4]	TraceVer	0x1	Trace unit System registers is implemented.
[3:0]	DebugVer	0xa	FEAT_Debugv8p8 is implemented.

2.8. AArch64 Debug Feature Register 1, EL1 (ID_AA64DFR1_EL1)



Bits	Name	Value	Function
[63:0]	-	0x00000000_00000000	Reserved, RES0

2.9. AArch64 Instruction Set Attribute Register 0, EL1 (ID_AA64ISAR0_EL1)



Bits	Name	Value	Function
[63:60]	RNDR	0x1	FEAT_RNG is implemented.
[59:56]	TLB	0x2	FEAT_TLBIOS and FEAT_TLBIRANGE are implemented.
[55:52]	TS	0x2	FEAT_FlagM2 is implemented.
[51:48]	FHM	0x1	FEAT_FHM is implemented.
[47:44]	DP	0x1	FEAT_DotProd is implemented.

[43:40]	SM4	*	Indicates support for SM4 instructions. Defined values are: 0x0 FEAT_SM4 is not implemented. 0x1 FEAT_SM4 is implemented.
[39:36]	SM3	*	Indicates support for SM3 instructions. Defined values are: 0x0 FEAT_SM3 is not implemented. 0x1 FEAT_SM3 is implemented.
[35:32]	SHA3	0x1	FEAT_SHA3 is implemented.
[31:28]	RDM	0x1	FEAT_RDM is implemented.
[27:24]	TME	0x0	TME instructions are not implemented.
[23:20]	Atomic	0x2	FEAT_LSE is implemented, but FEAT_LSE128 is not implemented.
[19:16]	CRC32	0x1	FEAT_CRC32 is implemented.
[15:12]	SHA2	0x2	FEAT_SHA256 and FEAT_SHA512 are implemented.
[11:8]	SHA1	0x1	FEAT_SHA1 is implemented.
[7:4]	AES	*	Indicates support for AES instructions. Defined values are: 0x0 FEAT_AES and FEAT_PMULL are not implemented. 0x2 FEAT_AES and FEAT_PMULL are implemented.
[3:0]	-	0x0	Reserved, RES0

2.10. AArch64 Instruction Set Attribute Register 1, EL1 (ID_AA64ISAR1_EL1)

63 60	59 56	55 52	51 48	47 44	43 40	39 36	35 32
LS64	XS	I8MM	DGH	BF16	SPECRES	SB	FRINTTS

31 28	27 24	23 20	19 16	15 12	11 8	7 4	3 0
GPI	GPA	LRCPC	FCMA	JSCVT	API	APA	DPB

Bits	Name	Value	Function
[63:60]	LS64	0x3	FEAT_LS64, FEAT_LS64_V, and FEAT_LS64_ACCDATA are implemented.
[59:56]	XS	0x1	FEAT_XS is implemented.
[55:52]	I8MM	0x1	FEAT_I8MM is implemented.
[51:48]	DGH	0x0	FEAT_DGH is not implemented.
[47:44]	BF16	0x1	FEAT_BF16 is implemented, but FEAT_SME_F64F64 is not implemented.
[43:40]	SPECRES	0x2	FEAT_SPECRES and FEAT_SPECRES2 are implemented.

[39:36]	SB	0x1	FEAT_SB is implemented.
[35:32]	FRINTTS	0x1	FEAT_FRINTTS is implemented.
[31:28]	GPI	0x0	Generic Authentication using an IMPLEMENTATION DEFINED algorithm is not implemented.
[27:24]	GPA	0x0	FEAT_PACQARMA5 is not implemented, but FEAT_PACQARMA3 is implemented.
[23:20]	LRCPC	0x2	FEAT_LRCPC and FEAT_LRCPC2 are implemented, but FEAT_LRCPC3 is not implemented.
[19:16]	FCMA	0x1	FEAT_FCMA is implemented.
[15:12]	JSCVT	0x1	FEAT_JSCVT is implemented.
[11:8]	API	0x0	Address Authentication using an IMPLEMENTATION DEFINED algorithm is not implemented.
[7:4]	APA	0x0	FEAT_PACQARMA5 is not implemented, but FEAT_PACQARMA3 is implemented.
[3:0]	DPB	0x2	FEAT_DPB and FEAT_DPB2 are implemented.

2.11. AArch64 Instruction Set Attribute Register 2, EL1 (ID_AA64ISAR2_EL1)

63 60	59 56	55 52	51 48	47 44	43 40	39 36	35 32
ATS1A	LUT	CSSC	RPRFM	RES0	PRFMSLC	SYSINSTR_128	SYSREG_128

31 28	27 24	23 20	19 16	15 12	11 8	7 4	3 0
CLRBHB	PAC_frac	BC	MOPS	APA3	GPA3	RPRES	WFxT

Bits	**Name**	**Value**	**Function**
[63:60]	ATS1A	0x0	FEAT_ATS1A is not implemented.
[59:56]	LUT	0x1	FEAT_LUT is implemented. See Arm A-profile Architecture Registers [6].
[55:52]	CSSC	0x0	FEAT_CSSC is not implemented.
[51:48]	RPRFM	0x0	FEAT_RPRFM is not implemented.
[47:44]	-	0x0	Reserved, RES0
[43:40]	PRFMSLC	0x0	FEAT_PRFMSLC is not implemented.
[39:36]	SYSINSTR_128	0x0	FEAT_SYSINSTR128 is not implemented.
[35:32]	SYSREG_128	0x0	FEAT_SYSREG128 is not implemented.
[31:28]	CLRBHB	0x1	FEAT_CLRBHB is implemented.

[27:24]	PAC_frac	0x1	FEAT_CONSTPACFIELD is implemented.
[23:20]	BC	0x1	FEAT_HBC is implemented.
[19:16]	MOPS	0x1	FEAT_MOPS is implemented.
[15:12]	APA3	0x5	FEAT_PACQARMA3 is implemented, and FEAT_FPACCOMBINE is also implemented.
[11:8]	GPA3	0x1	FEAT_PACQARMA3 is implemented.
[7:4]	RPRES	0x0	FEAT_RPRES is not implemented.
[3:0]	WFxT	0x2	FEAT_WFxT is implemented.

2.12. AArch64 Instruction Set Attribute Register 3, EL1 (ID_AA64ISAR3_EL1)

63 32
RES0

31 8	7 4	3 0
RES0	FAMINMAX	RES0

Bits	Name	Value	Function
[63:8]	-	0x00000000_000000	Reserved, RES0
[7:4]	FAMINMAX	0x1	FEAT_FAMINMAX is implemented. See Arm A-profile Architecture Registers [6].
[3:0]	-	0x0	Reserved, RES0

2.13. AArch64 Memory Model Feature Register 0, EL1 (ID_AA64MMFR0_EL1)

63 60	59 56	55 48	47 44	43 40	39 36	35 32
ECV	FGT	RES0	ExS	TGran4_2	TGran64_2	TGran16_2

31 28	27 24	23 20	19 16	15 12	11 8	7 4	3 0
TGran4	TGran64	TGran16	BigEndEL0	SNSMem	BigEnd	ASIDBits	PARange

Bits	Name	Value	Function
[63:60]	ECV	0x2	Enhanced Counter Virtualization is implemented, including support for CNTHCTL_EL2.ECV and CNTPOFF_EL2.
[59:56]	FGT	0x1	FEAT_FGT is implemented, but FEAT_FGT2 is not implemented.
[55:48]	-	0x0	Reserved, RES0
[47:44]	ExS	0x1	FEAT_ExS is implemented.
[43:40]	TGran4_2	0x2	4KB granule supported at stage 2.
[39:36]	TGran64_2	0x2	64KB granule supported at stage 2.
[35:32]	TGran16_2	0x1	16KB granule not supported at stage 2.
[31:28]	TGran4	0x0	4KB granule supported.
[27:24]	TGran64	0x0	64KB granule supported.
[23:20]	TGran16	0x0	16KB granule not supported.
[19:16]	BigEndEL0	0x0	Mixed-endian support. The SCTLR_ELx.EE and SCTLR_EL1.E0E bits can be configured.
[15:12]	SNSMem	0x1	Support a distinction between Secure and Non-secure Memory.
[11:8]	BigEnd	0x1	Mixed-endian support. The SCTLR_ELx.EE and SCTLR_EL1.E0E bits can be configured.
[7:4]	ASIDBits	0x2	FEAT_ASID16 is implemented.
[3:0]	PARange	0x5	48-bits Physical Address range is supported.

2.14. AArch64 Memory Model Feature Register 1, EL1 (ID_AA64MMFR1_EL1)

63 60	59 56	55 52	51 48	47 44	43 40	39 36	35 32
ECBHB	CMOW	TIDCP1	nTLBPA	AFP	HCX	ETS	TWED

31 28	27 24	23 20	19 16	15 12	11 8	7 4	3 0
XNX	SpecSEI	PAN	LO	HPDS	VH	VMIDBits	HAFDBS

Bits	Name	Value	Function
[63:60]	ECBHB	0x1	FEAT_ECBHB is implemented.
[59:56]	CMOW	0x1	FEAT_CMOW is implemented.

[55:52]	TIDCP1	0x1	FEAT_TIDCP1 is implemented.
[51:48]	nTLBPA	0x1	FEAT_nTLBPA is implemented.
[47:44]	AFP	0x1	FEAT_AFP is implemented.
[43:40]	HCX	0x1	FEAT_HCX is implemented.
[39:36]	ETS	0x2	FEAT_ETS2 is implemented.
[35:32]	TWED	0x1	FEAT_TWED is implemented.
[31:28]	XNX	0x1	FEAT_XNX is implemented.
[27:24]	SpecSEI	0x1	The PE might generate an SError exception due to an External abort on a speculative read.
[23:20]	PAN	0x3	FEAT_PAN3 is implemented.
[19:16]	LO	0x1	FEAT_LOR is implemented.
[15:12]	HPDS	0x2	FEAT_HPDS2 is implemented.
[11:8]	VH	0x1	FEAT_VHE is implemented.
[7:4]	VMIDBits	0x2	Number of VMID bits is 16-bits.
[3:0]	HAFDBS	0x2	FEAT_HAFDBS is implemented, but FEAT_HAFT is not implemented.

2.15. AArch64 Memory Model Feature Register 2, EL1 (ID_AA64MMFR2_EL1)

63 60	59 56	55 52	51 48	47 44	43 40	39 36	35 32
E0PD	EVT	BBM	TTL	RES0	FWB	IDS	AT

31 28	27 24	23 20	19 16	15 12	11 8	7 4	3 0
ST	NV	CCIDX	VARange	IESB	LSM	UAO	CnP

Bits	Name	Value	Function
[63:60]	E0PD	0x1	FEAT_E0PD is implemented.
[59:56]	EVT	0x2	FEAT_EVT is implemented, and HCR_EL2.{TTLBOS, TTLBIS, TOCU, TICAB, TID4} traps are supported.
[55:52]	BBM	0x2	FEAT_BBM is implemented, and Level 2 support for changing block size is supported.
[51:48]	TTL	0x1	FEAT_TTL is implemented.
[47:44]	-	0x0	Reserved, RES0
[43:40]	FWB	0x1	FEAT_S2FWB is implemented.

[39:36]	IDS	0x1	FEAT_IDST is implemented.
[35:32]	AT	0x1	FEAT_LSE2 is implemented.
[31:28]	ST	0x1	FEAT_TTST is implemented.
[27:24]	NV	0x2	FEAT_NV and FEAT_NV2 is implemented.
[23:20]	CCIDX	0x1	FEAT_CCIDX is implemented.
[19:16]	VARange	0x0	Neither FEAT_LVA nor FEAT_LVA3 is supported.
[15:12]	IESB	0x1	FEAT_IESB is implemented.
[11:8]	LSM	0x0	FEAT_LSMAOC is not supported.
[7:4]	UAO	0x1	FEAT_UAO is implemented.
[3:0]	CnP	0x1	FEAT_TTCNP is implemented.

2.16. AArch64 Memory Model Feature Register 3, EL1 (ID_AA64MMFR3_EL1)

63 60	59 56	55 52	51 48	47 44	43 40	39 36	35 32
Spec_FPACC	ADERR	SDERR	RES0	ANERR	SNERR	D128_2	D128

31 28	27 24	23 20	19 16	15 12	11 8	7 4	3 0
MEC	AIE	S2POE	S1POE	S2PIE	S1PIE	SCTLRX	TCRX

Bits	**Name**	**Value**	**Function**
[63:60]	Spec_FPACC	0x1	The speculative use of pointers processed by a PAC Authentication is not materially different in terms of the impact on cached microarchitectural state between passing and failing of the PAC Authentication.
[59:56]	ADERR	0x0	FEAT_ADERR is not supported, and All error exceptions for Device memory loads are taken synchronously.
[55:52]	SDERR	0x1	FEAT_ADERR is not supported, and All error exceptions for Device memory loads are taken synchronously.
[51:48]	-	0x0	Reserved, RES0
[47:44]	ANERR	0x0	FEAT_ADERR is not supported, and All error exceptions for Normal memory loads are taken synchronously.
[43:40]	SNERR	0x1	FEAT_ADERR is not supported, and All error exceptions for Normal memory loads are taken synchronously.
[39:36]	D128_2	0x0	128-bit translation table descriptor Extension is not supported.
[35:32]	D128	0x0	128-bit translation table descriptor Extension is not supported.
[31:28]	MEC	0x1	FEAT_MEC is implemented.

[27:24]	AIE	0x0	FEAT_AIE is not implemented.
[23:20]	S2POE	0x0	FEAT_S2POE is not implemented.
[19:16]	S1POE	0x0	FEAT_S1POE is not implemented.
[15:12]	S2PIE	0x0	FEAT_S2PIE is not implemented.
[11:8]	S1PIE	0x0	FEAT_S1PIE is not implemented.
[7:4]	SCTLRX	0x1	SCTLR2_EL1, SCTLR2_EL2, SCTLR2_EL3 resisters, and their associated trap controls are implemented.
[3:0]	TCRX	0x1	TCR2_EL1, TCR2_EL2, and their associated trap controls are implemented.

2.17. AArch64 Memory Model Feature Register 4, EL1 (ID_AA64MMFR4_EL1)

63 32
RES0

31 8	7 4	3 0
RES0	EIESB	RES0

Bits	Name	Value	Function
[63:8]	-	0x00000000_ 000000	Reserved, RES0
[7:4]	EIESB	0x0	Early Implicit Error Synchronization event. Indicates whether the implicit Error synchronization event inserted on taking an exception to ELx when SCTLR_ELx.IESB is 1 is inserted before or after the exception is taken. 0x0 Behavior is not described.
[3:0]	-	0x0	Reserved, RES0

2.18. AArch64 Floating-point Feature Register 0, EL1 (ID_AA64FPFR0_EL1)

63	32
RES0	

31	30	29	28	27	2	1	0
F8CVT	F8FMA	F8DP4	F8DP2	RES0		F8E4M3	F8E5M2

Bits	Name	Value	Function
[63:32]	-	0x00000000	Reserved, RES0
[31]	F8CVT	0x1	FEAT_FP8 is implemented. See Arm A-profile Architecture Registers [6].
[30]	F8FMA	0x1	FEAT_FP8FMA is implemented. See Arm A-profile Architecture Registers [6].
[29]	F8DP4	0x1	FEAT_FP8DOT4 is implemented. See Arm A-profile Architecture Registers [6].
[28]	F8DP2	0x1	FEAT_FP8DOT2 is implemented. See Arm A-profile Architecture Registers [6].
[27:2]	-	0x0000000	Reserved, RES0
[1]	F8E4M3	0x1	Arm FP8 E4M3 format is supported.
[0]	F8E5M2	0x1	Arm FP8 E5M2 format is supported.

2.19. AArch64 Auxiliary Feature Register 0, EL1 (ID_AA64AFR0_EL1)

63	32
RES0	

31	0
RES0	

Bits	Name	Value	Function
[63:0]	-	0x00000000_ 00000000	Reserved, RES0

2.20. Cache Size ID Register, EL1 (CCSIDR_EL1)

63 56	55 32
RES0	NumSets

31 24	23 3	2 0
RES0	Associativity	LineSize

Bits	Name	Value	Function
[63:56]	-	0x00	Reserved, RES0
[55:32]	NumSets	0x*	The number of sets in cache.
[31:24]	-	0x00	Reserved, RES0
[23:3]	Associativity	0x*	The associativity of cache.
[2:0]	LineSize	0x2	The cache line size of FUJITSU-MONAKA is 64-Bytes.

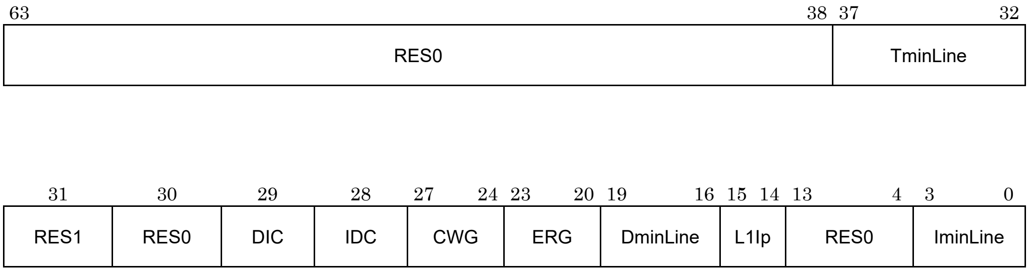
2.21. Cache Level ID Register, EL1 (CLIDR_EL1)

63 47	46 45	44 43	42 41	40 39	38 37	36 35	34 33	32
RES0	Ttype7	Ttype6	Ttype5	Ttype4	Ttype3	Ttype2	Ttype1	ICB

31 30	29 27	26 24	23 21	20 18	17 15	14 12	11 9	8 6	5 3	2 0
ICB	LoUU	LoC	LoUIS	Ctype7	Ctype6	Ctype5	Ctype4	Ctype3	Ctype2	Ctype1

Bits	Name	Value	Function
[63:47]	-	0x00000	Reserved, RES0
[46:45]	Ttype7	0x0	FEAT_MTE2 is not implemented.
[44:43]	Ttype6	0x0	FEAT_MTE2 is not implemented.
[42:41]	Ttype5	0x0	FEAT_MTE2 is not implemented.
[40:39]	Ttype4	0x0	FEAT_MTE2 is not implemented.
[38:37]	Ttype3	0x0	FEAT_MTE2 is not implemented.
[36:35]	Ttype2	0x0	FEAT_MTE2 is not implemented.
[34:33]	Ttype1	0x0	FEAT_MTE2 is not implemented.
[32:30]	ICB	0x2	Level 2 cache is the highest Inner Cacheable level.
[29:27]	LoUU	0x0	Level of Unification Uniprocessor is before the Level 1 cache.
[26:24]	LoC	0x2	Level of Coherence is the Level 2 Cache.
[23:21]	LoUIS	0x0	Level of Unification Inner Shareable is before the Level 1 cache.
[20:18]	Ctype7	0x0	Level 7 cache is not implemented.
[17:15]	Ctype6	0x0	Level 6 cache is not implemented.
[14:12]	Ctype5	0x0	Level 5 cache is not implemented.
[11:9]	Ctype4	0x0	Level 4 cache is not implemented.
[8:6]	Ctype3	0x0	Level 3 cache is invisible.
[5:3]	Ctype2	0x4	Unified instruction and data caches at Level 2.
[2:0]	Ctype1	0x3	Separate instruction and data caches at Level 1.

2.22. Cache type register, EL0 (CTR_EL0)



Bits	Name	Value	Function
[63:38]	-	0x000000	Reserved, RES0
[37:32]	TminLine	0x0	FEAT_MTE2 is not implemented.
[31]	-	0x1	Reserved, RES1
[30]	-	0x0	Reserved, RES0

[29]	DIC	0x1	Instruction cache invalidation requirements for data to instruction coherence. 0x1 Instruction cache invalidation to the Point of Unification is not required for data to instruction coherence.
[28]	IDC	0x1	Data cache clean requirements for instruction to data coherence. 0x1 Data cache clean to the Point of Unification is not required for instruction to data coherence.
[27:24]	CWG	0x4	Cache writeback granule size is 64-Bytes.
[23:20]	ERG	0x4	Exclusive reservation granule size is 64-Bytes.
[19:16]	DminLine	0x4	The smallest cache line of all the data caches is 64-Bytes.
[15:14]	L1Ip	0x3	Level 1 instruction cache policy. 0b11: Physical Index, Physical Tag (PIPT).
[13:4]	-	0x0	Reserved, RES0
[3:0]	IminLine	0x4	The smallest cache line of all the instruction caches is 64-Bytes.

2.23. Data Cache Zero ID, EL0 (DCZID_EL0)

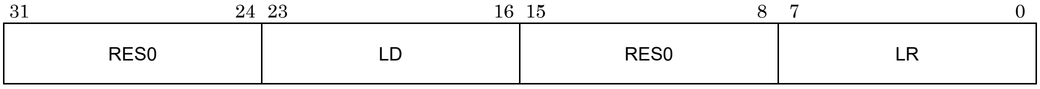
63	32
RES0	

31	5	4	3	0
RES0		DZP	BS	

Bits	Name	Value	Function
[31:5]	-	0x0000000	Reserved, RES0
[4]	DZP		Data Zero Prohibited. This field indicates whether use of DC ZVA instructions is permitted or prohibited. The value of this field depends on the configuration.
[3:0]	BS	0x4	The block size of DC ZVA instructions is 64-Byte.

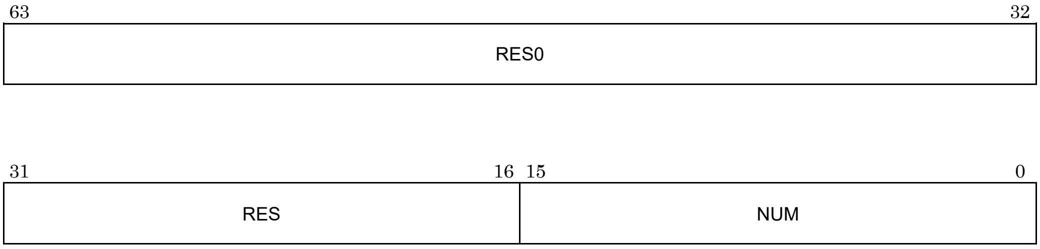
2.24. Limited Order Region ID Register (LORID_EL1)

63	32
RES0	



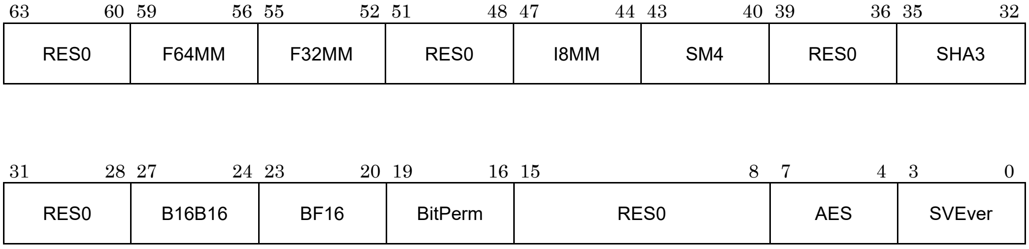
Bits	Name	Value	Function
[63:24]	-	0x0	Reserved, RES0
[23:16]	LD	0x0	The number of LORegion descriptors is zero.
[15:8]	-	0x0	Reserved, RES0
[7:0]	LR	0x0	The number of LORegion is zero.

2.25. Error Record ID Register EL1 (ERRIDR_EL1)



Bits	Name	Value	Function
[63:16]	-	0x0	Reserved, RES0
[15:0]	NUM	0x1	The number of error records is one.

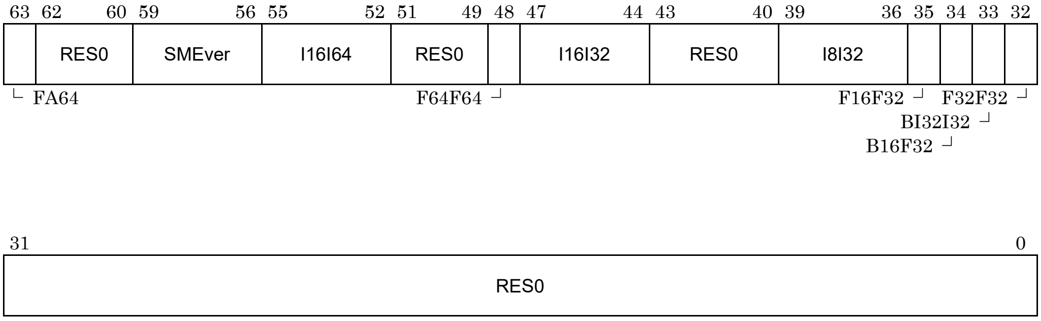
2.26. SVE Feature ID Register 0 (ID_AA64ZFR0_EL1)



Bits	Name	Value	Function
[63:60]	-	0x0	Reserved, RES0
[59:56]	F64MM	0x0	FEAT_F64MM is not implemented.
[55:52]	F32MM	0x0	FEAT_F32MM is not implemented.

[51:48]	-	0x0	Reserved, RES0
[47:44]	I8MM	0x1	FEAT_I8MM is implemented.
[43:40]	SM4	*	Indicates support for SVE SM4 instructions. Defined values are: 0x0 FEAT_SVE_SM4 is not implemented. 0x1 FEAT_SVE_SM4 is implemented.
[39:36]	-	0x0	Reserved, RES0
[35:32]	SHA3	0x1	FEAT_SVE_SHA3 is implemented.
[31:28]	-	0x0	Reserved, RES0
[27:24]	B16B16	0x0	FEAT_SVE_B16B16 is not implemented.
[23:20]	BF16	0x1	FEAT_BF16 is implemented, but FEAT_SME_F64F64 is not implemented.
[19:16]	BitPerm	0x1	FEAT_SVE_BitPerm is implemented.
[15:8]	-	0x0	Reserved, RES0
[7:4]	AES	*	Indicates support for SVE AES instructions. Defined values are: 0x0 FEAT_SVE_AES and FEAT_SVE_PMULL128 are not implemented. 0x2 FEAT_SVE_AES and FEAT_SVE_PMULL128 are implemented.
[3:0]	SVEver	0x1	FEAT_SVE and FEAT_SVE2 are implemented.

2.27. SME Feature ID register 0 (ID_AA64SMFR0_EL1)



Bits	Name	Value	Function
[63]	FA64	`0x0`	FEAT_SME is not implemented.
[62:60]	-	`0x0`	Reserved, RES0
[59:56]	SMEver	`0x0`	FEAT_SME is not implemented.
[55:52]	I16I64	`0x0`	FEAT_SME is not implemented.
[51:49]	-	`0x0`	Reserved, RES0
[48]	F64F64	`0x0`	FEAT_SME is not implemented.
[47:44]	I16I32	`0x0`	FEAT_SME is not implemented.
[43:40]	-	`0x0`	Reserved, RES0
[39:36]	I8I32	`0x0`	FEAT_SME is not implemented.
[35]	F16F32	`0x0`	FEAT_SME is not implemented.
[34]	B16F32	`0x0`	FEAT_SME is not implemented.
[33]	BI32I32	`0x0`	FEAT_SME is not implemented.
[32]	F32F32	`0x0`	FEAT_SME is not implemented.
[31:0]	-	`0x00000000`	Reserved, RES0

2.28. MPAM ID Register EL1 (MPAMIDR_EL1)

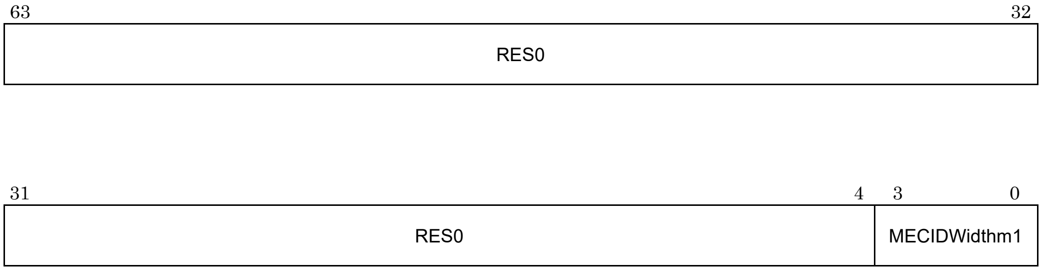
63 62	61	60	59	58	57	56 40	39 32
RES0	HAS_SDEFLT	HAS_FORCE_NS	SP4	HAS_TIDR	HAS_ALTSP	RE0	PMG_MAX

31 21	20 18	17	16	15 0
RES0	VPMR_MAX	HAS_HCR	RES0	PARTID_MAX

Bits	Name	Value	Function
[63:62]	-	`0x0`	Reserved, RES0
[61]	HAS_SDEFLT	`0x1`	MPAM3_EL3.SDEFLT bit is implemented.
[60]	HAS_FORCE_NS	`0x0`	MPAM3_EL3.FORCE_NS bit is not implemented.
[59]	SP4	`0x1`	4 PARTID space is supported.
[58]	HAS_TIDR	`0x1`	MPAM2_EL2.TIDR is implemented.
[57]	HAS_ALTSP	`0x1`	Alternative PARTID spaces are implemented.
[56:40]	-	`0x00000`	Reserved, RES0
[39:32]	PMG_MAX	`0x1`	The largest value of PMG that the implementation can generate is 1.

[31:21]	-	`0x0000`	Reserved, RES0
[20:18]	VPMR_MAX	`0x3`	The number of virtual PARTIDs supported by the implementation is 16.
[17]	HAS_HCR	`0x1`	MPAM virtualization is supported.
[16]	-	`0x0`	Reserved, RES0
[15:0]	PARTID_MAX	`0x3e`	The largest value of PARTID that the implementation can generate is 62.

2.29. MEC Identification Register (MECIDR_EL2)



Bits	Name	Value	Function
[63:4]	-	`0x0`	Reserved, RES0
[3:0]	MECIDWidthm1	`0xb`	The number of bits of MECID supported by the PE is 12-bits.

3. RAS Extension

The FUJITSU-MONAKA processor supports the RAS (Reliability, Availability, and Serviceability) features compliant with ARM RAS Extension. This section details the implementation of these RAS features. Refer to the Arm® Architecture Reference Manual for A-profile architecture[1].

3.1. Overview

The FUJITSU-MONAKA processor conforms to the RAS Extension and RAS System Architecture v1.1.

3.2. Error Record Group

FUJITSU-MONAKA has two types of Error Nodes: PE, accessed via system registers, and Non-PE, accessed via memory-mapped registers.

A single PE Error Node consists of a single Error Record Group containing one Error Record.

See the Table 3-1 for details on Non-PE Error Nodes.

3.3. System Registers

In the FUJITSU-MONAKA processor, the Error Record in a PE Error Node can be accessed using MSR/MRS instructions through the ERX* system registers. This section describes the FUJITSU-MONAKA processor's implementation of these system registers. Access to the Error Records in Non-PE Error Nodes is through memory-mapped registers. See 3.4 Memory Mapped Registers.

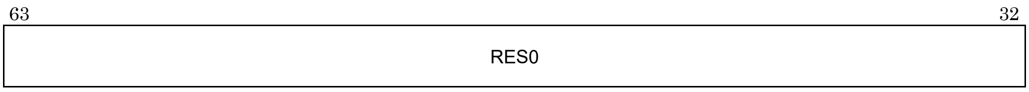
3.3.1. Error Record 0 Feature Register (ERR0FR)

63 55	54 53	52	51	50	49	48	47 32
RES0	CE	DE	UEO	UER	UEU	UC	RES0

31	30 26	25 24	23 22	21 20	19 18	17 16	15	14 12	11 10	9 8	7 6	5 4	3 2	1 0
FRX	RES0	TS	CI	INJ	CEO	DUI	RP	CEC	CFI	UE	FI	UI	RES0	ED

Bits	Name	Value	Function
[63:55]	-	0x0	Reserved, RES0
[54:53]	CE	0x0	The PE does not record Corrected errors.
[52]	DE	0x0	The PE does not record Deferred errors.
[51]	UEO	0x1	The PE records Latent or Restartable errors.
[50]	UER	0x1	The PE records Signaled or Recoverable errors.
[49]	UEU	0x1	The PE records Unrecoverable errors.
[48]	UC	0x0	The PE does not record Uncontainable errors.
[47:32]	-	0x0	Reserved, RES0
[31]	FRX	0x1	ERR0FR[63:48] are defined by the Arm architecture.
[30:26]	-	0x0	Reserved, RES0
[25:24]	TS	0x1	The PE implements a timestamp register in ERR0MISC3, and the timestamp uses the same timebase as the system Generic Timer.
[23:22]	CI	0x0	The PE does not support the critical error interrupt.
[21:20]	INJ	0x1	The PE supports the Common Fault Injection Model Extension.
[19:18]	CEO	0x0	The PE does not implement the Corrected error counter.
[17:16]	DUI	0x0	The PE does not record Deferred errors.
[15]	RP	0x0	The PE does not implement the Corrected error counter.
[14:12]	CEC	0x0	The PE does not implement the Corrected error counter.
[11:10]	CFI	0x0	The PE does not record Corrected errors.
[9:8]	UE	0x1	In-band error response is supported and always enabled.
[7:6]	FI	0x2	Fault handling interrupt is supported and controllable using ERR0CTLR.FI.
[5:4]	UI	0x2	Error recovery interrupt is supported and controllable using ERR0CTLR.UI.
[3:2]		0x0	Reserved, RES0
[1:0]	ED	0x1	Error reporting and logging always enabled.

3.3.2. Error Record 0 Control Register (ERR0CTLR)



31 14	13	12 11	10	9	8	7 5	4	3	2	1	0
RES0	CI	RES0	DUI	RES0	CFI	RES0	UE	FI	UI		ED
										IMP DEF	

Bits	Name	Value	Function
[63:14]	-	0x0	Reserved, RES0
[13]	CI	0x0	The PE does not support the critical error interrupt.
[12:11]	-	0x0	Reserved, RES0
[10]	DUI	0x0	The PE does not record Deferred errors.
[9]	-	0x0	Reserved, RES0
[8]	CFI	0x0	The PE does not record Corrected errors.
[7:5]	-	0x0	Reserved, RES0
[4]	UE	0x0	In-band error response is always enabled.
[3]	FI	RW	Fault handling interrupt is controllable.
[2]	UI	RW	Error recovery interrupt is controllable.
[1]	-	0x0	Reserved, RES0
[0]	ED	0x0	Error reporting and logging always enabled.

3.3.3. Error Record 0 Status Register (ERR0STATUS)

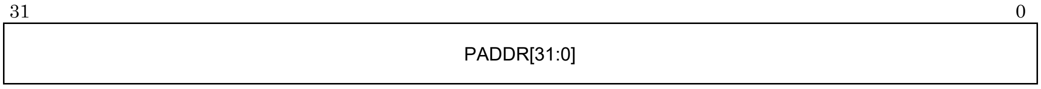
63 32
RES0

31	30	29	28	27	26	25 24	23	22	21 20	19	18 16	15 8	7 0
AV	V	UE	ER	OF	MV	CE	DE	PN	UET	CI	RES0	IERR	SERR

Bits	Name	Value	Function
[63:32]	-	0x0	Reserved, RES0
[31]	AV	W1C	When a synchronous error (Data Abort or Instruction Abort) is detected, this field is updated to 0b1.
[30]	V	W1C	When an error is detected, this field is set to 1.
[29]	UE	W1C	When an error is detected, this field is set to 1.
[28]	ER	W1C	When an error is detected, this field is set to 1.
[27]	OF	W1C	Indicates that multiple errors have been detected.
[26]	MV	W1C	When an error is detected, this field is set to 1.
[25:24]	CE	0x0	This field is always set to 0.
[23]	DE	0x0	This field is always set to 0.
[22]	PN	W1C	When a MARKED_UE is detected, this field is set to 1.
[21:20]	UET	W1C	Uncorrected Error Type. See the Armv9-A arm[1].
[19]	CI	0x0	This field is always set to 0.
[18:16]	-	0x0	Reserved, RES0
[15:13]	IERR[7:5]	RW	Reserved, RES0
[12:8]	IERR[4:0]	RW	IMPLEMENTATION DEFINED error code. This field is valid when ERR0STATUS.SERR is set to 0x01. The FUJITSU-MONAKA processor might detect the following error codes: 0x01: UE_raw_L1 0x02: UE_raw_L2 0x04: DG_L1TLB 0x05: DG_L2 0x07: MARKED_UE 0x08: ue_address_error (software fault) 0x10: GPR, SP register error 0x11: FP&SIMD register error Others: Reserved
[7:0]	SERR	RW	Architecturally-defined primary error code. The FUJITSU-MONAKA processor might detect the following error codes: 0x00: No Error 0x01: IMPLEMENTATION DEFINED error 0x04: Assertion failure 0x05: Internal data path 0x11: Internal control register

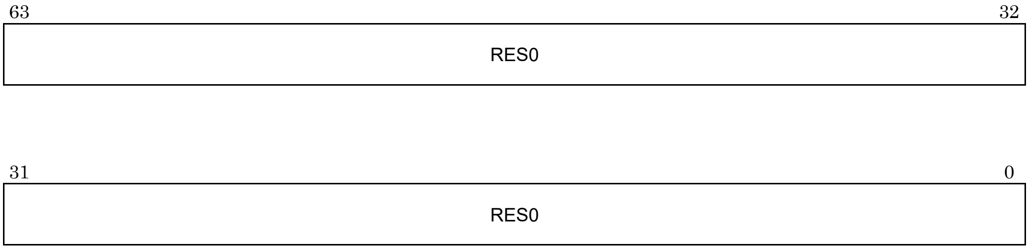
3.3.4. Error Record 0 Address Register (ERR0ADDR)

63	62	61	60	59	58 56	55 32
NS	SI	AI	VA	NSE	RES0	PADDR[55:32]



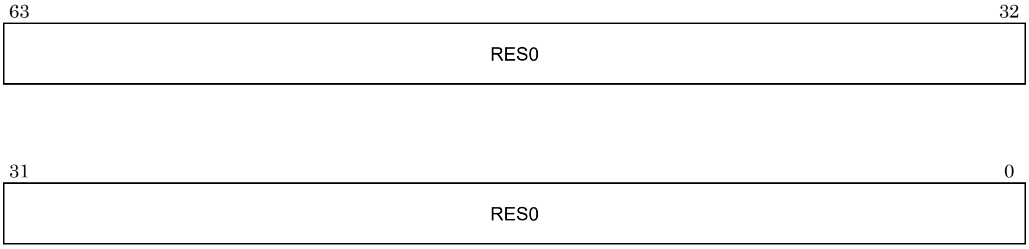
Bits	Name	Value	Function
[63]	NS	RW	Non-secure attribute. See the Armv9-A arm[1].
[62]	SI	RW	In the FUJITSU-MONAKA processor, when ERR0STATUS.AV = 1, this field always reports 0.
[61]	AI	RW	In the FUJITSU-MONAKA processor, when ERR0STATUS.AV = 1, this field always reports 0.
[60]	VA	0x0	This field is always set to 0.
[59]	NSE	RW	Physical Address Space. See the Armv9-A arm[1].
[58:56]	-	0x0	Reserved, RES0
[55:48]	PADDR[55:48]	0x0	In the FUJITSU-MONAKA processor, since the physical address size is 48-bits, this field is RAZ/WI.
[47:6]	PADDR[47:6]	RW	In the FUJITSU-MONAKA processor, granule size of the physical address is 64 bytes. This field is valid when the PE detects the following synchronous errors: • MARKED_UE • Uncorrectable Address Error (Load, Instruction access)
[5:0]	PADDR[5:0]	0x0	In the FUJITSU-MONAKA processor, the lower 5-bits of PADDR field are RES0.

3.3.5. Error Record 0 Miscellaneous 0 Register (ERR0MISC0)



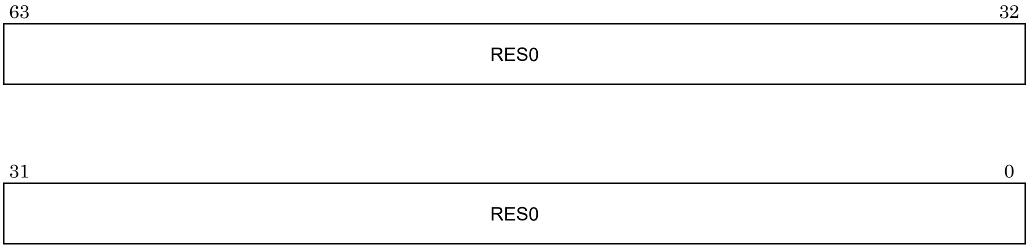
Bits	Name	Value	Function
[63:0]	-	0x0	Reserved, RES0. The PE does not implement the Corrected error counter.

3.3.6. Error Record 0 Miscellaneous 1 Register (ERR0MISC1)



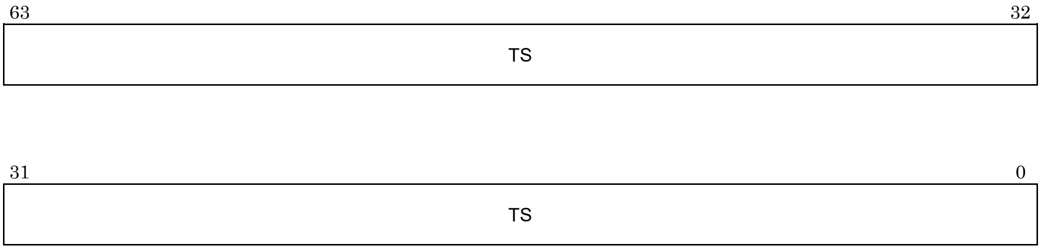
Bits	Name	Value	Function
[63:0]	-	0x0	Reserved, RES0

3.3.7. Error Record 0 Miscellaneous 2 Register (ERR0MISC2)



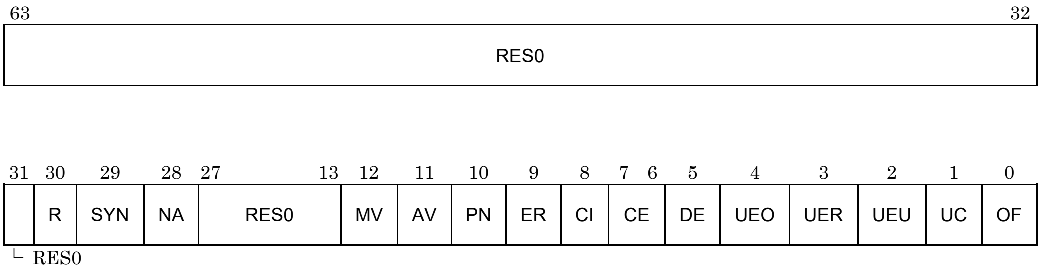
Bits	Name	Value	Function
[63:0]	-	0x0	Reserved, RES0

3.3.8. Error Record 0 Miscellaneous 3 Register (ERR0MISC3)



Bits	Name	Value	Function
[63:0]	TS	RW	Timestamp value recorded when the error was detected.

3.3.9. Error Record 0 Pseudo fault Generation Feature Register (ERR0PFGF)



Bits	Name	Value	Function
[63:31]	-	0x0	Reserved, RES0
[30]	R	0x1	Error Generation Counter restart mode is implemented.
[29]	SYN	0x1	When an injected error is recorded, the node does not update the ERR0STATUS.{IERR, SERR} fields. ERR0STATUS.{IERR, SERR} are writable when ERR0STATUS.V is 0.
[28]	NA	0x0	The node fakes detection of the error on a memory access.
[27:13]	-	0x0	Reserved, RES0
[12]	MV	0x1	When an injected error is recorded, ERR0STATUS.MV is set to ERR0PFGCTL.MV and ERR0MISC3 is not updated.
[11]	AV	0x1	When an injected error is recorded, ERR0STATUS.AV is set to ERR0PFGCTL.AV and ERR0ADDR is not updated.
[10]	PN	0x1	When an injected error is recorded, ERR0STATUS.PN is set to ERR0PFGCTL.PN.
[9]	ER	0x1	When an injected error is recorded, ERR0STATUS.ER is set to ERR0PFGCTL.ER.
[8]	CI	0x0	When an injected error is recorded, the node does not update the ERR0STATUS.CI field.
[7:6]	CE	0x0	When an injected error is recorded, the node does not update the ERR0STATUS.CE field.
[5]	DE	0x0	The node does not support Deferred Error.
[4]	UEO	0x1	The node supports the generation of Latent or Restartable errors.
[3]	UER	0x1	The node supports the generation of Signaled or Recoverable errors.
[2]	UEU	0x1	The node supports the generation of Unrecoverable errors.
[1]	UC	0x0	The node does not support Uncontainable Error.
[0]	OF	0x1	When an injected error is recorded, ERR0STATUS.OF is set to ERR0PFGCTL.OF.

3.3.10. Error Record 0 Pseudo-fault Generation Control Register (ERR0PFGCTL)

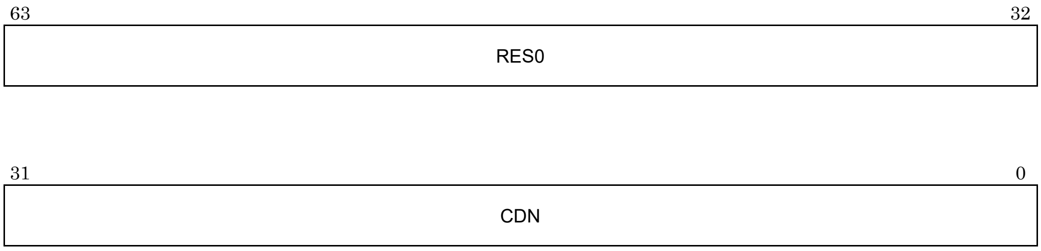
63 32
RES0

31	30	29 13	12	11	10	9	8	7 6	5	4	3	2	1	0
	R	RES0	MV	AV	PN	ER	CI	CE	DE	UEO	UER	UEU	UC	OF

└ CDNEN

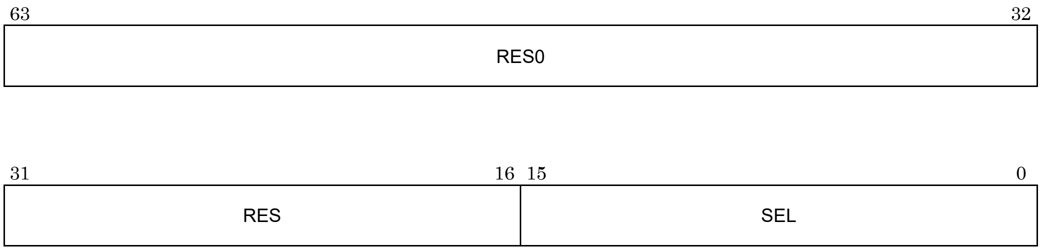
Bits	Name	Value	Function
[63:32]	-	0x0	Reserved, RES0
[31]	CDNEN	RW	Countdown Enable. See the Armv9-A arm[1].
[30]	R	RW	Error Generation Counter restart mode is implemented.
[29:13]	-	0x0	Reserved, RES0
[12]	MV	RW	Miscellaneous syndrome. When an injected error is recorded, ERR0STATUS.MV is set to ERR0PFGCTL.MV
[11]	AV	RW	Address syndrome. When an injected error is recorded, ERR0STATUS.AV is set to ERR0PFGCTL.AV.
[10]	PN	RW	Poison flag. When an injected error is recorded, ERR0STATUS.PN is set to ERR0PFGCTL.PN.
[9]	ER	RW	Error Reported flag. When an injected error is recorded, ERR0STATUS.ER is set to ERR0PFGCTL.ER.
[8]	CI	0x0	When an injected error is recorded, the node does not update the ERR0STATUS.CI field. This field is RAZ/WI.
[7:6]	CE	0x0	When an injected error is recorded, the node does not update the ERR0STATUS.CE field. This field is RAZ/WI.
[5]	DE	0x0	The node does not support Deferred Error. This field is RAZ/WI.
[4]	UEO	RW	The node supports the generation of Latent or Restartable errors.
[3]	UER	RW	The node supports the generation of Signaled or Recoverable errors.
[2]	UEU	RW	The node supports the generation of Unrecoverable errors.
[1]	UC	RW	The node does not support Uncontainable errors.
[0]	OF	RW	Overflow flag. When an injected error is recorded, ERR0STATUS.OF is set to ERR0PFGCTL.OF.

3.3.11. Error Record 0 Pseudo-fault Generation Countdown Register (ERR0PFGCDN)



Bits	Name	Value	Function
[63:32]	-	0x0	Reserved, RES0
[31:0]	CDN	RW	Countdown value. See the Armv9-A arm[1].

3.3.12. Error Record Select Register EL1 (ERRSELR_EL1)



Bits	Name	Value/Attr.	Function
[63:16]	-	0x0	Reserved, RES0
[15:0]	SEL	RW	In the FUJITSU-MONAKA processor, when a value other than 0 is set in this field, the ERX* register is treated as RAZ/WI.

3.4. Memory Mapped Registers

In the FUJITSU-MONAKA processor, the Error Records in non-PE can be accessed using memory access instructions through memory-mapped registers. This section describes the FUJITSU-MONAKA processor's implementation of these memory-mapped registers.

Table 3-1 shows the base address for each Error Group and the corresponding node and record for each group. Since each node only contain a single record, some error information will be lost if multiple errors are detected on a node simultaneously.

Table 3-1 The base address and the node of RAS memory-mapped registers

Group	Base address	Record	Node
SRD#0	0x0010_3310_0000	Record#0	SRD#0

Group	Base address	Record	Node
SRD#1	0x0020_3310_0000	Record#0	SRD#1
SRD#2	0x0030_3310_0000	Record#0	SRD#2
SRD#3	0x0040_3310_0000	Record#0	SRD#3
SRD#4	0x0060_3310_0000	Record#0	SRD#4
SRD#5	0x0070_3310_0000	Record#0	SRD#5
SRD#6	0x0080_3310_0000	Record#0	SRD#6
SRD#7	0x0090_3310_0000	Record#0	SRD#7
IOD#0	0x0000_3310_0000	Record#0	IOD#0
IOD#1	0x0050_3310_0000	Record#0	IOD#1
MAC#00-CH#0	0x0001_811E_0000	Record#0	MAC#00-CH#0
		Record#1	DIMM#00A/#00B (CH-A)
MAC#00-CH#1	0x0001_813E_0000	Record#0	MAC#00-CH#01
		Record#1	DIMM#00A/#00B (CH-B)
MAC#01-CH#0	0x0001_815E_0000	Record#0	MAC#01-CH#0
		Record#1	DIMM#01A/#01B (CH-A)
MAC#01-CH#1	0x0001_817E_0000	Record#0	MAC#01-CH#1
		Record#1	DIMM#01A/#01B (CH-B)
MAC#02-CH#0	0x0001_819E_0000	Record#0	MAC#02-CH#0
		Record#1	DIMM#02A/#02B (CH-A)
MAC#02-CH#1	0x0001_81BE_0000	Record#0	MAC#02-CH#1
		Record#1	DIMM#02A/#02B (CH-B)
MAC#03-CH#0	0x0001_891E_0000	Record#0	MAC#03-CH#0
		Record#1	DIMM#03A/#03B (CH-A)
MAC#03-CH#1	0x0001_893E_0000	Record#0	MAC#03-CH#1
		Record#1	DIMM#03A/#03B (CH-B)
MAC#04-CH#0	0x0001_895E_0000	Record#0	MAC#04-CH#0
		Record#1	DIMM#04A/#04B (CH-A)
MAC#04-CH#1	0x0001_897E_0000	Record#0	MAC#04-CH#1
		Record#1	DIMM#04A/#04B (CH-B)
MAC#05-CH#0	0x0001_899E_0000	Record#0	MAC#05-CH#0
		Record#1	DIMM#05A/#05B (CH-A)
MAC#05-CH#1	0x0001_89BE_0000	Record#0	MAC#05-CH#1
		Record#1	DIMM#05A/#05B (CH-B)
MAC#06-CH#0	0x0001_851E_0000	Record#0	MAC#06-CH#0
		Record#1	DIMM#06A/#06B (CH-A)
MAC#06-CH#1	0x0001_853E_0000	Record#0	MAC#06-CH#1
		Record#1	DIMM#06A/#06B (CH-B)
MAC#07-CH#0	0x0001_855E_0000	Record#0	MAC#07-CH#0
		Record#1	DIMM#07A/#07B (CH-A)
MAC#07-CH#1	0x0001_857E_0000	Record#0	MAC#07-CH#01
		Record#1	DIMM#07A/#07B (CH-B)
MAC#08-CH#0	0x0001_859E_0000	Record#0	MAC#08-CH#0
		Record#1	DIMM#08A/#08B (CH-A)
MAC#08-CH#1	0x0001_85BE_0000	Record#0	MAC#08-CH#1
		Record#1	DIMM#08A/#08B (CH-B)
MAC#09-CH#0	0x0001_8D1E_0000	Record#0	MAC#09-CH#0
		Record#1	DIMM#09A/#09B (CH-A)
MAC#09-CH#1	0x0001_8D3E_0000	Record#0	MAC#09-CH#1

Group	Base address	Record	Node
		Record#1	DIMM#09A/#09B (CH-B)
MAC#10-CH#0	0x0001_8D5E_0000	Record#0	MAC#10-CH#0
		Record#1	DIMM#10A/#10B (CH-A)
MAC#10-CH#1	0x0001_8D7E_0000	Record#0	MAC#10-CH#1
		Record#1	DIMM#10A/#10B (CH-B)
MAC#11-CH#0	0x0001_8D9E_0000	Record#0	MAC#11-CH#0
		Record#1	DIMM#11A/#11B (CH-A)
MAC#11-CH#1	0x0001_8DBE_0000	Record#0	MAC#11-CH#01
		Record#1	DIMM#11A/#11B (CH-B)
MAC#12-CH#0	0x0051_811E_0000	Record#0	MAC#12-CH#0
		Record#1	DIMM#12A/#12B (CH-A)
MAC#12-CH#1	0x0051_813E_0000	Record#0	MAC#12-CH#1
		Record#1	DIMM#12A/#12B (CH-B)
MAC#13-CH#0	0x0051_815E_0000	Record#0	MAC#13-CH#0
		Record#1	DIMM#13A/#13B (CH-A)
MAC#13-CH#1	0x0051_817E_0000	Record#0	MAC#13-CH#01
		Record#1	DIMM#13A/#13B (CH-B)
MAC#14-CH#0	0x0051_819E_0000	Record#0	MAC#14-CH#0
		Record#1	DIMM#14A/#14B (CH-A)
MAC#14-CH#1	0x0051_81BE_0000	Record#0	MAC#14-CH#01
		Record#1	DIMM#14A/#14B (CH-B)
MAC#15-CH#0	0x0051_891E_0000	Record#0	MAC#15-CH#0
		Record#1	DIMM#15A/#15B (CH-A)
MAC#15-CH#1	0x0051_893E_0000	Record#0	MAC#15-CH#01
		Record#1	DIMM#15A/#15B (CH-B)
MAC#16-CH#0	0x0051_895E_0000	Record#0	MAC#16-CH#0
		Record#1	DIMM#16A/#16B (CH-A)
MAC#16-CH#1	0x0051_897E_0000	Record#0	MAC#16-CH#01
		Record#1	DIMM#16A/#16B (CH-B)
MAC#17-CH#0	0x0051_899E_0000	Record#0	MAC#17-CH#0
		Record#1	DIMM#17A/#17B (CH-A)
MAC#17-CH#1	0x0051_89BE_0000	Record#0	MAC#17-CH#01
		Record#1	DIMM#17A/#17B (CH-B)
MAC#18-CH#0	0x0051_851E_0000	Record#0	MAC#18-CH#0
		Record#1	DIMM#18A/#18B (CH-A)
MAC#18-CH#1	0x0051_853E_0000	Record#0	MAC#18-CH#01
		Record#1	DIMM#18A/#18B (CH-B)
MAC#19-CH#0	0x0051_855E_0000	Record#0	MAC#19-CH#0
		Record#1	DIMM#19A/#19B (CH-A)
MAC#19-CH#1	0x0051_857E_0000	Record#0	MAC#19-CH#01
		Record#1	DIMM#19A/#19B (CH-B)
MAC#20-CH#0	0x0051_859E_0000	Record#0	MAC#20-CH#0
		Record#1	DIMM#20A/#20B (CH-A)
MAC#20-CH#1	0x0051_85BE_0000	Record#0	MAC#20-CH#01
		Record#1	DIMM#20A/#20B (CH-B)
MAC#21-CH#0	0x0051_8D1E_0000	Record#0	MAC#21-CH#0
		Record#1	DIMM#21A/#21B (CH-A)
MAC#21-CH#1	0x0051_8D3E_0000	Record#0	MAC#21-CH#01
		Record#1	DIMM#21A/#21B (CH-B)

Group	Base address	Record	Node
MAC#22-CH#0	0x0051_8D5E_0000	Record#0	MAC#22-CH#0
		Record#1	DIMM#22A/#22B (CH-A)
MAC#22-CH#1	0x0051_8D7E_0000	Record#0	MAC#22-CH#1
		Record#1	DIMM#22A/#22B (CH-B)
MAC#23-CH#0	0x0051_8D9E_0000	Record#0	MAC#23-CH#0
		Record#1	DIMM#23A/#23B (CH-A)
MAC#23-CH#1	0x0051_8DBE_0000	Record#0	MAC#23-CH#01
		Record#1	DIMM#23A/#23B (CH-B)
PCIe#0	0x0002_8781_0000	Record#0	PCIe#0
PCIe#1	0x0002_8F81_0000	Record#0	PCIe#1
PCIe#2	0x0002_9781_0000	Record#0	PCIe#2
PCIe#3	0x0002_9F81_0000	Record#0	PCIe#3
PCIe#4	0x0002_A781_0000	Record#0	PCIe#4
PCIe#5	0x0002_AF81_0000	Record#0	PCIe#5
PCIe#6	0x0052_8781_0000	Record#0	PCIe#6
PCIe#7	0x0052_8F81_0000	Record#0	PCIe#7
PCIe#8	0x0052_9781_0000	Record#0	PCIe#8
PCIe#9	0x0052_9F81_0000	Record#0	PCIe#9
PCIe#10	0x0052_A781_0000	Record#0	PCIe#10
PCIe#11	0x0052_AF81_0000	Record#0	PCIe#11

3.4.1. Error Record Feature Register (ERR<n>FR)

63 55	54 53	52	51	50	49	48	47 32
RES0	CE	DE	UEO	UER	UEU	UC	RES0

31	30 26	25 24	23 22	21 20	19 18	17 16	15	14 12	11 10	9 8	7 6	5 4	3 2	1 0
FRX	RES0	TS	CI	INJ	CEO	DUI	RP	CEC	CFI	UE	FI	UI	RES0	ED

Field description of SRD/IOD error nodes

Bits	Name	Value	Function
[63:55]	-	0x0	Reserved, RES0
[54:53]	CE	0x0	The node does not record Corrected errors.
[52]	DE	0x0	The node does not record Deferred errors.
[51]	UEO	0x1	The node records Latent or Restartable errors.
[50]	UER	0x0	The node does not record Signaled or Recoverable errors.
[49]	UEU	0x0	The node does not record Unrecoverable errors.
[48]	UC	0x0	The node does not record Uncontainable errors.

[47:32]	-	0x0	Reserved, RES0
[31]	FRX	0x1	ERR<n>FR[63:48] are defined by the Arm architecture.
[30:26]	-	0x0	Reserved, RES0
[25:24]	TS	0x1	The node implements a timestamp register in ERR<n>MISC3, and the timestamp uses the same timebase as the system Generic Timer.
[23:22]	CI	0x0	The node does not support the critical error interrupt.
[21:20]	INJ	0x1	The node supports the Common Fault Injection Model Extension.
[19:18]	CEO	0x0	The node does not implement the Corrected error counter.
[17:16]	DUI	0x0	The node does not record Deferred errors.
[15]	RP	0x0	The node does not implement the Corrected error counter.
[14:12]	CEC	0x0	The node does not implement the Corrected error counter.
[11:10]	CFI	0x0	The node does not record Corrected errors.
[9:8]	UE	0x0	The node does not support the in-band error response.
[7:6]	FI	0x2	Fault handling interrupt is supported and controllable using ERR<n>CTLR.FI.
[5:4]	UI	0x2	Error recovery interrupt is supported and controllable using ERR<n>CTLR.UI.
[3:2]	-	0x0	Reserved, RES0
[1:0]	ED	0x1	Error reporting and logging always enabled.

Field description of MAC error nodes

Bits	**Name**	**Value**	**Function**
[63:55]	-	0x0	Reserved, RES0
[54:53]	CE	0x0	The node does not record Corrected errors.
[52]	DE	0x1	The node records Deferred errors.
[51]	UEO	0x1	The node records Latent or Restartable errors.
[50]	UER	0x0	The node does not record Signaled or Recoverable errors.
[49]	UEU	0x1	The node records Unrecoverable errors.
[48]	UC	0x0	The node does not record Uncontainable errors.
[47:32]	-	0x0	Reserved, RES0
[31]	FRX	0x1	ERR<n>FR[63:48] are defined by the Arm architecture.
[30:26]	-	0x0	Reserved, RES0
[25:24]	TS	0x0	The node does not support a timestamp register.
[23:22]	CI	0x0	The node does not support the critical error interrupt.
[21:20]	INJ	0x1	The node supports the Common Fault Injection Model Extension.
[19:18]	CEO	0x0	The node does not implement the Corrected error counter.

[17:16]	DUI	0x2	Error recovery interrupt on deferred errors is supported and controllable using ERR<n>CTLR.DUI.
[15]	RP	0x0	The node does not implement the Corrected error counter.
[14:12]	CEC	0x0	The node does not implement the Corrected error counter.
[11:10]	CFI	0x0	The node does not record Corrected errors.
[9:8]	UE	0x0	The node does not support the in-band error response.
[7:6]	FI	0x2	Fault handling interrupt is supported and controllable using ERR<n>CTLR.FI.
[5:4]	UI	0x2	Error recovery interrupt is supported and controllable using ERR<n>CTLR.UI.
[3:2]	-	0x0	Reserved, RES0
[1:0]	ED	0x1	Error reporting and logging always enabled.

Field description of DIMM error nodes

Bits	Name	Value	Function
[63:55]	-	0x0	Reserved, RES0
[54:53]	CE	0x0	The node does not record Corrected errors.
[52]	DE	0x1	The node records Deferred errors.
[51]	UEO	0x0	The node does not record Latent or Restartable errors.
[50]	UER	0x0	The node does not record Signaled or Recoverable errors.
[49]	UEU	0x0	The node does not record Unrecoverable errors.
[48]	UC	0x0	The node does not record Uncontainable errors.
[47:32]	-	0x0	Reserved, RES0
[31]	FRX	0x1	ERR<n>FR[63:48] are defined by the Arm architecture.
[30:26]	-	0x0	Reserved, RES0
[25:24]	TS	0x0	The node does not support a timestamp register.
[23:22]	CI	0x0	The node does not support the critical error interrupt.
[21:20]	INJ	0x1	The node supports the Common Fault Injection Model Extension.
[19:18]	CEO	0x1	Corrected Error overwrite. See the Armv9-A arm[1].
[17:16]	DUI	0x2	Error recovery interrupt on deferred errors is supported and controllable using ERR<n>CTLR.DUI.
[15]	RP	0x1	Repeat counter. See the Armv9-A arm[1].
[14:12]	CEC	0x4	Corrected Error Counter. See 3.4.5 Error Record Miscellaneous Register 0 (ERR<n>MISC0).
[11:10]	CFI	0x2	Enabling and disabling of fault handling interrupts on corrected errors is supported and controllable using ERR<n>CTLR.CFI.
[9:8]	UE	0x0	The node does not support the in-band error response.

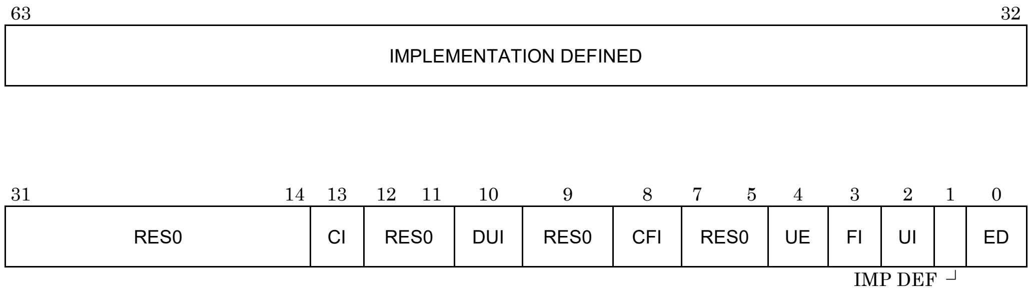
[7:6]	FI	0x2	Fault handling interrupt is supported and controllable using ERR<n>CTLR.FI.
[5:4]	UI	0x2	Error recovery interrupt is supported and controllable using ERR<n>CTLR.UI.
[3:2]	-	0x0	Reserved, RES0
[1:0]	ED	0x1	Error reporting and logging always enabled.

Field description of PCIe error nodes

Bits	Name	Value	Function
[63:55]	-	0x0	Reserved, RES0
[54:53]	CE	0x0	The node does not record Corrected errors.
[52]	DE	0x0	The node does not record Deferred errors.
[51]	UEO	0x0	The node does not record Latent or Restartable errors.
[50]	UER	0x1	The node records Signaled or Recoverable errors.
[49]	UEU	0x0	The node does not record Unrecoverable errors.
[48]	UC	0x0	The node does not record Uncontainable errors.
[47:32]	-	0x0	Reserved, RES0
[31]	FRX	0x1	ERR<n>FR[63:48] are defined by the Arm architecture.
[30:26]	-	0x0	Reserved, RES0
[25:24]	TS	0x0	The node does not support a timestamp register.
[23:22]	CI	0x0	The node does not support the critical error interrupt.
[21:20]	INJ	0x1	The node supports the Common Fault Injection Model Extension.
[19:18]	CEO	0x0	The node does not implement the Corrected error counter.
[17:16]	DUI	0x0	The node does not record Deferred errors.
[15]	RP	0x0	The node does not implement the Corrected error counter.
[14:12]	CEC	0x0	The node does not implement the Corrected error counter.
[11:10]	CFI	0x0	The node does not record Corrected errors.
[9:8]	UE	0x0	The node does not support the in-band error response.
[7:6]	FI	0x2	Fault handling interrupt is supported and controllable using ERR<n>CTLR.FI.
[5:4]	UI	0x2	Error recovery interrupt is supported and controllable using ERR<n>CTLR.UI.
[3:2]	-	0x0	Reserved, RES0
[1:0]	ED	0x1	Error reporting and logging always enabled.

address
Base + 0x000+64n

3.4.2. Error Record Control Register (ERR<n>CTLR)



Field description of SRD/IOD/PCIe error nodes

Bits	Name	Value	Function
[63:14]	-	0x0	Reserved, RES0
[13]	CI	0x0	The node does not support the critical error interrupt.
[12:11]	-	0x0	Reserved, RES0
[10]	DUI	0x0	The node does not record Deferred errors.
[9]	-	0x0	Reserved, RES0
[8]	CFI	0x0	The node does not record Corrected errors.
[7:5]	-	0x0	Reserved, RES0
[4]	UE	0x0	The node does not support the in-band error response.
[3]	FI	RW	Fault handling interrupt enable. See the Armv9-A arm[1].
[2]	UI	RW	Uncorrected error recovery interrupt enable. See the Armv9-A arm[1].
[1]	-	0x0	Reserved, RES0
[0]	ED	0x0	Error reporting and logging always enabled.

Field description of MAC error nodes

Bits	Name	Value	Function
[63:14]	-	0x0	Reserved, RES0
[13]	CI	0x0	The node does not support the critical error interrupt.
[12:11]	-	0x0	Reserved, RES0
[10]	DUI	RW	Error recovery interrupt for Deferred errors enable. See the Armv9-A arm[1].
[9]	-	0x0	Reserved, RES0
[8]	CFI	0x0	The node does not record Corrected errors.
[7:5]	-	0x0	Reserved, RES0
[4]	UE	0x0	The node does not support the in-band error response.
[3]	FI	RW	Fault handling interrupt enable. See the Armv9-A arm[1].
[2]	UI	RW	Uncorrected error recovery interrupt enable. See the Armv9-A arm[1].
[1]	-	0x0	Reserved, RES0
[0]	ED	0x0	Error reporting and logging always enabled.

Field description of DIMM error nodes

Bits	Name	Value	Function
[63:14]	-	0x0	Reserved, RES0
[13]	CI	0x0	The node does not support the critical error interrupt.
[12:11]	-	0x0	Reserved, RES0
[10]	DUI	RW	Error recovery interrupt for Deferred errors enable. See the Armv9-A arm[1].
[9]	-	0x0	Reserved, RES0
[8]	CFI	RW	Fault handling interrupt for corrected error events enable. See the Armv9-A arm[1].
[7:5]	-	0x0	Reserved, RES0
[4]	UE	0x0	The node does not support the in-band error response.
[3]	FI	RW	Fault handling interrupt enable. See the Armv9-A arm[1].
[2]	UI	RW	Uncorrected error recovery interrupt enable. See the Armv9-A arm[1].
[1]	-	0x0	Reserved, RES0
[0]	ED	0x0	Error reporting and logging always enabled.

Address
Base + 0x008+64n

3.4.3. Error Record Status Register (ERR<n>STATUS)

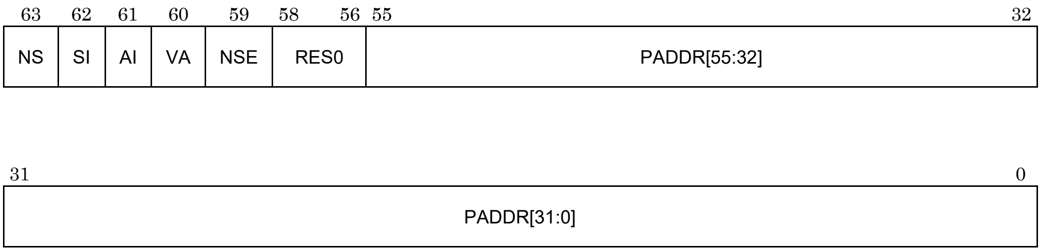
63 32
RES0

31	30	29	28	27	26	25 24	23	22	21 20	19	18 16	15 8	7 0
AV	V	UE	ER	OF	MV	CE	DE	PN	UET	CI	RES0	IERR	SERR

Bits	Name	Value	Function
[63:32]	-	0x0	Reserved, RES0
[31]	AV	W1C	Address Valid. In the FUJITSU-MONAKA processor, this field is set to 1 when some types of errors are detected in the DIMM error node.
[30]	V	W1C	Status Register Valid. See the Armv9-A arm[1].
[29]	UE	W1C	Uncorrected Error. See the Armv9-A arm[1].
[28]	ER	W1C	Error Reported. See the Armv9-A arm[1].
[27]	OF	W1C	Overflow. See the Armv9-A arm[1].
[26]	MV	W1C	Miscellaneous Registers Valid. See the Armv9-A arm[1].
[25:24]	CE	W1C	Corrected Error. See the Armv9-A arm[1].
[23]	DE	W1C	Deferred Error. In the FUJITSU-MONAKA processor, this field is set to 1 when some types of errors are detected in the MAC or DIMM error node.
[22]	PN	W1C	Poison. See the Armv9-A arm[1].
[21:20]	UET	W1C	Uncorrected Error Type. See the Armv9-A arm[1].
[19]	CI	0x0	Critical Error. This field is always set to 0.
[18:16]	-	0x0	Reserved, RES0
[15:8]	IERR	RW	IMPLEMENTATION DEFINED error code. This field is reserved.
[7:0]	SERR	RW	Architecturally-defined primary error code. The FUJITSU-MONAKA processor might detect the following error codes: 0x00: No Error 0x01: IMPLEMENTATION DEFINED error 0x05: Internal data path 0x13: ATS invalidation timeout

Address
Base + 0x010+64n

3.4.4. Error Record Address Register (ERR<n>ADDR)



Field description of SRD/IOD/PCIe/MAC error nodes

Bits	Name	Value	Function
[63]	NS	0x0	This field is always set to 0.
[62]	SI	0x0	This field is always set to 0.
[61]	AI	0x0	This field is always set to 0.
[60]	VA	0x0	This field is always set to 0.
[59]	NSE	0x0	This field is always set to 0.
[58:56]	-	0x0	Reserved, RES0
[55:0]	PADDR	0x0	This field is always set to 0.

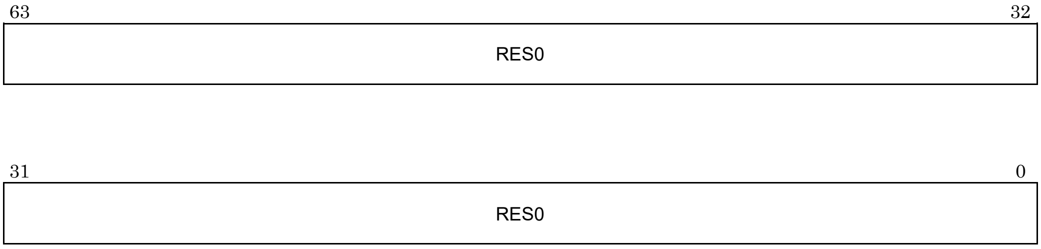
Field description of DIMM error nodes

Bits	Name	Value	Function
[63]	NS	0x0	This field is always set to 0.
[62]	SI	RW	Secure Incorrect. See the Armv9-A arm[1].
[61]	AI	RW	Address Incorrect. See the Armv9-A arm[1].
[60]	VA	RW	Virtual Address. See the Armv9-A arm[1].
[59]	NSE	0x0	This field is always set to 0.
[58:56]	-	0x0	Reserved, RES0
[55:48]	PADDR[55:48]	0x0	This field is always set to 0.
[47:0]	PADDR[47:0]	RW	When ERR<n>STATUS.AV is set to 1, this field reports MAC Channel Physical Address (MAC-CH-PA).

Address
Base + 0x018+64n

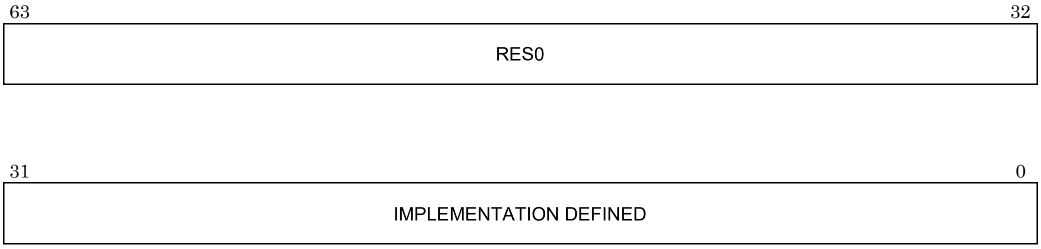
3.4.5. Error Record Miscellaneous Register 0 (ERR<n>MISC0)

Field description of SRD/IOD error nodes



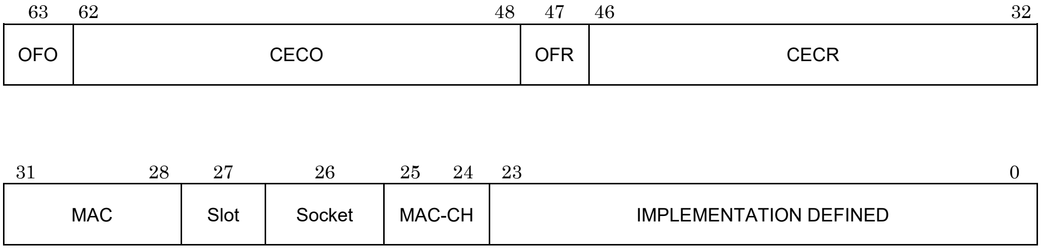
Bits	Name	Value	Function
[63:0]	-	0x0	Reserved, RES0

Field description of MAC error nodes



Bits	Name	Value	Function
[63:32]	-	0x0	Reserved, RES0
[31:0]	IMPLEMENTATION DEFINED	RW	This field is set to IMPLEMENTATION DEFINED code.

Field description of DIMM error nodes



Bits	Name	Value	Function
[63]	OFO	RW	Set to 1 when ERR<n>MISC0.CECO is incremented and wraps through zero.
[62:48]	CECO	RW	This field counts corrected errors for the DIMM in Slot 1 (Far).
[47]	OFR	RW	Set to 1 when ERR<n>MISC0.CECR is incremented and wraps through zero.
[46:32]	CECR	RW	This field counts corrected errors for the DIMM in Slot 2 (Near).
[31:28]	MAC	RW	The MAC number (0x0-0xB) corresponding to the DIMM that detected the error is set.
[27]	Slot	RW	The slot number (0x0-0x1) corresponding to the DIMM that detected the error is set.
[26]	Socket	RW	The socket number (0x0-0x1) corresponding to the DIMM that detected the error is set.
[25:24]	MAC-CH	RW	The MAC Channel number (0x0-0x1) corresponding to the DIMM that detected the error is set.
[23:0]	IMPLEMENTATION DEFINED	RW	This field is set to IMPLEMENTATION DEFINED code.

Field description of PCIe error nodes

63 48	47 32
RES0	BDF

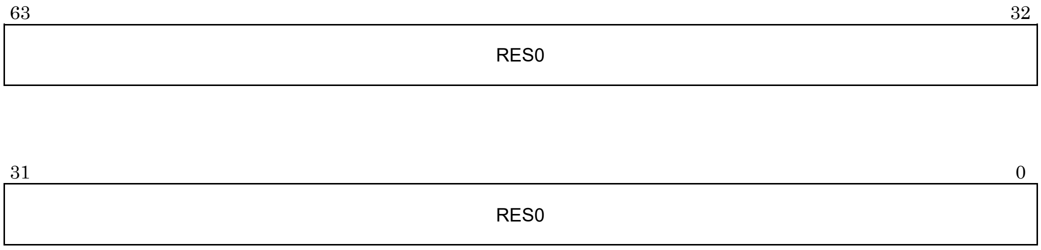
31 1	0
RES0	V

Bits	Name	Value	Function
[63:48]	-	RW	This field is always set to 0.
[47:32]	BDF	RW	When an ATS invalidation timeout is detected, this field is set to the BDF number to which the ATS invalidation was issued.
[31:1]	-	RW	This field is always set to 0.
[0]	V	RW	When an error is detected, this field is set to 1.

Address
Base + 0x020+64n

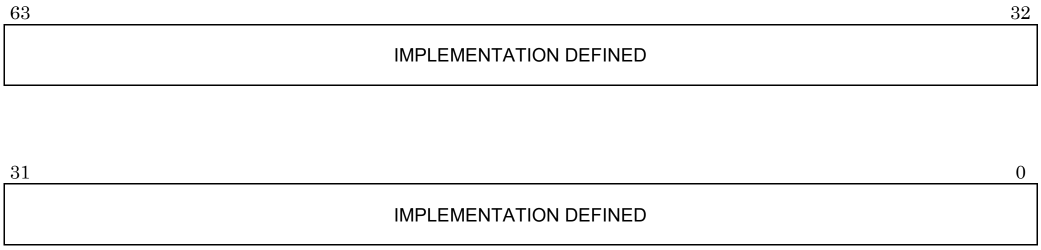
3.4.6. Error Record Miscellaneous Register 1 (ERR<n>MISC1)

Field description of SRD/IOD error nodes



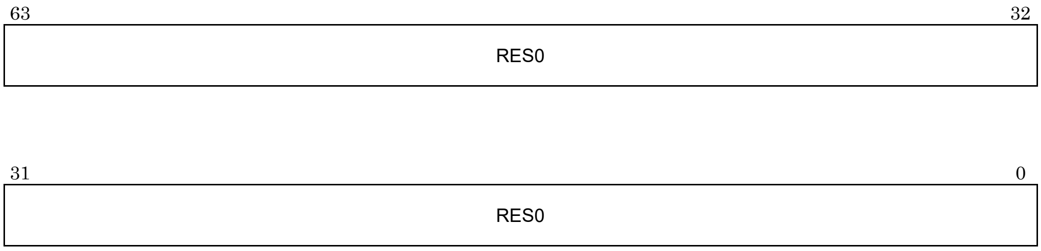
Bits	Name	Value	Function
[63:0]	-	0x0	Reserved, RES0

Field description of MAC/DIMM error nodes



Bits	Name	Value	Function
[63:0]	IMPLEMENTATION DEFINED	RW	This field is set to IMPLEMENTATION DEFINED code.

Field description of PCIe error nodes



Bits	Name	Value	Function
[63:0]	-	RW	Reserved, RES0. This field is always set to 0.

Address
Base + 0x028+64n

3.4.7. Error Record Miscellaneous Register 2 (ERR<n>MISC2)

Field description of SRD/IOD error nodes

63 32

RES0

31 0

RES0

Bits	Name	Value	Function
[63:0]	-	0x0	Reserved, RES0

Field description of MAC/DIMM error nodes

63 32

IMPLEMENTATION DEFINED

31 0

IMPLEMENTATION DEFINED

Bits	Name	Value	Function
[63:0]	IMPLEMENTATION DEFINED	RW	This field is set to IMPLEMENTATION DEFINED code.

Field description of PCIe error nodes

63 32

RES0

31	0
RES0	

Bits	Name	Value	Function
[63:0]	-	RW	Reserved, RES0. This field is always set to 0.

Address
Base + 0x030+64n

3.4.8. Error Record Miscellaneous Register 3 (ERR<n>MISC3)

Field description of SRD/IOD error nodes

63	32
TS	

31	0
TS	

Bits	Name	Value	Function
[63:0]	TS	RO	When the error was detected, the timestamp value is recorded.

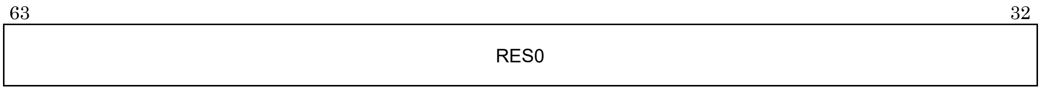
Field description of MAC/DIMM error nodes

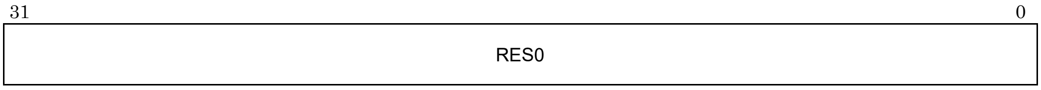
63	32
RES0	

31	0
RES0	

Bits	Name	Value	Function
[63:0]	-	0x0	Reserved, RES0

Field description of PCIe error nodes

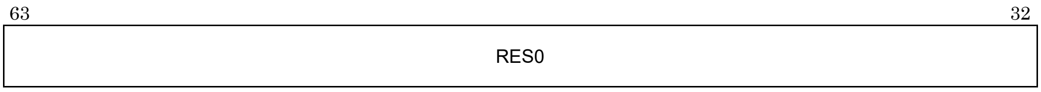


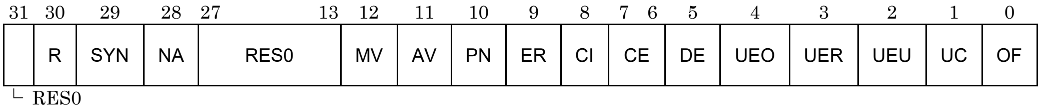


Bits	Name	Value	Function
[63:0]	-	RW	Reserved, RES0. This field is always set to 0.

Address
Base + 0x038+64n

3.4.9. Pseudo-fault Generation Feature Register (ERR<n>PFGF)





Field description of SRD/IOD error nodes

Bits	Name	Value	Function
[63:31]	-	0x0	Reserved, RES0
[30]	R	0x1	Error Generation Counter restart mode is implemented.
[29]	SYN	0x1	When an injected error is recorded, the node does not update the ERR<n>STATUS.{IERR, SERR} fields. ERR<n>STATUS.{IERR, SERR} are writable when ERR<n>STATUS.V is 0.
[28]	NA	0x1	The component fakes detection of the error spontaneously in the fault injection state.
[27:13]	-	0x0	Reserved, RES0
[12]	MV	0x0	When an injected error is recorded, the node always updates the ERR<n>MISC3 register.
[11]	AV	0x0	The node does not update ERR<n>ADDR register.
[10]	PN	0x0	The node does not support poison flag.

[9]	ER	0x0	The node does not support in-band error response.
[8]	CI	0x0	The node does not support critical error flag.
[7:6]	CE	0x0	When an injected error is recorded, the node does not update the ERR<n>STATUS.CE field.
[5]	DE	0x0	The node does not support Deferred Error.
[4]	UEO	0x1	The node supports the generation of Latent or Restartable errors.
[3]	UER	0x0	The node does not support the generation of Signaled or Recoverable errors.
[2]	UEU	0x0	The node does not support the generation of Unrecoverable errors.
[1]	UC	0x0	The node does not support Uncontainable Error.
[0]	OF	0x0	When an injected error is recorded, the node sets ERR<n>STATUS.OF according to the architecture-defined rules for setting the OF field.

Field description of MAC error nodes

Bits	Name	Value	Function
[63:31]	-	0x0	Reserved, RES0
[30]	R	0x1	Error Generation Counter restart mode is implemented.
[29]	SYN	0x1	When an injected error is recorded, the node does not update the ERR<n>STATUS.{IERR, SERR} fields. ERR<n>STATUS.{IERR, SERR} are writable when ERR<n>STATUS.V is 0.
[28]	NA	0x1	The component fakes detection of the error spontaneously in the fault injection state.
[27:13]	-	0x0	Reserved, RES0
[12]	MV	0x0	When an injected error is recorded, the node might update the ERR<n>MISC<m> registers.
[11]	AV	0x0	The node does not update ERR<n>ADDR register.
[10]	PN	0x1	When an injected error is recorded, ERR<n>STATUS.PN is set to ERR<n>PFGCTL.PN.
[9]	ER	0x0	The node does not support in-band error response.
[8]	CI	0x0	The node does not support critical error flag.
[7:6]	CE	0x0	When an injected error is recorded, the node does not update the ERR<n>STATUS.CE field.
[5]	DE	0x1	The node supports the generation of Deferred errors.
[4]	UEO	0x1	The node supports the generation of Latent or Restartable errors.
[3]	UER	0x0	The node does not support the generation of Signaled or Recoverable errors.
[2]	UEU	0x1	The node supports the generation of Unrecoverable errors.
[1]	UC	0x0	The node does not support Uncontainable Error.

[0]	OF	0x1	When an injected error is recorded, ERR<n>STATUS.OF is set to ERR<n>PFGCTL.OF.

Field description of DIMM error nodes

Bits	Name	Value	Function
[63:31]	-	0x0	Reserved, RES0
[30]	R	0x1	Error Generation Counter restart mode is implemented.
[29]	SYN	0x1	When an injected error is recorded, the node does not update the ERR<n>STATUS.{IERR, SERR} fields. ERR<n>STATUS.{IERR, SERR} are writable when ERR<n>STATUS.V is 0.
[28]	NA	0x1	The component fakes detection of the error spontaneously in the fault injection state.
[27:13]	-	0x0	Reserved, RES0
[12]	MV	0x1	When an injected error is recorded, ERR<n>STATUS.MV is set to ERR<n>PFGCTL.MV and ERR<n>MISC<m> is not updated.
[11]	AV	0x1	When an injected error is recorded, ERR<n>STATUS.AV is set to ERR<n>PFGCTL.AV and ERR<n>ADDR is not updated.
[10]	PN	0x1	When an injected error is recorded, ERR<n>STATUS.PN is set to ERR<n>PFGCTL.PN.
[9]	ER	0x0	The node does not support in-band error response.
[8]	CI	0x0	The node does not support critical error flag.
[7:6]	CE	0x3	The fault generation feature of the node allows generation of corrected errors.
[5]	DE	0x1	The node supports the generation of Deferred errors.
[4]	UEO	0x0	The node does not support the generation of Latent or Restartable errors.
[3]	UER	0x0	The node does not support the generation of Signaled or Recoverable errors.
[2]	UEU	0x0	The node does not support the generation of Unrecoverable errors.
[1]	UC	0x0	The node does not support Uncontainable Error.
[0]	OF	0x1	When an injected error is recorded, ERR<n>STATUS.OF is set to ERR<n>PFGCTL.OF.

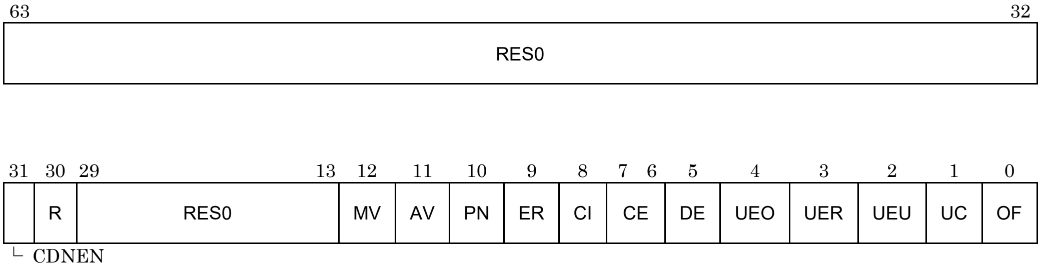
Field description of PCIe error nodes

Bits	Name	Value	Function
[63:31]	-	0x0	Reserved, RES0
[30]	R	0x1	Error Generation Counter restart mode is implemented.
[29]	SYN	0x1	When an injected error is recorded, the node does not update the ERR<n>STATUS.{IERR, SERR} fields. ERR<n>STATUS.{IERR, SERR} are writable when ERR<n>STATUS.V is 0.

[28]	NA	0x1	The component fakes detection of the error spontaneously in the fault injection state.
[27:13]	-	0x0	Reserved, RES0
[12]	MV	0x1	When an injected error is recorded, ERR<n>STATUS.MV is set to ERR<n>PFGCTL.MV and ERR<n>MISC<m> is not updated.
[11]	AV	0x0	The node does not update ERR<n>ADDR register.
[10]	PN	0x0	The node does not support poison flag.
[9]	ER	0x0	The node does not support in-band error response.
[8]	CI	0x0	The node does not support critical error flag.
[7:6]	CE	0x0	When an injected error is recorded, the node does not update the ERR<n>STATUS.CE field.
[5]	DE	0x0	The node does not support Deferred Error.
[4]	UEO	0x0	The node does not support the generation of Latent or Restartable errors.
[3]	UER	0x1	The node supports the generation of Signaled or Recoverable errors.
[2]	UEU	0x0	The node does not support the generation of Unrecoverable errors.
[1]	UC	0x0	The node does not support Uncontainable Error.
[0]	OF	0x1	When an injected error is recorded, ERR<n>STATUS.OF is set to ERR<n>PFGCTL.OF.

Address
Base + 0x800+64n

3.4.10. Pseudo-fault Generation Control Register (ERR<n>PFGCTL)



Field description of SRD/IOD error nodes

Bits	Name	Value	Function
[63:31]	-	0x0	Reserved, RES0

[31]	CDNEN	RW	Countdown Enable. See the Armv9-A arm[1].
[30]	R	RW	Error Generation Counter restart mode is implemented.
[29:13]	-	0x0	Reserved, RES0
[12]	MV	0x1	When an injected error is recorded, the node always updates the ERR<n>MISC3 register and sets ERR<n>STATUS.MV to 1.
[11]	AV	0x0	The node does not update ERR<n>ADDR register.
[10]	PN	0x0	The node does not support poison flag.
[9]	ER	0x0	The node does not support in-band error response.
[8]	CI	0x0	When an injected error is recorded, the node does not update the ERR<n>STATUS.CI field. This field is RAZ/WI.
[7:6]	CE	0x0	When an injected error is recorded, the node does not update the ERR<n>STATUS.CE field. This field is RAZ/WI.
[5]	DE	0x0	The node does not support Deferred Error.
[4]	UEO	RW	The node supports the generation of Latent or Restartable errors.
[3]	UER	0x0	The node does not support the generation of Signaled or Recoverable errors.
[2]	UEU	0x0	The node does not support the generation of Unrecoverable errors.
[1]	UC	0x0	The node does not support Uncontainable errors.
[0]	OF	0x0	When an injected error is recorded, the node sets ERR<n>STATUS.OF according to the architecture-defined rules for setting the OF field.

Field description of MAC error nodes

Bits	Name	Value	Function
[63:31]	-	0x0	Reserved, RES0
[31]	CDNEN	RW	Countdown Enable. See the Armv9-A arm[1].
[30]	R	RW	Error Generation Counter restart mode is implemented.
[29:13]	-	0x0	Reserved, RES0
[12]	MV	0x0	When an injected error is recorded, the node might update the ERR<n>MISC<m> registers.
[11]	AV	0x0	The node does not update ERR<n>ADDR register.
[10]	PN	RW	Poison flag. When an injected error is recorded, ERR<n>STATUS.PN is set to ERR<n>PFGCTL.PN.
[9]	ER	0x0	The node does not support in-band error response.
[8]	CI	0x0	When an injected error is recorded, the node does not update the ERR<n>STATUS.CI field. This field is RAZ/WI.
[7:6]	CE	0x0	When an injected error is recorded, the node does not update the ERR<n>STATUS.CE field. This field is RAZ/WI.
[5]	DE	RW	The node supports the generation of Deferred Error.
[4]	UEO	RW	The node supports the generation of Latent or Restartable errors.

[3]	UER	0x0	The node does not support the generation of Signaled or Recoverable errors.
[2]	UEU	RW	The node supports the generation of Unrecoverable errors.
[1]	UC	0x0	The node does not support Uncontainable errors.
[0]	OF	RW	Overflow flag. When an injected error is recorded, ERR<n>STATUS.OF is set to ERR<n>PFGCTL.OF.

Field description of DIMM error nodes

Bits	Name	Value	Function
[63:31]	-	0x0	Reserved, RES0
[31]	CDNEN	RW	Countdown Enable. See the Armv9-A arm[1].
[30]	R	RW	Error Generation Counter restart mode is implemented.
[29:13]	-	0x0	Reserved, RES0
[12]	MV	RW	Miscellaneous syndrome. When an injected error is recorded, ERR<n>STATUS.MV is set to ERR<n>PFGCTL.MV
[11]	AV	RW	Address syndrome. When an injected error is recorded, ERR<n>STATUS.AV is set to ERR<n>PFGCTL.AV.
[10]	PN	RW	Poison flag. When an injected error is recorded, ERR<n>STATUS.PN is set to ERR<n>PFGCTL.PN.
[9]	ER	0x0	The node does not support in-band error response.
[8]	CI	0x0	When an injected error is recorded, the node does not update the ERR<n>STATUS.CI field. This field is RAZ/WI.
[7:6]	CE	RW	The fault generation feature of the node allows generation of corrected errors. See the Armv9-A arm[1].
[5]	DE	RW	The node supports the generation of Deferred Error.
[4]	UEO	0x0	The node does not support the generation of Latent or Restartable errors.
[3]	UER	0x0	The node does not support the generation of Signaled or Recoverable errors.
[2]	UEU	0x0	The node does not support the generation of Unrecoverable errors.
[1]	UC	0x0	The node does not support Uncontainable errors.
[0]	OF	RW	Overflow flag. When an injected error is recorded, ERR<n>STATUS.OF is set to ERR<n>PFGCTL.OF.

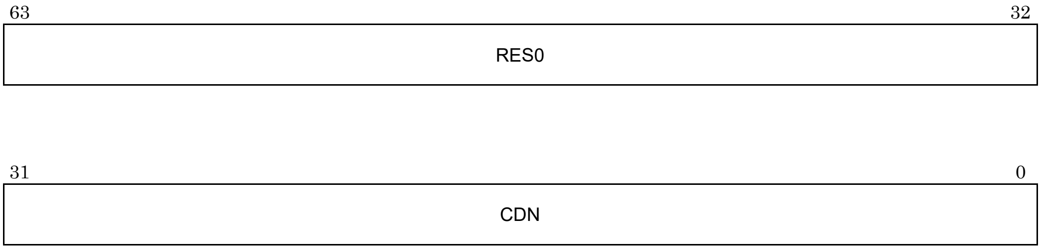
Field description of PCIe error nodes

Bits	Name	Value	Function
[63:31]	-	0x0	Reserved, RES0
[31]	CDNEN	RW	Countdown Enable. See the Armv9-A arm[1].

[30]	R	RW	Error Generation Counter restart mode is implemented.
[29:13]	-	0x0	Reserved, RES0
[12]	MV	0x1	When an injected error is recorded, the node always updates the ERR<n>MISC3 register and sets ERR<n>STATUS.MV to 1.
[11]	AV	0x0	The node does not update ERR<n>ADDR register.
[10]	PN	0x0	The node does not support poison flag.
[9]	ER	0x0	The node does not support in-band error response.
[8]	CI	0x0	When an injected error is recorded, the node does not update the ERR<n>STATUS.CI field. This field is RAZ/WI.
[7:6]	CE	0x0	When an injected error is recorded, the node does not update the ERR<n>STATUS.CE field. This field is RAZ/WI.
[5]	DE	0x0	The node does not support Deferred Error.
[4]	UEO	0x0	The node does not support the generation of Latent or Restartable errors.
[3]	UER	RW	The node supports the generation of Signaled or Recoverable errors.
[2]	UEU	0x0	The node does not support the generation of Unrecoverable errors.
[1]	UC	0x0	The node does not support Uncontainable errors.
[0]	OF	RW	Overflow flag. When an injected error is recorded, ERR<n>STATUS.OF is set to ERR<n>PFGCTL.OF.

Address
Base + 0x808+64n

3.4.11. Pseudo-fault Generation Countdown Register (ERR<n>PFGCDN)



Bits	Name	Value	Function
[63:32]	-	0x0	Reserved, RES0
[31:0]	CDN	RW	Countdown value. See the Armv9-A arm[1].

Address
Base + 0x810+64n

3.4.12. Error Group Status Register (ERRGSR)

63 56	55 32
RES0	S

31 0
S

Bits	Name	Value	Function
[63:56]	-	0x0	Reserved, RES0
[55:0]	S	RO	The status for error record. A read-only copy of ERR<n>STATUS.V. See the Armv9-A arm[1].

Address
Base + 0xE00

3.4.13. Error Group Implementation Identification Register (ERRIIDR)

31 20	19 16	15 12	11 0
ProductID	Variant	Revision	Implementer

Bits	Name	Value	Function
[31:20]	ProductID	0x03	Part number.
[19:16]	Variant	0x0	Component major revision.
[15:12]	Revision	0x0	Component minor revision.
[11:0]	Implementer	0x004	Contains the JEP106 code of the company that implemented the RAS component: 0x004: Fujitsu Ltd.

Address
Base + 0xE10

3.4.14. ERRIRQCR<n>, n=0-15

63–32
RES0

31–0
RES0

Bits	Name	Value	Function
[63:0]	-	0x0	Reserved, RES0

Address
Base + 0xE80+64n

3.4.15. Device Affinity Register (ERRDEVAFF)

63–40	39–32
RES0	Aff3

31	30	29–25	24	23–16	15–8	7–0
F0V	U	RES0	MT	Aff2	Aff1	Aff0

Bits	Name	Value	Function
[63:40]	-	0x0	Reserved, RES0
[39:32]	Aff3	0x0	The error group of error records does not have affinity with a PE. This field is RAZ.
[31]	F0V	0x0	The error group of error records does not have affinity with a PE. This field is RAZ.
[30]	U	0x0	The error group of error records does not have affinity with a PE. This field is RAZ.
[29:25]	-	0x0	Reserved, RES0

[24]	MT	0x0	The error group of error records does not have affinity with a PE. This field is RAZ.
[23:16]	Aff2	0x0	The error group of error records does not have affinity with a PE. This field is RAZ.
[15:8]	Aff1	0x0	The error group of error records does not have affinity with a PE. This field is RAZ.
[7:0]	Aff0	0x0	The error group of error records does not have affinity with a PE. This field is RAZ.

Address
Base + 0xFA8

3.4.16. Device Architecture Register (ERRDEVARCH)

31 21	20	19 16	15 12	11 0
ARCHITECT	PRESENT	REVISION	ARCHVER	ARCHPART

Bits	Name	Value	Function
[31:21]	ARCHITECT	0x23B	Architect. Defines the architect of the component: 0b01000111011: JEP106 continuation code 0x4, ID code 0x3B. Arm Limited.
[20]	PRESENT	0x1	DEVARCH present. 0b1: Device Architecture information present.
[19:16]	REVISION	0x1	Revision. This component compliant with RAS system architecture v1.1.
[15:12]	ARCHVER	0x0	Architecture Version. This component compliant with RAS system architecture v1.1.
[11:0]	ARCHPART	0xA00	Architecture Part. 0xA00: RAS system architecture

Address
Base + 0xFBC

3.4.17. Device Configuration Register (ERRDEVID)

31 16	15 0
RES0	NUM

Field description of SRD/IOD/PCIe error groups

Bits	Name	Value	Function
[31:16]	-	0x0	Reserved, RES0
[15:0]	NUM	0x1	The SRD, IOD, and PCIe error groups each contains one record.

Field description of MAC error group

Bits	Name	Value	Function
[31:16]	-	0x0	Reserved, RES0
[15:0]	NUM	0x2	The MAC error group contains two records.

Address
Base + 0xFC8

3.4.18. PIDR<n>, CIDR<n>

In the FUJITSU-MONAKA processor, PIDR<n> and CIDR<n> are not implemented, therefore register access is treated as RAZ/WI.

4. Memory map

Physical Address			Size	Description	Security	Type	Chip
0000_0000_0000			64GB	Chip Area (PKG#0-IOD)			(#1)
	0_0000_0000		7936KB	AP SRAM	Root	CN	0,5
	0_007C_0000		256KB	AP SRAM	NS	CN	0,5
	0_0080_0000		504MB	Reserved			0,5
	0_2000_0000		8MB	NI for MSS		D1	0,5
	0_2080_0000		8MB	NI for Peripheral		D1	0-9
	0_2100_0000		148MB	Reserved			0-9
	0_2A40_0000		92MB	Sub-system Peripherals		D1	0-9
		2A40_0000	64KB	IOD/SRD Platform Management Block (Root)	Root	D1	0-9
		2A41_0000	64KB	IOD/SRD Platform Management Block (Non-Secure)	NS	D1	0-9
		2A42_0000	64KB	UART#0 controller	CFG1	D1	0
		2A43_0000	64KB	UART#1 controller	CFG1	D1	0
		2A44_0000	64KB	UART#2 controller	CFG1	D1	0
		2A45_0000	64KB	UART#3 controller	CFG1	D1	0
		2A46_0000	64KB	UART#4 controller	CFG1	D1	0
		2A47_0000	64KB	GPIO#0 controller(#3)	CFG1	D1	0
		2A48_0000	64KB	GPIO#1 controller(#3)	CFG1	D1	0
		2A49_0000	64KB	GPIO#2 controller(#3)	CFG1	D1	0
		2A4A_0000	64KB	GPIO#3 controller(#3)	CFG1	D1	0
		2A4B_0000	64KB	SMBus#0 controller(#4)	CFG1	D1	0
		2A4C_0000	64KB	SMBus#1 controller(#4)	CFG1	D1	0
		2A4D_0000	64KB	SMBus#2 controller(#4)	CFG1	D1	0
		2A4E_0000	64KB	SMBus#3 controller(#4)	CFG1	D1	0
		2A4F_0000	3008KB	Reserved			0
		2A7E_0000	64KB	RNG0	Root	D1	1-4,6-9
		2A7F_0000	64KB	RNG1	Root	D1	1-4,6-9
		2A80_0000	64KB	AP Root Power Management	Root	D1	0,5
		2A81_0000	64KB	AP Non-Secure Energy and Power Monitor	NS	D1	0,5
		2A82_0000	896KB	Reserved			0,5
		2A90_0000	64KB	AP to SCP Non-secure MHU send frame : MHUv3	NS	D1	0,5
		2A91_0000	64KB	AP to SCP Non-secure MHU receive frame : MHUv3	NS	D1	0,5
		2A92_0000	64KB	AP to SCP Secure MHU send frame : MHUv3	S	D1	0,5
		2A93_0000	64KB	AP to SCP Secure MHU receive frame : MHUv3	S	D1	0,5
		2A94_0000	64KB	AP to SCP Root MHU send frame : MHUv3	Root	D1	0,5
		2A95_0000	64KB	AP to SCP Root MHU receive frame : MHUv3	Root	D1	0,5
		2A96_0000	128KB	Reseved			0,5
		2A98_0000	64KB	AP to SCP Non-secure MHU sender RAS	Root	D1	0,5
		2A99_0000	64KB	AP to SCP Non-secure MHU receiver RAS	Root	D1	0,5
		2A9A_0000	64KB	AP to SCP Secure MHU sender RAS	Root	D1	0,5
		2A9B_0000	64KB	AP to SCP Secure MHU receiver RAS	Root	D1	0,5
		2A9C_0000	64KB	AP to SCP Root MHU sender RAS	Root	D1	0,5
		2A9D_0000	64KB	AP to SCP Root MHU receiver RAS	Root	D1	0,5
		2A9E_0000	128KB	Reseved			0,5
		2AA0_0000	64KB	AP to MCP Non-secure MHU send frame : MHUv3	NS	D1	0,5
		2AA1_0000	64KB	AP to MCP Non-secure MHU receive frame : MHUv3	NS	D1	0,5
		2AA2_0000	64KB	AP to MCP Secure MHU send frame : MHUv3	S	D1	0,5
		2AA3_0000	64KB	AP to MCP Secure MHU receive frame : MHUv3	S	D1	0,5
		2AA4_0000	64KB	AP to MCP Root MHU send frame : MHUv3	Root	D1	0,5
		2AA5_0000	64KB	AP to MCP Root MHU receive frame : MHUv3	Root	D1	0,5
		2AA6_0000	128KB	Reserved			0,5
		2AA8_0000	64KB	AP to MCP Non-secure MHU sender RAS	Root	D1	0,5
		2AA9_0000	64KB	AP to MCP Non-secure MHU receiver RAS	Root	D1	0,5
		2AAA_0000	64KB	AP to MCP Secure MHU sender RAS	Root	D1	0,5
		2AAB_0000	64KB	AP to MCP Secure MHU receiver RAS	Root	D1	0,5
		2AAC_0000	64KB	AP to MCP Root MHU sender RAS	Root	D1	0,5
		2AAD_0000	64KB	AP to MCP Root MHU receiver RAS	Root	D1	0,5
		2AAE_0000	640KB	Reserved			0,5
		2AB0_0000	64KB	AP to RSE Non-secure MHU send frame : MHUv3	NS	D1	0,5

Physical Address	Size	Description	Security	Type	Chip
2AB1_0000	64KB	AP to RSE Non-secure MHU receive frame : MHUv3	NS	D1	0,5
2AB2_0000	64KB	AP to RSE Secure MHU send frame : MHUv3	S	D1	0,5
2AB3_0000	64KB	AP to RSE Secure MHU receive frame : MHUv3	S	D1	0,5
2AB4_0000	64KB	AP to RSE Root MHU send frame : MHUv3	Root	D1	0,5
2AB5_0000	64KB	AP to RSE Root MHU receive frame : MHUv3	Root	D1	0,5
2AB6_0000	64KB	AP to RSE Realm MHU send frame : MHUv3	Realm	D1	0,5
2AB7_0000	64KB	AP to RSE Realm MHU receive frame : MHUv3	Realm	D1	0,5
2AB8_0000	64KB	AP to RSE Non-secure MHU sender RAS	Root	D1	0,5
2AB9_0000	64KB	AP to RSE Non-secure MHU receiver RAS	Root	D1	0,5
2ABA_0000	64KB	AP to RSE Secure MHU sender RAS	Root	D1	0,5
2ABB_0000	64KB	AP to RSE Secure MHU receiver RAS	Root	D1	0,5
2ABC_0000	64KB	AP to RSE Root MHU sender RAS	Root	D1	0,5
2ABD_0000	64KB	AP to RSE Root MHU receiver RAS	Root	D1	0,5
2ABE_0000	64KB	AP to RSE Realm MHU sender RAS	Root	D1	0,5
2ABF_0000	64KB	AP to RSE Realm MHU receiver RAS	Root	D1	0,5
2AC0_0000	64KB	SCP(remote) to SCP(local) MHU Send	S	D1	0,5
2AC1_0000	128KB	Reserved			0,5
2AC3_0000	64KB	SCP(remote) to SCP(local) MHU Sender RAS	Root	D1	0,5
2AC4_0000	128KB	Reserved			0,5
2AC6_0000	64KB	MCP(remote) to MCP(local) MHU Send	S	D1	0,5
2AC7_0000	128KB	Reserved			0,5
2AC9_0000	64KB	MCP(remote) to MCP(local) MHU Sender RAS	Root	D1	0,5
2ACA_0000	128KB	Reserved			0,5
2ACC_0000	64KB	RSE(remote) to RSE(local) MHU Send	S	D1	0,5
2ACD_0000	128KB	Reserved			0,5
2ACF_0000	64KB	RSE(remote) to RSE(local) MHU Sender RAS	Root	D1	0,5
2AD0_0000	64KB	SPI0 controller#0 (CS#0)	Root	D1	0
2AD1_0000	64KB	SPI0 controller#1 (CS#1)	CFG1	D1	0
2AD2_0000	64KB	SPI1 controller (TPM)	CFG1	D1	0
2AD3_0000	64KB	Reserved			0
2AD4_0000	64KB	TPM access area	CFG1	D1	0
2AD5_0000	68288KB	Reserved			0
2F00_0000	4MB	Reserved for Shared SRAM (SCP, MCP & RSE) Local SCP/MCP/RSE through ATU			0,5
2F40_0000	12MB	Reserved			0,5
0_3000_0000	16MB	NI for GIC,TIM, ARM-RAS, SMMU			0-9
3000_0000	88KB	NI Configuration	Root/S	D1	0-9
3001_6000	16296KB	Reserved			0-9
0_3100_0000	16MB	IOD GIC700			0,5
3100_0000	64KB	GICD main page		D1	0,5
3101_0000	64KB	GICM message-based interrupts		D1	0,5
3102_0000	64KB	GIC trace and debug page		D1	0,5
3103_0000	64KB	GIC PMU page		D1	0,5
3104_0000	256KB x7	ITS#0-#6 pages (GIC)		D1	0,5
3120_0000	64KB	Alias to GICD		D1	0,5
3121_0000	14208KB	Reserved			0,5
0_3100_0000	16MB	SRD GIC700			1-4,6-9
3100_0000	64KB	GICD main page		D1	1-4,6-9
3101_0000	64KB	GICM message-based interrupts		D1	1-4,6-9
3102_0000	64KB	GIC trace and debug page		D1	1-4,6-9
3103_0000	64KB	GIC PMU page		D1	1-4,6-9
3104_0000	256KB	ITS#0 pages (GIC)		D1	1-4,6-9
3108_0000	256KB x36	GICR#0-#35 registers		D1	1-4,6-9
3198_0000	64KB	Alias to GICD		D1	1-4,6-9
3199_0000	6592KB	Reserved			1-4,6-9
0_3200_0000	16MB	SA_SMMU (SMMU for GIC, TIM, SYSC, SYSA)			0-9
3200_0000	1MB	TCU		D1	0-9
3210_0000	512KB	TBU for GIC		D1	0-9
3218_0000	512KB	TBU for SYSA		D1	0,5
3220_0000	14MB	Reserved			0-9
0_3300_0000	16MB	peripheral for SA			0-9
0_3300_0000	1MB	On CHIP Timers			0-9
3300_0000	64KB	AP_System Generic Counter control frame(CNTcontrol)	Root	D1	0-9
3301_0000	64KB	AP_System Generic Counter read frame	NS	D1	0
3302_0000	64KB	AP Generic Timer Control Frame(CNTCTL)	S,NS	D1	0
3303_0000	64KB	AP Secure Generic Timer control base frame	S	D1	0
3304_0000	64KB	AP Non-secure Generic Timer control base frame	NS	D1	0

Physical Address			Size	Description	Security	Type	Chip
		3305_0000	192KB	Reserved			0
		3308_0000	64KB	AP Non-secure WatchDog control frame	NS	D1	0
		3309_0000	64KB	AP Non-secure WatchDog refresh frame	NS	D1	0
		330A_0000	64KB	AP Root WatchDog control frame	Root	D1	0
		330B_0000	64KB	AP Root WatchDog refresh frame	Root	D1	0
		330C_0000	64KB	AP Secure/trusted WatchDog control frame	S	D1	0
		330D_0000	64KB	AP Secure/trusted WatchDog refresh frame	S	D1	0
		330E_0000	64KB	Reserved			
		330F_0000	64KB	TIM_CC	Root		0-9
	0_3310_0000		1MB	RAS_ERROR_GROUP Non-core			0-9
		3310_0000	64KB	Non-core	NS	D1	0-9
		3311_0000	960KB	Reserved			
	0_3320_0000		14MB	Reserved			0-9
	0_3400_0000		16MB	GIC700 ACE-CC		D1	0-9
	0_3500_0000		176MB	Reserved			0-9
	0_4000_0000		1MB x36	Core#0-#35 Processor Block			1-4,6-9
	0_4C00_0000		320MB	Reserved	—	—	—
	0_6000_0000		512MB	MMIOL	GPT	D3	0(#10)
	0_8000_0000		2GB	Main Memory	GPT	CN	0(#10)
	1_0000_0000		1GB	CMN GPV		D1	0-9
	1_4000_0000		1GB	Reserved	—	—	—
	1_8000_0000		256MB	MAC#0-#11 Reg(#5)		D1	0,5
	1_9000_0000		3.75GB	Reserved	—	—	—
	2_8000_0000		128MB x6	PCIe core#0~#5 Reg(#6)		D1	0,5
	2_B000_0000		5.25GB	Reserved	—	—	—
	4_0000_0000		8GB	Debug memory region		D1	0-9
	6_0000_0000		128MB	Flash ROM#0 access area	Root	NN	0
	6_0800_0000		128MB	Flash ROM#1 access area	CFG1	NN	0
	6_1000_0000		9984MB	Reserved	—	—	—
	8_8000_0000		1GB	[Reserved for ACS] PPU/ASD for ACS	—	—	—
	8_C000_0000		1GB	[Reserved for ACS] Trickbox registers for ACS	—	—	—
	9_0000_0000		28GB	Reserved	—	—	—
0010_0000_0000			64GB	Chip Area (PKG#0-SRD#0)			1
0020_0000_0000			64GB	Chip Area (PKG#0-SRD#1)			2
0030_0000_0000			64GB	Chip Area (PKG#0-SRD#2)			3
0040_0000_0000			64GB	Chip Area (PKG#0-SRD#3)			4
0050_0000_0000			64GB	Chip Area (PKG#1-IOD)			5
0060_0000_0000			64GB	Chip Area (PKG#1-SRD#0)			6
0070_0000_0000			64GB	Chip Area (PKG#1-SRD#1)			7
0080_0000_0000			64GB	Chip Area (PKG#1-SRD#2)			8
0090_0000_0000			64GB	Chip Area (PKG#1-SRD#3)			9
00A0_0000_0000			256MB	PCI NCI Memory Space2 (ECAM)	GPT	D2	(#2)
00A0_1000_0000			768MB	Reserved			
00A0_4000_0000			256GB	PCI NCI Memory Space3 (MMIOH)	GPT	D3	(#2)
00E0_4000_0000			127GB	Reserved			
0100_0000_0000			7TB+2GB	Reserved			
0800_8000_0000			56TB-2GB	Main memory	GPT	CN	(#2)
4000_0000_0000			64TB	CXL Memory	GPT	CN	(#2)
8000_0000_0000			128TB	Reserved			

Security:
Root/Realm/S/NS : Initialized to the corresponding security state by the system FW
GPT : Depending on the setting of the Granule Protection Table
CFG1 : Depending on the setting of the Access Protection Unit

Type:
CN : Cacheable Normal memory
NN : Non-cacheable Normal memory
D1/D2/D3 : Device memory

Chip:
0: PKG#0-IOD
1: PKG#0.SRD#0
2: PKG#0.SRD#1
3: PKG#0.SRD#2
4: PKG#0.SRD#3

5: PKG#1-IOD
6: PKG#1.SRD#0
7: PKG#1.SRD#1
8: PKG#1.SRD#2
9: PKG#1,SRD#3
(#1): The first 4GB of the address space contains regions that are implemented on other chips.
(#2): The physical implementations of these regions are mixed between PKG#0-IOD and PKG#1-IOD.
(#3): ACPI _HID for the GPIO controller is “FUJI200A”.
(#4): ACPI _HID for the SMBus controller is “FUJI200B”.
(#5): ACPI _HID for the Uncore PMU registers for MAC is “FUJI200C”.
(#6): ACPI _HID for the Uncore PMU registers for PCIe core is “FUJI200D”.
(#10): Only the region (offset 0) within PKG#0-IOD is valid. The physical implementations of these regions are mixed between PKG#0-IOD and PKG#1-IOD.

5. Interrupt assign

Type	Interrupt ID	Event	Edge/ Level	Security
SGI	0~15	SGI	Edge	
PPI	16	Error Recovery Interrupt for PE	Edge	NS(#1)
	17	Fault Handling Interrupt for PE	Edge	NS(#1)
	18	TRBIRQ	Level	*
	19	CNTHVS	Level	*
	20	CNTHPS	Level	*
	22	COMMIRQ	Level	*
	23	Performance Monitors Interrupt	Level	*
	24	CTIIRQ	Edge	*
	25	GIC Maintenance Interrupt	Level	*
	26	Overflow interrupt from CNTHP	Level	*
	27	Overflow interrupt from CNTV	Level	*
	28	Overflow interrupt from CNTHV	Level	*
	29	Overflow interrupt from CNTPS	Level	*
	30	Overflow interrupt from CNTP	Level	*
	31	Reserved	-	
	1056	PMF#1 CPPC Minimum Excursion	Edge	NS
	1057	PMF#1 CPPC Guaranteed Performance Change	Edge	NS
SPI	32	SRD0_INTREQERRNS	Level	NS
	33	SRD0_INTREQERRS	Level	S
	34	SRD0_INTREQFAULTNS	Level	NS
	35	SRD0_INTREQFAULTS	Level	S
	36	SRD0_INTREQMPAMERRNS	Level	NS
	37	SRD0_INTREQMPAMERRS	Level	S
	38	SRD0_INTREQMPAMERRRT	Level	Root
	39	SRD0_INTREQMPAMERRRL	Level	NS
	40	SRD0_INTREQPMU0	Level	NS
	41	SRD0_INTREQPMU1	Level	NS
	42	SRD0_CMN_PM_OVFL_INTR0	Level	NS
	43	SRD0_CMN_PM_OVFL_INTR1	Level	NS
	44	SRD0_CMN_PM_OVFL_INTR2	Level	NS
	45	SRD0_CMN_PM_OVFL_INTR3	Level	NS
	46	SRD0_CMN_PM_OVFL_INTR4	Level	NS
	47	SRD0_CMN_PM_OVFL_INTR5	Level	NS
	48	SRD0_CMN_PM_OVFL_INTR6	Level	NS
	49	SRD0_CMN_PM_OVFL_INTR7	Level	NS
	50	SRD0_CMN_PM_OVFL_INTR8	Level	NS
	51	SRD0_CMN_PM_OVFL_INTR9	Level	NS
	52	SRD0_CMN_PM_OVFL_INTR10	Level	NS
	53	SRD0_CMN_PM_OVFL_INTR11	Level	NS
	54	SRD0_CMN_PM_OVFL_INTR12	Level	NS
	55	SRD0_CMN_PM_OVFL_INTR13	Level	NS
	56	SRD0_CMN_PM_OVFL_INTR14	Level	NS
	57	SRD0_CMN_PM_OVFL_INTR15	Level	NS
	58	SRD0_CMN_PM_OVFL_INTR16	Level	NS
	59	SRD0_CMN_PM_OVFL_INTR17	Level	NS
	60	SRD0_gicd_pmu_int	Level	NS
	61	SRD0_tcu_cmd_sync_irpt_ns	Edge	NS
	62	SRD0_tcu_cmd_sync_irpt_r	Edge	NS
	63	SRD0_tcu_cmd_sync_irpt_s	Edge	S
	64	SRD0_tcu_event_q_irpt_ns	Edge	NS
	65	SRD0_tcu_event_q_irpt_r	Edge	NS
	66	SRD0_tcu_event_q_irpt_s	Edge	S
	67	SRD0_tcu_global_irpt_ns	Edge	NS
	68	SRD0_tcu_global_irpt_r	Edge	NS
	69	SRD0_tcu_global_irpt_s	Edge	S

Type	Interrupt ID	Event	Edge/ Level	Security
	70	SRD0_tcu_gpf_far	Edge	Root
	71	SRD0_tcu_gpt_cfg_far	Edge	Root
	72	SRD0_tcu_pmu_irpt	Edge	NS
	73	SRD0_tcu_pri_q_irpt_ns	Edge	NS
	74	SRD0_tcu_pri_q_irpt_r	Edge	NS
	75	SRD0_tcu_ras_lt_eri	Level	NS(#1)
	76	SRD0_tcu_ras_lt_fhi	Level	NS(#1)
	77	SRD0_tbu0_ace_lite_pmu_irpt	Edge	NS
	78	SRD0_tbu0_ace_lite_ras_lt_eri	Level	NS(#1)
	79	SRD0_tbu0_ace_lite_ras_lt_fhi	Level	NS(#1)
	80	SRD0_CD_GIC_nPMUINTERRUPT	Level	NS
	81	SRD0 Error Recovery Interrupt for non-PE	Level	NS(#1)
	82	SRD0 Fault Handling Interrupt for non-PE	Level	NS(#1)
	83	SRD0_SYSC2GIC_P_CD0_PMU_INTR	Level	NS
	84	SRD0_SYSC2GIC_P_CD1_PMU_INTR	Level	NS
	85	SRD0_SYSC2GIC_P_PD0_INTR	Level	S
	86	SRD0_SYSC2GIC_P_PD0_NS_INTR	Level	NS
	87~95	Reserved		
	96~159	SRD1*		
	160~223	SRD2*		
	224~287	SRD3*		
	288	IOD0_INTREQERRNS	Level	NS
	289	IOD0_INTREQERRS	Level	S
	290	IOD0_INTREQFAULTNS	Level	NS
	291	IOD0_INTREQFAULTS	Level	S
	292	IOD0_INTREQMPAMERRNS	Level	NS
	293	IOD0_INTREQMPAMERRS	Level	S
	294	IOD0_INTREQMPAMERRRT	Level	Root
	295	IOD0_INTREQMPAMERRRL	Level	NS
	296	IOD0_INTREQPMU0	Level	NS
	297	IOD0_INTREQPMU1	Level	NS
	298	IOD0_INTREQPMU2	Level	NS
	299	IOD0_INTREQPMU3	Level	NS
	300	IOD0_CMN_PM_OVFL_INTR0	Level	NS
	301	IOD0_CMN_PM_OVFL_INTR1	Level	NS
	302	IOD0_CMN_PM_OVFL_INTR2	Level	NS
	303	IOD0_CMN_PM_OVFL_INTR3	Level	NS
	304	IOD0_gicd_pmu_int	Level	NS
	305	IOD0_tcu_cmd_sync_irpt_ns	Edge	NS
	306	IOD0_tcu_cmd_sync_irpt_r	Edge	NS
	307	IOD0_tcu_cmd_sync_irpt_s	Edge	S
	308	IOD0_tcu_event_q_irpt_ns	Edge	NS
	309	IOD0_tcu_event_q_irpt_r	Edge	NS
	310	IOD0_tcu_event_q_irpt_s	Edge	S
	311	IOD0_tcu_global_irpt_ns	Edge	NS
	312	IOD0_tcu_global_irpt_r	Edge	NS
	313	IOD0_tcu_global_irpt_s	Edge	S
	314	IOD0_tcu_gpf_far	Edge	Root
	315	IOD0_tcu_gpt_cfg_far	Edge	Root
	316	IOD0_tcu_pmu_irpt	Edge	NS
	317	IOD0_tcu_pri_q_irpt_ns	Edge	NS
	318	IOD0_tcu_pri_q_irpt_r	Edge	Realm
	319	IOD0_tcu_ras_lt_eri	Level	NS(#1)
	320	IOD0_tcu_ras_lt_fhi	Level	NS(#1)
	321	IOD0_tbu0_ace_lite_pmu_irpt	Edge	NS
	322	Reserved		
	323	IOD0_tbu0_ace_lite_ras_lt_eri	Level	NS(#1)
	324	Reserved		
	325	IOD0_tbu0_ace_lite_ras_lt_fhi	Level	NS(#1)
	326	Reserved		
	327	IOD0_CD_GIC_nPMUINTERRUPT	Level	NS
	328	IOD0 Secure System wakeup timer#0	Edge	S
	329	IOD0 NonSecure System wakeup timer#1	Edge	NS
	330	IOD0 Root Watchdog timer WS0	Edge	Root
	331	IOD0 Secure Watchdog timer WS0	Edge	S
	332	IOD0 Nonsecure Watchdog timer WS0	Edge	NS

Type	Interrupt ID	Event	Edge/ Level	Security
	333	IOD0 Nonsecure Watchdog timer WS1	Edge	Root
	334	IOD0 Error Recovery Interrupt for non-PE	Level	NS(#1)
	335	IOD0 Fault Handling Interrupt for non-PE	Level	NS(#1)
	336	IOD0_SYSC2GIC_P_UART0_INTR	Level	CFG1
	337	IOD0_SYSC2GIC_P_UART1_INTR	Level	CFG1
	338	IOD0_SYSC2GIC_P_UART2_INTR	Level	CFG1
	339	IOD0_SYSC2GIC_P_UART3_INTR	Level	CFG1
	340	IOD0_SYSC2GIC_P_UART4_INTR	Level	CFG1
	341	IOD0_SYSC2GIC_P_GPIO0_INTR	Level	CFG1
	342	IOD0_SYSC2GIC_P_GPIO1_INTR	Level	CFG1
	343	IOD0_SYSC2GIC_P_GPIO2_INTR	Level	CFG1
	344	IOD0_SYSC2GIC_P_GPIO3_INTR	Level	CFG1
	345	IOD0_SYSC2GIC_P_SMB0_INTR	Level	CFG1
	346	IOD0_SYSC2GIC_P_SMB1_INTR	Level	CFG1
	347	IOD0_SYSC2GIC_P_SMB2_INTR	Level	CFG1
	348	IOD0_SYSC2GIC_P_SMB3_INTR	Level	CFG1
	349	IOD0_SYSC2GIC_P_SPI00_INTR	Level	CFG1
	350	IOD0_SYSC2GIC_P_SPI01_INTR	Level	CFG1
	351	IOD0_SYSC2GIC_P_SPI1_INTR	Level	CFG1
	352	IOD0_SYSC2GIC_P_CD0_PMU_INTR	Level	NS
	353	IOD0_SYSC2GIC_P_CD1_PMU_INTR	Level	NS
	354	IOD0_SYSC2GIC_P_PD0_INTR	Level	S
	355	IOD0_SYSC2GIC_P_PD0_NS_INTR	Level	NS
	356	IOD0_MCP2APMHU_NS	Level	NS
	357	IOD0_MCP2APMHU_S	Level	S
	358	IOD0_MCP2APMHU_RT	Level	Root
	359	IOD0_SCP2APMHU_NS	Level	NS
	360	IOD0_SCP2APMHU_S	Level	S
	361	IOD0_SCP2APMHU_RT	Level	Root
	362	IOD0_RSE2APMHU_NS	Level	NS
	363	IOD0_RSE2APMHU_S	Level	S
	364	IOD0_RSE2APMHU_RT	Level	Root
	365	IOD0_RSE2APMHU_RL	Level	NS
	366	IOD0_PD_0_INTERRUPT	Level	S
	367	IOD0_PD_0_NS_INTERRUPT	Level	NS
	368	IOD0_CLK_APB_nPMUINTERRUPT	Level	NS
	369	IOD0_CLK_MSS_nPMUINTERRUPT	Level	NS
	370	IOD0_CLK_PCI_nPMUINTERRUPT	Level	NS
	371	IOD0_RSE INTERRUPT	Level	NS
	372	IOD0_SCP INTERRUPT	Level	NS
	373	IOD0_MCP INTERRUPT	Level	NS
	374	IOD0_ras_lt_fhi	Level	NS(#1)
	375	IOD0_ras_lt_eri	Level	NS(#1)
	376	IOD0_ras_lt_cri	Level	NS
	377	IOD0_ras_lt_irpt_v	Level	NS
	378	IOD0_pmu_irpt	Edge	NS
	379	IOD0_crit_err	Edge	NS
	380	IOD0 TPMI (TPM Interrupt)	Level	NS
	381	IOD0 NMI	Edge	NS
	382~399	Reserved		
	400	IOD0 MAC ch00 Performance Monitors Interrupt	Level	NS
	401	IOD0 MAC ch00 MAC Error Recovery Interrupt for non-PE	Level	NS(#1)
	402	IOD0 MAC ch00 DIMM Error Recovery Interrupt for non-PE	Level	NS(#1)
	403	IOD0 MAC ch00 MAC Fault Handling Interrupt for non-PE	Level	NS(#1)
	404	IOD0 MAC ch00 DIMM Fault Handling Interrupt for non-PE	Level	NS(#1)
	405	IOD0 MAC ch00 Non-secure MPAM error interrupt	Level	NS
	406	IOD0 MAC ch00 Secure MPAM error interrupt	Level	S
	407	IOD0 MAC ch00 Root MPAM error interrupt	Level	Root
	408	IOD0 MAC ch00_Realm MPAM error interrupt	Level	NS
	409	Reserved		
	410~419	IOD0 MAC ch01 *		
	420~429	IOD0 MAC ch10 *		
	430~439	IOD0 MAC ch11 *		
	440~449	IOD0 MAC ch20 *		
	450~459	IOD0 MAC ch21 *		
	460~469	IOD0 MAC ch30 *		

Type	Interrupt ID	Event	Edge/ Level	Security
	470~479	IOD0 MAC ch31 *		
	480~489	IOD0 MAC ch40 *		
	490~499	IOD0 MAC ch41 *		
	500~509	IOD0 MAC ch50 *		
	510~519	IOD0 MAC ch51 *		
	520~529	IOD0 MAC ch60 *		
	530~539	IOD0 MAC ch61 *		
	540~549	IOD0 MAC ch70 *		
	550~559	IOD0 MAC ch71 *		
	560~569	IOD0 MAC ch80 *		
	570~579	IOD0 MAC ch81 *		
	580~589	IOD0 MAC ch90 *		
	590~599	IOD0 MAC ch91 *		
	600~609	IOD0 MAC cha0 *		
	610~619	IOD0 MAC cha1 *		
	620~629	IOD0 MAC chb0 *		
	630~639	IOD0 MAC chb1 *		
	640	IOD0 PCIE0 PCIINTA	Level	NS
	641	IOD0 PCIE0 PCIINTB	Level	NS
	642	IOD0 PCIE0 PCIINTC	Level	NS
	643	IOD0 PCIE0 PCIINTD	Level	NS
	644	IOD0 PCIE0 SMMU, CMD_SYNC_NS	Edge	NS
	645	IOD0 PCIE0 SMMU, CMD_SYNC_R	Edge	NS
	646	IOD0 PCIE0 SMMU, EVENTQ_NS	Edge	NS
	647	IOD0 PCIE0 SMMU, EVENTQ_R	Edge	NS
	648	IOD0 PCIE0 SMMU, GROBAL_NS	Edge	NS
	649	IOD0 PCIE0 SMMU, GROBAL_R	Edge	NS
	650	IOD0 PCIE0 SMMU, GPF_FAR	Edge	Root
	651	IOD0 PCIE0 SMMU, GPF_CFG_FAR	Edge	Root
	652	IOD0 PCIE0 SMMU, PRIQ_NS	Edge	NS
	653	IOD0 PCIE0 SMMU, PRIQ_R	Edge	NS
	654	IOD0 PCIE0 ARMRAS/SMMU, TCU-FHI	Level	NS(#1)
	655	IOD0 PCIE0 ARMRAS/SMMU, TCU-ERI	Level	NS(#1)
	656	IOD0 PCIE0 ARMRAS/SMMU, TBU0-FHI	Level	NS(#1)
	657	IOD0 PCIE0 ARMRAS/SMMU, TBU0-ERI	Level	NS(#1)
	658	IOD0 PCIE0 ARMRAS/SMMU, TBU1-FHI	Level	NS(#1)
	659	IOD0 PCIE0 ARMRAS/SMMU, TBU1-ERI	Level	NS(#1)
	660	IOD0 PCIE0 ARMRAS/PCI, FHI	Level	NS(#1)
	661	IOD0 PCIE0 ARMRAS/PCI, ERI	Level	NS(#1)
	662	IOD0 PCIE0 PMU/NI800, CD0-INT	Level	NS
	663	IOD0 PCIE0 PMU/NI800, CD1-INT	Level	NS
	664	IOD0 PCIE0 PMU/SMMU, TCU-INT	Edge	NS
	665	IOD0 PCIE0 PMU/SMMU, TBU0-INT	Edge	NS
	666	IOD0 PCIE0 PMU/SMMU, TBU1-INT	Edge	NS
	667	IOD0 PCIE0 PMU/PCI, INT	Edge	NS
	668	IOD0 PCIE0 IDE0 CM0	Level	Root
	669	IOD0 PCIE0 IDE0 CM1	Level	Root
	670	IOD0 PCIE0 IDE0 CM2	Level	Root
	671	IOD0 PCIE0 IDE0 CM3	Level	Root
	672	IOD0 PCIE0 IDE0 IO0	Level	Root
	673	IOD0 PCIE0 IDE0 IO1	Level	Root
	674	IOD0 PCIE0 IDE1 CM0	Level	Root
	675	IOD0 PCIE0 IDE1 CM1	Level	Root
	676	IOD0 PCIE0 IDE1 CM2	Level	Root
	677	IOD0 PCIE0 IDE1 CM3	Level	Root
	678	IOD0 PCIE0 IDE1 IO0	Level	Root
	679	IOD0 PCIE0 IDE1 IO1	Level	Root
	680	IOD0 PCIE0 DTI0	Level	NS
	681	IOD0 PCIE0 DTI1	Level	NS
	682~685	Reserved		
	686~731	IOD0 PCIE1 *		
	732~777	IOD0 PCIE2 *		
	778~823	IOD0 PCIE3 *		
	824~869	IOD0 PCIE4 *		
	870~915	IOD0 PCIE5 *		
	916~927	Reserved		

Type	Interrupt ID	Event	Edge/ Level	Security
	4096~4159	SRD4*		
	4160~4223	SRD5*		
	4224~4287	SRD6*		
	4288~4351	SRD7*		
	4352~4991	IOD1 *		

Security:

Root/Realm/S/NS	: Initialized to the corresponding security state by the system FW
CFG1	: Depending on the setting of the Access Protection Unit
*	: Depending on the state of the PE when the interrupt is generated
Blank	: Expected to be initialized by software
(#1)	: Changed to "Root" during Firmware first RAS handling